Security Guidance

For Critical Areas of Focus in Cloud Computing v5

cloud security alliance®

The permanent and official location for the CSA Security Guidance Working Group is https://cloudsecurityalliance.org/research/working-groups/security-guidance

Acknowledgments

Lead Authors

Rich Mogull
Mike Rothman

Contributors

Jackie Donnelly
Moshe Ferber
Larry Hughes
Michael Roza
Peter van Eijk

Reviewers

Mohammad Aamir
Frank Addo
Daniel Adjorlolo
Hafiz Ahmed Sheikh Adnan
Ilango Allikuzhi
Shonnie Almeida J.
Babs Alo
Aakash Alurkar
Agbu Amachundi Enoch
Stephen Amolo
Divya Aradhya
Robyn Bailey
Adeel Bakht
Suramya Bakshi
Mohamed Balushi Al
Vinay Bansal
Robin Basham
Myriam Batista
Allen Baylis
Renu Bedi
Paul Benedek
Bachir Benyammi
Jamie Beth
Shirin Bhambhani
Roberto Bonalumi
Karl Brooks
Jasper Brouwer
Amit Butail
Varun Carlay
Dhanushraj Chandrahasan
Senthilkumar Chandrasekaran
Akshay Chandrasekaran Sekar
Shankar Chebrolu
Anand Chirathadam Abraham
John Chiu
Anand Choksi
Vipul Dabhi
Joseph Dacuma
Michel-Ange Dagrain
Thomas Defise
Neelima Devana
Mankirat Dhodi Singh
Balaram Dhulipudi
Dr. Ivan Djordjevic
Ivan Djordjevic
Moses Dlamini
Keinaz Domingo N.
David Dorsey
Rob Doyon
Vinay Dubey
Swapna Dulganti
Nielet D'Mello
Mohamed Elbashir
Mahmood Elrefai
Joseph Emerick
Dr. Marco Ermini
Kingsley Ezeocha
Ahmed Fawzy
Lorena Ferreyro
Kenneth Ferris
Jonathan Fessenden
Jose Figueredo-Maseda C.
Elaine Flesch
Fernando Fonseca
Park Foreman
Adame Frances
Aadithya Francis

André Gaio Alexandre
Luca Gattobigio
Viktor Gazdag
Jan Gerst
Hussein Ghazy
Tulika Ghosh
Ricardo Giorgi
Andriana Gkaniatsou
Saurabh Goswami
Trevor Gregorio
Nageswara Gude Rao
Madhu Guthikonda
Ahmed Harris
Lyle Hearne
Johnny Hernandez
Dirce Hernandez Eduardo
Aldo Hernández Villaseca
Moreno Hill Sint
Matthew Hoerig
Abdulsalam Ibrahim B.
Ricci Ieong
Frank Iheonu
Arron Johnson
Rahul K
Prasannakumar K G
Patrick Kabongo B.
Nithin Kadumberi Mohan Thattiot
Ruchi Kandpal
Sivakumar Karthikeyan
Shakthi Kathirvelu Priya
Sunil Katwal
Arpitha Kaushik
Alon Kendler
Rohit Khosla
Vana Khurana
Jari Kiero
Brenda Killingsworth L.
Morgan King
Samantha Kloos-Kilkens
Simon Kok
Vivek Krishnan
Sunil Kumar
Francois Laas
Hadir Labib
Daniel Lai
Raymond Lai
Law Lain Chamber
Sundas Latif
Seshagirirao Lekkala
Yutao Ma
Stephen Macomber
Yuvaraj Madheswaran
Niclas Madsen
Ahmed Mahmoud Nabil
Vaibhav Malik
Mohamed Malki
Cecil Martin
Marcus Maxwell
Bilal Mazhar
Mark McDonagh
José Medina Carlos Vargas
Santiago Medranda
Ashish Mehta
Shobhit Mehta
Andre Mess
Enida Metaj
Akhil Mittal
Adeeb Mohammed
Victor Monga
Kenneth Moras
Masahiro Morozumi
Andrew Morrow B.
Venkata Nedunoori
Harry Ngai
Fredrick Ogonda
Esborn Okero
Opeyemi Onifade
Joseph Orsetto
John Oseh B.
Jed Owens
Iyiola Oyinloye
Mayur Pahwa
Govindaraj Palanisamy
Meghana Parwate
Vaibhav Patkar
Martino Pavone
Eric Peeters
Eliza Popa
Kunal Pradhan
Ramon Quimesó Domingo
Adnan Rafique
Sonali Rajesh ZoIT R
Marappan Ramiah
Alex Rebo
Aldo Richner
Rangel Rodrigues

Vishakha Sadhwani
Shahid Saleem
Joshua Salvador
Mukund Sarma
Patnana Sayesu
Davide Scatto
Thomas Schmidt
Michael Schmitz
Kg.Seow
Vikrant Shah
Rakesh Sharma
Alex Sharpe
Akshay Shetty
Dr. Ian Silvester
Ryan Simon
Gurpratap Singh
Gaurav Singh
Anamika Singh
Serenity Smile
Heinrich Smit
Jorge Soboredo González
Silvano Sogus
Mikhail Sokolov
Dr. Chantal Spleiss
Dr. Manish Srivastava Kumar
Kevin Stander
Roy Stultiens
Yuanji Sun
Pratibha Swamy
Manjunath T A
Mohammed Tanveer
Billy Teow
Kim Tham Fui
Timothy Thatcher
Michael Theriault
Larry Timmons
Chee Tiong
Ilia Tivin
Wiem Tounsi
Micheal Troutman
Nsikak-Abasi Una Shammah
Pieter VanIperen
Ashish Vashishtha
Peter Ventura
Vaishnav Vijayakumar
Antonio Villamor Magallanes Jr
Alex Webling
Henry Werchan
Udith Wickramasuriya
Pawel Wilczynski
Rini Wilson
Wai Wong Kong
Ben Woods
Ezra Woods
James Yankelvich
Tsutomu Yoneyama
Bader Zyoud
Dennis de Caes
Peter van Loon
Tiaan van Schalkwyk

CSA Global Staff

Judy Bagwell
Hillary Baron
Marina Bregkou
Josh Buker
Daniele Catteddu
Emily Everett
Ryan Gifford
Frank Guanco
Sean Heide
Erik Johnson
Alex Kaluza
Claire Lehnert
Stephen Lumpe
Cion Mensidor
Hannah Rock
Andy Ruth
Anna Schorr Campbell
Stephen Smith
Adriano Sverko
John Yeoh

Table of Contents

Introduction to Security Guidance v5

Welcome to the fifth version of the Cloud Security Alliance's *Security Guidance for Critical Areas of Focus in Cloud Computing, or Security Guidance for short.* The rise of cloud computing as an ever-evolving technology brings with it a number of opportunities and challenges. With this document, we aim to provide both guidance and inspiration to support business goals while managing and mitigating the risks associated with the adoption of cloud computing technology.

The Cloud Security Alliance promotes implementing best practices for providing security assurance within the domain of cloud computing and has delivered a practical, actionable roadmap for organizations seeking to adopt the cloud paradigm. The fifth version of the *Security Guidance* is built on previous iterations of the *Security Guidance*, dedicated research, and public participation from the Cloud Security Alliance members, working groups, and the industry experts within our community. This version incorporates advances in cloud, security, and supporting technologies; reflects on real-world cloud security practices; integrates the latest Cloud Security Alliance research projects; and offers guidance for related technologies.

The advancement toward secure cloud computing requires active participation from a broad set of globally-distributed stakeholders. CSA brings together this diverse community of industry partnerships, international chapters, working groups, and individuals. We are profoundly grateful to all who contributed to this release.

Please visit cloudsecurityalliance.com to learn how you can work with us to identify and promote best practices to ensure a secure cloud computing environment.

Best regards,

Jim Reavis
Chief Executive Officer (CEO)
Cloud Security Alliance

Illena Armstrong
President
Cloud Security Alliance

Domain 1: Cloud Computing Concepts & Architectures

This domain provides the conceptual framework for the rest of the Cloud Security Alliance (CSA) *Security Guidance*. It describes and defines cloud computing, sets out baseline terminology, and details the overall controls, deployment, and architectural models used in the rest of the document.

Cloud computing can be viewed from various perspectives, such as a technology, a collection of technologies, an operational model, a business model, or an economic paradigm, just to name a few. Cloud computing is transformative and disruptive to legacy computing systems and has taken over as the dominant digital transformation model. While the reference models included in the early versions of the CSA CCSK *Security Guidance* are still relevant, they require updates to reflect ongoing advancements (e.g., new tools and technologies from cloud service providers (CSP), Zero Trust, AI, and evolving practices) as the industry matures. Even with this update, the ongoing rise of automation and artificial intelligence capabilities cannot account for all evolution in the coming years.

Cloud computing can provide significant agility, resiliency, security, and economic benefits. However, these benefits are only materialized if cloud models are properly understood and adopted, and cloud architectures and practices are aligned with the features and capabilities of cloud platforms. A cloud service customer (CSC[1]) migrating an existing application or asset by simply moving it to a CSP without any changes (known as rehosting or "lift-and-shift") will often not provide the expected agility, resiliency, and security, all while increasing costs. In short, the benefits of cloud computing are closely tied to understanding the proper use of cloud computing models, cloud-native capabilities, and services.

This domain aims to build the foundation on which the rest of this guide and its recommendations are based. The intent is to provide a common language and understanding of cloud computing, while highlighting the differences between cloud and traditional computing. In addition to the already stated cloud benefits, this domain will help guide cloud security professionals and other relevant stakeholders toward adopting cloud-approaches that ensure a better security posture

The Cloud Security Alliance is not setting out to create an entirely new taxonomy or reference model. Our objective is to distill and harmonize existing models – most notably the work in NIST SP 800-145[2], ISO/IEC 22123-1:2023[3], and ISO/IEC 22123-2:2023[4] – focusing on the most relevant security considerations for professionals working within the cloud computing field.
Additional references are provided to further enhance the understanding of the fundamental principles and explore specific topics and practical strategies for implementing cloud security practices.

[1] The acronym CSC is used interchangeably to mean any cloud service customer, cloud service consumer, or cloud service client.
[2] NIST. (2011) SP 800-145: *The NIST Definition of Cloud Computing*
[3] ISO/IEC. (2023) 22123-1:2023: Information technology - Cloud computing Part 1: Vocabulary
[4] ISO/IEC. (2023) 22123-2:2023: Information technology - Cloud computing Part 2: Concepts

Learning Objectives

In this domain, you will learn to:

- Define cloud computing.
- Identify cloud computing models.
- Recognize reference and architecture models in cloud computing.
- Understand cloud security scope, responsibilities, and models.

1.1 Defining Cloud Computing

Cloud computing is an operational model and a set of technologies used to manage shared pools of computing resources through abstraction of compute, network, storage, etc. The cloud model envisions a world where components and resources can be rapidly orchestrated, provisioned, implemented, scaled up or down, and decommissioned, providing an on-demand utility-like model for allocation and consumption. Benefits include stakeholder collaboration, agility, elasticity, availability, resiliency, and cost reduction.

The following are definitions of cloud computing according to *The U.S. National Institute of Standards and Technology (NIST) and The International Organization for Standardization (ISO) and International Electrotechnical Commission (IEC):*

NIST SP 800-145 defines cloud computing as: *"Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction."*[5]

ISO/IEC 22123-1:2023 defines cloud computing as: *"Paradigm for enabling network access to a scalable and elastic pool of shareable physical or virtual resources with self-service provisioning*[6] *and administration on-demand."*[7]

A simpler way of describing the cloud is that it takes a set of resources, such as processors and memory, and puts them into a big pool (in this case, using virtualization[8]). CSCs request resources they need out of the pool, based on their requirements (such as 8 CPUs and 16 GB of memory). The underlying cloud computing technology orchestrates those resources to the CSC, who then connects to and uses the resources over the network. When the CSC is done, they can release the resources back into the pool for others to use.

A cloud can consist of nearly any computing resource, ranging from our raw infrastructure examples of processors, memory, and networks, to higher-level software resources like databases and applications. For

[5] NIST. (2011) SP 800-145: *The NIST Definition of Cloud Computing.*
[6] *Self-service provisioning refers to the provisioning of resources provided to cloud services performed by CSCs through automated means. Examples of resources include servers, operating systems, networks, software, applications, and storage equipment.*
[7] ISO/IEC. (2023) Information Technology - Cloud Computing - Part 1: Vocabulary.
[8] NIST. (2018) SP 800-125A: Security Recommendations for Hypervisor Deployment on Servers *Virtualization.*

example, subscribing to a customer relationship management (CRM) application for 500 employees on a service shared by hundreds of other organizations is just as much cloud computing as launching 100 remote servers on a compute cloud.

1.1.1 Abstraction & Orchestration

The key concepts that enable the cloud environment are abstraction and orchestration. Resources are abstracted from the underlying physical infrastructure to create pools of resources, and orchestration (and automation) is used to coordinate, allocate, and deliver resources from pools to CSCs. Intrinsic to this is a level of standardization, so that all CSCs are essentially getting the same functional service they can, then integrate in their own flexible way. As you will see, these two concepts create all the essential characteristics we use to define something as a "cloud."

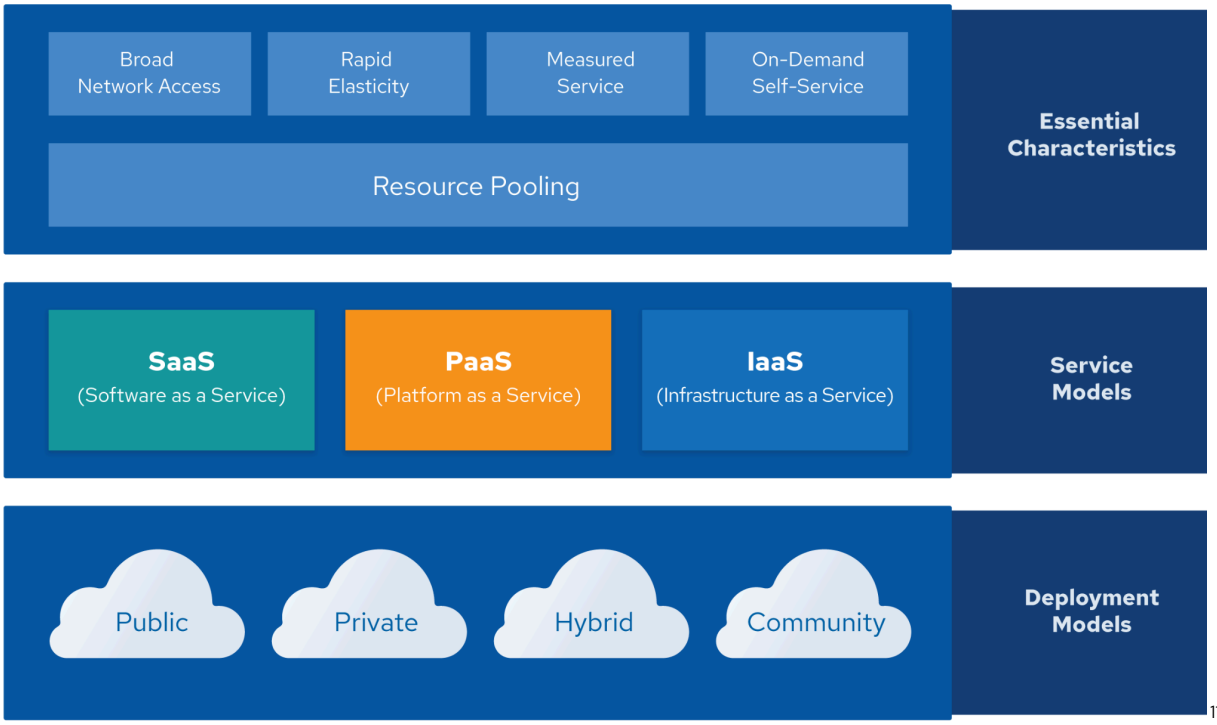
Clouds are *multi-tenant*[9] by nature. Multiple CSCs share the same pool of resources but are logically and at times physically *segregated* and *isolated* from each other. Segregation allows the CSP to divvy up resources to the different CSCs, and segregation ensures they cannot see or modify each other's assets, which is fundamental for the confidentiality and integrity of CSC data. In addition, a CSP's ability to measure and constrain the overuse of resources is critical for the democratic use and availability of the service delivered to each CSC. Multi-tenancy extends beyond inter-organizational use to facilitate resource distribution among various units within a single organization, often referred to as a "private cloud."

1.2 Cloud Computing Models

The CSA uses the NIST SP 800-145 model for cloud computing as it is the standard for defining cloud computing[10]. CSA also endorses the more in-depth ISO/IEC model 22123-1:2023 and 22123-2:2023, which also serves as a reference model. Throughout this domain, we will reference both.

NIST describes cloud computing based on five essential characteristics, three cloud service models, and four cloud deployment models, which are summarized in the following sections.

[9] In this reference to multi-tenant, a tenant is a cloud customer or CSC.
[10] CSA has chosen to align with the NIST definition of cloud computing (NIST 800-145) to drive consensus for a common language and focus on use cases rather than semantics. This material is intended to be broadly usable and applicable to organizations globally. While NIST is a U.S. government organization, the selection of this reference model should not be interpreted to suggest the exclusion of other points of view or geographies.



[11]

Figure 1: Overview of Cloud Computing Models Based on NIST and ISO/IEC Standards

1.2.1 Essential Characteristics

The NIST model describes the cloud by five essential characteristics, which sets cloud computing apart from traditional hosting services or other types of cloud services, such as hosting and virtualization. Understanding these characteristics is critical for leveraging the full potential of cloud computing and for the strategic planning of cloud adoption.

Following are the five essential characteristics described by NIST.

- **Resource Pooling:** Cloud computing pools various physical and virtual resources to serve multiple CSCs using a *multi-tenant* model. These resources, like storage, processors, memory, and network bandwidth, are dynamically assigned and reassigned according to demand.
- **Broad Network Access:** Services are available over the network and accessed through web browsers or specialized applications that promote use by heterogeneous thin or thick client platforms (e.g., servers, mobile phones, laptops, IoT devices, and tablets).
- **Rapid Elasticity:** Resources can be rapidly and elastically provisioned, in some cases automatically, to rapidly scale out and back. To the CSC, the provisioned capabilities often appear unlimited and can be purchased in any quantity at any time

[11] Depiction of the NIST Model of Cloud Computing

- **Measured Service**: Cloud systems automatically control and optimize resource use by leveraging a metering capability at some level of abstraction appropriate to the type of service (e.g., storage, bandwidth, active user accounts). Resource usage can be measured, monitored, controlled, and reported, providing transparency for both the CSP and the CSCs of the utilized service. This enables billing based on usage, which promotes cost efficiency and accountability (e.g. pay-as-you-go model).
- **On-Demand Self-Service**: A CSC can unilaterally request cloud resources on demand for automatic provisioning by the CSP and computing capabilities, such as computing time and network storage, as needed without requiring human interaction with each CSP.

ISO/IEC 22123: 2023 lists six key characteristics, the first five being identical to the NIST characteristics listed above. The only addition is multi-tenancy, which is distinct from resource pooling.

1.2.2 Cloud Service Models

NIST defines three service models describing the different foundational categories of cloud services:

- **Software as a Service (SaaS)** is an application that is managed and hosted by the CSP. CSCs access it using a web browser, mobile application, application programming interfaces (APIs), or lightweight client application. In this model, the CSC only worries about the application's configuration, not the underlying resources.
- **Platform as a Service (PaaS)** abstracts and provides platforms, such as application platforms (e.g., a place to develop and run code), databases, file storage, and collaborative environments. Other examples include application processing environments for machine learning, big data processing or API access to SaaS functions. The key differentiator is that, with PaaS, the CSC does not manage the underlying infrastructure.
- **Infrastructure as a Service (IaaS)** offers access to a resource pool of fundamental computing infrastructure, such as network, or storage. In IaaS the CSCis responsible for managing the underlying virtual infrastructure, such as virtual machines, networking, storage, and running applications.

ISO/IEC 22123-3:2023 uses a more complex definition with a cloud capabilities type that maps closely to the SaaS, PaaS, and IaaS (also known as SPI) service model tiers (application, platform, and infrastructure capability types). It then expands into cloud service categories that are more granular, such as communications-as-a-service (CaaS), network-as-a-service (NaaS), data storage as a service (DSaaS), and data recovery as a service.

These categories are somewhat permeable; some cloud services span the SPI tiers, while others do not fall neatly into a single service model. Practically speaking, there is no reason to assign everything into these three categories, or even the more granular categories in the ISO/IEC model. This is a descriptive tool, not a rigid framework.

Both approaches are valid, but since the NIST model is more concise and broadly used, it is the definition predominantly used in CSA research.

1.2.3 Cloud Deployment Models

NIST and ISO/IEC use the same four cloud deployment models; these are how the technologies are deployed, consumed, and applied across the entire range of service models.

- **Public Cloud**: The cloud infrastructure is made available to the general public or a large industry group and is owned by a CSP.
- **Private Cloud**: The cloud infrastructure is operated solely for a single organization. It may be managed by the organization or a third party and may be located on-premises or off-premises.
- **Community Cloud:** The cloud infrastructure is shared by several organizations and supports a specific community with shared concerns (e.g., mission, security requirements, policy, compliance considerations). It may be managed by the CSC(s) or a third party, and may be located on-premises or off-premises.
- **Hybrid Cloud:** The cloud infrastructure is a composition of two or more clouds (i.e., private, community, or public) that remain unique entities, but are bound together by standardized or proprietary technology that enables data and application portability (e.g., cloud bursting[12] for load balancing between clouds).[13]

Other deployment models:

- **Multi-cloud:** In a multi-cloud environment, a CSC utilizes multiple cloud services, such as applications and systems, from different CSPs. This approach is commonly adopted to reduce dependency on a single cloud provider and build technical resilience into the architecture design.
- **Hybrid multi-cloud:** A combination of public cloud and private resources, typically a connection to a traditional datacenter.

1.3 Reference & Architecture Models

There is a wide range of evolving technologies and techniques used to build and operate cloud services, potentially rendering aspects of any single reference or architectural model obsolete. The objective of this section is to provide some fundamentals, and to provide a baseline for understanding complex and emerging models to help security professionals make informed decisions. We recommend ISO/IEC 22123 and NIST 500-292[14] as in-depth reference architectural models, which complement the NIST cloud

[12] A configuration setup where an application running in a private cloud or data center dynamically extends to a public cloud to access additional computing resources when the demand exceeds the capacity of the primary environment, ensuring consistent performance and availability.
[13] Hybrid is also commonly used to describe a non-cloud data center bridged directly to a CSP. The original NIST definition refers to a mixture of cloud solutions. Since then it has become more narrowly focused on mixtures of cloud models.
[14] NIST. (2011) NIST Cloud Computing Reference Architecture

computing definitions. Additionally, we suggest exploring the CSA Enterprise Architecture Model,[15] which aims to consolidate features from four discrete organizational architectures.

One way of looking at cloud computing is as a stack where SaaS is built on PaaS, which in turn is built on IaaS. This is not representative of all (or even most) real-world deployment models, but it serves as a useful reference baseline. The SPI stack is evolving, and we are seeing overlaps and less clear distinctions between the service models as service offerings mature. So first, let us get a feel for the standard architectures of each cloud service model (layer in the SPI stack), and then some examples to show how the lines are blurring. Finally, we conclude with the CSA Enterprise Architecture Model, which can assist anyone who is developing cross-platform capabilities and patterns or is interested in an integrated, multi-cloud approach.

1.3.1 Infrastructure as a Service

Physical facilities and infrastructure hardware form the foundation of IaaS. With cloud computing, we abstract and pool these resources, but at the most basic level, we always need physical hardware, networks, and storage to build on. These resources are pooled using abstraction and orchestration. Abstraction, often through virtualization, frees the resources from their physical constraints to enable pooling. Then a set of core connectivity and delivery tools (orchestration) ties these abstracted resources together, creates the pools, and provides the automation to assign and deliver them to CSCs.

Orchestration is generally facilitated using APIs. APIs are typically the underlying communications method for components within the cloud, some of which are exposed to the CSC to manage resources and configurations. Most cloud APIs these days use Representational State Transfer (REST), which runs over the HTTP, making it well-suited for Internet services.

In most cases, those APIs are both remotely accessible and wrapped into a web-based user interface. This combination is the cloud management or *control plane*, since CSCs use it to manage and configure cloud resources, such as launching virtual machine instances or configuring virtual software-defined networks. From a security perspective, it is the biggest difference from protecting on-premises infrastructure, since management interfaces are now facilitated over a network. If an attacker compromises a management plane, they have privileged access to the cloud infrastructure.

[15] CSA. (2021) CSA Enterprise Architecture Reference Guide.

Here is an extremely simplified architectural example of a compute IaaS platform.

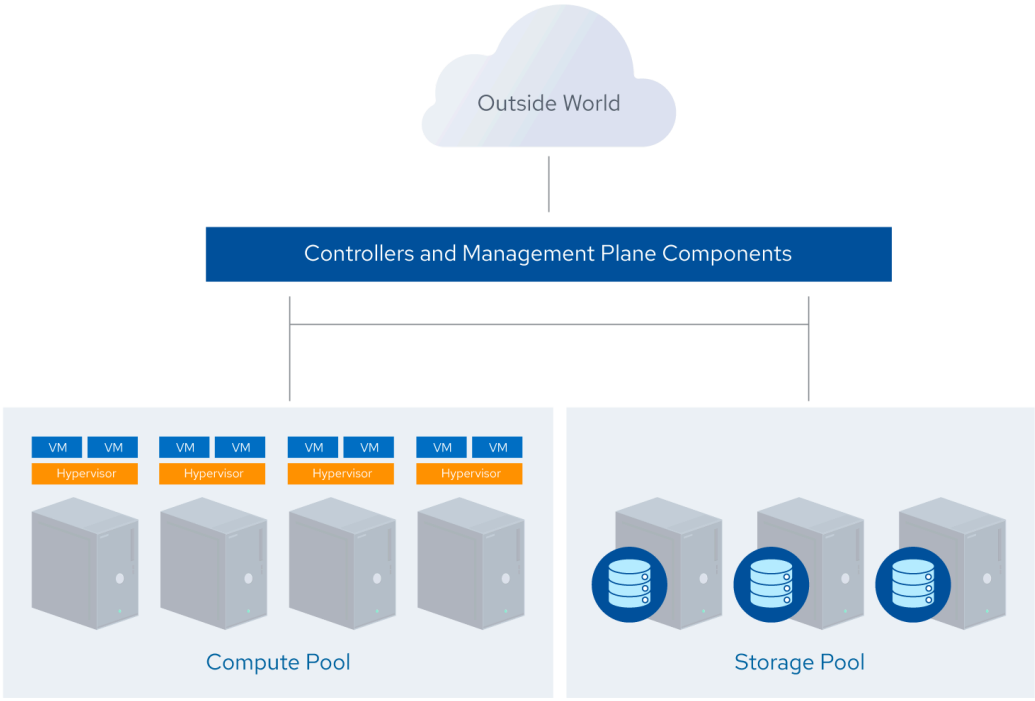


Figure 2: Simplified Architecture of an IaaS Compute Platform

This example showcases an IaaS compute platform with physical servers running hypervisors[16] and orchestration software. The cloud controller allocates resources, creates virtual instances, configures networking and storage, and brokers connectivity information for CSCs to access the instances.

1.3.2 Platform as a Service

Of all the service models, PaaS is the hardest to definitively characterize due to the wide range of PaaS offerings and varying approaches. PaaS services generally integrate application development frameworks, middleware capabilities, and supporting services such as databases, message queues, and event logging. These services allow developers to build applications on the platform with programming languages and tools supported by the stack.

One option, frequently seen in the real world and illustrated in our model, is to build a platform on top of IaaS. For example, integration, persistence, and middleware layers are built on an IaaS platform, then pooled together, orchestrated, and made accessible to CSCs through APIs as a PaaS service.

[16] A hypervisor, also known as a virtual machine monitor (VMM), is a software, firmware, or hardware platform that creates and runs virtual machines by abstracting and managing the underlying physical hardware resources, allowing multiple operating systems to run concurrently on a single physical host.

This could be a database as a service (DBaaS) built and deployed using modified database management system software instances. The CSC manages the database via API and/or a web console and accesses it through normal database network protocols and/or an API.

In PaaS, the cloud user only sees the platform (or the application presentation layer that leverages it), but not the underlying infrastructure. In our example, the database service expands or contracts as needed based on utilization without the CSC having to manage individual servers, networking, patches, and so on.

The following is a simplified architecture that shows a PaaS running on top of our IaaS architecture.

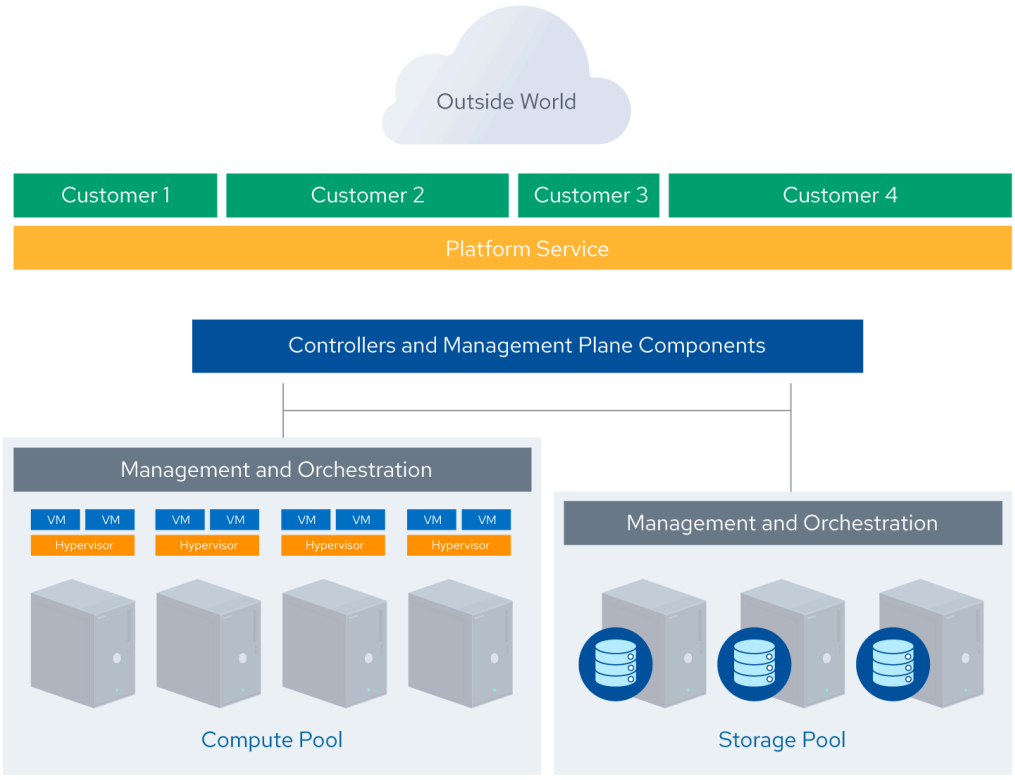


Figure 3: Simplified Architecture of a PaaS Built on IaaS

PaaS does not necessarily need to be built on top of IaaS; there is no reason it cannot be a custom-built, stand-alone architecture. The defining characteristic is that CSCs access and manage the platform, not the underlying cloud infrastructure. One potential example could be a custom AI and machine learning integration service supporting use cases like AI-powered development tools, machine learning operations (MLOps)[17], and AI lifecycle management.

[17] Machine Learning Operations (MLOps) is a set of practices that streamline the entire lifecycle of machine learning models. It unifies ML application development with ML system deployment and operations.

1.3.3 Software as a Service

SaaS services are complete applications, encompassing all the architectural complexities typical of any large software platform. Many SaaS CSPs build on top of IaaS and PaaS due to the increased agility, resilience, and economic benefits.

Most modern cloud SaaS applications combine IaaS and PaaS, sometimes across different CSPs. Many also offer public APIs for some or all functionality. They are often needed to support various CSCs, especially web browsers, APIs, and mobile applications.

SaaS services tend to have an application/logic layer and data storage, API and presentation layer services that commonly support web browsers and mobile application user interfaces, as well as Internet API access.

The simplified architecture below is taken from a real SaaS platform, but generalized to remove references to the specific products in use.

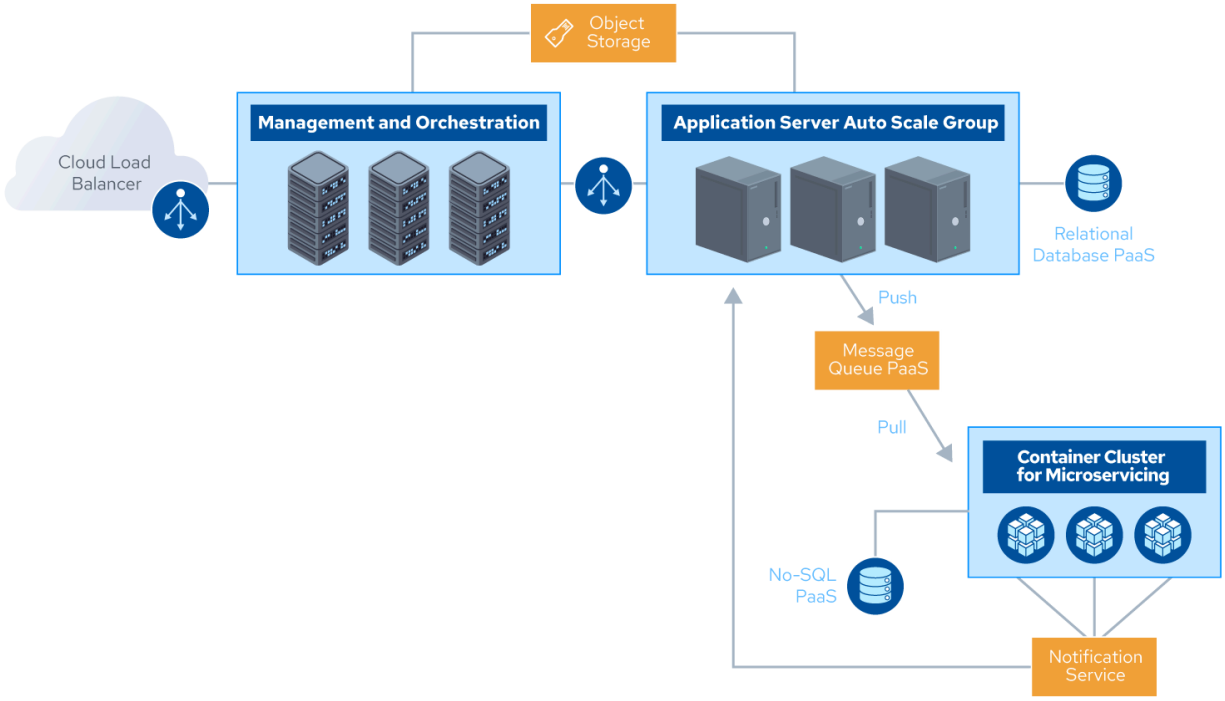


Figure 4: Simplified Architecture of a SaaS Platform Built on PaaS and IaaS

1.3.4 Anything as a Service

Anything as a Service, or XaaS, represents a broad, encompassing term that captures the essence of various services delivered over the Internet, rather than provided locally or on-premises. This model is foundational to cloud computing, where "X" can represent virtually any service, application, or platform component delivered to users as a service, for example, Security as a Service, DBaaS, Directory as a

Service, Identity as a Service, AI as a Service. These nearly always fit into the IaaS/PaaS/SaaS model but are more specific and descriptive in their naming.

1.3.5 Overlapping Service Models

While the SPI cloud service model is frequently represented hierarchically, with each layer built upon the one below it (IaaS, PaaS, SaaS), the ways these services are implemented and utilized in practice can be much more flexible. The SPI stack is a useful guide for understanding the different cloud service models. It is important to recognize the inherent flexibility and overlapping layers. The SPI implementations do not need to be built on a strict hierarchy, and the lines are often blurred between the models, known as overlapping service models. Overlapping service models are not defined by a strict hierarchy and often encapsulate characteristics of multiple service models simultaneously

For example, many services encapsulate characteristics of both SaaS (a fully delivered application in a web browser) and PaaS (APIs to integrate some of the platform's capabilities into customer architectures).

1.3.6 CSA Enterprise Architecture Model

The CSA Enterprise Architecture (EA) is both a methodology and a set of tools. It is a framework, that is, a comprehensive approach for the architecture of a secure cloud infrastructure. It can be used to assess opportunities for improvement, create roadmaps for technology adoption, identify reusable security patterns, and assess various CSPs and security technology vendors against a common set of capabilities.

To create the CSA EA, CSA Research has leveraged four industry standard architecture models across the following four domains:

- **Business Operation Support Services (BOSS)** – Sherwood Applied Business Security Architecture (SABSA)
- **IT Operation Services (ITOS)** – IT Infrastructure Library (ITIL)
- **Technology Solution Services (TSS)**, including Infrastructure (InfraSrv), Information (InfoSrv), application (AS), and Presentation (PS) Services – The Open Group Application Framework (TOGAF)
- **Security and Risk Management (SRM)** – OpenGroup Security Forum (formerly known as the Jericho Forum)

Figure 5: Building Blocks of the CSA Enterprise Architecture

CSA combines the best architecture paradigms into a comprehensive approach to cloud security while merging business drivers. The CSA EA supports the value proposition of cloud services within an enterprise business model.

The CSA EA was adopted by NIST SP 500-292, solidifying the importance of the CSA approach.

1.4 Cloud Security Scope, Responsibilities, & Models

Cloud security and compliance includes everything a security team is already responsible for, just in the cloud. The iterative process of identifying security requirements, selecting appropriate cloud services, and implementing controls to mitigate risks effectively in cloud computing is delineated by the principle of *shared responsibility*. We describe the model based on this principle. We outline the division of security responsibilities, with CSPs accountable for infrastructure security, and CSCs responsible for their deployed applications and data. This separation of responsibilities varies across service models (IaaS, PaaS, SaaS) and CSPs, underscoring the importance for CSCs to understand their divided obligations. Additionally, we explore the role of frameworks and tools, such as the CSA Consensus Assessments Initiative Questionnaire (CAIQ) and CSA Cloud Controls Matrix (CCM), in facilitating compliance and alignment with security standards. All the traditional security domains remain, but the nature of risks, roles and responsibilities, and implementation of controls, change often and rapidly.

Though the overall scope of security and compliance does not change, the portions any given cloud actor is responsible for certainly do. Think of it this way: Cloud computing is a shared technology model where different organizations are responsible for implementing and managing different parts of the stack. As a result, security responsibilities are also distributed across the stack and, thus, across the organizations involved.

This is commonly referred to as the Shared Security Responsibility Model (SSRM), sometimes shortened to SRM. Think of it as a responsibility matrix that depends on the particular cloud provider and feature/product, service, and deployment model.

1.4.1 Shared Security Responsibility Model

In cloud computing, security is a joint effort between CSPs and CSCs. The term “shared responsibility” is widely used by multiple CSPs and refers to the delineation where the CSP takes responsibility for security operations and controls. Below the line is the CSP’s responsibility; anything a CSC builds above the line is the CSC’s responsibility. This varies greatly as you change service models. This SSRM model outlines that CSPs are responsible for the security "of the cloud," including infrastructure, hardware, and network. CSCs, however, are responsible for what they deploy in the cloud.

The delineation of responsibilities will differ for IaaS, PaaS, and SaaS and often between different CSPs. It is critical for CSCs to understand where the demarcation lies to ensure that they are appropriately protecting their own cloud tenants, applications, data, etc., and to provide baselines for holding CSPs accountable.

At a high level, security responsibility maps to the degree of control any given actor has over the architecture stack.

- **Software as a Service:** The CSP is responsible for most of the security since the cloud user can only access and manage their use of the application, and cannot alter how the application works. Even if the CSC responsibilities are narrower and more limited in each security domain, they seldom drop to zero. For example, a SaaS CSP is responsible for perimeter security, logging/monitoring/auditing, and application security, whereas the CSC is still responsible for the management of authorization and entitlements.

- **Platform as a Service:** The CSP is responsible for the platform's security, while the CSC is responsible for everything they implement on the platform, including how they configure any offered security features. The responsibilities are thus more evenly split. For example, when using a DBaaS, the CSP manages fundamental security, patching, and core configuration at a given service level. The CSC is responsible for everything else, including which security features of the database to use, managing accounts, or even authentication methods.

- **Infrastructure as a Service:** Just like PaaS, the CSP is responsible for foundational security, while the CSC is responsible for everything they build on the infrastructure. Unlike PaaS, this places far more responsibility on the CSC. For example, the IaaS CSP will likely monitor their perimeter for attacks, but the CSC is fully responsible for how they define and implement their virtual network security based on the tools available on the service.

As we move down the SPI stack, the CSP's responsibilities decrease, and the CSC's responsibilities increase. IaaS stops lower in the stack. Thus customers are responsible for securing the operating systems and applications. PaaS is in the middle and may offer some level of security within the platform, but the CSC would still be required to make the API calls within the application and maintain a secure configuration. SaaS is a bit different because the CSP is responsible for the entire stack. Thus, the burden is on them to protect any information within their service. As you can imagine, data security is very important to SaaS CSPs since a breach or failure could result in the proverbial “run on the bank” and put the entire business in danger.

These roles are further complicated when using cloud brokers or other intermediaries and partners. Understanding where the CSP's responsibility ends and where the CSC's begins is crucial. It is not just about leveraging the cloud but doing so securely by recognizing the CSC’s role in the partnership. CSCs must regularly review and understand their obligations, especially in configuration and management, to ensure that security policies/measures align with the sensitivity of the data and resources in use in their organization.

The following diagram illustrates SSRM, highlighting the division of responsibilities between CSPs and CSCs across different service models. This model underscores the varying degrees of control and responsibility each actor has over the architecture stack:

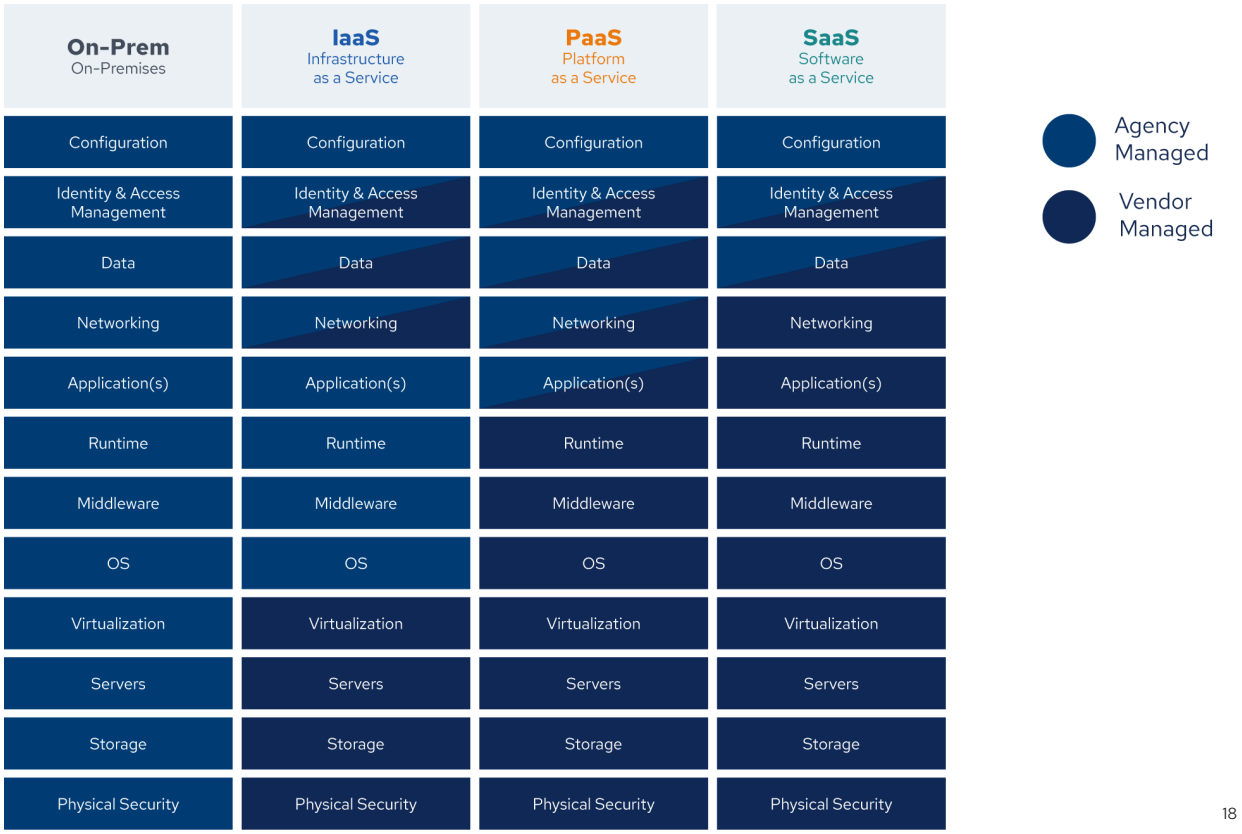


Figure 6: Shared Security Responsibility Model

The key to effective cloud security is understanding the division of responsibilities in any cloud project. Knowing precisely who is responsible for what is crucial, regardless of specific security controls offered by CSPs. This understanding allows organizations to fill control gaps with their measures or consider alternative CSPs. A user's ability to directly control security is very high for IaaS, and less so for SaaS.

To ensure clear allocation of security responsibilities in the cloud, we recommend the following:

- **CSPs** should thoroughly document internal security controls and CSC features so that they can make informed decisions. CSPs should also properly design and implement those controls.
- **CSCs** should build a roles-and-responsibilities matrix for their security responsibilities. This matrix should document who is responsible for implementing specific security controls and ensure alignment with relevant compliance standards.

CSA provides tools to help meet these requirements:

- The CAIQ is a standard template for CSPs to document their security and compliance controls.
- The CCM lists cloud security controls and maps them to multiple security and compliance standards. The CCM can also be used to document security responsibilities.

[18] CISA. (2021) Cloud Security Technical Reference Architecture

Both documents will need tuning for specific organizational and project requirements but provide a comprehensive starting template and can be especially useful for ensuring compliance requirements are met.

Additional resources and guidance on the SSRM include:

- CSA Enterprise Architecture - This framework provides a comprehensive approach towards the architecture of a secure cloud infrastructure. Please refer to the section above on CSA EA.
- Enterprise Architecture to CCM Shared Responsibility Model - The mapping helps users understand cloud security responsibilities. It shows which security controls (per CCM) are the responsibility of the CSP or the CSC for different service models (IaaS, PaaS, SaaS).
- CCM Implementation Guidelines - Control ownership and implementation guidelines for CSPs and CSCs as it pertains to the CCM.

1.4.2 Cloud Security Frameworks & Patterns

Cloud security frameworks and patterns are tools to help guide security decisions. The term "model" can be somewhat unclear, so we break out the following types:

- **Conceptual models** or *frameworks* include visualizations and descriptions used to explain cloud security concepts and principles, such as the NIST model in this document.
- **Control models** or frameworks categorize and detail specific cloud security controls or categories of controls, such as the CSA CCM.
- **Reference architectures** are templates for implementing cloud security, typically generalized (e.g., an IaaS security reference architecture). They can be very abstract, bordering on conceptual, or they can be quite detailed, down to specific controls and functions.
- **Design patterns** are reusable solutions to particular problems. In security, an example is IaaS log management. As with reference architectures, they can be more or less abstract or specific, even down to common implementation patterns on particular cloud platforms.

The lines between these models often blur and overlap, depending on the goals of the model developers. Even grouping these together under the heading "model" is probably inaccurate, but since we see the terms used so interchangeably across different sources, it makes sense to group them.

CSA recommends the following models:

- The CSA EA[19]
- The CSA CCM[20]
- ISO/IEC CD 27017.2[21]

[19] CSA. (2024) Enterprise Architecture Working Group - Enterprise Architecture
[20] CSA. (2024) Cloud Controls Matrix
[21] ISO. (2024) Information Technology – Security Techniques – Code of Practice for Information Security Controls based on ISO/IEC 27002 for Cloud Services. This draft is under development and is meant to replace ISO/IEC 27017:2015.

1.4.2.1 A Simple Cloud Security Process Model

While the implementation details, necessary controls, specific processes, and various reference architectures and design models vary greatly depending on the specific cloud implementation, there is a relatively straightforward, high-level process for managing cloud security.

- Identify necessary security and compliance requirements and any existing controls
- Select the CSP, service, and deployment models
- Define the architecture
- Assess the security controls
- Identify control gaps
- Design and implement controls to fill the gaps
- Assess the effectiveness of the controls
- Manage changes over time

Each cloud project, even within the same CSP, may require unique configurations and technologies. Therefore, it is important to evaluate each project on its specific requirements and characteristics. For example, the security controls for an application deployed on pure IaaS in one CSP may look very different than a similar project that uses more PaaS from the same provider.

The key is to identify requirements, design the architecture, and then identify the gaps based on the capabilities of the underlying cloud platform. That is why it is essential to understand the CSP and architecture before implementing controls to meet the security requirements.

This is typically an iterative process. Understanding when to use native cloud service controls and when to implement the controls externally to close the gaps, is an important consideration that can have a significant impact on the overall security architecture.

Summary & Areas of Critical Focus - Governance & Operations

The 11 other domains which comprise the remainder of the CSA Guidance highlight areas of concern for cloud computing and are tuned to address both the strategic and tactical security "pain points" within a cloud environment, and can be applied to any combination of cloud service and deployment model.

The domains are divided into two broad categories: governance and operations. The governance domains are broad and address strategic and policy issues within a cloud computing environment, while the operational domains focus on more tactical security concerns and implementation within the architecture.

Title	Description
Cloud Computing Concepts & Architectures	Learn cloud computing concepts and architectures for cloud security professionals. Topics include cloud models, security frameworks, cloud-native capabilities, abstraction, orchestration, and multi-tenancy, emphasizing agility, resiliency, and security.
Cloud Governance	Learn cloud governance with a focus on security, covering strategies like DevOps, DevSecOps, Zero Trust, and AI/ML. Understand frameworks, risk management, compliance, and establish effective governance structures like CCoE and cloud registries.
Risk, Audit & Compliance	Covers risk management, audit processes, and compliance in cloud environments. Learn cloud risk evaluation, compliance requirements, laws, and audit processes. Topics include compliance inheritance and leveraging CSP compliance for regulatory standards.
Organization Management	Covers managing and securing cloud environments with major CSPs, focusing on organizational hierarchy, IAM, hybrid/multi-cloud security, and Zero Trust strategies. Learn to implement cohesive security controls and manage diverse cloud infrastructures.
Identity and Access Management	Covers Identity and Access Management (IAM) in cloud environments, focusing on federation, strong authentication, and authorization. Learners will explore advanced IAM models, Zero Trust strategies, and automation to enhance cloud security and compliance.
Security Monitoring	Covers cloud security monitoring challenges and solutions, including cloud telemetry, logs, hybrid/multi-cloud setups, and advanced monitoring tools. Topics include the SSRM, log storage, canaries, honey tokens, and the role of GenAI in enhancing cloud security.
Infrastructure and Networking	Covers managing and securing cloud infrastructure, including secure architecture design, SDNs, IaC, and secure cloud connectivity. It emphasizes Zero Trust, SASE, container security, and integrated security measures to protect cloud assets.
Cloud Workload Security	Covers securing cloud workloads, including VMs, containers, serverless functions, PaaS, and AI. Learn to secure VM images, manage container vulnerabilities, and implement encryption, access controls, runtime protection, and IAM best practices.
Data Security	Covers data security in cloud environments, focusing on data classification, encryption, access controls, and various cloud storage types. It addresses securing data at rest, in transit, and in use, along with AI system security and future data security technologies.
Application Security	Learn cloud application security, focusing on protecting apps from external threats. Learn about SDLC, threat modeling, secure coding, and testing. Topics include IaC, DevOps, third-party libraries, and emerging

	cloud security technologies.
Incident Response & Resilience	Covers incident response and resilience in cloud environments, crucial for organizational security. Learn CIR strategies, tools, and practices based on CSA and NIST guidelines. Topics include preparation, detection, containment, recovery, and resilience strategies.
Related Technology & Strategies	Covers cloud security strategies, focusing on Zero Trust, AI integration, and Threat and Vulnerability Management. Learn to secure cloud apps, systems, and data through multi-factor authentication, encryption, AI threat detection, and continuous monitoring.

Table 1: List of Security Guidance Domains

Recommendations

- Understand the differences between cloud computing and how abstraction and orchestration impact security.
- Become familiar with the NIST model for cloud computing and the CSA reference architecture.
- Use the SSRM tool to apportion and place responsibility and accountability/obligations for security between the CSC and CSP.
- Use tools and documents like the CSA CAIQ to evaluate and compare cloud providers.
- Cloud providers should document their security controls and features and publish them using tools like the CSA CAIQ.
- Use tools like the CSA CCM to assess and document cloud project security and compliance requirements and controls, as well as who is responsible for each.
- Use a cloud security process model to select providers, design architectures, identify control gaps, and implement security and compliance controls.

Additional Guidance

- CCSK PrepKit | CSA
- Cloud Security Alliance Glossary | CSA
- CSA Cloud Controls Matrix (CCM) | CSA
- CCM-Lite and CAIQ-Lite | CSA
- CCM v4 Implementation Guidelines | CSA
- CSA Enterprise Architecture Reference Guide | CSA
- Enterprise Architecture to CCM Shared Responsibility Model | CSA

Domain 2: Cloud Governance and Strategies

This domain focuses on cloud governance with an emphasis on the role of security. Governance is based upon a framework of policies, procedures, and controls that are designed to promote transparency and accountability to defined standards. Strong governance practices address strategic guidance, risk management and mitigation, compliance monitoring and remediation, budget allocation and cost controls. IT governance ensures that information and related technology support and enable the enterprise strategy and the achievement of enterprise objectives.

ISACA[22] defines governance as: *"The method by which an enterprise evaluates stakeholder needs, conditions and options to determine balanced, agreed-upon enterprise objectives to be achieved. It involves setting direction through prioritization, decision making and monitoring performance and compliance against the agreed-upon direction and objectives."*

Organizations can follow various and industry-specific governance standards and frameworks[23] to enhance their governance practices. For example, the ISO/IEC 38500:2024 standard provides guidance on the governance of IT for organizations. ISACA COBIT framework offers a comprehensive guide for the governance and management of enterprise IT.

For more information on governance standards:

- ISO/IEC 38500:2024 - Information Technology - Governance of IT for the Organization
- ISACA - COBIT - A Business Framework for the Governance and Management of Enterprise IT
- ISO/IEC 27014:2020 - Information Technology - Security Techniques - Governance of Information Security
- The Open Group Cloud Computing Governance Framework

Here are some examples of laws and regulations that affect IT governance requirements:

- Gramm-Leach-Bliley Act (GLBA)
- Sarbanes-Oxley Act (SOX)
- The Health Insurance Portability and Accountability Act (HIPAA)
- General Data Protection Regulation (GDPR)

[22] ISACA. (2024) Glossary - Governance
[23] Frameworks are discussed in detail in *2.3 The Governance Hierarchy*

Learning Objectives

In this domain, you will learn to:

- Identify the purpose of cloud governance.
- Define the governance hierarchy in cloud governance.
- Explore key strategies and concepts affecting governance in cloud computing.

2.1 Cloud Governance

Cloud computing's multi-tenancy nature, shared responsibility, redistribution of sensitive data, hosting of critical applications and infrastructure, security, and privacy concerns require effective governance to manage.

Compliance is another major concern when storing data in the cloud, with various regulations, jurisdictions, security, and privacy requirements. Without a sound governance approach, security, financial, and operational risks associated with cloud operations can exponentially grow and make the cloud an insensible option for businesses.

One of the primary drivers for cloud adoption is cost efficiency and speed to market. Many organizations perceive cloud computing as a way to achieve cost savings by shifting away from a capital expenditure data center model to an operational expenditure model (e.g., pay-as-you-go and subscription-based). As a result, a common migration strategy is to "lift and shift" existing applications and infrastructure to the cloud. Governance of security and privacy is critical in migrations like this since the shift to a new technology platform can introduce new technical risks, even (and especially) if the architecture stays the same. Furthermore, the application now has shared responsibilities with a cloud service provider (CSP) in the picture.

Strategic innovation is another significant driver of cloud adoption. Many organizations view software as a strategic asset that can provide a competitive advantage. Cloud offers the ability to rapidly develop and deploy software, enabling organizations to quickly bring new products and services to market. However, from a governance perspective, managing rapid change from accelerated deployments introduces risks such as misconfigurations and software supply chain hazards. It is crucial to have robust secure-by-design processes to ensure that software development, testing, and deployment in the cloud are secure and reliable.

Key takeaways from this discussion include:

- Cloud adoption is driven by factors including cost savings, operational expenses models, the objectives of the organization, and desire to achieve strategic innovation.
- Governance considerations in cloud adoption include managing information risk (data, applications, host operating systems, network, and supply chain), and physical risk, to acceptable levels (known as risk appetite). This is done by evaluating whether the IT objectives are aligned

with business objectives, and by ensuring compliance with legislative and regulatory requirements, including privacy obligations.

- Organizations should tailor their migration strategies to align with their specific business objectives, ensuring that they choose the right cloud services and implement appropriate governance measures.

2.1.1 Cloud Adoption & Governance

There are two primary ways cloud affects security governance:

1. The introduction of the Shared Responsibility Mode (SRM). Security governance responsibilities are now shared between the CSP and the CSC. To further complicate matters, in some cases, third-party service providers are introduced into the supply chain, each bearing their own security risks. Even if some of the responsibilities are offloaded to such a third party, the accountability remains with either the CSP or CSC.

2. The technical and operational differences created by the inherent nature of cloud computing.

Before the cloud, IT security governance relied extensively on the inherent nature of operating primarily within data centers. Data centers have limited resources – there is only so much space, computers, networking, etc. – that are in relatively isolated physical environments. The organization structures, policies, and controls are all aligned around the resource scarcity of these facilities.

Public cloud is the complete opposite. While cloud providers do not have infinite capacity, it is in their interest to always have enough capacity for their customers' requirements. Cloud is decentralized and different teams can provision entire stacks of resources with a login and credit card, none of which will be under any kind of central management without effective governance controls.

Cloud also fundamentally changes how we manage those resources. While resources can be decentralized, in a public cloud, the core management and administrative interfaces are unified and open to the Internet. There is no physical network perimeter and anyone with the right credentials can access the management plane and restructure the entire virtual infrastructure. This combination of decentralized usage, with a unified management plane that is open to the Internet, requires new, cloud-specific governance approaches.

In conclusion, the cloud revolutionizes IT governance by decentralizing infrastructure management, providing unified administrative access, and enabling access to resources. Organizations must embrace these changes while ensuring security and control to fully leverage the benefits of cloud computing.

2.1.2 Cloud Governance Complexities

As organizations increasingly adopt cloud services, they face unique governance challenges due to new business models, technologies, and management approaches. These challenges require organizations to update their governance frameworks to fit the cloud environment. Common considerations across Paas,

SaaS, and IaaS include control and accountability, legal and regulatory compliance, and the relationship between CSPs and Cloud Service Consumers (CSCs).

This section lists these considerations, highlighting the challenges as well as governance adjustments needed to manage cloud environments effectively.

The following considerations outline the key governance challenges and necessary adjustments organizations must address when managing cloud environments:

Control and Accountability

- Cloud might impose a loss of direct control over the IT infrastructure, forcing an organization to adopt a new governance framework and processes.
- Using cloud solutions does not necessarily mean outsourcing the accountability of controls to a third or fourth party.
- Cloud uses a number of different SRMs, which vary depending on the CSP and the technology stack. In turn, this requires a clear allocation of controls and responsibilities between the cloud service provider and the customer.
- CSCs have to rely more on assessment activities rather than actual testing.

Legal and Regulatory Compliance

- Cloud services and data may span multiple jurisdictions, forcing customers to comply with more laws and regulations, especially for privacy.
- Data ownership rights and classification, as well as privacy control, might not be intuitively clear and may need careful examination.

Visibility and Transparency

- Visibility and transparency into some cloud services can be challenging.

Customization and Standardization

- CSPs may have a standard offering that cannot be customized according to a CSC's specific requirements.
- CSPs might demonstrate different levels of maturity, and variety of services, licenses, and models, which complicates adoption of a one-size-fits-all cloud policy.

Governance Complexity

- Cloud services are often built on a chain of CSPs, which makes scoping of governance activities challenging (e.g., a SaaS provider that is running on the infrastructure of another IaaS provider).

- Hybrid cloud models can complicate governance due to the complexities of producing clear boundaries between CSP and CSC responsibilities.

CSP and CSC Dynamics

- CSPs may change rapidly, which has to be accounted for in governance models.
- Utilization of the cloud services may require additional skills that may not currently be present in the CSC, such as cloud auditing skills or cloud security skills, as well as knowledge of cloud-oriented security tools.

The following figure outlines the specific complexities associated with different cloud service models, highlighting unique governance challenges and responsibilities for each model.

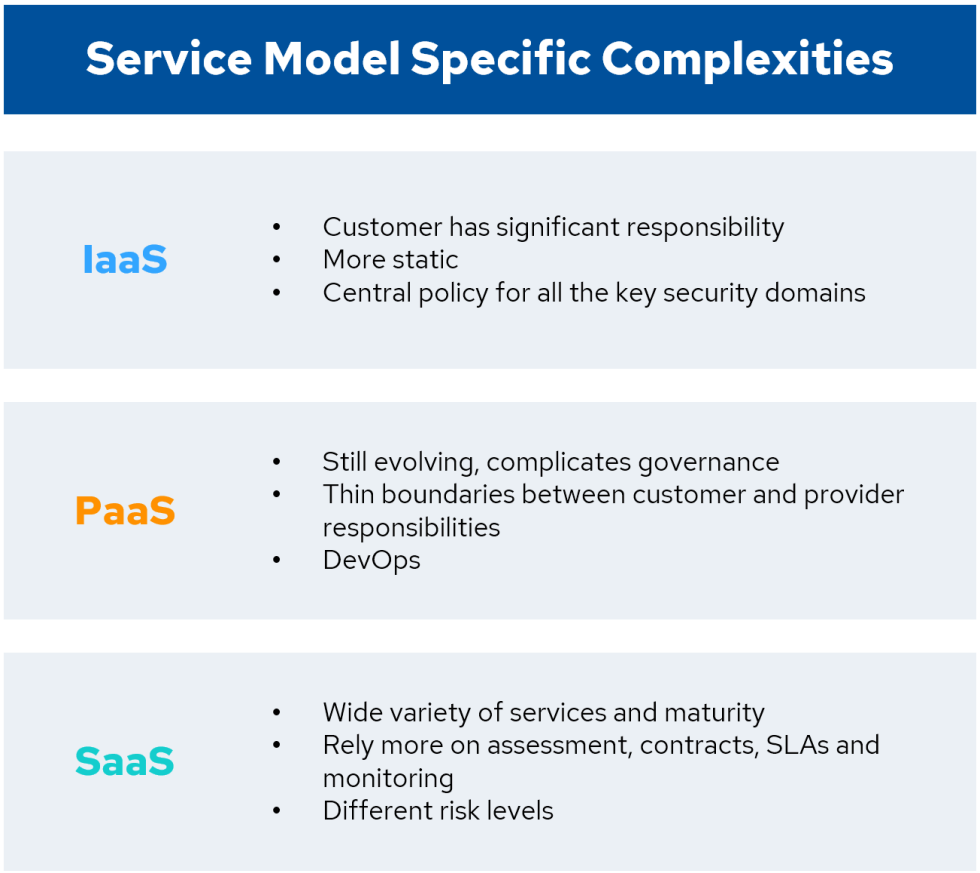


Figure 7: Service Model Specific Complexities

Effective cloud governance requires a flexible and robust strategy to address the unique challenges of IaaS, PaaS, and SaaS models. By understanding and managing these complexities, organizations can ensure security, compliance, and operational efficiency in their cloud environments.

2.1.2.1 Cloud Governance Complexities: Deployment Models

Moving forward, it's also essential to consider the governance complexities associated with different cloud deployment models. The following section explores these deployment models, highlighting unique governance challenges and responsibilities for each.

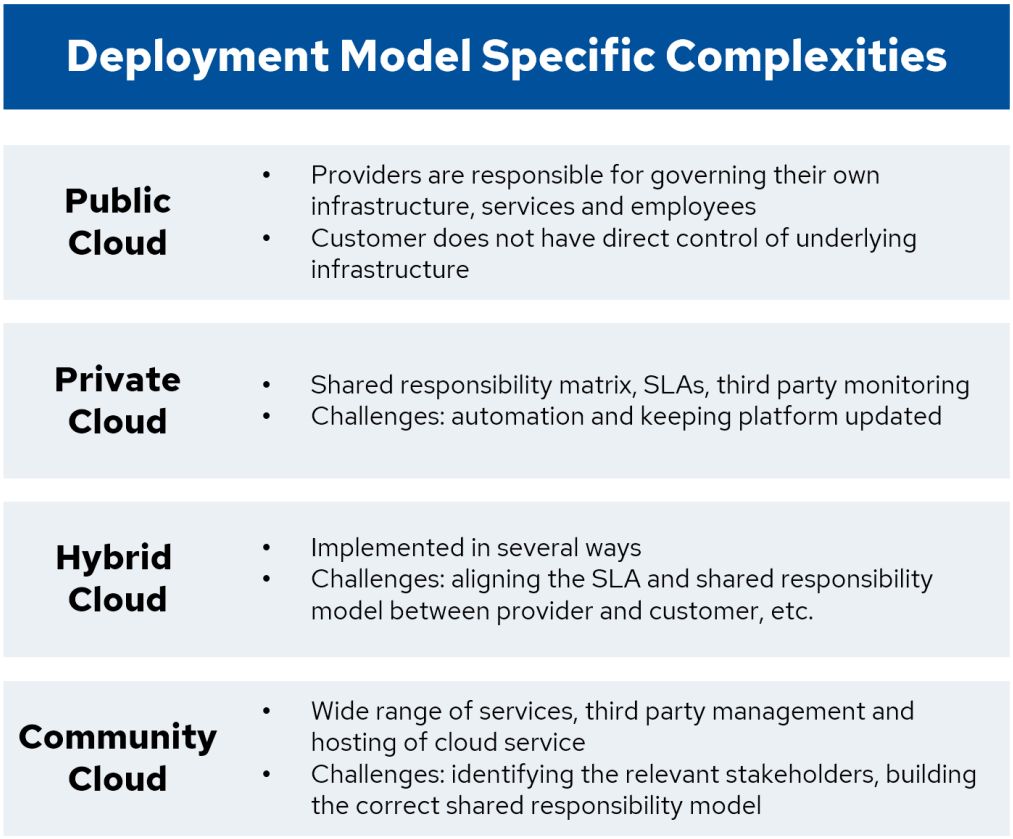


Figure 8: Deployment Model Specific Complexities

Public cloud: Public cloud is the most popular cloud deployment model, offering standard services to all customers. Providers typically resist customization requests, complicating governance as they manage their own infrastructure, services, and employees. Governance challenges arise from customer configurations, both initially and over time.

Public cloud relies on multi-tenancy, which brings governance challenges like segmentation and isolation. This often limits actions such as security scanning or penetration testing and reduces visibility into the infrastructure. These challenges require new governance approaches, using vendor risk management, service level agreements (SLAs), third-party audits, and compliance reports.The effectiveness of this new approach will be measured based on the cloud consumer's ability to mitigate the unique risks that cloud computing introduces.

Private cloud: Private clouds can be owned, managed, or hosted by the organization or a third party. Governance of self-managed private clouds is similar to traditional IT governance but must also address cloud-specific issues like attack vectors, multi-tenancy, and automation. Governance of private clouds

that are managed by third parties is the closest to traditional outsourcing models we already know. Governance challenges include understanding the shared responsibility matrix, setting SLAs, and building third-party monitoring capabilities to track policy breaches and insider threats. A major challenge is keeping the platform updated with the latest services, requiring special attention.

Hybrid cloud: Hybrid cloud services combine private and public cloud models. They can be implemented in various ways, complicating policy guidelines and SRMs. Governance challenges include aligning SLAs and responsibilities between provider and customer, protecting internal perimeters, scaling security configurations, and addressing skill gaps in cloud security and maturity.

Community cloud: Community cloud refers to a range of services managed and hosted by third parties, shared by multiple organizations but not fully public, reducing multi-tenancy challenges. Governance challenges include identifying stakeholders, building the correct SRM, and focusing on relationships and risks among organizations using the same community cloud.

2.2 Effective Cloud Governance

Effective cloud governance involves establishing a robust framework and set of policies to ensure effective, secure and compliant usage of cloud resources. It requires the implementation of a strong control framework and policies for secure, compliant, and efficient management of cloud resources. This includes:

- Defining roles and responsibilities
- Establishing a Cloud Center of Excellence (CCoE) or similar
- Conducting requirements gathering
- Risk-based planning
- Management of risks and remediations
- Classifying data and digital assets
- Complying with legal and regulatory requirements
- Maintaining a cloud registry[24]
- Establishing a governance hierarchy[25]
- Leveraging cloud-specific security frameworks
- Defining cloud security policies
- Setting control objectives and specifying control specifications

By implementing these components, organizations can maximize the benefits of cloud computing while mitigating potential risks. Below we explore some key components of effective cloud governance.

[24] Additional details provided later in this section - *Domain 2: Cloud Governance*.
[25] Additional details provided later in this section - *Domain 2: Cloud Governance*.

2.2.1 Cloud Governance Implementation Models

Cloud Center of Excellence (CCoE) and Cloud Advisory Council (CAC) models are widely adopted approaches for implementing effective cloud governance. The CCoE includes working members and evangelists. The CAC provides executive sponsorship and endorsement. More specifically:

- A CCoE is a centralized team or department that provides guidance, best practices, and support to the rest of the organization regarding cloud adoption and usage. The CCoE helps ensure consistency, standardization, and alignment with the CSC's goals and objectives.
- The CAC can include a group of senior leaders from IT, risk management, compliance, security, and general business functions that are responsible for setting the vision and direction of the CSC's cloud strategy and operating plan. The CAC is not covered in depth here but is important to be aware of.

CCoE and CAC are recommended components of IT governance and security within the CSC. They serve as a centralized hub, responsible for leading cloud initiatives and driving strategic, secure, compliant, and effective cloud adoption. Not all CSCs use the same terminology, but from a functional standpoint, these structures highlight the key elements required for effective cloud governance.

2.2.1.1 The Cloud Center of Excellence

One of the key functions of the CCoE is to provide strategic guidance. It ensures that cloud initiatives are aligned with the CSC's overall business objectives. By doing so, the CCoE ensures that cloud adoption supports the CSC's direction and contributes to its success.

The CCoE also develops and enforces a governance framework for cloud usage. This includes creating policies and standards that comply with external regulations and internal best practices. The CCoE is responsible for managing risks, ensuring data privacy and security, and maintaining compliance in the cloud environment.

The CCoE disseminates knowledge regarding cloud technologies and security measures. It provides training opportunities and resources to other departments, promoting a consistent level of cloud proficiency across the organization. This ensures that employees have the necessary skills and knowledge to leverage cloud services effectively and securely.

Amongst those responsibilities, security is a primary focus of the CCoE. It takes the lead in embedding security into the cloud infrastructure, ensuring that cloud deployments are secure by design. The CCoE ensures that the CSC's security and privacy requirements are met and responds to the evolving threat landscape.

The CCoE fosters cross-functional collaboration which involves various departments, such as (but not limited to) IT, security, compliance, and finance. This collaborative approach ensures that cloud decisions are made holistically, considering cost, security, compliance, and business needs.

The CCoE also fosters innovation and adaptability. It encourages the exploration of new cloud services and technologies and promotes a culture of innovation within the organization. At the same time, the CCoE remains adaptable to changes in technology and business environments, ensuring that the CSC can effectively leverage the latest advancements in cloud technology.

Ultimately a CCoE is useful for CSCs seeking to leverage cloud technologies effectively and securely. They go beyond technology and focus on aligning cloud adoption with business strategy, while ensuring governance, security, and compliance in the cloud environment. The following figure illustrates the key roles within a CCoE.

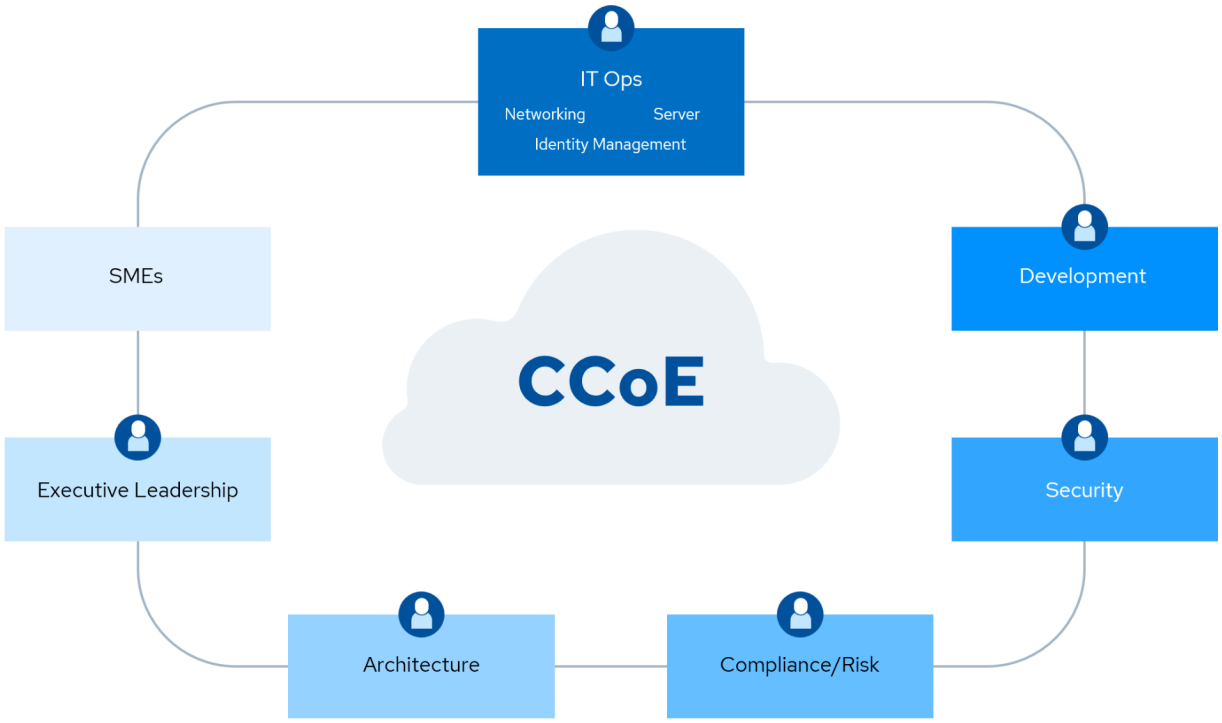


Figure 9: Key Roles in CCoE

2.2.2 Security Champions

In addition to having a CCoE, CSCs can also appoint Security Champions within business units including compliance, enterprise risk, legal, human resource, finance, and IT. These are individuals who understand the importance of cloud security and therefore can act as advocates for security, helping drive the implementation of security best practices and controls.

Security Champions should be appointed from within a team and have a hands-on role. They are typically not members of the security organization, but rather members of their own teams. This distinction allows them to focus on practical security implementation within their specific team dynamics. It is essential to differentiate the role of Security Champions from that of the Business Information Security Officer (BISO), Chief Information Security Officer (CISO) or Information Security Officer (ISO).

For example, in the DevOps team, the ideal candidate for the role of Security Champion is typically a developer, system administrator, or a DevOps or Platform Engineer who is already part of the team. This familiarity with the team dynamics and technical skills allows them to effectively advocate for security from the inside. They have the knowledge and expertise to understand the specific security challenges and solutions in cloud services and DevOps practices.

The role of Security Champions is crucial in the integration of security practices within the DevSecOps process. Acting as liaisons between security teams and development teams, they play a pivotal role in decentralizing responsibilities. By advocating for security within their teams, Security Champions promote a security culture within cloud and DevOps/Site Reliability Engineering (SRE) teams. Security Champions are more common in development-related teams but can also play an important role in other business units.

To cultivate the skills and interests of Security Champions, it is important to provide them with engaging and interactive security training. Workshops on ethical hacking, for example, can be an effective way to enhance their practical security skills. It is important, however, not to push Security Champions to become full-time security experts. They are developers and administrators who receive additional security training to serve as liaisons and experts within their teams.

The goal of empowering Security Champions is to enable them to play an advisory role rather than burdening them with additional duties better served by dedicated security teams. It is important to avoid burnout. By empowering Security Champions, CSCs can effectively bridge the gap between security and development, fostering a culture of security within cloud and DevOps teams.

In summary, Security Champions are essential in promoting a security culture within cloud, DevOps teams and business teams. By empowering them with the right mix of experience, training, and authority, CSCs can effectively integrate security practices within the development process. This ultimately leads to improved security outcomes.

2.3 The Governance Hierarchy

A key aspect of cloud governance is the establishment of a **governance hierarchy**. This involves defining decision-making processes and escalation paths for cloud-related issues. The governance hierarchy ensures that decisions are made at the appropriate level within the CSC and that there is a clear line of accountability and responsibility. CSCs can leverage cloud-specific security frameworks to guide their cloud governance efforts.

The governance hierarchy in information security is a structured approach to ensuring the security of an organization's systems and data. At the top of this hierarchy is the **Framework**, which provides a set of guidelines for cybersecurity practices. Examples of frameworks include the NIST Cybersecurity Framework (CSF), Cloud Controls Matrix (CCM), Center for Internet Security (CIS), and the IANS Cloud Security Maturity Model (CSMM)[26]. These frameworks serve as an overarching structure for organizations to follow in order to establish and maintain a strong cybersecurity posture.

Policies are the next level down in the governance hierarchy. They are narrative documents that outline the security requirements of an organization. Policies translate the guidelines from the frameworks into clear and actionable statements that guide the organization's security practices. Frameworks often require organizations to have specific policies in place to ensure adherence to regulatory and compliance requirements.

[26] IANS. (2024) Cloud Security Maturity Model Version 2.0 - *What is the Cloud Security Maturity Model*.

Control Objectives are more specific than policies and focus on the desired outcomes of security controls. They define the goals to minimize risk and maintain a secure environment. For example, a control objective might state that all user logins to cloud platforms must use multi-factor authentication (MFA). Control objectives provide organizations with a clear direction on what terms of security need to be accomplished.

Control Specifications and Implementation Guidance exists at the base level of the governance hierarchy. These are the technical embodiments that satisfy the control objectives. Control specifications are unique depending on the CSP and platform being used, such as AWS or Azure. They outline the technical controls that should be in place to achieve the desired security outcomes. For example, to meet the MFA control objective, a control specification for AWS might require that all users have MFA enabled for console access and that human Identity and Access Management (IAM) users have the "MFA Required" managed policy attached for API access. Compliance can be validated with these control specifications through automation. It is important to note that control objectives can lead to multiple control specifications, as different environments or technologies within the cloud infrastructure may require tailored implementations to meet the desired security outcomes.

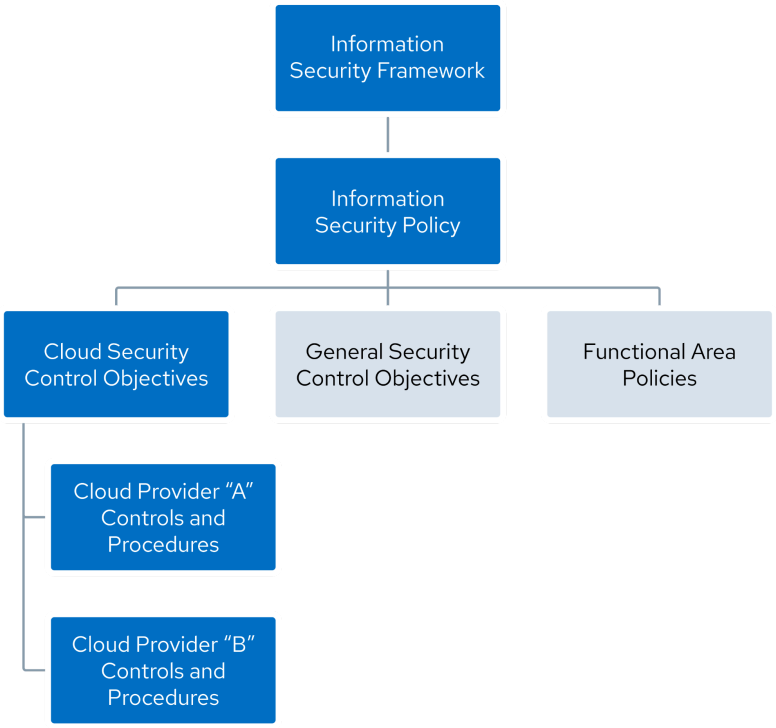


Figure 10: Structured Security Governance Hierarchy

The governance hierarchy provides organizations with a structured approach to information security, ensuring that security practices are aligned with industry standards, and compliance and regulatory requirements. By following this hierarchy, organizations can establish a strong security posture and effectively protect their systems and data.

2.3.1 Foundational Governance Principles & Guidelines

One of the first steps in establishing a solid governance framework for cloud adoption is to determine foundational governance principles. These principles will serve as guidelines for defining the policies, standards, control objectives, and control specifications and implementation guidance that will ensure the privacy, security, and regulatory compliance of cloud environments. The cloud governance framework should include definition of key roles and responsibilities of stakeholders responsible for cloud adoption including Senior Management, IT and technical personnel, business SMEs and security stakeholders. The governance hierarchy ensures alignment with industry standards and regulatory requirements,

helping organizations establish a strong security posture. The figure below illustrates the key elements of the cloud governance process, including risk tolerance, data classification, and control objectives.

Figure 11: Cloud Governance Process

2.3.1.1 Determining Risk Tolerance

Understanding risk tolerance is critical in determining the level of risk that is acceptable when operating in cloud environments[27]. Risk tolerance is the acceptable level of variation that management is willing to allow for any particular risk as the CSC pursues its objectives. This decision considers both qualitative and quantitative factors, such as the financial, legal, reputational, and operational impacts.

By assessing the level of risk tolerance, a CSC can establish a clear security posture and make informed decisions throughout the cloud adoption journey. The CCoE or cloud team should document and inform leadership regarding the risks associated with cloud adoption and operate within the defined risk tolerance.

Risk assessments should be based on consistent analysis of the likelihood and material impact of adverse cyber and operational incident scenarios pertinent to the organization, which can be accomplished using assessment methods like the impact likelihood matrix or Factor Analysis for Information Risk (FAIR)[28].

2.3.1.2 Classifying Data & Assets

Data and asset classification[29] is a critical aspect of cloud governance. CSCs need to classify data and assets based on sensitivity, criticality, and the potential impact associated with loss or compromise. Properly classified data will facilitate the appropriate selection of security controls and ensure compliance with legal, regulatory, and contractual requirements for the protection of data.

Common classifications may include Public, Internal, Confidential, and Highly Confidential. The classification of data and assets impacts where and how they should be stored and processed in the cloud, as well as the controls (legal, regulatory, organizational) which may be required to protect them. A cloud registry should document this classification for easy reference.

[27] Details for cloud risks and categories is provided in *Domain 3: Risk, Audit, and Compliance*.
[28] FAIR Institute. (2024) *What is FAIR?*
[29] Details on data classification are provided in *Domain 9: Data Security*.

Additionally, data location can be a concern in cloud computing because data can be hosted in another jurisdiction, sometimes even without the CSC knowing. Some governments or institutions have limits on data transfers outside their borders, or require additional controls, such as the European Union General Data Protection Regulation (GDPR).[30]

2.3.1.3 Identifying Regulatory & Legal Requirements

Identifying the regulatory and legal requirements that apply to a jurisdiction and/or industry and the types of data being handled is essential. For example, if a CSC is handling personal data of EU citizens, it needs to comply with the GDPR. Similarly, if the CSC deals with health information in the U.S., compliance with the Health Insurance Portability and Accountability Act (HIPAA)[31] is necessary.

In addition to regulatory and other legal requirements, it is important to determine additional requirements based on the specific risks identified during risk assessment.[32] This ensures that the cloud governance framework is comprehensive and aligned with the overall risk management strategy.

2.3.1.4 Requirements, Standards, Best Practices, & Contractual Obligations

To establish a robust governance framework, it is important to align with established standards, best practices, and contractual obligations. This includes adhering to standards and best practices such as CSA CCM, ISO/IEC 27001, ISO/IEC 27017, NIST CSF, or the CIS Benchmarks.

Understanding the contractual obligations of CSPs is crucial. This includes determining the shared security responsibilities between the CSC and CSP, as well as any specific security requirements outlined in a contract between them. Additionally, contractual obligations with CSCs and third-party partners should be considered, as they may impact cloud plans. This includes partnerships in which an organization can play the role of a CSC and CSP in the cloud supply chain.

It is also important to stay informed about current best practices. A CSP will often communicate its recommended best practices for using their services (e.g., AWS Well-Architected Framework, Azure Well-Architected Framework, IBM Cloud Well Architected Framework, Google Cloud Architecture Framework). While a CSC may need to deviate from these practices based on specific needs, best practices serve as a valuable reference point for establishing a secure cloud environment. Cloud security is not one-size-fits-all. It requires customization based on specific context, including industry, risk tolerance, and regulatory requirements. Continuous monitoring and adaptation are essential as new threats emerge and regulations evolve. CSCs are challenged with identifying the appropriate services provided by various vendors and configuring these services to meet requirements and compliance standards.

[30] GDPR is not defined or restricted by the borders of its members. GDPR is also applicable to any company that processes the personal data of any EU citizen or resident.
[31] Health and Human Services. (2024) *Health Information Privacy.*
[32] Details for risk assessment is provided in *Domain 3: Risk, Audit, and Compliance*.

2.3.1.5 Consulting with Key Stakeholders

To move forward with establishing a strong governance framework for cloud adoption, it is important to consult with key stakeholders. This ensures that the cloud security strategy is aligned with the business objectives.

Additionally, a CSC should develop a clear action plan for implementing appropriate security, privacy, and data protection controls to meet the identified requirements. This plan should outline the specific steps and timelines for implementing the necessary controls to ensure the security, privacy, and compliance of the cloud environment.

2.3.2 Cloud Registry

To facilitate effective cloud governance, CSCs can establish a Cloud Deployment Registry and a Cloud Service Registry. These each play a different role in cloud governance (and a CSC may use different terminology).

At a high level the **Cloud Service Registry** is the list of which cloud platforms and services are approved for which kinds of data (e.g., SaaS provider X is approved for data with classification Y).

A **Cloud Deployment Registry** is a tool used in maintaining an inventory of an organization's cloud presence across multiple providers and services. This is a centralized repository that maintains information about the organization's deployed cloud resources, including details such as ownership, usage, and security controls. This cloud registry helps ensure transparency and accountability in cloud resource management. Similar in nature to an Asset Registry, by having a comprehensive cloud registry, a CSC can effectively manage and secure its cloud resources.

Some CSCs use a standard risk register to track cloud services and deployments. This is acceptable as long as that register is generally available to security and operations teams, is kept up to date, and includes the information described in sections included here.

When building a cloud deployment registry, it is important to include elements like these:

1. **Cloud Service Provider (CSP):** Document the cloud service provider for each account, including major providers like AWS, Azure, and GCP, as well as SaaS platforms such as Salesforce and Microsoft 365. This information helps in understanding the underlying infrastructure and services utilized.

2. **Environment ID:** Assign a unique identifier to each cloud environment to facilitate tracking and management. This ID should appear in logs and other monitoring tools, providing a precise reference point for each environment.

3. **Descriptive Name:** Provide a meaningful name that accurately describes the purpose or nature of each cloud environment. This makes it easier to identify and understand the role of each environment within the organization.

4. **Compliance Classification:** Categorize each environment based on regulatory and compliance needs, such as PCI DSS, HIPAA, GDPR, and so on. Proper classification ensures that the appropriate security measures and controls are applied to meet compliance requirements.

5. **Risk Classification**: Assess and label the risk level of each environment to align with the CSC's risk management strategy. This helps prioritize resources and efforts for risk mitigation and ensures that the appropriate level of security controls is implemented.

6. **Environment Classification:** Distinguish between different types of environments, such as development, staging, and production. This classification helps manage and govern each environment based on its specific requirements.

7. **Owner:** Identify the business owner responsible for each cloud environment. This ensures accountability, responsibility, and clear lines of communication for decision-making and resource allocation.

8. **Technical Contact:** Designate a point of contact for technical issues and operational management of each environment. This helps streamline communication and ensures prompt resolution of technical challenges.

9. **CSP Contacts**: Include contact information for customer support and account management at the CAP. This information is essential for addressing any service-related issues and maintaining a healthy relationship with the CSP.

2.3.2.1 Cloud Deployment Registry Capabilities

Having a well-maintained cloud deployment registry offers several benefits:

- Better visibility and control over cloud resources: A comprehensive registry allows CSCs to clearly understand, document and track their cloud presence, enabling effective resource management, optimization, and change management.

- Consistent application of governance frameworks: By having a detailed registry, CSCs can ensure that appropriate governance frameworks, policies, and procedures are consistently applied across all environments.

- Support for incident response: The cloud registry provides all the necessary contact information, enabling swift incident response and effective coordination during security incidents or operational disruptions.

- Compliance with policies and regulations: By classifying environments based on compliance needs, CSCs can confirm that they adhere to internal policies and external regulations, minimizing non-compliance.

Begin by compiling the cloud registry if it is not already in place, identifying all necessary elements and gathering the required information for each environment. Review and update the cloud registry regularly at planned intervals. Additionally, update the registry whenever there are material changes to the business environment, legal/regulatory/contractual changes, changes in the cloud landscape, or organizational

structure. This ensures that the registry remains accurate and up-to-date, supporting effective governance and risk management.

2.3.3 Cloud Security Frameworks

One of the key purposes of a framework is to organize and prioritize security control objectives. These objectives represent the specific goals that an organization sets to achieve desired security outcomes. Frameworks provide a structure for categorizing these objectives and help organizations determine the most effective ways to implement and manage them. By organizing security control objectives, frameworks enable CSCs to take a systematic approach to cloud security and ensure that all necessary controls are in place.

A cloud-specific security framework is designed specifically for cloud environments, considering the unique characteristics of cloud computing. These frameworks address aspects such as on-demand resource allocation, SRMs, and rapid elasticity. By using a cloud-specific framework, CSCs can ensure that their security program aligns with the specific requirements and challenges of the cloud.
Examples of cloud relevant frameworks[33] are:

- CSA Cloud Control Matrix (CCM) - learn more below.
- ISO/IEC 27017:2015
- BSI Cloud Computing Compliance Criteria Catalog (C5)
- NIST SP 800-53 Rev. 5 - Security and Privacy Controls for Information Systems and Organizations
- PCI DSS Council Cloud Computing Guidelines

If a CSC already uses an information security framework, but that framework does not effectively cover the unique aspects of cloud, it can consider the concept of a "sidecar" or supplemental framework. This concept involves using a cloud-specific security framework alongside an existing primary framework. Many existing security frameworks were not originally designed for cloud computing and may not adequately address cloud-specific activities. By using a sidecar framework, CSCs can focus on cloud security activities while still utilizing existing frameworks for other areas of security.

2.3.3.1 NIST Cybersecurity Framework

While several security frameworks exist, the NIST Cybersecurity Framework (CSF)[34] offers to industry, government agencies, and other organizations a widely recognized structured security approach, even though it is not cloud-specific. The framework also known as the CSF Core offers a six-function taxonomy of high-level cybersecurity outcomes that can be used by any organization — regardless of its size, sector, or maturity — to better understand, assess, prioritize, and communicate its cybersecurity posture and program. The following CSF functions, provide a high-level, strategic view of the lifecycle of an organization's management of cybersecurity risk:

[33] For additional details about these frameworks, take the Certificate of Cloud Auditing Knowledge (CCAK) course available through ISACA.
[34] NIST. (2024) *The NIST Cybersecurity Framework (CSF) 2.0*

- **GOVERN (GV):** Establish and monitor the organization's cybersecurity risk management strategy, expectations, and policy.
- **IDENTIFY (ID):** Help determine the current cybersecurity risk to the organization.
- **PROTECT (PR):** Use safeguards to prevent or reduce cybersecurity risk.
- **DETECT (DE):** Find and analyze possible cybersecurity attacks and compromises.
- **RESPOND (RS):** Take action regarding a detected cybersecurity incident.
- **RECOVER (RC):** Restore assets and operations that were impacted by a cybersecurity incident.

2.3.3.2 CSA Security, Trust, Assurance, & Risk Registry

CSA Security, Trust, Assurance, and Risk (STAR) Registry is a program initiated to advance transparency and confidence in cloud services. The program offers a framework for CSPs to document their security practices and for CSCs to evaluate the security posture of CSPs.

The CSA STAR program comprises two primary components:

1. **CSA STAR Attestation:** Here, CSPs self-assess their security controls against the CSA CCM and publicly disclose the results of their assessments. This provides transparency into a provider's security posture and enables CSCs to make informed decisions about utilizing their services.
2. **CSA STAR Certification:** This entails an independent third-party evaluation of a CSP's security controls against the CSA CCM and other recognized industry standards like ISO/IEC 27001. Achieving CSA STAR Certification indicates that a CSP has implemented robust security measures and practices.

By facilitating standardized security assessments and promoting transparency, the CSA STAR program empowers CSCs to assess the security, privacy, and compliance practices of CSPs. This, in turn, helps CSCs make risk-aware decisions when selecting and utilizing cloud services, fostering trust and reliability in the cloud industry.

2.3.3.3 Cloud Controls Matrix

The CSA CCM v4[35] is a library of control objectives that structures 17 domains. It offers comprehensive coverage of a wide array of security topics, ranging from governance and risk management to operational security and data privacy. This makes it a valuable resource for CSPs and CSCs looking to enhance their cloud security.

[35] CSA. (2024) Cloud Controls Matrix (CCM)

One of the key strengths of the CCM is its alignment with leading standards such as ISO/IEC 27001/27002, PCI DSS, NIST CSF, and so on. By harmonizing with these established frameworks, the CCM ensures that organizations can achieve compliance across multiple standards and regulations. This alignment also extends to the STAR program, further enhancing its credibility and relevance in the industry.

The CCM is tailored to cloud environments, making it well-suited for securing multi-tenant, distributed, and dynamic cloud systems. The focus on the unique challenges of cloud computing sets it apart from more generic security frameworks like NIST CSF. Additionally, the CCM allows for control customization, enabling CSCs to adapt the security controls to their specific cloud architectures, delivery models (IaaS, PaaS, SaaS), and compliance needs.

Another key benefit of the CCM is its support for cloud governance. It assists CSPs in establishing and maintaining a solid cloud governance program that effectively manages and oversees cloud risks. This is valuable in ensuring that cloud deployments are aligned with organizational objectives and comply with relevant regulations.

The CCM is continuously updated to reflect the latest cloud security best practices. CSPs and CSCs can rely on the CCM as a reliable and relevant resource for their cloud security needs by staying up to date. Overall, frameworks are essential tools for CSCs looking to establish a robust and effective cloud security program. By selecting a cloud-specific framework, using a sidecar framework, and organizing security control objectives, CSCs can comprehensively address all security controls and align their programs with the unique requirements of the cloud.

2.3.4 Policies

Information security policies are important for establishing a strong security framework. Policies govern the protection of an organization's information assets and outline the necessary control objectives. For example, see the NIST CSF Policy Template Guide published by CIS.[36]

To ensure the effectiveness of policies, it is recommended that organizational leadership formally approve them. Such approval lends weight and authority to the policies and demonstrates the leadership's commitment to their enforcement. With leadership's backing, policies are implemented more consistently and effectively.

Compliance with various regulatory and legal frameworks, such as GDPR for data privacy or Sarbanes-Oxley (SOX) for financial reporting, often plays a significant role in the development of information security policies. Organizations need to have specific policies to be in place to meet these compliance standards, which are crucial for ensuring compliance with external requirements and avoiding penalties.

[36]CIS. (2024) *Policy Template Guide.*

2.3.4.1 Types of Policies

There are several key examples of information security policies organizations commonly implement:

- **The Information Security Policy** is the highest-level policy defining how the information security program should be run. It ideally refers to other policies and documents, such as control objectives, rather than trying to include all the specific technical requirements.
- **The Acceptable Use Policy (AUP)** defines the appropriate use of an organization's IT resources.
- **The Remote Work Policy** outlines the security measures and behaviors expected from employees when working remotely.
- **The Use of Cloud Services Policy** sets requirements for using data within the cloud.
- **The Data Handling Policy** describes how to classify, handle, store, and dispose of data to maintain its confidentiality, integrity, and availability.

Information security policies are essential, actionable documents that drive security practices and reinforce a strong cybersecurity culture by providing a clear framework for security practices and behaviors. They ensure that the right level of protection is applied to different environments. It is important for employees to familiarize themselves with their specific information security policies and understand their role in upholding the policies to contribute to the overall security posture.

2.3.5 Cloud Security Control Objectives

Cloud security control objectives serve as a checklist of desired or required controls within a cloud environment. These objectives are written to be outcome-focused, meaning that they prioritize the results rather than specifying how to implement them. The objectives should be measurable following the S.M.A.R.T. method (which stands for Specific, Measurable, Achievable, Relevant, and Time-bound).

The control objectives should be platform-agnostic, meaning no specific CSP or technology bounds them. This makes control objectives applicable to a wide range of cloud environments, ensuring their relevance and effectiveness.

Each security control implemented should map back to at least one specific control objective. This ensures that every measure has a defined purpose and contributes to the overarching security goals of the cloud environment.

Lastly, controls should be clearly defined, avoid vague directives which can be interpreted too broadly, and instead focus on the actual security outcome desired. The objectives should be detailed and provide clear direction, but not so granular that they become prescriptive methods.

In summary, cloud security control objectives guide the establishment of a robust security posture in the cloud. They are adaptable, measurable, and outcome-oriented, aligning with the dynamic nature of cloud

computing. Organizations can enhance cloud security and mitigate potential risks by following these objectives.

2.3.5.1 Mapping Control Objectives to Frameworks

The control objectives in a security program should align with the framework being used. This alignment provides structure and ensures that the program covers all necessary areas. The alignment may correspond to the organizational structure and operational responsibilities in larger organizations.

The framework is the foundation for security programs, outlining the overall structure and approach. The control objectives, on the other hand, define the desired outcomes and objectives. Linking the control objectives to the framework ensures the program has proper coverage and a clear scope.

It is important to note that if control objectives do not align with the framework, it indicates a gap in the strategy. This misalignment should be addressed to ensure the program is comprehensive and effective. Similarly, insufficient control objectives for a specific category in the framework may suggest an operational deficiency that needs to be addressed.

To maintain this mapping, it is recommended to include it directly in a control objectives repository. This allows for easy reference and ensures that the alignment is documented. The CSA CCM serves as an exemplary framework with a comprehensive list of mappings to relevant frameworks, standards, and legal and regulatory requirements, making it a valuable resource for managing the complexity of governance and compliance.

2.3.5.2 Control Specifications

Control specifications are an essential part of ensuring the security of a cloud environment. These specifications outline the detailed technical control capabilities that must be implemented to meet specific security requirements. It is important to note that control specifications should be vendor and technology-specific, meaning that there can be significant differences between different CSPs.

For example, consider the requirement of implementing an MFA. The technical procedure to enable MFA will vary depending on the cloud provider. Each CSP has its own unique way of configuring and enforcing MFA, and CSP-specific control specifications will need to be created.

Another area where control specifications can differ is network security. By default, Azure sets its networks to be open to inbound connections, meaning additional network security group settings must be configured to ensure a secure environment. On the other hand, AWS defaults its networks to a least-privilege setting for inbound connections, providing a higher level of security by default. Additionally, Azure supports both allow and deny network security rules, while AWS only supports allow rules and denies all other traffic. These differences highlight the need to tailor control specifications to the specific CSP being used.

Furthermore, control requirements may vary based on the classification of the data or resource. For example, if a CSC works with personally identifiable information (PII), it may have stricter control

requirements for the default deployment settings. On the other hand, data that is deemed public may have less restrictive control requirements. It is important to consider the sensitivity of the data or resource when defining control specifications.

In some cases, controls cannot be fully implemented within the CSP's ecosystem, and a third-party tool may be necessary. These tools can provide additional functionality and enhance the security posture of cloud environments. Defining control specifications that outline how a tool should be configured to meet the control objectives and requirements is important when using a third-party tool. This ensures that the third-party tool is effectively integrated into the security strategy.

Control specifications should evolve over a period of time to account for advancements in technology, new offerings, and new threat and attack vectors. The revision and adoption of revised specifications is critically important.

2.3.6 Shared Security Responsibility Model

The shared security responsibility model (SSRM)[37] is a fundamental cloud security concept. It establishes that the CSC and the CSP have separate but complementary obligations to ensure the cloud service operates properly and remains protected.This sharing of responsibilities can extend beyond the CSP and CSC to other parties that engage in delivering the service, such as supporting CSPs, agents and cloud platform integrators.

In terms of transparency within their own organization, the CSCs need to have a clear line of sight with how the internal IT team has mapped the shared responsibility model across the different control functions within their organization. The organization needs to carry out an appropriate level of discovery for the number of workloads it plans to deploy or services it will consume in the cloud, such as: how many people are needed to operate the controls so the service is trustworthy? Certain decisions will flow from this: does the organization need to scale up internally (by hiring full-time staff or working with external consultants)? Within an organization, the responsibilities need to be explicitly stated and reflected in the way the business runs its own operations, to avoid gaps between where its responsibilities end and the CSP's begin. It may be useful to outline those responsibilities – and their exact interpretations – in an agreement between both sides.

Cloud services by their nature change much more rapidly than traditional IT services; with new services released and old services deprecated. Therefore the shared responsibility model can evolve over time as the provider adds new services and upgrades existing services. For example, if a CSP decommissions a particular service offering, it's important to understand who is responsible for managing the lifecycle of data migration and conversion to the new service. Consequently, all parties need to consider their respective responsibilities through the lifecycle of services, not just at a point in time.

It is also worth noting that the cloud customer may be increasing their risk based on the nature of the cloud service. Non-core services in the cloud can be at risk being replaced or substantially changed. The dependency chain with core services can give confidence that it will continue for some time to come, and changes to fundamental services carry a much higher cost to all users of that service.

[37] Additional details provided in *Domain 1: Cloud Computing Concepts & Architectures*

In summary, the SSRM delineates security and some governance responsibilities, which vary based on the cloud service model and the security domain.

2.3.6.1 Responsibility & Accountability

CSCs may not have full visibility into the underlying infrastructure and processes of their CSPs, but are still accountable for governing use of the cloud. Additional internal challenges are presented by the usage of unauthorized cloud services known as "shadow IT," where an employee or department utilizes cloud services without the knowledge of IT and security teams. An awareness program covering shadow IT and associated risks should be designed and implemented to manage this risk[38].

2.4 Key Strategies & Concepts

This section explores important strategies and concepts in cloud computing security and governance. It starts with DevOps and DevSecOps, which can really help make software development smoother and safer. It also explores the Zero Trust security strategy, which focuses on continually verifying and controlling access to keep out new threats. Finally, it looks at how artificial intelligence (AI) and machine learning (ML) are used in cloud security to predict and spot problems quickly. Overall, this section shows how these ideas shape security and governance in cloud computing, making it easier to understand and control.

2.4.1 DevOps

DevOps is a set of practices that combines software development (Dev) and IT operations (Ops) with the goal to shorten the secure software development lifecycle (SSDLC[39]) while delivering features, fixes, and updates frequently in close alignment with business objectives. In cloud computing, DevOps methodologies are integral for deploying and managing applications and infrastructure due to the cloud's agility, scalability, and flexibility. CSCs can extend the tactical DevOps approach for application development to a more strategic, agile organization-wide approach based on automation.

2.4.1.1 DevSecOps

DevSecOps is a security approach that integrates security practices throughout the entire SSDLC, from initial design to deployment and ongoing monitoring. By integrating security practices at every stage, DevSecOps enables CSCs to identify and mitigate potential security risks and vulnerabilities early on, resulting in more secure and resilient software applications.

CSA has identified six key pillars that support a secure and efficient DevSecOps implementation, which are depicted in the figure below.[40] These pillars, aligned with the five stages of the DevOps Delivery

[38] CSA. (2023) *Defining Shadow Access: The Emerging IAM Security Challenge*
[39] SSDLC incorporate security into all stages of the development process. Additional details are provided in *Domain 10: Application Security.*
[40] CSA. (2024) *Six Pillars of DevSecOps Series.*

Pipeline, create a comprehensive foundation for building security into the very fabric of the software development process.

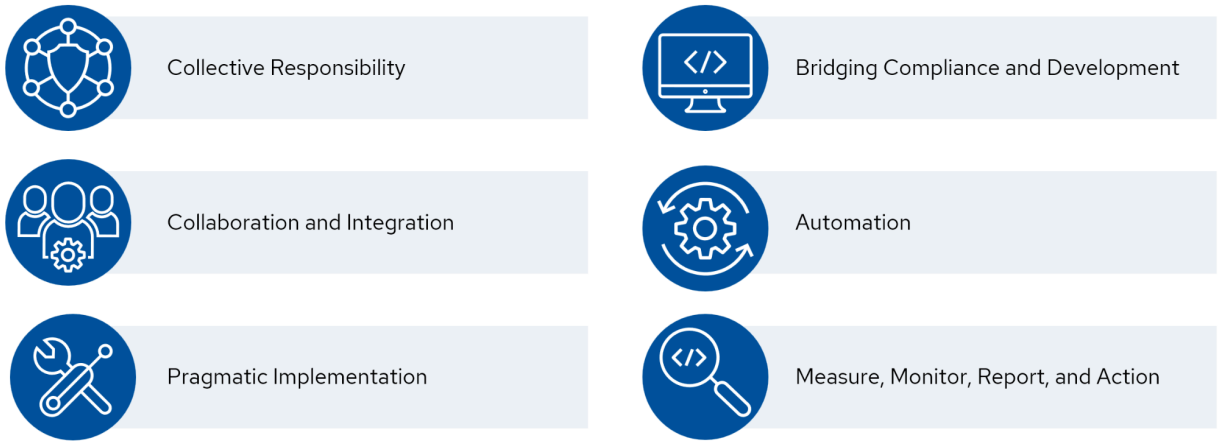


Figure 12: CSA Six Pillars of DevSecOps

Figure 13: CSA Five Stages of the SSDLC Pipeline

By leveraging the Six Pillars and the Five Stages of the CSA DevSecOps Delivery Pipeline, a CSC can build a secure and efficient development process, delivering high-quality, reliable applications with security and compliance built in from the ground up.

2.4.2 Zero Trust Security Strategy

Traditional "castle and moat" perimeter security architectures are ineffective in the current era of cloud computing and the remote workforce. Increasingly sophisticated threat actors are adept at exploiting any exposed technical or human vulnerability in modern, distributed enterprise networks that heavily leverage Internet connectivity. Significant security breaches are frequently publicized around the world. These incidents can cost CSCs more than what they would have spent implementing a modern security architecture based on the increasingly popular and continually maturing Zero Trust[41] [42] security strategy.

Zero Trust is a holistic security strategy that encompasses cloud/multi-cloud, internal and external partner/stakeholder user endpoints, on-prem/hybrid systems, and is inclusive of both operational technology (OT) and IoT. Zero Trust implementation involves defining an enterprise security architecture following risk-based design principles and leveraging multiple products/services and established security

[41] Additional details on Zero Trust are provided in *Domain 12: Related Technologies & Strategies.*
[42] CSA. (2024) *Zero Trust Resource Hub*.

principles, such as need to know and least privilege. It advises secure design from the inside out versus outside in, and enhances visibility and facilitates automated, real-time responses that allow organizations to keep pace with the evolving threat environment.

Zero Trust accounts for the increasingly sophisticated and aggressive threat actor landscape. It is a cybersecurity strategy premised on the idea that no user or asset is to be implicitly trusted. It assumes that a breach already has or will occur, and therefore, a user should not be granted access to sensitive information by a single verification performed at the organizational perimeter. Instead, each user, device, application, and transaction must be continually verified.[43] Key design principles of Zero Trust include:

- moving access controls closer to organizational resources
- denying access by default
- continuously validating explicitly authorized, granular access rights for users and devices
- implementing network micro segmentation to limit lateral movement
- monitoring all access closely
- encrypting all network traffic
- analyzing access patterns in real-time to detect and respond to anomalies promptly

When implemented correctly, a Zero Trust strategy and an architecture that supports it have the potential to provide a simpler, more secure, and flexible environment for business operations.

2.4.3 Artificial Intelligence & Machine Learning

Using AI or ML in cloud security involves the use of automated reasoning. ML, a sub-field of AI, involves algorithms that ingest data, learn from it, and then apply what has been learned to make informed decisions.

Cloud providers often utilize AI techniques to detect security misconfigurations and threats by analyzing vast amounts of data, which can be too complex to rely upon human oversight alone. For example, unsupervised learning, a type of ML, is used to identify certain threats from legitimate access without relying on labeling. These techniques are especially powerful when coupled with automated remedial measures.

Generative AI, including Large Language Models (LLMs), learn patterns and structure from large datasets to generate content, such as text, images, and video. These AI models often operate in the cloud due to their high but variable computational power and data storage requirements. Running Generative AI in the cloud raises important considerations for data privacy and application design, particularly when dealing with shared resources where data segregation and protection are paramount.

Recognizing the complexities introduced by AI and ML technologies, NIST has released an AI Risk Management Framework[44] with the aim of improving the governance and trustworthiness around the usage of AI systems and models. The framework itself provides guidance around identifying risks associated with AI systems and recommends a four-step approach to govern, map, measure, and manage the risks throughout the AI lifecycle.[45]

[43] CISA. (2022) THE PRESIDENT'S NATIONAL SECURITY TELECOMMUNICATIONS ADVISORY COMMITTEE - *NSTAC REPORT TO THE PRESIDENT*. Page 1, adopted as the official CSA definition of Zero Trust.
[44] NIST (2024) *AI RISK MANAGEMENT FRAMEWORK*.
[45] CSA. (2024) AI Governance & Compliance Resource Links Hub

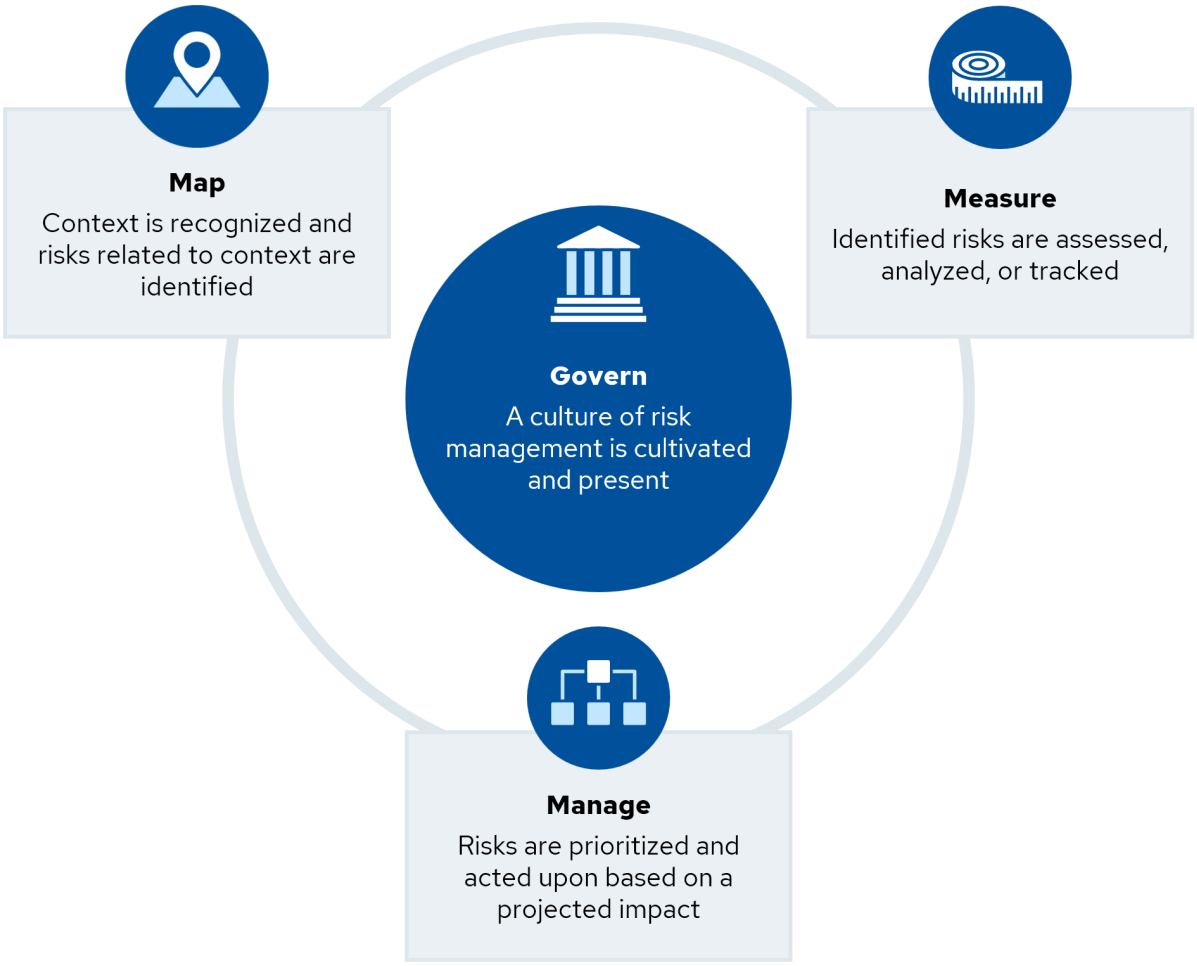


Figure 14: AI Risk Management Framework

In tandem with the NIST framework, the ISO/IEC has released ISO/IEC 42001:2023, *Information technology Artificial Intelligence Management system*, that specifies requirements for establishing, implementing, maintaining, and continually improving an Artificial Intelligence Management System within organizations. It is designed for entities providing or utilizing AI-based products or services, ensuring responsible development and use of AI systems. It addresses the unique challenges AI poses, such as ethical considerations, transparency, and continuous learning. For organizations, it sets out a structured way to manage risks and opportunities associated with AI, balancing innovation with governance.

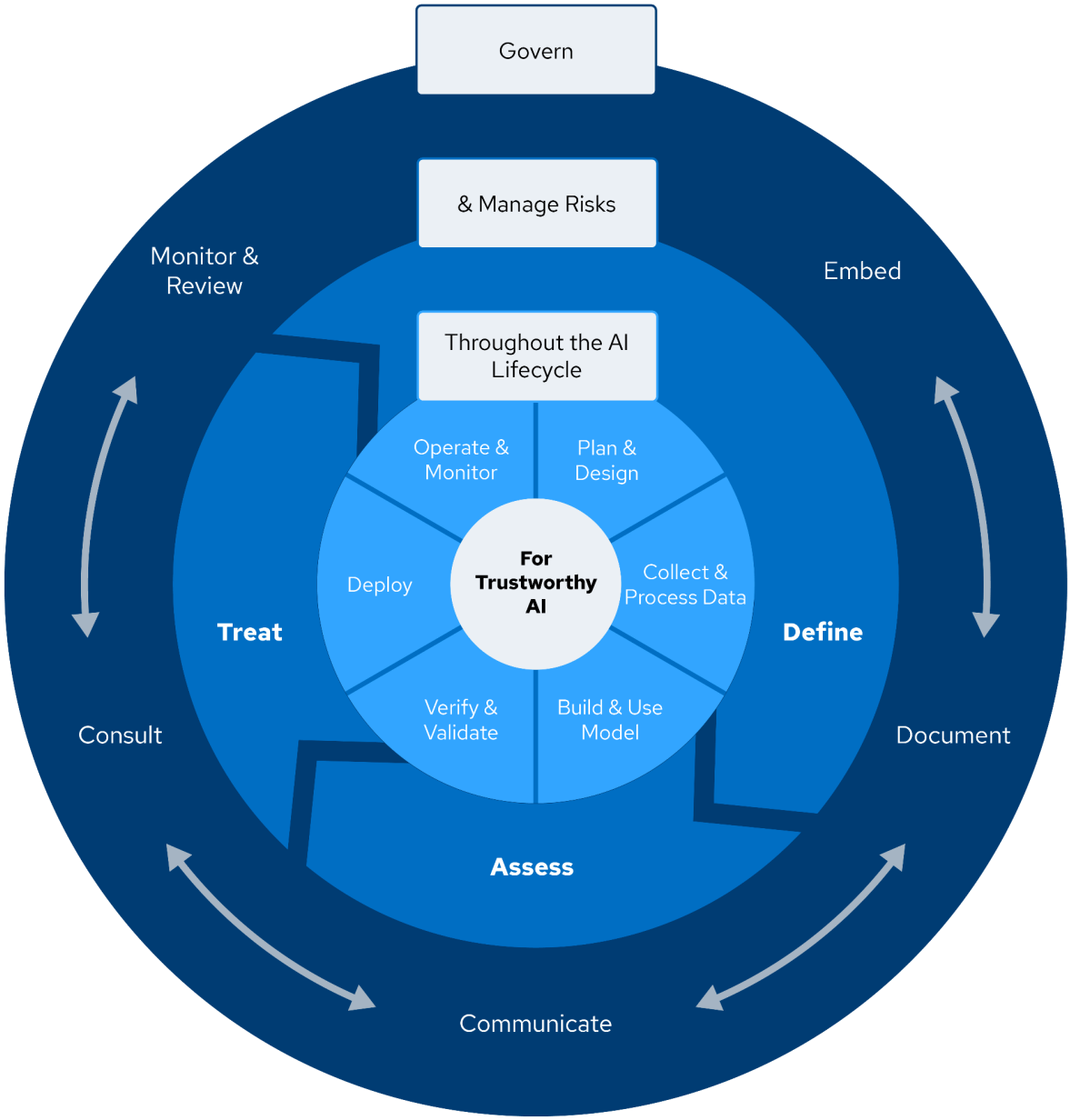


Figure 15: ISO/IEC AI Risk Management Lifecycle

The implementation of strategies such as DevSecOps, Zero Trust, and AI/ML integration, alongside adherence to frameworks like ISO/IEC 42001:2023, ensures a robust and secure cloud environment. These practices not only mitigate risks but also enhance the trustworthiness and reliability of cloud-based systems.

Summary

Effective cloud governance is a critical aspect of managing IT infrastructure in the cloud, and integral to effective corporate governance. Traditional data centers, where centralized teams have control over the entire IT infrastructure, require a different approach to governance than does the cloud.

Organizational adjustments are essential for effective cloud governance. Traditional hierarchies may not be suitable any longer. It is important to reassess structure to establish clear roles and responsibilities that align with the cloud environment.

It is also important to identify and document risk, regulatory, and security requirements. In the cloud, data may be stored and processed in a multitude of locations subject to different regulations and compliance standards. Identifying and documenting these requirements is essential for ensuring that the cloud infrastructure remains compliant and secure.

The chosen governance framework provides a high-level overview of the governance requirements and sets the direction for the rest of the governance process. CSCs should always align governance requirements with business strategies to ensure adoption of the right controls and guidelines.

CSCs should establish clear roles and responsibilities, establish a CCoE, assess and manage risks, and establish processes and procedures for governance of cloud assets, access, and internal resources. Governance must include establishing key metrics and measurements and re-assessment periodically of the security posture of the organization and of the CSPs[46].

Recommendations

- Understand that the technical and operational differences of cloud computing require new governance approaches to maintain effective security.
- Adapt your organization structure with concepts like the CCoE and/or CACI to improve your ability to govern the cloud.
- Implement a Security Champions program to distribute security knowledge more effectively, especially to development and cloud teams.
- Collect and understand your foundational requirements, including your risk tolerance, compliance obligations, business needs, and existing cloud usage.
- Starting with a security framework, organize your security policies, control objectives, and control specifications in a clear governance hierarchy.
- Understand the security and governance implications of other strategies and concepts commonly seen when working in the cloud, including DevOps, Zero Trust, and AI.

Additional Guidance

- Cloud Security Technical Reference Architecture | CISA
- Communicating the Business Value of Zero Trust | CSA

[46] For additional details on metrics, take the Certificate of Cloud Auditing Knowledge (CCAK) course available through ISACA.

- Zero Trust Guiding Principles | CSA
- SaaS Governance Best Practices for Cloud Customers | CSA
- COBIT Framework | ISACA
- ISO/IEC TR 3445:2022 Information Technology Cloud Computing Audit of Cloud Services

Domain 3: Risk, Audit and Compliance

This domain is dedicated to the core aspects of cloud security, as they relate to risk, audit, and compliance matters. However, it should be noted that this domain does not replace the need for thorough training and experience in comprehensive risk, audit, and compliance. For those seeking to delve deeper into these subjects, the *Certificate of Cloud Computing Audit Knowledge (CCAK)* offered by CSA, in collaboration with ISACA, is recommended.

When it comes to cloud risks, this domain explores approaches to evaluating cloud service providers (CSPs). It covers the establishment of cloud risk registries and the implementation of approval processes. Additionally, it references CSA's *Top Threats*[47] to provide context on common risks.

This domain outlines compliance and auditing, the different types of compliance, and the concept of compliance inheritance. To aid in the governance, risk, and compliance (GRC) process, the domain introduces various tools and technologies. This includes policies, procedures, controls, the role of automation, the software bill of materials (SBOM), and related technologies. Collectively, these components support the governance framework and help manage the intricate landscape of cloud computing risks and compliance requirements.

Learning Objectives

In this domain, you will learn to:

- Define, categorize, and use tools to manage cloud risk.
- Identify the regulatory and compliance constraints for which your cloud-based environment must undergo audits.
- Identify a set of technical and non-technical tools used when managing GRC.

3.1. Cloud Risk Management

Effective cloud risk management is mandatory in today's digital landscape, where organizations increasingly rely on cloud services. This section delves into the importance of understanding cloud risks and provides insights on establishing a cloud risk profile, assessing CSPs, maintaining a cloud risk registry, and conducting risk assessments, threat intelligence, and threat modeling. By implementing robust cloud risk management practices, organizations can proactively identify and mitigate potential risks, ensuring the security and resilience of their cloud environments.

[47] CSA. (2021) Research Topic: Top Threats

3.1.1 Cloud Risks

Let's start with an example. A company has a cloud storage bucket filled with personal information on customers. We call this an **asset**, which to an attacker (also known as a **threat actor**) is a **target**. One of the weaknesses of a cloud storage bucket is that it may be misconfigured. We call that a **vulnerability**, and to an attacker, this represents an **attack vector**.

A **risk** is that the personal data in the bucket leaks out, and the company gets fined by a regulator. Another risk could be that, through some action, the data becomes unavailable or corrupted.

A **control** or **countermeasure** is a way to reduce the risk. Typical controls here would be any policy that prevents these storage buckets from being accessible to the whole internet, or more specifically: the threat actor.

Ideally, we'll have enough controls to reduce the risk down to an acceptable level. This involves understanding what the important assets and threat actors are. This process is called **threat modeling** and is discussed in other places in this guidance, such as application security. Threat modeling, in a cloud world, starts with identifying the various places and cloud services where data is stored, and how data flows between them. See also CSA research[48].

The following examples show some of the most common risk factors and categories, both general and security risks. We also recommend reviewing the Cloud Security Alliance's *Top Threats*[49] research report. In the 2022 edition, the 'Pandemic Eleven', these were the top categories:

- Insufficient Identity, Credentials, Access, and Key Management
- Insecure Interfaces and APIs
- Misconfiguration and Inadequate Change Control
- Lack of Cloud Security Architecture and Strategy
- Insecure Software Development
- Unsecured Third-Party Resources
- System Vulnerabilities
- Accidental Cloud Data Disclosure
- Misconfiguration and Exploitation of Serverless and Container Workloads
- Organized Crime/Hackers/Advanced Persistent Threat (APT)
- Cloud Storage Data Exfiltration

There are many other sources of cloud threat intelligence. Consult the CSA website for up-to-date information. Additionally, the **MITRE ATT&CK®**[50] framework provides a comprehensive matrix of threat actor tactics.

[48] CSA. (2021) Publications: Cloud Threat Modeling
[49] CSA. (2021) Research Topic: Top Threats
[50] MITRE. (2024) Cloud Matrix

3.1.2 Establishing a Cloud Risk Profile

A cloud risk profile is an important assessment for organizations relying on cloud services. Serving as a foundational guide for cyber risk analysts and auditors, the profile provides insights into the CSC's risk landscape. It enables organizations to understand their risk exposure, ensuring alignment between their cloud strategy and business objectives within their risk appetite.

3.1.2.1 Cloud Risk Profile Starter Questions

The first step in risk evaluation is to establish the organization's cloud risk profile, which can be achieved with risk evaluation questions. Consider the following questions as a starting point for organizations to evaluate their risk profile. They are not necessarily exhaustive or the exact questions a given cloud service customer (CSC) should use.

- **Alignment with business strategy:**
    - How is cloud computing technology aligned with the overall business strategy?
    - How does adopting cloud computing support the organization's strategic objectives?
    - What benefits does cloud computing offer to the business model, and how does it enhance operational efficiency or market competitiveness?
- **Information security or cybersecurity policy:**
    - Have information security or cybersecurity policies been updated to reflect cloud technology management?
    - How frequently is the information security or cybersecurity policy reviewed and updated to incorporate changes due to cloud computing adoption?
    - Are all relevant stakeholders involved in reviewing and updating the information security or cybersecurity policy?
- **Third-party risk assessment for cloud computing:**
    - Does third-party risk assessment include risks specific to cloud computing technology (e.g., based upon data privacy and security laws)?
    - How comprehensive is the third-party risk assessment in evaluating CSPs?
    - Does the assessment address risks specific to data privacy, security laws, and compliance with industry regulations?
    - Does the assessment of cloud risks align with the organization's overall risk appetite and risk tolerance?

- **Cloud migration risk assessment:**
    - Was a thorough risk assessment conducted before cloud migration, specifically focusing on cloud technology-related risks?
    - How were the cloud migration risk assessment findings integrated into the overall risk management plan?
- **Inventory of CSPs and contracts:**
    - Is there a centralized inventory of all CSPs, including details of contracts, service-level agreements (SLAs), and third-party assessments or attestation reports?
    - How is the performance and compliance of CSPs monitored and reviewed against the contracts?
- **Business continuity/disaster recovery (BC/DR) plan:**
    - Has the organizational BC/DR plan been updated to reflect the cloud adoption?
    - How has the organization's BC/DR plan been adapted for cloud services?
    - Are there specific considerations for BC/DR in the event of a cloud service failure?
    - Does the BC/DR plan document responsibilities and dependencies on CSPs?
- **Privacy policy updates post-cloud migration:**
    - Has organizational privacy policy been updated to include cloud adoption?
    - How has the organizational privacy policy been revised to address data collection, data storage, processing, management, retention, and destruction in the cloud?
    - Does the updated privacy policy adequately cover data residency, cross-border data transfers, and user consent mechanisms?
- **Incident management policy:**
    - How has the incident management policy been updated to include incidents involving cloud services?
    - Are there clear procedures for responding to security breaches or data leaks involving cloud-based assets?
    - Are incident-related communication channels with CSPs defined and documented?
- **Secure software development lifecycle (SSDLC):**
    - Has the organizational SSDLC been updated to reflect the cloud migration?

- How has the SSDLC been modified to incorporate cloud-specific security considerations, particularly as they concern API security, identity and access management, and encryption?
- Are cloud-aware security tools and practices integrated into the development, deployment, and maintenance phases of the SSDLC?

3.1.2.2 Using the Cloud Risk Profile

The results of establishing a cloud risk profile can be used to inform the rest of the cloud risk management process. The goal is to identify both risk tolerance and the current cloud risk posture. A cloud risk profile should represent how well prepared a CSC is for migrating to cloud in terms of risk.

CSCs should also consider adopting cloud security frameworks and standards, such as those provided by CSA, to help benchmark practices in more detail and ensure comprehensive risk management in cloud environments.

3.1.3 Understanding Cloud Risk Management

Understanding cloud risk management involves a structured approach to identifying, assessing, and addressing risks associated with cloud computing. The risk management and methodologies used in cloud computing are not different from the ones adopted in the on-premises world. However, changes maybe expected with some of the specific actions taken during the definition of the scope and environment and the risk evaluation and treatment process.

The European Network and Information Security Agency (ENISA) *Risk Management Process*[51] provides a framework that organizations can adapt to manage these risks effectively. This process is designed to be integrated into a CSC's broader operational processes, ensuring a comprehensive approach to risk management. Here is an expansion on the key components of this process:

- Corporate risk management strategy
  - Including risk communication, awareness, and consulting
- Risk assessment
- Risk treatment
  - Including risk acceptance
- Interface to other operational and product processes
- Monitoring and review of plans, events, and quality

[51] ENISA. (2022) The Risk Management Process

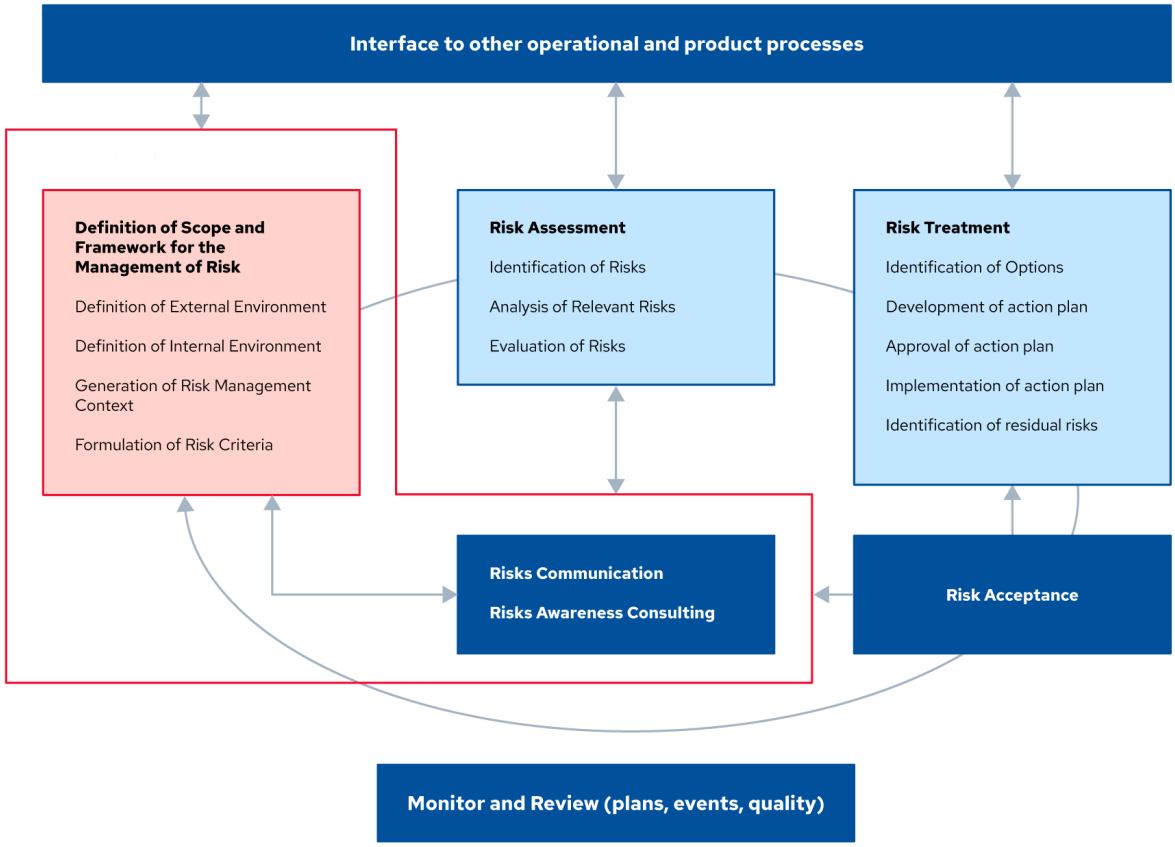


Figure 16: Comprehensive Cloud Risk Management Framework

3.1.3.1 Corporate Risk Management Strategy

This phase is about creating a context for risk management and formulating specific risk criteria to guide the assessment process.

- **Definition of scope and framework for risk management:** Establish clear boundaries for what the risk management process covers, including the specific aspects of cloud services and operations.
- **Definition of external environment:** Understand the external factors, such as regulatory requirements, market conditions, and technological advancements, that could impact cloud operations.
- **Definition of internal environment:** Assess internal factors, including organizational structure, culture, risk appetite, risk tolerance, and existing controls, that influence managing of risks.
- **Generation of risk management context:** Create a context that aligns with organizational objectives, enabling a focused and effective risk management strategy.

- **Formulation of risk criteria:** Develop criteria for evaluating risks, including the likelihood of occurrence and potential impact, tailored to the specific context.
- **Risk communication:** Raise awareness about risks, consult with relevant stakeholders, and ensure that information about risks and risk management activities is shared appropriately.
- **Risk awareness:** Promote awareness about the importance of cloud risk management and the specific risks being addressed. Implement mechanisms for effectively disseminating information about risks and risk management activities to ensure all relevant stakeholders are informed.
- **Consulting:** Engage with stakeholders to gather insights and ensure that diverse perspectives are considered in the risk management process.

3.1.3.2 Risk Assessment

This involves evaluating each risk to determine the likelihood of occurrence and the severity of its consequences.

- **Identification of risks:** Systematically identify potential risks associated with cloud services, including security breaches, data loss, and compliance violations.
- **Analysis of relevant risks:** Analyze identified risks to understand their nature, causes, and potential impact.
- **Evaluation of risks:** Evaluate the likelihood and impact of each risk to prioritize them based on potential effect to the objectives.

3.1.3.3 Risk Treatment

After assessing risks, develop and approve an action plan to mitigate, transfer, avoid, or accept each risk. Implement these action plans and identify any remaining (residual) risks:

- **Identification of options:** Explore different risk management strategies, such as mitigation, transfer, avoidance, or acceptance.
- **Development of action plan:** Formulate specific actions to mitigate or address each risk, considering resources and risk appetite.
- **Approval of action plan:** Ensure that the action plan is reviewed and approved by appropriate decision-makers and stakeholders.
- **Implementation of action plan:** Carry out the approved actions to manage identified risks within an acceptable time frame.
- **Identification of residual risks:** Evaluate and document the remaining risks after treatment actions have been implemented.

- **Risk acceptance:** Accept residual risk if it falls within risk appetite, and if the cost to further mitigate it is greater than its impact.
    - The decision to accept the risks should follow a cost-benefit analysis made by the business owner, after fully understanding the implications and possible consequences.
    - The residual risk, analysis and acceptance should be clearly documented.
    - Accepted risk should be periodically re-evaluated to recognize any changes in the risk profile, risk appetite, or available cost-effective mitigations.

3.1.3.4 Interface to Other Operational & Product Processes

Risk management should not be siloed; it should interface with other business processes to ensure that risk considerations are embedded throughout the CSC's operations and product lifecycle.

3.1.3.5 Monitoring & Review (Plans, Events, Quality)

Continuously monitor risk management plans, events, and the quality of risk management activities. Regular reviews ensure that risk management processes remain effective and adapt to any changes in the business environment.

- Continuously monitor the cloud environment, risk management plans, and effectiveness of implemented controls.
- Conduct periodic reviews of the risk management process to ensure it remains relevant and effective in addressing the evolving risk landscape and organizational changes.
- Implement relevant metrics to assess the effectiveness and efficiency of implemented risk treatments, such as Key Control Indicators[52].
- Assess the residual risk to ensure that it still falls under an acceptable level. Key Risk Indicators[53] may assist with timely monitoring of the cloud risk posture.

By following the ENISA *Risk Management Process*, CSCs can establish a robust framework for managing cloud risks, integrated with their overall risk management strategy and operational practices. This approach helps mitigate specific cloud-related risks and enhances the organization's resilience and agility in navigating the complexities of the cloud computing environment.

3.1.4 Assessing Cloud Services

One of the first steps in managing cloud risks is having a systemic process to assess CSPs and their

[52] Key control indicators are metrics that provide information about the extent a risk control is meeting its intended objectives.
[53] Key risk indicators are metrics that provide early signals of increasing risk exposures in various areas of the enterprise. They are also known as emergent risks.

service offerings. This assessment should align with business needs and risk tolerance. The challenge in assessing CSPs is that a CSC will rarely gain visibility into all CSP internal operations and technology. CSPs are constantly changing their services; in some cases, they may add major service changes every week. The CSC may also lack the ability to compensate with custom SLAs and contracts.

The following process is designed to account for these differences:

- Business requests
- Review CSP documentation
- Review external sources
- Map to compliance requirements
- Map to data classification
- Define required and compensating controls
- Approval process

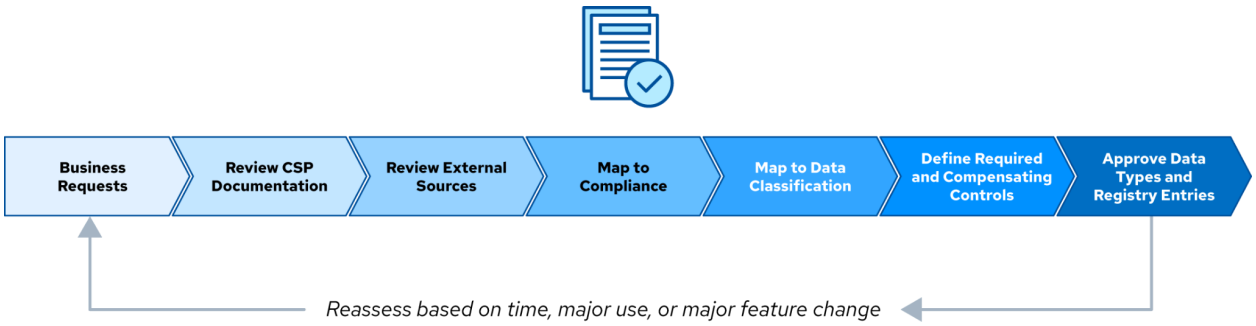


Figure 17: Systematic Process for Evaluating and Approving Cloud Services

3.1.4.1 Business Requests

A CSC's business unit requests the use of a CSP and service (e.g., a PaaS service in a large provider like Azure). Understanding the business requirements, including the specific CSPs and services needed, is the first step in assessing the risk of adopting a particular cloud solution.

- **Understanding business requirements:** Gather detailed requirements from the business unit and other stakeholders requesting the cloud service. This includes specific functionalities, performance expectations, and regulatory or data handling requirements.
- **Provider and service selection:** Evaluate potential CSPs and their services (e.g., PaaS, IaaS, SaaS) to determine which best fits the business requirements.

3.1.4.2 Review Cloud Service Provider Documentation

Scrutinize the CSP documentation thoroughly. Ensures this includes a CSA Consensus Assessments Initiative Questionnaire[54] (CAIQ) for security specifics, any certifications the CSP holds, detailed security

[54] CSA. (2024) Cloud Controls Matrix and CAIQ v4

and privacy policies, and the Terms of Service (ToS).

- **CAIQ and certifications:** The CAIQ provides a comprehensive set of questions that CSPs answer to disclose their security controls. CSP certifications (e.g., ISO/IEC 27001, SOC 2) offer third-party validation of their security practices.
- **Security and privacy documentation:** Review the CSP's published security policies, privacy policies, and data handling practices to ensure they align with relevant standards.
- **Service level agreements (SLA) and contracts:** SLAs outline the performance and uptime commitments of the CSP, while contracts detail the terms of service, including responsibilities and liabilities.
- **ToS:** Understanding the ToS is important to avoid legal or operational surprises post-adoption. These may be the only legal contracts between the CSC and CSP.

3.1.4.3 Review External Sources

In addition to reviewing the CSP documentation, a CSC might consider using tools like Cloud Access Security Brokers (CASB), Cloud Security Posture Management (CSPM), Cloud Workload Protection Platform (CWPP) and SaaS Security Posture Management (SSPM) solutions for independent assessments. CSCs should conduct additional research for reviews, vulnerabilities, and security incidents related to the CSP. This is also often a good time to review findings with the provider, although this may be limited to larger projects or smaller CSPs.

- **Tools:** Utilize tools for continuous third-party risk monitoring.
- **Research:** Investigate external reviews, reported vulnerabilities, and any past security incidents involving the CSP to gauge its security posture and response capabilities.
- **Provider interactions:** For significant projects, consider engaging directly with the CSP to discuss findings and any concerns. This can provide clarity and reassurance.

3.1.4.4 Map to Compliance Requirements

When selecting a CSP, it is essential to align their features and policies with the organization's compliance needs, such as General Data Protection Regulation (GDPR), Health Insurance Portability and Accountability Act (HIPAA), or Payment Card Industry Data Security Standard (PCI DSS). This ensures that regulatory requirements are met, and that data remains secure. Most CSPs publish detailed compliance documentation to demonstrate their adherence to various standards and regulations.

3.1.4.5 Map to Data Classification

Not all data needs the same risk management process. CSPs and services should be approved based on data types, which allows flexibility to accommodate for varying CSC's needs and requirements. For

example, it might be acceptable to use a more risky service with less valuable or public data.

- **Data sensitivity assessment:** Assess the sensitivity of data in transit and at rest. Not all data carries the same risk; thus, not all cloud services are required to meet the highest security standards.
- **Service approval based on data type:** Approve CSPs and their services based on the classification of data they will handle. This approach allows for flexibility and efficient use of resources.

3.1.4.6 Define Required & Compensating Controls

Before final approval, it is important to select and document the required controls (e.g., configuration settings within the CSP) and any compensating controls (e.g., third-party tools) that offer the appropriate level of security to mitigate the cloud risk.

3.1.4.7 Approval Process

Based on the gathered information and mapping, decide whether the CSP's services are appropriate for the intended data types. If so, approve its use and incorporate it into the cloud service registry.

3.1.5 The Cloud Register

A *cloud register* is a central repository of approved CSPs and services, and what types of data they are approved to handle at a given level of risk. This guides internal decisions on which providers and services to use for which projects. It also helps ensure that data is only used with compliant providers.

3.1.5.1 Understanding the Cloud Register

A cloud register is a strategic tool that catalogs all cloud services used by a CSC, along with information about the types of data it handles, the assessed level of risk, and when the risk assessment should be reviewed. It distinguishes between different data types (e.g., public, sensitive, personally identifiable information) and assigns risk levels to guide the use of cloud services.

3.1.5.2 Components of a Cloud Register

A cloud register[55] includes columns for the CSPs of the service, the specific service itself, the types of data allowed to be processed by it, the risk level (e.g., critical, high, moderate, low), the expiration date, and other key attributes for when the risk assessment must be re-evaluated. Attributes may include name, description, owner, expected/actual frequency, potential/actual magnitude, potential/actual business impact, and disposition.

[55] Aligned with the requirements included in the definition of risk register in CSA's *Certificate of Cloud Computing Audit Knowledge* (CCAK), create a repository of the key attributes of potential and known IT risk issues.

3.1.5.3 Purpose a Cloud Register

A cloud register streamlines the decision-making process used during audits to assess if the data and operations of a CSP meet applicable requirements. Functioning as a centralized database, it lists approved cloud services, categorizing them based on the types of data the CSP is approved to handle. This not only ensures that teams are aligned with organizational standards and compliance requirements, but also minimizes the duplication of efforts in assessing CSPs. By consulting the registry, teams can quickly identify which services are available for their specific data management requirements, thereby accelerating project initiation and reducing administrative overhead.

3.1.5.4 Risk Levels & Expiration

Inherent risk and residual risk are normally represented in risk levels such as critical, high, moderate, and low. The risk is derived based on impact and likelihood. Risk levels dictate the frequency and intensity of reviews and audits. For example, services handling personal identifiable information (PII)[56] and classified as critical risk, might be reviewed more frequently than services classified as moderate risk.

Provider	Service	Data Types	Risk	Expiration
ABC	Object storage	Public, sensitive	Low	Annual
ABC	Virtual networks	All	Low	Annual
GHI	CRM SaaS	PII	Moderate	Quarterly

Table 2: Cloud Registry Example

This fictitious example shows three specific services from two providers and lists the data types that are allowed to be processed. Based on that, risk is assigned, and the required review frequency. This allows teams to accelerate risk assessment.

3.1.6 Risk Assessments, Threat Intelligence & Modeling

Accompanying risk assessments with threat intelligence and modeling improves a CSC's cybersecurity posture, especially in the context of cloud computing. This requires continuous effort and updating, as both the threat landscape and cloud technologies evolve. Organizations can better prepare for and mitigate the risks associated with cloud computing by leveraging frameworks like those provided by the CSA and MITRE and conducting thorough threat modeling as part of the SSDLC. This proactive approach to cybersecurity ensures that cloud deployments are efficient, effective, and secure.

[56] NIST (2015) PII stands for Personally Identifiable Information; Any representation of information that permits the identity of an individual to whom the information applies to be reasonably inferred by either direct or indirect means.

- **Risk vs. threat:**
    - Risks are broad categories that refer to potentially negative outcomes, whereas threats represent opportunities for adversaries. Risks incorporate the potential for loss, damage, or destruction of assets due to a threat actor exploiting a vulnerability or other security defect.
    - Threats are more specific, representing a subset of risk that directly pertains to adversarial attacks aiming to exploit those vulnerabilities.
- **Current threat landscape:**
    - It is important to stay informed about the latest threats in cloud security, as the threat landscape is dynamic and constantly evolving. Understanding the actual threats a CSP and its services face is key to effectively assessing and mitigating risks in cloud deployments.
- **Threat modeling:**
    - Threat modeling is an integral part of the SSDLC. It is a structured approach whereby potential threats, such as structural vulnerabilities or privacy gaps, are identified and mitigated in the context of protecting a specific application or system.
- **CSA Top Threats:**
    - The CSA *Top Threats*[57] is frequently updated to identify the predominant cloud threats, and includes examples of specific techniques and public breaches.
- **Additional threat intelligence sources:**
    - Other valuable resources for threat intelligence include:
        - **MITRE ATT&CK®**[58] framework, which provides a comprehensive matrix of tactics. There is a cloud-specific matrix at
        - **CSA Research**[59]**:** created by the industry for the industry and is both vendor-neutral and consensus driven.

3.2 Compliance & Audit

Compliance and audit are essential to ensuring that information systems adhere to established standards, laws, regulations, and policies to protect data integrity, availability, and confidentiality. These processes are designed to identify vulnerabilities, assess risks, and implement controls to mitigate threats to information assets.

[57] CSA. (2023) Top Threats to Cloud Computing: Pandemic 11 Deep Dive
[58] MITRE. (2024) Cloud Matrix
[59] CSA. (2023) Understanding Cloud Attack Vectors

Compliance involves adhering to a set of predefined standards or regulations that govern security practices. Compliance ensures that organizations implement a prescribed set of security measures to protect sensitive information and systems.

An audit is an independent examination of records, operations, processes, and controls to ensure compliance with security policies, standards, and regulations. Audits help identify gaps in security measures and verify the effectiveness of implemented controls. Audits can be internal, meaning they are conducted by the CSC's own audit staff, or external, meaning they are performed by independent third parties. Regular audits ensure compliance, promote improved security posture, and build trust with customers and partners.

3.2.1 Types of Compliance & Cloud Impact

Compliance in the cloud environment is multifaceted, encompassing legal and regulatory requirements, adherence to international, national, regional and industry standards, and alignment with internal policies and standards. The dynamic, distributed and scalable nature of cloud computing introduces unique challenges and concerns for compliance. Below we delve deeper into these aspects.

3.2.1.1 Compliance Requirements

CSCs must balance leveraging the benefits of cloud services with maintaining compliance to protect their assets and meet their legal and regulatory obligations.

The following are some examples of contractual requirements, standards, and internal policies that impact compliance requirements:

- **Legal, regulatory, and contractual requirements:**
    - Cloud services often operate across multiple jurisdictions, which may have varying legal and regulatory requirements. Keeping track of these changes is essential for maintaining compliance.
    - Compliance inheritance refers to using cloud services that have already met certain compliance standards.
    - Continuously monitor for legal and regulatory requirements changes across jurisdictions where data is stored or processed. Implement a compliance inheritance strategy wisely, leveraging CSP certifications while ensuring end-to-end compliance.
    - Adhere to legal and regulatory frameworks governing cross-border data transfers, which are critical for operations spanning multiple countries. This requires comprehensive compliance with international and regional data protection regulations.

- **International, national, and industry standards:**
    - Gathering evidence to prove compliance can be performed manually or through automation. Cloud environments often facilitate automation, which can be more efficient but requires proper tooling.
    - A significant challenge is that many established standards have not been updated to reflect the specific needs and characteristics of cloud computing, potentially leaving gaps in compliance.
    - Invest in automation tools for compliance evidence gathering to enhance efficiency and accuracy. Actively participate in industry forums or regulatory bodies to advocate for and contribute to developing updated standards that reflect the realities of cloud computing.
- **Internal policies and standards:**
    - CSCs may find that their existing internal standards and controls are not entirely suitable for cloud environments. Adapting these standards to the cloud is necessary to ensure comprehensive governance.
    - Conduct a gap analysis to identify discrepancies between existing internal standards and cloud environments. Develop cloud-specific guidelines or adapt existing policies to cover cloud-specific issues, such as data residency, sovereignty and localization, access controls, and incident response in a cloud context.

3.2.2 Cloud-Relevant Laws & Regulations Examples

Many laws and regulations exist to protect various data types like personal information, financial data, and critical national infrastructure technology. There are a myriad of regulations to navigate, and each CSC has the obligation to understand its own specific set of legal and regulatory requirements. The following are some representative examples of regulations and industry standards that commonly affect cloud security and compliance.

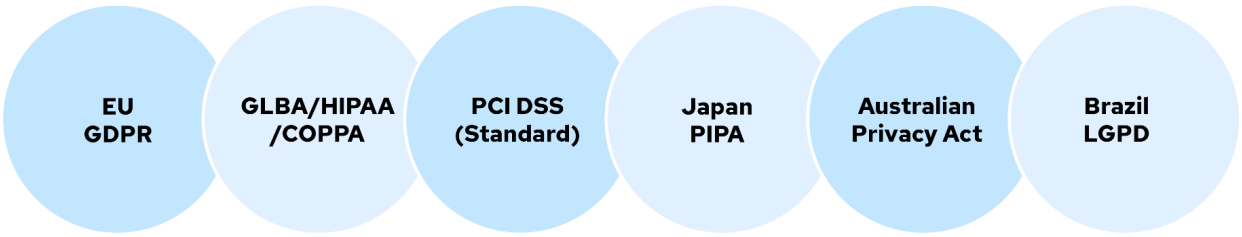


Figure 18: Key Privacy and Security Regulations Affecting Cloud Services

3.2.2.1 Privacy Laws & Regulation

- **EU GDPR:** Sets a high standard for data protection, emphasizing individuals' rights over their personal data, requiring consent for data processing, and imposing strict penalties for non-compliance.
- **US Regulations (CCPA/COPPA):** Focuses on specific sectors, protecting
    - Children's Online Privacy Protection Act (COPPA)
    - California Consumer Privacy Act (CCPA), and other state-level acts like it with detailed requirements for handling and safeguarding data.
- **Brazil LGPD:** Stands for General Personal Data Protection Law, strongly based on the EU GDPR. Like EU law, it also sets a high standard for data protection, emphasizing individuals' rights over their data, requiring consent for data collecting and processing, and imposing strict penalties for violations.
- **Japan Act on the Protection of Personal Information, Australian Privacy ACT:** National laws that regulate the collection, use, and disclosure of personal information, focusing on user consent, data accuracy, and cross-border data flow restrictions.

3.2.2.2 Other Relevant Laws & Regulation

- **US Regulations:**
    - Gramm-Leach-Bliley Act (GLBA), which imposes requirements on financial institutions in the United States to protect consumer information.
    - Health Insurance Portability and Accountability Act (HIPAA) safeguards medical privacy by establishing regulations on how healthcare providers, insurers, and others who handle data can use and disclose personal health information.
- **EU Laws and Regulations:**
    - EU Digital Operational Resilience Act (DORA) ensures operational resilience for critical financial market infrastructures operating in public cloud platforms.
    - EU AI Act establishes essential regulations to ensure the trustworthiness of Artificial Intelligence (AI) systems.
    - NIS 2, the recently enforced update to the Network and Information Systems Directive, strengthens cybersecurity measures for critical services across the EU.
    - EU Cybersecurity Act, currently under proposal, aims to fortify the digital defenses of EU institutions themselves.
    - EBA Guidelines on outsourcing arrangements by the European Banking Authority

- **Cybersecurity Law of the People's Republic of China:** Focuses on protecting the country's online infrastructure and data by outlining security obligations for companies, promoting public awareness of cyber threats, and granting authorities broad powers to monitor and regulate cyberspace.
- **PCI DSS:** A cross-jurisdictional standard for organizations that handle and process credit cardholder information, emphasizing financial data protection through comprehensive security measures.

3.2.2.3 Compliance in the Cloud

In general, some of the factors that are common across several laws and regulations are:

- **Secure handling:** Ensuring that access to sensitive data is tightly controlled and that data is processed to maintain its confidentiality and integrity.
- **Secure storage:** Implementing encryption and other protective measures to safeguard data at rest and in transit, ensuring proper data retention and deletion practices.
- **Due care:** Adhering to industry best practices and security standards to protect data from threats and vulnerabilities.
- **Audit trails:** Maintaining comprehensive records of data processing activities to demonstrate compliance with regulatory requirements and facilitate audits.

3.2.2.4 Adherence to Regulations & Standards

Cloud providers often achieve conformity against a variety of regulations, industry and national standards via certifications, attestation, and other forms of authorization. These include:

- ISO/IEC 27001-2022
- ISO/IEC 27017 Information security controls for cloud computing services
- ISO/IEC 27018 - Protecting PII in the Public Clouds
- Payment Card Industry Data Security Standard (PCI DSS)
- American Institute of Certified Public Accountants (AICPA)
- Service Organization Control Reports (SOC 1 and SOC 2)
- Cloud Security Alliance's STAR Certification
- Cloud Security Alliance's STAR Attestation
- U.S. Federal Risk and Authorization Management Program (FedRAMP)
- Singapore Multi-Tier Cloud Security standard
- Germany Cloud Computing Compliance Criteria Catalog (C5)
- Health Information Trust Alliance (HITRUST) Common Security Framework (CSF)
- EU Cloud Code of Conduct for GDPR

These certifications, attestations and authorizations are critical in demonstrating a CSP’s commitment to maintaining high standards of security and data protection.

3.2.3 Compliance Inheritance

Compliance inheritance represents a significant aspect of regulatory adherence within cloud computing, offering CSCs a way to leverage the security and compliance posture of their CSPs to meet various regulatory and industry standards. This concept is particularly relevant in environments where stringent data protection and security standards are mandated, such as financial services, healthcare, and sectors dealing with sensitive personal data.

Cloud compliance typically follows a shared responsibility model, where the CSP and the CSC are each responsible for certain aspects of compliance. Compliance inheritance aims to relieve some of the burden from the CSC, by allowing them to acquire a control set from a compliant CSP. Consider a cloud infrastructure provider that is PCI DSS-compliant. A CSC using their infrastructure services will inherit this set of controls and will be PCI DSS-compliant at the infrastructure level. The CSC, however, will be additionally responsible for ensuring that the software built on this infrastructure is also PCI DSS compliant.

Both the CSP and CSC are audited independently, and each must ensure that their respective controls are compliant.

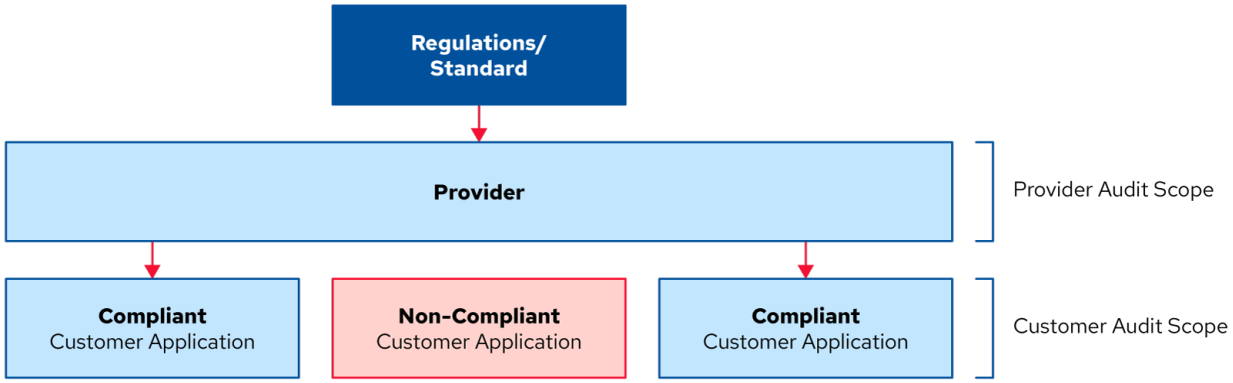


Figure 19: Audit Scope: Provider vs. Customer Responsibilities

3.2.3.1 Compliance Inheritance Limitations

Compliance inheritance generally does not shift the compliance responsibilities. It merely allows a CSC to leverage the activities and controls deployed by a CSP. When a CSC elects to rely on a CSP for whole or partial coverage of the CSC's obligations, it nevertheless retains the responsibility for compliance across the entire technology stack. In some standards and frameworks such as ISO 27001 or SOC 2, a CSC must take explicit steps at least annually to validate that its CSP continues to be compliant.

- **Provider audit scope:** In the model of Compliance inheritance, a CSP's infrastructure and services undergo rigorous audits to ensure adherence to specific compliance standards. These audits, often referred to as pass-through audits, validate the CSP's compliance, absolving CSCs from the need to audit the CSP's infrastructure and services themselves. The CSP assumes responsibility for the costs and ongoing maintenance of these certifications.

- **Customer audit scope:** While CSPs may offer a compliant infrastructure, the responsibility of building and maintaining compliant applications on this infrastructure lies with the CSC. This delineation of responsibilities means that, although the CSP's services may be compliant, any application or service that CSCs develop atop this infrastructure must also be independently assessed for compliance.
    - **Compliant Customer Application:** If a CSC constructs its service on a CSP's platform that is compliant with a particular standard or regulation (e.g., PCI DSS), the underlying infrastructure and operations provided by the CSP are considered compliant and are not within the CSC'S audit scope. This allows CSCs to focus their compliance efforts on their own applications and data management practices.
    - **Non-Compliant Customer Application:** Even when utilizing a compliant cloud infrastructure, CSCs can fail to meet regulatory requirements if their applications are not properly designed to comply with the relevant standards. This highlights the critical nature of the CSC's responsibilities in ensuring that its applications, processes, and data-handling practices adhere to compliance requirements.

3.2.4 Jurisdictions

Many cloud deployments may span different legal and regulatory jurisdictions. The complexity of compliance becomes magnified when operations extend across multiple regions, each with its own legal and regulatory frameworks governing data privacy, security, and other critical factors.

CSPss and CSCs operating in multiple regions will face a matrix of jurisdictions where various laws and regulations apply. This is affected by:

- The location of the CSP.
- The location of the CSC.
- The location of the data subject.
- The location where the data is stored.
- The legal jurisdiction of the CSP/CSC contract, which may be different than the locations of any stakeholders.
- Any treaties or other legal frameworks between those various locations.

An example could be the requirement to issue a breach notification in the country a CSP is operating in, even if the data was hosted in a different region.

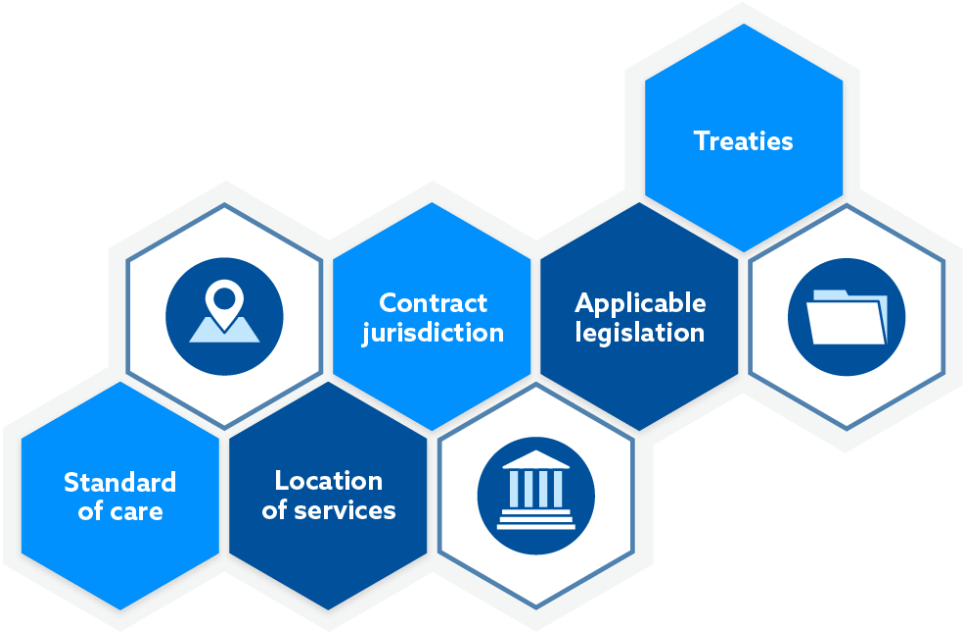


Figure 20: Factors Influencing Cloud Jurisdiction Compliance

3.2.5 Cloud Assurance Mechanisms

Assurances are the processes and methods used to validate compliance. Assurance encompasses a range of auditing, attestations, and evaluations, each with distinct focuses and methodologies that may vary significantly across CSPs. These processes verify regulatory, security, and operational standards compliance.

Term	Definition	Purpose	Scope	Level of Assurance	Focus	Example
Audit	Systematic examination of IT systems, processes, and controls	Provide reasonable assurance that IT controls are effective and secure	Focus on IT systems, security, and compliance	High. Rigorous investigation and verification of IT controls	Providing an independent opinion on IT compliance with criteria	IT security audits assessing network vulnerabilities and access controls
Attestation	Review of IT practices with a statement about posture.	Evaluate accuracy against a stated purpose, internal control, or system	Can include various IT subject matters beyond financial statements	Moderate. Provides a lighter version of audits for IT practices	Verifying specific IT controls or information against agreed-upon procedures	SOC 2 reports evaluating IT controls and issuing an attestation statement

Assurance	Unbiased evaluation of IT information to build confidence.	Enhance trust in IT processes and systems	Not limited to specific information; can cover various IT aspects	Varies based on the type of IT engagement	Building confidence in the reliability and security of an organization's IT infrastructure	Increased confidence in IT DR plan effectiveness based on a combined evaluation and audit
Evaluation	Assessment of IT performance, effectiveness, or outcomes against criteria	Measure success, identify areas for improvement, and inform IT decision making	Can apply to IT programs, projects, processes, or systems	Context dependent; may not always provide assurance in IT contexts	Assessing value, effectiveness, or outcomes of IT aspects	Evaluating the effectiveness of an IT DR plan or software development process
Assessment	Comprehensive analysis or evaluation of IT processes, risks, or performance	Thoroughly assess and analyze IT practices, risks, and compliance	Encompasses a wide range of IT-related evaluations, including risk assessments and compliance checks	Depends on the specific assessment objectives and methodology	Conducting a risk assessment for cloud migration strategy	Assessing the overall maturity IT governance framework

Table 3: Cloud Assurance Mechanisms: Definitions and Purposes

3.2.5.1 Third-Party Provider & Auditing

Some CSCs may be used to auditing third-party providers, but the nature of cloud computing and contracts with CSPs will often preclude things like on-premises audits. CSCs should understand that CSPs can (and often should) consider on-premises audits a security risk when providing multi-tenant services. Multiple on-premises audits from large numbers of customers present clear logistical and security challenges, especially when the provider relies on shared assets to create the resource pools.

Customers working with these providers will have to rely more on third-party attestations rather than audits they perform themselves. Depending on the audit standard, actual results may only be releasable under a nondisclosure agreement (NDA), meaning CSPs must enter into a legal agreement before gaining access to attestations for risk assessments or other evaluative purposes. This is often due to legal or contractual requirements with the audit firm, not due to any attempts and obfuscation by the CSP.

CSPss should understand that customers still need assurance that the it meets their contractual and regulatory obligations and should thus provide rigorous third-party attestations to prove it meets its obligations, especially when the CSP does not allow direct customer assessments. These should be based on industry standards, with clearly defined scopes and a list of specific controls evaluated. Publishing certifications and attestations (to the degree legally allowed) will greatly assist cloud customers in evaluating providers. The CSA STAR registry[60] offers a central repository for providers to release these documents publicly.

[60] CSA. (2023) CSA Star Registry

3.2.6 Artifacts of Compliance

Compliance artifacts are the logs, documentation, and other materials needed to support compliance activities. Both CSPs and CSCs have responsibilities for producing and managing their respective artifacts. CSCs are ultimately responsible for the artifacts needed to support their audits, and thus need to know what the CSP offers, so they can create artifacts to cover any gaps (e.g., building more robust logging into an application since server logs on PaaS may not be available).

Compliance artifacts demonstrate adherence to various regulatory and security standards within cloud environments. These artifacts serve as tangible evidence during audits, showcasing an organization's ability to manage and secure data effectively.

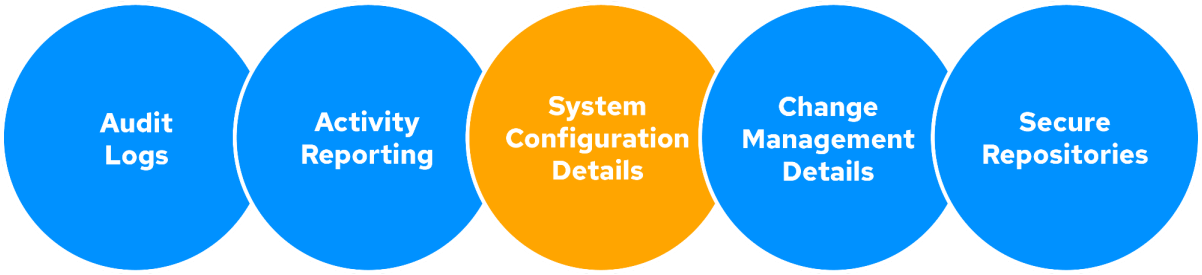


Figure 21: Essential Compliance Artifacts in Cloud Environments

The following are examples of compliance artifacts:

- **Audit Logs:** Detailed records of events, actions, and changes
- **Activity reporting:** Reports summarizing user activities, access patterns, and system interactions. Activity reports can help identify unauthorized access, track user actions, and ensure that operational practices align with compliance requirements
- **System configuration details:** Documentation of system configurations, including network settings, access controls, and security measures
- **Change management details:** Records of changes made to the system, including updates, modifications, and patches. These details are critical for ensuring that changes are authorized, tested, and implemented in a manner that maintains the integrity and security of the environment
- **Secure repositories:** Storing compliance artifacts requires secure, accessible repositories that protect the integrity and confidentiality of the data. These repositories should adhere to security standards and be capable of restricting access to authorized personnel only. In some cases, managing these repositories may cross CSC/CSP boundaries, necessitating clear agreements on access controls and data protection measures

3.3 Governance, Risk, and Compliance: Tools & Technologies

Technical and non-technical tools alike are in the GRC toolkit. Included is clear documentation of responsibilities, contracts, and repositories that store maintained risk registers and service registries. Also included is documentation describing frameworks and processes that have been adapted for their business context and adoption process for teams across the CSC. There are also a wide variety of technical tools that are used to automate tasks that would be too labor-intensive for humans to do using manual processes. The following provides the different types of tools a CSC is likely to encounter and a brief description for each.

3.3.1 Non-Technology Tools Supporting Governance, Assurance, & Compliance

Non-technical tools play a role in the governance, assurance, and compliance aspects of cloud computing, providing organizations with the frameworks and methodologies needed to manage risks, clarify responsibilities, and ensure compliance with regulatory requirements.

The following are examples of non-technical tools:

- **Shared responsibility model:**
    - As previously discussed, a shared responsibilities model delineates the security responsibilities shared between the CSP and the CSC. It is fundamental in cloud environments to understand the division of tasks related to security management, data protection, and compliance obligations. RACI charts (responsible, accountable, consulted, and informed) may be used to document roles and responsibilities.
    - It helps prevent gaps in security coverage by clearly defining who is responsible for specific security tasks, such as data encryption, network security controls, and incident response.
- **Contracts:**
    - Contracts specify the legal roles, responsibilities, and expectations in the relationship between the cloud provider and the customer. They include SLAs, data processing agreements, and confidentiality clauses.
    - They ensure both parties understand their obligations, legal rights, and remedies, thereby laying the groundwork for accountability and recourse in the event of disputes or compliance failures.

- **Risk register:**
    - A risk register is a document that lists all identified risks, their severity, and actions for their management.
    - It facilitates proactive risk management by enabling organizations to prioritize and address risks systematically, ensuring that mitigation efforts align with the organization's risk appetite.
- **Third-party risk management/provider/cloud register:**
    - Keeping a register of third-party providers and assessing their risks ensures that all external entities are evaluated and monitored for potential risk and security implications,
- **Cloud security maturity model (CSMM)**
    - It guides an organization in measuring and improving its overall cloud security program based on industry benchmarks.
    - It provides a roadmap for continuous cloud security improvement, helping organizations identify weaknesses and implement stronger security measures over time.
- **Service registry:**
    - A service registry is a database containing information about deployed internal services. It is used in cloud services to manage and discover dynamic services that teams may integrate with or use.
    - It enhances service interoperability and efficiency by allowing teams to easily locate and integrate with existing services, thereby reducing redundancy, and promoting reuse.
- **Control framework:**
    - This consists of a set of guidelines or best practices used to manage risks and implement effective controls within the organization.
    - It standardizes risk management and control implementation across the organization, ensuring consistent and effective governance practices.
- **Monitoring and auditing framework:**
    - These frameworks are for continuous oversight, allowing for the detection of anomalies, security incidents, and ensuring that controls are working as intended.
    - They enhance the ability to promptly identify and respond to potential security threats, ensuring compliance with regulatory and internal policies.
- **Data/asset classification and catalog:**

- A systematic categorization of data and assets helps apply appropriate security controls based on sensitivity and importance.
- It protects critical information and resources by ensuring that higher levels of security are applied to more sensitive assets, aligning with compliance and data protection requirements.

- **User/entities mapping:**
  - Mapping users and entities to their corresponding data access and processing activities helps maintain data governance and prevent unauthorized access.
  - It strengthens data governance by ensuring only authorized individuals can access specific data sets, minimizing the risk of data breaches and compliance violations.

3.3.2 Technologies Supporting Governance, Assurance, & Compliance

A range of technologies are used to assist us with governance, assurance, and compliance in the cloud. These technologies facilitate the enforcement of security policies, enable real-time monitoring for compliance, and automate the management of cloud resources to minimize risks associated with human error and misconfigurations.

3.3.2.1 Cloud Service Provider Policies

CSP policies are sets of preventative and technical rules integrated within the cloud platform to control and manage access, operations, and configurations.

Crafting and enforcing CSP policies requires a deep understanding of the organization's security objectives and the cloud environment's capabilities[61]. CSP policies should be aligned with best practices and regulatory requirements to ensure robust security governance.

3.3.2.2 Detective & Preventative Controls

Tools like cloud Security Information Event Management (SIEM), Cloud Security Posture Management (CSPM), cloud-native application protection platform (CNAPP), cloud workload protection platform (CWPP), and Security Service Edge (SSE) provide monitoring and management capabilities for deviations from security and compliance baselines. These tools can automate the detection of misconfigurations, vulnerabilities, and non-compliance with regulatory standards.[62]

[61] See *Domain 4: Organization Management* for a deeper dive into how CSP policies can be crafted and enforced to align with organizational security objectives.
[62] See *Domain 4: Organization Management* for more context on these tools.

3.3.2.3 Software Bill of Materials

SBOM is a comprehensive inventory of all components that make up a software application, including its open-source components. It is crucial for transparency and tracking software dependencies to ensure a secure and compliant software supply chain. Implementing an SBOM allows organizations to track and verify the components in their software, ensuring that they are up-to-date and do not contain known vulnerabilities.

3.3.2.4 Automation

Cloud deployments are often defined and deployed using automation, such as standard images and IaC. These enhance consistency and auditability[63].

Summary

Managing risk, audit, and compliance in cloud environments is crucial for ensuring the security and integrity of both CSPs and CSCs. This domain focuses on establishing robust risk management frameworks, maintaining compliance with regulatory standards, and leveraging various tools and technologies for effective governance.

Cloud risk management involves understanding and mitigating risks associated with cloud services. Key practices include evaluating CSPs, maintaining a cloud register, and implementing comprehensive risk management strategies. These strategies ensure that cloud risks are identified, assessed, and managed proactively.

Compliance and auditing are essential for adhering to standards, laws, and regulations. Compliance ensures that security measures meet regulatory requirements, while audits verify the effectiveness of these measures. The cloud's dynamic nature requires addressing legal, regulatory, and contractual challenges across multiple jurisdictions. Organizations must adapt their internal policies to remain compliant in both traditional and cloud environments.

Compliance inheritance leverages CSPs' compliance certifications to meet regulatory standards. This shared responsibility model allows CSCs to inherit compliance controls from CSPs while maintaining responsibility for their applications' compliance. This approach simplifies compliance management but requires continuous monitoring and validation.

GRC tools and technologies support the enforcement of security policies and compliance requirements. Non-technical tools like shared responsibility models and risk registers, along with technical tools like SIEM, CSPM, and SBOM, enhance security governance. Automation plays a key role in maintaining consistent and auditable cloud deployments.

[63] Auditability is covered in context throughout all of the relevant domains covered in the CCSK.

Artifacts of compliance such as audit logs, activity reports, and system configurations are vital for demonstrating adherence to regulatory standards. Secure repositories for these artifacts ensure their integrity and confidentiality, supporting effective compliance management.

In summary, securing cloud environments requires a comprehensive approach to risk management, compliance adherence, and continuous monitoring. By utilizing both non-technical and technical tools, organizations can effectively manage cloud risks, ensure compliance, and maintain the security and resilience of their cloud infrastructures.

Recommendations

Compliance, audit, and assurance should be continuous. They should not be seen as merely point-in-time activities, and many standards and regulations are moving more towards this model. This is especially true in cloud computing, where both the CSP and CSC are in constant flux.

Adopting a continuous approach to compliance, audit, and assurance in cloud computing is essential for navigating cloud services complexity and ensuring that CSPs and CSCs meet their regulatory and security obligations. By following these recommendations, CSPs and CSCs can foster a more secure and compliant cloud ecosystem, effectively mitigating risks and enhancing trust in cloud services.

Cloud Service Providers

- **Transparent communication:** Communicate their audit results, certifications, and attestations with particular attention to:
    - **Assessment scope:** Clearly define what aspects of the cloud service are assessed, including specific features and services.
    - **Coverage details:** Specify which services are covered in different locations and jurisdictions, helping CSCs understand where and how they can deploy compliant applications.
    - **Deployment guidelines:** Offer guidance on how CSCs can deploy applications and services in compliance with relevant standards and regulations.
    - **Customer responsibilities:** Highlight any additional responsibilities that customers need to be aware of, including any service limitations that might impact compliance.
- **Maintenance of certifications:** CSPs must maintain their certifications/attestations over time and proactively communicate any status changes.
- **Continuous compliance initiatives:** CSPs should engage in continuous compliance initiatives to avoid creating gaps, and thus exposures, for CSCs.
- **Provision of compliance artifacts:** Provide CSCs with commonly needed evidence and artifacts of compliance, such as logs of administrative activity the CSC cannot otherwise collect on their own.

Cloud Service Customers

- **Understanding compliance obligations**: CSCs should fully understand their compliance obligations before deploying, migrating, or developing in the cloud. This understanding is crucial for selecting suitable cloud services and configuring them in compliance with regulatory requirements.
- **Evaluating provider credentials:** Assess a CSP's third-party attestations and certifications against its compliance needs. Ensure that these credentials align with specific compliance obligations.
- **Scope and coverage of assessments:** Gain a clear understanding of the scope of the CSP's assessments and certifications, including the specific controls and services covered. This knowledge helps align the CSC's compliance strategy with the CSP's offerings.
- **Selecting experienced auditors:** When possible, choose auditors with expertise in cloud computing, especially for leveraging pass through audits and certifications to manage audit scope effectively.
- **Managing compliance artifacts:** Understand what compliance artifacts the provider offers and ensure efficient collection and management of these artifacts. Create and manage compliance artifacts when necessary to fill any gaps the CSP leaves.
- **Maintaining a Cloud Provider Register:** Keep an updated register of all CSP services utilized, noting relevant compliance requirements and the current compliance status of each service. Tools like the CSA CCM can aid in this activity, offering a structured approach to managing compliance across various cloud services.

Additional Guidance

- CSA's Perspective on Cloud Risk Management
- CSA Code of Conduct Gap Resolution and Annex 10 to the CSA Code of Conduct for GDPR Compliance
- Top Threats to Cloud Computing: Pandemic 11 Deep Dive | CSA
- Third-Party Vendor Risk Management in Healthcare | CSA
- Mitigating Hybrid Clouds Risks | CSA
- Enterprise Resource Planning and Cloud Adoption | CSA

Domain 4: Organization Management

Introduction

Organization Management refers to the overall management of a cloud environment, including organizing and validating the security assurance of Cloud Service Providers (CSPs) and securing individual cloud service deployments. These top-level security concerns span deployments and include the best ways to structure them for optimal "blast radius"[64] control and security management. Although the underlying technologies of each CSP are fundamentally different, they usually have enough feature parity to allow for consistent management practices.

Tenancy, a customer's allocation of resources in a multitenant environment, plays a crucial role in managing cloud environments. Establishing key controls to manage the hierarchy and address top-level security and compliance concerns is essential for maintaining visibility and structure.

As enterprises increasingly adopt multi-cloud strategies, understanding the hierarchical models used by major CSPs like AWS, Azure, and Google Cloud is important. This domain explores various organization hierarchy models, their capabilities, and best practices for managing and securing multiple cloud deployments. By examining structural differences and standardizing approaches, Cloud Service Customers (CSCs) can implement cohesive security controls and policies, enhancing their cloud management strategy and minimizing security risks.

This domain also addresses organization-level security management nuances, including identity provider mappings, CSP policies, shared services, and considerations for hybrid and multi-cloud environments. CSCs typically use multiple cloud services, including Software as a Service (SaaS), Infrastructure as a Service (IaaS), and Platform as a Service (PaaS). This, combined with hybrid connections and mergers and acquisitions (M&A), can lead to uncontrolled growth or so-called cloud sprawl, increasing costs and security risks.

The first steps in cloud security involve controlling non-essential sprawl, organizing the footprint, and implementing security controls that work across CSPs, as well as within each CSP, to secure individual deployments. This begins with understanding how cloud services can be divided into smaller units of control and establishing enterprise and tenancy-wide controls that live outside deployments.

[64] The potential amount of damage an incident can cause.

Learning Objectives

In this domain, you will learn to:

- Manage organization-level security within a provider.
- Leverage organization hierarchy for managing critical aspects of cloud deployments.
- Recognize security considerations for hybrid/multi-cloud deployments.
- Identify different cloud organization hierarchy models.

4.1 Organization Hierarchy Models

There are various models of organization hierarchy utilized in cloud environments, each having their own complexities of managing cloud resources across different CSPs. As CSCs expand their use of cloud technologies, understanding the structural differences and terminology used by major CSPs like AWS, Azure, and Google Cloud is crucial. This section aims to clarify these concepts and present a standardized approach to discussing and implementing organizational structures in the cloud. By comparing the hierarchical models of AWS, Azure, and Google Cloud, we provide insights into effectively applying security controls and policies across disparate cloud platforms, ensuring a unified and secure cloud management strategy.

4.1.1 Definitions

Navigating the topic of cloud organization constructs can be challenging, not just because of the objective complexity of a company's technological footprint (e.g., on-prem, cloud, OT, ICS), but also for a simple matter of vocabulary and terminology used by different CSPs for similar organizational structures. For instance, Amazon Web Services (AWS) uses terms such as Organization, Organization Units, and Accounts. In contrast, Microsoft Azure categorizes its structures into Tenant, Management Group, and Subscription. Google Cloud Platform (GCP) organizes its services into Organizations, Folders, and Projects.

Although these terminologies and their associated capabilities are not identical, they share enough similarities to allow the extraction of generalized security principles applicable across various CSPs. The hierarchical organization of these constructs primarily facilitates the consistent application of security controls and policies across multiple deployments.

For the sake of simplifying discussions and maintaining clarity, we will use a standardized set of terms:

- An "**organization**" denotes the top-most hierarchical structure within a CSP, equivalent to an Organization in AWS and GCP, or a Tenant in Azure.
- A "**group**" represents a collection of deployments, similar to an AWS Organization Unit, an Azure Management Group, or a GCP Folder.

- A "**deployment**" refers to an isolated environment within a CSP, analogous to an AWS Account, an Azure Subscription, or a GCP Project.

Below is a brief overview of the different terminology used by the major CSPs.

Cloud Service Provider	Organization	Group	Deployment
AWS	Organization	Organization Units	Accounts
GCP	Organizations	Folders	Projects
Microsoft Azure	Tenant	Resource Group	Subscription

Table 4: Cloud Service Provider Terminology Comparison

It is important to note that the exact terminology used may vary from one CSP to another. For example, the term "account" might typically be associated with identity and access management (IAM), "subscription" might refer to receiving periodic email updates, and "folder" could be understood as a place to store files.

The figure below illustrates the hierarchical structure for cloud resource management across different CSPs, helping to visualize the similarities and differences:

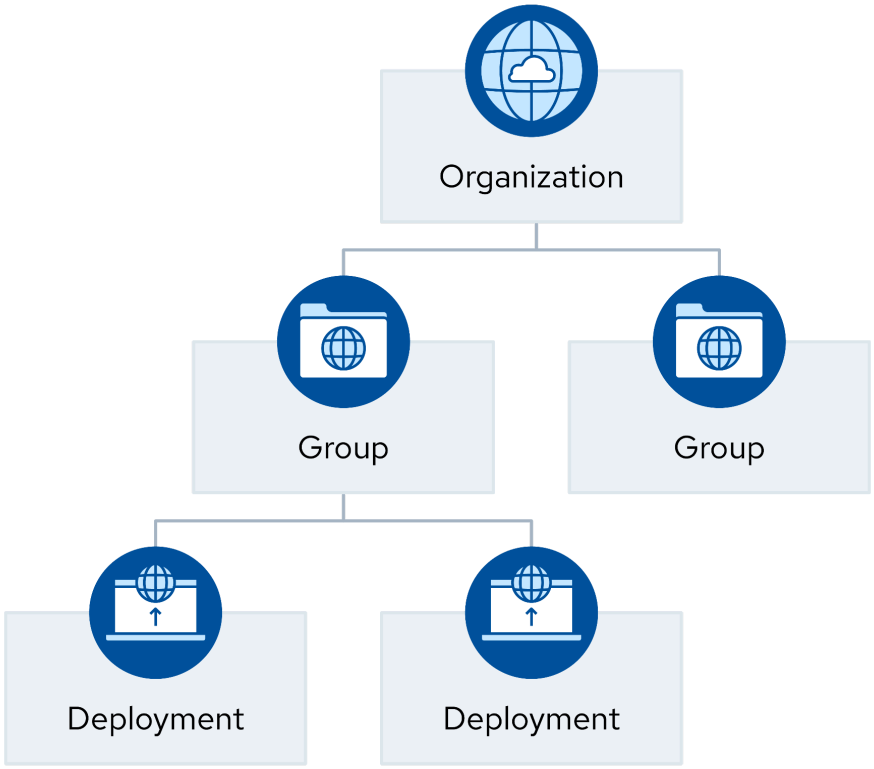


Figure 22: Hierarchical Structure for Cloud Resource Management

Utilizing multiple deployments is a strategic approach to reducing the impact of adverse events or breaches, adhering to service limits imposed by CSPs, and facilitating the logical separation of different technology stacks. This approach underscores the importance of adopting a structured and hierarchical model for organizing cloud resources, thereby enabling enhanced security, and streamlined management of resources across cloud environments.

4.1.2 Organization Security Objectives

Providing a consistent security approach for protecting an enterprise against malicious actors (that includes internal as well as external) is important to create defense-in-depth controls. Your objectives must be both technical and business-oriented. Your measurement of success involves reducing risk, improving governance and regulatory compliance, and aligning the organization's culture with the risk appetite of its leadership.

4.1.2.1 Organization

A well-structured deployment hierarchy is fundamental. It delineates the arrangement of resources and services within the cloud environment, facilitating efficient management and operational clarity.

Implementing a comprehensive tagging strategy and maintaining an accurate resource registry ensures that assets are easily identifiable, categorizable, and manageable, streamlining deployment and maintenance processes.

4.1.2.2 Visibility

Maintaining a clear view of all resource configurations is critical for security and operational efficiency. This visibility allows for the quick identification and correction of misconfigurations that could lead to vulnerabilities. Service configurations should be tracked to ensure that services are running with the correct settings and comply with security best practices. Monitoring activity across the cloud environment enables the detection of unusual or unauthorized actions that could signal a security threat.

4.1.2.3 Governance

Robust identity and access management (IAM) practices are essential to ensure that only authorized users have access to sensitive resources and can perform actions within their remit. Governance extends to configurations, which must be managed according to defined standards to avoid drift from security baselines.

4.1.2.4 Consistency

Centralized shared security services, such as identity providers and threat detection systems, provide uniform security coverage across the entire cloud landscape. "Account factories" enable the rapid and consistent creation of cloud accounts that adhere to organizational standards and security requirements,

ensuring consistency across deployments. Examples of account factory services include AWS Control Tower, Azure Blueprints, and Google Cloud Resource Manager.

4.1.3 Organization Capabilities Within a Cloud Service Provider

All IaaS and PaaS CSPs offer segmented and segregated customer environments, which is essential for implementing multi-tenancy. This ensures every CSC has their own secure and safe collection of resources allocated from the CSP's resource pool. CSCs quickly understood the benefits of having multiple independent deployments with a CSP to deploy different applications in each environment, effectively limiting the blast radius of any security issues. This is because each deployment was as isolated and segregated as those belonging to different CSCs, making each deployment a robust security barrier akin to utilizing multiple segregated data centers.

However, the challenge arose in organizing and managing these multiple deployments, a strategy that CSCs embraced even before the CSPs did. Over time, CSPs have increasingly introduced features for managing and securing multiple deployments, establishing this approach as a best practice.

There are four main capabilities that all major cloud service providers offer, which enable CSCs to significantly enhance security across their organization:

- Groups allow CSCs to structure their deployments into an isolation hierarchy.
- Policies are security rules that can apply to a group or a deployment. These typically enable and disable features, often down to specific API calls or even individual parameters.
- IAM centralization and/or federation supports centralized management of a CSC's users.
- Each CSP supports their own set of shared security services. These vary greatly, but support for central logging is nearly always available.

Some CSPs also introduce the concept of an "account factory" (aka "landing zone" or "account vending machine") This feature facilitates the creation of standardized deployments, usually utilizing Infrastructure as Code (IaC), equipped with a baseline of pre-configured security configurations and controls. Such an approach allows for quicker account provisioning, albeit with potentially less flexibility in individual policy adjustments. Through these methods, CSCs can enhance their security posture, while efficiently managing multiple deployments and maintaining a high level of security and operational efficiency.

4.1.4 Building a Hierarchy Within a Provider

When establishing the overall hierarchy of deployments within a cloud environment, CSCs must consider security and general management considerations, such as:

- **Will the hierarchy support effective use of policies?** Policies define and restrict the actions that can be performed within an account. Most CSCs support applying policies at the group or

deployment level. Applying them at the group level establishes security for the entire tree of nested groups and deployments in that branch.

- **Will the hierarchy support IAM requirements?** With most CSPs, a CSC can define user entitlements at the group level, not just within a deployment.
- **Is the hierarchy compatible with a CSC's structure in terms of business units, governance, and so on?** While not strictly a security factor, the deployment hierarchy is often closely coupled with the IAM hierarchy and can affect various resources such as billing and cost management.

CSCs typically adopt one of three models to define their hierarchy, each with its own advantages and operational implications. No single model is universally superior, and some CSCs may combine elements from different models to best reflect their operational realities.

- **Business Unit and Application-Based:** In this model, the cloud hierarchy is structured with business units at the top, followed by applications within these units, and then by environments (e.g., production vs. development). This arrangement aligns well with business-unit-focused IAM hierarchies, but it might be less efficient for policy management unless cloud features closely align with the business units and applications.
- **Environment-Based:** This model prioritizes environments—such as development, production, and testing—at the top of the hierarchy, followed by business units or applications. This approach is beneficial for policy management, allowing for the establishment of baseline security and operational policies for different environments. However, it may not align as neatly with IAM hierarchies or billing and cost management needs.
- **Geography-Based:** For CSCs that operate globally, the geography-based model is the structure of choice. It commences with geographic regions (e.g., EMEA, NA, or specific countries) at the top. Subsequently, it integrates business units or environments at subordinate levels. This structure is often advantageous for global CSCs facing diverse security and regulatory requirements specific to each region.

Most CSCs find maintaining a single organization per CSP optimal, but not essential. However, there are scenarios where supporting multiple organizations or tenants is necessary, such as meeting international regulatory or security requirements or large organizations that exceed a CSP's service limits for deployments within a single CSC. It is also common for hierarchies to include dedicated branches for Operations and Security, accommodating the shared services deployed by these teams across the infrastructure. This strategic organization of cloud deployments enables CSCs to manage security, compliance, and operational efficiency effectively.

When considering a Zero Trust Architecture, some elements of all three are necessary to accomplish either advanced or optimal maturity. Policy administration and enforcement points require dynamic consideration of factors from all three elements[65].

[65] Additional material on Zero Trust is provided in Domain 12: *Related Technologies & Strategies*. Context-specific material is also provided on Domain 2: *Cloud Governance & Strategies*, Domain 4: *Organization, Tenancy, & Enterprise Management*, Domain 5: *Identity and Access Management*, and Domain 7: *Infrastructure & Networking*.

4.2 Managing Organization-Level Security

One of the most impactful differences between cloud and traditional infrastructure is that, in the cloud, teams are typically able to create and manage their virtual environment, which is equivalent to an entire data center. There is no inherent need for a networking team, a server team, and the many other traditional silos we have relied on in traditional infrastructure due to working within a physical facility. Yes, all clouds still run in data centers, but the CSC rarely sees or interacts with the physical layer and creates entire networks and application stacks with web interfaces and API calls.

The goal of cloud security is to maintain acceptable risk without introducing friction that reduces or eliminates the benefits of cloud computing. It Is important to maintain control of the cloud footprint without impeding business objectives. CSPs offer a range of capabilities to support governance and security outside traditional security domains like network or application security. This starts with a well-defined tenant structure, which can be extended with additional controls.

4.2.1 Identity Provider & User/Group/Role Mappings

The Identity Provider (IdP) is a centralized system that manages users' identities and authentications. It's separate from individual cloud deployments, enabling a single identity to be used across different cloud services, including SaaS platforms. The IdP and user/group/role mappings are used to define deployment access. This is the IAM that lives outside the deployment, but there is additional IAM[66] within a deployment. In terms of Organization Management, there are two important factors to consider:

- Minimize access to the "root" of your organization. The objective is to have as few individuals as possible with high-level access that allows them to alter or access deployments down the hierarchy, or alter shared services with cascading effects or potential privilege escalations into deployments.
- Lock down who can create deployments and how, but support a low-friction process to make it easy for teams to obtain new accounts in accordance with policy. For example, set up an account factory that creates a properly configured account of a requested type (e.g., development, sandbox, production) in that team's branch of the hierarchy.

The IdP can be used across multiple CSPs and SaaS platforms, even if it's set up within just one CSP. The IdP defines user, group, and role mappings. These mappings are shared with the CSP during the federation process, where the CSP's IAM system assigns permissions based on these mappings. Additionally, attributes like business units can be used to fine-tune access controls. Depending on the CSP, these mappings can align with the organization's hierarchy.

4.2.2 Cloud Service Provider (Organization) Policies

In most CSPs, an organization policy (sometimes called an "organization policy type") is a construct that allows enabling and disabling services for deployments at the deployment or group level. Some CSPs also

[66] Additional material on IAM is provided in Domain 5: *Identity and Access Management*.

support detective or corrective policies that identify policy violations and automatically fix them by restoring the correct settings. Corrective and detective policies are useful in situations where the orchestration engine of the CSP involves multiple steps to coordinate multiple resources, and none of the individual steps are sufficient for a decision. For example, creating a resource and adding a tag might involve two or more API calls in the provider. A corrective control could delete the resource after creation if the tag is not applied, while a preventative control would fail since it cannot evaluate whether the tag is applied during the initial resource creation step.

Policies are notable for their ability to define security parameters for a deployment while remaining independent from the deployment. This external positioning ensures that even administrators with complete control over a deployment cannot modify or delete these policies.

Policies find applications in various scenarios, including:

- Enabling and disabling specific services, such as prohibiting using an unapproved platform service for deployment.
- Blocking particular API calls to prevent unauthorized or harmful operations.
- Disabling regions to comply with geographic regulatory requirements, maintaining data residency and sovereignty requirements.
- Defining conditions like permitting specific API calls only from authorized network sources (e.g., IP addresses). However, this requires both CSP- and service-level support, representing one of the more inconsistent capabilities across providers.
- Augmenting IAM practices to secure organization-level access and operational tools, including preventing a deployment administrator from restricting access to critical visibility and control accounts (e.g., in the event administrator credentials are compromised).

CSP policies can be categorized into three levels based on their scope:

1. **Organization-wide policies** are defined by the CSC and apply to every deployment. Typically, this category includes a limited set of policies due to the challenges in managing exceptions on such a broad scale.
2. **Group-level policies** cover all deployments within a specific group. This level is most commonly used for policy application, and policies at this level can accumulate and reinforce one another, especially when applied to sub-groups. CSPs enforce the combined set of policies, with ones that deny actions almost always taking precedence over any allowing policies at higher levels.
3. **Deployment-level policies** are tailored for individual deployments, allowing for precise security adjustments. While applying policies at the group level is generally considered a better management practice, certain scenarios necessitate deployment-level policies, particularly for deployments with specific and granular security requirements.

Let's look at how we can specifically apply a policy control at the top of the hierarchy, which sets a control on all users that have been added to a main account. Service control policies (SCPs) allow organizations

to specify and control which services and features can be accessed and used for the main account. (Depending on CSP, this is referred to as organization, tenant, or tenancy.)

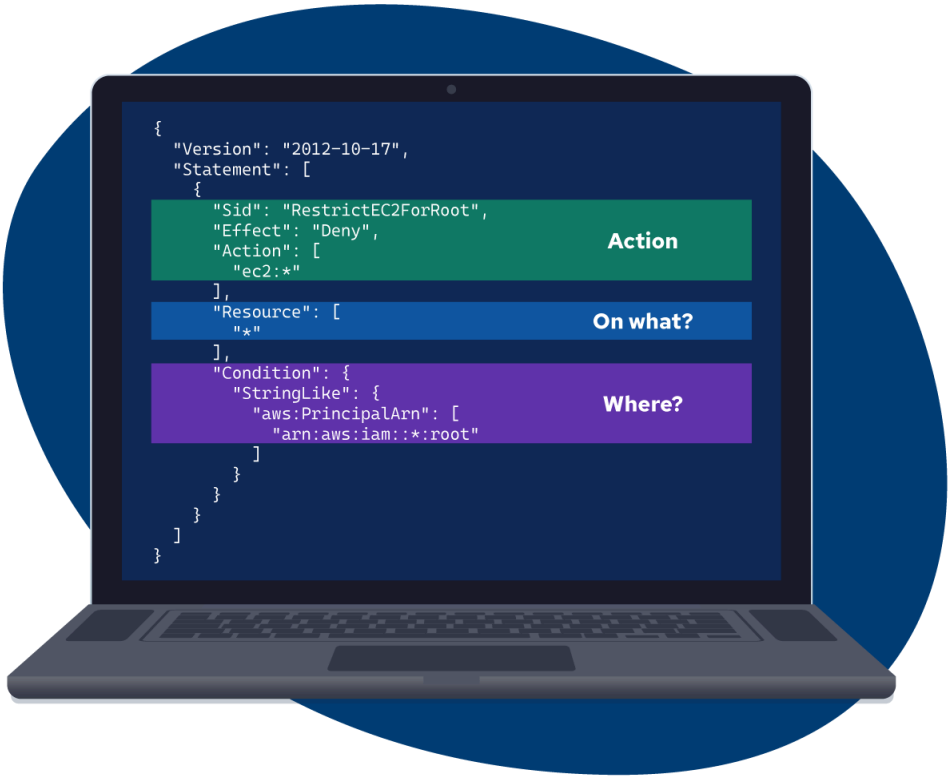


Figure 23: Example of an AWS SCP to Restrict Root User Actions

In the example shown, an AWS SCP is applied at the top of the hierarchy of an AWS Organization, controlling permissions consistently across all member accounts. SCPs do not grant permissions directly to individual accounts. Instead, they define guardrails or limits on actions that are possible in an account.

SCPs allow you to centrally manage and enforce security controls across all deployments within your organization. These policies provide a powerful mechanism to establish guardrails and ensure consistent adherence to security standards. Specifically, SCPs have several key characteristics:

- **Permission Structure:** SCPs use a "deny list" approach, explicitly specifying allowed services and actions, and implicitly denying all others.
- **Enforcement:** SCPs set restrictions at the organizational level, overriding permissive IAM policies attached to individual users or roles.
- **Hierarchy:** SCPs work with IAM policies, defining maximum permissions across a CSC's AWS accounts.
- **Common Use Cases:** SCPs often enforce security and compliance standards, restricting access to specific services or features organization-wide.

4.2.3 Common Organization Shared Services

Centralized logging and security telemetry support collecting desired security feeds to a single destination without complex, manual forwarding of feeds from different deployments and regions. This is necessary for effective security monitoring, threat detection, analysis, and compliance. It is also extremely useful when sending telemetry to a Security Information and Event Management (SIEM) platform or a security data lake[67].

CSP Threat Detection services (e.g., AWS GuardDuty[68]) offer continuous surveillance for malicious activities and unauthorized behaviors within cloud environments. These services are designed to safeguard deployments and workloads by identifying potential threats in real time, enabling prompt response measures to mitigate risks and protect cloud assets.

Tagging policies and standardization are often driven by cost allocation (to identify which internal team should pay for the resource), but can also be useful in implementing Attribute-Based Access Control (ABAC) as part of an IAM strategy[69].

Each of these tools should be mapped into the CSC's organization hierarchy and be tuned to meet the security requirements of the groups and deployments. For example, it is common to have different policies apply to development and production environments, with production being more-tightly locked down but allowing more Internet-exposed resources, and with development allowing more open use of cloud services but tightly restricting or banning any open Internet-facing resources.

There are two other tools that are not necessarily security controls, but are extremely helpful in implementing organization-level security.

- **Account factories** are automation platforms for creating new cloud deployments. The term first appeared in reference to creating AWS accounts and is still commonly used, even when working with other CSPs. Account factories create the new deployment and define the starting configuration. They are a powerful tool for security in ensuring that required security controls and configurations are implemented from the start.
- **IaC templates** define everything from the configuration of a single service to an entire complex application stack. They are usually core to an account factory but are also used extensively in modern development and deployment processes. Security teams can leverage IaC templates to deploy their stacks, provide secure baseline configurations, and integrate security controls into projects.

To the greatest degree possible, organization-level security should be managed outside of the organization/tenant root. Third-party or CSP tools that are used daily should be isolated into a dedicated security deployment. This reduces the chances that the top of the organizational hierarchy can be compromised and used to impact all deployments.

[67] Additional material on security monitoring for data lakes is provided in Domain 6: *Security Monitoring*.
[68] AWS. (2024) Amazon GuardDuty features
[69] Additional material on IAM is provided in Domain 5: *Identity and Access Management*.

4.2.4 Integrated Cloud Security & Management Platforms

Cloud security posture management (CSPM) tools connect to CSPs using APIs and assess the current configuration of cloud resources. They evaluate the posture/configuration at the management plane level but do not connect to resources like a virtual machine to examine the OS or internal configuration. Cloud workload protection platforms (CWPPs) are tools that use various techniques to assess workload (VM, container, or serverless configurations).

Cloud-native application protection platforms (CNAPPs) combine CSPM and CWPP and may include other capabilities, like IaC code scanning or cloud data repositories (CDR).

A fundamental capability of CSPM is its inventory functionality, involving a detailed process to identify every asset within a cloud environment. This process encompasses a wide variety of resources, including servers, storage solutions, databases, and various service configurations. It may also track and identify changes over time. The following diagram illustrates the core capabilities of CSPM:

Figure 24: Core Capabilities of Cloud Security Posture Management (CSPM)

Resource configuration assessment is the next important function performed by CSPM tools. This involves examining the configuration of cloud resources to ensure they conform to established security best practices and standards. By comparing resource settings against recognized industry benchmarks

and specific internal policies, CSPM tools can detect misconfigurations, thereby mitigating potential security risks.

Service configuration monitoring complements the resource configuration function. It focuses on verifying that the configurations of cloud services are secure and in compliance with security and compliance expectations. This is particularly important as cloud services are continually updated and expanded with new features, necessitating regular configuration reviews to maintain a secure environment.

Detecting misconfigurations is a core capability of CSPM solutions, aimed at identifying configuration errors that could pose a significant threat to cloud security. Swiftly detecting these misconfigurations allows CSCs to correct them promptly, reducing the window of opportunity for exploitation by adversaries.

The table below highlights the key differences and use cases between CSPM and CNAPP, showcasing their distinct focuses and functionalities:

	CSPM	CNAPP
Scope	Wide range of cloud security aspects, including:	Focus specifically on cloud-native applications, addressing:
	Infrastructure Configuration: Ensuring proper setup of servers, networks, and storage	Application Development Security: Embedding security into code during development
	Access Controls: Managing user permissions and authentication mechanisms	Deployment Security: Ensuring secure deployment and runtime protection
	Data Encryption: Implementing encryption for data at rest and in transit	Threat Intelligence Integration: Prioritizing critical vulnerabilities
	Logging and Monitoring: Setting up logs and alerts for continuous monitoring	Centralized Compliance Management: Ensuring adherence to standards
	Compliance Audits: Checking compliance with industry standards	Permissions Control: Enforcing least privileged access
	IAM: Managing user identities	Shift Left DevOps Security: Collaborating with developers early in the process
	Network Security Groups (NSGs): Defining firewall rules	Comprehensive Cloud Workload Protection: Detecting vulnerabilities
	Secrets Management: Safeguarding sensitive information	Ease of Use: Simplifying security tool stack
	Patch Management: Keeping software up to date	Depth and Breadth of Insights: Eliminating gaps in visibility
Functionality	Detects misconfigurations - Identifies exposed resources - Manages compliance	Protects cloud-native apps from inception to deployment - Integrates CSPM features with workload security - Includes continuous integration/continuous deployment (CI/CD) pipeline integration
Key Differences	Infrastructure-focused Primarily reactive Focuses on policy enforcement	Application-focused Proactive and preventative Focuses on threat detection and response

	May require integration with other tools	Offers a more holistic and integrated view
Audience	Primarily, security teams and compliance officers	DevOps, security, and development teams

Table 5: CSPM vs. CNAPP: Key Differences and Use Cases

Compliance management is another key function of CSPM. It automates the evaluation process for adherence to various standards and regulatory frameworks. This automation significantly reduces the manual effort associated with compliance audits and enables continuous monitoring of compliance status.

CWPPs provide runtime security for cloud environments beyond just point-in-time configuration checks. By leveraging host-based sensors, they gain visibility into workloads to monitor for threats and malicious activity. Integrated threat intelligence enables detecting known bad actors, while automated actions like firewall blocking can prevent attacks. Continuous protection is maintained even as resources scale. Overall, CWPP delivers full-lifecycle security for cloud workloads by combining configuration checks with runtime monitoring, visibility, and response.

CNAPP represents a more encompassing approach to cloud security and combines multiple capabilities. CNAPP is engineered to safeguard cloud-native applications throughout the development lifecycle and across cloud infrastructures. It typically integrates CSPM functionalities with workload security measures and additional features, such as integration with CI/CD pipelines. This approach centralizes a broad spectrum of security considerations inherent to cloud-based operations.

4.3 Considerations for Hybrid & Multi-Cloud Deployments

In today's diverse IT landscape, CSCs often rely on both hybrid and multi-cloud environments to meet their operational needs. Hybrid cloud deployments connect on-premises data centers with public cloud services, enhancing flexibility and scalability while presenting unique security challenges. Multi-cloud strategies, on the other hand, involve using multiple CSPs to avoid vendor lock-in and optimize performance – but they also increase complexity in security management. This section explores the key considerations for securing hybrid and multi-cloud environments, focusing on effective organization management, IAM, network security, and the strategic use of security tools. Understanding these aspects is crucial for maintaining robust security across interconnected and diverse cloud infrastructures.

4.3.1 Organization Management for Hybrid Cloud Security

A hybrid cloud connects an existing data center or facility to a CSP using a virtual private network (VPN) or dedicated network link. Hybrid cloud used to be thought of solely in terms of network security, but CSPs continue to expand their capabilities. Some examples include:

1. Deploying CSP services into data centers on dedicated hardware. For example, allowing virtual machines or databases to use a similar or the same technology stack a CSP uses in its facilities.

2. Extending management tools, usually through agents, to manage resources (virtual or physical) in a data center from the cloud management plane.

3. Extending identity constructs for use in the data center.

As a general strategy, good cloud and data center security will result in good hybrid cloud security. If weak on either, focus on isolating and compartmentalizing those weaknesses so one does not extend into the other.

The two areas to initially focus on for hybrid cloud security are IAM and networking. A compromised identity provider will affect both environments. Weak network security on either side can extend the blast radius of an attack. Do not assume the cloud is the weak point; attackers are now looking for ways to bridge from data centers into cloud deployments. IAM and cloud are the most common touch points that bridge the two environments. For example, an SSH key may be shared across both environments and expose cloud workloads after a datacenter breach, or the other way around.

In contrast, try to avoid normalizing security between each environment, including policies and tooling, as this is the primary pitfall of hybrid cloud. Cloud is so fundamentally different from the traditional technologies used in data centers that attempting to run one single set of controls will result in gaps and failures. It is important to use the right tools for the right jobs.

The different natures of cloud and data center environments create the second pitfall: hybrid cloud sprawl. Traditional infrastructure is more rigid, with a relative scarcity of resources compared to public CSPs. This is not always true, but is generally true for all but the largest CSCs. All but the most modernized data centers tend to have preset IP address ranges, network architectures, and a large percentage of long-running workloads on static IP addresses. Cloud, on the other hand, is more ephemeral with fewer boundaries operating in a more-distributed organizational structure that more resembles fleets of smaller data centers (per earlier the recommendations on Organizational Hierarchy).

Hybrid cloud sprawl is the complexity created when connecting a small number of data centers directly to a large number of cloud deployments. To be clear, this refers directly to a large volume of VPN or dedicated network links from a given data center into multiple cloud deployments. It also includes connecting multiple on-premises identity providers to multiple cloud deployments, often resulting from poor internal IAM management or M&A. This complexity creates additional security challenges, and a key hybrid-cloud security strategy minimizes sprawl to the greatest degree possible.

Effective hybrid cloud security starts with a strong security foundation across both on-premises and cloud, then carefully organizing and managing the connectivity between the environments. Start secure, know the touch points, and manage the blast radius.

4.3.2 Organization Management for Multi-Cloud Security

Moving to multiple IaaS/PaaS CSPs, especially before achieving maturity within the offerings of a single CSP, creates major security challenges. Every CSP is inherently different at the most fundamental technical levels, and effective security requires a deep understanding of the unique characteristics of each CSP and service. Also, supporting multiple CSPs with shared security services is incredibly challenging for all but the most mature CSCs. A CSC should not move to a second IaaS CSP until it has an effective and efficient security program for the primary CSP.

This recommendation is challenging for most CSCs to follow. Even a tightly governed CSC focused on a single IaaS CSP may find themselves using additional CSPs due to M&A or business relationship/partner requirements. Multi-cloud is an incredibly difficult security challenge but can be managed with sufficient staffing, organization management strategy, and key security-shared services actually designed for multi-cloud.

One common misperception of multi-cloud is that a cloud-agnostic container strategy will support fully portable workloads that allow the CSC to pick any CSP at any point in time, possibly for dynamic cost management. In reality, there are significant obstacles to achieving a cloud-agnostic implementation. These are as much operational challenges as security challenges:

- Containers create workload portability, but not management infrastructure portability. Building out the runtime and orchestration environments for containers still involves considerable overhead.
- Shared services are typically less portable unless fully stateless and containerized. Databases, message queues, notification buses, and other services that underlie modern applications are typically better served by a CSP service on dedicated, non-portable resources,
- You may lose economic, security, and operational benefits, provided by PaaS services from the CSP.

4.3.3 Organization Management for IaaS/PaaS Multi-Cloud

CSCs navigating the complexities of cloud infrastructure often adopt varying strategies regarding their use of CSPs for IaaS deployments. These strategies can be broadly categorized into three distinct approaches, each reflecting different levels of engagement with multiple CSPs based on the CSC's operational needs, maturity, and strategic goals.

There are three strategies for approaching multi-cloud:

- **Single provider:** The CSC uses one CSP for IaaS deployments. If an additional CSP is added due to M&A, that deployment is migrated to the primary CSP.
- **Primary/secondary provider:** All new deployments go to a primary CSP which represents the primary cloud footprint of the CSC. Additional CSPs are supported for limited/isolated deployments. These deployments should be by approval only if a business or technical need cannot be met with the primary provider. They are also supported, as needed, due to M&A

activity. Secondary providers are tightly locked down and use the smallest possible set of services to reduce security and operational management complexity.

- **Full multi-cloud support:** The CSC equally supports two or more major CSPs.

In an ideal world, a CSC selects the strategy best aligned with its maturity. It starts with a single provider, then selectively supports compartmentalized islands with additional CSPs as needed, until it is eventually mature enough to support multiple CSPs. While this is our recommendation, we also understand that many CSCs are forced to support multi-cloud before reaching the expected maturity level for reasons ranging from practical realities to internal politics and business relationships.

However, the journey towards multi-cloud adoption is not always linear or purely driven by CSC readiness. External factors often expedite the transition to multi-cloud strategies, pushing CSCs to navigate multi-cloud complexities before reaching the ideal maturity level. This reality underscores the need for adaptable and extensible cloud management practices and a security strategy such as Zero Trust to accommodate the compellingly dynamic nature of business, technology, and threat landscapes.

4.3.3.1 Tooling & Staffing for IaaS & PaaS Multi-Cloud

As with hybrid-cloud, multi-cloud security starts with good security within each CSP, including using the proper tools for the proper jobs. We discuss these tools throughout this domain, this subset can play an important role in multi-cloud security.

- **IAM/SSO/Federated Identity Brokers:** The vast majority of cloud security failures involve IAM. Starting with a solid identity provider is critical for multi-cloud security. Depending on the identity provider, a Federated Identity Broker may be needed to centralize and normalize the single sign-on (SSO) connections and group/role mappings to multiple providers and deployments.
- **Cloud-focused SIEM:** Every provider has their own range of security telemetry, with multiple sources and formats, and every CSP is different. Tools designed for easy integration with major CSPS that also include a range of pre-built threat detectors can ease the burden of multi-cloud support.
- **CSPM:** CSPM is an evolving category, with expanding capabilities that bleed into other new or existing product categories. CSPM allows you to monitor the configuration, security, and compliance of multiple CSPs from a central tool.

Many other tools will support a security program, but this set is the foundation for multi-cloud. It manages user connectivity to multiple clouds, tracks critical security telemetry centrally, and provides visibility into multi-cloud security and compliance configuration.

Staffing is a greater challenge than tooling. There is no shortage of security product vendors on the market, but there is still an ongoing shortage of skilled cloud security professionals. Many CSCs also attempt to transition to cloud without increasing staffing, forcing existing staff to develop cloud skills while still supporting traditional infrastructure.

Each CSP is fundamentally different at the deepest technical levels, and each demands specific knowledge. The more services consumed from a given CSP, the wider the range of knowledge is needed to secure the different services. At a minimum, a CSC should have at least one subject matter expert for each cloud platform that hosts any significant (or critical) footprint. A primary/secondary strategy can reduce the need for a dedicated expert for each platform.

Many CSCs, especially smaller ones, attempt to shift the burden of providing adequately skilled staffing levels to Managed CSPs (also known as *Managed Service Providers*). While this can be a viable strategy in many cases, it does not shift accountability for security and governance. Furthermore, it is vital to ensure that the managed service provider's vision, strategy and capabilities align with the desired future state of the CSC.

4.3.4 Organization Management for SaaS Hybrid & Multi-Cloud

CSCs today leverage various SaaS CSPs to enhance their operational capabilities. Unlike IaaS, which often involves consolidation and higher flexibility and security responsibilities for CSCs, the SaaS landscape has its own challenges. These include a wide range of offerings for various business applications, a broad spectrum of security maturity, and diverse technologies across CSPs. However, SaaS typically requires lower levels of security responsibility from customers. This variability stems from SaaS's ability to offer CSCs an effective and targeted means to harness innovation to meet their business needs.

Effective SaaS security management within CSCs begins with diligent portfolio management. As SaaS CSPs are evaluated for their ability to functionally meet business needs, they should also undergo thorough assessments regarding their security and compliance measures. Subsequently, they can be authorized to handle specific types of data based on classification. This authorization process, and the details of each SaaS CSP, should be meticulously documented and maintained in a central registry. Should there be a request from within the business units to adopt a new SaaS CSP within an already serviced category, a robust business justification could be required to support the addition of a new CSP over or along with an already approved one.

SaaS solutions frequently necessitate integrations with other applications, whether internal applications in a hybrid cloud model or other SaaS offerings. These integrations facilitate the flow of data between applications, sometimes in a manner that does not directly link back to individual users. Therefore, establishing governance over these integrations helps maintain security and control over data movement.

Two types of tools can help management of multiple SaaS CSPs within a security program:

1. **Federated Identity Brokers:** Federated Identity Brokers are integral to Identity-as-a-Service service offerings, and are useful for mediating federated identity management connections between a CSC's identity provider(s) and its cloud service instances. With pre-built integrations for major CSPs, and a unified dashboard for user access to different services, federated identity brokers significantly streamline the CSC and lifecycle administration of user access and permissions.
2. **Cloud Access and Security Brokers (CASB):** CASBs can be useful for managing a CSC's SaaS portfolio, offering access control and monitoring capabilities, and enforcing which SaaS CSPs are

utilized by which users and from which locations. As the CASB landscape continues to evolve, including the implementation of Zero Trust security principles, some vendors have expanded their focus to include configuration security, giving rise to the concept of SaaS Security Posture Management (SSPM). The primary benefits of CASBs include providing insights and a degree of control over a CSC's SaaS usage. Meanwhile, SSPM is concentrated on monitoring and maintaining security hygiene. Advanced CASB solutions may also feature real-time monitoring, data loss prevention (DLP), and other functionalities to enhance SaaS security.

By integrating these tools and strategies (ideally in alignment with Zero Trust security strategy and principles), CSCs can more effectively manage their SaaS portfolios, helping to ensure security and compliance while capitalizing on the innovative solutions offered by SaaS providers.

4.3.5 Zero Trust Security Strategy for Hybrid & Multi-Cloud

Successful cyberattacks generally exploit trust in some manner. This makes "trust" a dangerous vulnerability that should be mitigated and managed. Zero Trust is a cybersecurity strategy premised on the idea that no user or asset is to be implicitly trusted. It assumes that a breach has already occurred or will occur, and therefore, a user should not be granted access to sensitive information by a single verification performed at the enterprise perimeter. Instead, each user, device, application, and transaction must be continually verified.[70]

Zero Trust is an enterprise security strategy that encompasses cloud/multi-cloud, on-premises and hybrid systems, internal and external partner/stakeholder user (CSC-managed and bring your own device) endpoints, and is inclusive of operational technology (OT), Industrial Control Systems (ICS), Internet of Things (IoT), and physical security. Many security experts maintain that Zero Trust is the best enterprise security strategy for current, distributed enterprise cloud/multi-cloud and hybrid environments with a significant remote workforce and OT/IoT components. This strategy inherently and consistently leads to the types of localized access controls, and segmentation and isolation between environments and applications recommended in prior sections.

Summary

Leveraging the organization or tenant hierarchy within a cloud environment is a strategic approach to managing several critical aspects of cloud deployments, including minimizing the blast radius of potential security incidents, adhering to service limits, and achieving logical separation of deployments. This hierarchical structure is a foundation for aligning various security controls, underscoring the importance of thoughtful and strategic implementation. The hierarchy not only facilitates effective management but also enhances the security posture of cloud deployments.

The identity provider or directory is at the forefront of managing access and permissions within the CSP's environment. This component is the initial layer for managing individual permissions, which are subsequently enforced across the CSP's services. CSP policies play a pivotal role as preventative controls, offering a robust mechanism for governance across the hierarchy. These policies enable CSCs to control

[70]CISA. (2022) *NSTAC Report to the President on Zero Trust and Trusted Identity Management,* page 1, and adopted as the official CSA definition of Zero Trust.

the usage of services and enforce specific configurations, thereby adding an additional layer of security and compliance.

Tools such as Security Information and Event Management (SIEM), Security Data Lakes, and CSPM are indispensable for achieving centralized visibility within the cloud environment. They provide comprehensive insights into security events, configurations, and compliance status across cloud deployments, enhancing the CSC's ability to effectively detect and respond to potential security threats.

In hybrid cloud environments, the focus shifts towards IAM, particularly directories and network connection points. These components are crucial for securing the interface between on-premises infrastructure and cloud services, ensuring secure access and data flow between different environments.

For CSCs navigating multi-cloud strategy complexities, the most valuable asset is adequate subject matter expertise. Individuals with deep expertise in specific cloud platforms and services are instrumental in navigating the unique challenges and opportunities presented by multi-cloud deployments. They provide the knowledge and insights necessary to optimize the use of cloud services across different CSPs, ensuring that security, compliance, and operational efficiency are maintained at the highest levels. This strategic approach to cloud management, emphasizing hierarchical organization, preventative controls, centralized visibility, and expert guidance, is fundamental to achieving a secure and efficient cloud infrastructure.

Recommendations

Cloud Governance and Management

- Create a centralized Cloud Deployment Registry
- Define an organization hierarchy using multiple deployments
- Include exceptions for special use cases
- Support a low-friction process for creating new deployments
- Use CSP policies to manage services and capabilities

Security Strategy and Controls

- Adopt a comprehensive, modern enterprise security strategy
- Minimize access to the CSP's "root" or "Global Administrator" credentials
- Use a CSPM tool to monitor and maintain security and compliance
- Run security tooling from a deployment outside of the organization/tenant root
- Establish appropriate security policies for cloud and data center deployments
- Pay attention to IAM and network connections in hybrid deployments
- Formalize requirements for hybrid connections
- Ensure security controls for containers in hybrid/multi-cloud environments

Multi-Cloud Strategy

- Do not attempt multi-cloud in production unless sufficiently mature
- Establish a multi-cloud strategy corresponding with cloud maturity

- Have provider-specific security subject matter experts
- Ensure sufficient security staffing for multi-cloud

Cloud Security Monitoring and Management

- Use a CSPM that supports all CSPs in use
- Consider using a CASB tool to manage SaaS services
- Consider using an SSPM tool for SaaS platform visibility

Cloud Interoperability and Portability

- Consider the interoperability and portability strategy

SaaS Governance

- Maintain a registry of approved SaaS platforms

Additional Resources

- Roles and Responsibilities of Third-Party Security Services | CSA
- AWS Landing Zone
- Azure Landing Zone
- Google Landing Zone
- Oracle Cloud Infrastructure - Landing Zone

Domain 5: Identity and Access Management

Introduction

Identity and access management (IAM) ensures that only authorized identities have the right access to the right resources. With cloud platforms consolidating numerous administrative functions of data centers and services into unified Internet-accessible web consoles and application programming interfaces (APIs), IAM acts as the new perimeter in cloud-native security, protecting sensitive resources from unauthorized access and misuse.

In both public and private clouds, cloud service providers (CSPs) and cloud service customers (CSCs) are tasked with managing IAM within acceptable risk tolerances. While we will review fundamental IAM concepts, the focus will be on the characteristics and challenges of IAM in the cloud and ensuring their effective management.

Cloud computing introduces new dimensions to managing IAM comparative to on-premises systems. While the core security issues may not be new, their impact is magnified and can have rippling repercussions in the cloud landscape.

The key differences are:

- The relationship between the CSP and the CSC, and their respective responsibilities.
- The consolidation of multiple administrative interfaces.
- The exposure of these interfaces to the Internet, especially for public cloud environments.

IAM cannot be managed solely by the CSP or the CSC. It requires a trust relationship between both parties, a clear designation of responsibilities, and the technical mechanics to facilitate its management. Further, CSCs dealing with multiple CSPs have the added complexity of managing multiple IAM solutions in line with each provider's unique policy.

This domain focuses primarily on IAM between a CSC and *CSP*s or between *CSP*s and services. It does not discuss all aspects of managing IAM within a cloud application, such as the internal IAM for an enterprise application running on Infrastructure as a Service (IaaS).

Learning Objectives

In this domain, you will learn to:

- Define Identity Federation and its role in authentication.
- Differentiate between IAM policy types for cloud environments.
- Identify the key components of Identity and Access Management (IAM).
- Manage customer identities effectively in cloud applications.

5.1 How IAM is Different in the Cloud

IAM is always complex. At heart, some type of entity (e.g., a person, system, piece of code) is mapped to a verifiable identity associated with various attributes (which can change based on current circumstances), and then making a decision on what the entity can or cannot do based on entitlements. The complexity of getting this verifiably right increases with the number of disparate systems, services, and technologies involved.

There are three key differences with IAM for cloud computing:

1. IAM now spans multiple organizations in cloud computing - there can be multiple CSPs for any CSC. Those CSCs likely use a large number of services across the spectrum of cloud service models. Identity Federation is the primary tool to manage this problem by building trust relationships between organizations and enforcing them through standards-based technologies.

2. CSPs all use their own proprietary IAM systems. Not only does their technology differ, but the entire architecture and even much of the terminology are different. CSCs will need to learn, understand, and implement multiple different models. While this is also true of different applications and software stacks in traditional architectures, the cloud adds this layer to the entire management plane and even the infrastructure of connected services.

3. CSPs consolidate management and administrative functions into unified web consoles and APIs. In the case of public clouds, these are typically on the Internet and usually protected with little more than a username and a password (and maybe optional strong authentication or policy conditions). Private clouds and container platforms often expose their management planes to the Internet, either directly or via security misconfigurations.

Federation and the multitude of IAM systems define much of the complexity of managing cloud identity and access, while the combination of unifying administrative functions and placing them on the Internet dramatically increases their criticality. These issues are not theoretical; the vast majority of cloud-native security breaches typically originate with IAM failures.

Moving to the cloud also creates opportunities for IAM improvements. Major providers often support more recent capabilities such as attribute-based access control (ABAC)s, Policy-Based Access Controls (PBAC), Role-Based Access Controls (RBAC), risk-based authentication and authorization, temporary

credentials, secrets management, Just-In-Time[71] (JIT) Access and other advanced options. These create a potential responsiveness and granularity of control that security professionals have long worked towards.

IAM spans essentially every domain in the CCSK. The following section starts with a review of some fundamental IAM concepts and terminology that not all readers may be familiar with, then delves into the cloud impacts – first on identity, then on access management.

5.1 Fundamental Terms

IAM is a broad area of practice with its own terminology that can be confusing, especially since some terms have different meanings in different contexts (and are used in areas outside IAM). Even the term "IAM" is not universal and is also referred to as Identity Management (IdM).

Gartner defines IAM as "*the security discipline that enables the right individuals to access the right resources at the right times for the right reasons.*[72]" Before we get into the details, here are the high-level terms most relevant to a discussion of IAM in cloud computing:

- **Access control:** Restricting access to a resource, based on the permissions granted to the entity.
- **Assertion:** Statements from an Identity Provider (IdP) to a Relying Party (RP) that contain information about an entity. Federation technology is generally used when the IdP and the RP are not a single entity or are not under common administration. The RP uses the information in the assertion to identify the entity and make authorization decisions about their access to resources controlled by the RP.
- **Attribute:** A characteristic or property of an entity that describes its state, appearance, or other relevant aspects. Attributes can include a variety of information such as personal details, user roles, security clearance levels, the time of an access request, or the location from which the request is made.
- **Attribute-Based Access Control[73] (ABAC):** An access control or entitlement that requires specific attributes, such as multi-factor authentication (MFA), the user logging in from a managed system, or the targeted resource having a particular tag.
- **Authentication:** Verifies the identity of a user, process, or device, often as a prerequisite to allowing access to resources in a system.
- **Authoritative source:** A trusted system that holds the most accurate and up-to-date information about an entity's identity attributes. This information is then used by other IAM components for tasks like authentication and authorization.

[71] JIT access reduces the window for abuse by eliminating static credentials. Access is requested for a session and externally approved, then revoked at the end of the session.
[72] Gartner. (2024) *Gartner Glossary: Identity and Access Management*
[73] NIST (2024) *CSRC: Attribute Based Access Control*

- **Authorization:** The decision to permit or deny a subject access to system objects (e.g., network, data, application, service,)

- **Entitlement:** Maps identities to authorizations with required attributes (e.g., user X is allowed access to resource Y when Z attributes have designated values). We commonly refer to a map of these entitlements as an *entitlement matrix*. Entitlements are often encoded as machine-readable policies for distribution and enforcement.

- **Entity:** An entity refers to a unique, identifiable actor in a computer system. In the context of cybersecurity, an entity can be a user, a device, an application, or a system that is identified and authenticated by an IAM system. Entities can have different roles and permissions within the system, and their actions and access to resources are typically logged for auditing and security purposes.

- **Federated IdM:** Allows users to access multiple systems or applications using a single set of credentials, often provided by an IdP. This is the key enabler of Single Sign-On (SSO) and is a core capability in cloud computing.

- **IAM Principal:** A user, role, or other identity type that can request an action or operation on a CSP resource.

- **Identifier:** The artifact used to assert the identity. This could be digital as in the case of a cryptographic token, or it could be physical, such as a driver's license or passport.

- **Identity:** the unique expression of an entity within a given namespace. An entity can have multiple digital identities, such as a single individual having a work identity (or even multiple identities, depending on the systems), a social media identity, and a personal identity

- **Identity Provider (IdP):** The source of the identity in a federation. Responsible for enforcing authentication policies. IdP can also play an important role in authorization strategy by mapping CSP roles to IdP attributes. The IdP is not always the authoritative source, but can sometimes rely on one.

- **Multi-Factor Authentication (MFA):** A mechanism through which an identity is authenticated via additional factors such as something you know, something you have or something you are. This is a important technique in containing identity-based attacks such as stolen user ID / password etc. It is commonly used in authenticating identities before access is granted to critical systems such as finance, health, etc. This technique is also used in conditional access such as logging from an unknown device, unknown place / country (“impossible travel”), and so forth.

- **Persona:** A user-centric view to help understand how different user types interact with the system. It represents a category of users with similar characteristics and leads to the development of roles. For example, a cloud system could define the personas of a developer, a security analyst, a sales representative, or a content creator by describing what they need to do. This could then lead to developing unique roles and specific permissions.

- **Policy-Based Access Control (PBAC):** Access requirements defined in a machine-readable policy document that typically provides extensive flexibility and granularity with support for

various conditions and other variables, such as attributes. PBAC is complementary to RBAC and ABAC and is often how those are defined and managed. PBAC policy documents are also managed using version control repositories and infrastructure as code (IaC), sometimes called conditional access.

- **Relying Party (RP):** A service that relies on an IdPto verify a user's identity and access rights, and then grants entitlements to its own resources. Sometimes referred to as Service Provider.

- **Role:** Provides a permission-centric view, defining the access level for users to perform specific tasks. Roles can be unique to a user or shared among users. A single user might have multiple roles depending on their responsibilities. Conversely, multiple users can share the same role if they have the same access needs. For example, every persona who was defined under] "sales representative" would have the same permissions.

- **Role-Based Access Control (RBAC)** is a more common model than ABAC, where access is granted to all users with a given role (e.g., developer or administrator).

A few more terms, including the major IAM standards, will be covered in their relevant sections below. Find more definitions relating to IAM in CSA's IAM Glossary.[74]

5.2 Federation

Identity Federation establishes the relationship between an IdP, which handles authentication, and an RP where authorizations are managed. In the cloud, the RP is typically a cloud service or application. Because one IdP can federate[75] to many RPs, this consolidates user management (creation, role assignment, attributes, authentication, and deletion) while supporting authorizations and access controls among distributed systems.

There are quite a few IAM standards and frameworks, and many of them can be used in cloud computing. Despite the wide range of options available, the cloud security industry is converging around a core set that is commonly seen and supported by most IdPs.

5.2.1 Common Federation Standards

Below are some of the commonly used standards. This list does not reflect any particular endorsement and does not include all options but is merely a representative sample of what is most commonly supported by the widest range of providers.

- **Security Assertion Markup Language (SAML)** is an OASIS (Organization for the Advancement of Structured Information Standards) standard for federated IdM that supports authentication and authorization. It uses XML to make assertions between an IdP and an RP. Assertions can contain authentication statements, attribute statements, and authorization

[74] CSA. (2024) *Identity and Access Management Glossary*
[75] The process of linking the identity management systems of different organizations to allow users from one organization to access resources and services of another organization securely and seamlessly.

decision statements. Both enterprise tools and CSPs widely support SAML, but it can be complex to configure initially. SAML is well suited for traditional web-based client-server applications.

- **OAuth** is an IETF (Internet Engineering Task Force) standard for authorization widely used for web services (including consumer services). OAuth is considered an authorization protocol that allows users to grant third-party applications limited access to resources without sharing their credentials (like passwords) directly with those applications. OAuth is popular for authorizing API access or connecting 3rd parties to applications. OAuth is designed to work over HTTP and is most often used for delegating access control and authorizations between services.

- **OpenID Connect (OIDC)** is a standard for federated authentication widely supported for web services. It adds an authentication layer to OAuth and is based on HTTP with URLs used to identify the IdP and the user/identity (e.g., http://identity.identityprovider.com). OIDC 1.0 is very commonly seen in consumer services, and there is growing support for it in commercial products. One example would be Single Page Applications (SPA - e.g., Facebook). *OpenID* is a standard for authentication and is distinct from OIDC. OpenID 2.0 is deprecated and has been largely replaced by OIDC.

Two other standards that are not as common, but can be useful for cloud computing, are as follows.

- **eXtensible Access Control Markup Language (XACML)** is a standard for defining ABACs and authorizations. It is a policy language for defining access controls at a Policy Decision Point (PDP) and then passing them to a Policy Enforcement Point (PEP). It can be used with both SAML and OAuth since it solves a different part of the problem, i.e., what an entity is allowed to do with a set of attributes, as opposed to handling logins or delegation of authority.

- **The System for Cross-domain Identity Management** is a standard for exchanging identity information between domains. It can be used to provision and deprovision accounts in external systems and to exchange attribute information.

5.2.2 How Federated Identity Management Works

Federation involves an IdP making assertions to an RP after building a cryptographic trust relationship between them. A practical example is a user logging into their work network, which hosts a directory server for accounts. The IdP and RP share a secret. When the user opens a browser connection to a SaaS application, instead of a login process being initiated, there are a series of behind-the-scenes operations (steps 1 to 6 in the figure), where the idP (internal directory server) asserts the user's identity, authenticates the user, and possibly forwards any necessary attributes. The RP can then trust those assertions and consequently log the user in without the user entering any credentials. TheRP does not need a username or password for that user in its own namespace; instead, it relies on the IdP to assert successful authentication. The user simply goes to the website for the SaaS application and is logged in, assuming they are successfully authenticated with the internal directory.

This does not imply there are not other techniques or standards used in cloud computing for identity, authentication, and authorization. Most CSPs, especially IaaS, have internal IAM systems that might not

be used or can be connected to a CSC using these standards. For example, HTTP request signing[76] is commonly used for authenticating REST APIs, and internal policies on the CSP side to manage authorization decisions. The request signing might still support SSO through SAML, or the API might be completely OAuth-based, or even use its own token mechanism. All are commonly encountered, but most enterprise-class CSPs support federation.

The essential concepts when choosing an identity protocol are:

- No protocol is a silver bullet that solves all identity and access control problems.
- Identity protocols must be analyzed in a given use case context. For example, browser-based SSO, API keys, or mobile-to-cloud authentication could each benefit from different approaches.
- The key assumption should be that identity is a perimeter in and of itself, analogous to a demilitarized zone[77].

The following figure illustrates the workflow of OpenID federation in cloud security, detailing the steps from user authentication through an IdP to accessing services from a relying party.

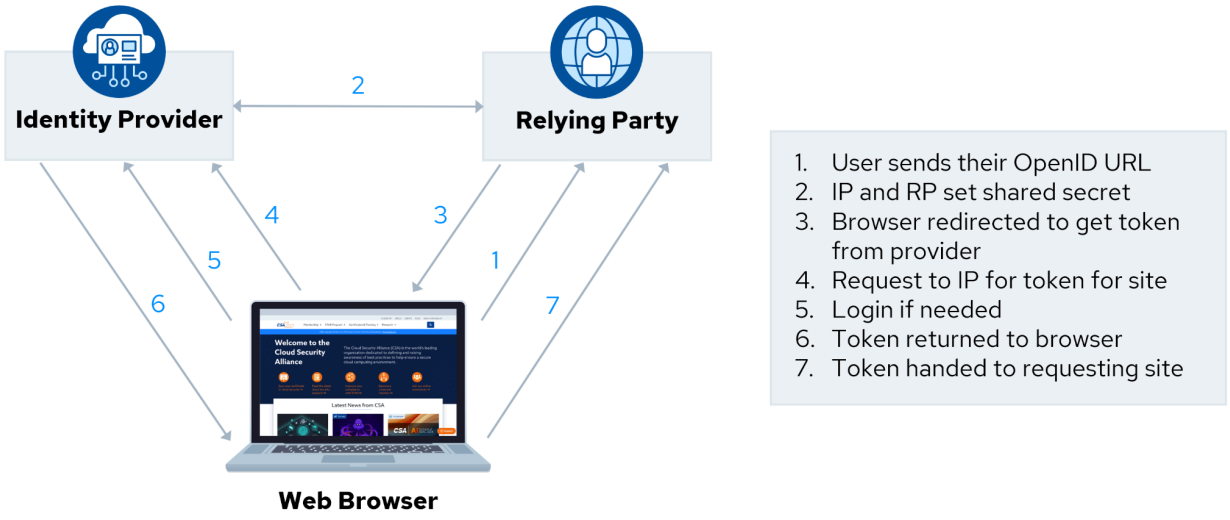


Figure 25: Workflow of OpenID Federation in Cloud Security

5.2.3 Managing Users & Identities for Cloud Computing

The "identity" part of identity management focuses on the processes and technologies for registering, provisioning, propagating, managing, and deprovisioning identities. Managing identities and provisioning them in systems are challenges that the information security sector has been tackling for decades. It was not so long ago that IT administrators needed to individually provision users in every different internal system. Even today, with centralized directory servers and a range of standards, true SSO for everything is relatively rare; users still require credentials, albeit a much smaller set than in the past.

[76] IETF. (2024) RFC 9421
[77] A demilitarized zone is a physical or logical subnetwork that acts as a buffer zone between an internal network (such as a corporate LAN) and an external network (typically the internet).

When deciding how to manage users and identities for cloud computing, CSPs and CSCs need to start with two fundamental decisions on how to manage identities.

- CSPs should support internal identities, identifiers, and attributes in their managed namespace for users directly accessing the service. Additionally, they should support federation to prevent CSCs from manually provisioning and managing every user in the provider's system and issuing separate credentials for each one.
- CSCs need to decide the best location for managing their identities and choose the appropriate architectural models and technologies to integrate with CSPs.

A CSC can log in to a CSP and create all their identities in their system. However, this approach is not scalable for most, besides maybe the smallest CSCs, which is why most turn to federation. There can be exceptions where it makes sense to keep all or some of the identities with the CSP isolated, such as backup administrator accounts to help debug problems with the federated identity connection.

When using federation, CSCs need to identify the authoritative source of unique identities, often an internal directory service. Next, they must decide whether to use this source directly as the IdP, use another identity source fed by it (like a directory from a human resources system), or integrate an identity broker.[78] There are two primary architectures:

- Hub and spoke: internal IdPs/sources communicate with a central broker or repository that then serves as the IdP for federation to CSPs.
- Free-form: internal IdPs/sources (often directory servers) connect directly to CSPs.

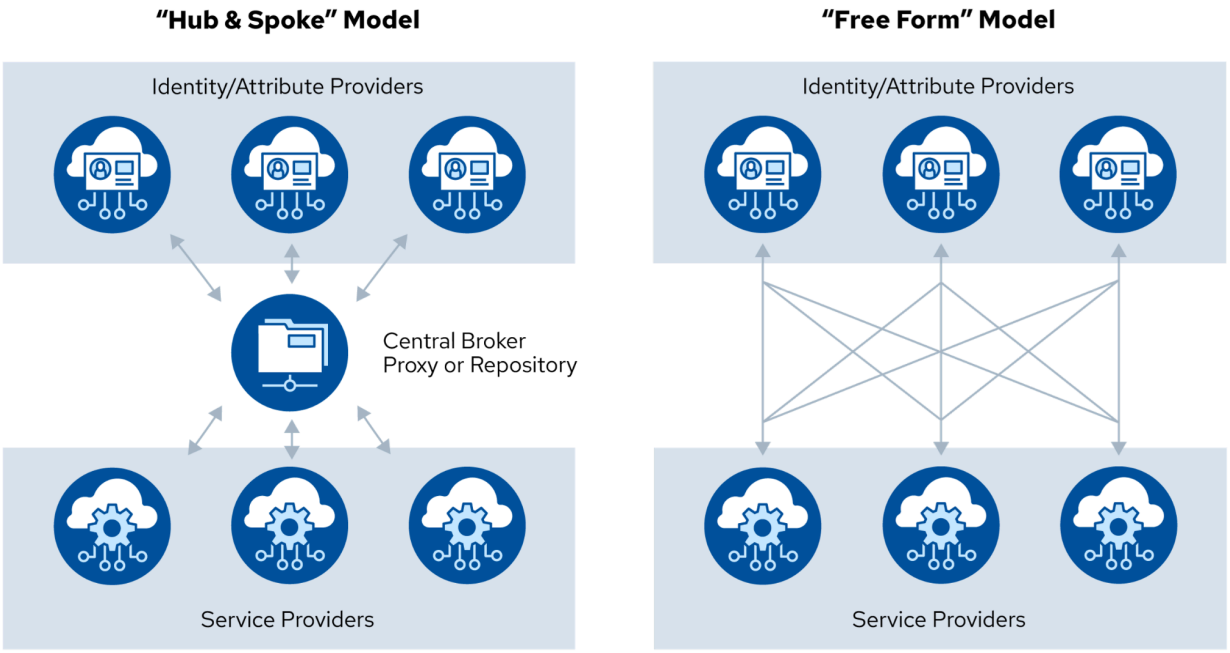


Figure 26: Architectural Models for Federated Identity Management: Hub & Spoke vs. Free Form

[78] *Broker* is an intermediary service that connects multiple service providers with multiple IdPs.

Directly federating internal directory services servers in the free-form model raises a few issues:

- The directory needs Internet access. This can be problematic if it violates security policy.
- It may require users to VPN back into the corporate network before accessing cloud services.
- Depending on the existing directory services server, and especially if there are multiple directory services servers in different organizational silos, federating to an external provider may be complex and technically difficult.

Federated Identity Brokers handle federating between IdPs and RPs. They can be located on the network edge or even in the cloud to enable web SSO. IdPs do not need to be located solely on-premises; many CSPs now support cloud-based directory services that can manage federation both internally and with other cloud services.

For example, more complex architectures can synchronize or federate a portion of a CSC's identities from an internal directory through an identity broker and then to a cloud-hosted directory. This cloud-hosted directory can then serve as an IdP for other federated connections.

When implementing these solutions, there are several process and architectural decisions to consider:

- How to manage identities for application code, systems, devices, and other services. The same model and standards may be leveraged or decided to take a different approach within cloud deployments and applications. Defining the identity provisioning process and how to integrate it into cloud deployments. There may also be multiple provisioning processes for different use cases, although the goal should be to have a unified process. An often-encountered example is associated with staff member onboarding versus contractor onboarding.

- Establish the deprovisioning process. Proper governance requires adequate and sometimes swift removal of identities and access rights, while maintaining a proper forensic trail of the usage of those rights.
    - If the CSC has an effective provisioning process in place for traditional infrastructure, this should ideally be extended into cloud deployments. However, if existing internal processes are problematic, then the CSC should instead use the move to cloud as an opportunity to build a new, more effective process.
- Provisioning and supporting individual CSPs and deployments. There should be a formal process for adding new CSPs into the IAM infrastructure. This includes the process of establishing any needed federation connections, as well as:
    - Mapping attributes (including roles) between the IdP and the RP.
    - Passing attributes to support ABAC/PBAC (e.g., MFA status or the IP address the user has authenticated from).
    - Enabling required monitoring/logging, including identity-related security monitoring, such as behavioral analytics.

- Building an entitlement matrix (discussed more in the next section).
- Documenting any break/fix scenarios in case there is a technical failure of any of the federation (or other techniques) used for the relationship. If an IdP goes down, or the Internet connection to the CSP goes down, or the CSP's federation support goes down, is there a business continuity plan?
- Ensuring incident response plans for potential account takeovers are in place.
- Implementing deprovisioning or entitlement change processes for identities and the CSP. With federation this requires effort by both sides of the relationship.

Lastly, CSPs need to determine which IdM standards to support. Some CSPs support only federation while others support multiple IAM standards plus their own internal user account management. CSPs that serve enterprise markets typically need to support federated identity, and most likely SAML.

5.3 Strong Authentication & Authorization

Ensuring robust authentication and authorization is vital for cloud security. This section outlines key practices for securing cloud access.

Authentication verifies user identity, essential for accessing cloud services. MFA is crucial, adding extra layers of security beyond passwords. Methods include hard tokens, soft tokens, and biometrics, each offering different levels of protection.

Authorization determines user permissions. Effective models like RBAC and PBAC manage and enforce these permissions, providing granular control.

CSPs enforce these policies, but CSCs must define and manage them. Advanced models like ABAC enhance security by allowing context-aware access decisions. By implementing strong authentication and authorization practices, organizations can protect their cloud resources and ensure secure access.

5.3.1 Authentication & Credentials

Authentication is the process used to verify an identity. It is important not just for logins, but for all situations where an identity must be verified and linked to specific permissions or roles within a system or process. The duty of ensuring reliable authentication falls on the IdP.

The biggest impact of cloud computing on authentication is the increased need for strong, MFA due to two main issues:

- **Broad network access**: Cloud services are always accessed over the network, often over the Internet. This means that if credentials are lost or stolen, they can lead to an account takeover more easily, as attacks are not restricted to the local network.

- **Greater use of federation for Single Sign-On (SSO):** Using a single set of credentials for multiple cloud services means that if those credentials are compromised, a larger number of services could be at risk.

MFA offers one of the strongest options for reducing account takeovers. While it is not a cure-all, relying on a single factor (password) for cloud services creates high risk. When using MFA with federation, the IdP can and should pass the MFA status as an attribute to the RP.

There are multiple options for MFA, including:

- Hard tokens are physical devices that generate one-time passwords (OTPs) for human entry or need to be plugged into a reader. These are the best option when the highest level of security is required.
    - Tokens that are plugged in are more reliable than tokens that generate a OTP that the user types in.
    - There are multiple examples of users being tricked into entering or sharing OTP codes via phishing or other targeted attacks.
- Soft tokens work similarly to hard tokens but are generated by software applications that run on a smartphone or computer. Soft tokens are an excellent option but could be compromised if the user's device is compromised, and this risk needs to be considered in any threat model.
- Out-of-band passwords are text or other messages sent to a user's phone (usually) and are then entered like any other OTP generated by a token. Although also a good option, any threat model must consider message interception, especially with SMS. SMS is no longer recommended due to SIM swapping and other attacks on that infrastructure.
- Biometrics are increasingly an option, thanks to biometric readers now commonly available on mobile phones. For cloud services, the biometric is a local protection that does not send biometric information to the CSP and is instead an attribute that can be sent to the provider. As such the security and ownership of the local device needs to be considered.

As organizations seek to strengthen their authentication mechanisms, additional methods beyond traditional approaches are being adopted. These methods aim to enhance security while improving user convenience.

5.3.1.1 Additional Authentication Methods

Passwordless Authentication: This approach utilizes a local token or certificate to circumvent the need for passwords, and is akin to an SSO token, associated with a service, user, and device. It simplifies the user experience and reduces the risks of phishing and password exposure during data breaches. Nevertheless, passwordless methods are not recommended for administrative-level cloud service accounts and are predominantly used for user authentication to consumer applications. It is important to note that passwordless systems should not replace MFA.

FIDO (Fast IDentity Online): As the current industry standard for passwordless authentication, FIDO might be recognized under various names, such as "Passkeys or "Webauthz," representing a technology step forward by providing a phishing-resistant authentication method. FIDO allows the user to define trusted devices that can be used as authentication factors during the login process. FIDO can also be enhanced with physical tokens that are plugged into or wirelessly connected to the access device. The FIDO Alliance, which develops the standard of passwordless authentication, is composed of major IT vendors including the CSPs and Identity and Access Management solution providers.

5.3.2 Entitlement & Access Management

The terms authorization, and access control all overlap somewhat and are defined differently depending on the context.

- Authorization is permission to do something — e.g., access a file or network, or perform a certain function like an API call on a specific resource.
- Access control allows or denies the usage of that authorization, so it includes aspects like assuring that the user is authenticated before allowing access.
- Cloud entitlement refers to the permissions or rights granted to users in cloud environments to access specific resources or services. Entitlements typically determine what actions a user can perform on a given resource, such as reading data, writing data, configuring settings, or managing other users.

An entitlement maps identities to authorizations and any required attributes (e.g., user X is allowed access to resource Y when attributes Z have designated values). We commonly refer to a map of these entitlements as an entitlement matrix. When using PBAC, entitlements are often encoded as technical policies for distribution and enforcement.

Entitlement	Super-Admin	Service-1 Admin	Service-2 Admin	Dev	Security - Audit	Security - Admin
Service 1 List	X	X		X	X	X
Service 2 List	X		X	X	X	X
Service 1 Modify Network	X	X		X		X
Service 2 Modify Security Rule	X	X				X
Read Audit Logs	X				X	X

Table 6: Sample Entitlement Matrix for Cloud Access Management

Cloud impacts entitlements, authorizations, and access management in multiple ways:

- CSPs and platforms have their own set of potential authorizations specific to them. Unless the CSP supports XACML (rare today) the CSC user usually needs to configure entitlements within the cloud platform directly.
- The CSP is responsible for enforcing authorizations and access controls.
- The cloud user is responsible for defining entitlements and properly configuring them within the cloud platform.
- Cloud platforms tend to have greater support for the ABAC and PBAC models for IAM, which offer greater flexibility and security than the RBAC model. RBAC is the traditional model for enforcing authorizations and relies on what is often a single attribute (i.e., a defined role). ABAC allows more granular and context-aware decisions by incorporating multiple attributes, such as role, location, authentication method, and more.
- PBAC supporting ABAC is the preferred model for cloud-based access management.
- When using federation, the cloud user is responsible for mapping attributes, including roles and groups, to the CSP and ensuring that these are properly communicated during authentication.

CSPs are responsible for supporting granular attributes and authorizations to enable ABAC and effective security for cloud users.

Here is a real-world cloud example. The CSP has an API for launching new virtual machines (VMs). That API has a corresponding authorization to allow launching new VMs, with additional authorization options for what virtual network a user can launch the VM within. The cloud administrator creates an entitlement that says users in the developer group can launch VMs in only their project network, and only if they are authenticated with MFA. The group and the use of MFA are attributes of the user's identity. That entitlement is written as a policy that is loaded into the CSP's system for enforcement.

5.3.2.1 Resource Access Controls & Policies

Until now we have mostly discussed authorizations and entitlements within the context of an entity of the CSP requesting an action from the centralized IAM management for the platform. Many CSPs also support rules and/or policies that are applied to an individual resource (e.g., a storage location) and can account for access outside of entities provisioned within the CSP.

For example, most storage services allow direct external access to objects by users from other locations using shared links, IP restrictions, or temporary passcodes.

These entitlements are implemented in resource-level policies that may circumvent central IAM governance. Especially with storage services, this is a common source of data exposure. It is even possible to have a situation where an IAM user in a CSC is explicitly denied access to a resource within the primary IAM system, but can still access the resources due to a weak resource policy.

To reduce this risk, some CSPs provide top-level security controls to restrict external sharing or using resource policies that potentially enable public or external access. CSCs can also use automation to identify and manage these policies.

5.3.3 Conditional Access, Tokens, Sessions, & IAM Perimeter Management

While it is easy to say "IAM is the new perimeter," it is important to understand exactly what that means. Within the cloud, or any time a service is offered over a network, attackers can target identities directly. If an attacker compromises an identity, or part of the IAM system, they can breach resources without engaging in a network attack. IAM-based attacks like phishing, scanning for exposed credentials, or credential theft via malware, have increased as we have improved our ability to protect networks, and now represent the single greatest origin of cloud-native breaches.

The IAM perimeter includes both authentication and authorization, all types of entities (users, systems, code, etc.) and extends across federated connections. Stepping back and looking at the IAM system, the perimeter is all of the touch points that are potentially accessible, from phishing a user's password to abusing credentials exposed in a public container definition file.

As described above, any federated authentication action generates a token. This token is tied to a session and has a defined Time To Live (TTL) which corresponds to the session length. IAM systems may integrate the concept of a refresh token which is requested automatically before the session expires, and then extends or creates a new session behind the scenes.

It is important to understand that a token is the product of authentication and can be stolen and abused to provide unapproved access without having to compromise a password. This is a very common attack technique. The attacker can steal a token and, in many cases, even use it from a system located elsewhere under their control until the session expires or the token is manually invalidated.

Securing the IAM perimeter relies on multiple techniques that are split between the IdPand the RP. The objective is to reduce both credential and token compromise and abuse. One core technology to enable this is conditional access which is typically implemented in the cloud using PBAC policies that support conditional statements. Conditional access may be enforced during authentication, authorization, or both.

While the implementation will be technology-specific, there are a few key elements of any IAM perimeter defense strategy:

- Strong authentication (primarily MFA) is a critical first step to defending the IAM perimeter, but that only handles user (and some system) authentication and is still open to abuse.
- Where possible, use cloud-provider managed access credentials that are automatically provisioned, rotated, and deprovisioned. This is commonly supported by all major CSPs for VMs, serverless functions, and other resource types that access other resources or services within the CSP.

- Device and location restrictions during authentication can restrict what devices are allowed and from what network locations. While this might be more difficult to implement for users in a decentralized organization, it can still easily be used for system/service authentication. Even highly decentralized CSCs can consider setting up a VPN/SASE[79].

- PBAC systems often support conditions for originating IP addresses, MFA status, and other restrictions on each authorization request. This is extremely powerful because it can prevent the abuse of stolen tokens since the authorization policy is checked on every API call. Even if the attacker steals a token, that token will not work from an outside location if IP restrictions are implemented on authorization.

- WIth JIT the window for abuse is reduced by eliminating static credentials. Access is requested for a session and externally approved, then revoked at the end of the session. The approval step is a form of dual-authority[80] and is managed out-of-band of the session creation. This is supported by some CSPs and third-party tools. The key benefit is a reduced attack surface: since users do not have perpetual rights, the window of opportunity for attackers is significantly reduced.

- Most IaaS providers support some form of internal service endpoints within their network architectures, and these can be leveraged in IAM policies to ensure system/resource API calls only originate from internal network connections.

- Some PBAC policies support different authorization requirements for the same persona for a given entitlement. For example, a change request could require a stricter set of attributes than read access. This would only allow an administrator to make a network or IAM change from the corporate network, but allow them access to logs from anywhere for debugging purposes.

Managing the IAM perimeter can be complex, but thanks to improving ABAC and PBAC capabilities, we are in a better position to not only manage identity based on a static concept of who or what someone is, but also based on where they are, the devices they are using, and other attributes that are evaluated continually.

Ultimately, PBAC is increasingly favored for cloud-based access management. It allows CSCs to enforce security policies responsive to cloud services' complex and dynamic nature, ensuring that access rights are adequately strict to secure sensitive data yet flexible enough to enable productivity. The CSP's role in supporting these sophisticated access control mechanisms is crucial, enabling the implementation of granular attributes and authorizations that facilitate cloud users' security and operational needs.

5.3.4 Privileged User Management

Imagine something so important to a kingdom – say, the vault that stores gold and silver reserves – that even if the king or queen of the kingdom enters the vault, a scribe is required to record the date and time of entry. This is the starting point of understanding Privileged Identity Management (PIM) and Privileged Access Management (PAM).

[79] CISCO. (2023) *What is Gartner's SASE model, and how will it affect your security stack?*
[80] NIST. (2024) Information Technology Laboratory: Computer Security Resource Center - dual authorization.

PIM and PAM represent important pillars in the secure governance of an organization's IT environment, particularly the management plane. PIM is concerned with overseeing and controlling privileged identities – those users who possess elevated rights to access and modify critical systems or sensitive data. PAM is dedicated to regulating and protecting the channels through which these assets and resources are accessed. It involves deciding who is granted access and the methodologies, timing, and scope of activities comprising that access.

A fundamental principle intrinsic to both PIM and PAM frameworks is JIT access. JIT allocates access rights just for the needed period, while also making sure the access is properly logged, thereby mitigating the risks associated with permanent privileges and the lack of access auditing. If left unchecked, permanent privileged access becomes a vulnerability that can be exploited through account breaches and session hijacking.

The practice of JIT is a practical application of the least privilege principle, ensuring that users are equipped only with the access levels essential to fulfill their immediate job functions. Similarly the principle of separation of duties is applicable in the provision of separate identities or accounts for privileged (administrator) versus non-privileged (regular user) access. For example, privileged access should not be granted to the same identity/account that an individual uses to read their email. Another important application of separation of duties supported by PIM and PAM services is their ability to allow access only when another authorized party (e.g., management) has approved it. Applying these principles consistently trims down exposure and access to sensitive systems and improves security posture.

Integrating PIM and PAM into an enterprise's security framework enhances its defenses. This move strengthens the overall security posture by reducing the likelihood of unauthorized access to critical resources. Minimizing the number of active privileged accounts at any given time narrows the potential attack vectors that cybercriminals could exploit. This integration ensures adherence to regulatory compliance requirements, as it establishes a system where activities associated with privileged accounts are monitored, documented, and made auditable, reinforcing the CSC's control over sensitive operations.

PIM and PAM solutions exhibit several key features for maintaining a secure and compliant IT environment. One of the most significant features is the automated rotation of credentials, which ensures that access cannot be maintained through old or compromised credentials, eliminating a common vulnerability in systems security. Additionally, these solutions enforce MFA. Moreover, they come equipped with comprehensive auditing and reporting tools. These tools are indispensable for conducting detailed forensic analysis and tracking compliance with organizational policies and regulatory standards, providing necessary insights to make informed security decisions.

5.4 IAM Policy Types for Public Cloud

In cloud computing, access control is governed through various policy layers designed to fine-tune permissions and bolster security. The primary types of policies are device-, identity-, resource-, and organization- or tenant-based.

Identity-based policies are those that are associated with an IAM identity. This could be either a federated user, who uses IdPs to gain temporary access to the cloud environment, or an internal (cloud native) IAM identity intrinsic to the CSC. Permissions for this identity are defined by the policy, which

determines allowed or forbidden actions and can be specifically attached to a user, a role, or distributed across a group. Despite the diversity in terminology used by different CSPs, the foundational concept remains consistent: these are permissions affixed to an individual's identity.

Device-based policies are associated with device identity registrations and compliance states. Devices are categorized as managed or unmanaged. Access to sensitive information and resources can be restricted to a specific level of device status and compliance, e.g., the device must have an updated and patched OS version and be registered as an CSC-managed device. Access to less sensitive data and resources can be allowed from a lower compliance state, e.g., access from unmanaged devices.

Resource-based policies diverge from device and identity-based policies in that they are linked directly to the cloud resource, whether an S3 bucket, a Lambda function, or any other service. Such policies regulate who can access the resource and determine other accounts, devices or users' permitted actions on that resource. They manage cross-account interactions and regulate access to Internet-exposed resources, including ensuring that only authorized entities can perform specific actions on a resource.

Organization or tenant-based policies have a much broader reach, encompassing the entire cloud deployment within a CSC account, across a subscription, or throughout a project. These policies are essential for enforcing consistent compliance and security standards across all cloud resources of the CSC. Typically established by cloud administrators, these policies are not subject to modifications by or for individual users or services, thereby upholding a consistent and secure baseline across the deployment.

These policies, each distinct in application and scope, guide the multi-layered access control possible within many cloud services . By embracing a PBAC model, CSPs allow for granular permissions that adhere to the principle of least privilege. This model ensures that users and services possess only the necessary access to complete their tasks, safeguarding against excess permissions that could lead to security attacks.

5.5 Least Privilege & Automation

The principle of least privilege is a foundational principle of security, based on the concept of providing individuals the minimum level of access necessary to perform their duties. Implementing this principle effectively on a large scale can be extraordinarily challenging. Cloud services, particularly IaaS, offer detailed entitlements that, while potentially enhancing the security posture, can also add significant complexity. An entitlement can be understood as a specific rule: *This entity can perform these actions on these resources under these conditions with these attributes.* With a growing number of options, entities, resources, and conditions, managing and predicting these variables becomes complex. This complexity often leads to over-privileging, posing a security risk, or under-privileging, which can impede business operations.

As the scale of cloud-based IAM grows, automation is becoming one of the few feasible strategies for achieving an effective balance of privileges. No single standard for automating IAM privileges exists; the approach depends on the specific technologies in use. However, certain automation methods have been successful in enhancing cloud security:

- **Usage Tracking:** This involves monitoring an entity's activities within a cloud platform over time. The system then analyzes the privileges assigned versus the actual usage. Privileges that are not utilized within a given timeframe are automatically revoked to enhance security.
- **Risk Scoring:** In this method, each entity and action is assigned a risk score based on a range of attributes, such as the IP address or time of the action. These scores are then input into a policy engine that permits or denies actions — not solely based on preset entitlements, but also on whether the risk level is acceptable for a given situation.
- **JIT Privileges:** JIT privileges are requested and granted as needed. Entities use templates to access a predetermined set of privileges for specific resources during a designated time frame, like during a maintenance window. JIT access is granted if it complies with policy constraints, and it may require additional authorization. JIT can be further enhanced when integrated with risk-scoring systems.
- **Continuous Assessment:** Tools such as cloud security posture management (CSPM), or identity-focused software, continually evaluate IAM configurations and actual access patterns within a cloud environment for misconfigurations, unnecessary privileges, and other security lapses. These issues can then be addressed manually or through automated remediation. For instance, a tool might scan through numerous deployments to flag the use of administrative roles without MFA, or identify the presence of unauthorized static access keys.

Tools like Cloud Identity and Entitlement Management may combine and implement these capabilities and even include additional options, such as corrective action.

5.5.1 Identity & Zero Trust

Identity is one of the core elements of any Zero Trust strategy. While there are multiple definitions and models of Zero Trust, every one of them tends to share the following IAM principles:

- Access and connections are identity-aware.
- Identity awareness extends to all entity types, not just humans.
- Attributes are tracked and used to feed decisions.
- Risk scores, based on the entity, attributes, connection, requested action, and resource are used to make decisions based on policies.

In a Zero Trust implementation, for example, a user might be allowed access to webmail from an untrusted system only during certain hours, only with MFA enabled, and only from certain geographies, with attachment downloading disabled. That same user could access attachments only from an official corporate system.

Zero Trust aligns with many of the principles discussed throughout this domain and is not cloud-specific, but is increasingly seen as a primary strategy for enabling access in CSCs moving towards decentralization and cloud computing. Zero Trust can also be a powerful option to improve the implementation of an IAM perimeter.

5.5.2 Customer Identities

Applications hosted in the cloud may have to manage their own identities. Developers have several options to address this need, each with unique advantages and considerations. Directly managing customer identities within an application's own user database is one such option. This approach demands establishing and maintaining a secure and scalable identity store, ensuring that user data is safely handled and can quickly adapt to growing demands.

Alternatively, federation can leverage existing credentials from external IdPs, such as Google, Facebook, or various enterprise SSO systems. This method allows CSCs to use their existing accounts to access services, simplifying the login process and enhancing user convenience. A hybrid approach combines the best of both worlds, offering the flexibility of self-managed identities while supporting federated login methods. This strategy enables a tailored user experience by accommodating diverse user preferences and requirements.

When applications enable CSCs to make direct API calls to the cloud service, securing this access becomes paramount. Implementing secure authentication methods, such as API keys, OAuth tokens, or other mechanisms, is used for controlling access and defining the scope of actions that actors can perform. Ensuring that solid authorization controls are in place is essential for delineating permissions across different access levels, such as read-only capabilities, write capabilities, or full administrative rights, depending on a given user's role.

CSPs, including AWS with its offering of AWS Cognito or Azure with B2C, streamline the management of customer identities. These services provide functionalities like sign-up, sign-in, and access control, simplifying the intricacies of IdM. Third-party identity solutions extend these capabilities further, enhancing user experience, bolstering security features, and facilitating easier integration across multiple platforms.

Summary

IAM is extremely important, with identity taking precedence over the traditional network perimeter as the primary means of managing access to public cloud services and deployments. A core tenet of cloud security is the recognition that the majority of cloud-native breaches are attributable to compromised credentials, emphasizing the importance of robust identity verification measures.

MFA is advocated as an essential requirement for all cloud access. This measure significantly reduces the risk of unauthorized access by requiring multiple forms of verification beyond just a password. Additionally, the implementation of JIT access or other advanced privileged IdM mechanisms is recommended for administrative-level access, thereby ensuring that rights are granted only at the necessary time and only for the duration required.

While CSPs typically offer their own identity pools, there is a strong case for enterprises to adopt federation. Federation allows seamless integration with existing IdPs, enabling users to authenticate using their established credentials from other services, simplifying the user experience and consolidating IdM.

Major CSPs have embraced PBAC, which enables fine-grained control over access permissions. While PBAC provides enhanced security through detailed policy enforcement, it also introduces additional complexity to the IAM framework.

Effective IAM strategies combine secure IdPs with strong authentication protocols. They focus on giving users just enough access they need for their jobs. They use rules based on different situations and types of policies.

When crafting a comprehensive IAM strategy, it is important to document and articulate these practices in detail, spanning various components of cloud architecture. Such documentation should cover the rationale behind the adoption of MFA, the use of JIT for privileged accounts, the benefits of federation over proprietary identity stores, and the intricacies of PBAC systems. It should also delve into the alignment of IAM practices with business objectives, the balance between security measures and user experience, and the continuous evolution of IAM in response to emerging threats and technologies.

Recommendations

Identity Management

- Develop a comprehensive policy, plan, and processes for managing cloud service identities and authorizations.
- Consider using identity brokers to increase governance over identity sources (where appropriate).
- CSPs should offer internal identities and federation using open standards.
- There are no magic protocols: Pick use cases and constraints first, then find the right solution(s).

Access Management

- When connecting to external CSPs, use federation to extend existing IdM if possible. Minimize silos of identities not tied to cloud CSC-provided identities.
- CSCs are responsible for maintaining the IdP and defining identities and attributes based on authoritative sources.
- Cloud users should use MFA for all cloud access and send MFA status as an attribute when using federated authentication.
- Document an entitlement matrix for each cloud deployment that aligns with security and business requirements.
- Translate entitlement matrices into technical policies when supported by the CSP or platform.
- Prefer ABAC and PBAC over RBAC.
- Assess and adopt more modern IAM processes and technologies such as usage tracking for improved least privilege, JIT access, and risk scoring.

Security Measures

- Consider implementing an IAM perimeter with location-based restrictions, especially for sensitive resources or administrative access,to reduce the risk of attacks using stolen credentials or session tokens.
- Eliminate the use of static cloud credentials (like hard-coded API keys) to the greatest degree possible.
- Use automated assessment tools to monitor IAM for misconfigurations, excessive access, compliance failures, and other issues. Consider automated remediation for gross policy violations.
- Log and monitor all IAM changes both at the IdP and the RP.

Incident Response

- Integrate plans and procedures for invalidating or restricting abused IAM session tokens into the incident response program.

Additional Guidance

- Machine Identity in Cybersecurity and IAM | CSA
- What is IAM for the Cloud? | CSA
- Zero Trust Principles and Guidance for Identity and Access | CSA
- Identity and Access Management Glossary | CSA

Domain 6: Security Monitoring

Introduction

This domain presents unique security monitoring challenges and solutions for cloud environments. It emphasizes the distinct aspects of cloud telemetry, management plane logs, service and resource logs, and the integration of advanced monitoring tools. It explores the complexities of hybrid and multi-cloud setups, including interoperability and security considerations. Further highlighted is the critical role of logs, events, and configuration detection in comprehensive security monitoring. Lastly follows the introduction of Generative Artificial Intelligence (GenAI) as an innovative tool for enhancing cloud security and providing a multi-faceted approach to protecting cloud infrastructures.

Learning Objectives

In this domain, you will learn to:

- Identify unique security monitoring challenges in cloud environments.
- Describe the importance of cloud telemetry sources in monitoring cloud environments.
- Analyze collection architectures for security telemetry in cloud environments.
- Recognize monitoring and alerting as foundational components of cloud security.
- Implement detection paths for comprehensive security monitoring.

6.1 Cloud Monitoring

The dynamic nature of cloud infrastructures introduces unique challenges to security monitoring in the cloud. The timing of alerts[81] and logs can vary due to the fast pace of changes within the cloud, and how resources are distributed. Specialized strategies are required. Additionally, the Shared Security Responsibility Model (SSRM) indicates that the cloud service customer (CSC) will be responsible for some aspects of monitoring, while the cloud service provider (CSP) will handle others.

The cloud adds complexity to security monitoring in the following ways:

[81] Alerts are different from events. In the context of security monitoring, events represent the raw data or activities captured within a system or network, while alerts are actionable notifications derived from the analysis of events, signaling potential security threats or incidents that demand attention and response from security teams. Events serve as the input for generating alerts, which help security professionals prioritize and respond to security events effectively and promptly.

1. **Management plane:** The management plane controls all administrative actions, like a captain navigating a ship. The cloud console must be monitored closely because it makes the most critical decisions and grants access to everything in the cloud.

2. **Velocity:** Changes occur in the cloud at a high speed. This rapid pace means security processes must be agile, and automated responses are necessary to keep up with potential threats.

3. **Distribution and segregation:** Cloud resources are spread out and isolated, like compartmentalized sections of a large warehouse. Proper distribution and segregation ensure that a breach in one area does not compromise the entire system. That said, a degree of centralization of the logs is also required to provide an overview of the entirety of the cloud estate.

4. **Cloud sprawl** refers to the widespread proliferation of diverse workload types and the adoption of multiple CSPs within a CSC's cloud environment. This phenomenon of dispersed cloud assets across various platforms and services complicates security monitoring and management. Managing cloud sprawl requires comprehensive strategies that address the complexities of monitoring and securing the diverse range of cloud assets.

On the other hand, cloud computing also creates opportunities for new security monitoring methods. Most CSP service configurations are available to review through simple APIs, creating opportunities for advanced posture management tools that analyze the configuration for insights.

6.1.1 Logs & Events

Logs and events are foundational in security monitoring, compliance, accountability, and the broader context of cloud security and risk management practices. They provide crucial insights into the activities and behaviors occurring within cloud systems, networks, and applications. They are different for each CSP.

Logs provide a relatively complete record of activities (i.e., Create, Read, Update, Delete), are very detailed, and are usually stored persistently. However, the quality of logs can vary by service, and their batched delivery may be delayed. Logs are considered durable, typically saved, and sometimes streamed.

On the other hand, **events** are distinct in that they typically record only changes (i.e., Create, Update, Delete (C-UD)). Specific conditions often trigger security alerts from services like Amazon Web Services (AWS) GuardDuty, Microsoft Sentinel, and Google Cloud Platform (GCP) Security Command Center. Unlike logs, events are ephemeral, meaning they are not retained unless explicitly saved. Events may lack the contextual detail that logs provide, but they are typically speedy, often becoming available within a few seconds of a recorded activity.

Logs provide the depth of data necessary for investigations, while alerts derived from events offer timely notifications crucial for rapid response measures.

Logs and events have an important role in activities such as:

- **Continuous monitoring and risk management:** By monitoring logs and events in real-time or near-real-time, organizations can enhance their ability to detect and respond to security incidents promptly.
- **Detection of anomalies and threats:** Logs and events are evaluated using statistical analysis, machine learning algorithms, or rule-based systems to detect anomalous behavior, such as unusual access patterns, unauthorized changes to configurations, or abnormal network traffic. Anomalous activity is not a definite indicator of an active threat, but rather an early warning of potential issues.
- **Incident response and forensics:** Logs and events provide a detailed trail of activities leading up to and following the incident, aiding in identifying the root cause, scope of impact, and remediation efforts.
- **Compliance and audit requirements:** Many regulatory frameworks mandate collecting and retaining logs for auditing purposes, ensuring accountability and transparency in cloud security practices.
- **Performance and operational insights:** Although not security-focused, monitoring metrics such as resource utilization, network traffic patterns, and application performance indicators can help optimize cloud infrastructure and enhance overall operational efficiency.[82]

6.1.2 Alerts & Monitoring

In the cloud, more attacks can be automated and executed quickly, significantly outpacing traditional detection methods, as its capabilities can accelerate the execution of attacks.[83] This characteristic demands an alert system to monitor the cloud management plane (also known as the console) to quickly identify and respond to threats. A cornerstone of this type of system is the maintenance of comprehensive logs, such as those provided by AWS CloudTrail, Azure Monitor, or GCP Cloud Monitoring, which offers comprehensive visibility into resource, user, API, and network activities within the cloud environment.

Additionally, understanding the various attack vectors that often target cloud environments is essential for designing robust security monitoring strategies. By examining common attack vectors and exploitation techniques in different cloud service models, such as IaaS and PaaS environments, organizations can better comprehend the security risks they face and tailor their monitoring efforts accordingly[84].

[82] In practice, logs can be considered as more permanent records of events, often retained for long periods and archived or stored in centralized repositories for future reference. It is technically possible to store all events and organize them into logs, however, organizations may need to prioritize what data to log based on factors such as storage, performance, and regulatory requirements.
[83] While attacks can be automated and executed quickly regardless of the hosting environment, the characteristics of cloud environments, such as elastic compute resources and automated provisioning, may enable attackers to scale their operations more efficiently, and rapidly perform malicious activities.
[84] CSA. (2023) Understanding Cloud Attack Vectors

6.1.3 Timeliness of Logs & Alerts

One of the key distinctions in cloud security is the pace at which log and event data must be received and processed, and associated alerts be generated. Significant delays in alerts are simply unacceptable, as cloud attacks operate at an accelerated pace, demanding an equally swift response. Additionally, due to the high risk involved in losing control of the management console, alerts must comprehensively cover the management plane to ensure suspicious activity is detected and addressed immediately.

This requires a deep dive into log management strategies, ensuring that logs are collected and analyzed in a manner that prioritizes the timely and accurate detection of emerging attacks.

6.1.4 Monitoring Key Indicators

Key indicators, such as unusual identity and access management (IAM) or network security activities, especially in unused regions, should be closely monitored as they can often signal the preliminary stages of a cyber attack. The MITRE ATT&CK®[85] framework is a resource that can provide context for which indicators relate to which attack tactics, allowing the organization to focus on those attacks most relevant to its environment.

6.2 Cloud Telemetry Sources

Cloud telemetry sources offer visibility into the organization's cloud environments, tracking everything from management actions to individual service interactions and resource performance. They provide the ability to 'see' and 'hear' what is happening in the cloud environment by continuously collecting and sharing detailed information. This information is then processed by security tools, administrators, or automated processes to analyze and understand the health, performance, and security of the CSC's cloud environment. Please reference the figure below for an overview of the cloud telemetry sources which will be elaborated on in the following sections.

[85] MITRE (2024) MITRE ATT&CK®

Management Plane Logs	Service Logs	Resource Logs	Cloud Tools
• Critical source given the importance of protecting the management plane.	• API Gateway: Access logs • Storage: Access logs • Network: VPC Flow logs • Function/Serverless: Activity logs • Cloud load balancer: Activity logs • Cloud DNS: Query logs • Cloud WAF/Firewall: Activity logs	• Workload: Instance, VM logs • Configuration change logs • Cloud function invocation logs • Database transaction logs • Object storage file access logs • Snapshot and image logs (block storage)	• CSPM (Cloud Security Posture Management - SPM) • CASB (Cloud Access Security Broker) • CNAPP (Cloud Native Application Protection Platform) • SSPM (SaaS SPM) • DSPM (Data SPM) • IAM analytics • Cloud detection and response

Figure 27: Cloud Telemetry Sources

6.2.1 Management Plane Logs

Management plane logs are akin to journals that detail the commands and controls executed in the cloud environment. They provide critical insights into how cloud resources are being managed. Analysis of management plane logs provides an organization visibility into who accessed the cloud infrastructure, what actions were performed, and when they occurred. This visibility is important for maintaining governance, compliance, and security in the cloud environment.

6.2.2 Service & Application Logs

Service and application logs act like a diary for individual services and applications, recording every interaction, such as API access and network traffic, which is essential for spotting suspicious activities and forensic investigations. These logs capture a wide range of activities, including user authentication attempts, network traffic, data transfers, and service-specific events. Examining service logs helps CSCs monitor their cloud services' health, performance, and security.

6.2.3 Resource Logs

Resource logs are specialized logs for resources like virtual machines (VMs), databases, and software-defined networks that record every operation and change. These include events, such as resource provisioning, configuration changes, data access and transfers, and system-level activities. Organizations can optimize resource utilization, troubleshoot issues, and detect unauthorized or anomalous behavior affecting individual cloud resources by analyzing resource logs.

6.2.4 Cloud Native Tools

Cloud tools are essential components for interpreting logs and automating responses. They play a pivotal role in interpreting and leveraging the wealth of information contained within cloud telemetry sources. Commonly used cloud tools include Cloud Security Posture Management (CSPM), Cloud Detection and Response (CDR), SaaS Security Posture Management (SSPM), Data Security Posture Management (DSPM), Cloud Workload Protection Platform (CWPP) and Cloud Native Application Protection Platform (CNAPP). These tools offer functionalities such as real-time threat detection, compliance monitoring, configuration management, and incident response automation. Organizations can effectively monitor, analyze, and respond to security events across their cloud environments by integrating cloud tools into their security operations.

Descriptions of a selected set of specialized types of cloud tools designed to address specific aspects of cloud security and compliance management (such as CSPM, CDR, SSPM, DSPM, CWPP, CNAPP) follow below.

- **Cloud Security Posture Management (CSPM)** are tools and practices that help organizations continuously monitor, assess, and improve the security status of their cloud infrastructure. They help identify misconfigurations, compliance violations, and security risks across cloud services and resources. Capabilities offered by CSPM tools include continuous monitoring, automated remediation, and compliance reporting, enabling CSCs to strengthen their overall security posture and adhere to regulatory requirements. This is like reinforcing the locks, alarms, and walls of a fortress while ensuring all defenses are up to the standards set.
- **Cloud Detection and Response (CDR)** are tools designed to detect and respond to security threats and incidents within cloud environments. They leverage advanced analytics, threat intelligence, and possibly machine learning algorithms to identify suspicious activities, anomalous behavior, and indicators of compromise. CDR tools facilitate rapid incident detection, investigation, and response, helping to mitigate the impact of security breaches and unauthorized access attempts in the cloud.
- **SaaS Security Posture Management (SSPM)** are tools that enable organizations to manage and monitor SaaS applications, ensuring proper configuration and entitlements. These tools offer centralized visibility into security controls, configurations, and compliance status across multiple SaaS applications. SSPM tools help assess the effectiveness of SaaS security, enforce security policies, and ensure alignment with contractual obligations and regulatory requirements.
- **Data Security Posture Management (DSPM)** are tools that protect sensitive data and ensure compliance with data protection regulations within cloud environments. They offer capabilities, such as data discovery, classification, encryption policy enforcement, and ensuring proper access controls to safeguard data against unauthorized access, data breaches, and insider threats. DSPM tools help maintain data privacy, integrity, and confidentiality across cloud-based applications, databases, and storage repositories. DSPM tools in cloud environments are like risk management officers for a bank. They ensure that sensitive data is well-protected and that all handling of this

data is in line with strict regulations, much like how a bank would protect its assets and follow financial laws to prevent theft and ensure integrity.

- **Cloud Workload Platform Protection (CWPP)** are tools that provide targeted security for workloads deployed across hybrid cloud architectures. These tools safeguard physical servers, virtual machines, containers, and cloud deployments, regardless of location (on-premises or public cloud). CWPP utilizes continuous monitoring to identify suspicious activity and potential threats, ensuring critical workloads' operational security and integrity.

- **Cloud-Native Application Protection Platform (CNAPP)** are tools that focus on securing cloud applications throughout their lifecycle by providing a unified platform for comprehensive security across the cloud application stack. These tools integrate functionalities like CSPM and CWPP, offering a holistic view of your application security posture. This enables proactive threat detection, vulnerability management, and granular permission controls to safeguard applications. Additionally, CNAPP tools often integrate compliance automation, simplifying adherence to data protection regulations.

6.2.5 Cloud-Native CSP Security Tools & Container Monitoring

CSP security tools and container monitoring are extensions of our security environment. These tools serve distinct but complementary roles.
Cloud-native security tools provided by CSPs, such as AWS Security Hub[86], Azure Defender[87], or GCP Security Command Center[88], are specialized security services embedded within, and integrated into, cloud platforms that provide built-in security intelligence. They function both as telemetry sources and aggregation points, and analyze data to provide this intelligence. These tools are crucial components of cloud telemetry sources, offering insights into various activities, such as user authentication attempts, network traffic, and service-specific events. However, there are limitations as each CSP has its own toolset, which complicates multi-cloud management. Thus, the volume of telemetry data gathered must be balanced with the ability to analyze and correlate it, ensuring the most critical security alerts can be acted upon.

Container monitoring tools integrate with CSP-provided security tools and services, such as AWS Security Hub, Azure Defender, or GCP Security Command Center, to provide comprehensive security coverage for cloud-native applications in the following ways:

- **Data aggregation:** Container monitoring tools often integrate with CSP-provided security tools to aggregate data from multiple sources, including container logs, performance metrics, and security events. This integration allows for centralized visibility into security-related activities across the cloud environment.

- **Correlation and analysis:** By integrating with CSP tools, container monitoring tools can correlate container-specific data with broader security telemetry. This correlation enables more

[86] AWS Security Hub is a CSPM service that performs security checks, aggregates alerts, and enables automated remediation.
[87] Microsoft Defender for Cloud is a CNAPP that is made up of security measures and practices that are designed to protect cloud-based applications from various cyber threats and vulnerabilities.
[88] Security Command Center provides proactive and reactive security for posture management and threat detection for code, identities and data.

accurate threat detection and incident response by identifying patterns or anomalies that may indicate security breaches or vulnerabilities.

- **Automated remediation:** Some container monitoring tools offer integration with CSP-provided automation and orchestration services to automate remediation actions in response to security incidents. For example, if a container is found to be exhibiting suspicious behavior, the monitoring tool can trigger automated actions to isolate the container, block network access, or scale down resources to mitigate the impact of the incident.

Key challenges in container monitoring can be:

- **Data volume:** This entails managing the volume of monitoring data generated by numerous containers running across multiple hosts. Best practices include implementing data aggregation and filtering mechanisms to prioritize critical events and reduce noise, as well as leveraging scalable storage solutions to accommodate large volumes of monitoring data.
- **Visibility across dynamic environments:** Containerized environments are highly dynamic, with containers being created, deployed, and terminated dynamically in response to workload demands. Best practices include ensuring visibility across these dynamic environments by implementing monitoring solutions that can automatically discover and monitor newly created containers and tracking container lifecycle events.
- **Alerts and incident response:** This involves effective alerts and incident response for timely detection and mitigation of security threats in containerized environments. Best practices include setting up alerts based on predefined thresholds or anomaly detection algorithms, establishing incident response procedures to investigate and remediate security incidents promptly, and regularly conducting tabletop exercises or simulations to test incident response readiness.

The following table provides an overview of key features, functionalities, use cases, and benefits of container monitoring, including CWPP and CSPM solutions.

Feature[89]	Container Monitoring Tools	CWPP	CSPM
Focus	Individual containers and containerized applications	Detect vulnerabilities and misconfiguration on cloud workloads, including container and serverless runtime monitoring	Detect vulnerabilities and misconfiguration on the cloud security posture of the cloud management plane
Key Functionalities	• Monitor resource utilization (CPU, memory, network) • Track health and performance • Identify crashes and errors • Basic security features (vulnerability scanning)	• Identify vulnerabilities in container images and environments • Configuration mistakes • Compliance checks • Runtime anomalies	• Monitor cloud security posture • Detect misconfigurations in cloud services • Test configuration against compliance controls

[89] Table 7 provides a high-level comparison between CWPP and CSPM.

	• Container log insights		
Use Cases	• Troubleshoot container issues • Optimize container performance • Maintain application health	• Protect workloads against vulnerabilities • Deploy security policy • Assure detection and remediation of misconfiguration	• Proactively identify and mitigate security risks • Ensure compliance with regulations
Benefits	• Real-time insights into container health and performance • Rapid identification and troubleshooting of container issues	• Real-time insights into workload security • Fast identification of compliance and vulnerabilities • Advance security for containerized applications	• Proactive cloud security management • Mitigate security risks before exploitation • Compliance assurance • Holistic view of cloud security posture

Table 7: High-Level Comparison between CWPP and CSPM

6.2.6 Cloud Telemetry Limitations

Telemetry systems face challenges in securely sharing data across decentralized networks, as they struggle to monitor and trace data flows across disparate and decentralized environments effectively. These limitations highlight the importance of a comprehensive security strategy.

One significant limitation is the inability to capture API calls that are not logged through traditional means, which include monitoring and collecting data in on-premises and non-cloud environments. CSPs frequently update and introduce new APIs, some of which may not be adequately documented or integrated into existing telemetry systems. As a result, these new and/or unlogged API calls may go unnoticed, potentially leading to blind spots in security monitoring and threat detection.

Moreover, the sheer volume of telemetry data generated by cloud environments can overwhelm monitoring tools, leading to delays in distinguishing between legitimate and suspicious behavior. This delay can impact the timely detection and response to potential security threats.

To mitigate telemetry limitations, CSCs must supplement cloud telemetry with other security controls, such as host-based security tools, threat intelligence feeds, and SIEM platforms, to ensure comprehensive coverage and effective threat detection and response capabilities. Adopting a mix of advanced threat detection technologies is essential to counter cloud-based threats effectively. This approach addresses the diverse and evolving nature of threats, including those that are difficult to detect, like insider threats or sophisticated, targeted attacks.

6.3 Collection Architectures

Cloud computing significantly alters how organizations collect their security telemetry. The decentralized nature of cloud deployments, spanning multiple data centers and cloud providers, requires new approaches to telemetry collection. Key factors driving these changes include:

1. **Decentralization:** Unlike traditional centralized data centers, cloud deployments are often spread across various locations and providers, necessitating diverse telemetry collection methods across IaaS, PaaS, and SaaS models.
2. **New Telemetry Sources:** Cloud environments introduce additional telemetry sources, such as cloud management planes, cloud events (often not logged by default), cloud security tool feeds, and various service-specific logs.
3. **Speed Variation:** The varying speeds at which different log sources generate data, coupled with the need for near real-time threat detection and response, complicates log management.
4. **Log Storage and Analysis Options:** Organizations can choose from various log storage and analysis solutions, including security data lakes, to manage their telemetry efficiently.

There is no single correct collection architecture; each provider and technology stack presents unique requirements. This section highlights core principles, various collection options, and major architectural approaches to effectively manage security telemetry in cloud environments.

6.3.1 Log Storage & Retention

Cloud computing introduces new capabilities for storing large amounts of data, and CSPs will typically save logs to their own storage services first. Cloud customers are charged for this storage but also pay data transfer rates when exporting that data to other locations, such as an on-premise SIEM.

An effective and efficient log collection architecture will account for the costs and complexity of moving the logs around. The most cost-effective option is likely to leave the logs in the CSP's storage service, but this could create problems with detection, analysis, and other activities. An organization may then be limited to only using the CSP's analysis tools incompatible with other security monitoring efforts or not meeting performance requirements. Moving logs back on-premise could result in even larger costs in terms of data transfer and physical storage requirements. Third-party SIEM/analysis tools are another option. Security data lakes, which are large pools of storage that accept logs from a wide range of sources and formats, are also possible.

Log retention considerations also play a role in ensuring effective monitoring, troubleshooting, compliance adherence, and the cost of the system. Determining the appropriate log retention period involves balancing operational needs, regulatory requirements, and cost considerations. Retaining logs for an adequate duration allows CSCs to analyze historical data to identify trends, detect security incidents, and troubleshoot system issues. Also, keep in mind the need to move some log events to off-cloud storage for resilience and regulatory purposes. However, prolonged retention periods can lead to increased storage costs and have potential privacy implications. Thus, CSCs must establish clear policies defining which logs to retain, for how long, and with what level of access control.

The following factors will usually guide the decision on where to store logs:

- The default storage location of the CSP
- The default storage costs
- Integration with analysis tools (CSP or third-party, e.g., a SIEM, SOAR[90] or SIEM as a Service)
- Data transfer costs for moving the logs
- Destination storage costs when moving the logs
- Ability to implement effective access controls, which may require providing cloud teams access to their logs, but not the logs of other deployments, to meet operational requirements

6.3.2 Cascading Log Architecture

Cascading log architecture is a hierarchical approach to log management. With it, logs are collected, aggregated, and analyzed in a cascading fashion, flowing from one layer to another to facilitate centralized monitoring and analysis. Cascading log architecture is not inherently specific to hybrid or multi-cloud environments and can be implemented in any environment where there is a need to aggregate and analyze logs from multiple sources across different infrastructure layers. However, cascading logs may be particularly beneficial in hybrid or multi-cloud environments due to the distributed nature of these architectures, and where logs may need to be collected from various on-premises and cloud-based resources.

The following figure presents a sensible architecture for security purposes when managing logs in a cloud environment. Development (Dev), Testing (Test), and Production (Prod) environments each generate their logs that are sent to a centralized log management system for multiple accounts associated with a specific project.[91]

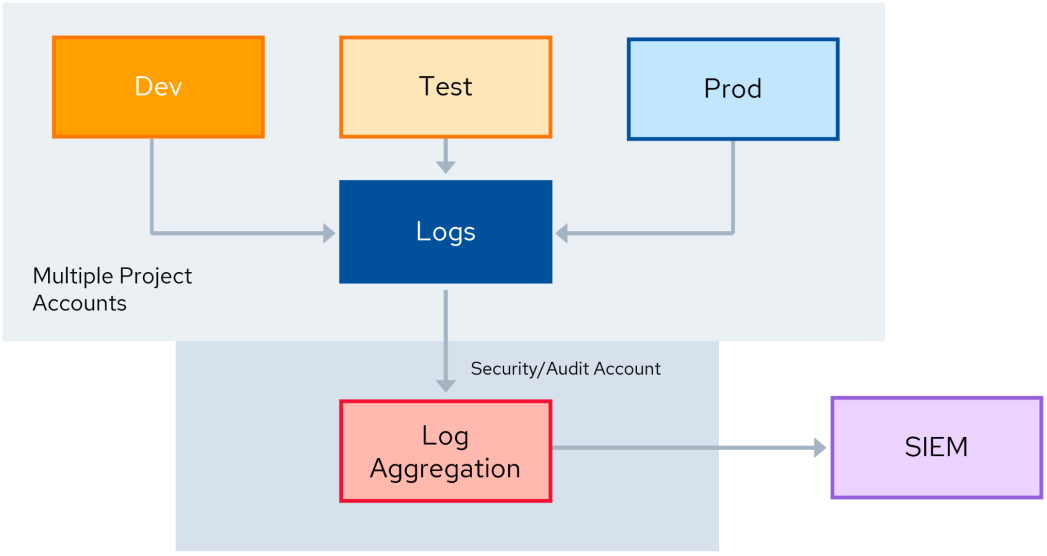


Figure 28: Cascading Log Architecture

[90] SOAR is the acronym for Security orchestration, automation and response.
[91] The CSC's account hierarchy must be considered when designing the cascading log architecture.

Each environment (Dev, Test, Prod) can be configured to forward logs to a central repository. The central log system aggregates these logs and sends the security-relevant ones into a single security/audit environment, ensuring that they are retained securely and consolidated, which is essential for effective security analysis and compliance.

Finally, the aggregated logs can be fed into a data center or cloud SIEM system. The SIEM system analyzes these logs to identify potential security incidents. This architecture provides a view of security-related events across all cloud environments, facilitating timely detection and response to threats.

6.3.3 Cloud Security Monitoring Strategy Guidance

When building a monitoring strategy, understand that consolidating logs in one location is not always optimal, particularly in multi-cloud settings. Instead, a cascading and filtering approach is advised, as shown in the previous section.

This means:

- Constructing distinct pathways for logging and alerts, considering the speed of log generation and the source's importance to the detection effort.
- Forwarding all pertinent alerts and selected logs to the Security Operations Center (SOC) while keeping most raw logs in their localized accounts for cost-efficiency and resource optimization.
- Moving less-utilized logs to lower-cost storage environments can archive the logs for posterity without adding unnecessary scale to the logging system.

The SOC should initially focus on alerts and then delve into the selected logs to verify incidents, respond to them, and proactively hunt for threats.

The choice of commercial tools can influence the feasibility of such an architecture; hence, a balance between log data proximity[92] and the depth of analysis is crucial. This nuanced strategy facilitates effective monitoring and quick incident response, which is essential for maintaining robust cloud security.

6.3.3.1 Log Processing Speeds

Another key concept in understanding monitoring is differentiating log processing speeds within various cloud services. Monitoring and analysis are usually classified into two separate tracks: the slow path and the fast path. The slow path is exclusively for logs and may be subject to delays of up to 15 minutes before being available for analysis. The fast path is usually for events but can also be designed for some logs, which can be analyzed and alerts generated in near real-time.

[92] Log data proximity refers to how physically or logically closely located log data are to the system responsible for collecting and processing those logs. A lesser proximity implies that the log data sources are closer to the central logging system, while a greater proximity indicates that there may be delays or inefficiencies in collecting and processing log data.

This is illustrated here with an AWS security tooling example:

- The "slow path" logs, such as those from CloudTrail[93] or Resource Logs (e.g., S3 access logs, ALB logs), are typically detailed and used for in-depth investigations but may not be available for immediate analysis.
- On the other hand, "fast path" events provided by services like CloudWatch[94] are designed for rapid detection and response. They trigger quick alerts to potential, high-impact issues.
- It is important to understand that these solutions are not mutually exclusive and offer different capabilities, which should be used in tandem for comprehensive security monitoring.
- Fast path logs are essential for immediate security incident responses, while slow-path logs are valuable for thorough post-incident analysis and forensics. Services like GuardDuty[95], Access Analyzer, and Security Hub provide threat intelligence and monitoring, while tools like Detective and Athena[96] aid in analyzing and responding to security events.
- The key is to utilize both paths to ensure timely responses to threats and detailed investigation of security incidents.

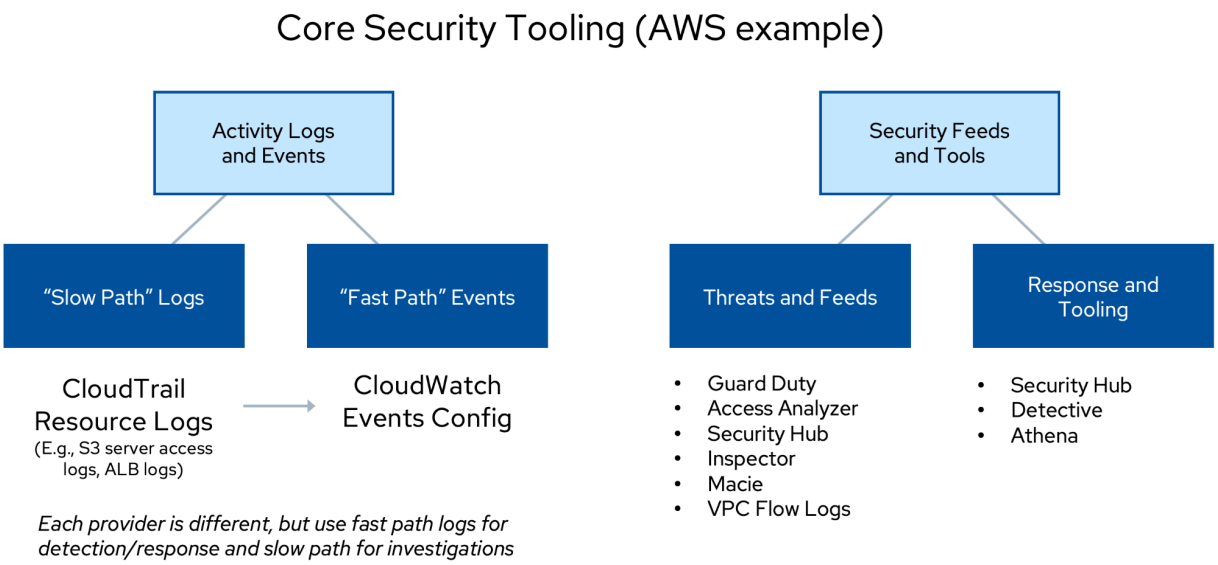


Figure 29: Log processing speeds within various cloud services, illustrated using AWS security tooling

[93] AWS CloudTrail is a service that enables governance, compliance, operational auditing, and risk auditing of AWS accounts.
[94] Amazon CloudWatch is a service that monitors applications, responds to performance changes, optimizes resource use, and provides insights into operational health.
[95] Amazon GuardDuty combines ML and integrated threat intelligence from AWS and leading third parties to help protect your AWS accounts, workloads, and data from threats.
[96] Amazon Athena is an interactive query service that makes it easy to analyze data directly in Amazon Simple Storage Service (Amazon S3) using standard SQL.

6.3.4 Security Data Lake

A security data lake describes a centralized repository designed to handle and analyze the quantities and relevant types of security data gathered from diverse cloud environments and tools. This data architecture should provide scalability to manage large data volumes, including structured and unstructured forms. Its flexibility should support advanced analytics, enabling machine learning and AI to extract insights and identify threats from the security data efficiently. The objective of centralizing and consolidating security data is to improve incident detection, analysis, and response, and to enhance the overall security posture.

Security data lakes provide:

- Improved incident response and forensic analysis
- Access to a comprehensive and historical dataset for response and threat hunting

6.4 Detection & Security Analytics

Logs, events, and configuration detection paths are essential for a comprehensive monitoring system that detects and responds to various security issues.

As described earlier, logs represent the slow path due to their volume and complexity, which requires more analysis time. However, they are comprehensive and can support alerts based on patterns over time, such as repeated login attempts from different IPs. Logs are typically analyzed using third-party or internally developed tools.

Events represent the fast path and may provide near real-time alerts. They might include data similar to logs but focus on C-UD actions[97] within the cloud. Events are often generated by CSP security tools like AWS GuardDuty, Azure Defender, or GCP Security Command Center. Although challenging to capture, they do provide high-value data.

Configuration detection involves identifying vulnerable settings in cloud deployments, which can be discovered within logs, events, or CSPM tools. Properly configured, these can be crucial in identifying misconfigurations and malicious activity. Infrastructure as Code (IaC) can also be a great source of configuration information that can be scanned for vulnerabilities before deployment.

[97]C-UD is an acronym for Create, Update, and Delete, where a bad actor will perform an action like creating an administrative account that is used to create a second administrative account, then change the second account so that MFA is not required to authenticate, then delete the first administrative account created. The goal is to make actions performed on behalf of the bad actor hard to detect or trace through log files.

Logs	Events	Configuration
Slow Path	Fast path	Think of them like "vulnerability alerts"
Higher volume of data	May overlap with logs, but typically C-UD focused	Can pull from logs, events, or CSPM tooling
Supports alerts based on time-series events (e.g., n logins from y IPs in z minutes)	Near real-time	With proper rules, extremely valuable to pick up both configuration errors and bad actors
Handles nearly all possible sources	Many generated from CSP security tools (e.g., GuardDuty/Azure Defender for Cloud)	Can be tougher to get in real-time
Analysis typically in external tool (third party or self deployed)	Tougher to capture, but often higher value/fidelity	Tied to IaC can be even more powerful

Figure 30: Key Detection Paths in Cloud Security: Logs, Events, and Configuration

6.4.1 Comparing Different Tools for Detection

The following table is an example of a comparison of some of the popular tools used in cloud security monitoring and is not an exhaustive list of the following tools capabilities.

Feature	SIEM	CSP Alerts	CSPM
Focus	• Overall IT Infrastructure	• Specific Cloud Issues	• Continuous Cloud Monitoring
Data Sources	• Logs & events from various security sources (e.g., network devices, applications, cloud platforms)	• Alerts generated by a CSPM tool • Pose some challenges, since they need environment-specific filtering • Difficult to aggregate because alerts are cloud-specific and vary widely in format and detail	• Cloud resources, configurations, and activity • Require heavy tuning to ensure effectiveness and accuracy
Functionality	• Aggregates, analyzes, and correlates security data • Typically slow path, but the best option for most log-based analyses	• Provide notifications for potential security issues in the cloud • Need understanding of the source/timing (the alert may only be generated when a	• Monitors for misconfigurations, vulnerabilities, and compliance risks • Some work in parallel with SIEM complementing

		particular rule executes, e.g., the rule runs every 24 hours)	detection and analysis functions • Better at reports than alerts • Cloud Detection and Response Tools
Use Cases	• Investigate security incidents, identify trends, monitor overall security posture	• Proactively address potential cloud security issues	• Continuously improve cloud security posture
Examples	• Security analyst investigates suspicious login attempts identified by SIEM	• The cloud security team receives an alert about an open S3 bucket flagged by CSPM	• CSPM identifies a non-compliant encryption setting on a cloud storage resource
Alerts	• Generates various alerts based on security data correlation	• Specific alerts related to security misconfigurations, vulnerabilities, or compliance violations	• May trigger CSP alerts for identified security issues
Benefits	• Provides a centralized view of security posture across IT infrastructure	• Enables proactive cloud security management	• Helps maintain a secure cloud environment and avoid security incidents

Table 8: Tools Comparison in Cloud Security Monitoring

6.4.2 Security Monitoring & Analysis in Practice

A common approach used in security analytics for detecting and responding to threats is a cascade-and-filter pattern, depicted below. It involves the sequential application of multiple detection mechanisms, or filters, to incoming data streams or logs. Each subsequent filter in the cascade refines and prioritizes the data, focusing on specific criteria or indicators of potential threats. The initial data stream contains various log entries or network traffic data. The following is an example:

- The first filter in the cascade might focus on high-level indicators of suspicious activity, such as anomalous login attempts or unusual network behavior.
- Subsequent filters progressively narrow the scope of the analysis, focusing on specific attributes or patterns associated with known attack vectors or threat actors.
- If a security threat is conclusively detected, the final filter in the cascade might trigger an alert or response action.
- Finally, it is important to leverage advanced analytics techniques, such as machine learning and behavioral analysis, to enhance the effectiveness of detection mechanisms within the cascade.

The following is an example of a cascade-and-filter pattern:

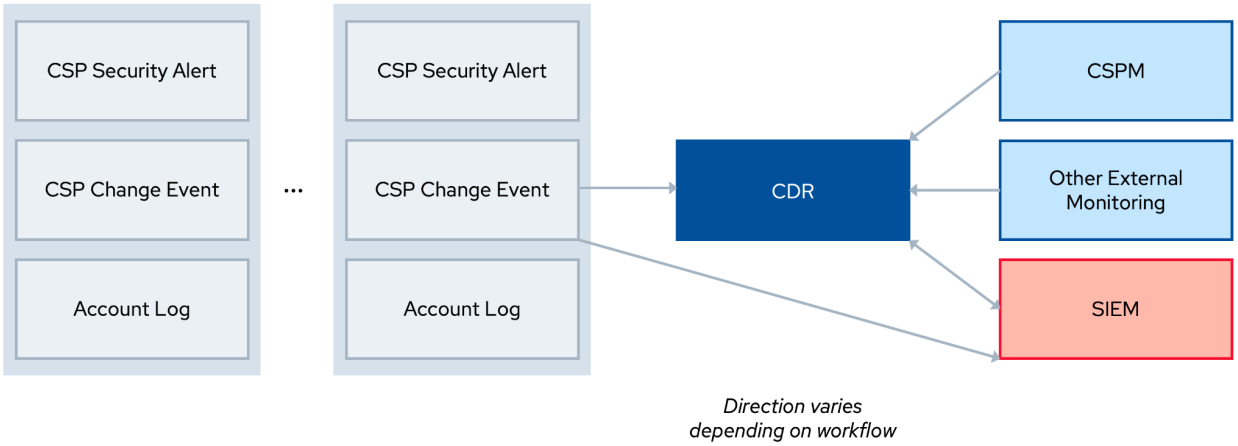


Figure 31: Cascade-and-Filter Pattern Example

In this example, CSP security alerts, change events, and account logs are collected from multiple accounts. These are then filtered and sent to appropriate tools for further action. The CDR pathway is designed for immediate, automated responses to identified threats. CSPM focuses on assessing and improving the security posture, often handling compliance and configuration management. The SIEM system is a comprehensive analysis platform that integrates data from various sources for in-depth examination.

The key is to have a well-organized detection and response strategy that leverages multiple tools and pathways to ensure a secure cloud environment. Understanding how to appropriately filter and channel security information is crucial for effective cloud security operations.

6.4.3 Cloud Detection & Response

While log aggregation is the initial step in monitoring, the CDR process is effective through the rules and handling of events. CDR goes beyond simple log aggregation by incorporating multiple functionalities. It:

- filters the data to eliminate noise and detects potential threats based on defined patterns.
- enriches the information contextually to aid in analysis.
- notifies the necessary personnel or systems.
- may encompass Security Orchestration, Automation, and Response (SOAR) features, providing automated responses and investigative support.

Cloud events are managed in near real-time within CDR, ensuring timely responses to threats. However, integrating a SIEM system with CDR can vary, depending on the tools and configurations employed.

6.4.3.1 Cloud Detection Best Practices

While thinking about best practices for cloud attack detection, remember that Indicators of Compromise (IOC[98]), even for virtual workloads, operate similarly to traditional workloads. So many of the same detections will be applicable.

Priority should be given to specific data sources that are most indicative of security issues, such as:

- logs from the management plane.
- IAM activity.
- changes in public-facing resources.
- structural network modifications.
- cross-account access or subscription peering.
- production configuration changes.

To enhance cloud detection capabilities, external threat intelligence and advanced technologies can be utilized, and the following two methods can be used for the early detection of security threats:

- Integrate threat intelligence feeds into cloud detection systems. This way, an organization can stay updated on emerging threats, attack vectors, and indicators of compromise (IOC). Integrating threat intelligence allows for identifying and mitigating security risks by leveraging external expertise and knowledge.

- Leverage machine learning algorithms and advanced analytics to detect anomalous activities indicative of potential security threats. Machine learning algorithms and advanced analytics to analyze large volumes of data and identify anomalous activities within cloud environments can detect patterns and behaviors that may signify security threats or breaches.

Monitoring development (or non-production) environments can be challenging due to the high volume of data and potential noise, given the prototype nature of these environments. Robust filtering is required to manage the noise. Additionally, an "I intentionally did this" button or similar mechanism is recommended to differentiate between intended and possibly malicious changes, acting as a filter by marking legitimate activities in the cloud, so that time is not wasted validating an authorized change.

6.4.3.2 Example of Detectors (CIS Benchmarks)

Below is a list of specific activities and changes within cloud environments that should be monitored to detect potential security threats. These indicators are based on Center for Internet Security (CIS[99]) benchmarks, and various best practices for securing IT systems and data against attacks.

Access Management:

- Unauthorized API calls

[98] CISA. (2023) *Understanding Indicators of Compromise* - IOCs are the digital and informational "clues" that incident responders use to detect, diagnose, halt, and remediate malicious activity in their networks.
[99] CIS. (2024) *CIS Benchmarks List*.

- Management console login without MFA
- Disabling or scheduling the deletion of a customer-managed key
- Any IAM policy change
- Any use of the root account

Resource Management:

- Cloud storage policy changes
- Configuration monitoring changes
- Security group changes
- Network Access Control List (ACL) changes
- Network gateway changes
- VPC changes (e.g., subnets, routing tables, service endpoints)

Logging & Monitoring:

- Logging service configuration changes
- Management console authentication failures

6.4.4 Advanced Monitoring: Canaries & Honey Tokens

While there can never be a truly proactive status in detecting attacks, there are methods to reduce the time between an attack and its detection significantly. Such methods include "canaries" and "honey tokens," which are decoy credentials or data designed to mimic authentic resources.

Canaries and honey tokens are used to monitor unauthorized access. When attackers interact with these decoys, an alert triggers, indicating an attempted or actual breach. For example, canaries may be placed in credential stores to appear as legitimate user data. Honey tokens can be deployed in various locations that might attract attackers, such as databases or documents.

6.5 Generative AI for Security Monitoring

Generative Artificial Intelligence (GenAI) presents significant potential to dramatically improve the efficiency and effectiveness of SOCs, such as automating log data analysis, scaling processing data, and enhancing the accuracy of malicious activity identification.

Additionally, GenAI aids in creating simulated attack scenarios, which serve as a form of robust vulnerability testing. It improves the relevance and precision of alerts by adapting to observed network behavior patterns and augments log data with extensive context. This helps analysts, especially less experienced ones, understand complex security events and the pathways attacks might take.

Moreover, GenAI provides workflow suggestions, offering guidance in validating and responding to alerts, which is critical for maintaining an effective security posture. This AI-driven approach is increasingly being leveraged by cloud security tooling and is becoming integral to cloud security infrastructures, empowering teams to better anticipate, identify, and respond to threats.

The following is a list of some items that GenAI has the potential to influence:

- Enhance real-time threat detection through predictive analytics models
- Automate and scale log analysis, finding potential malicious activity, and improving efficiency and accuracy
- Generate simulated attacks for robust vulnerability testing
- Improve alerts over time, based on learned network behavior patterns, thereby reducing alert fatigue
- Enrich logs and add extensive context to various security logs/events
- Assist junior analysts in making sense of attack paths
- Provide workflow suggestions to guide analysts in validating and responding to alerts
- Generate synthetic data for testing and training that mimics real-world scenarios, which makes it valuable for testing security models without exposing sensitive information

GenAI-based security solutions have shown a pace of innovation not seen before in the security industry. Expect continued rapid innovation and capabilities to transform how SOCs work.

6.5.1 Challenges & Considerations of GenAI

While GenAI holds great promise in enhancing the detection and resolution of cloud attacks, it is crucial to acknowledge the potential challenges and ethical dilemmas it presents. Continuous data and training is required to keep AI models relevant and effective. However, this necessity for large data sets can raise privacy and data protection concerns, particularly when large language models (LLMs) incorporate training data into their responses.

Furthermore, the rapid advancement in AI capabilities demands scalable security solutions. Differentiating between legitimate and AI-generated activities is also an increased challenge, which could mislead security monitoring efforts and create more noise in the system.

One of the most pressing concerns is adversarial AI, a sophisticated AI system designed to evade or deceive security mechanisms. Equally important are the ethical considerations surrounding AI's involvement in human surveillance and monitoring activities, necessitating alignment with privacy standards and regulations.

It is critical to take a balanced and thoughtful approach to adopting new, groundbreaking technologies like AI. However, it seems that AI's integration into security operations is inevitable, requiring practitioners to adapt to maintain robust and ethical security practices.

To learn more about GenAI, refer to the Cloud Security Alliance training *Introduction to Generative AI & Prompt Engineering*[100] and the work being done by the Cloud Security Alliance's AI Safety Initiative.[101]

[100] CSA, (2023) *Introduction to Generative AI & Prompt Engineering.*
[101] CSA. (2024) AI Safety Initiative.

Summary

This domain addresses unique cloud security monitoring challenges, emphasizing cloud telemetry, management plane logs, service/resource logs, and advanced tools. It covers hybrid/multi-cloud complexities, the critical role of logs/events, and the innovative use of Generative Artificial Intelligence (GenAI).

Telemetry offers visibility into cloud environments, tracking actions and resource performance. However, it faces challenges like blind spots in API call logging and data volume management. Effective strategies include supplementing telemetry with other security controls and using advanced threat detection technologies.

A cascade-and-filter approach is advised for log management, balancing cost and resource efficiency. Focus on forwarding pertinent alerts to the SOC and archiving less-used logs cost-effectively. Differentiate log processing speeds for timely threat response (fast path) and in-depth analysis (slow path).

A centralized repository for handling and analyzing large security data volumes, supporting advanced analytics to enhance incident detection and response.

Logs (slow path) and events (fast path) are essential for comprehensive monitoring. Advanced monitoring methods, like canaries and honey tokens, reduce detection time. Key activities to monitor include unauthorized API calls, IAM policy changes, and network configuration changes.

GenAI enhances security by automating log analysis, generating simulated attacks, improving alert precision, and assisting analysts. Challenges include maintaining privacy, managing data, and addressing adversarial AI risks.

Recommendations

Monitoring and alerts are a foundational component of cloud security. It is important to:

- focus on rapid detection due to automated cloud attacks.
- monitor the management plane.
- leverage a combination of log management and alert-timing strategies (e.g., utilize slow path logs for response and forensics, and alerts on fast path events to detect high-risk activity, including the management plane).
- monitor telemetry and cloud tools utilization (e.g., management plane logs, service logs, and the application of CSPM, CASB, and CNAPP) for enhanced security monitoring.
- strategically collect and analyze logs (e.g., consider deploying a cascading log architecture and selective alerts to manage costs and enhance SOC efficiency in multi-cloud environments).

- deploy canaries and honey tokens to provide deterministic alerts without false positives and generative AI provides a potential path to improve threat detection and response efficiency.

Additional Guidance

- Understanding Cloud Attack Vectors | CSA
- AI Safety Initiative | CSA
- MITRE ATT&CK® Cloud Matrix

Domain 7: Infrastructure & Networking

Introduction

This domain covers managing the overall infrastructure footprint and network security. It also includes a small section on the cloud service provider's (CSP's) infrastructure security responsibilities. Infrastructure in Infrastructure as a Service (IaaS) refers to the compute, network, and storage resource pools residing in the cloud.

In previous versions of the CSA Security Guidance and the Certificate of Cloud Security Knowledge (CCSK) training[102], we included a deeper discussion of the infrastructure used to build and host a cloud service, either public or private. Due to the extensive evolution of underlying technologies, that content is beyond the scope of this training, which is geared towards security professionals working for cloud service customers (CSCs), not CSPs. Other domains cover compute (workloads) and storage (data) security concerns.

While much of infrastructure security is covered in the workload, data, and network sections, there are some higher-level capabilities that span the full range of cloud infrastructure options. These include:

- core security techniques, like shifting left, guardrails, and monitoring.
- secure architecture, including the well-architected frameworks.
- Infrastructure as Code (IaC).
- different cloud migration strategies (e.g., lift and shift).

This domain focuses on network security. It starts with the concepts of Software Defined Networks (SDN), which are used in every IaaS platform. Diving into security groups and beyond, as well as container networking. It then covers different connectivity options, such as connecting to a CSC's data center (hybrid) and workloads. The domain finishes with a Zero Trust Architectures (ZTA) and SASE discussion. SASE frameworks are rapidly becoming dominant models for implementing cloud networking and are largely driven by security requirements.

Learning Objectives

In this domain, you will learn to:

- Understand the areas and techniques used in securing cloud infrastructures.

102

- Understand cloud network fundamentals.
- Manage container networking.
- Manage cloud network security and design secure architectures.
- Apply Zero Trust techniques to securing cloud infrastructure and networks.
- Techniques used to manage security for a Secure Access Service Edge (SASE).

7.1 Cloud Infrastructure Security

Cloud infrastructure refers to the hardware, firmware and software components, such as servers, storage, networking, and virtualization tools, needed to support the delivery of cloud computing services and resources over the Internet. It enables organizations to build, deploy, and manage applications and data in a scalable, flexible, and cost-effective manner. It is up to CSCs to design and build their architecture on the secure services CSPs offer. Since so much of IaaS and Platform as a Service (PaaS) design relies on the CSC, it is important to understand the proper use of the infrastructure and how to construct well-architected implementations that help achieve the benefits of the cloud.

7.1.1 Secure Architecture: Well-Architected Pillars

One architect approach, supported by cloud providers in slightly different ways, is to follow the principles of the Well-Architected Framework.[103] When followed, these pillars guide design and implementation decisions to improve customer outcomes (e.g., security and costs) when using a cloud.

The Well Architected Framework has six pillars:

Figure 32: Key Pillars of a Well-Architected Cloud Framework

Pillar 1: Security

- Protect information, systems, and assets while delivering business value
- Apply security at all architectural layers
- Automate security best practices, enable traceability, and manage access control

Pillar 2: Operational Excellence

- Focus on running and monitoring systems to deliver business value
- Continuously improve processes and procedures

[103] AWS. (2024) AWS Well-Architected.

- Automate changes, respond to events, and define standards to manage daily operations

Pillar 3: Reliability

- Ensure a workload performs its intended function correctly and consistently
- Recover quickly from failures to meet business and customer demand
- Test recovery procedures, scale horizontally to increase availability, and automatically recover from failure

Pillar 4: Performance Efficiency

- Use computing resources efficiently to meet system requirements
- Maintain efficiency as demand changes and technologies evolve
- Experiment more often, use serverless architectures, and design systems to be able to go global in minutes

Pillar 5: Cost Optimization

- Run systems to deliver business value at the lowest price point
- Use cost-effective resources, match supply with demand, and increase expenditure awareness
- Optimize over time by measuring, monitoring, and improving resource utilization

Pillar 6: Sustainability

- Minimizing the environmental impacts of running cloud workloads
- Understand impact, and maximize utilization to minimize required resources and reduce downstream impacts

These pillars assist in the development of a consistent approach to evaluating architectures and implementing scalable, secure, and efficient designs. This allows CSCs to focus on delivering business value through their applications and services.

7.1.2 Foundational Infrastructure Security Techniques

The following four foundational techniques are important to consider when creating and maintaining secure infrastructure.

- **Secure architecture:** This starts with designing cloud infrastructure with security as a key principle. It includes properly segregating resources and networks, implementing least privilege access, and ensuring secure storage, communications, and service configurations. A landing zone and baseline configurations should be established to ensure consistent security across all environments. IaC tools can automate the deployment of secure architectures and reduce the risk of manual misconfigurations.

- **Secure deployment and configuration:** This involves configuring and/or deploying resources and services, and hardening all cloud infrastructure components, including virtual machines (VMs), containers, storage, and networking. It includes applying security benchmarks and best

practices, such as Center for Internet Security (CIS) benchmarks,[104] to ensure proper configuration of cloud assets .

- **Shifting security left**: This means embedding security controls and testing early in the development lifecycle rather than treating it as an afterthought. It includes implementing code analysis tools, automated security testing, and Continuous Integration/Continuous Deployment (CI/CD) pipeline security gates. Developers should be trained on secure IaC coding practices, and be provided with tools and frameworks to build security into applications.
- **Continuous monitoring and guardrails[105]:** These are for maintaining security. They involve using automated systems to monitor cloud environments and enforce policies. It includes using Cloud Security Posture Management (CSPM) or Cloud-Native Application Protection Platform (CNAPP) tools for logging, monitoring, and ongoing assessments, as well as implementing AWS Config rules, Service Control Policies, or Azure Policies to enforce policies and prevent deviations from established standards. Regular security audits and penetration testing verify the effectiveness of these measures.

7.1.3 CSP Infrastructure Security Responsibilities

Infrastructure security starts with the CSP, which ensures a secure platform for CSCs to build on. Under the shared security responsibility model (SSRM), infrastructure security is primarily the CSP's responsibility. CSP infrastructure security responsibilities include the following:

- **Facilities**: The CSP is responsible for ensuring the physical security of the facilities where the cloud infrastructure is housed. This includes measures like access control, surveillance, and environmental protection.
- **Employees**: The CSP screens, trains, and manages employees with access to cloud infrastructure, helping maintain organization integrity and trustworthiness.
- **Physical network, storage, and compute**: The CSP secures and maintains the underlying physical components of the cloud infrastructure, such as servers, storage devices, and networking equipment.
- **Virtualization layers**: The CSP is responsible for securing the virtualization technology that enables the creation and isolation of VMs and containers running on the physical infrastructure.
- **Management plane**: The CSP secures and controls access to the web-based interfaces and API endpoints that customers use to manage their cloud resources and services.
- **PaaS and SaaS services**: The CSP offers higher-level platforms and software services that handle the security of the underlying infrastructure and applications based on the SSRM.

[104] CIS. (2024) *Foundational Cloud Security with CIS Benchmarks*.
[105] Guardrails are preventative and reactive controls that either block an undesired outcome (e.g., block use of regions, public object storage, or specific cloud services that aren't approved) or auto-remediate or correct a policy violation.

In summary, the CSP secures the physical facilities, hardware, virtualization layer, and management interfaces that comprise the cloud infrastructure. CSCs focus on securing what they consume and deploy on that infrastructure. The following figure outlines the layered components of cloud service models (IaaS, PaaS, and SaaS), highlighting the various elements and their integration to provide cloud services.

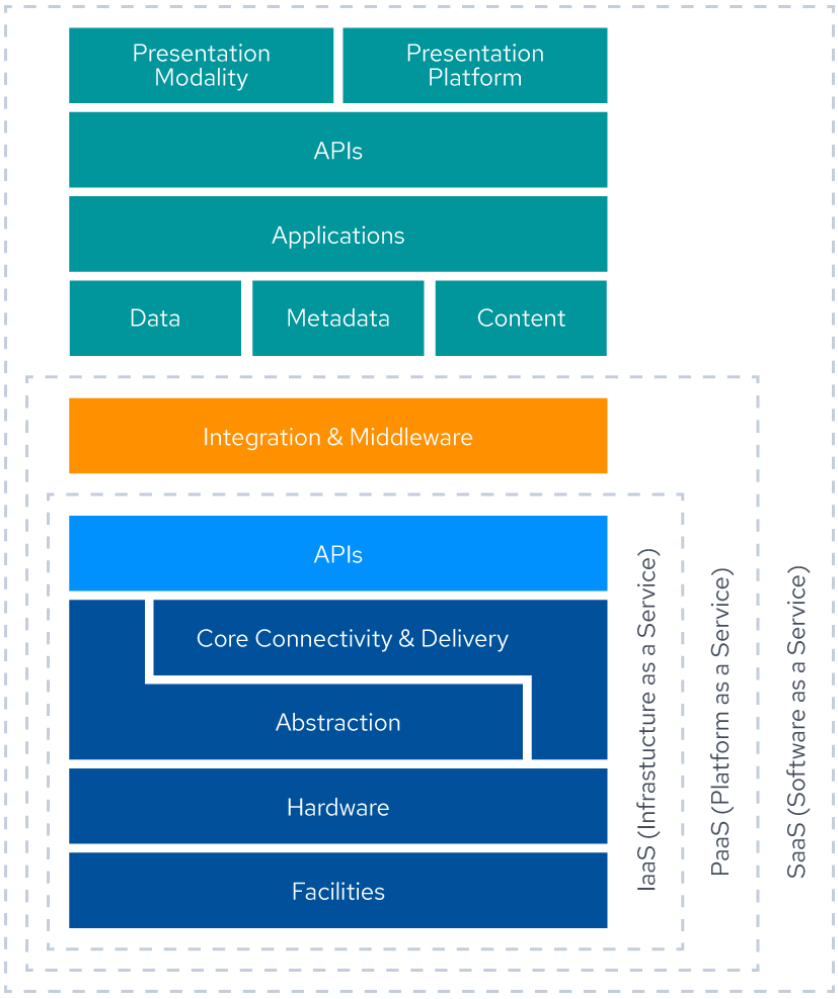


Figure 33: Layered Components of Cloud Service Models: IaaS, PaaS, and SaaS

7.1.4 Infrastructure as Code

IaC is defined in NIST SP 800-172 as "The process of managing and provisioning an organization's IT infrastructure using machine-readable configuration files, rather than employing physical hardware configuration or interactive configuration tools."[106] IaC is the dominant model for deploying cloud resources and is supported by every major provider. Key IaC concepts include defining architectures in a machine-readable format, from low-level network design to high-level application components, and typically deploying them using automated CI/CD pipelines. Security scanning for misconfigurations can

[106] NIST (2024) Glossary - infrastructure as code (IaC)

be integrated into the pipeline, with full version and control change tracking, ensuring consistent and secure deployments. This practice, known as Shift Left infrastructure security, embeds security early in the development process. IaC is discussed in multiple areas of this training, including the sections on application and workload security.

IaC offers several security benefits listed below.

1. **Consistency and standardization:**

    - IaC allows for defining and enforcing secure configurations consistently across all environments.
    - Security best practices, such as least privilege access, can be codified into the IaC templates.
    - Can reduce the risk of misconfigurations and ensure a standardized security posture.

2. **Version control and auditability:**

    - Infrastructure code files can be stored in version control systems, providing a complete history of infrastructure changes.
    - Changes can be tracked, reviewed, and audited, enabling better visibility and accountability.
    - Can facilitate collaboration and peer review of infrastructure code to identify and address security issues.

3. **Automated security testing:**

    - Security scanning and testing can be integrated into the deployment pipeline.
    - Automated tools can validate the security of the infrastructure code before deployment.
    - Can catch security issues early in the development process, reducing the risk of vulnerabilities in production.

4. **Rapid and secure deployments:**

    - IaC enables fast and repeatable deployments of infrastructure, reducing the time and effort required.
    - Security controls can be automatically applied during deployment, ensuring consistent protection.
    - Can enable quick response to security incidents by allowing rapid redeployment of secure infrastructure.

5. **Scalability and flexibility:**

    - IaC supports the dynamic scaling and provisioning of resources, based on demand.
    - Security policies and controls can be automatically applied to new resources as they are created.
    - Can enable maintaining security in highly dynamic and distributed cloud environments.

By leveraging IaC, CSCs can embed security into the foundation of their cloud infrastructure. It provides a way to define, deploy, and manage secure and compliant infrastructure at scale. IaC helps shift security left, catch issues early, and ensure a consistently secure environment throughout the development lifecycle.

7.1.5 Cloud Migration Architecture & Security Implications

While some cloud deployments are completely new, in IaaS many deployments are due to migrations from data centers or even other providers. Migration is not necessarily a simple process when moving between two entirely different types of infrastructure with different available security capabilities. There are different models of migration, each with its own security and cost tradeoffs. The following guidance considerations and approaches apply to the security and architecture of cloud migration initiatives.

A clear definition of requirements and a thorough assessment of the current security posture should guide the migration approach and implementation. Organizations might need to use a combination of approaches to cloud migration. The different tactics utilized depend on each application's specific needs and risks. Generally, an organization can re-architect/rebuild, refactor, or rehost existing applications.

Figure 34: Cloud Migration Strategies: Re-architect, Refactor, Rehost

7.1.5.1 Re-Architecting/Rebuilding

When applications are completely re-architected or rebuilt from scratch to be cloud native, it is the most time and resource-intensive approach that maximizes the benefits of the cloud. This approach allows for *security by design*, incorporating security controls and practices throughout the development process. By rebuilding the application, it can fully leverage cloud-native security features and automation. However, this requires significant changes to security processes, tools, and staff skills. It is essential to ensure secure design, configuration, and testing of the rebuilt application.

7.1.5.2 Refactoring

When applications are modified and optimized to leverage cloud-native services and features as much as possible, it is more time consuming than rehosting, but allows for improved performance, scalability, and resilience. It requires updating security policies, procedures, and staff skills for the refactored application.

This approach provides an opportunity to integrate security best practices and controls into the refactored application and leverage CSP security services such as IAM, encryption, and logging. However, if not properly architected and configured, it may introduce new risks.

7.1.5.3 Rehosting (Lift & Shift)

When applications are moved to the cloud with minimal changes, retaining the existing architecture, it is the fastest migration approach but the least optimized for the cloud. In terms of security considerations, existing security controls and issues may not effectively transfer to the cloud due to architectural differences. Additionally, existing security controls may not fully utilize cloud native security features and automation. This approach requires adapting security processes and tools for cloud monitoring and incident response.

7.2 Cloud Network Fundamentals

Cloud networks are SDNs. Enforcing strong separation between customer tenant environments is key. SDNs have emerged as a key technology revolutionizing network design, management, and operation. SDN decouples the network control plane from the data plane, allowing the network to be programmatically configured and controlled through software. The control plane manages routing, network/subnet definitions, and similar while the data plane moves the network traffic between resources and networks. This shift from traditional hardware-based networking to software defined approaches brings numerous benefits to cloud environments.

The following figure demonstrates the network control logic in a SDN environment, illustrating the encapsulation and unwrapping of packets as they traverse between physical hosts through network control logic.

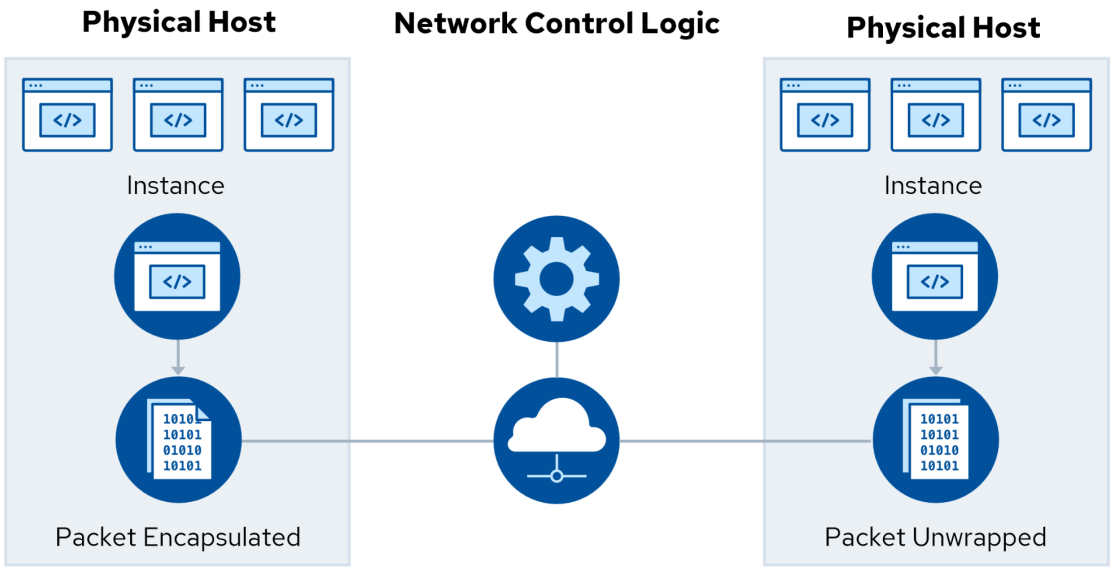


Figure 35: Network Control Logic

One of the primary advantages of SDN is its enhanced flexibility and agility. With SDN, network administrators can dynamically configure and manage network resources through software, enabling rapid provisioning and modification of network services. This flexibility allows CSPs to quickly respond to changing CSC requirements, scale network resources on demand, and optimize network performance based on real-time traffic patterns. SDN also facilitates the implementation of network virtualization, enabling the creation of multiple logical networks on top of a shared physical infrastructure. This enables better utilization of network resources, improved network segmentation, and easier management of multi-tenant environments. SDN can be used on any network, but is the default on all IaaS platforms.

SDN simplifies network operations and management. By centralizing network control and providing a unified view of the network, SDN reduces the complexity of managing large-scale cloud networks. Network administrators can use SDN controllers and APIs to automate network configuration tasks, monitor network performance, and troubleshoot issues more efficiently. SDN also enables the integration of network services with cloud orchestration platforms, allowing for seamless provisioning and management of network resources alongside compute and storage resources. This integration streamlines the deployment and operation of cloud applications, improving overall efficiency and reducing operational costs.

7.2.1 Security Benefits of SDN

The SDNs common to major cloud computing platforms, while different, tend to support a core set of powerful security benefits.

- Networks are either default deny or can quickly be configured as such. This means the networking fabric will not carry packets unless there is a defined route and specific destination, and the ports/protocols are approved by the security groups. (A security group is a rule in the network to allow or discard traffic.) This reduces or eliminates common network attack techniques like port scanning or sniffing.

- Policy-based management means the networks are managed by configuration policies, not by configuring disparate technologies. This improves consistency and control.

- Granular segmentation is far easier to implement than on a physical network since it is managed in the SDN control plane and requires no physical configuration. This is incredibly flexible and powerful, and makes it easy to deploy only the network components (e.g., subnets) required for a given application.

- Security groups, and in some cases, other security capabilities, are built into the fabric of the network. No firewall is required to maintain them .

7.2.2 Minimum Viable Network

SDN capabilities enable a concept called the Minimum Viable Network (MVN). In a MVN, only the network components required for minimum connectivity are deployed, and each layer in the architecture only allows the absolute minimum routes, ports, and protocols required for the application. This is enforceable

per resource and inherent to the network design, enabling and supporting micro-segmentation. Thus, the Internet can only communicate to the load balancer on the HTTPS port (443). The web server will only accept connections from the specified load balancer and also only on port 443. The application server will only allow inbound connections from the web server on expected ports. The database server only accepts connections to the application server's approved ports. All of this is enforced natively in the network fabric without any need for additional security tools. There is, for example, no route for an attacker to compromise the database (outside of an application vulnerability), nor is there any way to connect back to command-and-control infrastructure if similar rules apply to outbound traffic.

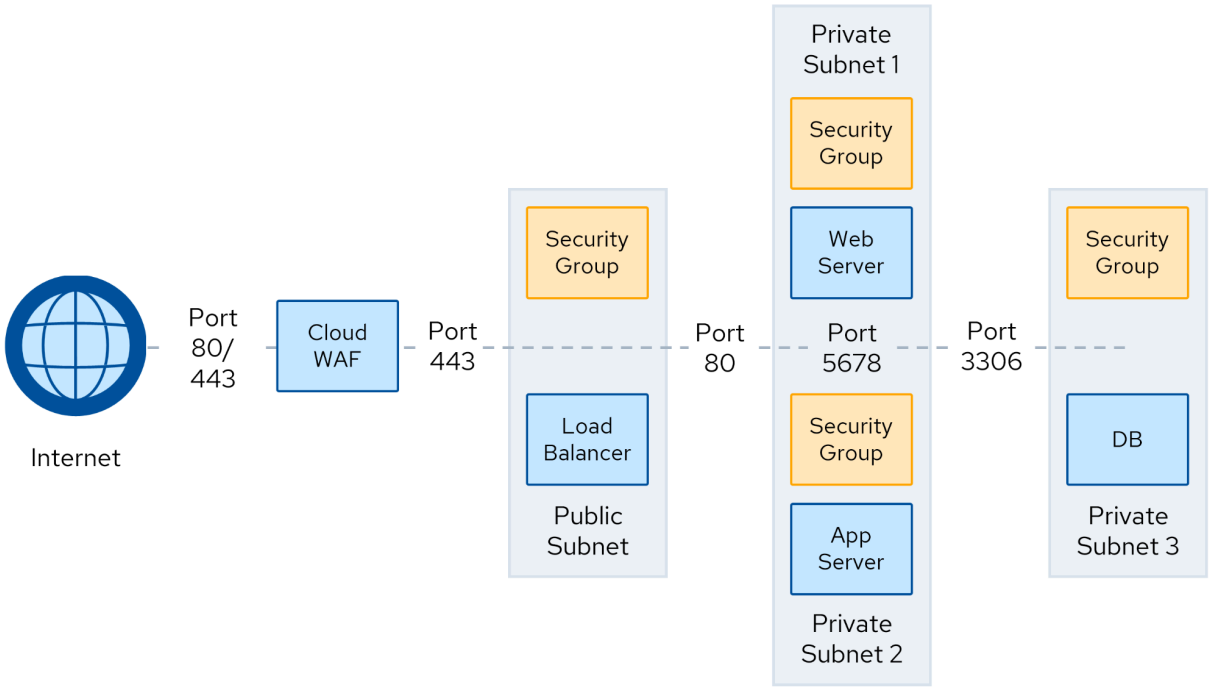


Figure 36: Secure Layered Architecture in MVN

In the example above, the Internet traffic is first received by the cloud Web Application Firewall (WAF), which acts as the primary entry point. The WAF accepts all incoming traffic and filters out malicious requests before forwarding legitimate traffic to the load balancer. The load balancer, in turn, only accepts traffic from the WAF and distributes it to the web server tier. Each subsequent layer (web server, application server, and database) only accepts traffic from the layer directly above it, creating a tightly controlled, circuit switched-like network.

The port numbers in the figure represent the specific ports allowed for communication between the layers. For instance, the web server may accept traffic on port 80 (HTTP) and 443 (HTTPS) from the load balancer, while the application server may accept traffic on a specific port (e.g., 5678) from the web server. The database, located in a separate private subnet, only accepts traffic from the application server on the designated port. All other traffic not explicitly allowed by the security group is dropped by default.

This MVN architecture makes it exceptionally difficult for attackers to penetrate the system, as they are limited in their ability to explore the network and directly access internal components. Attackers are

essentially confined to targeting application-layer vulnerabilities, as the network ports are not exposed for direct exploitation. To compromise the system, an attacker would need to successfully breach the cloud WAF, load balancer, and each subsequent layer, which is a significant challenge given the limited attack surface and the inherent security controls provided by the cloud platform.

7.2.3 Common SDN-Based Components

Most cloud networks share a consistent set of foundational components. The following are some common SDN-based components for the cloud.

Virtual Networks/Virtual Private Clouds:[107]

- Some providers use Virtual Networks (VNets), and others use Virtual Private Clouds (VPCs)
- Logically isolates virtual networks within a cloud environment
- Enables CSCs to define their IP address range and network topology
- Provides a secure and private networking environment for cloud resources

Subnets (Public & Private):

- Smaller network segments within a VNet/VPC
- Allows for further segmentation and organization of resources
- Enables application of different security and access control policies

Route tables:

- Define how network traffic is directed within a VNet/VPC
- Specify the paths for traffic between subnets and external networks
- Enable custom routing configurations for optimal network performance

Cloud Network Security Groups:

- Security groups are similar to a stateful firewall, but are implemented within the network fabric itself
- Act as virtual firewalls at the network interface, instance, or subnet level
- Control inbound and outbound traffic based on IP addresses, ports, protocols, and other criteria
- Provide a granular level of security for individual resources or groups of resources

Cloud Network Access Control Lists (NACLs):

- By specifying which packets can pass through network devices and environments, ACLs control both inbound and outbound traffic
- ACLs work lower in the network stack than Security Groups, and are typically stateless[108]

[107] VNet is a term commonly used by Microsoft Azure, whereas VPC is used by Amazon Web Services (AWS) and Google Cloud Platform (GCP). Both terms refer to the network environment residing in the cloud, controlled by software, and able to replace the on-premises data center or network infrastructure.
[108] A stateful firewall is a firewall that maintains a "state" or stores information about active network connections.

- Security groups most-often apply to resources (e.g., instances) while ACLs apply to subnets/networks
- Implementations of both are different on various CSPs

Network Address Translation (NAT) gateways:

- Enable instances in private subnets to access the Internet or other external services
- Translate private IP addresses to public IP addresses for outbound traffic
- Provide a layer of security by hiding internal IP addresses from the public Internet

Internet gateways:

- Serve as the entry and exit point for Internet traffic in a VNet/VPC
- Allow resources within the VNet/VPC to communicate with the public Internet
- Enable inbound and outbound Internet connectivity for cloud resources

Hybrid leased lines:

- Dedicated, private network connections between on-premises infrastructure and the cloud
- Provide high-bandwidth, low-latency, and secure connectivity for hybrid cloud environments
- Enable seamless integration of on-premises and cloud resources
- Examples include multiprotocol label switching, AWS DirectConnect, or Azure ExpressRoute

VPN gateways:

- Establish secure, encrypted connections between on-premises networks and the cloud
- Allow remote users or offices to access cloud resources securely
- Provide a cost-effective alternative to dedicated leased lines for smaller-scale connectivity

Service endpoints:

- Enable private connectivity between a VNet/VPC and other cloud services
- Allow resources within the VNet/VPC to access cloud services without traversing the public Internet
- Enhance security by keeping traffic within the CSP's network

Peering/transit connections:

- Establish direct, private connections between VNets/VPCs within the same cloud region
- Enable resources in different VNets/VPCs to communicate with each other
- Provide a cost-effective and low-latency option for inter-VPC communication

These elements work together to create a robust and secure software-defined networking environment in the cloud, enabling customers to build scalable, flexible, and customizable network architectures.

7.2.4 Cloud Network Security Groups

Within the SDN components, the cloud network security groups are fundamental to securing resources within a cloud environment. They are the virtual firewalls that control inbound and outbound traffic at the instance, VM, or subnet level, providing a granular level of security for individual resources or groups of resources. Security groups are leveraged by defining rules that allow or deny traffic based on various parameters, such as IP addresses, ports, and protocols. Some providers, like AWS, default to deny all policy, and the CSC must create allow rules. Azure, on the other hand, defaults to a permissive policy, and the CSC can create deny rules. Security groups can also internally reference other security groups, eliminating the need to define IP addresses in rules.

One of the key principles of security groups is defining rules in a policy. Administrators create security group policies that contain a set of rules governing network traffic. These rules can be configured to allow or deny specific types of traffic, such as Secure Shell (SSH), Remote Desktop Protocol (RDP), or HTTP/HTTPS. By carefully crafting these rules, administrators can implement the principle of least privilege, granting only the necessary permissions for resources to function properly while blocking all other traffic.

Once a security group policy is defined, it can be applied to specific resources within the cloud environment. Applying policies to resources ensures that the desired security measures are enforced consistently across the infrastructure. Security groups can be associated with individual instances, network interfaces, or entire subnets (depending on the CSP), providing flexibility. Resources that share common security requirements can be grouped and assigned the same security group, simplifying management and reducing the chances of misconfigurations.

Another important principle of security groups is that they are enforced per resource by the network fabric. Each resource within a security group has its own set of inbound and outbound rules applied to it. Traffic between resources within the same security group is not automatically allowed. If communication between resources in the same security group is required, explicit rules must be defined to permit that traffic. This principle of explicit allowance helps maintain a tight security posture and prevents unintended communication between resources.

It's worth noting that security groups are supported by every major CSP. Whether using AWS, Microsoft Azure, Google Cloud Platform (GCP), or any other CSP, a CSC will find security groups as a standard feature. While the specific terminology and configuration interfaces may vary slightly among CSPs, security group core principles and functionality remain consistent. This widespread support makes security groups a reliable and portable security mechanism across different cloud environments.

In summary, cloud network security groups are a powerful tool for enforcing granular security policies at the resource level. Security groups provide a robust layer of defense by defining rules in policies, applying those policies to resources, and enforcing them through the network fabric. The principle of explicit allowance between resources in the same security group further enhances security. With consistent support for security groups across CSPs, CSCs can confidently leverage this feature to protect their cloud-based assets and maintain a strong security posture.

7.2.5 Beyond the Security Group

While we covered some of these common CSP cloud architecture tools, we have not discussed others. As a CSC gains experience developing cloud-based network environments, it will also learn about recommended reference architectures. Regardless of what mixing and matching a CSC does with an on-premises setup, it is important to have a solid idea of what each one of these CSP services does, and why it has been built for all the major hyperscalers and CSPs (e.g., AWS, Azure, GCP, IBM, Oracle).

Preventative Security Measures:

- **CSP Firewalls:** CSP firewalls, such as Amazon VPC Firewall or Azure Firewall, are built into the cloud platform. They offer the advantage of not requiring additional instances or servers to maintain, simplifying management and reducing operational overhead. However, they may have limitations in terms of customization and advanced features compared to virtual appliances.
- **Virtual Appliances:** Virtual firewall appliances provide greater flexibility and control over firewall rules and configurations. They can be deployed in a load-balanced configuration to ensure high availability. However, this approach adds complexity and requires ongoing maintenance of the VMs or instances running the firewall software. Virtual appliances are commonly available for next generation firewalls (NGFWs) and intrusion detection systems and intrusion prevention systems (IDS/IPS) products.
- **WAF:** WAFs specifically protect web-facing applications from common exploits like SQL injection, cross-site scripting (XSS), and other OWASP Top 10[109] vulnerabilities. Depending on the CSP and CSC requirements, WAFs can be deployed as a cloud-native service or as a virtual appliance. (Some CSPs offer these as a native service.)
- **Egress Filtering/Management:** Egress filtering controls outbound traffic to the Internet or other networks. It can be achieved using CSP firewalls, self-hosted proxies, or virtual appliances. Still, it is important to note that it only covers resources within the specific network where the filter is deployed.

Detective Security Measures:

- **Flow Logs and DNS Logs:** Flow logs and DNS logs provide valuable visibility into network traffic patterns and help detect anomalous activities. Flow logs capture information about the source, destination, protocol, and other attributes of network flows, while DNS logs record domain name resolution requests and responses. These logs can help identify potential security breaches, unauthorized access attempts, and data exfiltration.
- **Traffic Mirroring:** Traffic mirroring allows you to duplicate network traffic for monitoring and analysis purposes. However, Cybersecurity and Infrastructure Security Agency[110] (CISA) has identified it as a potential security risk. If attackers access the mirrored traffic, they could intercept sensitive data. To mitigate this risk, it is recommended to implement strict access controls, encrypt mirrored traffic, store it securely, and regularly audit the configurations and

[109] OWASP. (2021) OWASP Top Ten - *Top Ten Web Application Security Risks*.
[110] CISA (2024) *Identifying and Mitigating Living Off the Land Techniques*.

access logs.

PaaS Security Considerations:

- **API Gateways:** API Gateways are the entry point for accessing PaaS services. They provide features like authentication, rate limiting, and request/response transformations. Some include built-in security capabilities.
- **Resource Policies:** CSPs offer resource-level access control policies, such as AWS IAM policies or Azure Role-Based Access Control (RBAC), to define fine-grained permissions for accessing PaaS services. Properly configuring these policies based on the principle of least privilege is essential.
- **WAF/CDN:** Many PaaS services can be integrated with WAF and Content Delivery Network (CDN) services to enhance security and performance. WAFs protect against web-based threats, while CDNs help mitigate Distributed Denial of Service (DDoS) attacks by absorbing and filtering malicious traffic.
- **Service Endpoints on VPC/VNet:** Service endpoints connect PaaS services directly to a VPC or VNet, enabling a CSC to apply consistent security policies and control traffic flow between the services and virtual network.
- **Inherit the Network Security:** PaaS services often inherit the network security controls applied to the VPC or VNet they are associated with when connected through a network. (The default for many is a direct Internet connection.) This means the same firewall rules, access controls, and monitoring configured for the network extend to the PaaS services, providing a consistent security posture across the cloud environment.

7.2.6 Container Networking

Dedicated container security controls are necessary to address containerization's specific vulnerabilities and threats. Containers are ephemeral, lightweight, and highly dynamic, making traditional security measures less effective. Due to their larger attack surfaces, containers represent a potential entry point for attackers. Vulnerabilities in container images, orchestration platforms, or networking configurations can be exploited to gain unauthorized access or launch attacks. When deploying containerized applications, organizations have multiple options for the network stack, including overlay networks, host networking, and cloud-native networking solutions. Each network stack has security implications and considerations, emphasizing the need for tailored security measures based on the chosen architecture.

A CSC should not assume it can manage all network security at the container level alone. Security groups and perimeter security are still needed to protect the container host systems. These measures are often more effective for perimeter security because they leverage dedicated services or VMs that are more effective and scalable. When it comes to container networking, there are several primary options available for Docker[111] and Kubernetes[112].

[111] Docker. (2016) *Understanding Docker Networking Drivers and their use cases*.
[112] Kubernetes. (2023) *Extending Kubernetes*.

Docker Networking Options:

- **Bridge Network:** This is the default networking mode in Docker. Each container connects to a virtual bridge network on the host, allowing containers to communicate with each other. The bridge network is isolated from the host's stack, providing network isolation.

- **Host Network:** In this mode, containers share the same network stack as the host machine. This means containers directly access the host's network interfaces and can bind to host ports. However, this mode sacrifices network isolation for simplicity.

- **Overlay Network:** Overlay networks allow containers running on different hosts to communicate seamlessly. This is achieved by creating a distributed network across multiple hosts using virtual extensible LAN (VXLAN) or IPSec tunnels. Overlay networks are commonly used in multi-host Docker deployments.

- **Macvlan Network:** Macvlan networks assign a unique MAC address to each container, making them appear as different physical devices on the network. Containers can be connected directly to the physical network, bypassing the host's network stack. This mode is useful when containers need to have their IP addresses on the physical network.

Kubernetes Networking Options:

- **Pod Networking:** In Kubernetes, pods are the smallest deployable units and can contain one or more containers. Each pod gets its IP address, and containers within a pod share the same network namespace, allowing them to communicate using localhost. Kubernetes requires a Container Network Interface (CNI) plugin to handle pod networking.

- **Service Networking:** Kubernetes services provide a stable IP address and DNS name for a set of pods. Services act as load balancers, distributing traffic to the pods based on labels and selectors. There are several types of services, including ClusterIP (internal to the cluster), NodePort (exposed on each node's IP), and LoadBalancer (externally accessible through a CSP's load balancer).

- **Ingress**[113]**:** Ingress is a Kubernetes resource that manages external access to services within a cluster. It acts as a single entry point for HTTP/HTTPS traffic and provides features like URL routing, SSL termination, and virtual hosting. Ingress controllers, such as NGINX[114] or Traefik[115], implement the ingress rules.

- **Network Policies:** Kubernetes network policies allow rule definitions for controlling traffic flow between pods and namespaces. A CSC can specify which pods can communicate with each other based on labels and selectors. Network policies provide a way to enforce network segmentation and restrict unauthorized access within the cluster.

[113] The ingress concept is used in controlling external access to services running within the containerized environment.
[114] NGINX is open-source web server software used for reverse proxy, load balancing, and caching.
[115] Traefik is a open source reverse proxy and ingress controller that streamlines deploying services and APIs.

- **CNI**[116] **Plugins:** Kubernetes relies on CNI plugins to handle pod networking. Some popular CNI plugins include:
    - Flannel, a simple overlay network that assigns a subnet to each node and uses VXLAN or host-gw for inter-node communication.
    - Calico, a highly scalable and performant networking solution that supports both overlay and non-overlay modes, and advanced network policy enforcement.
    - Weave Net, an overlay network that uses a gossip protocol[117] to create a VNet across multiple hosts, enabling automatic discovery and encryption.

These are just a few examples of the container networking options in Docker and Kubernetes. Security must be enforced at both the container and cloud networking layers, and this will vary greatly depending on the network stack chosen.

7.3 Cloud Connectivity

One of the essential characteristics of cloud from the NIST model introduced in *Domain 1: Cloud Computing Concepts & Architectures* is broad network access. Even in a private cloud, configurations and resources are managed and accessed over networks. For the public cloud, this network is either the public Internet or a private leased line used to create a hybrid connection.

Cloud connectivity can be broken into three major categories:

- Connecting to resources in the cloud (e.g., virtual machines)
- Connecting separate virtual networks within a CSP to each other
- Connecting from a data center network to the cloud, or between two different CSPs

7.3.1 Connecting to Resources

This figure provides an overview of securely connecting to resources like VMs or containers running in the cloud.

[116] CNI plugins are modular components that implement the CNI specification, allowing container runtimes to configure network interfaces, manage IP addresses, and establish connectivity for containers within the networking environment.
[117] Github. (2019) weaveworks/mesh - gossip protocol is also known as epidemic protocol used in peer-to-peer communications.

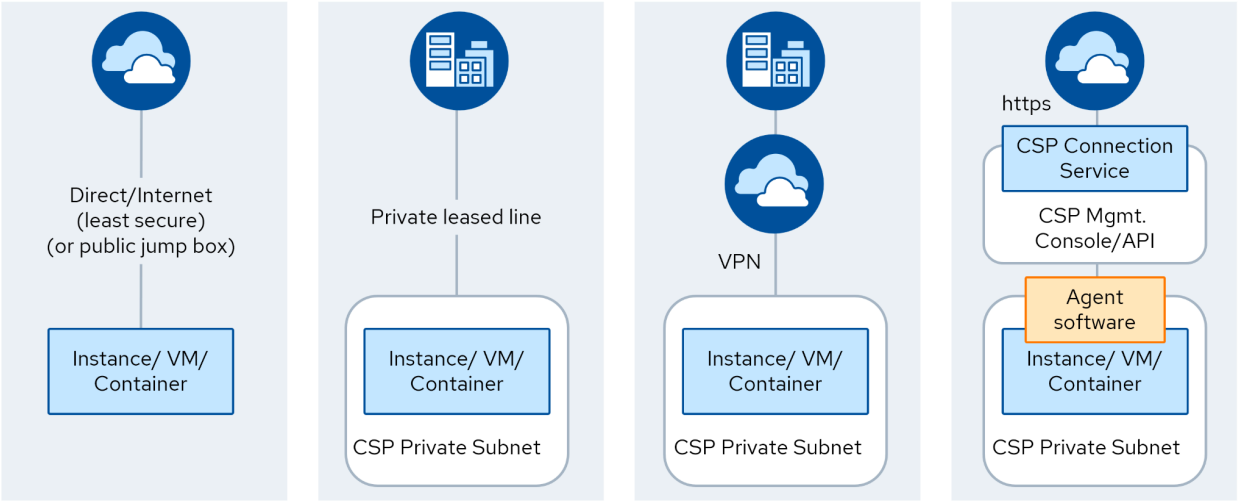


Figure 37: Methods of Connecting to Cloud Resources

Direct Internet or Private Leased Lines
The most direct method is connecting through the public Internet., It is also the least secure. A more secure option is establishing a private leased line between an on-premises network and the CSP. This provides a dedicated, private connection.

VPN
Another common approach is to use a VPN. The VPN creates an encrypted tunnel over the public Internet between a CSC's network and cloud-based resources.

CSP Connection Service
To access the actual VMs or containers, a CSC needs to connect through what is called a *connection service*, such as AWS Session Manager or Azure just-in-time (JIT), thereby getting access through the web console or software designed to support port forwarding. This important CSP service acts as a secure gateway. It also allows a CSC to manage, monitor, and audit access to cloud resources. One advantage is that these services can use IAM/RBAC permissions from the cloud management plane, reducing or eliminating the need for SSH keys or other credentials.

In the figure, note that the VMs (based on approved, secure VM images) and instances (spawned by approved containers) are grouped into private subnets. These provide network-level isolation and security. The connection service allows authorized users to securely connect to resources in those private subnets without exposing them to the Internet. The connection service is managed through a central management console and API. This allows the CSC to configure access policies, monitor connections, and audit activity. This can be considered a form of Zero Trust, a topic discussed later.

Other options constantly evolve, including agent-based network overlays, different forms of port tunneling, or even issuing asynchronous commands through fleet management software.

In summary, using a connection service with private networking provides a secure way to remotely manage cloud-based resources while maintaining tight control and visibility over access.

7.3.2 Connecting Virtual Networks (within a CSP)

To support various enterprise and application requirements, there is a wide range of options and architectures for connecting different virtual networks (e.g., VNets and VPCs) within a CSP. Some, like service endpoints, are designed to only connect to specific services, even if the networks share overlapping IP address ranges. The following are some examples for connecting virtual networks within a CSP.

Peering:

- Establishes a direct, private connection between two virtual networks (VNets/VPCs)
- Highly secure as traffic never traverses the public Internet
- Quick and easy to set up for simple architectures
- Complexity increases rapidly as more networks are added, leading to a mesh of connections that becomes difficult to manage and troubleshoot

Transit/Mesh:

- Provides a hub-and-spoke model for connecting virtual networks
- Uses a managed service like AWS Transit Gateway or Azure Virtual WAN as a central connection point
- Simplifies network architecture and management compared to complex peering meshes
- Easier to implement consistent security, monitoring, and routing policies across connected networks
- Incurs additional costs for the transit service, and sometimes a small latency increase vs. direct connections

Service Endpoints:

- Allow select services to be projected into a virtual network, enabling services to be privately accessed
- For example, connecting multiple application VPCs to a shared database without traversing the public Internet
- Highly secure as the public endpoint is completely disabled, only allowing access from authorized subnets
- Limited to specific supported services (varies by CSP)
- Useful for securing critical data stores, but not a general-purpose network connectivity solution

Other Options:

- Overlaying cloud networks using software gateways and SD-WAN[118] solutions for greater control and flexibility
- Shared private connections (e.g., AWS PrivateLink, Azure Private Endpoints) to privately access services across accounts

[118] SD-WAN is a software-defined approach to managing the WAN.

- Sharing a single virtual network with multiple cloud deployments (using cross-deployment privileges) allows different teams to deploy workloads into the same network

The right choice depends on scale, security needs, management overhead, and types of resources being connected. A typical architecture combines multiple approaches. The key is striking the right balance between security, performance, complexity, and cost.

7.3.2.1 Example: Cloud Network Perimeter Consolidation

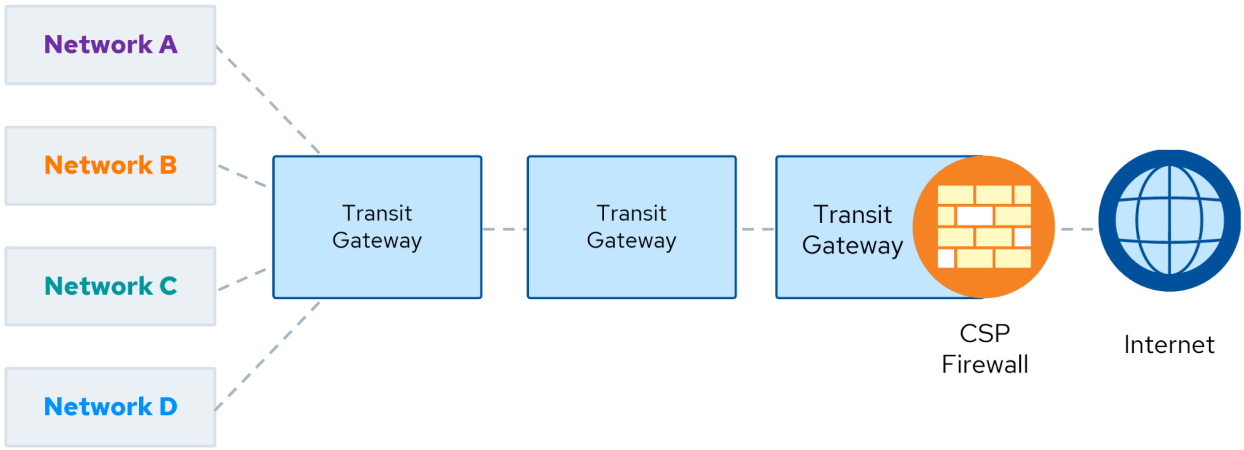


Figure 38: Consolidated Cloud Network Perimeter for Outbound Traffic

In this example, we look at an architectural framework designed to manage and secure outbound Internet traffic originating from several distinct networks, labeled A, B, C, and D, by directing through a unified perimeter network. This setup employs a firewall service provided by a CSP.

Within this architecture, the networks A through D are connected to a central transit gateway. Significantly, these networks are configured without direct access to the Internet. Instead, all Internet-bound traffic is funneled through the transit gateway and then directed to a specialized perimeter network (e.g., VNet or VPC). Within this perimeter network, a CSP firewall — AWS Network Firewall or Azure Firewall — is operational, scrutinizing and filtering the amalgamated traffic.

This design creates a solitary, governed egress point for all traffic intended for the Internet, as opposed to permitting each network to access the Internet autonomously. Such a centralized approach to egress filtering yields multiple benefits, including the establishment of uniform security protocols across all networks for Internet access, which streamlines the management and surveillance of outbound traffic. It effectively diminishes the potential attack surface by concentrating Internet exposure to a single, fortified network and enhances the specificity of logging, inspection, and oversight of traffic leaving the network. The architecture also provides the flexibility to integrate additional security functionalities like IDS/IPS, web filtering, and data loss prevention (DLP) within the perimeter.

However, to optimize this architecture, several considerations must be taken into account. It is crucial to properly size and maintain high availability for both the perimeter network and the firewall to circumvent performance bottlenecks. Routing and security protocols should be carefully tailored to permit only essential outbound traffic. Moreover, to ensure robustness, resilience, and continuity, the use of multiple accounts or CSPs, as well as availability zones, is recommended for redundancy and failover capabilities.

7.3.3 Connecting to Data Centers & Between Providers

When creating a hybrid network, including multi-cloud networking, different technologies are used to carry traffic over the Internet or on private backbones. The following are some examples of options for connecting to data centers and CSPs.

Leased Lines:

- Provide a dedicated, private, high-speed connection between the on-premises data center and the cloud
- Their semi-permanent nature means they are relatively quick to set up and tear down compared to building fiber
- Very fast with predictable performance as the connection is not shared
- Requires compatible hardware and IP addressing at both ends, which can introduce complexity
- Usually, connecting to a meet-me point provided by the network carrier is needed, which then connects to the CSP's network
- Not used for cloud-to-cloud connections as CSPs have their own interconnects

VPN:

- Establish an encrypted tunnel over the Internet between networks, data centers, and cloud VPCs/VNets
- Highly flexible and can be set up and torn down through software configuration
- Requires appropriate hardware (or virtual appliances) at each end to terminate the VPN tunnels
- Performance is dependent on the quality of the Internet connection. Can be impacted by congestion outside the CSC's network
- It is still usually preferred to connect to a CSP's transit network (e.g., AWS Transit Gateway, Azure Virtual WAN)
- Commonly used for backup connectivity and secure remote access to cloud resources

Hybrid Mesh:

- Provides any-to-any connectivity between on-premises and multiple clouds using SD-WAN or software gateways
- Build an overlay network on top of existing connectivity, using software and policies to define topology and traffic flow
- Offers greater flexibility and manageability compared to many point-to-point links
- Reduced dependency on underlying physical networks enables greater resiliency
- Abstraction and automation can greatly reduce configuration burden and the chance of errors
- Incurs some performance costs due to traffic processing at each hop and additional software/licensing costs
- Need to maintain the software-defined policies and network controller

The choice between these depends on the scale, criticality, and variability of the workloads being connected. Leased lines offer the highest performance but least flexibility. VPNs are quick and flexible but can be unpredictable. SD-WAN and hybrid mesh offer a programmable middle-ground for large-scale deployments on-prem and in multiple clouds. Many CSCs use a mix, such as leased lines for primary data

center cloud connection, VPNs for backup and user access, and a software overlay to unify management. The key is understanding the trade-offs and aligning the choice to business requirements.

7.3.3.1 Example: Transit Gateway Hub & Spoke

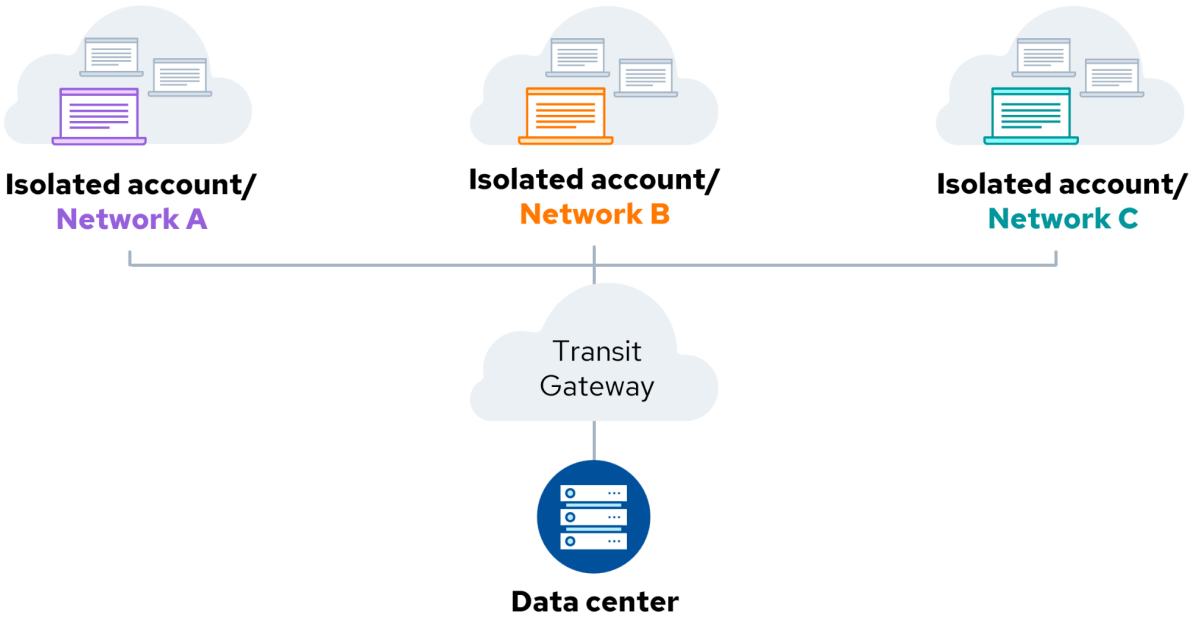


Figure 39: Transit Gateway Hub & Spoke Architecture

This example presents an illustration of a network architecture utilizing a transit gateway, a powerful networking service that enables the connection of isolated networks through a centralized data center. Specifically, it outlines how the transit gateway, known as Transit Gateway in AWS and Azure WAN in Azure, serves as a crucial hub that orchestrates the routing of traffic among isolated networks, labeled in the figure as A, B, and C.

The architecture is governed by a route table within the transit gateway, which meticulously dictates the permitted traffic flow between the networks. The route table specifies that

- Networks A and C are granted the capability to route their traffic towards the data center
- Network A is provisioned to establish communication links with network B, and
- Network B is allowed to connect with network C.

Importantly, the architecture restricts network A from establishing a direct communication pathway with network C, and similarly, network B is prohibited from direct access to the data center.

In the context of AWS, the transit gateway is linked to VPCs across each account by means of Transit Gateway Attachments. In the Azure ecosystem, this connectivity is achieved through a Virtual WAN that ties together the Virtual Networks.

Additionally, the architecture leverages a leased line to form a physical connection between the on-premises data center and the cloud-based transit gateway. By adopting this hub-and-spoke model, the network design succeeds in segmenting and managing the traffic flows between multiple isolated networks. This configuration not only centralizes the enforcement of routing and security policies but also facilitates the availability of shared services located in the data center. The selective accessibility ensures that these services can be utilized from designated networks, thereby negating the need for direct communication across all network segments.

7.4 Zero Trust & Secure Access Service Edge

Zero Trust (ZT) operates on the assumption that trust is never implicit, and robust verification is required at all times for any user or device accessing the network. This section introduces the Zero Trust architectural framework and the technologies that support it, such as software-defined perimeter (SDP) and Zero Trust Network Access (ZTNA). Additionally, it details how SASE integrates various security functions into a unified, cloud-delivered service, addressing the needs of increasingly distributed environments. SASE complements this by merging network and security functions to deliver secure, scalable access to cloud services, optimizing performance and security for distributed environments. This section explores the foundational concepts, benefits, and implementation strategies of Zero Trust and SASE, detailing how they effectively protect cloud architectures and networks.

7.4.1 Zero Trust for Cloud Infrastructure & Networks

Zero Trust as a general security strategy is discussed in *Domain 2: Cloud Governance* and a number of other useful references are available on the CSA Zero Trust Resource Hub[119], including the ZT Guiding Principles document.

Zero Trust is a cybersecurity strategy premised on the idea that no user or asset is to be implicitly trusted. It assumes that a breach has already occurred, or will occur, and therefore, a user should not be granted access to sensitive information by a single verification performed at the enterprise perimeter. Instead, each user, device, application, and transaction must be continually verified.

Implementing a ZTA involves a comprehensive, full stack, multi-pillar approach to security that assumes no trust, whether access is requested from inside or outside the network perimeter[120].

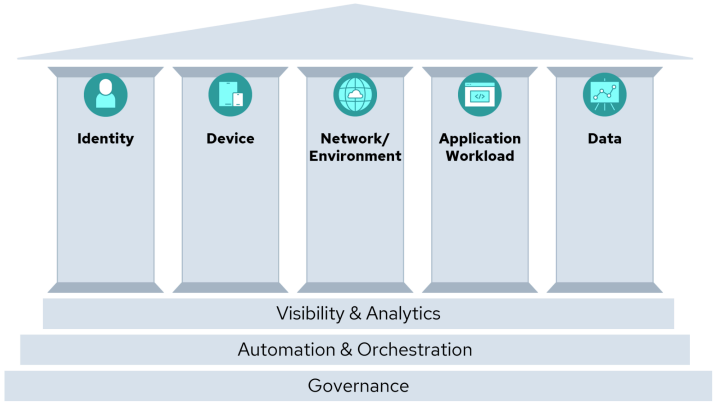


Figure 40: The Five Pillars of ZTMM

[119] CSA. (2024) Zero Trust Resource Hub.
[120] Zero Trust pillars are covered in more detail in *Domain 11: Incidence Response & Resilience*.

7.4.1.1 Foundational ZT Concepts & Capabilities

Zero Trust is a security strategy that is particularly relevant for modern cloud infrastructure and networks, where business applications and assets are often distributed across different environments, and users are often remote and frequently access business systems over the Internet. By adhering to Zero Trust principles, organizations can enhance the security posture of their cloud environments, reducing the risk and potential impact of security breaches, unauthorized access, and lateral movement by attackers. However, implementing Zero Trust generally requires a holistic, full-stack approach, integrating people, processes, and technology across the cloud landscape. This can be accomplished by implementing an appropriately tailored combination of the following security measures.

Continuous Verification:

- Implement phishing-resistant multi-factor authentication (MFA) for all user and administrator access, including cloud console access and API calls.
- Continuously validate user identities, device posture, and session context throughout the user's session by implementing Context Based Access Control (CBAC), which can include RBAC and Attribute-Based Access Control (ABAC).
- Use security analytics and user/entity behavior analytics (UEBA) to detect anomalies and risky behavior, particularly for highly sensitive and administrative access.

Least-Privileged Access:

- Follow the principle of least privilege by granting users and applications only the minimum permissions necessary for their business function.
- Use JIT access and time-limited credentials for elevated privileges.
- Regularly review and revoke unused or excessive permissions, and promptly revoke all access of terminated users through access governance processes.

Micro-Segmentation:

- Implement network segmentation using VPCs, VNets, virtual firewalls, and similar constructs.
- Divide cloud networks into smaller, isolated segments based on workload criticality and security requirements using network security groups (NSGs), and network access control lists (NACLs).
- Enforce granular segmentation policies and control lateral movement between segments.
- Restrict communication between segments and services based on the principle of least privilege.

Infrastructure and Workload Security:

- Deploy IDS/IPS to detect and block malicious traffic.
- Deploy workloads in dedicated, isolated environments (e.g., VMs, containers, serverless functions) with secure boundaries.
- Utilize service mesh architectures and identity-based communication between microservices.
- Implement workload and container runtime protections, vulnerability management, and firewalling.
- Leverage hardware security features like encrypted VMs and confidential computing enclaves.

- Automate security processes such as vulnerability scanning, patch management, and configuration management using DevSecOps practices.
- Adopt immutable infrastructure and ephemeral workload patterns for security and consistency.
- UtilizeIaC tools to provision and configure cloud resources securely.

Data Security:

- Encrypt data at rest and in transit (e.g., with mutually authenticated TLS connections) using strong encryption algorithms and robust key management practices.
- Implement robust backup and disaster recovery (DR) mechanisms to ensure business continuity (BC) in the event of a security breach, ransomware attack, or data loss.
- Monitor and audit data access and usage patterns for potential misuse and exfiltration attempts.
- Implement DLP controls and data masking techniques.

Monitoring and Logging:

- Implement centralized access and traffic logging and monitoring for cloud infrastructure, networks, and workloads.
- Collect and analyze security logs, flow logs, and audit trails for threat detection and incident response leveraging cloud-native logging, monitoring and alerting services.
- Configure alerts and triggers to notify administrators of suspicious activities or security breaches.
- Implement security information and event management (SIEM systems for log aggregation, correlation, and analysis.
- Automate security response and remediation with security orchestration and automation (SOAR) tools.

***Example**:* A Zero Trust fine-grained access control policy for highly sensitive information in the cloud includes several steps. First, validate the user's strong, recent authentication. Next, check the identity and security hygiene of their endpoint device. Ensure that their network and geolocation are acceptable for the time and type of data and workload access requested. Additionally, verify that they are not logged in from multiple geographic locations simultaneously. Finally, use behavioral analytics to ensure that the requested access does not fit an insider risk access profile.

By following these guidelines and principles, a CSC can establish a robust security posture for its cloud infrastructure and networks based on the Zero Trust strategy, helping to mitigate security risks and protecting sensitive data and resources.

7.4.1.2 Zero Trust Conceptual Architecture

NIST SP 800-207 Zero Trust Architecture (ZTA) provides a component model that is also covered in detail in the CSA CCZT Training[121], as depicted below.

[121] Training on zero trust architecture is available in CSA's Certificate of Competence in Zero Trust (CCZT).

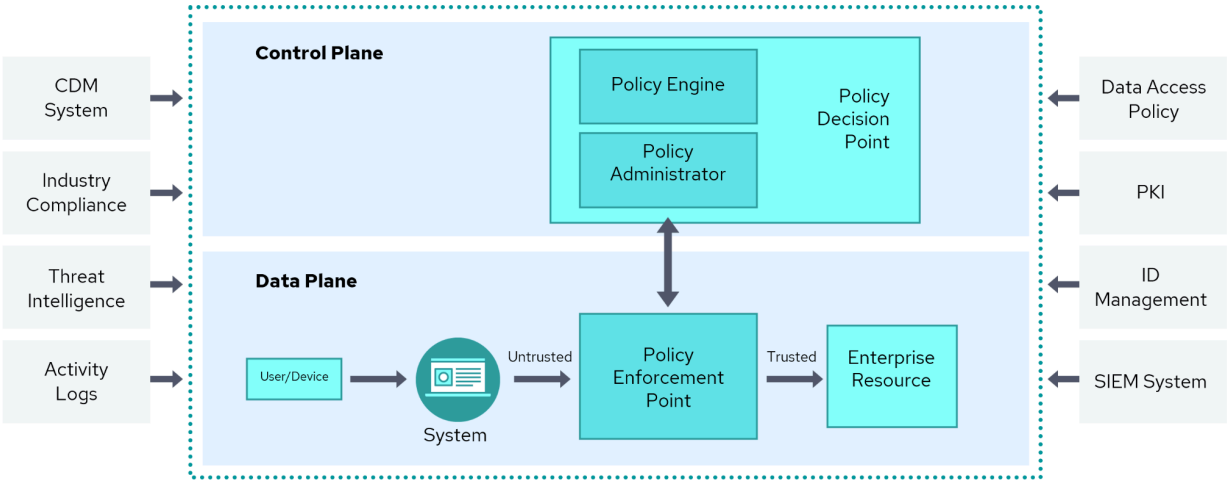


Figure 41: ZTA Core Logical Components (NIST 800-207, pg. 9)

NIST 2020 SP 800-207 provides a simple representation of the key logical components of a ZTA (shown above). In the NIST ZT model, ZT access policies are defined, managed, and enforced using Policy Decision Point (PDP) and the Policy Enforcement Point (PEP). The PDP and PEP regulate resource access by placing them in the traffic access workflow. The PDP comprises a policy administrator and policy engine, which determine the rules and commute them to the PEP. The PEP acts as a gateway to ensure that the correct access has been granted to the right entity, with the correct access level, to an approved resource.

NIST defines the PDP as residing in the control plane, and is the component of the logical architecture that has responsibility to collect, analyze, and transform data first into intelligence and then into rules to govern the access to resources. The PEP resides in the data plane and is the ZT component that, based on input passed by the control plane, has the responsibility to enforce the rules and provide access to resources (data).

Various security-related data sources feed information to the PDP to maintain the rules and keep the overall decision-making process current. Various sources of intelligence feed into the policy engine and support the policy administrator in defining and refining the access rules.[122]

7.4.2 Software Defined Perimeter & Zero Trust Network Access

Two key technology approaches that enable Zero Trust network security areSDP and ZTNA. These approaches are not mutually exclusive, and elements of each can be combined into a tailored ZT security implementation.

Software Defined Perimeter (SDP)

- Establishes a secure, "dark" network that is invisible to unauthorized users and devices.

[122] Training on PDPs available in CSA's Certificate of Competence in Zero Trust (CCZT)

- Implements a "blackout" approach, where the network is inaccessible by default.
- Users and devices must authenticate and be authorized before they can access the SDP-protected resources.
- SDP leverages identity-centric controls and micro-segmentation to limit lateral movement.

ZTNA

- Replaces traditional VPNs with a more granular, application-specific access control model.
- Users are verified and authorized based on identity, device, location, and other contextual factors.
- Access is provided to specific applications or resources, rather than granting broad network access.
- ZTNA solutions can be cloud-hosted (ZTNA-as-a-Service) or on-premises.

By implementing Zero Trust network security principles, organizations can significantly enhance their overall security posture, reduce the risk of data breaches, and better protect their assets in the face of evolving cyber threats. The combination of ZTNA and SDP provides a robust framework for securing modern, cloud-centric and remote-access-intensive IT environments. NIST SP 800-215, Guide to a Secure Enterprise Network Landscape[123] is a good reference for these topics.

7.4.2.1 Software Defined Perimeter

SDP[124] is a Zero Trust network security architecture that is implemented to provide full (OSI network) stack security. SDP implementations hide assets and authorize access using a separate control plane and data plane prior to allowing any connections to hidden assets. SDP implements foundational Zero Trust principles.

ZT implementations require the verification of anything and everything attempting to access assets before authorization. Additionally, ZT requires continued evaluation of sessions and their risk levels during the entire duration of the connection. A ZT implementation using SDP enables organizations to defend new variations of old attack methods constantly surfacing in existing network and infrastructure perimeter-centric networking models. Implementing SDP improves the security posture of businesses that face the challenge of continuously adapting to expanding attack surfaces that are increasingly more complex.[125] The CSC must monitor the security posture of the assets. SDP enforces this access management strategy by enabling a default drop-all gateway until users/devices are properly authenticated and authorized to access the hidden assets. By requiring the pre-vetting of connections, SDP enables complete control over who can connect, from which devices to what services and infrastructure, and other conditions and contextual factors, such as hours of operation and geolocation.

As described in the SDP Architecture Guide v2, SDP consists of the following major components:

- The client/initiating host
- The service/accepting host, also referred to as the PEP per NIST's ZTA model
- An SDP controller to which the accepting host and initiating host both connect, also referred to as the PDP per NIST's ZTA model

[123] *NIST. (2022) Guide to a Secure Enterprise Network Landscape*
[124] CSA. (2020) *Software-Defined Perimeter (SDP) and Zero Trust.*
[125] *CSA. (2020) Software-Defined Perimeter (SDP) and Zero Trust.*

- An SDP gateway that implements the drop-all firewall

According to the SDP Architecture Guide v2, SDP works in the following manner:

- The SDP client software on the initiating host opens a connection to the SDP. Initiating host devices, such as laptops, tablets, and smartphones, are user-facing, meaning the SDP client software is run on the devices themselves. The network can be outside the control of the enterprise operating the SDP.
- Accepting host devices receive connections from initiating hosts and provide a set of SDP-protecting/secured services. Accepting hosts typically reside on a network under the CSC's control (and/or under the control of a direct representative[126]).
- An SDP gateway provides authorized users and devices with access to protected processes and services. The gateway can also enact monitoring, logging, and reporting on these connections.

Initiating host and accepting host devices connect to an SDP controller, which is a device/appliance or process that secures access to isolated services by ensuring that:

1. Users are authenticated and authorized
2. Devices are validated
3. Secure communications are established
4. User and management traffic remain separate on the network

The controller and accepting host are invisible and inaccessible to unauthorized users and devices. SDP implementations can support several different connectivity configurations for different communications use cases. Reference the Software-Defined Perimeter (SDP) Specification v2.0 for details.

7.4.2.2 Zero Trust Network Access

ZTNA is a key component of the Zero Trust security model, specifically focused on secure remote access to applications and resources. ZTNA replaces traditional VPNs with more granular rules and an application-specific access control model. Users are verified and authorized to access specific applications or resources based on identity, device, location, and other contextual factors.

By implementing ZTNA principles, organizations can significantly reduce their attack surface, enforce granular access controls, to mitigate the risk of unauthorized access, data breaches, and lateral movement within networks, and applications.[127]

[126] CSA. (2022) Software-Defined Perimeter (SDP) Specification v2.0 - *SDP Accepting Hosts (AH)*.
[127] Training on ZTNA available in CSA's Certificate of Competence in Zero Trust (CCZT).

7.4.3 SASE

SASE is an emerging cybersecurity concept that combines network security functions with WAN and proxy capabilities to deliver a comprehensive, cloud-native service. It is designed to address the challenges of securing endpoint devices and access to applications and data in a cloud-first, mobile-first world, where users and resources are increasingly distributed outside of traditional network perimeters.

7.4.3.1 SASE Framework & Architecture Overview

SASE is a framework or architectural approach that combines networking and security functions into a single, cloud-delivered service. SASE aims to provide secure access to applications and data for users, regardless of their location, while also ensuring consistent security policies and controls across the organization's network.

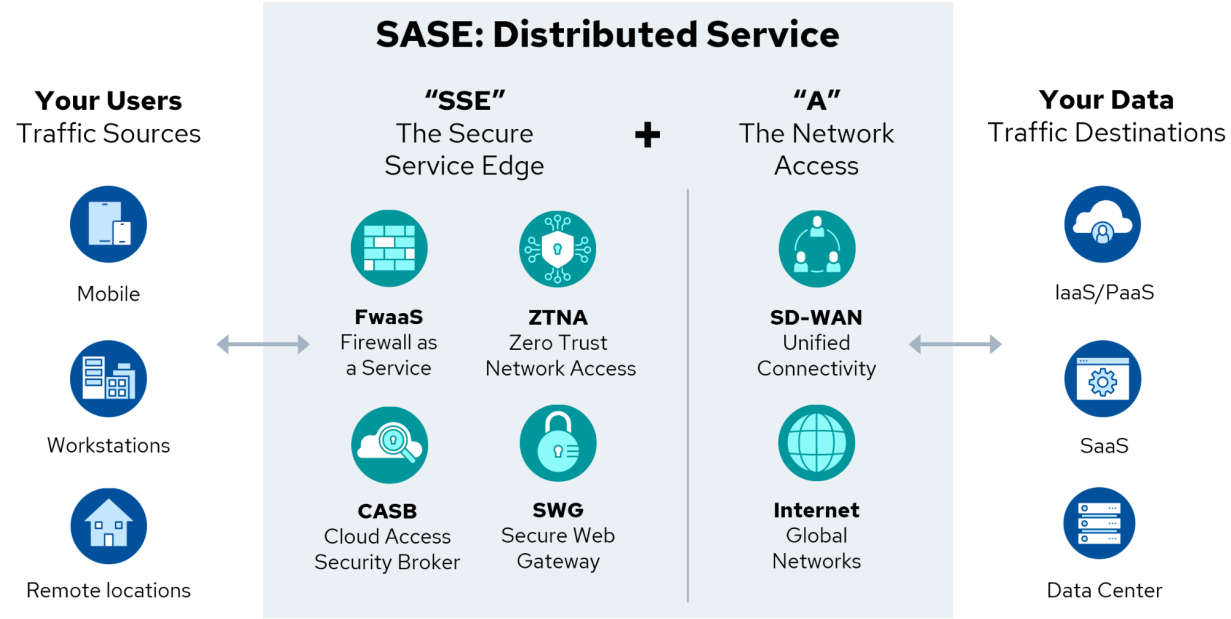


Figure 42: SASE Framework and Architecture Overview

SASE plays a significant role in enabling Zero Trust security in cloud environments. Zero Trust is a security model that assumes no implicit trust and continuously verifies every access request, regardless of where it originates. SASE supports this by providing a unified platform for enforcing granular, context-aware access policies across all users, devices, and applications. It integrates security functions like secure web gateways, Cloud Access Security Brokers (CASBs), ZTNA, and traditional firewall capabilities into a single, cloud-delivered service. This allows organizations to consistently apply security policies and monitor access to cloud resources, irrespective of the user's location or device.

7.4.3.2 SASE Implementation & Benefits

SASE's integration of network and application layer security mechanisms simplifies the deployment and management of Zero Trust in cloud environments. By delivering security as a cloud-native service, SASE reduces or eliminates the need to manage multiple point products and enables organizations to scale their security infrastructure quickly as their cloud footprint grows. It also provides a more user-centric approach to security, with the ability to enforce policies based on user identity, device posture, and application sensitivity. This is essential for implementing least-privilege access in the cloud, where remote users require access to specific applications and data rather than entire networks.

As organizations continue to adopt cloud services and embrace remote work, SASE will be instrumental in realizing the full potential of Zero Trust security. By providing a unified, cloud-delivered platform for securing access to any application, from any device, over any network, SASE enables organizations to consistently enforce Zero Trust policies across their entire digital estate, including cloud, on-premises and hybrid environments. This not only improves overall security posture but also allows for fully leveraging the agility and scalability of the cloud without compromising enterprise security.

Summary

Securing cloud infrastructure is a dual-focus task that safeguards both the CSP's setup and the configurations deployed by CSCs. The core pillars of infrastructure security encompass creating secure architectures, ensuring configurations are secure from the start, integrating security early in the development lifecycle (shift-left practices), and maintaining vigilance through monitoring and applying guardrails.

Cloud networks, built on the principles of SDNs, offer advanced security capabilities such as implementing a default deny policy, managing access and rules based on policies, and allowing for detailed network segmentation. These features significantly bolster the security framework within cloud environments.

Incorporating Zero Trust principles, exemplified by SDP and SASE, is essential for securing multi-cloud connectivity and achieving secure remote access. These models ensure that access is tightly controlled and provided based on verified identity and context, enhancing security in distributed environments.

Container networking introduces a new layer of complexity by adding an abstraction layer above traditional virtualized cloud networks. This necessitates applying security measures at both the container and cloud network layers to prevent vulnerabilities from being exploited.

Finally, cloud network security is not limited to just security groups. It also includes preventive measures like deploying firewalls, IDS/IPS, and WAFs, alongside detective controls, such as flow logs, and traffic mirroring. These elements work together to form a robust defense against cyber threats, ensuring the integrity and resilience of cloud infrastructures for businesses leveraging cloud technology.

Recommendations

Cloud Infrastructure Security

- Follow the principles of the Well-Architected Framework or equivalent to guide design and implementation decisions for improved security and cost-effectiveness when using a cloud.
- Shift Security Left: Embed security controls and testing early in the development lifecycle rather than treating it as an afterthought..
- Use IaC to manage and provision IT infrastructure using machine-readable configuration files.

Cloud Network Fundamentals

- Implement Software-Defined Networking (SDN), to enhance flexibility, agility, and simplified network operations and management.
- Leverage Cloud network security groups.
- Consider preventative and detective security measures.

Cloud Connectivity

- Use a connection service with private networking for secure remote management of cloud-based resources.
- Consider peering or transit/mesh architectures for connecting virtual networks within a CSP.
- Evaluate different options for connecting data centers and between CSPs in a hybrid network.

Zero Trust & Secure Access Service Edge

- Implement Zero Trust: a cybersecurity strategy that assumes no user or asset is implicitly trusted and requires continuous verification for every user, device, application, and transaction.
- Use SASE to provide secure access to applications and data for users, regardless of their location.

Additional Guidance

- How to Design a Secure Serverless Architecture | CSA
- Cloud OS Security Specification v2.0 | CSA
- The Six Pillars of DevSecOps: Automation | CSA
- Software-Defined Perimeter as a DDoS Prevention Mechanism | CSA
- CSA IoT Security Controls Framework | CSA

Domain 8: Cloud Workload Security

Introduction

This domain covers securing cloud workloads. **Cloud workload** refers to the various tasks, applications, services, and processes run in cloud computing environments. Cloud workloads allow for scalability, flexibility, and efficiency, enabling businesses and individuals to access and run applications or data processing tasks without investing heavily in physical hardware. Cloud workloads encompass a range of resources, including virtual machines (VMs), containers, serverless functions (also referred to as function as a service (FaaS)), AI, and platform as a service (PaaS). The dynamic nature of cloud environments, with their constantly changing and expanding resources, requires a distinct approach to security compared to traditional methods.

Learning Objectives

In this domain, you will learn to:

- Understand the challenges and uniqueness of creating security approaches for cloud security workloads.
- Understand security considerations for virtual machines.
- Understand security considerations used for securing containers.
- Understand security considerations to provide PaaS security.
- Understand security considerations for securing serverless or function as a service workloads.
- Understand security considerations for AI Workloads.

8.1 Introduction to Cloud Workload Security

For businesses using the cloud, securing these workloads is not just about protecting data. It is also about ensuring that their operations can continue without interruption and that they comply with legislation and regulations, including data protection and privacy regulations.

Outlined below are the key differences between cloud workloads and traditional environments:

- **Dynamic and expansive**: Unlike traditional environments where data and workloads are relatively static, the cloud is a dynamic, expansive canvas that is constantly evolving. This environment's fluid nature demands a security approach that is just as agile and adaptable. For

security professionals venturing into the cloud, this means rethinking standard security measures and adapting to a landscape where the rules of engagement are continuously rewritten.

- **Complexity and diversity**: There are many different types of workloads, each with its own requirements, so a one-size-fits-all approach to security does not work.
- **Integrity, confidentiality, and availability**: The core of cloud workload security lies in maintaining data integrity, confidentiality, and availability – principles that are the bedrock of cybersecurity. In the cloud, it is vital to ensure that data is unaltered (integrity), only accessible to authorized users (confidentiality), and available when needed (availability).

8.1.1 Types of Cloud Workloads

Various cloud workloads are utilized within cloud environments, each with its distinct characteristics and security implications. From managing virtual instances and securing containerized applications, to ensuring the safety of serverless and artificial intelligence (AI) operations, this section provides essential guidance for navigating the complex landscape of cloud security, underscoring the importance of rigorous governance and proactive security measures.

- **Virtual machines (VMs) and instances**: VMs, also called instances, are a cornerstone of cloud computing. They offer isolation through separate operating systems and enforced security boundaries by the hypervisor and other management plane components. The hypervisor is a key component maintained by the cloud service provider (CSP). However, the security of the guest OS within each VM is typically handled by the cloud service customer (CSC), requiring meticulous configuration and patching. Moreover, "VM sprawl" can pose significant security risks. Additionally, managing snapshots and images is crucial to prevent sensitive data leakage, highlighting the need for strict governance.
- **Containers**: These are isolated runtime environments that share the host operating system's kernel but run as separate, self-contained processes with their own file systems, libraries, and configurations. Containers provide a lightweight and efficient alternative to VMs but present different security challenges. Since containers share the host OS kernel, they inherently offer weaker isolation. Security in containerized environments hinges on correctly configuring OS-level controls, maintaining container image security, and ensuring the container's runtime environment is configured properly. Despite the benefits of orchestrators like Kubernetes in enhancing security, orchestrators introduce additional complexity that must be navigated carefully to prevent breaches.
- **Platform as a Service (PaaS**): These workloads extend the functionality of cloud platforms by offering a suite of tools and services that facilitate the development, deployment, and management of applications with greater efficiency and less overhead. These services, ranging from databases and messaging systems to content delivery networks (CDNs), present different security considerations.
- **Serverless or Function as a Service (FaaS**): FaaS is a cloud computing model whereby developers write and deploy individual functions that are executed in response to events or requests, without the need to manage the underlying infrastructure. This serverless model

entrusts a greater share of security responsibilities to the CSP. This trust reallocation capitalizes on the CSP's specialized security expertise and advanced protective measures, thus minimizing the attack surface. The short-running nature of the execution environment, coupled with enforced isolation by CSPs, offers inherent security benefits. However, managing secrets and configuring functions with the least privilege is paramount to safeguarding serverless applications against unauthorized access and potential attacks, such as denial of service or financial exhaustion through auto-scaling.

- **AI Workloads**: These workloads process vast amounts of data to learn, make decisions, or offer predictions. As such, they introduce unique security challenges. Ensuring the integrity and privacy of data becomes paramount, with specific emphasis on safeguarding against adversarial attacks, preventing model theft, and protecting against prompt injections. Despite these vulnerabilities, AI Workloads leverage the advanced computational resources and scalability of cloud environments.

In general, when it comes to cloud workloads, the management responsibilities shift towards the CSP, especially in models like serverless computing. While the attack surface may diminish, visibility, control, and governance challenges persist. Security monitoring and governance thus becomes critical in maintaining a robust security posture across all cloud workloads, ensuring that operations can continue without interruption and in compliance with data protection regulations.

8.1.2 Cloud Workloads: Short & long-running

The concept of *short-running (ephemeral)* vs. *long-running* (*immutable*) in cloud workloads represents two different approaches to managing and securing workloads. The short-running/ephemeral approach involves treating workloads as interchangeable and disposable resources. In contrast, the long-running/immutable approach treats workloads as indispensable and requires manual upkeep. Traditionally in computing, infrastructure has been primarily treated using the short-running model, but the introduction of cloud computing requires us to rethink our approach. Both models have implications for security, operational management, and scalability, so it is important to understand the differences and when it is appropriate to use each approach.

Short-running (Ephemeral)
In a cloud-native architecture, most workloads operate under the short-running model. These are transient services – they come and go as needed, sometimes only existing for a brief period to handle specific tasks or workloads. Security in short-running workloads is proactive and baked in; it is part of the creation process of the VM or container image and does not require manual configuration or post-deployment. The use of immutable infrastructure means that instead of patching or reconfiguring, new workloads are spun up to replace any that are compromised or harmful. This model supports auto-scaling and self-healing capabilities and is becoming the dominant pattern in modern, cloud-native application architectures due to its efficiency and the enhanced security posture it offers.

Long-running (Immutable)
In contrast, long-running workloads are those that are carefully nurtured and maintained over extended periods. These workloads are often unique, manually built and managed, and have security software installed and updated manually. Such an approach is time-consuming and prone to human error, which can lead to inconsistent security practices. long-running workloads are typically seen in scenarios where traditional on-premises workloads are moved to the cloud without altering the underlying management

philosophy (known as 'lift and shift[128]'). While long-running workloads can be key for certain applications, such as databases that need special care, they are less resilient and can be costly to maintain when issues arise.

Comparing Short Vs. Long-running in Cloud Security
When it comes to security, short-running workloads tend to be more secure than long-running workloads because their short lifetimes limit exposure to threats, and the automated nature of their configuration ensures consistency and reduces errors. Immutable infrastructure prevents configuration drift and unpatched vulnerabilities, making it easier to maintain and scale security measures. Testing the security of an image is also more straightforward than a long-running workload.

The short-running model advocates for upfront investment in automating security and integrating it into deployment pipelines, which pays off with scale. This strategy, while efficient, may not be suitable for all cases. Some short-running workloads, due to their nature or the requirements of the business, continue to be treated as long-running. These should be the exception rather than the rule, protected and isolated within the cloud environment to minimize risks. Adopting the short-running model by default and limiting the use of long-running to special cases is considered a best practice in cloud security.

The shift to ephemeral and immutable workloads for technology practitioners represents a strategic move towards a use-and-replace methodology. It is a decisive step away from the traditional *fix-and-patch* model, steering towards an operational landscape that favors robustness and minimizes vulnerability. In this evolved framework, the cloud environment becomes a more secure, reliable, and predictable space for hosting virtual resources.

8.1.3 Impact on Traditional Workload Security Controls

For someone just starting in technology, think of cloud workload security as setting up the appropriate guardrails (technical preventative controls[129]), watching over them efficiently (monitoring)[130], and regularly checking their health and readiness (assessment). But doing so in a very large, ever-changing virtual space where things move and change at a much faster pace than in traditional computing environments.

The following are some important considerations for cloud workload security controls.

Enforcing Controls: Many organizations use security agents,[131] like endpoint protection platforms or endpoint detection and response (EDR), with their cloud workloads. These tools need to embrace and support the dynamic and virtualized nature of the cloud. Agents should be lightweight so they do not significantly increase compute costs. They should be cloud-aware and not rely on fixed IP addresses or other static configurations. The agents should be able to self-register when new workloads are launched to make them usable in autoscale groups and immutable scenarios. The tools should also not require inbound network ports in security groups, which can increase the attack surface if an attacker does get onto the virtual network.

[128] The "Lift and Shift" approach to cloud migration is also covered in *Domain 7: Infrastructure and Networking*.
[129] This reference is specific to technical preventative controls. Additional control mechanisms are covered in *Domain 2: Cloud Governance*.
[130] Security monitoring is covered in detain in *Domain 6: Security Monitoring*.
[131] Agents are specialized software components that are installed on devices for performing specific security-related "actions".

Monitoring: Tools that typically use agents to capture workload logs generated by the OS. These logs should be sent quickly to a central location quickly due to the transient nature of cloud resources. In a non-cloud environment, monitoring agents typically move logs to a log server over the network, but in the cloud those logs can be saved directly to native cloud storage which may be more cost-efficient. It's important to be cost-effective and flexible to accommodate the varying storage and computational requirements in the cloud. Log entries need to be enriched to support workload identity since IP addresses or system names alone may refer to multiple workloads and change frequently, as names and addresses are reused during scaling operations or across different cloud deployments.

Assessment: Vulnerability assessments (scans) are traditionally performed over the network, but this may not be effective in the cloud, since even internal networks have "default-deny" controls and security groups restrict connections on a per-workload basis. One cannot rely on putting an assessment server on the same subnet and scanning for vulnerabilities. Three options that are better suited for cloud deployments. The first is to assess virtual machine and container images as they are built before the VMs are deployed. Fixing vulnerabilities in images, and tracking the history of images, prevents vulnerabilities in new VMs and allows fast auditing to determine which running VMs are on vulnerable versions. Second, organizations can perform runtime vulnerability assessments by taking snapshots of VMs and assessing those offline, without affecting the running workload. Finally, it is an option to build vulnerability assessment agents into images.

Cloud Workload Protection Platforms (CWPP): These are cloud and container-specific workload tools that offer multiple workload security capabilities. They can perform in-depth vulnerability scans across cloud workloads (e.g., VMs, containers, serverless) and prioritize the findings based on exploitability and business impact. Some tools also integrate log and activity collection, additional monitoring, and even runtime protection.

8.1.4 Software Composition Analysis

Software composition analysis (SCA) tools and the software bill of materials (SBOM[132]) are important tools used in image pipelines to improve workload security. These tools are crucial in managing dependencies, identifying vulnerabilities, and ensuring compliance across different cloud service models.

SCA tools are essential for examining cloud workloads for open-source and commercial components. Whether dealing with VMs, containers, or serverless functions, SCA helps pinpoint known vulnerabilities and licensing issues within these components. By integrating SCA into the Continuous Integration/Continuous Deployment (CI/CD) pipeline, developers can ensure that potential security risks are addressed early in the application lifecycle. The proactive vulnerability management facilitated by SCA allows teams to detect and address vulnerabilities in their dependencies, ensuring all components comply with organizational licensing policies and reducing the risk of legal and security issues.

Key benefits of SCA across cloud workloads include:

[132] NIST. (2021) *Executive Order 14028* - A formal record containing the details and supply chain relationships of various components used in building software. Software developers and vendors often create products by assembling existing open-source and commercial software components. The SBOM enumerates these components in a product.

- **Proactive Vulnerability Management**: Helps identify and rectify vulnerabilities before deployment, enhancing the security posture of the cloud environment.
- **License Compliance**: Ensures that all software components comply with the organization's licensing agreements, thus avoiding legal issues.
- **Risk Assessment**: Offers a risk score for each identified vulnerability, helping prioritize fixes based on their potential impact.

8.1.5 Software Bill of Materials

An SBOM acts as a detailed recipe for software, listing every component along with its version and how it interacts within the software environment. This level of detail is critical for managing potential vulnerabilities and ensuring quality across all types of cloud workloads. Generating an SBOM provides transparency that is vital for effective vulnerability management and regulatory compliance, making it easier to track the use and interaction of open-source and proprietary components.

The importance of SBOM in cloud workloads includes:

- **Enhanced transparency:** Provides a comprehensive breakdown of all software components, contributing to better governance and control over the cloud workload's software supply chain.
- **Improved security response:** Facilitates faster identification and remediation of vulnerabilities by pinpointing the exact components affected.
- **Regulatory compliance:** Assists in meeting compliance requirements that mandate the disclosure of software components, something crucial in industries with stringent regulatory requirements.

In summary, integrating SCA and SBOM into the development and deployment processes of cloud workloads — be it VMs, containers, or serverless architectures — not only bolsters security but also ensures the reliability and compliance of these environments. These practices are indispensable for organizations looking to secure their cloud operations against the evolving landscape of cyber threats.

8.2 Virtual Machines

VMs are entire operating systems running on hypervisors. VMs, also called instances, are the primary method of running cloud workloads since they are close to the hardware and commonly understood. VMs, through hypervisor-enforced isolation, provide stringent separation between workloads within and between customers. This segregation ensures that each VM maintains its full-stack OS.

VM deployment is consistently initiated from a standardized base image, establishing a uniform foundation for security configurations. Also, the cloud's capability to autoscale VMs facilitates the use of immutable workloads, enhancing efficiency and adapting to fluctuating demand.

8.2.1 Virtual Machine Challenges & Mitigations

Despite the security afforded by isolation, VMs operating on shared physical hardware may be susceptible to side-channel attacks, in which a perpetrator could deduce information from a VM by analyzing the hardware behavior. To mitigate such risks, each VM is not only individually accessible but also demands a meticulous security configuration that is established from the base VM image. This ensures an unassailable security posture is maintained, fortifying VMs against unauthorized access and providing a consistent layer of protection across the cloud infrastructure.

The unique security challenges of VMs include:

- **Image control:** Ensuring that VM images are deployed securely and stay up-to-date poses a challenge, particularly when users can provide their own images.
- **Patch management:** Regularly updating base images with the latest security patches is essential but can be resource-intensive
- **Change management:** Allowing CSCs to alter running VMs could inadvertently introduce vulnerabilities or lead to configuration drift.
- **Attack surface management:** The operating systems and applications within VMs create a larger attack surface relative to more streamlined workload types, such as containers.
- **Lifecycle management:** Long-running, manually configured VMs that are infrequently replaced present difficulties in maintaining a strong security posture.
- **Network security:** Granular control mechanisms required to secure network access to VMs, including Secure Shell (SSH), addressing the challenge of inadvertent disclosure of SSH private keys used to access VM instances.
- **Rootkits and bootkits:** Infecting firmware and operating systems using rootkits and bootkits that have kernel-level privileges

Effective vulnerability management for VMs involves identifying, assessing, and mitigating security flaws before they can be exploited. A strategic approach emphasizing regular assessment, prioritization, automation, and integration is essential to navigate vulnerability management challenges. While managing runtime vulnerabilities is incredibly important, mitigating vulnerabilities in images should always be prioritized.

To address these challenges, the following measures should be taken:

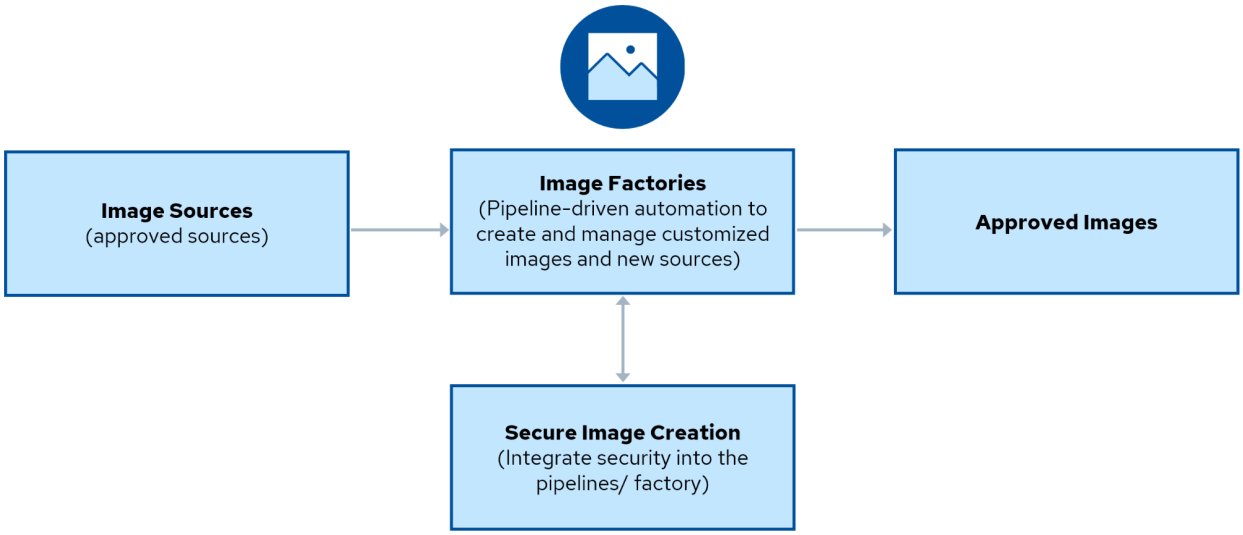
- **Secure base images:** Enforce secure base VM images from a centrally managed catalog. Images should be versioned and immutable once built. These images are typically created using deployment pipelines that may be called “image factories” when automated.
- **Scanning:** Scan VM images for vulnerabilities and misconfiguration before approving them for use.

- **Minimize attack surface:** Remove unnecessary OS components and enforce hardening of the OS configuration.
- **Prioritization:** Focus on vulnerabilities with the highest risk in the environment, considering exploitability and potential impact.
- **Automation:** Leverage automation for scanning, patching, and reporting to enhance efficiency and reduce human error.
- **Integration:** Ensure vulnerability management tools integrate with existing security and IT management systems for a unified approach.
- **Embrace short-running VMs:** Where possible, adopt an immutable infrastructure approach where VMs are ephemeral and replaceable, minimizing long-running VMs that are difficult to maintain securely.
- **Configuration management:** Use configuration management and infrastructure as code (IaC) to maintain desired states and avoid configuration drift.
- **Monitoring and logging:** Centralize the collection of logs and implement metrics that indicate intrusion attempts or suspicious activities. Effective monitoring and logging provide visibility into VM activities, enabling timely detection and response to security incidents.
- **Access controls and least privileges:** Grant applications and users minimum permissions to access only authorized software packages, libraries, and other digital assets associated with the VM. Implementing least privilege principles reduces the attack surface and limits the potential impact of security breaches.
- **Host-based firewalls and SSH hardening:** Restrict network access to VM instances by controlling ports, protocols, and packet types using host-based firewalls, such as the Linux IP tables firewall. Harden SSH on all VM instances through the use of SSH configuration options.
- **Secure boot:** Safeguard against potential malware that can attack the preboot environment to outsmart the OS and antivirus software.
- **Specialized security tools:** Implement tools designed for cloud environments that continuously monitor the hypervisor. This monitoring function is comparable to a security guard that supervises an entire floor within an apartment complex, ensuring the safety and integrity of the virtual infrastructure.

Proper security controls and management of VMs can yield a secure and flexible foundation for cloud workloads. However, given the increased surface area and control, there is a higher responsibility for security and the potential for misconfiguration. Adhering to cloud security best practices, such as immutable infrastructure, ephemeral workloads, and automated configuration management can significantly mitigate these risks, ensuring a secure, efficient, and resilient cloud environment.

8.2.2 Creating Secure Virtual Machine Images with Factories

Creating and managing VM images is pivotal to securing VM environments. This process involves embedding security measures from the ground up. As such, it is essential to establish a set of practices that streamlines the creation of VM images and integrates security seamlessly into every layer. Two key secure VM image creation aspects are image factories and image sources.



Approved sources + approved process = approved images.

Figure 43: Secure Virtual Machine Image Creation Process

Image Factories are automated processes and tools for assembling and customizing VM images. Consider them as the kitchen where the recipe (using the image sources) is followed to create the final VM image (the meal). Image factories ensure consistency and repeatability in the VM creation process, with a strong focus on security.

Image factories serve as the assembly lines for VM images, where consistency, security, and efficiency are paramount. This can include:

- Building, testing, and fine-tuning VM images to ensure consistency across deployments.
- Minimizing discrepancies that could lead to security vulnerabilities.
- Streamlining the integration of security updates and configuration changes.

Image Sources are the starting points for building a VM image. They provide the core components like the OS, applications, libraries, and configuration files. Think of them as the ingredients for the VM recipe.

Image Sources focus on the careful curation and maintenance of the components that constitute VM images, including:

- Preserving a library of source code and settings essential for creating VM images.
- Incorporating security checks within the build process.
- Keeping a comprehensive version history for easy rollback in case of issues.

Secure VM image creation encompasses a series of best practices aimed at reinforcing the security posture of VM images through:

- **Least privilege:** To minimize potential vulnerabilities, set up VM images with only the essential software and access rights.
- **Patch management:** Regularly update the VM images with the latest security improvements to protect against new threats.
- **Configuration management:** Use standardized templates and scripts to ensure all VM images meet the required security standards to automate the image creation workflow and reduce manual errors.
- **Validation and testing:** Before using a VM image, thoroughly check it for security weaknesses and operational issues to ensure it is safe and functions correctly. VM images must always come from trusted sources.
- **Use golden images:** Create a "golden" image, which is a pristine, minimal VM image containing only the essential OS and configuration settings. This image can be the base for all other VM images, promoting consistency and reducing sprawl.

Ultimately, securing VM images is about creating a consistent and repeatable process that embeds security into the blueprint of VMs, ensuring that as new VMs are spun up, they are already equipped to resist cyber threats.

8.2.2.1 Recommended Tools & Best Practices for VMs

Leveraging the right tools is important to the success of any vulnerability management program. These tools offer specialized functions that cater to the diverse needs of VM security:

- **Cloud Workload Protection Platforms (CWPP)** typically include features for in-depth vulnerability scans across cloud workloads (VMs, containers, serverless) and prioritize the findings based on exploitability and business impact.
- **Traditional vulnerability scanners** tend not to be as effective in the cloud, however many vulnerability scanners now also support agents. These products may be rebranded as CWPP, depending on the product.
- **Configuration management tools** automate patch deployment and configuration hardening.
- **Endpoint Detection and Response (EDR)** agents perform runtime monitoring and some support vulnerability assessment.
- **Security Information & Event Management (SIEM)** for real-time monitoring and reporting.

The following vulnerability management lifecycle represents a systematic approach to handling VM vulnerabilities, from discovery to resolution. In the cloud, this cycle should expand to cover images and

alternatives to patching, like replacing running VMs with updated images (the concept behind immutability). The cycle is comprised of:

- **Identification:** Use automated tools to scan VMs for known vulnerabilities.
- **Assessment:** Analyze and evaluate the risk associated with identified vulnerabilities, considering the VM's role and the classification of data processed/stored, such as data sensitivity.
- **Mitigation & Reporting:** Apply patches, configure security settings, and employ workarounds to address vulnerabilities.
- **Documentation:** Keep detailed records of vulnerabilities, assessments, remediation actions for reporting, compliance and auditing.

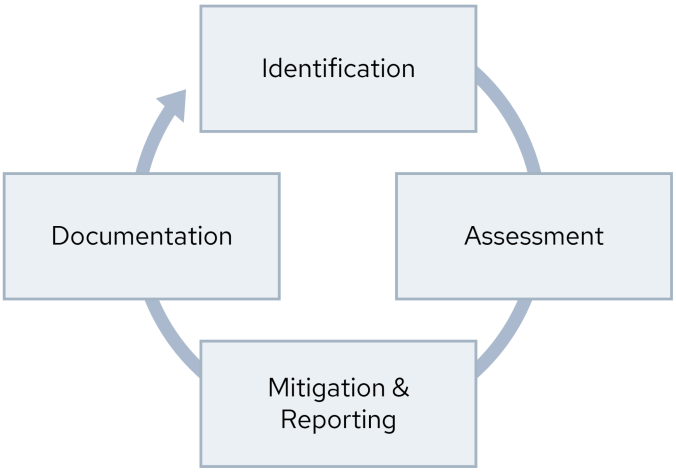


Figure 44: Vulnerability Management Lifecycle for VMs

By integrating these strategies, tools, and practices into the security framework, organizations can significantly enhance the protection of their VM environments against the many vulnerabilities that threaten digital assets in the contemporary cybersecurity landscape.

8.2.3 Creating Secure Images with Deployment Pipelines

Creating secure images through deployment pipelines (which may be done within an image factory) is a structured process that guarantees virtual environments are built with security at their core. This methodology aligns with DevSecOps principles, integrating security as a fundamental component of the development lifecycle. Creating secure images is not a single action but a series of carefully orchestrated steps within a deployment pipeline.

The following figure illustrates the secure image deployment pipeline process, showcasing each step from source code to production deployment, integrating security measures throughout the development lifecycle.

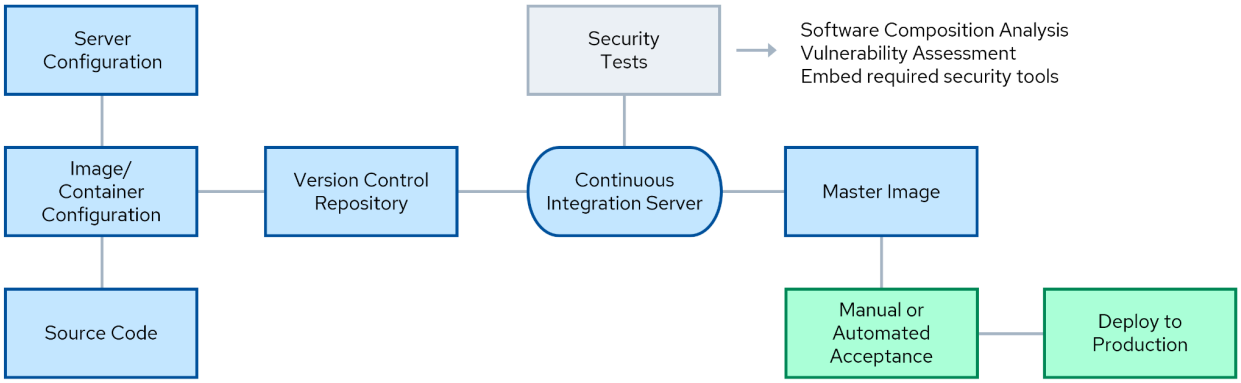


Figure 45: Secure Image Deployment Pipeline Process

The steps are:

1. **Source Code:** Any potential source code to be compiled and installed on the image.
2. **Server configuration:** The foundation of a secure image starts with a well-defined server configuration. Utilizing IaC, this step involves specifying the OS, network settings, and security policies that will form the baseline of the server environment.
3. **Image configuration:** The focus shifts to the image or container. This stage involves packaging the application and its dependencies in a lean and secure configuration based on the predefined server setup.
4. **Version control repository:** The journey of the image configuration files continues as they are checked into a version control system. Employing tools like Git repositories or container registries, this practice facilitates change tracking, collaboration, and accountability in the build process.
5. **Continuous integration server:** Automation takes center stage here. A continuous integration service or server builds images from the configuration files and performs security checks each time changes are committed.
6. **Security testing and enforcement:** Here, security testing becomes an integrated pipeline component. With tools for SCA and vulnerability scanning, the pipeline identifies and rectifies security issues, fortifying the image before it progresses.
7. **Master image:** A secure master image is born. The culmination of the process results in a hardened, vetted master image that is securely stored and ready for deployment.
8. **Manual or automated acceptance:** At this juncture, the image undergoes a rigorous examination. Depending on its risk and criticality, it may pass through manual reviews or automated acceptance tests.
9. **Deploy to Production:** The master image is in the production environment. The image's transition to production is consistent, secure, and deployed with precision using IaC and automated tools.

By embedding these steps into practices, organizations can confidently deploy secure images that have been scrutinized and prepared for the threats of the digital world. This approach minimizes vulnerabilities and ensures that security keeps pace with the speed of deployment, scaling efficiently across the entire environment. As we delve deeper into data security, these practices form a cornerstone for safeguarding information and maintaining robust defenses in the cloud.

8.2.4 Snapshots & Public Exposures/Exfiltration

Snapshots are essential in managing VM lifecycles, providing nearly instant copies of storage volumes for preservation and recovery. A snapshot is a saved state of a VM at a given moment, much like a detailed photograph of a workspace, capturing everything from files to settings. This also includes sensitive data. For this reason, snapshots need careful handling to prevent unauthorized access, inadvertent leaks, and data exfiltration.

Mitigating Public Exposure
While beneficial, snapshots' comprehensive nature introduces risks, especially when containing sensitive data. Enacting stringent access controls to manage who can create or retrieve snapshots is crucial. Imagine these controls as the equivalent of a master key – only trusted personnel should wield such power.

Encryption of snapshots adds an essential layer of security, like a cipher that renders a secret message unreadable to unintended recipients. Should snapshots inadvertently become public, the encrypted data remains protected and inaccessible without the corresponding decryption key.

Preventing Data Exfiltration
Maintaining snapshots also involves regular review, like shredding sensitive documents that are no longer required. This process bolsters security and optimizes cloud resource spending by eliminating unnecessary data storage.

Monitoring tools like Cloud Security Posture Management (CSPMs) act as vigilant guards over snapshots, scrutinizing who accesses or modifies them. Implementing alerts for unusual activities is comparable to having surveillance cameras trained on a vulnerable point, ensuring that unauthorized attempts are detected and addressed swiftly.

Snapshots must be accorded the same degree of security as the live systems they represent. Since they encapsulate all the data and configurations of a VM at their creation, they could become potential vectors for data leaks if mismanaged. An exhaustive approach to snapshot security goes beyond just safeguarding data – it also involves ensuring that these snapshots do not pose risks or liabilities.

8.3 Securing Containers

This section delves into the importance of building secure container images, orchestrating containers efficiently and safely, and managing the myriad security challenges that arise in containerized environments. It provides comprehensive insights into securing each step of the container lifecycle, from the creation of a container image to orchestrating its deployment with systems like Kubernetes.

8.3.1 Container Image Creation

A container image is a lightweight, standalone, and executable software package that includes everything needed to run an application: code, runtime, system tools, libraries, and settings. Container images are created from a set of instructions, typically defined in a Dockerfile[133], that specify the base OS, dependencies, and application code. These images can be easily shared and deployed across different environments, ensuring consistency and portability of the application.

Containers need to be built using secure, approved base images. Additional security can be added and assessed using tools that evaluate the instructions (Dockerfile). It is also important to ensure the security of the *artifact repository*, which is where the container images are registered and stored.

Containers inherently promote the concept of immutable infrastructures. Once a container image is built and deployed, it is not modified; updates and changes are made by replacing the container with a new image. This is comparable to swapping out a faulty component in machinery with a new one instead of mending it.

8.3.2 Container Networking

Container networking is an extension of the host operating system (often Linux) networking.

Kubernetes networking, and therefore network isolation, happens on multiple levels ranging from individual containers to application-aware load balancers (e.g., Ingress Controller). Various technologies exist for defining network policies. Again, some of these may be offerings of a provider, while they could also be self-managed.

8.3.3 Container Orchestration & Management Systems

Container orchestration systems have become essential tools for managing the complex lifecycle of containerized applications. Kubernetes (K8s) is a leading open-source platform among these systems, due to its flexibility and comprehensive feature set. Kubernetes orchestrates the deployment, scaling, and management of applications deployed in containers across machine clusters, enabling seamless automation and consistent operation. Containers host microservices (application components) and ensure components run in consistent environments.

[133] A Dockerfile is a text document that contains all the commands a user could call on the command line to assemble an image.

Major CSPs have adopted and adapted Kubernetes, offering customized versions tailored to their cloud environments (e.g., Amazon with EKS, Microsoft with Azure Kubernetes Service, Google with GKE). These services add proprietary features to the robust foundation of standard Kubernetes, offering users a blend of familiarity and provider-specific enhancements.

While using open-source Kubernetes, it is important to be cautious of the default settings as they can be insecure or non-aligned with your desired security posture. These defaults might include:

- Open dashboards, that can inadvertently expose valuable information if not properly secured
- Default service accounts with broad permissions that may grant more access than necessary
- Network configurations that fail to meet the stringent security requirements of a particular deployment

This image shows a basic Kubernetes architecture with the core management components, two "pods" where the containers run, and a load balancer through which users access the deployed application.

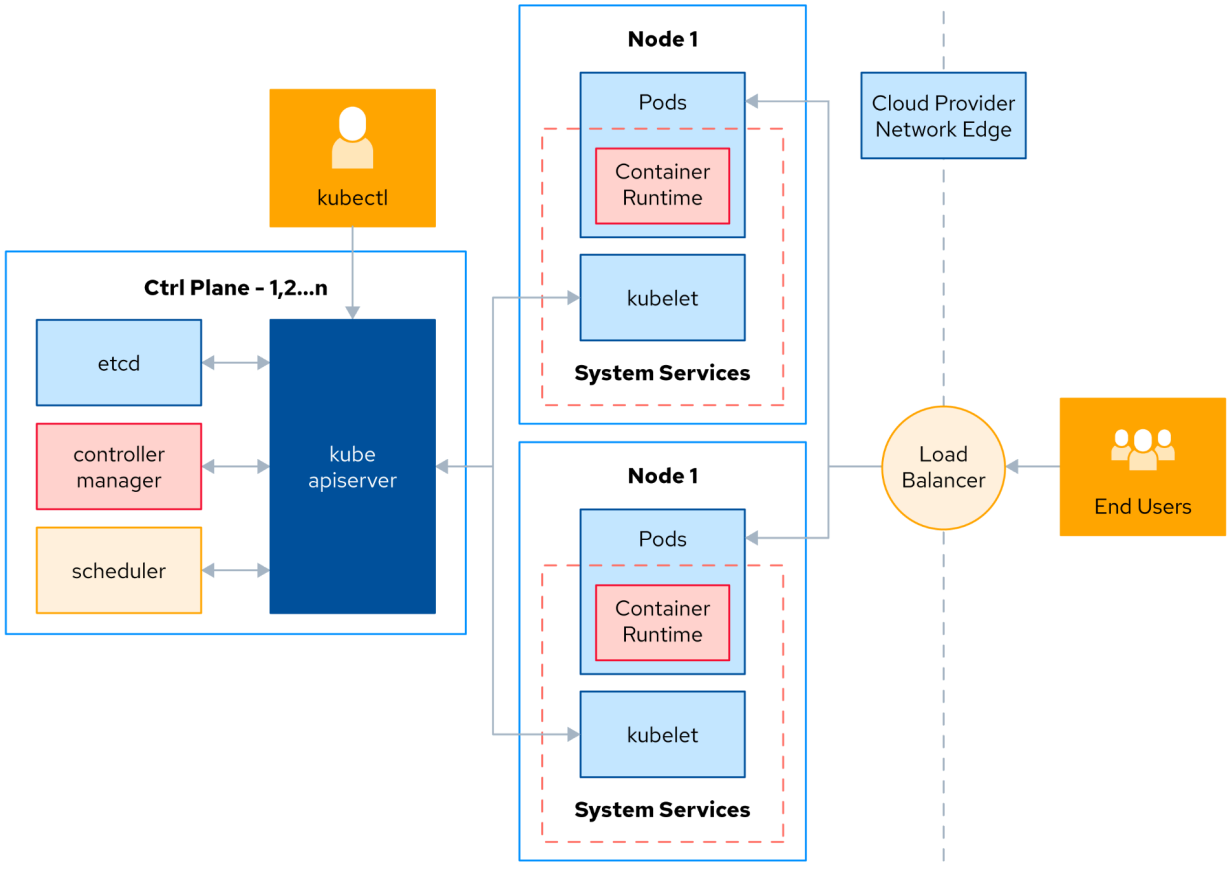


Figure 46: Basic Kubernetes Setup with Load Balancer and Pods

8.3.4 Container Orchestration Security

Container orchestration platforms like Kubernetes have become essential for managing containerized workloads in the cloud. However, securing these complex platforms can be challenging. Kubernetes consists of multiple components, application programming interface (APIs), and network interfaces that attackers can target if not properly configured and hardened. Misconfigurations, unpatched vulnerabilities, and overly permissive access controls can lead to breaches and compromises of the container environment.

Best practices include:

- **Use CSP services:** When deploying containerized applications, it is best to utilize CSP services if available. CSPs typically offer a suite of tools designed to automate and enhance security. These may include managed services for orchestration, like Kubernetes-as-a-Service, that come with default configurations that are security-focused and compliance-ready.
- **Harden services:** Service hardening is a proactive approach to minimizing the attack surface of a system. This involves securing the orchestrator by disabling unnecessary features, ensuring least privilege access, and implementing network policies and firewalls that restrict traffic between containers. Hardening should also encompass the use of secure base images for containers, employing security context settings within Kubernetes, and utilizing so-called "admission controllers" to enforce good security practices.
- **Patch/update:** Regular patching and updating of all components in the container ecosystem is crucial. This includes not only the containers themselves but also the hosts, orchestration platforms like Kubernetes, and other supporting services. Automation of patch management processes can help ensure that patches are applied as soon as they are available, thus mitigating the risk of vulnerabilities being exploited.
- **Container security policies:** Define and enforce security policies using tools like Kubernetes Security Policies, PodSecurityPolicy, and Network Policies to restrict and monitor network access between pods.
- **Utilize security benchmarks and tools:** Standards and standardization tools, such as the CIS benchmarks[134] for Kubernetes, provide a structured approach to detecting and remedying insecure defaults.
- **Secure image repository:** A secure image repository is central to container security. Use private repositories with role-based access controls (RBAC) to manage who can push and pull images. Implement scanning and vulnerability detection for container images both pre-deployment and routinely in production. Utilize image signing to establish a chain of trust, to verify that images have not been tampered with from the time they were created to the time they are deployed.
- **Secure configuration:** To ensure the integrity and security of containerized environments, it is critical to begin with robust and secure configurations. These foundational settings cover every aspect of the container environment and are integral to delivering a security environment.

[134] CIS. (2024) Center for Internet Security: *CIS Benchmarks*

- *Cluster hosts:* Protect cluster resources by hardening the host's OS, maintaining a minimal installation, and ensuring host-level security controls are in place.
- *Storage layer:* Apply encryption for data at rest and in transit, use access control lists (ACLs) or policies to limit access to persistent volumes, and employ logging to monitor access to sensitive data.
- *Network layer:* Implement network segmentation and firewalling to control the flow of traffic between services. Use network policies to enforce rules on how containers communicate with each other and with external networks.
- *Image validation/signing:* Incorporate steps in your CI/CD pipeline to validate and sign images. This can involve using tools that check for known vulnerabilities, and ensuring that only images signed by a trusted authority are run by orchestrators.

8.3.4.1 Secure Artifact Repositories

Secure artifact repositories act as vaults for software components, including container images, ensuring that:

- Repositories enforce content trust mechanisms like digital signatures and verification to guarantee the authenticity and integrity of container images.
- Access is strictly controlled, allowing only verified users to push or pull images, much like a bank permitting transactions from only authenticated customers.
- Images are routinely scanned for vulnerabilities, akin to performing regular health check-ups to catch signs of illness early.
- Container images should be immutable.
- The provenance of images is thoroughly documented and protected, offering a clear lineage of their origins and makers, similar to a well-kept public registry.
- Secure artifact repositories seamlessly integrate with continuous integration and deployment pipelines to automate the scanning and validation of container images before deployment.

8.3.4.2 Best Practices for Using Secure Repositories

Securing artifact repositories entails diligent practices that reflect broader cybersecurity principles, such as those below.

- **Only enable secure sources:** Just as one would avoid using dubious spare parts of unknown origin for important machinery, developers should only use images from secure and trusted sources.

- **Sign and verify images:** Digital signatures serve as seals of authenticity, confirming that an image is what it purports to be and remains untampered with, similar to a wax seal on a letter in historical times.

- **Scan for vulnerabilities:** Before deployment, images should undergo thorough vulnerability scanning, which can be likened to a comprehensive pre-flight inspection for aircraft.

- **Access control:** Restricting repository access ensures that only those with a legitimate need can retrieve or alter images, likened to accessing sensitive information in a secure facility.
- **Audit trails:** Keeping meticulous records of who accessed or modified repository contents is crucial. This provides transparency and aids compliance, much like a ship's log captures events that occur on board.
- **Regular updates:** Continuously update and patch repository software to protect against known vulnerabilities and maintain a secure environment for storing images.

Securing container images is an exercise in establishing a solid foundation, enforcing stringent processes, integrity, and meticulous record-keeping. By embedding security within each stage of the container image lifecycle, organizations can fortify their containerized applications against threats and meet compliance mandates.

8.3.5 Managing Container Vulnerabilities

Securing modern software deployment processes involves managing container vulnerabilities. Like any technology, containers present a unique set of potential security issues requiring systematic management.

The following are some key considerations when managing container vulnerabilities:

- **CI/CD pipeline integration:** Integrating vulnerability management tools into CI/CD pipelines is akin to embedding a rigorous quality assurance process in a production line. This integration ensures that at every stage of a container's development and deployment, it undergoes thorough security checks, much like the meticulous inspection that each product component undergoes before moving down the assembly line.
- **Regular updates:** It is imperative to keep container images and their dependencies up-to-date:
- **Immutable containers:** The principle of immutability in container management is a vital defensive tactic. Once a container is deployed, it is not altered; any necessary updates result in the deployment of a new container. This method is like using replaceable parts in machinery to ensure optimal performance, rather than making ad-hoc fixes.
- **Security policy enforcement:** Implementing and enforcing security policies that dictate using pre-scanned and approved images for deployment creates a secure gate, similar to a selective bouncer who ensures that only verified guests are allowed entry into a venue.
- **Role-based access control (RBAC):** RBAC regulates access to container management tools and resources. It guarantees that team members are only granted the access necessary to fulfill their roles – no more, no less. This is akin to issuing different keys for different areas within a secured facility, restricting access based on one's responsibilities.

- **Attribute-based access control (ABAC[135]):** ABAC provides a more granular approach specifically suited to containerized environments due to their inherent flexibility and dynamic nature. With ABAC, access decisions are made by considering attributes in addition to roles. These attributes can include user location, device type, data classification (e.g., sensitive, public) information stored within the container, or other relevant characteristics. This allows for more flexible and dynamic access control within a containerized cloud environment.

Embedding these practices into a container's lifecycle — from its inception in development to its deployment and beyond — helps create a fortified workflow. It supports the idea that security is not just an afterthought but an integral part of the process. Additionally, by implementing RBAC, organizations can maintain a secure and efficient workflow while ensuring that the right individuals have the appropriate level of access at all times.

8.3.6 Runtime Protection for Containers

Runtime protection for containers ensures that potential threats or malfunctions are detected and managed as they occur, keeping the containerized applications secure and running smoothly.

There are several important aspects of runtime protection for containers:

- **Real-time visibility:** Effective runtime protection begins with real-time visibility. Monitoring tools are watchful eyes that observe container activities continuously, scanning for any unusual behavior that could indicate a security threat or an operational anomaly.
- **Logging and auditing:** Meticulous logging and auditing creates a detailed logbook of container activities and user interactions. Log records are invaluable for post-incident analysis, serving the same purpose as security camera footage in crime investigations.
- **Microsegmentation:** To minimize the impact of a breach, network segmentation is implemented, creating isolated compartments for containers, similar to the watertight sections of a ship. As with water, the threat is contained in the event of a breach.
- **Container-specific firewalls:** These firewalls act as traffic regulators, establishing and enforcing rules to manage network traffic flow. These are akin to strategically placed checkpoints that control vehicle entry and exit, ensuring order and security.
- **Automated responses:** The final aspect is the capability for automated responses. This emergency protocol snaps into action when a threat is detected, isolating compromised containers, denying access, or reverting systems to a known good state, much like an automated defense system that reacts to intrusions without human intervention.

Runtime protection is about constant vigilance, creating a robust and responsive system that watches over containers and quickly neutralizes any threats through proactive and reactive security measures, ensuring the integrity and resilience of the containerized applications throughout their lifecycle.

[135] NIST. (2024) *CSRC Projects: Attribute Based Access Control (ABAC).*

8.4 PaaS Security

PaaS from CSPs often includes services that replace workload components (e.g., a message queue service that replaces the need for queueing software on a server) and supportive hosting platforms for running workloads themselves, like containers. It is not unusual for a PaaS service to automate and orchestrate standard software stacks on VMs, like a database service for SQL Server or Oracle, that manages the underlying VMs so the customer only needs to manage configuration settings and their database.

In other words, PaaS covers a very broad range of options from a service that automates and coordinates common software platforms, to ones that host arbitrary workloads like containers or serverless functions, to services that fully abstract a capability, like a message queue, without the customer ever knowing what is running underneath the hood.

8.4.1 General Security Practices for PaaS

The security of PaaS hinges on a multi-layered approach, integrating general security practices with specific measures tailored to unique PaaS environment components:

- **Security audits:** Regular vulnerability assessments, or health checks, of PaaS components are essential for identifying and mitigating potential security threats. These audits should be conducted routinely to adapt to evolving threats and changes within the PaaS environment.
- **Logging and monitoring:** Effective security hinges on visibility. Implementing comprehensive logging and real-time monitoring of activities within the PaaS platform allows for the early detection of suspicious behaviors or potential breaches, facilitating rapid response and mitigation efforts.
- **Least privilege:** Adhering to the principle of least privilege minimizes the risk of unauthorized access or data breaches. Organizations can significantly reduce their attack surface by granting users and services only the minimum access levels necessary for their roles.
- **Multi-factor authentication (MFA):** Strengthening access controls with MFA adds a layer of security, making it significantly more challenging for attackers to gain unauthorized access. This approach is akin to a bank requiring both a card and a PIN for transactions, enhancing the security of sensitive operations.
- **Access reviews:** Periodic re-evaluation of access permissions ensures that only the appropriate individuals and services can access critical resources. This process helps promptly revoke access no longer required, further tightening the security posture.

8.4.2 Encryption & Access Controls

In PaaS security, managing identities, encrypting data, and controlling access form the pillars of a sturdy security posture. This section explores the integral roles of encryption and access controls in securing PaaS environments.

Encryption: Protecting data at rest and in transit through robust encryption methods is like securing valuables in a safe and providing secure transport. Managing encryption keys with utmost care guarantees that only authorized entities can access the encrypted data.

Access Controls:

- **Network Segmentation and Firewalls:** Implementing network segmentation and deploying firewalls helps create secure zones within the PaaS environment, controlling traffic flow and reducing the potential impact of breaches.
- **RBAC:** Systems assign access based on specific roles, ensuring that individuals or services have access only to the resources necessary for their assigned functions.
- **ABAC:** Assigns access based on attributes allowing for more flexible and dynamic access decisions in a cloud environment.
- **API Gateway Policies:** Stringent policies for API gateways control how external entities interact with the PaaS, similar to a bouncer managing entry at a club.

These practices mean building a multi-layered defense strategy for PaaS environments to ensure they are as resilient as possible against various security threats.

8.4.3 Securing Specific PaaS

Beyond general security measures, certain PaaS platforms demand specialized protection strategies due to their unique vulnerabilities. This section focuses on securing specific PaaS, such as CDNs, notification services, and message queues, each requiring tailored security measures to protect against threats.

- **Content delivery networks (CDNs):** Secure sockets layer (SSL) or transport layer security (TLS) encryption for data moving through CDNs ensures that the information remains confidential and unaltered. Strong access control and authentication mechanisms limit access to the stored content.
- **Notification services:** Encrypting notifications and using secure delivery channels protect the information within, similar to sending sensitive documents through a trusted courier. Strong authentication methods verify the legitimacy of services and users authorized to send notifications.
- **Message queues:** Encryption of messages at rest and in transit, as well as secure access policies and RBAC, safeguard sensitive data in message queues. This guarantees that only authorized entities can publish or subscribe to the queues, maintaining communication integrity.

It is essential to understand that each PaaS component has unique vulnerabilities and requires tailored security measures to protect data integrity, ensure privacy, and maintain reliable service operations. The security of PaaS demands a diligent, detailed approach to addressing general and service-specific vulnerabilities. By implementing these strategies, organizations can build a resilient defense against various security threats, ensuring their cloud-based applications' and services' integrity, privacy, and reliability.

8.5 Securing Serverless or Function as a Service

Serverless computing, commonly known as function as a service (FaaS), is a way for developers to write and deploy code without handling the underlying infrastructure. The cloud provider manages the servers, which includes provisioning them, scaling them to handle different loads, and maintaining them. This lets developers focus purely on coding, without worrying about the underlying work that goes into server management.

The term "serverless" is somewhat of a misnomer because servers are still used to run applications. However, managing these servers does not fall on the application owners. Instead, it is abstracted away by the CSP. This shift away from a traditional focus on infrastructure means developers only pay for the computing power they use, often billed down to the exact number of milliseconds of code execution.

The primary advantage of serverless computing is its operational simplicity. Developers provide the code, and the cloud provider takes care of the rest, including all the operational aspects like system maintenance and scalability. The system automatically adjusts computing resources based on the application's needs, providing a highly flexible and efficient solution for developers who want to build and scale applications quickly and with minimal overhead.

Each function in serverless is typically executed within a lightweight, single-use container that resides on a single-use virtual machine. This method ensures that each function invocation operates in a distinctly isolated environment, promoting strong isolation and preventing any interference between functions. The ephemeral nature of these execution environments significantly reduces the attack surface as there is no persistent OS to manage, thereby minimizing potential security risks. Furthermore, this model enhances security by ensuring that each function's execution is both isolated and transient.

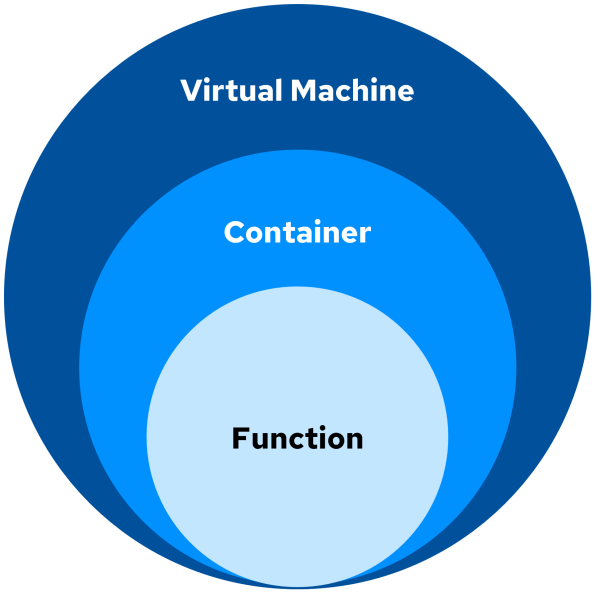


Figure 47: The FaaS Paradigm

Nonetheless, developers must remember that, despite the cloud provider assuming many of the security responsibilities, they must still secure their application code, effectively manage access controls, and safeguard sensitive data that is

transferred to and from the functions. By correctly leveraging FaaS, developers can focus on coding while relying on the cloud provider to manage infrastructure security, making FaaS a secure and scalable option for executing application logic with reduced operational complexity.

The figure illustrates the relationship between the function, container, and virtual machine in the FaaS model:

- The innermost circle represents the individual function, which contains the application code.
- The function is encapsulated within a container, which provides the necessary runtime environment and dependencies.
- The container is then executed on a virtual machine managed by the CSP.

8.5.1 FaaS Security Issues

When it comes to serverless computing, there are common security issues to be aware of.

- **Third-party services and APIs:** creates the potential for attacks. If these interfaces are compromised, attackers could grant them the power to make unauthorized configurations or spy on the cloud environment.
- **Vulnerable dependencies:** Serverless functions often depend on external libraries, which can harbor vulnerabilities or malicious code. If these dependencies are not rigorously checked and updated, they can serve as backdoors for attackers.
- **Misconfigurations:** Incorrect or overly permissive configurations can inadvertently open access to sensitive resources within a serverless architecture. Security settings related to who can execute functions and what those functions can access must be tightly controlled to restrict actions.
- **Overly-privileged IAM for a function:** When functions are granted excessive permissions, it significantly heightens the risk of unauthorized access and potential data breaches. Such configurations can allow functions more access than necessary, enabling attackers to exploit these privileges for nefarious purposes.
- **Direct Internet access for functions:** Functions may have direct Internet access without proper network controls, such as network segmentation and ACLs. This lack of restriction not only exposes functions to external threats but also becomes a potential channel for data exfiltration by external entities.

Each issue above underscores the need for careful security practices within the serverless model, from vetting third-party APIs to diligently managing dependencies and configurations. Despite the serverless model's advantages in scalability and cost, vigilance in these areas is essential to protect against vulnerabilities. Moreover, the unique serverless environment results in specific security considerations that are not issues in other workload types.

The following are unique serverless security considerations:

- **Stateless nature:** Serverless functions operate without retaining an internal state, significantly altering the security approach. Without a persistent server environment to monitor, the spotlight turns to the security of the code itself and its various dependencies. This shift demands a thorough understanding of how the code is written, the libraries it calls, and the data it processes. Developers must ensure their functions are self-contained, with all necessary security measures in the execution environment.

- **Event-driven security:** The event-driven nature of serverless computing presents unique challenges. Serverless functions are typically executed in response to events, which could range from a user request to a scheduled task. This model requires stringent validation of events to prevent malicious triggers. Careful crafting and verification of the event input become critical to ensure that functions only respond to legitimate and intended triggers. Ensuring the security of these events involves scrutinizing event sources and enforcing rigorous validation checks before allowing a function to execute.

- **Reliance on CSPs:** In serverless architectures, the security measures available are often defined by the offerings of the CSP. This limitation means that understanding and leveraging the shared responsibility model is key. While the CSP secures the cloud infrastructure, the CSC must focus on securing their code and data. It is vital to know which security aspects the provider handles and which responsibilities fall on the CSC, such as identity and access management (IAM), code security, data encryption, and policy enforcement. Navigating this shared landscape means staying informed about the CSP's tools and services and integrating them effectively.

Navigating serverless security effectively requires adapting to a model where the server management is out of one's control, yet the responsibility for securing the code and execution environment remains. This adaptation involves relying on the CSP's tools and rigorously following best configuration, event management, and dependency security practices. Understanding the nuances of statelessness, event-driven triggers, and the shared responsibility model is key to crafting a secure serverless application.

8.5.2 IAM for Serverless

IAM is a cornerstone of securing serverless architectures. As serverless applications can integrate and interact with various services across domains, managing trust and access becomes complex. Establishing and maintaining stringent IAM practices to safeguard against unauthorized access and potential breaches is crucial.

The following are some IAM best practices for serverless architectures:

- **Least privilege access:** In serverless computing, the principle of least privilege must be implemented. This means giving functions the minimum level of access, or permissions, required to operate. Regularly updating these permissions ensures that functions don't have unnecessary access rights, which could expose sensitive systems or data to threats.

- **Fine-grained access control:** In addition to RBAC, which governs access based on predefined roles, serverless environments benefit from fine-grained access controls. This approach enables precise specification of permissions at the level of individual functions or resources, ensuring least privilege access and reducing the attack surface.
- **Context-aware authorization:** Serverless architectures lend themselves to context-aware authorization, going beyond traditional RBAC. Contextual attributes, such as user identity, device characteristics, time of access, and environmental factors can influence access decisions dynamically. Implementing context-aware policies enhances security by adapting access controls based on real-time circumstances.
- **Immutable infrastructure and secret management:** Serverless functions are stateless and ephemeral. Best practices include leveraging secrets-management services provided by CSPs, rotating credentials regularly, and adopting immutable infrastructure principles to mitigate the risk of credential exposure.
- **Review and update IAM policies:** It is also vital to routinely review and update IAM policies to ensure that permissions align with current requirements. As serverless applications evolve, so too do their access needs. Regularly auditing these policies ensures that permissions are neither too lax nor unnecessarily strict, balancing operational efficiency with security.

For teams looking to secure their serverless functions, embracing these IAM best practices and keeping an eye on emerging industry solutions like Security Production Identity Framework For Everyone (SPIFFE) or SPIFFE Runtime Environment (SPIRE)[136] is vital. These solutions create a secure, manageable, and reliable serverless environment that can scale safely across multiple platforms and domains.

8.5.3 Network Connectivity & Access Patterns

Network design plays an integral role in the security of serverless architectures. Isolating serverless functions within virtual networks enhances security by reducing the risk of unauthorized access. Fine-grained access controls, such as ACLs, can be established to define who or what can access these functions and under what conditions.

Securing the interaction between serverless functions and other services is also essential. API gateways are often the entry point for incoming requests and must be rigorously secured. In addition to ensuring the robustness of API gateways, it is critical to encrypt data in transit. Although network security is largely under the CSP's control, the CSC must configure the security settings that protect the application layer's data movement.

[136]SPIFFE. (2024) SPIFFE is a set of open-source standards for securely identifying software systems in dynamic and heterogeneous environments. SPIRE is a production-ready implementation of the SPIFFE APIs.

8.5.4 Environment Variables & Secrets

The handling of sensitive information within serverless applications requires careful consideration. Environment variables[137] should be utilized instead of hardcoding secrets, such as passwords or API keys, into the code. These variables can be dynamically managed and injected at runtime, minimizing the exposure of sensitive information.

Cloud services like AWS Secrets Manager or Azure Key Vault provide reliable mechanisms for secrets management, allowing for the secure storage, retrieval, and rotation of credentials. Regularly rotating these secrets reduces the risk of older, possibly compromised credentials being exploited. Additionally, controlling access to these secrets through IAM roles guarantees that only authorized entities can retrieve or alter them.

By adopting these practices, a serverless environment can be constructed whereby network connectivity and sensitive data are managed securely and efficiently. This helps maintain the integrity and confidentiality of serverless applications, adhering to a Zero Trust security model where trust is never assumed and must be continually verified.

8.6 AI Workloads

AI stands at the forefront of technological advancement, transforming how we live, work, and interact. AI workloads refer to the tasks, processes, or operations that are involved in building, delivering, or utilizing AI capabilities. These workloads enable machines to learn from data, make predictions, and simulate human intelligence in decision-making processes. From recommending products based on user behavior to autonomously piloting vehicles, AI workloads encompass a wide range of complexities and applications.

AI workloads are characterized by their intensive data requirements and computational complexity. They require large datasets for model training and substantial processing power, leveraging specialized hardware like Graphics Processing Units (GPUs) and Tensor Processing Units (TPUs) for efficiency. Additionally, these workloads must scale dynamically to accommodate fluctuating demands, underscoring the importance of flexible computing resources, such as those provided by cloud environments.

The applications of AI workloads are vast and varied. They reshape industries by automating tasks, enhancing customer experiences, and offering unprecedented insights into complex problems. As AI technologies evolve, understanding and managing these workloads becomes crucial for organizations aiming to leverage AI's full potential. The journey into AI workloads is not just about harnessing computational power but also about navigating the intricacies of data, algorithms, and real-time processing to unlock innovation and value across sectors.

[137] Environment variabilities are dynamic values that are used to configure application settings, manage secrets, and control behavior without altering the source code, allowing for easier deployment and environment-specific customization.

8.6.1 AI-System Threats

The security of AI infrastructure is a critical concern due to the potential consequences of a breach. This infrastructure has multiple components, each presenting unique challenges and requiring tailored security measures. By understanding the specific threats and implementing appropriate mitigation strategies, the integrity, confidentiality, and availability of AI systems can be ensured.

Following are some key AI system threats grouped by category:

Data Security Threats:

- *Data poisoning:* Poisoning the data involves maliciously introducing false information, leading to inaccurate model outputs.
- *Privacy breaches:* Unauthorized access to sensitive data can result in privacy violations and related legal issues.
- *Data leakage:* Accidentally exposing training data can occur through model outputs, risking disclosing confidential information.

Model Security Threats:

- *Model theft:* unauthorized copying of a machine learning model. This allows the attacker to circumvent intellectual property laws and could also reveal how to trick the model, compounding the risk.
- *Adversarial attacks:* Inputs can be manipulated to exploit design weaknesses and cause incorrect predictions.
- *Model inversion attacks:* These attacks can reconstruct input data from model outputs, threatening the confidentiality of the training data.
- *Prompt injection:* Maliciously crafted inputs can exploit AI model vulnerabilities to trigger unintended actions or reveal sensitive information, similar to how social engineering tricks individuals into compromising security.

Infrastructure Security Threats:

- *Unauthorized access:* Intrusions into AI infrastructure can lead to data theft, malicious alterations, or the deployment of harmful software.
- *DDoS attacks:* Overwhelming traffic can be used to disrupt services.
- *Hardware vulnerabilities:* Exploits targeting GPUs and TPUs can include side-channel attacks, risking the disclosure of sensitive information.

Supply Chain Threats:

- *Software dependencies:* Third-party libraries may introduce vulnerabilities or malicious code.
- *Third-party services:* Relying on external data processing and storage services can introduce vulnerabilities.

8.6.2 AI Mitigation Strategies

The following are some key AI system migration strategies grouped by category.

Data Security:

- *Encryption:* Protect data confidentiality during transmission and storage.
- *Differential privacy:* Introduce randomness into data or queries so that individual records can't be traced back to a person. It's like adding noise to a conversation to mask private details.
- *Secure multi-party computation:* Process data from multiple sources without exposing sensitive information by anonymizing or tokenizing sensitive information as part of the flows.
- *Confidential computing:* Use Trusted Execution Environments[138] to safeguard data during processing and protect AI model execution.

Model Security:

- *Model hardening:* Defend against adversarial attacks to enhance model resilience.
- *Robust training:* Employ techniques to improve generalizability and reduce overfitting[139].
- *Adversarial training:* Strengthen AI models against attacks by incorporating manipulated examples into their training data, enhancing their resilience, much like a fighter learning to counter different moves.
- *Model watermarking:* Embed unique identifiers to assert ownership and deter theft.
- *Output manipulation:* Altering the AI's responses to obscure its decision-making process can thwart potential thieves, much like a poker player's bluff.

[138] TEEs are secure areas within a processor that ensure code and data loaded inside are protected with confidentiality and integrity, providing a safe execution environment resistant to software and hardware attacks.
[139] A scenario where a machine learning model learns the training data too well, including noise and outliers, resulting in poor generalization to new, unseen data, and potentially making the model vulnerable to adversarial attacks.

Infrastructure Security:

- *GPUs and TPUs:* To maintain system integrity, utilize hardware-based security features, regular firmware updates, and network security measures.
- *AI services:* Follow best practices for cloud services, including access controls and real-time monitoring.
- *Quotas and rate limiting:* Apply quotas and rate limiting to identify and prevent DoS and DDoS attacks.

Supply Chain Security:

- *Policies:* Define and approve a cybersecurity policy for the supply chain.
- *Software supply chain risk management:* Regularly audit and update third-party dependencies.
- *Vetting third-party Services:* Conduct security assessments before integration.
- *Trusted sources:* Rely on reputable sources for software dependencies, maintaining an approved list.

By proactively addressing these threats with the outlined strategies, organizations can fortify their AI infrastructure against current and emerging dangers, ensuring the resilience of their AI systems.[140]

Summary

Cloud workload protection is an evolving discipline that addresses the unique security challenges found in the diverse and dynamic nature of cloud environments. Traditional security measures are insufficient in the cloud; hence, specialized controls are necessary to safeguard various workloads effectively.

For VMs, security begins at the image level. VM image security automation helps embed protections early in the deployment cycle. Practices like enforcing the principle of least privilege and prioritizing regular vulnerability assessments are foundational in maintaining robust defense against threats.

When it comes to container orchestration with Kubernetes, customizing configurations to enhance security is crucial. Scanning container images for vulnerabilities and controlling who has the authority to access and manage these images are vital practices. Additionally, implementing runtime protection mechanisms ensures containers are continuously monitored and defended against ongoing threats.

Serverless applications demand a focused approach to security, starting with rigorous IAM policies. Securing API endpoints against unauthorized access and managing secrets with high scrutiny is key to preventing exploitation.

[140] MITRE. (2024) *Adversarial Threat Landscape for Artificial-Intelligence Systems (ATLAS)* is a globally accessible, living knowledge base of adversary tactics and techniques against AI-enabled systems. It is based on real-world attack observations and realistic demonstrations from AI red teams and security groups and can help in the safeguarding of AI infrastructures.

The field of AI workload security is particularly fast-paced and requires continuous learning. To protect AI workloads from compromise, it is essential to incorporate adversarial training, safeguard AI models from unauthorized access or theft, and employ data privacy techniques such as differential privacy.

For PaaS environments, regular security audits are necessary to identify and remediate vulnerabilities. Data encryption, both at rest and in transit, alongside strict IAM controls and secured communication channels, forms the foundation

Cloud workload protection is not a one-size-fits-all solution, rather a tailored approach that adapts to the characteristics of each workload type. Staying current on the latest threats and mitigation strategies is essential for maintaining a secure cloud ecosystem.

Recommendations

Cloud Workload Management

- Create a centralized Cloud Deployment Registry: Maintain a comprehensive inventory of all cloud workloads and deployments for efficient tracking and management.
- Define an organization hierarchy using multiple deployments: Structure cloud environments to mirror organizational units for better security and administrative control.
- Support a low-friction process for creating new deployments: Streamline processes to ensure adherence to security policies without impeding operational efficiency.

Virtual Machine (VM) Security

- Enforce secure base VM images: Use centrally managed, versioned, and immutable base images for all deployments.
- Implement image factories: Automate the creation, testing, and deployment of VM images to ensure consistency and security.
- Scan VM images for vulnerabilities: Regularly scan and update VM images to mitigate security risks.
- Adopt short-running VMs: Use immutable infrastructure and ephemeral VMs to reduce risks associated with long-running instances.
- Use configuration management and Infrastructure as Code (IaC): Maintain desired states and prevent configuration drift.
- Implement host-based firewalls and SSH hardening: Control network access and secure SSH configurations on VM instances.

Container Orchestration Security

- Use CSP services for orchestration: Utilize managed services like Kubernetes-as-a-Service for enhanced security.
- Harden orchestration services: Disable unnecessary features, ensure least privilege access, and implement network policies and firewalls.

- Regular patching and updates: Automate patch management for containers, hosts, and orchestration platforms.
- Define and enforce security policies: Use tools like Kubernetes Security Policies, PodSecurityPolicy, and Network Policies.
- Utilize security benchmarks and tools: Follow CIS benchmarks for Kubernetes to ensure secure configurations.
- Protect cluster hosts and storage: Harden the host's OS, apply encryption for data at rest and in transit, and use access control lists (ACLs).

Monitoring and Assessment

- Utilize CSPM tools: Continuously monitor cloud security posture using Cloud Security Posture Management (CSPM) tools.
- Implement continuous monitoring: Use real-time monitoring tools to track workload activities and detect potential security incidents quickly.
- Use SCA tools: Integrate Software Composition Analysis (SCA) tools into the CI/CD pipeline to manage dependencies and identify vulnerabilities early.
- Generate and maintain an SBOM: Create a Software Bill of Materials (SBOM) for all workloads to enhance transparency, security response, and regulatory compliance.
- Endpoint Detection and Response (EDR) agents: Perform runtime monitoring and support vulnerability assessment.
- Security Information & Event Management (SIEM): Provide real-time monitoring and reporting.

Training and Awareness

- Conduct regular security exercises: Perform scenario-based exercises and tabletop drills to prepare teams for real-world incidents.
- Encourage a Just Culture approach: Focus on systemic improvements and accountability without assigning undue blame for security incidents.

PaaS Security

- Regular security audits: Conduct vulnerability assessments to identify and mitigate potential threats.
- Comprehensive logging and monitoring: Implement logging and real-time monitoring to detect and respond to suspicious behaviors.
- Principle of least privilege: Grant users and services only the minimum access levels necessary.
- Multi-factor authentication (MFA): Enhance access controls with MFA.
- Periodic access reviews: Regularly re-evaluate access permissions to ensure appropriate access levels.

Securing Serverless or Function as a Service (FaaS)

- Vet third-party services and APIs: Ensure they are secure and reliable to avoid unauthorized configurations or data exposure.
- Manage vulnerable dependencies: Regularly update and check external libraries for vulnerabilities or malicious code.

- Correct misconfigurations: Ensure security settings restrict function execution and access appropriately.
- Limit IAM privileges for functions: Grant the minimum permissions necessary to reduce the risk of unauthorized access and data breaches.
- Control direct Internet access: Implement network segmentation and ACLs to prevent functions from accessing the Internet directly.

AI Mitigation Strategies

- Data security: Use encryption, differential privacy, and secure multi-party computation to protect data.
- Model security: Harden models against adversarial attacks, use robust training techniques, and embed unique identifiers to deter theft.
- Infrastructure security: Implement quotas and rate limiting, and follow best practices for cloud services.
- Supply chain security: Define cybersecurity policies, regularly audit third-party dependencies, and use trusted sources.

Additional Guidance

- Cloud Industrial Internet of Things (IIoT) - Industrial Control Systems Security Glossary | CSA
- Best Practices in Implementing a Secure Microservices Architecture | CSA
- Cloud Adversarial Vectors, Exploits, and Threats (CAVEaT™): An Emerging Threat Matrix for Industry Collaboration | CSA
- Integrating SDP and DNS: Enhanced Zero Trust Policy Enforcement | CSA
- Ransomware in the Healthcare Cloud | CSA
- Cloud Security Complexity | CSA
- The 12 Most Critical Risks for Serverless Applications | CSA

Domain 9: Data Security

Introduction

The rapid expansion and adaptation of cloud services and the increasing sophistication of cyber threats demand a resilient approach to safeguarding information. Data security practices are pivotal in preserving organizational integrity, confidentiality, and customer trust while also ensuring compliance with regulatory requirements.

This domain delves into the complexities of data security within the cloud, exploring essential strategies, tools, and practices that organizations can adopt to ensure their data remains protected, in-transit, and at rest. From understanding the nuances of data classification and cloud storage types to implementing advanced encryption methods and access controls, this section provides a guide for navigating the ever-evolving data security landscape. This domain is also a primer on cloud storage. Additionally, we'll explore key concepts and technologies that are shaping the future of how data is secured in cloud environments, ensuring that readers understand the critical measures needed to prevent data breaches and uphold data privacy.

Learning Objectives

The learning objectives for this domain aim to provide readers with knowledge on:

- Understanding data security fundamentals.
- Data classifications and states.
- Cloud storage types and their related security measures.
- Data security techniques, such as key management.
- Protecting various types of computing workloads.
- Posture management.
- Advanced data security concepts.

9.1 Data Classification & Storage Types

By categorizing data based on its type, sensitivity, and criticality, organizations can implement proper security methods per data type. Improper handling of data can lead to data breaches, compliance violations, and data loss. From a strategy perspective, understanding and implementing data classification practices help organizations align with operational and compliance strategies.

As organizational needs and regulatory landscapes evolve, data classification must adapt, ensuring that data governance initiatives remain effective and responsive to incidents. Additionally, recognizing different data states—at rest, in motion, and in use—requires tailored security measures. Coupled with this, comprehending various cloud storage types, such as object storage, volume storage, database storage, Software as a Service (SaaS) storage, and PaaS-specific storage, allows organizations to choose the most suitable solutions for their specific data needs and security requirements.

9.1.1 Data Classification

Data classification is a vital, ongoing process involving data categorization based on type, sensitivity, criticality, and potential exposure impact, considering both operational and compliance viewpoints. Incorporating data classification processes in an organization's data governance practices is crucial to protecting the entire data lifecycle. A robust data classification approach aids in addressing and mitigating the risk of data breaches by providing a clear understanding of asset protection priorities. In essence, data classification drives the definition of operational security and compliance strategies.

As organizations evolve, the operating environment changes and new laws and regulations impact the requirements under which the organization operates. Pairing a robust data classification strategy with clear asset and data ownership assignments empowers an organization to respond swiftly to incidents and successfully propel diverse data governance initiatives forward.

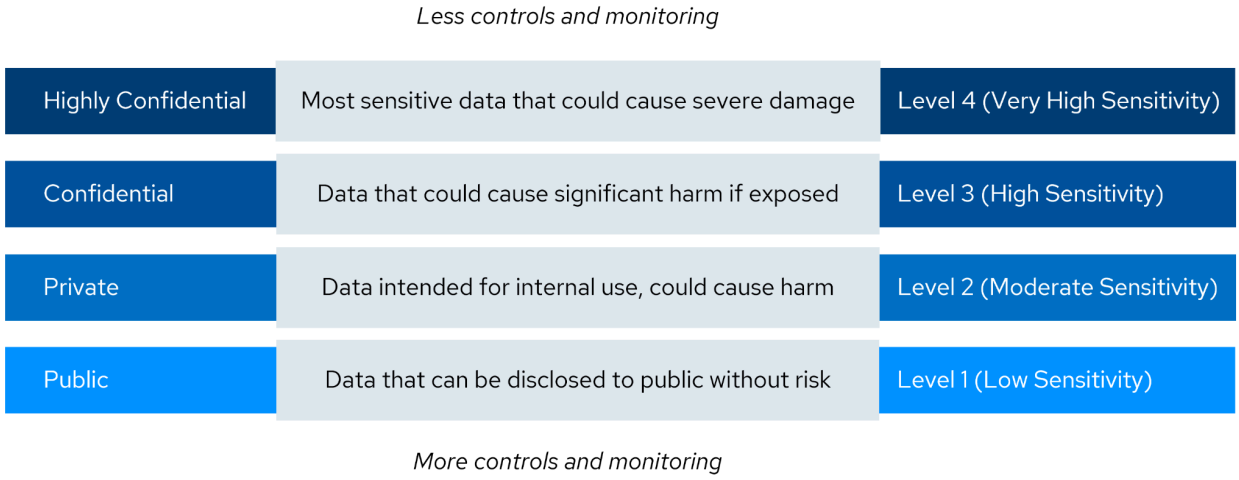


Figure 48: Data Classification Scale

9.1.2 Data States

In the context of cloud security, it is essential to recognize that data continuously transitions through different states, each requiring specific security measures. This discussion highlights the three main data states: data at rest (in storage), data in motion, and data in use.

Data at rest pertains to data that is kept in various forms within the cloud environment. This encompasses data held in volumes, such as virtual disks, objects like files in object storage, databases,

and platform as a service (PaaS) offerings. Effective security measures for data in storage typically involve encryption, establishing access controls, and maintaining regular backups to safeguard data integrity.

Data in motion refers to data actively being transmitted or transferred between locations. This movement occurs within an internal network, across the internet, or through physical media, such as USB drives or external hard drives. To secure data in motion, it is crucial to use encryption protocols,, secure communication channels, and ensure the integrity and confidentiality of data during its transmission.

Data in use is data currently engaged in processing, manipulation, or interaction by applications or services. This includes data utilized by applications, involved in AI training and inference, or undergoing analytics processing. Security measures for data in use include implementing stringent access controls, monitoring user activities, and safeguarding the integrity and confidentiality of the data during its processing.

It is crucial to understand that data security must be upheld at every level and state. As data transitions between storage, motion, and use, tailored security measures are imperative to protect it throughout its lifecycle. By comprehending the different states of data and applying targeted security controls at each stage, organizations can adopt a comprehensive approach to cloud security, thereby protecting their sensitive data from unauthorized access, alteration, or exposure.

9.1.3 Cloud Storage Types

Understanding the different types of cloud storage helps identify the most suitable storage solution for specific data needs.

Different types of cloud storage include:

- Object storage for large volumes of unstructured data
- Volume storage for low-latency access similar to virtual hard drives
- Database storage for managing relational and non-relational data
- Other specialized storage types used in PaaS and SaaS environments

Each category has unique characteristics, use cases, and products from leading cloud providers.

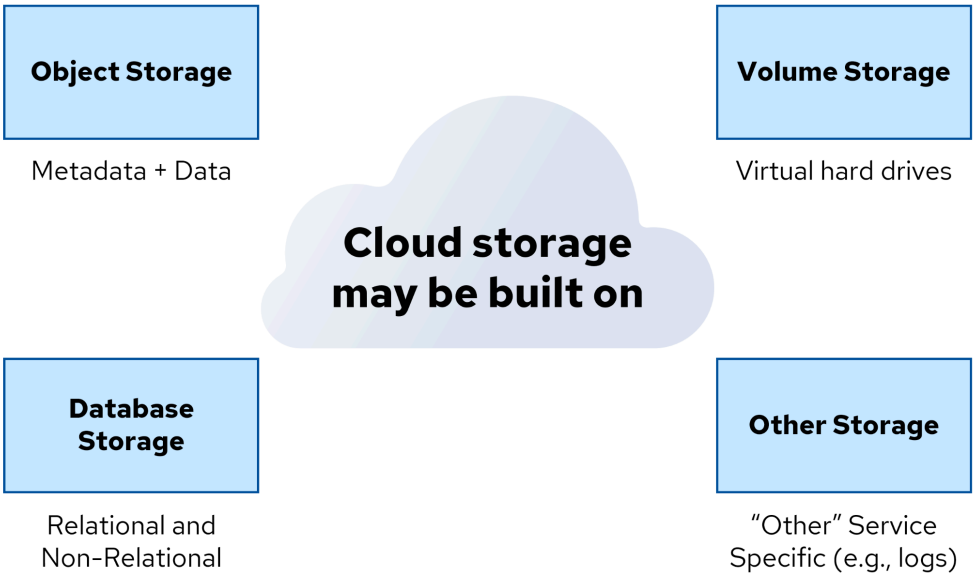


Figure 49: Types of Cloud Storage Solutions

9.1.3.1 Object Storage

Object storage is designed to store and retrieve large amounts of unstructured data, such as documents, images, videos, and backups. It provides a simple application programming interface (API) for storing and accessing objects identified by unique keys. Object storage is highly scalable and durable, making it suitable for various use cases like backup, archiving, and serving static website content. These kinds of storage are classified as infrastructure as a service (IaaS) services because the cloud provider provides the storage infrastructure. Examples of object storage services include Amazon S3, Google Cloud Storage, and Azure Blob Storage.

9.1.3.2 Volume Storage

Volume storage provides virtual hard drives that can be attached to virtual machines in the cloud. It allows you to store and access data in a way similar to traditional hard drives. Volume storage is typically used for operating system files, application data, and other persistent data that require low-latency access. Although their usage differs from Object Storage, this category of storage is also considered an IaaS service. Cloud providers offer different types of volume storage, such as Amazon Elastic Block Store (EBS), Google Persistent Disk, and Azure Managed Disks, with varying performance characteristics and pricing options.

9.1.3.3 Database Storage

This includes both relational and non-relational databases. Cloud providers offer managed services for relational databases like Amazon Relational Database Service, Google Cloud Structured Query Language (SQL), Microsoft Azure SQL Database, and Oracle Database services. These services provide familiar

database engines, such as MySQL, Oracle, PostgreSQL, and SQL Server. Non-relational databases, also known as NoSQL databases, are designed for scalability and flexibility. Examples include Amazon DynamoDB, Google Cloud Datastore, Oracle NoSQL Cloud DB, and Azure Cosmos DB. These databases are highly scalable and can handle large amounts of unstructured data.

9.1.3.4 Other Types of Storage

PaaS storage refers to cloud platforms' various service-specific storage options. These can include logging services like Amazon CloudWatch, Google Cloud Logging, Oracle Events, and Azure Monitor, which store and analyze log data from applications and infrastructure. Message queues enable reliable communication between distributed application components, such as Amazon Simple Queue Service (SQS), Google Cloud Pub/Sub, and Azure Queue Storage. Oracle Cloud Infrastructure (OCI) Streaming and other PaaS storage services may include caches, in-memory databases, and more. Cloud storage may also be offered as Software as a Service (SaaS), such as Google Drive, Dropbox, Microsoft OneDrive, Box, and others. These services enable users to securely access and share files and resources, enabling robust collaboration over the Internet.

9.2 Securing Specific Cloud Workload Types

Each type of cloud workload has unique security needs and challenges. Different tools and techniques are required to address these variations effectively. This section explores the essential data security tools that form the core of protecting data storage in the cloud. These include identity and access management (IAM) systems for governing access, access policies for defining permissions and network rules, and encryption and key management to safeguard data integrity and confidentiality. Additionally, techniques such as masking, tokenization, and anonymization, along with data loss prevention (DLP) and data security posture management (DSPM) tools, play vital roles in ensuring robust data security. By implementing these measures, organizations can effectively manage and mitigate risks associated with cloud data storage and processing.

9.2.1 Data Security Tools & Techniques

Although technically, all information security is data security, these tools form the core toolkit for focusing on the security of the data storage itself. Additional details on each of these tools will be provided in the remainder of the domain.

- **Identity and Access Management (IAM[141])**: IAM systems govern entities' access to particular resources in the cloud environment when making API calls or working within the service where the user and the data exist in the platform. This is different from access controls, which can also govern external access. In IaaS and PaaS, for example, access can be managed in a user-based IAM policy or in a resource policy attached to the storage.
- **Access Policies**: Access policies govern resource access. They define the access and allowed actions (i.e., permissions) for specific resources and determine the network rules that regulate the traffic flow between resources. Both resource and network policies help enforce security

[141] IAM is covered in detail in *Domain 5: Identity and Access Management*.

boundaries.

- **Encryption and Key Management:** Encryption safeguards data by transforming it into unreadable ciphertext, which can only be deciphered by those possessing the appropriate decryption keys. Key management systems securely store and manage these encryption keys, ensuring they are kept separate from the cloud service provider (CSP), either within their cloud infrastructure or on an external key management server. This combined approach ensures data confidentiality and integrity within the cloud environment.
- **Masking:** Technique that replaces sensitive data with fictitious or partially obscured values, preserving format and length. For example, showing only the last four digits of a credit card number or creating false personally identifiable information (PII) data for test environments
- **Tokenization**: Process of replacing sensitive data with unique identifiers (tokens) while maintaining referential integrity and security. It requires another database that stores the original data and associated token to convert the token back to the original data value.
- **Anonymization**: Process of removing PII from data sets, rendering individuals unidentifiable. Anonymization techniques are typically irreversible, making it impossible to recover the original data.
- **Data Loss Prevention (DLP)**: DLP refers to systems that enforce policies to safeguard critical data, such as Intellectual Property and customer information, and ensure it doesn't escape from the enterprise to unintended parties. DLP solutions help identify, monitor, and protect sensitive data, including data stored in cloud environments. These solutions are capable of discovering and classifying sensitive information, enforcing security policies, and preventing unauthorized data sharing or exfiltration.
- **Data Security Posture Management (DSPM)**: DSPM tools continuously assess, monitor, and remediate the security posture of cloud data. They provide visibility into security events, misconfigurations, and compliance issues, enabling organizations to identify and address security gaps proactively and support risk management.

9.2.2 Access Controls & Policies

In cloud computing, managing access is foundational for ensuring security and operational integrity. Access controls and policies, implemented through frameworks like IAM and role-based access control (RBAC), define and enforce user permissions across diverse cloud services. These mechanisms handle various access methods, including API and non-API interactions, and ensure that permissions are consistently applied, even when resource-level policies might override them. Access policies further support these controls by setting clear rules for allowed actions on resources, and governing network interactions within the cloud. This section will explore these mechanisms, providing practical examples to illustrate their roles in maintaining cloud security[142].

[142] Details for risk assessment is provided in *Domain 5: Identity and Access Management*.

9.2.2.1 Access Controls

Access controls are vital components of cloud security, specifically applied to identified users, usually through mechanisms like IAM or RBAC policies. These controls are necessary to govern the variety of access methods, such as ensuring management of API calls, non-API interactions, and other access approaches that can differ significantly among CSPs. It's important to note that resource-based policies might override user or IAM policy denials if the resource is accessed through a channel not considered by the IAM policy. For instance, a user might still gain access to a resource through a web interface or an application that doesn't involve an API call.

Consider this example: An Amazon Web Services (AWS) IAM policy has been crafted to allow the 'AppRead' role to interact with the 'ApplicationData' S3 bucket. This policy uses JSON format and details various aspects, such as the policy version, statements, the effect (in this case, 'allow'), the principal (which would be the 'AppRead' role), actions (specifically, 's3:GetObject' and 's3:ListBucket' permissions), and the designated resource (identified as 'ApplicationData/*' and 'ApplicationData/' within the S3 bucket).

This scenario highlights the strategic use of an IAM policy to assign precise permissions to a role, facilitating access to a designated S3 bucket and its contents. Meticulous planning and execution of access controls are imperative to maintain the integrity and security of resources in the cloud.

9.2.2.2 Access Policies

Access policies govern resource access. They define the access and allowed actions (i.e., permissions) for specific resources and determine the network rules that regulate the traffic flow between resources. Both resource and network policies help enforce security boundaries.

Resource policies serve as the rulesets directly attached to specific resources, such as object storage, enabling them to be accessed independently of API calls, for instance, through HTTP. These policies are integral in scenarios where CSPs facilitate access to the resources for entities that are not part of your IAM user group. Additionally, they often encapsulate network rules, which can impose restrictions based on IP addresses.

It's important to understand that these resource policies can override access controls or identity-based policies established in previous configurations. For example, if an identity-based policy previously prohibited a role from accessing a resource, the resource policy can grant access because cloud providers typically prioritize the evaluation of the resource policy over the identity policy when it comes to direct access scenarios. This mechanism is crucial for maintaining access flexibility while also enforcing security parameters on cloud platforms.

Network policies are a set of rules that govern the data flow over a network and can also be applied directly to resources within that network, such as Microsoft Azure. They are primarily used to manage access through IP addresses or designated IP ranges, establishing a perimeter of communication that determines who can or cannot interact with a network resource.

As an example, consider a resource policy applied to an S3 bucket, which is designed to authorize a role

from an external AWS account, granting it permission to access from certain approved IP addresses. This policy would be articulated in JSON format, specifying elements, such as the policy version, detailed statements, and the effect, which would be to 'allow.' The principal here would be the combination of the AWS account ID and the specific role. It would define actions like 's3:GetObject' and 's3:ListBucket', applicable to resources identified as 'ApplicationData/*' and 'ApplicationData/' within the S3 bucket. A critical part of this policy is the condition that designates the IP addresses from which access is permitted.

This configuration exemplifies the flexibility of resource policies in cloud environments, enabling roles from varied accounts to be granted access while simultaneously implementing IP-based restrictions to bolster security. Both resource and network policies should be meticulously crafted and executed to secure cloud resources, ensuring that they provide the exact level of access control and network security required by the organization.

9.2.3 Cloud Data Encryption

The image illustrates the different layers where data can be encrypted in the cloud, starting from the lowest layer (volume or object storage) and moving up to the application layer. As you move up the encryption layers, you gain more granular control and protection for your data, but it also involves more complexity in implementation and management. The appropriate encryption layer(s) should be chosen based on the sensitivity of the data, compliance requirements, performance needs, and the level of control and management required.

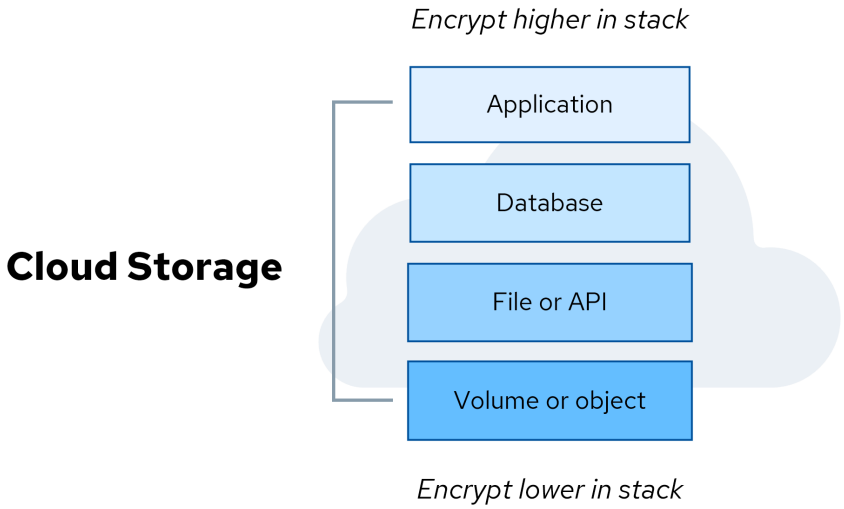


Figure 50: Cloud Data Encryption Layers

Application
Encrypting the data at the application layer provides business data protection and control. Applications can encrypt specific sensitive data elements, thereby ensuring protection at a more precise level. However, application-layer encryption demands more effort to implement and manage compared to encryption at lower layers.

Database

Encrypting data at the database layer protects all information within databases and their backups. The encryption method provides detailed control over which data is encrypted, and helps meet regulatory compliance mandates.

File/API
Encryption at the file or API level offers more granular protection than volume or object storage encryption. This layer allows for encrypting specific files or data accessed through APIs, providing more targeted security.

Volume or Object Storage
At this lowest layer, data is encrypted at rest. Encryption at this level is easier to manage and performs better than higher layers. However, the data is less granularly protected, as encryption is applied to the entire volume or object repository. Volume encryption is a form of Total Disk Encryption, and object repositories (e.g., buckets or storage accounts) can be set to encrypt all the objects in that repository. CSPs typically default encrypt all their storage, but customers may be able to choose and manage their own keys. We discuss this in the *Bring Your Own Key Encryption* section later in this document.

9.2.3.1 Cloud Data Encryption Strategies

All encryption systems are composed of three primary components:

- The data to be encrypted
- The encryption engine (the component that encrypts/decrypts)
- The encryption key

These functional components can be in different locations and this is a core principle in cloud encryption. For example, the data can be located in the cloud service but encrypted by the customer before storing in the service, using a key stored in yet a third location (a version of client-side encryption). Or the key can be held by the customer, provided at runtime to the provider, with the data being encrypted by the CSP (a form of server-side encryption). Finally, the key can be managed by a service in the provider but controlled by the customer and used in server-side encryption (Bring Your Own Key).

9.2.3.2 Client-Side Encryption

Client-side encryption is a security measure employed when a cloud provider stores only encrypted data. In this model, the customer is responsible for encrypting their data before it is sent to the cloud, which guarantees that the cloud provider is unable to access the data in its unencrypted state. Consider an organization that decides to encrypt its sensitive files using an encryption tool on its own premises. After encryption, they upload these files to a cloud storage service, such as Amazon S3 or Google Cloud Storage. Since the nature of encrypted files makes them unusable for active processing, this approach is typically adopted for the purposes of backing up data, archiving, or storing it in a rarely accessed, or 'cold,' state.

9.2.3.3 Server-Side Encryption

Server-side encryption is a service offered by most cloud providers where data is encrypted using keys managed by the provider itself. It is designed for ease of setup and typically doesn't require any specific configuration by the customer, which can make it an appealing choice for organizations that may have less rigorous security needs. The security of server-side encryption depends on the cloud provider's own encryption protocols and their management of the encryption keys. The primary protection offered by this type of encryption is against attacks that involve physical access to the storage hardware, such as attempts to access the hard drive directly. An example of server-side encryption in action is AWS S3's default encryption feature, where Amazon takes charge of the encryption keys and automatically encrypts all data when it is at rest.

9.2.3.4 Customer-Managed Encryption Keys

Customer-Managed Encryption Keys allow customers to take an active role in managing their encryption keys through the cloud provider's key management service (KMS). Despite managing the keys, the actual encryption process is executed by the cloud provider. This method gives customers authority over the encryption key lifecycle, encompassing their creation, rotation, and deletion. Meanwhile, the cloud provider's infrastructure is responsible for performing the encryption of data. When employed correctly, Customer-Managed Encryption Keyscreates a clear separation of responsibilities: the customer maintains control over the keys, while the CSP takes care of the encryption. Generally, this strikes a balance between maintaining control over the keys and convenience of use, making it a favorable option for many organizations. A practical application of Customer-Managed Encryption Keys is evident when organizations use services like Azure Key Vault for key management, with these keys then used to encrypt data stored in Azure's storage solutions, such as Azure Blob Storage or Azure File Storage.

9.2.3.5 Customer-Provided Encryption Keys

Customer-Provided Encryption Keys, also referred to as Bring Your Own Key (BYOK) is a model where the encryption of data takes place after it has been uploaded to the cloud, with the cloud provider carrying out the encryption process. In this system, customers are typically required to supply their own encryption keys to the cloud provider's KMS. This process might necessitate the transfer of the encryption key into the CSP's KMS, which could introduce additional security risks. This is a form of BYOK where the key is managed by the customer on their side and only provided at the time of use, as opposed to using a KMS where the keys are stored and managed at the cloud provider but under customer control.

While this approach affords customers greater control over the encryption keys that the CSP uses, it also depends on the CSP's encryption services. Opting for this solution can potentially restrict the range of services or features the CSP can offer because of the limitations imposed by managing external keys. A common scenario that illustrates this encryption model is when an organization brings its own key to the Google Cloud Platform's Cloud KMS for the purpose of encrypting data stored in Google Cloud Storage.

9.2.3.6 Custom Application-Level Encryption

This method encompasses hybrid situations and application-level encryption where the customer bears the full responsibility for both the encryption and the management of encryption keys. It necessitates that customers completely separate the encryption keys from the CSP, opting instead to handle the encryption process through third-party solutions or by utilizing industry-standard encryption libraries that have robust key management features.

Though this strategy offers the highest degree of control over encryption and keys, it correspondingly increases the complexity and the management burden for the customer. An example of this approach in action is the implementation of client-side encryption within a custom application using tools, such as the AWS Encryption Software Development Kit (SDK). In such a case, the application is designed to encrypt data before it is transmitted to the cloud, and all encryption keys are managed on the client side, not in the cloud.

When choosing a cloud data encryption strategy, organizations should consider factors, such as the sensitivity of the data, regulatory requirements (e.g., PCI-DSS, HIPAA, GDPR), the desired level of control, and the balance between security, ease of use, and availability. The appropriate strategy will depend on each organization's specific needs and constraints. It's essential to carefully assess the risks and benefits of each approach and select the one that best aligns with the organization's security goals and resources.

9.2.3.7 Confidential Computing

Confidential computing is an approach that focuses on ensuring that sensitive data remains encrypted and secure even while it is being processed or analyzed (data-in-use). By using hardware-based enclaves, the entire workload runtime and memory are encrypted, enabling very tight security at all layers of the processing stack.

9.2.4 Key Management Service & Bring Your Own Key

Today, most of the services for Customer-Provided Encryption Keys/BYOK solutions focus on using CSP services. A customer may or may not be bringing a key from an external source. This figure outlines how these encryption systems work, although each CSP will be different at a technical level.

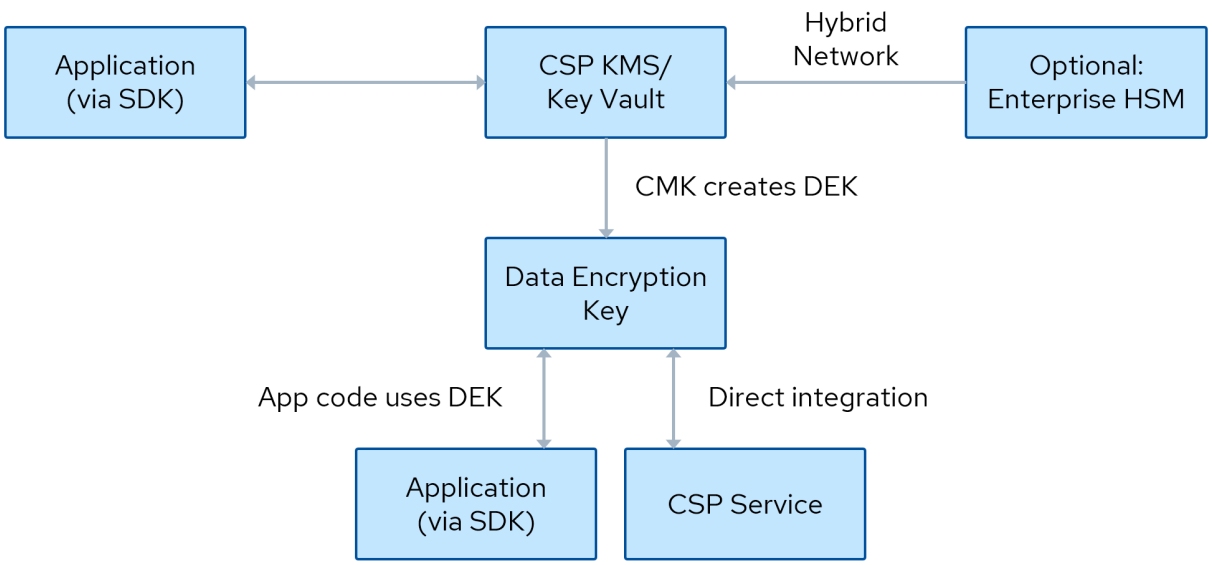


Figure 51: Key Management Service and Bring Your Own Key Encryption

CSPs offer key management services, such as AWS KMS, Google Cloud KMS, Azure Key Vault, and OCI KMS. These services perform various functions, including key creation, deletion, storage, policies, and rotation. They are generally implemented as a multi-tenancy system for all CSP customers, drawing similarities to traditional hardware security modules (HSMs). Additionally, some providers offer a dedicated cloud-based HSM that provides a non-multi-tenancy option.

Customers typically manage encryption through the creation of Customer-Managed Encryption Keys (CMEK) using the CSP's KMS. In certain situations, keys may be created within an on-premise HSM, or the HSM might be used to generate initial keying material. This is common for customers who already utilize HSMs in their data centers and require synchronization of keys across hybrid or multi-cloud applications.

The CMEK is leveraged to generate individual data encryption keys (DEKs). It is a standard practice that the CMEK remains within the KMS, with the DEK being employed for actual encryption tasks. Services like AWS S3 can utilize a KMS key to encrypt objects in a customer's bucket, and Azure Key Vault has the capability to encrypt managed storage volumes.

For application-level encryption, customers often use the DEK, typically through a CSP-provided SDK that is specifically designed for carrying out encryption operations. This allows customers to maintain the security of their applications while utilizing the robust encryption frameworks provided by the CSP.

Some KMS systems offer an optional feature where customers can send plaintext data using an API to the service. The service then encrypts the data, and the ciphertext is returned. This process helps prevent key exposure at the application level and can be integrated as an added security measure when customers opt for Enterprise HSM solutions that support this functionality.

9.2.5 Data Encryption Recommendations

Having explored the fundamentals of data encryption in cloud environments, the following are recommended strategies for enhancing data encryption, each aimed at improving security, compliance, and overall data protection for your cloud-based operations:

- **Key Management Services (KMS):** To secure cloud applications and services, it is recommended that you utilize a KMS provided by your cloud provider. These services help manage cryptographic keys for the organization.
- **SaaS Considerations:** If you're using SaaS, the KMS may be your only encryption option. SaaS often doesn't allow much customization, so you'll rely on the provider's tools for data protection.
- **Default Encryption:** This typically means that your data-at-rest is encrypted using the cloud provider's keys. It's generally included in the service at no additional cost and can address compliance requirements related to data protection.
- **Different Keys for Services:** Using different encryption keys for different services or deployments is a good practice. This approach enhances security by isolating the encryption domains and limiting a compromised key's potential impact.
- **IAM Policies on Keys:** Apply IAM policies to your keys to enforce the principle of least privilege. Doing so ensures that only authorized users and services can use a particular key, and you can define what actions they can perform with it.
- **Alignment with Threat Models:** Make sure your encryption strategies align with your threat models. For example, encrypting your database is less effective if an attacker can compromise application credentials or those of a Database Administrator. In such cases, the attacker can access or exfiltrate encrypted data through legitimate channels.

9.2.6 Cloud Data Loss Prevention

DLP tools are used to discover sensitive data, monitor use, and prevent or alert on policy violations. DLP in the cloud presents unique challenges due to the massive scale of data. The sheer volume and associated costs make comprehensive DLP scanning difficult, especially for IaaS or PaaS environments. As a result, DLP tends to be used more extensively for SaaS applications than IaaS or PaaS.

When DLP is implemented for IaaS or PaaS, the cloud provider's native DLP services are often limited in scope. For example, AWS Macie focuses primarily on S3 storage. External DLP tools integrating with IaaS or PaaS frequently must utilize data sampling techniques to manage the volume and mitigate costs.

Organizations should view IaaS or PaaS DLP as analogous to "DLP for the data center" - something they likely have not implemented historically due to complexity and scale. In contrast, DLP for SaaS is more aligned with traditional DLP practices focusing on monitoring user activity across email, web browsing, and cloud application usage.

To build an effective cloud DLP strategy, organizations must carefully assess their data landscape, prioritize high-risk environments, and balance the usage of cloud-native and third-party DLP solutions. A risk-based approach, combined with strong access controls, can help manage the challenges of cloud DLP at scale while still protecting sensitive information across the enterprise cloud footprint.

Cloud DLP a primary capability of Cloud Access Security Broker (CASB) which provides visibility and security control by monitoring user operations in the cloud and adopting the defined policy according to procedure. CASB consolidates multiple types of security policy enforcement. Example security policies include encryption, tokenization, and so on.

9.2.7 Data Security Posture Management

DSPM is an emerging category of tools designed to focus on data-centered security. Where cloud security posture management (CSPM) manages your IaaS cloud configuration and posture, and SaaS security posture management (SSPM) manages your SaaS security, DSPM provides visualization and management capabilities for the data.

This includes data discovery and classification, which may also include DLP-like capabilities, to help you understand where you have data and its sensitivity. DSPM tools can then pull and evaluate all the overlapping access controls, IAM policies, resources, and network policies to assess and visualize WHO has access to the data and how. These tools then offer suggestions and/or directly manage remediation or provide specific recommendations, such as infrastructure as code (IaC) templates or policies.

The challenge in cloud data security is handling all the potential overlapping controls, all managed in different areas, that don't necessarily provide a complete view of your data use and exposure. DSPM is designed to fill that gap.

9.3 Securing Specific Storage Types

Object storage services, such as AWS S3 and Azure Blob Storage have been an important component of CSP offerings. Cybersecurity experts consider them to be an area of significant organizational data exposure risk. High-profile incidents highlight these vulnerabilities, often stemming from misconfigurations. Typically, cloud providers set storage objects to private by default, but users may inadvertently change these settings, exposing sensitive business and private data. The complexity of access settings further complicates security.

To mitigate risks, cloud providers offer features to block public access at the deployment level, although this can sometimes interfere with legitimate data access needs. Encrypting data and using content delivery networks (CDNs), as will be discussed below, adds additional layers of security. Continuous monitoring and a proactive security strategy is an additional layer of protection to avoid breaches and minimize the blast radius.

9.3.1 Object Storage Security

Object storage services, such as AWS S3 and Azure Blob Storage, are essential for cloud operations, yet they pose significant data exposure risks. Notable incidents[143], underscore these vulnerabilities. Typically, cloud providers configure storage objects to private settings by default to protect data. Nevertheless, breaches frequently happen when users inadvertently alter these settings, thus making sensitive data publicly available.

Misconfigurations and misunderstandings of complex access settings further contribute to data exposure. AWS's nuanced permission system, involving IAM roles and policies along with resource-based policies, exemplifies the challenge of securing cloud storage without a thorough understanding of these elements. To combat unintended public data exposure, cloud providers have introduced the ability to block public access at the deployment level. For instance, AWS allows users to enforce account-wide settings that prevent data leaks due to misconfigured bucket permissions.

However, blocking public access universally can hinder applications that legitimately require open data access. Managing exceptions to these blocks requires careful configuration to ensure that only necessary data is accessible publicly. Adding another layer of security, encrypting data using services like the cloud provider's KMS protects the data, provided that the encryption keys are managed separately from the identities that have the capability to alter object storage permissions. This ensures that even if a storage container is inadvertently made public, the encryption keys remain secure.

Some applications might utilize a CDN to distribute data stored in private object storage, like a private S3 bucket linked to a public-facing CDN. Although this setup does not directly expose the storage, it results in public data access through the CDN. Continuous monitoring using tools, such as CSPM and Data Security Posture Management (DSPM) is crucial. These tools help detect and correct deviations from security best practices by continuously monitoring the security posture.

The concept of 'shift-left' in cybersecurity plays a vital role in enhancing the security of cloud infrastructure. It involves early detection and prevention of misconfigurations in IaC, ensuring that best practices are integrated early in the development process. This proactive approach helps avoid potential security issues before they become significant problems in the production environment. Thus, continuous monitoring and a proactive security strategy are imperative to maintain the integrity and confidentiality of data in cloud-based object storage.

9.3.2 Cloud Database Security

When deploying databases in the cloud, organizations primarily choose between two approaches:

- Traditional database as a service (DBaaS)
- Cloud-Native Databases

[143] CSA. (2024) *Research Topic - Top Threats*. CSA sponsors a working group and discussion community to track and address threats like the Capital One breach in its Top Threats report.

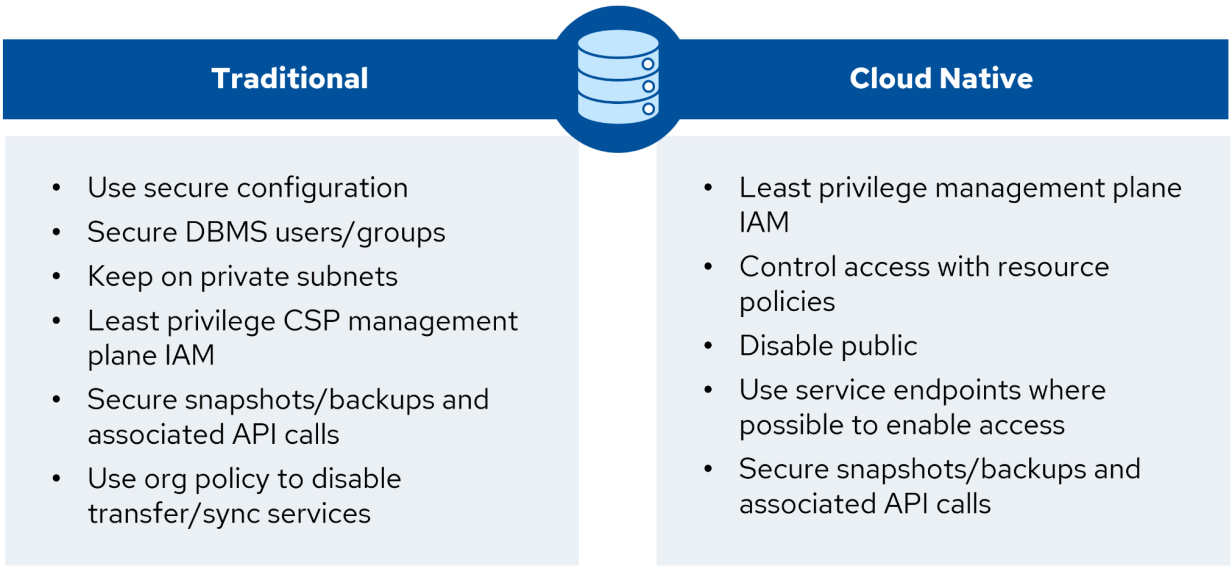


Figure 52: Traditional DBaaS vs. Cloud-Native Databases

Traditional Database as a Service (DBaaS)
Organizations can opt for well-known database engines, such as MySQL, PostgreSQL, or Microsoft SQL Server, which are provided as managed services by cloud providers. To safeguard these databases, it is advisable to adhere to established best practices. This includes employing a secure configuration that incorporates strong authentication measures and comprehensive access controls. It is also important to create secure database user accounts and roles, store data within private subnets, and implement least-privilege access principles to the CSP's management plane IAM roles. Furthermore, leveraging the CSP's managed backup and snapshot capabilities can enhance data protection but must be securely maintained to prevent unauthorized access, a common method for data exfiltration. If services like cross-region replication are unnecessary, they should be deactivated to align with the organization's policy and compliance demands. Well-known examples of traditional DBaaS include Amazon RDS, Azure SQL Database, and Google Cloud SQL.

Cloud-Native Databases
Alternatively, organizations might consider cloud-native databases that are specifically designed and optimized for cloud platforms, featuring capabilities such as serverless operations and automatic scaling. To protect these databases, it is essential to start with assigning least-privilege permissions to the management plane IAM roles. Access to the data plane should be regulated through resource-based policies and public access should be disabled to prevent unauthorized data exposure. Where feasible, using dedicated private endpoints or virtual private cloud (VPC) integration can provide secure pathways for applications to connect to these databases. It is also beneficial to utilize the automatic backup and snapshot features of the service, along with making secure API calls for application integration. Examples of cloud-native databases include Amazon DynamoDB, Azure Cosmos DB, and Google Cloud Firestore.

For both types of cloud databases, it is vital to:

- Adhere to the shared responsibility model and reinforce your database configurations.
- Encrypt data, as necessary, to meet compliance requirements.
- Implement strong authentication measures and uphold least-privilege principles.

- Monitor logs actively and utilize cloud-native threat detection services. It's advisable to consider activating data-level logs/events, which may not be enabled by default.
- Develop and maintain a comprehensive backup and recovery plan.

9.3.3 Data Lake Security

In the era of big data, the concept of Data Lakes has emerged as a pivotal element for managing and analyzing vast amounts of data from a multitude of sources. As defined in CSA's Data Security Glossary[144] "A Data Lake is a centralized repository that ingests and stores large volumes of data in its original form. The data can then be processed and used as a basis for a variety of analytic needs. Due to its open, scalable architecture, a data lake can accommodate all types of data from any source, from structured (database tables, Excel sheets) to semi-structured (XML files, webpages) to unstructured (images, audio files, tweets), all without sacrificing fidelity." This definition encapsulates the essence of data lakes as comprehensive data consolidation points equipped to handle and preserve the complexity of diverse data forms.

However, the integration of such extensive data sets from varied sources presents substantial security challenges as data lakes become prime targets for cyber threats. The security strategies deployed within data lakes must, therefore, be multifaceted to protect sensitive information while maintaining accessibility and utility. Segregation and compartmentalization of data based on sensitivity and confidentiality, along with implementing robust access control systems, are critical for maintaining data integrity and security. Through measures such as encryption, network security, and continuous monitoring, organizations can strive to fortify their data lakes against potential vulnerabilities, ensuring the safe and effective use of their consolidated data resources.

To ensure the security and integrity of the vast and varied data stored within, the following strategies have been developed.

- **Data Lakes as Data Consolidation Points:** Data lakes are powerful because they consolidate a wide variety of data from numerous sources, which can vary widely in sensitivity and security classification. The challenge is to maintain strong security across this diverse data set.
- **Security Levels and Data Segregation:** Given the variety of data within a data lake, not all users or applications should have access to all data. Some data might be public, while other data could be highly confidential. It's critical to segregate this data effectively.
- **Compartmentalization through Views/Access Points:** Create views or access points that act as windows into the data lake, each tailored to show only the data that is relevant and permissible for a particular user or application to access. This is akin to creating virtual partitions within the data lake.

9.3.3.1 Baseline Data Security Practices

The following are measures for fortifying the security of your data environment. These fundamental

[144] CSA. (2024) Download Publication - *CSA Data Security Glossary*.

measures establish the groundwork for defending against vulnerabilities and threats, setting the stage for implementing more specialized or advanced security strategies as needed.

- **Continuous Vulnerability Assessment and Remediation Management:** To protect against vulnerabilities, ensure all components interacting with the data lake are up to date with the latest security patches.
- **Identity and Access Management (IAM):** Implement detailed IAM policies. Access should be based on the principle of least privilege, ensuring users and applications only have the permissions necessary to perform their functions.
- **Encryption:** Apply encryption at rest, in transit, and in use to protect data from exposure if other controls fail. For example, use AWS KMS to manage encryption keys for S3 buckets in the data lake.
- **Network Security:** Utilize network security measures, such as VPCs, security groups, and network access control lists, to control traffic flow into, and out of, the data lake.
- **Ongoing Log Management, Monitoring, and Alerting:** Continuously monitor access patterns and review permissions to ensure the implemented security measures remain effective over time. Remember, securing a data lake is not a one-time task but an ongoing process that involves regular review and updates as new data is added and access requirements change.
- **Penetration Testing:** Test the effectiveness and resiliency of information assets through identifying and exploiting weaknesses in controls (people, processes, and technology), and simulating the objectives and actions of an attacker.

9.3.4 Data Security for Artificial Intelligence

As AI technologies become more prevalent and integrated into critical business processes, ensuring the security and integrity of AI systems is paramount. Data security for AI involves implementing measures to protect AI systems, algorithms, and data assets from various security threats and vulnerabilities There are two main approaches to deploying AI: AI as a service (AIaaS) and self/cloud-hosted AI.

9.3.4.1 AI as a Service

In the AIaaS model, third-party providers offer AI capabilities and services over the Internet on a subscription basis. Organizations can access pre-trained AI models, APIs, and tools to integrate AI functionality into their applications and workflows. Examples include Anthropic's Claude, OpenAI's ChatGPT, and Google Cloud's Vertex AI. When using AIaaS, it's essential to:

- Understand the Service-Level Agreement (SLA) to ensure the provider meets your availability, performance, and support requirements.
- Conduct a thorough assessment of the provider’s data security practices, including encryption methods, access controls, and monitoring capabilities to safeguard data from unauthorized access.

- Verify the provider's compliance with relevant regulations and standards, such as GDPR, HIPAA, or SOC 2, to ensure the protection of sensitive data and adherence to industry best practices.
- Avoid sending proprietary or sensitive data if the provider doesn't offer secure enclaves or dedicated environments.
- Clarify data deletion and retention policies with the provider to ensure that data lifecycle management meets security requirements.
- Evaluate the provider's AI security measures, including safeguards against adversarial attacks, such as model poisoning, prompt injection, and so on, to ensure the integrity and reliability of AI-driven services and applications.

9.3.4.2 Self/Cloud-Hosted AI

In the self/cloud-hosted AI approach, organizations develop, deploy, and manage their AI models and infrastructure either on-premises or in the cloud. Organizations have full control over the AI development process, including data collection, model training, optimization, and deployment. The organization is fully responsible for securing the AI system. Key considerations include:

- Securing the training data repository by implementing access controls, encryption mechanisms, and data governance policies to protect sensitive data from unauthorized access or tampering.
- Establishing secure AI system access, including network segmentation, firewall configurations, and granular IAM policies to control and restrict system access to authorized users and entities.
- Filtering training data to prevent model poisoning, which aims to manipulate the AI's behavior
- Implementing safeguards against prompt injection attacks, where malicious inputs attempt to bypass the AI's intended behavior by exploiting vulnerabilities in the input mechanism.
- Protecting against AI jailbreak attempts by deploying robust prompt scanning mechanisms that detect and block data within the AI system.
- Regularly updating and patching the AI system to address emerging security vulnerabilities.
- Ensuring the AI system complies with relevant AI ethics guidelines and regulations to prevent biased or discriminatory outputs.

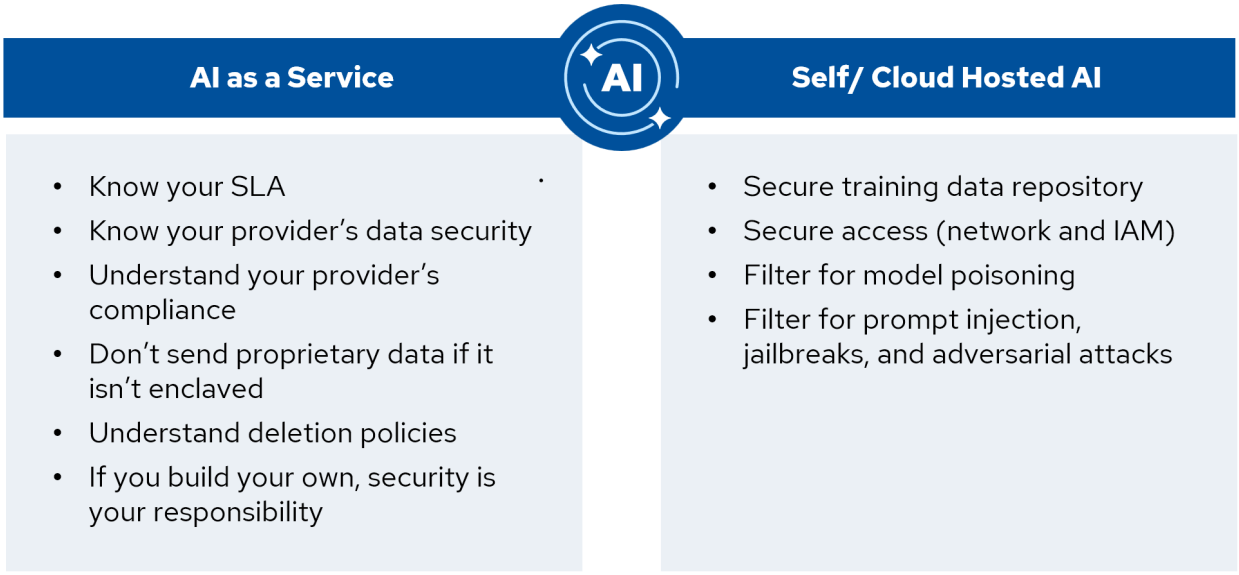


Figure 53: AI as a service vs. Self/Cloud-Hosted AI

9.3.4.3 Additional AI Considerations

To enhance the security of AI systems and ensure their safe integration and operation, consider the following security practices.

- Employ secure APIs and encryption protocols when integrating the AI system with other applications.
- Implement strong authentication and access controls for users interacting with the AI system.
- Conduct regular audits and security assessments of the AI systems, including penetration testing and threat modeling.
- Develop and maintain an incident response plan to detect, investigate, and remediate AI security incidents or breaches.
- Foster a culture of AI security awareness and train developers, administrators, and users.

Summary

This domain addresses the need for robust data security in cloud environments amid the rapid evolution of cloud services and increasing cyber threats. It emphasizes the importance of data security for maintaining organizational integrity, confidentiality, customer trust, and regulatory compliance. The domain explores various aspects of data security, including data classification, cloud storage types, and specific security measures for different data states—data at rest, in motion, and in use. It covers essential security tools and techniques such as IAM, encryption, and access control policies, providing a comprehensive guide for securing cloud data. Overall, the domain serves as a foundational guide for organizations looking to fortify their data security practices in the cloud.

Recommendations

- Cloud security starts with understanding and managing the data you transfer to the cloud and applying access controls.
- Know the differences between IAM-based access controls, resource policies, and network policies. All must be aligned to prevent data exposure.
- Encryption only improves security if it is aligned with the threat model and the keys and the data (and access to each) are segregated. Encryption is worthless if the attacker executes an SQL injection attack and can get the date through the application.
- When implementing encryption in IaaS or PaaS, start with the CSP KMS service.
- Cloud DLP in IaaS mirrors DLP for your traditional data center. DLP for SaaS is like DLP for your email service and applications.
- Ensure the security of data lakes by controlling access to specific data (and following core data storage security controls).
- AI security should focus on potential data exposures, which vary depending on whether you are scrutinizing a public service or a self-hosted model.

Additional Guidance

- Key Management in Cloud Services | CSA
- Cloud Key Management System with External Origin Key | CSA
- Recommendations for Using a Customer Controlled Key Store | CSA
- Cloud Key Management Foundations | CSA
- Cloud Key Management Foundations II | CSA
- Key Management Lifecycle Best Practices | CSA
- An Agile Data Doctrine for a Secure Data Lake | CSA
- Understanding Cloud Data Security and Priorities | CSA

Domain 10: Application Security

Introduction

This domain covers application security, which is the practice of using security controls to protect computer applications from external threats. Application security encompasses an incredibly complex and large body of knowledge: everything from early design and threat modeling to maintaining and defending production applications. Application security is also evolving at a rapid pace as the practice of application development continues to progress and embrace new processes, patterns, and technologies. Cloud computing is one of the biggest drivers of these advancements and brings the urgent and imperative need to ensure that progress is stable, scalable, and secure.

From the initial design phase to ongoing maintenance, the security of cloud-based applications requires careful consideration and proactive measures. An overview of the unique challenges and opportunities presented by application security in the cloud environment (which apply to many private-cloud and on-premises environments, as well) is described below:

- Applications are often built as a constellation of microservices and external services, which necessitates a more detailed analysis of attack surfaces and control boundaries.
- An attack surface often includes significant exposure over APIs.
- In a cloud context, applications are often developed using DevOps approaches with rapid feature development, which can be both a risk, as well as an opportunity.
- Applications can be built on libraries that are under the control of the provider (e.g., PaaS provider, or serverless), which requires attention to the shared responsibility model.
- Applications frequently leverage third-party libraries, including open-source components, introducing supply chain risks and additional attack vectors. This complexity is further compounded when considering the integration of Software as a Service (SaaS) solutions for version management and development repositories, requiring robust security measures to mitigate risks associated with external dependencies.
- Security features, such as identity management, logging, and monitoring, are often sourced from a cloud provider, which may or may not match the application security requirements.
- Applications are often deployed on programmable infrastructure (Infrastructure as a Code (IaC), or orchestrators, such as Kubernetes).
- Applications operating at scale within cloud environments necessitate a keen awareness of the underlying infrastructure's vulnerabilities. Stateless architectures, which prioritize scalability and

resilience, are commonly employed to mitigate the impact of infrastructure failures. However, while these architectures offer flexibility and agility, they also introduce complexities that can undermine the overall security posture.

Learning Objectives

The learning objectives for this domain aim to provide readers with knowledge on:

- Implement a secure development process for creating secure applications.
- Recognize the critical role of architecture in ensuring the security of cloud applications.
- Automate the integration of security throughout the Secure Software Development Lifecycle (SSDLC) using DevSecOps.

10.1 Secure Development Lifecycle

Secure applications start with a secure development process. Simply hoping safe and secure code will be created by teams of developers whose primary job is building functionality is not sufficient. To address these concerns, the development and security professions have merged around a set of processes called the Software Development Lifecycle (SDLC), sometimes also referred to as the Secure Software Development Lifecycle (SSDLC).

Cloud introduces some new implications in various parts of these processes, largely stemming from the tighter integration between application and cloud infrastructure. Developers also tend to use newer, faster methodologies like DevOps when working in the cloud. Finally, IaC and deployment pipelines are standard practice in the cloud, but not necessarily used to the same degree for traditional data center applications.

10.1.1 CSA Secure Development Lifecycle

The Cloud Security Alliance (CSA) Development, Security, Operations (DevSecOps) *Secure Development Lifecycle*[145] (SSDLC) defines five stages that map to the commonly agreed-upon phases of application development[146]. Each of these stages identifies key processes, tools, and design patterns to be implemented in successful DevSecOps programs[147].

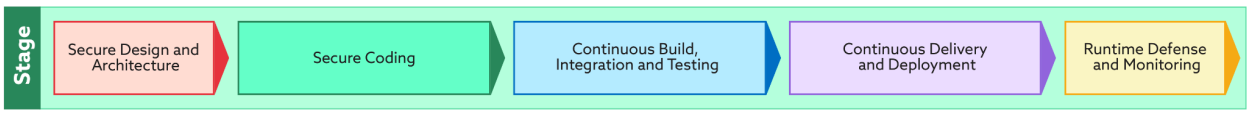


Figure 54: Stages of the Secure Software Development Lifecycle (SSDLC)

[145] CSA. (2024) Secure Development Lifecycle.
[146] Similar development lifecycles have been created by software vendors, such as Microsoft's Security Development Lifecycle (SDL) that offer additional guidance of secure development practices.
[147] CSA. (2022) Specific implementation details are available in Pillar 3 - Pragmatic Implementation of the Six Pillars of DevSecOps Series.

Stages of the SSDLC

1. **Secure Design and Architecture:** The design section references the technologies and tools that can be applied during the design of a product. We recognize that design is continuous, so new product features and changes will route through design activities. Without including security in the design phase, security measures would later be introduced with a higher operating impact and cost at deployment or runtime. Measures would also be difficult to scale, resulting in security bottlenecks slowing the development speed and impacting release timelines.

2. **Secure Coding:** These capabilities are applied during the development phase to ensure security is integrated into the application as it is being built. Coding security controls that rely on automated tools can better and more consistently identify weaknesses and vulnerabilities in code compared to manual human reviews. Without including security analysis and design during the development phase, there is a risk that vulnerabilities in source code will be unidentified and deployed into production environments, which is more expensive to fix, in the later states of the SSDLC.

3. **Continuous Build, Integration, and Testing:** Integration and testing includes the tools and processes to test for security vulnerabilities of an application/product prior to deployment. Without these, security vulnerabilities will be exploited and could result in exploitation such as data breaches, unauthorized access, and various other implications to the availability of the application/service.

4. **Continuous Delivery and Deployment:** Pre-deployment safety checks ensure that an application/product is deployed onto secure infrastructure. Without including security analysis during the deployment phase, there is a risk that vulnerabilities and poor security practices could result in exposing the application, product or service to exploitation and attack in production environments.

5. **Runtime Defense and Monitoring:** These capabilities and practices are applied after an application/product has been released into production. Runtime security enables continuous improvement by identifying inefficiencies, vulnerabilities, and weaknesses and enabling incident response.

10.1.2 Threat Modeling

Threat modeling[148] is a structured process used in risk management to identify, assess, and remediate potential security threats to an organization's assets. It involves understanding the system architecture, identifying security objectives, and analyzing the potential threats that could affect those objectives. By performing threat modeling, you can prioritize the identified threats based on their severity and likelihood and develop strategies to mitigate or prevent associated risks. This process helps create a more secure system design by focusing attention on areas more vulnerable to attacks, ensuring that security measures are integrated early in the development lifecycle.

[148] OWASP. (2024) *Threat Modeling Process*.

STRIDE is a framework used for identifying and categorizing security threats. It stands for **S**poofing, **T**ampering, **R**epudiation, **I**nformation disclosure, **D**enial of Service, and **E**levation of Privilege. Here's a summary of each threat category.

1. **Spoofing**: This involves an attacker pretending to be someone else, such as a user or a system, to gain unauthorized access. Examples include phishing attacks where attackers mimic legitimate sites to steal login credentials.

2. **Tampering**: This refers to unauthorized alterations of data or messages. This could happen in transit or within a storage system, compromising the data's integrity.

3. **Repudiation**: This occurs when a party denies acting despite having done so. This undermines the system's ability to attribute actions to the correct source, complicating accountability.

4. **Information Disclosure**: This involves unauthorized access to confidential information. Techniques include eavesdropping on unsecured communications or exploiting vulnerabilities to access sensitive data.

5. **Denial of Service**: This is the exhaustion of system resources leading to the unavailability of the system. This disrupts operations and can cause significant downtime.

6. **Elevation of Privilege**: When an attacker gains higher access levels than permitted, they bypass access controls to execute actions reserved for more privileged accounts.

Understanding these threats allows for the identification of potential vulnerabilities in a system and guides the development of effective countermeasures to protect against these security risks.[149]

10.1.3 Secure Design & Development

In the cloud, there is generally a greater coupling between the architecture and the application code. Infrastructure and services are commonly deployed using IaC, which is completely integrated with application code, using the same version control repository and continuous deployment pipeline.

Most cloud applications also make heavy use of Platform as a Service (PaaS) services from the cloud service provider (CSP), all of which need to be properly configured for both the needs of the application and security requirements. This also typically requires application code and services to be assigned privileges to make API calls to use those services, expanding the potential attack surface to the management plane.

Secure application design is always important, but this deeper entanglement between the application, the infrastructure, and the management plane requires a greater focus on secure design at the beginning of any project. Cloud architectures are fundamental to application security; a wide range of security risks can be mitigated using the cloud architecture (and services) alone. For example, hosting a single-page application in a static object storage bucket means your application doesn't need a web server.

When designing a cloud application, security principles including the following should be considered:

[149] CSA. (2022) Specific techniques in Pillar 3 - Pragmatic Implementation of the Six Pillars of DevSecOps Series.

- PaaS services push more security responsibility onto the CSP and could reduce or eliminate the need for the customer to maintain a secure, fully patched, and configured server or service.
- Implement least privilege Identity and Access Management (IAM) for all application components and PaaS services.
- Use CSP services, like load balancers and highly-restrictive security groups, to reduce internet-facing exposures.

Once design is complete, standard secure development practices should be followed. The Cloud Security Alliance (CSA) recommends using a DevSecOps process, which is highlighted in section 10.5.

10.1.4 Testing: Pre-Deployment

Pre-deployment testing is a crucial step in ensuring the security and functionality of the software before it goes into production. Teams can save significant time and resources by integrating testing early in the development process, specifically before deployment. This approach allows for the early detection and correction of issues, contributing to a more secure and reliable software product.

The following are examples of key pre-deployment testing methods.

1. **Static Application Security Testing (SAST)**:This process examines an application's source code to identify existing security flaws or vulnerabilities. It is an automated way to perform a security code review and might be integrated into a continuous integration/continuous deployment (CI/CD) pipeline or in the developer's integrated development environment (IDE). It specifically looks for logic errors, examines spec implementation, and checks style guidelines and security flaws (such as those listed by OWASP Top 10 and SANS Top 25). Additionally, it scans the code for hardcoded credentials, keys, tokens, and secrets, preventing them from entering the repository, and identifies potential leaks among other activities. SAST can be prone to false positives, which is why it needs tuning and prioritization to not alienate developers.
    a. **Manual security code review**: In this process, experienced developers examine each code review submitted looking for flaws. An automated process does not catch some important errors like business logic errors, thus a manual code review is highly recommended. This could be enforced using the Pull Request (PR) process. The best approach is to run the automated process with the manual process because they complement each other.
2. **Software Composition Analysis (SCA):** SCA involves auditing the external components your software relies on, such as libraries and system components. This method ensures these components are current and free from known vulnerabilities, and the type of license of these components helps avoid bringing licensing risks into the project. It's vital for creating secure virtual machines (VMs), container images, and serverless functions. SCA can also assist in creating a Software Bill of Materials (SBOM), providing transparency on all components used in the software.

3. **Static Vulnerability Scanning:** Vulnerability scanning is crucial in cloud environments to identify and mitigate potential security threats. There are two main types of scans: static and dynamic. Static scanning analyzes source code (IaC) and configurations at rest, including files like VM images or templates, container images, Dockerfiles, docker-compose files, Kubernetes YAMLs, Terraform or Cloudformation files, and so on. This type of scan is typically performed during the pre-deployment phase of the SSDLC, examining configuration files, infrastructure templates, and source code before deployment. Static scanning helps identify vulnerabilities and configuration errors that can be addressed before they pose issues in production.

10.1.5 Testing: Post-Deployment

Post-deployment testing verifies the security and functionality of software after it's deployed, challenging the assumptions made during its design and integration, particularly for cloud applications. This phase ensures the software operates effectively in real-world conditions, mirroring traditional data center testing processes.

The following are some examples of essential post-deployment tests.

1. **Dynamic Vulnerability Scanning:** Dynamic scanning takes place after deployment in the SSDLC. It involves actively probing the running environment, and emulating real-world attack scenarios to identify vulnerabilities that could be exploited by malicious actors. Unlike static analysis, which examines code and configurations at rest, dynamic scanning assesses the system's security posture in real time, providing insights into potential weaknesses that may have been missed during pre-deployment static analysis. By simulating attacks, dynamic scanning helps organizations understand their systems' resilience to various threats and allows them to remediate vulnerabilities before they can be exploited. When combined with static analysis, which identifies vulnerabilities in the code and configurations before deployment, dynamic scanning provides a comprehensive approach to securing cloud environments, ensuring that applications and infrastructure are protected against potential threats throughout the SSDLC.
2. **Dynamic Application Security Testing (DAST):** DAST is a method of testing in which a tester examines a web application while it is running but has no knowledge of the application, its internal interactions or designs at the system level, and no access or visibility into the source program. DAST is also known as "black box" testing as it looks at an application from the outside in, examines its running state, and observes its responses to simulated attacks using certain techniques and a testing tool. An application's responses to these simulations help determine whether the application is vulnerable and could be susceptible to a real malicious attack.
    - **Dynamic Analysis (Fuzzing):** This involves inputting unpredictable data into the software to identify errors and vulnerabilities that could be exploited during operation.
    - **Interactive Application Security Testing (IAST):** IAST is a method of application security testing that tests the application while the application is run by an automated test, human, or any activity interacting with the application functionality. IAST can be considered a combination of SAST and DAST to achieve its objective of having an overview of issues on source code and execution on runtime.

3. **Penetration Testing:** Conducted as a simulated cyberattack, penetration testing aims to exploit known vulnerabilities, testing the resilience of security measures and the software's ability to withstand attacks. Penetration (pen) tests can be applied using automated tools or manual work. For the long term, it is recommended to apply both. Pen tests can also be applied to test the application components level or performed at the cloud deployment level to identify flaws in the cloud configurations.

4. **Bug Bounty Program:** These programs offer monetary rewards to ethical hackers for successfully discovering and reporting a vulnerability or bug on the live application. Bug Bounty programs allow organizations to leverage the ethical hacker community to improve their systems' security posture over time. While Bug Bounty programs are not always essential, they can be considered by organizations as part of their security strategy.

10.2 Secure Cloud Applications Architecture

The architecture of a system plays a critical role in ensuring the security of cloud applications. It serves as the blueprint for designing and deploying secure cloud-based solutions. By integrating security principles and practices at the architectural level, organizations can create a solid foundation for protecting data, maintaining privacy, and ensuring compliance with regulatory standards. This involves carefully planning the components and interactions within the cloud environment to mitigate potential threats, secure data transmission, and manage access controls effectively. By moving components or not implementing "features," the threat may no longer exist against a specific asset/data flow. A well-designed architecture enhances security and improves the scalability, reliability, and overall performance of cloud applications.

10.2.1 Cloud Impacts on Architecture-Level Security

Cloud computing shifts the paradigm of traditional software and infrastructure development, emphasizing that everything is software. This shift streamlines operations and tightly integrates infrastructure with applications, requiring a new approach to security.

1. **Infrastructure and Application Integration:** The cloud merges infrastructure with applications, making elements like servers and databases integral to the application's functionality. This integration can bolster security through seamless operation. Yet, it also brings about IAM risks, where incorrect permissions could lead to breaches. The key here is meticulous management of identities and access to mitigate potential vulnerabilities.

2. **Application Component Credentials:** In the cloud, components such as microservices communicate using specific permissions and credentials often designated just for that service. Exposure or mismanagement can lead to significant security incidents. Ensuring that credentials are securely handled and access is tightly controlled is critical to prevent breaches.

3. **Infrastructure as Code and Pipelines:** Defining infrastructure through code has become a hallmark of cloud practices, offering consistency and efficiency in deployments. However, these deployment pipelines can attract attackers. A breach in the pipeline could compromise the entire

software supply chain. Protecting these pipelines safeguards the development and deployment processes.

4. **Immutable Infrastructure**: With the maturing of virtualization and Infrastructure as a Service (IaaS) technology, many security experts have shifted to a paradigm in which the management and deployment of a machine configuration is never maintained or tampered with. More specifically, once an instance of an application, a server, or a system configuration is created, it is never modified. Instead, if changes are needed, a new instance is built from a common template and replaced entirely. This approach contrasts with traditional maintenance practices, such as mutable infrastructure.

In summary, the move to cloud computing requires a reevaluation of security at the architectural level, focusing on integrated systems' unique challenges and opportunities. Early and proactive security planning and robust management of identities and credentials form the cornerstone of a secure cloud-based architecture.

10.2.2 Cloud Impacts on Application Design & Architecture

The cloud environment requires transformative changes to how applications are designed and architected, emphasizing flexibility, scalability, and security. Application security should be discussed and planned into a project as early as the requirements-building phase. You may choose to refer to the primary frameworks and guidelines that inform the SSDLC, such as NIST 800-64[150] for a detailed guide on integrating security measures into the development process, or ISO/IEC 27034[151] for guidelines on how to weave security into the lifecycle of an IT system, among others.

DAST tests running applications and includes tests such as web vulnerability testing and fuzzing. Due to the terms of service with the CSP, DAST may be limited or may require pre-testing permission from the provider, which can be time-consuming with certain CSPs. With cloud and automated deployment pipelines, it is possible to stand up entirely functional test environments using IaC and then perform deep assessments before approving production changes.

Below are some of the most common practices in developing, integrating, and deploying for cloud environments:

1. **Segregation by Default:** Cloud platforms allow applications to run in isolated environments, such as separate virtual networks or accounts/sub-accounts. This segregation aids security by compartmentalizing development and production environments, enabling stricter access controls where necessary. Cloud computing facilitates segregating services onto different servers or containers, improving scalability and security. This approach, which usually involves microservices, requires careful management of communication security between microservices and secure configuration of service discovery, scheduling, and routing.

[150] NIST. (2022) *NIST 800-64* originally provided NIST guidance for secure development but has be withdrawn and replaced with *NIST SP 800-64 Revision 2*. We provide both for clarity as you browse the Internet for details.
[151] ISO/IEC. (2018) *ISO/IEC 27034-3:2018(E)*.

2. **Automation of Deployments and Testing:** In Cloud platforms, organizations aspire to develop and deploy software faster than before. This means that for security and operations teams, previously manual work like deploying a test environment or doing security testing on application code should be automated for efficiency and pace. Automation needs lead to adopting new tools and increasing their use in CI/CD pipelines.

3. **Immutable Infrastructure:** By disabling remote logins, adding file integrity monitoring, and incorporating these practices into incident recovery, immutable infrastructure reduces the risk of security breaches.

4. **PaaS and Serverless Architectures:** PaaS and serverless computing reduce the attack surface by offloading the management of underlying services and operating systems to cloud providers. The security of these architectures largely depends on the cloud provider's commitment to securing the platform and meeting the user's security requirements.

Each of these aspects underscores the importance of adapting security strategies to the unique opportunities and challenges presented by cloud computing.

10.2.3 Infrastructure as Code & Application Security

IaC revolutionizes IT infrastructure setup by using configuration files to define and manage resources, akin to auto-constructing a building from its blueprint.

This approach streamlines the deployment and management of cloud resources, significantly enhancing application security.

1. **Automated Compliance Checks:** IaC facilitates automatic validation against security standards and regulations, ensuring compliance whenever infrastructure is provisioned or modified. This automation acts like a relentless inspector, constantly ensuring adherence to security policies.

2. **Consistent Security Posture:** By codifying infrastructure setup, IaC guarantees that every element, from servers to databases, is consistently configured according to security best practices. This eliminates human error associated with manual setups, detects and eliminates configuration drifts, and maintains a uniform security level across all resources. Importantly, IaC also supports the management of exceptions to these security policies through centralized exception management. For instance, specific resources may require deviations from standard configurations due to valid business needs, such as a Simple Storage Service (S3) bucket being configured for public access. By documenting and managing these exceptions within the IaC framework, organizations can ensure that such deviations are recognized, approved, and tracked, maintaining security oversight while supporting necessary operational flexibility.

3. **Rapid Response to Threats:** IaC enables swift modifications to infrastructure code in response to identified vulnerabilities, allowing for patching and hot fix deployment across the entire infrastructure. This capability is akin to remotely updating a building’s security system to address weaknesses without physical intervention.

4. **Rapid CI/CD Rollback:** IaC supports rapid CI/CD rollback capabilities, enhancing operational resilience. When updates to containers or virtualized environments are made, performance can be

statistically compared against benchmarks. If a new automated rollout, such as a canary release, fails to meet these benchmarks, IaC allows for the automatic rollback to previous, stable configurations. This not only minimizes downtime but also ensures that security and operational stability are not compromised by new changes.

The following figure illustrates the iterative process of designing, coding automation, creating infrastructure templates, and deploying in a secure cloud architecture.

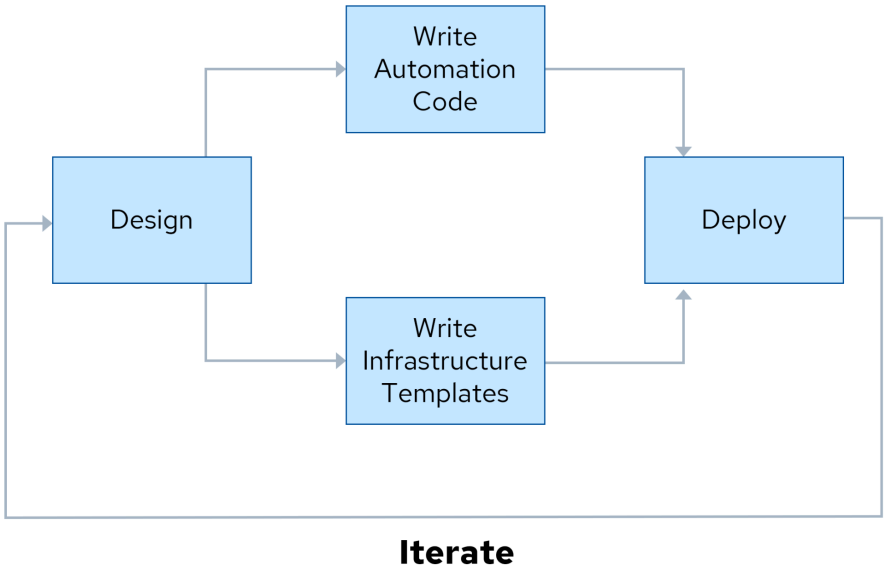


Figure 55: Infrastructure as Code: Enhancing Security Through Automation

Organizations should also rely more on automated testing in the cloud. Infrastructure is more often in scope for application testing due to IaC, where the infrastructure itself is defined and implemented through templates and automation. Leveraging IaC not only enhances operational efficiency but also, by embedding security into the very foundation of the infrastructure, significantly improves the security posture of cloud-based applications.

10.2.4 Best Practices for API Security

When it comes to protecting APIs, there are several options to consider. One option is to use API gateways, which serve as a centralized point for managing authentication, rate limiting, and access control for incoming API requests. This helps ensure that only authorized users and systems can access the API. Another option is to implement a service mesh, which focuses on securing communication between different services within an application. Service mesh provides built-in encryption and authentication mechanisms, which help protect sensitive data as it travels between services.

In addition to these measures, it is important to carefully define the API contract to avoid any potential leaks of sensitive information. The API contract should not be overly permissive, and access should be restricted to only the necessary data and functionality. Furthermore, you should incorporate automated

API security testing into the CI/CD pipeline. By doing so, vulnerabilities can be detected early in the development process, allowing for prompt remediation and reducing the risk of security breaches.

By implementing these protection options and following best practices, organizations can enhance the security of their APIs and safeguard sensitive data from unauthorized access or exposure.

10.3 Identity & Access Management Application Security

IAM plays a pivotal role in enhancing application security. It encompasses the technologies and policies designed to manage identities and regulate user access within organizations. By effectively controlling who has access to what resources and how that access is granted and revoked, IAM systems ensure that the right individuals access the appropriate resources at the right times for the right reasons. Integrating IAM with application security strategies is critical for preventing unauthorized access and protecting sensitive data from potential threats.

10.3.1 Setting Permissions on Application Components

Imagine IAM[152] as the gatekeeper of your digital assets, carefully reviewing each entity's credentials before granting access. This system defines who can enter and outlines their capabilities within the digital realm, akin to a bouncer's role in a club.

1. **Principle of Least Privilege:** Assign access rights akin to distributing keycards that only unlock necessary doors for one's role, minimizing the risk of unauthorized access to sensitive areas.
2. **Continuous Monitoring:** Maintain vigilance through constant surveillance, akin to security teams monitoring CCTV footage, to promptly detect and address any aberrant access patterns.
3. **Segregation of Duties:** Implement multiple security layers, like various checkpoints within a building, to dilute the concentration of access power, thus averting potential misuse or compromise. This also means that developers should use different privileges for different environments (e.g., Dev vs. Prod).
4. **Federation:** Streamline access across diverse systems or organizations by implementing a universal access protocol, much like a universally accepted keycard, to simplify and secure cross-platform interactions.

The following figure depicts the IAM process, illustrating the interaction between the user, identity provider, and service provider to authenticate and grant access.

[152] IAM is covered in detail in *Domain 5: Identity and Access Management*.

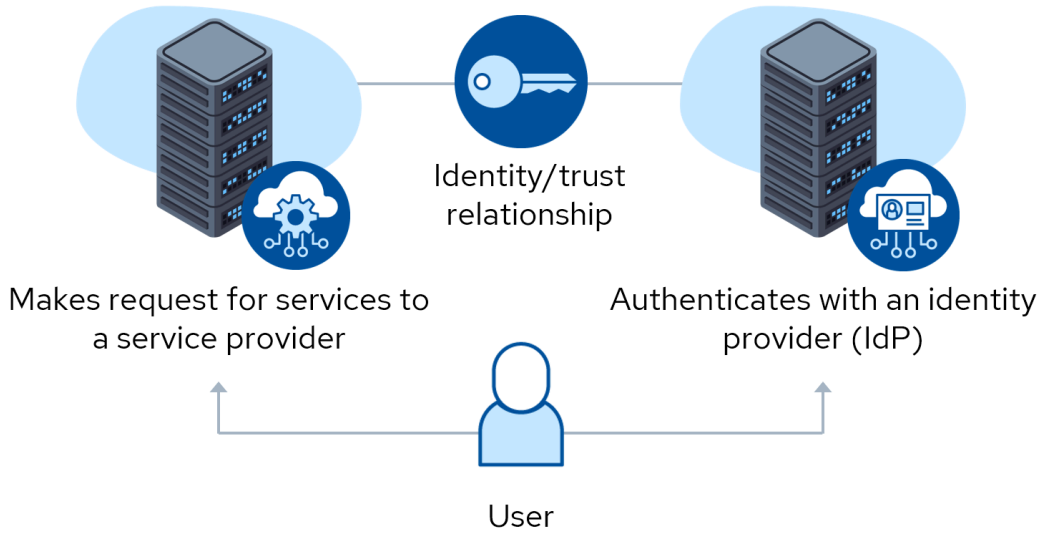


Figure 56: IAM and Secrets Management for Application Security

10.3.2 Secrets Management

Secrets are digital authentication credentials (such as passwords, keys, and tokens) that are used by application or infrastructure services to communicate between them or with other services (as opposed to authentication credentials used by humans). Secrets management is the practice of securely handling those credentials. Effective secrets management ensures these sensitive elements are stored, accessed, and managed securely, preventing unauthorized access, and mitigating the risk of data breaches. It involves tools and policies to create, distribute, rotate, and revoke access credentials systematically, along with secret leak detections, thus safeguarding the integrity and confidentiality of data across the infrastructure.

The following are some functions of Secrets Management:

- **Automatically Supply Credentials:** This is like having a trusted assistant who provides you with the right key at the right time, ensuring that services can access what they need without human intervention (and human error).
- **Secure Storage:** Secrets are stored in a secure way, similar to how valuables would be stored in a bank's vault.
- **Integration with APIs:** Secrets are provided to applications via secure channels when they need to verify themselves, like showing ID when asked.
- **Sharing Secrets Across Teams:** When teams need to use the same credentials, secrets management systems let them do so without seeing the secret, like a shared bank account where you can spend money without knowing the account number.

Secrets management is typically deployed in one of two ways: embedded or Client-Server. Both models aim to keep secrets secure while making them accessible to authorized parts of an application. The choice

between embedded and client-server models often depends on the application's specific needs, such as scalability and security requirements. Choosing the right model is crucial to ensure that secrets– the digital keys to the kingdom– are well-protected yet functional within the application's ecosystem.

- **Embedded Model:** In this model, secrets management is built directly into the application or system—think of it as having a safe in every room of a hotel. It's mostly seen in containerized environments like Kubernetes, where applications are packaged with all their dependencies (including the secrets). Secrets are used once and can be widely shared within the container environment, but this can sometimes be a bit too open, like leaving the safe unlocked when you leave the room.

- **Client-Server Model:** Secrets management in this deployment model is more like a bank with several branches. You have a central server (the main branch) where all the secrets are stored, and clients (other branches) request access to these secrets as needed. This setup can handle a high volume of requests because it's designed to spread the workload across multiple servers. Additionally, it replicates secrets across different servers to ensure they are always available when needed and provides backup in case one server fails. This approach balances security with accessibility, ensuring secrets are safe but still readily available to authorized parts of the system.

Most of the cloud providers today offer alternatives to static secrets. Depending on the deployment scenario, secrets can be avoided by assigning IAM roles/identities to services. For scenarios where secrets must be used, all IaaS/PaaS providers offer secure storage services for keeping secrets safe. These integrate with IAM and eliminate the need to keep secrets in application code, configuration files, or other insecure storage. Third-party services also exist for multi-cloud and on-prem deployments.

The following figure illustrates the process of generating and managing X.509 certificates, including the creation of a key pair, submission of a certificate signing request (CSR), and issuance by a certificate authority (CA).

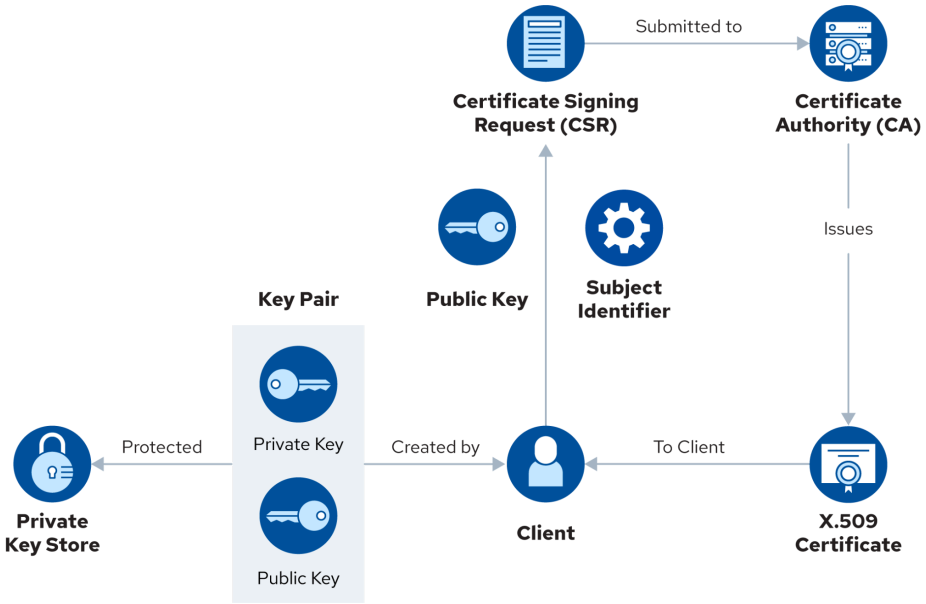


Figure 57: IAM and Secrets Management Process

10.4 DevSecOps: CI/CD & Application Testing

DevSecOps is short for development, security, and operations which automates the integration of security throughout the SSDLC. DevSecOps introduces the "security" piece to the DevOps pipeline. The DevOps pipeline is a set of automated processes and tools that allows developers and operations professionals to collaborate on building and deploying code to a production environment and produce software products rapidly. It builds on the concept of having a CI/CD model. DevSecOps enhances and fortifies this DevOps CI/CD model by ensuring the inclusion of security.

- **Continuous Integration (CI):** Developers frequently merge their code changes into a shared repository. Pre-deployment automated security tests are required in this phase (e.g. SAST)
- **Continuous Deployment (CD):** Once the code passes the CI stage, it is automatically deployed to testing or staging environments. This ensures that code changes are delivered quickly and consistently. Post-deployment automated security tests are required in this phase (e.g. DAST).

DevSecOps emphasizes the presence of security measures and testing from the earliest stages of CI/CD pipelines, ensuring that security considerations are an integral part of the application development and deployment cycle. This approach aims to automate core security tasks by embedding security checks, scans, and tests into the CI/CD workflow, facilitating rapid, yet secure, delivery of code changes.

10.4.1 DevSecOps

DevSecOps melds security with the agile collaboration and automation of DevOps, emphasizing a holistic approach to software development and deployment. At its heart, it leverages CI/CD to automate and streamline processes, making cloud integration seamless and more secure. Adopting standardization ensures that all environments, from development to production, are consistent, reducing the chance of errors. Automated testing integrates security checks into the CI/CD pipeline, enhancing security without sacrificing speed. The concept of immutability in infrastructure, where environments are quickly and reliably created, supports automated deployments and reduces risks associated with manual changes.

Moreover, improved auditing and change management capabilities offer transparency and traceability of changes, bolstering security postures. By integrating security practices into DevOps, DevSecOps optimizes operational efficiencies and significantly enhances the security resilience of applications and infrastructure.

The following figure illustrates the DevSecOps CI/CD cycle, highlighting the integration of DevSecOps phases to ensure continuous and secure delivery of software.

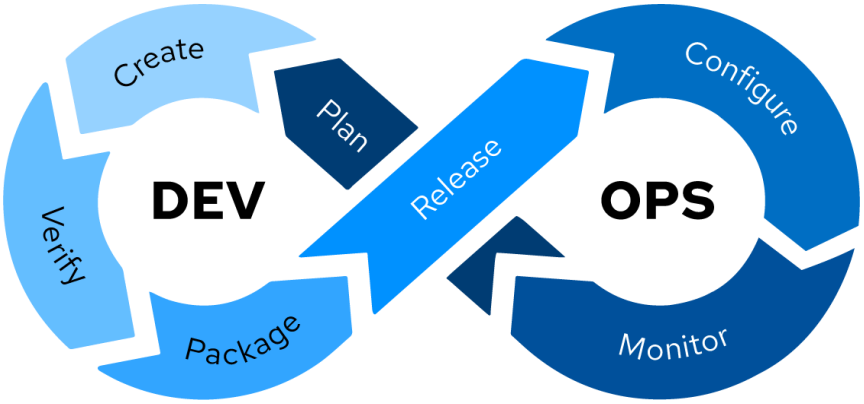


Figure 58: DevSecOps CI/CD Cycle

10.4.2 The Six Pillars of DevSecOps

The six pillars of DevSecOps offer a comprehensive framework for integrating security into the fabric of DevOps practices, aiming to develop secure software efficiently. These pillars underscore the importance of shared responsibility, collaboration, pragmatic tools, practice selection, harmonizing compliance with development, automation to minimize errors, and continuous improvement through actionable metrics. Each pillar is foundational to transforming the culture, processes, and tools used in software development, ensuring security is an integral part of the lifecycle from inception to deployment. This holistic approach facilitates a shift towards proactive security practices, where every team member is empowered to contribute to the security posture of their projects, bridging traditional gaps between development, operations, and security teams.

The six pillars of DevSecOps:

1. **Collective Responsibility**[153]
    a. Security is a shared responsibility across all teams.
    b. Cultivate a proactive and security-aware organizational culture.
2. **Collaboration and Integration**[154]
    a. Essential for DevSecOps success through cross-functional teamwork.
    b. Bridges knowledge gaps and fosters a unified security-conscious mindset.

[153] CSA. (2020) The Six Pillars of DevSecOps: Collective Responsibility.
[154] CSA. (2024) The Six Pillars of DevSecOps: Collaboration and Integration.

3. **Pragmatic Implementation**[155]
    a. Select tools and practices fitting the organization's needs.
    b. Focus on seamless security integration into the development process.
4. **Bridging Compliance and Development**[156]
    a. Align compliance with agile practices through automation.
    b. Integrate security measures within the software lifecycle for enhanced risk mitigation.
5. **Automation**[157]
    a. Central to DevSecOps for streamlining processes and reducing errors.
    b. Ensures efficient, consistent security checks and improves software quality.
6. **Measure, Monitor, Report, and Action**[158]
    a. Implement measurable and actionable metrics for continuous improvement.
    b. Focus on deployment frequency, patch times, testing coverage, and vulnerability response.

This framework highlights a comprehensive approach to integrating security into DevOps, ensuring secure software development through collaboration, automation, and continuous improvement.

10.4.3 DevSecOps in Practice

Expanding on the provided concepts for making DevSecOps work in practice, we develop a structured approach to integrate security seamlessly into DevOps processes.

- **Detect:** Implement real-time monitoring systems that act like vigilant sentinels, scanning for and identifying security issues, threats, or misconfigurations as soon as possible, ensuring rapid response.
- **Automate:** Leverage technology to automate repetitive security tasks, akin to having smart systems that operate independently, from deploying patches to managing configurations, ensuring that the security measures are always up-to-date and consistently enforced.
- **Deliver:** Establish efficient and direct communication protocols ensuring that security alerts reach the appropriate experts through familiar tools, optimizing the response time and effectiveness of the team's actions.

[155] CSA. (2022) The Six Pillars of DevSecOps: Pragmatic Implementation.
[156] CSA. (2022) The Six Pillars of DevSecOps: Compliance and Development.
[157] CSA. (2020) The Six Pillars of DevSecOps: Automation.
[158] CSA. (2024) The Six Pillars of DevSecOps: Measure, Monitor, Report, and Action.

- **Fix:** Integrate security maintenance into daily routines, resolving security issues a regular and proactive part of operations, just as routine cleaning would be in maintaining hygiene standards in a restaurant.

By following these key requirements, organizations can embed security within their daily operations, fostering a culture where security and development go hand in hand to continuously maintain and improve the security posture.

10.4.3.1 Shift-Left & Build Security In

If the SSDLC is seen as a horizontal process of steps, then security has mostly been present only in the last step, in the maintenance phase - as a reactive measure after a security incident on a production application. Shift-Left is the phrase used to indicate that security should move to earlier phases in the SSDLC to ensure secure-by-design and secure-by-default products. Shift-Left drives proactive security by ensuring each phase of the SSDLC, starting from inception and planning, is viewed through the lens of security. This approach is also cost-effective compared to bolt-on security at a later phase in the SSDLC.

The following figure outlines the integration of security across different phases of development in the DevSecOps model, from architecture and development to production, emphasizing continuous security testing and monitoring.

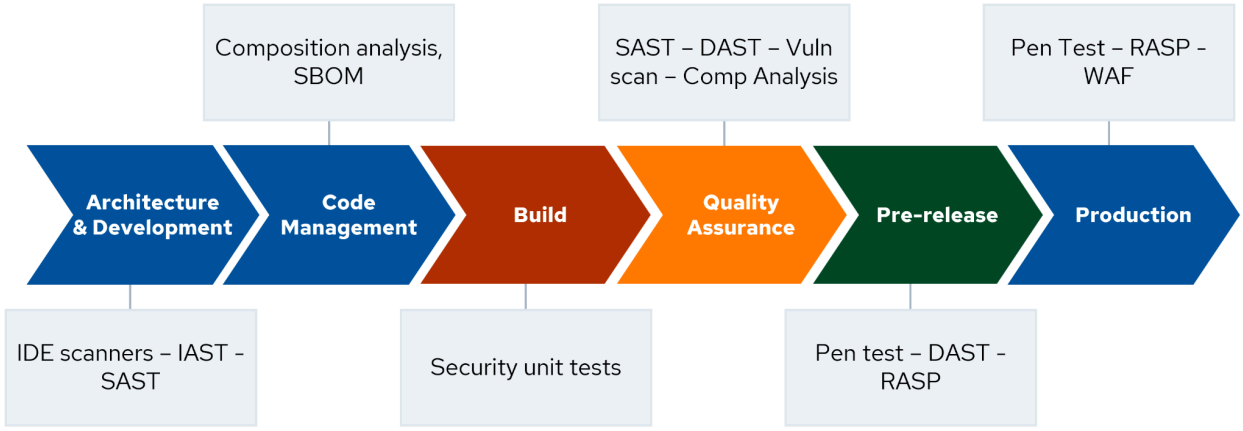


Figure 59: DevSecOps: Integrating Security Across Development Phases

Shift-Left also helps in the early detection of vulnerabilities. Rather than wait to test the completely built product, or worse, wait for attackers in the wild to exploit vulnerabilities, shift-left security tools can empower developers to identify vulnerabilities early and fix them, thereby fortifying the entire development process to produce resilient products.

The table below summarizes proactive security measures applied at various stages of the SSDLC, highlighting where specific security techniques are implemented and their purpose.

Where?	What?	Why?
Integrated Development Environments (IDE)	SAST	Detects source code vulnerabilities and provides the developer real-time feedback as they code
Repository	Software Composition Analysis (SCA)	Detects vulnerable dependencies and libraries
Build phase	Security Unit Tests	Detects module-level security vulnerabilities
Quality Assurance phase	SAST IAST DAST	Detects static code and operation vulnerabilities
Pre-release phase	DAST	Detects operation vulnerabilities
Production	Web Application Firewall (WAF) Runtime Application Self Protection (RASP)	Monitors and prevents attacks

Table 9: Proactive Security Measures Across the SSDLC

10.4.3.2 SecOps: Web Application Firewalls & DDoS

Web applications continue to require robust security measures even after deployment. Implementing Gateway Services, Distributed Denial-of-Service (DDoS) protection and Web Application Firewalls (WAFs) is essential. These tools are designed to ensure that an application remains accessible to legitimate users and can efficiently manage the influx of web traffic, safeguarding against potential crashes due to overloading. It is important to highlight that a WAF is a preventive control and must not be used as a corrective control or as a protection for poorly developed applications. Always remember to bring security to the left on the SSDLC, as was discussed earlier.

There are four common deployment scenarios for WAF & DDoS protection in IaaS/PaaS services:

1. **Agent deployment:** When using IaaS VMs as web servers, a WAF Agent can be installed on top of the OS. This option usually doesn't have DDoS mitigation capabilities.
2. **Cloud provider service:** IaaS/PaaS providers offer integrated WAF and DDoS protection services, usually deployed on the load balancer services.

3. **Third-party marketplace service:** IaaS/PaaS marketplace offers a wide variety of third-party commercial WAF software deployed on dedicated VMs. It is the customer's responsibility to deploy the WAF and ensure routing, redundancy, and load balancing.

4. **WAF & DDoS as a Service**: Using DNS redirect, consumer traffic is routed to a third-party WAF service, examined and filtered, and then routed to the cloud provider environment.

The following figure illustrates the API Gateway security architecture, highlighting the integration of WAF and DDoS protection to ensure secure and efficient management of web traffic.

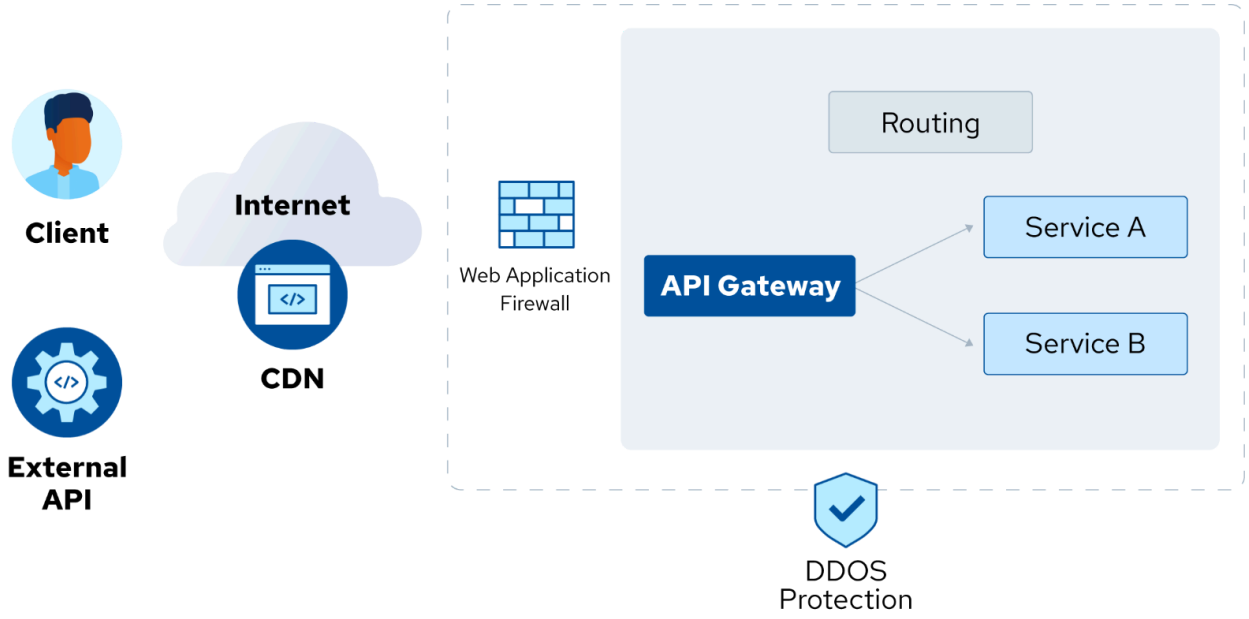


Figure 60: API Gateway Security Architecture

10.5 Serverless & Containerized Application Considerations

In the evolving application deployment landscape, serverless computing and containerization are growing in importance. Serverless computing enables organizations to build applications without managing any underlying infrastructure, offering scalability and cost-efficiency. Meanwhile, containerization encapsulates applications in consistent environments, enhancing portability. Both technologies shape modern deployment practices with their unique advantages, necessitating an understanding of their specific security implications. We explore both in this section.

10.5.1 Serverless & Container Impacts on Application Security

In the evolving landscape of application deployment, serverless and container technologies are reshaping security practices. The security measures and strategies must adapt to these new environments, each with its unique set of considerations. Understanding these considerations is critical for maintaining application security as deployment methodologies advance.

10.5.1.1 Serverless Considerations

The following are examples of serverless considerations:

- **Reduced attack surface**: The transient nature of serverless functions, which perform single, short-lived operations without persistent storage, inherently limits exposure to attacks.
- **Dependency risks**: Reliance on external code or services introduces security risks, akin to using third-party components with unknown safety records in product manufacturing.
- **IAM complexity**: The ephemeral and distributed nature of serverless functions necessitates complex access management, comparable to maintaining security across numerous, constantly changing access points.

10.5.1.2 Container Considerations

The following are examples of container considerations:

- **Isolation risks**: Insufficient isolation in containerized environments can lead to security breaches, similar to inadequate barriers in connected rooms that could allow easy passage for an intruder.
- **Immutable infrastructure**: Containers are designed to be immutable post-deployment, promoting consistency and reducing risks like tamper-evident packaging.
- **Complex configuration management**: As the scale increases, managing the intricate security configurations of containers becomes a challenge, akin to overseeing a complex network of advanced security systems across multiple facilities.

The following figure highlights key considerations for container security, including code, runtime, libraries, environment, and configurations, essential for maintaining robust security in containerized applications.

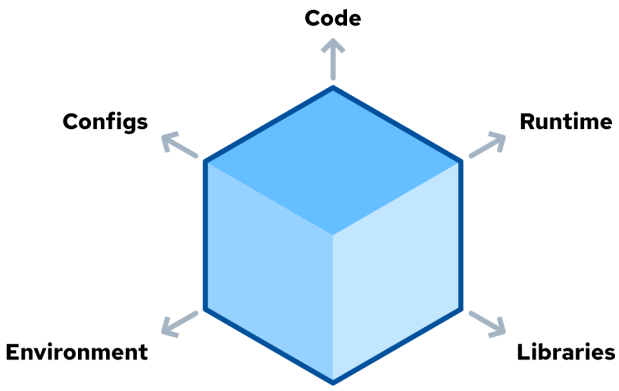


Figure 61: Container Considerations

Summary

Cloud computing is a major driver of application security advancements, necessitating that progress is stable, scalable, and secure. The Secure Development Lifecycle provides essential techniques and methodology guidelines to help craft and maintain secure cloud applications.

To truly fortify your digital ecosystem against ever-evolving cyber threats, it is crucial to embed application security principles at the core of your cloud computing strategy. This involves integrating security from the initial design phase through to deployment and ongoing maintenance.

Key elements include:

- **Secure Architecture:** Building a secure foundation by incorporating security measures into the design phase ensures robust protection against potential threats.
- **Identity and Access Management (IAM):** Implementing a secrets management policy to safeguard business data processed by applications is vital. IAM and secrets management together form the backbone of access control and data protection strategies.
- **DevSecOps:** The integration of DevSecOps emphasizes a commitment to application security throughout the entire development lifecycle. This approach is particularly relevant in modern deployment practices such as serverless computing and containerization.
- **Continuous Monitoring and Improvement:** Employing continuous monitoring, threat modeling, and automated security testing helps in identifying and mitigating vulnerabilities early, ensuring a resilient application security posture.

By following these guidelines and leveraging the Cloud Security Alliance (CSA) recommendations, organizations can create a secure, resilient, and scalable application environment that addresses the unique challenges and opportunities presented by cloud computing.

Recommendations

CSA Secure Development Lifecycle (SSDLC):

- Secure Design and Architecture: Apply technologies and tools during the design phase to integrate security early, avoiding higher costs and bottlenecks later.
- Continuous Build, Integration, and Testing: Employ tools and processes to test for vulnerabilities before deployment to prevent security breaches.
- Continuous Delivery and Deployment: Conduct pre-deployment safety checks to ensure applications are deployed on secure infrastructure.
- Runtime Defense and Monitoring: Implement practices to continuously identify and mitigate vulnerabilities and inefficiencies post-deployment.

Adopt Structured Threat Modeling:

- Apply the STRIDE framework to categorize threats: Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, and Elevation of Privilege.

Focus on Secure Cloud Design:

- Use Platform as a Service (PaaS) and other CSP services to offload security responsibilities to the provider.
- Implement least privilege Identity and Access Management (IAM) for all components.
- Use CSP services like load balancers and security groups to minimize internet-facing exposures.

Integrate Security Testing Methods:

- Static Application Security Testing (SAST): Automate code reviews to identify vulnerabilities and logic errors before deployment.
- Software Composition Analysis (SCA): Audit external components for vulnerabilities and licensing risks, and create a Software Bill of Materials (SBOM) for transparency.

Conduct Comprehensive Post-Deployment Testing:

- Dynamic Application Security Testing (DAST): Perform black-box testing to assess the application's security posture from an external perspective.
- Dynamic Analysis (Fuzzing): Input unpredictable data to identify errors and vulnerabilities during operation.
- Interactive Application Security Testing (IAST): Combine SAST and DAST to identify vulnerabilities both in the code and at runtime.
- Penetration Testing: Conduct simulated attacks to exploit known vulnerabilities and test the system's resilience.
- Bug Bounty Programs: Utilize the ethical hacker community to discover and report vulnerabilities.

Enhance Access Control:

- Apply the Principle of Least Privilege to minimize unauthorized access.
- Implement continuous monitoring to detect and address aberrant access patterns.
- Use segregation of duties to dilute access power and prevent misuse.
- Adopt federation to streamline and secure cross-platform interactions.

Secrets Management:

- Automatically supply credentials to minimize human error.
- Store secrets securely, akin to valuables in a vault.
- Integrate secrets with APIs via secure channels.
- Facilitate sharing of secrets without exposing them, like a shared bank account.

Integrate Security into CI/CD Pipelines:

- Implement continuous integration and deployment with security checks embedded.

- Use shift-left strategies to identify and address vulnerabilities early in the SSDLC.
- Automate repetitive security tasks to ensure consistent enforcement and timely updates.
- Foster collaboration between development, operations, and security teams.

Adapt Security Strategies for Modern Deployments:

- Serverless Considerations:
    - Leverage the reduced attack surface of transient serverless functions.
    - Address dependency risks and manage complex IAM requirements.
- Container Considerations:
    - Ensure robust isolation to prevent breaches.
    - Use immutable infrastructure to promote consistency and security.
    - Manage complex security configurations as container deployments scale.

Additional Guidance

- Six Pillars of DevSecOps | CSA
- Information Security Management through Reflexive Security | CSA
- FaaS Serverless Control Framework (Set) based on NIST 800-53 R5 Controls | CSA
- The Six Pillars of DevSecOps - Pragmatic Implementation | CSA
- Recommendations for Adopting a Cloud-Native Key Management Service | CSA
- Security Guidelines for Providing and Consuming APIs | CSA
- C-Level Guidance to Securing Serverless Architectures | CSA
- The Six Pillars of DevSecOps: Collective Responsibility |CSA

Domain 11: Incident Response & Resilience

Introduction

Incident response (IR) is a critical aspect of any information security program, as it is highly likely that your organization will experience a security breach at some point, regardless of how strong your security posture is. While many organizations have an IR plan for investigating attacks, cloud adaptation introduces distinct variations in processes, technologies, and governance, adding complexity to responding to incidents.

This domain seeks to identify and explain best practices for cloud incident response (CIR) and resilience that security professionals may use as a reference when developing their own incident plans and processes . This domain is organized according to the commonly accepted IR Lifecycle described in the CSA Cloud Incident Response Framework[159] and NIST Computer Security Incident Handling Guide (NIST SP 800-61 Rev. 2)[160]. Other resources include the CSA incident response research hub[161] and other international standard frameworks for IR, such as ISO/IEC 27035 and the ENISA Strategies for IR and cyber crisis cooperation[162]. Security professionals may use these as a reference when developing IR plans and conducting other activities during the IR lifecycle.

Learning Objectives

The learning objectives for this domain aim to provide readers with knowledge on:

- Distinguish between events, incidents, and breaches and use a response process to react.
- Prepare and respond to incidents.
- Detect and analyze relevant data.
- Contain, eradicate, and recover.
- Perform resilience planning for failure.

[159] CSA. (2021) Cloud Incident Response Framework.
[160] NIST. (2012) Computer Security Incident Handling Guide. Comment period for potential revisions closed mid-2024.
[161] CSA. (2024) CSA Research landing page.
[162] ENISA. (2024) Cyber Crisis Management.

11.1 Incident Response

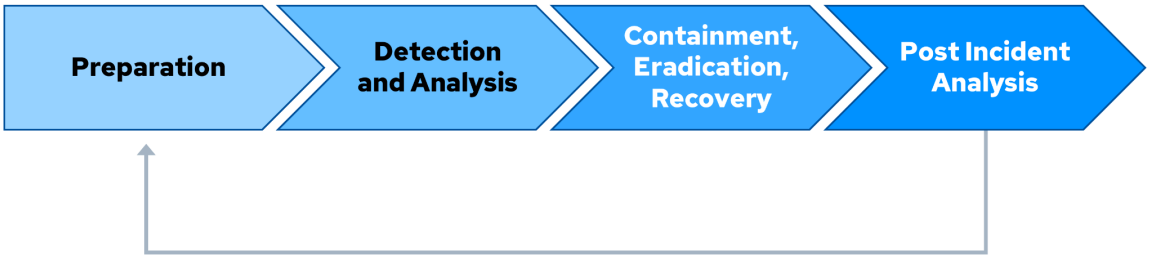
IR is about dealing with unexpected events. This necessitates a clear differentiation between events, incidents, and breaches, each representing a distinct level of threat and requiring tailored response strategies. Recognizing and classifying these occurrences accurately is foundational to maintaining the integrity, availability, and confidentiality of cloud services, and ultimately, safeguarding the digital assets and trust of stakeholders involved.

An *event*, in the context of cloud security, is any observable occurrence in a system or network, which may or may not indicate an underlying issue that is related to security.[163] All *incidents* are events but not all events become incidents unless they violate explicit or implied security policies, potentially compromising normal operations and posing a threat to the cloud environment. Incidents demand immediate attention to contain and mitigate their effects, preventing escalation. At the apex are *breaches* which signify a successful penetration or circumvention of security measures, leading to unauthorized access or extraction of data. Understanding the gradations from events, through incidents, to breaches is used to create an effective IR strategy.

11.1.1 Incident Response Life Cycle

IR and management frameworks have been developed and documented by many organizations. Different frameworks have their objectives and target audiences. CSA has adopted the commonly accepted phrase of *Incident Response Life Cycle* described in NIST Computer Security Incident Handling Guide (NIST 800-61 rev2 08/2012)[164].

An IR lifecycle serves as a go-to guide for cloud customers to effectively prepare for and manage cloud incidents. The IR life cycle described by NIST includes the following phases and major activities: Preparation, Detection and Analysis, Containment, Eradication and Recovery, and Post-Incident Activity.[165]



Based on NIST 800-61rev2 (Post Incident Analysis replaces Post Mortem) [166]

Figure 62: Phases of IR Life Cycle in Cloud Security

[163] Additional detail on events is covered in Domain 6: *Security Monitoring*.
[164] NIST. (2012) Computer Security Incident Handling Guide. Comment period for potential revisions closed mid-2024.
[165] The Incident Response Life Cycle is described in the CSA Cloud Incident Response Framework and NIST 800-61rev2. This content focuses on incident response for the cloud environment, however, while cloud IR is primarily separate, it often overlaps with traditional IR. Responders focused on both environments should work closely together using this consistent response process.
[166] NIST. (2012) Computer Security Incident Handling Guide, Figure 3-1. Incident Response Life Cycle, Page 21.

Preparation: Establish an IR process.

- Build a team with assigned roles and responsibilities. Include training, and run exercises.
- Establish a communication plan and facilities.
- Grant responder access to environments and tools, such as incident analysis services, hardware, and software.
- Create internal documentation (port lists, asset lists, network traffic baseline).
- Evaluate infrastructure: proactive scanning and monitoring, vulnerability and risk assessments.
- Subscribe to third-party threat intelligence services.
- Evaluate cloud service providers and their capabilities to aid in IR regarding the services/resources you consume.
    - Audit logs, snapshots, forensics capabilities, and e-discovery features.
- Conduct backup restoration testing regularly and DR tests at least once per year to ensure that your IR plans are up-to-date and effective.

Detection & Analysis: Identify security incidents and analyze their impact.

- Conduct detection engineering
- Utilize alerts from Cloud Security Posture Management (CSPM), security information and event management (SIEM), workload protection and network security monitoring
- Validate alerts to reduce false positives and escalation
- Estimate the scope of the incident
- Assign an Incident Manager to coordinate further actions
- Build a timeline of the attack
- Determine the extent of the potential data loss or impact
- Notify proper channels and coordinate activities
- Communicate the incident containment and recovery status to senior management

Containment, Eradication & Recovery: Isolate the incident to prevent further damage and resolve the root cause. Recover and restore affected systems.

- Containment: Involves isolating identities and workloads, taking systems or services offline, and considering data loss versus service availability.
- Eradication and recovery: Clean up compromised assets and restore systems and services to normal operation. Deploy controls to prevent similar incidents.
- Capture incident data: Document the incident and gather forensic evidence (chain of custody).

Post-Incident Analysis:[167] Learn from the incident, document, and improve future responses.

- Lessons learned: What could have been done better? Could the attack have been detected sooner? What additional data would have been helpful to isolate the attack faster? Does the IR process need to change? If so, how?
- Sharing learnings: Share lessons learned with the broader security community.

[167] Some incident response framework versions describe a postmortem phase that the industry now refers to as *Post-Incident Analysis*. CSA, NIST, and other industry authoritative bodies follow this change in nomenclature.

Cloud impacts activities in all IR phases, bringing new benefits and challenges. At the same time, many incidents may span cloud and traditional infrastructure and devices, requiring incident responders to ensure they don't develop tunnel vision and only look at one facet of an incident.

11.2 Preparation

The Preparation phase can be broken into four major categories which will be covered in detail in the following sections:

- Changes due to the relationship with the cloud provider
- Changes in responder training
- Changes needed to support the CIR process
- Changes required to support CIR technologies

The Preparation phase is often the most challenging part of starting a CIR program. The fundamentals of an IR process don't change, but the technical and operational differences of cloud computing significantly impact the mechanics of those processes. Failure to account for these differences in the Preparation phase will severely constrain any ability to respond effectively.

The following are some key CIR differences:

- Cloud operations tend to be more distributed, with individual teams defining and managing their own infrastructure. This leads to access and telemetry collection issues.
- Every cloud platform is deeply different from every other at the most fundamental technical levels. IR requires proper and compatible tooling and deep subject matter expertise.
- Cloud attacks can be highly automated and occur incredibly quickly. Cloud resources can easily become public or shared over the internet with just a single configuration change. This requires extremely rapid responses for certain incident types, which may require process and technical changes to support and enable rapid 24/7 response capabilities.
- Responders will often need extensive direct access to affected deployments and resources on-demand. Log access is not typically sufficient to fully analyze and respond to a cloud incident. Analysts and responders will need to quickly understand the context of what they see in logs and this will likely involve directly reviewing configurations and resources through application programming interface (APIs) or web UIs.

11.2.1 Incident Response Preparation & Cloud Service Providers

Cloud incidents are shared incidents, even when the customer owns all of the affected resources. Any incident using a public cloud requires an understanding of the contractual agreement, including the specific service level agreements (SLAs), and which resources your provider offers. Depending on your relationship with the provider, you may not have direct points of contact and might be limited to whatever is offered through standard support. Many providers have different support levels, and you should consider subscribing to the level of support that is appropriate to the business criticality of the service (e.g., direct contacts and faster support for any business-critical deployments or deployments with highly sensitive or regulated data).

In addition to paid support, some providers will extend some level of IR assistance to customers at no additional cost. For each provider you work with, it's important to have a list of the incident support options (paid and free) and contact information for each of them. This should be integrated into a cloud deployment registry.[168]

At some point in time, your provider will likely detect an incident involving your resources/deployment. Therefore, it is important for each deployment and provider that you keep your contact information up to date and ensure it routes to your Security team as one of the default recipients.

These notifications will tend to fall into the following categories:

- Abuse notices when the provider detects your resources potentially being used to harm others. Keep in mind these are not always accurate and will require validation.
- Notices of security exposures within your resources, such as the detection of private access credentials posted into a public code repository.
- Notices of suspicious activity that could indicate a breach.
- Notices of an incident at the cloud provider that could affect your data or resources (e.g., the provider has a breach, detects a successful attack, or has a data exposure).
- For click-through services, notifications will likely be sent to your registration email address; these should be controlled by the enterprise and monitored continuously.

At this stage, it's also important to plan for incidents that affect your provider and are out of your control. For example, there are documented cases of public vulnerabilities and denial-of-service attacks that impact a provider. As a customer, you aren't necessarily helpless though, and there may be response activities you can perform yourself, depending on the nature of the incident. The IR team should understand this risk and plan for some possible scenarios with corresponding mitigations. This will usually require coordination with business continuity activities.

11.2.2 Training for Cloud Incident Responders

Although CIR practices share many characteristics and processes with traditional IR practices, responders need to understand the process and technology differences.

A combination of training and various exercises can quickly help a responder or team develop the required skills, such as:

- Generalized CIR training helps build the foundational skills that work across cloud providers. This is also a good choice to improve cloud awareness even with responders that won't be dedicated to cloud incidents.
- Provider-specific technical training is essential for any responder working on a major platform, particularly IaaS. This training should not be limited to just using the IR tools offered by the provider, but get into the deep details, such as how to quarantine an exposed service credential, how to analyze logs, and so on.

[168] Cloud deployment registry is covered in detail in *Domain 2: Cloud Governance*.

- Scenario-based exercises in simulated environments help gain practice at core skills like log analysis, threat hunting, and resource quarantine.
- Full exercises and red teaming are designed to test the entire IR process.
- Tabletop exercises with distributed cloud teams and leadership help ensure different teams can work together and coordinate efforts. Tabletop exercises may include simulations of large-scale incidents, such as a provider breach.

11.2.3 Updates to Support Cloud Incident Response Processes

The core of the IR process doesn't really change, but IR teams will want to adjust their processes to account for some of the differences. Standard IR runbooks[169] and playbooks[170] are not optimized for most cloud incidents. They tend to focus on network packet captures, forensics, and other activities that, even when needed for cloud, come after other priorities like ensuring there was no Identity and Access Management (IAM) or management plane privilege escalation. New playbooks and runbooks should be created for expected incident types, and these should be cloud-specific. Cloud playbooks and runbooks should include when to engage non-CIR experts and processes for incidents where both sets of skills and processes are needed, such as a cloud breach that extends to resources in the data center across a hybrid connection, or malware analysis for a compromised cloud resource.

On-premise playbooks and runbooks should also be updated for two major incident types. First, the detected exposure and abuse of cloud credentials obtained by an attacker in a non-cloud attack. Then, attacks on cloud resources from compromised resources such as a server or workstation in the non-cloud environment. As a process, there should be a requirement that after a new type of incident, a playbook or runbook is created for that incident type. Unless you are strictly cloud, you will experience incidents that bridge cloud and traditional infrastructure, such as incidents that involve employee devices. Thus, it's important to ensure that if you have responders dedicated to the cloud, processes exist to handle hybrid incidents that span both environments.

Business continuity, leadership, legal, and compliance teams, if present, will need to understand their roles in cloud incidents and adjust their processes accordingly. An exposure of your customer data by a cloud provider will have different crisis communications and legal and regulatory requirements. Response processes, including playbooks, runbooks, and overall processes, must pay particular attention to the implications of any cloud incident involving the management plane. Attackers today may bridge into the management plane and escalate privileges. If responders focus only on cloud resources, such as a compromised virtual machine (VM), they may miss the most damaging aspects of the attack. Cloud incidents can be blindingly fast due to attacker automation. Processes must account for this difference in speed. For example, you can't effectively respond to a real-time cloud incident if the incident handler is working off log data that is 15-60 minutes old.

11.2.3.1 Enable Responder Access

Some key adjustments are needed by other teams and the organization to support CIR. The CIR team should have persistent read access to all deployments. It is effectively impossible to investigate a cloud

[169] A *runbook* is a set of instructions for completing a routine task.
[170] A *playbook* outlines the organization's approach and worker responsibilities. More details provided in 11.2.4.1.

incident without being able to review the resources and configurations involved. All use of these privileges should be logged and reviewed. Two levels should be supported, depending on the capabilities of the cloud provider:

- Read access to metadata and configurations, sometimes called *security audit*, should be persistent, and the default level of access for responders.
- Full-read access, which allows review of the data, not just the metadata, can, and in many cases should, require multiple approvals to use and follow a break-glass process.

Higher-level responders should have write access to critical situations that need an immediate response, such as the public exposure of data. They will be unable to use traditional approaches like blocking access with a firewall. This access should be tightly controlled, require approval, and use a break-glass process but does need to be available 24/7. These responders should be very familiar with the cloud platform and should only use this access in the most critical of incidents. This will typically involve senior-level approval, as this gives the authority to trade business continuity for security.

In cloud environments, distributed teams are more likely to manage their own infrastructure (IaaS/PaaS) directly. This can result in increased types of activity that are hard to distinguish from attacks. Centralized IR teams will lack the context or knowledge to understand if much of what they see is indicative of an attack or is an intended activity. It is essential to establish clear, real-time communications with deployment owners to include them in the IR process. Many organizations are seeing success by using ChatOps to send issues to teams to validate if they were intentional or a potential indicator of an attack. This can be integrated in a variety of ways and allows teams the ability to click responses to escalate or de-escalate potential incidents. ChatOps can be an especially good option if cloud teams are already using it since that allows security to integrate using the same communications tool as the teams. Email and ticketing systems are also options, but are slower and those time delays may lead to responders having to hunt down contacts.

The IR team should have access to the cloud deployment registry and that registry should have current information to contact the business owner and technical leads. Incident responders may need access to continuous integration/continuous deployment (CI/CD) pipelines, code repositories, and other locations that manage and modify cloud configurations. Response processes for Containment, Eradication, and Recovery will likely need to use these resources and services. This is a case where the IR team and the deployment or application owners must be prepared to work together.

11.2.4 Technology Updates to Support Cloud Incident Response

The most important technology changes required to support CIR are collecting the required security telemetry and implementing cloud-native threat detectors.[171] This section focuses on additional preparatory technology changes to support the CIR process.

[171] Additional details for telemetry and cloud incident response are covered in *Domain 4: Organization, Tenancy, & Enterprise Management* and *Domain 6: Security Monitoring*.

Key technology updates include:

- **Build an incident response analysis environment:** This is typically a deployment with any needed analysis tools and the ability to connect to other cloud deployments to extract forensics, logs, and other data. Access to other accounts can involve a break-glass emergency access process.
- **Build an incident response environment:** This environment is often a separate cloud deployment and has response tools that can modify resources and configurations in target deployments. It can be the same deployment as the IR analysis environment but is ideally separate due to the higher-risk entitlements it needs, such as full admin rights.
- **Cloud detection and response (CDR):** CDR tooling can overlap with SIEM technology but focuses on handling real-time security event data (as opposed to logs), routing and triaging alerts, and automatically enriching alerts. These tools are where threat detectors for cloud live and are often used for triaging, escalating, triggering automated remediations/responses.
- **Forensics:** Cloud forensics for VMs and containers require updated tooling that will likely need to run within the same cloud provider as the source resources. Other forensics sources, like source log files, may need to be copied and preserved, which in some cases may only be performed by the provider.
- **Security, Orchestration, Automation and Response (SOAR):** SOAR tooling should support cloud operations and automation, such as connecting to a cloud provider to enrich and support analysis, automatically performing forensic imaging, and other cloud actions.
- **Other automation and response tools (e.g., "Jump Kits"):** Most cloud incident responders find themselves using a mix of commercial tools, custom scripts, and open-source tooling to support their investigations and responses. These may or may not be integrated into the Cloud SOAR and CDR platforms, depending on the other tooling that's available. Even in those cases where these tools are not integrated, many responders have additional tools they will want available for different kinds of investigations.
- **Attack simulation:** Tools that assist in responder training and red teaming[172] by simulating attacks either in designated simulation deployments or through fault injection/simulations in production environments. These are important not only for training, but to validate that detectors and telemetry are working properly.
- **Detection engineering:** Cloud-native threat detectors typically involve a mix of log analysis, real-time event monitoring, and configuration change monitoring. Detection engineering activities will need to account for these new kinds of data sources and activity flows and will likely require technology changes to support the lifecycle management of cloud threat detectors.

[172] Redteam.guide. (2022) The process of using Tactics, Techniques, and Procedures (TTPs) to emulate a real-world threat with the goals of training and measuring the effectiveness of the people, processes, and technology used to defend an environment.

11.2.4.1 Runbooks & Playbooks

Runbooks and playbooks are documented processes for handling specific incident types. Organizations will need to update these for cloud incidents and create new runbooks/playbooks for responding to new cloud incident types. Runbooks and playbooks have different definitions but, at the core, achieve the same goal. In the IR context, they are the series of documented steps to take when investigating and responding to specific incident types.

For example, if an alert fires for the potential external abuse of a credential from an unknown source IP, the runbook/playbook would provide step-by-step guidance on how to investigate and respond. Modern playbooks will often be implemented in an automation system (like a SOAR platform) which can perform some steps using automation and include others like pre-build analysis queries.[173]

The following are important considerations for runbooks and playbooks:

- **Specificity of runbooks and playbooks:** Begin by emphasizing that runbooks and playbooks should be tailored for specific platforms and services. This ensures that the response is relevant and effective for the unique aspects of each situation.
- **Version control:** Stress the importance of maintaining these documents in a version-controlled repository or SOAR system. This practice helps in tracking changes over time, ensuring that the team is always working with the most up-to-date information.
- **Creating new runbooks/playbooks:** Any time there's a new type of incident, it's crucial to develop a corresponding playbook. This proactive approach ensures that if this type of incident happens again, the team is prepared for it.
- **Planning for SOAR failures:** Even if a SOAR system is in place, it's necessary to have a plan for its potential failure. Know that technology can fail, and having a manual process or backup plan is essential.
- **Automation integration:** Discuss how automation can be woven into the IR process. Explain that automation should trigger actions that assist in quicker resolution of incidents, but there should also be checks to ensure automation does not interfere or exacerbate the issue.

11.3 Detection & Analysis

The fundamentals of incident detection and analysis don't necessarily change at a high level with the introduction of the cloud, but the details do change significantly.

Key cloud differences include:

- New telemetry for detection and analysis that cloud introduces.
- Attack surface of the management plane has to be the primary focus during any response.

[173] AWS. (2020) Well-Architected Framework: Concepts - Runbook and Playbook.

- Higher rate of activities in the cloud, which include the speed of attackers (who are highly automated) and the speed of change of cloud environments themselves.
- Lack of a traditional network perimeter and the addition of IAM blast radius.
- API-driven nature of cloud and the ephemeral nature of resources.
- Decentralized management of infrastructure by cloud and development teams.
- Automation, infrastructure as code, serverless, and other cloud-native technologies.

These differences sometimes improve our ability to detect and analyze incidents, while others create new challenges. The guidance in this section highlights these major differences and how to adjust detection and response activities.

11.3.1 Detection & Threat Detectors

It is necessary to build threat detectors for the management plane and IAM activity. This is where the most destructive activity can occur since an attacker can potentially directly modify infrastructure. These detectors need to be activity-focused and not threat-actor-focused. Cloud attackers won't often show up using IP lists or connection headers. Activity is directly at the API level and often originates from compromised environments in the same cloud provider. For example, an API call to share a snapshot with an unknown deployment.
Most attacks on the cloud management plane rely on lost, stolen, or abused credentials. The use of credentials from outside your known network and IAM perimeters can indicate a potential incident. Since many attacks happen with compromised identity/credentials, only behavioral detectors[174] will detect this type of intrusion. In addition, cloud attacks can be heavily automated and move incredibly quickly, resulting in data being exfiltrated or even made public within seconds or minutes of the initial compromise. Detectors for the most critical activities (e.g., private data becoming public) should operate in real-time where possible, or at least within minutes. This is a much tighter timeframe than most responders are used to dealing with in traditional infrastructure.

Configuration changes, such as creating a new IAM user or sharing a resource with an unknown account/subscription/project, can be excellent sources for cloud detectors. These "configuration alerts" can come from direct instrumentation or using a CSPM tool (from the cloud service provider (CSP), a third party, self-created, or Open Source). An incident responder probably won't know if a particular misconfiguration is an attack, a mistake, or required for that application stack. This is why clear and direct communications with the cloud account team are important so the responder can quickly determine if it is a mistake or an attack. Some organizations are adopting ChatOps or similar communications and sending the alerts directly to the responsible team, so they can respond in-app with buttons to indicate a mistake, an exemption request, or that the activity is unexpected and should be escalated.

CSP-security alerts are often an excellent source for detection but can be problematic when they aren't tuned and filtered. For example, they can be high quality in a locked-down production environment but result in a large number of false positives in a development environment. Some alerts, such as those for crypto-mining and potential ransomware, tend to be of higher quality even in less structured

[174]NIST (2021) *Detecting Abnormal Cyber Behavior Before a Cyberattack* - Behavioral anomaly detection involves the continuous monitoring of systems for unusual events or trends. The monitor looks in real time for evidence of compromise, rather than for the cyberattack itself.

environments. Others that focus on user behavior tend to be less useful in more dynamic non-production environments.

Detection engineering also needs to account for "traditional" sources of events, such as compromised operating systems, web attacks, and database attacks. Cloud networks will rarely be instrumented for full packet capture and monitoring due to the inherent differences in software-defined networks. However, flow and DNS activity can be excellent sources for building cloud-native network threat detectors. Detection engineers can and should use up-to-date sources on attacker techniques, such as the CSA *Top Threats* report and MITRE ATT&CK for Cloud[175]. These models describe attacker activities that can be used to build threat detectors based on attacker actions, not just tool or origin signatures.

Detectors should have a lifecycle and ideally be managed using version control and CI/CD pipelines compatible with modern DevOps/DevSecOps practices. Since compromised and abused credentials are a major source of cloud breaches, the use of canaries and honey tokens can be an excellent detection tool to identify compromised identity repositories.[176]

Canaries and honey tokens should be integrated into the IR process, and trigger an immediate investigation that focuses on tracing how credentials were obtained and using this to track the attacker. Honeypots may also be used, which are system/network-based as opposed to the credential-focused canaries/honey tokens.

11.3.2 Cloud Impact on Incident Response Analysis

Cloud computing environments necessitate a departure from traditional IR analysis due to their ephemeral nature, scalability, and decentralized control. The focal point of incident analysis in cloud settings is often the management plane, which offers a comprehensive view of cloud activities through its logs. These logs are invaluable for identifying unauthorized access, misconfigurations, and other anomalies that could indicate a security incident.

The dynamic nature of cloud environments, where resources can be rapidly provisioned and decommissioned, demands that IR teams adapt their methodologies. This includes leveraging automation and machine learning to keep pace with the speed of cloud operations and configuration changes. Analysis in cloud environments also prioritizes identifying publicly exposed resources, necessitating swift action to mitigate potential breaches or compliance issues.

Due to the different nature of the cloud, the priorities for analysis should change. The primary focus should be on management plane activity, not resources. Attackers can cause extensive damage or data theft depending on their ability to access the management plane. Attackers will also try to compromise resources in a deployment, and then start trying to use the resource privileges to bridge into the management plane. Thus, even an isolated resource compromise, such as a hacked VM, can open up the management plane to the attacker if that resource has any internal permissions or stored credentials.

Another key focus is any resource that was made public or shared unexpectedly with another cloud deployment of unknown origin. These are common exfiltration techniques, and any completely public

[175] MITRE. (2024) *Threat Intelligence Program*.
[176] These techniques are covered in more detail in *Domain 6: Security Monitoring*.

exposure is obviously of very high concern. Immediate containment may be required before fully investigating the remaining scope of the incident. Since lost, stolen, or abused credentials are the most common source of cloud-native breaches, the analysis should focus on identifying any IAM entities involved in the account and determining their scope of entitlements (the "IAM Blast Radius") and all corresponding activity. Indicators, such as different source IP addresses, can help discern between expected and unexpected use of these credentials.

Cloud management plane activity logs are exceptionally powerful tools to trace an attack since they often show all activities and can't be modified or deleted by an attacker. Even if an attacker can delete stored logs, most major cloud providers will either still provide something on the order of 90 days of activity in API logs or have copies of the logs that are retrievable if you engage incident support. Some providers only record change events in their API logs and will not show Read activity, which eliminates the ability to track reconnaissance.

When security telemetry is fed into a central SIEM or security data lake[177], the analysts may need to directly access and review local versions of the logs using their read access. They may need to see raw logs after a log is ingested and normalized by the logging platform without keeping the original, complete version. The analysis will often require the analysts to review the configuration of involved resources to handle an incident properly. They won't necessarily be able to rely on log analysis alone. For example, the change of a security group won't help them understand if there is an exposed resource using that group and what the possible exposure risk is.

The analysis may still require traditional skills since cloud breaches aren't necessarily limited just to the cloud platform. For example, the cloud analyst may determine that a credential was stolen from an employee's laptop or the laptop itself was the source of the attack without the employee's involvement. The analyst should engage their appropriate peer with the skills for attacks that cross subject matter domains. For environments managed using CI/CD, the analysis should include the pipeline since that is a major target for attackers and a powerful vector to compromise cloud applications and infrastructure.

There are multiple places to perform detection as seen in the figure on the following page:[178]

- CDR
- SIEM
- CSPM/Cloud Native Application Protection Platform (CNAPP)
- CSP/Identity Provider (IdP)

[177] This topic is covered in more detail in *Domain 6: Security Monitoring*.
[178] This topic is covered in more detail in *Domain 6: Security Monitoring*.

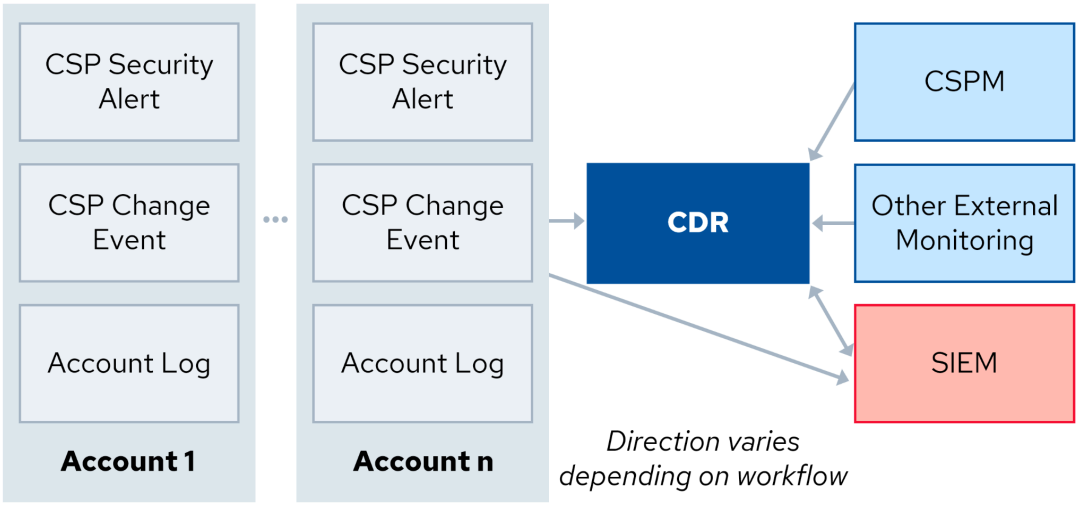


Figure 63: IR Analysis Workflow

11.3.3 Analysis Priorities: RECIPE PICKS

RECIPE PICKS is a mnemonic developed by Rich Mogull of Securosis for training cloud incident responders on their initial analysis priorities. These represent the first places to focus analysis during a cloud management plane incident and can be used to resolve a high percentage of incidents.

- **R**esource (current config/state)
- **E**vents (API call(s) on that resource)
- **C**hanges (diff plus associated API calls)
- **I**dentity (who made the triggering change or API call)
- **P**ermissions (of the identity; informs the blast radius)
- **E**ntitlements (of the resource; e.g., it's IAM role or managed identity)
- **P**ublic (is it public?)
- **I**P (all API calls from that IP address)
- **C**aller (all other API calls from the calling identity)
- Trac**K** (look for indications of a pivot; e.g., role chaining)
- Foren**S**ics (on a resource, or digging into resource logs)

Figure 64: RECIPE PICKS: IR Analysis Priorities

Note: the order doesn't matter, except the last two (especially forensics). It is more important that all this information is collected and analyzed early in the process.

11.3.4 Cloud System Forensics

Cloud forensics falls into two major categories: analysis of management plane, service, and other logs, and system forensics for VMs and containers. Traditional digital (systems) forensics methodologies often

rely on physical access to hardware and local data storage, which is not possible in the cloud. Instead, cloud forensics requires IR teams to work within the constraints and capabilities provided by CSPs.

Key aspects of cloud forensics include:

- **Snapshots:** Nearly all cloud providers and container management systems support snapshots, which can be used for forensics analysis. Grasp how and why to take snapshots of storage volumes immediately when an incident is detected to preserve the state of the VM for analysis.
- **Volatile memory acquisition:** Without the ability to instrument hardware, if memory forensics are required, the responders will need to install software tools which will also affect the system.
- **Log analysis:** Management plane logs, along with system, application, and user activity logs, can also be used to present a fuller picture of incidents even when the focus is on VMs/containers. They can, for example, help identify an attacker who obtained system credentials and pivoted to the management plane.
- **Evidence preservation:** Preserving digital evidence in cloud environments requires a thorough understanding of the backup and data retention policies of both the CSP and the CSC and the chain of custody of snapshots.

The following is an example of using a forensics acquisition and analysis environment and collecting storage volume snapshots from a compromised workload in a separate deployment.

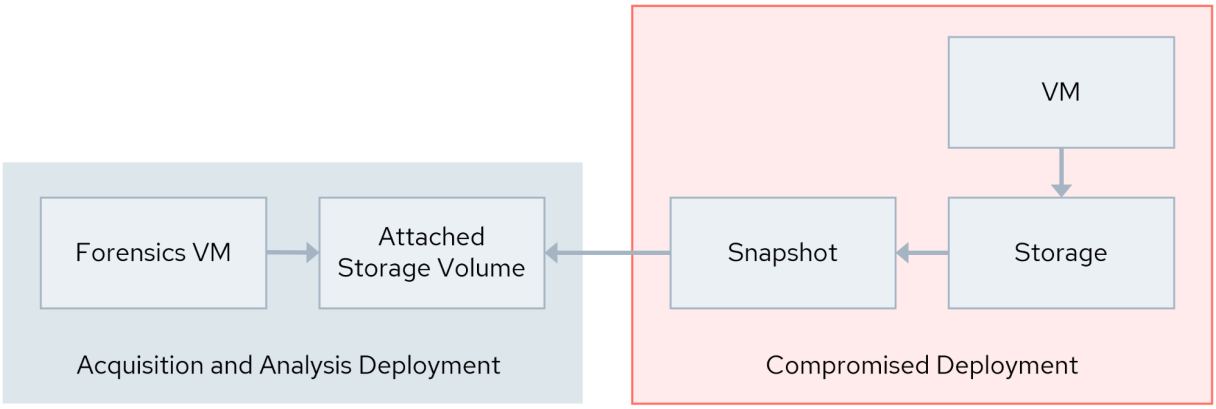


Figure 65: Cloud Forensics: Snapshot Acquisition and Analysis Process

11.3.4.1 Cloud Forensics: Container & Serverless Considerations

The rise of containerization and serverless computing introduces additional layers of complexity to cloud forensics.

The following are key container and serverles considerations:

- **Containers:** Containers are naturally ephemeral, often existing for only short periods. This transience poses significant challenges for forensic data collection and analysis. Forensic strategies must include capturing container logs and snapshots of container states to provide insights into the activities and data container processes. Because of that, it is highly recommended that you redirect container logs, VM logs, and every service log to external log storage. The advantage of this approach is to make possible the integration of these logs into a SIEM/SOAR tool to improve threat detection/response and the possibility of executing forensic analysis due to the brief life cycle of the containers and other cloud services. Note that sometimes it's possible to pause and dump the state of a running container for forensic purposes.
- **Serverless computing:** Serverless architectures further abstract the execution environment from the user, with CSPs managing the underlying infrastructure. Forensic analysis in serverless environments relies heavily on the logs generated by serverless functions, including execution logs, access logs, and application logs. Understanding the invocation and execution patterns of serverless functions is crucial for reconstructing events during an incident. It is also important to have robust monitoring in place to potentially catch an attack in progress and to provide detailed logs for forensics.

11.4 Containment, Eradication & Recovery

Of all the IR activities, those in the Containment, Eradication, and Recovery phases are most deeply affected by the specific technical and architectural models used in cloud deployments. Immutable Infrastructure as Code (IaC), autoscaling, microservices, identity federation, and the underlying technologies profoundly impact the processes and techniques available for these activities. In many cases this brings significant advantages compared to responses in traditional data centers.

11.4.1 Containment

It will be important to engage the cloud and application owner, when possible, to assist in determining the proper containment plan. They may also be the best to implement the plan since they know their deployment environment better than any central IR team can. IAM and management plane containment should be the top priorities in any security incident. Proper IAM containment can be very difficult due to the separation of authentication (AuthN) and authorization (AuthZ Gap) mechanisms. Cloud applications rely on federated identity, where authentication and authorization are separated and often performed on different platforms. Typically the IdP will issue a session token to an authenticated user, and this token is used by the relying party (authorization). Relying parties will usually keep accepting a token until the end of the Time to Live (TTL), or the end of the session. In many systems, even if a user/entity is blocked or removed from the system, they will still hold a session token. The relying party may still accept that token until the end of the session based on the TTL since it hasn't expired and there hasn't been any action that would trigger another validation of the token. However, most authorization systems will check entitlements on every request. As a result, containment may require different actions on both the IdP and within the relying party, where the relying party will need to alter entitlements (such as using a deny policy) or add conditions (such as only accepting tokens issued after a particular time). Zero Trust architecture recommends checking entitlements continuously (every 10 minutes).

A second complication is when service account credentials are compromised and abused. Containing these without coordinating with the cloud or application owners could break the required application functionality. Instead of adding deny policies and modifying entitlements, the responder may need to insert conditions like source IP address restrictions. This type of Attribute-Based Access Control (ABAC) is not universally supported by all cloud providers, especially SaaS providers. It is critical for analysis to determine the source of the abused credential so the attacker can't merely re-abuse their access and establish a new session. IAM containment also requires a good understanding of whether the attacker was able to use their access to escalate or pivot into different identities, just as we track attackers pivoting around a network. This may require tight coordination between an analyst and a responder if those are separate roles.

Management plan containment doesn't only include IAM containment. It's also important to look at interconnected services and identify the blast radius of affected services and resources. While this is often clearly indicated by reviewing IAM permissions, in some cloud platforms there may be internal connections allowed between services that only show up directly in service and resource configurations.

Network containment is often easier in cloud networks since they rely on Software Defined Networking. Rules can be changed very quickly and easily using API calls and web consoles. However, the responder must understand the networking specifics of the platform. For example, as with IAM changing a network security group rule in some providers may or may not break an active networking session (since the logic is evaluated on connection until the end of the session).

The use of autoscaling for ephemeral resources (such as FaaS, Serverless, etc.) can enhance containment. The responder or application owner can modify the launch requirements for the autoscaling group to use a patched version of the workload, then split off the compromised resource and isolate it for further analysis. Containment activities should also prioritize any resources that have been made public or were shared with an unknown destination (e.g., an unknown cloud account, subscription, or project in the same provider). For critical data, containment may need to risk breaking application functionality temporarily, and incident responders should have a timely escalation path to the authority and capability to make this decision and take action in highly critical situations. Compromised resources, such as a VM, container, or serverless code, can allow an attacker to pivot into the management plane. Analysis should have determined this IAM Blast Radius, which should feed into containment priorities. Pay particular attention to any involved CI/CD pipeline. Containing an attacker's access to a pipeline is also a top priority since those usually have full administrative capabilities to affect deployments.

11.4.2 Eradication

Eradication is usually best implemented by the cloud and application owner, and their administrators and developers. As with analysis and containment, the primary focus of eradication should be to remove the attacker from the management plane, but now on a permanent basis. This could be through credential rotation, adding additional policy conditions, adding multi factor authentication (MFA) or digital certificates, and similar techniques. This is only possible when the origin of the incident is determined so the source IAM/access can be locked down. An analysis will need to be performed during and after an incident, not just on detection, to identify if the attacker was able to pivot within the management plane or IAM system.

In the cloud, it's often easier to replace a resource than try to expel an attacker from it. This is especially true if the resource is ephemeral thanks to autoscaling or Infrastructure as Code (IaC). Don't try to fix it; just wipe and replace it when possible. This type of eradication is far easier when used with cloud-native applications and is more difficult when applications are deployed with a lift and shift approach.

Eradication will often require deleting old versions of images, serverless code, and IaC. Attackers may use these to re-compromise a deployment, especially if an employee accidentally re-deploys an old version in whole or in part. Eradication may also require a full review of CI/CD pipelines and any materials stored in version control and artifact repositories.

11.4.3 Recovery

IaC, autoscaling, and other automation are incredibly powerful for incident recovery. They can rapidly deploy hardened versions of applications and infrastructure or even deploy a clean version into an entirely new environment. All images, resources, and templates used in recovery should be analyzed to ensure that the root cause was eliminated and that the attacker didn't leave any lingering backdoors. For example, an attacker may have given themselves access to an IAM entity that seems unrelated to the incident and wasn't used in the primary attack. Or the attacker could have embedded backdoor keys or code into applications or images to allow future access.

11.5 Post-Incident Analysis

One of the most important, yet often overlooked, stages of IR is determining the lessons learned and then taking active steps to reduce the likelihood or impact of future, similar events. This is done in the Post-Incident Analysis phase. In this phase the responders determine the root cause of the incident, analyze the response process, and try to identify areas of improvement. This is less about assigning blame, and more about trying to identify any structural issues that can be modified to prevent, or limit, future events.

The fundamentals of the Post-Incident Analysis phase aren't different for cloud, but there are a few best practices to highlight.

- Since many cloud incidents involve working with the team that managed the cloud deployment, they should be included in any post-incident analysis.
- Responders should be required to create a new runbook/playbook for any new incident type they encounter.
- Many cloud security incidents are the result of misconfigurations. Rather than fixing the blame, the Cloud Security Alliance recommends following a Just Culture[179] approach which focuses on the identification of any systemic failures before blaming individuals, while still holding individuals responsible for their actions. For example, if over-privileged IAM was the source of the breach, the organization may consider adding tools to identify potential IAM issues, security may provide common baselines and work with teams to review permissions, or the organization can move from

[179] Just Culture is a concept that is related to systems thinking. The concept emphasizes that mistakes are typically a fault of the organizational culture rather than a fault of a person. The idea is to shift from "who did it" to "what went wrong".

static credentials to Just in Time entitlements combined with strong authentication, but using frictionless tooling that doesn't slow down developers.

11.6 Resilience

In the realm of cloud computing, resiliency refers to the ability of an application or system to continue operating seamlessly in the face of various types of disruptions, ranging from minor faults to major outages. The concept of cloud resiliency is layered and can be scaled according to the criticality and budgetary constraints of the services in question.

At the base level, *single-region resiliency* is where most applications begin their journey towards being resilient. In this setup, the application is hosted within a single cloud provider's region, and it employs strategies like autoscaling and load balancing to handle sudden spikes in traffic and to be fault-tolerant against individual component failures. Backup and recovery strategies are also put in place to protect data. This foundational level is also the most cost-effective option since it takes advantage of the cloud provider's existing infrastructure and services without the need for significant duplication of resources.

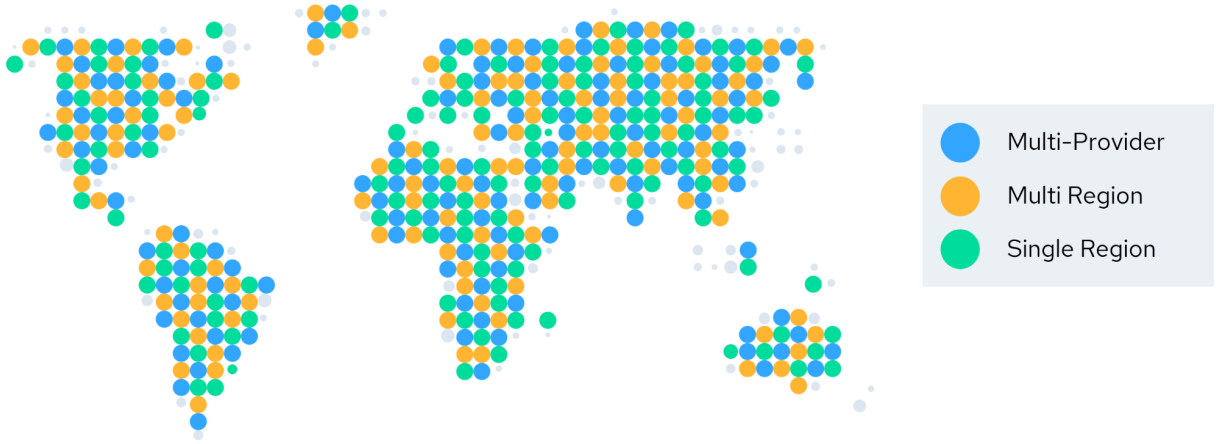


Figure 66: Global Cloud Resiliency Strategies

However, single-region deployment is vulnerable to regional outages which, although rare, can have a significant impact on the availability of the application. To mitigate this risk, organizations can step up to *multi-region resiliency*. This involves running parallel deployments of the application across multiple regions within the same cloud provider's network. While this significantly improves fault tolerance and geographic diversity, it also introduces additional costs. These costs come not just from running multiple instances of the application but also from the need to synchronize data across regions. Furthermore, data transfer charges incurred for moving data between regions can quickly become substantial, making this a more expensive option than single-region deployment.

The hardest cloud resiliency is *multi-provider resiliency*. This level is achieved by spreading the application's footprint across multiple cloud providers. The intention is to safeguard the application from a scenario in which an entire cloud provider goes down. Achieving multi-provider resiliency is complex due to the technological differences between cloud providers. Containerization technology can ease some of this complexity by abstracting the application from the underlying infrastructure, but challenges remain.

These include managing disparate networking, storage, and security models, as well as orchestrating deployment and operations across environments that are fundamentally different. Costs can escalate rapidly, not just in terms of direct operational expenses but also due to the increased overhead required for design, development, testing, and ongoing maintenance. Despite the costs and complexities, for critical applications that demand the utmost availability—such as those involved in financial transactions, health services, or global commerce—multi-provider resiliency can be a necessary investment.

11.6.1 IaaS/PaaS Resiliency Tools

IaaS and PaaS rely at their core on abstraction (virtualization) and orchestration. These additional layers over the raw hardware introduce more opportunities for failure. To account for this, CSPs offer their customers multiple options for architecting for better resiliency from single failures. IaaS and PaaS include a number of tools that can be leveraged to improve resiliency.

- **Architecture:**
    - *Autoscaling:* Leveraging auto scaling capabilities allows systems to dynamically adjust resources and replace them on failure.
    - *Serverless computing:* Serverless is highly fault tolerant since it is designed inherently to scale based on demand.
    - *Platform as a Service (PaaS):* PaaS offerings abstract from having to maintain operating systems and infrastructure, and many have very high resiliency SLAs.
- **Infrastructure as Code (IaC):**
    - *Image definitions:* Defining VMs and containers using IaC for images improves the ability to generate replacements and rapidly adapt.
    - *Infrastructure definitions:* IaC can provide portability to entire application stacks.
- **Automation and Backups:**
    - *CI/CD pipelines:* Rapidly automate the deployment of remediations or even new stacks into new environments.
    - *Backups:* Many providers also support automated backups, particularly of their PaaS services like databases.
- **Chaos Engineering:**
    - *Principles and tools:* used to inject deliberate faults into development and production applications to continuously validate resiliency. This guides teams to building with the assumption that there will be infrastructure and service failures, as opposed to assuming little or no downtime.

11.6.2 Resiliency for SaaS

When discussing the concept of resiliency for Software as a Service (SaaS) applications, it is essentially referring to the ability of the service to continue operating in the face of various disruptions. Resiliency in this context is about ensuring business continuity and disaster recovery (BCP/DR) plans, even when there are outages or other issues with the SaaS provider. Unlike with IaaS and PaaS, there are typically few or no options for a customer to manage aspects of their own resiliency when using the service.

The following are some challenges with SaaS:

- **Extremely limited options:** SaaS applications often run on the provider's infrastructure, which means that the control over the resiliency and redundancy of the applications is largely in the hands of the provider, not the end user. The options for enhancing resiliency are limited by what the provider offers. Therefore, it's important for businesses to choose SaaS providers that offer robust disaster recovery and high availability features.
- **Data extraction/migration support:** While some major platforms do support data extraction and migration, these features are typically designed for switching platforms or making backups, rather than for real-time disaster recovery. The process is not continuous and may involve significant delays between data exports. In the event of an outage, the latest data may not be available, which can be problematic for operations that require up-to-the-minute data.
- **Periodic data extracts:** In many cases, the best option available to businesses is to perform periodic data extracts. This includes local data synchronization where feasible. Although this does not offer real-time recovery, it can mitigate the risk of data loss. How often these extracts should be performed depends on the nature of the business and the criticality of the data. For some, nightly backups may suffice, while others may require more frequent intervals.
- **Examine and know your SaaS SLAs:** SLAs are critical documents that define the level of service you can expect from a SaaS provider, including uptime guarantees and the provider's responsibilities in the event of service disruption. It is crucial to thoroughly examine these agreements to understand what continuity and recovery options are promised, how the provider handles data backups, and what compensation is provided if the service fails to meet the agreed standards.

In addition to the points above, businesses should consider the following strategies for ensuring continuity with SaaS applications:

- **Multiple providers:** Depending on the criticality of the services, it may be worthwhile to use multiple SaaS providers for redundancy. This is particularly true for services that are critical to business operations.
- **Hybrid solutions:** Some businesses might opt for hybrid solutions where essential applications are hosted on-premises or on a private cloud, while less critical applications are hosted with SaaS providers.

- **Regularly updated recovery plan:** Companies should have a well-documented and regularly tested recovery plan. This plan should be updated to account for changes in the SaaS applications and the business operations they support.
- **Insurance:** Some companies may also consider insurance options to cover losses due to SaaS downtime, although this is a financial buffer rather than a continuity solution.
- **Training and preparedness:** Ensure that staff are trained on procedures to switch to backup systems or manual processes where possible during an outage.

Resiliency planning for SaaS applications requires an understanding of the limitations and active planning around them to ensure that business operations can continue with minimal disruption.

Summary

Organizations should develop a solid understanding of the CIR process—and its IR capabilities—to prepare for any potential incidents.

This domain explores the CIR framework and the preparation required to respond to incidents effectively. It serves as a go-to guide for a CSC to prepare for and manage cloud incidents through the entire lifecycle of a disruptive event. It also provides a transparent, common framework for CSPs and CSCs to share CIR practices.

Building a Strong Foundation: The CIR framework equips organizations to handle security breaches in the cloud. The first phase, *Preparation*, focuses on building a strong foundation. This involves establishing a dedicated Cloud Incident Response Team (CIRT) to manage incidents. The CIRT then develops comprehensive strategies, procedures, and a communication plan to guide the response. Additionally, technical preparations are required to account for the differences in cloud, especially security telemetry and responder access. This includes implementing security tools for early detection and ensuring forensics and analysis capabilities are in place for in-depth investigations.

Responding and Learning: When a security breach occurs, the Cloud Incident Response framework transitions to the *Detection and Analysis* phase. Here, the focus is on identifying the incident early and understanding its root cause. Multiple detection methods are employed to achieve this, and the framework emphasizes the importance of swift notification and resolution based on potential business impacts. Once the threat is contained, the *Containment, Eradication, and Recovery* phase comes into play. This stage involves choosing the right strategy to stop the attacker and prevent further damage while investigations and forensics are conducted.

Continuous Improvement: The Cloud Incident Response framework concludes with the *Post-Incident Analysis* phase. This stage is vital for learning from the experience. The CIRT analyzes the incident to identify weaknesses in personnel, processes, or technology. These lessons learned are fed into the Preparation phase to continuously improve the organization's incident-handling capabilities. This cyclical approach ensures a constantly evolving security posture, allowing organizations to navigate the ever-changing threat landscape in the cloud effectively.

Coordination and information sharing: Effective communication is imperative due to the shared nature of cloud security incidents. This includes establishing clear channels between CSPs and users, facilitating regular updates for impacted users, and fostering information sharing among stakeholders. Furthermore, early planning for communication with internal IR teams, law enforcement, and key partners strengthens overall CIR capabilities.

Recommendations

Incident Response Planning

- Develop a comprehensive incident response (IR) plan tailored for cloud environments.
- Establish a dedicated Cloud Incident Response Team (CIRT) with defined roles and responsibilities.
- Ensure responders have access to required environments and tools, including incident analysis services, hardware, and software.
- Maintain internal documentation such as port lists, asset lists, and network traffic baselines.

Preparation

- Perform proactive scanning, monitoring, vulnerability, and risk assessments.
- Subscribe to third-party threat intelligence services.
- Evaluate cloud service providers' capabilities to support IR.
- Regularly audit logs, snapshots, forensic capabilities, and e-discovery features.
- Conduct regular backup restoration and disaster recovery tests.

Detection & Analysis

- Focus on management plane and IAM activity as primary areas for threat detection.
- Implement activity-focused threat detectors rather than threat-actor-focused ones.
- Use configuration changes as sources for cloud detectors and integrate these into CSPM tools.
- Ensure clear and direct communications with cloud account teams for rapid incident validation.
- Employ canaries and honey tokens to detect compromised identity repositories and trigger immediate investigations.
- Leverage automation and machine learning to manage the dynamic nature of cloud environments.
- Use snapshots to preserve the state of VMs for forensic analysis.

Containment, Eradication & Recovery

- Prioritize IAM and management plane containment.
- Isolate identities and workloads, taking systems or services offline as needed.
- Replace compromised resources using autoscaling and Infrastructure as Code (IaC) where possible.
- Rotate credentials, add policy conditions, and implement multi-factor authentication (MFA) for eradication.
- Delete old versions of images, serverless code, and IaC to prevent re-compromise.

Post-Incident Analysis

- Follow a Just Culture approach to identify systemic failures without assigning blame.

Resilience Planning

- Start with single-region resiliency using autoscaling, load balancing, and backup strategies.
- Consider multi-region resiliency to improve fault tolerance and geographic diversity.
- Evaluate multi-provider resiliency for critical applications requiring the highest availability.
- Leverage IaaS and PaaS tools such as autoscaling, serverless computing, and Infrastructure as Code (IaC) for improved resiliency.
- Use CI/CD pipelines to automate the deployment of remediations and new stacks.
- Implement Chaos Engineering principles to validate resiliency through deliberate fault injection.

Additional Guidance

- Cloud Incident Response Framework | CSA
- Cloud Incident Response Framework – A Quick Guide | CSA
- CSA Medical Device Incident Response Playbook | CSA
- Cloud Penetration Testing Playbook | CSA
- Cloud Penetration Testing Guidance | CSA

Domain 12: Related Technologies & Strategies

Introduction

With cloud security, we must consider varying angles for analysis to understand the challenges with it. *Lenses*, which are essentially perspectives, provide us with unique ways to help us examine issues from different viewpoints, so we can take strategic considerations into account. *Processes* provide methodologies and frameworks that guide us in making informed decisions and taking necessary actions in a repeatable manner. By combining these two, we have a comprehensive strategy to ensure the security and compliance of cloud applications, systems, and data.

We explore a spectrum of lenses and processes that traverse various critical security domains, such as organization management, identity and access management (IAM), security monitoring, network, workload, application, and data. Lenses and processes are important security areas that span multiple domains.

Learning Objectives

The learning objectives for this domain aim to provide readers with knowledge on:

- Discuss the benefits of integrating AI into threat and vulnerability management for cloud security.
- Explain the role of Artificial Intelligence in cloud security.
- Identify the key components of the Zero Trust cybersecurity approach.

12.1 Zero Trust

Zero Trust (ZT) is a cybersecurity approach[180] focused on protecting resources-users, assets, and data beyond network perimeters. ZT moves beyond trusted or untrusted users and networks, and relies on continuous multi factor authentication (CMFA), micro-segmentation, encryption, endpoint security, automation and analytics. Additionally, ZT addresses enhanced auditing for data, applications, assets, and services (DAAS). The primary objective of Zero Trust Architecture (ZTA) is to mitigate security risks inherent in the assumption of trust, or inadequate access controls. Common strategies for mitigating these risks include minimizing the attack surface and enhancing the efficiency and granularity of security

[180] Zero Trust is also covered in *Domain 2: Cloud Governance & Strategies*, *Domain 7: Infrastructure and Networking*, and the CSA CCZT Training.

measures. These priorities are particularly important in the cloud, due to the need to manage multi-tenancy, highly distributed access, and a broad, Internet-facing attack surface.

ZTA provides a holistic and consistent security approach that safeguards an enterprise from internal and external threats and attacks, which may exploit inherent or introduced gaps in conventional protection methods and defense-in-depth controls.

The key differentiator in ZTA is the ephemeral nature of access granted to requesting parties to resources, data, and computing workloads. This differentiator, combined with capabilities like dynamic policy enforcement and dynamic policy decisions, strengthens enterprise environments spanning cloud and on-premises segments. This is true for both internal and external threats that exploit exposed access mechanisms.
A ZT approach serves both technical and business objectives. Technically, it provides a framework for protecting resources, streamlines the user experience, minimizes attack surface and complexity, enforces least privilege, enhances control and resilience, and reduces blast radius. From a business perspective, ZT helps mitigate risk, improve compliance, and align the organization's culture with its leadership's risk appetite[181] and governance framework[182].

12.1.1 Technical Objectives of Zero Trust

The technical objectives of ZT can all be used to improve cloud security. The following are examples of some of the technical objectives and how it relates to ZT.

Protective Framework

ZT establishes a protective framework and a novel approach to cybersecurity. ZT's core assumption is that an organization should not inherently trust any principle within or beyond its boundaries. The new protective framework allows a focus shift to more business-oriented goals, with systems designed around the value of the data and specific protection needs. Many formerly successful security procedures and strategies are no longer fully effective. As a result, an organization's investments in older cybersecurity techniques and technology are increasingly yielding limited results and inadequate protection.

Relying on approaches and frameworks anchored in physical objects or code signatures is no longer feasible. With the rise in frequency and scale of attacks and the interconnected nature of today's world, businesses must reassess everything from network configuration to detection and prevention methods.

Simplified User Experience

ZTA streamlines the user experience by implementing a uniform access model across the entire environment, including both network and other components. Every access request, whether explicit or implicit, is presented with the same logic which includes various determinations, such as: Who are you? Exactly which data do you need? Do you need this access now? Once approved, the user is granted access to a specific resource for a specified period of time.

[181] NIST. (2024) Computer Security Resource Center - Risk Appetite is defined in the glossary with links to the standards that provide more detail on risk appetite in cybersecurity.
[182] NIST. (2020) *NISTIR 8286: Integrating Cybersecurity and Enterprise Risk Management (ERM).*

In ZTA mode, there is no:

- Complicated diagram of nested groups with potential legacy Access Control Lists (ACL) that allow or deny, producing unexpected results.
- Layers of groups managed by potentially no-longer-relevant decision makers.
- Orphaned groups whose owners have moved on, or unpredictable authorization mechanisms, such as local versus global.
- Delay in either provisioning, deprovisioning, or revocation of access. Every access request is handled by policy decision points (PDPs) consistently and just in time.

Reduced Attack Surface

ZTA implements strict access controls, continuous authentication, and least privilege principles across the entire network and infrastructure. This involves assuming that threats may already exist within the network and adopting a "never trust, always verify" approach to access and permissions. By continuously verifying the identity and security posture of users, devices, and applications, Zero Trust aims to prevent lateral movement by attackers and limit the potential impact of security breaches.

Reduce Complexity

As mentioned in this document's introduction, an organization's ever-expanding digital footprint potentially complicates its IT environment. Especially with actual access decisions made months or years before it is requested, used, or even necessary. Those decision makers often move on organically as time progresses, leaving orphaned objects, over whose access, no one presides. Such complexity represents one of the biggest security challenges for an organization. This often generates additional reduced visibility, complex configurations, weaknesses and vulnerabilities, making it easier for attackers to exploit them. The ZT approach reduces this complexity.

Examples of such complexity are:

- Hybrid integration of cloud and on-premises environments
- Multi-cloud architecture
- Edge computing

Each one of these examples, from the standpoint of access control policies, brings extreme complexity. In a ZT environment, it is assumed that all application access is potentially malicious or undesirable. Thus, instead of trying to police all the borders and paths across the organization’s network, islands of applications and data are created. These islands can be protected in a much more focused way. This is because far more attributes are required to establish ZT strategies than standard security mechanisms. As organizations simplify networks and consolidate data centers in favor of agility, ZT provides an enhanced security mechanism to reduce complexity in any security architecture by creating perimeters around applications and identity. This also means tighter control of what each user identity can do and stricter visibility about an individual’s access rights and privileges. Especially those of third parties and suppliers.

Enforce Principle of Least Privilege

This principle is that users and programs should only have the necessary privileges to complete their tasks. Essentially, a user gets access to exactly what they need to conduct their business, when they need it. Simplified access provisioning makes it much easier for the security operations and governance teams to manage the continuously changing security landscape. It also enhances the end-user experience by delivering the right services at the right time.

To further enhance security posture, organizations should consider implementing additional measures, such as User and Entity Behavior Analytics (UEBA), Privileged Access Management (PAM), and Identity Access Governance. These practices provide insights into user behavior, manage privileged access, and ensure governance over identity-related access controls, thereby reinforcing the principle of least privilege within the ZTA framework.

Improved Security Posture and Resilience

From outside the organization, ZTA ensures reduced hacker visibility of the IT infrastructure and individual assets, reducing the attacker surface.

From within the organization, ZTA emphasizes:

- Minimal lateral movement
- Limiting opportunities for cross-network and cross-system attacks
- Reducing exposure from any malicious actor that manages to get inside any segment

The external user is contained and controlled within a small area of the network, ensuring resilience of the entire IT infrastructure. Therefore, the attack occurrence can be easily contained and addressed on a small blast radius and quickly return to an earlier state. The reduced attack surface ensures that all source scanning and mapping initiated by internal or external users are unsuccessful unless the user is authorized within the ZT implementations. The two-layer architecture, where the control plane and the data plane are separated, helps ensure that the user is allowed inside the organization's network only after the proper authentication and authorization of the user and their respective devices.

Improved Incident Containment and Management

Making the organization's incident management process more effective and efficient is one of the main goals of a ZTA. The key drivers to achieve this objective are embedded in the core and design principles behind a ZTA. One is the assumption that no entity can be trusted unless otherwise proven. Another is the assumption that a breach could be ongoing and that any entity's behavior in the system must be monitored continuously. The micro-segmentation and continuous authorization for network access reduces the blast radius of a potential breach as it allows better control over the attacker's lateral movement. When a breach occurs, the organization can limit the event's impact through more effective containment and easier eradication and remediation given the limited scope of the incident. In addition, the continuous monitoring capabilities included in ZTA allow for more effective identification of anomalies and incidents. The incident-related data is also used to update the PDP. This allows a dynamic policy definition and its enforcement. These precautions further limit the spread of an incident across the organization's network.

12.1.2 Zero Trust Business Objectives

Similar to technical objectives, Zero Trust business objectives can also help enhance an organization's security posture, help protect sensitive data, and demonstrate the actual business value of ZT. The following are examples of some of the business objectives and how it relates to ZT.

Reducing Risk

- Zero Trust helps reduce organizations' overall cybersecurity risk. By adopting this approach, businesses implement a proactive security model that assumes threats can exist inside and outside the network perimeter.
- Traditional security models rely on perimeter-based defenses, assuming everything inside the network is trusted. However, with the increasing sophistication of cyber threats and the rise of remote work and cloud computing, this perimeter-centric approach is no longer sufficient.
- Zero Trust mitigates risk by enforcing strict access controls, continuous authentication, and least privilege principles. This means that even if a threat actor gains access to the network, their movement and access to sensitive resources are restricted, limiting the potential impact of a security breach.

Improving Compliance

- Compliance with regulatory requirements and industry standards, is crucial for businesses, especially those operating in highly regulated finance, healthcare, and government sectors.
- ZT helps organizations improve compliance by providing granular control over access to sensitive data and resources. By implementing ZT principles, businesses can demonstrate to regulators and auditors that they have taken proactive measures to protect their data and systems.
- Regulations like GDPR and HIPAA, require organizations to implement strong access controls and data protection measures. ZT aligns with these requirements by enforcing strict authentication, encryption, and access policies.

Aligning with Organizational Culture and Leadership's Risk Appetite

- Adopting Zero Trust requires buy-in from all levels of the organization, from frontline employees to top executives. It fosters a culture of security awareness, accountability, and continuous improvement.
- ZT aligns with the leadership's risk appetite by providing a proactive and adaptive security approach that prioritizes risk reduction and resilience. It emphasizes the importance of continuously assessing and mitigating risks, rather than relying solely on reactive security measures.
- By embracing ZT principles, organizations demonstrate their commitment to cybersecurity and resilience, which can enhance trust and confidence among customers, partners, and stakeholders.

12.1.3 Zero Trust Pillars & Maturity Model

ZT security principles are grouped into pillars that broadly align with our control domains as depicted in the figure below. They are designed to work together in concert to provide enhanced protections for key assets and resources. These pillars and their respective capabilities and functions are described in the US Cybersecurity and Infrastructure Security Agency (CISA) ZT Maturity Model (ZTMM)[183] and DoD ZT Reference Architecture[184] reference documents. While their depictions of the pillars differ somethat, their models are basically equivalent and fundamentally consistent.

The capabilities of the ZT security strategy and framework can be used in conjunction with the cloud shared security responsibility model (SSRM) to secure cloud deployments, leveraging CSP-provided infrastructure security and services. As an enterprise security strategy, ZT is also applicable to securing multi-cloud and hybrid environments.

Below are the pillars and cross-cutting capabilities in the CISA ZTMM.

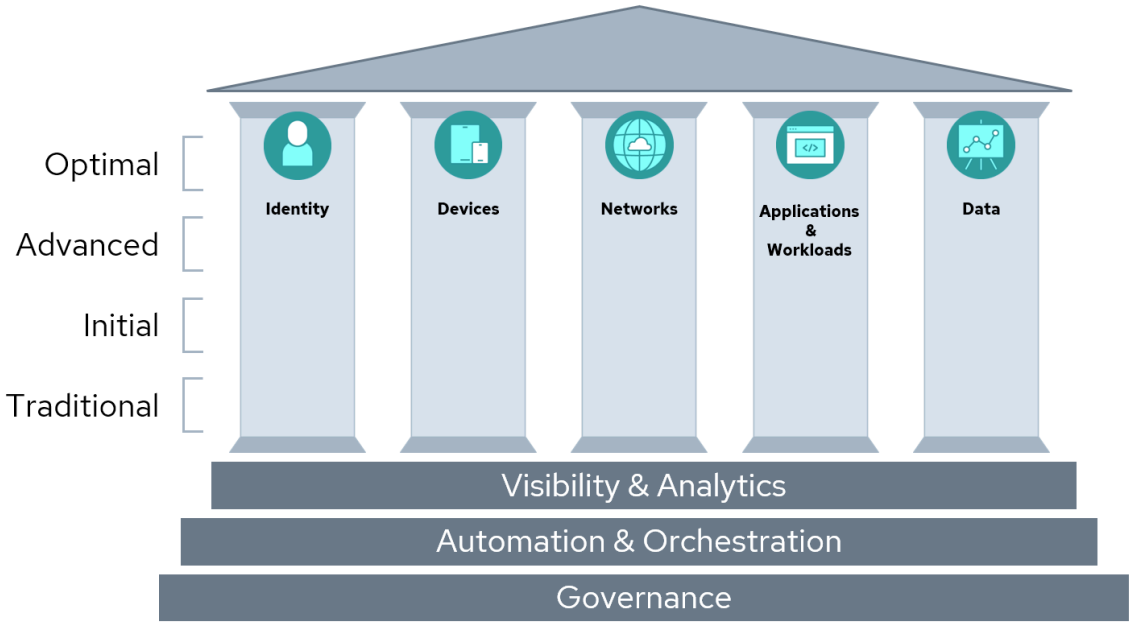


Figure 67: The CISA Zero Trust Maturity Model

- **Identities (also known as Users in some models):** Securing, limiting, and enforcing access for person, non-person, and federated entities' to data, applications, assets and services (DAAS), encompasses the use of identity, credential, and access management capabilities, such as multi factor authentication (MFA) and CMFA. Organizations need the ability to continuously authenticate, authorize, and monitor activity patterns to govern users' accesses and privileges while protecting and securing all interactions. Role-based access control (RBAC) and attribute-based access control (ABAC) will apply to policies within this pillar to authorize users to access applications and data based on dynamic, context-based access policies.

[183] CISA. (2023) Zero Trust Maturity Model
[184] DOD. (2022) Department of Defence (DOD) - Zero Trust Reference Architecture

- **Devices:** In a ZT approach, it is crucial to possess capabilities for identifying, authenticating, authorizing, inventorying, isolating, securing, remediating, and controlling all devices. Real-time assurance validation and patching of enterprise devices are critical functions in this regard. Some solutions, such as mobile device managers or compliance-to-connect programs, offer valuable data for validating device security hygiene. For every access request, proper validation (e.g., examination of compromise state, anomaly detection, software versions and patch status, protection status, encryption enablement, etc.) should be conducted.

- **Networks:** In a ZT approach, organizations should both logically segment (virtually) and physically separate, isolate, and control the network environments (on-premises and cloud/off-premises). This involves implementing granular access controls and policy restrictions. As the perimeter becomes more refined through macro segmentation, enabling micro-segmentation enhances protection and control over DAAS elements. It is important to:

    - Control privileged access
    - Manage internal and external data flows
    - Prevent lateral movement

- **Applications and workloads**: Applications and workloads should include tasks on systems or services on-premises and applications or services running in a cloud environment. ZT workloads should span the complete application stack from the application layer to the hypervisor. Securing and properly managing the application layer, compute containers, and virtual machines should be central to ZT adoption. It is crucial to implement robust processes for code review, vulnerability scanning, and security testing throughout the software development lifecycle to mitigate risks and prevent security breaches. Internal source-code and common libraries should be vetted through DevSecOps development practices to ensure the security of applications from inception.

- **Data:** ZTA safeguards critical DAAS. A clear understanding of an organization's DAAS is critical for successfully implementing a ZTA. Organizations need to categorize their DAAS in mission criticality and use this information to develop a comprehensive data management strategy for their overall ZT approach. This can be achieved by categorizing data, developing schemas, and encrypting data at rest and in transit. Solutions like data rights management (DRM), data loss prevention (DLP), software-defined networking (SDN), and granular data tagging are key in protecting critical data.

- **Visibility and analytics** (a cross-cutting capability in the CISA model)**:** Vital, contextual details, must be included to provide a greater understanding of performance behavior, and activity baseline across the various ZT pillars. This visibility improves anomaly detection and enables dynamic changes to security policy and real-time contextual access decisions. Additionally, other monitoring systems, such as sensor data and telemetry, are used to help fill out the picture of what is happening with the environment. This will aid in the triggering of alerts used for responses. A ZT enterprise will capture and inspect traffic, looking beyond network telemetry and into the packets to observe threats and appropriately orient defenses.

- **Automation and orchestration** (a cross-cutting capability in the CISA model)**:** Automate manual security processes to take policy-based actions across the enterprise quickly and at scale. Security orchestration, automation, and response (SOAR) improves security and reduces response times. Security orchestration integrates security information and event management (SIEM) and other automated security tools to assist in managing disparate security systems.

Automated security response requires defined processes and consistent security policy enforcement across all ZT enterprises for proactive command and control.

- **Governance** (a cross-cutting capability in the CISA model)**:** Governance is an important function because it ensures that business strategy, risk, and IT perspectives are aligned with one another. Governance helps define ZTA policies, such as access and process data.From a non-technical perspective, governance should also manage and reduce complexity. To succeed in complexity reduction, the focus should be on the protected surface. From a technical point of view, governance policies should be enforced by the Policy Enforcement Point (PEP).

The CISA ZTMM helps organizations enhance their ZT strategies as it outlines maturity stages–Traditional, Initial, Advanced, Optimal–across the ZT pillars (Identity, Devices, Networks, Applications and Workloads, and Data) and capabilities (visibility, automation, governance). These maturity stages help organizations assess, plan, and implement the necessary measures to progress toward a more secure ZTA. The CISA ZTMM journey, depicted in the accompanying figure, represents a path towards achieving optimal ZT maturity. This practical visual representation shows how companies advance through ZT's various maturity levels.

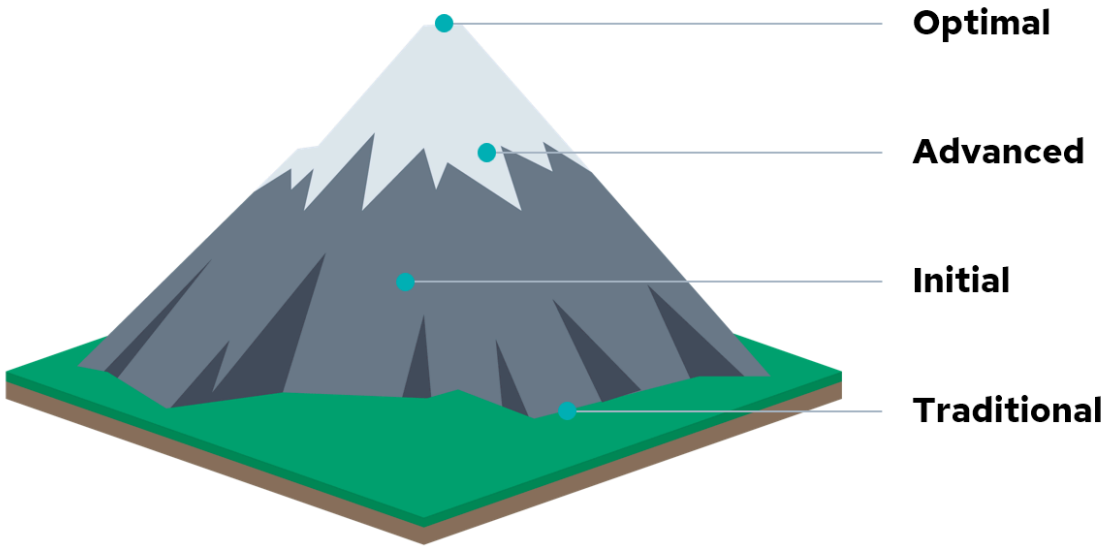


Figure 68: ZT Maturity Journey

To utilize the CISA ZTMM effectively, grasp the framework, refine your functions and assess your current ZT maturity. Finally, plan steps for maturity advancement and align them with organizational projects and priorities, using a prioritization model to guide you.

By understanding the characteristics and objectives associated with each maturity stage, organizations can assess their current state, identify areas for improvement, and develop a roadmap for advancing through the Zero Trust maturity journey. The table below shows the characteristics of each maturity stage within the Zero Trust Maturity Model.

Maturity Stage	Description	Characteristics
Traditional	The Traditional maturity stage represents the starting point for organizations on their Zero-Trust journey. At this stage, security practices are typically perimeter-focused, with implicit trust in internal network traffic.	• Security controls are primarily based on perimeter defenses, such as firewalls and intrusion detection systems (IDS). • Access to resources is often granted, based on network location rather than user identity or device posture. • Security policies tend to be static and reactive, with limited visibility into user activities and network traffic.
Initial	Organizations adopt foundational Zero Trust principles and technologies in the initial maturity stage to enhance their security posture.	• Organizations start centralizing identity management processes and implementing basic authentication controls, such as password policies and MFA for critical systems. • Device security measures, such as endpoint protection software and device encryption, may be introduced to improve the security posture of endpoints. • Network segmentation efforts may begin, to reduce the attack surface and limit lateral movement within the network environment.
Advanced	At the Advanced maturity stage, organizations have significantly progressed in implementing Zero Trust practices and technologies across multiple domains.	• Identity access management becomes more centralized and automated, with strong authentication mechanisms, such as adaptive and continuous authentication deployed. • Device security measures are enhanced, with continuous monitoring of device health and compliance status and automated remediation of security vulnerabilities. • Network segmentation efforts are expanded, with dynamic access controls based on user context and application sensitivity, and encrypted communication channels widely deployed.
Optimal	The Optimal maturity stage represents the highest level of Zero Trust maturity, where organizations have fully integrated Zero Trust principles into their security strategy and operations.	• Identity governance processes are fully automated, with self-service capabilities for user onboarding, offboarding, access requests, and advanced analytics for detecting and mitigating identity-related threats. • Device security measures are tightly integrated into the Zero Trust architecture, which includes comprehensive endpoint detection and response (EDR) capabilities and advanced threat intelligence for proactive threat hunting. • Network segmentation is granular and adaptive, with automated threat detection and response mechanisms integrated into the network fabric, and Zero Trust access controls enforced at the application level.

Table 10: Zero Trust Maturity Stages

12.1.4 ZT Design & Implementation Steps

When considering the tactical aspects of adopting ZT, there are four key design principles for building a resilient architecture, and a repeatable, five-step implementation process for interactive and incremental execution to secure organizational assets with risk-based prioritization.

Alongside these, the CISA ZTMM which is not part of the five steps, but important in understanding the incremental progression of ZT in the organization's Zero Trust implementation journey.

The ZT Design Principles include:

- **Focus on Business Outcomes**: Understanding how ZT aligns with and supports the organization's primary business goals.
- **Design from the Inside Out**: Developing a security strategy that starts within the organization before extending outwards.
- **Determine Who or What Needs Access**: Identifying which users and devices require access to specific resources.
- **Inspect and log key traffic**: Aim to monitor and record critical activity for potential threats as a targeted approach.

The five steps for a repeatable ZT implementation process are:

- **Step 1: Define the Protect Surface**[185]: Identify and evaluate critical business information systems, including constituent data and resources (DAAS elements), classifying their business risk level, and assessing their current security maturity to help with implementation prioritization.
- **Step 2: Map the Transaction Flows**: Understand the movement of information within and outside the system and organization, and the potential classification of each flow. The goal is to understand where the most sensitive information and assets are and where you can have the most impact for controlling access, all as inputs to ZTA design and development.
- **Step 3: Build a Zero Trust Architecture (ZTA)**: Design and develop the infrastructure, capabilities, and controls necessary to implement ZT protections for key business systems and assets.
- **Step 4: Create ZT Policy**: Implement policy controls and establish guidelines and rules for network, system, and data access, and security for key business systems and assets.
- **Step 5: Monitor and Maintain the Network (Environment)**: Continuously oversee the ZT environment to ensure ongoing security and adapt to new threats.

These elements are vital in shaping and executing an effective ZT security strategy aligning with an organization's objectives and risks.

[185] CSA. (2024) *Defining the Zero Trust Protect Surface*.

12.1.5 Zero Trust & Cloud Security

The table below summarizes ZT principles, mapping them to security domains for how these can be applied to mitigate risks and enhance an organization's overall cybersecurity posture.

Security Domain	Zero Trust Principle
Organizational Management	ZT as an enterprise security and connectivity strategy, best implemented with a ZT culture
Identity and Access Management	Continuous, phishing-resistant MFA with context-based authorization of users, devices, and access requests
Security Monitoring	Monitor everything; presume breaches, detect suspicious activity early and dynamically adjust access
Network	Micro-segmentation, ZT Network Architecture & Software-Defined Perimeter
Workload	ZT device and workload security and integrity verification, malware and data exfiltration monitoring , with ZT workload access controls
Application	Fine-grained, least privilege access authorization with separation of duties; limit user permissions to the minimum required data and functionality
Data	Classify, protect, and monitor data at rest, in transit, and in use with strict ZT data access controls

Table 11: CCSK Security Domains and Corresponding Zero Trust Principles

Here's how some important tenets of ZT security impact key cloud security domains:

- **Identity and Access Management (IAM)**: Central to implementing Zero Trust, focusing on strong, continuous authentication and detailed authorization to ensure secure, context-aware access to resources.
- **Security Monitoring**: Enhance operational security monitoring and alerting by integrating Zero Trust principles, assuming potential breaches, and minimizing impact through strategic network access controls and rigorous incident response procedures.
- **Network**: Enforce Zero Trust access, use micro-segmentation to reduce attack surfaces, apply virtual firewalls and encryption for security, and follow Zero Trust principles, such as least privilege and continuous authentication, to control network access effectively.

- **Endpoint Security**: Enforce Zero Trust principles, protect against threats like malware and ransomware, and ensure rigorous authentication, authorization, and access control for devices accessing cloud resources.

To explore these thoroughly, we recommend studying ZT use cases, provided by experienced cybersecurity experts. You may wish to study CSA's use-case collection found in the Certificate of Competence in Zero Trust Training.

12.2 Artificial Intelligence

Artificial intelligence (AI) serves as both a cloud-hosted service and an emerging tool to bolster cloud security. While AI services are typically hosted in the cloud, it is important to note that on-premises hosting solutions are also available for certain applications. Additionally, AI presents a dual role in cloud security; it can be utilized to enhance cloud security measures, but it also poses a risk as an emerging attack tool. AI-powered algorithms have the capability to discover vulnerabilities, craft exploits, and execute sophisticated attacks, highlighting the importance of securing AI services and implementing robust security measures to defend against AI-driven threats.

12.2.1 Artificial Intelligence & Cloud Security

As with Zero Trust discussed earlier, AI intersects with multiple critical cloud security domains. Additionally, Zero Trust architectures today often rely on AI technologies for various tasks such as enforcing impossible travel policies and making context-aware access decisions. This intersection underscores the evolving role of AI in enhancing security practices across the cloud landscape.

Security Domain	AI Aspects
Organizational Management	AI where and how
Identity and Access Management (IAM)	AuthN/Z
Security Monitoring	AI monitoring and logging; AI for detection and analysis
Network	Network Security when self or cloud hosting AI
Workload	Secure AI workload hosting
Application	AI integration, API security
Data	Training data, data storage, data leaks

Table 12: Intersection of Security Domains with AI Aspects

- For organizational management, the organization must determine AI policies, providers, and expectations. Based on the service's specifics, accounts and services will be established, and security controls enabled.
- As with any cloud service, IAM will be the most important security control. For AI, this will affect users, administrators, the AI model/workload itself, and access to any underlying training or analytics data.
- Monitoring should include prompts, output, and data access.
- The underlying network must be secured when hosting an AI service. Network security may also be required to restrict access to any AI as a service (AIaaS) platform.
- Any workloads running AI, or accessing AI, will need to follow basic security practices.
- Much AI security is implemented at the application layer, including application logic and API security.
- All training, analysis, and other data repositories must be secured. These stores will often have massive amounts of data.

12.2.1.1 How AI Intersects with Cloud Security

AI is reshaping how organizations approach security in digital environments and its intersection with cloud security represents a paradigm shift, to address evolving cybersecurity challenges.

As organizations increasingly rely on cloud services to host critical workloads and sensitive data, integrating AI technologies introduces new opportunities to enhance security measures and mitigate risks. From AIaaS offering to leveraging cloud infrastructure for AI model deployment, there is a spectrum of options to utilize AI's power in securing cloud environments.

AI-enhanced security tools affect threat detection, access control, and policy enforcement. Understanding the various AI consumption models and their integration with cloud security tools becomes essential for organizations seeking to fortify their defenses and navigate the complexities of modern cybersecurity landscapes.

AI intersects with cloud security in various ways, categorized into four main categories:

1. **AI as a Service for consumption (full SaaS)**: In this model, AI is provided as a complete, ready-to-use service by the cloud provider. Offerings like Claude can leverage AI capabilities without having to build or train an organization’s own models. Full software as a service (SaaS) is ideal for organizations that want to quickly adopt AI without deep technical expertise, because you can easily select only those services that have been approved by your organization. Most products in this category include data privacy upgrades, the option to only allow approved data and the ability to easily track prompts and results.

2. **AI as a Service (PaaS and Foundation model hosting[186])**: The cloud provider offers the underlying infrastructure and tools to host and run AI models, but leaves the model development and application building to the customer. AWS Bedrock is an example of this; it provides the foundation models and hosting environment, but customers create their own solutions built upon it. This model gives organizations more control and customization, and defends against adversarial attacks, such as injections or jailbreaks. Other features include secure training data, secure application integration and deployment environments, and secure users and access.

3. **Cloud as workload host for AI (Bring Your Own Model)**: In this scenario, organizations develop their own AI models from scratch or deploy off-the-shelf models (code) and simply use the cloud as the hosting environment. They are responsible for the entire AI lifecycle, from data preparation to model training to deployment. The cloud just provides the raw compute resources. This offers the most flexibility but requires the most in-house AI skills and has the same responsibilities as building an in-house application.

4. **AI-enhanced security tools**: In addition to the hosting options, AI is being embedded into various cloud security products to make them smarter and more effective. Think AI-powered threat detection, intelligent access control, automated policy enforcement, and so on. As AI matures, expect to see it enhance more traditional security solutions.

12.2.2 AI Enhanced Security Tools

AI and Machine Learning (ML) are already used extensively in security for use cases like malware detection and user behavior analytics. With the advent of Large Language Models (LLMs), new categories of AI-enhanced security tools are rapidly emerging, especially for analyzing and prioritizing data sets.

Figure 69: Use Cases Where AI Enhances Security Tools and Processes

[186] In the context of AI as a service (AIaaS) within a platform as a service (PaaS) model, foundation model hosting involves offering infrastructure and resources optimized for deploying, running, and managing foundational AI models.

Key Areas of AI and LLM applications in security are:

- **Threat Detection:** Leverages AI and ML to analyze network traffic and system behavior, enhancing the identification of emerging threats while utilizing LLMs for generating insightful threat intelligence reports. AI algorithms can sift through massive amounts of data to identify suspicious patterns and potential threats much faster and more accurately than traditional rules-based systems. This helps security teams stay ahead of the constantly evolving threat landscape.

- **Log Analysis:** Utilizes AI and ML, including Natural Language Processing, to analyze unstructured log data, detect security patterns and anomalies, and provide actionable insights. Modern cloud environments generate a staggering volume of log data that are impossible for humans to review manually. AI can automatically parse these logs, correlate events across different systems, and flag anomalies that might indicate a security incident.

- **Incident Response:** AI enhances incident response by automating workflows, using ML models to prioritize alerts by risk, and employing LLMs to generate detailed incident reports and recommend actions. When a threat is detected, AI can help investigate the scope of the compromise, identify the root cause, and even automatically contain the damage by isolating affected systems or revoking compromised credentials. This dramatically speeds up response times.

- **Posture Assessments:** Leverage AI to continuously monitor and evaluate an organization's security across all domains, utilizing ML to pinpoint misconfigurations and security lapses, while LLMs provide detailed summaries and actionable recommendations for enhancing security measures.

- **Secure Code Analysis:** Leverages AI to scrutinize source code for vulnerabilities, refine recommendations through ML models, and utilizes LLMs to explain and suggest secure coding practices. This 'shift left' approach, which includes Architecture Risk Analysis, Dynamic Analysis, and other proactive techniques, catches issues before they move into production, reducing risk.

- **Malware Analysis:** AI enhances malware reverse engineering by automating tasks like code deobfuscation and behavior analysis, while ML classifies malware families and identifies patterns, and LLMs generate detailed analysis reports to foster researcher collaboration.

- **Risk Prioritization**: Leverages AI and ML to analyze data from security tools and external sources, quantify risks based on multiple factors, and articulate risk assessments to inform stakeholders effectively. By quantifying risk, organizations can make data-driven decisions about where to focus their limited security resources for maximum effect.

- **Entitlement Management**: AI and ML enhance access control by analyzing roles and activity patterns to streamline permissions, optimize policies based on least privilege, and automate the generation of access review reports.

As AI and LLMs continue to advance, we expect to see even more innovative applications in the cybersecurity domain. However, organizations must also be mindful of AI's potential risks and limitations, such as bias, explainability, and adversarial attacks. By combining the power of AI with human expertise

and oversight, security teams can enhance their capabilities and stay ahead of evolving threats in an increasingly complex digital landscape.

12.3 Threat & Vulnerability Management

Threat and vulnerability management (TVM) allows organizations to better anticipate, detect, and respond to threats in a dynamic cloud environment, thereby reducing their vulnerability to cyber attacks and ensuring continuous security compliance. Additionally, the integration of AI into TVM is expected to become a standard practice in securing cloud environments. This section sets the stage for discussing further innovations and challenges in threat management for cloud services.

The following table presents TVM aspects that intersect with security domains.

Security Domain	TVM Aspects
Organizational Management	Organization policies, blast radius control, CSPM/CNAPP[187]
IAM	Credential protection, PIM/PAM
Security Monitoring	Detection and analysis
Network	Blast radius control, flow, DNS monitoring
Workload	Endpoint protection, detection, response
Application	Application and API security
Data	Resource policies, data logs

Table 13: Security domains belonging to Threat and Vulnerability Management

Here is a short analysis of each of the security domains mapped to TVM:

- **Organization Management:** Establishes organization-wide security policies, such as configuring blast radius controls and security posture management (CSPM/CNAPP) activities across the cloud environment.
- **IAM:** Handles credential protection and implements privileged identity management (PIM) and privileged access management (PAM).[188]

[187] Cloud Security Posture Management (CSPM) is a continuous and automated process to identify and remediate risks. Cloud-native application protection platform (CNAPP) secures the full application development lifecycle from code to production, and can replace tools, such as CSPM, System Information and Event Management (SIEM), and Cloud Workload Protection Platform (CWPP).
[188] PIM specifically deals with managing and protecting privileged identities and PAM focuses on controlling and monitoring privileged access to resources.

- **Security Monitoring:** Core domain for proactive threat detection and analyzing security events and alerts.
- **Network:** Controls network access and traffic flows by providing blast radius control through network segmentation and network traffic monitoring (e.g., flow logs, DNS queries).
- **Workload:** Focuses on protecting endpoints, detecting threats and coordinating incident response actions securing the compute instances, containers and serverless functions that run applications.
- **Application:** Secures the application layer through secure design, strict access control, and active protection; and leverages tools for app and API-specific vulnerabilities.
- **Data:** Defines policies for data handling, logs data events, scans for unusual access patterns, and investigates data leakage incidents.

12.3.1 Updating Threat Management for the Cloud

When it comes to maintaining and updating threat management strategy for cloud services, the responsibility for securing those cloud services falls under different parties, depending on the situation and organizations involved. For example, the CSP is responsible for keeping their services secure, while, the organization or consumer[189] are responsible for configuring and using those services.

In the cloud, the management plane (e.g., CSP console, APIs) becomes a primary target for attackers. It is important to focus on defending and monitoring cloud management plane access and activity by protecting, in addition to rotating access keys and credentials and monitoring unusual behavior or suspicious actions.

When considering the management plane as a new attack surface, consider the following.

- **Management Plane as New Attack Surface:** Focus on defending and monitoring cloud management plane access and activity, because the management plane becomes a primary target for attackers.
- **Vulnerability Scanning:** Vulnerability Scanning uses tools like CSPM, SSPM, and CASBs to identify and fix security issues in IaaS and SaaS setups, and Infrastructure as Code (IaC) scanners to correct misconfigurations early in development.
- **Securing Containers and VMs:** Utilize modern vulnerability management tools like CWPP/CNAPP for dynamic cloud environments and integrate scanning into continuous integration/continuous delivery (CI/CD) pipelines to proactively detect issues, as traditional methods falter with ephemeral assets like containers.

[189] Cloud security practitioners, IT administrators, cybersecurity professionals, and individuals responsible for managing and securing cloud environments.

- **Credential Theft and Privilege Escalation:** To mitigate the risk of credential theft and privilege escalation, it is crucial to implement robust IAM controls, enforce least privilege access policies, regularly adjust and review permissions, and monitor for any suspicious credential usage.
- **Cloud-Native Threat Detection:** Utilizes cloud platform features such as VPC flow logs, DNS logs, and both agent-based and agentless solutions to monitor and identify threats across network and cloud workloads (CWPP/CNAPP).
- **Software Supply Chain Security:** Implements measures like code or image signing, automated vulnerability scanning, and secure artifact management to safeguard software dependencies and enhance response to code vulnerabilities.
- **Threat Intelligence:** Use threat intelligence from CSPs and third-party feeds to stay informed, and leverage threat intel to proactively hunt for indicators of compromise.

Recommendations and strategies for teams in an organization to collaboratively identify, mitigate, and respond to emerging threats on the cloud, can be found in the table below.

Security Practice or Strategy	Implementation Recommendations
Management Plane as New Attack Surface	In the cloud, the management plane (e.g., CSP console, APIs) becomes a primary target for attackers. • Focus on defending and monitoring cloud management plane access and activity • Protect and rotate access keys and credentials • Look for unusual behavior or suspicious actions in management plane logs
Vulnerability Scanning	• Cloud Security Posture Management (CSPM) tools scan an organization's IaaS environment to identify misconfigurations and security gaps • SaaS Security Posture Management (SSPM) and Cloud Access Security Brokers (CASB) assess the organization's SaaS security settings and usage • Regularly use these tools to find and fix vulnerabilities and misconfigurations before attackers can exploit them. These misconfigurations are often instantly accessible on the Internet since the organization has no network perimeter to control them • IaaC (Infrastructure as a Code) scanners to detect security misconfigurations before the changes are deployed to production. In other words, shift left strategies are deployed to reduce vulnerabilities as early in the software development lifecycle
Securing Containers and VMs	• Use modern vulnerability management tools and processes designed for dynamic cloud environments (e.g., CWPP/CNAPP)

	• Traditional scanning approaches will not work well for short-lived assets like containers • Integrate vulnerability scanning into CI/CD pipelines to catch issues early

Credential Theft and Privilege Escalation	• Attackers increasingly target cloud credentials and privileges to gain unauthorized access and pivot attacks
	• Implement strong IAM controls:
	  ○ Use least privilege access policies
	  ○ Regularly review and right-size permissions
	  ○ Monitor for suspicious credential usage

Cloud-Native Threat Detection	• Cloud platforms provide native capabilities for network and host-based threat detection • Leverage features like VPC flow logs and DNS logs for network security monitoring • Use agent-based or agentless solutions for threat detection on cloud workloads (CWPP/CNAPP)

Software Supply Chain Security	• Securing an organization's software dependencies and understanding its Software Bill of Materials will help respond more efficiently to vulnerabilities in its code and image bases.
	• Implement controls like:
	  ○ Code or image signing and integrity verification
	  ○ Automated vulnerability scanning and patching
	  ○ Secure artifact repositories and release management processes

Threat Intelligence	• Leverage an organization's CSP's threat intelligence feeds to stay informed of cloud-specific threats and trends • Supplement with third-party threat intelligence to get a more comprehensive view • Use threat intel to proactively hunt for indicators of compromise and improve detections.

Table 14: Cloud Security Practices and Implementation Recommendations

12.3.2 Cloud Threat Intelligence Sources

Many threat intelligence sources focus on threat actors and indicators of compromise that are irrelevant to the cloud, or not cloud-specific. The following sources can enhance an organization's threat intelligence for the cloud.

- The CSA *Top Threats*[190] report includes threat modeling for the most common active breaches, as seen in public incidents. The project also includes deep dives into major incidents. This is incredibly useful in understanding how real breaches are occurring[191].
- MITRE ATT&CK®[192] includes a Cloud Matrix describing the tactics and techniques (and sub-techniques) attackers use to target enterprise cloud deployments.
- Many vendors release threat research and reports. These increasingly include information on cloud attacks seen by their research and response teams. However, it is necessary to carefully evaluate these reports because vendors may have selection bias based on their business, products, and prior experiences.
- Open Sources are run independently and collect and share public threat data. Breaches.cloud[193] is an example that tracks known public breaches and is being actively maintained.

Summary

We have begun to analyze cloud security challenges through the prism of related technologies and strategies. With ZT, you continuously verify all users and devices, minimize trust, and apply least privilege principles, using techniques such as multi factor authentication, micro-segmentation, and encryption to protect resources, reducing the attack surface and enhancing security resilience. With AI, you can enhance cloud security through threat detection, access control, and policy enforcement. AI can also leverage machine learning for improved anomaly detection and risk management. With TVM, you identify, assess, and mitigate security threats, using tools such as CSPM and continuous monitoring. TVM also helps you protect cloud environments and ensure compliance. Integrating AI into your TVM enhances threat detection and response strategies, helping to maintain a robust security posture.

Recommendations

Effective cloud security always starts with establishing a solid governance model that provides the framework for organizing responsibilities, identifying risks, and managing policy controls.

Governance and Framework: This involves clearly defining roles and responsibilities across the organization, identifying key risks associated with cloud usage, and implementing a framework to manage security controls consistently. The governance structure should align with overall business objectives while providing adequate oversight and accountability.

The IANS Cloud Security Maturity Model[194] can guide and support security programs.

From IaaS to SaaS, IAM is the first place to focus security control efforts.

[190] CSA. (2022) *Top Threats to Cloud Computing Pandemic Eleven*.
[191] CSA. (2023) *The Common Cloud Miscofigurations That Lead to Cloud Data*.
[192] MITRE (2024) ATT&CK®.
[193] Public Cloud Security Breaches is a website that tracks security breaches on public clouds.
[194] IANS. (2024) Cloud Security Maturity Model Version 2.0 - *What is the Cloud Security Maturity Model*.

- Use organization management to control the organization's blast radius and cloud security posture.
- Establish consistent security telemetry collection for effective monitoring.
- Network, workload, application, and data security will usually have a set of shared services, but security will need to be customized for the needs of different deployments.
- An organization's cloud security control specifications will define the baseline requirements, but it is necessary to collaborate with DevOps and cloud teams on secure design and architecture.
- Use continuous assessment to identify misconfigurations that lead to public exposures or create IAM vulnerabilities, and incident response (including threat detection) to identify and remediate attacks and exposures rapidly.

The Cloud Security Maturity Model also helps structure and guide cloud security program development. Some recommendations follow below.

IAM as a Priority

IAM should be a top priority for any cloud security program, regardless of the service model (IaaS, PaaS, or SaaS). IAM controls who can access what resources and what actions they can perform. Misconfigured or poorly managed IAM can lead to unauthorized access, data breaches, and other security incidents. Focus on implementing strong authentication mechanisms, applying the principle of least privilege, regularly reviewing and rotating access keys, and monitoring anomalous activities.

Organization Management for Blast Radius Control and Monitoring

Use the organization management capabilities that cloud platforms provide to control the blast radius (the potential impact of a security incident). This can be achieved through proper account structuring, using separate accounts for different environments (e.g., production, staging, development), and implementing network segmentation. Establish a consistent security telemetry collection process to centralize logs and events from various sources, enabling effective monitoring and incident response.

Customizing Security for Different Deployments

While there may be a set of shared security services across the organization, it is important to tailor security controls to the specific needs of different deployments. Network security requirements for a public-facing web application will differ from those of an internal database. Similarly, container workload security measures will vary from those for serverless functions. Work closely with application teams to understand their unique security requirements and implement appropriate controls.

Collaboration with DevOps and Cloud Teams

Cloud security should not operate in a silo. Collaborate with DevOps and cloud teams to incorporate security into the design and architecture phase of projects. Define baseline security control specifications that teams can reference, but be open to adapting them based on specific use cases. Foster a culture of shared responsibility, where everyone plays a role in maintaining the security posture.

Continuous Assessment and Incident Response

Implement continuous assessment processes to proactively identify misconfigurations that can lead to security risks, such as public exposure of resources or IAM vulnerabilities. Regularly scan for and remediate these issues. Have a robust incident response plan to quickly detect, investigate, and mitigate security incidents. Leverage threat detection tools and automate response workflows to minimize the impact of potential breaches.

It is important to remember that cloud security is an ongoing process that requires continuous monitoring, assessment, and improvement. Regular reviews and updates of the organization's security policies, procedures, and controls are recommended to keep pace with the evolving threat landscape and changes in the cloud environment.

Additional Guidance

- Introduction to Generative AI & Prompt Engineering | CSA
- Principles to Practice: Responsible AI in a Dynamic Regulatory Environment | CSA
- AI Resilience: A Revolutionary Benchmarking Model for AI Safety | CSA
- AI Organizational Responsibilities - Core Security Responsibilities | CSA
- Certificate of Competence in Zero Trust (CCZT) | CSA
- DoD Zero Trust Reference Architecture | DoD
- Zero Trust Maturity Model | CISA
- Cybersecurity Framework | NIST
- SP 800-207A, A Zero Trust Architectural Model | NIST
- CIS Critical Security Controls
- ISO/IEC 27001:2022, Information security, cybersecurity and privacy protection - Information security management systems requirements