AWS CERTIFIED

CLOUD PRACTITIONER

Security and Compliance

AMBER ISRAELSEN

Developer | Technical Trainer

Course Outline

Course Introduction

Cloud Concepts

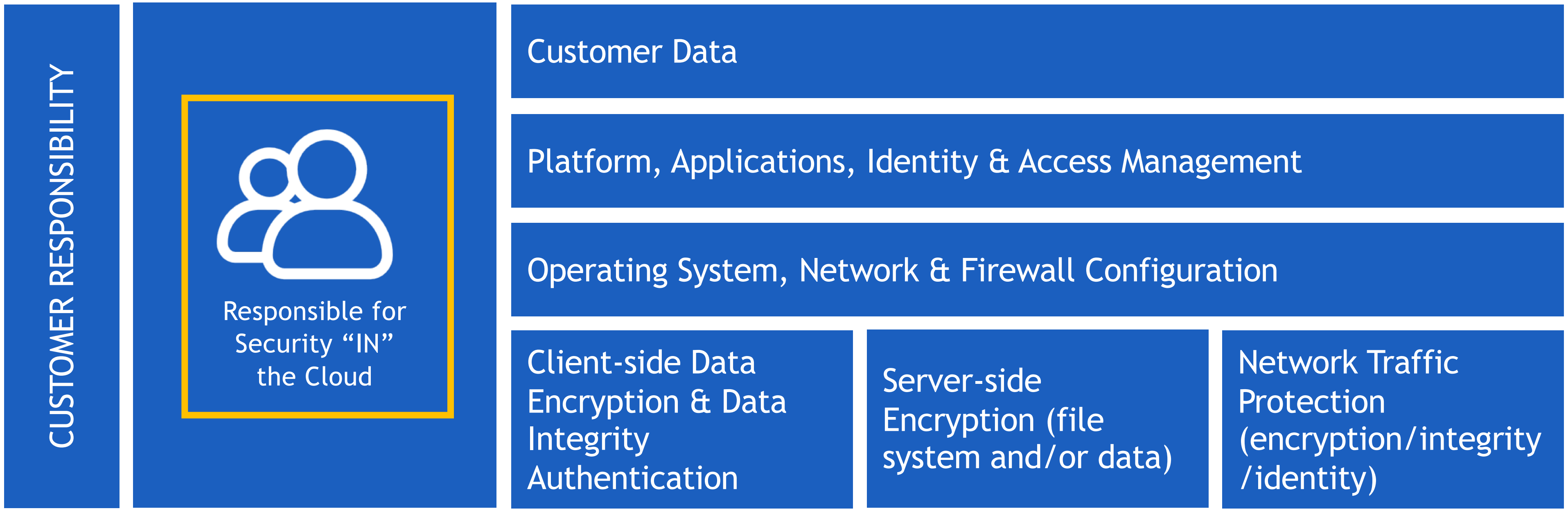
Security and Compliance

Cloud Technology and Services

Billing, Pricing and Support

Preparing for the Exam

AWS Shared Responsibility Model



CUSTOMER RESPONSIBILITY

Responsible for Security “IN” the Cloud

Customer Data

Platform, Applications, Identity & Access Management

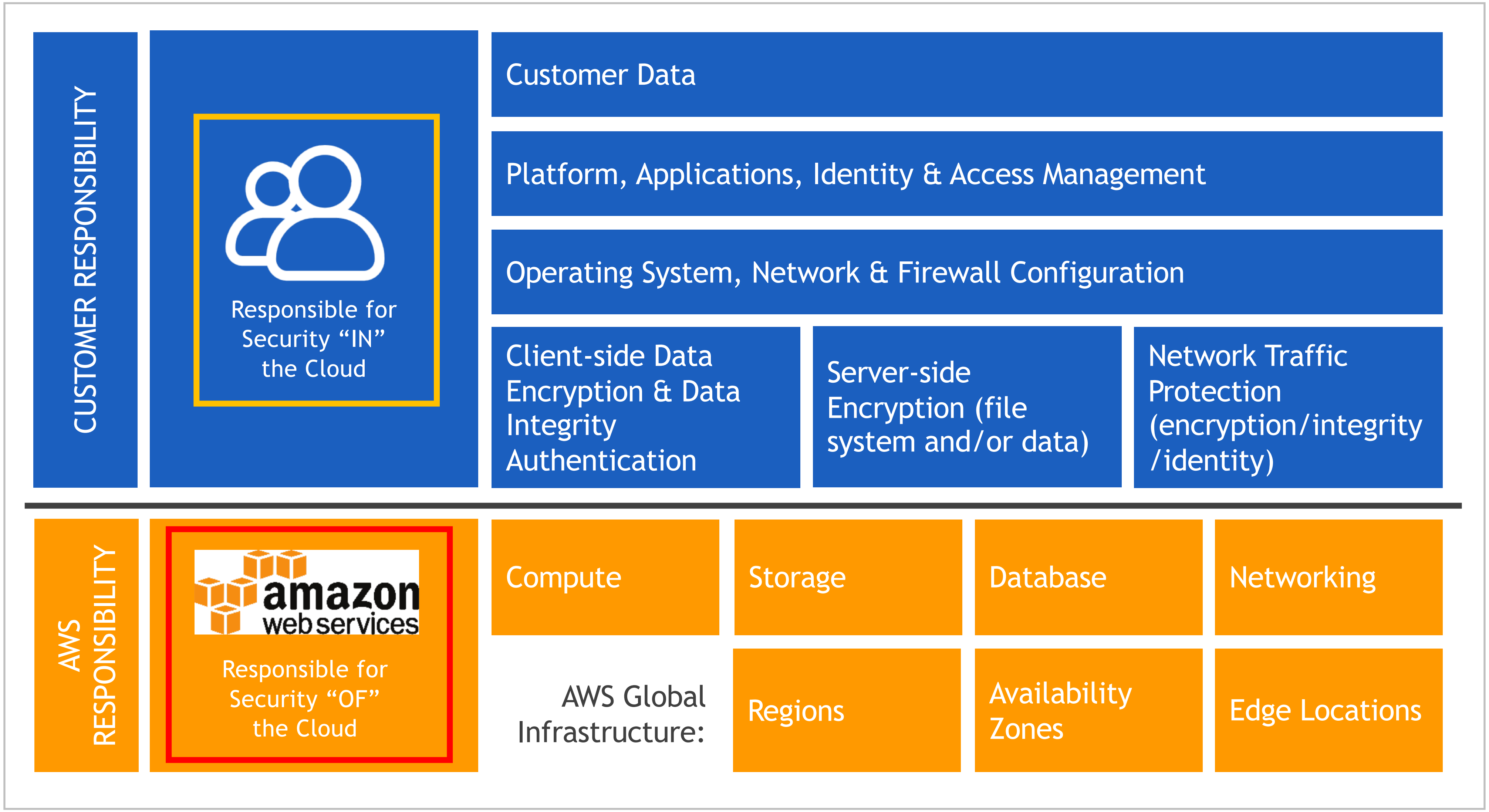
Operating System, Network & Firewall Configuration

Client-side Data Encryption & Data Integrity Authentication

Server-side Encryption (file system and/or data)

Network Traffic Protection (encryption/integrity /identity)

AWS Shared Responsibility Model



CUSTOMER RESPONSIBILITY

Responsible for Security “IN” the Cloud

Customer Data

Platform, Applications, Identity & Access Management

Operating System, Network & Firewall Configuration

Client-side Data Encryption & Data Integrity Authentication

Server-side Encryption (file system and/or data)

Network Traffic Protection (encryption/integrity /identity)

AWS RESPONSIBILITY

amazon web services

Responsible for Security “OF” the Cloud

Compute

Storage

Database

Networking

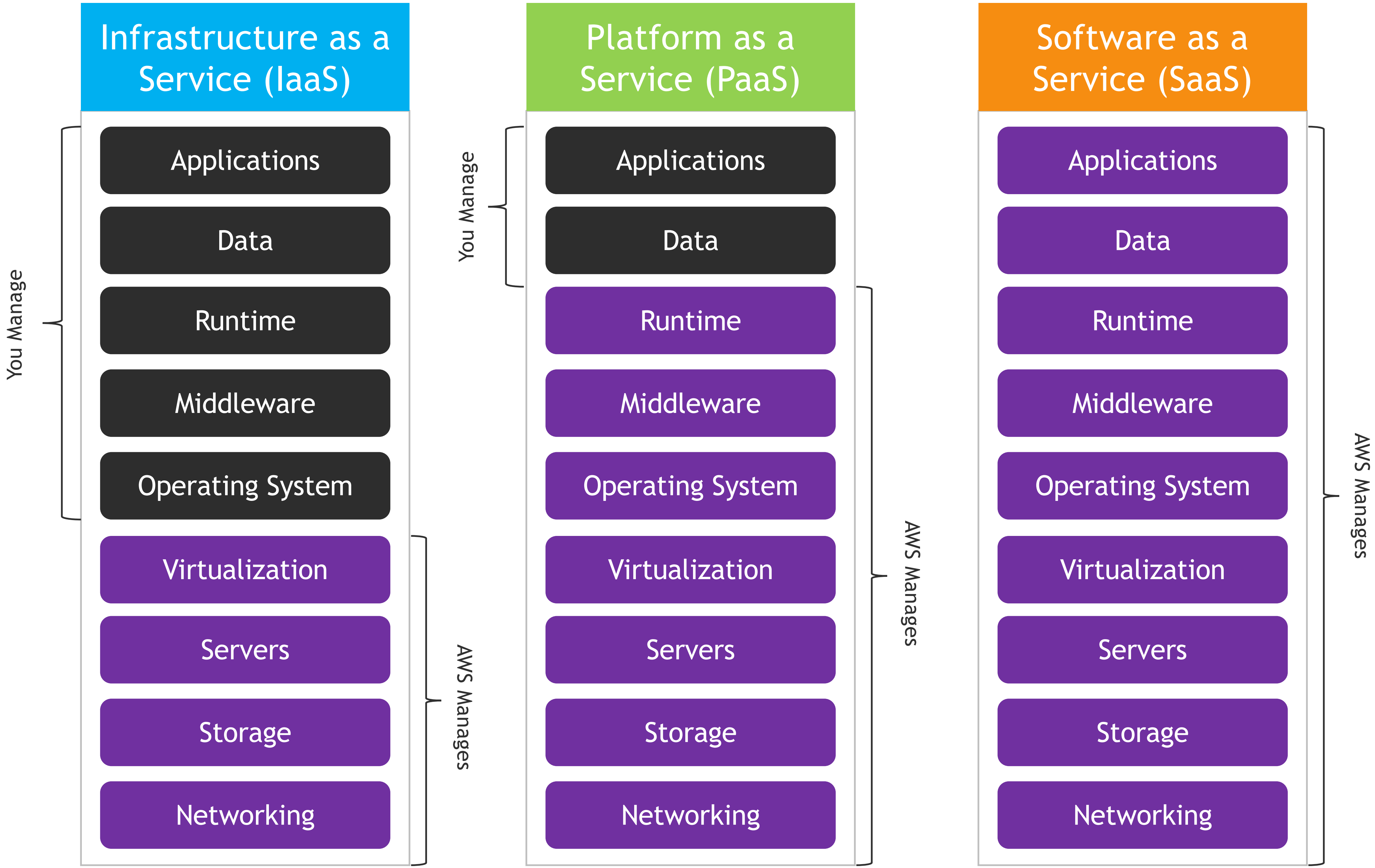
AWS Global Infrastructure:

Regions

Availability Zones

Edge Locations

Your responsibilities can vary depending on the service (i.e., managed vs. non-managed)



Infrastructure as a Service (IaaS)

You Manage

- Applications
- Data
- Runtime
- Middleware
- Operating System

AWS Manages

- Virtualization
- Servers
- Storage
- Networking

Platform as a Service (PaaS)

You Manage

- Applications
- Data

AWS Manages

- Runtime
- Middleware
- Operating System
- Virtualization
- Servers
- Storage
- Networking

Software as a Service (SaaS)

AWS Manages

- Applications
- Data
- Runtime
- Middleware
- Operating System
- Virtualization
- Servers
- Storage
- Networking

Identity and Access Management (IAM)

Service used to securely control access to your AWS resources

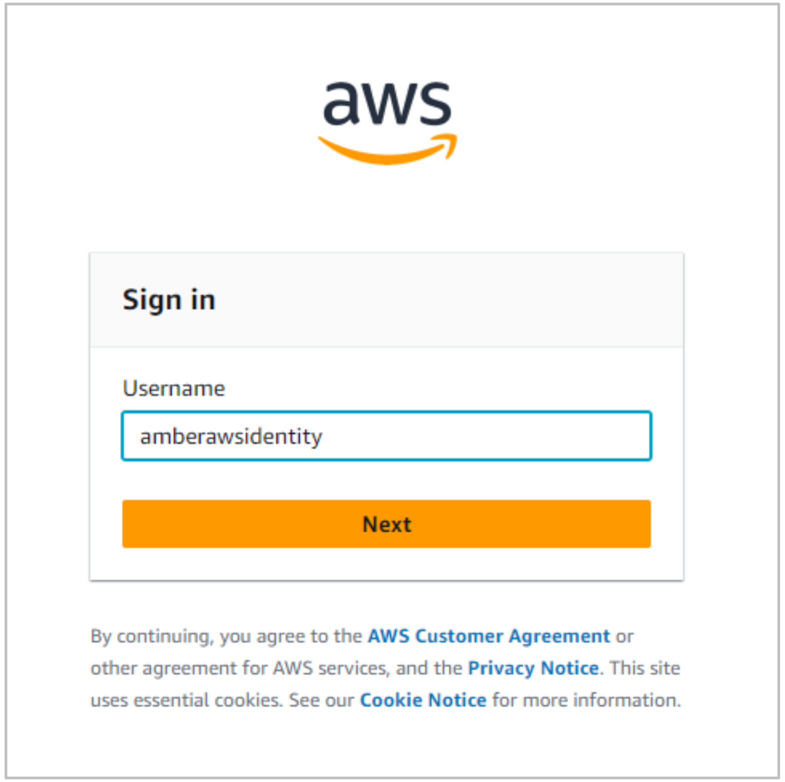
Controls authentication (who) and authorization (what they can do)

IAM Identity Center (formerly AWS Single Sign-On or SSO)

Provides a single login across:

- All AWS accounts (leveraging AWS Organizations)
- Cloud-based applications like Salesforce, Box, Microsoft 365
- EC2 instances running Windows
- SAML 2.0-enabled applications

Can use multiple identity providers, such as Active Directory, Okta, the built-in IAM store and more



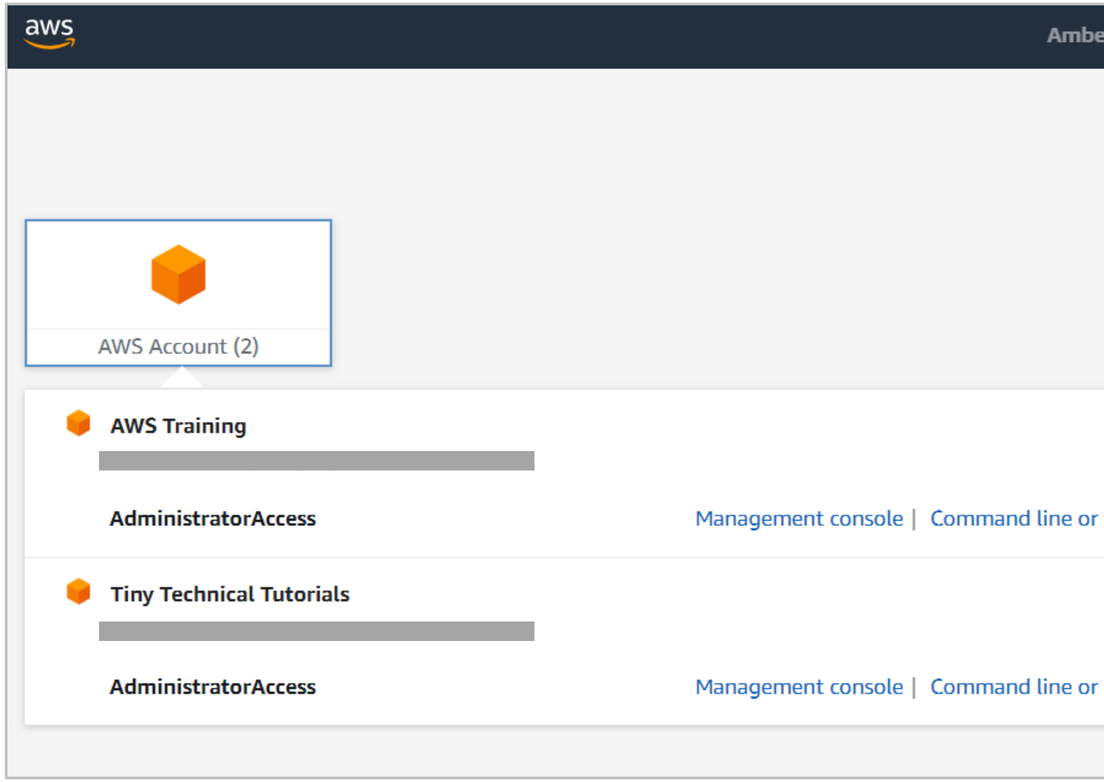
Sign in

Username

amberawsidentity

By continuing, you agree to the **AWS Customer Agreement** or other agreement for AWS services, and the **Privacy Notice**. This site uses essential cookies. See our **Cookie Notice** for more information.

A single login page



AWS Account (2)

AWS Training

[redacted]

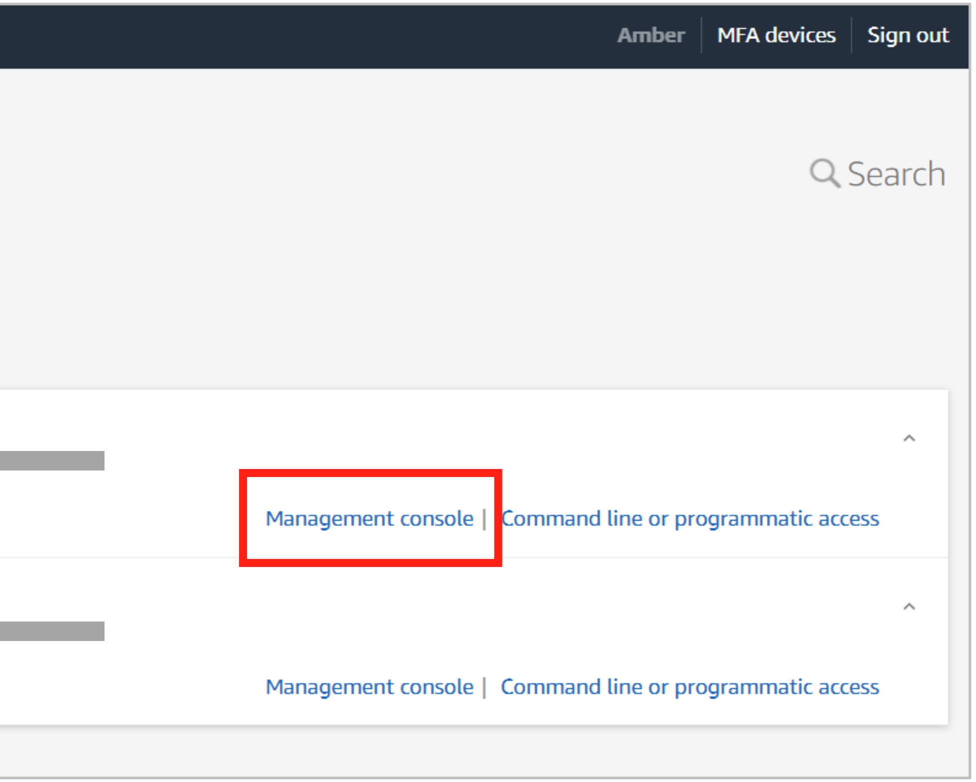
AdministratorAccess Management console | Command line or

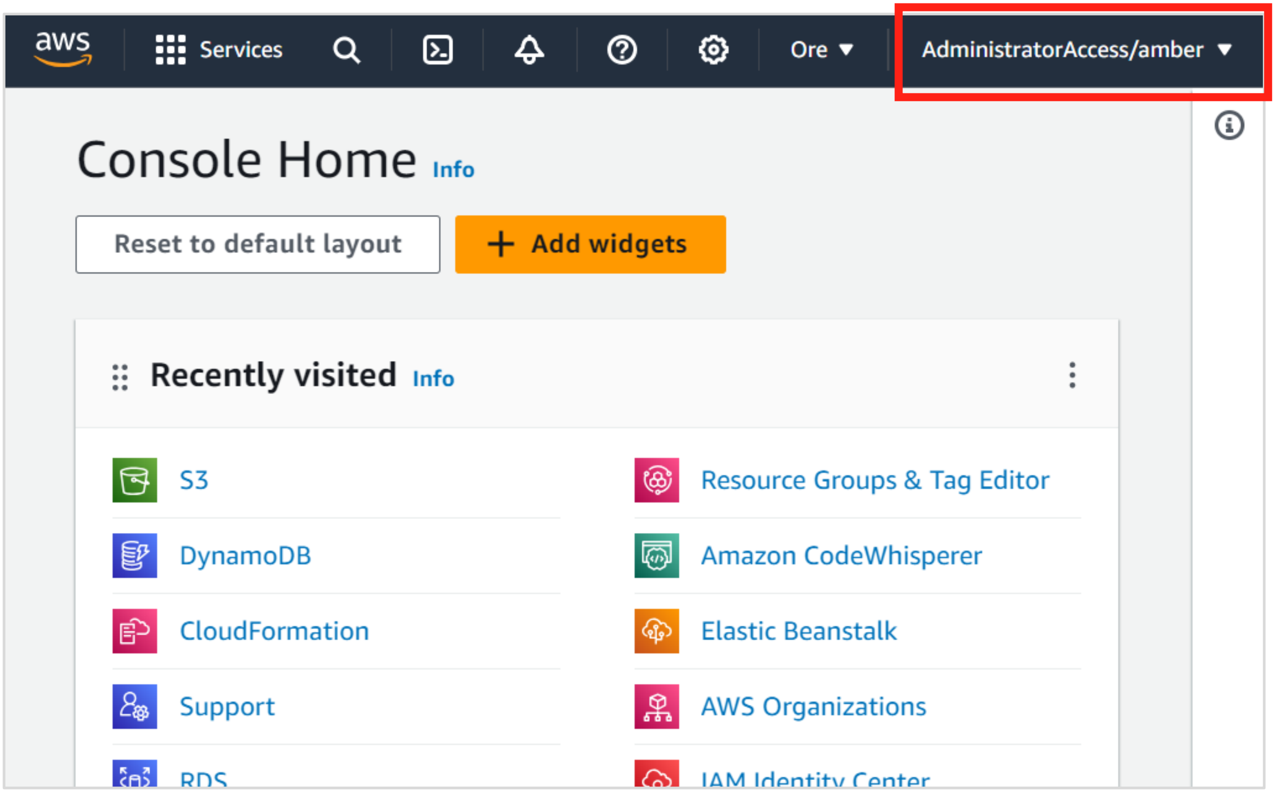
Tiny Technical Tutorials

[redacted]

AdministratorAccess Management console | Command line or

A list of all my accounts


Amber
MFA devices
Sign out
Search
Management console | Command line or programmatic access
Management console | Command line or programmatic access


aws
Services
Ore ▼
AdministratorAccess/amber ▼
Console Home Info
Reset to default layout
+ Add widgets
Recently visited Info
S3
DynamoDB
CloudFormation
Support
RDS
Resource Groups & Tag Editor
Amazon CodeWhisperer
Elastic Beanstalk
AWS Organizations
IAM Identity Center

IAM vs. IAM Identity Center

Manage identities and access in a single AWS account

Not using AWS Organizations

Centrally manage access across multiple accounts and applications

Using AWS Organizations

The recommended way to manage Console access, command line and programmatic access (i.e., access keys IDs and secret access keys)

1. Users
2. User Groups
3. Roles
4. Policies (and attach them)

Root User vs. IAM User

One per account	Multiple per account
Unrestricted access	Users can be deleted or disabled
Difficult to restrict or revoke access	Easy to restrict access
Can perform the following tasks:	

Close an AWS account

Change an AWS support plan

Change AWS account settings

BEST PRACTICES

- Always work in your IAM account, not the root account
    - Set up IAM users with least number of permissions needed
- Don’t create access keys for the root account (or delete them if you have them)
- Enable multi-factor authentication

1. Users
2. User Groups
3. Roles
4. Policies (and attach them)

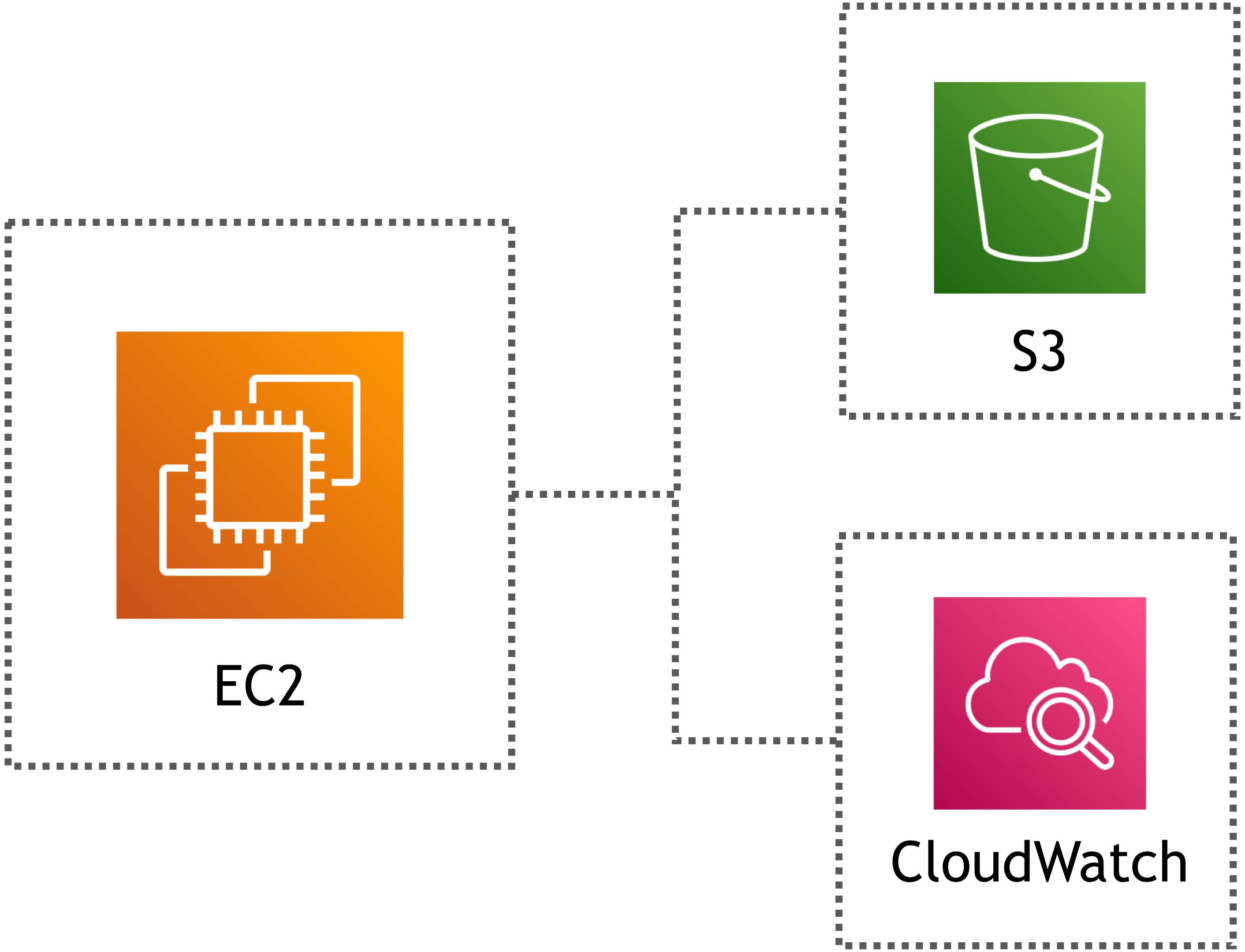
1. Users
2. Groups
3. Roles
4. Policies (and attach them)

Role

Similar to a user (an identity with permissions)

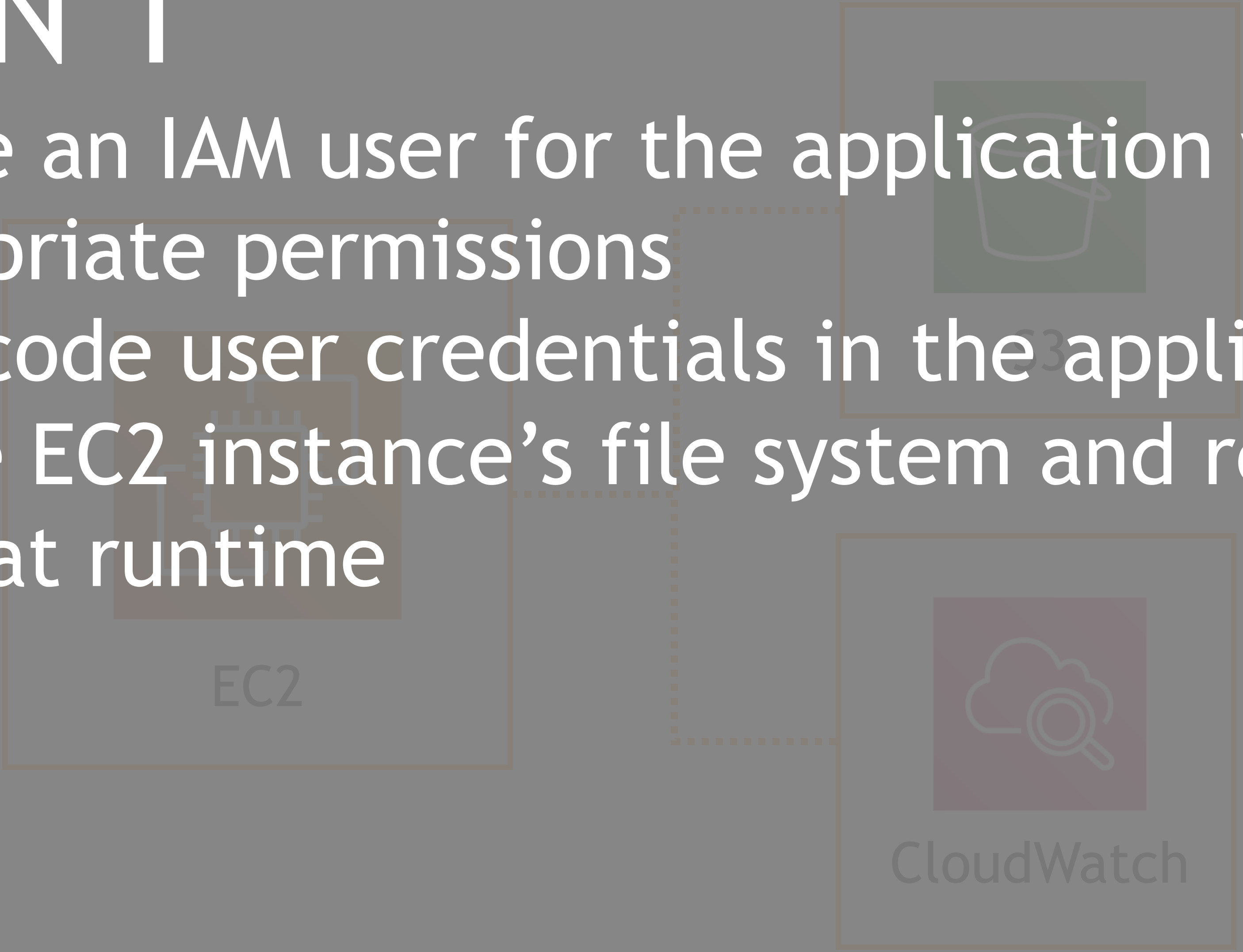
Does not have credentials (password or keys)

Assumable, temporarily, by anyone who needs it


S3
EC2
CloudWatch

OPTION 1

- Create an IAM user for the application with appropriate permissions
- Hard-code user credentials in the application or on the EC2 instance’s file system and retrieve them at runtime



OPTION 1

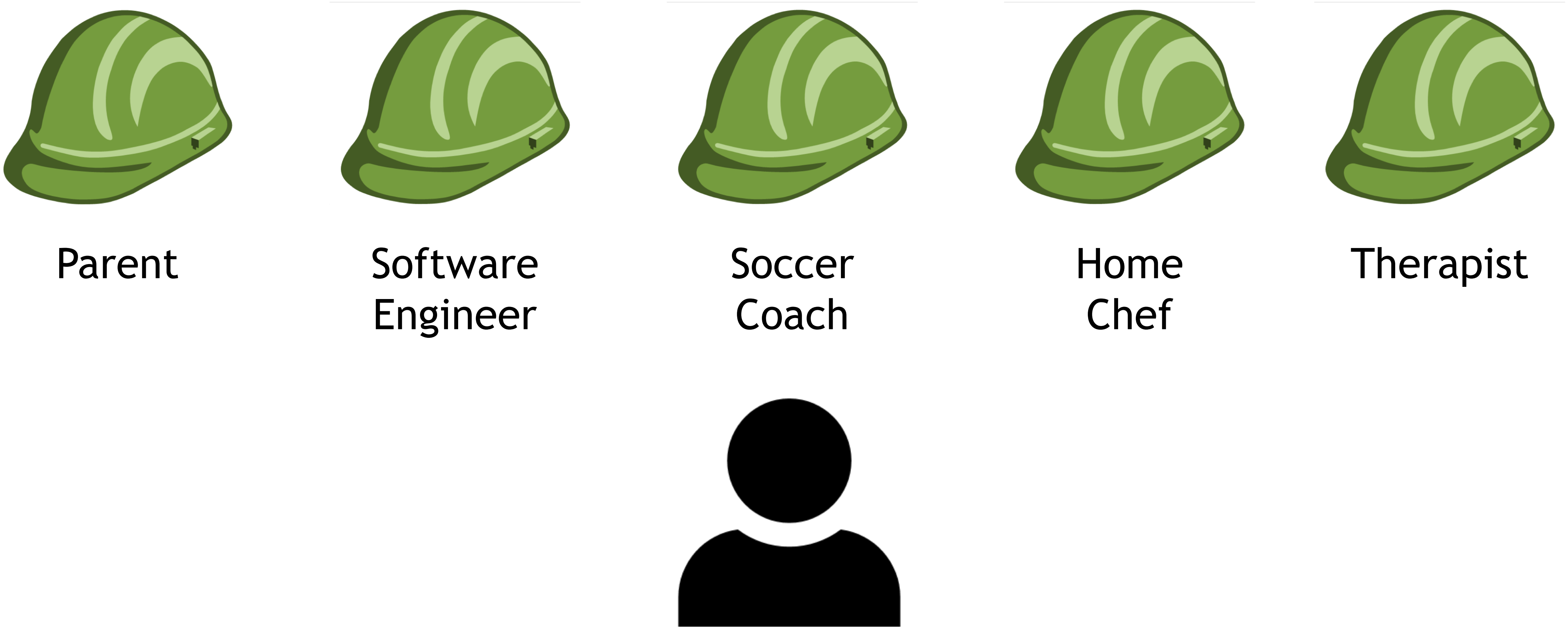
- Create an r th n with appropria
- Hard-code us application or on the EC2 insta n and retrieve them at runtim

OPTION 2

- Create an IAM role for the application with appropriate permissions
- When creating the EC2 instance, assign it this role
- No credentials required

ROLE

A Real-Life Analogy



Parent

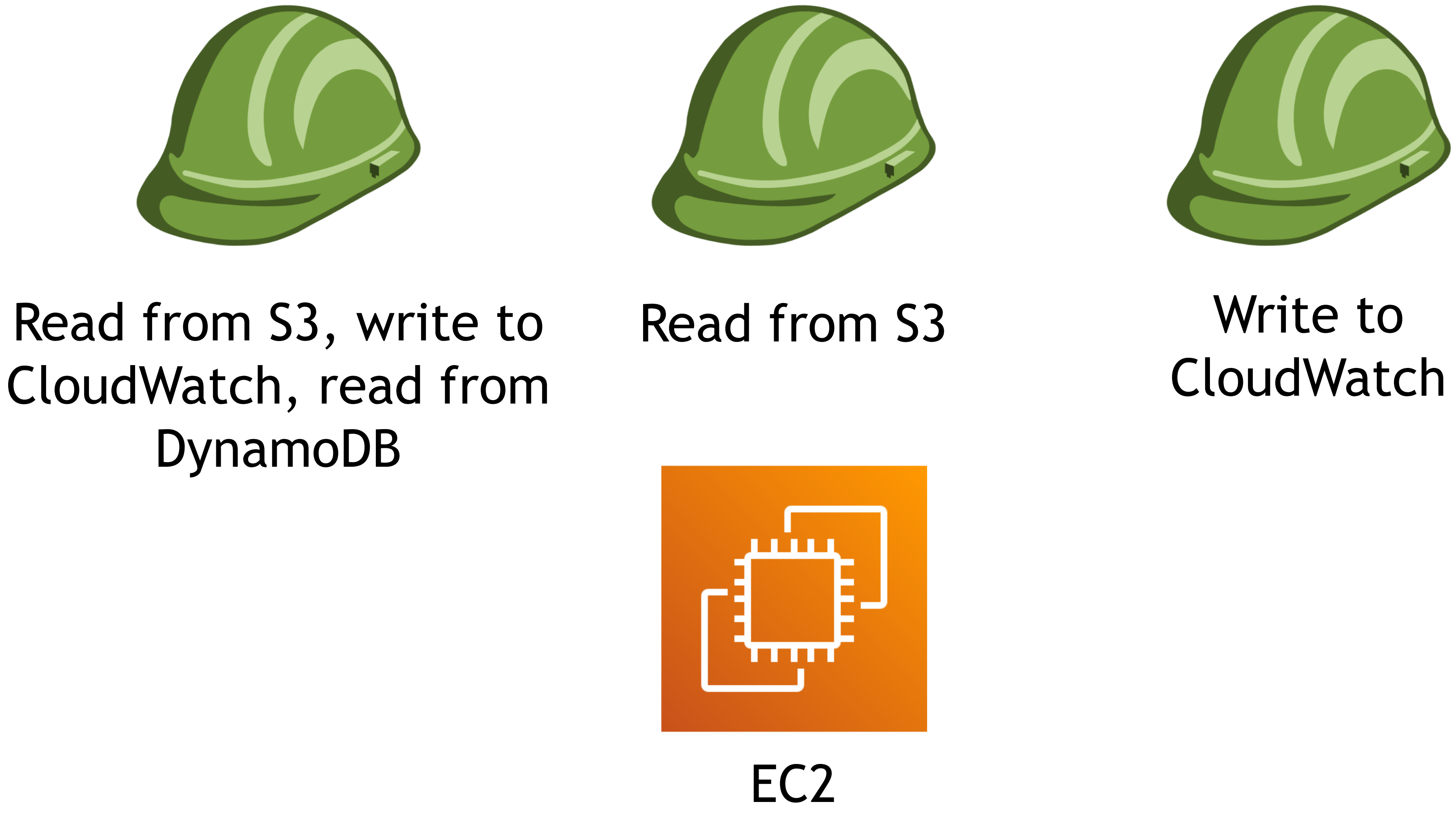
Software Engineer

Soccer Coach

Home Chef

Therapist

An AWS Example



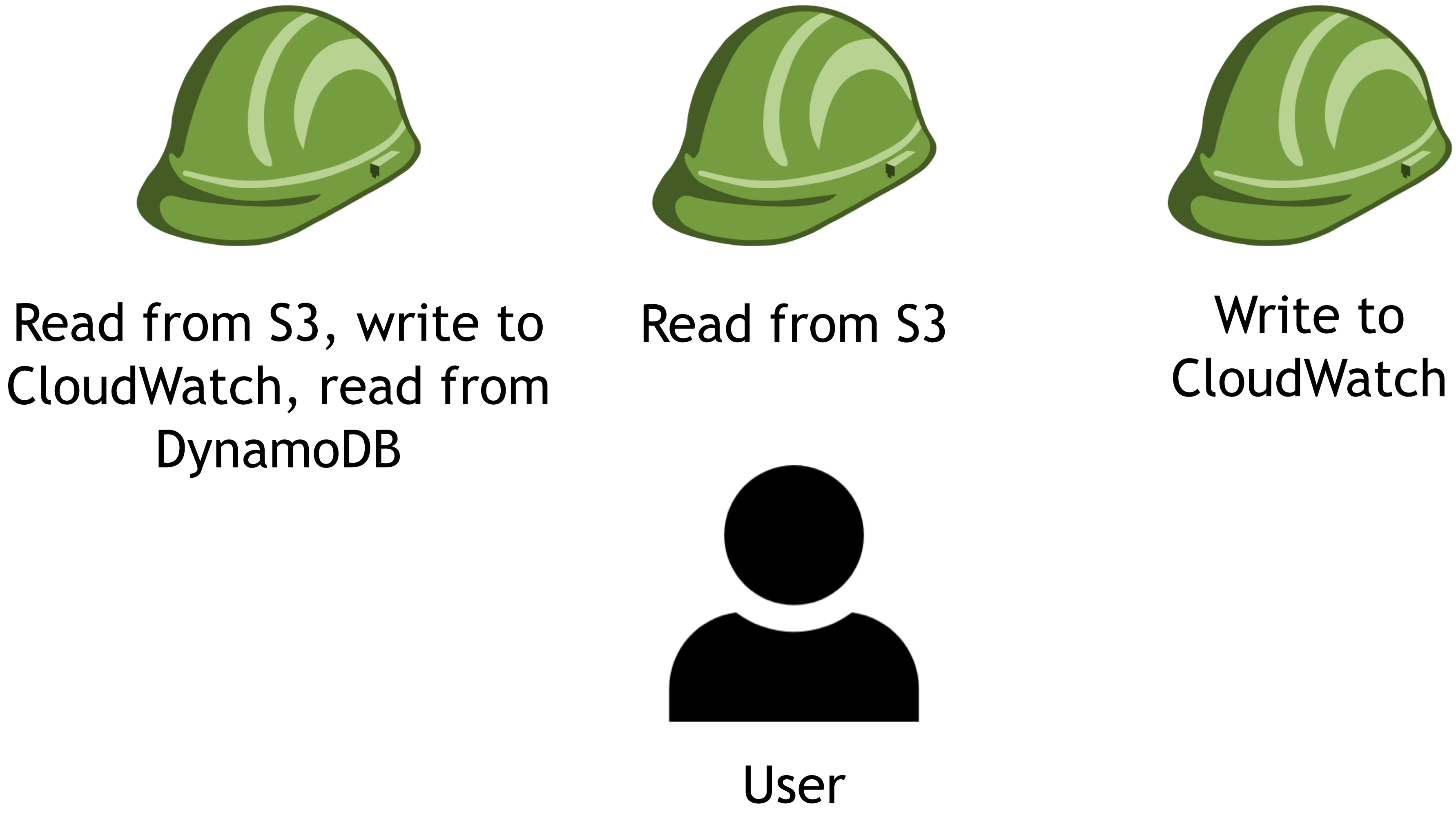
Read from S3, write to CloudWatch, read from DynamoDB

Read from S3

Write to CloudWatch

EC2

An AWS Example

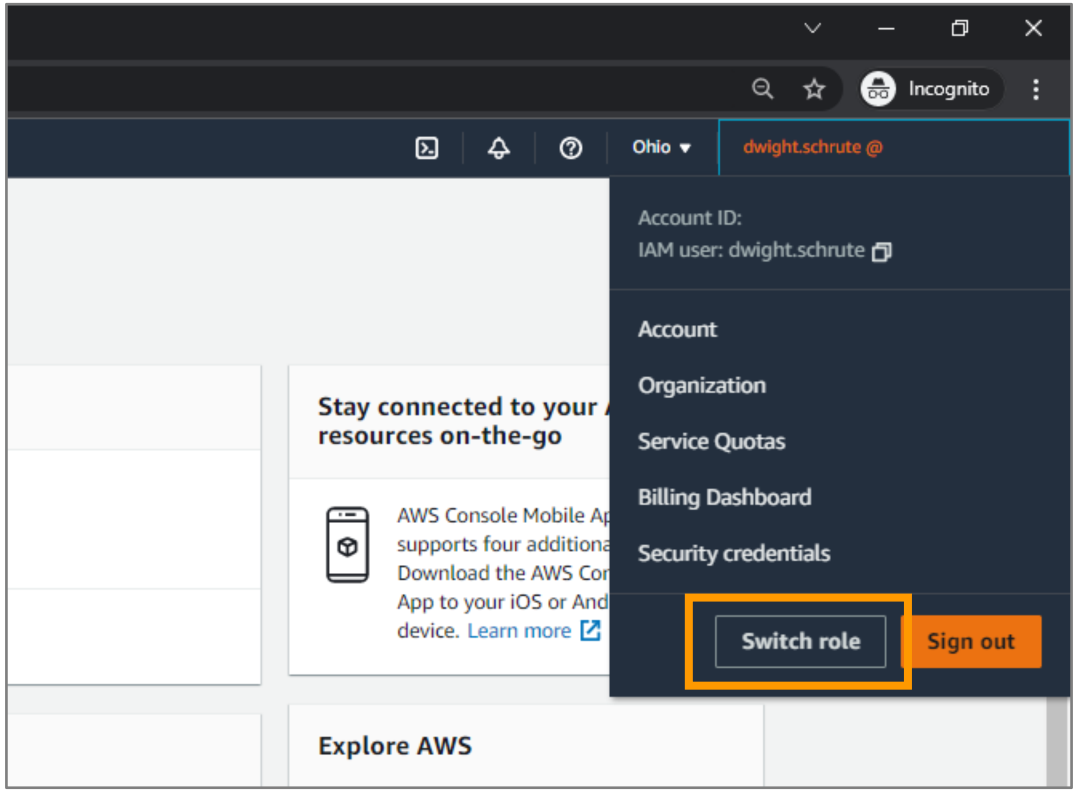


Read from S3, write to CloudWatch, read from DynamoDB

Read from S3

Write to CloudWatch

User



Account ID:
IAM user: dwight.schrute

Stay connected to your
resources on-the-go

AWS Console Mobile Ap
supports four additiona
Download the AWS Co
App to your iOS or And
device. Learn more

IDENTITIES (the “Who”)

USER

USER GROUP

ROLE

I can’t do anything.

1 Users

2 Groups

3 Roles

4 Policies (and attach them)

JSON

POLICY

Who can do what to which resources and when

JSON

POLICY

“Allow IAM users to rotate their own credentials programmatically and in the console.”

JSON

POLICY

“Allow a user to start and stop EC2 instances.”

JSON

POLICY

“Allow a Lambda function to access a DynamoDB table.”

```
{
    "Statement": [
        {
            "Effect":"effect",
            "Action":"action"
            "Resource":"arn"
            "Condition":{
                 "condition":{
                      "key":"value"
                 }
             }
        }
    ]
}
```

```
{
    "Statement": [
        {
            "Effect":"effect",
            "Action":"action"
            "Resource":"arn"
            "Condition":{
                "condition":{
                    "key":"value"
                }
            }
        }
    ]
}
```

Allow or Deny

Default is Deny

```
{
    "Statement": [
        {
            "Effect":"effect",
            "Action":"action"
            "Resource":"arn"
            "Condition":{
                 "condition":{
                      "key":"value"
                 }
            }
        }
    ]
}
```

Corresponds to API calls to AWS services

Example:
`s3:DeleteBucket`

```
{
    "Statement": [
        {
            "Effect":"effect",
            "Action":"action",
            "Resource":"arn"
            "Condition":{
                "condition":{
                    "key":"value"
                }
             }
        }
    ]
}
```

Amazon Resource Name

The resource name you want to apply permission to

Example:
`arn:aws:ec2:*:*:instance/instance-id`

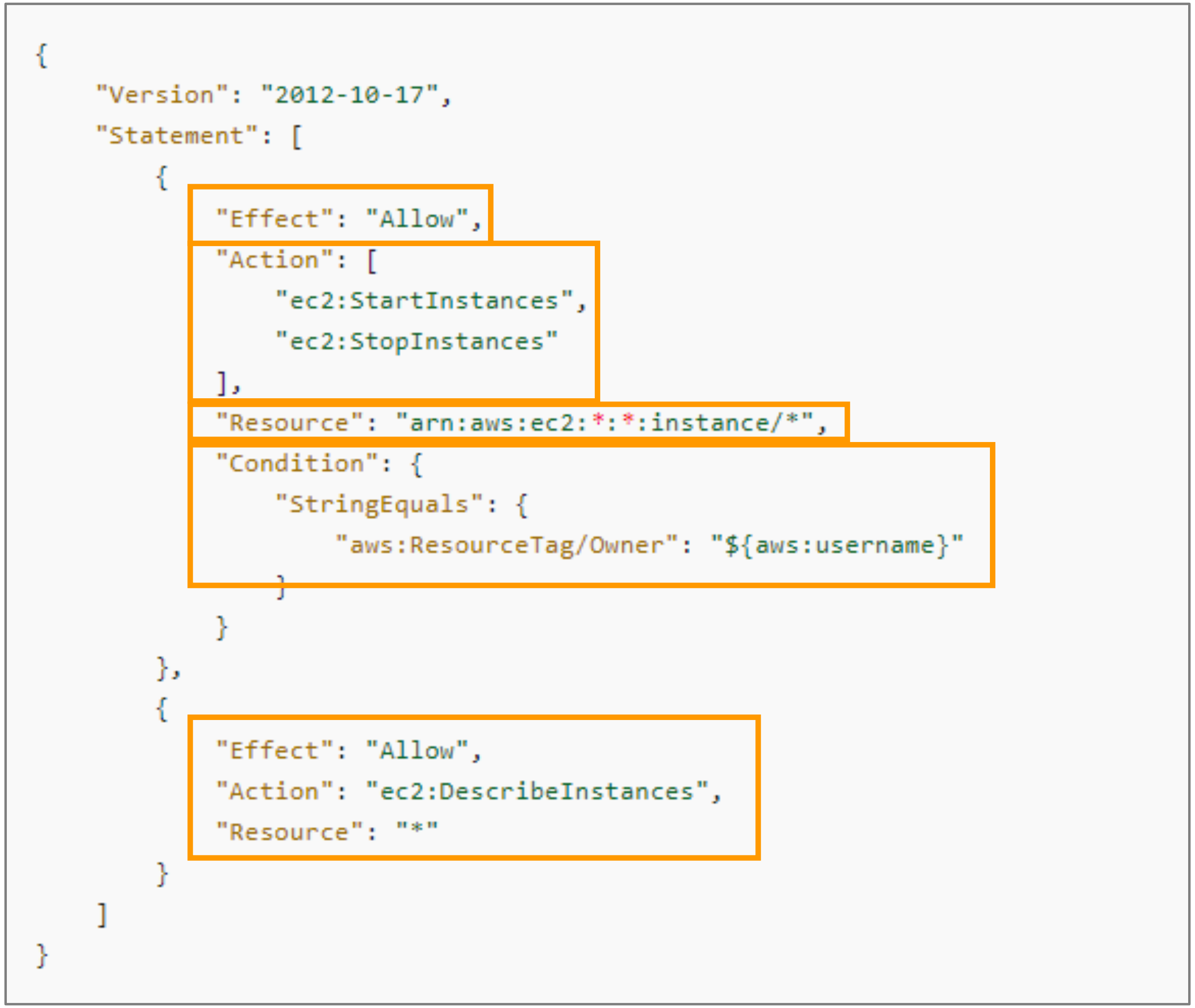
```
{
    "Statement": [
        {
            "Effect":"effect",
            "Action":"action"
            "Resource":"arn"
            "Condition":{
                 "condition":{
                      "key":"value"
                 }
            }
    ]
}
```

Optional conditions to control when this policy is in effect

Example:

```
{ "StringEquals" : { "aws:username" : "johndoe" }}
```

Policy Example



```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:StartInstances",
                "ec2:StopInstances"
            ],
            "Resource": "arn:aws:ec2:*:*:instance/*",
            "Condition": {
                "StringEquals": {
                    "aws:ResourceTag/Owner": "${aws:username}"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": "ec2:DescribeInstances",
            "Resource": "*"
        }
    ]
}
```

I still can’t do anything.

JSON

POLICY

Who can do what to which resources and when

1 Users

2 Groups

3 Roles

4 Policies (and attach them)

The “Who” Identities

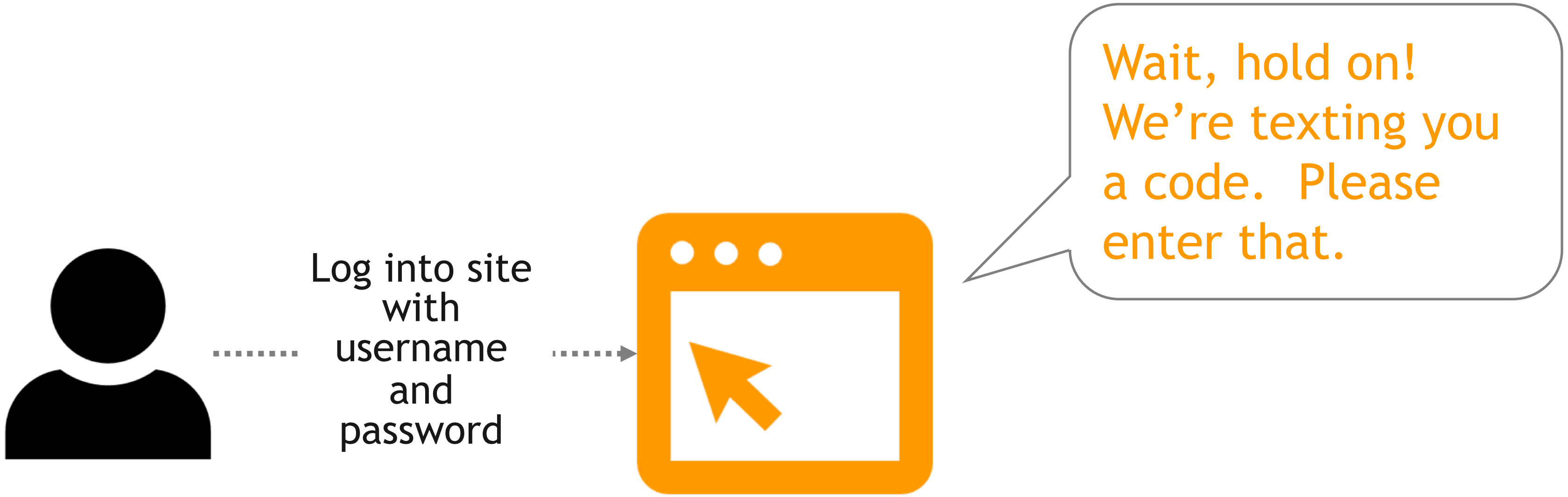
USER

USER GROUP

ROLE

1. Users
2. User Groups
3. Roles
4. Policies (and attach them)

Multi-Factor Authentication (MFA)



Log into site with username and password

Wait, hold on! We’re texting you a code. Please enter that.

Multi-Factor Authentication (MFA)

Two or more “factors” in order to authenticate (i.e., figure out who you are)

- Something you know (e.g., a password)
- Something you have (e.g., a phone or a hardware token)
- Something you are (e.g., fingerprint)

Three Types of MFA Devices

Virtual MFA device for smartphone or tablet

Examples: Google Authenticator, Microsoft Authenticator, Authy

U2F security key

Example: YubiKey

Other hardware MFA device

Example: Gemalto

BEST PRACTICES

Enable MFA for your root account and IAM users aws

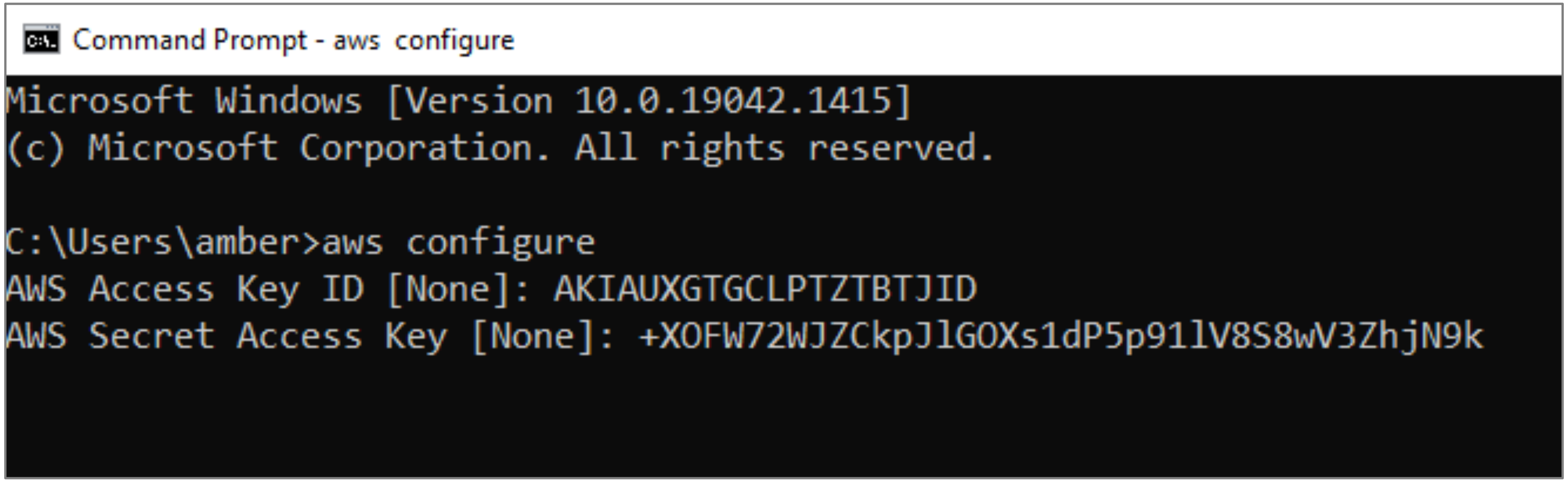
aws certified
Cloud
Practitioner

DEMO

Enabling Multi-Factor Authentication

Access Keys

Using Access Keys for Programmatic Access



```
Microsoft Windows [Version 10.0.19042.1415]
(c) Microsoft Corporation. All rights reserved.

C:\Users\amber>aws configure
AWS Access Key ID [None]: AKIAUXGTGCLPTZTBTJID
AWS Secret Access Key [None]: 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
```

Access Keys

Password Policies

IAM: Best Practices

Use an IAM user for day-to-day work, NOT the root account

Use roles to give permissions to AWS services (e.g., EC2 instances)

Don’t share credentials (user name, password, access keys, etc.)

Assign permissions to groups (made up of users) rather than to individual users

When assigning permissions (policies), give the least amount possible

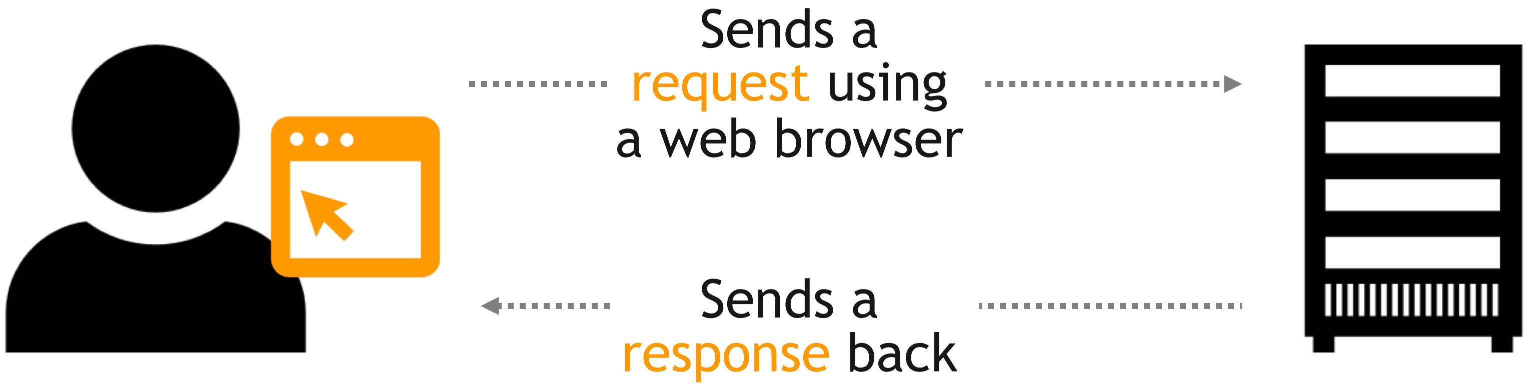
Enforce MFA and strong password policies

Use the IAM Credentials Report to audit permissions

Security and Compliance Services

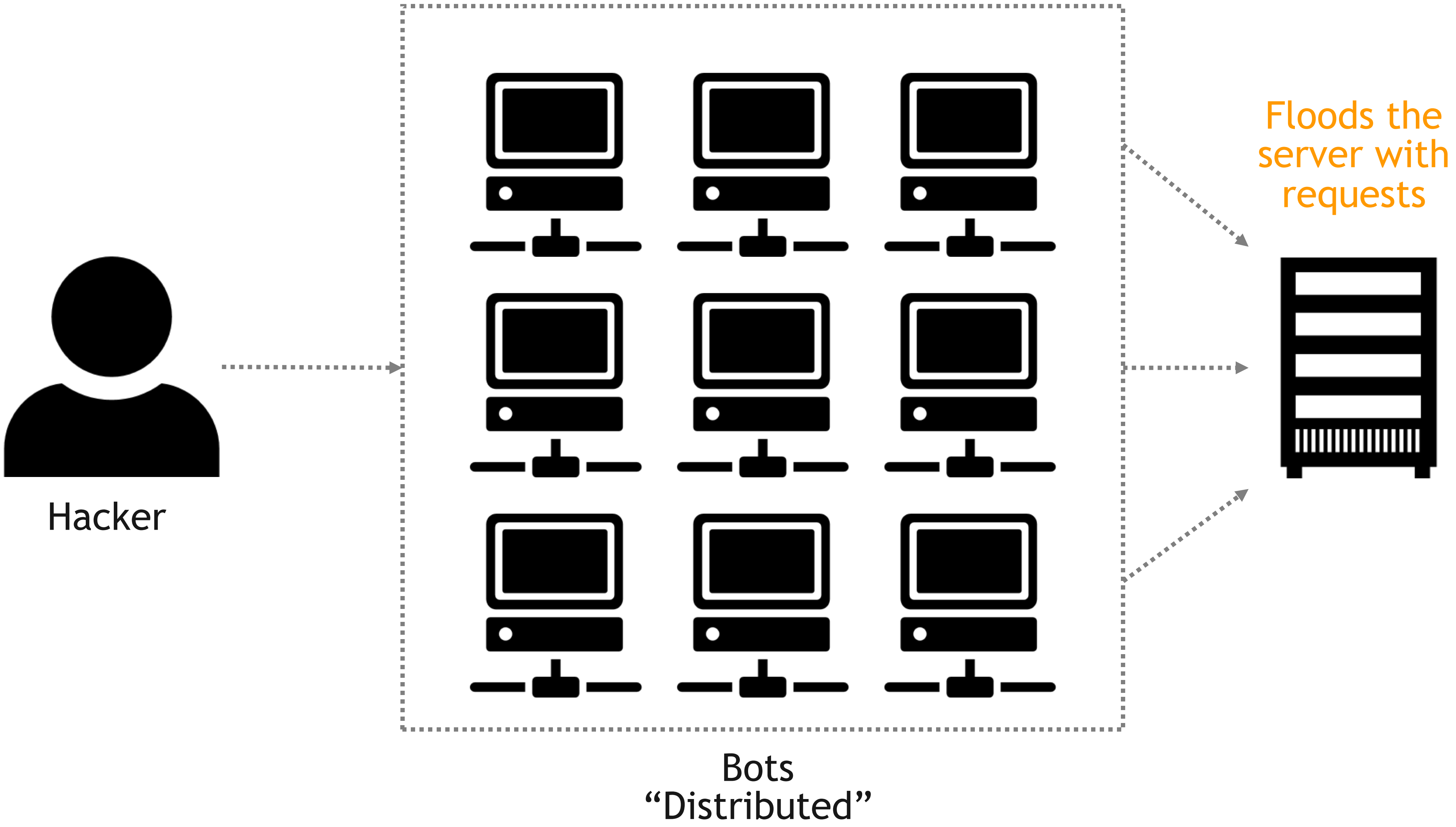
Infrastructure Protection: AWS Shield
Infrastructure Protection: AWS Web Application Firewall (WAF)
Data Protection: AWS Key Management System (KMS) and CloudHSM
Data Protection: AWS Certificate Manager (ACM)
Data Protection: AWS Secrets Manager
Data Protection: Amazon Macie
Detection: Amazon Inspector
Detection: Amazon GuardDuty
Detection: AWS Config
Detection: AWS Security Hub
Incident Response: Amazon Detective
Compliance: AWS Artifact

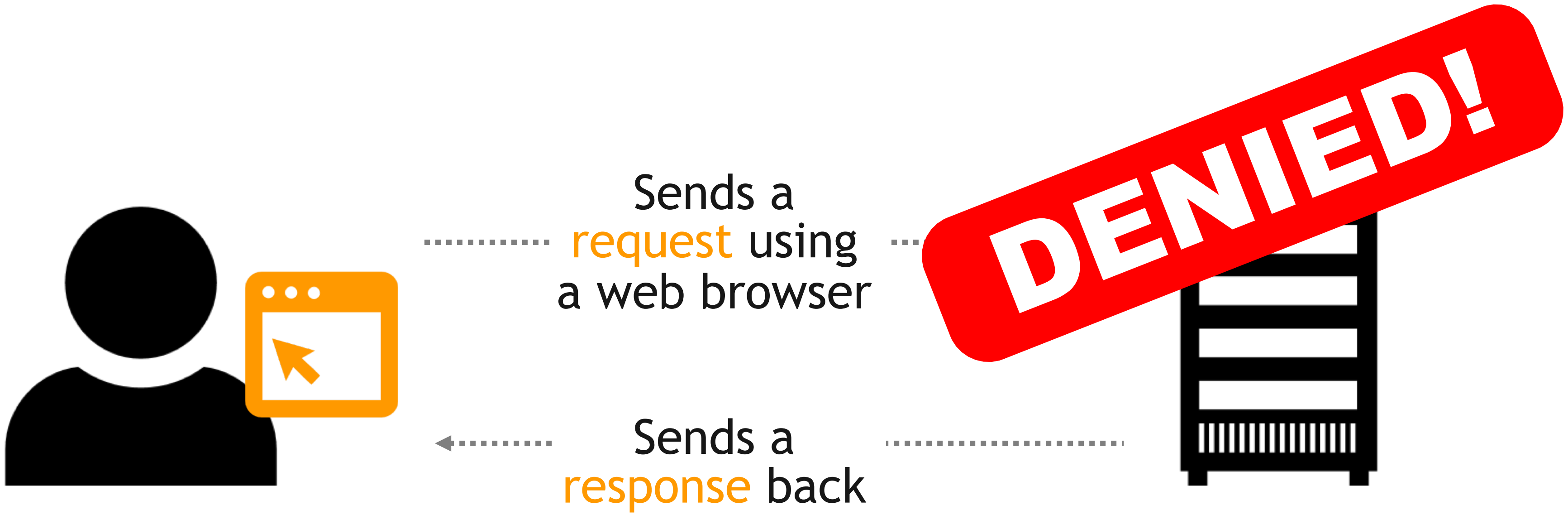
Distributed Denial of Service (DDoS)



Sends a request using a web browser

Sends a response back


Hacker
Bots
“Distributed”
Floods the server with requests



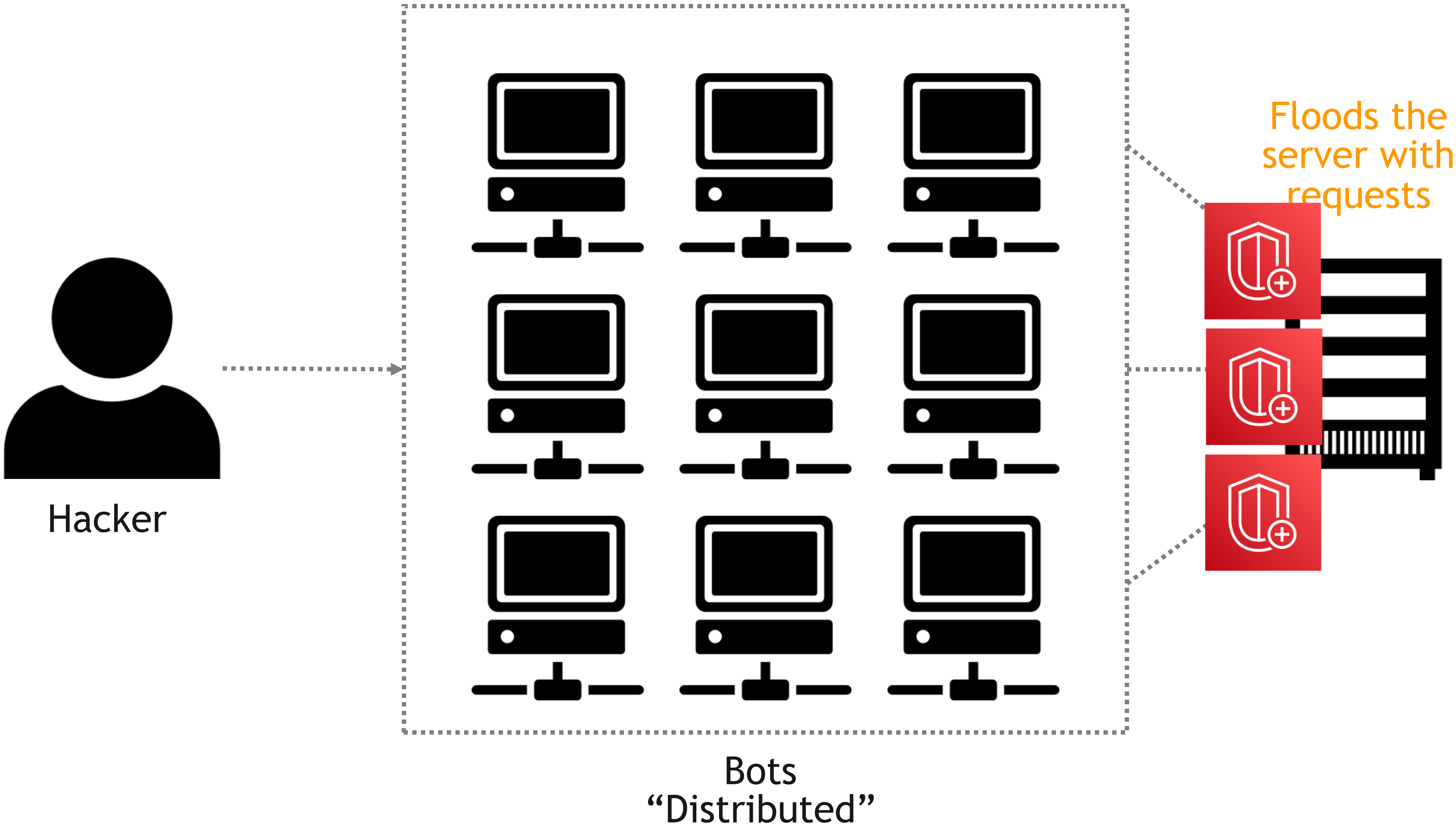
Sends a request using a web browser

Sends a response back

DENIED!

Protecting Against DDoS on AWS

AWS Shield


Hacker
Floods the server with requests
Bots
“Distributed”

AWS Shield

STANDARD

Automatically protects all AWS customers

Free

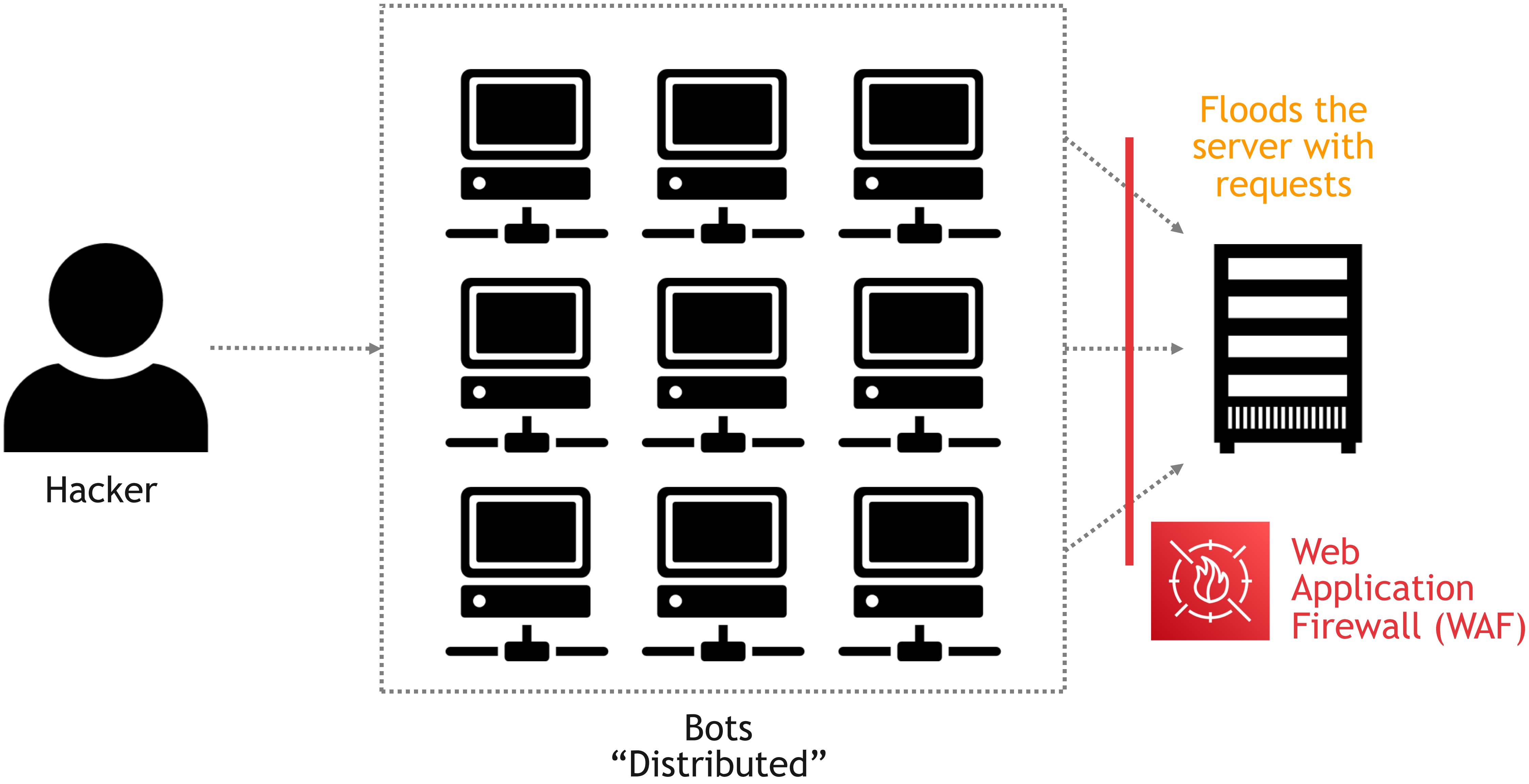
Protects from the most common types of DDoS attacks

ADVANCED

A paid service that protects against more sophisticated attacks

Integrates with other services like CloudFront, Route 53 and Elastic Load Balancing

AWS Web Application Firewall (WAF) included at no extra cost



Hacker
Bots
“Distributed”
Floods the server with requests
Web Application Firewall (WAF)

AWS Web Application Firewall (WAF)

Configure rules

- Allow, block, monitor/count
- IP addresses, country of origin, presence of a script, URL strings, etc.
- Example:
  - Block IP addresses and values in the request that are used by known attackers
  - A specific IP address can only send 100 requests to your application in 5 minutes

ENCRYPTION

UNITED STATES NATIONAL BANK

Two Types of Encryption

AT REST

Data that's stored or archived on a device

Examples:

S3 bucket

Hard disk

Database

IN TRANSIT

Data being transferred from one location to another

Examples:

Moving data from an EC2 instance to an S3 bucket

Moving data from an on-premises data center to AWS

Two Types of Encryption

AT REST

Data that's stored or archived on a device

Examples:

bucket

Hard disk

Database

IN TRANSIT

Data being transferred from one location to another

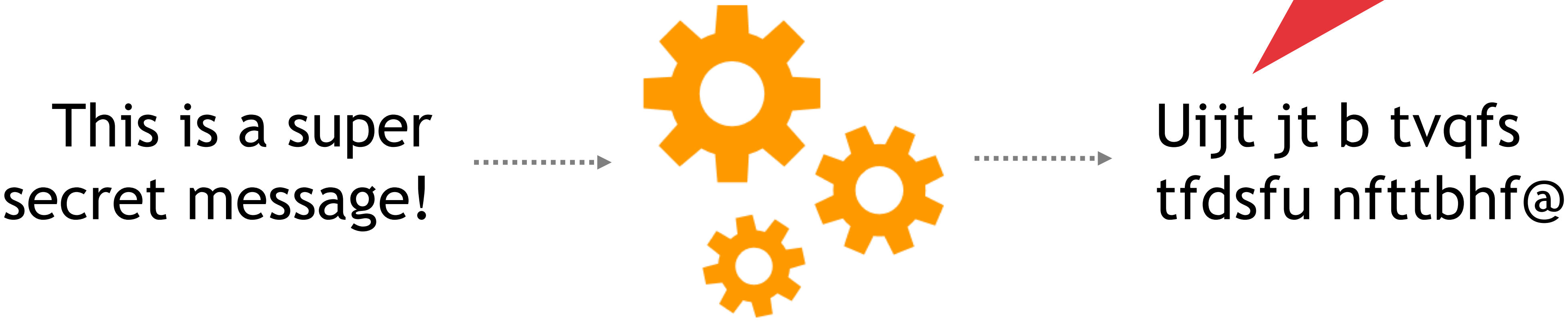
Examples:

Moving data from an EC2 instance to an S3 bucket

Moving data from an on-premises data center to AWS

HOW?

Encryption Keys



This is a super secret message!

Uijt jt b tvqfs tfdsfu nfttbhf@

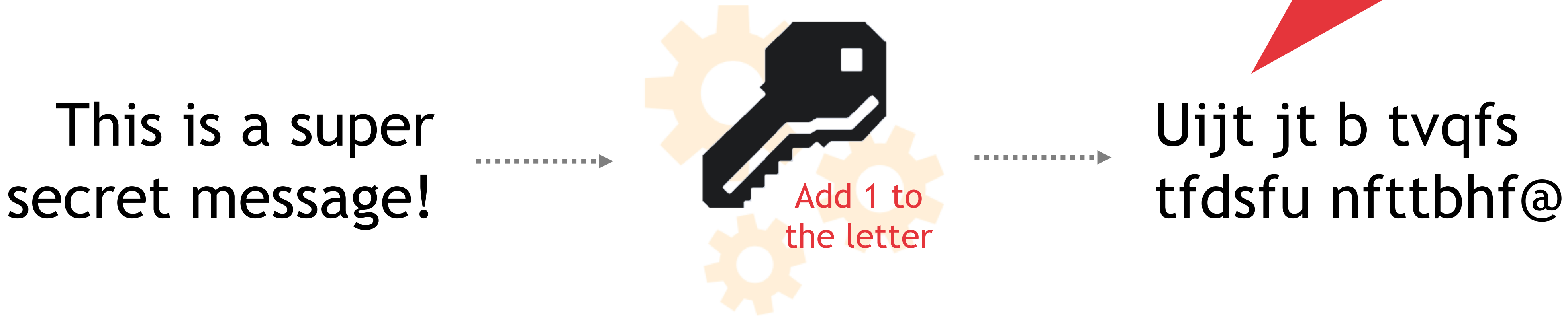
Encrypted = scrambled

This is a super
secret message!

Uijt jt b tvqfs
tfdsfu nfttbhf@

How did we
arrive at this?

How did we arrive at this?



This is a super secret message!

Add 1 to the letter

Uijt jt b tvqfs tfdsfu nfttbhf@

How do we get back to the original message?

This is a super secret message!

Subtract 1 from the letter

Uijt jt b tvqfs tfdsfu nfttbhf@

AWS Key Management System (KMS)

Primary service for encryption in AWS

AWS manages the encryption hardware, software and keys for you

Integrated with many AWS services, including EBS, S3, Redshift and CloudTrail

- Example: I want to encrypt a document stored in an S3 bucket

FIPS 140-2 Compliance: **Level 2 overall** (3 in some areas)

AWS CloudHSM

Hardware Security Module

AWS provisions the hardware and you do everything else

- AWS cannot access your keys
- AWS cannot recover your keys

Integrated with a limited number of other AWS services

FIPS 140-2 Compliance: **Level 3** (considered more secure)

Types of Keys

AWS MANAGED

AWS creates and manages

Used by AWS services

- aws/lambda
- aws/cloud9
- aws/s3

CUSTOMER MANAGED

You (customer) create and manage

Can create policies to rotate keys

Specify who can use and manage the keys

Supports “bring your own key”

CUSTOM KEY STORES

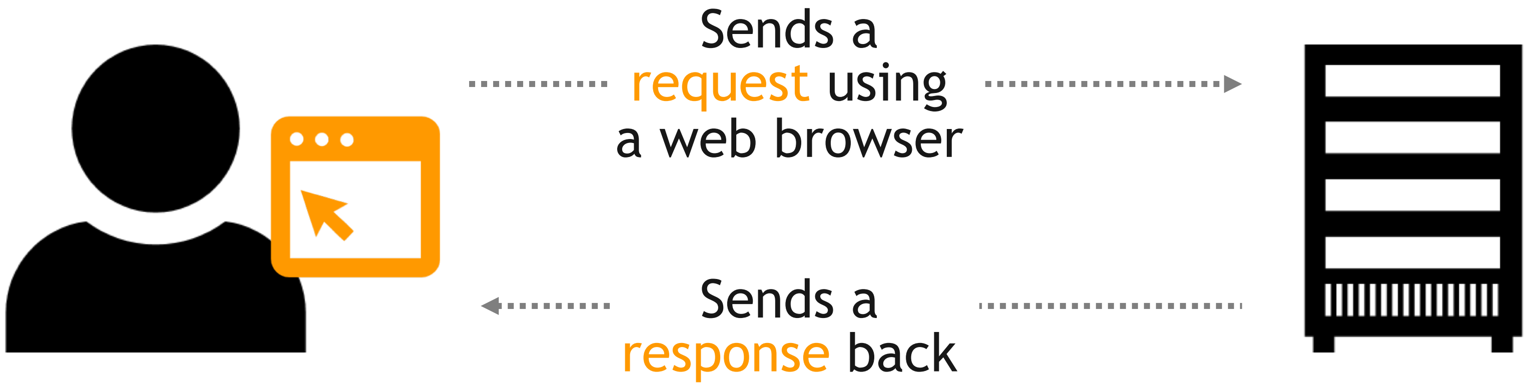
Created with CloudHSM

You own and manage

DEMO

Working with Keys in AWS KMS

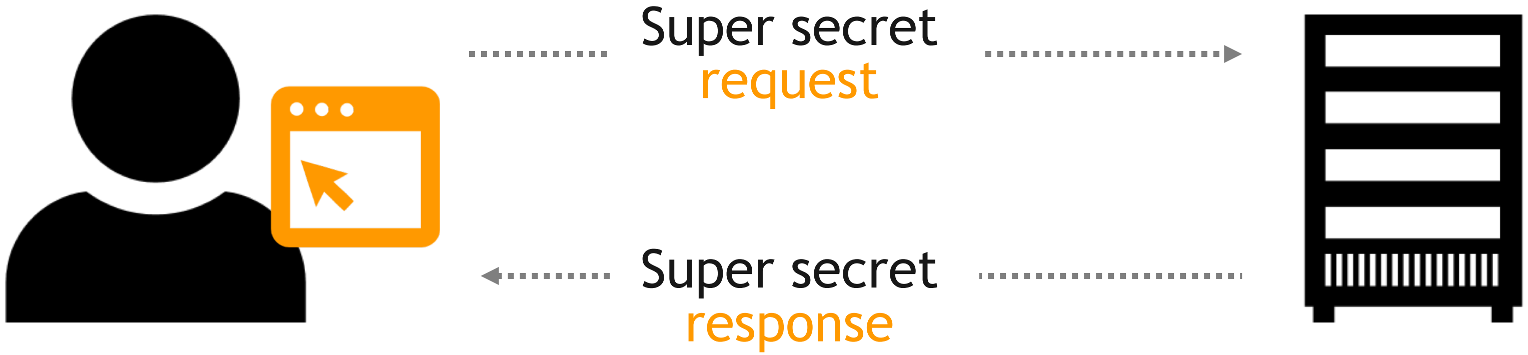
Understanding Certificates



Sends a request using a web browser

Sends a response back

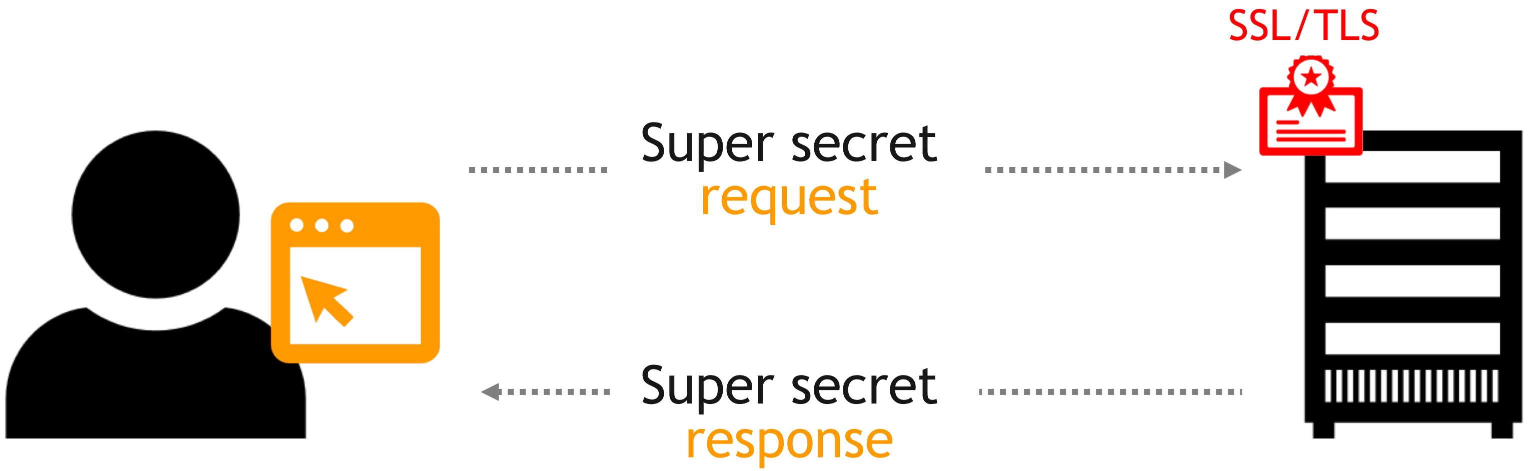
Understanding Certificates



Super secret request

Super secret response

Understanding Certificates



SSL/TLS

Super secret request

Super secret response

1) Identifies the server as reputable
2) Ensures communication between us is encrypted

AWS Certificate Manager (ACM)

Provision, manage, and deploy public and private SSL/TLS certificates

- Public = for resources on the public internet (these certificates are free)
- Private = for resources on private networks

Loads certificates on:

- API Gateway
- Elastic load balancers
- CloudFront distributions

[illegible]

Store credentials on file system

AWS Secrets Manager

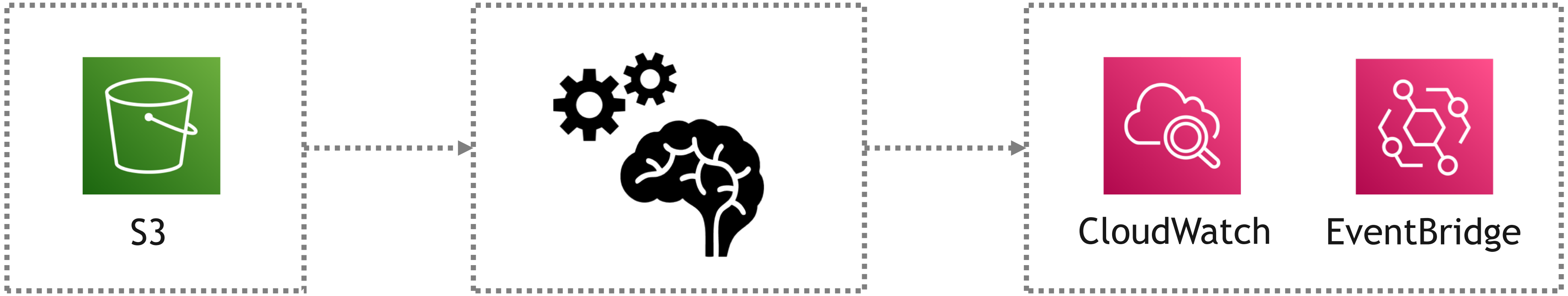
The recommended way to protect secrets (e.g., user names and passwords) needed by your applications and services

DEMO

Working with AWS Secrets Manager

Login details
Date of birth
Social security number
Email address
Credit card numbers
Cookies
Full name
Telephone number
Passport number
Driver's license number
Device serial number
Home address
Owned properties
Personally Identifiable Information (PII)

Amazon Macie



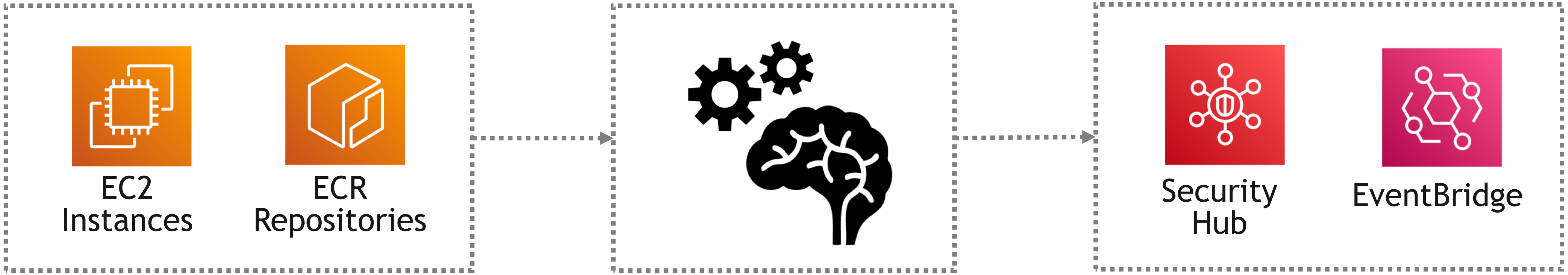
Automatically inventories S3 buckets

Identifies and analyzes PII data using machine learning and pattern matching

Uses findings to automate workflows and remediation

AMAZON INSPECTOR

Amazon Inspector



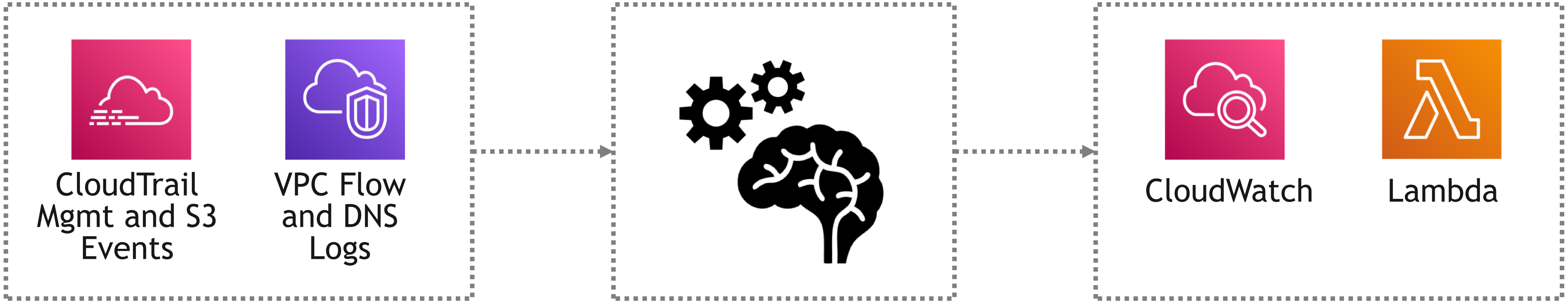
Automatically detects and scans for software vulnerabilities and network exposure

Makes sense of the findings and assigns a risk score

Uses findings to automate workflows and ticketing

AMAZON GUARDDUTY

Amazon GuardDuty



CloudTrail Mgmt and S3 Events

VPC Flow and DNS Logs

CloudWatch

Lambda

Continuously analyzes network, account and data access

Using machine learning, identifies and prioritizes potential threats

Uses findings to automate workflows and remediation

AWS Config

Inventory, record and audit the configuration of your AWS resources

Example use cases:

- Inventory all your S3 buckets, and when one of them becomes publicly accessible, receive an alert
- Receive an alert when an unauthorized port opens on a security group
- During a compliance audit, show when configurations changed

DEMO

Working with AWS Config

AWS Security Hub

Pulls everything together into a consolidated place where you can view and take actions on security issues

- Requires AWS Config
- Cross-account
- Aggregates data from GuardDuty, Inspector, Macie, IAM Access Analyzer, Systems Manager and Firewall Manager

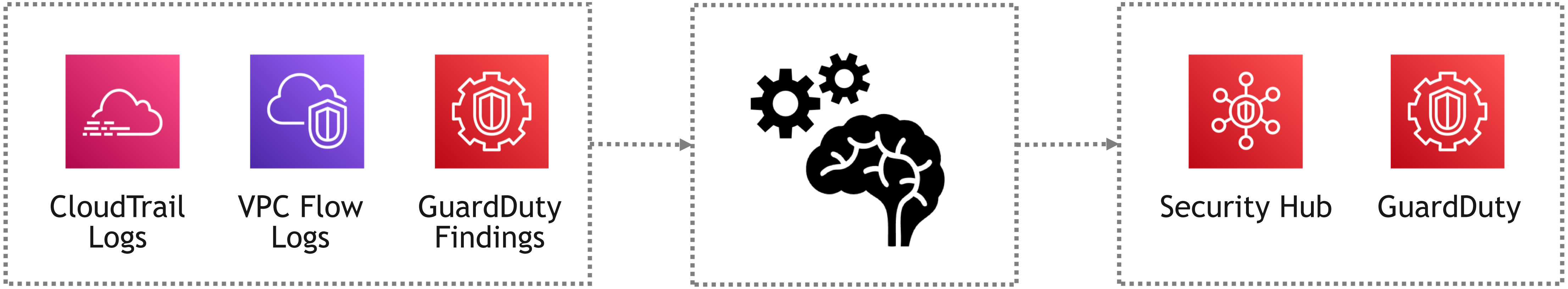
DEMO

A Tour of AWS Security Hub

AMAZON DETECTIVE

Amazon Detective

Finding Root Cause Quickly



CloudTrail Logs

VPC Flow Logs

GuardDuty Findings

Security Hub

GuardDuty

Automatically distills and organizes data into a graph model

Builds a linked set of data using machine learning, statistical analysis and graph theory

Provides visualizations, context and detailed findings to help get to the root cause

AWS Artifact

Self-service portal to access AWS's internal compliance reports and agreements

Free

DEMO

A Tour of AWS Artifact

Important Points to Remember

SHARED RESPONSIBILITY MODEL

- AWS is responsible for security OF the cloud
- Customer (you) are responsible for security IN the cloud

IDENTITY AND ACCESS MANAGEMENT (IAM)

- Root account has permissions to do everything, including access to billing info
    - Do NOT use it for everyday work; use an IAM user instead
- By default, IAM users have no permissions
- Multi-factor authentication (MFA) can be enforced for the root account and individual users
- Access keys are required for programmatic access (CLI, SDK)
- Roles should be used to give permissions to other AWS services
- Permissions are controlled by policies; give least privileges

Important Points to Remember

SERVICE	PRIMARY FUNCTION	POINTS TO REMEMBER
AWS Shield	Infrastructure Protection	• Protects against DDoS attacks
AWS Web Application Firewall (WAF)	Infrastructure Protection	• Controls incoming and outgoing traffic for applications and websites • Based on rules like, “Block traffic from IP address X”
AWS Key Management System (KMS)	Data Protection	• Primary service for encryption in AWS • AWS manages the encryption hardware, software and keys for you
AWS CloudHSM	Data Protection	• AWS provisions the hardware and you do everything else
AWS Certificate Manager (ACM)	Data Protection	• Provision, manage and deploy SSL/TLS certificates
AWS Secrets Manager	Data Protection	• Securely store and rotate secrets, such as a database name/password
Amazon Macie	Data Protection	• Scans S3 for personally identifiable information (PII)

Important Points to Remember

SERVICE	PRIMARY FUNCTION	POINTS TO REMEMBER
Amazon Inspector	Detection	• Monitors EC2 instances and ECR repositories for software vulnerabilities and network exposure
Amazon GuardDuty	Detection	• Monitors AWS accounts, network and S3 for malicious activity
AWS Config	Detection	• Inventory of resources and recording of configuration/changes
AWS Security Hub	Detection	• Consolidated view of all things security (pulls from many other services into a dashboard) • Works for multiple accounts
Amazon Detective	Incident Response	• Used to quickly get to the root cause of security issues
AWS Artifact	Compliance	• View AWS's internal compliance reports and agreements