ACADEMIND

Get Certified!

Start Today!

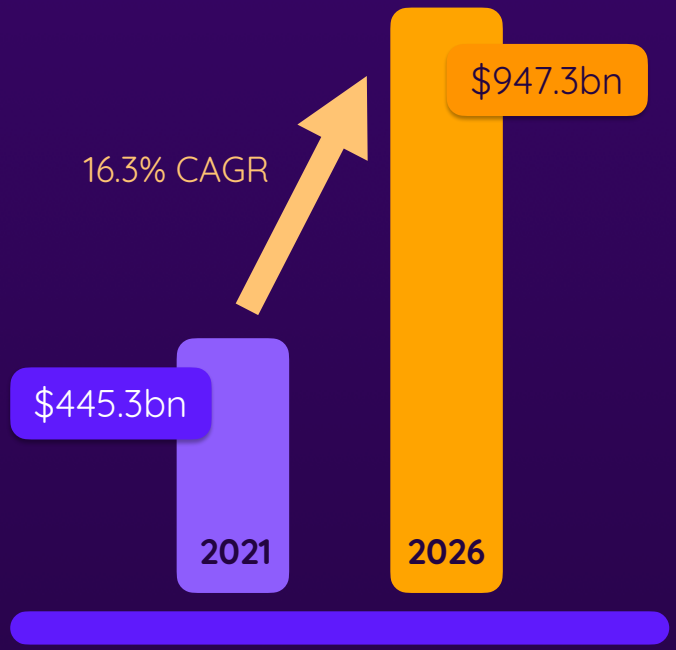
AWS

ACADEMIND

AWS leads the market

Gartner Magic Quadrant for Cloud Infrastructure and Platform Services
(https://www.gartner.com/doc/reprints?id=1-2710E4VR&ct=210802, 2021)

Strong cloud computing forecasts



MARKETSANDMARKETS (2021)
(https://www.marketsandmarkets.com/Market-Reports/cloud-computing-market-234.html)

ACADEMIND

- All key AWS concepts & services explained from the ground up
- No prior AWS or cloud computing knowledge required
- Basic IT knowledge suffices

ACADE MIND

Let’s succeed together!

ACADEMIND

Try the course
risk-free!

ACADEMIND

Two Options

Get started with this course

Refresh your AWS knowledge

What Is AWS?

Amazon Web Services

A subsidiary of Amazon (amazon.com)

A Cloud Services Provider

What Is “Cloud Computing”?

Example Time

Host a Website

Run Web Server Software
(which serves the site)

Without Cloud Computing

Build & run your own server
(your own computer)

With Cloud Computing

Rent a remote server & install + run your software on that

What Is “Cloud Computing”?

AWS-managed infrastructure & services granting access to that infrastructure

Machines & data centers are “in the cloud” because, as a customer, you don’t need to worry about infrastructure at all

Send commands like “start a new server” or “create a new managed database”

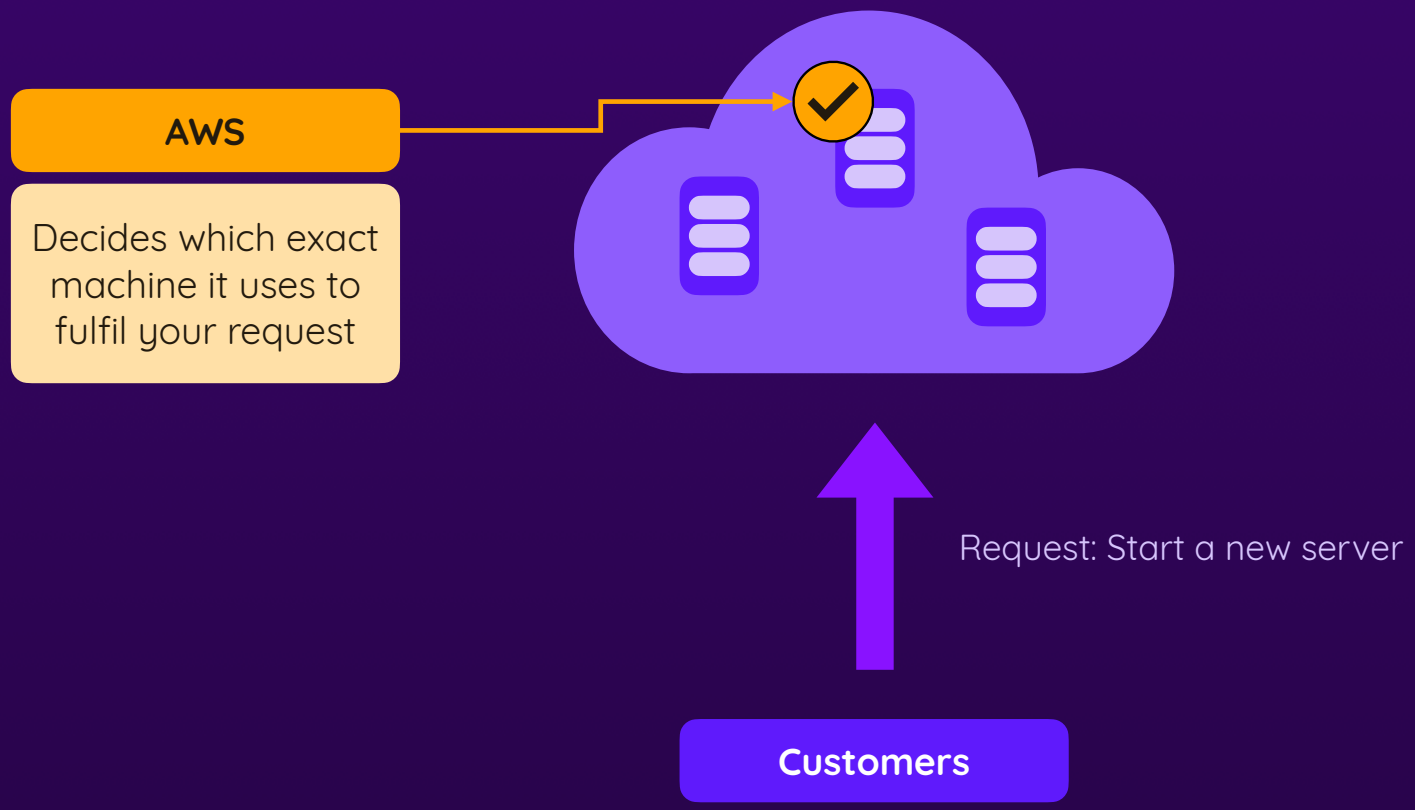
➜ Use AWS’ infrastructure and run your workloads on that managed infrastructure

Customers

AWS Customers

As an AWS customer, you only worry about your workloads, **not about the management of any hardware or infrastructure**

AWS Does The Heavy Lifting



What Is AWS?

Amazon Web Services

A Cloud Services Provider

What Is “The Cloud”?

Cloud Computing

On-demand usage & delivery of IT resources

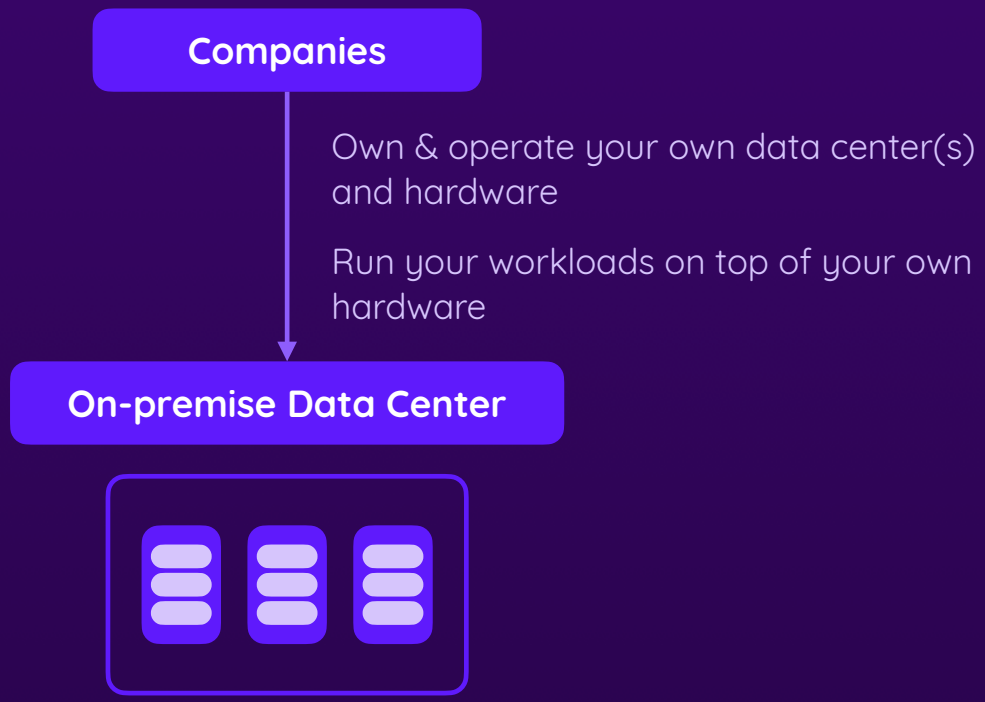
Examples
Rent a server & serve a website
Rent file storage volumes

Flexible usage of compute power & data storage capabilities

No need to provision and maintain your own data center

Only pay for the resources you use, when you use them

Without Cloud Computing



On-premises Advantages & Disadvantages

You have **full control** over your physical infrastructure & hardware

You **know exactly where** your computers (and data) are

You are **responsible for maintaining** the infrastructure

You are responsible for (long-term) **capacity planning & upgrading**

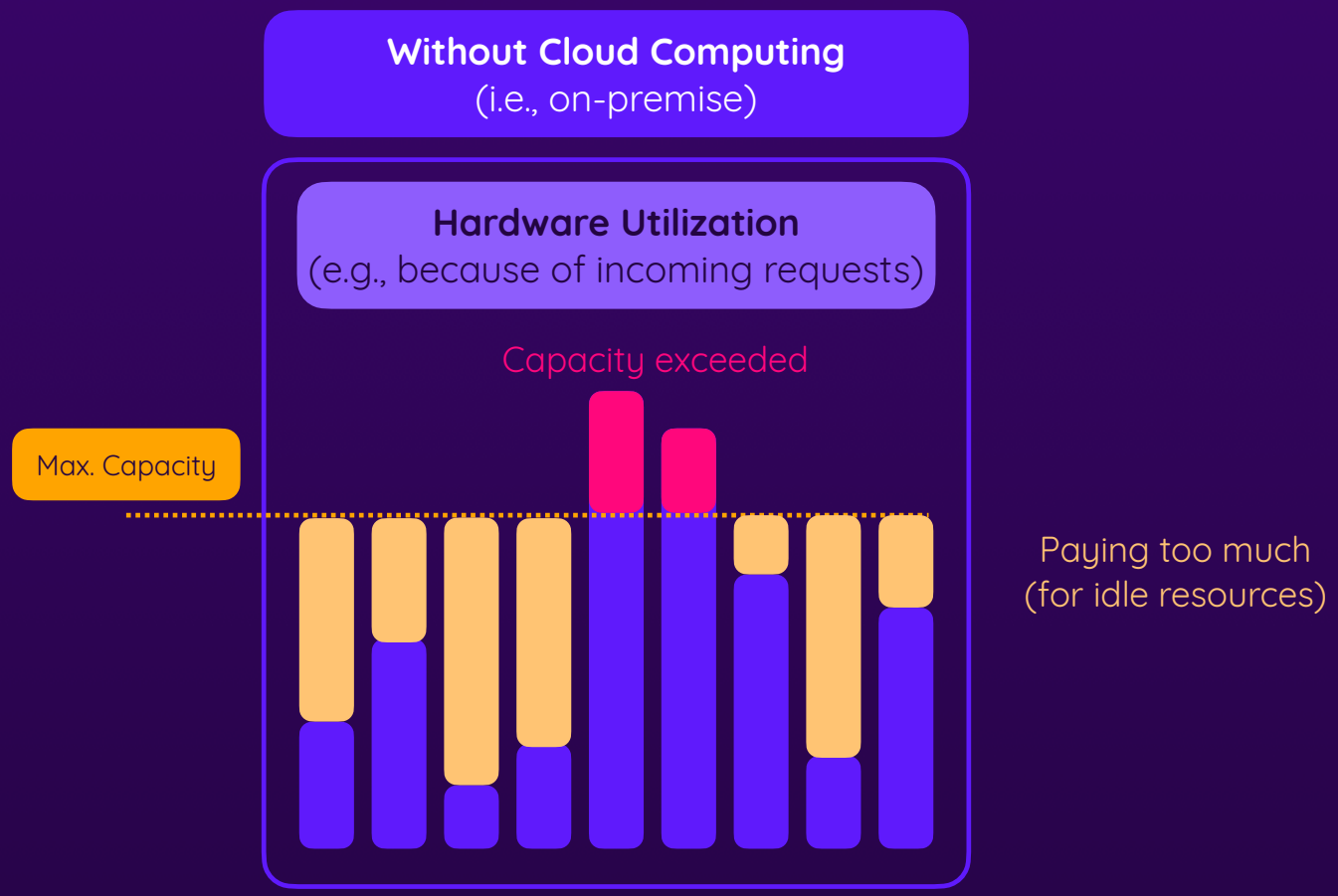
You are responsible for **securing** the infrastructure

You **can't react quickly** to workload spikes (e.g., more requests)

You also **pay for idle resources**

You are typically stuck to **one or a few locations**

Flexible Scaling Required!



Cloud Computing To The Rescue

AWS is responsible for operating & maintaining the infrastructure

- Maintenance
- Capacity planning & upgrades
- Security

Reliability	Agility, Elasticity & Scalability	Pay-as-you go	Global Reach & High Availability
AWS SLAs & global reach	Scale up or down as needed, anytime	Only pay for services used	Choose perfect location
Tools & services for building reliable solutions	Instant access to services & resources	Don't pay for services you're not using (anymore)	Spread our workloads to ensure high availability

Reliability

Rely on AWS

AWS is responsible for managing the underlying infrastructure

Service Level Agreements (SLA) are available for many key services

Build Reliable Applications

Various services help you build reliable applications

Your workload performs its intended function correctly & consistently

Rely on Global Reach

Fall back to other regions or go with a multi-region setup right from the start

Move your workload within minutes to hours, instead of days or weeks

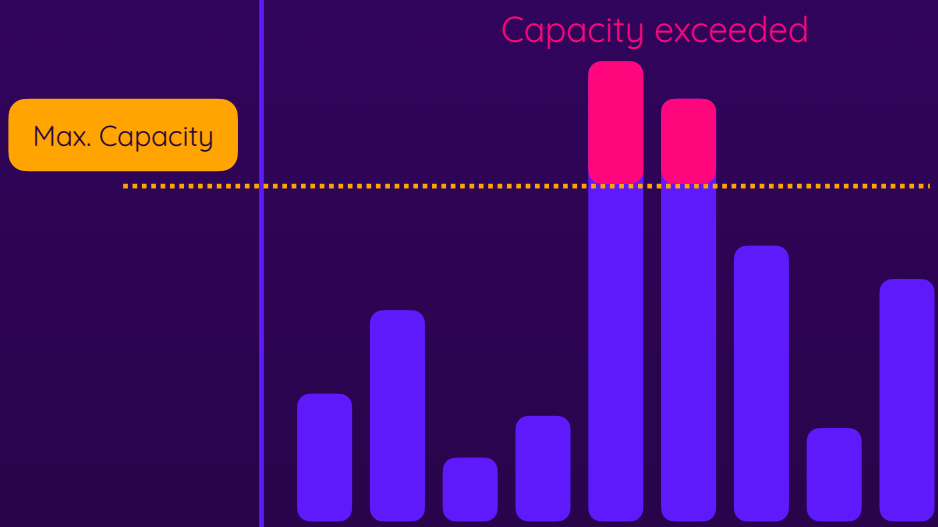
Agility, Elasticity & Scalability

Agility	Elasticity	Scalability
Use cloud resources within seconds or minutes	Start using more or less resources, just as needed	Scale up or down, as required
Launch resources with a click or command	No long-term planning required	Use auto-scaling services to reduce manual workload

Flexible Scaling Required!

Without Cloud Computing
(i.e., on-premise)

Hardware Utilization
(e.g., because of incoming requests)



With Cloud Computing
(e.g., via AWS services)

Hardware Utilization
(e.g., because of incoming requests)

Pay-as-you Go

You only pay for what you use

There are a few upfront or subscription payment models

No Usage	Average Usage	Peak Usage
$0	$50	$150

Cost-Related Benefits

You're **trading fixed expense for variable expense**

No **CapEx** (capital expenditure) for purchasing or operating your own hardware

Less **OpEx** (operating expenditure) since you only pay for the service usage, not for staff or power

Benefit from AWS' **economies of scale**: AWS can realize discounts & savings on hardware procurements (+ other advantages) which you couldn't

Global Reach & High Availability

AWS own & operates a world-wide network of data centers

If one data center or group of data centers would go down, you can run your workloads in one of the many other regions

AWS' World-Wide Infrastructure

AWS operates data centers all over the world

Region

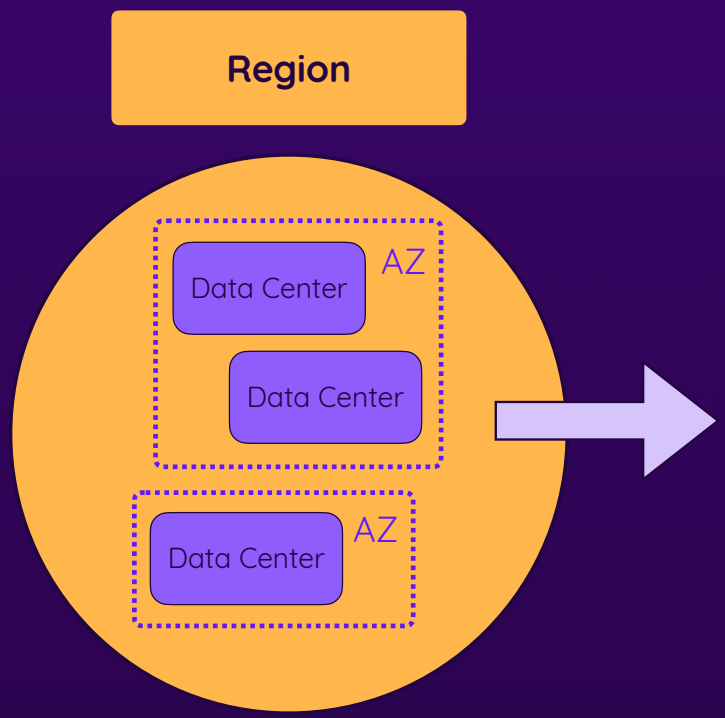
One region contains multiple data centers

!

Regions are physically **isolated** from each other

For many services, you can **choose** in which region it should run

Regions & Availability Zones (AZs)



One region contains **at least <u>three</u>** AZs

Data centers are organized (grouped) into **Availability Zones (AZs)**
AZs are also separated from each other

One AZ contains **at least <u>one</u>** data center

There also are "**Local Zones**", "**Edge Locations**", "**Wavelength Zones**" & "**AWS Outposts**"

Reasons For Picking A Certain Region

Different Pricing	Service Availability	Legal Reasons	Availability & Latency
AWS faces different costs for operating its infrastructure in different places of the world	Not all services can be used in all Regions	Companies might be legally required to use certain services in certain Regions only	Workloads can be executed in multiple Regions to increase availability & reliability
As a result, service prices can differ between Regions	Some services are only available in certain Regions	Example: A company must store user data in the EU	Applications can be run close to end users / customers to reduce latency
You can use the service pricing pages or the "Pricing Calculator" to learn about pricing differences			

AWS' World-Wide Infrastructure

AWS also operates a world-wide network to connect all their regions

Self-Service & Managed Services

AWS offers two main kinds of services

Self-service	Managed
Do it yourself	AWS manages the hard parts
Fully configurable	Partially configurable
Underlying hardware is still managed by AWS	Can be used together with other services (incl. self-service)

ACADEMIND

Which Services Does AWS Offer?

- Compute
- Data Storage
- Database
- Networking & Content Delivery
- Application Integration
- Security
- Cloud Management
- Migration & Edge Computing
- Analytics & Data Ingestion
- Machine Learning & Artificial Intelligence
- Developer Tools
- Business Applications

ACADEMIND

Which Services Does AWS Offer?

- Compute
- Data Storage
- Database
- Networking & Content Delivery
- Application Integration
- Security
- Cloud Management
- Migration & Edge Computing
- Analytics & Data Ingestion
- Machine Learning & Artificial Intelligence
- Developer Tools
- Business Applications

About This Course & The Exam

This course has **two goals**

- Help you pass the exam
- Give you a solid AWS foundation

ACADEMIND

Accessing & Using AWS

How to use your account

- Different Ways of Accessing & Using AWS Services
- Service Pricing & Cost Management
- Getting Help

AWS Can Cost You Money!

AWS is a for-profit business

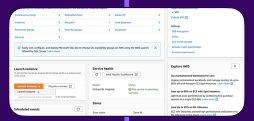
They charge you money for using their services

But: Many services can be used for free within the first 12 months after creating an account ("**AWS Free Tier**")

More details will follow in later lectures!

Different Ways of Accessing AWS

everything is an API call!



```
ec2 run-instanc
```

```
var ec2 = new AWS.EC2();
ec2.applySecurityGroupsTo
  if (err) console.log(er
  else     console.log(da
});
```

```
https://ec2.amazonaws
&InstanceId.1=i-12345
&AUTHPARAMS
```

Management Console	AWS CLI	SDKs	API Calls
Easy to access & navigate	Command-driven access	Programmatic access	Use services via HTTP requests
No setup needed	Prior installation & configuration needed	Infrastructure as code	Request configuration can be challenging
Perfect for exploring services & features	Perfect for executing well-known tasks	Perfect for automating tasks	Typically, you'd go for one of the other solutions
Complex, repeated or large-scale setups can become cumbersome	Can simplify complex, repeated or large-scale setups	Can simplify setups & allows for building (automated) tools	

AWS Pricing & Cost Management

AWS offers a variety of pricing & payment models
(depending on the actual service used)

In Advance	On Demand	Subscription
Pay for reserved capacity or min. usage	Most common pricing model	Pay a flat monthly / yearly fee
Discounted price	Only pay for the resources used	Not related to usage
Only offered for some services or service features	Which resources & factors are charged depends on the actual services used	Not very common, but offered for some services / service features

AWS Pricing & Cost Management

Check the pricing pages & information of the specific service(s) you plan to use

Keep an eye on your monthly invoice (it's updated daily!)

Use advanced cost control features

- Budgets
- Alarms
- Cost Explorer

AWS Support & Status

AWS Support

AWS offers different support levels

Service Health Dashboard

General AWS services health & personal health dashboard

Service Documentation

Learn more about service usage & configuration with help of AWS' official online documentation

Summary

AWS Account Access	Support & Help	Pricing & Cost Management
Management Console (Web UI)	Different (paid) support levels	Check pricing on service documentation pages
AWS CLI	Health dashboards	Free tier for many services (within first 12 months)
AWS SDKs	Service documentation pages	Cost explorer & billing dashboard
		Use budgets & alarms

ACADEMIND

Getting Started with AWS Security

Accounts, Authentication & Service Protection

- The AWS Security Model
- Managing Accounts & Authentication
- Understanding Permissions & Access Control

AWS Security Model

Shared Responsibility Model

Security is a shared responsibility

AWS

- Secure the infrastructure & physical machines
- Protect their internal systems & processes
- Secure managed services & everything you can't configure or control

You

- Secure your applications & workloads
- Protect your account & control account access
- Secure everything you are able to control & configure

Protecting Your Account

Secure Credentials

- Choose a strong password
- Change it frequently
- Don’t share your credentials!

Multi-Factor Authentication

- Enable MFA
- Use a digital or physical solution

Utilize IAM Users

- Create IAM users for accessing your account
- Every person (e.g. colleague) should use a separate user

IAM

What Is IAM?

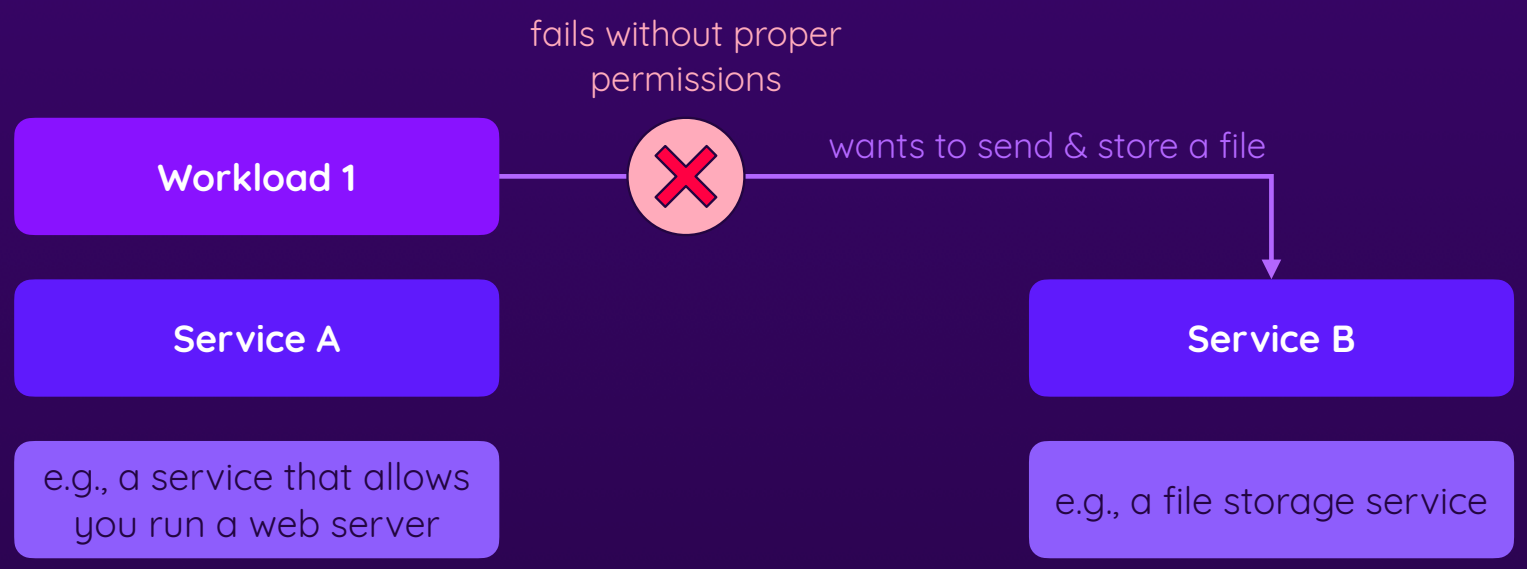
Identity & Access Management

Identities	+	Access Management
The **entity** for which access rights / permissions are controlled		The permissions that are granted (or not granted) to an entity
Who is allowed (or not allowed) to do something?		**What** is an entity allowed to do (with a given service)?
Users, User Groups & Roles		**Permissions**, managed via **Policies**

Users, User Groups & Roles

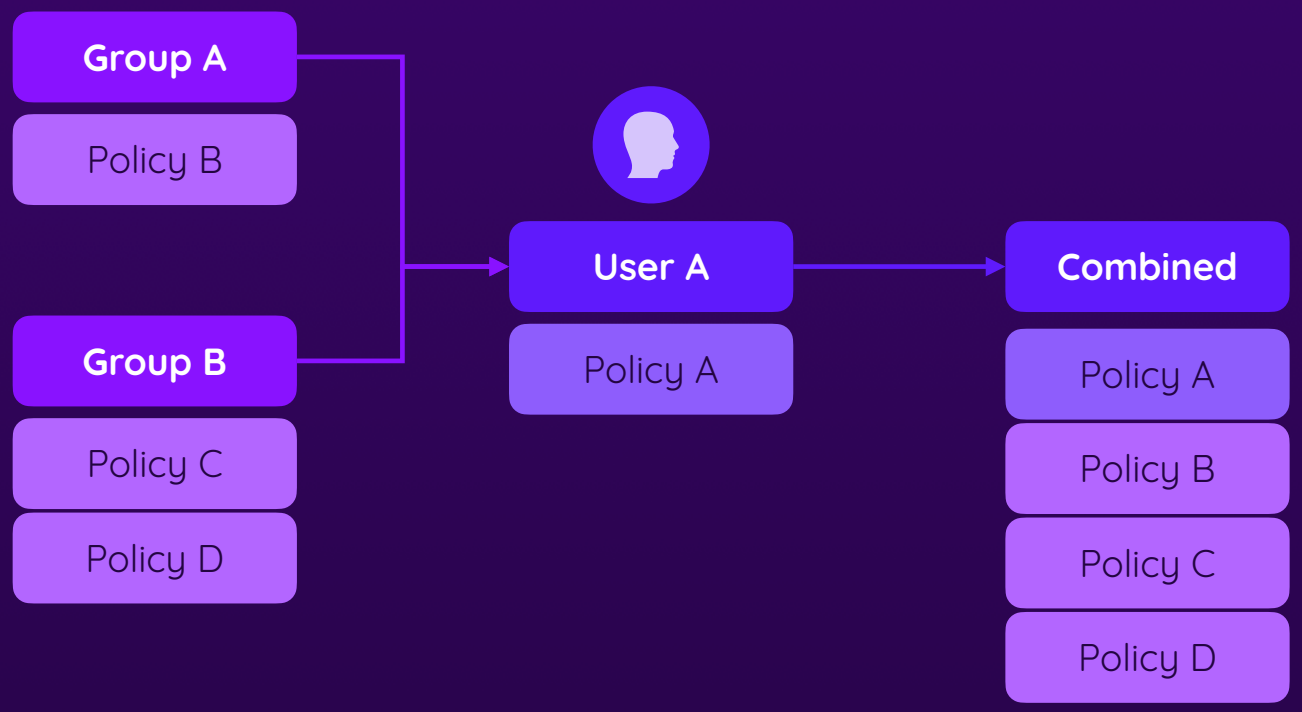
Users	User Groups	Roles
Typically assigned to humans	Group users to share permissions	Typically used by services
Every person that should be able to access the AWS account gets a user	Avoid unnecessary permission copying or tedious per-user access management	Allows services to perform AWS tasks (e.g., start or use another service)

Understanding Roles

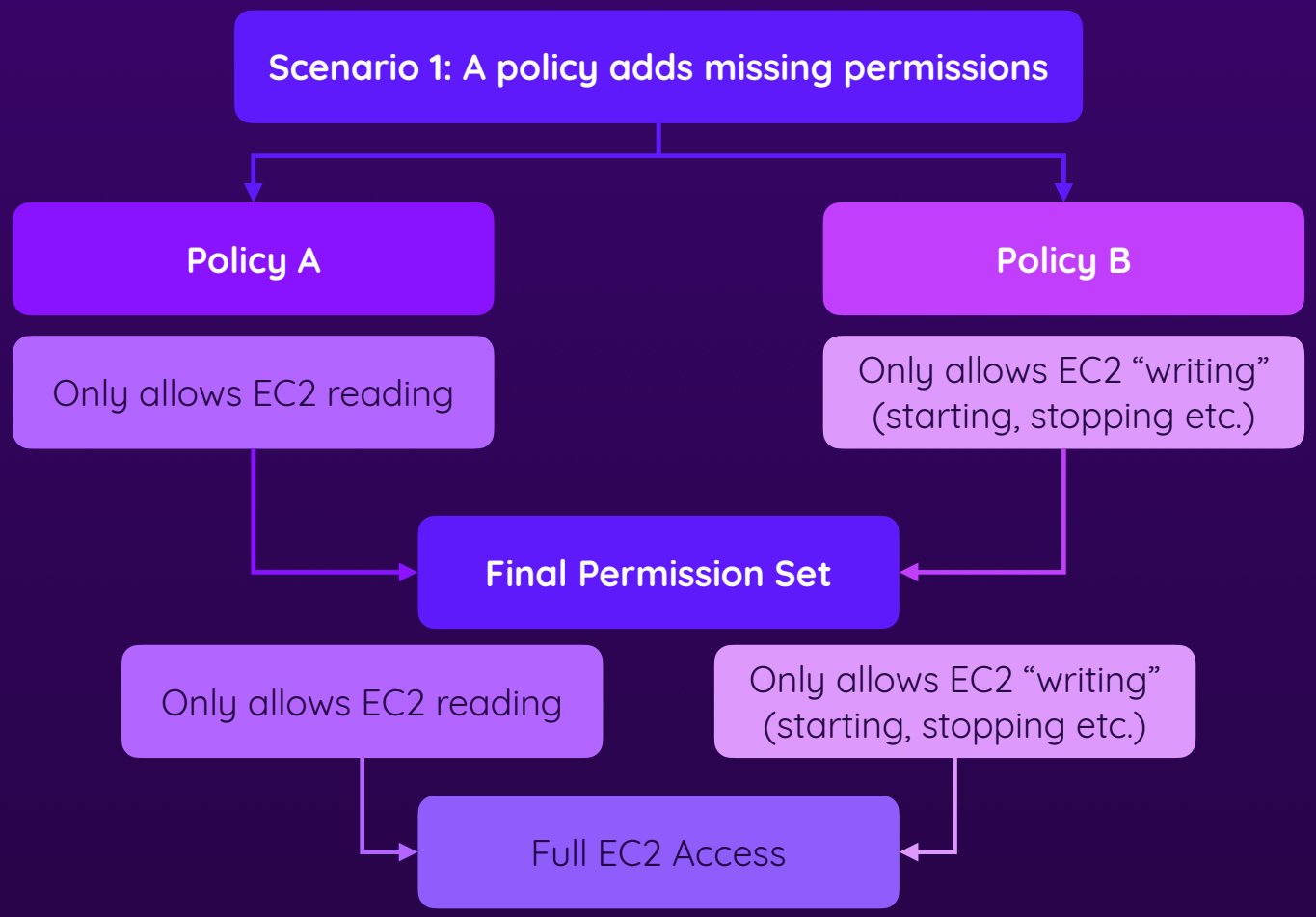


Understanding Roles

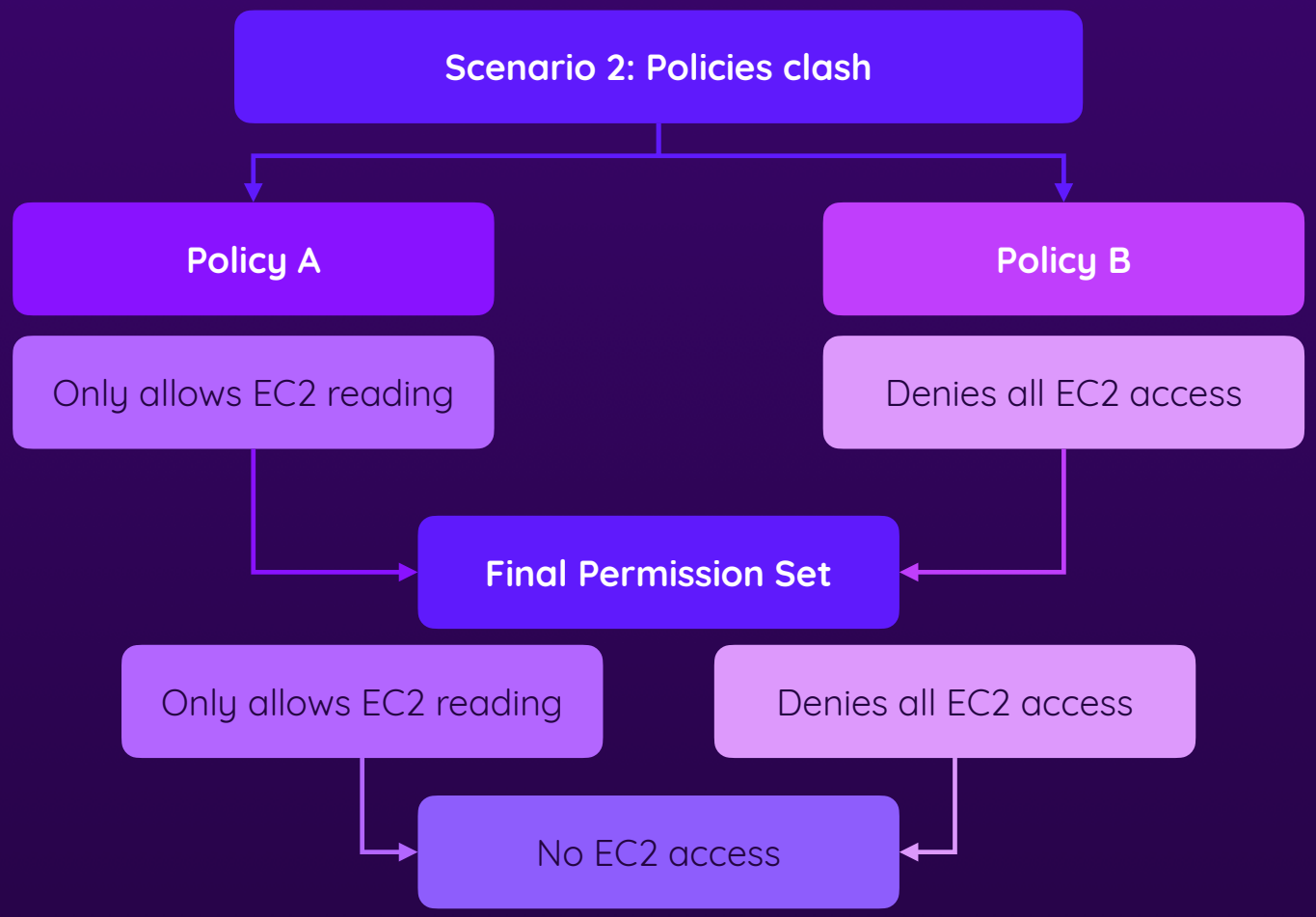
Policies Are Combined



What Happens If Permissions Clash?



What Happens If Permissions Clash?



Explicit DENY Statements Always Win!

```
{
  "Effect": "Allow",
  "Action": "ec2:*"
}
```

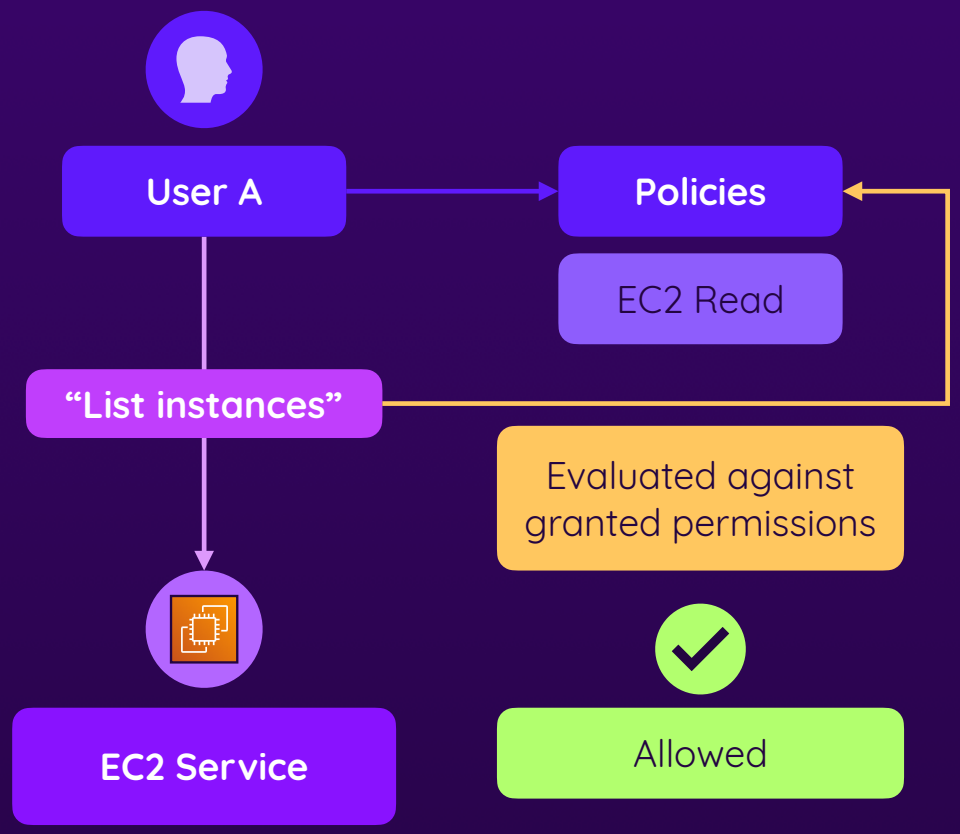
```
{
  "Effect": "Deny",
  "Action": "ec2:*"
}
```

No EC2 access rights!

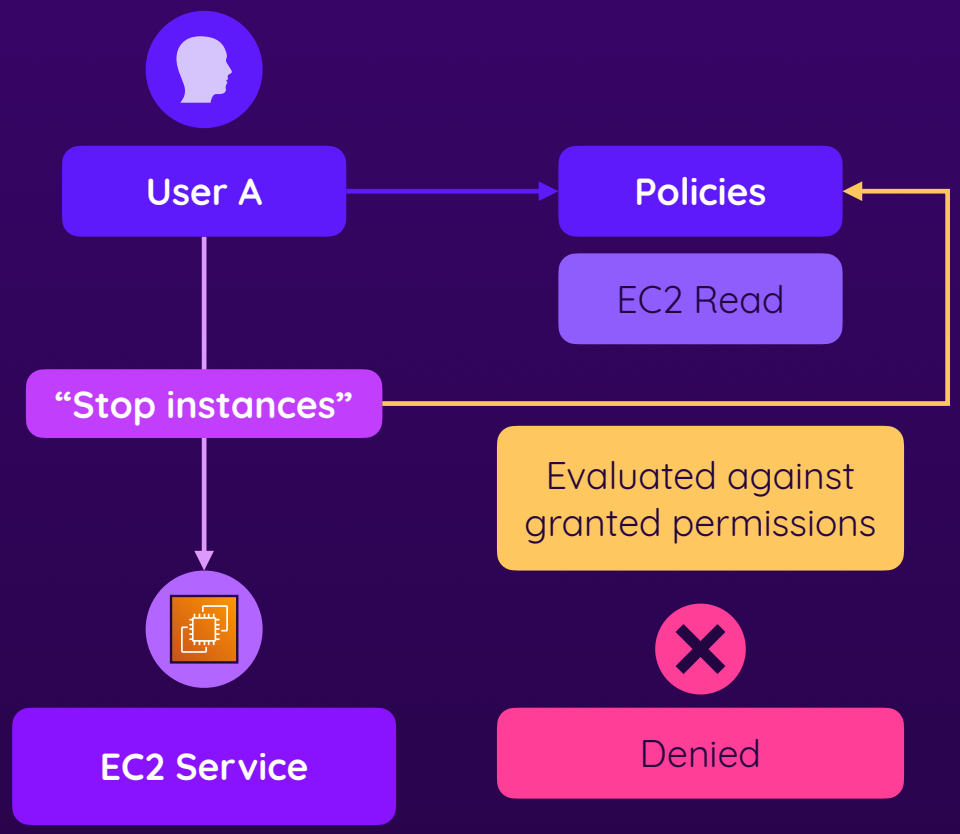
Core IAM Policy & Permission Rules

1. By default, no permissions are granted to any identity (user, user group, role)
2. Multiple policies can be combined to extend the set of permissions
3. Explicit DENYs overwrite explicit ALLOWs

When Are Permissions Evaluated?



When Are Permissions Evaluated?



Summary

Shared Responsibility Model

- Every party secures the things it controls
- AWS secures the infrastructure & managed services
- You secure your applications, workloads & service configs

Account Protection

- Use secure passwords & MFA
- Use IAM users instead of root access
- Don’t share credentials

IAM Key Concepts

- IAM identities (users, user groups, roles) define the WHO
- Policies & permissions define the WHAT
- Policies contain permissions and are attached to identities
- No permissions by default, DENY overwrites ALLOW

ACADEMIND

Compute Services: EC2 & More

Using AWS to run compute tasks in the cloud

- Which Compute Services Does AWS Offer?
- Getting Started with the EC2 Service
- Configuring & Using EC2

ACADEMIND

Which Services Does AWS Offer?

- Compute
- Data Storage
- Database
- Networking & Content Delivery
- Application Integration
- Security
- Cloud Management
- Migration & Edge Computing
- Analytics & Data Ingestion
- Machine Learning & Artificial Intelligence
- Developer Tools
- Business Applications

AWS Compute Services

ECS / EKS (Elastic Container Service)	EC2 (Elastic Compute Cloud)	Lambda (Serverless Code Execution)
An alternative to EC2 - for **containerized workloads**	One of the **most important & popular** services!	The **most popular serverless** compute service
Run clusters of containers in the cloud	Rent a (virtual) remote server / computer	Run code without provisioning any infrastructure
Managed (with vast amount of configuration options)	Fully configurable	No access to the underlying machine or OS
Run any kind of containerized workload	Run any kind of workload in the cloud	Run any kind of code upon pre-defined events

What Are "Containers"?

Containers are "**packages**" of code + the code's dependencies (e.g., OS, required software)

Containers allow developers to distribute and deploy reproducible code environments (including the code itself)

No server configuration required
(since the container already includes the operating system, software, configuration etc.)

Containers can be deployed into **all environments that support containers**

Supporting environments are still computers / servers — they **host the containers**, not the app itself though

ECS / EKS

What Are "Serverless" Services?

Serverless services allow you to run code **without configuring or controlling any servers**

You can perform tasks in response to events by just providing the code that should be executed

AWS Lambda

Serverless services allow you to **focus on your code**, instead of the environment that runs the code

Often, multiple serverless services / tasks **must be combined** to handle more complex workloads

AWS manages the (hidden) underlying server configuration

Understanding EC2

Rent a (virtual) remote machine

"A computer in the cloud"

It's actually (typically) not an entire machine

Rent an Instance

Fully configurable

Choose hardware profile, operating system & install any software

Run any kind of application or task on the instance

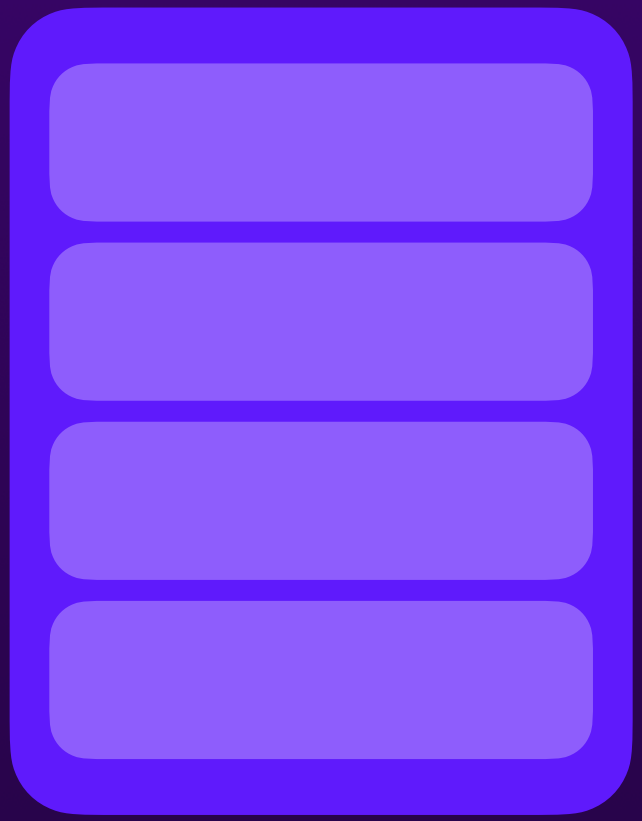
A virtual machine using a "slice" of a the physical machine

Completely isolated from other instances

Dedicated hardware assigned to the virtual machine

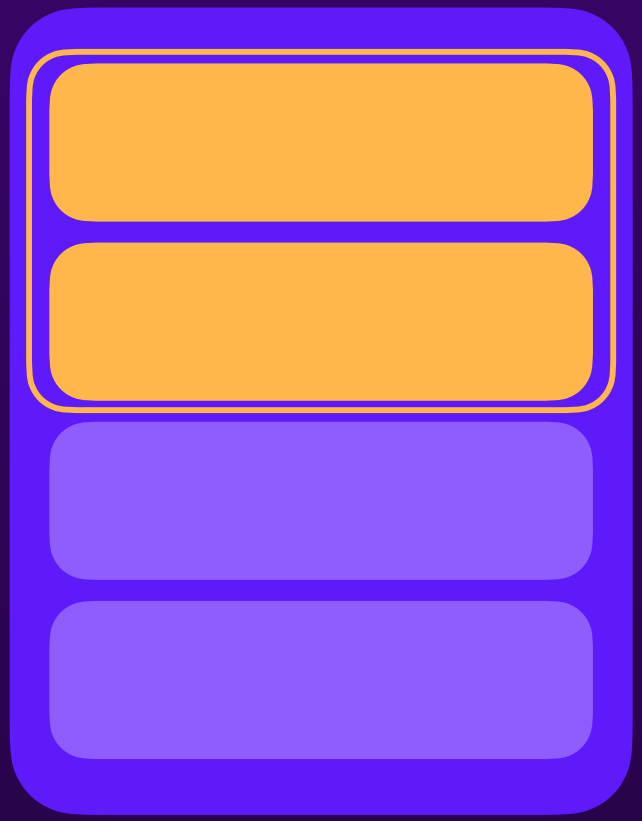
Rent A Slice Of A Computer

Physical Machine in AWS Data Center



Rent A Slice Of A Computer

Physical Machine in AWS Data Center



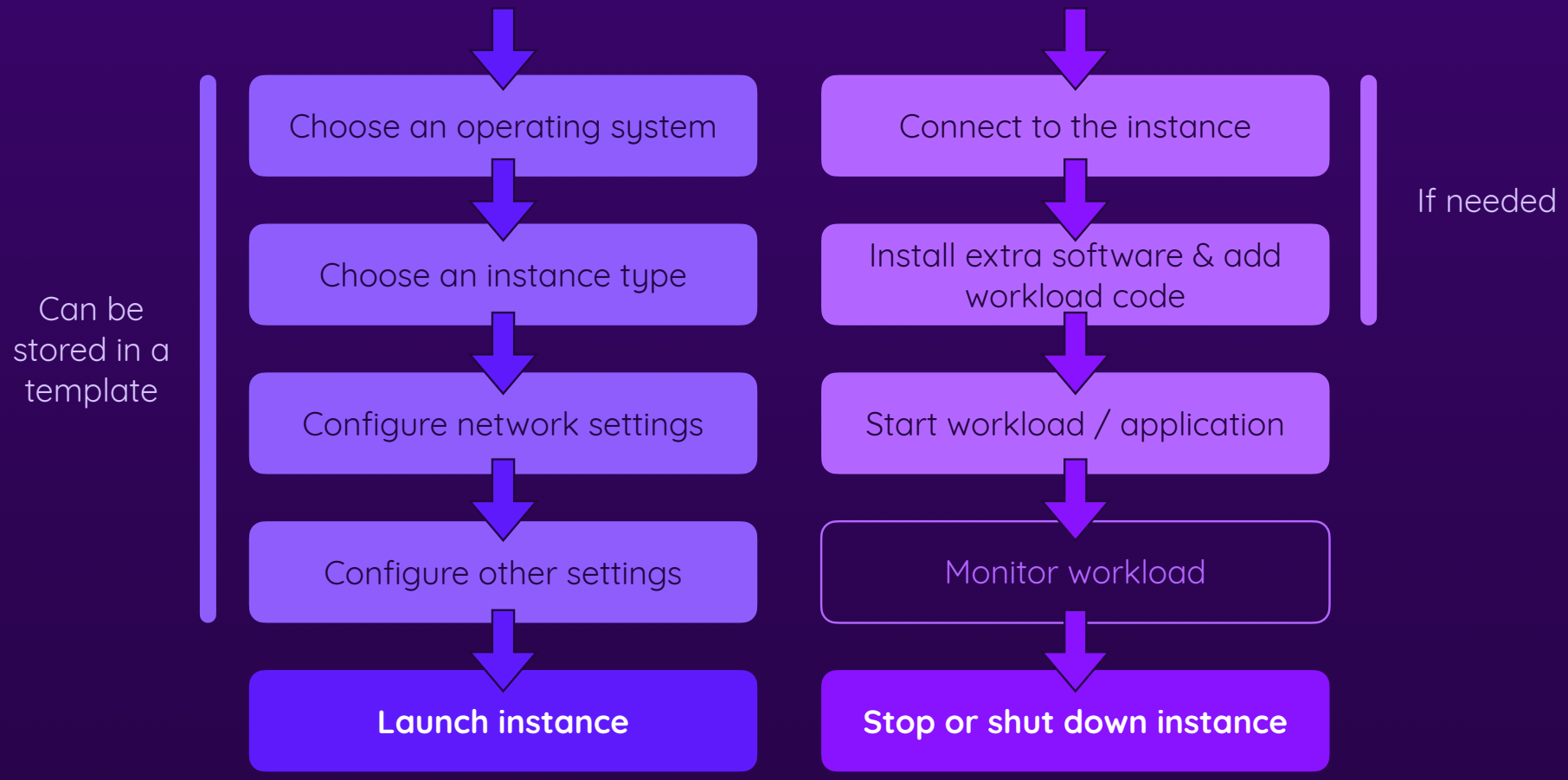
You rent a “virtual server”

An **EC2 Instance**

A slice of the physical machine

Fully isolated from other slices (other instances), with its own **dedicated hardware**

Using EC2 Instances



Amazon Machine Images (AMIs)

Packages of software & setup instructions

Different images yield different operating systems with different additional software and configuration

- You can also create (and share) your own images
- Or use one of AWS' official images
- Or use an image shared by other AWS users & partners

EC2 Pricing

On Demand Instances	Spot Instances	Savings Plans	Reserved Instances
Default & most flexible option	Must be selected	Must be bought separately	Must be bought separately
Pay for usage	Spare instances, can be reclaimed any time	Pay in advance (for chosen amt. of usage)	Pay in advance (for chosen instance types)
No discounts	Discounts over on-demand pricing	Discounts over on-demand pricing	Discounts over on-demand pricing
Price depends on chosen config	Price depends on chosen config	You can use other compute services	Not very flexible & only EC2
	Ideal for workloads that can be interrupted	Ideal if you can commit long-term	

Summary

Multiple Compute Services	EC2 Instances	Running Workloads via EC2
ECS / EKS for containerized workloads	EC2 allows you to “rent” “slices” of real machines: Instances	Connect to EC2 instances via ssh or EC2 Instance Connect
Lambda for serverless compute tasks	Each instance is fully isolated from other instances	Run commands, install software, download code, etc.
“Serverless” = You only provide the code, no server config	Instance configuration (AMI, instance type, etc.) is up to you	Run one or multiple scripts / commands / programs
EC2 for fully customizable server configuration	AMIs define the operating system + base software / config	Stop or terminate whenever you want
Run any workload / task on EC2	Control network access via security groups	Advanced options & different pricing models

What About Lambda & ECS / EKS?

Will be covered later in the course!

But also less important for the exam

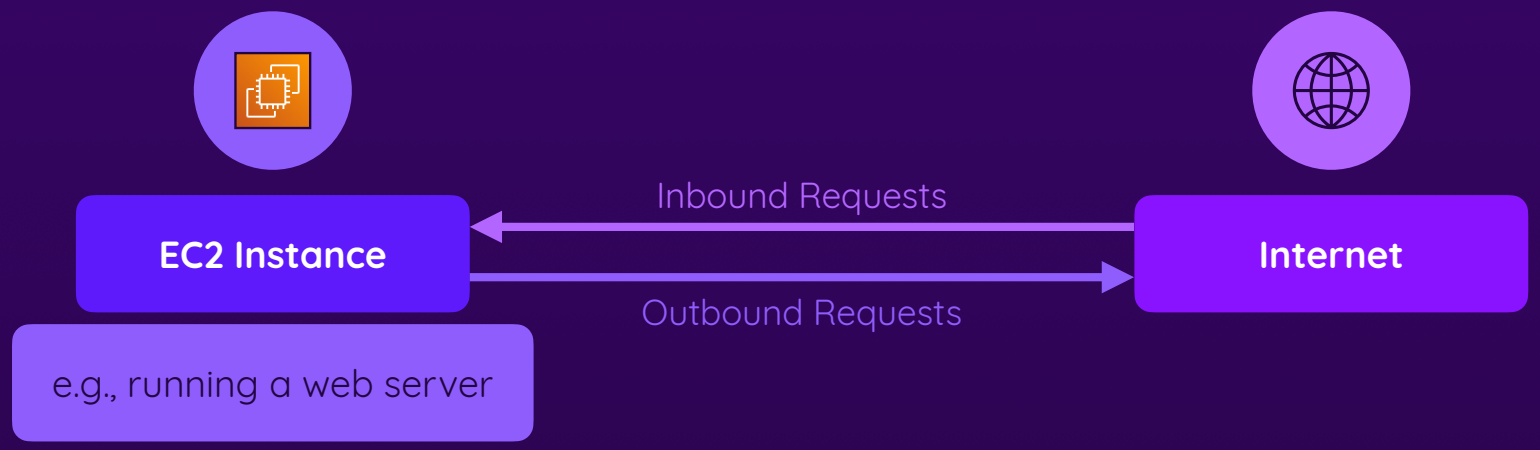
ACADEMIND

VPCs & Multiple EC2 Instances

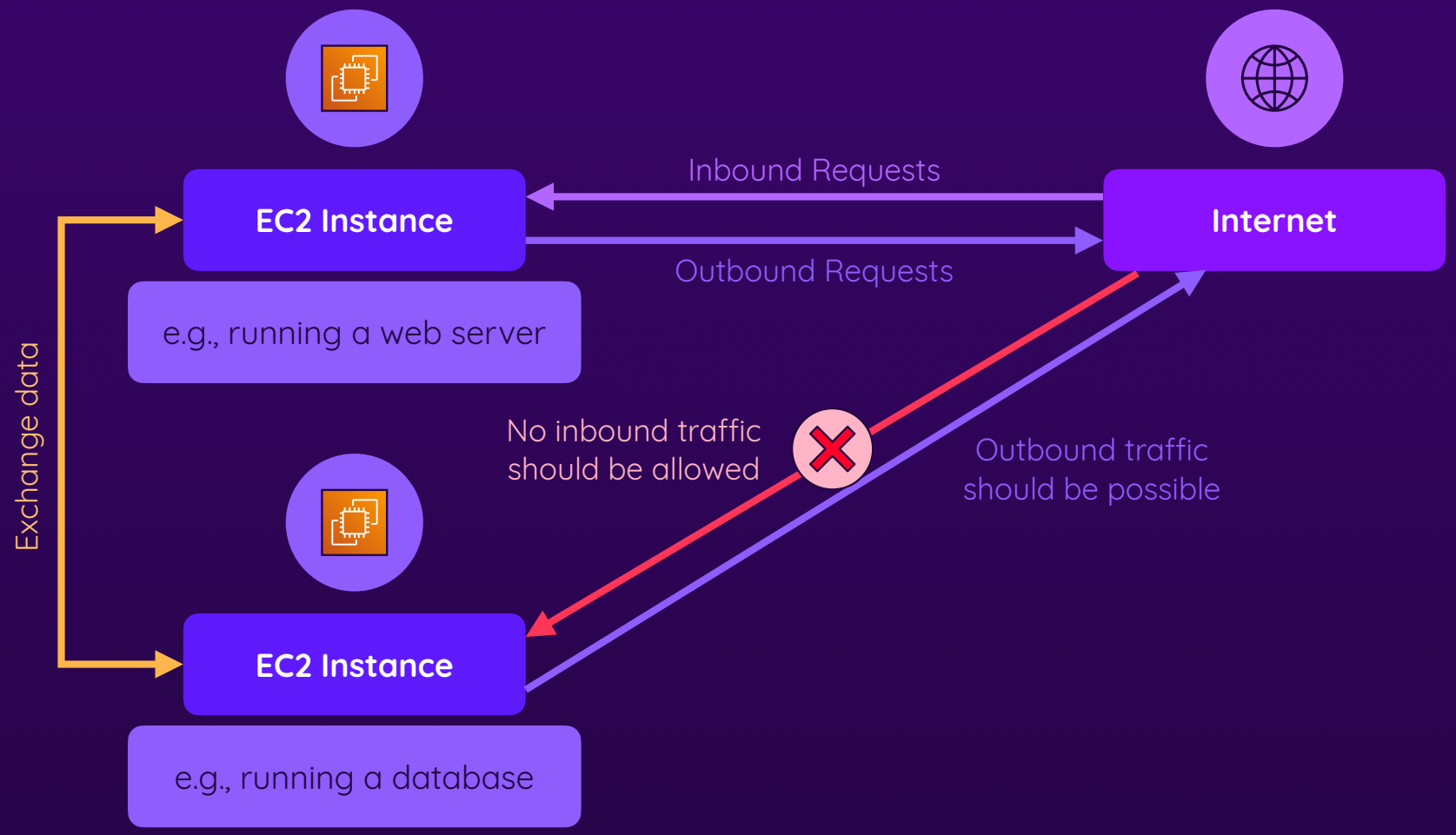
Managing your own network in the cloud

- Understanding VPCs
- Private vs Public Instances
- Managing Network Requests

A Simple Setup



A More Realistic Setup



Public vs Private IP Addresses

Every device in a network has an IP address

Private IPs for internal communication

Multiple servers can send requests between each other via private (internal) IP addresses

Public IPs for internet communication

!

Public IP addresses are a rare resource

Solution 1: IPv6

Solution 2: Limiting IPv4 address usage

Understanding IP (IPv4) Addresses

An IP address is a 32-bit number

172 . 31 . 0 . 0

4 x 8-bit

This is just a notation thing though (for human readability)

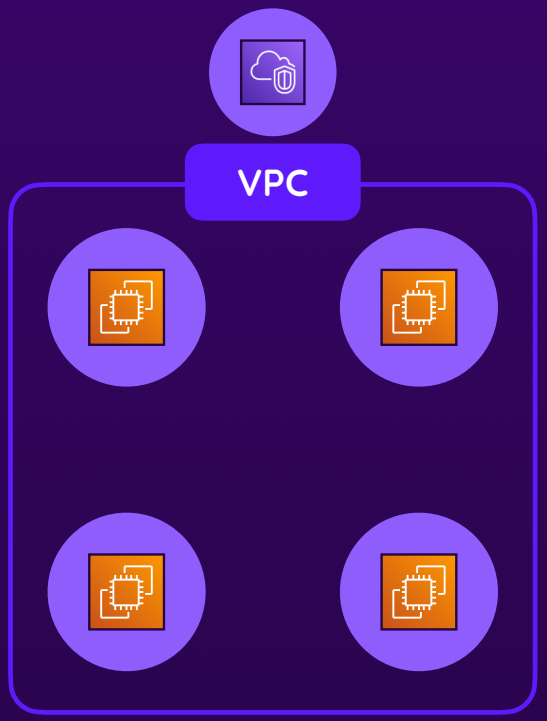
IPv4 Addresses Are A Rare Resource

Less than 4.3bn available IPv4 addresses

↓

Not enough for all the devices (with internet access) we have around the world

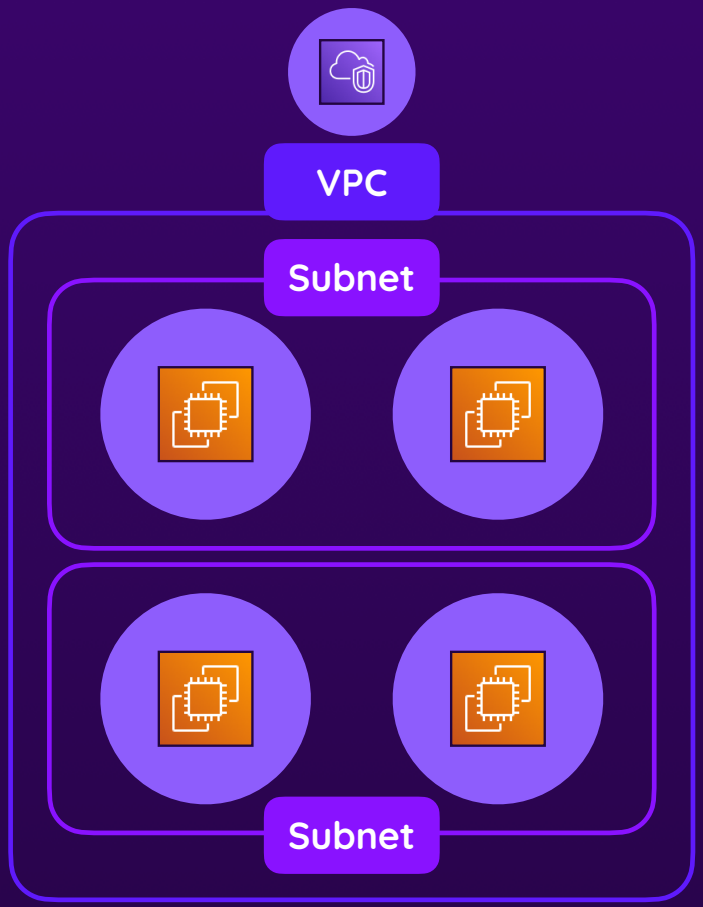
Introducing Virtual Private Clouds (VPCs)



Resources (e.g., EC2 Instances) are grouped & organized in a customer-managed network

IP address assignment & network traffic can be controlled via VPCs

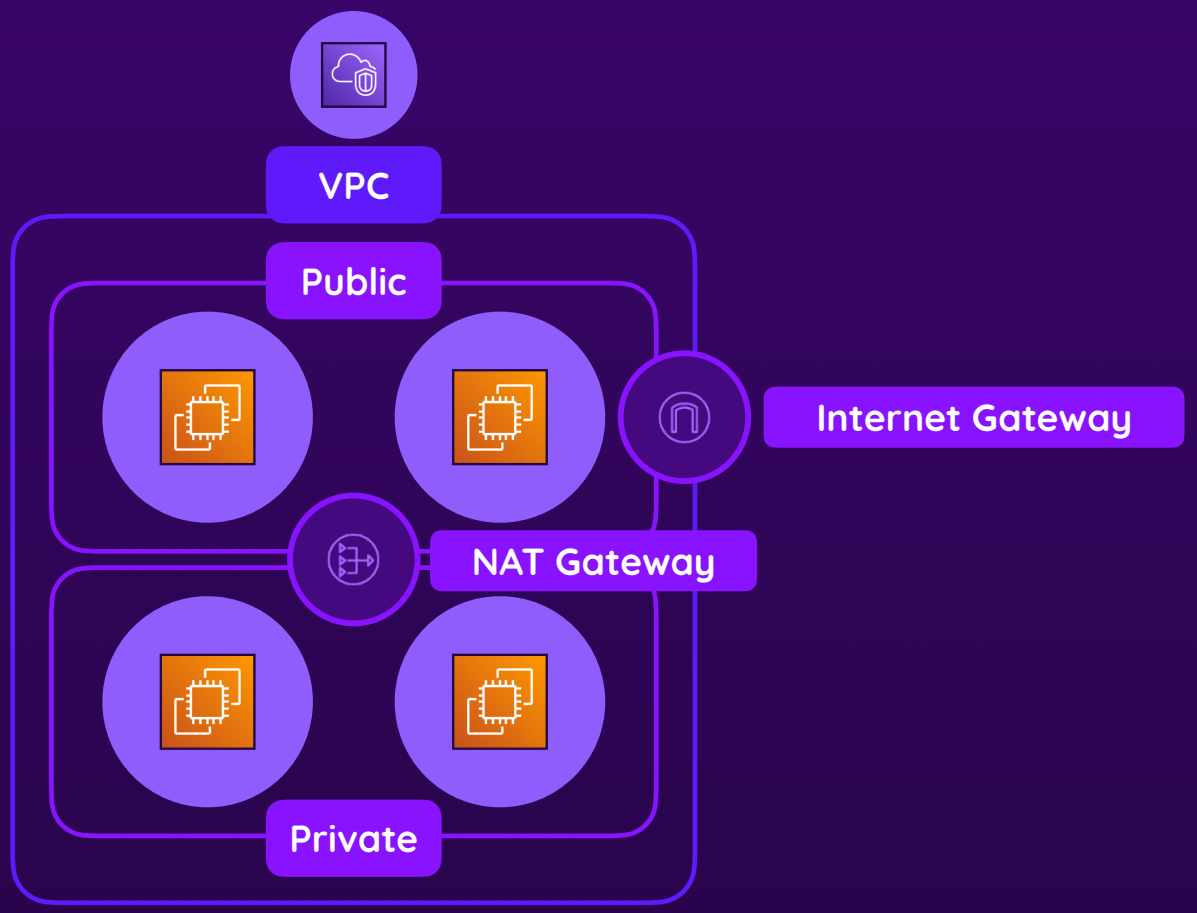
VPCs & Subnets



You actually control network request settings on subnet-level

This allows you to make subnets "**private**" (only internal requests) or "**public**" (internet requests are possible)

Public vs Private Subnets



What About Security Groups?

Security Group	Private Subnets
Firewalls, directly attached to EC2 instances	Not directly attached to instances
Technically, requests still reach the instances	Instead: Contain multiple EC2 instances
Security groups just allow which requests to allow or block	Technical isolation from the internet: Requests technically don't reach the instances

Public vs Private Subnets

Public Subnet

Associated with a route table that has an internet gateway route

Instances can communicate with each other **AND the internet**

Private Subnet

Associated with a route table that has **no internet gateway route**

Instances can communicate with each other only

To still allow for outgoing internet access, a **NAT gateway** can be used

Understanding CIDR IP Ranges

An IP address is a 32-bit number

172 . 31 . 0 . 0 / 16

Defines how many bits are fixed

4 x 8-bit

This is just a notation thing though (for human readability)

Understanding CIDR IP Ranges

172 . 31 . 0 . 0 / 16	172 . 31 . 0 . 0 / 24	0 . 0 . 0 . 0 / 0
16 bits are fixed	24 bits are fixed	0 bits are fixed
Range	**Range**	**Range**
172.31.0.0 ↕ 172.31.255.255	172.31.0.0 ↕ 172.31.0.255	Unlimited
65,536 available addresses	256 available addresses	All possible IP addresses

A <u>higher</u> /X number implies <u>less</u> available IP addresses in the range

Elastic IPs

Automatically assigned IPs (by subnet) will change when instances are stopped / restarted

You can't control which public IP address gets assigned to an instance

Elastic IPs are managed & assigned by you

Elastic IPs don't change and can be re-assigned

Always Use Elastic IPs?

Automatically assigned IPs (by subnet) will change when instances are stopped / restarted

You can't control which public IP address gets assigned to an instance

Elastic IPs are managed & assigned by you

Elastic IPs don't change and can be re-assigned

Elastic IPs should be used with care

Scarce resource: You can only have a few EIPs per region & account

Unused EIPs incur charges

There often are better alternatives

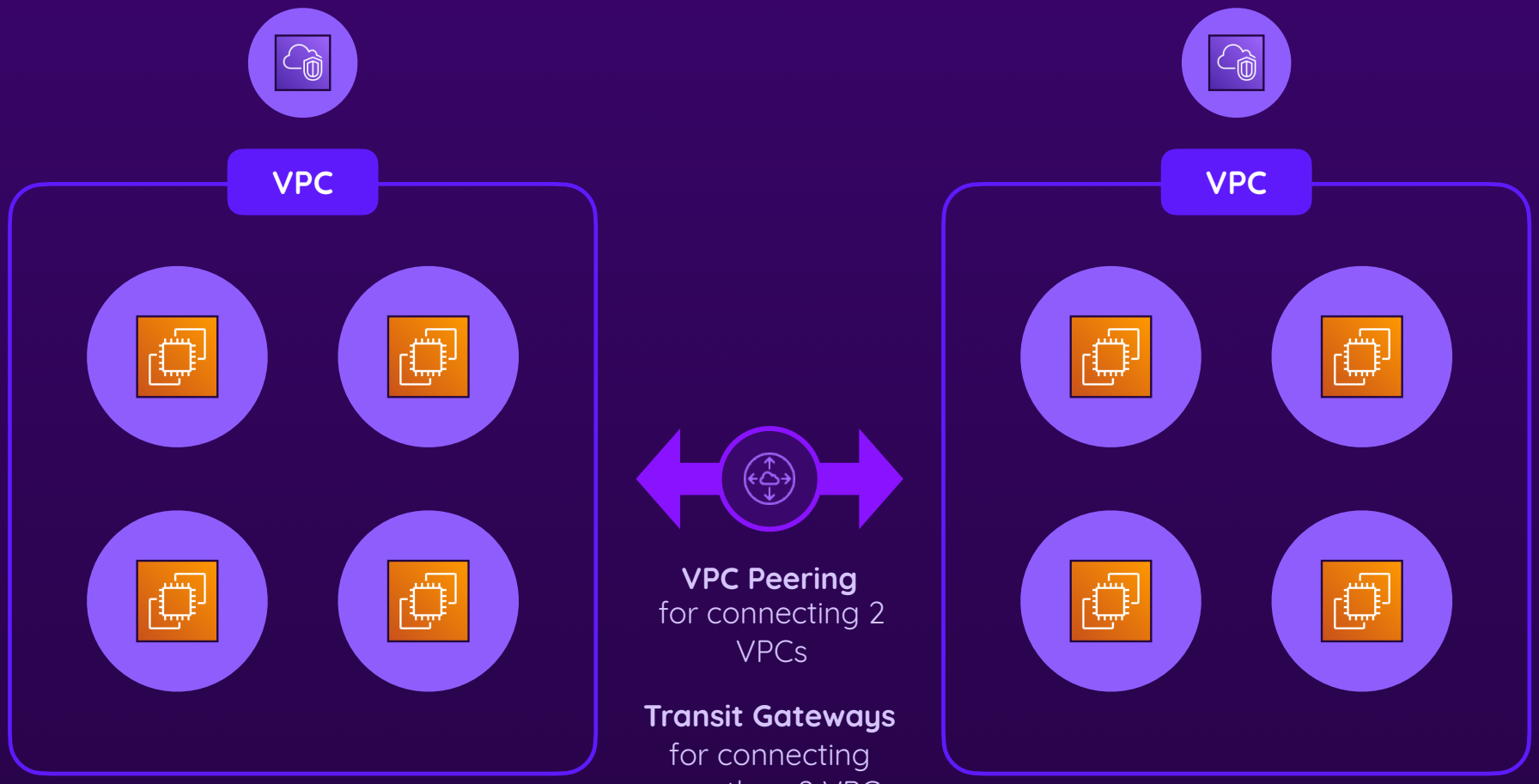
e.g., use DNS for exposing applications / websites to the world

e.g., use application integration services (like SQS) for connecting workloads

Security Groups & Network ACLs

Security Group Preferred	Network ACL (Access Control List)	Private / Public Subnets
Firewall for a single EC2 instance	Firewall for entire subnets	Defines technical connectivity
Checks incoming / outgoing requests and conditionally blocks or allows them	Checks incoming / outgoing requests and conditionally blocks or allows them	No internet access without internet gateway (incoming + outgoing) or NAT gateway (outgoing)
Stateful: Responses are always allowed (if the request passed)	Stateless: Requests & responses are decoupled	Does not control any requests or responses
Multiple instances can have different security groups	One NACL can be associated with multiple subnets	Multiple instances can be in the same subnet
Security groups can be re-used for multiple instances		

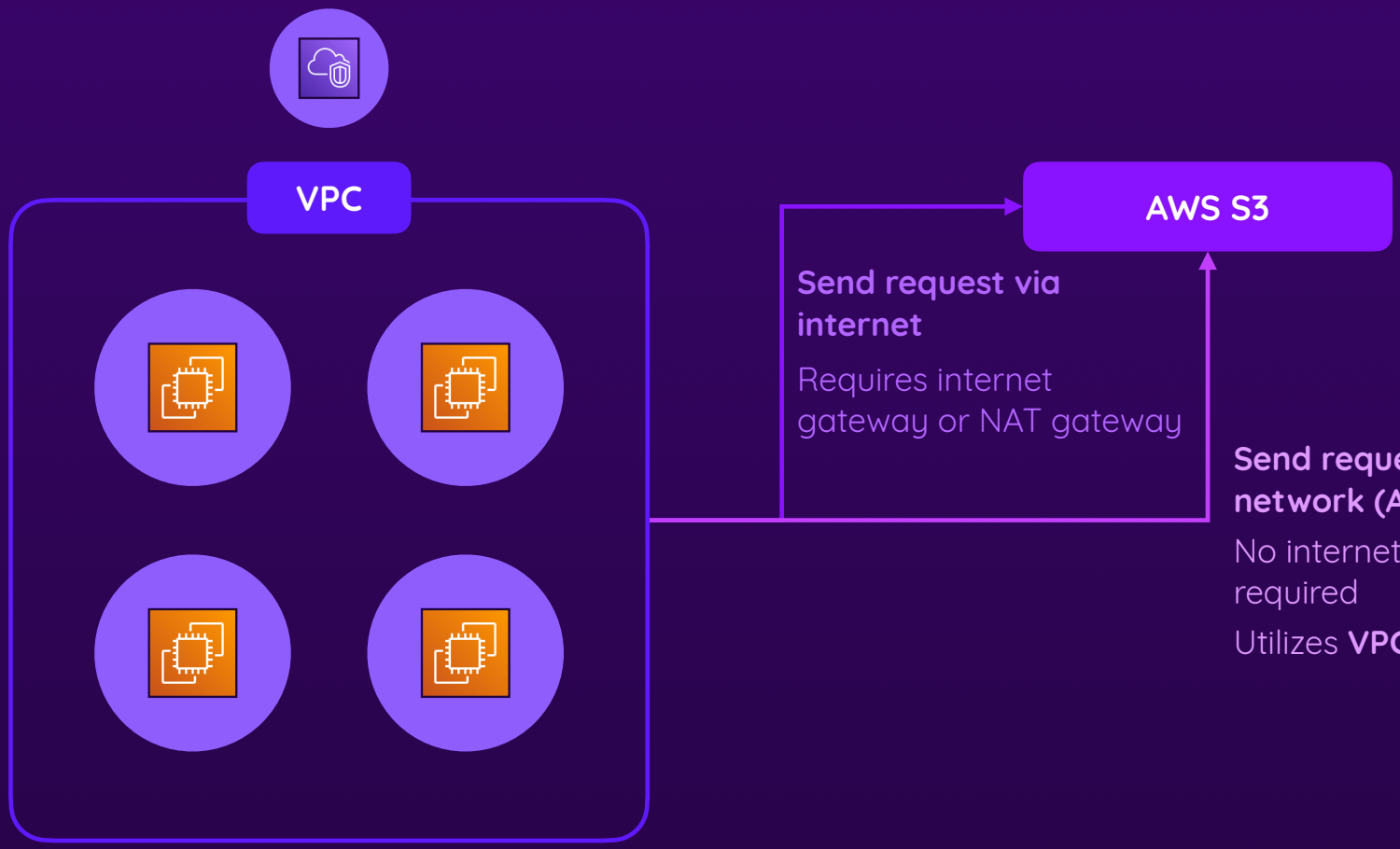
VPC Peering & Transit Gateways



VPC Peering
for connecting 2 VPCs

Transit Gateways
for connecting more than 2 VPCs

VPC Endpoints & AWS PrivateLink



Send request via internet

Requires internet gateway or NAT gateway

Send request via AWS network (AWS PrivateLink)

No internet connectivity required

Utilizes **VPC endpoints**

Summary

VPCs (Virtual Private Cloud)	Network Management	Request Management
Your own network in the cloud (for grouping EC2 instances)	Every VPC has an IP CIDR block assigned (range of IPs)	Internet gateways allow for two-way internet access
A VPC contains at least two subnets & one route table	Subnets get parts of the VPC CIDR block assigned	NAT gateways enable outgoing internet requests
Subnets can be "**public**" or "**private**"	EC2 instances receive auto-assigned public and private IPs	NACLs allow or deny requests on subnet-level
Route table settings control subnet "visibility"	Elastic IPs can be used for fixed IP addresses	Endpoints (PrivateLink) connect AWS services to VPCs
	VPC peering or transit gateways can connect VPCs	

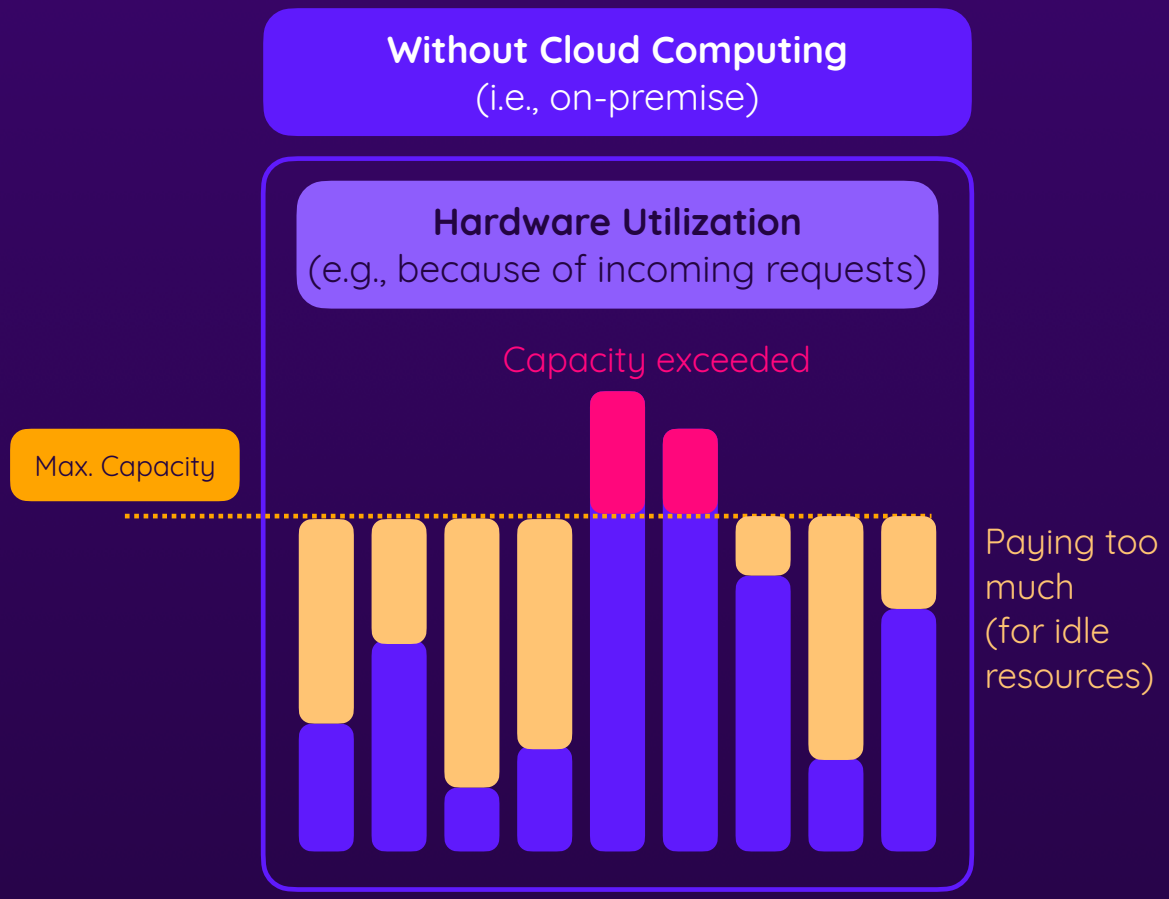
ACADEMIND

Dynamic Scaling & Load Balancing

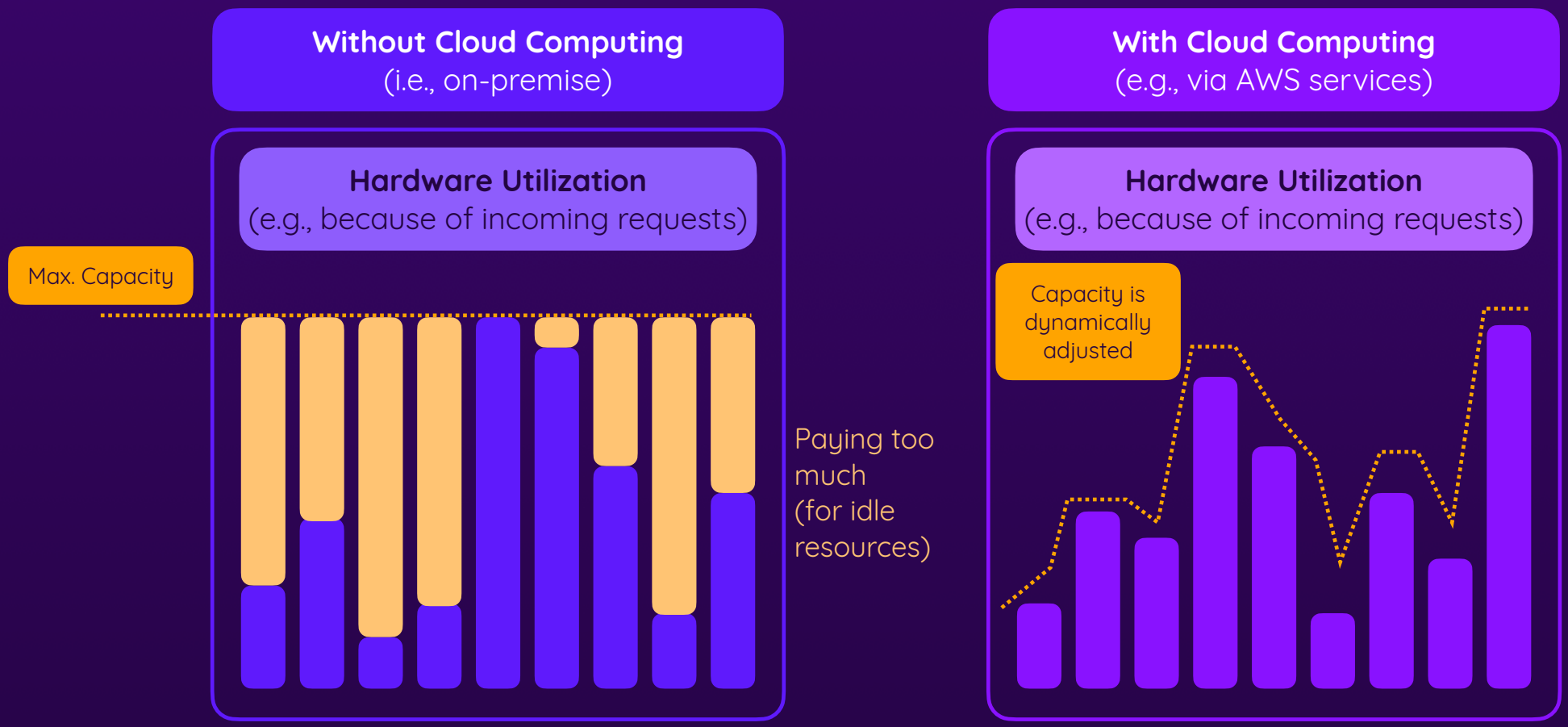
Building for scale

- Scaling & Load Balancing: What & Why?
- Understanding AWS Auto Scaling
- Understanding AWS Elastic Load Balancers

The Need For Flexibility



The Need For Flexibility



AWS Compute Scaling Services

EC2 Auto Scaling

Service which can be used to automatically add / remove EC2 instances (based on conditions)

Ensures sufficient capacity at all times, without over-provisioning

Elastic Load Balancer (ELB)

Service to distribute load (e.g., incoming requests) evenly across available instances

Ensures that all available instances are utilized equally

Application Load Balancer

Network Load Balancer

Elastic Load Balancer

Application Load Balancer

- Feature-rich
- Broad variety of request forwarding conditions & rules
- Capable of SSL termination
- Can reduce app complexity

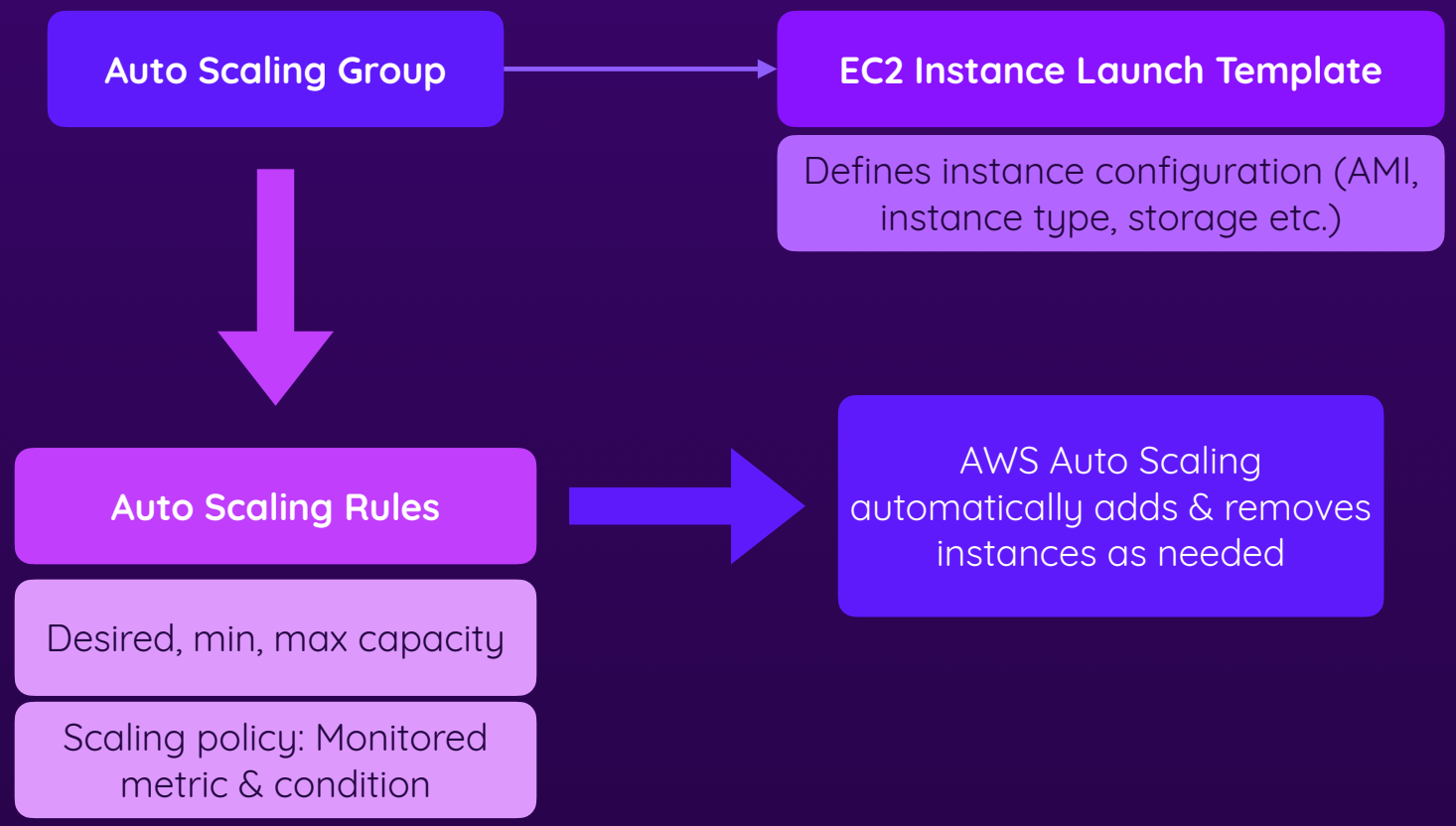
Use for (most) HTTP apps

Network Load Balancer

- Very lean
- Limited configuration options
- Fixed IP address
- Perfect for non-HTTP traffic

Use for non-HTTP services

Using Auto Scaling



Using Load Balancers

ALB or NLB

Core Settings

Handled requests (type + port) & VPC / Subnet

Target Groups

VPCs & Subnets containing the instances

Health checks for determining which instances are available

ELBs automatically forward incoming requests to available instances (evenly)

After creation, additional forwarding rules can be added

Summary

Elasticity, Scalability & High Availability	Auto Scaling	Elastic Load Balancer
Workloads don't necessarily have even load patterns	Automatically add / remove instances	ALB & NLB can be used for distributing traffic evenly
Too little or too much capacity can be a big problem	Set clear rules and min / max requirements	Define target groups (in VPCs / Subnets) and forwarding rules
Being able to scale instantly & automatically is important	Instance count is adjusted to incoming load based on rules	ALB is perfect for HTTP traffic (and feature-rich)
Load should also be distributed evenly to avoid downtimes	Use launch templates & VPC / subnet settings	NLB is great for other network traffic

ACADEMIND

File Storage with EBS, EFS & S3

Storing & managing files

- Understanding Different File Storage Services
- EC2 & EBS or EFS
- Configuration Options & Settings

File Storage?

Files generated by (web) applications

- User uploads
- Generated invoices
- Transformed images
- ...

File archiving

- Accounting files
- Legal documents
- Vacation images
- ...

Different Kinds of File Storage

Block Storage	Object Storage	File System
An unformatted hard drive	No information about underlying system	A (network) file system
Format (and structure) before using	Store & retrieve files of any kind of size as needed	A pre-formatted & -configured file system
Create custom structure & store any files	No (real) custom structure can be created	Create custom structure & store any files
Attach a (virtual) hard drive to a server	Store files without caring about the underlying system	Get a (virtual) file system without any manual setup
EBS	S3 (Explored later in the course!)	EFS, FSx

Understanding EBS

EBS
Elastic Block Store

Hard drives, attached to EC2 instances

Multiple volumes can be added to the same instance

Can be formatted & structured as needed

Via the command line (or code), from inside the instance

EBS Core Features

Different Types	Elastic Volumes	Snapshots	Multi-Attach
SSD vs HDD	If needed, volumes can scale dynamically	EC2 instance data & state can be saved	Attach volumes to multiple instances
Optimized for different workloads & tasks	Extra feature, must be enabled / managed	Restore snapshots on (new) EC2 instances	Supported on some instances

Important: EBS is exclusively available for EC2 instances!

EC2 Instance Storage

All EC2 instance to have some "base storage"

Contains operating system, base software etc.

EC2 Instance Store

Hard drive, which is part of the machine / rack in the data center

Default

EBS-backed

A (managed) EBS volume

Understanding EFS

EFS
Elastic File System

Scalable, pre-formatted file systems

Can be attached & accessed to / from multiple services

Can be structured as needed

Via the command line or code

EFS vs EBS

EBS Elastic Block Store	**EFS** Elastic File System
Unformatted hard drive	Pre-formatted file system
Less automatic scaling, more manual work	Scales automatically
Multi-attach is possible but not the focus	Multi-attach is a core feature
EC2-exclusive (only for EC2 instances)	Can be used with multiple services

FSx Lustre & Others

FSx

(Sometimes) an alternative to EFS

For high-performance file access workloads

No feature equality!

Summary

File Storage Services	EBS	EFS
Store application, user, business or personal files	Attachable block storage (unformatted hard drives)	Attachable pre-configured file system
Different kinds of storage: hard drives, file systems, objects	Format, structure & use manually	Built for (dynamic) scalability & multi-access
EBS, EFS, FSx Lustre & S3 are AWS' main storage services	For EC2 instances only	For multiple AWS services
	Extra features: Snapshots, elastic volumes, multi-attach	FSx Luster for high-performance file access tasks

ACADEMIND

Object Storage with S3

Storing (and accessing) any files from anywhere

- Understanding S3
- Buckets, Settings & Accessing Files
- Beyond Simple Storage: Static Websites & File Archival

Understanding Object Storage

Focus on the file, not the underlying system

Store any kind of file with (almost) any kind of size

Simple Storage Service (S3) Key Features

File Storage

- Upload via console, CLI, HTTP API, SDK, ...
- Delete & read via console, CLI, etc.

Buckets

- Files are organized in buckets (“folders”)
- Buckets are regional
- Unique name required

No File System

- Flat structure, no directories
- Prefixes help with organizing files

Detailed Permissions System

- Control bucket- and file-level access
- Via Bucket Policies (Recommended)
- Via ACLs

Files Are Stored In Buckets

Files Are Stored In Buckets

S3 does not provide a file system where you could create subfolders. Instead, all files are stored in buckets.

You can create as many buckets as needed.

Storage Classes

Different file access patterns

Frequent Access

Accessed very frequently (e.g., every second / minute ...)

Instant access required

Highest flexibility but no cost savings

Infrequent Access

Accessed infrequently / only from time to time

Instant access possible

Cost savings but retrieval cost

Archive

Almost never accessed

Instant access not always possible

High cost savings but less flexibility

S3 Intelligent Tiering

S3 Advanced Features

Versioning
Store multiple versions of a file

Lifecycle Management
Transition files between classes

Inventory & Analytics
Understand stored files & data

Compliance & Object Lock
Prevent object deletion / changes

Replication
(Auto-)replicate objects cross-bucket

Data Protection & Encryption
Automatically encrypt stored data

Static Website Hosting
Upload & host (static) website files

S3 vs EBS vs EFS

EBS Elastic Block Store	EFS Elastic File System	S3 Simple Storage Service
Attachable hard drives	Attachable file systems	Independent object storage
EC2 only	Multiple services	Access with or without services
Automatic scaling & multi attach possible	Automatic scaling & multi attach are key features	Unlimited scaling built-in

Summary

S3 - Object Storage	Managing Objects & Storage	Advanced Features
Focus on the objects / files, not the underlying system	Different storage classes for different access patterns	Inventory overview & data analytics
Organize files into buckets	Lifecycle management	Static website hosting
Access (upload, delete, retrieve) via services, CLI, HTTP API, ...	Fine-grained permission management	Versioning & object lock
	Encryption possible	Cross-region or single-region cross-bucket replication

ACADEMIND

Databases on AWS

Storing application data

- Self-managed Databases vs Managed Database Services
- Understanding & Using RDS (incl. Aurora)
- Understanding & Using DynamoDB

So Many Options!

Different Types of Databases

e.g., SQL vs NoSQL

Different data or workload requirements favor different database types

AWS allows you to run & use all database types

Different Hosting Options

Self-hosted vs managed

You can install + operate databases yourself (e.g., on EC2 instances) or let AWS do that

AWS supports both options

Database Types: SQL vs NoSQL

Data can be stored & structured in different ways

Two main kinds of database types

Relational & Fixed Schema (SQL)	**Non-Relational & No Schema (NoSQL)**
Data is **normalized** and **split** across multiple tables	Data is **not normalized** and only split across a **few** (or **no**) tables
Every table has a clearly defined **data schema** which **enforces homogenous** data records	Tables **don't enforce** a fixed structure and allow for **heterogenous** data records
Examples: MySQL, PostgreSQL, Oracle SQL, ...	Examples: MongoDB, AWS DynamoDB, ...

Self-Hosted: Advantages & Disadvantages

Self-hosted Databases	Managed Databases
Install & operate database software manually	Let AWS manage setup & database operations
e.g., on EC2 instances	Key services: **RDS**, **DynamoDB**
Full control but also full responsibility	Less control but also less responsibility
Important duties: Keep software patched, manage backups etc.	You define general rules but AWS does the heavy lifting

Relational Database Service (RDS)

Managed SQL Databases

Choose Database Engine

- e.g., MySQL, PostgreSQL
- Database version
- Auto-update settings

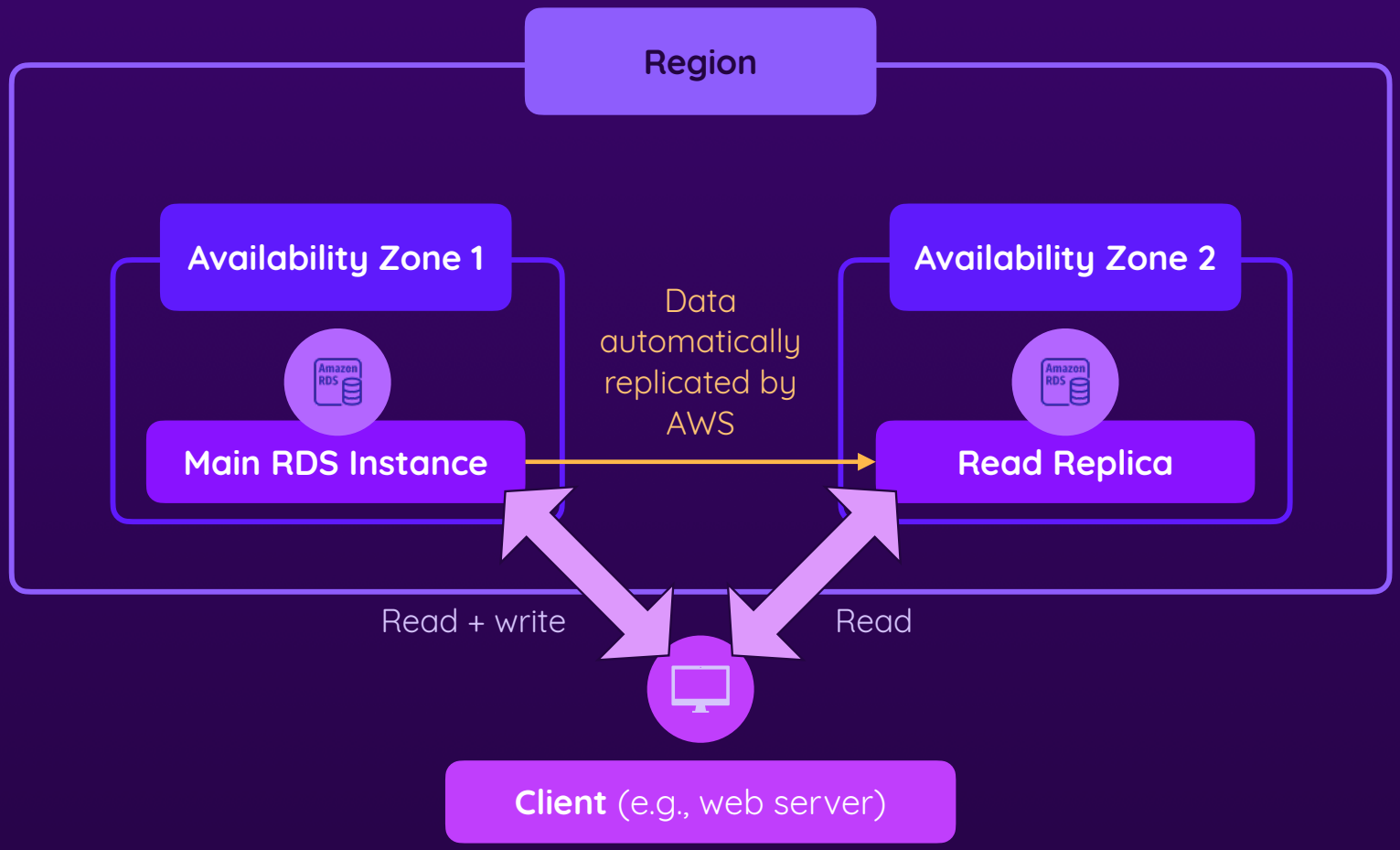
Choose Hardware & Network Configuration

- Instance class (hardware profile)
- VPC, subnet & security group

Configure Database Server

- Login credentials & port
- Replication (for high availability & performance)
- Monitoring settings
- Encryption & backup settings

High Availability Thanks To Replication

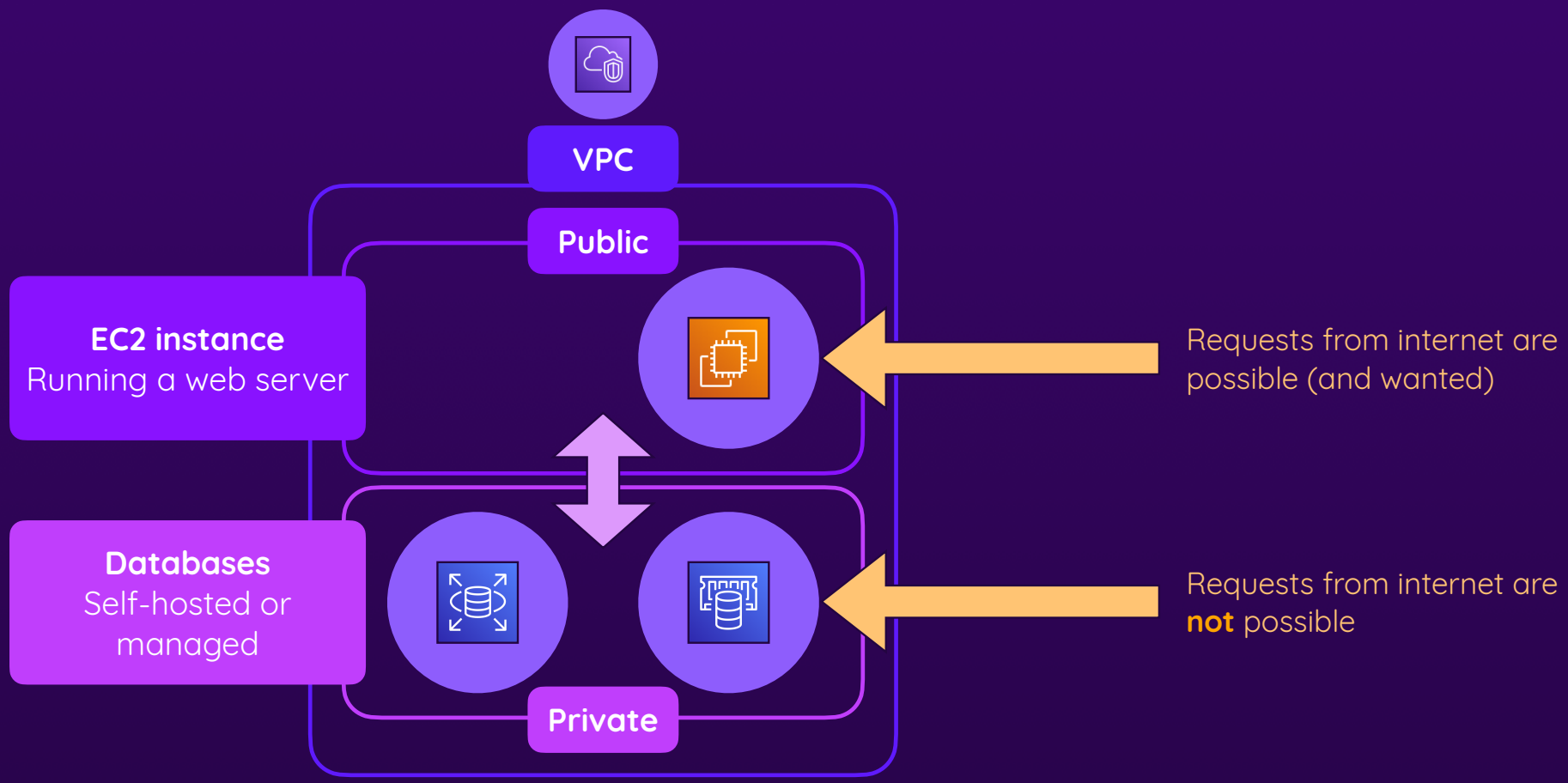


Amazon Aurora

Amazon's own SQL database engine

MySQL & PostgreSQL compatible database engine with great scalability & performance

Databases & VPCs



Caching Data with ElastiCache

A fully managed in-memory caching database

Choose Database Engine

- Redis or Memcached
- Optionally enable cluster mode (for scaling)
- Choose engine version

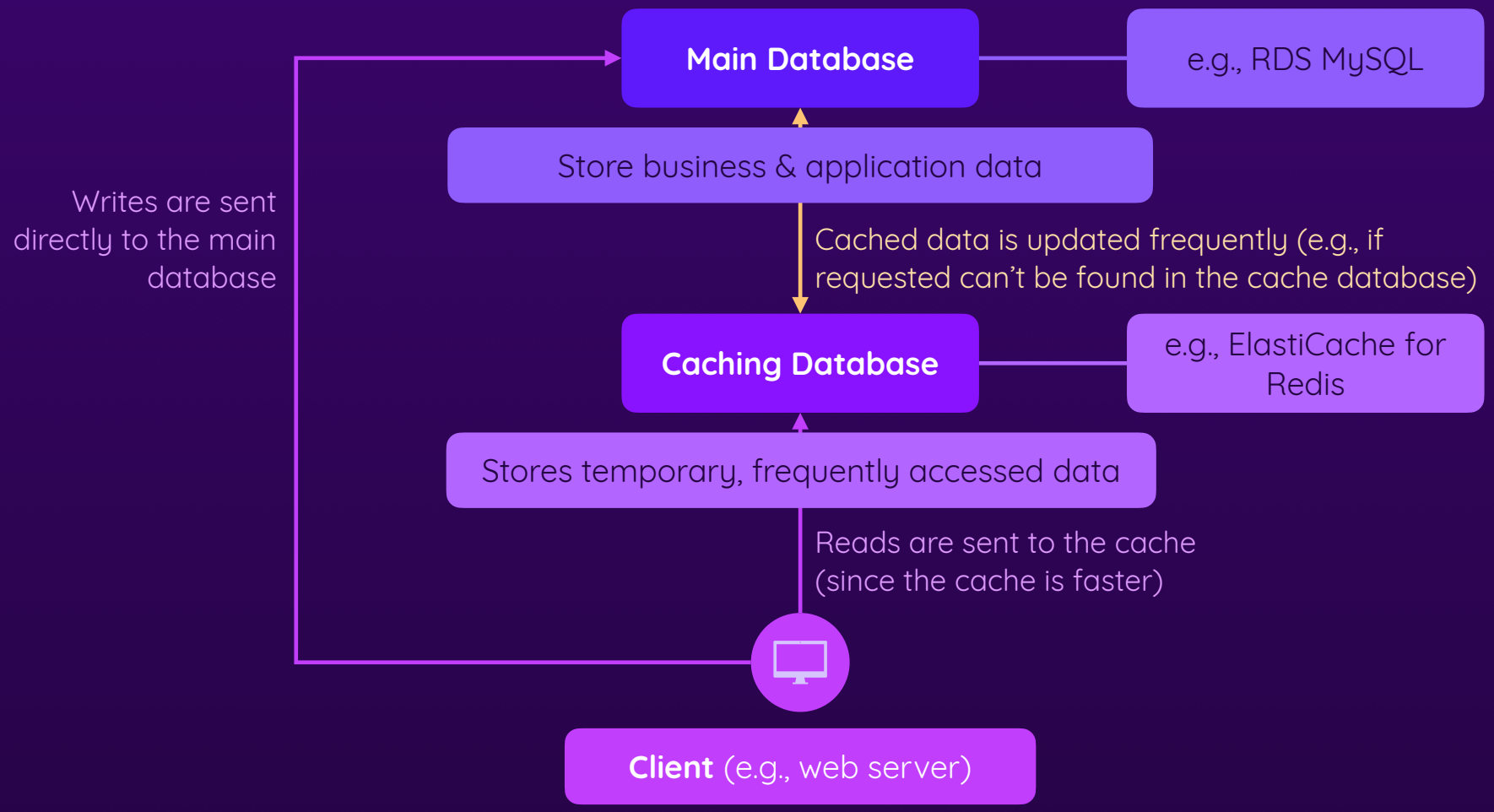
Choose Hardware & Network Configuration

- Node type (hardware profile)
- VPC, subnet & security group

Configure Database Server

- Encryption & backup settings
- Maintenance settings
- Monitoring settings

What Is “Caching”?



SQL Alternative: NoSQL Key-Value Stores

Concept

No enforced schema, no cross-table relations + Instead: Values stored by key in a single table → With the right key & strategy, this can offer huge performance benefits

Table 1

	Key	Value(s)	
Data entry 1	pb1	title: Product 1	price: 49
Data entry 2	ala	title: Product 1	tags: ['free']
Data entry 3	bca	name: Product 1	price: 89

Understanding DynamoDB

A fully managed NoSQL key-value database

Create Tables

- Set name & key(s)
- Set expected read / write capacities (or on-demand)
- Choose encryption settings

Manage Data

- Write & read via AWS API / CLI / SDK
- Configure backups

More DynamoDB Features

Streams

- Time-ordered series of database item changes
- Subscribe to process item changes

Global Tables

- Fully managed multi-region database
- High availability thanks to automatic replication
- Great performance thanks to global reach

DAX

- Managed in-memory cache for DynamoDB
- Accelerates database queries

Other Database Services

MemoryDB

Persistent in-memory storage

DocumentDB

Document (nested data structure) database

Keyspaces

Wide column database (flexible column formats)

Neptune

Graph database (complex data relations)

Timestream

Time series database

Quantum Ledger Database

Immutable log of data changes

AWS Backup

Centralized Backup Management

Create Plan

- Use template or create custom
- Set frequency, retention period etc.
- Define destination & timeframe

Manage Resources & Backups

- Assign resources (e.g., RDS cluster) to backup plan
- Access & restore backups if necessary

Summary

Different Database Services	RDS, Aurora & ElastiCache	DynamoDB & More
Self-hosted (on EC2) vs managed services	Managed relational database services	DynamoDB: Managed high-performance key-value database
SQL (RDS, Aurora) vs NoSQL (DynamoDB, DocumentDB, ...)	Configure database cluster hardware, network & behavior	Define partition keys & read / write capacity (or on-demand)
Different database for different workloads / data requirements	Leverage built-in scaling & availability (replication) features	Access DynamoDB data via AWS API / SDKs
	Access databases via HTTP endpoints / SQL queries	Other databases for specific use-cases

ACADEMIND

Content Delivery & Global Networking

From servers to users (and back)

- Managing Custom Domains
- Caching & Delivering Content via CloudFront
- Other Networking Features & Services

Understanding DNS

Domain Name System (DNS)

Translate domain names to IP adresses

academind.com → 163.12.19.201

There are many domain name registrars

External

AWS Route 53

Understanding Route 53

Managed DNS Service

Manage Domains	Manage DNS
Register or transfer domains	Create routing records
Create hosted zones for domains (config containers)	Manage failover or create complex routing rules

Delivering Content with CloudFront

Managed CDN
(Content Delivery Network)

→ **Uses AWS' Edge Locations**

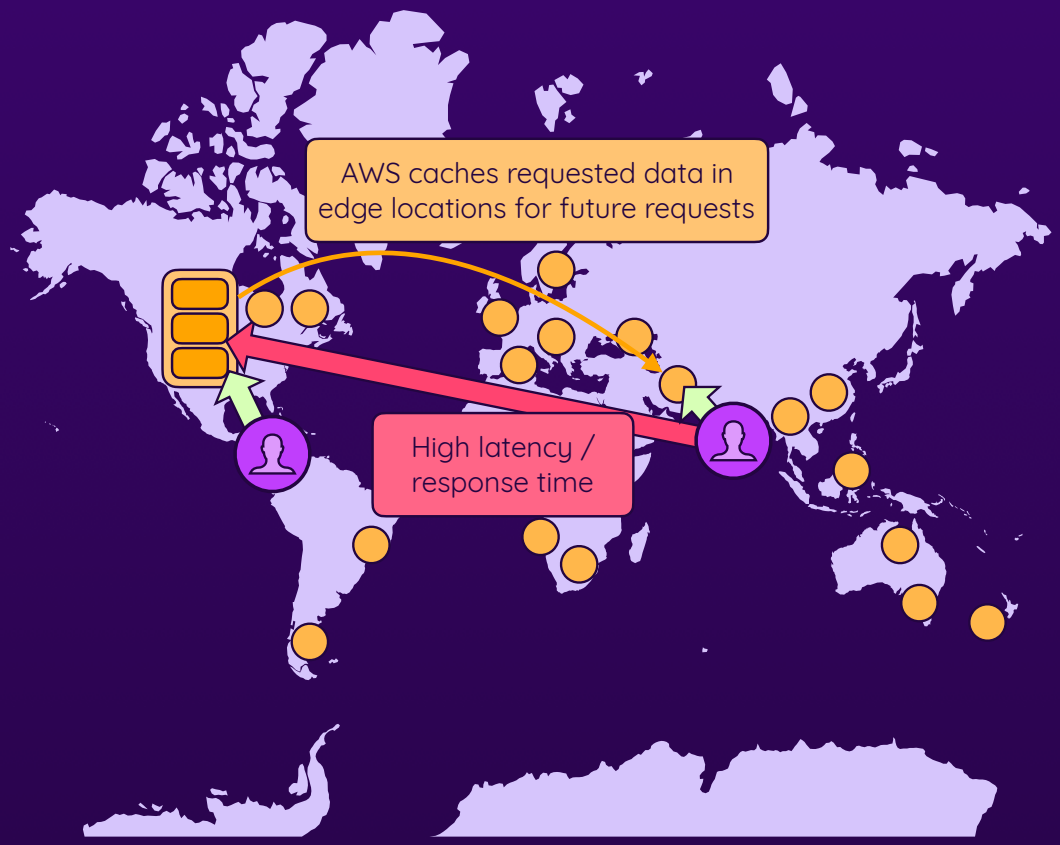
Manage Content Origins

- Create a distribution connected to content sources
- Define distribution behaviors
- Set logging, SSL & security settings

Deliver & Cache Content

- Create caching policies (containing cache rules)
- Connect caching policies to distribution
- Use functions for request / response manipulation

Understanding CDNs



Local Zones, Outposts, Wavelength

Local Zones	Outposts	Wavelength Zones
Smaller AWS regions close to big metropolitan areas	Add AWS infrastructure to your on-premises	AWS services embedded into 5G networks
Perfect for achieving ultra-low latency	AWS-managed infrastructure	Perfect for achieving lowest latency possible
Limited set of supported services	Limited set of supported services	Limited set of supported services
Extends VPCs to local zones	Extends VPCs to outposts	Connect to other services running in a region

Global Accelerator & Transfer Acceleration

Global Accelerator

- Improve user traffic performance via AWS network
- Receive two unique, stable IP addresses
- Use GA for balancing multi-region traffic

S3 Transfer Acceleration

- Improve data transfer speed via AWS edge network
- Less network variability, more bandwidth utilization

AWS Certificate Manager (ACM)

Managed SSL Certificates

Request Certificates	Assign Certificates
Request new SSL certificates	Use SSL certificates with a variety of AWS services
Issued by AWS, free to use for AWS services	Examples: ALB, NLB, CloudFront
Alternative: Import your own certificate	Services handle SSL encryption

Summary

Various Networking Services	Route 53 & CloudFront	AWS Network & Acceleration Services
VPC: Cloud-internal	Register & manage domains with Route 53	**Local Zones**, **Wavelength**: Run services closer to your customers
Route 53: DNS service Register domains, define routes	Define request forwarding rules for (sub-)domains	**Outposts**: Run services closer to your infrastructure
CloudFront: CDN service, using AWS edge locations	Use CloudFront for distributing (cached) content globally	Accelerate traffic or data (file) transfers with accelerators
Local Zones, **Outposts**, **Wavelength**: Extended regions	Target (and "wrap") other services with CF / Route 53	
Global Accelerator, **Transfer Acceleration**: Traffic acceleration		

ACADEMIND

Beyond EC2: Serverless & Containers

From instances to Lambda functions & ECS clusters

- A Closer Look At “Serverless Services” & Containers
- Understanding AWS Lambda & ECS / EKS
- When To Use It

What Are "Serverless Services"?

Services where you don't need to provision, configure and pay for servers

Instead: **Define the task** that should be performed (e.g., a code snippet that should be executed) and **when** it should be performed

Task

e.g., execute code that moves an image file

triggers

Event

e.g., image was uploaded to S3 bucket

AWS Lambda

There Are Other Serverless Services!

AWS Lambda is the main serverless compute service

But compute isn't everything!

e.g., you can think of S3 as a serverless storage service

A Closer Look At AWS Lambda

Code	Event	Configuration
You upload or define the code that should be executed	Choose a supported event source	Timeout, underlying architecture, file systems, ...
Write code in console, upload ZIP file or Docker container	e.g., a file was uploaded to S3	Attach an execution role for permissions
Choose a supported programming language	Advanced setup (e.g., event filtering)	Connect to a VPC (it's NOT placed in there though)
Advanced setup (e.g., environment variables)		

EC2 vs Lambda

EC2	Lambda
Spin up instances, install software & run your code	Upload your code & define execution events
You can install & run any software	You can only executed code (can't install software)
e.g., run web server, run databases, ...	e.g., no easy way of running web servers, no databases, ...
Extremely versatile & configurable	Focused on event-driven code execution
Does requires lots of manual setup work & pay for uptime	Almost no manual setup work required & only pay for usage

What Are Containers?

Packages of code and execution environment

You create so-called “images” which contain container definitions

Containers are than instantiated based on images

Created with software like Docker

Single Image Application	Multi Image Application
One container contains all the parts that make up the application	Multiple containers contain the parts that make up the application
e.g., web server & database in one single image	e.g., web server & database in two separate images
Multiple containers may be started (based on same image) for scaling	Multiple containers for multiple images (+ potential scaling containers)

Running Containers

ECS

EKS

Server that's able to create + run containers → Because Docker is installed on it

Instantiates & manages containers

Image 1	Image 2
Container 1	Container 3
Container 2	Container 4

ECS & EKS

ECS
Elastic Container Service

EKS
Elastic Kubernetes Service

Use AWS' custom container cluster deployment solution

Both are services that help with deploying, running, managing & scaling **containerized applications in clusters**

Use the open-source Kubernetes system

Because manually creating & managing clusters of servers (which run your containers) is very challenging

Understanding ECS & EKS

Managed Container Clusters

Services that help with launching, scaling & managing containers

Define Cluster Structure

- Define tasks: images & image configurations
- Choose EC2 instances or **Fargate** as container host
- Configure default network & security settings

Operate & Scale Containers

- Define service- / task-specific settings
- Monitor containers
- Start or stop when needed

The Need For Container Image Repositories

Image must be available
(in the environment, where the container should be created)

Goal: Run a container based on an image

Because the image is the blueprint for the container — it defines the OS, software, application code etc.

Option 1
Local development environment on your machine

Image is stored locally, no remote registry (storage) is needed

Option 2
On some server (e.g., a Fargate instance)

Image must be stored on a distribution server: e.g., Docker Hub or AWS ECR

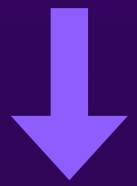
Managing Images with ECR

Managed Container Image Registry

Manage Repositories	Manage Images
Repositories contain images	Push images to ECR repositories
Create public or private repositories	Use ECR-stored images in other services like ECS
Enable encryption or image scanning	Share public ECR-stored images with others

Understanding Fargate

Serverless Container Execution Environment



Don’t worry about picking EC2 instance types or instance configuration

Summary

Serverless & Containers	AWS Lambda	ECS, EKS, ECR
Alternative to EC2 (where you rent entire servers)	Serverless, event-driven code execution	Managed container clusters, help with running containers
Serverless: On-demand code execution (with a timeout)	Provide code + define event triggers + execution configuration	Provide images & environment configuration
Containers: Packages of code + required execution environment	Many supported event types (e.g., S3 file changes, ...)	Run on top of EC2 instances or Fargate (serverless)
Different problems benefit from different solutions	Assign execution role for permissions	Manage & distribute images with ECR

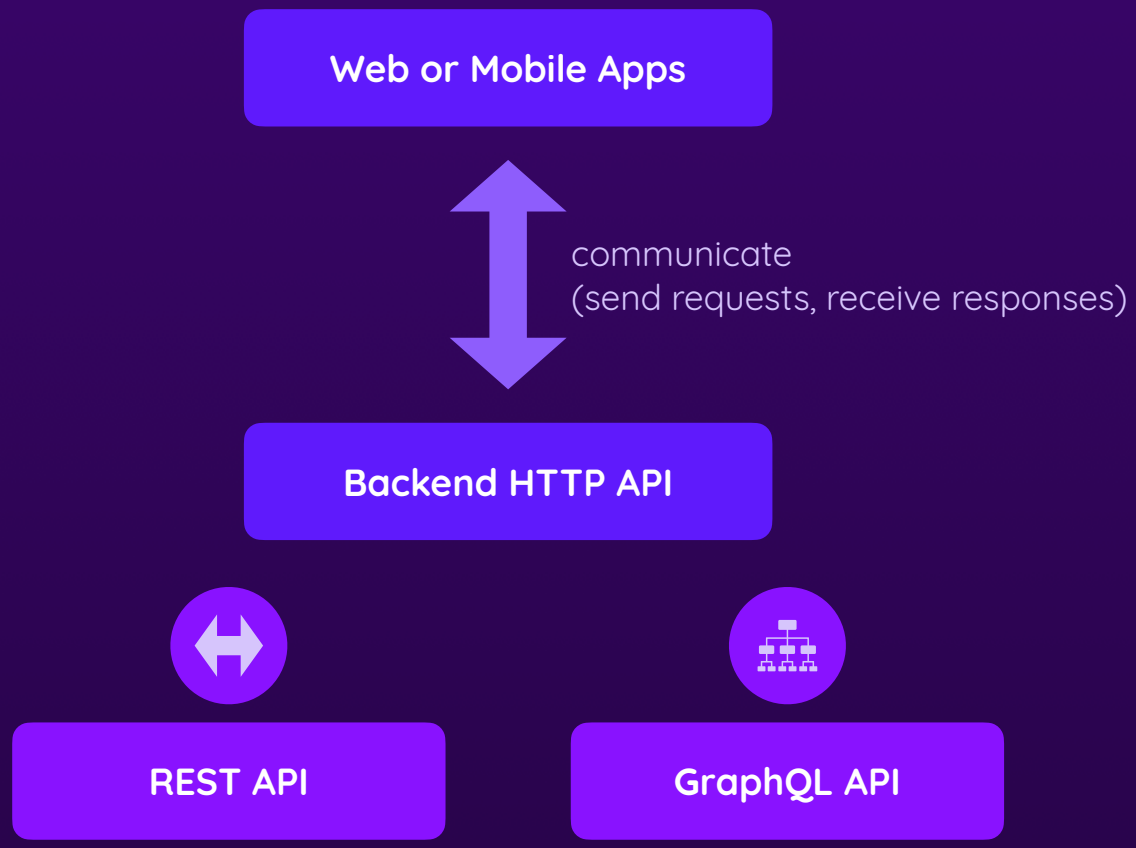
ACADEMIND

Serverless APIs & Web Applications

Building complex serverless web applications with ease

- Building Serverless REST & GraphQL APIs
- Letting AWS Handle User Authentication
- Using AWS Services In (Frontend) Code

Web & Mobile Apps & AWS



Building Web (HTTP) APIs

REST	GraphQL
Request targets resource paths (e.g., /books/1)	Request contains GraphQL query statement
Request entire data for a selected resource	Request partial data (only the data needed)
Utilize HTTP verbs (GET, POST, ...)	POST requests only
Extremely common	Popular because of reduced redundancy

REST APIs with API Gateway

Managed REST API Service

Define API Structure	Handle Requests	Test & Deploy
Define resources (paths & HTTP methods)	Define rules for handling & parsing requests	Test during development
Enforce query parameters or authentication	Configure response creation & forwarding logic	Deploy with stages
Define response codes & schema	Handle real-time connections (websockets)	

GraphQL APIs with AppSync

Managed GraphQL API Service

Define API Structure

- Define schemas, queries & mutations
- Connect schema to data sources & resolvers
- Add authentication

Handle Requests

- Supports real-time and on-demand connections
- Built-in query optimization & caching

User Authentication With Cognito

Managed App User Authentication Service

Manage User Pools

- Configure user credentials requirements
- Configure user authentication experience
- Integrate with your applications
- Assigns temporary IAM permissions to users

Allow Social Sign In

- Create federated identity pool
- Add Google, Facebook, Apple etc. authentication
- Assigns temporary IAM permissions to users

Amplify

Application Development Platform & Framework

Focus on the Product	Build a Backend or Host an Application
Don’t focus on the underlying services	Let Amplify manage services & app data
Creates infrastructure on your behalf (other services)	Use Amplify Studio to configure the backend
Integrate with client-side code via SDK	Manage data via Amplify Studio

Summary

Connecting Cloud Services to Frontend Apps

- Many cloud services should be used by frontend apps
- Typical frontend<=>backend communication uses HTTP APIs
- REST & GraphQL APIs are common HTTP API solutions
- **API Gateway:** Build serverless REST APIs
- **AppSync:** Build serverless GraphQL APIs

Building APIs

- Define resources & request handling with API Gateway
- Create query definitions & schemas with AppSync
- Define Lambda functions that should be executed
- Handle requests without running your own API server
- Implement user authentication with Cognito

Simplifying the Cloud

- Amplify is a platform that simplifies cloud app development
- You can let Amplify create and manage AWS resources for your
- Amplify also provides a complete CMS (if needed)
- Use the Amplify client-side SDK for frontend cloud integration

ACADEMIND

Simplifying Application Deployment

Using EC2 & other services with less effort

- What's the Problem?
- Solutions: Elastic Beanstalk & More

Easier (Web) Apps with Elastic Beanstalk

Simplified (Web) Application Deployment

→ Configure web app environments in a single place

Create Applications & Environments

- Define programming language & environment
- Choose a preset & adjust as needed
- Configure instance types, security settings & more

Add Load Balancing, Scaling, Databases

- Add load balancing etc. as needed
- Add a connected database in the same creation wizard
- Configure updates & deployments

Lightsail - Simplifying EC2

A Simplified EC2 Management Console

Launch EC2 instances with a simplified wizard

+

Focus on specific tasks (e.g., install + configure Wordpress)

Simplifying Container Deployment

App Runner

A service that simplifies the process of deploying containers

Use ECS & Fargate under the hood

Copilot

A CLI that simplifies container app creation & deployment

Used to create cloud environments & deploy apps

Summary

Why Simplify?	Elastic Beanstalk & Lightsail	App Runner & Copilot
Especially for beginners, AWS services can be challenging	Elastic Beanstalk helps with creating EC2-based apps	App Runner simplifies the deployment of containers
Focus on the goal, instead of the tools	Configure network, security, load balancing etc. on one screen	Uses ECS etc. under the hood
Goal: Make AWS more accessible to a broader audience	Add database if needed	Copilot is a CLI for deploying containerized apps
Not necessarily used by experts	Lightsail focuses on customers looking for a hosting provider	

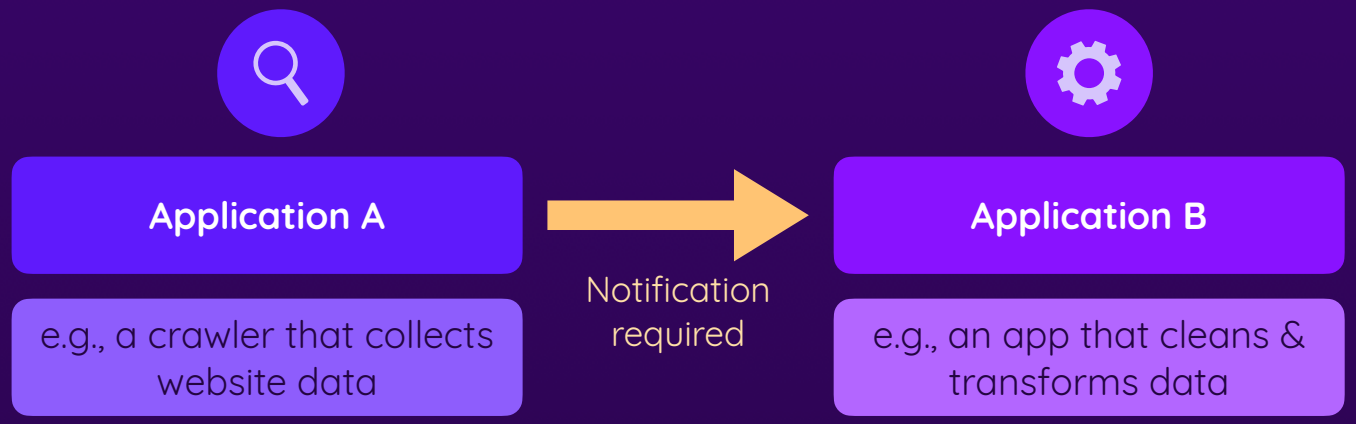
ACADEMIND

Application Integration

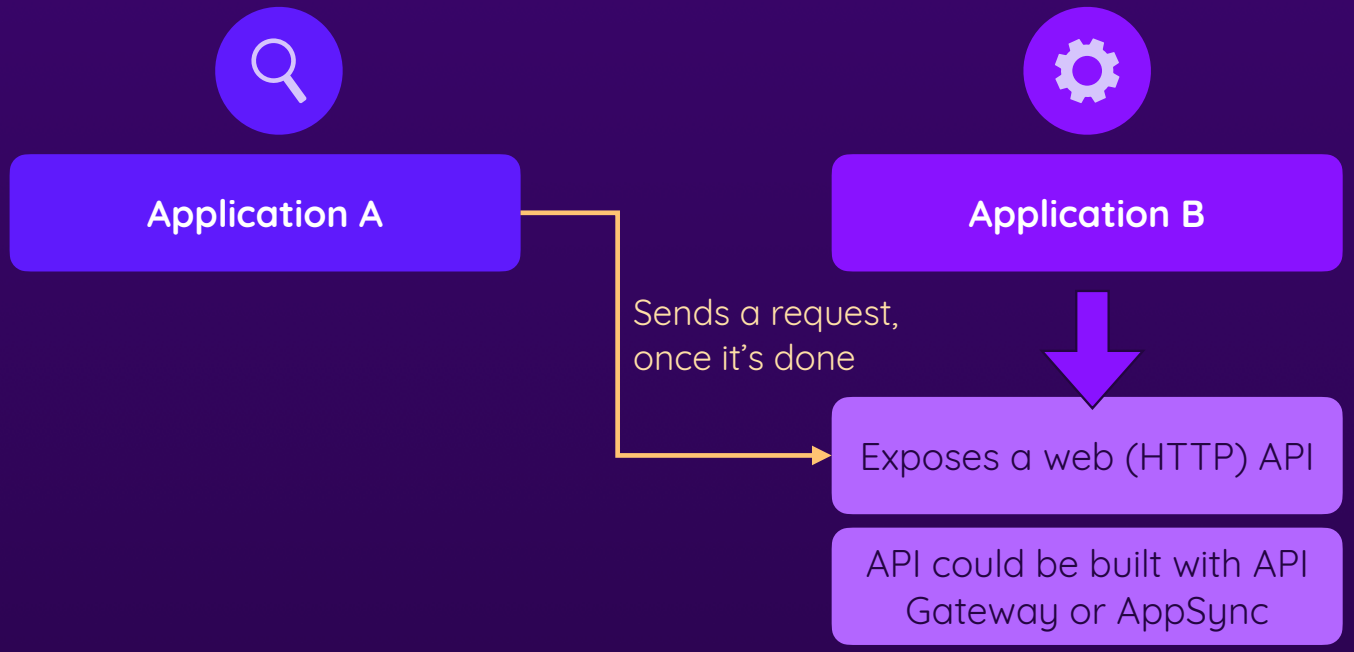
Connecting cloud applications & workflows

- What's The Problem? What's (Not) The Solution?
- Communicating via SNS & SQS
- EventBridge & Other Services

What's The Problem?



A Possible Solution



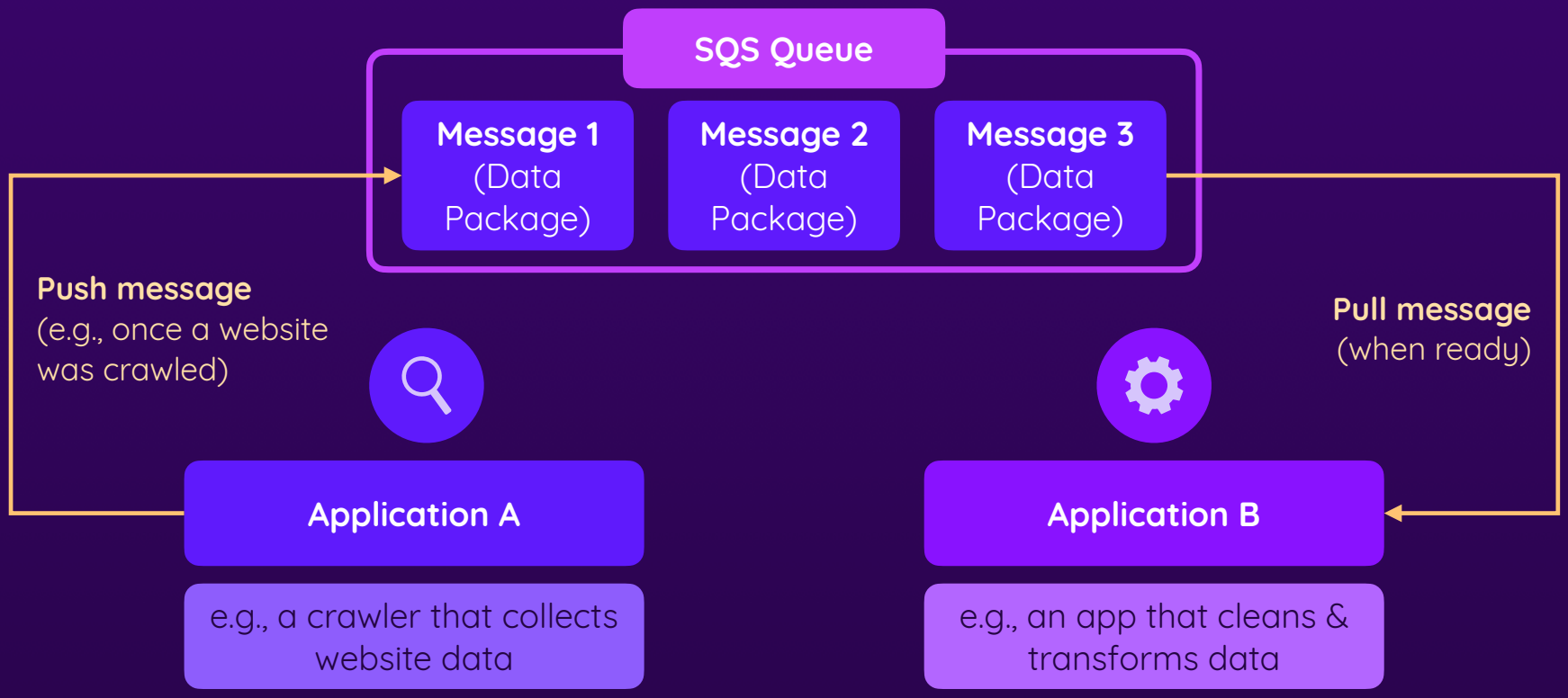
Problems

- Overhead / extra work & costs
- Same solution must be re-implemented many times
- Connectivity between apps (& services) could be an issue

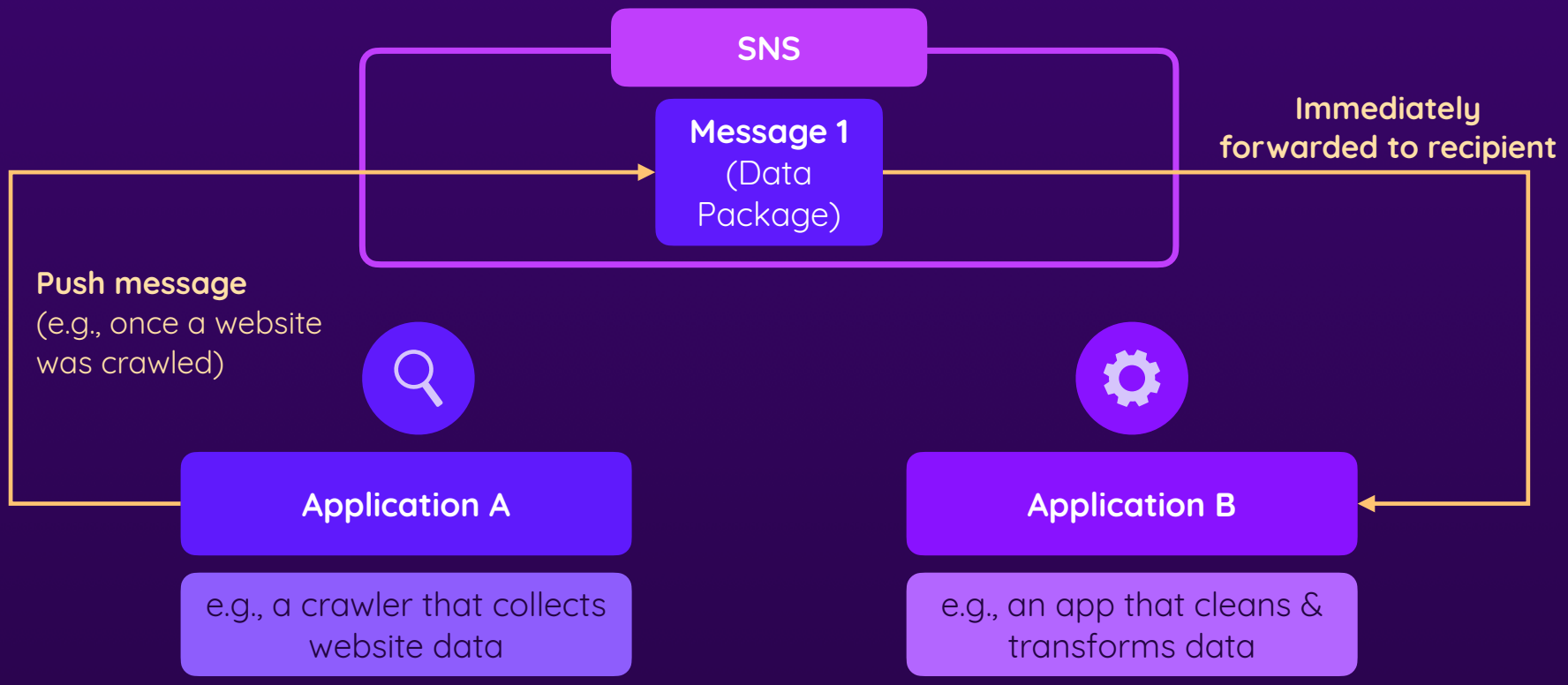
Instead: Use App Integration Services

SQS Simple Queue Service	**SNS** Simple Notification Service	**EventBridge**
A managed message (data package) queue service	A managed push message (data package) service	Listen to events & trigger actions
Push & read messages to the queue	Push messages directly to interested subscribers	React to all kinds of events & trigger other services
Asynchronous processing	**Synchronous** processing	**Synchronous** processing
Directly triggered from inside other services / code	**Directly** triggered from inside other services / code	**Indirectly** triggered because of events

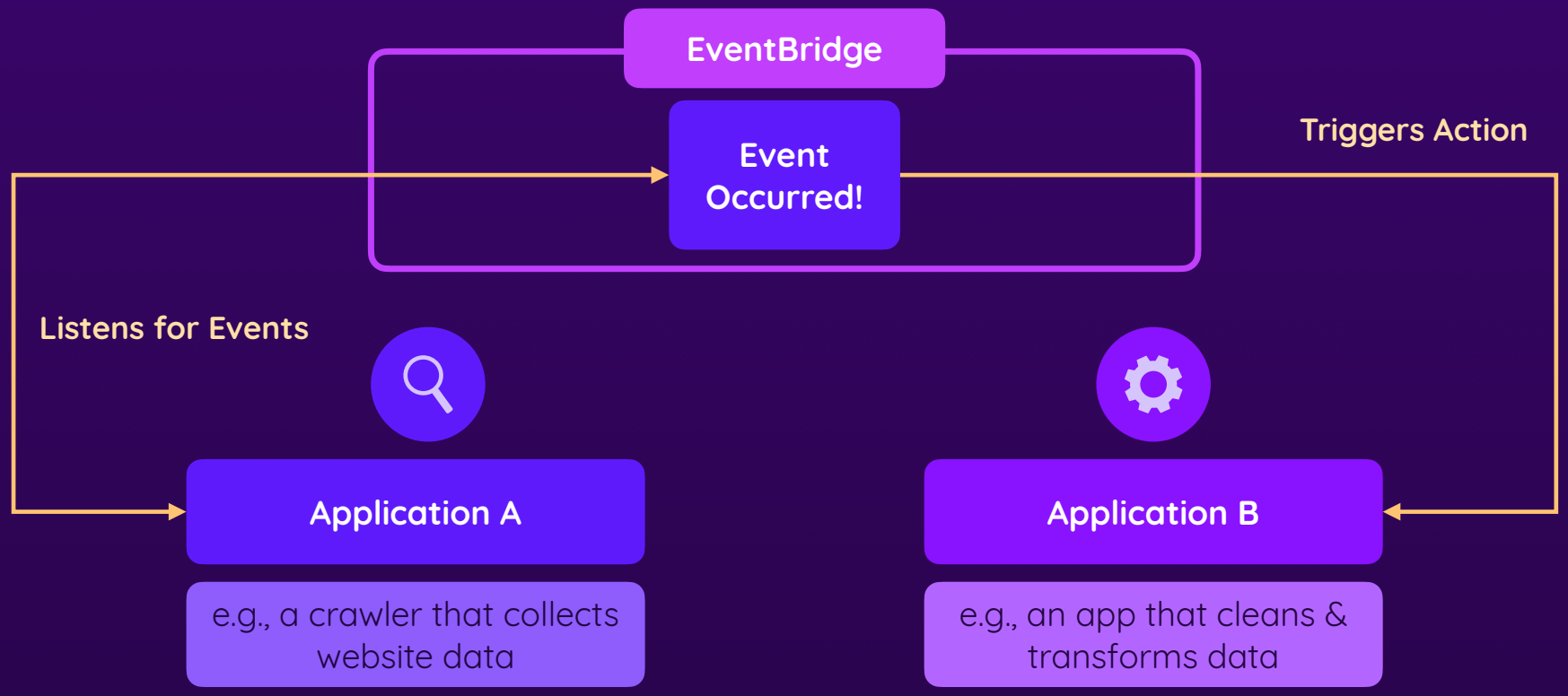
Understanding SQS



Understanding SNS



Understanding EventBridge



Indirectly Related: SES

Simple Email Service

Used to send transactional or batch emails to users

Not used for intra-service communication

For Advanced Scenarios: CloudMap

Advanced Application Discovery & Communication Service

Can be helpful for advanced micro-services architectures

CloudMap

Registry of (assigned) resource names

Resources (apps) can be referenced by name across applications

Understanding Step Functions

Reliable Pre-defined Execution Sequences

Define Steps	Combine Steps
Define step inputs, outputs & code to be executed	Define step relations & order of execution
Configure logging & execution permissions	Run in parallel, based on conditions or as alternatives

Summary

Apps Must Be Integrated!	SQS, SNS & EventBridge	Step Functions
Different apps often need to communicate with each other	SQS is a message queuing service (asynchronous)	Step Functions can be used to orchestrate (complex) workflows
Apps are not necessarily on the same server or network	SNS is a push notification service (synchronous)	Steps are linked to executable code / tasks
Building custom solutions (API) can be difficult & expensive	EventBridge is an event listening & processing service	Steps can be combined in various ways (e.g., conditional)
AWS offers dedicated integration services		

ACADE MIND

Monitoring

Keeping track of your services & applications

- Monitoring Goals & Requirements
- Understanding CloudWatch & More

Monitoring: What & Why?

Workloads should work as intended

e.g., a web server should be up-and-running and serving the website pages

But do they?

e.g., a server might crash or produce 4xx or 5xx HTTP errors

Proper monitoring allows you to react & fix issues early

Monitoring with CloudWatch

Managed Monitoring Service

Collect Data

- Application logs (e.g., web server logs)
- Service metrics & information
- Detailed data collection via API or **CloudWatch Agent**

Analyze Data

- Log insights and ServiceLens
- Charts & metric calculation
- Container & lambda insights

Act

- Forward metrics or logs (e.g., to S3 or external)
- Alarms

Get Alerted

Create CloudWatch Alerts

Select a Metric

- e.g., avg. bytes stored in a bucket in 1 day
- e.g., avg. cpu utilization on an EC2 instance in 5 minutes

Define Alert Conditions

- e.g., > 5,000 bytes
- e.g., > 80%

Define Alarm Action

- e.g., send notification via SNS
- e.g., trigger auto-scaling

Cross-Service Insights With XRay

In-app Request Flow Tracing

Prepare Apps For Tracing

- Install daemon service in app environment
- Save tracing events (e.g., handled requests, failures)
- Daemon sends tracing event batches to AWS

Analyze Traces

- View traffic flow across applications or services
- Combine with other monitoring tools

Summary

Monitoring: What & Why?	CloudWatch	Monitoring Other Services
Services & workflows might not always work as intended	CloudWatch is AWS's key monitoring service	Default metrics or logs are collected by most services
You want to detect & solve problems as early as possible	It collects data (logs, metrics) from other AWS services	Extra logging (paid) can be turned on for many services
Monitoring allows you to gain insights & solve problems	You can create dashboards, chart metrics & perform analyses	CloudWatch Agent or XRay daemon offer more details
Alerts make sure that you get notified about issues early	XRay & other advanced monitoring features are offered	

ACADEMIND

Managing Compute Workflows

Beyond basic use-cases & small companies

- Managing Compute Tasks At Scale
- Managing Instance & Server Fleets
- Managing Configuration & Parameters At Scale

Planning & Performing Batch Jobs

AWS Batch

Create Job Definitions	Execute Jobs
Define Fargate or EC2 instance jobs	Submit or schedule jobs
Define image & basic hardware requirements	AWS provisions resources & executes job
Extra configuration: Permissions, file systems, ...	Jobs & job status can be tracked

Optimizing Compute Resources

AWS Compute Optimizer

Uses machine learning to analyze CloudWatch metrics & resource configurations

Recommends improvements (e.g., to use a different instance type or memory settings)

Managing Large Scale Systems

Systems Manager

A service with many capabilities that help with managing large fleets of servers & applications

Node Management	Operations Management	Application Management	Change Management
Group, visualize & manage a fleet of servers	Manage server-wide operations	Manage application parameters	Manage fleet changes & updates
Connect to servers via Session Manager	Manage incidents	Provide & manage application configuration	Automate change requests
Orchestrate patches & server-wide commands	Fleet monitoring	Easily deploy or roll back configuration changes	Configure standardized maintenance windows

Provide Standardized Service Solutions

Problem

Not every account user should create a custom solution

Service Catalog

Create standardized, configurable AWS service usage templates

e.g., a VPC with an EC2 instance and a RDS instance

can be combined

Proton

Create standardized serverless & container deployments

e.g., a VPC with an ECS cluster on Fargate

Launch Wizard

Helps with launching standardized, pre-built (by AWS) applications

e.g., launch a SAP application

Summary

Size Matters

- Micro-management does not work for large-scale cloud usage
- Operating & monitoring individual services is not possible
- System-wide solutions are needed: Systems Manager etc.

Optimizing & Managing Compute Resources

- **Systems Manager:** Manage server fleets & all applications
- Manage updates, incidents or changes globally
- Perform batch operations with less effort via **AWS Batch**
- Optimize compute usage via **Compute Optimizer**

Standardizing Applications & Resources

- Account users shouldn't create different, custom solutions
- Standardized recipes via **Proton** or **Service Catalog**
- Pre-built (AWS-managed) apps via **Launch Wizard**

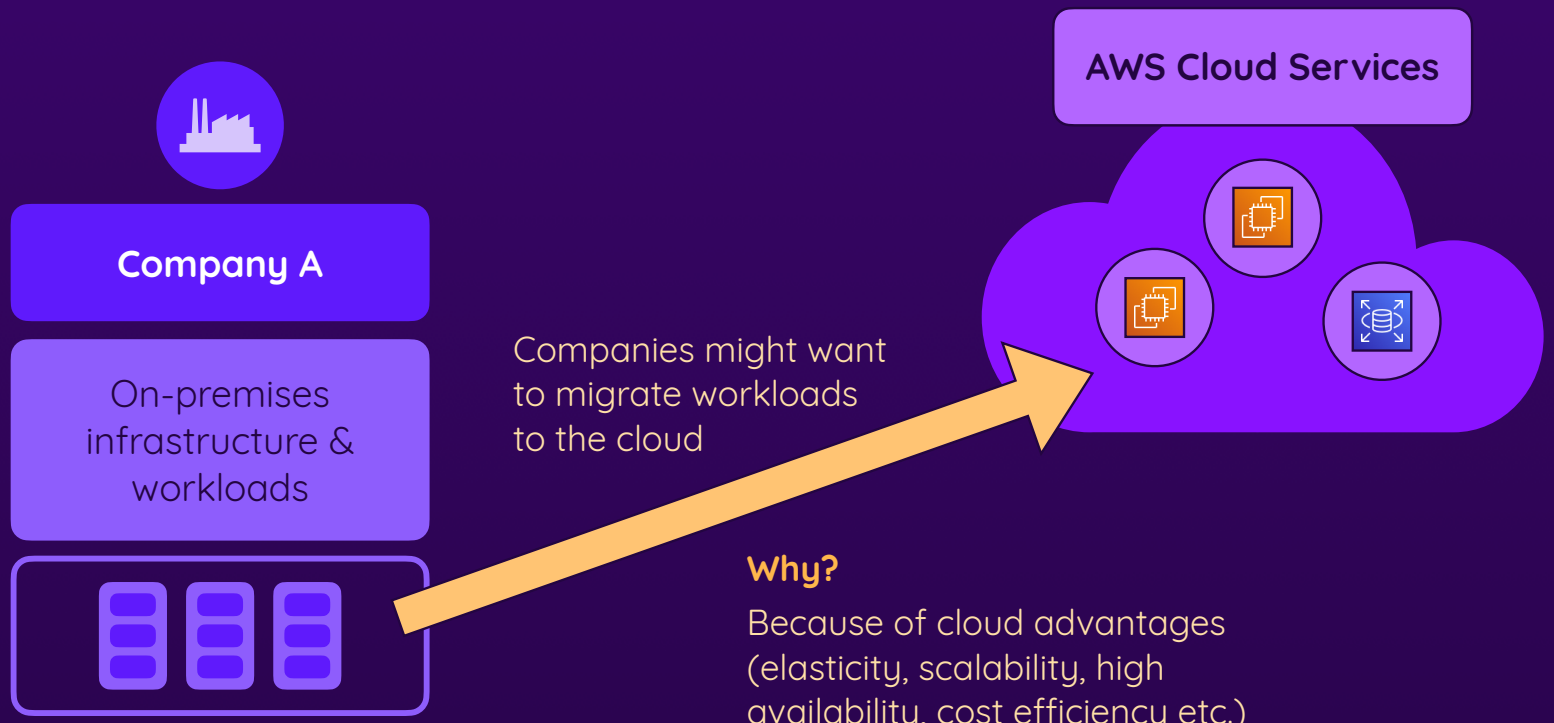
ACADEMIND

Migration & Hybrid Cloud Computing

From on-premises to cloud — or not

- How AWS Helps With Migration
- Key Services: Snow Family & Migration Services
- Building Hybrid Infrastructures: On-premises & Cloud

Migration: What & Why?



Why?

Because of cloud advantages (elasticity, scalability, high availability, cost efficiency etc.)

Migration Challenges

Workloads must be migrated **without interruption**

Expected **costs** must be **estimated**

Some workloads might need **adjustment** to run correctly (on cloud services)

Bonus

Some (or even all) workloads could be updated or rewritten to fully embrace AWS services & cloud benefits

Solutions & Migration Approaches

Migrate step-by-step, workload after workload	Use AWS & on-premises side-by-side During the migration period or forever (**Hybrid Cloud**)	Monitor & analyze migrated services & workloads
Start by migrating individual servers or basic workloads, then continue step by step	Connect AWS services to on-premise workloads & infrastructure	Use AWS monitoring & cost management services for insights
AWS Migration Hub, Application Migration Service, Database Migration Service ...	Storage Gateway, Outposts, Direct Connect, VPN, ...	CloudWatch, Cost Explorer, Budgets, ...

Key Migration Services

Migration Hub

A central place to track the overall migration process

Application Migration Service

Automated server application migration (agent software analyzes + replicates system)

Database Migration Service

Automated database migration (homo- and hetero-geneous + schema conversion)

DataSync

Synchronizes (copies) on-premises data with cloud data in EFS, S3 or FSx (e.g., via local NFS)

Transfer Family

Maps (S)FTP endpoints to S3 or EFS storage

Snow Family

Physical devices for moving data and / or performing compute tasks (at the edge)

Hybrid Cloud Computing

Run some workloads in the cloud & some on-premises

- Outposts
- Snow Family
- ECS / EKS Anywhere
- Storage Gateway
- DataSync & Transfer Family
- Systems Manager

Hybrid Cloud Computing

Outposts

AWS infrastructure, added to your local on-premises infrastructure

Snow Family

Portable devices, usable for data migration or edge computing

ECS / EKS Anywhere

Tooling & APIs for running ECS / EKS on local infrastructure

Storage Gateway

Interface for enabling on-premises workloads to use cloud storage (S3 only)

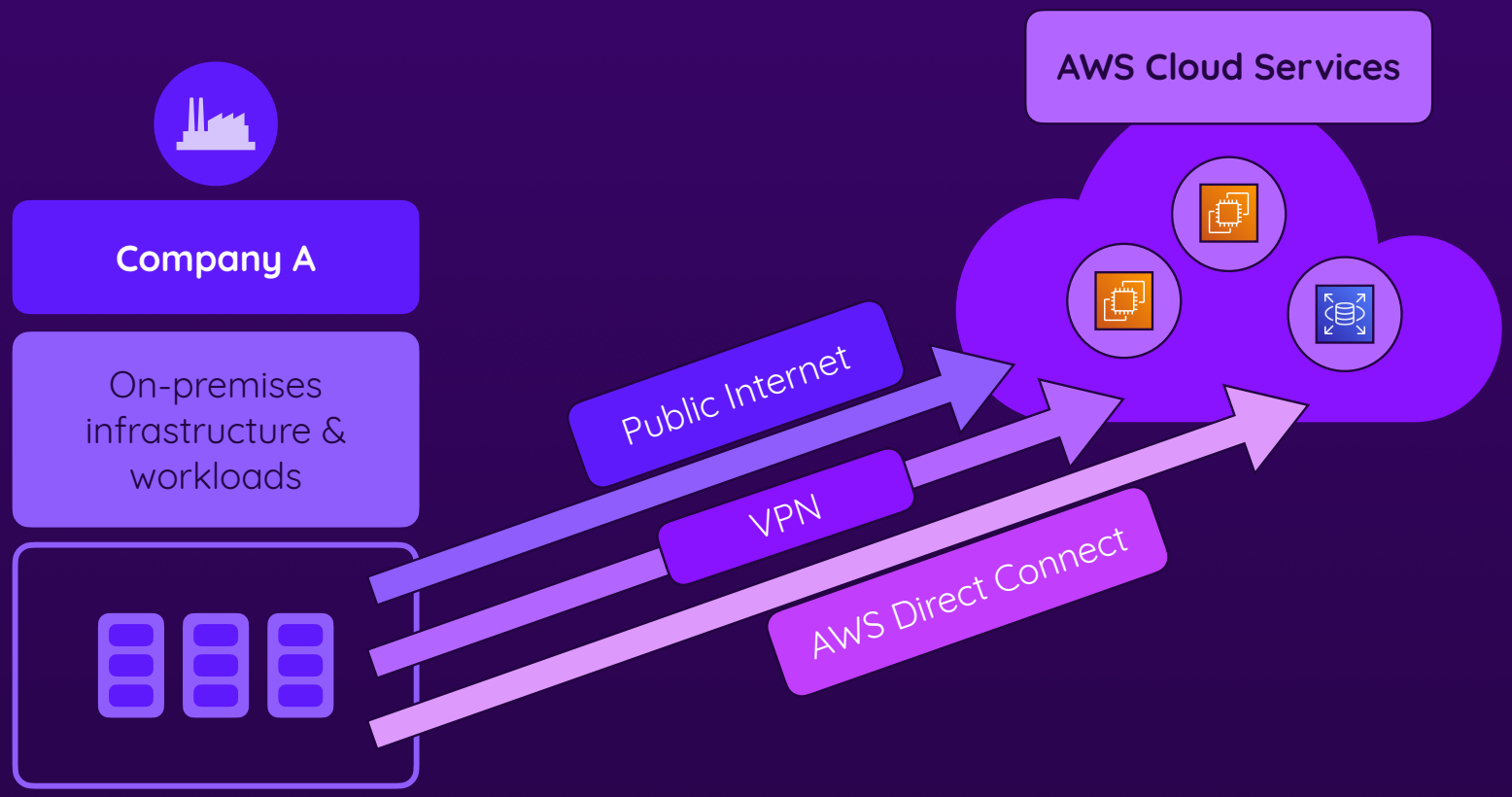
DataSync & Transfer Family

Service for syncing data between cloud & on-premises

Systems Manager

Manage large-scaling server fleets (EC2 & local), parameters, incidents & more

Different Connection Options



Different Connection Options

Public Internet

- Easy to use, no extra setup or costs
- Data transfers could be compromised

VPN

- Private network on top of the internet
- Higher protection due to private network
- **Client VPN**: Software solution for connection people to networks
- **Site-to-site VPN**: Uses **Transit Gateways** or **Virtual Private Gateways**

AWS Direct Connect

- Private, dedicated AWS-network connection
- Highest protection due to isolated network
- Extra costs
- Uses **Direct Connect locations**, **Virtual Private Gateways** & **Direct Connect Gateways**

Summary

A Challenge: Moving To The Cloud	Migration	Hybrid Cloud
Not all companies start “in the cloud”	Various migration services offered by AWS	Instead of going “all-in”, hybrid solutions could be preferred
Migration processes can be challenging	Application & Database Migration Services, DataSync etc.	Storage gateway, Systems Manager, Outposts etc.
AWS offers services that helps with migration	Transfer data via internet, VPN or Direct Connect	Transfer data via internet, VPN or Direct Connect
Companies could also aim for hybrid solutions		

ACADEMIND

Analytics & Data Science

Beyond applications & compute: anaylzing data

- Analytics & Data Science with Help of AWS
- Data Ingestion & Streams
- Transforming & Analyzing Data

Data Analytics: What & Why?

Modern businesses generate huge amounts of data

Structuring & analyzing such data efficiently can be a competitive advantage

AWS offers various services that help with gathering, storing, extracting, transforming & analyzing data

Ingesting Data

There's a broad variety of data sources

Applications	Crawlers & Scheduled Tasks	Devices & Sensors	Manual Data Entry
e.g., orders, user data, website analytics, logs, ...	e.g., crawled website data, weather data, ...	e.g., temperatures, movement speeds, ...	e.g., accounting data, documentation, ...

Ingestion Frequency

Slow	Moderate	Fast
e.g., manual data entry, weather data	e.g., user orders	e.g., website logs, sensor data

Ingesting Data with AWS

There's a broad variety of data sources

Applications	Crawlers & Scheduled Tasks	Devices & Sensors	Manual Data Entry
Application code can write data to S3, RDS etc.	(Batch) Tasks can store data to S3, RDS etc.	Kinesis helps with handling data streams	Backend can write data to S3, RDS etc.

Ingestion Frequency

Slow	Moderate	Fast

Ingesting Streaming Data with Kinesis

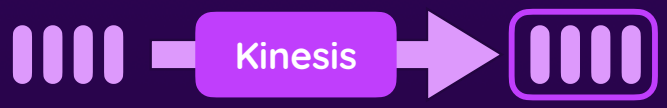
What are "Data Streams"

High-frequency data ingestions

e.g., multiple data points per second or millisecond

AWS Kinesis is able to accept incoming data at a fast rate and forward it to other services without overwhelming them

Kinesis forwards bulk data and hence throttles data ingestions



e.g., RDS

A Closer Look At AWS Kinesis

Collection of features that simplify dealing with data streams (continuous high frequency data ingestion)

Kinesis Data Streams

Scalable service which captures incoming streaming data

Kinesis Firehose

Loads data into other AWS services (e.g., S3)

Kinesis Data Analytics

Perform real-time data stream analytics

Storing Data: Data Lakes & Warehouses

both might be needed

Data Lakes	Data Warehouses
Store all (structured & unstructured) data in the same place	Store formatted & structured data in data warehouse database
Key AWS Service: **S3**	Key AWS Service: **Redshift**
Great for machine learning & data discovery	Great for business intelligence, reporting & visualizations

EMR	Glue	Athena	Redshift	OpenSearch	QuickSight
Extraction & transformation		Data querying	Store	Data search	BI

A Data Warehouse Solution: Redshift

A Scalable & Flexible Data Warehouse

A data warehouse database

SQL-based database, optimized for analytics usage

Store data & run analytics queries against data

RDS: Transactional data / daily operations
Redshift: Analytics data / data warehouse

Redshift can also be used to query data stored in other data sources (e.g., S3)

Extracting & Transforming Data with Glue

Serverless, managed ETL service

ETL: Extract, Transform, Load

Simplifies the process of crawling, parsing & transforming raw data

Main features: Data schema creation, data transformation jobs, visual job editor

Self-managed Big Data Computation: EMR

EMR
Elastic Map Reduce

A service that simplifies spinning up your own big data compute clusters

Environment Setup

Creates a compute environment (e.g., cluster of EC2 instances)

Run Big Data Workloads

You choose a big data framework (e.g., Apache Spark)

Analyzing Data with Athena & QuickSight

Athena

- Query data in S3 (via SQL queries)
- No data movement to databases required
- Standard SQL commands supported
- Other data sources are also supported (e.g., CloudWatch logs, DynamoDB)

QuickSight

- A business intelligence service
- Build charts, reports & dashboards
- Perform various analyses (average, sums etc.)

Searching & Visualizing Data

OpenSearch	Grafana	CloudSearch
Managed search service for searching & analyzing data	Managed Grafana service (helps with visualizing data)	Not primarily focused on data analysis
Running OpenSearch server (“domain”) can be used to connect & search or analyze data	Simplifies the creation of live, interactive data visualizations	Instead: Managed service that simplifies the creation & management of website or application search solutions

Summary

Utilizing Data: A Complex Problem	Data Ingestion & Storage	Transformation & Analytics
Data must be ingested, transformed, stored & analyzed	**Kineses** helps with ingesting high-frequency streaming data	Manual big data workloads can be executed with **EMR**
Data ingestion can be tricky because of frequency / size	Data is buffered and (typically) forwarded to other services	**Glue** is a managed, serverless ETL solution
Transformation & extraction tasks require efficient compute	Data is often stored on **S3**, following a “Data Lake” approach	Query raw data (e.g., in S3) with SQL & **Athena**
Different analytics tasks need different tools (search vs visuals)	Data warehouses can be built with **Redshift**	Perform BI with **QuickSight**, search & visual with **OpenSearch** & **Grafana**

ACADEMIND

Cloud Management

Managing complex cloud environments efficiently

- Managing Multiple Accounts
- Deploying & Configuring Services Efficiently
- Managing Cloud Configuration At Scale

Using Multiple AWS Accounts

Possible
One account, multiple IAM users

Can be better
Multiple accounts (with one or multiple IAM users each)

- Use account limits multiple times (e.g., max number of Elastic IPs)
- Use free tier or monthly free usage multiple times (e.g., monthly free Lambda executions)
- Improve organization & management by splitting by workload, team or other criteria

Multiple Accounts & Organizations

Manage multiple accounts via **AWS Organizations**

Advantages: Centralized billing, centralized management, use cross-account service configurations & more

Group accounts into organizational units (OUs) and enforce policies

Consolidated Billing

Multiple accounts, one bill

Utilize “Consolidated Billing” to get & pay a single bill for multiple accounts

Track charges across accounts & create consolidated reports

Share savings plans or volume pricing discounts across multiple accounts

Organizations & Control Tower

Manage multiple accounts via **AWS Organizations**

Advantages: Centralized billing, centralized management, use cross-account service configurations & more

Group accounts into organizational units (OUs) and enforce policies

Create a **brand-new multi-account** environment

Creates a best-practice multi-account environment for you

A great starting point for a multi-account strategy / company

Budgets & Cost Management

Manage bills and costs with **Cost Explorer** & **Budgets**

View your bills (daily updated) and analyze your costs with the cost management tools provided (e.g., **Cost Explorer**)

Set budgets & alerts to control spending

Creating Cloud Resources Manually Is Bad

Creating cloud resources (i.e., using cloud services) manually is great for getting started, basic use-cases etc.

But for **large companies & complex use-cases**, it's typically **not the best solution**

- Repetitive & slow
- Error-prone & hard to debug
- Possibly different solutions for the same problem

Using CloudFormation

Manage cloud resources with **Infrastructure as Code**

Model & define your cloud infrastructure declaratively

Configure dependencies & dynamic parameters

Automate infrastructure deployments & updates

Service & Workload Configuration

Manage Workloads Centrally via Systems Manager

- Manage application parameters via **AppConfig** or **Parameter Store**
- Manage server fleets from a central place (e.g., issue commands, update, changes)

Define & Share Standardized Cloud Resources

- **Proton** for serverless compute (Lambda) & containers
- **Service Catalog** for multi-service solutions
- **Resource Access Manager (RAM)** for shared resources

Control Service Configuration

- Use **AWS Config** to enforce service configurations
- Manage software license requirements via **AWS License Manager**

ACADEMIND

Summary

Cloud Environments Can Become Complex

- Multiple accounts may be used to separate workloads or teams
- Configuring & controlling multiple accounts can be difficult
- **AWS Organizations** helps with managing multiple accounts
- Many services support multi-account environments by default

Working with Multi-Account Environments

- Use **AWS Organizations** (and **Control Tower** to get started)
- Many services like **Backup** support **Organizations**
- Create OUs and enforce organization-wide policies
- Share resources via **RAM**
- Manage billing centrally

Managing Workloads & Services

- Deploy environments with **CloudFormation**
- Manage applications via **Systems Manager**
- Application configuration via **AppConfig** or **Parameter Store**
- Standardized cloud "products" via **Proton** & **Service Catalog**
- Managing service configuration via **AWS Config**

ACADEMIND

Security & Compliance

Securing your account, services & applications

- Account Security: It's More Than IAM
- Securing Applications, Traffic & Data
- Reacting To Threats & Handling Incidents

Security Matters – Everywhere

Account Protection

- Account must not get compromised
- Prevent malicious account / service usage
- Secure cross-account service usage

Compliance & Standardization

- Single Sign-On, service config, compliance reports

Application Protection

- Detect application / software vulnerabilities
- Detect insecure configurations
- Investigate security issues & incidents

Network Protection

- Detect malicious network traffic
- Protect against DDoS attacks

Data Protection

- Encrypt data at rest & in transit
- Protect code secrets
- Prevent unintended data exposure

ACADEMIND

Security Matters – Everywhere

Account Protection

- IAM & SSO
- CloudTrail
- GuardDuty
- RAM
- Organizations

Compliance & Standardization

- Artifact
- Config, Audit M.

Application Protection

- Inspector
- Detective

Network Protection

- WAF
- Network Firewall
- Firewall Manager
- Shield

Data Protection

- KMS, CloudHSM
- Secrets Manager
- ACM
- Macie

Managing Permissions with IAM

Manage Identities & Access Rights

Define & Manage Identities	Control Permissions
Users, user groups & roles	Permissions are defined via policies
Attach permissions (policies) to identities	Pre-defined policies provided by AWS
By default: No permissions are added to any identity	You can create your own policies
Explicit deny > explicit allow	Multiple policies can be combined

User Authentication

Single Sign-On & Active Directory

Single Sign-On

- Simplify signing into AWS accounts
- Use AWS credentials or other sources

AWS Directory Service

- Use Active Directory for authentication
- Helps with connection or migrating AD workloads

Track & Protect Account Usage

Prevent Malicious Usage

Track API Usage

- **CloudTrail** allows you to track AWS API calls
- Identify identities and their actions

Detect Malicious Patterns

- Detect suspicious behavior via **GuardDuty**
- Uses machine learning to detect and surface issues

Cross-Account Service Usage

Manage Multiple Accounts & Their Resources

Combine & Manage Accounts	Share Resources Cross-Account
Use **Organizations** to combine multiple accounts	Share resources via **Resource Access Manager**
Workload separation with global management	Ideal with **Organizations**: Create centrally, use locally
Organization-wide policies & rules can be enforced	e.g., create a VPC and share with other accounts

Stay Compliant & Meet Legal Requirements

Enforce & Prove Compliance

Enforce Compliance & Best Practices

- Use **AWS Config** to define & track service configuration
- Enforce organization policies & guidelines
- Monitor & resolve configuration deviations

Prove AWS Compliance

- Download compliance reports via **AWS Artifact**
- Prove AWS compliance with regulations & rules

Prove Your Compliance

- Track compliance issues with **Audit Manager**
- Generate auditor-friendly reports
- Connect with AWS Config for data collection

Protecting Applications with Inspector

Automated Vulnerability Management

Account-wide Vulnerability Scanning

- Enable for single- or multi-account scanning
- Automatically discovers vulnerabilities & issues
- Analyzes containers & EC2 instances

Detailed Insights for Instances & Containers

- Learn which instances or containers are affected
- Information about the kind of vulnerability
- Provides vulnerability details

Manage Incidents with Detective

Investigate Incidents

Explore AWS Resources

- Search for users, instances, roles & more
- Explore actions by / on resource
- Get a list of automatic findings

Analyze Findings & Actions

- Explore finding details (date, involved resources, ...)
- Explore details for suspicious activity

Analyzing Network Traffic with Firewalls

Blocking Unwanted Traffic

Web App Firewall (WAF)	Network Firewall
Inspect HTTP(S) traffic and block it based on content	Inspect any traffic and protect entire networks
Analyze metadata & request bodies	Analyze IPs, ports, protocol etc.
Define rules for blocking traffic	Define stateful or stateless rules for blocking traffic

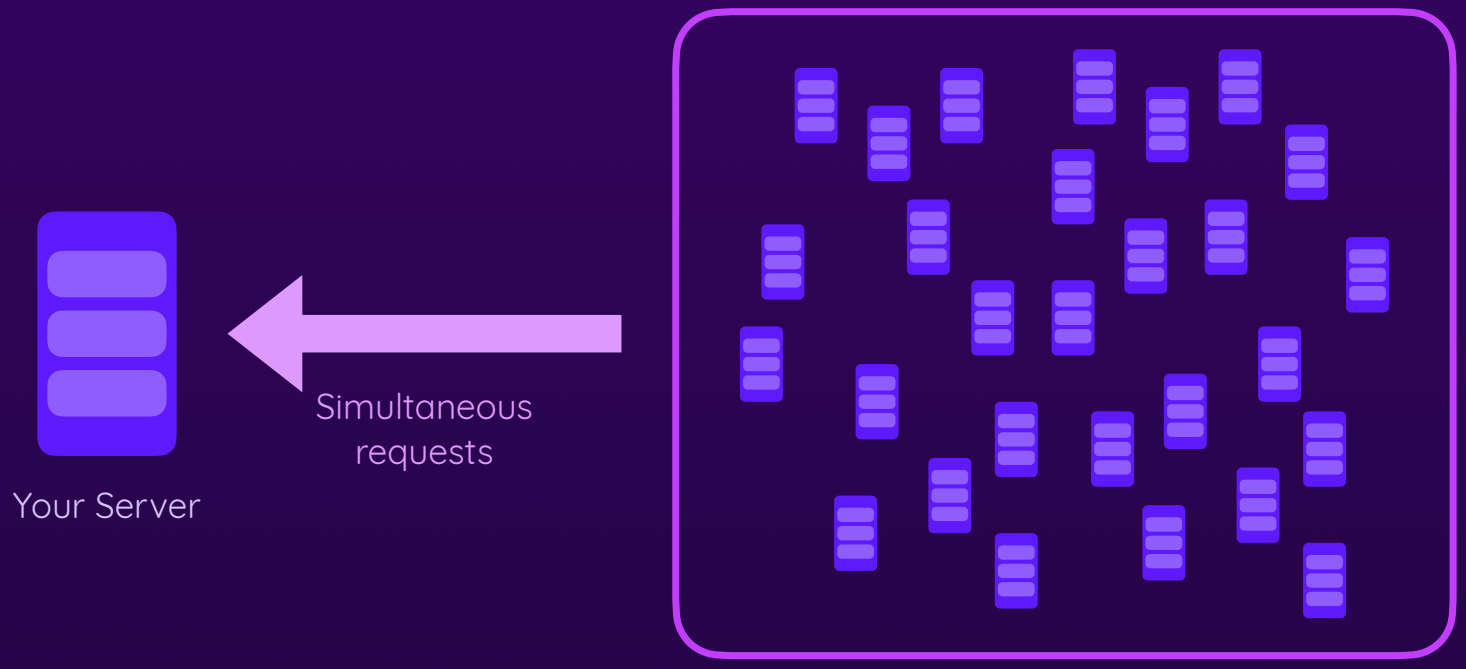
Global Firewall Management via **Firewall Manager**

Avoid DDoS Attacks

Distributed Denial of Service

An attacker sends a huge amount of simultaneous requests to your server

Typically via a network of (hacked) bot machines



Protecting Against DDoS Attacks

DDoS Protection via Shield

AWS Shield Standard	AWS Shield Advanced
Free & enabled by default	Monthly cost, not enabled by default
Basic DDoS protection based on network flow	Customizable protection rules
No anomaly detection	Anomaly detection & dedicated AWS support

Encrypting Data

Encrypt Data — At Rest & In Transit

At Rest	In Transit
Encrypt data via **KMS** or **CloudHSM**	Encrypt network traffic with **ACM**
Automatic encryption & decryption	Use with services like CloudFront or ALB
Control encryption across many AWS services	Get & use free SSL certificates
KMS: AWS-managed keys **CloudHSM**: Custom key store	

Managing Code & Application Secrets

Securely manage Secret Parameter Values

Manage Secrets	Use Secrets
Securely store secret values with **Secrets Manager**	Access secret values from inside application code
Built-in auto-rotation support for RDS & more	Access or set secrets via other services
Control access permissions	

Protecting Sensitive Data with Amazon Macie

Discover Data Protection Issues with Amazon Macie

Configure & Use

- Detect sensitive data via machine learning
- Add custom-defined sensitive data types
- Scan data on demand or on a schedule

Monitor & Discover

- Macie highlights exposed or unprotected sensitive data
- e.g., detect unencrypted or public sensitive data

Using Security Hub

Consolidated Security Status Management

Consolidate Other Security Services

- Group GuardDuty, Inspector & Macie output
- Control security service behavior centrally

Take Action

- Take action across services & accounts
- Build customized actions

ACADEMIND

Summary

Security Matters — Always!

- A secure cloud environment is a combination of things
- Protect your account & ensure compliance
- Protect applications, traffic & data (and therefore your users)
- Use different services & service combinations for full protection

Account Security & Compliance

- Use **IAM** for managing identities & permissions
- Use **CloudTrail** & **GuardDuty** to detect & track suspicious actions
- Use **SSO** & **Managed Directory Service** for advanced login
- Use **Organizations** & **RAM** to manage multi-account setups
- Be compliant with **Artifact**, **Audit Manager** & **AWS Config**

Application, Traffic & Data Security

- Secure applications with **Inspector** & **Detective**
- Secure traffic with Firewalls (**WAF**, **Network Firewall** & more)
- Protect against DDoS with **Shield**
- Encrypt your data with **KMS** or **CloudHSM**
- Protect data with **Secrets Manager** & **Macie**

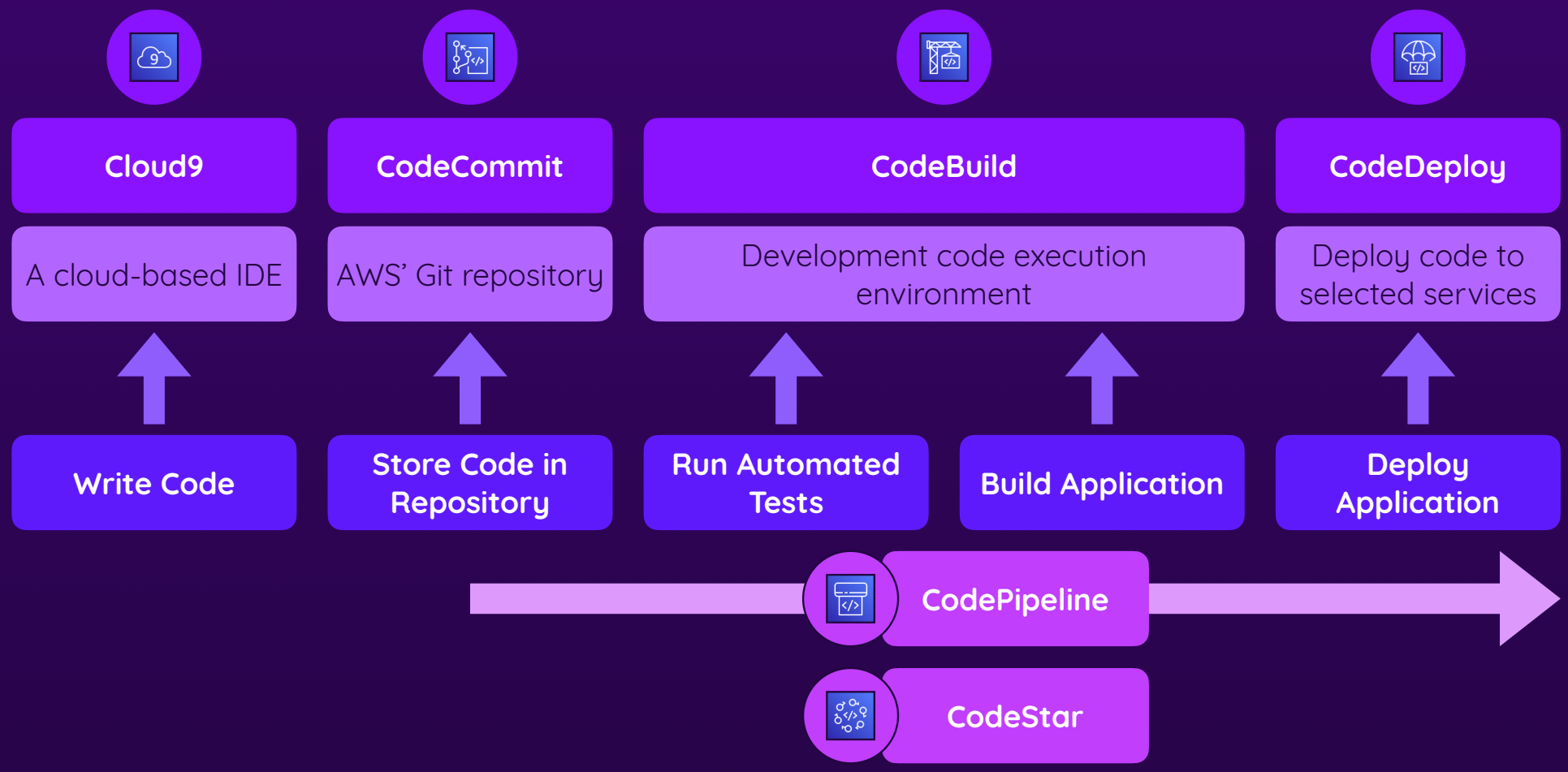
ACADEMIND

Developer Tools

Supporting developers & simplifying deployments

- Building Applications in & with the Cloud
- Improving Deployment Workflows
- Helper & Simplification Services

Building an Application



Writing Code: Cloud9 & CodeCommit

Cloud9

- Write code in a cloud-based IDE
- Uses an EC2 instance under the hood

CodeCommit

- Push Git commits (code changes) to a cloud code repository
- Code is stored in private repositories

Test & Build Code with CodeBuild

Managed Execution Environment

Configure Environment

- Define code source
- Define execution environment (OS, software)
- Add timeout, environment variables & more

Configure Execution Steps

- Define a “buildspec” file with execution process details
- Define build output (“artifacts”) details
- Run manually or based on triggers

CodeArtifact

- Managed repository for private and public application packages
- Can be used during code build process
- Alternative to public / third-party repositories

Deploy Code with CodeDeploy

Managed Deployment Service

Configure Deployment

- Configure deployments for EC2, ECS & Lambda
- Choose from different deployment strategies
- Use managed or custom configurations (strategies)

Perform Deployment

- Run deployments manually or via triggers
- Monitor, retry or roll back deployments

CI / CD with CodePipeline

Managed CI / CD Pipeline

Combines the other Code services

Define Stages

- Define different CI / CD stages (test, build, deploy, ...)
- In the stages: Use CodeBuild, CodeDeploy etc.
- Optionally add manual approval or script stages

Execute Pipeline

- Execute manually or via triggers (source changes)
- Enable or disable transitions between stages
- Monitor & retry pipeline executions

CodeStar

- Simplified CI / CD workflow setup (uses Code services)
- Create projects based on templates
- Creates build & deployment steps automatically

Improving Code & Application Environments

CodeGuru

- ML-based code analysis & recommendations
- Analyses code for best practices & issues

DevOpsGuru

- ML-based analysis of running applications
- Analyses CloudWatch logs etc.
- Detects (potential) issues & provides recommendations

Summary

Developing Applications Is A Multi-Step Process	Write, Store, Build & Deploy Code	Manage Entire Code-based Workflows
It includes: Writing, storing, testing, building & deploying code	Write code via **Cloud9**, store via **CodeCommit**	Integrate all build steps via **CodePipeline**
All steps can be performed or initiated locally	Manage code artifacts (packages) via **CodeArtifact**	Define stages (source, testing, build, deploy, manual approval, ...)
Use cloud-based tools: Better performance, always available	Test & build code with help of **CodeBuild** (output via S3)	Define triggers & monitor pipeline executions
Another advantage: Shared environment & settings	Deploy code to EC2, ECS or Lambda via **CodeDeploy**	Simplified alternative: **CodeStar** (uses Code services)

ACADEMIND

Other Services

Advanced & niche services

- Machine Learning
- Internet of Things (IoT)
- Business Applications & More

Machine Learning

AWS offers a broad selection of ML services

Self-managed ML Workloads

- Run your own servers / tasks with your own machine learning processes
- Use servers (EC2/ EMR) or containers (e.g., via ECS)

SageMaker

- Managed infrastructure & tools for developing and testing machine learning models
- Code & visual editor, beginner friendly
- Create & deploy machine learning models

AI Services

- Finished AI services for routine tasks like image processing, text extraction etc.

Amazon Connect

A Managed Contact Center

A managed contact center: real agents, supported by ML features

ACADEMIND

Best Practices by AWS

Using AWS “correctly”

- The Well-Architected Framework
- Automatic Checks & Recommendations
- Terms of Use

The Well-Architected Framework

Cloud Best Practices

A framework that describes best practices

Helps you analyze your cloud infrastructure

Analyze your infrastructure with help of **six pillars**

Well-Architected: Six Pillars

Operational Excellence	Security	Reliability	Performance Efficiency	Cost Optimization	Sustainability
Support development, run workloads effectively	Use cloud technologies to protect accounts, data, workloads	Ensure consistent workload functionality	Use (computing) resources efficiently	Optimize cost & keep track of spending	Optimize energy consumption & improve efficiency
Gain insights & continuously improve processes		Workloads should function under different circumstances	Embrace & utilize demand and technology changes		

Trusted Advisor

Automatic Cloud Resources Analysis

- Automatically checks resources & offers recommendations
- Provides cost saving, security & other recommendations
- Additional checks via higher-level support plans

ACADEMIND

Beyond AWS Services

More things you should know (for the exam)

- The AWS Partner Network
- The Marketplace
- AWS Professional Services

The AWS Partner Network (APN)

A network of registered AWS users & service vendors

Get approved (and endorsed) by AWS

Sell services, software or hardware via the AWS marketplace & sites

The AWS Marketplace

Sell or buy SaaS solutions or professional services

↓

Buy AMIs, pre-built ML models, data analytics software & more

AWS Professional Services

Get professional support by AWS

Get help for complex use-cases and cloud migration tasks

Can be backed up with professional services by AWS partners

AWS Managed Services

Ongoing professional support by AWS

Professional services: Projects

Managed services: ongoing support

e.g., incident management, cost savings, monitoring support etc.

Preparing For The Certification Exam

- Watch the videos
- Learn with help of the slides
- Take the practice exam
- Explore the official exam guide & example questions
- Go through service FAQs & documentation
- Dive into AWS whitepapers & blog posts

What You (Don't) Need To Know

To pass the Cloud Practitioner exam

What you don't need to know	What you should know
No in-depth service knowledge required	What AWS & cloud computing is about
Questions are mostly about key services, not about niche services	Why using the cloud could be beneficial (key advantages)
You don't need to know all configuration options or defaults	What the key AWS services are about (problems solved)
You don't need to be able to come up with complex solutions or example cloud architectures	How key services differ from each other (e.g., EC2 vs Lambda)

Example & How To Answer Questions

In order to be prepared for increased website traffic, a data center administrator deploys more IT resources than currently required.

Which cloud computing advantage could be utilized here?

- Protect data in transit
- Shared Responsibility Model
- Stop guessing capacity
- Go global in minutes

How To Answer Questions

In order to be prepared for increased website traffic, a data center administrator deploys more IT resources than currently required.

Which cloud computing advantage could be utilized here?

- Protect data in transit
- Shared Responsibility Model
- **Stop guessing capacity**
- Go global in minutes

Max' Exam Recommendations

90 minutes is plenty of time	Take your time, don't get stressed & don't rush through the exam!
Don't get stuck	If you're not sure about a question, don't waste time on it
You can mark questions for "review"	Come back to those questions after answering all other questions
Not sure? Guess!	There's no penalty for guessing!

Share Your Success!

Share your success with me!

Let me know on Twitter @maxedapps!