

Getting Started with AWS and Cloud Computing

Seshagiri Sriram

Agenda

Introduction to Cloud and AWS
AWS services

Compute

Storage

Databases

Networking

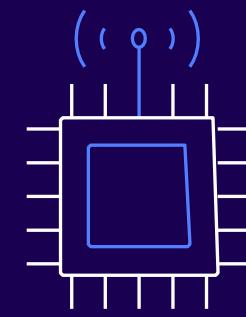
Security

AI and ML with AWS

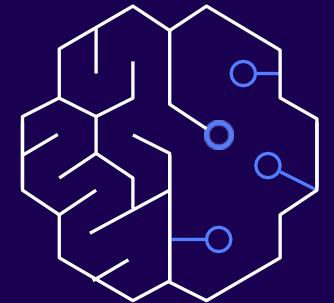
Deploying Apps in AWS

Next steps

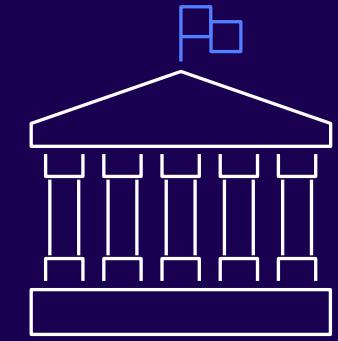
Innovation with AWS



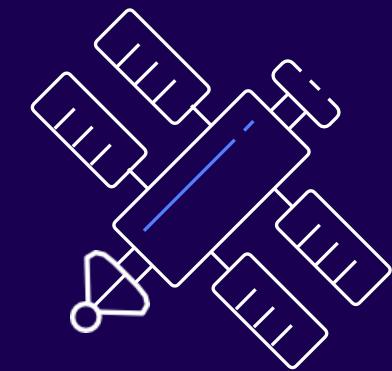
Internet of
Things
(IoT)



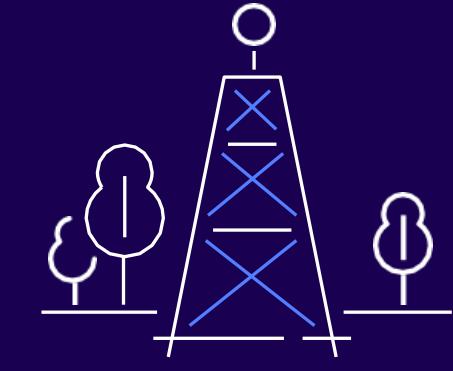
Machine
learning
(ML)



Blockchain



AWS Ground
Station

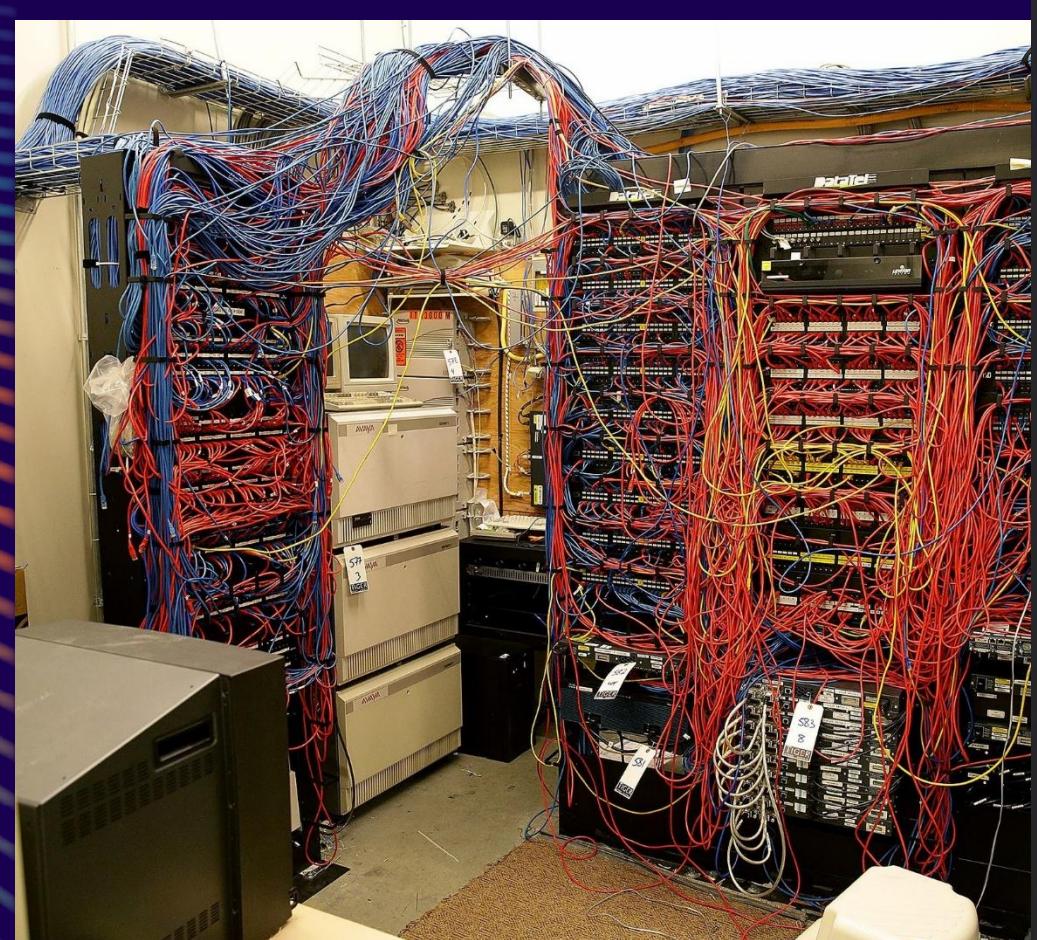


AWS
Wavelength

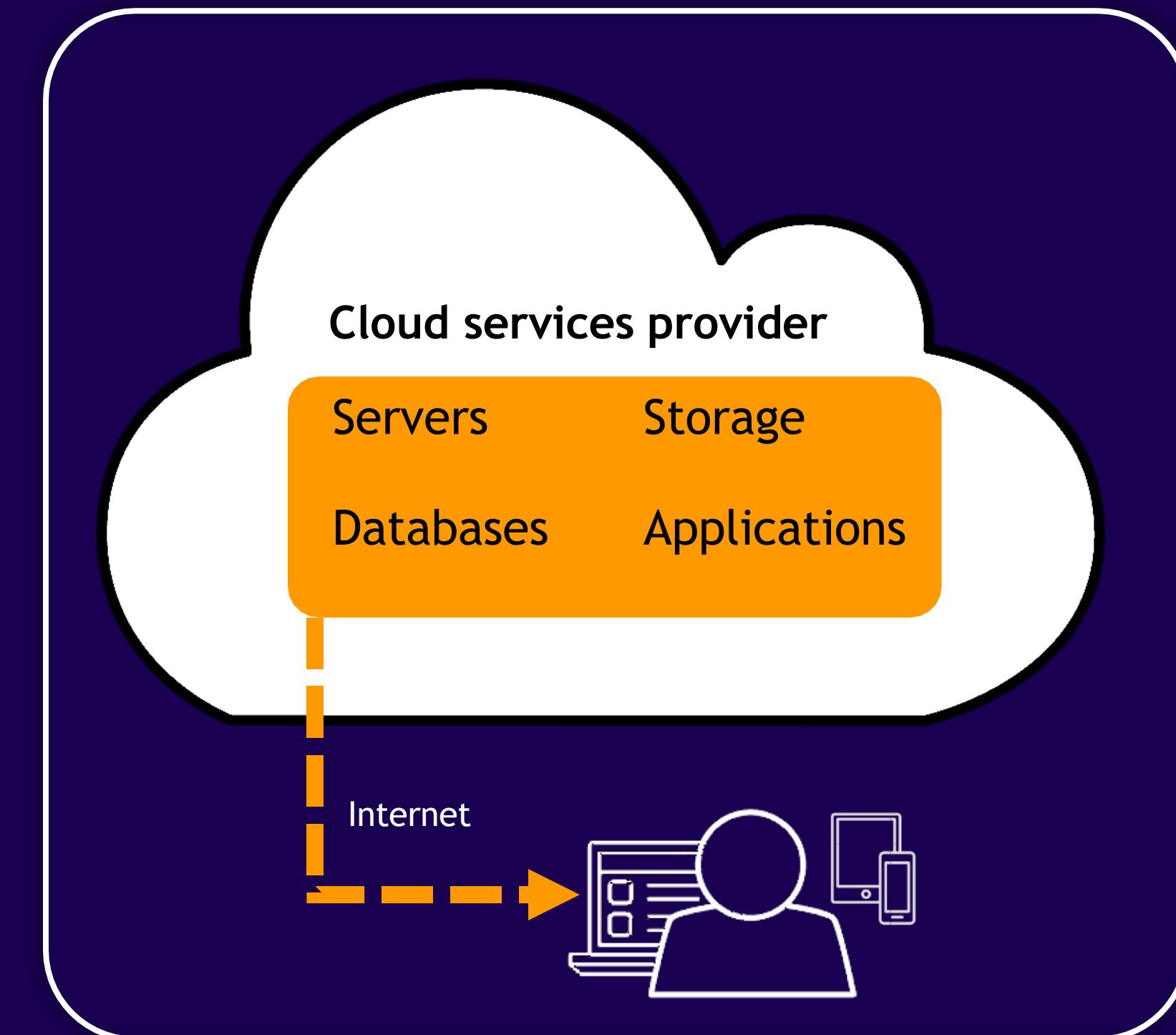
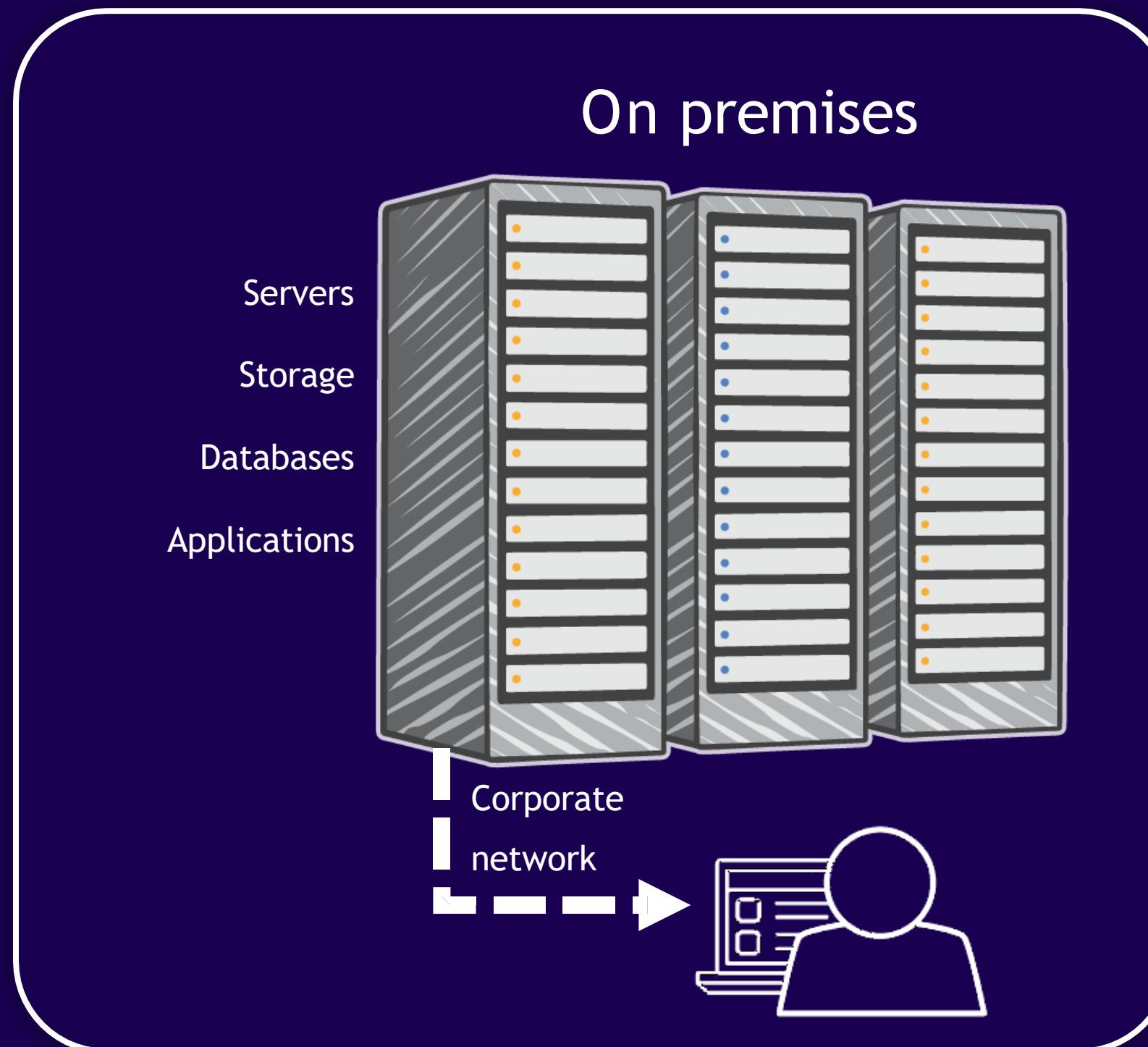
Introduction to Cloud & AWS

Part 1: Introduction to the cloud

From Traditional Server Rooms to the Cloud



What is the cloud?



What is the cloud?

cloud com-put-ing

noun

the practice of using a network of remote servers hosted on the Internet to store, manage, and process data, rather than a local server or a personal computer.

On-Premise

- You own the servers
- You hire the IT people
- You pay or rent the real-estate
- You take all the risk

Cloud Providers

- Someone else owns the servers
- Someone else hires the IT people
- Someone else pays or rents the real-estate
- You are responsible for your configuring cloud services and code, someone else takes care of the rest.

What is the cloud?



Everything is virtualized SW

Everything accessible over network

Use and pay as Required.

Services and Resources Managed for you.

What is the cloud?

Delivery of Computing Services

Software & Compute

Analytics

Devices - Servers, Printers, IoT

Storage

Databases

Networking

Security

Monitoring

...

Over the Network/Internet in a virtualized Manner

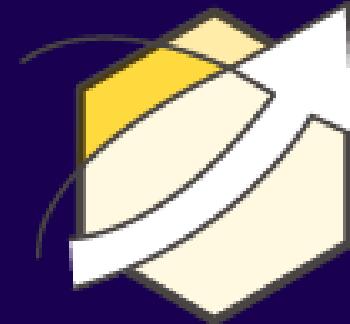
Using API

as Services, paying as you go

What is the cloud?

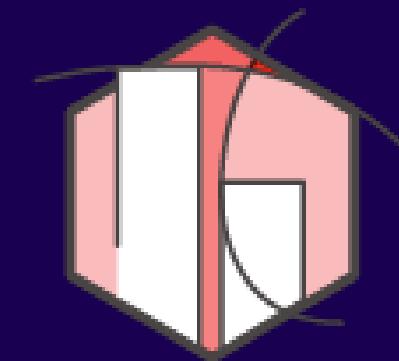
With cloud computing, you can stop thinking of your infrastructure as hardware, and instead think of it (and use it) as software.

Agility



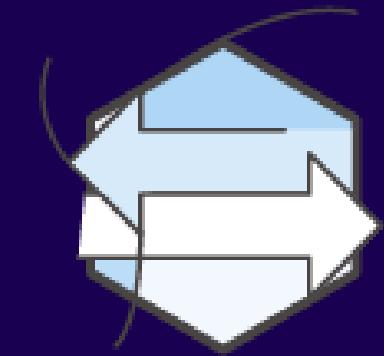
On-demand self-service

Elasticity



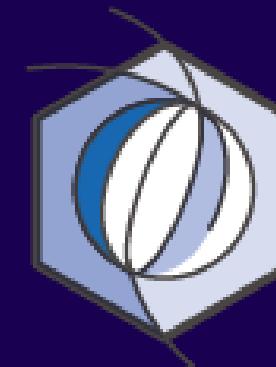
Scale rapidly to meet demand

Cost Savings



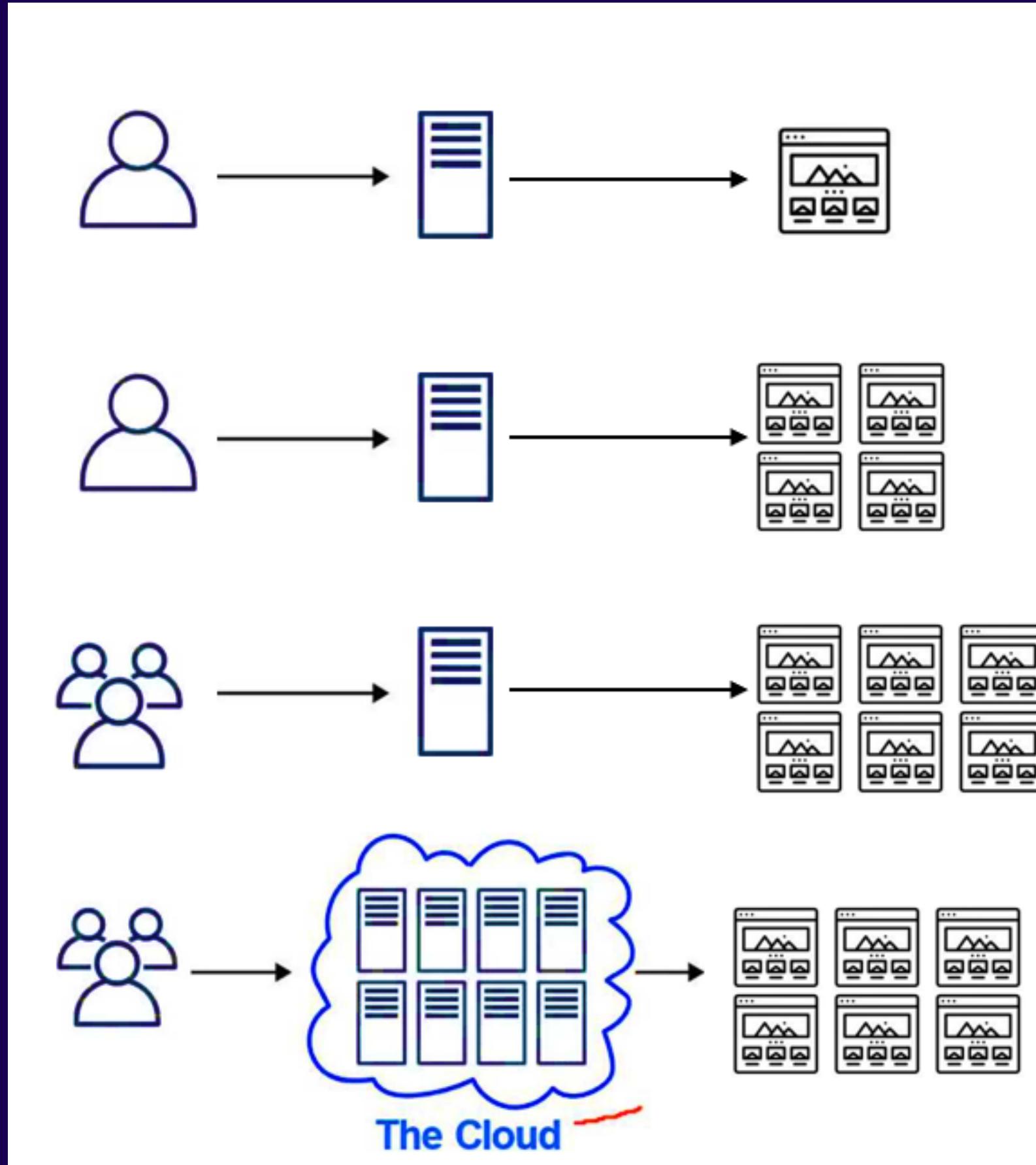
Only pay for IT as you consume it

Deploy globally



Broad network access

The Evolution of Cloud Hosting



Dedicated Server

One physical machine dedicated to single a business.
Runs a single web-app/site.
Very Expensive, High Maintenance, *High Security

Virtual Private Server (VPS)

One physical machine dedicated to a single business.
The physical machine is virtualized **into sub-machines**
Runs multiple web-apps/sites

Better Utilization and Isolation of Resources

Shared Hosting

One physical machine, shared by hundred of businesses
Relies on most tenants under-utilizing their resources.
Very Cheap, Limited functionality, Poor Isolation

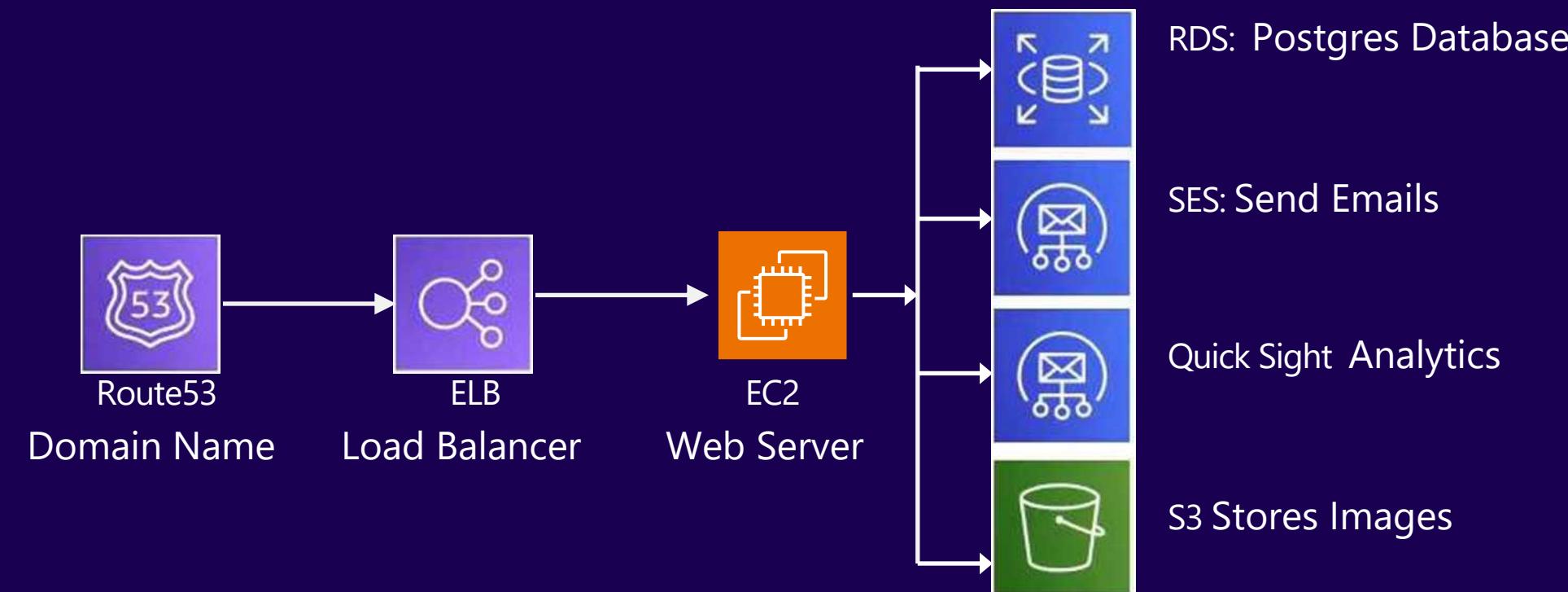
Cloud Hosting

Multiple physical machines that act as one system.
The system is abstracted into multiple **cloud services**
Flexible, Scalable, Secure, Cost-Effective, High Configurability

What is a Cloud Service Provider (CSP)?

A **Cloud Service Provider (CSP)** is a company which

- provides multiple Cloud Services e.g. tens to hundreds of services
- those Cloud Services **can be chained together** to create cloud architectures
- those Cloud Services are accessible **via Single Unified API** eg. AWS API
- those Cloud Services utilized **metered billing** based on usage e.g. per second, per hour
- those Cloud Services have rich monitoring built in eg. AWS CloudTrail
- those Cloud Services have an Infrastructure as a Service (IaaS) offering
- Those Cloud Services offers **automation** via Infrastructure as Code (IaC)



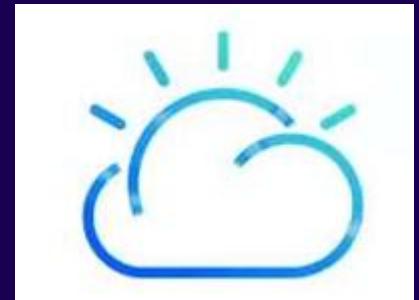
If a company offers multiple cloud services under a single UI but do not meet most of or all of these requirements, it would be referred to as a Cloud Platform e.g. Twilio, HashiCorp, Databricks

Landscape of CSPs

Tier-1 (Top Tier) - Early to market, wide offering, strong synergies between services, well recognized in the industry



Tier-2 (Mid Tier) - Backed by well-known tech companies, slow to innovate and turned to specialization



IBM Cloud

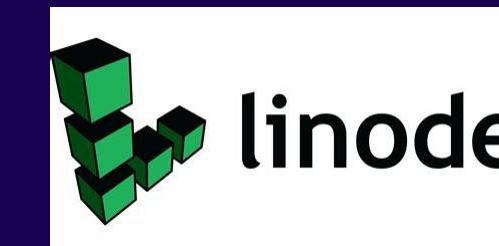


Oracle Cloud



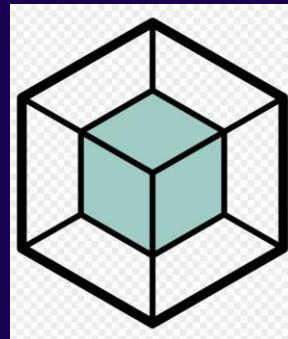
Rackspace (OpenStack)

Tier-3 (Light Tier) - Virtual Private Servers (VPS) turned to offer core IaaS offering. Simple, cost-effective



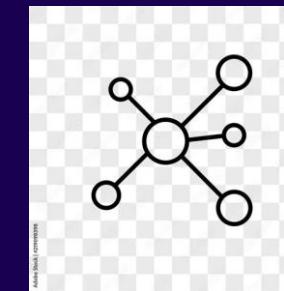
Common Cloud Services

A cloud service provider **can have hundreds of cloud services** that are grouped into various types of services. The four most common types of cloud services (*the 4 core*) for Infrastructure as a Service (IaaS) would be:



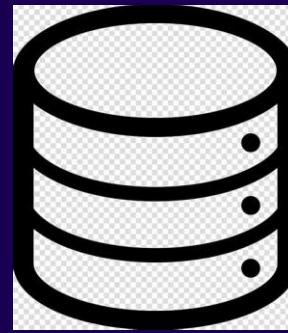
Compute

Imagine having a virtual computer that can run application, programs and code.



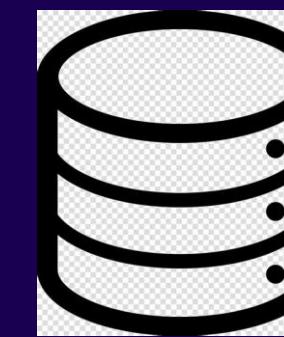
Networking

Imagine having virtual network defining internet connections or network isolations between services or outbound to the internet



Storage

Imagine having a virtual hard-drive that can store files



Databases

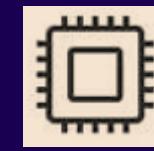
Imagine a virtual database for storing reporting data or a database for general purpose web-application

AWS has over 200+ cloud services

The term "Cloud Computing" can be used to refer to all categories, even though it has "compute" in the name.

Technology Overview

Cloud Service Provider (CSPs) that are **Infrastructure as a Service (IaaS)** will always have **4 core cloud service** offerings:



Compute



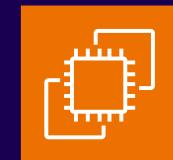
Storage



Database



Networking



Amazon Elastic
Compute Cloud
(Amazon EC2)



Amazon Elastic
Block Store
(Amazon EBS)



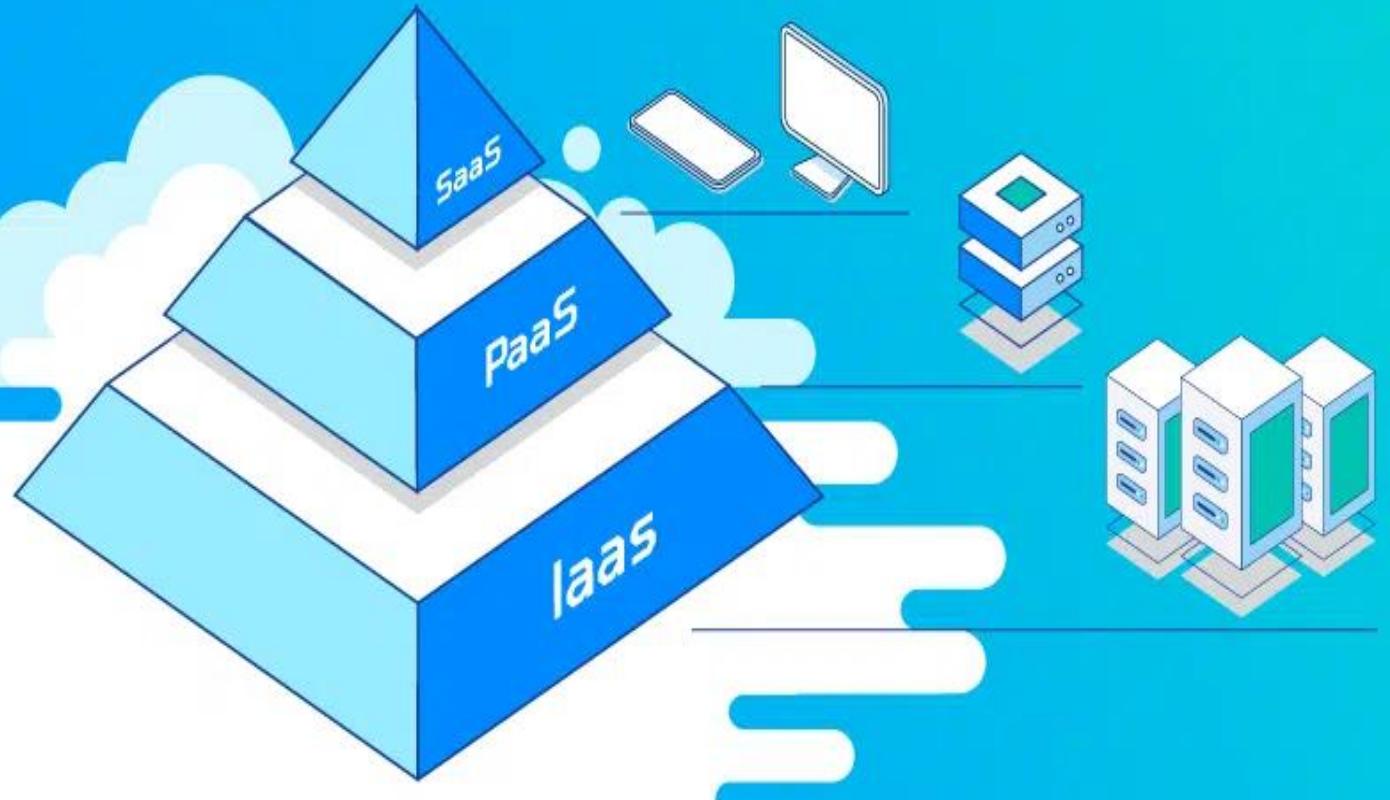
Amazon Relational
Database Service
(Amazon RDS)



Amazon Virtual Private Cloud
(Amazon VPC)

Types of Cloud Computing

IaaS, PaaS, and SaaS



SaaS Software as a Service For Customers

A product that is run and managed by the service provider

Don't worry about how the service is maintained.

It just works and remains available.

PaaS Platform as a Service For Developers

Focus on the deployment and management of your apps.

Don't worry about, provisioning, configuring or understanding the hardware or OS.

IaaS Infrastructure as a Service For Admins

The basic building blocks for cloud IT. Provides access to networking features, computers and data storage space.

Don't worry about IT staff, data centers and hardware

SAAS vs PAAS vs IAAS vs ...AAS



Google App
Engine

Gmail

VS

VS

Are all cloud computing service models, but they differ in what they provide and the level of control you have.

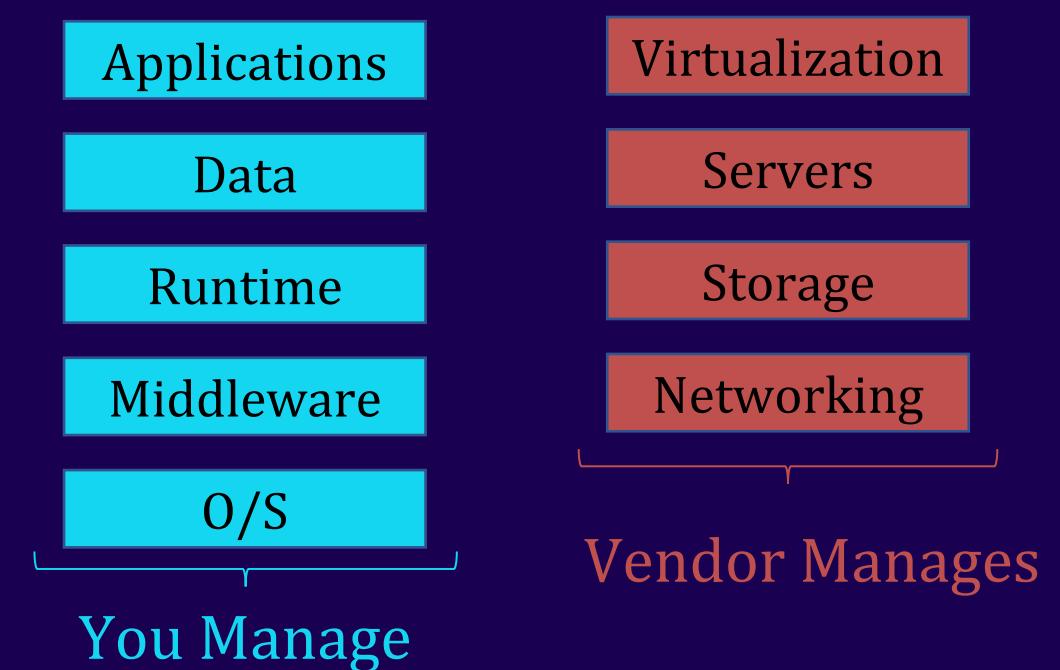
IAAS

- Similar to renting a vacant lot and utilities
e.g. water/electricity
- You are responsible for everything else
- Maximum Control
- Requires Great Technical Expertise



IAAS

- This is where pre-configured hardware is provided via a virtualized interface.
- There is no high-level infrastructure software provided, this must be provided by the buyer embedded with their own virtual applications.
- Some vendors like Oracle will provide operating system also as part of their IaaS service but not all.
- Other Examples: Azure, IBM Cloud, Google Cloud



PAAS

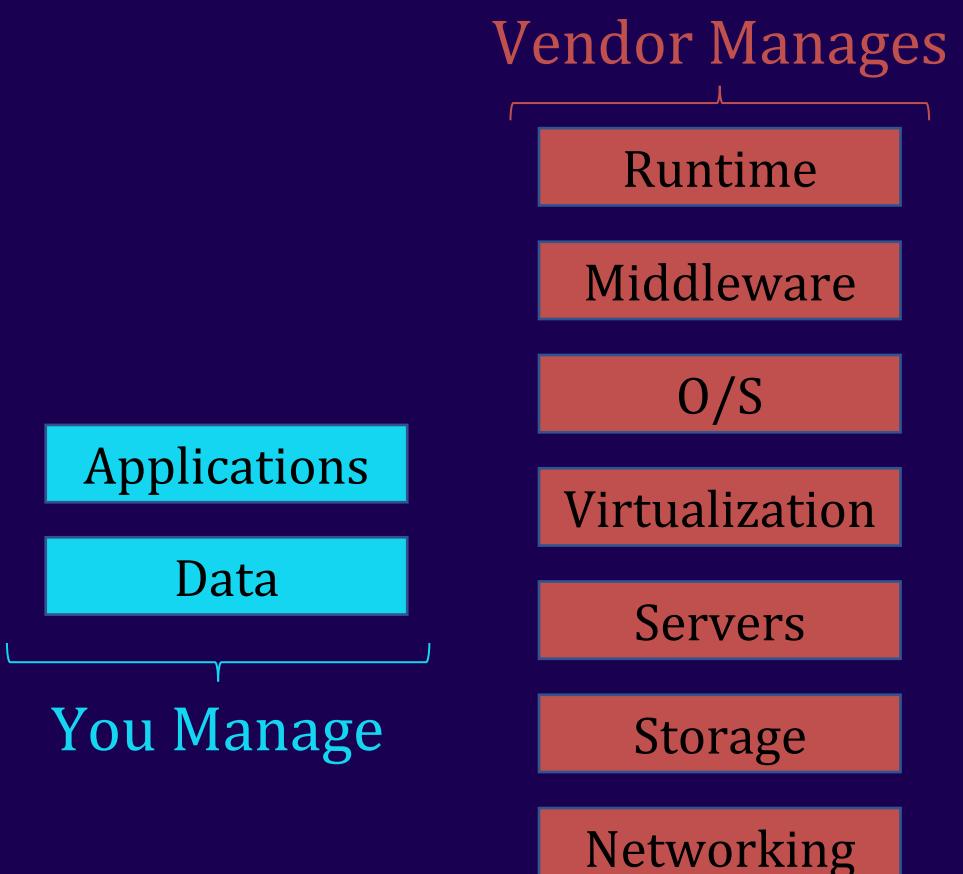
- This is like getting a pre-built shell of a house with plumbing and electrical wiring already installed.
- You can customize the interior - add rooms, paint the walls, lay down flooring.
- PaaS offers a development environment for you to build your applications without worrying about the underlying infrastructure.
- An example of PaaS is Google App Engine



Google App
Engine

PAAS

- Cloud-based platform services that provide developers with a framework they can use to build custom applications upon.
- includes the operating environment included the operating system and software tools (database/middleware/development tools)



SAAS

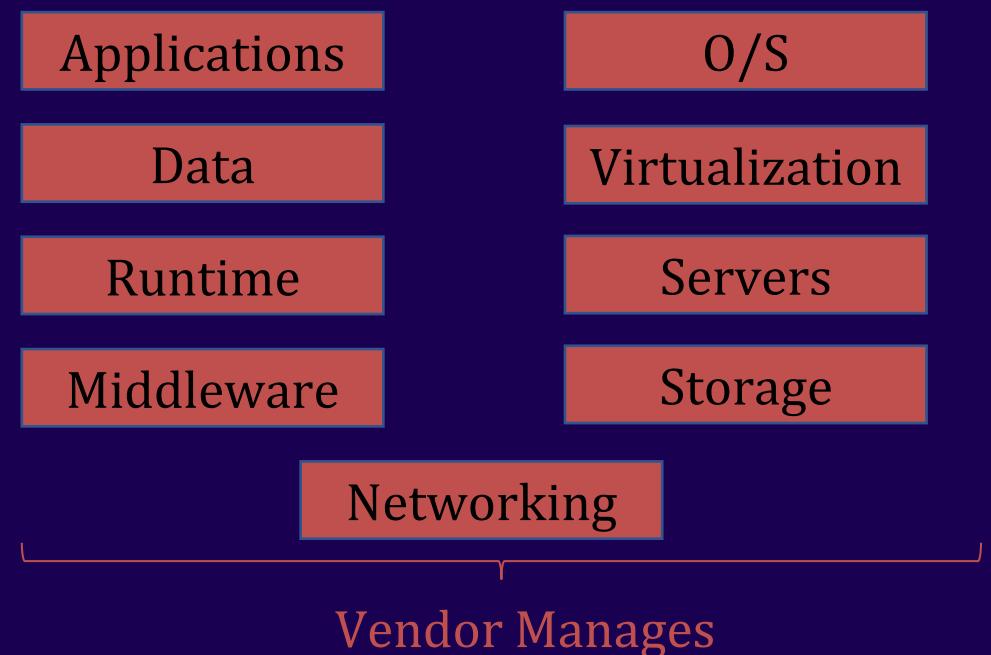
- Alike renting a fully furnished apartment.
- You can move in right away and use everything that's already there.
- SaaS offers ready-made software applications that you can access over the internet.
- Examples of SaaS include Gmail, Dropbox, JIRA, HubSpot and Salesforce



Gmail

SAAS

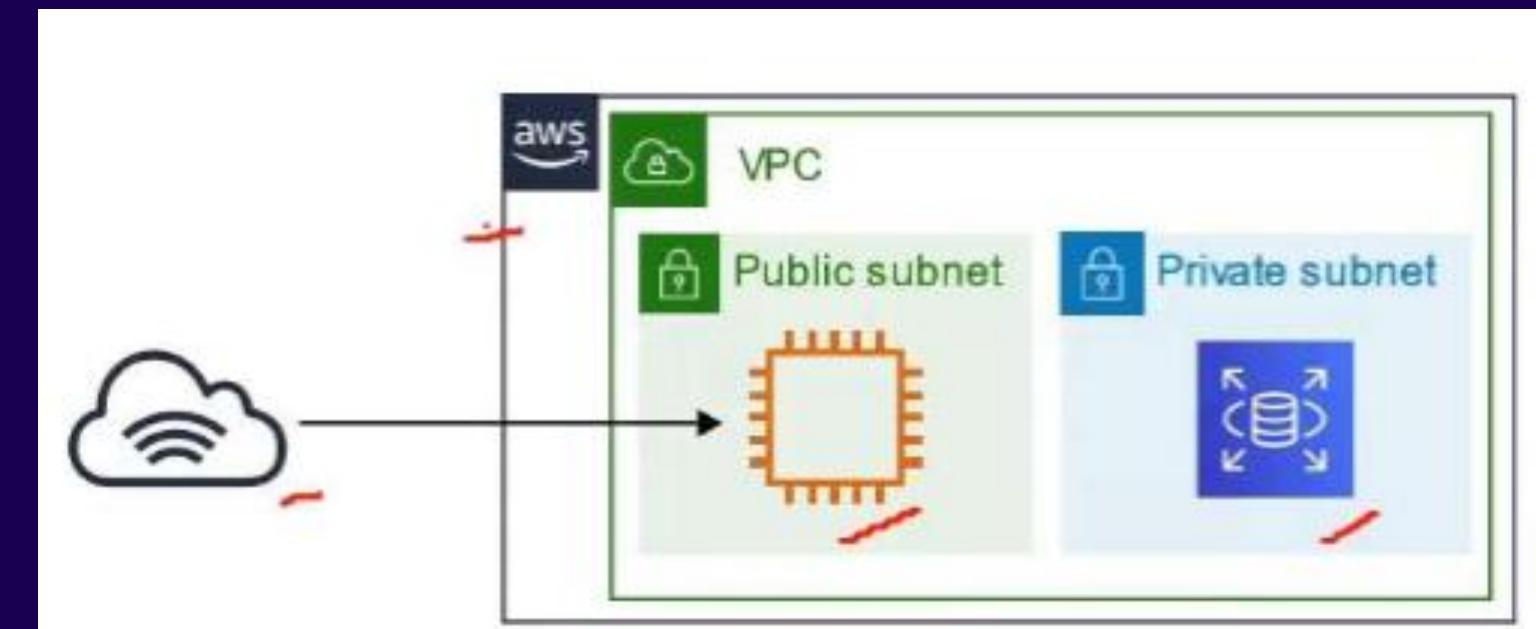
- software that is hosted online by a company and is available for purchase on a subscription basis and is delivered via the internet.
- Offers fully functional applications on-demand to provide specific services such as CRM, ERP, email management, web conferencing and an increasingly wide range of other applications.



Cloud Computing Deployment Models

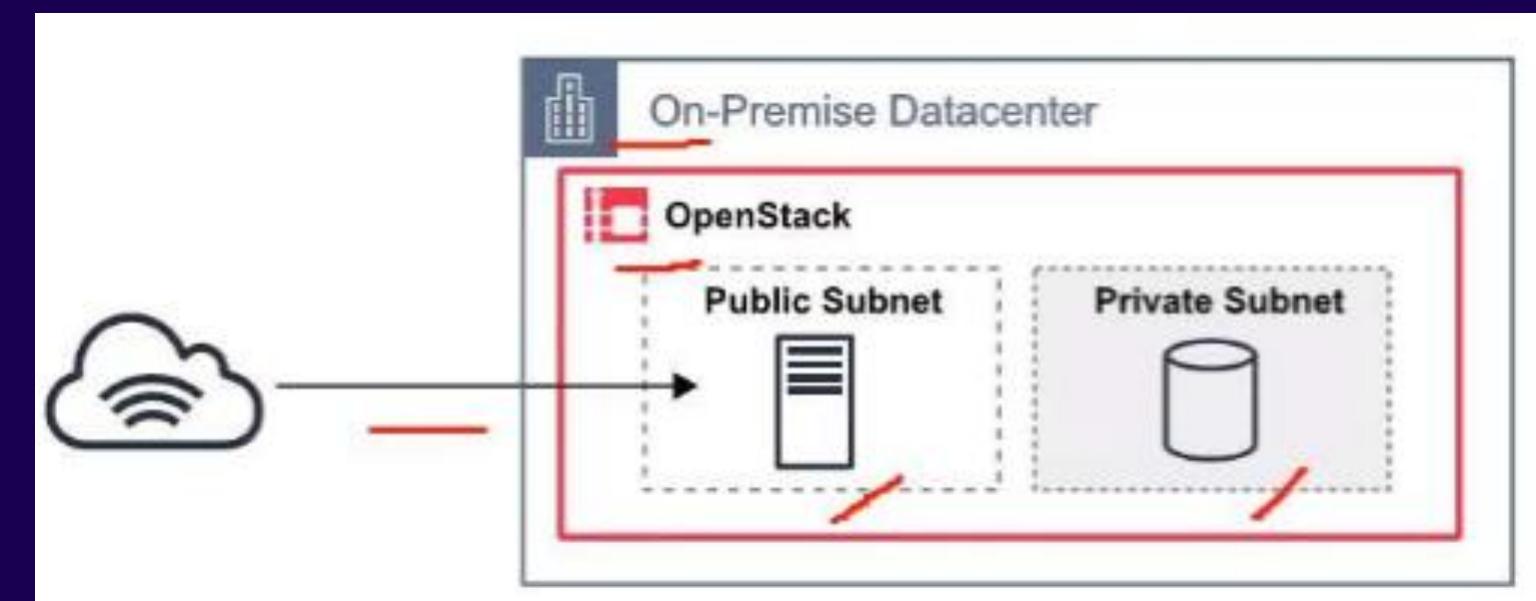
Public Cloud

Everything (the workload or project) is built on the CSP
Also known as: *Cloud-Native or Cloud First



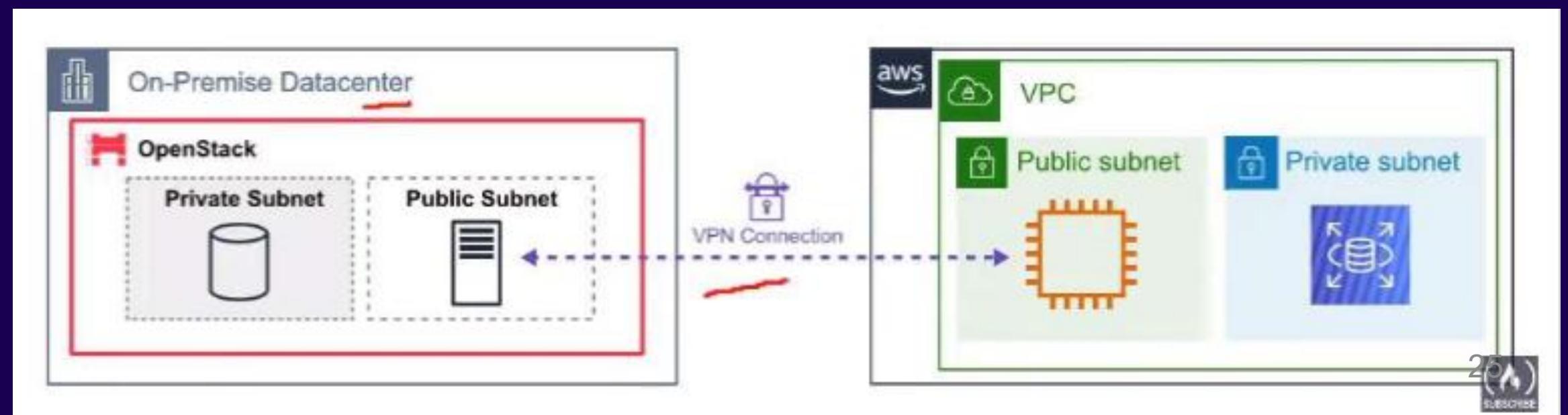
Private Cloud

Everything built on company's datacenters
Also known as **On-Premise** The cloud could
be **OpenStack**



Hybrid

Using both **On-Premise** and
A **Cloud Service Provider**

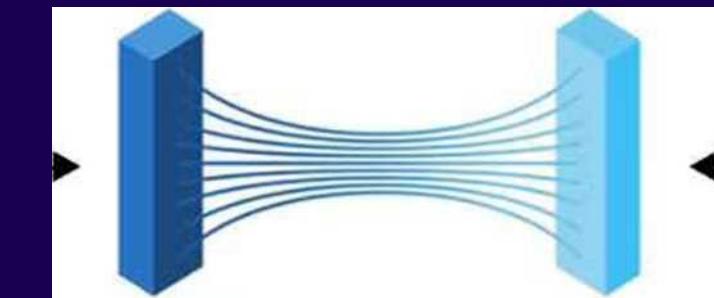


Cloud Computing Deployment Models

Cross-Cloud
Using **Multiple Cloud Providers**
Aka multi-cloud, "-hybrid-cloud"



Amazon EKS



Azure Arc



GCP Kubernetes Engine



Anthos is GCP's offering for a control plane for compute across multiple CSPs and On-premise environments

Cloud Computing Deployment Models

Cloud

Fully utilizing cloud computing



Companies that are starting out today, or are small enough to make the leap from a VPS to a CSP.

Startups SaaS offerings
New projects and companies

Hybrid

Using both Cloud and On-Premise

Deloitte.



Organizations that started with their own datacenter, can't fully move to cloud due to effort of migration or security compliance

Banks
FinTech, Investment Management
Large Professional Service providers
Legacy on-premise

On-Premise

Deploying resources on-premises, using virtualization and resource management tools, is sometimes called "private cloud".



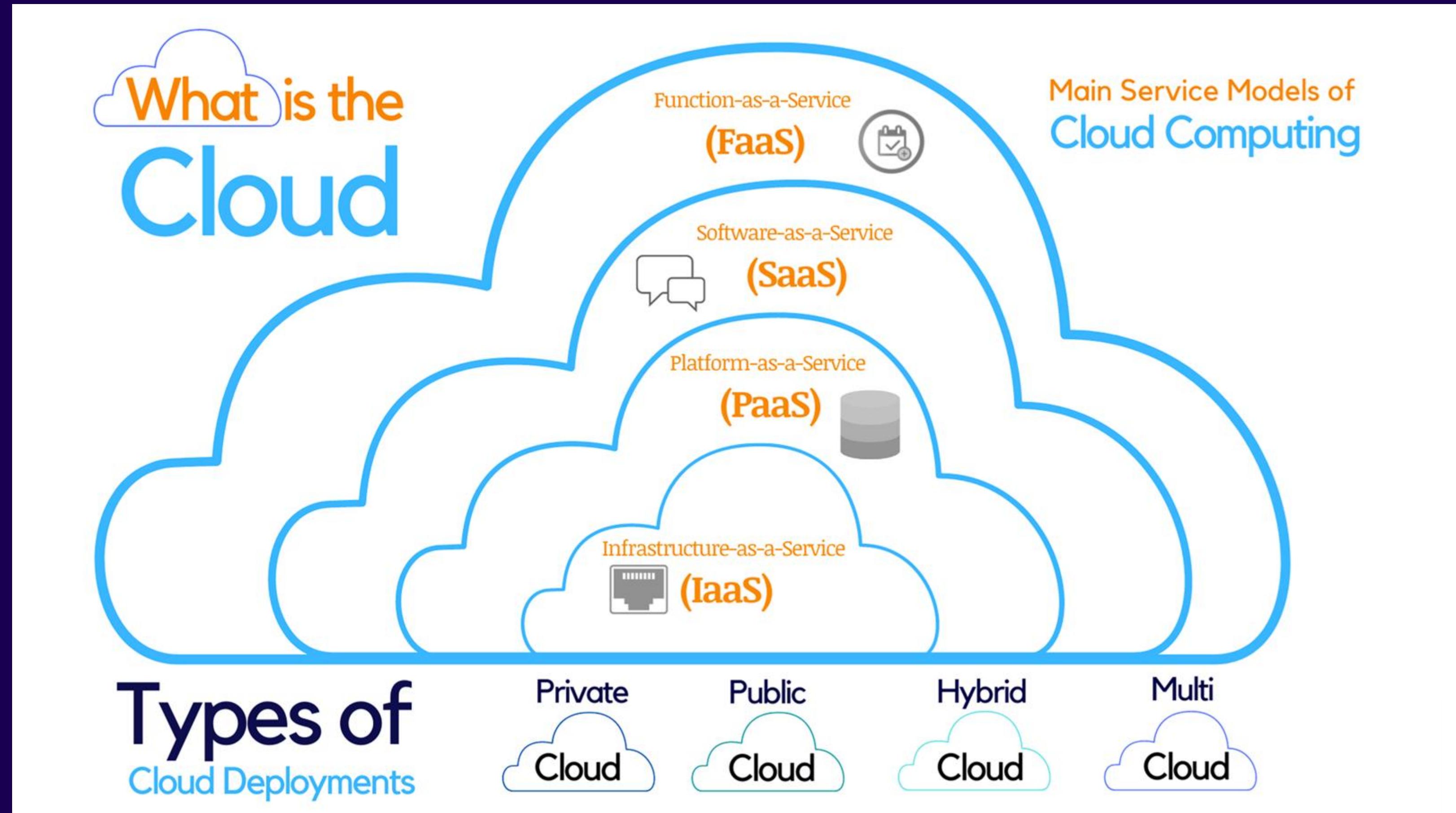
Canada

Organizations that cannot run on cloud due to strict regulatory compliance or the sheer size of their organization

- Public Sector eg. Government
- Super Sensitive Data eg. Hospitals
- Large Enterprise with heavy regulation eg. insurance Companies

There really isn't reason to **be fully on-premise**

Cloud Computing Deployment Models & Types



Cloud Computing Deployment Models & Types

#GCPSketchnote



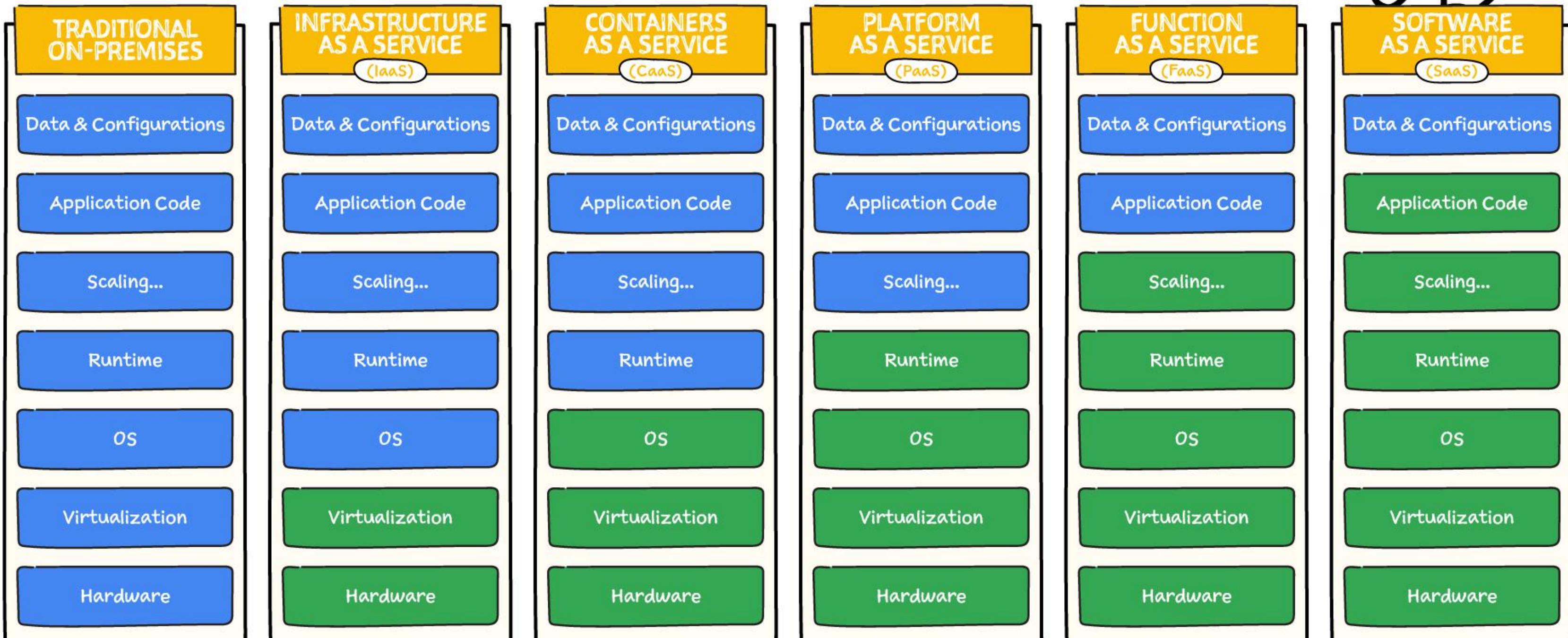
@PVERGADIA

THECLOUDGIRL.DEV

08.II.2021



Wait... what is **Cloud** again?



You Manage



Cloud Provider Manages

Cloud Computing Deployment Models & Types

You want to	Use
Provision HW e.g. servers	IAAS
Want to Run an app without servers	FAAS
Want to build and deploy applications	PAAS
Want to re-use Existing SW from others	SAAS
Want to sell your SW so that others can use	SAAS

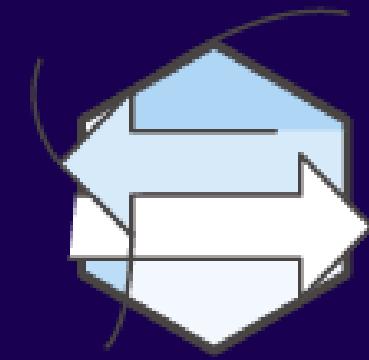
Some use cases for FAAS

- Web Application Backends – User Authentication, Generating Thumbnails, User Registrations, ...
- Real Time Data Processing – Fraud Detection, Real Time Analytics, Alert Triggering
- Async Processing and Workflows
- Event Driven Automation – Backup and Recovery
- Chatbots and Voice Assistants

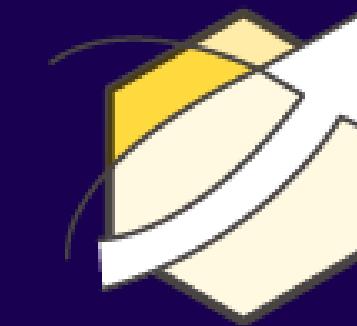
Cloud Computing Deployment Models & Types

Type	Common Examples
SaaS	Google Workspace, Dropbox, Salesforce, Cisco WebEx, Concur, GoToMeeting
PaaS	Amazon Web Services (AWS) Elastic Beanstalk, Windows Azure, Heroku, Force.com, Google App Engine, Apache Stratos, Red Hat OpenShift
IaaS	DigitalOcean, Linode, Rackspace, AWS, Cisco Metapod, Microsoft Azure, Google Compute Engine (GCE)
DBaaS	AWS RDS, AWS SQL Server, AWS DynamoDB, Astra DB, Google BigTable, MongoDB Atlas, Azure Cosmos DB, Microsoft Azure SQL Database, Amazon Aurora,
CaaS	Amazon ECS, Google Container Engine, IBM Kubernetes Service, Oracle Container Services, Azure Container Services, Docker Enterprise
FaaS	AWS Lambda, Azure Functions, Google Cloud Functions

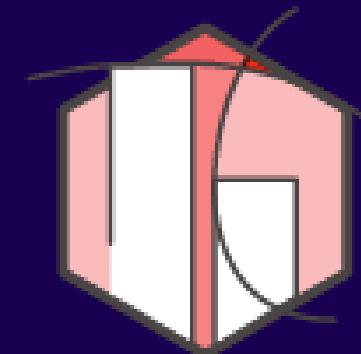
Six advantages of cloud computing



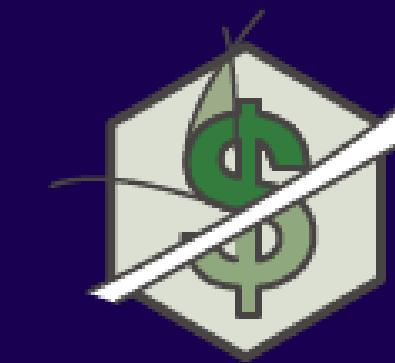
Trade upfront
expense for
variable expense



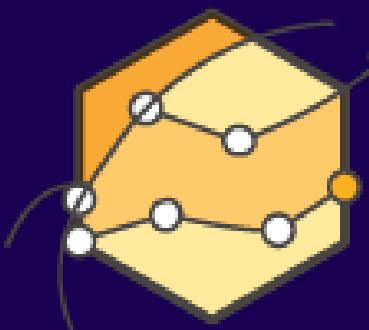
Increase speed
and agility



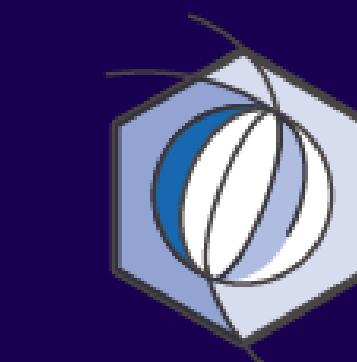
Benefit from massive
economies of scale



Stop spending
money on running and
maintaining data centers



Stop guessing
capacity



Go global
in minutes

Seven Advantages to Cloud

Cost-effective

You **pay for what you consume**, no up-front cost. On-demand pricing or Pay-as-you-go (PAYG) with thousands of customers sharing the cost of the resources

Global

Launch workloads **anywhere in the world**, Just choose a region

Secure

Cloud provider takes care of physical security. **Cloud services can be secure by default** or you have the ability to configure access down to a granular level.

Reliable

Data backup, disaster recovery, data replication, and fault tolerance

Scalable

Increase or decrease resources and services based on demand

Elastic

Automate scaling during spikes and drop in demand

Current

The underlying hardware and managed software is patched, upgraded and replaced by the cloud provider without interruption to you.

Cloud Terminologies and Role

Solution Architect

A role in a technical organization that architects a technical solution using multiple systems via researching, documentation, experimentation.

A Solutions Architect needs to always consider the following business factors:

- (Security) How secure is this solution?
- (Cost) How much is this going to cost?



Cloud Terminologies and Role

Cloud Architect

A solutions architect that is focused solely on architecting technical solutions using cloud services.

A cloud architect need to understand the following terms and factor them into their designed architecture based on the business requirements.

- **Availability** - Your ability to ensure a service remains available eg. **Highly Available (HA)**
- **Scalability**-Your ability to grow rapidly or unimpeded [
- **Elasticity**-Your ability to shrink and grow to meet the demand
- **Fault Tolerance** - Your ability to prevent a failure
- **Disaster Recovery** - Your ability to recover from a failure eg. **Highly Durable (DR)**

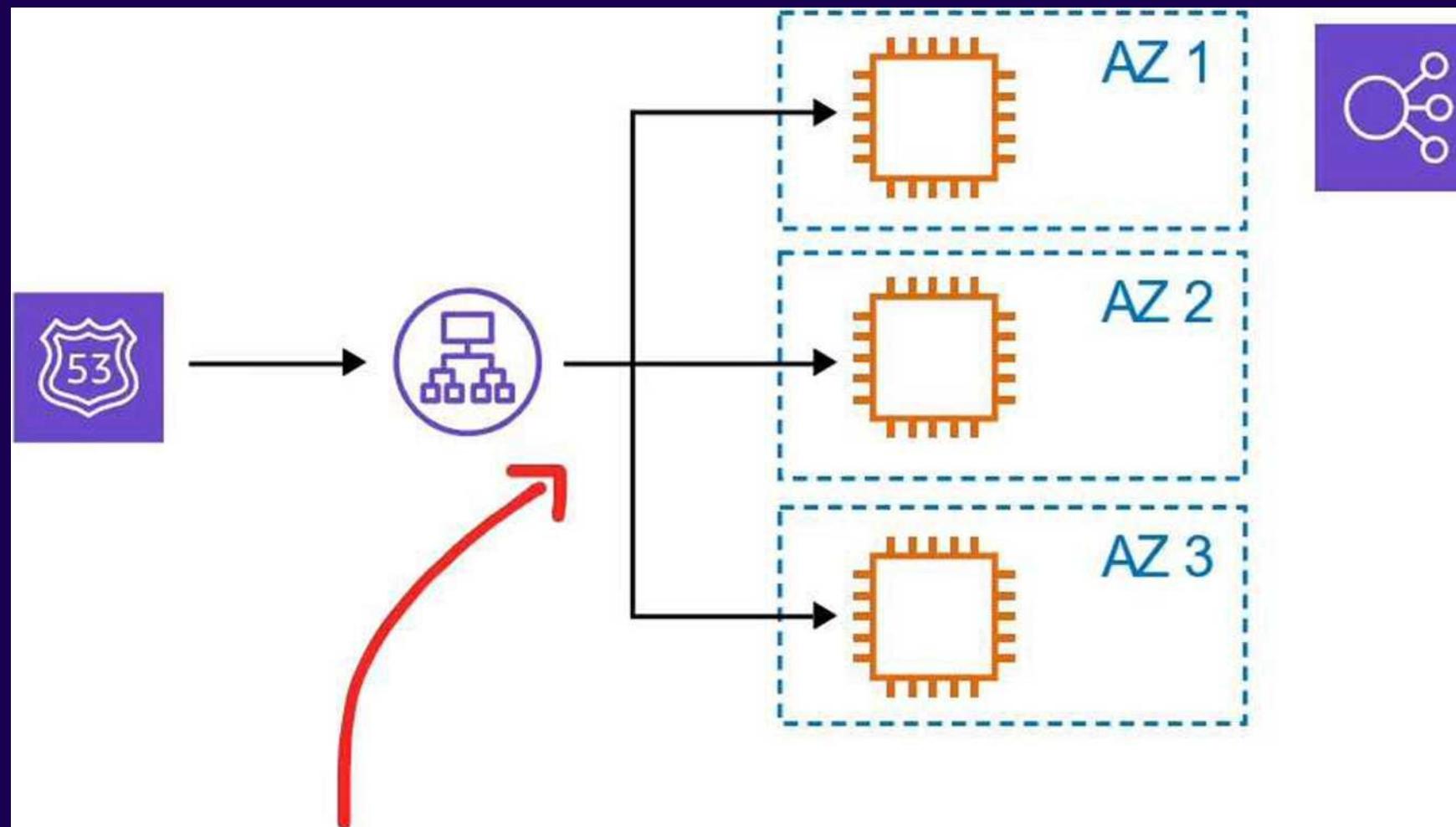
Cloud Terminologies

High Availability

Ability of a service to **remain available** by ensuring there is ***no single point of failure*** and/or ensure a certain level of performance

Cloud Terminologies

High Availability



Elastic Load Balancer

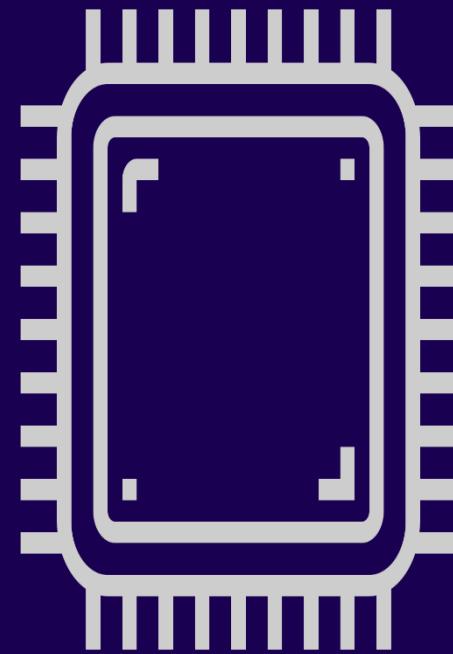
A load balancer allows you to evenly distribute traffic to multiple servers in one or more datacenter. If a datacenter or server becomes unavailable (unhealthy) the load balancer will route the traffic to only available datacenters with servers.

Running your workload across multiple **Availability Zones** ensures that if 1 or 2 **AZs** become unavailable your service / applications remains available.

Cloud Terminologies

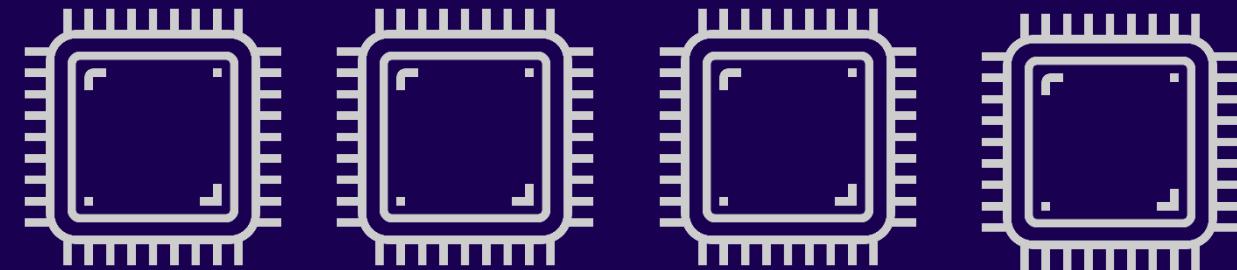
High Scalability

Ability to **increase capacity** based on the increasing demand of traffic, memory and computing power



Vertical Scaling

Scaling Up
Upgrade to a bigger server



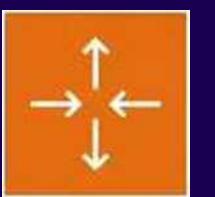
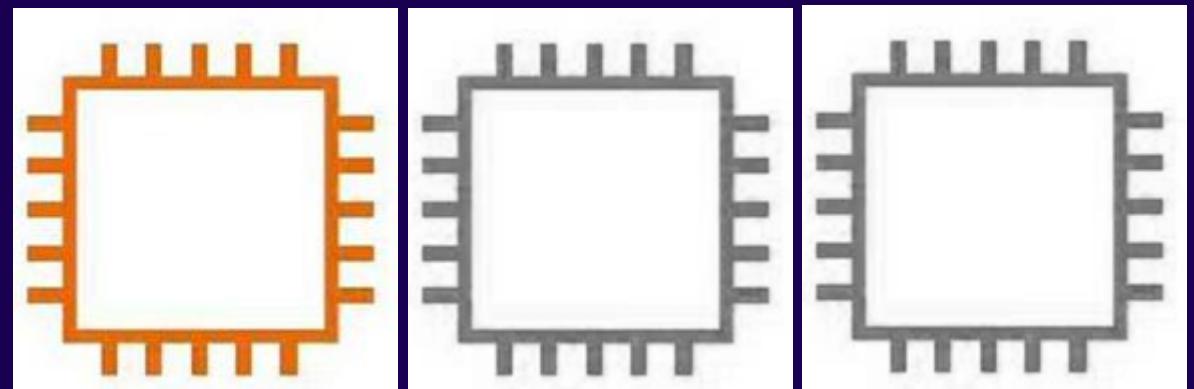
Horizontal Scaling

Scaling Out/In
Add more servers of the same size

Cloud Terminologies

High Elasticity

Ability to **automatically increase or decrease** your capacity based on the current demand of traffic, memory and computing power



Auto Scaling Groups (ASG) is an AWS feature that will automatically add or remove servers based on scaling rules you define based on metrics

Cloud Terminologies

High Elasticity

Horizontal Scaling

Scaling Out — Add more servers of the same size

Scaling In — Removing underutilized servers of the same size

Vertical Scaling is generally hard for traditional architecture so you'll usually only see horizontal scaling described with Elasticity.

Cloud Terminologies

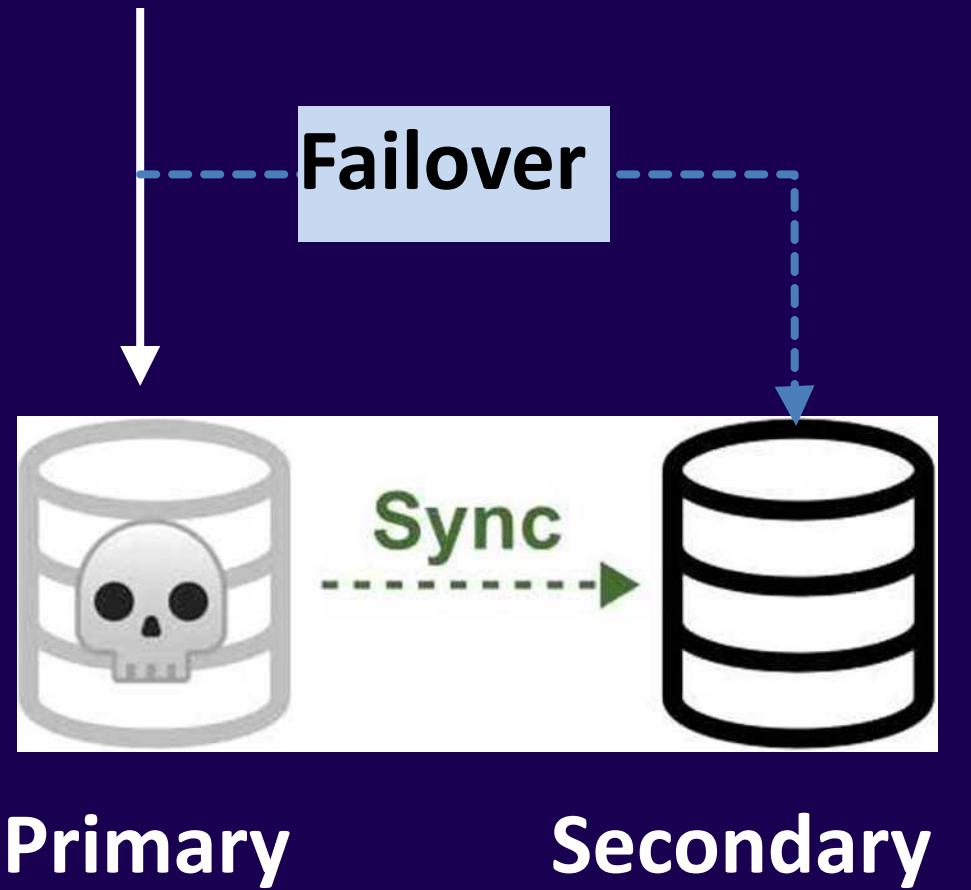
Highly Fault Tolerant

Ability of a service to ensure there is no **single point of failure,(SPOF)**
preventing or minimizing the chance of failure

Cloud Terminologies

Highly Fault Tolerant

Fail-overs is when you have a plan to **shift traffic** to a redundant system in case the primary system fails



RDS Multi-AZ is when you run a duplicate standby database in another Availability Zone in case your primary database fails.

A common example is having a copy (secondary) of your database where all ongoing changes are synced. The secondary system is not in-use until a fail over occurs and it becomes the primary database.

Cloud Terminologies

High Durability

Ability to **recover** from a disaster and to prevent **the loss** of data.

Solutions that recover from a disaster is known as **Disaster Recovery (DR)**

- Do you have a backup?
- How fast can you restore that backup?
- Does your backup still work?
- How do you ensure current live data is not corrupt?

Summary

- High Availability
- High Scalability
- High Elasticity
- High Durability and Fault Tolerance

Others

- Cloud First vs Cloud Native
- The formal definition of cloud Computing from Cloud Native Computing Foundation

Cloud native practices empower organizations to develop, build, and deploy workloads in computing environments (public, private, hybrid cloud) to meet their organizational needs at scale in a programmatic and repeatable manner. It is characterized by **loosely coupled systems** that interoperate in a manner that is secure, resilient, manageable, sustainable, and observable.

Cloud native technologies and architectures typically consist of **some combination of containers, service meshes, multi-tenancy, microservices, immutable infrastructure, serverless, and declarative APIs** — this list is non-exhaustive.

Others

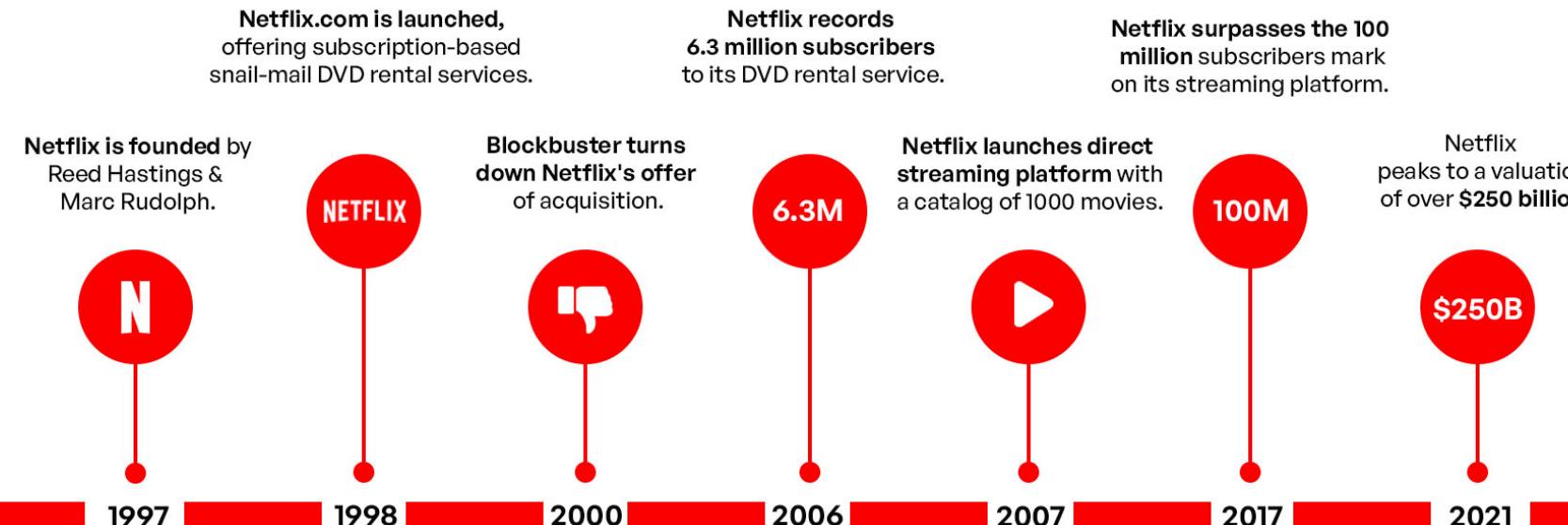
Parameters	Cloud-native	Cloud-based	Cloud-enabled	SaaS
Definition	Applications designed for the cloud, leveraging cloud-native technologies and practices.	Applications hosted on cloud infrastructure, but may not be specifically designed for the cloud.	Traditional applications hosted in the cloud, with some cloud integrations and capabilities.	Software applications delivered over the internet as a service, typically on a subscription basis.
Infrastructure Ownership	Self-managed or third-party cloud provider.	Third-party cloud provider.	Third-party cloud provider.	Third-party cloud provider.
Application Development	Built specifically for cloud platforms using cloud-native architectures (e.g., microservices, containers).	Can be traditional applications adapted to run on cloud infrastructure.	Traditional applications, with limited cloud-specific optimizations.	Pre-built software applications delivered as a service.

Others

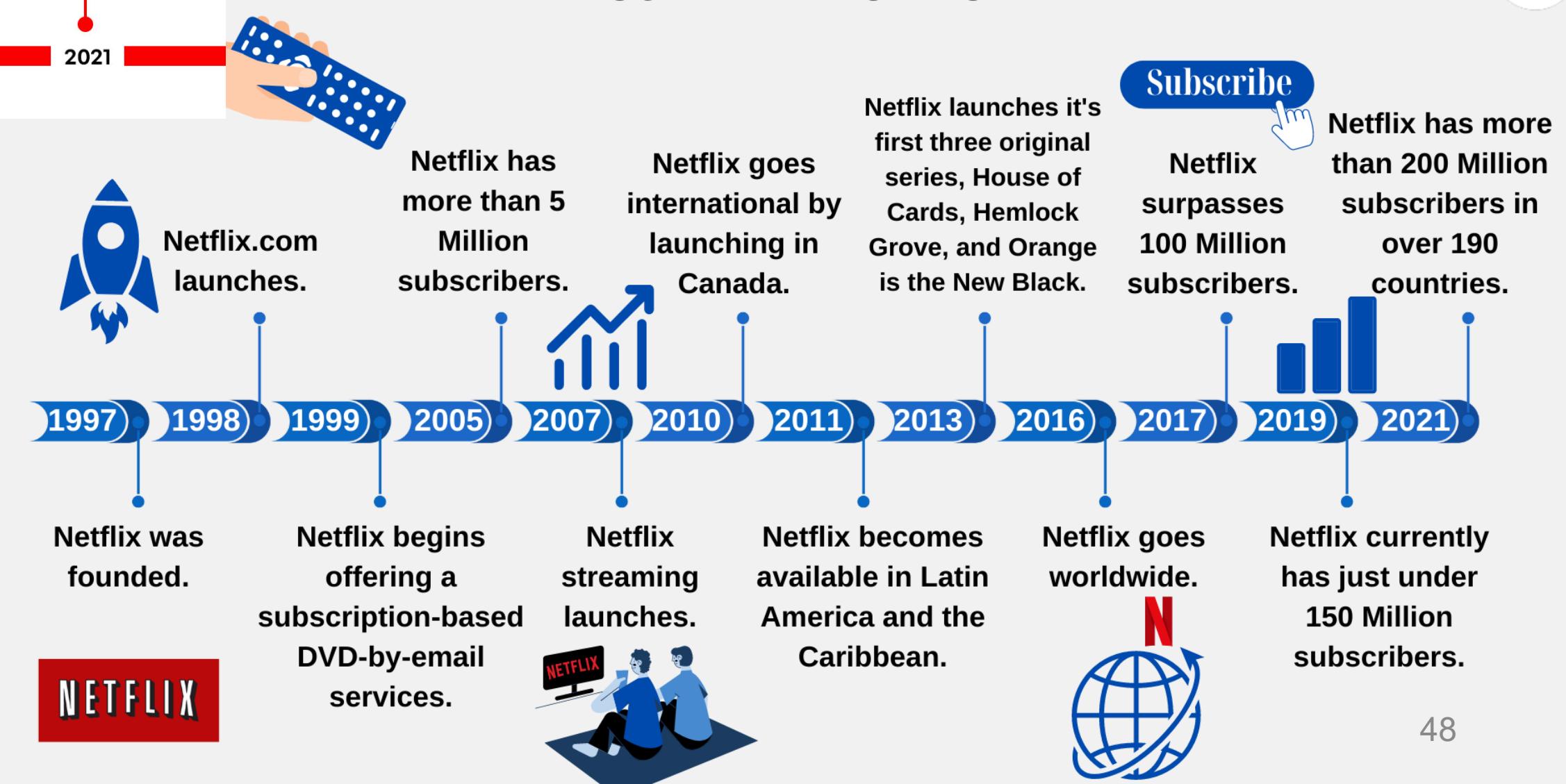
Parameters	Cloud-native	Cloud-based	Cloud-enabled	SaaS
Scalability	Highly scalable, designed for dynamic scaling of components and services.	Scalable based on cloud infrastructure capabilities and configurations.	Scalable based on cloud infrastructure capabilities and configurations.	Scalable based on the SaaS provider's infrastructure and configurations.
Customization	High level of customization and flexibility, allowing tailored solutions.	Customization possible, but limited to the provided infrastructure and configurations.	Customization possible, but limited to the provided infrastructure and configurations.	Limited customization options, mainly configuration-based.
Deployment Model	Can be deployed in various models (public, private, hybrid, multi-cloud).	Can be deployed in various models (public, private, hybrid, multi-cloud).	Can be deployed in various models (public, private, hybrid, multi-cloud).	Typically deployed as a public cloud service.
Examples	Netflix, Airbnb, Spotify	Dropbox, CRM systems, e-commerce platforms	On-premises applications integrated with cloud services	Salesforce, Microsoft Office 365, Google Workspace

Netflix and AWS

The Historic Timeline of Netflix



Netflix Timeline



Netflix and AWS

- In 2008, Big crash of Netflix - No DVDs could be shipped.
- Netflix realized that the model of vertical scaling in Data Centres cannot be relied on and need to move to horizontal scaling
- Started moving to AWS in 2008 and most of customer facing apps moved to cloud by 2015.
- Moving Billing and Employee/Customer Management to secure cloud by 2016.
- By 2023, DVD service shut down and all fully digitally transformed.

Netflix and AWS

- 8 times as many customers as from 2008.
- Scale up massively == 1000s of Virtual Servers and petabytes of data in minutes
- Speed up delivery in localized regions - by 2016 over 130 countries
- Highly available
- Build highly reliable services with fundamentally unreliable but redundant components
- Graceful degradation of services
- Cost Reduction
- Adding more value added services

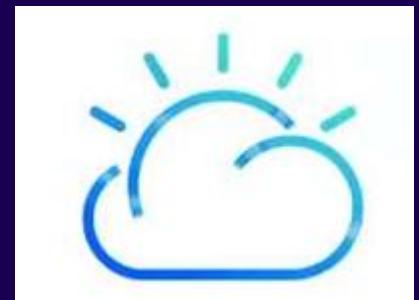
Part 2 - Introduction to AWS

Landscape of CSPs

Tier-1 (Top Tier) - Early to market, wide offering, strong synergies between services, well recognized in the industry



Tier-2 (Mid Tier) - Backed by well-known tech companies, slow to innovate and turned to specialization



IBM Cloud

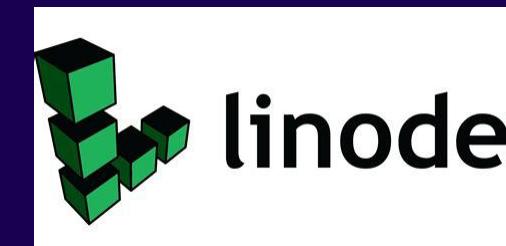


Oracle Cloud



Rackspace (OpenStack)

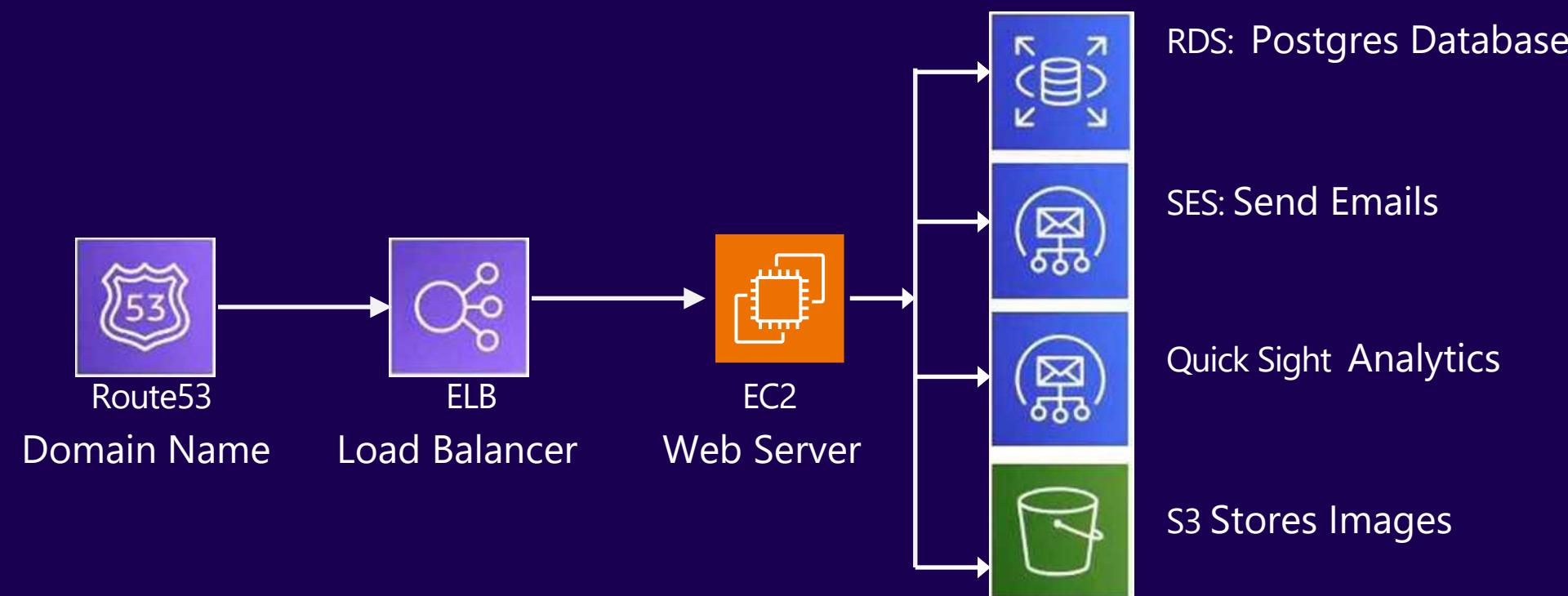
Tier-3 (Light Tier) - Virtual Private Servers (VPS) turned to offer core IaaS offering. Simple, cost-effective



What is a Cloud Service Provider (CSP)?

A **Cloud Service Provider (CSP)** is a company which

- provides multiple Cloud Services e.g. tens to hundreds of services
- those Cloud Services **can be chained together** to create cloud architectures
- those Cloud Services are accessible **via Single Unified API** eg. AWS API
- those Cloud Services utilized **metered billing** based on usage e.g. per second, per hour
- those Cloud Services have rich monitoring built in eg. AWS CloudTrail
- those Cloud Services have an Infrastructure as a Service (IaaS) offering
- Those Cloud Services offers **automation** via Infrastructure as Code (IaC)



If a company offers multiple cloud services under a single UI but do not meet most of or all of these requirements, it would be referred to as a Cloud Platform e.g. Twilio, HashiCorp, Databricks

Welcome to AWS



An American multinational computer technology corporation headquartered in **Seattle, Washington**



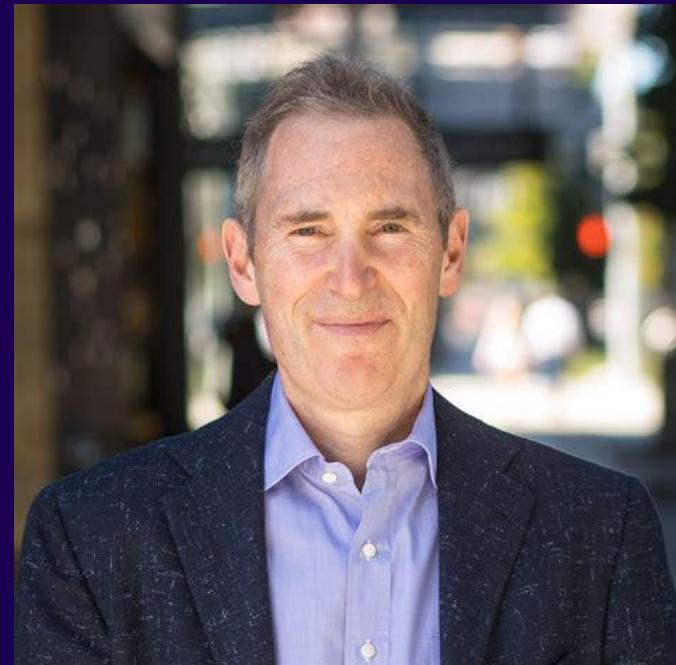
Amazon was founded in 1994 by **Jeff Bezos** and the company started as an online store for books and expanded to other products.

@timothyeberry on Unsplash

Amazon & AWS



Jeff Bezos



Andy Jassy is the current CEO of Amazon.
Previously CEO of AWS.

Amazon has expanded beyond just an online e-commerce store into:

- **cloud computing** (Amazon Web Services)
- digital streaming
- Amazon Prime Video
- Amazon Prime Music
- Twitch.tv
- Grocery Stores (Whole Foods Market)
- artificial intelligence
- Low orbit satellites (Kuiper Systems)
- And more!

People to know



Adam Selipsky

Former CEO of AWS

Former CTO of Tableau, spent a decade with AWS as VP of Marketing, Sales and Support



Matt Garman

CEO of AWS



Jeff Barr

Chief Evangelist

Werner Vogels

CTO of AWS

"**Everything fails, all the time/**



Amazon Web Services

Amazon calls their cloud provider service

Amazon Web Services
Commonly referred to just **AWS**



Old Logo



New Logo

AWS was launched in ***2006** is the **leading cloud service provider** in the world.

Cloud Service Providers can be initialized as **CSPs**

What is Amazon Web Services (AWS)?



Simple Queue Service (SQS) was the first AWS service launched for public use in 2004



Simple Storage Service (S3) was launched in March of 2006



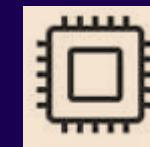
Elastic Compute Cloud (EC2) was launched in August of 2006

In November 2010, it was reported that all of Amazon.com's retail sites had migrated to AWS

To support industry-wide training and skills standardization, AWS began offering a certification program for computer engineers, on April, 2013

Technology Overview

Cloud Service Provider (CSPs) that are **Infrastructure as a Service (IaaS)** will always have **4 core cloud service** offerings:



Compute



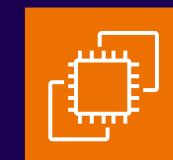
Storage



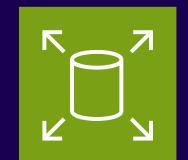
Database



Networking



Amazon Elastic
Compute Cloud
(Amazon EC2)



Amazon Elastic
Block Store
(Amazon EBS)



Amazon Relational
Database Service
(Amazon RDS)



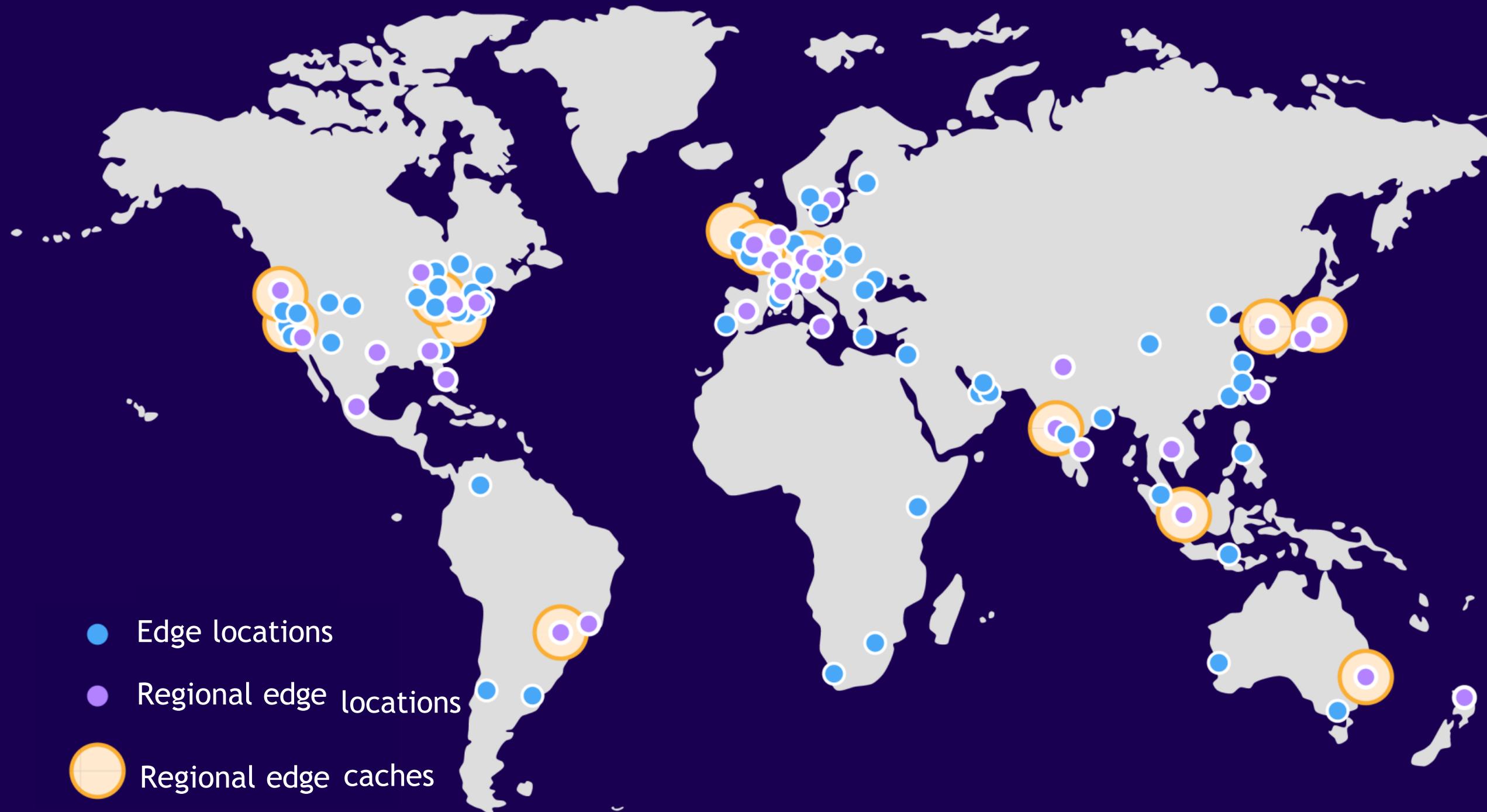
Amazon Virtual Private Cloud
(Amazon VPC)

An overview

AWS provides over 200+ services.

These are backed up by a very powerful global infrastructure

AWS global infrastructure: Current Regions



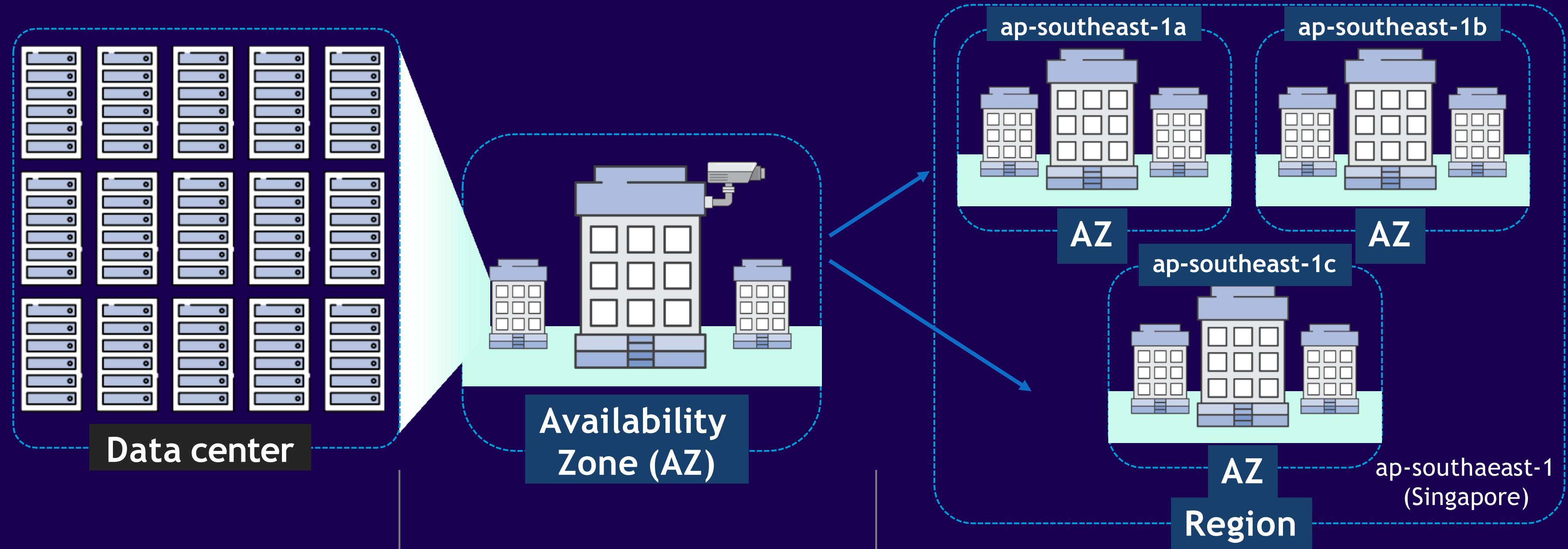
Choose a Region

- Data governance
- Latency
- Cost

Edge infrastructure

- Amazon CloudFront (content delivery network)
- AWS Outposts
- AWS Local Zones
- AWS Wavelength

AWS global infrastructure



Typically houses thousands of servers

- One or more data centers
- Designed for **fault isolation**

- Each AWS Region is made up of **two or more AZs**
- AWS has **36 Regions worldwide and growing**

AWS edge infrastructure

Moving the cloud closer to the endpoint

AWS Outposts



AWS Local Zones



AWS Wavelength



Overview

AWS infrastructure and services **on premises**

Use cases

Migration, local critical applications, data residency

Service model

Expandable capacity in customer's data center, colocation, on-premises location

AWS infrastructure and services **in large metro centers**

Migration, low latency, local data processing

Scalable capacity in facility managed & operated by AWS

AWS infrastructure and services **in Commercial Service Provider (CSP) 5G networks**

Ultra-low latency, local data processing

Scalable capacity in CSP data center managed and supported by AWS

More on this later!⁶³

AWS Outposts



AWS Outposts is a fully managed service that offers the same AWS infrastructure, AWS services, APIs, and tools to virtually any datacenter, co-location space, or on-premises facility for a truly consistent hybrid experience.

AWS Outposts is rack of servers running AWS Infrastructure **on your physical location**

42U Rack

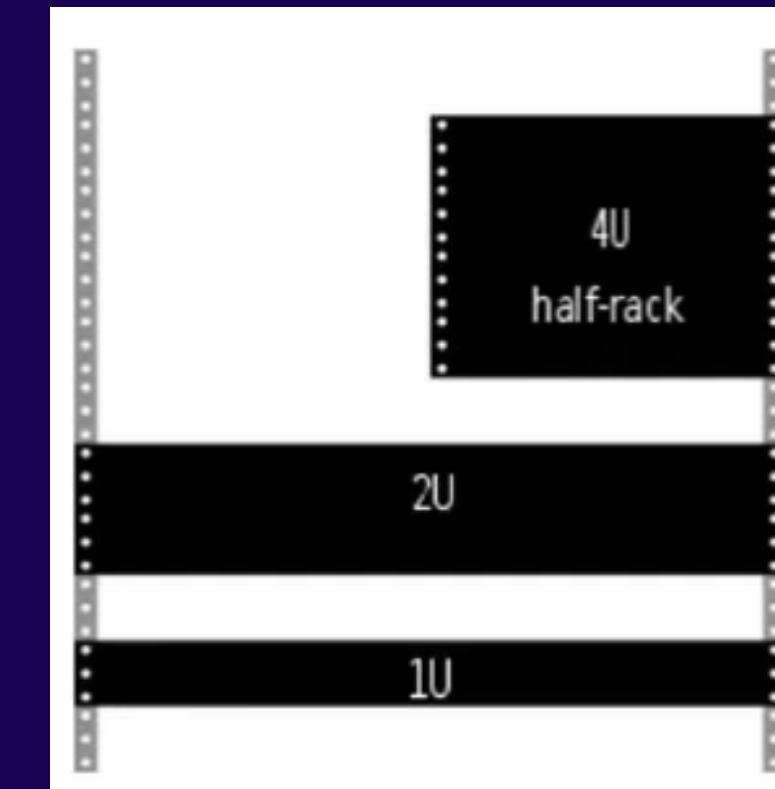


What is a Server Rack? Rack Heights

A frame design to hold and U stands for "rack units" or "U spaces" with is equal to 1.75 inches, organize IT equipment.

The industry standard rack size is 48U (7 Foot Rack)

- full-size rack cage is 42U high
- equipment is typically 1U, 2U, 3U, or 4U high



AWS Outposts

42U



1U

suitable for 19-inch wide 24-inch deep cabinets AWS Graviton2 (up to 64 vCPUs) 128 GiB memory 4 TB of local NVMe storage

2U

suitable for 19-inch wide 36-inch deep cabinets, Intel processor (up to 128 vCPUs) 256 GiB memory 8TB of local NVMe storage

AWS delivers it to your preferred physical site fully assembled and ready to be rolled into final position. It is installed by AWS and the rack needs to be simply plugged into power and network.

AWS Global infrastructure

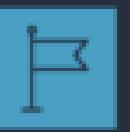
What is the AWS Global Infrastructure?

The AWS Global Infrastructure is **globally distributed hardware and datacenters that are physically networked together** to act as one large resource for the end customer.

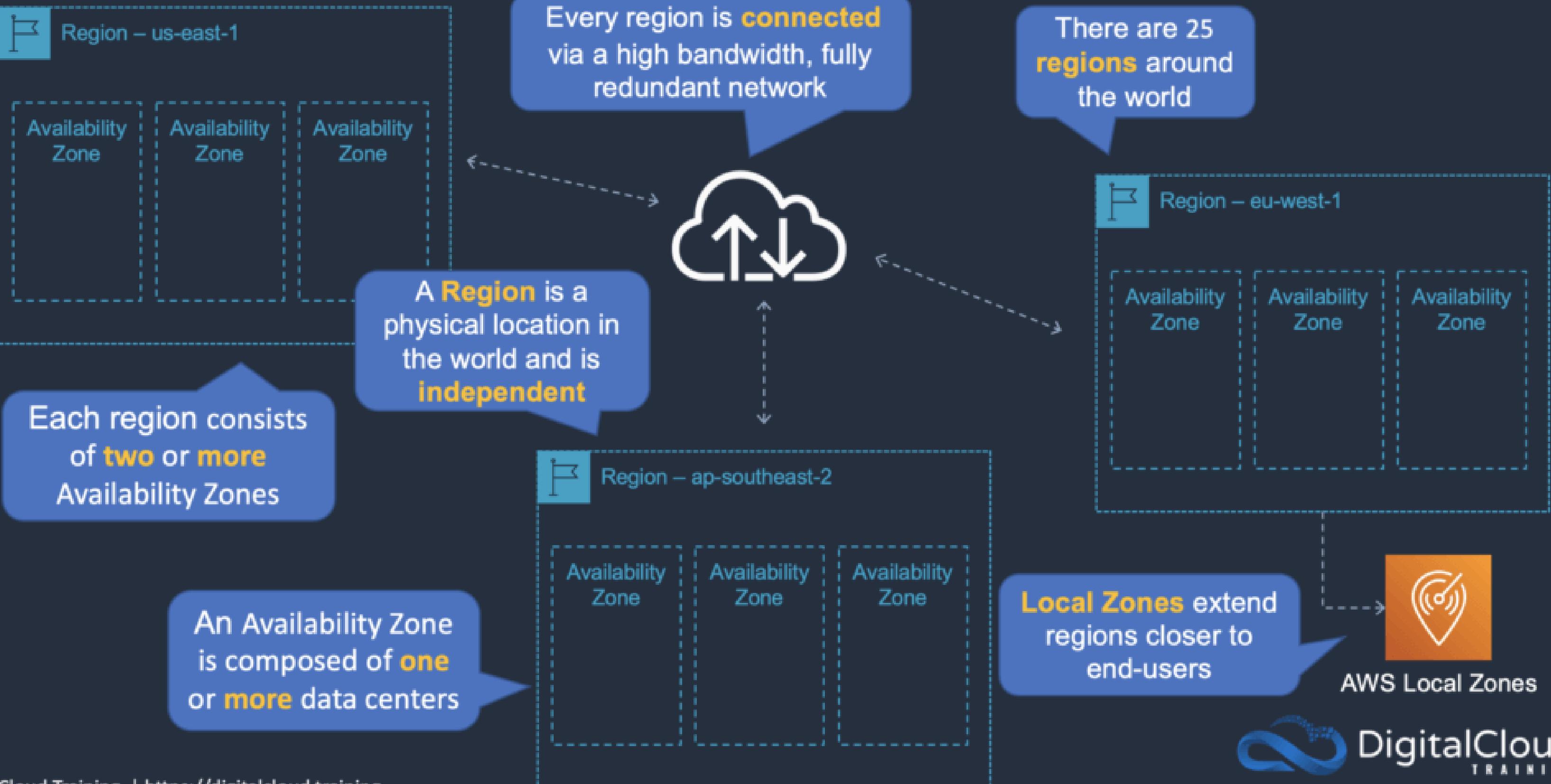
The AWS Global Infrastructure is made up of the following resources:

- 245+ Countries and territories
- 36 Launched Regions
- 114 Availability Zones
- 135 Direct Connection Locations
- 700+ Points of Presence
- 43 Local Zone
- 31 Wavelength Zones

AWS has **millions** of active customers and **tens of thousands** of partners globally



AWS Global Infrastructure



Choosing a Region

Every region has 3 or more Availability zones (Generally)

When AWS introduces a new service, it is usually available first in US-EAST.

All Billing Information appears in US-EAST-1 (North Virginia)

Not all regions have all services enabled.

Cost of services will vary from region to region.

Choosing a Region

When you choose a region there are four factors you need to consider:

1. What Regulatory Compliance does this region meet?
2. What is the cost of AWS services in this region?
3. What AWS services are available in this region?
4. What is the distance or latency to my end-users?

AWS Global Infrastructure

Regional Services

AWS **scopes** their AWS Management Console on a selected Region.

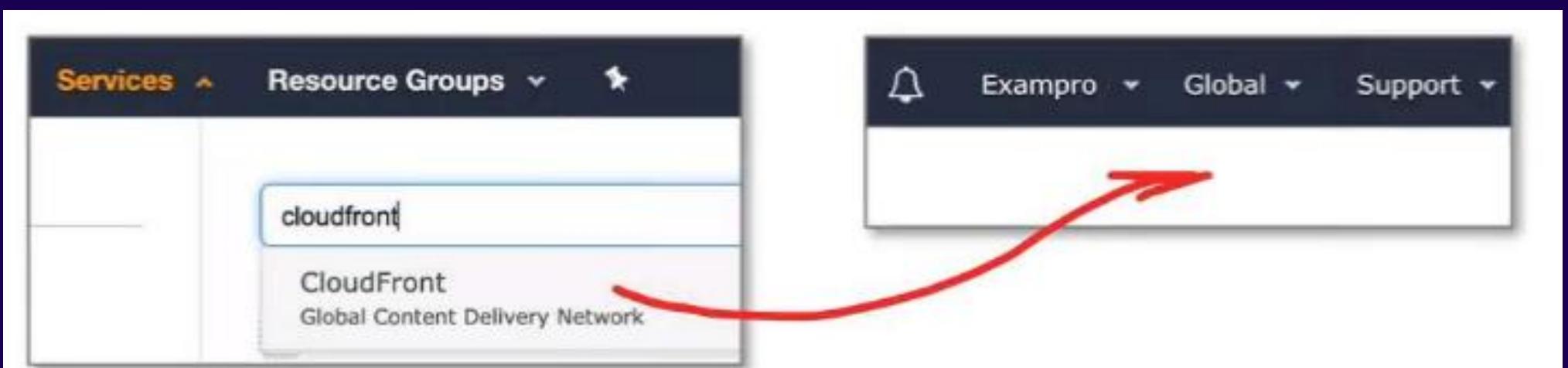
This will determine where an AWS service will be launched and what will be seen within an AWS Service's console.

You generally don't explicitly set the Region for a service at the time of creation

Global Services

Some AWS Services operate across multiple regions and the region will be fixed to "Global"

E.g. Amazon S3, CloudFront, Route53, IAM

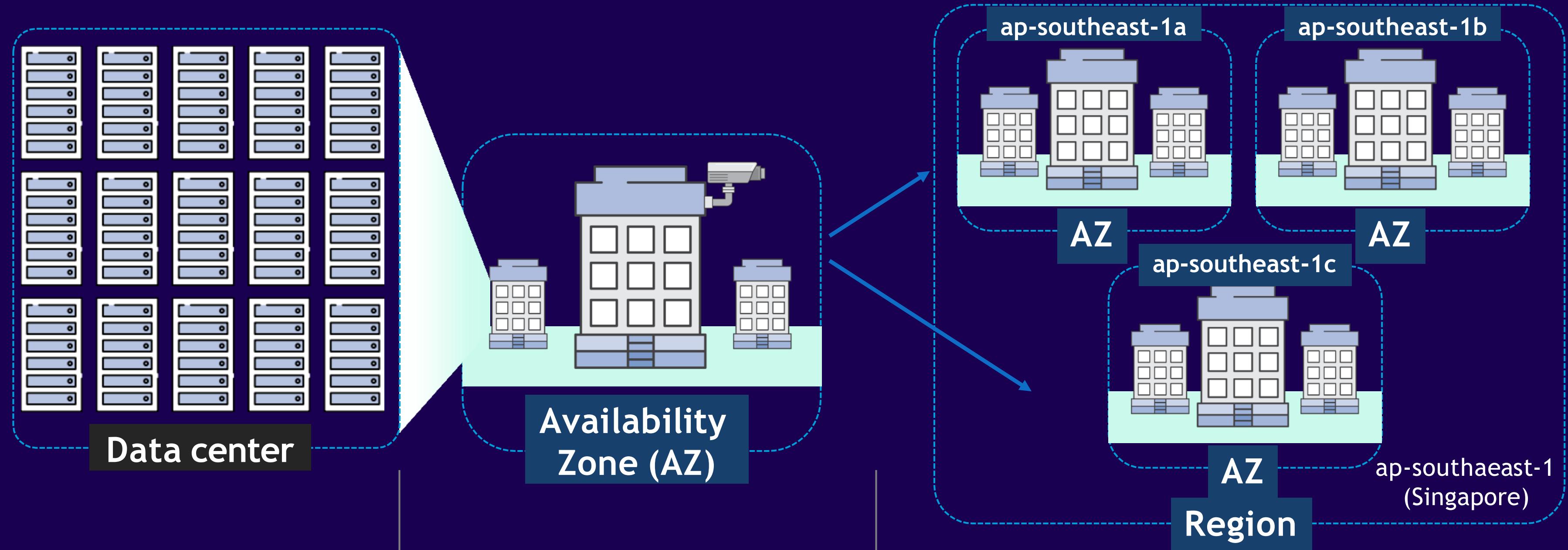


For these global services at the time of creation:

- There is no concept of region. eg. IAM User
- A single region must be explicitly chosen eg. S3 Bucket
- A group of regions are chosen eg. CloudFront Distribution

AWS Regions	
US East (N. Virginia)	us-east-1
US East (Ohio)	us-east-2
US West (N. California)	us-west-1
US West (Oregon)	us-west-2
Africa (Cape Town)	af-south-1
Asia Pacific (Hong Kong)	ap-east-1
Asia Pacific (Mumbai)	ap-south-1
Asia Pacific (Osaka)	ap-northeast-3
Asia Pacific (Seoul)	ap-northeast-2
Asia Pacific (Singapore)	ap-southeast-1
Asia Pacific (Sydney)	ap-southeast-2
Asia Pacific (Tokyo)	ap-northeast-1
Canada (Central)	ca-central-1
Europe (Frankfurt)	eu-central-1
Europe (Ireland)	eu-west-1
Europe (London)	eu-west-2
Europe (Milan)	eu-south-1
Europe (Paris)	eu-west-3
Europe (Stockholm)	eu-north-1
Middle East (Bahrain)	me-south-1
South America (São Paulo)	sa-east-1

AWS global infrastructure



Typically houses thousands of servers

- One or more data centers
- Designed for **fault isolation**

- Each AWS Region is made up of **two or more AZs**
- AWS has **36 Regions worldwide and growing**

AWS Infrastructure

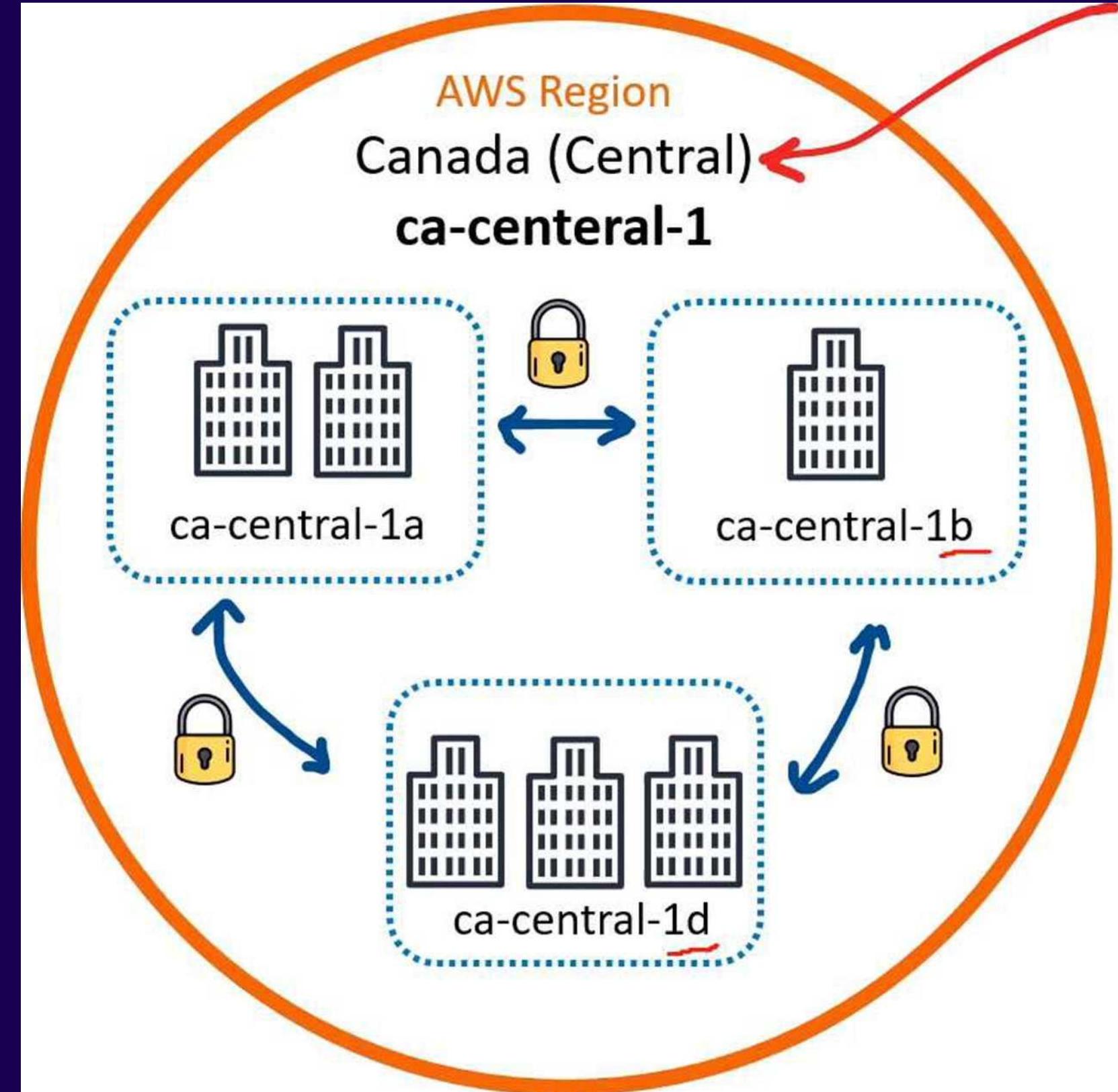
A region has multiple Availability Zones

An Availability Zone is made up of **one or more** datacenters

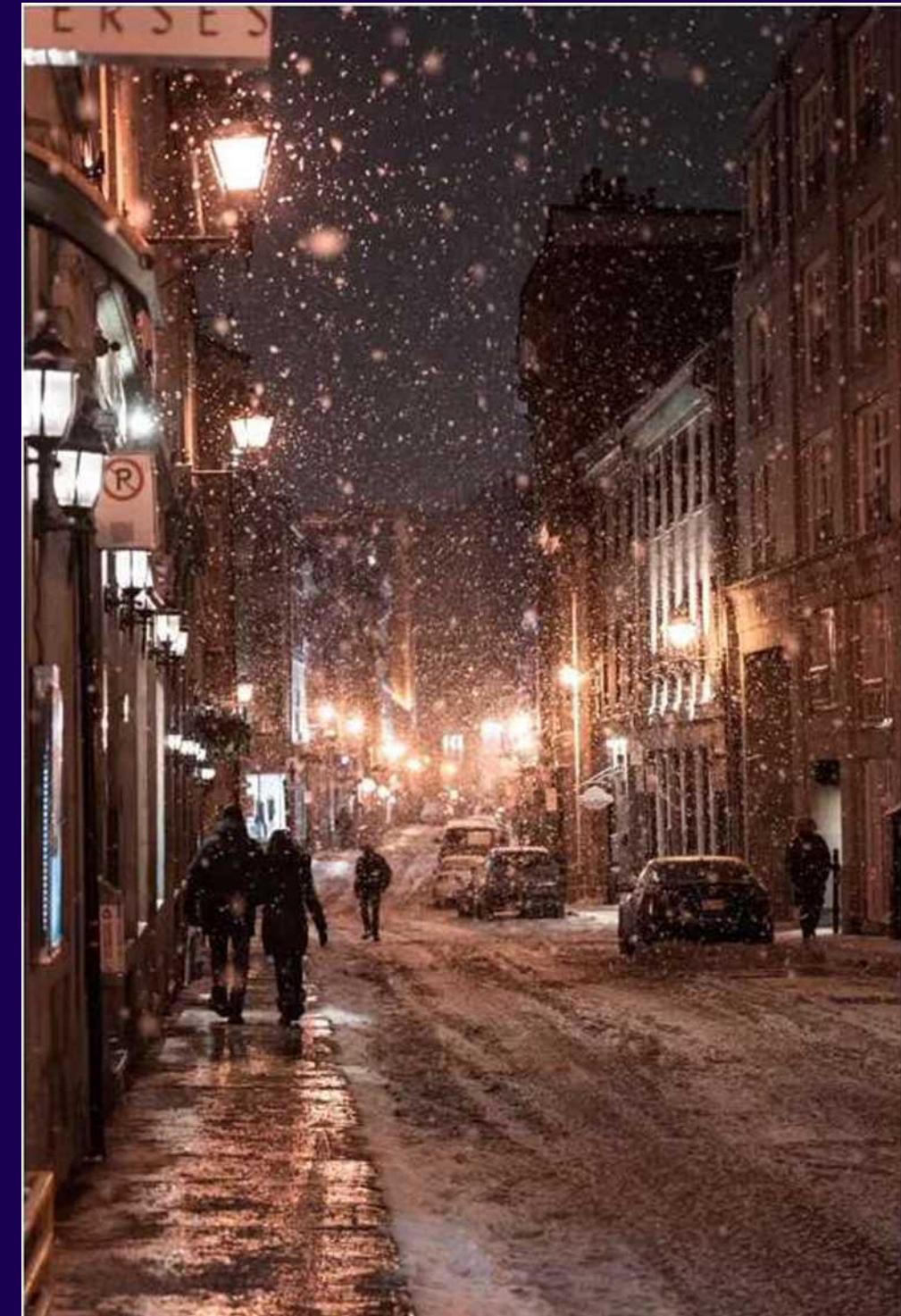
All AZs in an AWS Region are interconnected with high-bandwidth, low-latency networking, over fully redundant, dedicated metro fiber providing high-throughput, low-latency networking.

All traffic between AZs is encrypted

AZs are within 100 km (60 miles) of each other.



Montreal



AWS Infrastructure

Each Amazon Region is designed to be completely **isolated** from the other Amazon Regions.

- This achieves the greatest possible fault tolerance and stability

Each Availability Zone is **isolated**, but the Availability Zones in a Region are connected through low-latency links

Each Availability Zone is designed as an **independent failure zone**

- A "*Failure Zone*" is AWS describing a Fault Domain.

Failure Zone

- Availability Zones are physically separated within a typical metropolitan region and are located in lower risk flood plains
- discrete uninterruptible power supply (UPS) and onsite backup generation facilities
- data centers located in different Availability Zones are designed to be supplied by independent substations to reduce the risk of an event on the power grid impacting more than one Availability Zone.
- Availability Zones are all redundantly connected to multiple tier-1 transit providers



Multi-AZ for High Availability

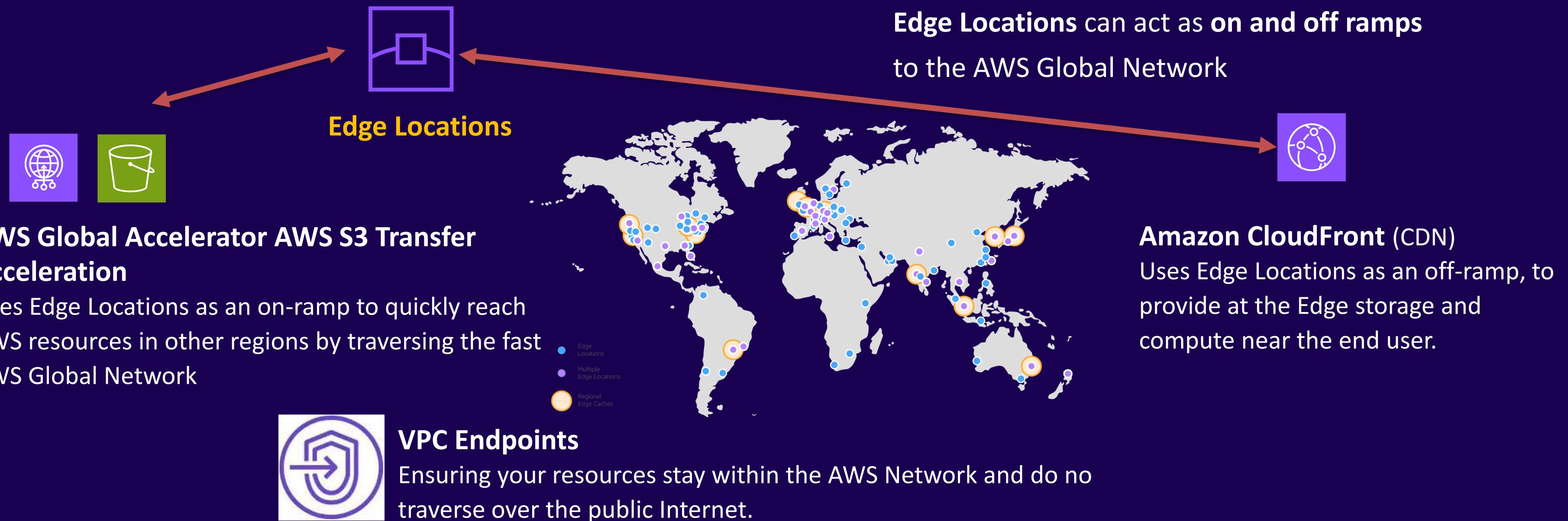
If an application is partitioned across AZs, companies are better isolated and protected from issues such as **power outages, lightning strikes, tornadoes, earthquakes, and more.**

AWS Global Infrastructure

The AWS Global Network represent the **interconnections between AWS Global Infrastructure**.

Commonly referred to as the "The Backbone of AWS"

Think of it as private expressway, where things can move very fast between datacenters.



AWS Global Infrastructure

Points of Presence (PoP) is an intermediate location between an AWS Region and the end user, and this location could be a datacenter or collection of hardware.

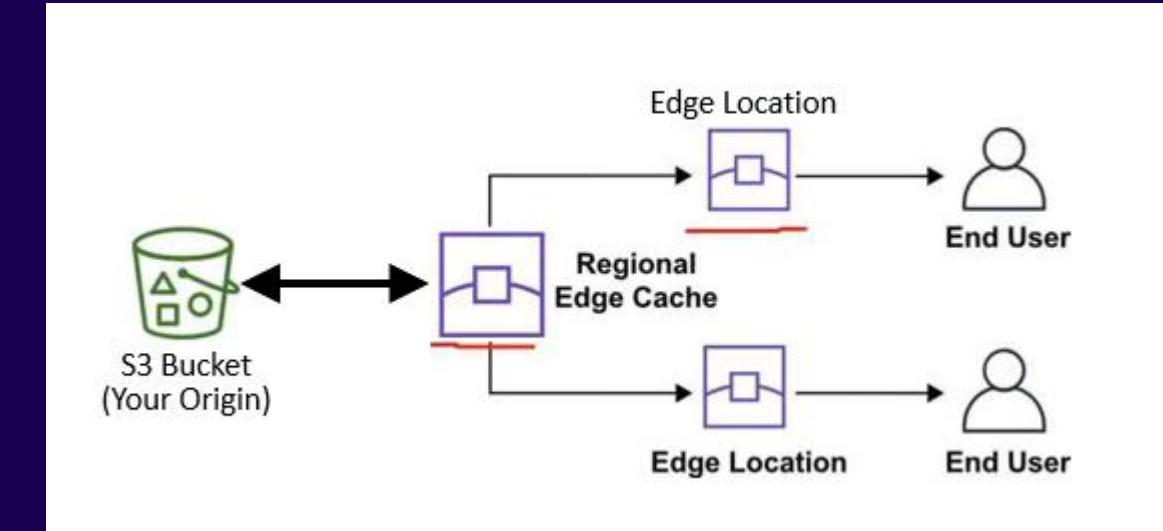
For AWS a Point of Presence is a data center **owned by AWS or a trusted partner** that is utilized by AWS Services related **for content delivery or expedited upload**.

PoP resources are:

- **Edge Locations**
- **Regional Edge Caches**

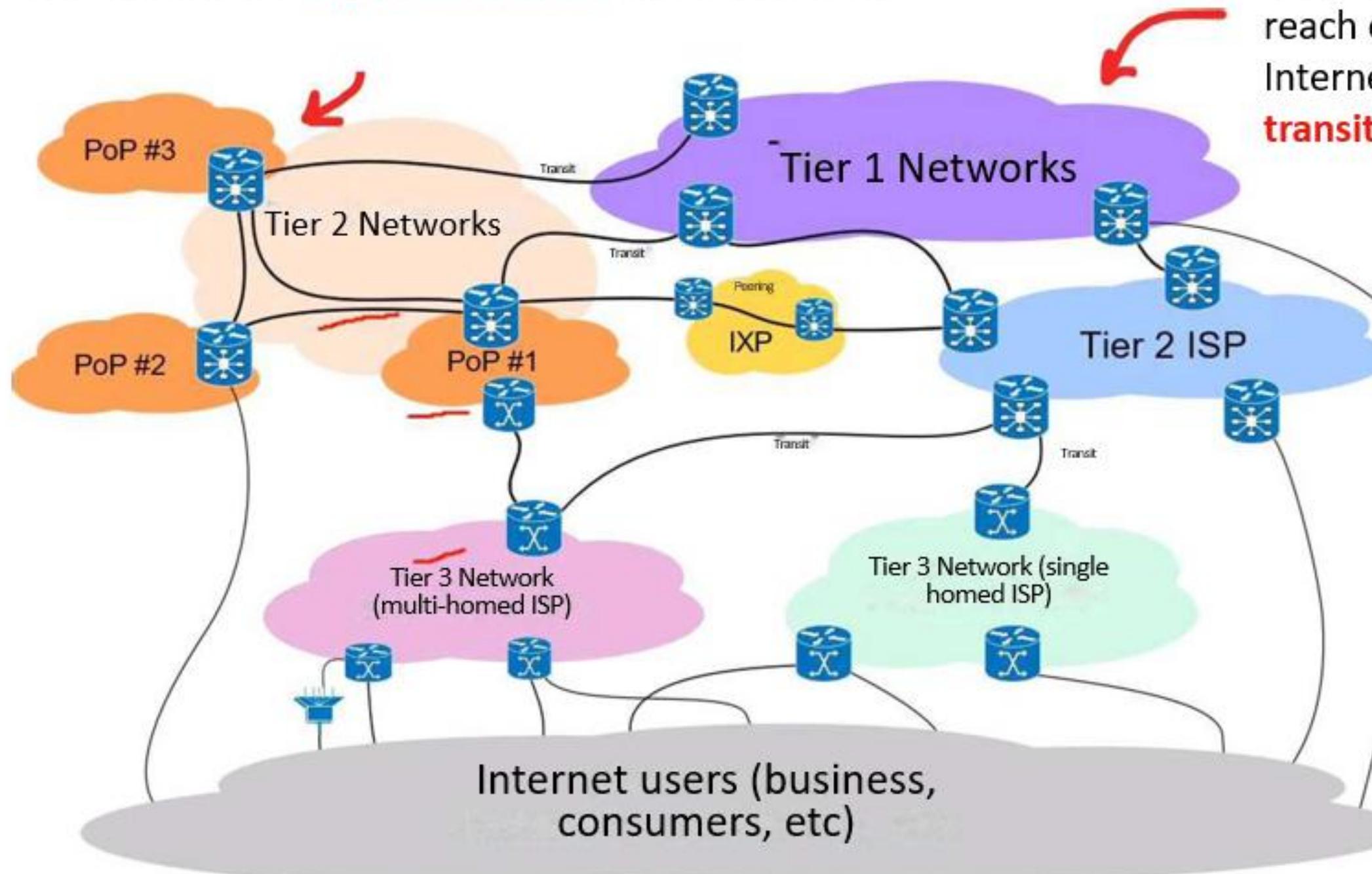
Edge Locations are datacenters that hold cached (copy) on the most popular files (eg. web pages, images and videos) so that the delivery of distance to the end users are reduced.

Regional Edge Locations are datacenters that hold much larger caches of less-popular files to reduce a full round trip and also to reduce the cost of transfer fees.



AWS Global Infrastructure

PoPs live at the **edge/intersection** of two networks

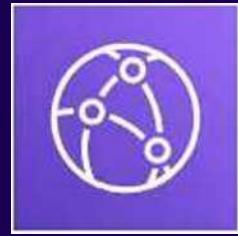


Tier 1 network is a network that can reach every other network on the Internet **without purchasing IP transit or paying for peering.**

AWS Availability Zones are all redundantly connected to multiple **tier-1 transit providers**

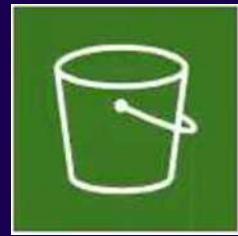
AWS Global Infrastructure

The following AWS Services use PoPs for content delivery or expedited upload.



Amazon CloudFront is a **Content Delivery Network (CDN) service** that:

- You point your website to CloudFront so that it will route requests to nearest Edge Location cache
- allows you to choose an **origin** (such as a web-server or storage) that will be source of cached
- caches the contents of what origin would returned to various Edge Locations around the world



Amazon S3 Transfer Acceleration allows you to generate a special URL that can be used by end users to upload files to a nearby Edge Location. Once a file is uploaded to an Edge Location, it can move much faster within the AWS Network to reach S3.



AWS Global Accelerator can find the optimal path from the end user to your web-servers. Global Accelerator are deployed within Edge Locations so you send user traffic to an Edge Location instead of directly to your web-application.

AWS Infrastructure

Other Services

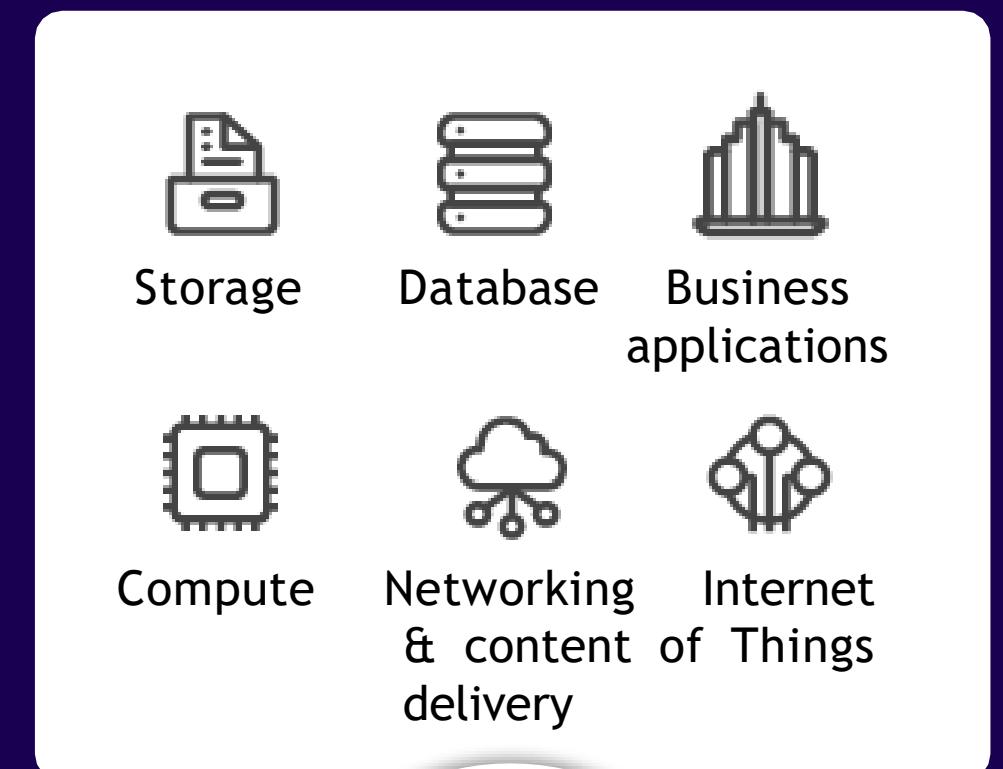
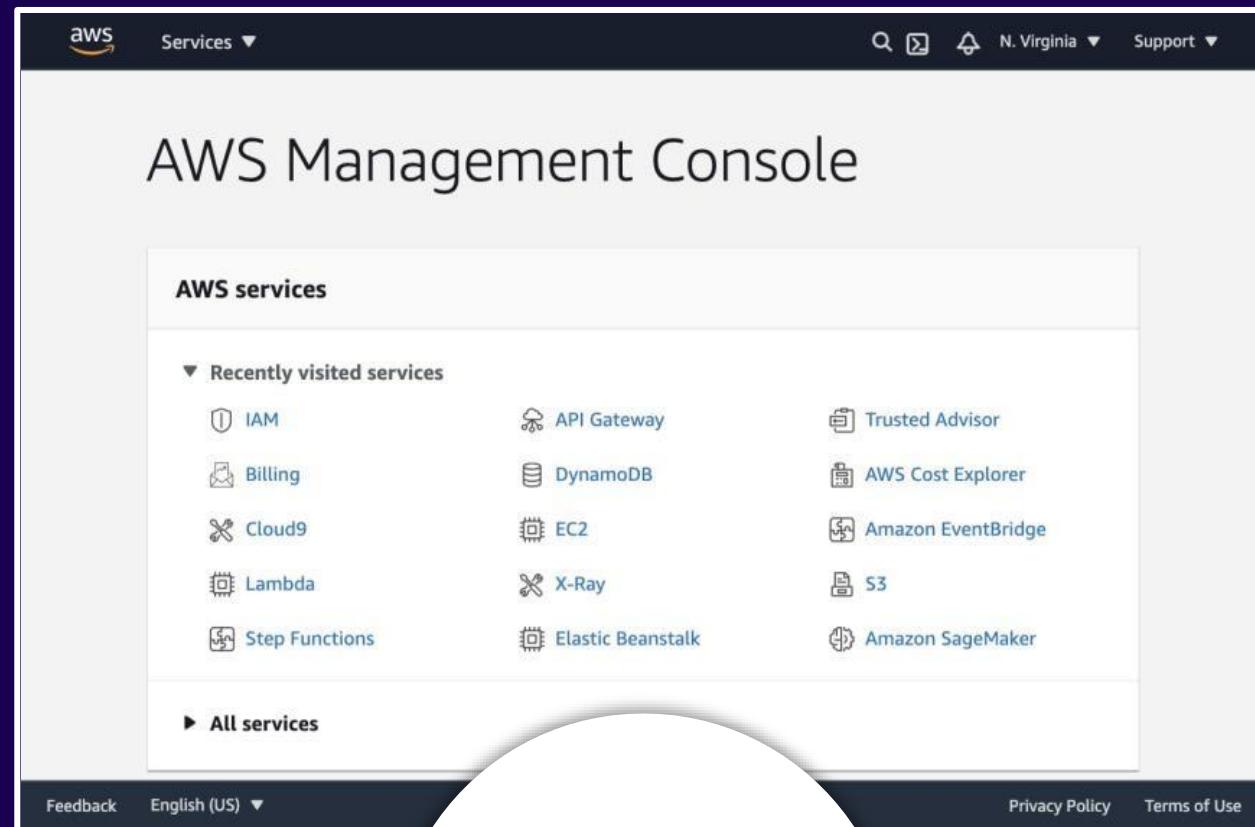
- AWS Direct Connect Service – Trusted Data Centers
- Local and Wavelength zones
- AWS Outposts

Other Requirements

- Data Residency
- AWS for Government
- AWS in China

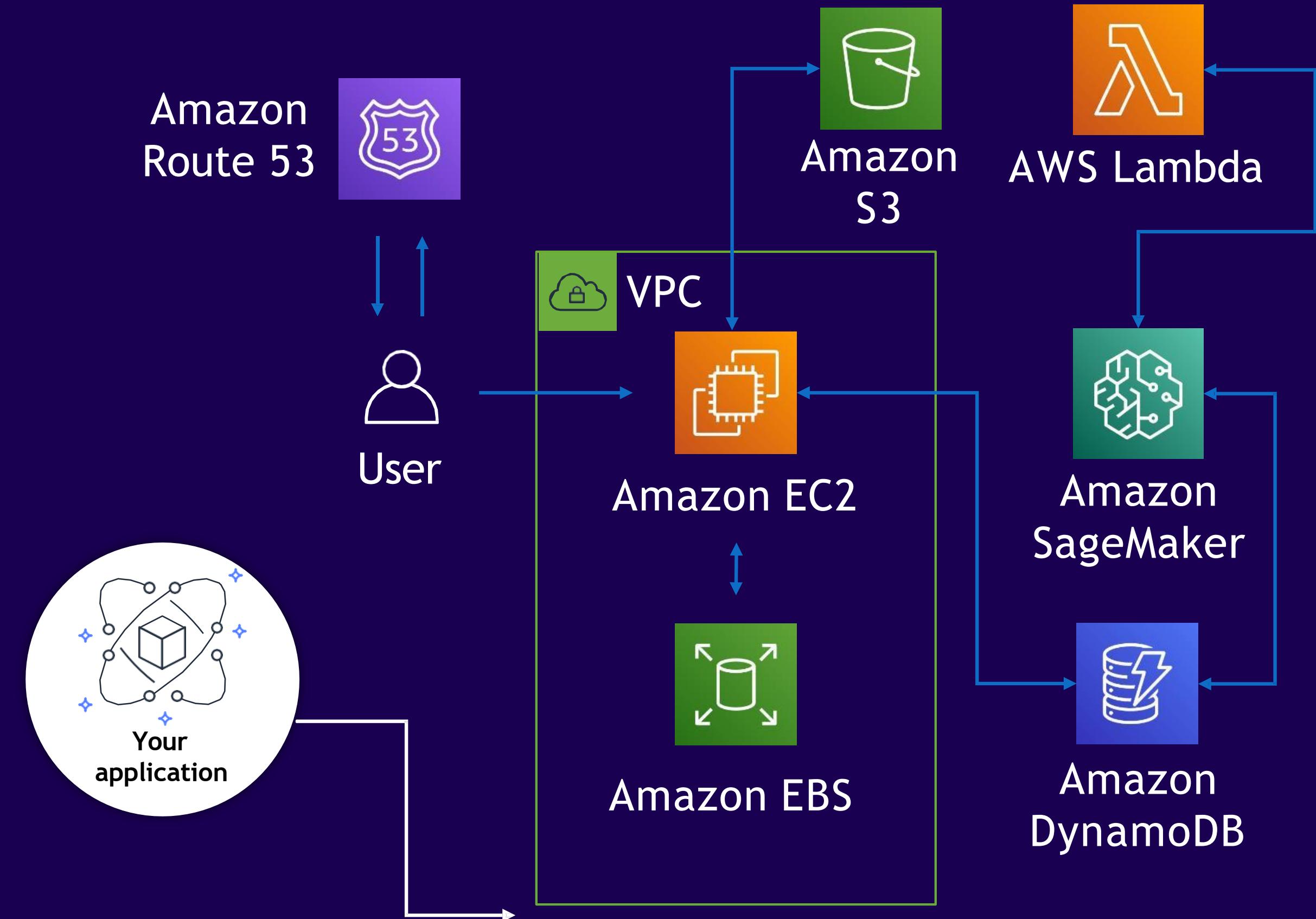
How does it work?

- AWS owns and maintains the network-connected hardware
- You provision and use what you need



Key service areas

Compute
Storage
Databases
Containers
Serverless
Machine Learning



Before we go on...

A few common cloud terms

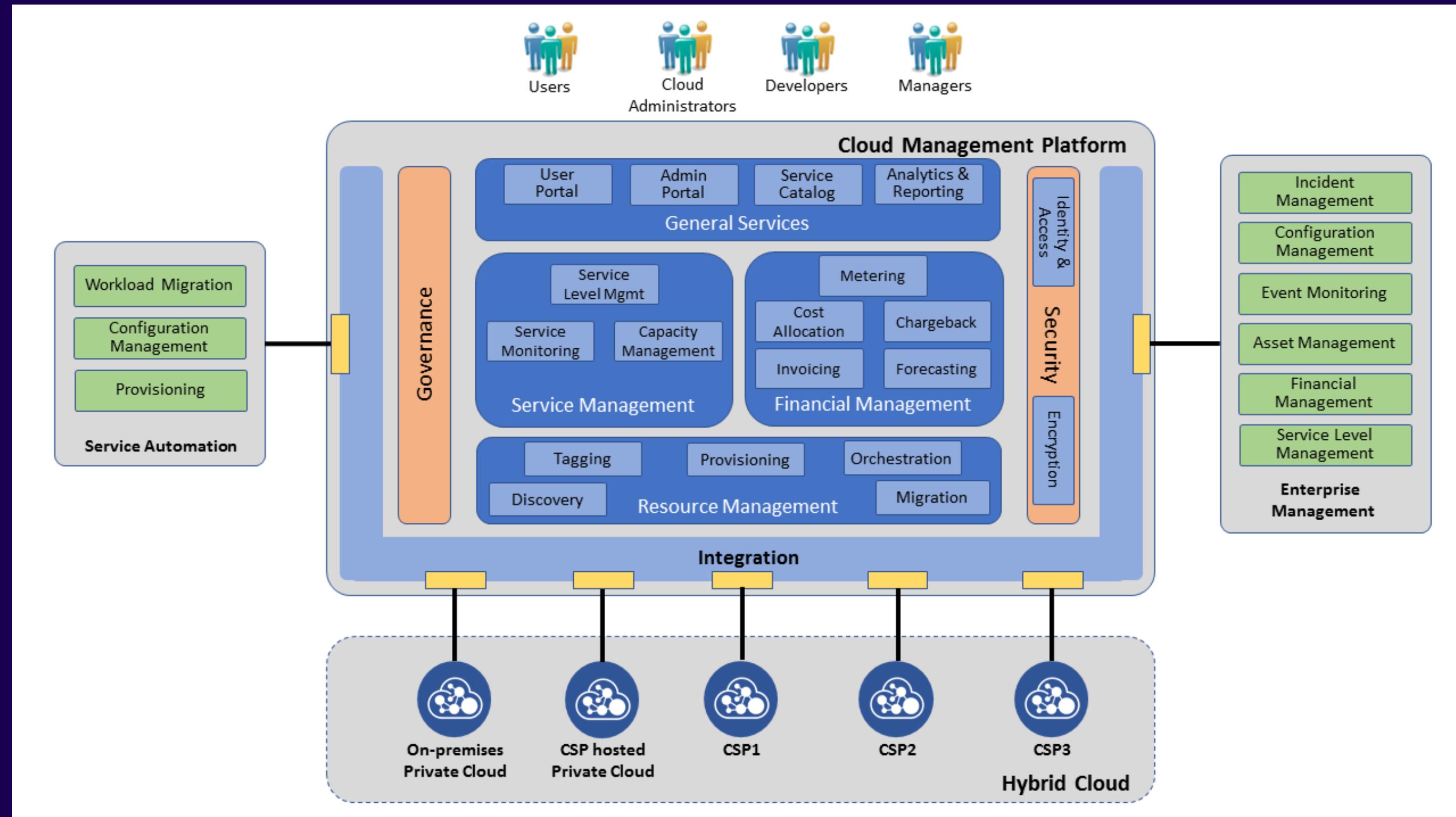
- Auto Scaling
- Big Data
- Cloud Computing
- Cloud Management Platform (CMP)
- Elasticity
- Platform-as-a-Service (PaaS)
- Public Cloud
- Serverless/Serverless Computing
- Backend-as-a-Service (BaaS)
- Cloud Broker
- Cloud Service Provider
- Hybrid Cloud
- Infrastructure-as-a-Service (IaaS)
- Private Cloud
- Software-as-a-Service (SaaS)

Cloud Management Platforms (CMP)

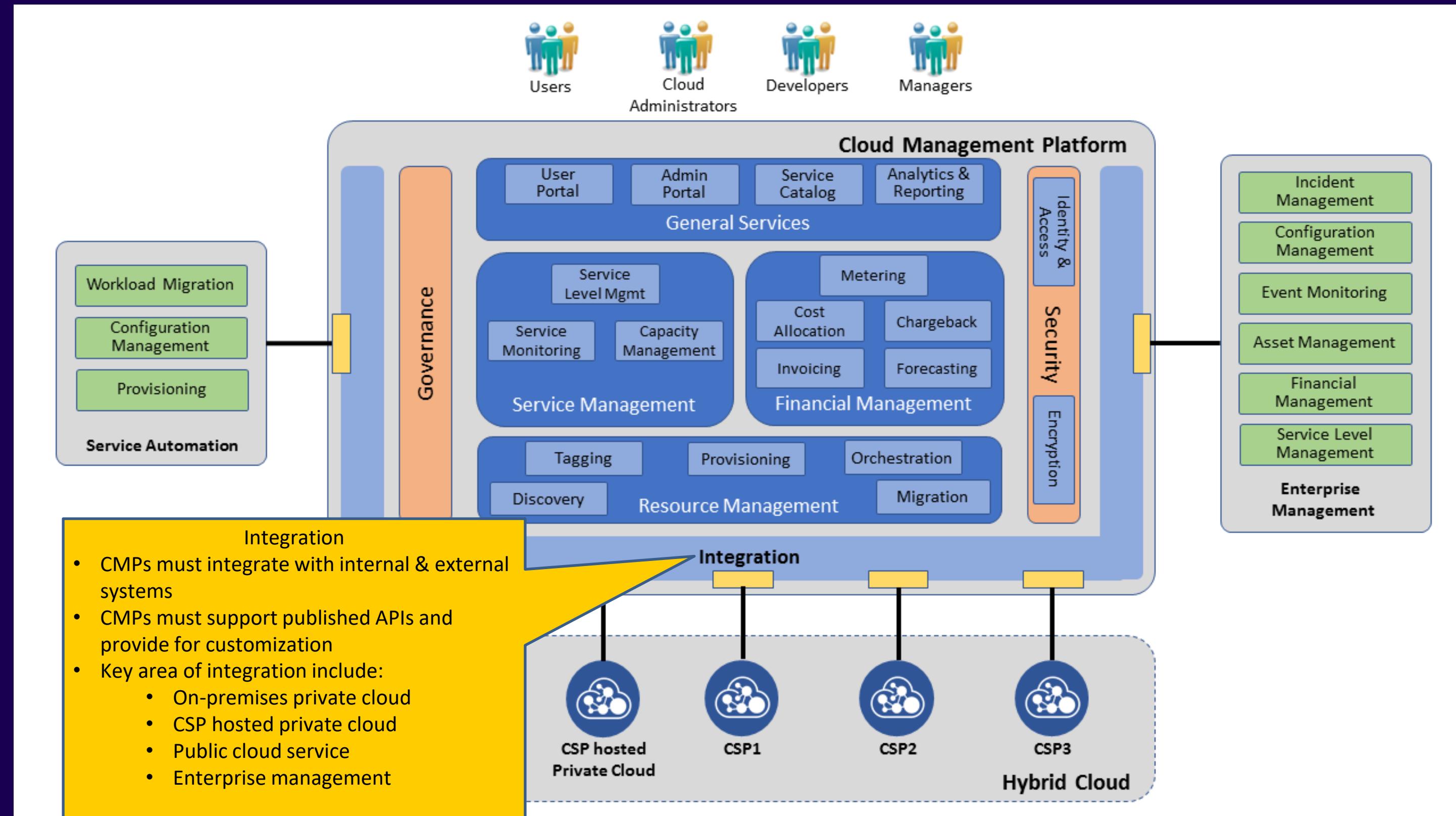
CMPs incorporate self-service interfaces, provision system images, enable metering and billing, and provide for some degree of workload optimization through established policies

Functionality	Integration Points
<ul style="list-style-type: none">■ Access and authorization management■ Resource management across environments■ Financial management relating to subscribed cloud services■ Service catalogs to support self-service provisioning or resource approvals■ Cloud brokerage – rules-based guidance for asset placement decisions■ Integration with the relevant target cloud environments & enterprise systems	<ul style="list-style-type: none">■ Service delivery systems■ Identity and access management■ ERP and financial systems■ Automation Tools■ Infrastructure monitoring■ Business process rules systems or other business systems

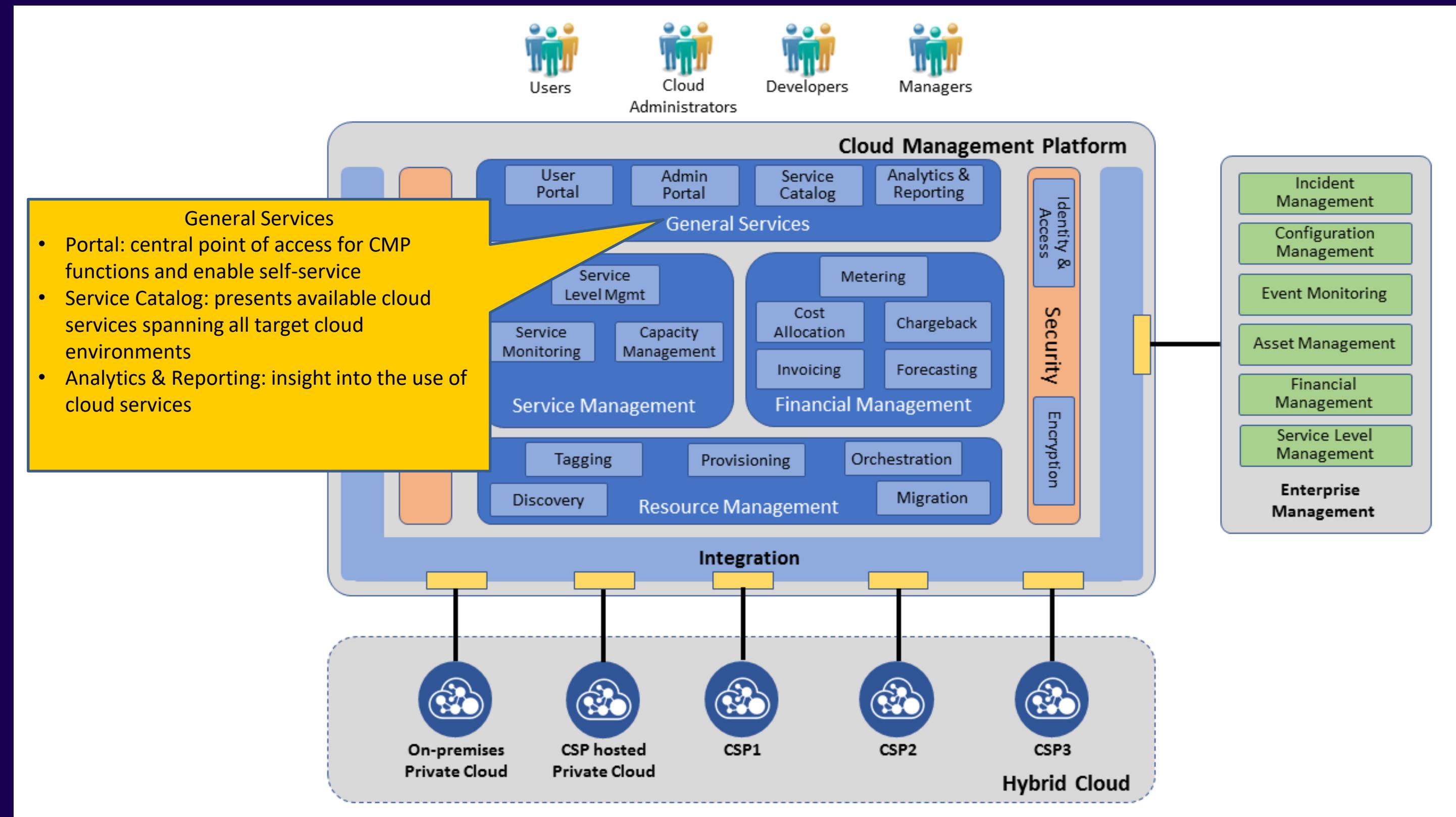
Cloud Management Platforms (CMP)



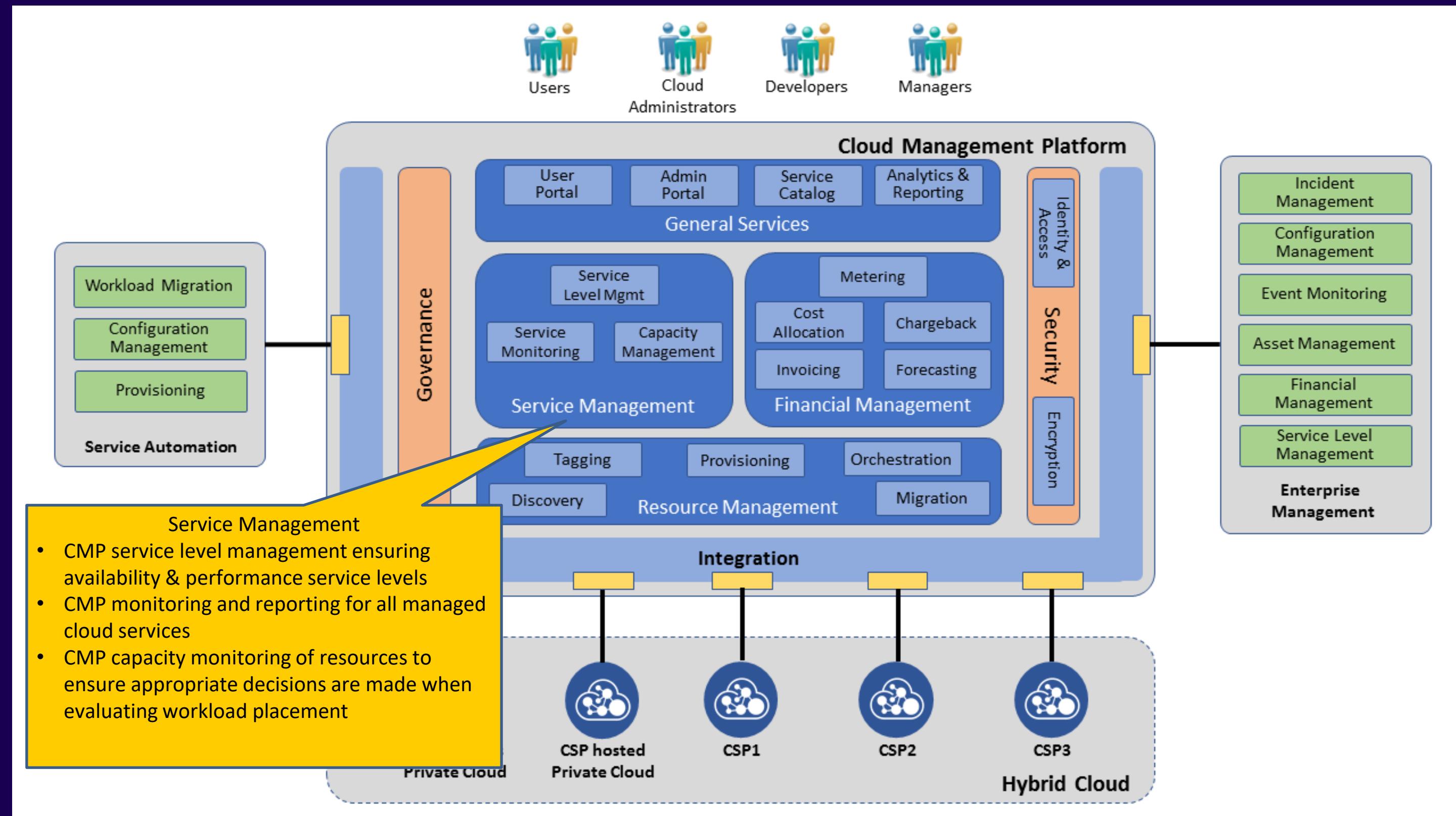
Cloud Management Platforms (CMP)



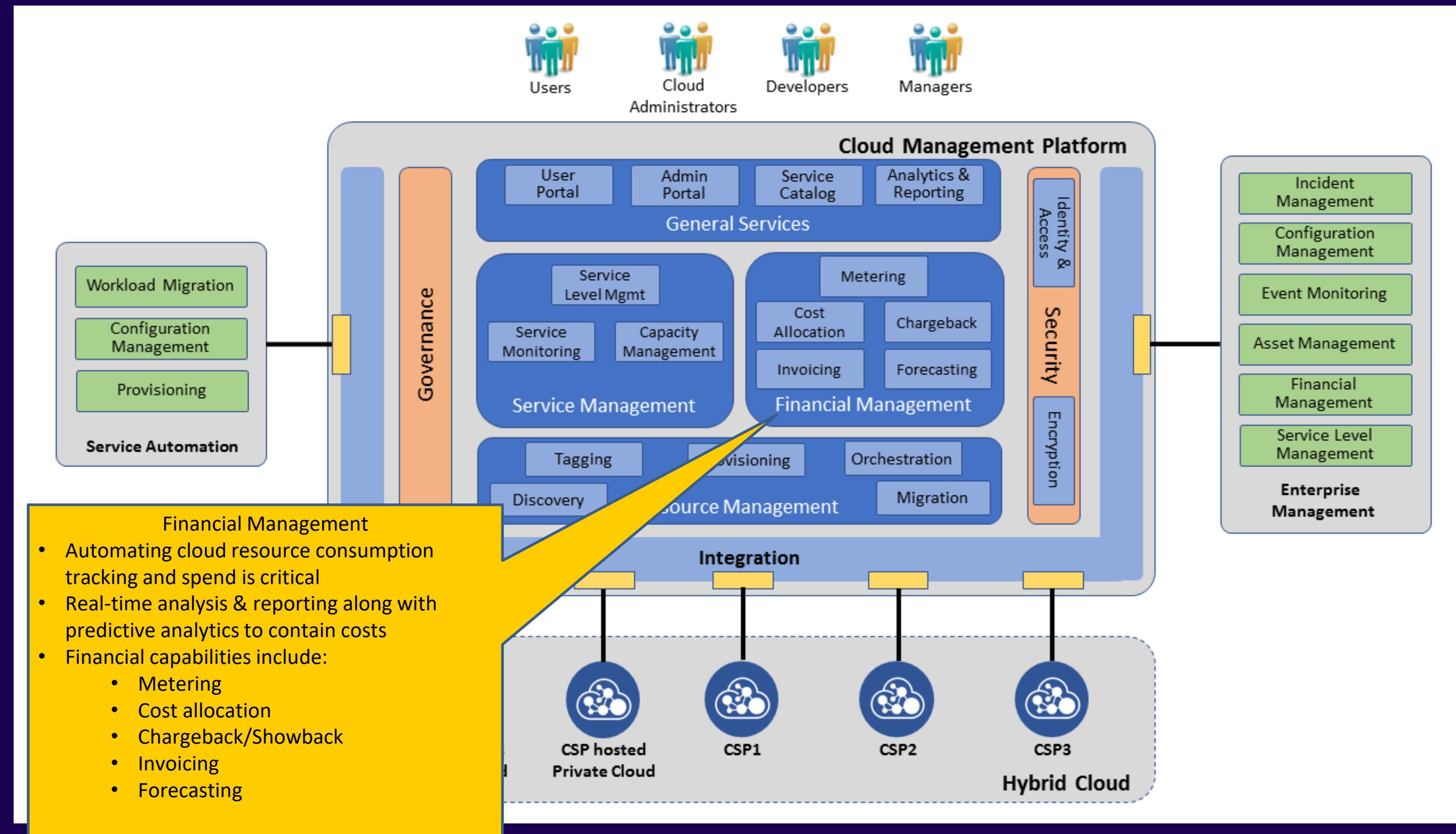
Cloud Management Platforms (CMP)



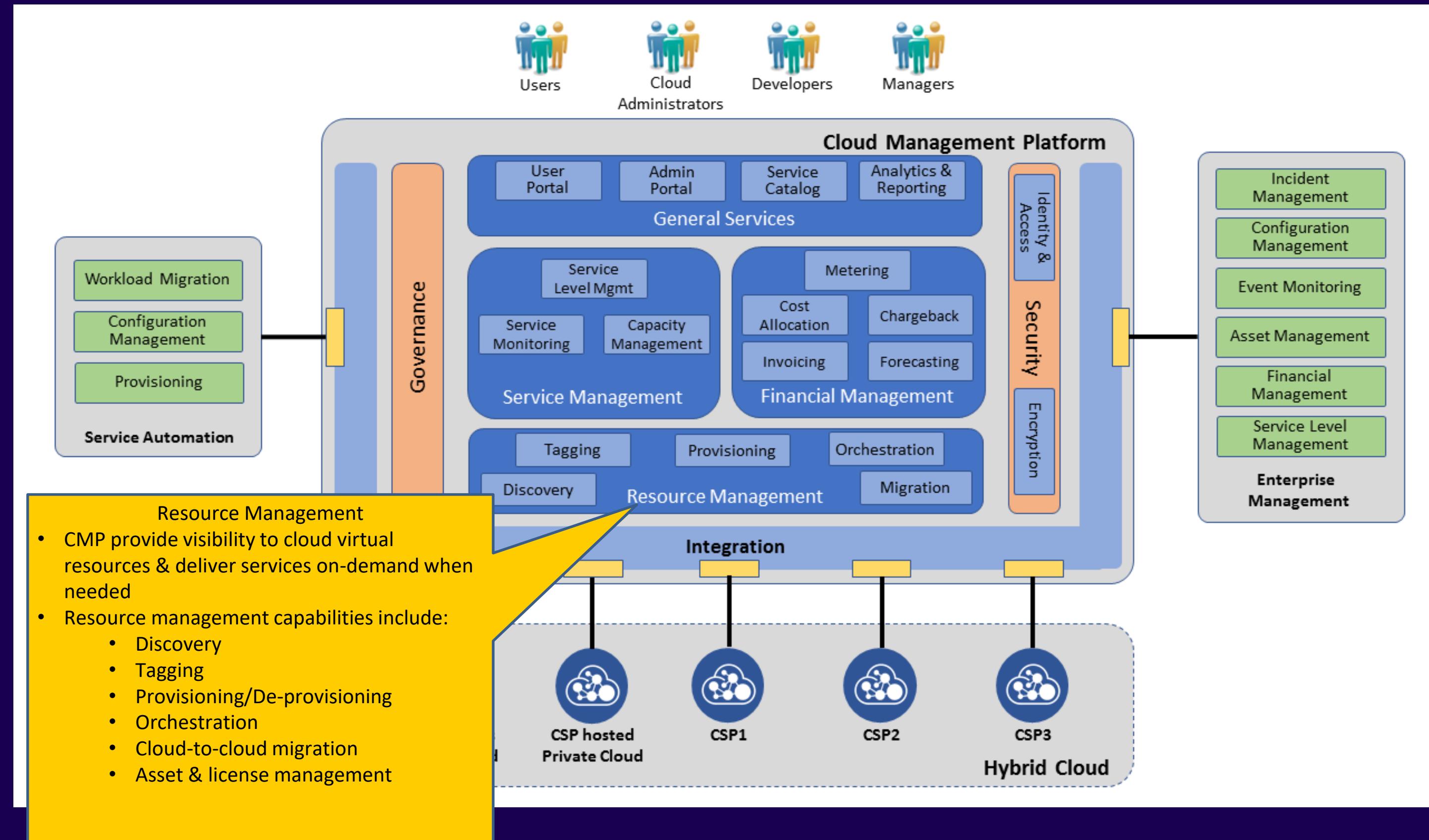
Cloud Management Platforms (CMP)



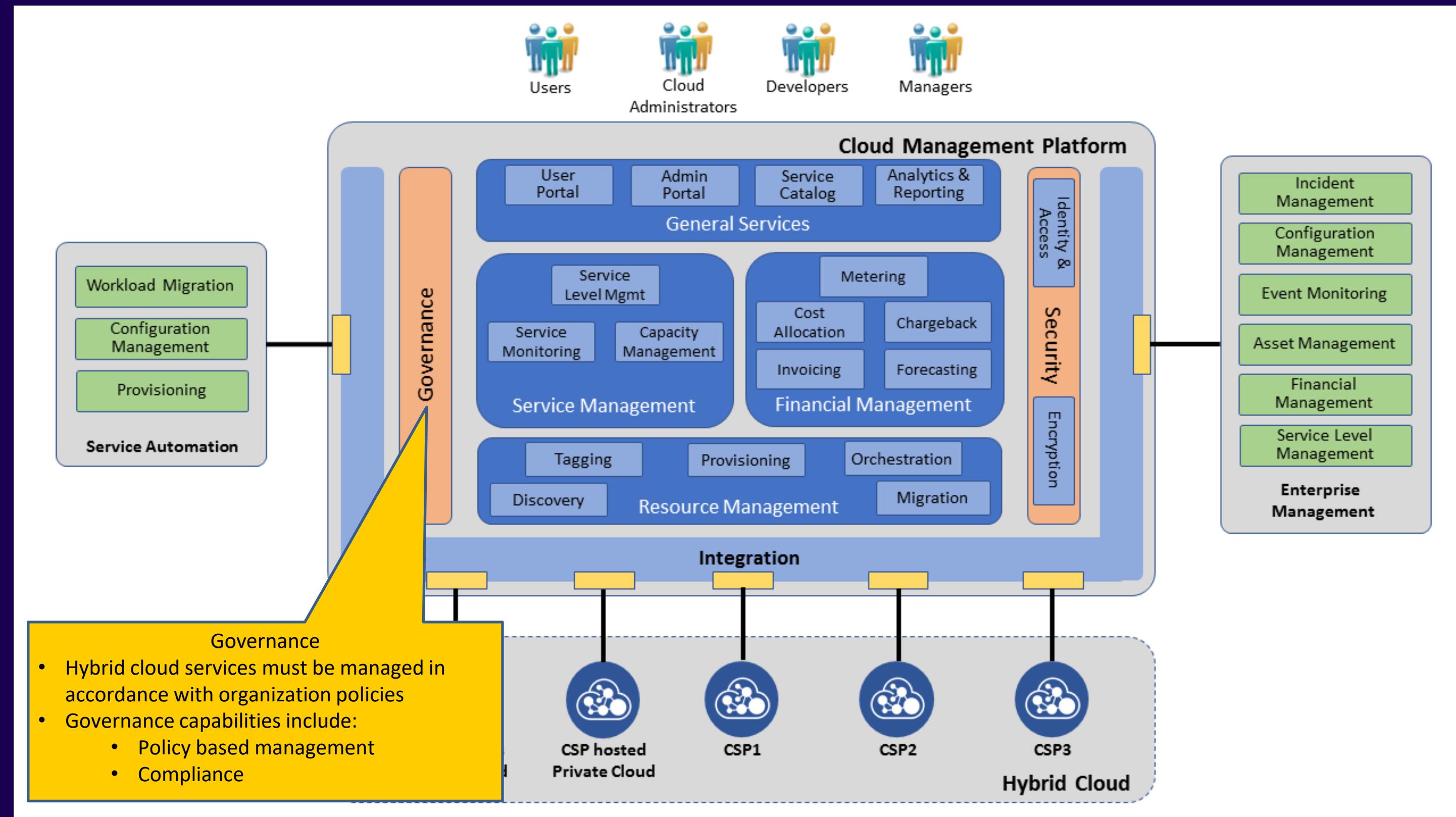
Cloud Management Platforms (CMP)

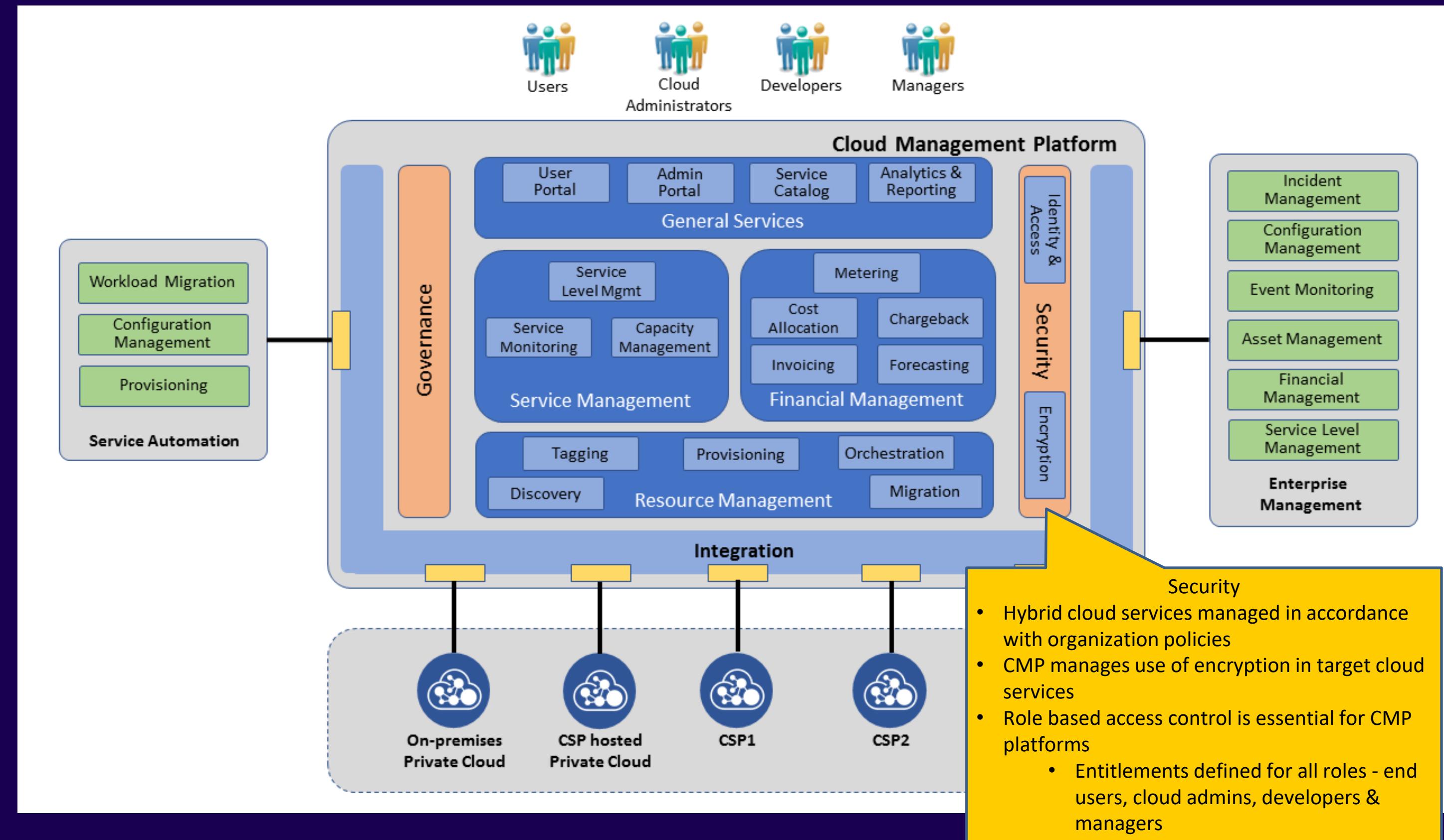


Cloud Management Platforms (CMP)



Cloud Management Platforms (CMP)





Backend as a Service

- A BaaS or mBaaS or Backend as a Service is a platform that automates backend side development and takes care of the cloud infrastructure.
- Using a BaaS, responsibilities of running and maintaining servers are outsourced to a third party and focus on the frontend or client-side development.
- A BaaS will provide a set of tools to help you to create a backend code and speed up the development process.
- It has ready to use features such as data management, APIs, social media integrations, file storage, and push notifications.

Backend as a Service

Some use cases for using a backend as a service platform:

- Making an MVP - Minimum Viable Product
- Stand-alone apps or applications that require a small number of integrations
- Enterprise apps that are not mission-critical
- For these cases, using a BaaS is a no-brain and will save you a lot of money and time.

Backend as a Service

Advantages of a Backend as a Service

- Development speed - It's super fast
- Development price - It's really cheap
- It's serverless, and you don't need to manage infrastructure

Disadvantages of a Backend as a Service

- Less flexibility in comparison to custom coding
- A lower level of customization in comparison to a custom backend
- Vendor lock-in for closed source platforms

Backend as a Service

Most common features of a backend as a service.

- Data Management
- User authentication
- Social Integration (Facebook, LinkedIn, Twitter, etc.)
- Email Verification
- Push Notifications
- Cloud Code Functions
- Geolocation
- Push Notifications
- Logs
- CDN and Cache
- Infrastructure (Security settings, auto-scaling, data backup, DB optimization)

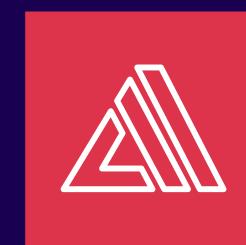
Backend as a Service

A BaaS platform is a technical service and designed for developers.

A user with no specialized skills will face challenges in using it.

The most common uses cases are:

- Frontend engineers with limited knowledge in backend development
- Backend engineers that want to speed up development
- Engineers that wish to outsource low value/repetitive tasks to a third party

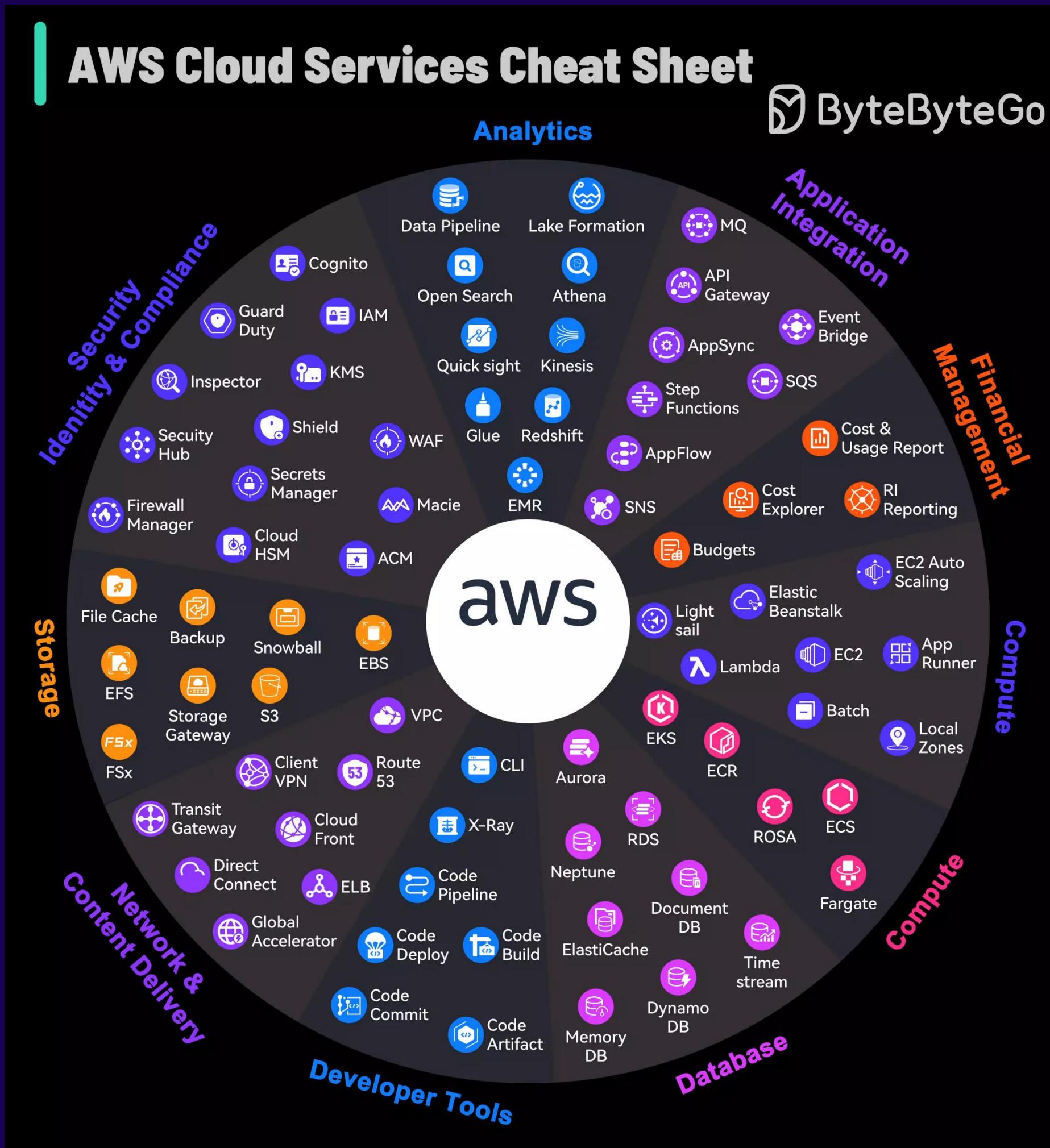


AWS Amplify



AWS Amplify Studio

AWS Services



Some Core Service Categories

General Resources

Analytics

Application Integration

Blockchain

Business Applications

Cloud Financial Management

Compute

Contact Center

Containers

Customer Enablement

Database

Developer Tools

End User Computing

Front-End Web & Mobile

Games

Internet of Things

Machine Learning

Management & Governance

Media Services

Migration & Transfer

Networking & Content Delivery

Quantum Technologies

Robotics

Satellite

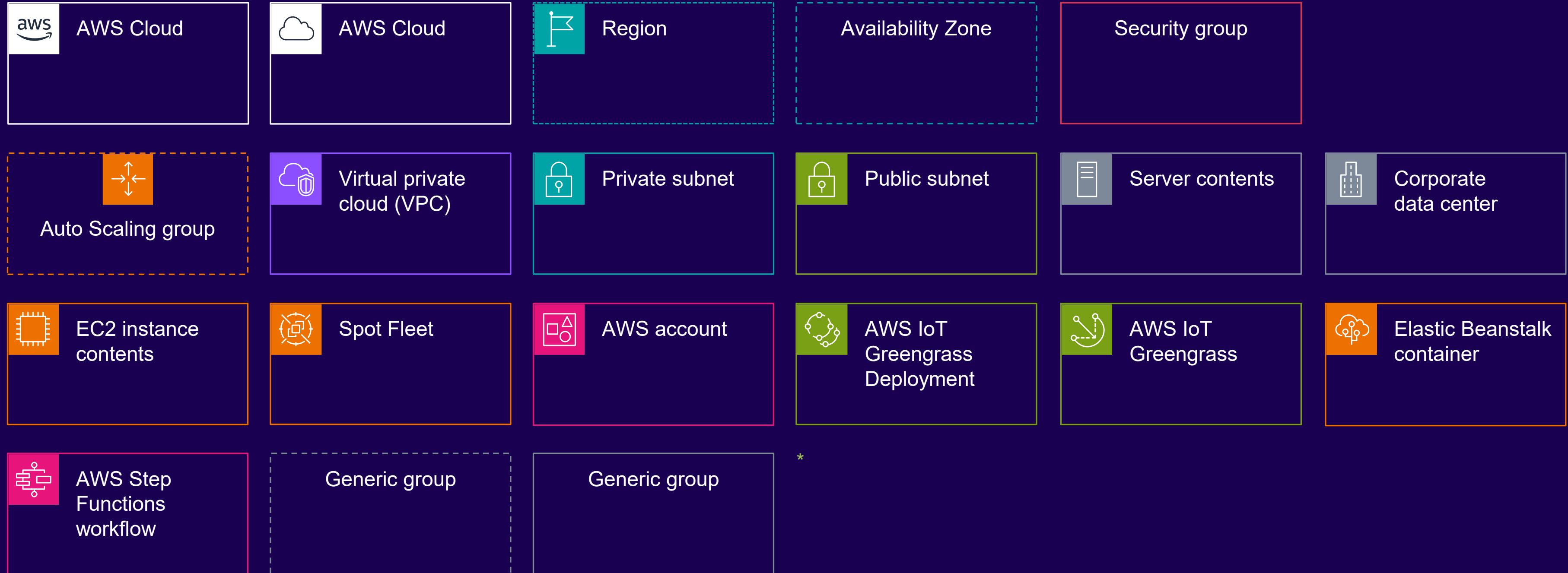
Security, Identity, & Compliance

Serverless

Storage

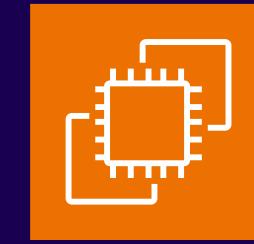
Groups

*



Compute

Services

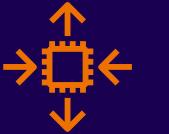


Amazon Elastic Compute
Cloud (Amazon EC2)

Resources



AMI



Auto scaling



Elastic IP
address

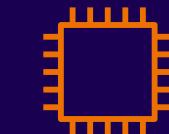


Rescue

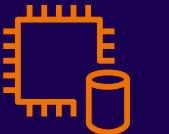


AWS Microservice
Extractor for .NET

Instances



Instance*



DB instance



Instances



Spot Instance



Instance with
CloudWatch

*Specify the instance type. Example labels: M5n instance, C4 instance, High Memory instance.
Refer to the latest list of [EC2 instance types](#).

Compute

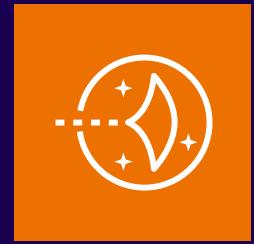
Services



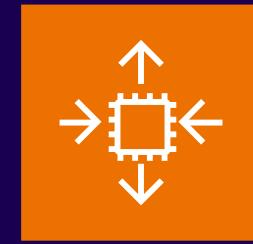
AWS Elastic Beanstalk



AWS Lambda



Amazon Lightsail



Amazon EC2
Auto Scaling

Resources



Deployment



Application



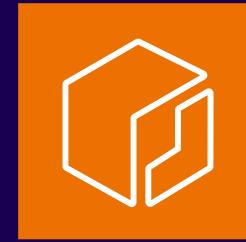
Lambda function

Containers

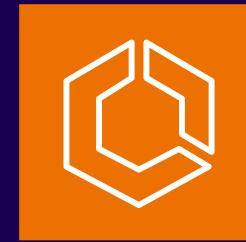
Services



Amazon Elastic Kubernetes
Service (Amazon EKS)



Amazon Elastic Container
Registry (Amazon ECR)



Amazon Elastic Container
Service (Amazon ECS)

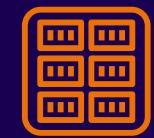


Amazon ECS Anywhere

Resources



EKS on Outposts



Registry



Image



Container 1



Container 2



Container 3



Task



Service



ECS Service
Connect



Copilot CLI

Containers

Services



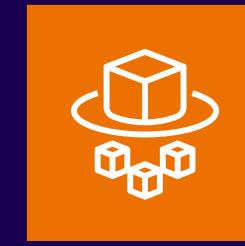
Amazon EKS Anywhere



Amazon EKS Cloud



Amazon EKS Distro

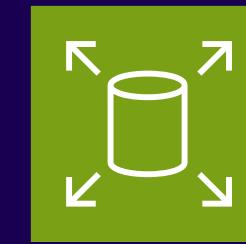


AWS Fargate

Resources

Storage

Services



Amazon Elastic Block Store
(Amazon EBS)



AWS Storage
Gateway

Resources



Snapshot



Multiple
volumes



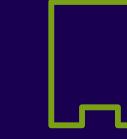
Amazon Data
Lifecycle Manager



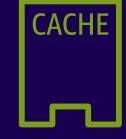
Volume



Volume gp3



Noncached
volume



Cached
volume



Virtual tape
library



Tape
Gateway



Volume
Gateway



File
gateway



Amazon FSx
File Gateway



Amazon S3
File Gateway

Storage

Services



Amazon Simple Storage
Service (Amazon S3)

Storage Classes



S3 Standard



S3 Intelligent-
Tiering



S3 Standard-IA



S3 One
Zone-IA



S3 Glacier
Flexible Retrieval



S3 Glacier
Deep Archive



S3 Glacier
Instant Retrieval



S3 on
Outposts

Resources



Bucket



Bucket with
objects



S3 Replication



S3 Replication
Time Control



S3 Storage
Lens



Access points



VPC access
points



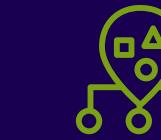
Object



S3 Object
Lambda



S3 Object Lambda
Access Points



S3 Multi-Region
Access Points



S3 Select



S3 Object Lock



S3 Batch
Operations

Storage

Services



Amazon Elastic File System
(Amazon EFS)



AWS Elastic Disaster Recovery
(AWS DRS)



Amazon FSx

Resources



File system



EFS Standard



EFS Standard–
Infrequent Access



EFS Intelligent-
Tiering



EFS One Zone



EFS One Zone–
Infrequent Access



Elastic Throughput

Database

Services



Amazon Aurora



Amazon DocumentDB
(with MongoDB compatibility)



Amazon DynamoDB

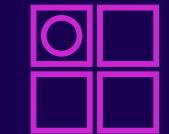
Resources



Trusted Language
Extensions for
PostgreSQL



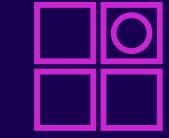
Amazon DocumentDB
Elastic Clusters



Attribute



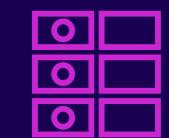
Items



Item



Table



Attributes



Global
secondary
index



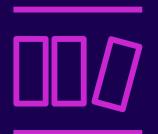
Amazon DynamoDB
Accelerator (DAX)



Standard Access
table class



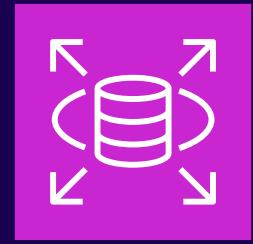
Standard-Infrequent
Access table class



Stream

Database

Services



Amazon Relational Database
Service (Amazon RDS)



Amazon RDS on
VMware



Amazon MemoryDB
for Redis

Resources



Multi-AZ



Multi-AZ DB
cluster



Blue/Green
Deployments



Optimized
Writes



Trusted Language
Extensions for
PostgreSQL

Database

Services



Amazon ElastiCache

Resources



Cache node



ElastiCache
for Redis



ElastiCache for
Memcached

Database

Resources

 Amazon Aurora instance	 Amazon Aurora instance alternate	 Amazon RDS instance	 Amazon RDS instance alternate	 MySQL instance	 MySQL instance alternate
 Oracle instance	 Oracle instance alternate	 PostgreSQL instance	 PostgreSQL instance alternate	 PIOPS instance	 Amazon RDS proxy instance alternate
 MariaDB instance	 MariaDB instance alternate	 SQL Server instance	 SQL Server instance alternate	 Amazon RDS proxy instance	

Developer Tools

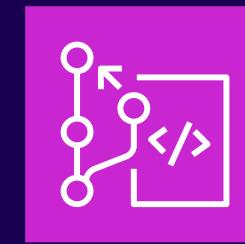
Services



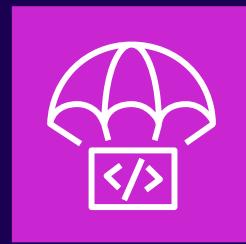
AWS Cloud9



AWS CodeBuild



AWS CodeCommit



AWS CodeDeploy

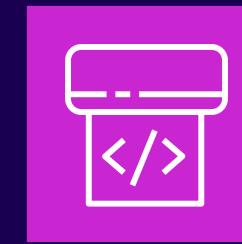
Resources



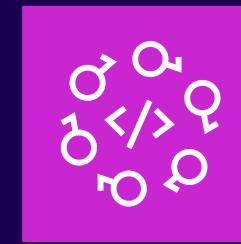
Cloud9

Developer Tools

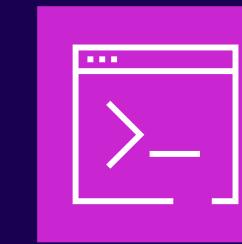
Services



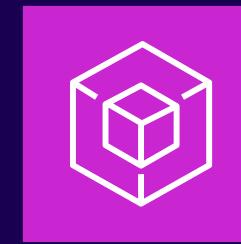
AWS CodePipeline



AWS CodeStar



AWS Command Line Interface (AWS CLI)



AWS Tools and SDKs

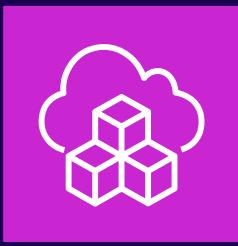
Resources

Developer Tools

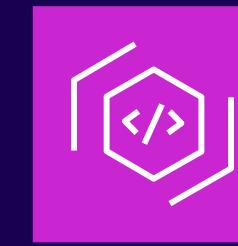
Services



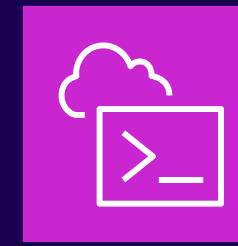
AWS X-Ray



AWS Cloud Development Kit
(AWS CDK)



AWS CodeArtifact



AWS CloudShell

Resources

Developer Tools

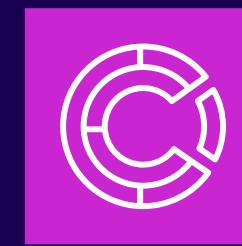
Services



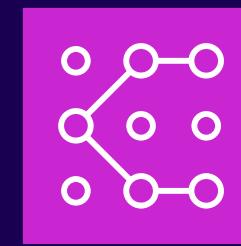
Amazon Corretto



AWS Cloud Control API



Amazon CodeCatalyst

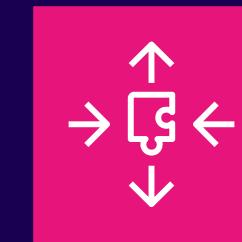


AWS Application Composer

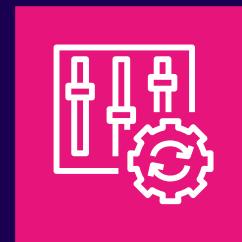
Resources

Management & Governance

Services



AWS Application
Auto Scaling



AWS Config



AWS License Manager



AWS Health Dashboard

Resources



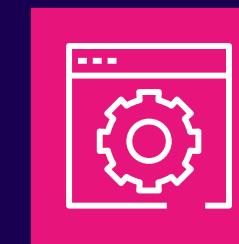
License blending



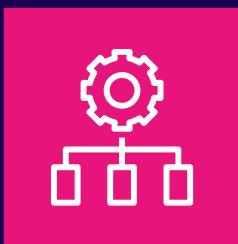
Application discovery

Management & Governance

Services



AWS Management
Console



AWS OpsWorks

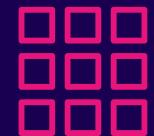


AWS CloudTrail

Resources



Apps



Instances



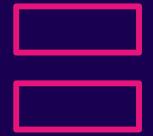
Monitoring



Permissions



Deployments



Layers



Resources



Stack2



CloudTrail Lake

Networking & Content Delivery

Services



Amazon Virtual Private Cloud
(Amazon VPC)

Resources



Customer gateway



Internet gateway



VPN gateway



Flow logs



Endpoints



Router



Traffic mirroring



NAT gateway



Elastic network
interface



Elastic network
adapter



Network access
control list



VPN connection



Peering
connection



Reachability
Analyzer



Carrier gateway



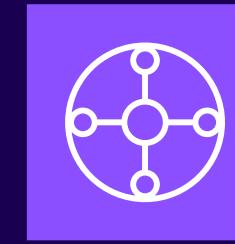
Network Access
Analyzer



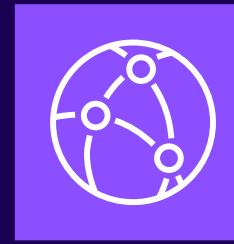
Virtual private
cloud (VPC)

Networking & Content Delivery

Services



AWS Transit Gateway



Amazon CloudFront



Amazon Route 53

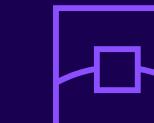
Resources



Attachment



Download distribution



Edge location



Streaming distribution



Functions



Hosted zone



Route table



Readiness checks



Resolver



Resolver DNS firewall



Routing controls



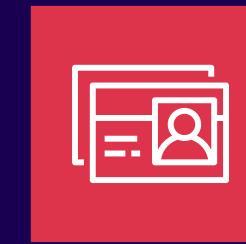
Resolver query logging



Route 53 Application Recovery Controller

Security, Identity, & Compliance

Services



AWS Directory Service



AWS Firewall Manager



AWS Identity and Access Management (IAM)

Resources



Simple AD



AD Connector



AWS Managed Microsoft AD



Add-on



Permissions



MFA token



Role



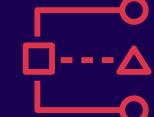
AWS STS



AWS STS alternate



Long-term security credential



IAM Access Analyzer



Encrypted data



Data encryption key



Temporary security credential



IAM Roles Anywhere

Security, Identity, & Compliance

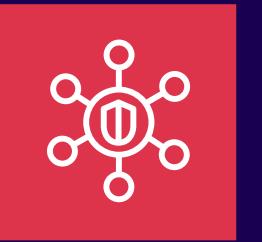
Services



AWS Key Management Service (AWS KMS)



AWS Secrets Manager



AWS Security Hub



AWS Shield

Resources



External Key Store (XKS)



Finding



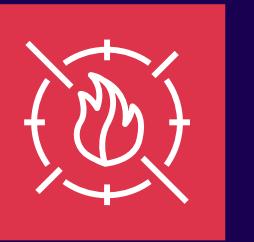
AWS Shield Advanced

Security, Identity, & Compliance

Services



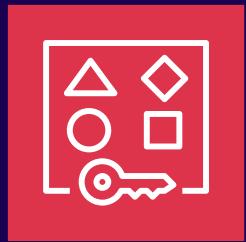
AWS IAM Identity Center



AWS WAF



Amazon Detective

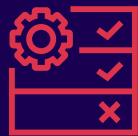


AWS Resource Access Manager

Resources



Rule



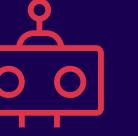
Managed rule



Filtering rule



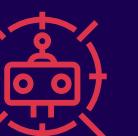
Labels



Bot



Bad bot



Bot control

ARNS

Amazon Resource Names (ARNs) uniquely identify AWS resources.

ARNs are required to specify a resource unambiguously across all of AWS

ARNS

The ARN has the following *arn:partition:service:region:account-id:resource-id*
format variations *arn:partition:service:region:account-id:resource-type/resource-id*
arn:partition:service:region:account-id:resource-type:resource-id

Partition

- aws - AWS Regions
- aws-cn - China Regions
- aws-us-gov - AWS GovCloud (US) Regions

Service - Identifies the service

- ec2
- s3
- iam

Region - which AWS resource

- us-east-1
- ca-central-1

Account ID

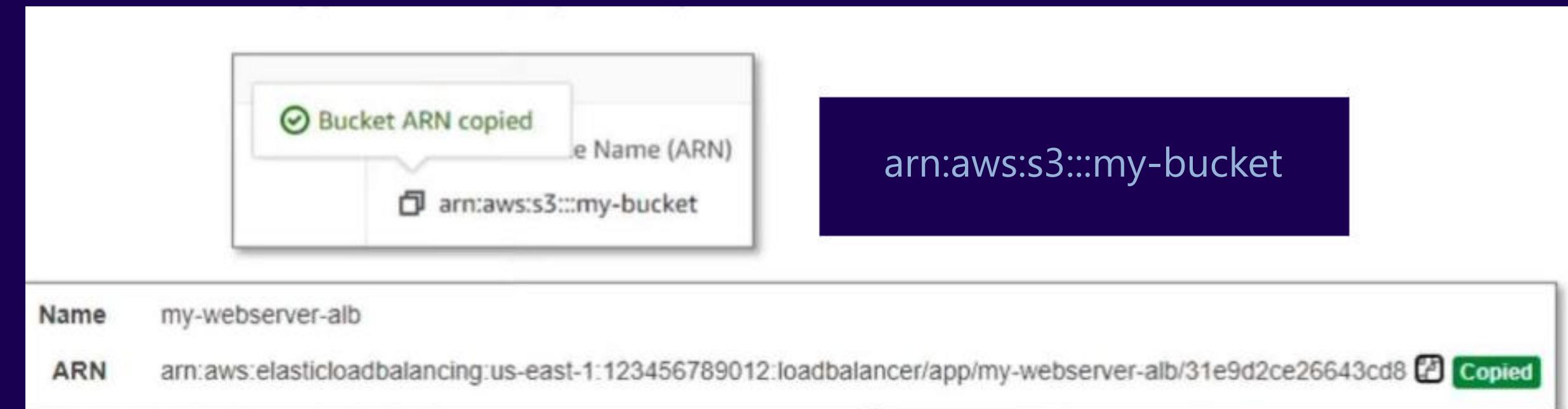
• 23456789012

Resource ID

Could be a number name or path:

- user/Bob
- instance/i-1234567890abcdef0

In the AWS Management Console its common to be able to copy the ARN to your clipboard



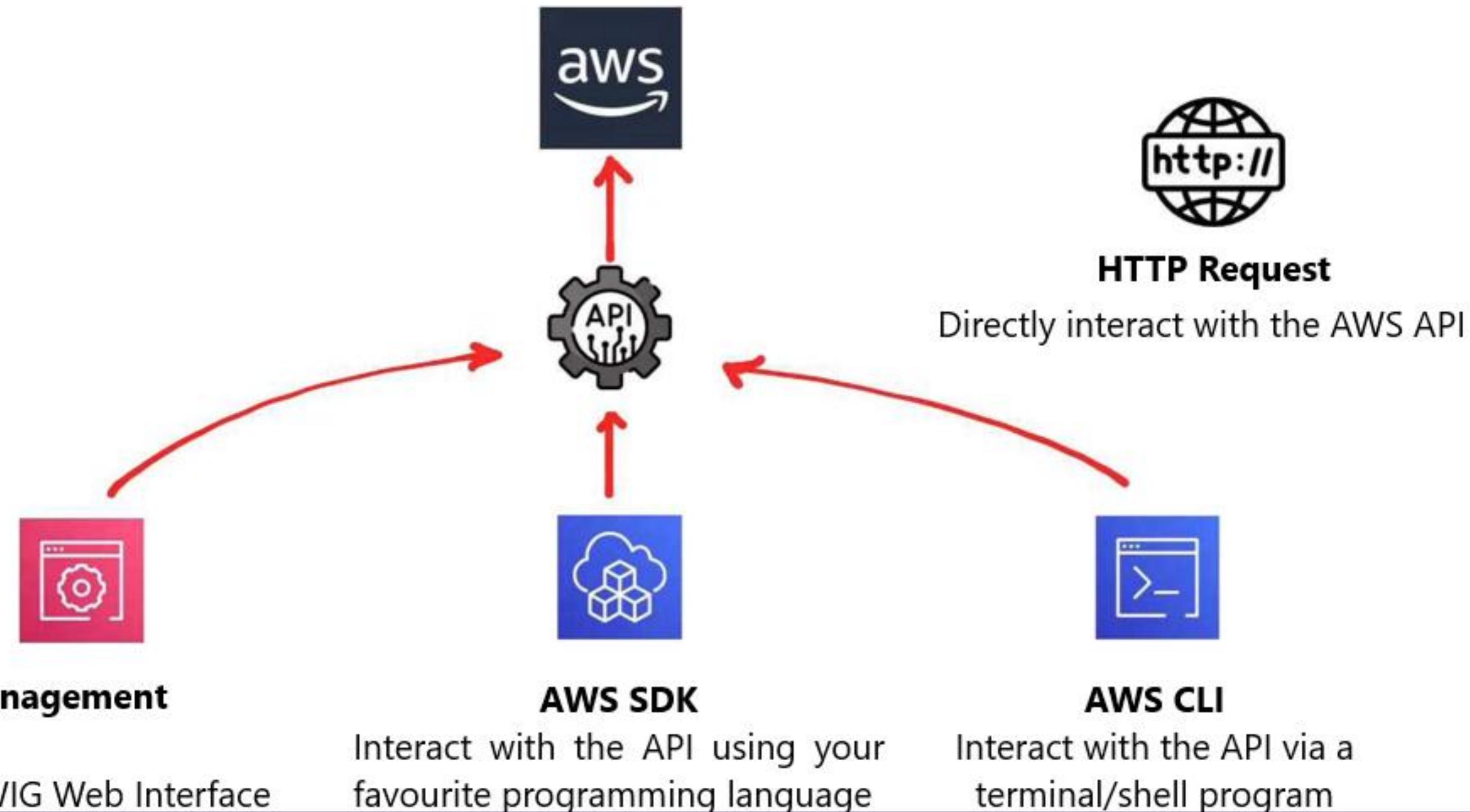
The API Way to interact with AWS

AWS provides a HTTP API. You can call this using Postman or Curl.
Just don't.

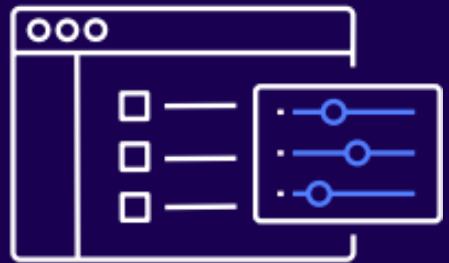
```
GET / HTTP/1.1    1
host: monitoring.us-east-1.amazonaws.com
x-amz-target: GraniteServiceVersion20100801.GetMetricData x-
amz-date: 20180112T092034Z
Authorization: AWS4-HMAC-SHA256
Credential=REDACTEDREDACTED/20180411/
Content-Type: application/json Accept: application/json Content-Encoding: amz-1.0
Content-Length: 45 Connection: keep-alive
```

Refer to the API document on the Payloads that can be passed

Why do this when you can ...?

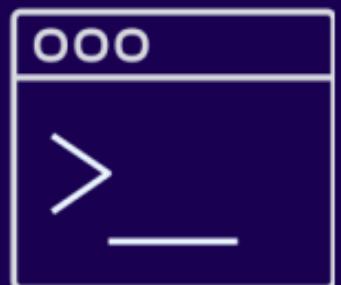


Three ways to interact with AWS



AWS Management Console

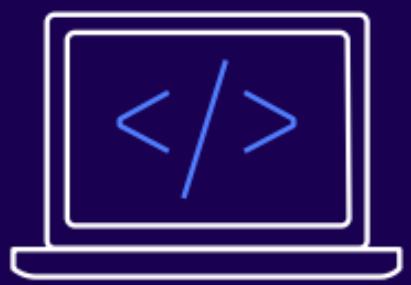
Easy-to-use graphical interface



AWS Command Line Interface (AWS CLI)

Access to services by discrete command

```
$ aws s3 ls
```



Software development kits (SDKs)

Access services in your code



Other ways to interact



AWS Cloud Formation

Declarative IAC Tool
JSON/YAML
Verbose



AWS Cloud Development Kit

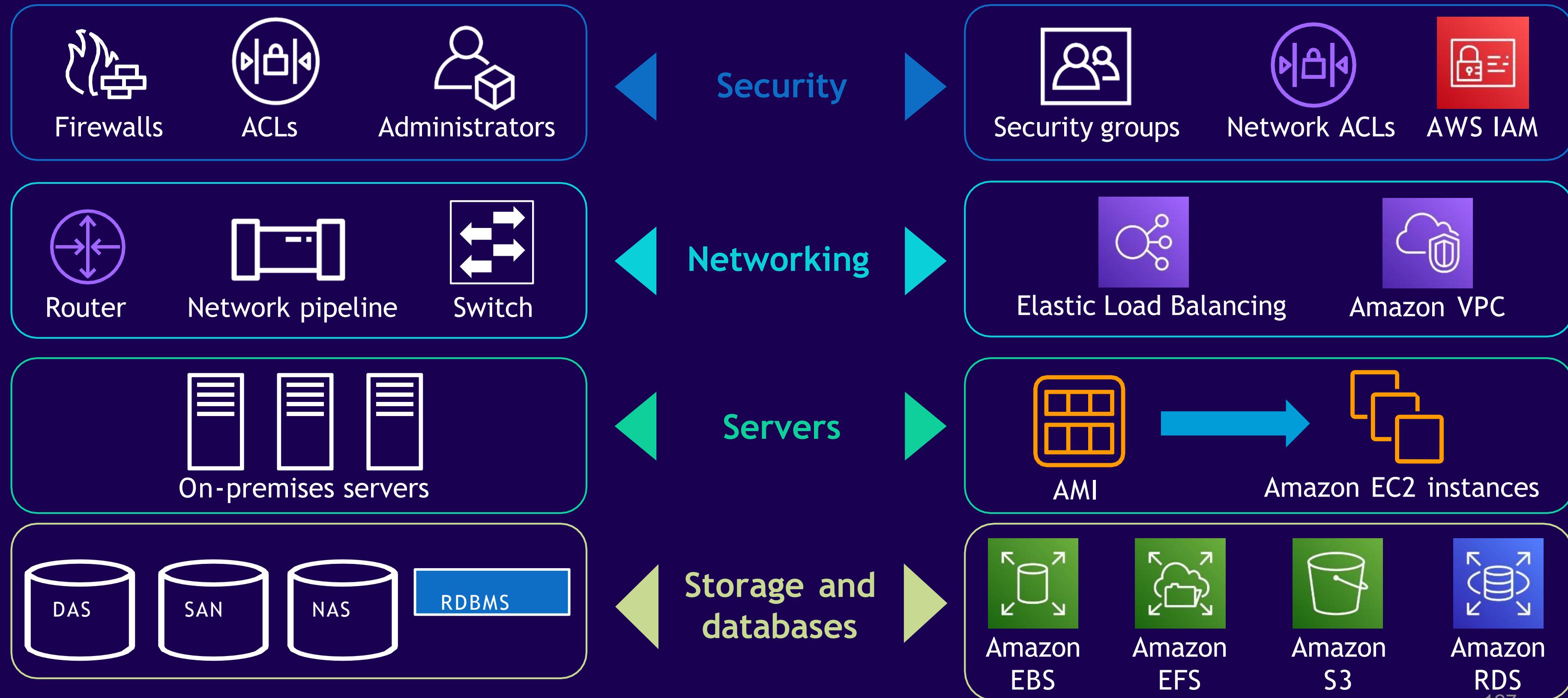
Imperative IAC Tool
Uses Programming Languages like
Python, Ruby, JavaScript



Terraform

Cloud Agnostic
Easy to Pick and Learn

AWS core infrastructure and services - Summary



Compute

Agenda

Introduction to Cloud and AWS

Cloud Computing

Terminologies

AWS Global Infra & services

Compute

Storage

Databases

Networking

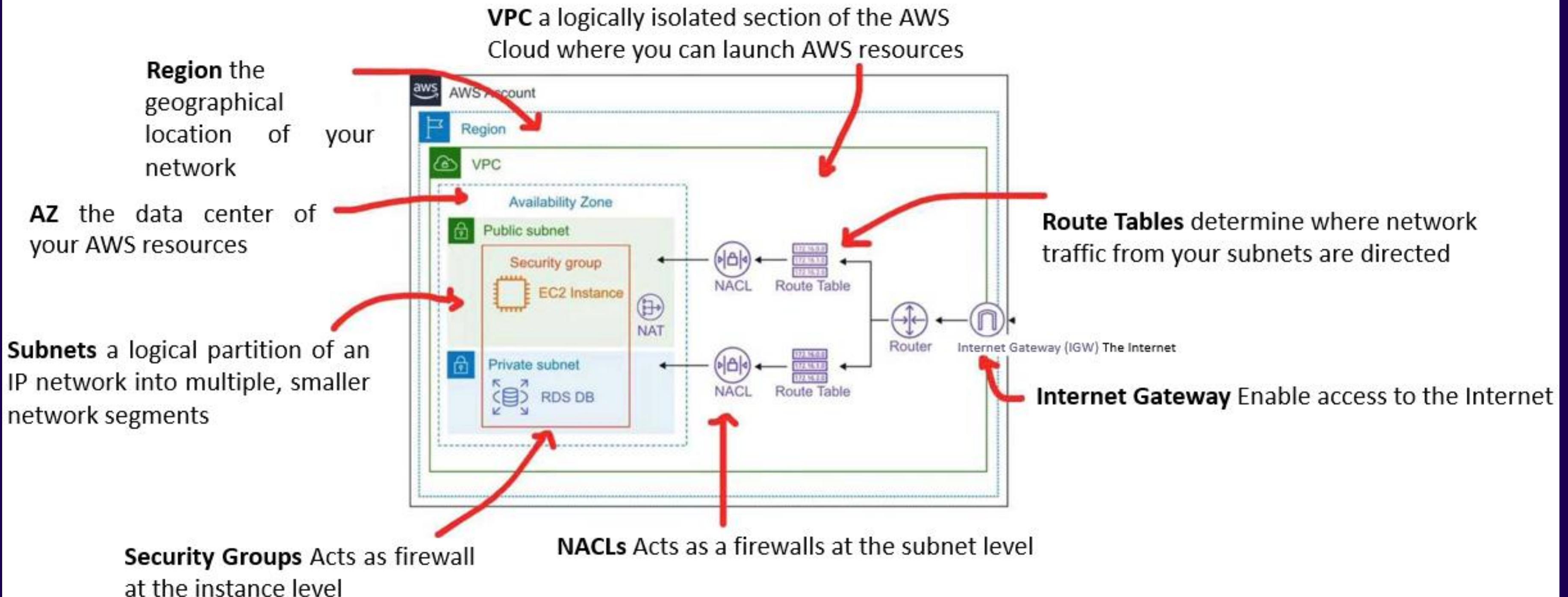
Security

AI and ML with AWS

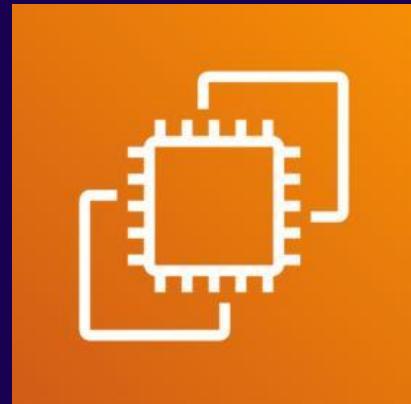
Deploying Apps in AWS

Next steps

Before we start...

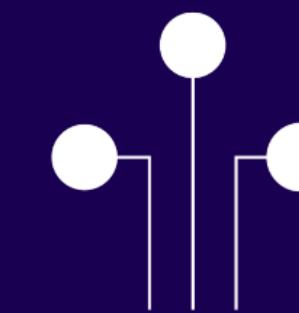


Virtual machines vs. physical servers



Amazon EC2 can solve some problems that are more difficult with an on-premises server

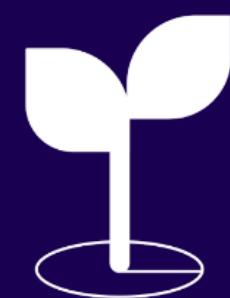
When using disposable resources



Data-driven
decisions

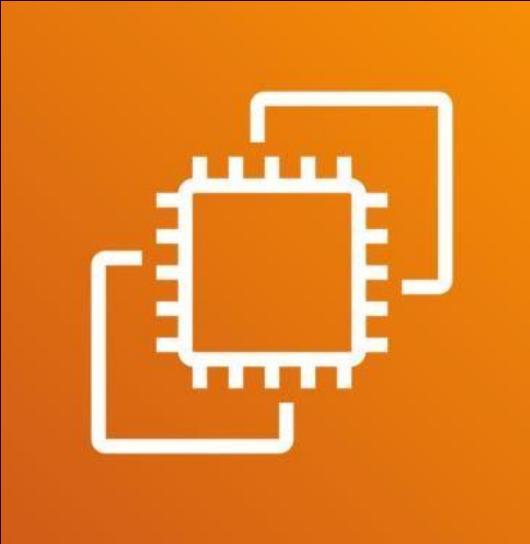


Quick
iterations



Free to make
mistakes

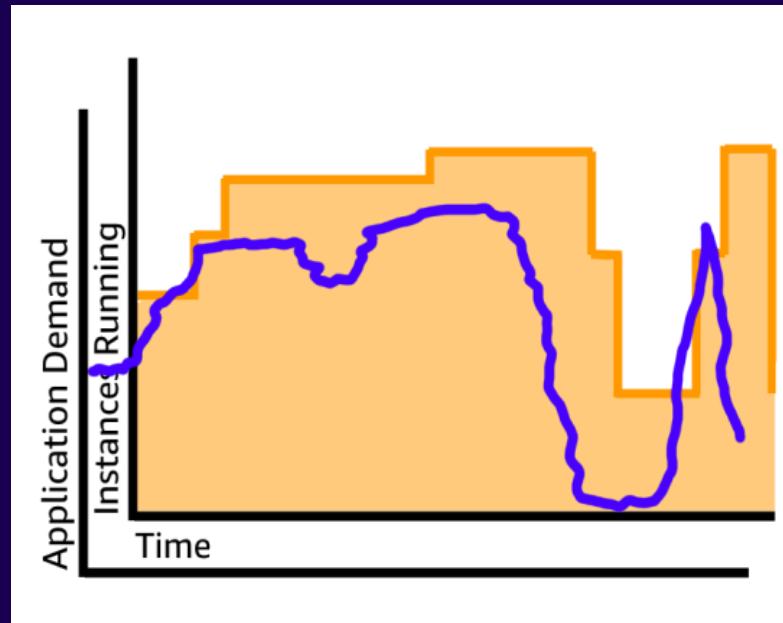
Amazon Elastic Compute Cloud (Amazon EC2)



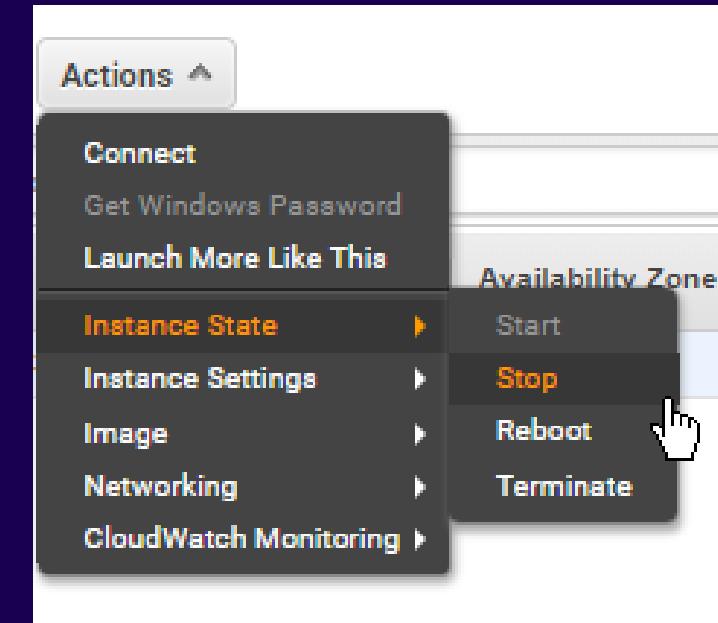
Amazon
EC2

- Resizable compute capacity
- Complete control of your computing resources
- Reduced time required to obtain and boot new server instances

Benefits of Amazon EC2



Elasticity

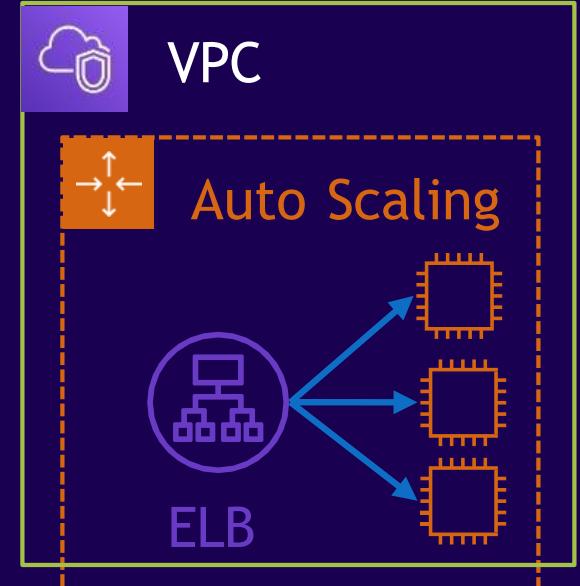


Control

A screenshot of the AWS Lambda Step Functions interface during the 'Step 2: Choose an Instance Type' step. It shows a table of instance types filtered by 'Compute optimized' and 'Current generation'. The table includes columns for Family, Type, vCPUs, and a details icon. Three rows are listed:

Family	Type	vCPUs
Compute optimized	c5d.large	2
Compute optimized	c5d.xlarge	4
Compute optimized	c5d.2xlarge	8

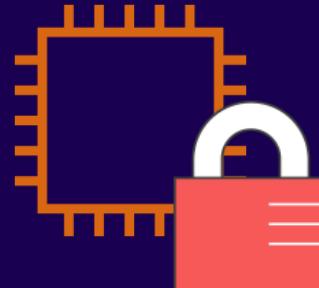
Flexibility



Integrated



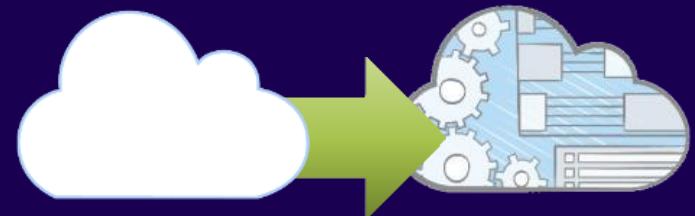
Reliable



Secure



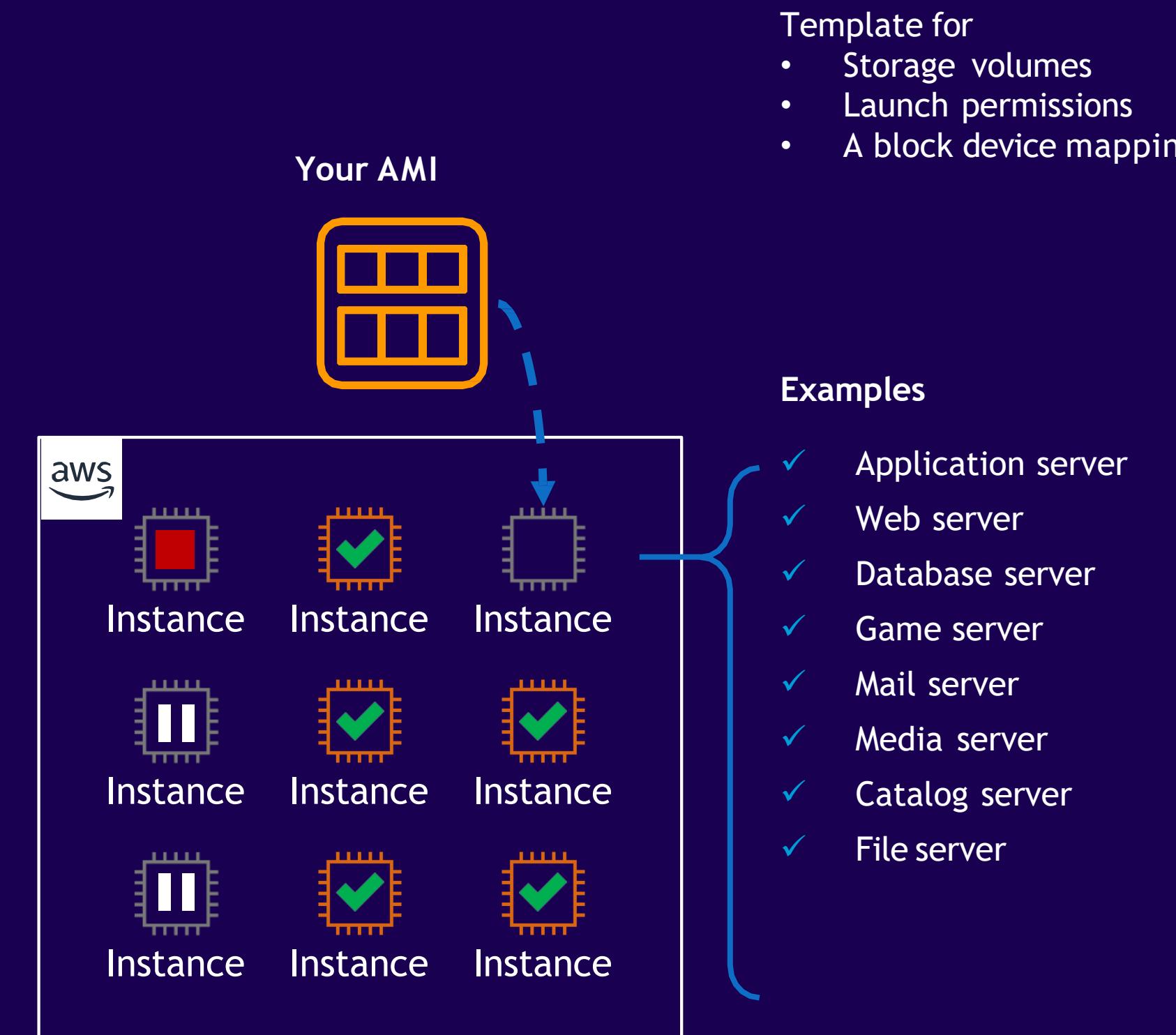
Inexpensive



Simple

Amazon EC2

Amazon EC2 provides pay-as-you-go pricing and a broad selection of hardware and software that's available via the AWS Marketplace by using Amazon Machine Images (AMIs)



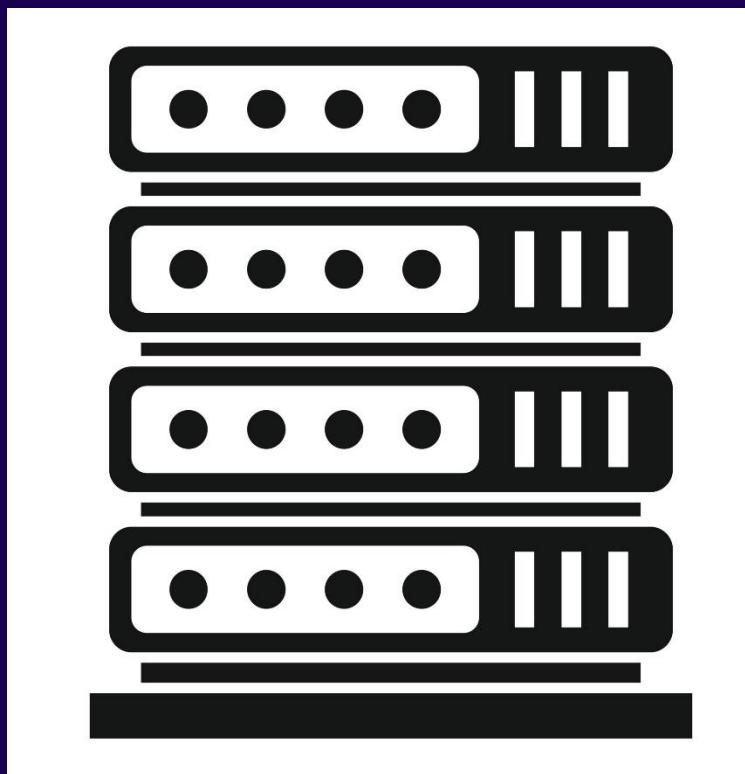
EC2 - Dedicated Host

Dedicated Hosts are single-tenant EC2 instances designed to let you Bring-Your-Own-License (BYOL) based on **machine characteristics**

	Dedicated Instance	Dedicated Hosts
Isolation	Instance Isolation	Physical Server Isolation
Billing	Per instance billing (+\$2 per region fee)	Per host billing
Visibility of Physical characteristics	No Visibilities	Sockets, cores, host ID
Affinity between a host and instance	No Affinity	Consistency deploy to the same instances to the same physical server
Targeted instance placement	No control	Additional control over instance placement on physical server
Automatic instance placement	Yes	Yes
Add capacity using an allocation request	No	Yes

EC2 Tenancy

EC2 has three levels of tenancy:



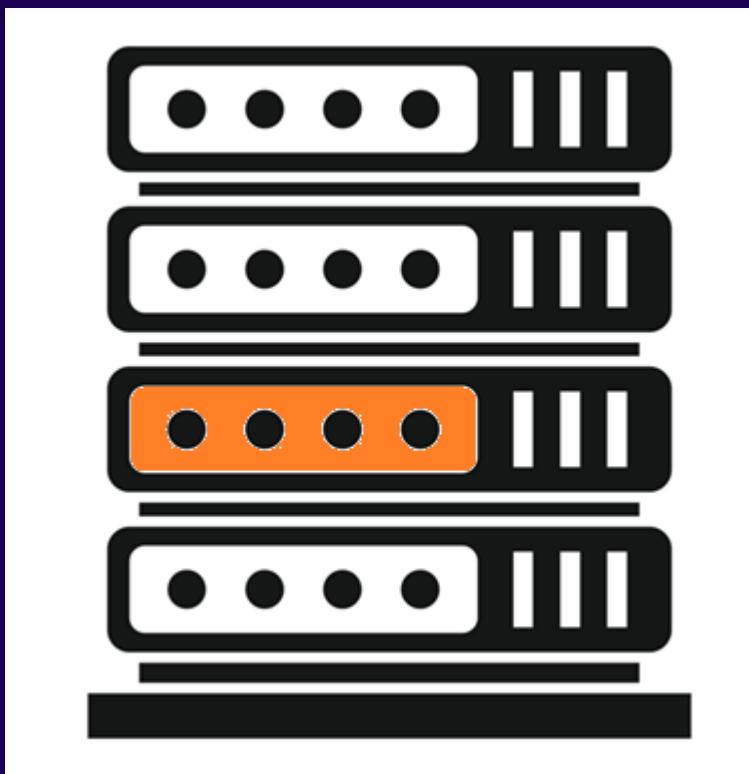
Dedicated Host

Your server lives somewhere here
and you have control of the physical
attributes



Dedicated Instance

Your server always lives here



Default

You instance live here *until reboot*

Amazon EC2 pricing

On-Demand
Instances

Reserved
Instances

Savings
Plans

Spot
Instances

EC2 Pricing Models

There are 5 different ways to pay for EC2 (Virtual Machines)

On-Demand

- low cost and flexible
- only pay per hour or the *second
- short-term, spiky, unpredictable workloads
- cannot be interrupted
- For first time apps

Least Commitment

Spot up to 90%

Biggest Savings

- request spare computing capacity
- flexible start and end times
- Can handle interruptions (server randomly stopping and starting)
- For non-critical background jobs

Reserved up to 75% off

Best Long-term

- steady state or predictable usage
- commit to EC2 over a 1 or 3 year term
- Can resell unused reserved instances

Dedicated

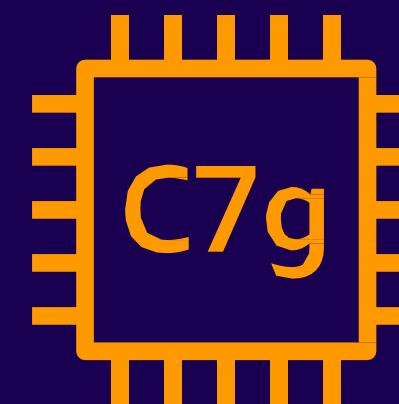
Most Expensive

Dedicated servers
Can be on-demand or reserved or spot When you need a guarantee of isolate hardware (enterprise requirements)

AWS Savings Plan is another way to save but can be used for more than just EC2.

Amazon EC2 instance families and names

Choosing the correct type is very important for
efficient use of your instances and cost reduction



Instance family	Use cases
General purpose e.g., A1, T3, T3a, T2, M6g, M5	<ul style="list-style-type: none">Low-traffic websites and web applicationsSmall databases and midsize databases
Compute optimized e.g., C5, C5n, C4, C7g	<ul style="list-style-type: none">High-performance web serversVideo encoding
Memory optimized e.g., R5, R5n, X1e, X1, z1d	<ul style="list-style-type: none">High-performance databasesDistributed memory caches
Storage optimized e.g., I3, I3en, D2, H1	<ul style="list-style-type: none">Data warehousingLog or data processing applications
Accelerated computing e.g., P3, P2, Inf1, G4, G3, F1	<ul style="list-style-type: none">3D visualizationsMachine learning

EC2 Instance Families

What are Instance Families?

Instance families are different combinations of CPU, Memory, Storage and Networking capacity.

Instance families allow you to choose the appropriate combination of capacity to meet your application's unique requirements.

Different instance families are different because of the varying hardware used to give them their unique properties.

Commonly instance families are called "Instance Types" but an instance type is a combination of size and family

General Purpose

A1 T2_ T3 T3a T4g M4 M5 M5a M5n M6zn M6g M6i Mac

balance of compute, memory and networking resources *Use-cases web servers and code repositories*

Compute Optimized

C5 C4 Cba C5n C6g C6gn

Ideal for compute bound applications that benefit from high performance processor *Use-cases scientific modeling, dedicated gaming servers and ad server engines*

Memory Optimized

R4 R5 R5a R5b R5n XI Xle High Memory zld

fast performance for workloads that process large data sets in memory.

Use-cases in-memory caches, in-memory databases, real time big data analytics

Accelerated Optimized

P2 P3 P4 G3 G4ad G4dn F1 Infl VT1

hardware accelerators, or co-processors

Use-cases Machine learning, computational finance, seismic analysis, speech recognition

Storage Optimized

I3 I3en D2 D3 D3en HI

high, sequential read and write access to very large data sets on local storage *Use-cases NoSQL, in-memory or transactional databases, data warehousing*

EC2 Instance Types

An instance type is a particular **instance size** and **instance family**:

A common pattern for instance sizes:

- nano
- micro
- small
- medium
- large
- xlarge
- 2xlarge
- 4xlarge
- 8xlarge

	Family	Type	vCPUs	Memory (GiB)
	t2	t2.nano	1	0.5
■	t2	t2.micro <small>Free tier eligible</small>	1	1
	t2	t2.small	1	2
	t2	t2.medium	2	4
	t2	t2.large	2	8
	t2	t2.xlarge	4	16

There are many exceptions to this pattern for sizes e.g.

- c6g.metal - is a bare metal machine.
- C5.9xlarge - Is not a power of 2 or even number size

Unmanaged services compared to managed services



Unmanaged

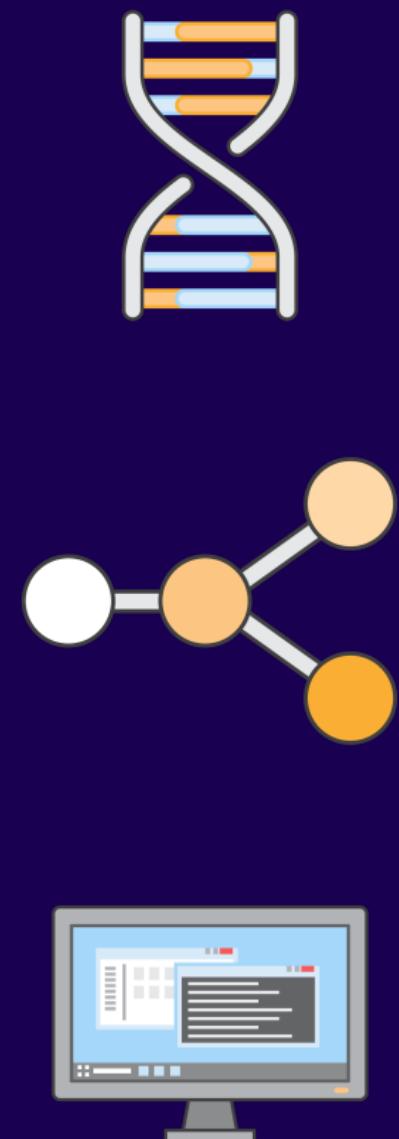
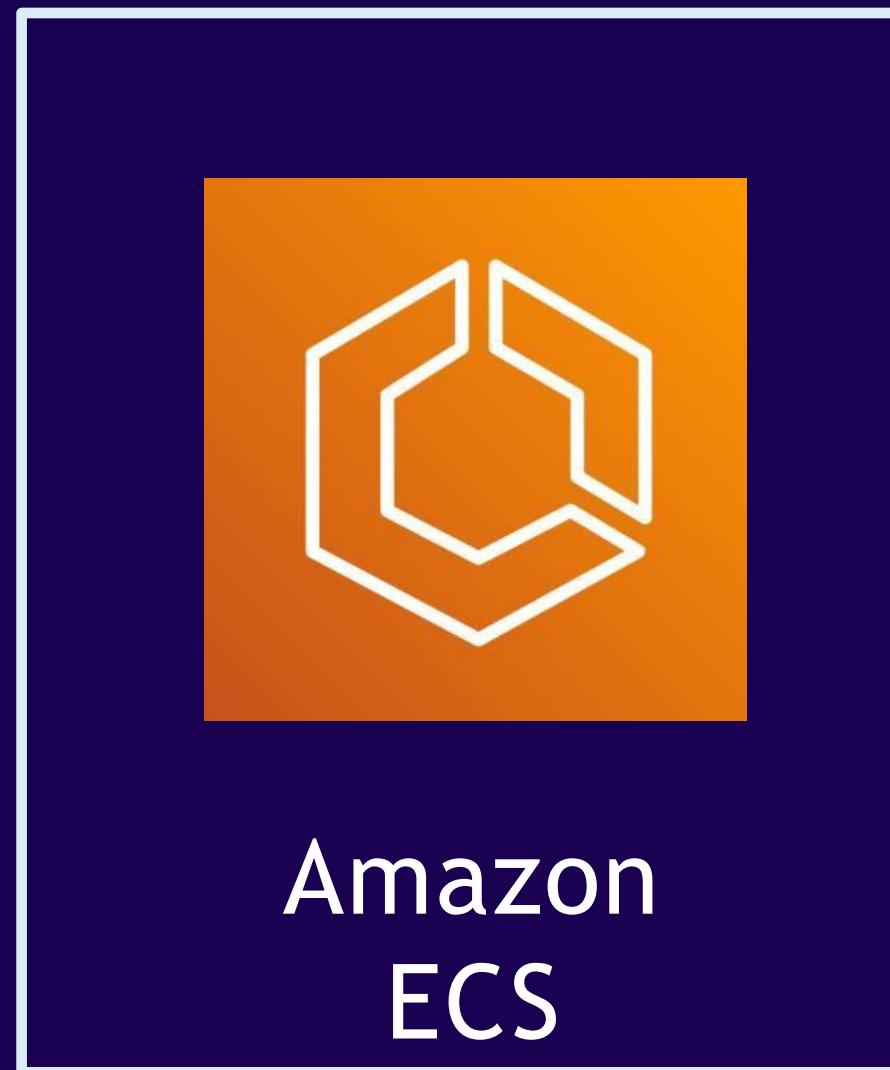
You manage scaling, fault tolerance, and availability



Managed

Scaling, fault tolerance, and availability are typically built in to the service

Amazon Elastic Container Service (Amazon ECS)



Orchestrates the execution of containers

Maintains and scales the fleet of nodes running your containers

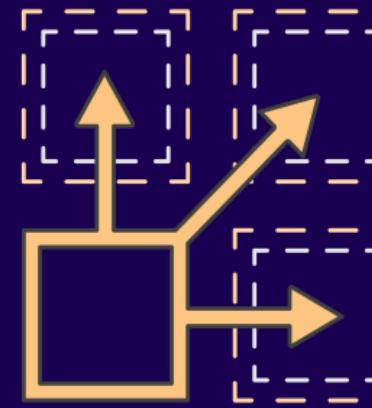
Removes the complexity of standing up the infrastructure

EC2

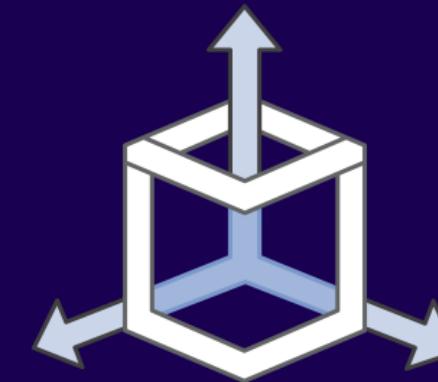
Create an EC2 instance
It should be based on Unix
Should have a Web serve installed and running

What is serverless computing?

Building and running applications and services without managing servers



No servers to provision or manage



Scales with usage



Never pay for idle



Availability and fault tolerance built in

Serverless means:

Greater agility

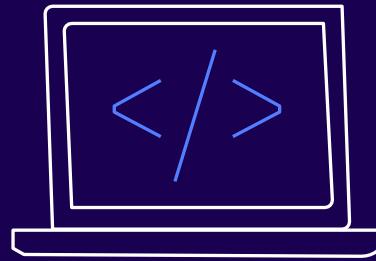
Less operations

More product focus

Faster time to market

Cost that grows with your business

Serverless application use cases



Web applications

Static websites

Complex web applications

Packages for Flask and Express

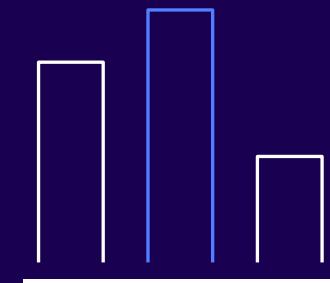


Backends

Applications and services

Mobile

IoT



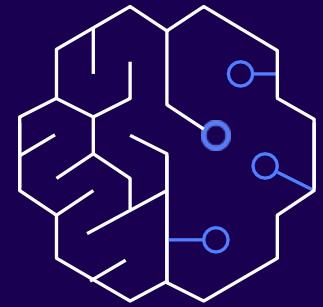
Data processing

Real time

MapReduce

Batch

Machine learning inference



Chatbots

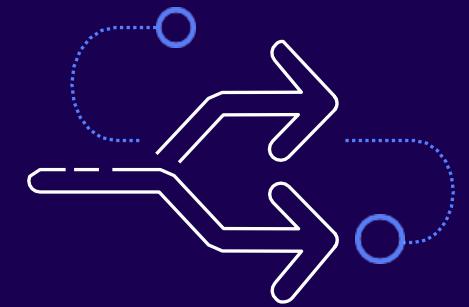
Powering chatbot logic



Amazon Alexa

Powering voice-enabled applications

Alexa Skills Kit



IT automation

Policy engines

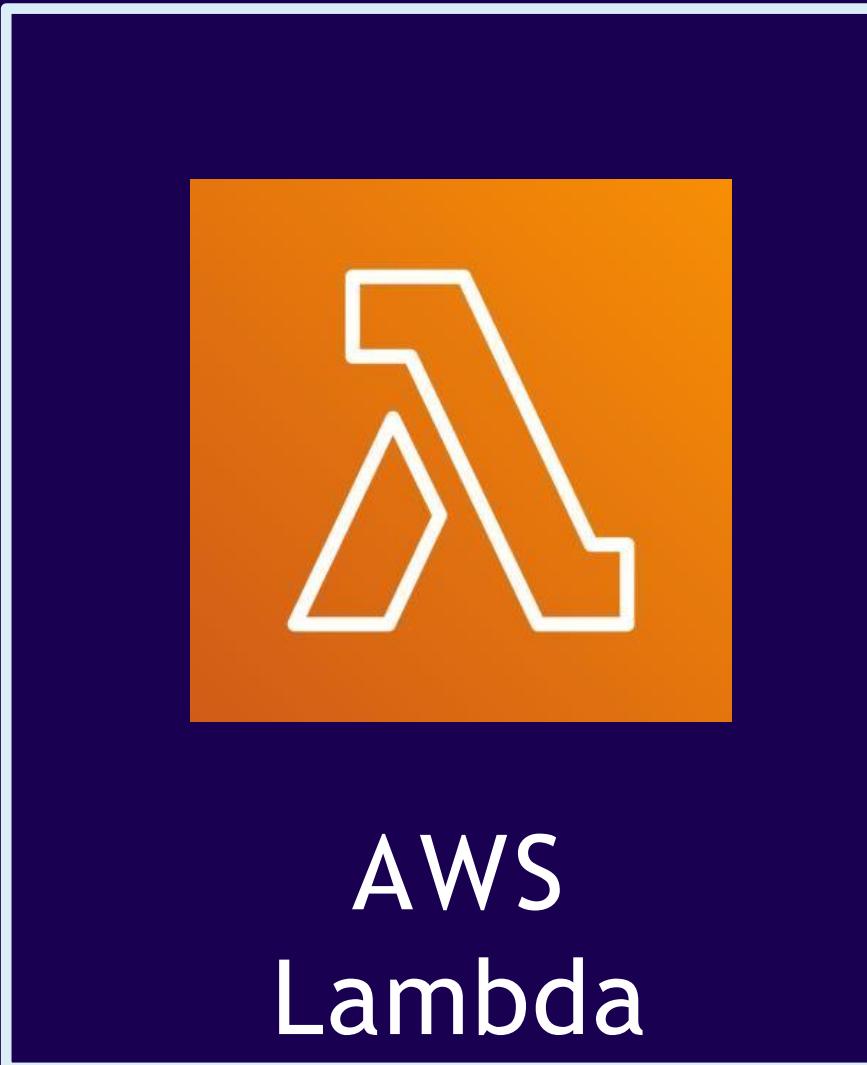
Extending AWS services

Infrastructure management

AWS Lambda

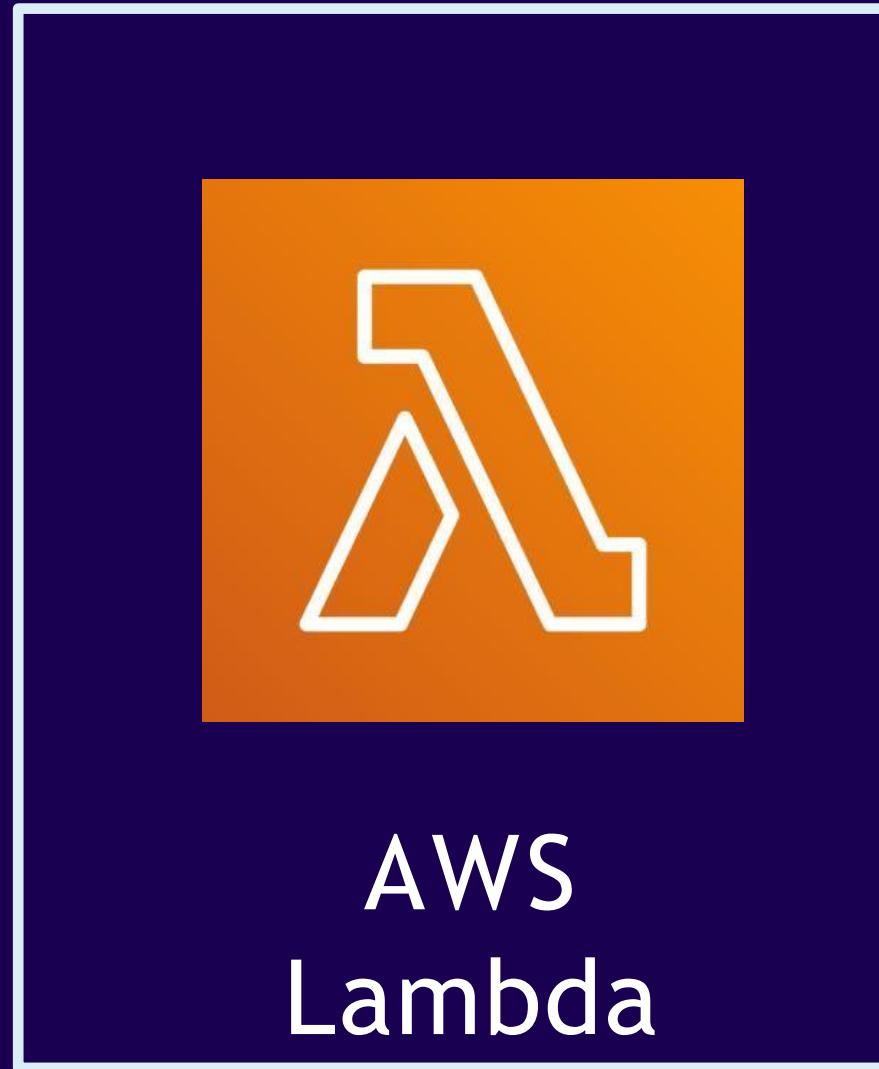


AWS Lambda



- Fully managed compute service
 - Runs stateless code
 - Supports multiple languages
 - Runs your code on a schedule or in response to events (for example, changes to data in an Amazon S3 bucket or Amazon DynamoDB table)

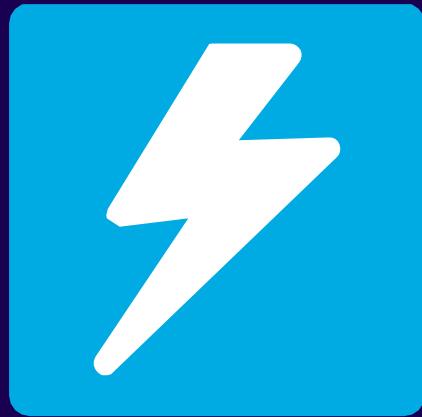
AWS Lambda



- Load Balancing
- Auto Scaling
- Handling Failures
- Security Isolation
- OS Management
- Utilization Management
- And many other things....

Serverless applications

Event source



Function



Changes in
data state



Node.js
Python
Java

Requests to
endpoints



C#
Go
Ruby

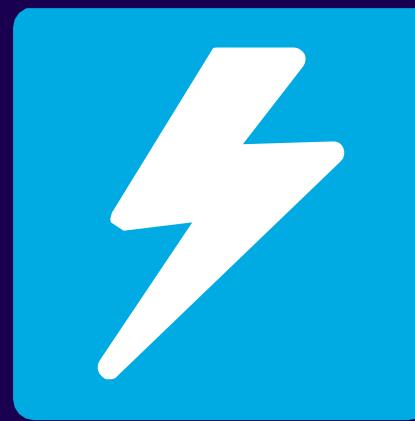
Changes in
Resource state



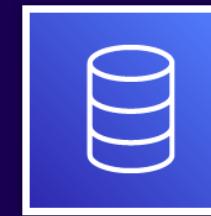
Runtime API

Serverless applications

Event source



Changes in
data state



Requests to
endpoints



Changes in
Resource state

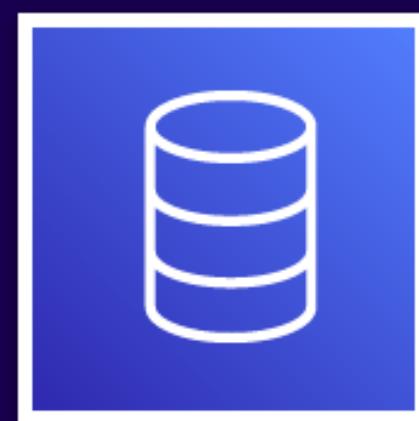


Function



Node.js
Python
Java
C#
Go
Ruby
Runtime API

Services



Anatomy of a Lambda function

Handler() function

Function to be executed upon invocation

Event object

Data sent during Lambda function Invocation

Context object

Methods available to interact with runtime information (request ID, log group, more)

```
import json

def lambda_handler(event, context):
    # TODO implement
    return {
        'statusCode': 200,
        'body': json.dumps('Hello World!')
    }
```

```
Import sdk
Import http-lib
Import ham-sandwich

Pre-handler-secret-getter()
Pre-handler-db-connect()

Function myhandler(event, context) {
    <Event handling logic> {
        result = SubfunctionA()
    }else {
        result = SubfunctionB()
    }
    return result;
}
```

Your handler

```
Function Pre-handler-secret-getter() {
```

```
}
```

```
Function Pre-handler-db-connect(){
}
```

```
Function subFunctionA(thing){
    ## logic here
}
```

```
Function subFunctionB(thing){
    ## logic here
}
```

```
Import sdk  
Import http-lib  
Import ham-sandwich
```

Dependencies, configuration information, common helper functions

```
Pre-handler-secret-getter()  
Pre-handler-db-connect()
```

```
Function myhandler(event, context) {  
    <Event handling logic> {  
        result = SubfunctionA()  
    }else {  
        result = SubfunctionB()  
  
    return result;  
}
```

Your handler

```
Function Pre-handler-secret-getter() {  
}
```

```
Function Pre-handler-db-connect(){  
}
```

```
Function subFunctionA(thing){  
    ## logic here  
}
```

```
Function subFunctionB(thing){  
    ## logic here  
}
```

```
Import sdk  
Import http-lib  
Import ham-sandwich
```

Dependencies, configuration information, common helper functions

```
Pre-handler-secret-getter()  
Pre-handler-db-connect()
```

```
Function myhandler(event, context) {  
    <Event handling logic> {  
        result = SubfunctionA()  
    }else {  
        result = SubfunctionB()  
  
    return result;  
}
```

Your handler

```
Function Pre-handler-secret-getter() {  
}
```

```
Function Pre-handler-db-connect(){  
}
```

```
Function subFunctionA(thing){  
    ## logic here  
}
```

```
Function subFunctionB(thing){  
    ## logic here  
}
```

Common helper functions

```
Import sdk  
Import http-lib  
Import ham-sandwich
```

Dependencies, configuration information, common helper functions

```
Pre-handler-secret-getter()  
Pre-handler-db-connect()
```

```
Function myhandler(event, context) {  
    <Event handling logic> {  
        result = SubfunctionA()  
    }else {  
        result = SubfunctionB()  
  
    return result;  
}
```

Your handler

```
Function Pre-handler-secret-getter() {  
}
```

```
Function Pre-handler-db-connect(){  
}
```

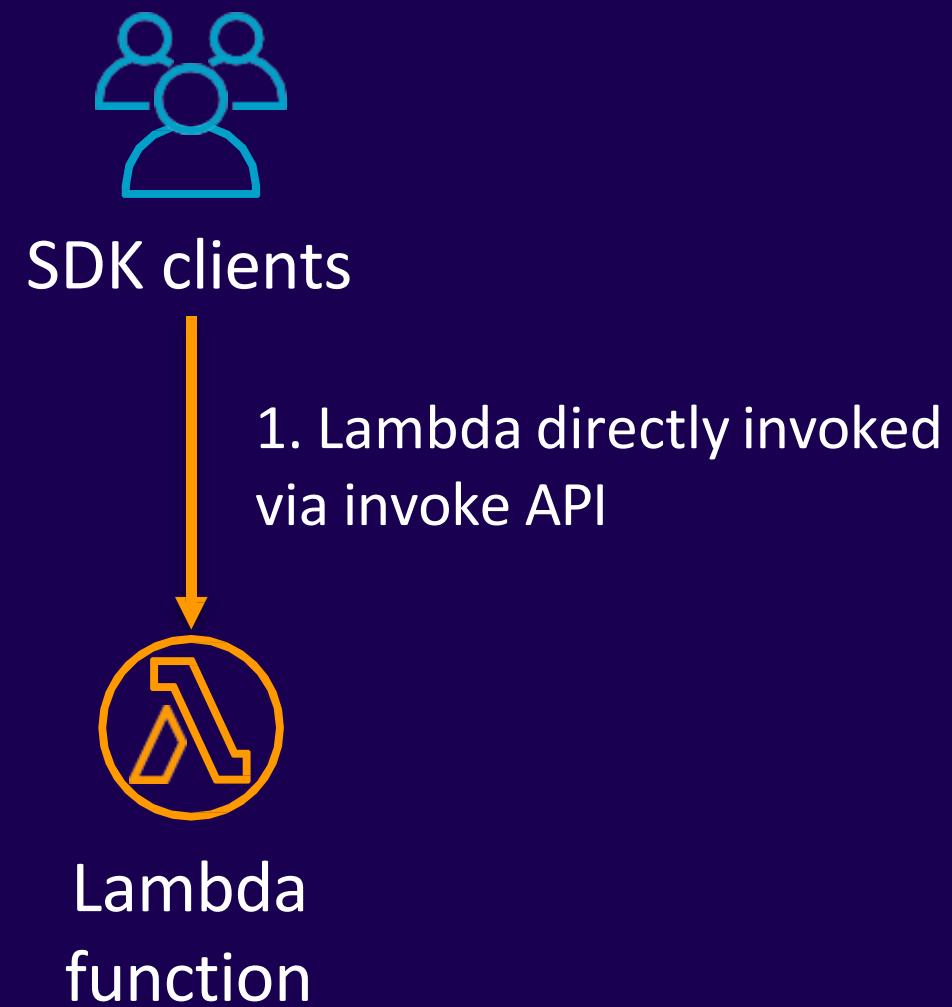
```
Function subFunctionA(thing){  
    ## logic here  
}
```

```
Function subFunctionB(thing){  
    ## logic here  
}
```

Common helper functions

Business logic sub-functions

Lambda API

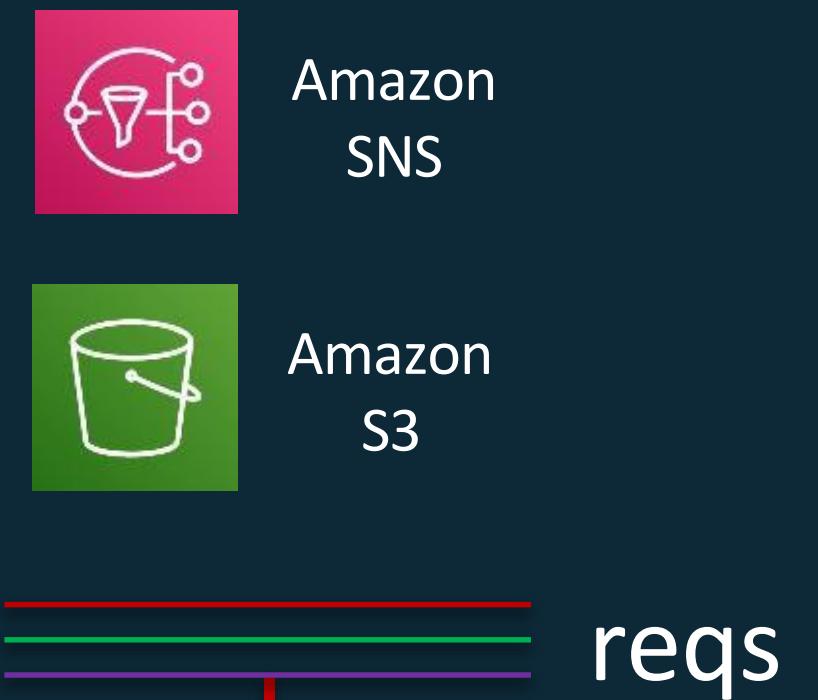


Lambda execution model

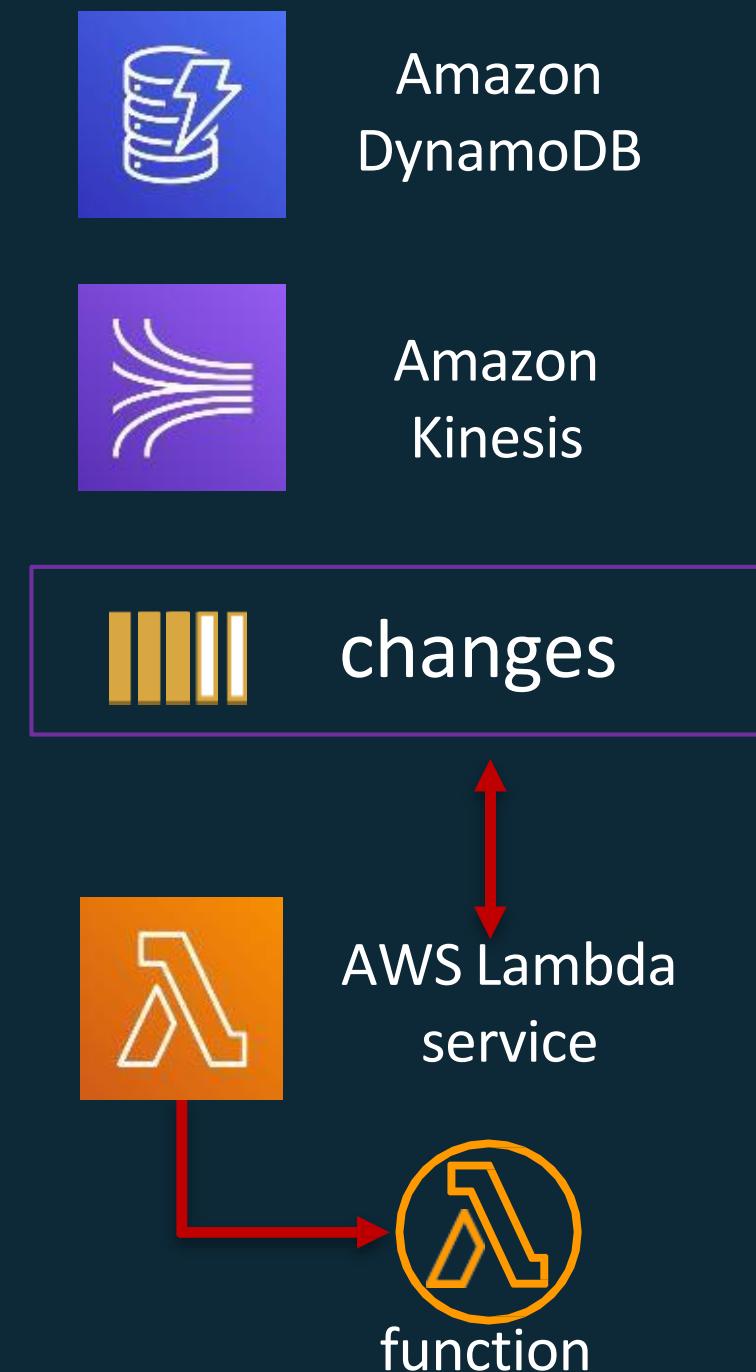
Synchronous
(push)



Asynchronous
(event)



Stream
(Poll-based)



Lambda permissions model

Function policies:

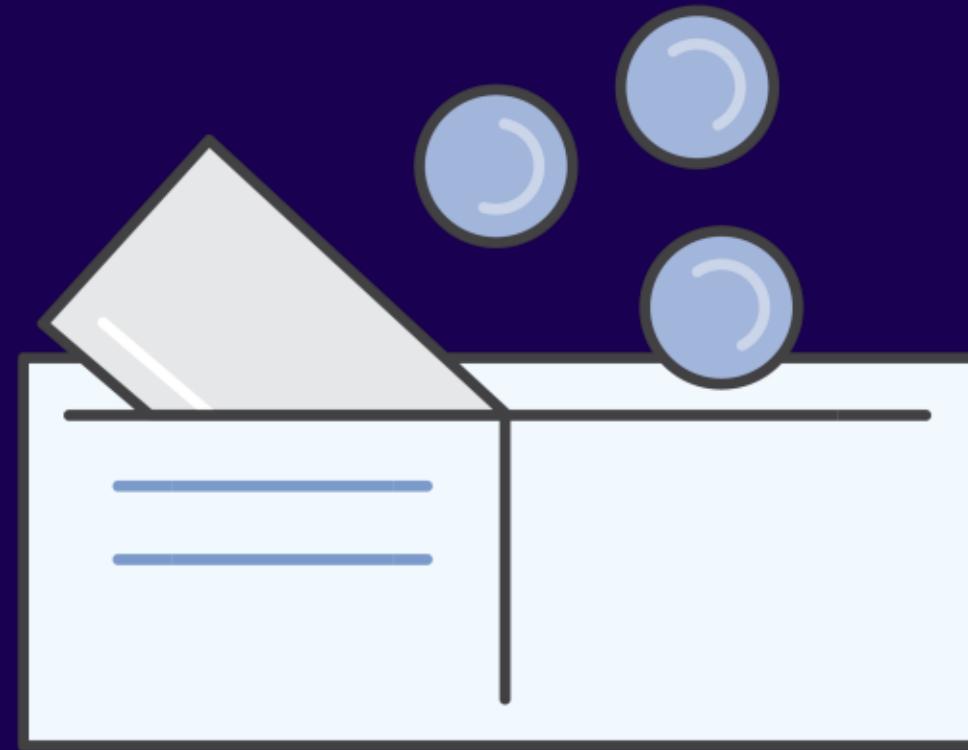
- “Actions on bucket X can invoke Lambda function Z”
- Resource policies allow for cross account access
- Used for sync and async invocations

Execution role:

- “Lambda function A can read from DynamoDB table users”
- Define what AWS resources/API calls can this function access via IAM
- Used in streaming invocations



Fine-grained pricing



Free Tier

1M requests and 400,000 GBs of compute.

Every month, every customer.

Buy compute time in 100ms increments

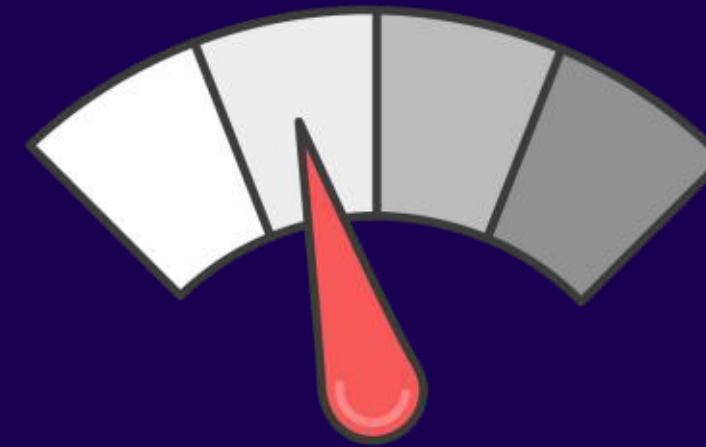
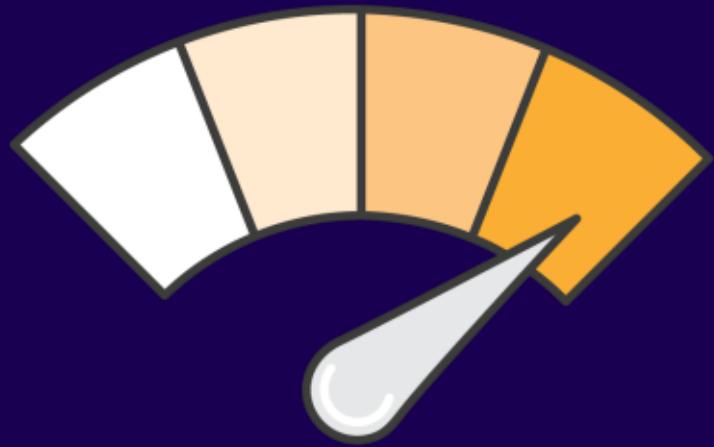
Low request charge

No hourly, daily, or monthly minimums

No per-device fees

Never pay for idle

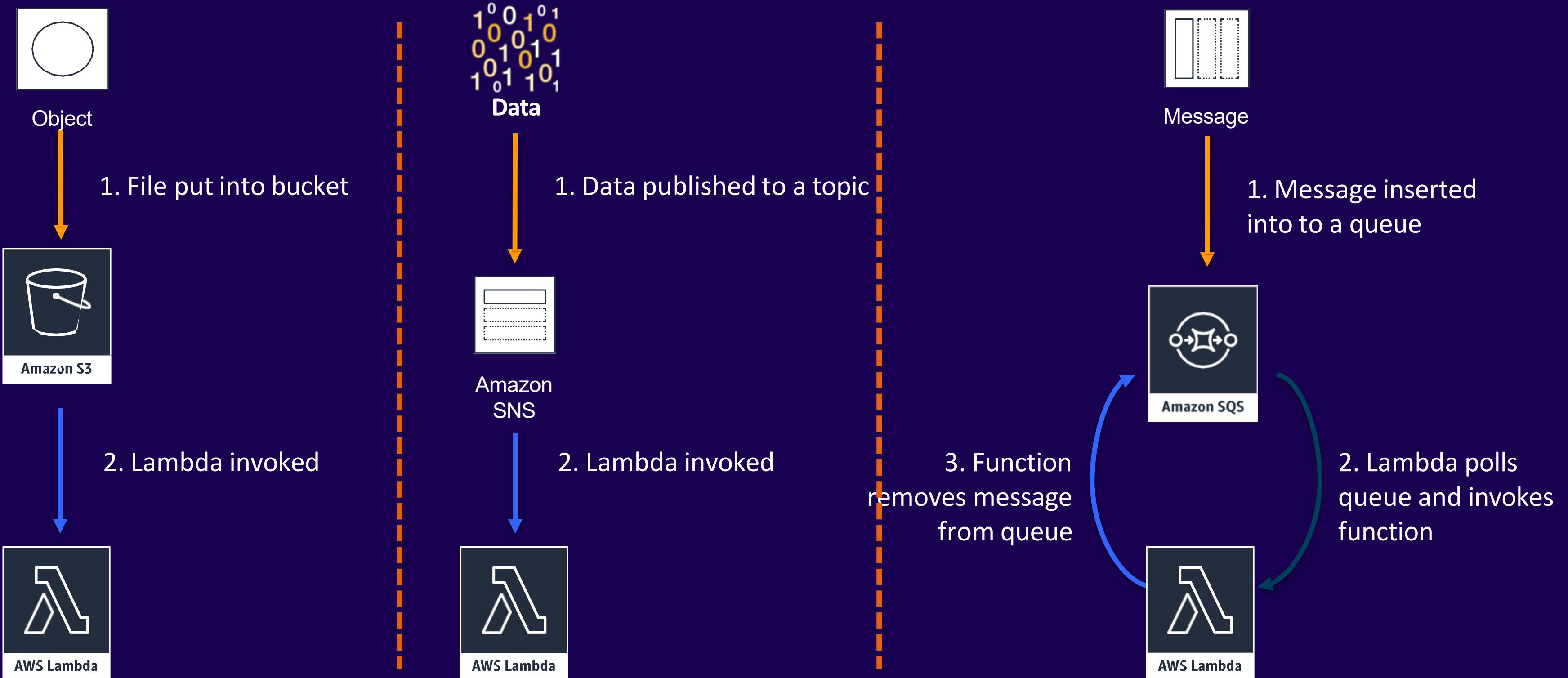
Tweak your function's computer power



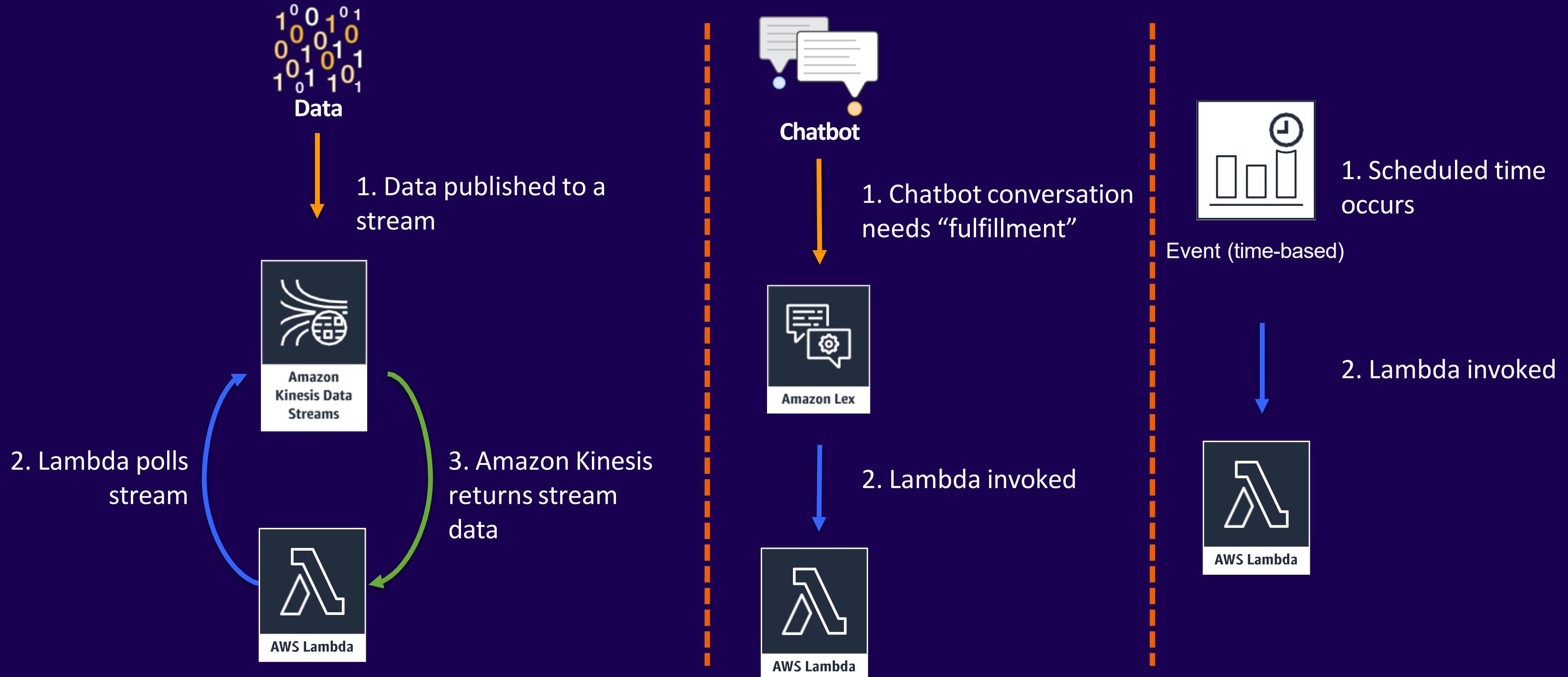
Lambda exposes only a memory control, with the **% of CPU core and network capacity** allocated to a function proportionally
Is your code CPU, Network or memory-bound? If so, it could be **cheaper** to choose more memory.

DEMO

Serverless architectures

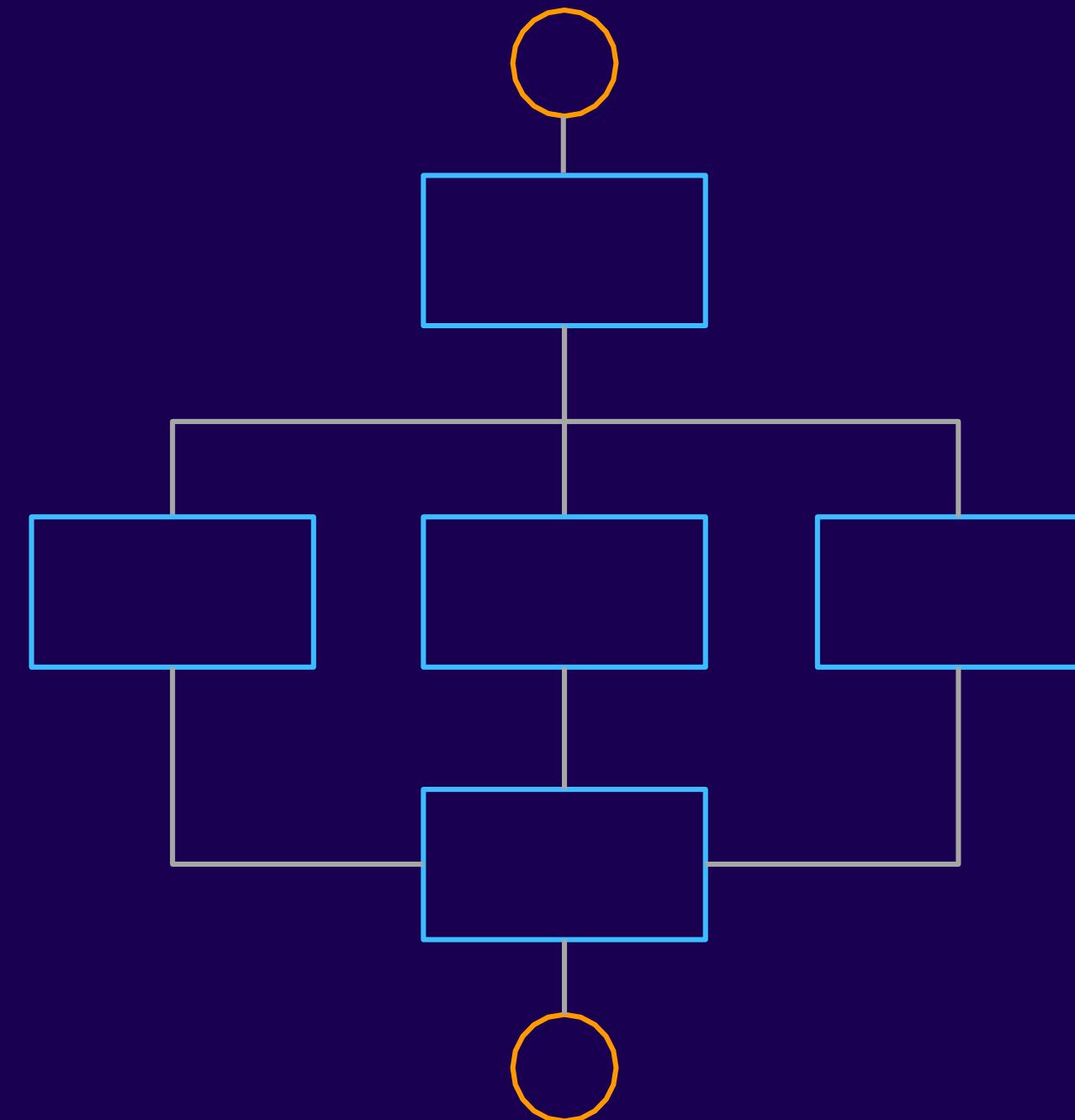


Serverless architectures



Keep orchestration out of code

Track status of data
and execution

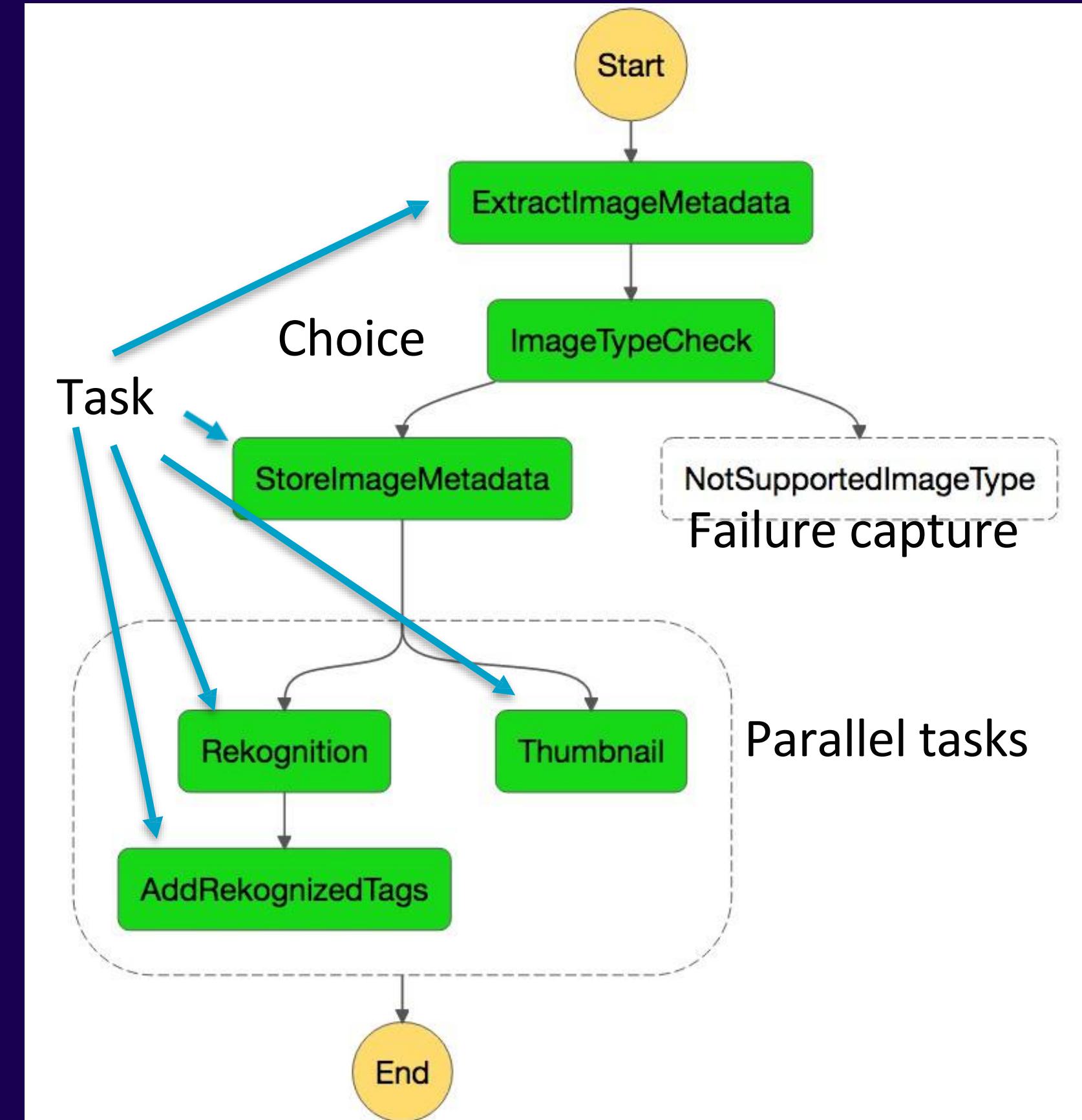


Remove redundant
code

AWS Step Functions

Serverless workflow management with zero administration

- Makes it easy to coordinate the components of distributed applications and microservices using visual workflows
- Automatically triggers and tracks each step and retries when there are errors, so your application executes in order and as expected
- Logs the state of each step, so when things do go wrong, you can diagnose and debug problems quickly



What are AWS Step Functions?

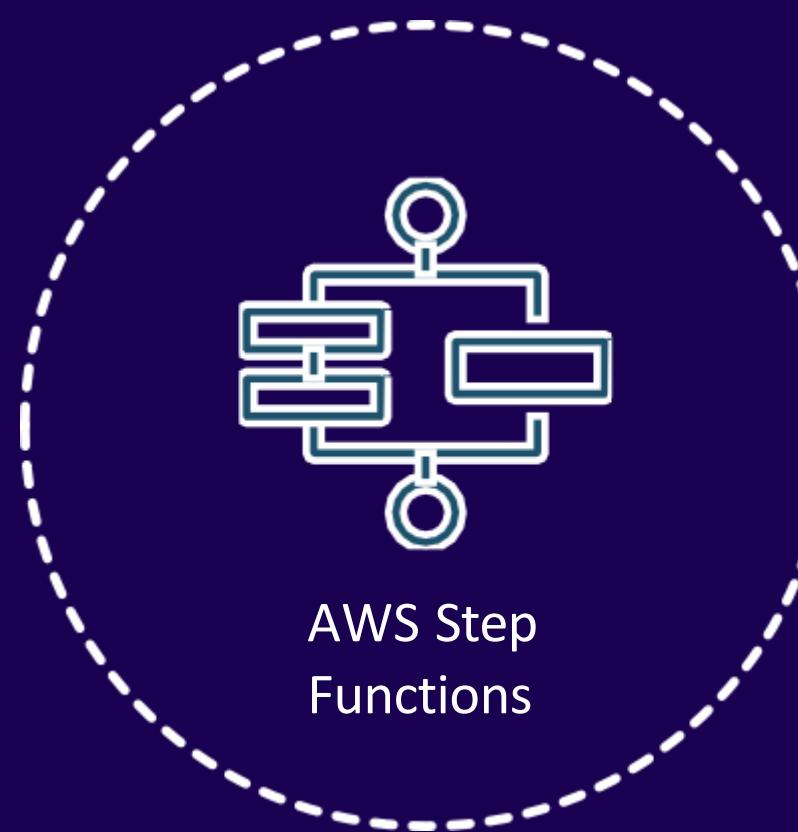


AWS Step Functions

Serverless workflows that help you:

- Build and update apps quickly
- Improve resiliency
- Write less code
- Orchestrate long-running tasks
- Modernize monoliths
- Integrate with managed services
- Handle errors and retries

Step Functions: Integrations



Simplify building workloads such as order processing, report generation, and data analysis

Write and maintain less code; add services in minutes

More service integrations:



Amazon Simple
Notification
Service



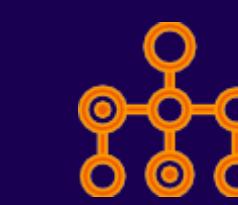
Amazon Simple
Queue Service



Amazon
SageMaker



AWS Glue



AWS Batch



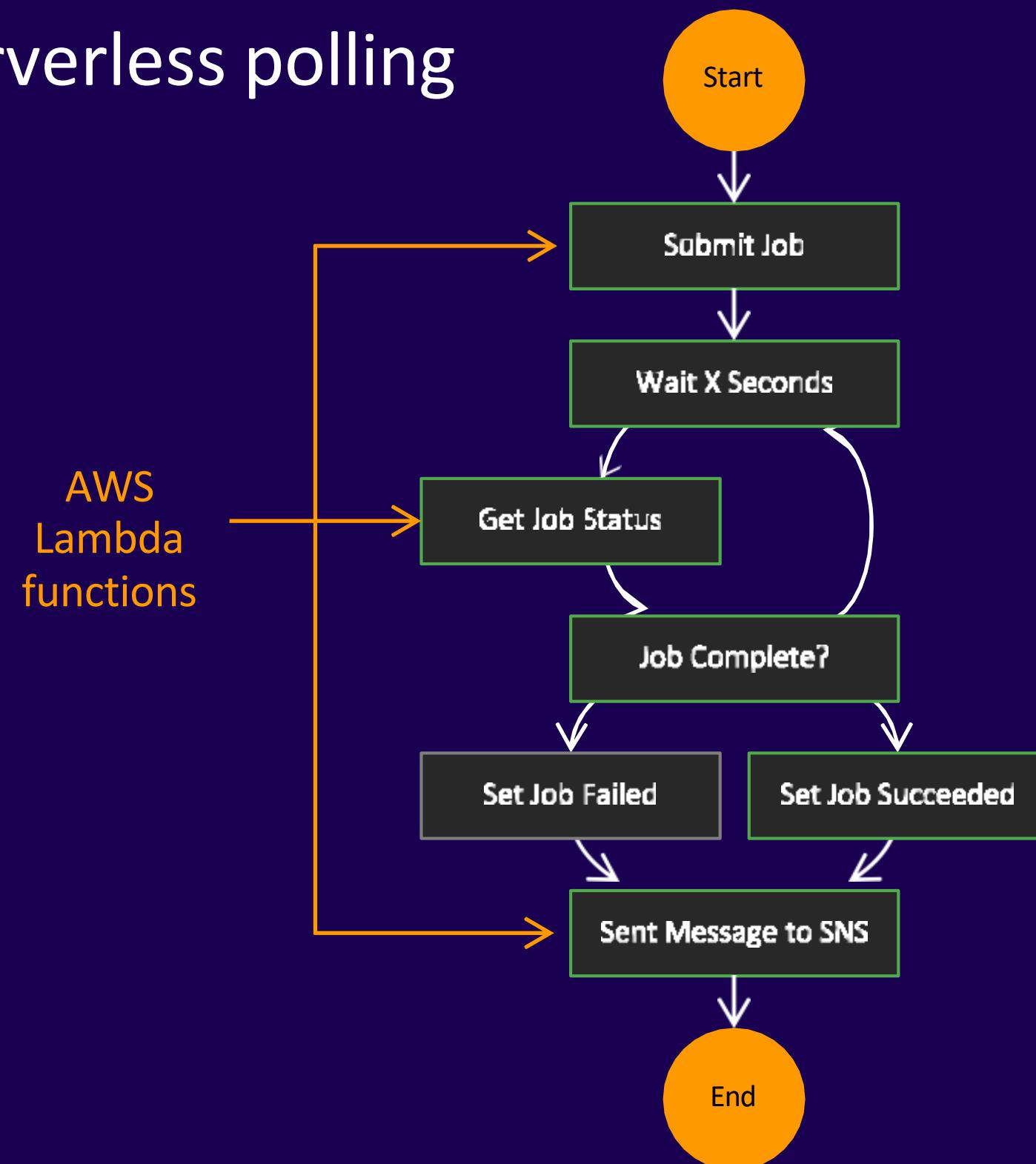
Amazon Elastic
Container Service



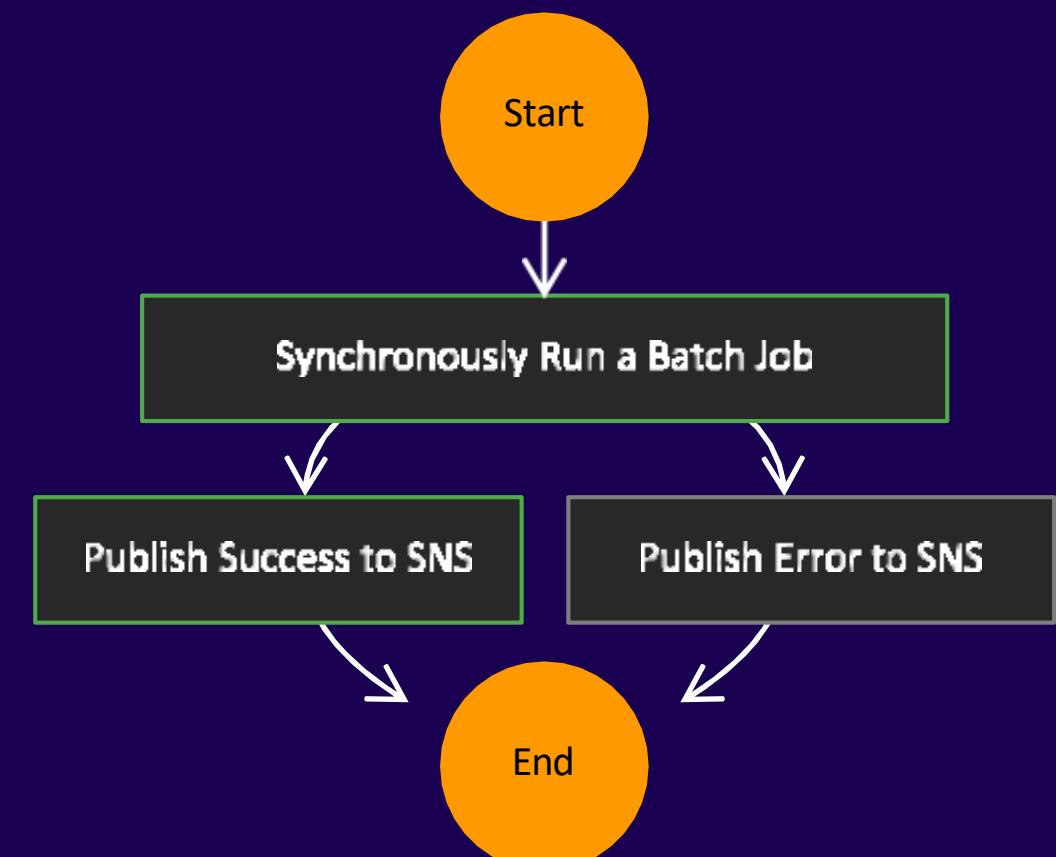
AWS Fargate

Simpler integration, less code

With serverless polling



With direct service integration

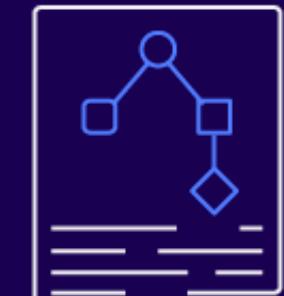


Introduction to Step Functions



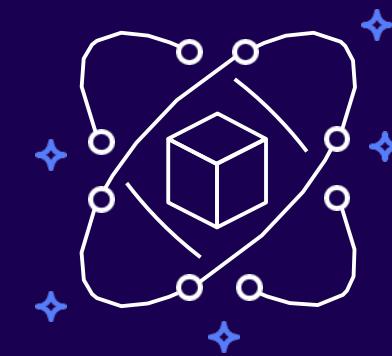
Tasks

All work in your state machine is done by tasks



State Machines

States are elements in your state machine



Integrations

You can directly call other AWS services



Error Handling

Retries and fallbacks are available to you



Monitoring

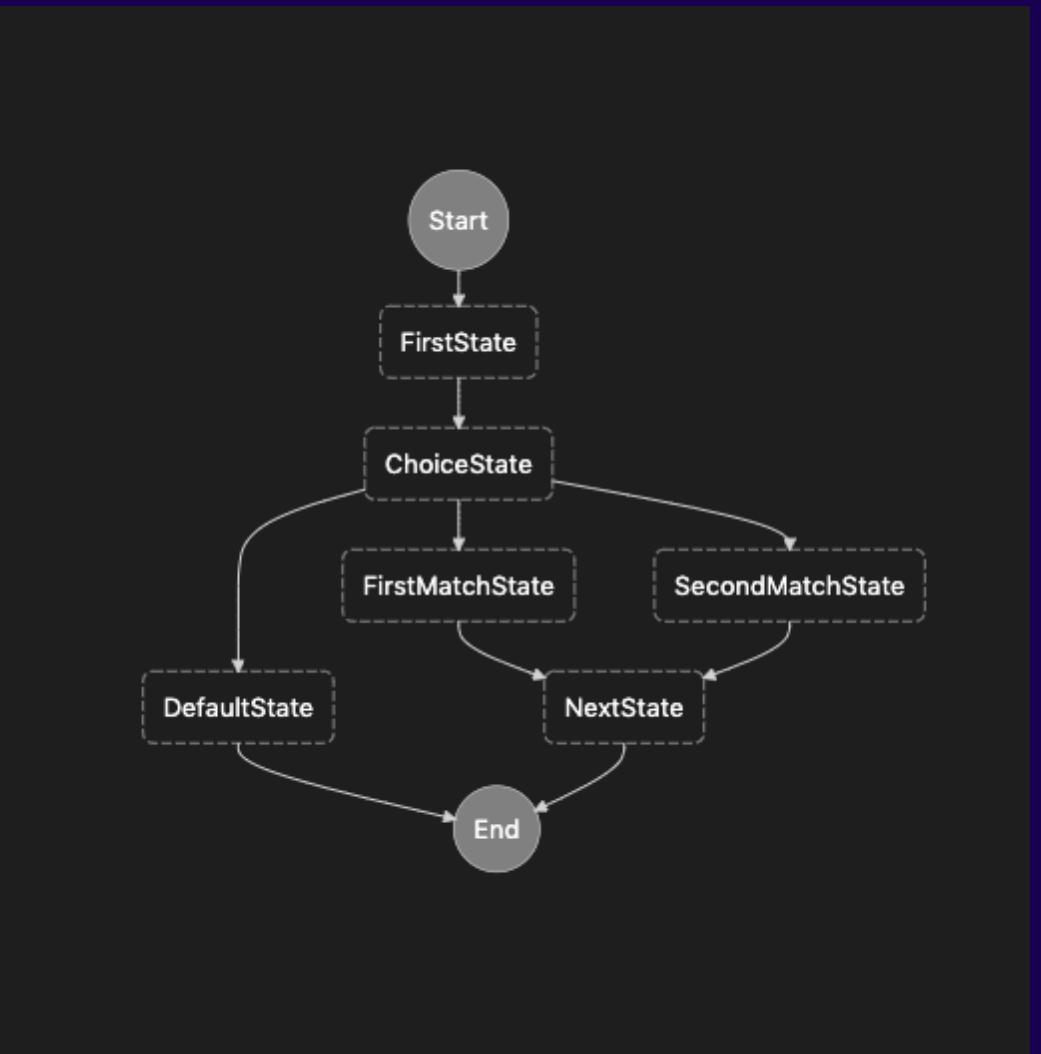
Use Amazon CloudWatch to monitor your workflows

Serverless workflows

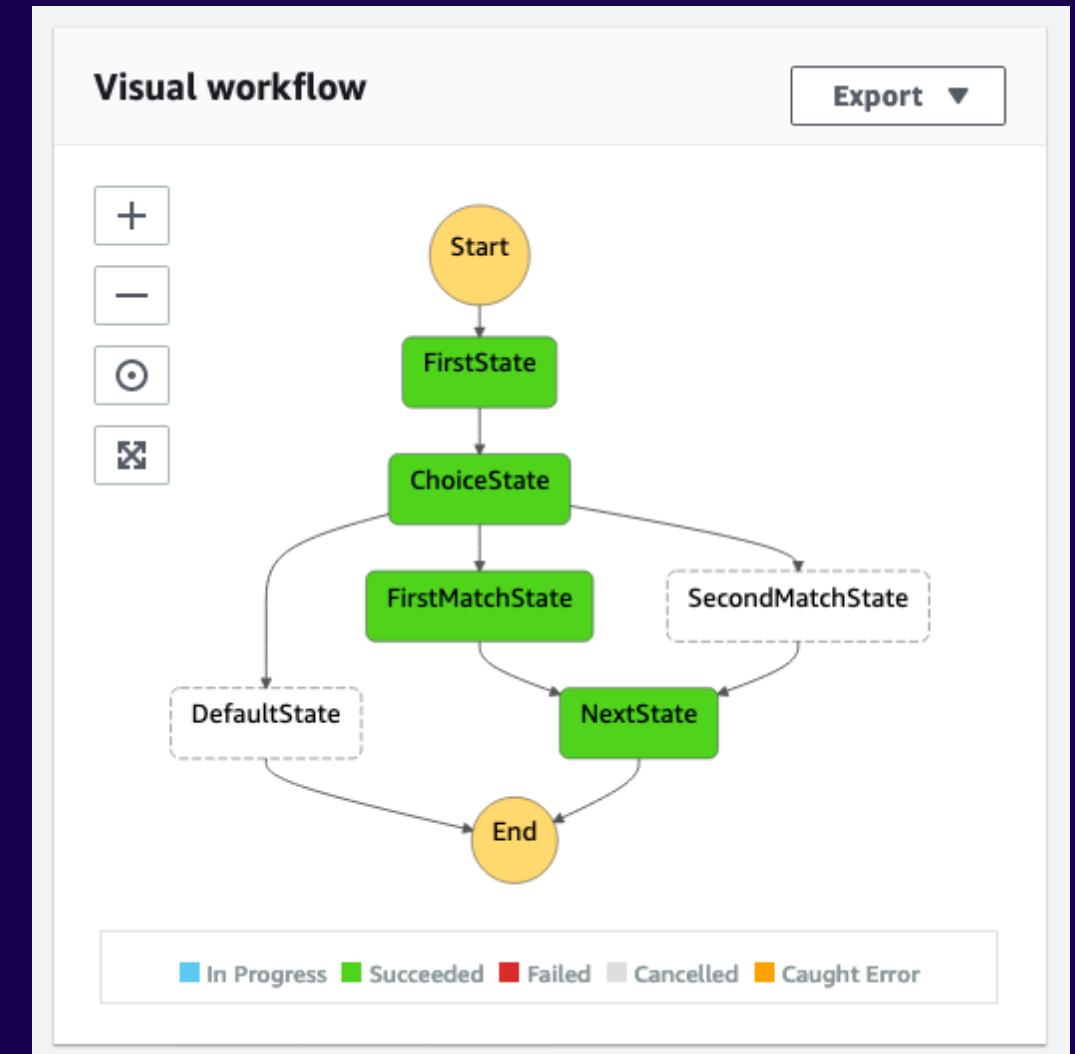
Define

```
{  
    "Comment": "An example of the Amazon States Language using a choice state.",  
    "StartAt": "FirstState",  
    "States": {  
        "FirstState": {  
            "Type": "Pass",  
            "Parameters": {  
                "foo": 1  
            },  
            "Next": "ChoiceState"  
        },  
        "ChoiceState": {  
            "Type": "Choice",  
            "Choices": [  
                {  
                    "Variable": "$.foo",  
                    "NumericEquals": 1,  
                    "Next": "FirstMatchState"  
                },  
                {  
                    "Variable": "$.foo",  
                    "NumericEquals": 2,  
                    "Next": "SecondMatchState"  
                }  
            ],  
            "Default": "DefaultState"  
        },  
        "FirstMatchState": {  
            "Type": "Pass",  
            "Next": "NextState"  
        },  
        "SecondMatchState": {  
            "Type": "Pass",  
            "Next": "NextState"  
        },  
        "DefaultState": {  
            "Type": "Fail",  
            "Error": "DefaultStateError",  
            "Cause": "No Matches!"  
        },  
        "NextState": {  
            "Type": "Pass",  
            "End": true  
        }  
    }  
}
```

Visualize



Monitor



Nested workflows

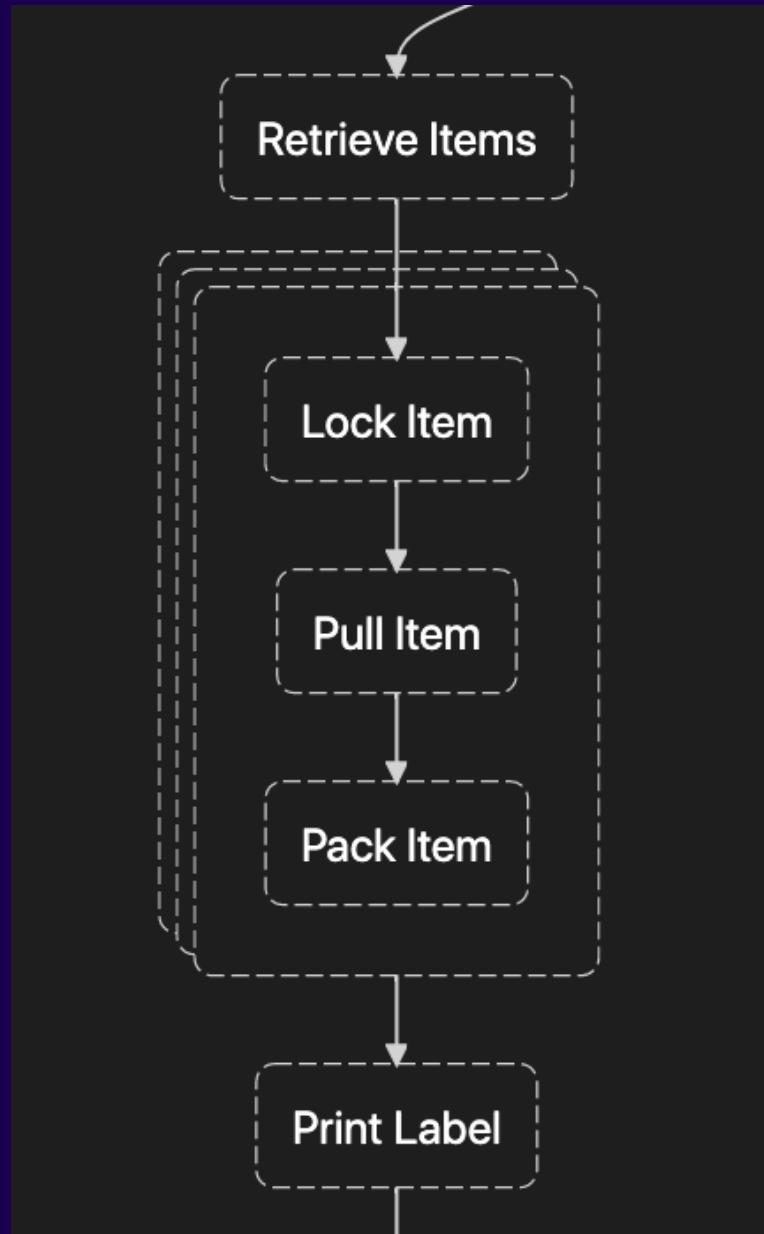
- Build larger, more complex workflows out of smaller, simpler workflows
- Create reusable building blocks and build faster
- Swap workflow modules without customizing code
- Return a string or a JSON object



```
{  
  "StartAt": "Nested Workflow",  
  "States": {  
    "Nested Workflow": {  
      "Type": "Task",  
      "Resource": "arn:aws:states:::states:startExecution.sync:2",  
      "Parameters": {  
        "StateMachineArn": "${NestedWorkflowArn}",  
        "Name": "NestedWorkflow",  
        "Input": {  
          "SomeVariable.$": "$.SomeValue"  
        }  
      },  
      "End": true  
    }  
  }  
}
```

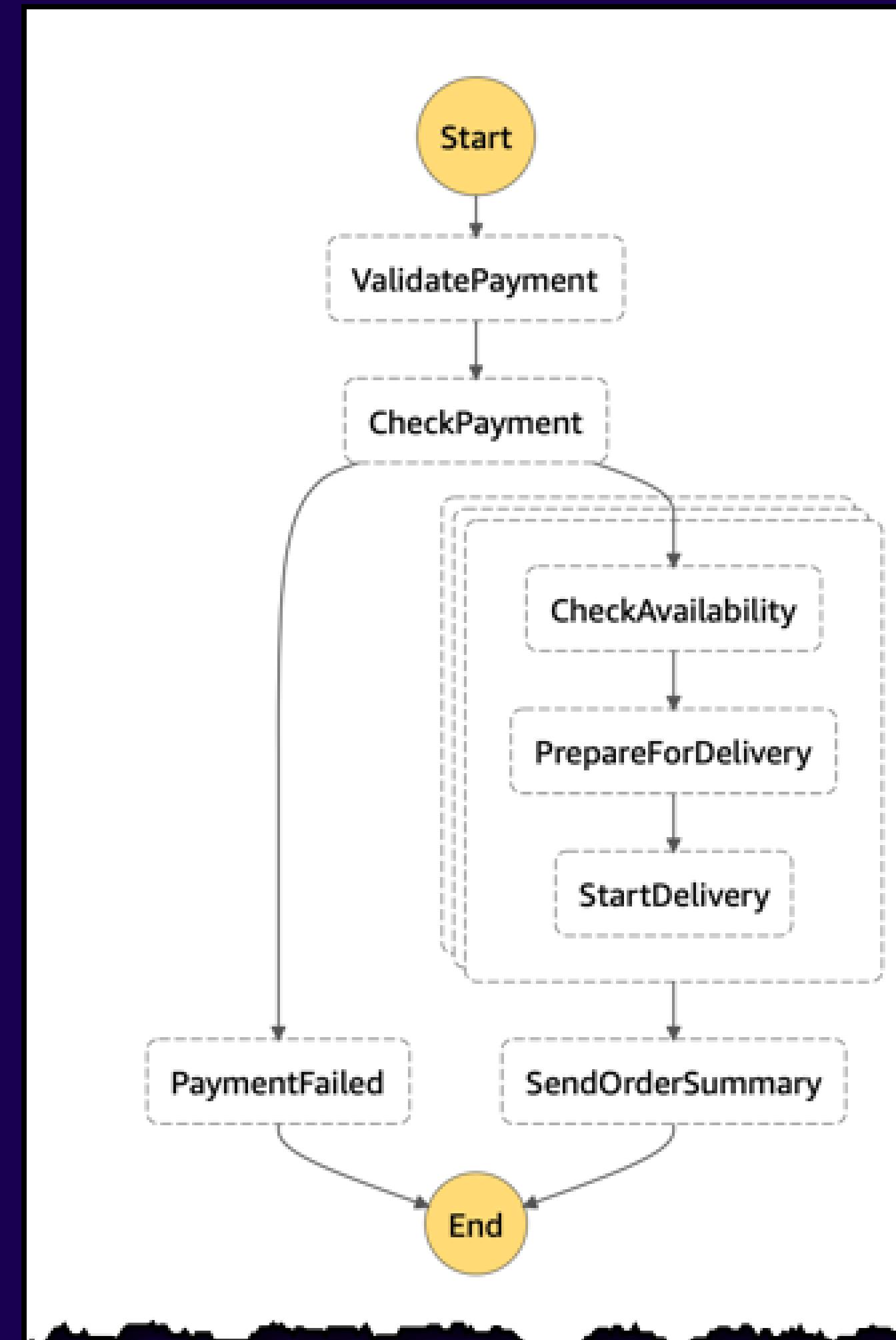
A screenshot of a dark-themed code editor window. The title bar shows three colored dots (red, yellow, green). The main area contains a JSON configuration for an AWS Step Functions state machine. The code defines a start state named 'Nested Workflow' which triggers a task state. The task state uses the 'startExecution.sync' built-in activity, specifying the state machine's ARN and a parameter 'SomeVariable' set to '\$.SomeValue'. The task state is marked as the end of the workflow.

Dynamic parallelism



- “Map State”
- Run identical tasks in parallel - MaxConcurrency
- Patterns
- **Fanout pattern** - dispatch a list of identical tasks to simplify workflows like order processing and instance management
- **Scatter-gather pattern** - accelerate workflows such as file processing

Dynamic parallelism



Step Functions

```
import { Stack, StackProps } from 'aws-cdk-lib';
import { Construct } from 'constructs';
import * as lambda from 'aws-cdk-lib/aws-lambda';
import * as sfn from 'aws-cdk-lib/aws-stepfunctions';
import * as sfnTasks from 'aws-cdk-lib/aws-stepfunctions-tasks';
import { join } from 'path';
import { LambdaInvoke } from 'aws-cdk-lib/aws-stepfunctions-tasks';

export class ParallelizedStepFunctionsStack extends Stack {
  constructor(scope: Construct, id: string, props?: StackProps) {
    super(scope, id, props);

    // Create state that generates a timestamp from the timeoutInSeconds
    const convertTimeoutToTimestampTask = this.getConvertTimeoutTask();

    // Wait state that waits until time specified by convertTimeoutToTimestampTask step
    const waitState = new sfn.Wait(this, 'Wait for Timer', {
      time: sfn.WaitTime.timestampPath('$.waitTimestamp.isoString'),
    });

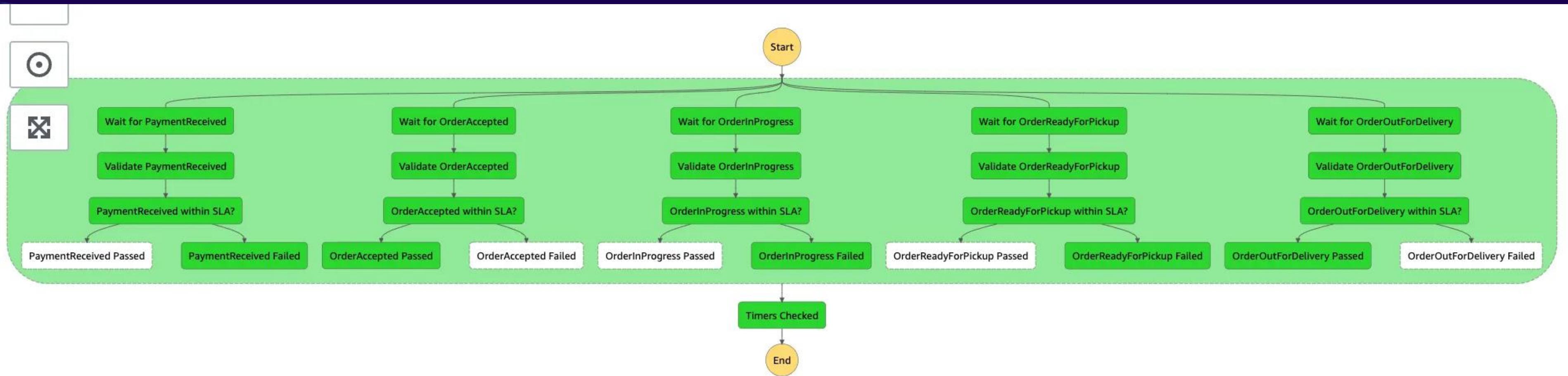
    // State that checks to see if some work has been completed after timeout
    const checkTimeoutTask = this.getCheckTimeoutTask();

    // States that handle pass/fail (these are mocks for simplicity)
    const timeoutPass = new sfn.Pass(this, 'Timeout Pass');
    const timeoutFail = new sfn.Pass(this, 'Timeout Fail');
    const choiceState = new sfn.Choice(this, 'SLA Met?')
      .when(sfn.Condition.stringEquals('$Payload.status', 'PASSED'), timeoutPass)
      .when(sfn.Condition.stringEquals('$Payload.status', 'FAILED'), timeoutFail);
```

Step Functions

```
// All of the state machine components have been created
// So now let's define the `Map` state, which is the dynamically parallelized state
const map = new sfn.Map(this, 'Parallelized timers', {
  inputPath: sfn.JsonPath.entirePayload,
  // The number of objects in the `timers` list will determine the number of parallelized states
  itemsPath: sfn.JsonPath.stringAt('$timers'),
  // Maximum number of parallelized states. If `timers` has more
  // than 50 object, it will handle the first 50, then the next 50, and so on
  maxConcurrency: 50,
  parameters: {
    // This will pass each list entry from the `timers` list
    // as an object named `timeoutData` to each map state
    'timeoutData.$': '$$.Map.Item.Value',
    // This will pass the top level `orderId` to each map state
    'orderId.$': '$.orderId'
  }
});
// Add the steps to the parallelized state machine (in order)
map
  .iterator(
    convertTimeoutToTimestampTask
      .next(waitState).next(checktimeoutTask).next(choiceState)
  );
// Create State Machine
new sfn.StateMachine(this, 'OrderSlaAlerts', {
  definition: map,
});
}
```

Step Functions



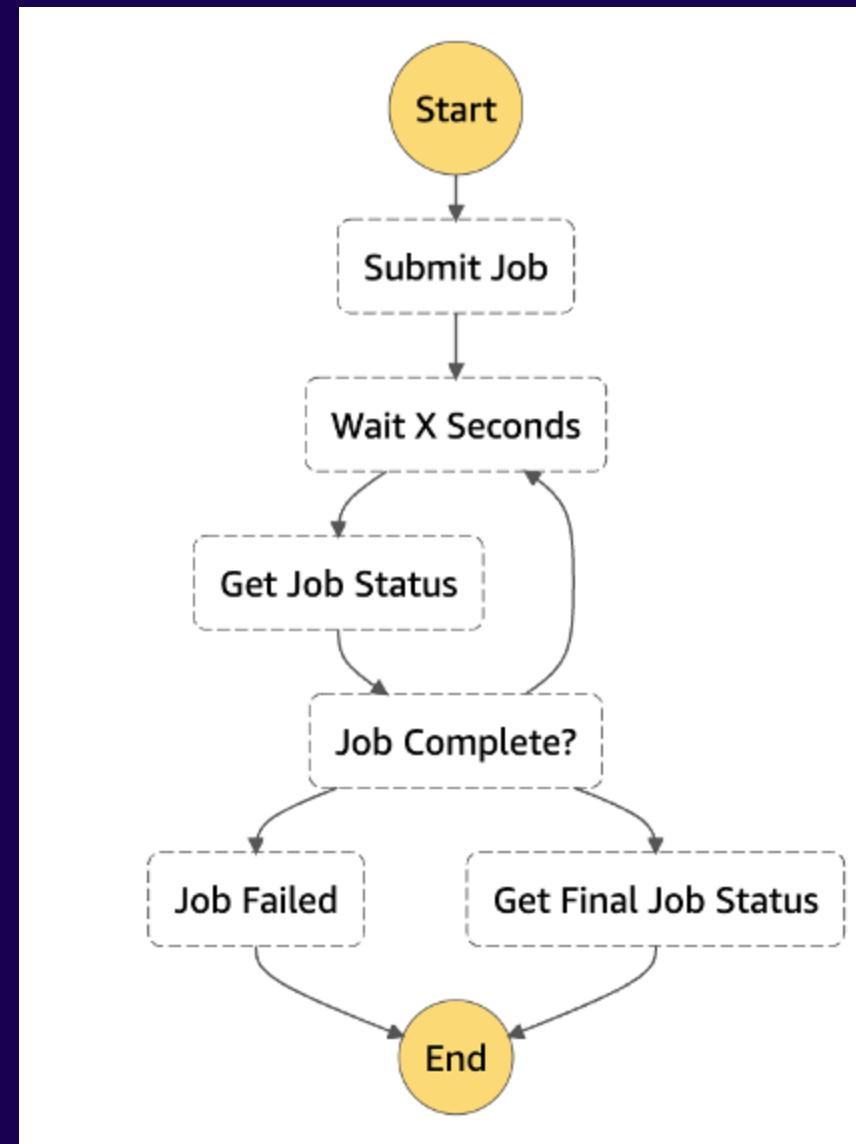
Express Workflows

	Express Workflows	Standard Workflows
Maximum duration	5 minutes	1 year
Execution start rate	Over 100,000/s	Over 2,000/s
State transition rate	Nearly unlimited	Over 4,000/s
Pricing	GB (memory) * s (time)	Per state transition
Execution history	CloudWatch logs	API, console, logs
Execution semantics	At-[least most]-once	Exactly-once
Service integrations	No Job-run or Callback	All integrations and patterns

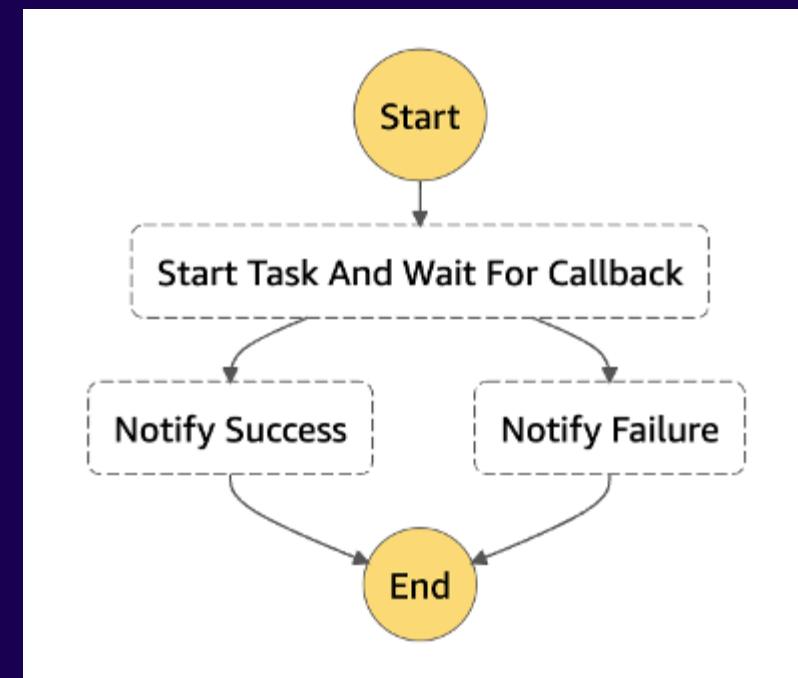
Seven state types

Task	A single unit of work
Choice	Adds branching logic
Parallel	Fork and join the data across tasks
Wait	Delay for a specified time
Fail	Stops an execution and marks it as a failure
Succeed	Stops an execution successfully
Pass	Passes its input to its output

Some Anti-Patterns/Patterns



VS



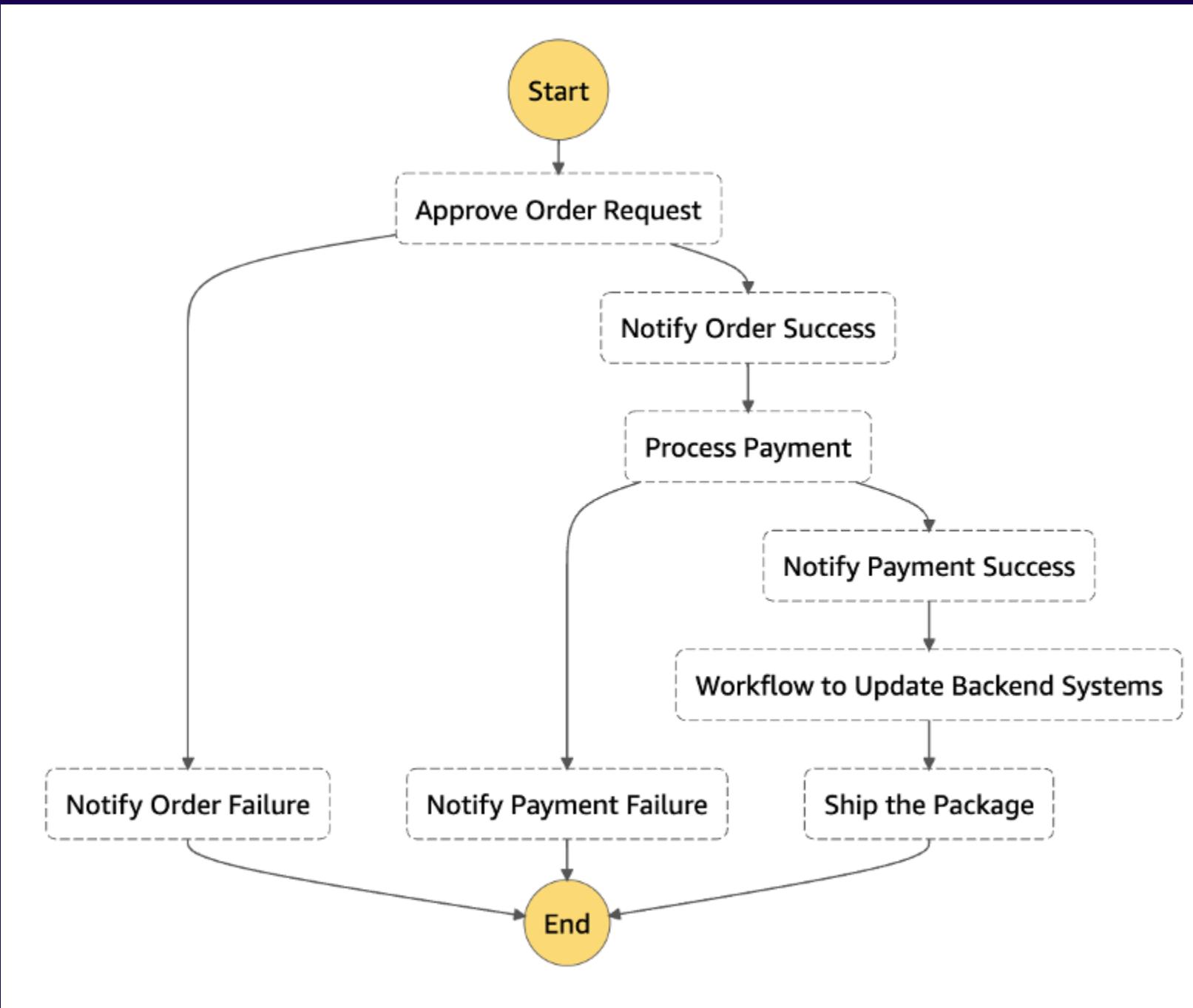
Use .sync and Callback()
AWS Batch and AWS ECS as examples

Some Anti-Patterns/Patterns

Express Workflows should be idempotent. Use it for high-volume event-processing workloads such as IoT data ingestion streaming data processing and transformation, and mobile application backends

Express and Standard Workflows

These can be combined. Depends on your use case



Error handling

Task and parallel states can have a field named Retry, which represents a certain number of retries, usually at increasing time intervals

When a state reports an error and either there is no Retry field, or if retries fail to resolve the error, Step Functions can fallback using a Catch field

Retriers

- Task, Parallel, and Map States
- Switch/case statement on *ErrorEquals* value
- *IntervalSeconds* is the number of seconds before first retry
- *MaxAttempts* is the number of retries (may be 0)
- *BackoffRate* is a multiplier



The screenshot shows the AWS Lambda function configuration interface. In the top right corner, there are three colored dots: red, yellow, and green. Below them, the "Retries" section is displayed. It contains two entries in an array:

```
"Retry": [ { "ErrorEquals": [ "States.Timeout" ], "MaxAttempts": 0 }, { "ErrorEquals": [ "States.All" ], "IntervalSeconds": 3, "MaxAttempts": 2, "BackoffRate": 1.5 } ]
```

Catchers

```
● ● ●  
"Catch": [  
  {  
    "ErrorEquals": [ "States.Timeout" ],  
    "ResultPath": "$.error-info",  
    "Next": "RecoveryState"  
  },  
  {  
    "ErrorEquals": [ "States.ALL" ],  
    "Next": "EndMachine"  
  }  
]
```

- Task, Parallel, and Map States
- Also known as *Fallback States*
- Scanned in array order when there is an error and no Retrier or all retries have failed
- Switch/case statement on *ErrorEquals* value
- *ResultPath* for storing error
- *Next* defines the next state

AWS Serverless Application Model (AWS SAM)

- AWS::Serverless::StateMachine component
- Use *DefinitionURI* and *DefinitionSubstitutions* to create workflows from separate files
- Apply AWS SAM policy templates to workflows
- AWS SAM event sources

```
$ sam init                                     <aws:administrator>
Which template source would you like to use?
  1 - AWS Quick Start Templates
Choice: 1

Which runtime would you like to use?
  4 - go1.x
Runtime: 4

Project name [sam-app]: step-functions-sam

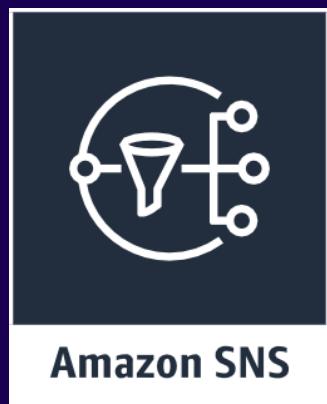
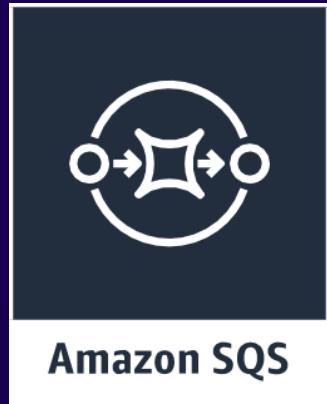
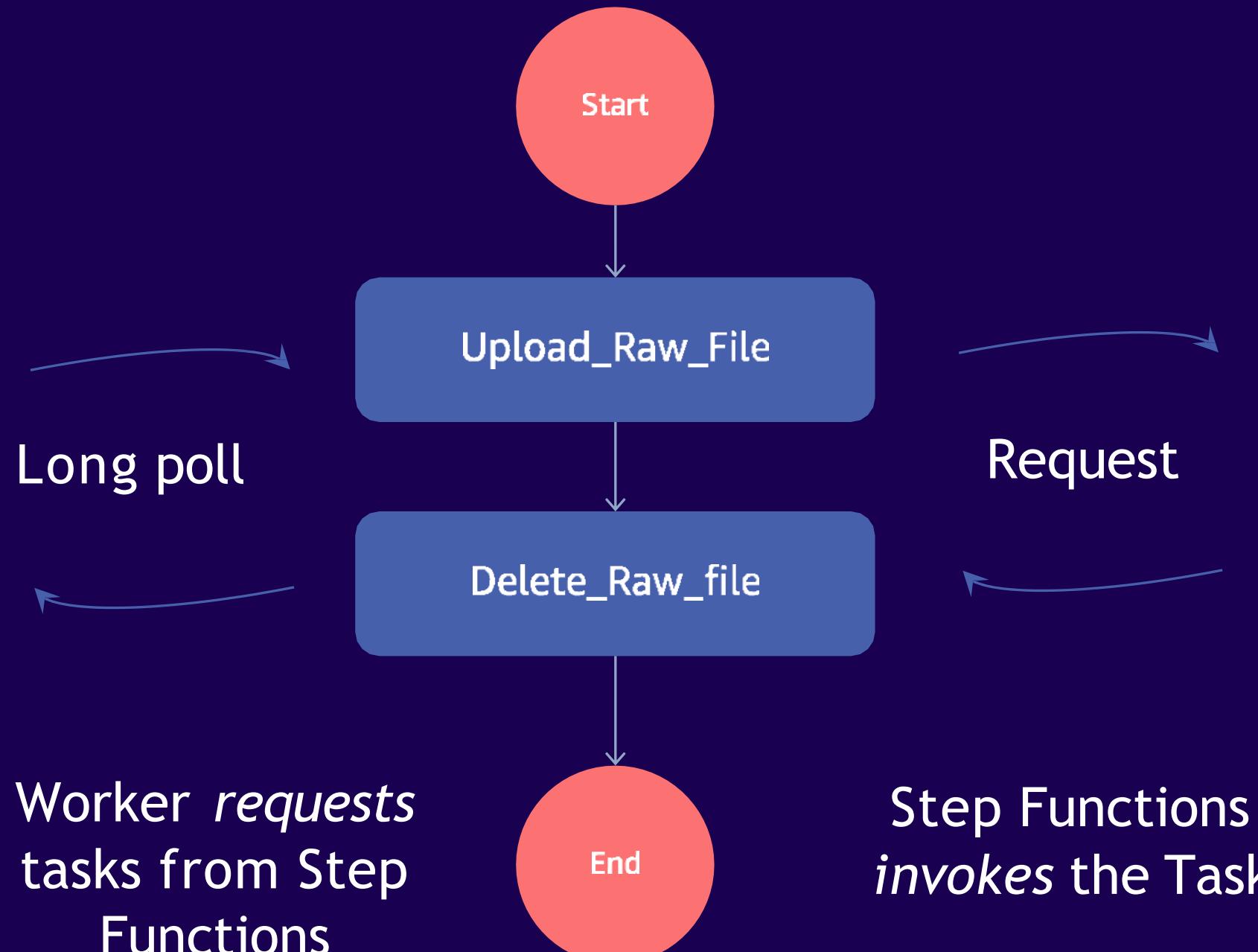
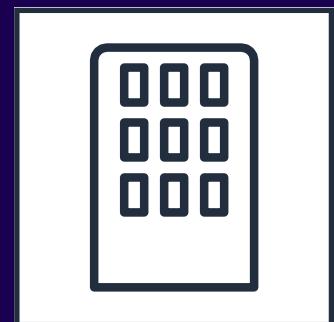
Cloning app templates from https://github.com/awslabs/aws-sam-cli-app-templates.git

AWS quick start application templates:
  2 - Step Functions Sample App (Stock Trader)
Template selection: 2

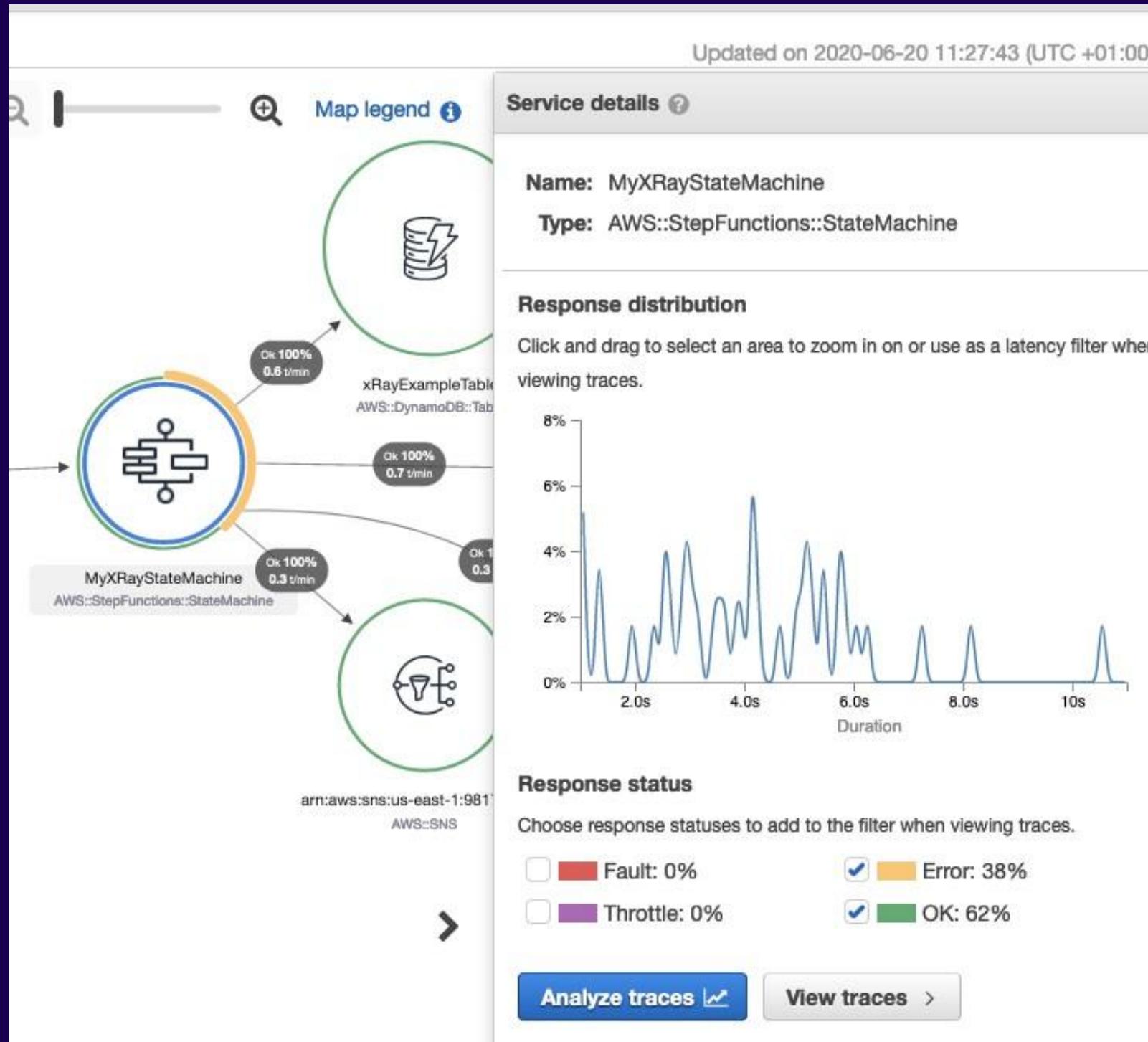
-----
Generating application:
-----
Name: step-functions-sam
Runtime: go1.x
Dependency Manager: mod
Application Template: step-functions-sample-app
Output Directory: .

Next steps can be found in the README file at ./step-functions-sam/README.md
```

Integration with AWS services



AWS X-Ray



- Distributed tracing
- Service map view shows information about a workflow and its services
- Trace map view shows a single trace in detail
- Trace timeline shows the propagation of a trace and latency distribution

DEMO

Knowledge check

Which of the following is not a feature of Amazon EC2?

- A. Broad selection of instance types for different workloads
- B. Fully managed compute service
- C. Multiple pricing options and per-second billing
- D. Complete control over instance and remote access options
- E. Reusable templates for launching additional instances (AMIs)

Knowledge check

Which of the following is not a feature of Amazon EC2?

- A. Broad selection of instance types for different workloads
- B. Fully managed compute service (Lambda)
- C. Multiple pricing options and per-second billing
- D. Complete control over instance and remote access options
- E. Reusable templates for launching additional instances (AMIs)

Answer: B

Key Takeaways

- EC2 instances - Servers in the cloud!
 - Pay as you go pricing
 - Scale in/out as needed automatically
 - Different instance types (hardware) for your workloads
- Amazon ECS
 - Orchestration for your container deployments
- Serverless
 - You create the code, AWS manages the underlying compute
 - Lambda - On demand, per-request pricing to run code
 - Step Functions - Decouple code and Orchestration



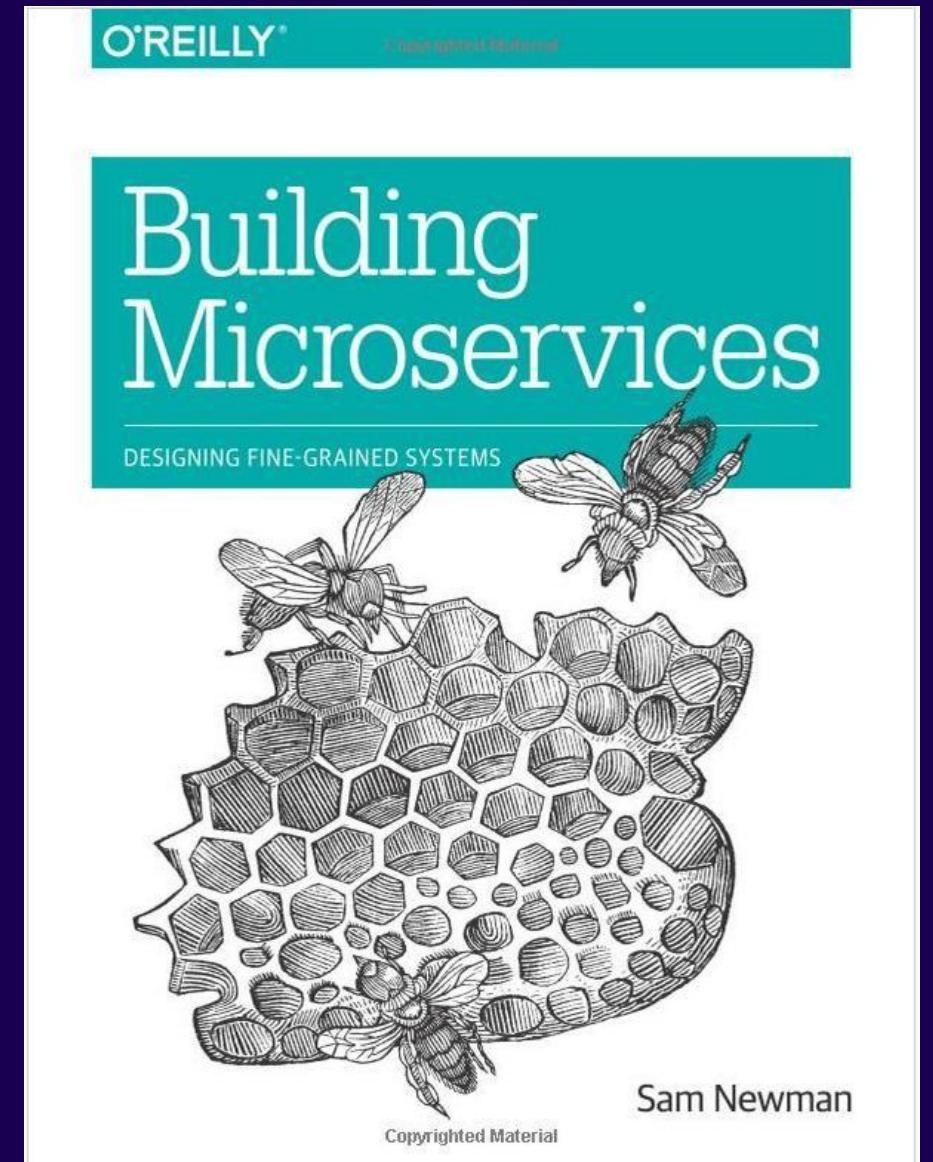
What makes a
microservice
“micro”?

What makes a microservice “micro”?

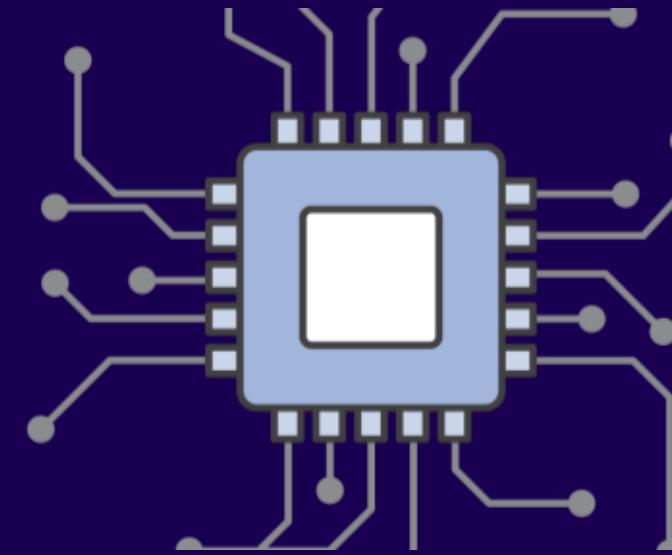
Too big of a topic to get into depth today!

Read about:

- Domain driven design
- Bounded Contexts
- CQRS models
- Smart endpoints, dumb pipes
- Sam Newman's book “**Building Microservices**” O'Reilly Publishing is a great place to start!



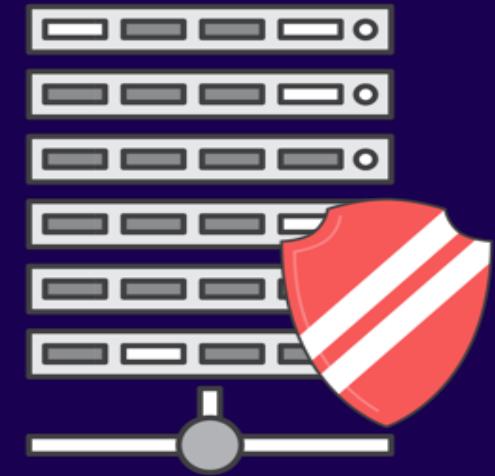
Amazon API Gateway



Create a unified
API frontend for
multiple micro-
services



DDoS protection
and throttling for
your backend



Authenticate and
authorize
requests to a
backend



Throttle, meter,
and monetize API
usage by third-
party developers

REST API
HTTP(s) API
WebSocket API

An Aside on APIs

REST APIs and HTTP APIs are both RESTful API products.

REST APIs support more features than HTTP APIs

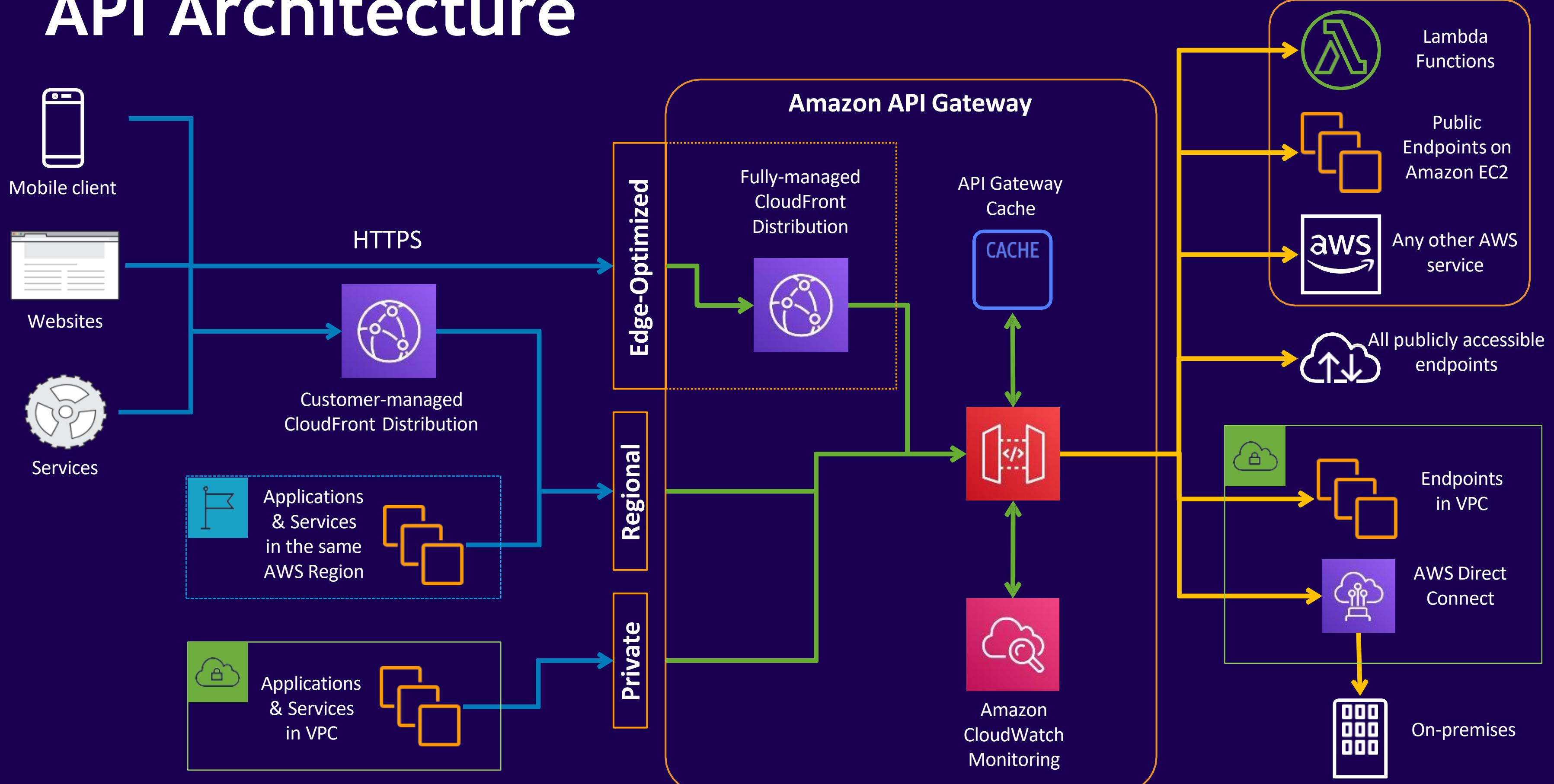
Choose REST APIs if you need features such as
API keys
per-client throttling
request validation
AWS WAF integration, or
private API endpoints.

An Aside on APIs

Key Differences Between HTTP API and REST API

Feature	HTTP API	REST API
Cost	Lower	Higher
Performance	Faster and lightweight	Slightly slower due to more features
Learning Curve	Easier for beginners	More complex
Use Case	Basic APIs, modern apps	Enterprise apps, legacy systems
Request/Response Mapping	Not supported	Fully supported
Caching	Not available	Supported
API Keys & Usage Plans	Not available	Available
Throttling & Quotas	Basic	Detailed control
Integration Support	Lambda, HTTP endpoints	Lambda, HTTP, other AWS services

API Architecture



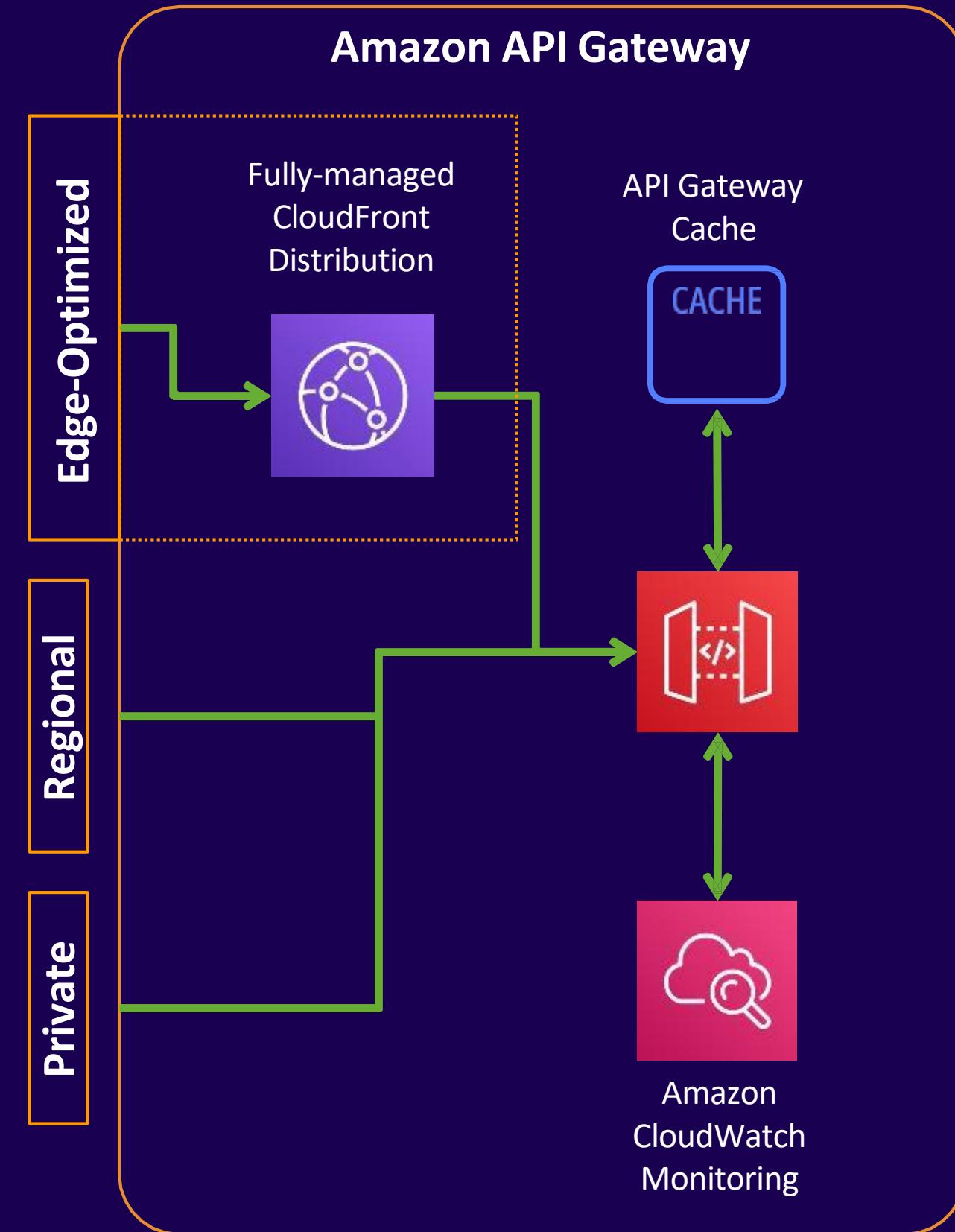
Type of APIs Available

Edge-Optimized

- Utilizes CloudFront to reduce TLS connection overhead (reduces roundtrip time)
- Designed for a globally distributed set of clients

Private

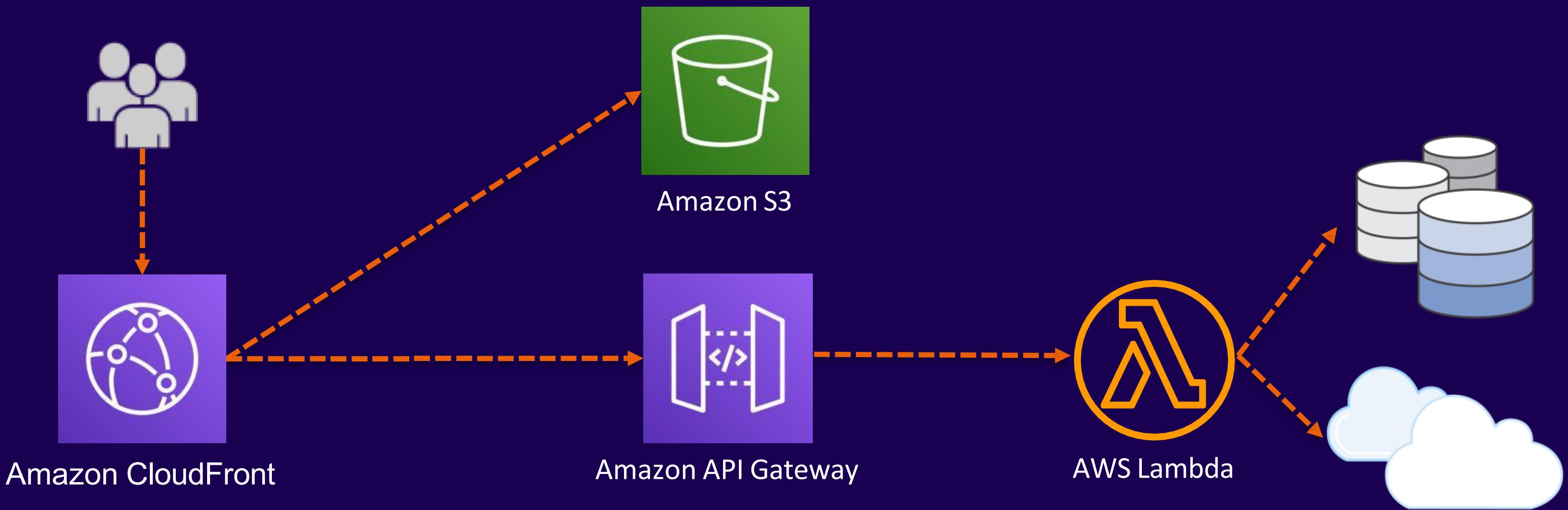
- Only accessible from within VPC (and networks connected to VPC)
- Designed for building APIs used internally or by private microservices



Regional

- Recommended API type for general use cases
- Designed for building APIs for clients in the same region

The coming wave of serverless web applications

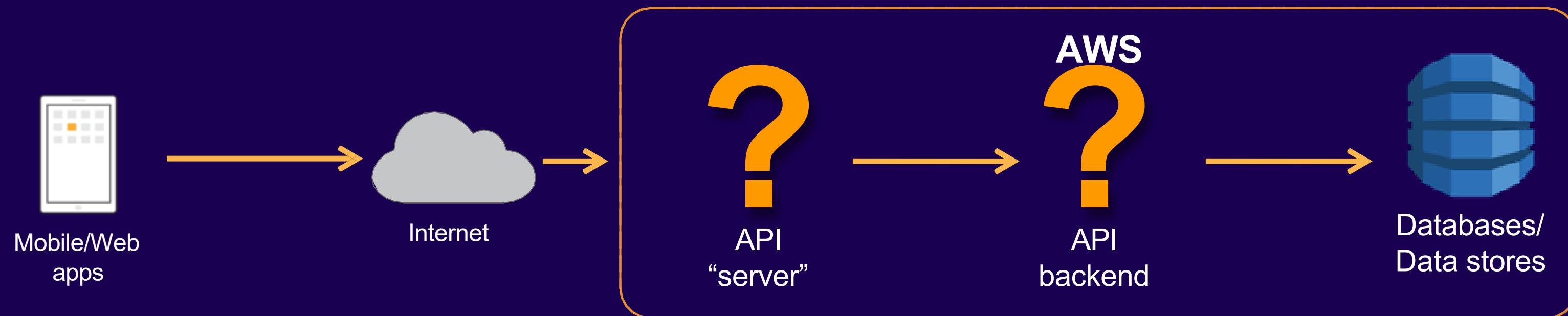


Amazon Simple Storage Service (Amazon S3) stores all of your static content: CSS, JS, images, and more. You would typically front this with a CDN such as CloudFront.

API Gateway handles all your application routing. It can handle authentication and authorization, throttling, DDOS protection, and more.

Lambda runs all the logic behind your website and interfaces with databases, other backend services, or anything else your site needs.

Basic API technology stack



API Management Challenges



Managing multiple versions and stages of an API is difficult.



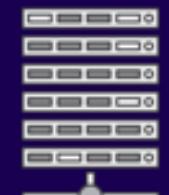
Monitoring third-party developers' access is time consuming.



Access authorization is a challenge.

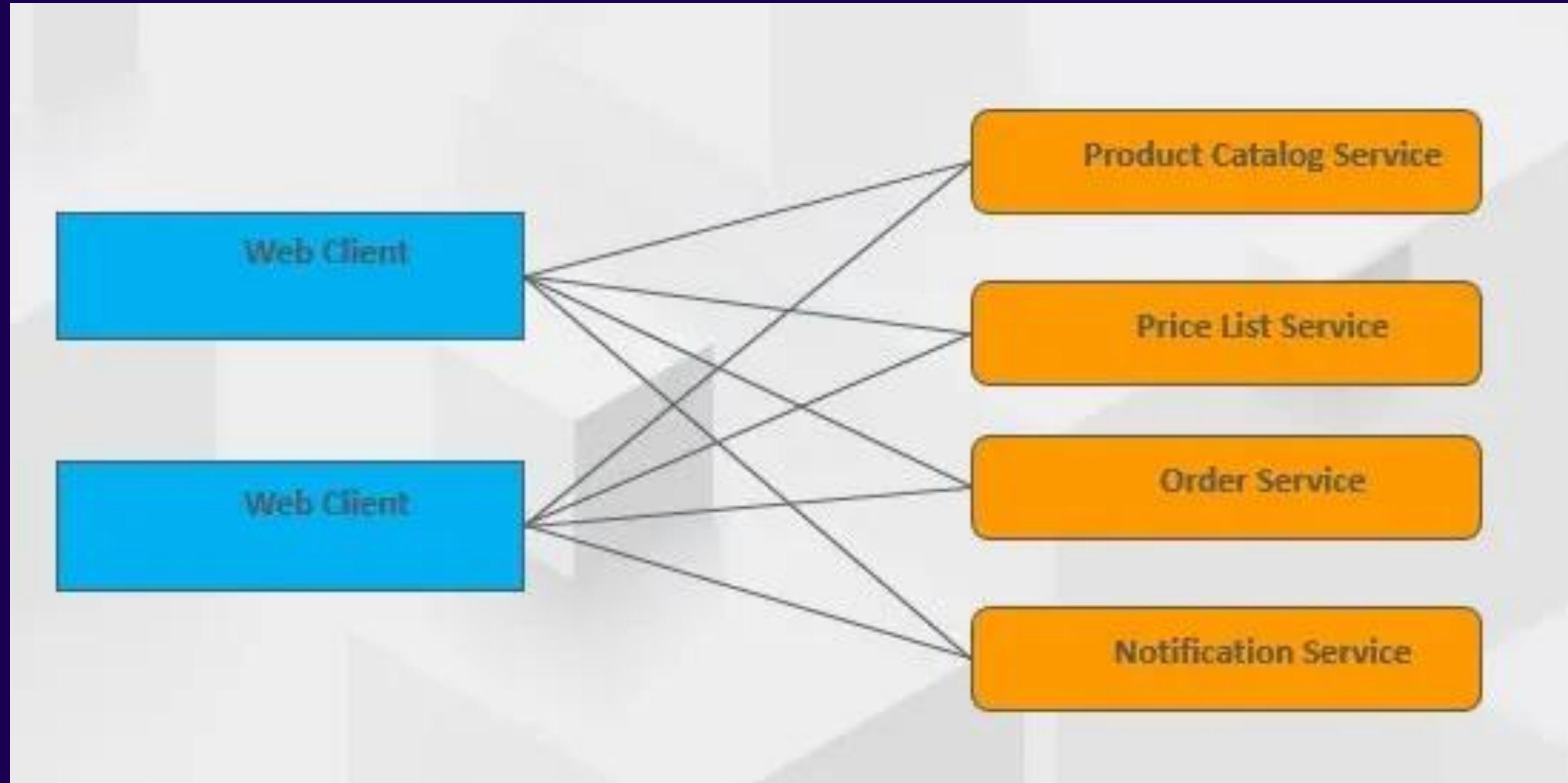


Traffic spikes create an operational burden.



Dealing with increased management overhead

API Management Challenges



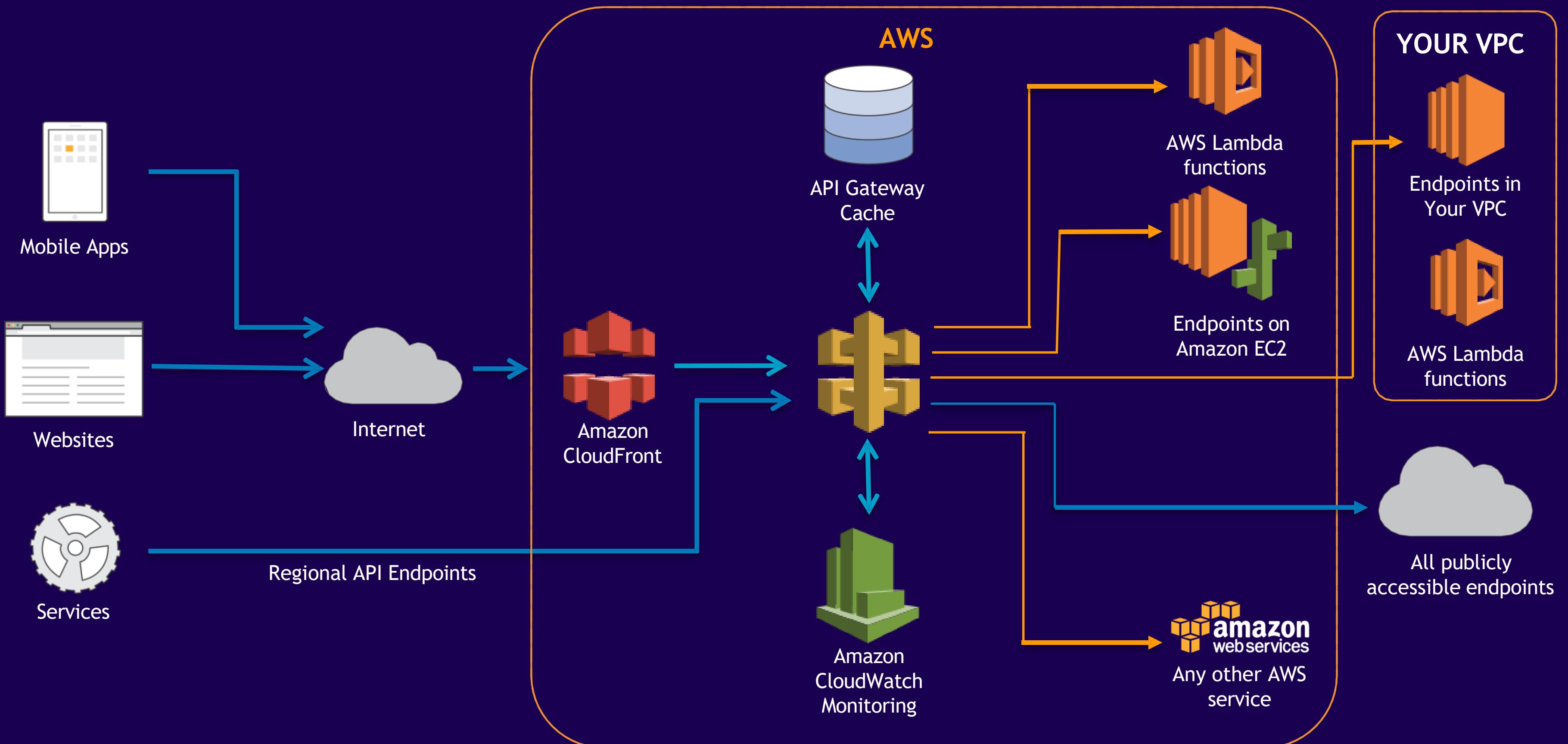
Introducing Amazon API Gateway

Amazon API Gateway is a fully managed service that makes it easy for developers to create, publish, maintain, monitor, and secure APIs at any scale:



- Host multiple versions and stages of your APIs
- Create and distribute API Keys to developers
- Throttle and monitor requests to protect your backend
- Leverage signature version 4 to authorize access to APIs
- Request / Response data transformation and API mocking
- Reduced latency and DDoS protection through CloudFront
- Optional Managed cache to store API responses
- SDK Generation for Java, JavaScript, Java for Android, Objective-C or Swift for iOS, and Ruby
- Swagger support

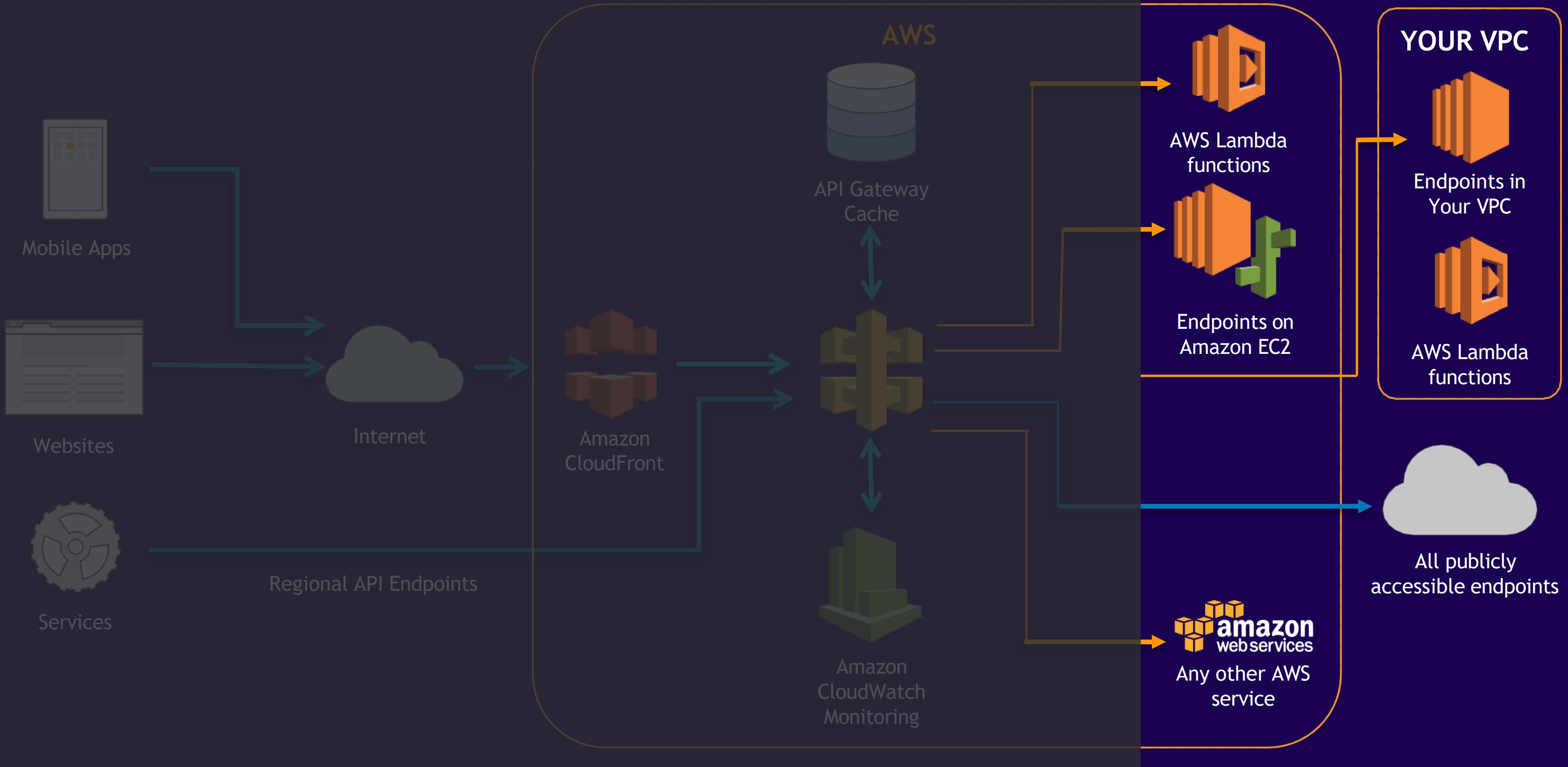
API Gateway integrations



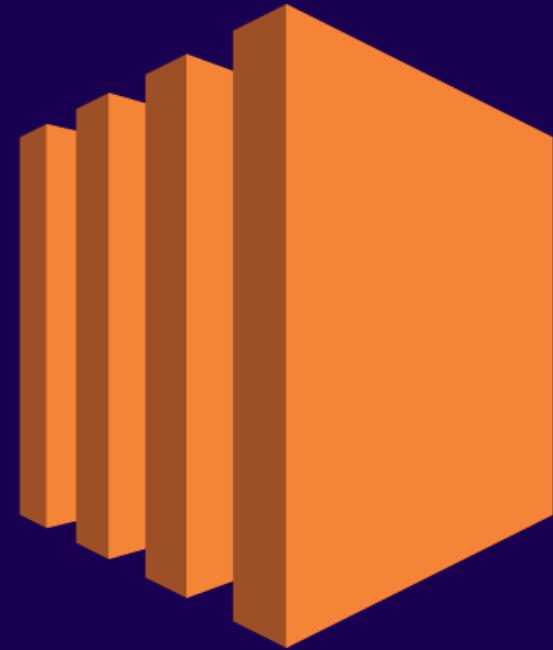
Basic API technology stack



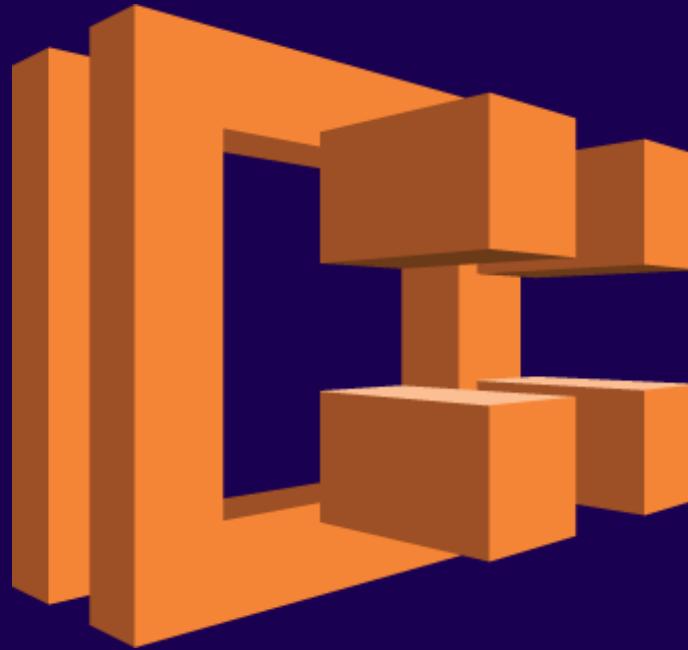
API Gateway backend integrations



AWS Compute Services



Amazon
EC2



Amazon
Elastic
Container
Service
(ECS)

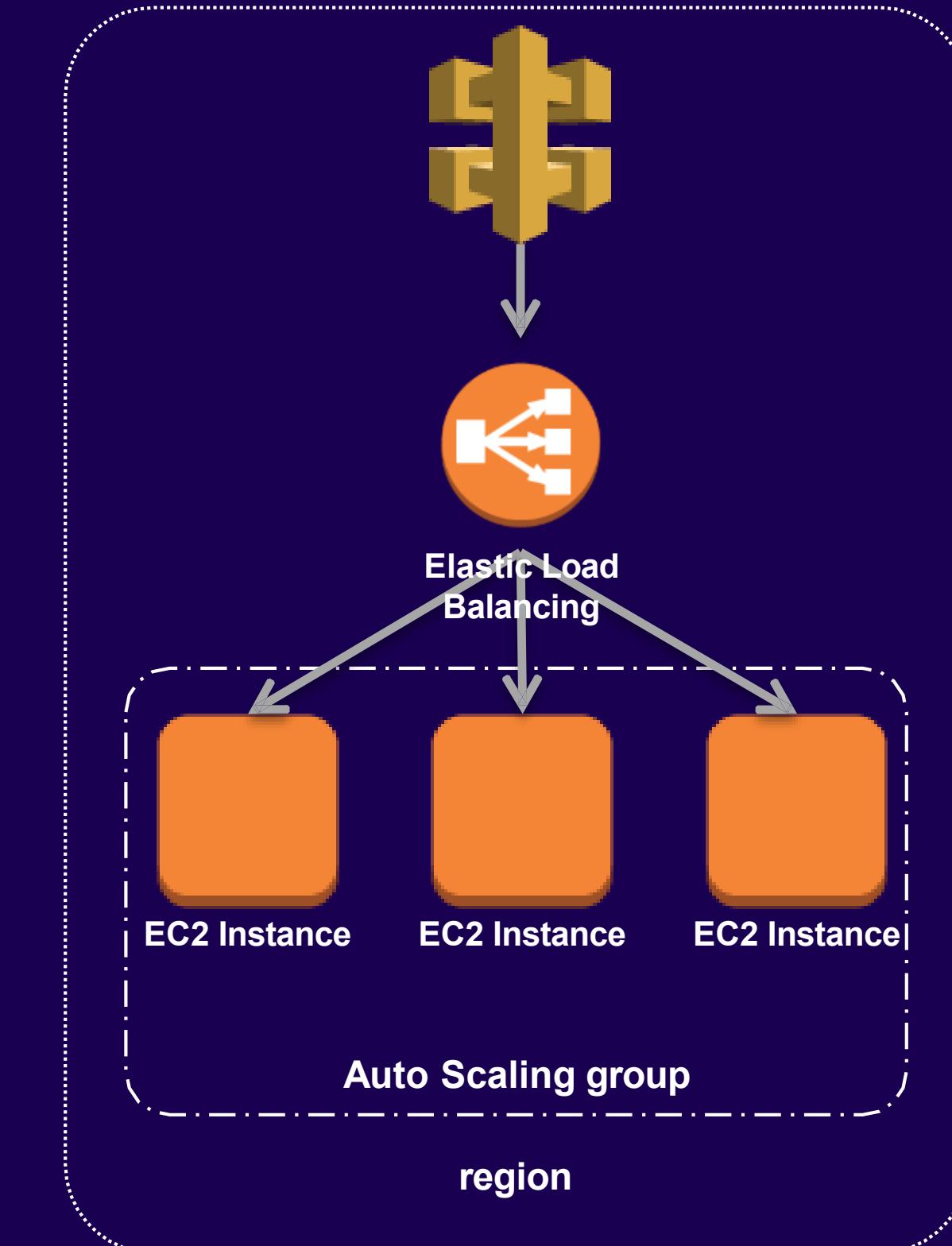


AWS
Lambda

Deploying Microservices on Amazon EC2

Recommendation:

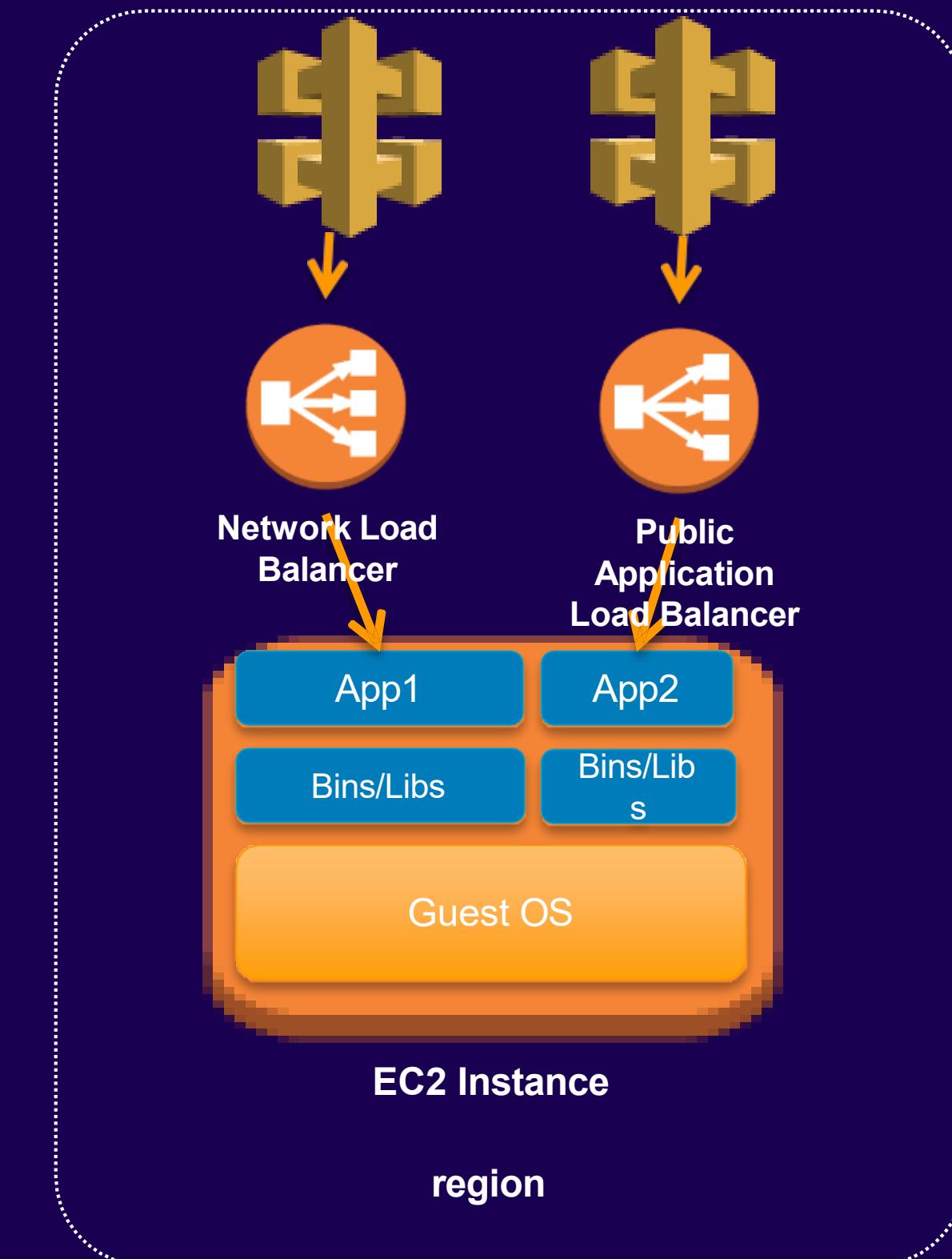
- Single service per host
- Start with small instance sizes
- Leverage Auto Scaling and AWS Elastic Load Balancing/Application Load Balancer/Network Load Balancer(if in VPC)
- Automate the ability to pump out these environments easily
 - Leverage CodeDeploy, CloudFormation, Elastic Beanstalk or Opsworks



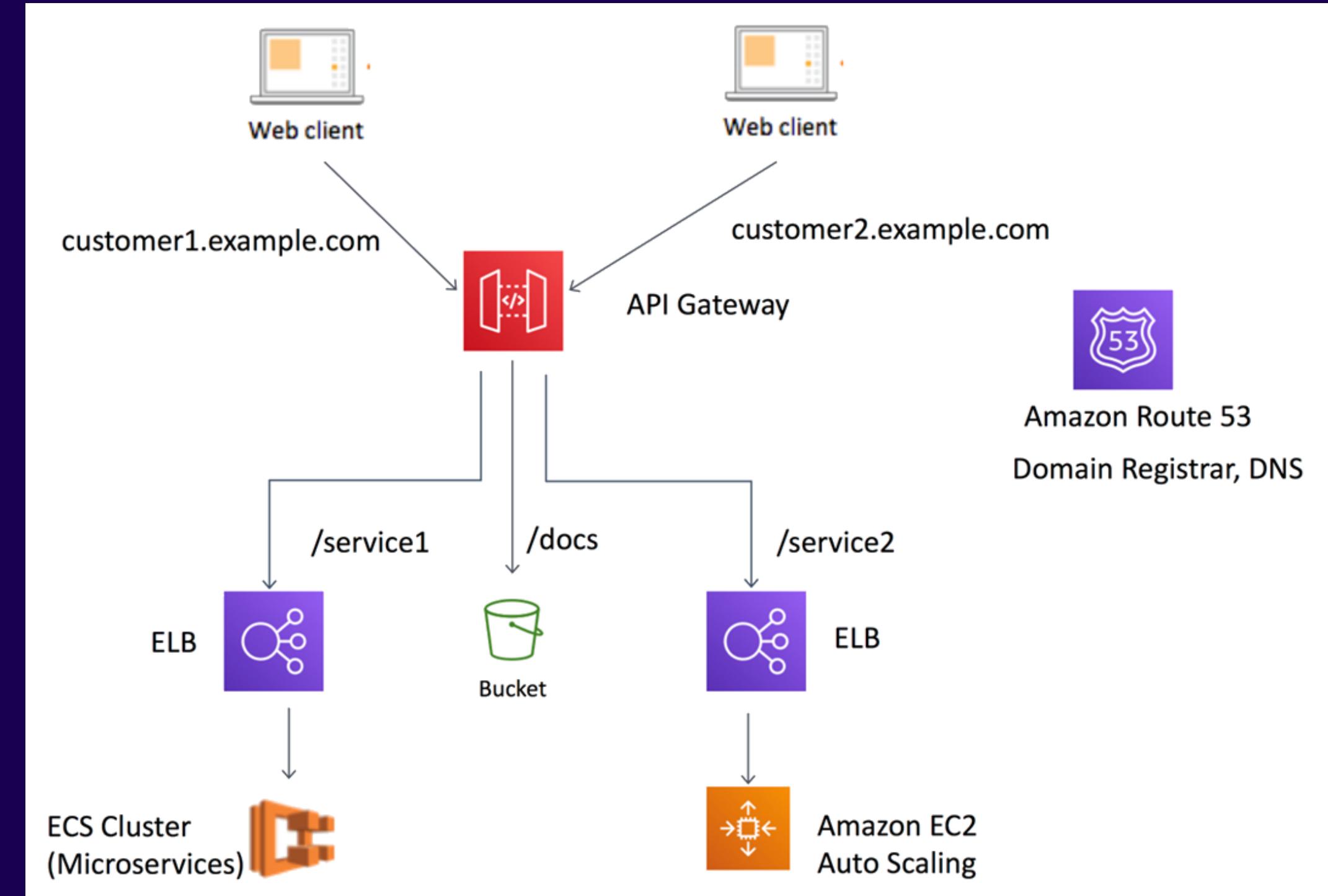
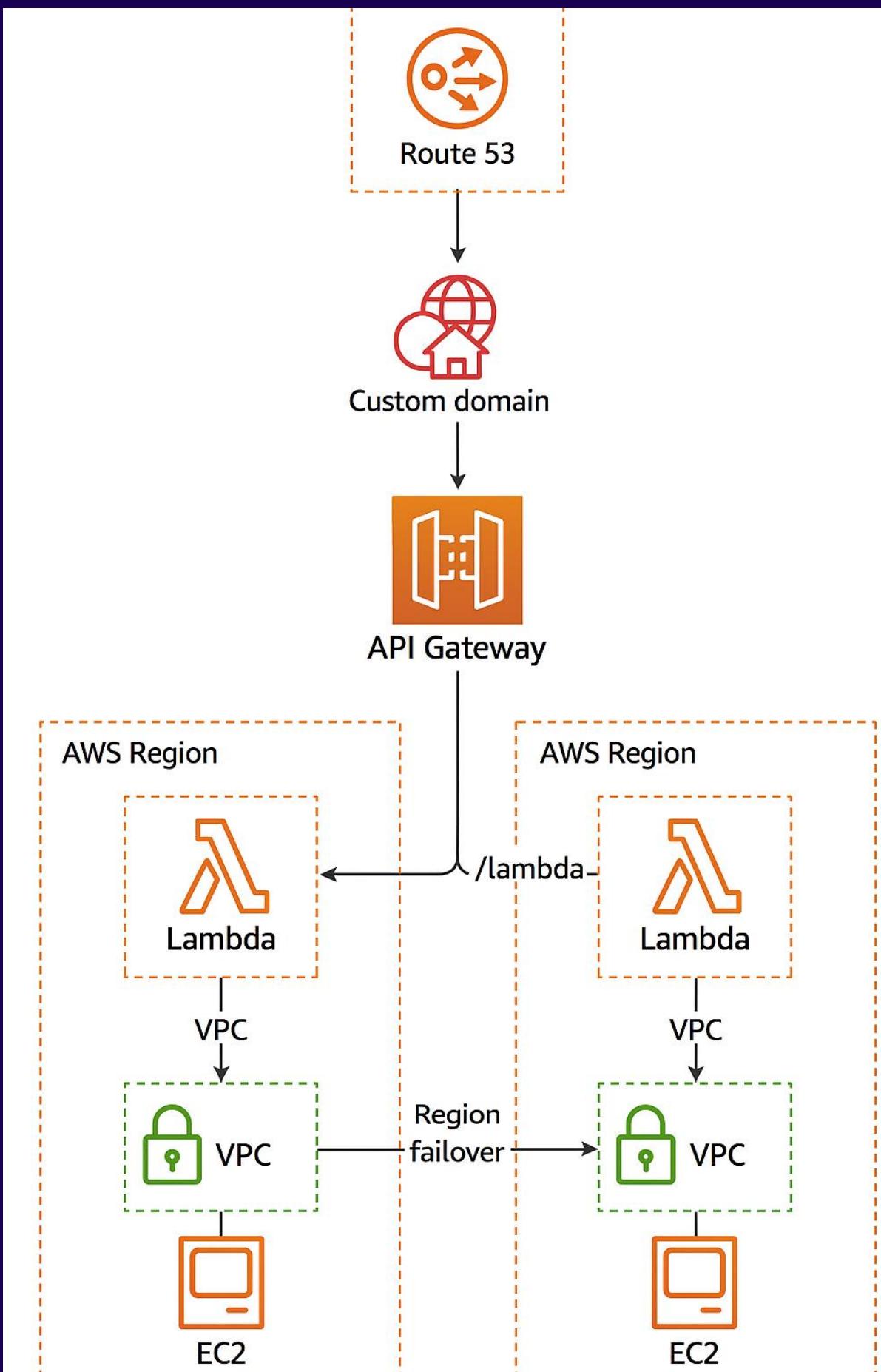
Deploying Microservices with ECS

Recommendation

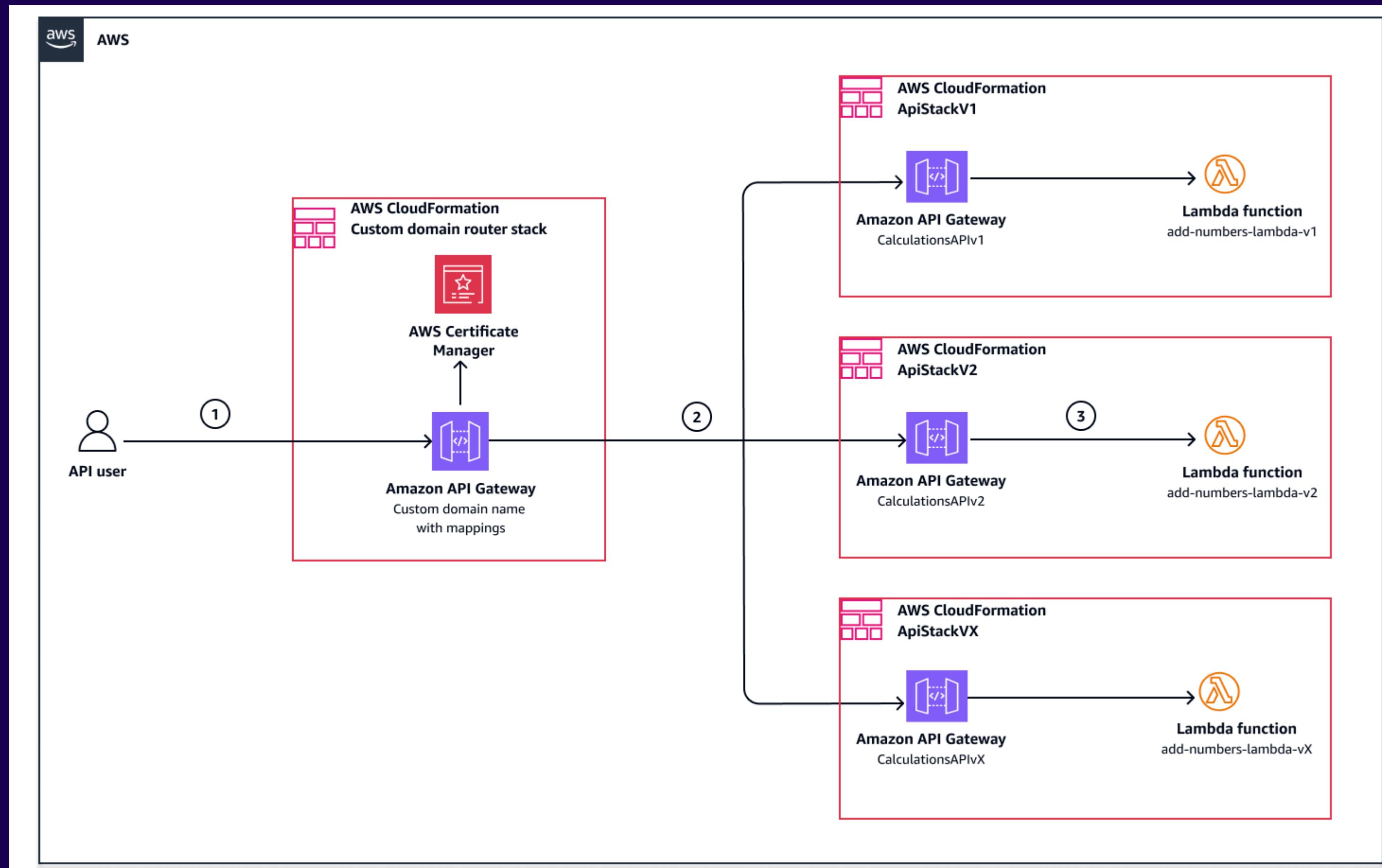
- Put multiple services per host
- Make use of larger hosts with much more CPU/RAM
- Run helper services on the same host as other dependent services
- Leverage Auto Scaling and AWS Elastic Load Balancing/Application Load Balancer/Network Load Balancer(if in VPC)
- Use AWS Fargate for even less administrative overhead!



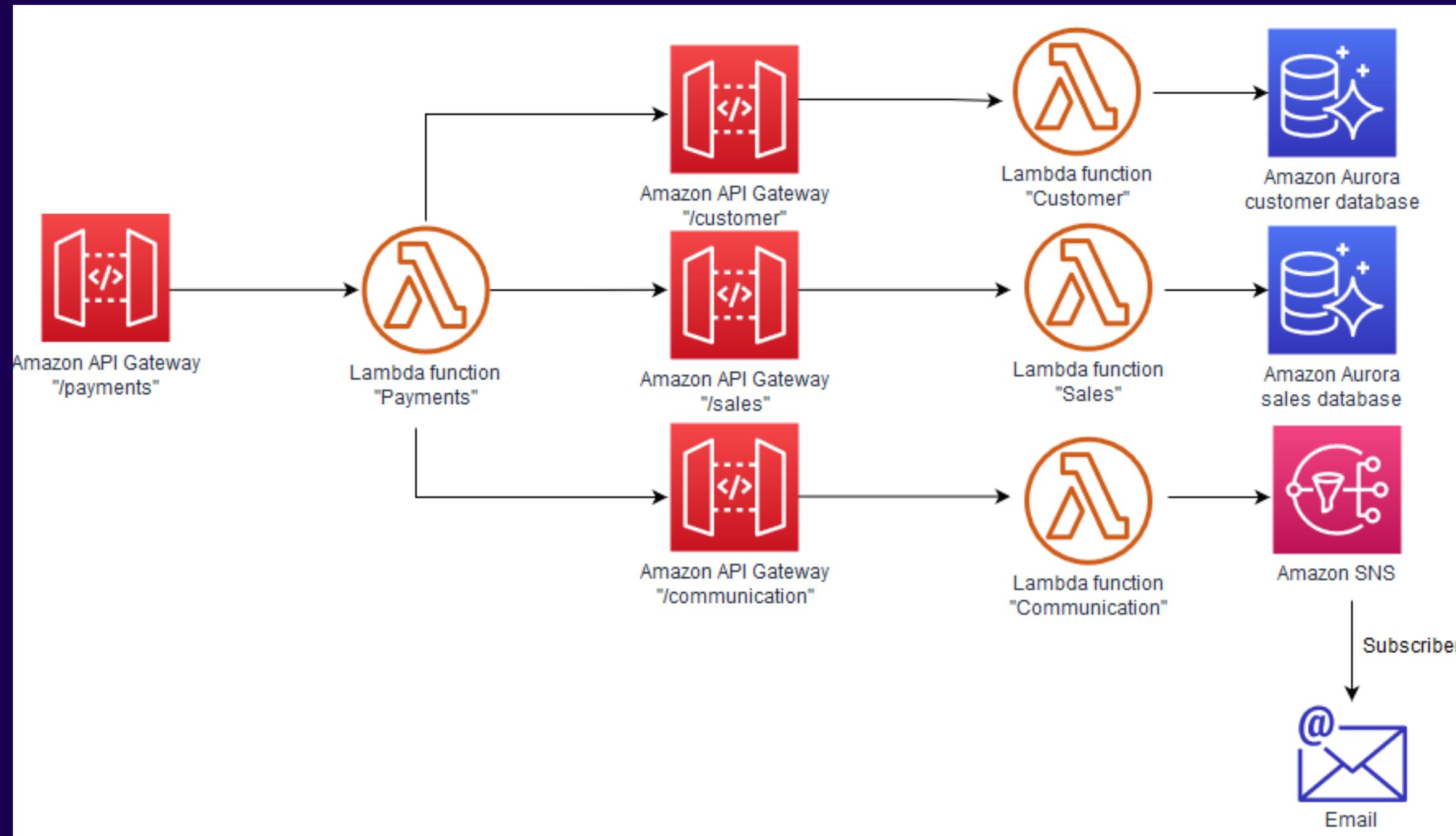
API Gateway



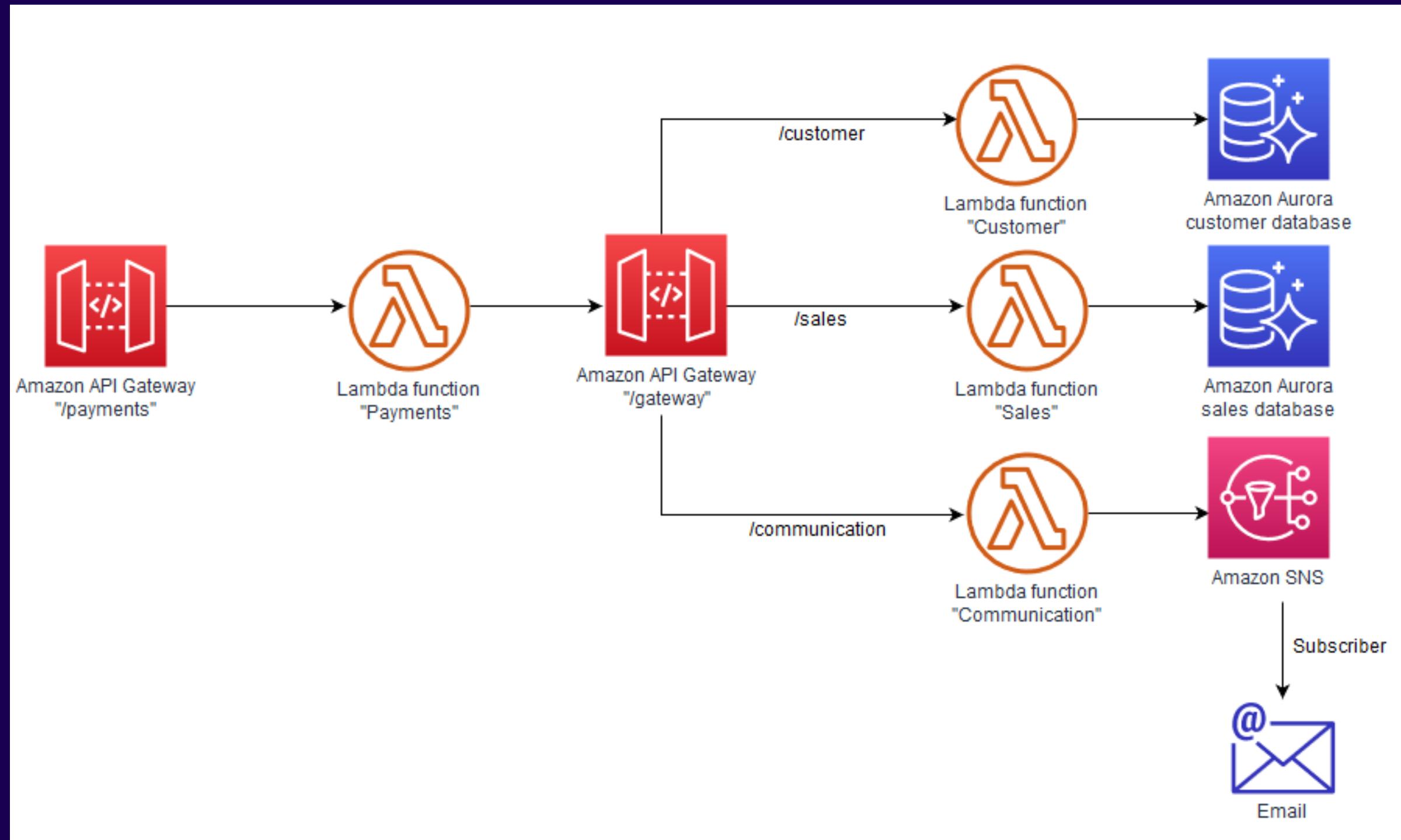
API Gateway



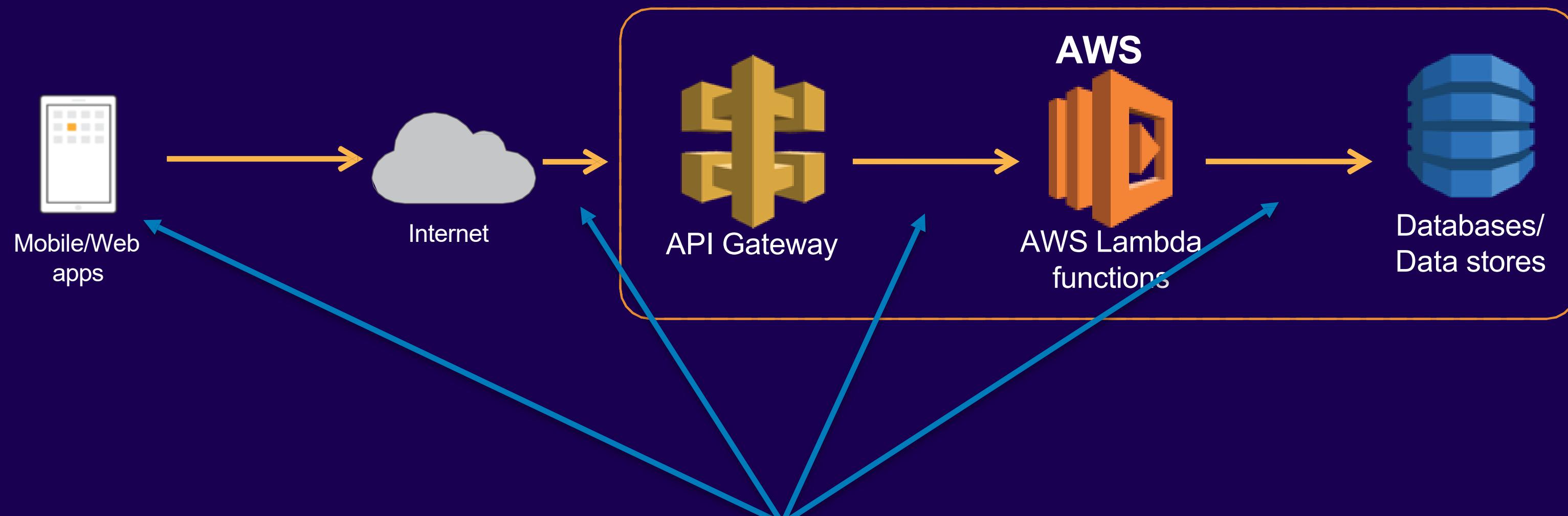
API Gateway



API Gateway



Basic Serverless API technology stack



places where we can secure our application

Amazon API Gateway Security

Several mechanisms for adding Authz/Authn to our API:

- IAM Permissions
 - Use IAM policies and AWS credentials to grant access
- Custom Authorizers
 - Use Lambda to validate a bearer token(Oauth or SAML as examples) or request parameters and grant access
- Cognito User Pools
 - Create a completely managed user management system

Authentication type comparison

Feature	AWS_IAM	TOKEN	REQUEST	COGNITO
Authentication	X	X	X	X
Authorization	X	X	X	
Signature V4	X			
Cognito User Pools		X	X	X
Third-Party Authentication		X	X	
Multiple Header Support			X	
Additional Costs	NONE	Pay per authorizer invoke	Pay per authorizer invoke	NONE



NEW!!!



Amazon EventBridge

Serverless event bus for ingesting and processing data
across AWS services and SaaS applications

- Removes friction of writing “point-to-point integrations”
- Fully managed, pay-as-you-go
- Works across dozens of AWS and SaaS applications
- Provides simple programming model



NEW!!!



Amazon EventBridge

Serverless event bus for ingesting and processing data
across AWS services and SaaS applications

- 90+ AWS Services as sources
- 17 AWS Services as targets
- 1\$ per 1 Million events put in to a bus
- No additional cost for delivery to targets

Amazon EventBridge

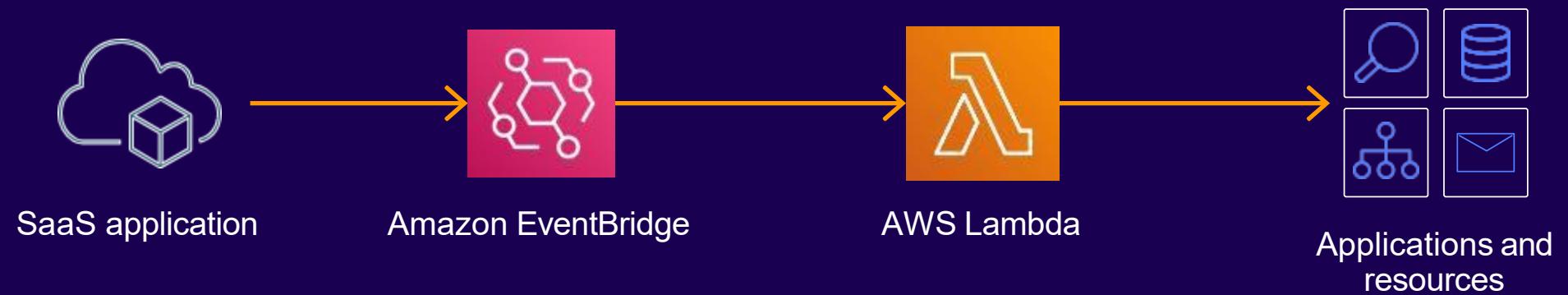
- Built on the same technology as CloudWatch Events
- Durably stores messages and retries failed connections to targets (for up to 24 hours)
- Secured by AWS Identity and Access Management (IAM)
- Event payload is JSON based, no dictated schema
- Events Put in via single API call from AWS-SDKs or via 3rd party partner SaaS providers



Amazon
EventBridge

Common use cases

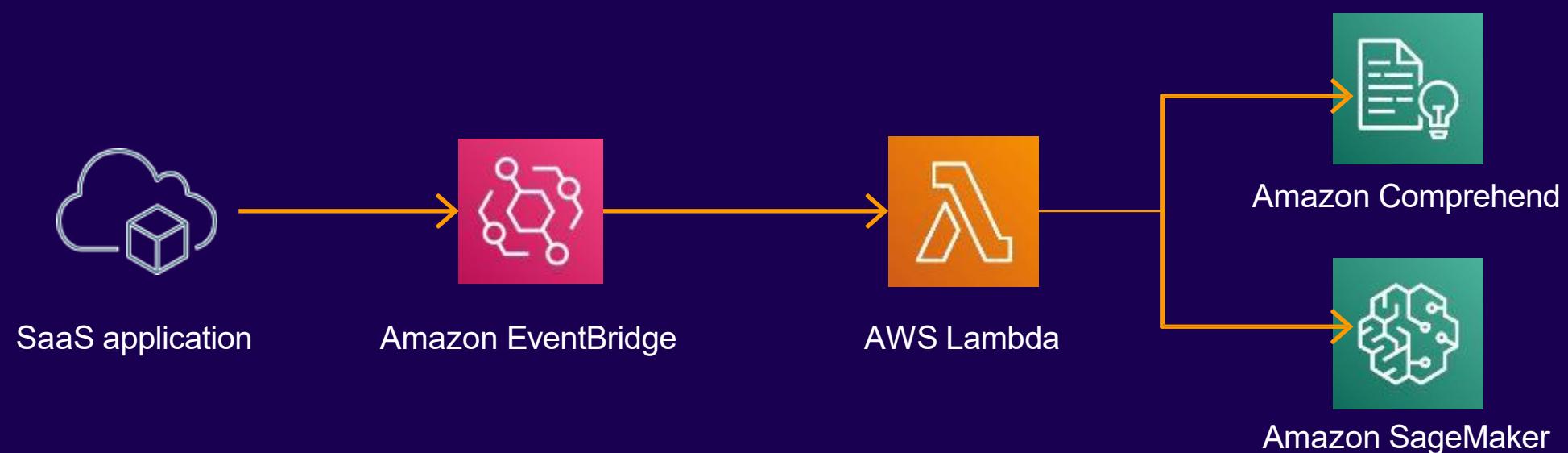
Take action



Run workflows



Apply intelligence

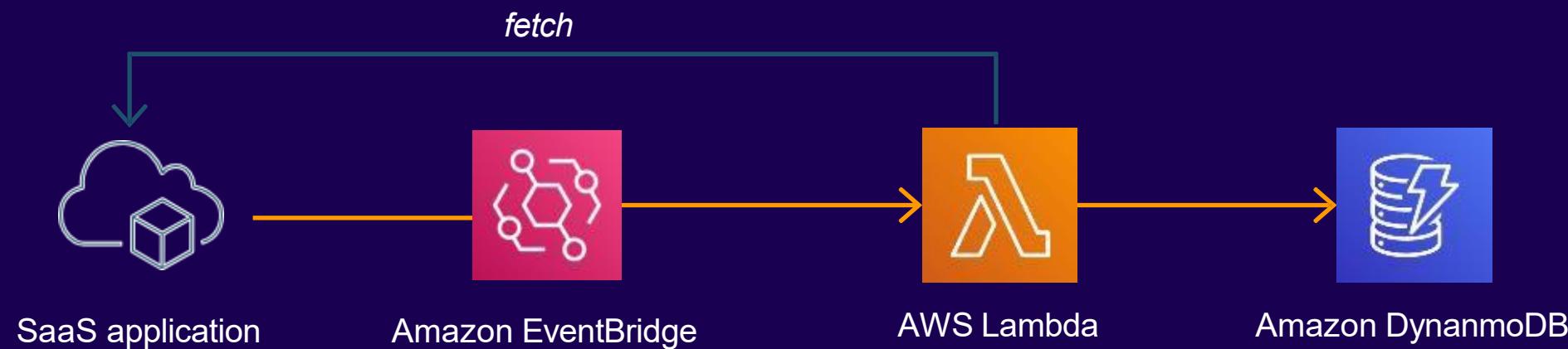


Common use cases

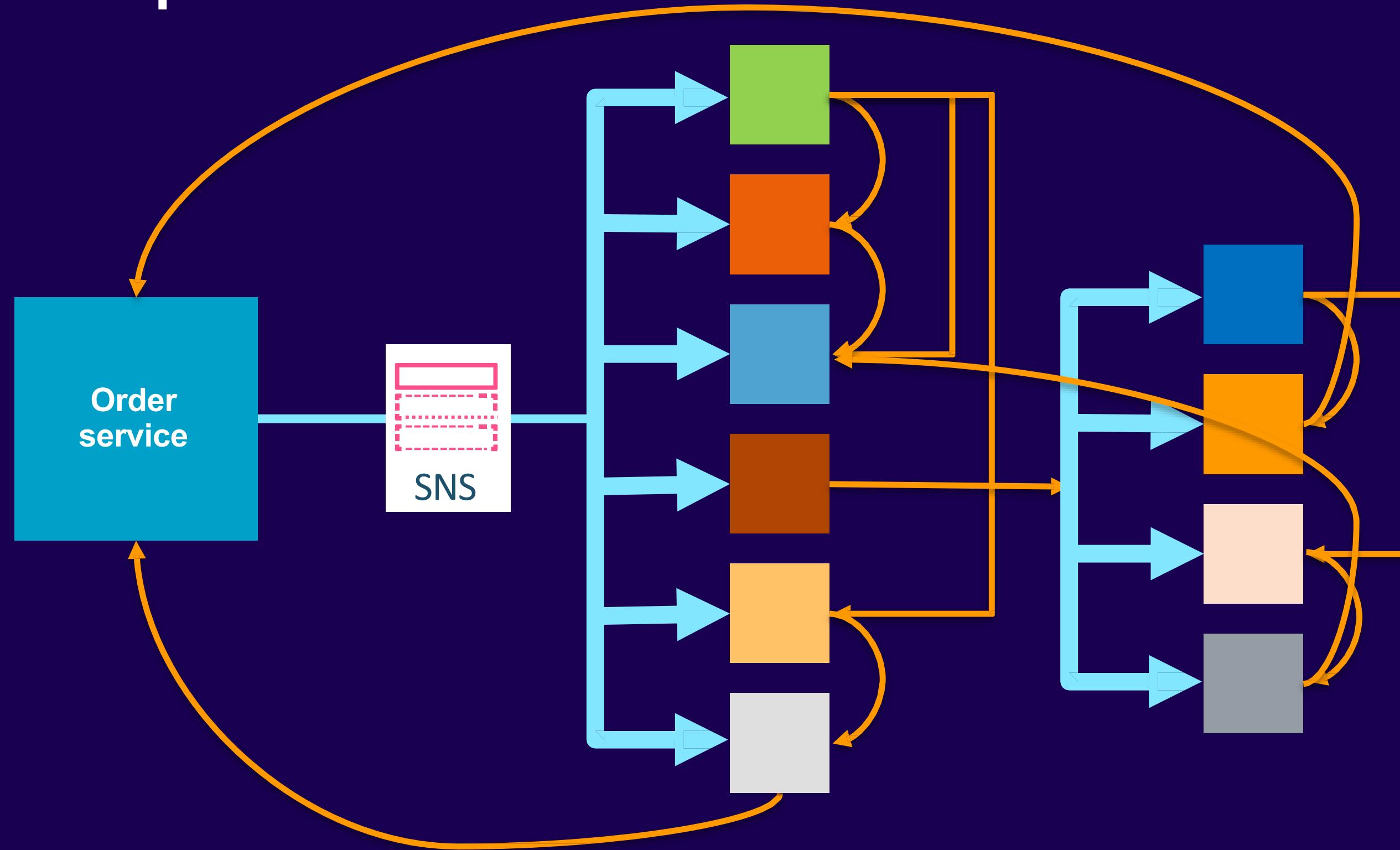
Audit and analyze



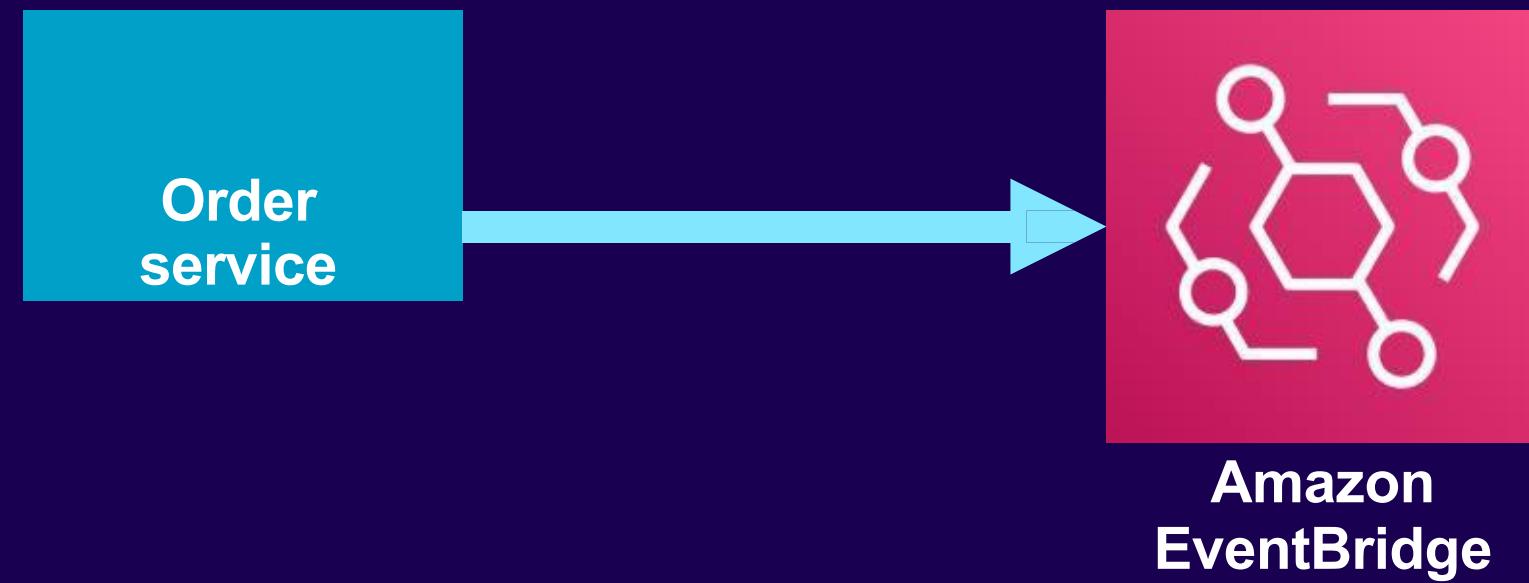
Synchronize data



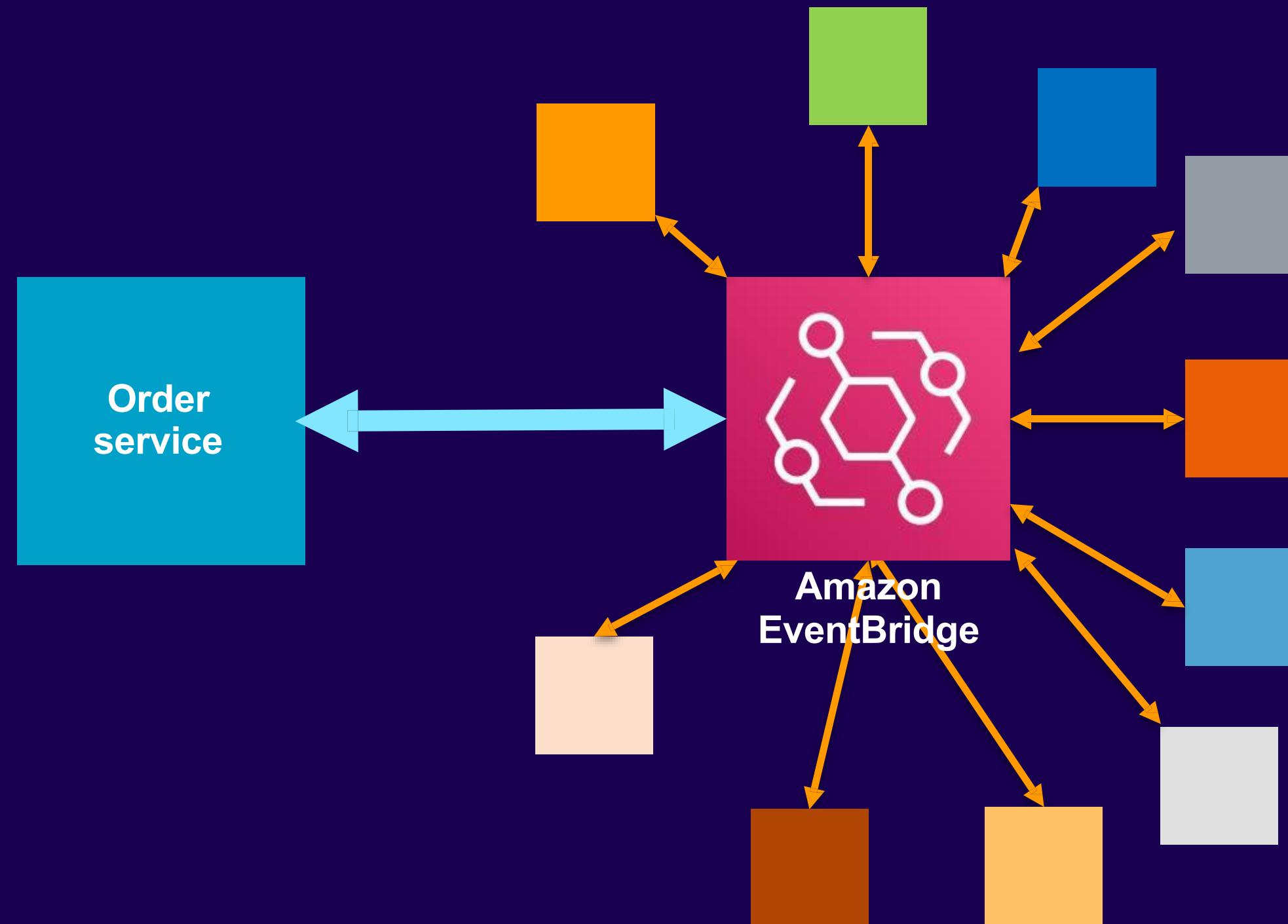
An example flow



Events with Amazon EventBridge



Events with Amazon EventBridge



- Your services can both produce messages onto the bus and consume just the messages they need from the bus
- Services don't need to know about each other, just about the bus.

Storage

Agenda

Introduction to Cloud and AWS

Cloud Computing

Terminologies

AWS Global Infra & services

Compute

Step functions, Review of EC2 and Storage

Databases

Networking

Security

AI and ML with AWS

Deploying Apps in AWS

Next steps

AWS storage options



Amazon S3

Scalable, highly durable object storage in the cloud



Amazon S3 Glacier

Low-cost, highly durable archive storage in the cloud



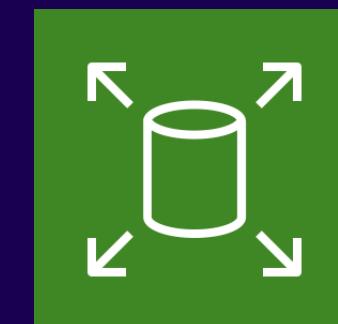
Amazon EFS

Scalable network file storage for Amazon EC2 instances



AWS Storage Gateway

Hybrid cloud storage service that gives you on-premises access to virtually unlimited cloud storage.



Amazon EBS

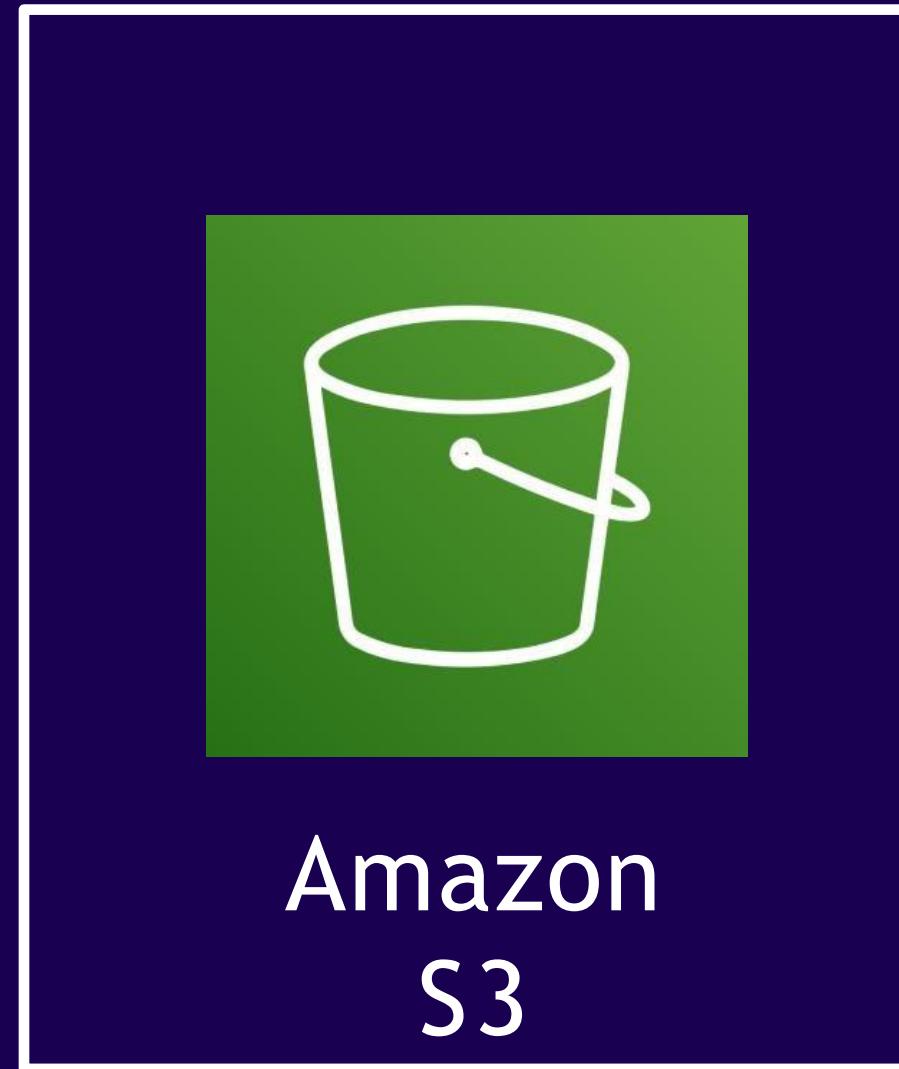
Network-attached volumes that provide durable block-level storage for Amazon EC2 instances



Amazon FSx

Fully managed, cost-effective file storage offering the capabilities and performance of popular commercial and open-source file systems

Amazon S3



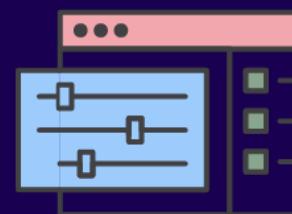
Amazon
S3



Object-level
storage



Designed for
99.99999999%
durability

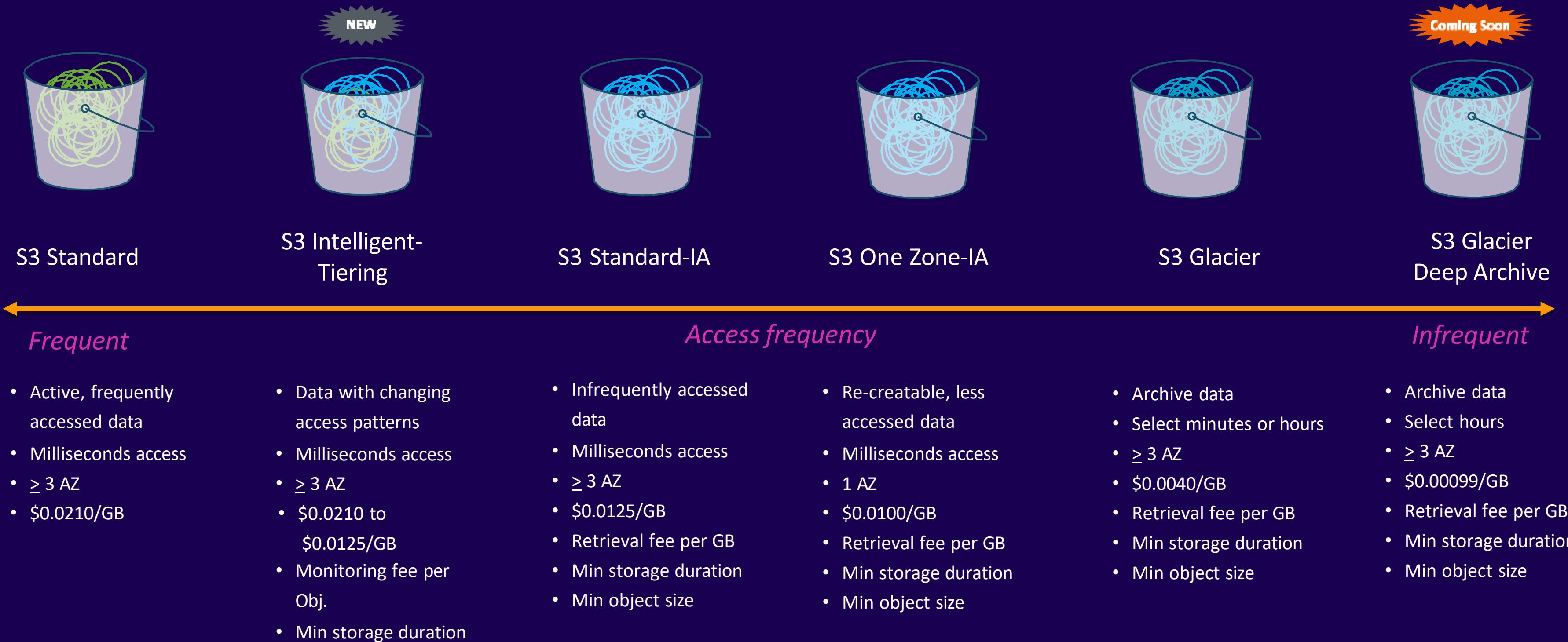


Event triggers

Use cases

- Content storage and distribution
- Backup and archiving
- Big data analytics
- Disaster recovery
- Static website hosting

Your choice of Amazon S3 storage classes

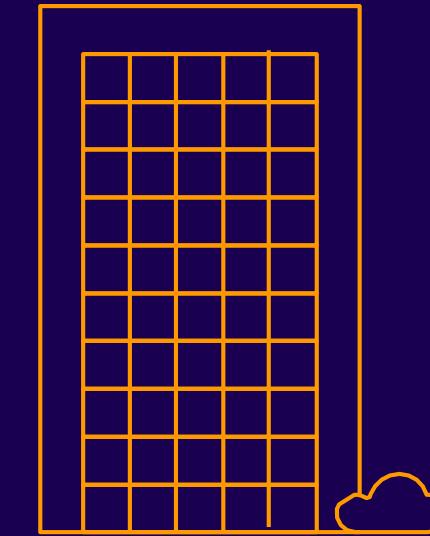


Ideal use cases for S3 Intelligent-Tiering



Big Data, Data Lakes

Storage with changing access patterns used by multiple applications



Enterprises

Storage accessed by fragmented applications from various organizations



Startups

Constraint on resources and experience to optimize storage themselves

Dynamic cost optimization with no performance impact and no operational overhead

File services use cases



Amazon EFS

- Simplify Development Operations (DevOps)
- Modernize application development
- Enhance content management systems
- Accelerate data science



Amazon FSx for Lustre

- Accelerate machine learning
- Enable high performance computing
- Unlock big data analytics
- Increase media workload agility

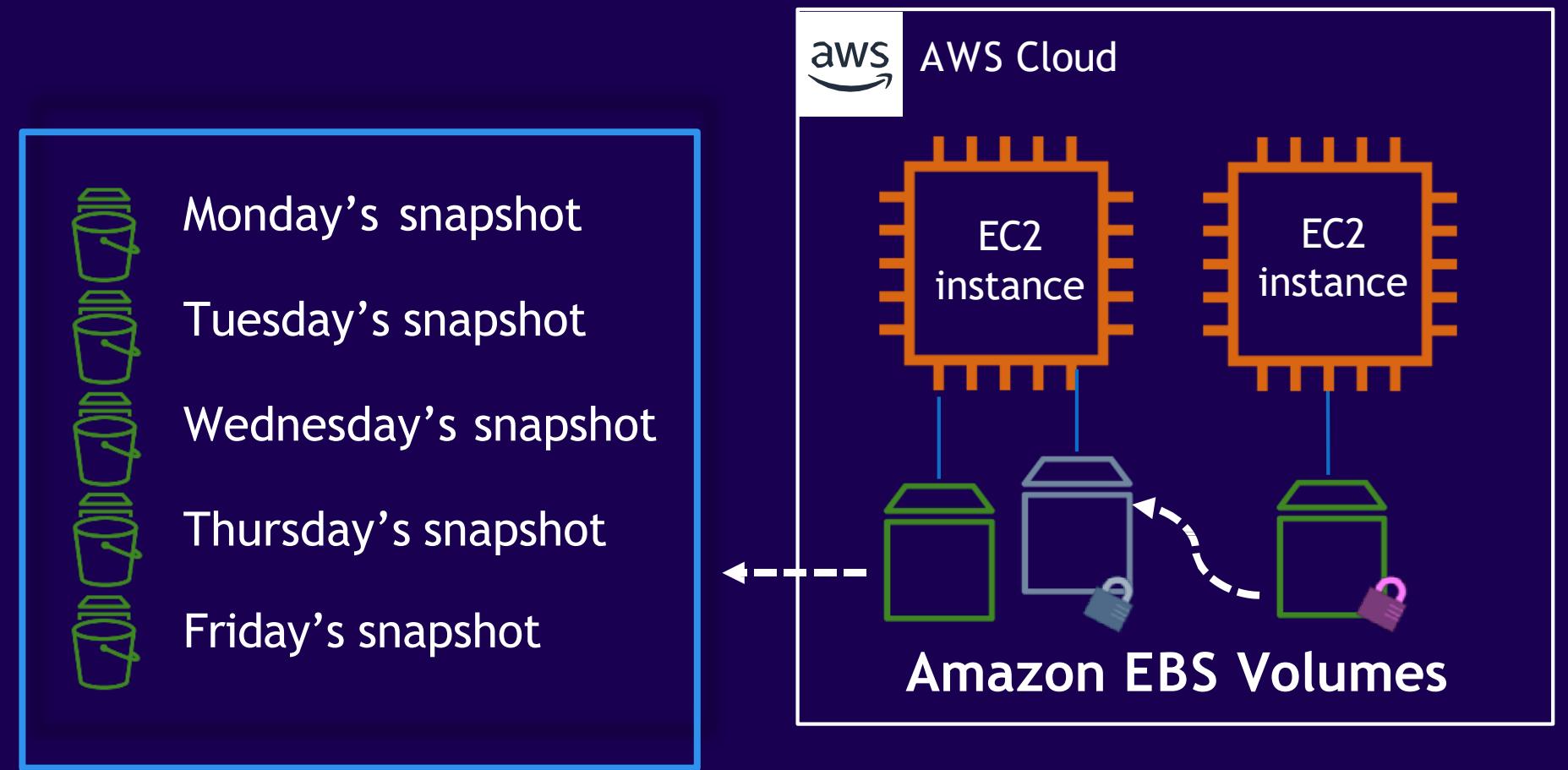


Amazon FSx for Windows

- Migrate Windows file servers to AWS
- Accelerate hybrid workloads
- Reduce Microsoft SQL Server deployment cost
- Simplify virtual desktops and streaming

Amazon Elastic Block Store (Amazon EBS)

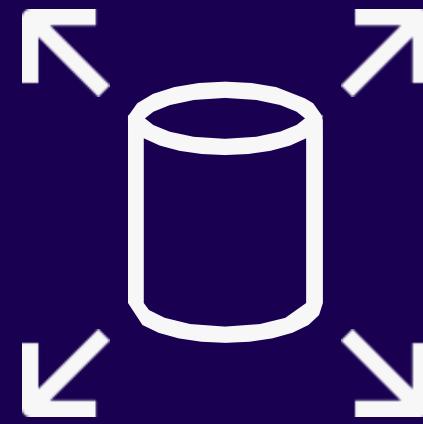
- Persistent block storage for instances
- Protected through replication
- Different drive types
- Scale up or down in minutes
- Pay for only what you provision
- Snapshot functionality
- Encryption available



Create volume snapshots
for backup and recovery

Detach and reattach volumes
to other EC2 instances

Simple, Scalable & Reliable Block Storage Solution



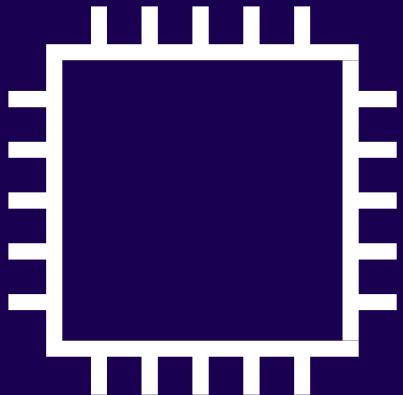
Amazon EBS Volumes

Easy to use, high performance
block storage service



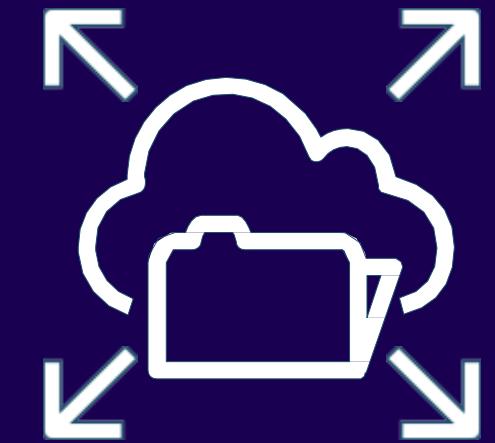
Snapshots

Incremental point-in-time
copies of EBS volumes



Instance storage

Temporary block-level storage
attached to host hardware

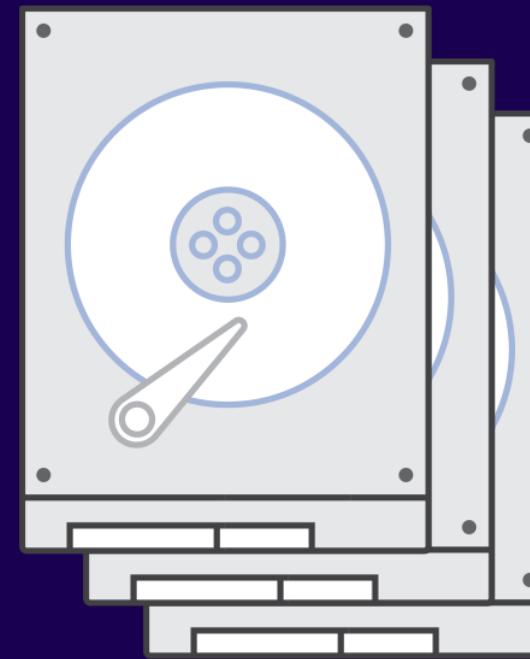


Data Services

Elastic Volumes, Data
Lifecycle Management

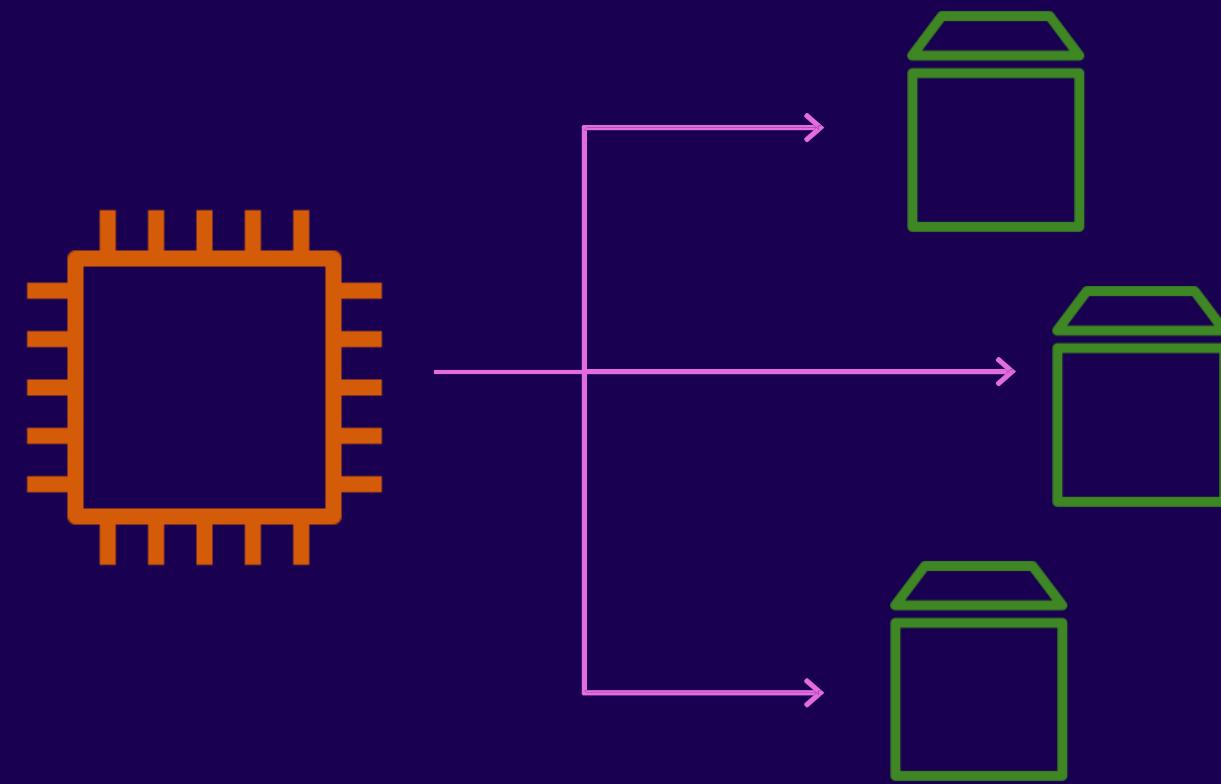
What is EBS?

EBS 



- Network block storage as a service for Amazon EC2 instances
- Data services features
- Large distributed system

EBS in two parts: Data plane



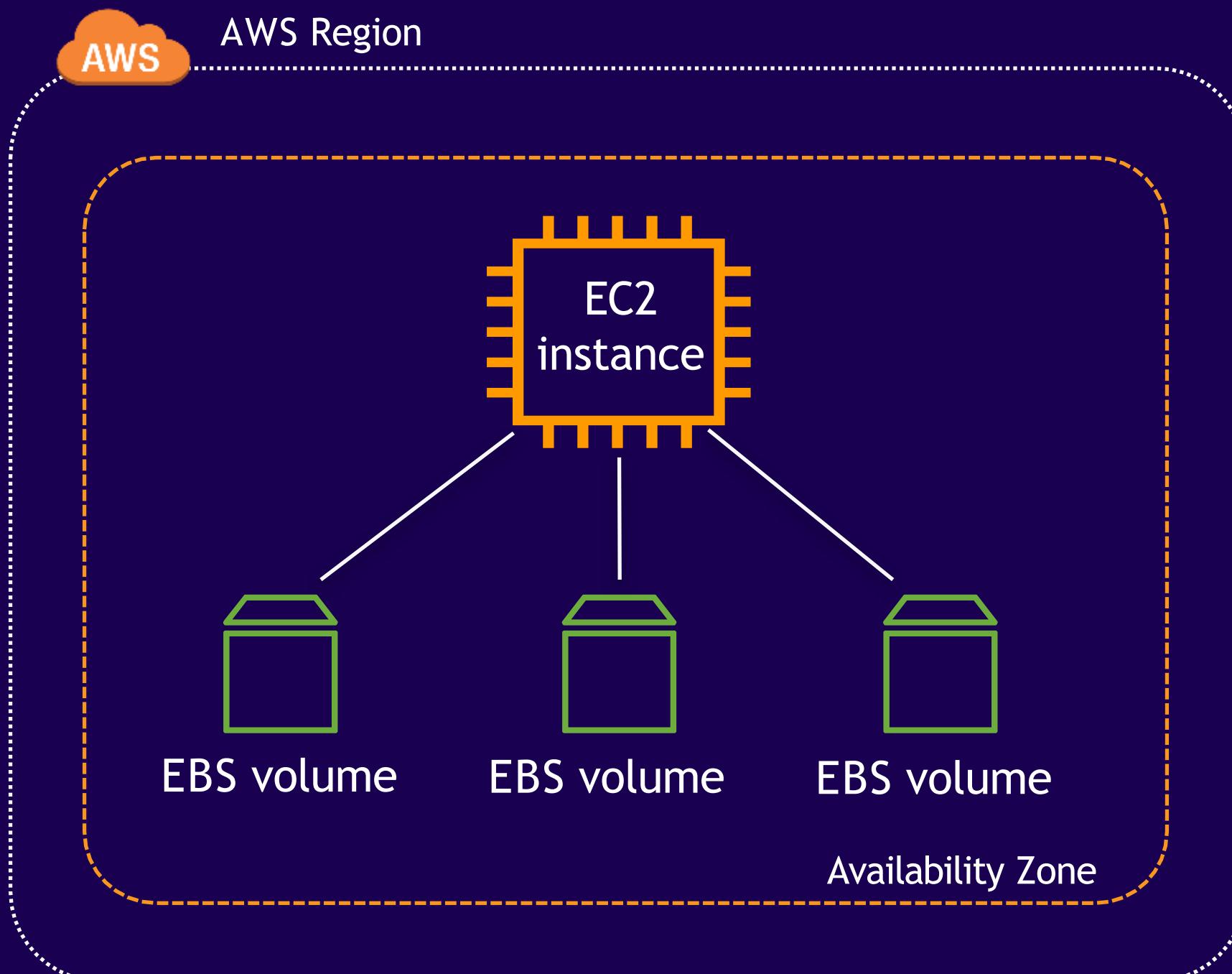
- Compute instance
- Nitro controller
- Network
- Storage servers

EBS in two parts: Control plane



- API front ends
- Volume metadata
- Data placement services
- Repair workflows

Amazon EBS - Block storage as a service



- EBS Volumes attach to EC2 instance
- Many volumes can attach to an EC2 instance
- Volumes persist independent of EC2 instance lifecycle
- New : Multi-attach **io1** volume to up to 16 EC2 instances in select AWS regions

EBS is designed for a wide range of workloads

Enterprise applications



SAP ERP, Oracle
ERP, Microsoft
SharePoint,
Microsoft Exchange

Relational databases



MySQL, PostgreSQL,
SQL Server, Oracle DB,
SAP HANA

Non-relational/ NoSQL databases



Cassandra,
MongoDB, CouchDB

Big data analytics



Kafka, Splunk, Hadoop,
Data warehousing

File/media



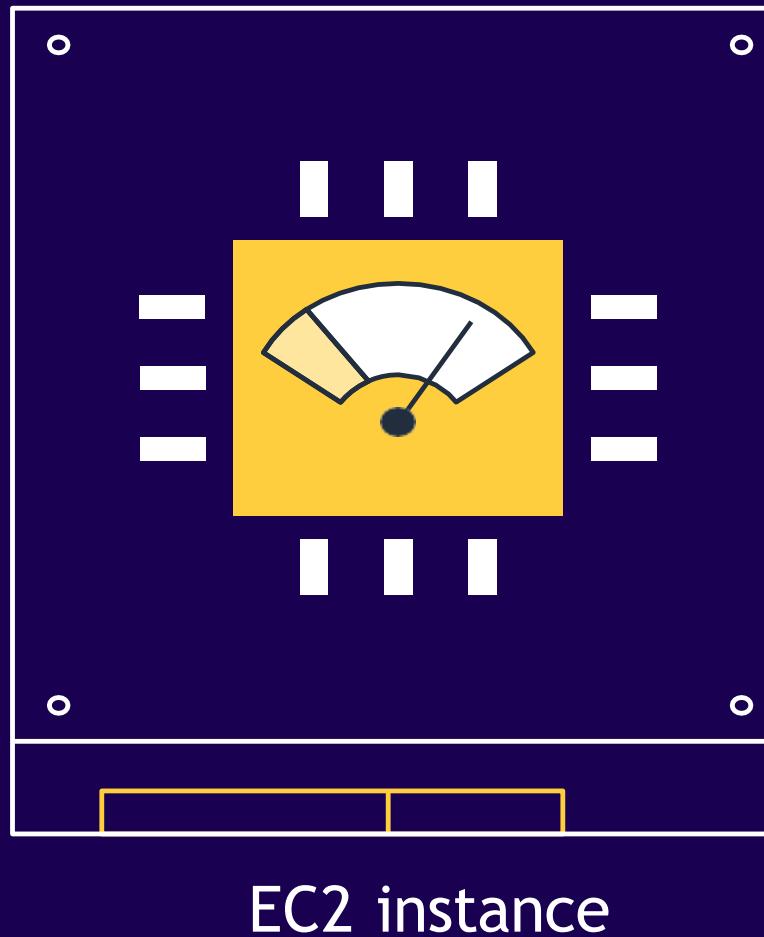
CIFS/NFS, transcoding,
encoding, rendering

LOW LATENCY AND CONSISTENT, HIGH IOPS AND THROUGHPUT

SCALABLE WITHOUT DISRUPTION TO YOUR WORKLOAD

99.999% AVAILABILITY AND AN ANNUAL FAILURE RATE (AFR) OF BETWEEN 0.1% - 0.2%

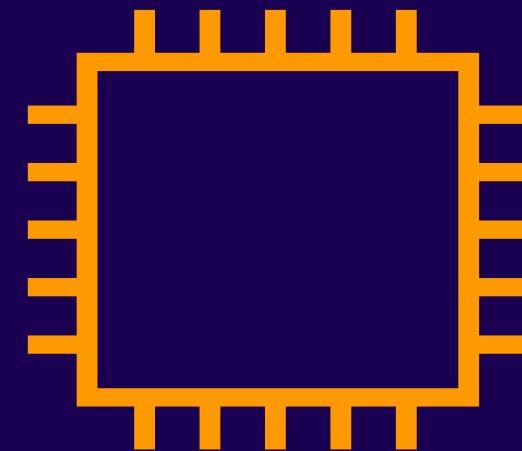
EBS optimized EC2 instances



- Dedicated network bandwidth for Amazon EBS I/O
- Enabled by default on most current-generation instances
- New : Max supported EBS bandwidth for select Nitro instances is now 19Gbps, a 36% increase from 14 Gbps.

Right sizing EC2 for better performance

- EC2 instances have performance thresholds
- EBS volumes have performance thresholds
- The lesser of these will dictate your EBS max performance

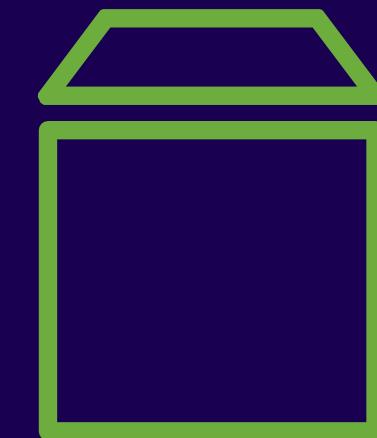


c4.4xlarge

EBS optimized bandwidth

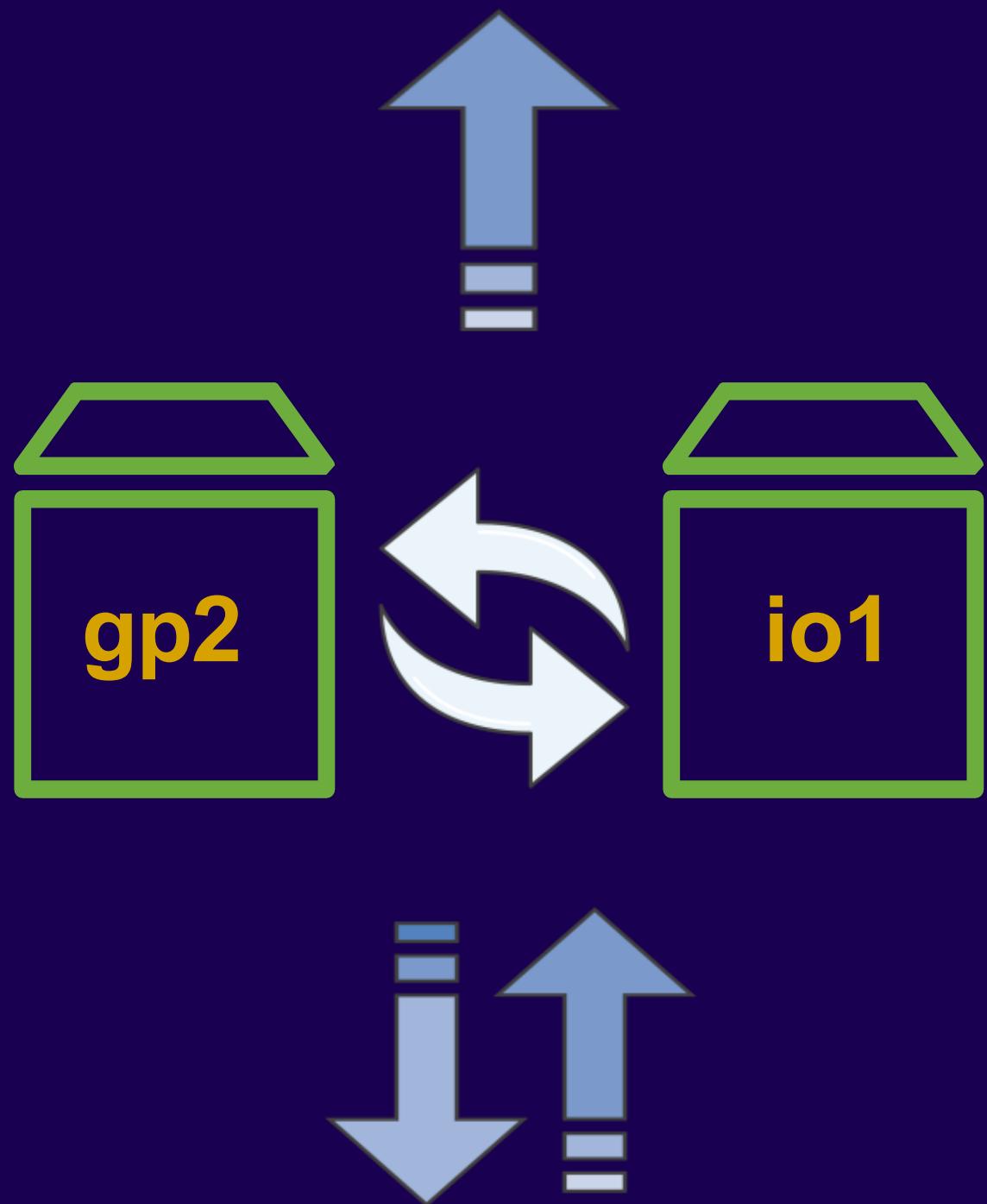
2Gbps ~ 250 MiB/s

16,000 16K IOPS



2 TiB GP2 volume
6,000 IOPS
250 MiB/s max throughput

EBS Elastic Volumes



Increase volume size

Change volume type

Increase/decrease provisioned IOPS

EBS Volumes and Snapshots

An Amazon EBS volume is a durable, block-level storage device that you can attach to your instances

Once attached, you can think as if it's a physical disk.

For current-generation volumes attached to current-generation instance types, you can
dynamically increase size,
modify the provisioned IOPS capacity,
and change volume type on live production volumes.

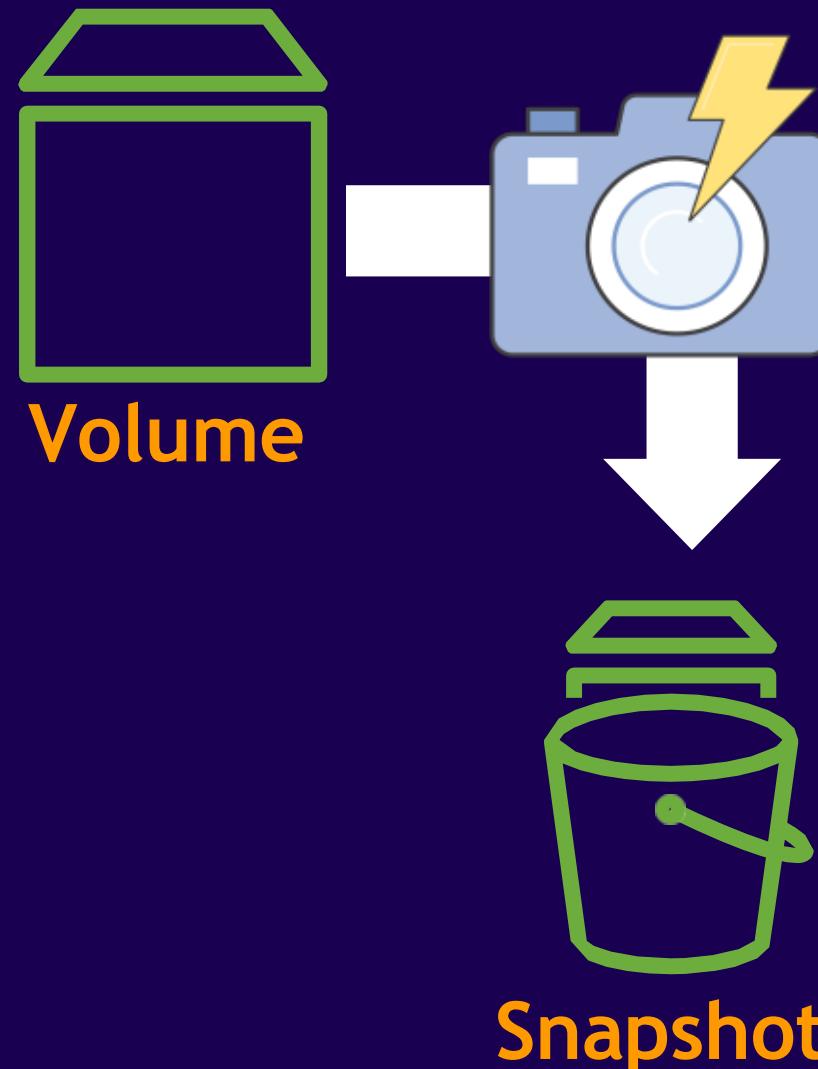
EBS Volumes and Snapshots

An instance can have multiple Volumes attached.

Volume and Instance must be in same AZ.

You could use Multi-Attach to mount volume to multiple instances (depends on volume and instance types)

EBS Snapshots



- Point-in-time copy of an EBS volume
- Incremental - only changed blocks are saved
- Stored in S3 (11x 9's of durability)
- Crash consistent
- Contains all information necessary to restore a volume

EBS Fast Snapshot Restore (FSR)

6x lower recovery time objective

Predictability

Speed

Scale

Cost

Manage RTO based
on size and credits

Instant access to volume
from snapshots

Up to 10 volume restores
instantly

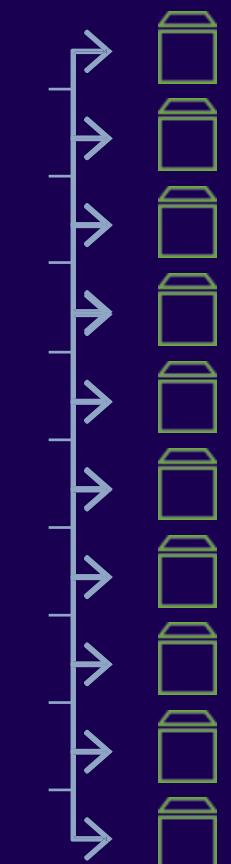
No need for additional
EC2 instances

Fast snapshot restore (FSR)
enabled snapshot



Create regular
snapshot

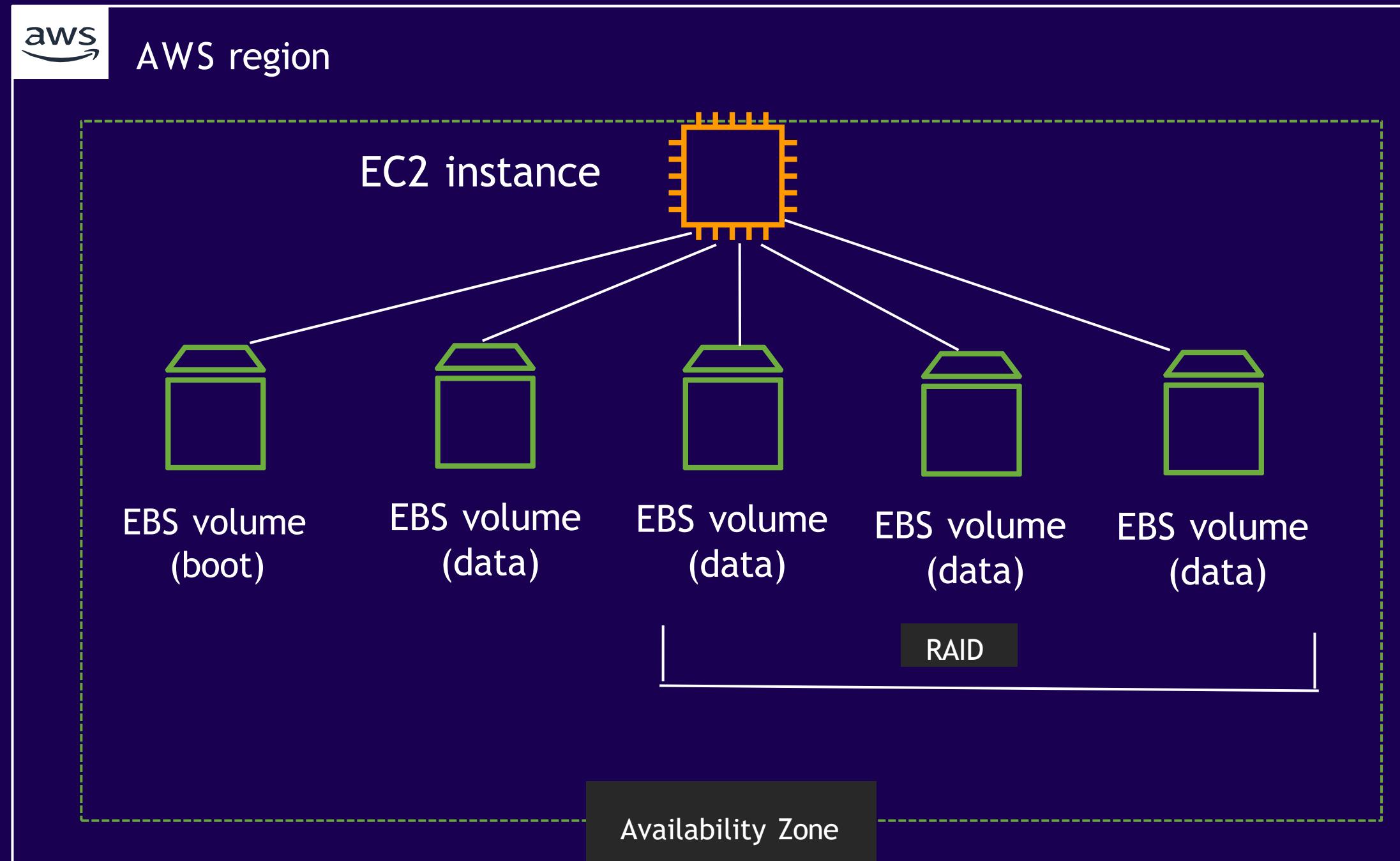
Approximately
60 min. per 1 TiB
of snapshot size



Restore up to 10 volumes
simultaneously

FSR can be enabled at any point
during or after snapshot creation

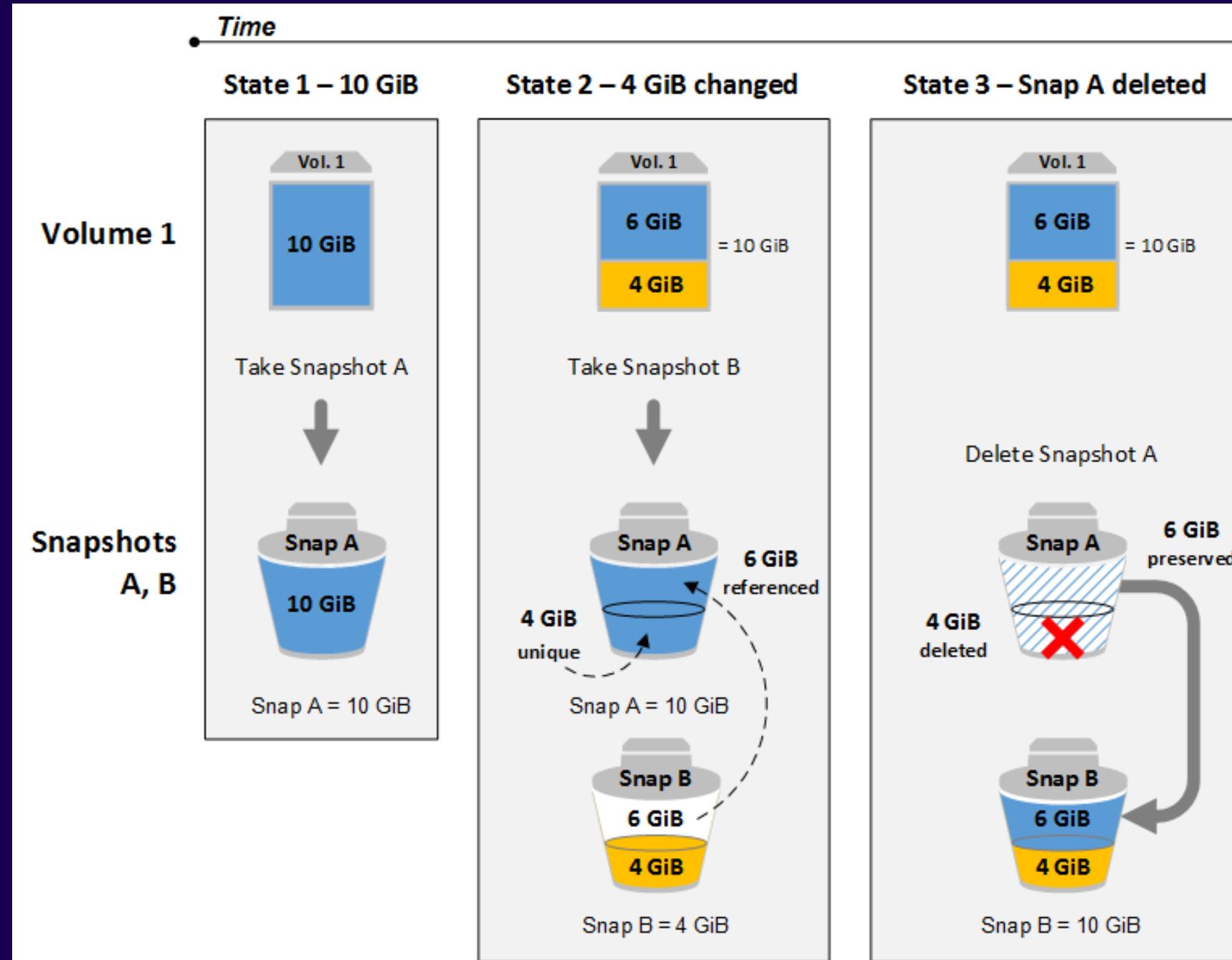
Multi-volume crash consistent Snapshots



Multi-volumes are chosen by specifying an instance
Boot volume can be excluded

DEMO

Deleting EBS snapshots



When a snapshot for a volume is deleted, data referenced exclusively by that snapshot is removed. But data referenced by other snapshots is preserved.

To delete multi-volume snapshots, retrieve all of the snapshots for your multi-volume group using the tag you applied to the group when you created the snapshots. Then, delete the snapshots individually.

Cost Allocation Tags

User-Defined Cost Allocation Tags

✓ Finished loading tags.

Activating tags for cost allocation tells AWS that the associated cost data for these tags should be made available throughout the billing data pipeline. Once activated, cost allocation tags can be used as a filtering and grouping dimension in AWS Cost Explorer, as a filtering dimension in AWS Budgets, and as a dedicated column in the AWS Cost & Usage Report.

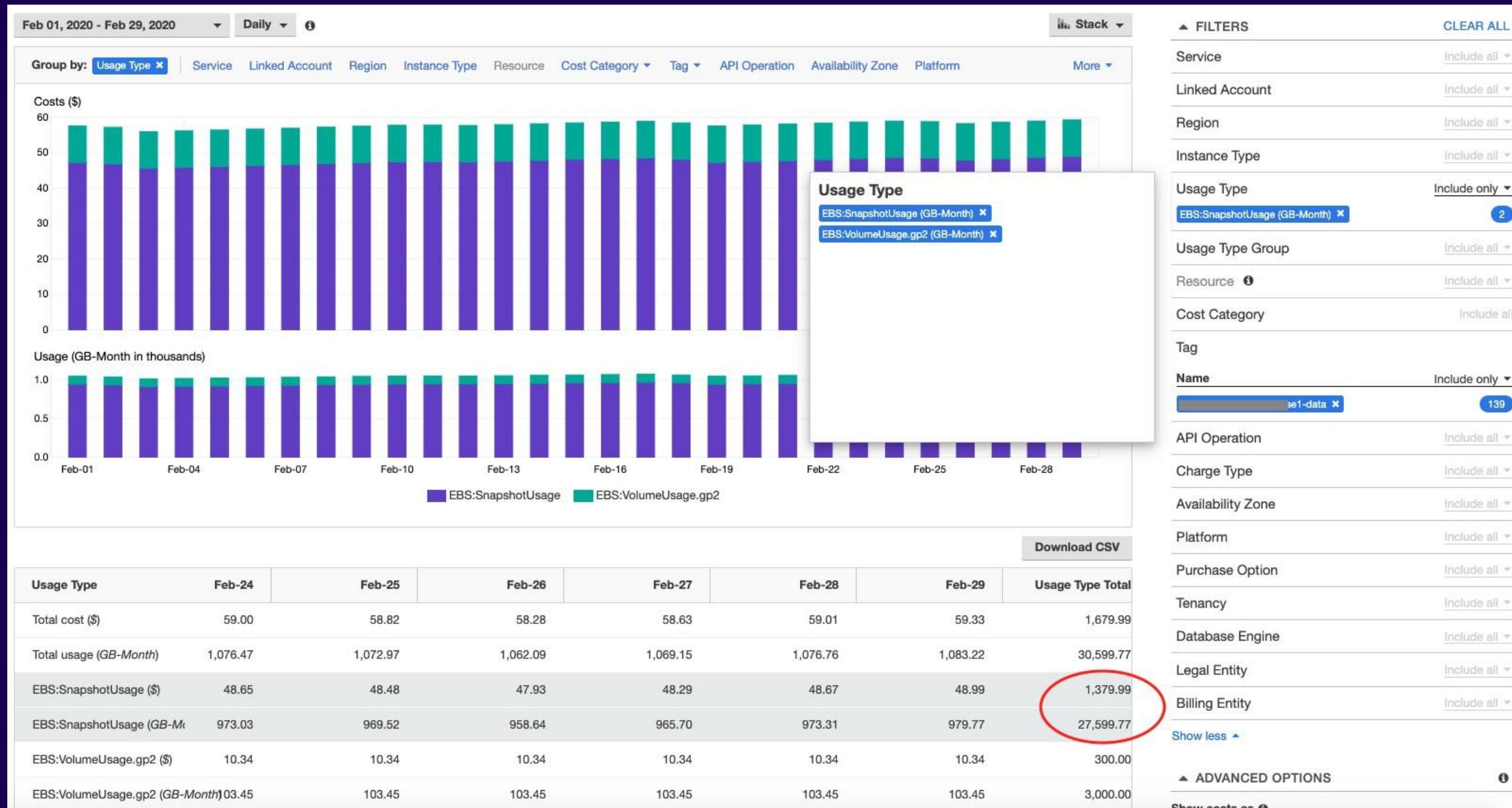
Please note that certain tagged resources (such as accounts) are not billable and will not flow through the billing pipeline, even if they are activated as cost allocation tags. If you would like to create account-level groupings for cost allocation purposes, such as tracking costs by Organizational Unit, please use [AWS Cost Categories](#).

Clicking the Refresh button will prioritize your account for updates, so that tags from your linked accounts are visible to you sooner. Please note that the Refresh operation can only be triggered once every 24 hours.

The screenshot shows a user interface for managing cost allocation tags. At the top, there are three buttons: 'Activate' (blue), 'Deactivate' (grey), and 'Undo' (grey). To the right is a 'Refresh' button. Below these are filter options: 'Filter: All tags' with a dropdown arrow, a search bar 'Search for a tag key...', and 'Tags per page: 100' with a dropdown arrow. The main area displays a table of tags. The first row has a checkbox and the text 'Tag key*'. The second row has a checked checkbox and the text 'Name'. To the right of the table are 'Status' and 'Active' buttons with dropdown arrows. The table has columns for 'Tag key*', 'Name', 'Status', and 'Active'.

Activate user-defined tags for cost allocation

Cost Explorer - EBS & Snapshot Usage & costs

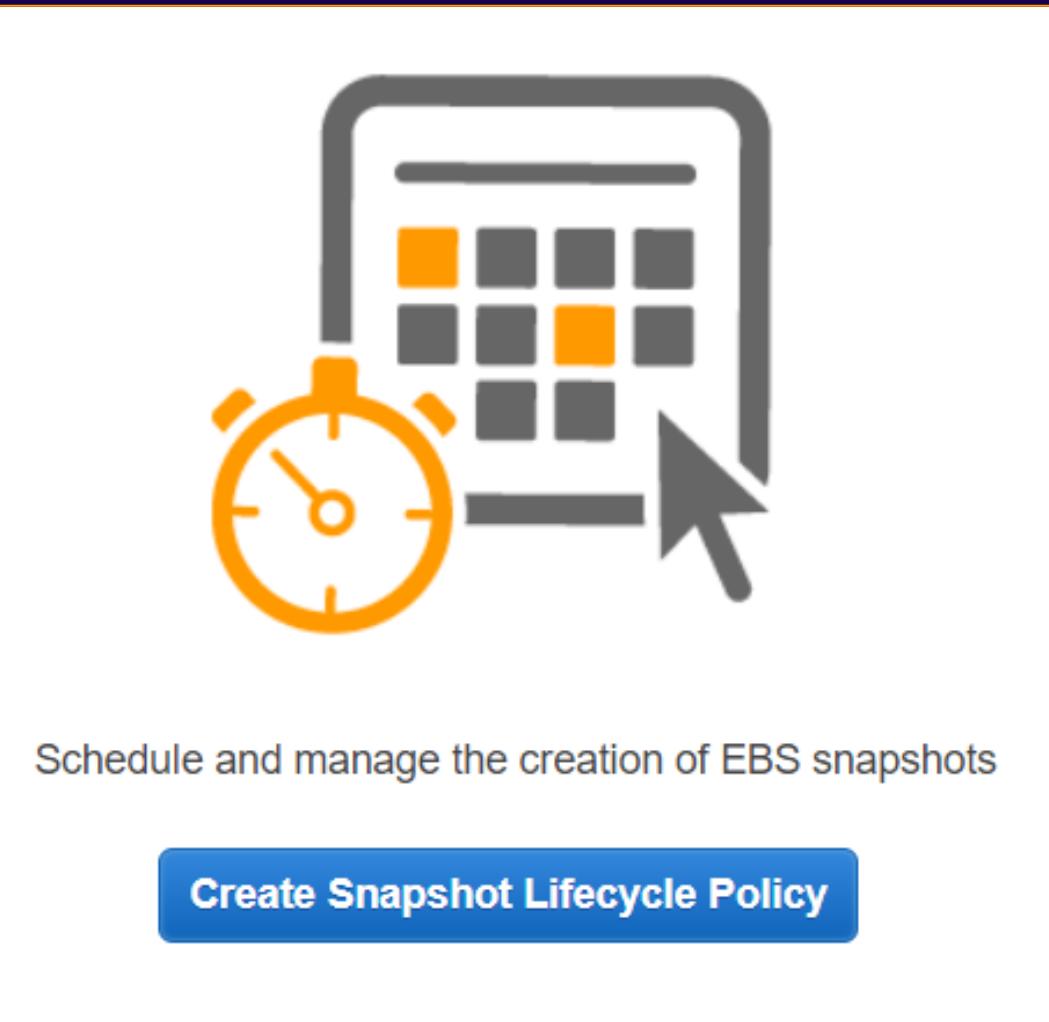


Console view - Usage and costs broken down by “Name” tag value for an EBS Volume and its associated snapshots

Same data you can extract programmatically using Cost Explorer **get-cost-and-usage APIs** or **Cost and Usage Report (CUR)**

Amazon Data Lifecycle Manager (DLM)

- Policy and tag-based snapshot management solution
- Automated scheduling
- Automated snapshot retention management
- No cost to use
- Cross region copy of snapshots to meet DR and compliance
- Set your remote region retention independent of source region policy



EBS Notes

Volume Type	Min GB	Max GB	IOPS
SSD – GP2	1	16384	16000 for Size > 5333 G Min 100 Baseline 3 / GB Burstable 3000
SSD – GP3	1	16384	Min 3000 Max 16000 Throughput 125 Mbs/s to 1000 Mbs/s
Provisioned IOPS (io1)	4	16384	Min 100 Max 64000 Upto 50 / GB (4 GB Disk == 200 IOPS)
Provisioned IOPS (io2)	4	65536	Min 100 Max 256000 Upto 1000 / GB
Cold HDD (sc1)	125	16384	12 MBs/TB,
Throughput Optimized HDD (st1)	125	16384	40 Mbs/TB

EBS Notes

https://aws.amazon.com/blogs/aws/new-ebs-volume-type-io2-more-iops-gib-higher-durability/?source=post_page-----7177e59fff3c-----



- August 2012
 - 1,000 IOPS per EBS volume.
 - 2,000 IOPS per EBS volume.
 - 4,000 IOPS per EBS volume.
 - Up to 256K per I/O request (16x larger).
- November 2012
- May 2013
- August 2014
- March 2015
- December 2017
- November 2018
 - 20,000 IOPS per EBS volume.
 - 32,000 IOPS per EBS volume.
 - 64,000 IOPS per EBS volume.

io2 in 2020

Which one???

Depends.

What's the nature of your application and amount of data?

How much is your budget?

Which one? - Assume 200 GB

Volume Type	IOPS	Cost	Cost / Month
SSD – GP2	Max(100 , 200*3) = 600	USD 0.1/GB	200 * .1 = 20 USD
SSD – GP3	3000 (125 Mb/s)	USD 0.08/GB	0 (IOPS) + 16 (Storage) + 0 (throughput= 16 USD)

Repeat Same Exercise for provisioned IO

Factor in Snapshot, Backup and Recovery costs.

With no snapshots/recovery, 200 GB io2 with 3000 IOPS – USD 220/Month

With no snapshots/recovery, 200 GB io1 with 3000 IOPS – USD 220/Month

With no snapshots/recovery, 200 GB gp2 with 600 IOPS – USD 20/Month

With no snapshots/recovery, 200 GB gp3 with 600 IOPS – USD 16/Month

In summary

Use right choice of EBS volume type to match your application need

Use the EBS Optimized EC2 instances to prevent performance bottlenecks

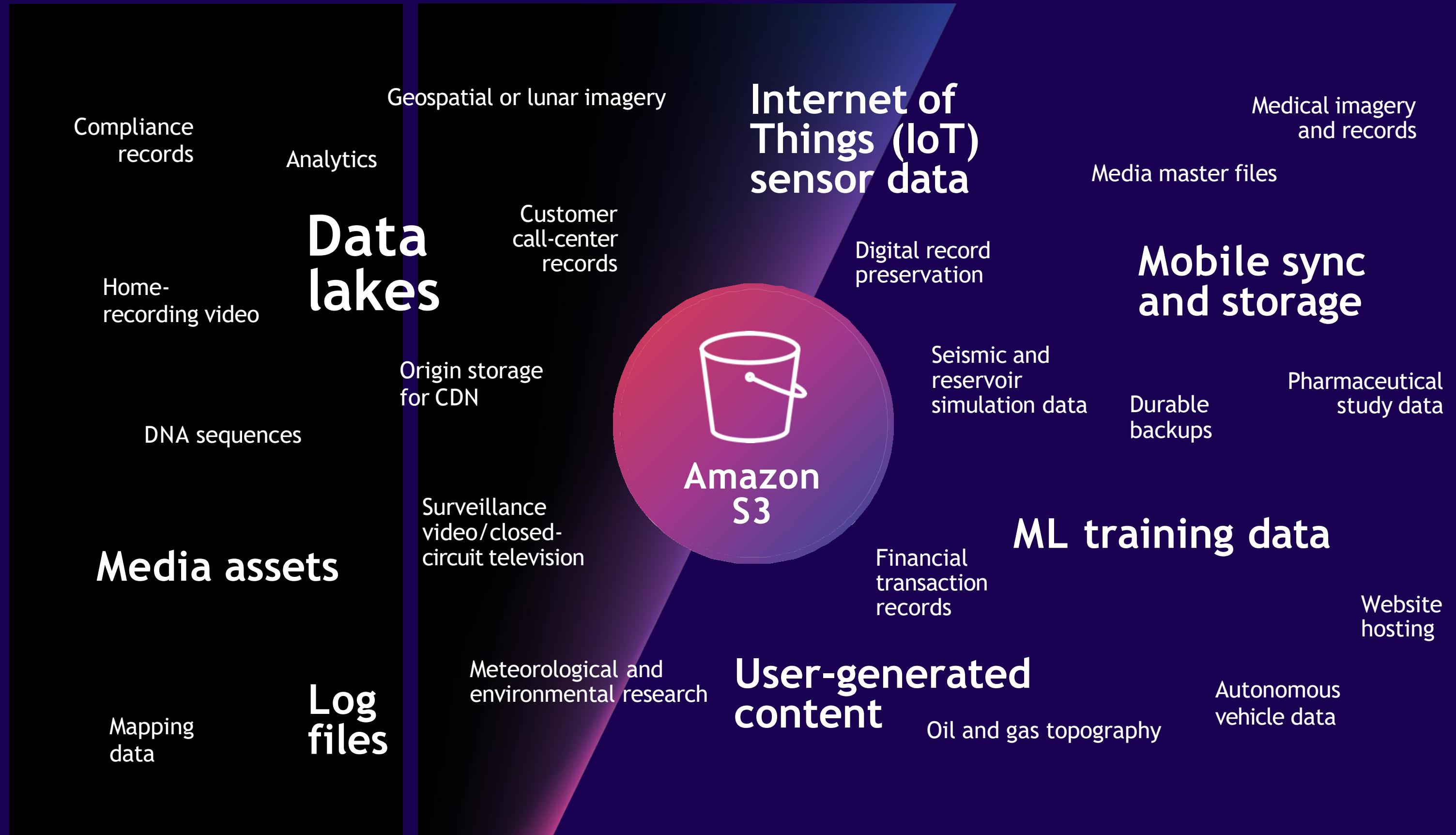
Tag everything & establish cost visibility

Understand where the low-hanging fruit is in your case for optimization

Leverage tools for finding unused resources and optimize cost

Increase elasticity to avoid waste

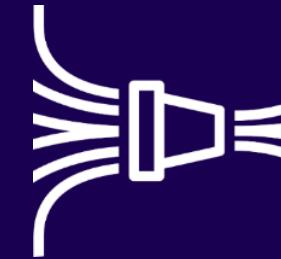
AWS S3



Amazon S3 has more options for data transfer



AWS
Direct Connect



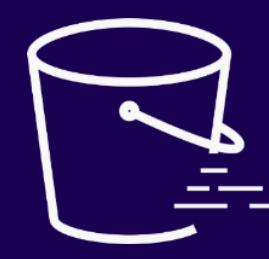
Amazon
Kinesis Data
Firehose



Amazon Kinesis
Data Streams



Amazon Kinesis
Video Streams



Amazon S3
Transfer
Acceleration



AWS
Storage
Gateway



AWS
Snowball



AWS
Snowball Edge



AWS
Snowmobile



AWS
DataSync

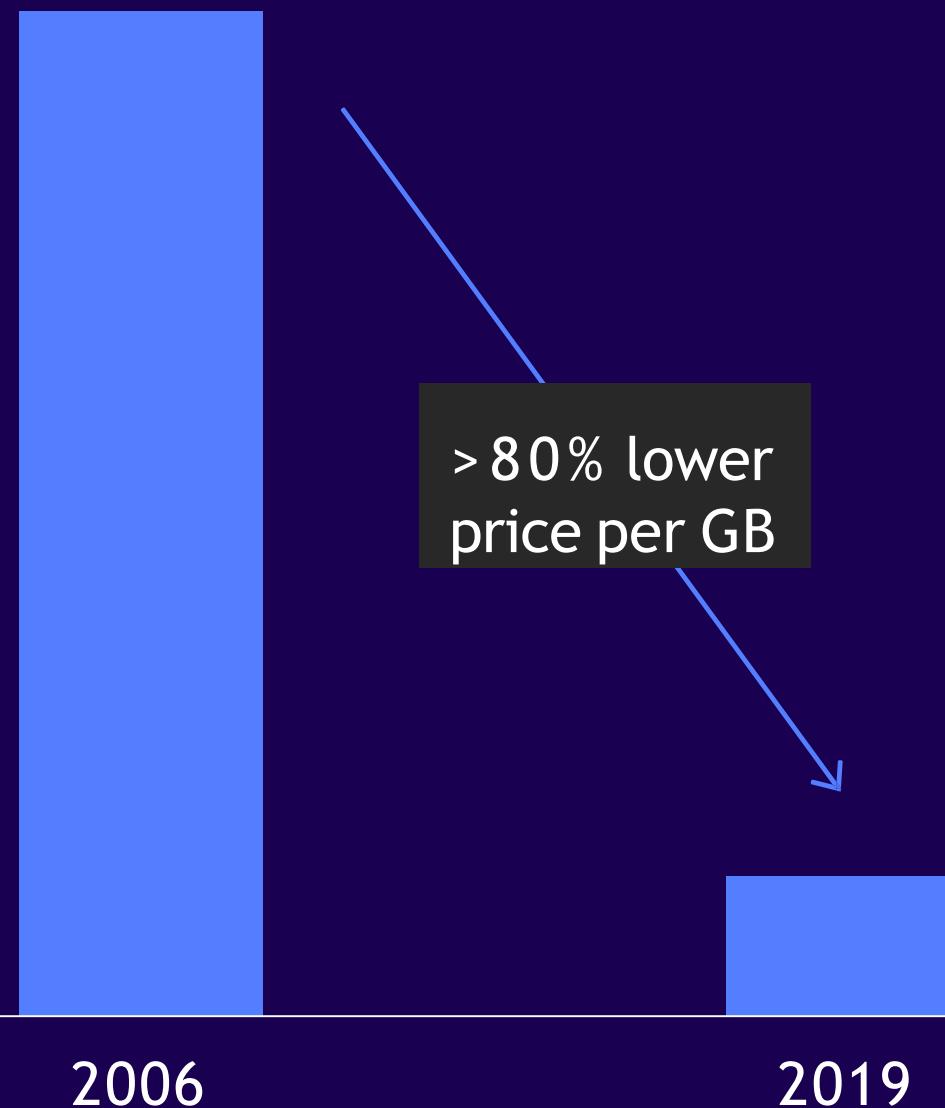


AWS SFTP

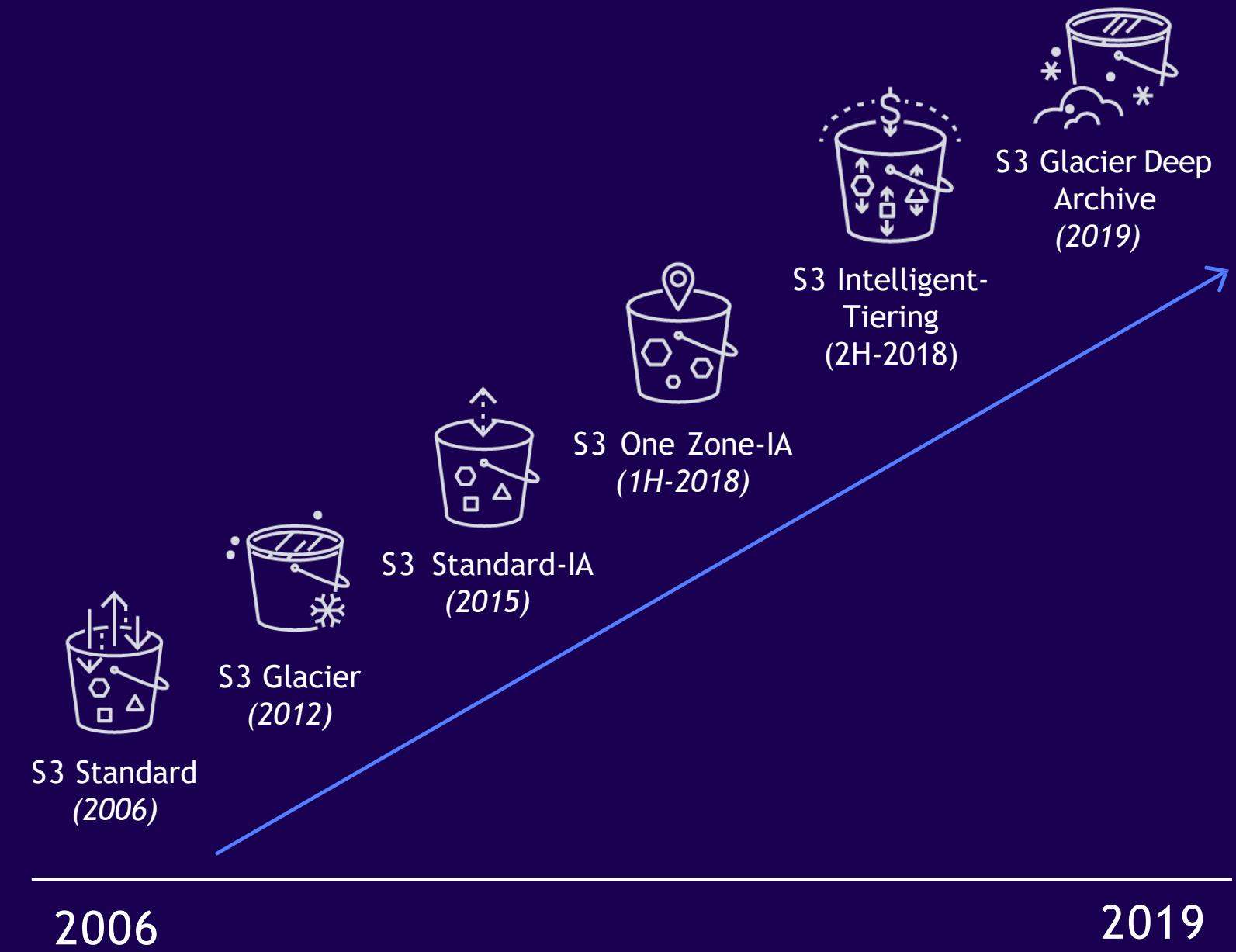
Amazon S3 storage classes

Optimize your storage cost by utilizing all Amazon S3 storage classes

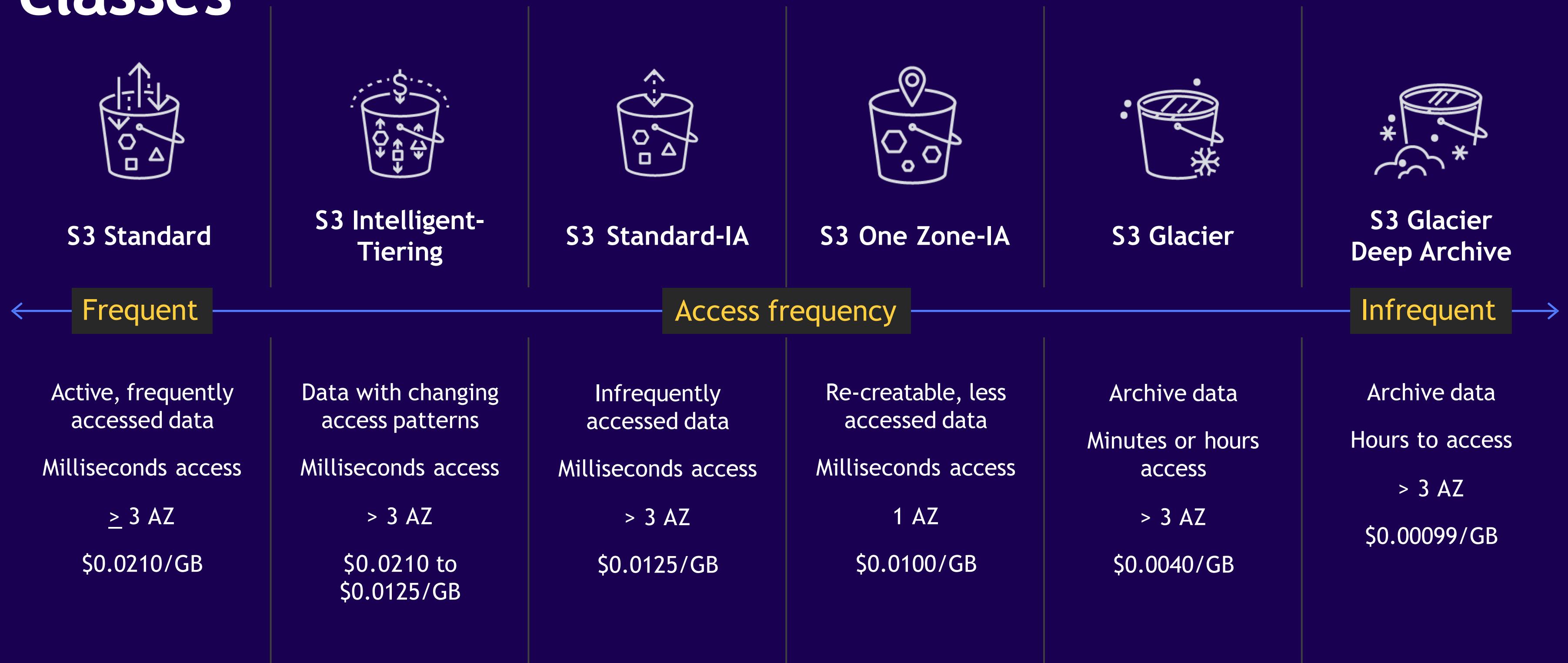
Decreasing storage prices



Accelerating innovation



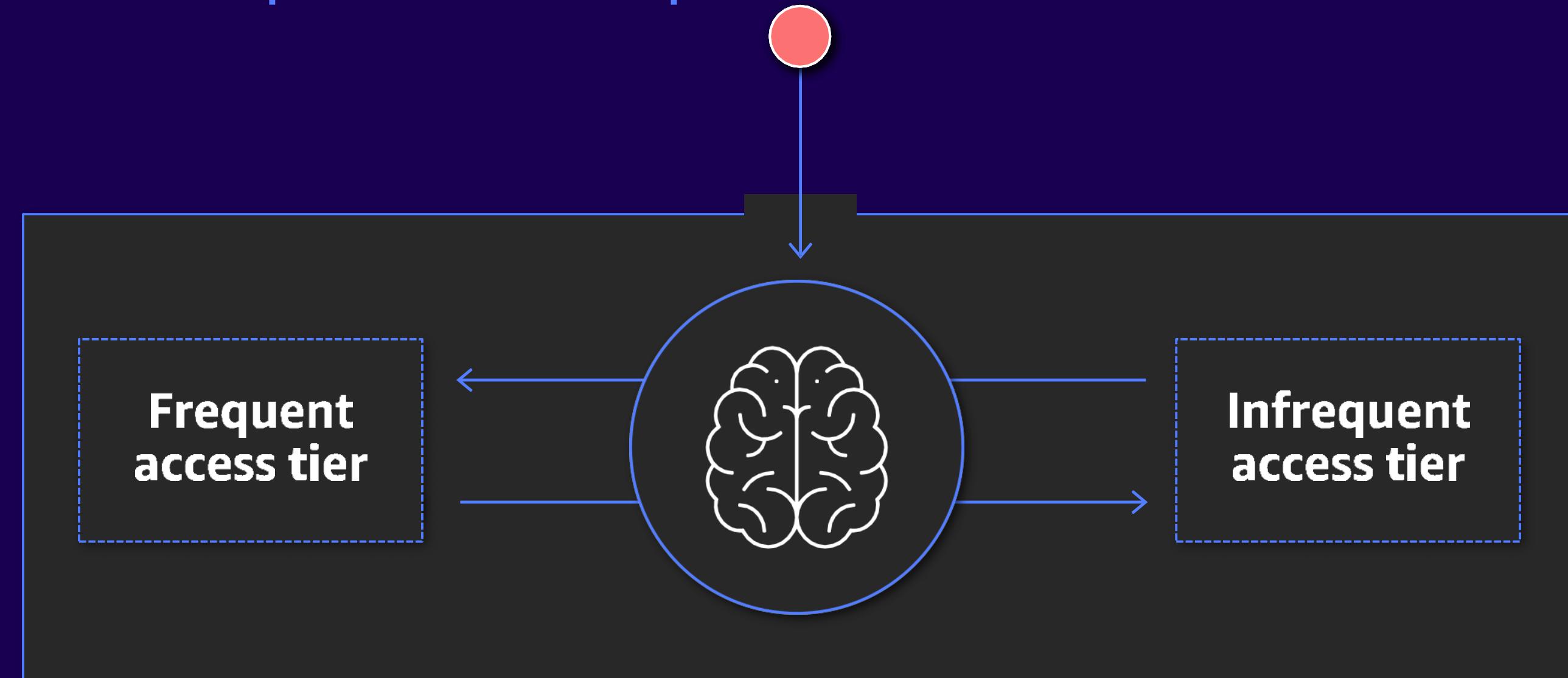
Your choice of Amazon S3 storage classes



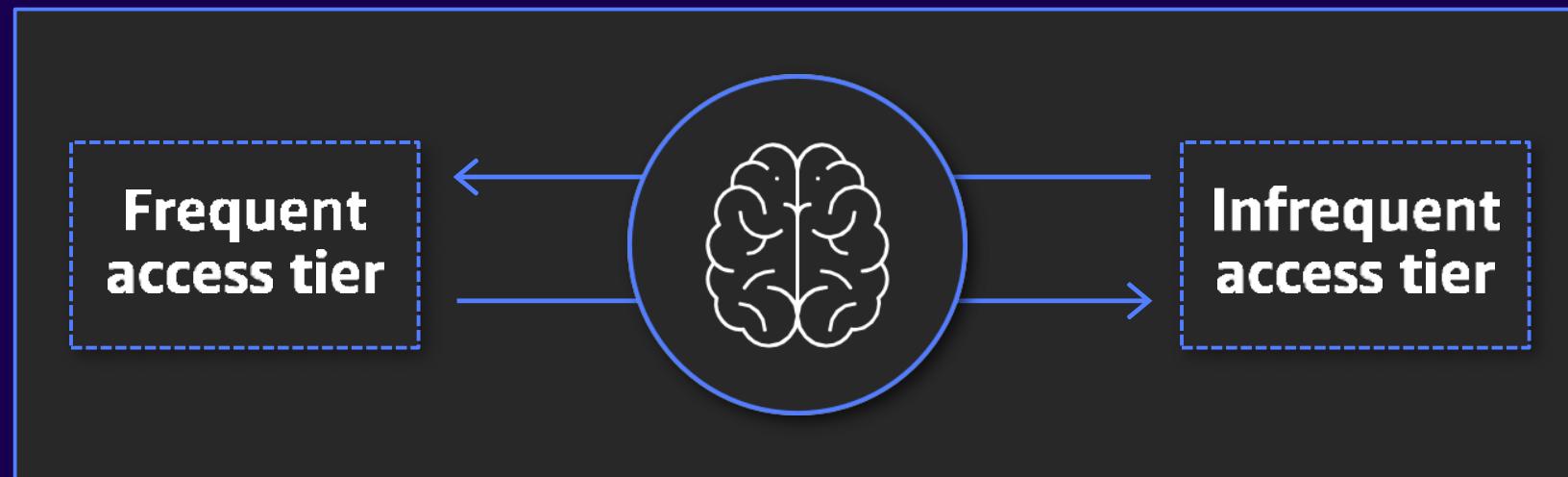
Amazon S3 Intelligent-Tiering



Automatic cost optimization with
no performance impact and no operational overhead



Amazon S3 Intelligent-Tiering automates cost savings



Automatically optimizes storage costs for data with changing access patterns

Stores objects in **two access tiers**, optimized for frequent and infrequent access

Monitors access patterns and optimizes cost on granular object level

No performance impact, no operational overhead, no retrieval fees

Customers of all sizes and virtually every industry use S3 INT and save automatically

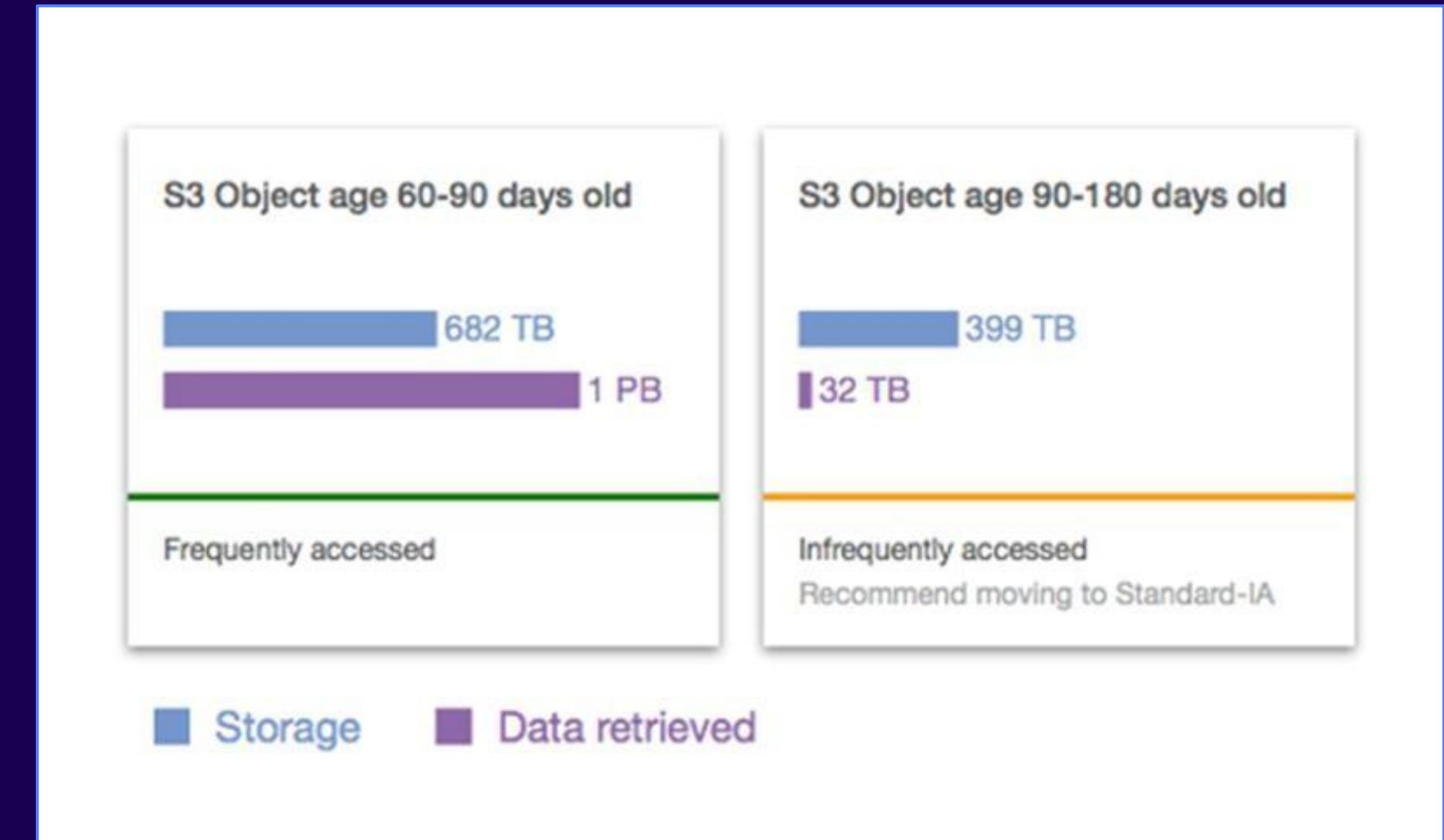
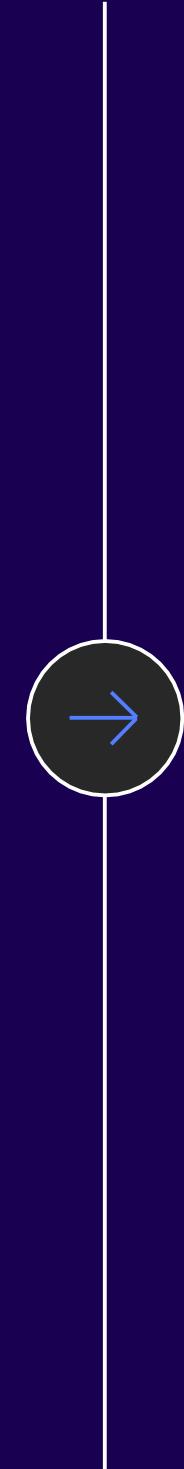
S3 Storage Class Analysis

Provides lifecycle policy recommendations based on access patterns

Monitors access patterns

Classifies data as frequently or infrequently accessed

Can be filtered by bucket, prefix, or object tag



Lifecycle policies use rules to manage your storage

Use lifecycle policies to transition objects to another storage class

Lifecycle rules take action based on object age. Here's an example:

1. Move objects older than 30 days to S3 Standard – Infrequent Access
2. Move objects older than 365 days to S3 Glacier Deep Archive



S3 Standard



S3 Standard –
Infrequent Access



S3 Glacier
Deep Archive

Object tags work with lifecycle policies

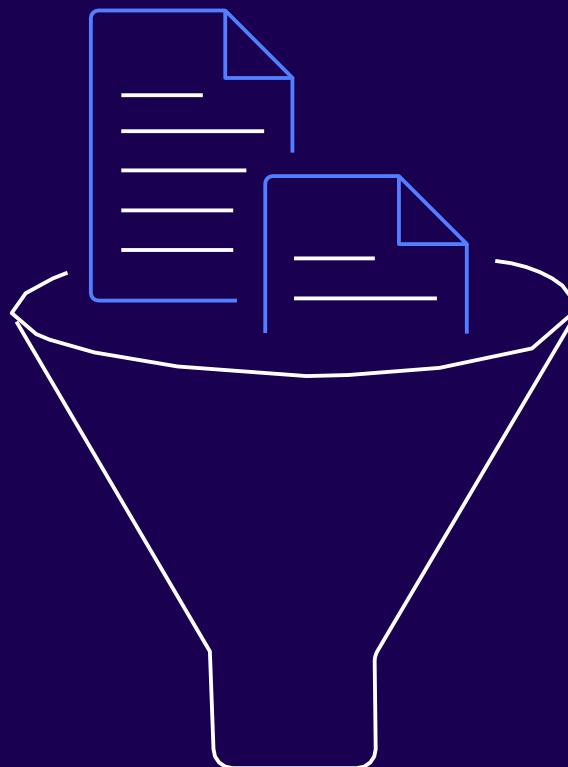
Perform automated actions on a subset of your data with object tags

Lifecycle

Specify a tag filter to transition or expire objects

Use S3 batch operations to apply object tags at scale

Ex: Transition all objects tagged “Project : Delta” to S3 Glacier



Use lifecycle policies with object tag filters

Object tag filters simplify lifecycle policies when the same action needs to be performed across multiple prefixes in the bucket

```
<Filter>
  <And>
    <Tag>
      <Key>Project</Key>
      <Value>Delta</Value>
    </Tag>
  </And>
</Filter>
```

Performance best practices on Amazon S3

Use the latest version of the AWS SDKs to automatically see performance improvements from:

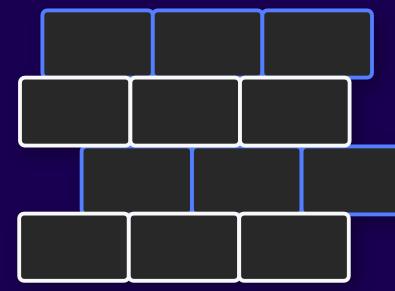
- Automatic retries
- Handling timeouts
- Parallelized uploads and downloads with TransferManager

See the [Optimizing Amazon S3 Performance whitepaper](#) to learn more about:

- Scaling horizontally for more throughput
- Caching data
- Using Amazon S3 Transfer Acceleration for faster data transfer

Security is at the core of S3

Data stored in Amazon S3 is secure by default



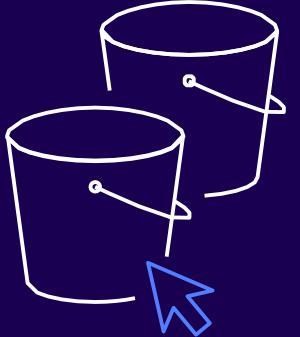
Amazon S3
Block Public
Access



Encrypt data by
default in
Amazon S3



Encryption
status in Amazon
S3 inventory

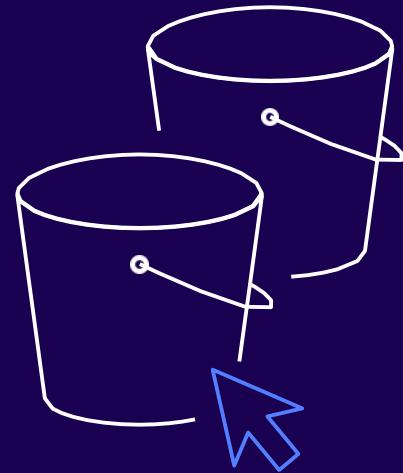


Bucket
permission
checks



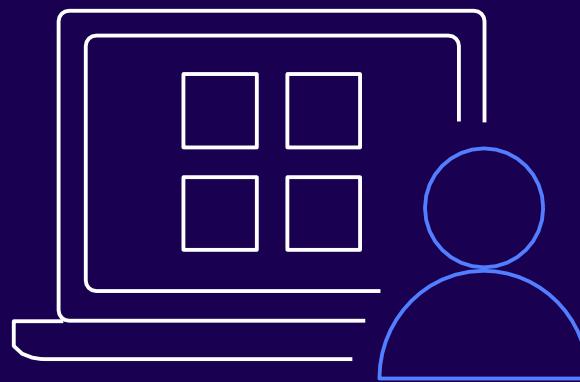
Free checks
with AWS
Trusted Advisor

Layers of access control



Resource-based

- Object Access Control Lists (ACLs)
- Bucket Access Control Lists (ACLs)
- Bucket policies

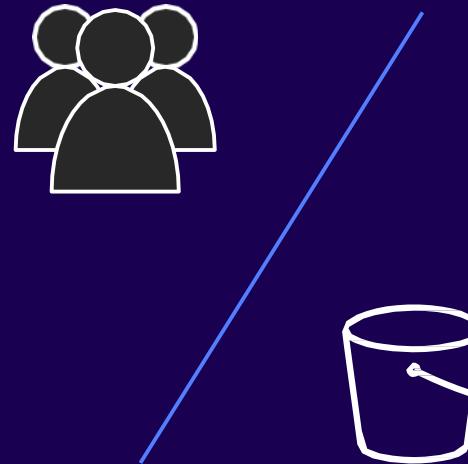
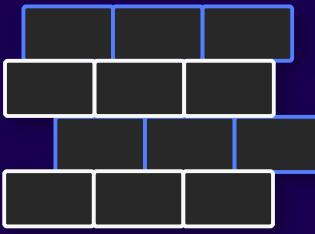


User-based

- Identity and Access Management (IAM) policies

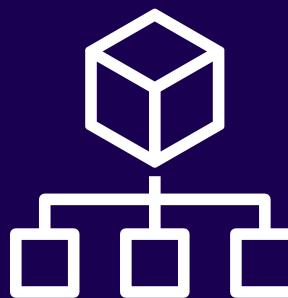
Amazon S3 recommends using bucket policies and IAM policies

Amazon S3 Block Public Access



Can be applied to accounts or buckets

Four security settings to deny public access



Use AWS Organizations Service Control Policies (SCPs)
to prevent settings changes

Amazon S3 Block Public Access settings

Block *all* public access

On

– Block public access to buckets and objects granted through *new* access control lists (ACLs)

On

– Block public access to buckets and objects granted through *any* access control lists (ACLs)

On

– Block public access to buckets and objects granted through *new* public bucket policies

On

– Block public and cross-account access to buckets and objects through *any* public bucket policies

On

Amazon S3 default encryption



One-time
bucket-level
setup



Automatically
encrypts all new
objects



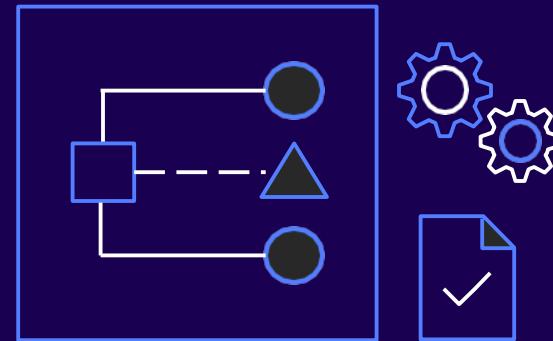
Simplified
compliance



Supports SSE-S3
and SSE-KMS

Provides Amazon S3 encryption-at-rest support for applications that do not otherwise support encrypting data in Amazon S3

Access Analyzer for Amazon S3 buckets



Continuous analysis

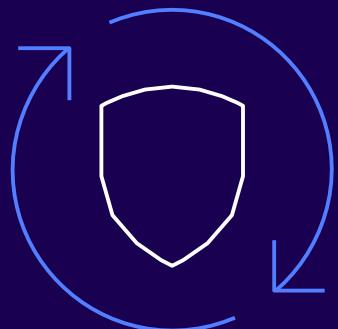
Continuously monitors and automatically analyzes resources

Surfaces buckets with public & shared access in the S3 Management Console



Provides insights

Drilldown into source and level of public and shared access



Swift remediation

Lock public buckets down with a single click

Acknowledge shared access as intended

Know exactly where and what remediation actions to apply

Amazon S3 Access Points

Amazon S3 Access Points simplify access control for large, shared buckets such as data lakes

Every application that interacts with a multi-tenant bucket can have a dedicated access point with custom permissions

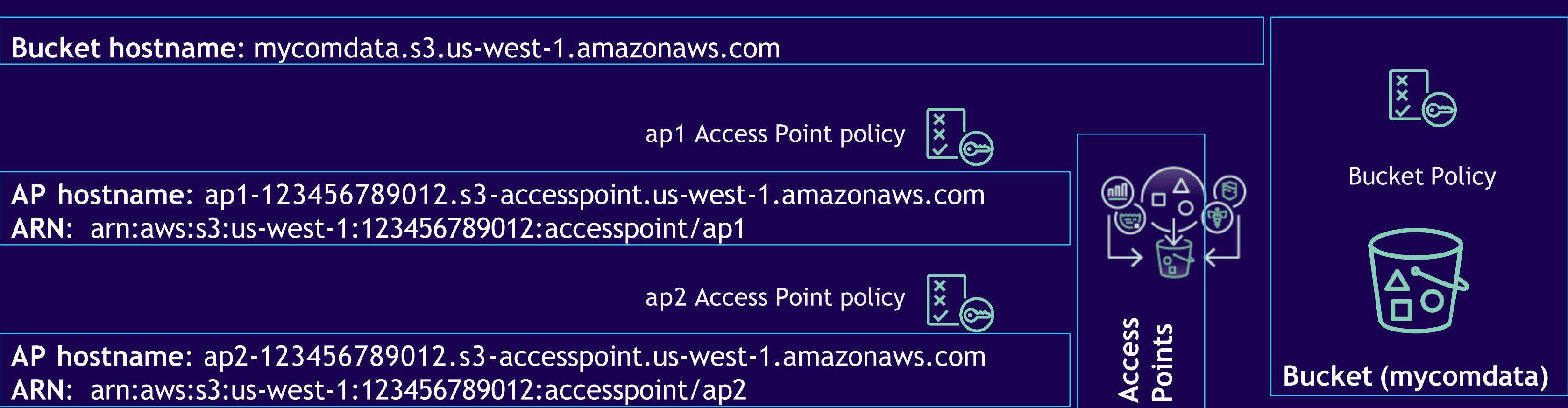
Amazon S3 Access Points can be set to only allow access from a Virtual Private Cloud (VPC)

VPC access points do not allow requests from the Internet. S3 restricts request traffic to the specified VPC

What is an Amazon S3 Access Point?

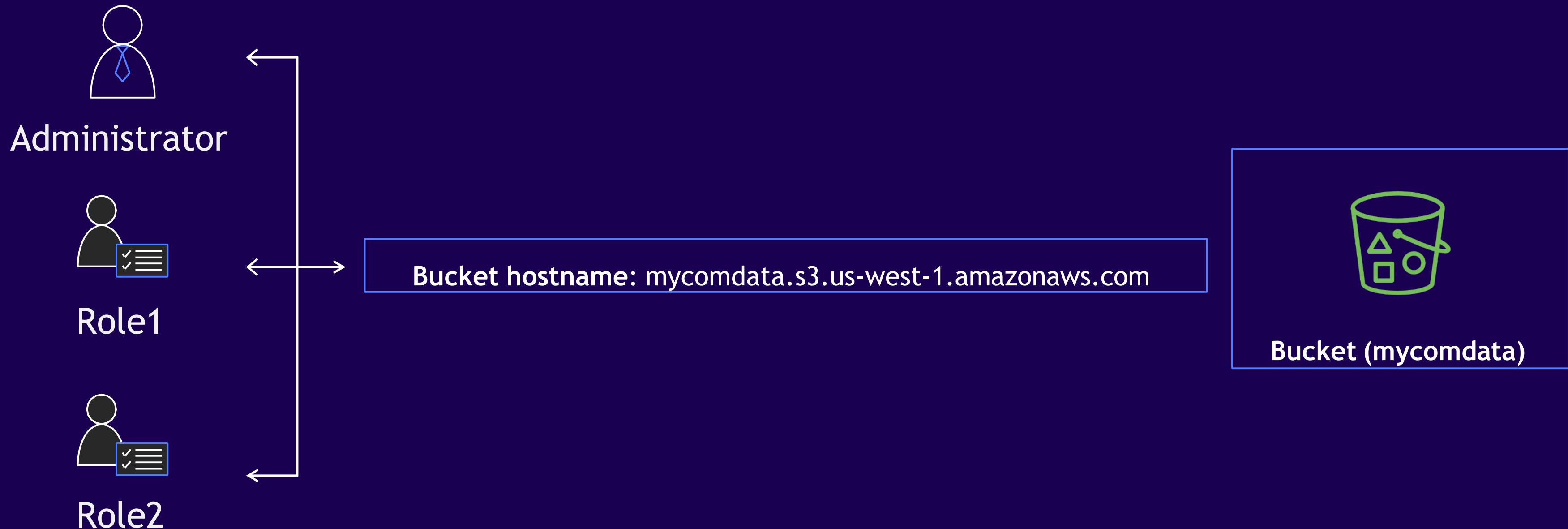
A new S3 resource with a hostname, ARN and an IAM resource policy

- Applications use Access Points to access objects in a bucket
- Access Points can be limited to a specified VPC
- Access Points have a Access Point specific Block Public Access setting
- Access Point names live in a private namespace that is unique to an account and the region
- Access Point ARNs and hostname have the account ID and region embedded in them



Accessing objects in Amazon S3 – Previously

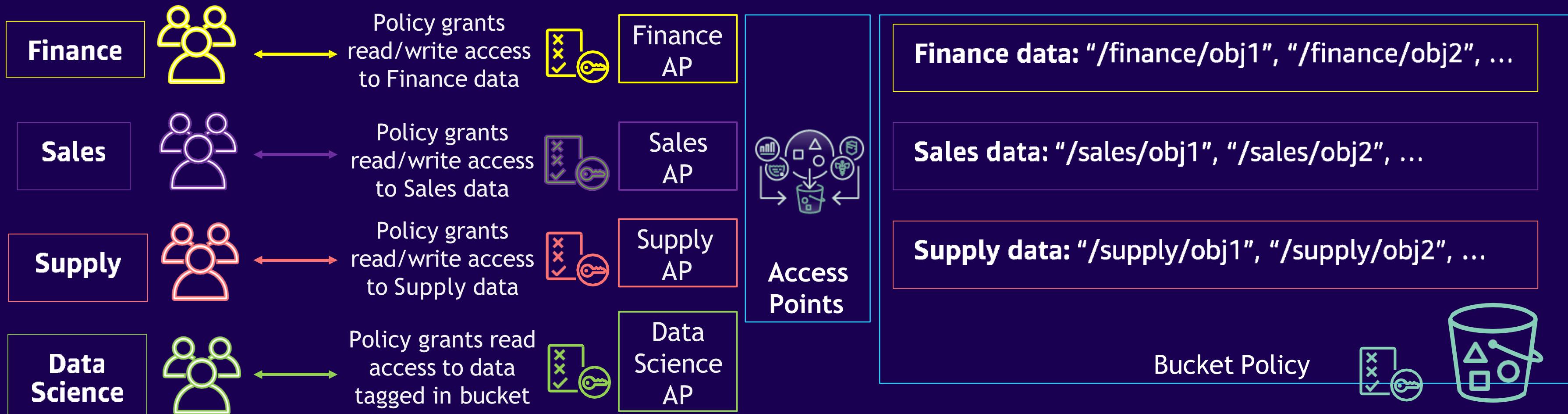
All users would access objects directly through the bucket using the bucket hostname



Use case: simplify access control for shared buckets

Now, we can grant custom access to multiple teams using Access Points

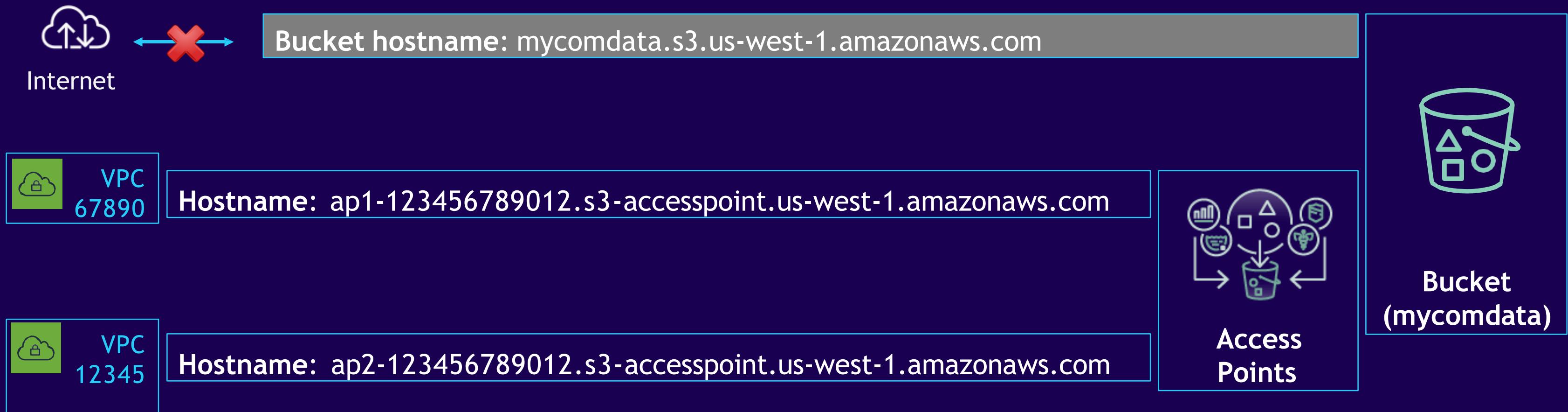
Access Point policies can establish granular control within limits enforced by the bucket policy



Use case: enforce VPC only data access for a bucket

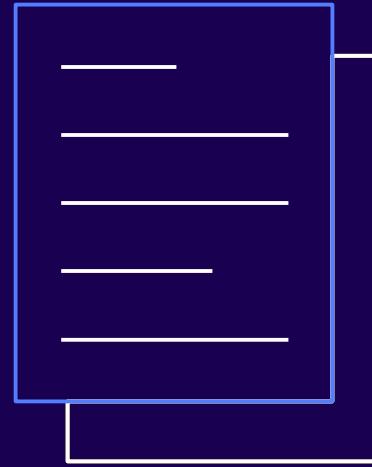
Access Points can be configured to limit access to a specified VPC only

- Create AWS Organization Service Control Policy to enforce VPC only access points for applications using the bucket
- Data access through the bucket directly disabled (enforced through bucket policy)



Amazon S3 inventory

A managed alternative to using the LIST API



Regularly generates a list of objects for **analytics** and **auditing**.

- Storage class
- Creation date
- Encryption status
- Replication status
- Object size, and more
- S3 Intelligent-Tiering access tier new!

Use Amazon Athena to filter S3 inventory reports

This query selects bucket, object key, version id for unencrypted objects

```
select s._1, s._2, s._3 from s3object s where s._6 = 'NOT-SSE'
```

Example results:

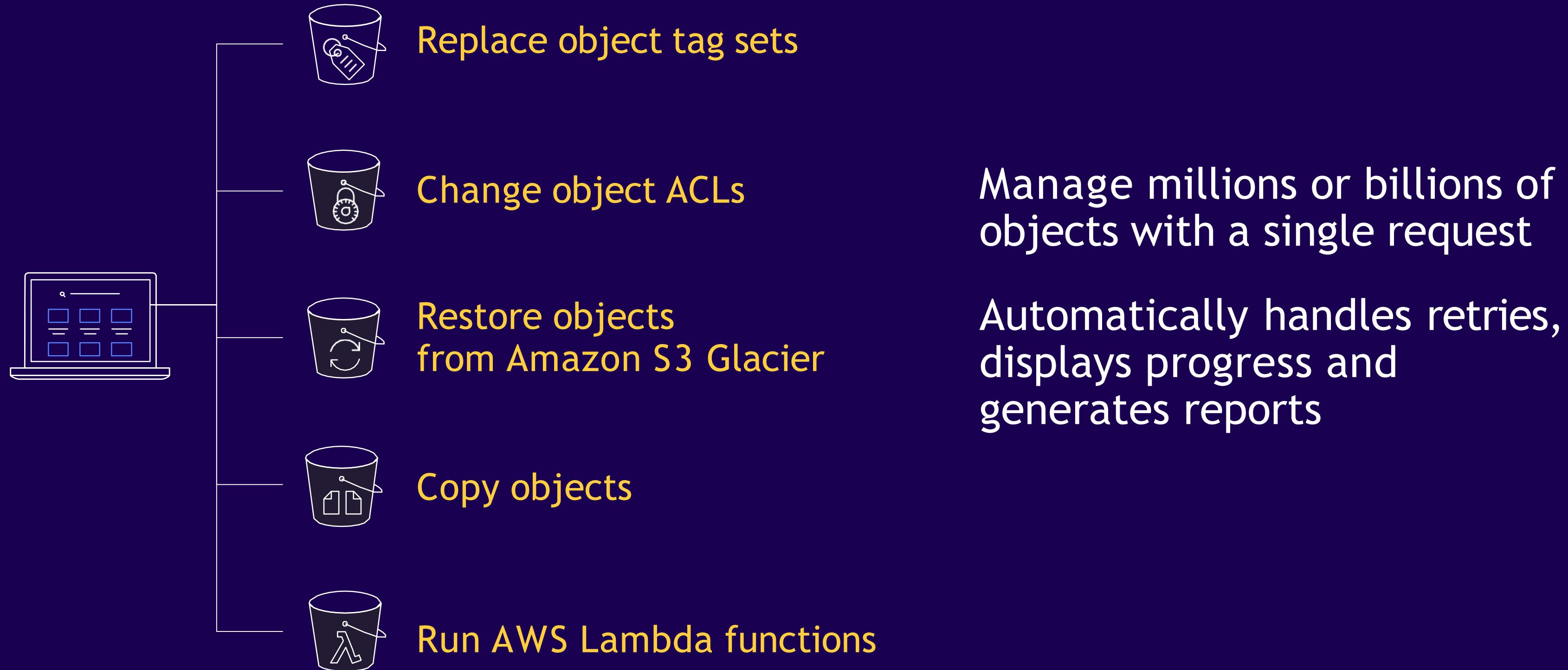
batchoperationsdemo,0100059%7Ethumb.jpg,lsrtlxksLu0R0ZkYPL.LhgD5caTYn6vu

batchoperationsdemo,0100074%7Ethumb.jpg,sd2M60g6Fdazoi6D5kNARIE7KzUibmHR

batchoperationsdemo,0100075%7Ethumb.jpg,TLYESLn1mXD5c4BwiOlinqFrktddkoL

Amazon S3 batch operations

Save time when performing one-time or recurring actions at scale



Amazon S3 batch operations

Choose objects

- S3 Inventory report
- CSV list

Select an operation

- Copy
- Restore from S3 Glacier
- Put Access Control List (ACL)
- Replace object tag sets
- Run AWS Lambda functions

View progress

- Object level progress
- Completion report

Use Amazon S3 batch operations to encrypt objects

Choose objects

- S3 Inventory report

- Filter S3 Inventory report with Amazon Athena
- Identify all unencrypted objects

Select an operation

- Copy

- Copy objects to the same bucket
- Specify desired encryption type

View progress

- Completion report

- Retain completion report of all tasks for object-level visibility

Amazon S3 batch operations and AWS Lambda

Run your custom code across billions of objects in Amazon S3



Manifest selection:

- Specify existing Amazon S3 objects
- Use URL-encoded JSON to pass object-level parameters
- Invoke general purpose AWS Lambda functions



AWS Lambda

AWS Lambda function:

- Invoke AWS services like Amazon Rekognition
- Use Amazon S3 operations like copy with parameters
- Run your own custom code

S3 Data Protection capabilities

GOAL



Replicate data for compliance
and bad actor protection



Protect data from accidental
deletes



Protect data for governance
and compliance purposes

AMAZON S3 AND S3 GLACIER FEATURES

Use [S3 Replication with Replication Time Control](#) and [ownership override](#)

Use [bucket versioning](#) while reducing cost with [Lifecycle policies](#)

Use [S3 Object Lock](#) to store objects as write-once-read-many (WORM)

Amazon S3 Replication

Amazon S3 Replication automatically copies
your data to the same or different AWS region



NEW!

Same-Region
Replication (SRR)



Cross-Region
Replication (CRR)

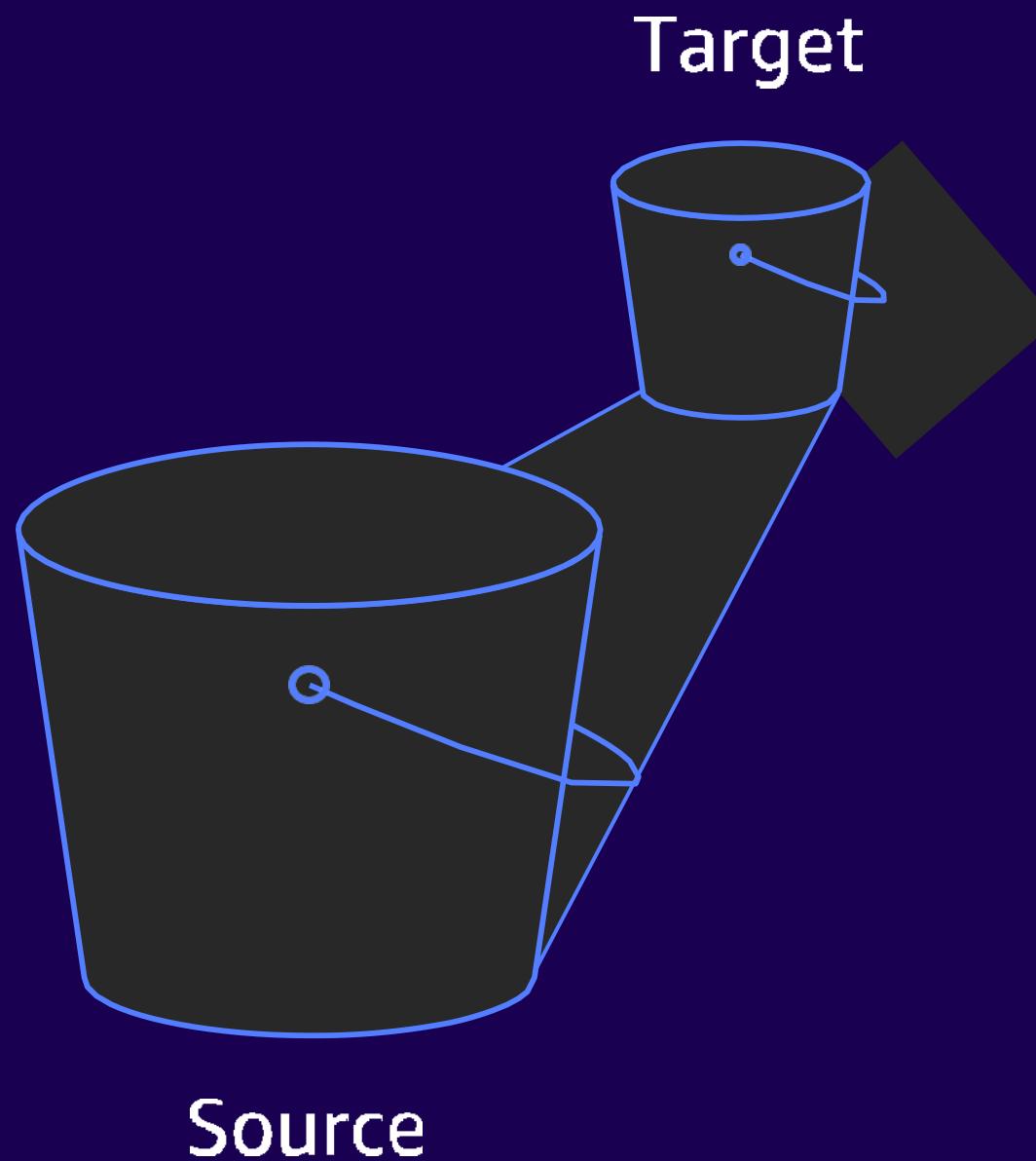


Source bucket



Destination bucket

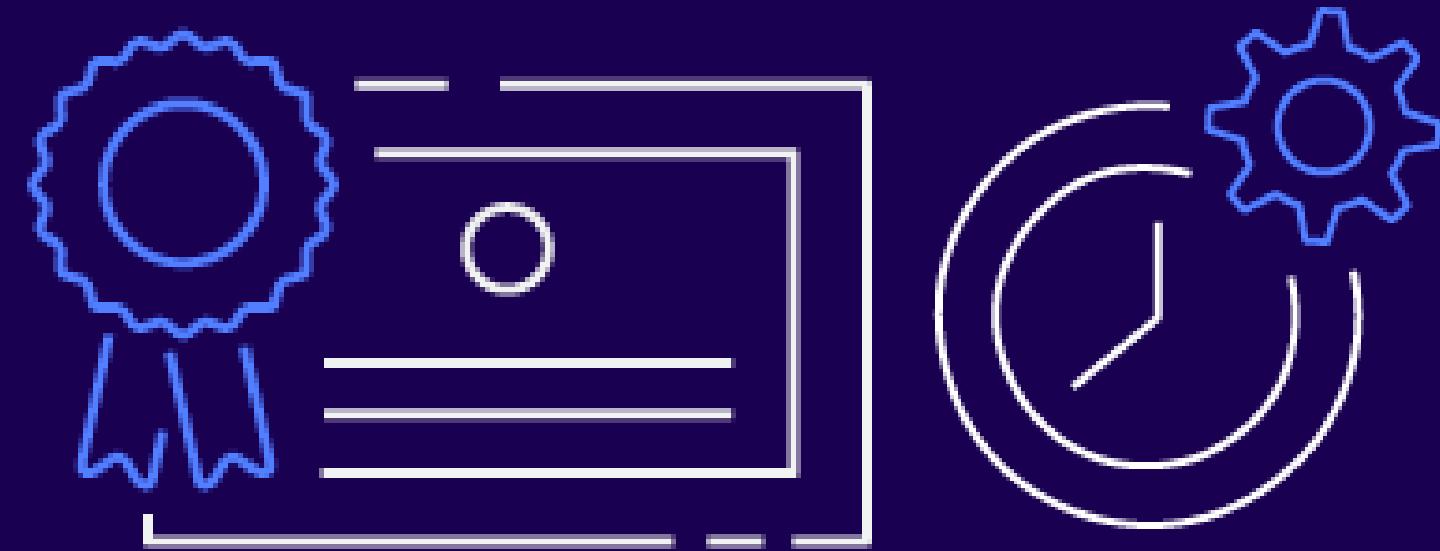
Amazon S3 Replication



Select	Select data	Replicate the whole bucket .. or based on a prefix .. or based on object tags
Protect	Select a region	Replicate to any second AWS region Continuous replication as your data changes Satisfy distance and residency requirements
	Change ownership	Automatically change ACLs on replica objects Reset object level permissions Protect against bad actors & IAM account compromise
	Cross account	Replicate into a second account Often this is a locked down "Archive" account Protect against AWS root account compromise
Optimize	Set storage class	Maintain production access characteristics .. or put and Lifecycle .. or replicate straight to Amazon S3 Glacier

Amazon S3 Replication time control

Designed to replicate 99.99% of objects within 15 minutes



15 minute replication
time backed by an
**AWS Service Level
Agreement (SLA)**

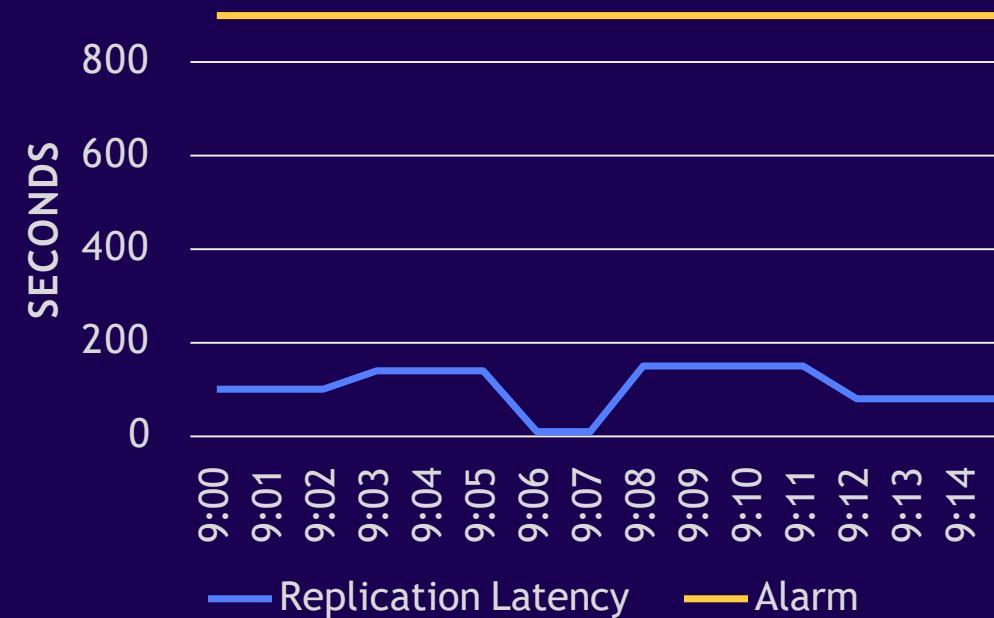


Monitor replication
using Amazon
CloudWatch metrics
and event notifications

Amazon S3 Replication time control

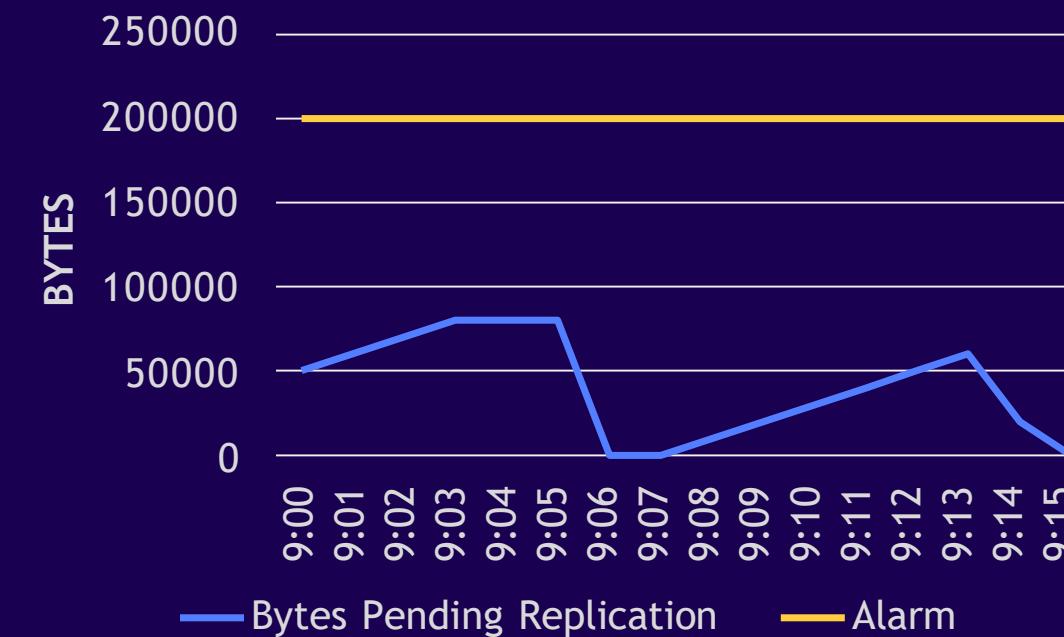
Designed to replicate 99.99% of objects within 15 minutes

Monitor your replication with 3 new CloudWatch metrics Optional: Set up
alarms on your metrics



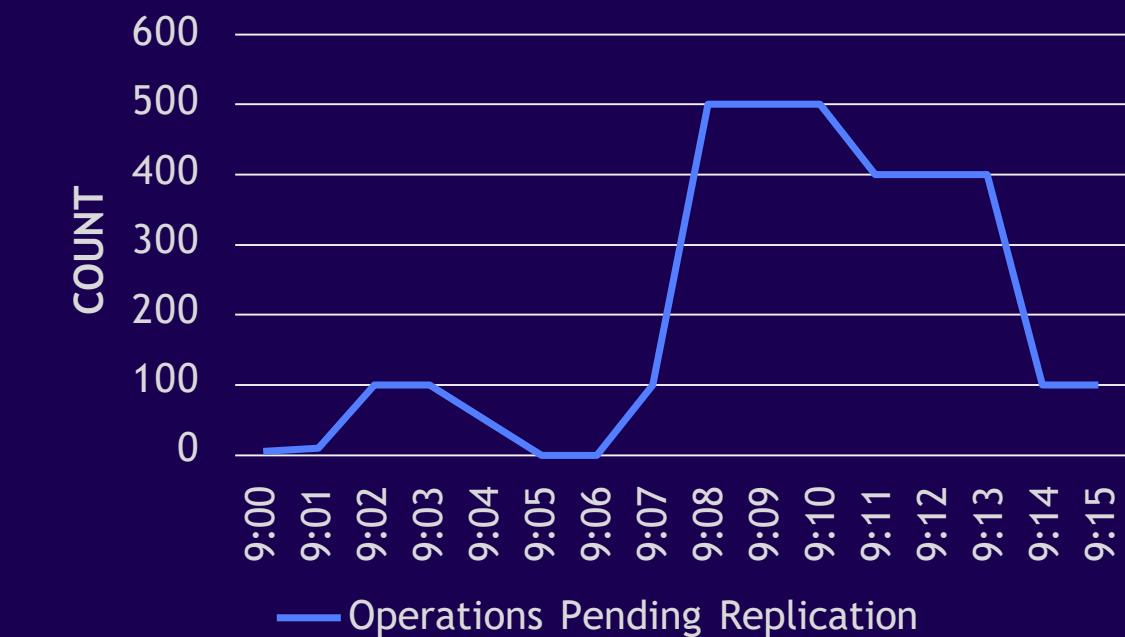
Replication latency

The maximum number of seconds by which the destination region is behind the source region for a given replication rule



Bytes pending replication

The total number of bytes of objects pending replication for a given replication rule

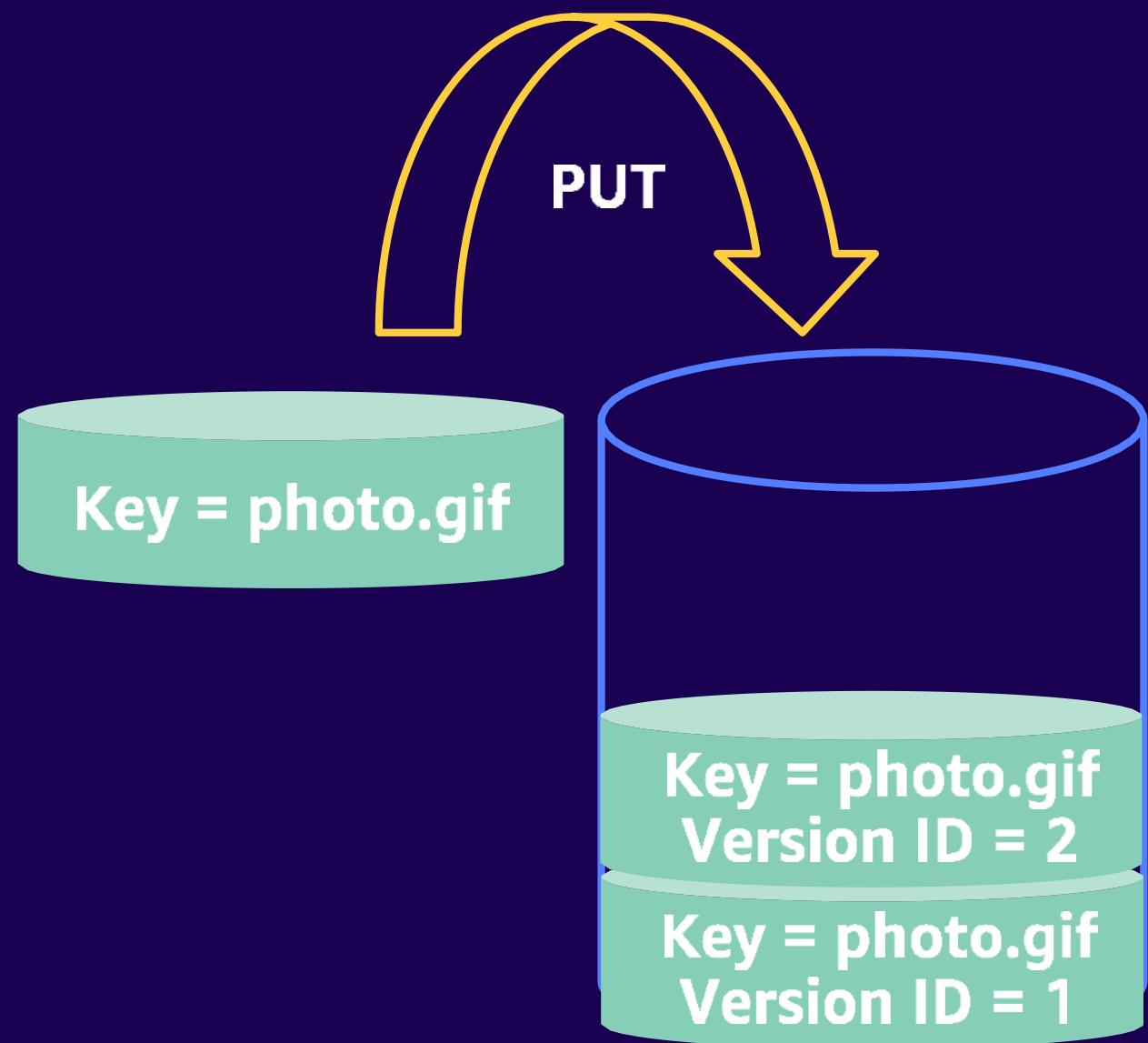


Operations pending replication

The number of operations pending replication for a given replication rule

Enable Amazon S3 bucket versioning

Use versioning to protect your data from accidental deletion



Create a **new version with every upload**
Previous versions are retained, not overwritten

Making **delete requests without a version ID**
removes access to objects, but keeps the data

Manage **previous versions with lifecycle**
Transition or expire objects a specified number
of days after they are no longer the current
version

Use lifecycle policies to expire object versions

Set lifecycle policies to control the cost of noncurrent versions

```
<LifecycleConfiguration>
  <Rule>
    <ID>objectversionexpiration</ID>
    <Filter>
      <Prefix></Prefix>
    </Filter>
    <Status>Enabled</Status>
    <NoncurrentVersionExpiration>
      <NoncurrentDays>7</NoncurrentDays>
    </NoncurrentVersionExpiration>
  </Rule>
</LifecycleConfiguration>
```

Amazon S3 Object Lock

Use **Object Lock** to store objects as write-once-read-many (WORM)



Compliance
mode

Store compliant
data

Governance
mode

Store data in
WORM format;
privileged users
can modify
retention controls

Legal
hold

If you're unsure
how long you
want your objects
to stay immutable

GA Dec. 3, 2024

Amazon S3 Tables

Fully managed Apache Iceberg tables
in Amazon S3



PREVIEW

PREVIEW Dec. 3, 2024

Amazon S3 Metadata

Automatic metadata generation,
accessible with simple SQL semantics

Video generated via Amazon Bedrock is automatically annotated with custom metadata

Amazon
Bedrock



In Amazon Bedrock's PUT request:

x-amz-meta-content-source: AmazonBedrock

x-amz-meta-content-model-id: arn:aws::::model/xyz-v1

Journal
table



Query from Amazon S3 Metadata:

```
SELECT bucket, key, user_defined_metadata, row_type
FROM aws_s3_metadata.my_metadata_table
WHERE user_defined_metadata['content-source'] = AmazonBedrock
```

GA Nov. 14, 2024

Higher S3 Bucket Limits

Amazon S3 now supports up to
1 million buckets per AWS account



GA Dec. 1, 2024

New data integrity defaults

Data validation over the wire,
new checksum info in metadata

GA 2024

Conditional S3 Requests

Put-if-absent and Put-if-match
conditions simplify distributed apps

GA Aug. 21, 2024

New Context in 403 Responses

More information for requests made
within an AWS Account

GA Aug. 21, 2024

New Context in 403 Responses

More information for requests made
within an AWS Account

GA re:Invent 2023

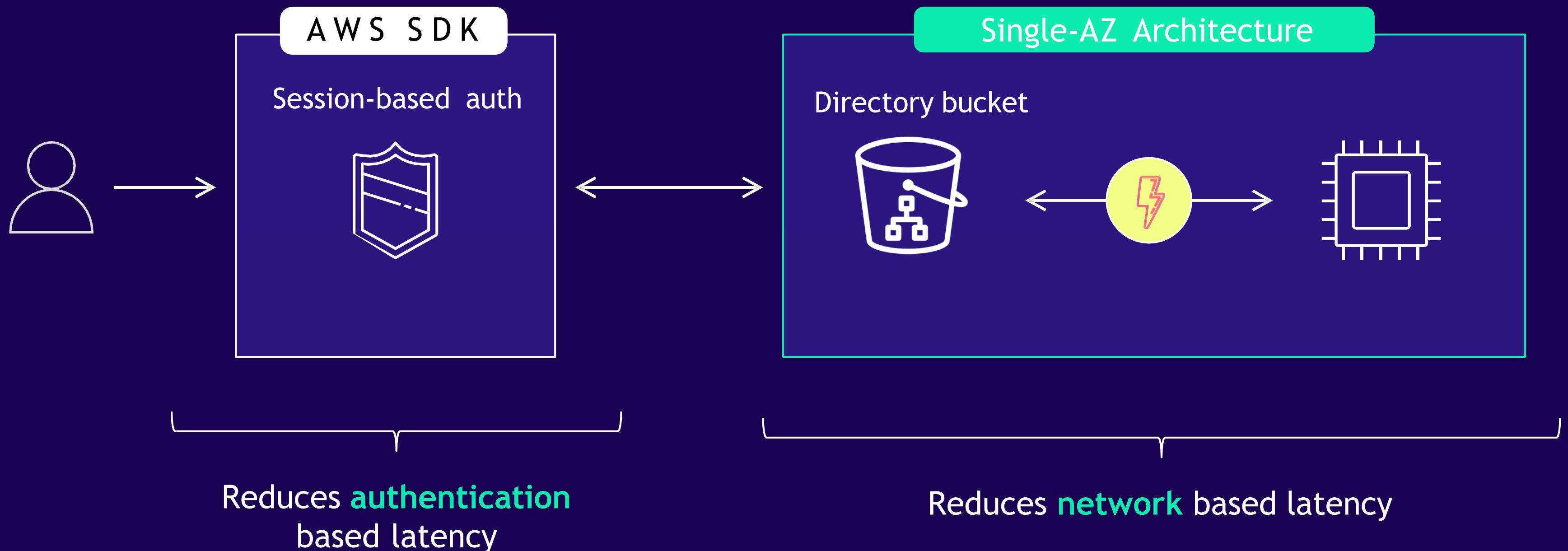
Amazon S3

Express One Zone Storage Class

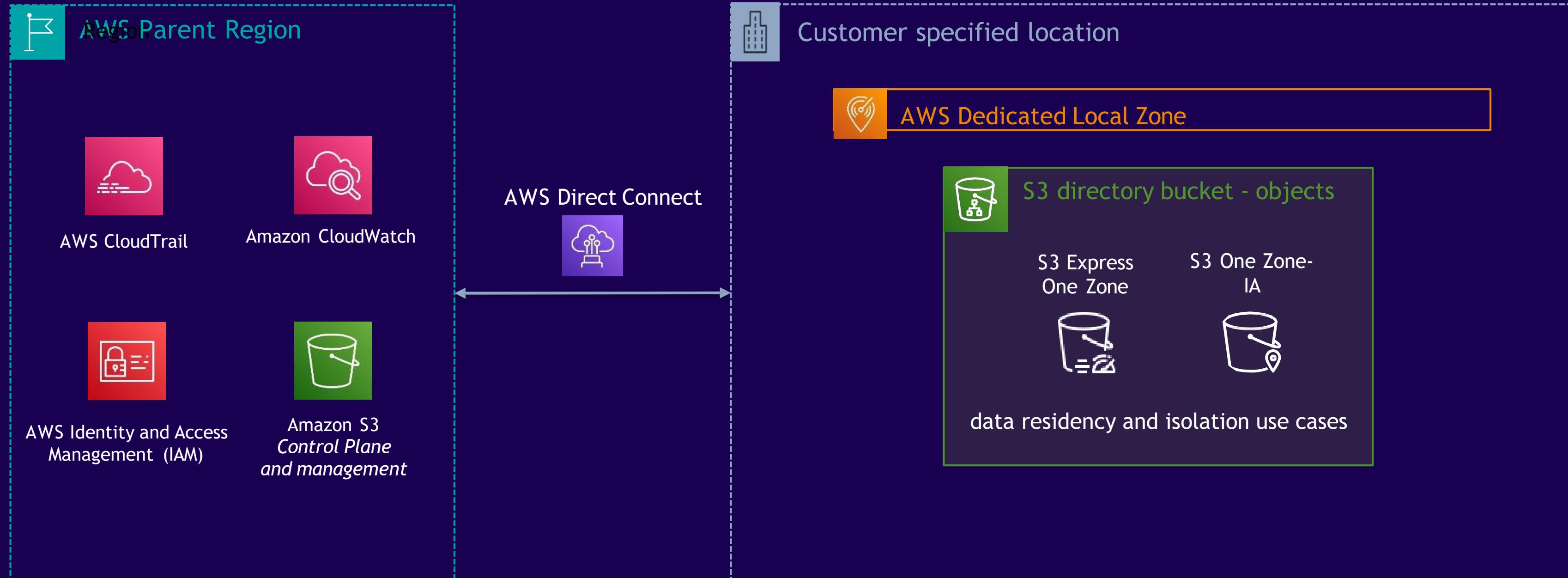
For compute-intensive workloads



Designed for performance



AWS Dedicated Local Zones



Bucket Location =  **AWS Local Zones**

GA Oct. 30, 2024

Amazon S3 static website hosting with AWS Amplify

Seamlessly host static website content
stored on S3 with just a few clicks

The AWS storage portfolio

Amazon S3

- Object storage: Data presented as buckets of objects
- Data access via APIs over the Internet

Amazon
Elastic Block
Store

- Block storage (analogous to SAN): Data presented as disk volumes
- Lowest-latency access from single Amazon EC2 instances

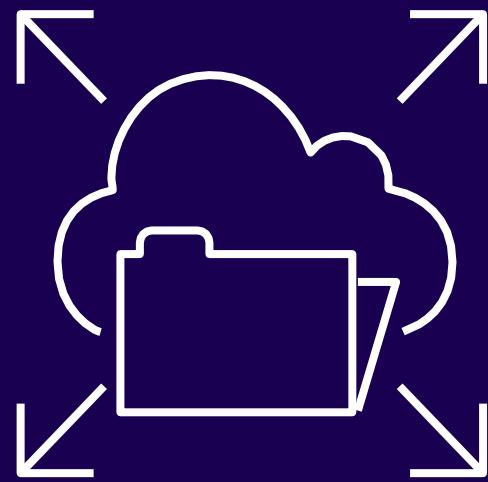
Amazon
Glacier

- Archival storage: Data presented as vaults/archives of objects
- Lowest-cost storage, infrequent access via APIs over the Internet

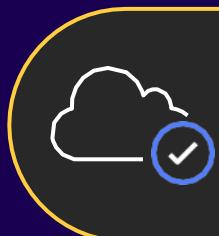
Amazon
EFS

- File storage (analogous to NAS): Data presented as a file system
- Shared low-latency access from multiple EC2 instances

Amazon Elastic File System (Amazon EFS)



Amazon EFS
is a fully managed file system that is...



Cloud native

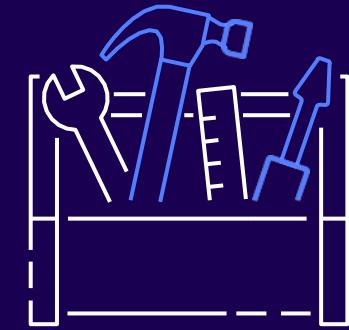


Highly reliable



Cost optimized

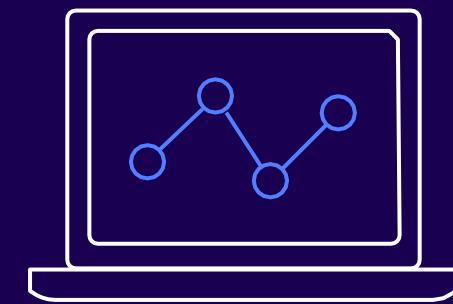
Use cases for Amazon EFS



Home directories
Container storage
Application test/dev
Metadata-intensive jobs



Lift and shift enterprise apps
Web serving
Content management
Database backups



Analytics
Media workflows

Low latency and serial I/O

High throughput and parallel I/O

Using the right tool



Amazon Elastic
File Service (EFS)

Linux-Based
Workloads



Amazon FSx
for Windows File Server

Windows-Based
Workloads



Amazon FSx
for Lustre

Compute-
Intensive
Workloads

FILE SYSTEMS FOR BUSINESS WORKLOADS

FILE SYSTEM FOR COMPUTE-
INTENSIVE WORKLOADS

Amazon EFS is simple

- Fully managed
 - No hardware, network, file layer
 - Create a scalable file system in seconds!
- Seamless integration with existing tools and apps
 - NFS v4—widespread, open
 - Standard file system semantics
 - Works with standard OS file system APIs
- Simple pricing = simple forecasting



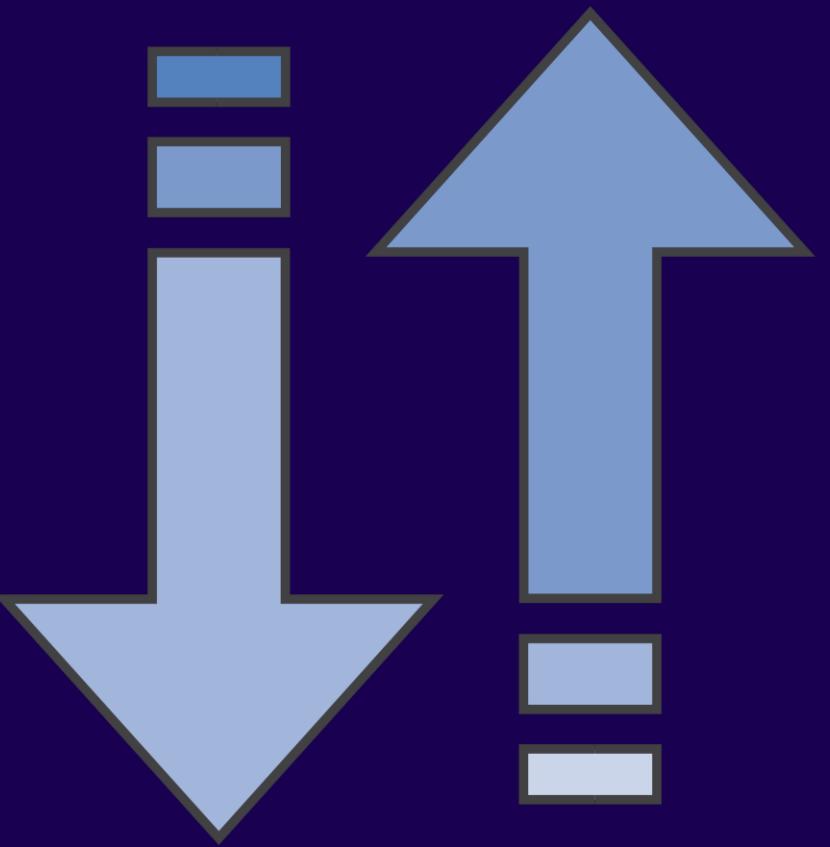
Amazon EFS is elastic

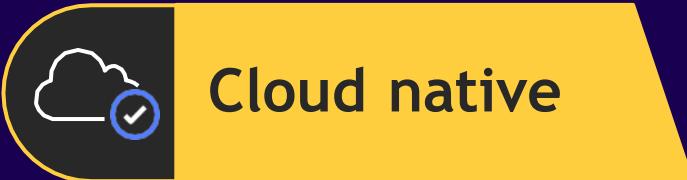
- File systems grow and shrink automatically as you add and remove files
- No need to provision storage capacity or performance
- You pay only for the storage space you use, with no minimum fee



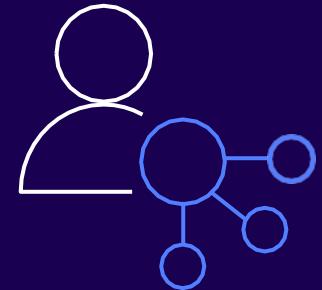
Amazon EFS is scalable

- File systems can grow to petabyte scale
- Throughput and IOPS scale automatically as file systems grow
- Consistent low latencies regardless of file system size
- Support for thousands of concurrent NFS connections



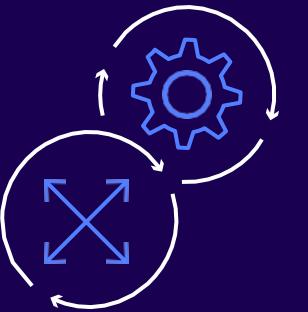


Cloud native



Elastic

- Grow & shrink on demand
- No need to provision and manage infrastructure & capacity
- Pay as you go, pay only for what you use
- Simple to use, create a file system in seconds



Scalable

- Grow up to petabytes
- Performance modes for low latencies and maximum I/O
- Throughput that scales with storage
- Provisioned throughput available



Integrated

- Integrated with various AWS computing models
- Shared access from on-premises, inter region, and cloud-native applications
- Access concurrently from thousands of Amazon EC2 instances
- Attach to containers launched by both Amazon ECS (AWS Fargate) and Amazon EKS (AWS Fargate - coming soon)
- Use with Amazon SageMaker notebooks



Amazon EFS



Highly reliable



Highly available, durable

- Stores data across three availability zones for high availability and durability
- Access your file system from multiple AZs
- Strong consistency for concurrent access



Secure

- Control network traffic
- Control file and directory access
- Control administrative (API) access with AWS IAM
- Encrypt data at rest and in transit
- Control NFS client access with AWS IAM
- Manage application access with EFS Access Points



Amazon EFS



Global footprint

- Amazon EFS is available in ALL regions
- Globally expanding regional service
- 90-day SLA for launch in new regions: Cape Town and Milan (5/13)



Cloud native

Highly integrated, shared access





Highly reliable

Benefits of 3 independent AZs in each AWS Region



99.99% Availability
SLA



Designed for
99.99999999% durability



Cost optimized

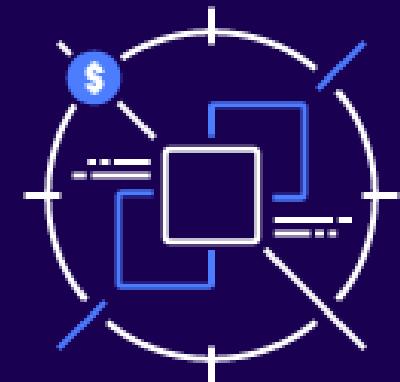
Cost optimized | Amazon EFS



No minimum commitments or upfront fees



No need to provision storage



Use with Spot Instances



Automatic lifecycle management to lower cost storage

Amazon EFS Infrequent Access

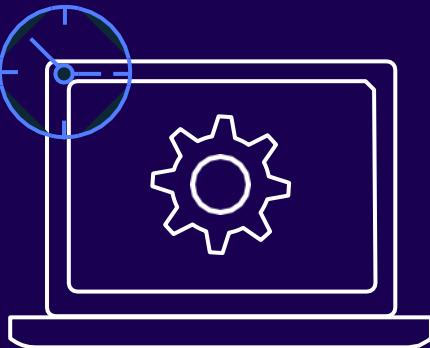
Amazon EFS IA storage class for infrequently accessed files for \$0.025/GB per month*



No changes to existing applications using Amazon EFS



Cost savings up to 92%



Automated lifecycle management

* Pricing in the US East (N. Virginia) region

EFS performance modes



General Purpose (default)

Recommended for the majority of workloads

What it's for

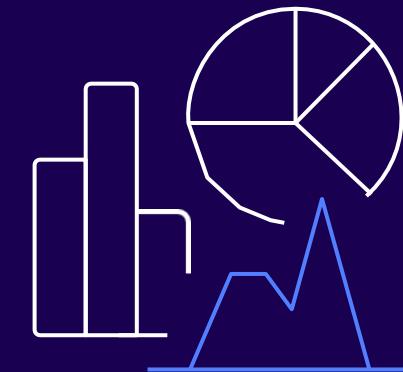
Latency-sensitive applications and general-purpose workloads

Large-scale and data-heavy applications

Advantages

Lowest latencies for file metadata operations

Virtually unlimited ability to scale out IOPS



Trade-offs

35k read ops/sec

Slightly higher metadata latencies

Max I/O

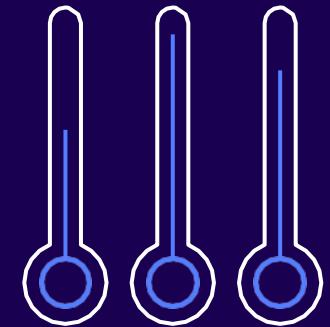
Recommended for scale-out workloads

When to use

Best choice for most workloads

Consider for large scale-out workloads

EFS throughput modes



Bursting Throughput (default)

Recommended for the majority of workloads

What it's for

Varying throughput workloads | Increased, more consistent throughput workloads

Advantages

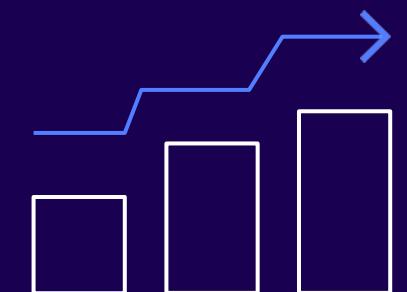
Auto-scaling throughput | User-defined throughput

Trade-offs

Fixed throughput-to-storage ratio | Separate throughput charge

When to use

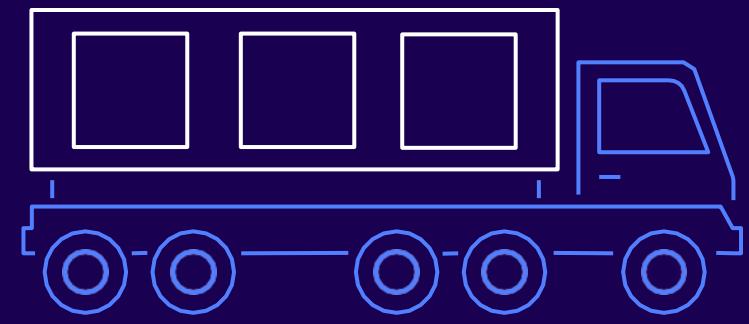
Best choice for most workloads | Higher throughput-to-storage ratio
Loading more than 2.1 TB



Provisioned Throughput

Recommended for higher throughput-to-storage ratio workloads

Provisioned Throughput



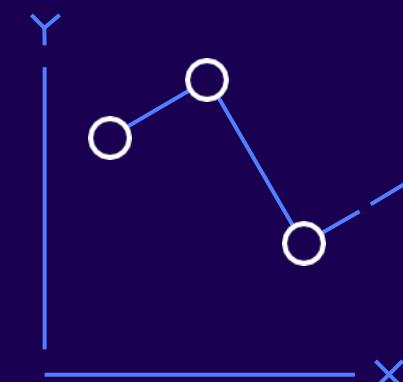
**Independent
throughput**

Provision throughput
independently of
data stored



Increase

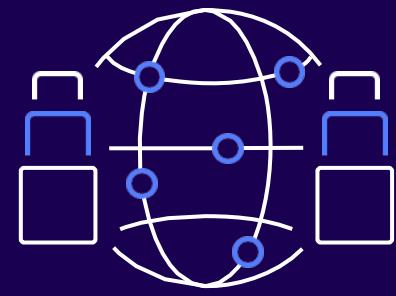
As often as you need



**Switch or
decrease**

Once every 24+ hours

Security and compliance



Control network traffic

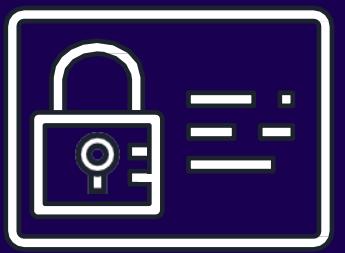
Using Amazon VPC security groups and network ACLs



Control file and directory access

Using POSIX permissions

control client identity/mount permissions with access points



Control administrative & application access

Using AWS IAM, action-level and resource-level permissions, identity-based policies

Manage application access with EFS Access Points



Encrypt data

At rest and in transit



Achieve compliance

HIPAA
GDPR

PCI-DSS

SOC

ISO

FedRAMP

AWS integrations |Amazon EFS



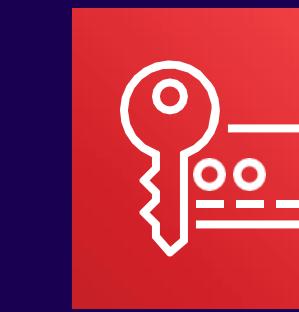
Amazon EFS



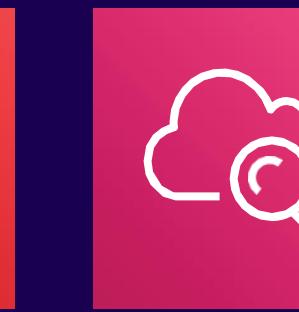
Amazon
VPC



AWS
IAM



AWS
KMS



Amazon
CloudWatch



AWS
CloudTrail



AWS
CloudFormation



AWS Direct
Connect



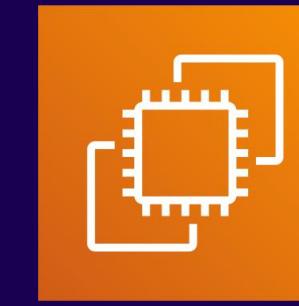
AWS VPN



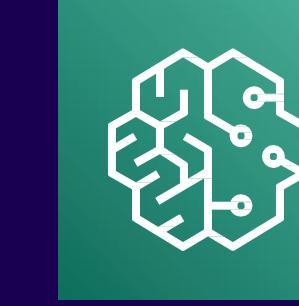
AWS
DataSync



AWS
Backup



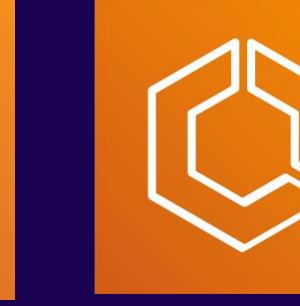
Amazon
EC2



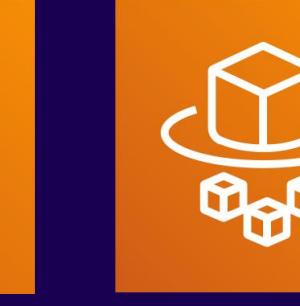
Amazon
SageMaker



Amazon
EKS

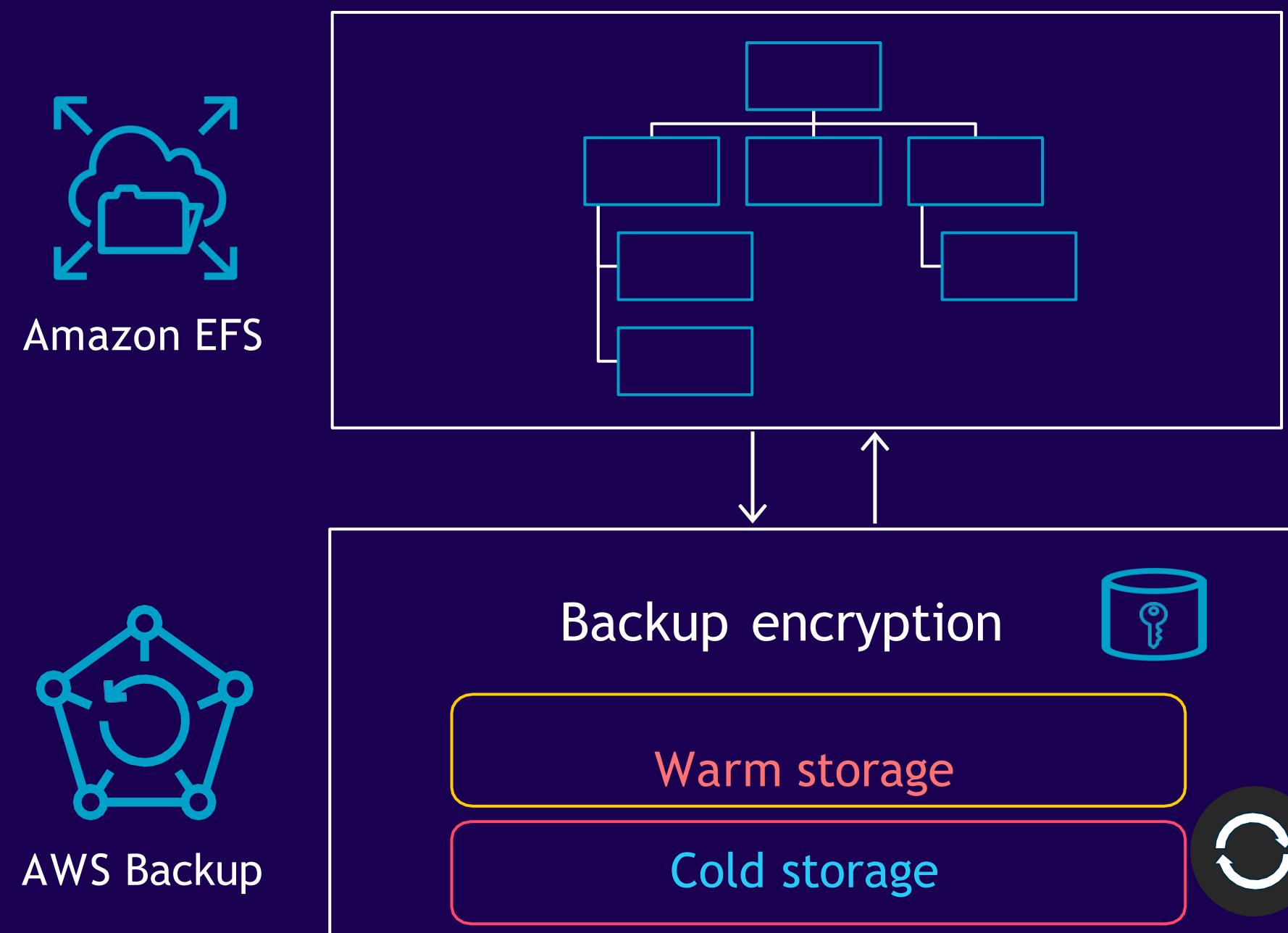


Amazon
ECS



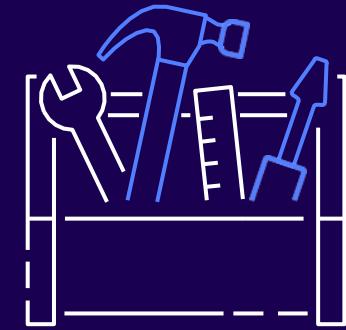
AWS
Fargate

Backup for Amazon EFS



- EFS file systems can be backed up and restored using AWS Backup
- AWS Backup provides automated backup scheduling and retention per user defined policy
- AWS Backup offers two classes of service backup storage with the ability to lifecycle to cold storage
- Restore individual files and directories

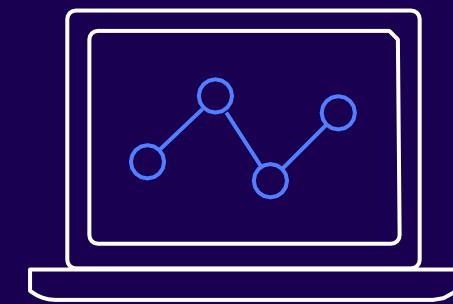
Use cases for Amazon EFS



Home directories
Container storage
Application test/dev
Metadata-intensive jobs



Lift and shift enterprise apps
Web serving
Content management
Database backups

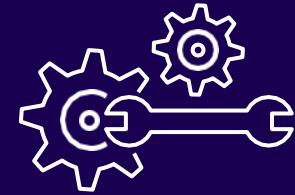


Analytics
Media workflows

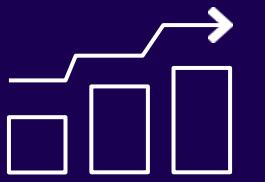
Low latency and serial I/O

High throughput and parallel I/O

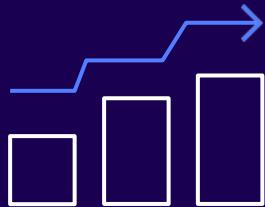
Amazon EFS | Best practices



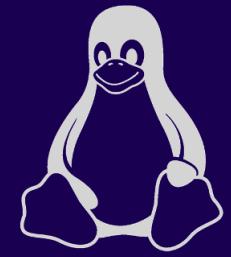
Test in
General Purpose
performance mode



Start with
Bursting
Throughput mode



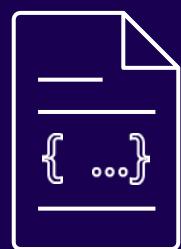
Consider Provisioned
Throughput mode for
loading >2.1 TB



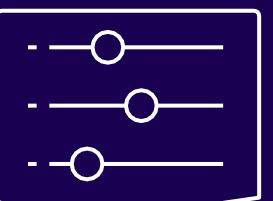
Linux kernel 4.3+



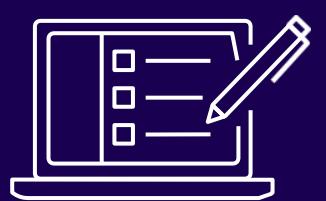
EFS mount helper
(NFSv4.1)



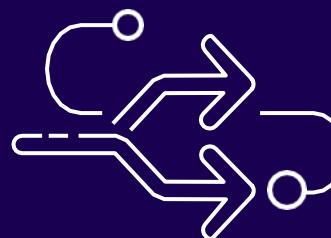
Large IO size
(aggregate IO)



Multiple
threads



Multiple
instances



Multiple
directories

Amazon EFS | Best practices



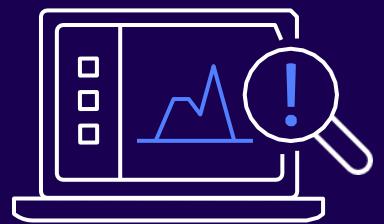
Enable Lifecycle Management
to automatically
save up to 92%



Enable encryption
at rest and in transit
for sensitive workloads



Create a backup
plan to further
protect your data



Monitor throughput
utilization, burst credits,
and PercentIOLimit

Knowledge check

Which of the following services offers object-based storage?

- A. Amazon Elastic Block Store (Amazon EBS)
- B. AWS Storage Gateway
- C. Amazon Elastic File System (Amazon EFS)
- D. Amazon S3
- E. Amazon Machine Images (AMIs)

Knowledge check

Which of the following services offers object-based storage?

- A. ~~Amazon Elastic Block Store (Amazon EBS)~~
- B. ~~AWS Storage Gateway~~
- C. ~~Amazon Elastic File System (Amazon EFS)~~
- D. Amazon S3
- E. ~~Amazon Machine Images (AMIs)~~

Answer: D

Key takeaways

AWS provides a variety of storage options

- Object (Amazon S3)
- File (Amazon EFS and Amazon FSx)
- Block storage (Amazon EBS)
- Customers are using AWS storage services to build:
 - Home directories
 - Data lakes
 - Modern and business-critical applications

Databases

What are we covering

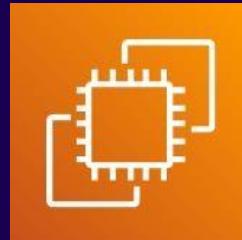
Managed vs Unmanaged

Managed DB services offered in AWS

Selecting the Right Type for you

Managed vs Unmanaged

DIY (Unmanaged services) compared to AWS database services (managed services)



Databases on Amazon EC2

- Operating system access
- Need features of specific application



AWS database services

- Simple to set up, manage, maintain
- Push-button high availability
- Focus on performance
- Managed infrastructure

Managed services transform operations

App optimization

Scaling

High Availability

Database backups

DB software patches

DB software installs

OS patches

OS installation

Server maintenance

Rack & stack

Power, HVAC, net

Operating
Databases
in the Old World



Operating
Databases
in AWS

App optimization



Scaling

High Availability

Database backups

DB software patches

DB software installs

OS patches

OS installation

Server maintenance

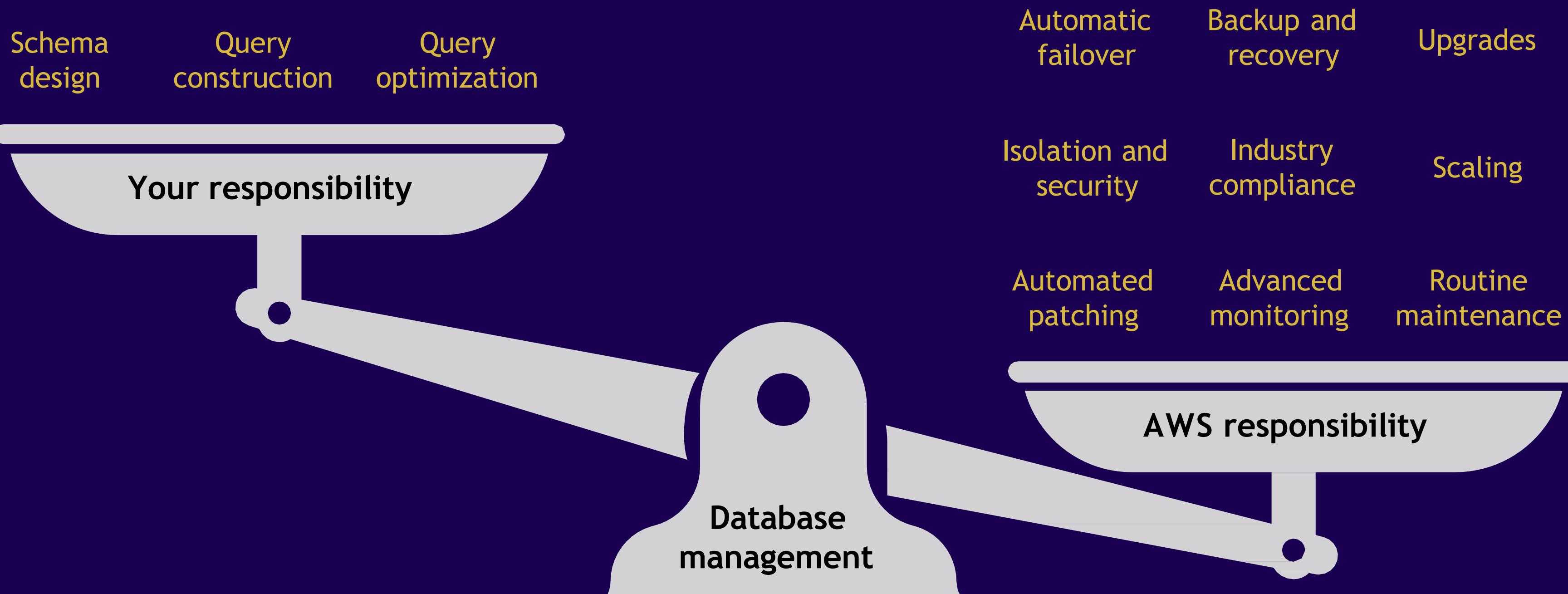
Rack & stack

Power, HVAC, net



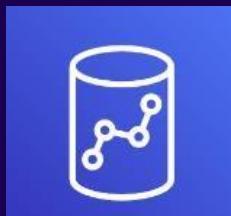
Accelerate path to innovation with managed databases

SPEND TIME INNOVATING AND BUILDING APPS, NOT MANAGING INFRASTRUCTURE



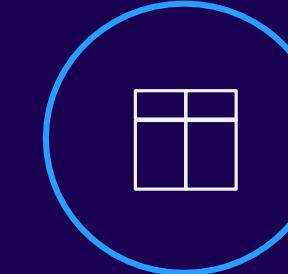
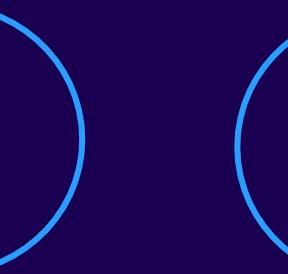
Quick overview of services

Purpose-built databases

Relational	Non Relational (NoSQL) databases for specific data models and have flexible schemas for building modern applications						
	Key-value	In-memory	Document	Wide-Column	Graph	Ledger	Time Series
 Amazon RDS	 Amazon DynamoDB	 Amazon ElastiCache	 Amazon DocumentDB	 Amazon Keyspaces (for Apache Cassandra)	 Amazon Neptune	 Amazon QLDB	 Amazon Timestream
 Amazon Aurora							
 Amazon Redshift		 Amazon MemoryDB for Redis					

Purpose-built databases

AWS OFFERS A MODERN DATABASE PORTFOLIO OF 10+ PURPOSE-BUILT DATABASES

								
	Relational Referential integrity, ACID transactions, schema-on-write	Key-value High throughput, low-latency reads and writes, endless scale	Document Store documents; quickly access querying on any attribute	In-memory Query by key with microsecond latency	Graph Quickly and easily create and navigate relationships between data	Time-series Collect, store, and process data sequenced by time	Ledger Complete, immutable and verifiable history of all changes to application data	Wide column Scalable, highly available, and managed Apache Cassandra-compatible service
AWS service(s)	 Aurora  RDS	 DynamoDB	 DocumentDB	 ElastiCache  MemoryDB	 Neptune	 Timestream	 QLDB	 Keyspaces

AWS database services

AWS OFFERS A MODERN DATABASE PORTFOLIO OF 10+ PURPOSE-BUILT DATABASES

Relational



Aurora



MySQL



PostgreSQL



RDS



MySQL



PostgreSQL



MariaDB



ORACLE



Microsoft SQL Server

Non-relational



DynamoDB
Key value



ElastiCache
In-memory



Neptune
Graph



DocumentDB
Document



Timestream
Time series



MemoryDB
In-memory



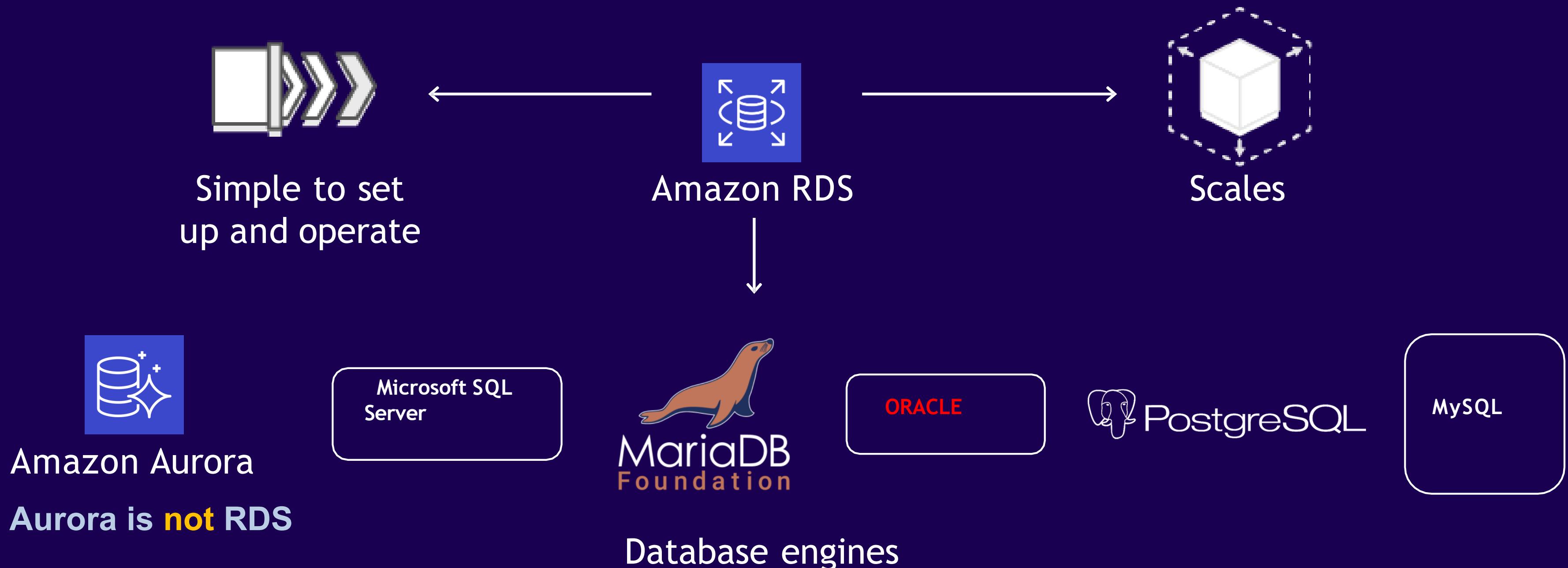
Keyspaces
Wide column



QLDB
Ledger

Amazon RDS

Set up, operate, and scale a relational database in the cloud with just a few clicks



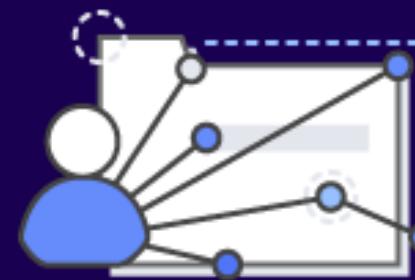
Amazon Aurora

Relational database built for the cloud; compatible with MySQL and PostgreSQL



Amazon DynamoDB

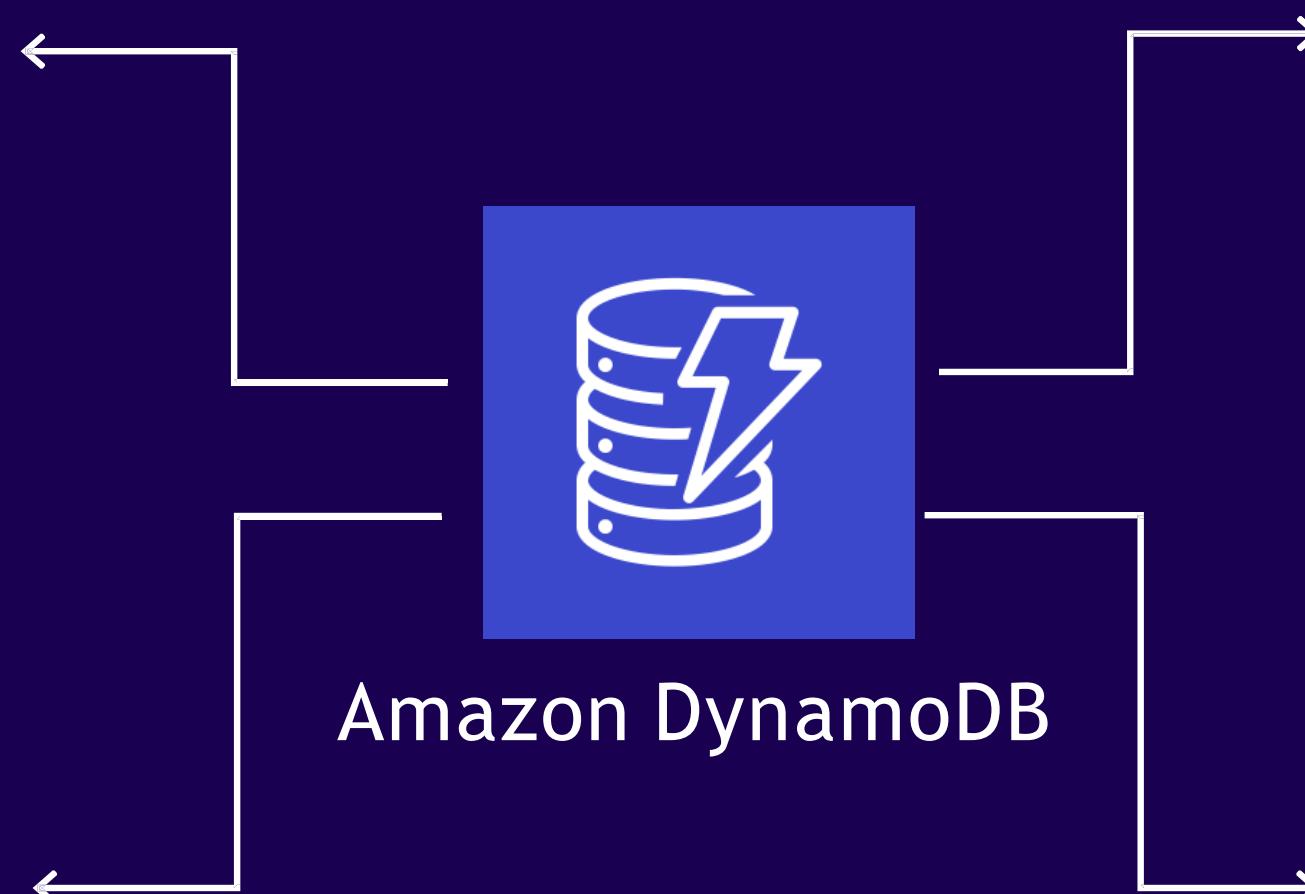
Fast and flexible NoSQL database service for any scale



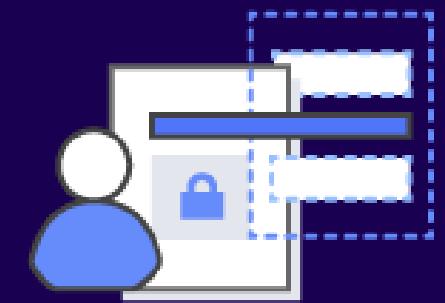
Fully managed



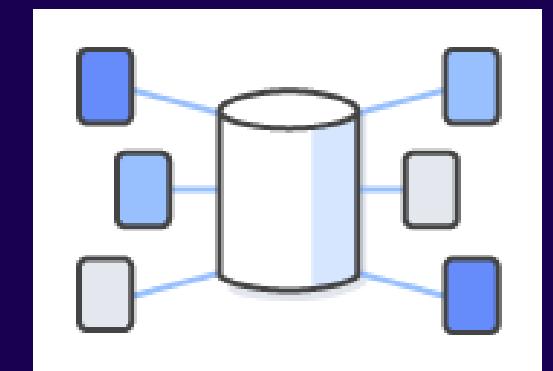
Fast,
consistent
performance



Fine-grained
access control



Flexible



AWS databases and analytics

Broad and deep portfolio, purpose-built for builders

Business Intelligence & Machine Learning



QuickSight



SageMaker



Comprehend

Relational Databases



Aurora



RDS

Non-Relational Databases



DynamoDB



ElastiCache
(Redis, Memcached)



Neptune
(Graph)

Data Lake



S3/Glacier



Glue

(ETL & Data Catalog)

Data Movement

Database Migration Service | Snowball | Snowmobile | Kinesis Data Firehose | Kinesis Data Streams

Selecting the right Service

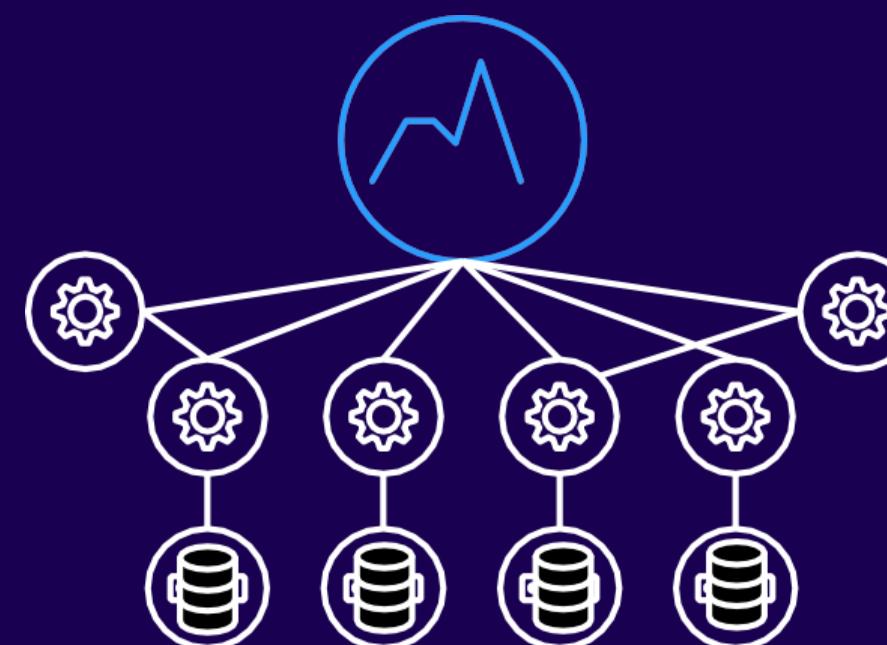
Major trends driving innovation

Explosion of data



Data grows 10x
every 5 years

Modern architecture changes
data and analytics requirements



Purpose-built databases provide
optimized performance and cost
savings.

Rapid rate of change



Transition from IT to
DevOps increases rate of
change

Modern application requirements

REQUIRES MORE PERFORMANCE, SCALE, AND AVAILABILITY



E-commerce

Media streaming

Social media

Online gaming
Shared economy

Users	1M+
Data volume	Terabytes–petabytes
Locality	Global
Performance	Microsecond latency
Request rate	Millions per second
Access	Mobile, IoT, devices
Scale	Virtually unlimited
Economics	Pay as you go
Developer access	Instance API access
Development	Apps and storage are decoupled

One-size-fits-all approach
of using a relational
database for everything is
no longer working



Instead of a monolithic application...



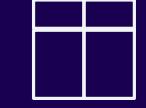
...build microservices with purpose-built tools

Choosing the right database allows
your teams to focus on building
applications that **meet your**
specific business needs

Scale
faster

Focus on
innovation

Accelerate
time to
market



How to choose a database?



Data Type

Structured,
Unstructured, JSON



Data Size

Text messages, Book,
Size of a collection,
Growth rate



Access Pattern

Query, batch, eventual
vs immediate
consistency

AWS databases and analytics

Broad and deep portfolio, purpose-built for builders

Business Intelligence & Machine Learning



QuickSight



SageMaker

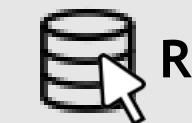


Comprehend

Relational Databases



Aurora



RDS

Non-Relational Databases



DynamoDB



ElastiCache
(Redis, Memcached)



Neptune
(Graph)

Data Lake



S3/Glacier



Glue

(ETL & Data Catalog)

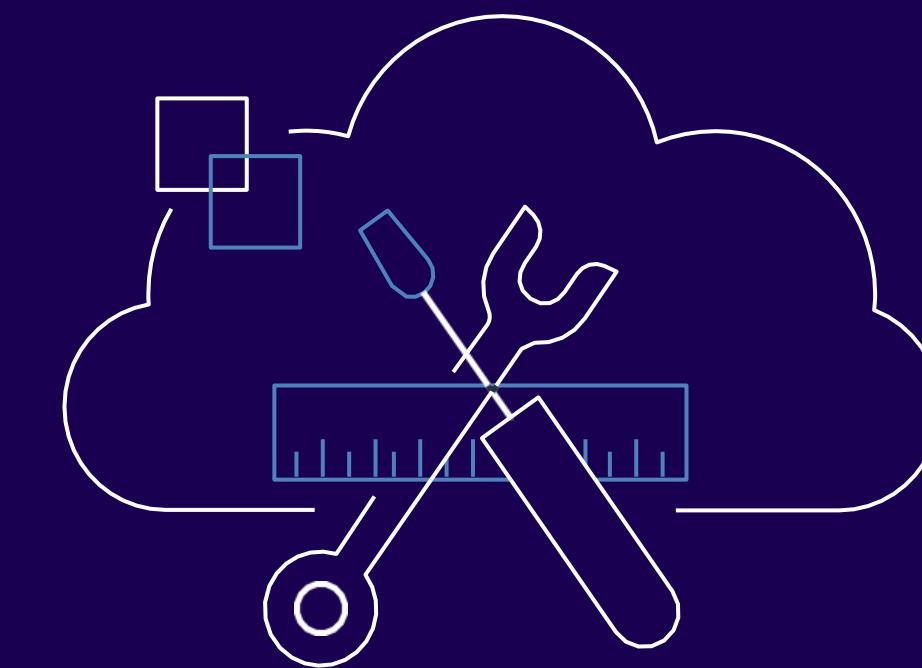
Data Movement

Database Migration Service | Snowball | Snowmobile | Kinesis Data Firehose | Kinesis Data Streams

Two fundamental areas of focus



“Lift and shift” existing
apps to the cloud



Quickly build new
apps in the cloud

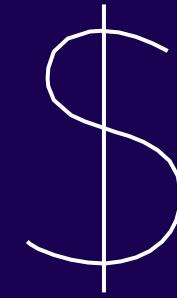
Two fundamental areas of focus



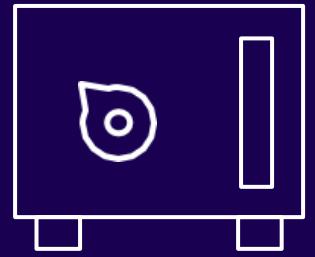
“Lift and shift” existing
apps to the cloud

Quickly build new
apps in the cloud

Old-guard commercial databases



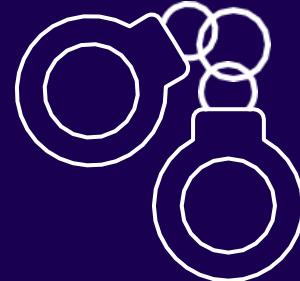
Very
expensive



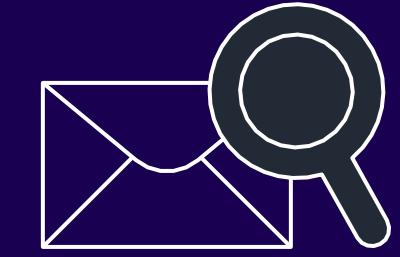
Proprietary



Lock-in



Punitive
licensing



You've
got mail

Relational data

- Divide data among tables
- Highly structured
- Relationships established via keys enforced by the system
- Data accuracy and consistency

No of patients visits made by each doctor last week grouped hospital wise

Your own very sweet query...
simple or complicated...



Amazon Relational Database Service (Amazon RDS)

MANAGED RELATIONAL DATABASE SERVICE WITH YOUR CHOICE OF DATABASE ENGINE



Amazon
Aurora



Microsoft
SQL Server

Oracle

Easy to administer



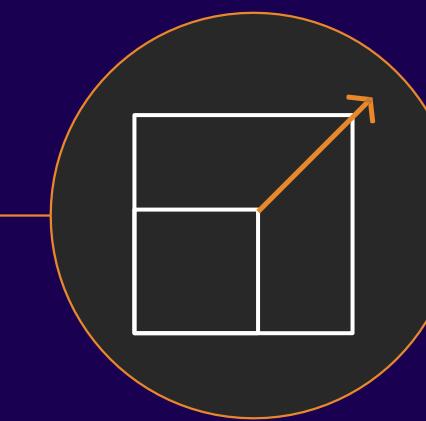
Easily deploy and maintain hardware, OS and DB software; built-in monitoring

Available and durable



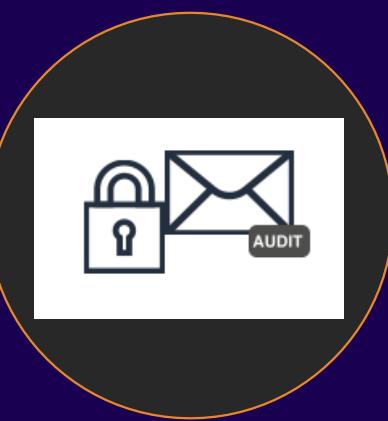
Automatic multi-AZ data replication; automated backup, snapshots, failover

Performant and scalable



Scale compute and storage with a few clicks; minimal downtime for your application

Secure and compliant



Data encryption at rest and in transit; industry compliance and assurance programs

Amazon Aurora

MySQL and PostgreSQL-compatible relational database built for the cloud

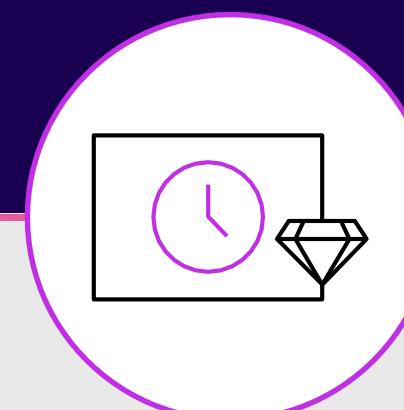
Performance and availability of commercial-grade databases at 1/10th the cost

Performance and scalability



5x throughput of standard MySQL and 3x of standard PostgreSQL; scale-out up to 15 read replicas

Availability and durability



Fault-tolerant, self-healing storage; six copies of data across three Availability Zones; continuous backup to Amazon S3

Highly secure



Network isolation, encryption at rest/transit

Fully managed



Managed by RDS: No hardware provisioning, software patching, setup, configuration, or backups

Large relational databases with Amazon Aurora

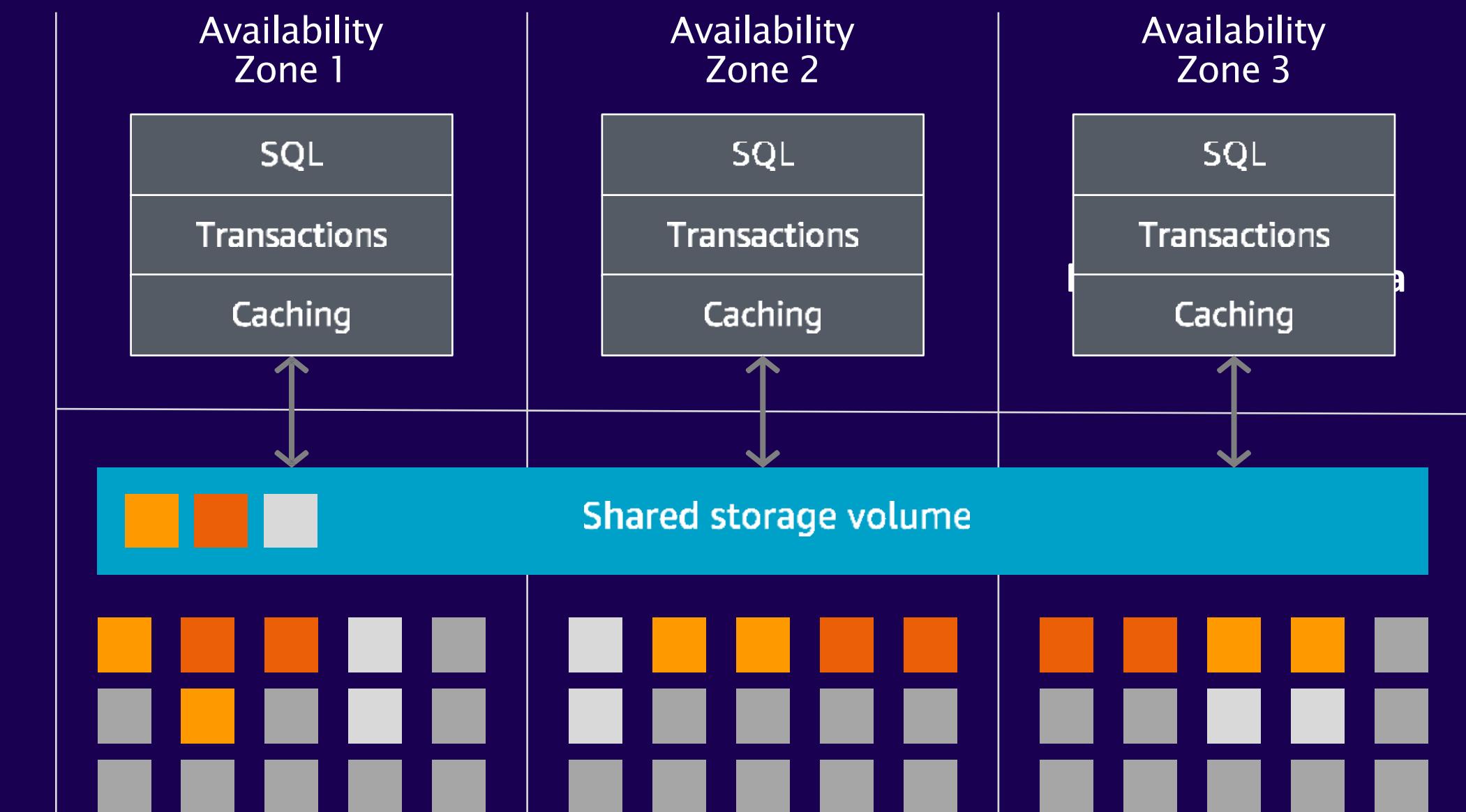
Scale-out, distributed, multi-tenant architecture

Fully compatible with PostgreSQL and MySQL, with 3x – 5x the throughput

Storage volume striped across hundreds of storage nodes distributed over 3 different availability zones

Six copies of data on SSD, two copies in each availability zone, to protect against AZ+1 failures

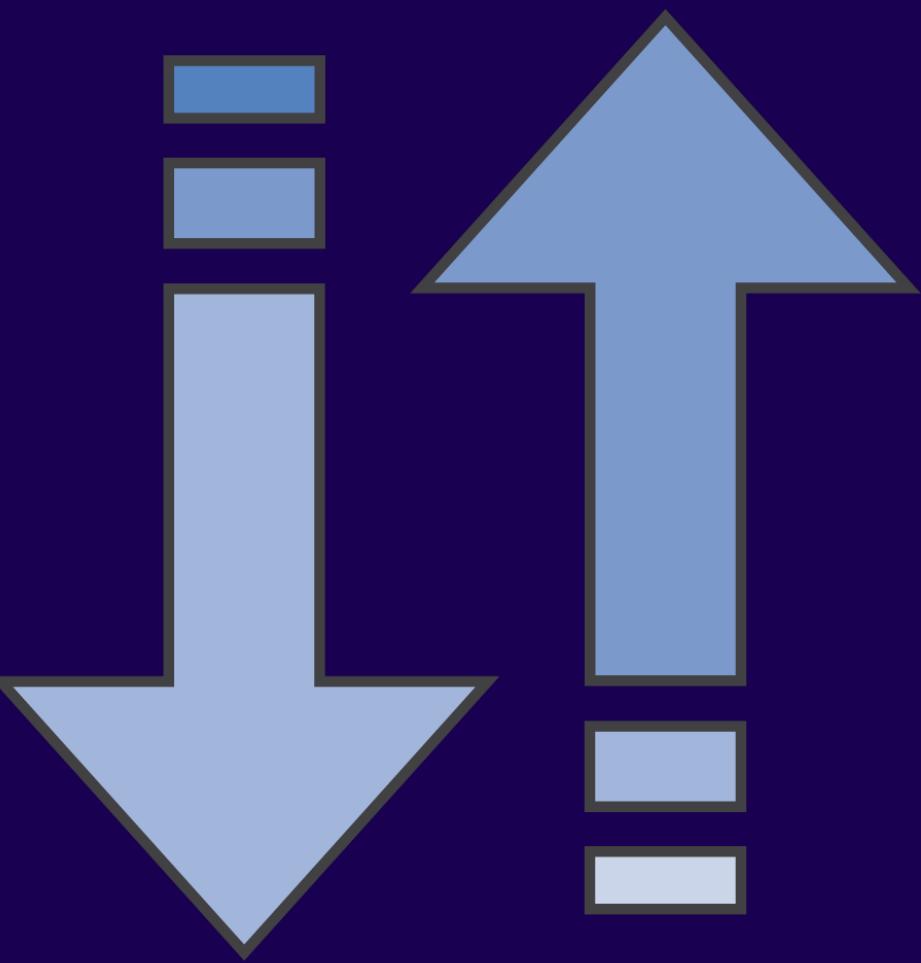
Continuous backup to Amazon S3 (built for 99.99999999% durability)



Aurora Serverless . . .

Responds to your application load automatically

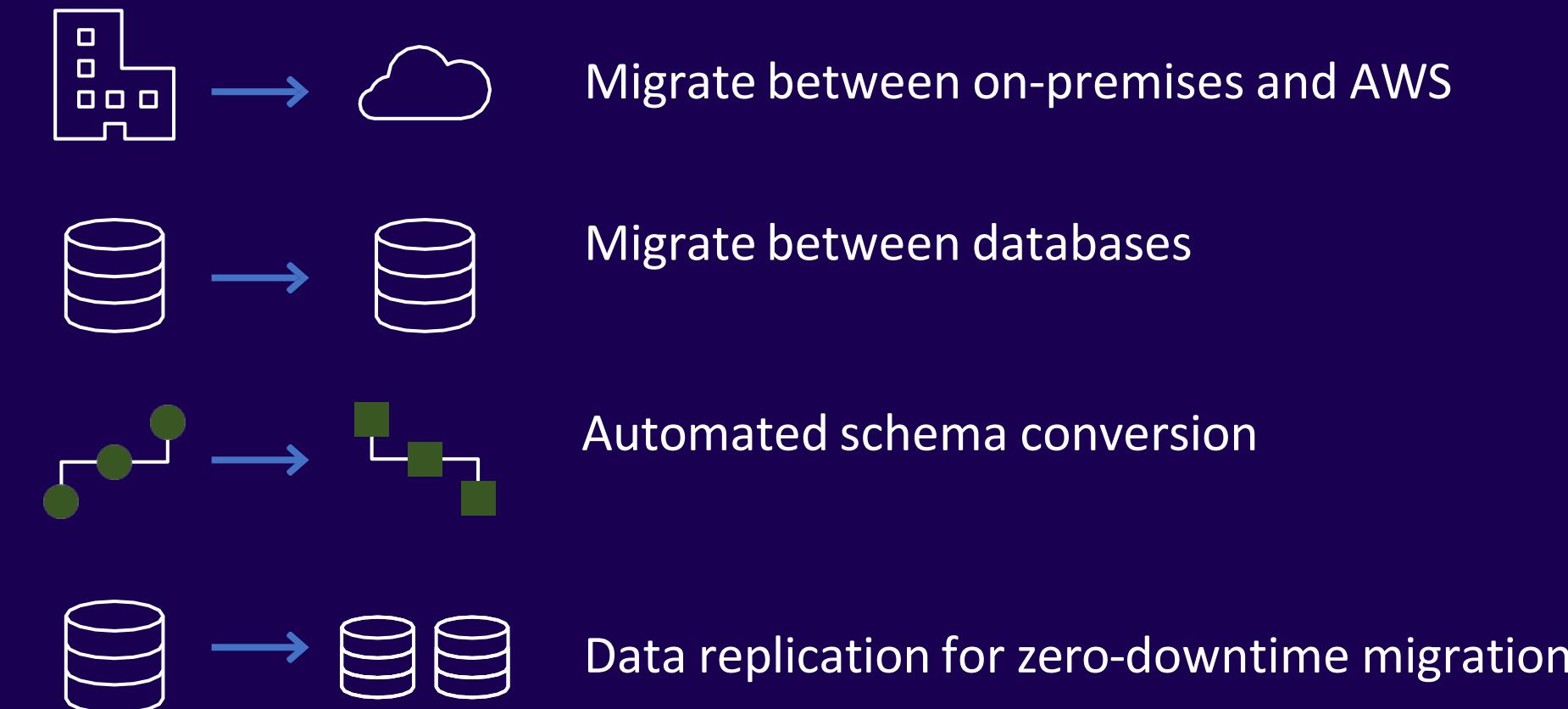
- Scale capacity with no downtime
- Multi-tenant proxy is highly available
- Scale target has warm buffer pool
- Shuts down when not in use



AWS Database Migration Service

Migrating
Databases to AWS

150,000+
Databases migrated



Quickly build new apps in the cloud

“Lift and shift” existing
apps to the cloud



Quickly build new
apps in the cloud



Modern apps create new requirements



Ride hailing



Media streaming



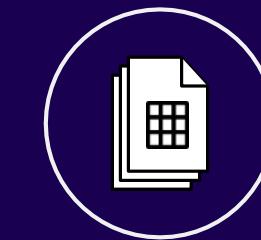
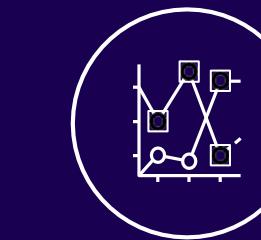
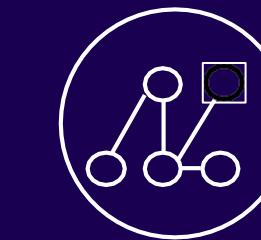
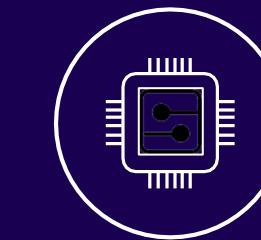
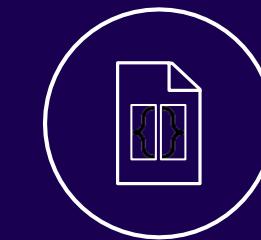
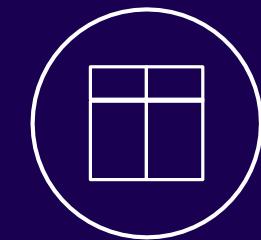
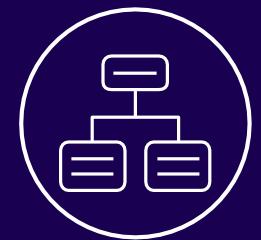
Social media



Dating

- Users: 1 million+
- Data volume: TB–PB–EB
- Locality: Global
- Performance: Milliseconds–microseconds
- Request rate: Millions
- Access: Web, mobile, IoT, devices
- Scale: Up-down, Out-in
- Economics: Pay for what you use
- Developer access: No assembly required

Common data categories and use cases



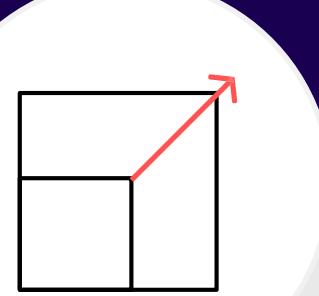
Relational	Key-value	Document	In-memory	Graph	Time-series	Ledger
Referential integrity, ACID transactions, schema-on-write	High throughput, low-latency reads and writes, endless scale	Store documents and quickly access querying on any attribute	Query by key with microsecond latency	Quickly and easily create and navigate relationships between data	Collect, store, and process data sequenced by time	Complete, immutable, and verifiable history of all changes to application data
Lift and shift, ERP, CRM, finance	Real-time bidding, shopping cart, social, product catalog, customer preferences	Content management, personalization, mobile	Leaderboards, real-time analytics, caching	Fraud detection, social networking, recommendation engine	IoT applications, event tracking	Systems of record, supply chain, health care, registrations, financial

Amazon DynamoDB

Fast and flexible key value database service for any scale



Performance at scale



Consistent, single-digit millisecond response times at any scale; build applications with virtually unlimited throughput

Serverless



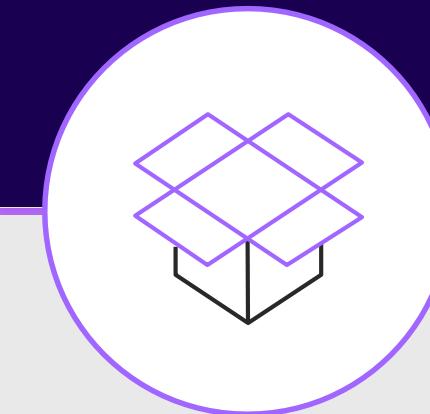
No server provisioning, software patching, or upgrades; scales up or down automatically; continuously backs up your data

Comprehensive security



Encrypts all data by default and fully integrates with AWS Identity and Access Management for robust security

Global database for global users and apps

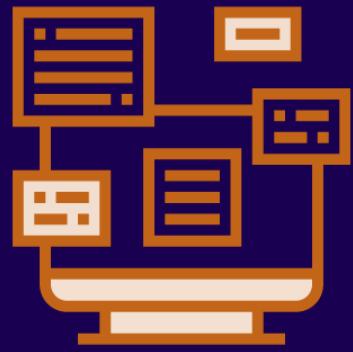


Build global applications with fast access to local data by easily replicating tables across multiple AWS Regions

Use cases for highly connected data



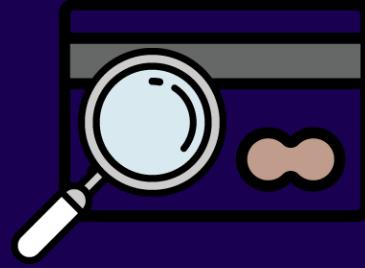
Social networking



Recommendations



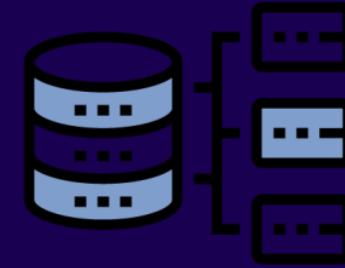
Knowledge graphs



Fraud detection



Life Sciences



Network & IT operations



Airbnb uses different databases based on the purpose

User search history: Amazon DynamoDB

- **Massive data volume**
- **Need quick lookups for personalized search**

Session state: Amazon ElastiCache

- **In-memory store for submillisecond site rendering**

Relational data: Amazon RDS

- **Referential integrity**
- **Primary transactional database**

Amazon RDS

CHOICE OF OPEN-SOURCE AND COMMERCIAL DATABASES

Cloud-native engine



Open-source engines



Commercial engines



Automatic failover
Backup and recovery
Cross-Region replication

Isolation and security
Industry compliance
Automated patching

Advanced monitoring
Routine maintenance
Push button scaling

Why Amazon RDS?

- Hardware monitoring and maintenance
- OS monitoring and maintenance
- SQL Server monitoring and maintenance
- Minor version upgrade
- Automated backups and PiTR
- High availability
- In-Region and cross-Region disaster recovery (DR)
- In-Region and cross-Region read scale-out
- Performance monitoring tools

Amazon RDS Custom for SQL Server

(new since 12/1/2021)

SQL Server on Amazon EC2

Self-managed

- ✓ Control over all aspects of your infrastructure, OS, and SQL Server configuration
- ✓ Choose any version, edition, and OS configuration needed
- ✓ Choose any underlying cloud infrastructure across compute, storage, and networking
- ✓ Surround SQL Server with cloud services such as monitoring, alarming, and actions (Amazon CloudWatch, Amazon SNS, AWS Lambda)
- ✓ Ability to mount shared storage volumes such as Amazon FSx for Windows File Server and Amazon FSx for NetApp ONTAP

RDS Custom for SQL Server

Shared management responsibility

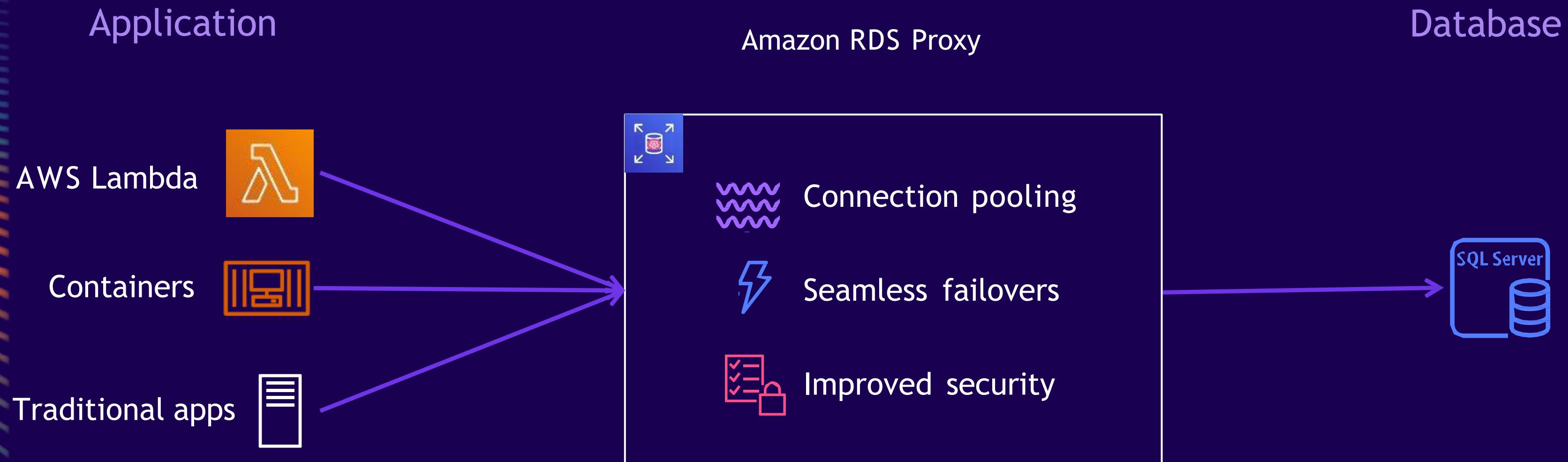
- ✓ Sysadmin and OS admin access
- ✓ Custom drivers and components (for example, CLR, MS replication)
- ✓ Enables all SQL Server security roles
- ✓ Third-party applications, for example, MS SharePoint, MS Dynamics
- ✓ User automation can be inserted into Amazon automation
- ✓ Flexible HA and DR configurations
- ✓ Unrestricted BI stack
- ✓ xp_cmdshell
- ✓ Resource governor

RDS for Microsoft SQL Server

AWS-managed

- ✓ Fully managed SQL Server database experience
- ✓ Runs on native Windows Server and SQL Server
- ✓ Automated provisioning, monitoring, backup, restore, point-in-time recovery, and scalable compute
- ✓ Access the latest major/minor versions
- ✓ Amazon RDS Multi-AZ provides one standby and up to five in-Region read replicas
- ✓ AWS provides a DB endpoint to connect to the database and read replicas
- ✓ 99.95% SLA

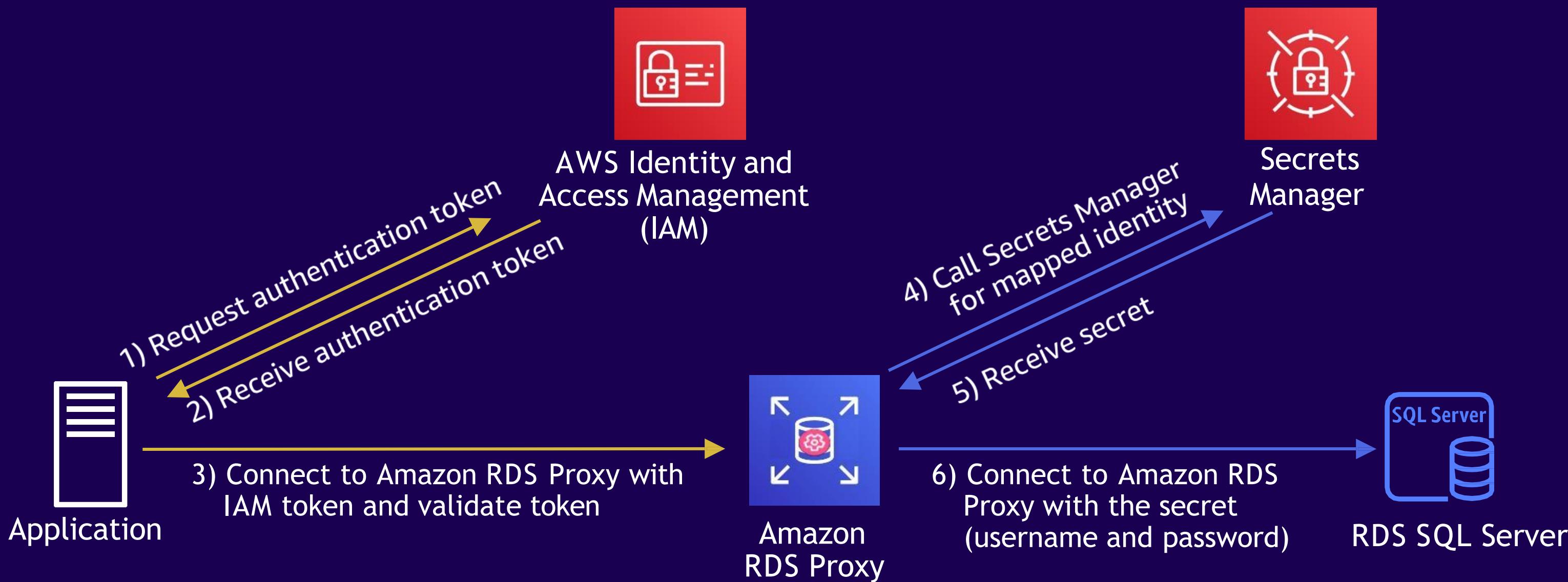
RDS Proxy for SQL Server



Improved application security

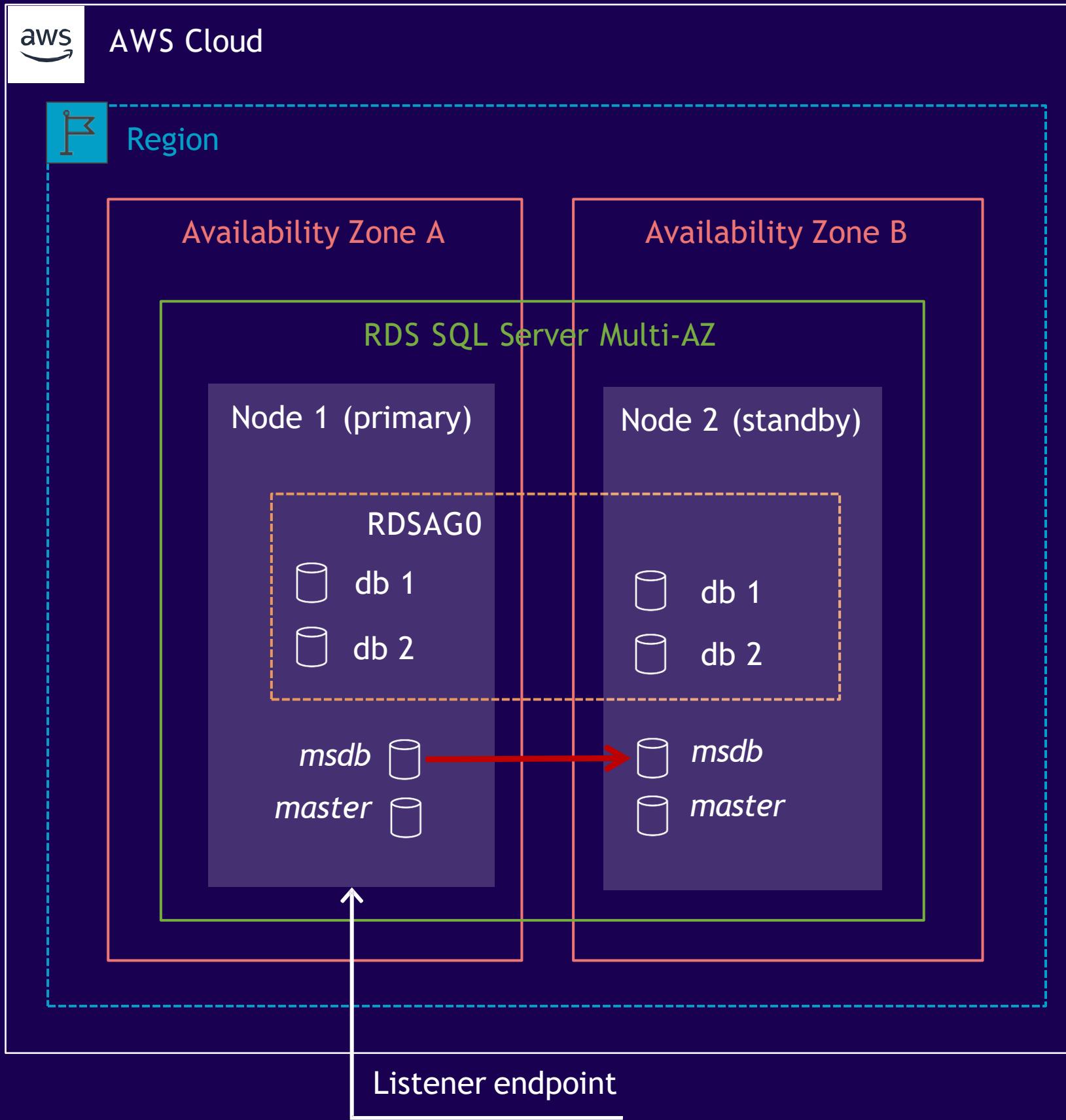
Microsoft
SQL Server

Centrally manage database credentials using AWS Secrets Manager



Multi-AZ and SQL Agent job replication

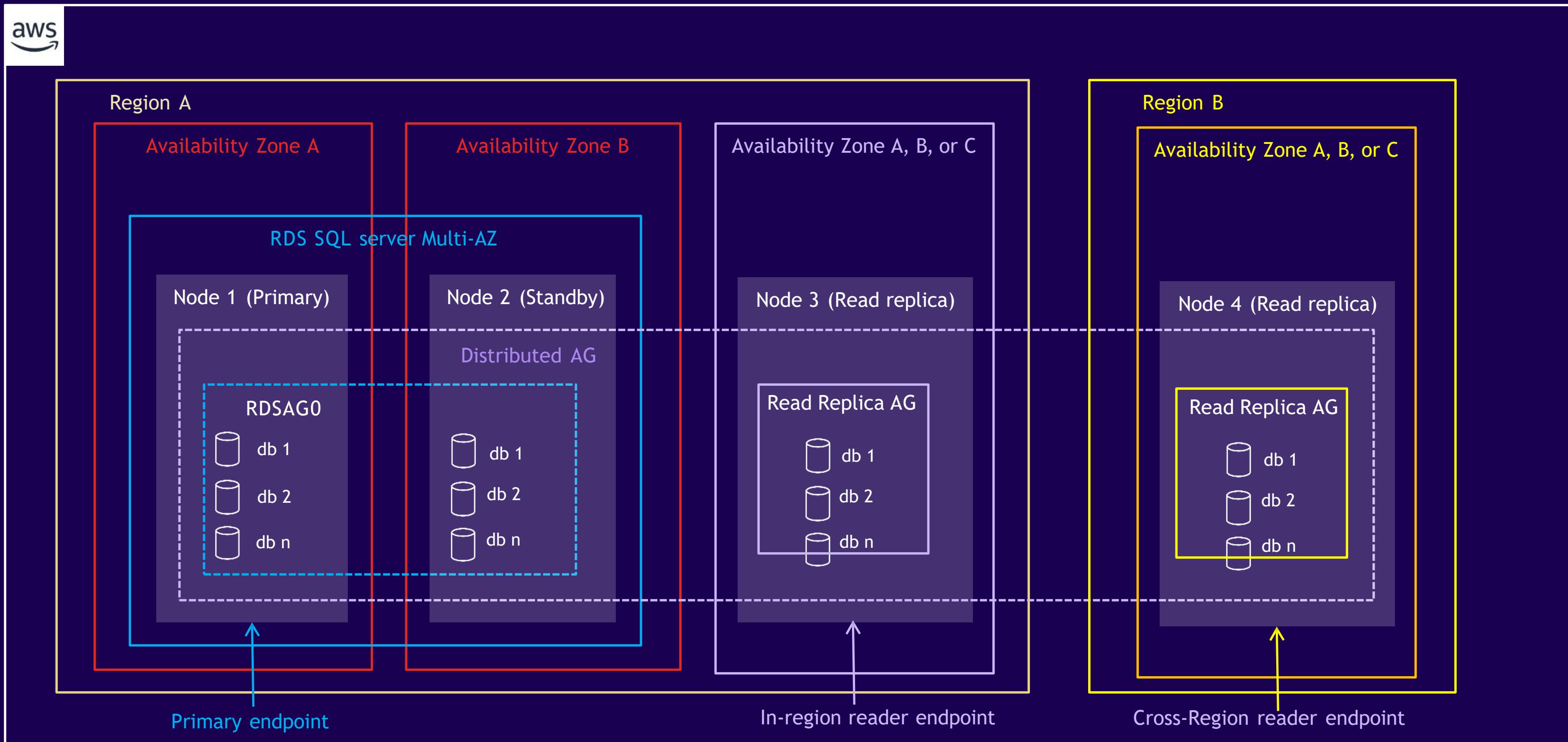
Microsoft
SQL Server



- Both Enterprise and Standard
- Built on Always On AG (2016 and up)
- Built on mirroring (2012 and 2014)
- Backup retention > 0
- Full recovery mode
- Synchronous replication
- Automatic and manual failover
- 5-6 seconds failover time*
- No read traffic
- Ability to add up to 5 read replicas (Enterprise only)
- SQL Agent jobs synchronization (max: 100 jobs)

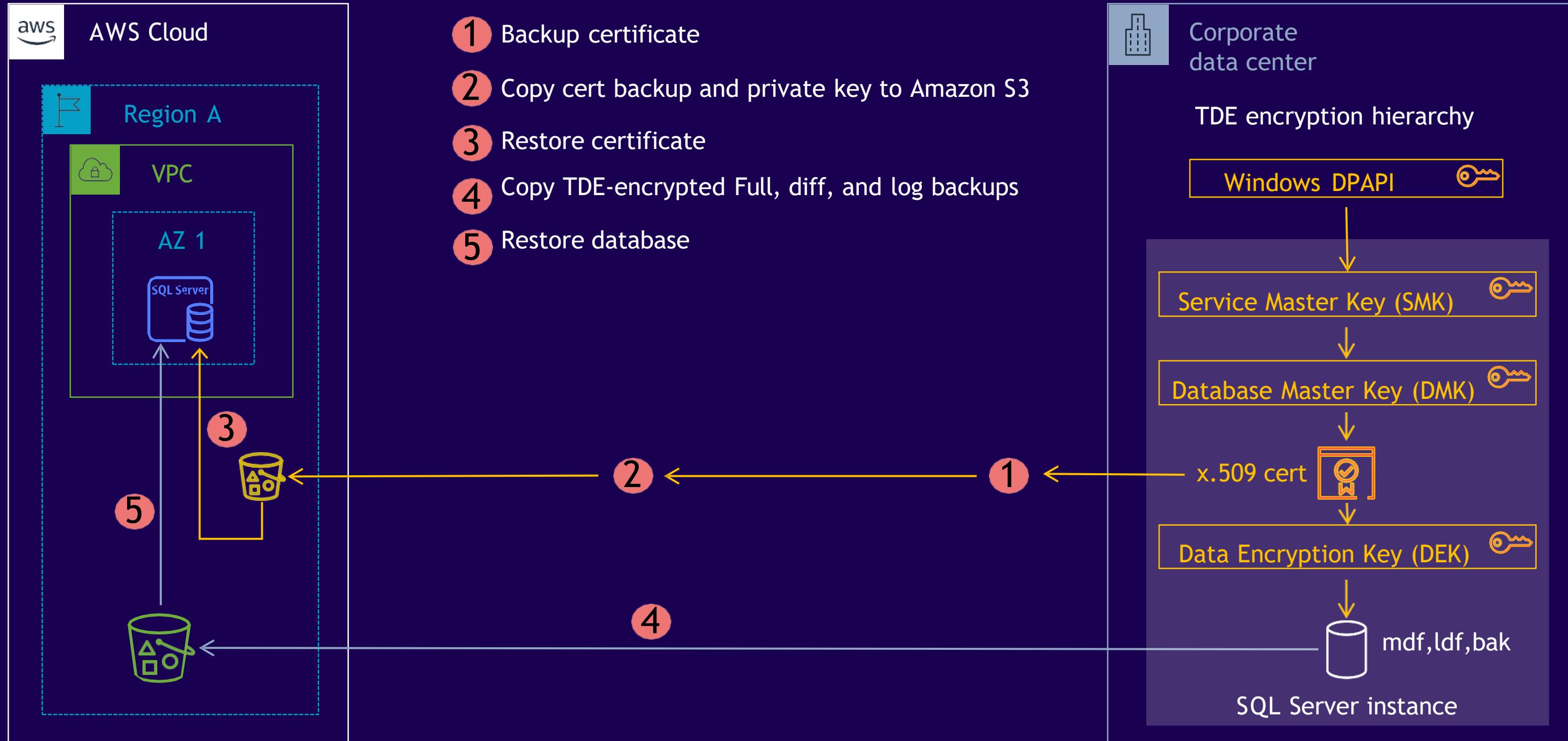
*Crash recovery times not included; MultiSubnetFailover = True

In-Region and cross-Region read replicas



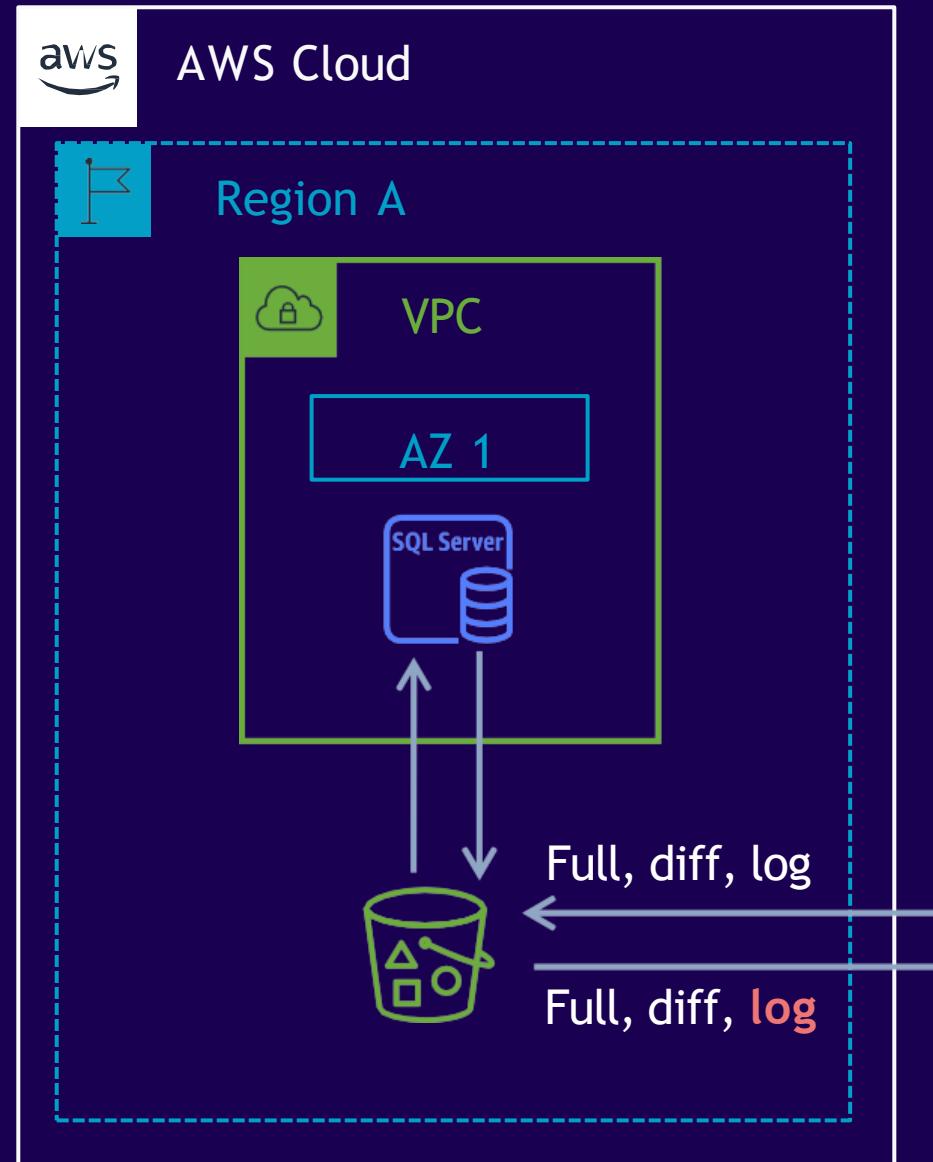
Database native backup and restore with TDE

Microsoft
SQL Server



Native backup and t-log access

Microsoft
SQL Server



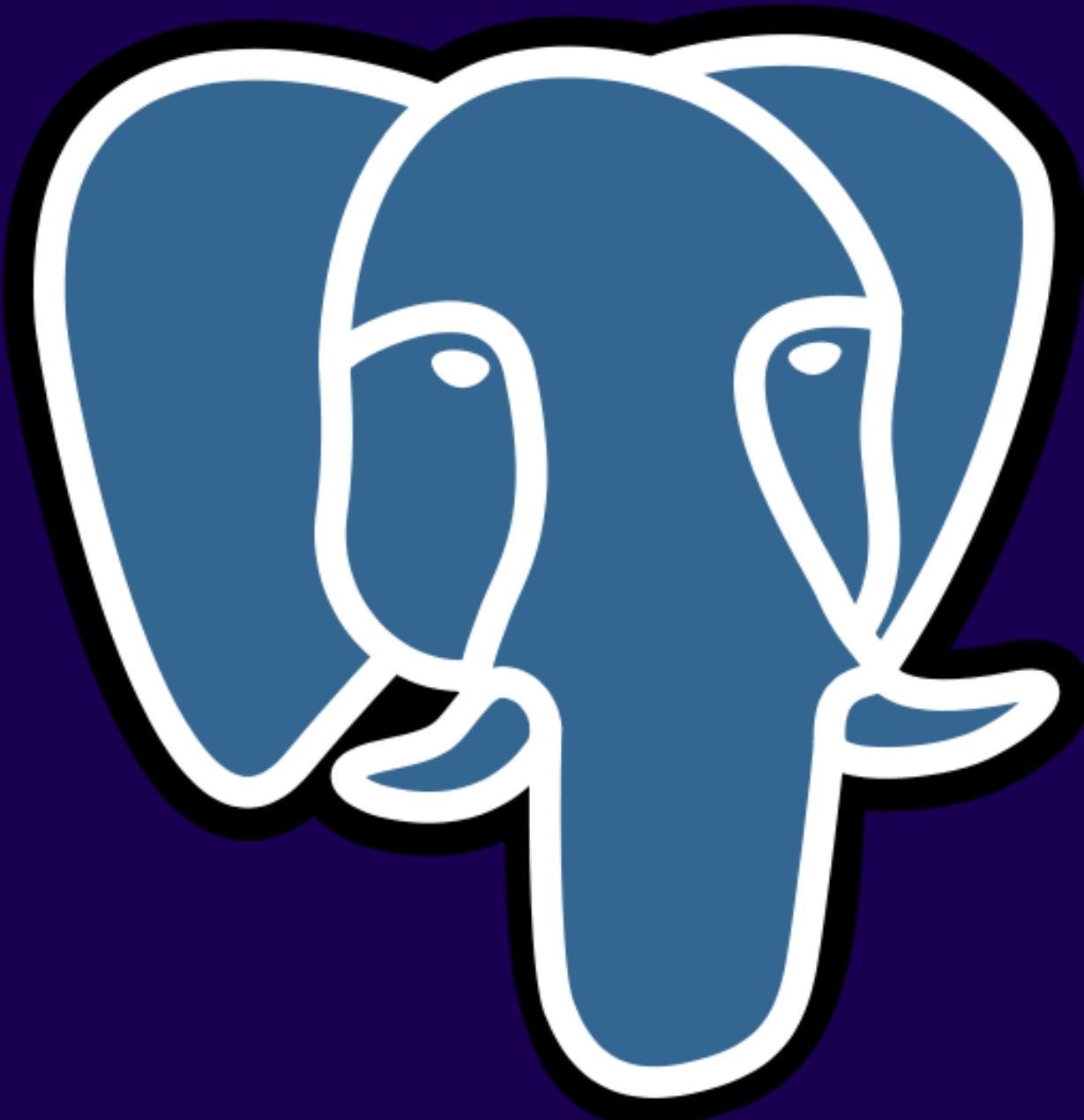
- Backup and restore directly to and from Amazon S3 bucket
- Supports compression
- Full, diff, and t-log backup and restores
- Multi-file backup/restore
- Database level PiTR
- Self-service log shipping

Supported Versions

Microsoft
SQL Server

- 2012
 - 2014
 - 2016
 - 2017 and
 - 2019
-
- Express
 - Web
 - Standard
 - Enterprise
-
- License Included Model

Why PostgreSQL?



Open source

- In active development for more than 30 years
- Controlled by a community, not a single company

Performance and scale

- Extensive index support for a number of use cases
- Parallel processing for complex queries
- Native partitioning for large tables

Amazon RDS for PostgreSQL

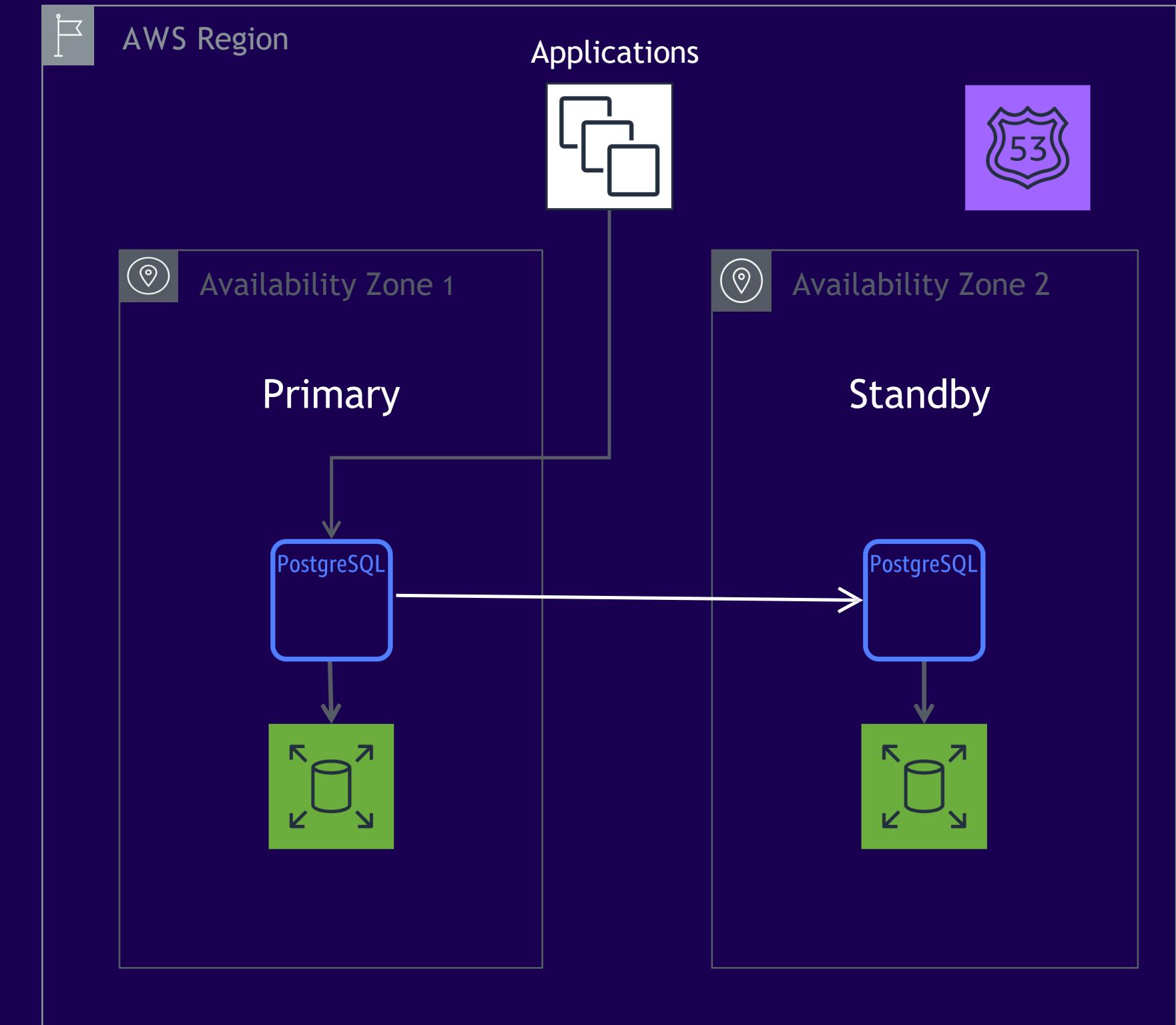
PostgreSQL

PostgreSQL community version with easy configuration and management

Supports 11 to 17 (10 and 9.6 are deprecated - 18 is beta)

Supports TLE for PostgreSQL

High availability across two availability zones



Amazon Aurora with PostgreSQL compatibility

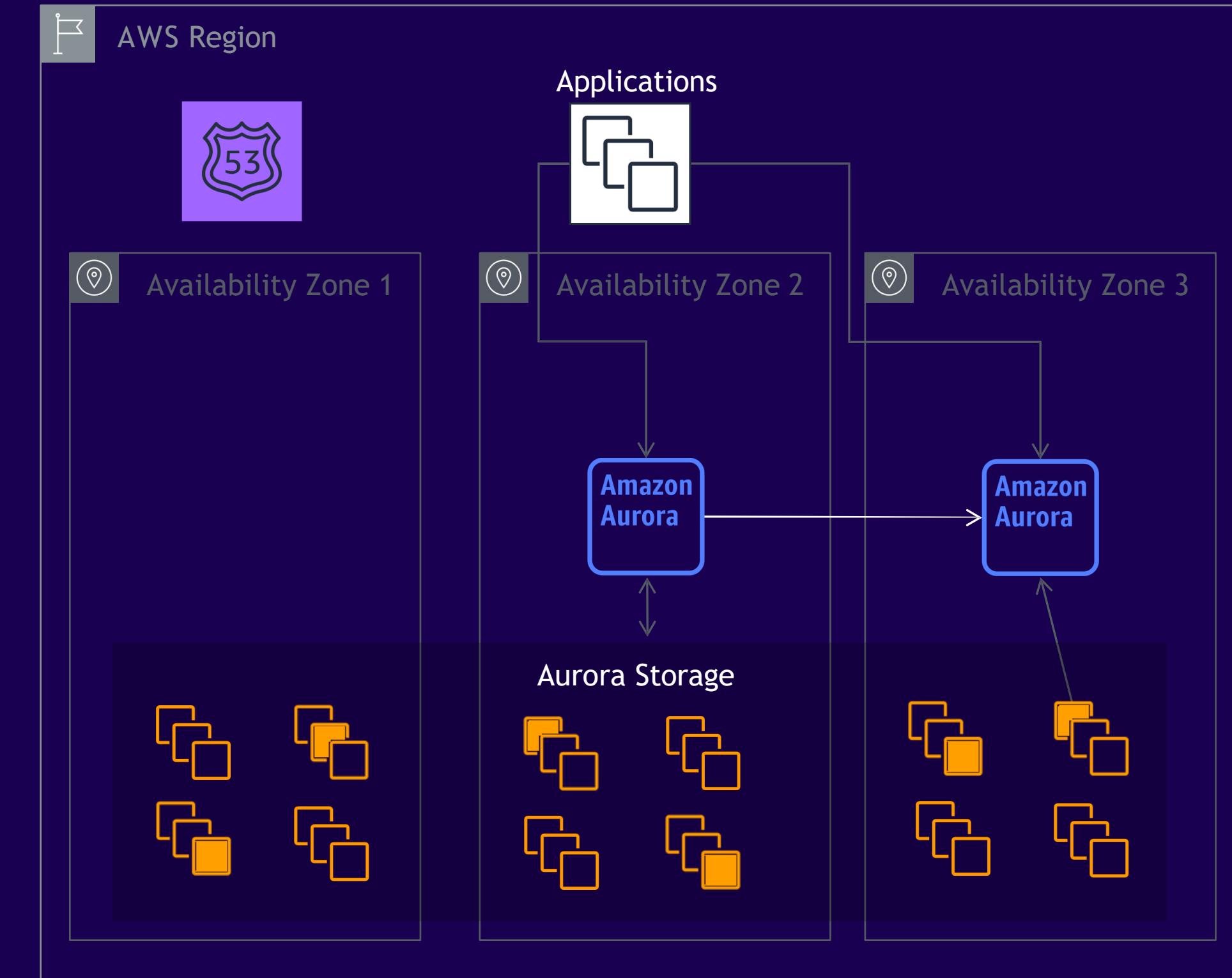
Built from the ground up to leverage AWS services

Supports 10 to 15

Up to 2-3x better throughput on the same instance sizes

Highly-available, durable, and fault-tolerant storage layer with 6-way persistence across 3 Availability Zones

Scalable up to 128 TiB



Amazon RDS for PostgreSQL: Instance types

T family

- **Burstable instances**
- 1 vCPU/1 GB RAM > 8 vCPU
32 GB RAM
- Moderate networking performance
- Good for smaller or variable workloads
- T2.micro is eligible for the AWS Free Tier

M family

- **General purpose instances**
- 2 vCPU/8 GiB RAM > 64 vCPU
256 GiB RAM
- High-performance networking
- Good for running CPU-intensive workloads
- M5 offers up to 96 vCPU / 384 GiB RAM

R family

- **Memory-optimized instances**
- 2 vCPU/16 GiB RAM > 64 vCPU 488 GiB RAM
- High-performance networking
- Good for query-intensive workloads or high connection counts
- R5 offers up to 96 vCPU 768 GiB RAM

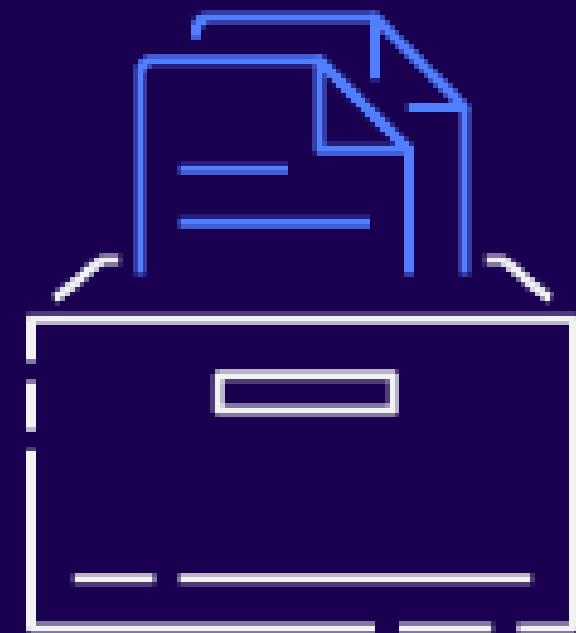
High-performance database storage

General purpose (GP2)

- SSD storage
- Auto scale up to 64 TiB
- Latency in milliseconds
- IOPS determined by volume size
- Affordable performance

Provisioned IOPS (IO1)

- SSD storage
- Auto scale up to 64 TiB
- Single digit millisecond latencies
- Maximum of 80 K IOPS
- Delivers within 10% of the IOPS performance, 99.9% of the time
- High performance and consistency



High availability for RDS PostgreSQL

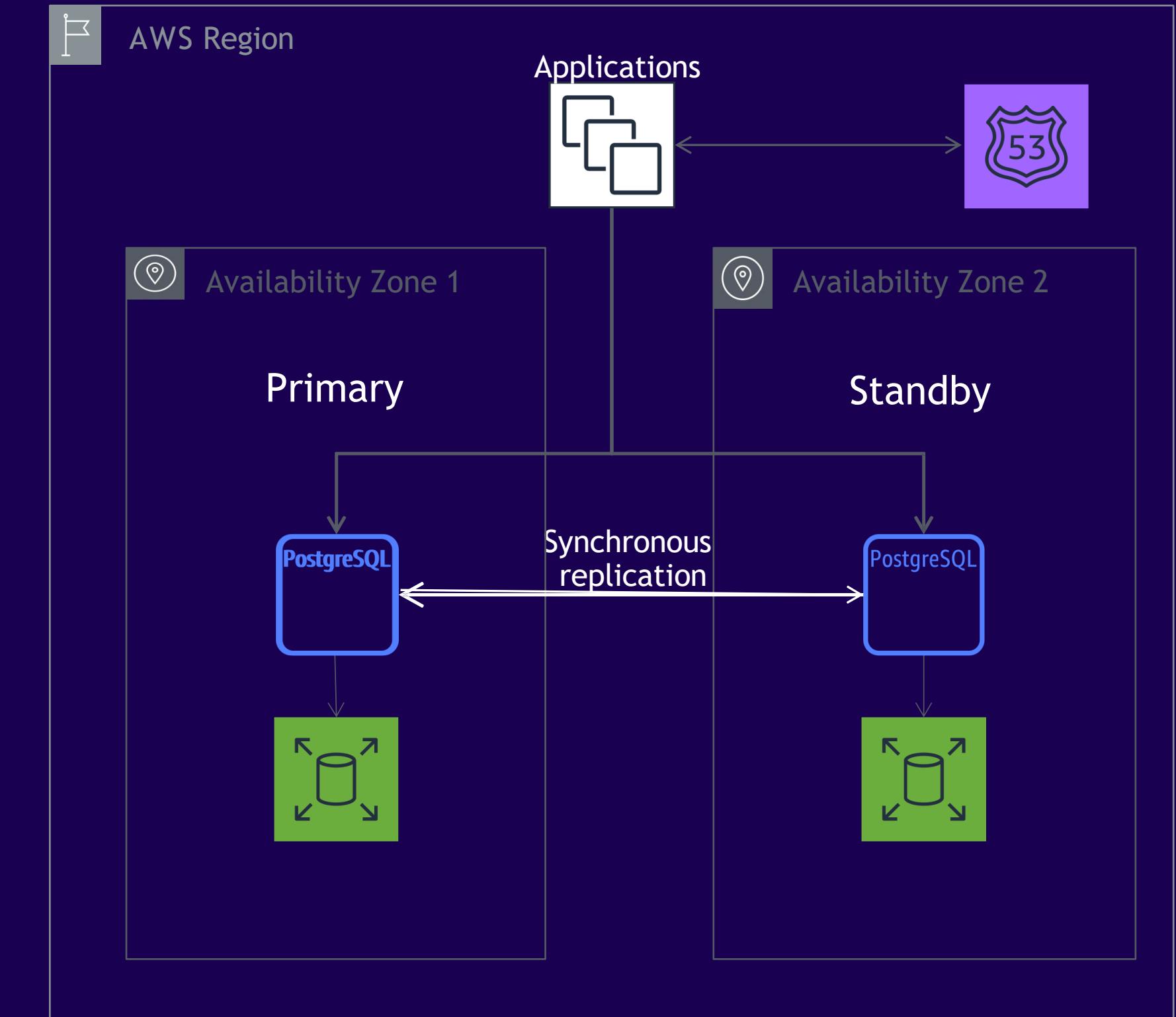
PostgreSQL

Multi-AZ configurations provide a fault-tolerant solution for PostgreSQL

Each host synchronously maintains a set of volumes with a full copy of the data

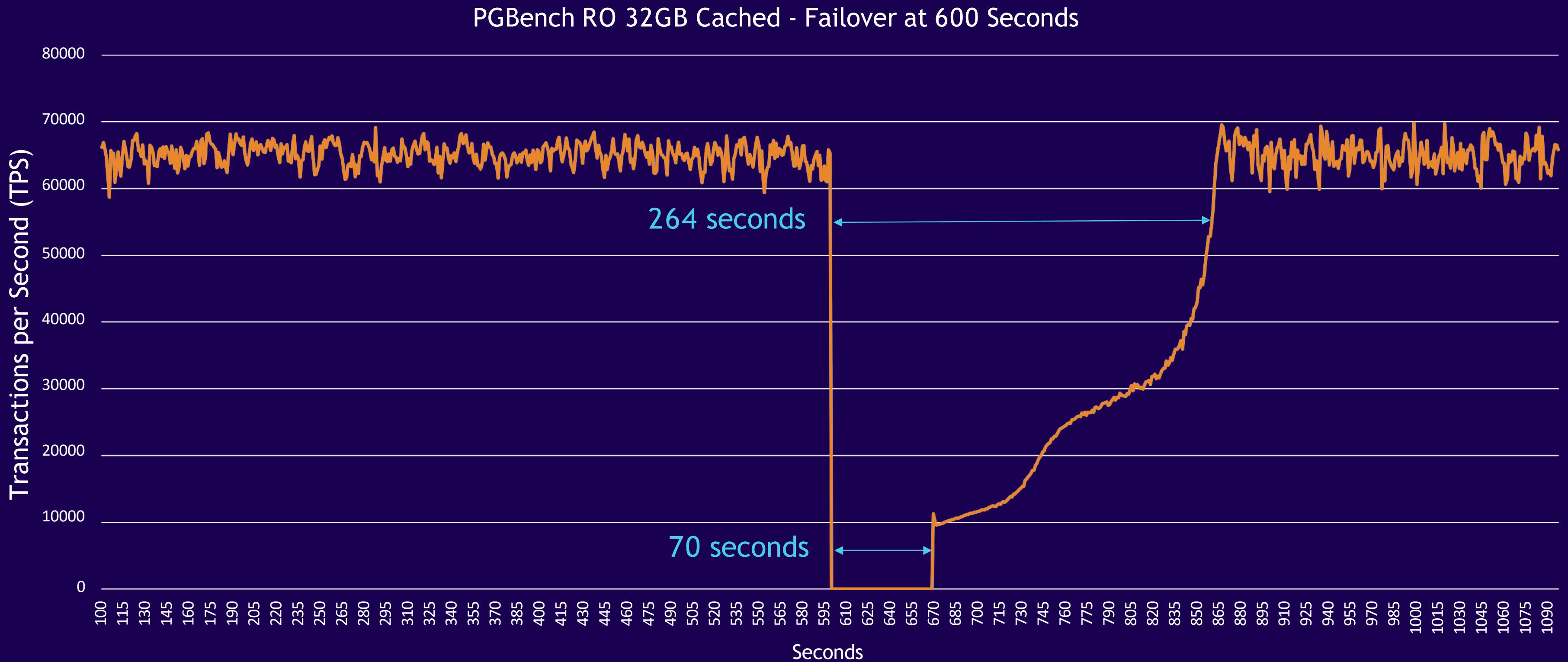
Redirection to the new primary instance is provided through DNS (Route 53)

Detects infrastructure issues, not database engine problems



Cache warming

PostgreSQL

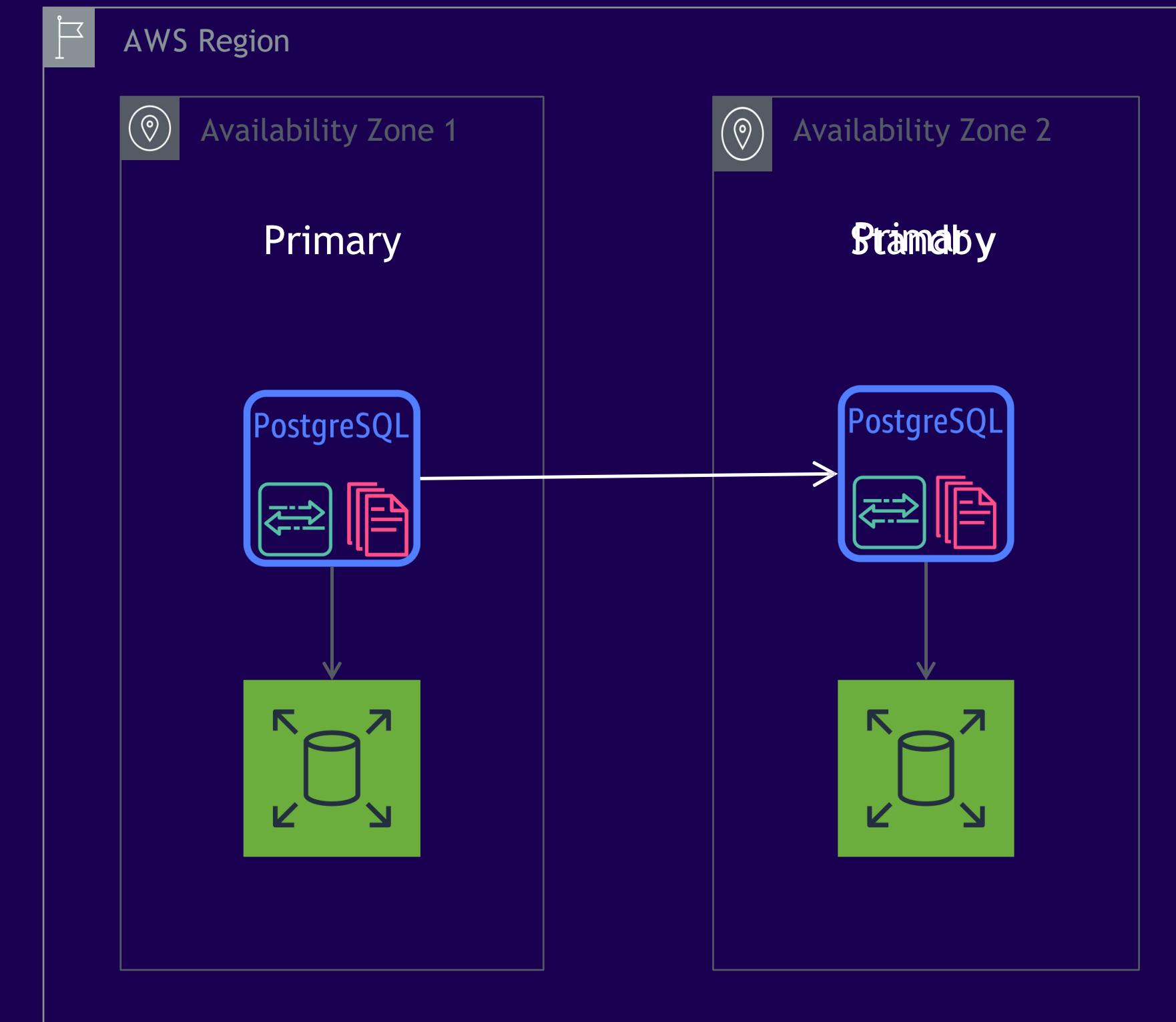


Pg_prewarm

Extension available in all supported versions of PostgreSQL

Can manually load tables and indexes into cache

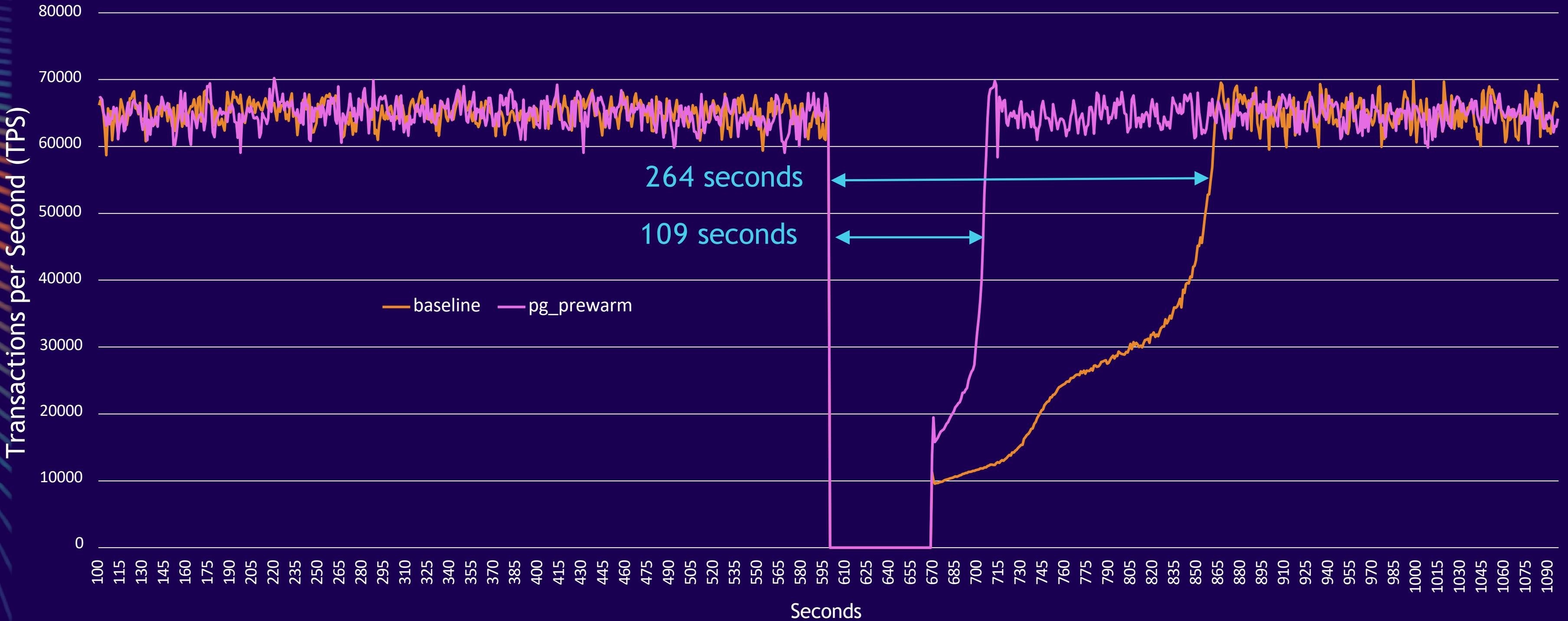
PostgreSQL 11 introduced auto prewarm to restore the cache after a restart or failover



Automatic cache warming

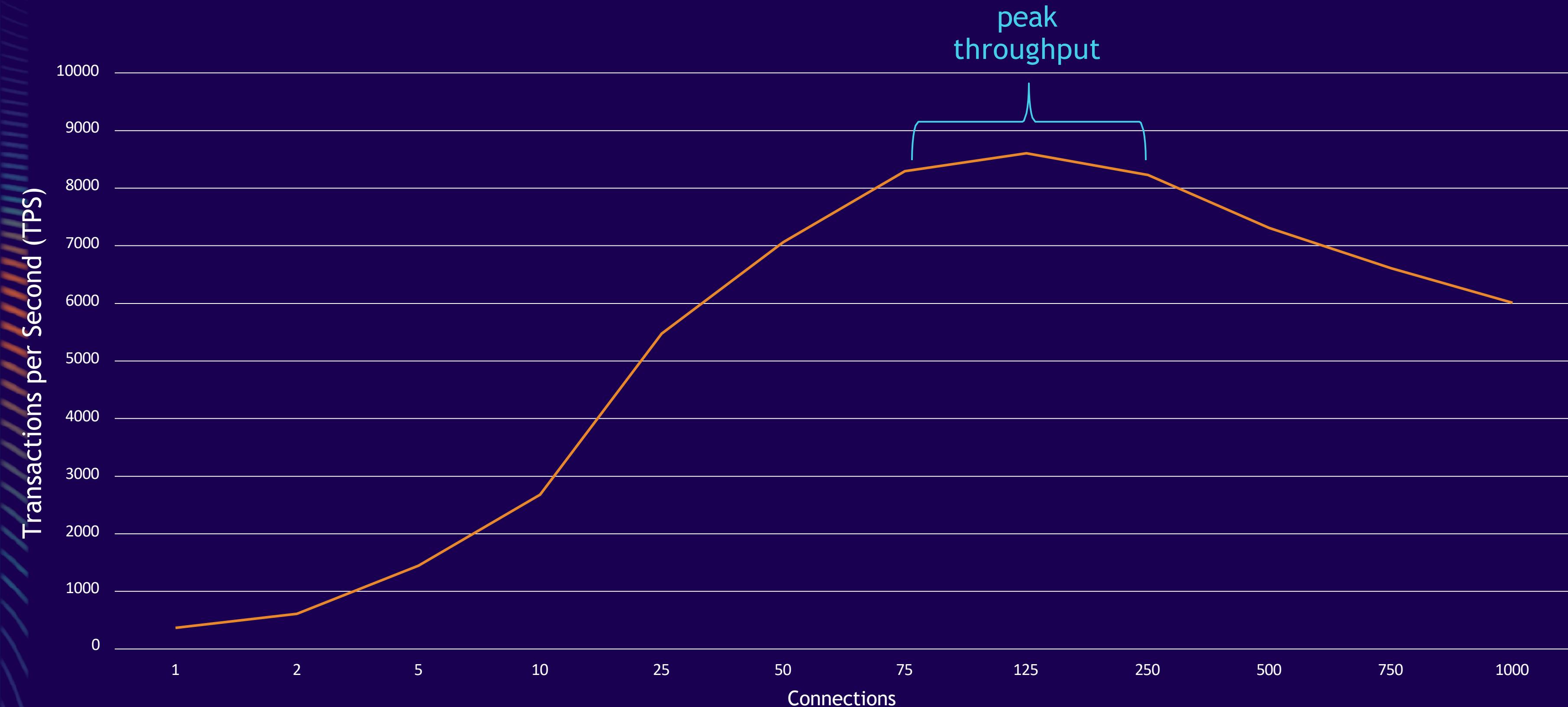
PostgreSQL

PGBench RO 32GB Cached - Failover at 600 Seconds



Connection scaling

PostgreSQL



Amazon RDS Proxy

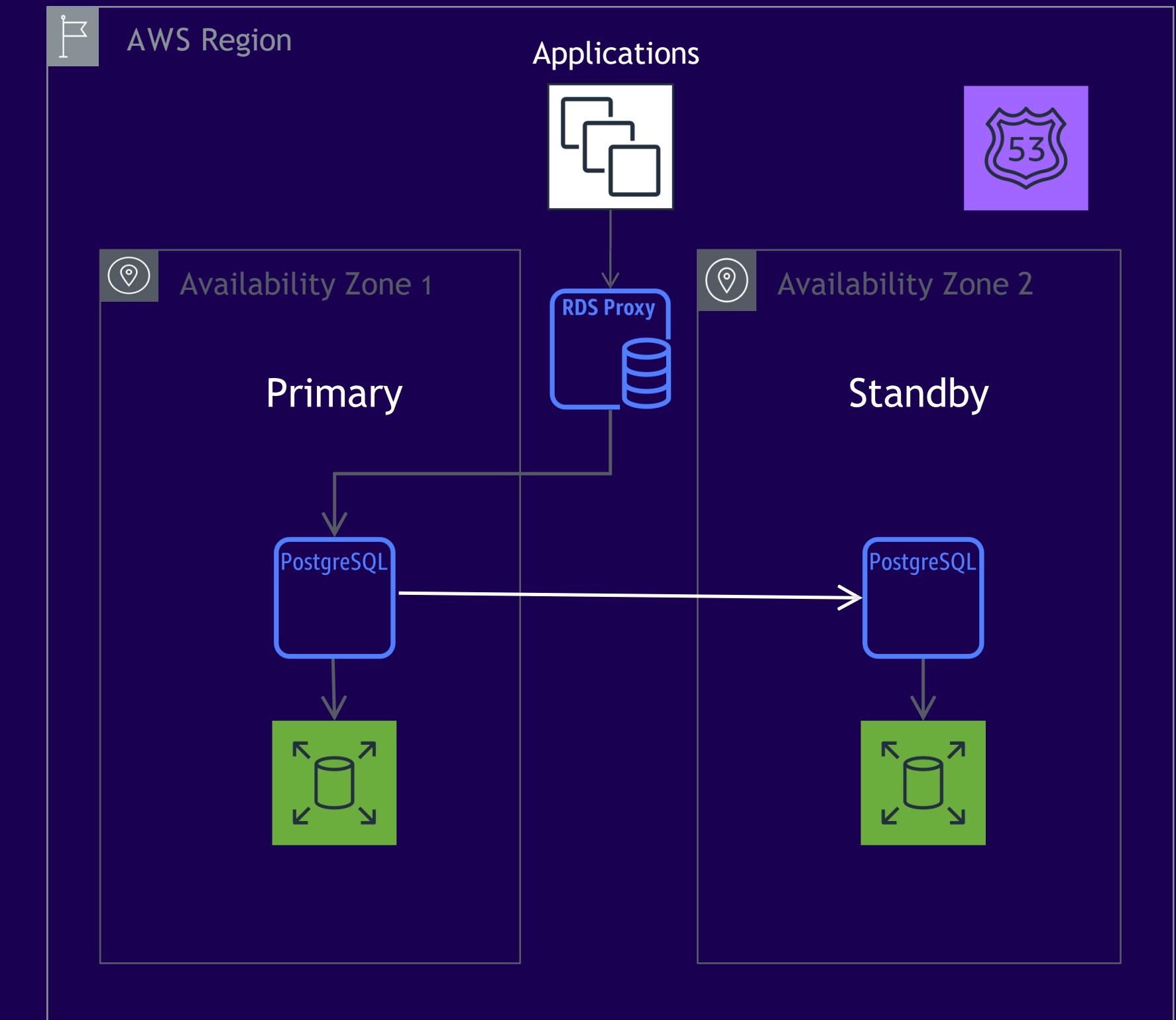
PostgreSQL

Fully managed database proxy

Pools and shares database connections

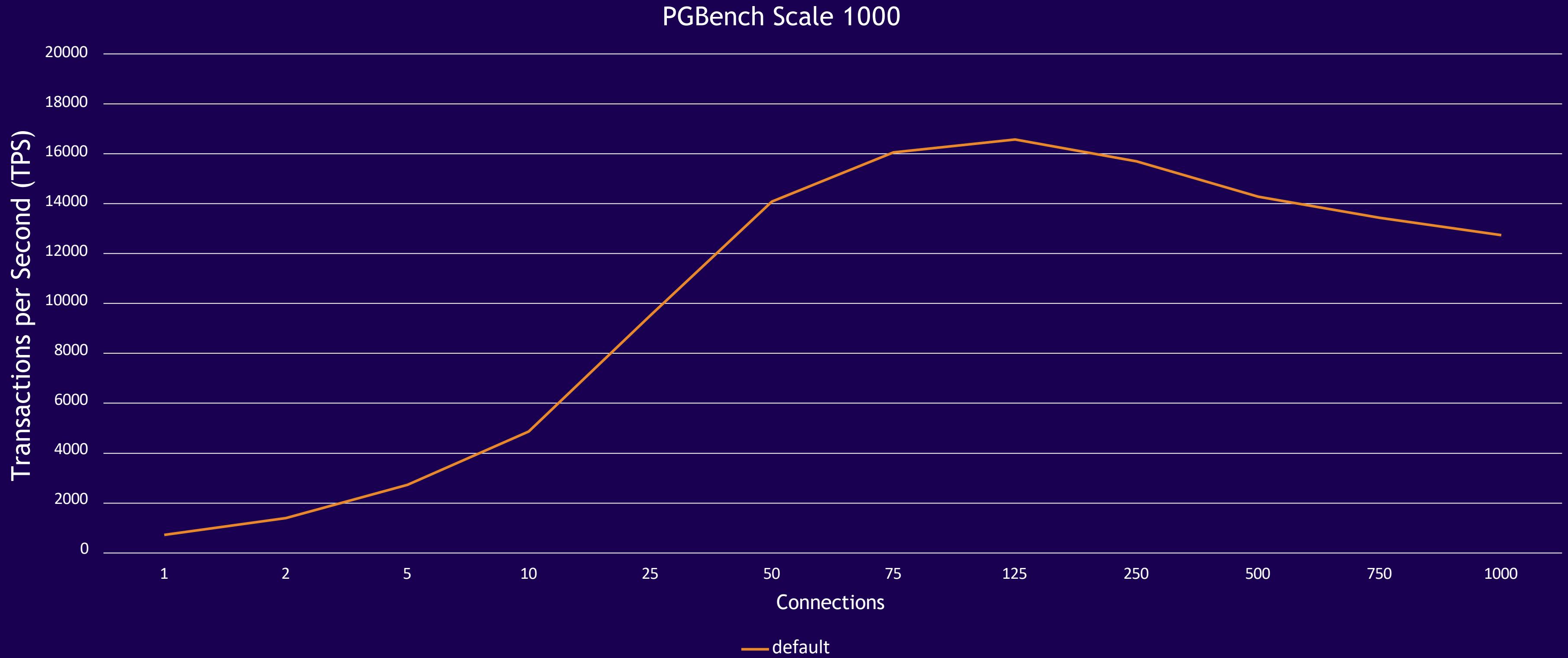
Increases application availability

Reduces database failover times



Parameter changes

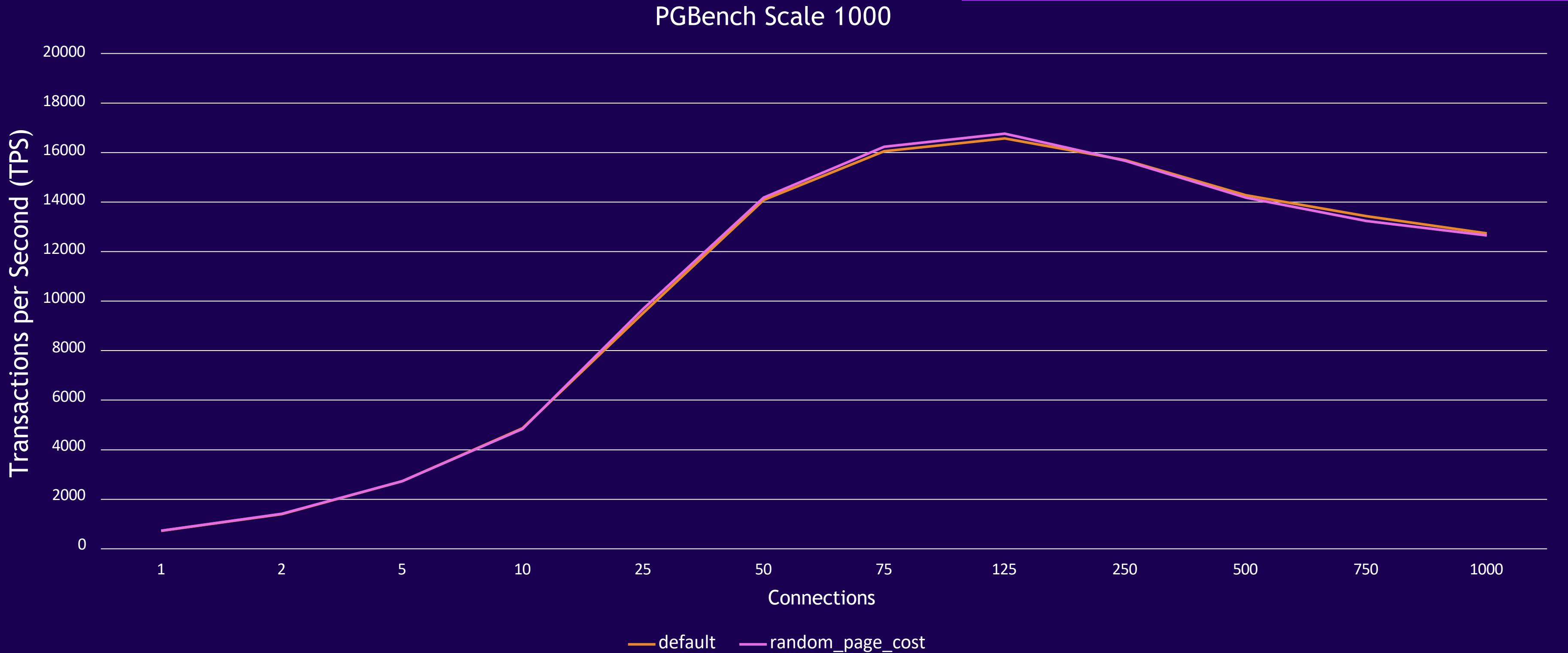
PostgreSQL



Parameter changes

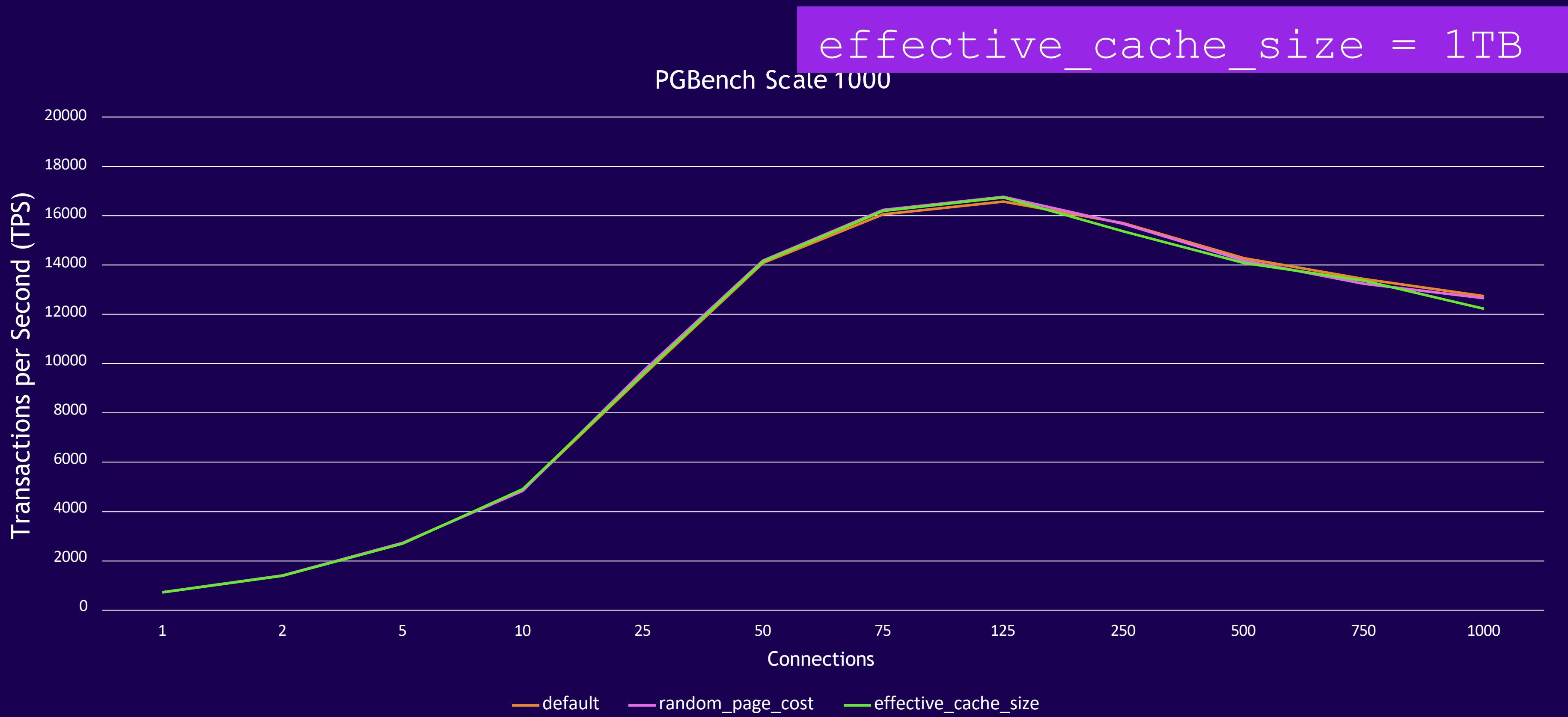
PostgreSQL

random_page_cost = 1



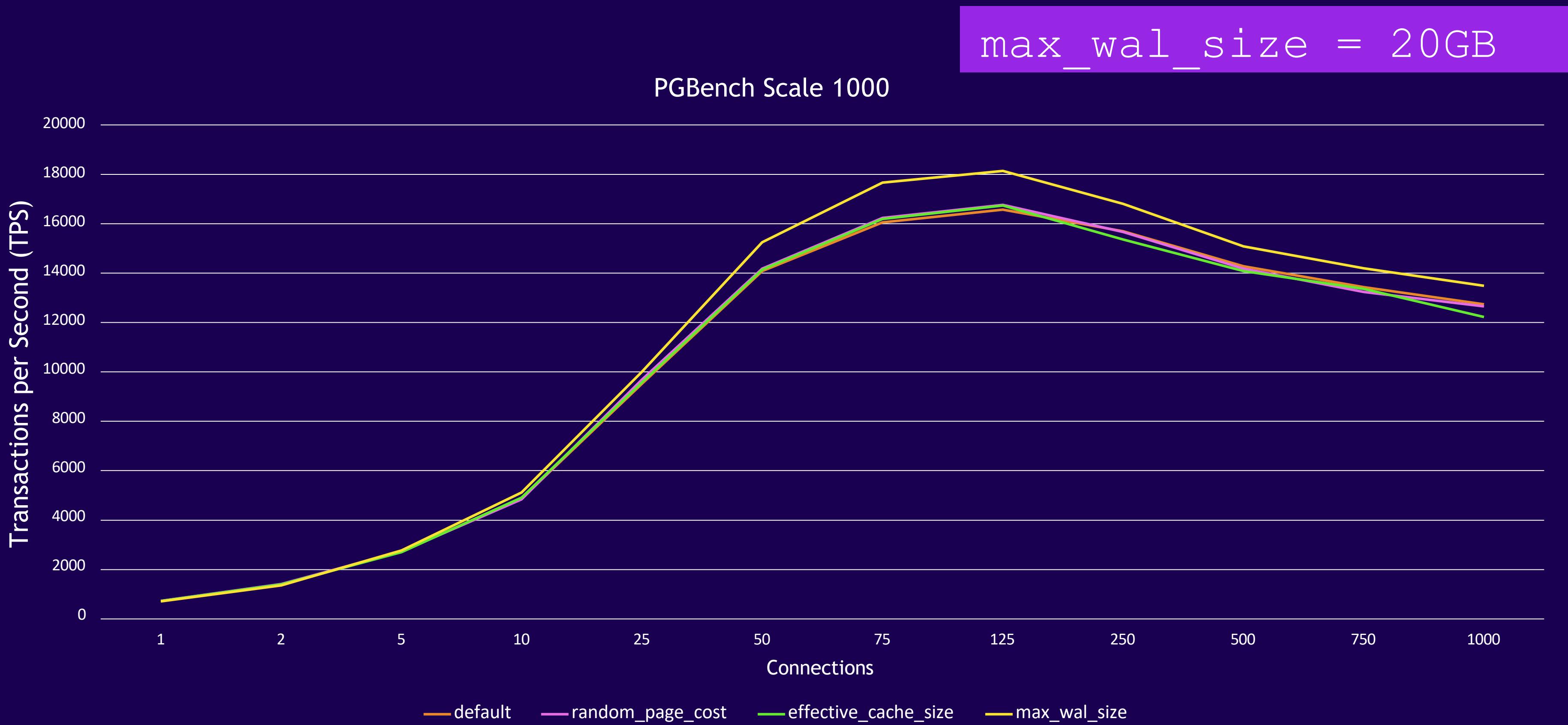
Parameter changes

PostgreSQL

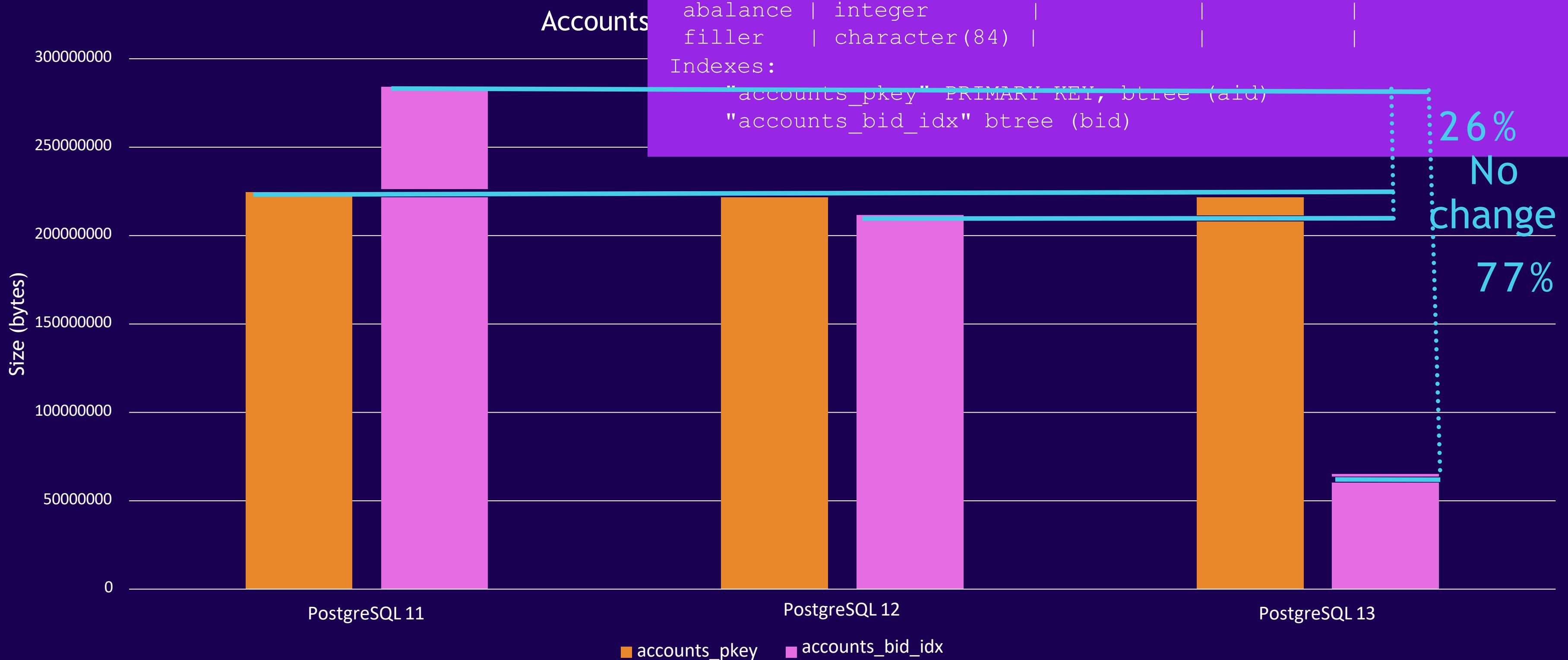


Parameter changes

PostgreSQL



Index sizes

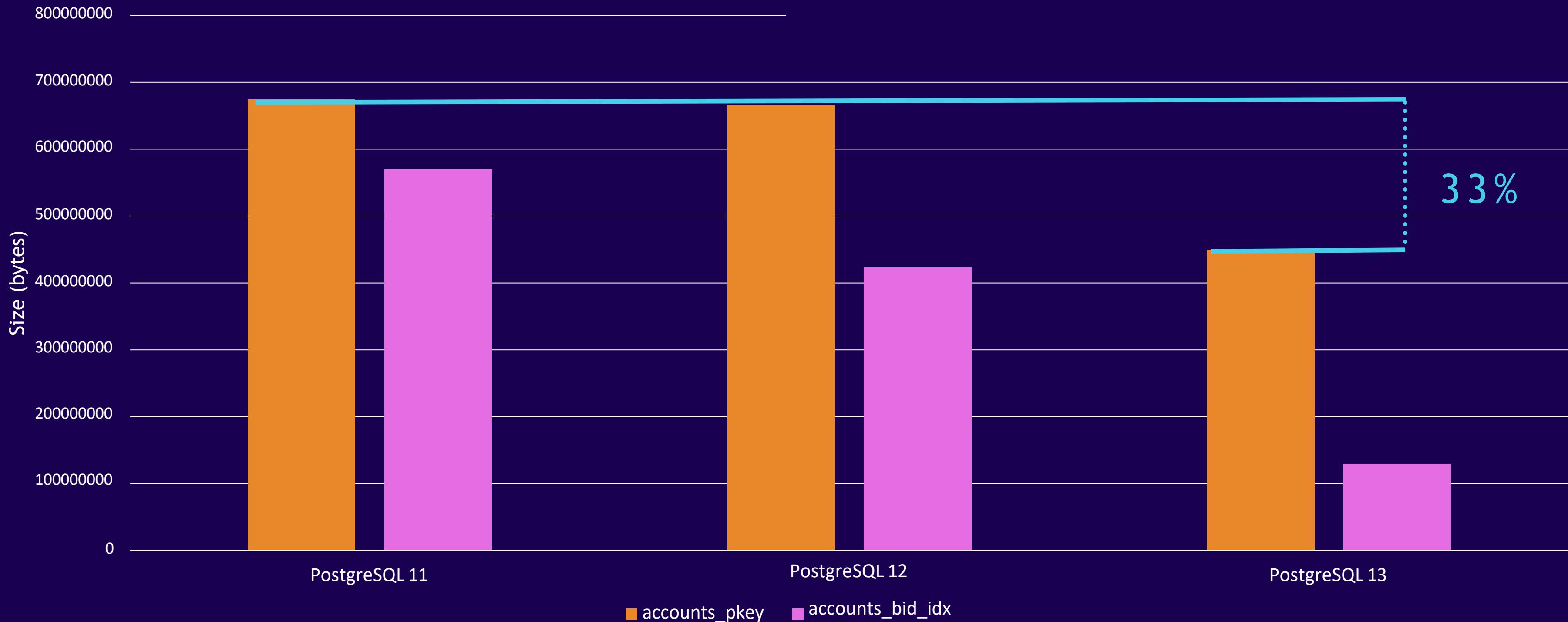


Index sizes

PostgreSQL

```
UPDATE accounts  
SET filler = 'reinvent';
```

Accounts table after backfilling all rows



Monitoring

Enhanced monitoring for Amazon RDS

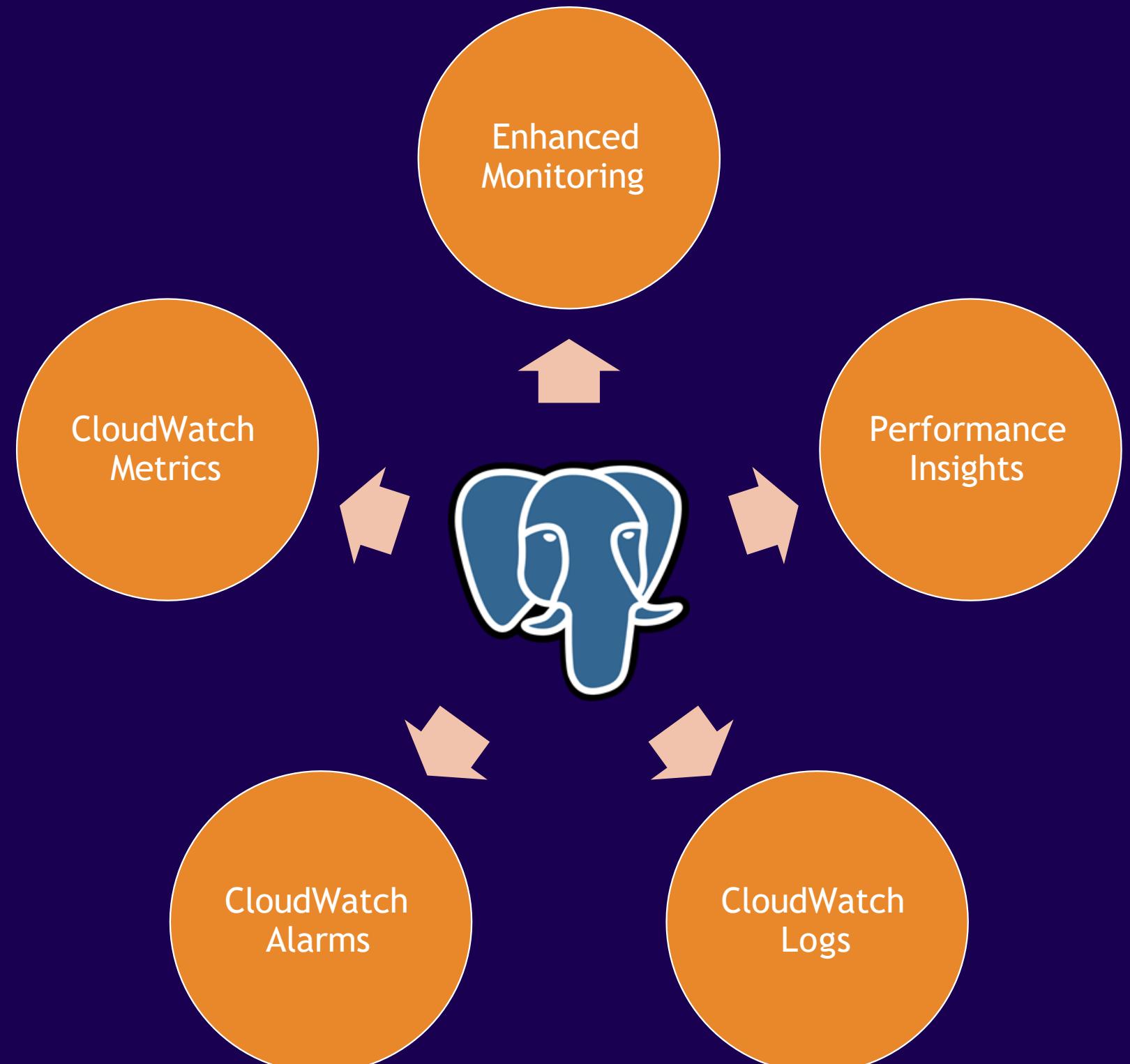
- Access to over 50 CPU, memory, file system, and disk I/O metrics

Amazon CloudWatch Metrics

- Displayed in the Amazon RDS console or in personalized CloudWatch dashboards

Amazon CloudWatch Alarms

- Alarms triggered based on metric values crossing configurable thresholds



CPU intensive functions

Many applications put complex logic in stored procedures

Some are CPU intensive

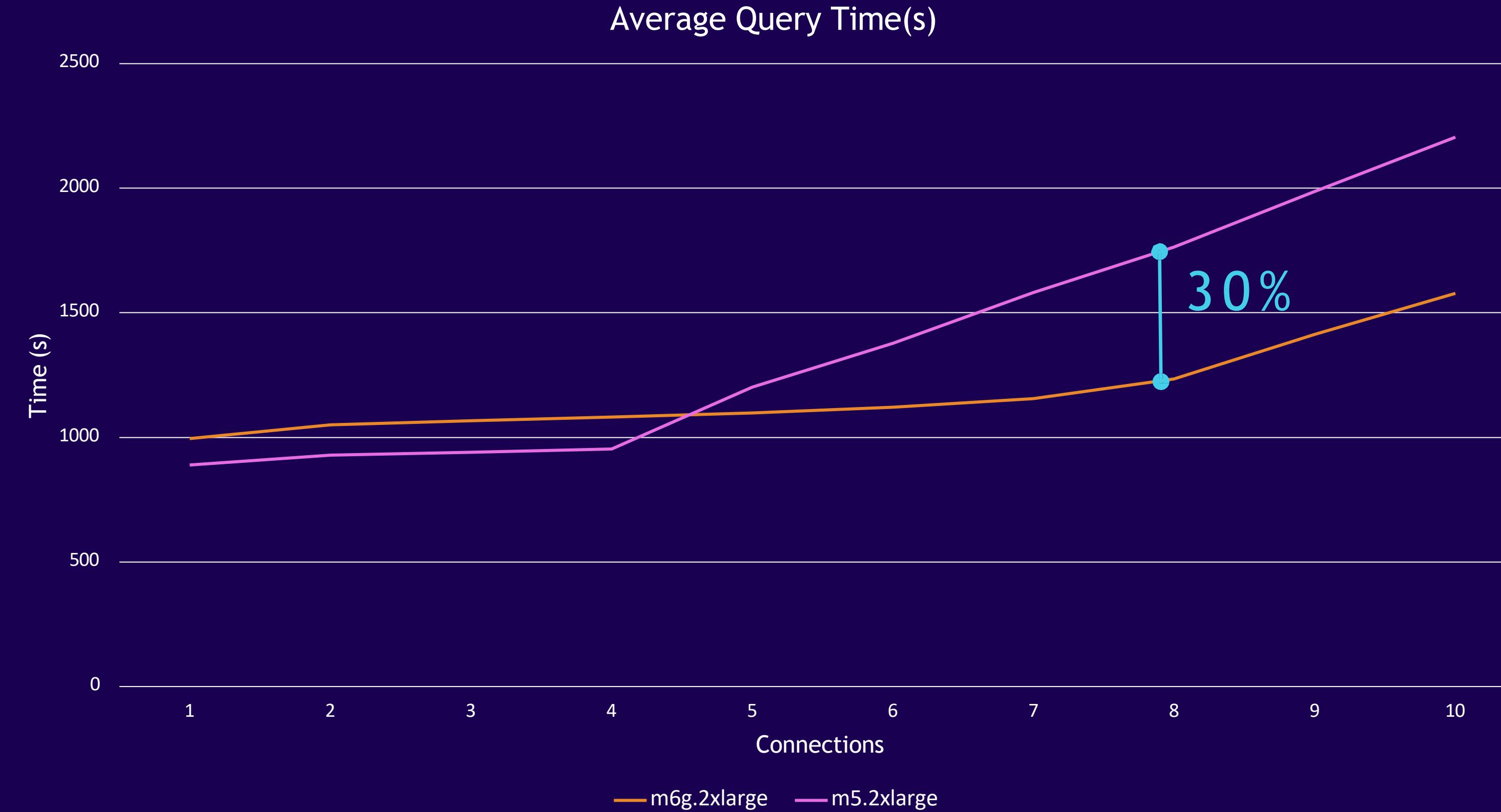
- Geospatial
- Full-text search

```
CREATE FUNCTION fib(n int)
    RETURNS int AS
$$
BEGIN
    IF n <= 1 THEN
        RETURN n;
    END IF;

    RETURN fib(n-1) + fib(n-2);
END;
$$ LANGUAGE plpgsql;
```

CPU intensive functions on Graviton

PostgreSQL



Amazon RDS performance insights

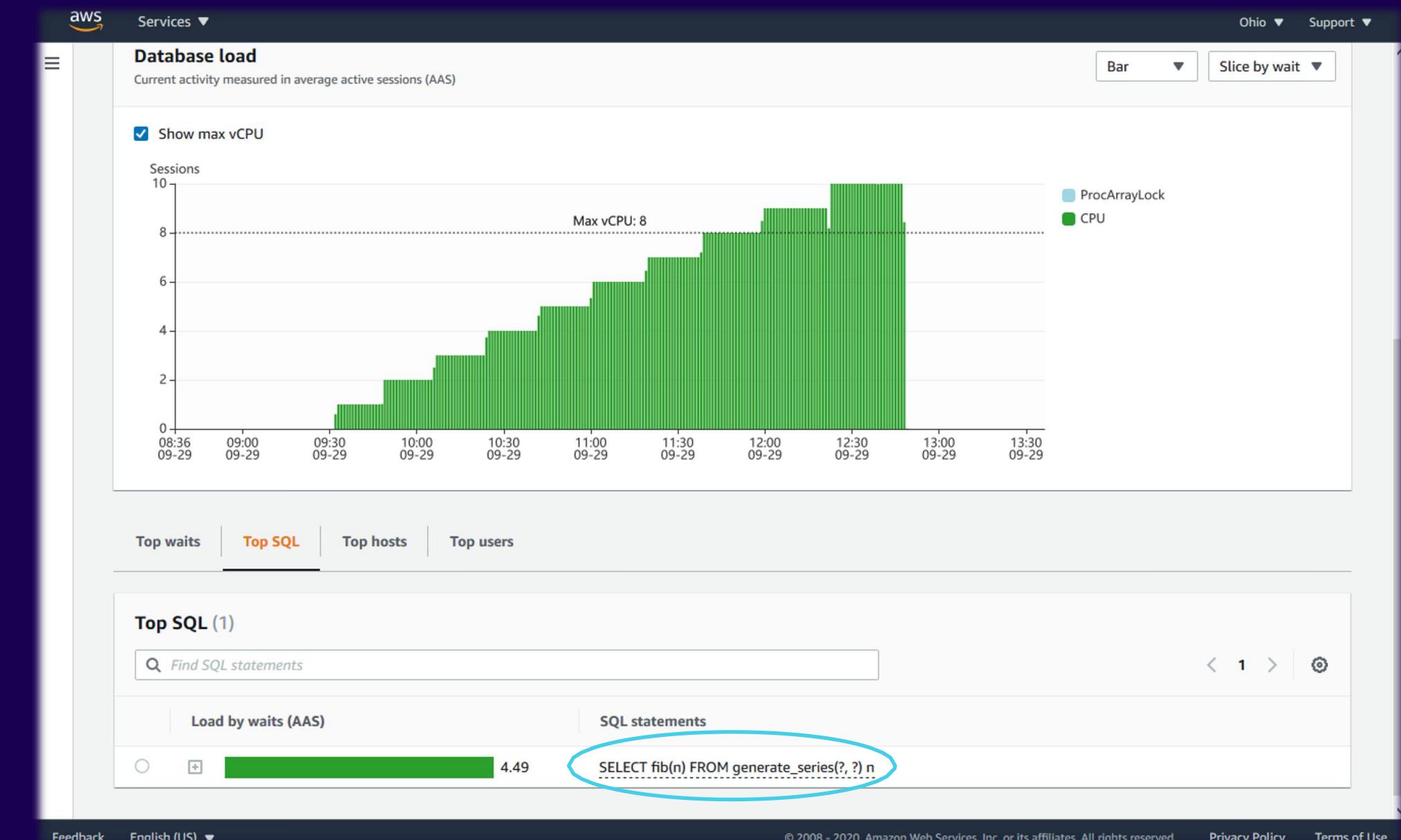
PostgreSQL

Dashboard shows database load over time

Identifies bottlenecks

- Sort by top SQL
- Slice by wait events, host, user

Store up to 2 years of metrics



OS level visibility

PostgreSQL

Pg_proctab exposes the OS /proc information through SQL

Available as an extension

Can tie individual queries to CPU, memory and IO usage

```
postgres=> SELECT substr(sa.query, 0, 14) as query, sa.wait_event,
    pt.state, pt.utime, pt.stime, pt.rss,
    pt.rchar, pt.wchar
  FROM pg_proctab() pt, pg_stat_activity sa
 WHERE sa.pid = pt.pid
   AND sa.pid != pg_backend_pid();
```

query	wait_event	state	utime	stime	rss	rchar	wchar
	AutoVacuumMain	S	939	1510	8020	461888028	231830
	LogicalLauncherMain	S	5	5	6832	562	2
SELECT value	ClientRead	S	897	684	15944	2094218	546611799
SELECT fib(n)		R	129222	45	16192	172295	0
SELECT fib(n)		R	129380	45	16184	172295	0
SELECT fib(n)		R	131358	45	16300	172295	0
SELECT fib(n)		R	131427	33	16276	172295	0
SELECT fib(n)		R	130163	51	16252	172295	0
SELECT fib(n)		R	130397	47	16232	172295	0
SELECT fib(n)		R	131418	47	16252	172295	0
SELECT fib(n)		R	131615	49	16276	172295	0
SELECT fib(n)		R	130999	43	16320	172295	0
SELECT fib(n)		R	131542	46	16396	172295	0
	BgwriterHibernate	S	150	123	4332	69	21
	CheckpointerMain	S	1702	4080	6260	52	77683956205
	WalwriterMain	S	1550	241	4332	2439	20261207

(16 rows)

Profiling stored procedures

Plprofiler is an extension that creates performance profiles of stored procedure code

Provides the execution count and timing of individual code lines

Helps pinpoint code bottlenecks

```
SHOW shared_preload_libraries ;
shared_preload_libraries
-----
rdsutils,pg_stat_statements,plprofiler
(1 row)

CREATE EXTENSION plprofiler;

SELECT pl_profiler_set_enabled_local(true);
SELECT pl_profiler_set_collect_interval(0);

SELECT fib(35);
```

Profiling stored procedures

PostgreSQL

```
SELECT (regexp_split_to_array(
    prosrc,
    '\n'))[line_number]
    as line,
    exec_count,
    total_time
FROM pl_profiler_linestats_local()
    pg_proc
WHERE func_oid = oid
ORDER BY line_number;
```

line	exec_count	total_time
BEGIN	29860703	2189639410
IF n <= 1 THEN	0	0
RETURN n;	29860703	2188447361
END IF;	29860703	16213
RETURN fib(n-1) + fib(n-2);	14930352	1089
END;	0	0
	0	0
	14930351	2184767917
	0	0
	0	0
	0	0

2184 seconds

Upgrades

Minor version upgrades

- Patches to the binaries
- No new functionality
- May contain important security fixes

DB engine version
Version number of the database engine to be used for this database
PostgreSQL 10.12-R1
PostgreSQL 10.12-R1
PostgreSQL 10.13-R1
PostgreSQL 10.14-R1
PostgreSQL 11.7-R1
PostgreSQL 11.8-R1
PostgreSQL 11.9-R1
PostgreSQL 12.2-R1

Major version upgrades

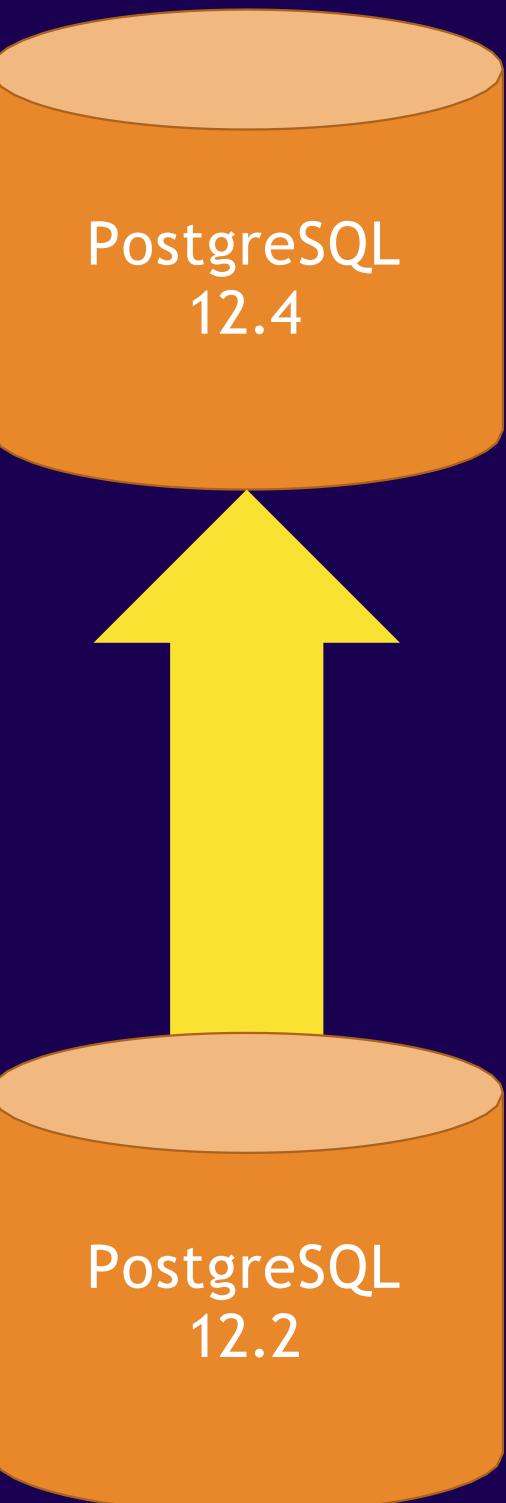
- Tracks the community yearly release cycle
- Introduces new functionality
- May change system catalogs and page formats

Minor version upgrades

The minor version upgrade process typically takes less than 5 to 10 minutes

- Normal shutdown of the instance
 - The time can be extended with long in-flight transactions
- Replace version binaries
- Start the instance

Can be performed manually or automatically



Auto minor version upgrade (AmVU)

Scheduled when the newer minor version is marked as preferred

By default, AmVU is ON

If a database is on a lower version, the instance is upgraded to the preferred version in the next maintenance window

The screenshot shows a database instance named 'database-1' with the following details:

DB identifier	database-1	CPU	Info	Class
Role	Instance	0.92%	Available	db.m4.xlarge
		Current activity	Engine	Region & AZ
		0 Sessions	PostgreSQL	us-east-2a

The 'Maintenance & backups' tab is selected. Under 'Maintenance', the 'Auto minor version upgrade' setting is shown as 'Enabled'. The 'Pending maintenance' section indicates a 'next window'.

A table titled 'Pending maintenance (1)' lists one item:

Description	Type	Status	Apply date
Automatic minor version upgrade to postgres 10.6	db-upgrade	next window	September 6th 2019, 8:19:00 pm UTC-7 (local)

Major version upgrades

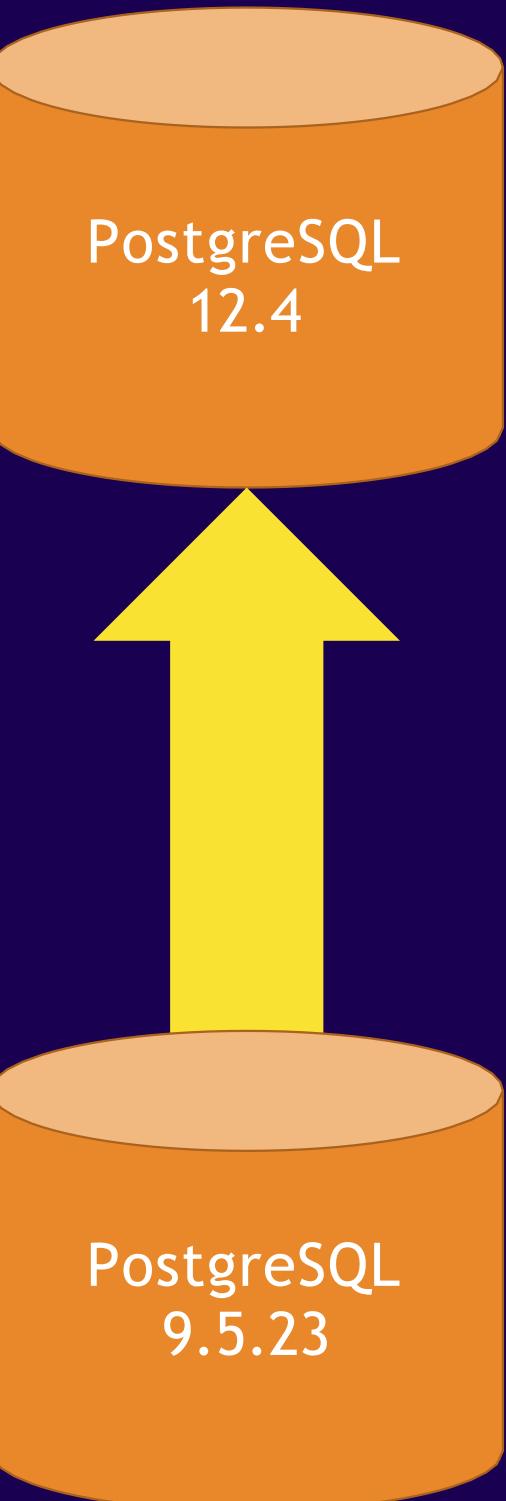
Upgrades can skip major versions with multi major version upgrade

The time needed to upgrade depends more on the number of database objects than the database size

Snapshots taken before and after the upgrade

Statistics are not upgraded so **ANALYZE** is needed on the new version

A new parameter group is needed for the new version



Major version upgrades with read replicas

PostgreSQL

Replicas within the same region
are upgraded with the primary

Replicas are upgraded after the
primary has successfully upgraded

Cross region replicas are
disconnected from the primary

To prevent replicas from being
upgraded, promote them before
the upgrade

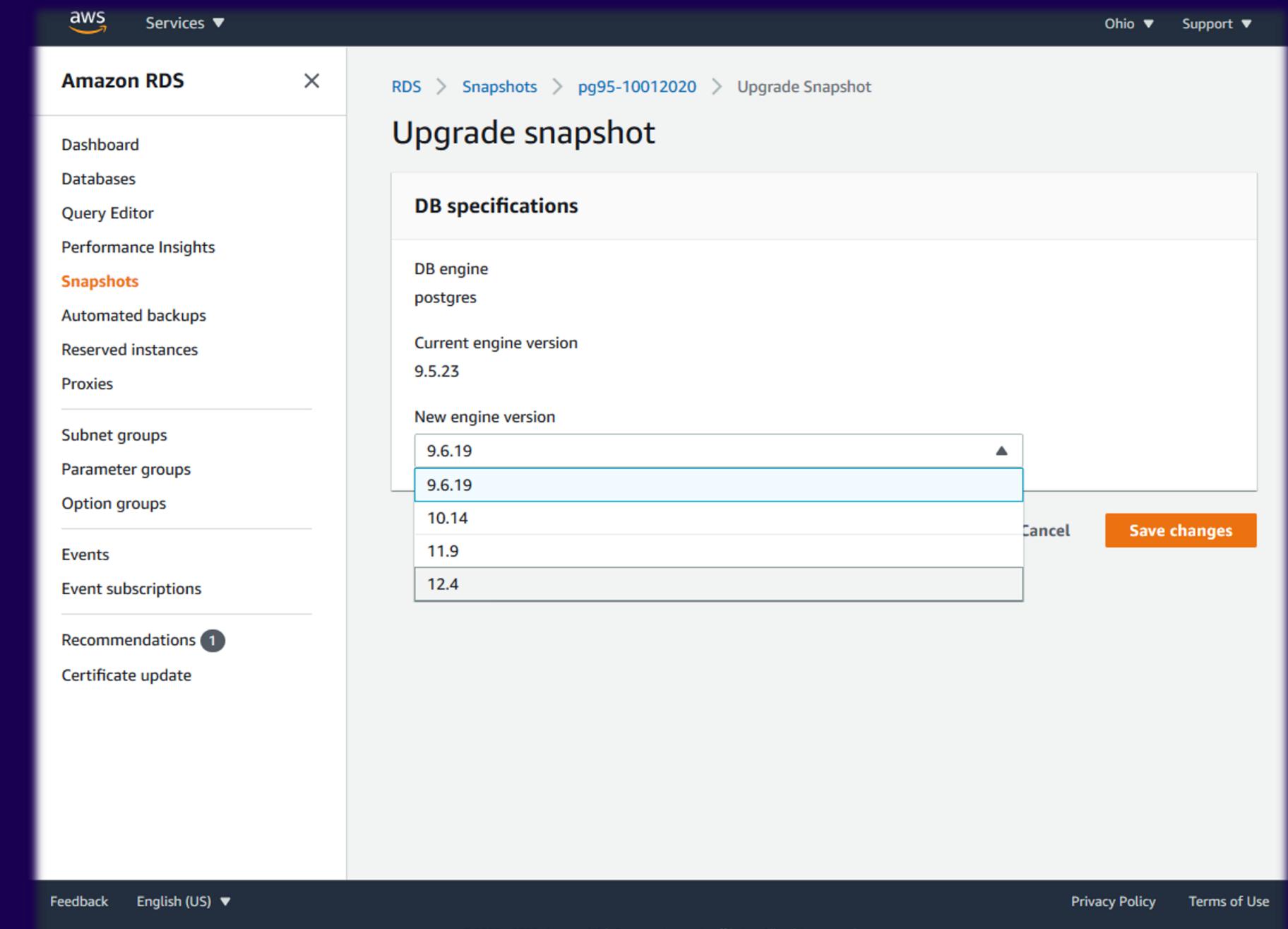
Databases										
<input checked="" type="checkbox"/> Group resources <input type="button" value="C"/> <input type="button" value="Modify"/> <input type="button" value="Actions ▾"/> <input type="button" value="Restore from S3"/> <input type="button" value="Create database"/>										
<input type="text"/> Filter databases < 1 > ①										
DB identifier	Role	Engine	Region & AZ	Size	Status	CPU	Current activity	Maintenance	VPC	
postgres-primary	Primary	PostgreSQL	ap-south-1b	db.m5.xlarge	Upgrading	2.00%	0 Sessions	none	vpc-56f49	<input type="button" value="Promote"/>
postgres-replica1	Replica	PostgreSQL	ap-south-1b	db.m5.xlarge	Upgrading	0.00%	0 Sessions	next window	vpc-56f49	<input type="button" value="Promote"/>

Snapshot upgrades

Upgrades the engine version of user-initiated snapshots to a higher supported version

Useful when the original database version is no longer supported

Copies of snapshots can be upgraded keeping the original

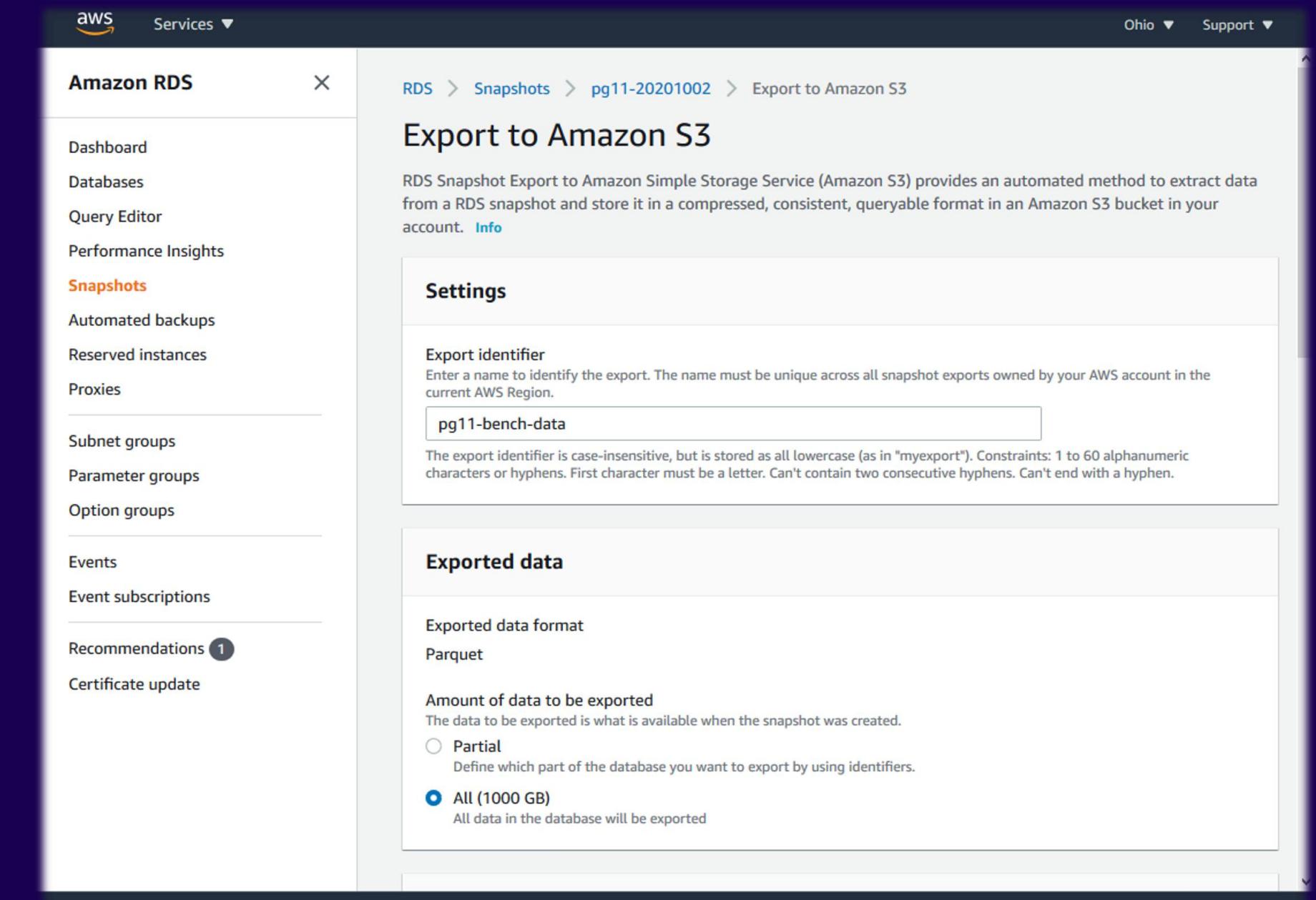


Snapshot exports to Amazon S3

Exports a whole or partial database snapshot to an Amazon S3 bucket

Actions are offline from the database so there is no performance impact

Files created with a Parquet format to be consumed by Amazon Athena, Apache Spark (on Amazon EMR) or Amazon Redshift Spectrum

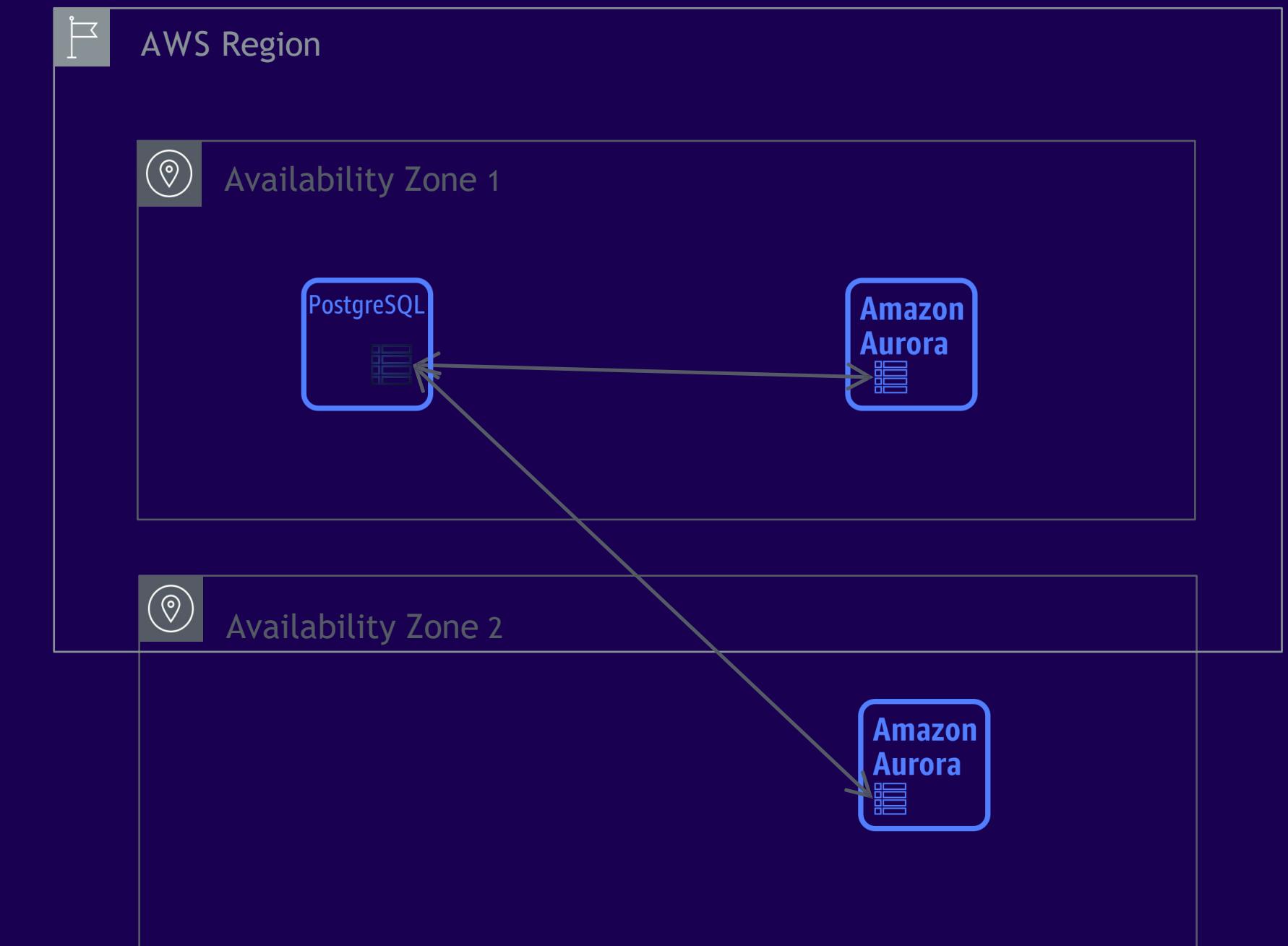


PostgreSQL foreign data wrappers

Creates a database link to another PostgreSQL database

Allows SELECT, INSERT, UPDATE and DELETE

Can push down many operations like joins, aggregates and sorts



PostgreSQL foreign data wrappers

```
CREATE EXTENSION postgres_fdw;
```

```
CREATE SERVER redshift
FOREIGN DATA WRAPPER postgres_fdw
OPTIONS
(dbname 'dw',
 host 'redshift-dw.us-east-2.redshift.amazonaws.com',
 port '5432');
```

```
CREATE USER MAPPING for CURRENT_USER
SERVER redshift
OPTIONS (user 'dwuser', password 'secret');
```

PostgreSQL foreign data wrappers

```
CREATE SCHEMA redshift;  
  
CREATE FOREIGN TABLE redshift.transaction_history (  
    tid      bigint,  
    bid      integer,  
    aid      integer,  
    delta    integer OPTIONS (column_name 'amount'),  
    mtime   timestamp,  
    filler   varchar  
)  
SERVER redshift  
OPTIONS (schema_name 'public',  
        table_name 'transaction_history');
```

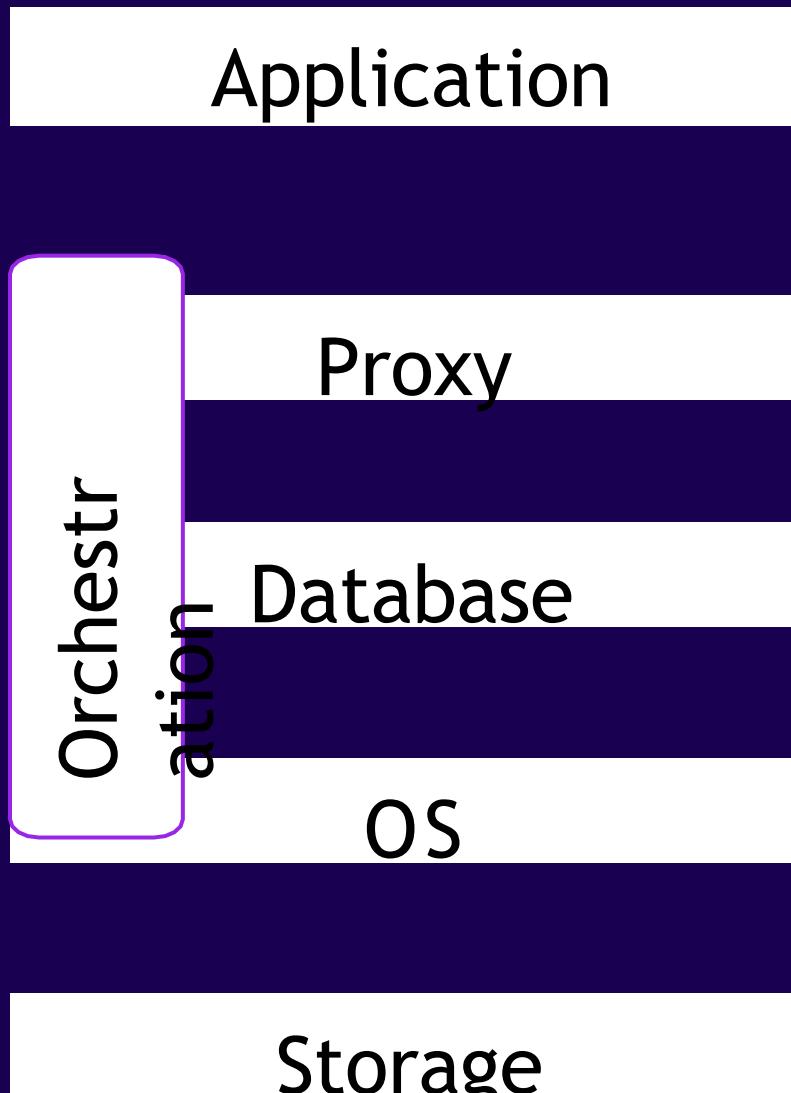
Options

MySQL options on AWS

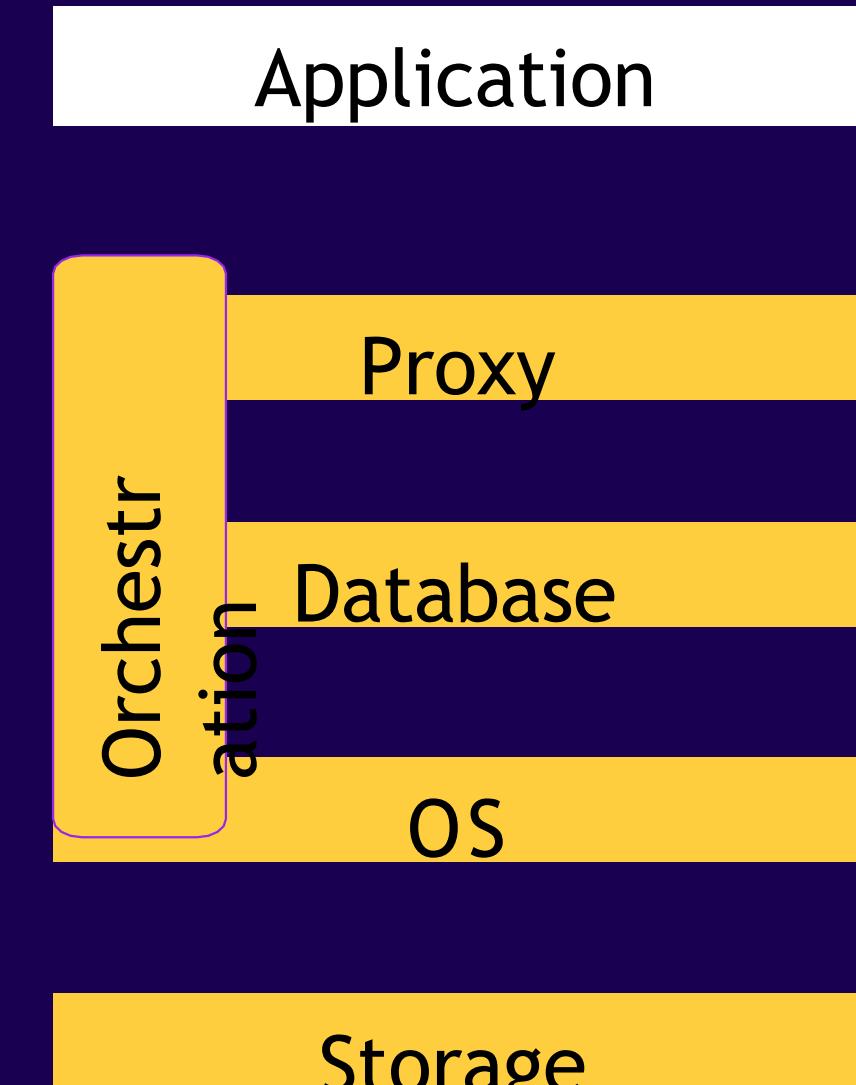
1. Self-managed MySQL on Amazon Elastic Compute Cloud (Amazon EC2)
2. Amazon Relational Database Service (Amazon RDS) for MySQL & Amazon RDS for MariaDB
3. Amazon Aurora MySQL - provisioned and serverless
4. Hybrid cloud: Amazon RDS for MySQL on VMware
5. Hybrid cloud: Amazon RDS for MySQL on AWS Outposts

Deep dive: which option to use when

AWS offers five options for running MySQL



Self-managed



Amazon Aurora



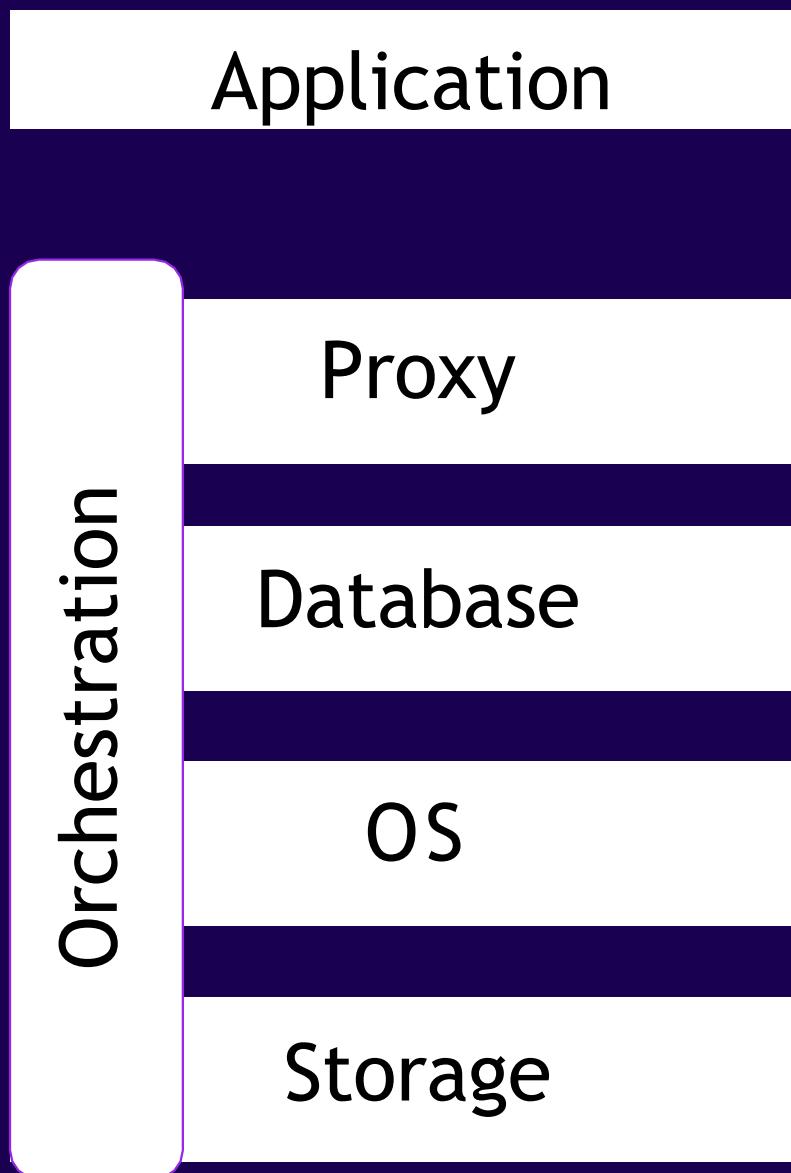
Amazon RDS on Outposts



Amazon RDS MySQL & MariaDB on VMware



Benefits of self-managed MySQL on Amazon^{MySQL} EC2



Full administrative control

Software-controlled networking and security

Choice of ephemeral or Amazon Elastic Block Store (Amazon EBS) storage

Custom hardware and software configurations

What do you want from your database service?

Choice

Manageability

Performance

Availability

Security

Developer
Productivity

What do you want from your database service?

Choice

Manageability

Performance

Availability

Security

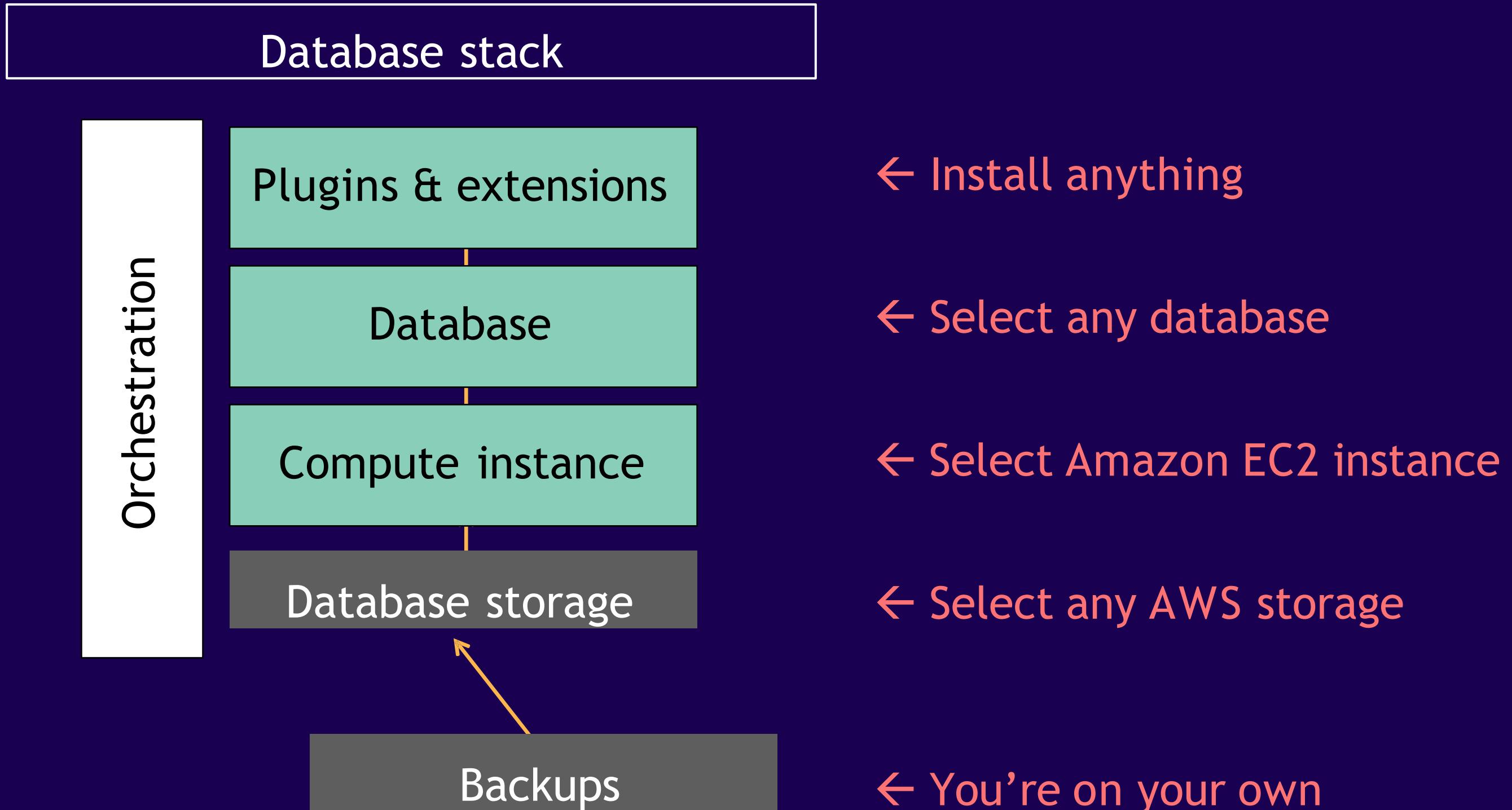
Developer
Productivity

Amazon EC2 (self-managed) customizations

MySQL

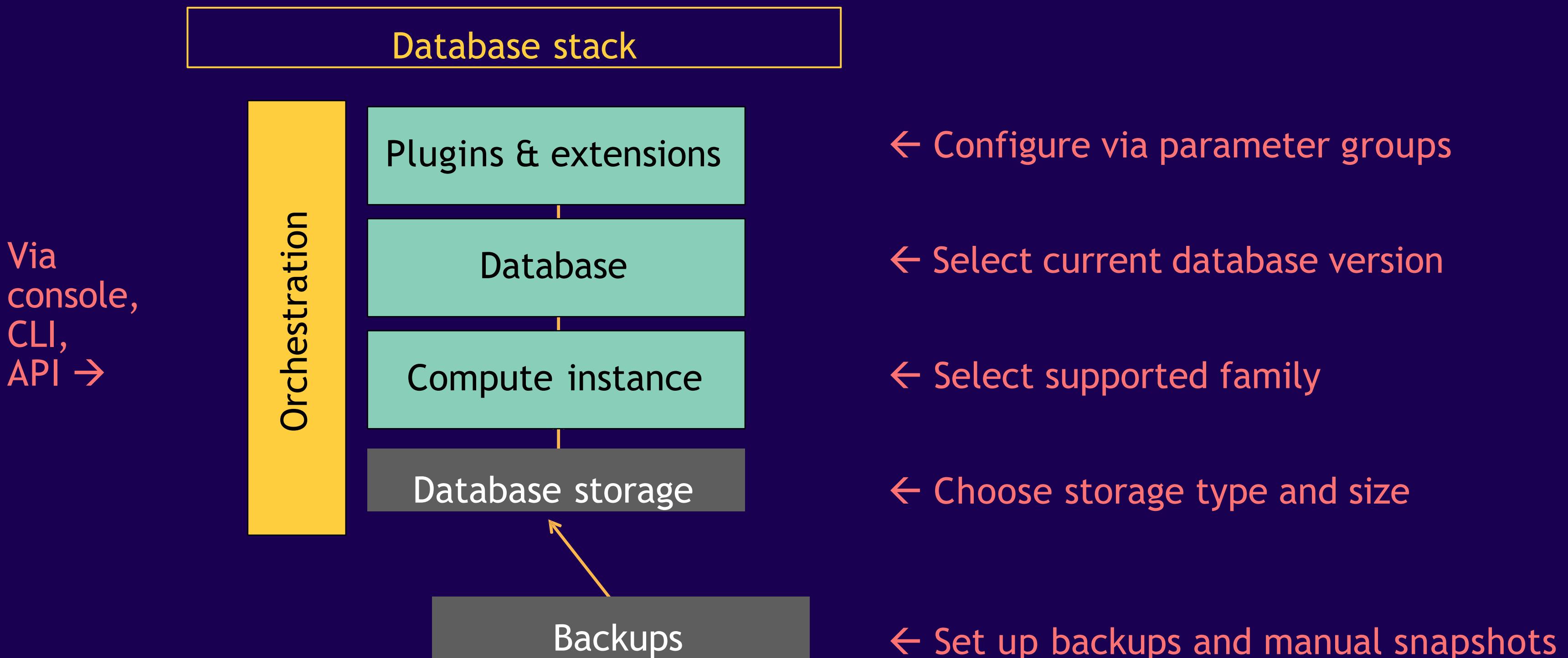
SUITABLE FOR SOME EDGE CASES

You're on
your own →



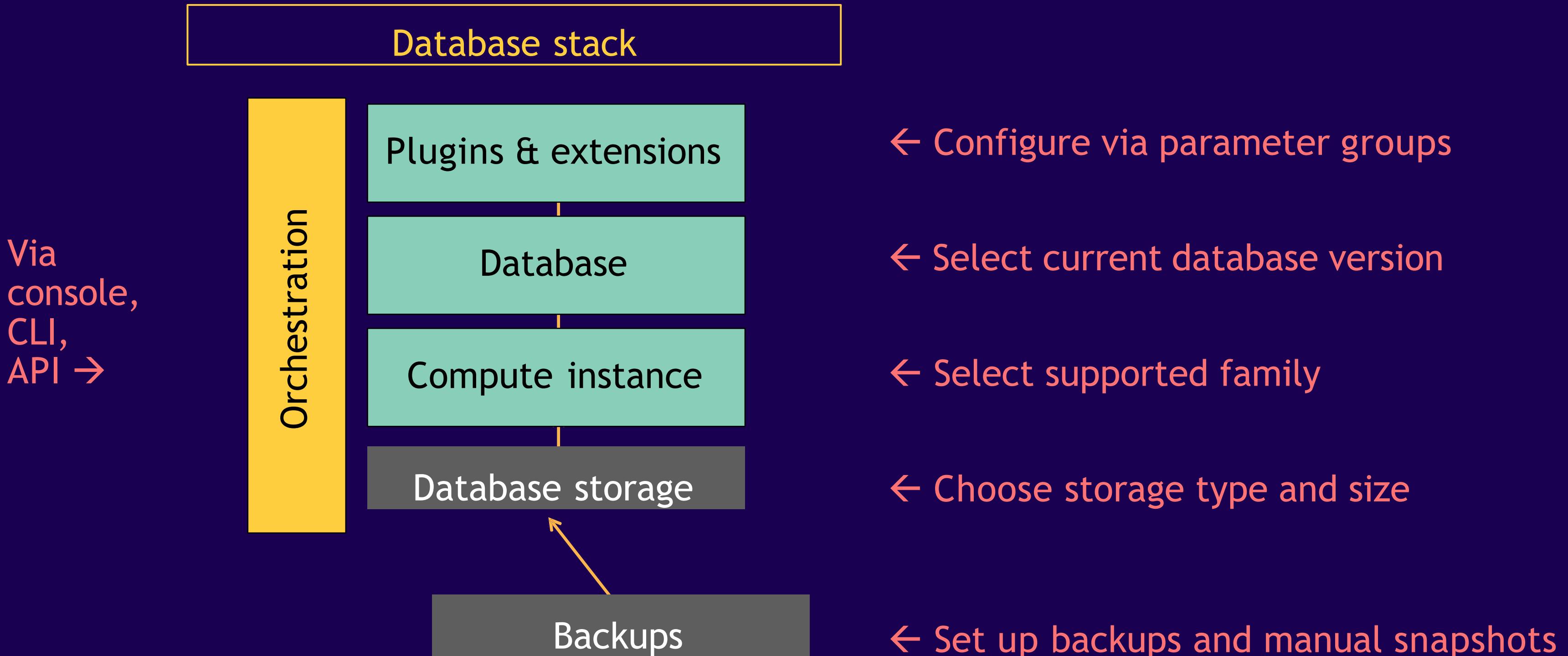
Amazon RDS (AWS-managed) customizations

SUITABLE FOR ALMOST ALL MySQL CUSTOMERS



Amazon RDS (AWS-managed) customizations

SUITABLE FOR ALMOST ALL MySQL CUSTOMERS



Choice of database engines and instance types

INCLUDING AMAZON RDS GRAVITON2 INSTANCES

MySQL versions: 5.7, 8.0, 8.4

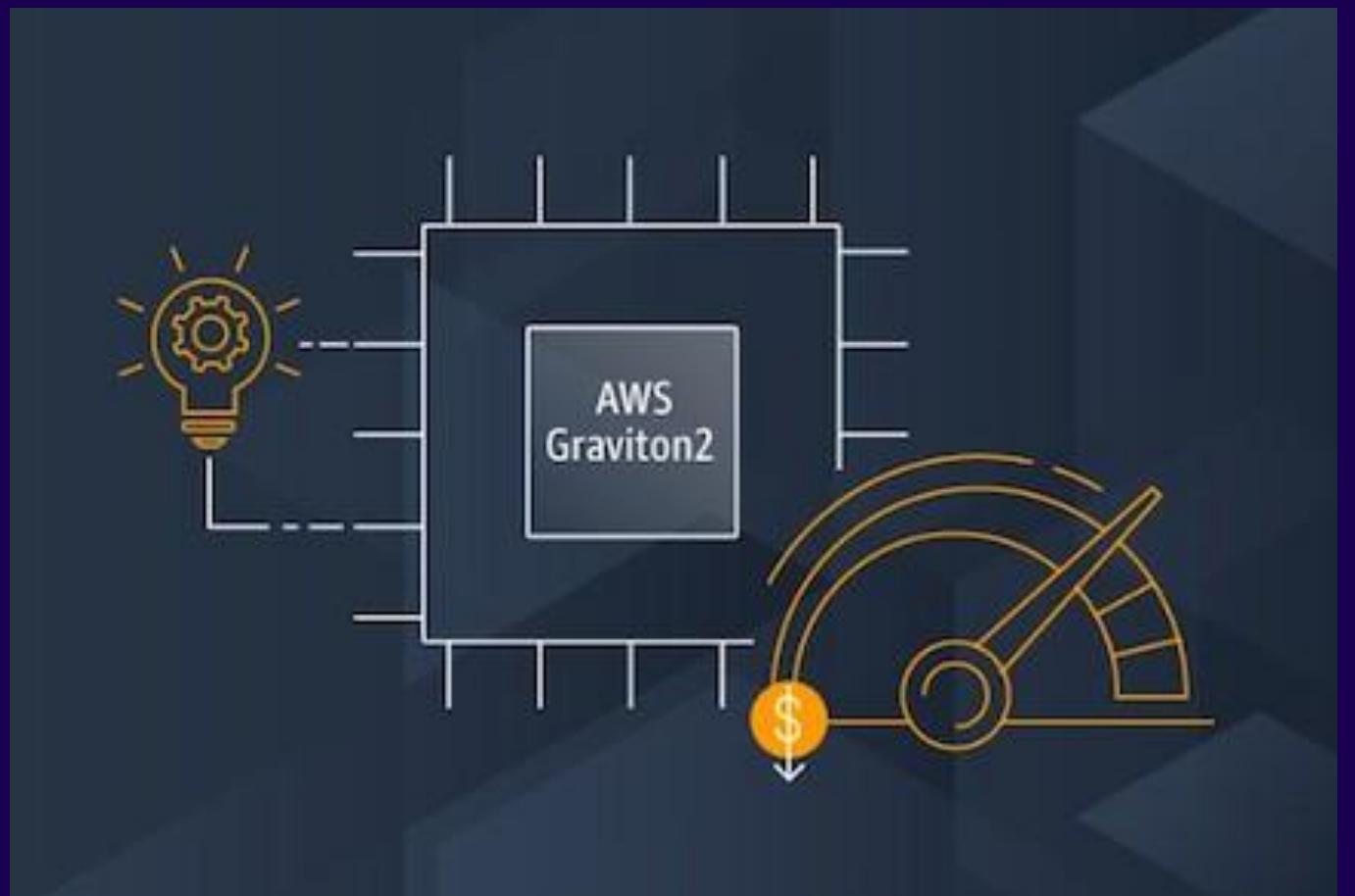
MariaDB versions: 10.4, 10.5, 10.6, 10.11, 11.4

Aurora MySQL versions: 5.7, 8.0

Instance families including R5, M5, T3, R6G, M6G

R6G and M6G instances powered by AWS Graviton2 -

- Available for Amazon RDS for MySQL and Amazon RDS for MariaDB
- 13% better performance and 27% better price/performance for Amazon RDS for MySQL



Amazon RDS for MySQL hybrid options

Amazon RDS on VMware

- Familiar Amazon RDS capabilities including Amazon RDS Console, API, and CLI as in the AWS regions
- Support for read replicas and RDS Proxy
- For running on VMware vSphere on-premises



Amazon RDS on
VMware

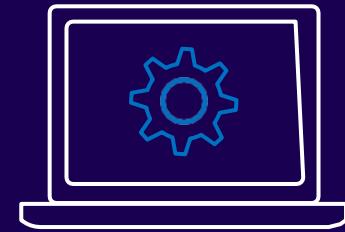
Amazon RDS on AWS Outposts

- Familiar RDS capabilities including RDS Console, API, and CLI as in the AWS regions
- For running on AWS Outposts on-premises



Amazon RDS on
Outposts

Amazon RDS for MySQL hybrid use cases



Latency

- Applications and processes sensitive to network and disk latency
- Financial services applications such as trading and brokerage
- Security and fraud applications that require a quick response time

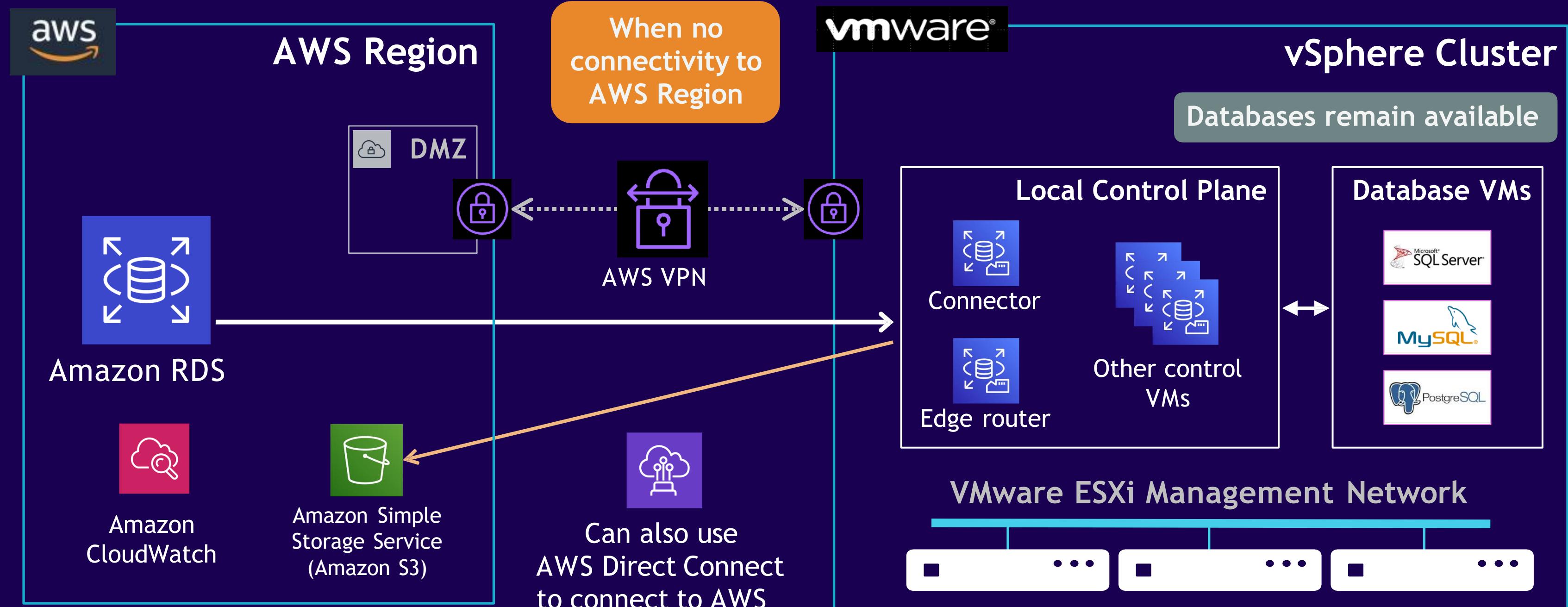


Residency

- **Regulations dictate that data and infrastructure reside inside the country**
- **Contracts specify where applications are deployed**
- **You're note-ready to move to AWS regions for information security or other reasons**

How Amazon RDS on VMware works

MySQL



Amazon RDS connector establishes secure VPN connection with AWS region

Amazon RDS is deployed in your datacenter through VPN tunnel

Amazon RDS provisions and manages on-premises instances, including automated backups and restores

What do you want from your database service?

Choice

Manageability

Performance

Availability

Security

Developer
Productivity

Amazon RDS manageability highlights

- Automated OS and database upgrades
- Push-button scaling
- Managed binlog replication
- Log upload to Amazon CloudWatch Logs
- Per-second billing
- Auto-scaling storage up to 64TiB and 80K PIOPS (Aurora: 128TiB)
- Snapshot export to Amazon S3 NEW!



Amazon RDS

NEW!

Amazon RDS for MySQL backups, snapshots, and point-in-time restore

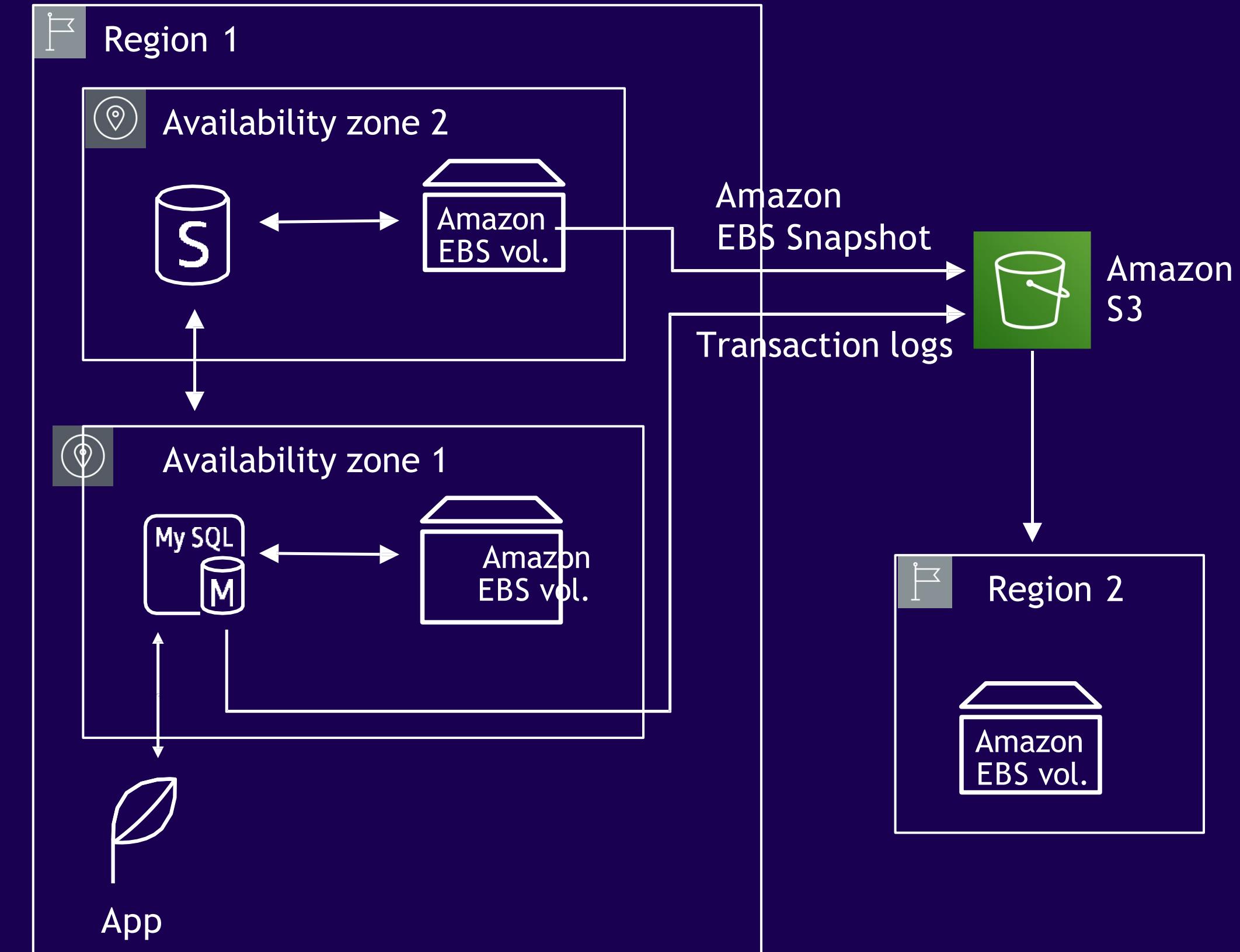
Two options—automated backups and manual snapshots

Amazon EBS snapshots stored in Amazon S3

Transaction logs stored every 5 minutes in Amazon S3 to support point-in-time recovery

No performance penalty for backups

Snapshots can be copied across regions or shared with other accounts



Aurora scale-out, distributed architecture

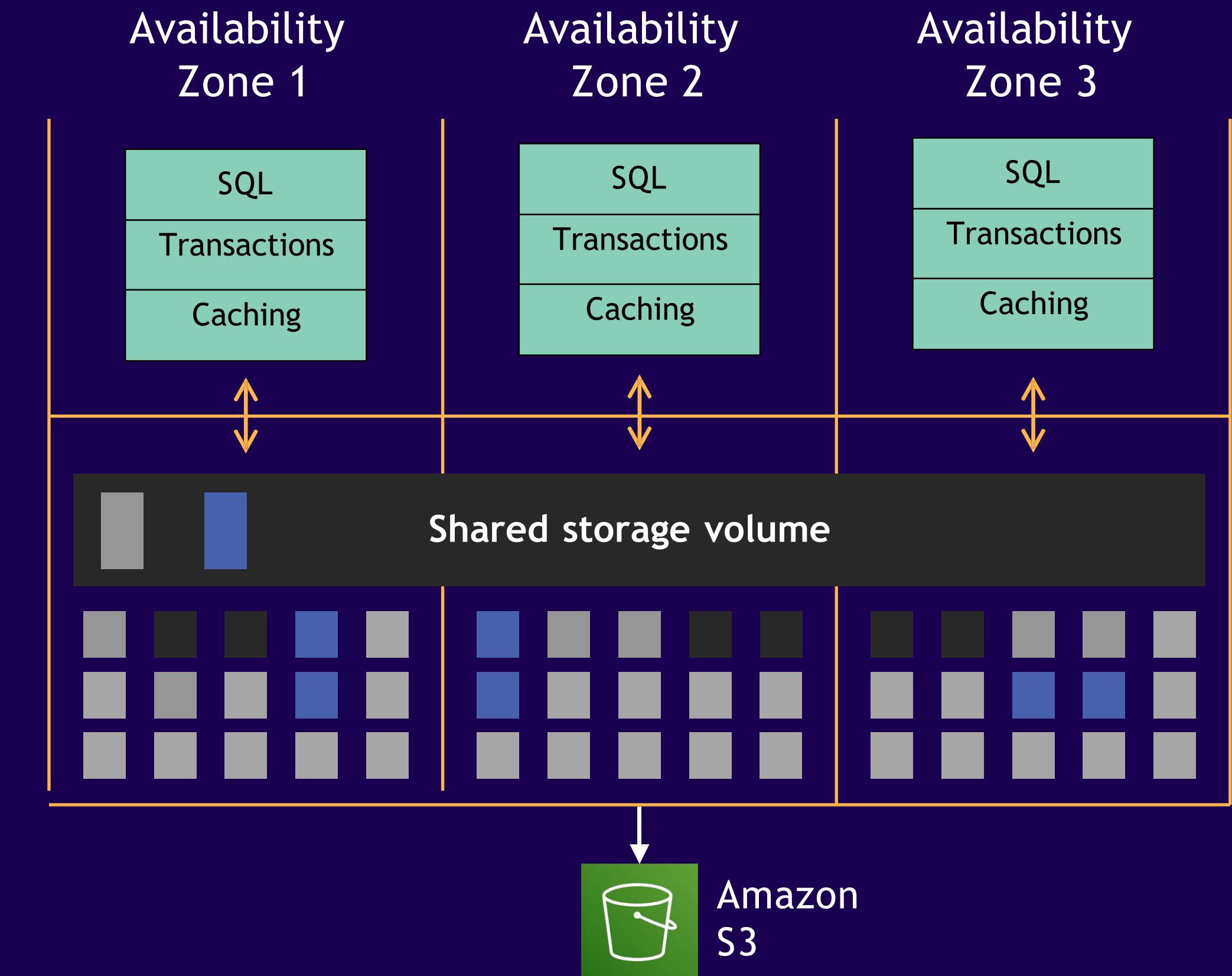
MySQL

Purpose-built, log-structured, distributed storage system designed for databases

Storage volume is striped across hundreds of storage nodes distributed over 3 different availability zones

Six copies of data, two copies in each availability zone to protect against AZ+1 failures

Continuous backups to Amazon S3 and faster restores



Amazon Aurora Serverless

Starts up on demand, shuts down when not in use

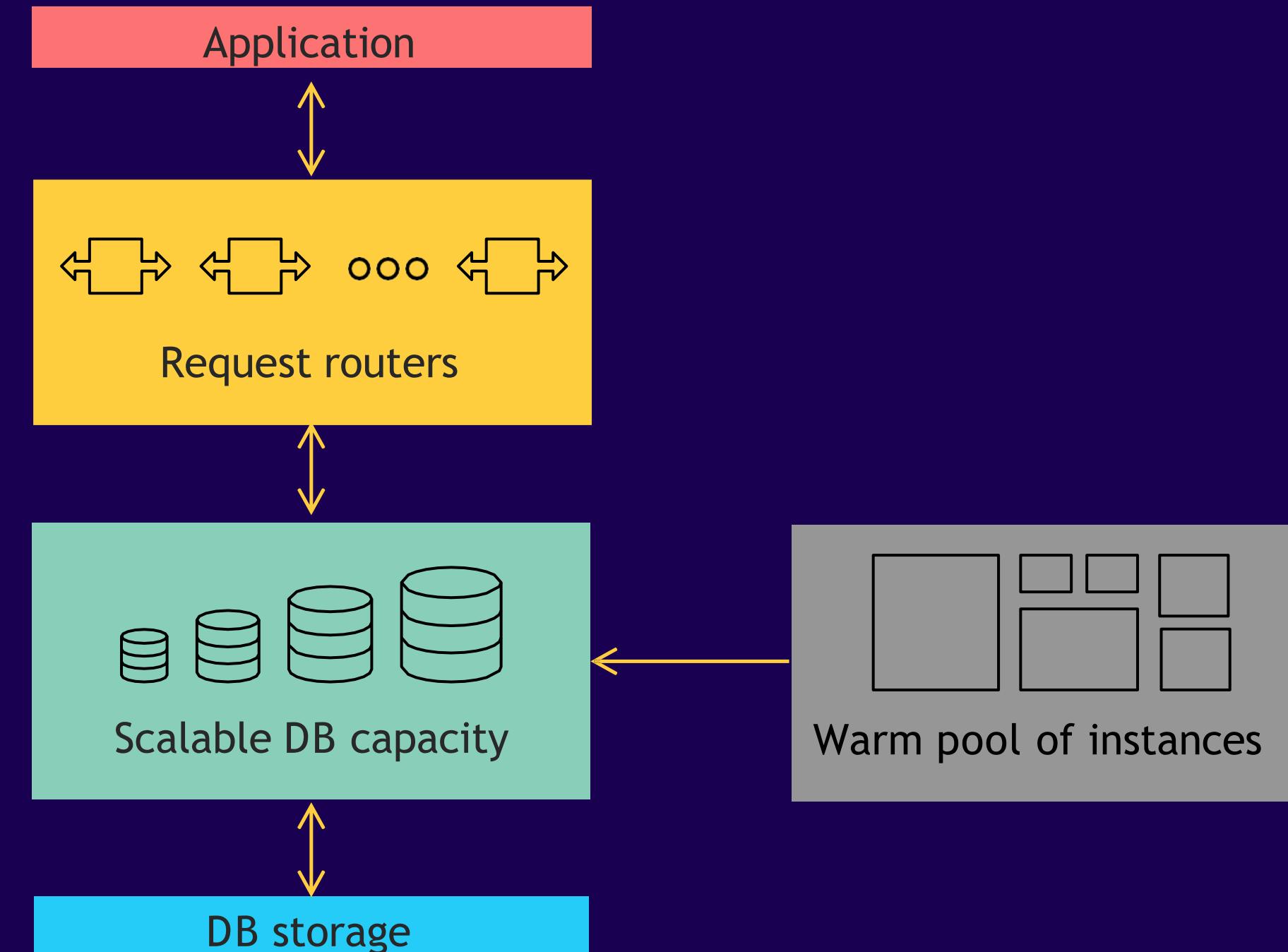
Adjusts capacity automatically

No application impact when scaling

Pay per second, one-minute minimum

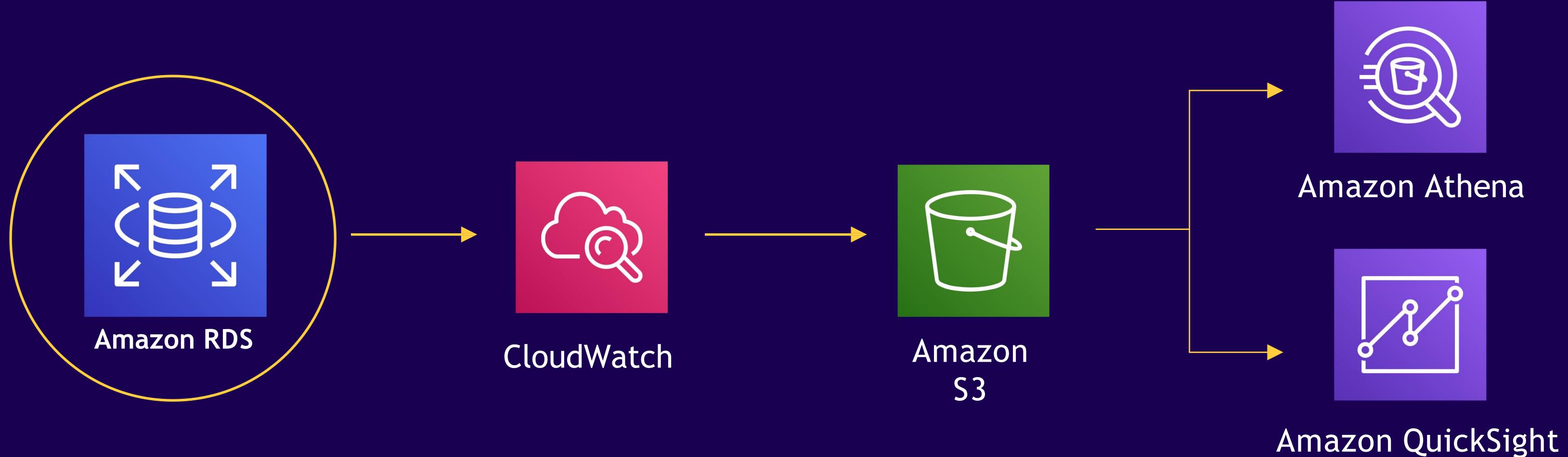
Great for infrequently-used, unpredictable, or cyclical workloads

Data API for running queries over HTTP



Log exports to CloudWatch Logs

MySQL



- Continuously monitor activity in your DB clusters by sending audit, general, slow query, and error logs to CloudWatch Logs
- Export to Amazon S3 for long term archival; analyze logs using Athena; visualize logs with QuickSight

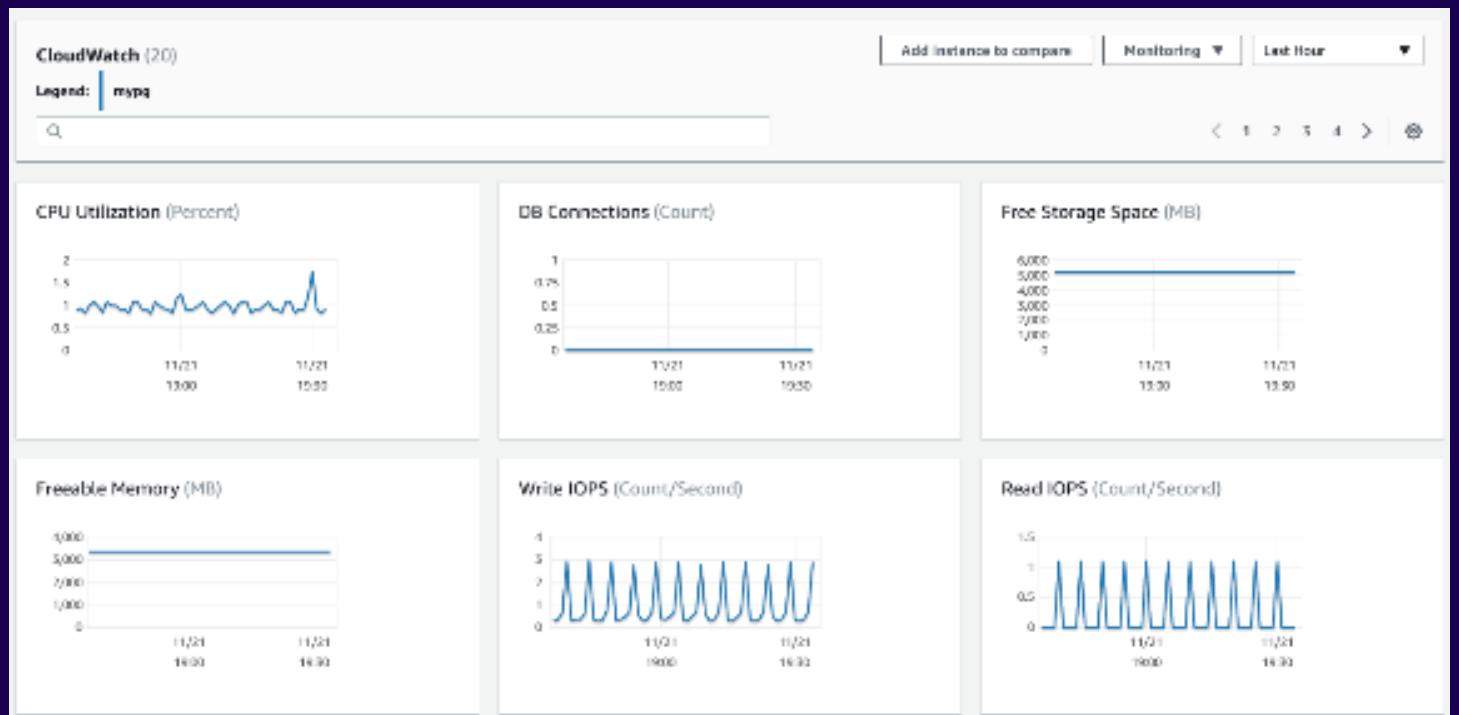
Search: Look for specific events across log files

Metrics: Measure activity in the Aurora DB cluster

Visualizations: Create activity dashboards

Alarms: Get notified or take actions

Monitor Amazon RDS with CloudWatch



CloudWatch metrics

- CPU/Storage/Memory
- Swap usage
- I/O (read and write)
- Latency (read and write)
- Throughput (read and write)
- Replica lag
- CloudWatch alarms
- Similar to on-premises monitoring tools

Enhanced monitoring

- Access to additional CPU, memory, file system, and disk I/O metrics
- As low as one-second intervals

Integration with third-party monitoring tools

Monitor Amazon RDS with CloudWatch

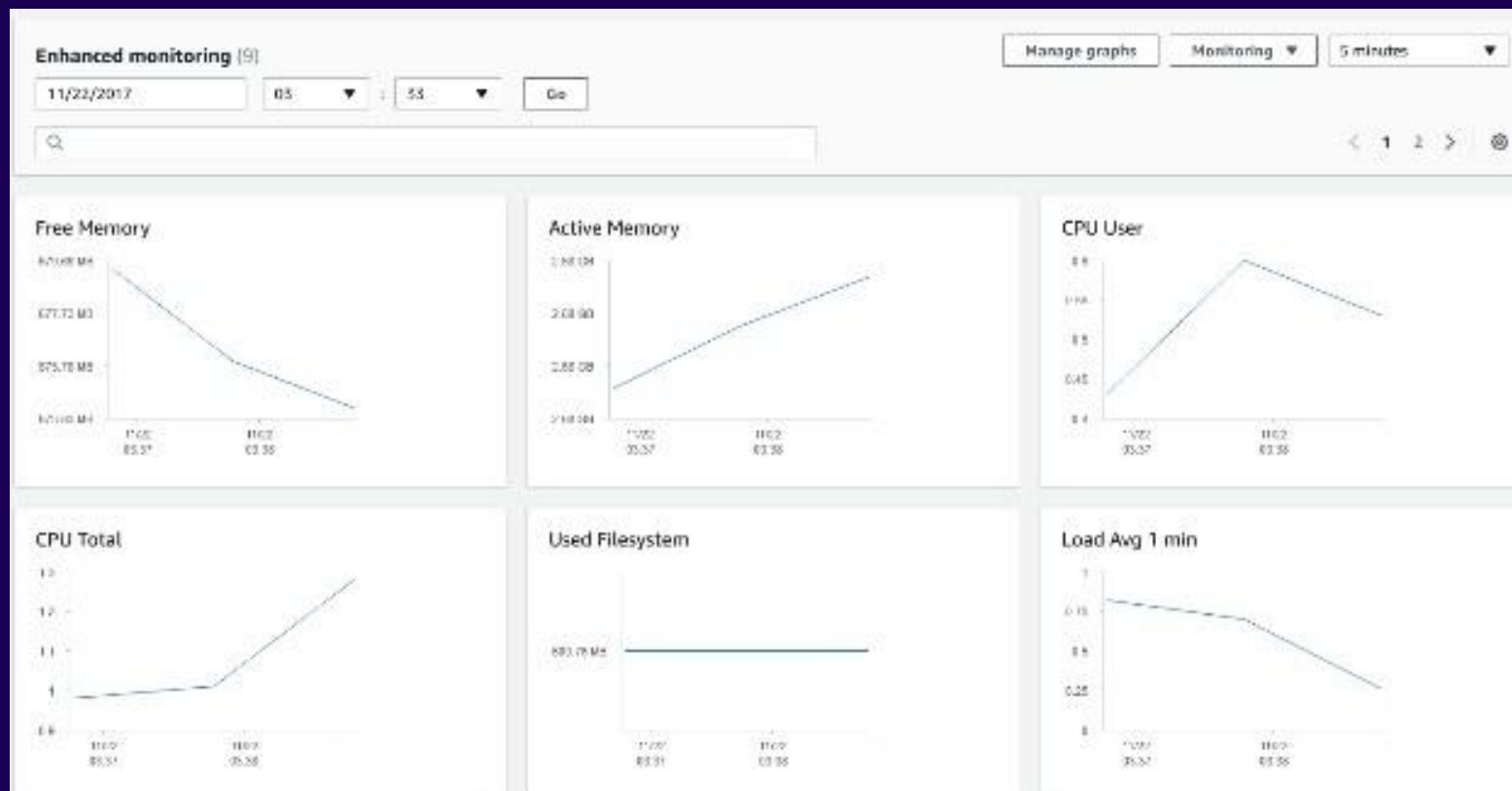
CloudWatch metrics

- CPU/Storage/Memory
- Swap usage
- I/O (read and write)
- Latency (read and write)
- Throughput (read and write)
- Replica lag
- CloudWatch alarms
- Similar to on-premises monitoring tools

Enhanced monitoring

- Access to additional CPU, memory, file system, and disk I/O metrics
- As low as one-second intervals

Integration with third-party monitoring tools



Monitor Amazon RDS with CloudWatch

CloudWatch metrics

- CPU/Storage/Memory
- Swap usage
- I/O (read and write)
- Latency (read and write)
- Throughput (read and write)
- Replica lag
- CloudWatch alarms
- Similar to on-premises monitoring tools

Operating system process list					
Process list					
NAME	VIRT	RES	CPU%	MEM%	
postgres [3213]	1.04 GB	52.74 MB	0	1.33	
postgres: relsadmin rdsadmin localhost:28320 idle [1771]	1.14 GB	8.04 MB	0	0.2	
postgres: logger process [3214]	67.42 MB	1.89 MB	0	0.04	
postgres: checkpointer process [3216]	1.04 GB	26.22 MB	0	0.65	
postgres: writer process [3217]	1.04 GB	9.51 MB	0	0.24	

Enhanced monitoring

- Access to additional CPU, memory, file system, and disk I/O metrics
- As low as one-second intervals

Integration with third-party monitoring tools

Performance Insights

All

Dashboard showing database load

- Easy - e.g. drag and drop
- Powerful - drill down using zoom in

Identifies source of bottlenecks

- Sort by top SQL
- Slice by host, user, wait events

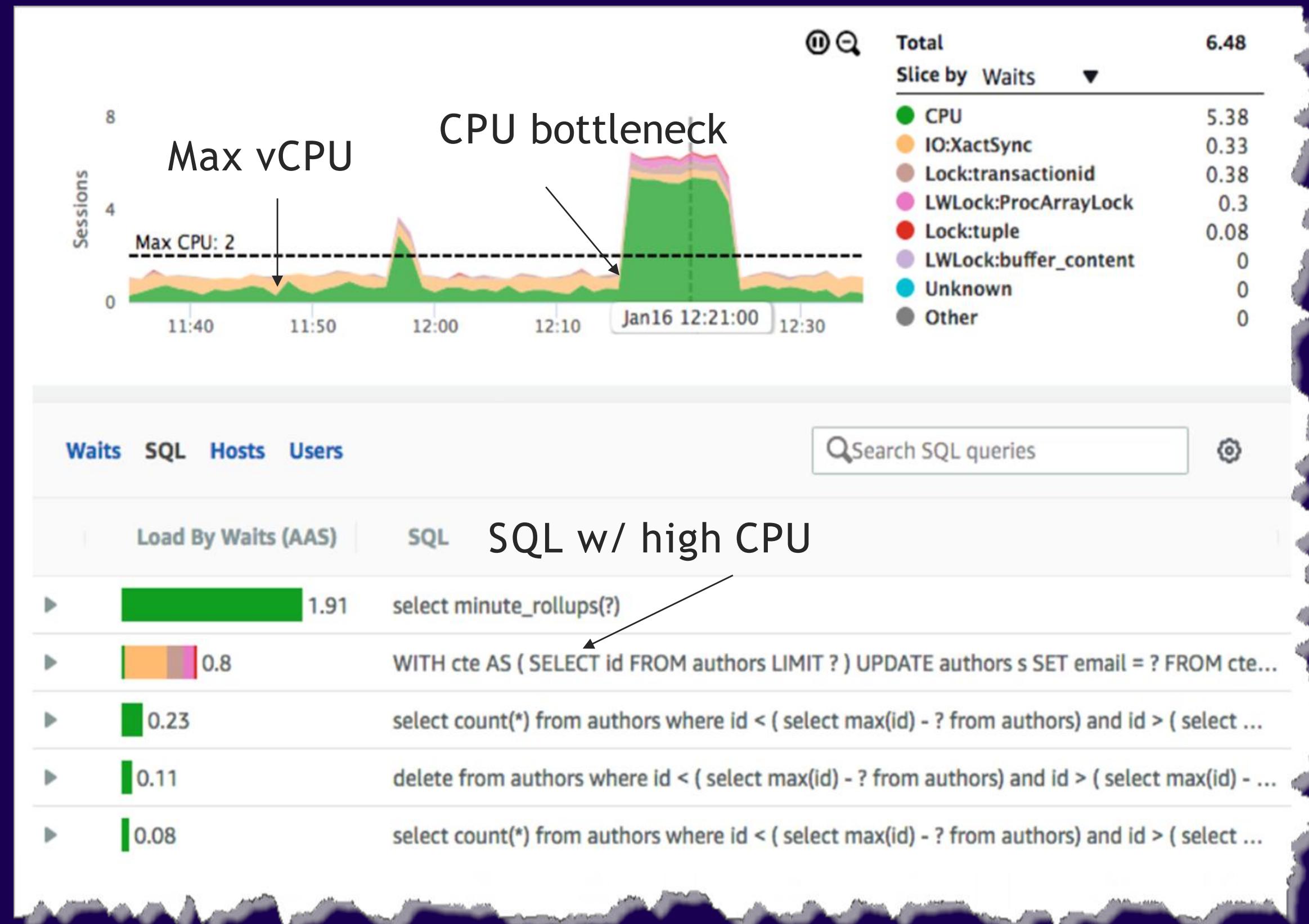
Adjustable time frame

- Hour, day, week, month
- Up to 2 years of data; 7 days free

Support for MySQL and MariaDB

- SQL-level metrics

NEW!



What do you want from your database service?

Choice

Manageability

Performance

Availability

Security

Developer
Productivity

Read scaling with RDS read replicas

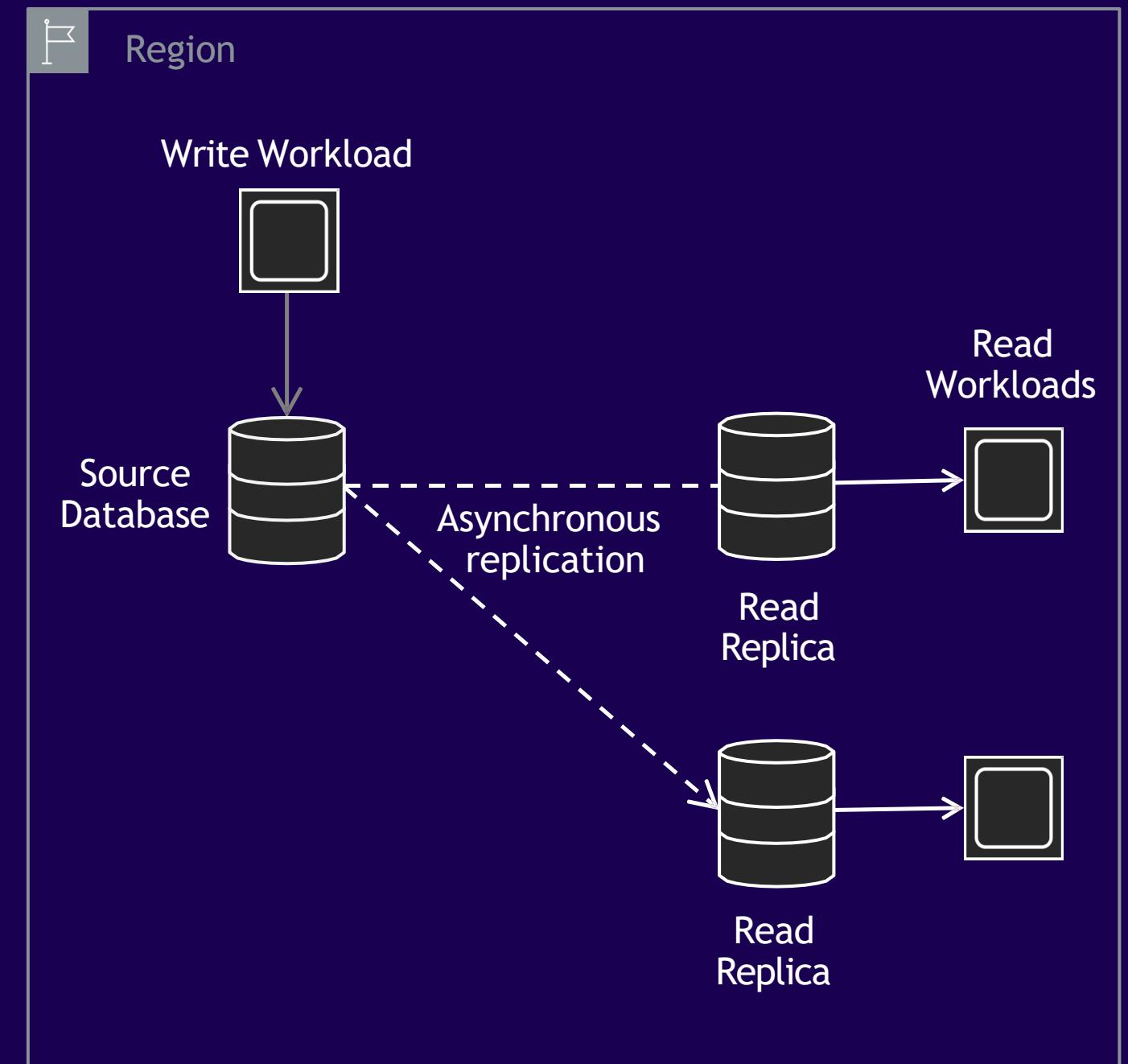
Read replicas offload reads from the source database

Create up to five replicas per source database

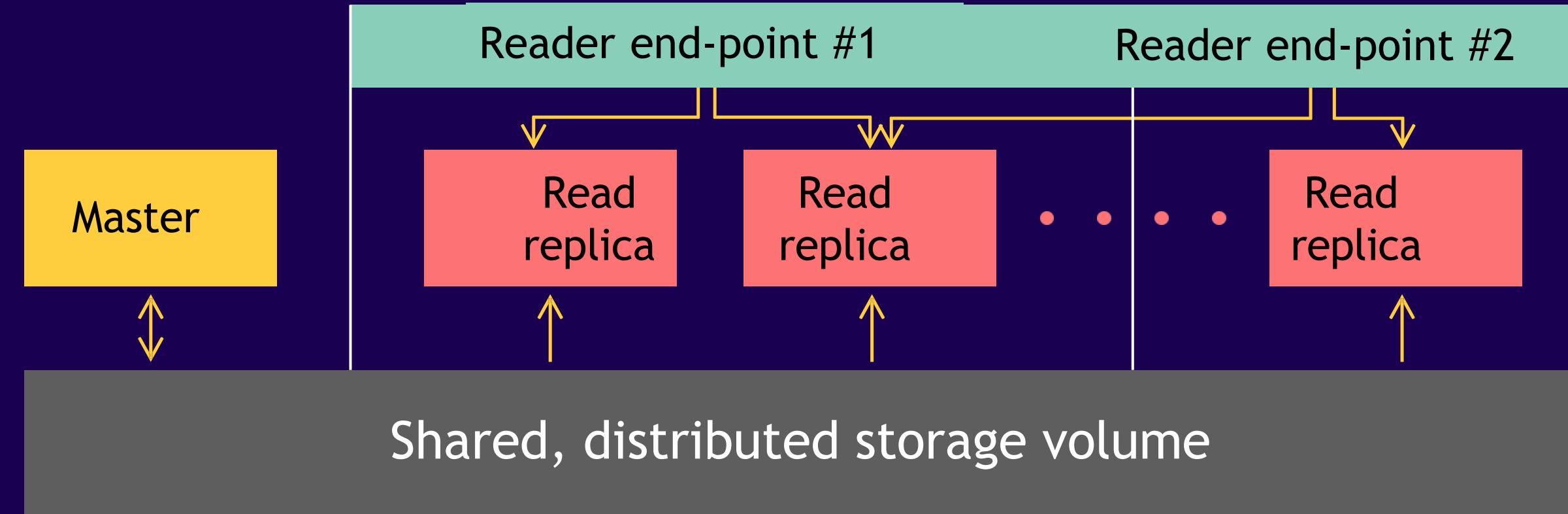
Binlog replication (file position or GTID)

Upgrades independent of the source database

Each read replica itself can be multi-AZ



Aurora read replicas and database auto-scaling



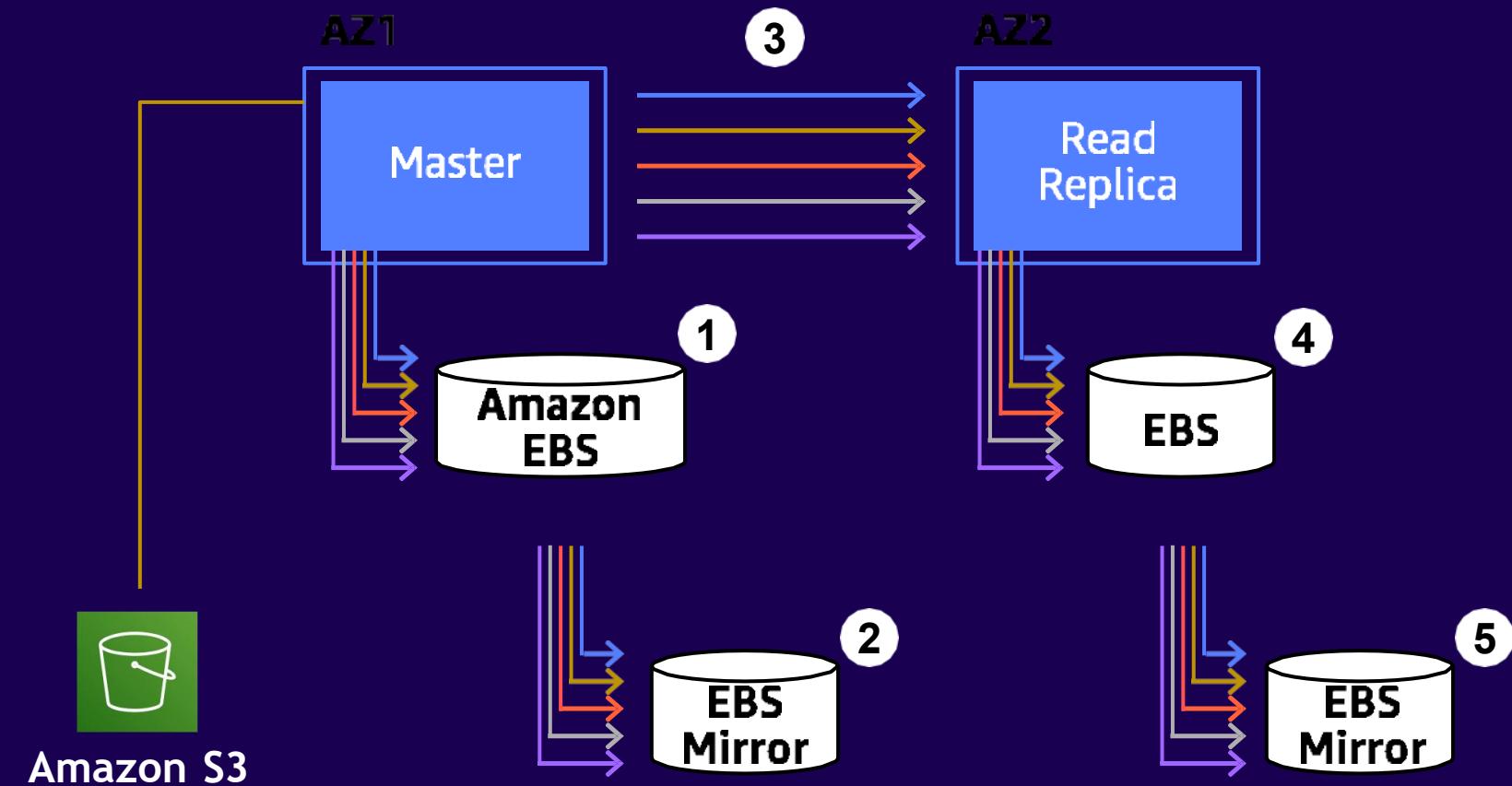
Up to 15 promotable read replicas across multiple availability zones

Redo log-based (physical) replication leads to low-replica lag—typically < 10ms

Custom reader end-point with load balancing and auto-scaling

MySQL vs. Aurora I/O profile

MySQL with replica

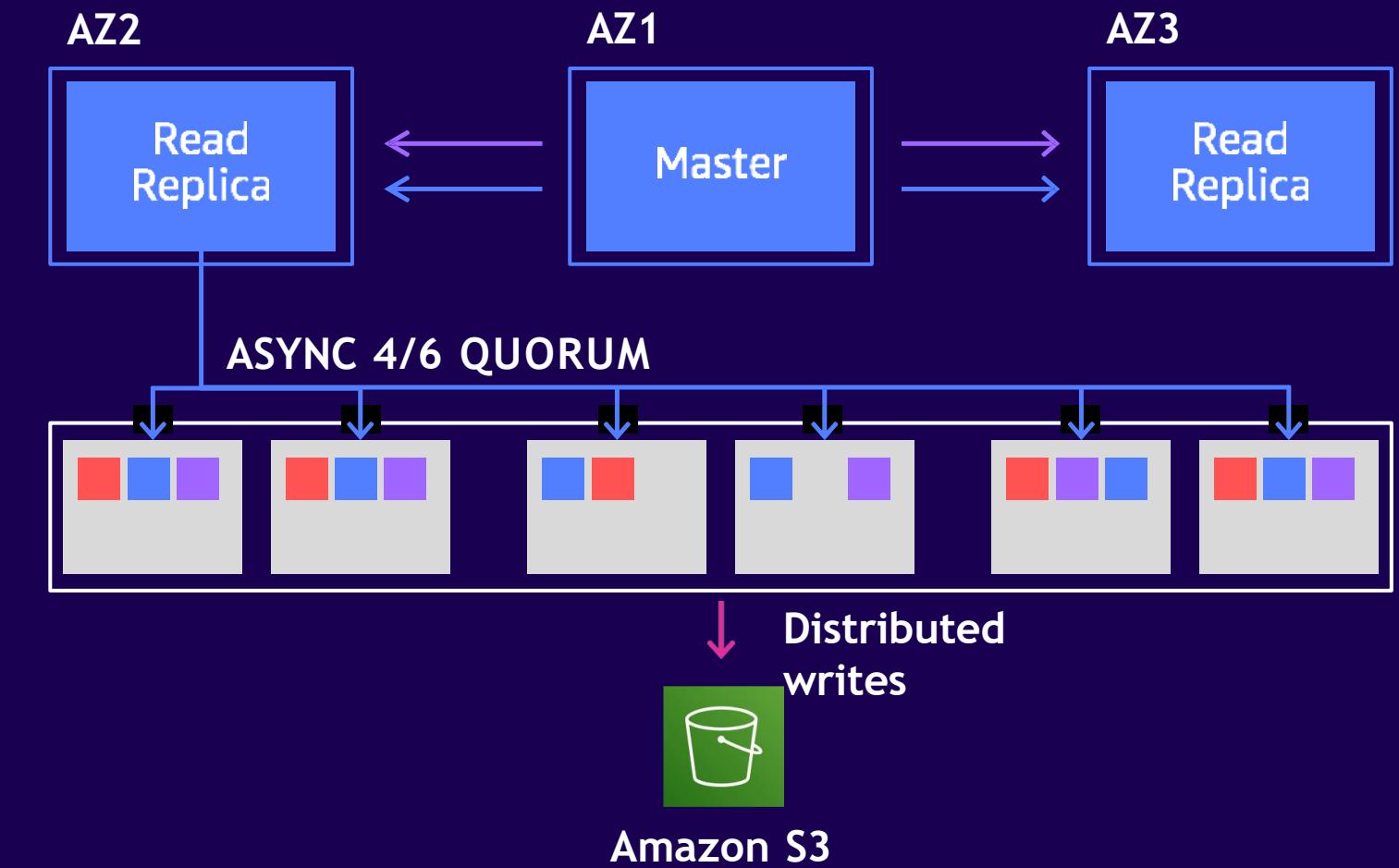


MySQL I/O profile for 30 min Sysbench run

0.78MM transactions

7.4 I/Os per transaction

Amazon Aurora



Aurora I/O profile for 30 min Sysbench run

27MM transactions

35X More

0.95 I/Os per transaction

7.7X Less

Type of write

→ Log

→ Binlog

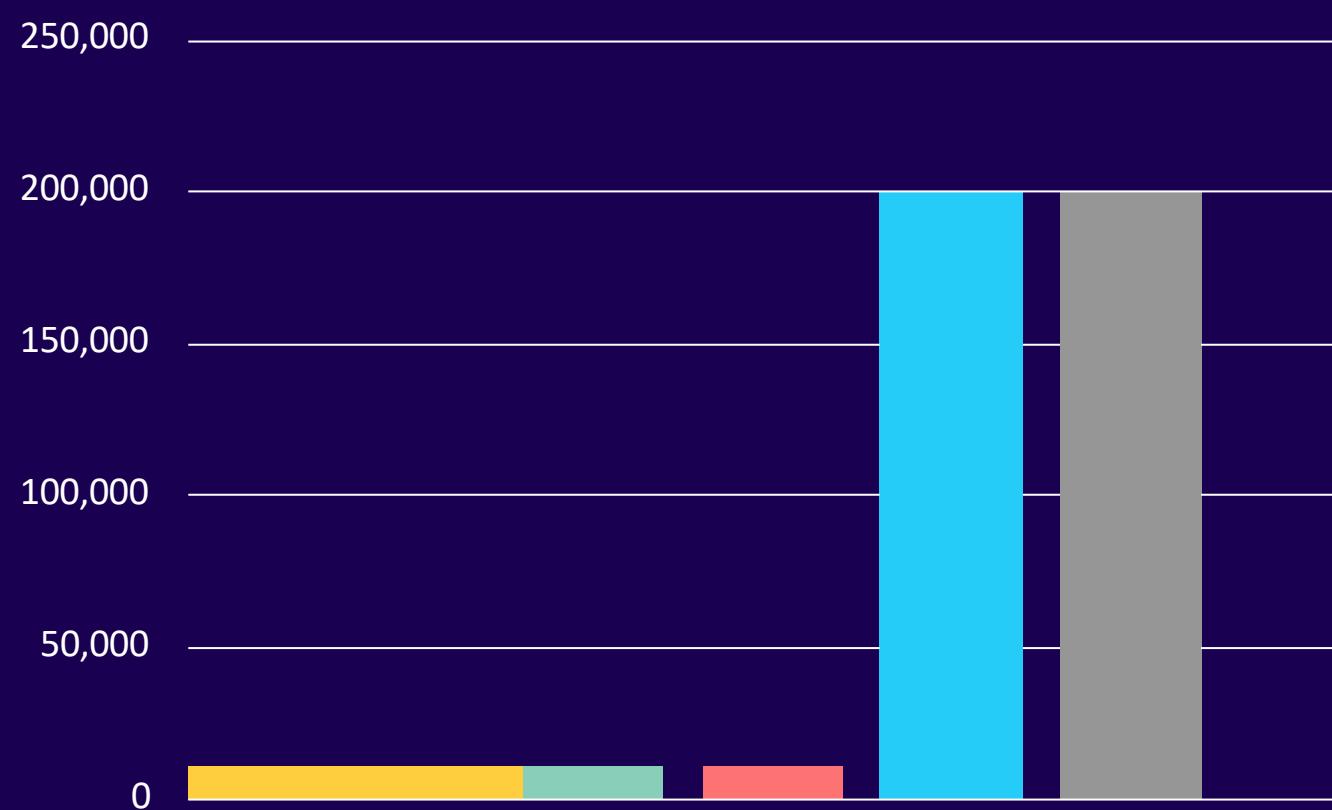
→ Data

→ Double write

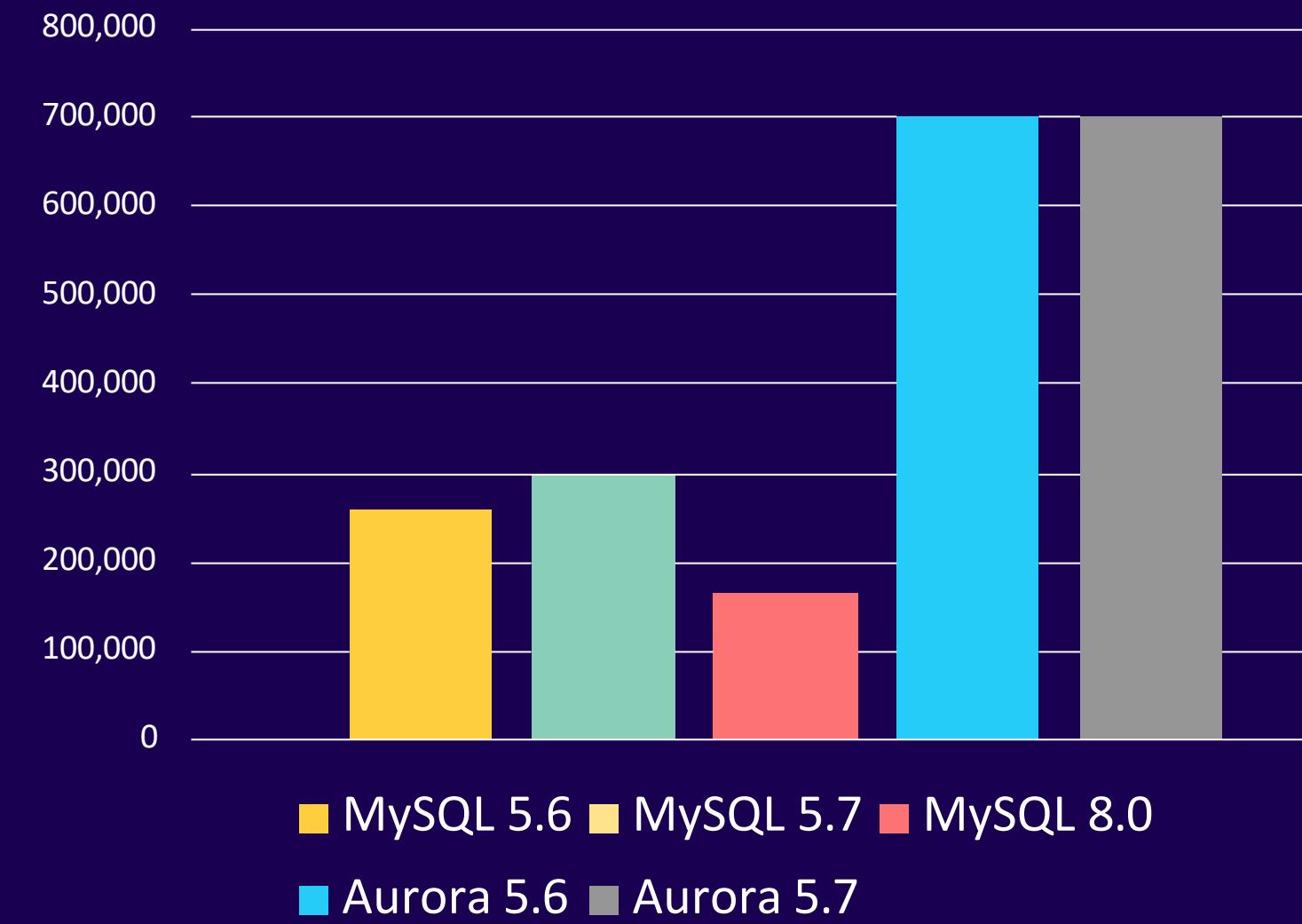
→ FRM files

Aurora write and read throughput

AURORA MYSQL IS 5X FASTER THAN MYSQL



Write Throughput



Read Throughput

Using Sysbench with 250 tables and 200,000 rows per table on R4.16XL

What do you want from your database service?

Choice

Manageability

Performance

Availability

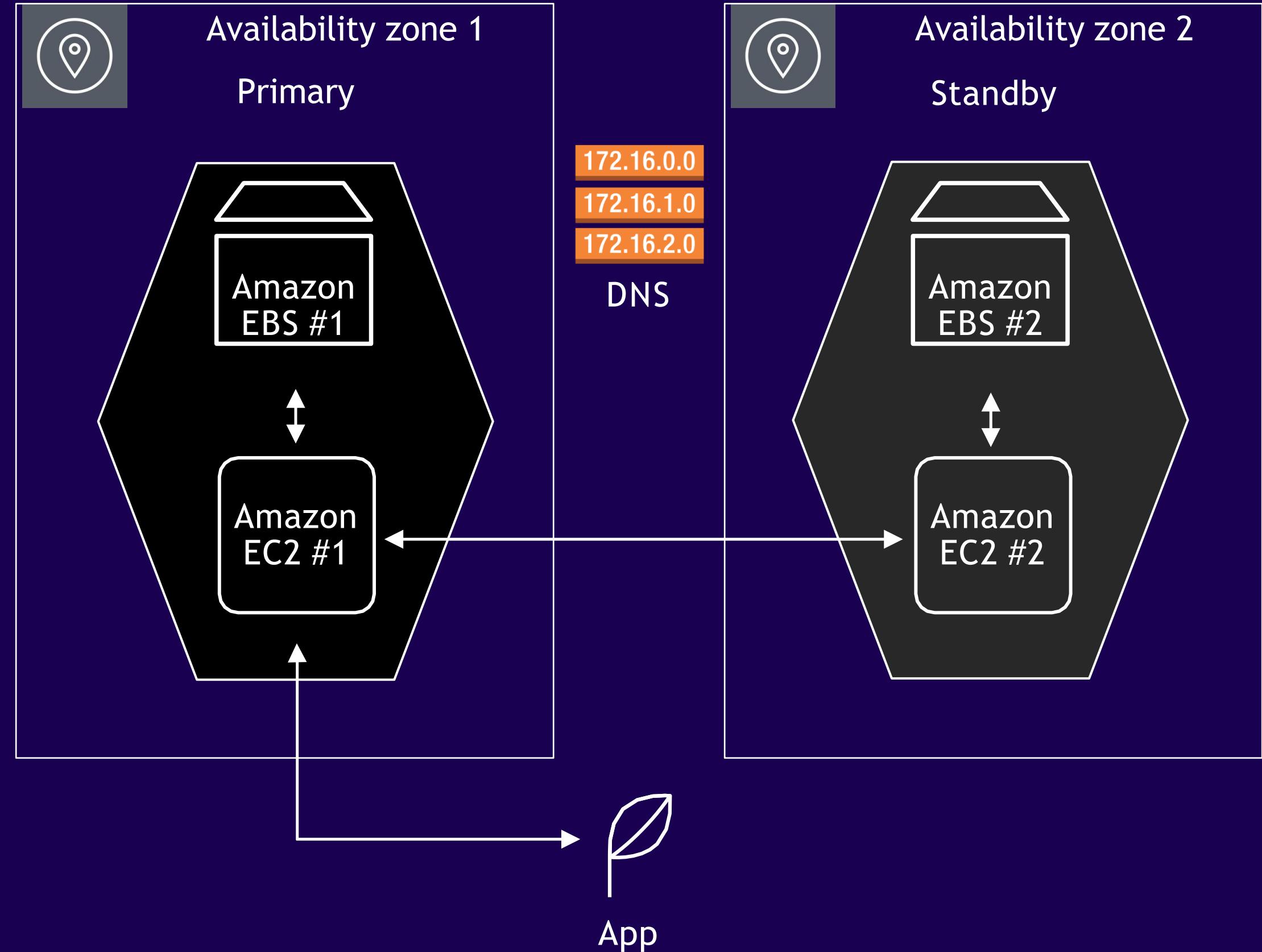
Security

Developer
Productivity

Automated, 0-RPO failover across AZs

Each host manages set of Amazon EBS volumes with a full copy of the data

DB instances are monitored by an external observer to maintain consensus over quorum

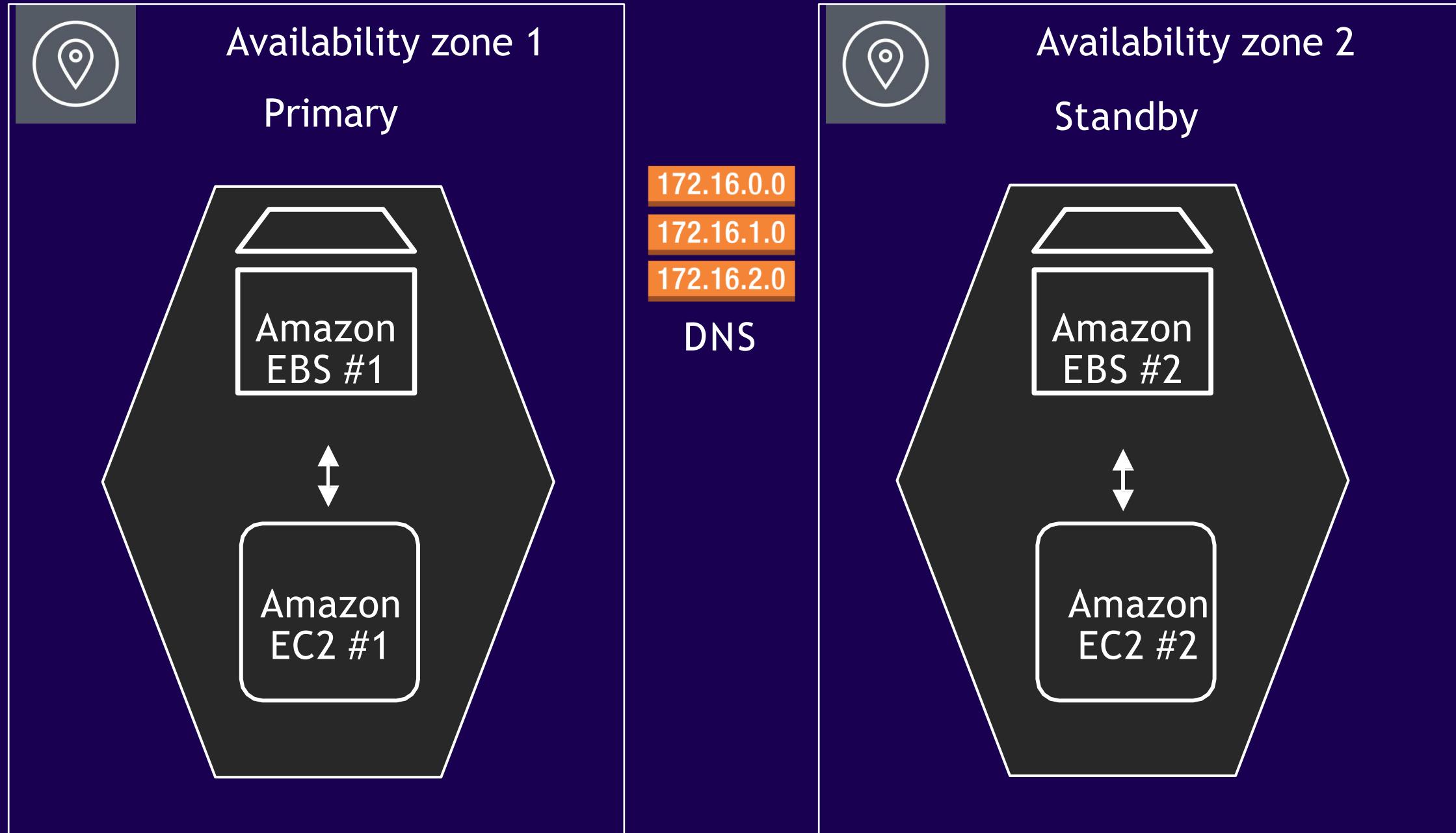


Automated, 0-RPO failover across AZs

Each host manages set of Amazon EBS volumes with a full copy of the data

DB instances are monitored by an external observer to maintain consensus over quorum

Failover initiated by automation or through RDS API



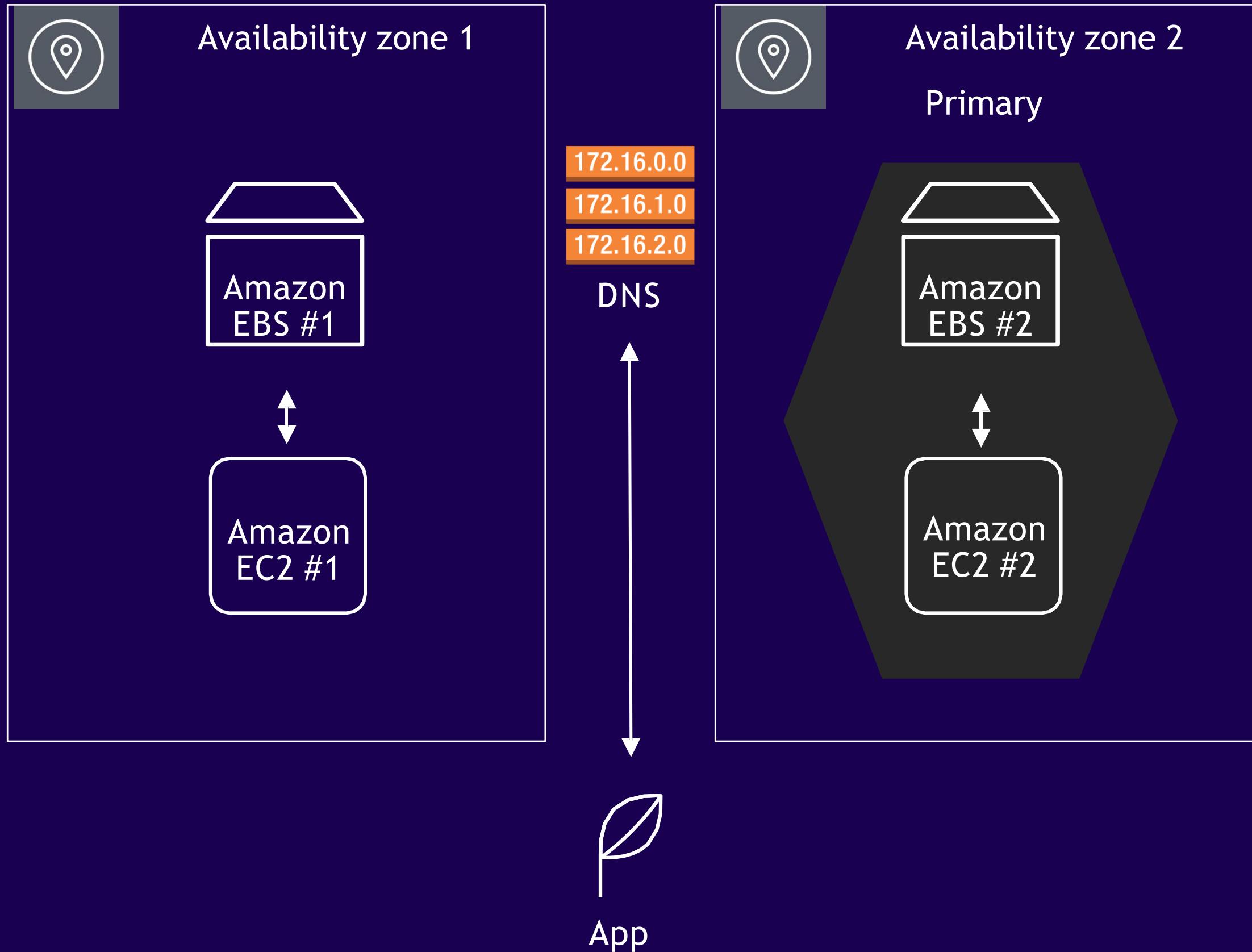
Automated, 0-RPO failover across AZs

Each host manages set of Amazon EBS volumes with a full copy of the data

DB instances are monitored by an external observer to maintain consensus over quorum

Failover initiated by automation or through RDS API

Redirection to the new primary instance is provided through DNS



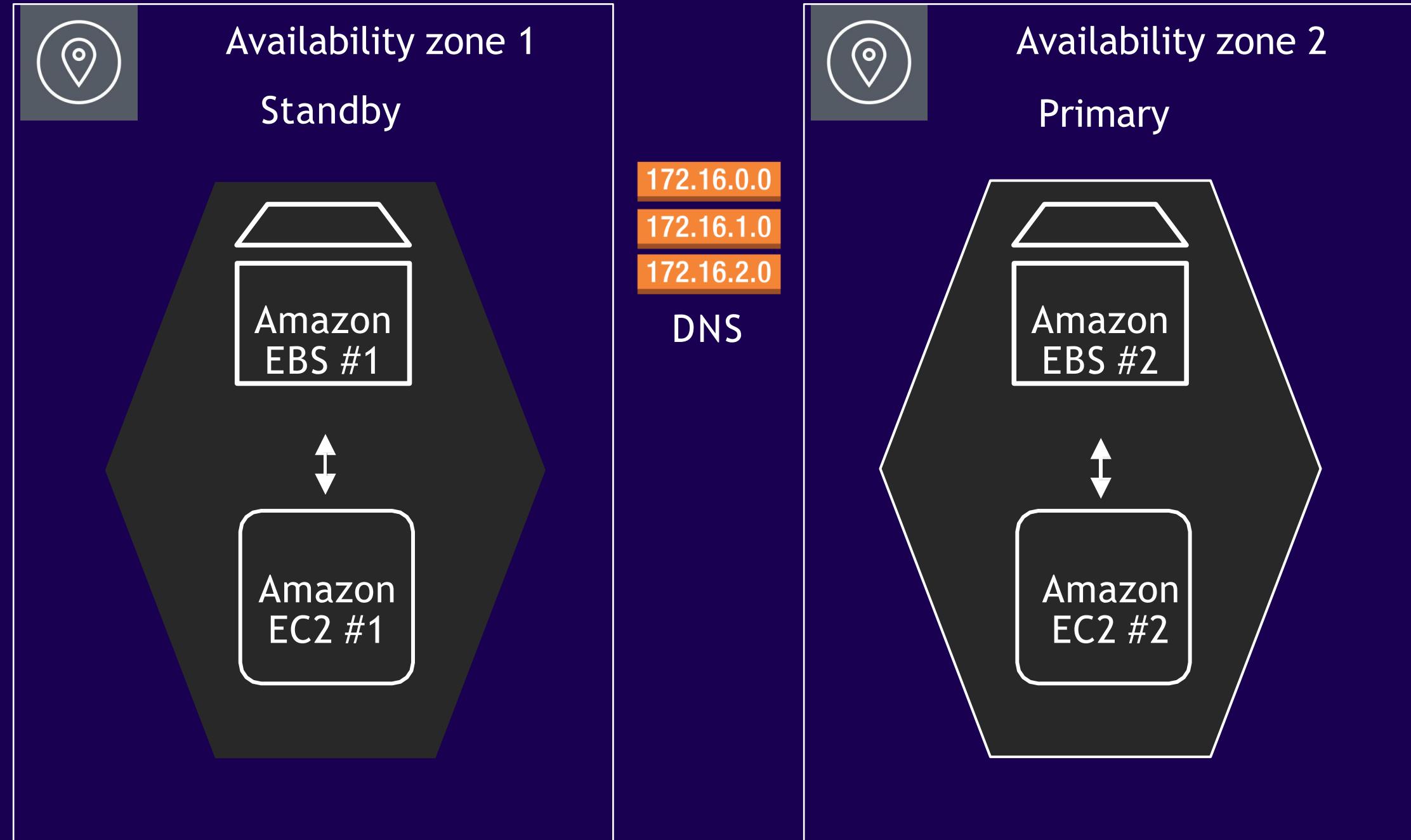
Automated, 0-RPO failover across AZs

Each host manages set of Amazon EBS volumes with a full copy of the data

DB instances are monitored by an external observer to maintain consensus over quorum

Failover initiated by automation or through RDS API

Redirection to the new primary instance is provided through DNS



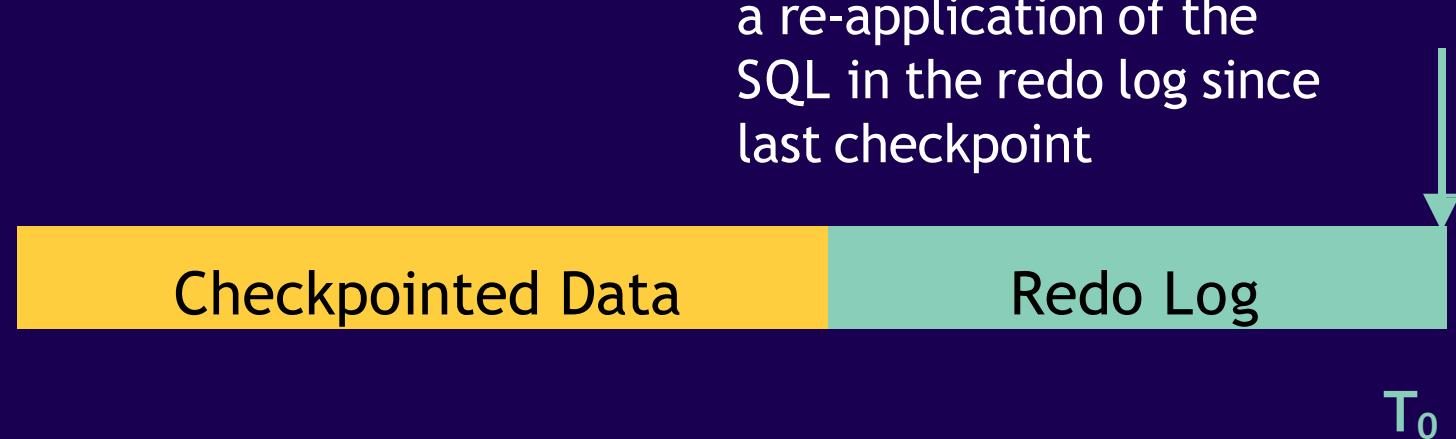
Aurora instant crash recovery

Traditional database

Have to replay logs since the last checkpoint

Typically 5 minutes between checkpoints

Single-threaded in MySQL; requires a large number of disk accesses



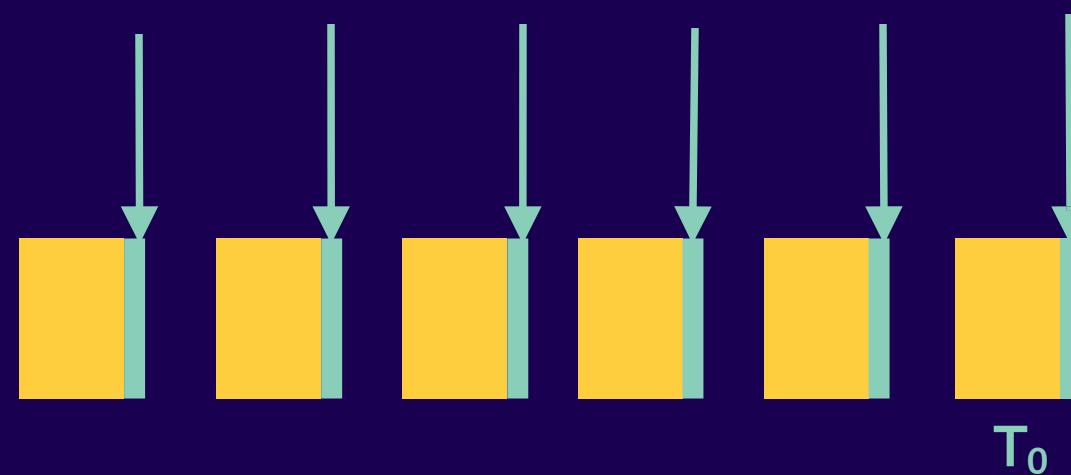
Amazon Aurora

Underlying storage replays redo records on demand as part of a disk read

Parallel, distributed, asynchronous

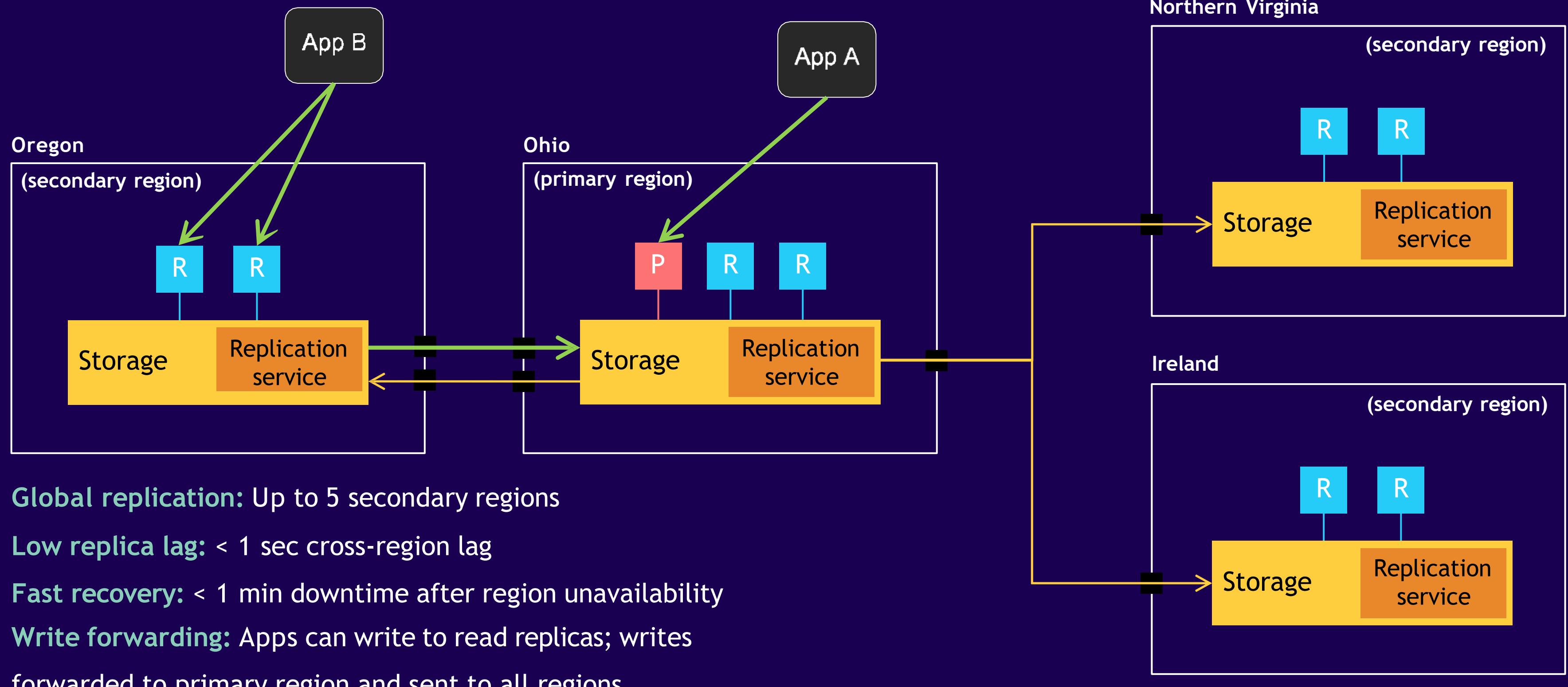
No replay for startup

Crash at T_0 will result in redo logs being applied to each segment on demand, in parallel, asynchronously



Aurora Global Database

FASTER DISASTER RECOVERY AND ENHANCED DATA LOCALITY



What do you want from your database service?

Choice

Manageability

Performance

Availability

Security

Developer
Productivity

How do I secure my database?



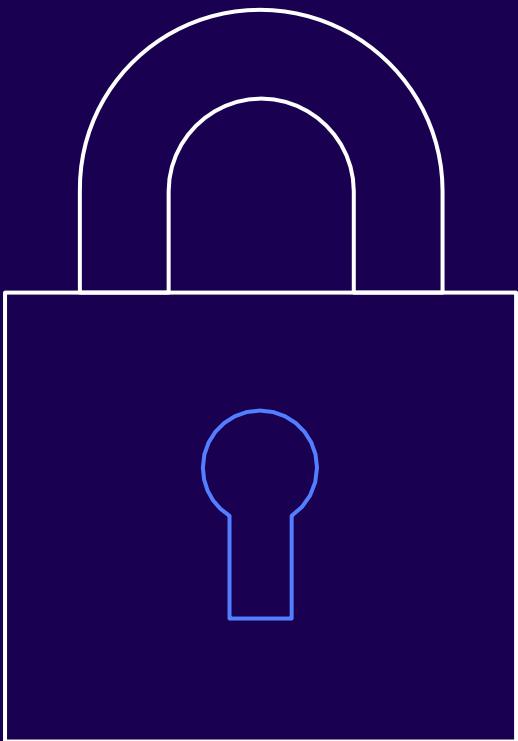
Amazon Virtual Private Cloud (Amazon VPC)
Network Isolation



AWS Identity and Access Management (IAM)
Resource-level permission controls



AWS Key Management Service (AWS KMS)
Encryption at rest



SSL
Protection for data in transit

Password validation

- Supports Amazon RDS for MySQL version 5.6, 5.7, and 8.0
- The MySQL DB instance performs password validation
- Install plugin

```
mysql> INSTALL PLUGIN validate_password SONAME 'validate_password.so';
Query OK, 0 rows affected (0.01 sec)
```

- Configure the Parameter group and reboot

Parameters								Cancel editing	Preview changes	Reset	Save changes
	Name	Values	Allowed values	Modifiable	Source	Apply type	Data type	Description			
<input checked="" type="checkbox"/>	validate-password	<input type="button" value="ON"/> <input type="button" value="OFF"/>	ON, OFF, FORCE, FORCE_PLUS_PERMANENT	true	user	static	string	This option controls how the server loads the validate_password plugin at startup.			

What do you want from your database service?

Choice

Manageability

Performance

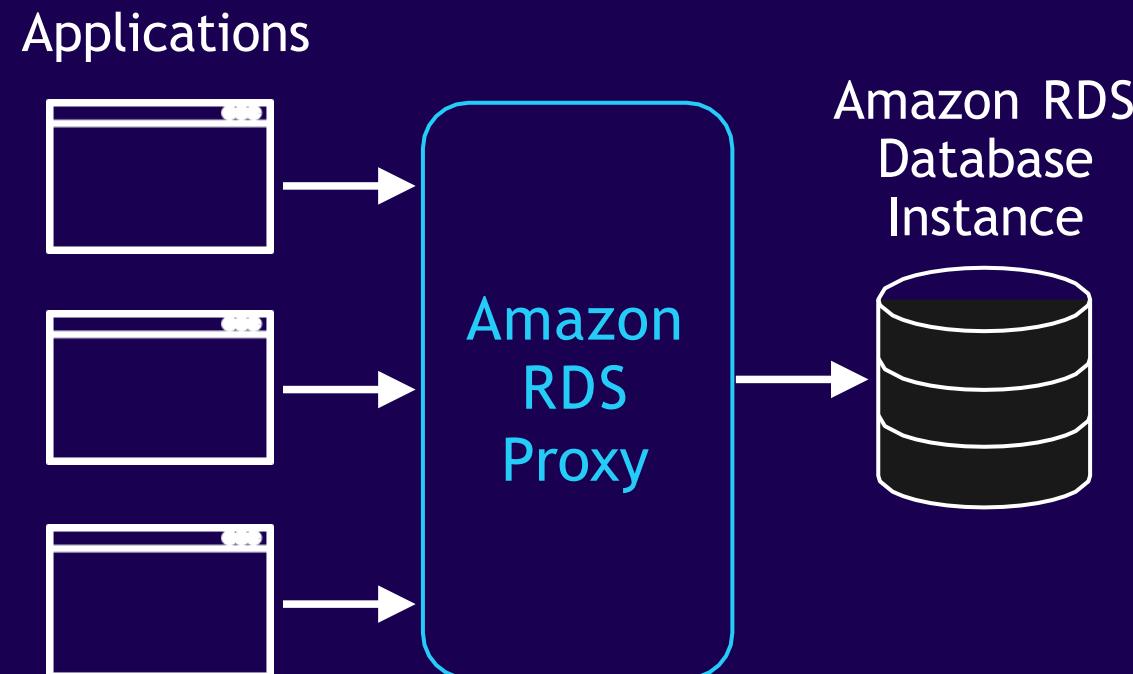
Availability

Security

Developer
Productivity

Amazon RDS Proxy

FULLY-MANAGED, HIGHLY-AVAILABLE DATABASE PROXY FOR AMAZON RDS



Connection Pooling

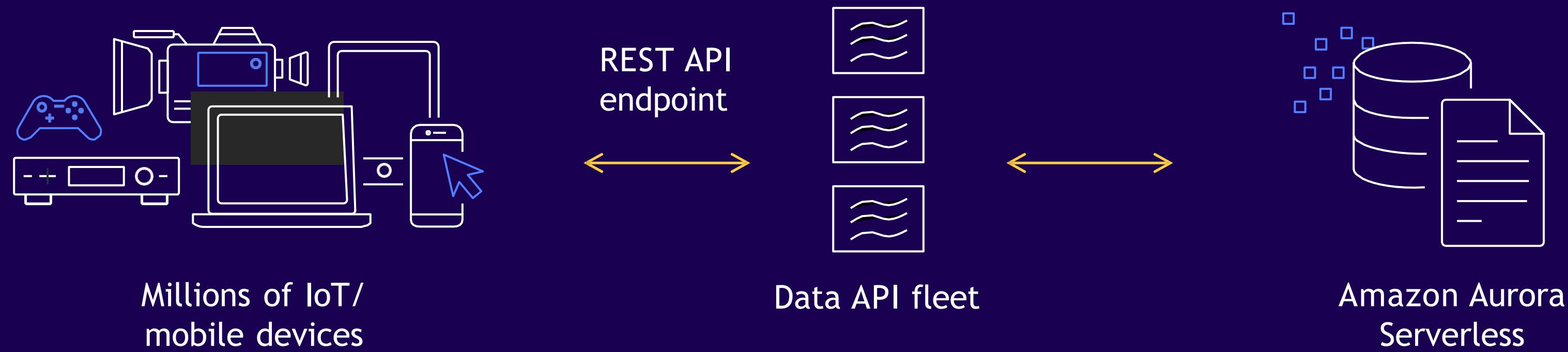
Supports a large number of application connections

Deployed across multiple AZs and fails over without losing a connection

Integrates with AWS Secrets Manager and AWS IAM

Get started with a few clicks in the console

Amazon RDS Data API for HTTP access



Serverless apps often have restrictions

Limited network connectivity to the DB

No persistent connection to the DB

Small client (e.g., IoT) with limited resources

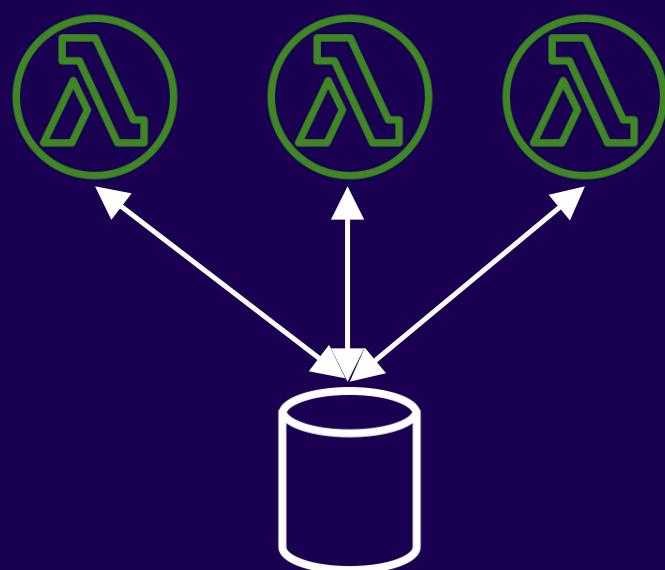
Amazon RDS Data API provides -

Public endpoint accessible via HTTP

Access without any client configuration

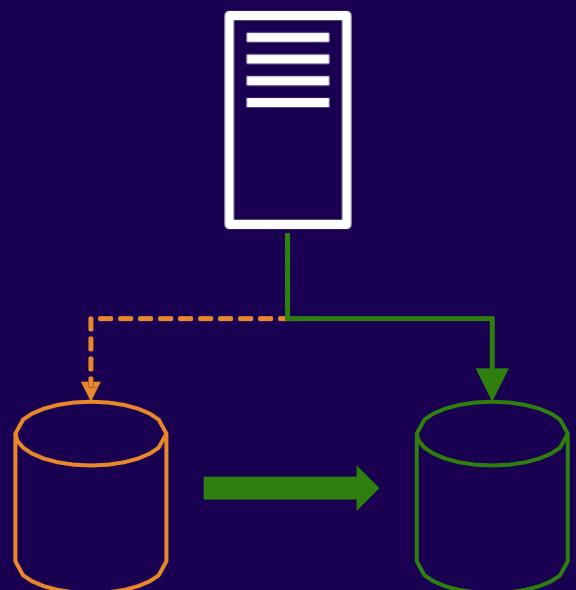
Today's applications demand

Scalability



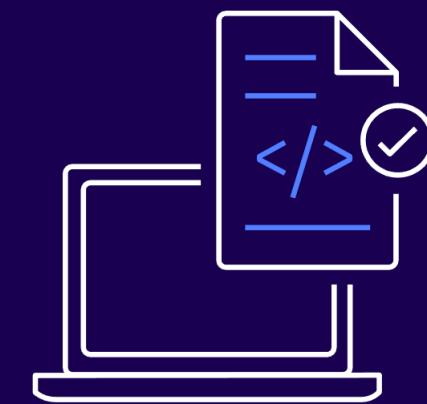
Scale to **hundreds of thousands of connections**

Availability



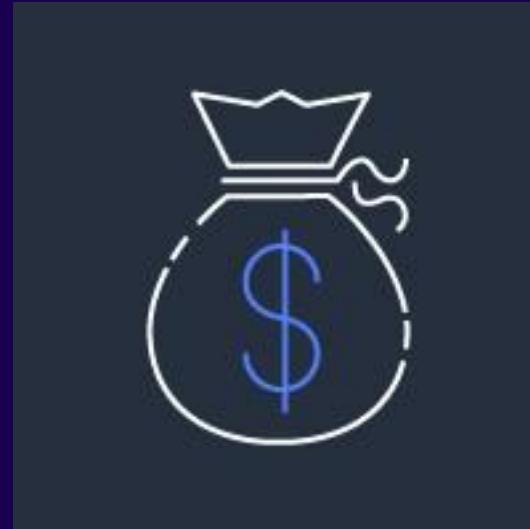
Increase app availability and reduce DB failover times

Security



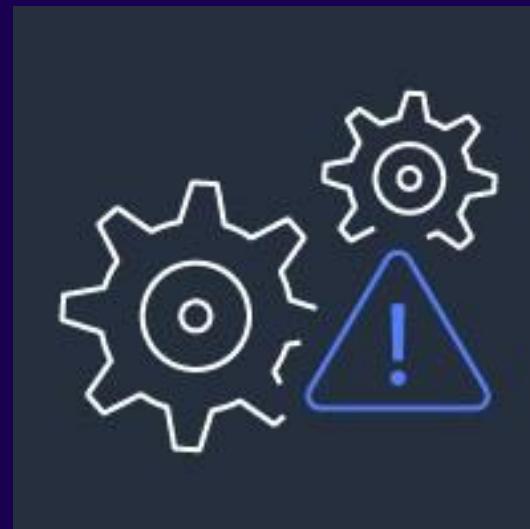
Manage app data security with DB access controls

Choices include



Overprovisioning

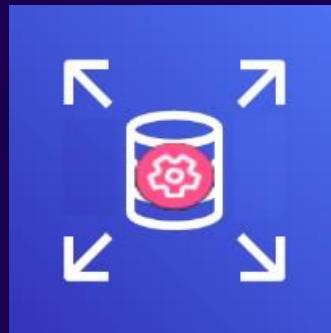
- Precious database compute resources spent on managing connections
- Maintain complex failure handling code to overcome transient failures



Self-managing a database proxy

- Deploy, patch, and manage yet another component
- Distribute across AZs for high availability

Amazon RDS Proxy: Skip the heavy lifting^{All}



Amazon
RDS Proxy

A fully managed, highly available database proxy for Amazon RDS and Amazon Aurora

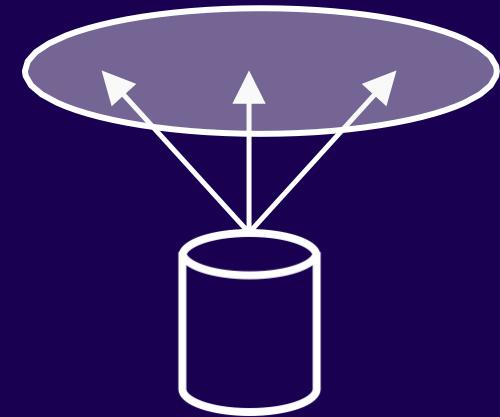
Pools and shares DB connections to make applications more scalable, more resilient to database failures, and more secure

Fully managed



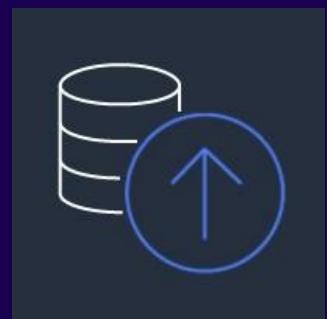
No need to deploy and maintain a proxy, highly available, MySQL- and PostgreSQL-compatible

Connection pooling



Pool and share DB connections for improved scalability

Fast and seamless failovers



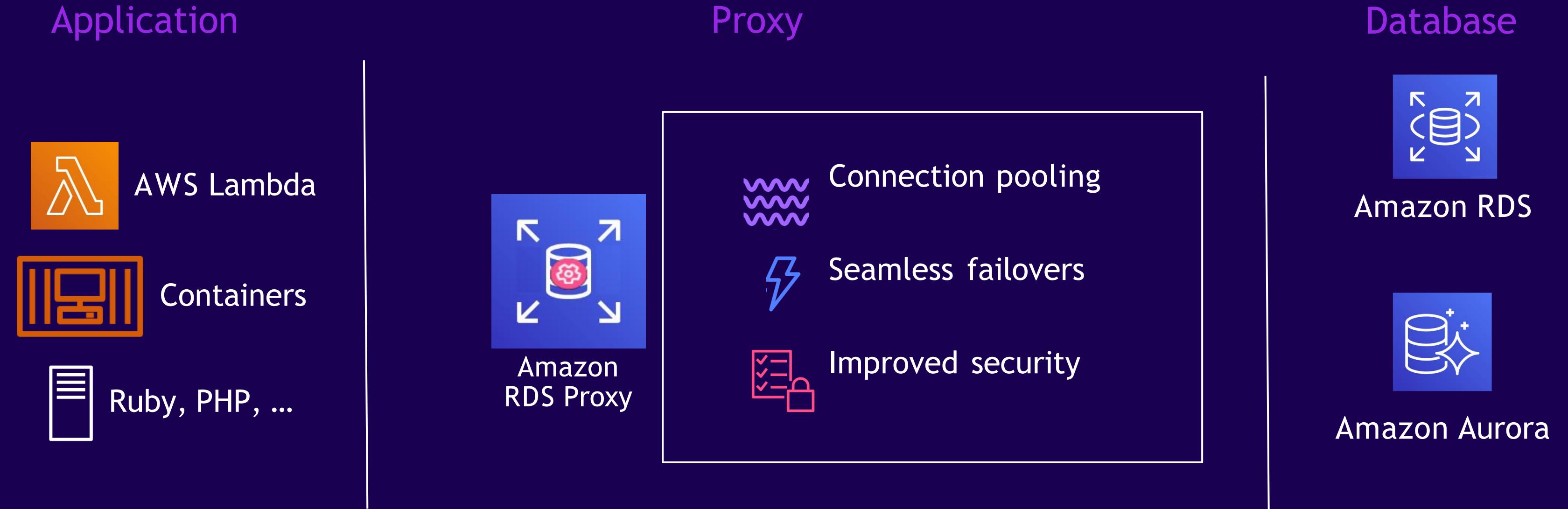
66% faster failovers and no loss of connectivity

Improved security

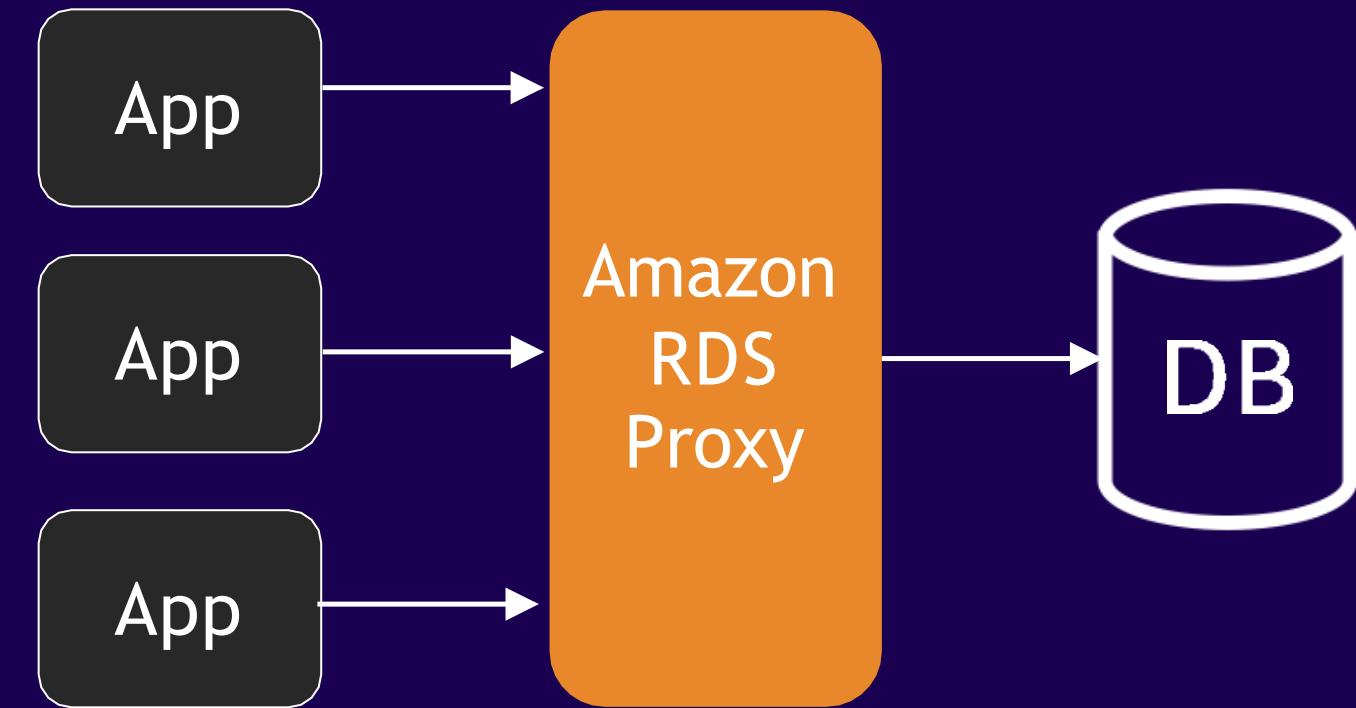


Store passwords in AWS Secrets Manager and enforce IAM authentication

How it works

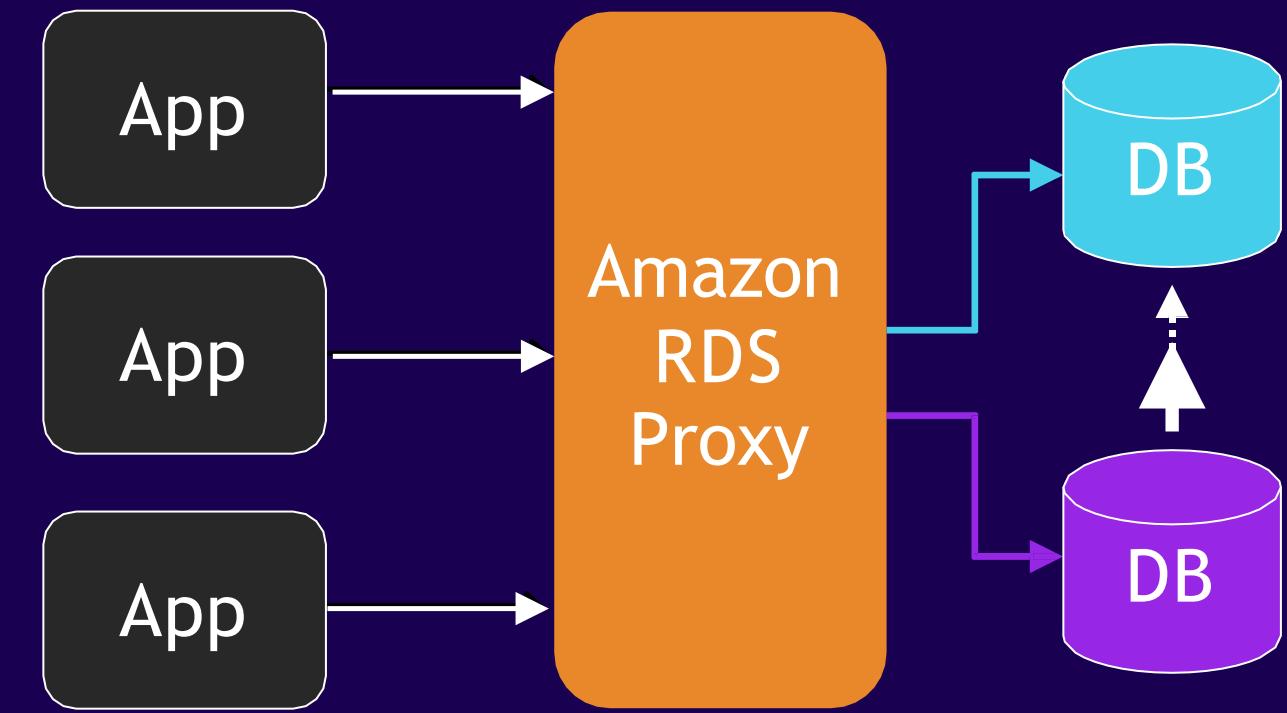


Connection pooling



Share database connections between transactions with **multiplexing**
Scale to support **hundreds of thousands of connections**

Seamless and fast failovers



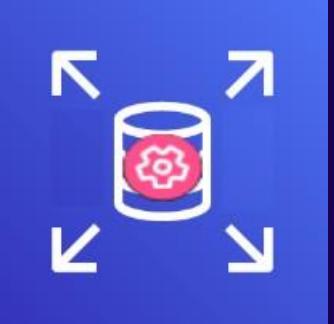
- Application **connections** are preserved and transactions are queued during failovers
- Detects failovers and connects to standby quicker, bypassing DNS caches and downstream TTLs
- Up to **66% faster** failover times

Amazon RDS Proxy authorization

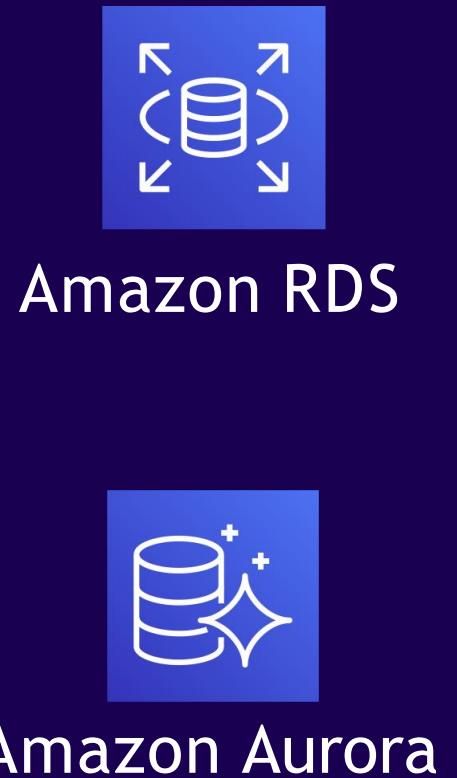
Application



Proxy

- 
- Connection pooling
 - Seamless failovers
 - Improved security

Database



Improved application security

Enforce IAM authentication with your relational databases

Connectivity

Secrets Manager secret(s) [Info](#)

Create or choose Secrets Manager secret(s) representing the credentials for database user accounts that the proxy can connect to.

Choose one or more secret(s)

[Create a new secret](#) 

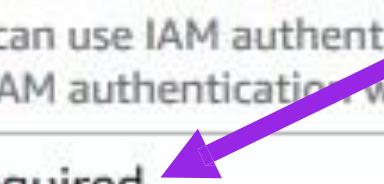
IAM role

Create or choose the IAM role the proxy will use to access the AWS Secrets Manager secret(s).

[Create IAM role](#)

IAM authentication

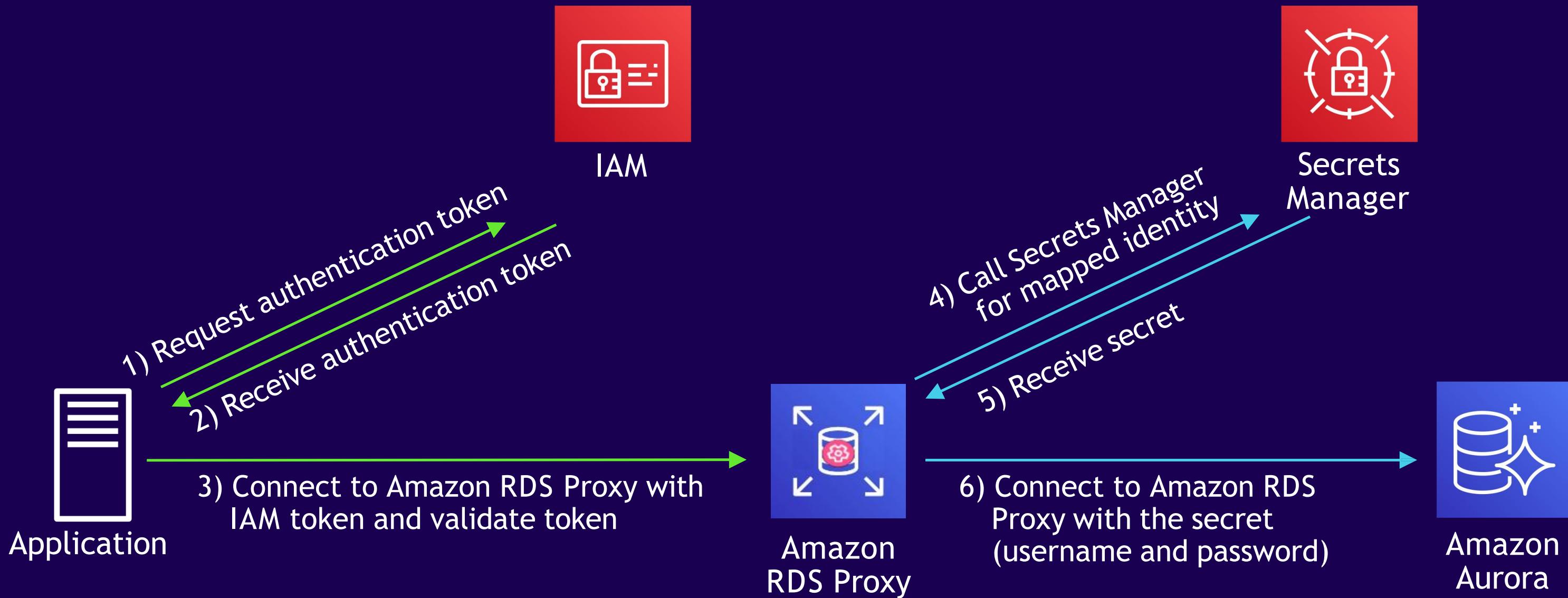
You can use IAM authentication to connect to the proxy in addition with specifying database credentials. Define how you want to use IAM authentication with the proxy. This will apply to all secrets selected above.

Required 

Improved application security

Most

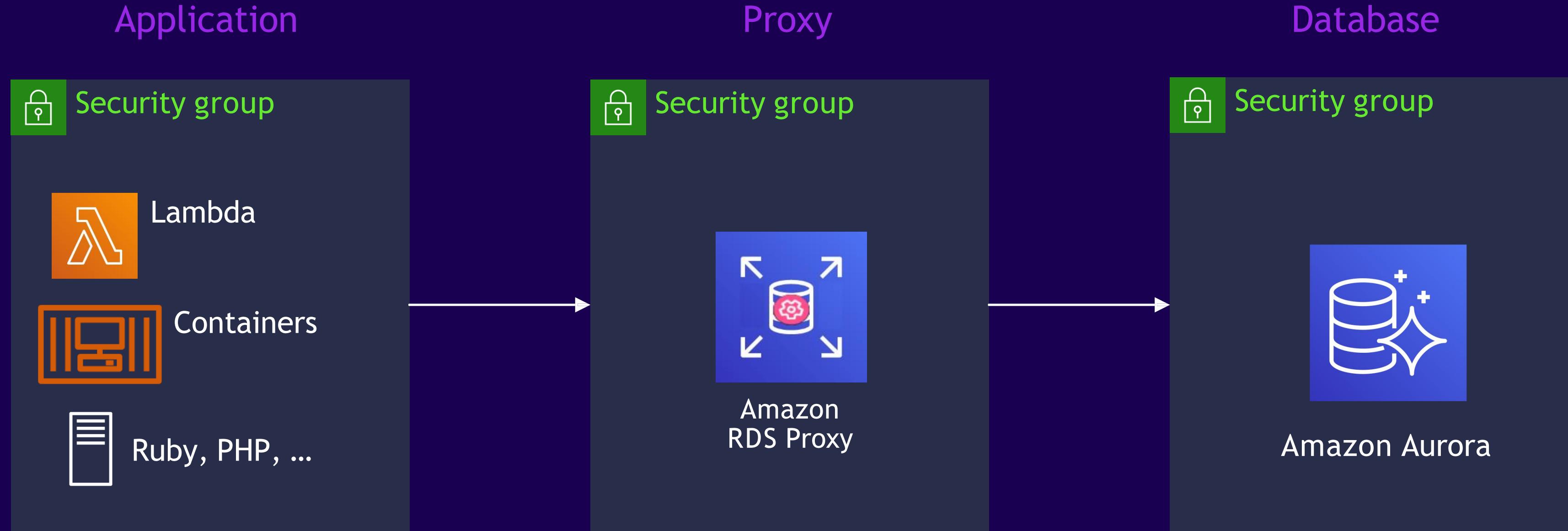
Centrally manage database credentials using Secrets Manager



Eliminate passwords embedded in code

```
...  
client = boto3.client("rds")  
DBEndPoint = os.environ.get("DBEndPoint")  
DatabaseName = os.environ.get("DatabaseName")  
DBUserName = os.environ.get("DBUserName")  
token = client.generate_db_auth_token(DBHostname=DBEndPoint, Port=3306,  
DBUsername=DBUserName)sslCert = {'ca': './AmazonRootCA1.pem'}  
conn = pymysql.connect(  
    host=DBEndPoint,  
    port=3306,  
    database=DatabaseName,  
    user=DBUserName,  
    password=token,  
    ssl=sslCert,  
    connect_timeout=5  
...  
...
```

Amazon RDS Proxy network security



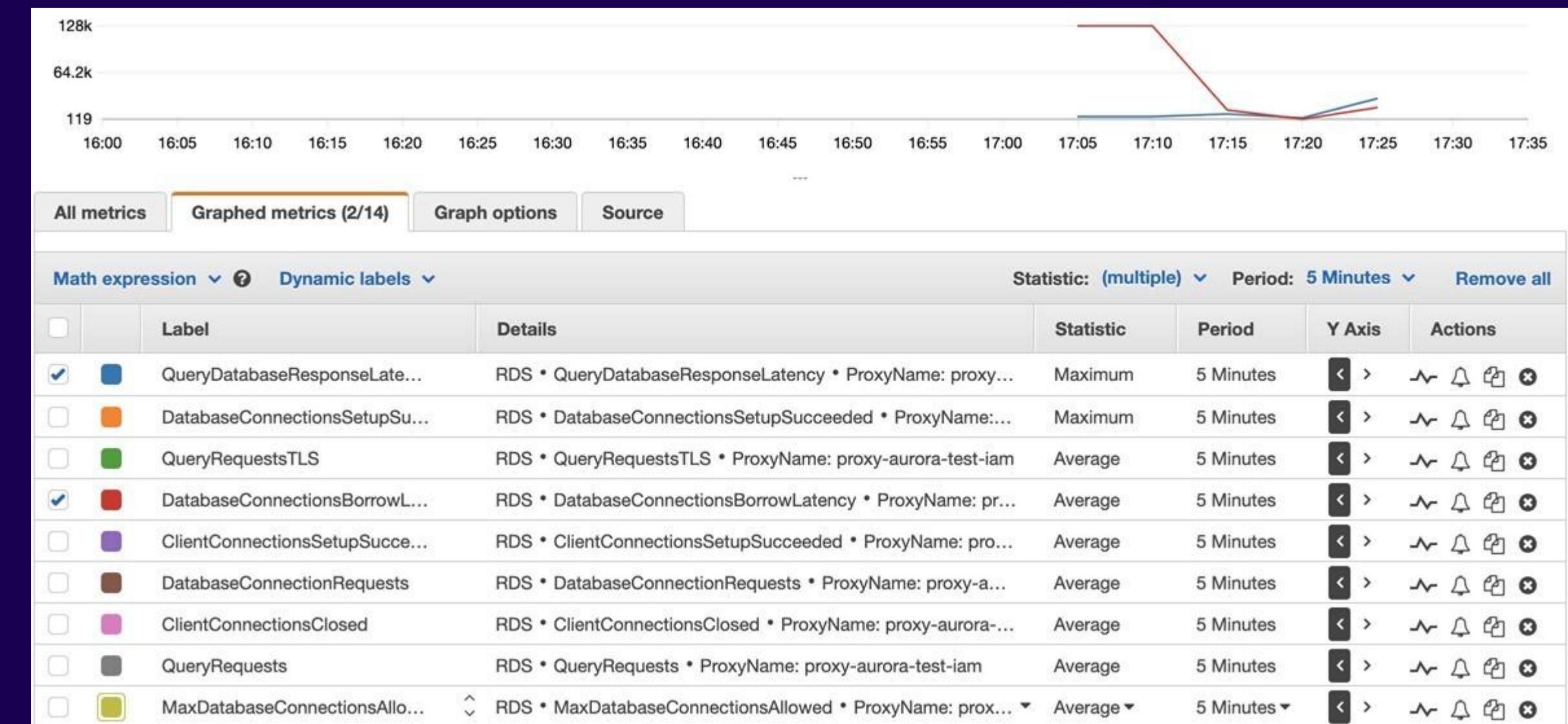
Monitoring Amazon RDS

99.99% SLA

Proxy

You can monitor Amazon RDS Proxy using 24 Amazon CloudWatch metrics

- AvailabilityPercentage
- ClientConnections
- ClientConnectionsClosed
- ClientConnectionsNoTLS
- ClientConnectionsReceived
- ClientConnectionsSetupFailedAuth
- ClientConnectionsSetupSucceeded
- ClientConnectionsTLS
- DatabaseConnectionRequests
- DatabaseConnectionRequestsWithTLS
- **DatabaseConnections**
- DatabaseConnectionsBorrowLatency
- **DatabaseConnectionsCurrentlyBorrowed**
- DatabaseConnectionsCurrentlyInTransaction
- **DatabaseConnectionsCurrentlySessionPinned**
- DatabaseConnectionsSetupFailed
- DatabaseConnectionsSetupSucceeded
- DatabaseConnectionsWithTLS
- MaxDatabaseConnectionsAllowed
- **QueryDatabaseResponseLatency**
- QueryRequests
- QueryRequestsNoTLS
- QueryRequestsTLS
- QueryResponseLatency



AWS AURORA

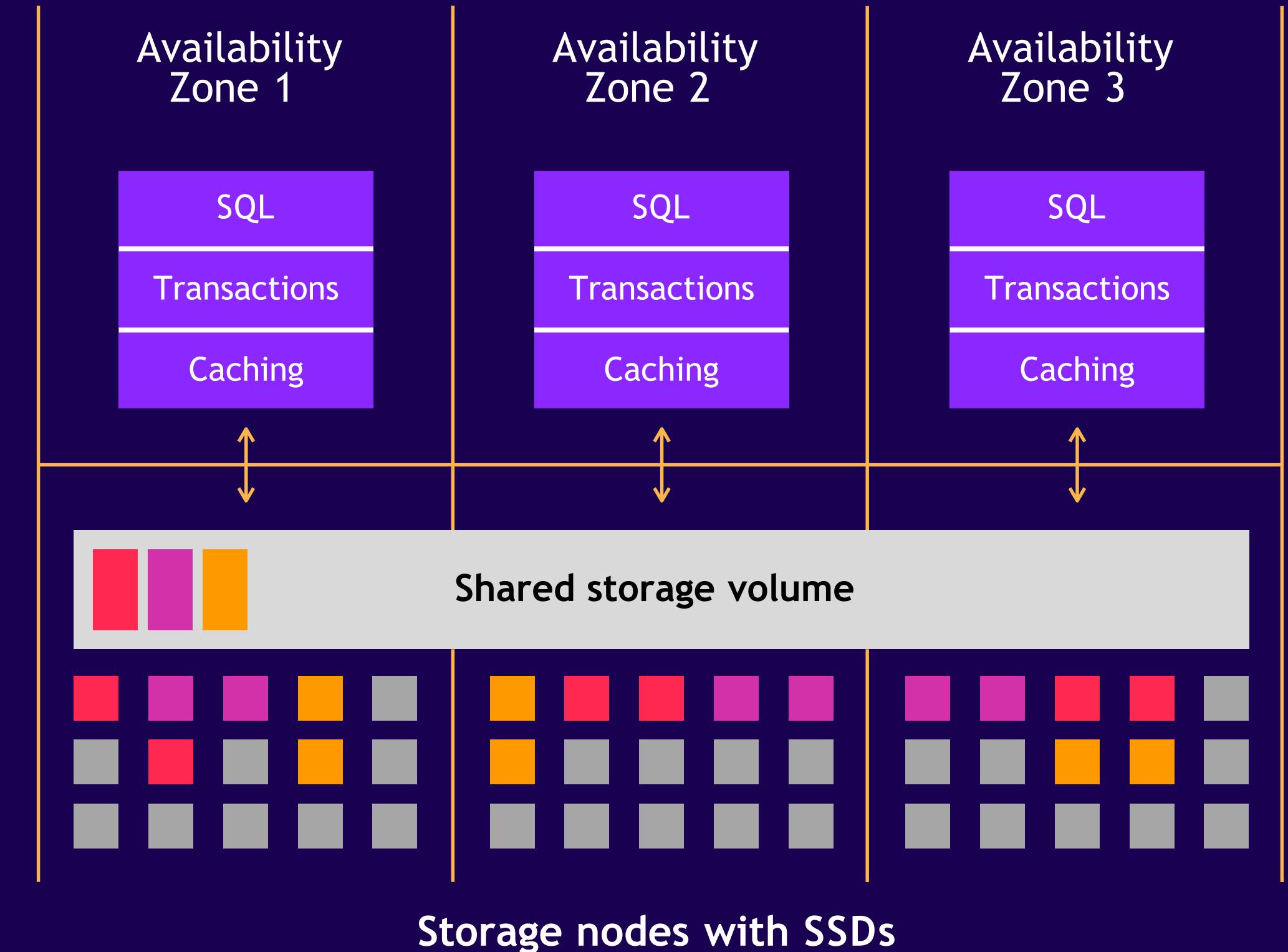
Aurora scale-out, distributed architecture

Purpose-built log-structured distributed storage system designed for databases

Storage volume is striped across hundreds of storage nodes distributed over 3 different availability zones

Six copies of data, two copies in each availability zone to protect against AZ+1 failures

Data is written in 10GB “protection groups”, growing automatically up to 64TB



Six-way replicated storage

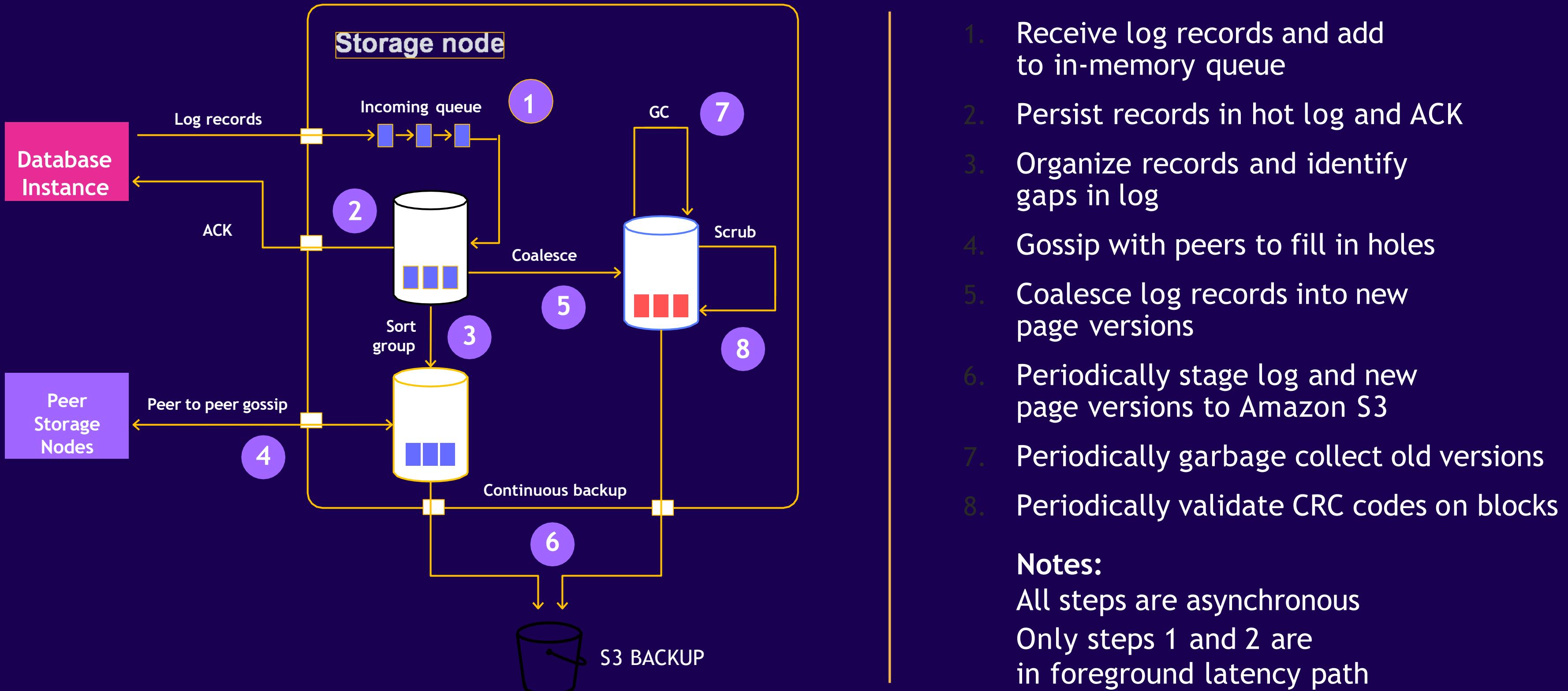


Data is written to all 6 nodes asynchronously, in parallel

Writes require a quorum of 4/6 nodes. Reads require up to 3/6 nodes (for recovery)

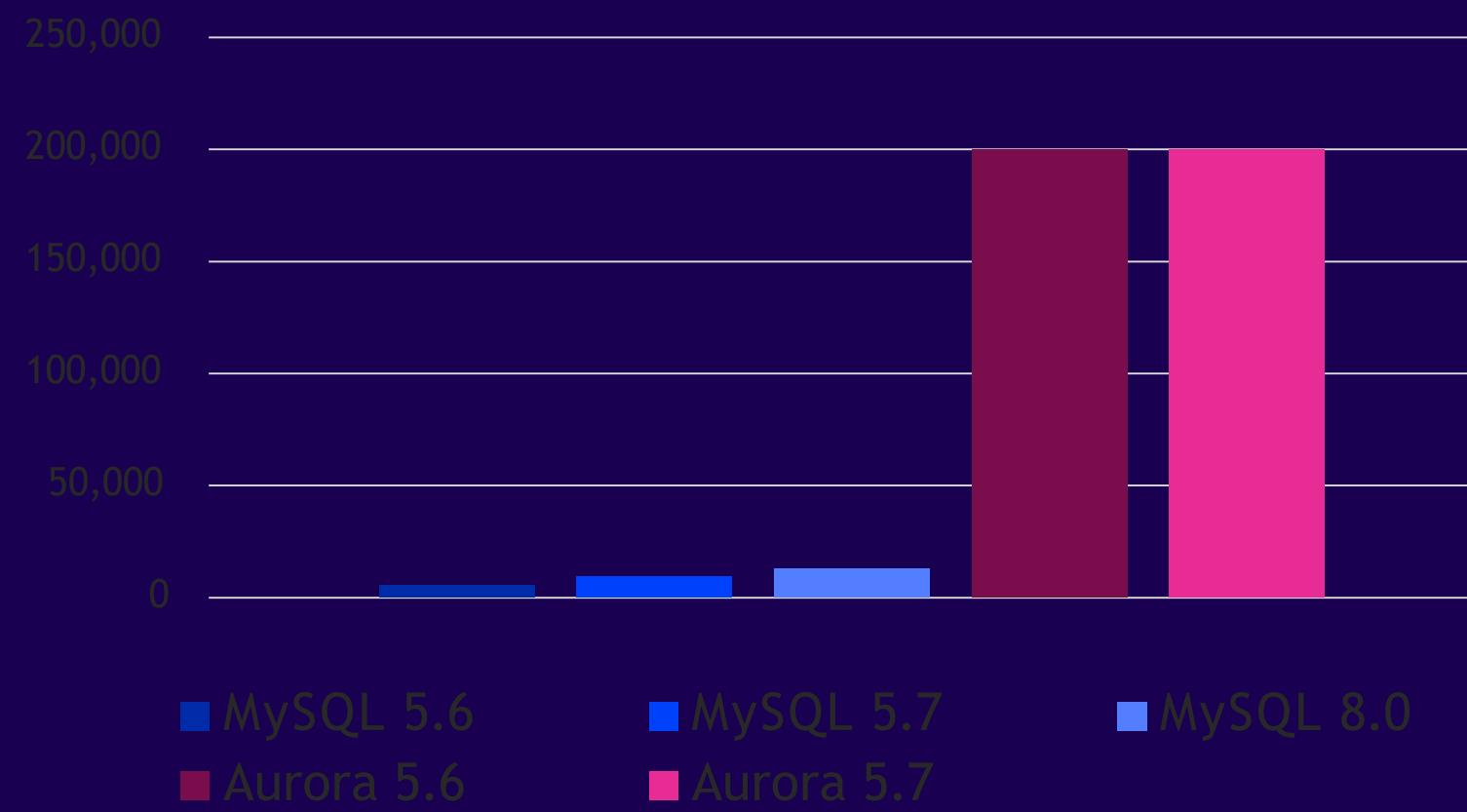
Peer to peer “gossip protocol” is used for repairs

I/O flow in Aurora storage node

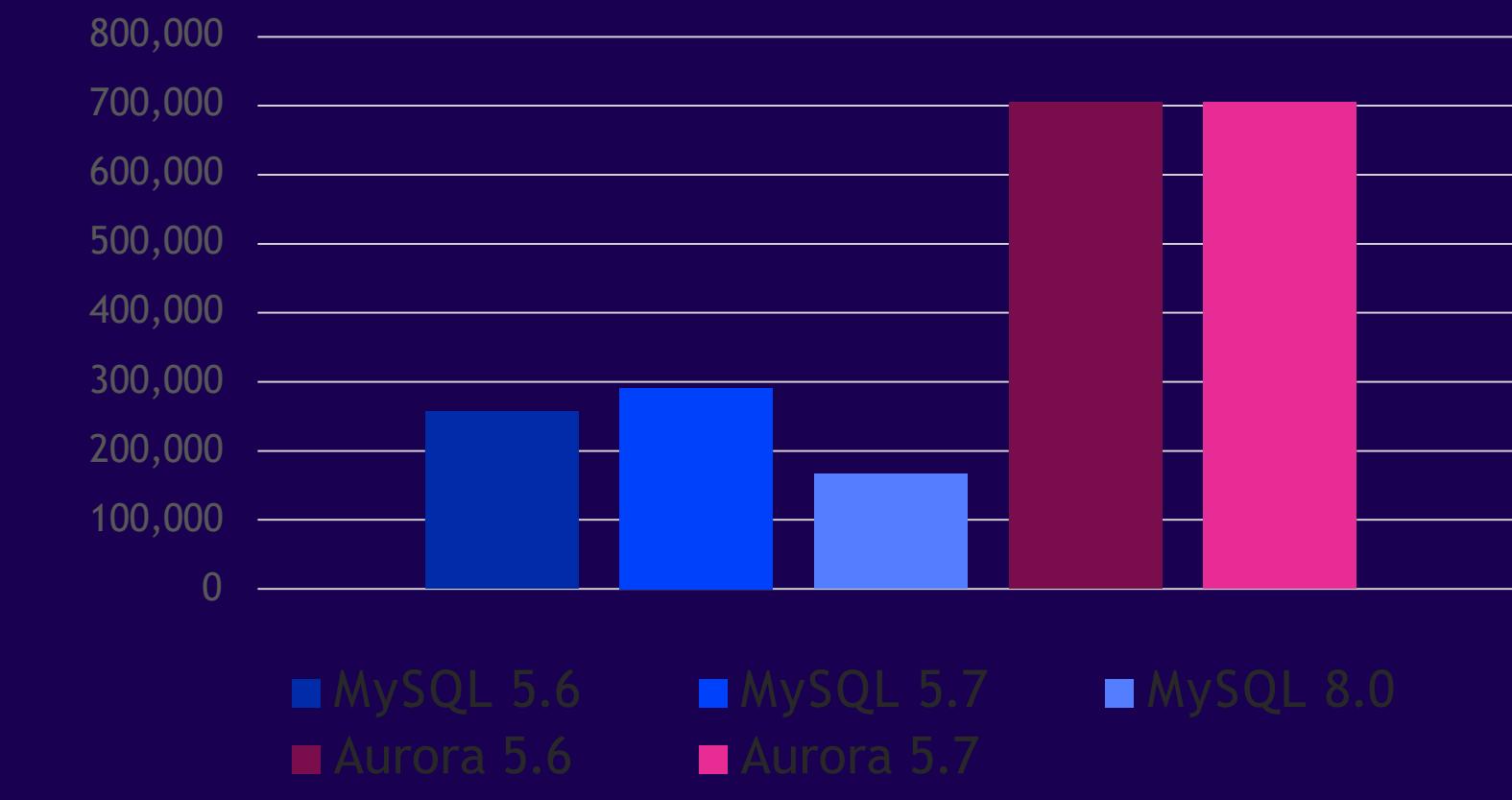


Write and read throughput

Aurora MySQL has 5x the throughput of MySQL



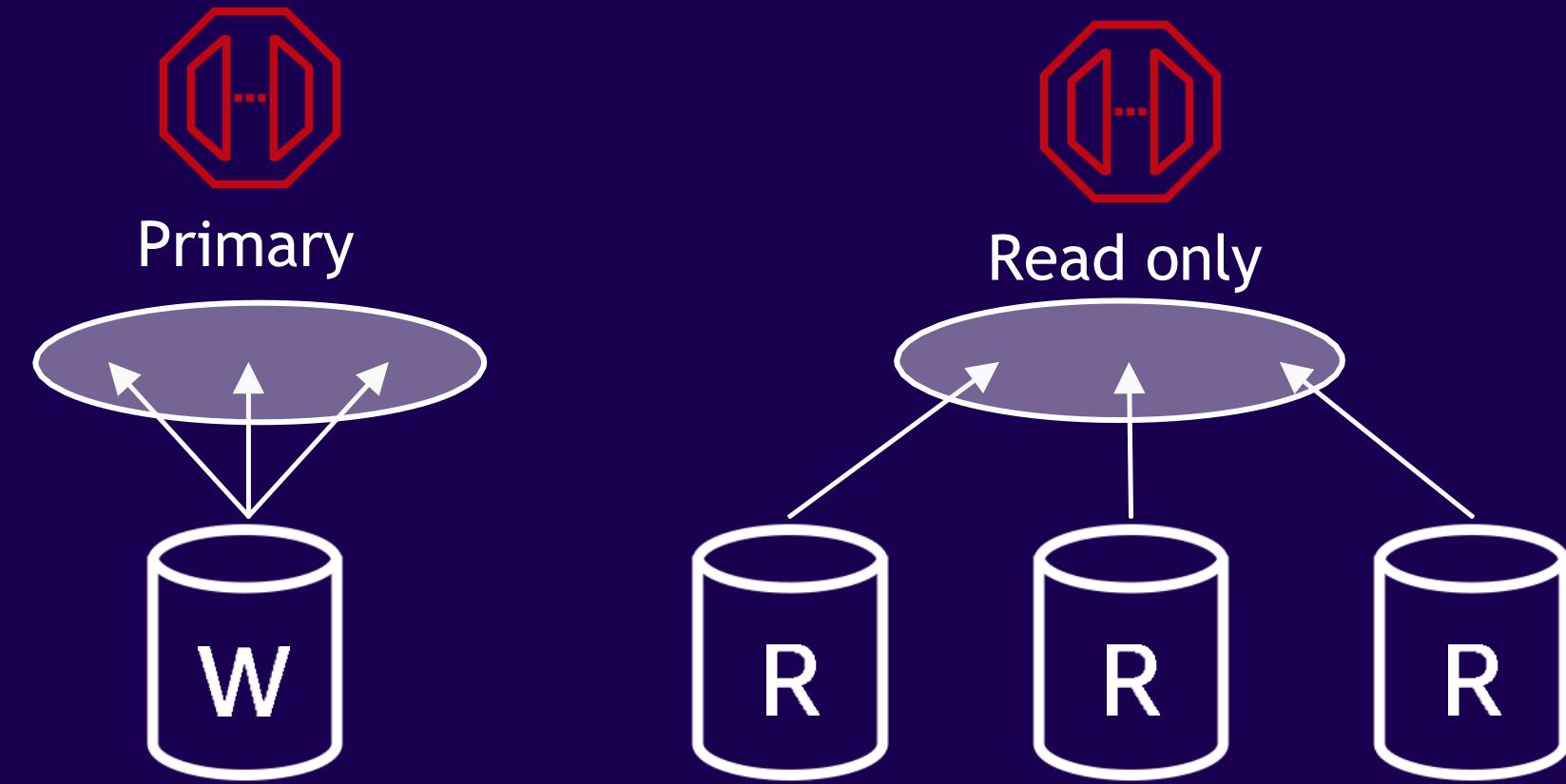
Write throughput



Read throughput

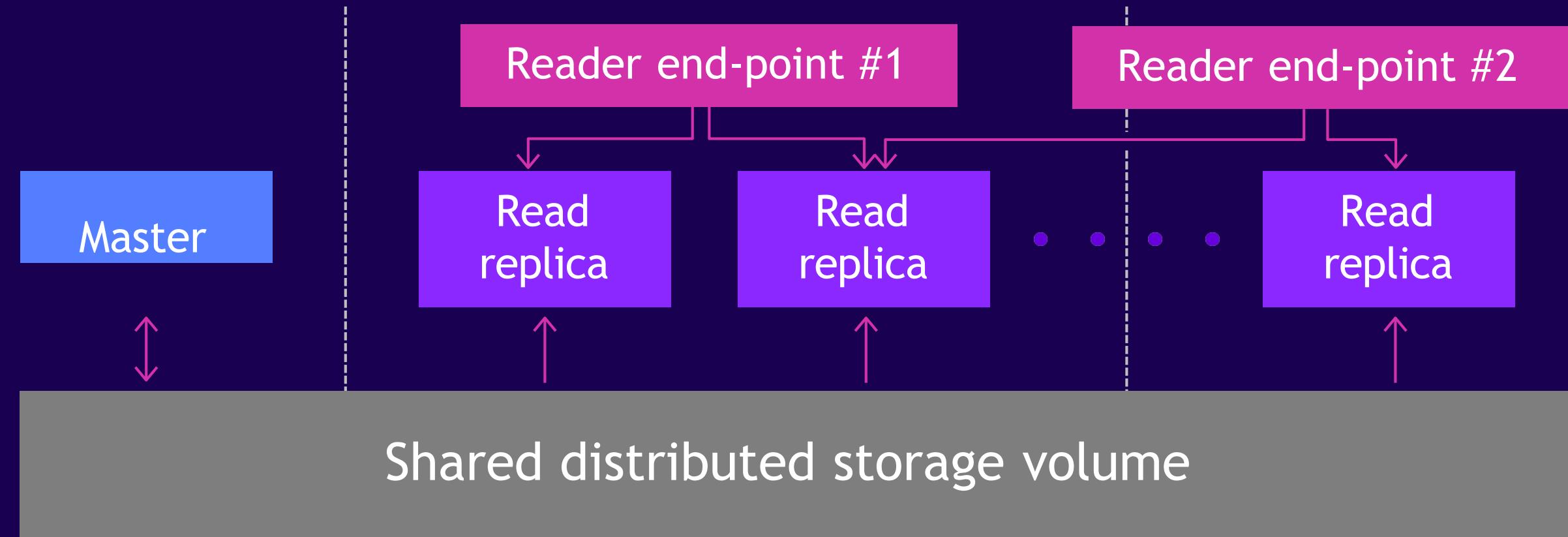
Using Sysbench with 250 tables and 200,000 rows per table on R4.16XL

Aurora Read Replica support



- Scale-out reads on Aurora Read Replicas
- Pool and share read-only connections
- Transaction-level load balancing
- Coming soon for Aurora MySQL and Aurora PostgreSQL read replicas

Aurora read replicas



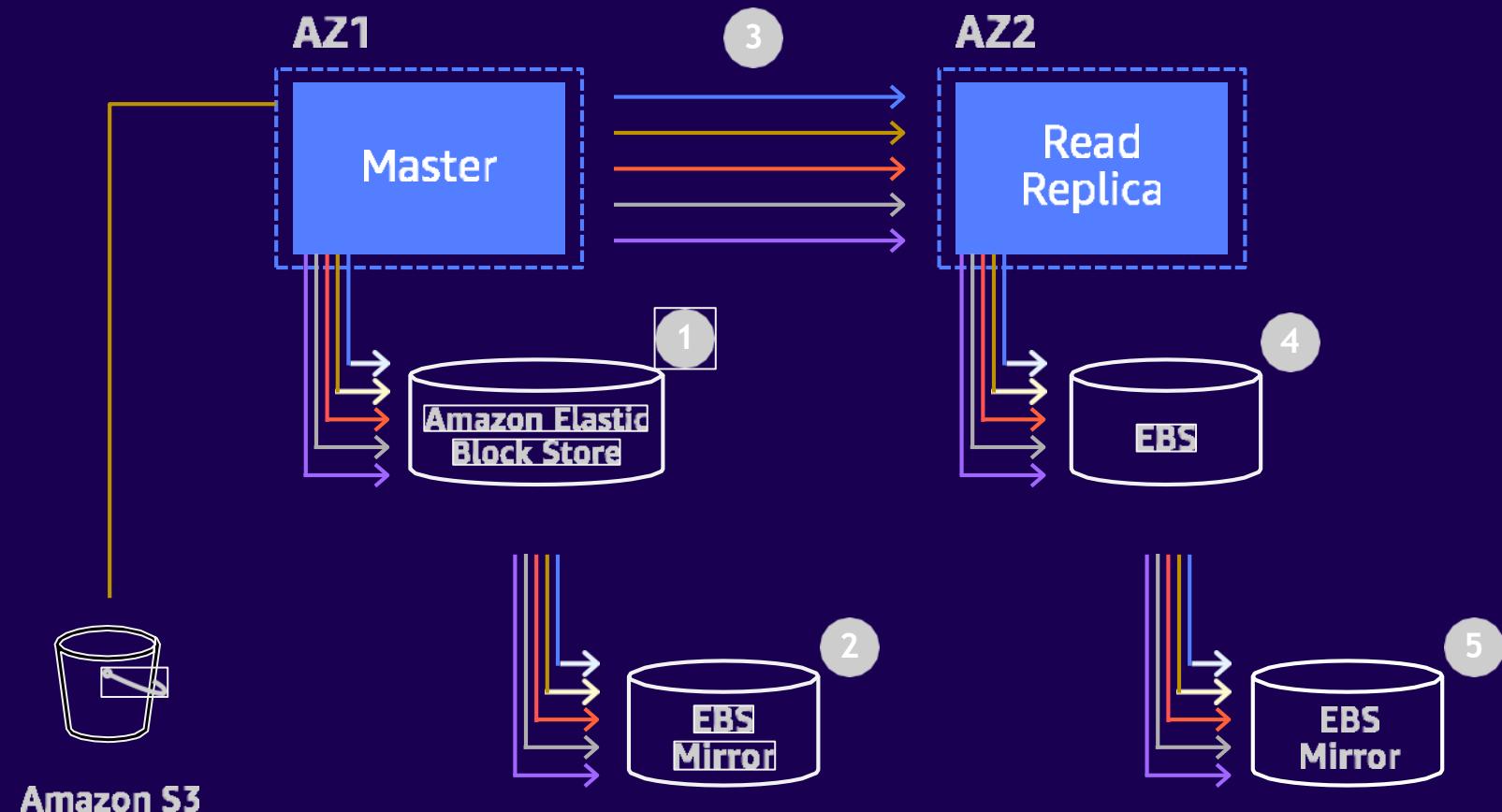
Up to 15 promotable read replicas across three availability zones

Redo log based (physical) replication leads to low replica lag—typically 20-40 ms

Custom reader end-point with configurable failover order

MySQL vs. Aurora I/O profile

MySQL with replica



MySQL I/O profile for 30 min Sysbench run

0.78MM transactions

7.4 I/Os per transaction

Type of write

→ Log

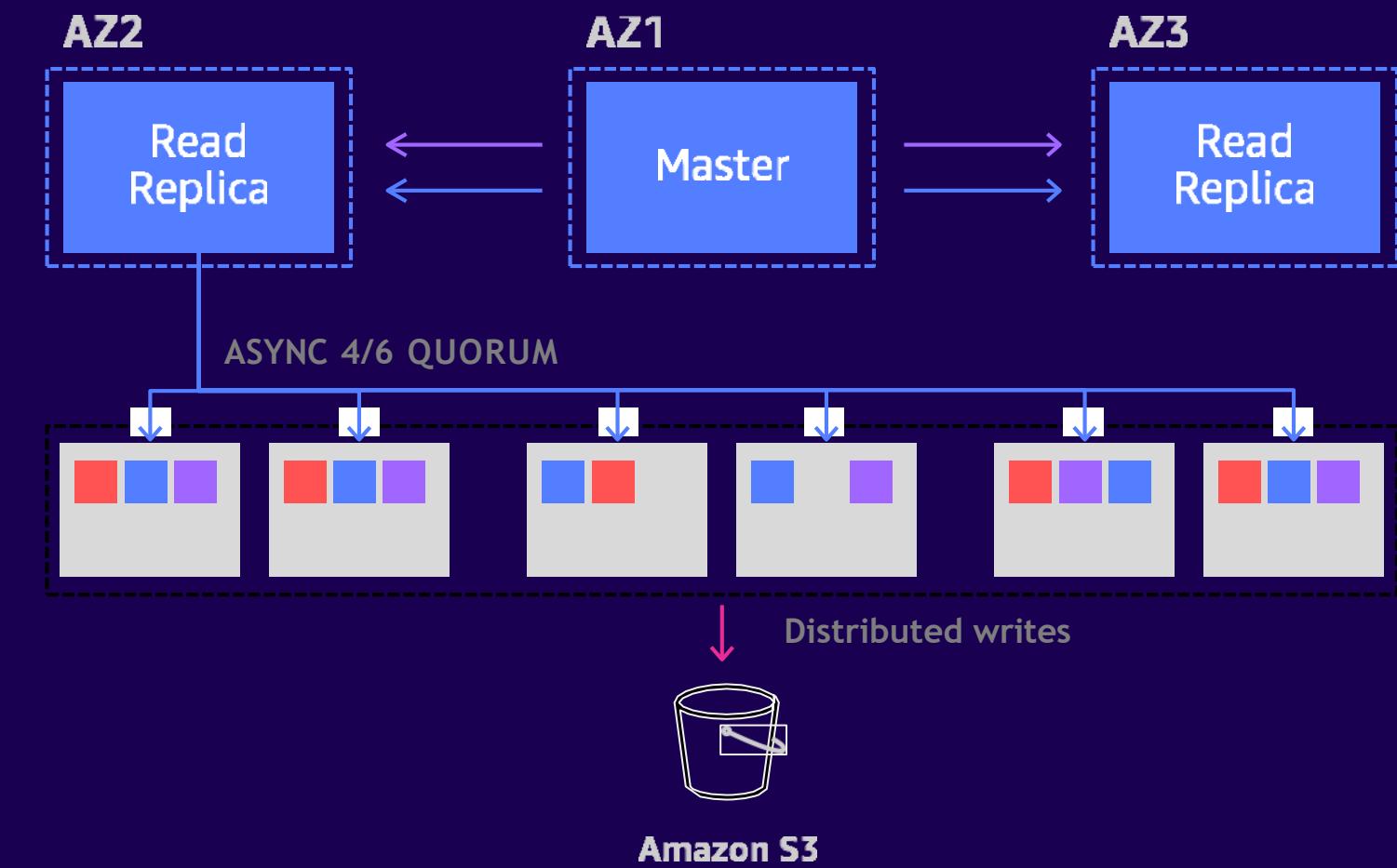
→ Binlog

→ Data

→ Double write

→ FRM files

Amazon Aurora



Aurora I/O profile for 30 min Sysbench run

27MM transactions

35X More

0.95 I/Os per transaction

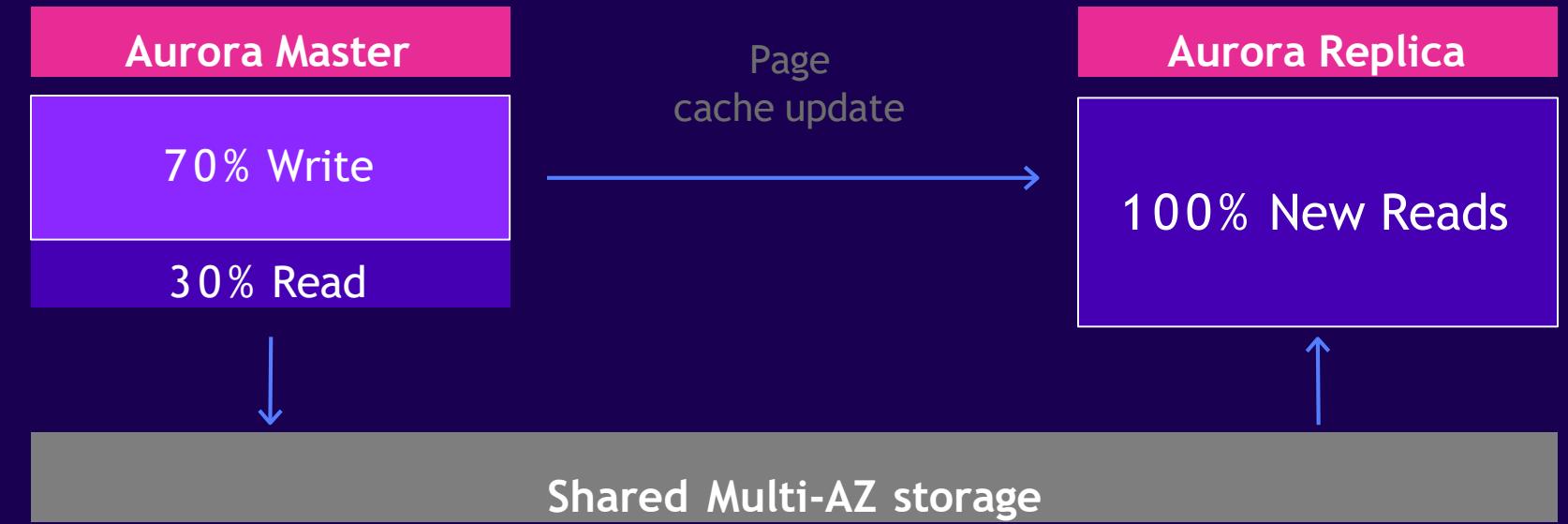
7.7X Less

Aurora read replicas are dedicated to reads

MySQL read scaling



Amazon Aurora read scaling



Logical using **delta** changes

Same write workload

Independent storage

Physical using **delta** changes

No writes on replica

Shared storage

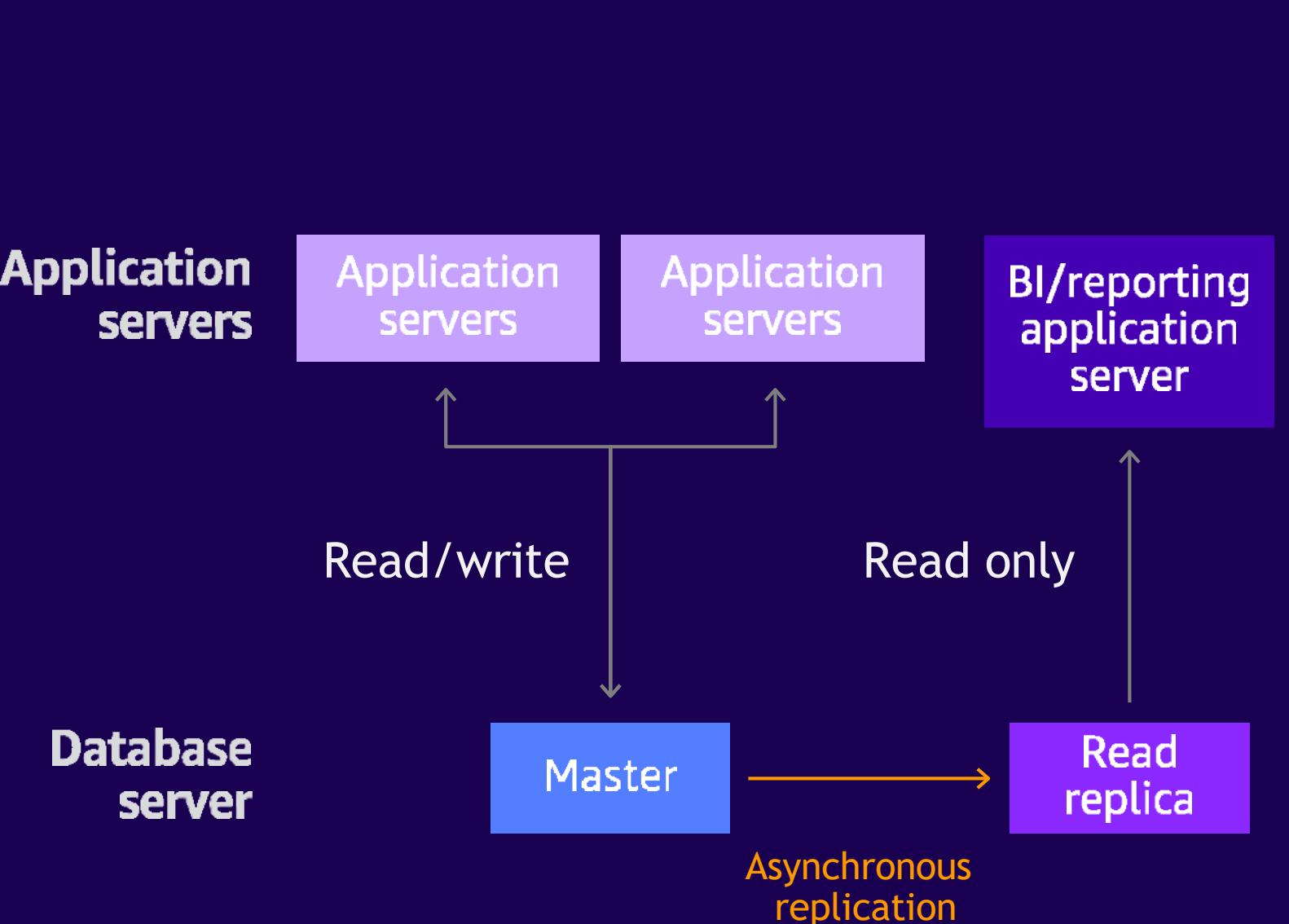
Aurora read scaling options

15 promotable read replicas per cluster

Auto-scaling to automatically add & remove replicas

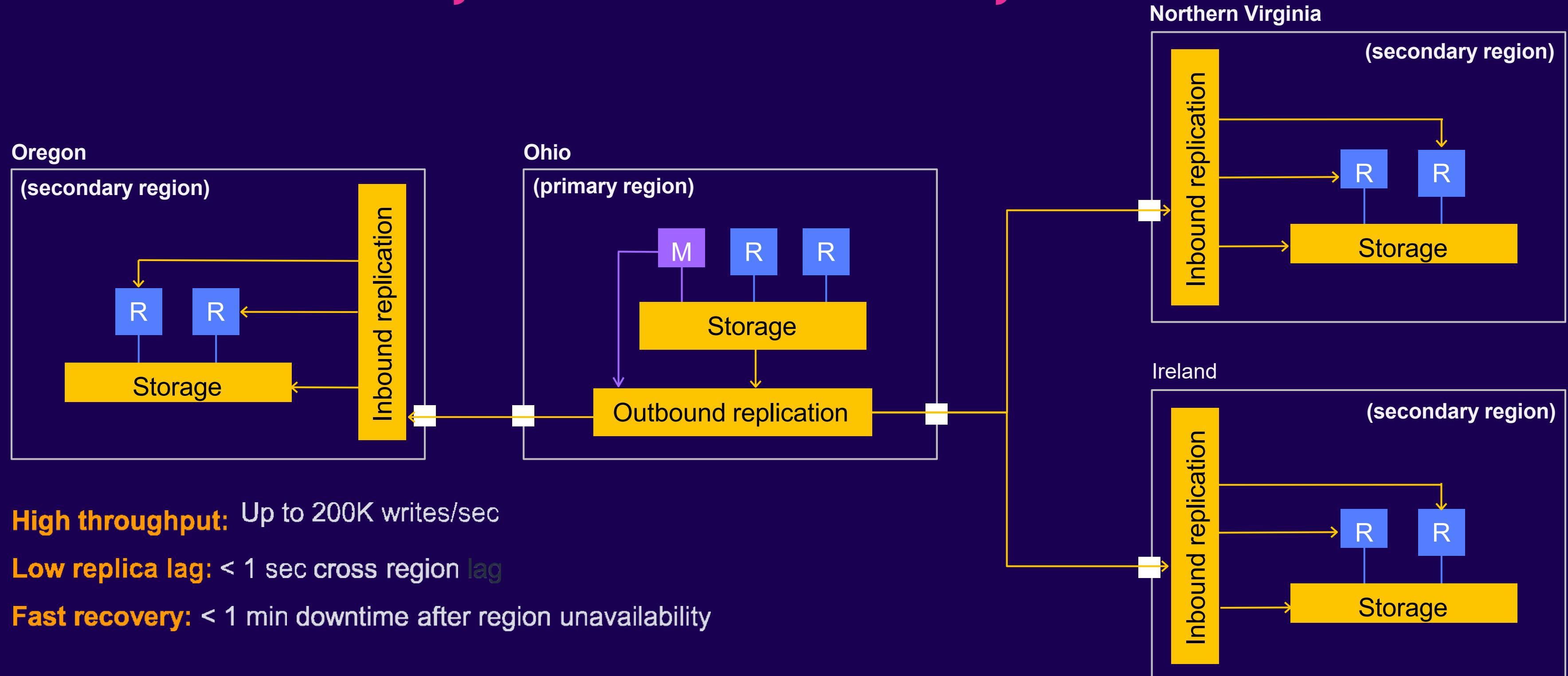
Physical replication across regions (Aurora Global Database)

Logical (e.g. binlog) replication to any database



Aurora Global Database

Faster disaster recovery and enhanced data locality



AWS DynamoDB

Why NoSQL?

SQL

NoSQL

Optimized for storage

Normalized/relational

Ad hoc queries

Scale vertically

Good for OLAP

Optimized for compute

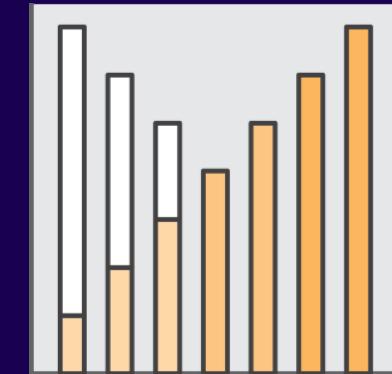
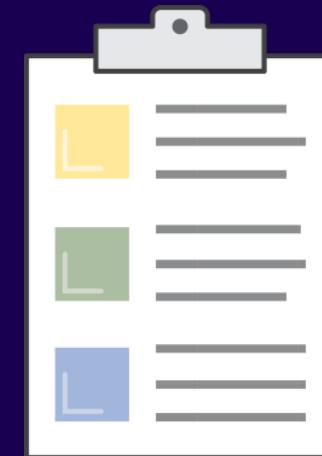
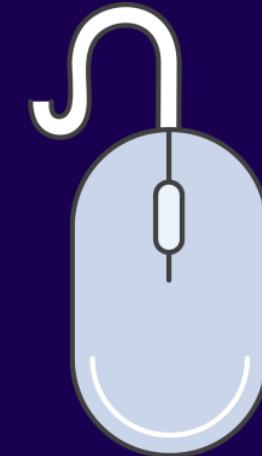
De-normalized/hierarchical

Instantiated views

Scale horizontally

Built for OLTP at scale

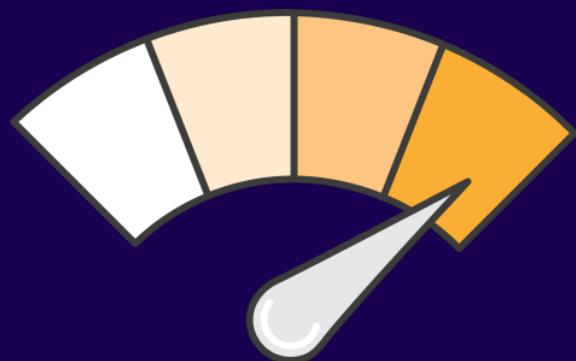
Amazon DynamoDB



Fully managed NoSQL

Document or Wide Column

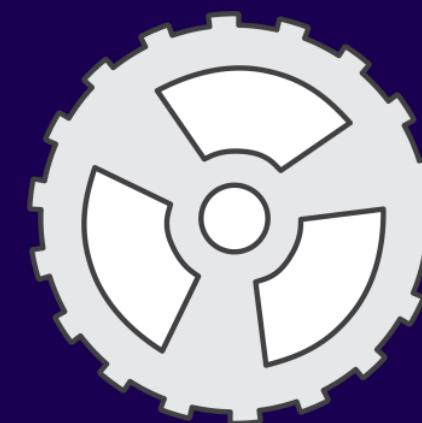
Scales to any workload



Fast and consistent



Access control



Event-driven programming

What is a Key / Value store?

A **key-value database** is a type of non-relational database (NoSQL) that uses a simple key-value method to store data.

A key/value stores a **unique key** alongside a value

Key	Value
Data	1010101000101011001010010101001
Worf	0110101100010101010101011100010
Ro Laren	001010100101011001010101010101010

Key values stores are **dumb and fast**. They generally lack features like:

- Relationships
- Indexes
- Aggregation

A key/value store can resemble tabular data, it does not have to have the consistent columns per row (hence its schema less)

Key	Value
Data	{species: android, rank: 'Lt commander'}
Worf	{species: klingon, rank: 'Lt commander'}
Ro Laren	{species: bajoran, affiliation: 'maquis'}

A simple key/value store will interpret this data resembling a dictionary (aka Associative arrays or hash)

Key (Name)	Species	Rank	Affiliation
Data	andriod	Lt commander	
Worf	klingon	Lt commander	
Ro Laren	bajoran		maquis

Due to their simple design they can scale well beyond a relational database

Table



Mandatory
Key-value access pattern
Determines data distribution

Optional
Model 1:N relationships
Enables rich query capabilities

All items for key
==, <, >, >=, <=
“begins with”
“between”
“contains”
“in”
sorted results
counts
top/bottom N values

Partition overloading

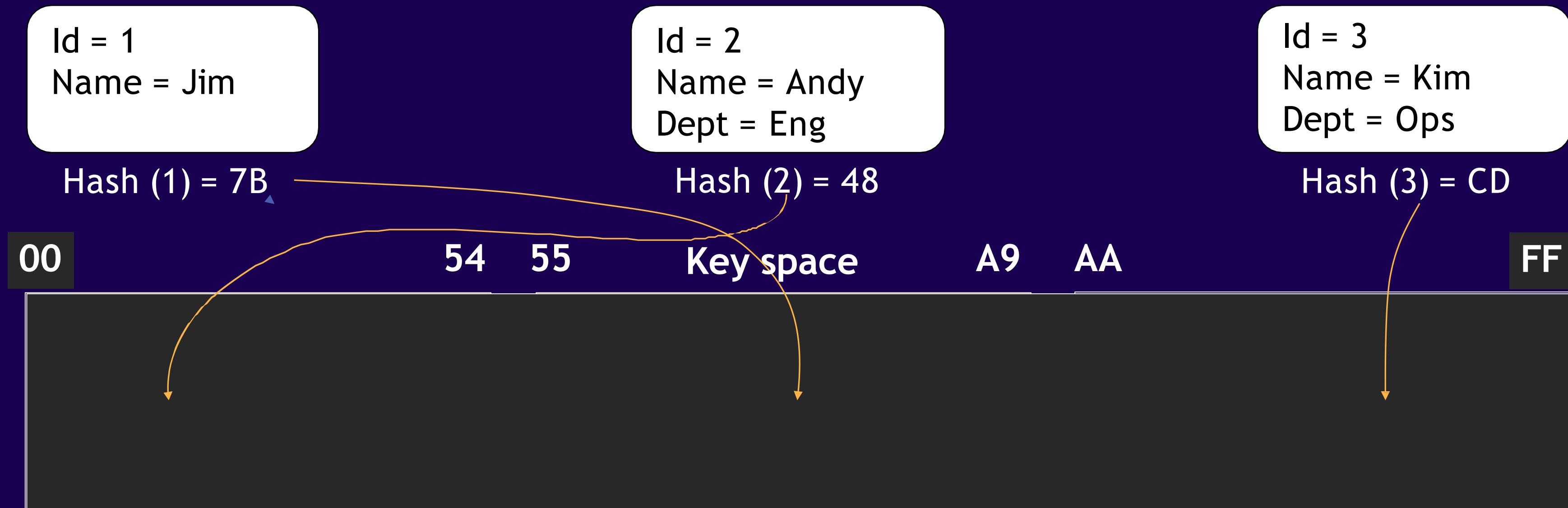
Use generic keys to facilitate heterogeneous partitions

Primary Key		Attributes			
PK	SK	Source	Location	URL	CustomerType
Customer_1	2019-11-29T08:31:28Z#O1	Online	US	www.amazon.com	Regular
	2019-11-29T08:31:28Z#O1#I1	ASIN	Status	Product	FCCID
	Customer_1	B07G6CQQYHG	PROCESSING	BOOM 3	JNzs00170
Customer_1		Login	Email	Name	Address
Customer_1		jdoe	john@example.com	John Doe	123 5th Street, New York, NY

SELECT * WHERE PK=Customer_1 AND SK > 2019-10-29

Partition/shard keys in NoSQL

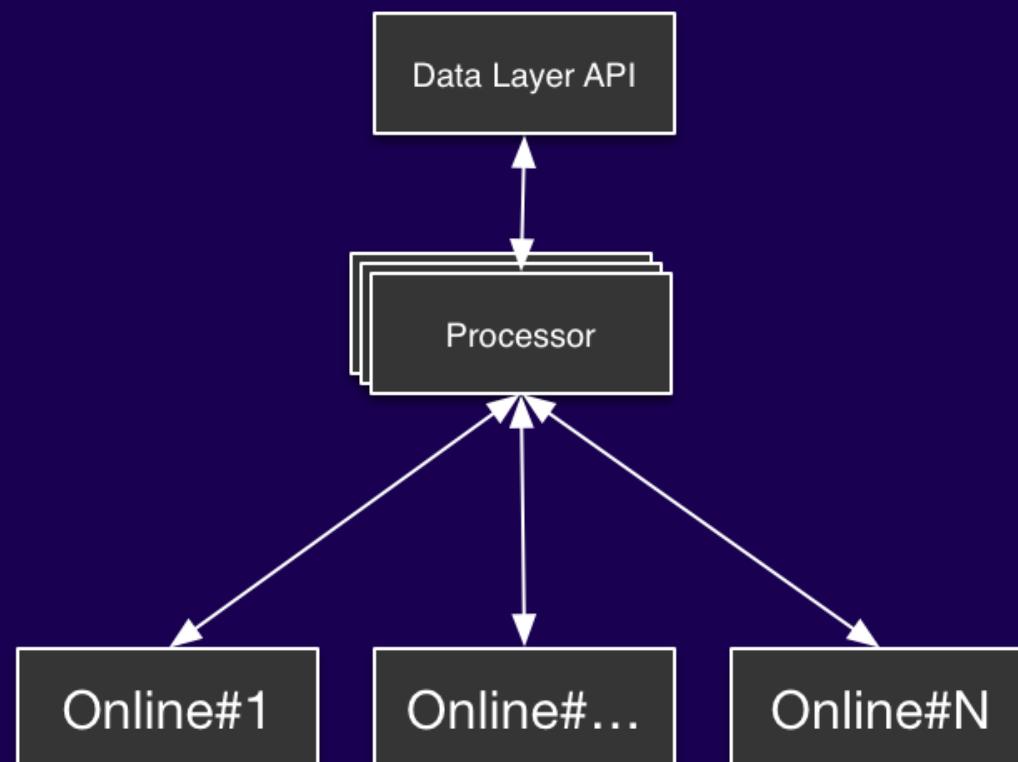
Partition/shard key is used for building an unordered hash index
Allows table to be partitioned for scale



Write sharding

Salt indexed keys to support high-density aggregations on GSIs

Primary Key		Attributes			
PK	SK	Source	Location	Store	CustomerType
Customer_1	2019-11-29T08:31:28Z#O1	Online#(0-N)	US	www.amazon.com	Regular
	2019-11-29T08:31:28Z#O1#I1	ASIN	Status	Product	FCCID
	Customer_1	B07G6CQQY#(0-N)	PROCESSING	BOOM 3	JNZS00170
		Login	Email	Name	Address
		jdoe	john@example.com	John Doe	123 5th Street, New York, NY



- Abstract partitioning from clients behind an API
- Write across many partitions
- Use parallel processes to increase read throughput

Index overloading

SELECT * WHERE PK=ONLINE#0 AND SK=US

...

SELECT * WHERE PK=ONLINE#N AND SK=US

Primary Key		Attributes			
GSI1PK	GSI1SK	PK	SK	Store	CustomerType
Online#(0-N)	US	Customer_1	2019-11-29T08:31:28Z#O1	www.amazon.com	Regular
B07G6CQQY#(0-N)	PROCESSING	Customer_1	2019-11-29T08:31:28Z#O1#I1	BOOM 3	JNZS00170

SELECT * WHERE PK=B07G6CQQY#0 AND SK=PROCESSING

...

SELECT * WHERE PK=B07G6CQQY#N AND SK=PROCESSING

Use generic keys once more to use indexes for multiple access patterns

AWS DocumentDB

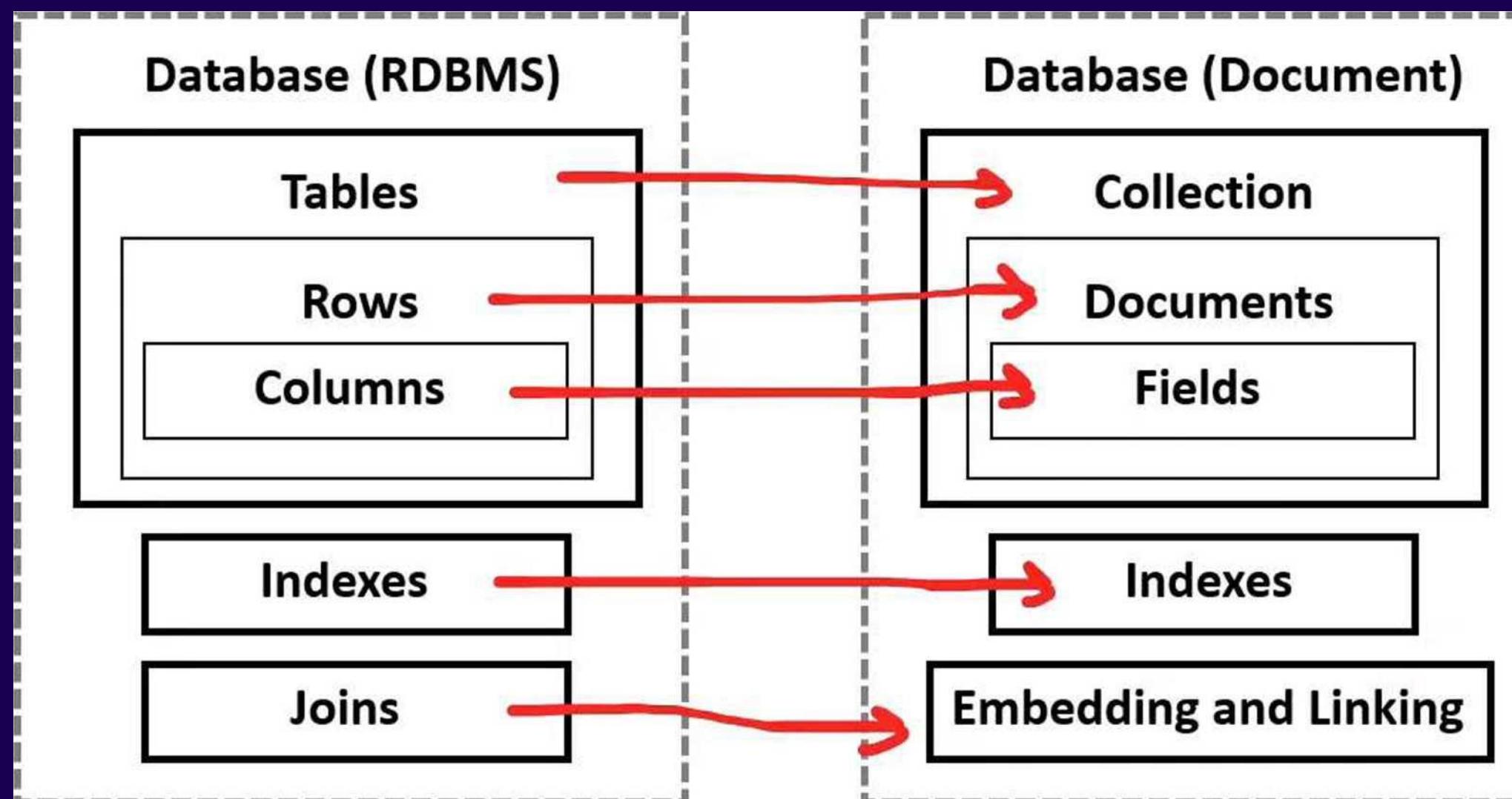
What is a Document store?

A **document store** is a NOSQL database that stores **documents** as its primary data structure.

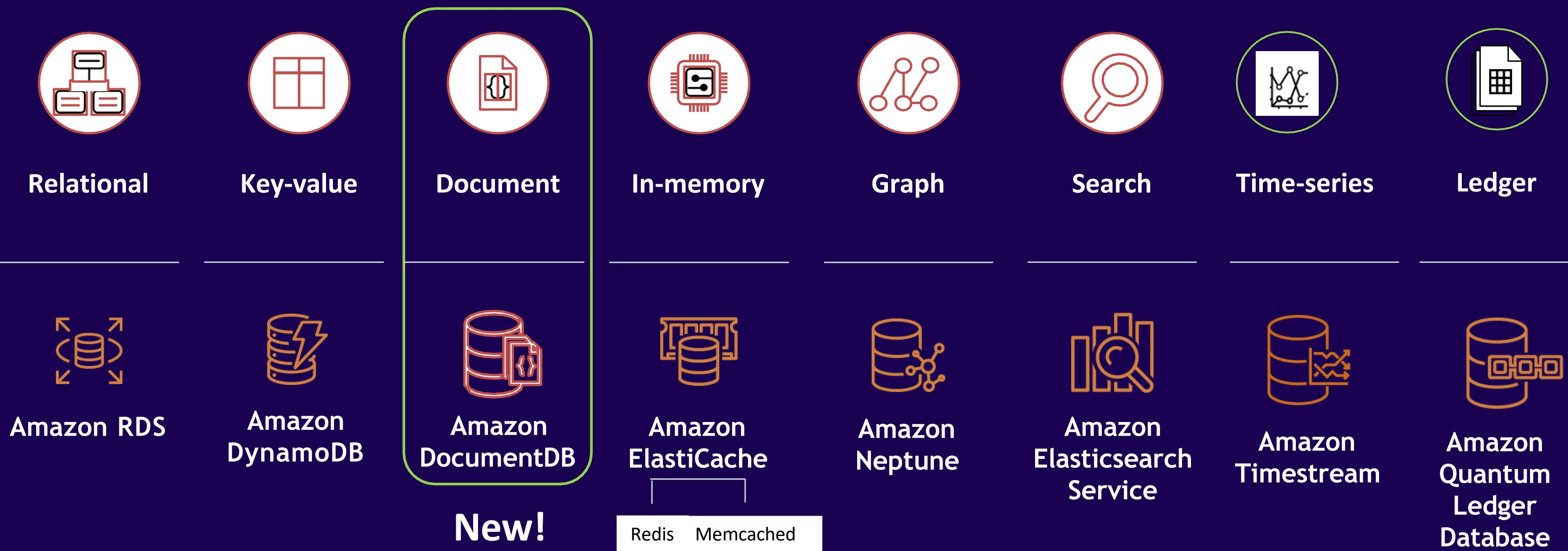
A document could be an XML but more commonly is JSON or JSON-Like

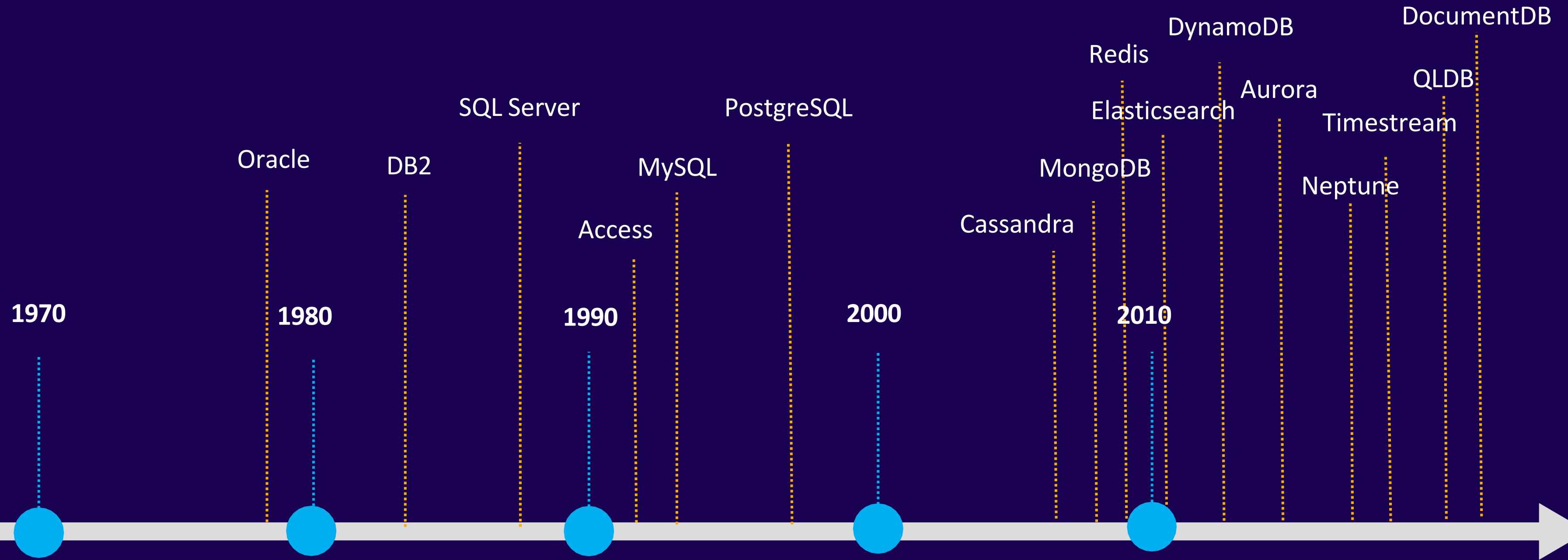
Document stores are sub-class of Key/Value stores

The components of a document store compared to Relational database



AWS: Purpose-built databases





Use cases for document data



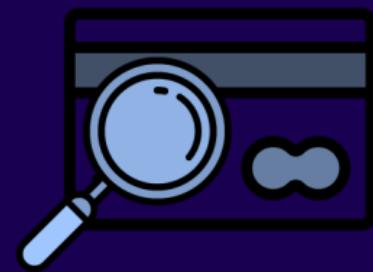
Content
management



Mobile



Personalization



Catalog



Retail and
marketing



User profiles

Amazon DocumentDB

Fast, scalable, and fully managed MongoDB-compatible database service

Fast



Millions of requests per second with millisecond latency; twice the throughput of MongoDB

Scalable



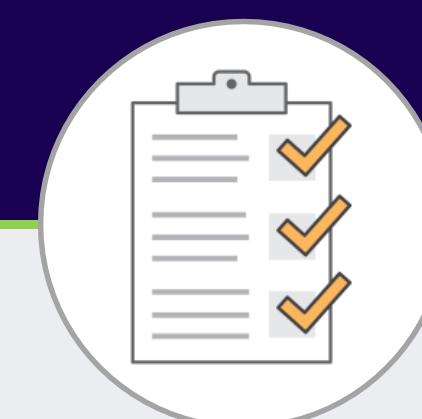
Separation of compute and storage enables both layers to scale independently; scale out to 15 read replicas in minutes

Fully managed



Managed by AWS: no hardware provisioning; auto patching, quick setup, secure, and automatic backups

MongoDB compatible



Compatible with MongoDB 3.6; use the same SDKs, tools, and applications with Amazon DocumentDB

Amazon DocumentDB

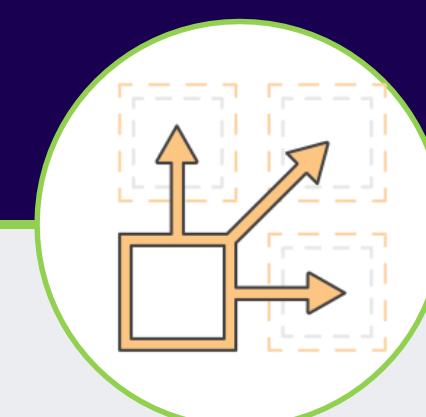
Fast, scalable, and fully managed MongoDB-compatible database service

Fast



Millions of requests per second with millisecond latency; twice the throughput of MongoDB

Scalable



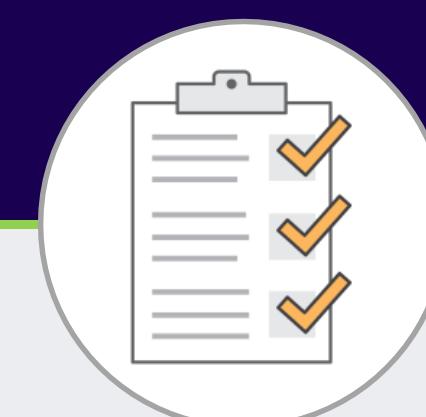
Separation of compute and storage enables both layers to scale independently; scale out to 15 read replicas in minutes

Fully managed



Managed by AWS: no hardware provisioning; auto patching, quick setup, secure, and automatic backups

MongoDB compatible



Compatible with MongoDB 3.6; use the same SDKs, tools, and applications with Amazon DocumentDB

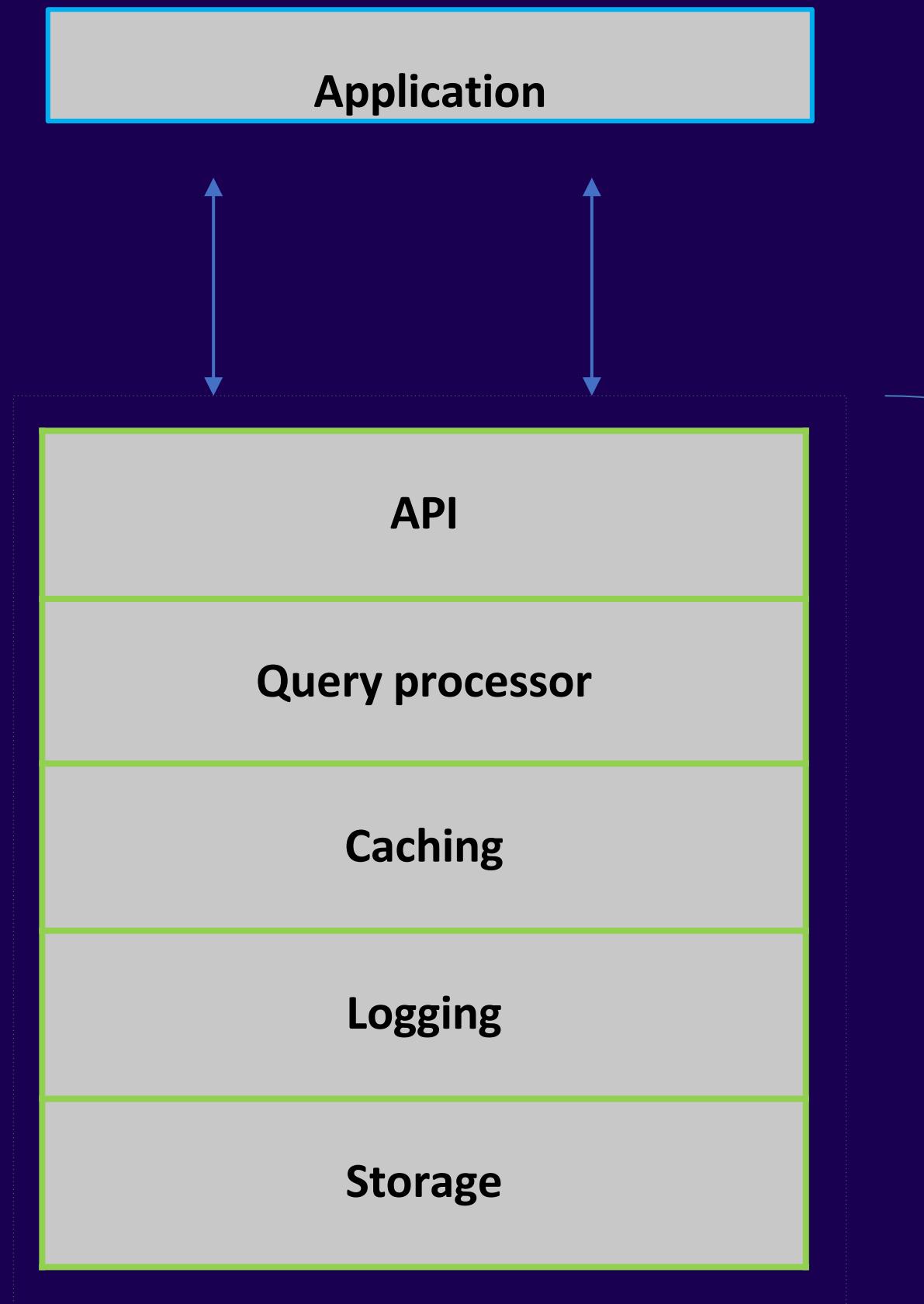
Amazon DocumentDB



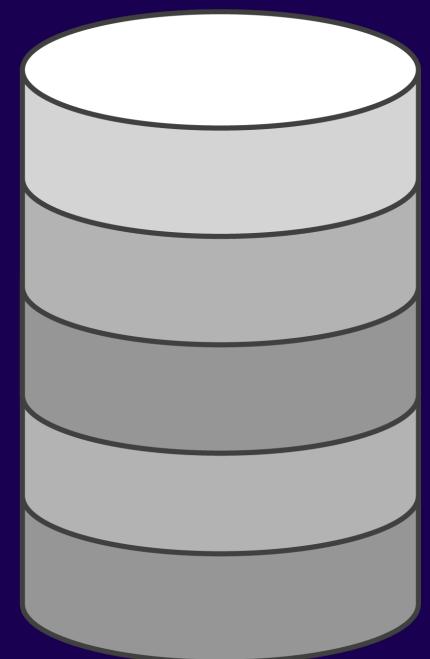
Amazon DocumentDB



Challenges with traditional database architectures



**Single monolithic
architectures**



Scale monolithically
Fail monolithically

Not Designed for the cloud

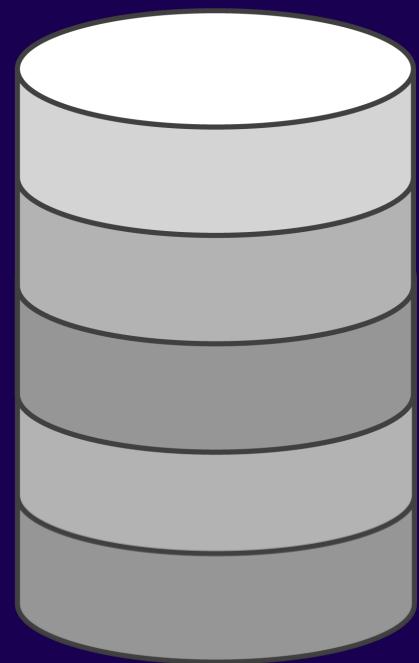


Challenges with traditional databases: Scaling

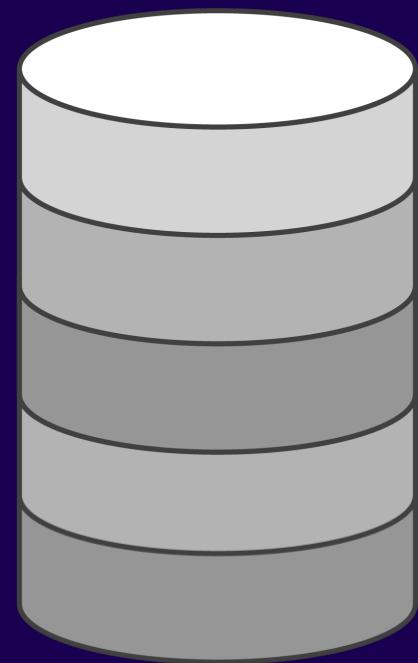
Scenario: Spike in traffic and you want to add additional read capacity quickly



Node 1



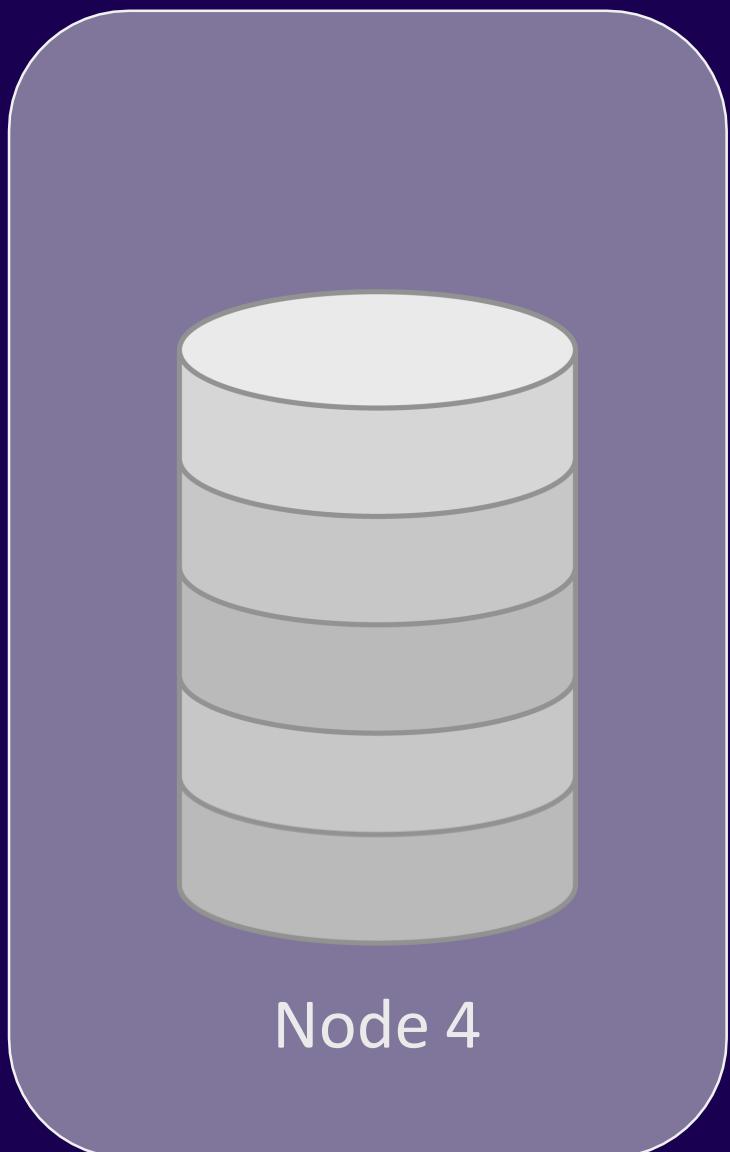
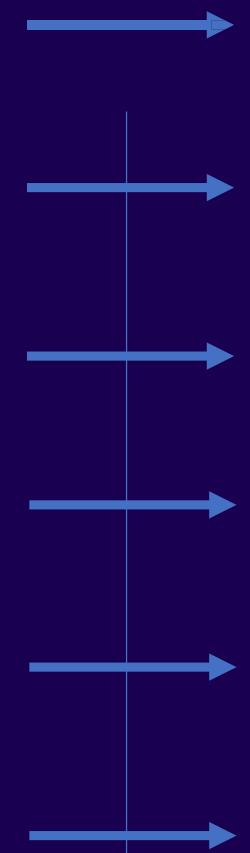
Node 2



Node 3



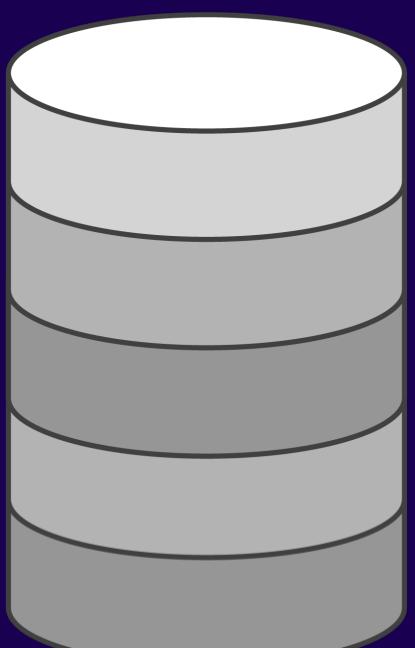
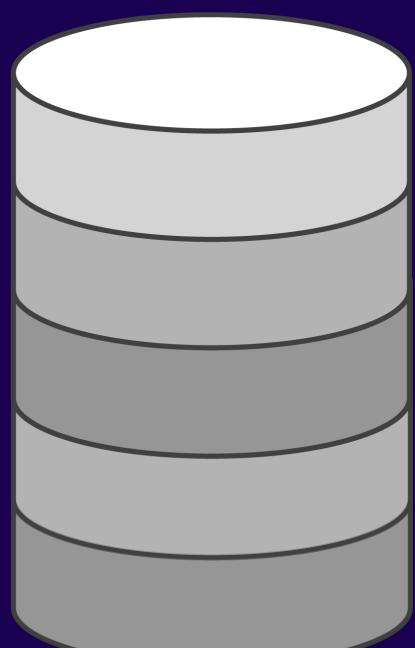
Replication



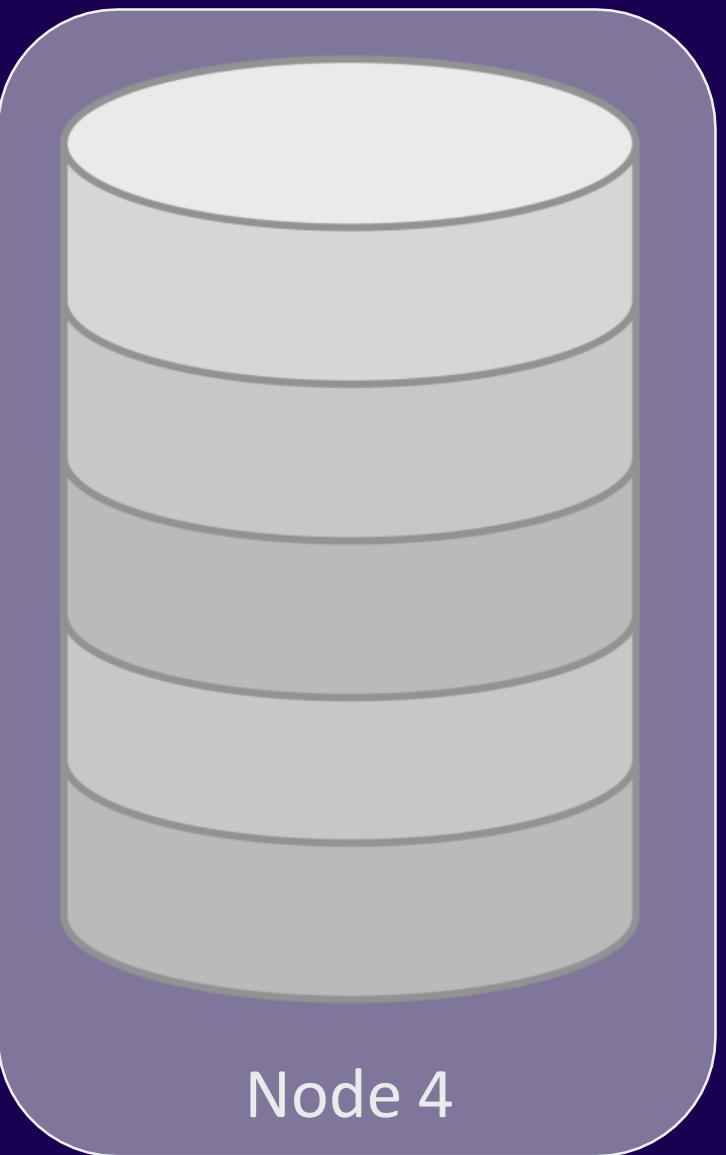
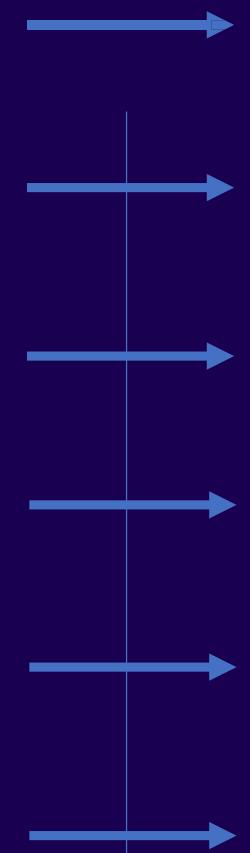
Node 4

Challenges with traditional databases: Scaling

Scenario: Scale up to run large analytical workloads on a replica



Replication



Challenges with traditional databases: Recovery

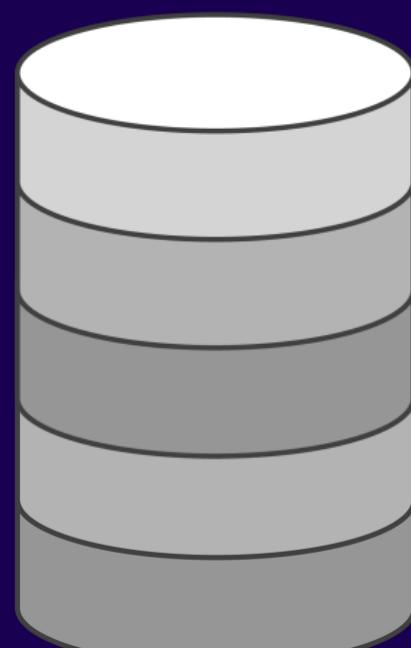
Scenario: An instance experiences a failure
and you want to recover quickly



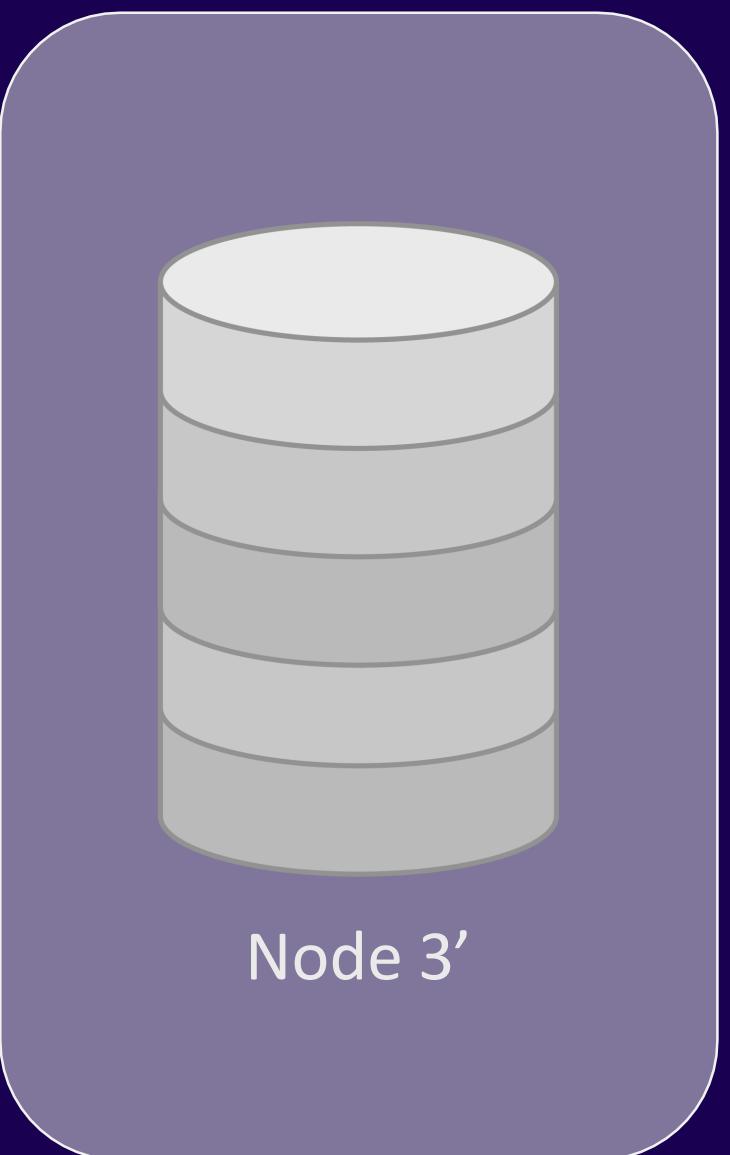
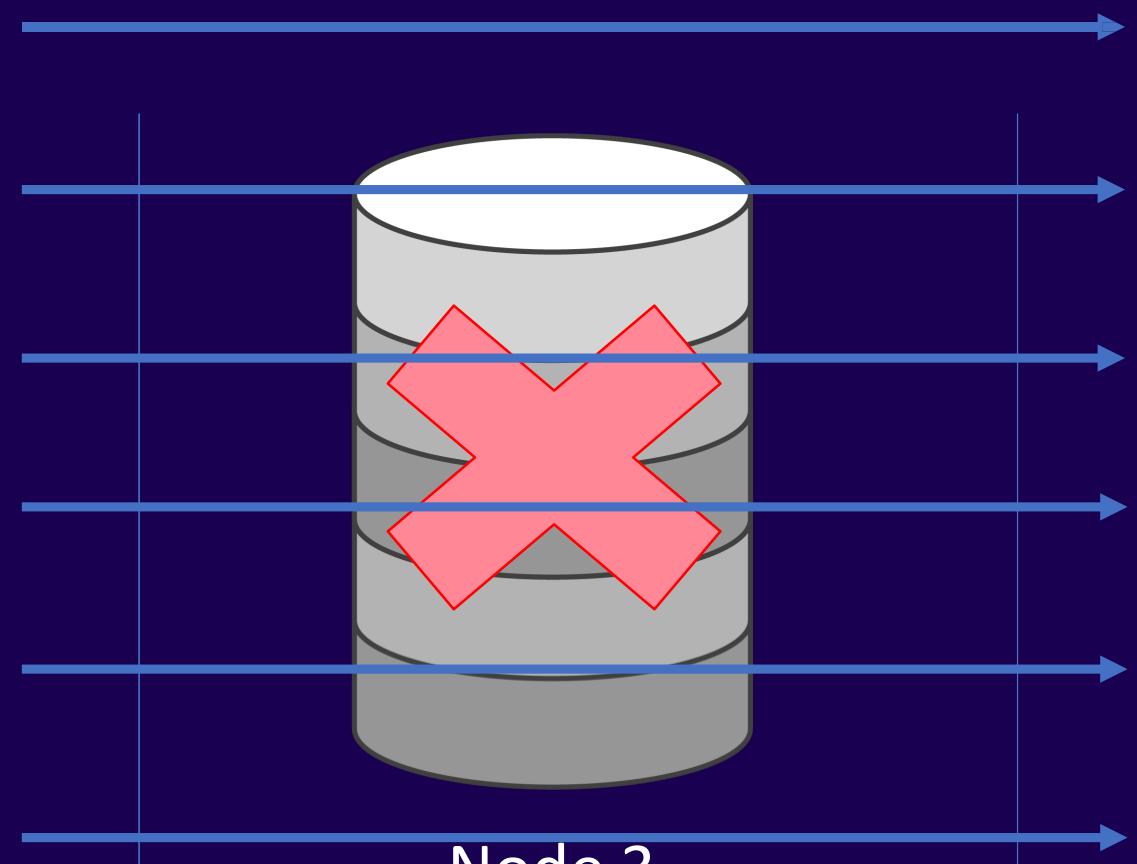
Replication



Node 1



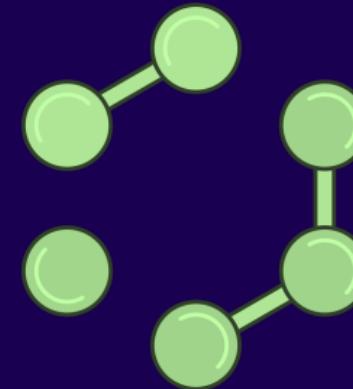
Node 2



Amazon DocumentDB: Modern cloud-native architecture

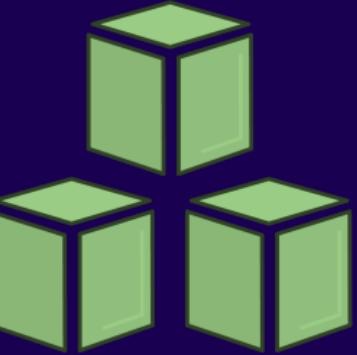
What would you do to improve scalability and availability?

1



Decouple
compute and
storage

2



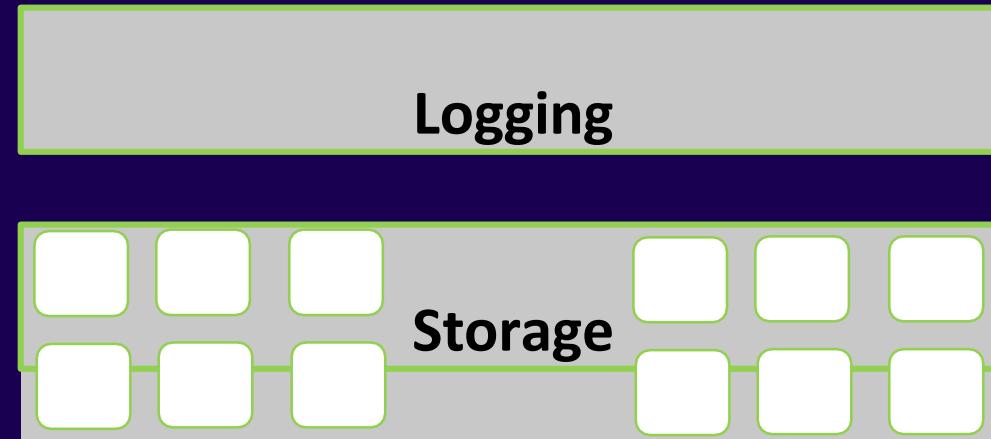
Distribute data in
smaller partitions

3



Increase the
replication of
data (6x)

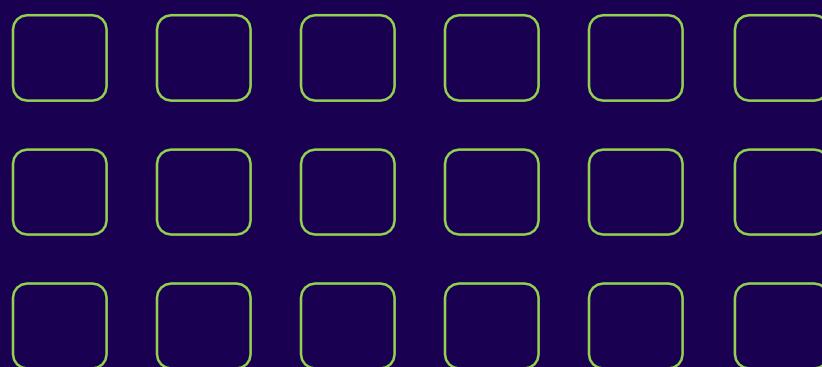
Amazon DocumentDB: Modern cloud-native architecture



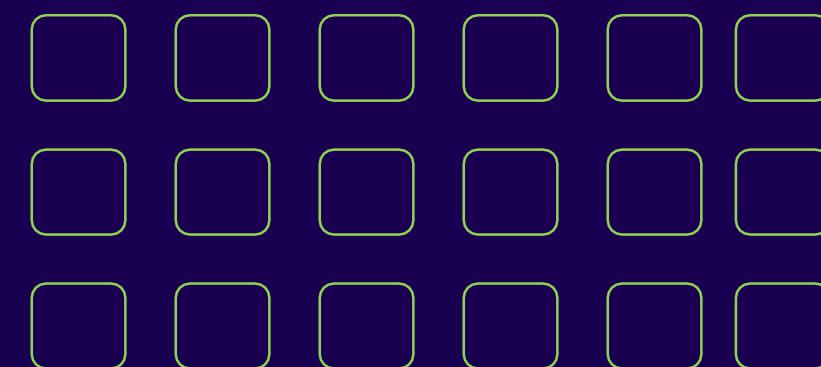
2

Distribute data in smaller partitions

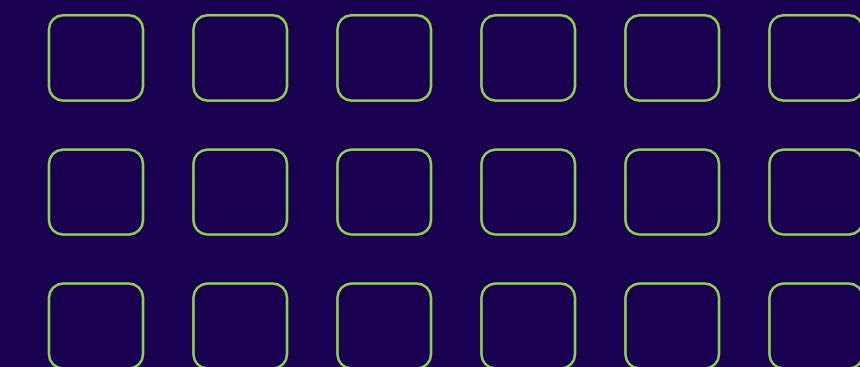
Distributed storage volume



AZ1

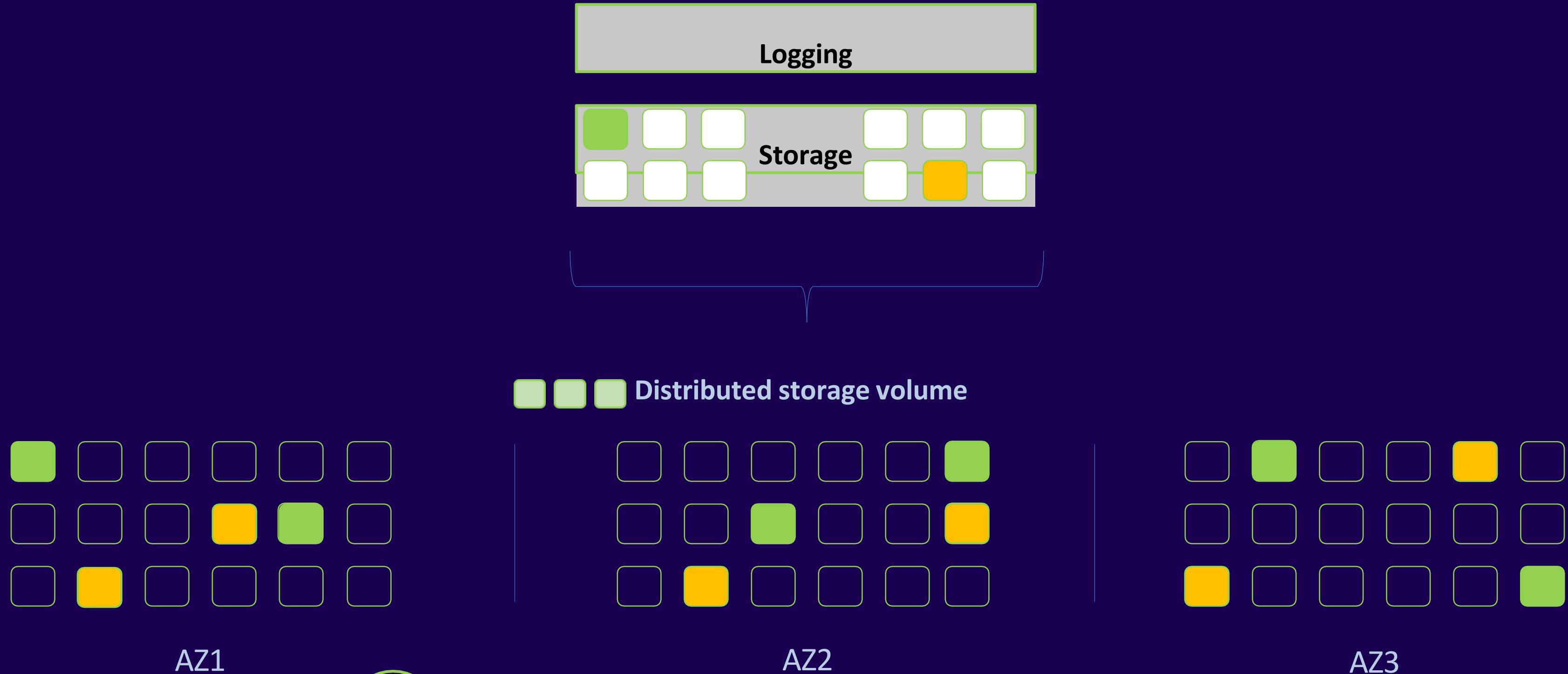


AZ2



AZ3

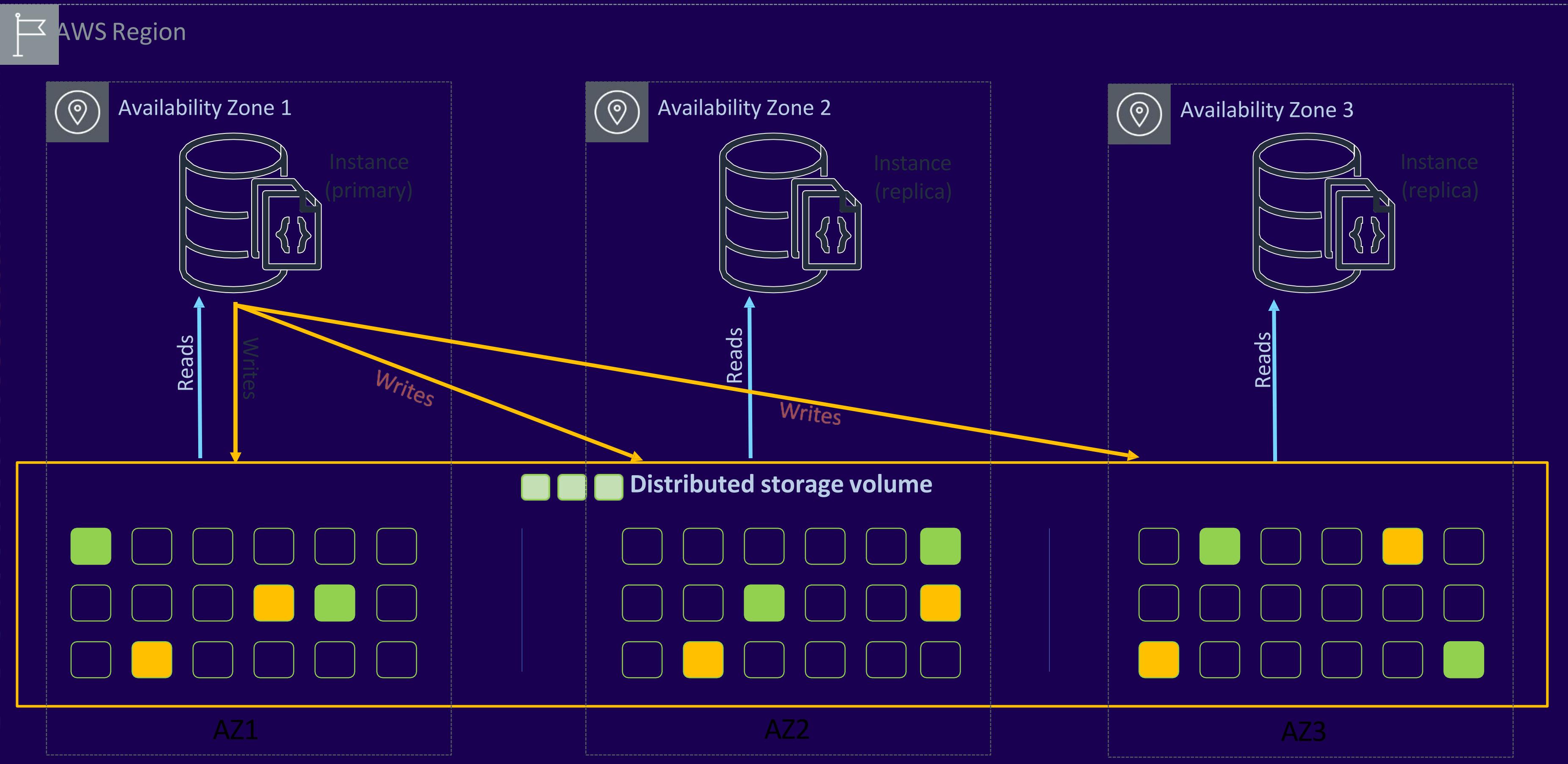
Amazon DocumentDB: Modern cloud-native architecture



3

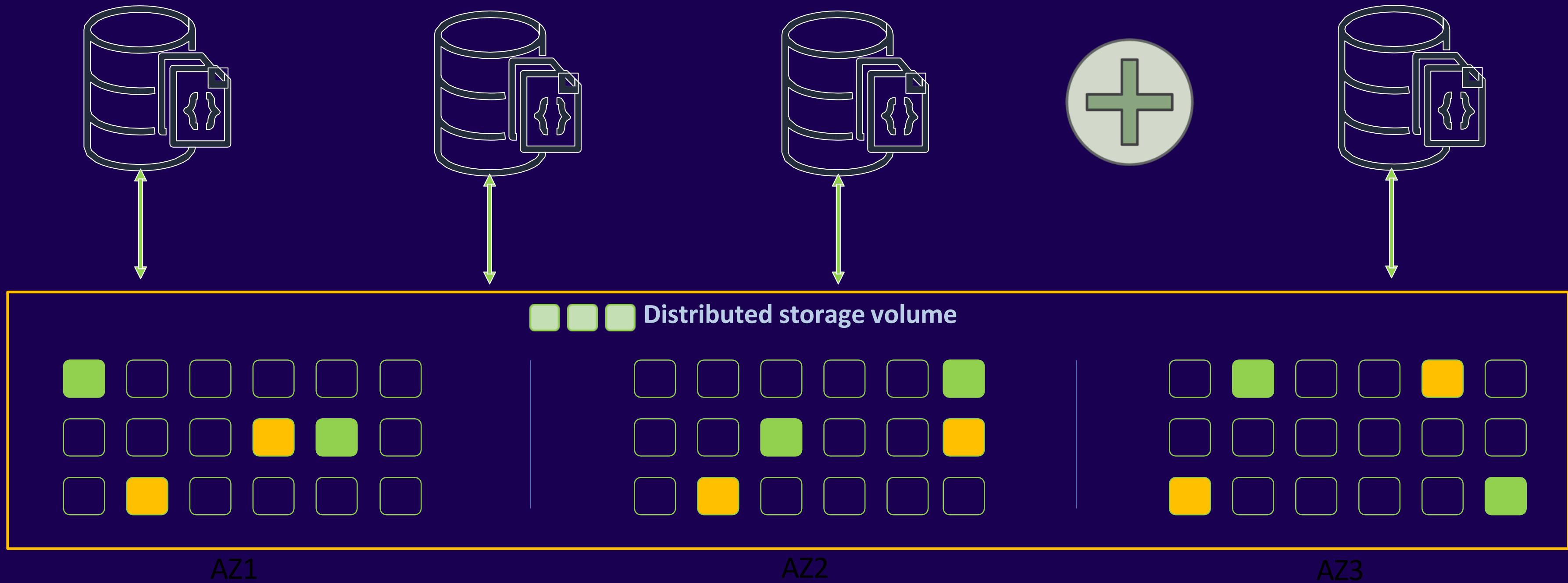
Increase the replication of data (6x)

Amazon DocumentDB: Modern cloud-native architecture



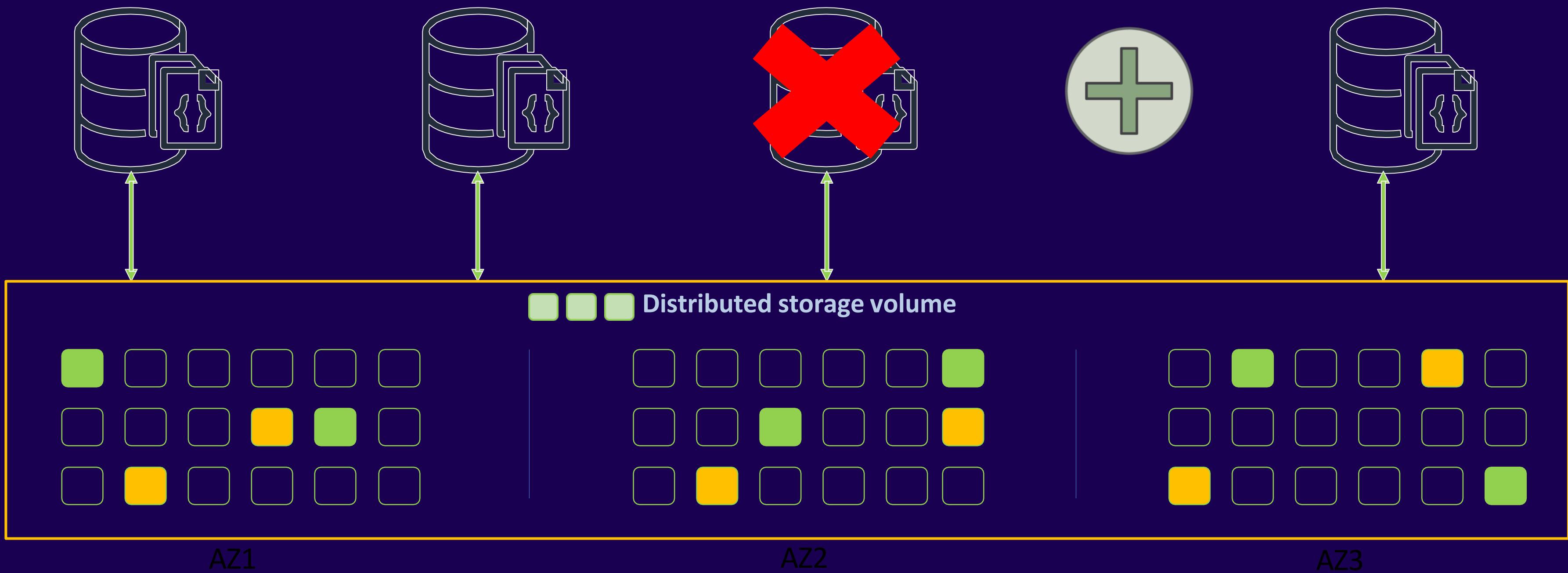
Amazon DocumentDB: Scaling

Scenario: A spike in traffic and you want to add additional read capacity quickly



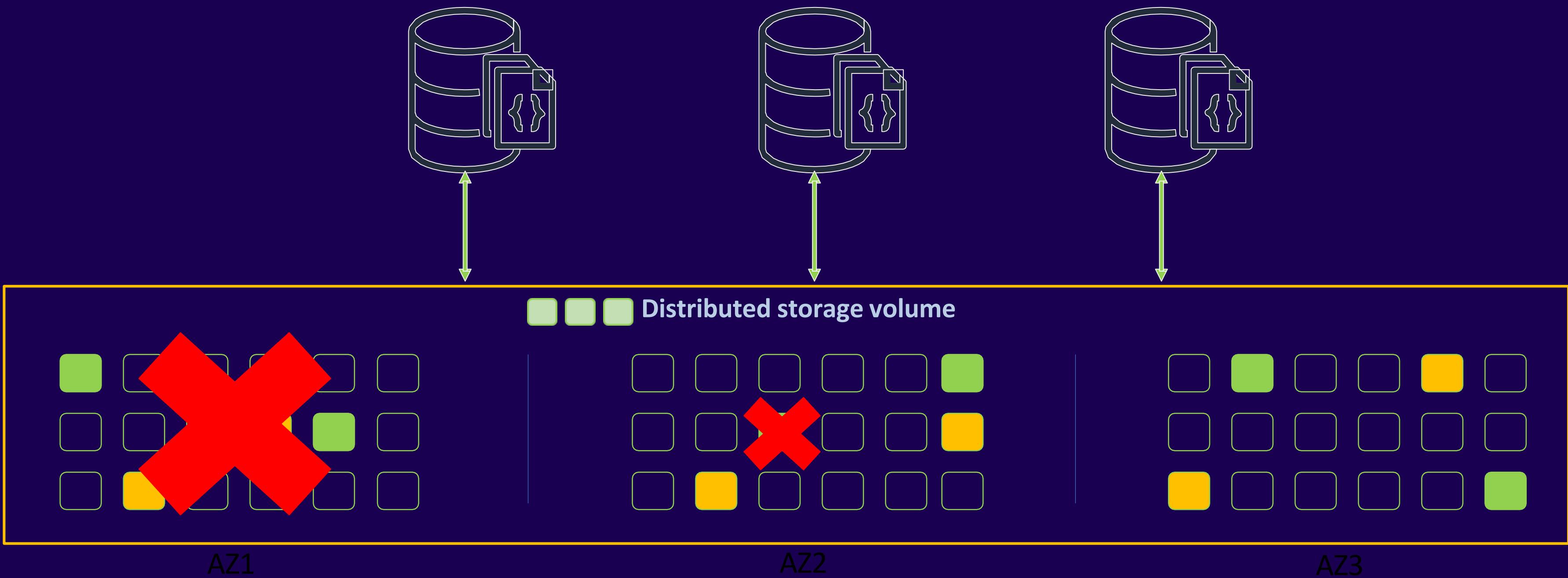
Amazon DocumentDB: Failure recovery

Scenario: An instance experienced a failure and you want to recover quickly



Amazon DocumentDB: Failure recovery

Scenario: Six-way replication across three Availability Zones provides the ability to handle AZ + 1 failures



Fast

Fast, scalable, and fully managed MongoDB-compatible database service

Fast



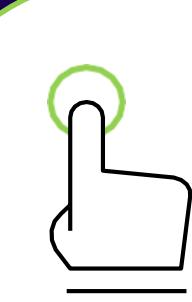
Millions of requests per second with millisecond latency

More throughput



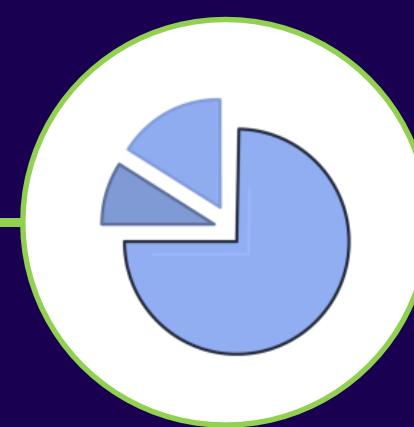
Separation of storage and compute layers offloads replication to the storage volume so that your instances can do more work; twice the throughput of MongoDB

Optimizations



Database engine optimizations to reduce the number of IOs and minimize network packets in order to offload the database engine

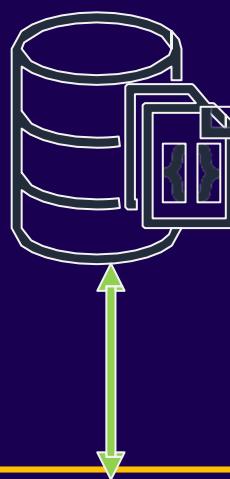
Flexible



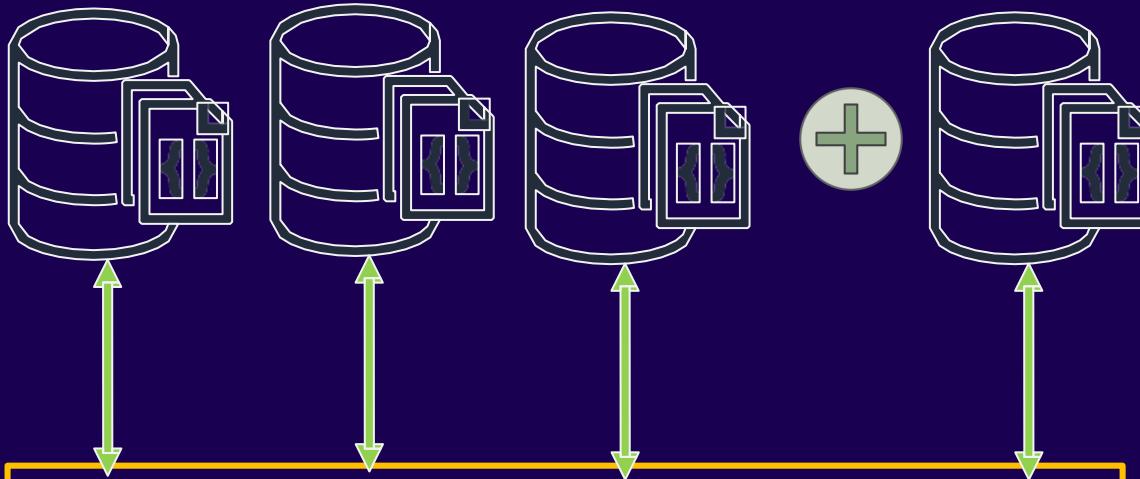
Scale up an instance in minutes for analytical queries and scale down at the end of the day

Flexible

Durability and replication are handled by the distributed storage volume



Distributed Storage Volume



Distributed Storage Volume

Scenario 1: Dev/test with a single instance



Distributed Storage Volume

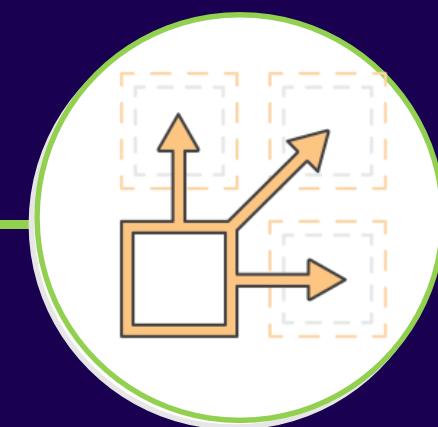
Scenario 2: Read scaling in minutes

Scenario 3: Scale-up and scale-out for analytics

Scalable

Fast, **scalable**, and fully managed MongoDB-compatible database service

Scale out
in minutes



Scale out read capacity by adding additional replicas (up to 15 replicas); adding replicas takes minutes regardless of data size

Scale up
in minutes



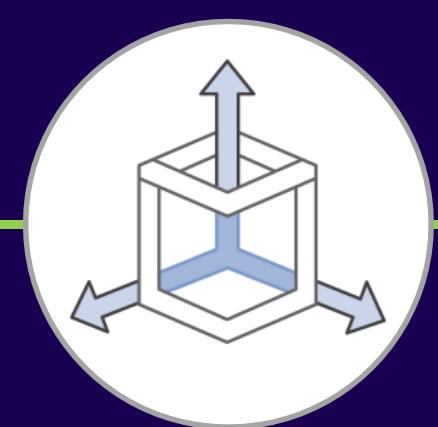
Scale up and down instances in minutes (15.25 GiB memory to 244 GiB memory)

Storage scales automatically



Storage volumes automatically grow from 10 GB to 64 TB without any user action

Load balancing



Load balancing across instances with replica sets

Fully managed

Fast, scalable, and **fully managed** MongoDB-compatible database service

Pay-as-you-go
pricing; enterprise
grade



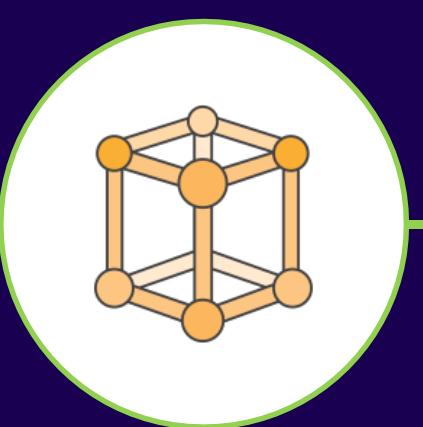
On-demand,
pay-as-you-go pricing
enables you to pay only
for the resources that you
need and only when you
use them

Automatic failure
recover and
failover



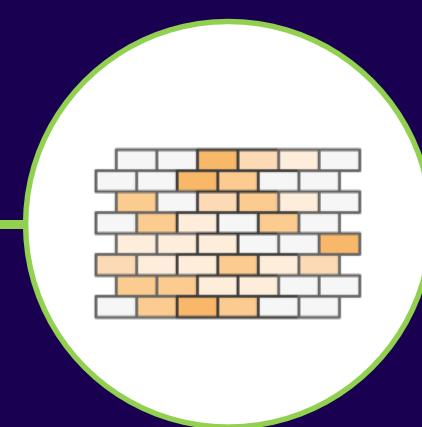
Replicas are
automatically promoted
to primary; failing
processes are
automatically detected
and recovered; no cache
warmup needed

Point-in-time
recovery



Automated backups are
stored in Amazon S3,
which is designed for
99.99999999%
durability

Durable, fault-
tolerant and self-
healing storage



Data at rest is
replicated six ways
across three AZs;
handle AZ + 1 failures

Fully managed

Fast, scalable, and **fully managed** MongoDB-compatible database service

Automatic patching



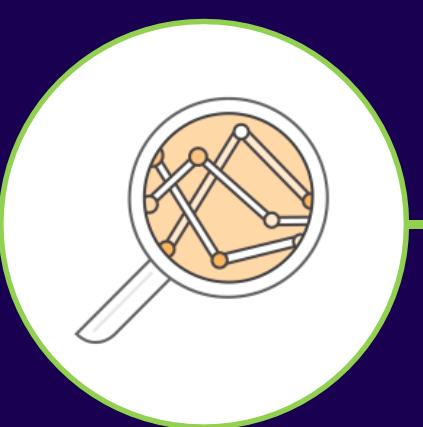
Up to date with the latest patches

AWS Support



AWS Support provides people, technology, and programs to help you achieve success

Monitoring



More than 20 key operational metrics for your clusters at no extra charge

Integrated



Deeply integrated with AWS services

MongoDB compatible

Fast, scalable, and fully managed **MongoDB-compatible** database

MongoDB 3.6



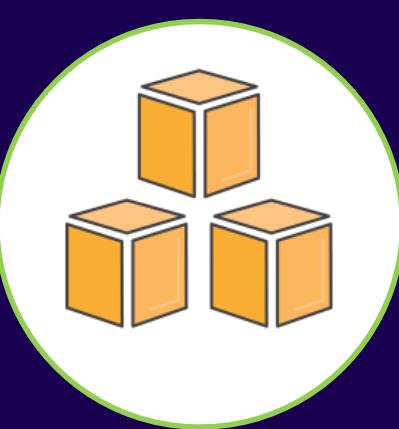
Compatible with MongoDB
Community Edition 3.6

Same drivers, tools



Use the same MongoDB
drivers and tools with
Amazon DocumentDB;
as simple as changing an
application connection string

Replica sets



Read scaling is easy with
automatic replica set
configurations

Migration with AWS DMS



Live migrations with
AWS DMS;
free for 6 months

Migration

Migrate to Amazon DocumentDB with Amazon Database Migration Service (AWS DMS)

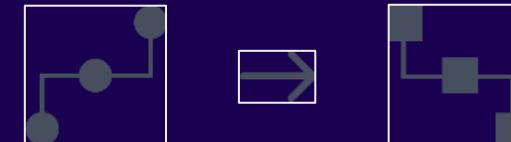
AWS DMS is free to use for 6 months if you are moving to Amazon DocumentDB



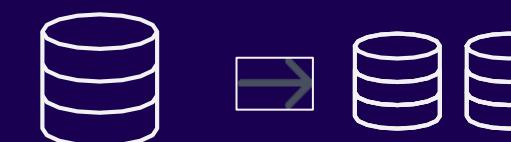
Migrate between on-premises MongoDB and Amazon DocumentDB



Migrate self-hosted MongoDB databases to Amazon DocumentDB



Data replication for virtually no downtime

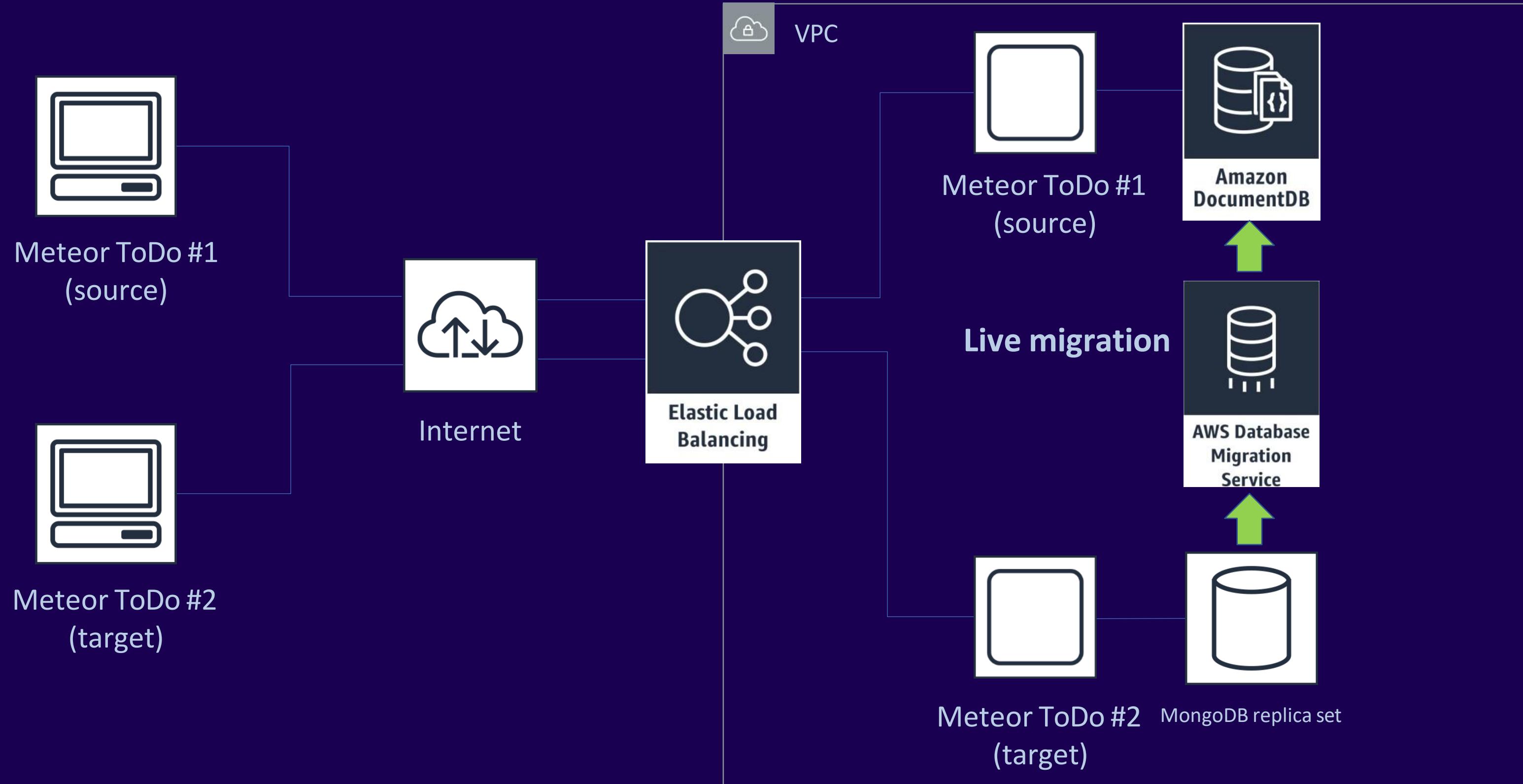


Migrate from replica sets and sharded clusters

DMS: 100,000+ Databases migrated

AWS DMS demo architecture

Migrate from MongoDB to Amazon DocumentDB with AWS DMS



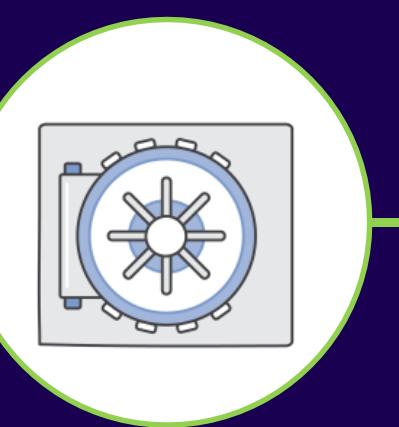
Security and compliance

Amazon VPC



Strict network isolation with Amazon Virtual Private Cloud (VPC)

Encryption by default



Encryption at rest with AWS KMS and customer-managed AWS keys; encryption in transit with TLS

Safe defaults



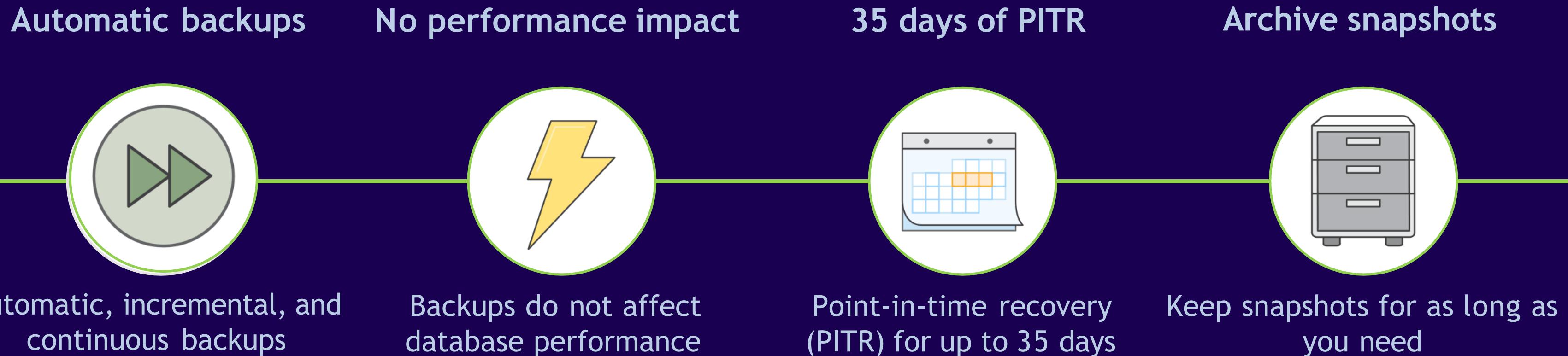
Best practices are the defaults

Compliance

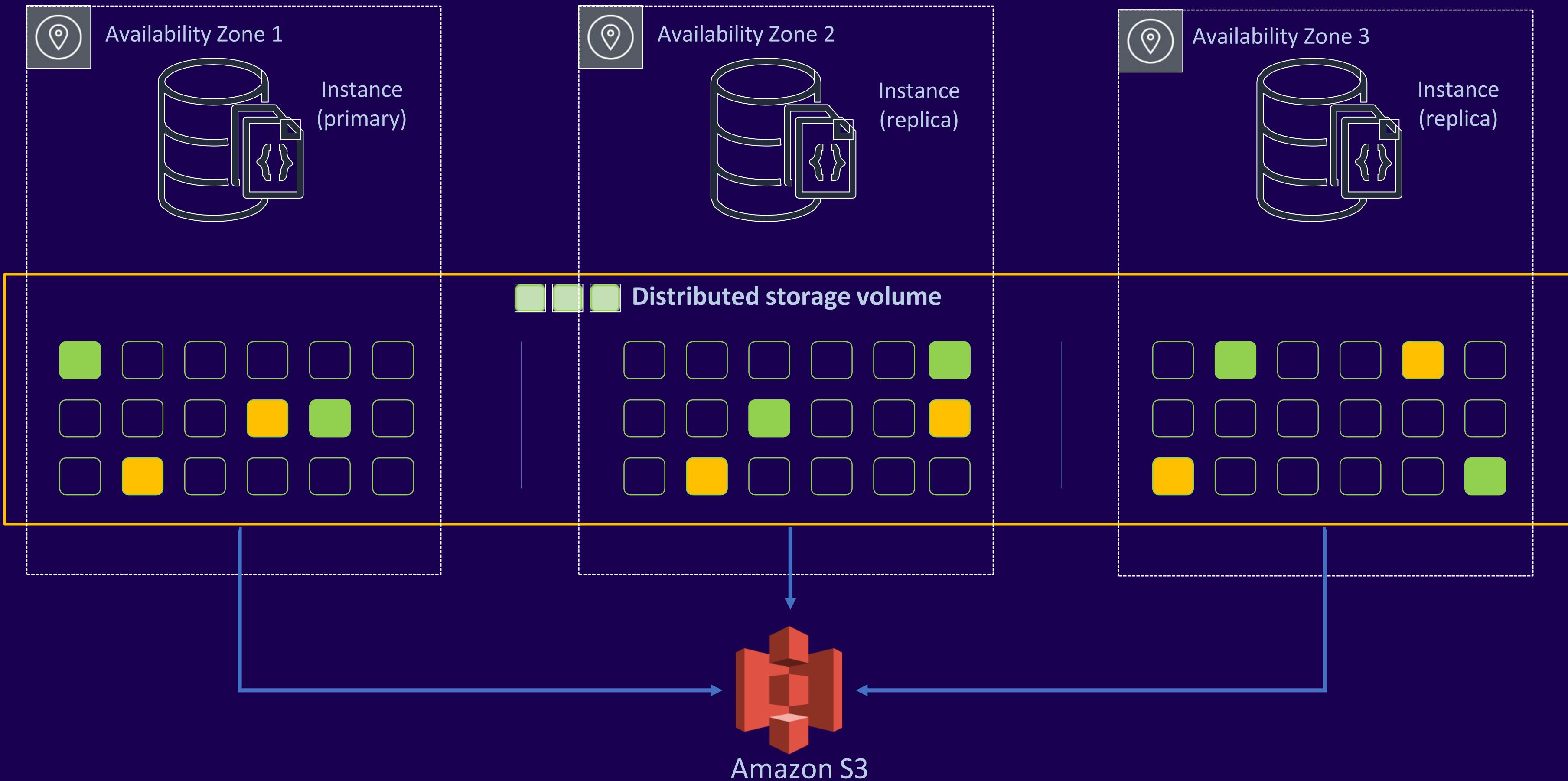


Amazon DocumentDB has been assessed to comply with PCI DSS, ISO 9001, 27001, 27017, and 27018, SOC 2, in addition to being HIPAA eligible

Backup



Amazon DocumentDB: Backups streamed to S3



AWS Aurora vs RDS

Feature	Amazon Aurora	Amazon RDS
Database Engines	MySQL, PostgreSQL	MySQL, PostgreSQL, MariaDB, SQL Server, Oracle
Performance	Up to 5x faster than MySQL and 2x faster than PostgreSQL	Standard performance based on the chosen instance type
Storage	Automatically scales from 10 GB to 128 TB	Storage scales up to 64 TB (SQL Server: 16 TB)
Replication	Supports up to 15 read replicas	Supports up to 5 read replicas
Failover	Automatic failover to read replicas	Manual failover (unless Multi-AZ is enabled)
Availability	Highly available with 6 copies of data across 3 AZs	High availability with Multi-AZ feature
Backup	Continuous, incremental backups with no performance impact	Periodic backups with potential performance impact
Pricing	More expensive but offers better performance and resilience	Cheaper but requires more manual management

AWS Aurora vs RDS

Use Case	Best Choice
Small to medium applications	RDS
Cost-sensitive projects	RDS
Enterprise-level workloads	Aurora
Highly available applications	Aurora
Read-intensive applications	Aurora
Multi-region deployments	Aurora

Knowledge check

Which of the following services can be used to deploy NoSQL workloads?

- A. Amazon Aurora
- B. Amazon RDS
- C. Amazon DynamoDB
- D. Amazon Redshift

Knowledge check

Which of the following services can be used to deploy NoSQL workloads?

- A. ~~Amazon Aurora~~
- B. ~~Amazon RDS~~
- C. Amazon DynamoDB
- D. ~~Amazon Redshift~~

Answer: C

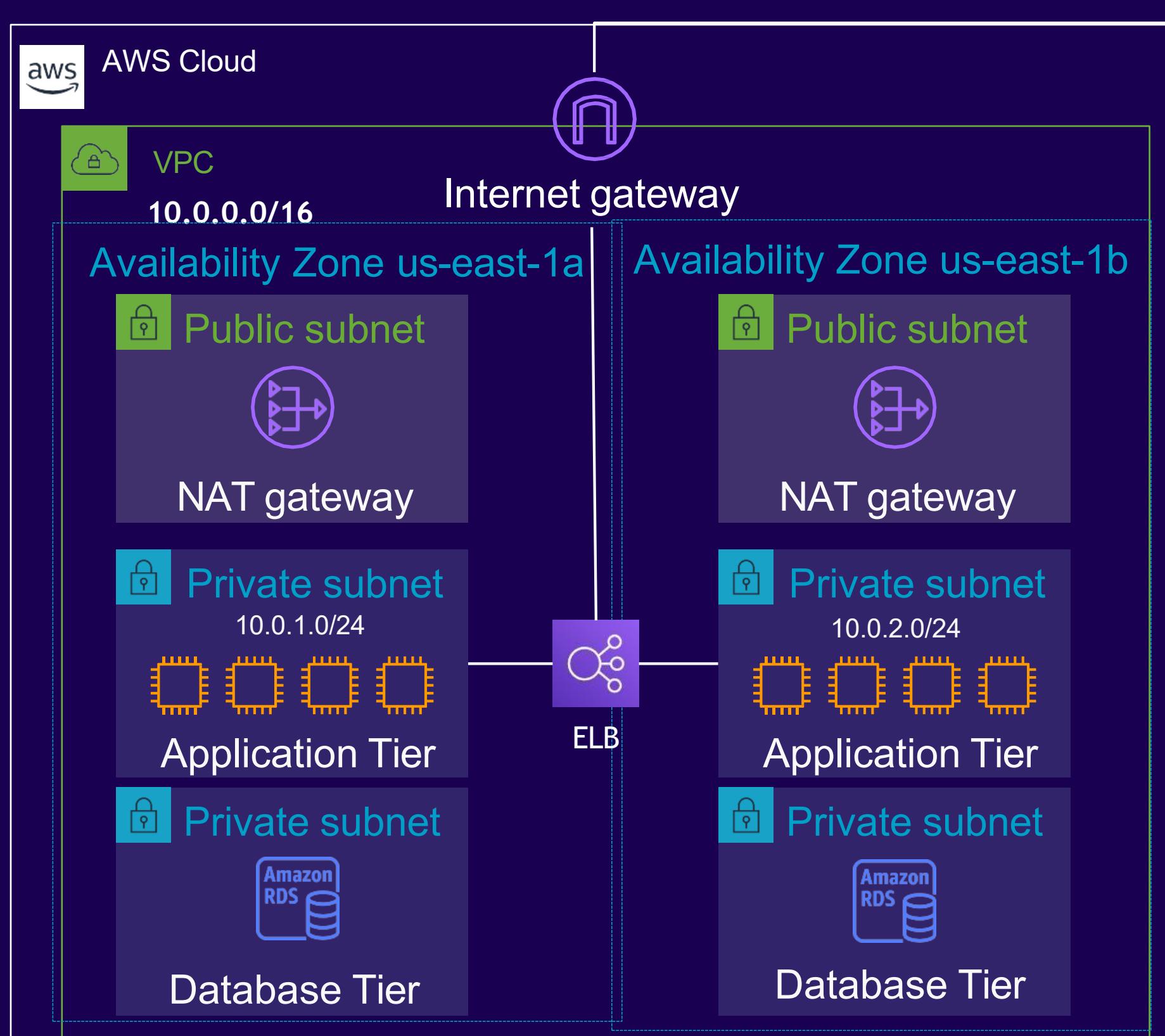
Key takeaways

AWS provides a variety of database options

- Relational (Amazon Aurora, Amazon RDS, Amazon Redshift)
- Nonrelational (Amazon DynamoDB, Amazon Neptune, Amazon DocumentDB, Amazon Keyspaces, Amazon ElastiCache, Amazon QLDB, Amazon Timestream)
- NoSQL databases are widely recognized for their ease of development, functionality, and performance at scale

Networking

Amazon Virtual Private Cloud (Amazon VPC)



Internet

VPC: Your private network space in the AWS Cloud

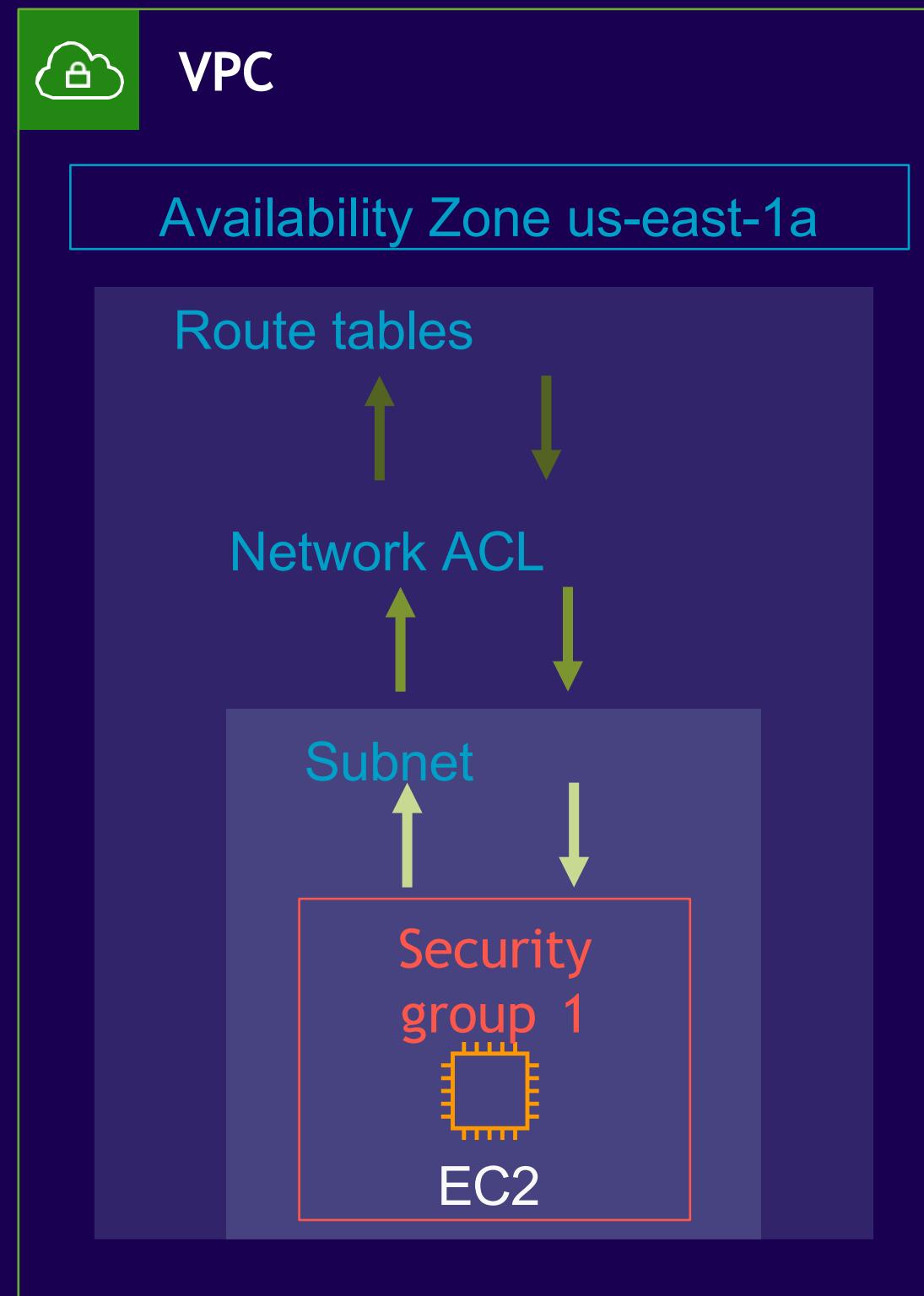
Subnets: Provide network isolation for your workloads

Public subnet: Directly accessible from the public internet

Private subnet: Not directly accessible from the public internet

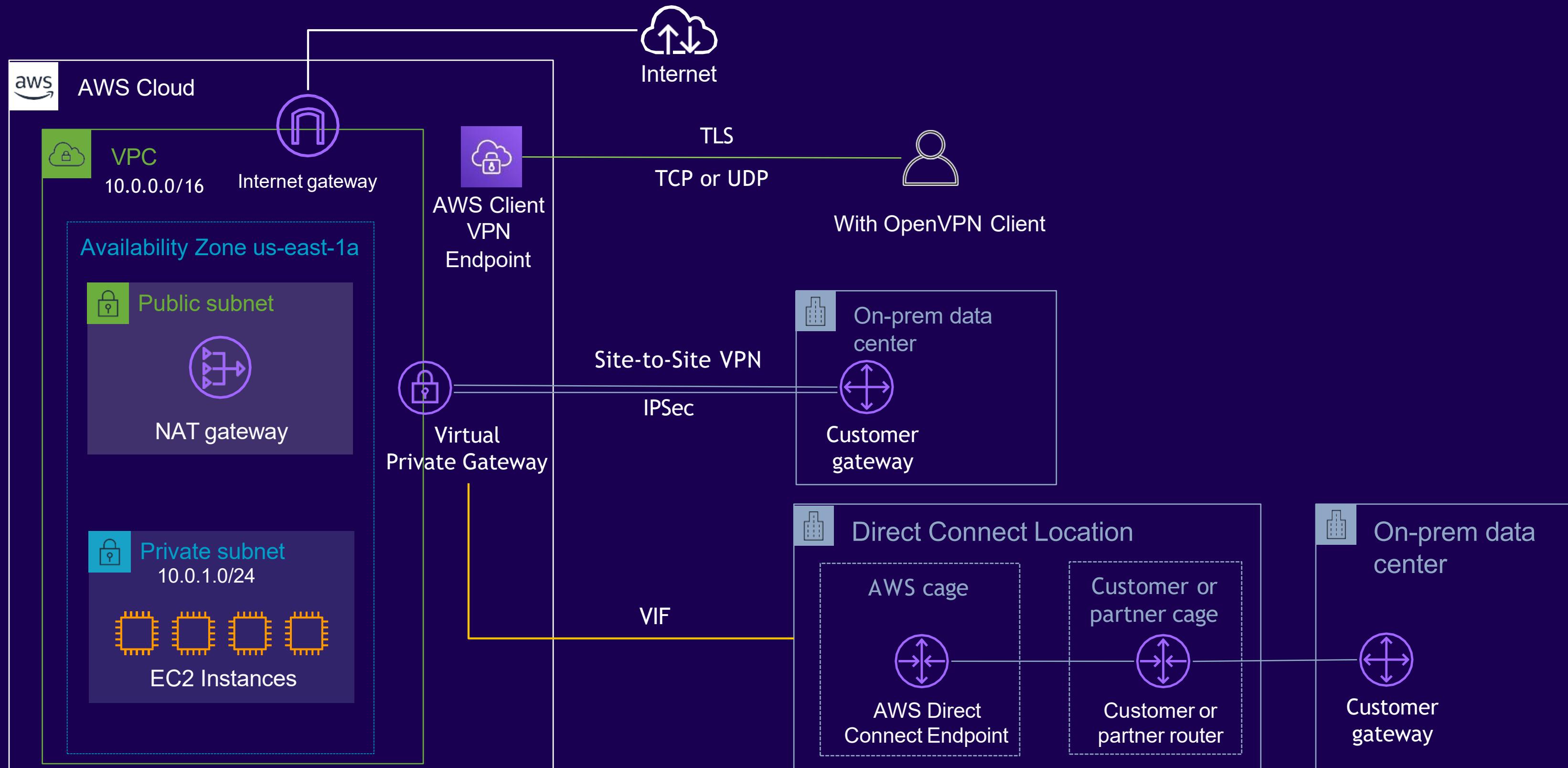
ELB: Load balancer distributes incoming application network traffic

Securing your infrastructure



- **Route tables**
 - Contains a set of rules, called routes, that are used to determine where network traffic is directed
 - Routes tables can have association with VPC, gateways, and subnets
- **Network access control lists (network ACLs)**
 - Allow or deny traffic in and out of subnets
 - Hardens security as a secondary level of defense at the subnet level
- **Security groups**
 - Used to allow traffic to and from at the network interface (instance) level
 - Usually administered by application developers

Connecting with your infrastructure



Getting Down to Essentials

Global infrastructure

Amazon Virtual Private Cloud (Amazon VPC) Basics

of VPC security

Peering, endpoints, and gateways

Global infrastructure

AWS global infrastructure

○ Region & number of Availability Zones (AZs)

GovCloud (U.S.)

U.S.-East (3), US-West (3)

U.S. West

Oregon (4)

Northern California (3)

U.S. East

N. Virginia (6), Ohio (3)

Canada

Central (3)

South America

São Paulo (3)

Africa

Cape Town (3)

Europe

Frankfurt (3), Paris (3),
Ireland (3), Stockholm (3),
London (3), Milan (3)

Middle East

Bahrain (3)

Asia Pacific

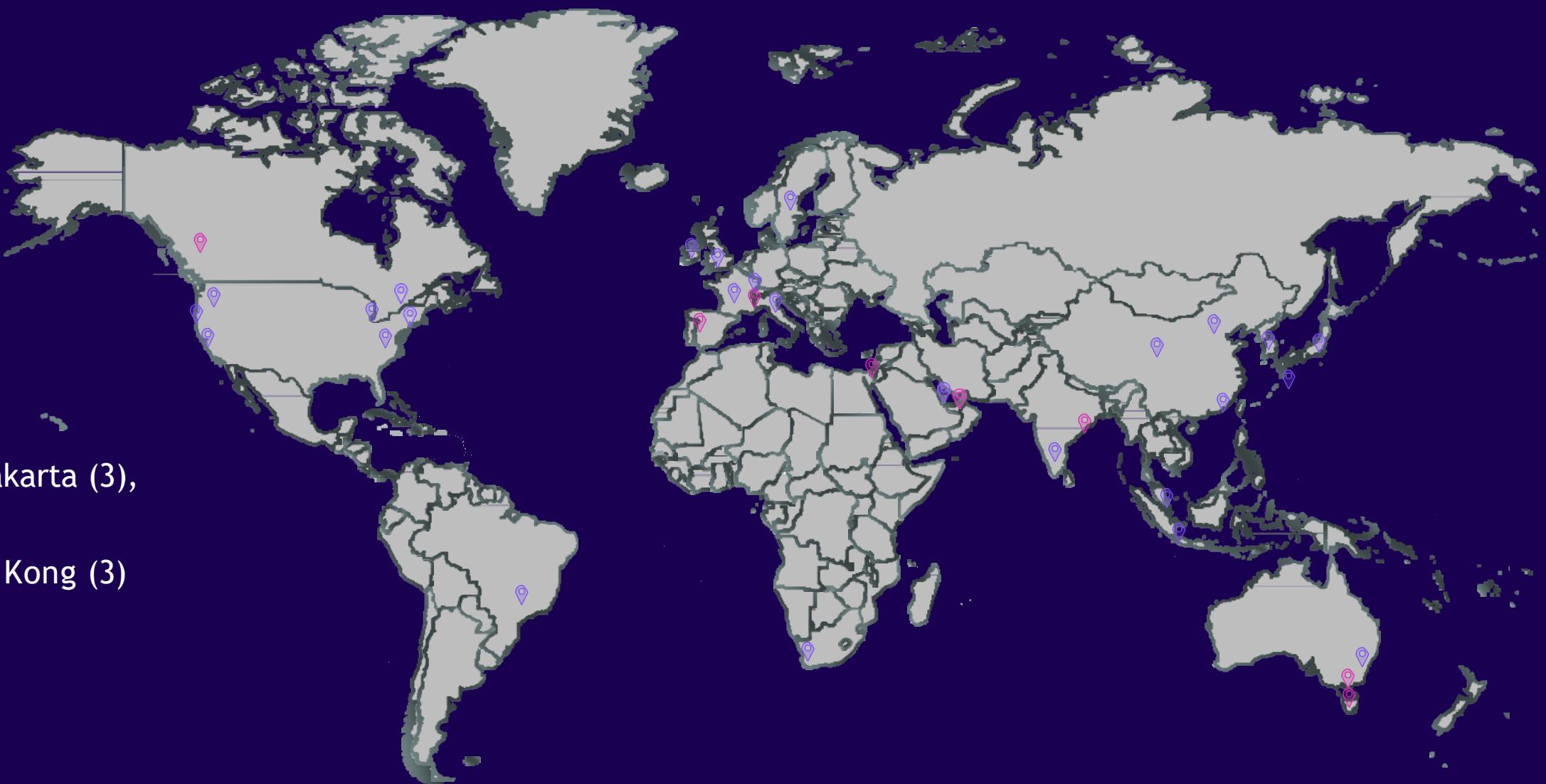
Singapore (3), Sydney (3), Jakarta (3),
Tokyo (4), Osaka (3)
Seoul (4), Mumbai (3), Hong Kong (3)

China

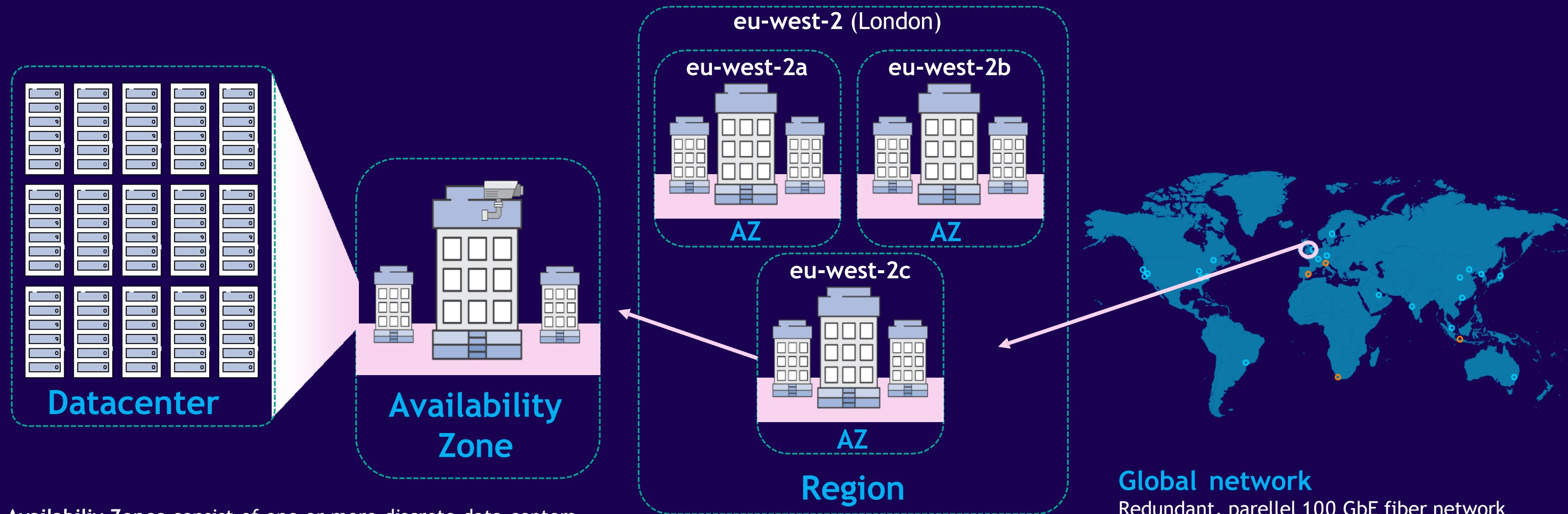
Beijing (3), Ningxia (3)

○ Announced Regions

8 Regions in Australia, Canada, India, Indonesia, Israel, Australia,
Switzerland, Spain, and United Arab Emirates (UAE)



AWS global network components

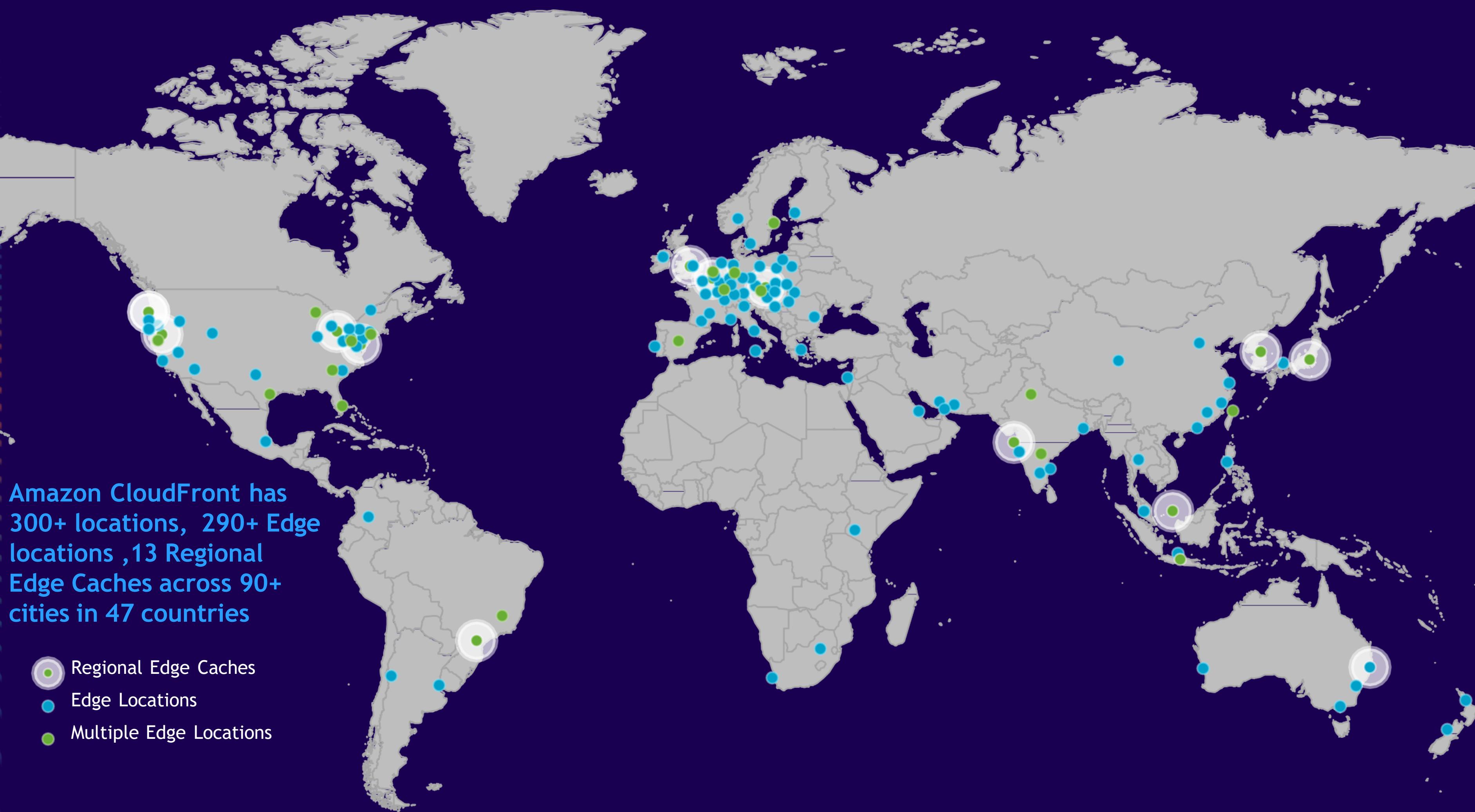


Availability Zones consist of one or more discrete data centers, each with redundant power, networking, and connectivity in an AWS Region.

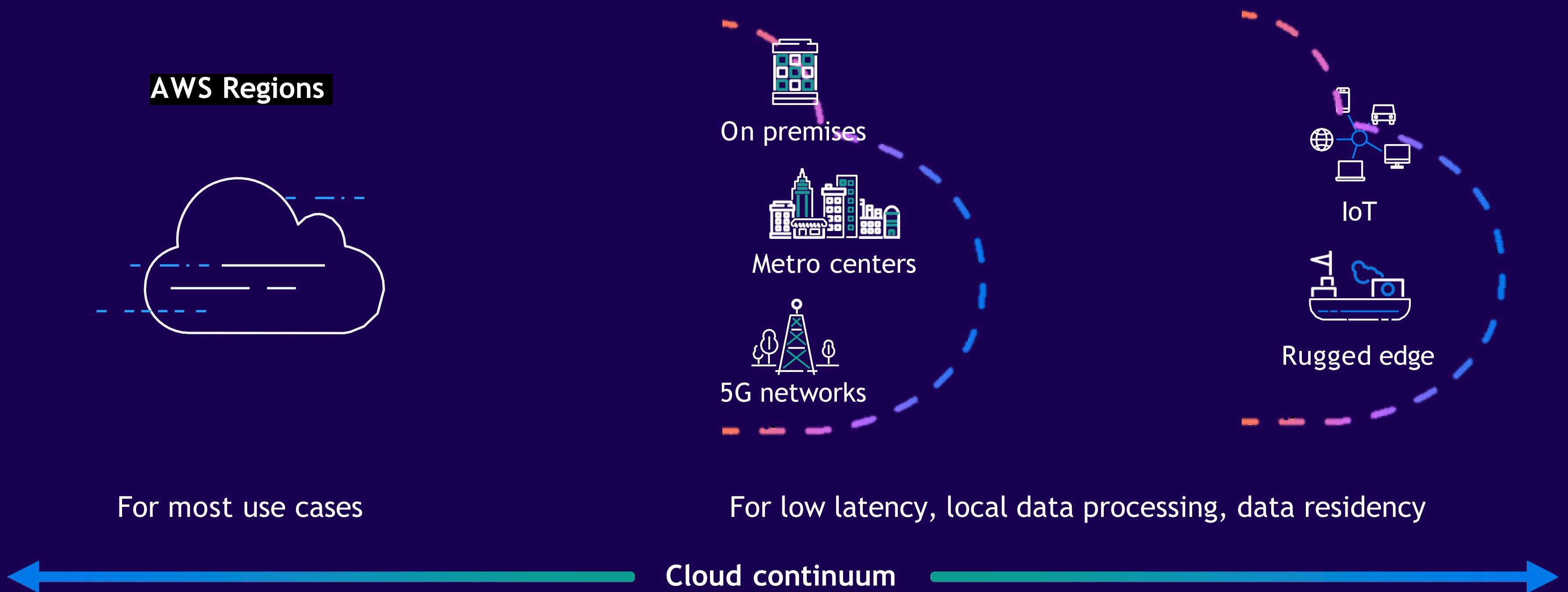
A Region is a physical location in the world where we have multiple Availability Zones.

Global network

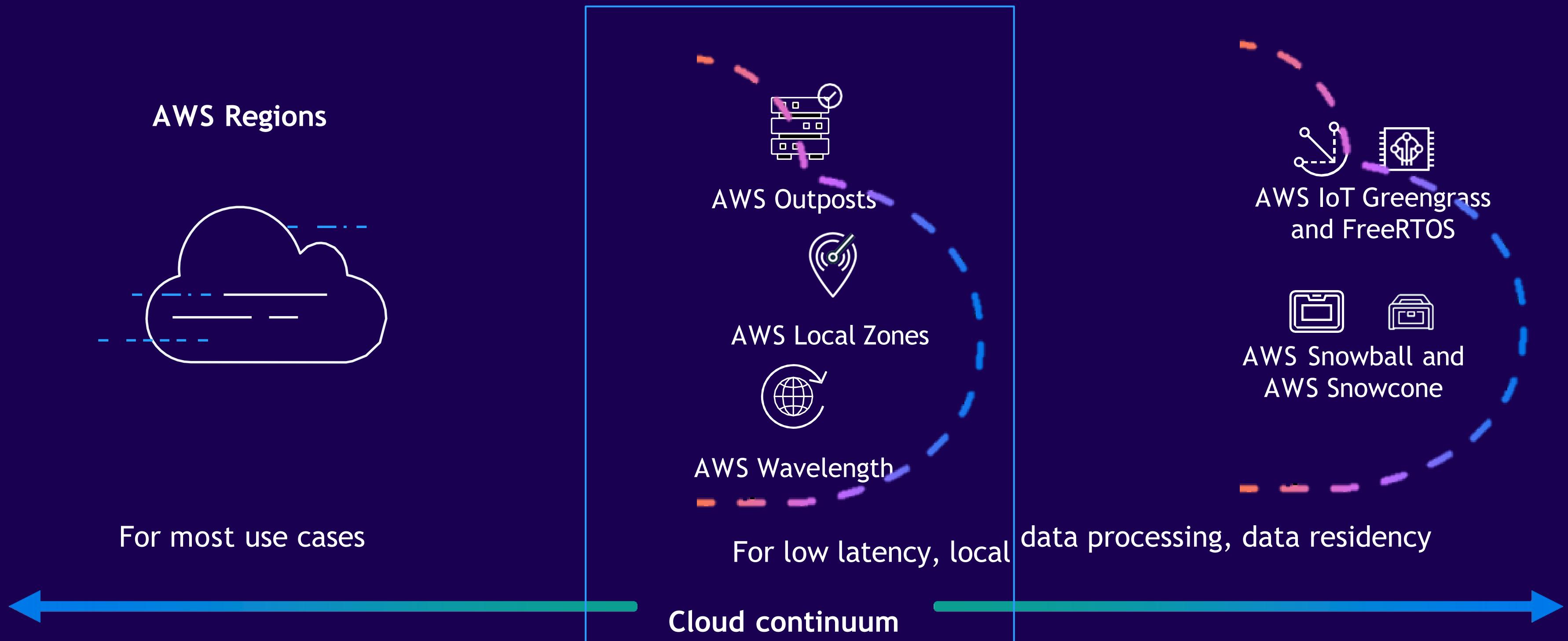
Redundant, parallel 100 GbE fiber network and low-latency private capacity between all regions except China. Includes trans-ocean cables.



Cloud continuum

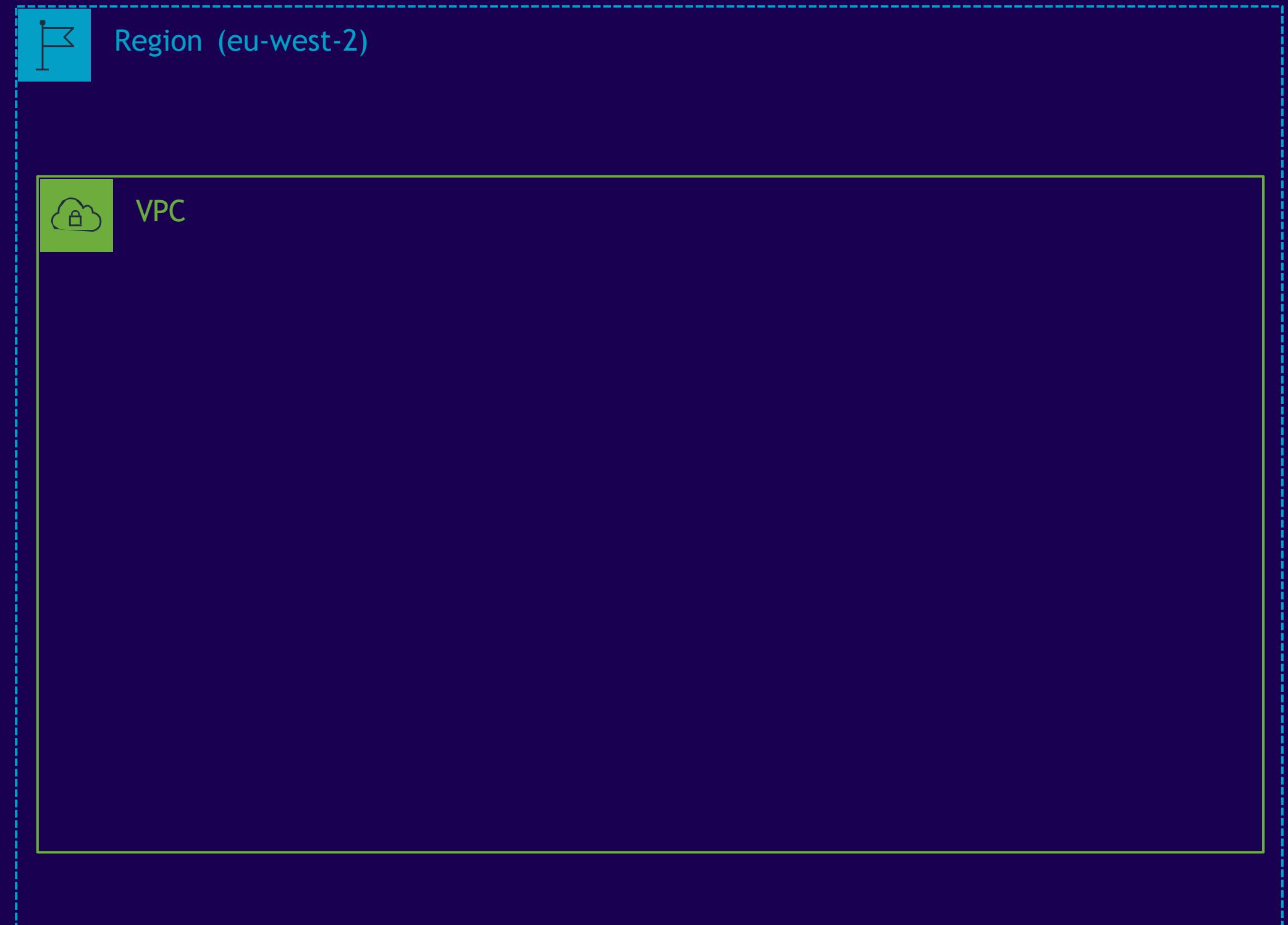
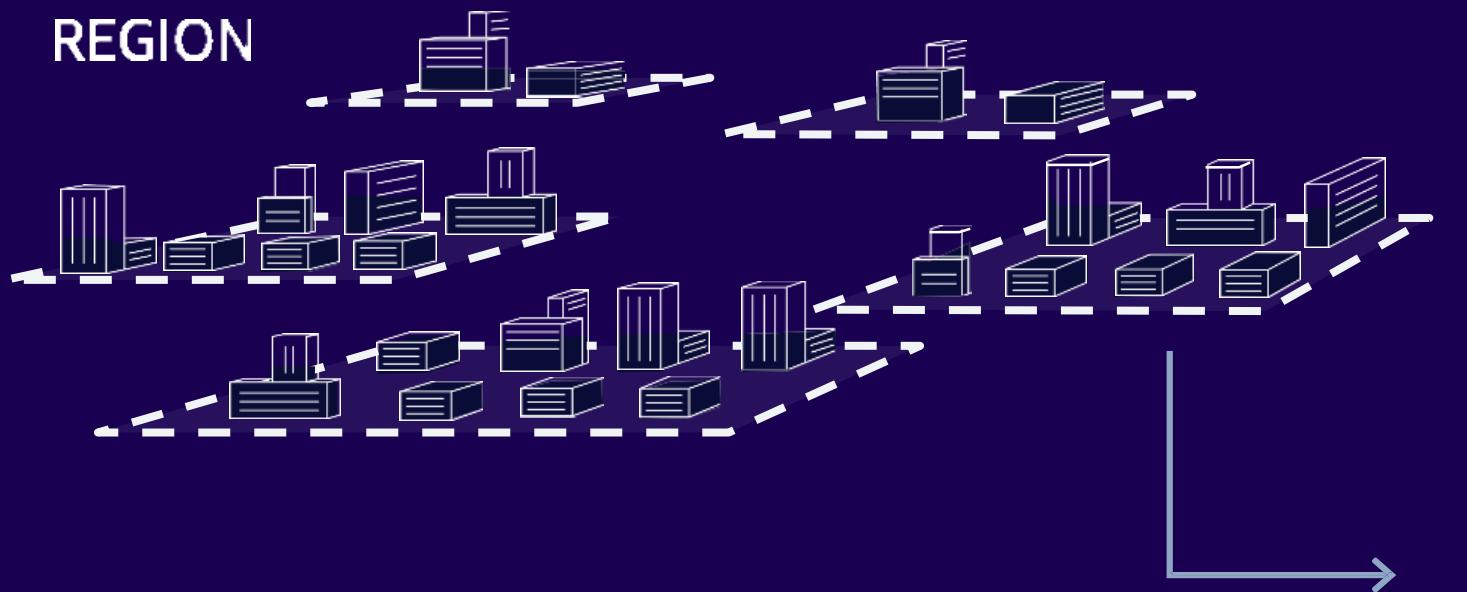


Bringing the cloud to where you need it

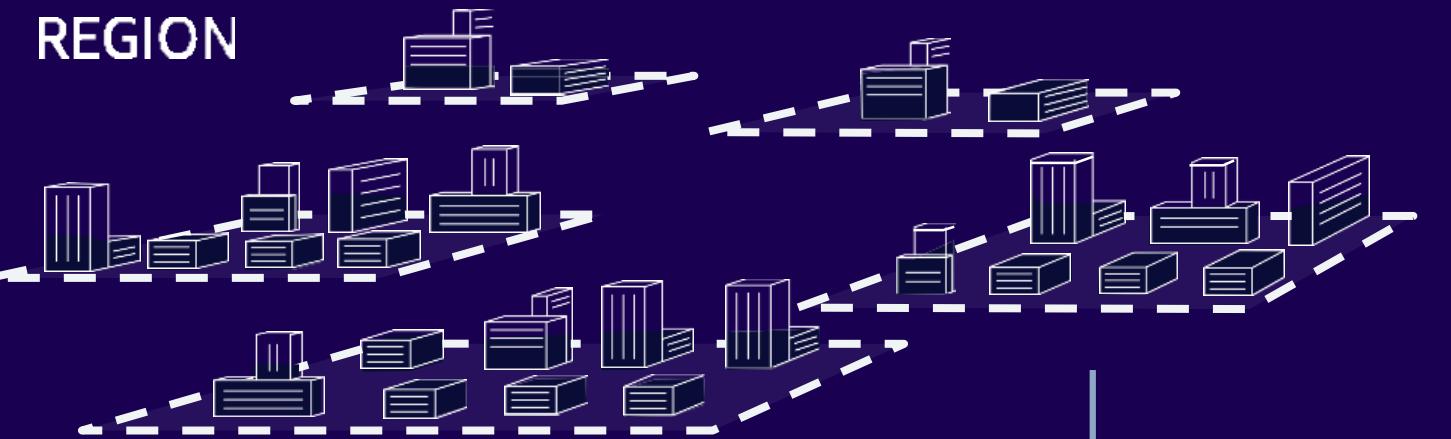


Amazon Virtual Private Cloud

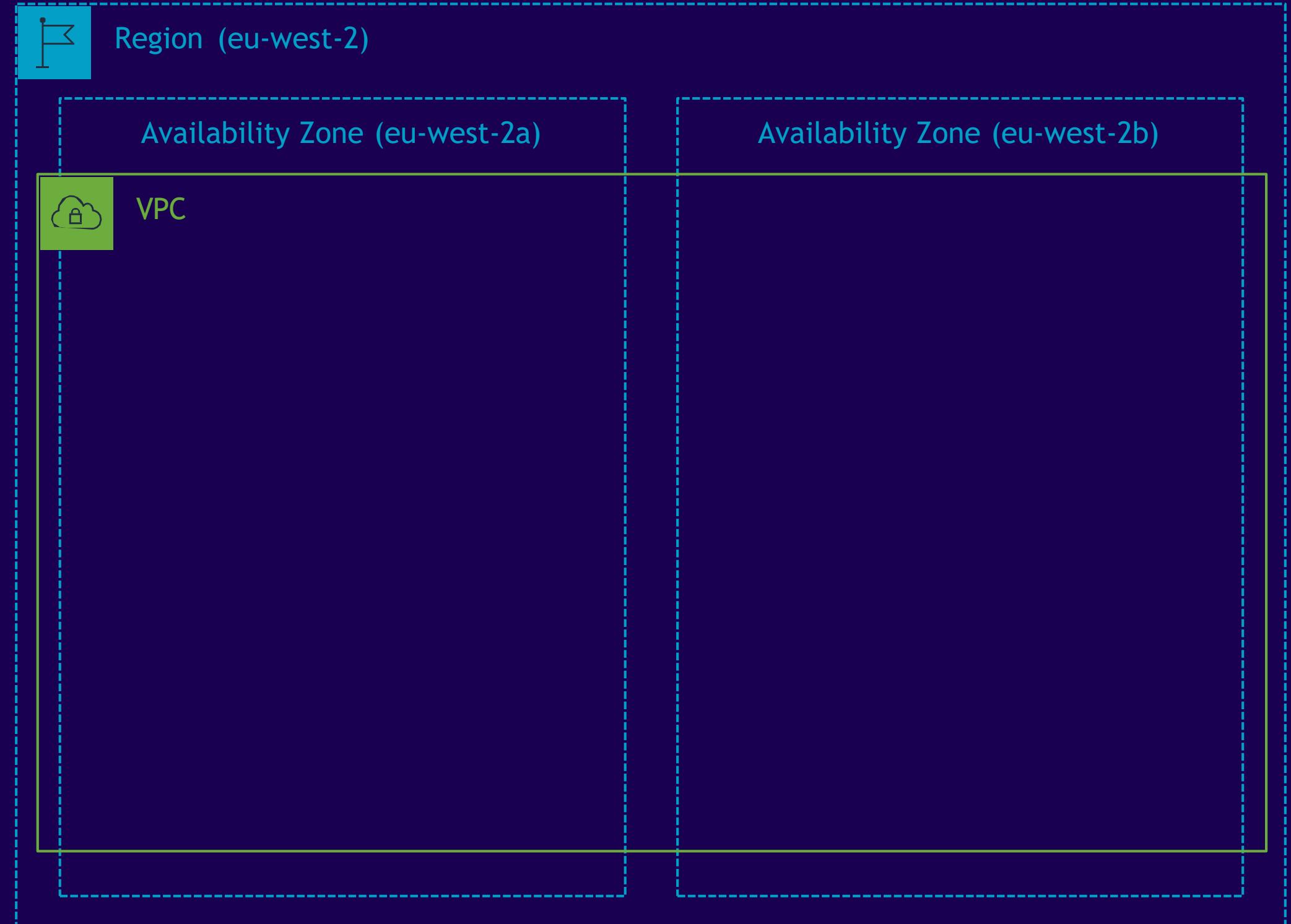
Building a VPC



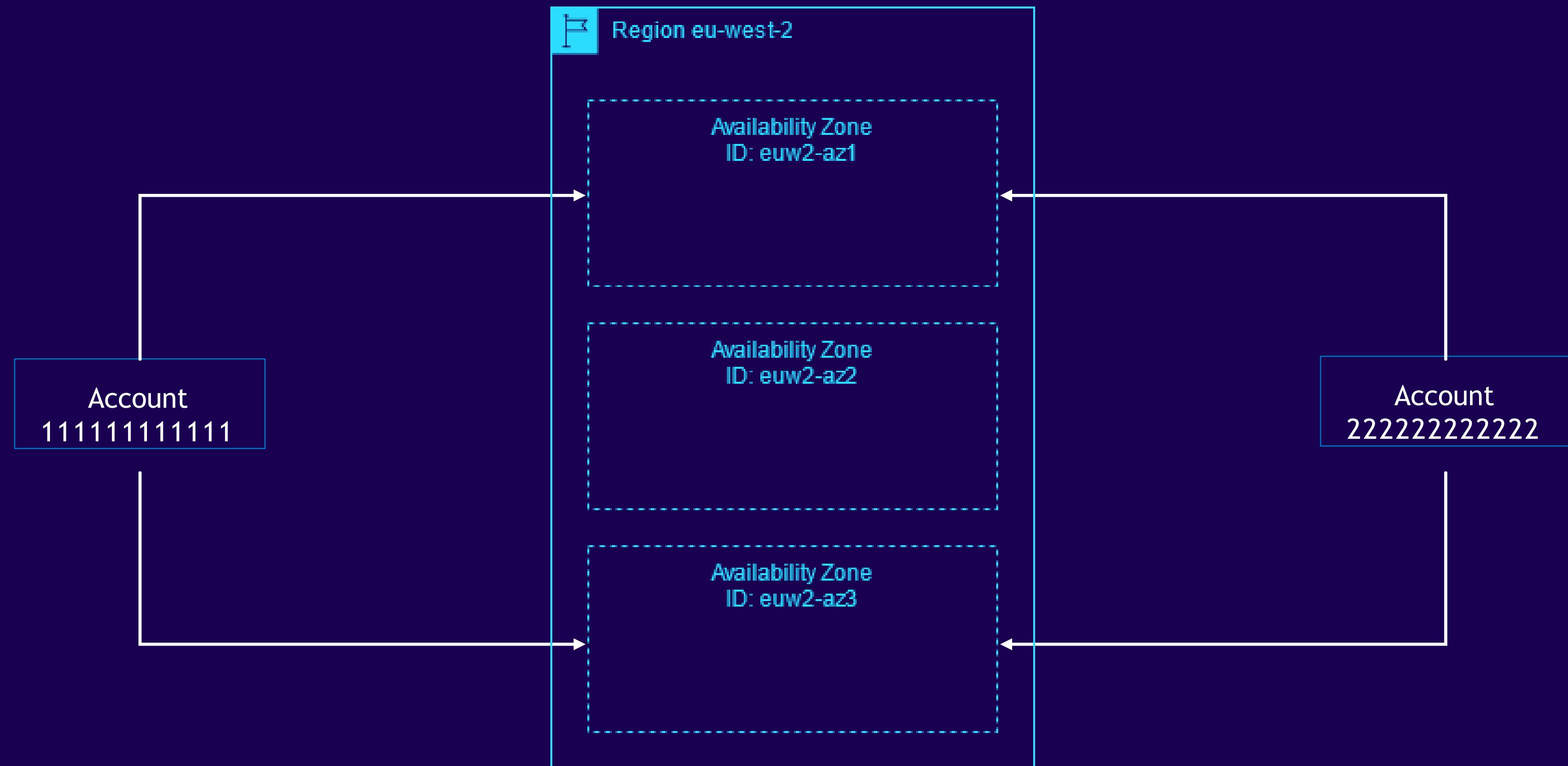
Building a VPC



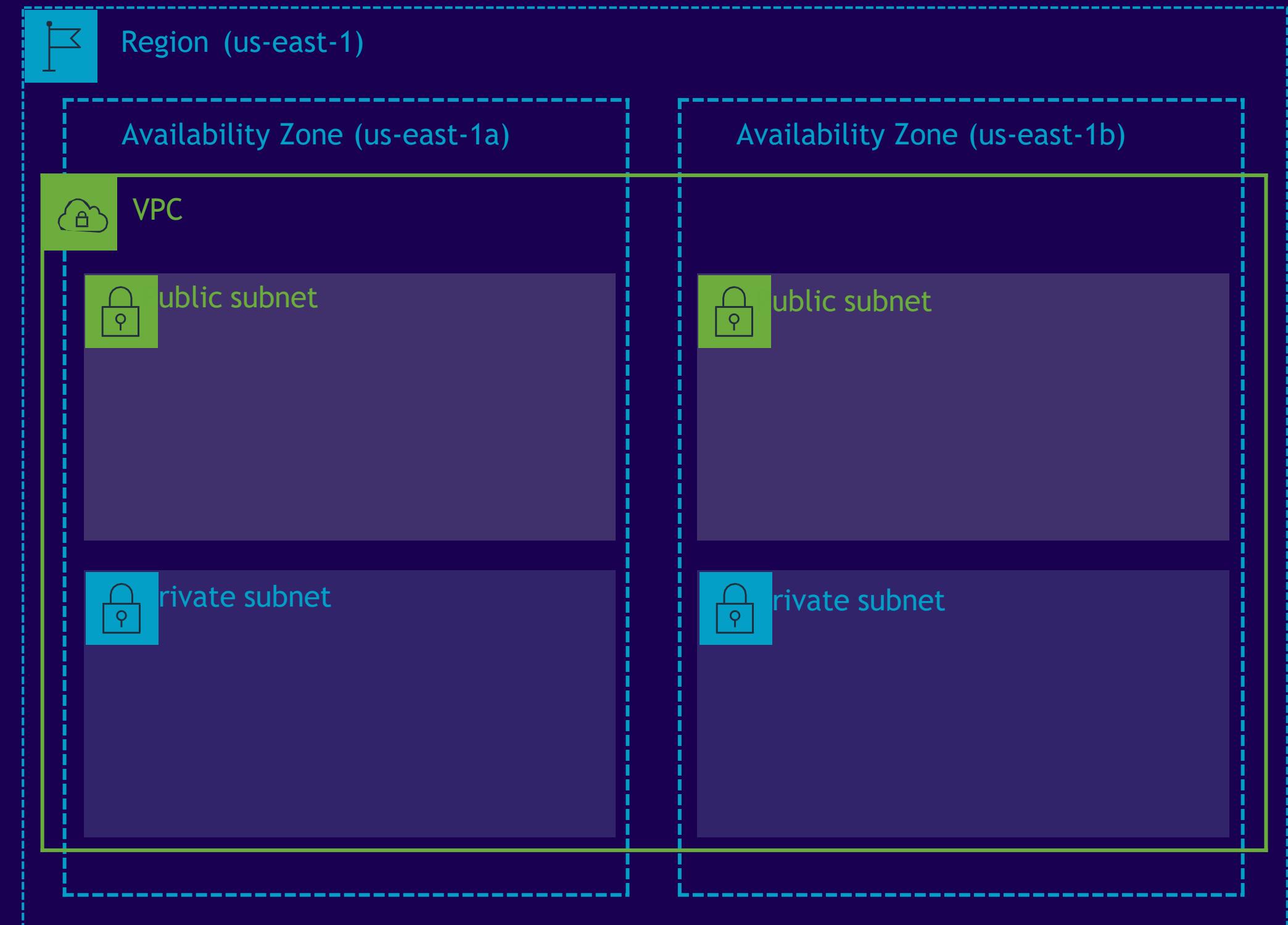
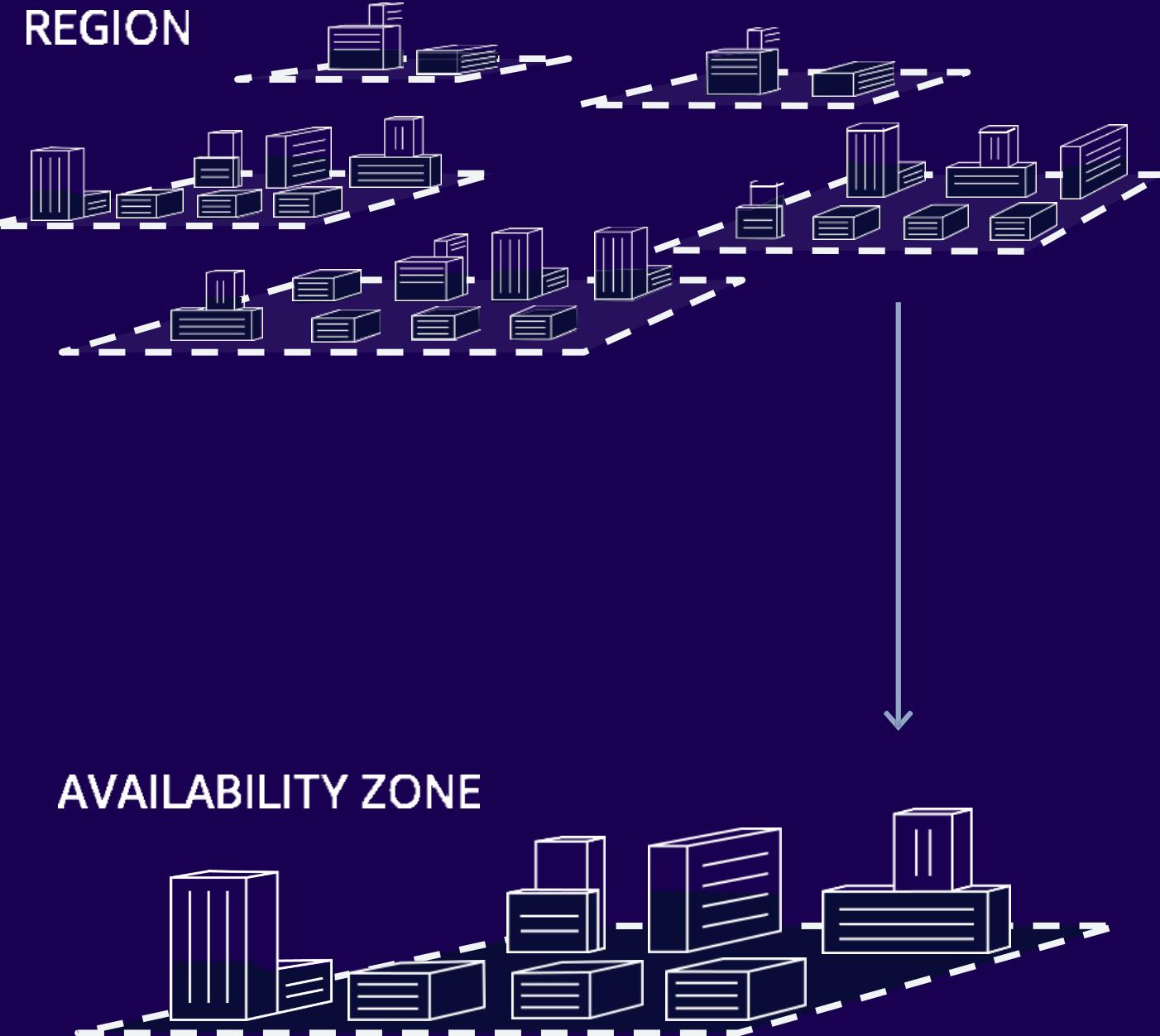
AVAILABILITY ZONE



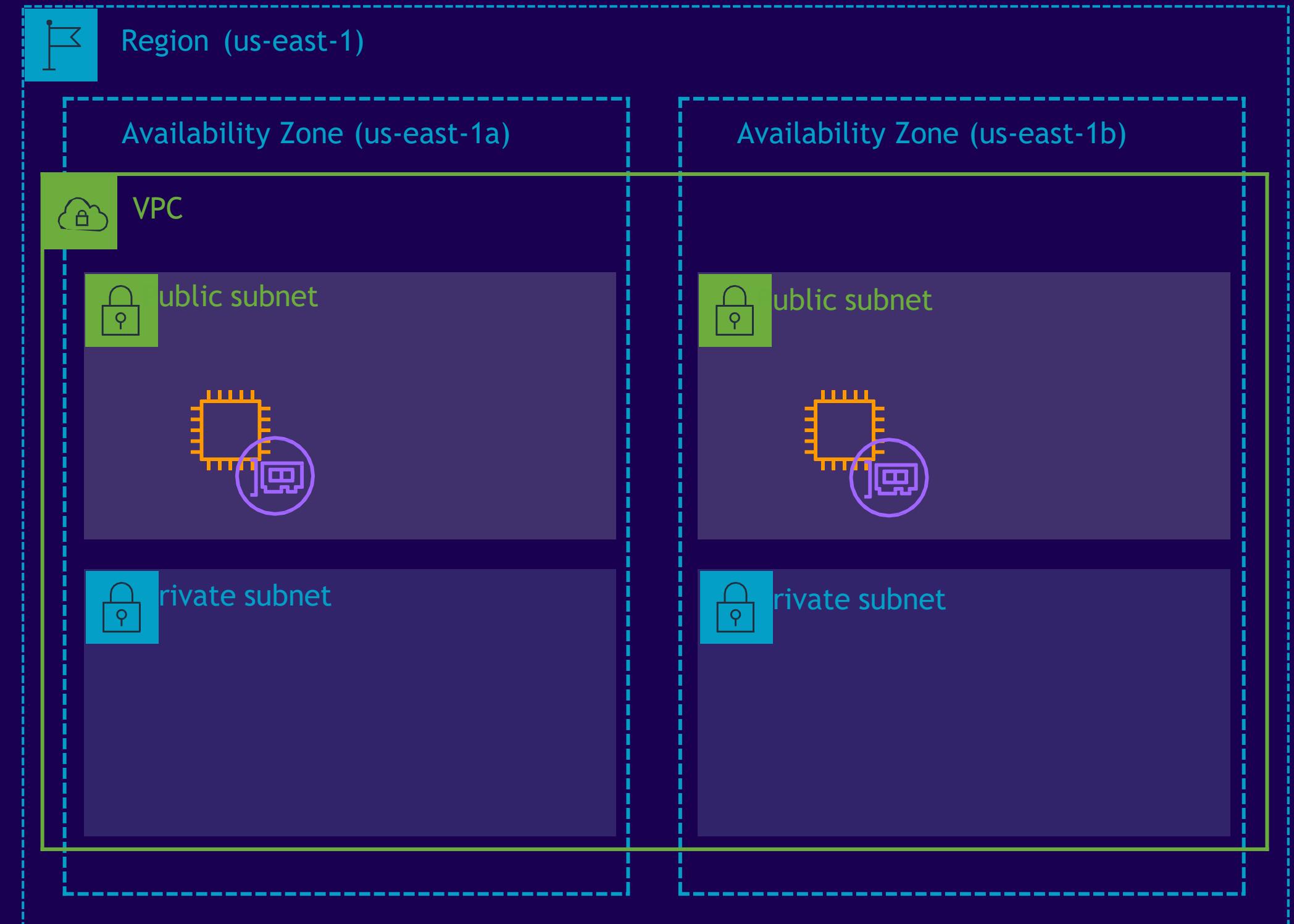
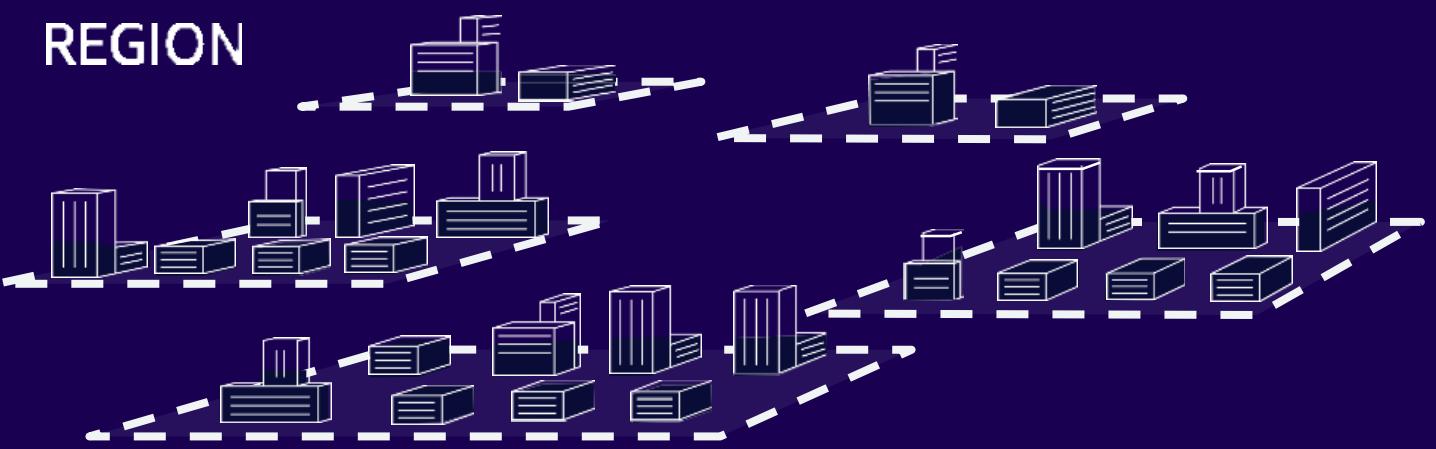
Availability Zone IDs for your AWS resources



Building a VPC



Building a VPC



IPv4 addressing



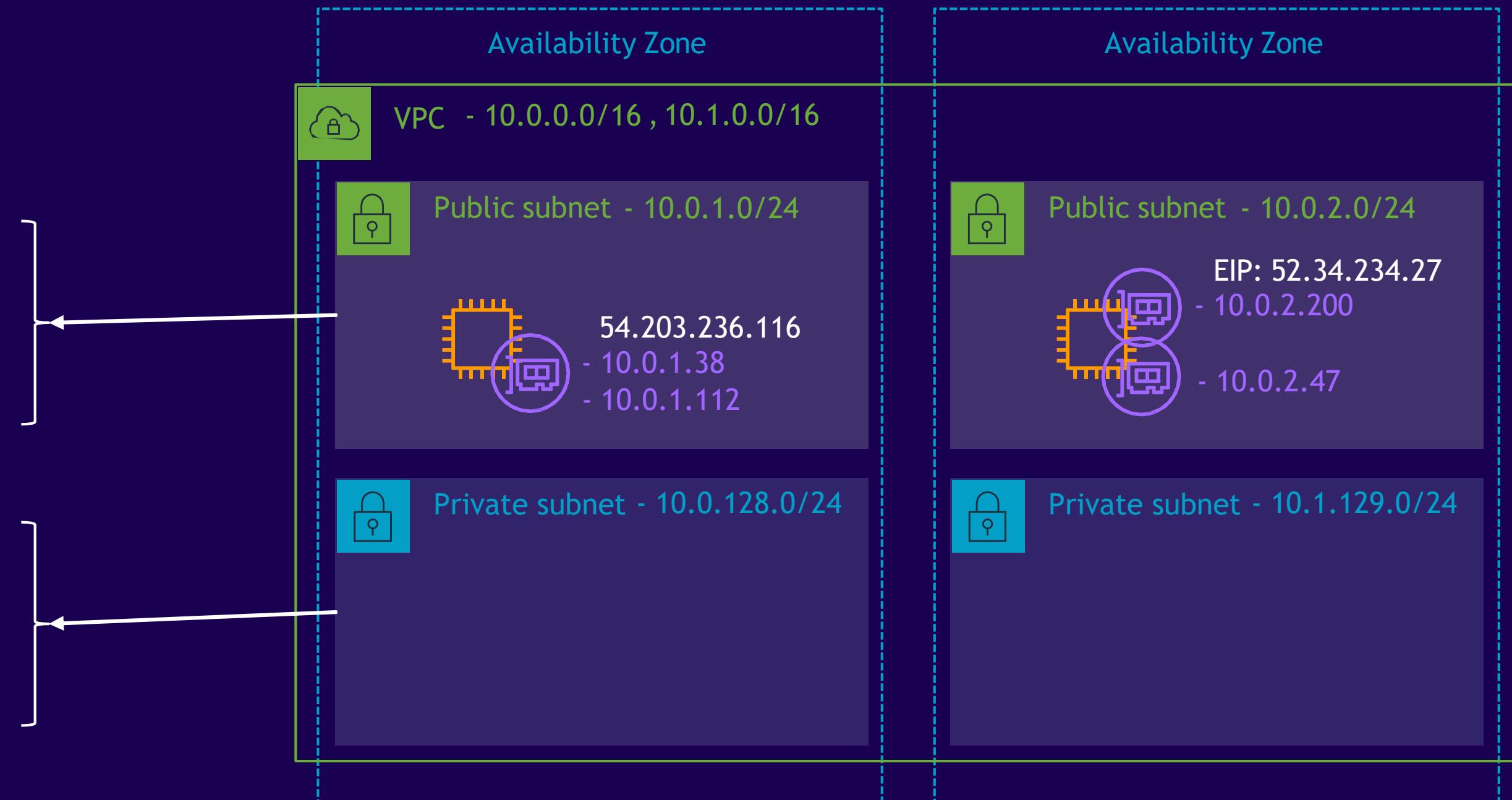
IPv4 addressing

Reserved

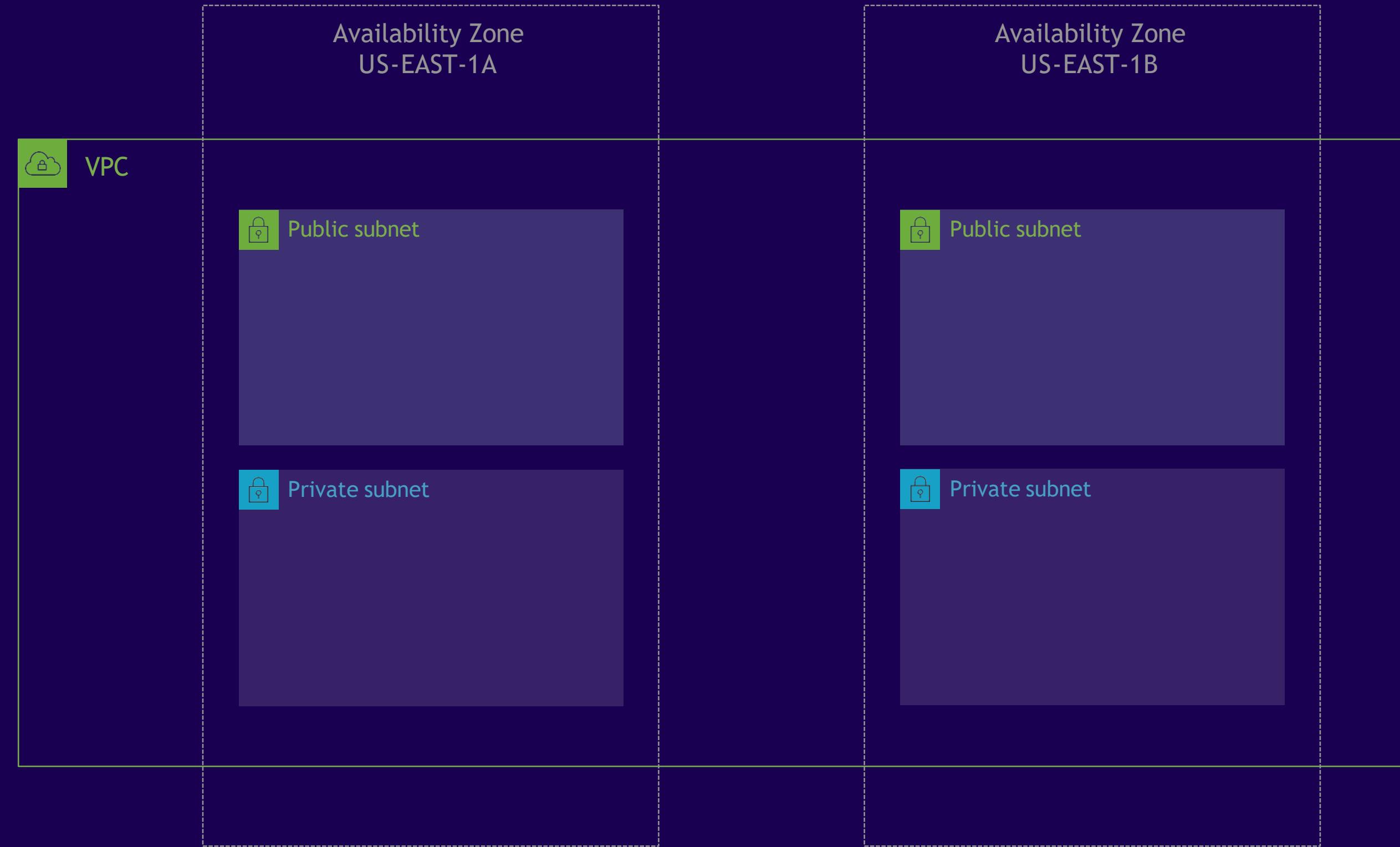
- 1. - Network Address
 - 2. - VPC Router
 - 3. - Reserved
 - 4. - Reserved
- 10.0.1.255 - Network Broadcast

...

- 1. - Network Address
 - 2. - VPC Router
 - 3. - Reserved
 - 4. - Reserved
- 10.0.128.255 - Network Broadcast



IP addressing



Private IP address range for your VPC - IPv4

- ”CIDR” Range ?
 - Classless Inter-domain Routing
 - No more Class A, B, C
- RFC1918
 - 192.168.0.0 /16
 - 172.16.0.0 /12
 - 10.0.0.0 /8
- How much ?
 - /16
 - /28

Updated by: [6761](#)

Network Working Group

Request for Comments: [1918](#)

Obsoletes: [1627](#), [1597](#)

BCP: 5

Category: Best Current Practice

BEST CURRENT PRACTICE

[Errata Exist](#)

Y. Rekhter

Cisco Systems

B. Moskowitz

Chrysler Corp.

D. Karrenberg

RIPE NCC

G. J. de Groot

RIPE NCC

E. Lear

Silicon Graphics, Inc.

February 1996

Address Allocation for Private Internets

Status of this Memo

This document specifies an Internet Best Current Practices for the Internet Community, and requests discussion and suggestions for improvements. Distribution of this memo is unlimited.

1. Introduction

For the purposes of this document, an enterprise is an entity autonomously operating a network using TCP/IP and in particular determining the addressing plan and address assignments within that network.

This document describes address allocation for private internets. The allocation permits full network layer connectivity among all hosts inside an enterprise as well as among all public hosts of different enterprises. The cost of using private internet address space is the potentially costly effort to renumber hosts and networks between public and private.

CIDR

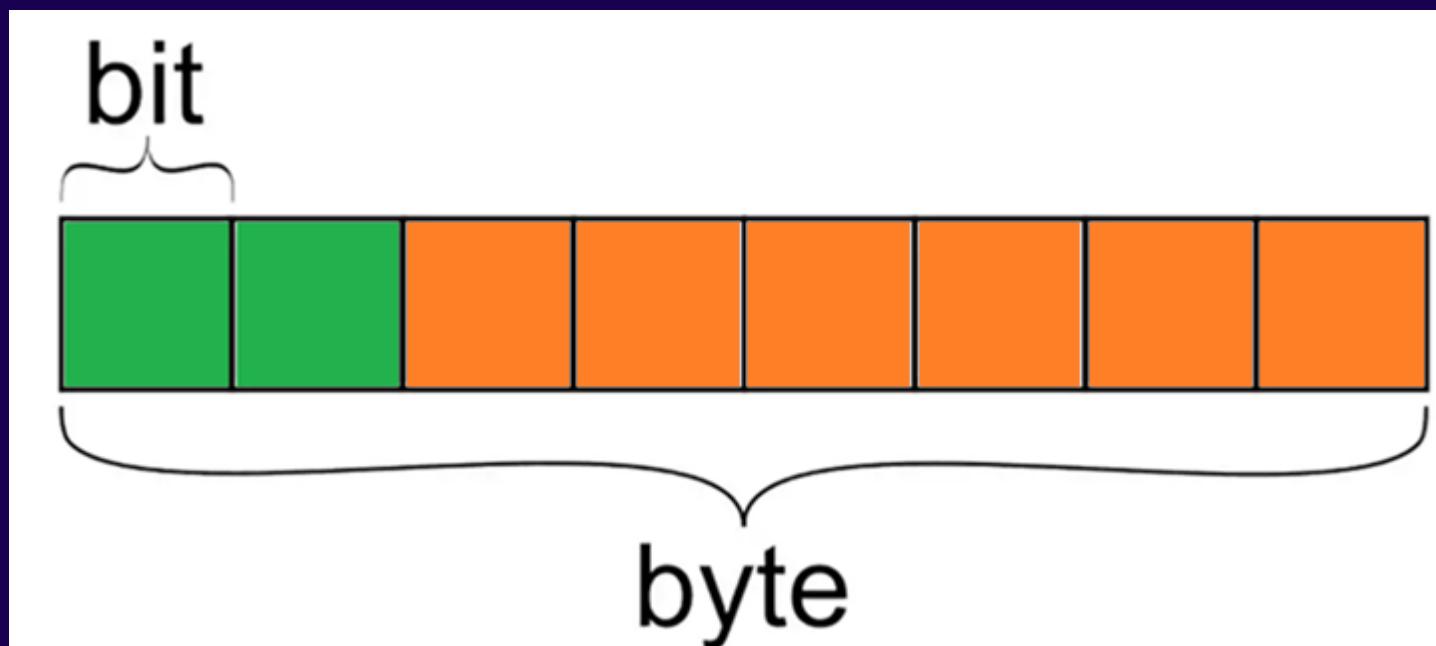
IP address is the identity of each host in the network

IPv4 = 32 Bits

e.g.

192.168.56.212

IPv6 = 128 Bits



e.g. $192 = 2^7 + 2^6 = 1\ 1\ 0\ 0\ 0\ 0\ 0$

11 000000 . 10101000. 00111000. 11010100

CIDR

192.168.0.0/16

192

.168

.0

.0

11000000.

10101000

00000000.

00000000

16 Bits

16 Bits

Network Address
192.168
Fixed

Host Addresses

0-255.0-255

0 to 255 in each available octet

$2^{16} - 2 = 65534$

CIDR

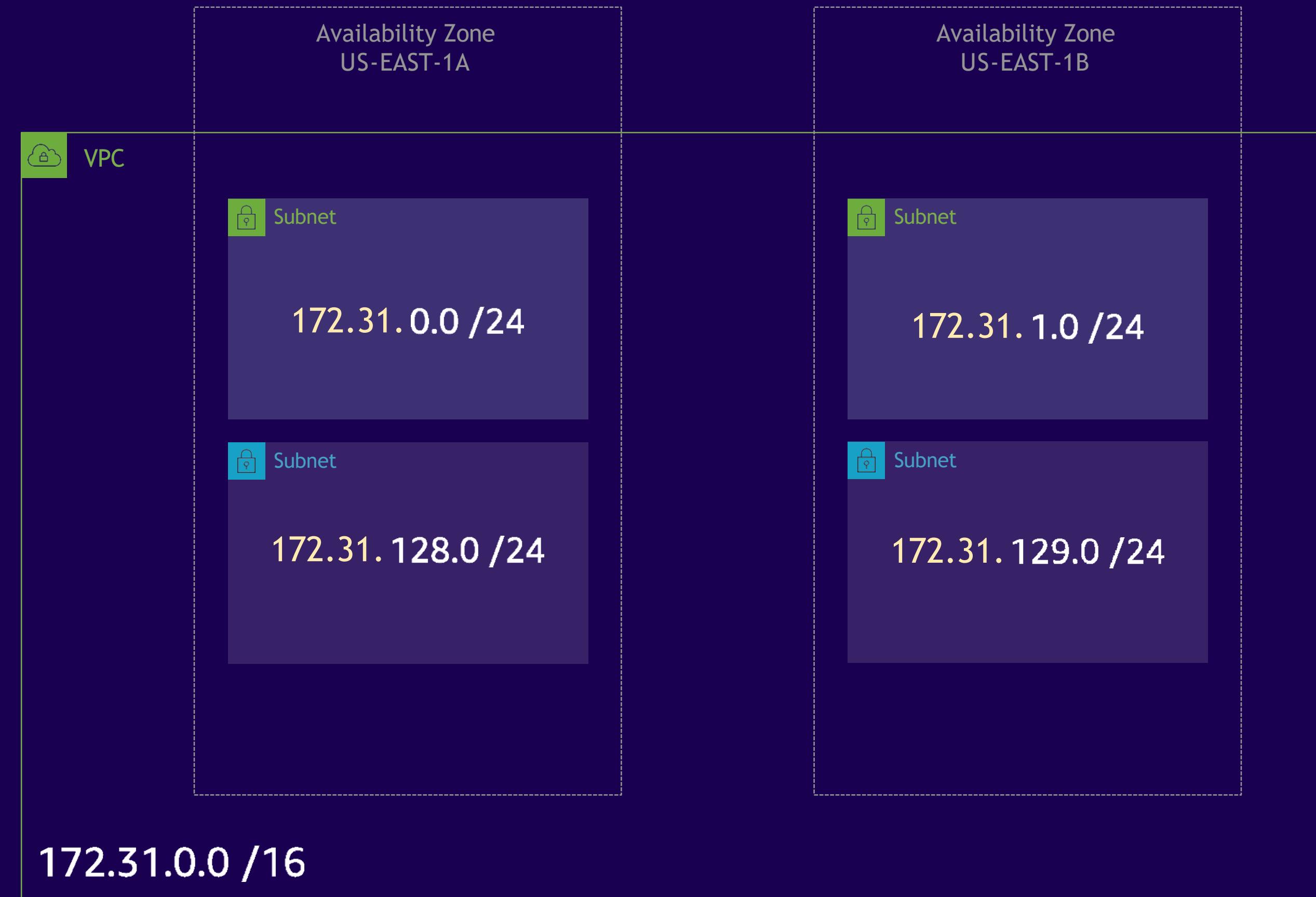
- **AWS VPC CIDR (IPv4)**

- VPC prefix between /16 (65536 IPs) and /28 (16 IPs)
- RFC 1918 IP ranges for Private network and corresponding AWS recommended ranges
 - 10.0.0.0/8 => 10.0.0.0 – 10.255.255.255 => **AWS CIDR 10.X.0.0/16**
 - 172.16.0.0/12 => 172.16.0.0 - 172.31.255.255 => **AWS CIDR 172.16.0.0/16 to 172.31.0.0/16**
 - 192.168.0.0/16 => 192.168.0.0 - 192.168.255.255 => **AWS CIDR 192.168.0.0/16**
- Subnet CIDR prefix between /16 to /28 (same as VPC CIDR)

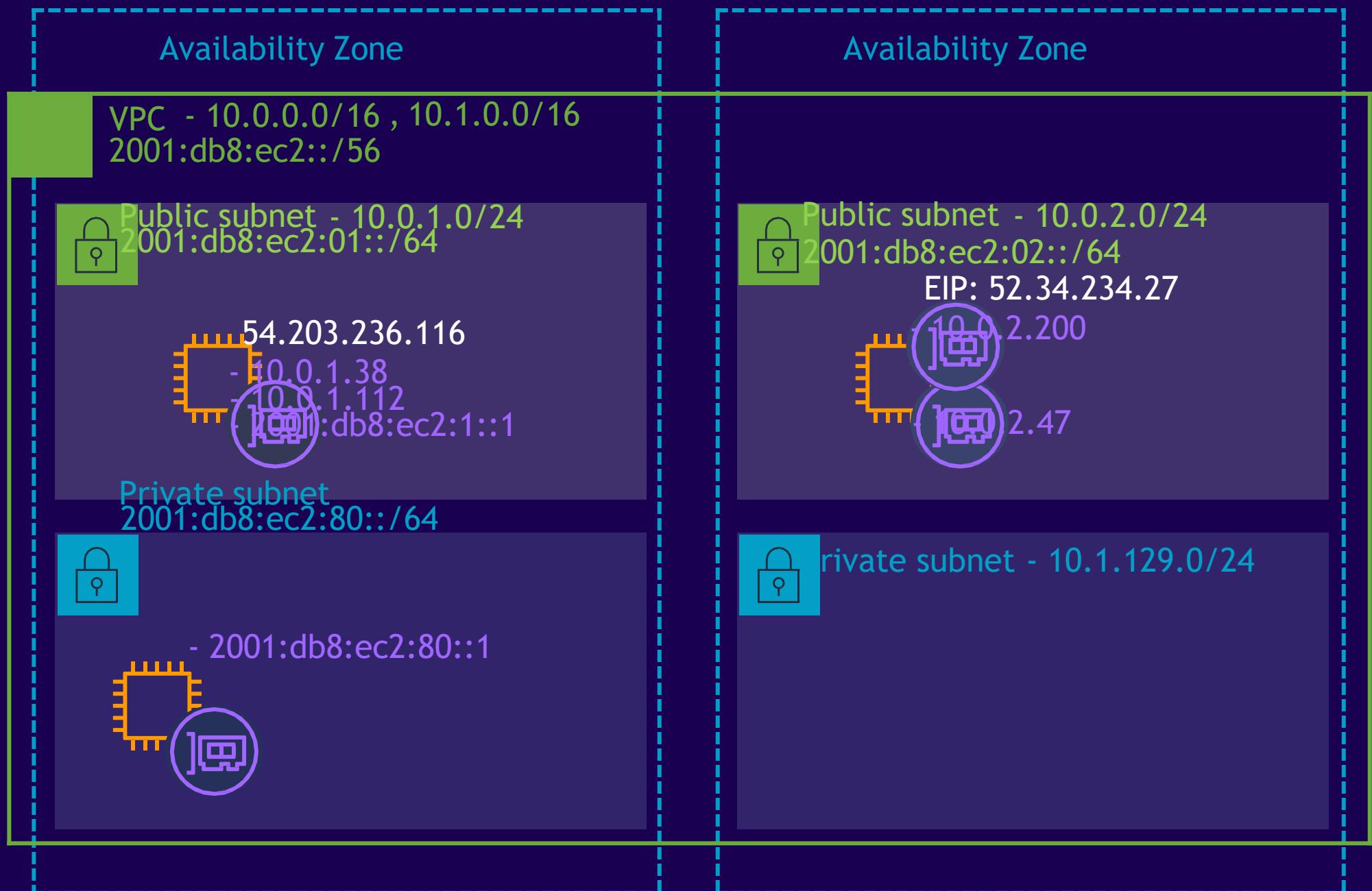
- **AWS VPC CIDR (IPv6)**

- VPC CIDR with prefix /56 (2^{72} IPs)
- IPv6 CIDR is allocated by AWS
- Subnet CIDR prefix /64
- IPv6 IP addresses are globally unique and publicly routable

Where to use IPv4 addresses ?



IPv6 addressing



Reserved
fd00:ec2::/32 - Reserved
fe80::X:Xff:feX:X/64 - VPC Router

2001:db8:ec2:01::0
2001:db8:ec2:01::1
2001:db8:ec2:01::2
2001:db8:ec2:01::3
2001:db8:ec2:01:ffff:ffff:ffff
...
2001:db8:ec2:80::0
2001:db8:ec2:80::1
2001:db8:ec2:80::2
2001:db8:ec2:80::3
2001:db8:ec2:80:ffff:ffff:ffff

IPv6 basics

IPv6: Colon-Separated Hextet Notation + CIDR

2001:0db8:0ec2:0000:0000:0000:0001/64

0000:0000:0000:0000:0000:0000:0001/128

2001:db8:ec2:0:0:0:0:1/64

0:0:0:0:0:0:1/128

2001:db8:ec2::1/64

::1/128

Unicast Addresses

Loopback Address

::1

Link Local Address (LLA)

fe80::/10 (fe80::/64 in practice)

Global Unicast Address (GUA)

2600:1f16:14d:6300::/64

Multicast Addresses (ff00::/8)

All Nodes

ff02::1

All Routers

ff02::2

Solicited Node

ff02::1:ff00:0/104



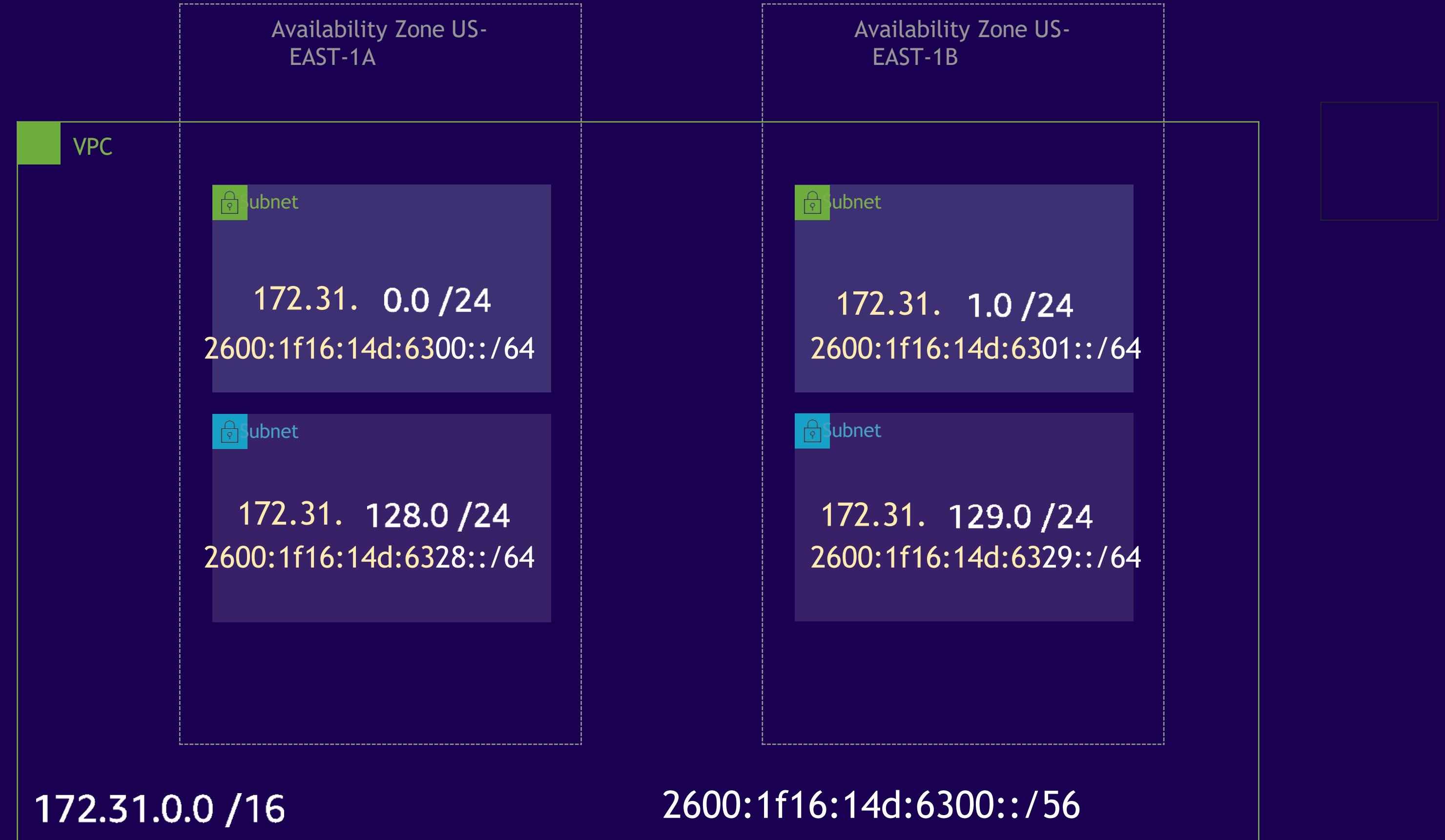
IPv6 on AWS

- /56 VPC
- /64 Subnets
- Dualstack
- Link Local Address and Global Unicast Address required

```
$ ifconfig
eth0      Link encap:Ethernet  Hwaddr 0E:A2:04:52:2A:44
          inet addr:172.31.0.250  Bcast:172.31.0.255  Mask:255.255.255.0
          inet6 addr: fe80::ca2:4ff:fe52:2a44/64 Scope:Link
          inet6 addr: 2600:1f16:14d:6300:7965:9a71:653a:822b/64 Scope:Global
          UP BROADCAST RUNNING MULTICAST  MTU:9001  Metric:1
          RX packets:35090 errors:0 dropped:0 overruns:0 frame:0
          TX packets:12411 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:49899286 (47.5 MiB)  TX bytes:840649 (820.9 KiB)
```

IPv4 Private Address
IPv6 Link Local Address (Private)
IPv6 Global Unicast Address (Public)

Where to use IPv6 addresses ?



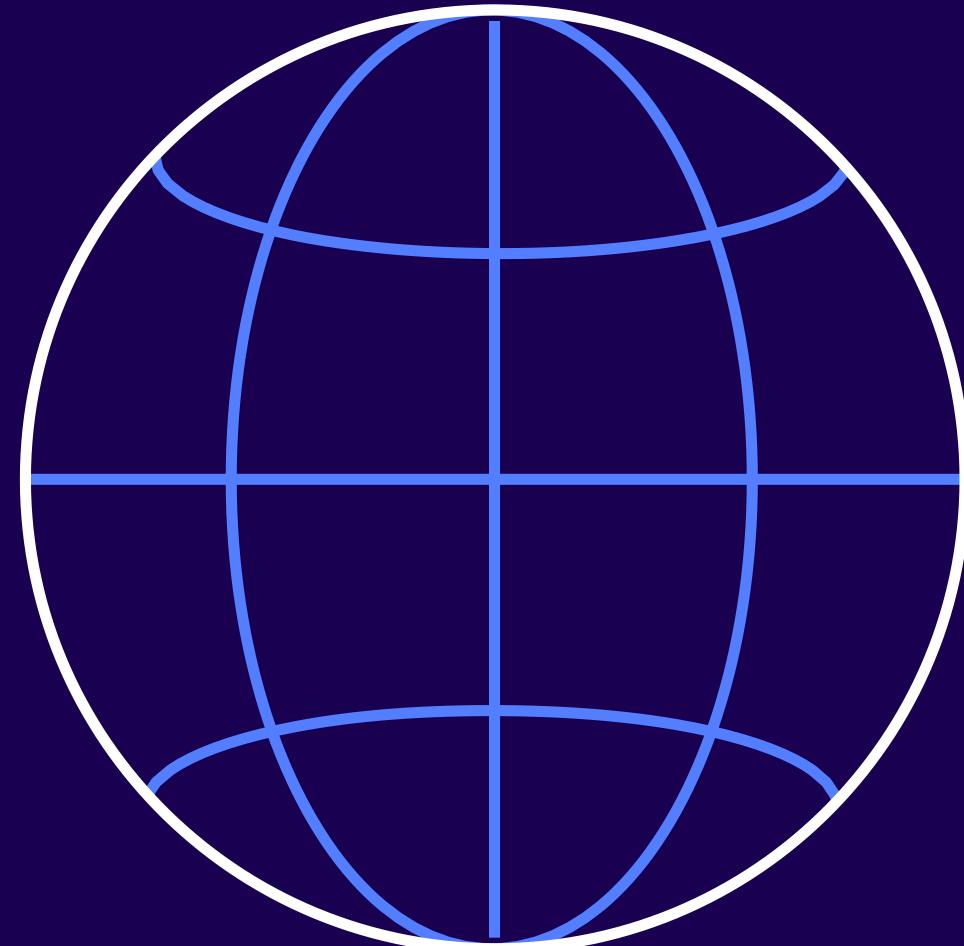
The “5 Things” required for Internet traffic

1. Public IP Address
2. Internet Gateway Attached to a VPC
3. Route to an Internet Gateway
4. NACL Allow Rule
5. Security Group Allow Rule

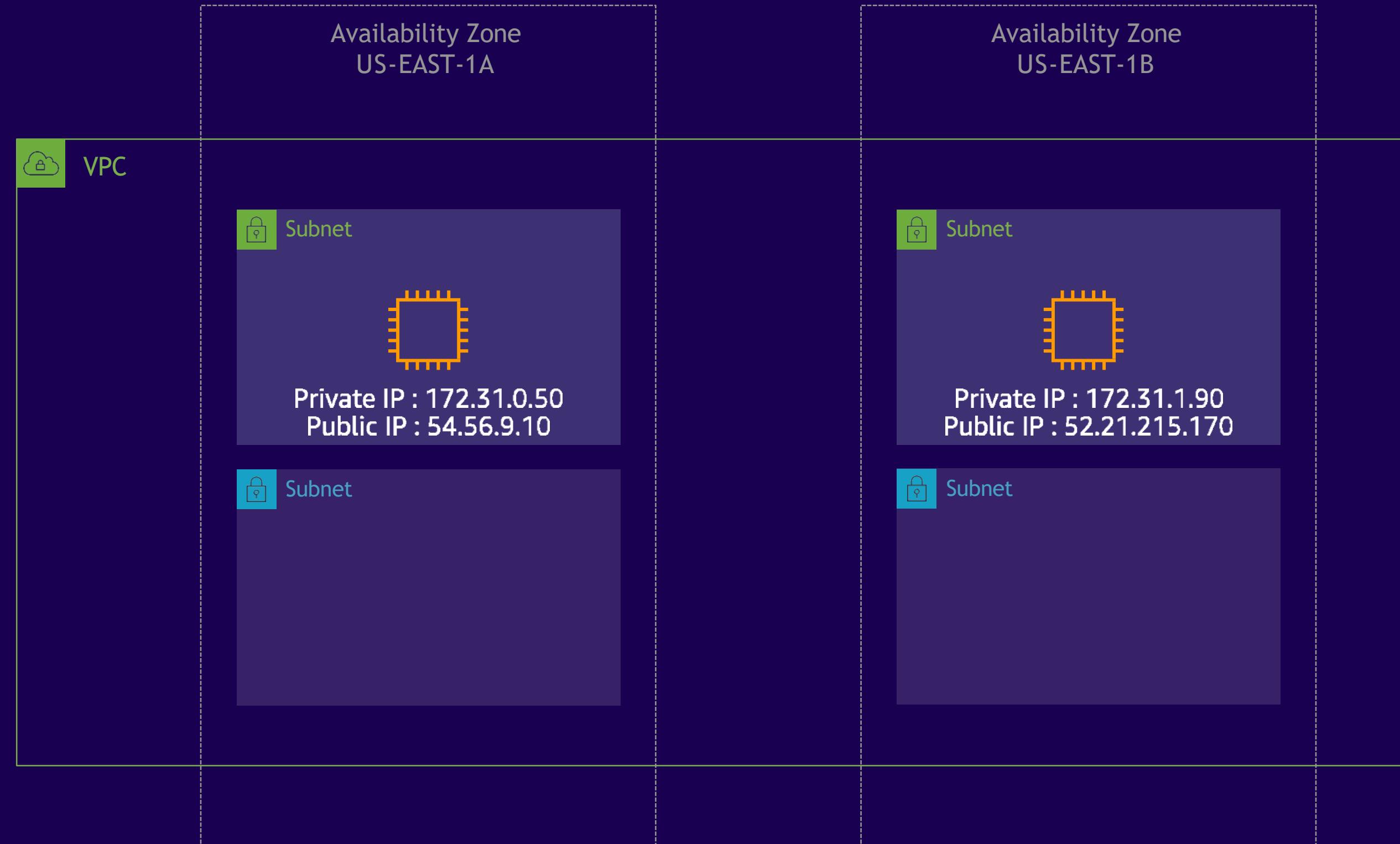


Public IP addresses for your instances

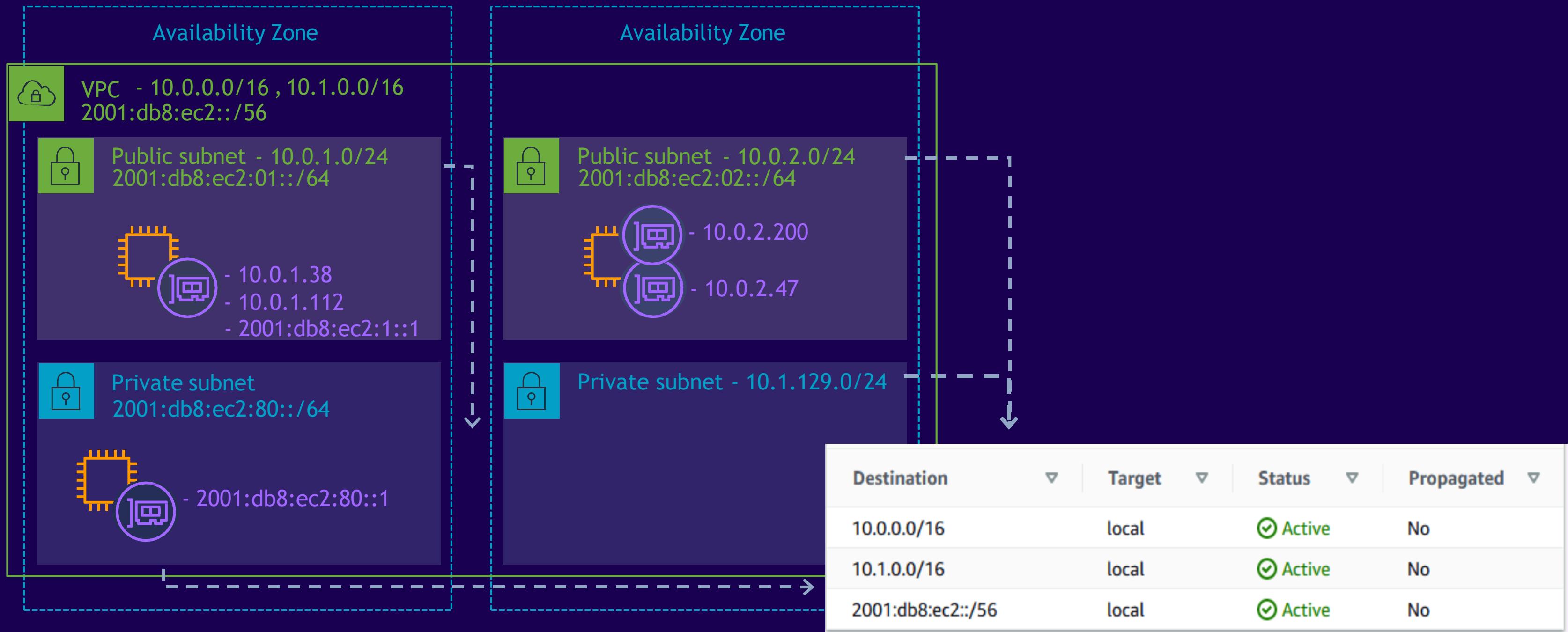
- Auto-assign public IP addresses
- Elastic IP Addresses (EIP)
 - Amazon EIP Pool
 - Bring Your Own IP (BYOIP) Pool



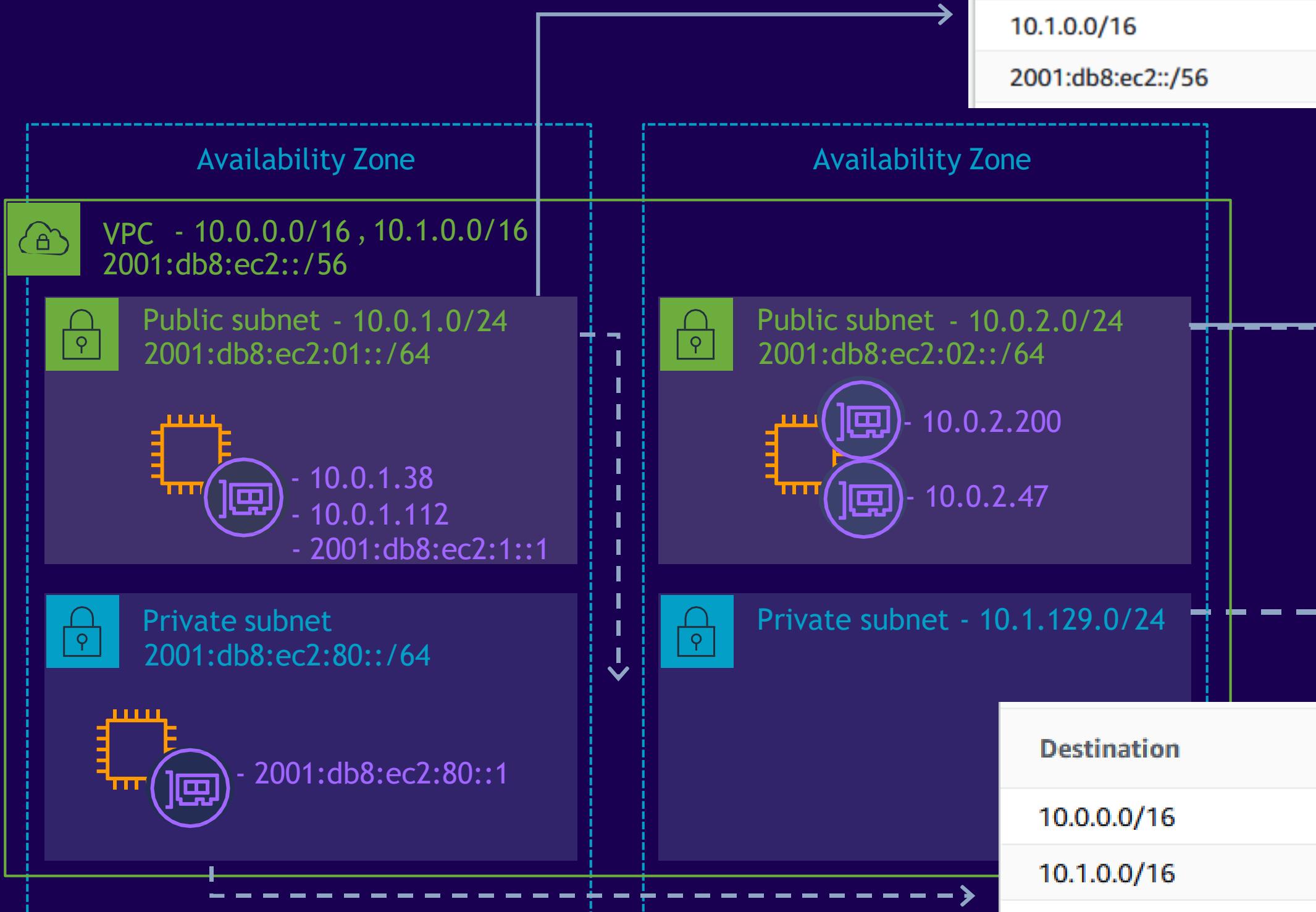
Public IP addresses



Intra-VPC routing



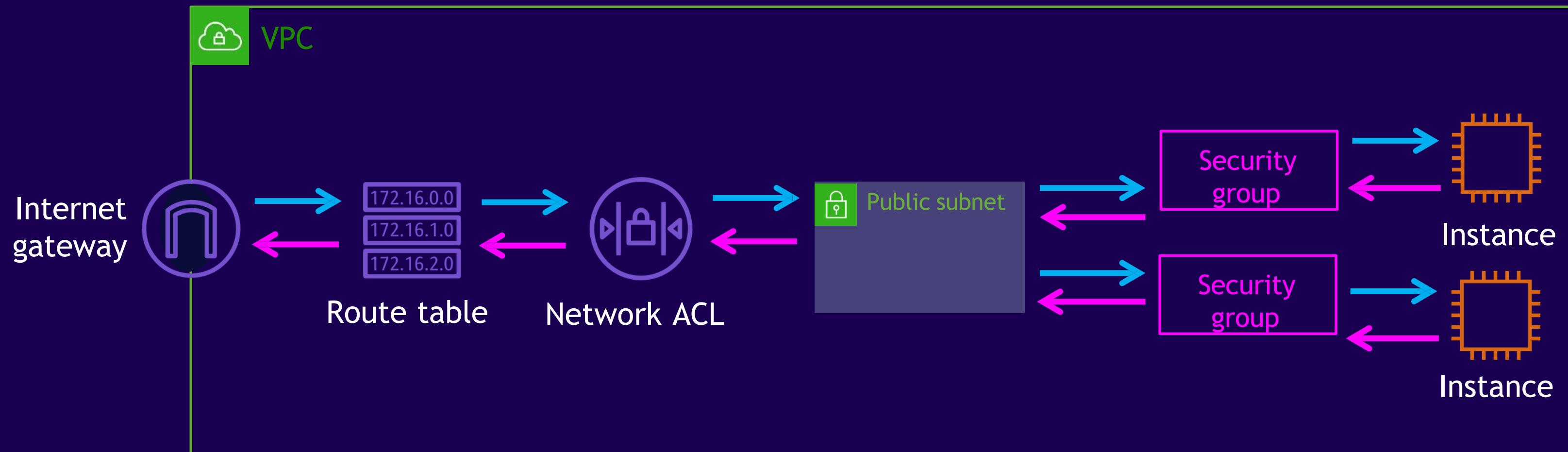
Intra-VPC routing



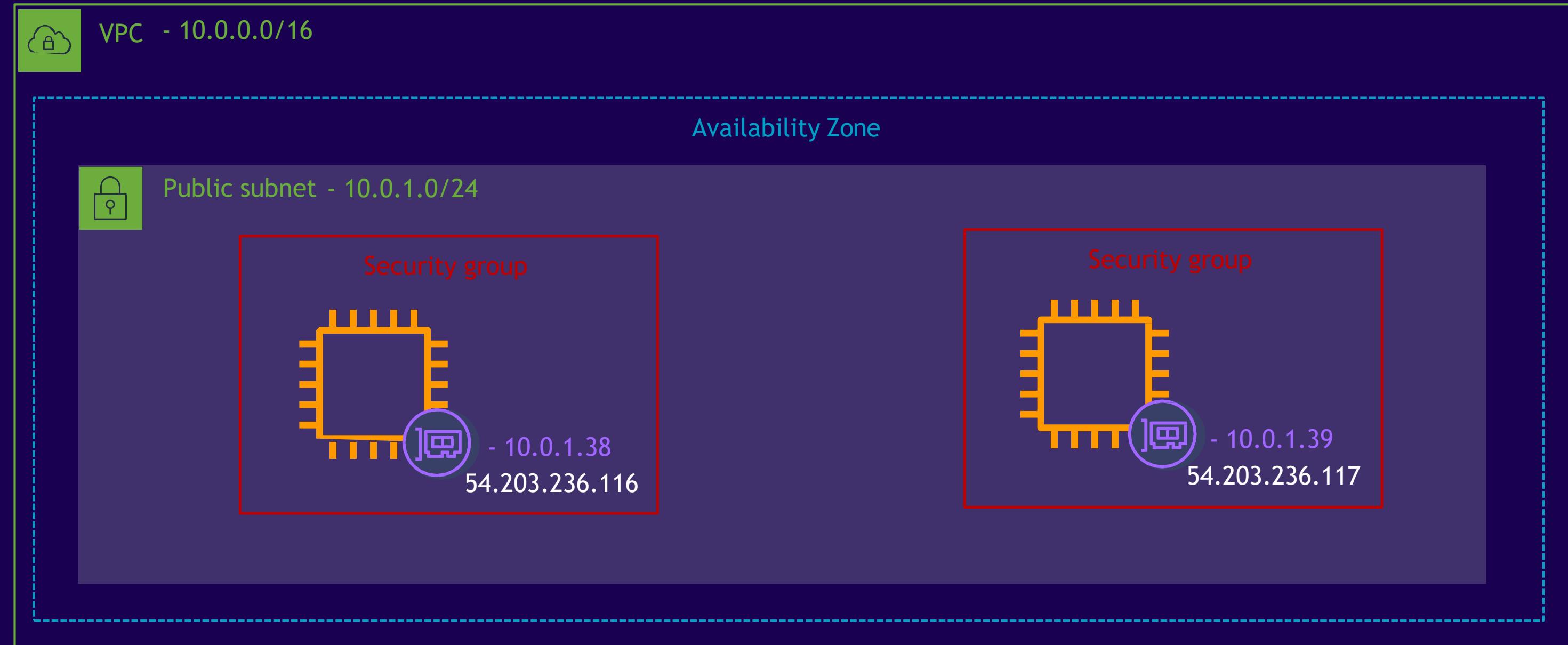


Basics of VPC security

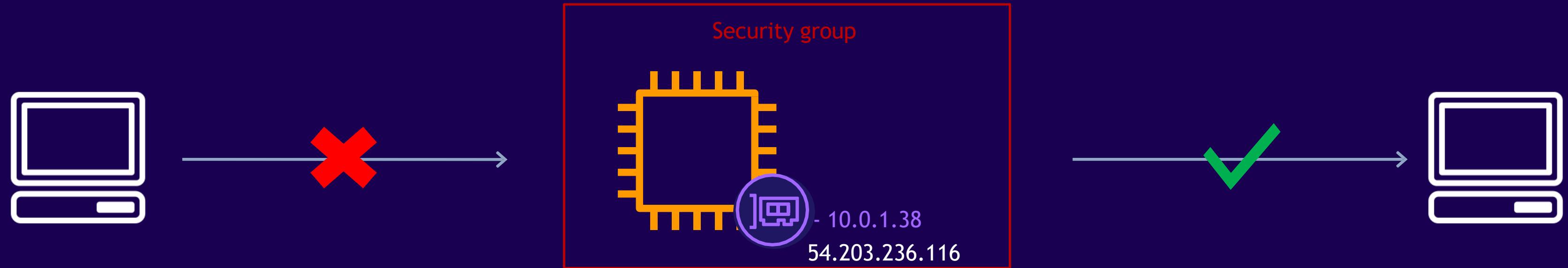
VPC defense in depth



Security groups



Security groups - default behavior

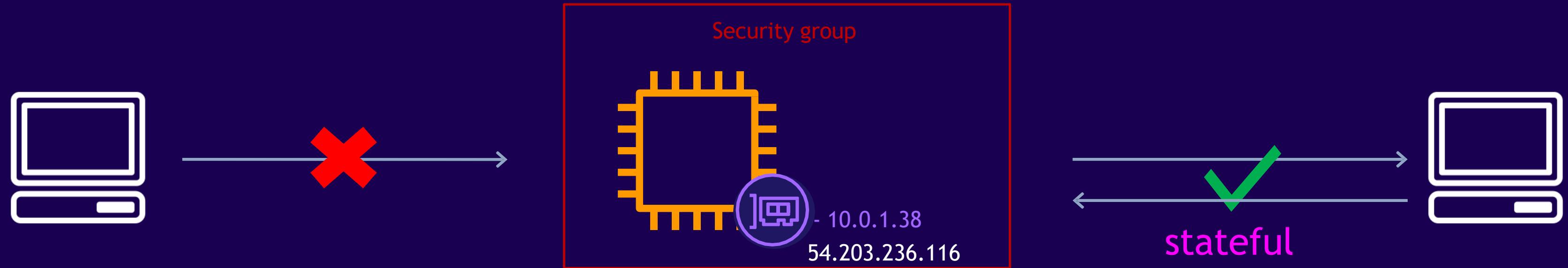


IP version	Type	Protocol	Port range	Source
No security group rules found				

IP version	Type	Protocol	Port range	Destination
IPv4	All traffic	All	All	0.0.0.0/0
IPv6	All traffic	TCP	All	::/0

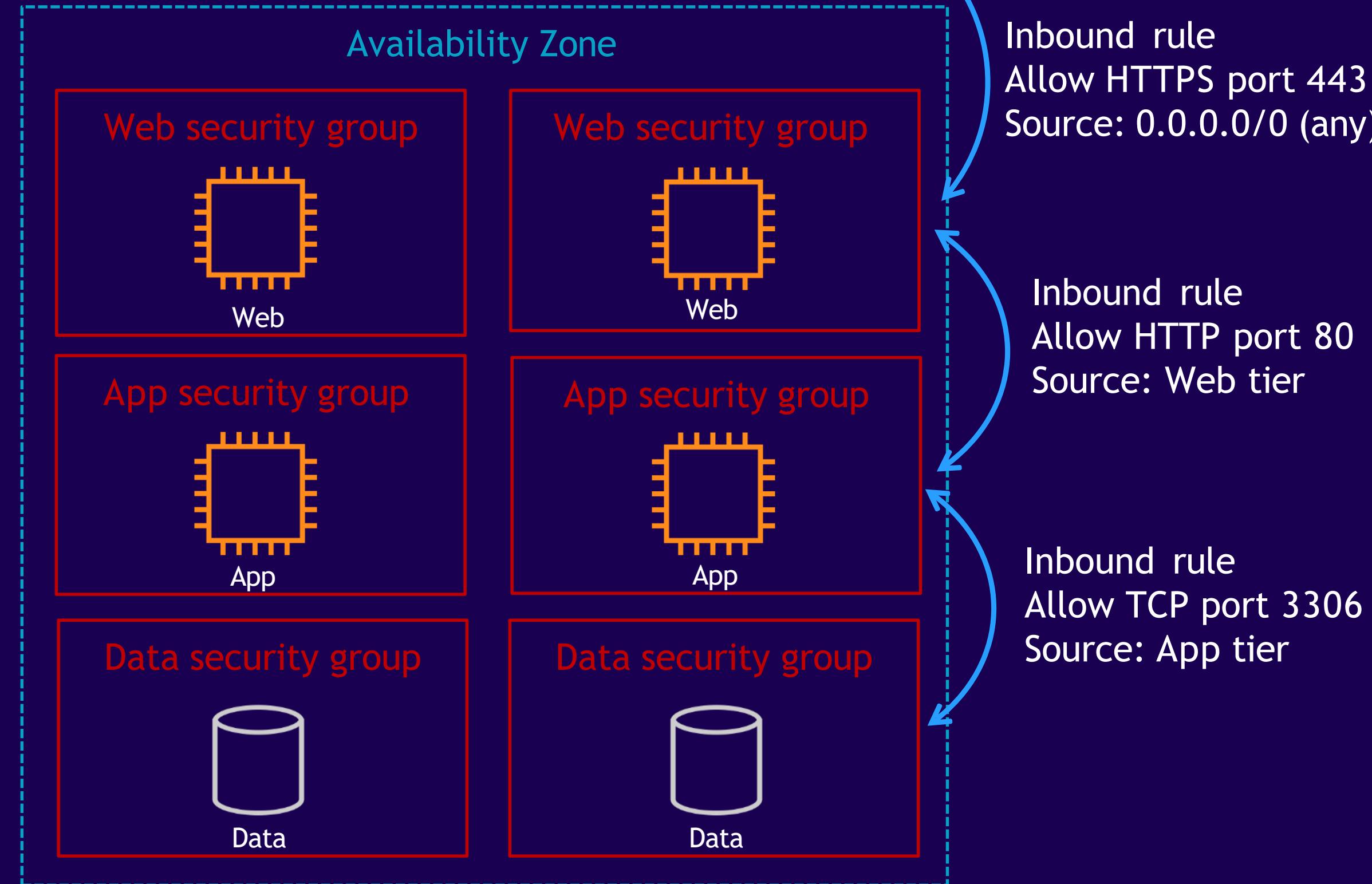
Outbound

Security groups - default behavior

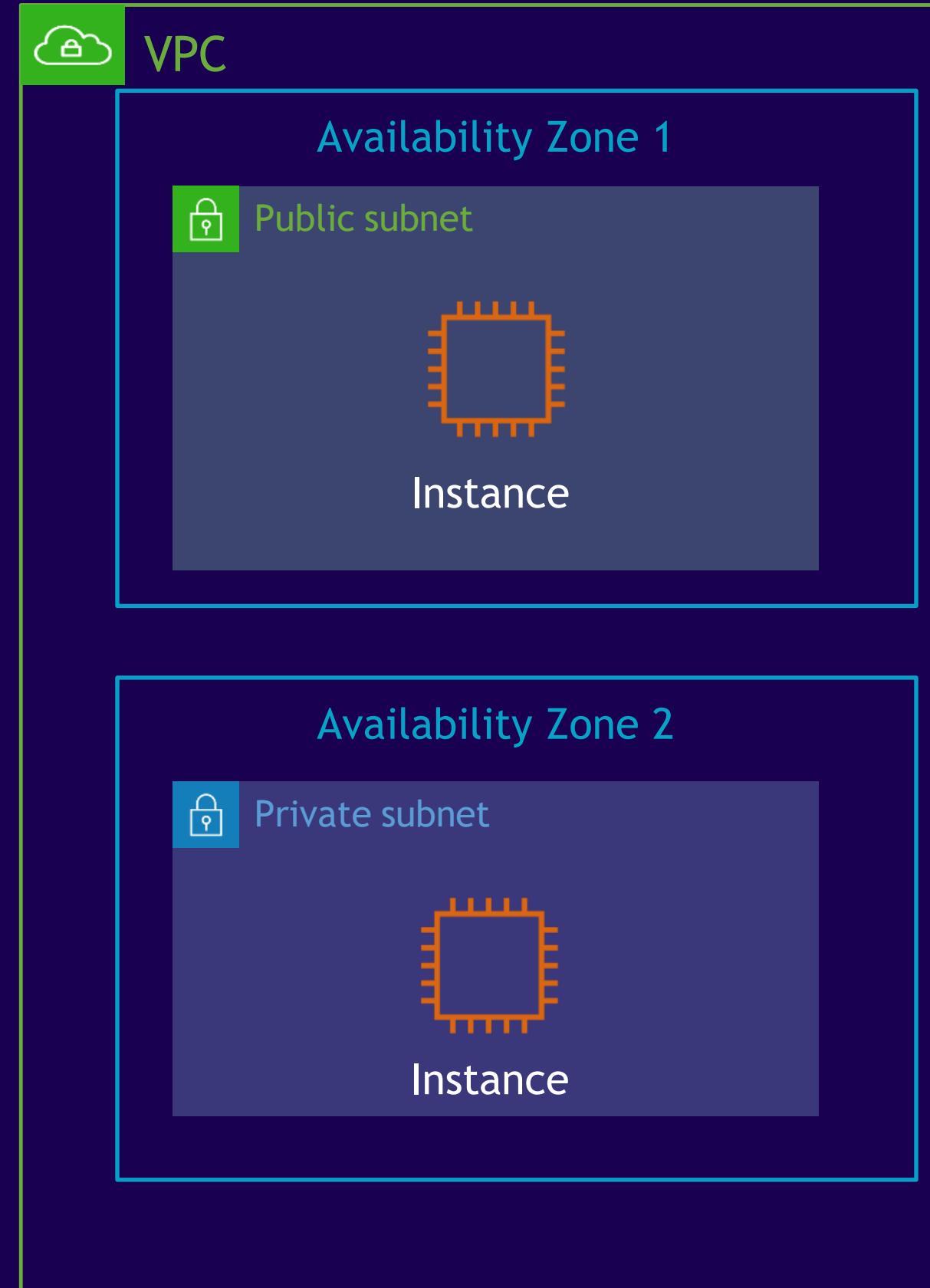


Inbound	IP version	Type	Protocol	Port range	Source
No security group rules found					
Outbound					
IP version	Type	Protocol	Port range	Destination	
IPv4	All traffic	All	All	0.0.0.0/0	
IPv6	All traffic	TCP	All	::/0	

Security Group Chaining



Network access control lists (NACLs)



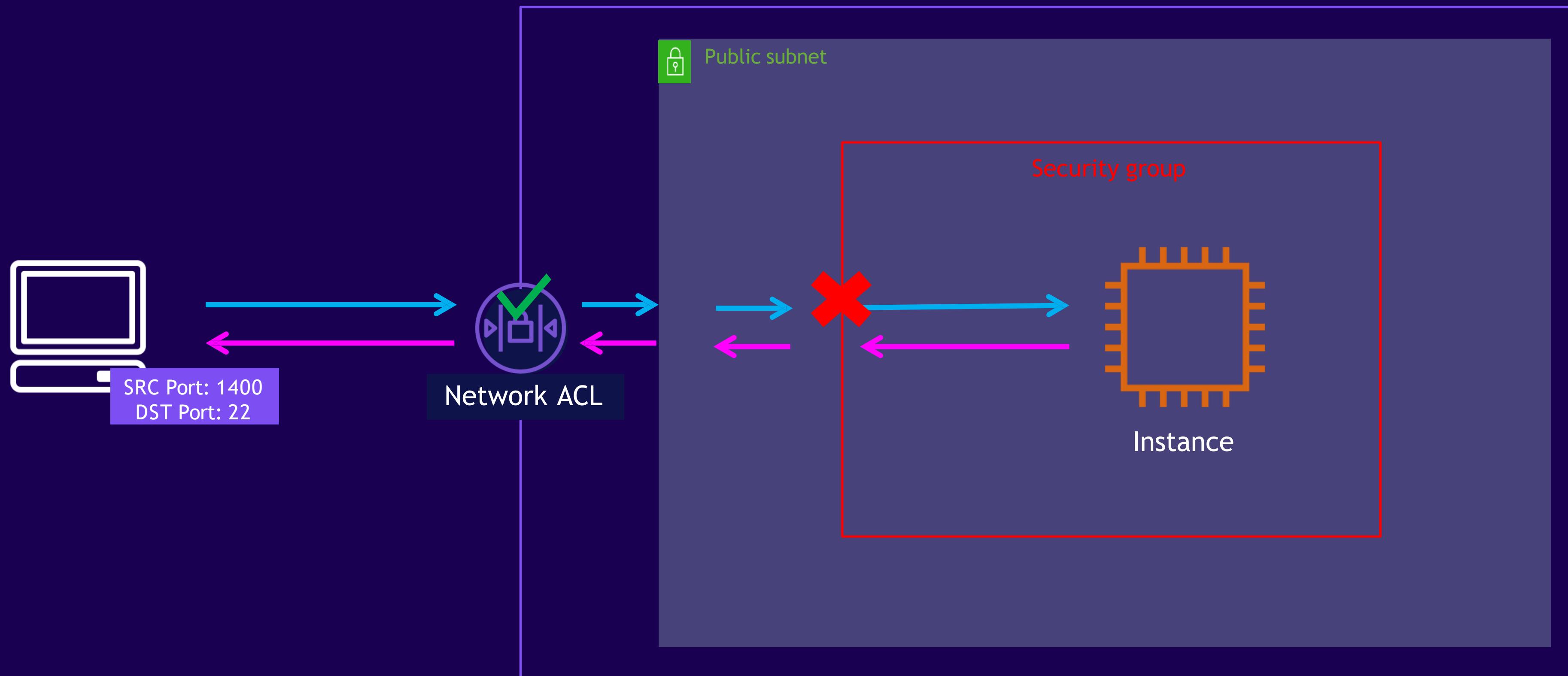
Inbound rules - default

Rule number	Type	Protocol	Port range	Source	Allow/Deny
100	All traffic	All	All	0.0.0.0/0	<input checked="" type="checkbox"/> Allow
101	All traffic	All	All	::/0	<input checked="" type="checkbox"/> Allow
*	All traffic	All	All	0.0.0.0/0	<input checked="" type="checkbox"/> Deny
*	All traffic	All	All	::/0	<input checked="" type="checkbox"/> Deny

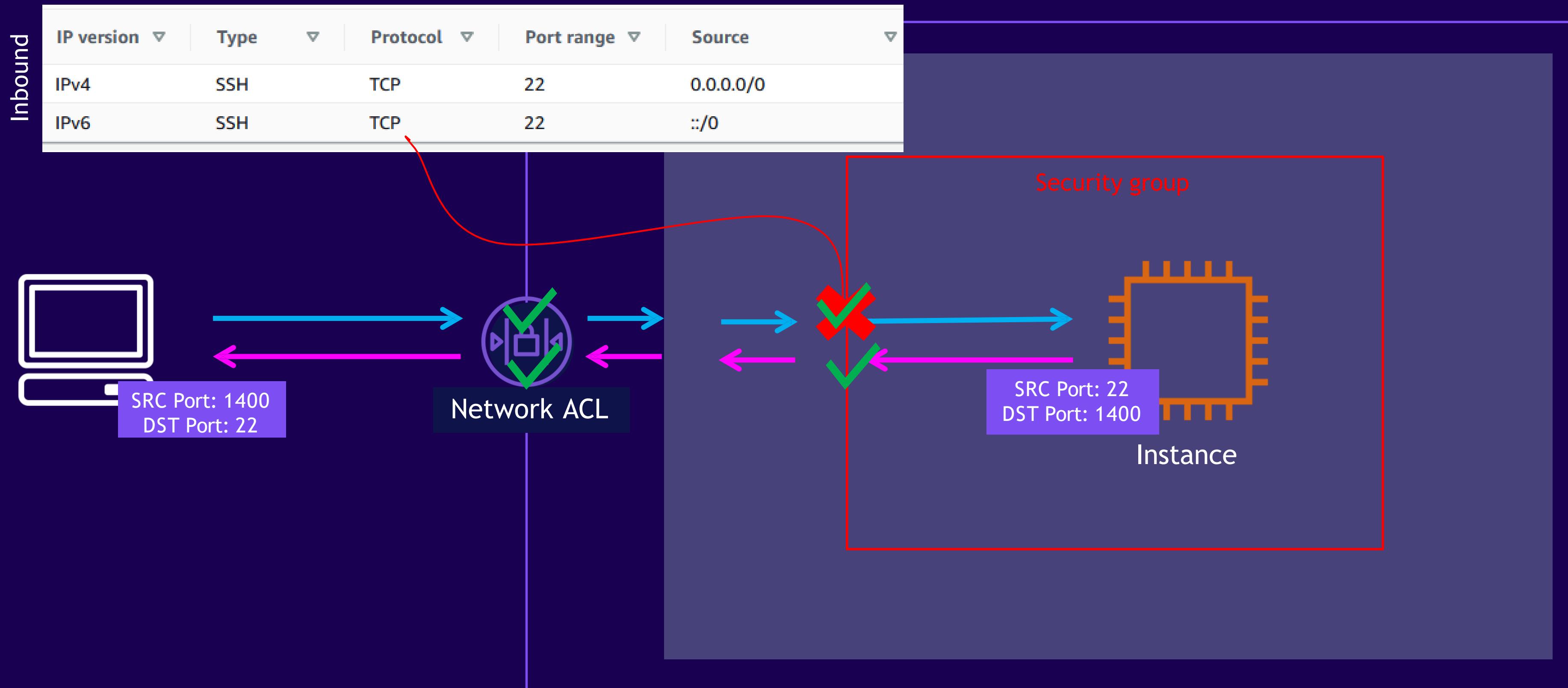
Outbound rules - default

Rule number	Type	Protocol	Port range	Destination	Allow/Deny
100	All traffic	All	All	0.0.0.0/0	<input checked="" type="checkbox"/> Allow
101	All traffic	All	All	::/0	<input checked="" type="checkbox"/> Allow
*	All traffic	All	All	0.0.0.0/0	<input checked="" type="checkbox"/> Deny
*	All traffic	All	All	::/0	<input checked="" type="checkbox"/> Deny

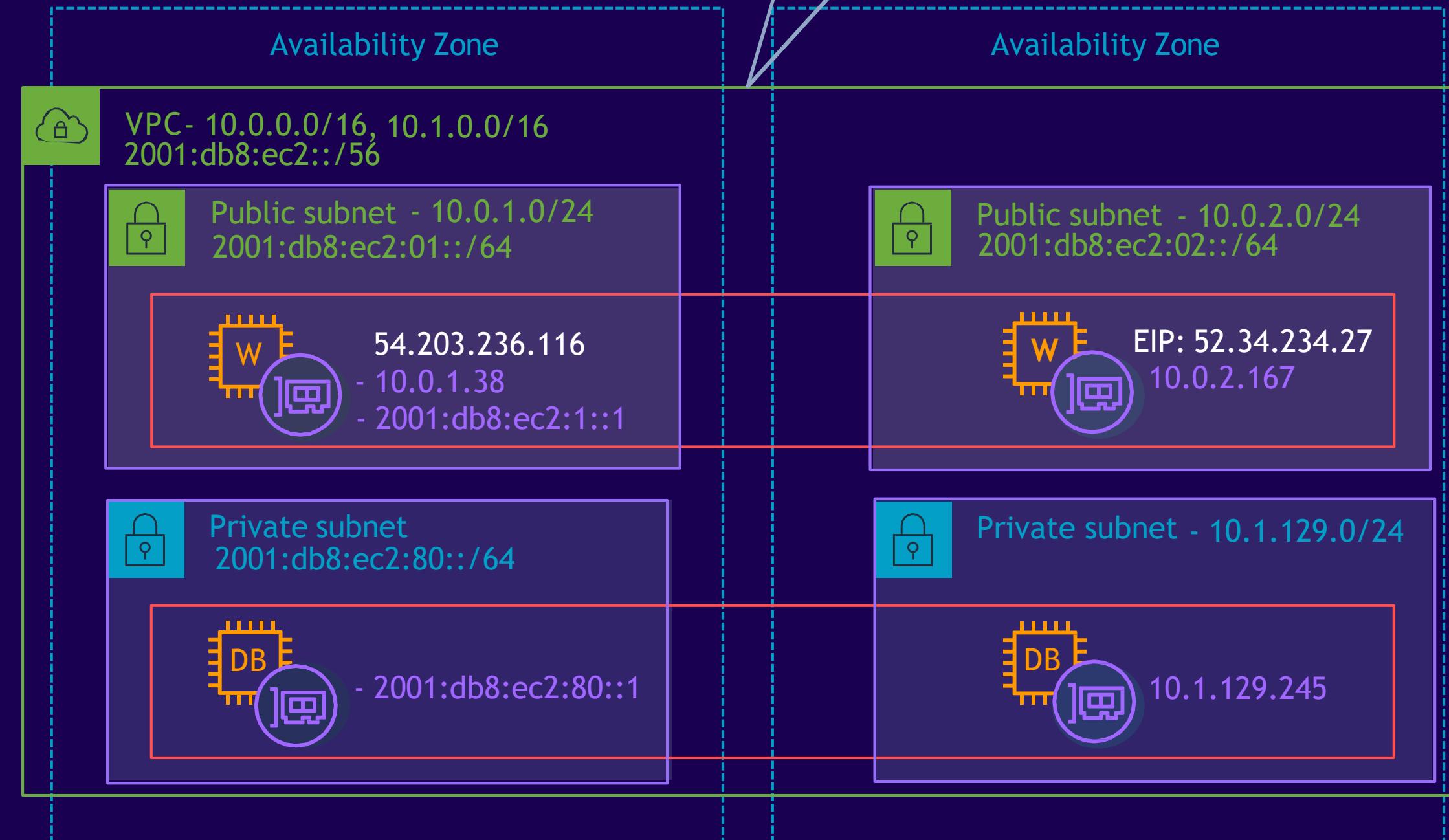
Additional configurations for inbound traffic



Additional configurations for inbound traffic



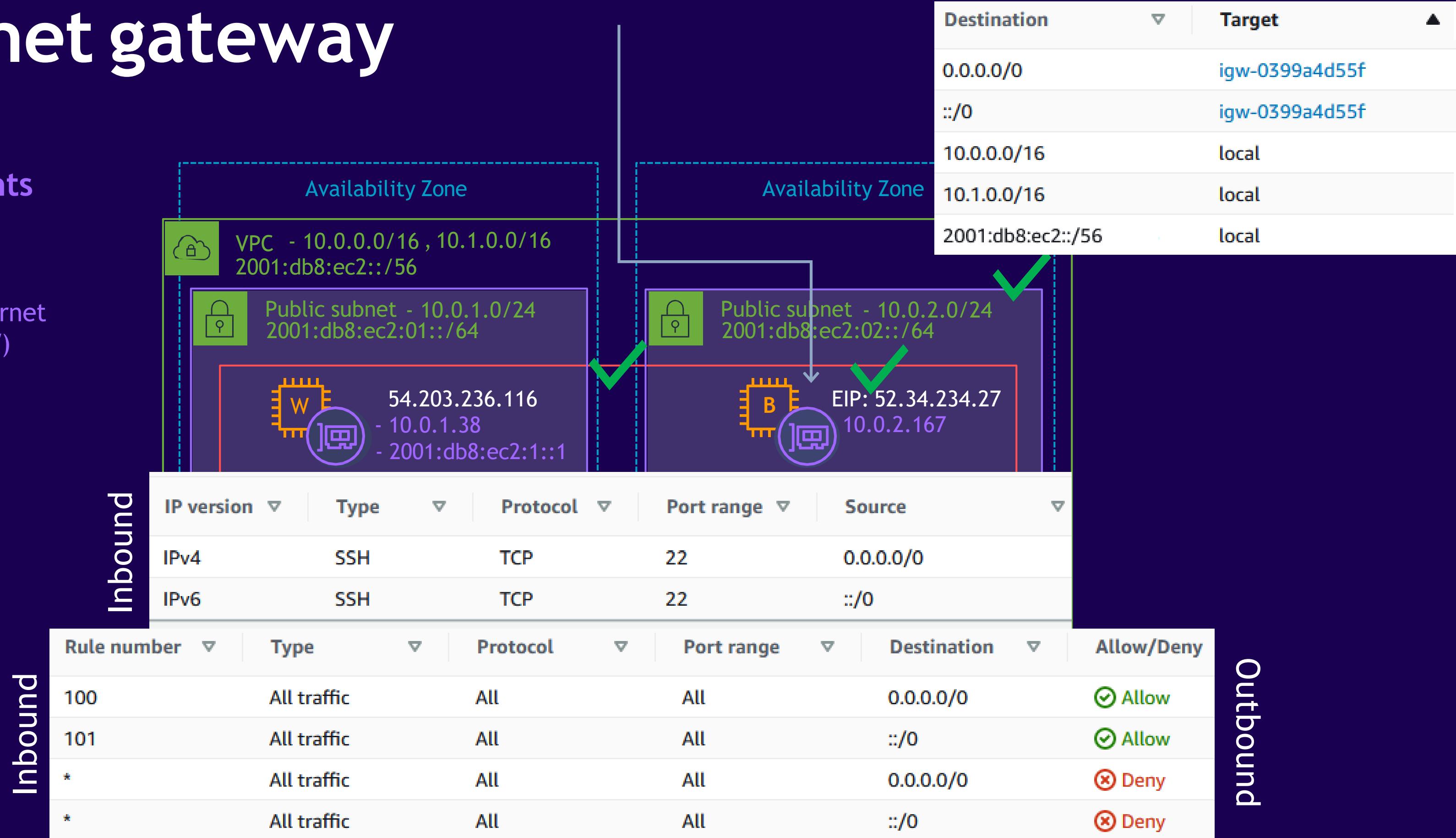
Internet gateway



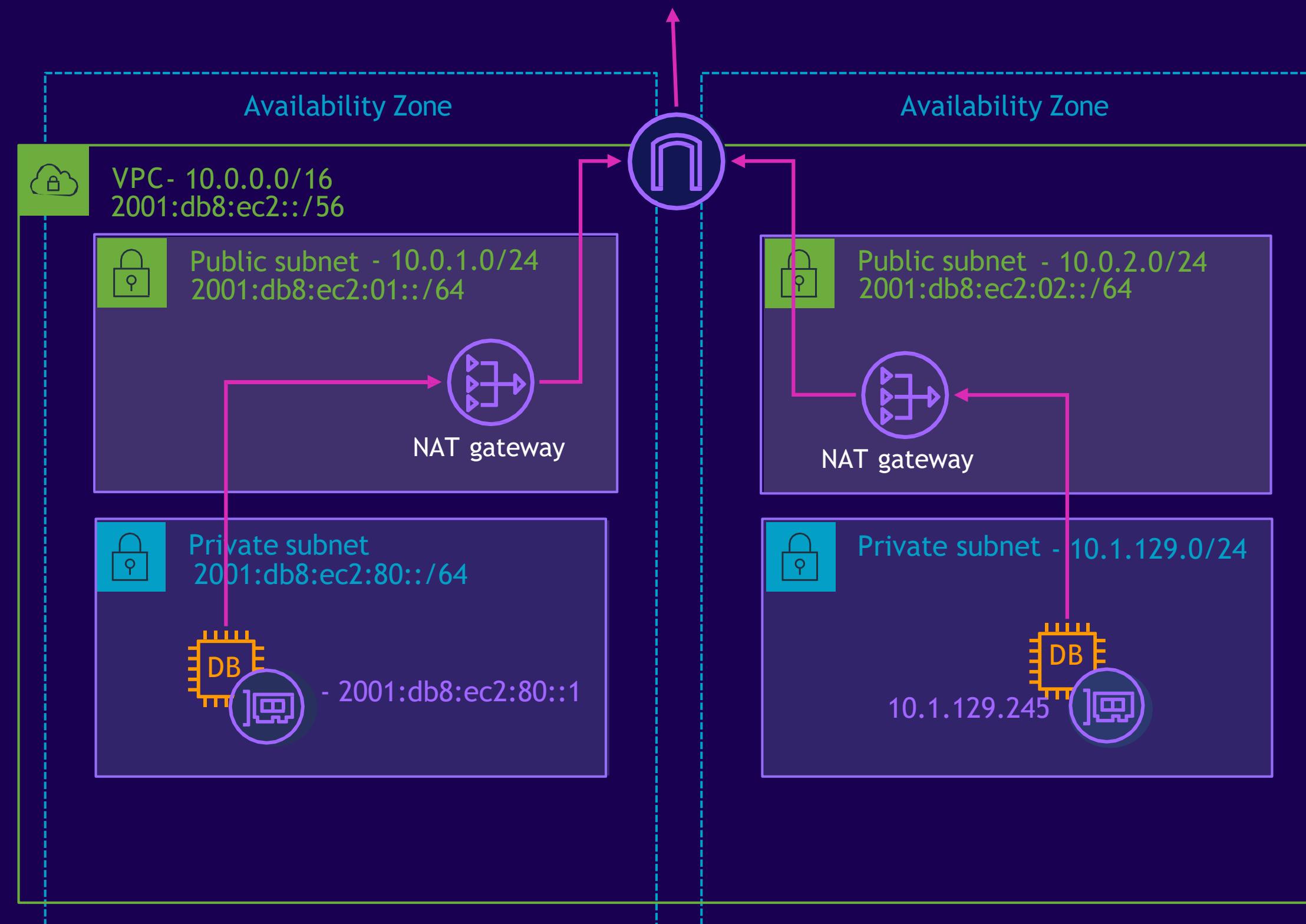
Internet gateway

5 Requirements

- 1) Public IP
- 2) SGs allow
- 3) NACLs allow
- 4) Attached internet gateway (IGW)
- 5) Route to IGW



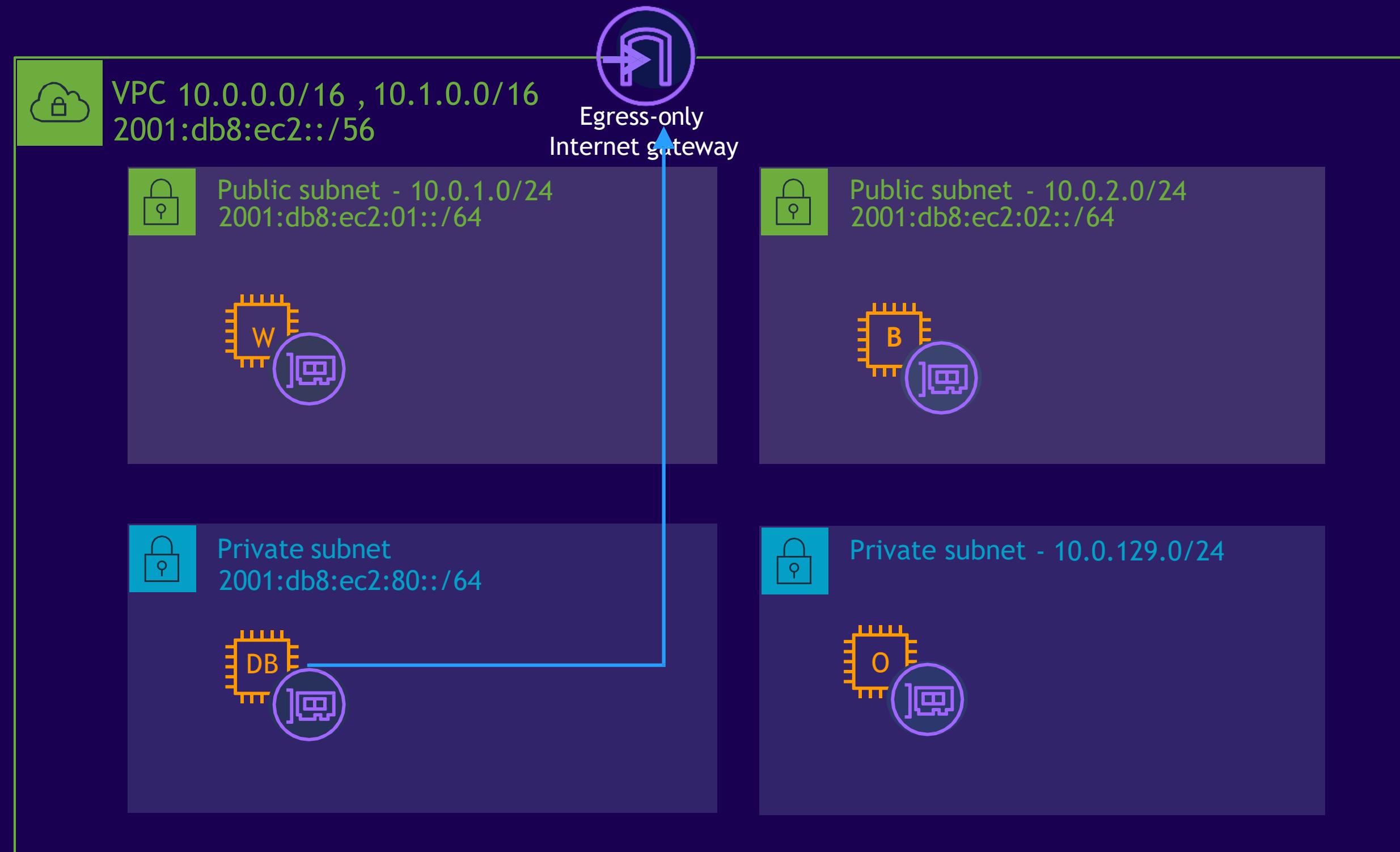
Connecting to the internet from private subnet



Destination	Target
0.0.0.0/0	igw-0399a4d55f
::/0	igw-0399a4d55f
10.0.0.0/16	local
10.1.0.0/16	local
2001:db8:ec2::/56	local

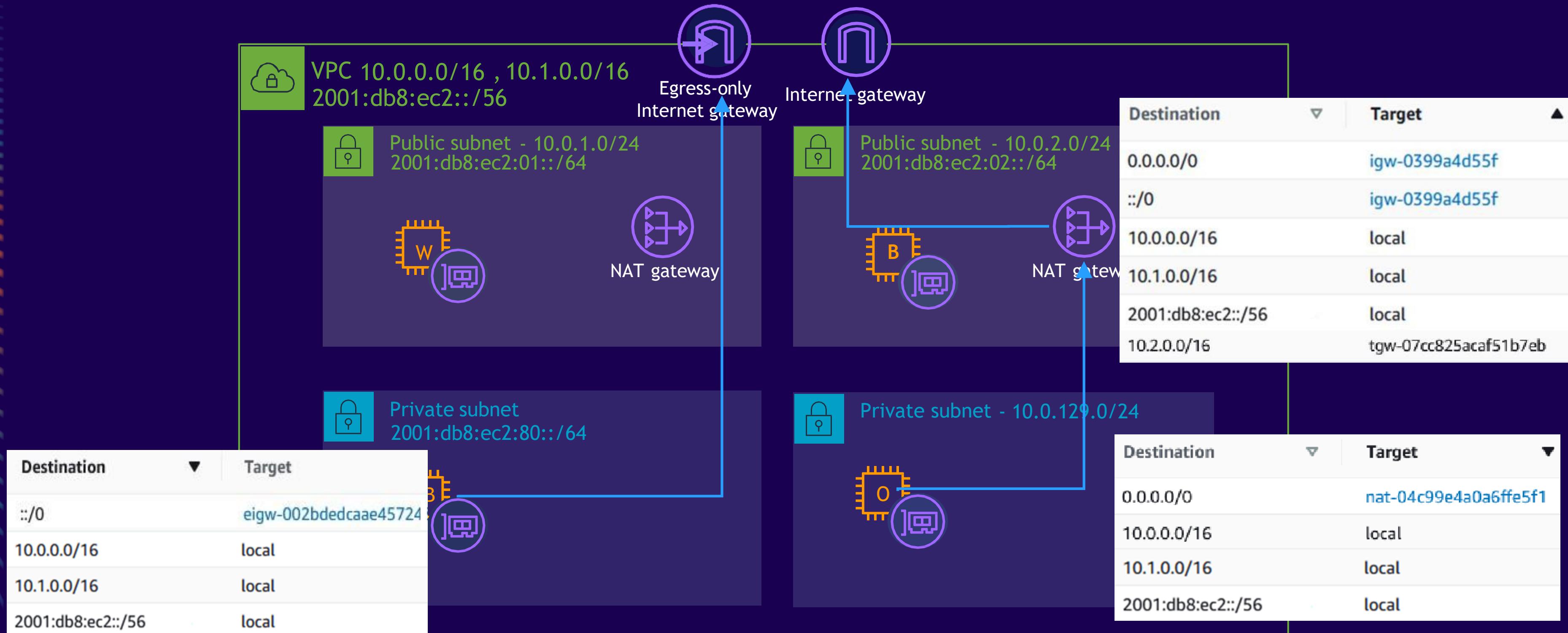
Destination	Target
0.0.0.0/0	nat-04c99e4a0a6ffe5f1
10.0.0.0/16	local
10.1.0.0/16	local
2001:db8:ec2::/56	local

Connecting to the internet: IPv6



Destination	Target
::/0	eigw-002bdedcaae45724
10.0.0.0/16	local
10.1.0.0/16	local
2001:db8:ec2::/56	local

Connecting to the internet



Internet access



172.16.0.0
172.16.1.0
172.16.2.0

Create internet gateway		Actions ▾	
<input type="text"/> Filter by tags and attributes or search by keyword			
Name	ID	State	VPC
	igw-09ef761d872bd7540	attached	vpc-0bcb5110cf0c...

Destination	Target	Status	Propagated
172.31.0.0/16	local	Active	No
2600:1f16:14d:6300::/56	local	Active	No
0.0.0.0/0	igw-09ef761d872bd7540	Active	No
::/0	igw-09ef761d872bd7540	Active	No

"To get to the IPv4 Internet (0.0.0.0/0) go via the Internet Gateway (IGW)"

"To get to the IPv6 Internet (::/0) go via the Internet Gateway (IGW)"

Internet access



172.16.0.0
172.16.1.0
172.16.2.0

Create Egress Only Internet Gateway Delete

Filter by attributes or search by keyword

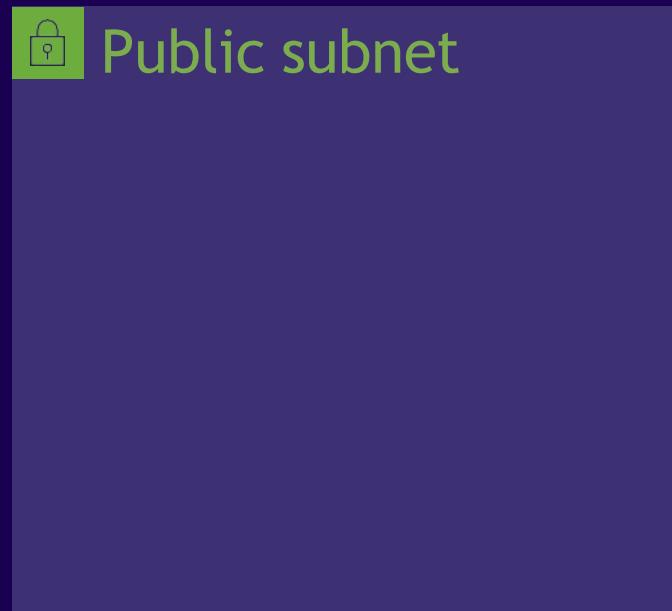
ID	VPC
eigw-063d49ed7b...	vpc-0c05afa3bd855...

Destination	Target	Status	Propagated
172.31.0.0/16	local	Active	No
2600:1f16:14d:6300::/56	local	Active	No
0.0.0.0/0	igw-09ef761d872bd7540	Active	No
::/0	eigw-063d49ed7bb0f8c36	Active	No

“To get to the IPv6 Internet (::/0) go via the Egress Only Internet Gateway (EIGW)”

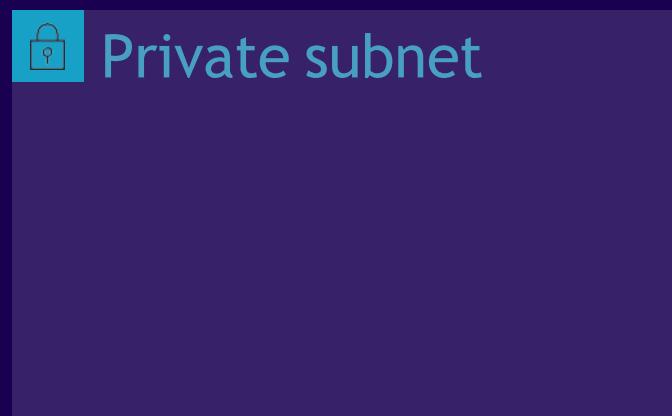
Different routes for different subnets

172.16.0.0
172.16.1.0
172.16.2.0



Destination	Target	Status	Propagated
172.31.0.0/16	local	Active	No
2600:1f16:14d:6300::/56	local	Active	No
0.0.0.0/0	igw-09ef761d872bd7540	Active	No
::/0	igw-09ef761d872bd7540	Active	No

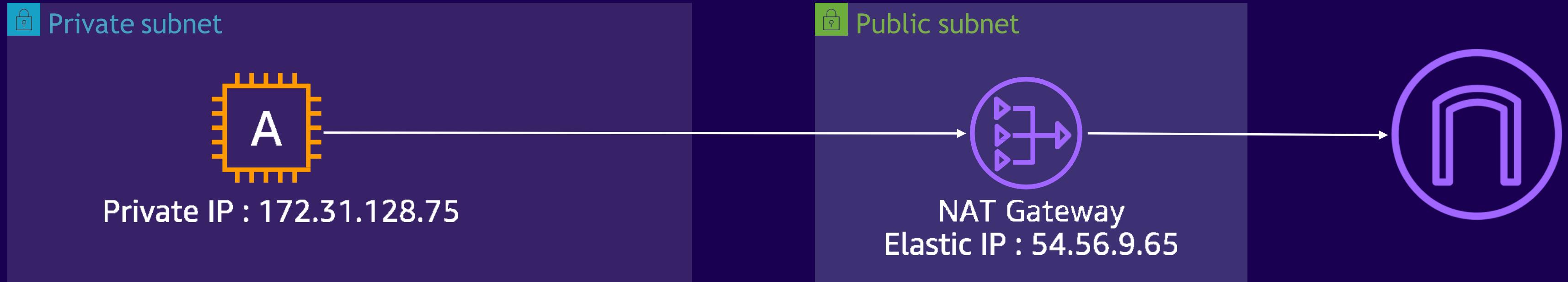
“To get to the Internet go via the Internet Gateway (IGW)”



Destination	Target	Status	Propagated
172.31.0.0/16	local	Active	No
2600:1f16:14d:6300::/56	local	Active	No

“To get to anything in the VPC – stay local. No route anywhere else.”

Network Address Translation (NAT) Gateway



Destination	Target	Status	Propagated
172.31.0.0/16	local	Active	No
0.0.0.0/0	nat-0964c62a07d6491f5	Active	No

Destination	Target	Status	Propagated
172.31.0.0/16	local	Active	No
2600:1f16:14d:6300::/56	local	Active	No
0.0.0.0/0	igw-09ef761d872bd7540	Active	No
::/0	igw-09ef761d872bd7540	Active	No

The Route Table for the Private Subnet says to send all IPv4 Internet Traffic to the NAT Gateway.

The NAT Gateway translates all traffic it receives such that it appears to come from itself.

The Route Table for the Public Subnet says to send all Internet Traffic to the Internet Gateway.



What's a route?

What's a route?

IPv4 and IPv6 network destination

Next-hop IP or interface to reach the destination

- ▼ Get on US-101 S in Alto
11 min (4.9 mi)
- ↑ Head southwest on Paradise Dr toward Mar W St
0.4 mi
- ↻ At the traffic circle, continue straight onto CA-131 E
4.4 mi
- ↗ Slight right to merge onto US-101 S
0.1 mi

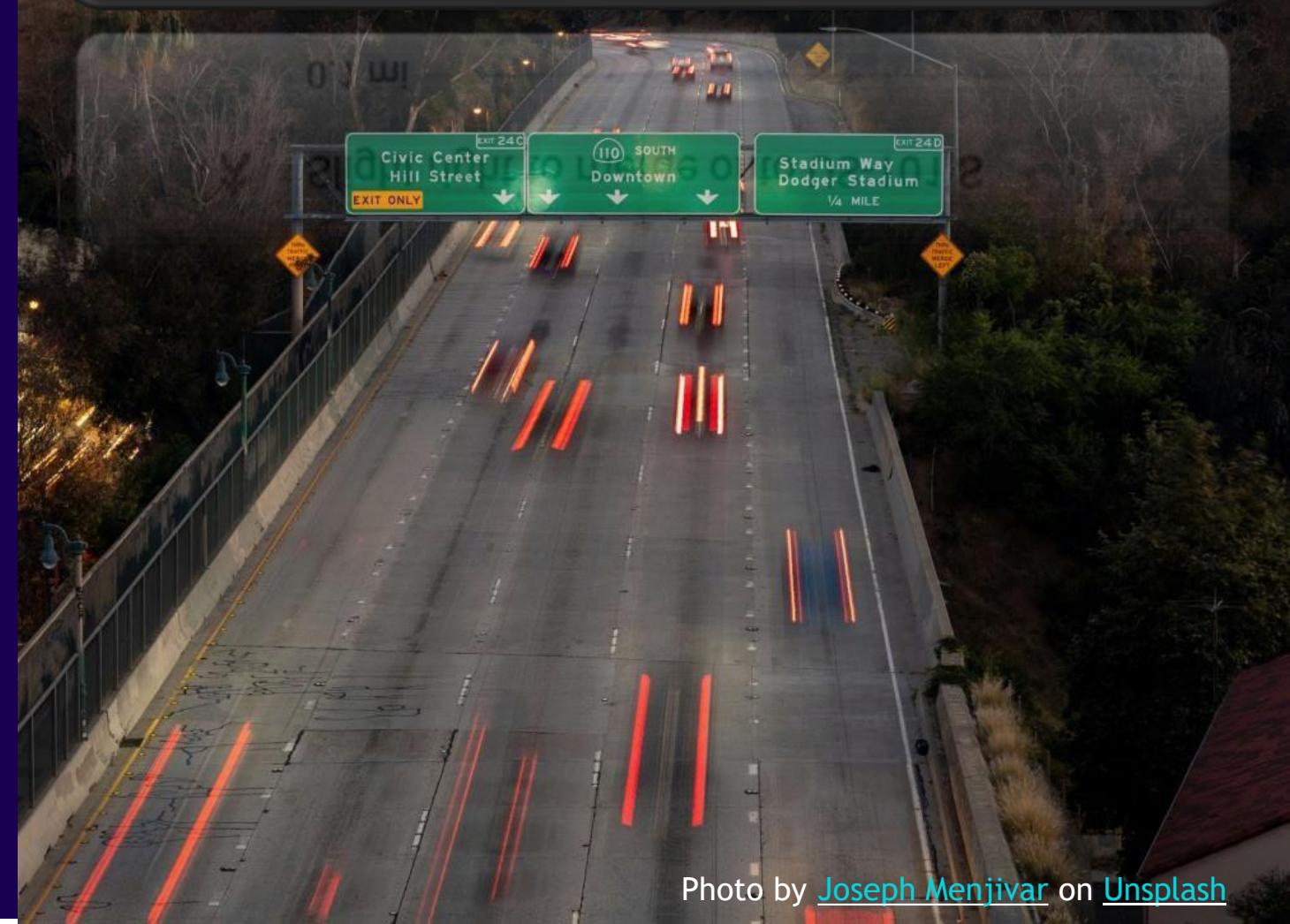


Photo by [Joseph Menjivar](#) on [Unsplash](#)

What's a route?

IPv4 and IPv6 network destination

203.0.113.0 /24

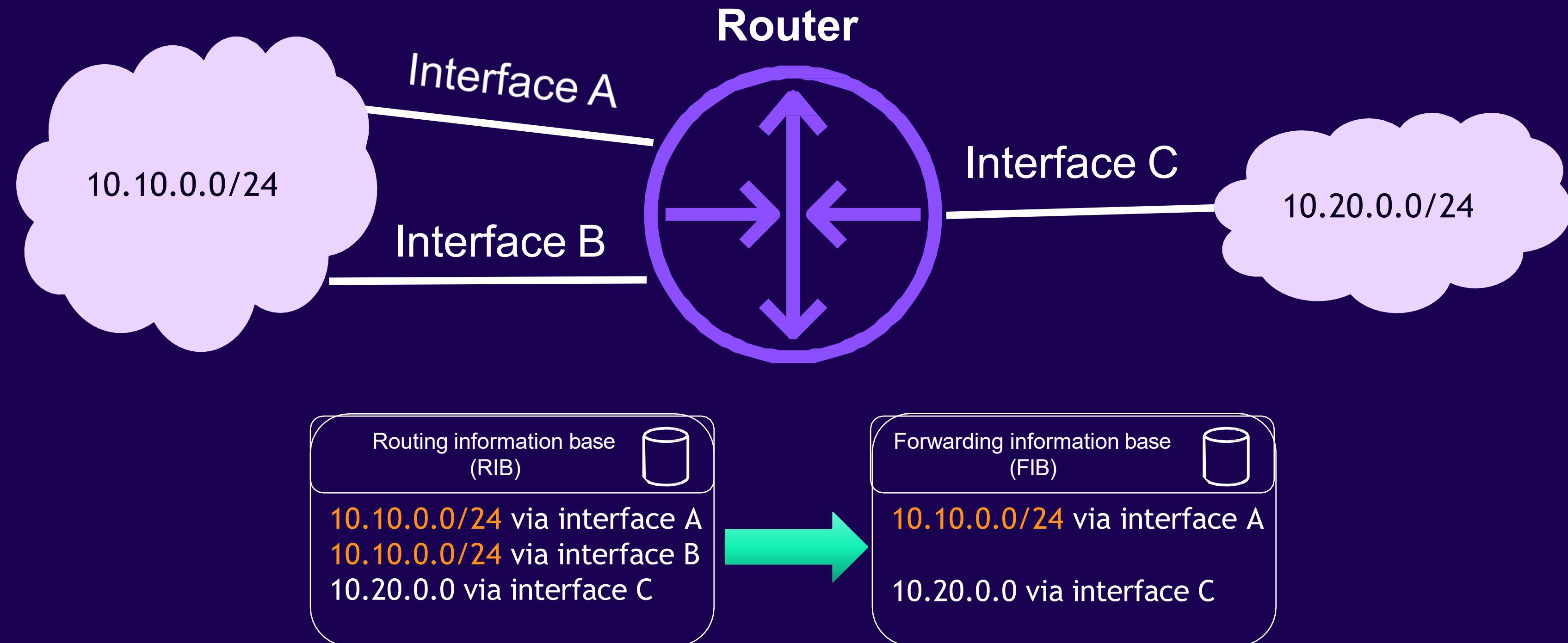
2001:db8:: /64

Next-hop IP or interface to reach the destination

via interface eth0

via 2001:db8::1

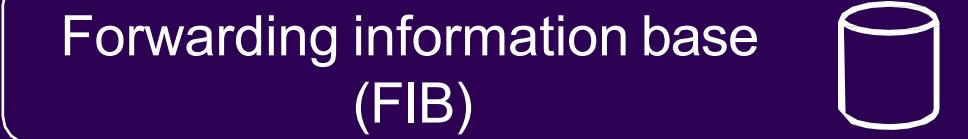
Routing vs. forwarding information bases



Routing vs. forwarding



All the routes



Best path(s) for each destination



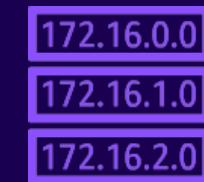
AWS networking

AWS routing participants

Virtual data center

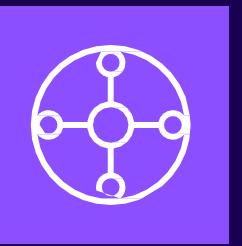


Amazon Virtual Private Cloud
(Amazon VPC)

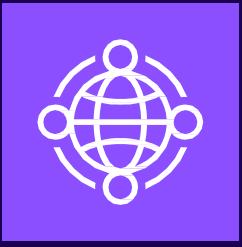


Route table

Global connectivity



AWS Transit Gateway



AWS Cloud WAN

Hybrid connectivity



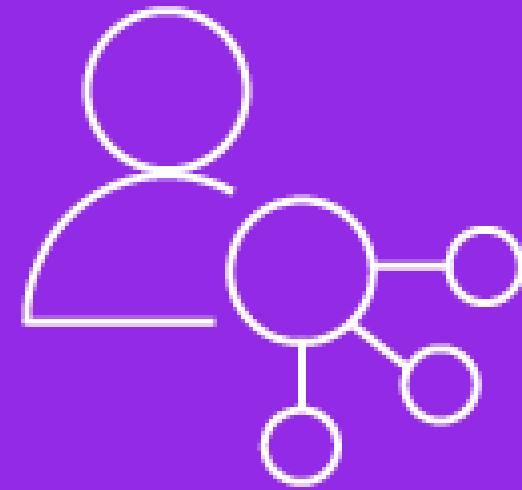
AWS Direct Connect



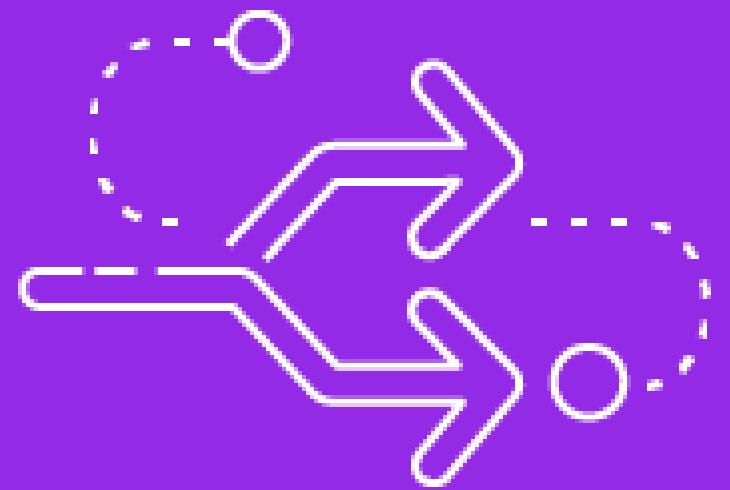
AWS Site-to-Site VPN

AWS common route types

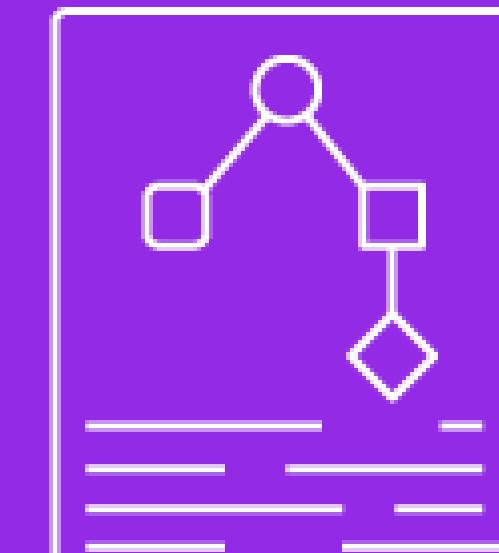
Static



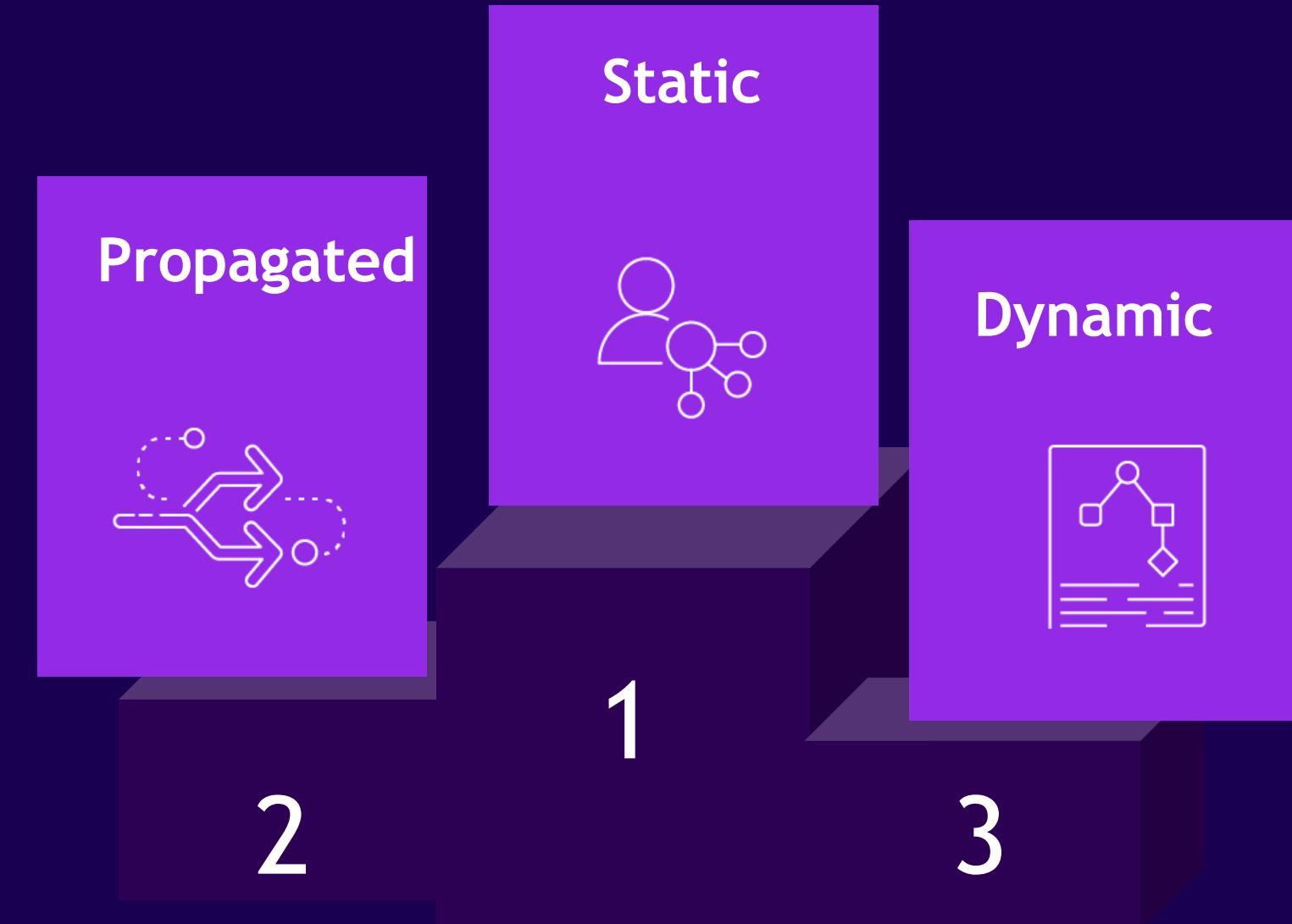
Propagated



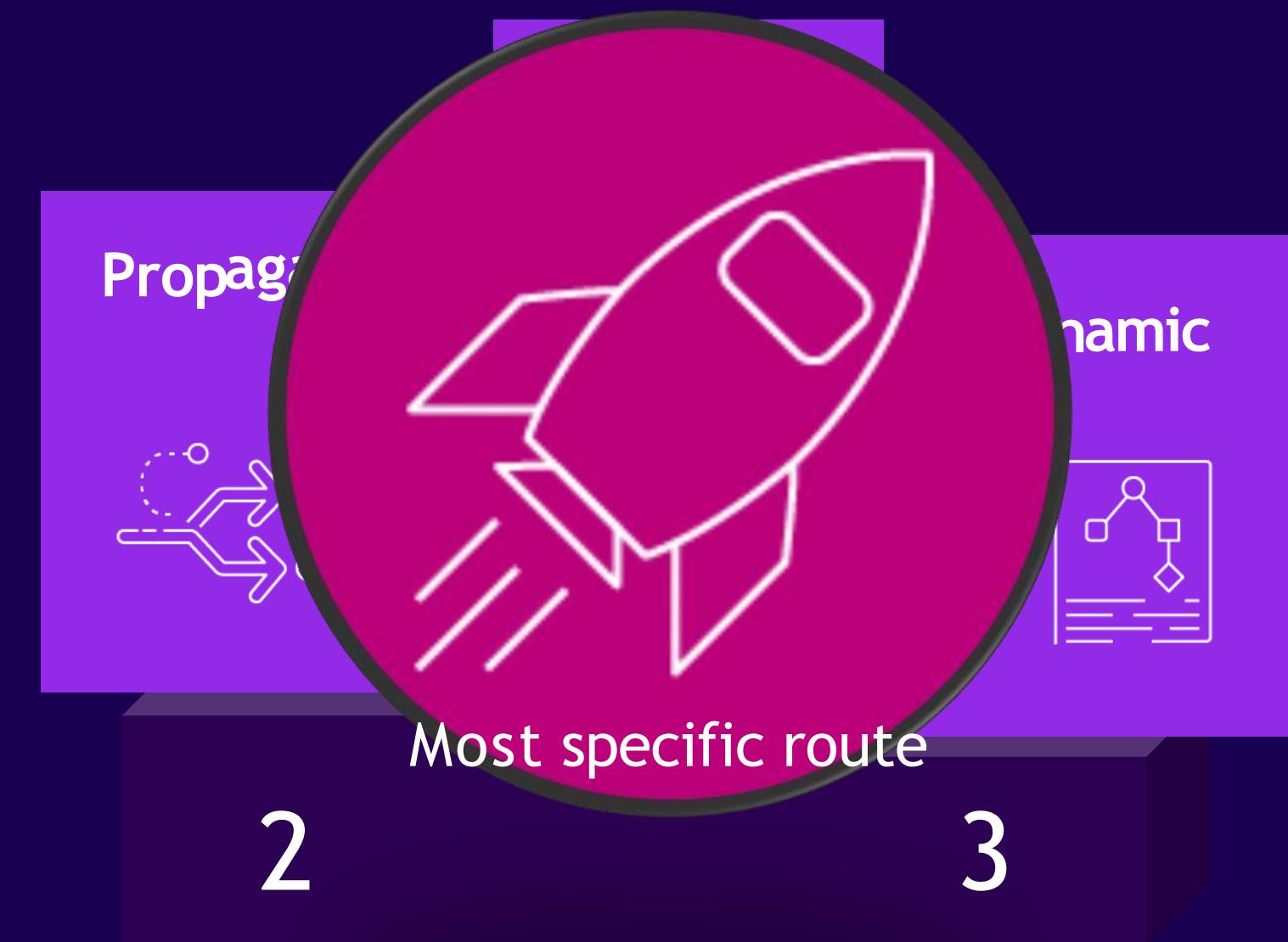
Dynamic



Route preference

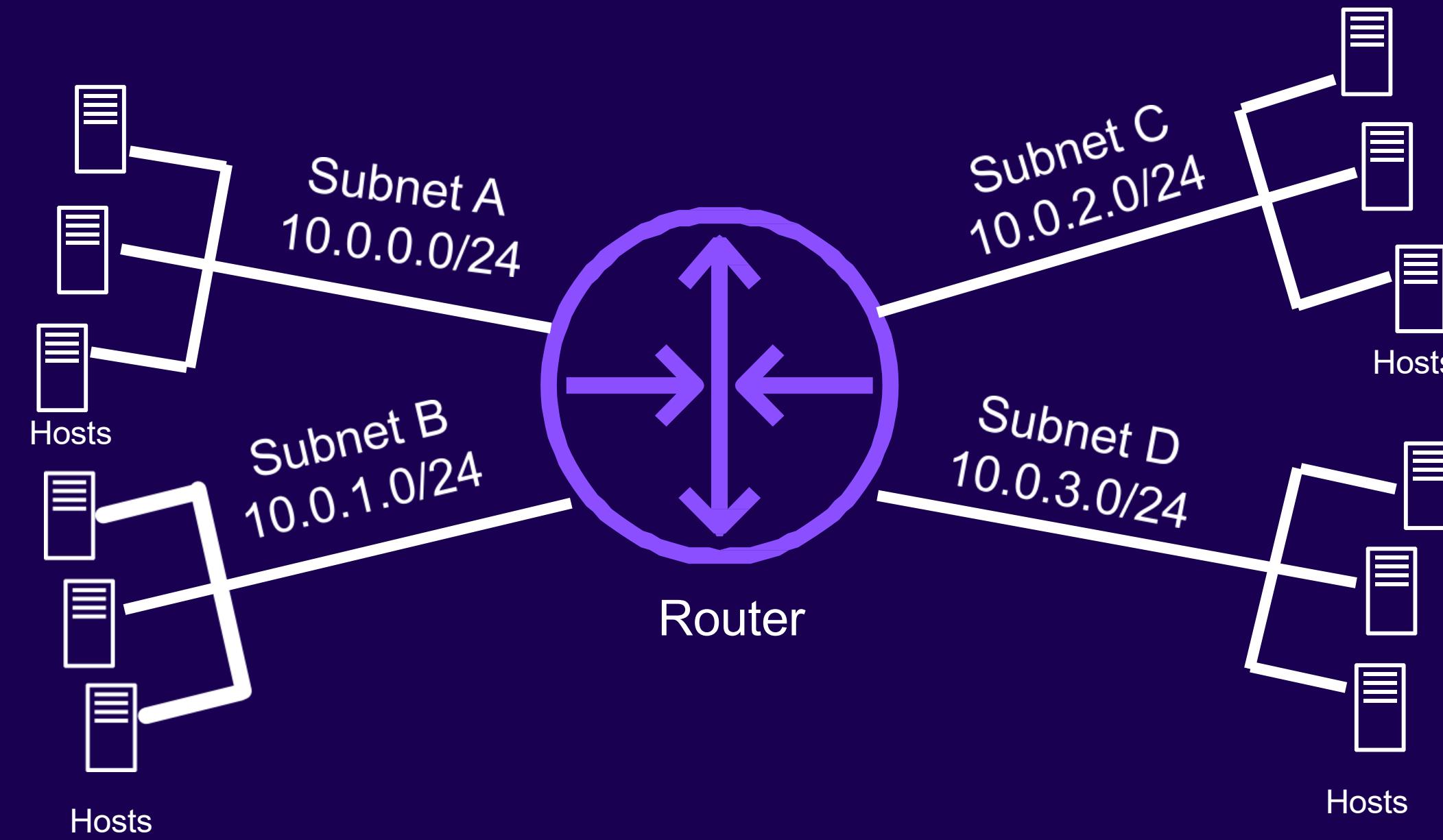


Route preference

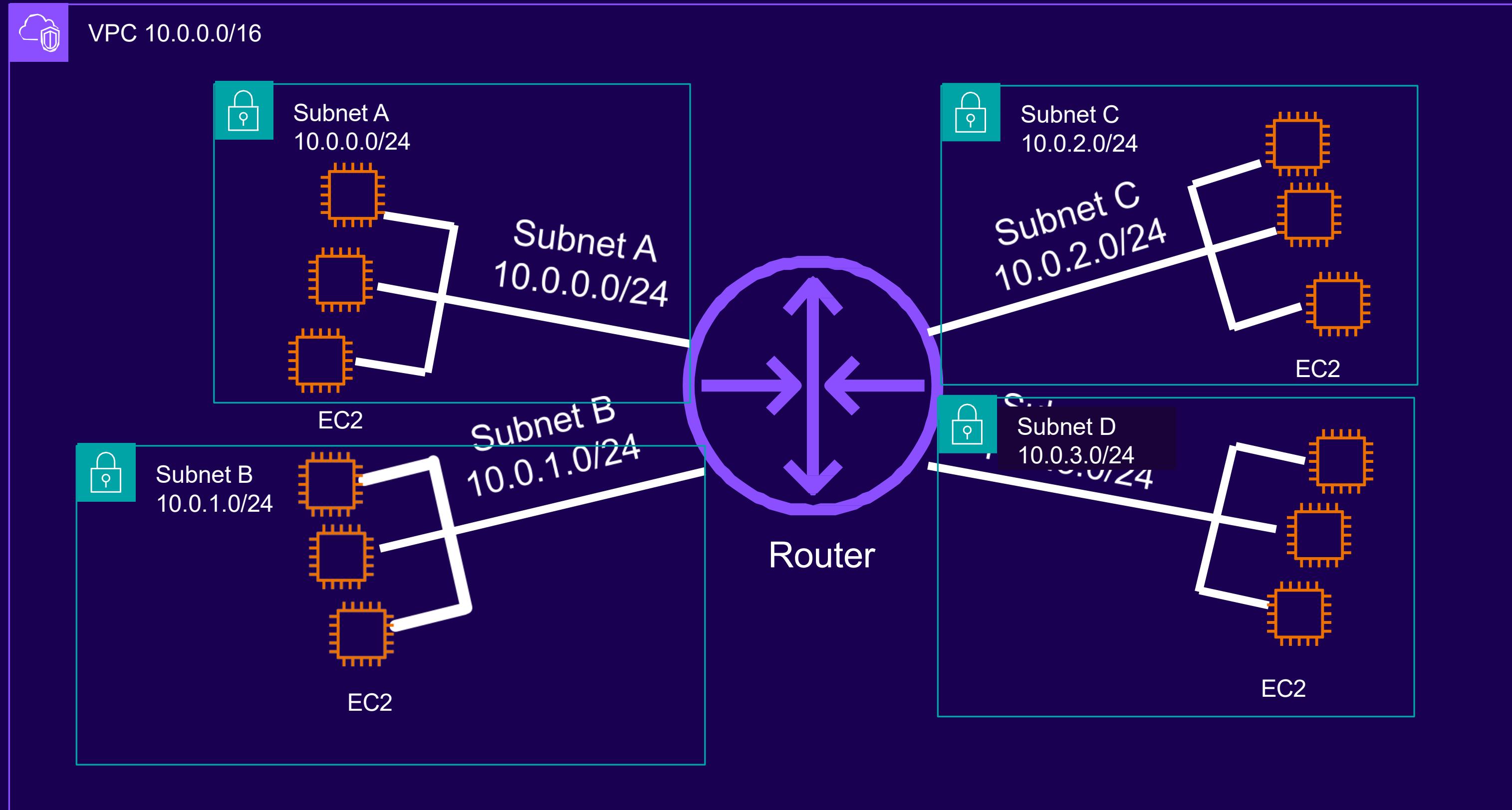


Routing scenarios: Single VPC, single AWS Region

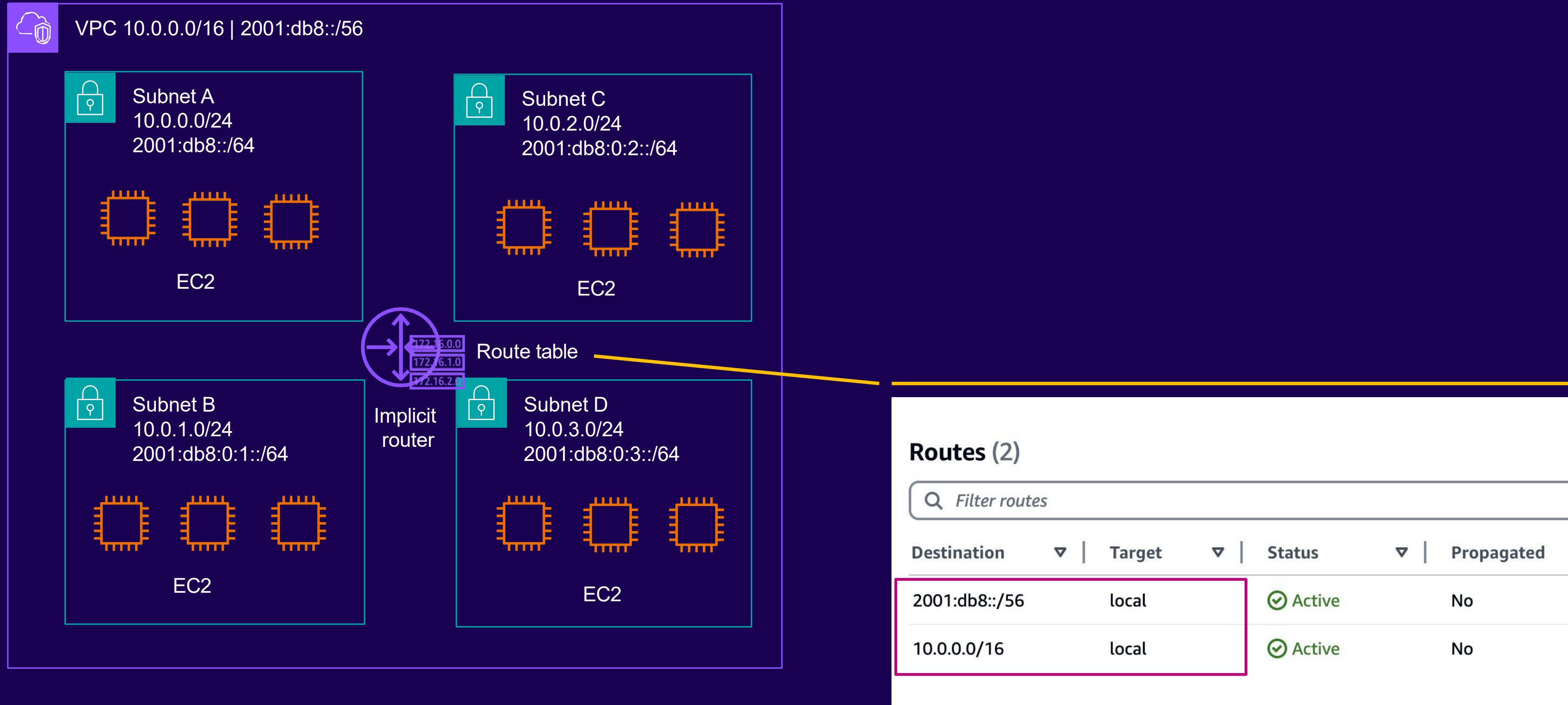
Subnets - On-premises vs. AWS



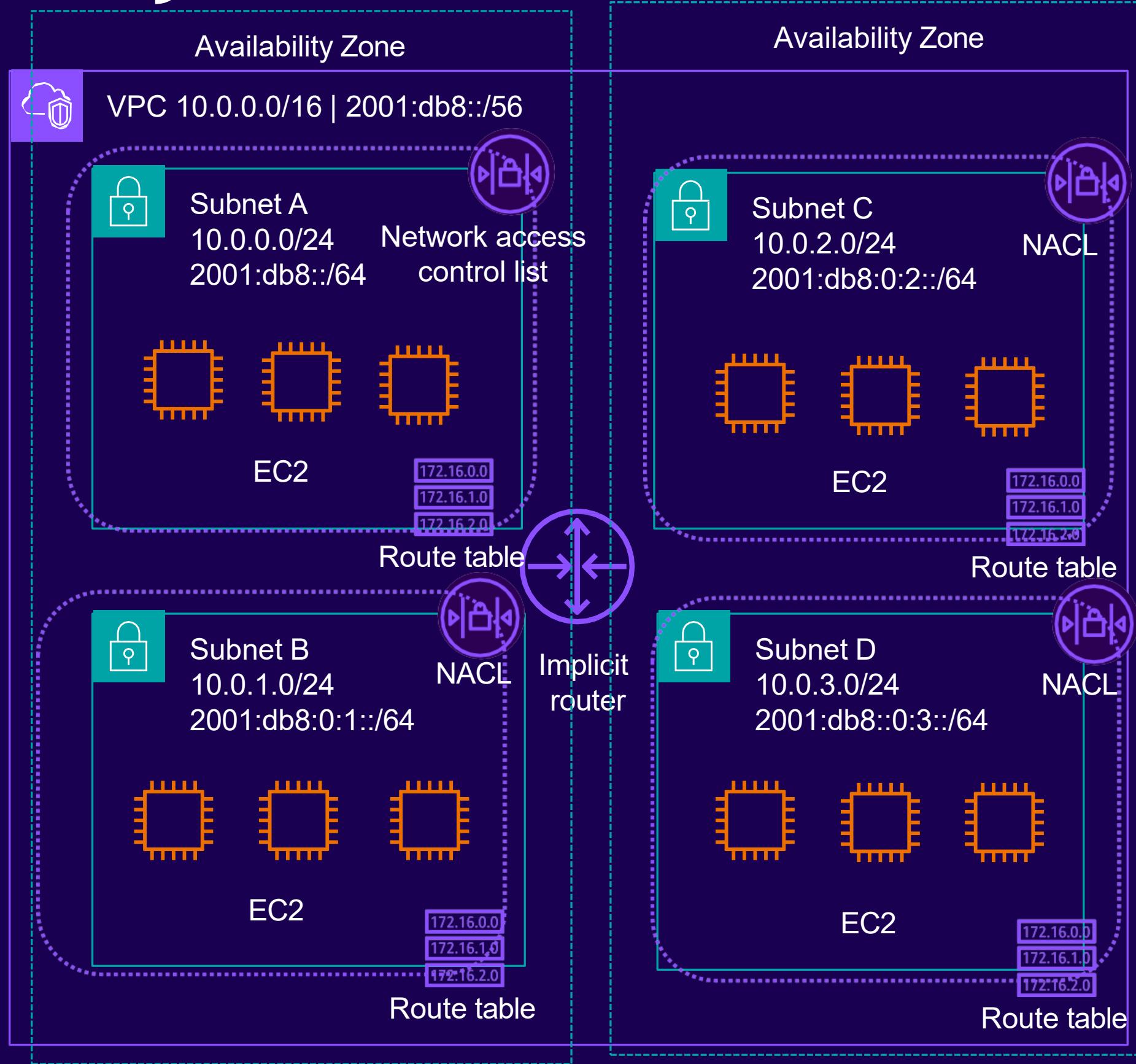
Subnets - On-premises vs. AWS



Subnets - On-premises vs. AWS

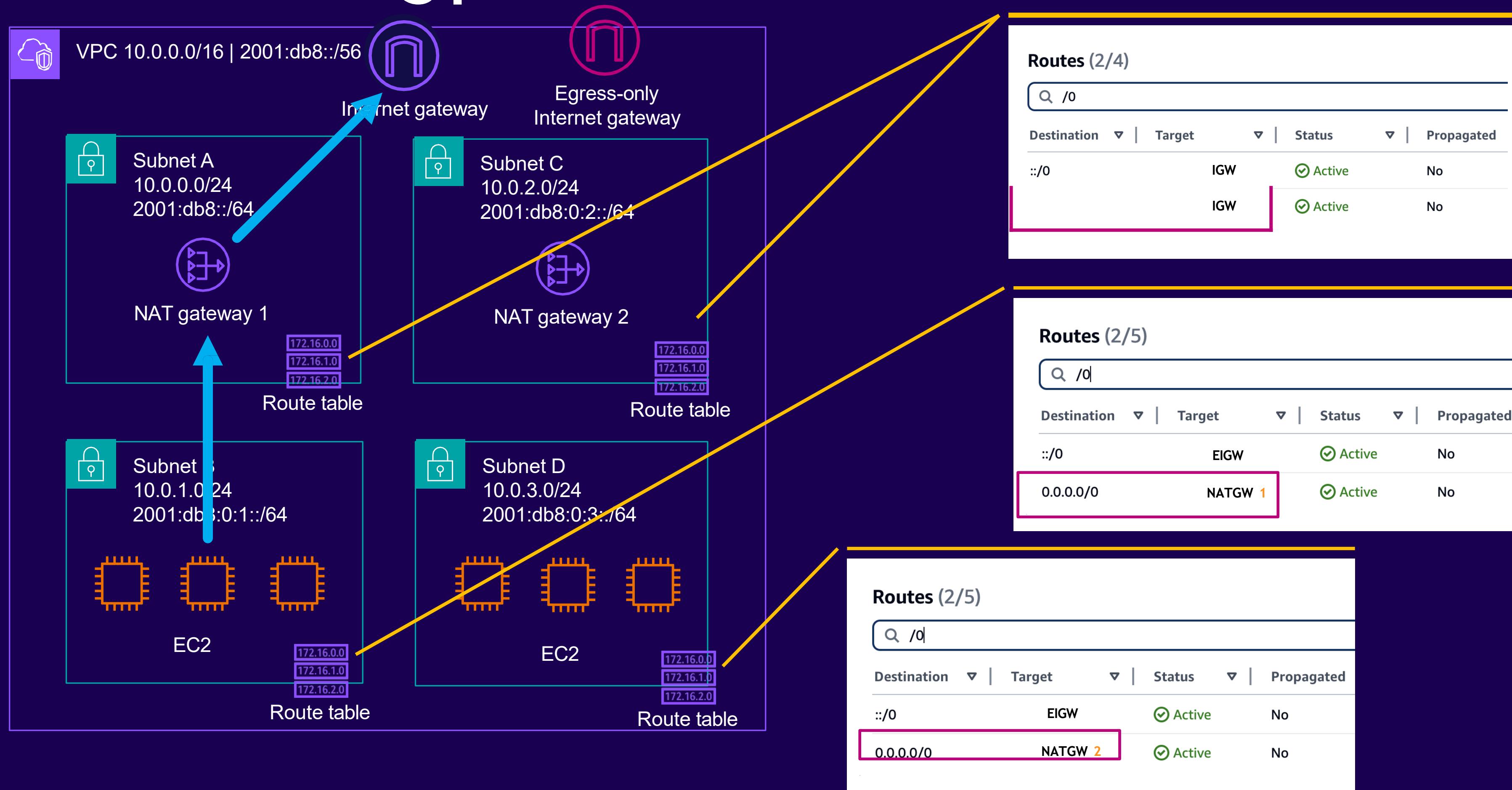


Why use different subnets?



- Deploy resources across different Availability Zones
- Apply network access control lists
- **Control routing per subnet**

Control routing per subnet



Control routing per subnet

Advanced settings

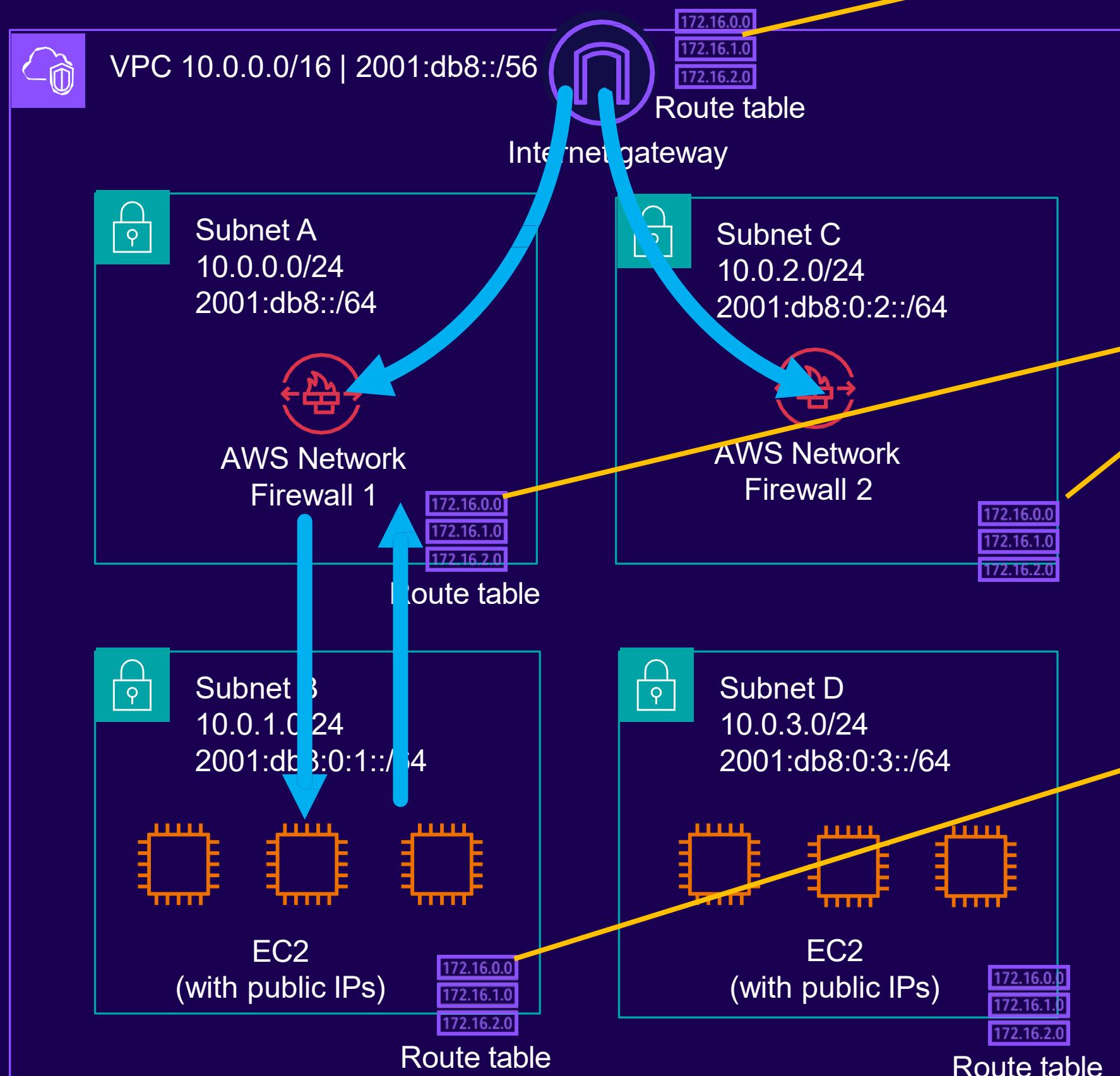
Ingress routing

Redirect **ingress** traffic from **internet gateway** or **virtual private gateway** to a network function

More specific routing

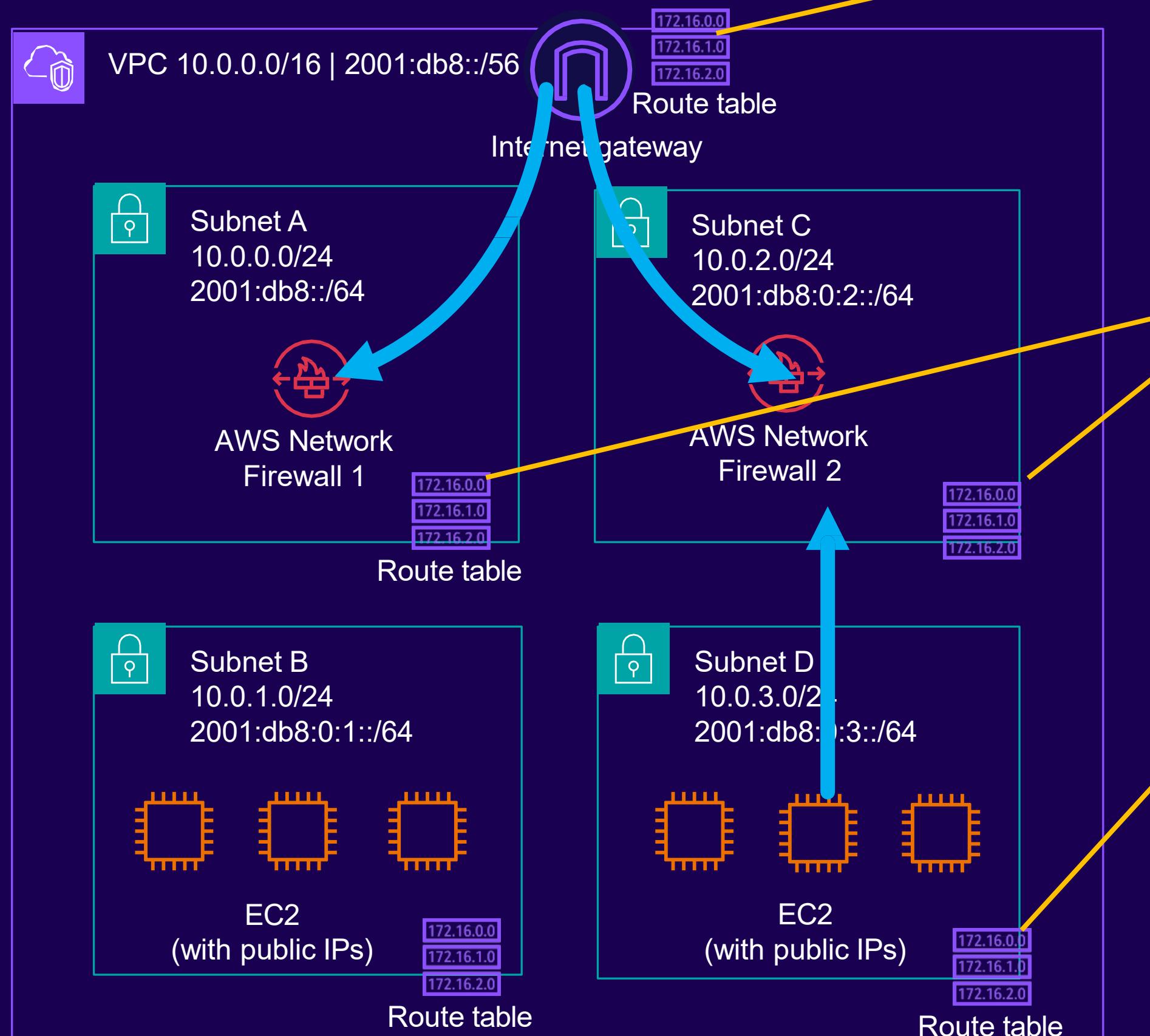
Redirect traffic **between subnets** to a network function

Ingress routing



Destination	Target	Status	Propagated
Subnet B	ANF 1	Active	No
Subnet D	ANF 2	Active	No
10.0.0.0/16	local	Active	No
2001:db8::/56	local	Active	No
Destination	Target	Status	Propagated
::/0	IGW IGW	Active	No
0.0.0.0/0		Active	No
10.0.0.0/16	local	Active	No
2001:db8::/56	local	Active	No
Destination	Target	Status	Propagated
::/0	ANF 1	Active	No
0.0.0.0/0	ANF 1	Active	No
10.0.0.0/16	local	Active	No
2001:db8::/56	local	Active	No

Ingress routing



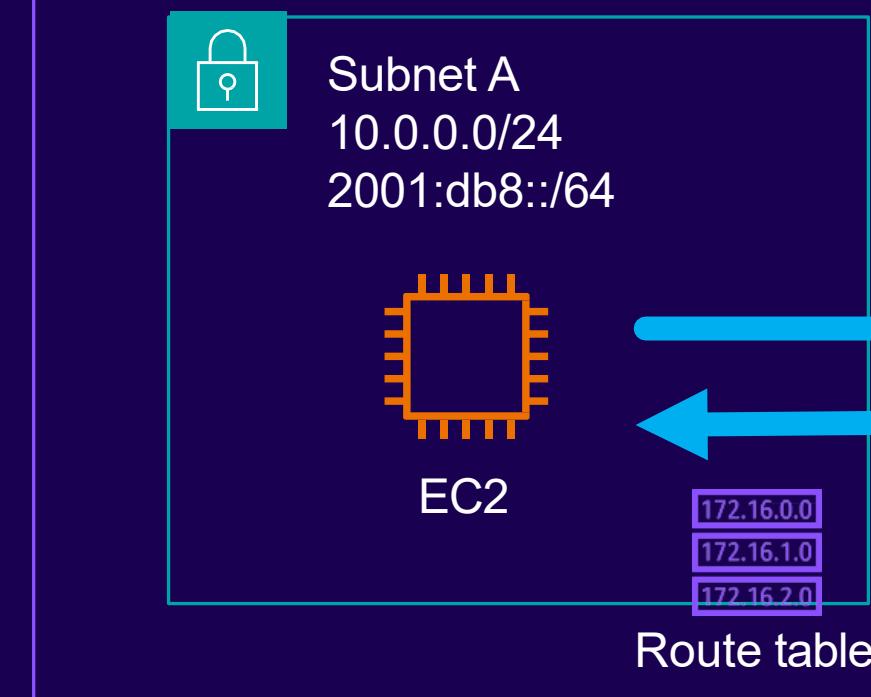
Destination	Target	Status	Propagated
Subnet B	ANF 1	Active	No
Subnet D	ANF 2	Active	No
10.0.0.0/16	local	Active	No
2001:db8::/56	local	Active	No

Destination	Target	Status	Propagated
::/0	IGW	Active	No
0.0.0.0/0	IGW	Active	No
10.0.0.0/16	local	Active	No
2001:db8::/56	local	Active	No

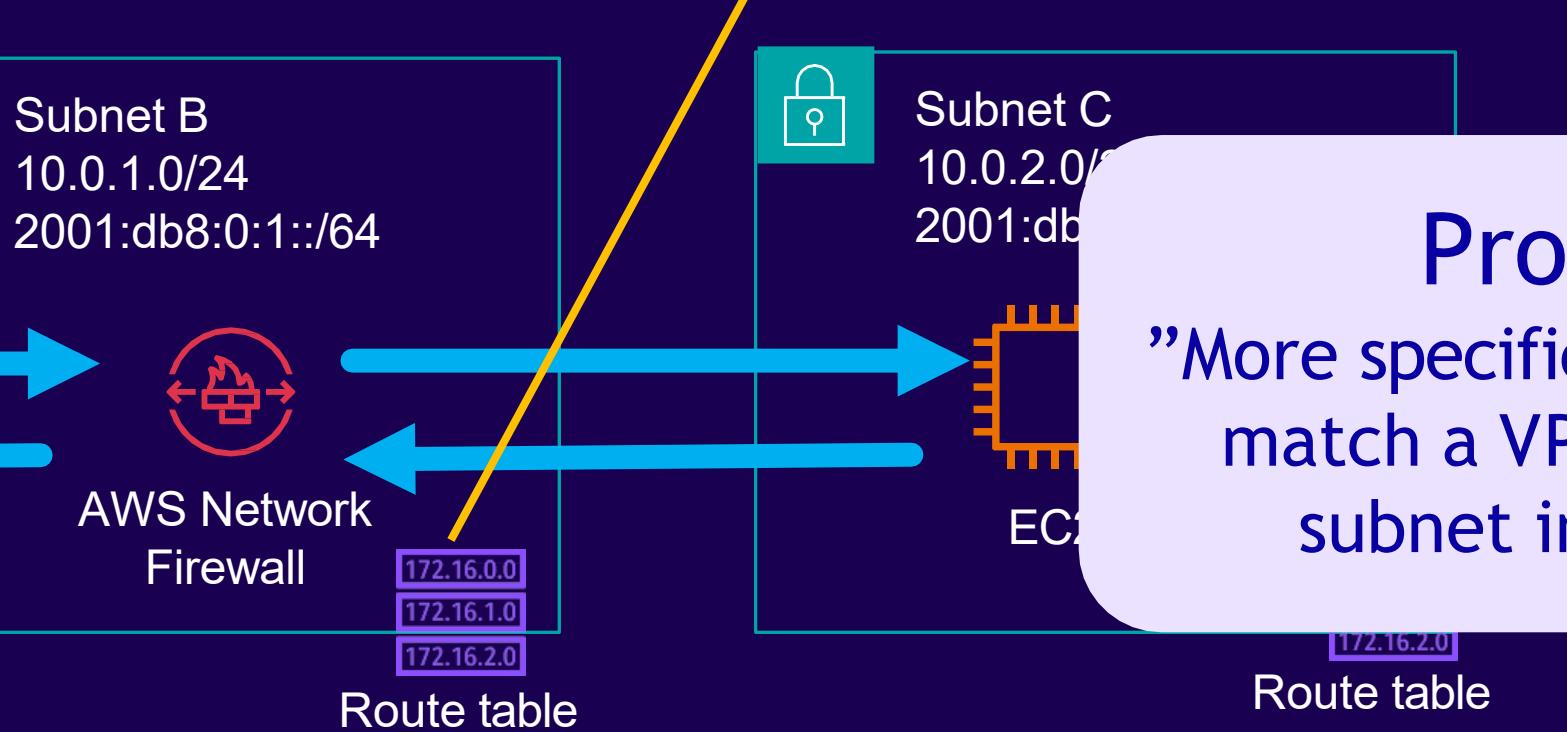
Destination	Target	Status	Propagated
::/0	ANF 2	Active	No
0.0.0.0/0	ANF 2	Active	No
10.0.0.0/16	local	Active	No
2001:db8::/56	local	Active	No

More specific routing

VPC 10.0.0.0/16 | 2001:db8::/56



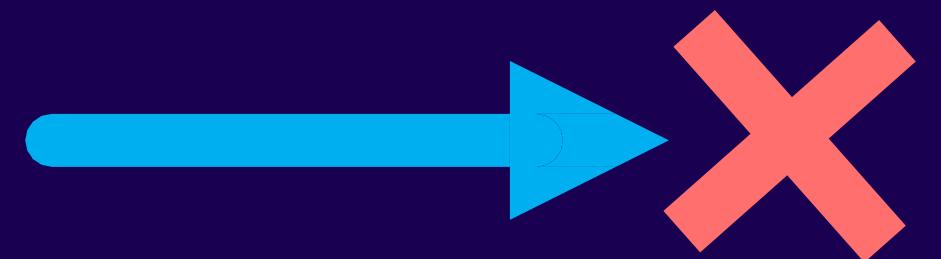
Destination	Target	Status	Propagated
10.0.0.0/16	local	Active	No
2001:db8::/56	local	Active	No



Destination	Target	Status	Propagated
Subnet C	ANF	Active	No
10.0.0.0/16	local	Active	No
2001:db8::/56	local	Active	No

Destination	Target	Status	Propagated
Subnet A	ANF	Active	No
10.0.0.0/16	local	Active	No
2001:db8::/56	local	Active	No

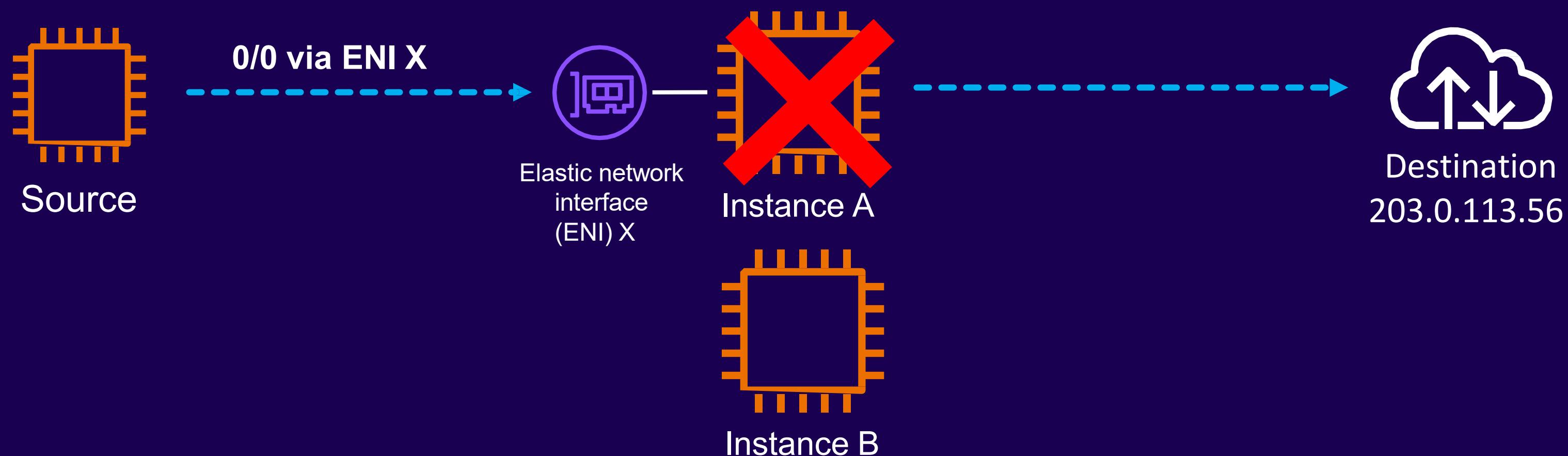
Solving next-hop resilience for VPC static routes



- Target AWS managed services - resilience built in (such as NAT Gateway, Network Firewall)
- For self hosted network functions:
 - AWS Gateway Load Balancer
 - Elastic network interface reattachment

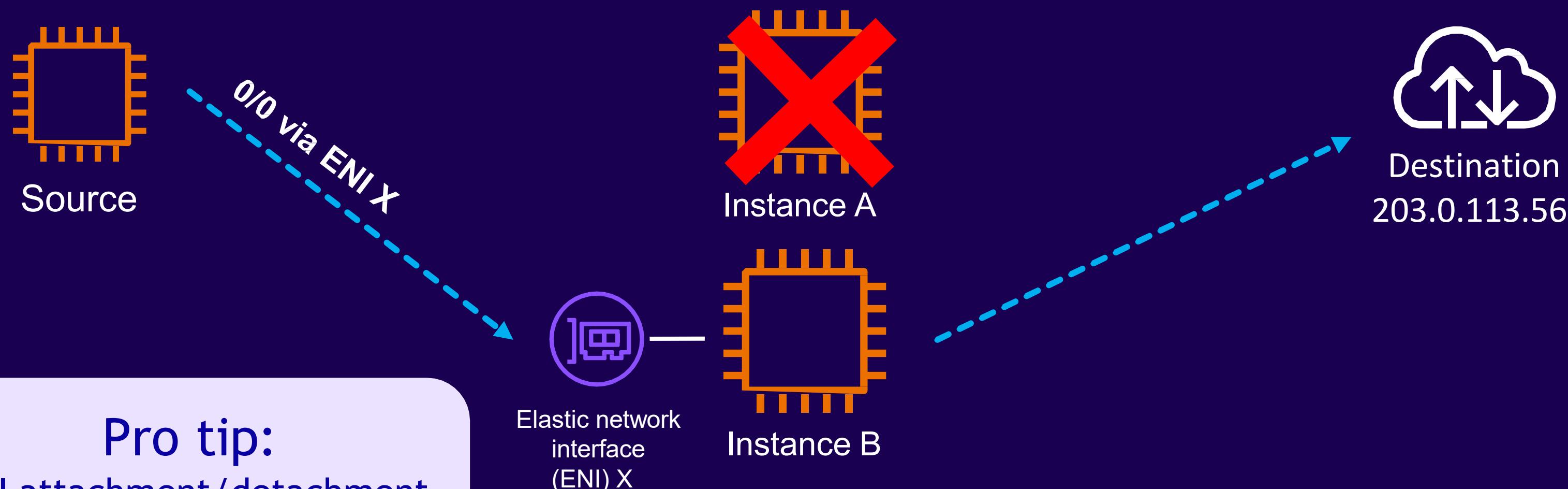
Resilience for static routing (VPC)

ENI reattachment



Resilience for static routing (VPC)

ENI reattachment

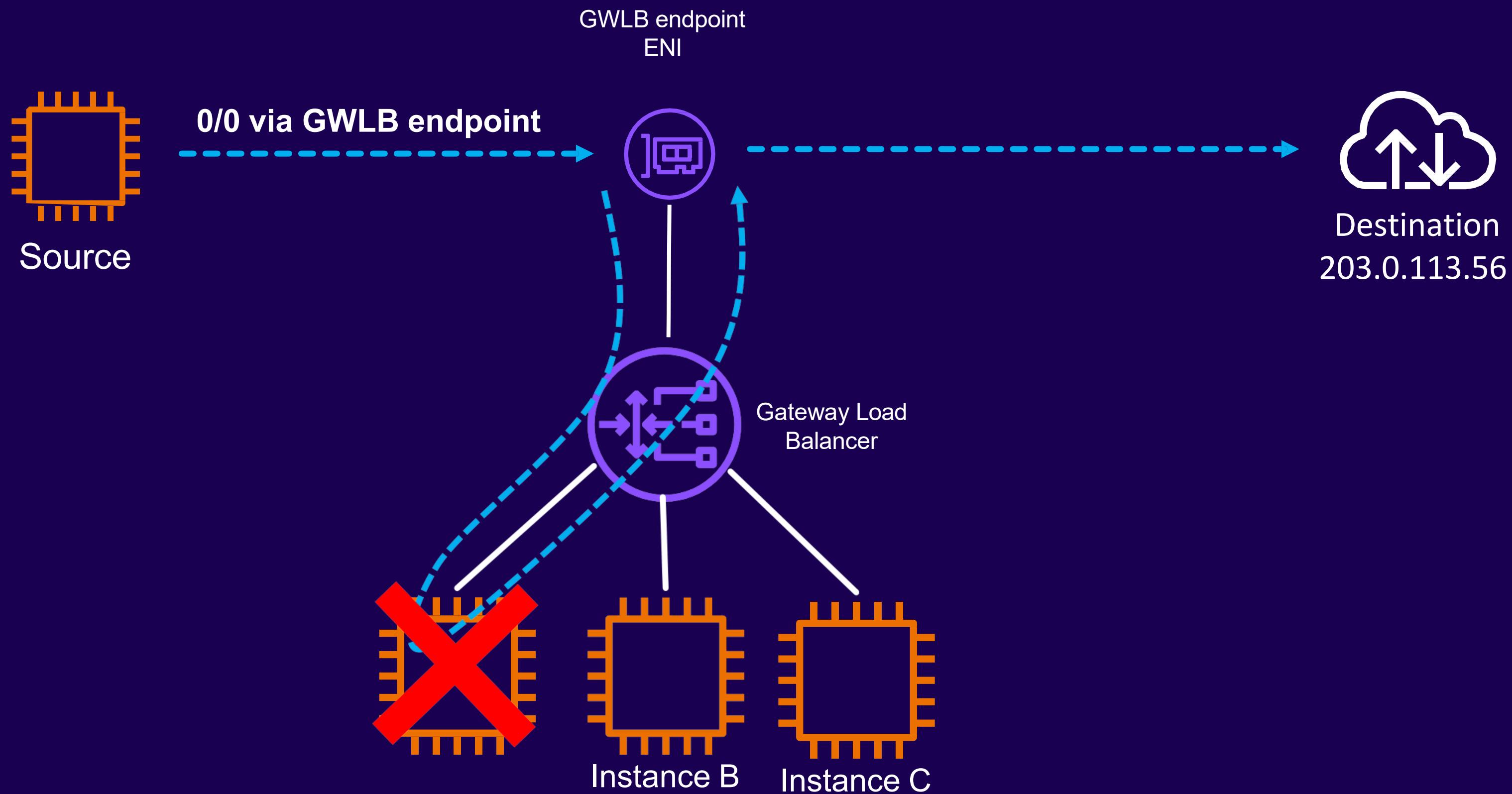


Pro tip:

ENI attachment/detachment
is a control plane
asynchronous API operation

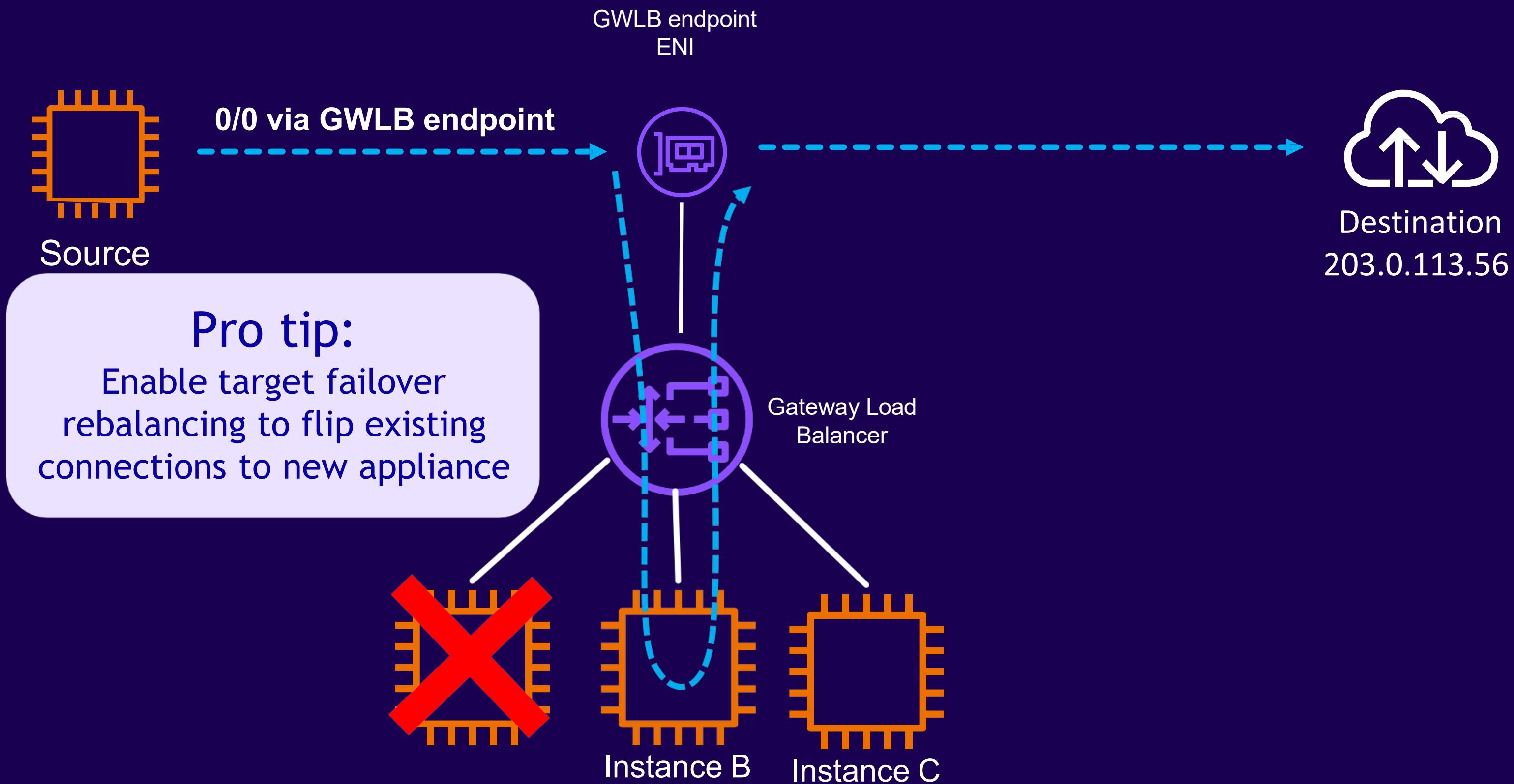
Resilience for static routing (VPC)

Gateway Load Balancer



Resilience for static routing (VPC)

Gateway Load Balancer



Resilience for static routing (VPC)

ENI reattachment

Customer managed failure detection

Manual failover

Control plane failover

Gateway Load Balancer

Automatic failure detection

Automatic failover

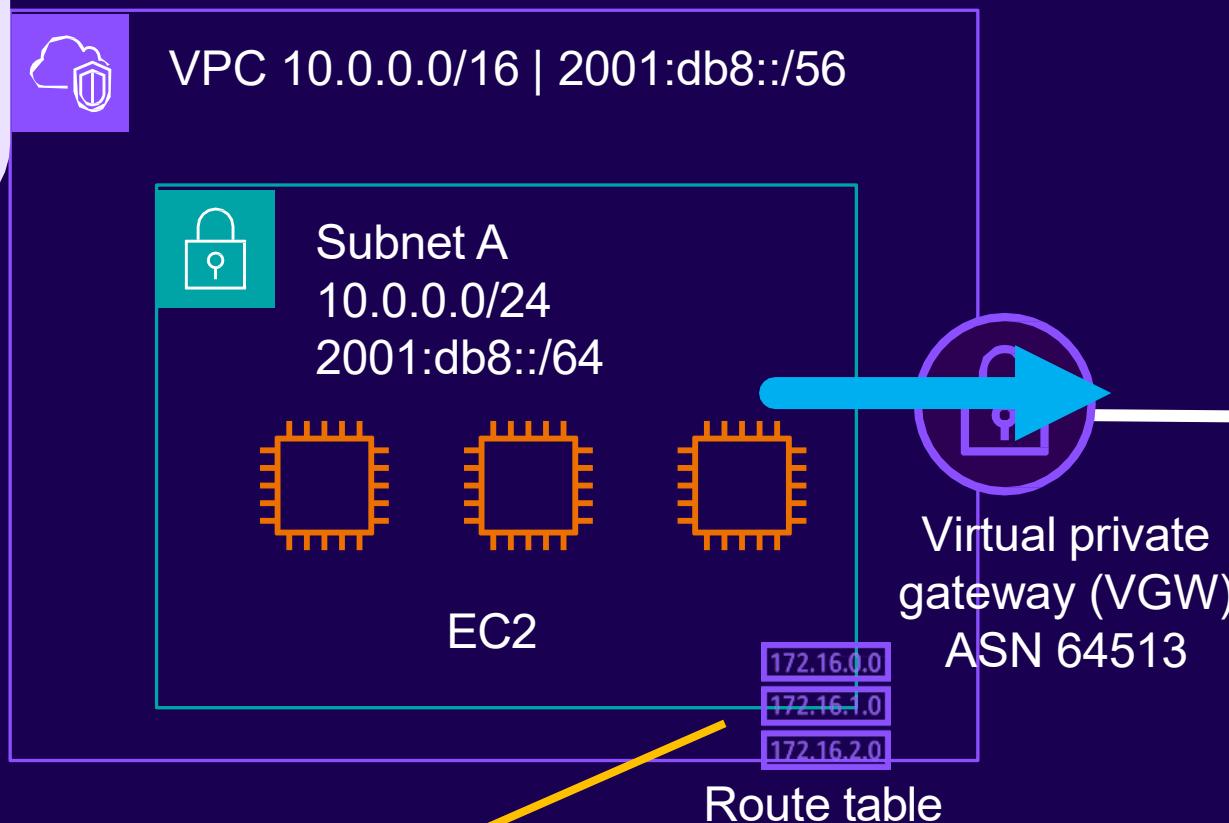
Data plane failover



Hybrid connectivity - Direct Connect

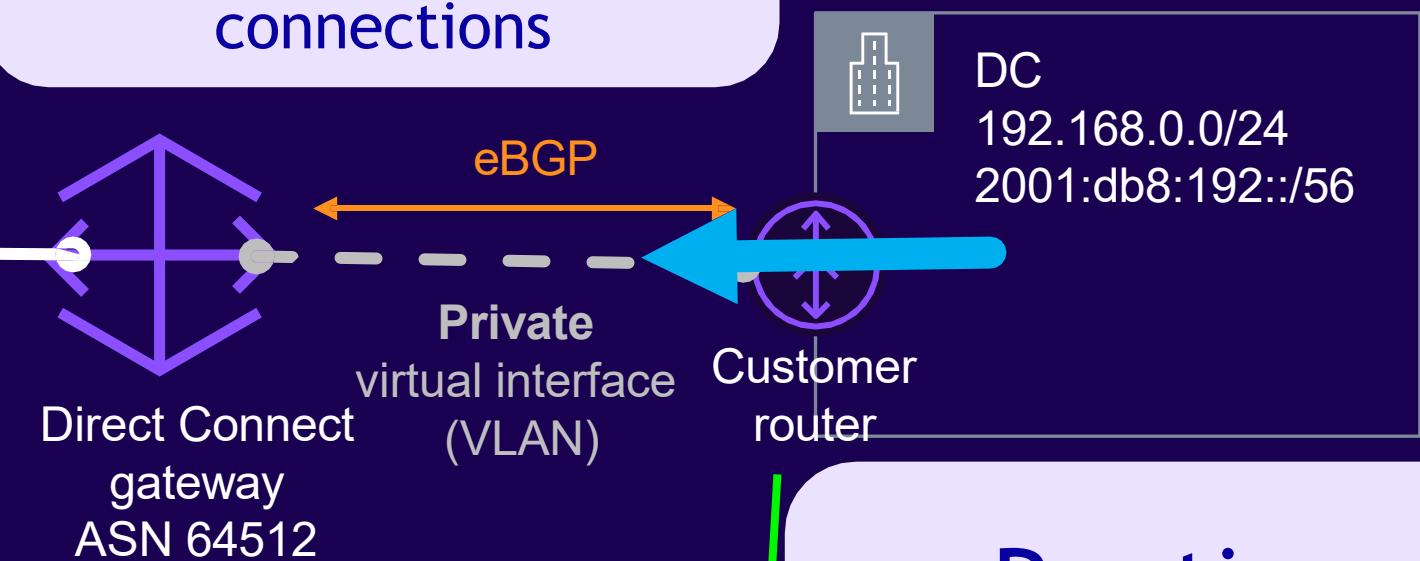
Pro tip:

VPC will advertise all CIDRs (IPv4 and IPv6) to customer router unless filtered on DXGW



Pro tip:

Ensure HA by using multiple Direct Connect connections



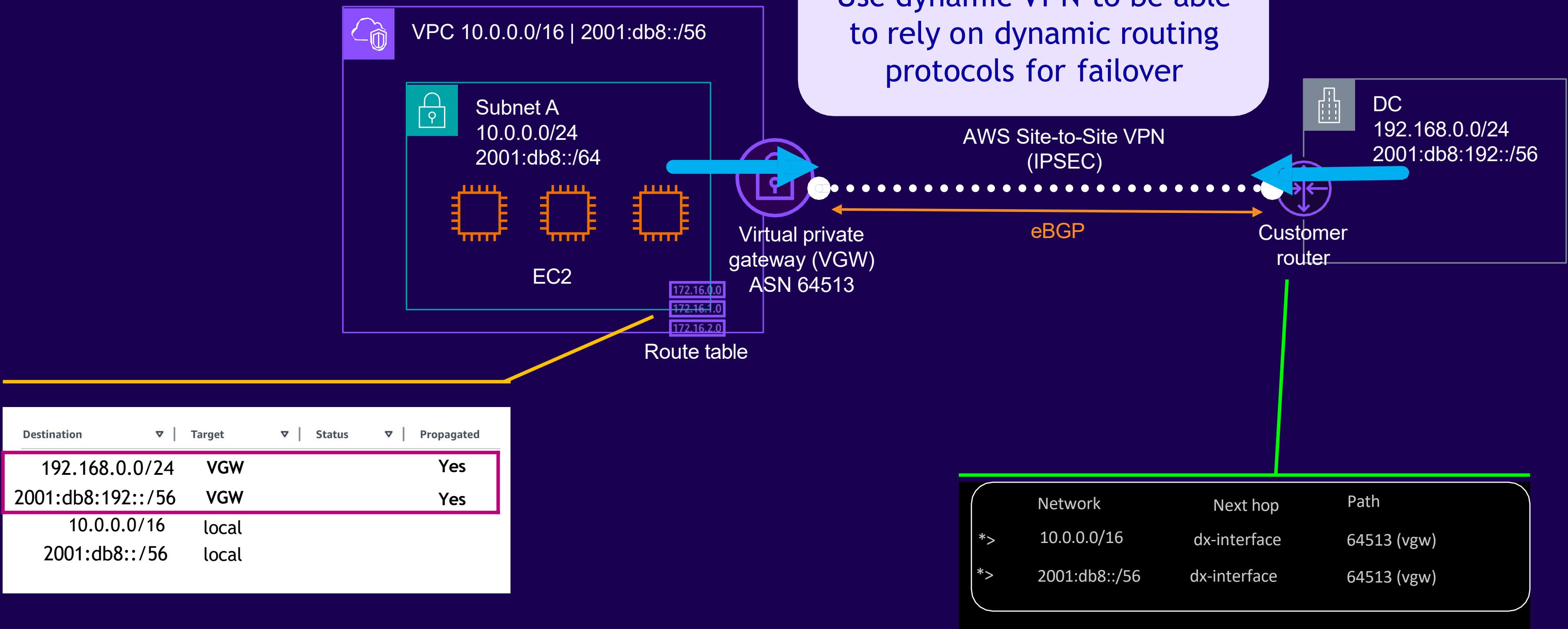
Pro tip:

DXGW “hides” the ASN of the VGW

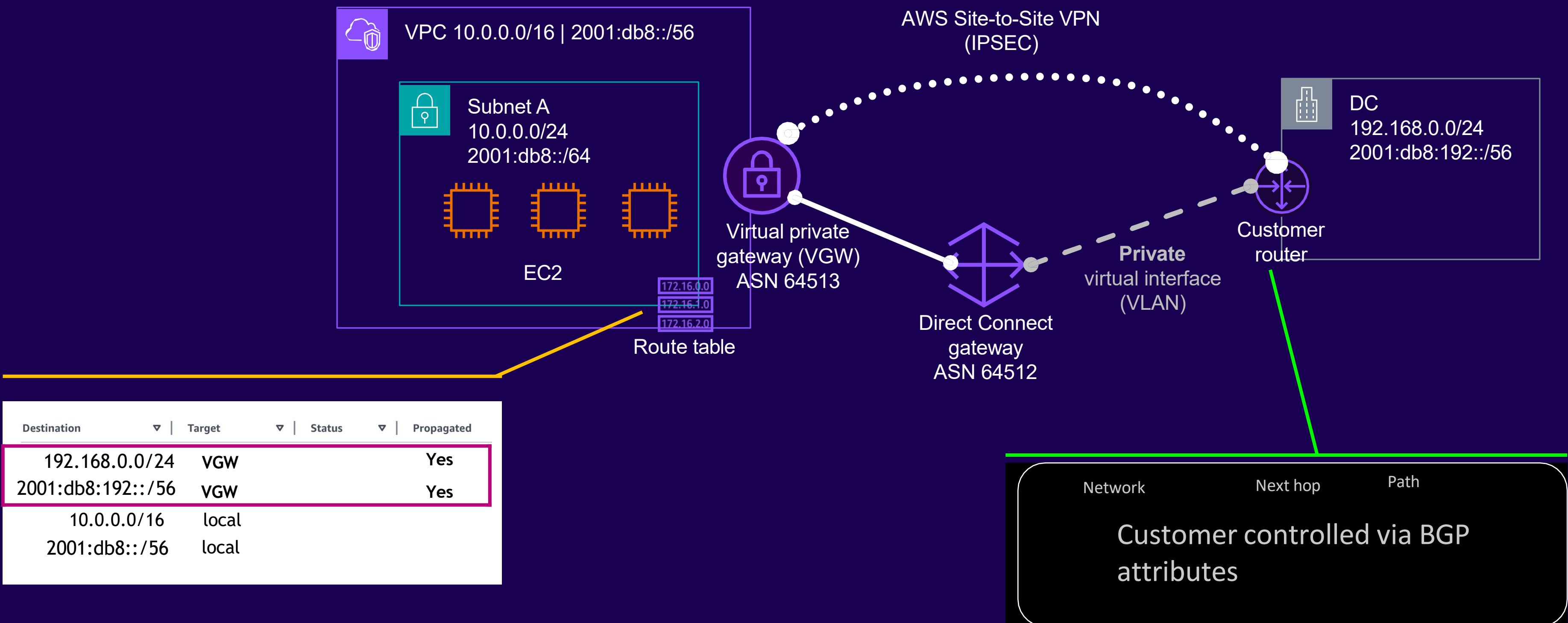
Destination	Target	Status	Propagated
192.168.0.0/24	VGW	Yes	
2001:db8:192::/56	VGW	Yes	
10.0.0.0/16	local		
2001:db8::/56	local		

	Network	Next hop	Path
*>	10.0.0.0/16	dx-interface	64512 (dxgw)
*>	2001:db8::/56	dx-interface	64512 (dxgw)

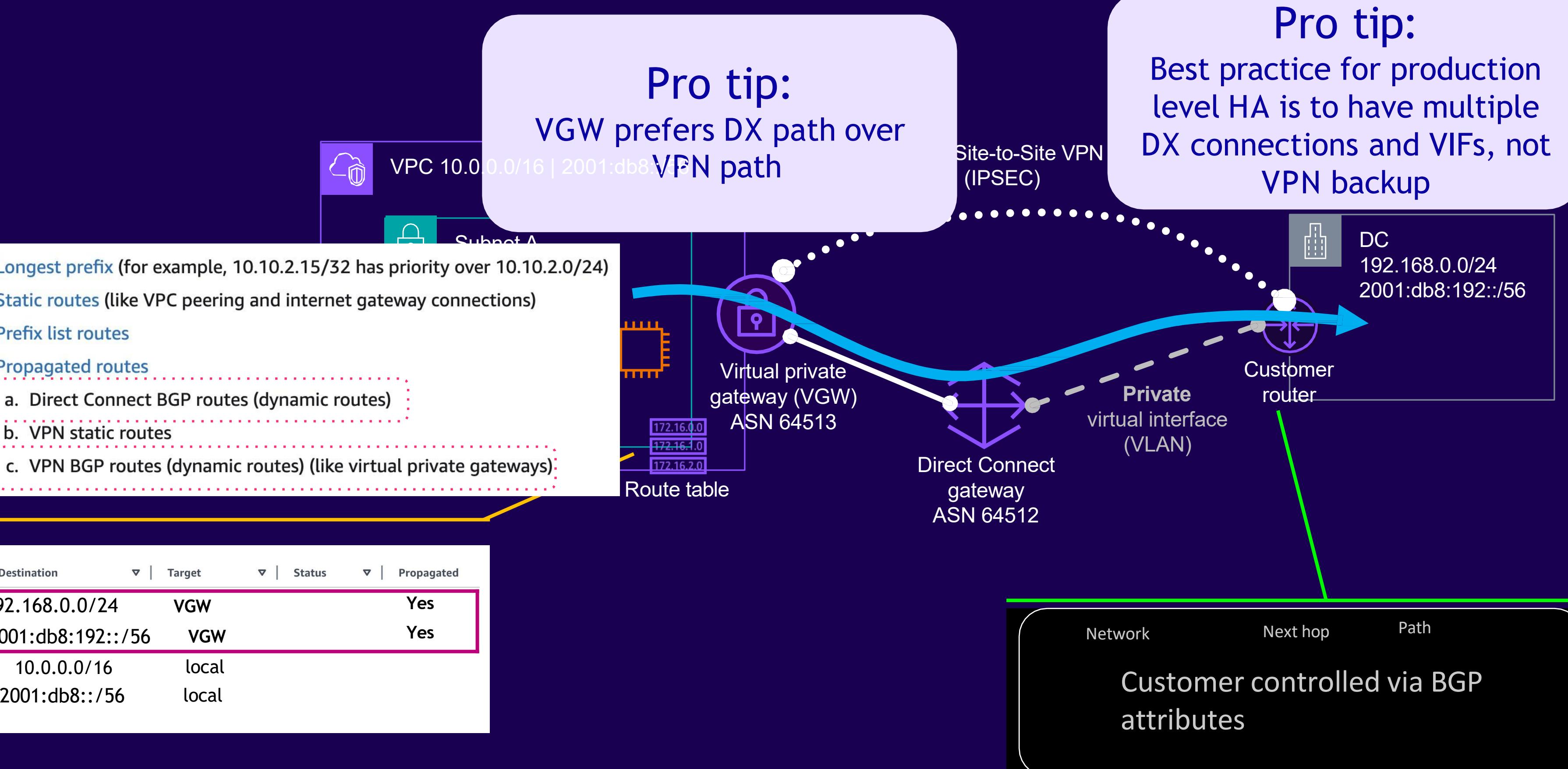
Hybrid connectivity - Site-to-Site VPN



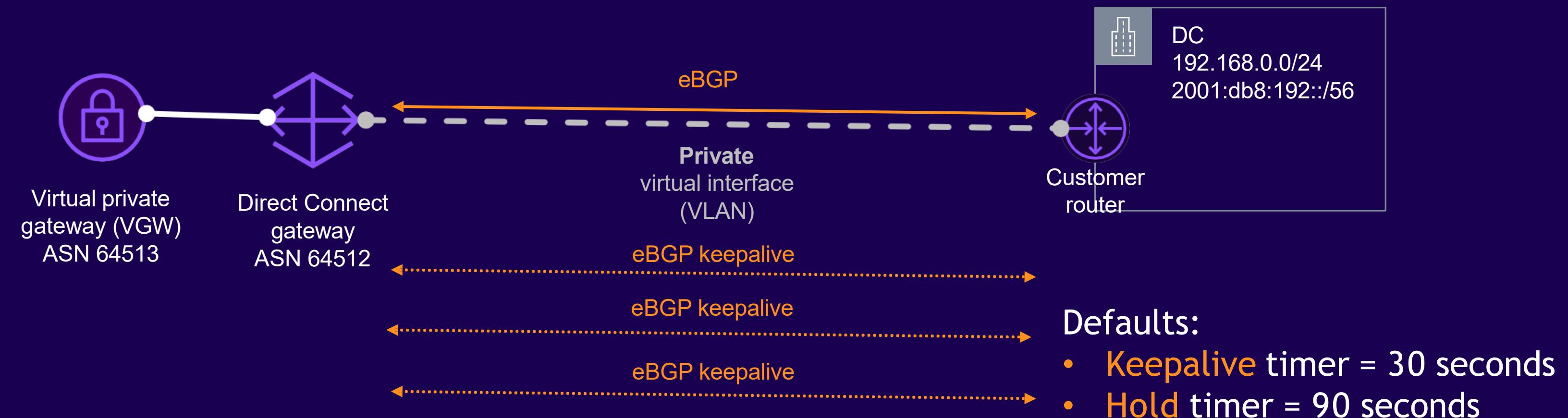
Hybrid connectivity - Route preference



Hybrid connectivity - Route preference



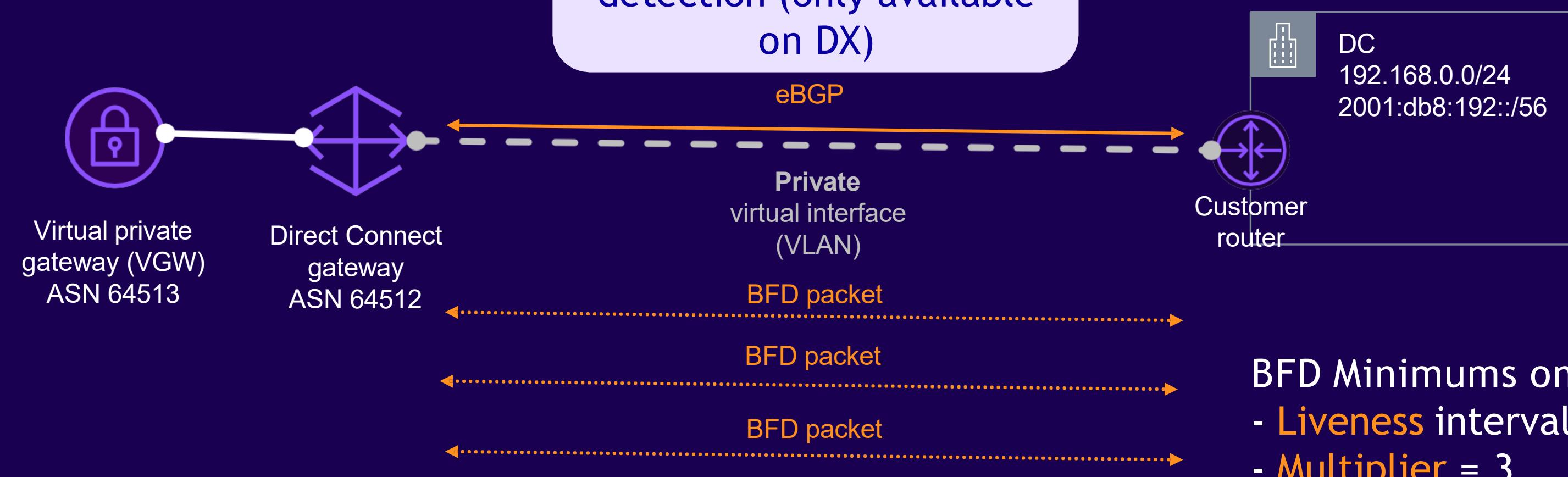
Hybrid connectivity - Failure detection



Failure detected in **90 seconds**

Hybrid connectivity - Failure detection

Bidirectional Forwarding Detection (BFD)



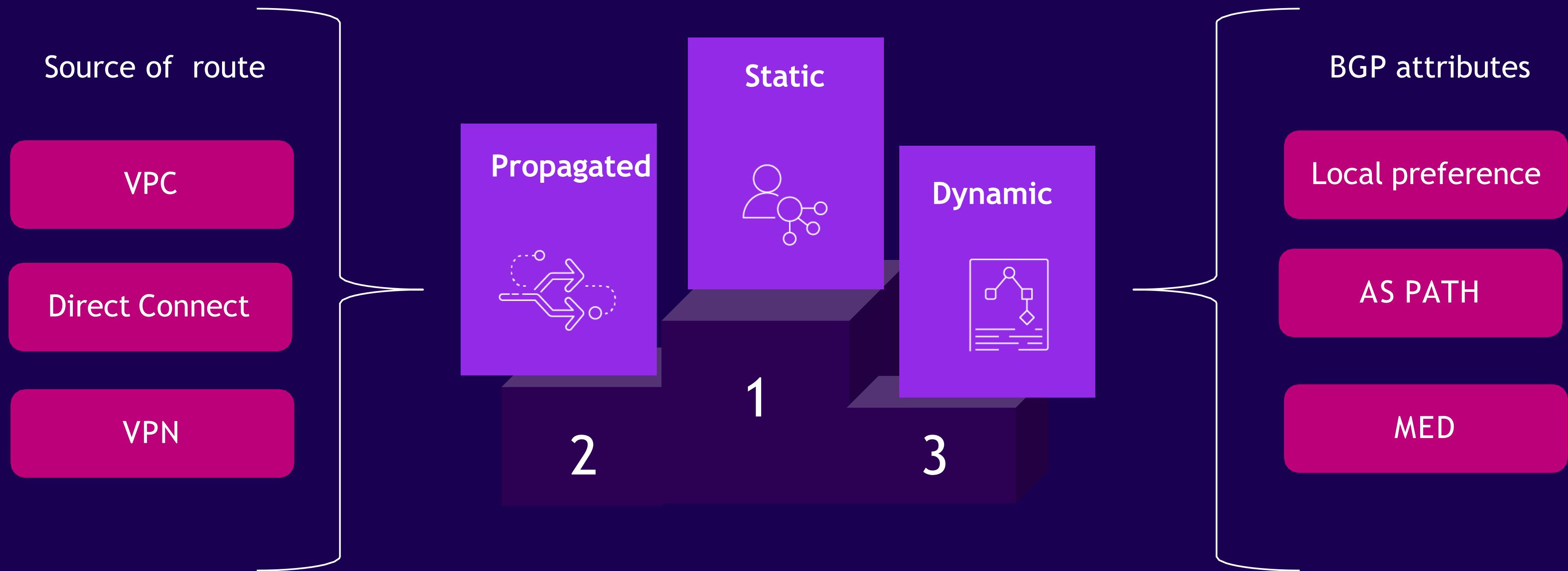
Pro tip:

Use BFD for faster failure detection (only available on DX)

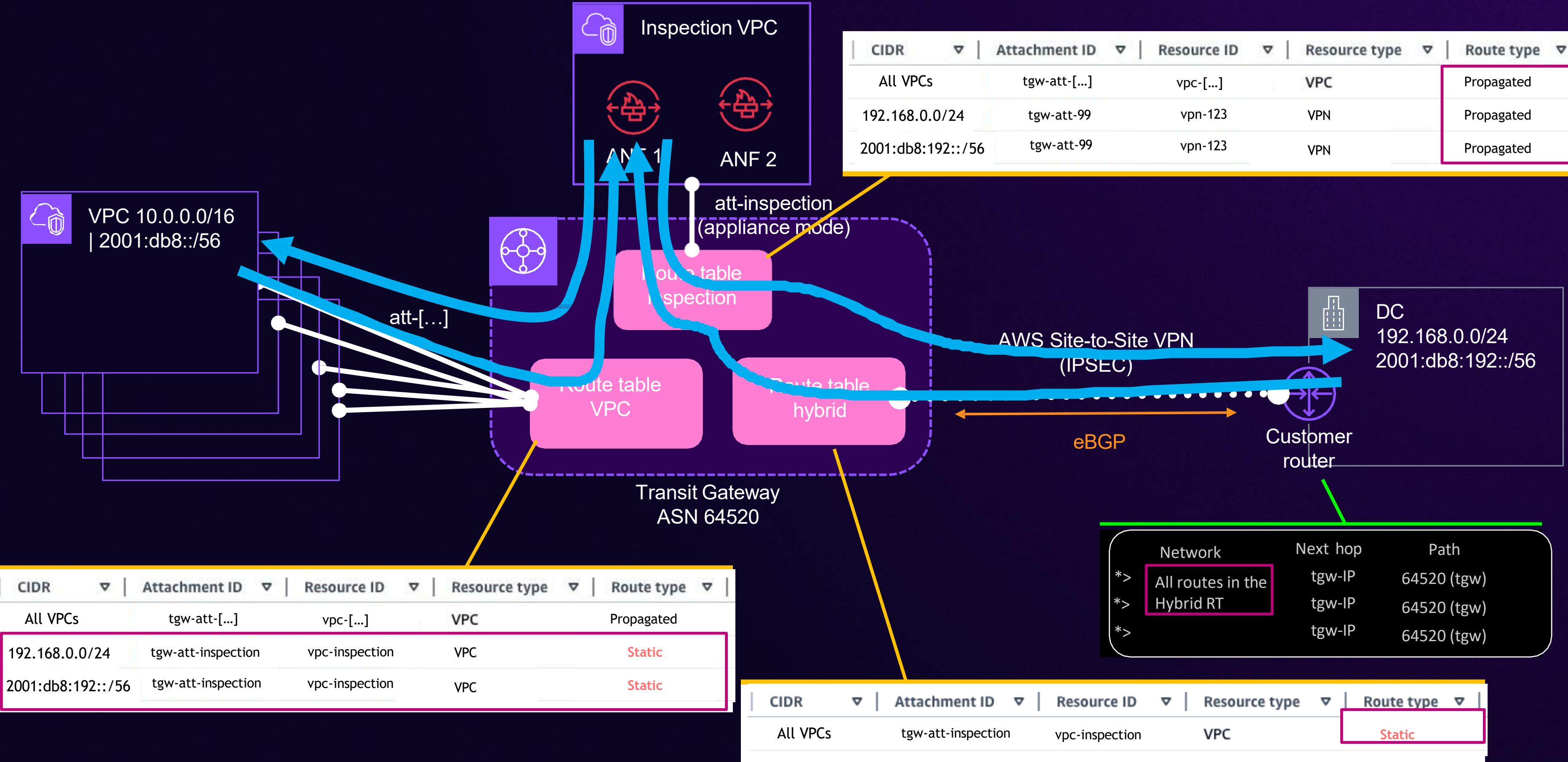
Failure detected in <1 second

- BFD Minimums on DX:**
- Liveness interval = 300 ms
 - Multiplier = 3

Route preference - Tie breakers

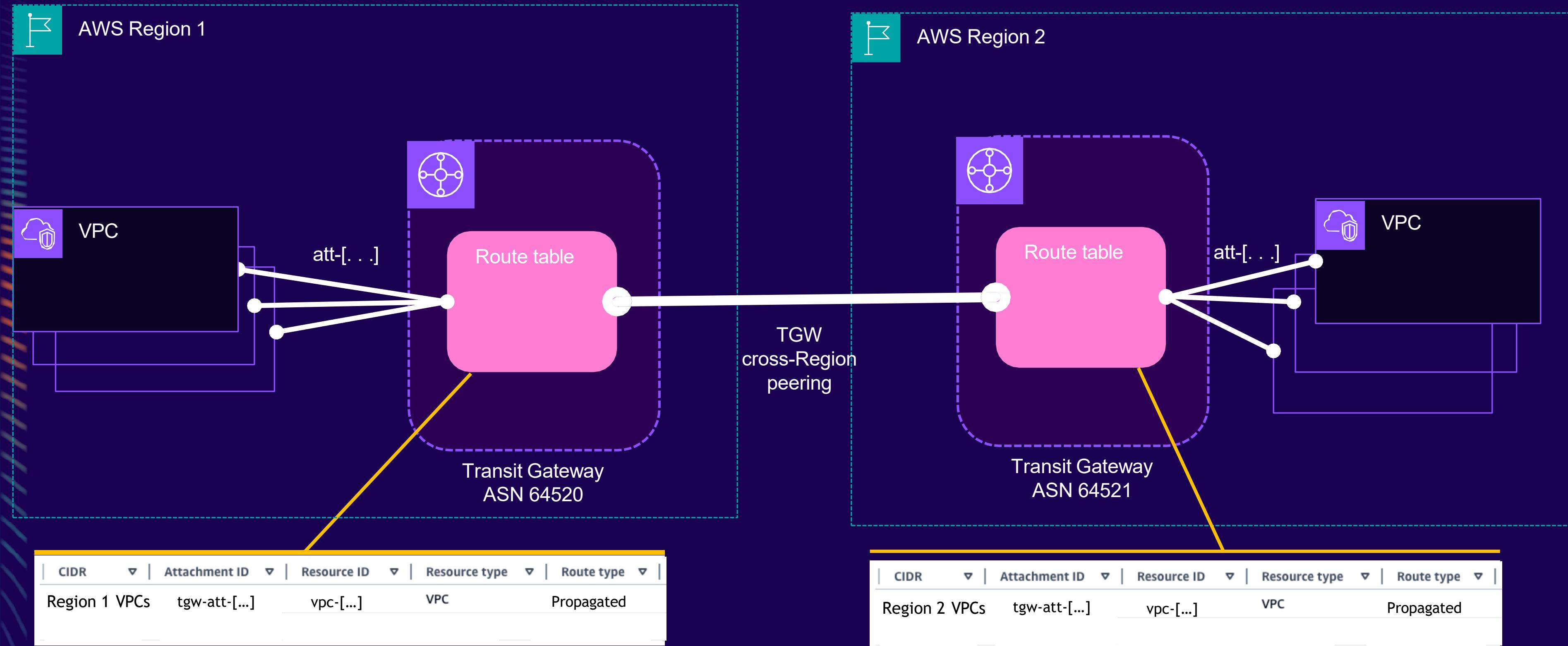


AWS Transit Gateway - VPN + inspection

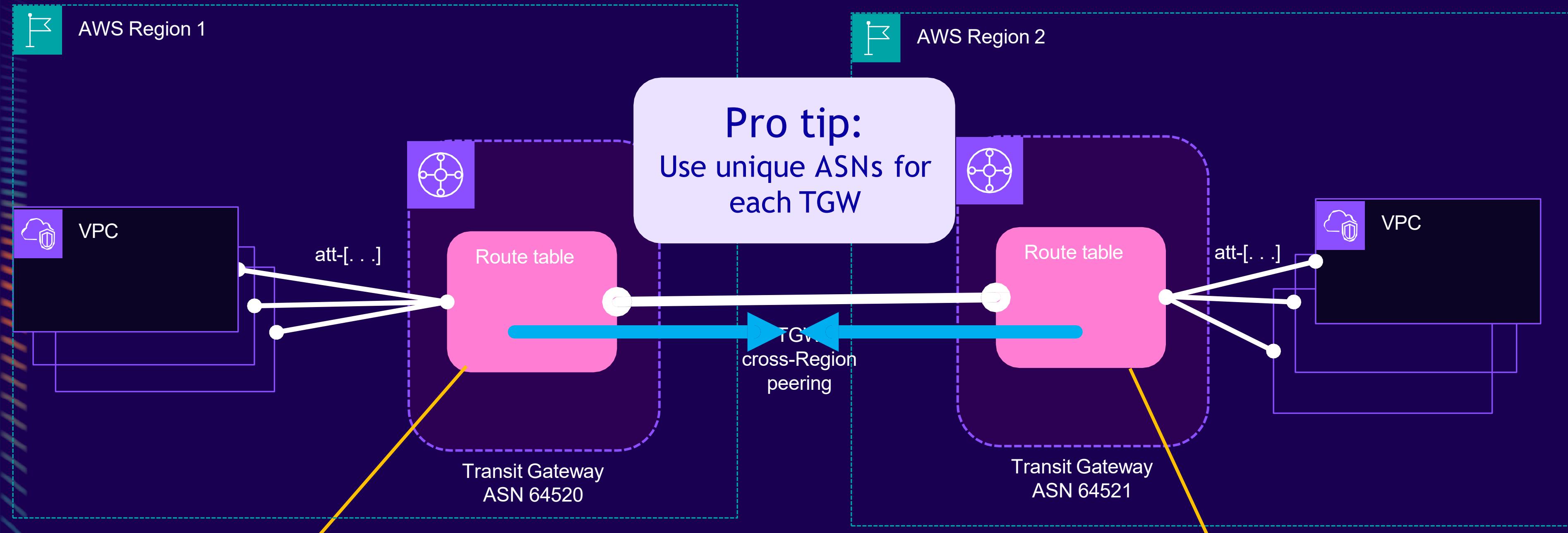


**Routing scenarios:
Multiple VPCs, multiple AWS
Regions**

AWS Transit Gateway peering



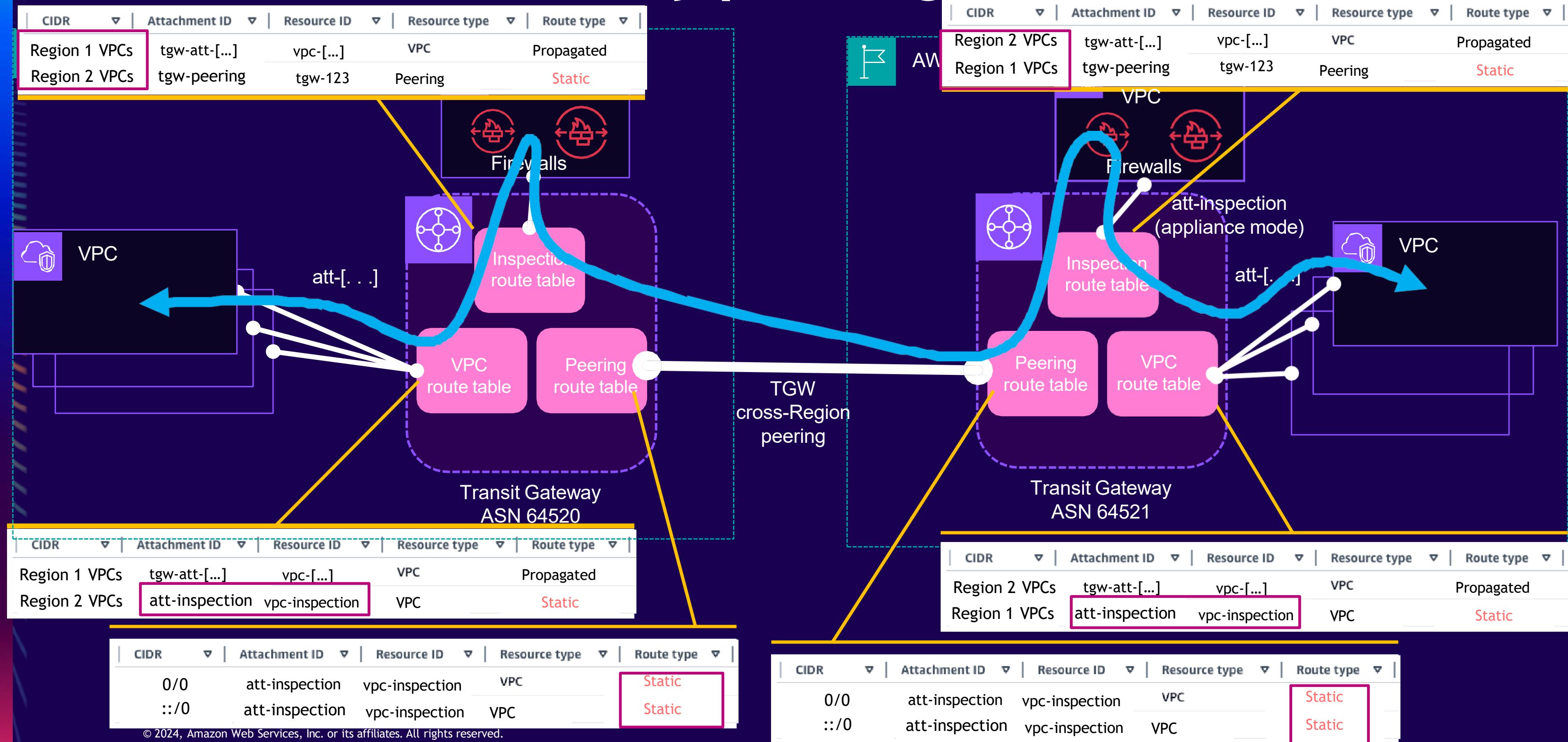
AWS Transit Gateway peering



CIDR	Attachment ID	Resource ID	Resource type	Route type
Region 1 VPCs	tgw-att-[...]	vpc-[...]	VPC	Propagated
Region 2 VPCs	tgw-peering	tgw-123	Peering	Static

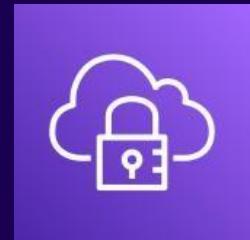
CIDR	Attachment ID	Resource ID	Resource type	Route type
Region 2 VPCs	tgw-att-[...]	vpc-[...]	VPC	Propagated
Region 1 VPCs	tgw-peering	tgw-123	Peering	Static

AWS Transit Gateway peering + inspection



Peering, endpoints, and gateways

Peering, endpoints, and gateways



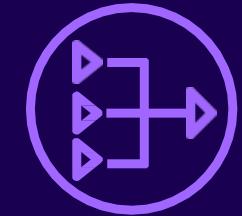
AWS Client VPN
endpoint



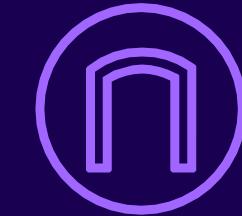
Virtual private
gateway



Direct Connect
gateway



NAT
gateway



Internet
gateway



AWS Transit
Gateway



Endpoints

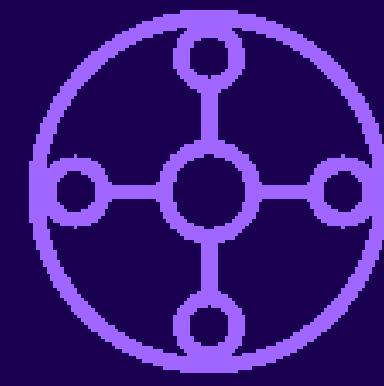


Peering
connection

Connecting multiple VPC

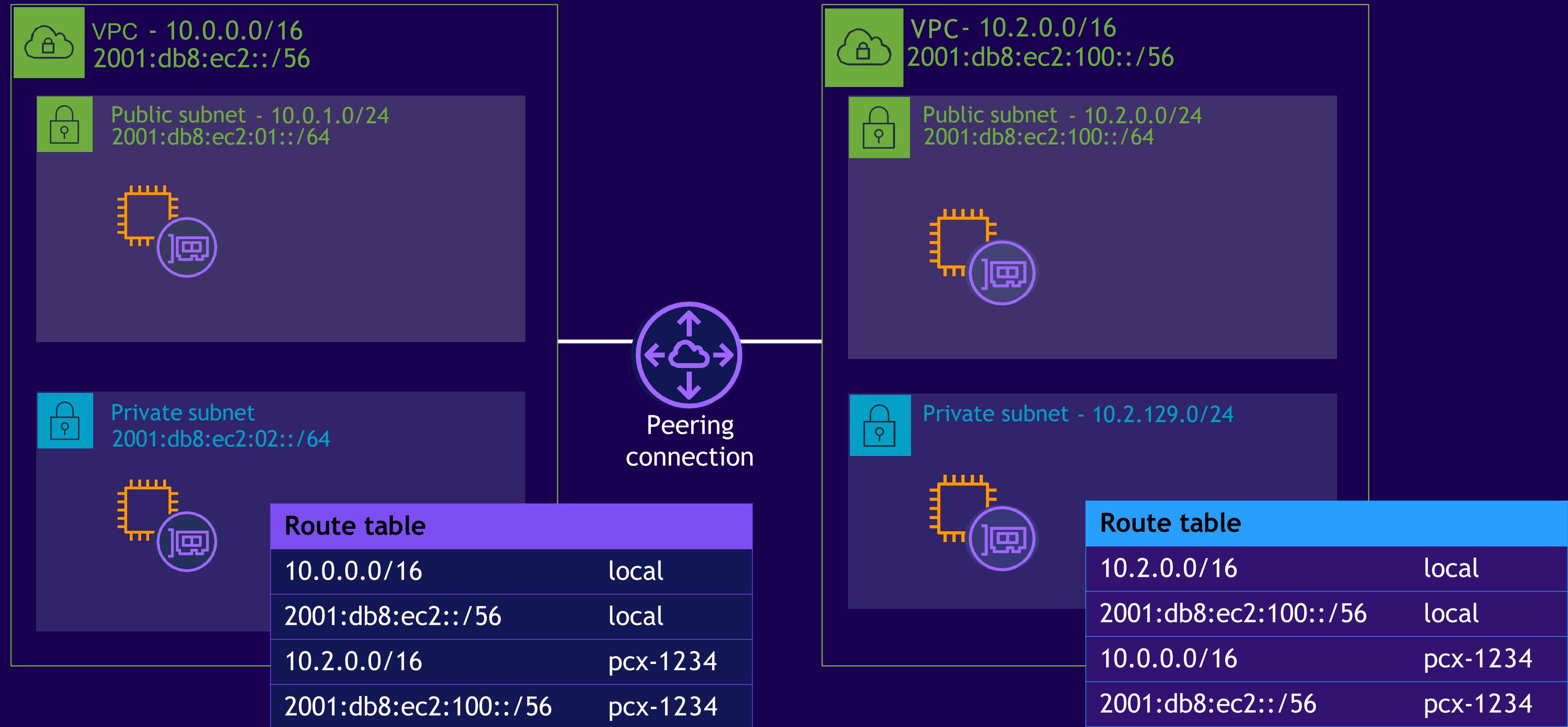


VPC Peering

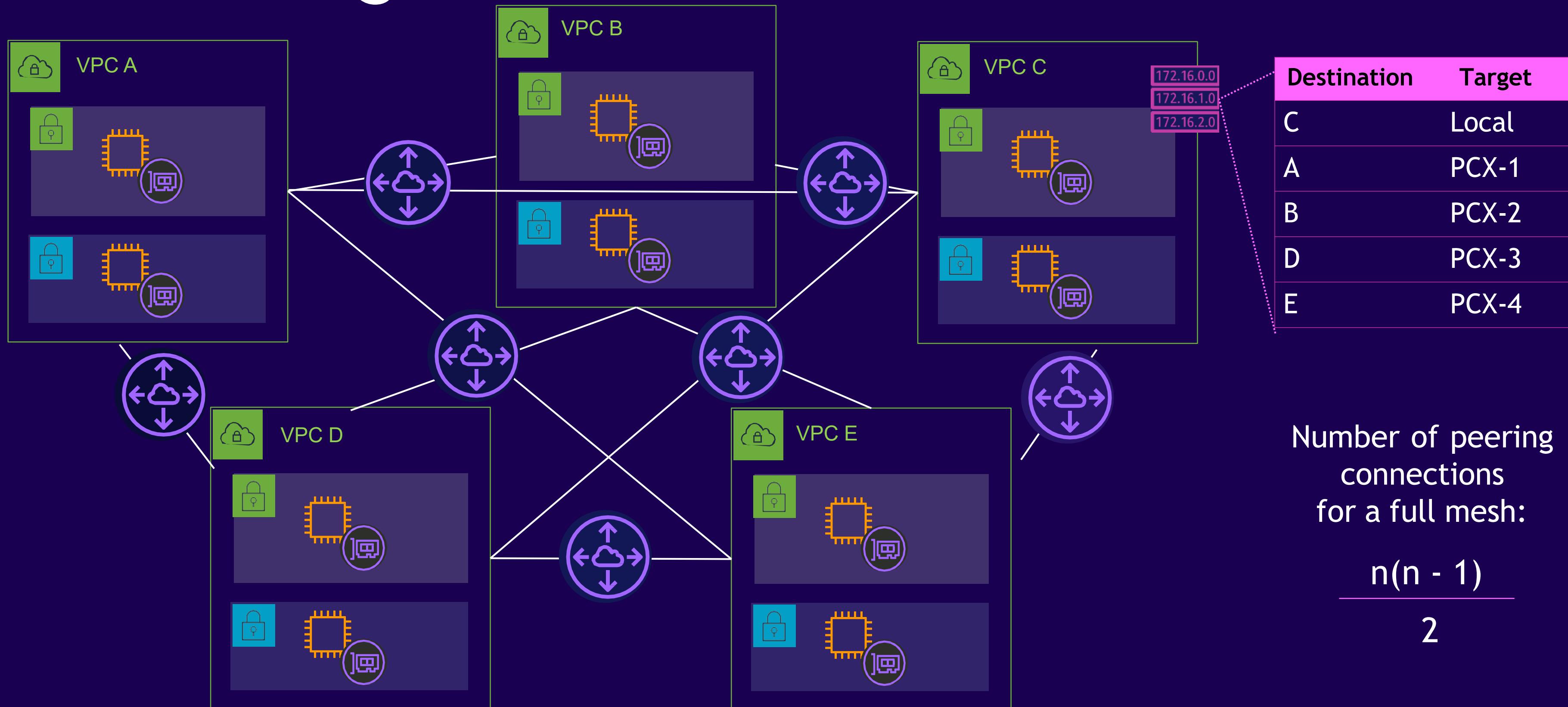


Transit Gateway

VPC Peering



VPC Peering



What is the problem?

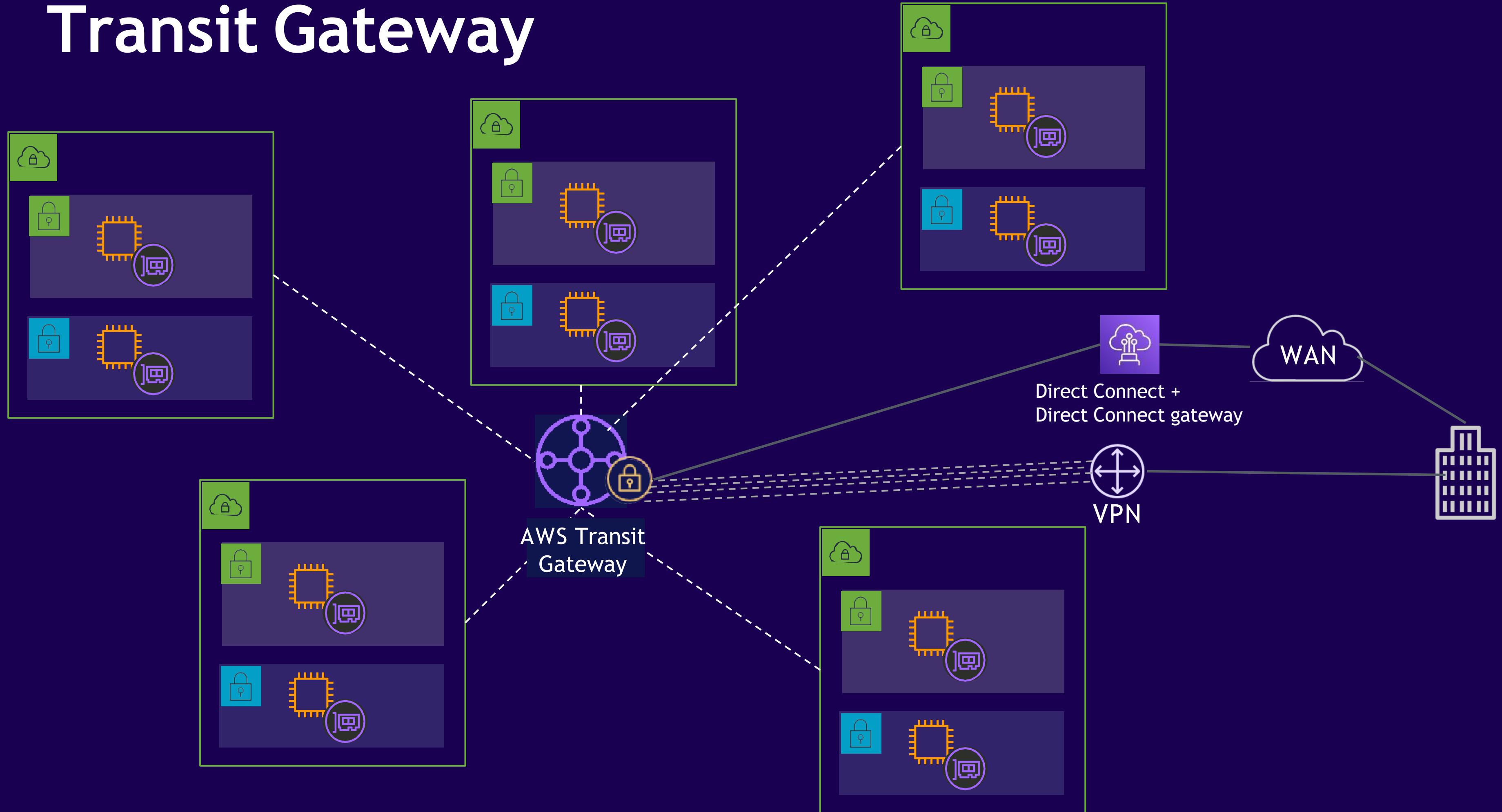
Complexity:

$$\frac{100 (100 - 1)}{2} = 4,950$$

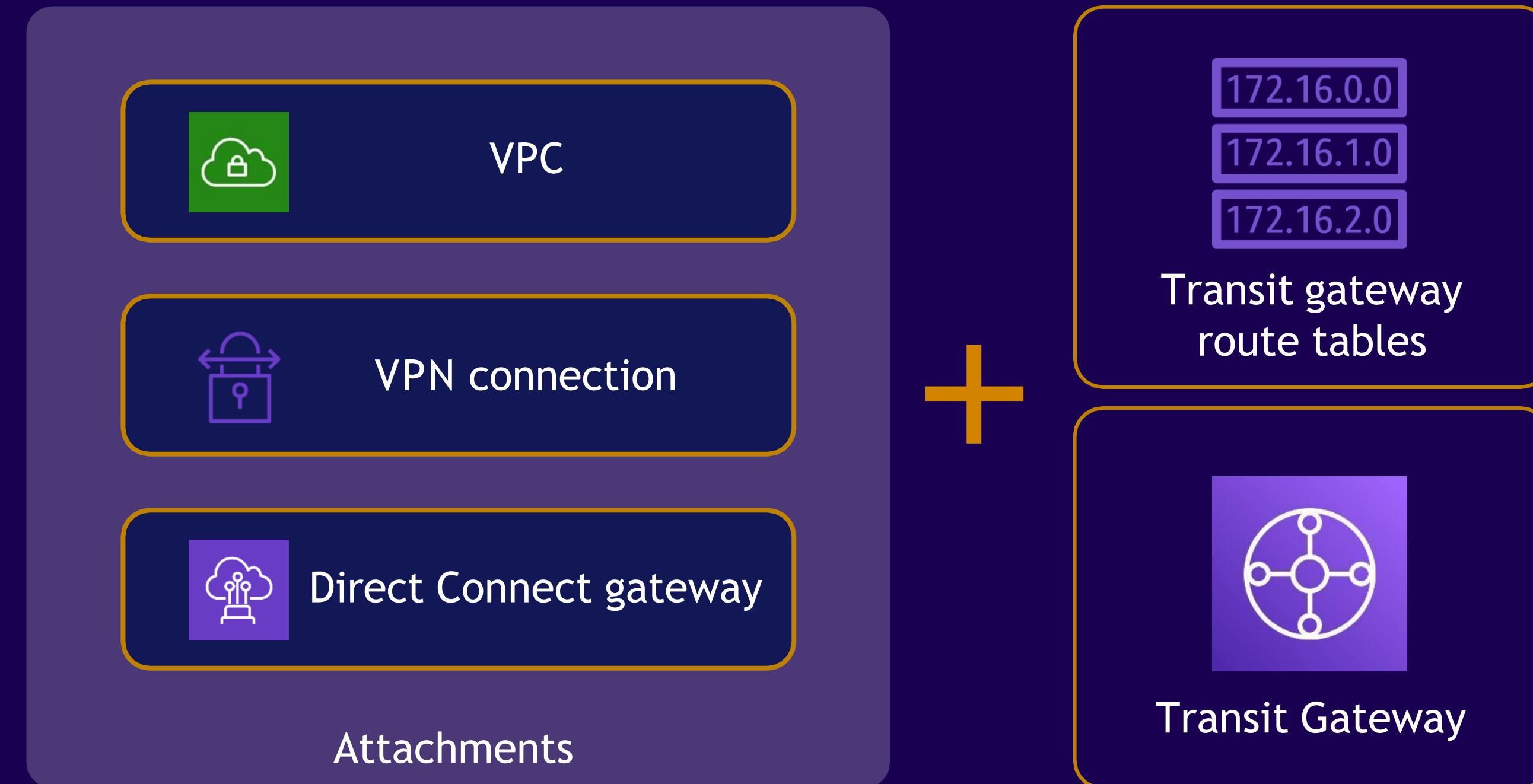
Service Limit:

Amazon VPC peering connections per Amazon VPC = 125

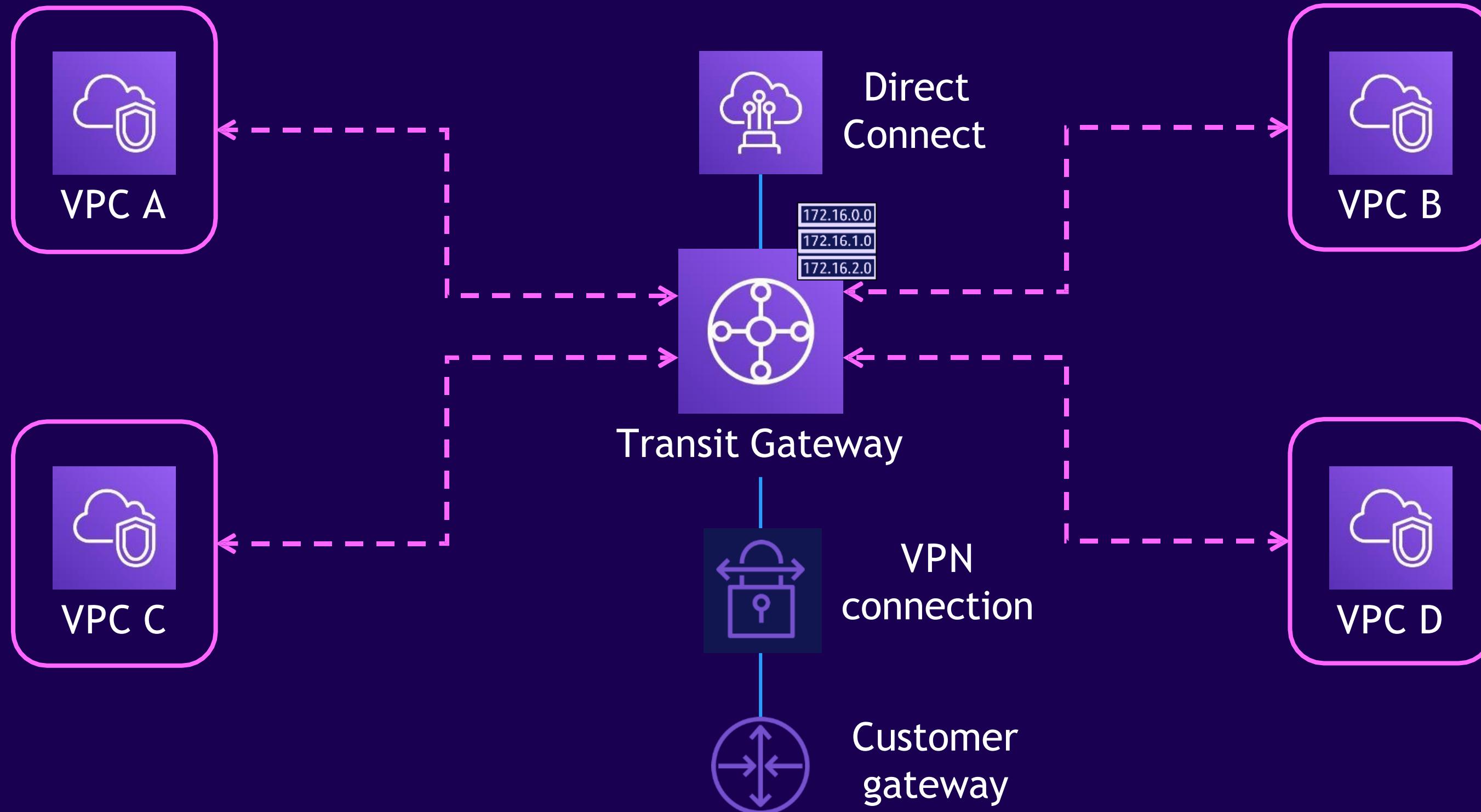
Transit Gateway



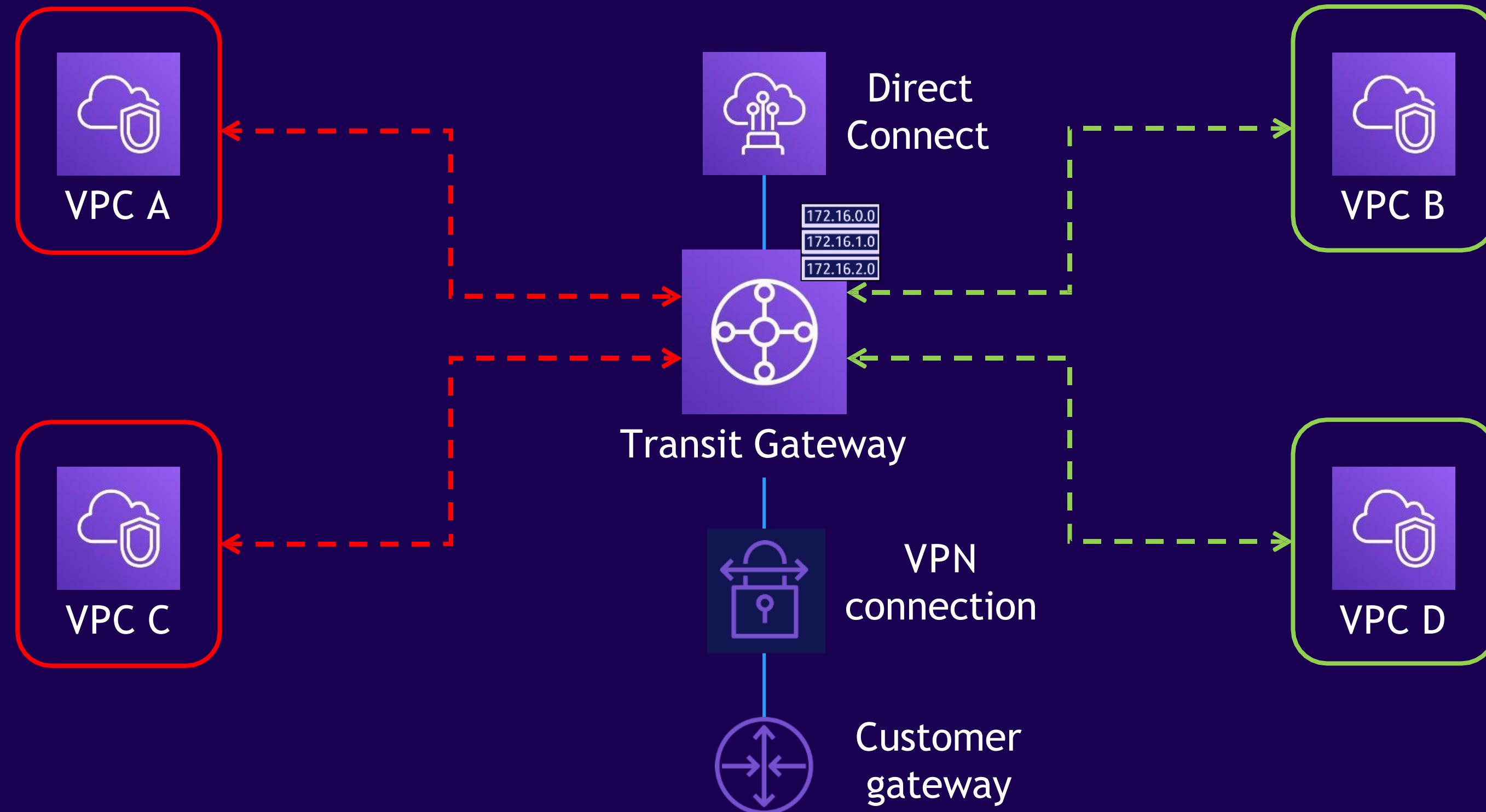
Transit Gateway component



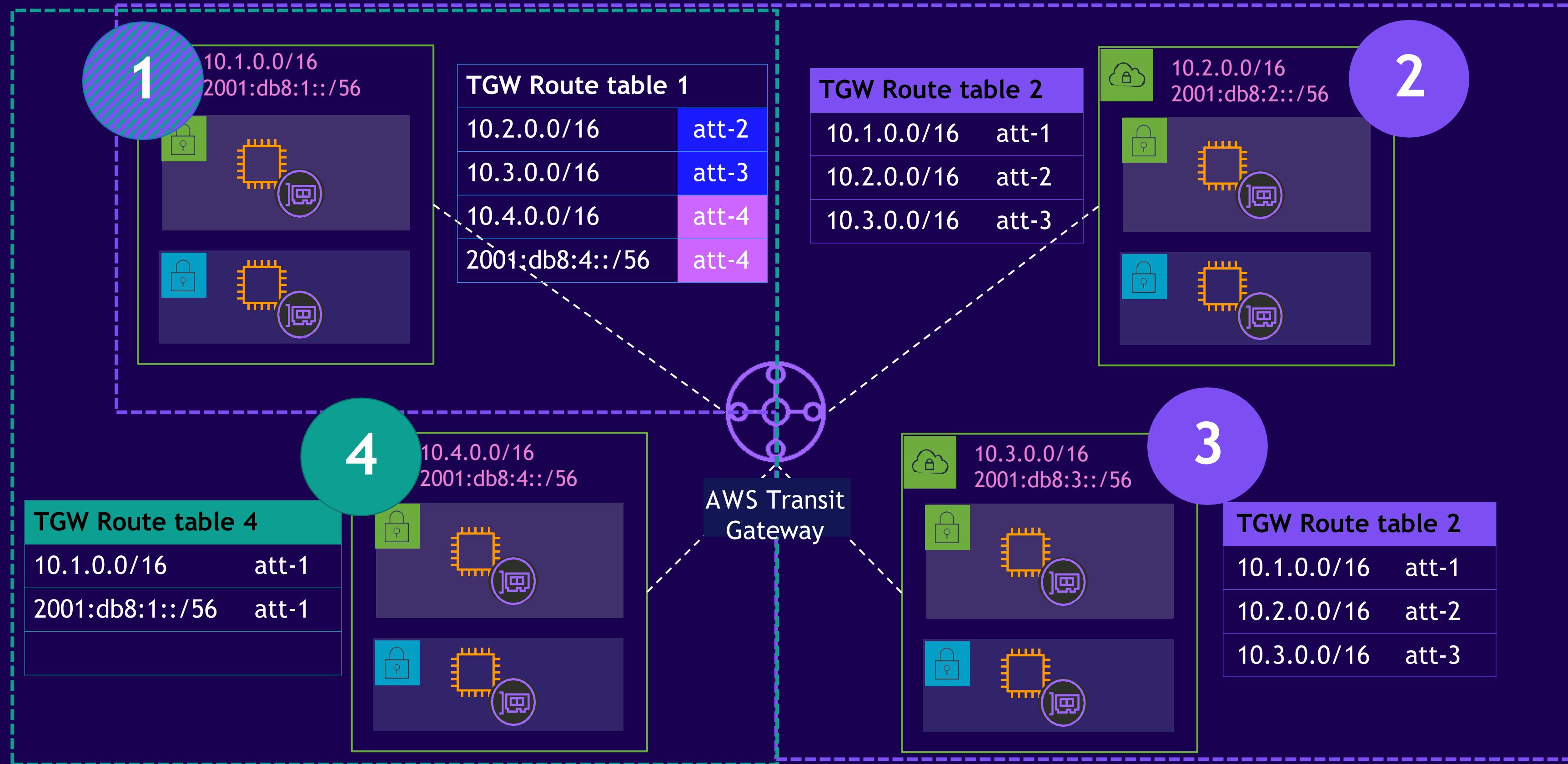
Full Connectivity



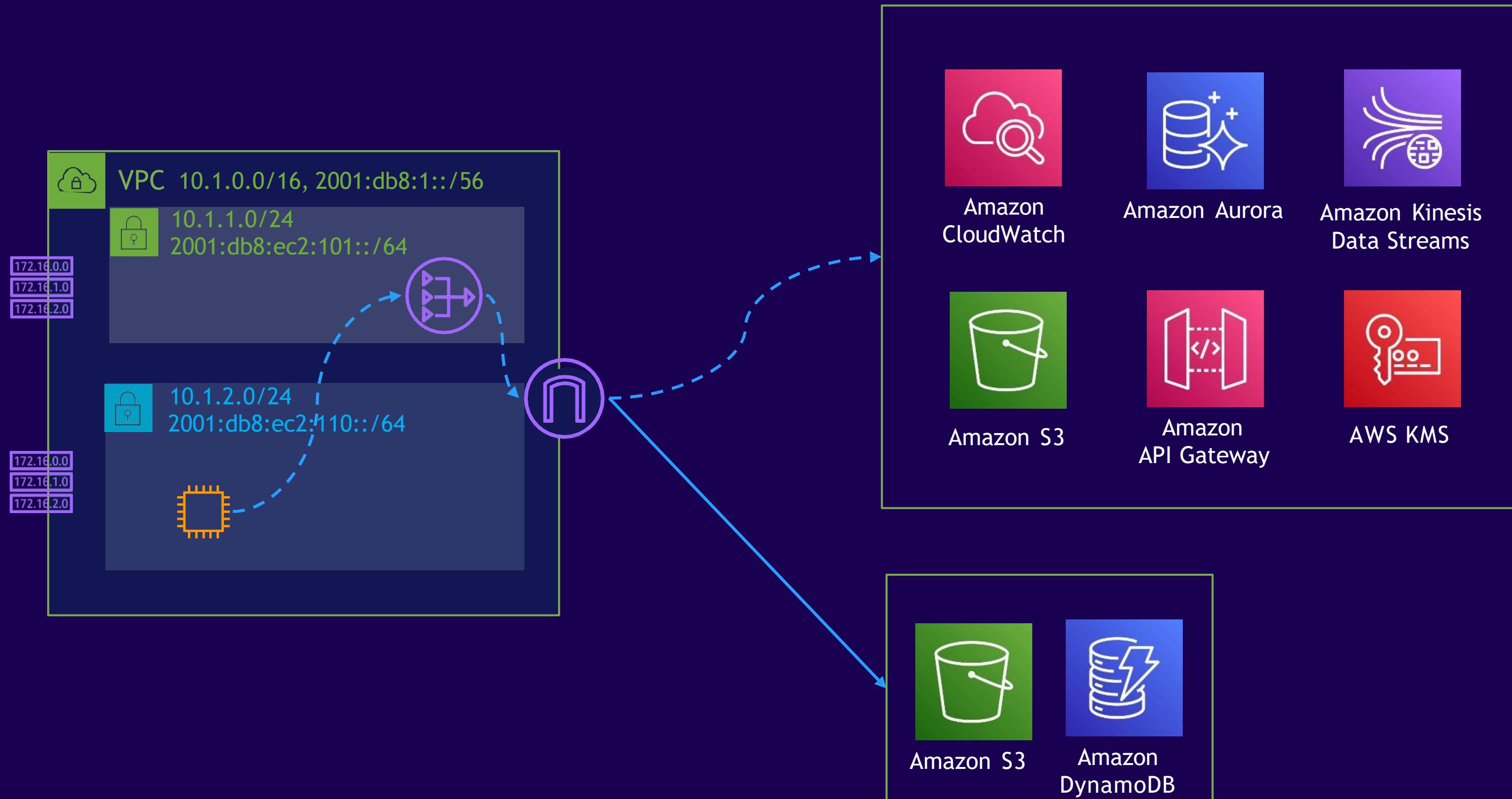
Partial connectivity



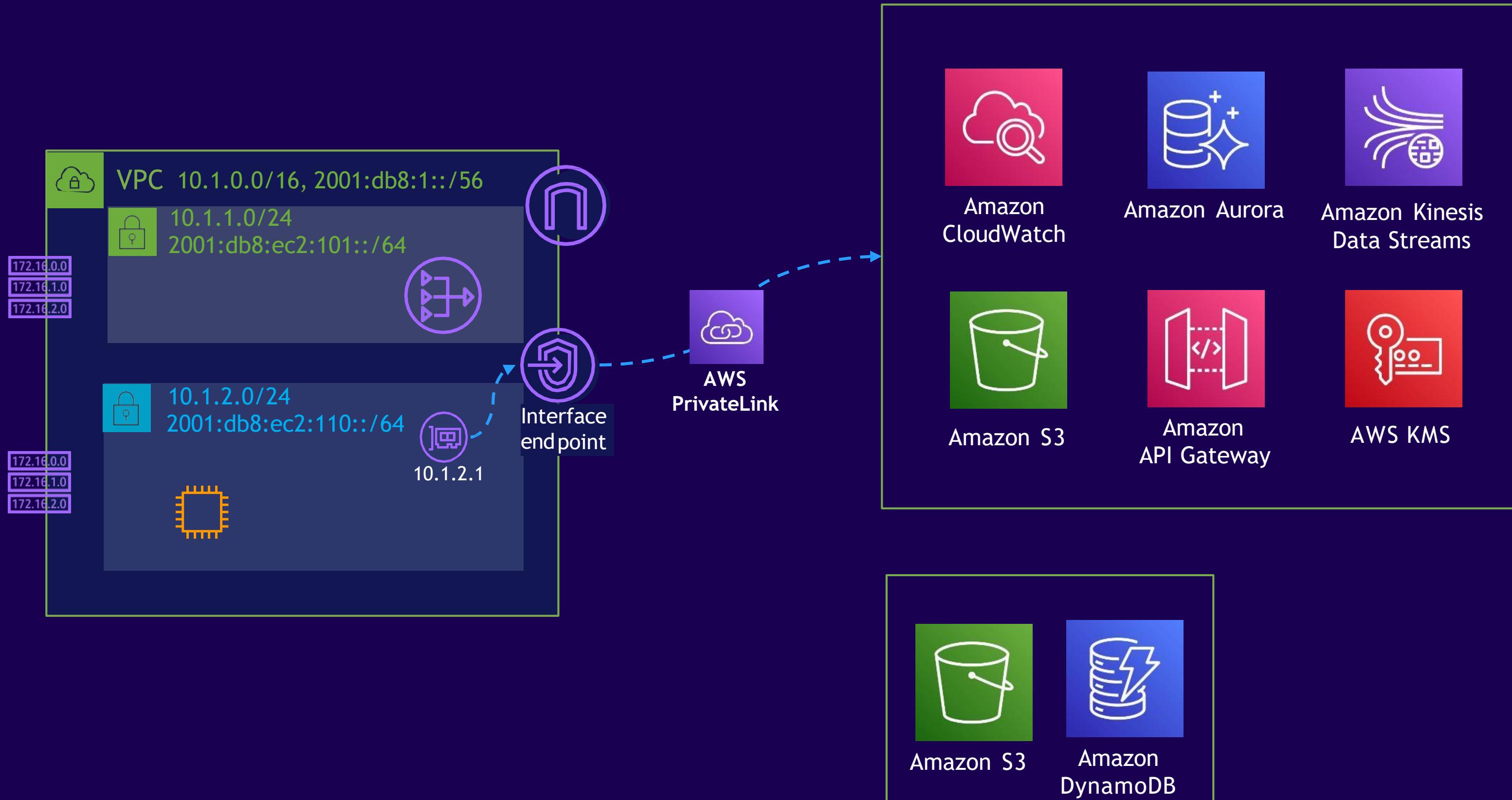
Transit Gateway route tables and domains



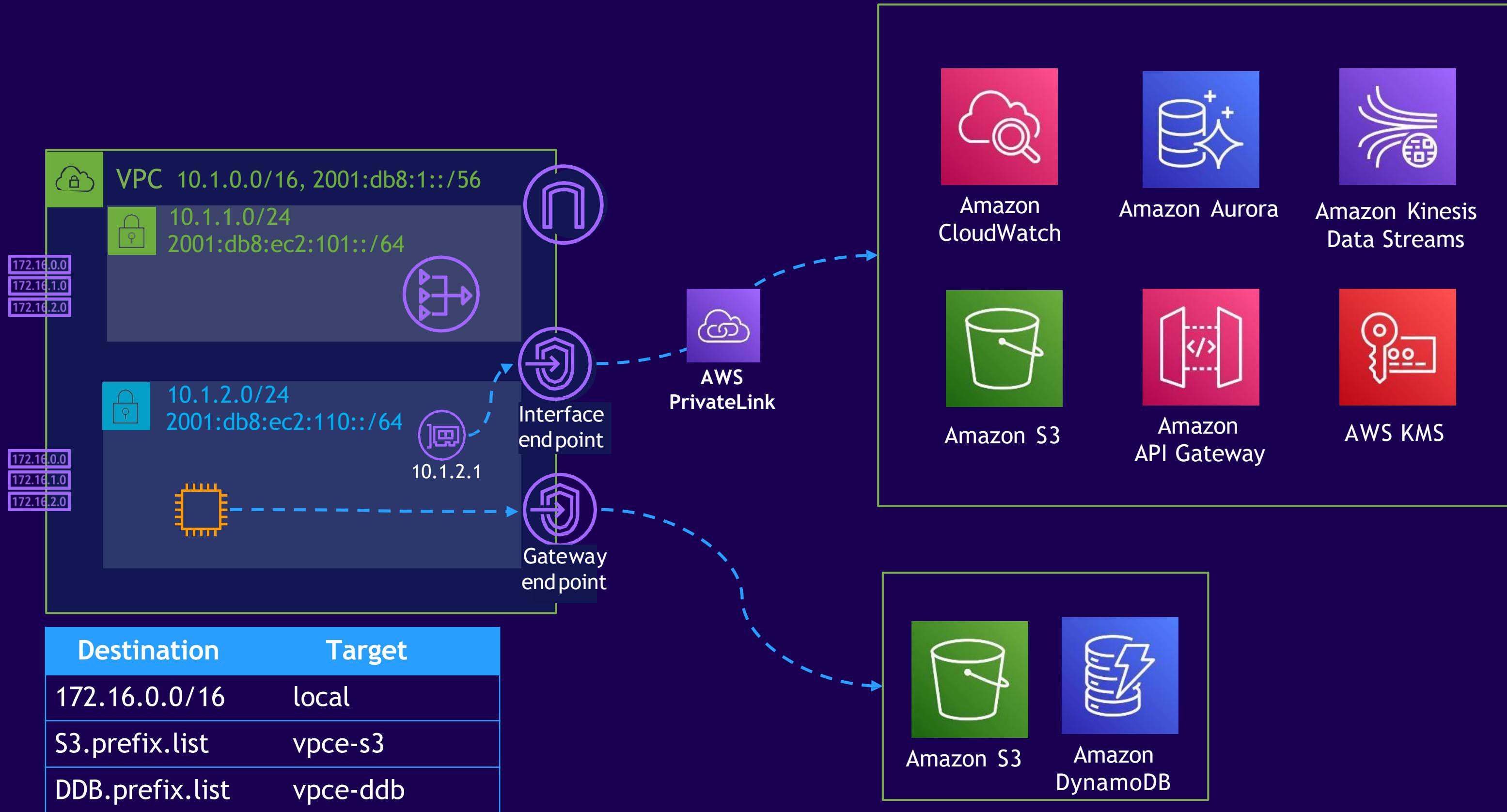
Without VPC endpoints



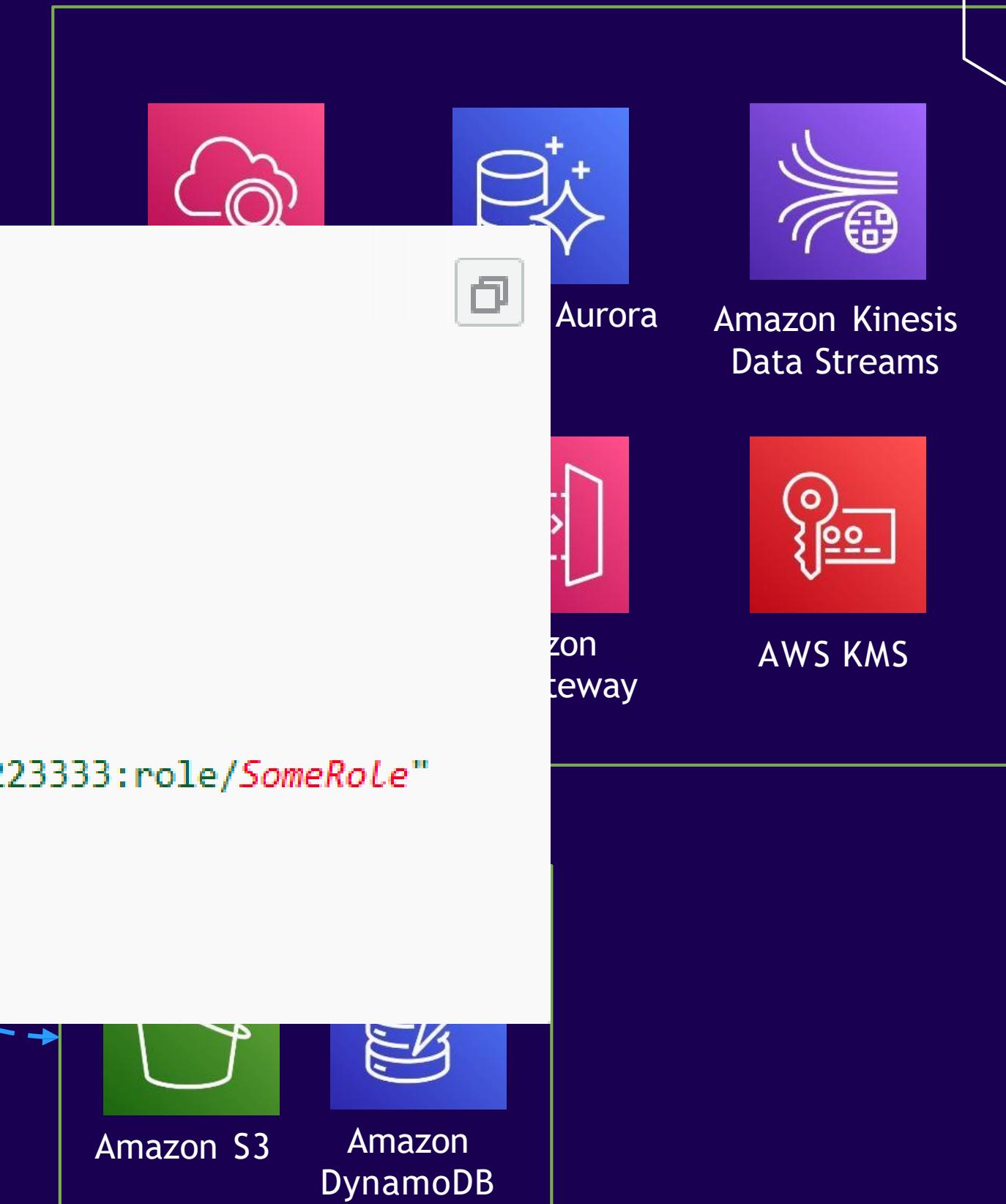
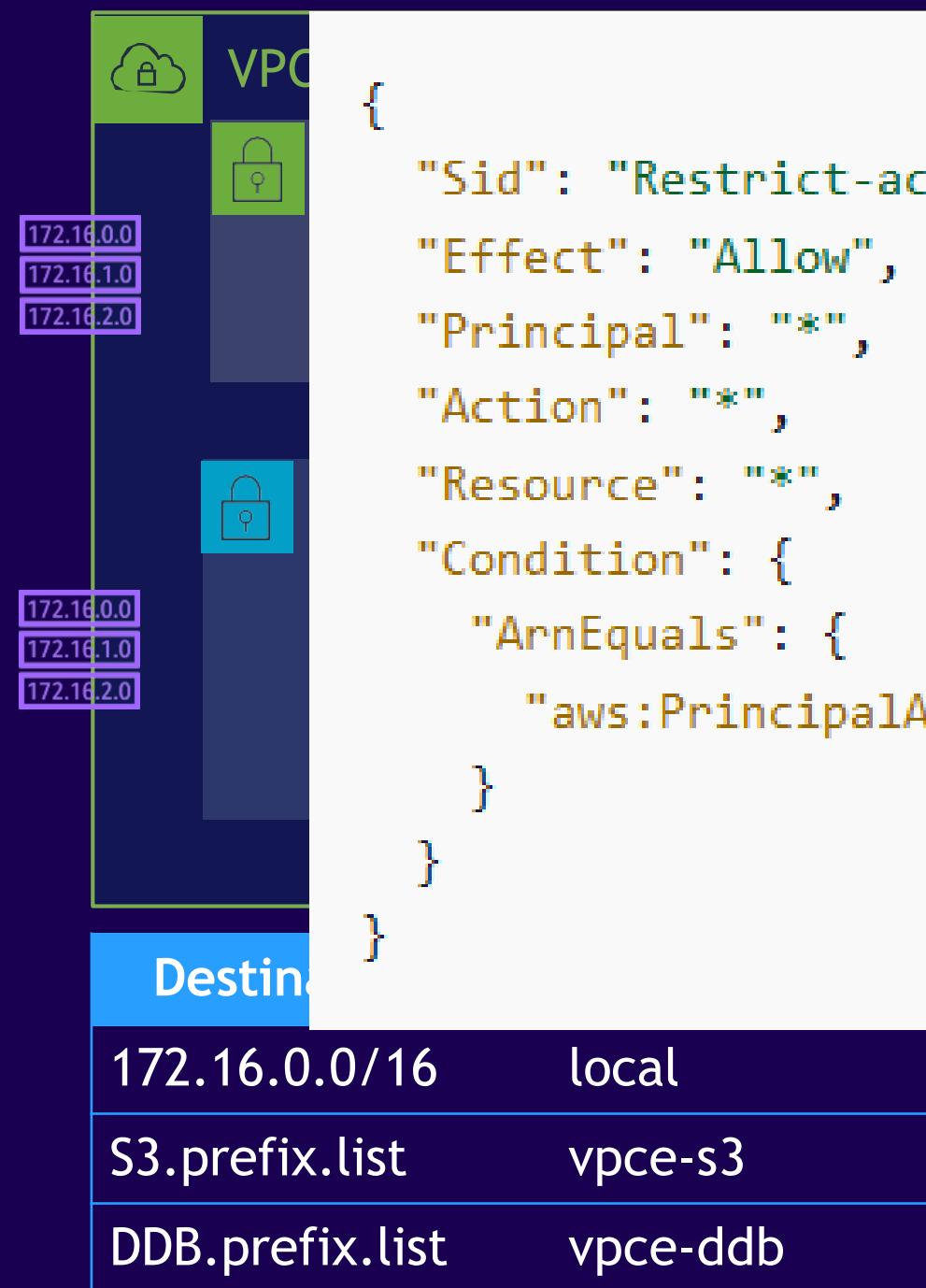
With VPC endpoints: Interface endpoints



With VPC endpoints: Gateway endpoints



With VPC endpoints: gateway endpoints



Hybrid connectivity and gateways

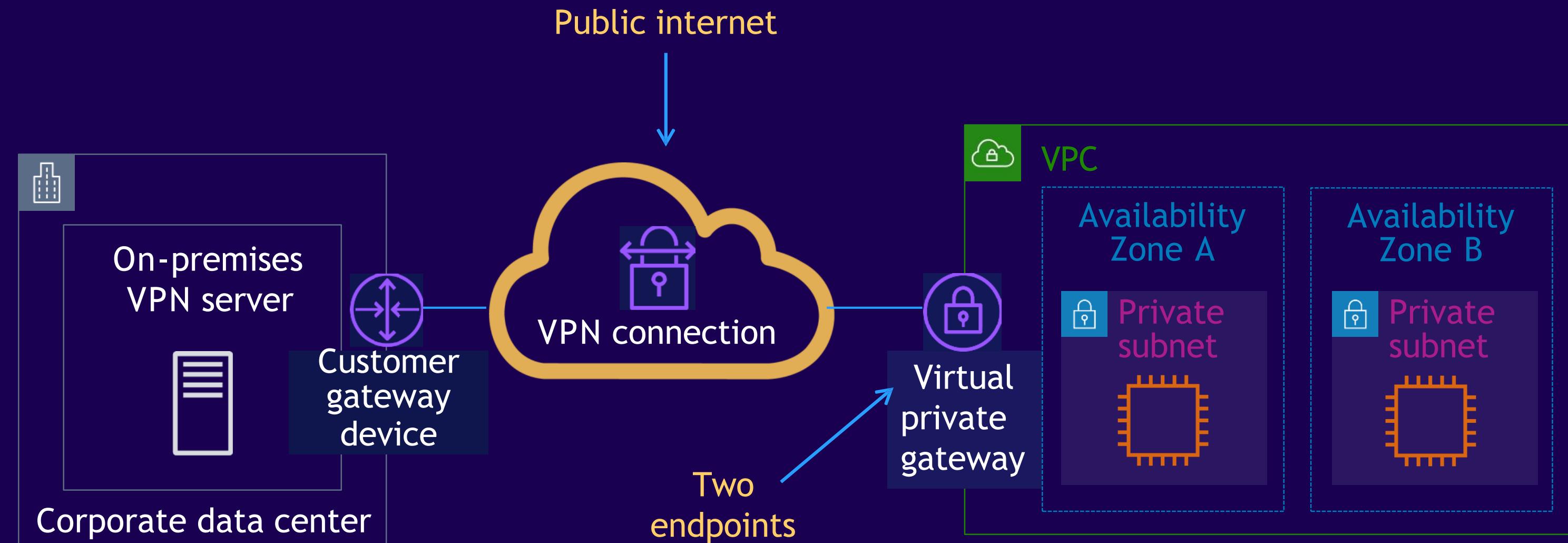


AWS Site-to-Site VPN

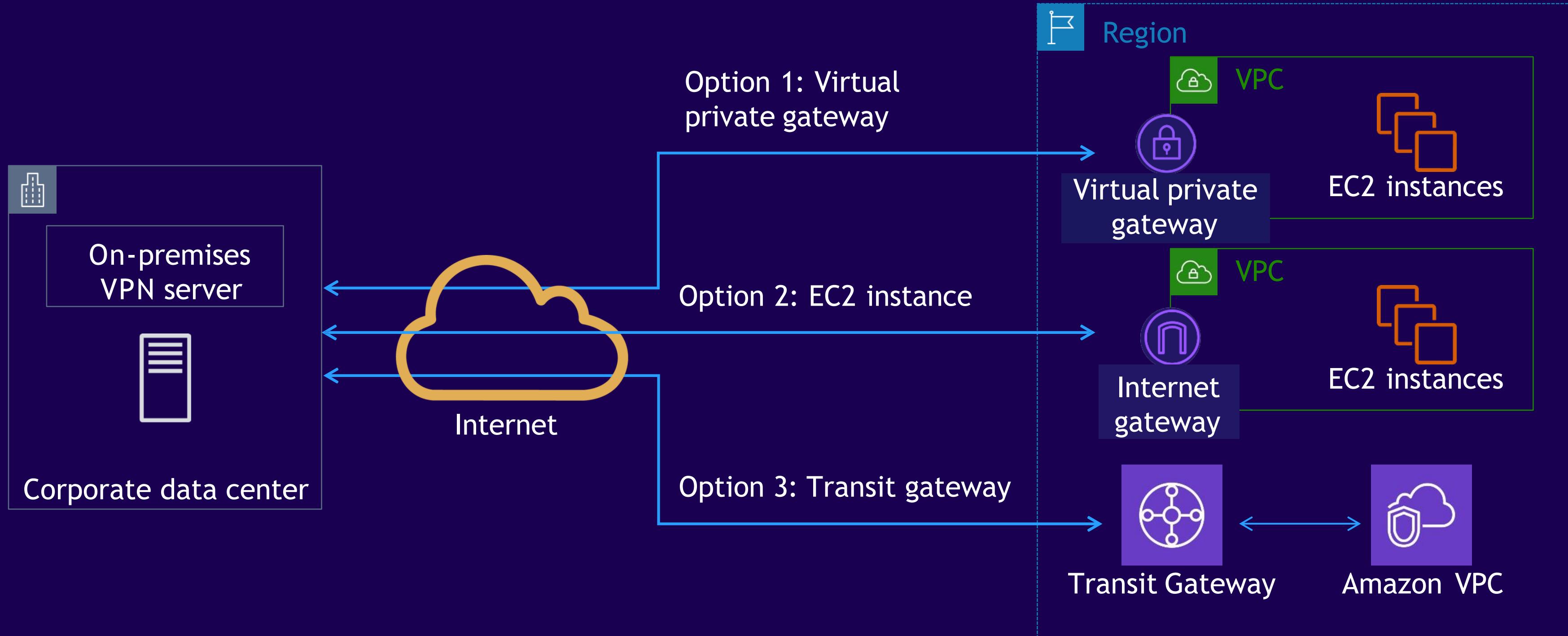


AWS Direct Connect

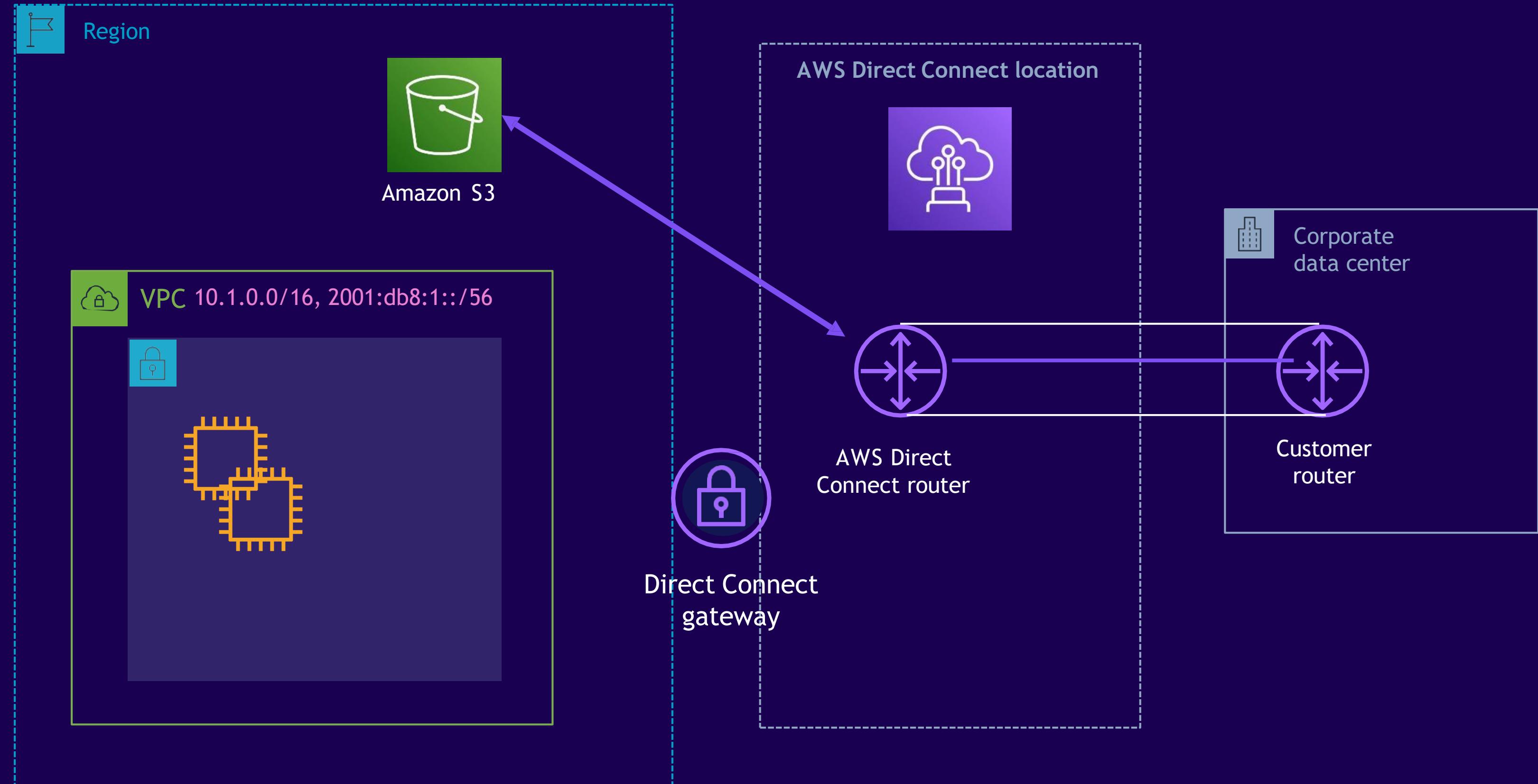
AWS Site-to-Site VPN



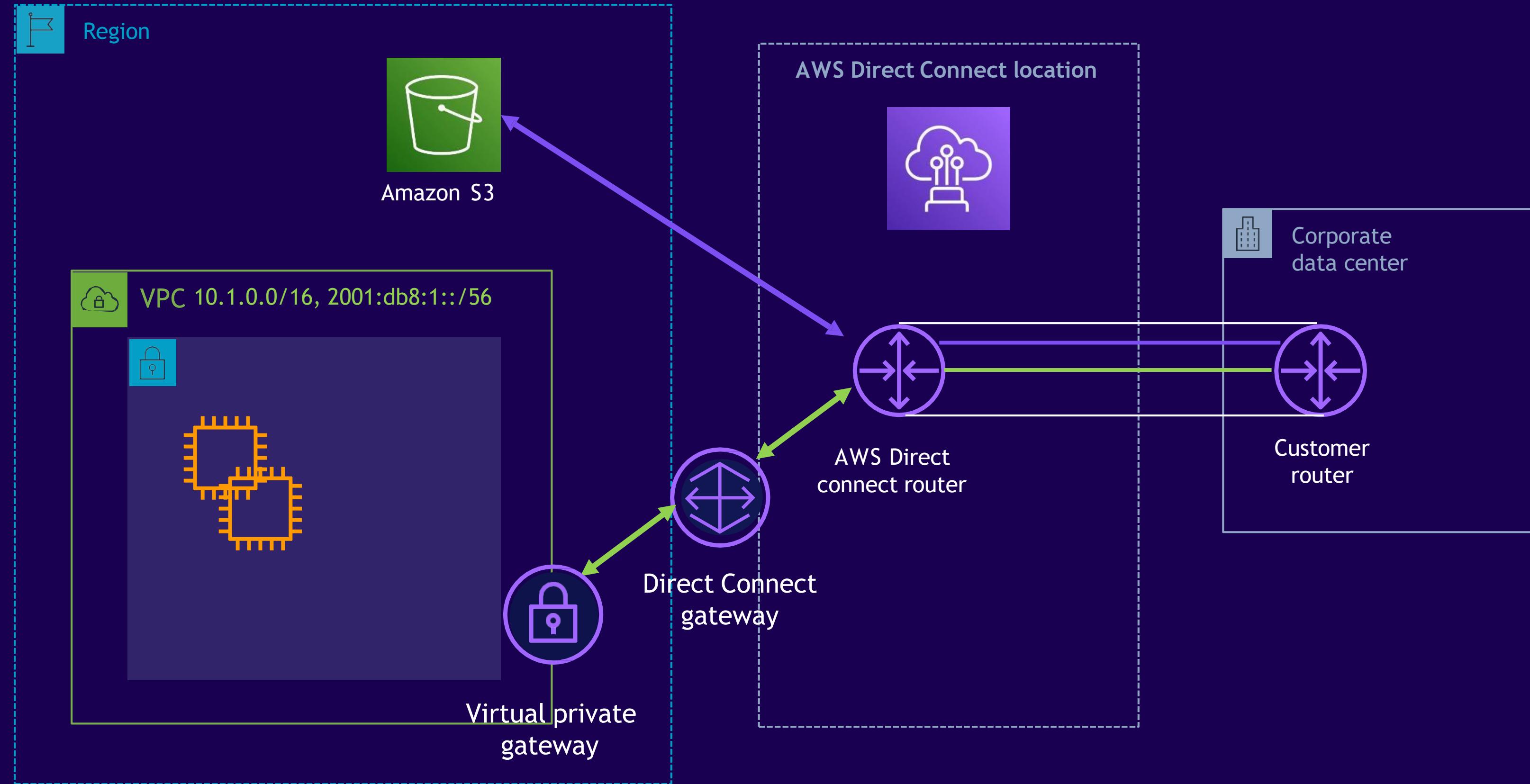
AWS Site-to-Site VPN



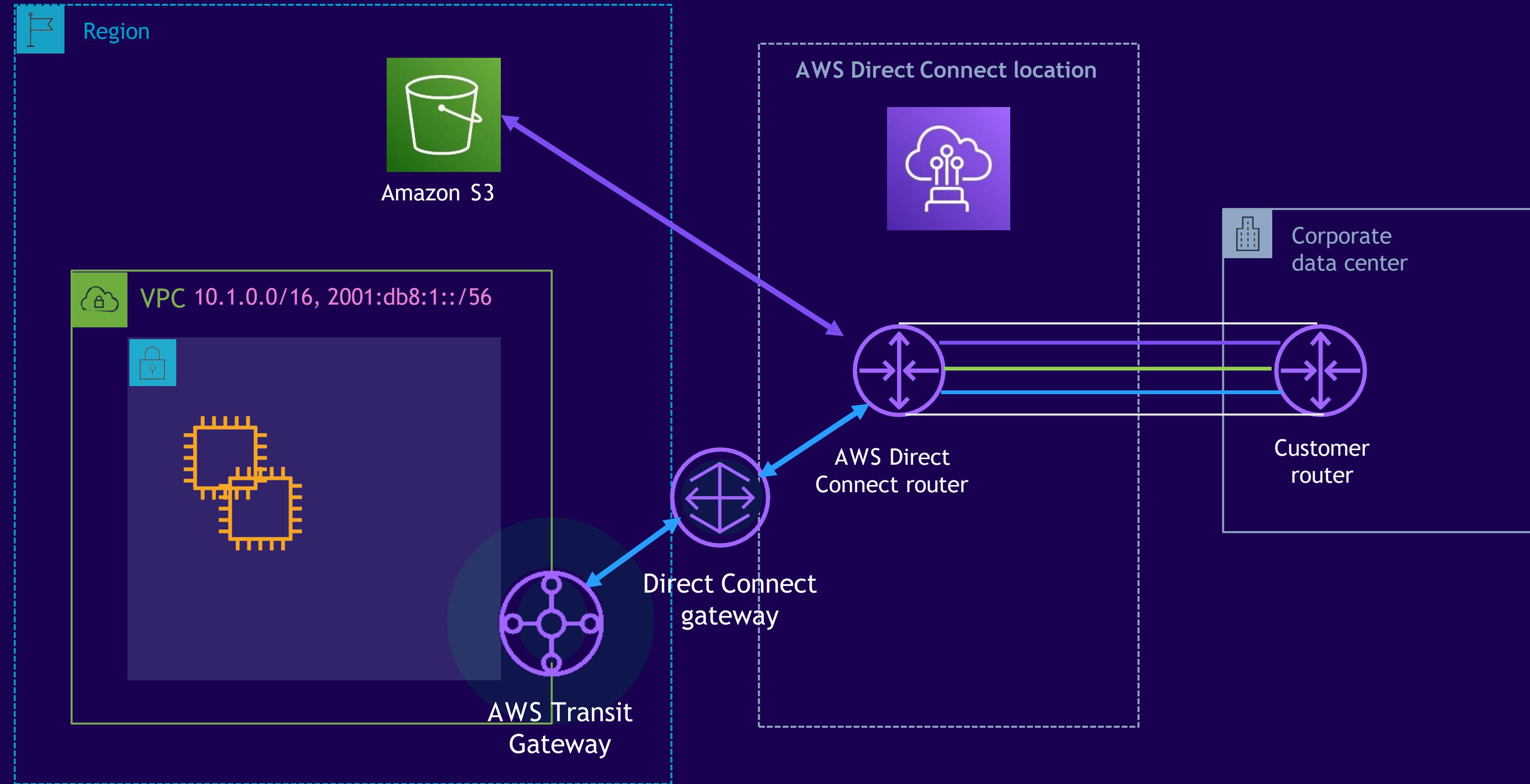
AWS Direct Connect



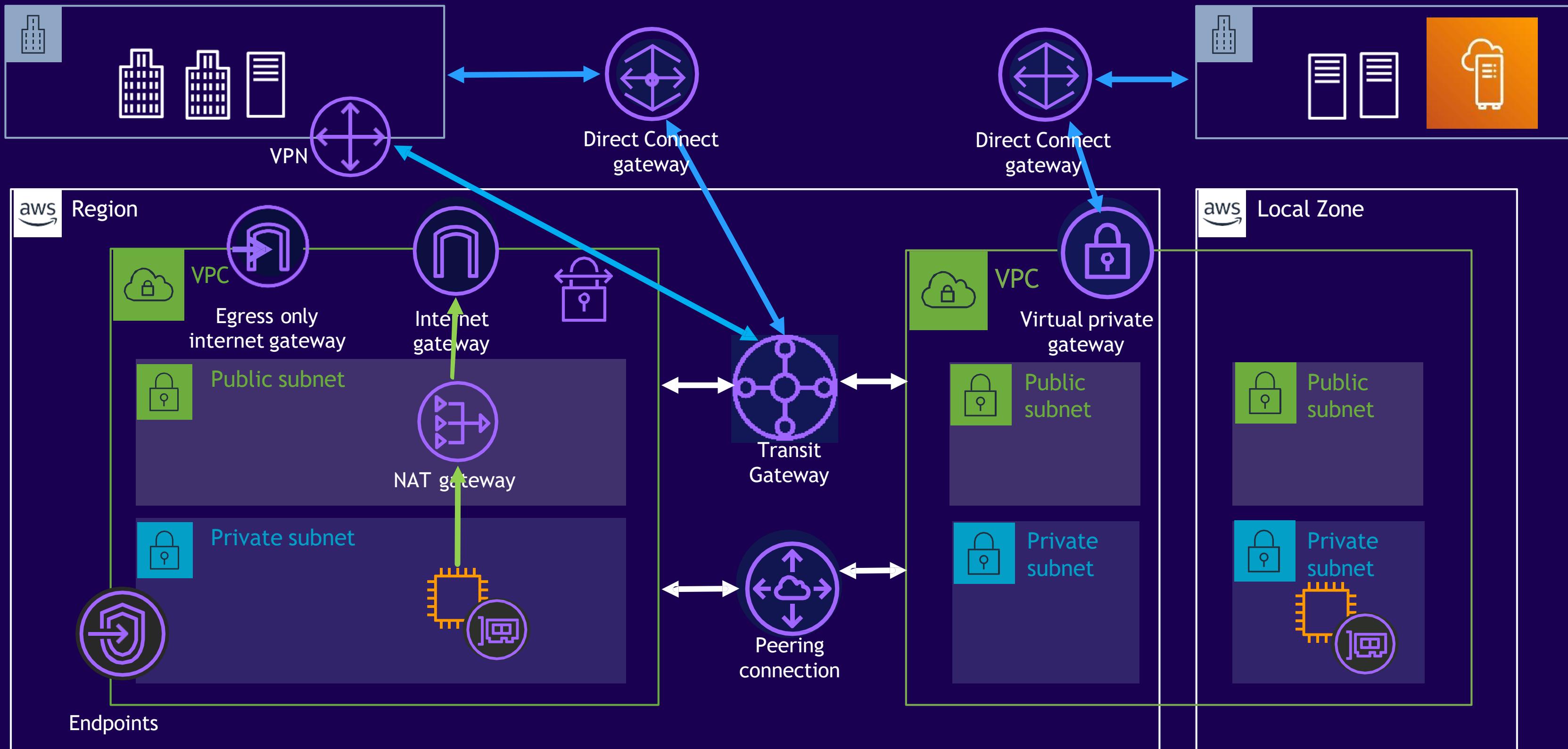
AWS Direct Connect



AWS Direct Connect



Bringing it all together



Route53

Route53 - DNS Background

- .The IPv4 space is a 32 bit field and has over 4 billion different addresses (4,294,967,296 to be precise).

Class A,B,C

0.0.0.0 - 255.255.255.255 256X256x256x256

10.0.1.0/24

- .IPv6 was created to solve this depletion issue and has an address space of 128bits which in theory is 340,282,366,920,938,463,463,374,607,431,768,211,456 addresses or 340 undecillion addresses.

Route53 - DNS Background

The SOA record stores information about:

- The name of the server that supplied the data.
- The administrator.
- The current version of the data file.
- The number of seconds a secondary name server should wait before checking for updates.
- The number of seconds a secondary name server should wait before retrying a failed zone transfer.
- The maximum number of seconds that a secondary name server can use data before it must either be refreshed or expire.
- The default number of seconds for the time-to-live file on resource records.

Route53 - DNS Background

NS Records:

NS stands for Name Server records and are used by Top Level Domain servers to direct traffic to the Content DNS server which contains the authoritative DNS records.

A Records:

An "A" record is the fundamental type of DNS record and the "A" in A record stands for "Address". The A record is used by a computer to translate the name of the domain to the IP address. For example <http://www.zekelabs.com> might point to <http://123.10.10.80>.

TTL:

The length that a DNS record is cached on either the Resolving Server or the users own local PC is equal to the value of the "Time To Live" (TTL) in seconds. The lower the time to live, the faster changes to DNS records take to propagate throughout the internet.

Route53 - DNS Background

CNames Records:

A Canonical Name (CName) can be used to resolve one domain name to another. For example, you may have a mobile website with the domain name <http://m.google.com> that is used for when users browse to your domain name on their mobile devices. You may also want the name <http://mobile.google.com> to resolve to this same address.

Alias Records:

Alias records are used to map resource record sets in your hosted zone to Elastic Load Balancers, CloudFront distributions, or S3 buckets that are configured as websites.

Alias records work like a CNAME record in that you can map one DNS name (example.com) to another ‘target’ DNS name 172.31.45.0 (elb1234.elb.amazonaws.com).

Key difference - A CNAME can't be used for naked domain names (zone apex). It must be either an A record or an Alias

Route53 - DNS Background

- Alias resource record sets can save you time because Amazon Route-53 automatically recognizes changes in the record sets that the alias resource record set refers to.
- For example, suppose an alias resource record set for zekelabs.com points to an ELB load balancer at elb1-1234.us-east.elb.amazonaws.com. If the IP address of the load balancer changes, Amazon Route 53 will automatically reflect those changes in DNS answers for example.com without any changes to the hosted zone that contains resource record sets for example.com.

Route53 Policy Types

Simple	Default. Used when single record
Weighted	Allows traffic routing based on weights
Failover	Make one Active/passive on failure. Automatic Need to create Health Check Status
Latency	Route Traffic to site with lowest latency
Geolocation	Send traffic based on region

Knowledge check

Which of the following are layers of network defense for VPCs?
(choose three)

- A. Amazon Machine Images (AMIs)
- B. Network access control lists (subnet level)
- C. Security groups (instance level)
- D. S3 lifecycle policies
- E. VPC route tables

Knowledge check

Which of the following are layers of network defense for VPCs?
(choose three)

- A. ~~Amazon Machine Images (AMIs)~~
- B. Network access control lists (subnet level)
- C. Security groups (instance level)
- D. ~~S3 lifecycle policies~~
- E. VPC route tables

Answer: B, C, E

Key Takeaways

Amazon VPC provides:

- Logically isolated network to launch applications
- Security Groups, network ACLs and route tables to secure your deployments

There are three main ways customers connect to AWS:

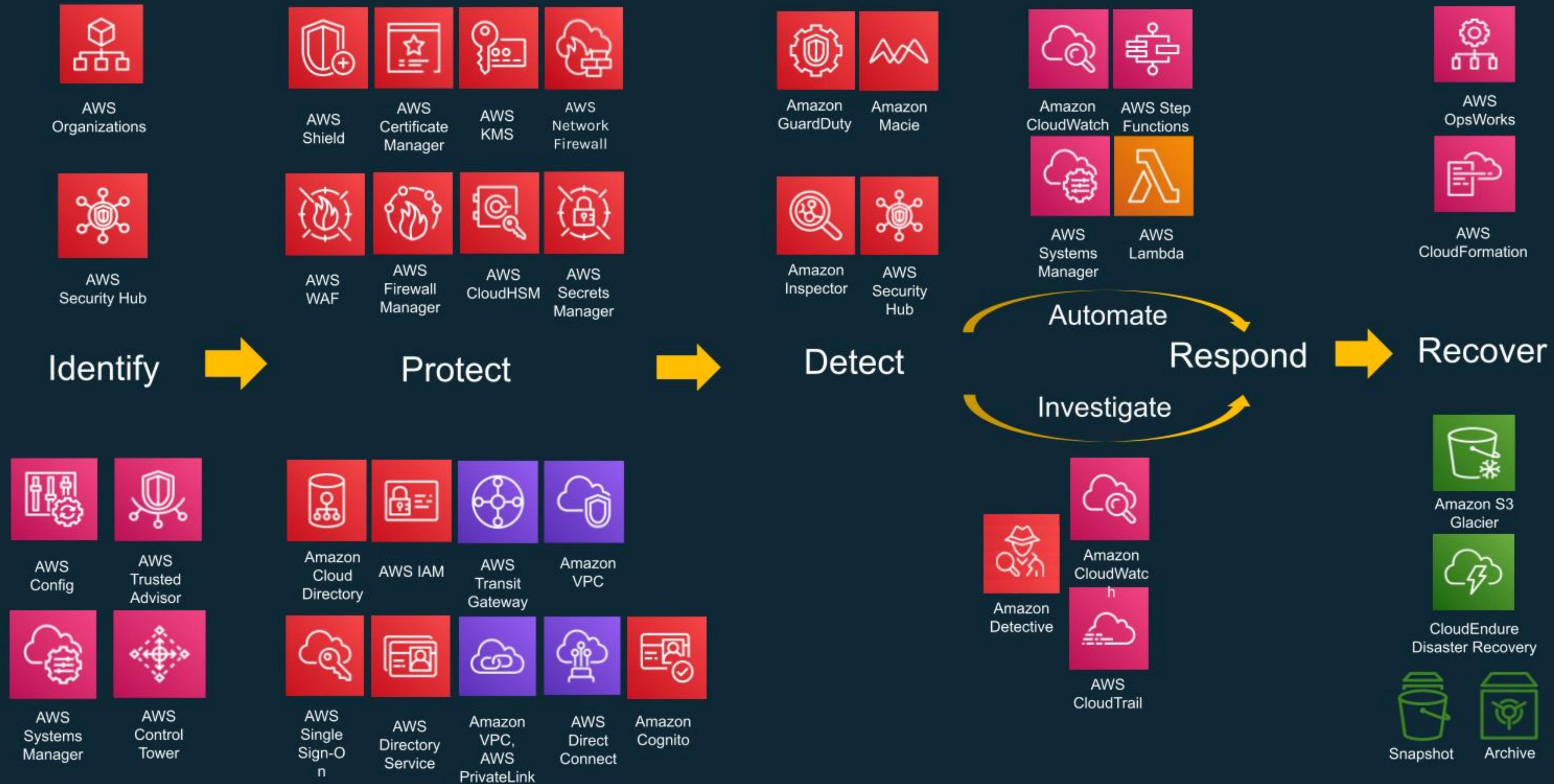
- Client VPN
- Site-to-site VPN
- Direct Connect

Security

AWS Security Tools



AWS foundational and layered security services



AWS Identity and Access Management (IAM)

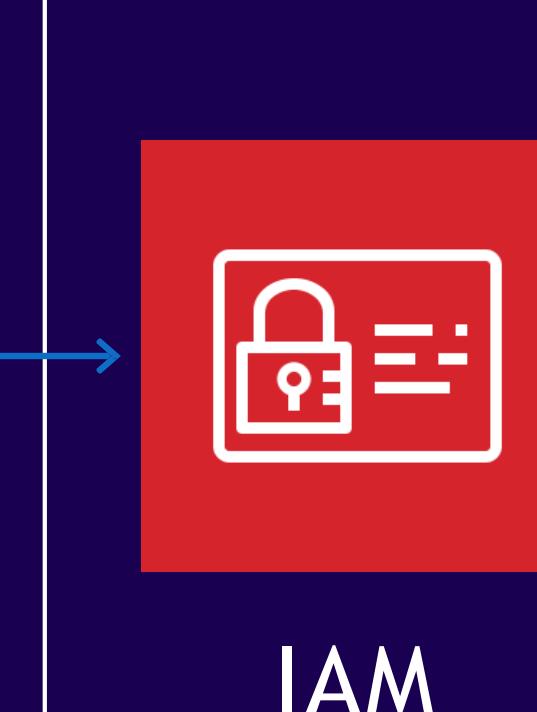
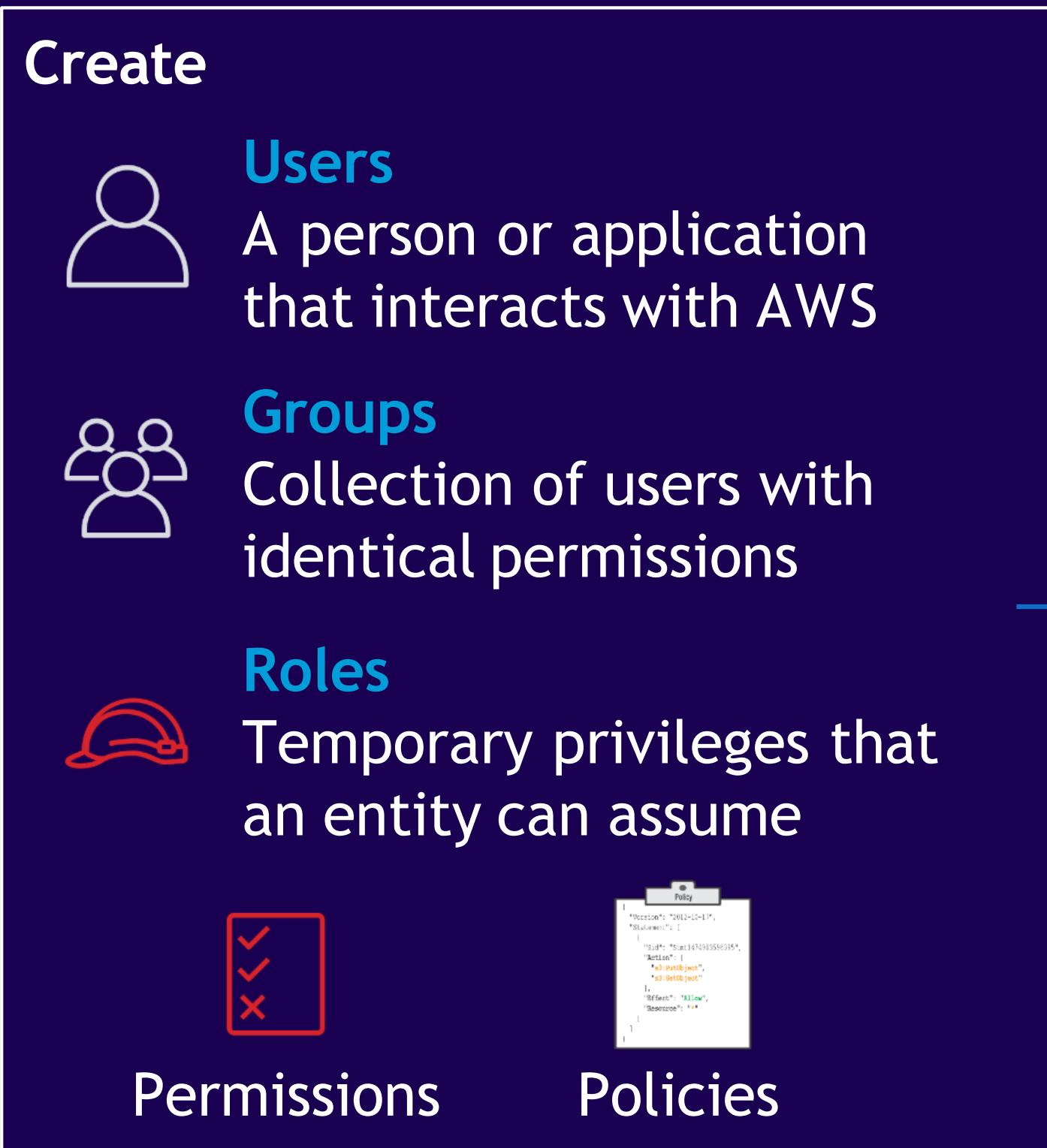


IAM

Securely control access to your AWS resources

- Assign granular permissions to users, groups, or roles
- Share temporary access to your AWS account
- Federate users in your corporate network or with an internet identity provider

IAM components



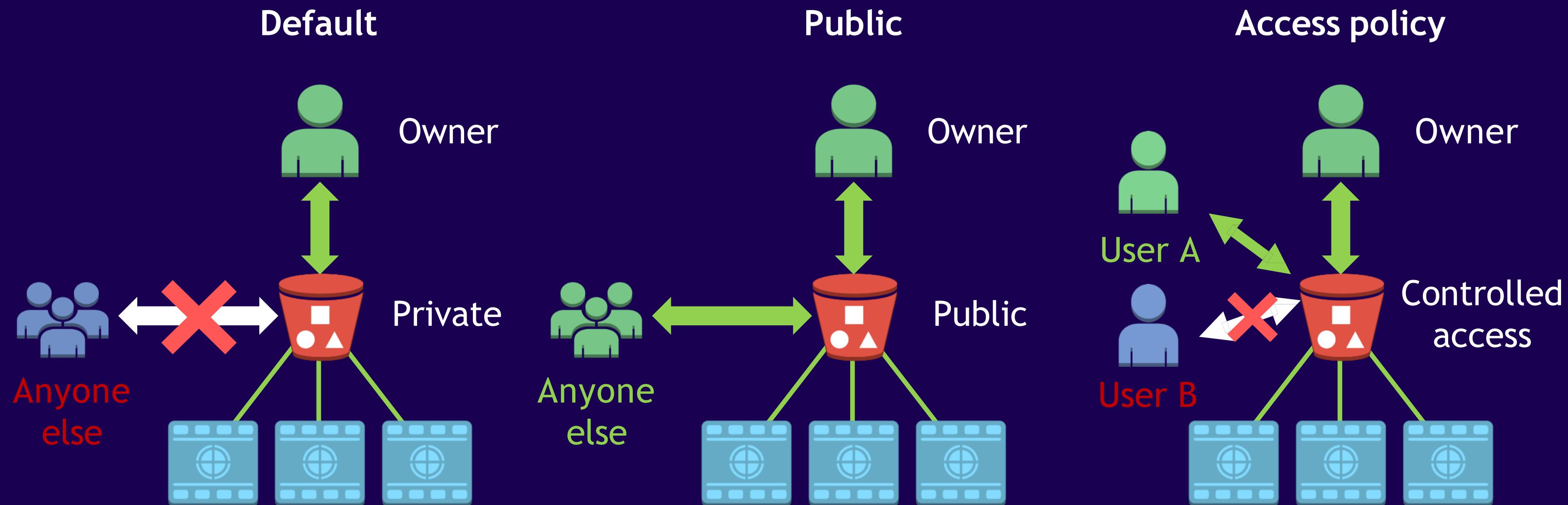
Defines permissions to control which AWS resources users can access

Helps you to meet identity and access control standards

- Authentication
- Authorization

Amazon S3 access control: General

Some services support resource-based policies, such as S3 bucket policies



AWS CloudTrail



AWS
CloudTrail

Track user activity and API usage in your AWS account

- Continuously monitor user activities and record API calls
- Useful for compliance auditing, security analysis, and troubleshooting
- Log files are delivered to Amazon S3 buckets

Who?

What?

When?

Where?

API security-relevant information

What is AWS Trusted Advisor?



AWS
Trusted Advisor

A service providing guidance to help you
reduce cost, increase performance, and
improve security

Security

**Shared Security Model
Tools and Services for
Securing EC2, S3, RDS, VPC and Networks
Key Management Services
AWS IAM and Security Roles/Permissions**

Security is a top priority for AWA



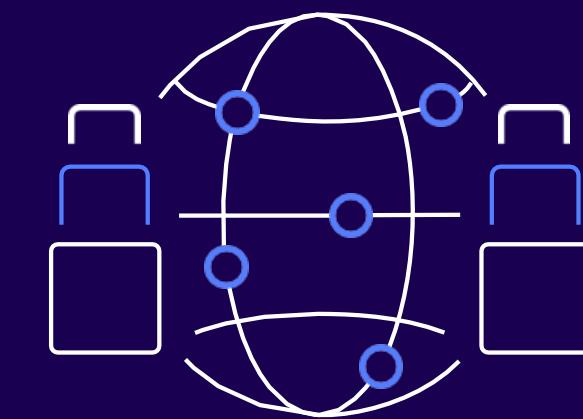
Designed for
security



Constantly
monitored



Highly
automated



Highly
available



Highly
accredited

AWS Shared Responsibility Model

You

Customer content

Platform, Applications, Identity & Access Management

Operating System, Network & Firewall Configuration

Client-side Data
Encryption

Server-side Data
Encryption

Network Traffic
Protection

Customers are
responsible for
their security and
compliance **IN**
the Cloud

AWS Foundation Services

Compute

Storage

Database

Networking

AWS is
responsible for
the security **OF**
the Cloud

**AWS Global
Infrastructure**

Availability Zones

Regions

Edge
Locations



AWS strengthens your security posture



“We work closely with AWS to develop a security model, which we believe enables us to operate more securely in the public cloud than we can in our own data centers.”

Rob Alexander - CIO, Capital One



Over 50 global
compliance
certifications and
accreditations



Benefit from AWS
industry leading
security teams 24/7,
365 days a year



Security infrastructure
built to satisfy military,
global banks, and other
high-sensitivity
organizations



Leverage security
enhancements from
1M+ customer
experiences

Security Audits: On-premises vs. On AWS

- Start with bare concrete
 - Functionally optional - you can build a secure system without it
 - Audits done by an in-house team
Accountable to yourself
 - Typically check once a year
 - Workload-specific compliance checks
 - Must keep pace and invest in security innovation
- Start on base of accredited services
 - Functionally necessary - high watermark of requirements
 - Audits done by third party experts
Accountable to everyone
 - Continuous monitoring
 - Compliance approach based on all workload scenarios
 - Security innovation drives broad compliance

on-prem

On AWS

What this means

- You benefit from an environment built for the most security sensitive organizations
- AWS manages 1,800+ security controls **so you don't have to**
- You get to define the right security controls for your workload sensitivity
- You always have full ownership and control of your data

AWS: more assurance programs than anyone

Certifications / Attestations	Laws, Regulations, and Privacy	Alignments and Frameworks
ISO 27001	HIPAA	CJIS
ISO 27017	IRS 1075	FISMA
ISO 27018	ITAR	GxP
PCI DSS Level 1	FERPA	CLIA
DoD SRG	CS Mark [Japan]	CMS Edge
FedRAMP	DNB [Netherlands]	FISC [Japan]
FIPS	EAR	FDA
IRAP [Australia]	Gramm-Leach-Bliley Act (GLBA)	MPAA
MLPS Level 3 [China]	HITECH	CMSR
MTCS Tier 3 [Singapore]	My Number Act [Japan]	FedRAMP TIC
SEC Rule 17a-4(f)	DPA – 1998 [U.K.]	G-Cloud [U.K.]
SOC 1, SOC 2, SOC 3	VPAT / Section 508	PHR
	EU Data Protection Directive [EU]	IT Grundschutz [Germany]
	Privacy Act [Australia & New Zealand]	MITA 3.0
	PDPA – 2010 [Malaysia & Singapore]	NERC
		NIST

Meet your own security objectives

You

Your own accreditation



Your own certifications



Your own external audits



Customer scope and effort is **reduced**

Better results through focused efforts

AWS Foundation Services

Compute

Storage

Database

Networking

Built on AWS **consistent** baseline controls

AWS Global Infrastructure

Availability Zones

Regions

Edge Locations



Data Locality

- Customer chooses **where to place data**
- AWS regions are geographically isolated by design
- **Data is not replicated to other AWS regions** and doesn't move unless you choose to move it

Data Locality in practice

- **Block level storage**
 - Instance Storage (Elastic Cloud Compute - EC2)
 - Elastic Block Storage (EBS)
- **Object level storage**
 - Simple Storage Service (S3)
- **Database storage**
 - Relational Databases (RDS)
 - NoSQL (DynamoDB)
 - Data Warehouse (Redshift)
 - Caching (Elasticache)

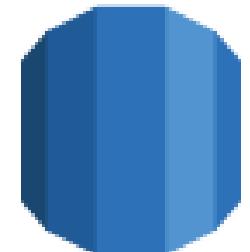
AWS Shared Responsibility Model Deep Dive

One model for all?

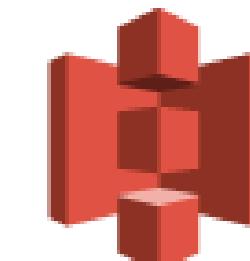
Infrastructure
Services



Managed
Services



Abstract
Services



AWS Shared Responsibility Model: for Infrastructure Services

Customer content

Platform & Applications Management

Operating System, Network & Firewall Configuration

Client-Side Data encryption
& Data Integrity Authentication

Server-Side Encryption
File System and/or Data

Network Traffic Protection
Encryption / Integrity / Identity

Optional – Opaque data: 1's and 0's (in transit/at rest)

AWS Foundation Services

Compute

Storage

Database

Networking

AWS Global Infrastructure

Availability Zones

Regions

Edge Locations

Customer IAM

AWS IAM

API Endpoints

Mgmt
Protocols

API
Calls

Managed by



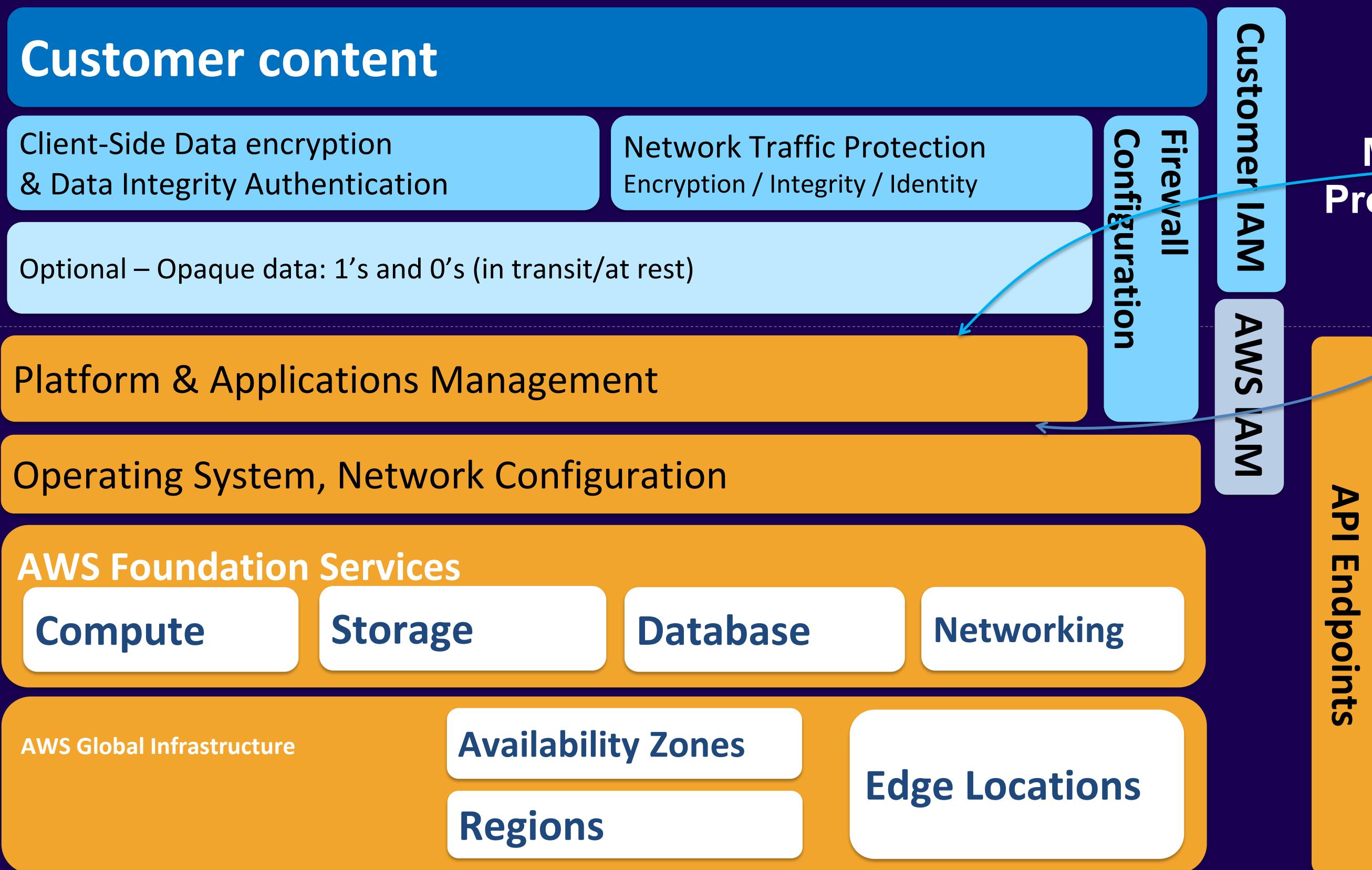
Customers

Managed by



AWS Shared Responsibility Model: for Container Services

Managed by



AWS Shared Responsibility Model: for Abstract Services

Customer content

(optional)

Opaque Data: 1's and 0's

(in flight / at rest)

Client-Side Data Encryption
& Data Integrity Authentication

Data Protection by the Platform
Protection of Data at Rest

Network Traffic Protection by the Platform
Protection of Data at in Transit

Platform & Applications Management

Operating System, Network & Firewall Configuration

AWS Foundation Services

Compute

Storage

Database

Networking

AWS Global Infrastructure

Availability Zones

Regions

Edge Locations

AWS IAM

API Endpoints

API Calls

Managed by



Managed by



Customers

Our Responsibilities - Examples

EC2

Customer Data
Customer Application
Operating System
Network and Firewall
Customer IAM
High Availability/Scaling
Instance Management
Data Protection
AWS IAM

RDS

Customer Data
Firewall (VPC)
Customer IAM
AWS IAM
High Availability/Scaling
Data Protection

S3

Customer Data
Data Protection (REST)
AWS IAM

Summary of Shared Responsibility in AWS

Infrastructure Services

Data

Customer IAM

AWS IAM

Applications

Operating System

Networking/Firewall

Managed Services

Data

Customer IAM

AWS IAM

Firewall

Abstract Services

Data

AWS IAM

The scale of cloud is the value





The power
of scale

1 quadrillion +

Metric observations monitored by Amazon
CloudWatch (**1,000,000,000,000+**)

230

Security, compliance, and governance
services and key features



The power
of scale

All services

Support server-side encryption

100%

Compliance with the General Data
Protection Regulation (GDPR)

Foundational security services



Amazon GuardDuty



Amazon Inspector



Amazon Macie

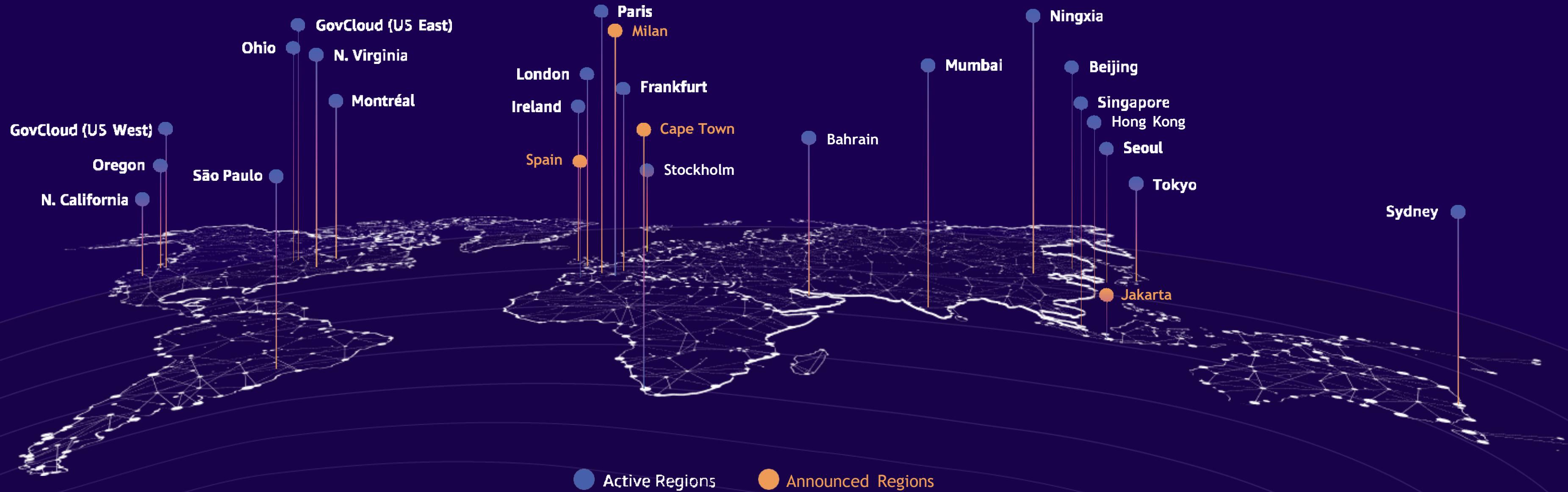


AWS Shield



AWS WAF

The scale of cloud is the value



Ten places your security group should spend time

- 1 Accurate account info
- 2 Use MFA
- 3 No hard-coding secrets
- 4 Limit security groups
- 5 Intentional data policies
- 6 Centralize AWS CloudTrail logs
- 7 Validate IAM roles
- 8 Take action on GuardDuty findings
- 9 Rotate your keys
- 10 Being involved in dev cycle

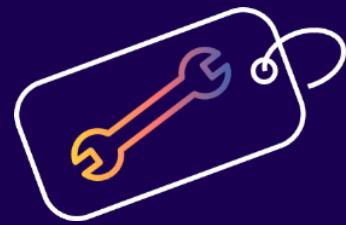
Ten places your security group should spend time

- 1 Accurate account info
- 2 Use MFA
- 3 No hard-coding secrets
- 4 Limit security groups
- 5 Intentional data policies
- 6 Centralize CloudTrail logs
- 7 Validate IAM roles
- 8 Take action on GuardDuty findings
- 9 Rotate your keys
- 10 Being involved in dev cycle

Ten places your security group should spend time

- 1 Accurate account info
- 2 Use MFA
- 3 No hard-coding secrets
- 4 Limit security groups
- 5 Intentional data policies
- 6 Centralize CloudTrail logs
- 7 Validate IAM roles
- 8 Take action on GuardDuty findings
- 9 Rotate your keys
- 10 Being involved in dev cycle

The first five things to automate



Ticketing



IAM policies



Logging



Threat detection



Alerting

UPDATES



Amazon GuardDuty

Protect your AWS accounts and workloads with intelligent threat detection and continuous monitoring

NEW DETECTIONS - BE ALERTED WHEN:

Amazon Simple Storage Service (Amazon S3) block public access is disabled

Amazon S3 server access logging is disabled

User attempts to assign a highly permissive policy to themselves

UPDATES



Amazon CloudWatch

Amazon CloudWatch is a monitoring and observability service built for DevOps engineers, developers, site reliability engineers, and IT managers

+ VPC TRAFFIC MIRRORING:
Applies ML algorithms to baseline
Anomaly Detection available in all
commercial regions
Features cross-account & cross-region
dashboards

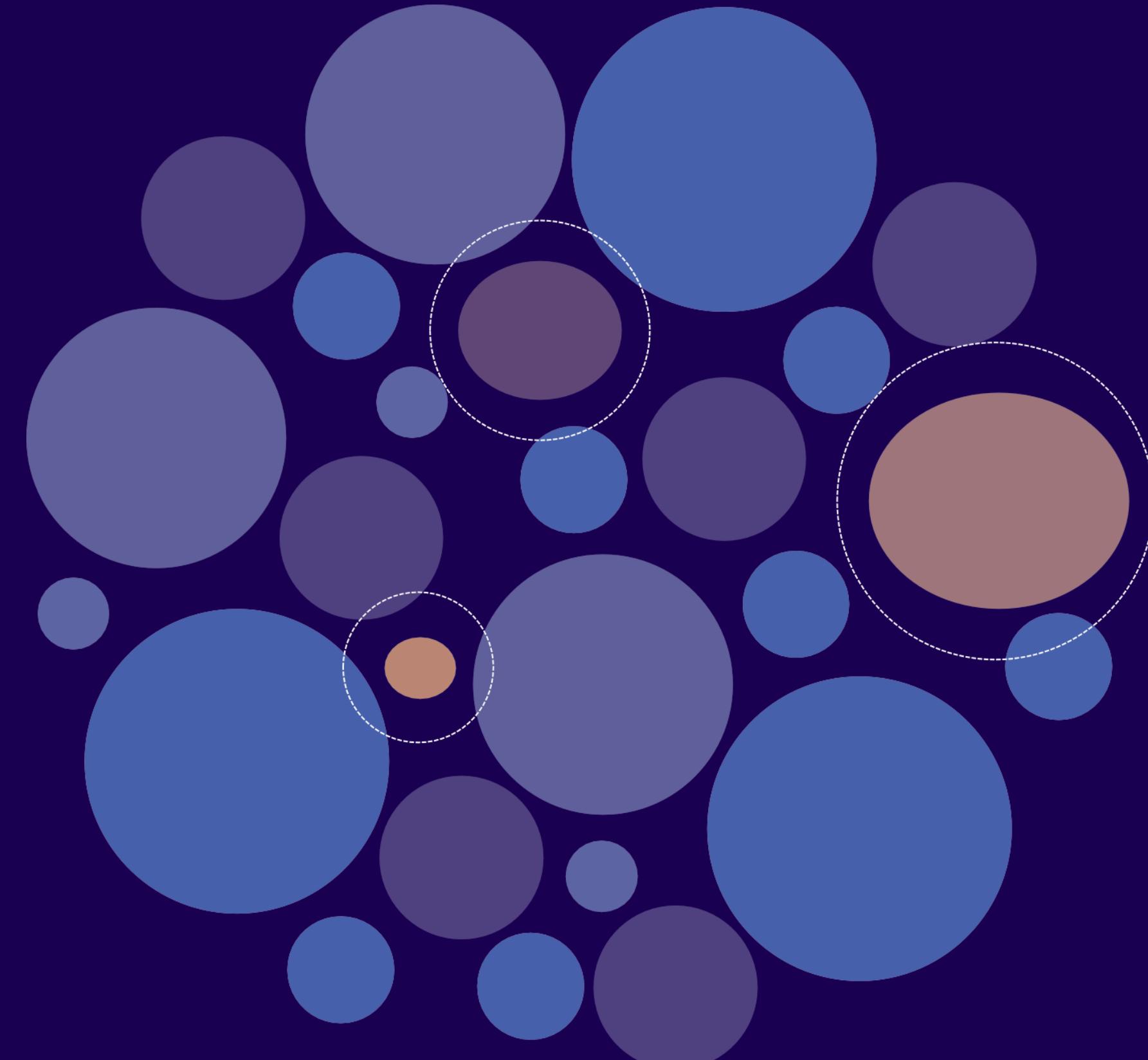
NOW IN PREVIEW



Amazon CloudWatch

Collects canary traffic, supports monitoring of REST APIs, URLs, and website content, checking for unauthorized changes from code injection and cross-site scripting

SYNTHETIC INSIGHTS



UPDATES



AWS Security Hub

Centrally view and manage security alerts and automate compliance checks

Single place to aggregate, organize, and prioritize security alerts

Use CloudWatch Events rules to send the findings to ticketing

Robust partner integrations

UPDATES



AWS Secrets Manager

Retrieve and manage secrets such as database credentials and API keys throughout their life cycle

Automatic rotation of secrets
Supports VPC endpoint policies
In-scope for SOC, HIPAA, PCI, and ISO

AWS IoT Secure Tunneling released



AWS IoT Device Defender - four new checks

PERMISSIONS CHECK

Do you have overly permissive permissions?

USAGE CHECK

Which services haven't been used in over 365 days?

VULNERABILITY CHECK

Find OpenSSL versions that have been identified as having predictable cryptographic keys

VERSION CHECK

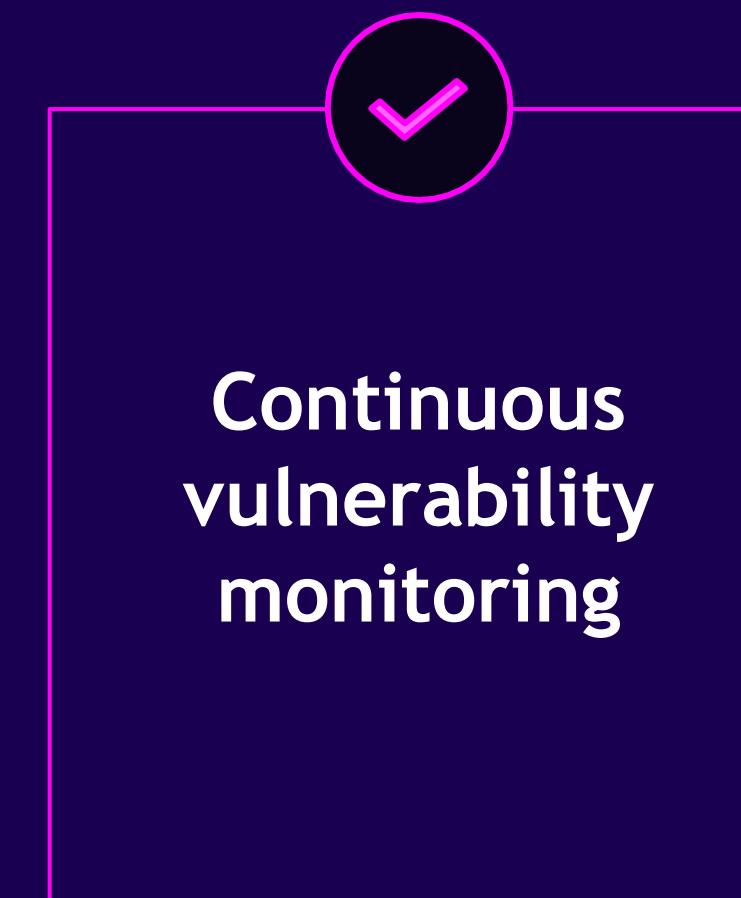
Find library versions that have been identified to mishandle RSA key generation

Amazon Inspector

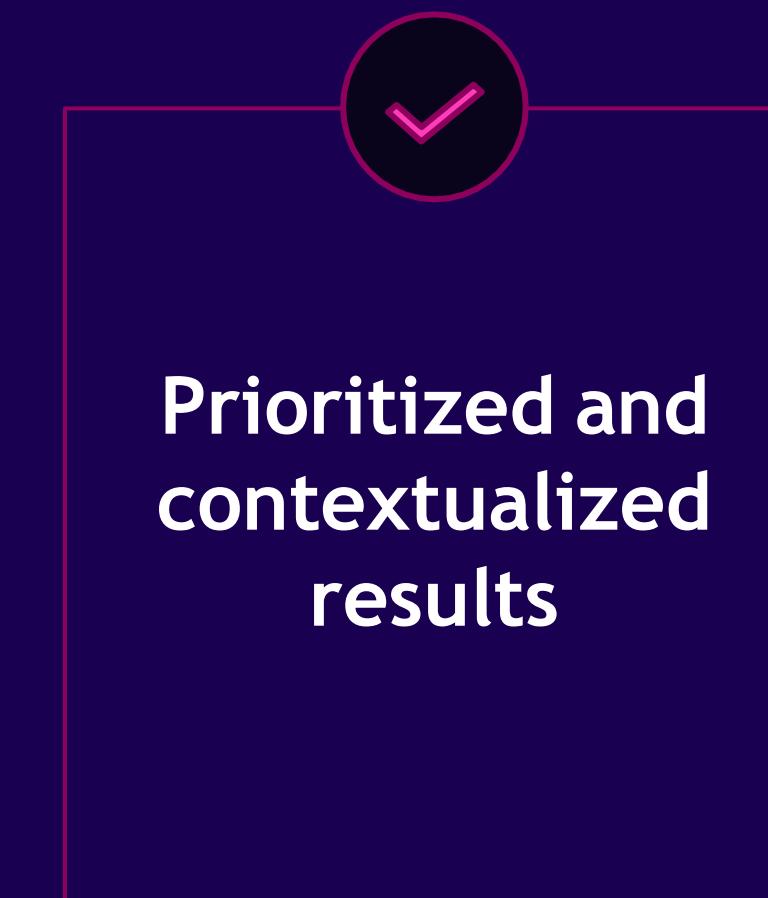
- **Amazon Inspector** is an automated vulnerability management service that **continually** scans AWS workloads for software vulnerabilities and unintended network exposure



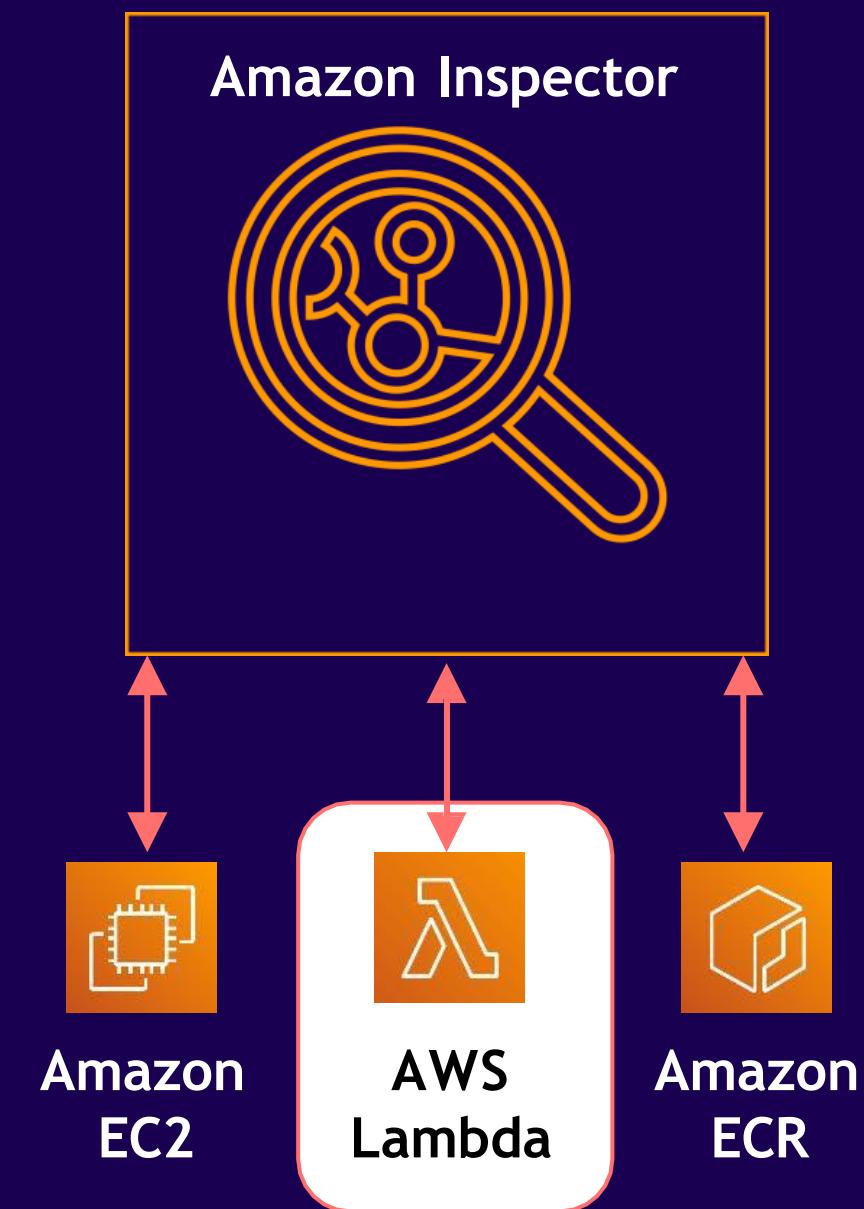
Frictionless
activation and
management



Continuous
vulnerability
monitoring



Prioritized and
contextualized
results



Amazon
EC2

AWS
Lambda

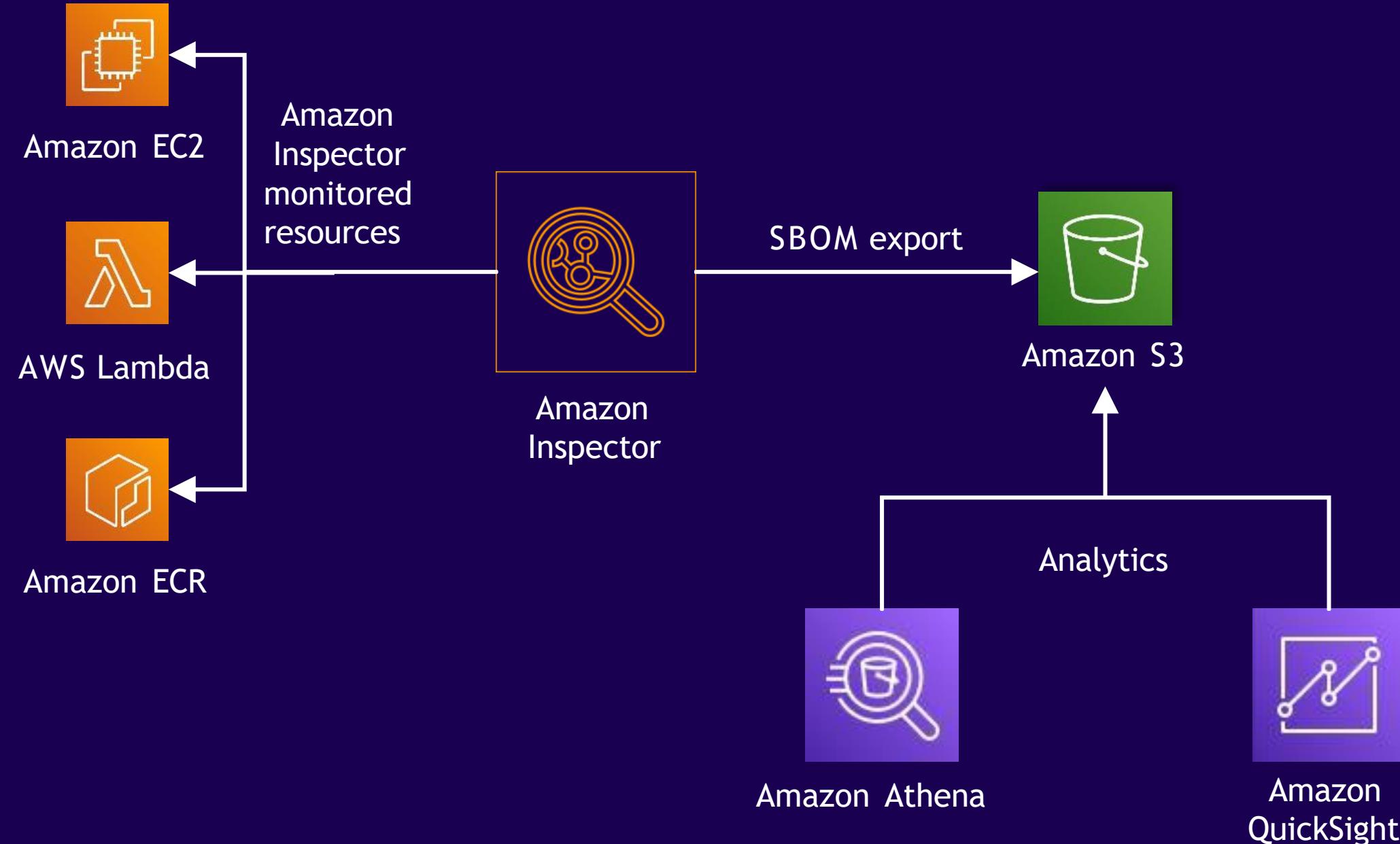
Amazon
ECR



Operate

Software bill of materials

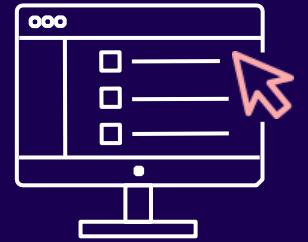
A **software bill of material (SBOM)** is defined as a “nested inventory, a list of ingredients” that make up software components



- ✓ SBOM formats supported: CycloneDX and SPDX
- ✓ SBOMs can be exported for the complete org or as granular as a resource
- ✓ Allows export for all resources being actively monitored by Amazon Inspector
- ✓ Includes both operating system (OS) packages and third-party programming language packages
- ✓ Free of cost

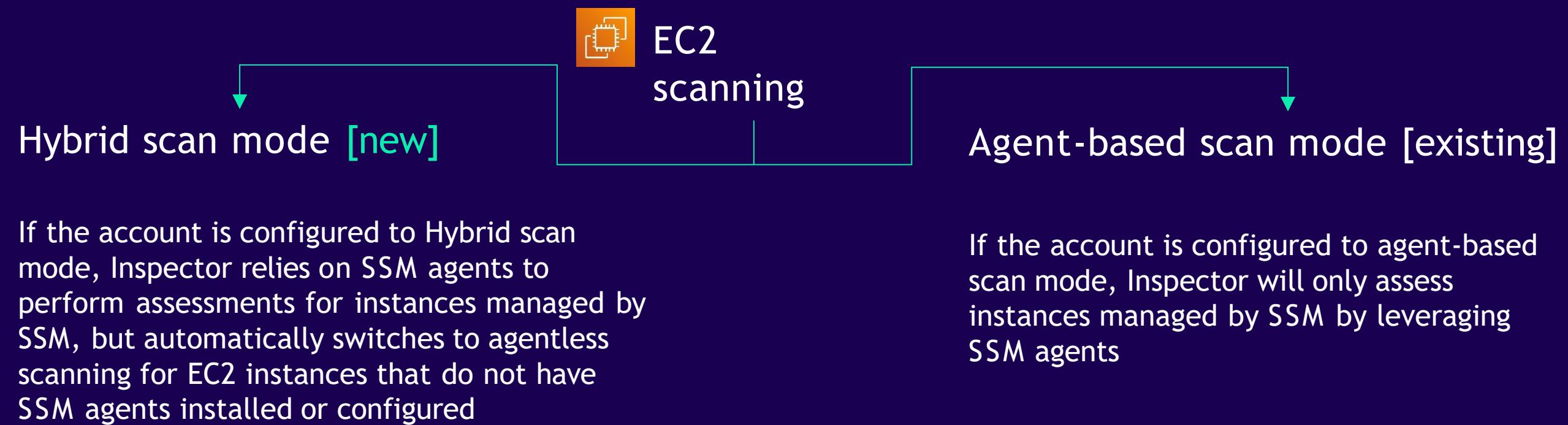
Amazon Inspector for Amazon EC2

Continuously monitor your EC2 instances for software vulnerabilities (CVEs) without installing an agent or additional software



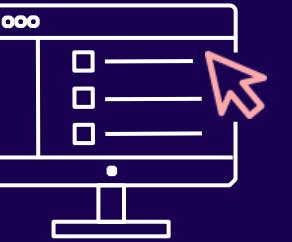
Operate

Agent vs. agentless?



- For agentless scans, Inspector snapshots EBS volumes to access filesystem data using EBS APIs, but snapshots are never copied outside of your account!

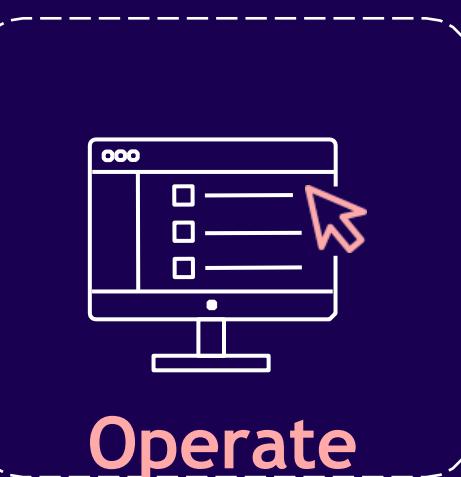
Amazon Inspector for Amazon ECR



Operate

Basic scanning	Enhanced scanning
Open-source Clair or AWS-native	Amazon Inspector integration
OS-level scan	OS and programming language scan
Two frequencies: manual and scan on push	Two frequencies: scan on push and continuous

Amazon Inspector for AWS Lambda



Standard scanning	Code scanning
Detects package vulnerabilities	Detects application code vulnerabilities
Example: Packages such as python-jwt	Example: Injection flaws, unintended data disclosure, weak cryptography
Activate standalone	Activate with Lambda standard scanning
Initial scan + invocation/ update + new CVE	Initial scan + invocation/ update + new code detector
Suggestions for upgrading package	Generative AI-powered code patches

Amazon Macie & Amazon GuardDuty

Key capabilities for Amazon S3

What is Amazon Macie?



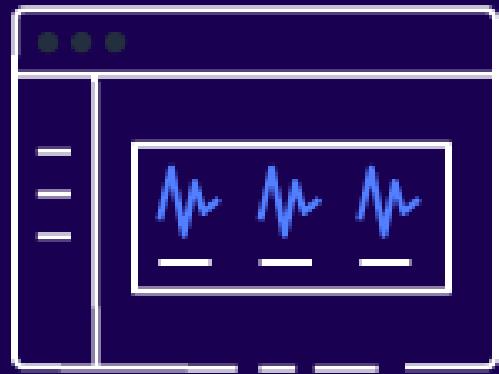
Amazon Macie is a fully managed data security and data privacy service that uses machine learning and pattern matching to discover and protect your sensitive data in AWS.

- ➡ Discover your **sensitive data** at scale
- ➡ Understand what is accessible (public, shared, unencrypted).

Amazon Macie



Discover and protect your sensitive data at scale



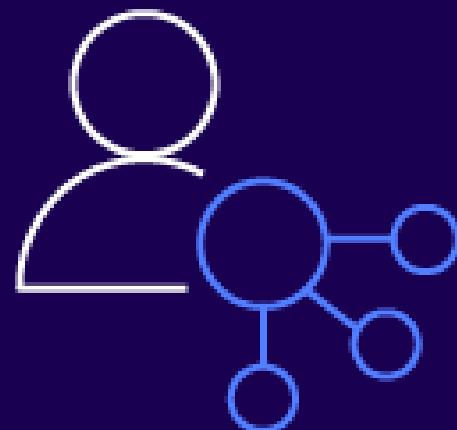
Gain visibility
and evaluate

- Bucket inventory
- Bucket policies



Discover
sensitive data

- Inspection jobs
- Flexible scope



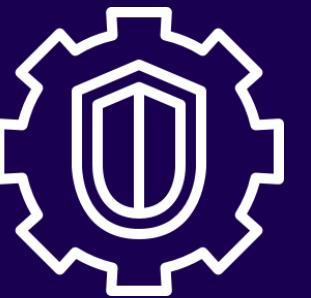
Centrally manage
at scale

- AWS Organizations
- Managed & custom data detections



Automate and
take actions

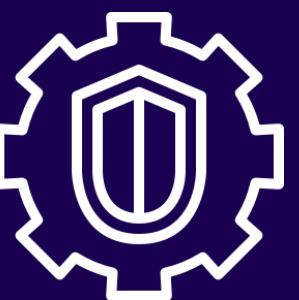
- Detailed findings
- Management APIs



What is Amazon GuardDuty?

Amazon GuardDuty is a threat detection service that uses machine learning, anomaly detection, and integrated threat intelligence to identify and prioritize potential threats.

- Identify malicious & highly suspicious activity
- Protects AWS accounts, workloads, and data stored in S3.



How Amazon GuardDuty works?

Amazon GuardDuty

Data Sources

VPC flow logs	
DNS Logs	
CloudTrail Events	
S3 Data Plane Events	

Threat Detection Types

Threat intelligence

Bitcoin Mining
C&C Activity

Anomaly Detection (ML)

Unusual User behavior
Example:

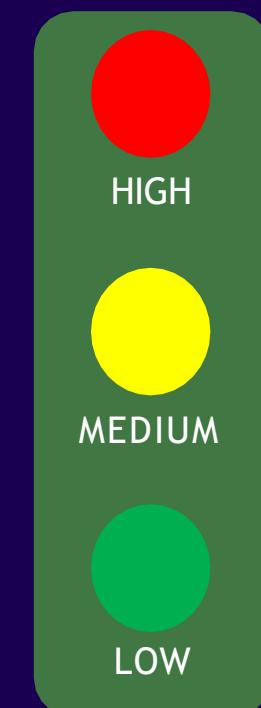
- Launch instance
- Change Network Permissions

Unusual traffic patterns
Example:

- Unusual ports and volume

Finding Types Examples

Findings



Amazon Detective

AWS Security Hub

CloudWatch Event

- Alert
- Remediate
- Partner Solutions
- Send to SIEM



**How can Amazon Macie & Amazon GuardDuty help
protect data in Amazon S3**

Easily enable the services on all accounts

Centrally manage at scale

Administrator/Member setup



- Designate a delegated administrator

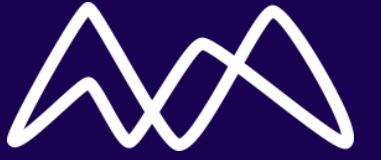


- Add all member accounts



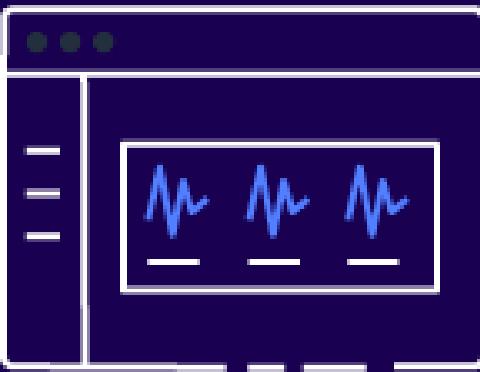
- Auto-enable Macie on all accounts
- Auto-enable GuardDuty on all accounts

Evaluate bucket security posture



Gain visibility and evaluate

- Gain visibility into S3 bucket inventory
 - Number of buckets
 - Storage size
 - Object count
- Monitor changes to S3 bucket policies
 - Publicly accessible buckets
 - Encrypted vs Unencrypted
 - Shared outside of the account
 - Replicated to external accounts



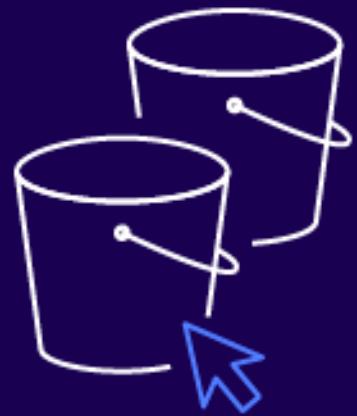
- Automatically and continuously updated for all accounts, with alerts for policy findings.



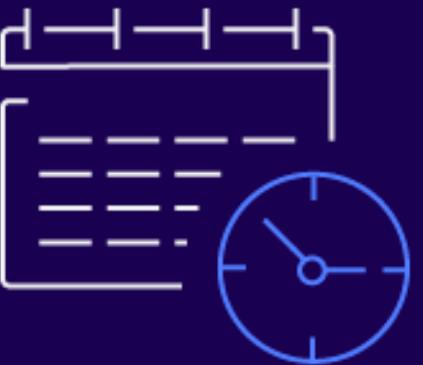
Run inspection jobs on data sets

Discover sensitive data

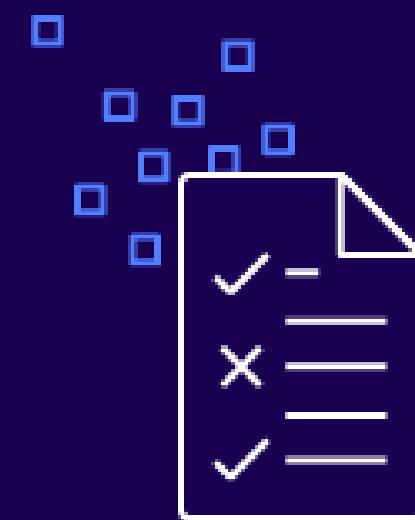
- Ongoing evaluation of your Amazon S3 environment and data



- Select target for data discovery
- Create and schedule jobs

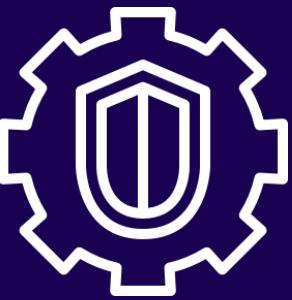


- Define the scope
- Scheduled frequency (one-time, daily, weekly, monthly)
- Object criteria (Tags, modified time, extension type, size)



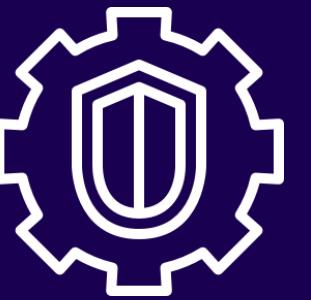
- Review classification findings
- Managed Data identifiers
 1. *Financial (card, bank account numbers...)*
 2. *Personal (names, address, contact...)*
 3. *National (passport, ID, driver license...)*
 4. *Medical (healthcare, drug agency ...)*
 5. *Credentials & secrets (AWS secret keys, private keys ...)*
- Custom Data identifiers (*RegEx*)

Amazon GuardDuty for S3 Protection



With a **single** click Amazon GuardDuty-for-S3 will continuously monitor and profile S3 data access events and S3 configurations across **your entire S3 estate** (all accounts and buckets).

It will **detect malicious and suspicious activities** such as requests coming from an unusual geo-location, disabling of preventive controls such as S3 block public access, or API calls consistent with attempts of data exfiltration.



Detect malicious & anomaly behavior

Proactively detect threats to data in Amazon S3

Policy

- Bucket made public
- Block public access disabled
- Logging disabled
- Root credentials used

Malicious access

- Data discovery, exfiltration, or modification from:
 - Tor
 - Leaked instance credentials
 - Malicious IPs

Anomalous behavior

- Unusual location
- Unusual bucket
- Unusual volume
- Unusual error volume

- Automatically and continuously updated for all buckets in an account.

Example Use Cases

Continuous inventory evaluation and data governance

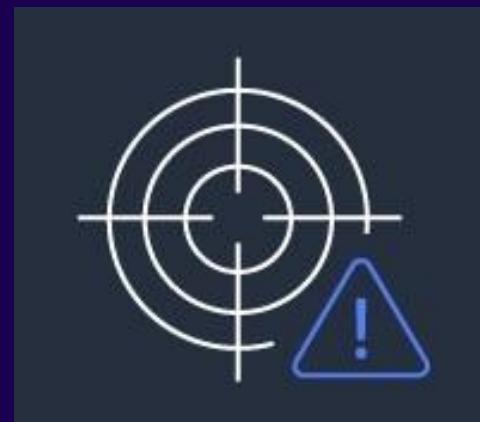
Maintain regulatory compliance



Compliance teams are required to **monitor** where **sensitive data** resides, protect it properly, and provide evidence that they are enforcing **data security** and privacy to meet **regulatory compliance** requirements.

Threat detection and monitoring

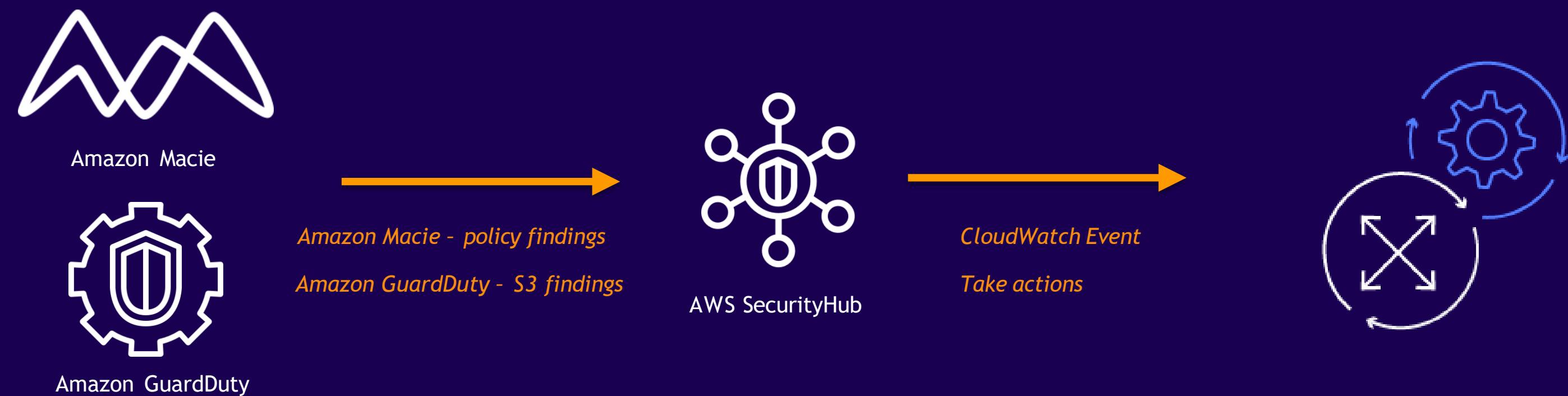
Automate threat response and remediation



Continuously monitor and profile S3 data access events (data plane operations) and S3 configurations (control plane APIs) to detect suspicious activities such as requests coming from an unusual geo-location, disabling of preventative controls such as S3 block public access, or API call patterns consistent with an attempt to discover misconfigured bucket permissions.

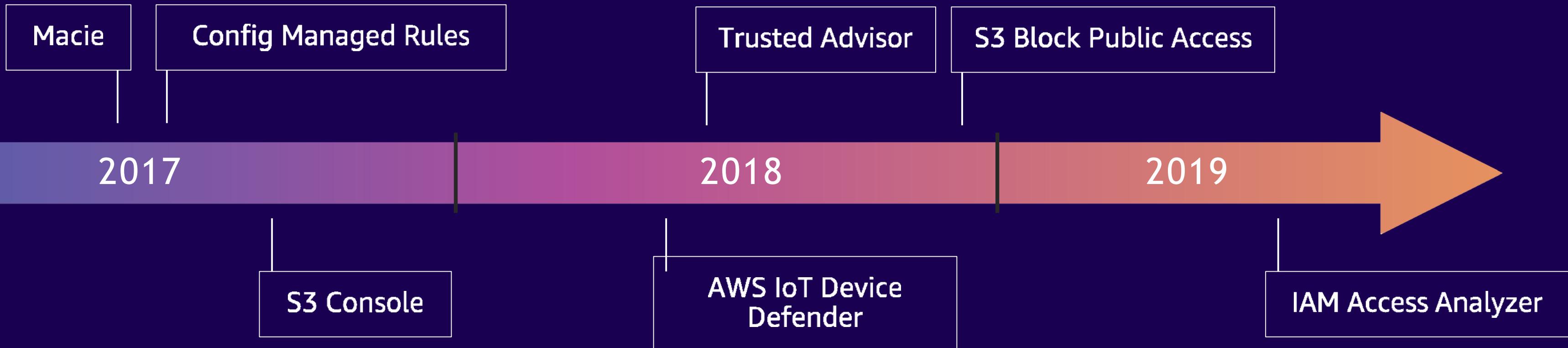
Automated response and remediation

Identify, protect, and continuously monitor for risks



Security Hub collects findings from the security services enabled across your AWS accounts, such as S3 findings from Amazon GuardDuty, and policy findings from Amazon Macie. Security Hub aggregates findings into pre-built dashboards that show you the current security status of your environment as well as trends so customers can easily identify potential issues, and take the necessary next steps. For example, you can send findings to ticketing, chat, email, or **automated remediation** systems using integration with Amazon CloudWatch Events.

Mathcing the cloud



Beyond Boolean

ZELKOVA

"Is my bucket
public?"



ACCESS ANALYZER

"Who has access
to what?"



AVAILABLE NOW



IAM Access Analyzer

ACCESS

A new IAM capability that generates comprehensive findings if your resource policies grant **public** or **cross-account** access

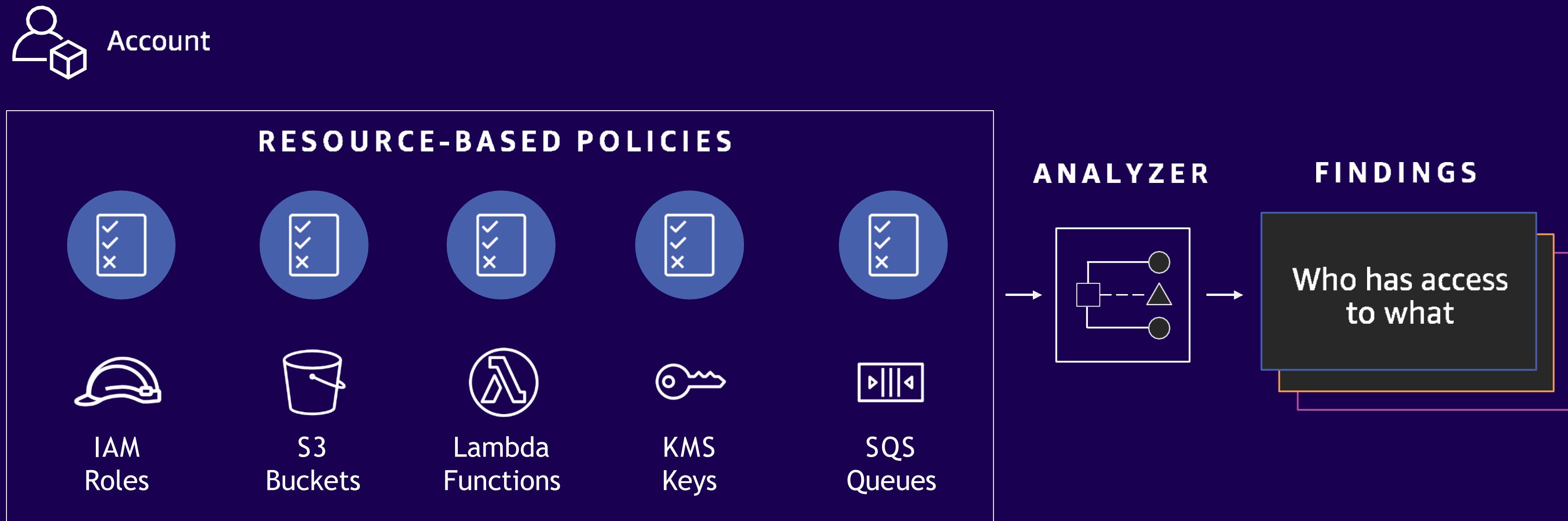
IDENTIFICATION

You can **quickly identify** resources with overly broad permissions without requiring deep knowledge of your policies

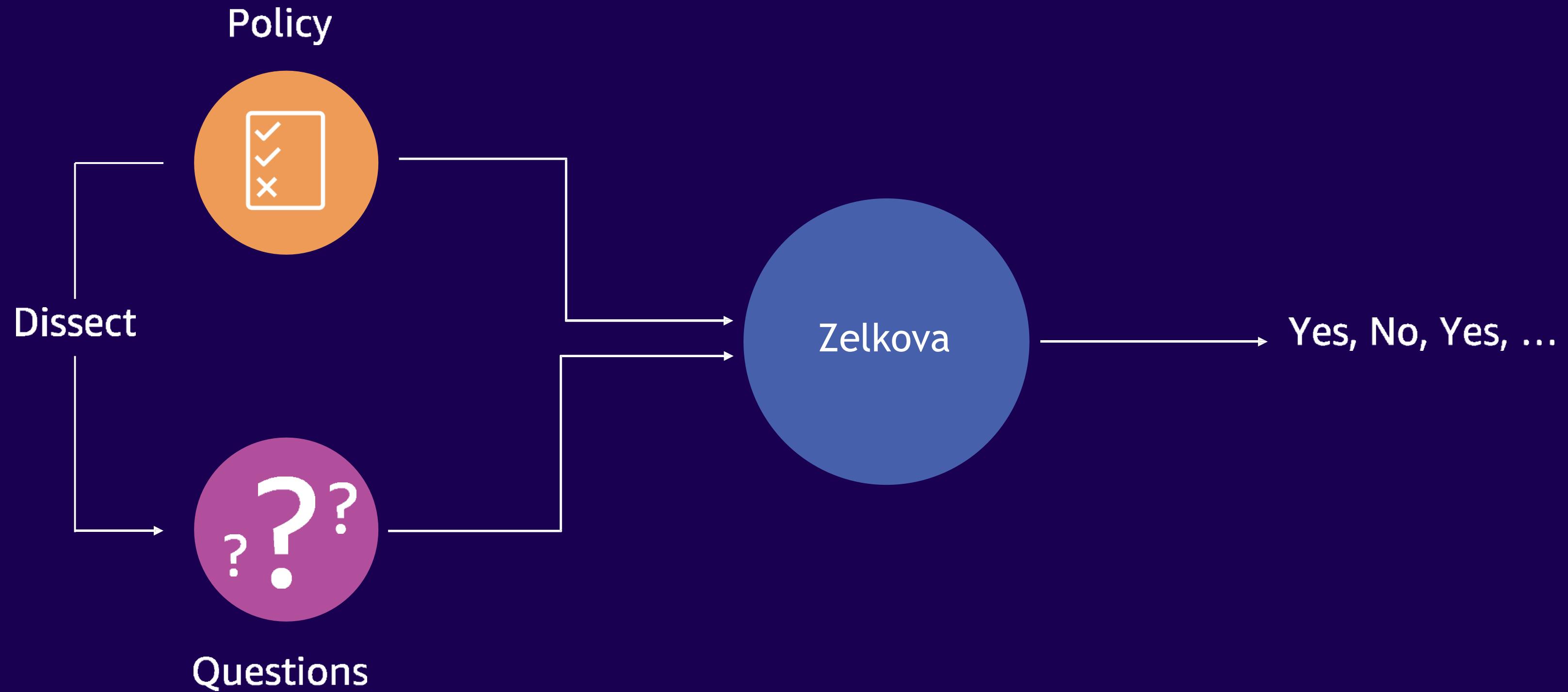
PROTECTION

Archive findings for intended access or resolve findings by updating policies to **protect your resources** from unintended access before it occurs

How does it work?



Under the hood



IAM Access Analyzer is comprehensive



TURN IT ON TODAY!



IAM Access Analyzer

Go to the IAM console and turn on IAM Access Analyzer with one click. There are APIs you can use to get findings.

Available at no charge!

AWS Encryption SDK

Released in JavaScript

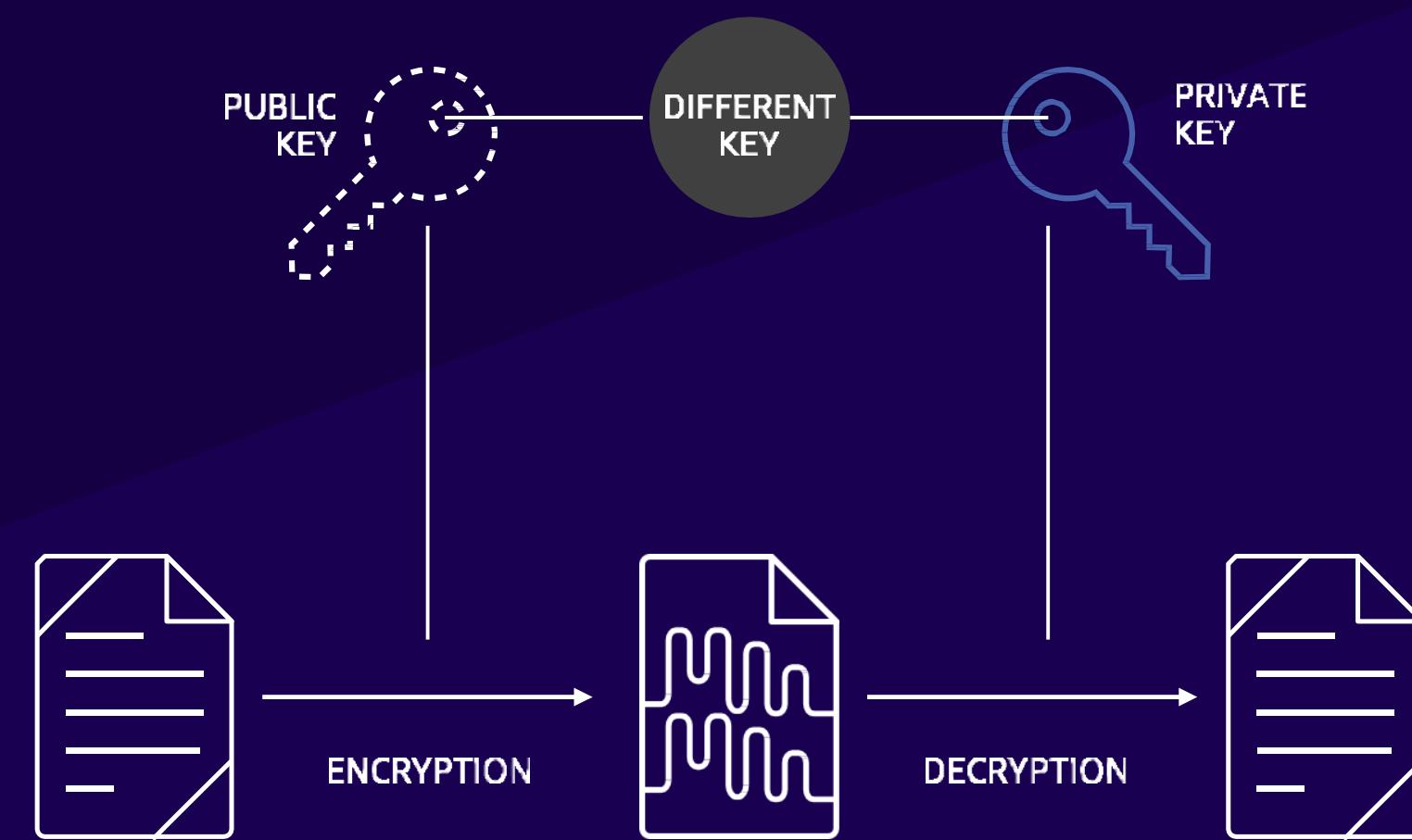
AVAILABLE NOW!



AWS KMS

AWS Key Management Service (KMS) makes it easy for you to create and manage cryptographic keys and control their use across a wide range of AWS services and in your applications.

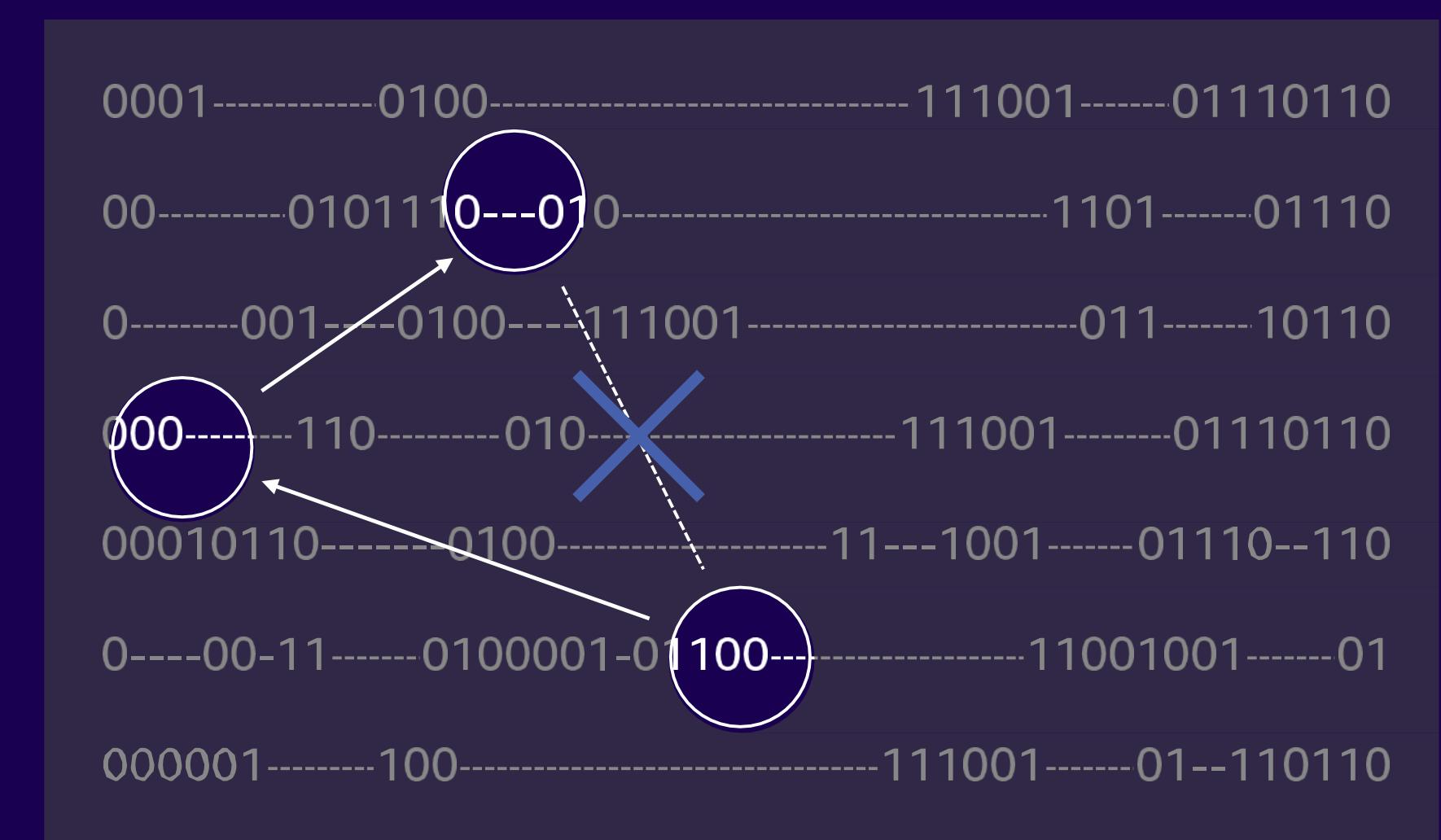
NOW WITH ASYMMETRIC KEY SUPPORT



NEW !

Short Cut Resistant Authenticated Mode (SCRAM)

Cryptographically prevent implementers
from taking short cuts such as looking at
decrypted data before it's been verified

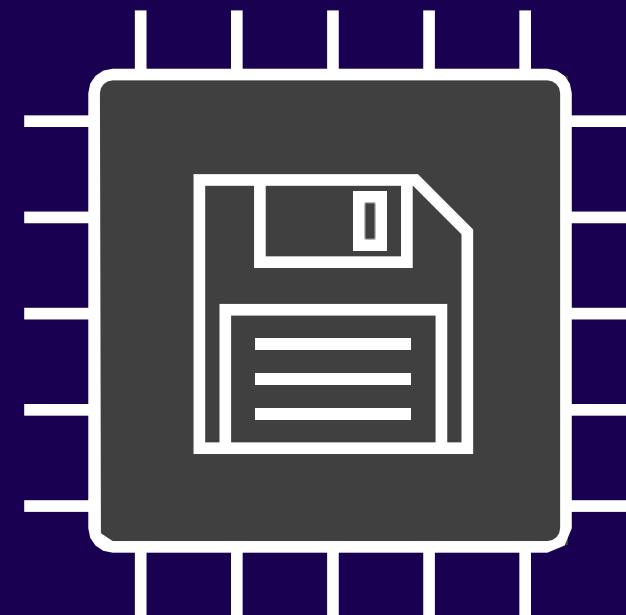
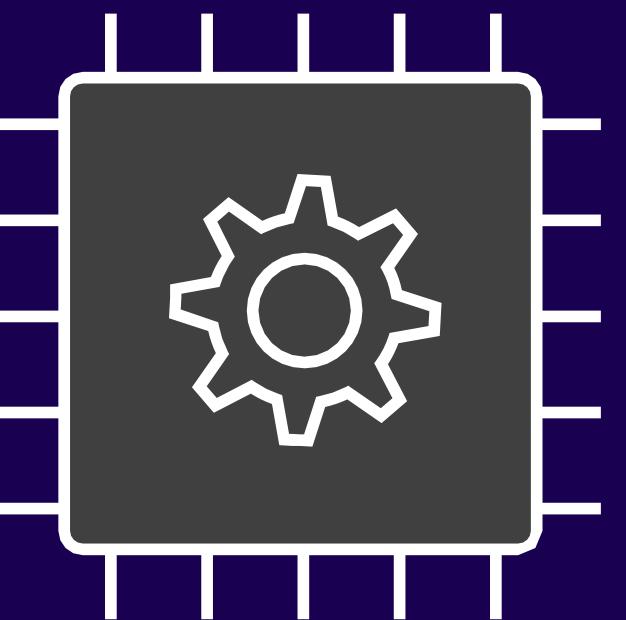


OPEN SOURCED TODAY!

NEW !

Amazon EC2 ARM Instances

Powered by the AWS Graviton2
Processor. Up to 45% lower cost for
scale-out workloads.

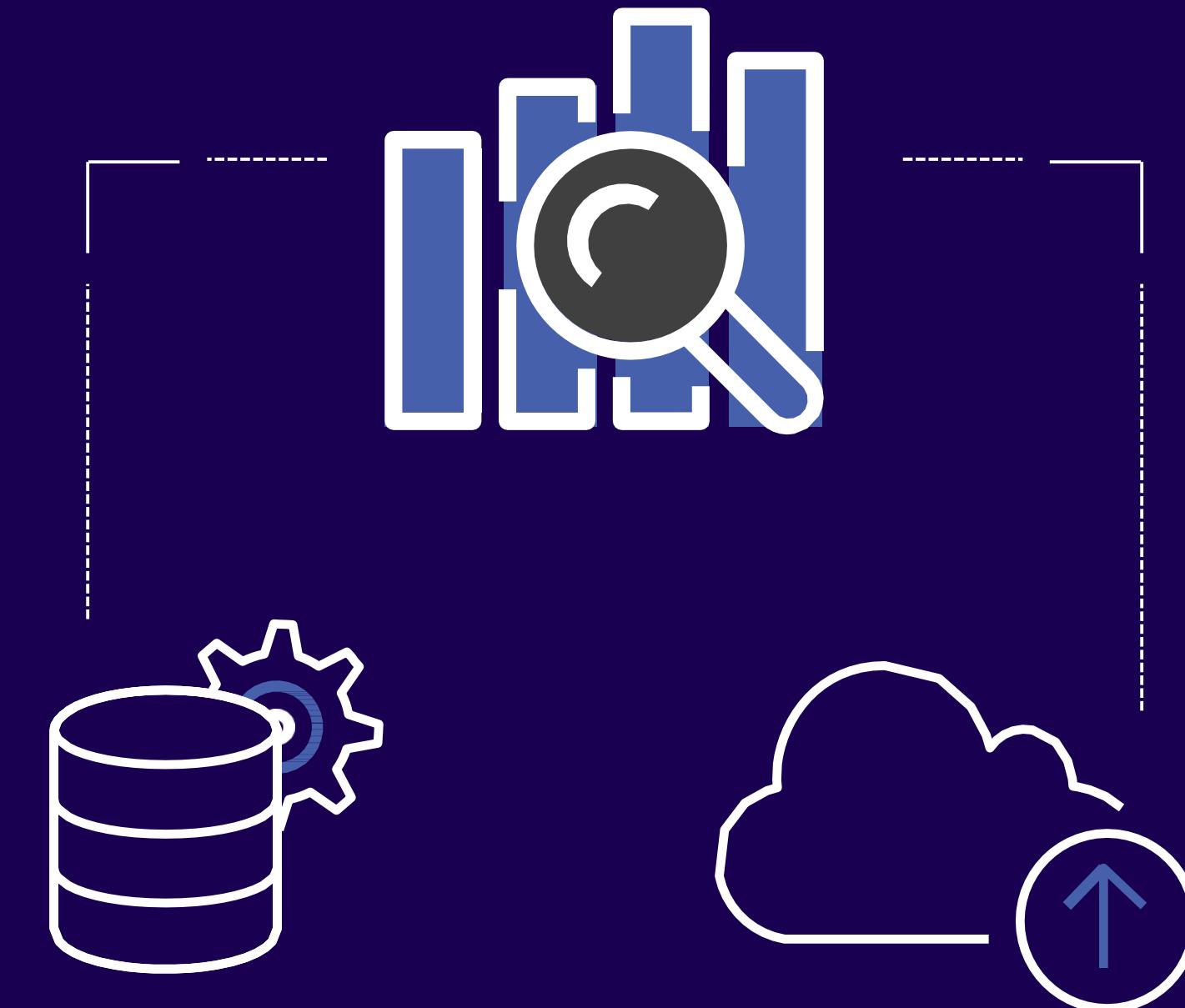


NEW !

UltraWarm

Provides customers decouple compute and storage from Amazon Elasticsearch Service while still providing the interactive analytics experience of Kibana.

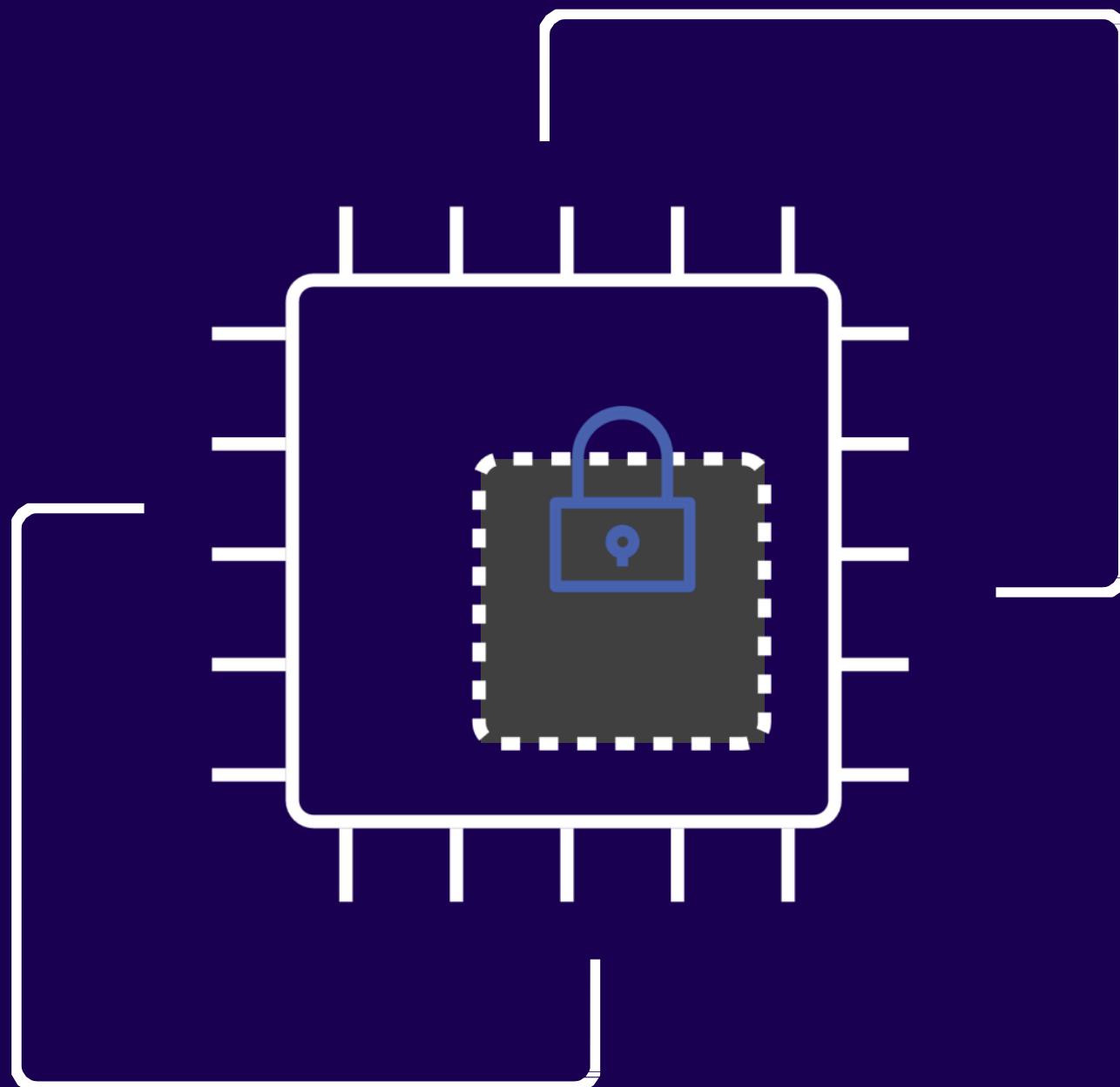
GENERALLY AVAILABLE TODAY



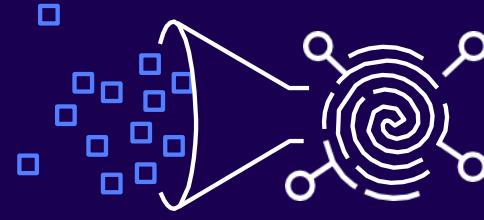
NEW !

Nitro Enclaves

Provides customers a mechanism to
create isolated compute environments
within Amazon EC2.



Investigation Challenges



Signal-to-noise ratio



Complexity



Skills shortage



Costs

INTRODUCING



Amazon Detective

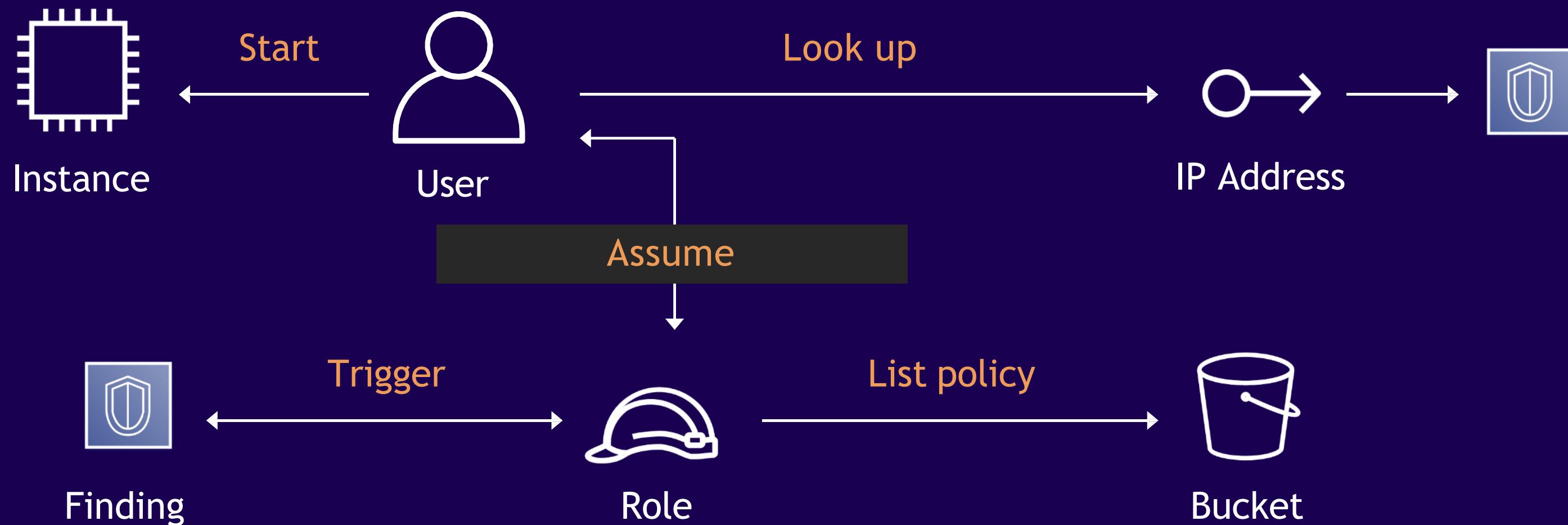
Quickly analyze, investigate, and identify the root cause of security issues

Built-in data collection

Automated analysis

Visual insights

Amazon Detective security behavior graph

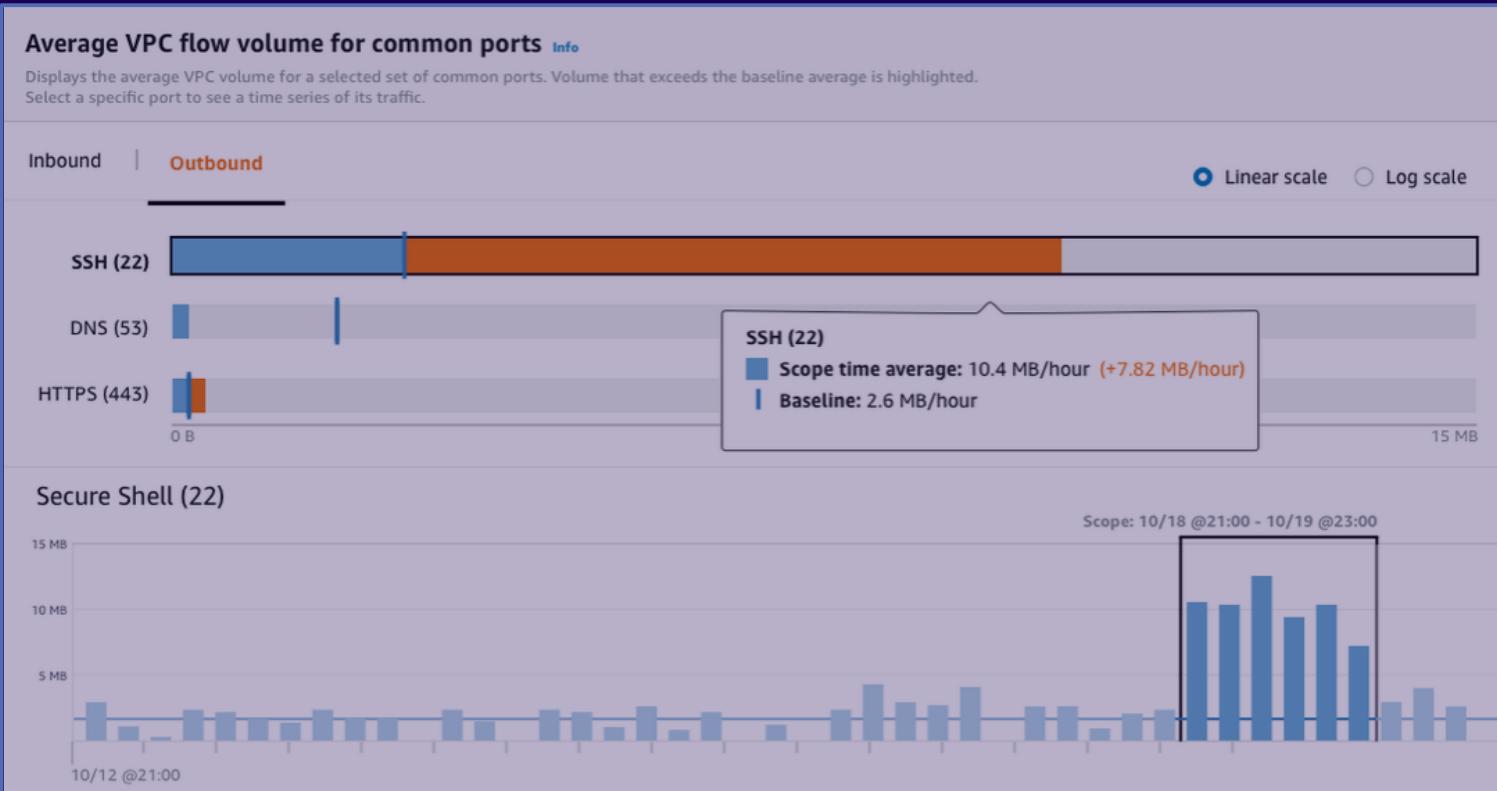


Answer your investigation questions



"What happened just before?"

"Are these call failures common?"



"How much data was sent?"

"Is this traffic normal?"

Detective - integrations and managed services

TECHNOLOGY PARTNERS



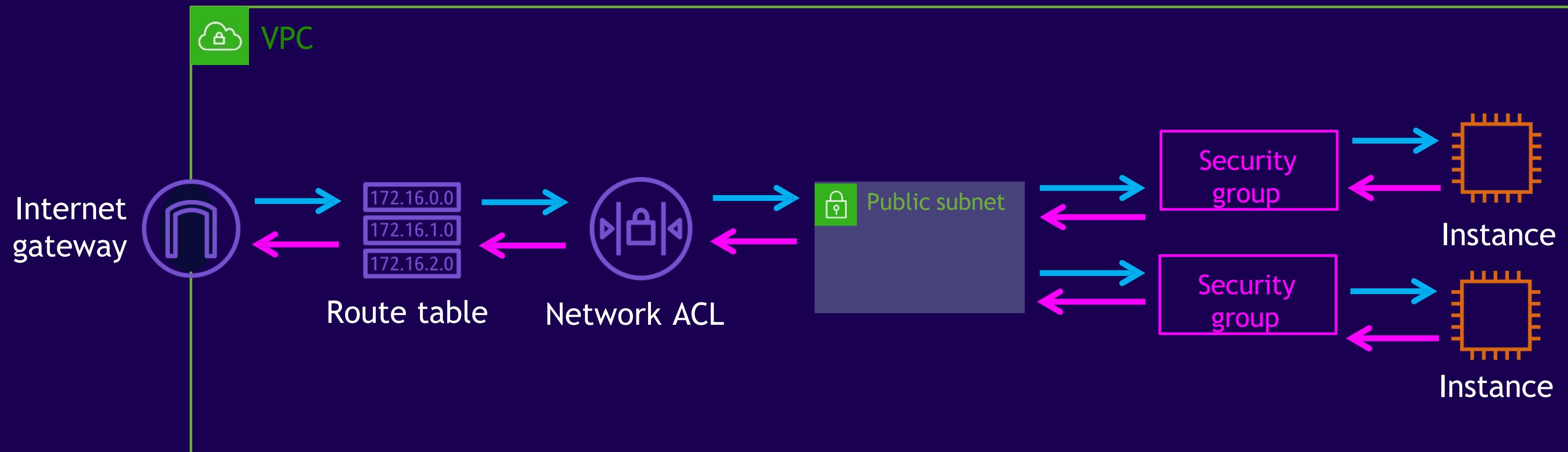
SERVICE PARTNERS



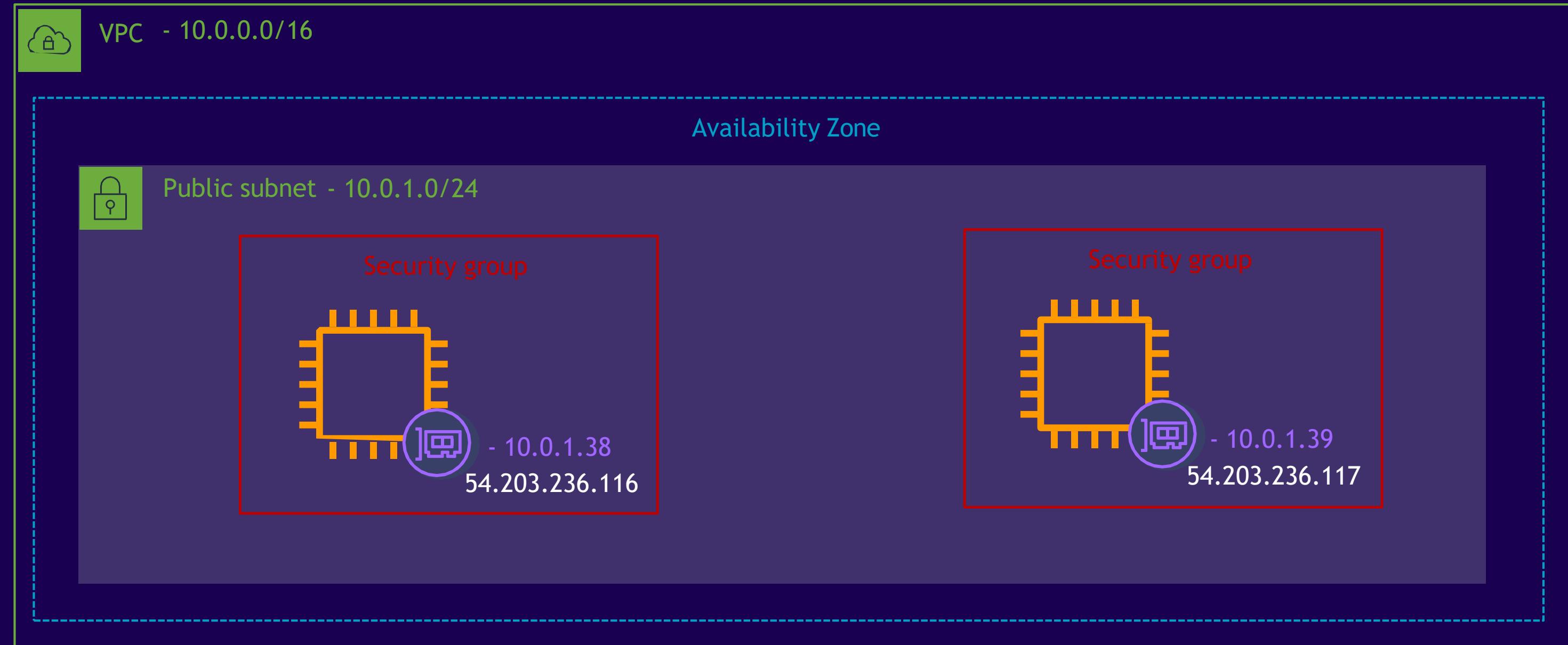
Basics of VPC security



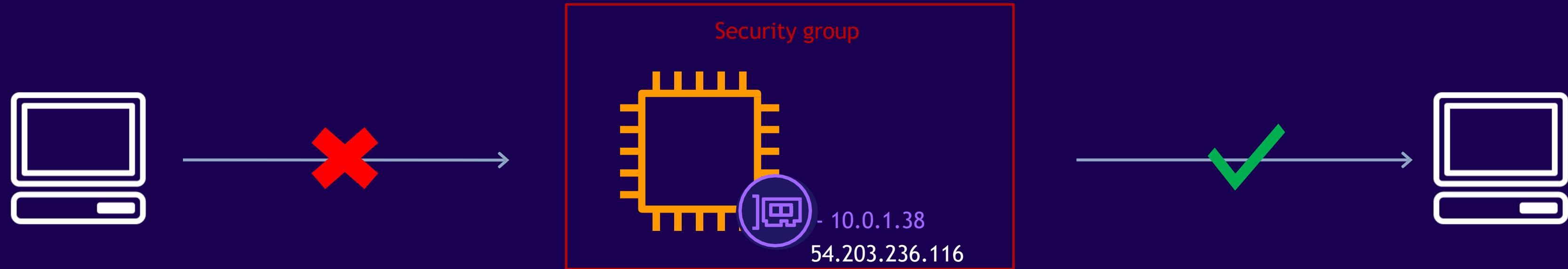
VPC defense in depth



Security groups



Security groups - default behavior

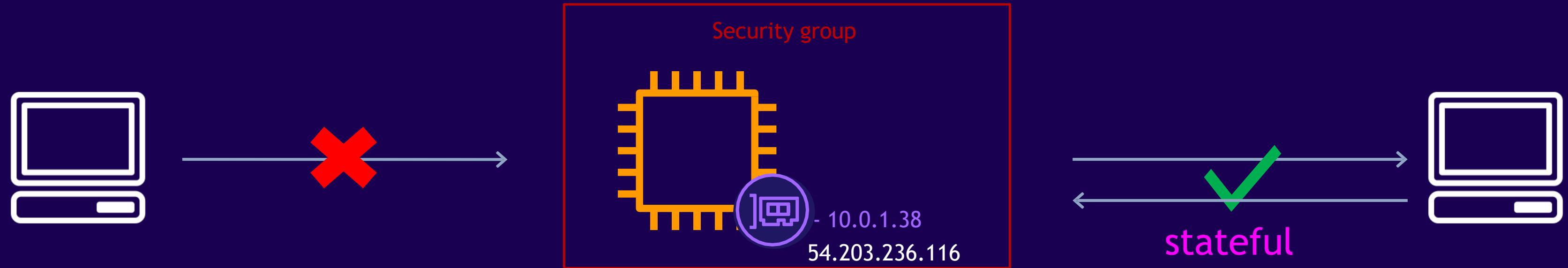


IP version	Type	Protocol	Port range	Source
No security group rules found				

IP version	Type	Protocol	Port range	Destination
IPv4	All traffic	All	All	0.0.0.0/0
IPv6	All traffic	TCP	All	::/0

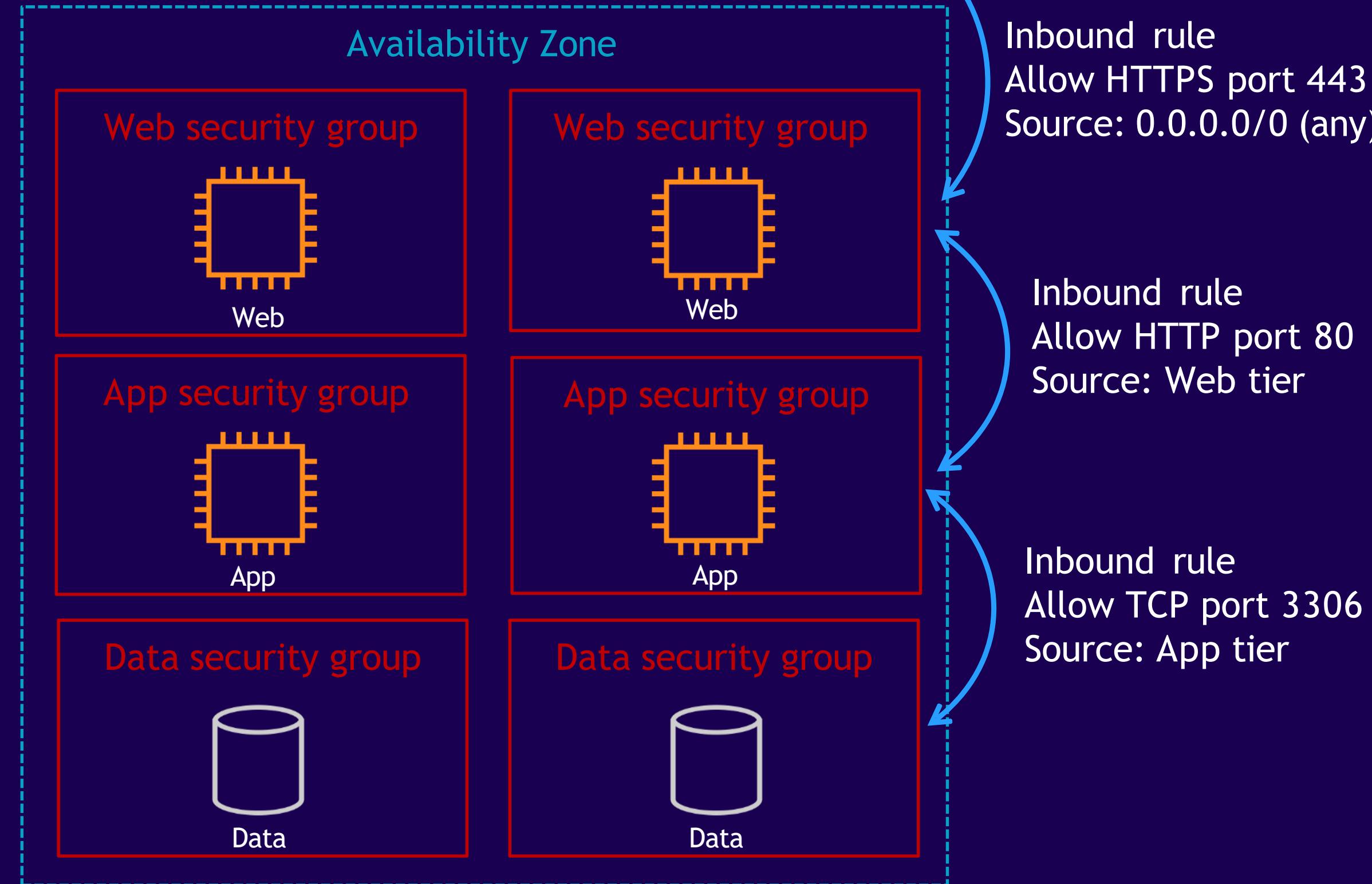
Outbound

Security groups - default behavior

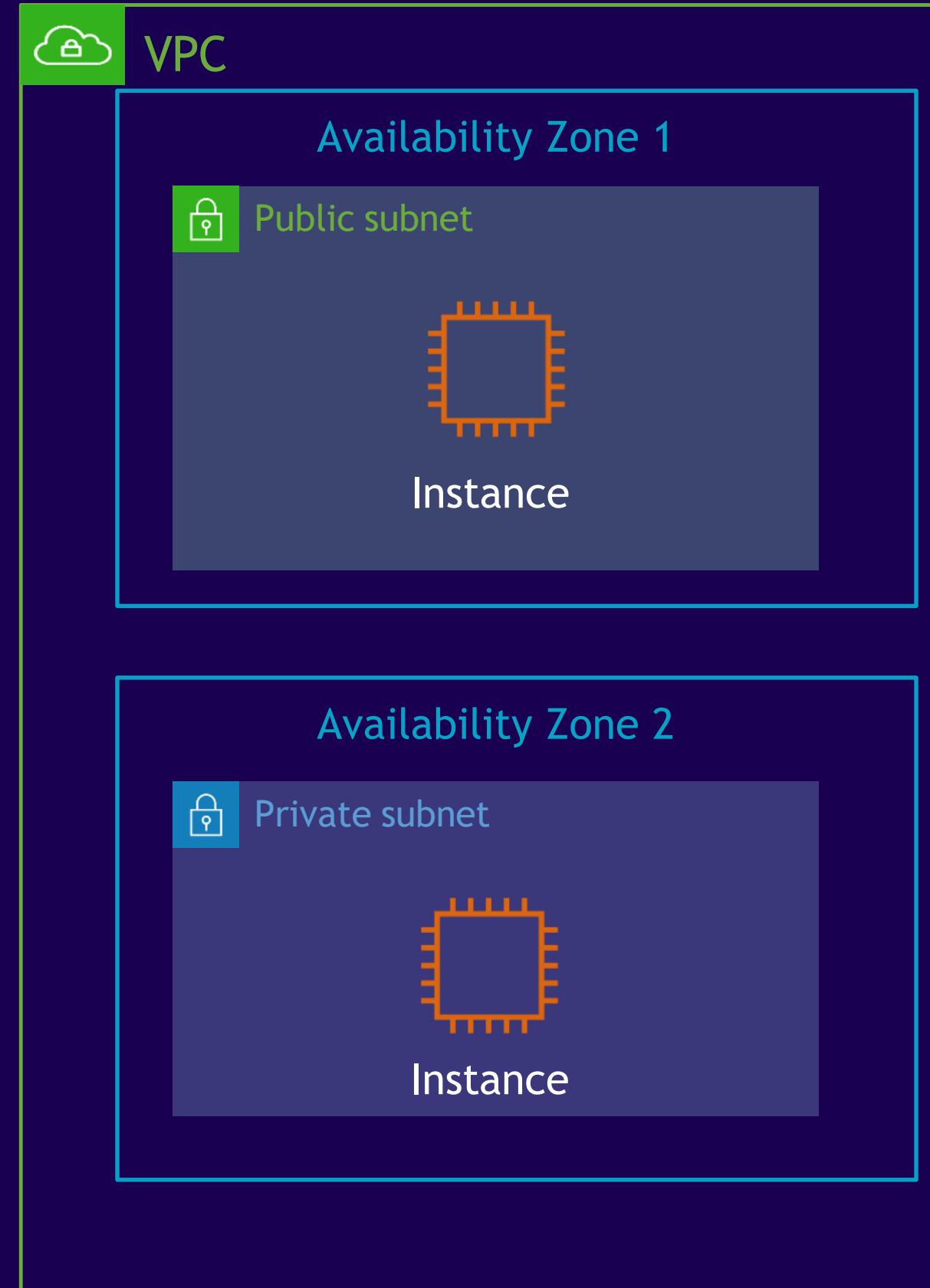


Inbound	IP version	Type	Protocol	Port range	Source
No security group rules found					
Outbound					
IP version	Type	Protocol	Port range	Destination	
IPv4	All traffic	All	All	0.0.0.0/0	
IPv6	All traffic	TCP	All	::/0	

Security Group Chaining



Network access control lists (NACLs)



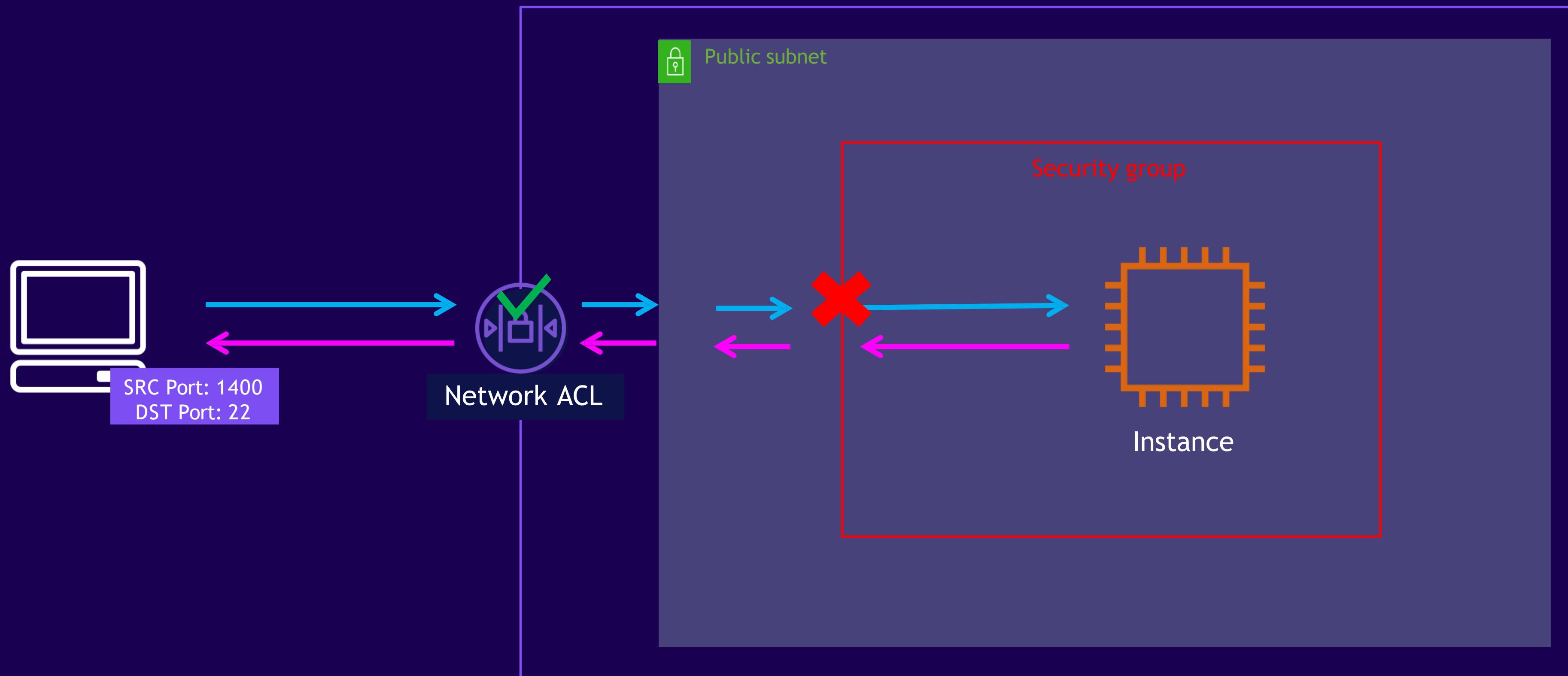
Inbound rules - default

Rule number	Type	Protocol	Port range	Source	Allow/Deny
100	All traffic	All	All	0.0.0.0/0	<input checked="" type="checkbox"/> Allow
101	All traffic	All	All	::/0	<input checked="" type="checkbox"/> Allow
*	All traffic	All	All	0.0.0.0/0	<input checked="" type="checkbox"/> Deny
*	All traffic	All	All	::/0	<input checked="" type="checkbox"/> Deny

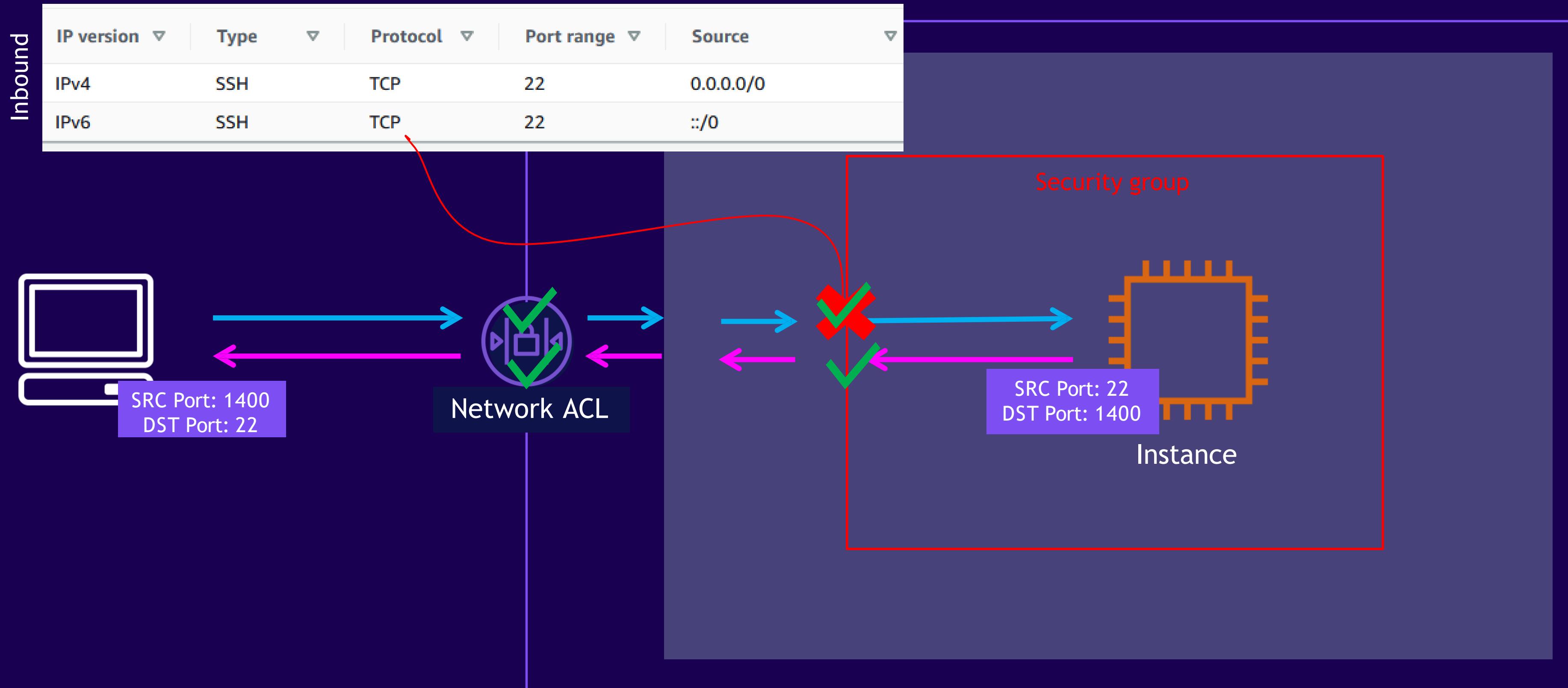
Outbound rules - default

Rule number	Type	Protocol	Port range	Destination	Allow/Deny
100	All traffic	All	All	0.0.0.0/0	<input checked="" type="checkbox"/> Allow
101	All traffic	All	All	::/0	<input checked="" type="checkbox"/> Allow
*	All traffic	All	All	0.0.0.0/0	<input checked="" type="checkbox"/> Deny
*	All traffic	All	All	::/0	<input checked="" type="checkbox"/> Deny

Additional configurations for inbound traffic



Additional configurations for inbound traffic



Securing EC2 & RDS Instances

Keep OS and apps patched at all times

Do not allow unrestricted access to the EC2 instances

Standard OS practices for hardening OS - remove unwanted services

Consider using hardened managed containers

Only open Ports that are required

Design and place instances in private subnets (esp. RDS instances)

Securing EC2 & RDS Instances

Create individual users for managing RDS resources

Only min. access should be granted

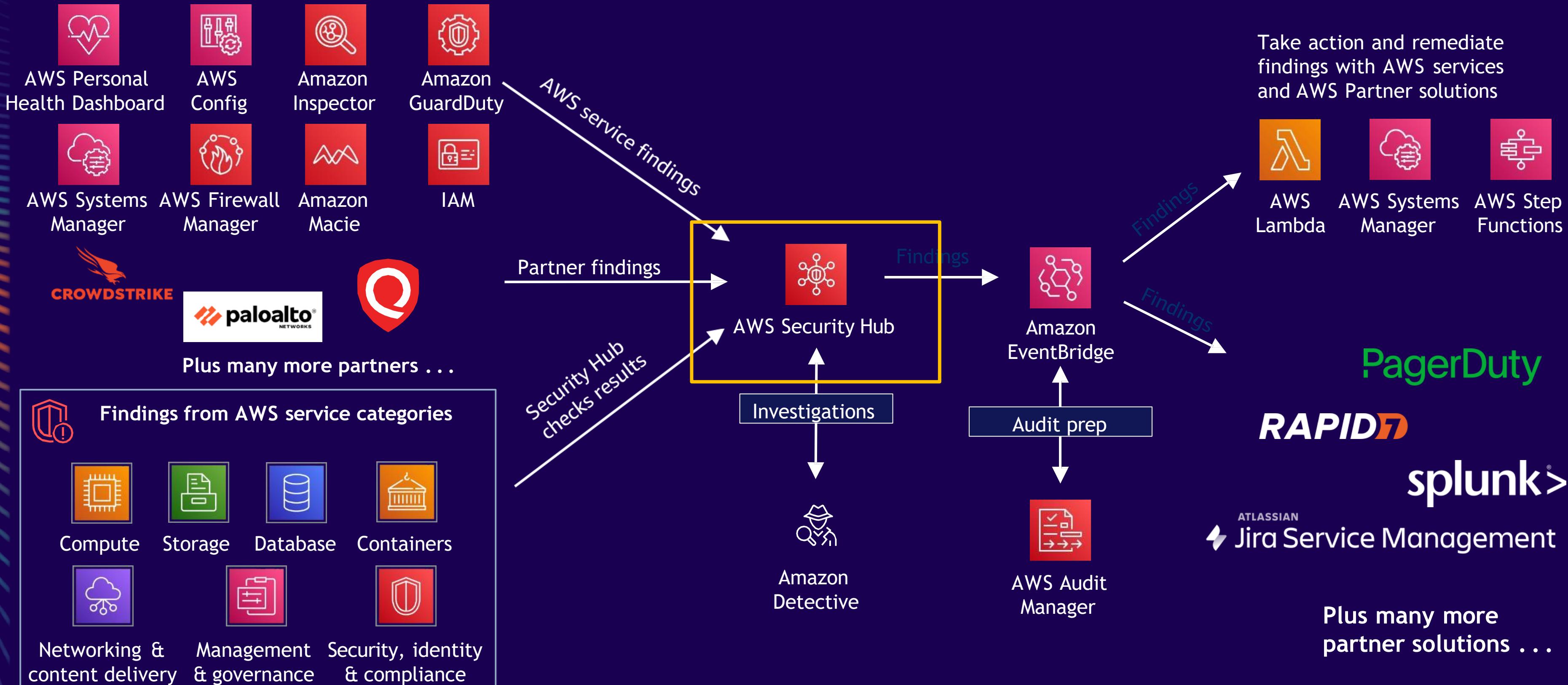
Applications - Keep Read and Write credentials separate

Applications - Do not keep credentials in code

Rotate IAM credentials regularly (hint: AWS Secrets Manager)

Assign Groups and Roles appropriately

Security flows with Security Hub



This is not a complete list. To view all AWS Partners for this category, visit AWS Partner Solutions Finder. This list of partners is current as of May 26, 2023.



Container security and compliance checks using AWS Security Hub

AWS Foundational Security Best Practices v1.0.0 by AWS

Description
The AWS Foundational Security Best Practices standard is a set of automated security checks that detect when AWS accounts and deployed resources do not align with security best practices. The standard is defined by AWS security experts. This curated set of controls helps improve your security posture in AWS, and covers AWS's most popular and foundational services.

Security score

64%
Updated 6 hours ago

[Disable](#) [View results](#)

CIS AWS Foundations Benchmark v1.2.0 by AWS

Description
The Center for Internet Security (CIS) AWS Foundations Benchmark v1.2.0 is a set of security configuration best practices for AWS. This Security Hub standard automatically checks for your compliance readiness against a subset of CIS requirements.

Security score

17%
Updated 6 hours ago

[Disable](#) [View results](#)

PCI DSS v3.2.1 by AWS

Description
The Payment Card Industry Data Security Standard (PCI DSS) v3.2.1 is an information security standard for entities that store, process, and/or transmit cardholder data. This Security Hub standard automatically checks for your compliance readiness against a subset of PCI DSS requirements.

Security score

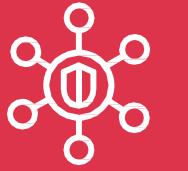
53%
Updated 6 hours ago

[Disable](#) [View results](#)

200+ fully automated, nearly continuous checks evaluated against preconfigured rules

Findings are displayed on main dashboard for quick access

Best practice information is provided to help mitigate gaps and be in compliance



Container security and compliance checks using AWS Security Hub

AWS Foundational Security Best Practices v1.0.0 by AWS

Description
The AWS Foundational Security Best Practices benchmark provides a curated set of security checks that align with security best practices. This curated set covers AWS's most critical services.

Security score: 64% (Updated 6 hours ago)

CIS AWS Foundations Benchmark v1.2.0 by AWS

Description
The CIS AWS Foundations Benchmark provides a curated set of security controls that align with the CIS AWS Foundations Benchmark. This curated set covers AWS's most critical services.

Security score: 53% (Updated 6 hours ago)

Findings (3)

A finding is a security issue or a failed security check.

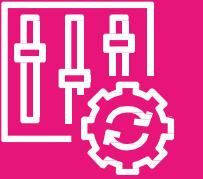
Add filter

Title starts with ECS X Workflow status is NEW X Workflow status is NOTIFIED X Record state is ACTIVE X Clear filters

Severity	Workflow status	Record State	Region	Account Id	Company	Product	Title	Resource	Compliance Status	Updated at
HIGH	NEW	ACTIVE	us-east-1	[REDACTED]	AWS	Security Hub	ECS services should not have public IP addresses assigned to them automatically	AwsEcsService InspectorBlogService	FAILED	13 hours ago
HIGH	NEW	ACTIVE	us-east-1	[REDACTED]	AWS	Security Hub	ECS containers should be limited to read-only access to root filesystems	AwsEcsTaskDefinition 1	FAILED	13 hours ago
MEDIUM	NEW	ACTIVE	us-east-1	[REDACTED]	AWS	Security Hub	ECS clusters should use Container Insights	AwsEcsCluster InspectorBlogCluster	FAILED	13 hours ago

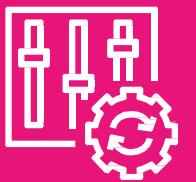
ated, nearly
ks evaluated
ured rules
layed on main
ick access
ormation is
mitigate gaps
ance

Security using AWS Config



- AWS Config keeps track of all changes to your resources
- AWS Config uses Security best practices for EKS: Conformance pack
 - Conformance packs
 - Provide a general-purpose compliance framework
 - Create security, operational, or cost-optimization governance checks
 - Use managed or custom AWS Config rules
 - Perform AWS Config remediation actions

Container configuration history



AWS Config X

Dashboard

Conformance packs

Rules

Resources

▼ Aggregators

Conformance packs

Rules

Resources

Authorizations

Advanced queries

Settings

What's new

Documentation

Events

All times are in America/Chicago (UTC-05:00)

Start date: 2021/09/03 Now Event type: All event types

September 3, 2021

10:59:07	Configuration change	6 field change(s)
10:45:01	Rule compliance 1 Noncompliant rule(s)	2 rule(s) applied
10:44:09	Rule compliance All compliant	1 rule(s) applied
10:43:40	Configuration change JSON diff - 0 field change(s) From: { } To: { }	0 field change(s)

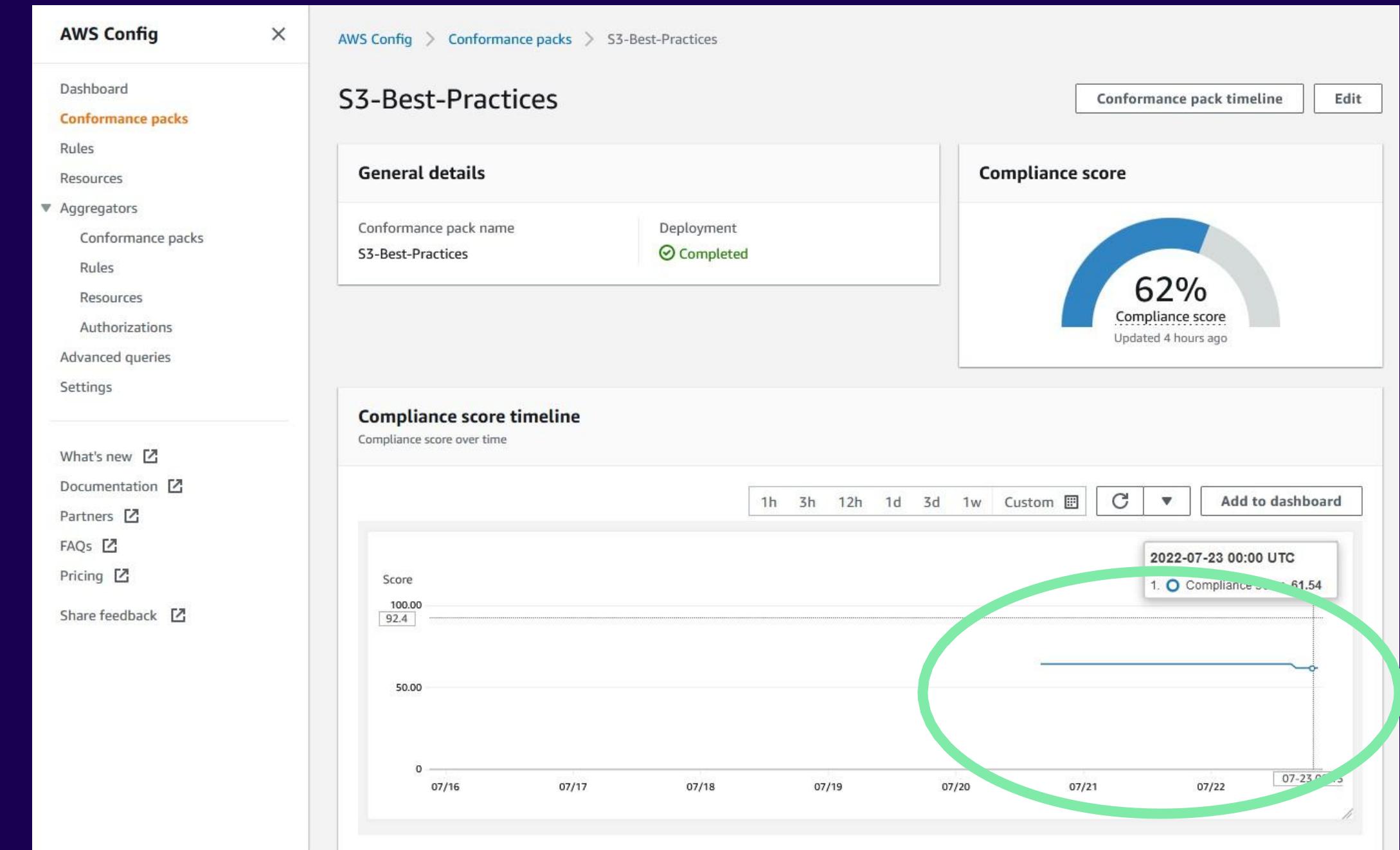
[View full record](#)

Compliance score for conformance packs



Historical view of compliance of conformance packs

Measure impact of changes



Secure IAM Roles

IAM consists of
Users
Groups
Roles
Permissions

AWS IAM

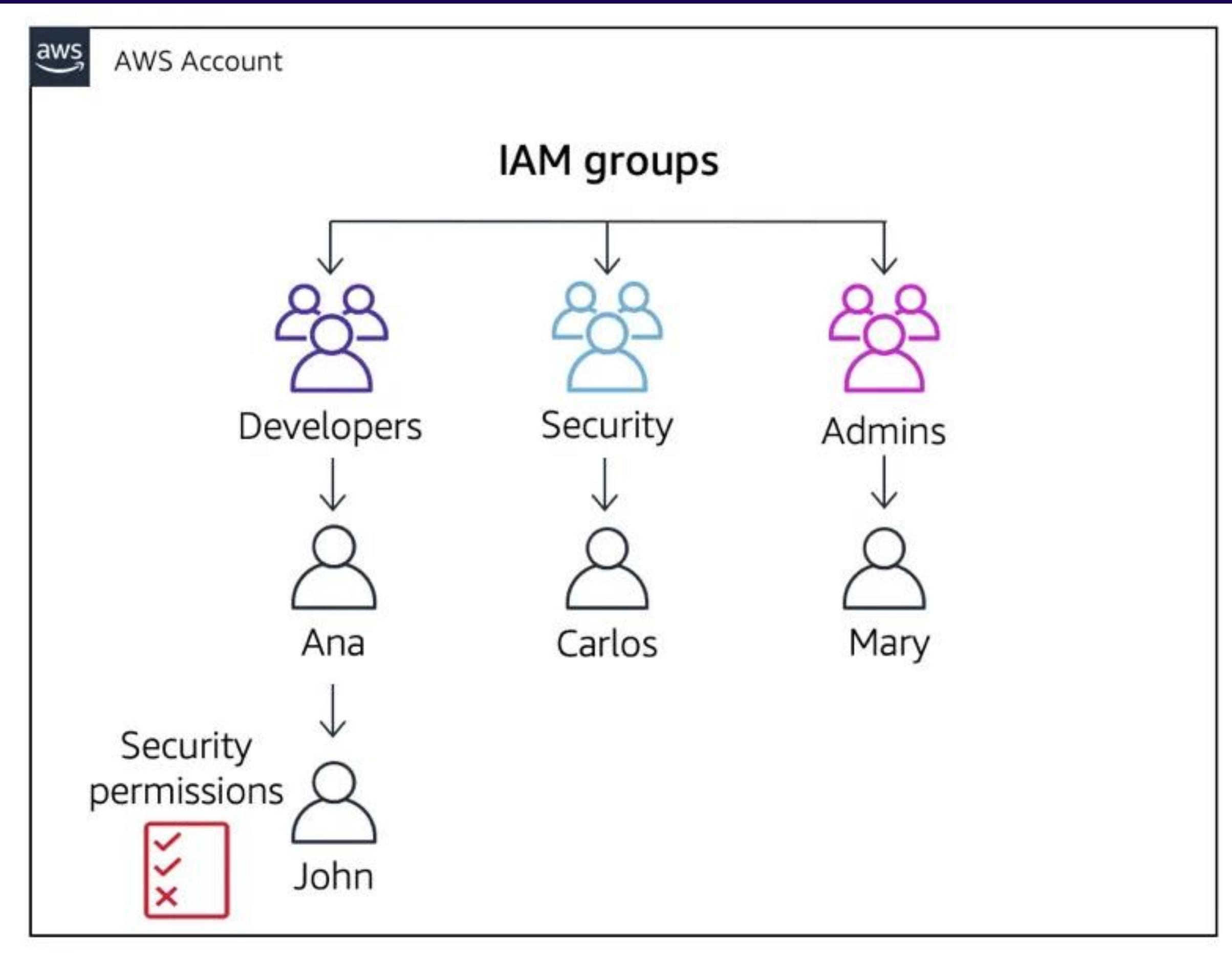
- **Users:** Individual identities with credentials to access AWS
- **Groups:** Collections of users sharing the same permissions
- **Roles:** Temporary access to AWS resources, used by applications or services
- **Policies:** JSON documents defining permissions



AWS IAM Policies

- **Types of Policies:**
 - AWS Managed Policies (predefined)
 - Customer Managed Policies (customizable)
 - Inline Policies (attached directly to an entity)
- **Structure:**
 - Actions: What can be done (e.g., s3>ListBucket)
 - Resources: Where the action is applied (e.g., arn:aws:s3:::bucket-name)
 - Effect: Allow or Deny

AWS Component Relations

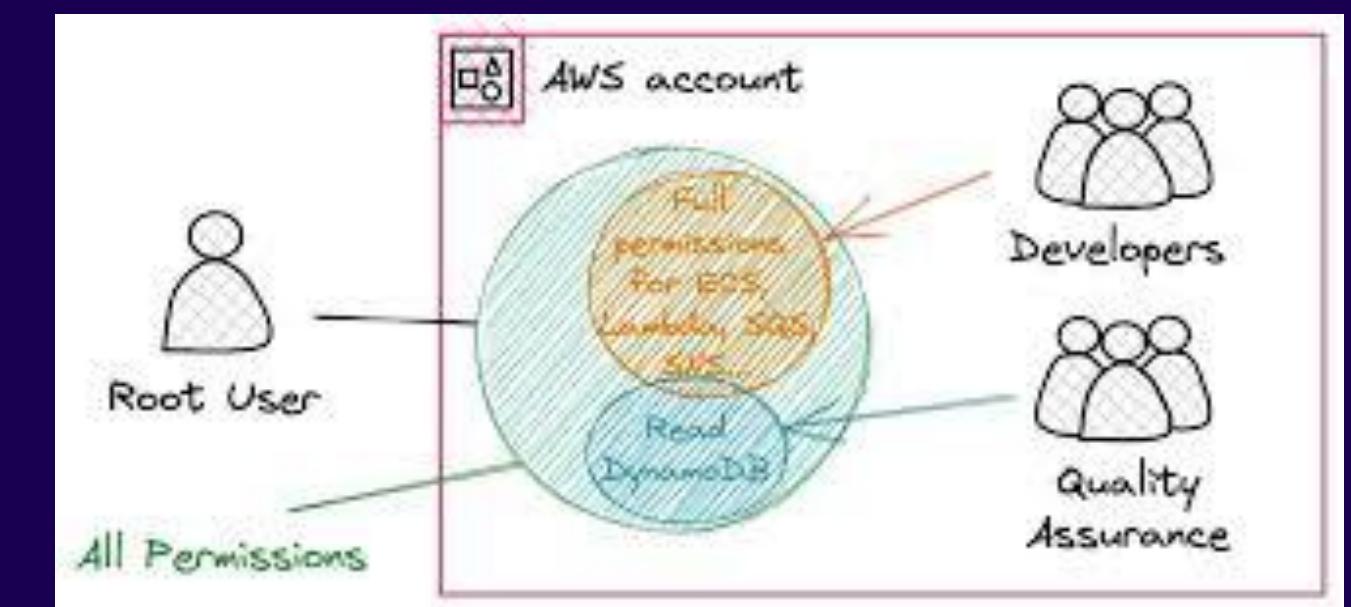


- Users belong to groups
- Groups have attached policies
- Roles are assumed by users or AWS services
- Policy can be attached to a user directly – not recommended.

Machines and instances can have roles.

E.g. You can Create a service Role EC2ToS3.
Attach the policy AmazonS3FullAccess to this role
Assign Role to an EC2 instance

How does AWS evaluate policies?



AWS IAM

First Authentication is done. Some services like S3 allow anonymous access.

The authenticated user is the Principal.

The request context consists of

- Principal
- Actions
- Resources
- Resource Data
- Env. Data

AWS IAM

This information is used to compare against policies to determine if action is allowed or not.

Principal
Action
Resource
Condition

PARC Principle.

AWS IAM

All applicable policies

Identity Policies

Resource based Policies

IAM Permissions Boundaries

Organization SCP

Organization RCP

Session Policies

Are used to evaluate. Explicit Denies over-ride allows.

AWS IAM

- Resource-based policies - Resource-based policies grant permissions for principals specified in the policy.
- The permissions define what the principal can do with the resource to which the policy is attached.

AWS IAM

- Identity-based policies - Identity-based policies are attached to an IAM identity (user, group of users, or role) and grant permissions to IAM entities (users and roles).
- If only identity-based policies apply to a request, then AWS checks all of those policies for at least one Allow.

AWS IAM

- Permissions boundaries - Permissions boundaries are a feature that sets the maximum permissions that an identity-based policy can grant to an IAM entity (user or role).
- When you set a permissions boundary for an entity, the entity can perform only the actions that are allowed by both its identity-based policies and its permissions boundary.
- In some cases, an implicit deny in a permissions boundary can limit the permissions granted by a resource-based policy.

AWS IAM

- service control policies (SCPs) -specify the maximum available permissions for principals within accounts in an organization or organizational unit (OU)
- Resource Control Policies (RCPs) - specify the maximum available permissions for resources within accounts in an organization or organizational unit (OU).

AWS IAM

Session policies - Session policies are policies that you pass as parameters when you programmatically create a temporary session for a role or federated user session.

To create a role session programmatically, use one of the AssumeRole* API operations.

When you do this and pass session policies, the resulting session's permissions are the intersection of the IAM entity's identity-based policy and the session policies.

To create a federated user session, you use the IAM user access keys to programmatically call the GetFederationToken API operation.

AWS IAM

- AWS Organizations service control policies (SCPs) -specify the maximum available permissions for principals within accounts in an organization or organizational unit (OU).
- SCPs apply to principals in member accounts, including each AWS account root user.
- If an SCP is present, permissions granted by identity-based and resource-based policies to principals in your member accounts are only effective if the SCP allows the action.
- The only exceptions are principals in the organization management account and service-linked roles.

AWS IAM

- AWS Organizations RCPs specify the maximum available permissions for resources within accounts in an organization or organizational unit (OU).
- RCPs apply to resources in member accounts and impact the effective permissions for principals, including the AWS account root user, regardless of whether the principals belong to your organization.
- RCPs don't apply to resources in the organization management account and to calls made by service-linked roles.
- If an RCP is present, permissions granted by identity-based and resource-based policies to resources in your member accounts are only effective if the RCP allows the action.

AWS IAM

By default, all requests are implicitly denied with the exception of the AWS account root user, which has full access.

Requests must be explicitly allowed by a policy or set of policies following the evaluation logic below to be allowed.

An explicit deny overrides an explicit allow.

AWS IAM

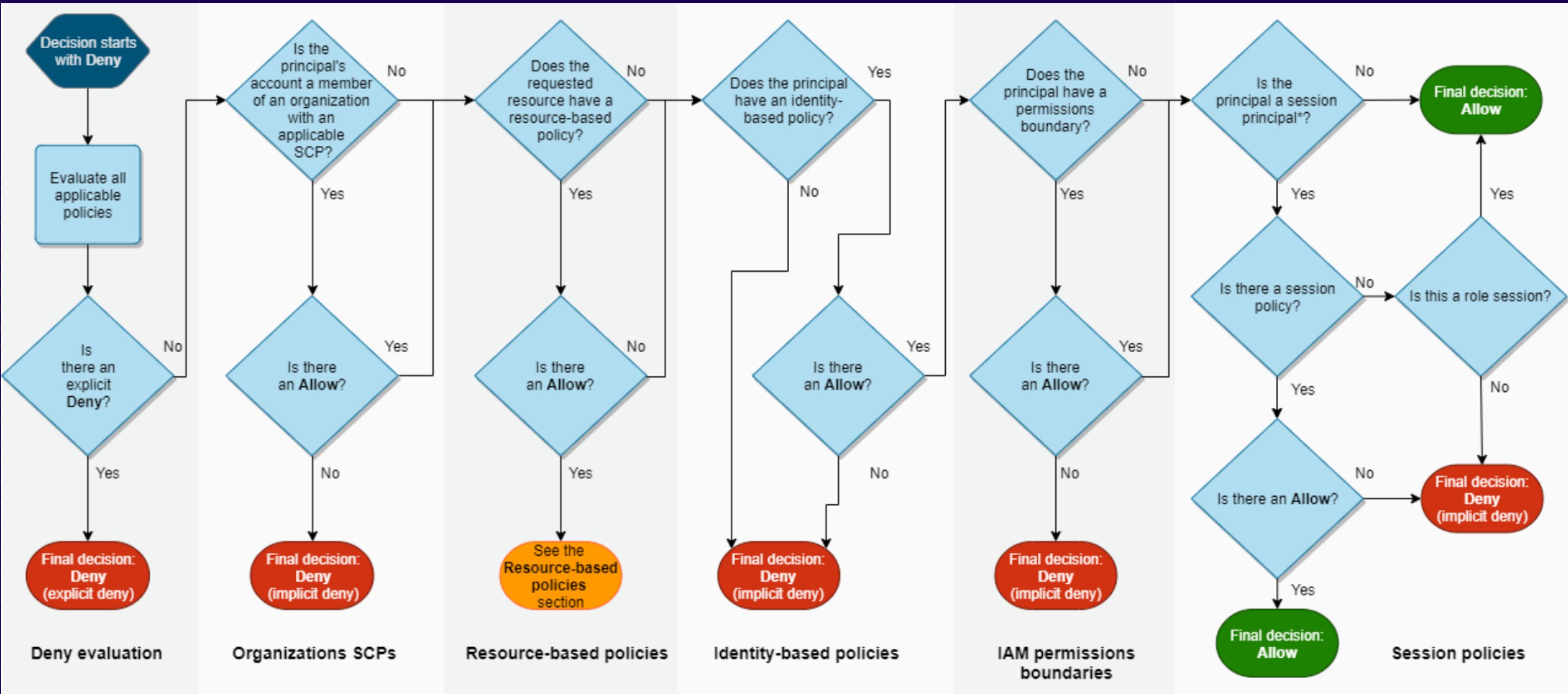
For simplicity, let us only consider requests for a single account.

By default, all requests are denied.

All policies are evaluated and if there's a single deny, it is denied.

If one allow is present, access allowed

AWS IAM



AWS IAM

See here

[AWS IAM Access Analyzer](#)

[AWS IAM Policy Evaluator](#) (not from AWS)

[A cool way to visualize this](#) (not from AWS)

Advanced Topics

User Federation

AWS Directory services

Delegated Permissions

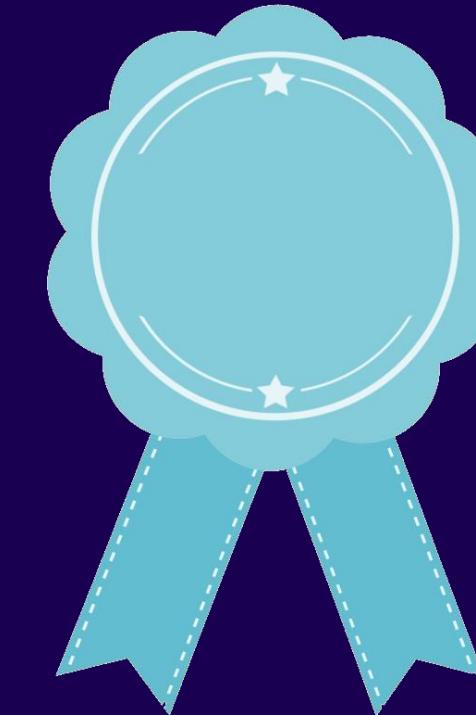
MFA

Security Devices

Least Privileged and Zero Trust architectures

Others - AWS Verified, Custom IDP, Integration with 3rd party IDPs and Platforms

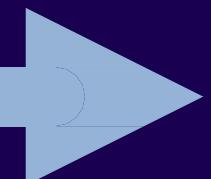
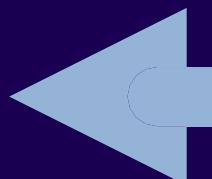
Cryptography: what, how and why



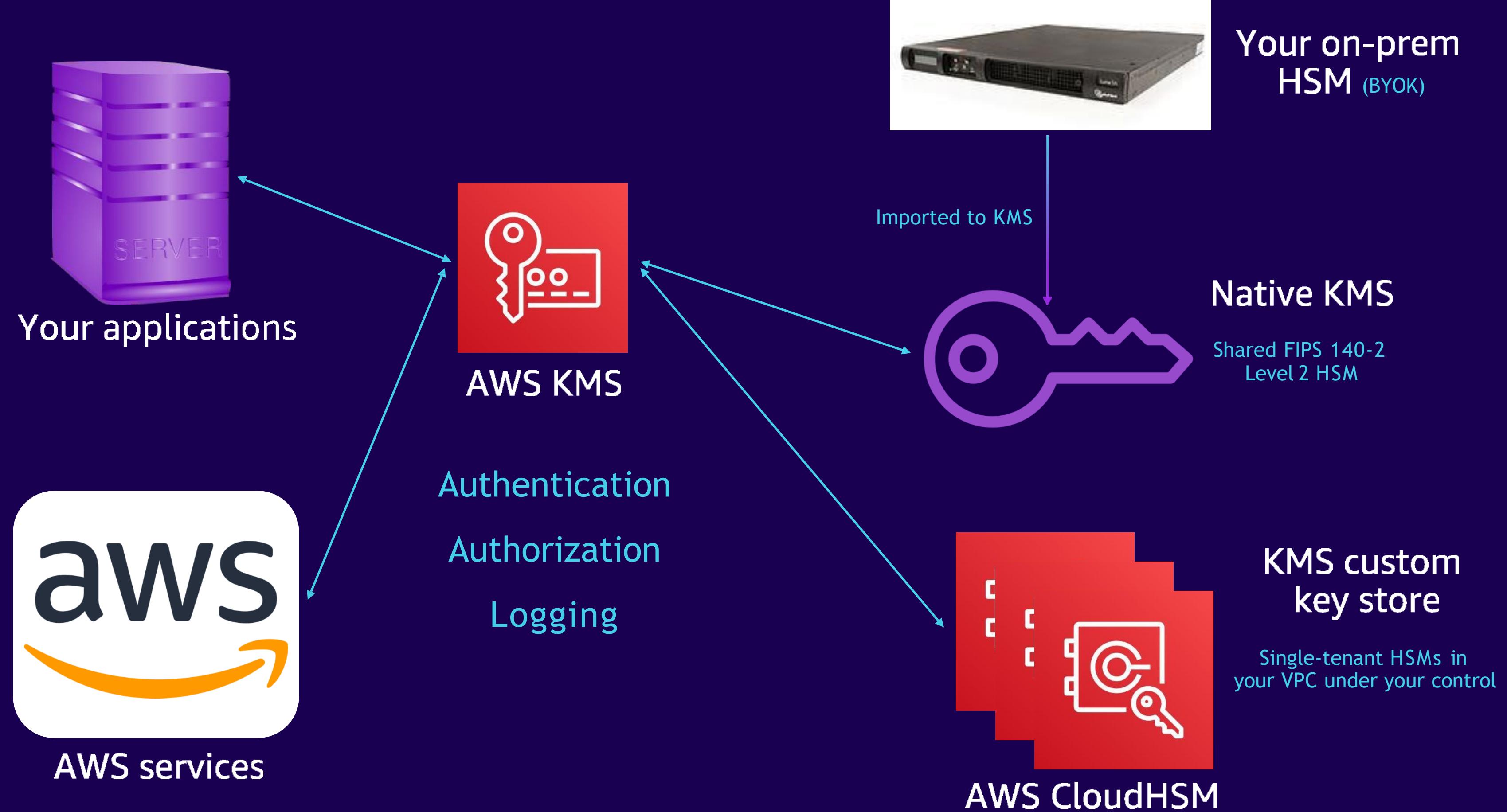
Hardware security module

Managed by
AWS service

Controlled
by you



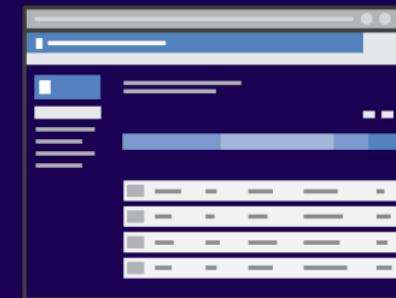
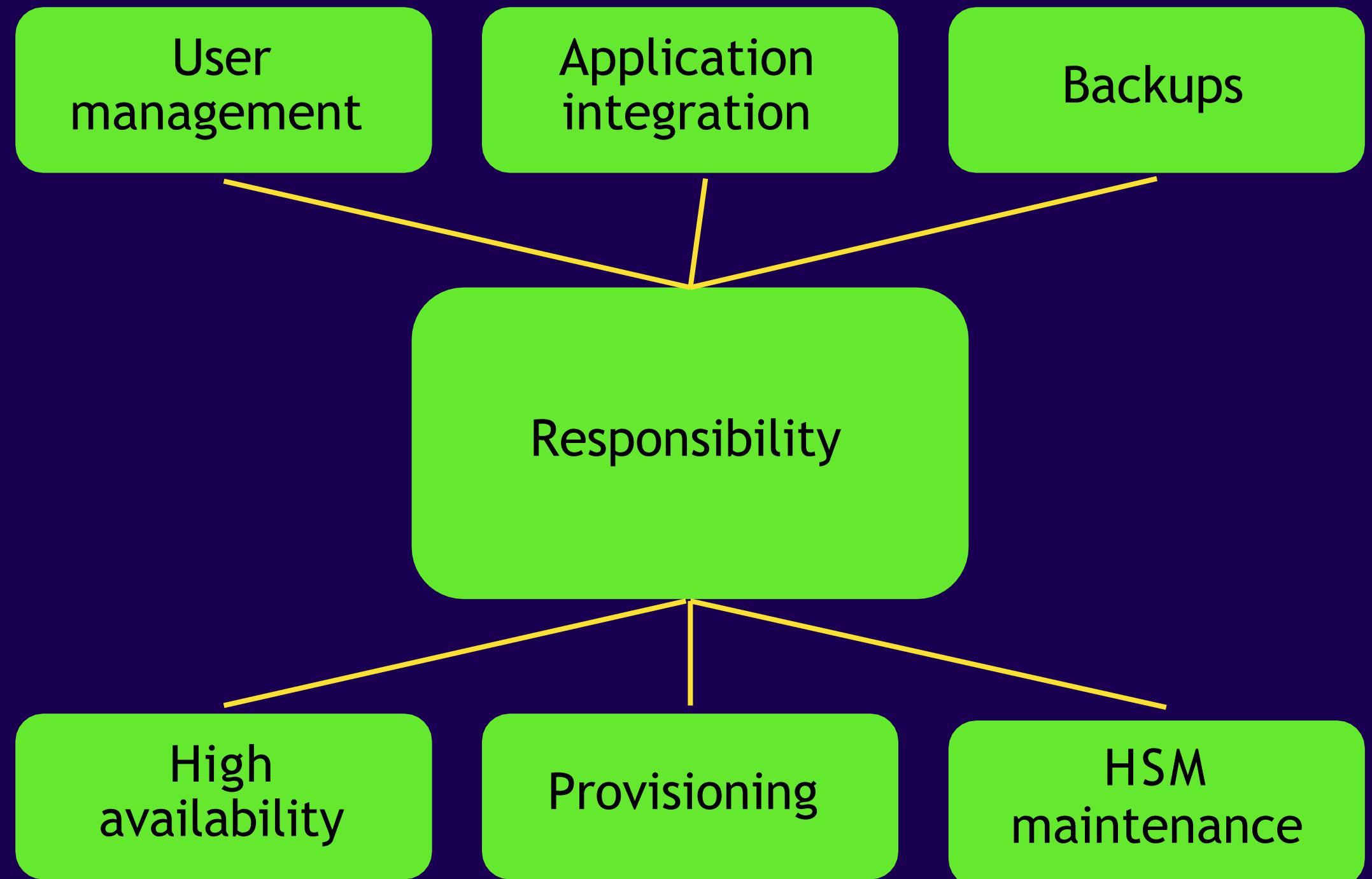
AWS KMS



Now with added asymmetric keys

- RSA and EC keys
- Signing for public key infrastructure (PKI), code, document, timestamp and more
- Encryption for key exchange, proof of knowledge and more
- Supported via AWS Command Line Interface (AWS CLI) today
- Keys generated and stored in FIPS-validated shared HSMs

CloudHSM simplifies management tasks



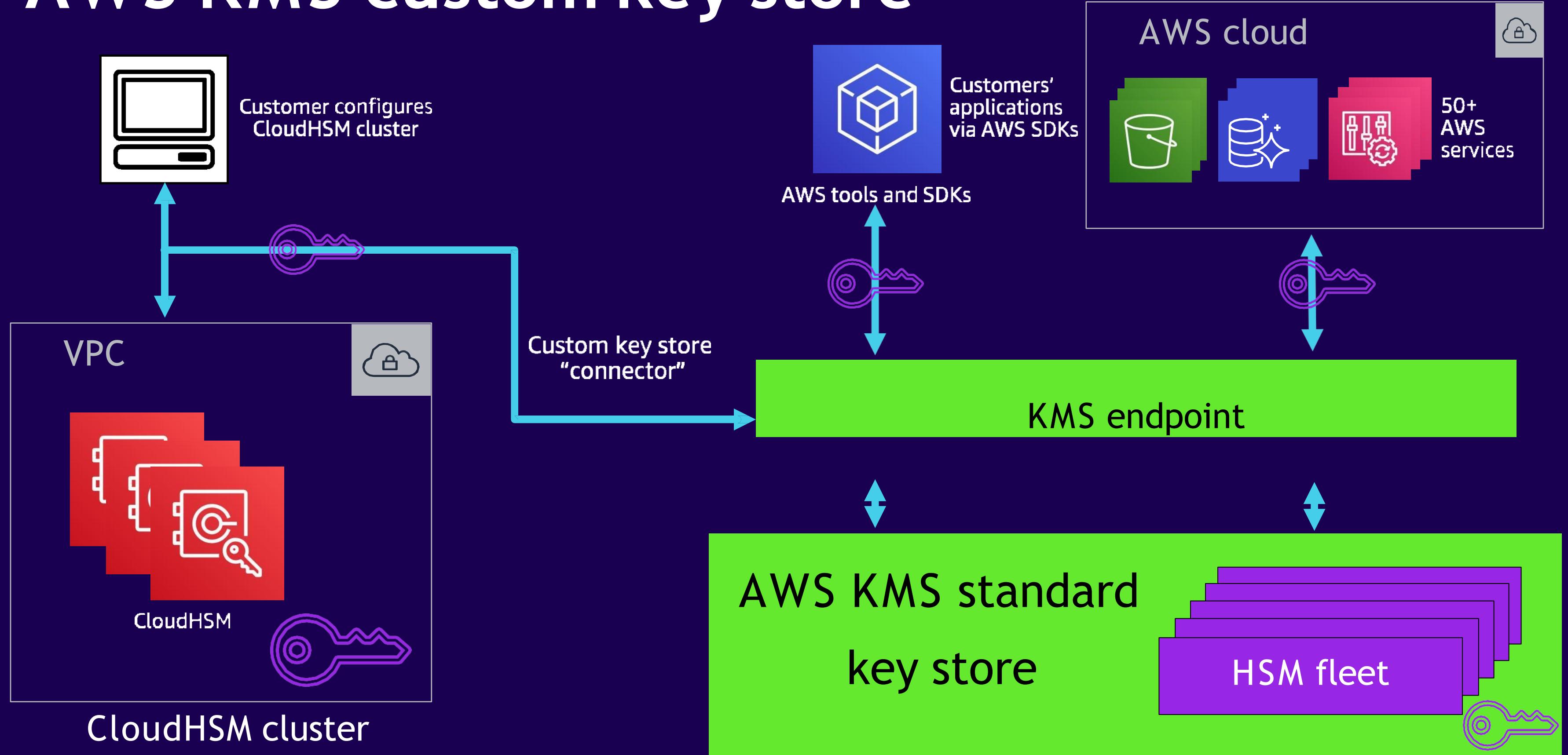
AWS KMS and CloudHSM

	AWS KMS	CloudHSM
Scope	AES-256 and RSA encrypt; RSA and ECC sign	Most general-purpose HSM functions (encrypt, sign /verify, derive, hash, wrap)
Secrets / keys stored in -	Shared FIPS-validated HSM	Single-tenant FIPS- validated HSM in customer VPC
HSM controlled by -	AWS	Customer
Key access by -	AWS Identity and Access Management / resource policies	Customer-defined credentials
Integrated with AWS services -	Yes	No
Secret / key operations implemented with -	AWS CLI / SDK or Encryption SDK	Customer-built application
Scalability managed by -	AWS	Customer
Keys managed by -	AWS	Customer
Rotation executed by -	AWS [not for BYOK and CKS]	Customer

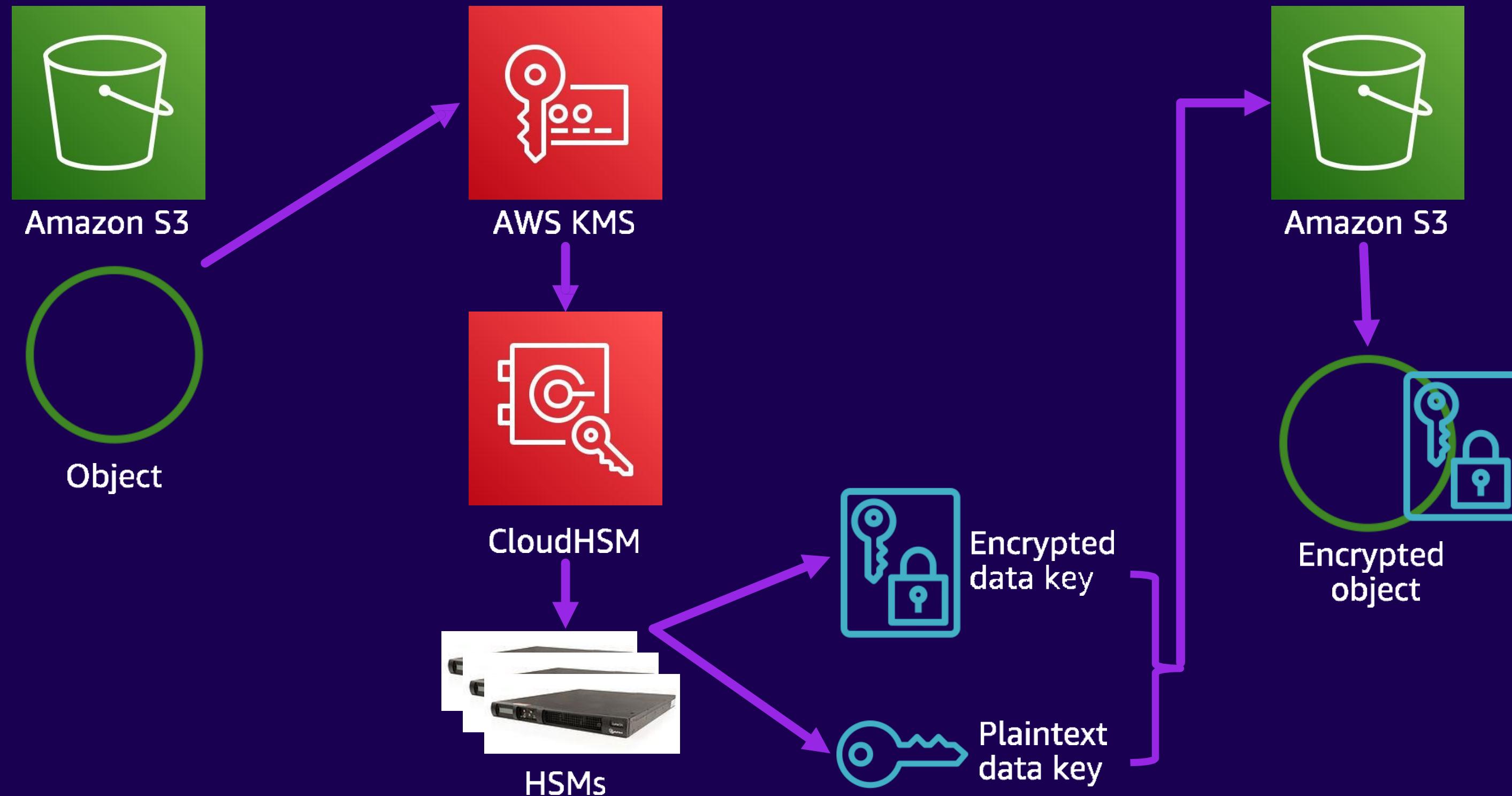
AWS KMS custom key store: best of two worlds

	AWS KMS	KMS custom key store (CloudHSM)
Where keys are generated	HSMs controlled by AWS	HSMs controlled by you
Where keys are stored	HSMs controlled by AWS	HSMs controlled by you
Where keys are used	HSMs controlled by AWS	HSMs controlled by you
How to control key use	JSON key policies you define	JSON key policies you define
Responsibility for performance / scale	AWS	You
Integration with AWS services?	Yes	Yes
Pricing model	\$1/key + usage	\$1/key + usage; hourly charge for each HSM

AWS KMS custom key store

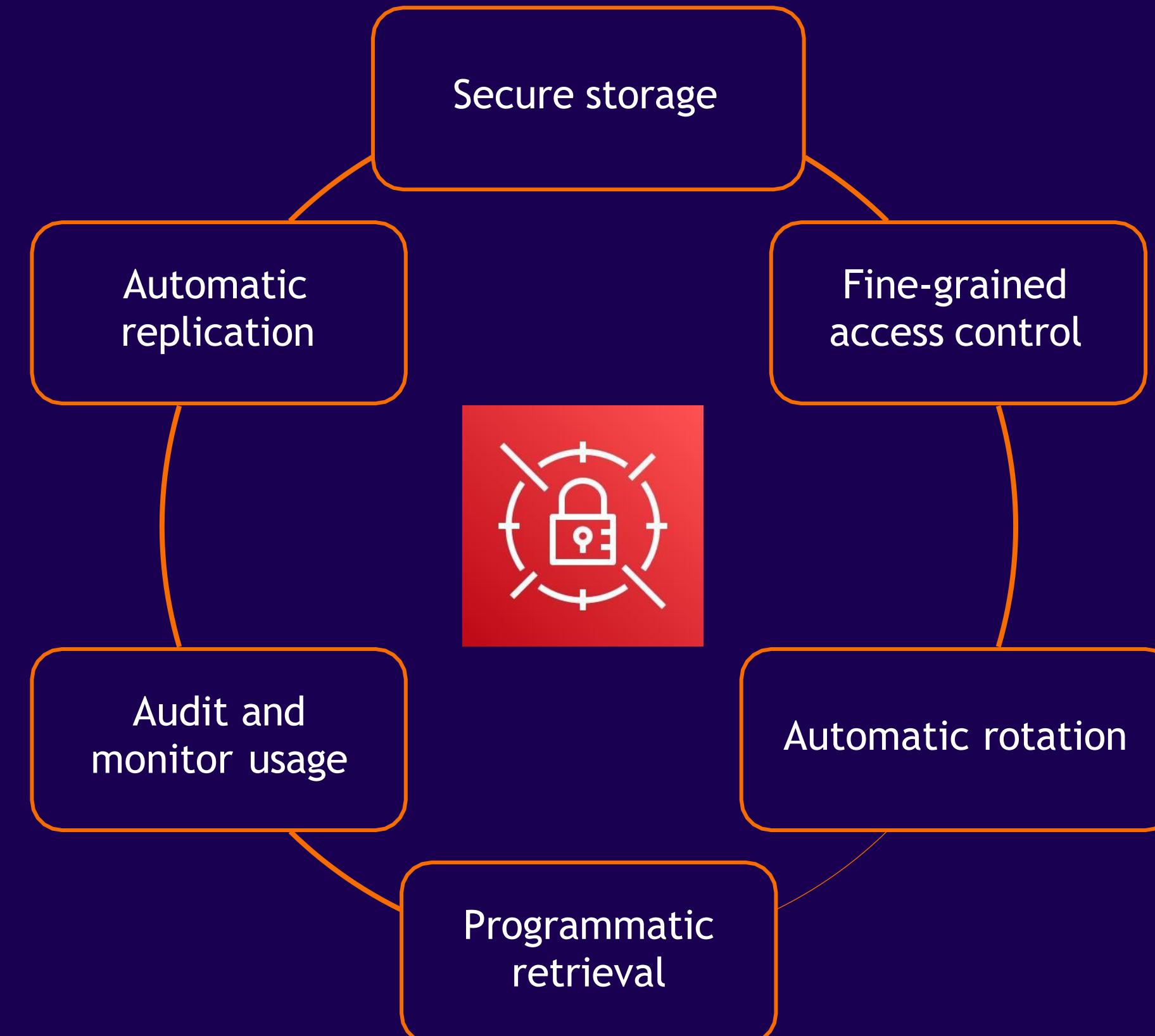


Encrypting an Amazon Simple Storage Service (Amazon S3) object using SSE-KMS

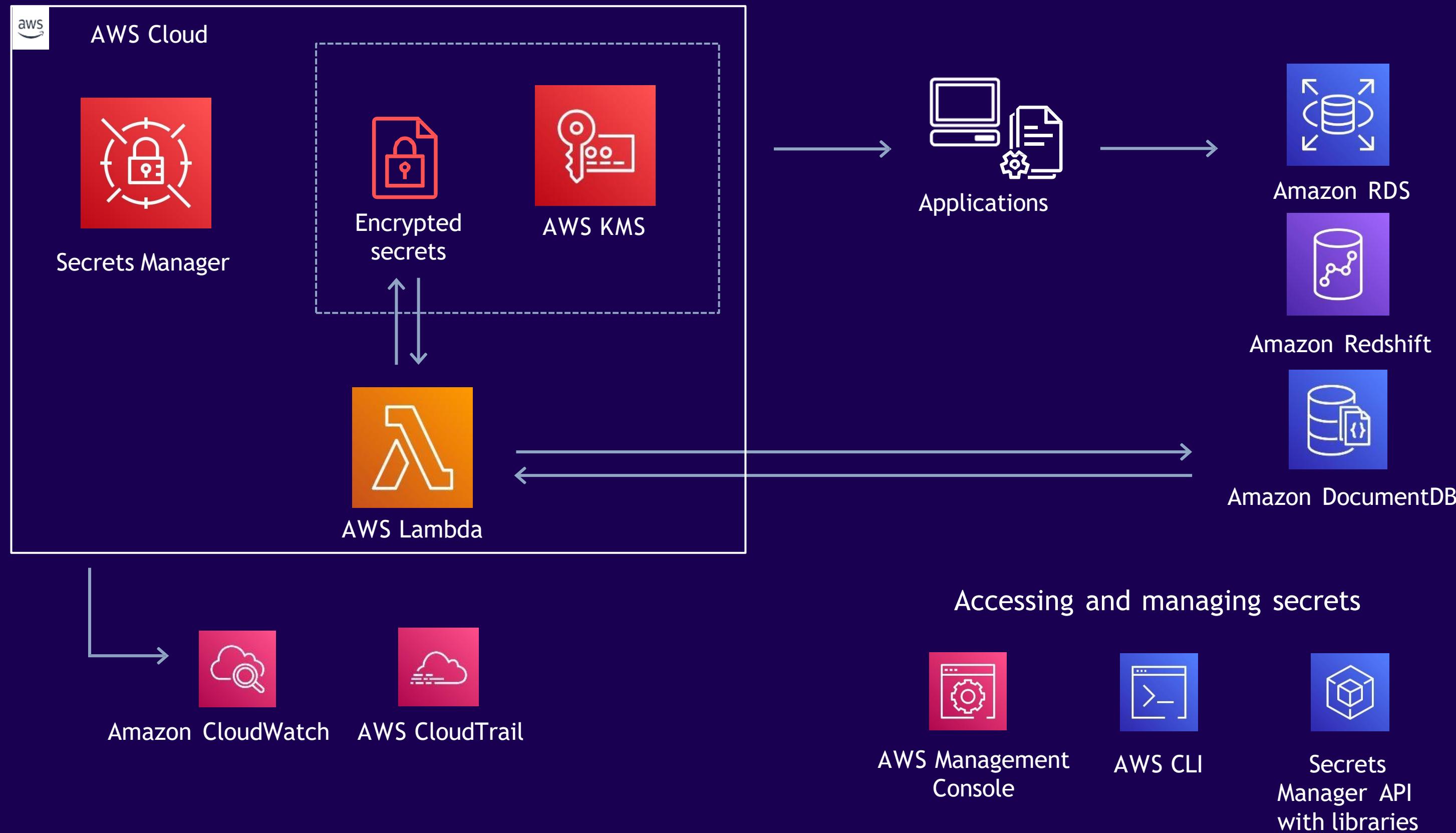


What is Secrets Manager?

Secrets Manager helps customers **manage**, **retrieve**, and **rotate** database credentials, API keys, and other **secrets** throughout their **lifecycles**



How Secrets Manager works



Fully managed database admin secrets

Overview

- You want to securely store and manage administrator credentials to your [Amazon RDS](#) and [Amazon Redshift](#) databases with as little operational overhead as possible
- When you create or modify your database, you can allow the database service to manage the admin password in [Secrets Manager](#), including automatic secret rotation
- Database service generates the password and stores it in [Secrets Manager](#)

Benefits

- You do not need to configure or manage AWS Lambdas for secret rotation
- Password is generated by the relevant database service (e.g., Amazon RDS) and is never visible to your administrators
- You are in full control of access permissions and encryption key configuration, like any other secret in [Secrets Manager](#)



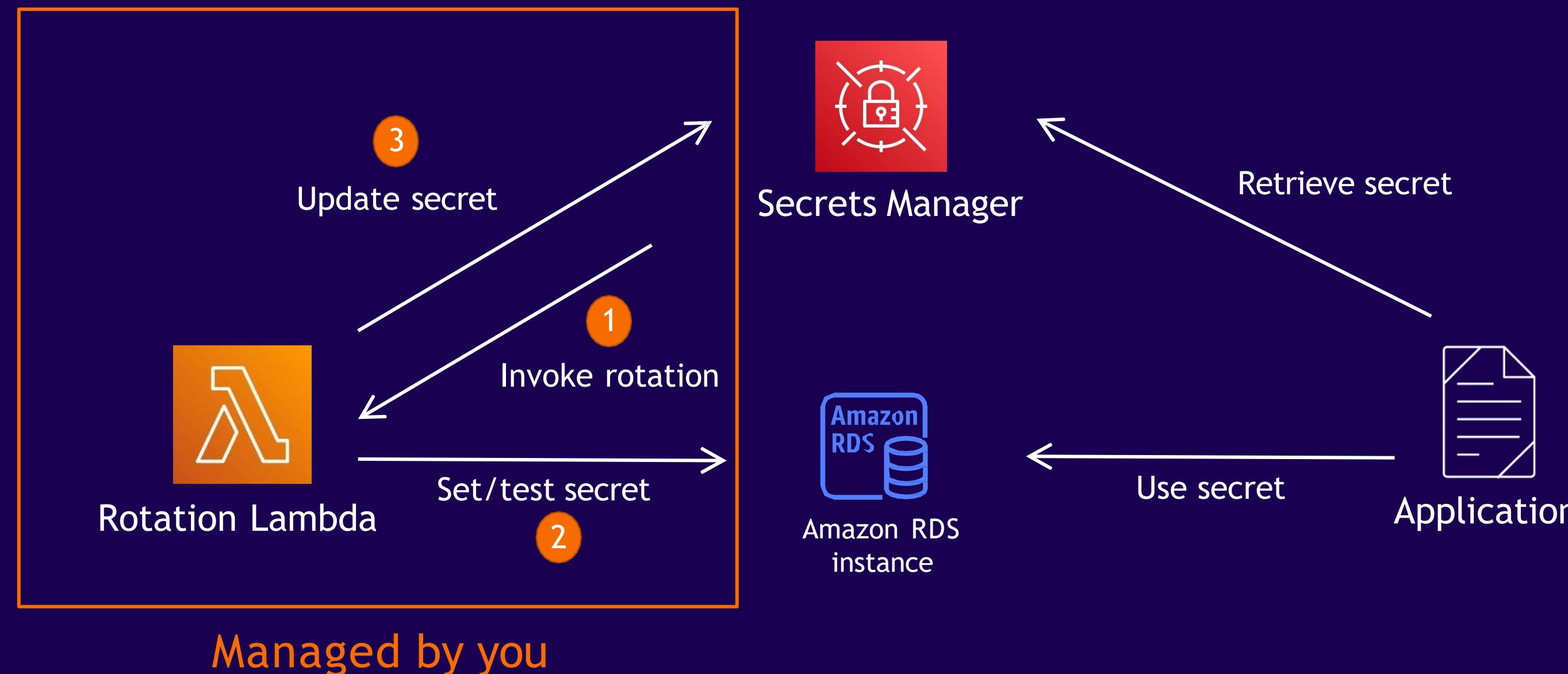
Amazon Redshift



Amazon RDS

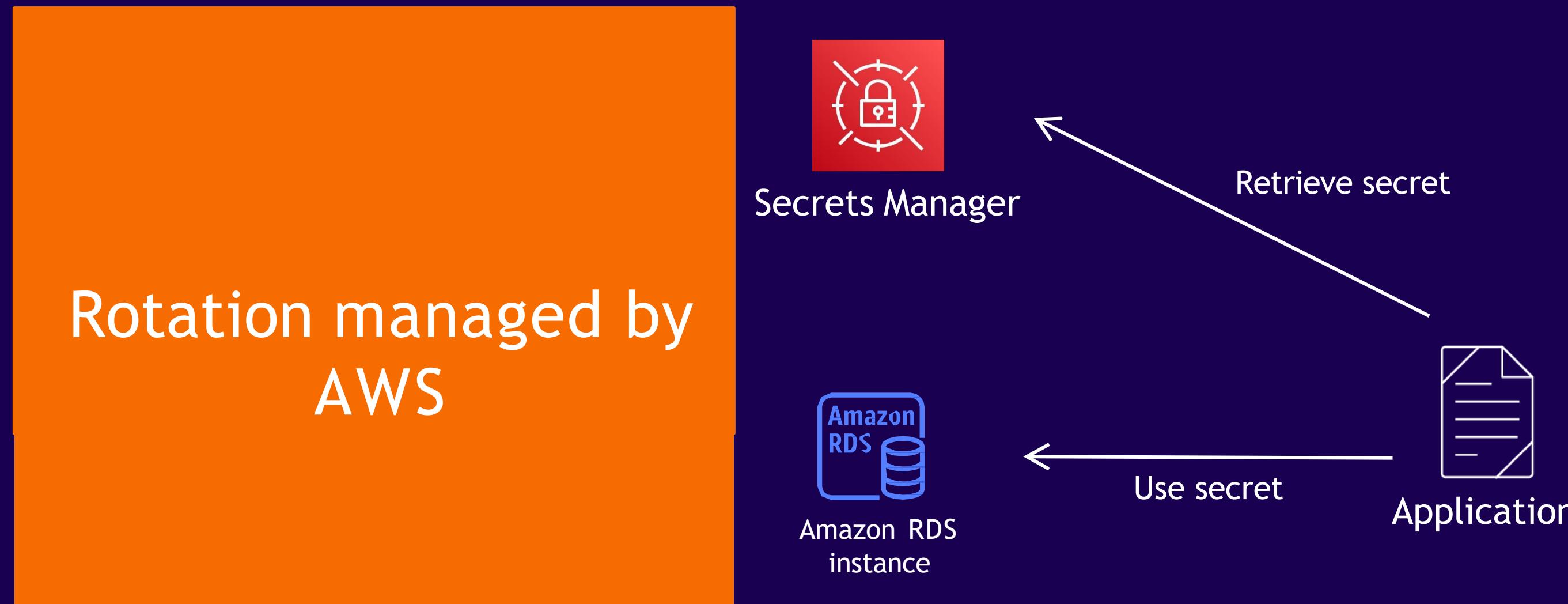
Secret rotation managed by you

For secrets that you create, you are responsible for managing rotation Lambdas



Secret rotation managed by AWS

When you enable admin password management with Secrets Manager, AWS manages rotation on your behalf

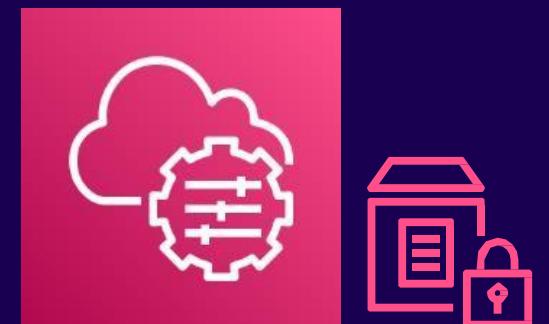


Secrets Manager and AWS Systems Manager Parameter Store

Systems Manager Parameter Store is a capability that allows customers to store configuration management data. While Parameter Store does allow for encryption of sensitive parameters, Secrets Manager offers several features that make it the preferred choice for managing access to secrets.

Parameter Store benefits

- Low cost - “standard” parameters are **free**
- Provides a scalable, auditable repository for configuration data and database strings that are **not considered sensitive**



Systems Manager Parameter Store

Secrets Manager benefits

- Automatic secret rotation (for supported services)
- Secrets **encrypted by default**
- Client-side caching libraries **can improve performance** and **reduce cost** during secret retrieval
- Random secret generation
- **Minimum recovery period** ensures secrets cannot be accidentally deleted and lost

Secrets Manager and AWS Systems Manager Parameter Store

Secrets Manager is preferred when securely storing and accessing **sensitive data**, such as database credentials or application secrets. Secrets Manager Parameter Store should primarily be used for **non-secret configuration data**, such as license information or database connection strings.

Parameter Store use cases

- Database strings
- Amazon AMI IDs
- License codes

Secrets Manager use cases

- Database credentials
- Login information (username and password)
- Application secrets (Lambda, Amazon EC2, Amazon ECS/Amazon EKS)
- Third-party API keys or authentication tokens
- SSH keys



Secrets Manager

Secrets Manager access control policy types

Identity-based policies	Resource policy on secret	Resource policy on KMS key
1 to many	Many to 1	Many to 1
Provide access to multiple secrets for a single IAM entity (e.g., role)	Provide access to a single secret for multiple IAM entities (e.g., roles)	Provide access to a single KMS key for multiple IAM entities (e.g., roles)
Control permissions to API actions for resources that don't yet exist, or grant permissions to create new resources	Only controls permissions to an existing secret - the secret to which the resource policy is attached	Only controls permissions to an existing key - the secret to which the resource policy is attached
Grant access to users, roles, or groups in the same account	Grant access to users or roles in other AWS accounts	Grant access to users or roles in other AWS accounts
Required	Optional	Required

Scaling access using tags (ABAC)

- For rapid growth environments
- Simplifies management of access control pattern
- Create 1 or a small number of policies to **scale out access** to N number of secrets
- Strict **tagging enforcement** required

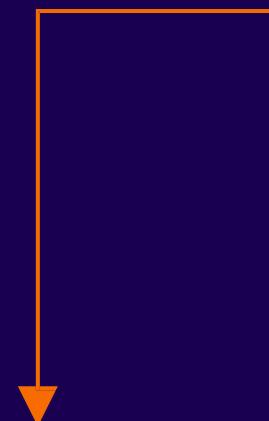
```
{  
    "Version" : "2012-10-17",  
    "Statement" : [ {  
        "Effect": "Allow",  
        "Principal" : {"AWS": "123456789012"},  
        "Condition" : {  
            "StringEquals" : {  
                "aws:ResourceTag/access-project": "${aws:PrincipalTag/access-project}"  
            }  
        },  
        "Action" : "secretsmanager:GetSecretValue",  
        "Resource" : "*"  
    } ]  
}
```

Example policy to manage secrets using tags

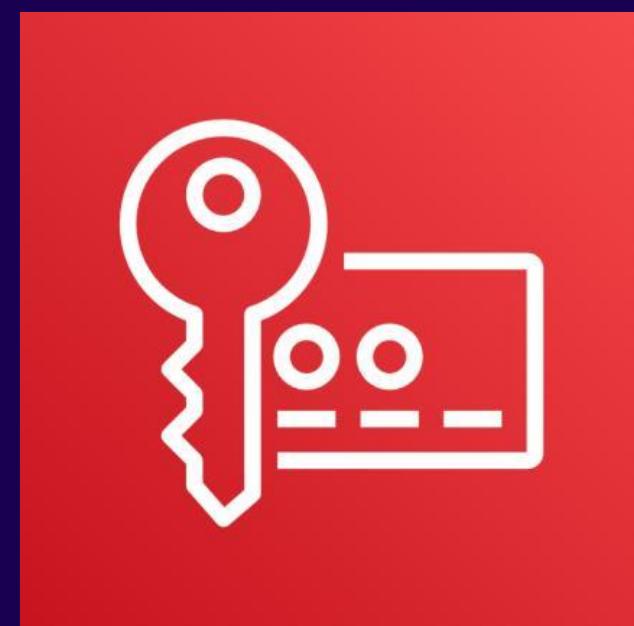
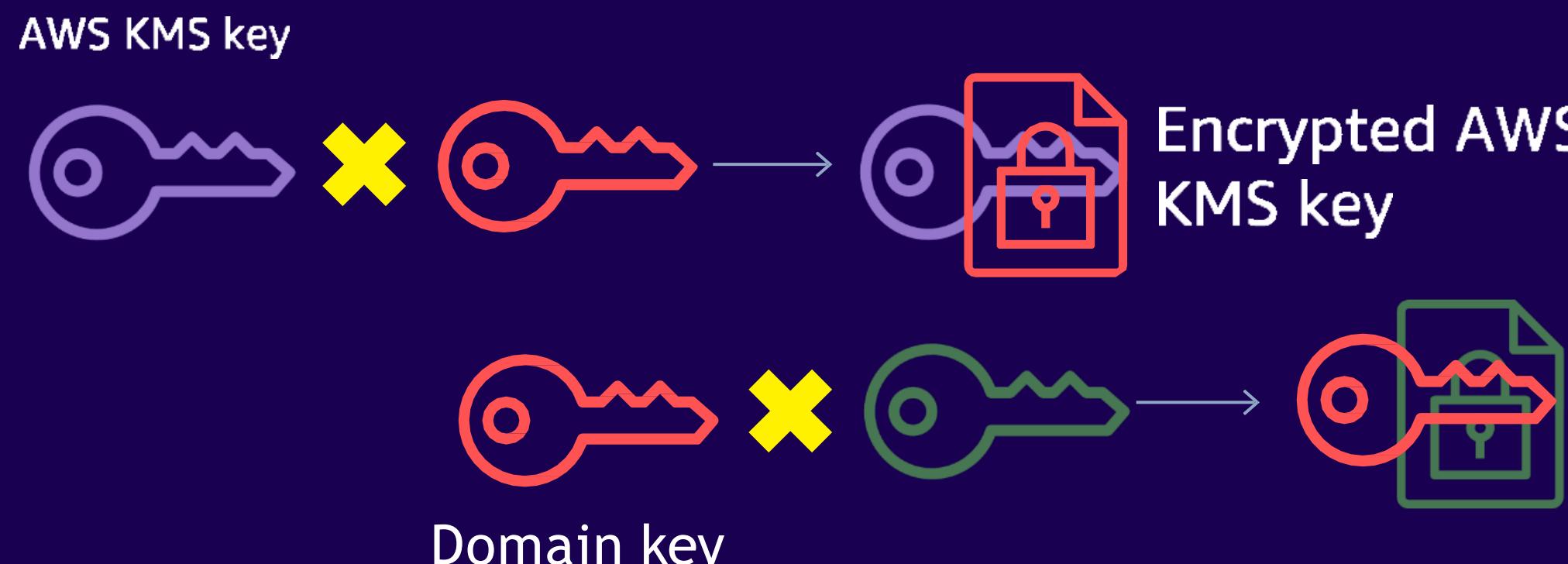
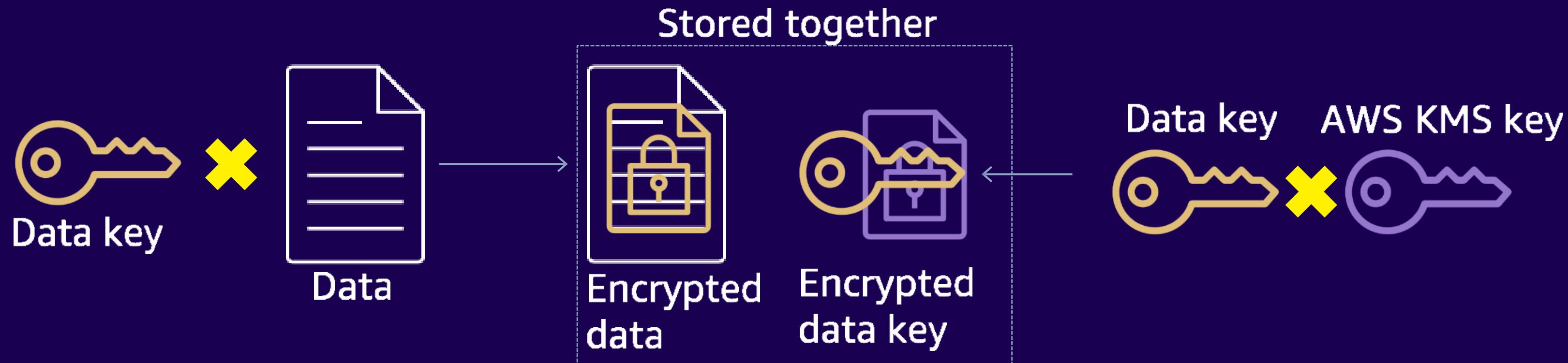
```
{
```

```
  "Effect": "Allow",
  "Action": [ "secretsmanager:GetResourcePolicy",
    "secretsmanager:GetSecretValue",
    "secretsmanager:DescribeSecret",
    "secretsmanager:RestoreSecret",
    "secretsmanager:PutSecretValue",
    "secretsmanager:UpdateSecretVersionStage",
    "secretsmanager>DeleteSecret",
    "secretsmanager>ListSecretVersionIds",
    "secretsmanager:UpdateSecret" ],
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "secretsmanager:ResourceTag/project": "${aws:PrincipalTag/project}"
    } } ] }
```

Only allow API actions
if these tags match



Envelope encryption primer

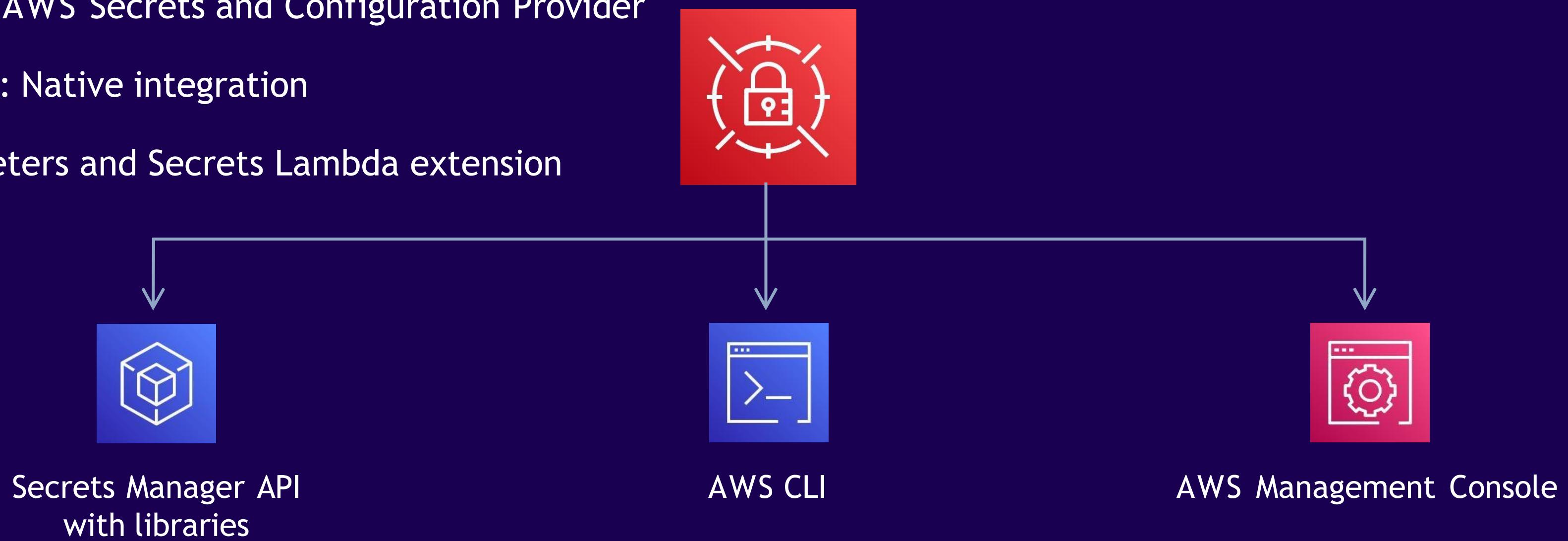


AWS KMS

Leverage existing integrations to consume secrets

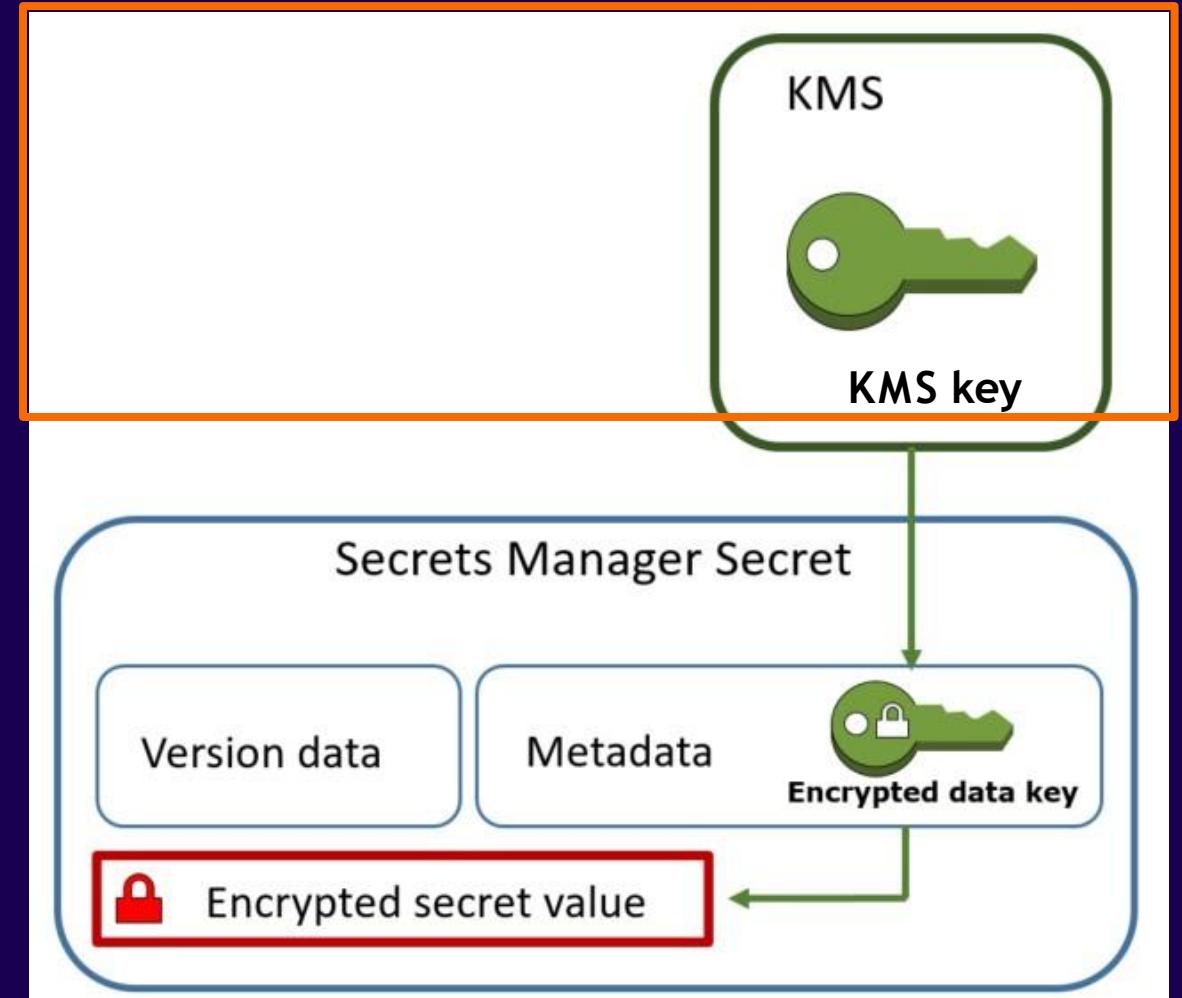
Specialized integrations for secret retrieval

- Kubernetes: AWS Secrets and Configuration Provider
- Amazon ECS: Native integration
- AWS Parameters and Secrets Lambda extension



How encrypting secrets works

- Encryption of secrets is enabled by default and cannot be disabled
- Secrets are encrypted with a data key, and that data key is encrypted with an AWS KMS key
- Encrypted data key is stored along with secret (in metadata)
- Secrets are scrubbed from memory after encryption and are not saved, unencrypted, in durable storage



Hybrid and multicloud access patterns

Overview

- You want to use AWS to store and manage database credentials that can be accessed by resources **in AWS and on-premises devices**, such as application servers or Kubernetes clusters
- You can use the **same IAM roles and policies** you have configured for your AWS workloads to provide access to AWS resources

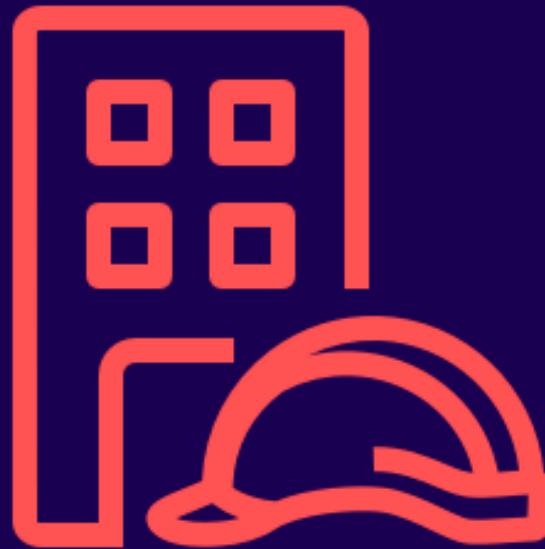
Benefits

- Eliminate use of long-lived credentials to access AWS services such as Secrets Manager
- Improve integration of hybrid workloads by standardizing on AWS security services
- Migrate from products with expensive licensing fees in favor of pay-as-you-go services from AWS

IAM Roles Anywhere

EXTENDS THE USE OF IAM ROLES TO WORKLOADS OUTSIDE OF AWS

IAM Roles Anywhere



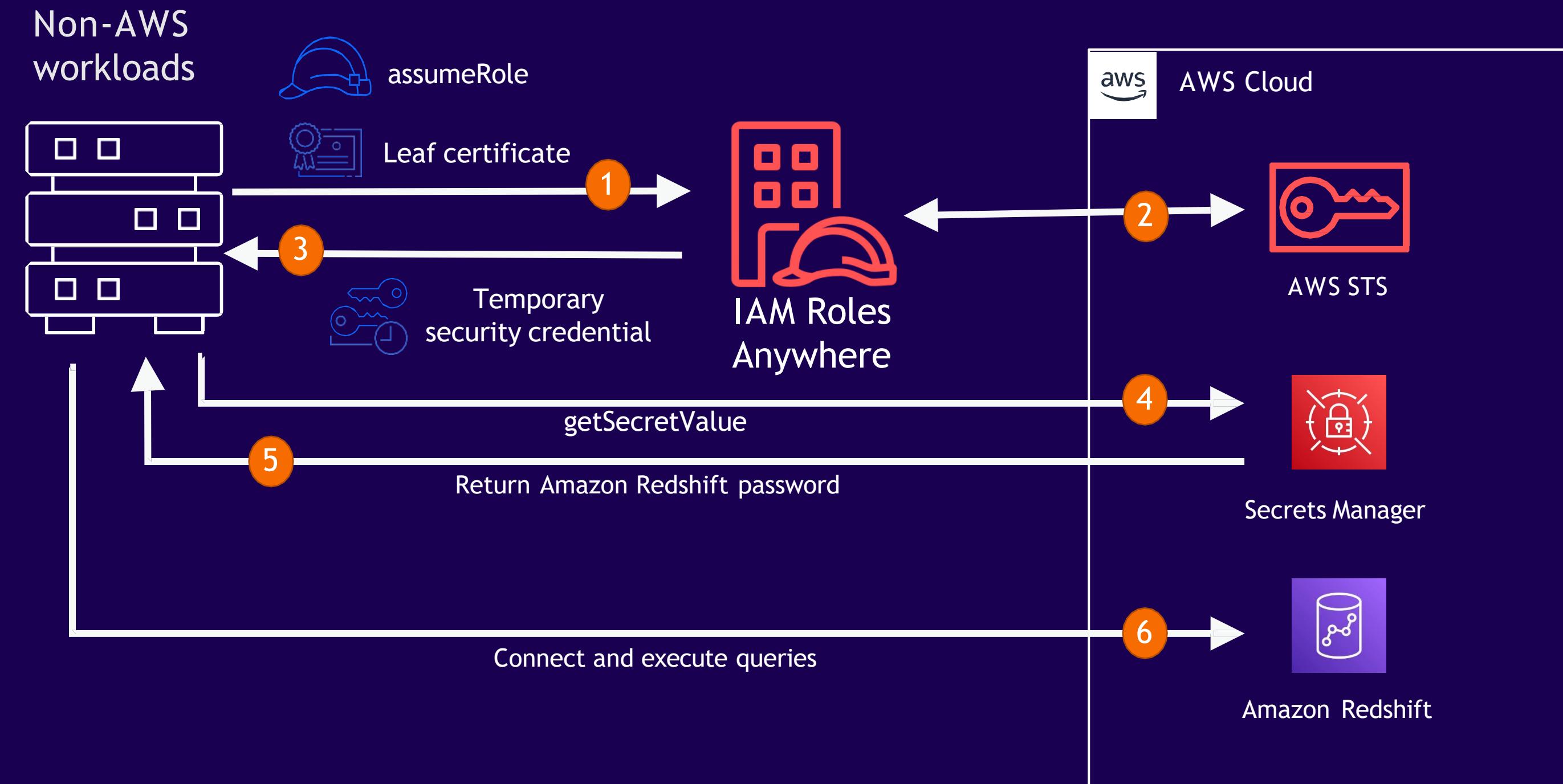
Uses the same IAM roles and policies you have already configured for your AWS workloads to access AWS resources on premises

Obtain temporary AWS credentials and eliminate long lived credentials

Use the same access controls, deployment pipelines, and testing processes across all your workloads

Simplify the migration of your workloads running outside of AWS

Hybrid workloads use case



New capability to retrieve secrets in a group

Overview

- You want to use AWS to retrieve multiple secrets at once, as part of a group of related secrets
- You can now use AWS Secrets Manager to group related secrets and retrieve them with a single API call



AWS Secrets Manager

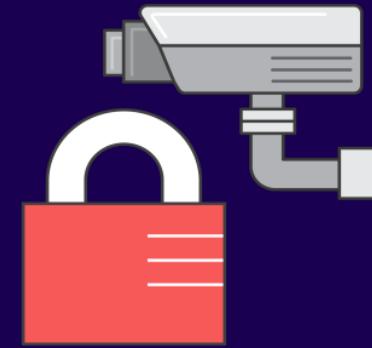
Benefits

- Optimize application workloads by streamlining the process to identify and retrieve a group of secrets
- Improve performance by decreasing the number of API calls, leading to reduced costs
- Enforce the same level of security, with the same access control policies in place for each individual secret

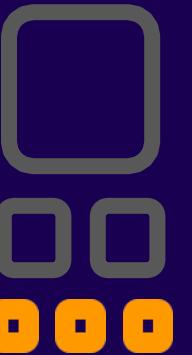
Challenges of operating private certificate authorities

- Security, accountability, and availability issues
- Require infrastructure and security expertise
- Complex
- Expensive

ACM Private CA



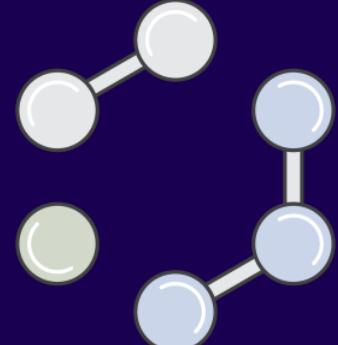
Secure and managed
private CA service



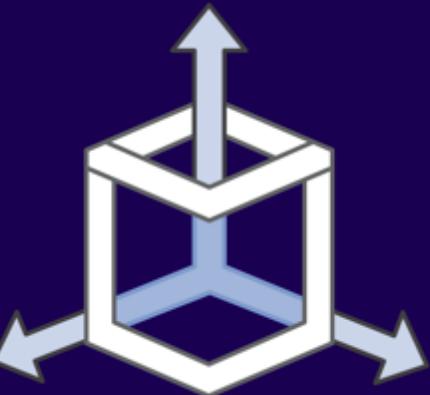
Subordinate CAs



Enable developer agility



Flexibility to customize
private certificates

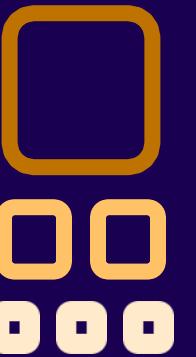


Manage certificates
centrally



Pay-as-you-go pricing

Root CA hierarchies for ACM Private CA

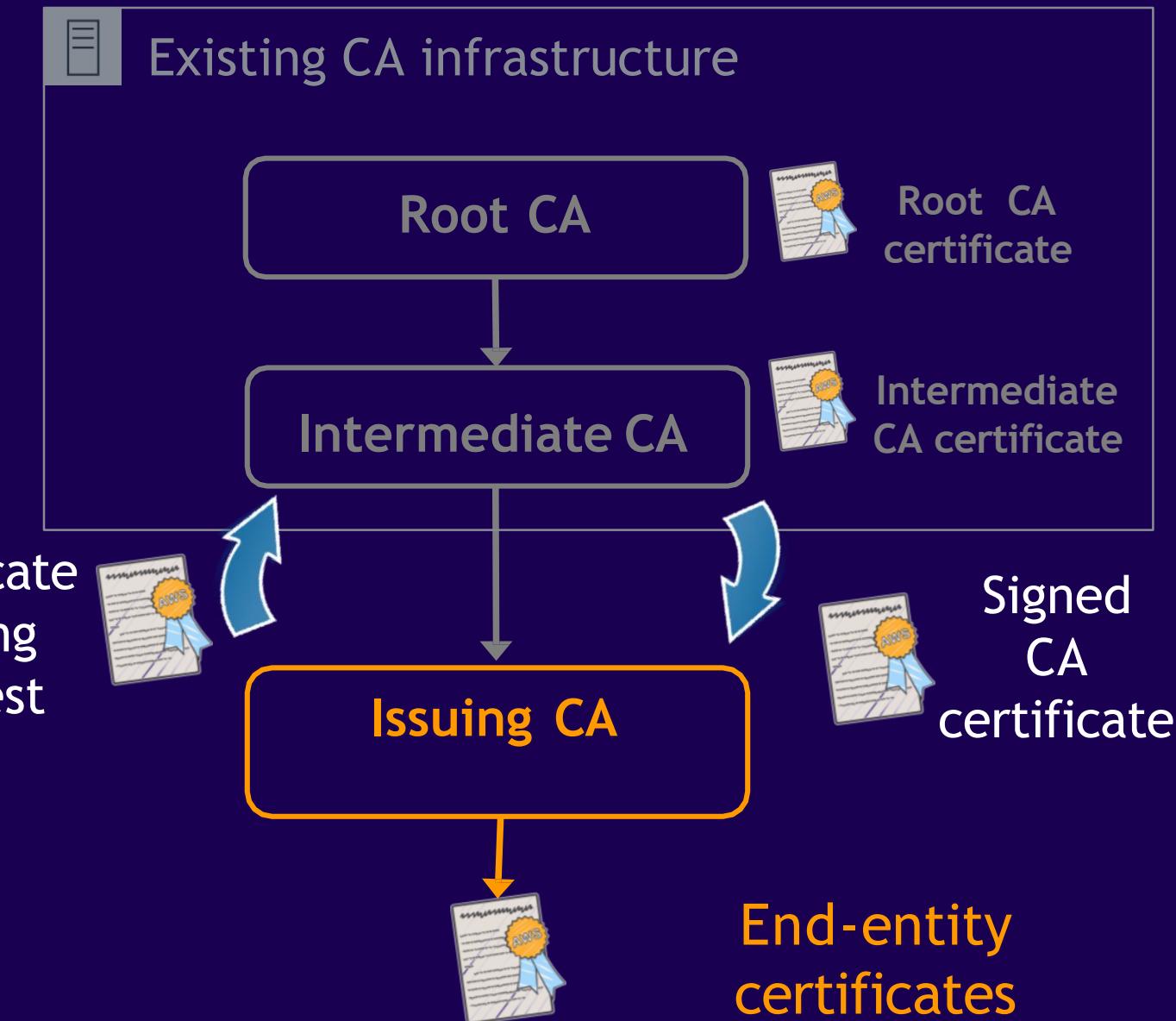


Root CA and complete
CA hierarchies

CA administrators can now create a **complete CA hierarchy**,
including root and subordinate CAs, with no need for
external CAs

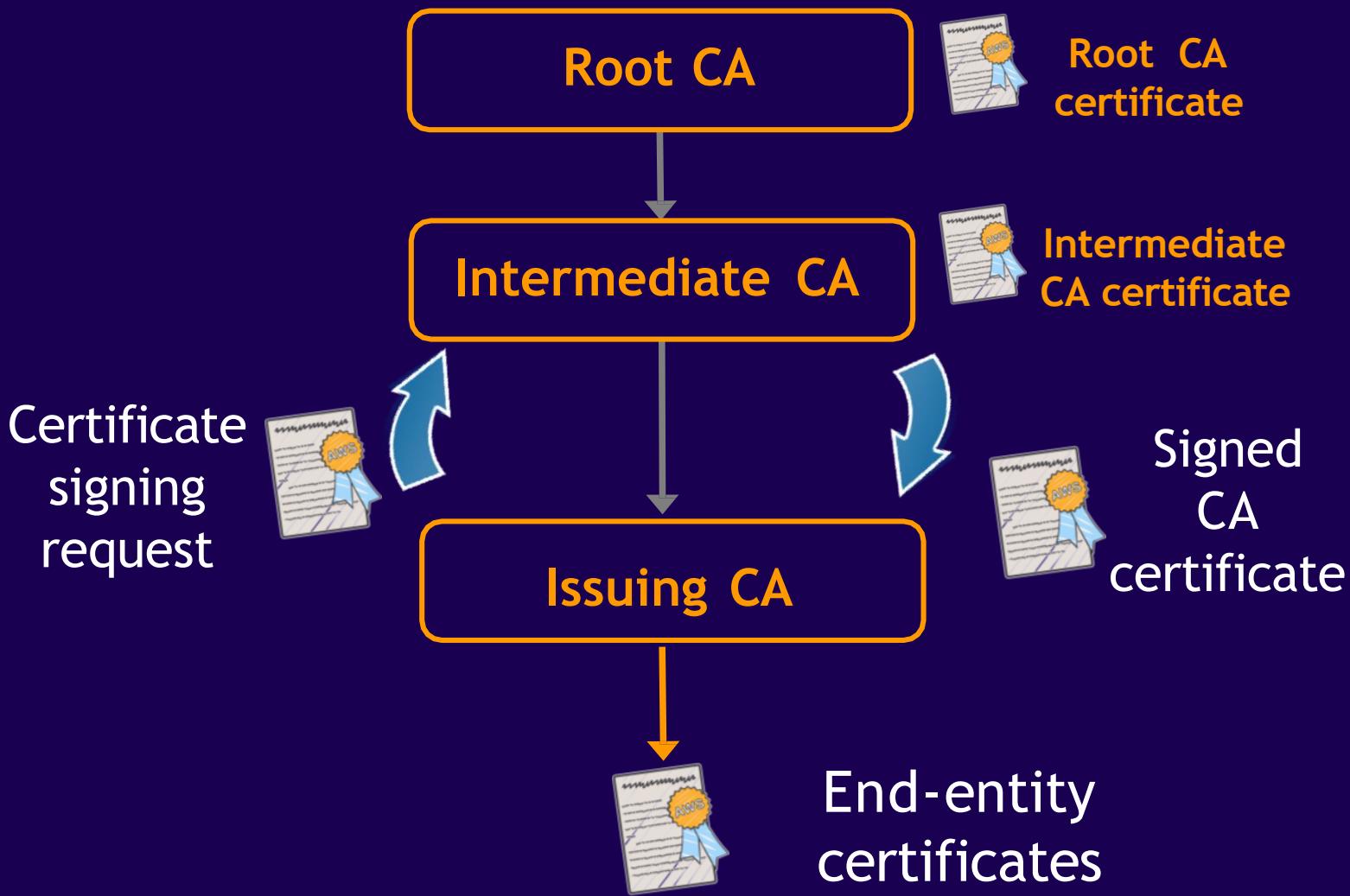
Before ACM Private CA hierarchies

- Subordinate issuing CA with existing (external) intermediate and root CA
- Issuing CA used for bulk issuance of end-entity certificates



ACM Private CA hierarchies

- Complete CA hierarchy, including root CA
- Third-party external CA is now **optional**



Why create a CA hierarchy?

- Restrict access to the root CA
- Grant more permissive access to subordinate CAs
- Delegate subordinate CAs for different applications/groups
- Audit and generate alarms for every certificate issued by root
- Audit random samples of bulk certificates issued by subordinates

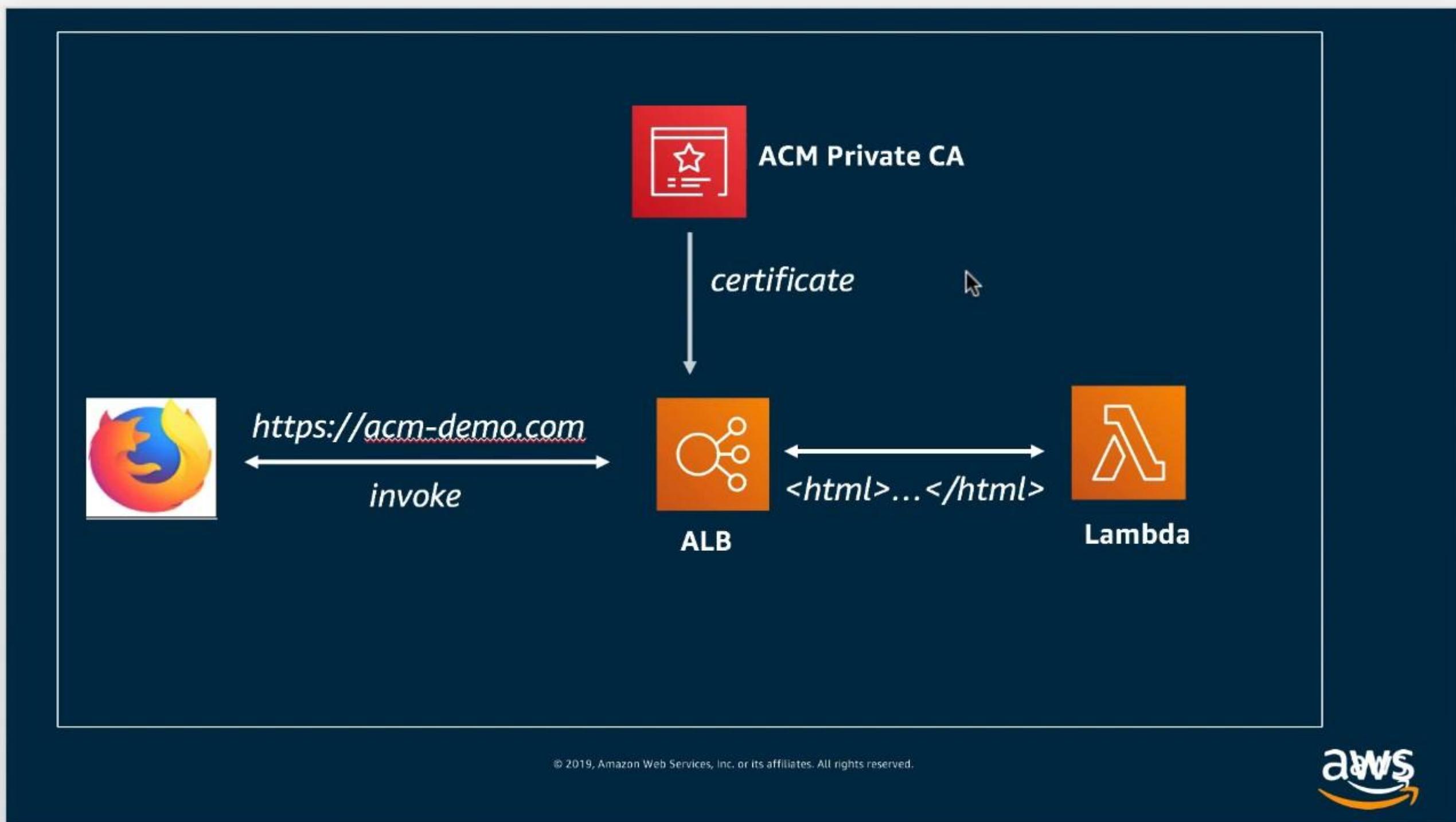
What can end-entity certificates identify?

- TLS endpoints and resources
- IPSec VPN endpoints
- Dynamic cloud resources
- IoT devices

Use cases

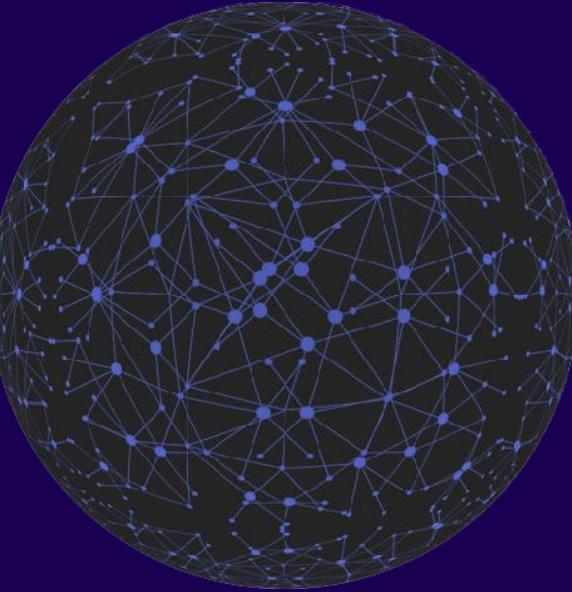
- Replace software/server-based CAs
- Replace offline root CA
- Complement an existing root CA
 - Identify cloud resources for dev/test/production with a cloud-hosted CA infrastructure and APIs

▼ AWS RE:INFORCE Template Dark



Offline vs. online root CA

Offline CA - physical HSM, network-disconnected, stored in a vault (typically)
Best choice depends on your requirements and internal policies



- Physical access controls and isolation
- One or more operators open the vault
- Perform signing ceremony to use the CA
- No network access
- Logical access controls and isolation
- Faster signing (important if a CA certificate expires)
- Easier management
- Doesn't require physical presence

Logical access controls and isolation

- Account separation
- Access controls
 - IAM-managed policies
 - Disable CAs by default
 - Custom policies for two-person control
- Auditing and logging
 - Alarm on certificate issuance for root CAs
 - Careful review of each certificate issued by root and other top-level CAs

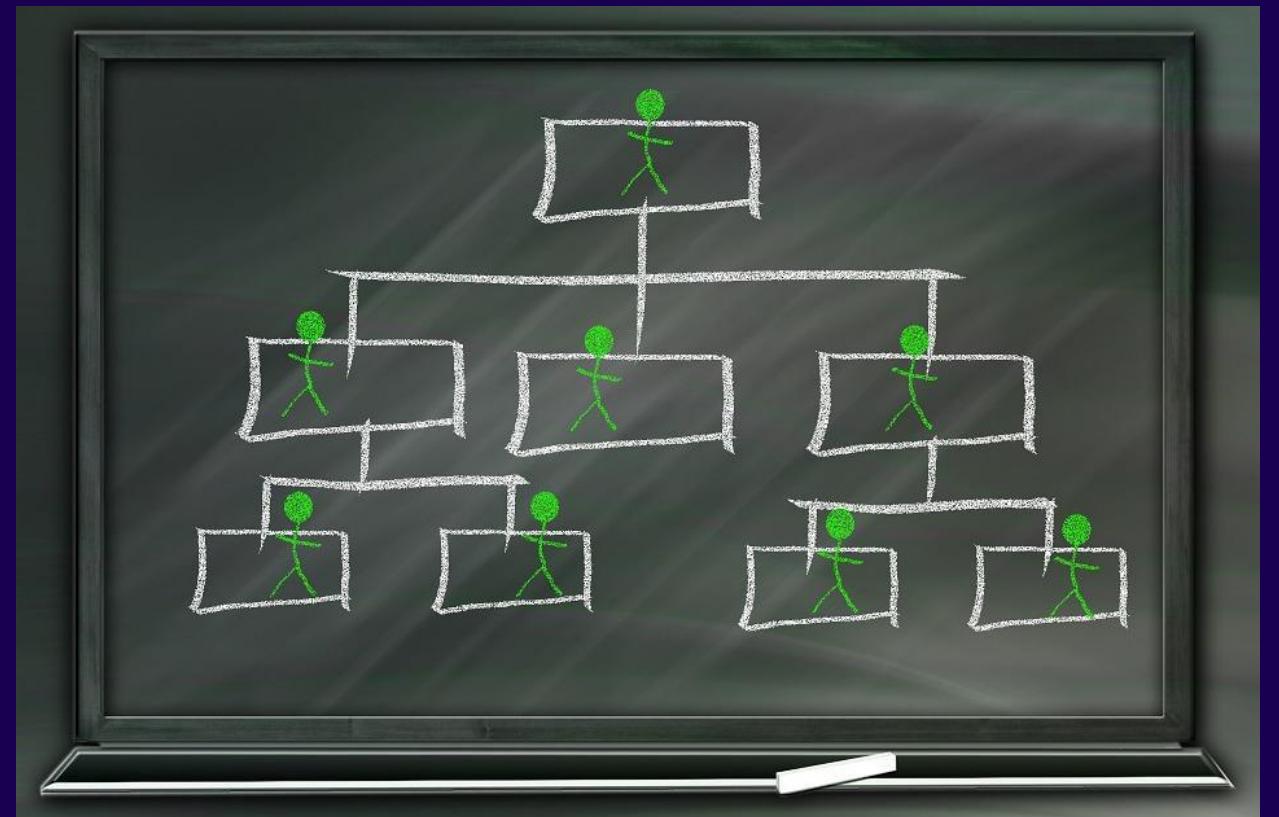


PrivilegedUser managed policy

```
{  
  "Effect": "Allow",  
  "Action": [  
    "acm-pca:IssueCertificate"  
,  
    "Resource": "arn:aws:acm-  
pca:*:*:certificate-authority/*",  
    "Condition": {  
      "StringLike": {  
        "acm-pca:TemplateArn": [  
          "arn:aws:acm-  
pca::::template/*CACertificate*/v*"  
        ]  
      }  
    }  
,  
  },  
},  
{  
  "Effect": "Deny",  
  "Action": [  
    "acm-pca:IssueCertificate"  
,  
    "Resource": "arn:aws:acm-  
pca:*:*:certificate-authority/*",  
    "Condition": {  
      "StringNotLike": {  
        "acm-pca:TemplateArn": [  
          "arn:aws:acm-  
pca::::template/*CACertificate*/v*"  
        ]  
      }  
    }  
,  
  },  
},
```

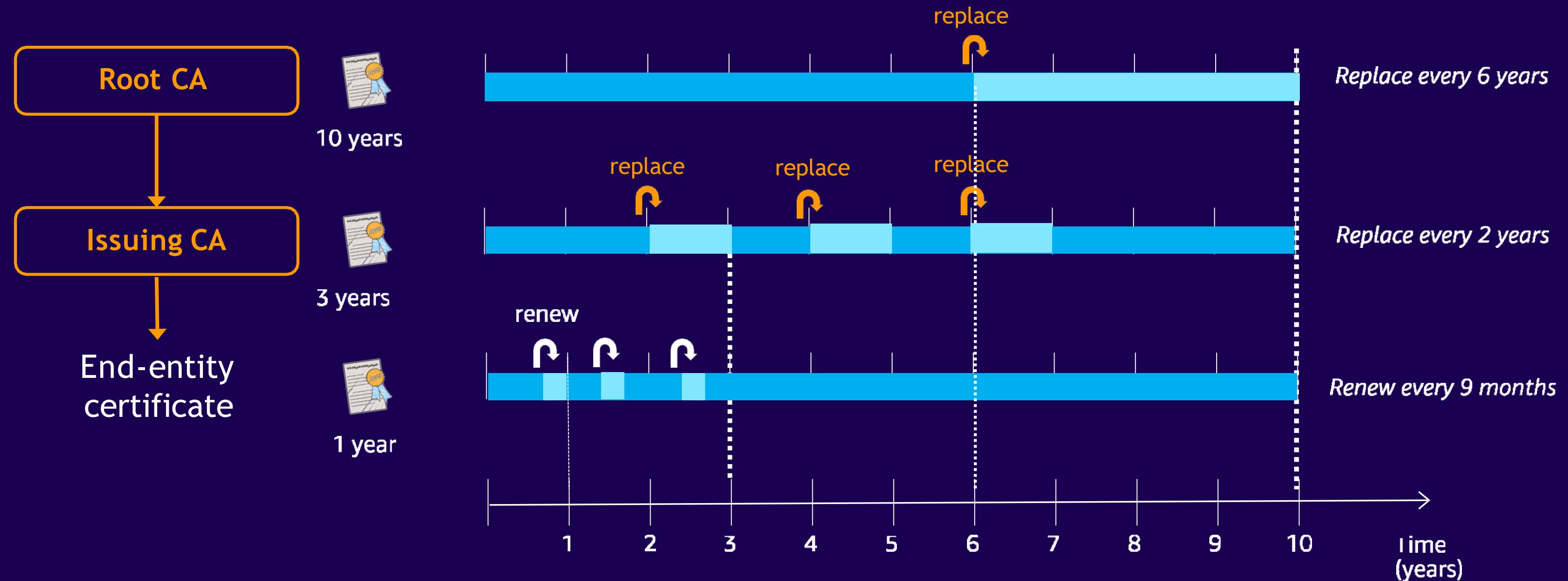
CA hierarchy

- Use path length constraint to limit CA height
- Reduce height when possible
 - Shorter chains reduce processing overhead
 - Use minimum tree height that meets your goals



Choosing CA validity period

Issuing CA validity period must be $>=$ lifetime of issued certificates



Operations

- Distributing root certificates/keys to trust stores
- Revocation and vending status information
- Redundancy/disaster recovery

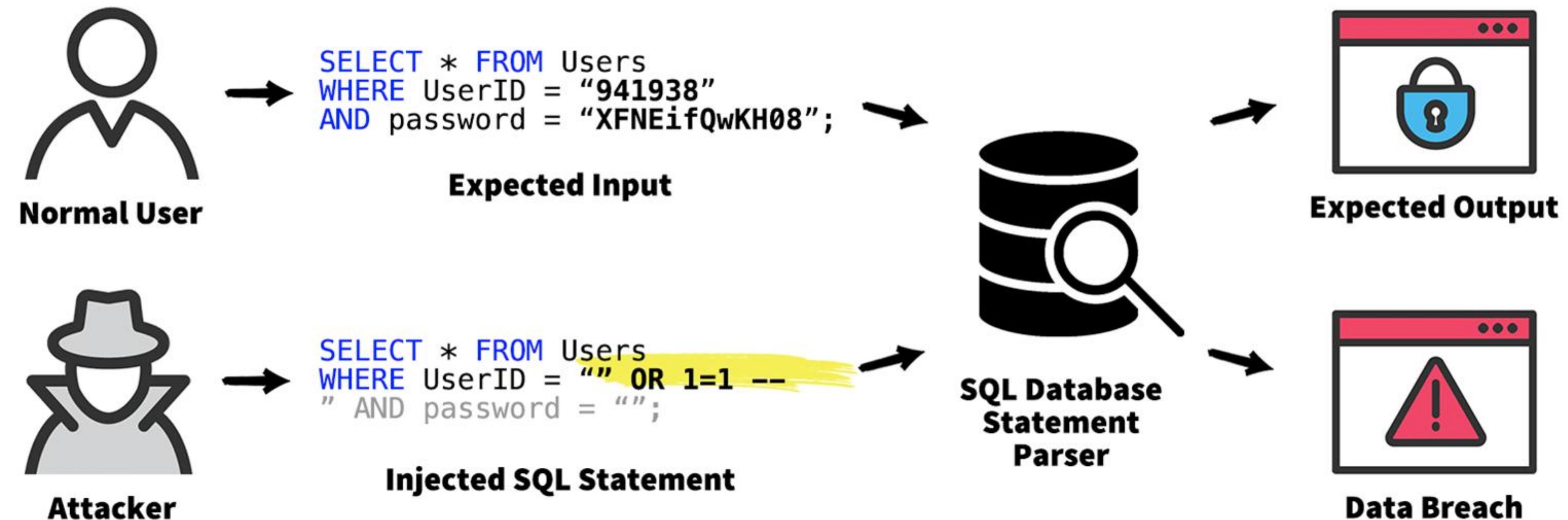
CA configuration

When you create a CA, think about

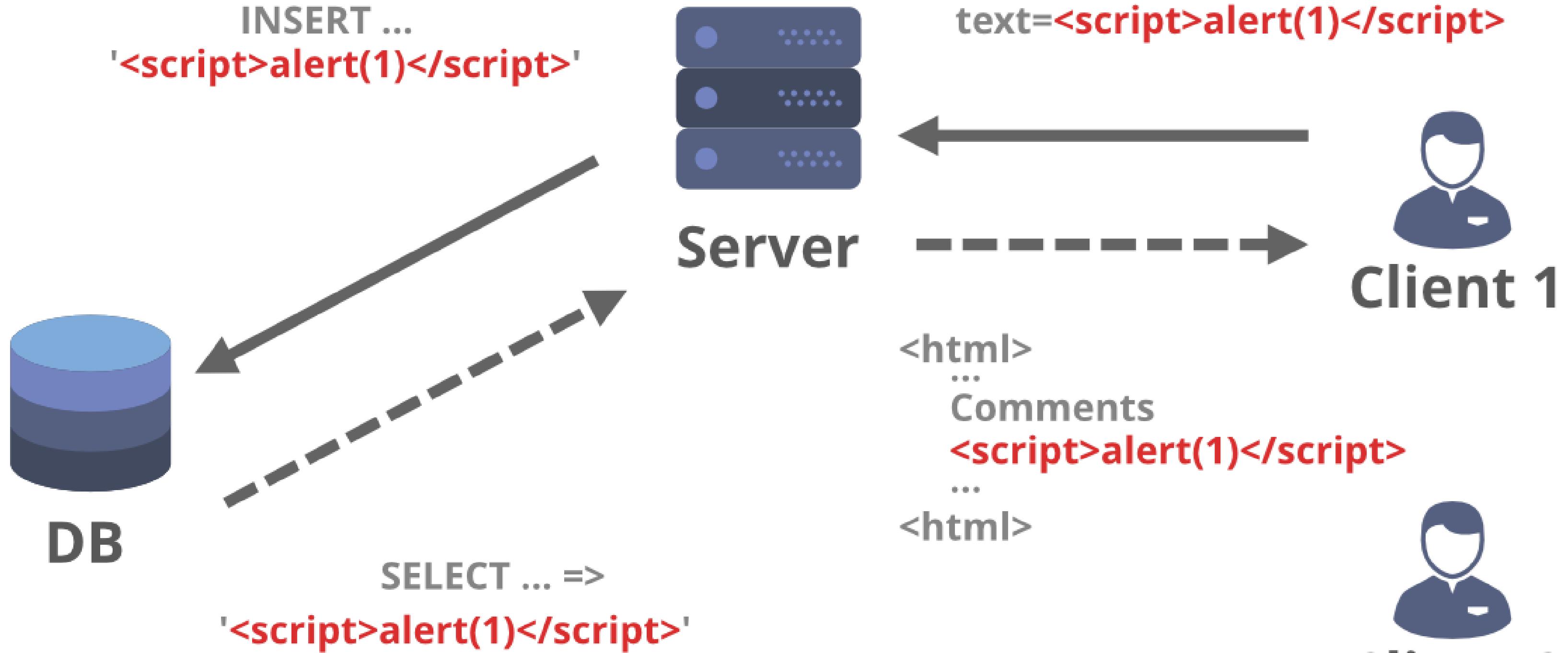
- Key types and sizes
RSA 2048, RSA 4096, ECDSA P256, ECDSA P384
- Revocation configuration
- AWS CloudTrail logging of API calls
- Amazon CloudWatch metrics - alarms and notifications
- Audit reporting
- Access policies
- CA lifecycle management



What is SQL Injection?

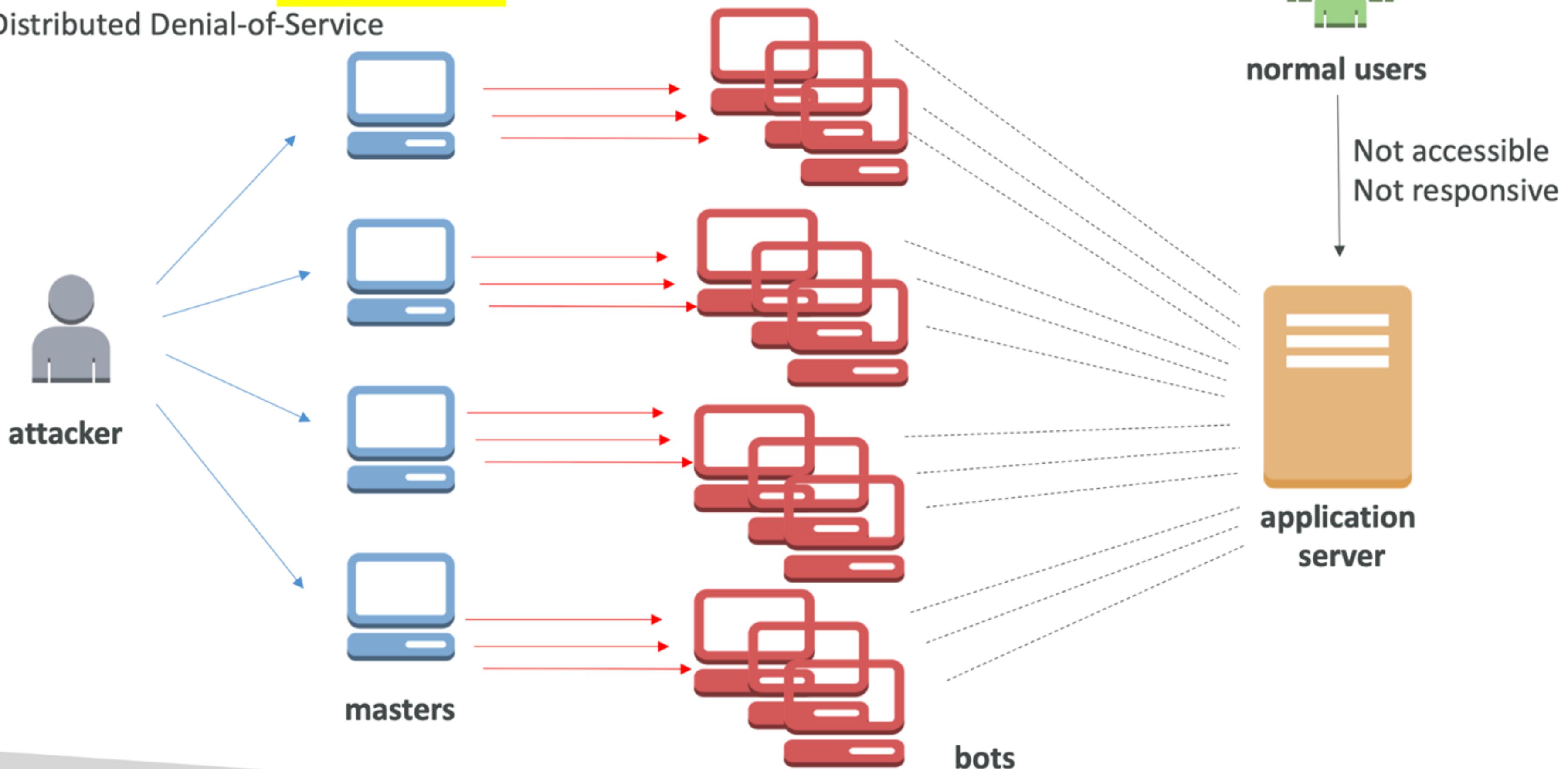


Cross Site Scripting(XSS)



What's a **DDOS*** Attack?

*Distributed Denial-of-Service



AWS Web Application Firewall (WAF)

- AWS WAF protects applications from SQL Injection, XSS, and DDoS attacks.
- Works with ALB, API Gateway, and CloudFront.
- Uses WebACLs (Web Access Control Lists) for filtering requests.
- Can block, allow, or monitor requests based on rules.

How AWS WAF Works

- WebACL Rules define what traffic is allowed or blocked.
- Managed Rules help block common threats.
- Rate-based rules protect against brute force and DDoS attacks.
- Works with Shield for enhanced DDoS protection.

AWS Shield for DDoS Protection

- AWS Shield provides DDoS protection for applications.
- Two types:
 1. **AWS Shield Standard** (free, automatic protection)
 2. **AWS Shield Advanced** (paid, 24/7 response, insurance)
- Integrated with CloudFront, Route 53, and WAF.

AWS Security Best Practices

- Enable AWS Shield and WAF for web applications.
- Use KMS for encrypting sensitive data.
- Implement IAM policies with least privilege access.
- Monitor security events using CloudWatch and CloudTrail.
- Regularly update WAF rules to prevent new threats.

Knowledge check

Which of the following are components of IAM?

- A. Group - collection of users with identical permissions
- B. Bucket - container for stored objects
- C. User - person or application that interacts with AWS
- D. Instance - copy of an AMI running as a virtual server
- E. Policy - formal statement of one or more permissions

Knowledge check

Which of the following are components of IAM?

- A. Group - collection of users with identical permissions
- B. Bucket - container for stored objects
- C. User - person or application that interacts with AWS
- D. Instance - copy of an AMI running as a virtual server
- E. Policy - formal statement of one or more permissions

Answer: A, C, E

Key Takeaways

- Security is EVERYONE'S responsibility
 - Security IN the Cloud / Security OF the Cloud
- IAM allows users to control access to AWS resources
 - Apply policies to Users, Groups & Roles
- When a S3 bucket is created, by default it is set to PRIVATE
- CloudTrail records API calls in AWS
 - Who, What, When, Where
- AWS Trusted Advisor provides recommendations
 - help you reduce cost, increase performance and improve security

Section 2 - AWS CI/CD and Dev. Tools

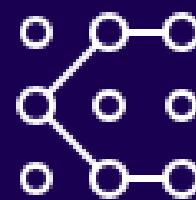
Let's build together



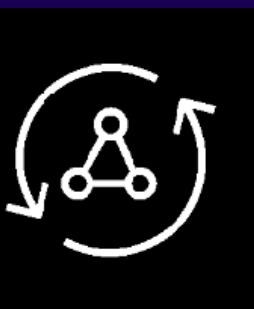
Amazon CodeCatalyst | codecatalyst.aws



Amazon CodeWhisperer | aws.amazon.com/codewhisperer



AWS Application Composer | aws.amazon.com/application-composer



AWS AppSync | aws.amazon.com/appsync

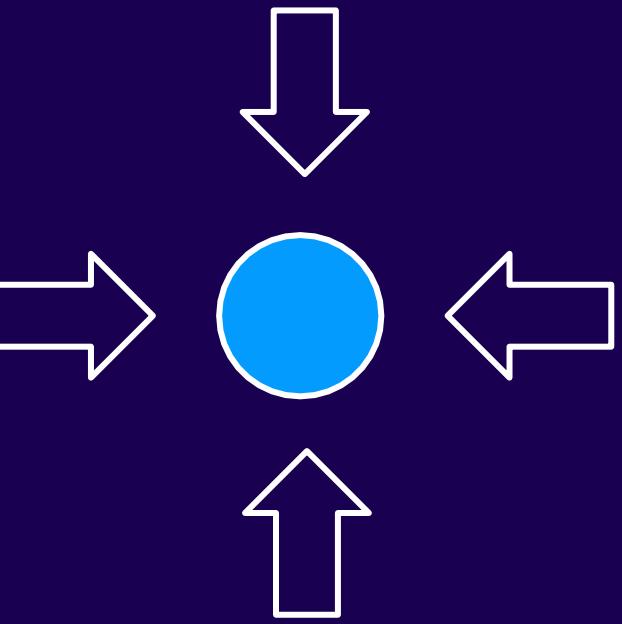


AWS Amplify | aws.amazon.com/amplify



Amazon CodeCatalyst

Amazon CodeCatalyst



Integrated experience

Unified interface that eliminates the need
to configure, operate, integrate, or switch
between disjointed tools

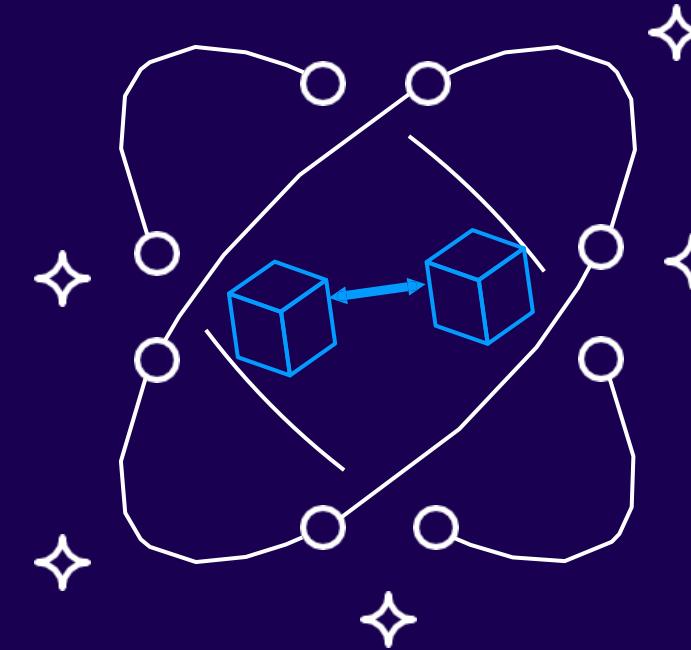
Amazon CodeCatalyst



Managed by AWS

Managed, scalable service operated by AWS
for high availability and security

Amazon CodeCatalyst

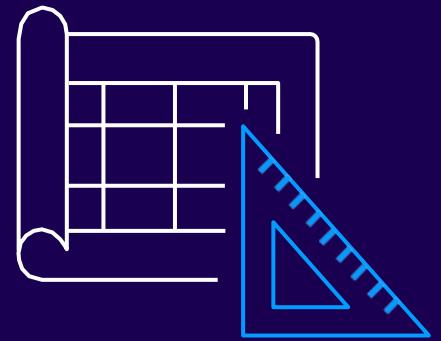


Integrates with popular tools

Continue working with popular tools like Jira and GitHub while maintaining a consistent software lifecycle experience

Amazon CodeCatalyst

KEY FEATURES



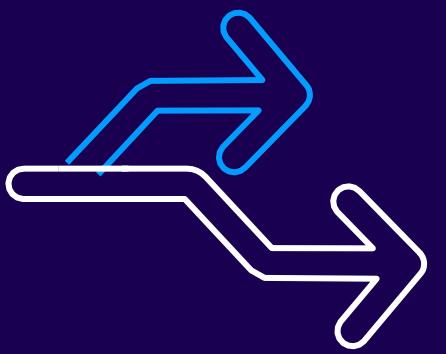
Project blueprints

Amazon CodeCatalyst

KEY FEATURES



Project blueprints



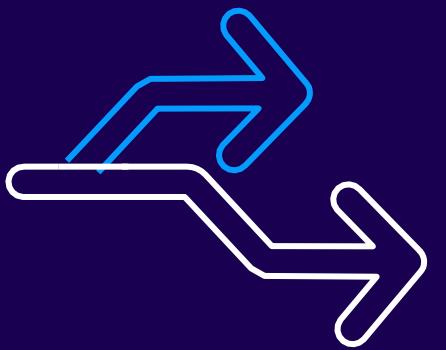
Managed CI/CD pipelines

Amazon CodeCatalyst

KEY FEATURES



Project blueprints



Managed CI/CD pipelines



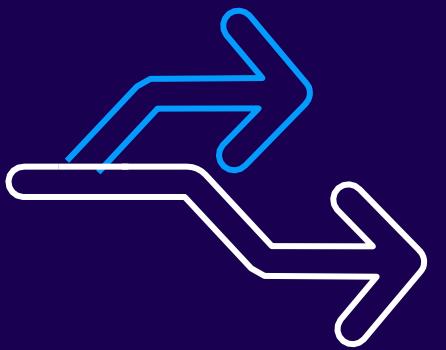
Dev environments

Amazon CodeCatalyst

KEY FEATURES



Project blueprints



Managed CI/CD pipelines



Dev environments

codecatalyst.aws

codecatalyst.aws/spaces/Initech/projects

Amazon CodeCatalyst Initech Projects Search code, issues, projects, and users Create Space Create Project

Initech

Projects Activity Members Installed extensions AWS accounts Space settings Billing

Projects (7)

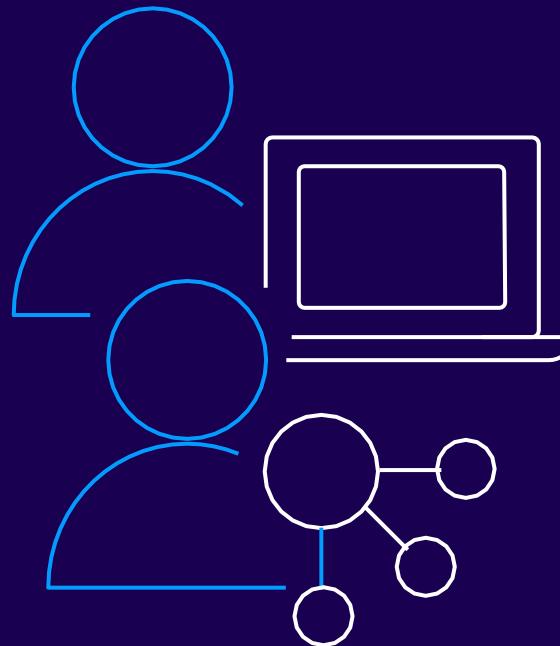
Your projects View settings

< 1 > |

Project	Last modified	Pull requests	Source repositories	Workflows	Dev Environments	Reports	Environments	Issues
shop-web-stack	20 hours ago	Pull requests	Source repositories	Workflows	Dev Environments	Reports	Environments	Issues
application-frontend	November 21, 2022 at 09:22 PM	Pull requests	Source repositories	Workflows	Dev Environments	Reports	Environments	Issues
dotnet-invite-service	November 21, 2022 at 09:36 PM	Pull requests	Source repositories	Workflows	Dev Environments	Reports	Environments	Issues
deploy-pipeline	20 hours ago	Pull requests	Source repositories	Workflows	Dev Environments	Reports	Environments	Issues
java-poller-service	November 22, 2022 at 09:55 AM	Pull requests	Source repositories	Workflows	Dev Environments	Reports	Environments	Issues
customer-portal	November 22, 2022 at 10:28 AM	Pull requests	Source repositories	Workflows	Dev Environments	Reports	Environments	Issues

© 2008 - 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved. [Cookie Preferences](#) [Privacy](#) [Site Terms](#)

Amazon CodeCatalyst



Collaborate fluidly

Easily invite a teammate to your project and provide immediate access to shared resources for efficient collaboration

codecatalyst.aws

codecatalyst.aws/spaces/Initech/projects/application-frontend/view

Amazon CodeCatalyst Initech application... Search code, issues, projects, and users

application-frontend... < Initech > application-frontend > overview

application-frontend summary

Members +

KS kk AS EF AS View all

Overview Issues Code CI/CD Reports Project settings

Repositories

View repository

spa-app

main / README.md

Project overview

This project creates a [React SPA](#) (single-page application) project as the front-end. The project uses the [TypeScript AWS Cloud Development Kit \(CDK\)](#) to deploy to [AWS Amplify Hosting](#).

Architecture overview

The front-end is a single-page application and loads a web document and updates it by using JavaScript APIs. The deployment pipeline deploys the SPA front-end to the development environment by default, using the connected AWS account.

Web application framework

[React](#) - powered by [Create React App](#)

Hosting

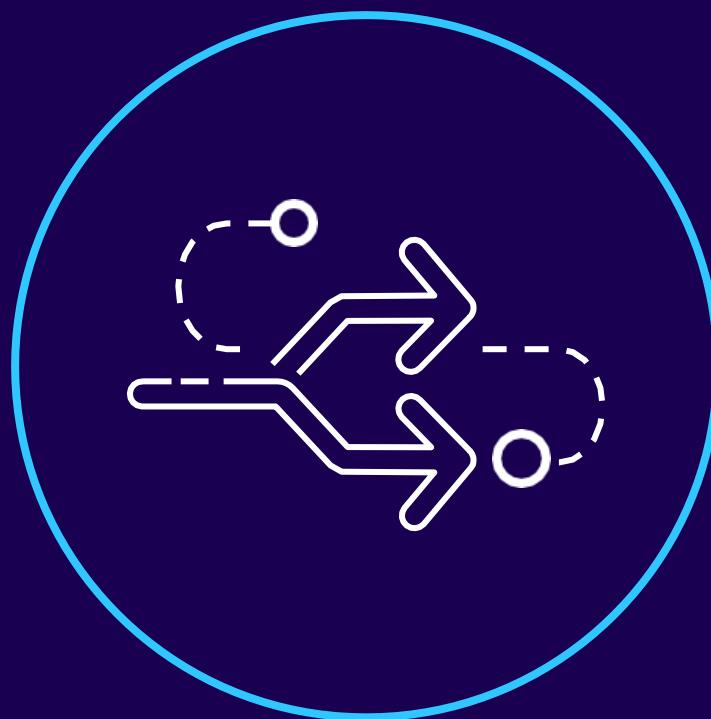
AWS Amplify Hosting

[AWS Amplify Hosting](#) offers a fully managed hosting service for web apps and static websites that can be accessed directly from the AWS console.

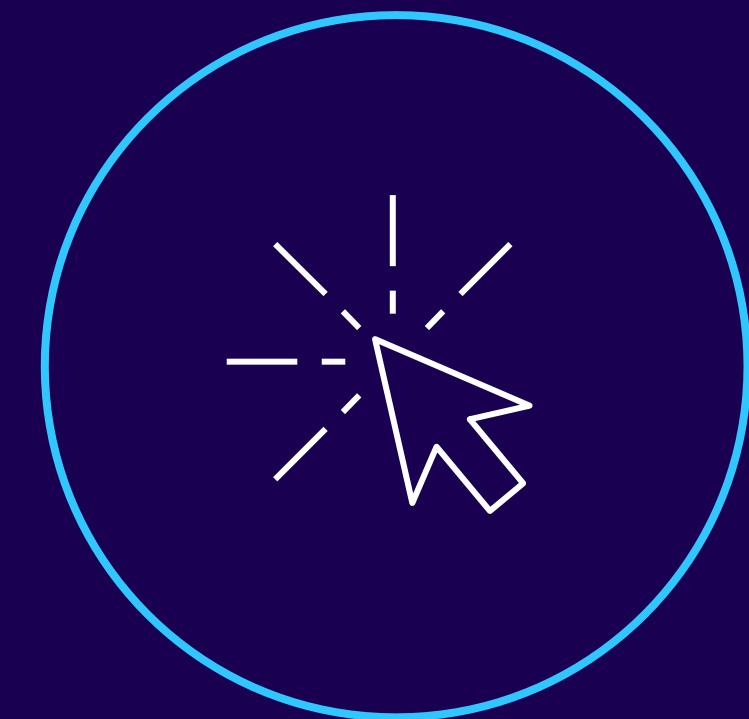
© 2008 - 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved. [Cookie Preferences](#) [Privacy](#) [Site Terms](#)

Get started with CodeCatalyst

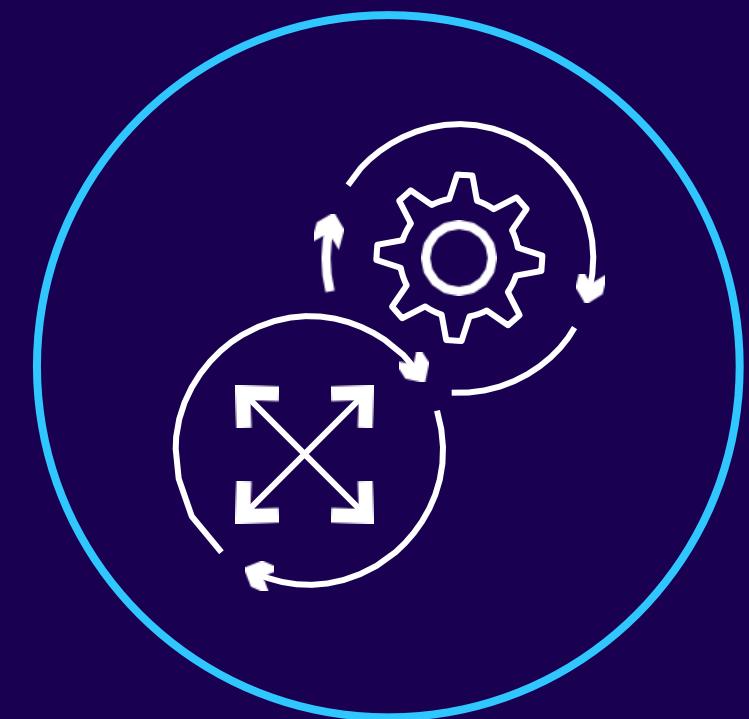
VISIT CODECATALYST.AWS



Accelerate
project setup



Automate daily
workflows



Initiate cloud
environments



Collaborate
fluidly

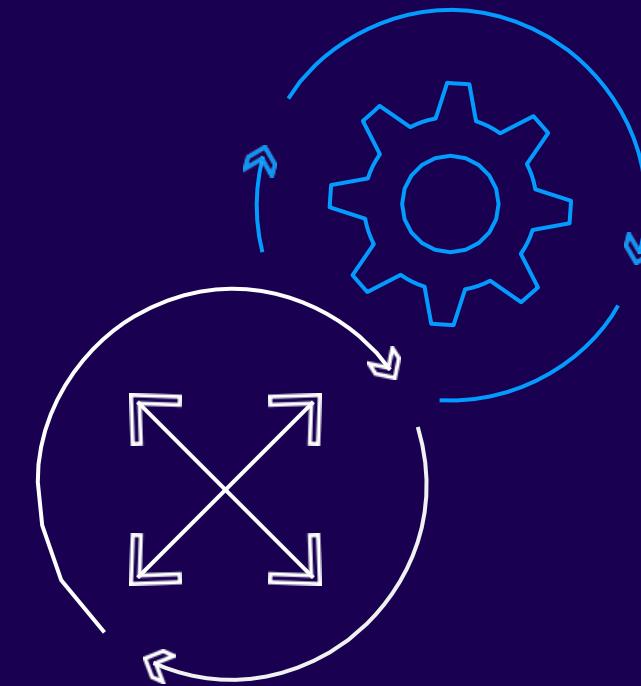
Amazon CodeWhisperer

AVAILABLE IN PYTHON, JAVA, JAVASCRIPT, TYPESCRIPT, AND C#

ALSO SUPPORTS

GO, RUST, PHP, RUBY, KOTLIN, C, C++, SHELL SCRIPTING, SQL, SCALA

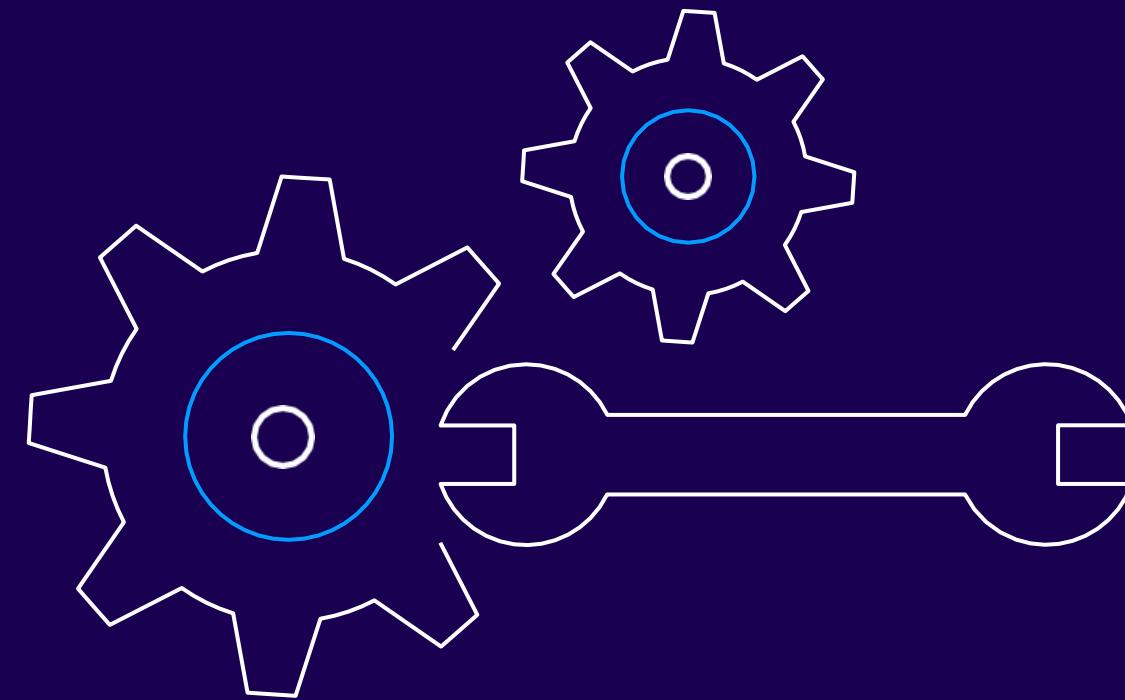
Amazon CodeWhisperer



Accelerate developer productivity

Support for the most popular programming
languages and IDEs

Amazon CodeWhisperer



Unmatched support for AWS APIs

Speed up development process with generated
code recommendations for AWS APIs

```
4 import com.amazonaws.services.s3.AmazonS3ClientBuilder;
5 import com.amazonaws.services.s3.model.AccessControlList;
6 import com.amazonaws.services.s3.model.EmailAddressGrantee;
7
8 import java.util.List;
9
10 public class S3Util {
11
12
13 }
```

The screenshot shows a Java code editor with the file `S3Util.java` open. The code defines a class `S3Util` with imports for various AWS SDK components and a single-line body. Below the code editor is the `AWS Toolkit` extension interface, which includes a status bar, tabs for `Explorer` and `Developer Tools`, and a section for `CodeWhisperer (Preview)` with options like `Resume Auto-Suggestions`, `Run Security Scan`, and `Open Code Reference Log`. The `Open Code Reference Log` button is highlighted with a blue background. At the bottom, a `CodeWhisperer Reference Log` panel displays a message about settings. The overall interface is dark-themed.

```
1 import software.amazon.awssdk.auth.credentials.ProfileCredentialsProvider;
2 import software.amazon.awssdk.regions.Region;
3 import software.amazon.awssdk.services.s3.S3Client;
4 import software.amazon.awssdk.services.s3.model.CopyObjectRequest;
5 import software.amazon.awssdk.services.s3.model.CopyObjectResponse;
6 import software.amazon.awssdk.services.s3.model.S3Exception;
7
8 import java.util.List;
9
10 public class S3Util {
11
12
13
14 }
```

AWS Toolkit

Connected with AWS Builder ID ... ?

Explorer Developer Tools

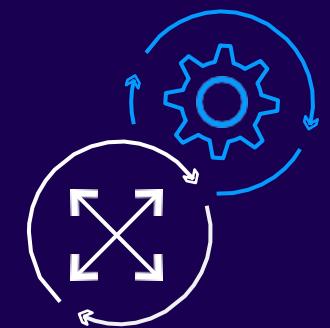
CodeWhisperer (Preview)

- ▶ Resume Auto-Suggestions
- ▶ Run Security Scan
- Open Code Reference Log

CodeWhisperer Reference Log

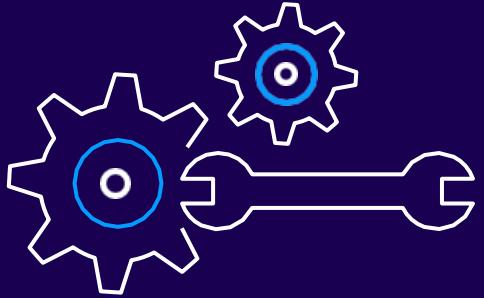
Don't want suggestions that include code with references? Edit in [CodeWhisperer Settings](#).

Amazon CodeWhisperer features



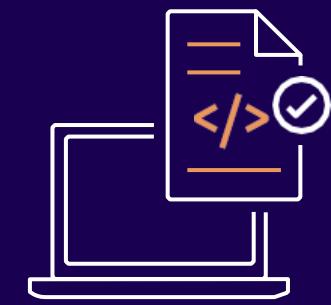
Accelerate developer productivity

Support for the most popular programming languages and IDEs



Unmatched support for AWS APIs

Speed up development process with generated code recommendations for AWS APIs

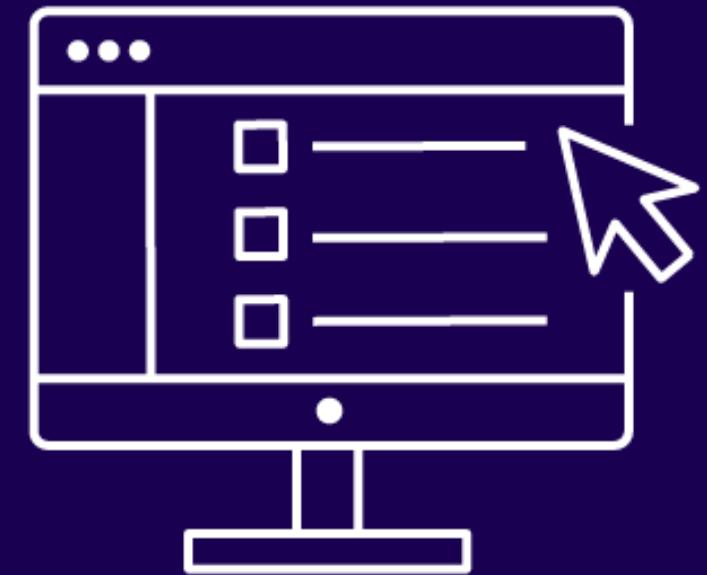


Responsible use of AI/ML

Built-in security scans, code reference tracker, and bias avoidance

AWS Application Composer

AWS Application Composer



Visual canvas

Drag, drop, and connect serverless
resources to simplify designing applications

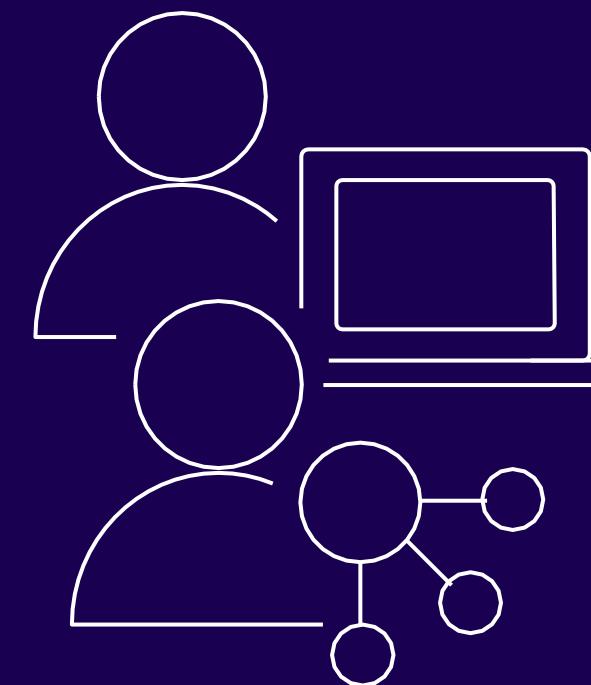
AWS Application Composer



Configures services automatically

Rapidly generate resource configuration
and ready-to-deploy infrastructure as code
(IaC) that follows best practices by default

AWS Application Composer



Architecture in real time

Keeps visual architectural representation and IaC in sync

learning-serverless

EXPLORER

LEARNING-SERVERLESS

Show All Commands ⌘ P

Go to File ⌘ E ⌘ E

Find in Files ⌘ F

Start Debugging F5

Toggle Terminal ⌘ ⌘

AWS Application Composer

Unconnected mode

List Resources

Search for a resource

API Gateway

Cognito UserPool

Cognito UserPoolClient

DynamoDB Table

EventBridge Event rule

EventBridge Schedule

Kinesis Stream

Lambda Function

Lambda Layer

S3 Bucket

SNS Topic

SQS Queue

Step Functions State machine

Canvas Template Arrange

Take a quick tour of composer Start

Create project

Type of project

Get started with a blank project or upload an existing one. Alternatively, get started with a sandbox project.

New blank project Creates a new CloudFormation template.

Load existing project Load an existing CloudFormation template.

New blank project

Creates a new project template with no resources. You can drag and drop your resources or paste YAML into the code view to begin building. To use a local file system connection, you must grant browser permissions.

Local file system connection

Use a local file system connection to automatically save your files as you work. If turned off, you can edit on a single template file.

On

Off

Project location

Select an empty folder where you want your project files to be saved.

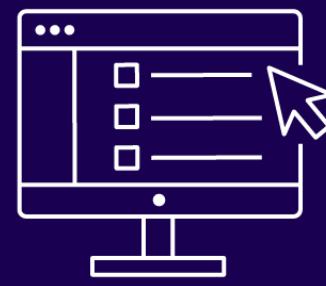
Select folder

Cancel Create

Feedback Looking for language selection? Find it in the new Unified Settings

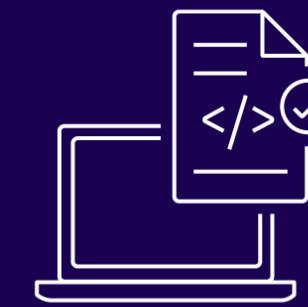
© 2022, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

AWS Application Composer



Visual canvas

Drag, drop, and connect serverless resources to simplify designing applications



Configures services automatically

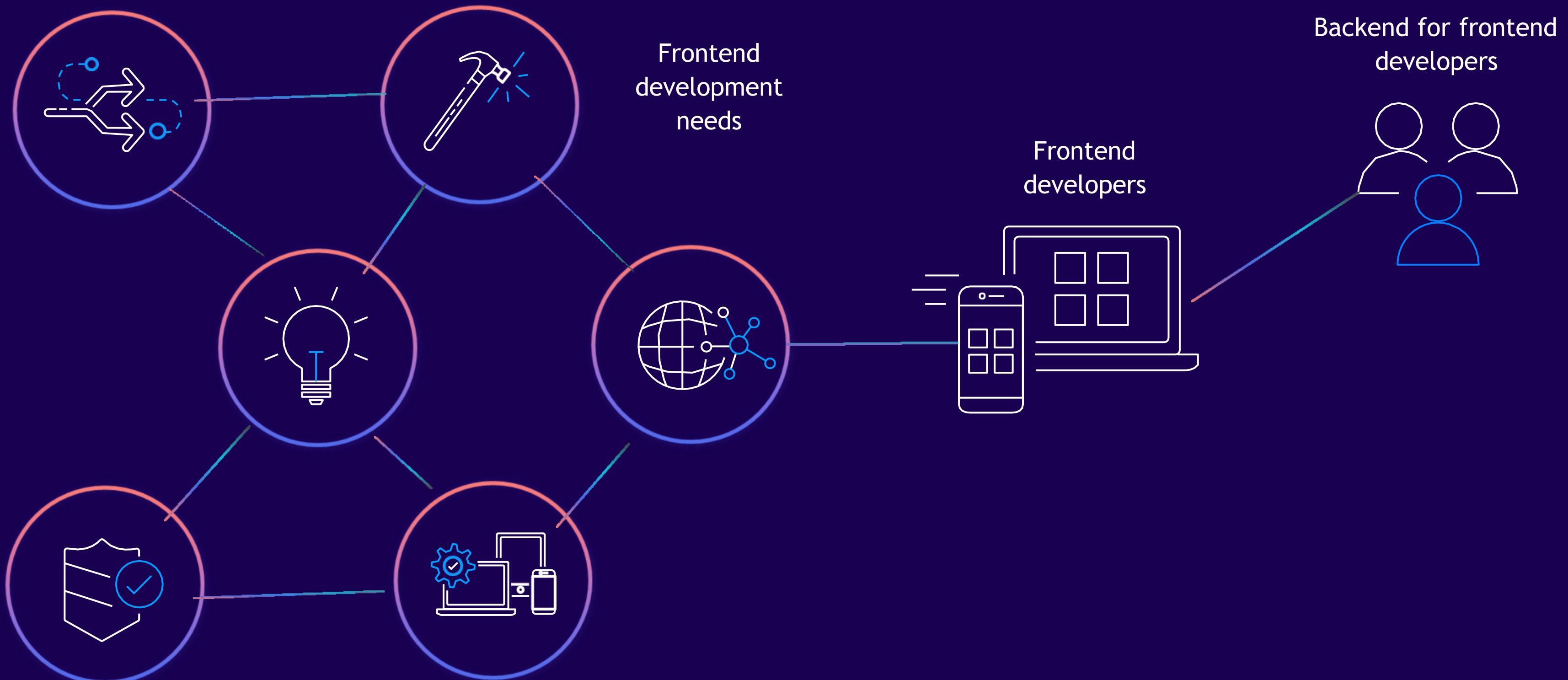
Rapidly generate resource configuration and ready-to-deploy infrastructure as code (IaC) that follows best practices by default



Architecture in real time

Visualize your architecture in real time to keep your team in sync

Frontend development

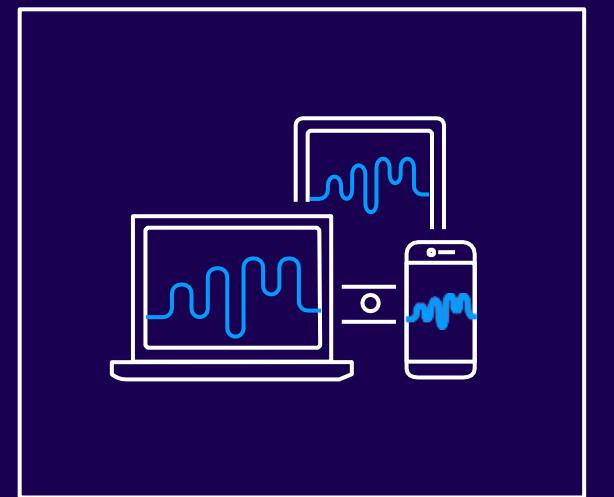


What is GraphQL?

SIMPLIFIED API STRUCTURES

Frontend

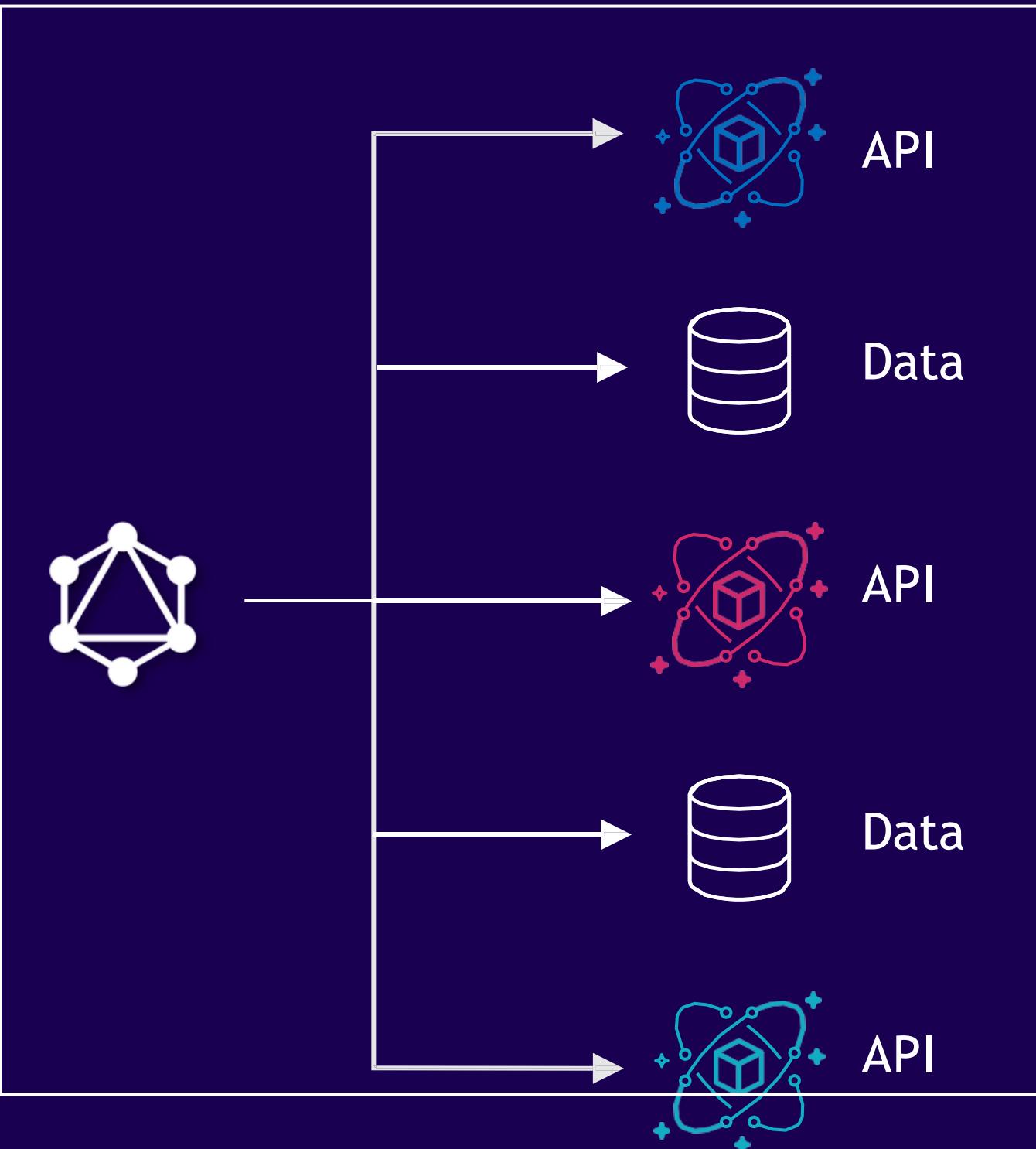
Unified to access to fetch
only the data you need
through a single API call



/graphql

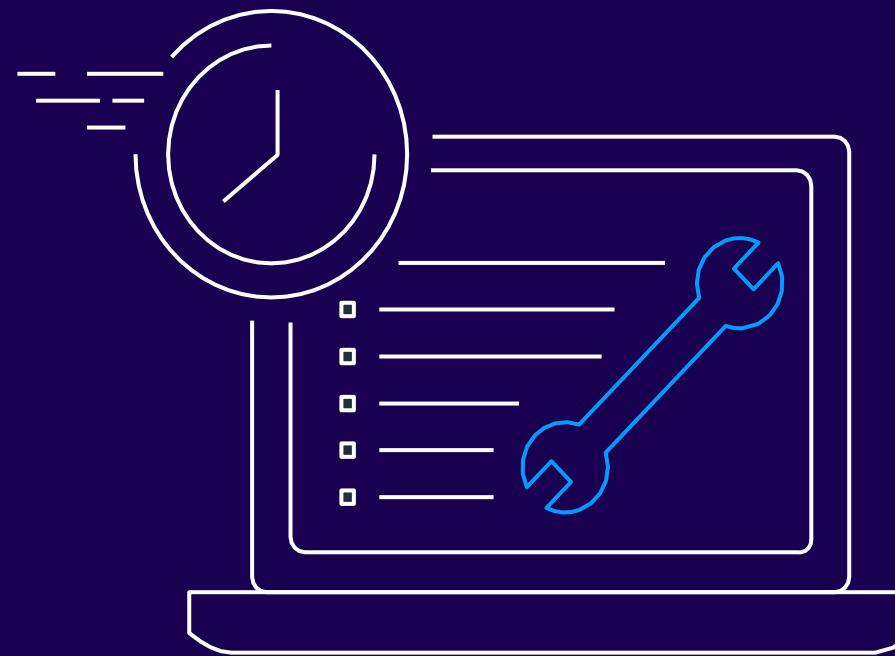
{JSON}
(client defined)

Backend for frontend



AWS AppSync

FULLY MANAGED SERVERLESS GRAPHQL AND PUB/SUB API SERVICE

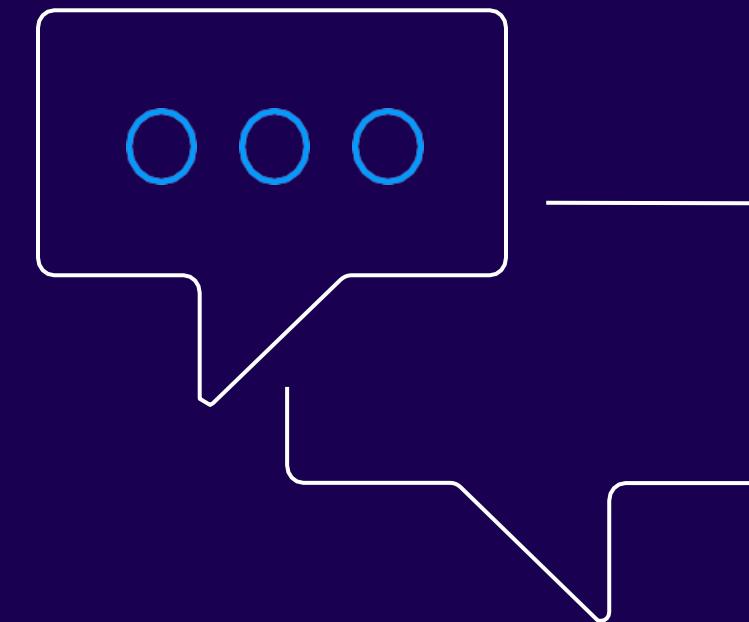


A single data API

Access data from one or more sources
or microservices with a single
network request

AWS AppSync

FULLY MANAGED SERVERLESS GRAPHQL AND PUB/SUB API SERVICE

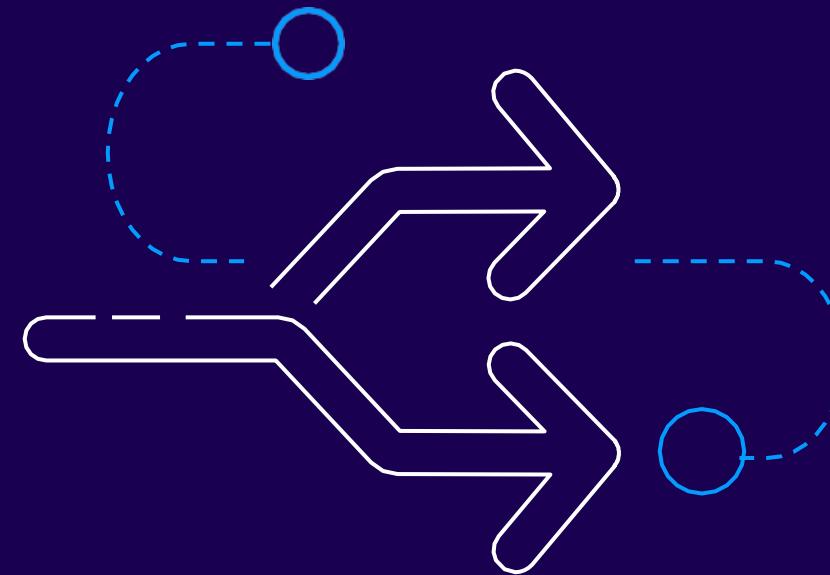


Real-time and offline

Build engaging real-time features;
automatically sync data between apps
and the cloud

AWS AppSync

FULLY MANAGED SERVERLESS GRAPHQL AND PUB/SUB API SERVICE

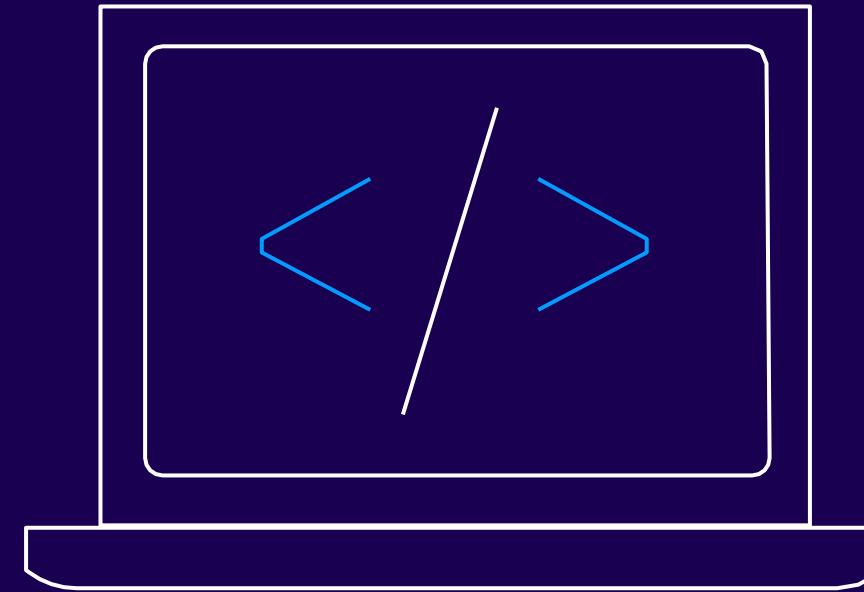


Serverless

Simplify operations with fully managed setup, administration, auto-scaling, and high availability

AWS AppSync

FULLY MANAGED SERVERLESS GRAPHQL AND PUB/SUB API SERVICE

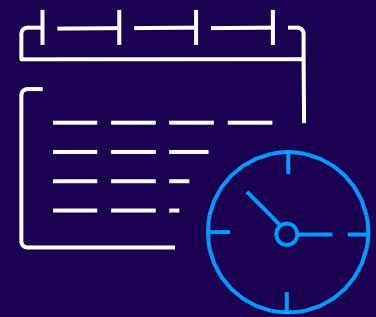


JavaScript support

Use JavaScript to connect AppSync
GraphQL and Pub/Sub APIs to data

AppSync benefits for teams

ACCELERATE DEVELOPMENT WITH SIMPLIFIED DATA ACCESS



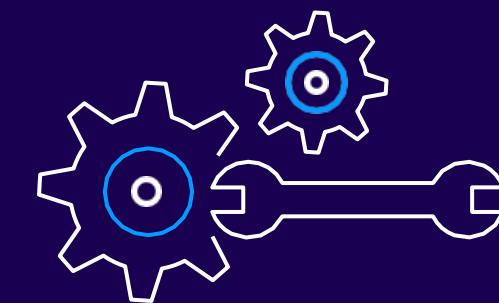
Create apps faster

Backend and frontend teams can work faster and independently



Build performant apps

Reduce network requests by fetching only the data you need



Built-in security

Utilize auth, encryption, activity logging, and other AWS services for compliance

AWS Amplify

SET OF TOOLS AND SERVICES FOR EASILY BUILDING, DEPLOYING, AND HOSTING FULL-STACK APPS FASTER



Host a web app

Amplify hosting



Build AWS backend

Amplify Studio
Amplify CLI



Build a web app UI

Amplify Studio



Connect to backend

Amplify Libraries
UI components

Amplify Hosting for Next.js

FULLY MANAGED HOSTING FOR SERVER SIDE RENDERED APPS BUILT WITH NEXT.JS 13



- Server-side rendering
- Zero-config development
- Integration with AWS services
- Fully managed infrastructure

Amplify Hosting for Next.js

FULLY MANAGED HOSTING FOR SERVER SIDE RENDERED APPS BUILT WITH NEXT.JS 13



- Server-side rendering
- Zero-config development
- Integration with AWS services
- Fully managed infrastructure



← →

next-amplify



EXPLORER

NEXT-AMPLIFY

- > .next
- > .vscode
- > components
- > graphql
- > node_modules
- > pages
 - > api
 - > details
 - JS** _app.js
 - JS** index.js
- > public
 - ★ favicon.ico
 - ❑ vercel.svg



...



Show All Commands

Go to File

Find in Files

Start Debugging

Toggle Terminal



next-amplify



EXPLORER



NEXT-AMPLIFY



.next



.vscode



amplify



components



Card.js



CardList.js



graphql



node_modules



pages



api



details



_app.js



index.js



JS index.js M X

pages > JS index.js > ...

```
1 import Head from 'next/head'
1 import styles from '../styles/Home.module.css'
2
3 export default function Home() {
4   return (
5     <div className={styles.container}>
6       <Head>
7         <title>Create Next App</title>
8         <meta name="description" content="Generated by create next app" />
9         <link rel="icon" href="/favicon.ico" />
10    </Head>
11
12    <main className={styles.main}>
13      <h1 className={styles.title}>
14        Welcome to <a href="https://nextjs.org">Next.js!</a>
15      </h1>
16
17      <p className={styles.description}>
18        Get started by editing{' '}
19        <code className={styles.code}>pages/index.js</code>
20      </p>
21
22      <div className={styles.grid}>
23        <a href="https://nextjs.org/docs" className={styles.card}>
24          <h2>Documentation &rarr;</h2>
25          <p>Find in-depth information about Next.js features and API
26        </a>
27      </div>
```



OUTLINE

TIMELINE

main*



0 △ 0



Live Share Already at oldest change



Ln 1, Col 13 Spaces: 2

UTF-8 LF

{} JavaScript



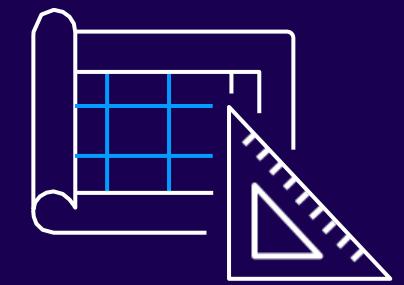
Amplify Studio form builder

BUILD, DESIGN, AND RENDER CLOUD-CONNECTED FORMS FOR ANY API WITH NEW REACT FORM BUILDER



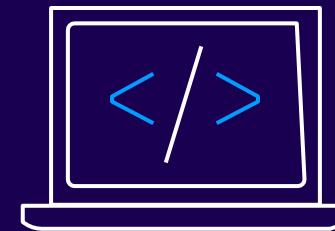
Forms for any API

Generate forms from an API definition or
create new forms from scratch



Design visually

Save time by visually configuring
common validation rules



Extend via code

Customize validation rules and
form styling in code

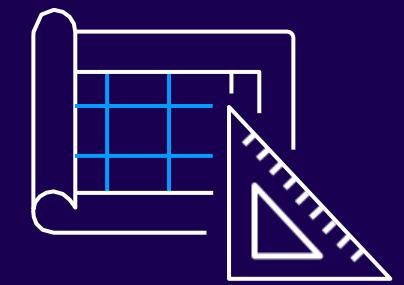
Amplify Studio form builder

BUILD, DESIGN, AND RENDER CLOUD-CONNECTED FORMS FOR ANY API WITH NEW REACT FORM BUILDER



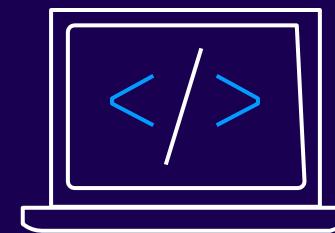
Forms for any API

Generate forms from an API definition or
create new forms from scratch



Design visually

Save time by visually configuring
common validation rules



Extend via code

Customize validation rules and
form styling in code

localhost:3000/details/photos X | AWS Amplify console X | https://main.dqsp8bef0b9ou.a... X | createnextapp - dev - Amplify X | Amplify Studio X +

sandbox.amplifyapp.com/ui-library/form/create

Amplify Studio

Share  Getting started progress 0% aws Unlock all of Amplify Deploy to AWS

Home Try without AWS account Data UI library

Sign in to AWS to use Host my app Authentication File storage Functions GraphQL API Analytics Predictions Interactions Notifications

Build React forms with Amplify

Quickly build React forms that are cloud-enabled, fully-customizable, and ready to deploy in your app.

 Blank form  JSObject

Create form

 **Configure form inputs**
Select form inputs to edit, remove, or add in to your existing form

 **Customize display**
Click on any input to edit label, placeholder, or description text

 **Add validation**
Add custom validation rules and error messages to any input

Documentation Support AWS Amplify Studio is supported by Amazon Web Services © 2022, Amazon Web Services, Inc. and its affiliates. All rights reserved. View the site terms and privacy policy .

AWS TOOLBOX

...

Quick Start X

> AWS SERVICE LIST

> REGION

AWS PROFILE

default

codium

AWS Toolbox

Open-Source Real-time AWS Resource Monitoring

AWS Toolbox is an innovative, open-source Visual Studio Code extension designed to enable developers and cloud engineers to manage and interact with AWS services directly within their editor. It offers a streamlined workflow for AWS resource management, making it an indispensable tool for enhancing AWS operations within VS Code.

STEP 1

Select AWS services to monitor

Select All (0)

CostExplorer



VPC



S3



EC2



Lambda



ECS



ECR



RDS



DynamoDB



Redshift



IAM



EventBridge

File Edit Selection View Go ⌄ < > materials 🔍

AWS TOOLBOX ... Quick Start CostExplorer X

stop

AWS SERVICE LIST

- CostExplorer
- VPC
- S3
- Lambda
- EC2
- ECS
- ECR
- RDS
- DynamoDB
- Redshift
- IAM
- EventBridge

August

\$0.07

August

\$0.07

2025-08-01 → 2025-08-21

Total: days

Daily Cost:

August Estimated

\$0.10

2025-08-01 → 2025-08-31

Total: days

Daily Cost:

July

\$1.18

July

\$1.18

2025-07-01 → 2025-07-31

Total: days

Daily Cost:

August / July

Estimated
\$0.10 / \$1.18 = +8%
\$0.10 - \$1.18 = -\$1.08

Current
\$0.07 / \$1.18 = +6%
\$0.07 - \$1.18 = -\$1.11

June

\$1.19

June

\$1.19

2025-06-01 → 2025-06-30

Total: days

Daily Cost:

August / June

Estimated
\$0.10 / \$1.19 = +8%
\$0.10 - \$1.19 = -\$1.09

Current
\$0.07 / \$1.19 = +6%
\$0.07 - \$1.19 = -\$1.12

Range: 2025-06-01 - 2025-08-21

Key Management Service 2.04

master 0 0 AWS: profile:default Amazon Q

Signed out



AWS ...

✓ EXPLORER ...

🔑 Connected with profile:d...

- ✓ Asia Pacific (Hyderabad)
 - > API Gateway
 - > CloudFormation
 - > CloudWatch Logs
 - > EC2
 - > ECR
 - > ECS
 - > Lambda
 - > Redshift
 - > S3
 - > SageMaker AI
 - > Step Functions
 - > Systems Manager
 - > Resources
- > Asia Pacific (Mumbai)
- > Asia Pacific (Singapore)

- > CDK
- > APPLICATION BUILDER
- > CODECATALYST



master 0 0 ✓ AWS: profile:default

▶ Amazon Q

Show All Commands + + P

Go to File + P

Open Chat + Alt + I

Toggle Terminal + `

Find in Files + Shift + F

Signed out



AWS SDKs by Programming Languages

C++

.NET

Go

Java

JavaScript

Kotlin

Node.js

PHP

Python

Ruby

Rust

SAP ABAP

Swift

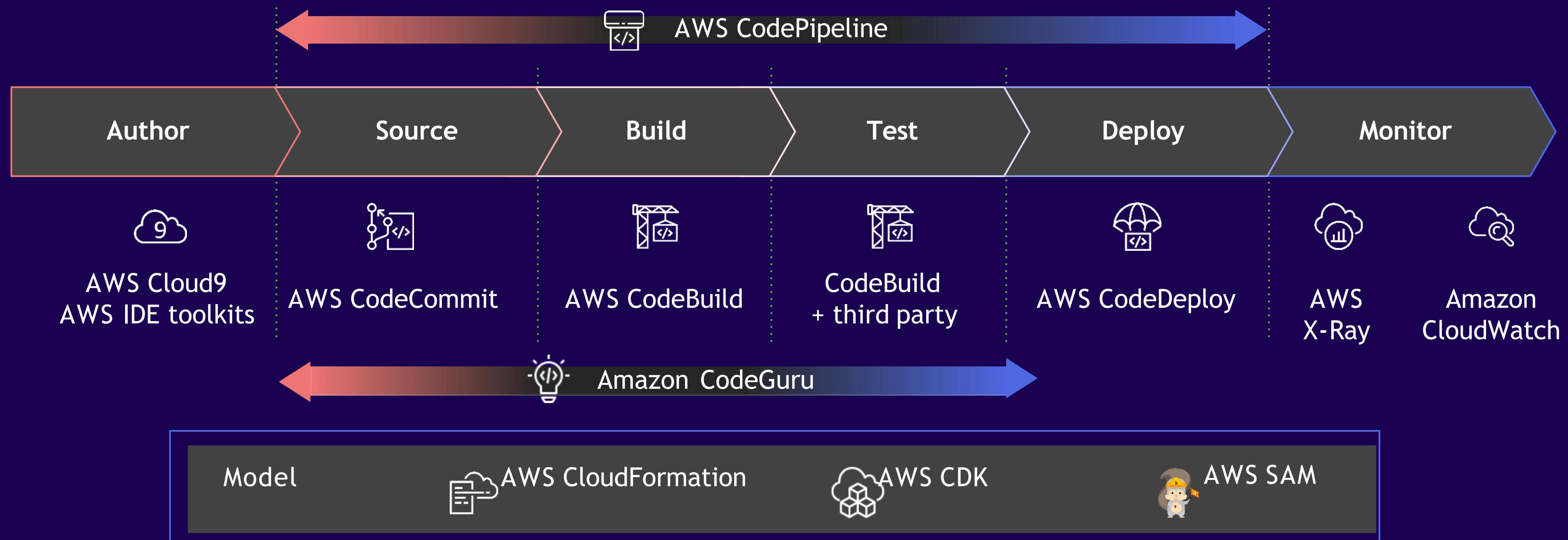
AWS CLIs

```
sesha@SRIRAM-LENOVO ~ 
python --version && aws --version && cdk --version && sam --version && eb --version && terraform.exe --version
Python 3.13.6
aws-cli/2.28.6 Python/3.13.4 Windows/11 exe/AMD64
2.1024.0 (build 8be6aad)
SAM CLI, version 1.142.1
EB CLI 3.25 (Python 3.13.6 (tags/v3.13.6:4e66535, Aug 6 2025, 14:36:00) [MSC v.1944 64 bit (AMD64)])
Terraform v1.13.0
on windows_amd64
```

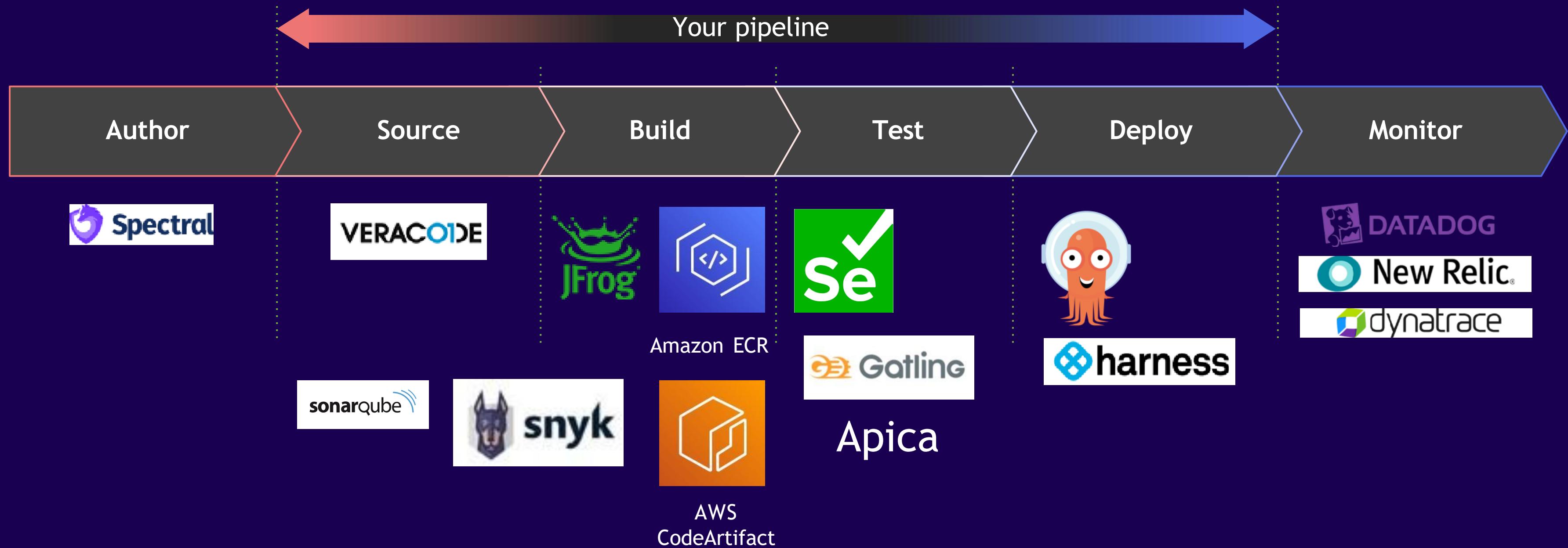
```
sesha@SRIRAM-LENOVO ~ 
-
```

```
aws default@ap-south-1
```

CI/CD for modern software delivery

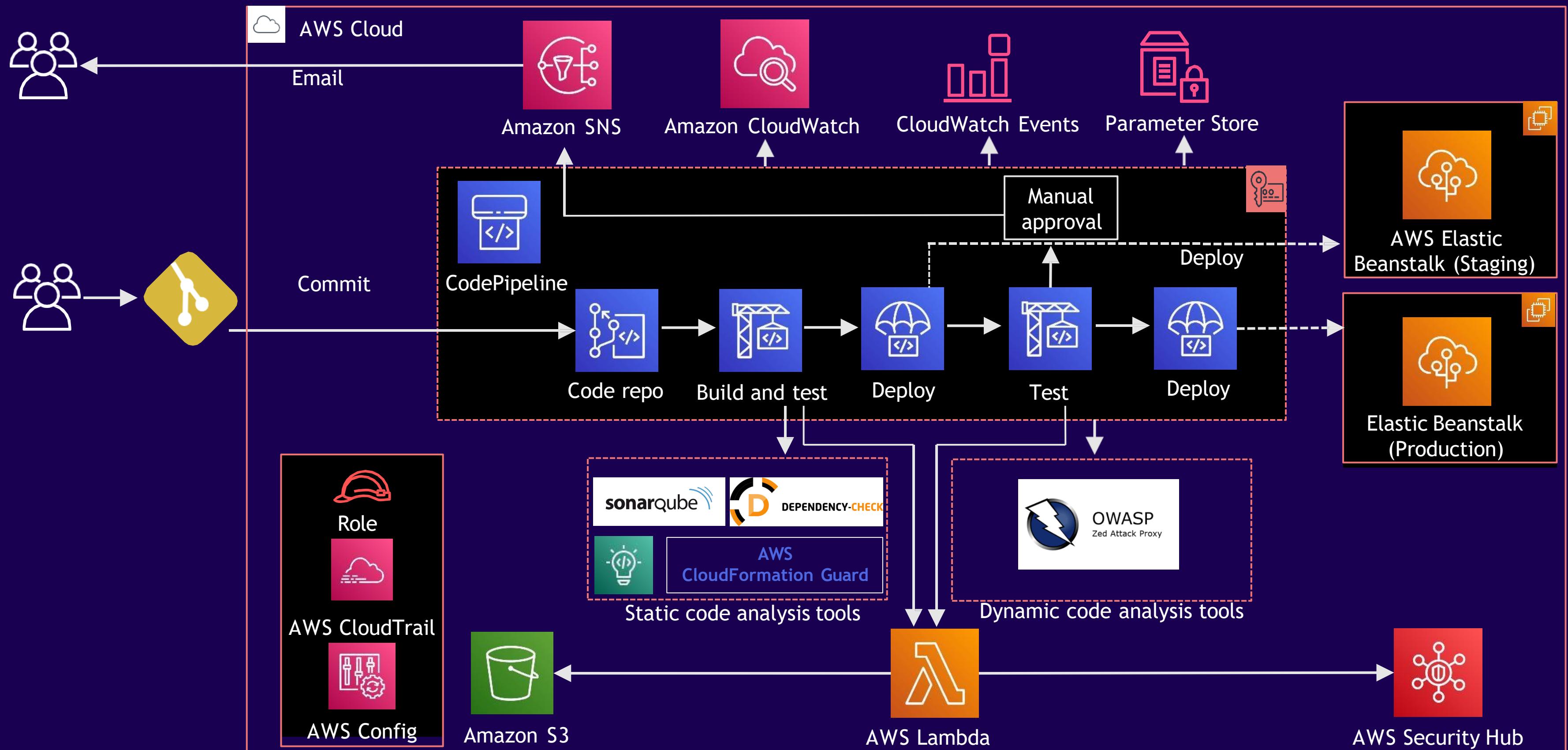


Value-added integrations

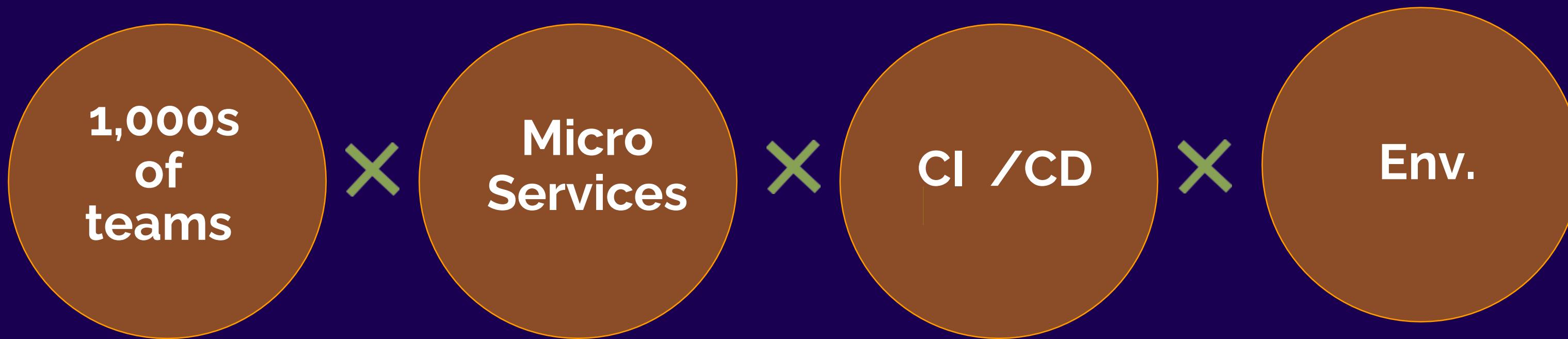


This is not a complete list. To view all AWS Partners for this category, visit AWS Partner Solutions Finder.

Additional integration (DevSecOps)



Deployment @ Scale @ Amazon



190 Million Deployments/Year

DevOps tooling is critically important for successful practices



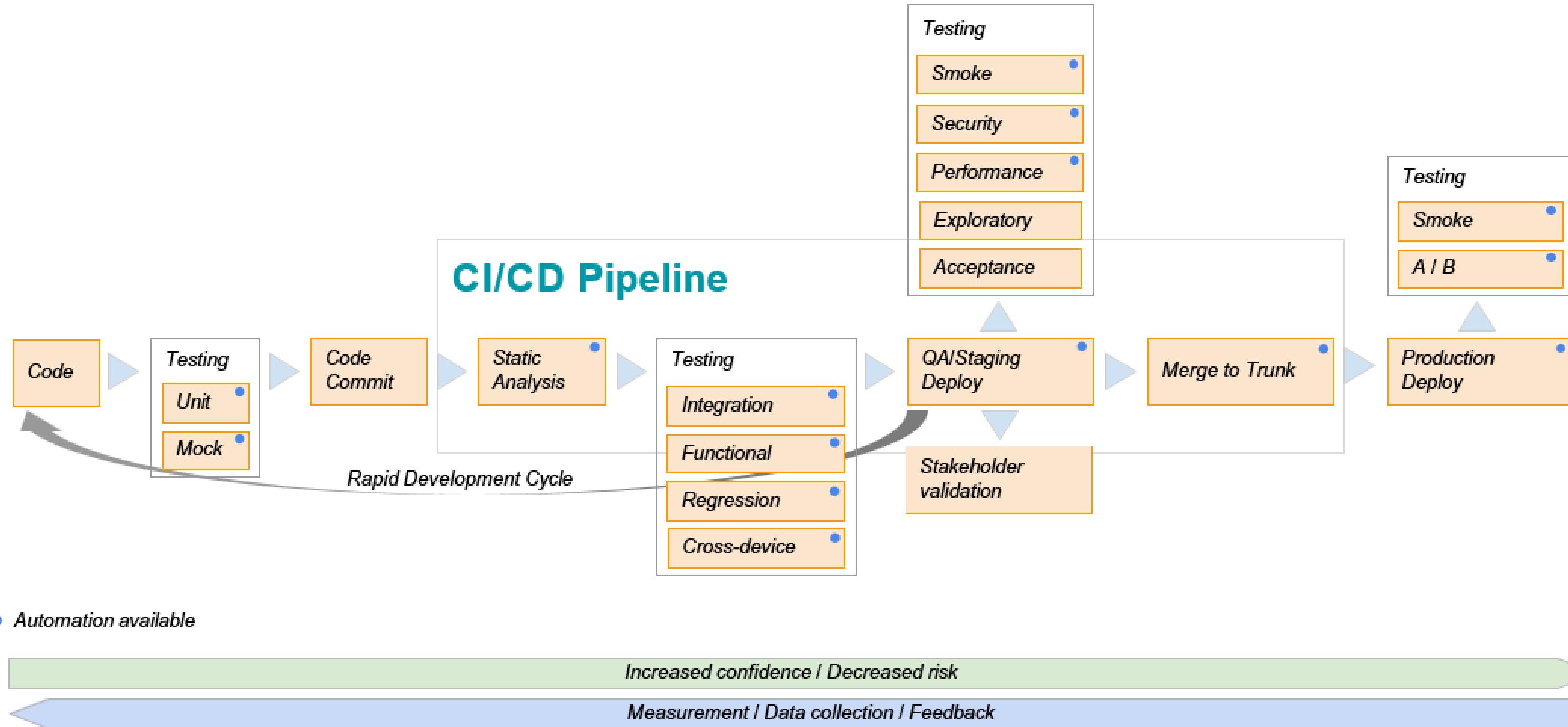
The Periodic Table of DevOps Tools (V4.2)

digital.c

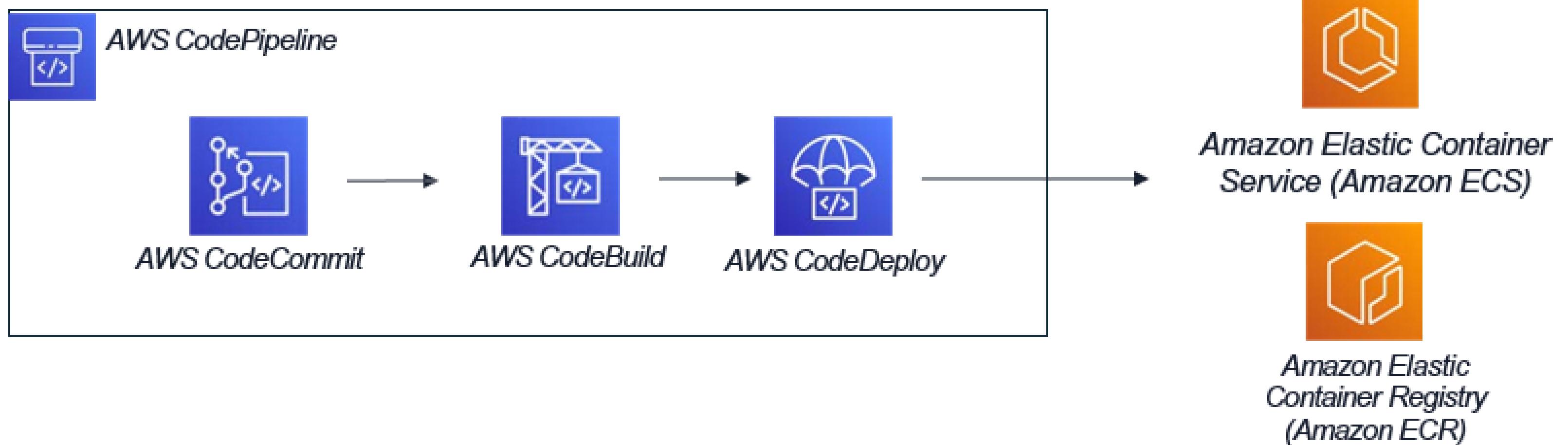
CollabNetVersionOne, XebiaLabs, Arxan, Numerify & Ex-



Sample Pipeline



Start Simple

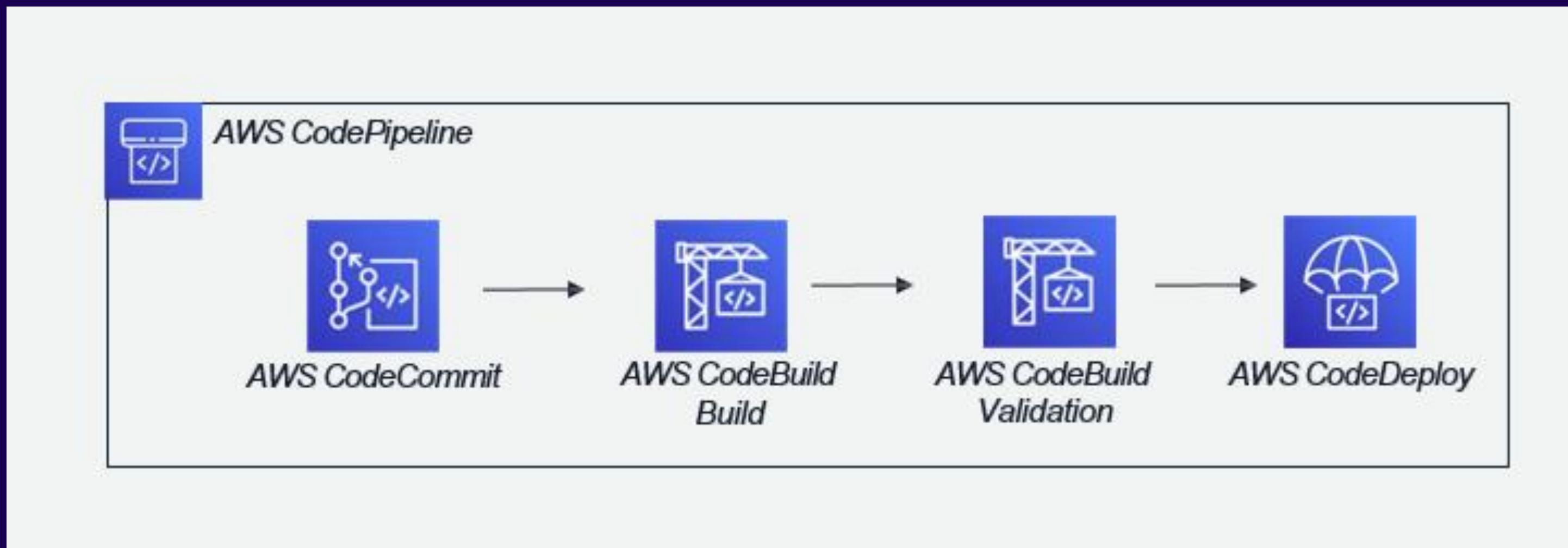


Start Simple

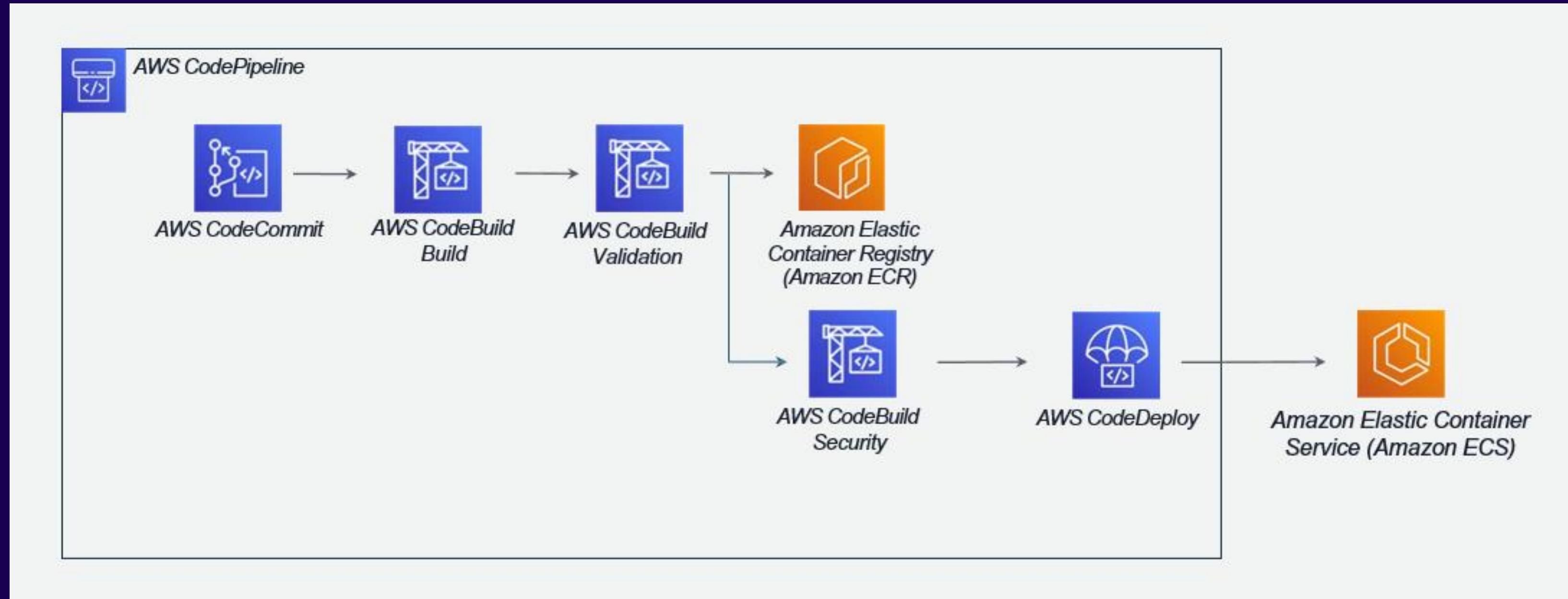
Start with Buildspec files (YAML based)

```
version: 0.2
phases:
  install:
    runtime-versions:
      docker: 18
  pre_build:
    commands:
      - echo Logging in to Amazon ECR...
      - $(aws ecr get-login --no-include-email --region $AWS_DEFAULT_REGION)
  build:
    commands:
      - echo Build started on `date`
      - echo building the C binary
      - make all
      - ./pytest.py
      - mkdir -p flaskapp\
      - cp flask/requirements.txt ./flaskapp
      - cp flask/application.py ./flaskapp
      - cp -R ./pycalc/ ./flaskapp
      - python3 -m venv ./flaskapp
      - cp ./bin/* ./flaskapp/bin/
      - echo Building the Docker image...
      - docker build -t $IMAGE_REPO_NAME:$IMAGE_TAG_LATEST .
      - docker tag $IMAGE_REPO_NAME:$IMAGE_TAG_LATEST $AWS_ACCOUNT_ID.dkr.ecr.$AWS_DEFAULT_REGION.amazonaws.com/$IMAGE_REPO_NAME:$IMAGE_TAG_LATEST
      - echo Pushing the Docker image...
      - docker push $AWS_ACCOUNT_ID.dkr.ecr.$AWS_DEFAULT_REGION.amazonaws.com/$IMAGE_REPO_NAME:$IMAGE_TAG_LATEST
  post_build:
    commands:
      - echo Build completed on `date`
      - ls
```

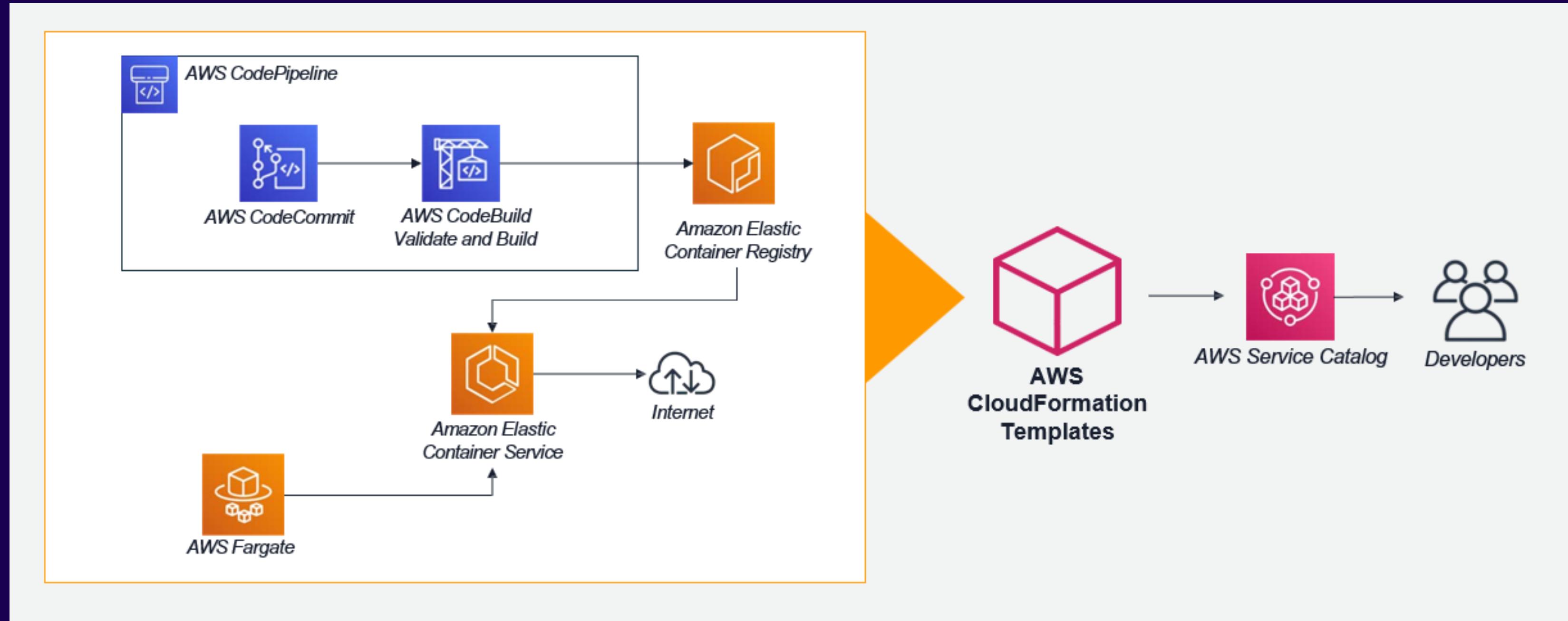
Increment - Add Validations



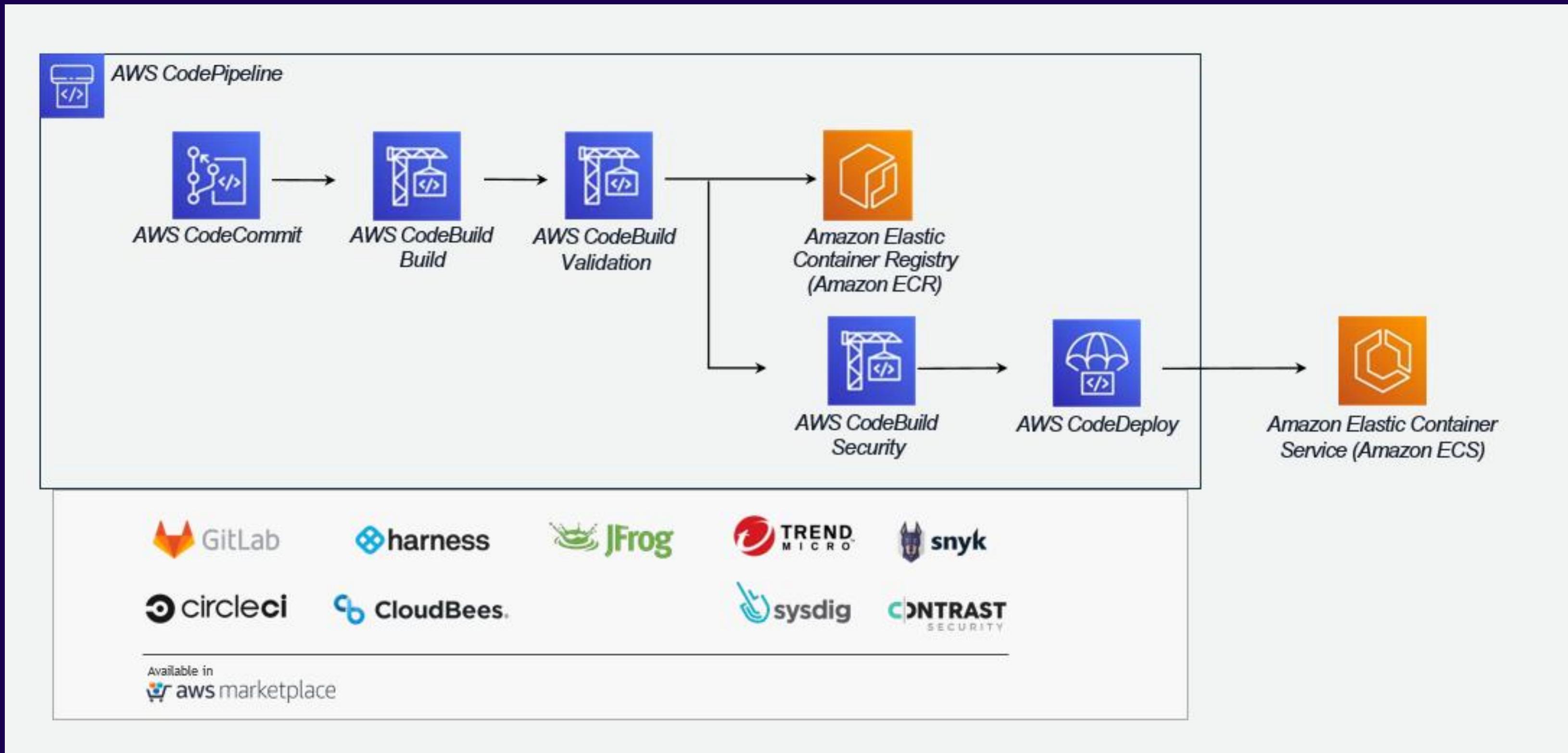
Increment - Add Validations + Security



Self Service Platform



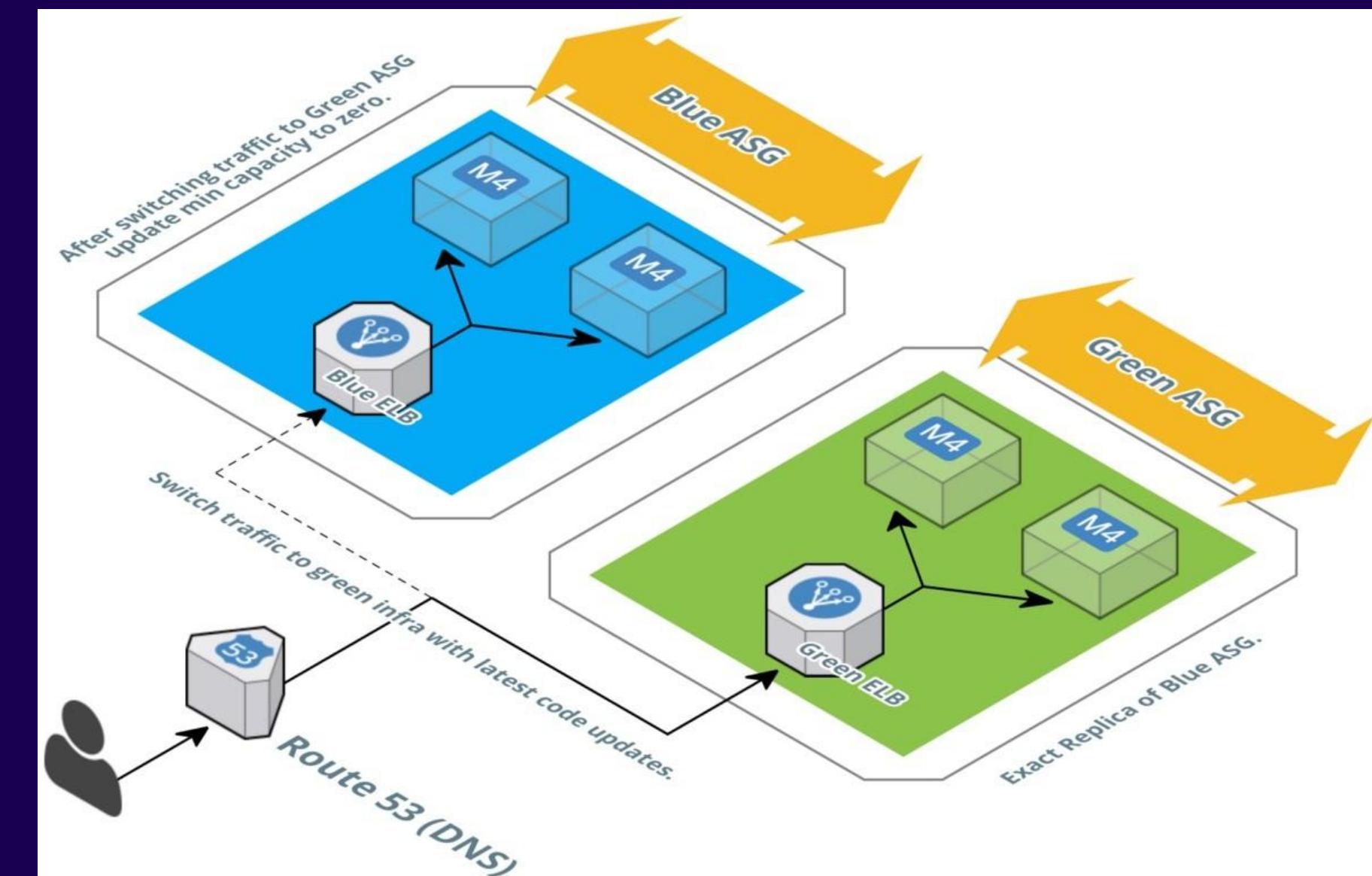
Enhance with 3rd Party Providers



Moving to Continuous Deployments

Patterns

- Blue / Green
- A/B
- Canary
- Feature Flags
- Cloud Native Architectures



Moving to Continuous Deployments

Cloud Native Principles

- Everything is automated
- Loose Coupling
- Packaged and Immutable
- High Observability
- Single Responsibility
- 15 factors

Moving to Continuous Deployments

12 + 3 Factors

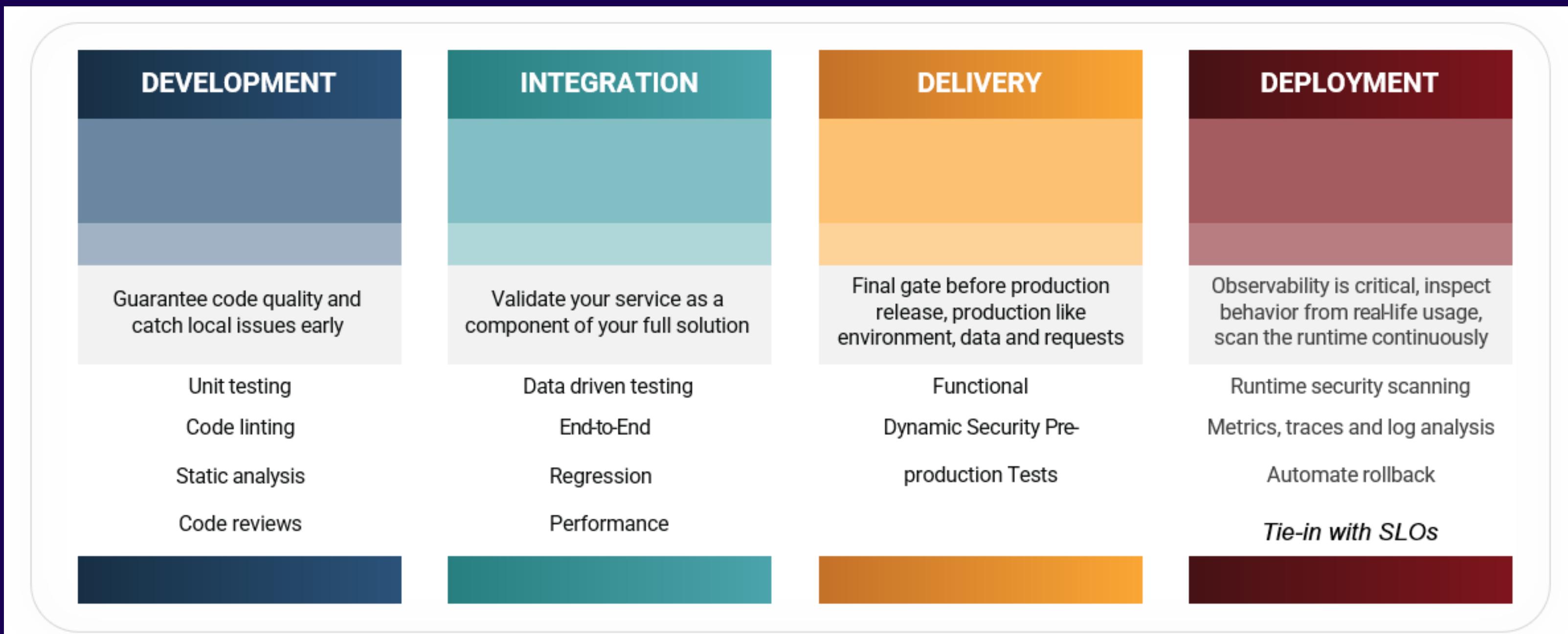
- First Published by Heroku in 2011
- 12factor.net
- 12 +3 (added by Kevin Hoffman @ Pivotal in 2016)

Moving to Continuous Deployments

1	One codebase, one app	7	Disposability	
2	API First	8	Backing Services	13
3	Dependency Management	9	Environment Parity	14
4	Design, build, release, run	10	Administrative Processes	15
5	Configuration, credentials and code	11	Port Binding	Authentication and Authorization
6	Logs	12	Stateless Processes	Telemetry

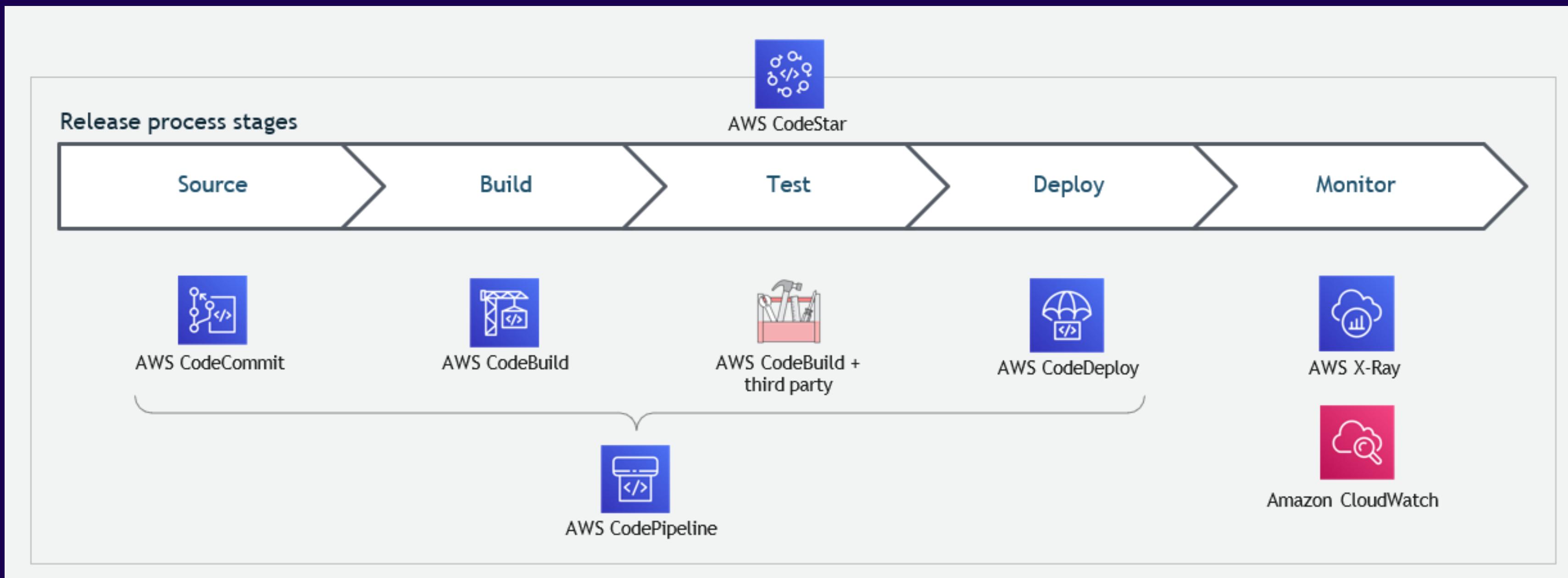
Moving to Continuous Deployments

Testing is the cornerstone of successful Devops



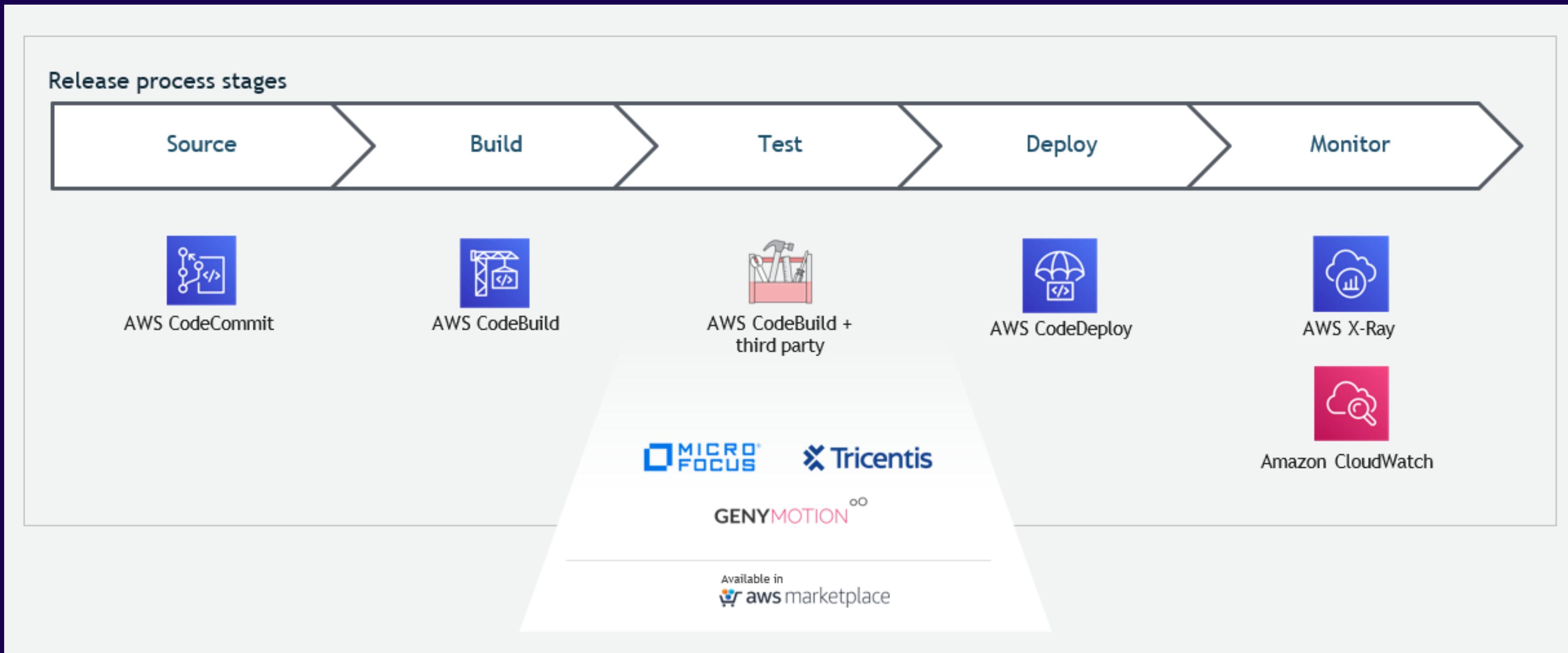
Moving to Continuous Deployments

Testing is the cornerstone of successful Devops

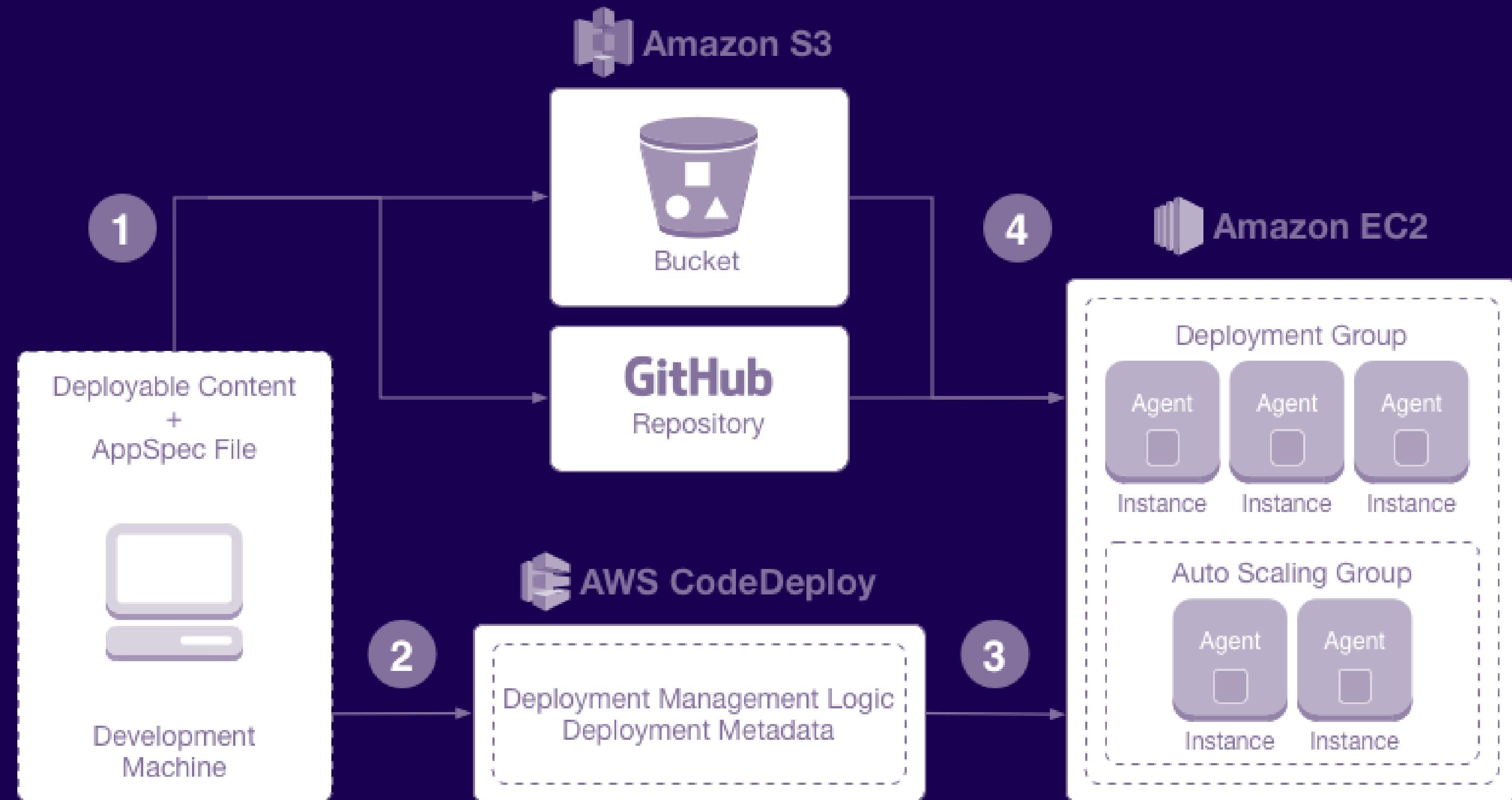


Moving to Continuous Deployments

Testing is the cornerstone of successful Devops



AWS CodeDeploy



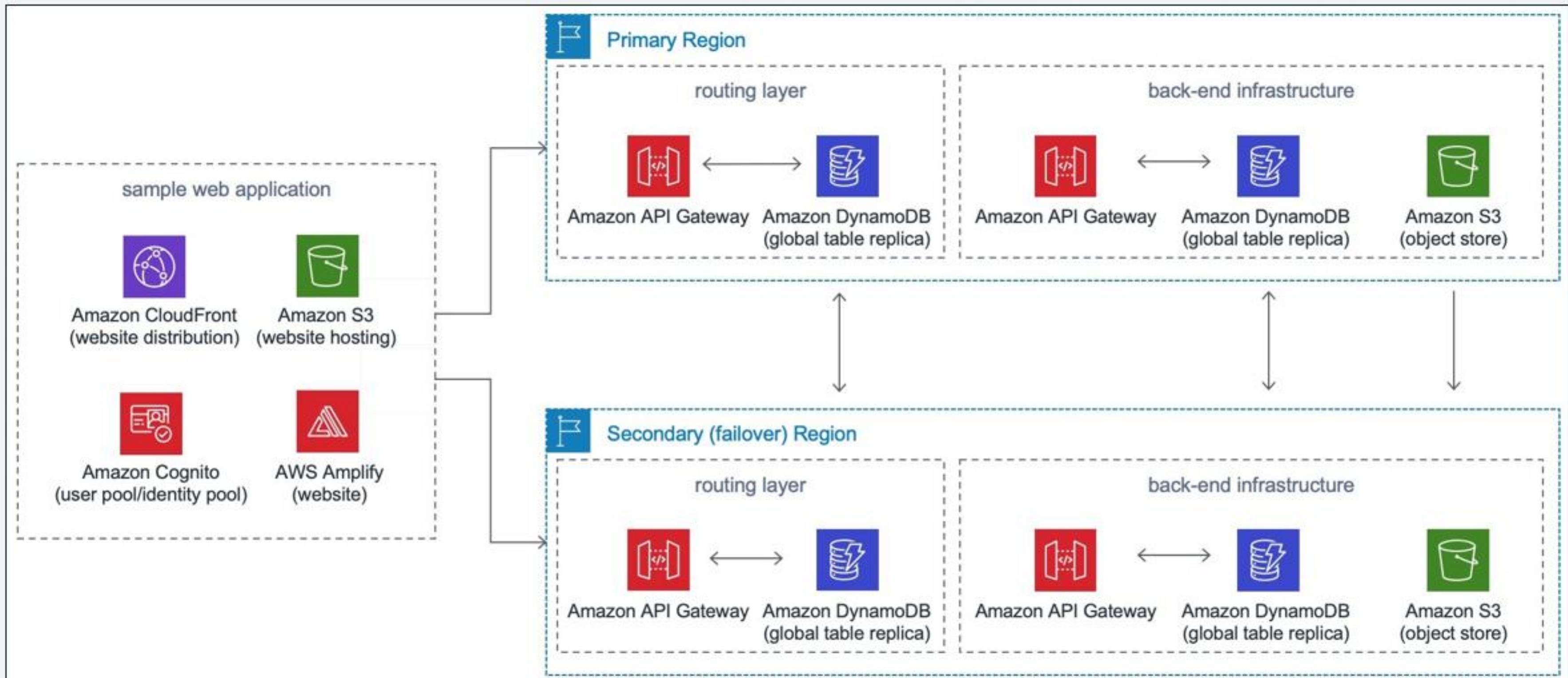
AWS CodeDeploy

- Coordinates Automated Deployments via spec files.
- Fast Auto Scaling
- Centralize Deployment Control and Monitoring
- AWS CloudFormation support creation of CodeDeploy resources

Supports

In Place
Blue/Green Deployments

Infrastructure as Code



Infrastructure as Code



AWS CloudFormation



Amazon Elastic Compute
Cloud (Amazon EC2)



Amazon Simple Storage
Service (Amazon S3)



Amazon DynamoDB



Amazon Simple Notification
Service (Amazon SNS)

AWS Cloud API



Available in
 aws marketplace

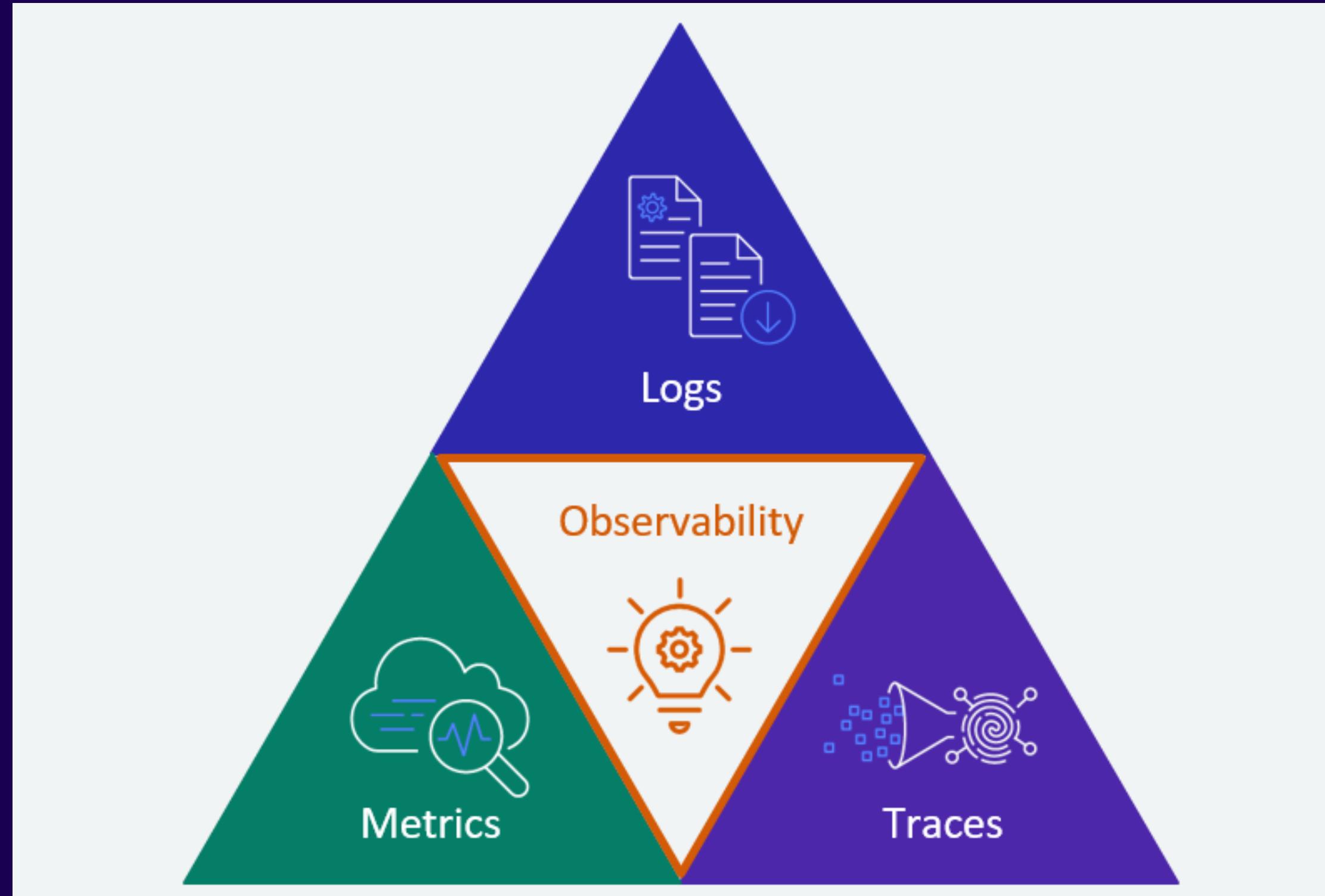
Other Practices

Continuous Testing

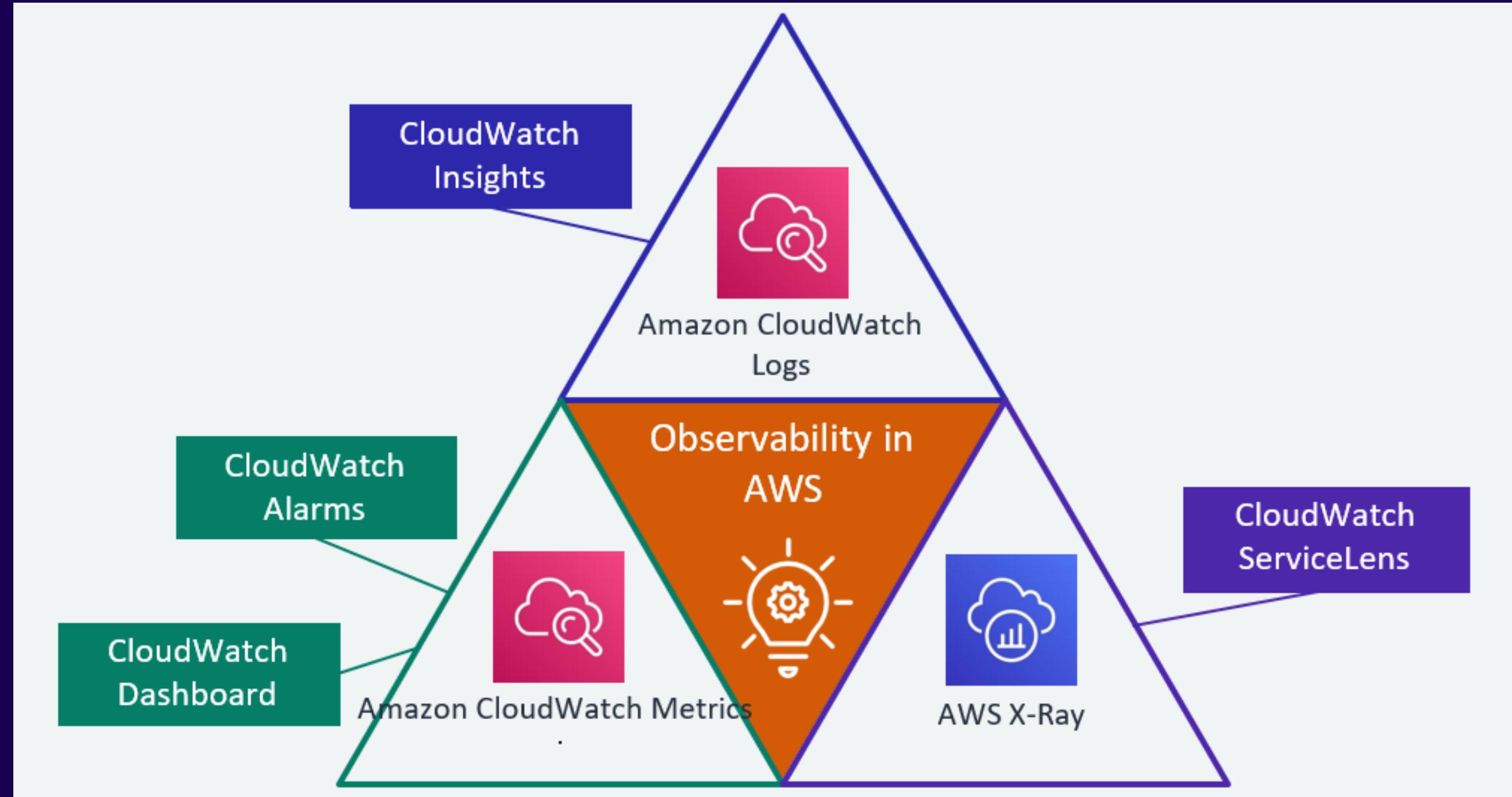
Monitoring and Observability

Site Reliability Engineering & Incident Management

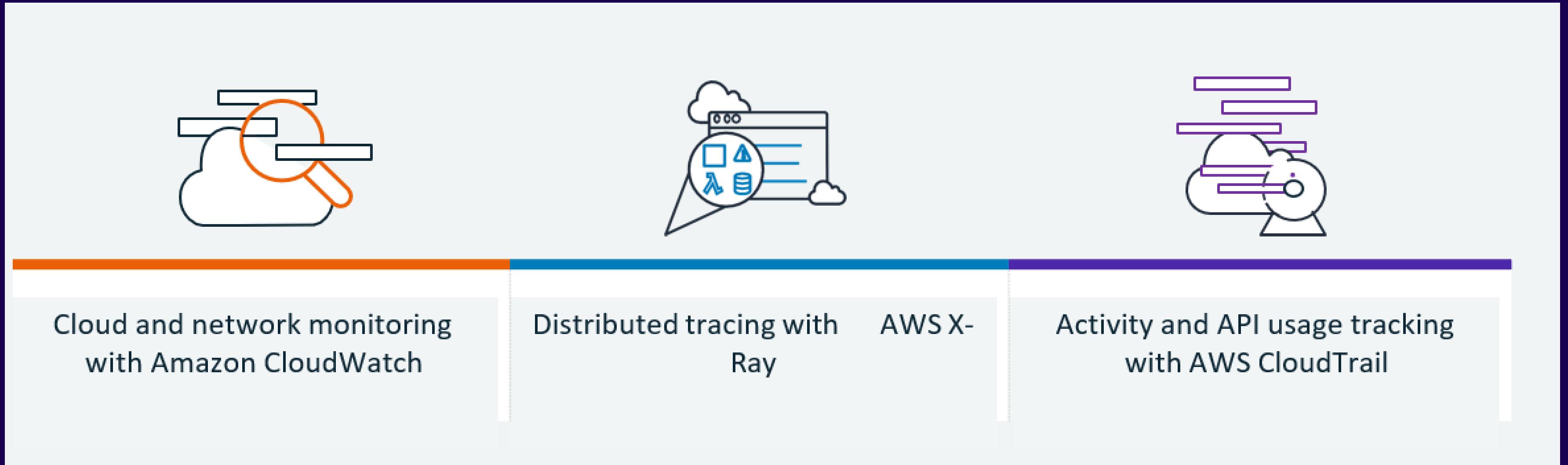
Monitoring, Observability, SRE and Incident Mgmt



Monitoring, Observability, SRE and Incident Mgmt

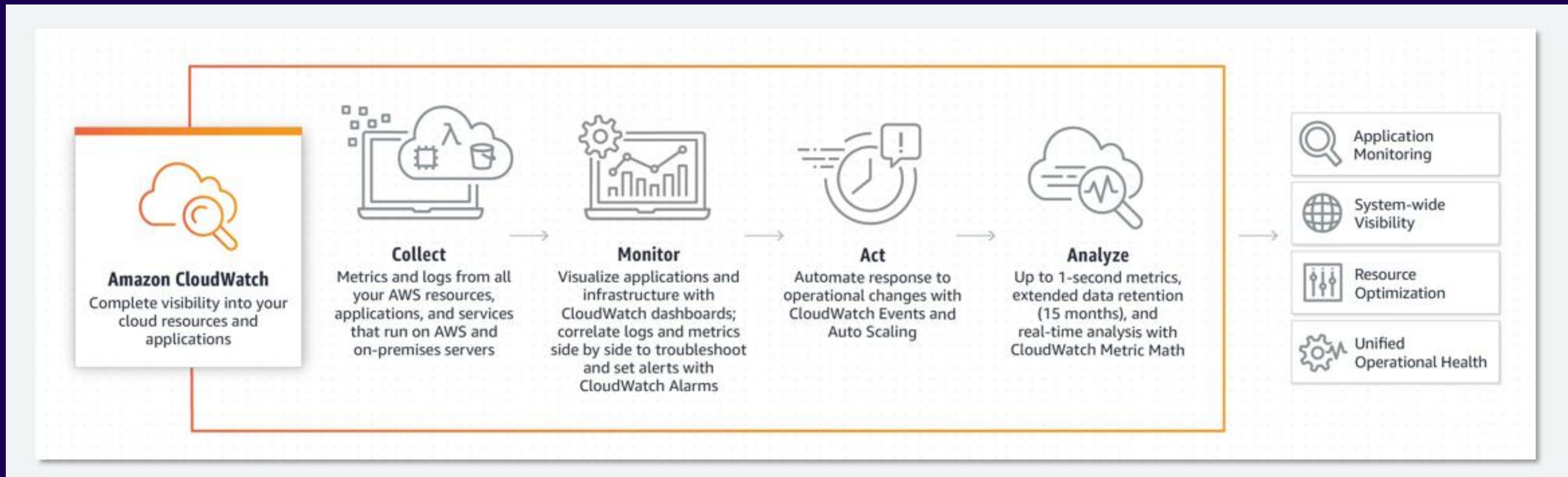


Monitoring, Observability, SRE and Incident Mgmt



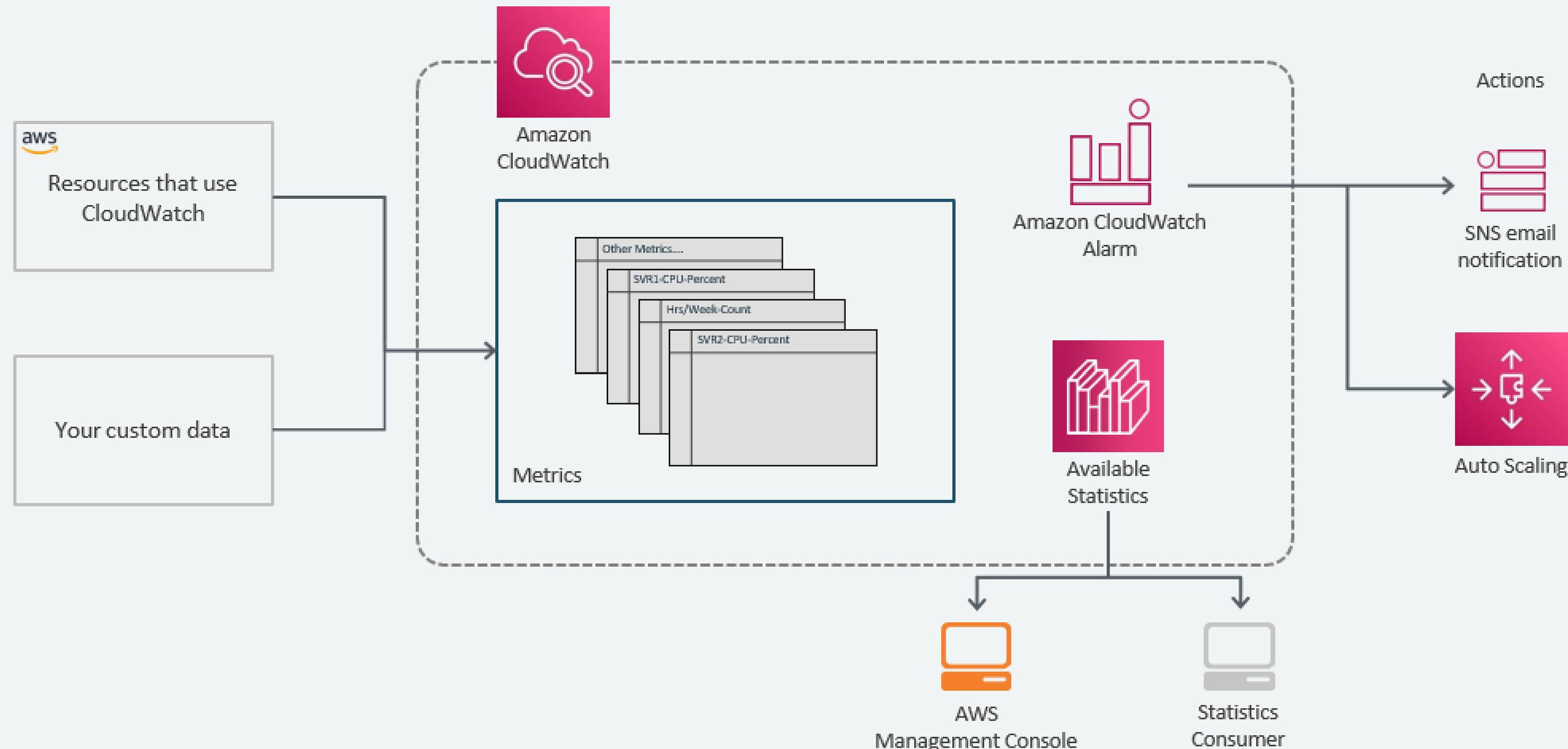
Record logs and monitor application and infrastructure performance in near real-time

Monitoring, Observability, SRE and Incident Mgmt



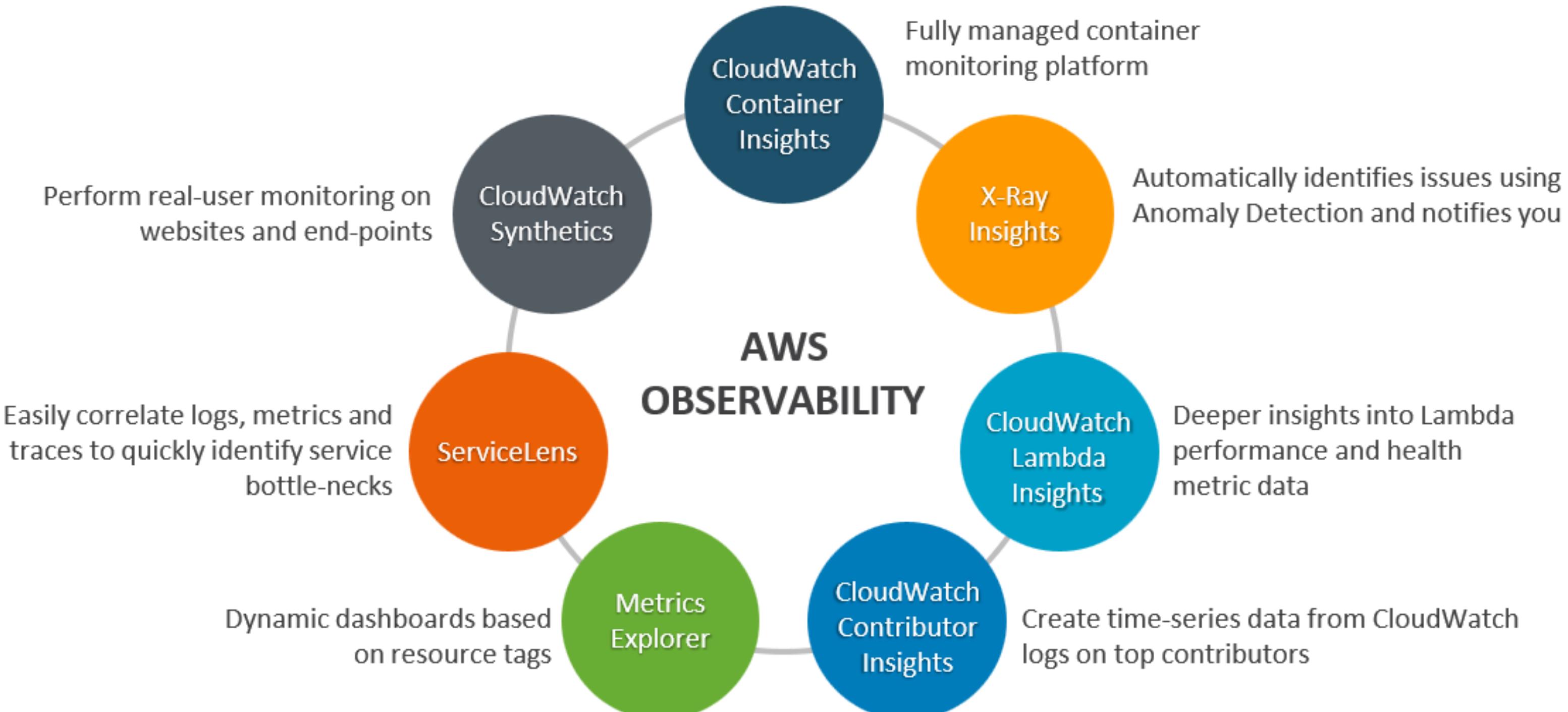
Continuously analyze telemetry data and identify anomalous behavior

Monitoring, Observability, SRE and Incident Mgmt



Monitoring, Observability, SRE & Incident Mgmt.

Infrastructure, application, and synthetic monitoring



Monitoring, Observability, SRE & Incident Mgmt.

OBSERVABILITY

AWS NATIVE MONITORING SERVICE

CloudWatch ServiceLens

Container Insights



CloudWatch Logs

Lambda Insights



CloudWatch Metrics

Synthetics

Contributor Insights



AWS X-Ray

OPEN-SOURCE MANAGED SERVICES

Amazon Managed Service for



Do it Yourself (DIY)



Amazon
Elasticsearch
Service—Logs



Amazon Managed
Service for
Prometheus



Jaeger & Zipkin
—Tracing

INSIGHTS AND MACHINE LEARNING

INSTRUMENTATION



CloudWatch Agent



X-Ray agent

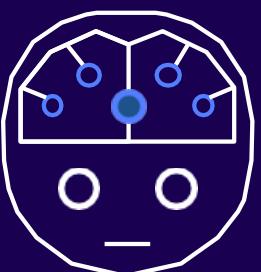


AWS Distro for OpenTelemetry (ADOT)

Section 3 - Innovation with AWS

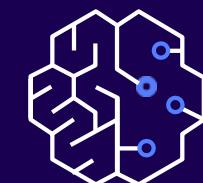
Machine Learning

What is machine learning?



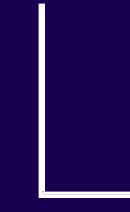
Artificial intelligence (AI)

Any technique that enables computers to mimic human intelligence using logic, if-then statements, and machine learning (including deep learning)



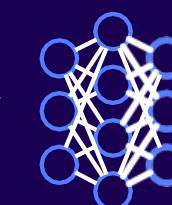
Machine learning (ML)

Subset of AI that uses machines to search for patterns in data to build logic models automatically



Deep learning

Subset of ML composed of deeply multi-layered neural networks that perform tasks like speech and image recognition



Getting started: Common machine learning use cases

Solve real-world problems with machine learning (ML)

Enhance the customer experience



Personalization



Contact center
intelligence



Media
intelligence

Delight customers
while reducing operational costs

Optimize the business



Intelligent
search



Intelligent
document
processing



Fraud
detection



Business
metrics
analysis

Improve productivity
and optimize business processes

Accelerate innovation



ML
modernization

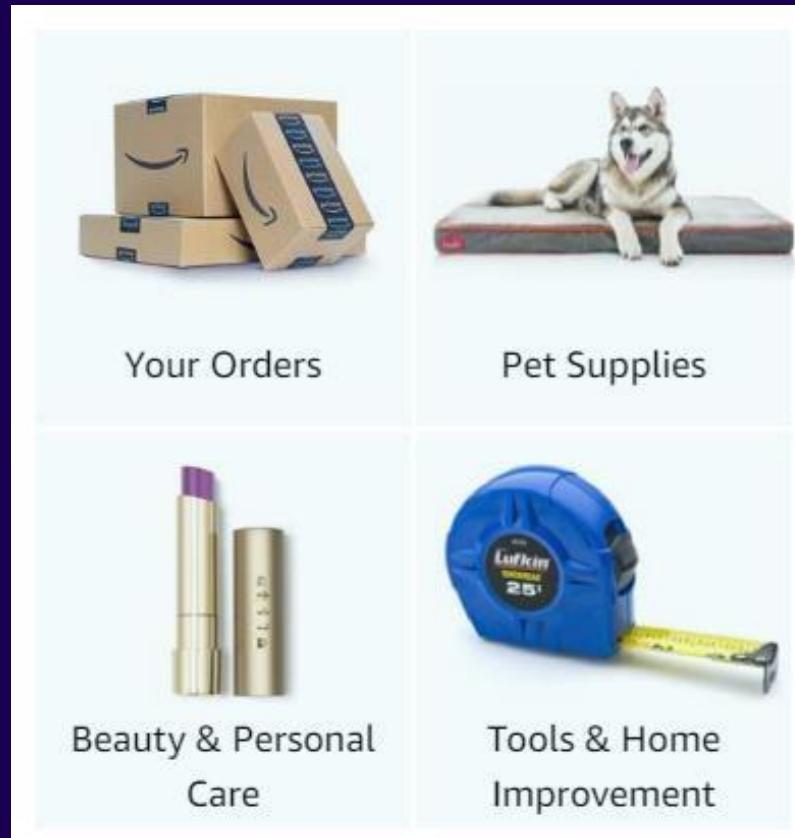


Next gen
DevOps

Speed up and scale up
innovation with ML

Amazon's machine learning innovation

Recommendations
for you



4,000 products per minute sold on Amazon.com



1.6 million packages every day



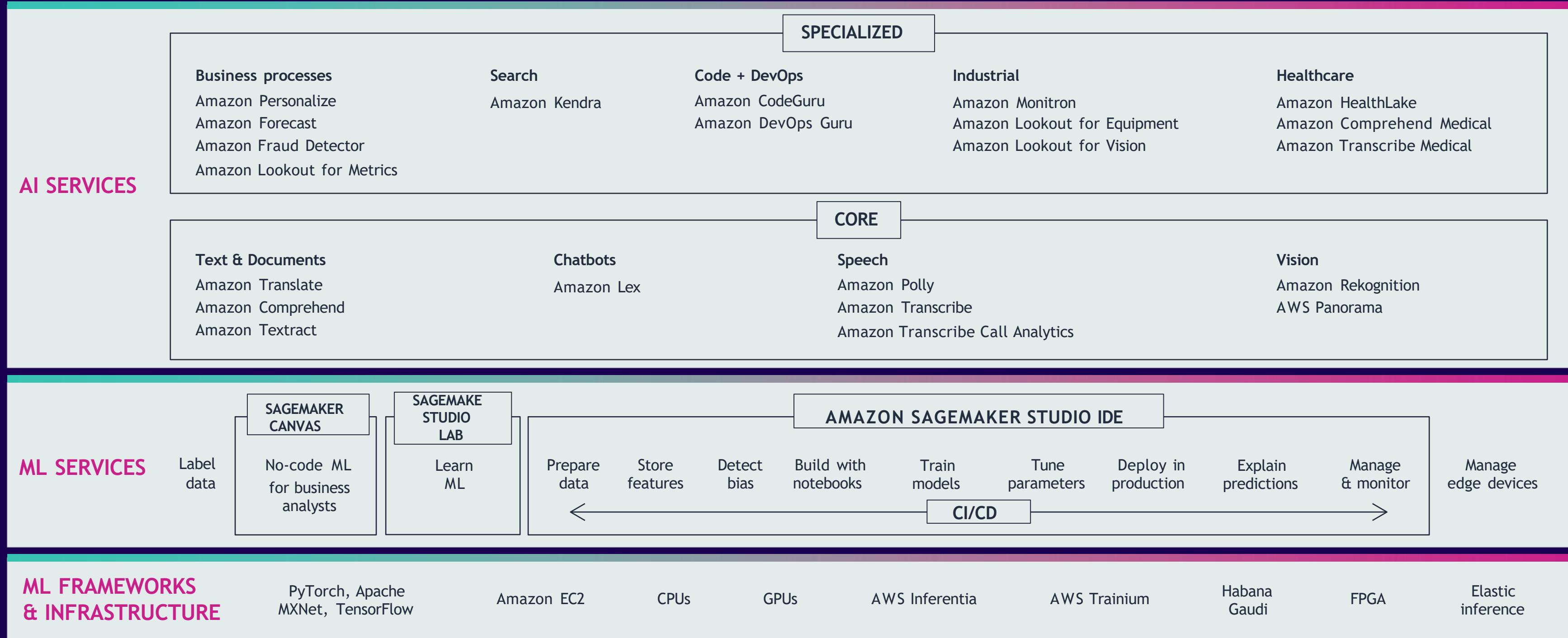
Billions of Alexa interactions each week



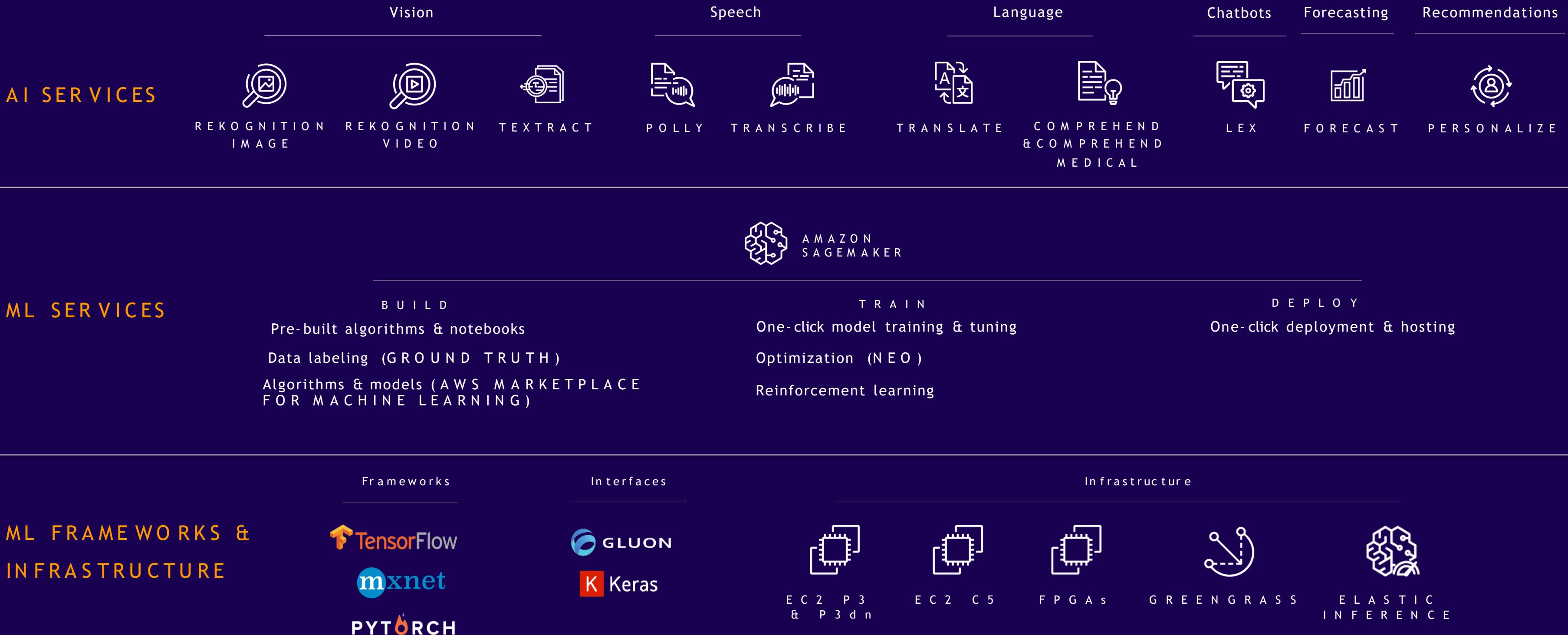
First Prime Air delivery on Dec. 7, 2016

The AWS ML stack

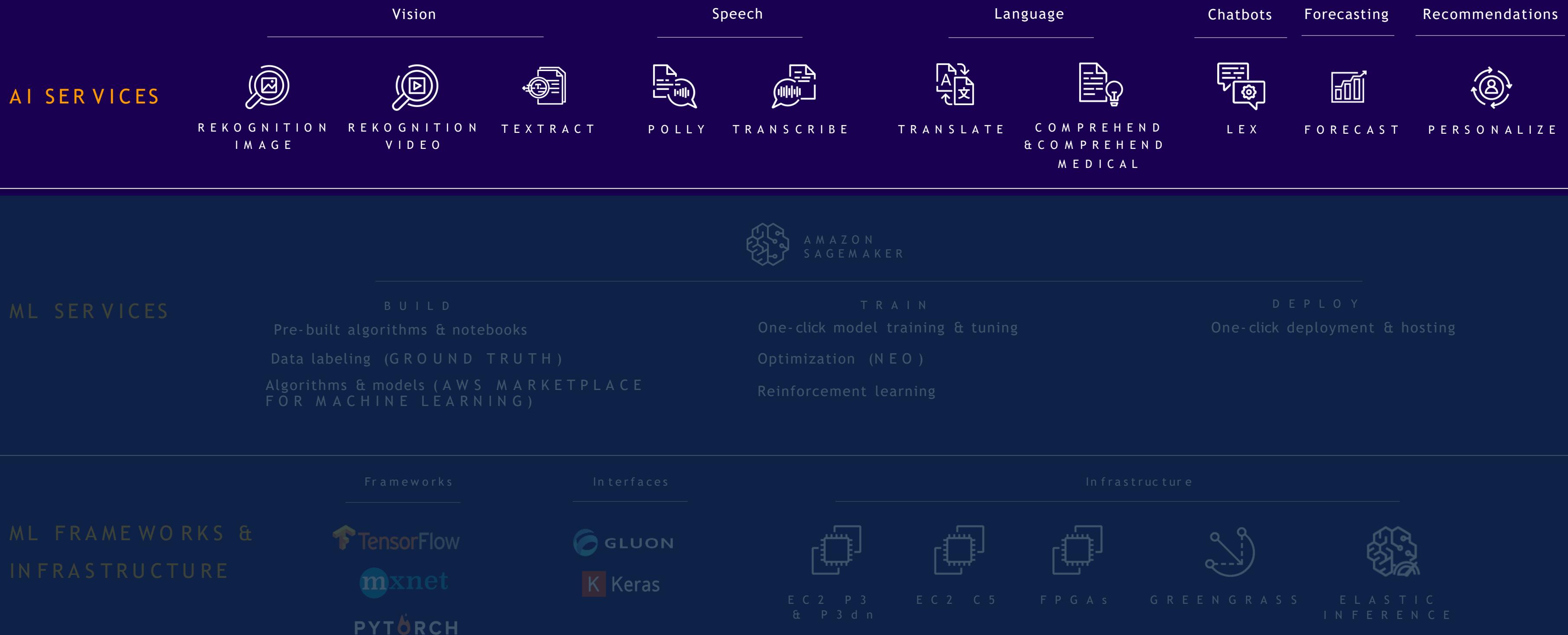
Broadest and most complete set of machine learning capabilities



The Amazon ML Stack: Broadest & Deepest Set of Capabilities

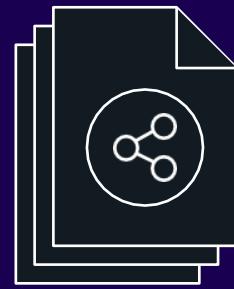


The Amazon ML Stack: Broadest & Deepest Set of Capabilities

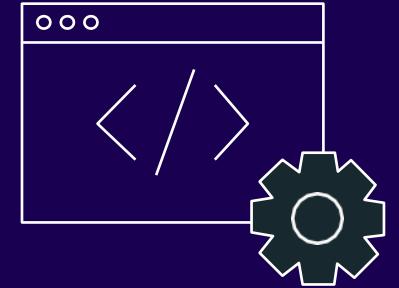


AI Services

Vision	Speech	Language	Chatbots	Forecasting	Recommendations
       	 				



Pre-trained AI services that require no ML skills or training



Easily add intelligence to your existing apps and workflows



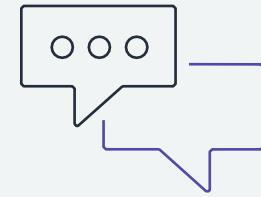
Quality and accuracy from continuously-learning APIs

AI Services: Easily add intelligence to applications

No machine learning skills required



Vision



Chatbots



Business tools



Search



Healthcare



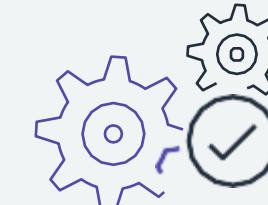
Speech



Text



Contact centers



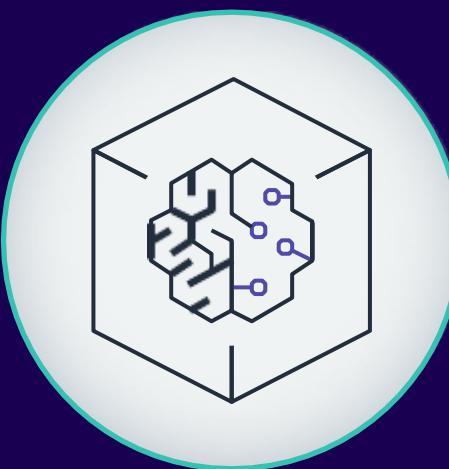
Code and
DevOps



Industrial

Amazon Fraud Detector

Identify fraud faster



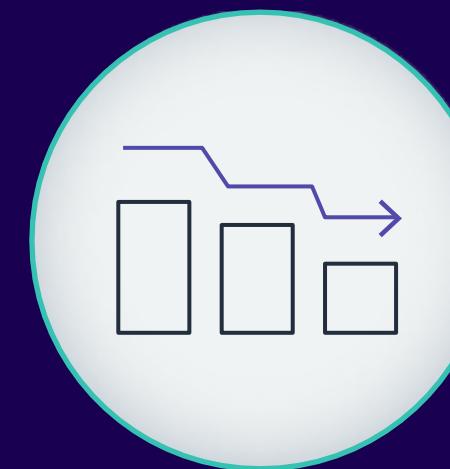
Enhance
fraud
detection
with ML



Any level of ML
expertise can
build ML fraud
models



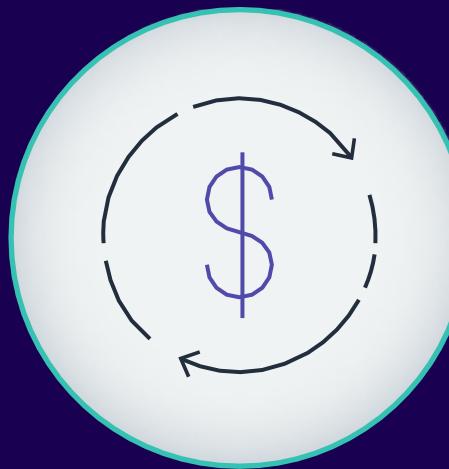
ML boost
from Amazon
experience and
enrichments



Fewer false
positives and
manual reviews



Fraud staff
self-service
to address
threats faster



Lower TCO and
faster TTV

Amazon Personalize

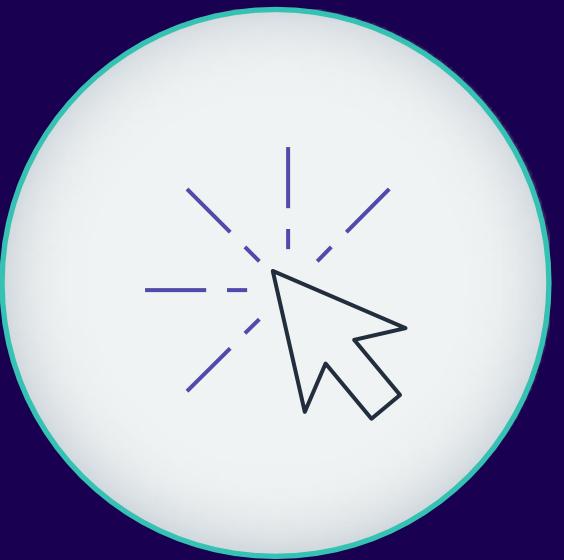
Delight customers and improve customer experience



Deliver
high-quality
recommendations



Adapt to changes
in customer intent
in real time



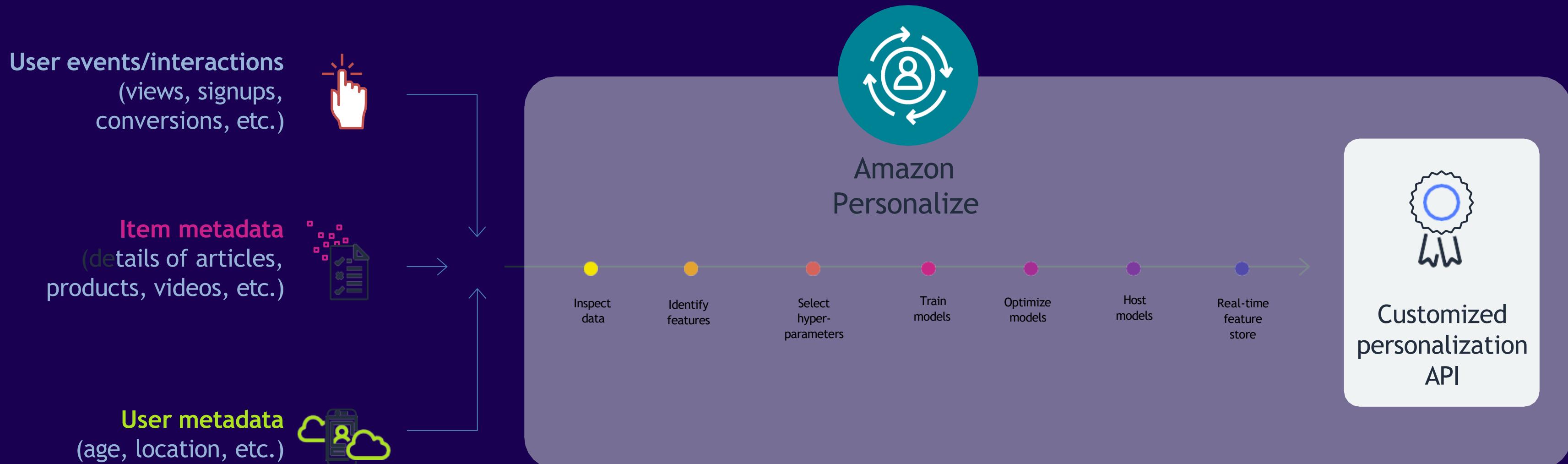
Train a
recommendation
model with a
few clicks



Generate
recommendations
for almost any
product or content

Amazon Personalize

How it works



Machine Learning APIs for :Vision

Amazon Rekognition - Image and Video Analysis





THE ROYAL WEDDING



PROFILES



David Beckham Former footballer

David Robert Joseph Beckham, OBE is an English former professional footballer. He played for Manchester United, Preston North End, Real Madrid, Milan, LA Galaxy.

Now playing...

ARRIVALS



Elton John
Singer



David Furnish
Elton John's husband



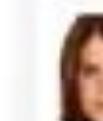
David Beckham
Former footballer



Victoria Beckham
Businesswoman



Princess Beatrice
Royal Family



Princess Eugenie
Royal Family

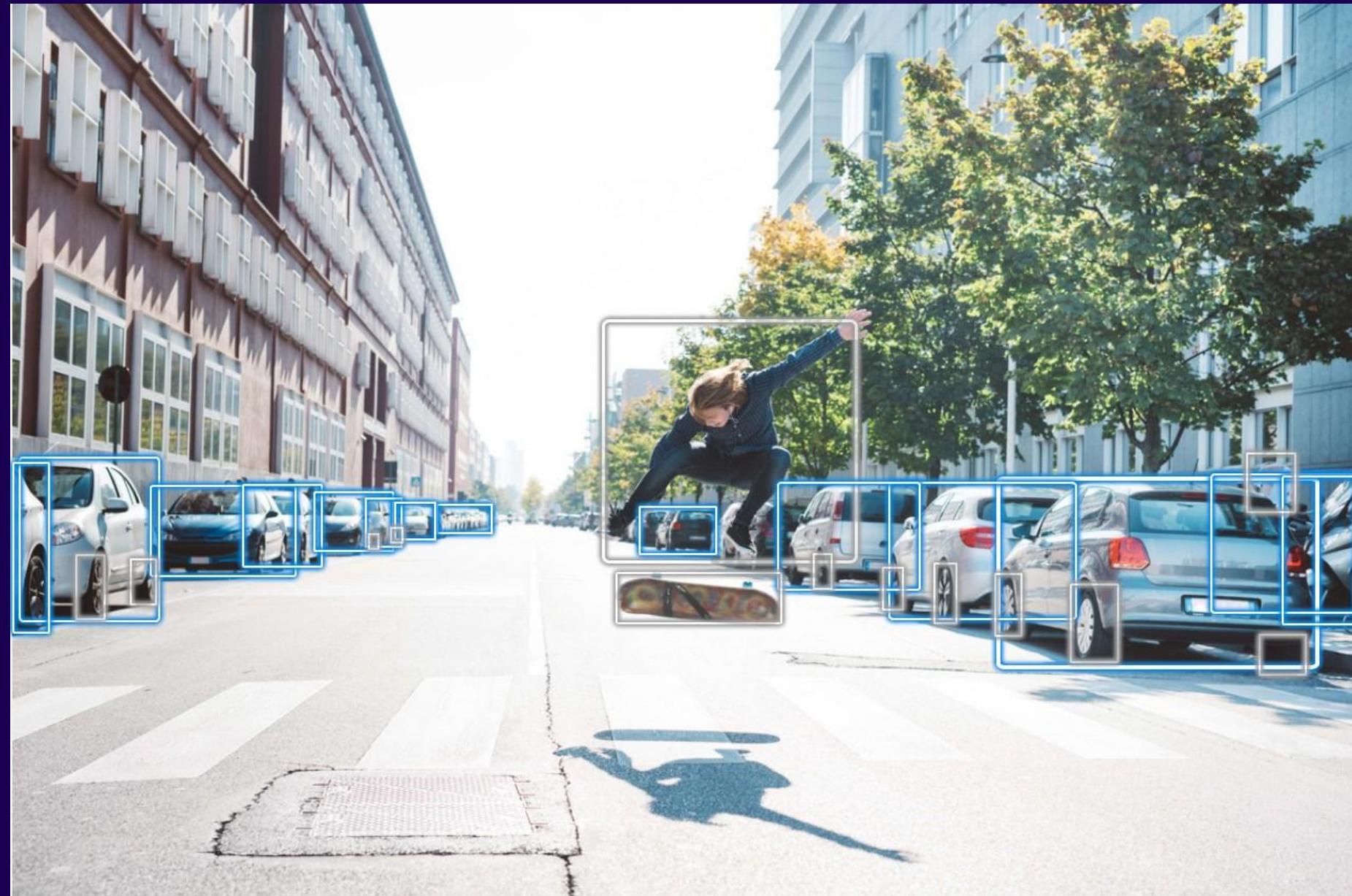


Prince Seeiso Bereng
Royal Family of Lesotho



Princess Mabereng Seeiso
Royal Family of Lesotho

Object & Scene Detection



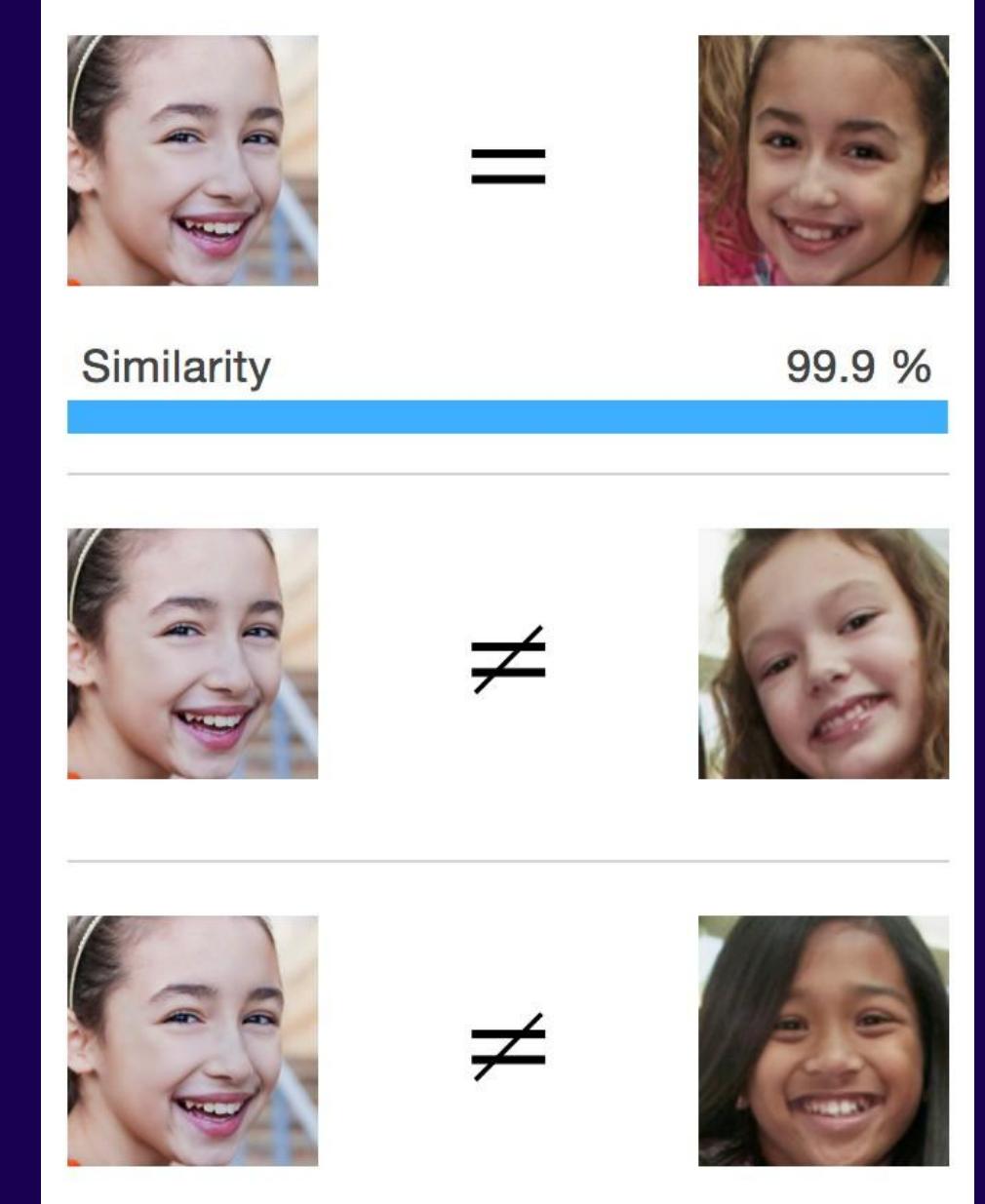
Automobile	98.8 %
Transportation	98.8 %
Vehicle	98.8 %
Car	98.8 %
Human	98.3 %
Person	98.3 %
Pedestrian	97.1 %
Skateboard	94.3 %
Sports	94.3 %
Sport	94.3 %
Road	92.4 %

Facial Analysis



looks like a face	99.9 %
appears to be female	99.8 %
age range	20 - 38 years old
smiling	98.6 %
appears to be happy	92.1 %
wearing glasses	99.9 %
wearing sunglasses	87.8 %
eyes are open	100 %
mouth is open	99.9 %
does not have a mustache	99.9 %
does not have a beard	99.9 %

Face Search/Comparison



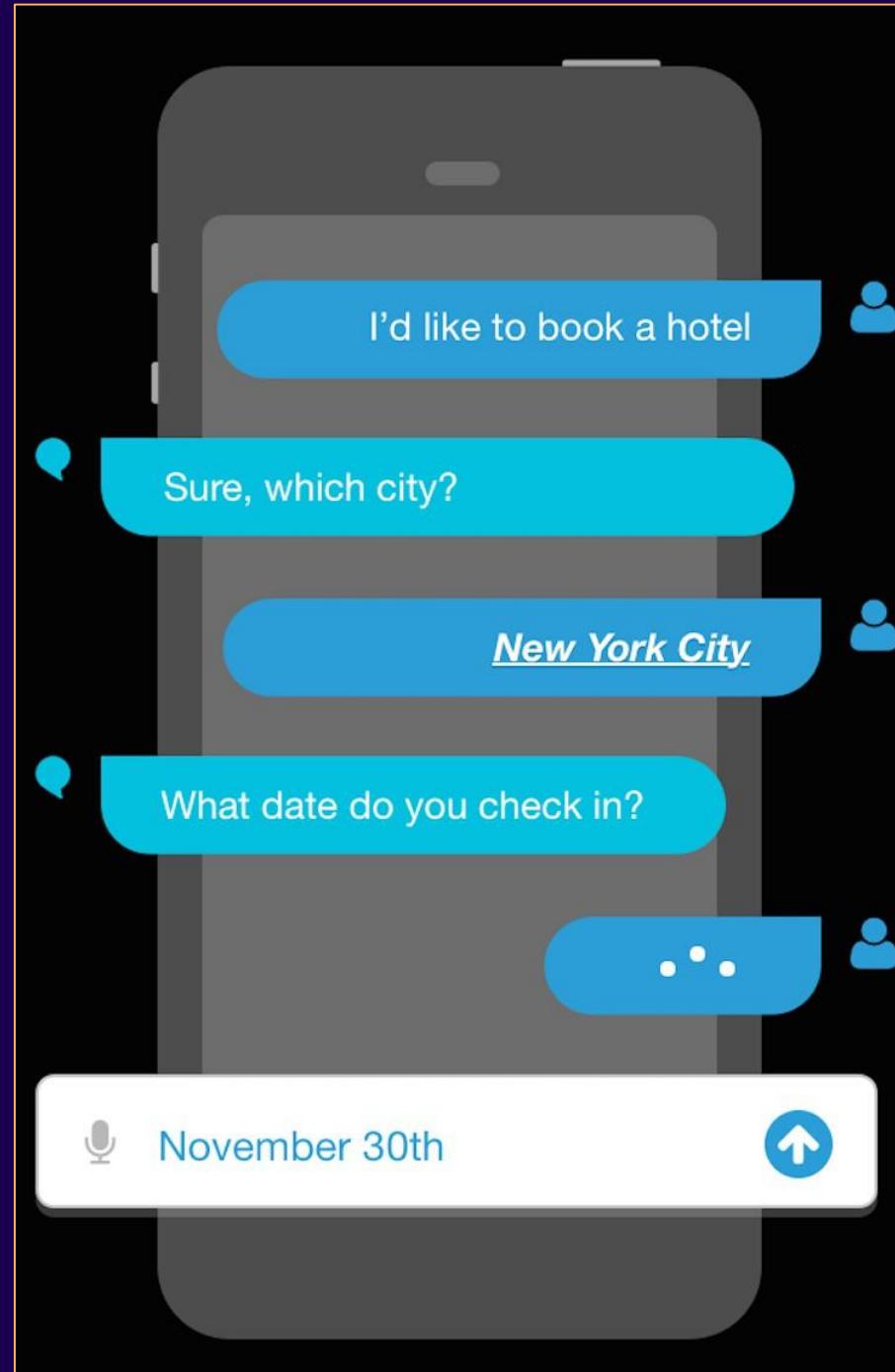
Machine Learning APIs for :Chatbots

Amazon Lex

A service for building conversational interfaces into your applications using voice and text

Amazon Lex Bots - key concepts

BookHotel



Intents

An intent performs an action in response to natural language user input

Utterances

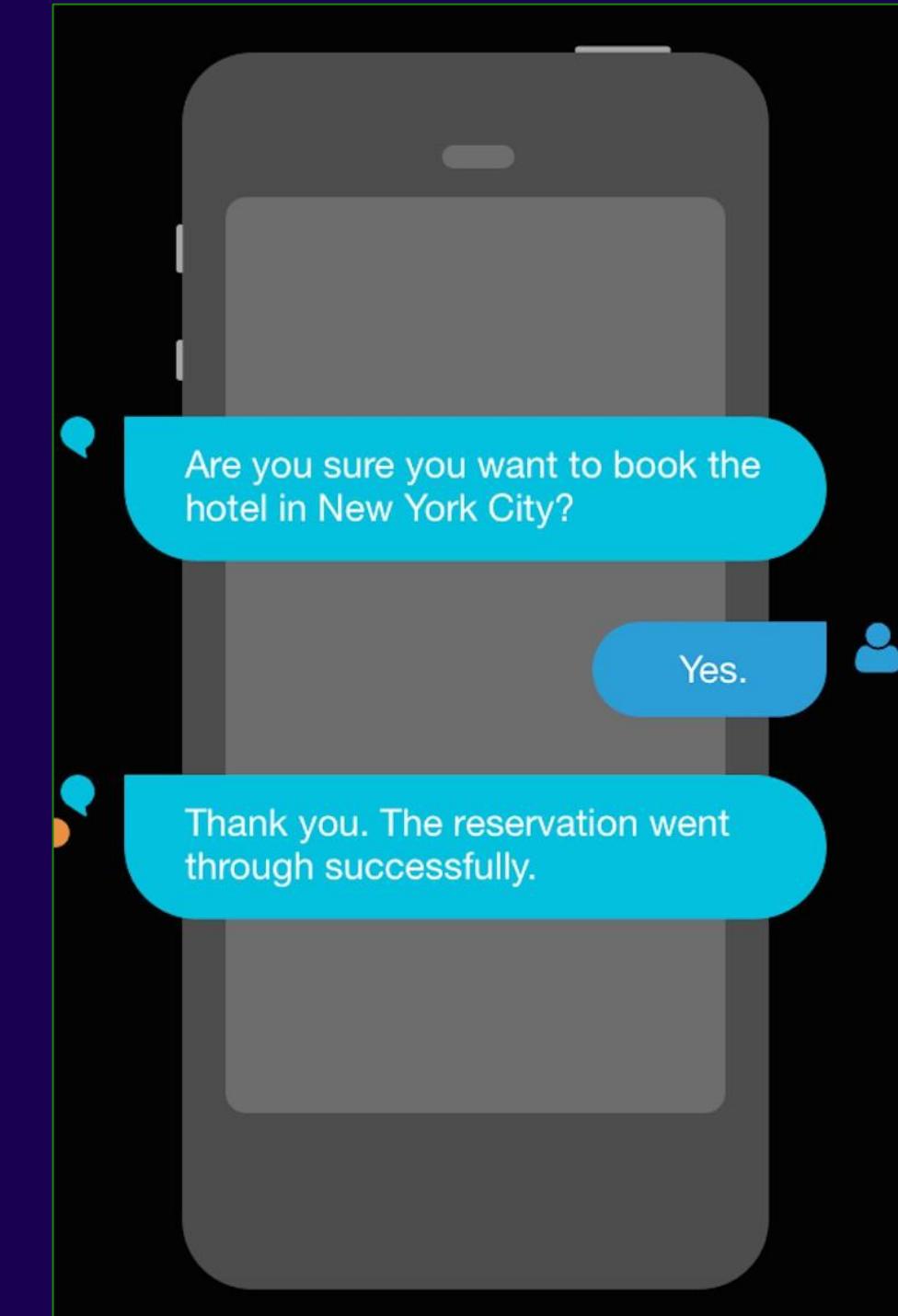
Spoken or typed phrases that invoke your intent

Slots

Slots are input data required to fulfill the intent

Fulfillment

Fulfillment mechanism for your intent



Machine Learning APIs for :Speech

Amazon Polly

Turn text into lifelike speech using deep learning

Amazon Polly

Use cases

- Content creation
- Mobile & desktop applications
- Internet of Things (IoT)
- Education & e-learning
- Telephony
- Game development

Key features

- 58 voices across 28 languages
- Lip-syncing & text highlighting
- Fine-grained voice control
- Custom vocabularies
- Available in 18 AWS Regions

Synthesize Speech API

```
$ aws polly synthesize-speech  
  --text "hello"  
  --voice-id Suman  
  --output-format mp3  
  [--lexicon-names mylex1 mylex2]  
  output.mp3
```

```
{  
  "ContentType": "audio/mpeg",  
  "RequestCharacters": "11"  
}
```

Amazon Transcribe

Turn speech into text



Amazon
Transcribe



“Hello, this is Allan
speaking”

Machine Learning APIs for :Language

How do you extract insights from unstructured text?

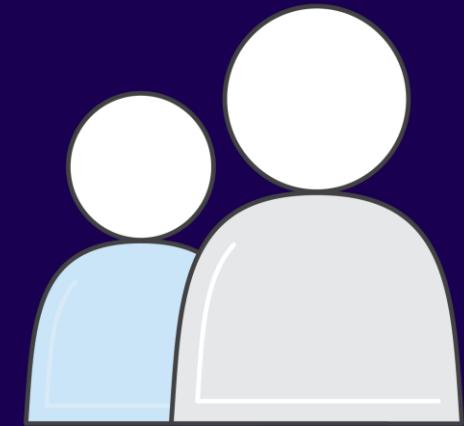
Amazon Comprehend

A fully managed and continuously trained service that helps you extract insights from unstructured text

Amazon Comprehend



Sentiment



Entities



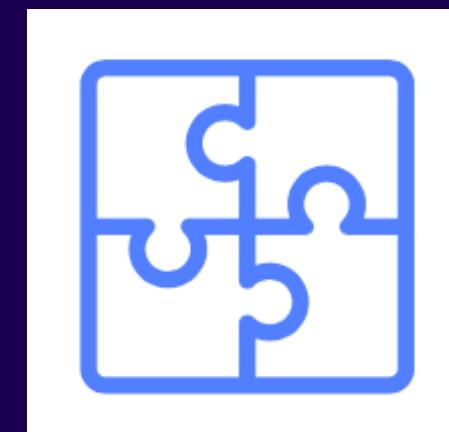
Keyphrases



Languages



Topic
modeling



Syntax

Amazon Comprehend - Natural Language Processing

Amazon.com, Inc. is located in Seattle, WA and was founded July 5, 1994 by Jeff Bezos. Our customers love buying everything from books to blenders at great prices

Named Entities

Amazon.com: Organization
Seattle, WA : Location
July 5th, 1994: Date
Jeff Bezos : Person

Keyphrases

Our customers
books
blenders
great prices

Sentiment

Positive

Language

English

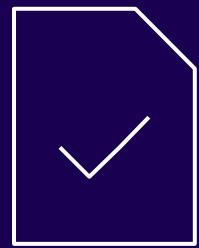
Sentiment Analysis

```
$ aws comprehend detect-sentiment  
--language-code 'en' --text 'I love cloud!'  
  
{  
    "Sentiment": "POSITIVE",  
    "SentimentScore": {  
        "Mixed": 0.012617903761565685,  
        "Positive": 0.9599817991256714,  
        "Neutral": 0.021758323535323143,  
        "Negative": 0.005641999188810587  
    }  
}
```

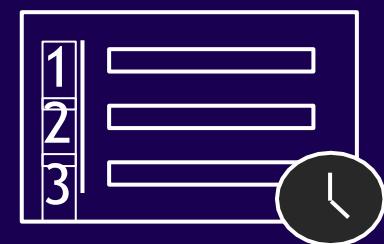
Amazon Translate

Yes, natural language translation ©

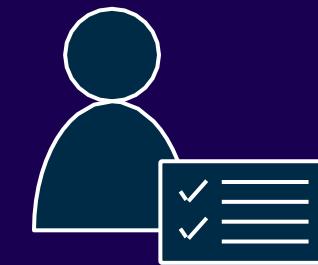
Amazon SageMaker: Build, Train, and Deploy ML Models at Scale



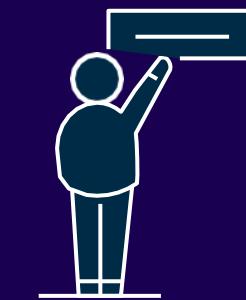
Collect and prepare
training data



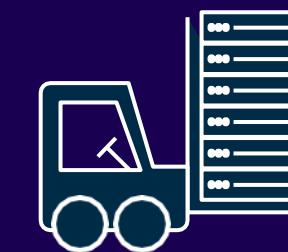
Choose and
optimize your
ML algorithm



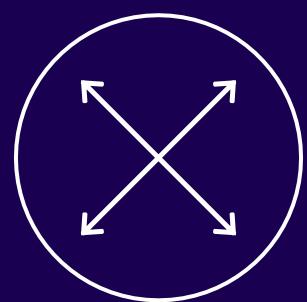
Set up and
manage
environments
for training



Train and
Tune ML Models



Deploy models
in production



Scale and manage
the production
environment

intuit



tinder



 **Liberty
Mutual**

CONVOY

SIEMENS



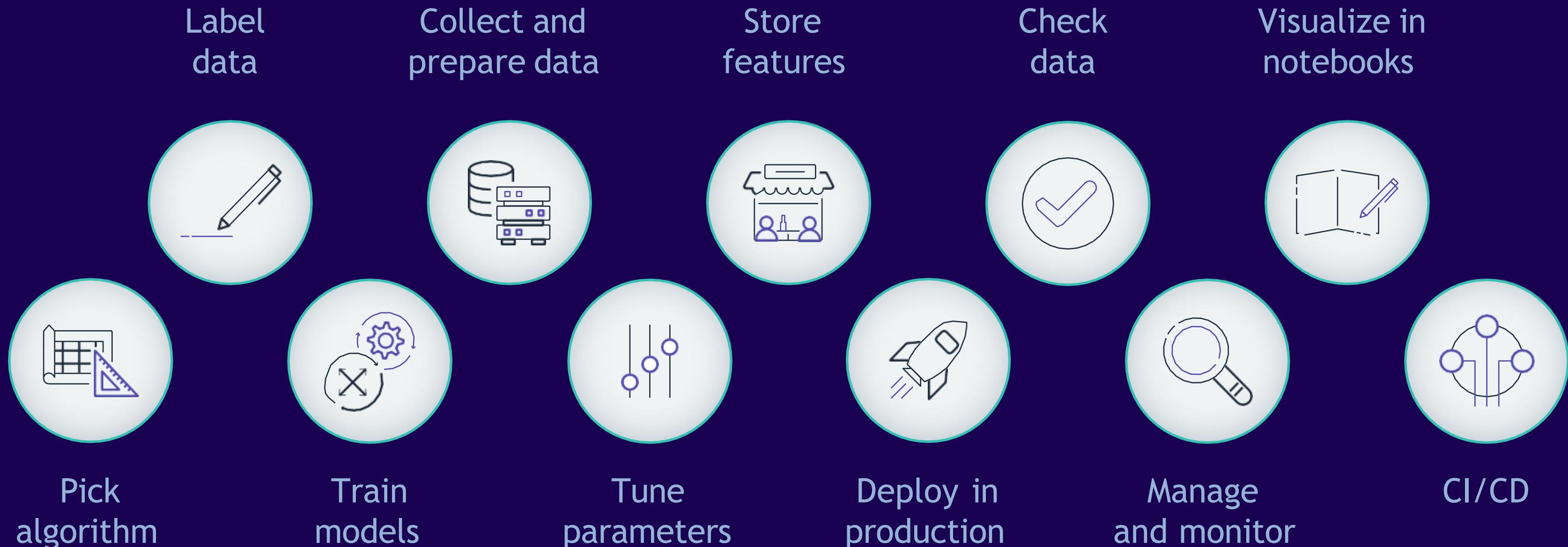
DOW JONES



SONY

 **GE Healthcare**

Amazon SageMaker: Built to make ML more accessible



SageMaker Studio IDE

Amazon SageMaker overview



Amazon SageMaker

Prepare →

SageMaker Ground Truth
SageMaker Data Wrangler
SageMaker Processing
SageMaker Feature Store

Build →

SageMaker Studio notebooks
Built-in and bring-your-own algorithms
Local mode
SageMaker Autopilot

Train & tune →

One-click training
SageMaker Experiments
Automatic model tuning
SageMaker Debugger
Managed spot training

Deploy & manage →

One-click deployment
Kubernetes & Kubeflow integration
Multi-model endpoints
Model Monitor
SageMaker Pipelines

SageMaker Studio
Integrated development environment (IDE) for ML

Amazon SageMaker overview

Amazon SageMaker

Prepare

SageMaker Ground Truth

Label training data
for machine learning

SageMaker Data Wrangler

Aggregate and prepare data
for machine learning

SageMaker Processing

Use built-in Python or bring your
own (BYO) R/Spark

SageMaker Feature Store

Store, update, retrieve,
and share features

Build

SageMaker Studio notebooks

Use Jupyter notebooks with
elastic compute and sharing

Built-in and BYO algorithms

Use dozens of optimized algorithms
or bring your own

Local mode

Test and prototype
on your local machine

SageMaker Autopilot

Automatically create machine
learning models with full visibility

Train and tune

One-click training

Distributed infrastructure
management

SageMaker Experiments

Capture, organize,
and compare every step

Automatic model tuning

Hyperparameter optimization

SageMaker Debugger

Debug training runs

Managed spot training

Reduce training cost by 90%

Deploy and manage

One-click deployment

Fully managed, ultralow latency,
high throughput

Kubernetes and Kubeflow integration

Simplify Kubernetes-based machine
learning

Multi-model endpoints

Reduce cost by hosting
multiple models per instance

SageMaker Model Monitor

Maintain accuracy
of deployed models

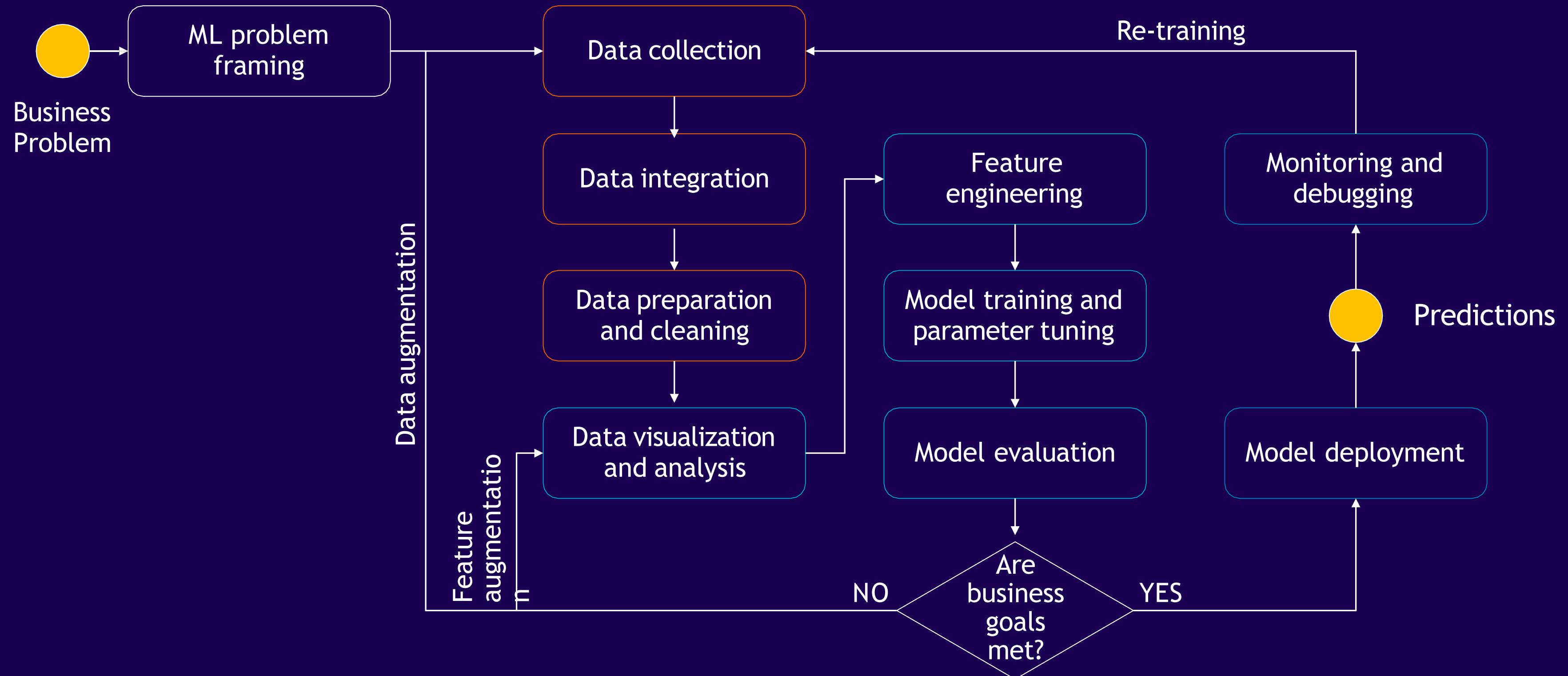
SageMaker Pipelines

Implement workflow orchestration
and automation

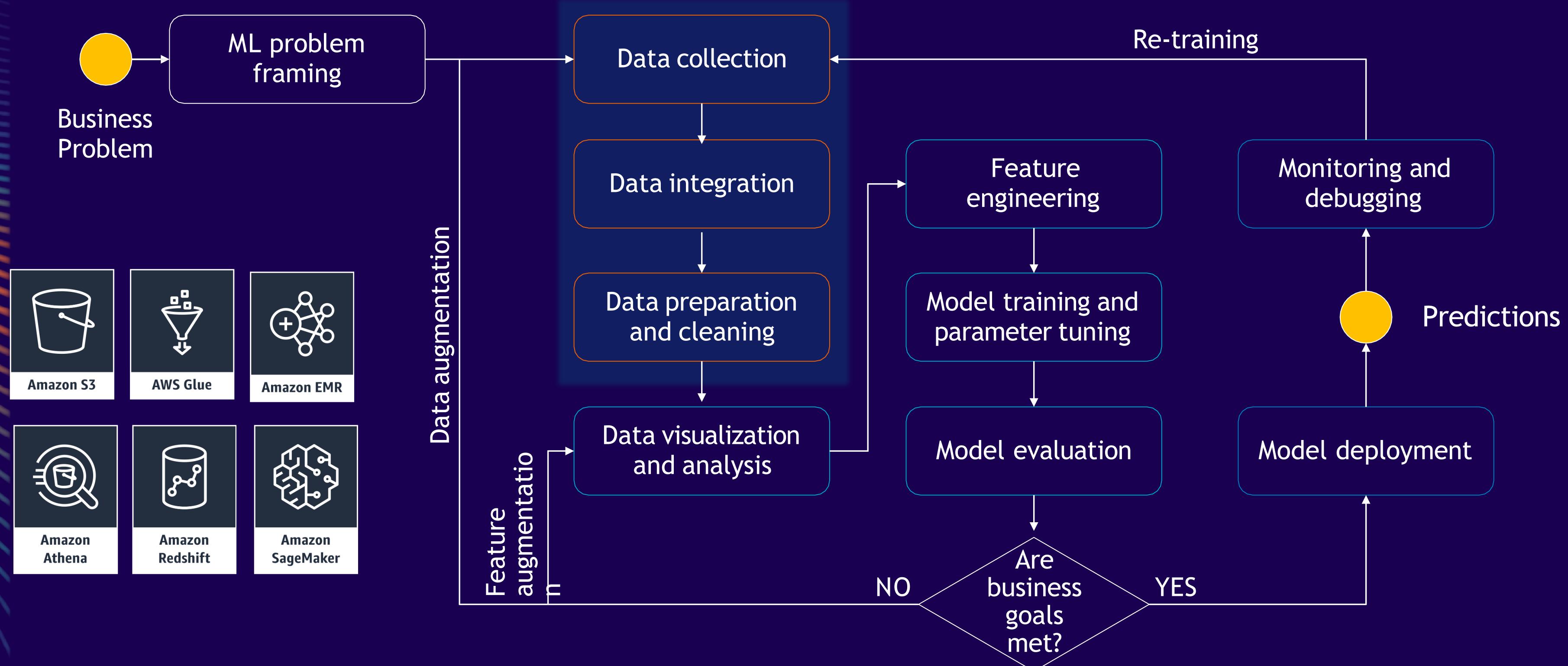
SageMaker Studio

Integrated development environment (IDE) for ML

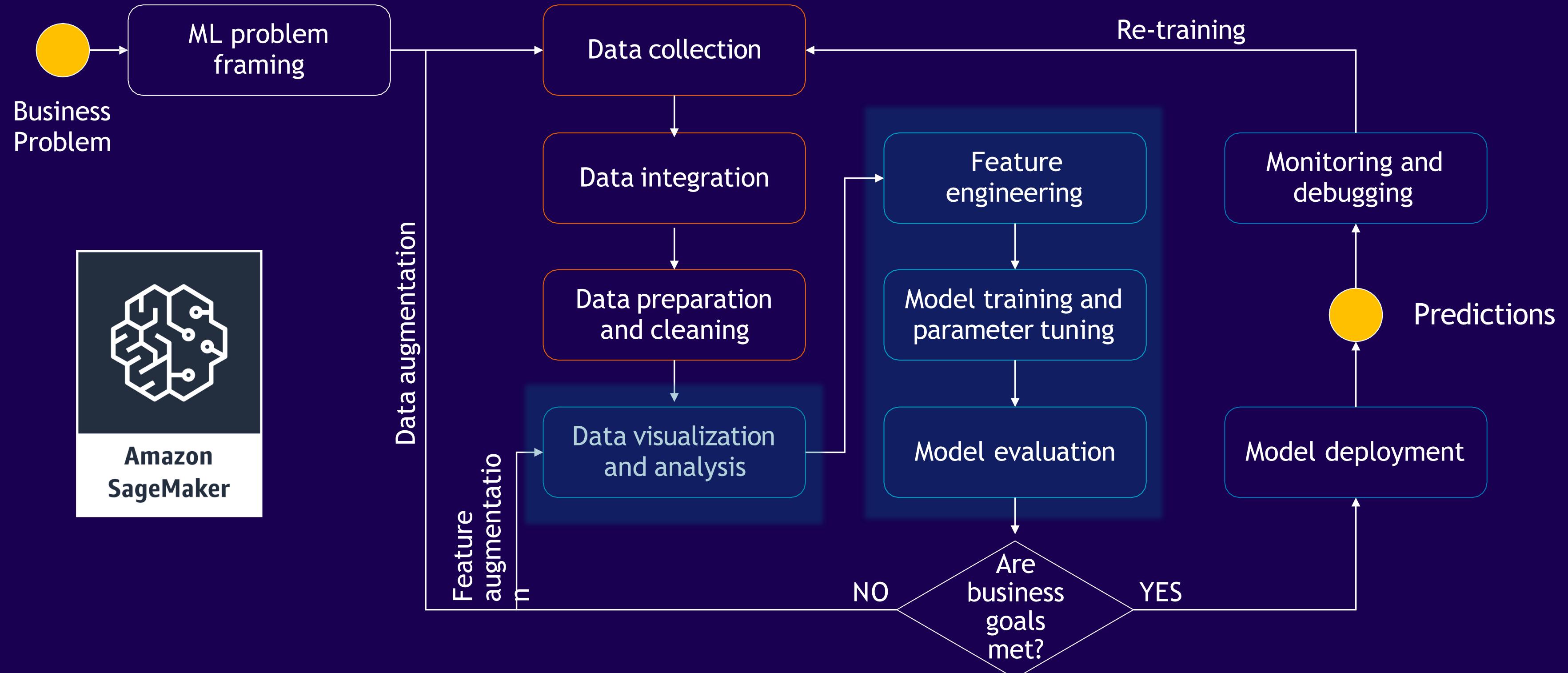
Machine learning cycle



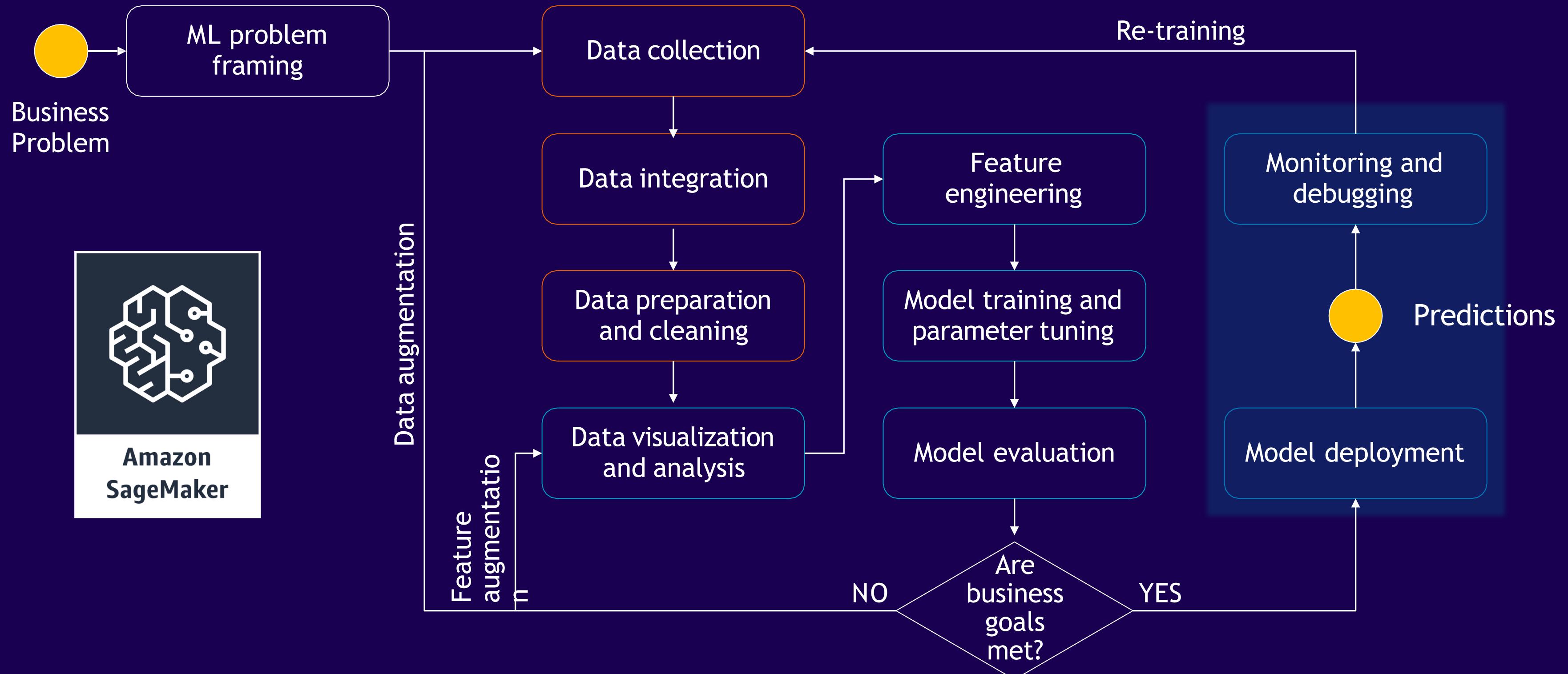
Manage data on AWS



Build and train models using SageMaker

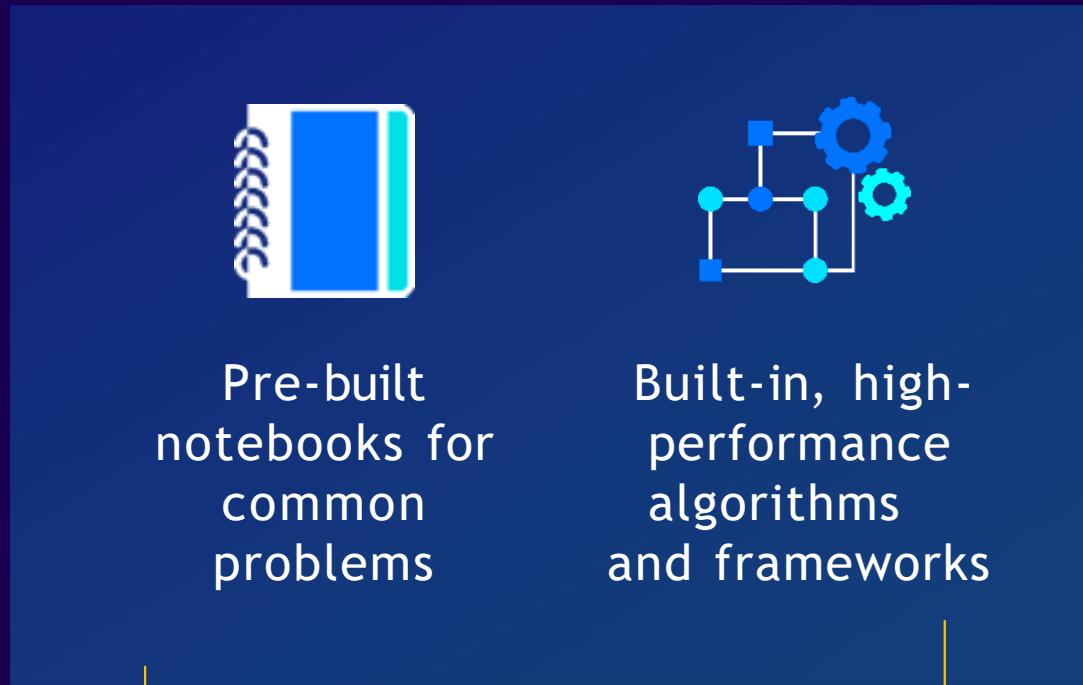


Deploy models using SageMaker



Amazon SageMaker

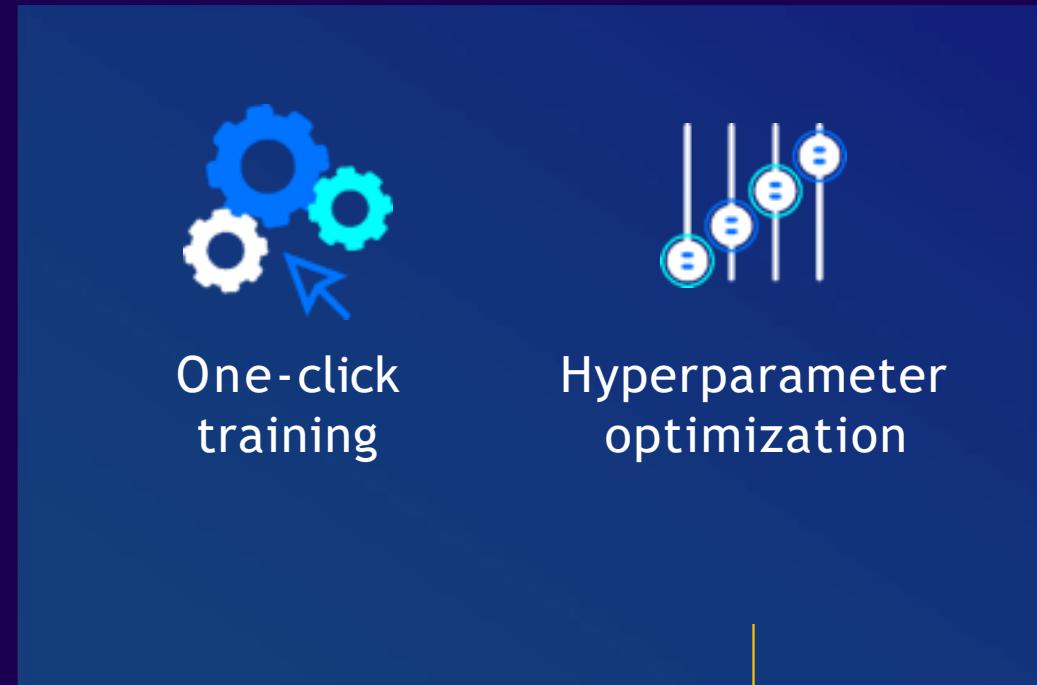
Build



Git integration
Elastic inference

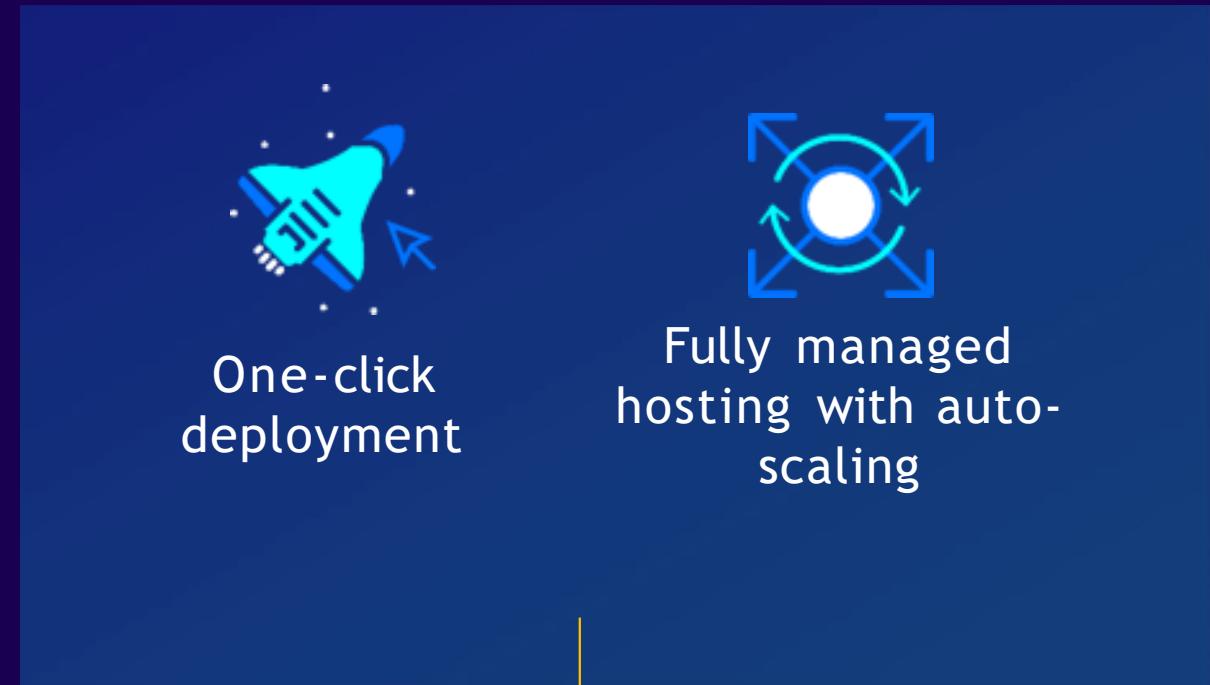
New built-in algorithms
scikit-learn environment
Model marketplace
Search

Train



P3DN, C5N
TensorFlow on 256 GPUs
Resume HPO tuning job

Deploy



Model compilation
Elastic inference
Inference pipelines

Amazon SageMaker



Amazon SageMaker

Infrastructure, tools,
visual interfaces,
workflows, orchestration,
and collaboration

Business analysts

Make ML predictions
using a visual interface
with Amazon
SageMaker Canvas

Data scientists
Prepare data and build,
train, and deploy ML
models with
Amazon SageMaker
Studio

MLOps engineers

Deploy and manage
models at scale with
Amazon SageMaker
MLOps

Approve models for production

The screenshot shows the Amazon SageMaker Studio interface. On the left, the 'Components and registries' sidebar is open, showing a 'Model registry' section with a search bar and a list of recent model groups. The main area displays the 'Recommendations Model - Latin America' dashboard, which includes a table of model versions. A modal window titled 'Update model version status' is open over the table, allowing the user to change the status of a selected version from 'Rejected' to 'Approved'. The modal also includes a comment field and a 'Cancel' or 'Update status' button.

Recommendations Model - Latin America

This Model group has recommendation models for the LA region.

Versions Settings

Name	Status	Step	Description	Status updated by	Modified on	Actions
version 6	Rejected	Staging	New model with SKL...	Jen Cabro	10/10/20	Open model version Update model version status...
version 5	Approved	Production	Model updated on 8/1...	Jen Cabro		Open model version
version 4	Approved	Archived	Model updated on 7/15...	Jen Cabro		Open model version
version 3	Approved	Archived	Model updated on 6/15...	Jen Cabro		Open model version
version 2	Approved	Archived	Model built on 5/15/20...	Jen Cabro	10/10/20	Open model version
version 1	Approved	Archived	Model built on 4/15/20...	Jen Cabro	10/10/20	Open model version

Update model version status

Name: **version 6**

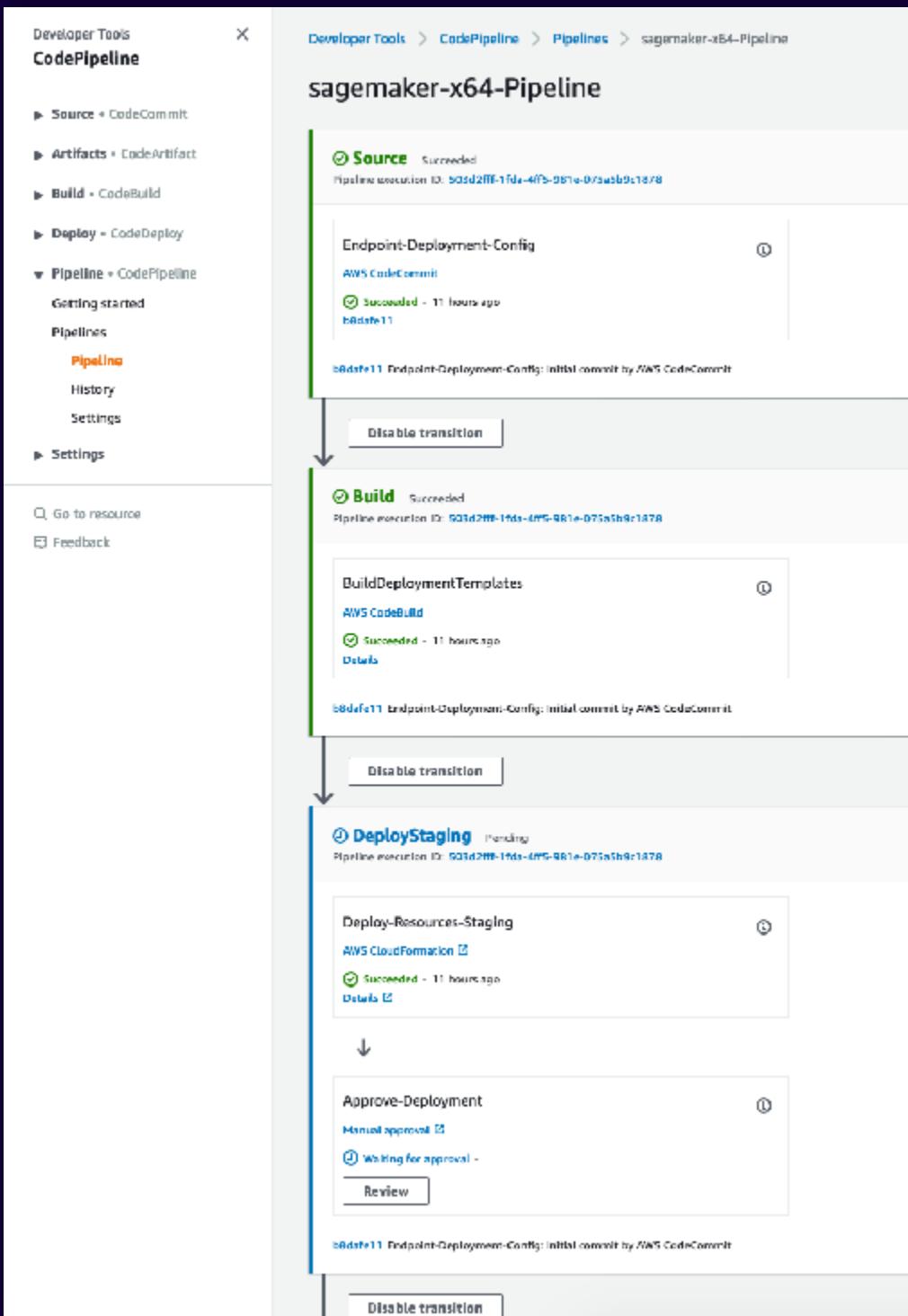
Update the model status and add comments. If this model group has a deployment pipeline, the new model version is deployed after it's approved.

Status: **Approved**

Comment - optional:
The model accuracy of this model looks good. Approved.

Cancel Update status

Deploy using fully managed CI/CD pipelines



View and compare evaluation metrics from training step

Amazon SageMaker Studio

File Edit View Run Kernel Git Tabs Settings Help

Components and registries

Choose the components or registry to view.

Model registry

Search Create model group

Name Modified on Actions

Recommendatio Model... Today

Recommendatio Model... Yesterday

Recommendatio Model... 10/22/20

Recommendatio Model... 09/12/20

half an minute ago

Launcher Recommendations Mode: - La... Compare model versions

Comparing model versions

Model group: modelGroup-5

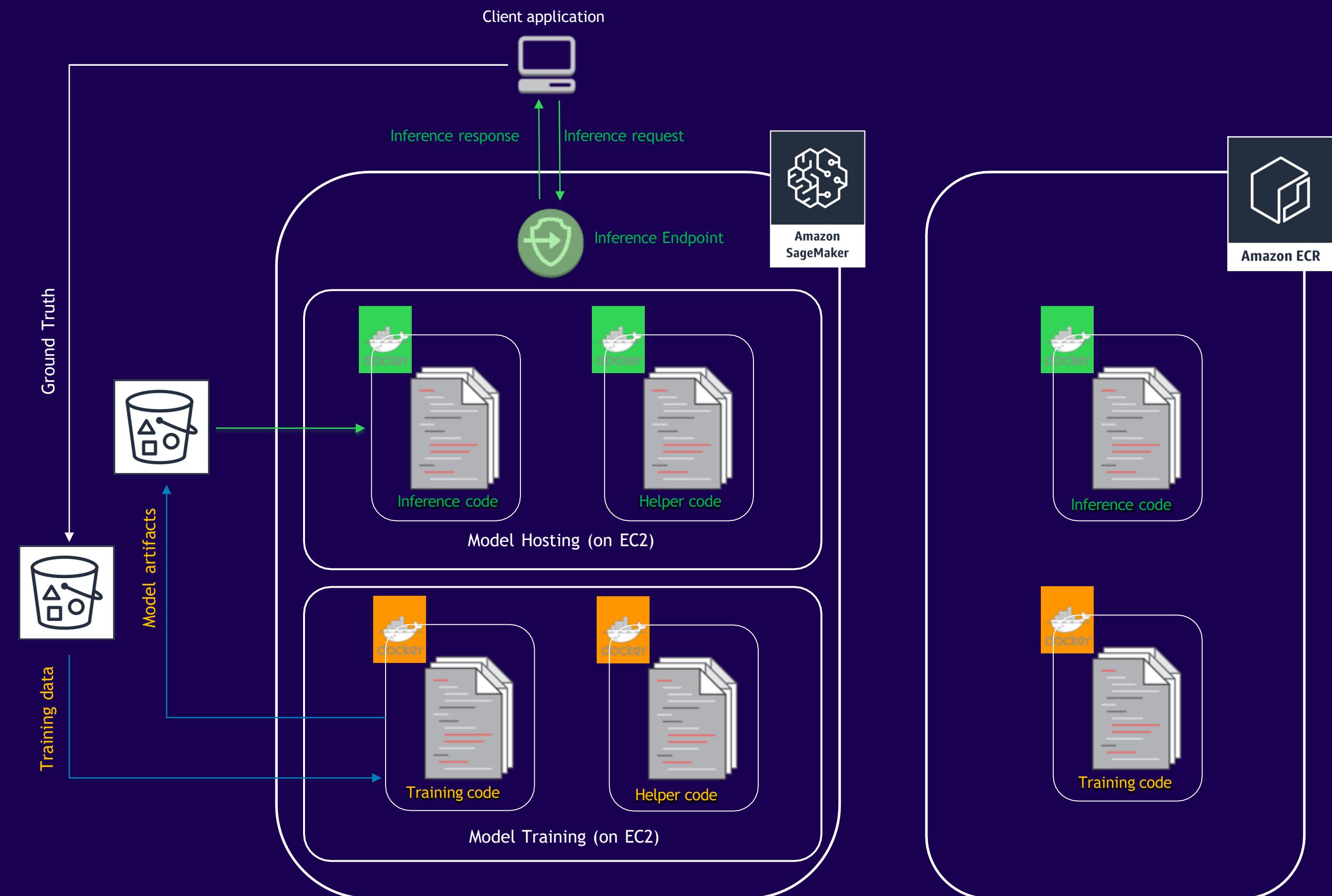
Model metrics 20

	version 3	version 4	Actions		
Confusion matrix					
ROC curve					
PRC					
Metric	Value	SD	Value	SD	
Recall	0.25	0.25	0.25	0.25	
Precision	0.25	0.25	0.25	0.25	
Accuracy	0.625	0.625	0.625	0.625	
Balanced accuracy	0.0	0.0	0.0	0.0	
Precision best constant classifier	0.0	0.0	0.0	0.0	
Accuracy best constant classifier	0.0	0.0	0.0	0.0	
True positive rate	0.25	0.25	0.25	0.25	
True negative rate	0.25	0.25	0.25	0.25	
False negative rate	0.25	0.25	0.25	0.25	
False positive rate	0.25	0.25	0.25	0.25	
AUC	1.0	0.0	1.0	0.0	

1 Python 3 (Data Science) | Idle

The Amazon SageMaker API

- Python SDK **orchestrating** all Amazon SageMaker activity
 - High-level objects for **algorithm selection, training, deploying, automatic model tuning**, etc.
 - Spark SDK (Python & Scala)
- AWS CLI: ‘*aws sagemaker*’
- AWS SDK: boto3, etc.



Amazon SageMaker

is DevOps ready

SECURITY

Security features to help you meet the strict security requirements of ML workloads

COMPLIANCE

Eligible for compliance with PCI, HIPAA, SOC 1/2/3, FedRAMP, and ISO 9001/27001/27017/27018

ML WORKFLOWS

Create automated workflows in minutes to support thousands of models

SCALABILITY

Train complex models with massive datasets

ORCHESTRATION

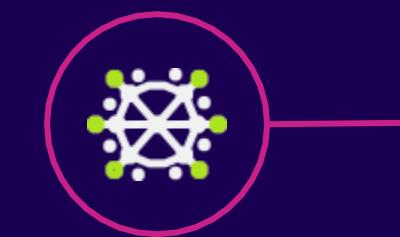
Automatically schedule and execute jobs with managed infrastructure

Amazon SageMaker key benefits

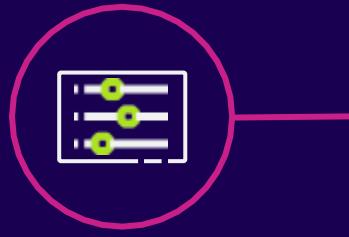
Most complete end-to-end ML service



Democratize ML innovation
Empower more groups of people, including business analysts



Prepare data at scale
Access, label, and process structured and unstructured data



Accelerate the ML lifecycle
Reduce training time from hours to minutes



Streamline ML processes
Automate and standardize MLOps processes



PROBLEM

3+ terabytes of data, 1,500+ hours of play time per week

Needed a solution for real-time stats

Lean team, no data science expertise

SOLUTION: NEXT GEN STATS

Engaged with Amazon ML Solutions Lab

Live data streamed to AWS from RFID tags on players and in game ball

Data processed in 100+ steps in under 1 second

ML models built on Amazon SageMaker make predictions in real time

IMPACT

Launched 20+ stats quickly with limited data science team

Sports announcers get interesting data points to engage fans

Machine Learning Marketplace

aws marketplace

Hello, julien ▾

Categories ▾ Delivery Methods ▾ Solutions ▾ Migration Mapping Assistant Your Saved List Partners Sell in AWS Marketplace Amazon Web Services Home Help

All Categories (61 results) showing 1 - 10

1 2 3 4 5 6 7 ►

Categories
All Categories
Infrastructure Software (27)
Business Software (7)
Machine Learning (61)
Filters
Clear all filters
Vendors
<input type="checkbox"/> RocketML (15)
<input type="checkbox"/> Sensifai (9)
<input type="checkbox"/> Intel® AI (7)
<input type="checkbox"/> Peak (5)
<input type="checkbox"/> Outpace Systems (5)
<input type="checkbox"/> improve.ai (4)
<input type="checkbox"/> H2O.ai (4)
<input type="checkbox"/> Dimensional Mechanics (4)
<input type="checkbox"/> TIBCO Software Inc. (3)
<input type="checkbox"/> bigfinite (1)
Show more

TIBCO® Data Science

Text Similarity Analyzer

★★★★★ (0) | Sold by TIBCO Software Inc.

Engineers word/document features on a corpus with NLP methods, and uses these features to compare new text to the corpus.

SENSIFAI

Automatic Image Tagging

★★★★★ (0) | Sold by Sensifai

Automatic Image Tagging and Recognition

PEAK

Demand Forecasting for Intermittent Data

★★★★★ (0) | Sold by Peak

An ensemble demand forecasting model, for intermittent data



Generative AI on AWS

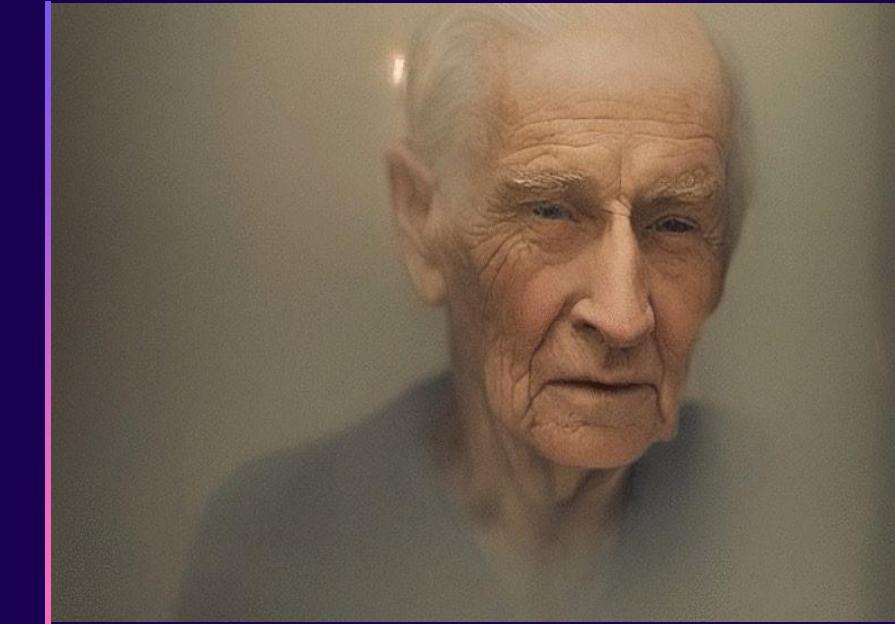
Self-managed and Managed ML workloads

GenAI is transforming AI

Image generation, transformation, upscaling



[Text to Image](#): Generated by Stable Diffusion 2.0 This interior does not exist

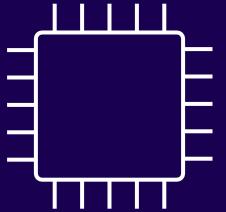


4x



Upscaling

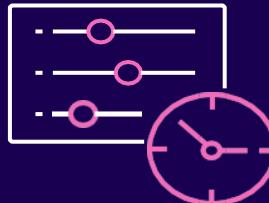
Challenges with training large-scale models



Hardware



Health checks



Orchestration



Data



Scaling up



Cost

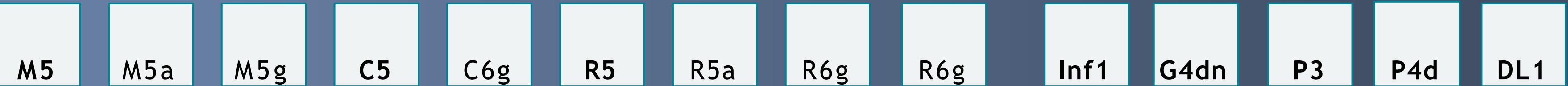
MACHINE LEARNING

Broadest and deepest compute infrastructure for AI/ML

Choice of CPUs, GPUs, and accelerators for your performance and budget needs

Traditional machine learning (ML)

Training and inference



Cascade Lake CPU
Skylake CPU

Habana Gaudi accelerators



EPYC CPU



Graviton CPU
Inferentia chip

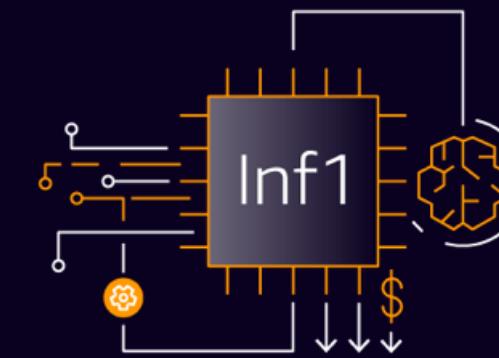


A100, V100, T4 GPUs

Broadest and deepest compute for AI/ML

Up to 70% lower cost per inference

AWS Inferentia1



Lowest cost inference in the cloud for running deep learning models

Traditional machine learning (ML)

Training and inference

M6a

M7g

C5

C7g

R5

R6a

R7g

Inf1

Inf2

G5g

Trn1

P4d
P4de

DL1

and more ..

Deep learning (DL)

Inference

Training



EPYC CPU



Graviton CPU
Inferentia and Trainium chip



A100, V100, T4 GPUs



AWS Nitro Enclaves



Elastic Fabric Adapter

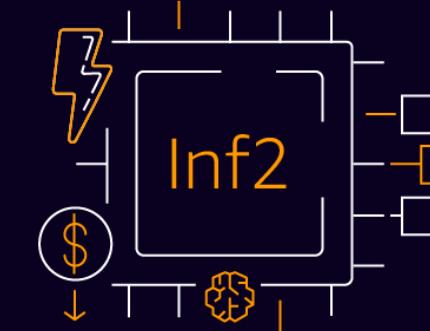


Amazon FSx
for Lustre

Broadest and deepest compute for AI/ML

Up to 40% better price
performance for Generative AI

AWS Inferentia2



High performance at the
lowest cost per inference for
LLMs and diffusion models

Traditional machine learning (ML)

Training and inference

M6a

M7g

C5

C7g

R5

R6a

R7g

Inf1

Inf2

G5g

Trn1

P4d
P4de

DL1

and
more
..

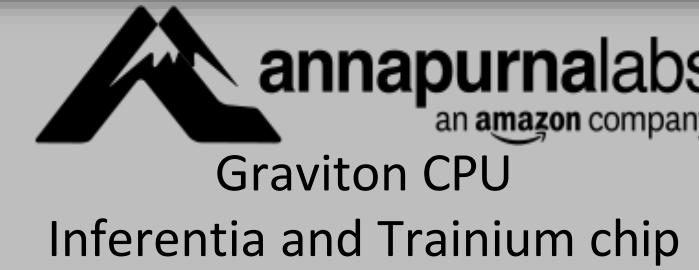
Deep learning (DL)

Inference

Training



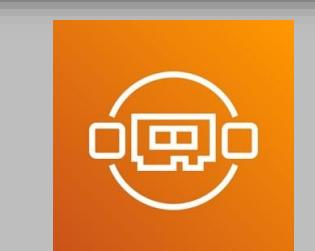
EPYC CPU



A100, V100, T4 GPUs



AWS Nitro Enclaves



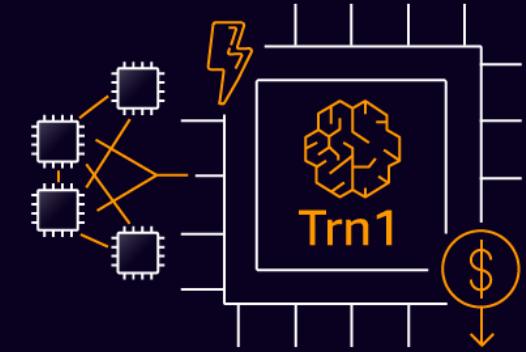
Elastic Fabric Adapter



Amazon FSx
for Lustre

Up to 50% cost-to-train savings

AWS Trainium



The most cost efficient for high-performance training of LLMs and diffusion models

Traditional machine learning (ML)

Training and inference

M6a

M7g

C5

C7g

R5

R6a

R7g

Inf1

Inf2

G5g

Trn1

P4d
P4de

DL1

and more ..

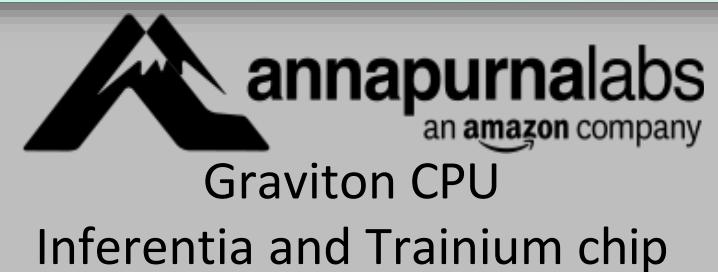
Deep learning (DL)

Inference

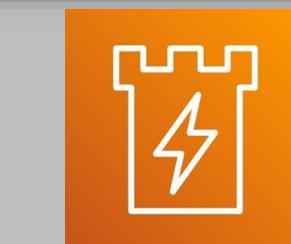
Training



EPYC CPU



A100, V100, T4 GPUs



AWS Nitro Enclaves



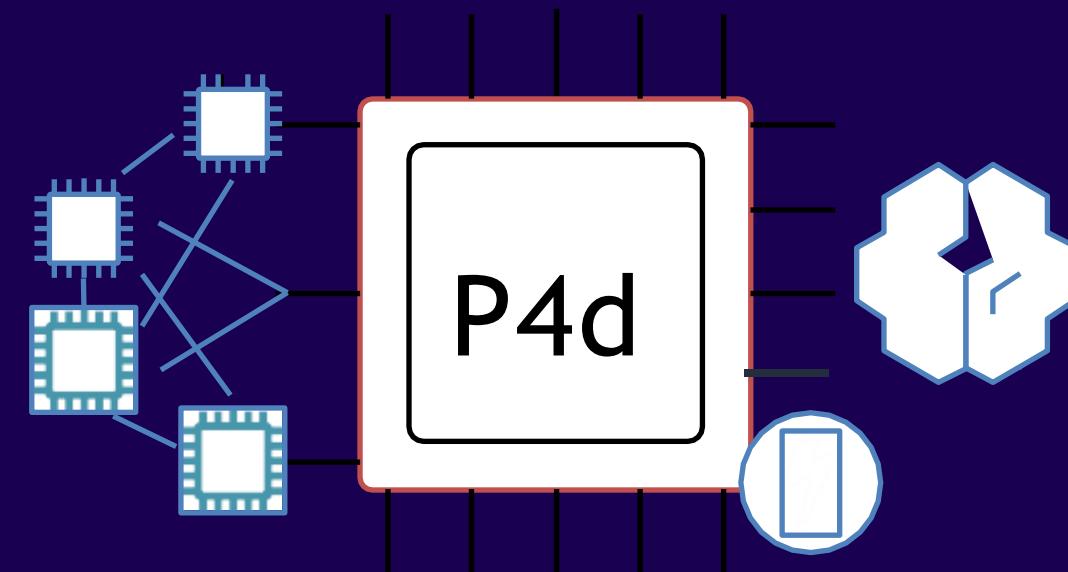
Elastic Fabric Adapter



Amazon FSx
for Lustre

Broadest and deepest compute for AI/ML

Introducing Amazon EC2 P4d instances



One of the most powerful GPU instances in the cloud

P4d instances

ML model with up to 60% lower cost to train, an average of 2.5x more deep learning performance, and 25% more GPU memory

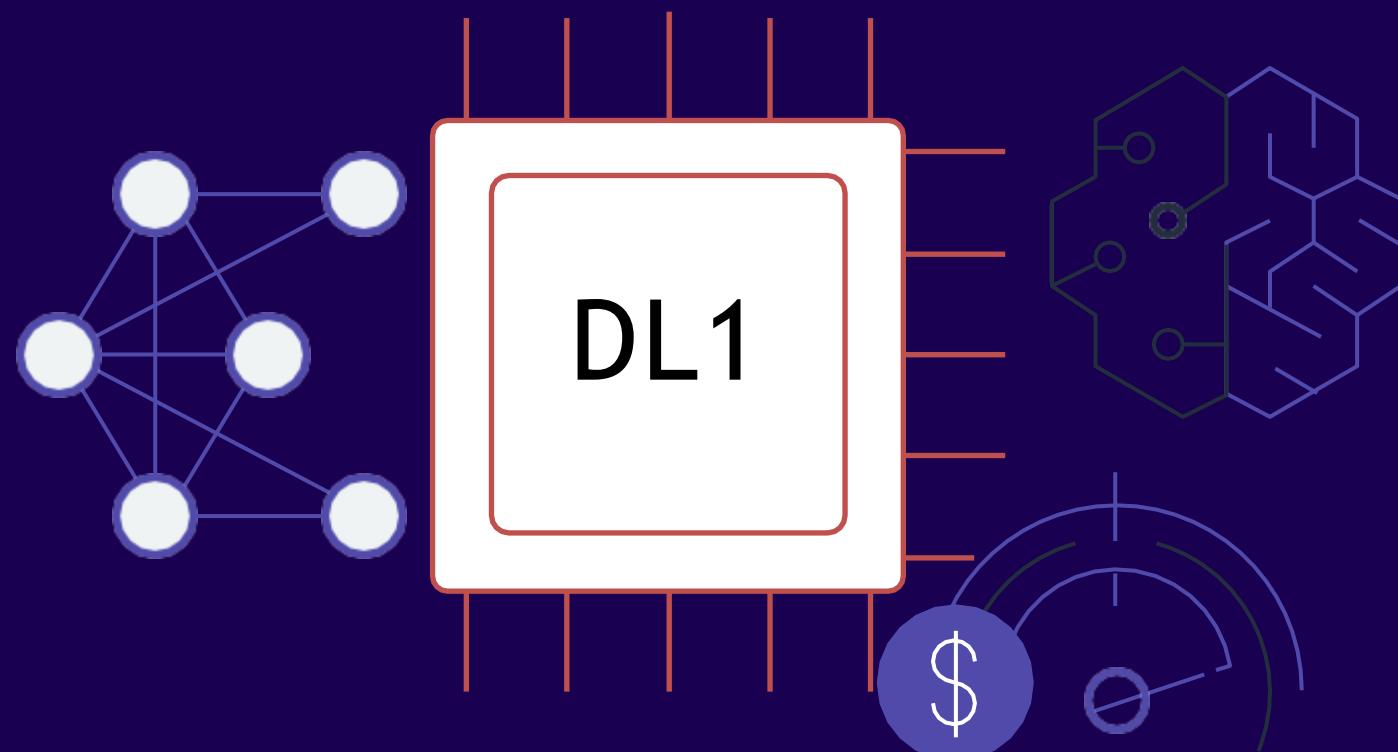
Powered by eight NVIDIA A100 GPUs and 400 Gbps of network bandwidth, and capable of 2.5 petaflops of performance

Deployed in UltraClusters consisting of thousands of tightly coupled GPUs, ideal for ML training and HPC

Introducing Amazon EC2 DL1 instances

Better price performance for training deep learning models

DL1 instances



Featuring up to eight Gaudi accelerators by Habana Labs (an Intel company)

Specifically built for training deep learning models

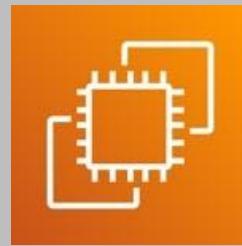
Up to 40% better price performance than the latest GPU instances

Custom software seamlessly integrated with TensorFlow and PyTorch

Get started easily using DLC, DL AMIs, or Amazon SageMaker

Launch DL1 instances via Amazon ECS and Amazon EKS for containerized ML applications

Broadest and deepest compute for AI/ML



Amazon EC2



AWS ParallelCluster



Amazon EKS



Amazon ECS



AWS Fargate

more



Amazon SageMaker

Self-Managed

Broadest and deepest compute services for AI/ML

Managed

M6a

M7g

C5

C7g

R5

R6a

R7g

Inf1

Inf2

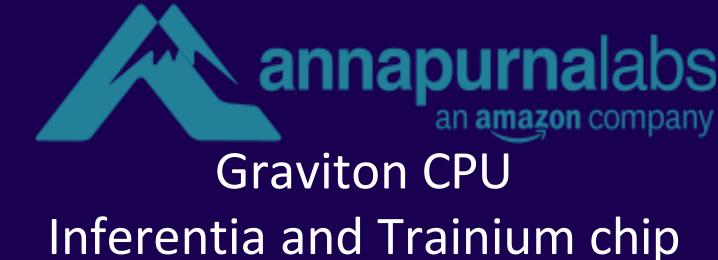
G5g

Trn1

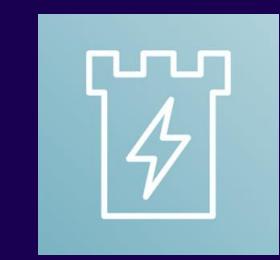
P4d
P4de

DL1

and
more
..



A100, V100, T4 GPUs



AWS Nitro Enclaves



Elastic Fabric Adapter

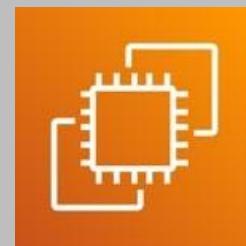


Amazon FSx
for Lustre

Broadest and deepest compute for AI/ML

AWS Deep Learning AMIs

AWS Deep Learning Containers



Amazon EC2



AWS ParallelCluster



Amazon EKS



Amazon ECS



AWS Fargate

more



Amazon SageMaker

Self-Managed

Broadest and deepest compute services for AI/ML

Managed

M6a

M7g

C5

C7g

R5

R6a

R7g

Inf1

Inf2

G5g

Trn1

P4d
P4de

DL1

and
more
..

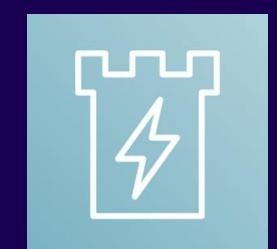


EPYC CPU

 **annapurnalabs**
an amazon company
Graviton CPU
Inferentia and Trainium chip



A100, V100, T4 GPUs



AWS Nitro Enclaves

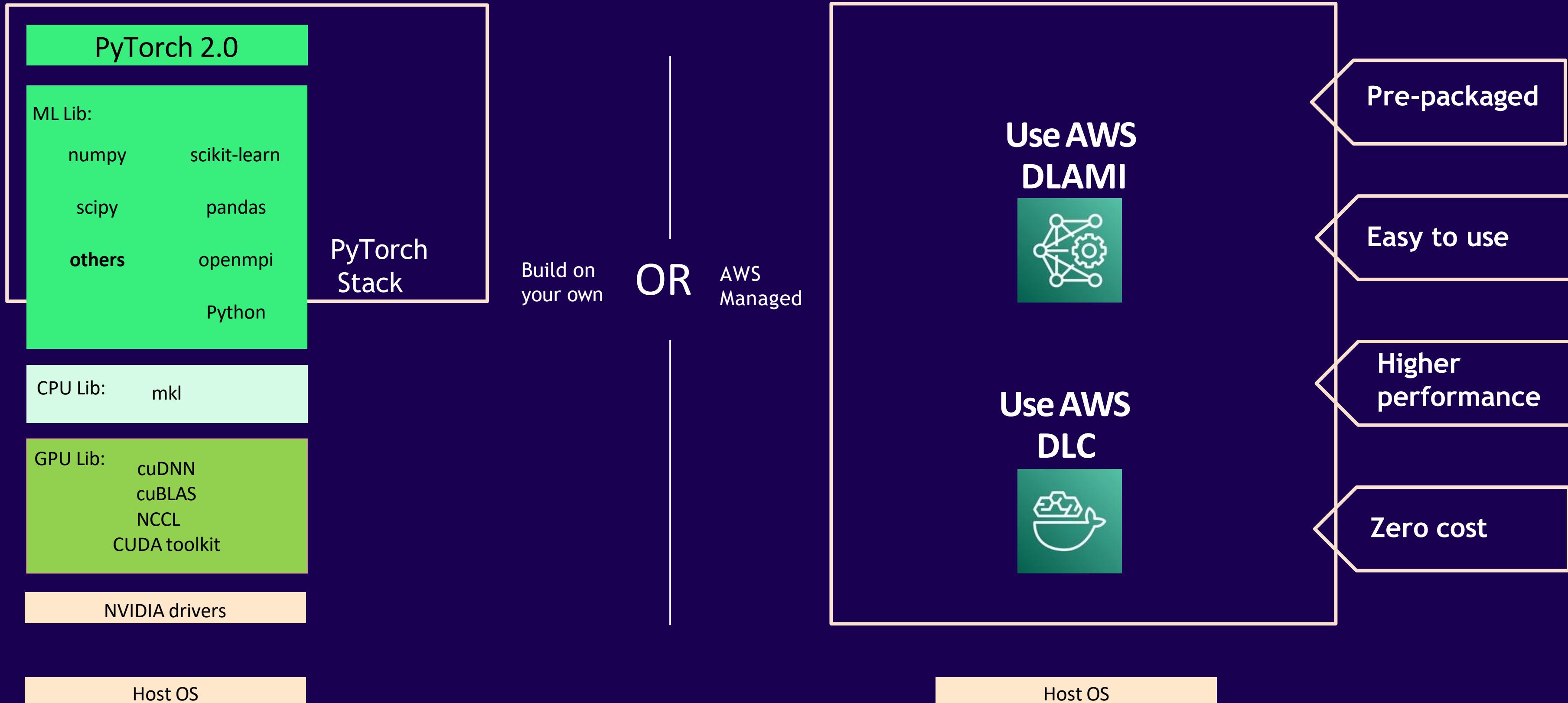


Elastic Fabric Adapter

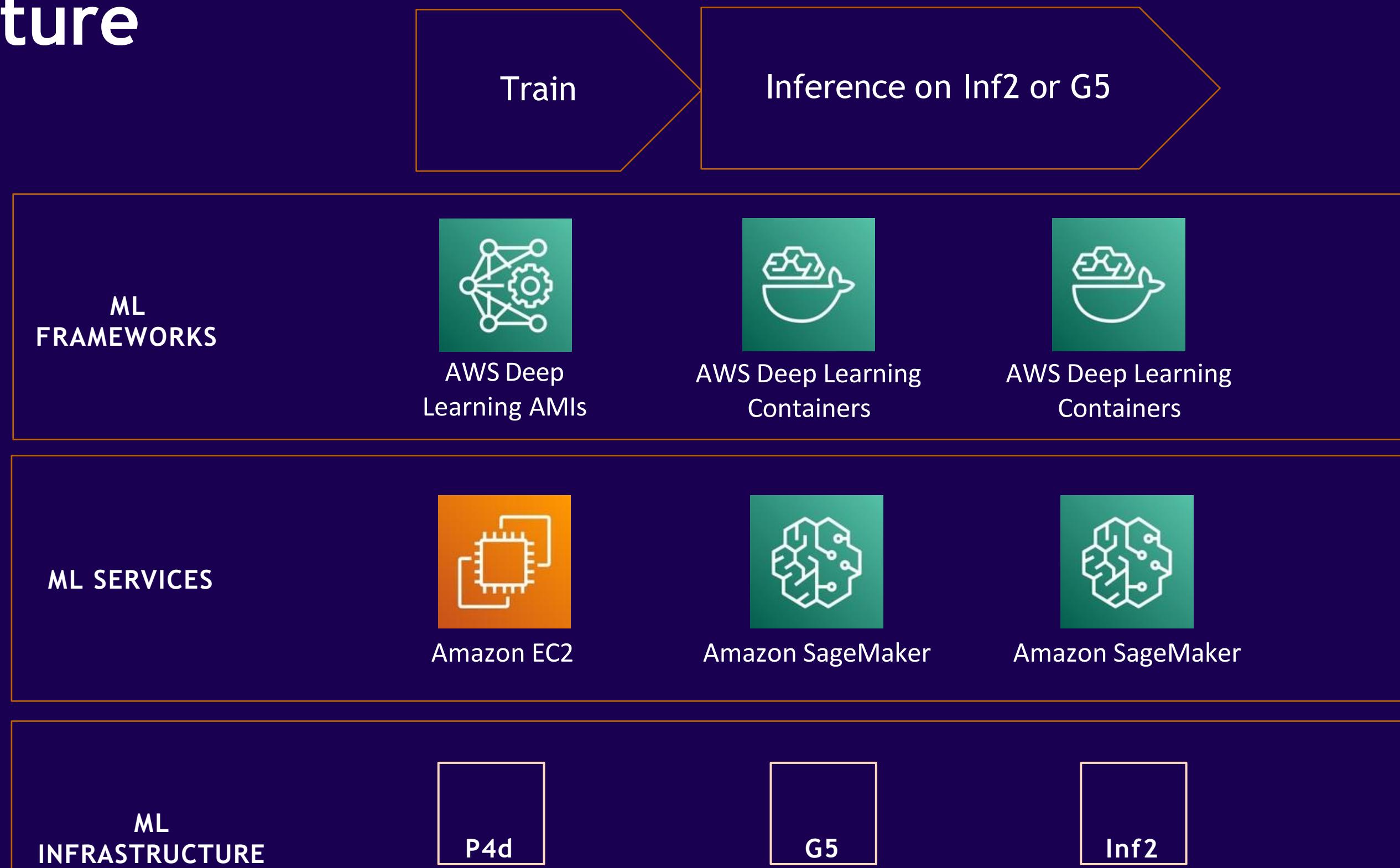


Amazon FSx
for Lustre

Why use DLAMI and DLC ?



Train and deploy Stable Diffusion model using Hybrid Architecture



Internet Of Things (IoT)

What is the Internet of Things (IoT)?



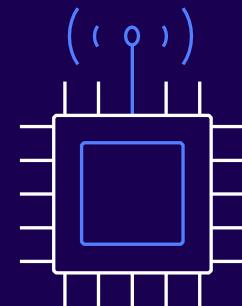
The Internet of Things (IoT) is where a system of integrated devices, such as appliances, watches, or features in a car, can be connected to various applications

These connections enable data to be transferred to and from devices in a bidirectional communication flow over a network

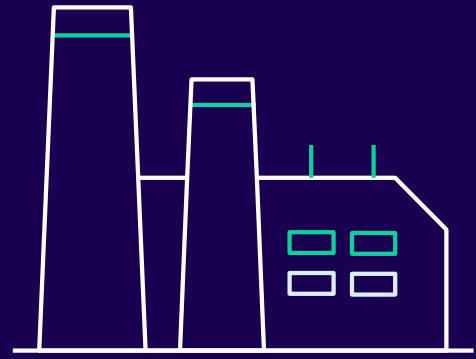
Challenges of managing “things”

Managing IoT devices poses a number of challenges

- Management and updates
 - Inconsistent or intermittent network connectivity
 - Remote devices that may not be physically accessible
 - Large fleets of devices in production
- Analytics
 - Low compute power, low-spec on-device resources
 - Devices may emit large quantities of streaming data



What customers are doing with AWS IoT



Improve the performance
and productivity of
industrial processes



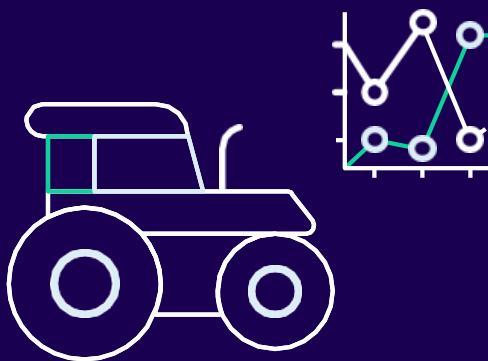
Remotely monitor
patient health &
wellness applications



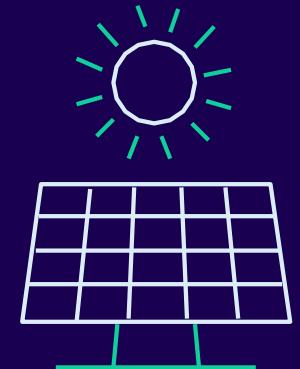
Track inventory
levels and manage
warehouse operations



Build smarter products &
user experiences in homes,
buildings, and cities



Grow healthier
crops with greater
efficiencies



Manage energy
resources more
efficiently

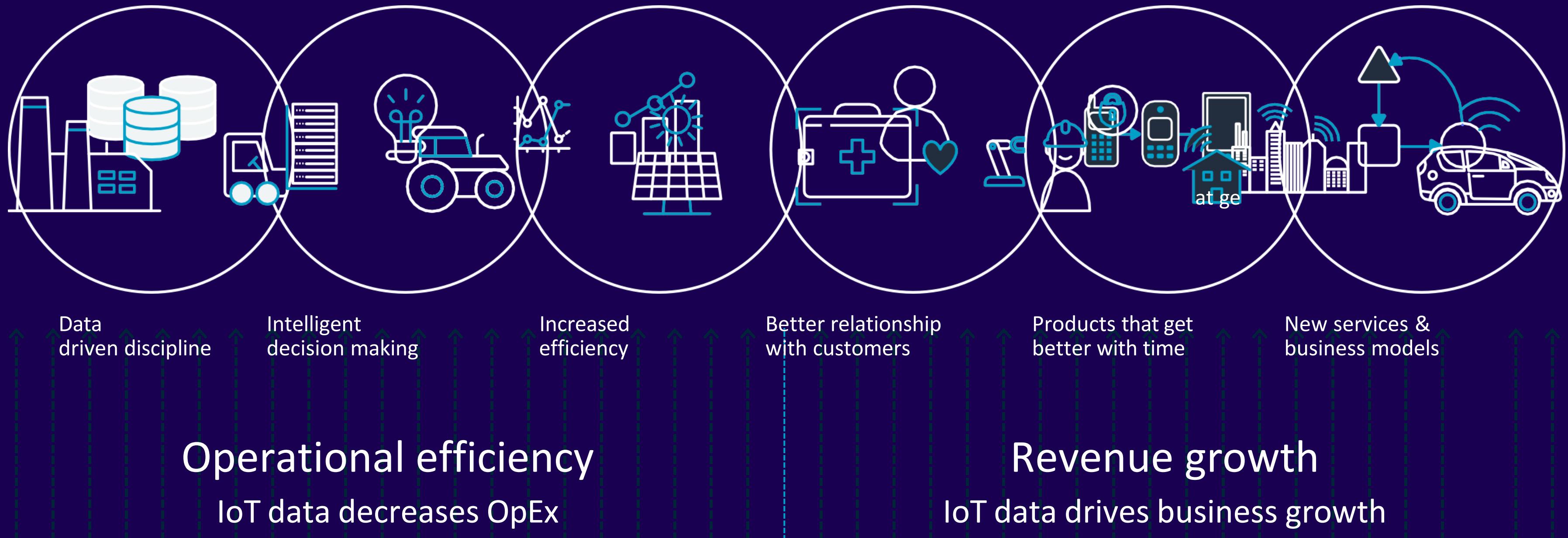


Transform transportation
with connected and
autonomous vehicles



Enhance safety in
the home, the office,
and the factory floor

What customers are doing with AWS IoT

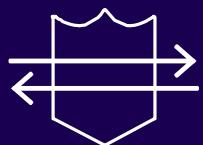


AWS IoT architecture



Analytics
Services

How can I make sense of my IoT data and take actions to solve business problems?



Connectivity &
Control
Services

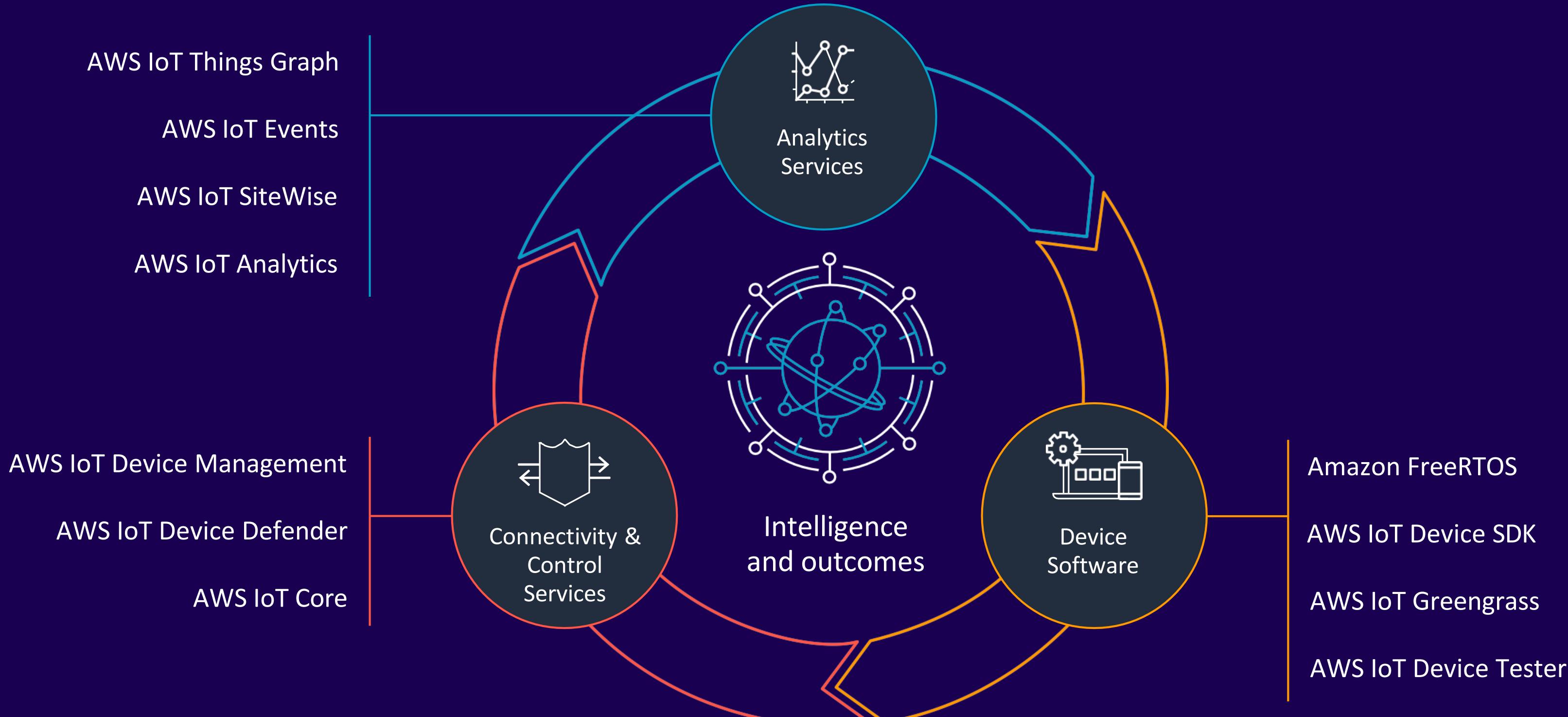
How can I connect, manage, and secure my devices at scale?



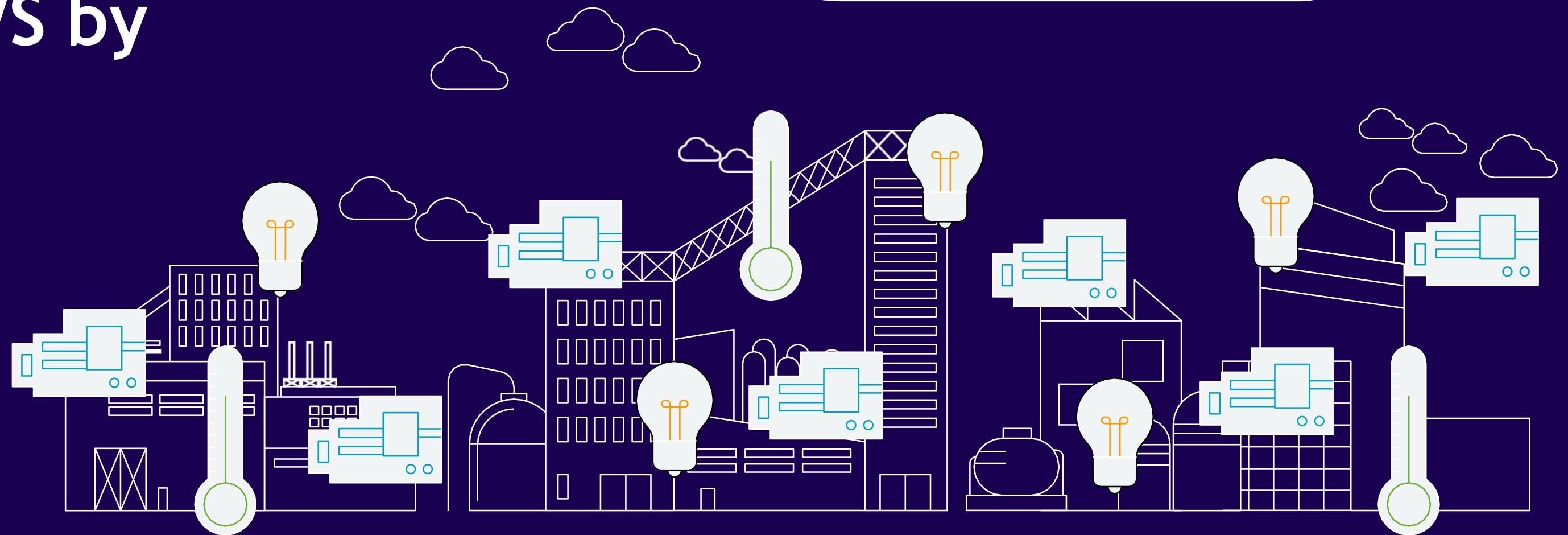
Device
Software

How can I build devices that operate at the edge that work with AWS by default?

IoT virtuous cycle



How can I build devices that operate at the edge and that work with AWS by default?

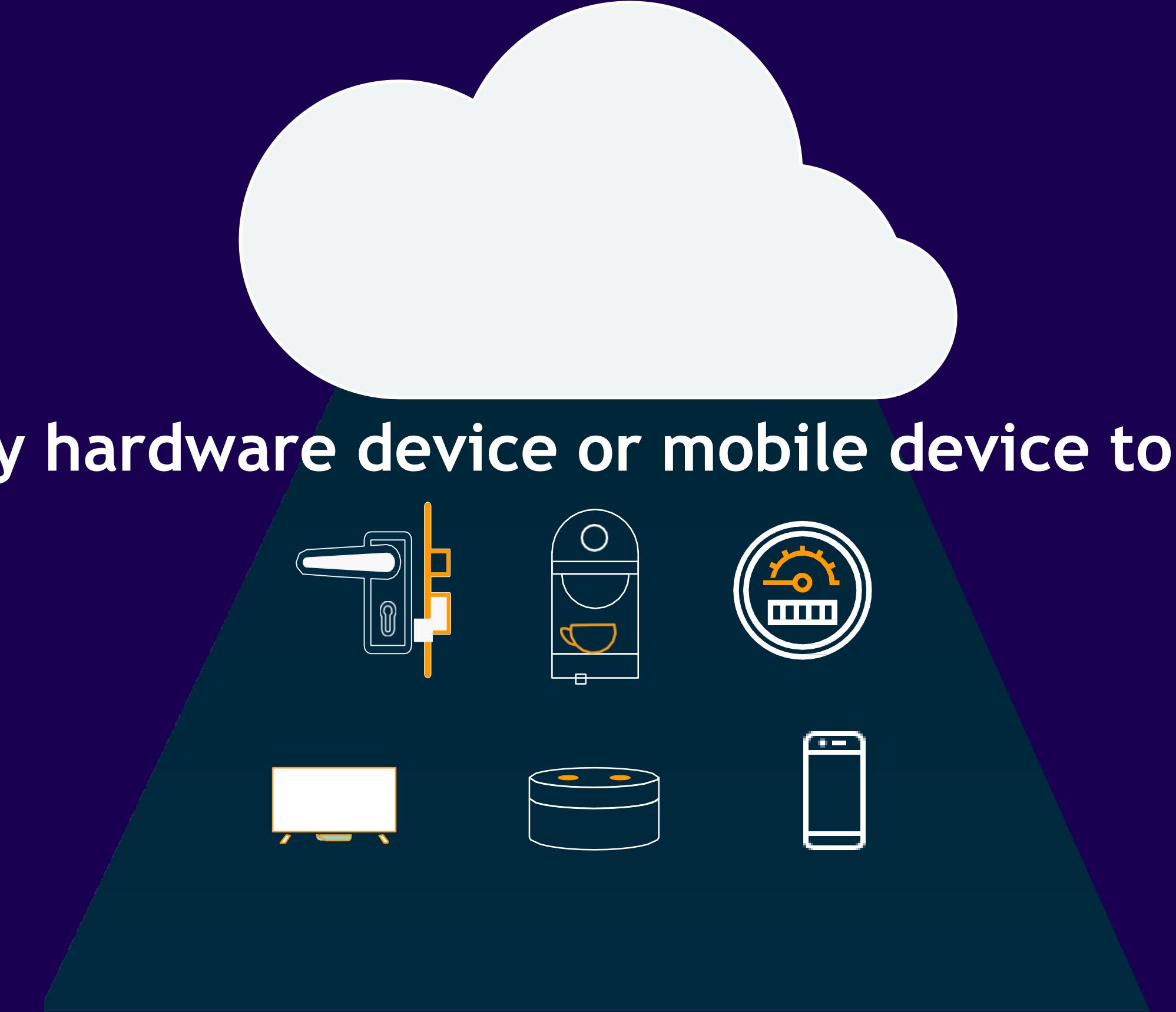


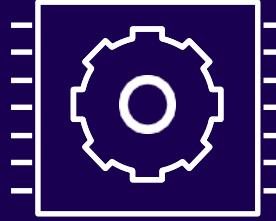
Device software

How can I securely connect any hardware device or mobile device to the cloud?



Device
software



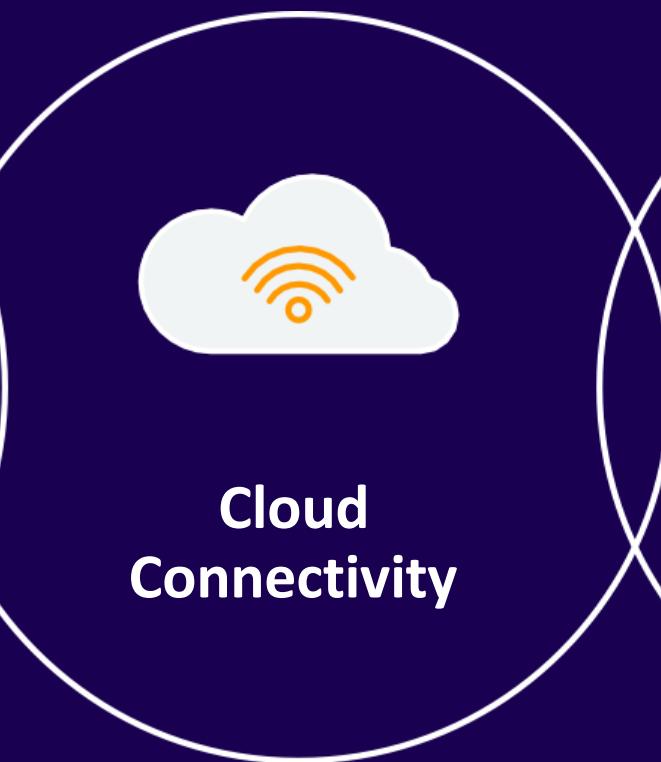


Amazon FreeRTOS Libraries



Local Connectivity

Communicate with AWS IoT Greengrass devices without a cloud connection



Cloud Connectivity

Easily collect data & take actions on microcontroller-based devices



Security

Secure device data and connections



OTA & Code Signing

Deploy security updates, bug fixes, and firmware updates to devices in the field



Device software

What customers are doing with Amazon FreeRTOS



Smart farms



Device
software



Connected home
appliances



Connected home
automation



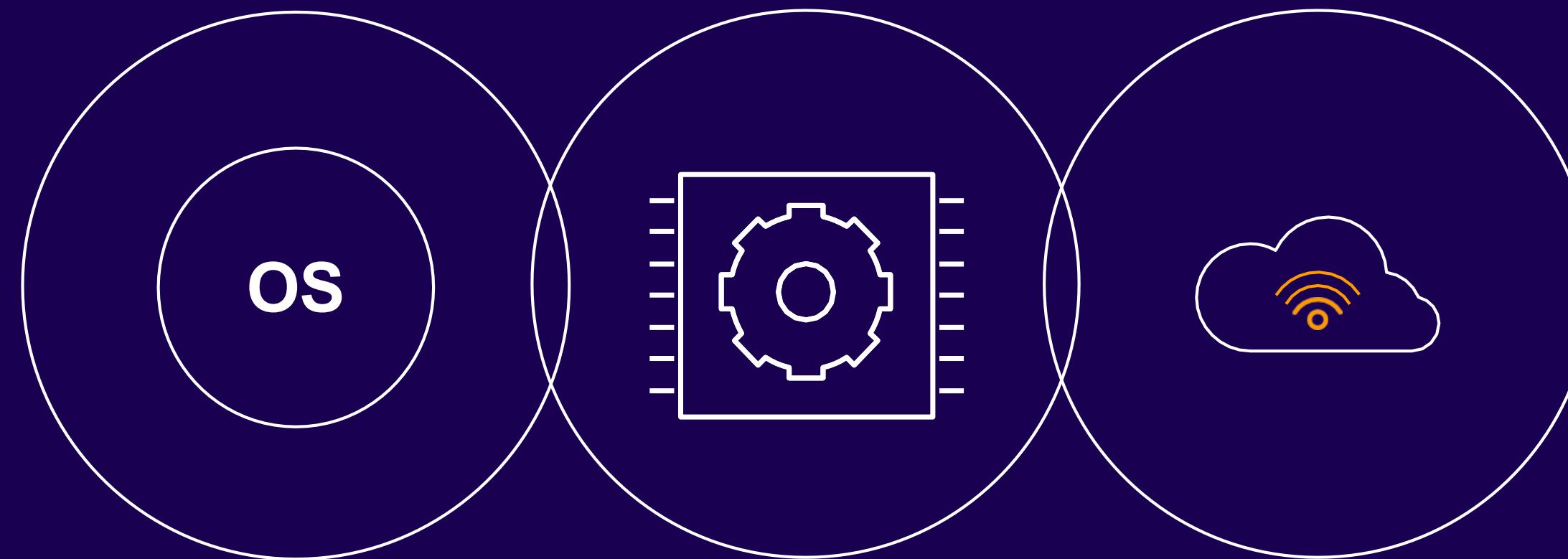
Smart building

Connect MCUs to AWS IoT using any operating system

AWS IoT Embedded C SDK enables you to connect microcontrollers or microprocessors using any operating system with the same IoT features as Amazon FreeRTOS



Device
software



Choose your Operating System

Use with any
operating system

Same code as Amazon FreeRTOS libraries

Receive all the same capabilities as
Amazon FreeRTOS

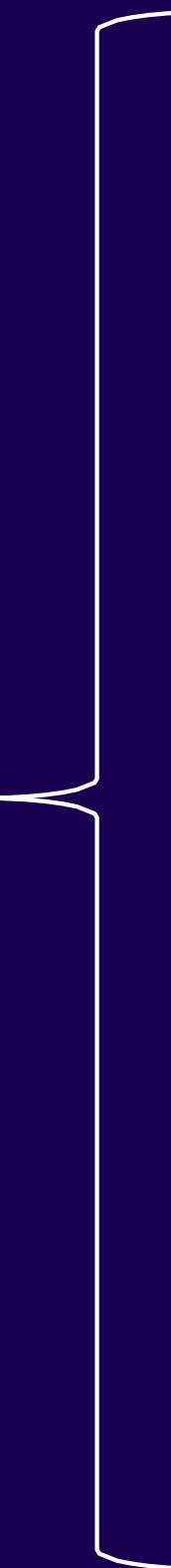
Individually Distributed

Easily integrate individual libraries
into your project

AWS IoT core: Rapid development



AWS IoT Core
Connect devices
to the cloud



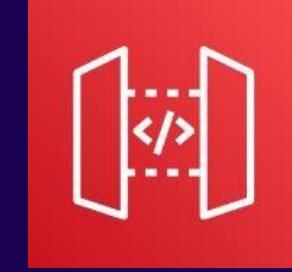
AWS Lambda
Run code in
response to events



Amazon DynamoDB
Predictable & scalable
NoSQL data store



Amazon Kinesis
Streaming
analytics



Amazon API Gateway
Build, deploy, and
manage APIs



Amazon Redshift
Petabyte-scale
data warehouse



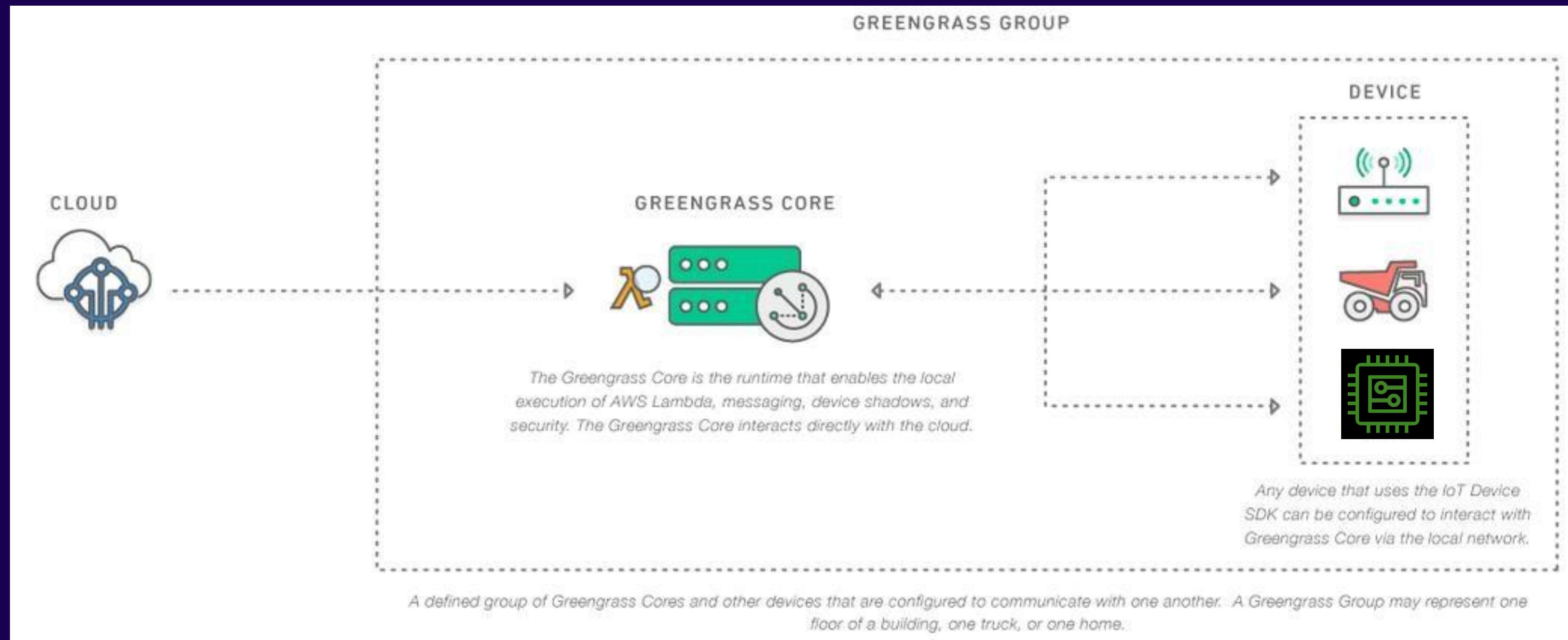
Amazon SNS
Mobile push
and notifications

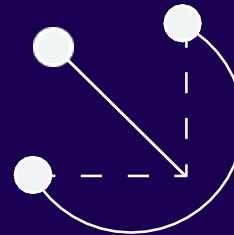


Amazon Cognito
User identity and data
synchronization

...and more

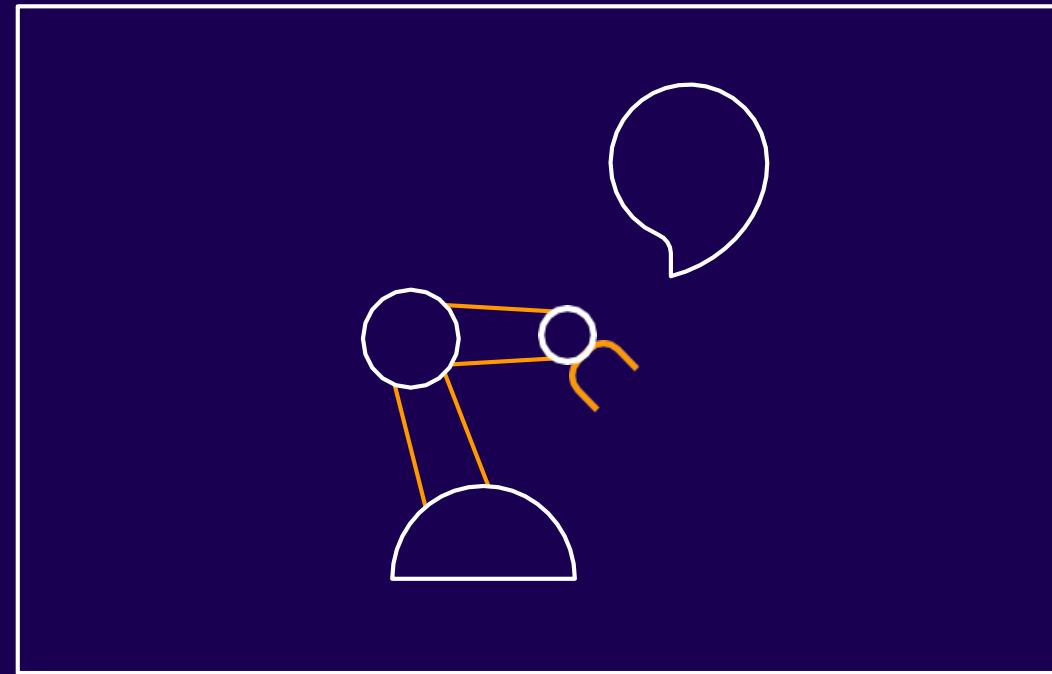
AWS IoT Greengrass



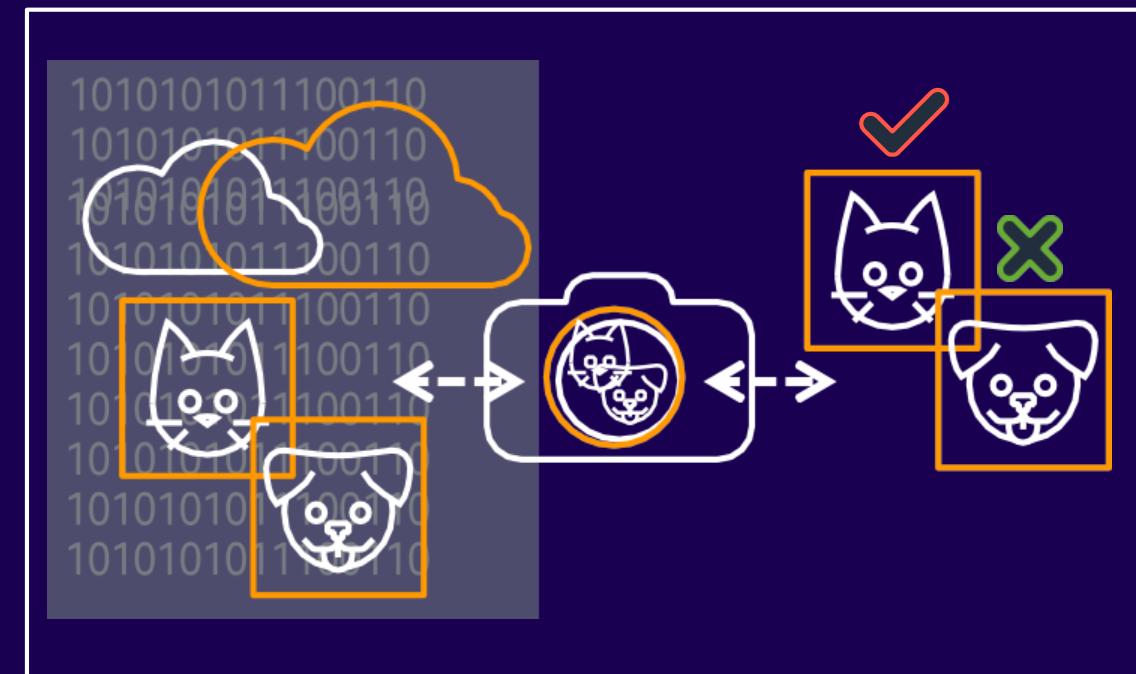


AWS IoT Greengrass

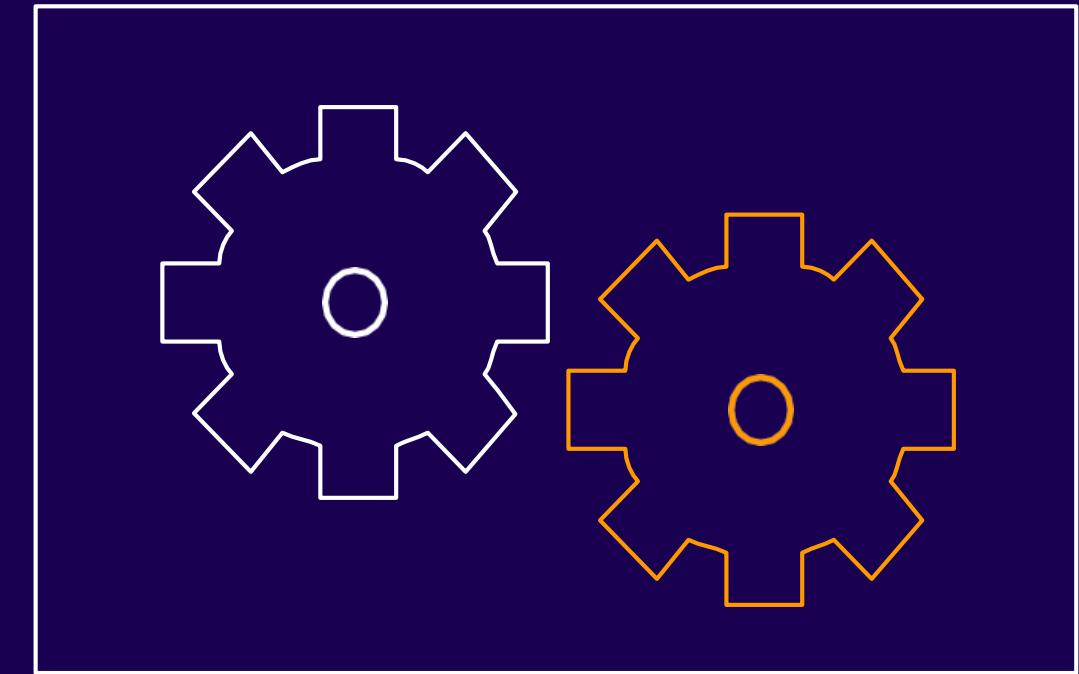
AWS IoT Greengrass extends AWS services onto your devices, so that they can act locally on the data they generate, while still taking advantage of the cloud.



Local Actions & Remote
Control



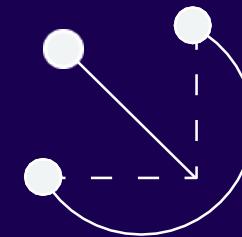
Machine Learning
Inference



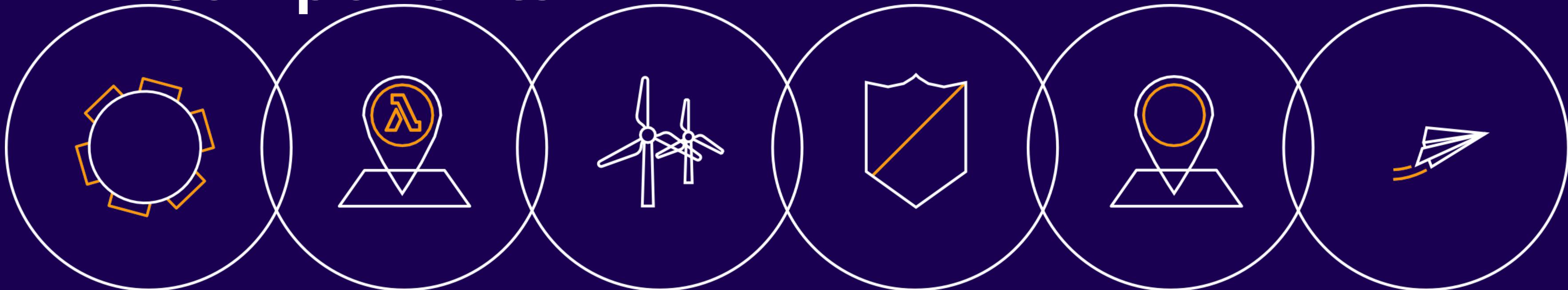
Extract,
Aggregate, Load



Device
software



AWS IoT Greengrass Foundational Components



Local Messages and Triggers

Enable device communication without a cloud connection

Local Actions

Simplify device programming with AWS Lambda

Data and State Sync

Operate devices offline & synchronize data when reconnected

Security

Mutual authentication & authorization between cloud and devices

Local Resource Access

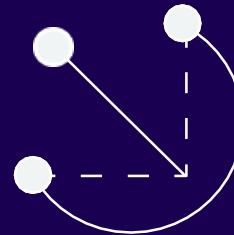
AWS Lambda functions can access & use local resources of a given device

Over the Air Updates

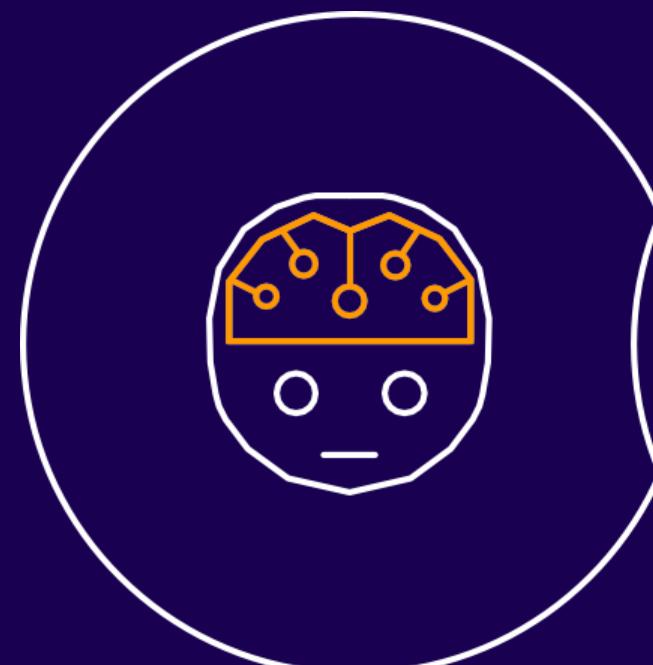
Easily update AWS IoT Greengrass Core



Device software



AWS IoT Greengrass Enhanced Capabilities

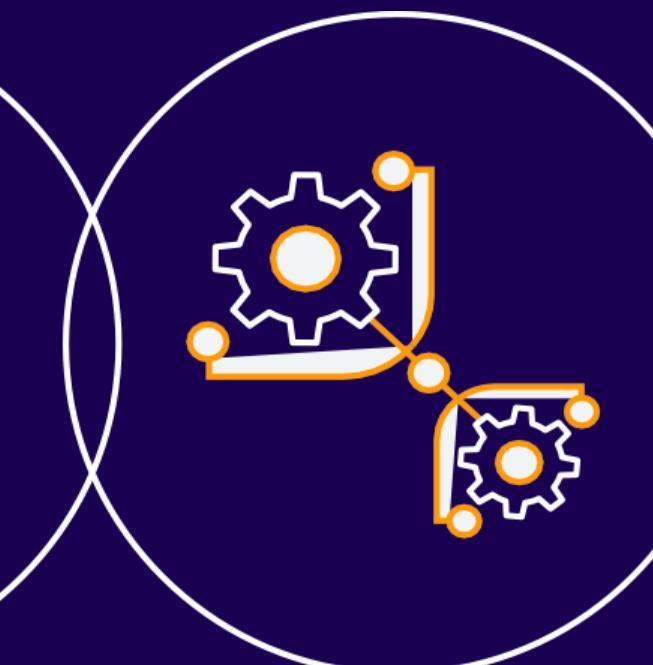


ML Inference

Perform ML Inference locally

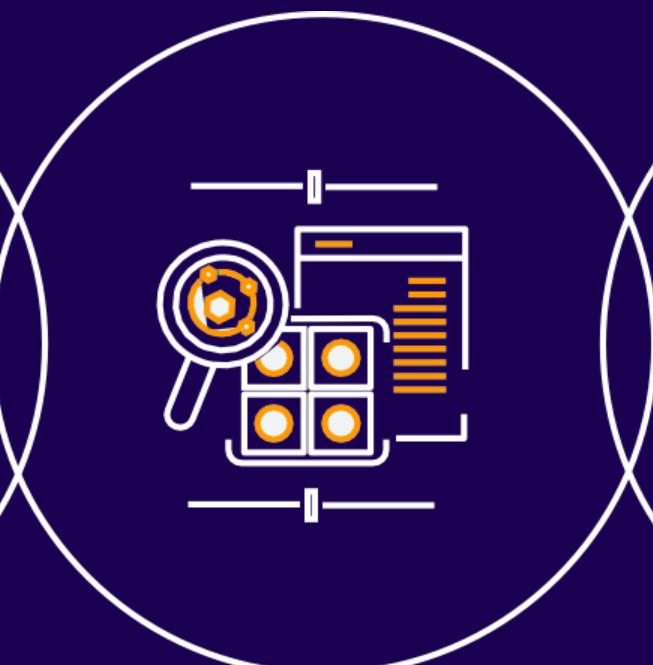


Device software



Connectors

Extend edge devices with connections to external services



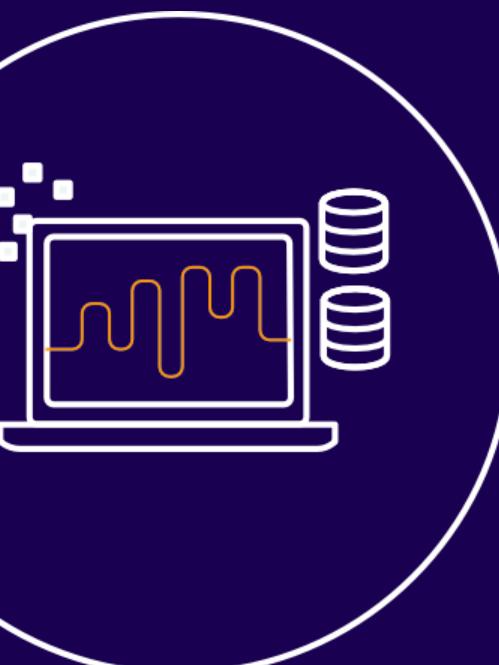
Secrets Manager

Deploy secrets to edge devices



Container Support

Use AWS Lambda, Docker, or a combination of both



Stream Manager

Collect, process, & export high-volume data streams from edge devices

Container Support for AWS IoT Greengrass

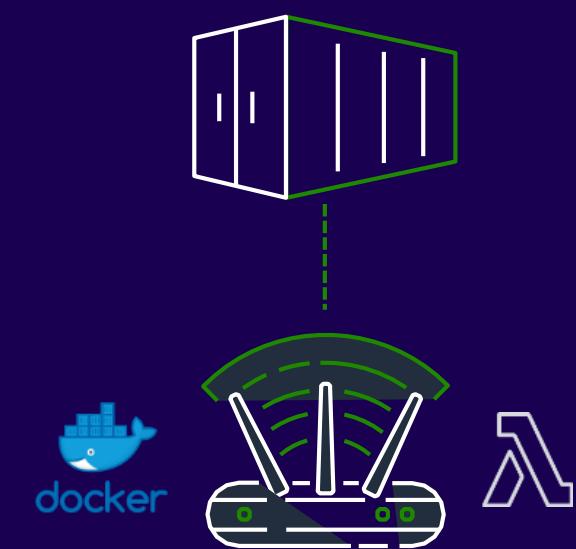
Deploy containers seamlessly to your edge devices.



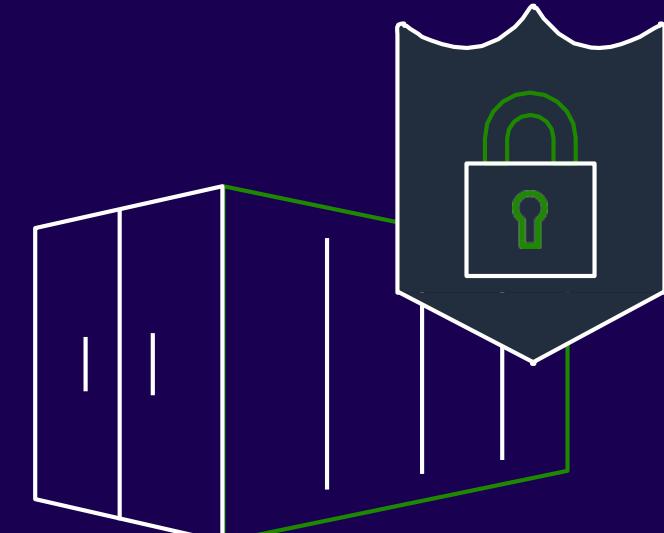
Move containers from the cloud to edge devices using AWS IoT Greengrass.



Device
software



Enables both Docker and AWS Lambda components to operate seamlessly together at the edge.



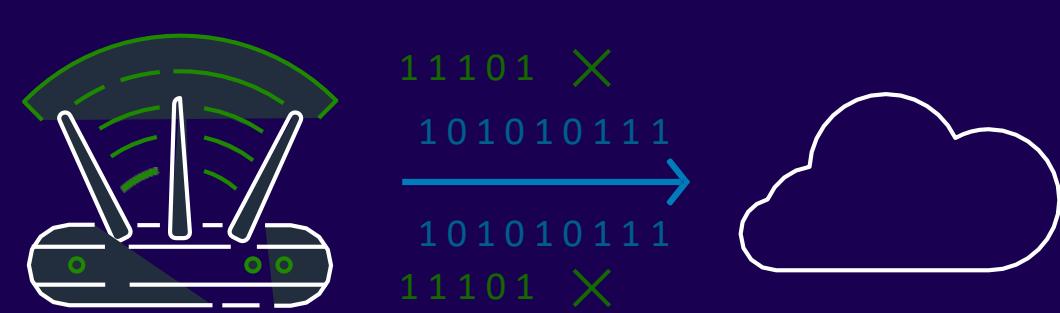
Use AWS IoT Greengrass Secrets Manager to manage credentials for private container registries.

Stream Manager for AWS IoT Greengrass

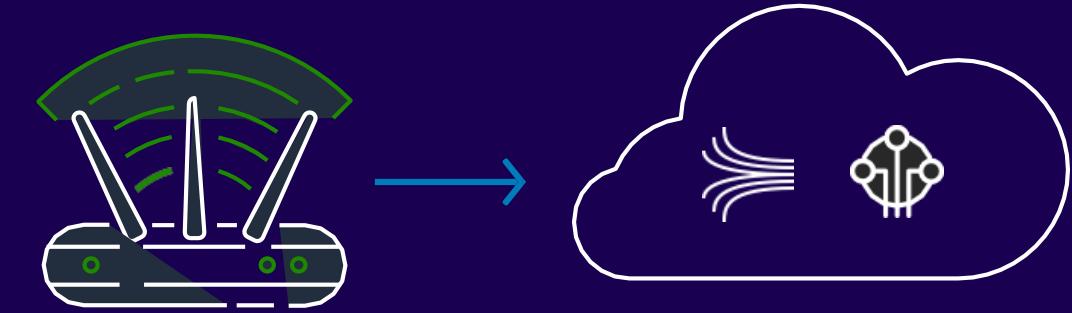
Preconfigure policies for collecting, processing, and exporting high-volume data streams on edge devices



Set policies for how data is processed and managed locally on devices with limited storage and compute.



Prioritize how data is streamed to the cloud with intermittent or limited connectivity.



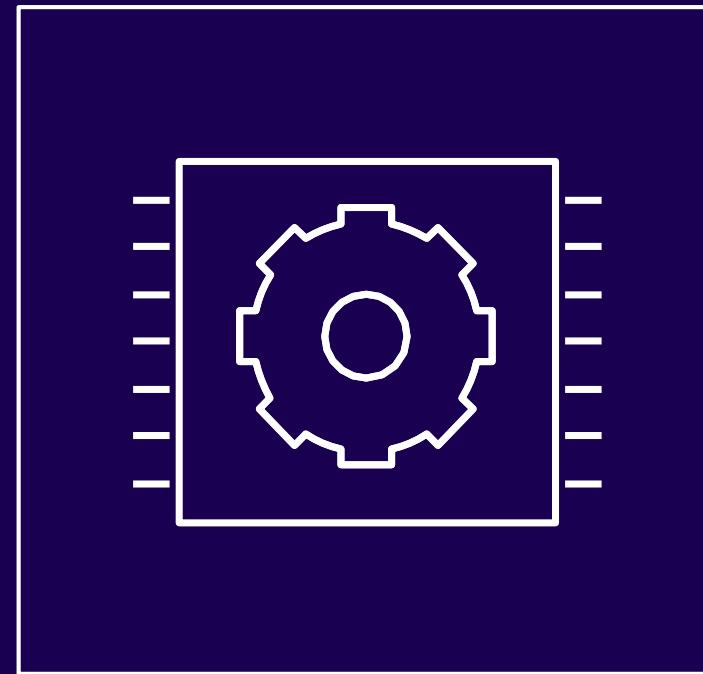
Stream data from edge devices directly to AWS services such as Amazon Kinesis and AWS IoT Analytics.



Device software

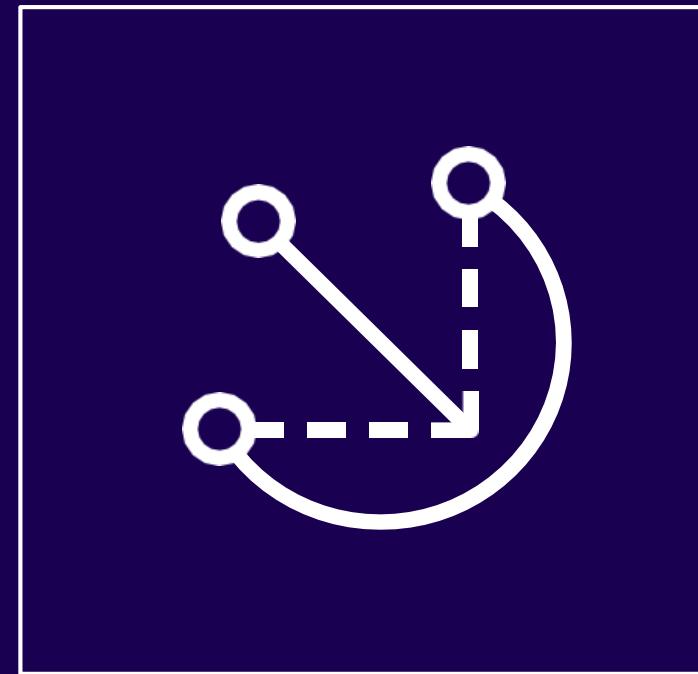
AWS IoT Device Tester

AWS IoT Device Tester is a test automation tool that lets you test Amazon FreeRTOS or AWS IoT Greengrass on your choice of devices.



AWS IoT Device Tester for Amazon FreeRTOS

Test if your device will run Amazon FreeRTOS and interoperate with AWS IoT services



AWS IoT Device Tester for AWS IoT Greengrass

Tests if the combination of device's CPU architecture, Linux kernel configuration, and drivers work with AWS IoT Greengrass



Device
software

Download AWS IoT Device Tester from
[Amazon FreeRTOS](#) and [AWS IoT Greengrass](#) product pages



AWS IoT Core

The AWS IoT Mobile and Device SDKs enable you to build your own connectivity capabilities to connect devices to AWS IoT Core.



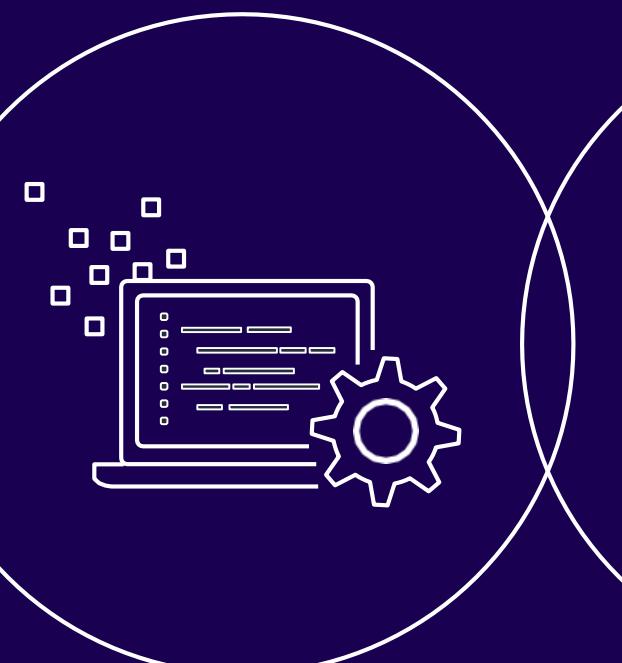
Abstract MQTT Protocol

Easily utilize Things, Topics, Shadows, and Jobs



Mobile Platform Support

Library for building mobile IoT applications supports both iOS and Android



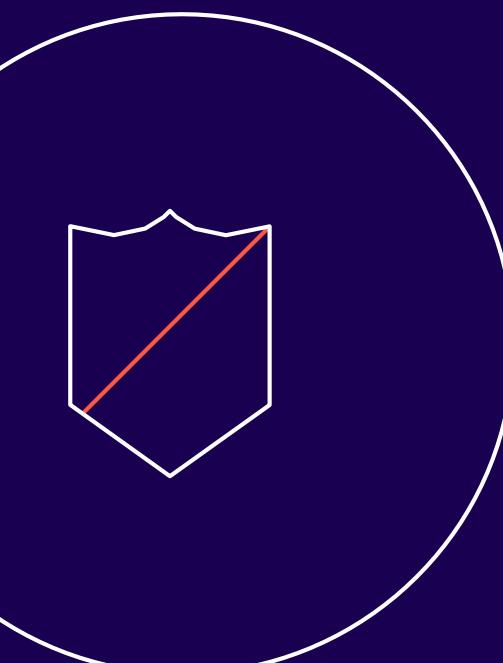
Common Languages

Supported programming in JavaScript, Arduino Yun, Python, Java, and C++



Code Flexibility

Build your own connectivity and IoT capabilities



Security

End-to-end cloud security with TLS 1.2 Websockets / TLS



Connectivity & Control Services

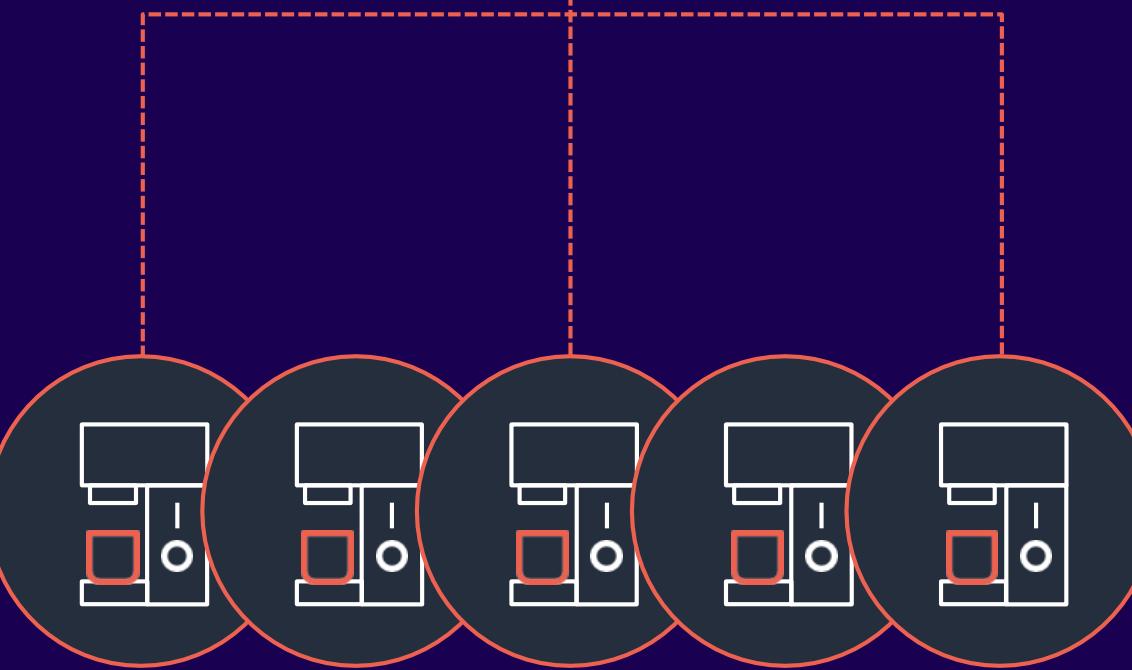
Fleet Provisioning

Automates device- and cloud-side configuration and authentication upon a device's first connection to AWS IoT Core.

Define templates for onboarding, and apply them to all devices for secure provisioning at scale.

Create unique identities to establish trust using the updated AWS IoT Device SDK or the AWS Mobile SDK in branded companion apps.

Automatically provision any number of devices upon first connection to AWS IoT Core.



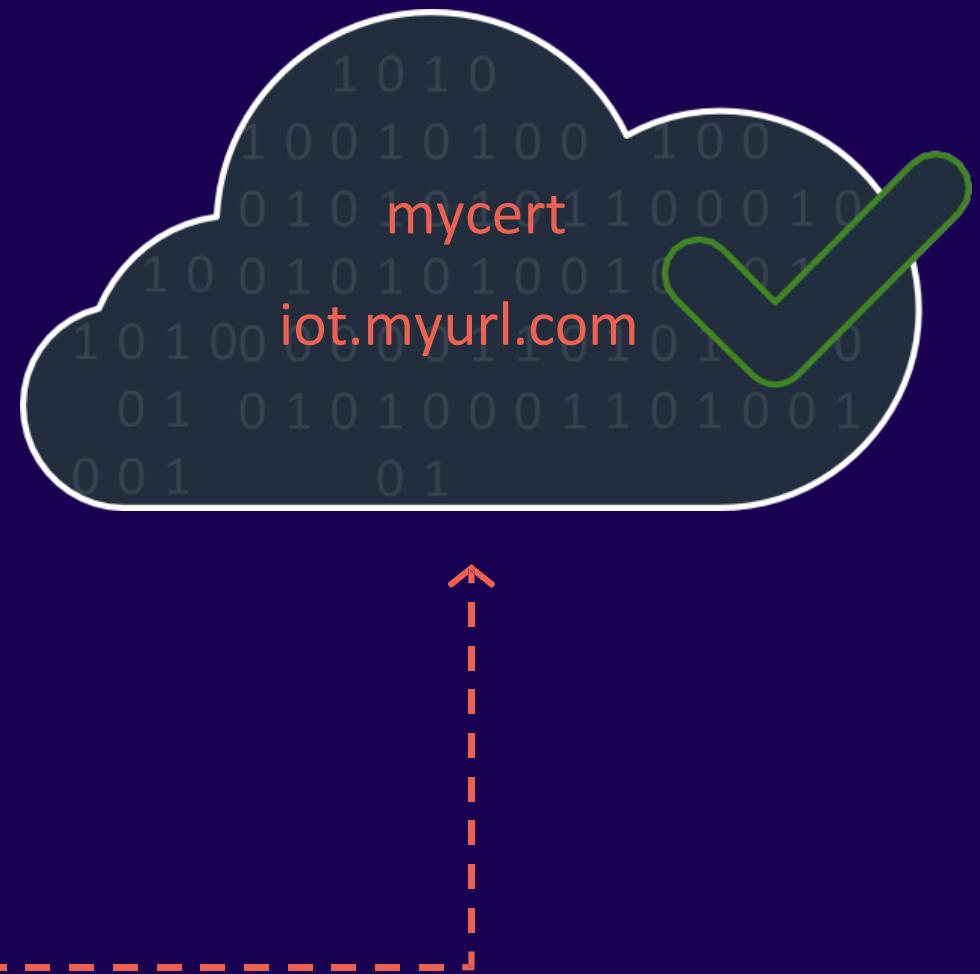
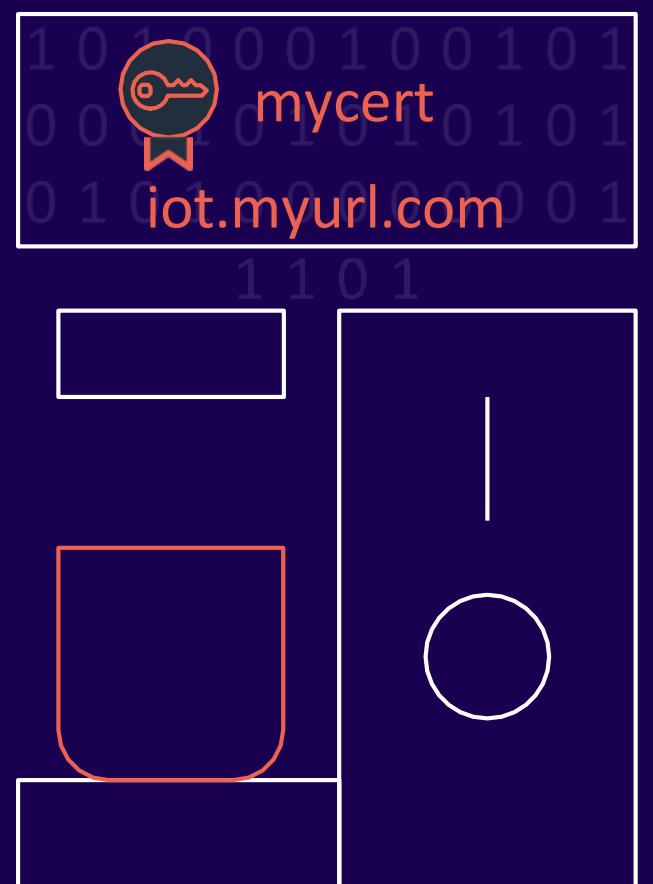
Configurable Endpoints

Easily migrate devices to AWS IoT by maintaining different configurations across diverse device fleets with minimal impact on existing devices and applications.

Create multiple IoT endpoints within a single AWS account and set up a unique configuration on each one.

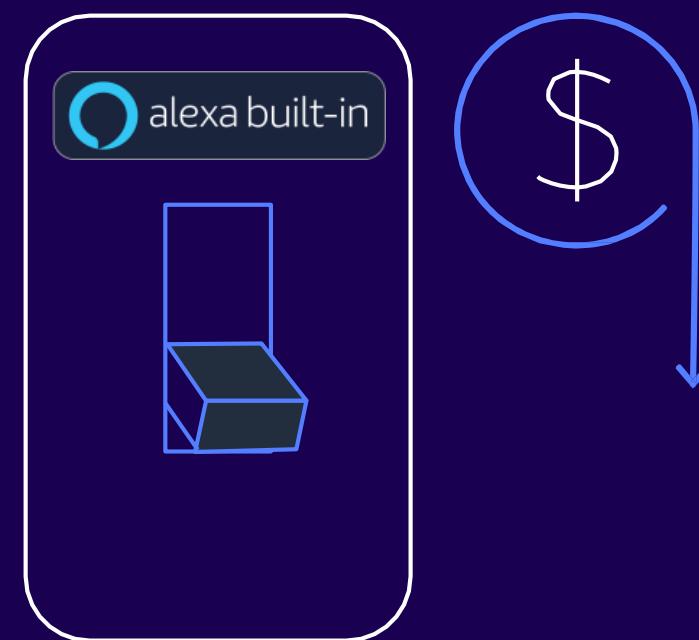
Continue to use your own domain names and associated server certificates after connecting to AWS IoT Core.

Keep your own identity and access management systems for provisioning new device identities.



Alexa Voice Service (AVS) Integration for IoT Core

Quickly and cost-effectively go to market with Alexa Built-in capabilities on new categories of products such as light switches, thermostats, and small appliances.



Lowers the cost of integrating Alexa Voice up to 50% by reducing the compute and memory footprint required.



Connectivity & Control Services



Create new categories of Alexa Built-in products on resource constrained devices (e.g., ARM 'M' class microcontrollers with <1MB embedded RAM).



Accelerate time to market with certified partner development kits that work with AVS Integration for IoT Core by default.



Problem

iDevices wanted to expand their smart home portfolio to include a product with onboard voice services. In-house engineers and designers developed Instinct™, a smart light switch with Alexa built-in. With the backend infrastructure and industrial design complete, the team needed to choose a cloud-based platform to execute and analyze IoT features.

Solution

iDevices uses the AVS Integration for IoT Core as the cloud-based messaging protocol to enable Alexa voice services, lighting control, and motion sensing functionality. Instinct allows users to invisibly integrate the power of Amazon Alexa throughout their homes and reap the benefits of whole-home voice control without sacrificing valuable counter space.

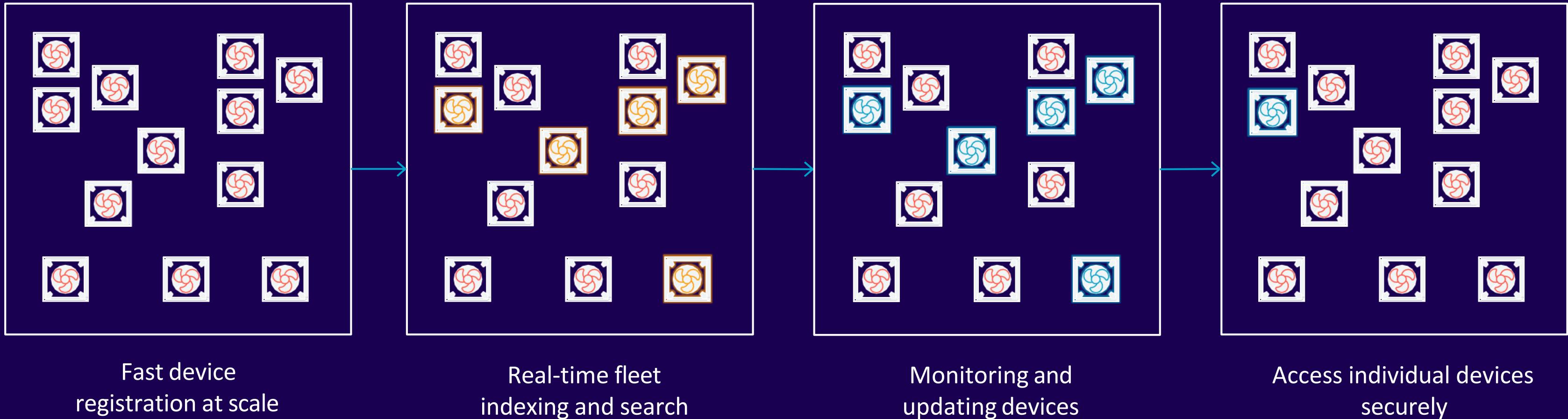
Impact

By employing AWS IoT, iDevices was able to accelerate time-to-market and optimize infrastructure costs while delivering an innovative product to the market.



AWS IoT Device Management

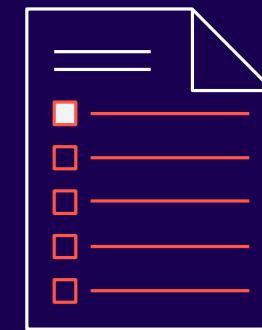
AWS IoT Device Management helps you register, organize, monitor, and remotely manage your growing fleet of connected devices.



Connectivity &
Control
Services

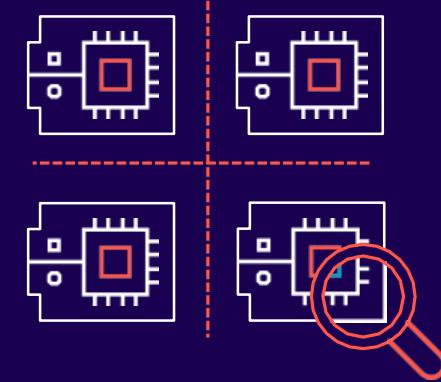


AWS IoT Device Management



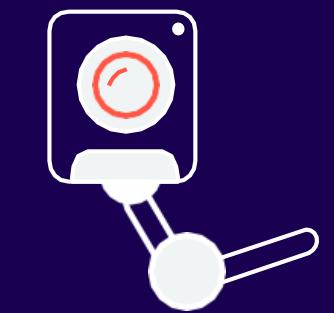
Bulk Things Registration

Register and configure devices with a few clicks



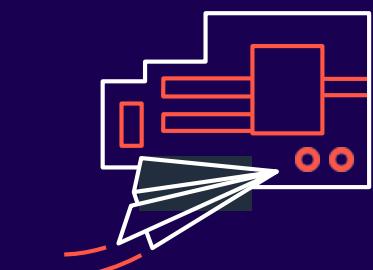
Fleet Indexing & Search

Understand the health and status of your device fleet



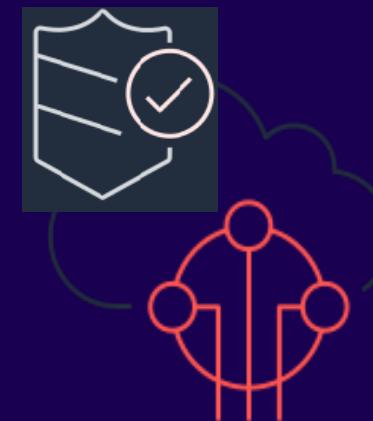
Device Logging & Monitoring

Collect device logs to quickly identify and remediate problems



Jobs

Organize and trigger actions on groups of devices



Secure Tunneling

Securely access devices behind restricted firewalls



Connectivity & Control Services

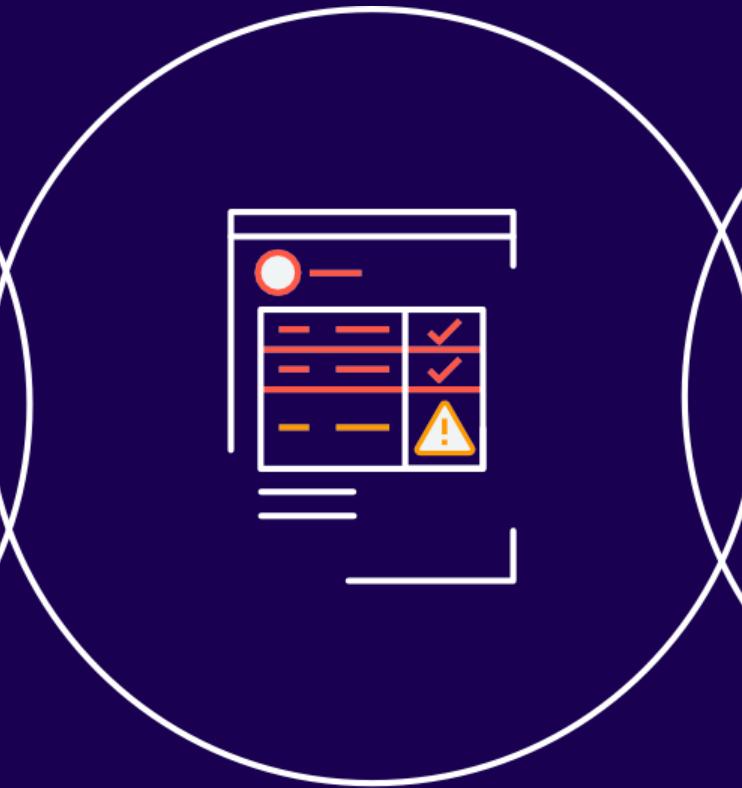


AWS IoT Device Defender



Audit

Validate IoT configuration is secure



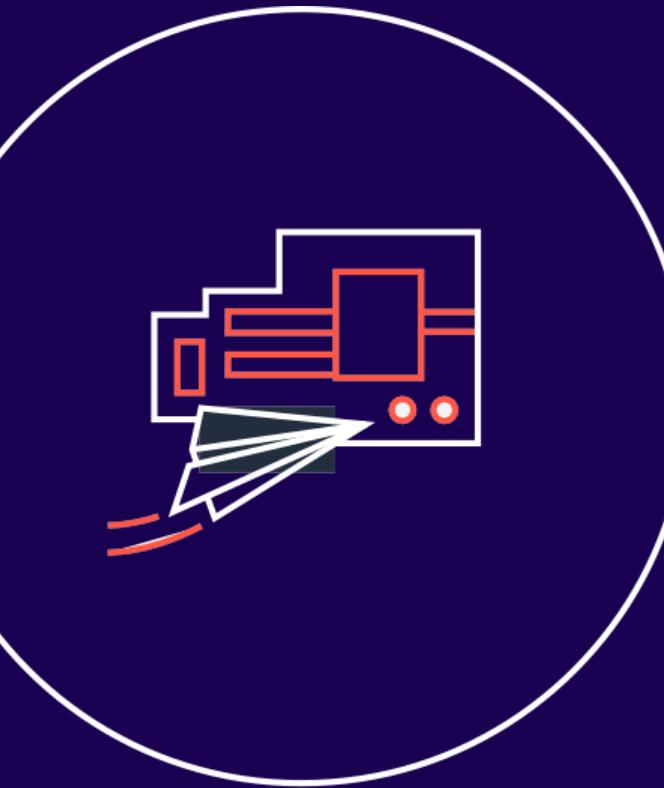
Detect

Detect anomalies in device behavior



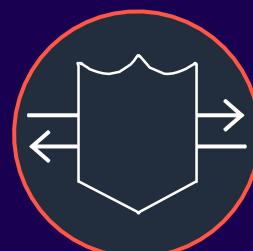
Alerts

Know when & what to investigate



Mitigate

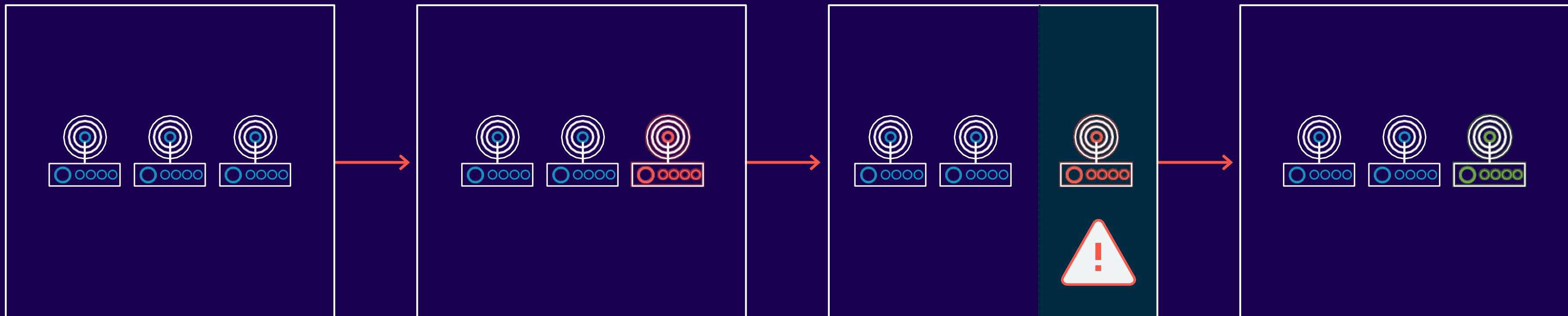
Remediate potential issues



Connectivity &
Control
Services

AWS IoT Device Defender

AWS IoT Device Defender is a fully managed IoT security service that enables you to secure your fleet of connected devices on an ongoing basis.



Audit device configurations, define and monitor device behavior

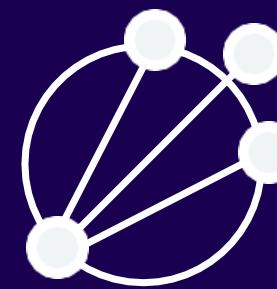
Identify drifts in security settings and detect device anomalies

Generate alerts

Patch security vulnerabilities

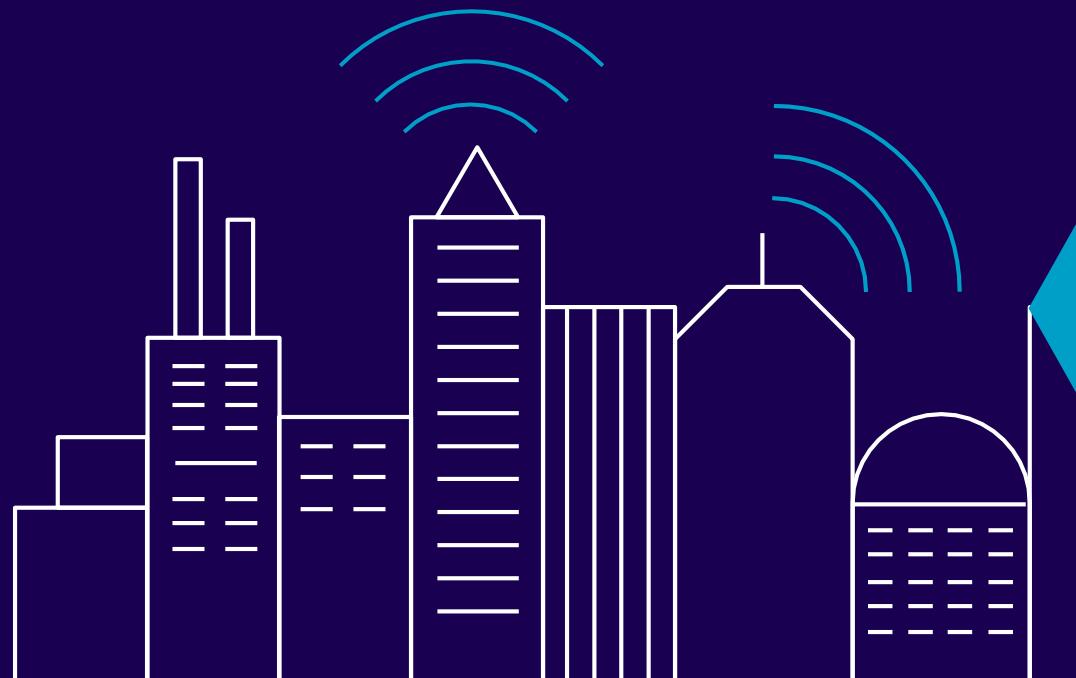


Connectivity & Control Services



AWS IoT Analytics

AWS IoT Analytics is a fully managed service that collects, pre-processes, enriches, stores, analyzes and visualizes IoT device data at scale.



**From raw
sensor data to
sophisticated
IoT analytics**



Analytics
Services

IoT transforms business processes

Most data collected on premises is never analyzed and thrown away. Use AWS IoT Analytics to transform business outcomes.





Valmet
FORWARD

Problem

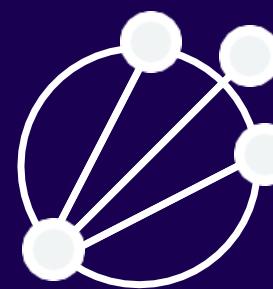
Valmet delivers technology and automation with multiple dependent processes running in parallel. Data analytics is needed to optimize Valmet's customers' processes.

Solution

Valmet is building a new digital twin capability to allow paper mill operators to view equipment and process data during production runs. AWS IoT Analytics is at the core of this solution training ML models for paper quality forecasting and scheduling metrics generation for digital twin view-generation.

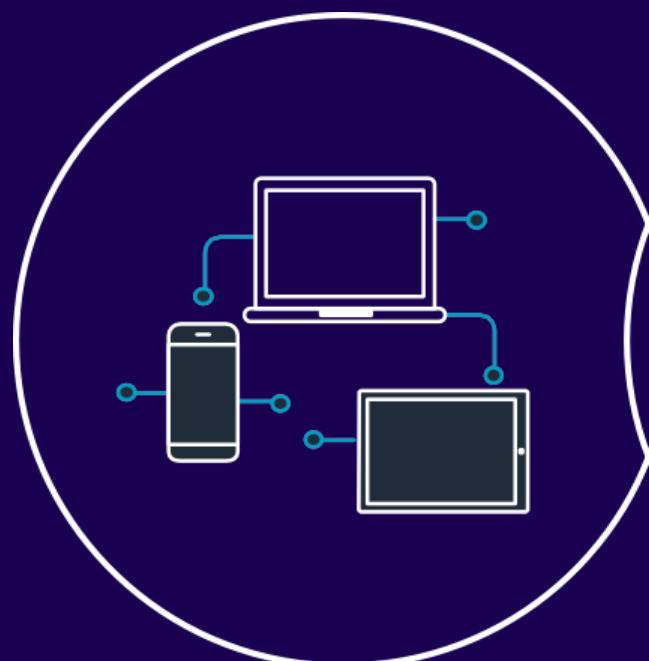
Impact

AWS IoT Analytics allows Valmet to combine historical models of equipment performance with live data from current operations to glean insights that help them to further provide solutions that enable their customers to produce paper with lower costs and optimum quality.



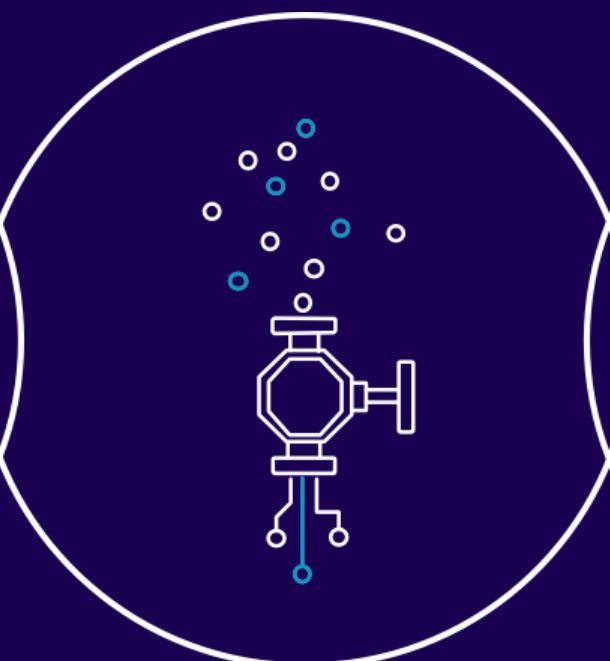
AWS IoT Analytics

AWS IoT Analytics is a service that processes, enriches, stores, analyzes, and visualizes IoT data for manufacturers and enterprises.



Collect

Collect only the data you want to store & analyze



Process

Convert raw data to meaningful information



Store

Store device data in time-series data store for analysis



Analyze

Get deeper insight into the health & performance of assets



Visualize

Quickly visualize your IoT data sets

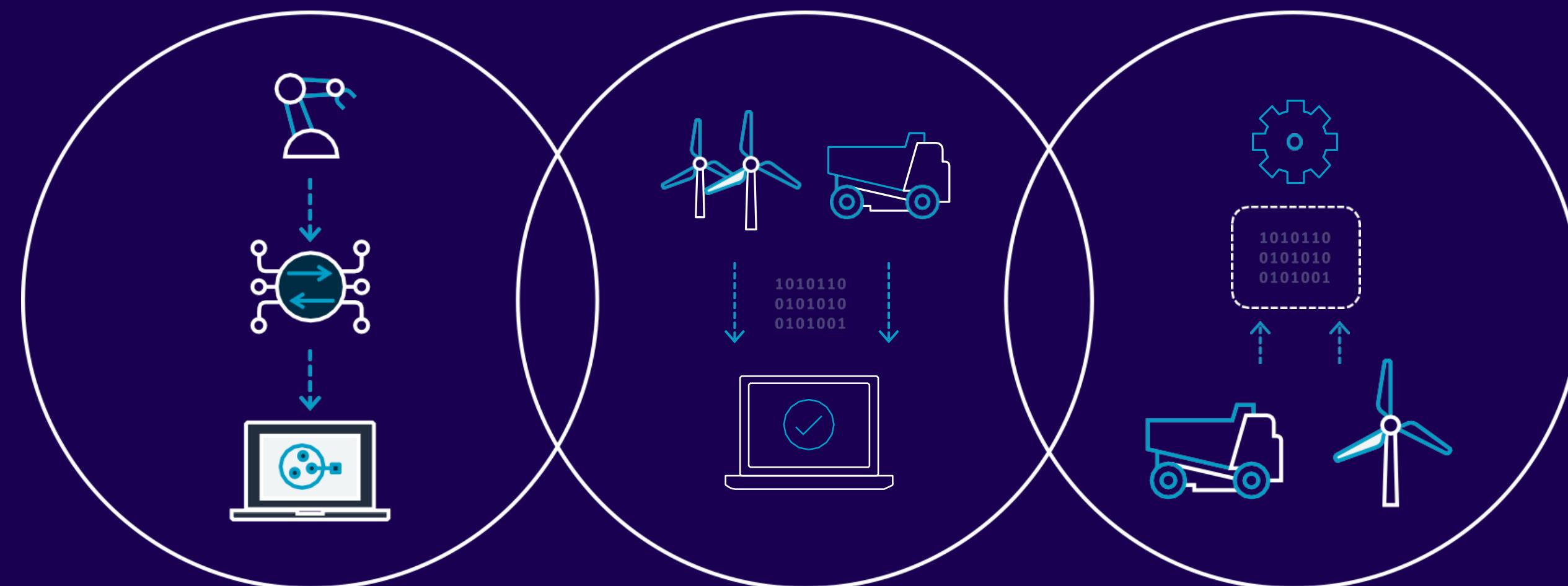


Analytics Services



AWS IoT SiteWise

AWS IoT SiteWise collects data from the plant floor with a local gateway, structures & labels that data, and generates real time KPIs & metrics to make better data-driven decisions.



Analytics
Services

Attach context
to machine data

Gain insights
into machine data

Visualize, interact with and
share machine data

SiteWise Monitor

Managed web applications to visualize and interact with equipment data

Fully managed web applications to visualize live and historical equipment data; no code, no resource management required

Supports single sign-on using LDAP or built-in credentials

Automatically discover assets and instantly visualize live equipment data through charts

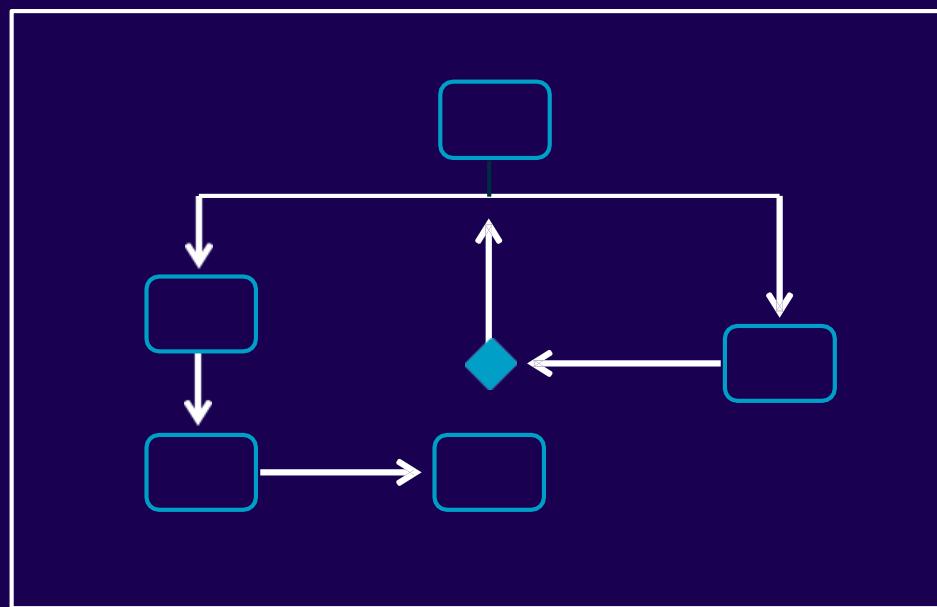
Create dashboards to organize and share equipment data with any team in your organization



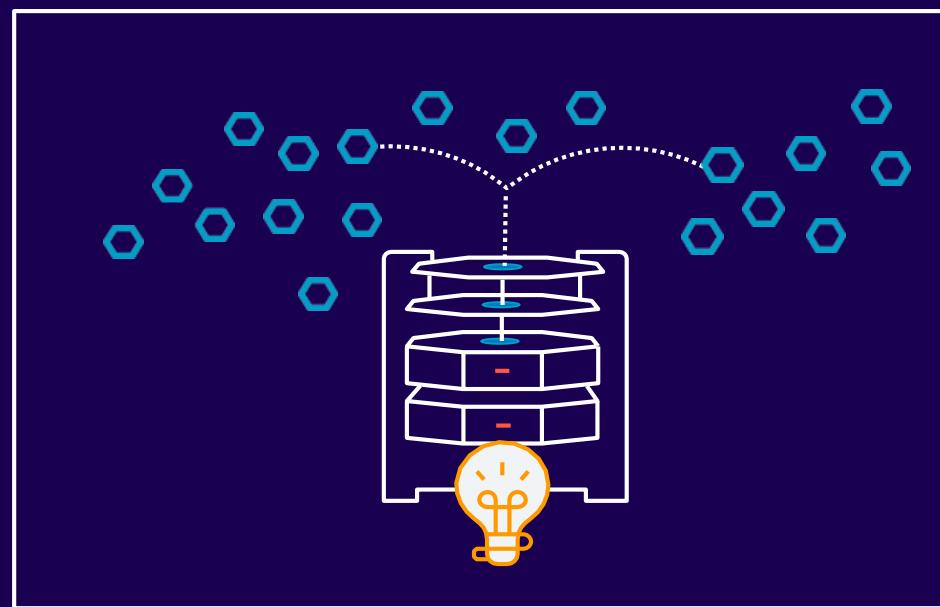


AWS IoT Events

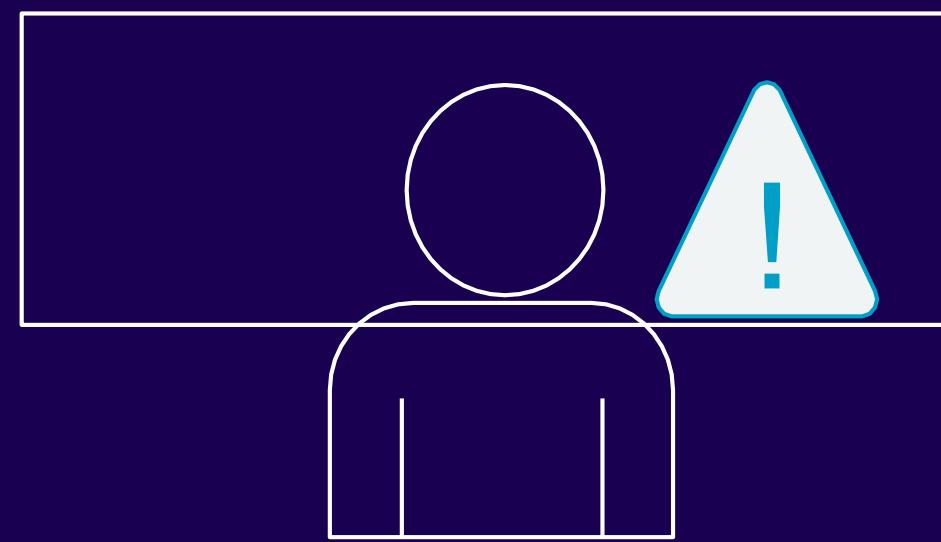
AWS IoT Events is a managed service that continuously monitors data from your equipment to identify their state, detect changes and trigger the appropriate responses when changes occur



Build simple logic to evaluate incoming telemetry data to detect stateful changes in equipment or a process



Detect events from data across thousands of sensors and other sources



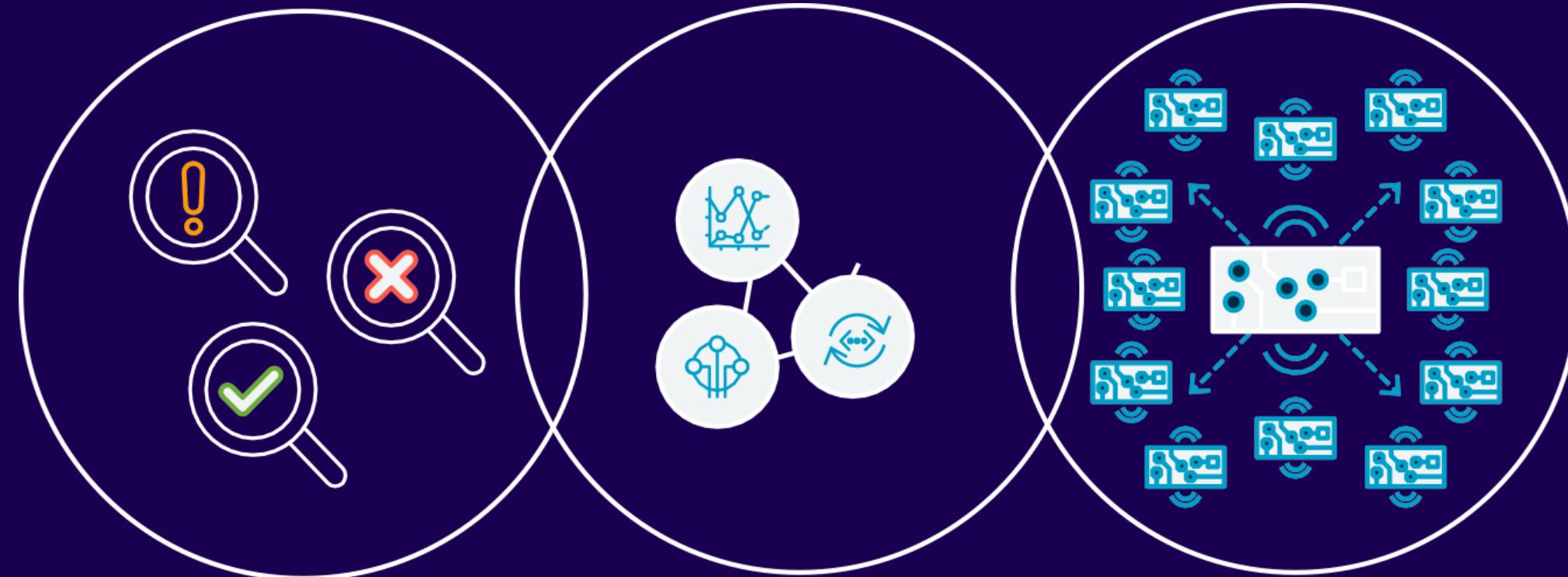
Trigger responses to optimize operations



Analytics Services



AWS IoT Events



Event Detector Models

Reduce the cost of
device maintenance

Integration with analytics tools & other AWS services

Uncover new
insights and trigger
actions

Scalability

Easily automate
operations



Analytics
Services



Bayer CropScience



Problem

In the seed business, it's important to gain better and faster visibility into what's going on in fields during planting and harvest within breeding research and supply chain organizations.

Solution

AWS IoT helps Bayer Crop Science manage the collection, processing, and analysis of seed-growing data. Data analysts use the new data collection platform to access data on their mobile devices via dashboards. The solution captures multiple terabytes of data from seed transportation, planting, and growing in the company's research fields across the globe.

Impact

Using AWS IoT, Bayer Crop Science can provide seed data to analysts in just a few minutes instead of a few days. This also helps farmers gain better visibility into field conditions and provides a robust edge processing and analytics framework.

Blockchain

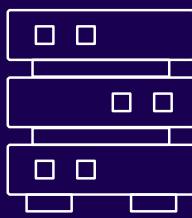
What is blockchain?



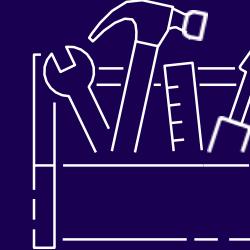
Blockchain makes it possible to build applications where multiple parties can execute transactions **without the need for a trusted, central authority**

Today, building a scalable blockchain network with existing technologies is complex to set up and hard to manage

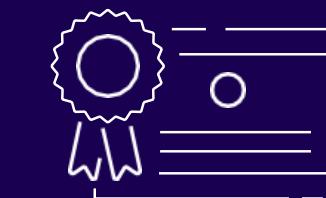
Each network member needs to



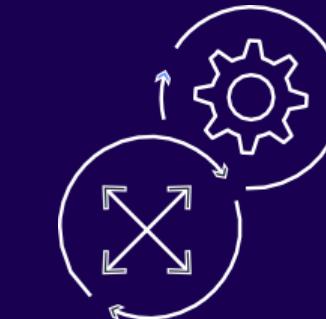
Manually provision hardware



Install software

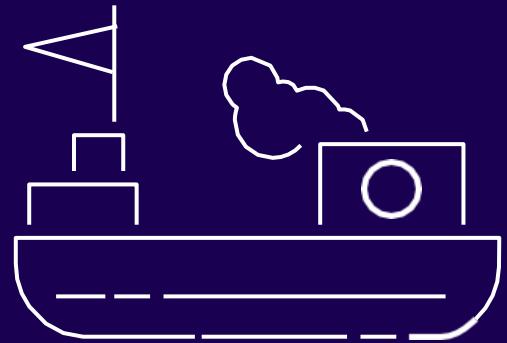


Create & manage certificates for access control



Configure networking components

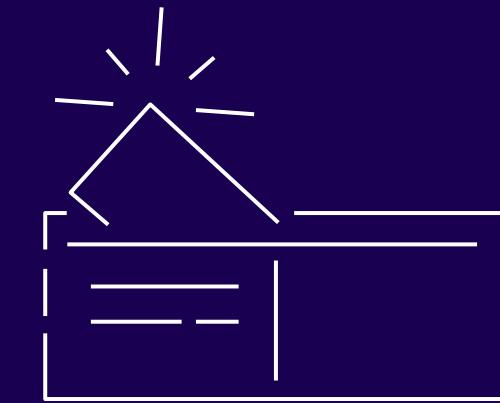
Example use cases



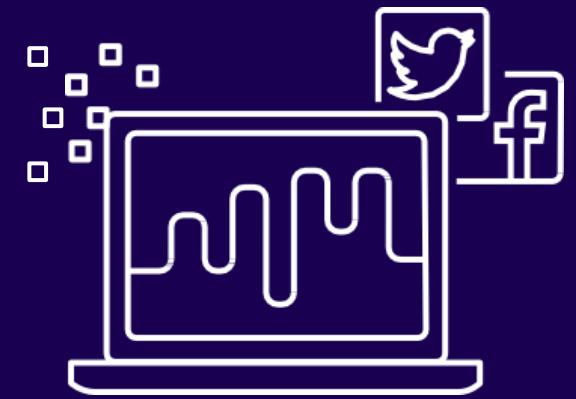
Shipping



Supply chain
management



Finance
and banking



Digital
advertising

AWS blockchain services

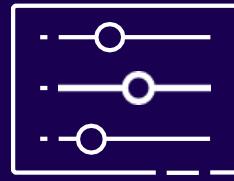


Amazon
Managed
Blockchain

Fully managed service that makes it easy to create and manage scalable blockchain networks using popular open-source frameworks

- Hyperledger Fabric
- Ethereum

Amazon Managed Blockchain features



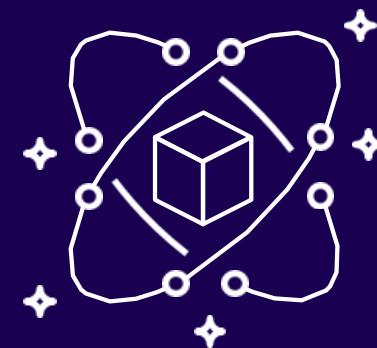
Fully managed
Create a blockchain network in minutes



Open-source variety
Support for two frameworks



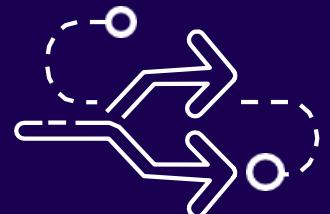
Decentralized
 Democratically govern the network



Reliable and scalable
Backed with Amazon QLDB technology



Low cost
Only pay for resources used



Integrated
With AWS services

Nestlé's chain of origin coffee cultivates supply chain transparency with Amazon Managed Blockchain

Challenge

Nestlé is the biggest procurer of coffee in the world, and it wanted to uncover transparency around its coffee bean supply chain beyond its brokers and buyers

Solution

Nestlé turned to Amazon Managed Blockchain to trace back through every step in its supply chain - from the farmer and grader to the roaster and packer

“Whether it’s how we ensure freshness, whether it’s making sure that the packaging being used is better for the planet, **it means that the value is going back to the farmers and the partners we’re working with.**”

Armin Nehzat, Digital Technology Manager, Nestlé

Benefits

- Nestlé can now grow one-on-one relationships with coffee farmers and roasting facilities
- Because the secure blockchain ledger is public, it provides greater accountability to everyone in the supply chain



Company: Nestlé

Country: Switzerland (CH)

Employees: 300,000+

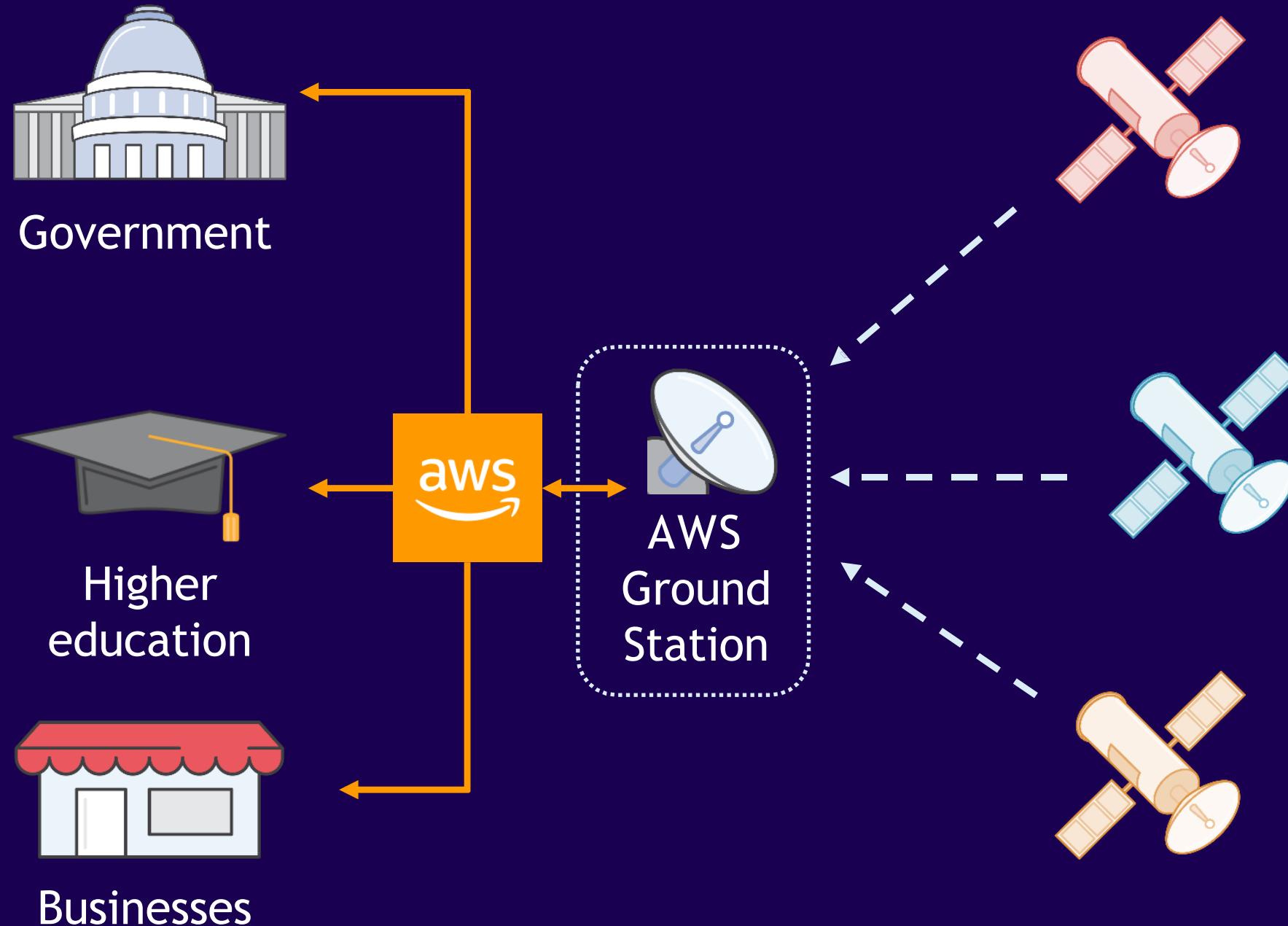
Website: Nestle.com

About Nestlé

Nestlé is the world's largest food and beverage company. It is present in 190 countries around the world, and it has 308,000 employees. Nestlé is also the biggest procurer of coffee globally.

AWS Ground Station

What AWS Ground Station offers



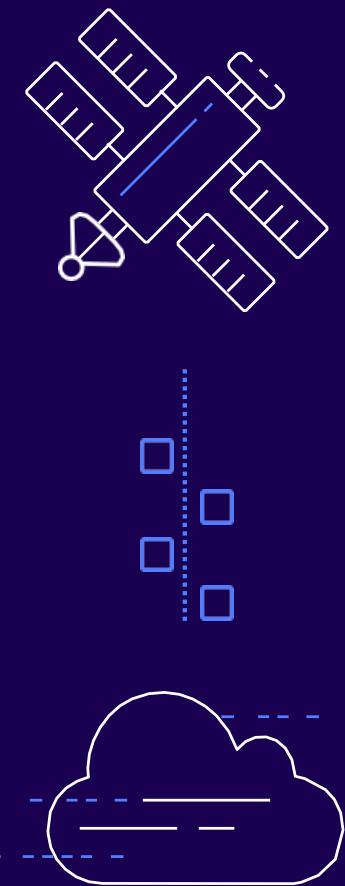
- Satellite ground support with no infrastructure commitments
- Pay-by-minute pricing
- Self-service scheduling
- Collocated ground stations and AWS data centers providing direct access to AWS resources and services
- Backhaul of base band data to customer Region of choice included in pricing
- Near-real-time data delivery

AWS Ground Station: What is it?

AWS Ground Station is a fully managed service that you can use control satellite communications, process data, and scale operations without having to worry about building or managing your own ground station infrastructure

These facilities provide communications between the ground and the satellites in space

- Low-latency global fiber network
- Direct access to AWS services
- Fully managed service (no infrastructure commitments)
- Pay-as-you-go pricing
- No licensing requirements
- Scale satellite communications on demand when your business needs it

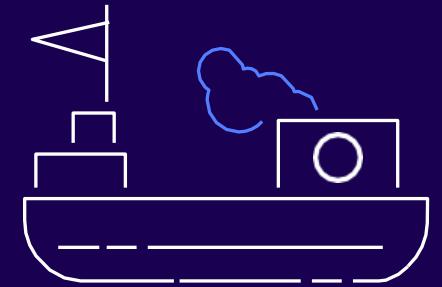


Common satellite data cloud processing use cases



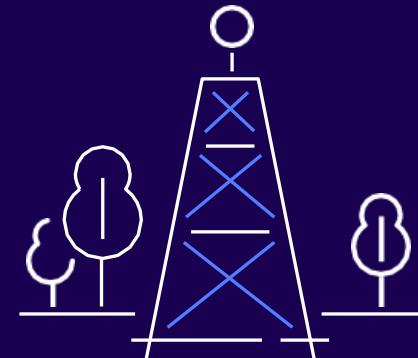
**Weather forecasting
and agriculture**

Commercial fruit producers can monitor crop health and water levels to ensure efficient use of limited resources



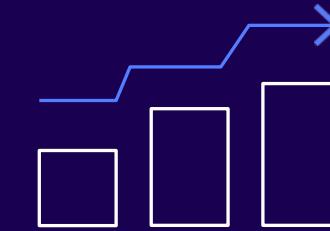
**Global shipping
and anti-piracy**

Use registries of ship placement, destination, and tracking to confirm accuracy of ship positioning, as well as be notified of any deviations from normal operations



**Earth observation
and fire safety**

Use low-latency access to high-resolution heat mapped images of the earth to inform frontline fire commanders on safest, lowest heat entry points to fight fires



**Retail
forecasting**

4.8 million satellite images from 44 major US retailers confirms numbers of cars in parking lots and yields an informational advantage to forecasting accuracy

AWS Wavelength

AWS Wavelength



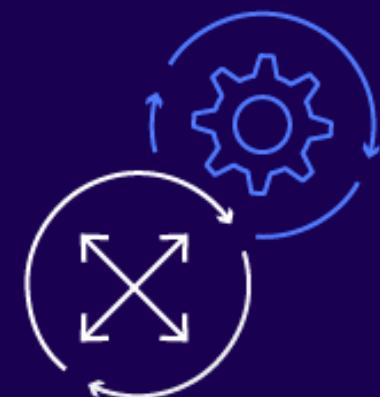
AWS
Wavelength

AWS Wavelength combines the high bandwidth and ultra-low latency of 5G networks with AWS compute and storage services to help developers innovate and build a whole new class of applications

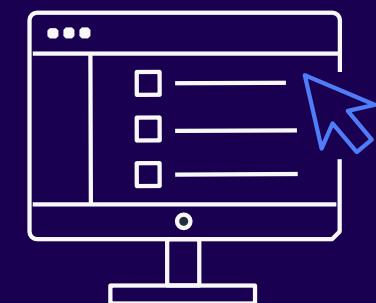
- AWS infrastructure and services in CSP 5G networks
- Ultra-low latency, local data processing
- Scalable capacity in CSP data center managed and supported by AWS

AWS Wavelength: Built for the mobile edge

AWS services from inside the CSP mobile network



AWS compute and storage infrastructure embedded inside CSP mobile network



Single pane of management, across Wavelength Zone and AWS Regions



Access to services in the AWS Region



Develop applications once and deploy for use with 5G network globally



Failover from Wavelength Zone to AWS Region

AWS Wavelength use cases

Healthcare



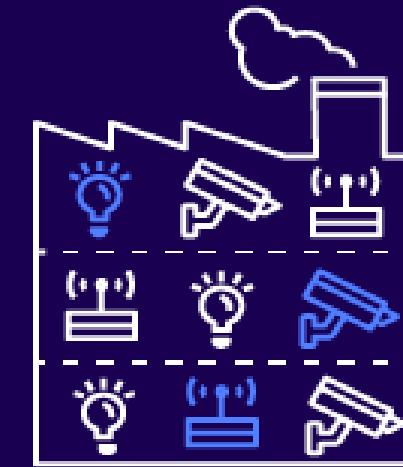
AI/ML solution for processing and analyzing video, images, and data for real-time diagnosis

Connected vehicles (C-V2X)



Real-time monitoring of data from sensors for road safety, secure connectivity, in-car telematics, and autonomous driving

Smart factory



Accelerating the industrial edge with AI/ML, video recognition for software-defined manufacturing

LG uses AWS Wavelength for low-latency, high-throughput delivery of V2X data



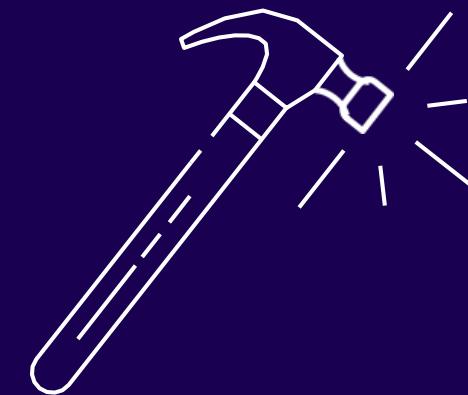
“5G gives us that connectivity piece with high bandwidth and low latency, while **Wavelength is providing the necessary compute power at the edge to supplement the 5G technology**. So, it’s about bringing security, privacy, connectivity, and compute together for the benefit of consumers and their safety.”

Harsh Kupwade Patil,
Security Leader & Principal Research Engineer, LG Electronics

Next Steps

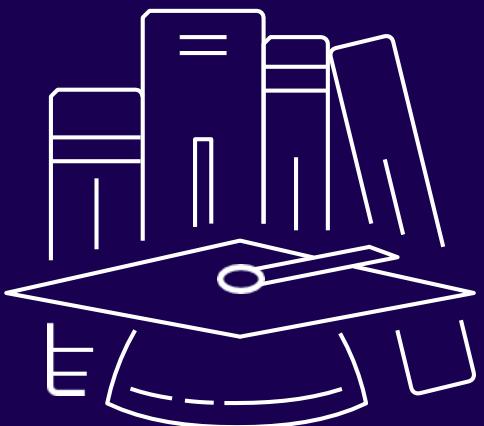
Choose your path

1



Start
building

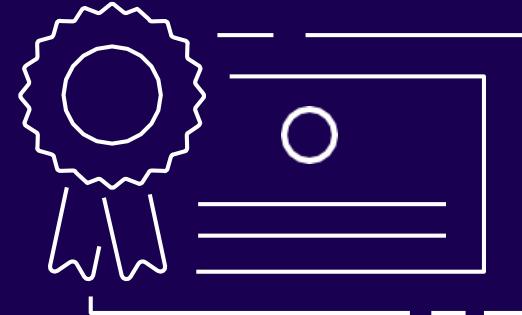
2



Continue
learning

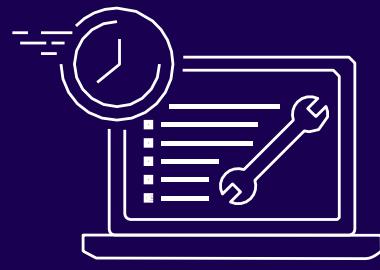
aws.amazon.com

3



Obtain
certification

Start building: Tips for getting started



AWS Free Tier

Gain free, hands-on experience with AWS products and services

aws.amazon.com/free

Billing alarms

Receive billing alerts that help you monitor the charges on your AWS bill

In the Billing Console

Tools

AWS developer tools, command line tools, IDE & IDE toolkits, SDKs, mobile and IoT device SDKs

aws.amazon.com/developer

Quick Starts

Automated, gold-standard deployments in the AWS Cloud

aws.amazon.com/quickstart

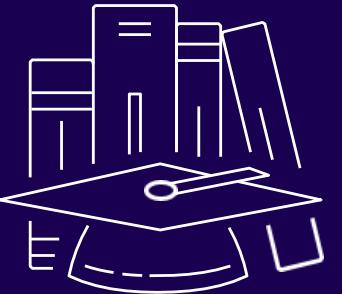
Continue learning

Learn at your
own pace



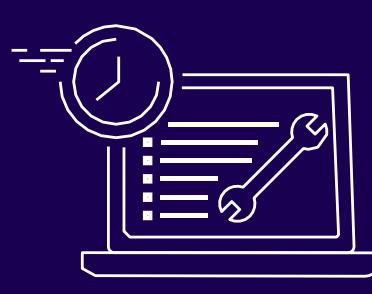
Expand your AWS Cloud skills with our self-paced digital course, [AWS Cloud Practitioner Essentials](#)

Learn from
AWS experts



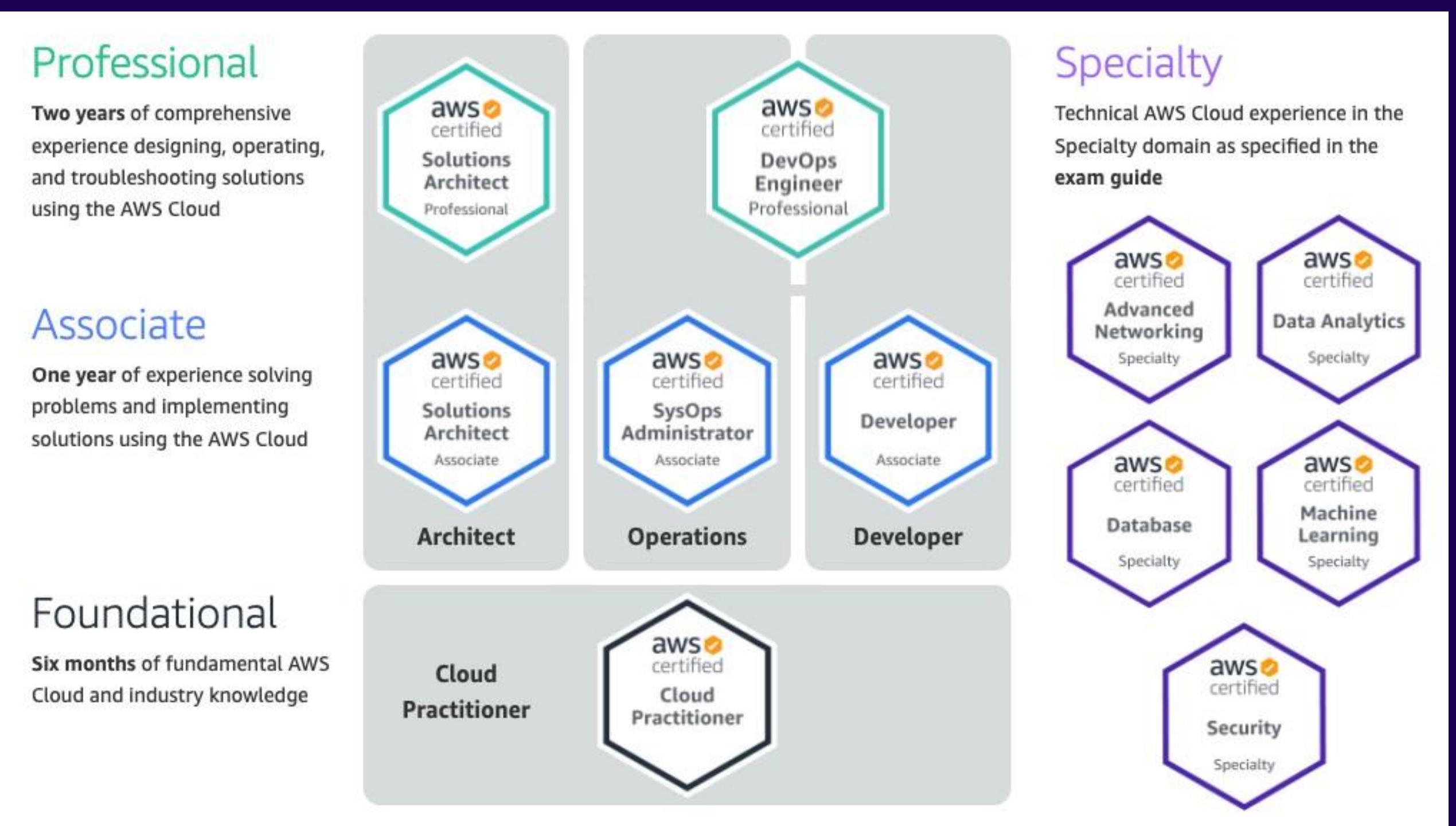
Build your AWS Cloud skills with our classroom course, [AWS Technical Essentials](#)

Ramp-Up
Guides



Our Cloud Practitioner [Ramp-Up Guide](#) offers a variety of resources to help build your knowledge of the AWS Cloud and prepare for the AWS Certified Cloud Practitioner certification

Validate expertise with AWS Certification



Why certify?

- Demonstrate your expertise
- Earn recognition and visibility
- Foster credibility with your employer and peers

Certification resources

- Exam Prep training
- Self-paced labs

Key takeaways

Start Building

- AWS free tier, Quick Start guides, developer tools, samples

Continue learning

- Self-paced labs, Classroom courses, Ramp-up guides

Certification

- Choice of Foundation, Associate, Professional and Specialty certification

Thank you!