

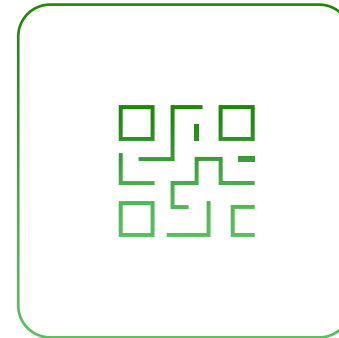
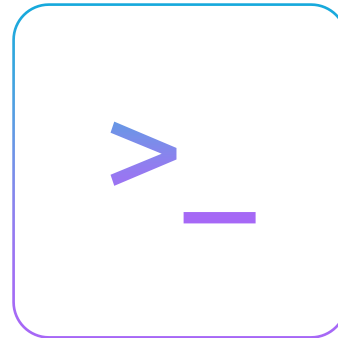


KodeKloud

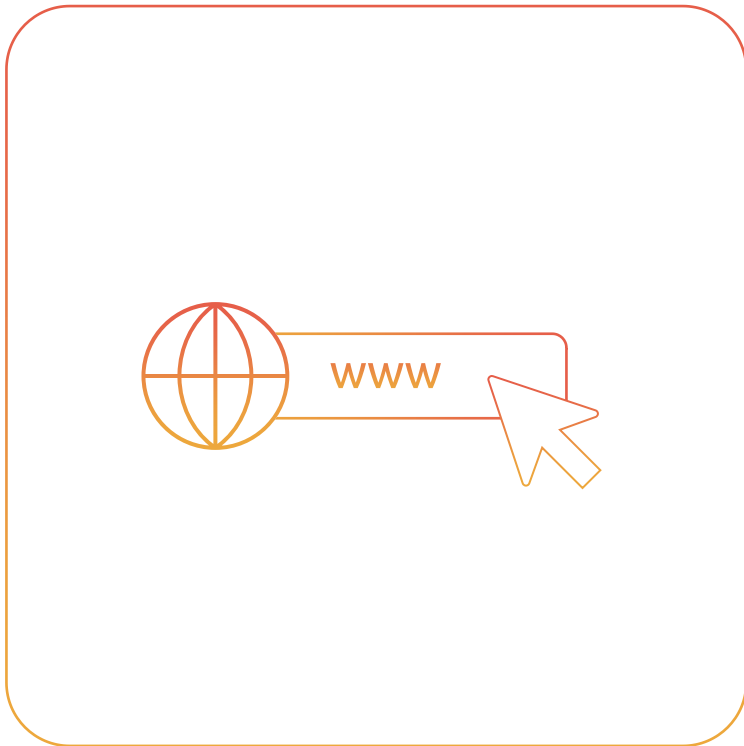
Methods of Deployment



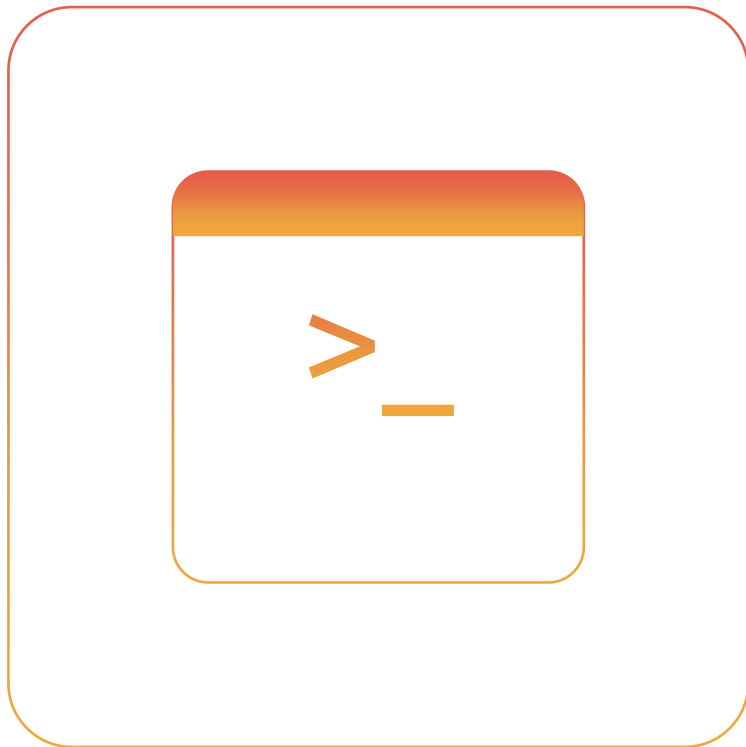
Deployment Methods



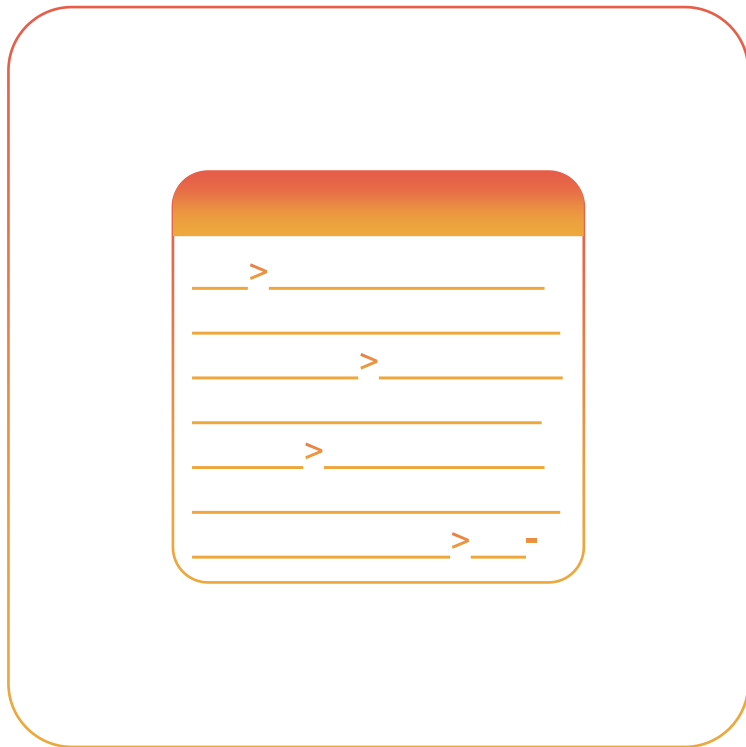
Console



Command Line Interface



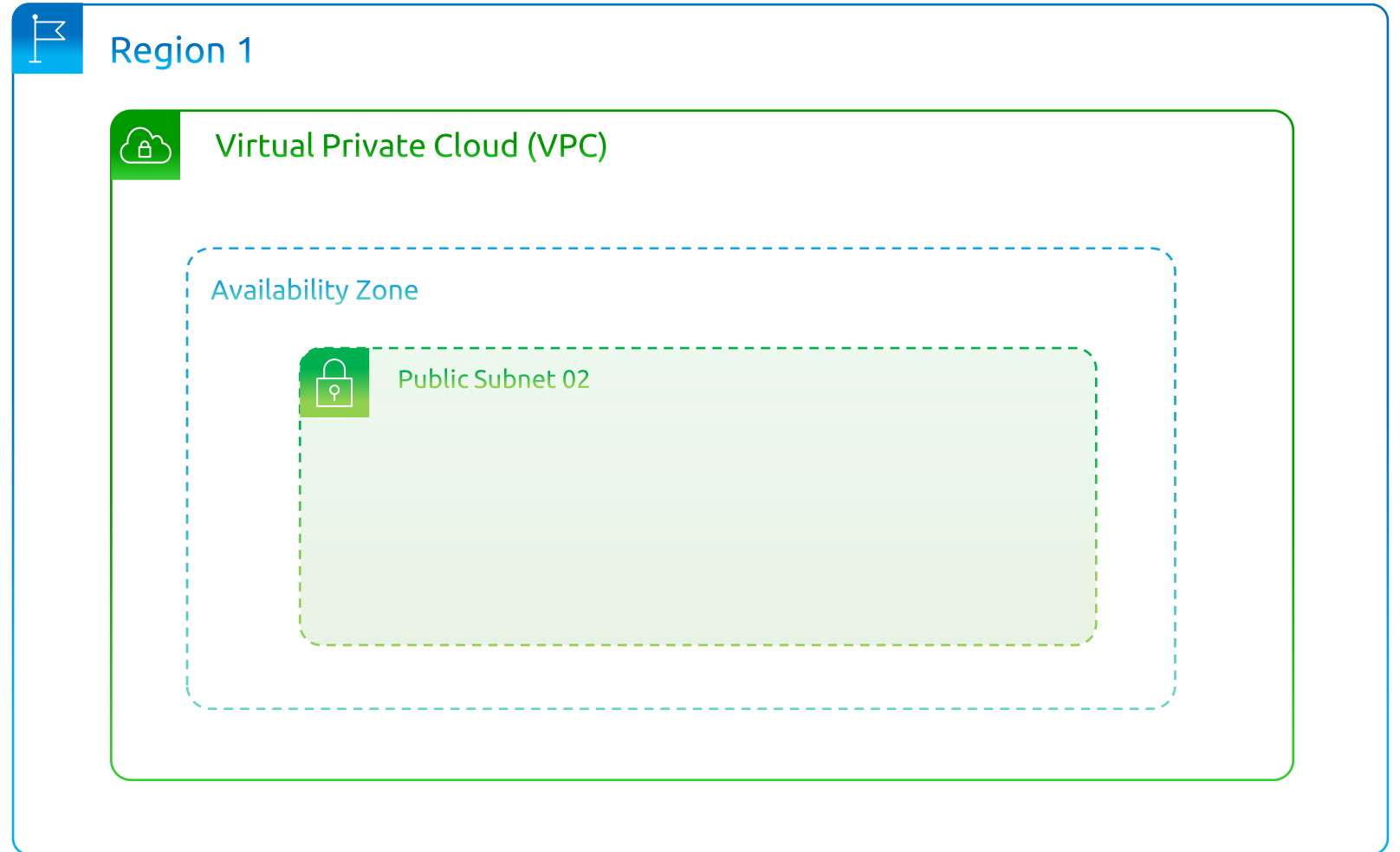
AWS SDK



Global Infrastructure



Global Infrastructure





Regions



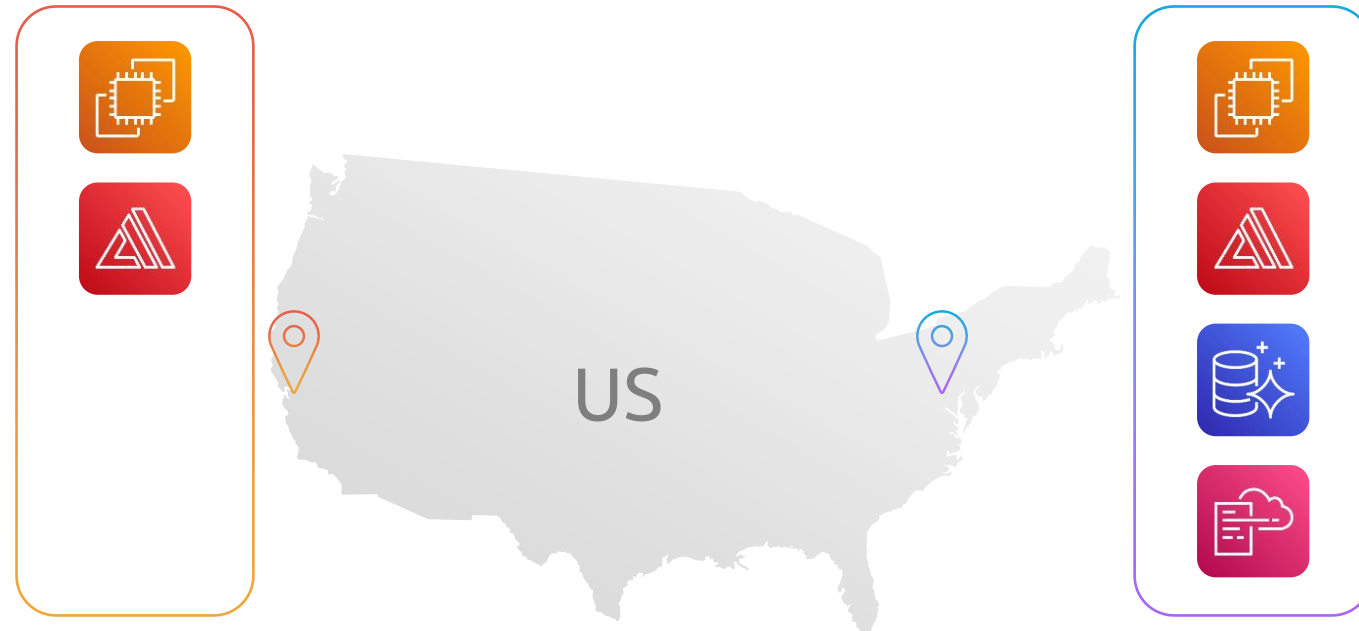
- An AWS **Region** is a geographic location where **resources** can be **deployed**
- Each **region** is designed to be isolated from other **regions**
 - Fault tolerance
 - Stability



Regions



Selecting a Region



- All **regions** are not the same
- Not all **services** are available in all **regions**
- GovCloud regions designed for **U.S. Government** and customers who need to meet **regulatory** and **compliance** requirements



Selecting a Region

Cost of services varies from region to region

You should select a region as close to your customers as possible

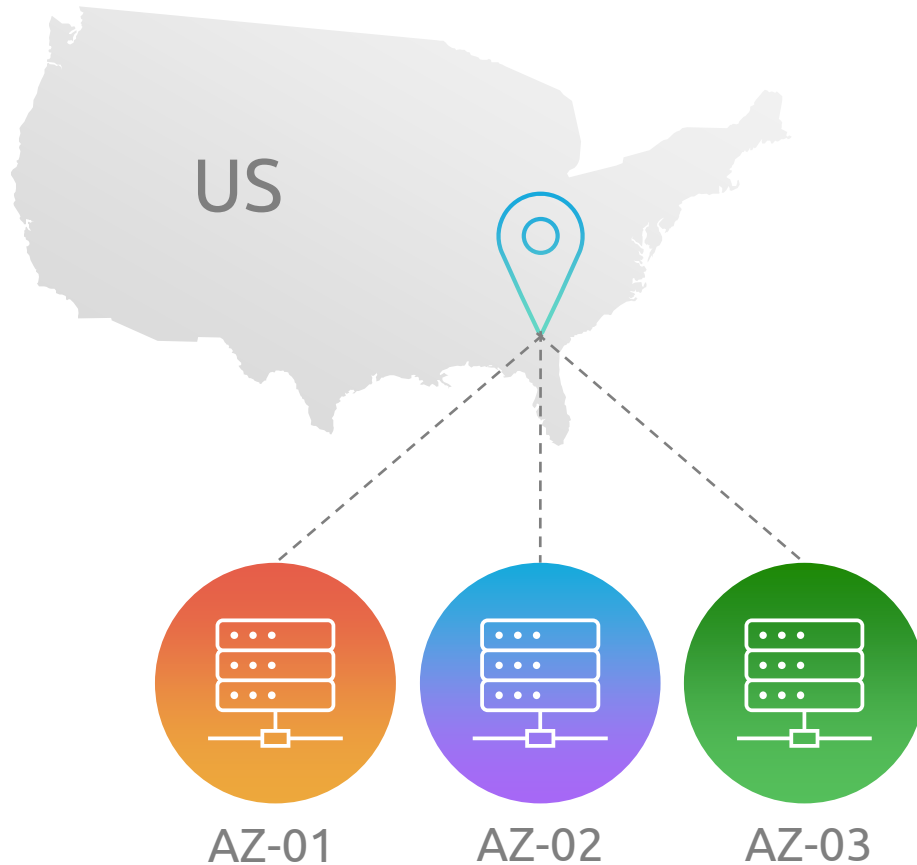


Selecting a Region

Due to **compliance** and **legal** requirements, you may need to run your **infrastructure** in certain areas/countries



Availability Zones (AZ)

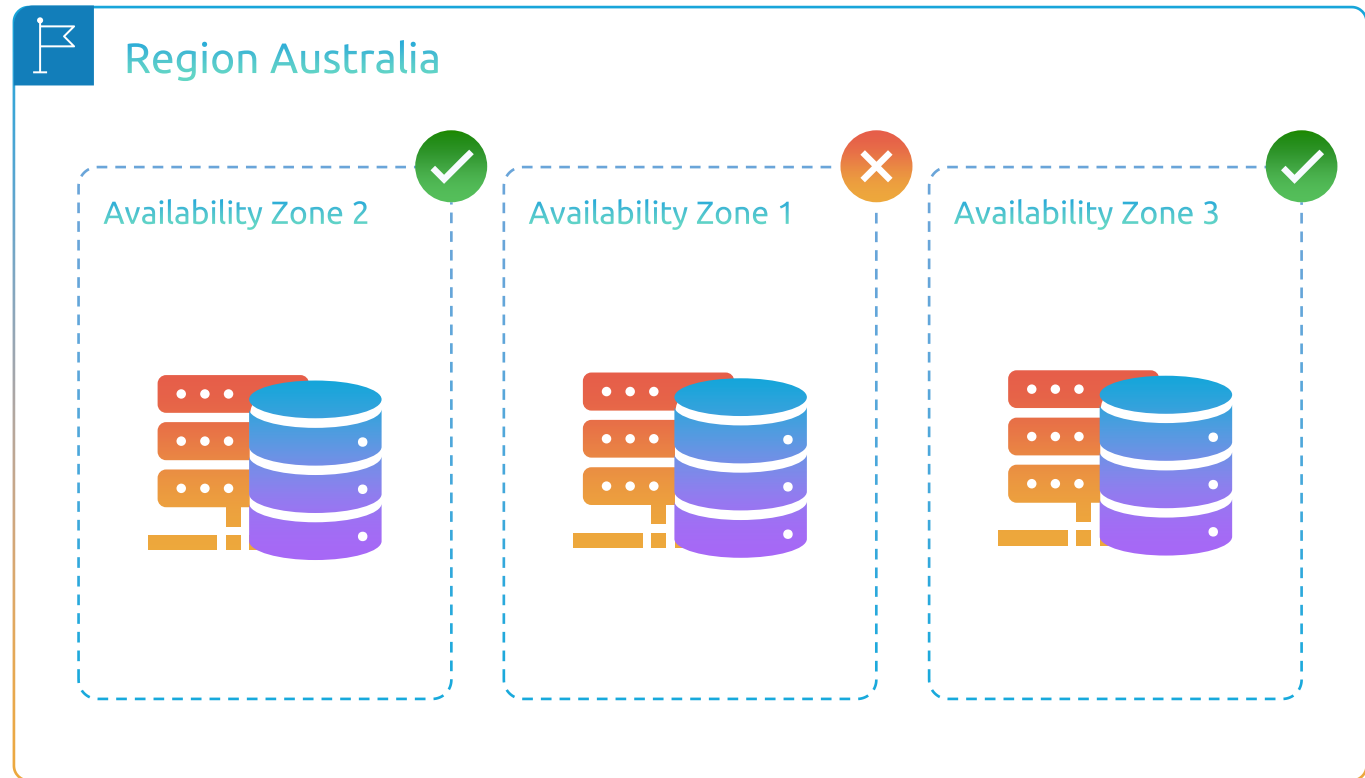


- Within a **region**, there are multiple **availability zones**
- An **Availability Zone (AZ)** is one or more discrete **data centers** with redundant power, networking, and connectivity in an AWS **region**
- **Availability zones** provide redundancy within a **region**

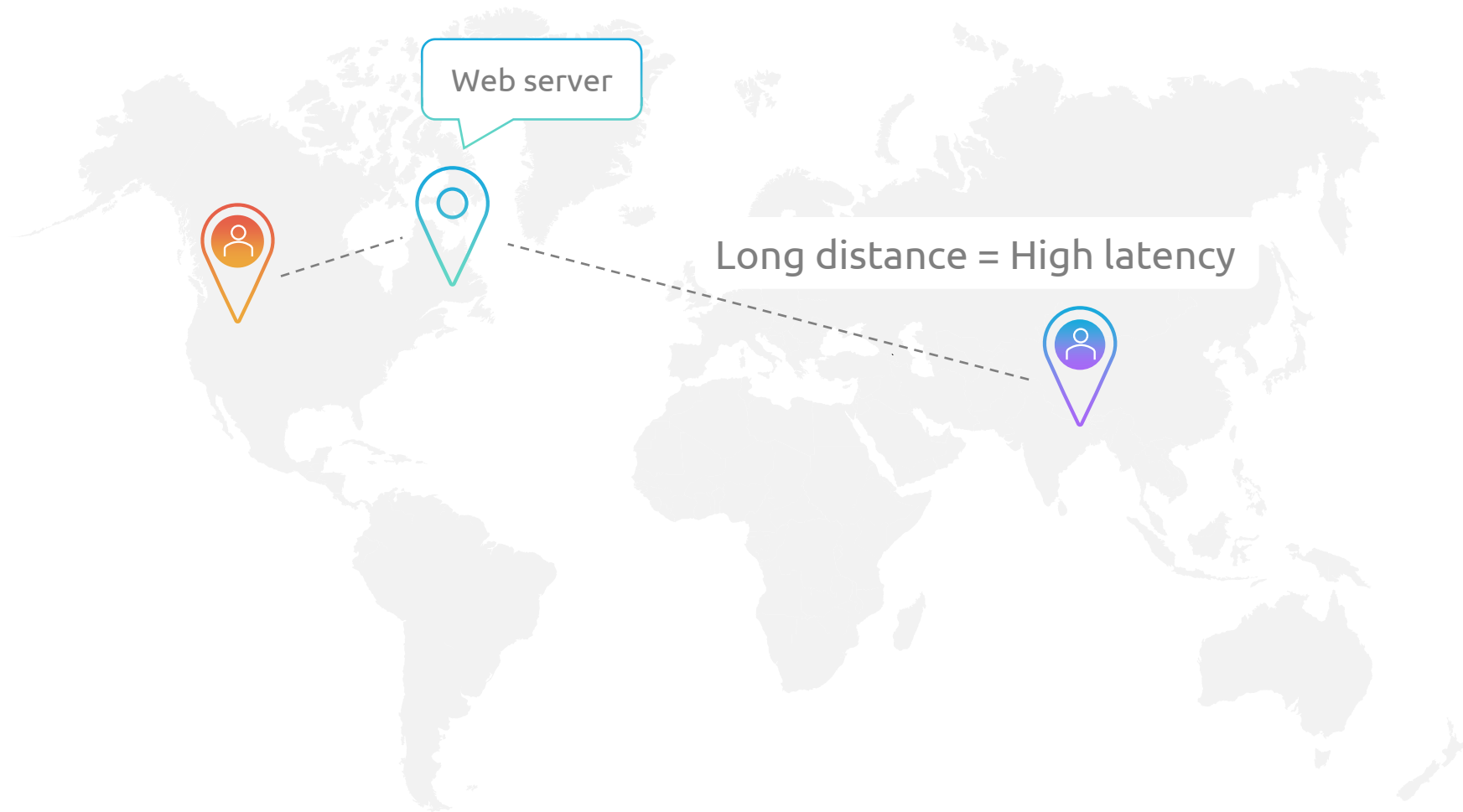


Availability Zones (AZ)

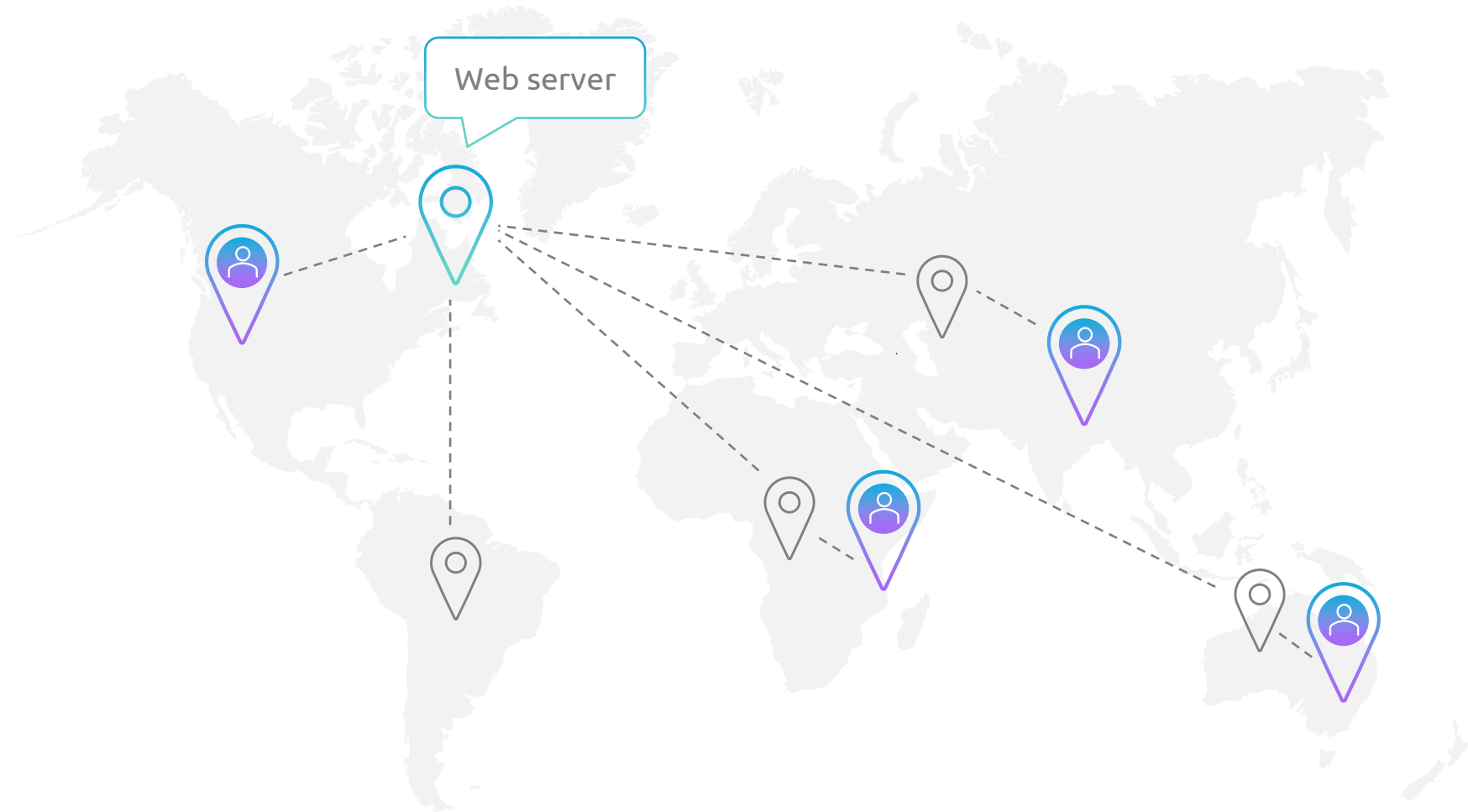
If you have an **application** deployed in only one **AZ**, then the entire **region** is affected in case of a failure in that **AZ**



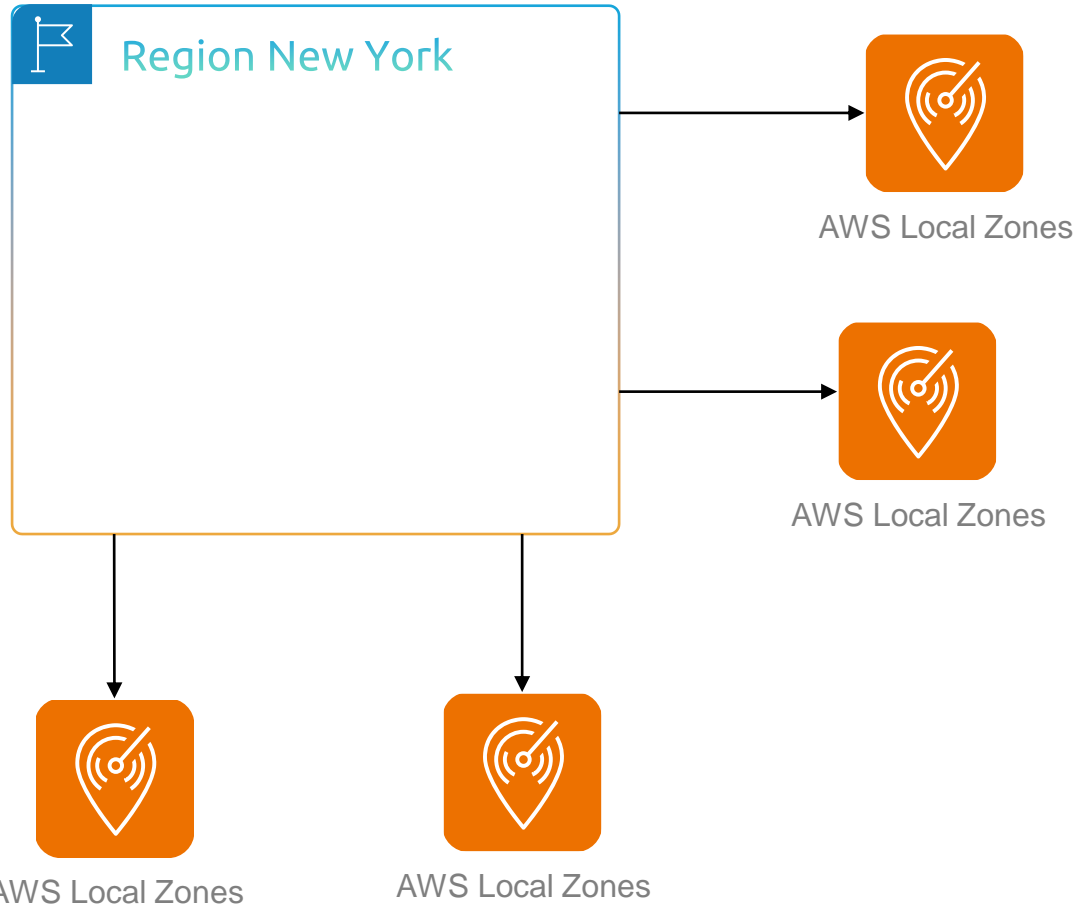
Global Content Delivery & Edge Locations



Global Content Delivery & Edge Locations



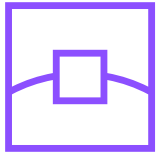
Local Zones



- **Local Zones** are extensions of an AWS **Region**
- **Local Zones** allow you to use select AWS services, like **compute**, and **storage** closer to **end-users**
- **Local Zones** provide high-bandwidth, secure connection to parent AWS **region**
- Great for low latency **applications** like real-time gaming, live streaming, and virtual workstations for the **local city**



Edge Locations vs Local Zones



Edge location



AWS Local Zones

- Both help get **services** closer to **end-users** to improve user experience and minimize **latency**
- **Local Zones** are extensions of a **regions** physically located in **major cities** across the world
- **Local Zones** have their own isolated infrastructure but are connected to the parent AWS region through a high bandwidth network link
- **Local Zones** Provides access to subset of **services** like **EC2** & **EBS**
- **Edge Locations** are small geographically dispersed compute sites that primarily support services like **CloudFront**(content delivery network), **Route 53**, **AWS WAF**
- **Edge Locations** are limited to mostly **CDN** services and do not provide full **compute** services
- There are hundreds of **Edge Locations** as they are not full sized datacenters; there are over 20+ **local zones**



Global Infrastructure - Summary



Regions are locations to which certain services can be deployed



Not all services are available in all Regions



Availability Zones (AZ) are isolated and independent datacenters inside Regions



Edge locations are smaller Points of Presence where services are run closer to customers



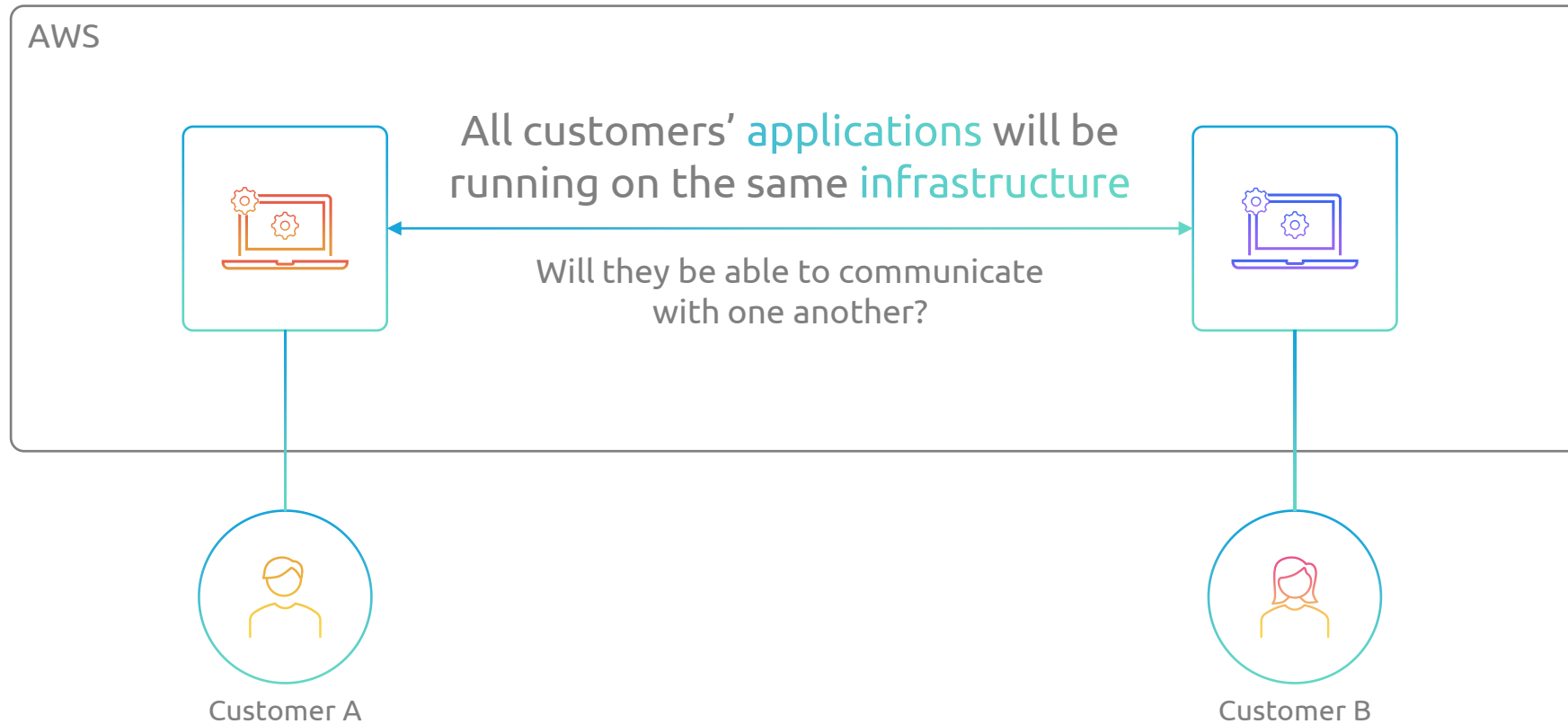
Local Zones are extensions of AWS regions located near users in select metropolitan areas

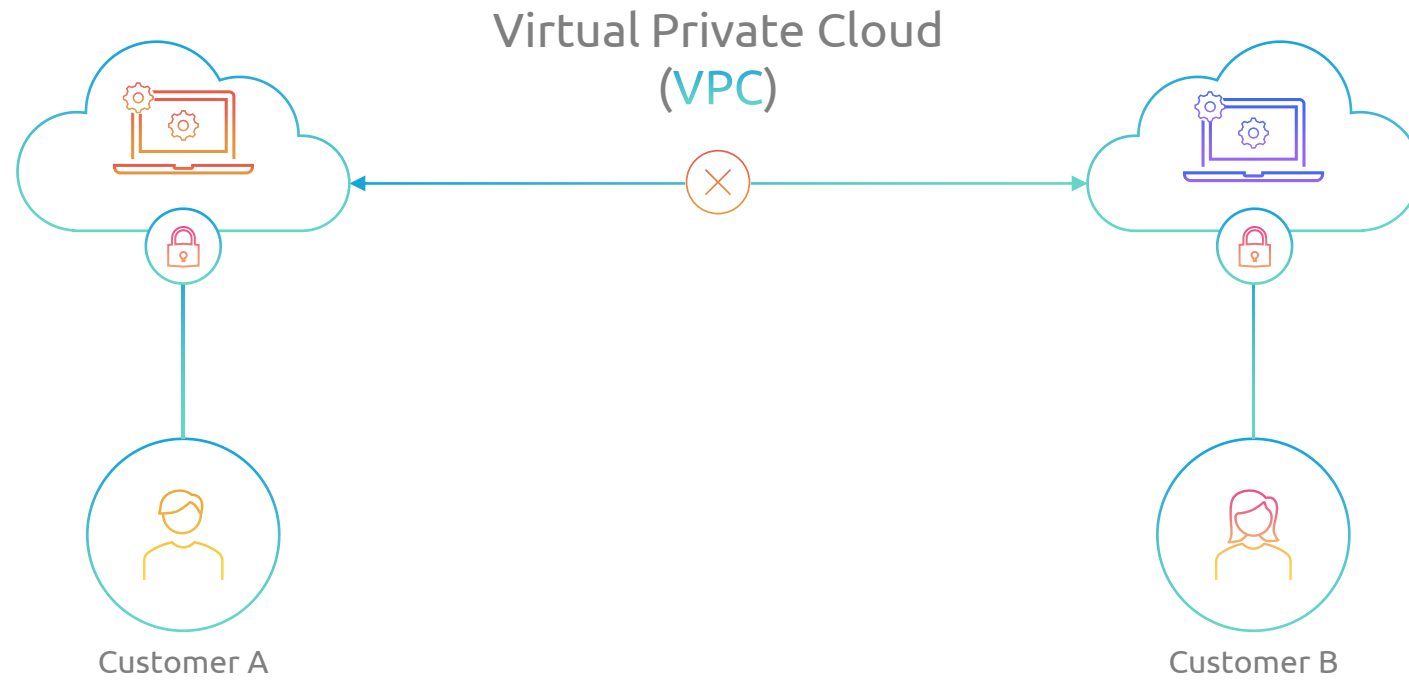


Core AWS Services Networking



AWS Networking





What Is a VPC?



Virtual private cloud (**VPC**) is a secure, isolated network segment hosted within **AWS**



VPC isolates **computing** resources from other **computing** resources available in the **cloud**



It gives the customer full control of the **networking** in the **cloud**



Subnetting (**IP address**)



Routing (**Route tables**)



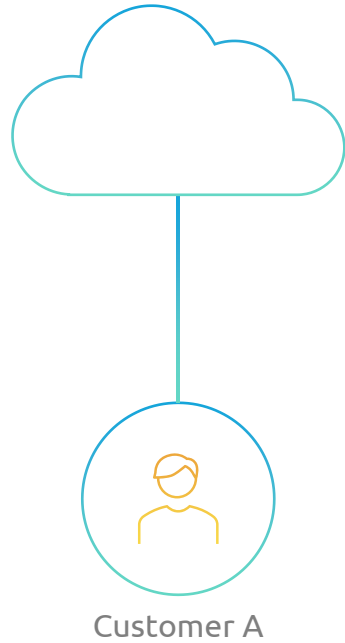
Firewalls
(**NACLs and Security groups**)



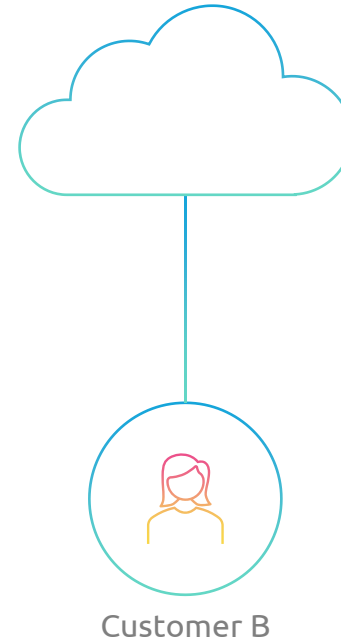
Gateways



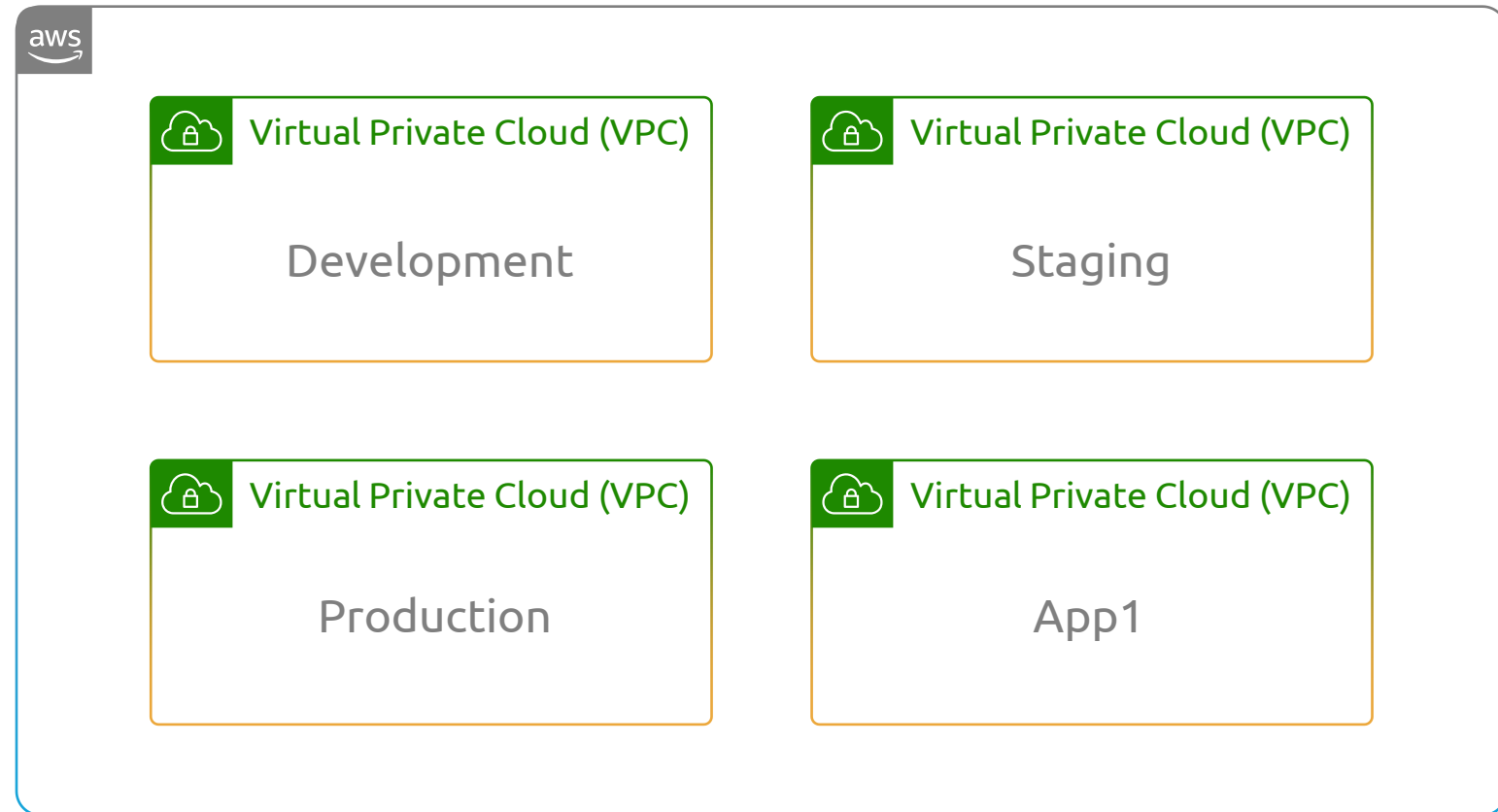
One VPC per Customer?



Does that mean every customer gets one **VPC**?

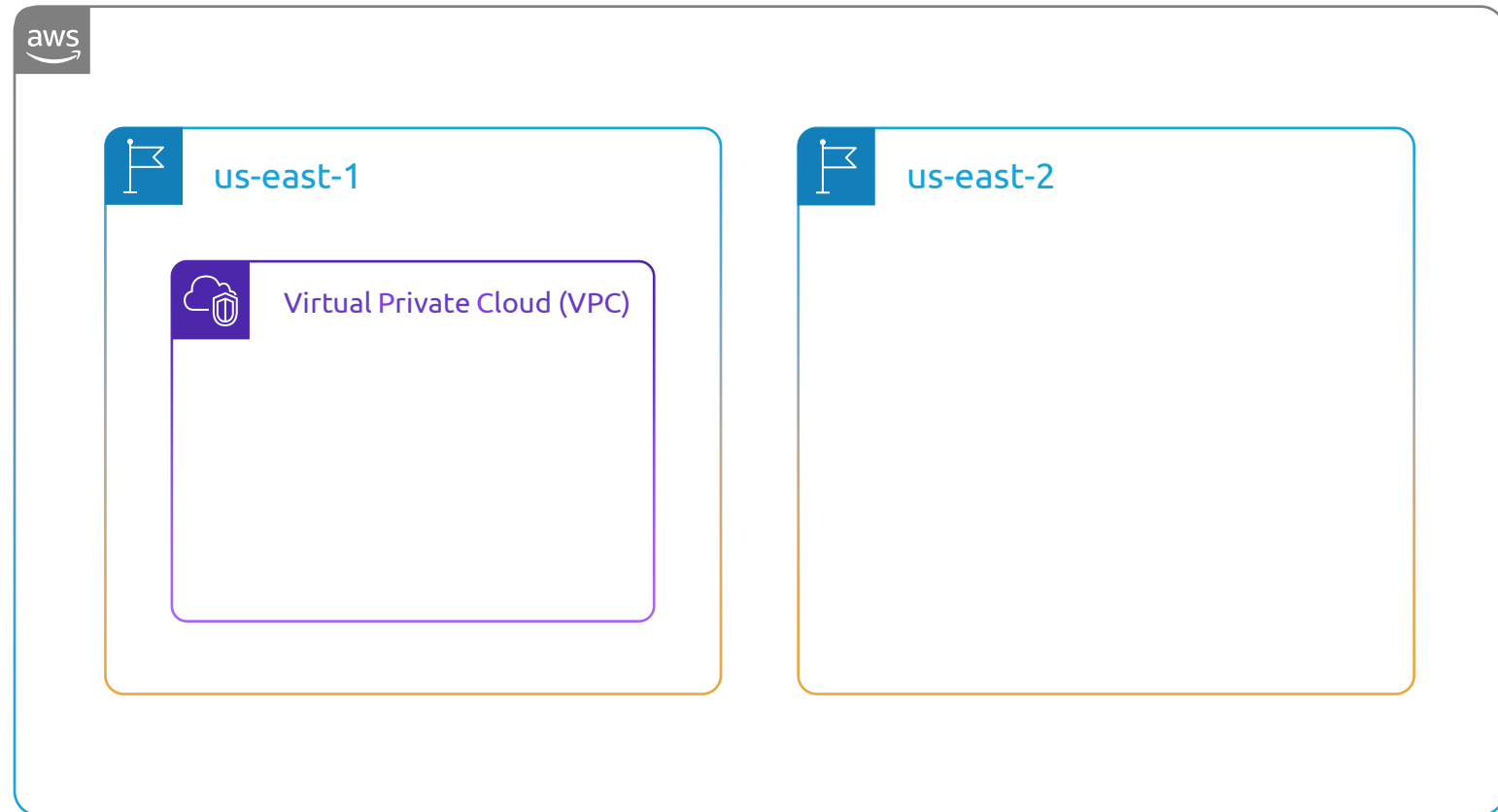


Multiple VPCs



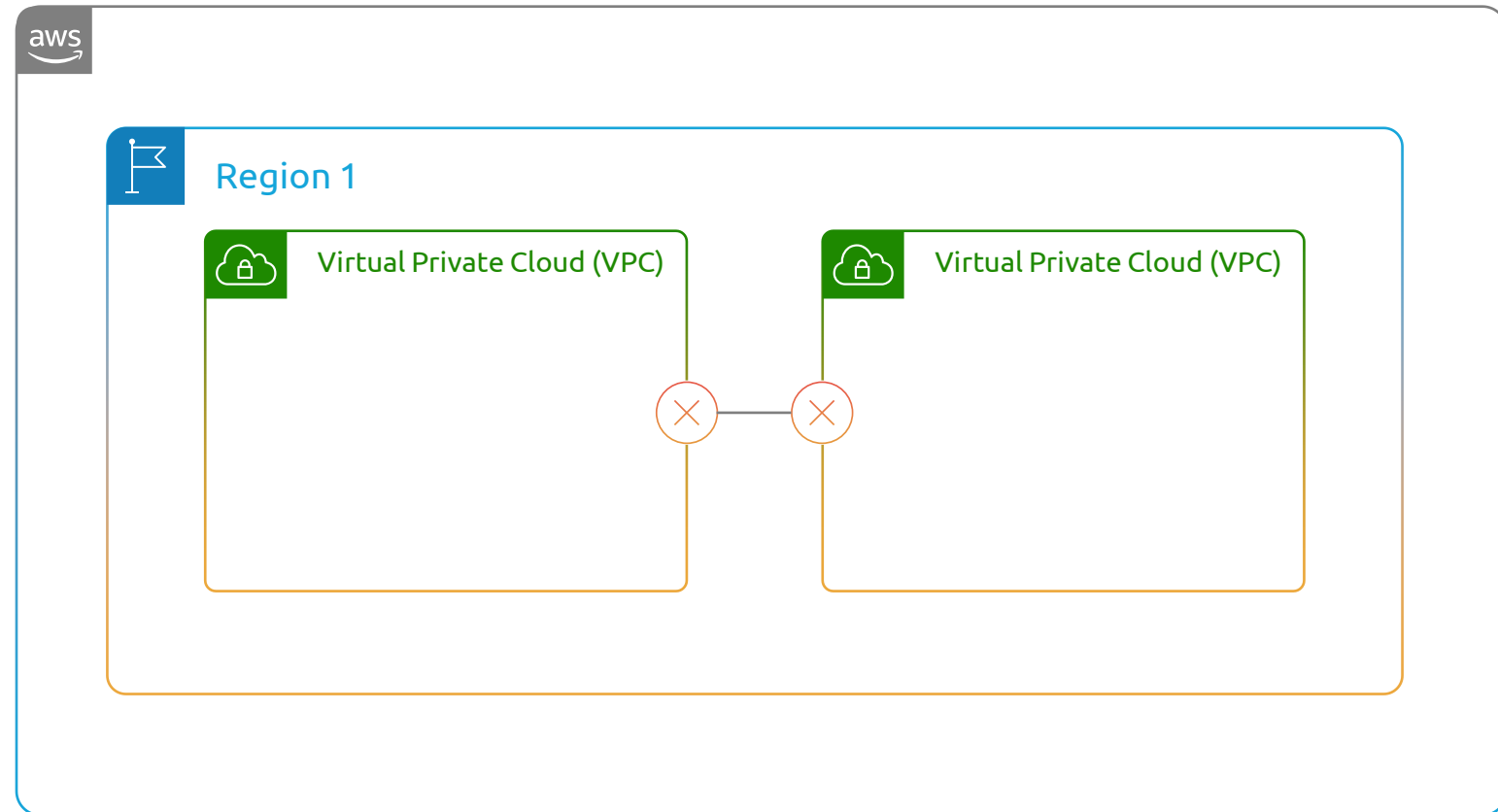
VPCs and Regions

A VPC is specific to a single region



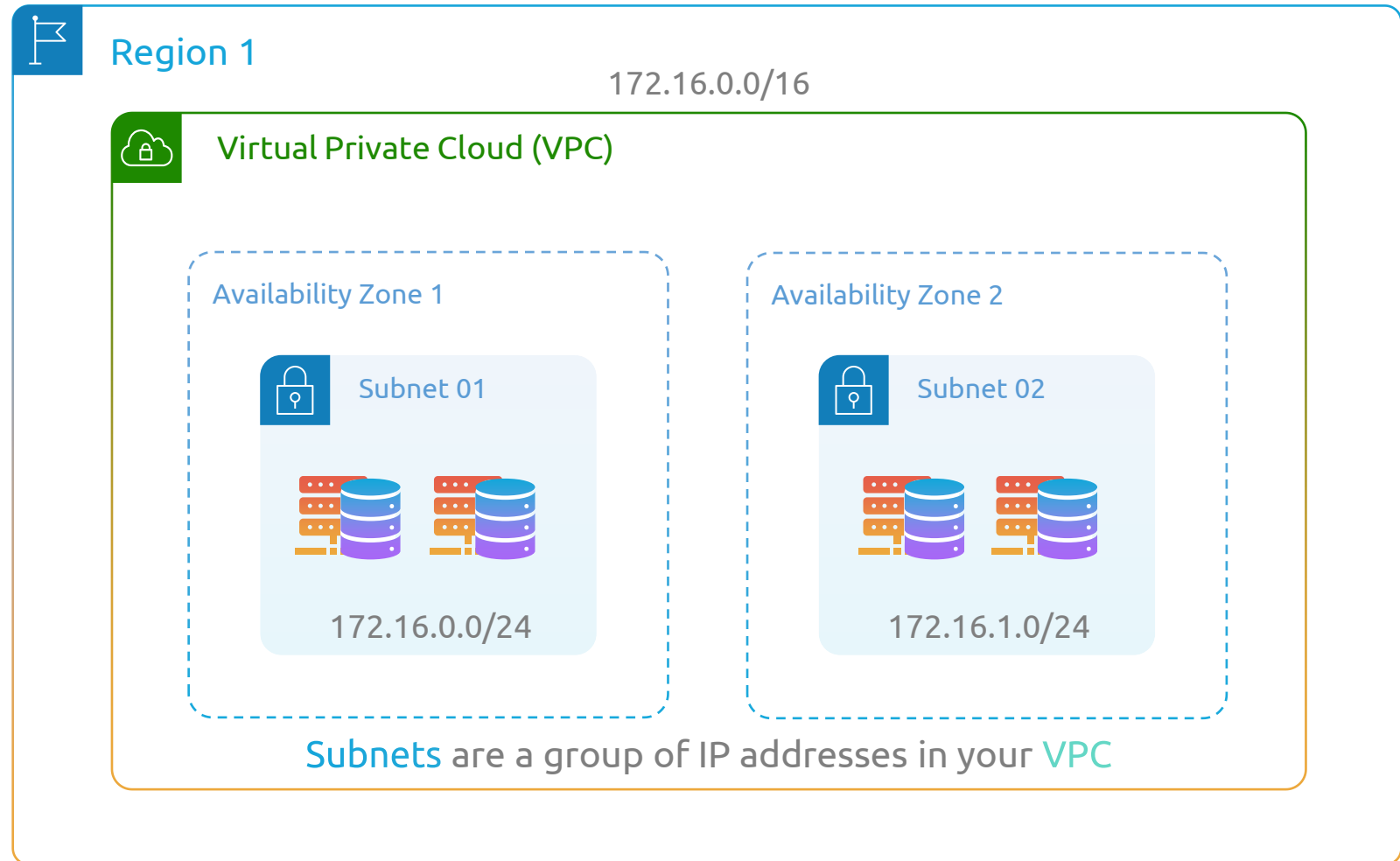
VPC

VPC acts as a
network boundary



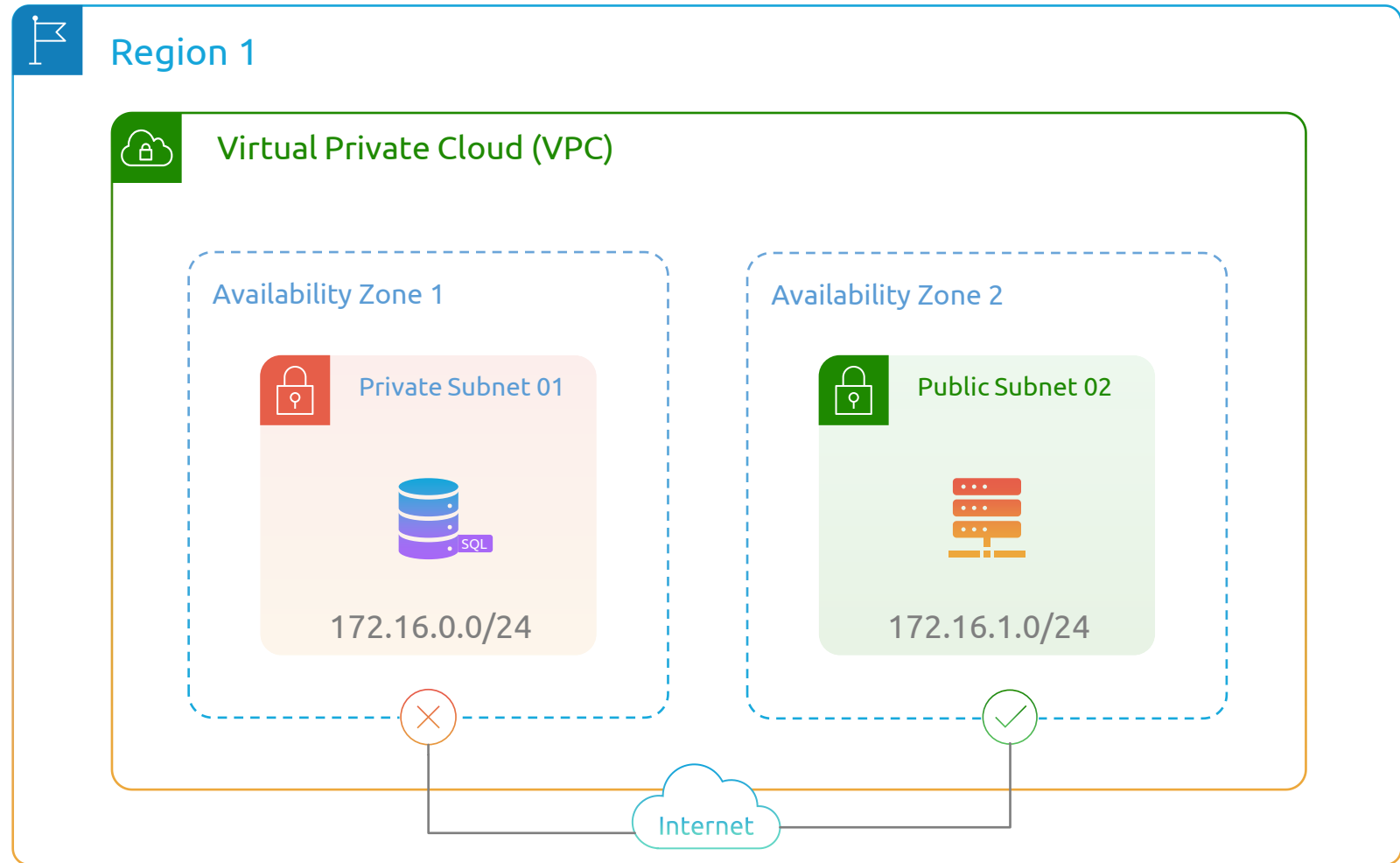
Subnets

A **subnet** resides within a single **availability zone**



Subnets

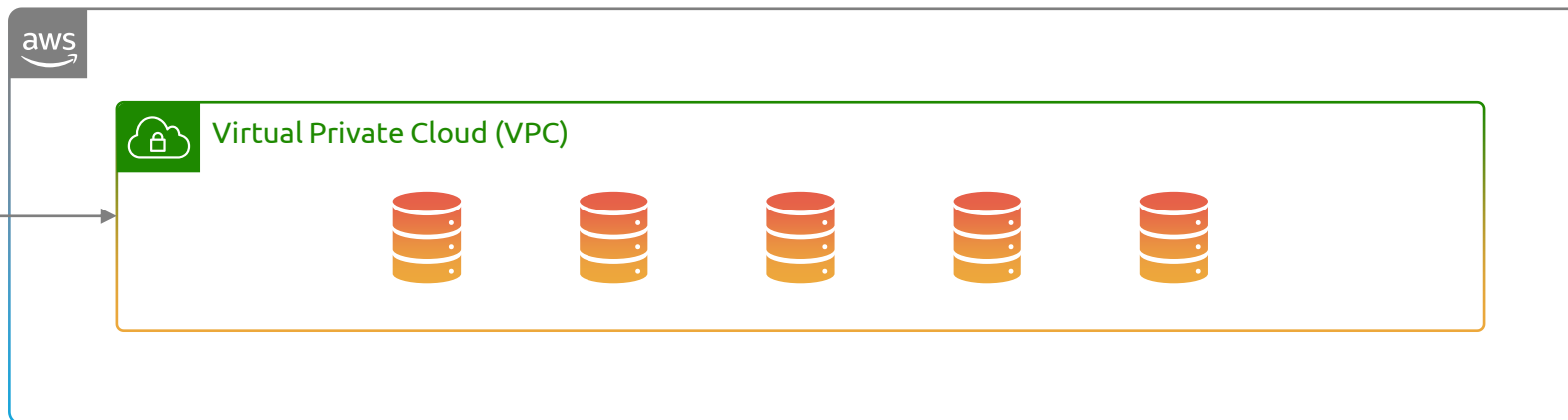
Subnets can be made **public** or **private** to allow external access to **resources** within them



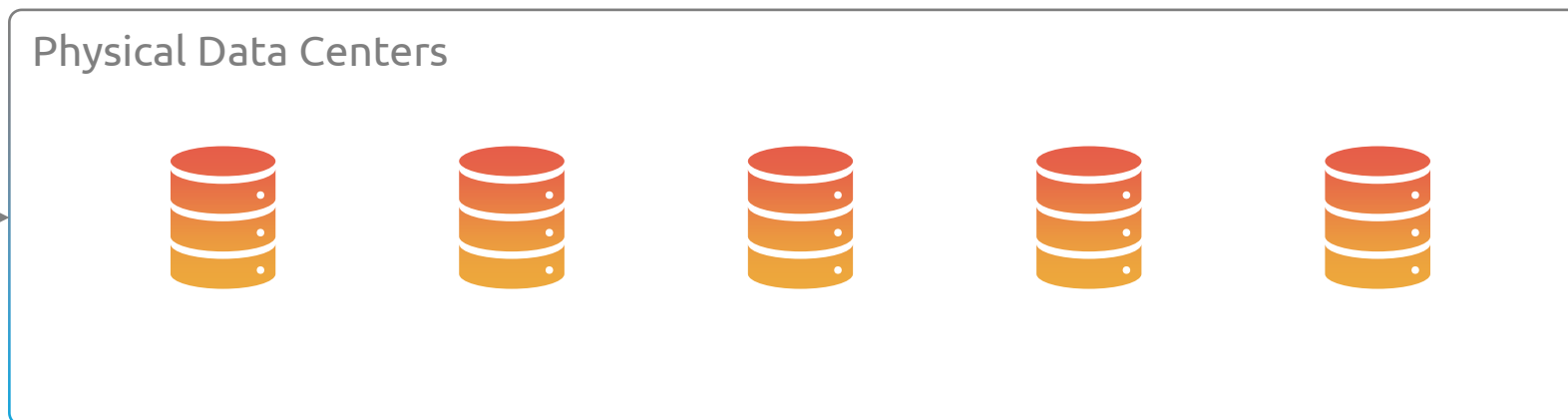


VPC

VPN can be used to connect physical data center to your VPC



AWS cloud acts as an extension of your data center





VPC

- Every **VPC** has a range of IP addresses assigned to it called the **CIDR block**
- A **CIDR** block defines the IP addresses that resources in the **VPC** can use
- A **CIDR** block size can be anywhere from a **/16** to a **/28**
- 192.168.0.0/16:
192.168.0.0 – 192.168.255.255

CIDR = 192.168.0.0/16

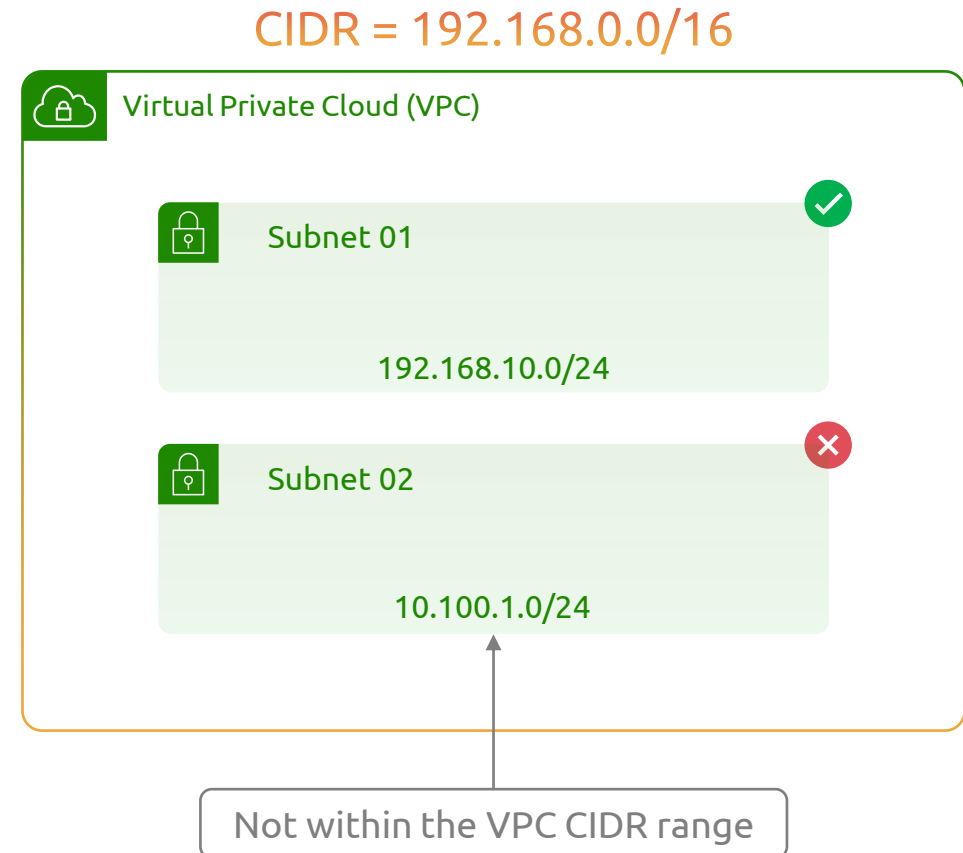


Virtual Private Cloud (VPC)



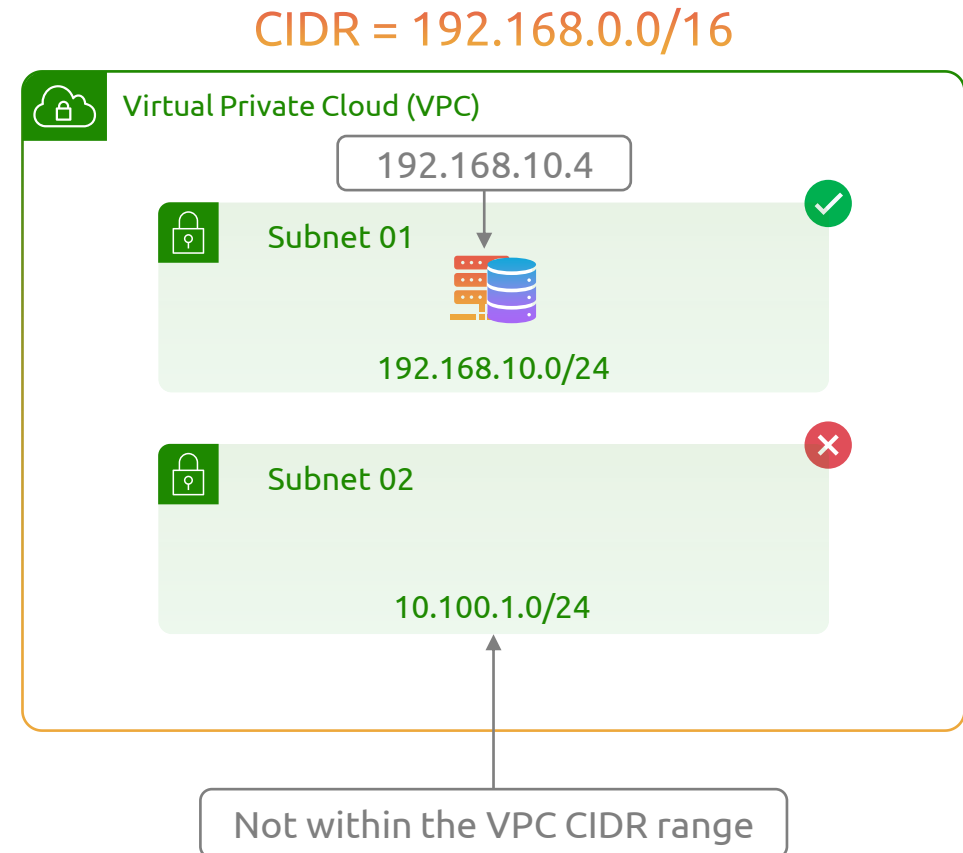
VPC

- Subnets within a VPC must be within the CIDR range
- Subnet block size must be between a /16 and a /28



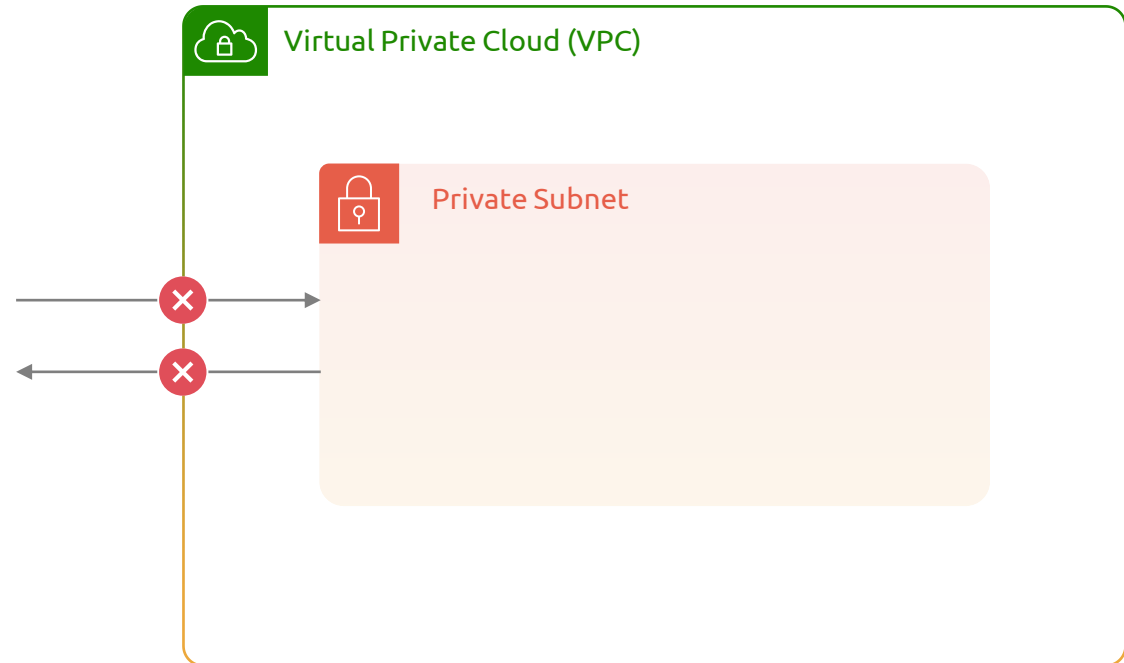
VPC

- The first 4 IP addresses of a subnet are reserved and cannot be used
 - 192.168.10.0: Network address
 - 192.168.10.1 – 192.168.10.3: Reserved for AWS
- The last IP address of a subnet is reserved as the broadcast address
 - 192.168.10.255



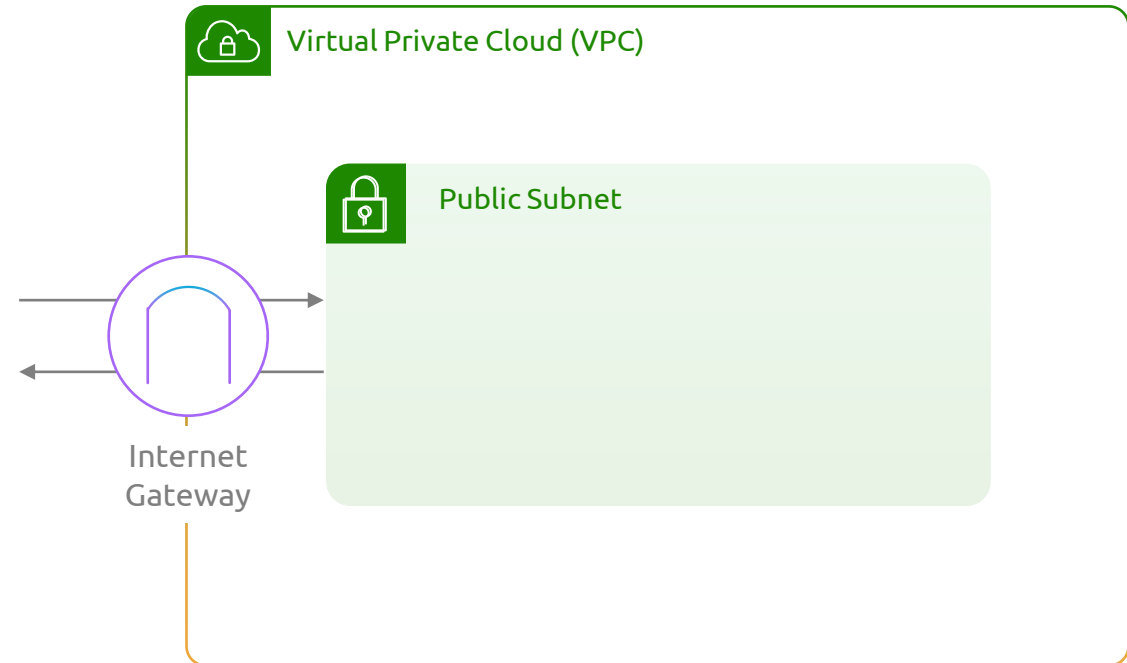
VPC External Connectivity

Subnets when first created are private and not exposed to the internet



Internet Gateway

Internet gateway allows subnets in a VPC to communicate with the Internet and vice versa

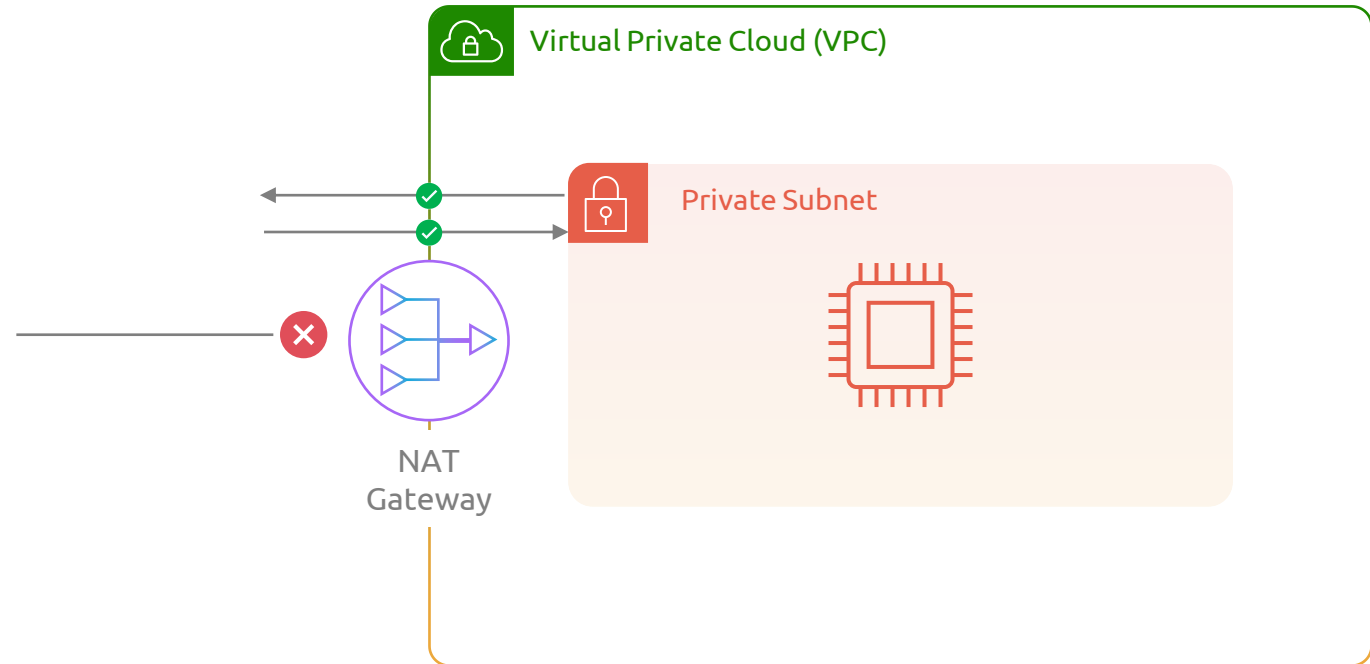


Access to Internet gateway determines whether a subnet is public or private

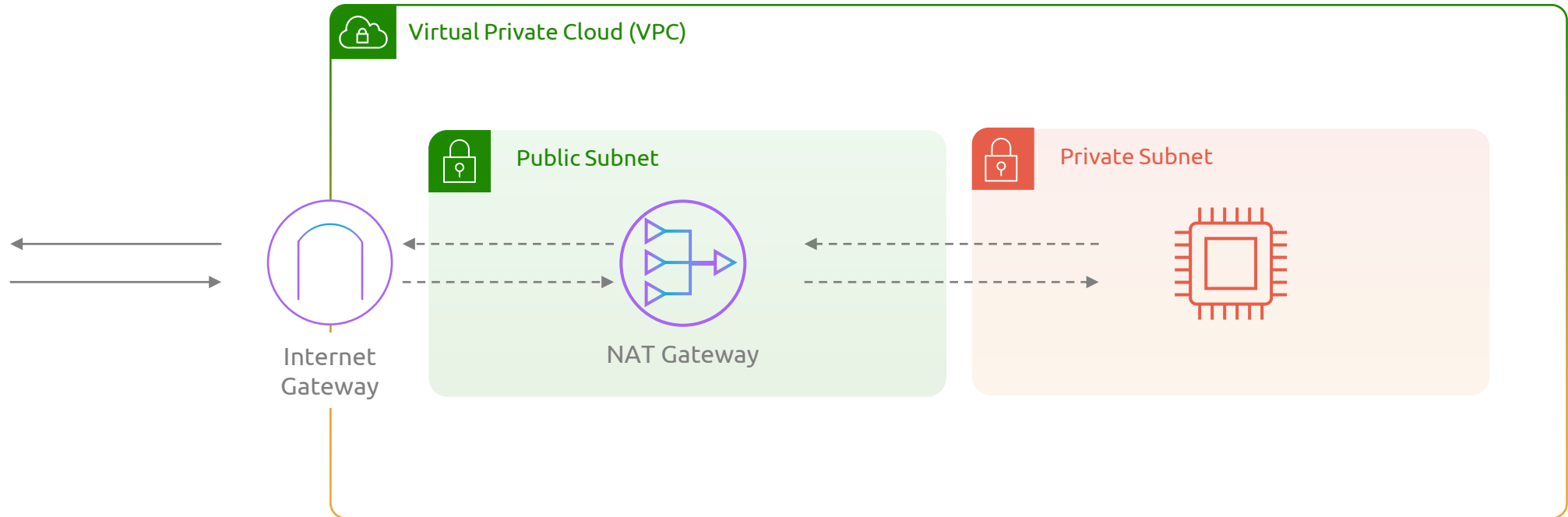


NAT Gateway

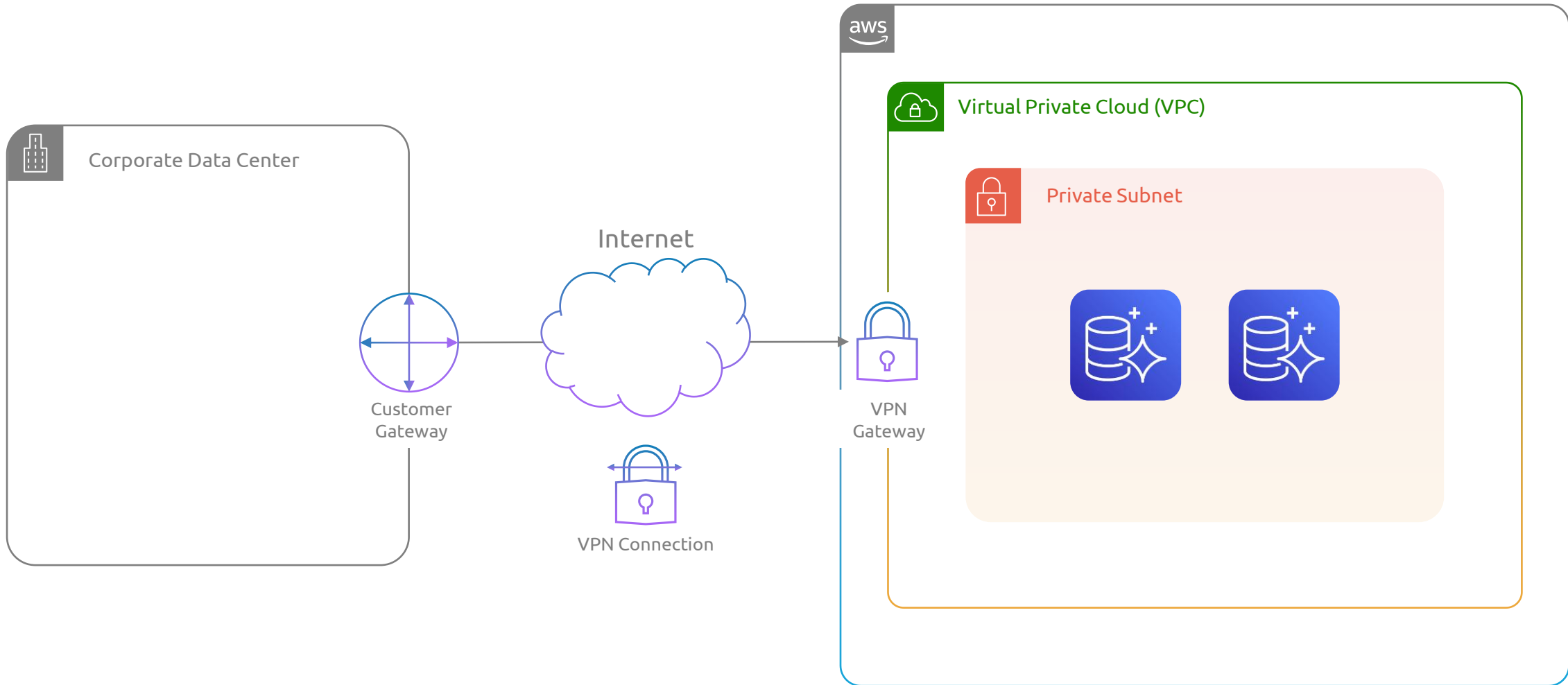
With NAT Gateway,
a connection must be
initiated from within the VPC



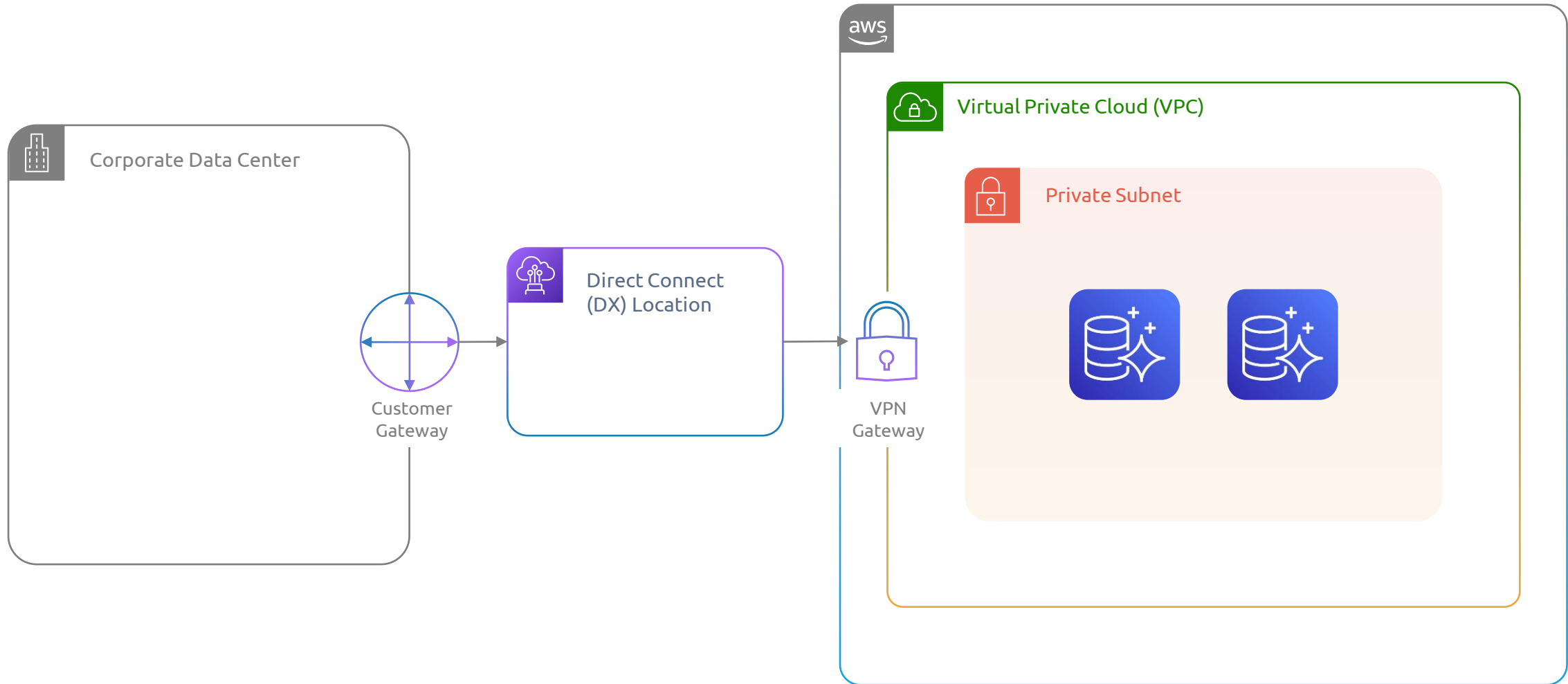
NAT Gateway



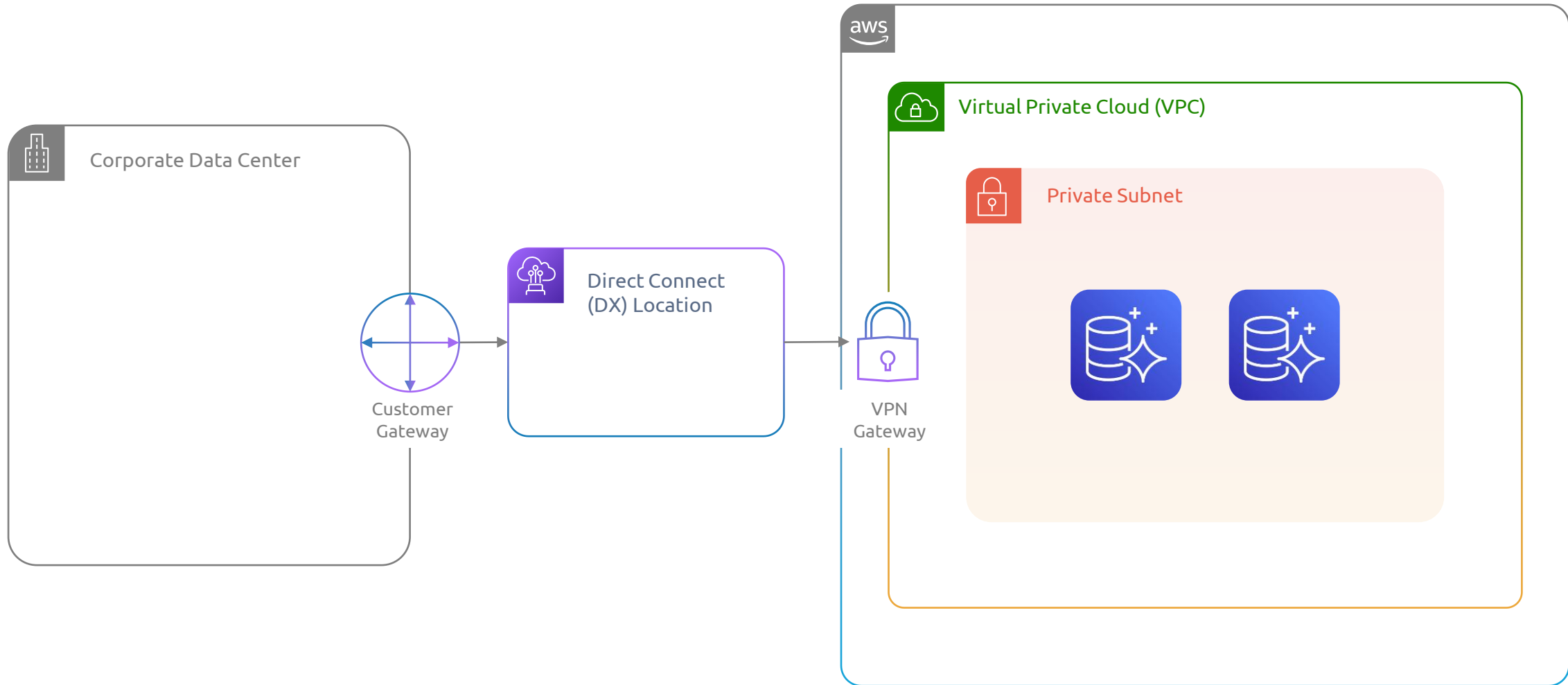
Virtual Private Gateway

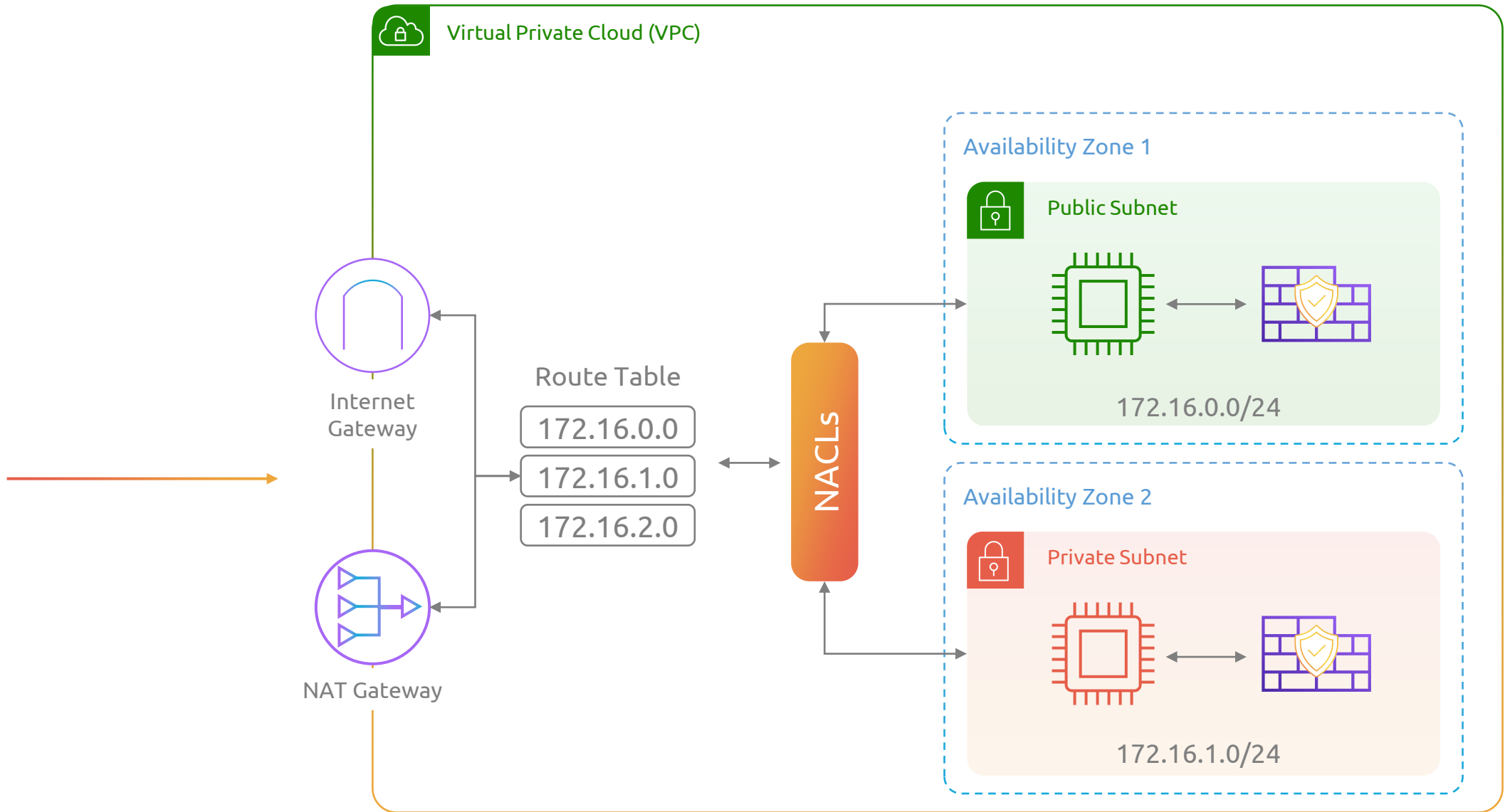


Direct Connect



Direct Connect





AWS Networking - Summary



VPC isolates computing resources from other computing resources available in the cloud



VPCs are isolated to a region



VPC CIDR block defines the IP addresses a VPC can use



Subnets are a range of IP addresses within a VPC



Subnets reside within a single Availability Zone



AWS Networking - Summary



Subnets can be made public/private using Internet Gateways & Nat Gateways



Internet Gateways allow subnets to communicate with internet & vice versa



NAT Gateways allow subnets to talk to the internet but connections must be initiated from within the VPC



Virtual Private Gateway enable secure access to private resources over the internet



Direct Connect(DX) is a direct connection into an AWS regions that provide low latency + high speeds

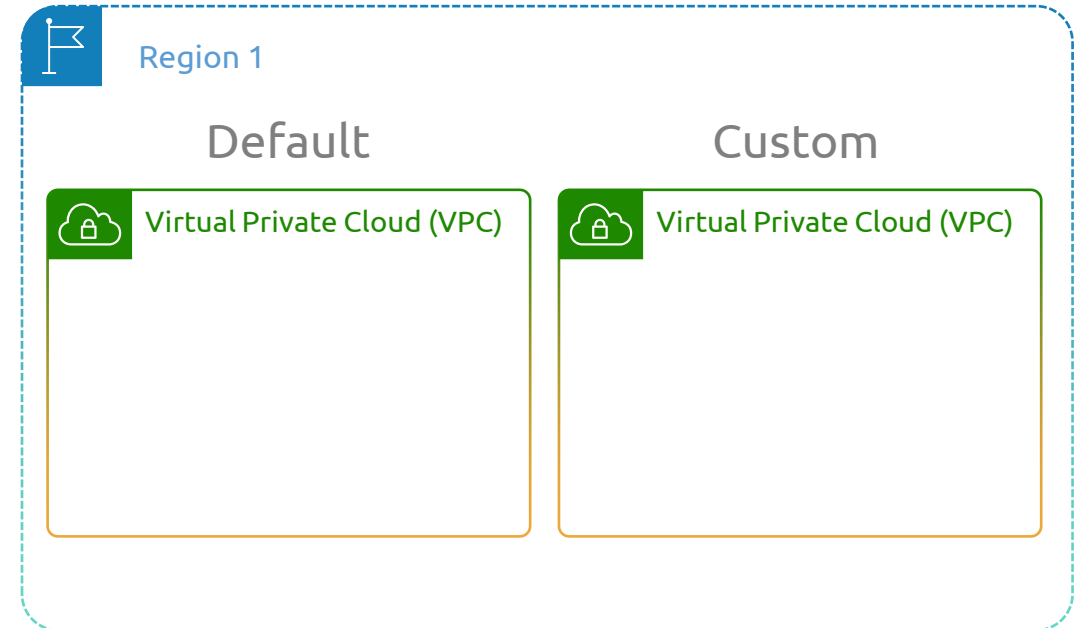


Default VPC



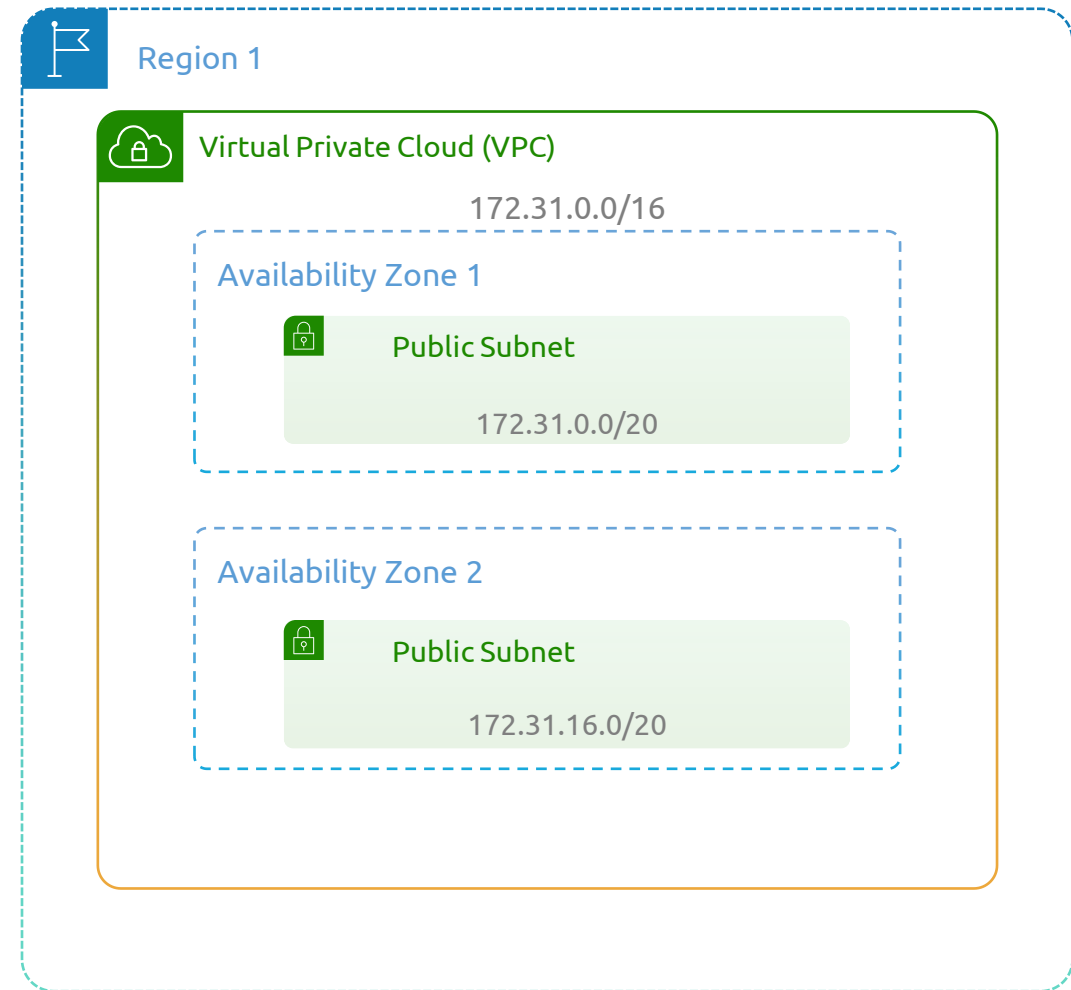
VPCs

- VPCs are of **two** types
 - Default
 - Custom
- Every account has a default VPC in each region
- Default VPCs have configurations defined by AWS
- Custom VPCs allow you to create/modify all configurations associated with the VPC



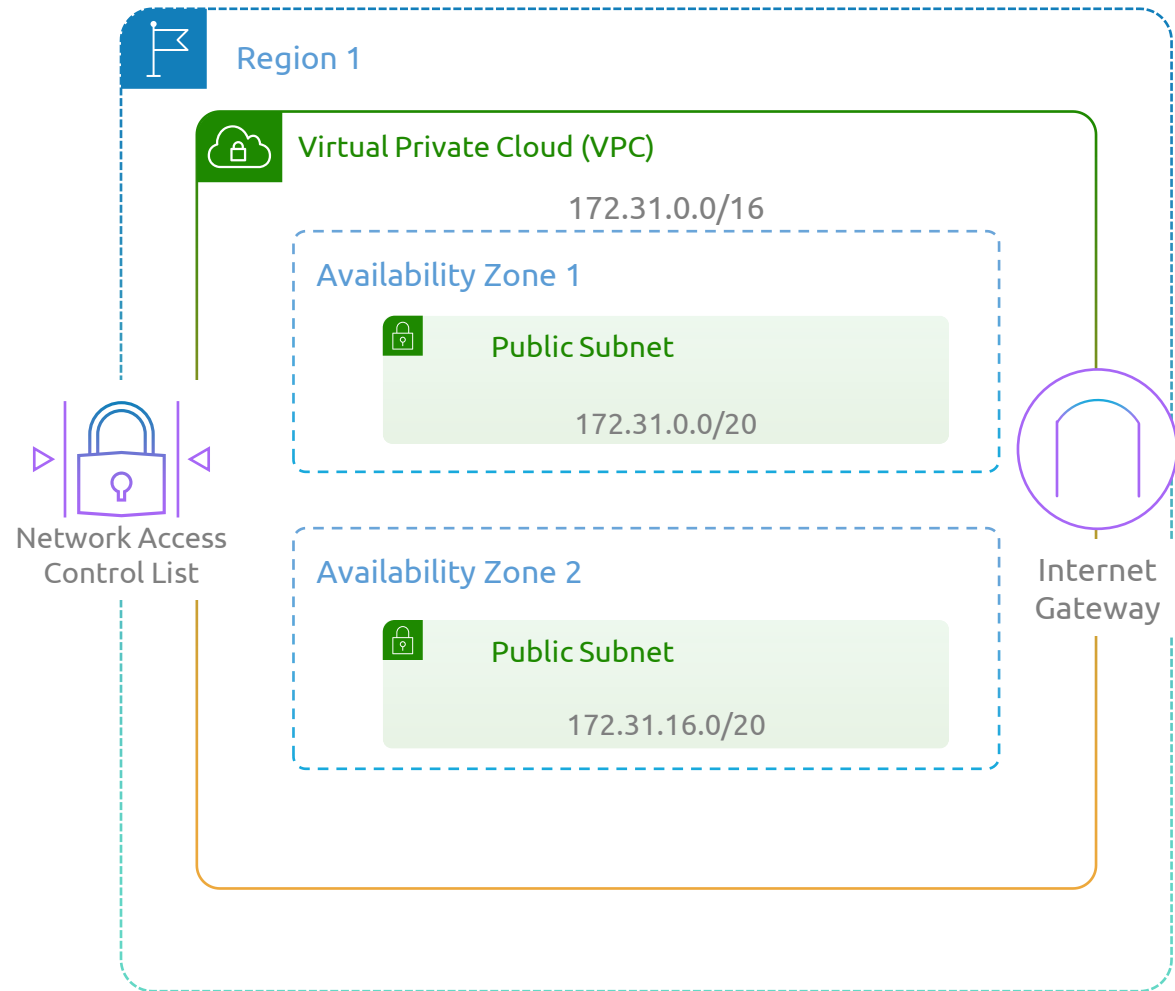
Default VPC

- One default VPC per **region**
- /16 IPv4 **CIDR** block 172.31.0.0/16 (65,536 addresses)
- /20 default **subnet** in each Availability Zone (4,096 addresses)



Default VPC

- Internet gateway attached to the VPC
- A route that points all traffic (0.0.0.0/0) to the internet gateway
 - Devices in these default subnets will be accessible from the internet
- Default security group
- Default network access control list



Default VPC - Summary



Every region has a Default VPC with default subnets, Security Groups, and NACLs



The CIDR block for the Default is 172.31.0.0/16



The Default VPC and its subnets have outbound access to the Internet by default.



One default subnet in each Availability Zone



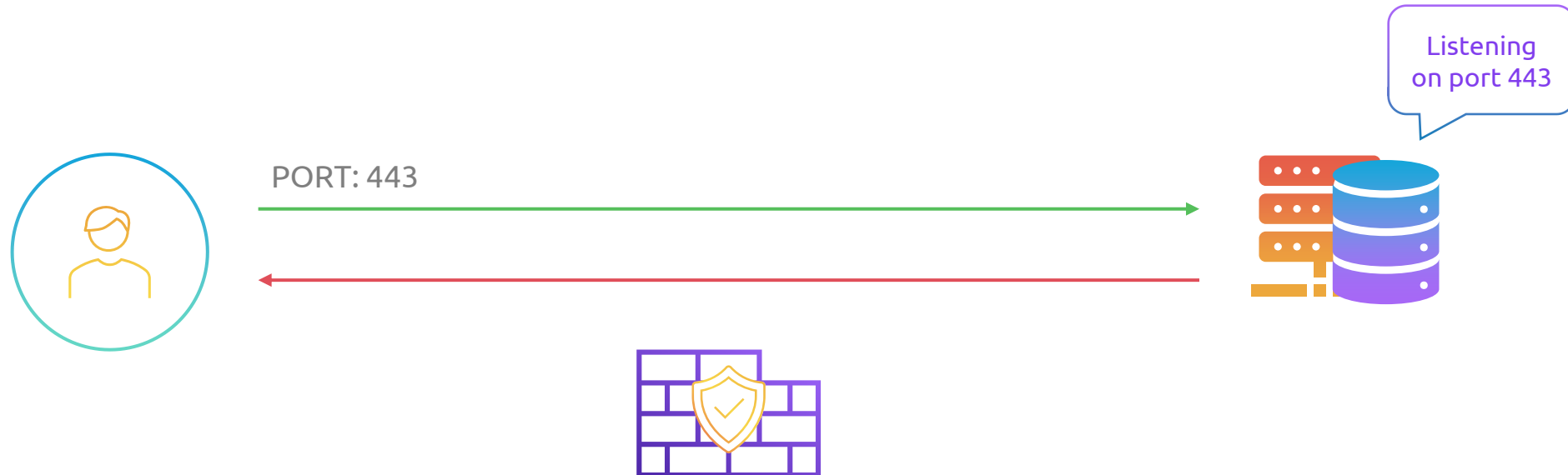
The Security Groups allow outbound and the NACLs are open in both inbound and outbound directions.



Firewalls



Stateless Firewalls



Firewalls monitor traffic and only allow traffic permitted by a set of predefined rules



Stateless Firewalls

Firewall rules are broken down into **inbound** and **outbound** rules

IP/Port	Action
443	Allow
1024-65535	Allow



PORT: 443 Inbound

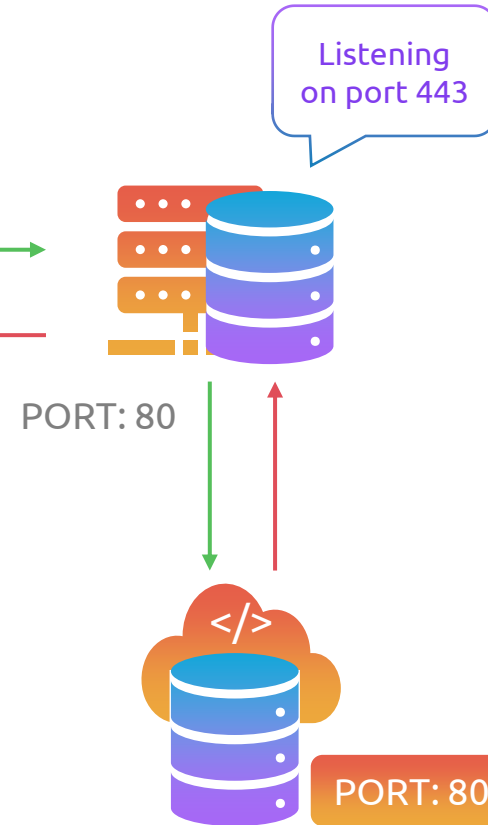


Outbound



IP/Port	Action
1024-65535	Allow
80	Allow

Stateless firewalls must be configured to allow both **inbound** & **outbound** traffic



Stateful Firewalls

Stateful firewalls are intelligent enough to understand which request and response are part of the same connection



PORT: 443 Inbound

IP/Port	Action
443	Allow
1024-65535	Allow



I know this a response to the original request

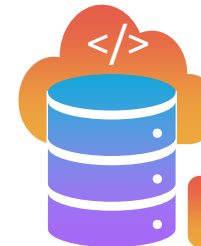
Listening on port 443



PORT: 80

Outbound

IP/Port	Action
1024-65535	Allow
80	Allow



PORT: 80

If a request is permitted, the response is automatically permitted as well in a stateful firewall

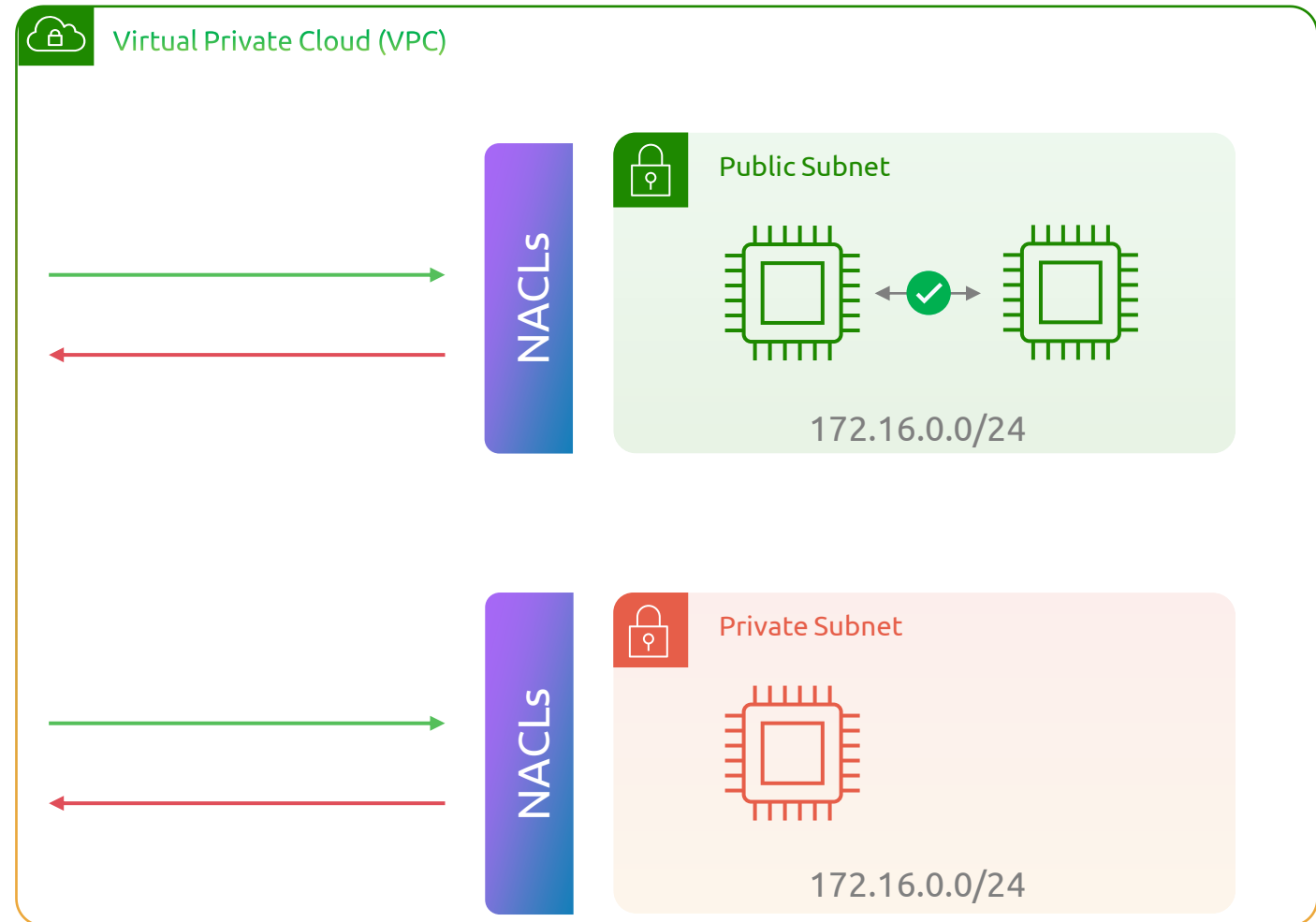


Network Access Control List (NACL)

NACLs filter traffic **entering** and **leaving** a **subnet**

NACLs do not filter traffic within a subnet.

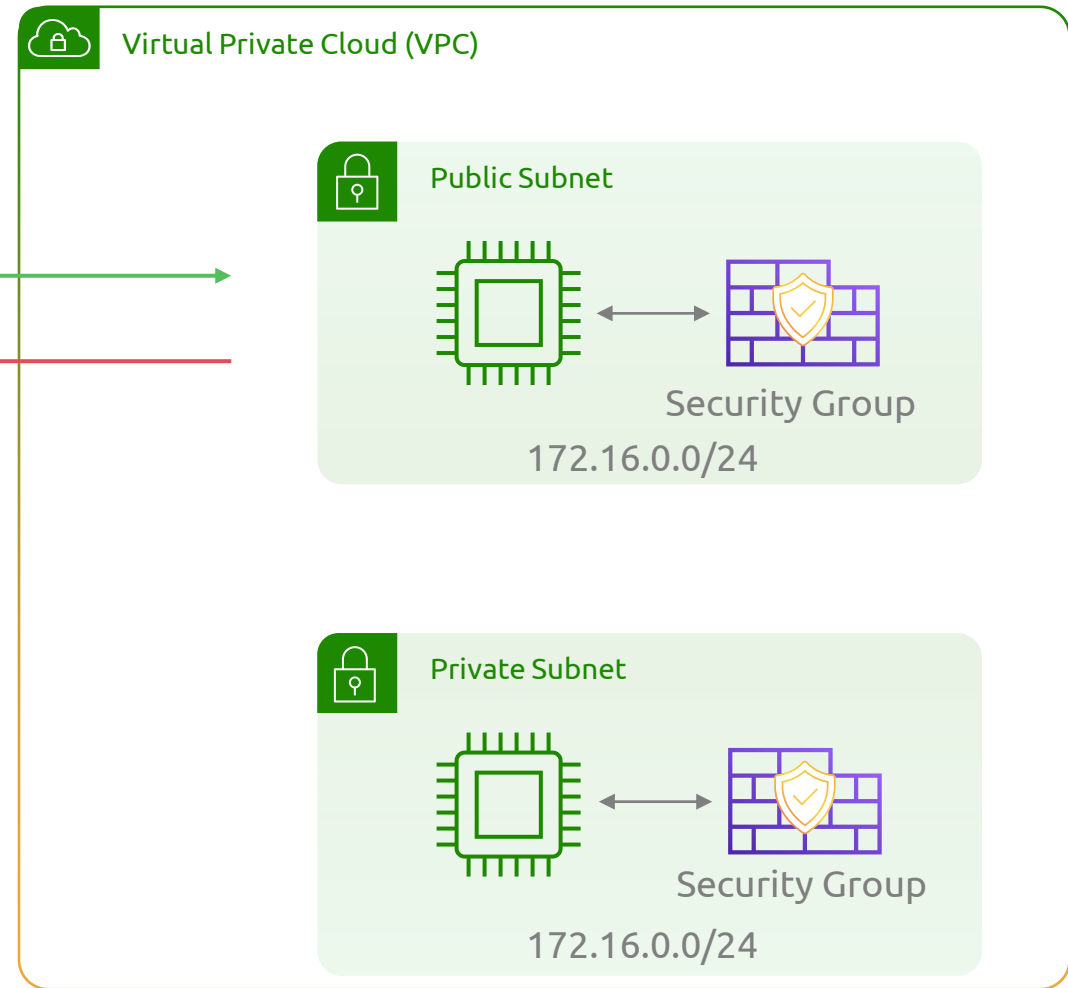
NACLs are **stateless firewalls**, so rules must be set for both **inbound** and **outbound** traffic



Security Group

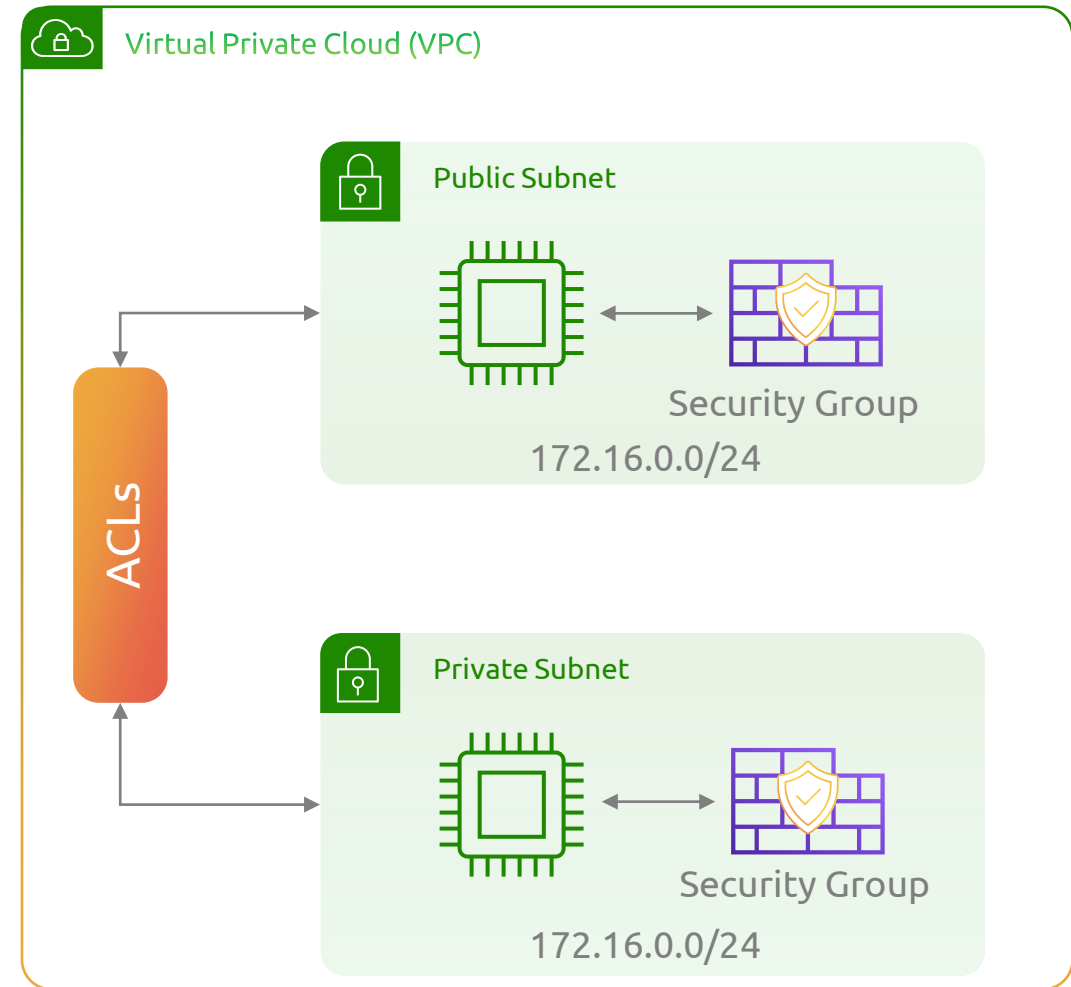
Security Groups act as **firewalls** for individual **resources** (EC2, LB, RDS)

Security Groups are stateful, so only the **request** needs to be allowed



NACLs vs Security Groups

- **NACLs** monitor traffic entering and leaving a **subnet**
- **NACLs** are stateless **firewalls**
 - Traffic must be permitted in both **ingress** and **egress** directions
- **Security groups** act as personal **firewalls** for individual **resources**
- **Security groups** are stateful
 - Only the direction of the **request** needs to be permitted
 - The **response** will automatically be permitted as well



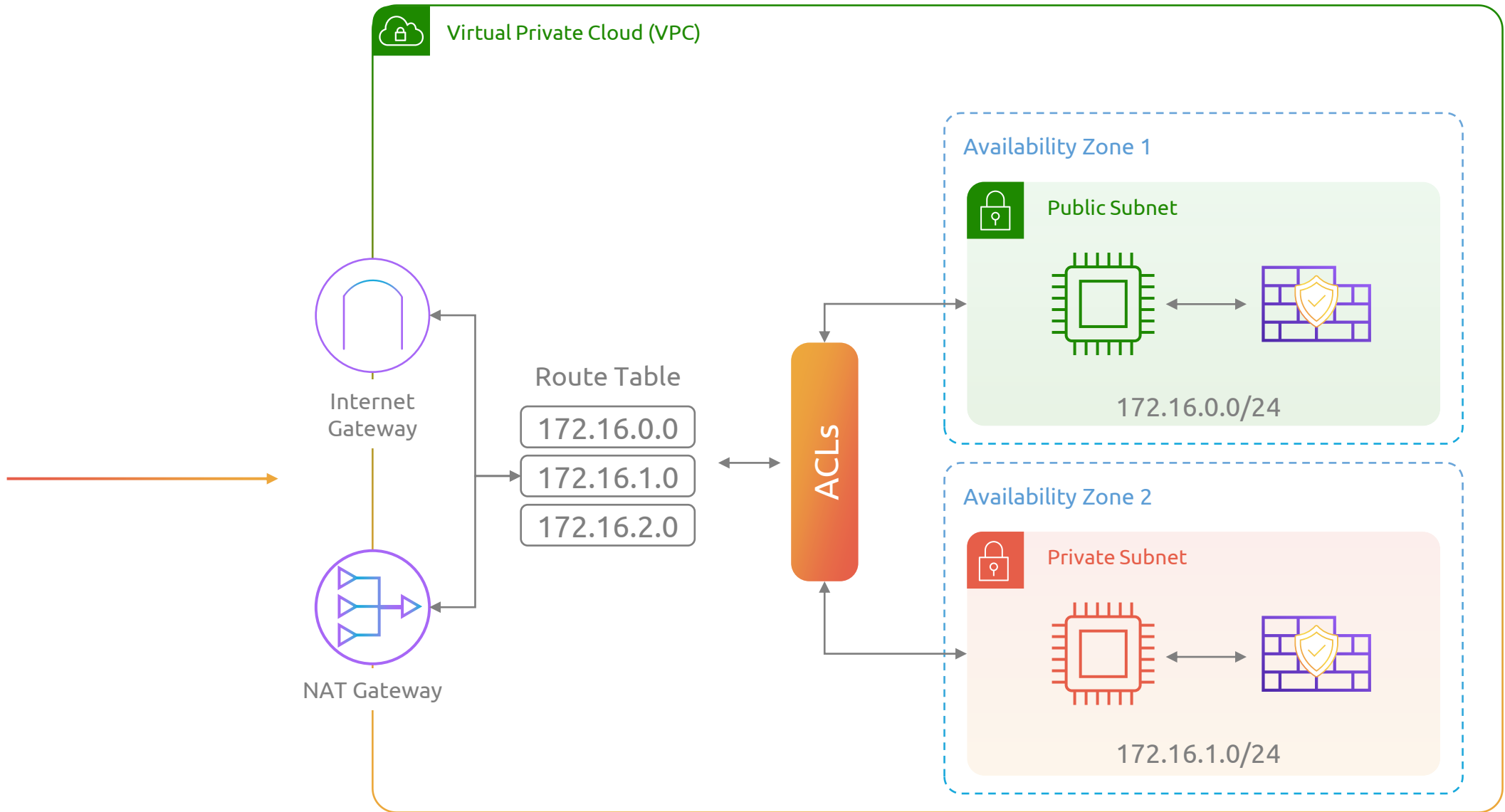
Firewalls - Summary

- ✓ Stateless firewalls require traffic to be explicitly permitted inbound & outbound
- ✓ Stateful firewalls are intelligent firewalls that track requests and allow response
- ✓ Network ACLs filter traffic entering & leaving a subnet
- ✓ Network ACLs are stateless firewalls
- ✓ Security Groups act as firewalls for individual resources such as EC2, NICs, and other network objects
- ✓ Security Groups are stateful firewalls



AWS Networking



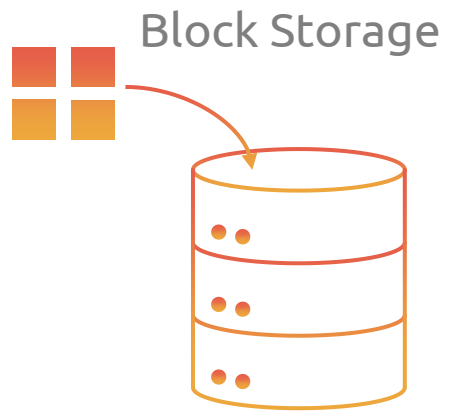


Core AWS Services

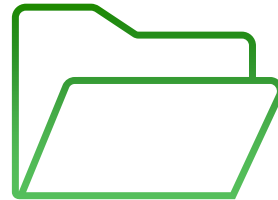
Storage



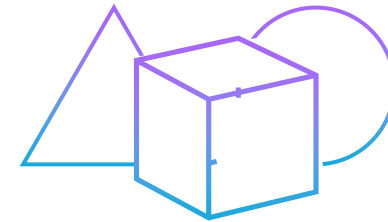
Types of Storage



File Storage



Object Storage

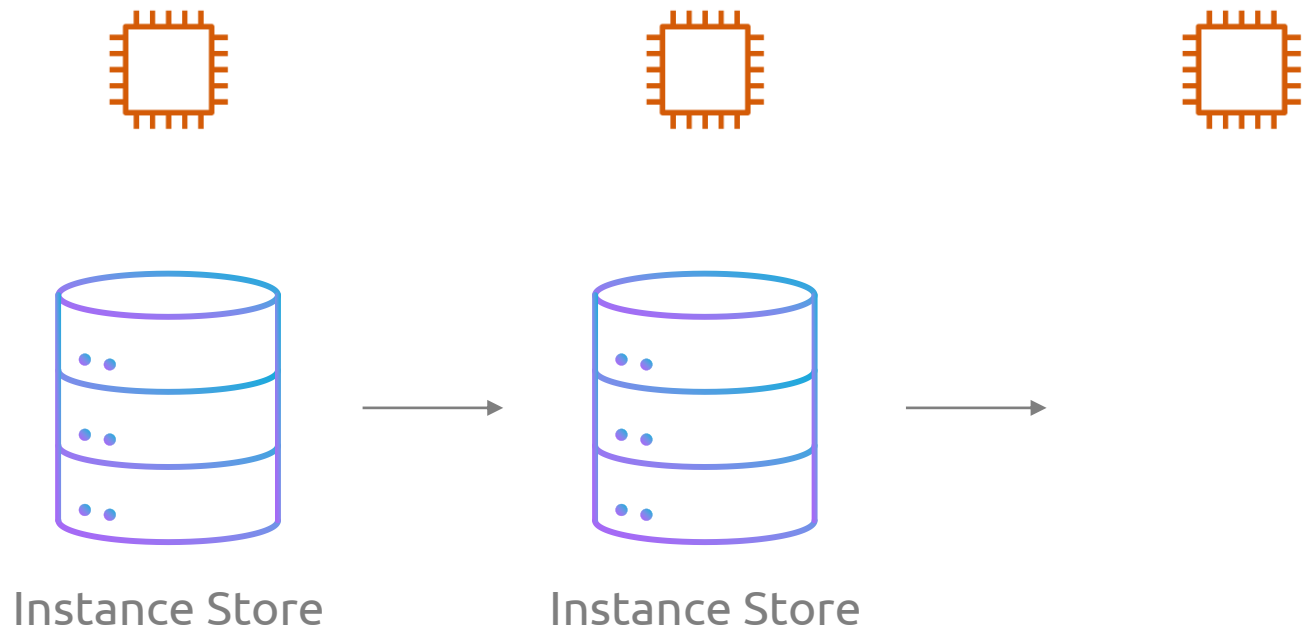


Block Storage





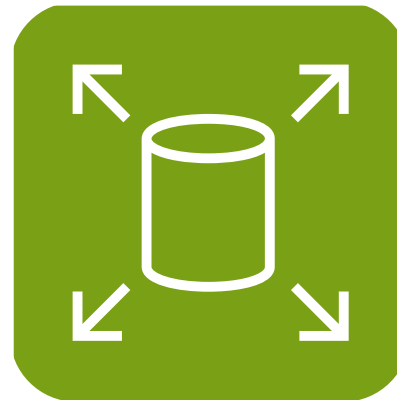
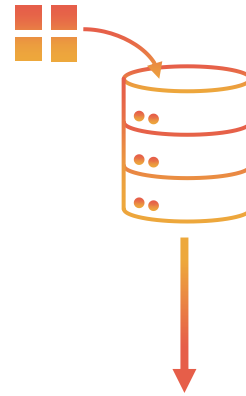
Instance Store





Elastic Block Store

Block Storage

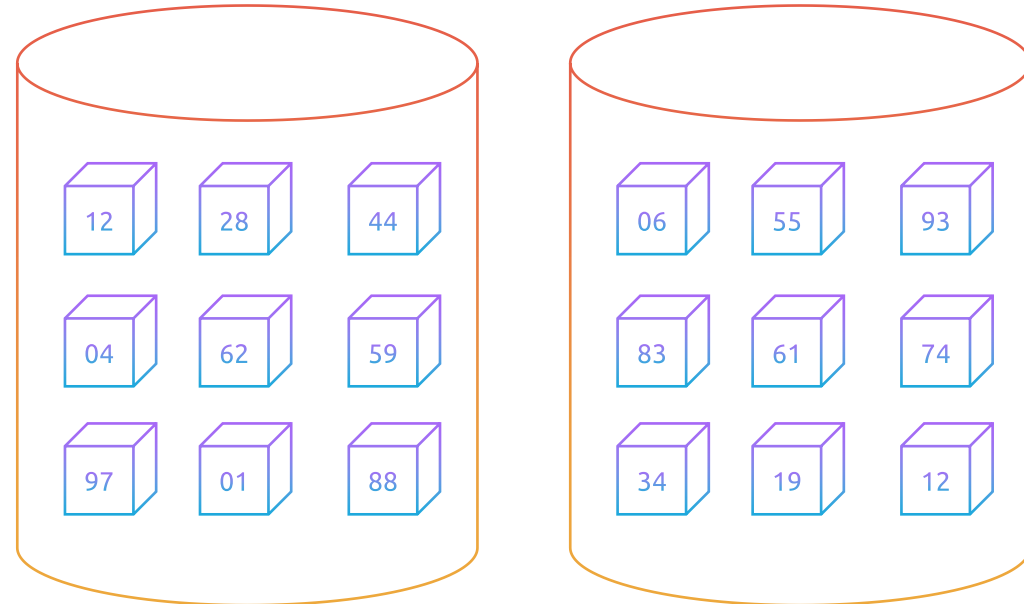
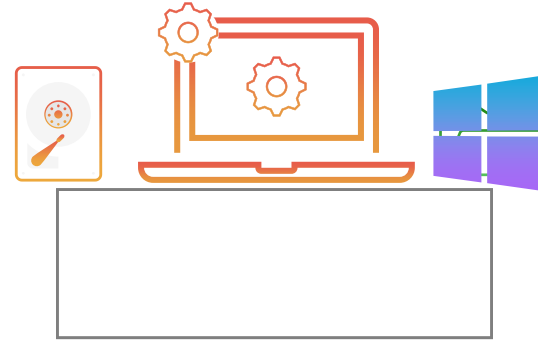


Amazon Elastic Block Store
(Amazon EBS)

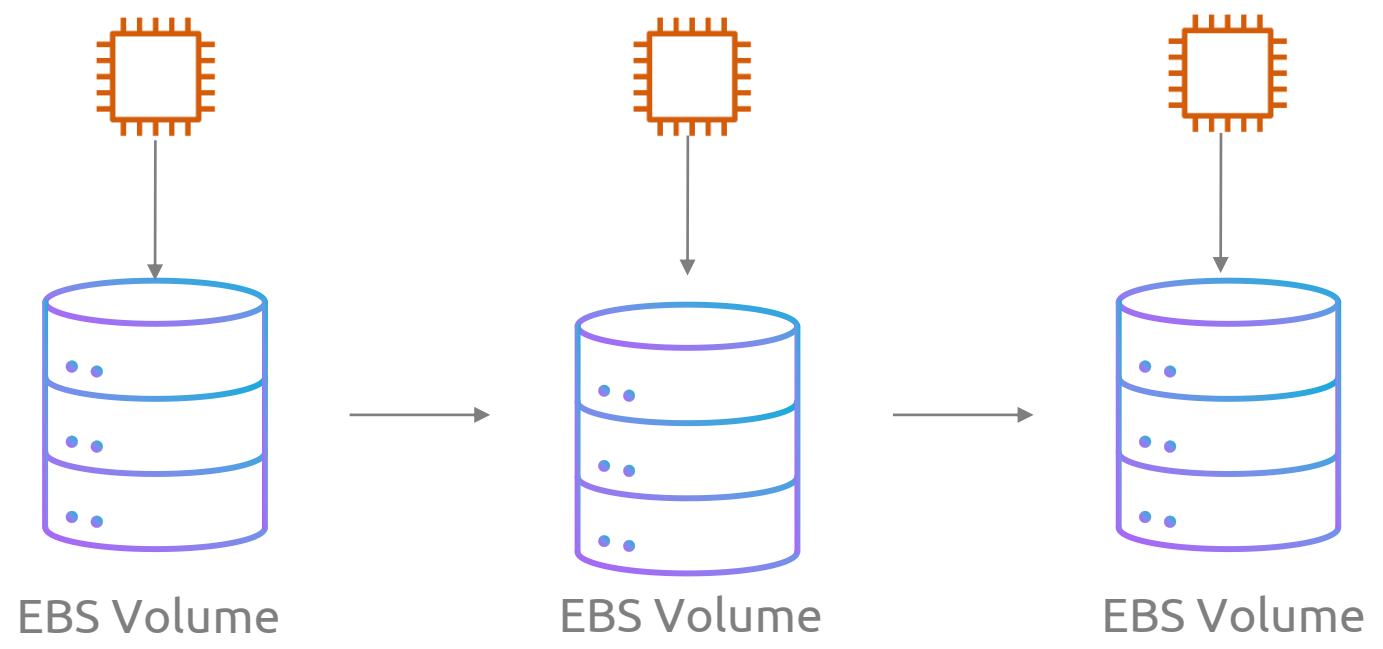


Block Storage

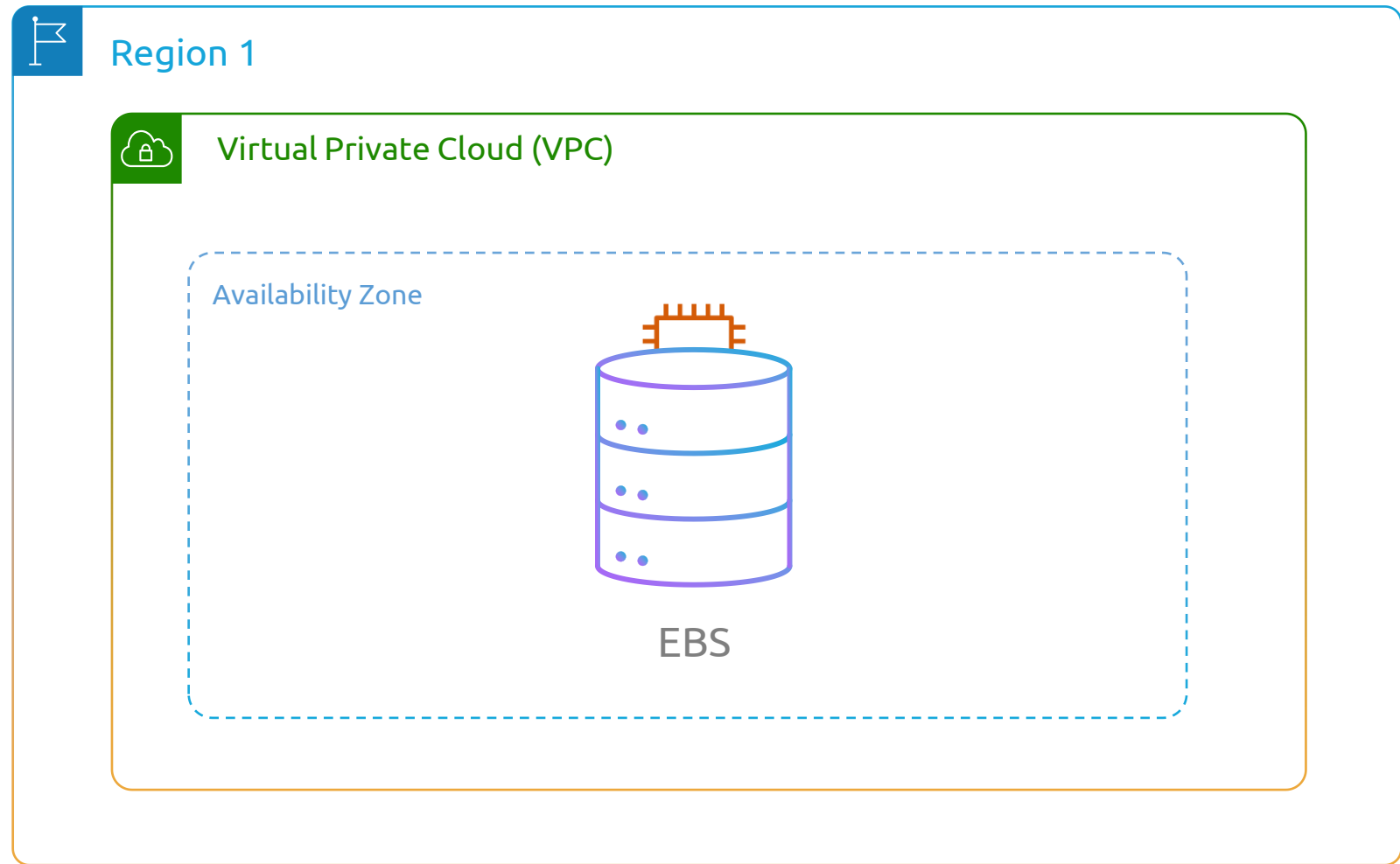
- **Block** storage breaks up data into **blocks** and then stores those **blocks** as separate pieces, each with a unique **identifier**
- A collection of **blocks** can be presented to the **OS** as a **volume**
 - The operating system then creates a filesystem on top of it
- A collection of **blocks** can be presented as a hard drive
 - Block storage is bootable, which means **operating systems** can be installed on it



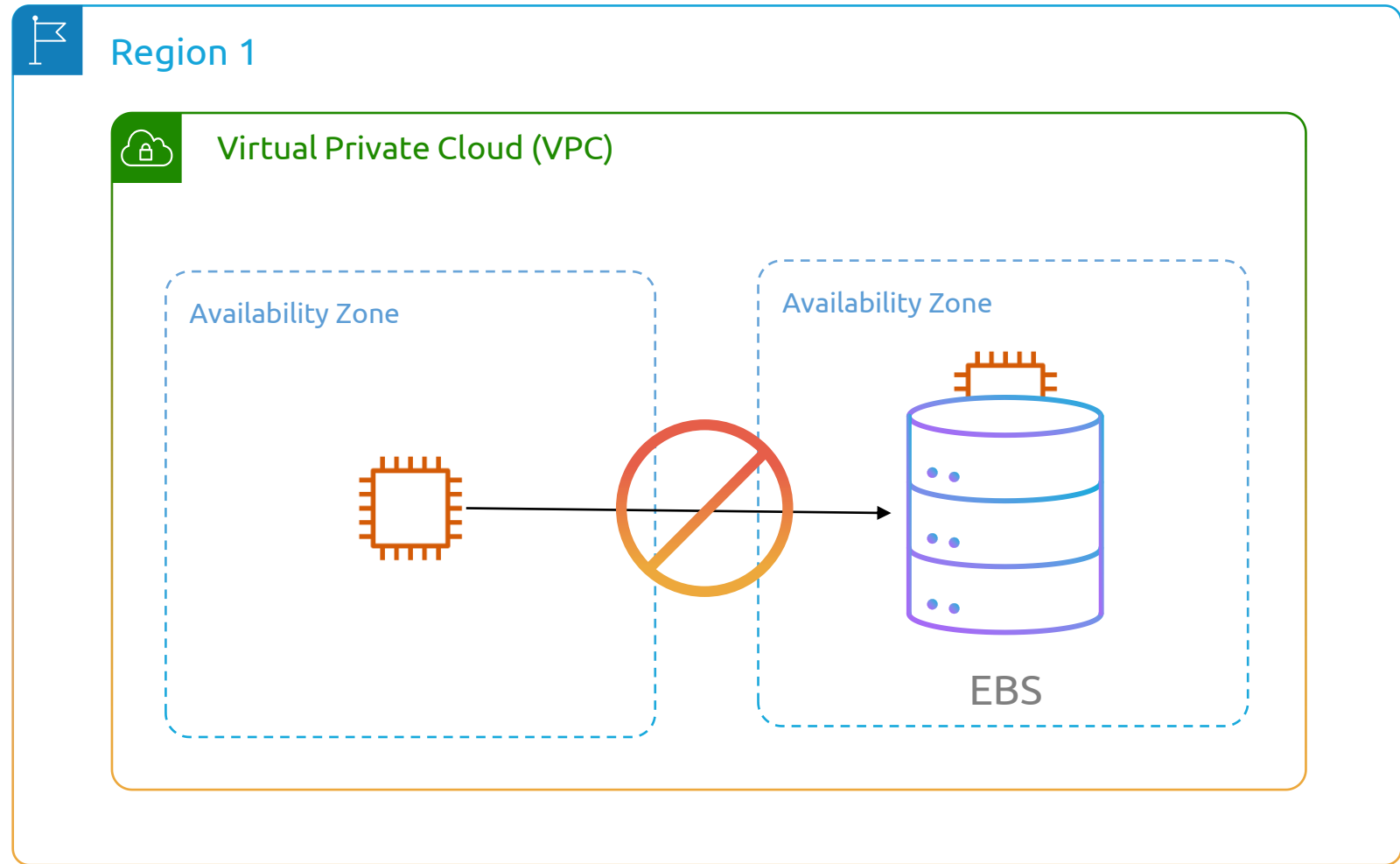
EBS Volumes



EBS Volumes



EBS Volumes



Block Storage - Summary



A collection of blocks can be presented to the OS as a volume



EBS Volumes can be mounted & booted



EBS Volumes are within an Availability Zone



Instance Stores are removed when EC2 instances are stopped/started



File Storage





Elastic File System

File Storage

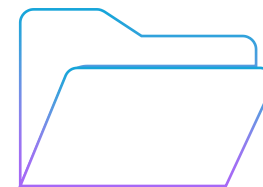


Amazon Elastic File System (Amazon EFS)

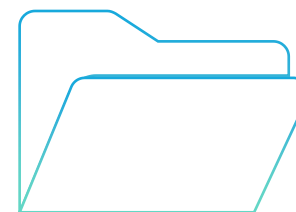




File Storage



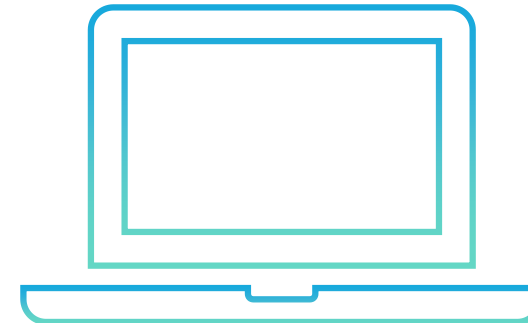
- **File** storage – Stores data in a hierarchical structure of **files** and **folders**
 - Similar to how your computer stores files/folders





File Storage

- **File** storage – Stores data in a hierarchical structure of **files** and **folders**
 - Similar to how your computer stores files/folders
- **Filesystem** that is accessible **remotely**



File Storage

- **File** storage – Stores data in a hierarchical structure of **files** and **folders**
 - Similar to how your computer stores files/folders
- **Filesystem** that is accessible **remotely**
 - Creates **subdirectories**
 - Stores and accesses **files** within



File Storage

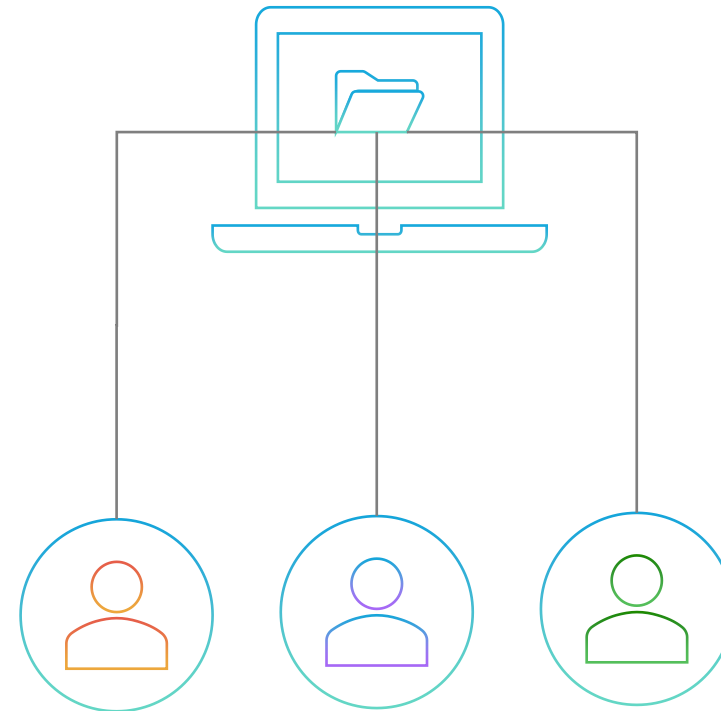
- **File** storage – Stores data in a hierarchical structure of **files** and **folders**
 - Similar to how your computer stores files/folders
- **Filesystem** that is accessible **remotely**
 - Creates **subdirectories**
 - Stores and accesses **files** within
- **File** storage can be **mounted**
- Not **bootable**, so you cannot install an **operating system** on it





File Storage

- File storage is accessible over the network
- Multiple clients can access the same data



File Storage - Summary



File Storage services like EFS store data in a hierarchical structure of files and folders



It is accessible over the network



EFS can be mounted as a file system inside an OS



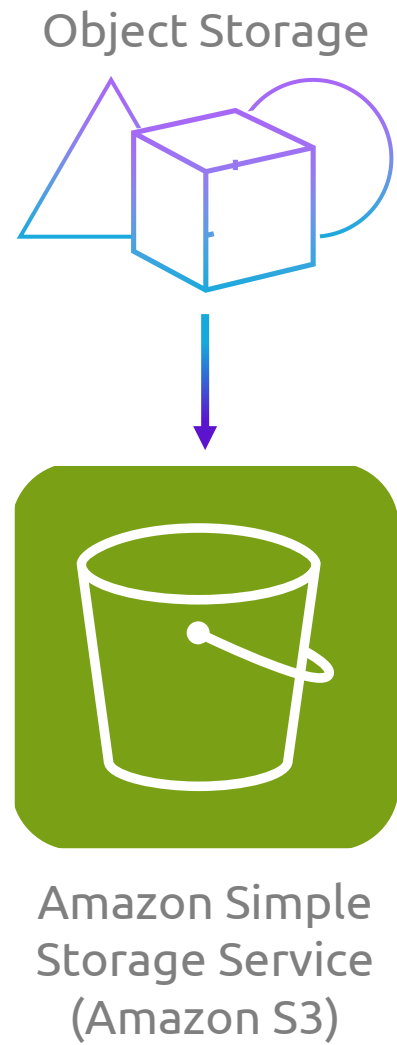
It cannot be used as a boot volume (can't install OS)



Object Storage

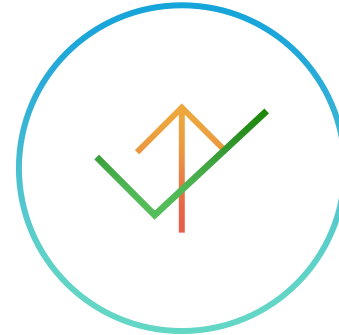


S3 - Object Storage



S3 - Object Storage

- Object Storage – Stores objects
 - Objects are nothing more than files
 - Can store any type of file





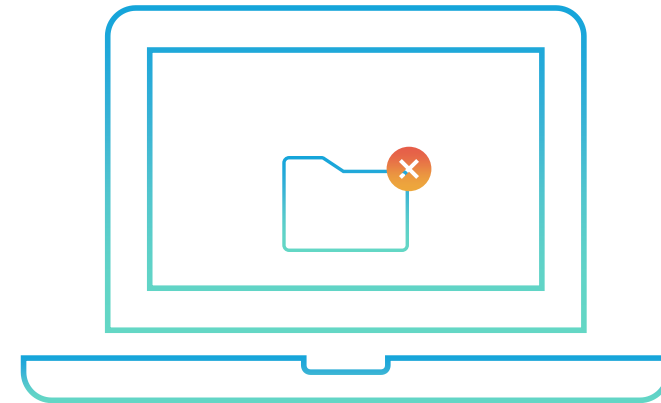
Object Storage

- Object Storage – Stores **objects**
 - **Objects** are nothing more than **files**
 - Can store any type of **file**
- Does not have a **folder** structure
 - Flat **file** structure – Everything is in the same **folder**



Object Storage

- Object Storage – Stores objects
 - Objects are nothing more than files
 - Can store any type of file
- Does not have a folder structure
 - Flat file structure – Everything is in the same folder
- Since there is no folder structure, object storage cannot be mounted or booted



Object Storage

- Object Storage – Stores objects
 - Objects are nothing more than files
 - Can store any type of file
- Does not have a folder structure
 - Flat file structure – Everything is in the same folder
- Since there is no folder structure, object storage cannot be mounted or booted
- Great for storing logs and media files



Storage Classes



Data Access



Resiliency



Cost



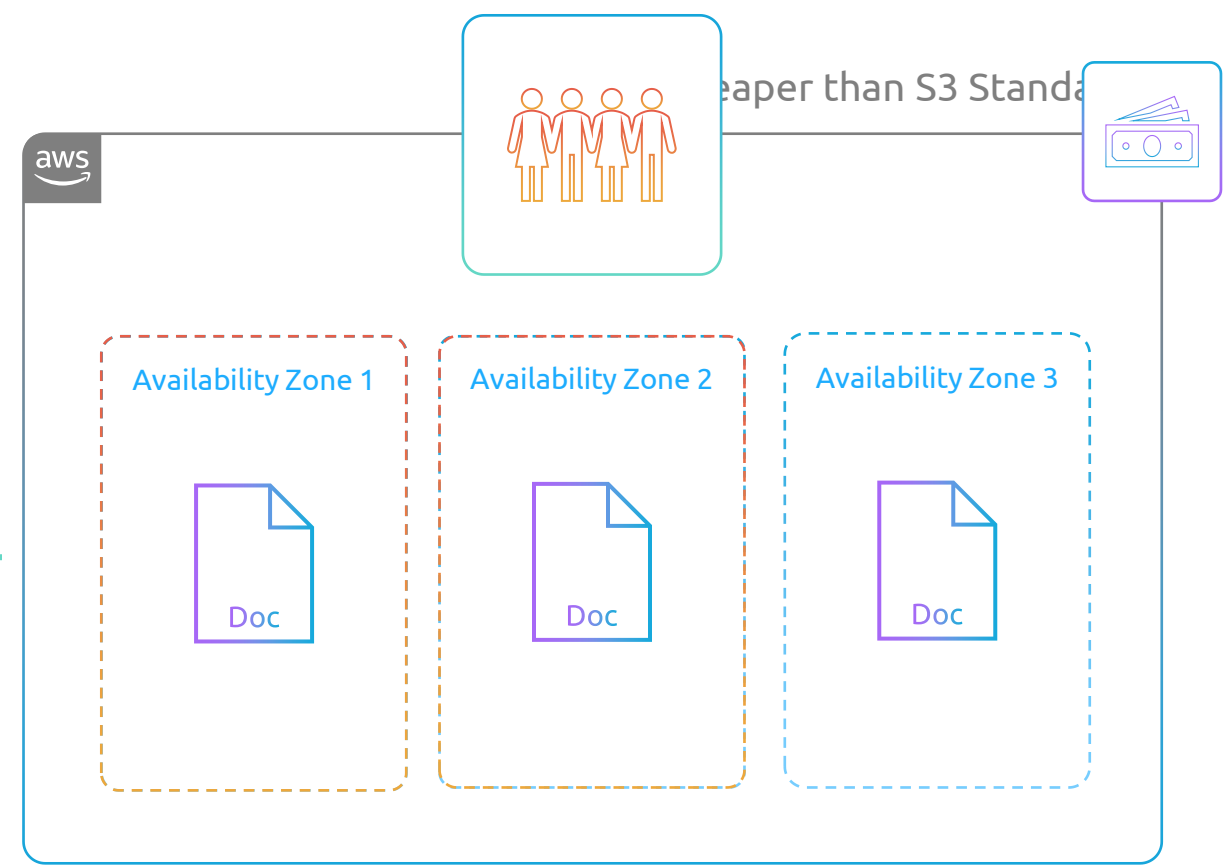
S3 Standard-IA

Has a retrieval fee
Minimum duration charge of 90 days



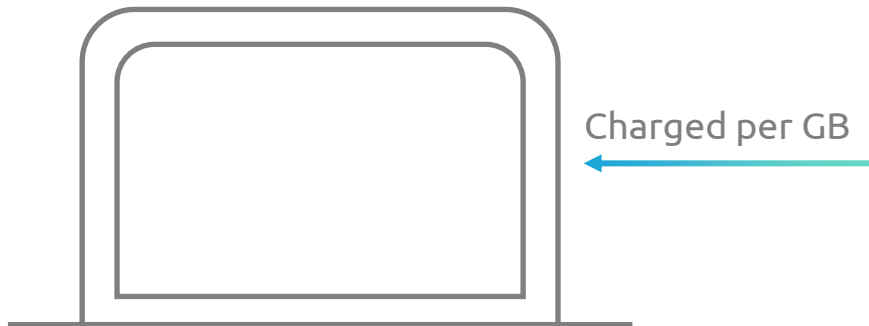
Minimum size charge of 128 KB per object

two 99.999999999% durability
plus AZ failures
Charged per GB



S3 One Zone-IA

Has a retrieval fee
Minimum duration charge of 90 days
Minimum size charge of 128 KB per object



Cheaper than S3 Standard and S3 Standard-IA



Note

Designed for IA data

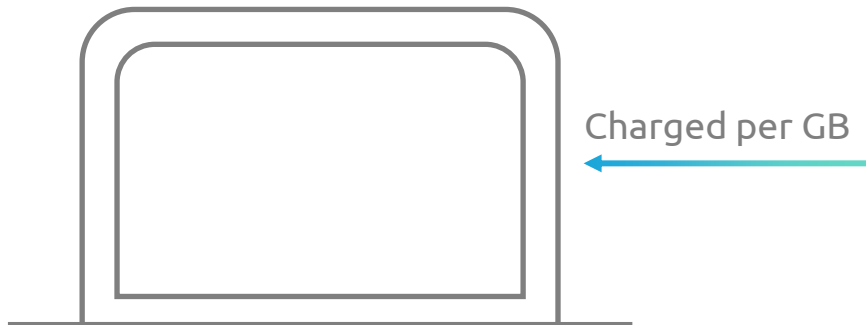
Not required to handle AZ failure

Replication still occurs within AZ



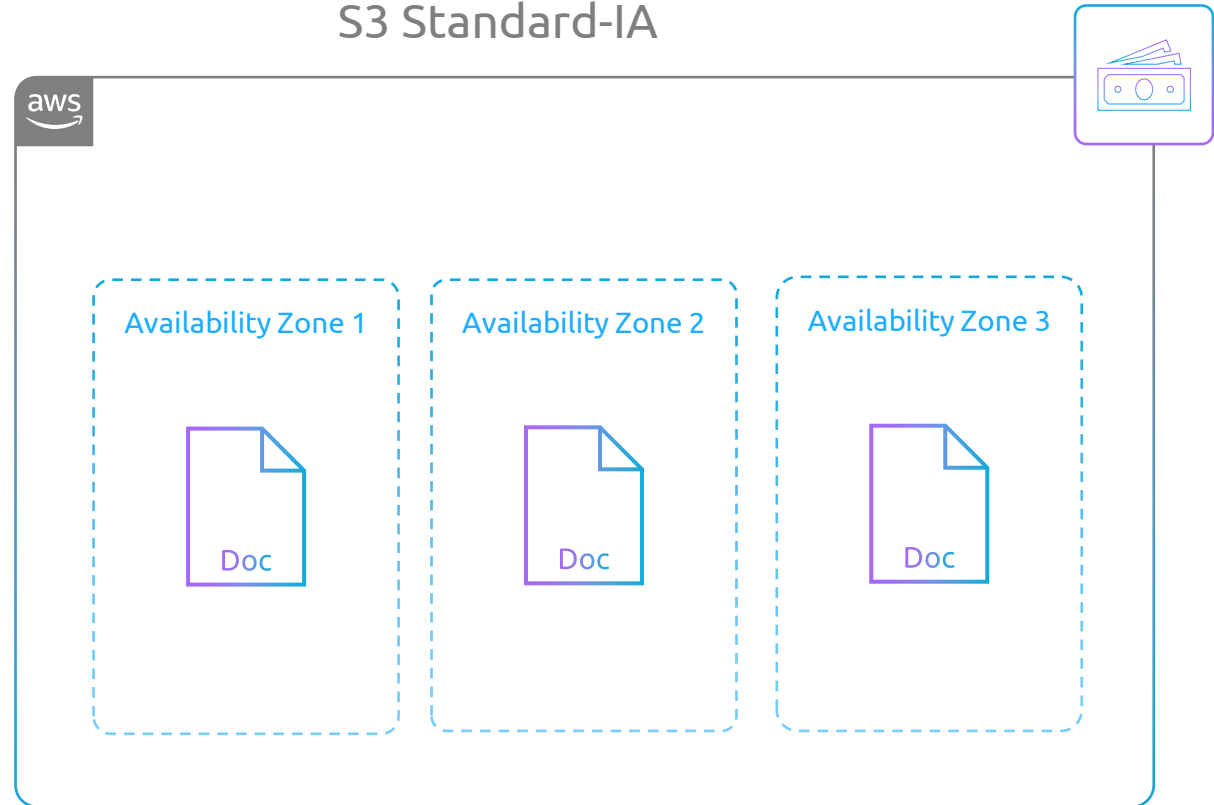
S3 Glacier-Instant

Has a retrieval fee
Minimum duration charge of 90 days
Minimum size charge of 128 KB per object



Note Low-cost option for rarely accessed data
Performance same as that of S3 standard

Cheaper than S3 Standard and S3 Standard-IA



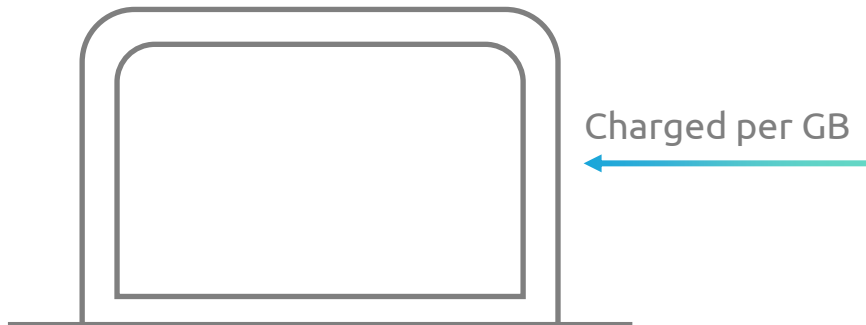
Summary
Very cheap storage

Higher retrieval cost
Longer minimum duration



S3 Glacier-Flexible

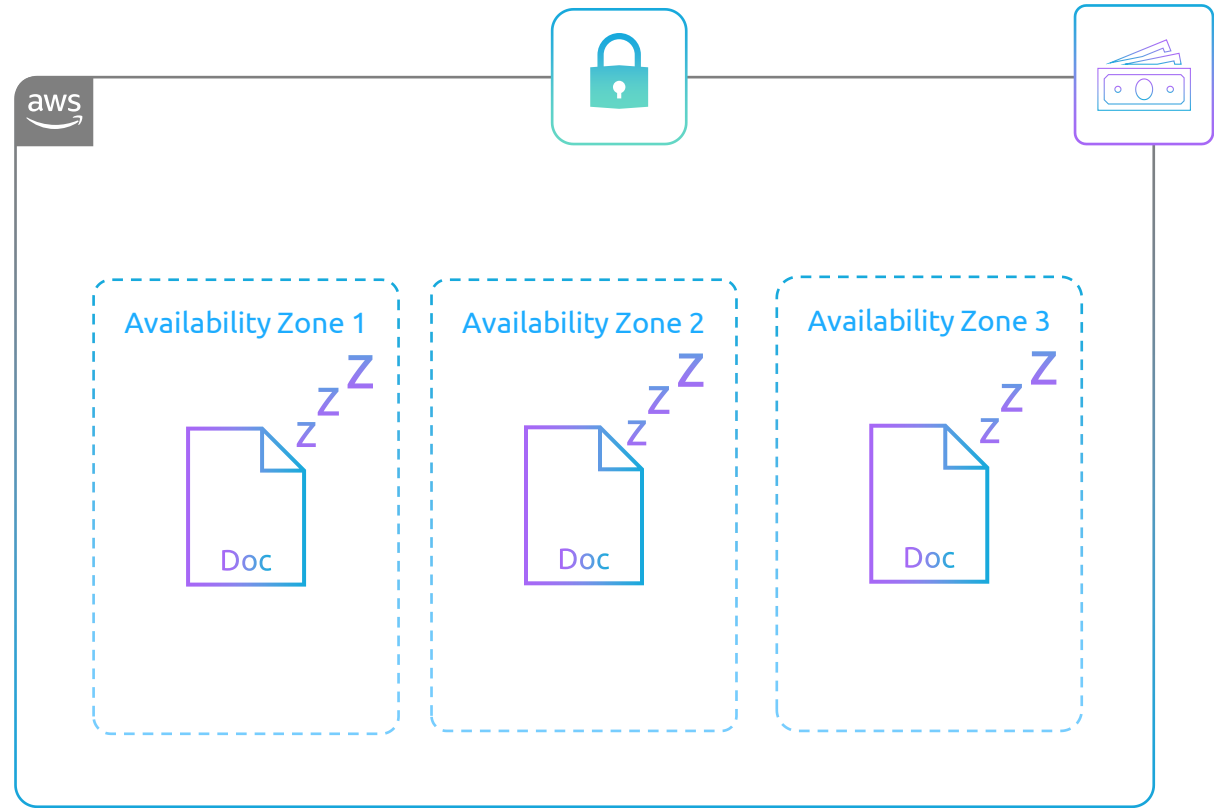
Has a retrieval fee
Minimum duration charge of 90 days
Minimum size charge of 40 KB per object



Options

- Bulk : 5–12 Hours
- Expedited : 1–5 Minutes
- Standard : 3–5 Hours

Cheaper than S3 Standard and so on...

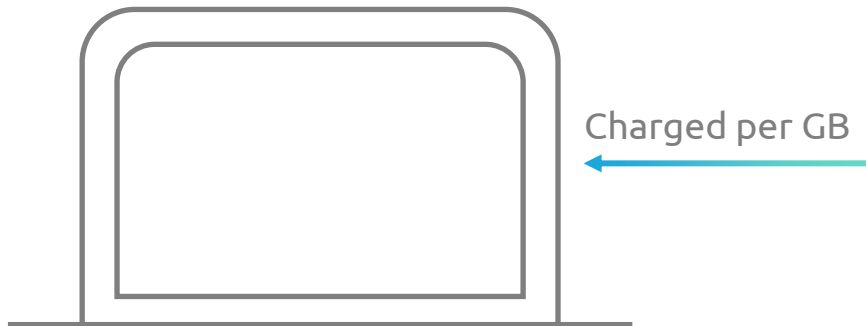


During retrieval, objects are stored in S3 Standard-IA class temporarily



S3 Glacier Deep Archive

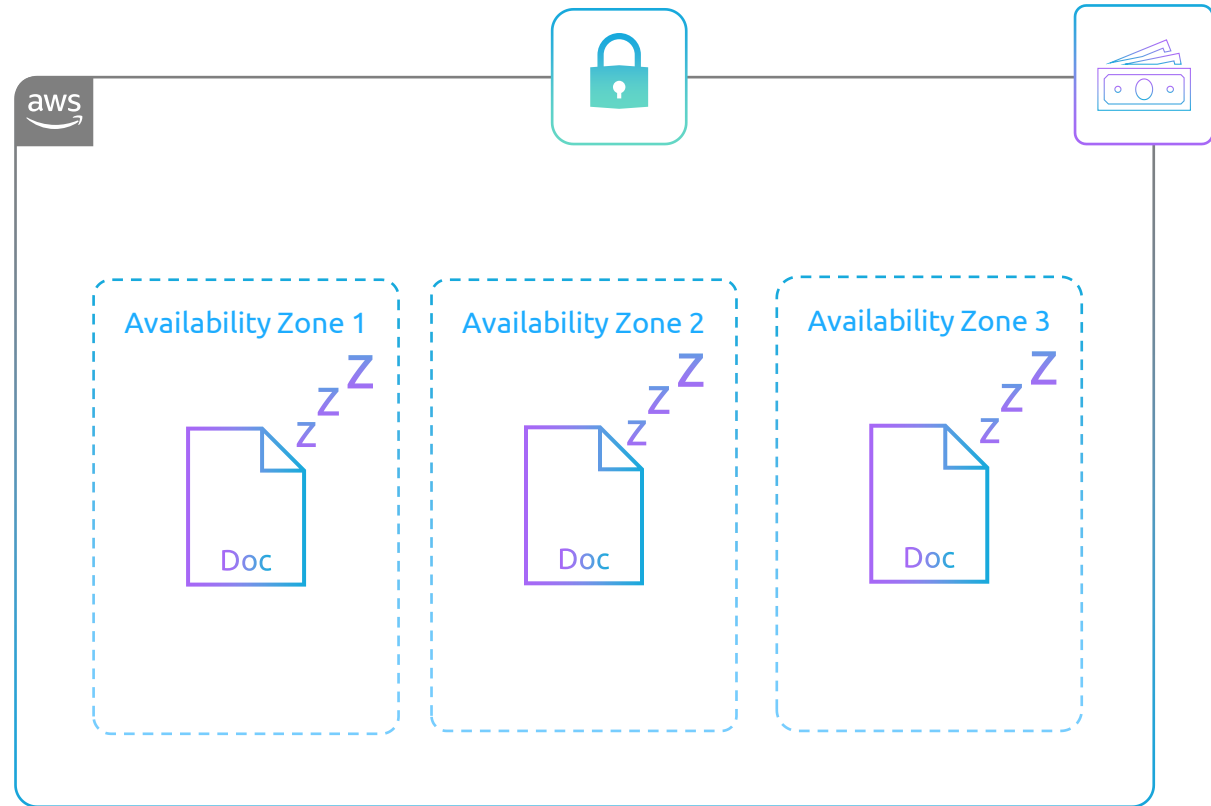
Has a retrieval fee
Minimum duration charge of 90 days
Minimum size charge of 40 KB per object



Options

- Standard : 12 Hours
- Bulk : 48 Hours

The cheapest storage class in S3



S3 Intelligent-Tiering

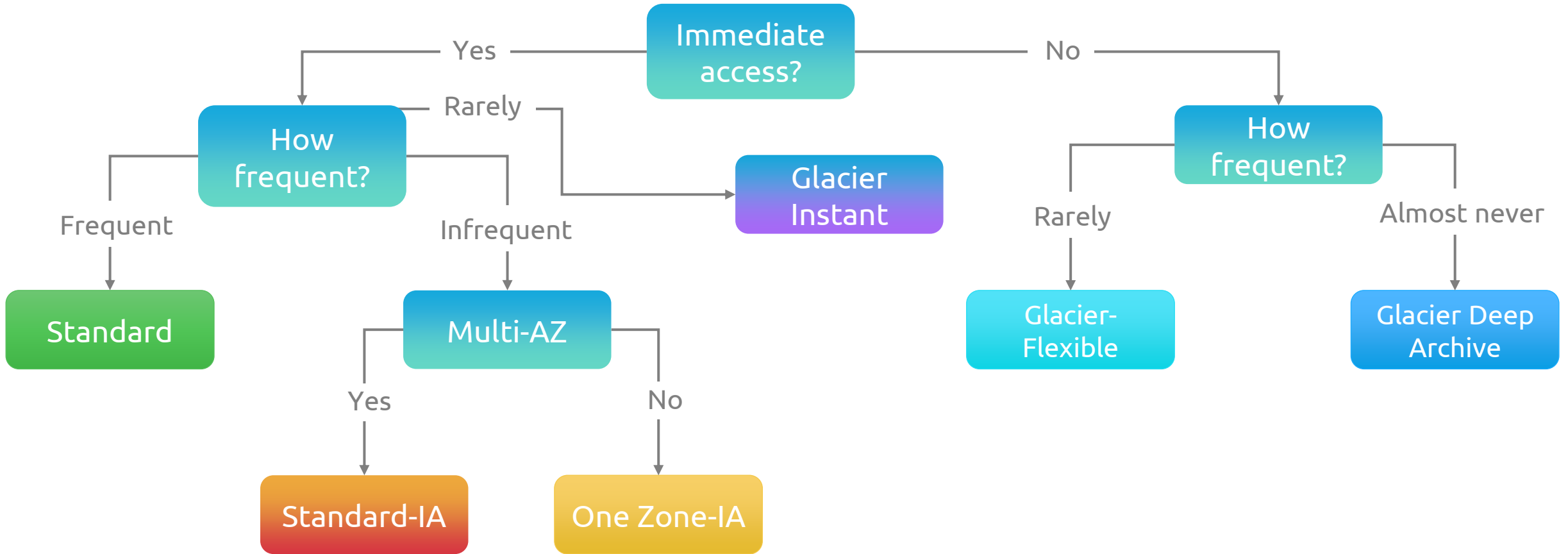
01

Automatically reduces storage costs by intelligently moving data to the most cost-effective access tier

02

Apart from the cost of a storage class an object gets assigned to, all objects will also incur a monitoring/automation cost per **1,000 objects**





Object Storage (S3) - Summary



Objects are just files



Flat file structure (no folders); but they look like folders



Great for storing media files, logs, audit reports or basically any file you want.



API storage so it cannot be and should not be mounted or boot.



Storage classes impact accessibility, resiliency, and cost



Core AWS Services Compute



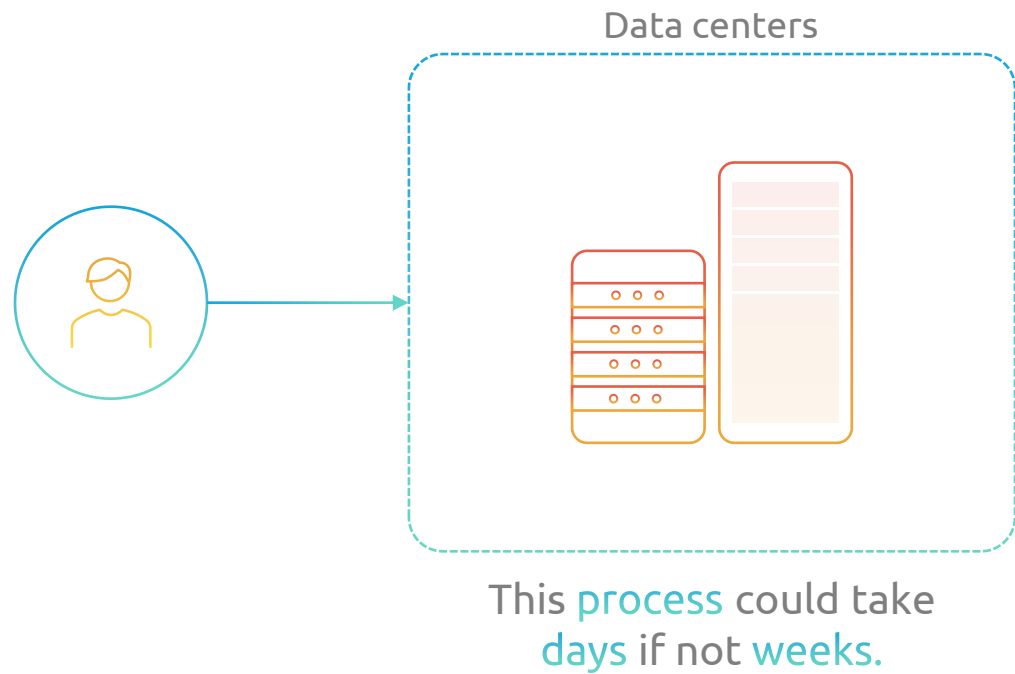
Compute



Servers are needed to run applications.



Deploying an Application



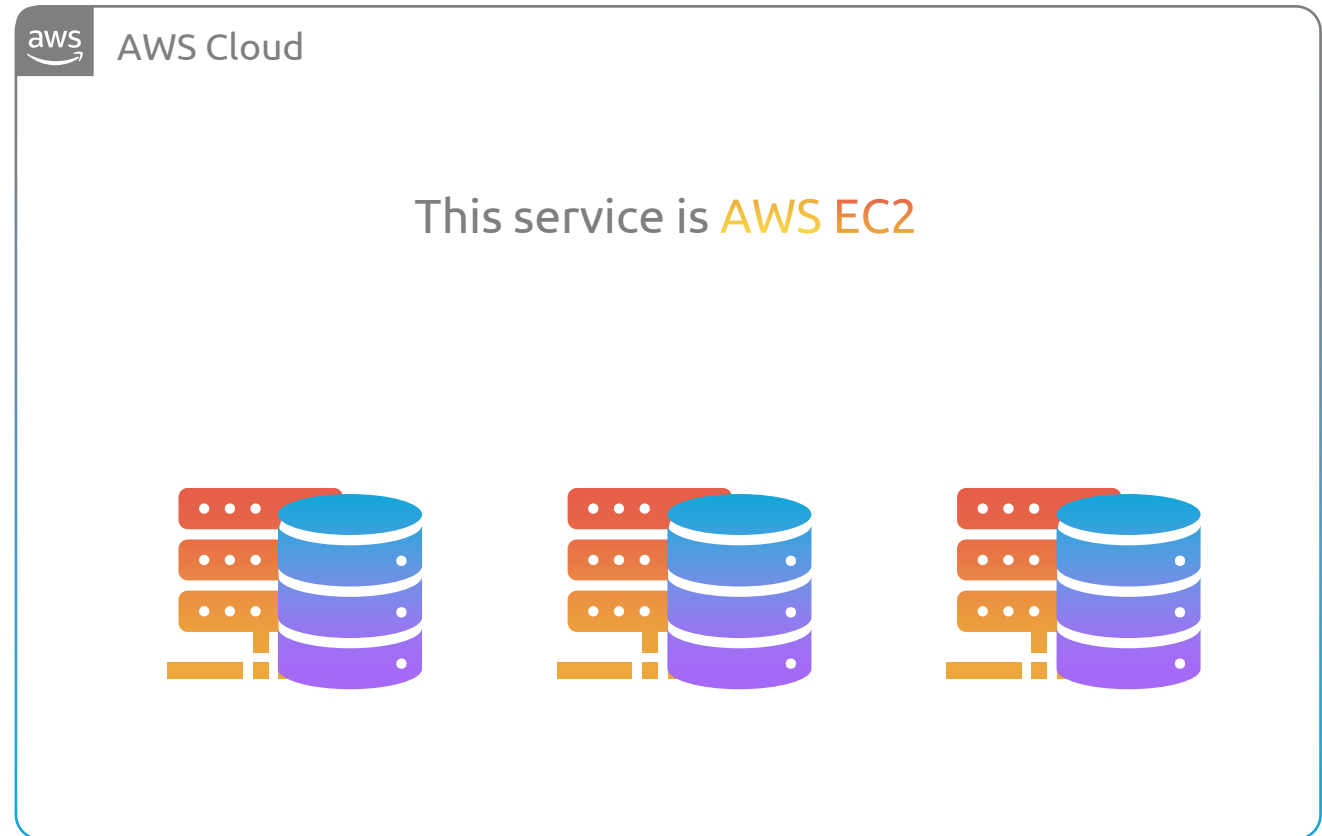
- Purchase a server
- Physically rack a server
- Install an operating system
- Handle hardware failures



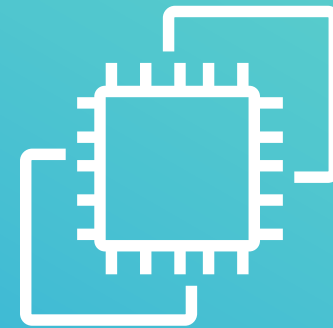
EC2



Provision a **server** with the proper **specs** needed to run our **application** (# of **CPUs**, **memory**, **storage**)



- Service that provides **secure, resizable** compute **capacity** in the **cloud**
- EC2 allows you to **provision** a server on **AWS** within minutes

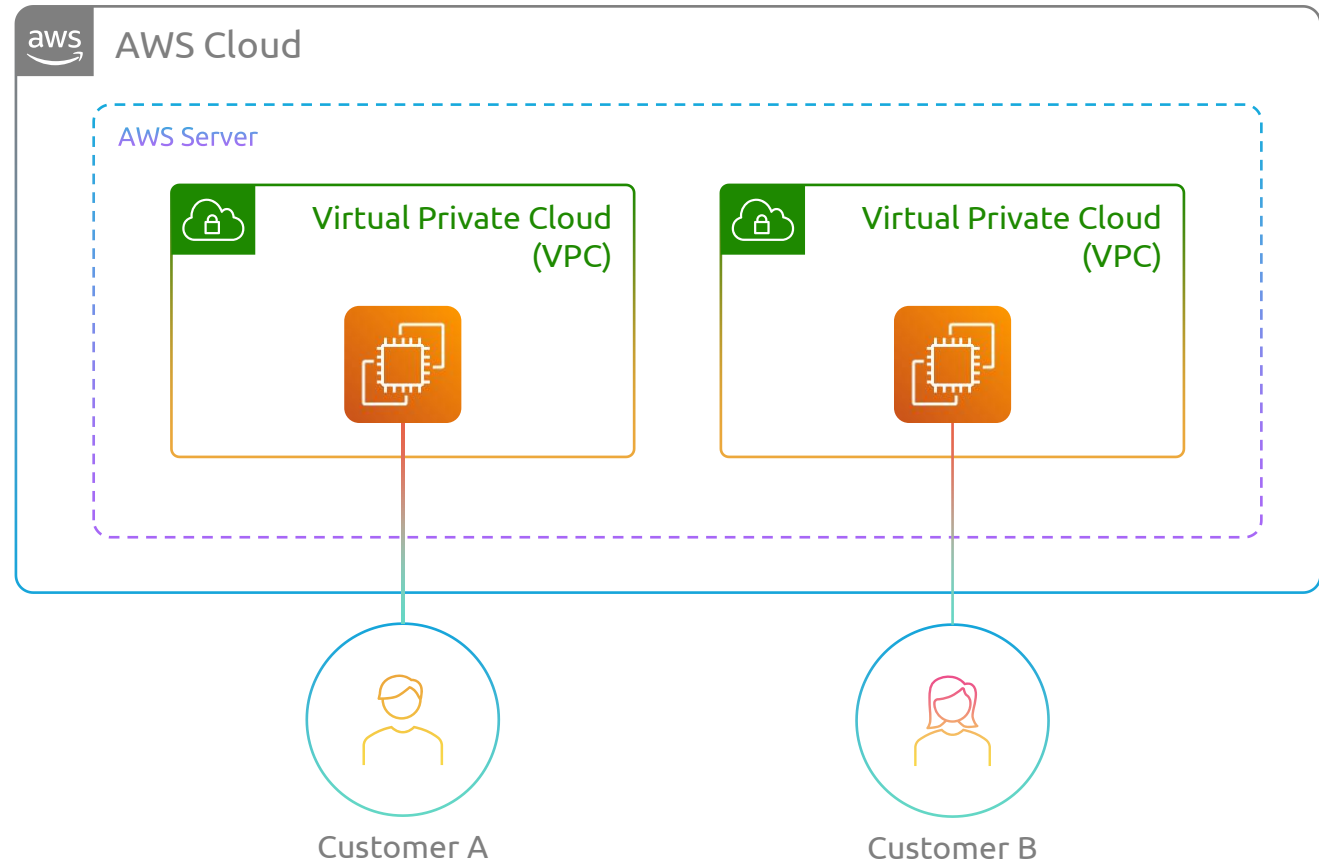


Amazon Elastic Compute
Cloud (Amazon EC2)

EC2

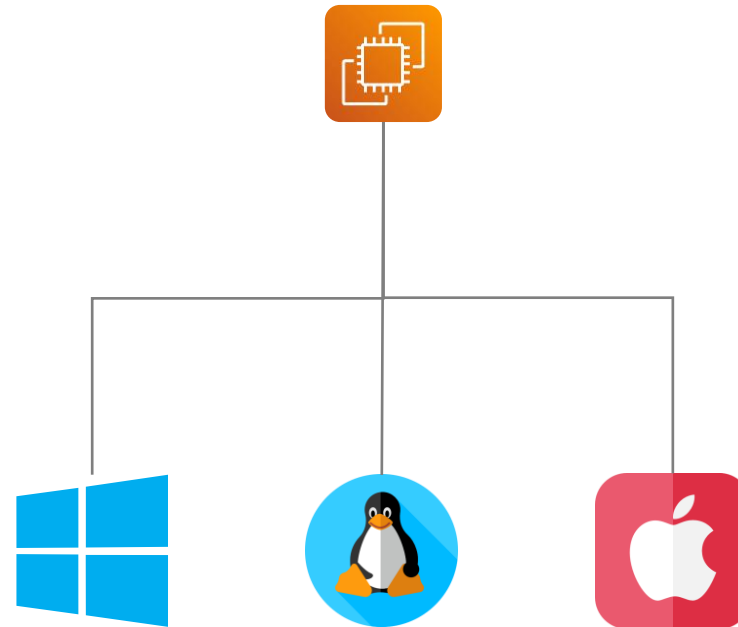
Multiple **EC2** instances can be deployed on a single **server** using **virtual machines**

VPCs are used to logically isolate each customer's infrastructure



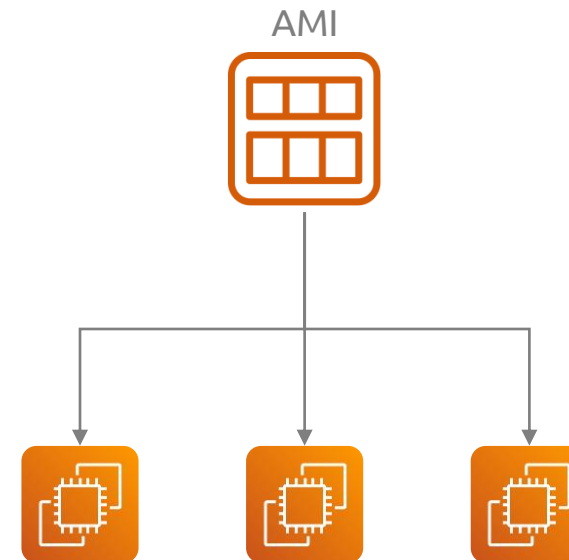
Images

How do we specify what **operating system** the **EC2** instance will use?



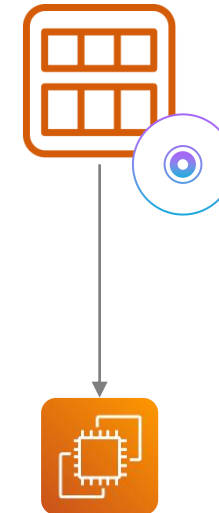
Amazon Machine Image (AMI)

- **AMI** is an image that provides the information required to launch an **EC2** instance
- **AMI** is a blueprint that contains information such as what **operating system** and what software/packages should be installed on an instance
- We can use this blueprint (**AMI**) to deploy identical **EC2** instances



Amazon Machine Image (AMI)

- Think of the **AMI** as an installer disk used for installing a new server
- **AMIs** can be fully customized to:
 - Add application source **code**
 - Add **dependencies**
 - Customize **OS** firewall



Instance Types



Instance Types

- EC2 provides a wide selection of **instance types** optimized to fit different use cases
- **Instance types** have varying combinations of **CPU, memory, storage, and networking capacity**
- Think of **instance types** as different types of server models to select from





Instance Types

General Purpose

- Provide a good **balance** of compute, memory, and networking resources
- Can be used for **diverse** workloads
- Ideal for applications that use resources in **equal** proportions

Compute Optimized

- Optimized for **compute-heavy** applications
- Contain **high-performance** CPUs
- Ideal for **batch processing** workloads, **media transcoding**, **machine learning**, and **gaming servers**





Instance Types

Memory Optimized

- Deliver **fast** performance for **memory-intensive** workloads
- Suited for **databases**

Storage Optimized

- Optimized for workloads that require high, **sequential read** & **write** access to **large data sets** on local storage
- Can deliver tens of thousands of **low-latency**, random I/O operations per second (**IOPS**)





Instance Types

Accelerated Computing

- Utilize hardware **accelerators** to perform expensive **calculations**
- Great for graphics processing and data pattern matching

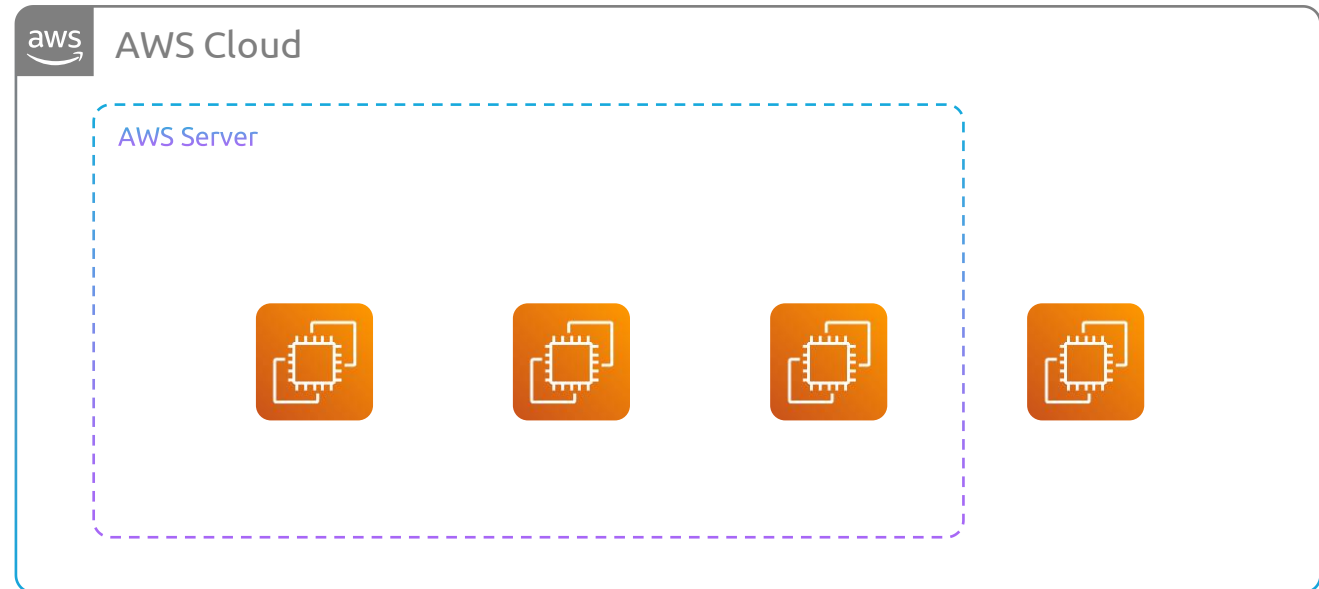


EC2 Pricing Options



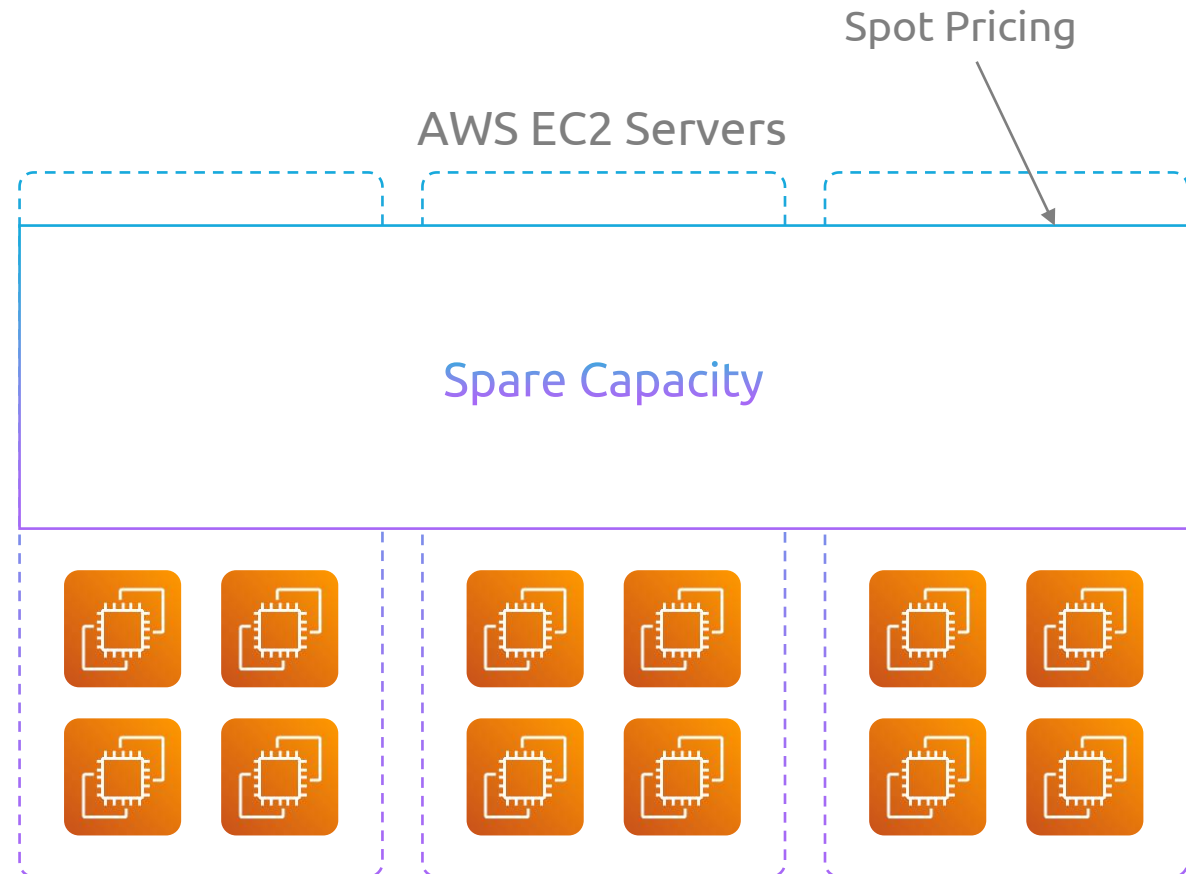
On-Demand Pricing

- Pay for **compute capacity** by the **hour**
- Only **billed** when instance is **running**
 - Still **charged** for **storage** attached to instance
- No **upfront** payment or long-term commitment
- Great for **short-term**, irregular, or **unpredictable** workloads
- Customer **EC2** instances run on **shared hosts**



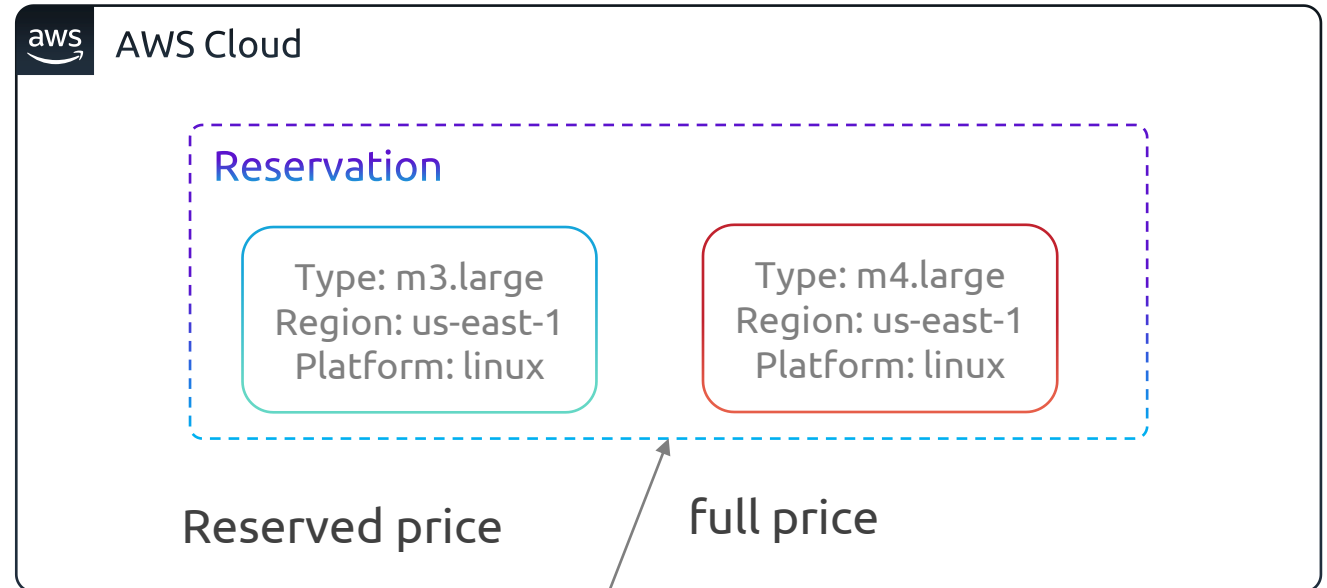
Spot Pricing

- Amazon offers **spare** compute capacity at **discounted** rates
- **Spot** instances are recommended for
 - Applications that have **flexible** start time and end time
 - Applications that need **low** compute **prices**
- Not suited for workloads that cannot tolerate **interruptions**



Reserve Pricing

- **Reserve** instance is a billing **discount** that allows you to save on your **EC2** costs
- Offers discounted **rates** when **reserved** for a certain period (**1-year** or **3-year** contract)
- When purchasing a **reserved** instance, you are not actually **buying** an instance
- You are merely you **committing** to using an on-demand instance for a **long-term** period
- When you deploy an on-demand instance with matching **attributes** as the reservation (**instance type**, **region**, **platform**), it will be charged at the **reserved** price and not the default



type: M3.large
Region: us-east-1
Platform: linux



Reserved Pricing

Standard



Two offering Classes

- Standard class
 - Can't be **exchanged** (certain attributes can be modified)
 - Stuck with **one** instance family
 - Can be sold in the Reserved Instance **Marketplace**
 - Can be bought in the Reserved Instance **Marketplace**
 - More **savings** with **Standard** class



Reserved Pricing

Convertible



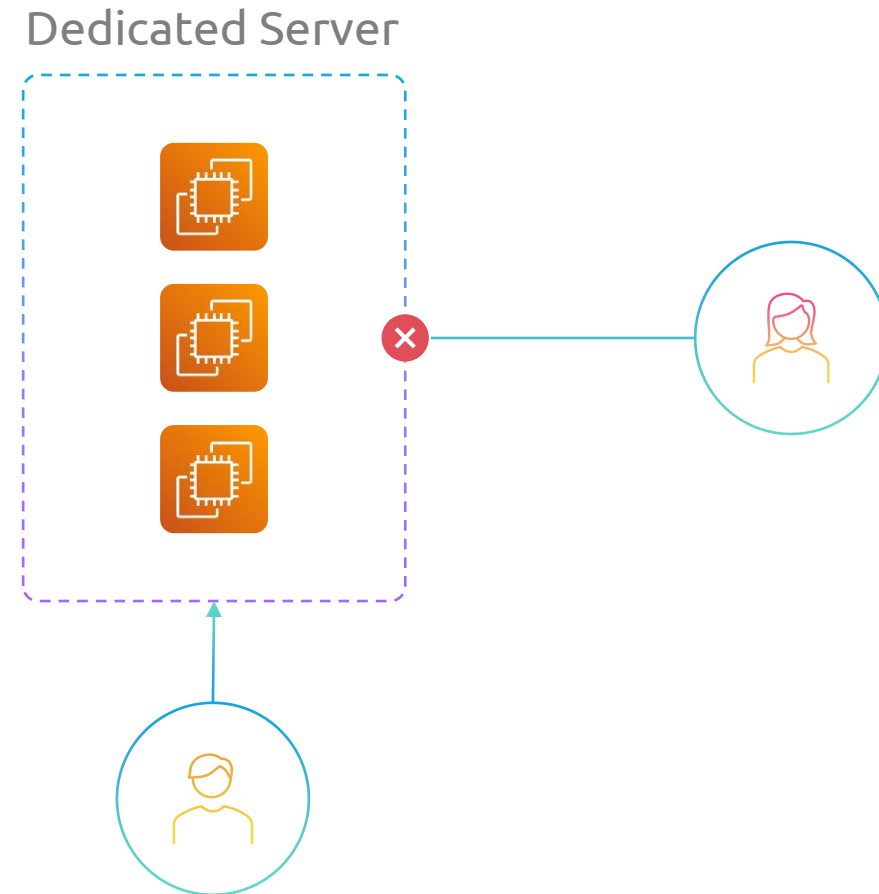
- **Convertible Class**

- Can be **exchanged** for another **Convertible** Reserved instance with a different configuration
 - Different instance family, operating system
- Can't be sold on Reserved Instance **Marketplace**
- Can't be bought in the Reserved Instance **Marketplace**



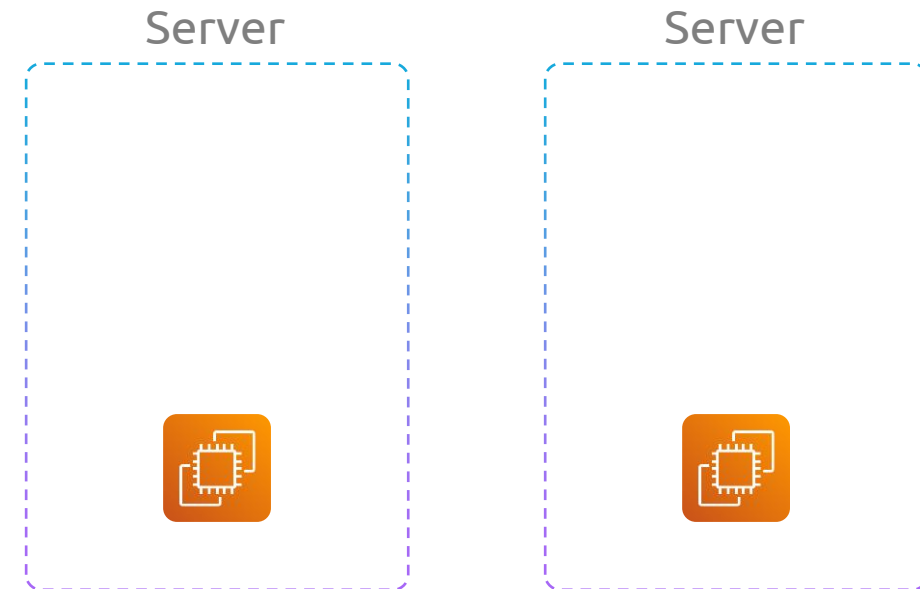
Dedicated Hosts

- A **physical** EC2 server is dedicated for your use
- Can help **reduce costs** by allowing you to use your existing server-bound **software licenses** (Windows server, SQL server, or SUSE Linux Enterprise)
- Can be purchased **either On-Demand** (hourly) or as a **reservation**
- Pay for the **host**; there is no per **instance charge**



Dedicated Instance

- Only **your** instances run on a **physical** server
- If an instance is **stopped** and then **started**, it could move to a different **host**
- With **dedicated** Host, you always have the **same** physical machine



Elastic Compute Cloud (EC2) - Summary



EC2 allows you to provision a server in AWS within minutes



AMIs are templates for deploying EC2 instances



AWS has a variety of instance types to support all your computing needs(Memory, Compute, Storage optimized)



AWS supports a wide variety of Operating Systems from RHEL, SUSE, Ubuntu, Amazon Linux and Windows



AWS also offers a variety of processors from ARM to AMD to Intel.



AWS marketplace has thousands of AMIs that offer a variety of prebuilt services(NGINX, Palo Alto Firewall, MongoDB)



Elastic Compute Cloud (EC2) - Summary



OnDemand Pricing – Pay for what you use, only billed when instance is running



Spot Pricing – discounted rates when Amazon has space capacity



Workloads on Spot Pricing need to tolerate interruptions



Reserved Pricing – Discounted rates when reserved for long periods of time(1-3 years)



Dedicated Host – Reserves an entire host(physical server) for you.



Dedicated instance – Only your instances run on a server, but that server can change if instances are stopped/started

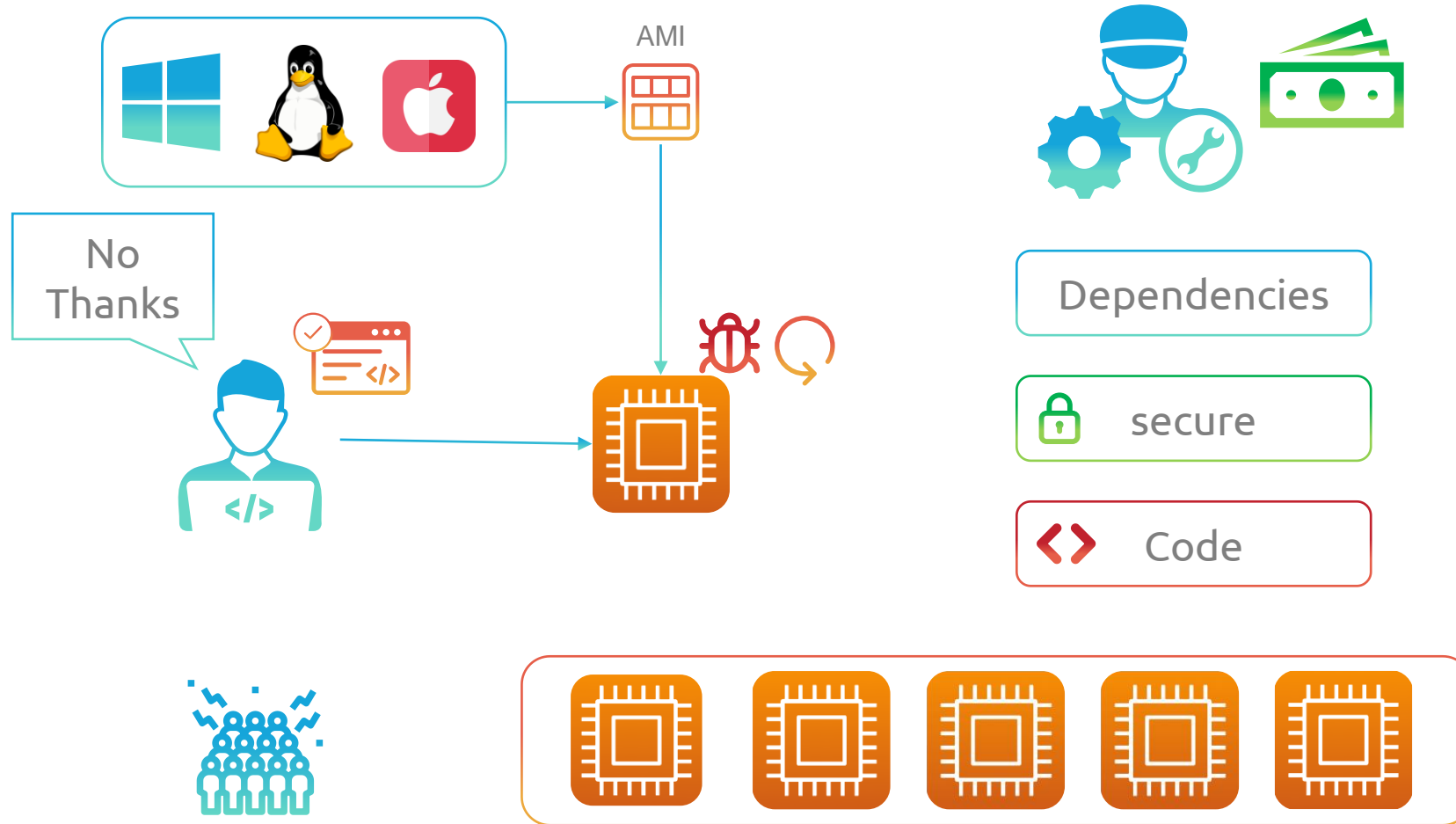




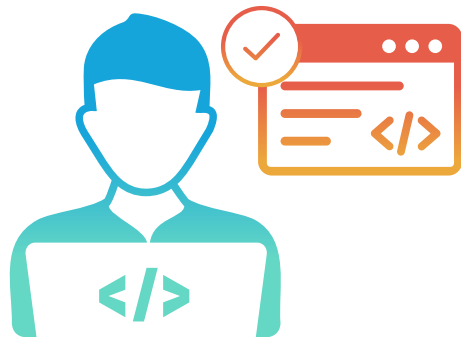
AWS Lambda



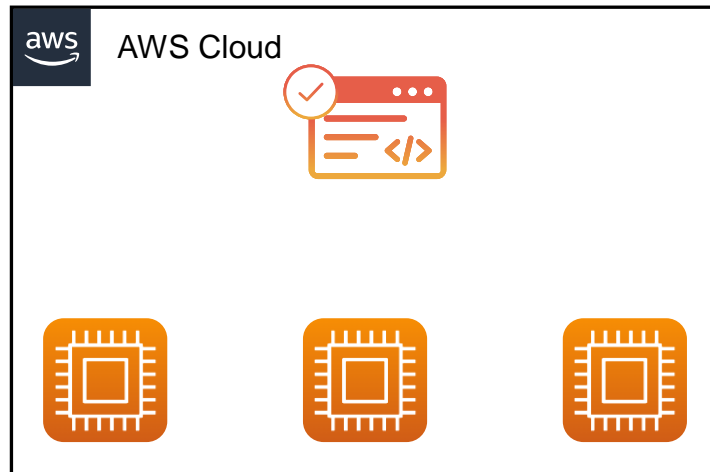
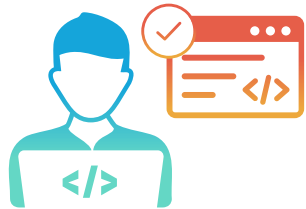
Deploying Application On AWS



Deploying applications on AWS



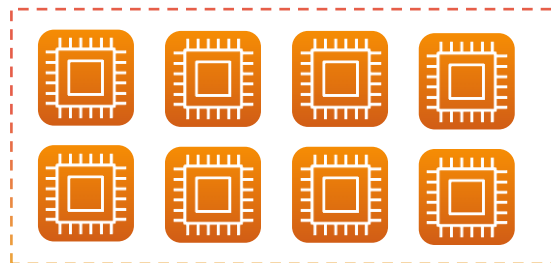
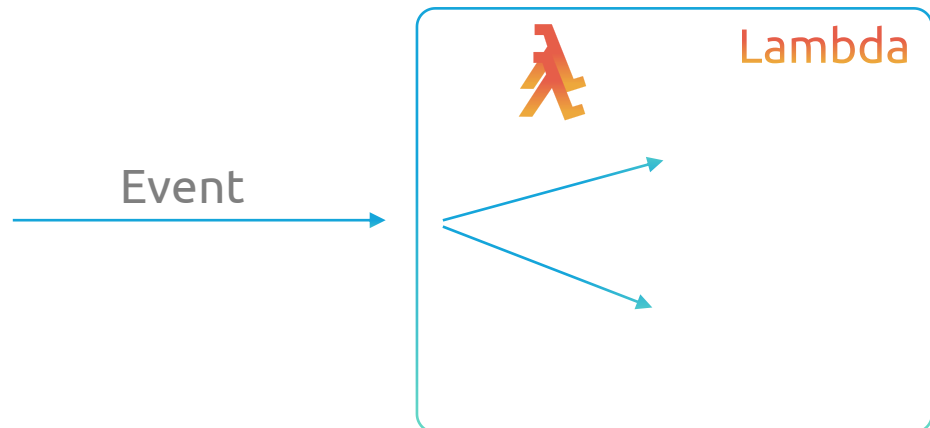
What is lambda



- AWS **Lambda** is a compute service that lets you run code without having to provision or manage servers
- Lambda is AWS' **serverless** offering
 - AWS manages the server **maintenance**, **scaling**, **capacity provisioning**, and **logging**



Serverless

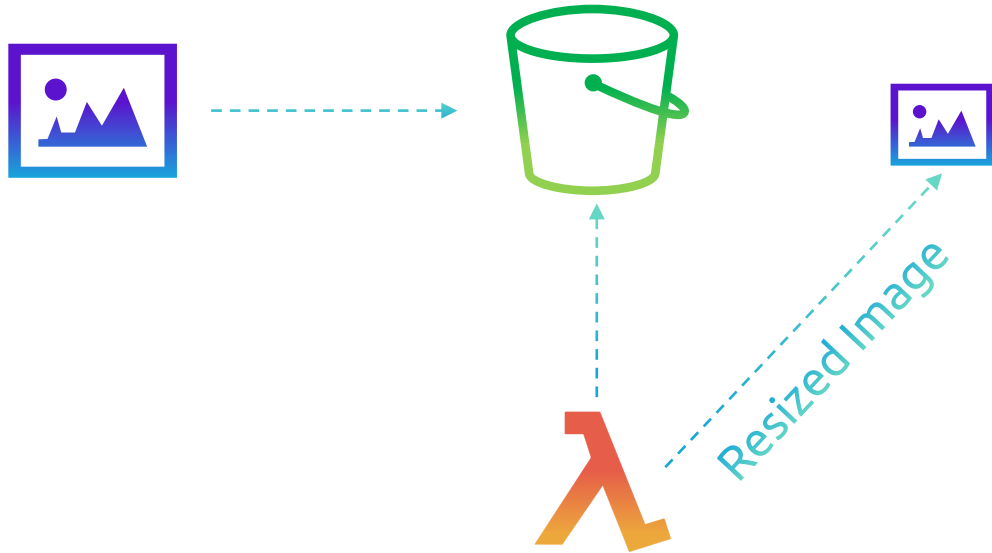


Lambda Service
Reserve Pool

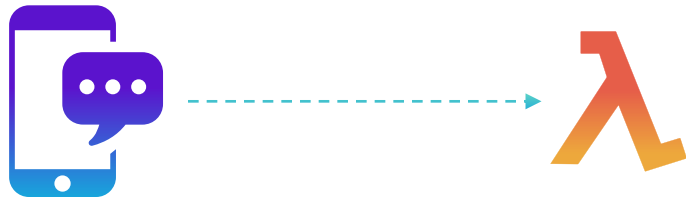
- Does **serverless** mean there's no servers?
- No, to run an **application** there will always need to be a server



Lambda Use cases



- File Processing
- Stream Processing
- Web application
- Mobile/Web Backend



Lambda

Lambda Function

```
Terminal
exports.handler = async function
(event, context) {
  console.log("Lambda function ran");
  return;
};
```

Trigger

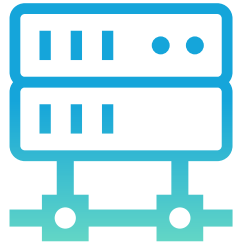
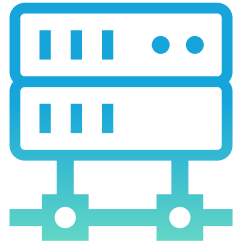
- ✓ Upload file S3
- ✓ Request to API Gateway
- ✓ DynamoDB update



Event



Benefits of Lambda



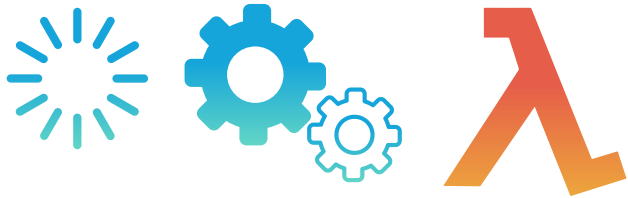
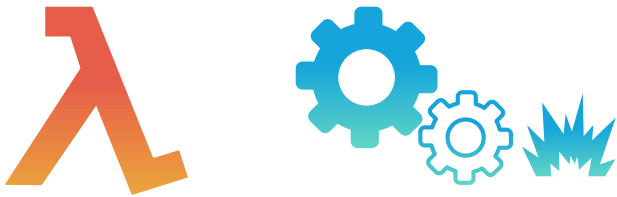
Working on something



- No **servers** to manage
 - No need for an **infra** team
 - No **patching/upgrading**
 - Faster **development**
- Auto scale to handle traffic **spikes**
- Pay for what you **use**
 - Pay per **invocation**
 - No extra **cost** during low traffic



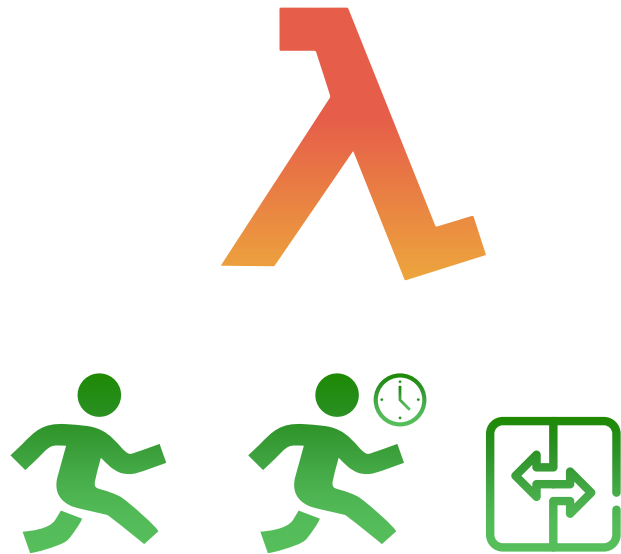
Downsides of Lambda



- No **local state**
 - Will need a separate **database** to store data that needs to **persist**
- **Limited** execution duration
 - Function can run at most for **15 minutes**
 - Not good for long running **tasks**
- Cold starts
 - Cold starts occur due to the time it takes to initialize and load the **function**
 - **SnapStart** and **provisioned concurrency** helps mitigate cold starts








Lambda Pricing



- Number of times function ran
- How long did it run for
- How much memory/cpu did it require



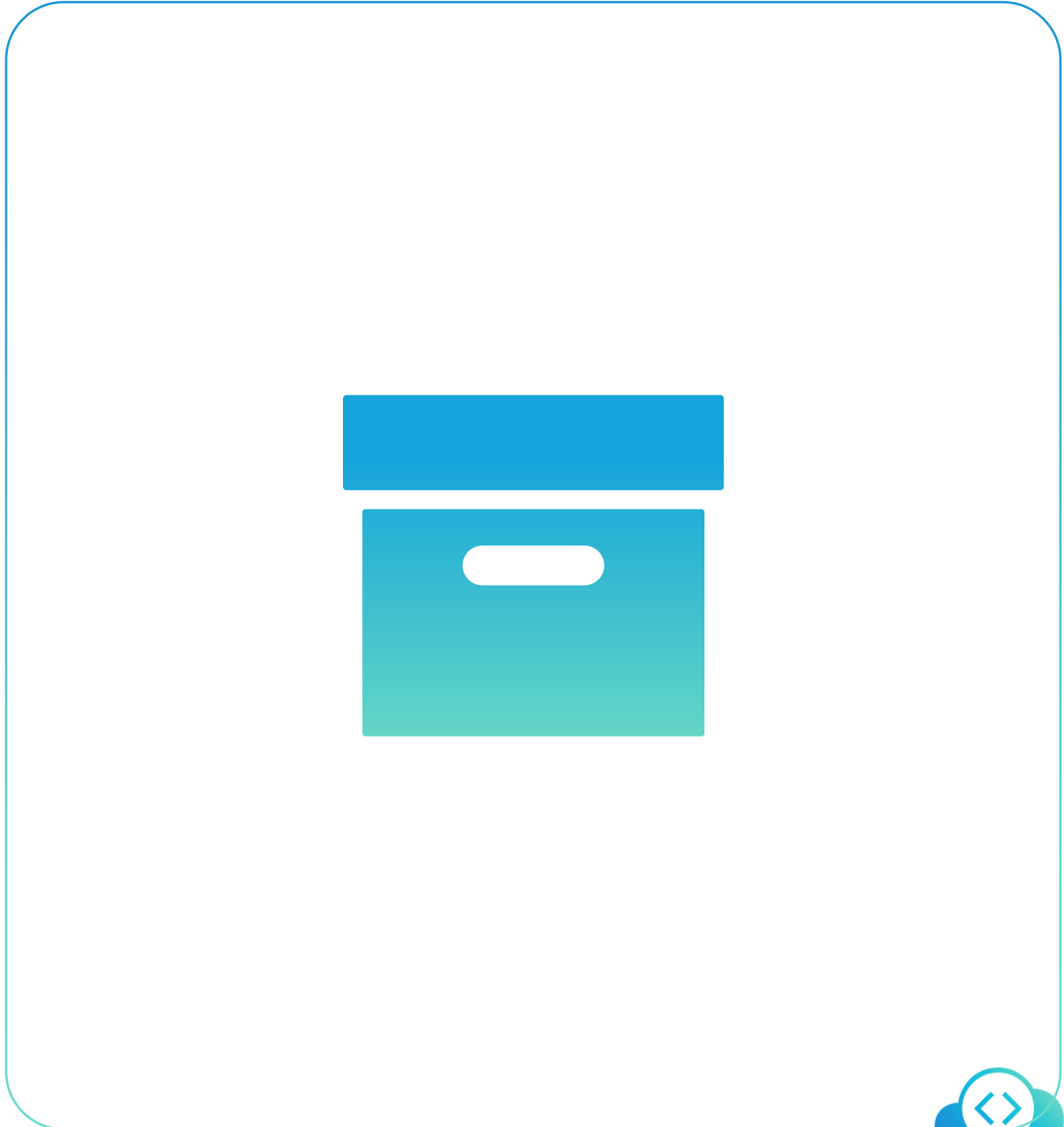
Lambda - Summary

-  AWS Lambda is a compute service that lets you run code without having to provision or manage servers
-  AWS manages the server maintenance, scaling, capacity provisioning, and logging
-  Use cases include file processing, mobile and web backend
-  Autoscales to handle spike in traffic
-  Pay per invocation, only pay for what you use





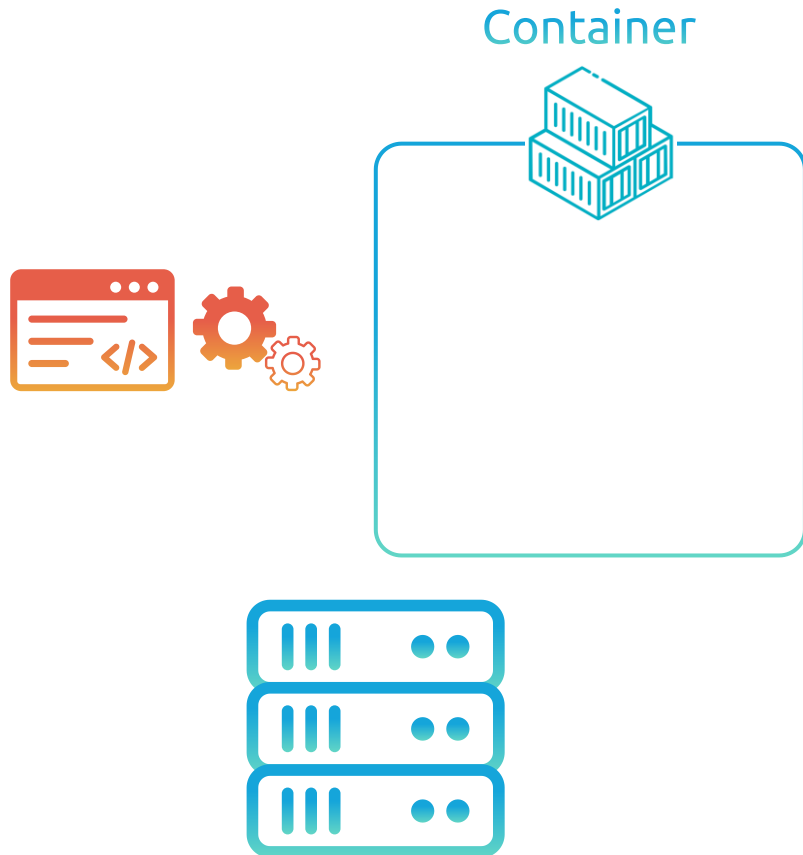
Containers



Containers with ECS & EKS



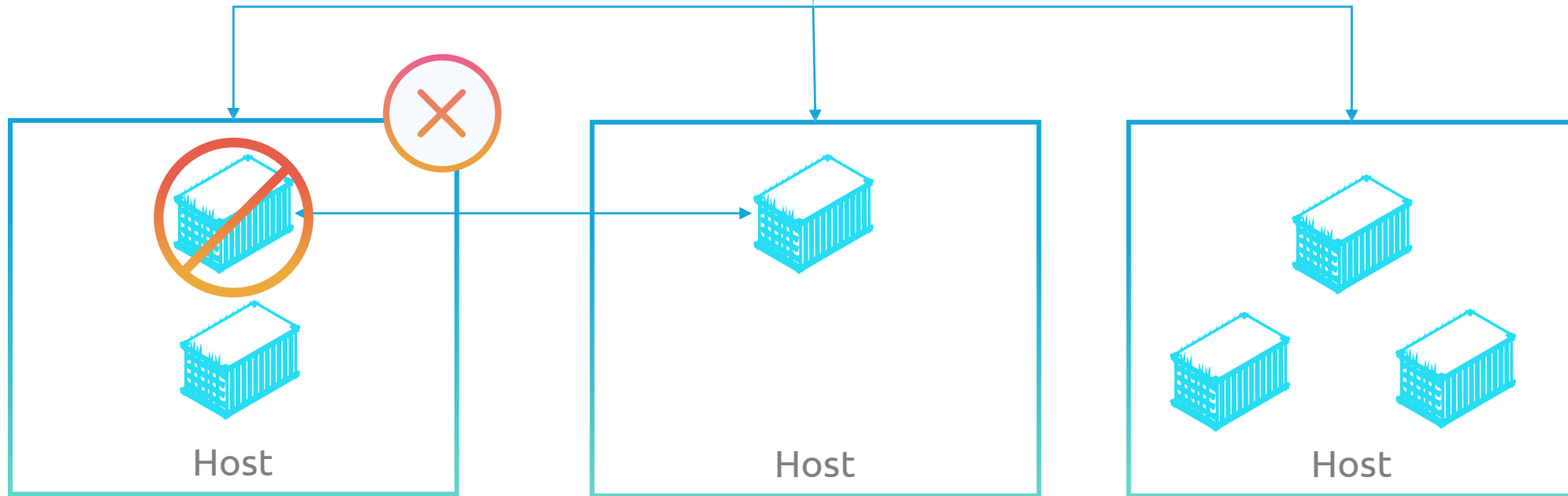
What are containers



- **Containers** are a tool that allows you to **package** an **application** and all of the necessary files, libraries, and dependencies the **application** needs to run
- **Containers** can be deployed on a **machine** and since they have everything the **application** needs to run there is nothing else that needs to be done
- Think of **containers** as small lightweight versions of **virtual machines**



Container Challenges



Container Orchestrators



Kubernetes



Apache Mesos

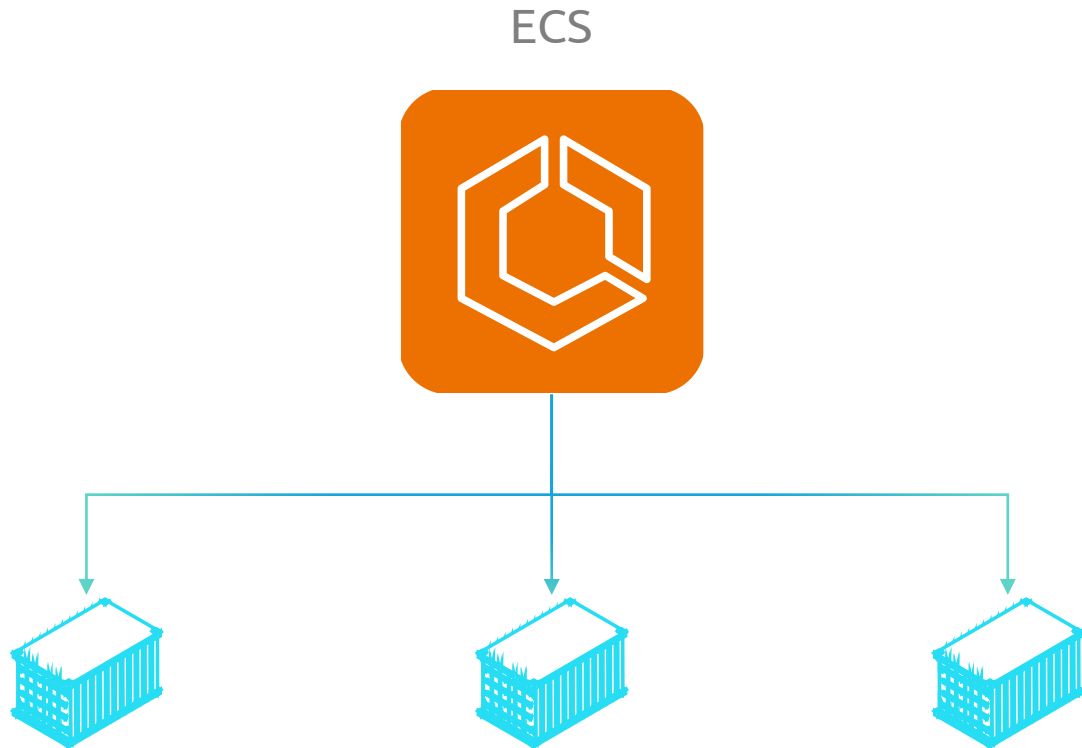


ECS

- **Container orchestrators** are the brains of a containerized environments
- **Responsibilities** include:
 - Deploying **containers** across all available **servers**
 - **Load-balancing** requests to containers
 - Providing **container** to **container** connectivity
 - **Restarting** failed containers
 - Moving containers when hosts **fail**



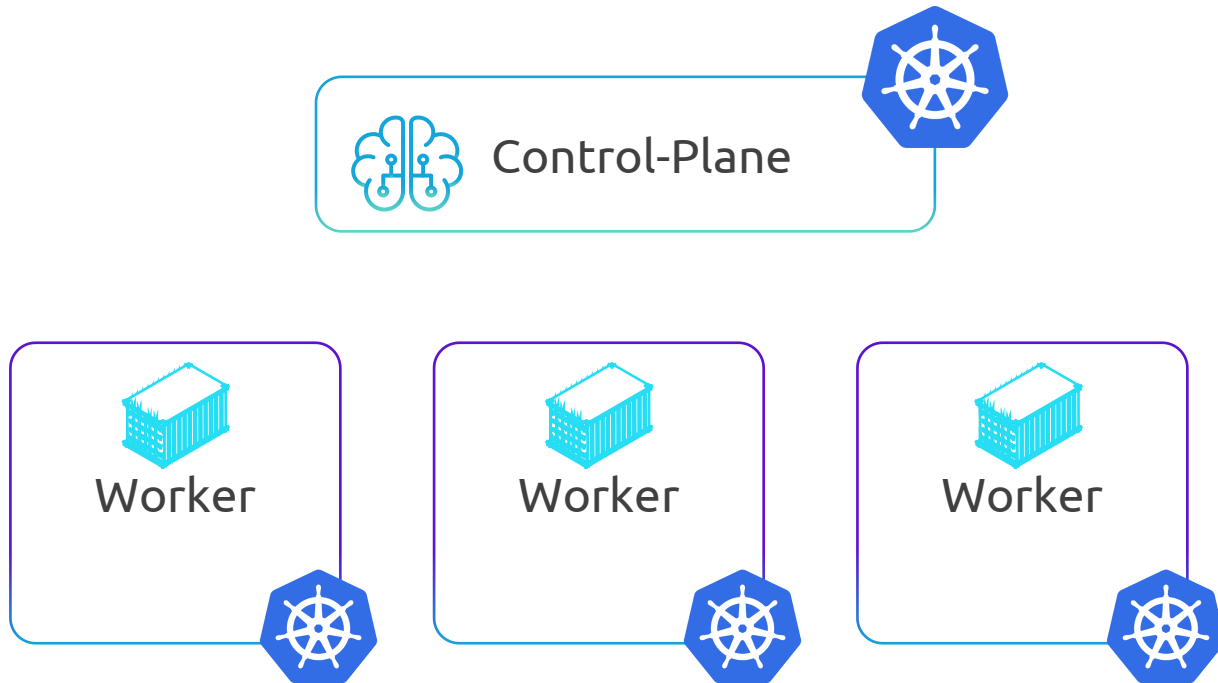
Elastic Container Service (ECS)



- Fully managed **container** orchestration service that helps **manage**, and **scale** containerized **applications**
 - **ECS** is managed by **AWS**
 - **Containers** run on **EC2 instances** or **Fargate**
- **ECS** is proprietary software (only available on **AWS**)
 - Migrating to a different **cloud provider** will be more difficult



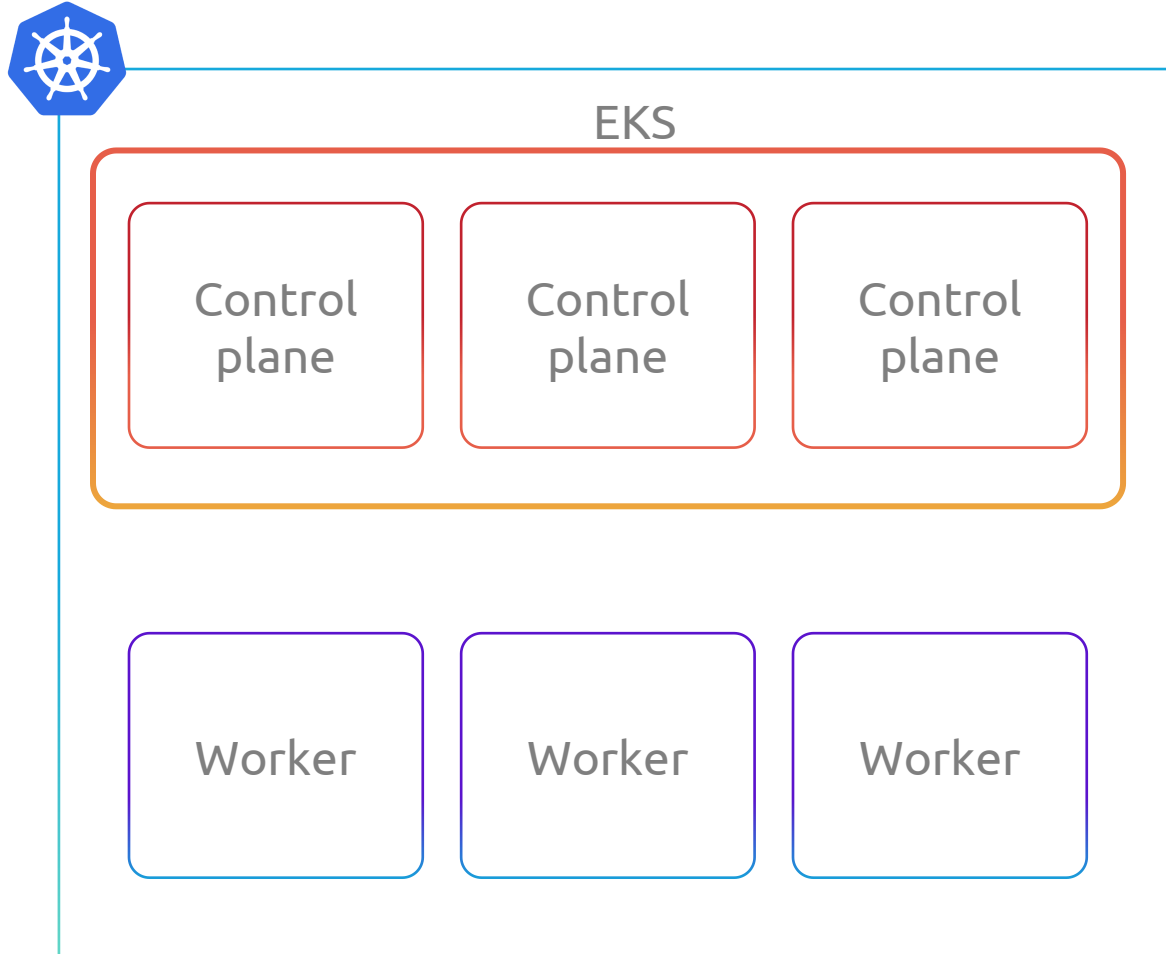
Kubernetes



- **Kubernetes** is an open-source **container** orchestrator
- **Kubernetes** cluster has two types of **nodes**
 - **Control-plane** nodes – managers of the cluster
 - Watch over **cluster** and make sure **cluster** is kept in a working state
 - **Worker** nodes – responsible for actually running the **containerized** workloads



Elastic Kubernetes Service (EKS)



- **Users** are responsible for managing both the **control-plane** and **worker** nodes
- Managing **control-plane/master** nodes is difficult
 - Running & scaling **control-plane**
 - Properly securing **control-plane**
- AWS Elastic Kubernetes Service (**EKS**) is a managed Kubernetes service
 - **EKS** manages the **control-plane** for you
 - **Users** are still responsible for managing **worker** nodes
 - Unless they decide to use **Fargate** (AWS will manage nodes)



Benefits of EKS



- Runs & scales **control-plane** across multiple **Availability Zones**
- Scales **control-plane** instances based on load
- Can integrate with other **AWS** services



AWS Identity and Access Management (IAM)



Elastic Load Balancing



Amazon Elastic Container Registry (Amazon ECR)

- **IAM** for authentication
- Elastic load Balancer
- **ECR** for container images



ECS vs EKS

- **ECS** is proprietary to AWS so migrating to another cloud provider can be difficult
- **EKS** is Kubernetes which is open source and can be run on any platform
 - Keep in mind the more aws services you configure your cluster to interact with the harder it will be to move to another provider
- **ECS** has a simpler architecture
 - Simpler API
 - Easier to ramp up new team members
- **Kubernetes** is very complex and has a steep learning curve
 - With EKS you'll have to learn Kubernetes as well as EKS specific features
- **Kubernetes** has a larger community
 - More support online
 - More tooling like Helm/Kustomize/ArgoCD
- **ECS** Pricing – no cost for control-plane, only pay for infrastructure running applications (EC2/Fargate, EBS)
- **EKS** Pricing – more expensive, you have to pay for control-plane and worker-nodes/Fargate



Containers – Summary

- ✓ Containers are a tool that allows you to package an application and all of the necessary files, libraries, and dependencies the application needs to run
- ✓ Container orchestrators deploy, manage, and scale containerized applications
- ✓ ECS is simple managed container orchestrator provided by AWS
- ✓ Kubernetes is an open source container orchestrator
- ✓ EKS is a managed Kubernetes service – where AWS manages the control plane for you





KodeKloud