

SECTION 1

Welcome

SOA-C02 Exam Guide





Who should take this exam?

- Minimum of 1 year of hands-on experience with AWS technology
- Experience in deploying, managing, and operating workloads on AWS
- Understanding of the AWS Well-Architected Framework
- Hands-on experience with the AWS Management Console and the AWS CLI
- Understanding of AWS networking and security services
- Hands-on experience in implementing security controls and compliance requirements



Format of the Exam

- 180 minutes
- 65 ‘scoring opportunities’ from:
 - Multiple choice / multiple response questions
 - Exam labs – **new** to the SOA-C02
- My experience:
 - 50 questions
 - 3 exam labs
- Delivery through Pearson VUE testing center or online proctored exam
- You get your results within 5 business days (just 1 day for me)



Exam Labs

The image shows two screenshots illustrating the AWS Management Console and a challenge lab interface.

AWS Management Console (Left):

- The URL is us-east-2.console.aws.amazon.com/console/home?region=us-east-2#.
- The search bar at the top contains the placeholder "Search for services, features, marketplace products, and docs".
- A modal window titled "Check out the new unified search" is displayed, explaining that users can search for services, features, documentation topics, and AWS Marketplace products.
- The main content area includes sections like "Build a solution" (Launch a virtual machine, Build a web app, Build using virtual servers), "Getting Started with AWS" (Learn the fundamentals, Pick a learning path, Dive deeper), and "Stay connected to your AWS resources on-the-go" (AWS Console Mobile App).
- At the bottom, there are links for "Feedback", "English (US)", and copyright information: "© 2008 - 2021, Amazon Web Services, Inc. or its affiliates. All rights reserved." and "Privacy Policy", "Terms of Use", "Cookie preferences".

Challenge Lab Interface (Right):

- The URL is labclient.labondemand.com/Instructions/31e8ed78-8965-4c60-8a....
- The title of the challenge is "AWS S3-003: Can You Serve a Simple, Static Website with S3? [Advanced]" with "1 Hour Remaining".
- The interface includes tabs for "Instructions", "Resources", and "Help".
- A note states: "Amazon Web Services (AWS) is a dynamic, constantly evolving environment. As you complete your challenge lab, you may find that the provided guidance is not identical to what you encounter in the AWS environment. If you encounter a difference between AWS and the challenge lab instructions, please let us know by submitting feedback directly to [Challenge Lab Feedback](#) so that we may update the content in as timely a manner as possible."
- A large blue box in the center contains the text: "Instructions relating to the task will be included here".
- Navigation links at the bottom include "< Previous" and "Next: Enable versioning >".



Content Outline

Domain	% of Examination
Domain 1: Monitoring, Logging, and Remediation	20%
Domain 2: Reliability and Business Continuity	16%
Domain 3: Deployment, Provisioning, and Automation	18%
Domain 4: Security and Compliance	16%
Domain 5: Networking and Content Delivery	18%
Domain 6: Cost and Performance Optimization	12%
TOTAL:	100%



My recommendations

- Do the Solutions Architect and Developer Associate before the SysOps
- Make sure you get plenty of hands-on practice with AWS (exam labs are 20% of final score)
- Practice tests are very important

SECTION 2

Getting Started

AWS IAM Users, Groups, Roles, and Policies

AWS Identity and Access Management (IAM)

Policies are documents that define permissions and can be applied to users, groups and roles



IAM Policy



IAM User

An IAM user is an entity that represents a person or service



IAM Policy



IAM Group

Groups are collections of users and have policies attached to them

Roles are “assumed” by trusted entities and can be used for delegation

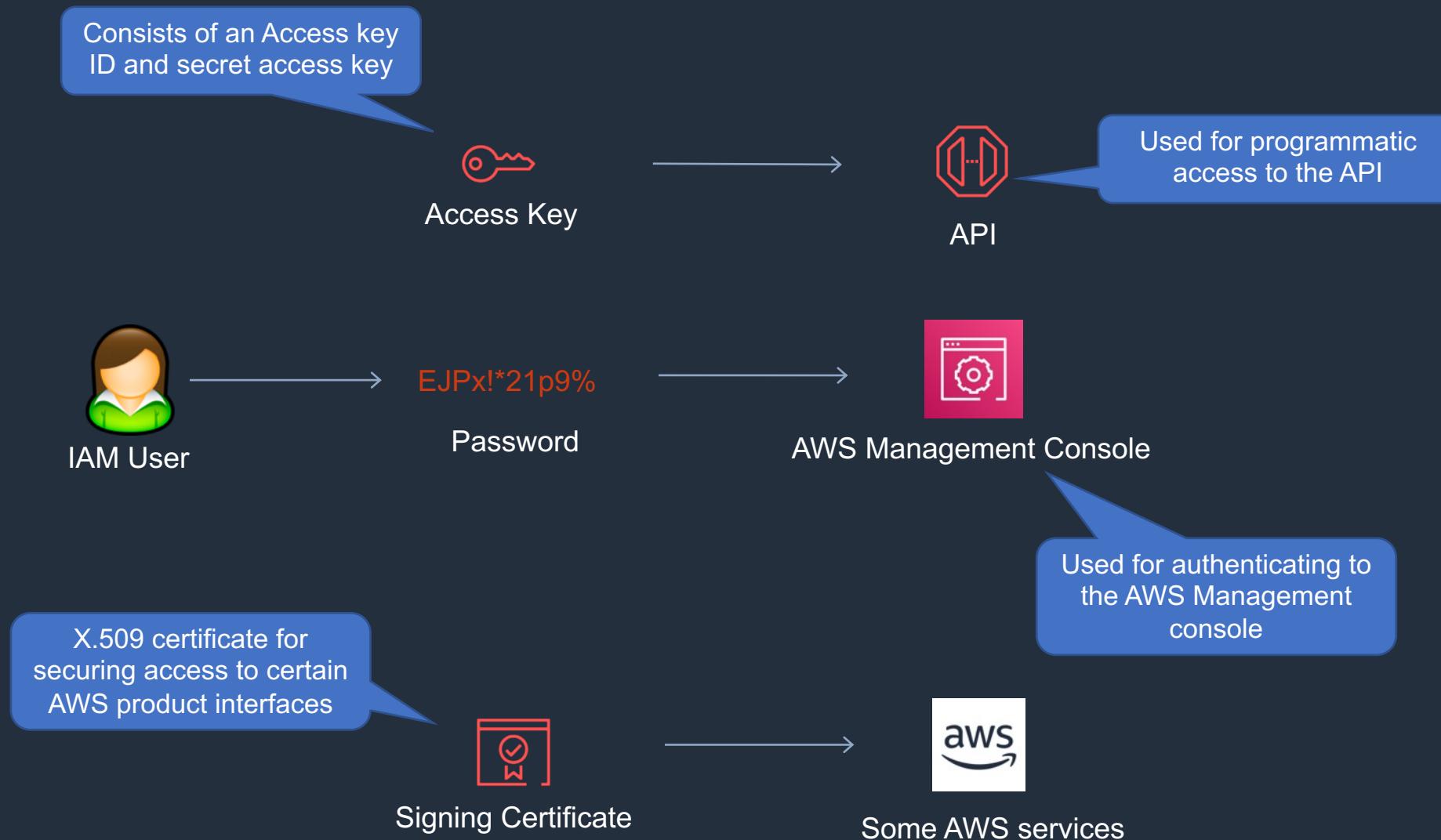


IAM Policy



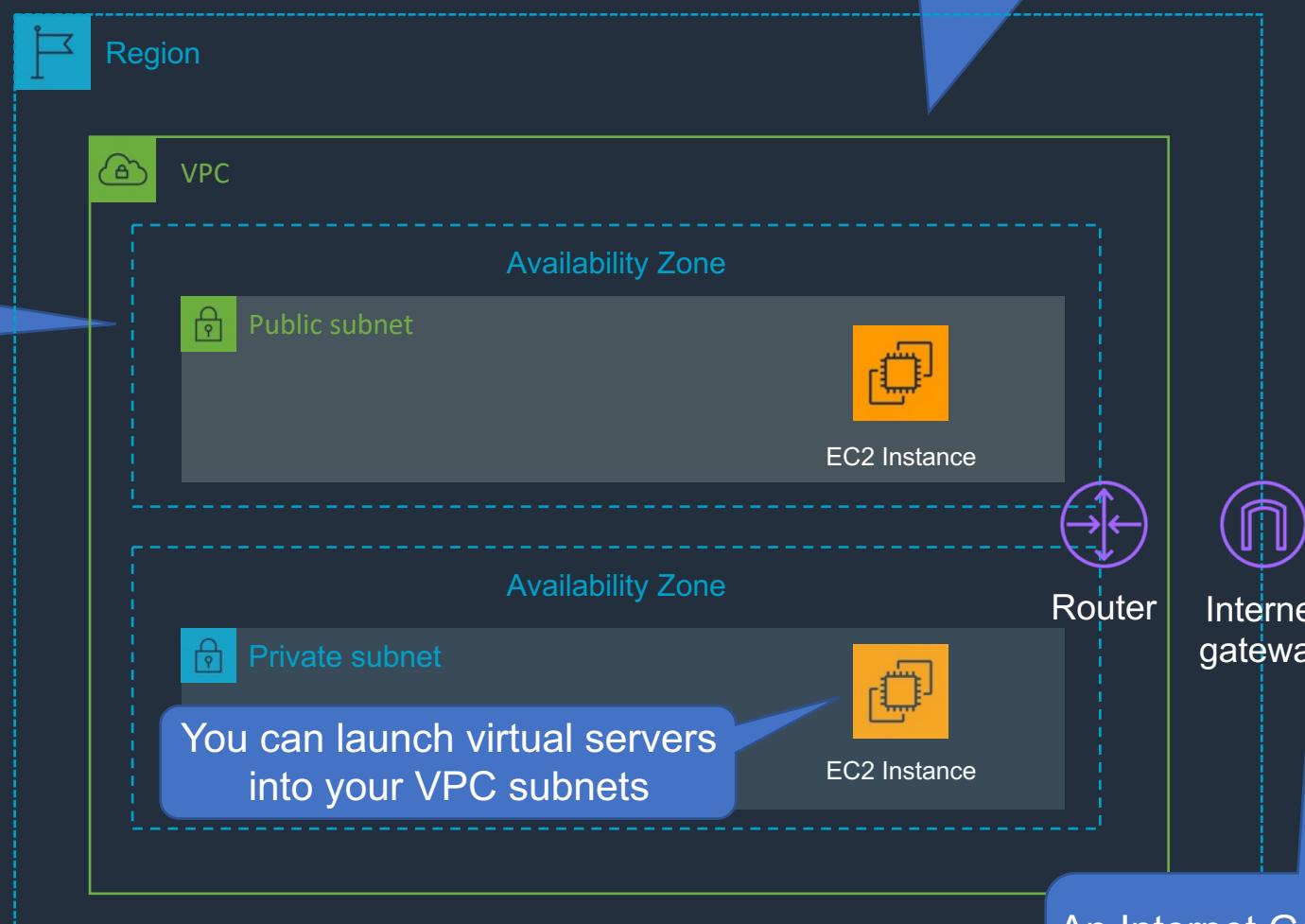
IAM Role

Authentication Methods



Amazon Virtual Private Cloud (VPC)

A VPC is a logically isolated portion of the AWS cloud within a region

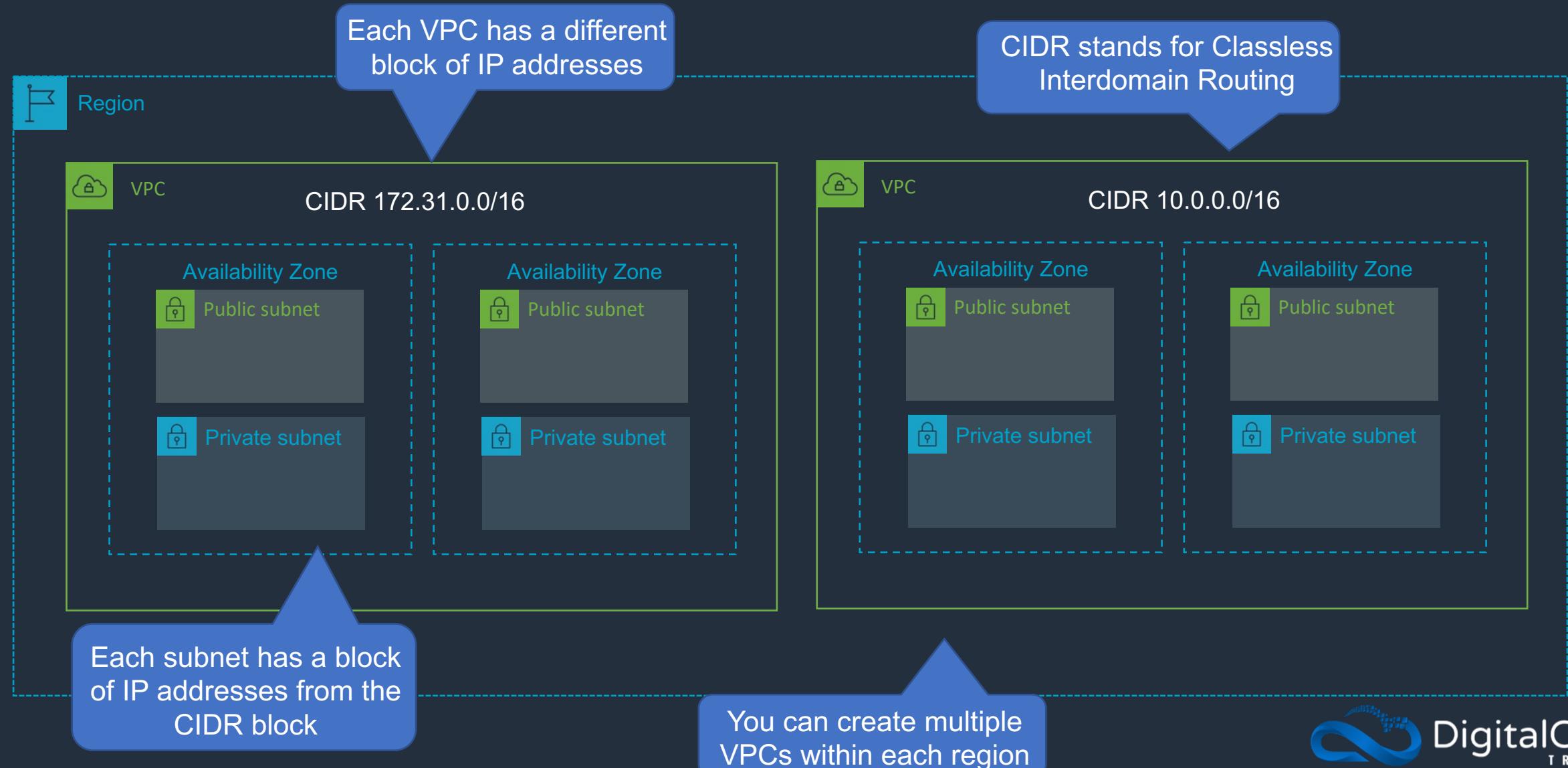


Main Route Table

Destination	Target
172.31.0.0/16	Local
0.0.0.0/0	igw-id

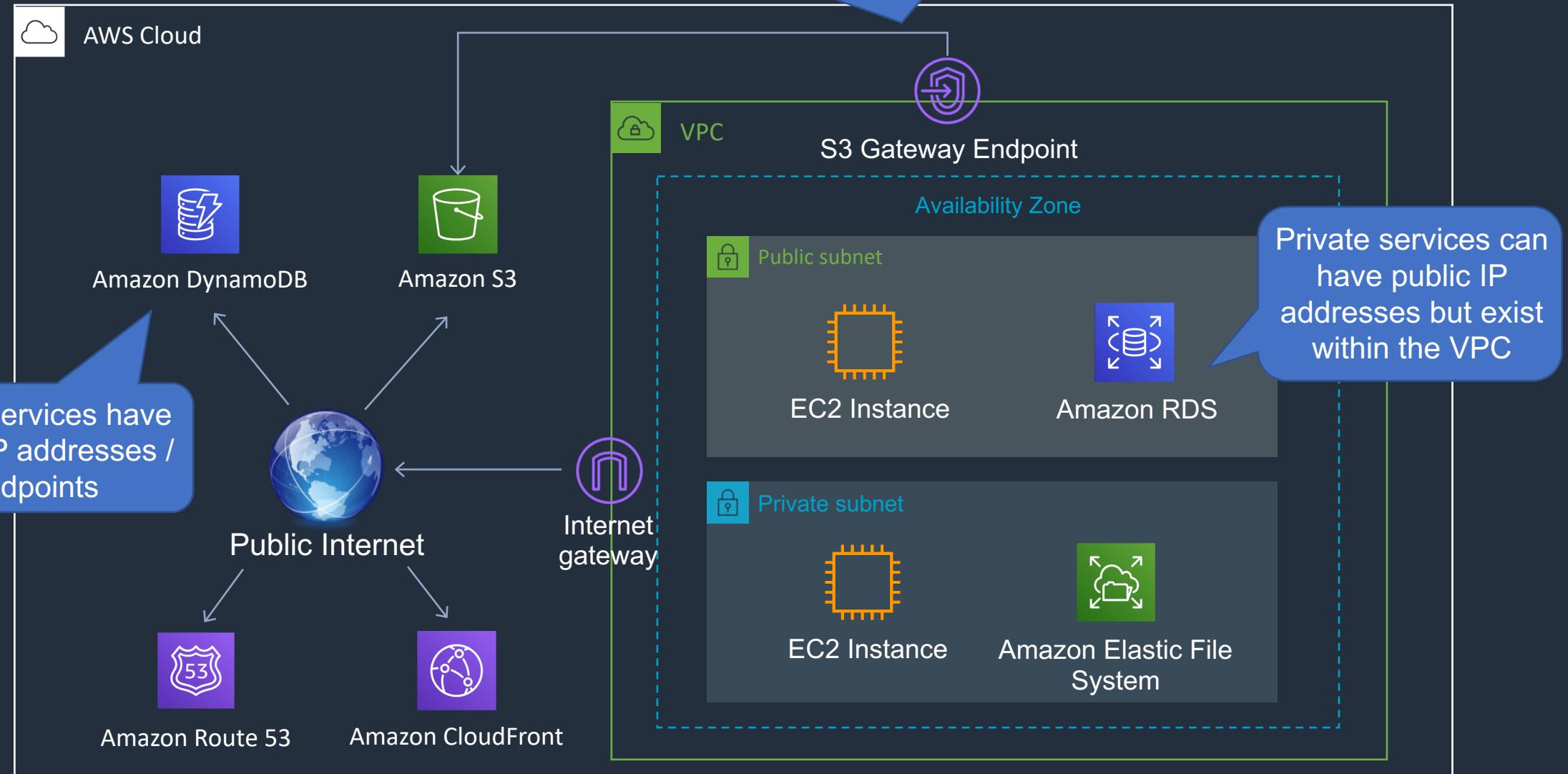
The route table is used to configure the VPC router

Multiple VPCs

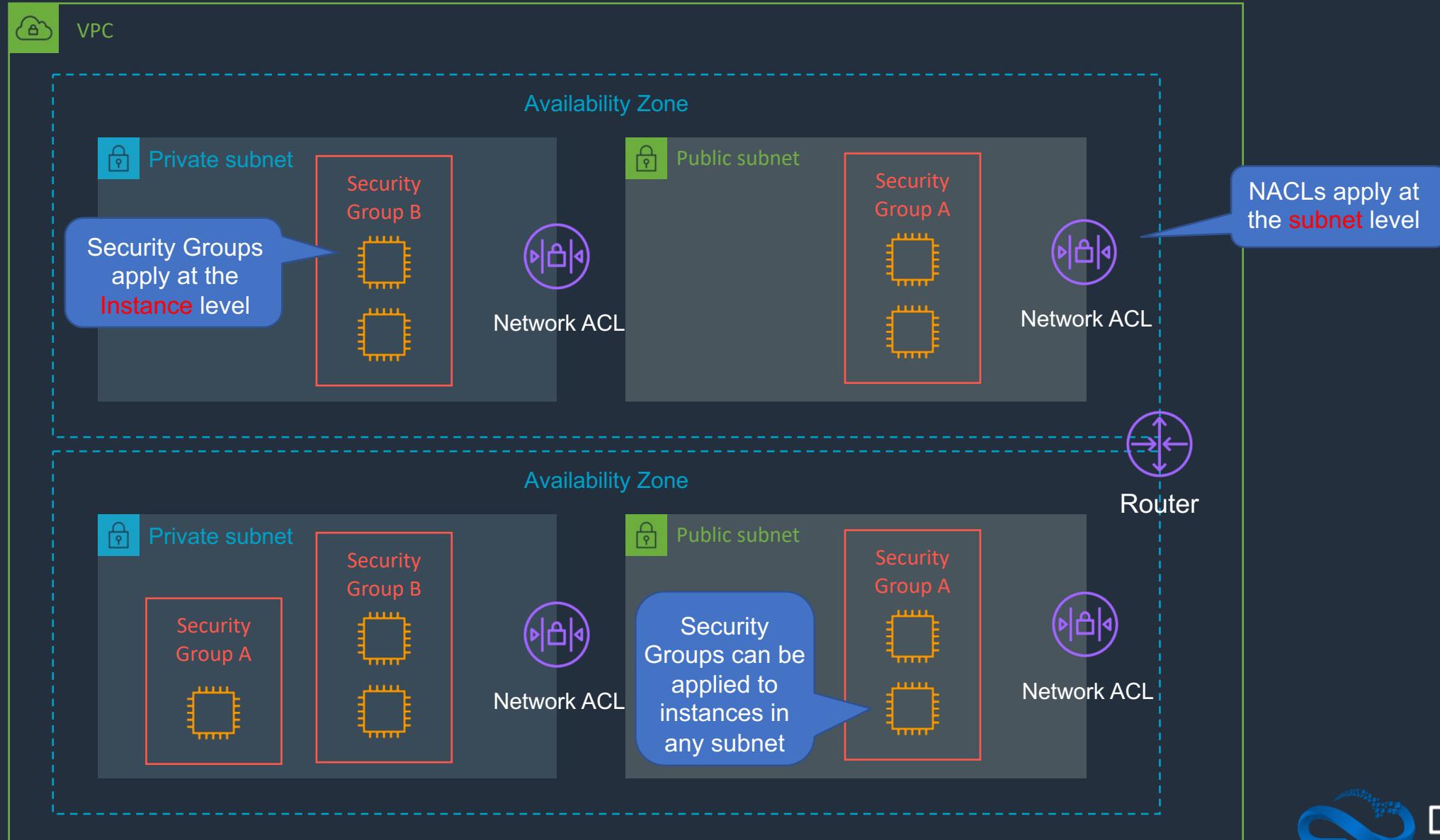


AWS Public and Private Services

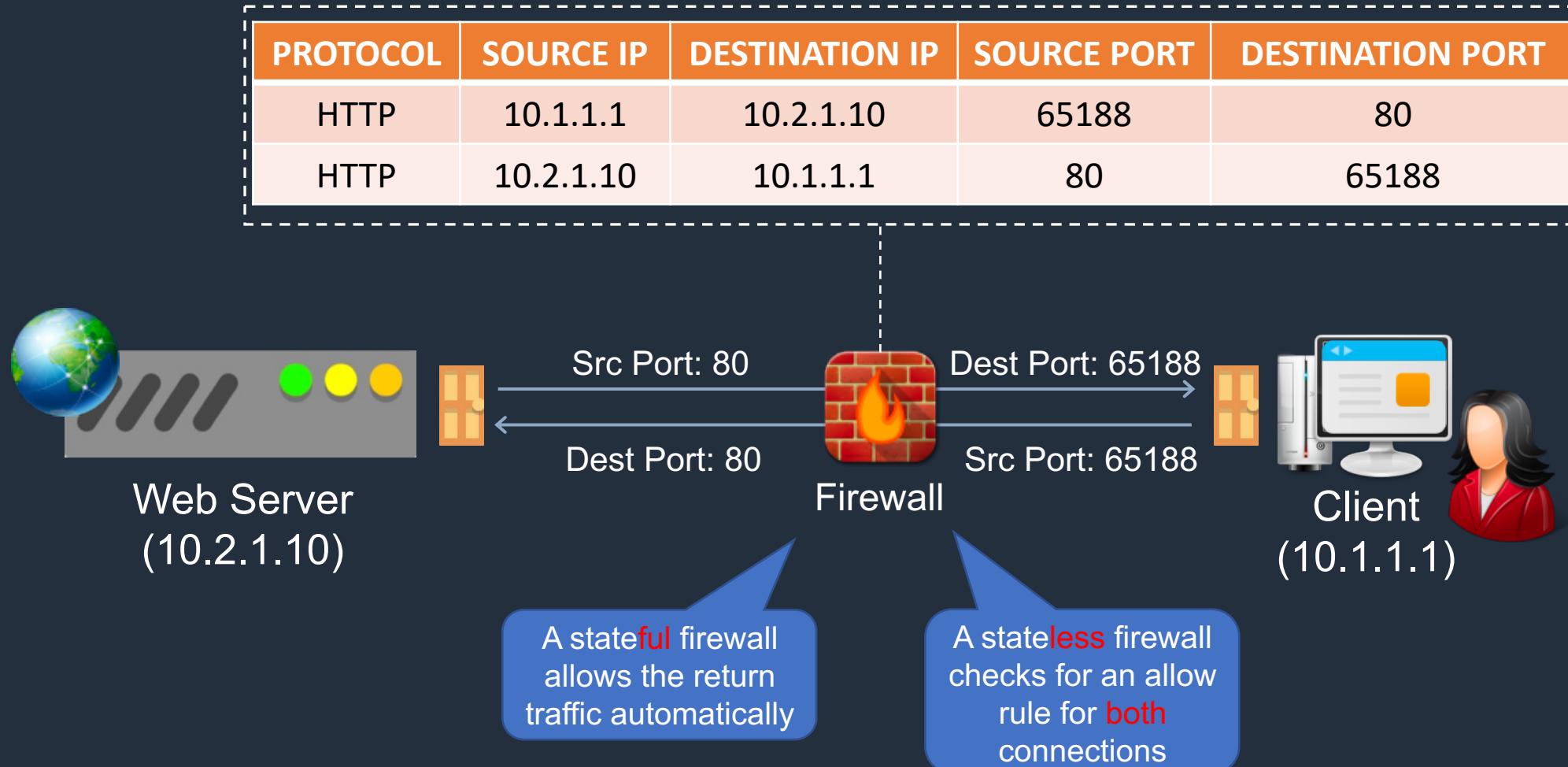
A VPC endpoint provides a private connection to a public services



Security Groups & Network Access Control Lists (NACLs)



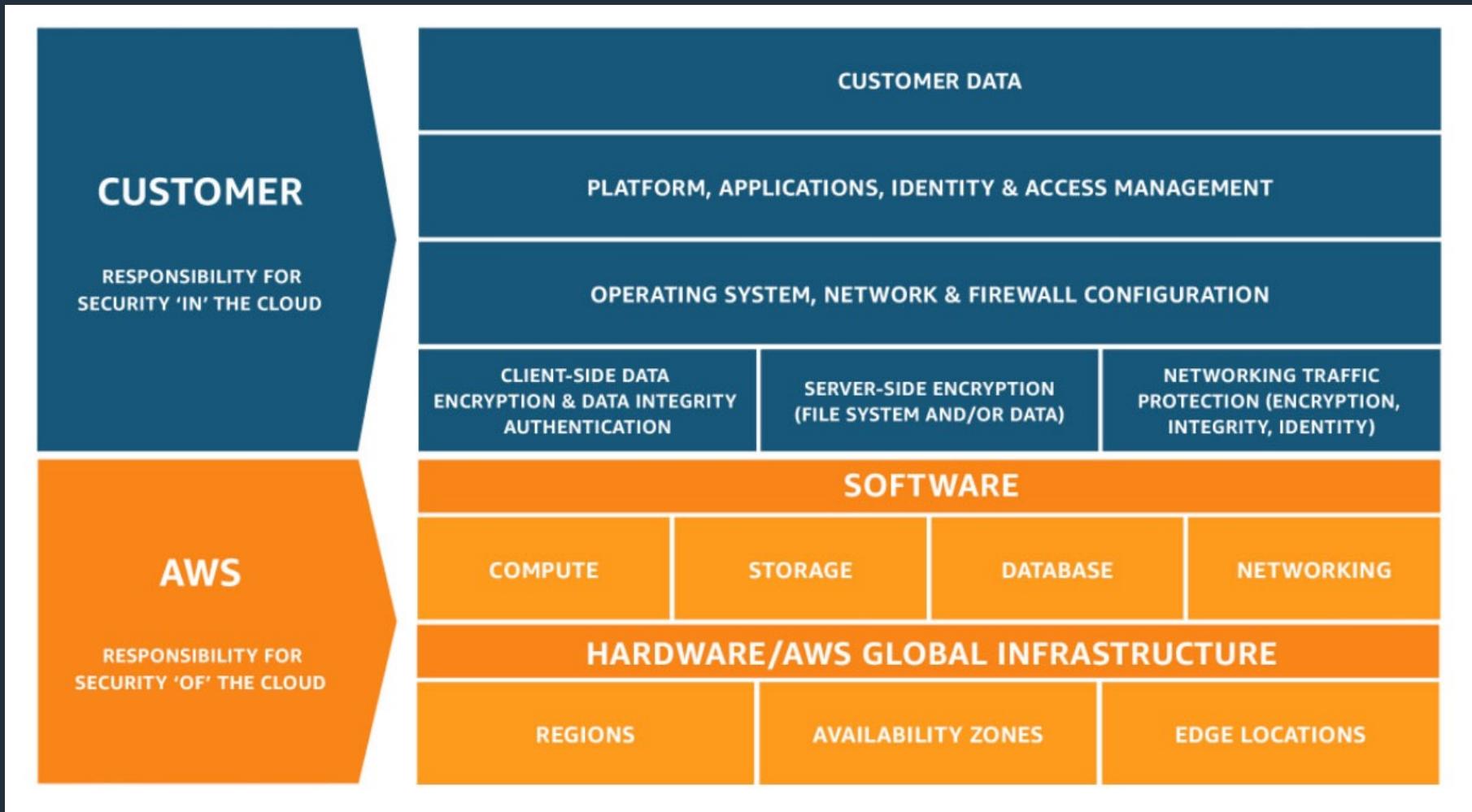
Stateful vs Stateless Firewalls



Security Groups & Network Access Control Lists (NACLs)

Security Group	Network ACL
Operates at the instance (interface) level	Operates at the subnet level
Supports allow rules only	Supports allow and deny rules
Stateful	Stateless
Evaluates all rules	Processes rules in order
Applies to an instance only if associated with a group	Automatically applies to all instances in the subnets its associated with

The Shared Responsibility Model



The Shared Responsibility Model - Examples

CUSTOMER RESPONSIBILITY



Bucket with objects



Role



Multi-Factor Authentication



Security Group



Patch management



Auto Scaling



Staff training



Data encryption



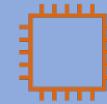
IAM User



Network ACL



SSL encryption



EC2 Instance



Elastic load balancer

AWS RESPONSIBILITY



Data center security



Network router



Network switch



Server



Storage



Database Server

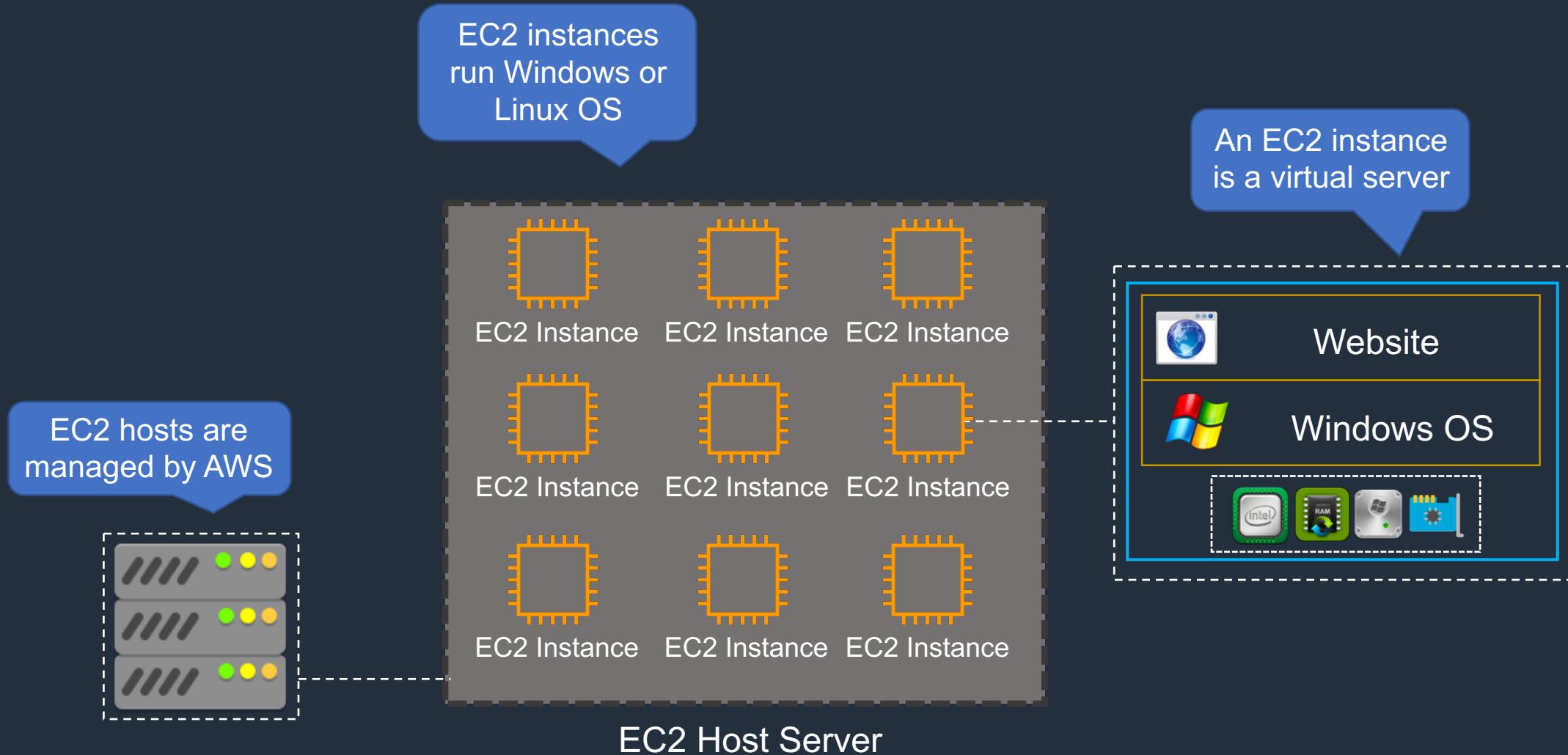


Disk drive

SECTION 3

Compute: Amazon EC2 and AWS Lambda

Amazon Elastic Compute Cloud



Launching an Amazon EC2 instance



Instance Type			
Family	Type	vCPUs	Memory (GiB)
General purpose	t2.micro	1	1
Compute optimized	c5n.large	2	5.25
Memory optimized	r5ad.large	2	16
Storage optimized	d2.xlarge	4	30.5
GPU instances	g2.2xlarge	8	15

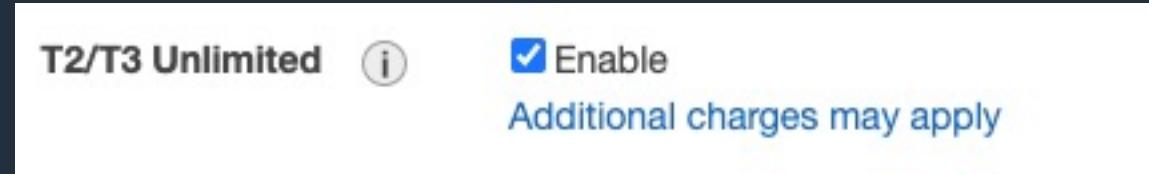
Amazon EC2 Reserved Instances

Burstable instances

- T3, T3a, and T2 instances, are designed to provide a baseline level of CPU performance with the ability to burst to a higher level when required
- Burstable performance instances are the only instance types that use credits for CPU usage
- A CPU credit provides for 100% utilization of a full CPU core for one minute
- Each burstable performance instance continuously earns (at a millisecond-level resolution) a set rate of CPU credits per hour, depending on the instance size

Amazon EC2 Reserved Instances

T2/T3 Unlimited

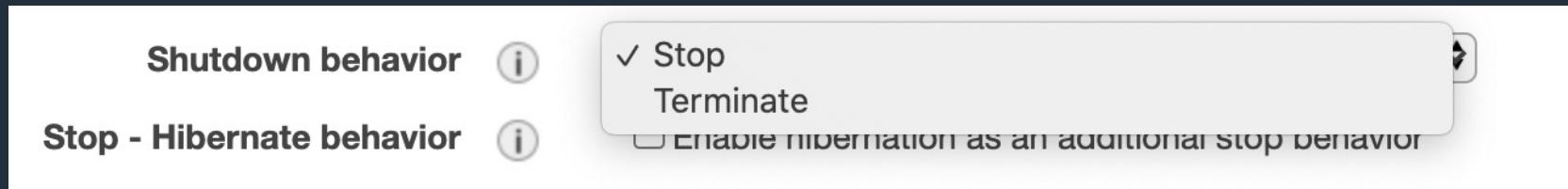


- T2 instances are a low-cost, general purpose instance type that provides a baseline level of CPU performance with the ability to burst above the baseline when needed
- T2 Unlimited instances can sustain high CPU performance for as long as a workload needs it
- The baseline performance and ability to burst are governed by CPU Credits
- T2 instances accumulate CPU Credits when they are idle, and consume CPU Credits when they are active

Launching an Amazon EC2 instance

Shutdown behavior

- Configure to Stop or Terminate (applies to OS-level shutdown)
- Can additionally enable hibernation (stores contents of RAM on the root volume)



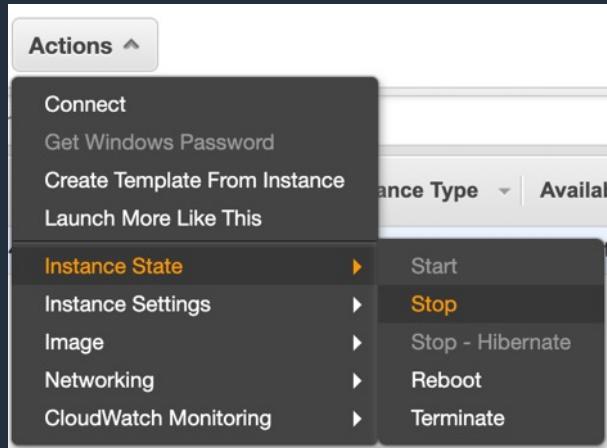
Termination Protection

- You can protect instances from being accidentally terminated
- Once enabled, you won't be able to terminate the instance via the API or the AWS Management Console until termination protection has been disabled

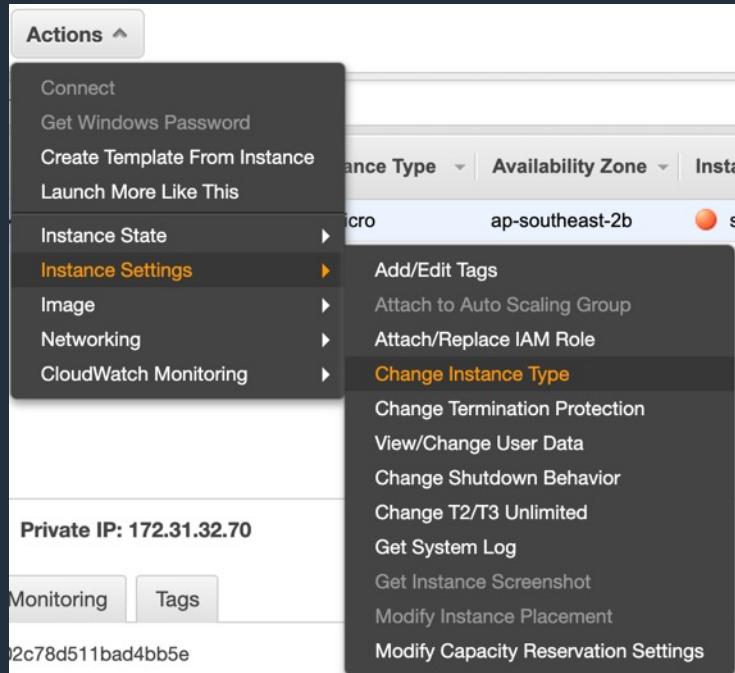


How to Change the EC2 Instance Type

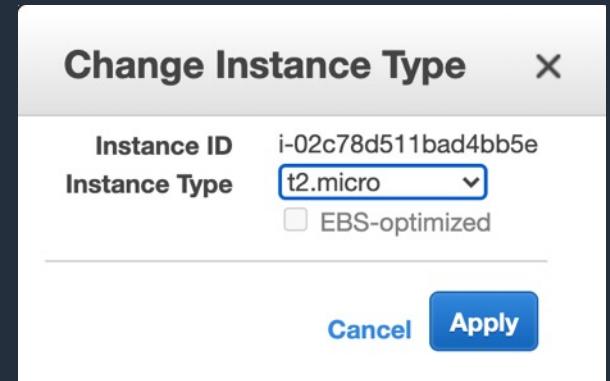
Stop the Instance



Select “Change Instance Type”



Choose the new instance type

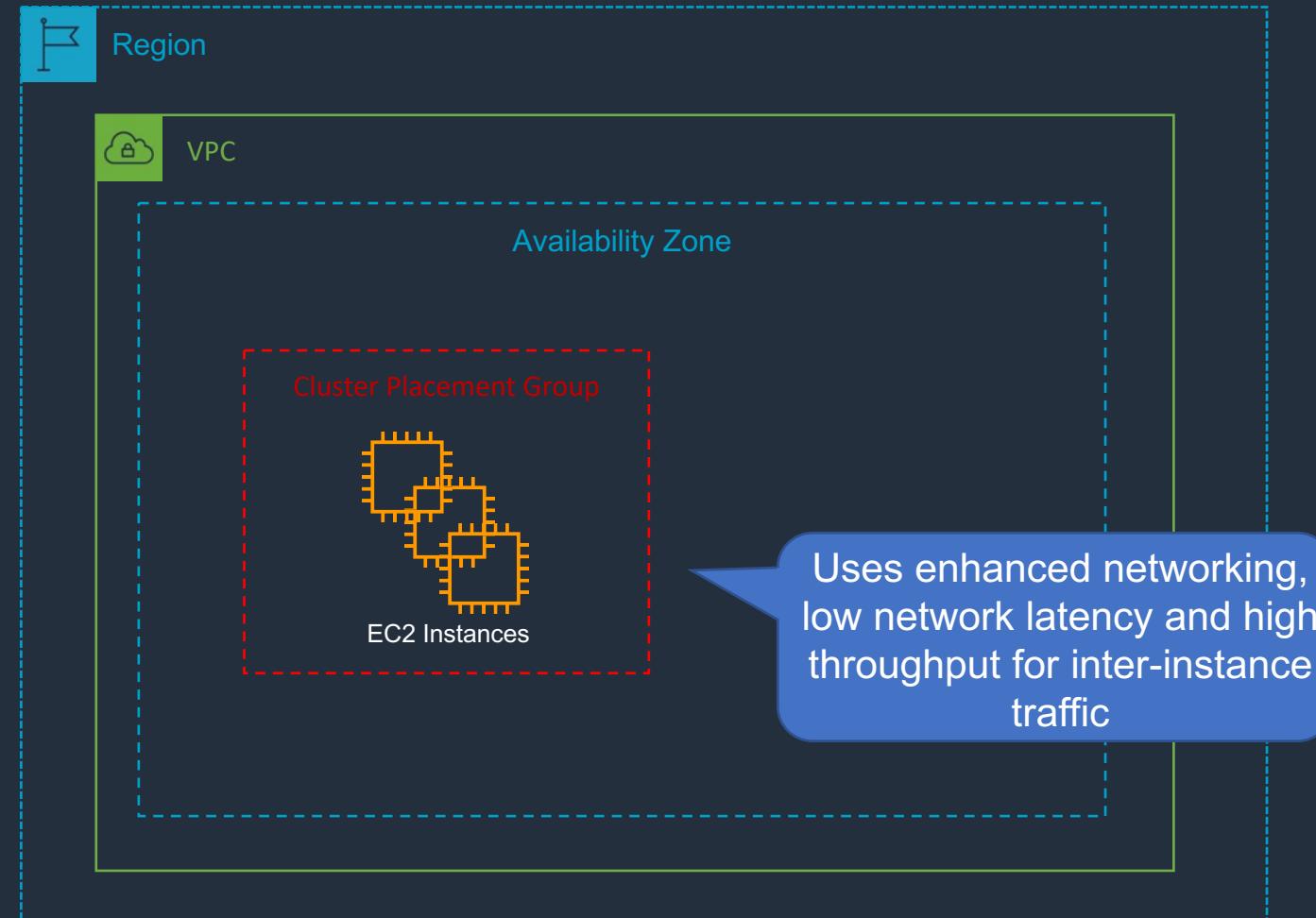


You can change instance types for EBS backed instances only

Amazon EC2 Placement Groups

- **Cluster** – packs instances close together inside an Availability Zone. This strategy enables workloads to achieve the low-latency network performance necessary for tightly-coupled node-to-node communication that is typical of HPC applications.
- **Partition** – spreads your instances across logical partitions such that groups of instances in one partition do not share the underlying hardware with groups of instances in different partitions. This strategy is typically used by large distributed and replicated workloads, such as Hadoop, Cassandra, and Kafka.
- **Spread** – strictly places a small group of instances across distinct underlying hardware to reduce correlated failures.

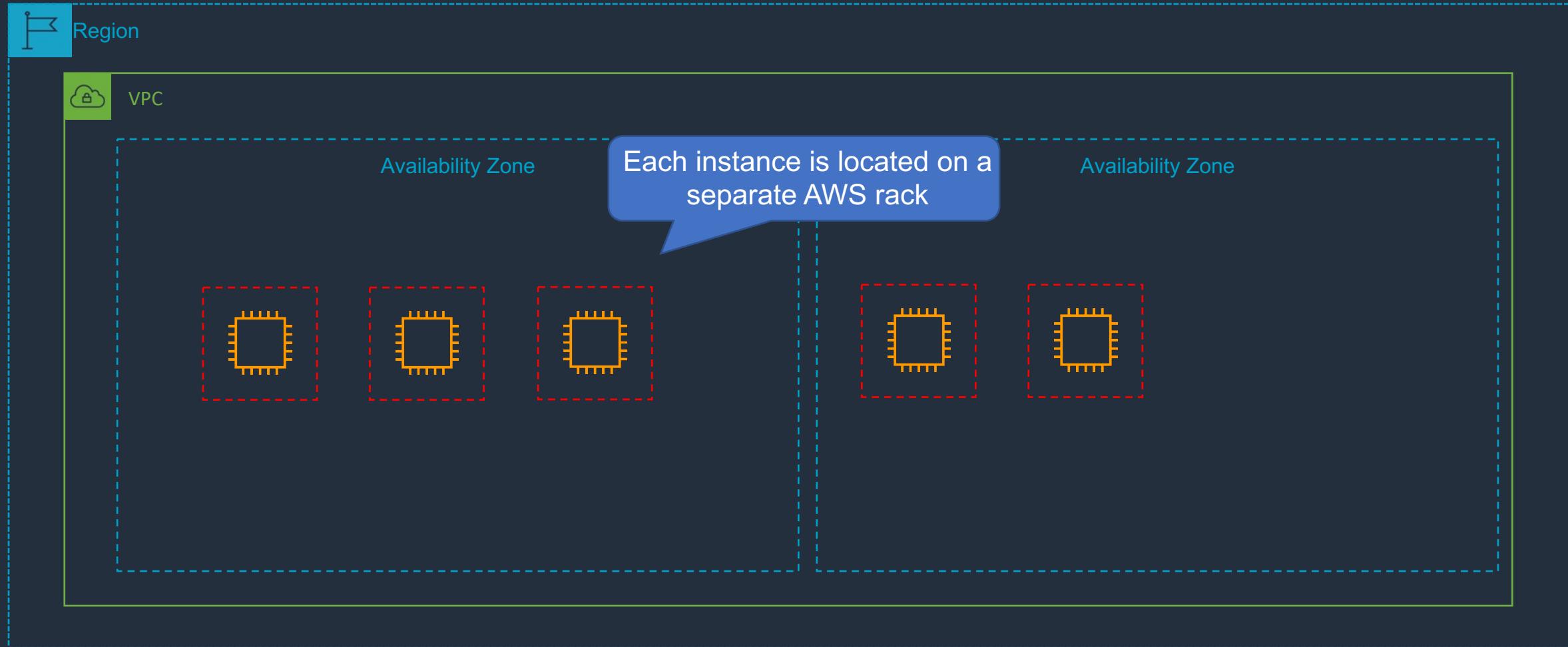
Cluster Placement Groups



Partition Placement Groups



Spread Placement Groups



Amazon EC2 Placement Groups

	Clustered	Spread	Partition
What	Instances are placed into a low-latency group within a single AZ	Instances are spread across underlying hardware	Instances are grouped into logical segments called partitions which use distinct hardware
When	Need low network latency and/or high network throughput	Reduce the risk of simultaneous instance failure if underlying hardware fails	Need control and visibility into instance placement
Pros	Get the most out of enhanced networking Instances	Can span multiple AZs	Reduces likelihood of correlated failures for large workloads.
Cons	Finite capacity: recommend launching all you might need up front	Maximum of 7 instances running per group, per AZ	Partition placement groups are not supported for Dedicated Hosts

Amazon EC2 Pricing Models

On-Demand	Reserved Instances	Savings Plans	Spot
No upfront fee	Options: No upfront, partial upfront or all upfront	Options: No upfront, partial upfront or all upfront	No upfront fee
Charged by hour or second	Charged by hour or second	Charged based on \$/hour	Charged by hour or second
No commitment	1-year or 3-year commitment	1-year or 3-year commitment	No commitment
Ideal for short term needs or unpredictable workloads	Ideal for steady-state workloads and predictable usage	More flexibility: Applies across Regions and instance families/types	Ideal for cost-sensitive, compute intensive use cases that can withstand interruption

Amazon EC2 Reserved Instances

A Reserved Instance has four instance attributes that determine its price:

- Instance type: For example, m4.large
- Region: The Region in which the Reserved Instance is purchased
- Tenancy: Whether your instance runs on shared (default) or single-tenant (dedicated) hardware
- Platform: The operating system; for example, Windows or Linux/Unix

Amazon EC2 Reserved Instances

Term commitment:

- One-year: A year is defined as 31536000 seconds (365 days)
- Three-year: Three years is defined as 94608000 seconds (1095 days)

Amazon EC2 Reserved Instances

Payment Options

- All Upfront: Full payment is made at the start of the term, with no other costs or additional hourly charges incurred for the remainder of the term, regardless of hours used
- Partial Upfront: A portion of the cost must be paid upfront and the remaining hours in the term are billed at a discounted hourly rate, regardless of whether the Reserved Instance is being used
- No Upfront: You are billed a discounted hourly rate for every hour within the term, regardless of whether the Reserved Instance is being used

Amazon EC2 Reserved Instances

Offering class:

- Standard: These provide the most significant discount but can only be modified
- Convertible: These provide a lower discount than Standard Reserved Instances but can be exchanged for another Convertible Reserved Instance with different instance attributes

Amazon EC2 Reserved Instances

Standard Reserved Instance	Convertible Reserved Instance
<p>Some attributes, such as instance size, can be modified during the term; however, the instance family cannot be modified. You cannot exchange a Standard Reserved Instance, only modify it.</p>	<p>Can be exchanged during the term for another Convertible Reserved Instance with new attributes including instance family, instance type, platform, scope, or tenancy. You can also modify some attributes of a Convertible Reserved Instance.</p>
<p>Can be sold in the Reserved Instance Marketplace.</p>	<p>Cannot be sold in the Reserved Instance Marketplace.</p>

Amazon EC2 Dedicated Instances and Hosts

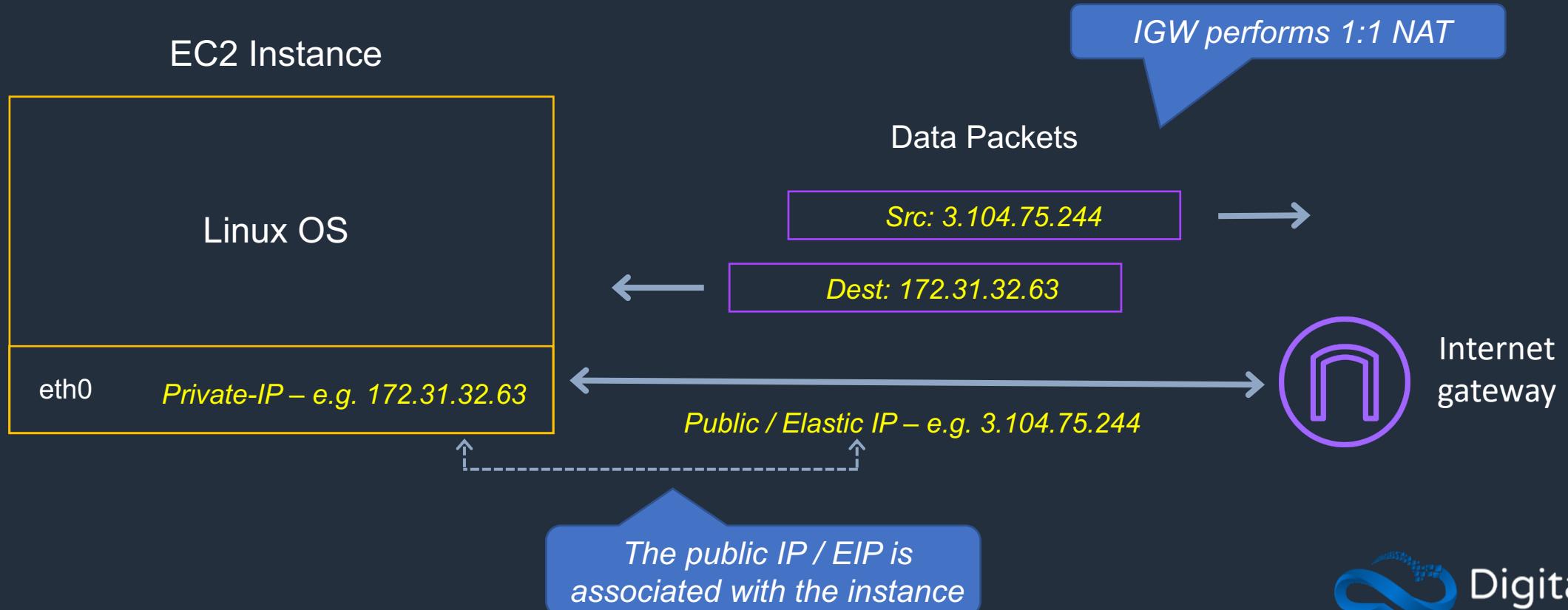
Characteristic	Dedicated Instances	Dedicated Hosts
Enables the use of dedicated physical servers	X	X
Per instance billing (subject to a \$2 per region fee)	X	
Per host billing		X
Visibility of sockets, cores, host ID		X
Affinity between a host and instance		X
Targeted instance placement		X
Automatic instance placement	X	X
Add capacity using an allocation request		X

Public, Private, and Elastic IP addresses

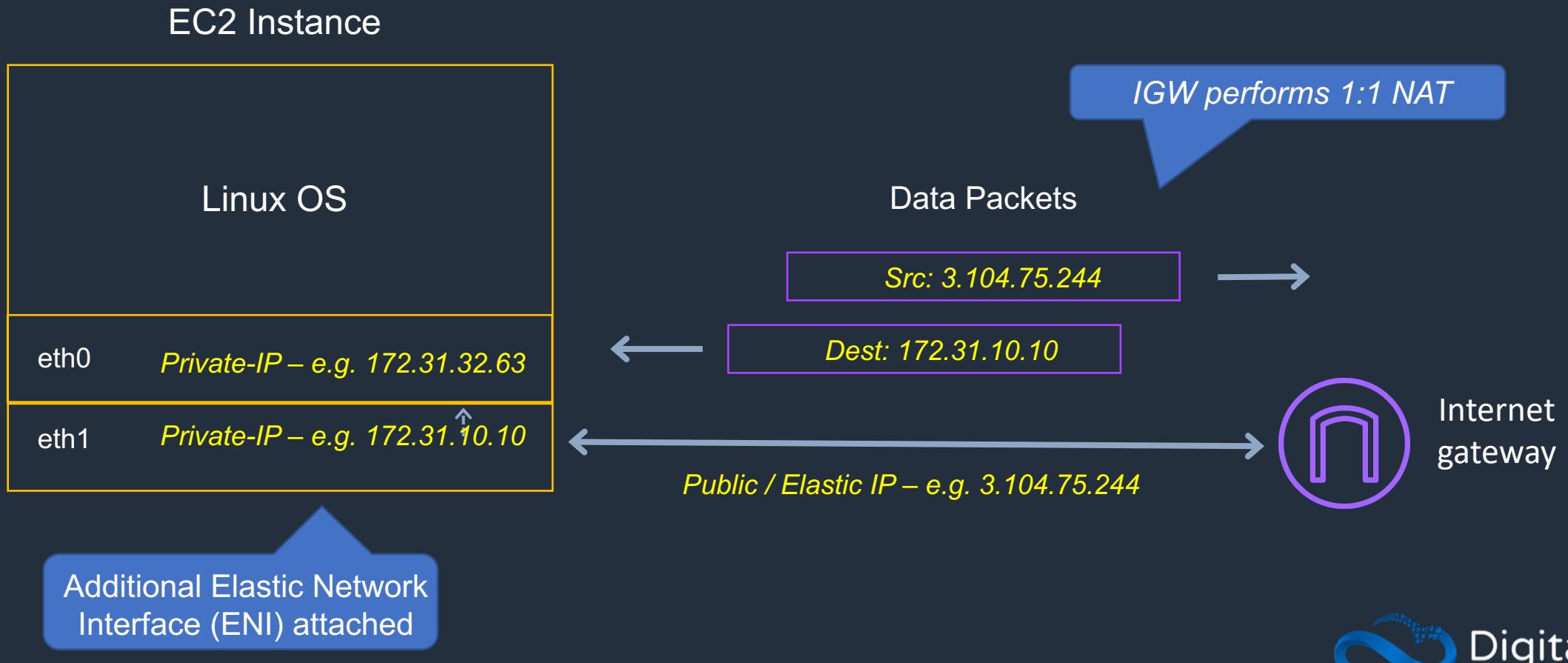
Name	Description
Public IP address	<p>Lost when the instance is stopped</p> <p>Used in Public Subnets</p> <p>No charge</p> <p>Associated with a private IP address on the instance</p> <p>Cannot be moved between instances</p>
Private IP address	<p>Retained when the instance is stopped</p> <p>Used in Public and Private Subnets</p>
Elastic IP address	<p>Static Public IP address</p> <p>You are charged if not used</p> <p>Associated with a private IP address on the instance</p> <p>Can be moved between instances and Elastic Network Adapters</p>

Public, Private and Elastic IPs

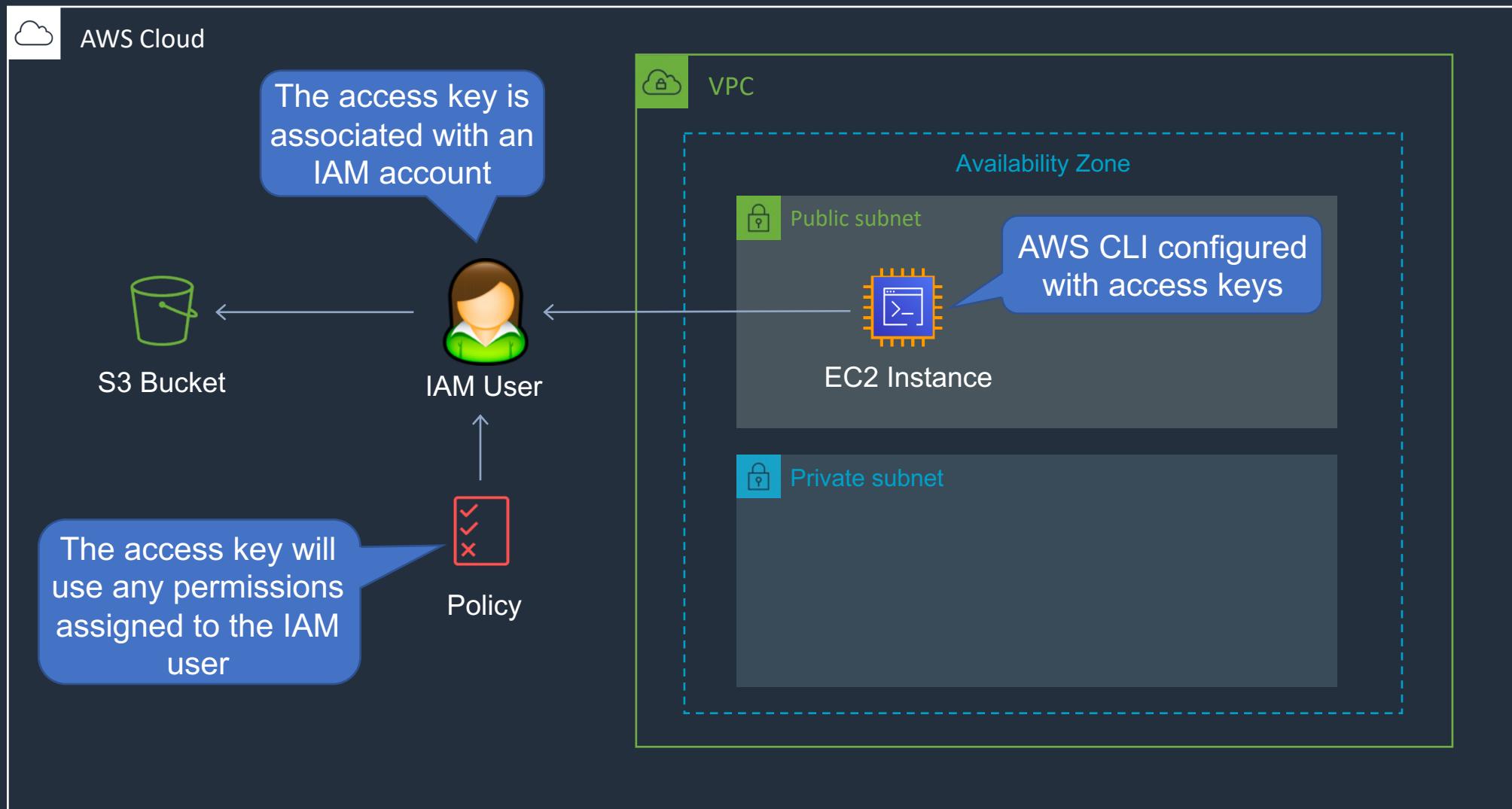
```
[ec2-user@ip-172-31-32-63 ~]$ ip addr show eth0
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 9001 qdisc pfifo_fast sta
te UP group default qlen 1000
    link/ether 06:06:db:ad:56:28 brd ff:ff:ff:ff:ff:ff
   inet 172.31.32.63/20 brd 172.31.47.255 scope global dynamic eth0
        valid_lft 3330sec preferred_lft 3330sec
    inet6 fe80::406:dbff:fead:5628/64 scope link
        valid_lft forever preferred_lft forever
```



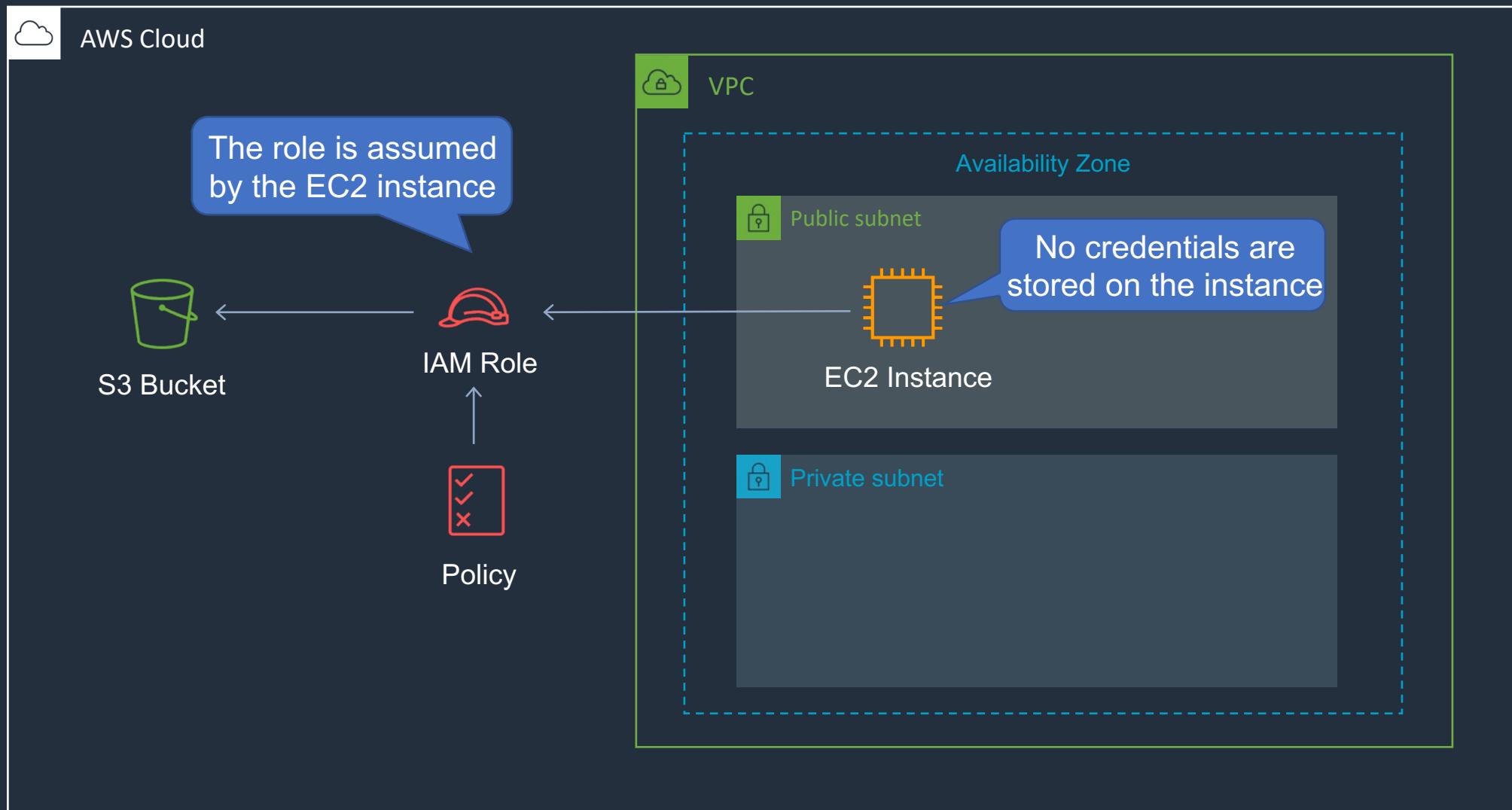
Public, Private and Elastic IPs – Additional ENI



Accessing other AWS Services Using Access Keys

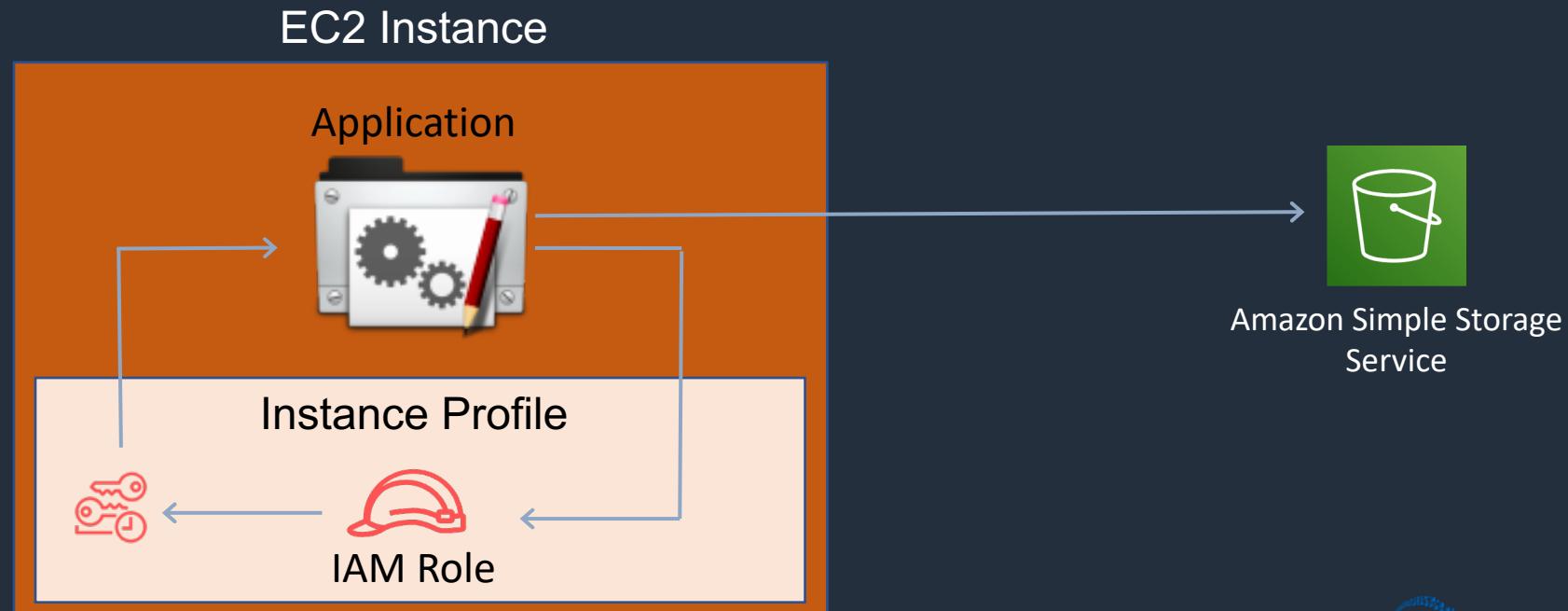


Accessing other AWS Services Using IAM Roles



IAM Instance Profiles

- An instance profile is a container for an IAM role that you can use to pass role information to an EC2 instance when the instance starts
- An instance profile can contain only one IAM role, although a role can be included in multiple instance profiles

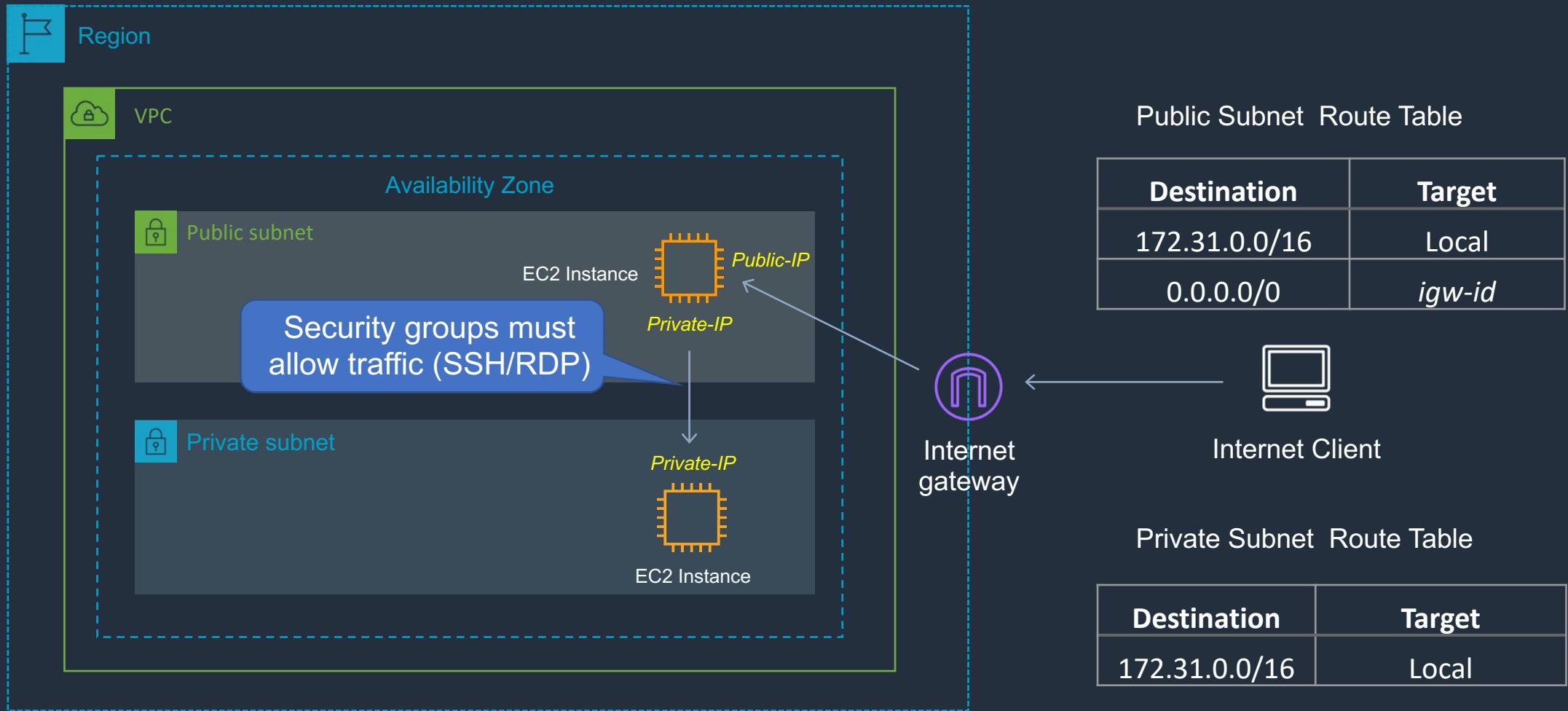


IAM Instance Profiles

You can use the following AWS CLI commands to work with instance profiles:

- Create an instance profile: `aws iam create-instance-profile`
- Add a role to an instance profile: `aws iam add-role-to-instance-profile`
- List instance profiles: `aws iam list-instance-profiles`, `aws iam list-instance-profiles-for-role`
- Get information about an instance profile: `aws iam get-instance-profile`
- Remove a role from an instance profile: `aws iam remove-role-from-instance-profile`
- Delete an instance profile: `aws iam delete-instance-profile`

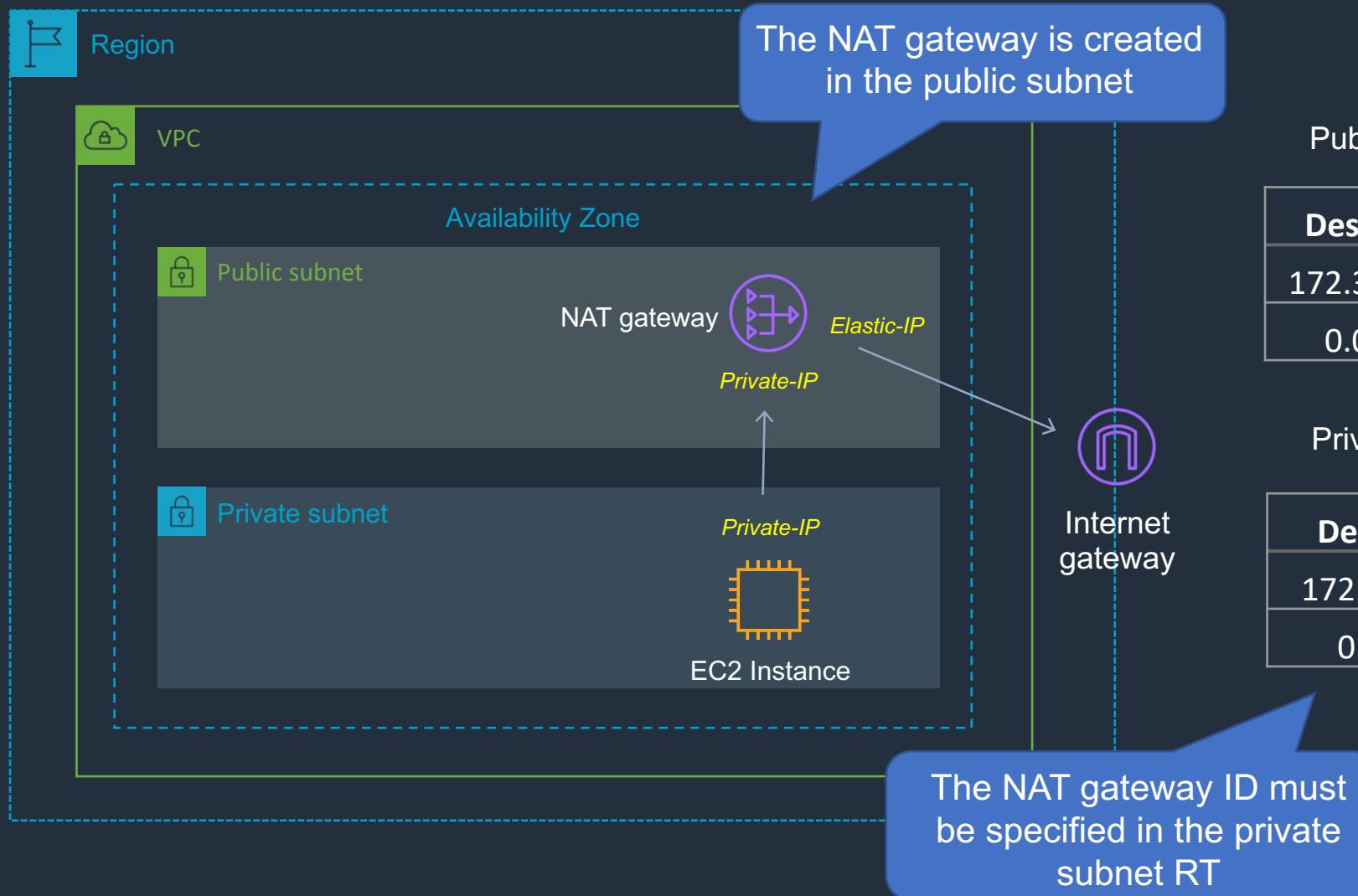
Private Subnets and Bastion Hosts



NAT Instance vs NAT Gateway

NAT Instance	NAT Gateway
Managed by you (e.g. software updates)	Managed by AWS
Scale up (instance type) manually and use enhanced networking	Elastic scalability up to 45 Gbps
No high availability – scripted/auto-scaled HA possible using multiple NATs in multiple subnets	Provides automatic high availability within an AZ and can be placed in multiple AZs
Need to assign Security Group	No Security Groups
Can use as a bastion host	Cannot access through SSH
Use an Elastic IP address or a public IP address with a NAT instance	Choose the Elastic IP address to associate with a NAT gateway at creation
Can implement port forwarding through manual customisation	Does not support port forwarding

Private Subnet with NAT Gateway



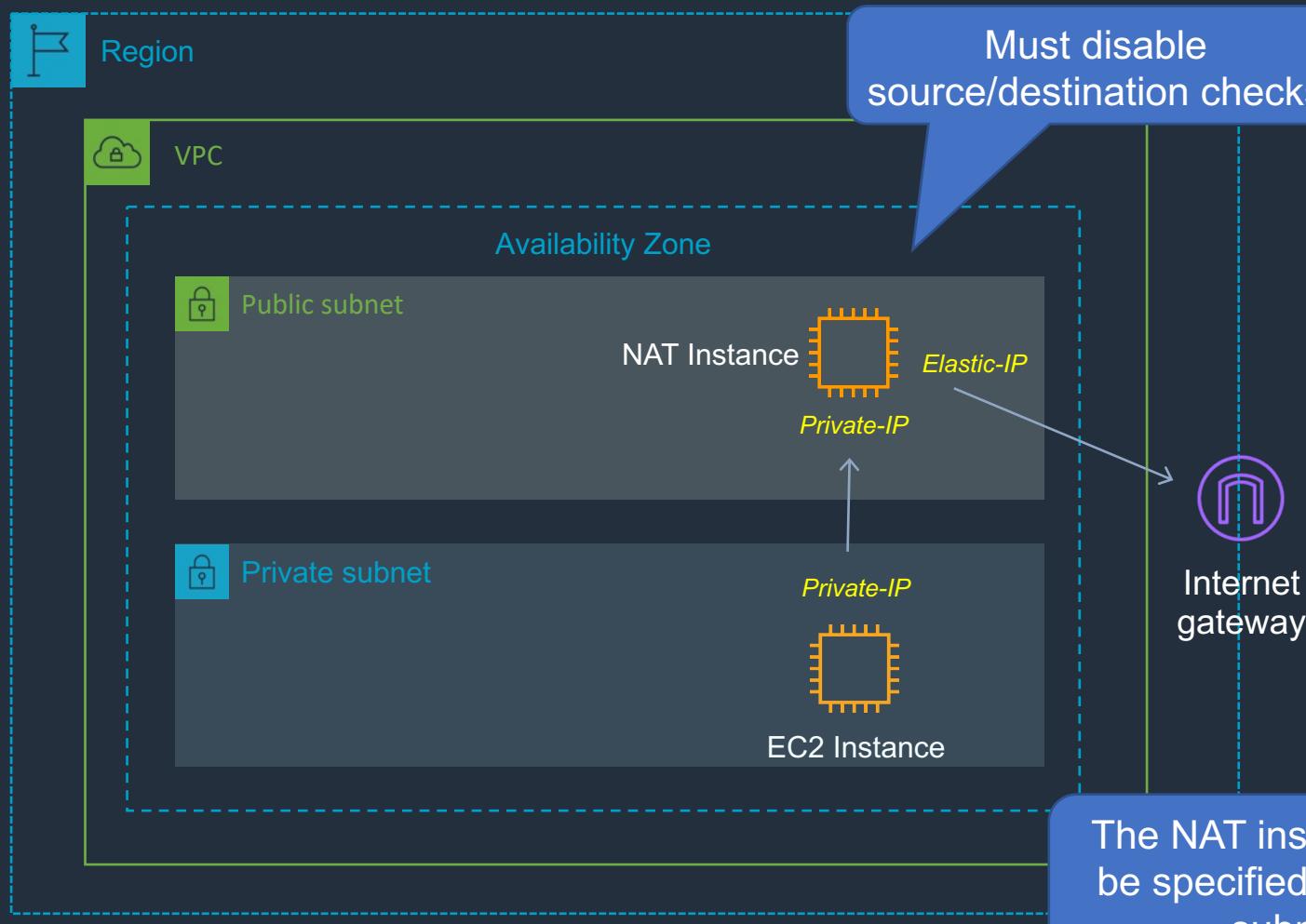
Public Subnet Route Table

Destination	Target
172.31.0.0/16	Local
0.0.0.0/0	<i>igw-id</i>

Private Subnet Route Table

Destination	Target
172.31.0.0/16	Local
0.0.0.0/0	<i>nat-gateway-id</i>

Private Subnet with NAT Instance



Public Subnet Route Table

Destination	Target
172.31.0.0/16	Local
0.0.0.0/0	<i>igw-id</i>

Private Subnet Route Table

Destination	Target
172.31.0.0/16	Local
0.0.0.0/0	<i>nat-instance-id</i>

Standard Amazon CloudWatch Metrics for EC2

InstanceId	Metric Name
i-0564f898a32460a7c	StatusCheckFailed_System
i-0564f898a32460a7c	StatusCheckFailed_Instance
i-0564f898a32460a7c	StatusCheckFailed
i-0564f898a32460a7c	MetadataNoToken
i-0564f898a32460a7c ▾	NetworkPacketsIn ▾
i-0564f898a32460a7c	NetworkPacketsOut
i-0564f898a32460a7c	CPUUtilization
i-0564f898a32460a7c	NetworkIn
i-0564f898a32460a7c	NetworkOut
i-0564f898a32460a7c	DiskReadBytes
i-0564f898a32460a7c	DiskWriteBytes
i-0564f898a32460a7c	DiskReadOps
i-0564f898a32460a7c	DiskWriteOps
i-0564f898a32460a7c	CPUCreditUsage
i-0564f898a32460a7c	CPUCreditBalance
i-0564f898a32460a7c	CPUSurplusCreditBalance
i-0564f898a32460a7c	CPUSurplusCreditsCharged

There are NO metrics for memory or disk utilization

Custom Amazon CloudWatch Metrics for EC2

- Can publish metrics using the API or AWS CLI
- Example CLI command: aws cloudwatch **put-metric-data** --metric-name TEST --namespace MyNameSpace --unit Bytes --value 231434333 --dimensions InstanceId=1-23456789,InstanceType=m1.small
- Or you can use the Unified Amazon CloudWatch Agent
- Collects system-level metrics from EC2 and on-premises servers

InstanceId	InstanceType	objectname	Metric Name
i-0564f898a32460a7c	t2.micro	Memory	Memory % Committed Bytes In Use

InstanceId	InstanceType	instance	objectname	Metric Name
i-0564f898a32460a7c	t2.micro	C:	LogicalDisk	LogicalDisk % Free Space

Custom Amazon CloudWatch Metrics for EC2

The unified CloudWatch agent enables you to do the following:

- Collect more system-level metrics from Amazon EC2 instances across operating systems. The metrics can include in-guest metrics, in addition to the metrics for EC2 instances
- Collect system-level metrics from on-premises servers. These can include servers in a hybrid environment as well as servers not managed by AWS
- Retrieve custom metrics from your applications or services using the StatsD and collectd protocols.
- Collect logs from Amazon EC2 instances and on-premises servers, running either Linux or Windows Server
- You can download and install the CloudWatch agent manually using the command line, or you can integrate it with SSM

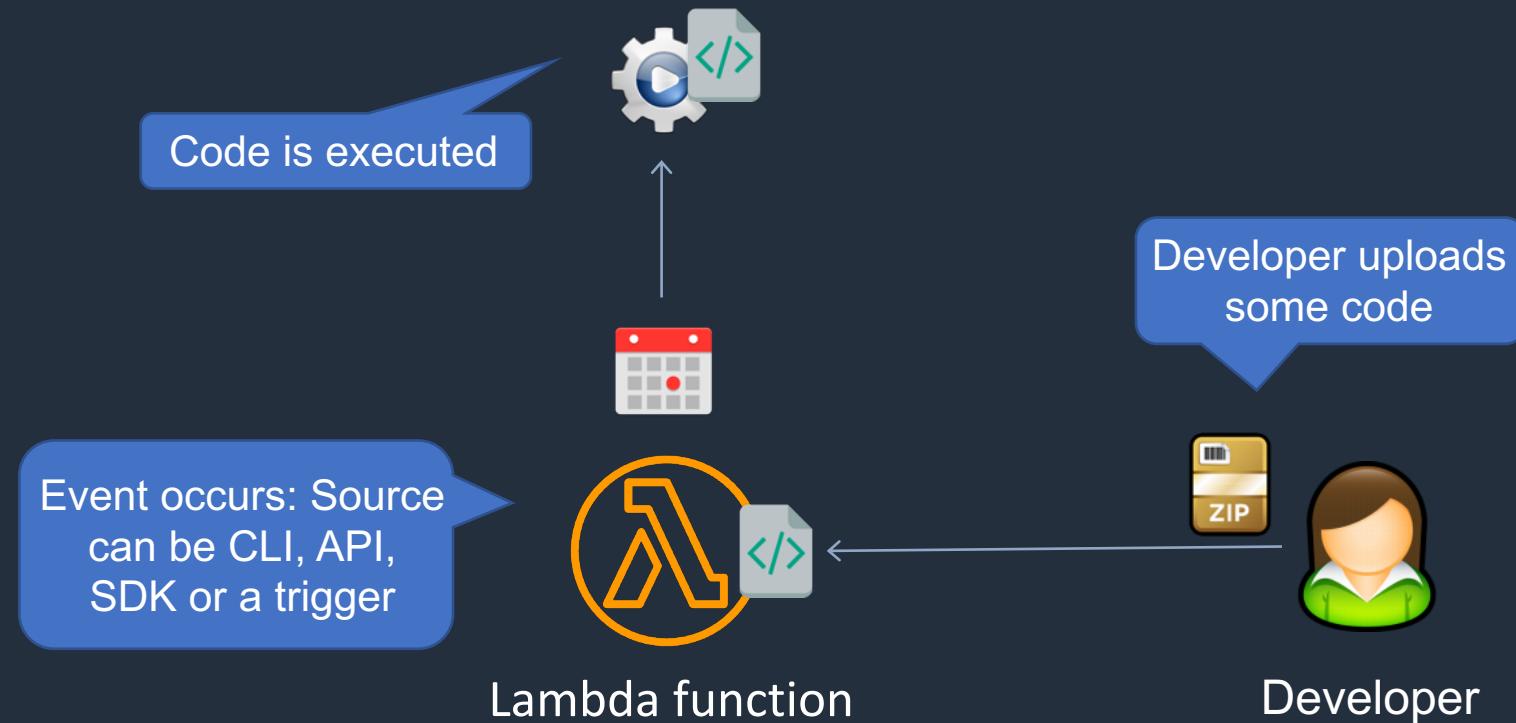
IAM Policy Example – Allow Full EC2 access in the us-east-2 Region

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Action": "ec2:*",  
            "Resource": "*",  
            "Effect": "Allow",  
            "Condition": {  
                "StringEquals": {  
                    "ec2:Region": "us-east-2"  
                }  
            }  
        }  
    ]  
}
```

IAM Policy Example – Limit Terminating EC2 Instances to an IP Address Range

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": ["ec2:TerminateInstances"],  
            "Resource": ["*"]  
        },  
        {  
            "Effect": "Deny",  
            "Action": ["ec2:TerminateInstances"],  
            "Condition": {  
                "NotIpAddress": {  
                    "aws:SourceIp": [  
                        "192.0.2.0/24",  
                        "203.0.113.0/24"  
                    ]  
                }  
            },  
            "Resource": ["*"]  
        }  
    ]  
}
```

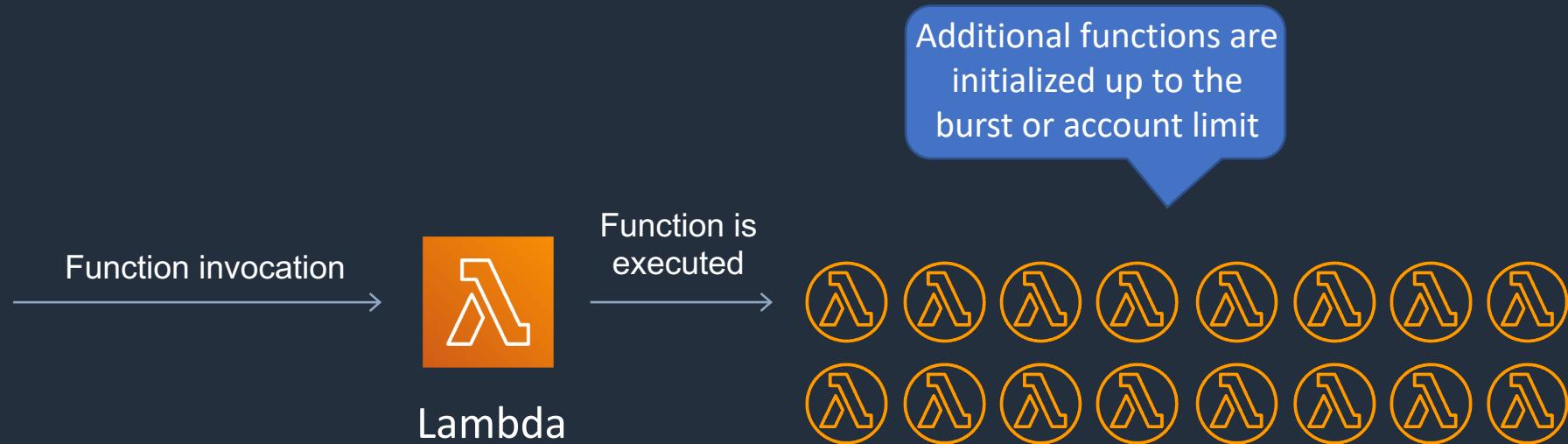
AWS Lambda



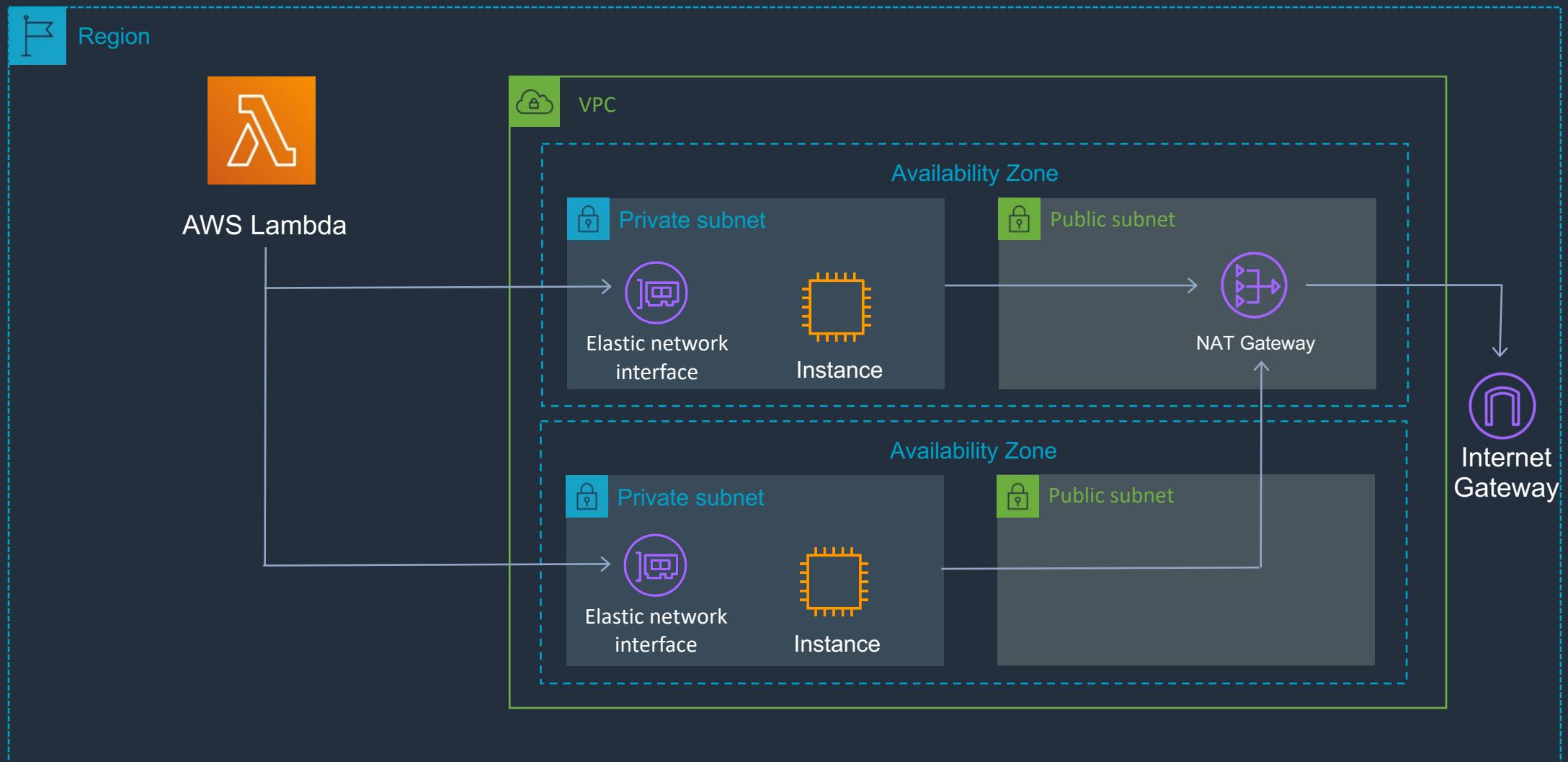
AWS Lambda

- Lambda is an event-driven compute service where AWS Lambda runs code in response to events such as changes to data in an S3 bucket or a DynamoDB table
- Lambda scales concurrently executing functions up to your default limit (1000)
- Lambda allocates CPU power proportional to the memory you specify using the same ratio as a general purpose EC2 instance type
- The maximum execution timeout is 15 minutes (900 seconds), default is 3 mins
- You can configure your Lambda function to access resources inside an Amazon VPC

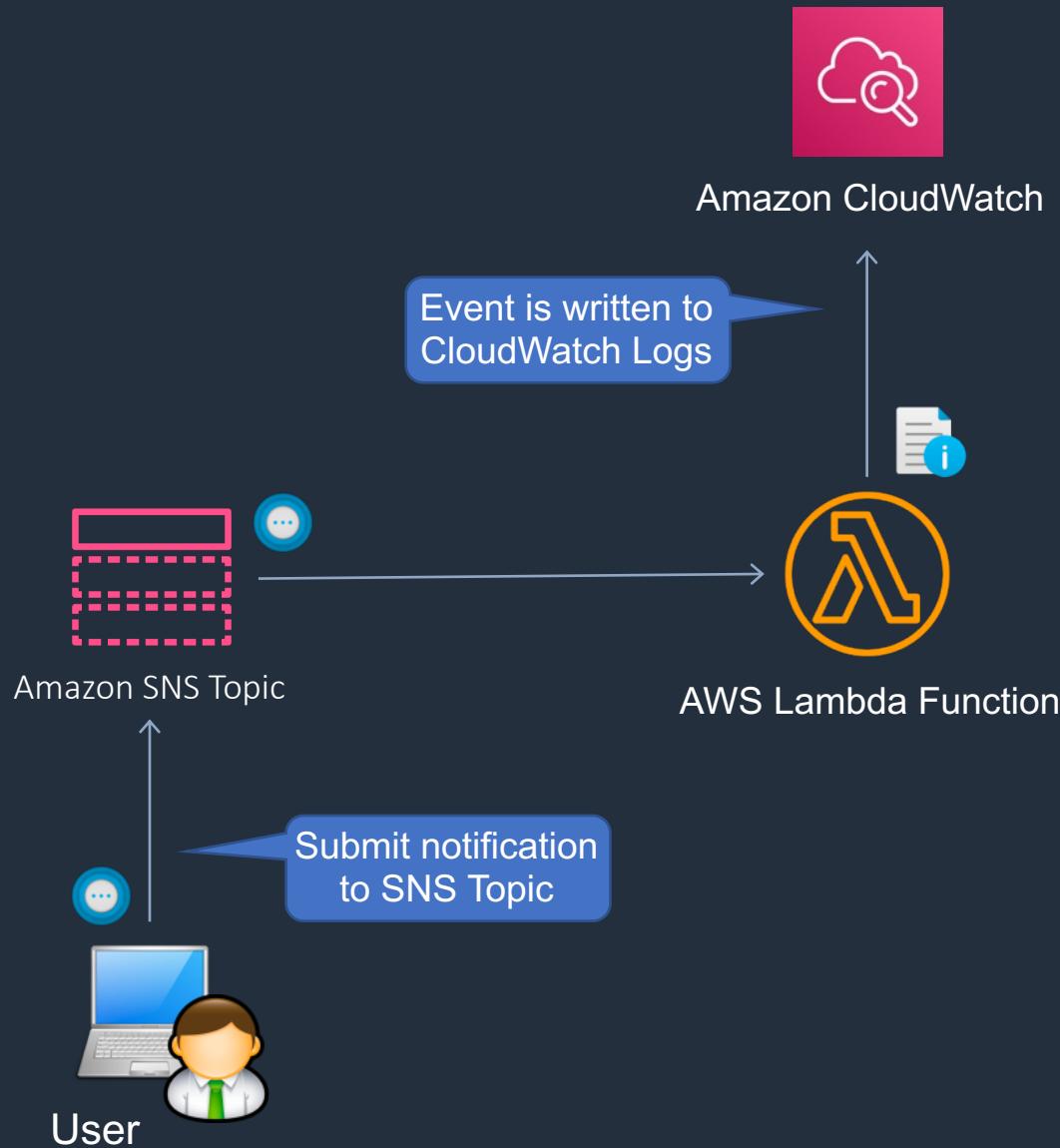
AWS Lambda - Concurrency



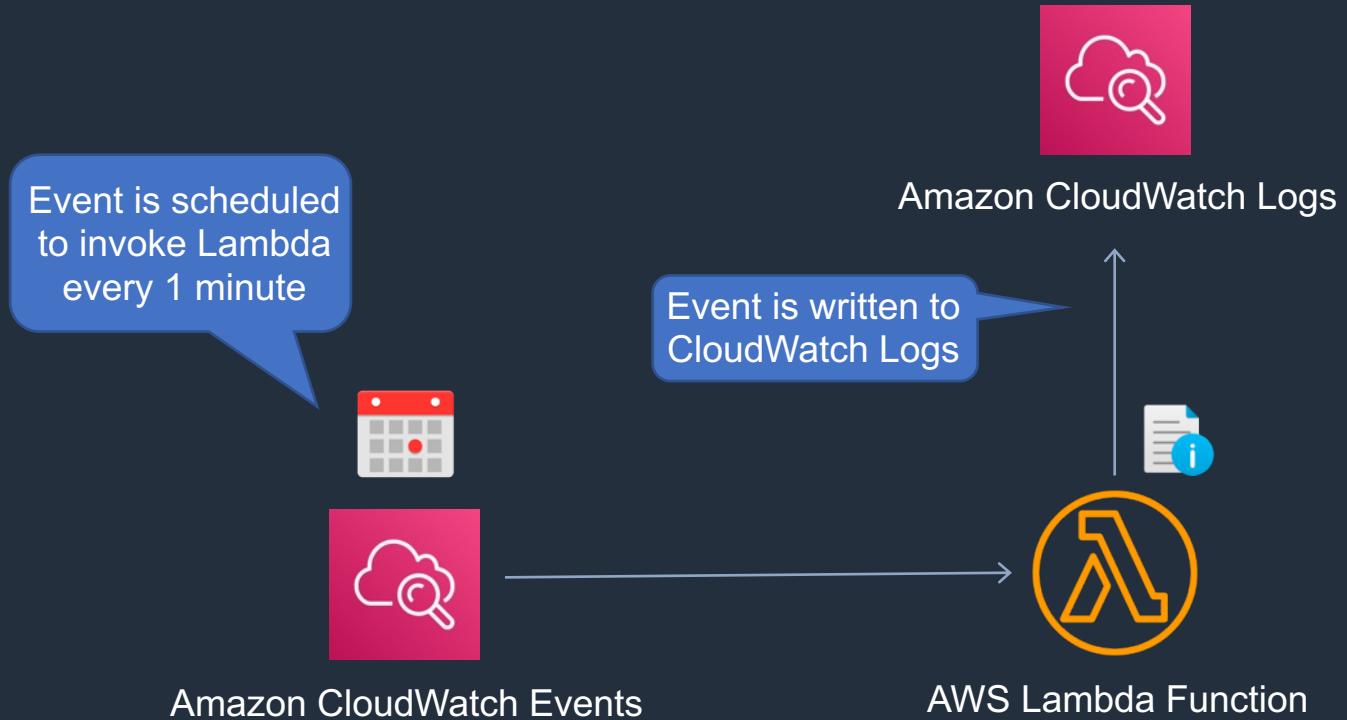
AWS Lambda in a Virtual Private Cloud (VPC)



Invoke Lambda Function with Amazon SNS



Invoke Lambda Function on a Schedule



Exam Scenarios

Exam Scenario	Solution
Administrator needs to check if any EC2 instances will be affected by scheduled hardware maintenance	Check the AWS Personal Health Dashboard
Scheduled hardware maintenance will affect a critical EC2 instance	Stop and start the instance to move it to different underlying hardware
When launching an EC2 instance the InsufficientInstanceCapacity error is experienced	This means AWS does not currently have enough capacity to service the request for that instance type. Try a different AZ or instance type
The error InstanceLimitExceeded is experienced when launching EC2 instances	EC2 instance limits have been reached, need to contact support to request an increased limit

Exam Scenarios

Exam Scenario	Solution
System status checks are failing for an EC2 instance	Stop and start again to move to a new host
For security and compliance reasons EC2 instances must not be able to access the internet	Launch them in a private subnet without a NAT gateway or NAT instance
EC2 instances must communicate with an internet-based service which whitelists a single source IP address	Place the instances behind a NAT gateway as the device will have a single elastic IP address that can be whitelisted
A distributed app is running on EC2 and can handle processing interruptions. Determine the best pricing model to use	Use Spot instances as the application can handle it if the instances are terminated

Exam Scenarios

Exam Scenario	Solution
Define AWS' responsibilities for EC2 hardware according to the AWS Shared Responsibility Model	AWS are responsible for managing the health of the underlying hosts
A nightly job runs on EC2 and stores results in S3. Takes 2 hours using multiple on-demand instances. If it fails, it must start again. Determine the best pricing model to use	Request a Spot block for time period required
An asynchronous process runs on EC2 and feeds data to a data warehouse for weekly/monthly reporting. Determine the best pricing model to use	Use Spot instances as the asynchronous nature of the reporting means the app can handle interruption if AWS need the capacity back
Need to track EC2 and on-premise computer memory utilization	Install the unified CloudWatch agent on both EC2 and on-premises servers

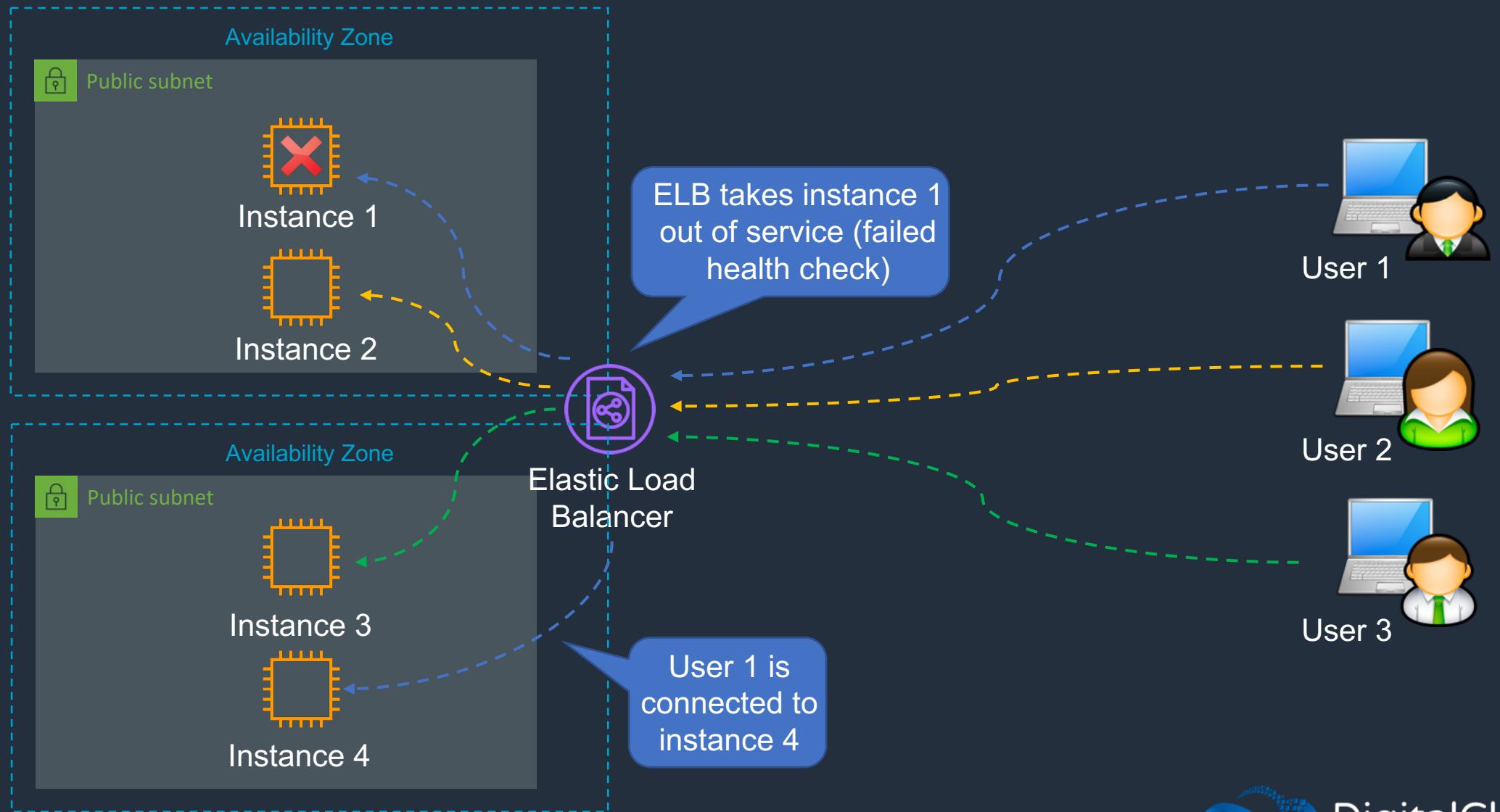
Exam Scenarios

Exam Scenario	Solution
Amazon EC2 Auto Scaling automatically terminates unhealthy instances but Administrator needs to keep the logs for subsequent analysis	Install the CloudWatch agent to stream logs to CloudWatch Logs
There is a suspected memory leak on an Amazon EC2 instance	Install the CloudWatch agent to monitor memory utilization
An AWS Lambda function is expected to see a large increase in traffic and must scale	Ensure the concurrency limit is higher than the expected simultaneous executions
Need to invoke an AWS Lambda function every 15 minutes	Create an event rule in Amazon CloudWatch events to execute the function periodically

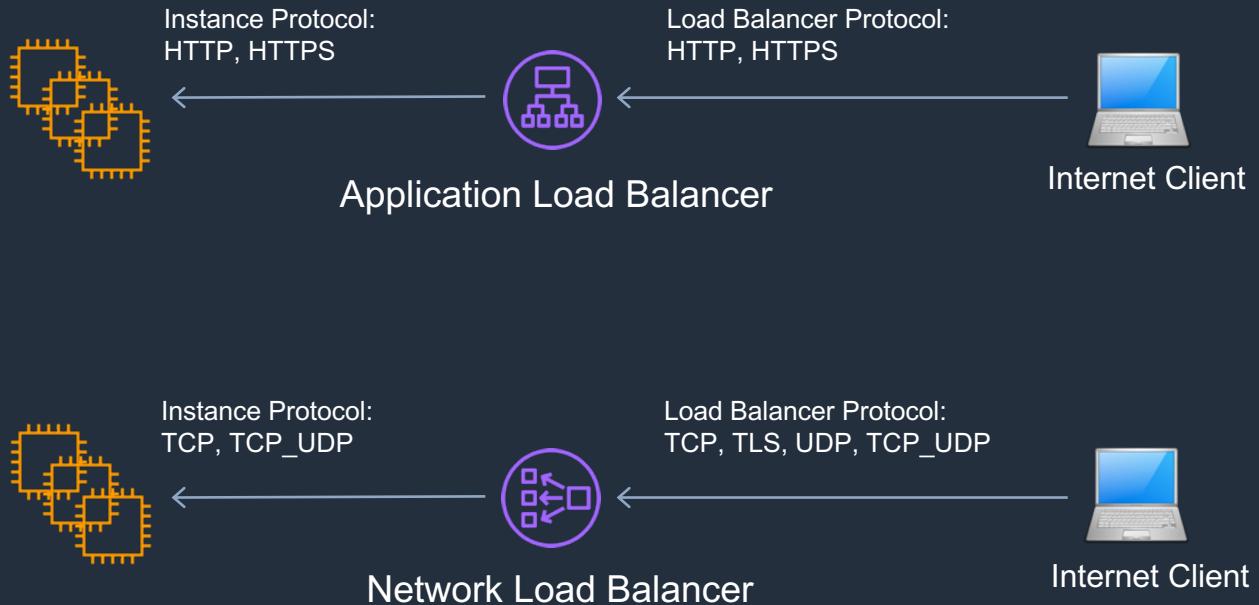
SECTION 4

Scaling Compute: Elastic Load Balancing and Auto Scaling

Elastic Load Balancing (ELB) Concepts



Elastic Load Balancing (ELB) Types



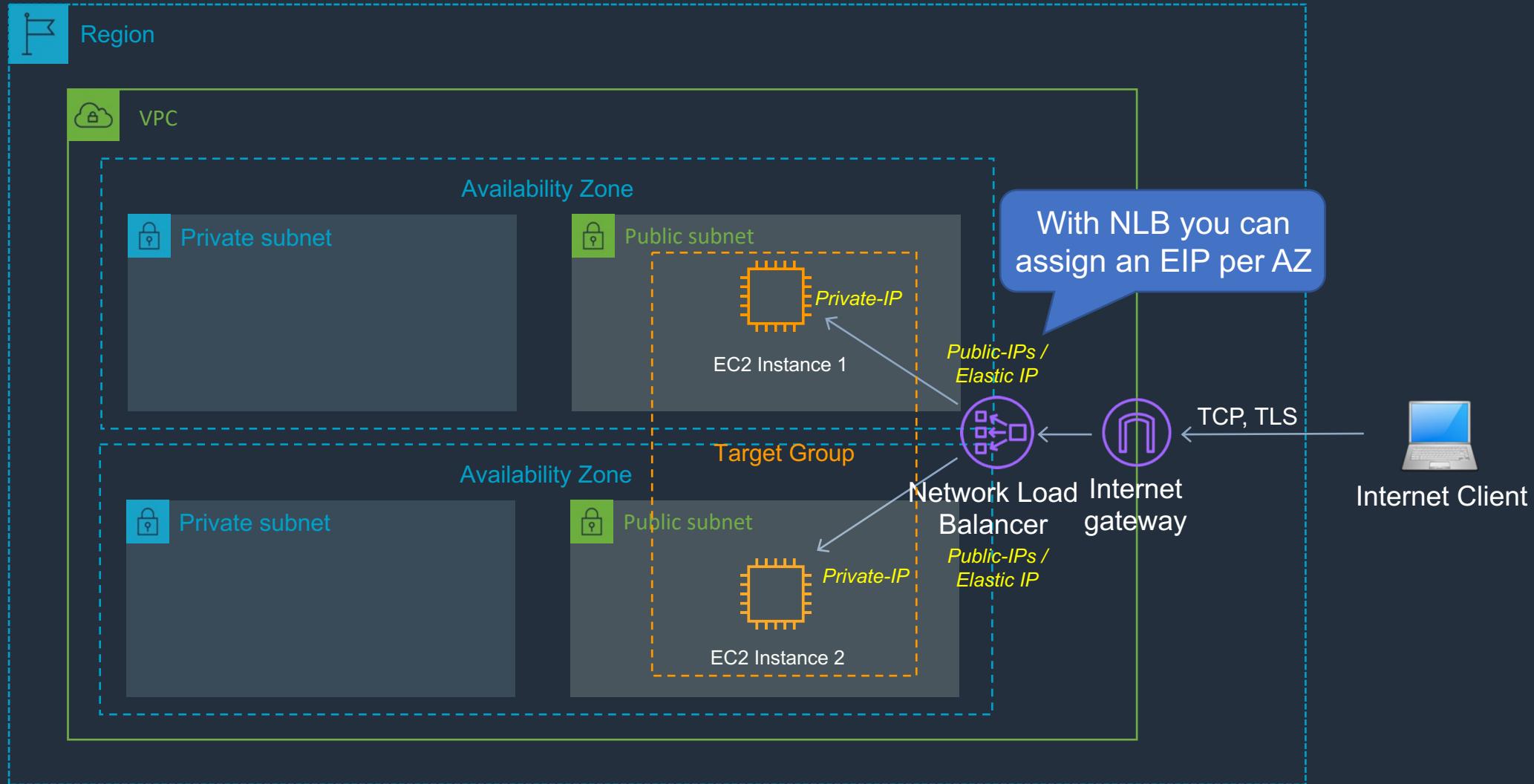
Application Load Balancer

- Operates at the request level
- Routes based on the content of the request (layer 7)
- Supports path-based routing, host-based routing, query string parameter-based routing, and source IP address-based routing
- Supports instances, IP addresses, Lambda functions and containers as targets

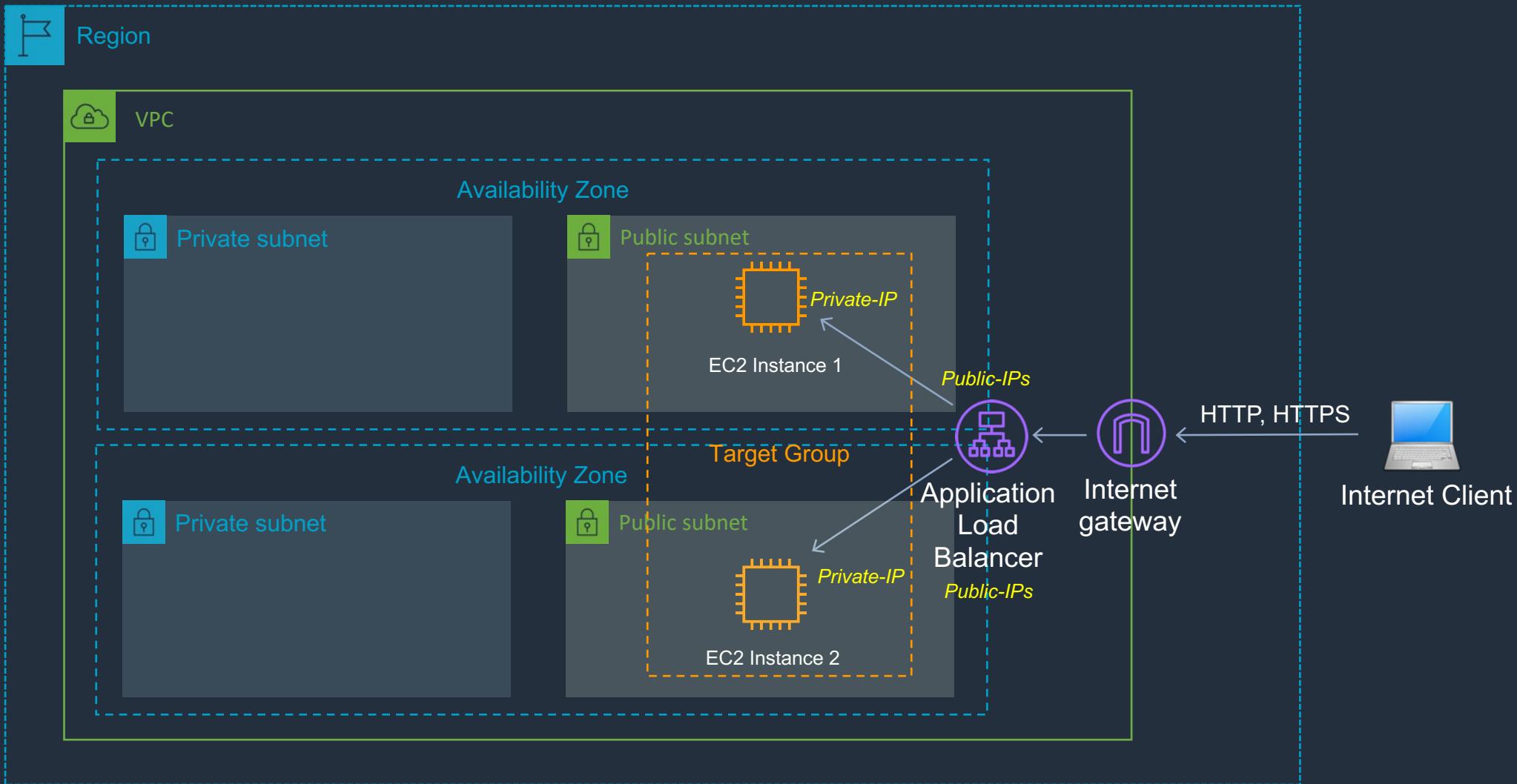
Network Load Balancer

- Operates at the connection level
- Routes connections based on IP protocol data (layer 4)
- Offers ultra high performance, low latency and TLS offloading at scale
- Can have a static IP / Elastic IP
- Supports UDP and static IP addresses as targets

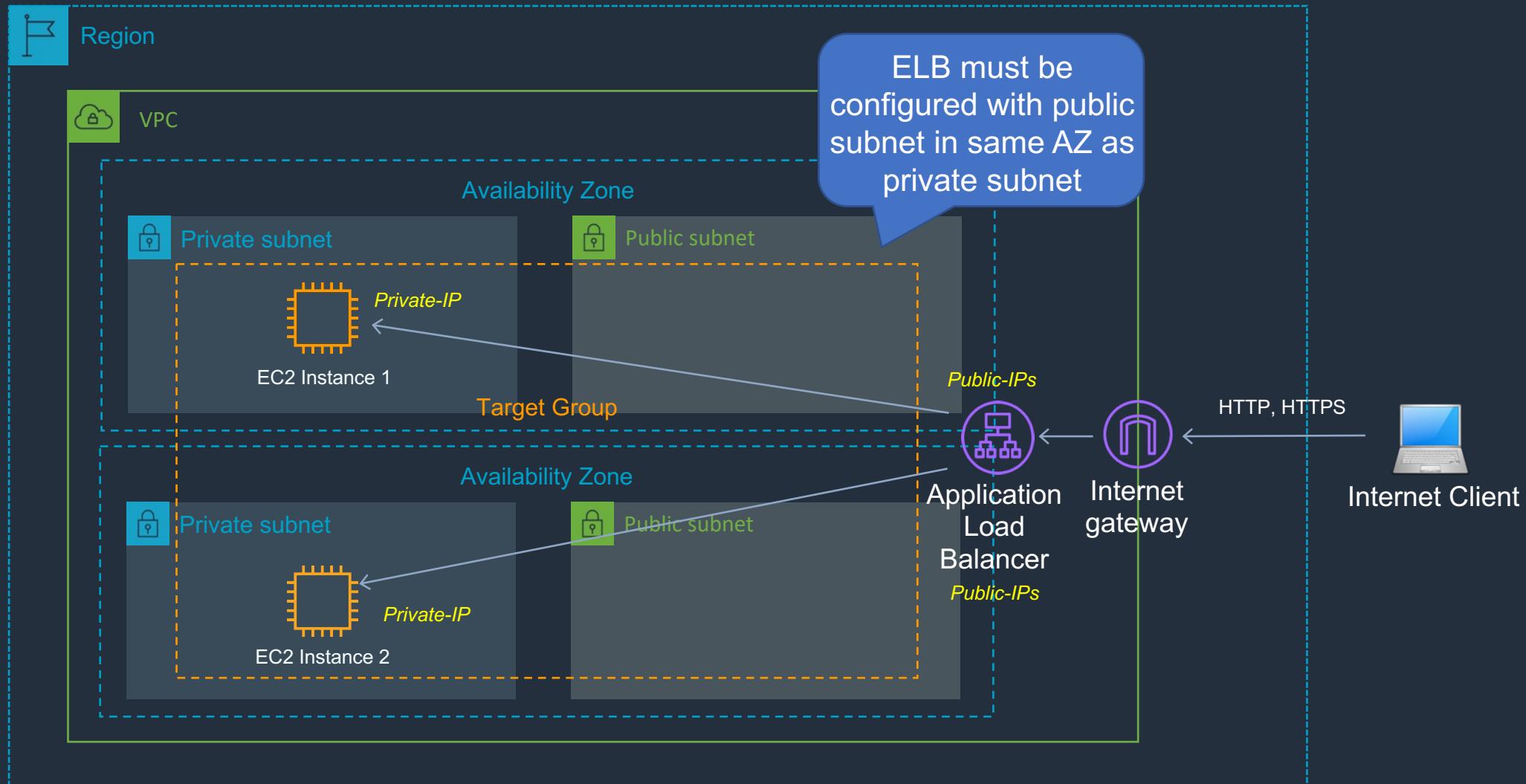
Network Load Balancer (Internet-Facing)



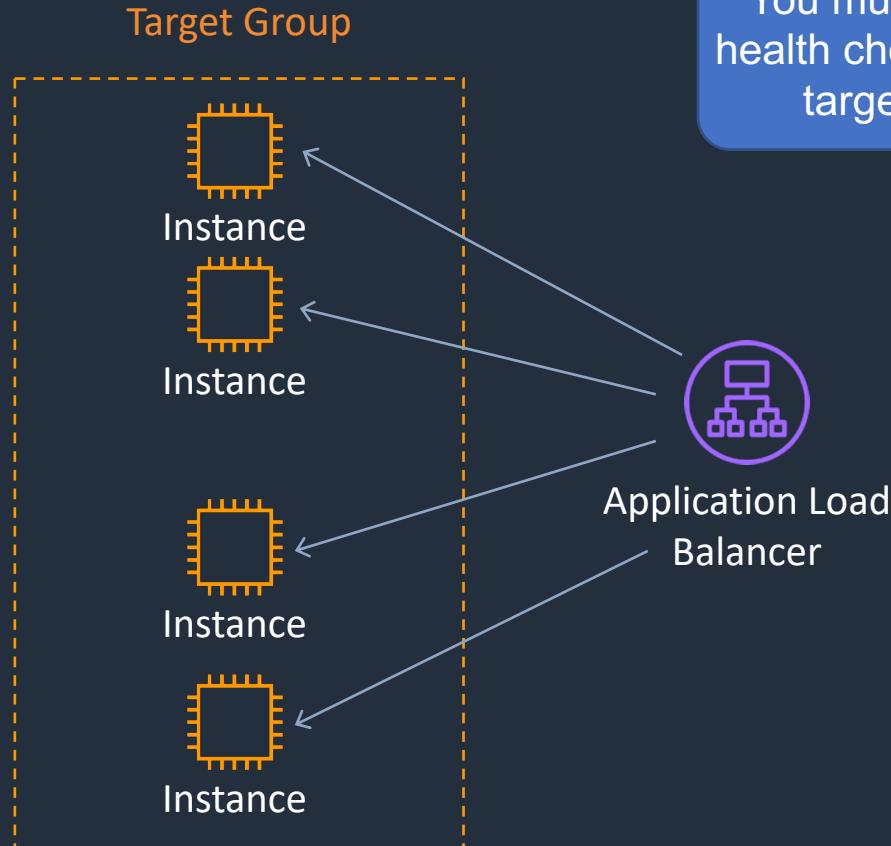
Application Load Balancer (Internet-Facing)



Application Load Balancer with Targets in Private Subnet



ELB Health Checks



Health Check Config

Protocol HTTP HTTPS Path /

Advanced health check settings

Port	<input checked="" type="radio"/> traffic port <input type="radio"/> override
Healthy threshold	5
Unhealthy threshold	2
Timeout	5 seconds
Interval	30 seconds
Success codes	200

Protocol is
HTTP/HTTPS for ALB
and can also be TCP
for NLB

ELB Health Checks

- Each load balancer node checks the health of each target, using the health check settings for the target groups with which the target is registered
- Each load balancer node routes requests only to the healthy targets in the enabled Availability Zones for the load balancer
- If a target group contains only unhealthy registered targets, the load balancer nodes route requests across its unhealthy targets

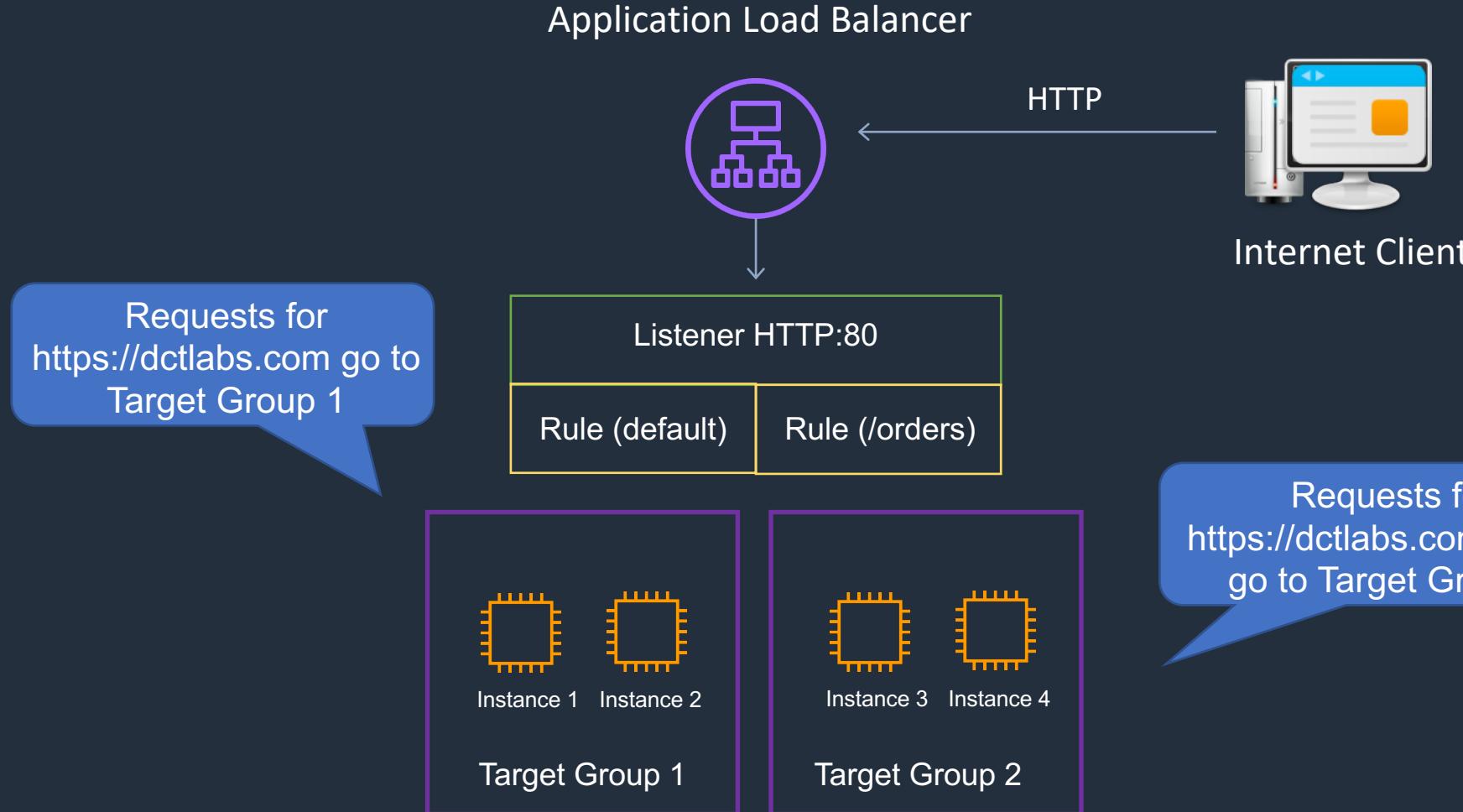
ELB Health Checks Settings

Setting	Description
HealthCheckProtocol	The protocol the load balancer uses when performing health checks on targets.
HealthCheckPort	The port the load balancer uses when performing health checks on targets.
HealthCheckPath	The ping path that is the destination on the targets for health checks. Specify a valid URI (/path?query). The default is /.
HealthCheckTimeoutSeconds	The amount of time, in seconds, during which no response from a target means a failed health check.
HealthCheckIntervalSeconds	The approximate amount of time, in seconds, between health checks of an individual target.
HealthyThresholdCount	The number of consecutive successful health checks required before considering an unhealthy target healthy.
UnhealthyThresholdCount	The number of consecutive failed health checks required before considering a target unhealthy.
Matcher	The HTTP codes to use when checking for a successful response from a target. The possible values are from 200 to 499.

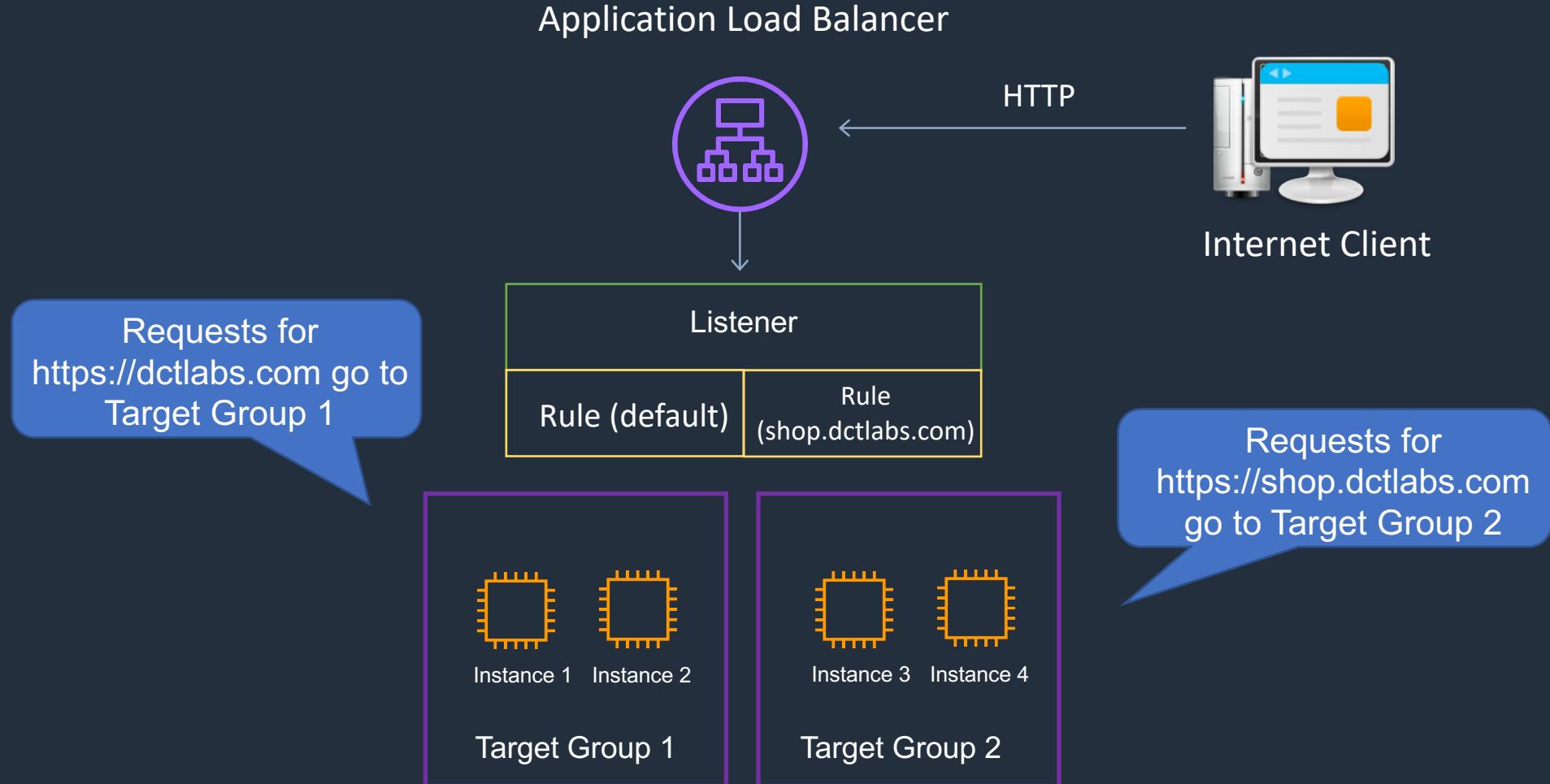
ELB Health Checks – Status Checks

Value	Description
initial	The load balancer is in the process of registering the target or performing the initial health checks on the target
healthy	The target is healthy
unhealthy	The target did not respond to a health check or failed the health check
unused	The target is not registered with a target group, the target group is not used in a listener rule, the target is in an Availability Zone that is not enabled, or the target is in the stopped or terminated state
draining	The target is deregistering and connection draining is in process
unavailable	Health checks are disabled for the target group

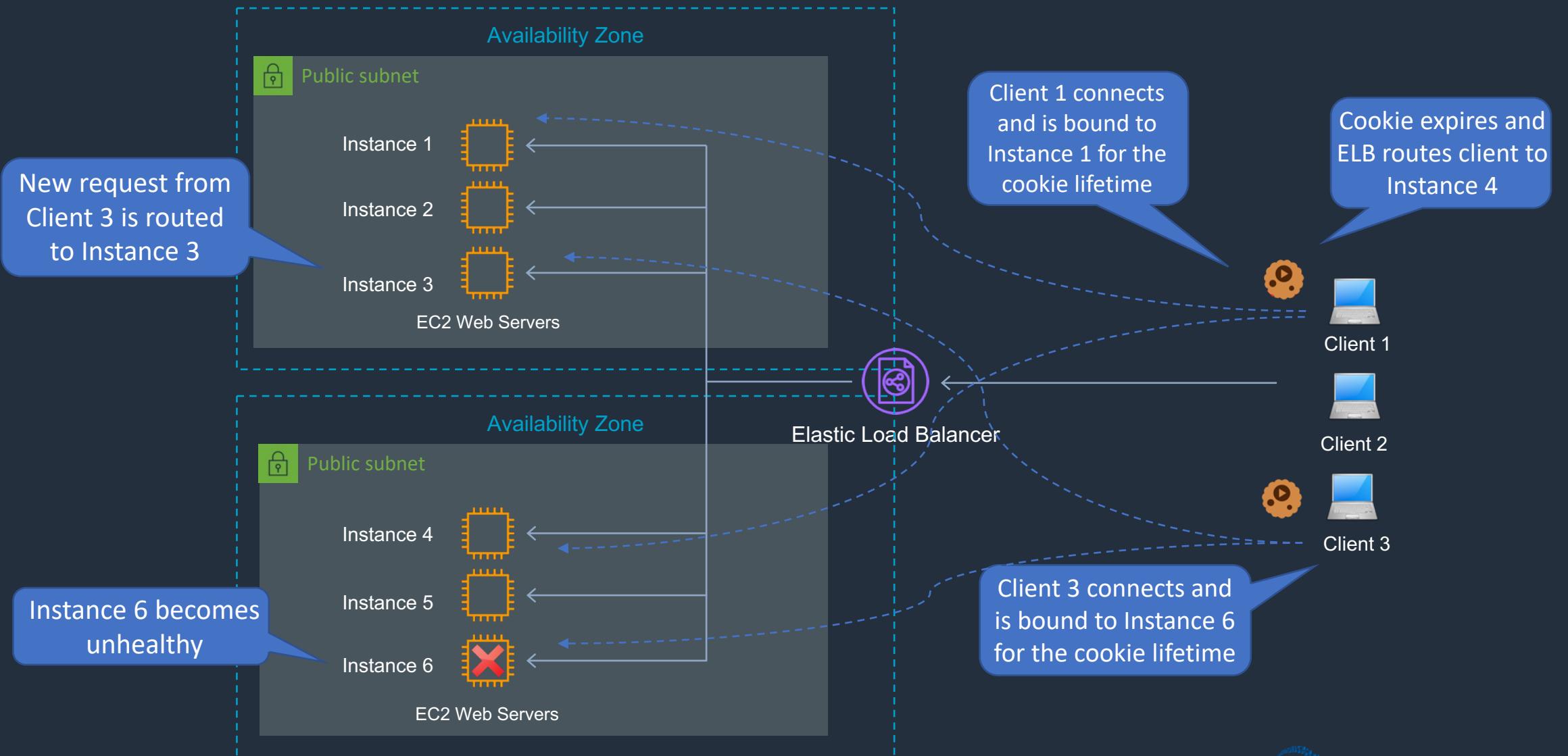
Application Load Balancer – Path-based Routing



Application Load Balancer – Host-based Routing



ELB Sticky Sessions



Sticky Sessions

Name	Supported?	Load Balancer Generated Cookie	Application Generated Cookie
ALB	Yes	Yes, "AWSALB"	Not supported
NLB	No	N/A	N/A

Sticky Session Configuration Options (ALB)

There are now two configuration options for sticky sessions:

- Duration-based cookies – always uses **AWSALB**
- Application-based cookies - set a custom app cookie name
- Both types are generated by the load balancer (not the application)
- For application-based cookies, cookie names have to be specified individually for each target group
- For duration-based cookies, AWSALB is the only name used across all target groups

Sticky Session Configuration Options (ALB)

Load balancing algorithm

Determines how the load balancer selects targets from this target group when routing requests.

Round robin

Least outstanding requests

Cannot be combined with the **Slow start duration** attribute.

Stickiness

The type of stickiness associated with this target group. If enabled, the load balancer binds a client's session to a specific instance within the target group.

Stickiness type

Load balancer generated cookie

Application-based cookie

Stickiness duration

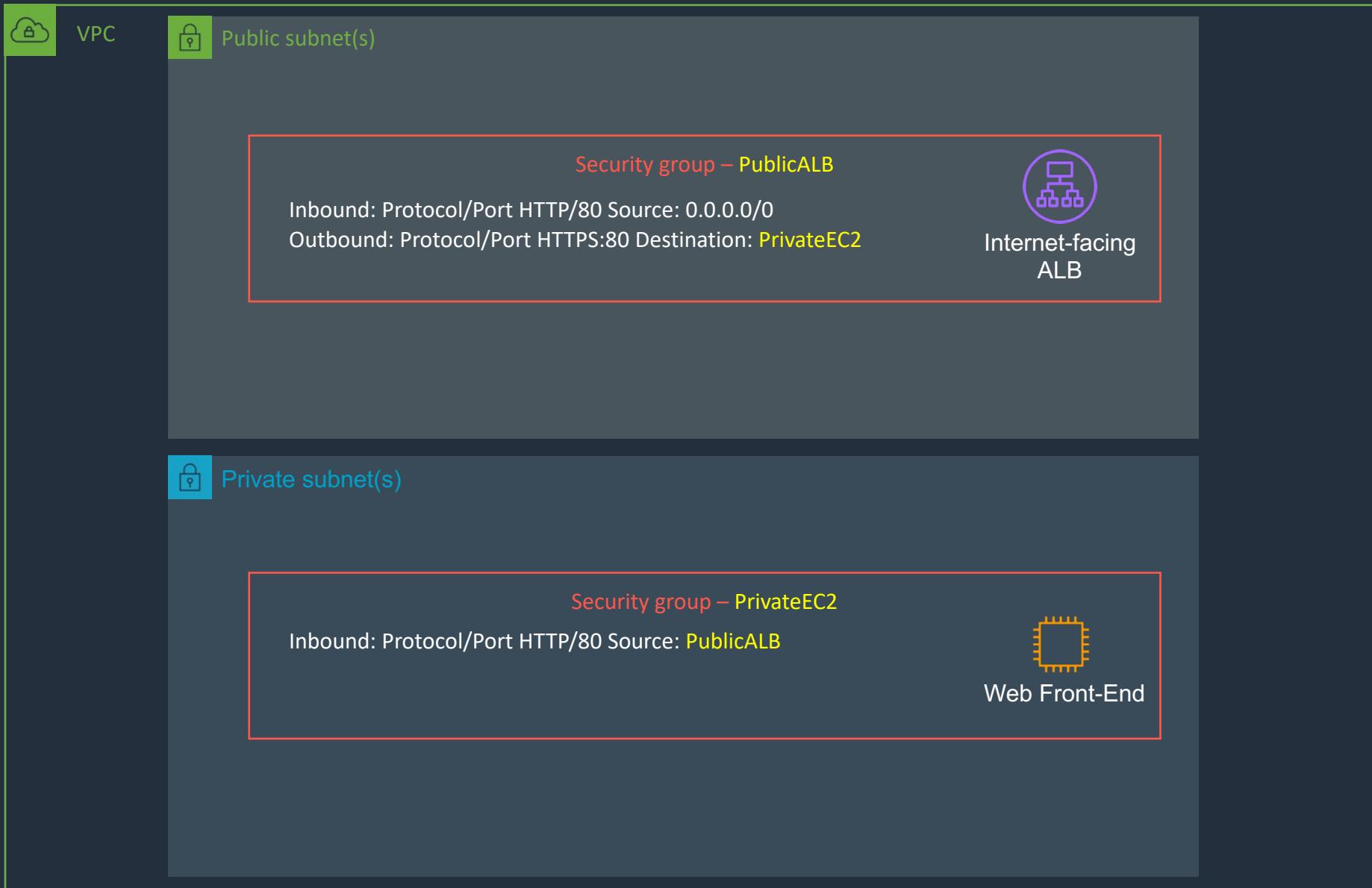
1

days

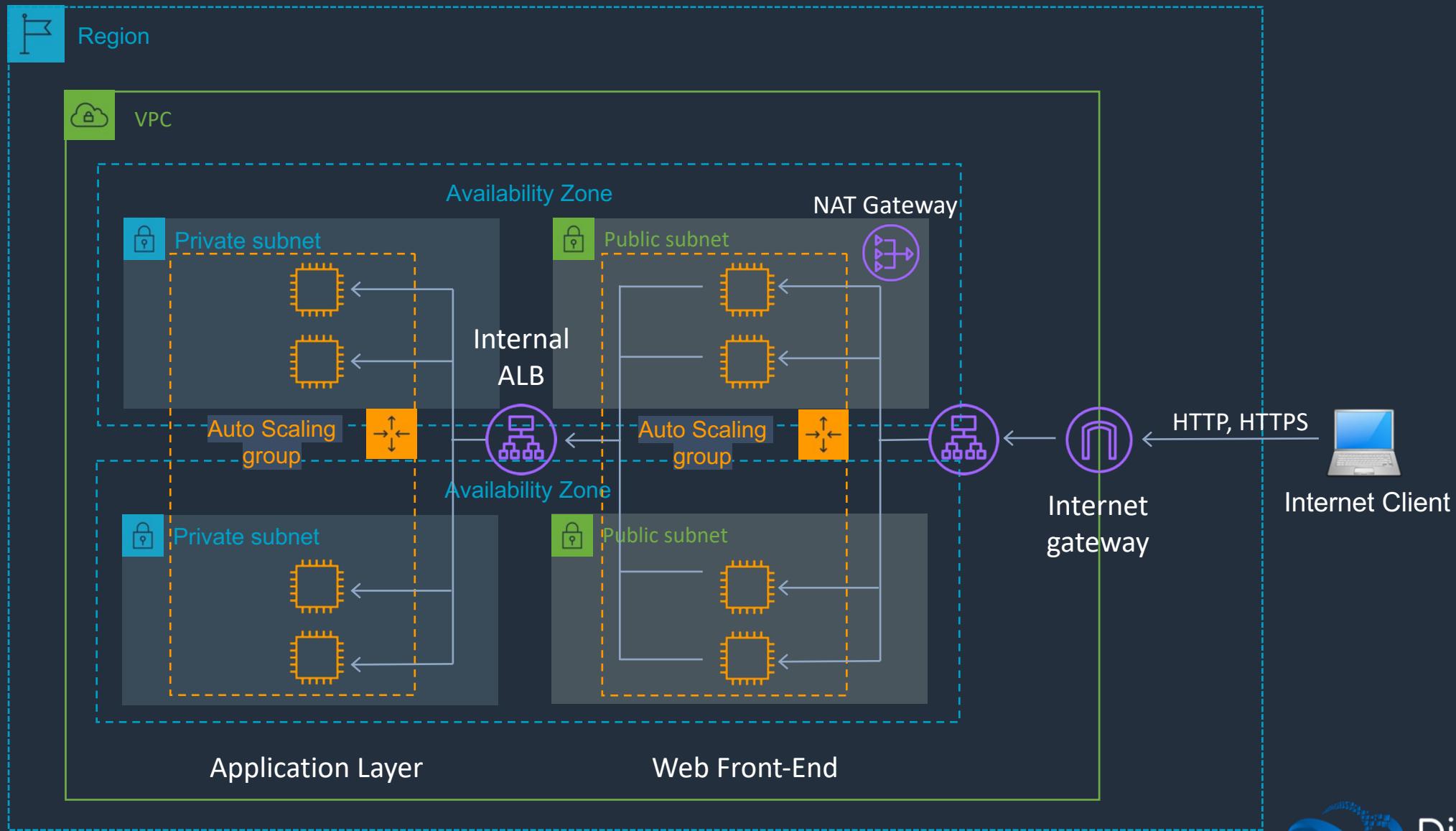


1 second - 7 days

Public ALB with Private Instances – Security Groups



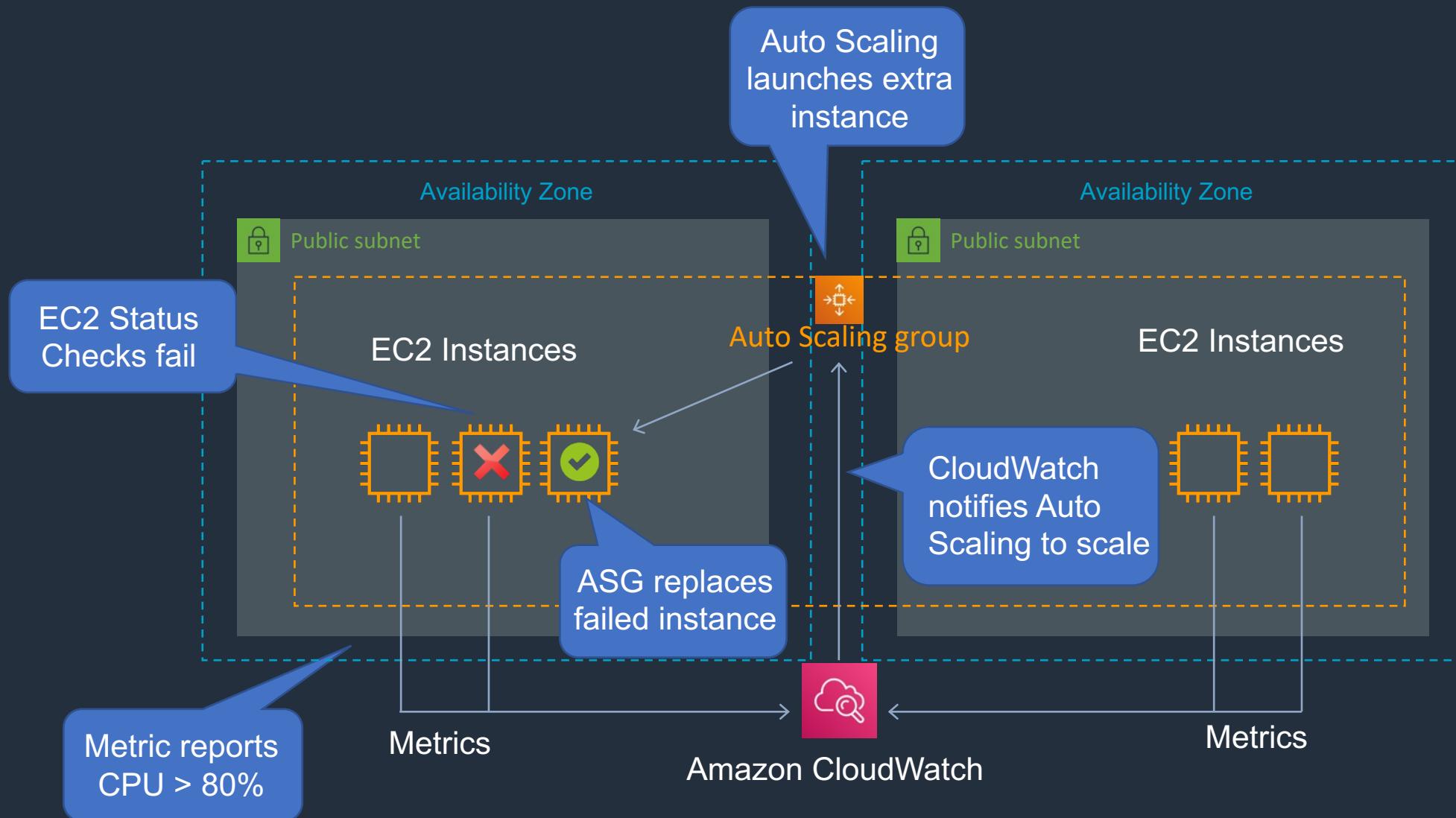
Multi-Tier Web Architecture



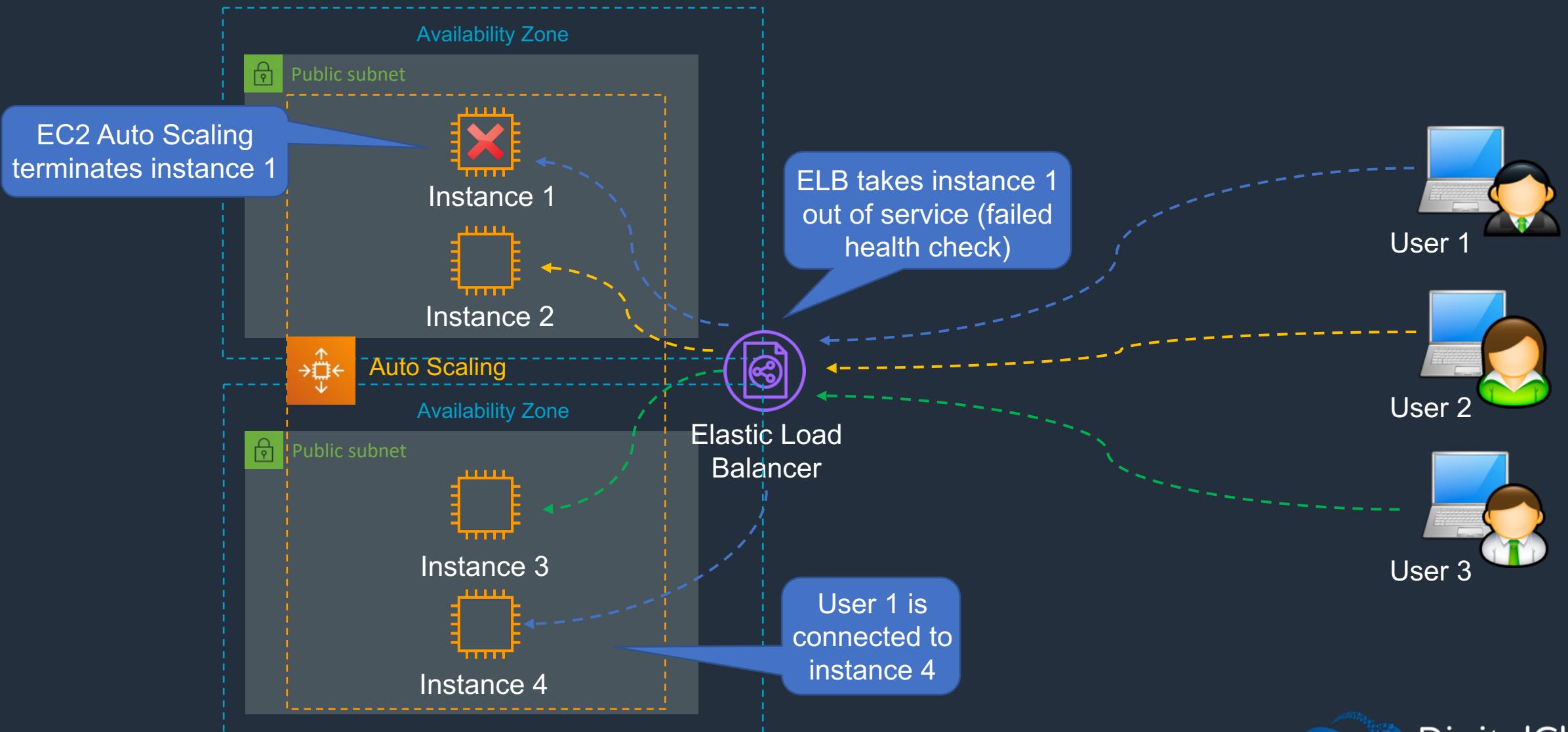
Multi-Tier Web Architecture – Security Groups



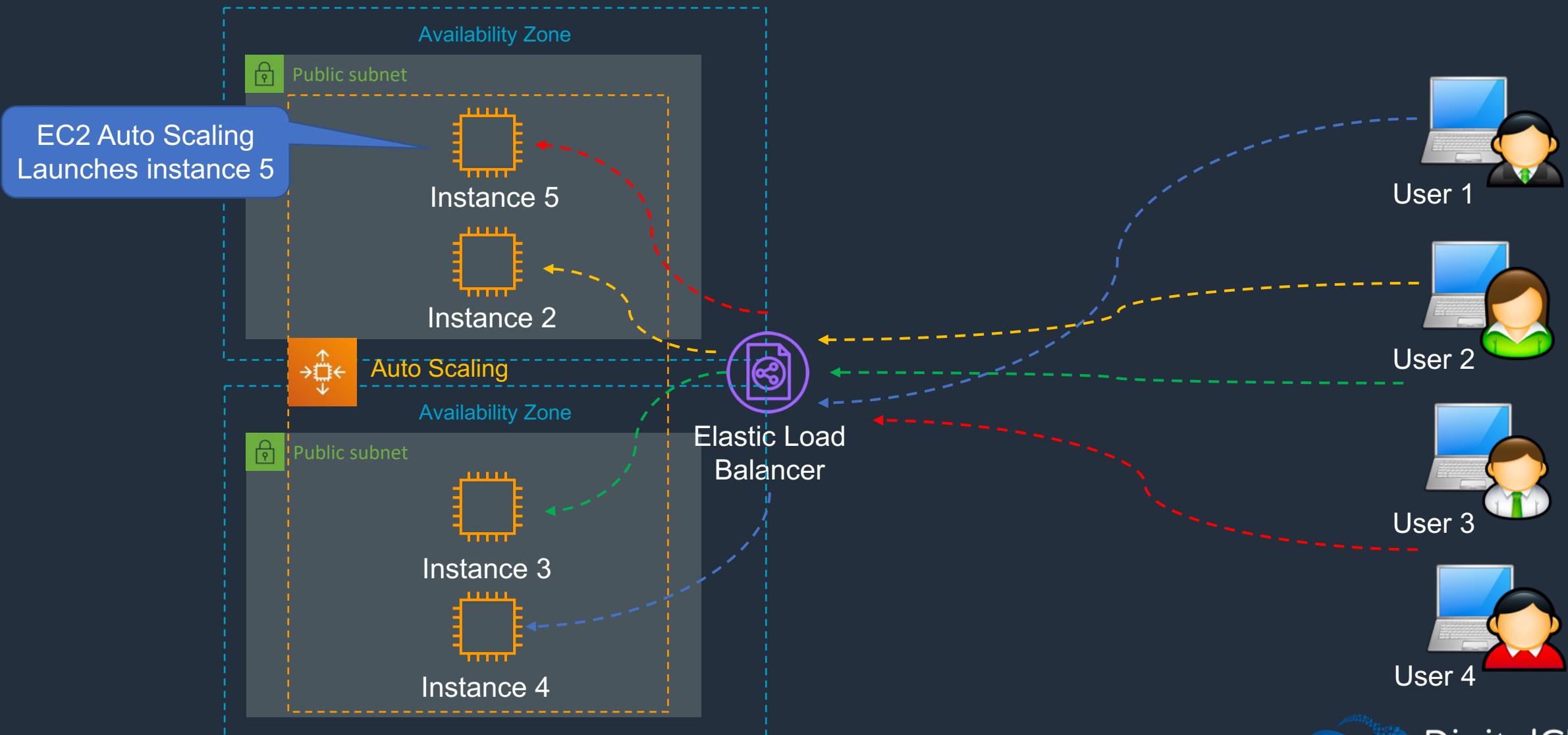
Amazon EC2 Auto Scaling



Amazon Elastic Load Balancing with EC2 Auto Scaling



Amazon Elastic Load Balancing with EC2 Auto Scaling



This architecture includes high availability and fault tolerance

EC2 Auto Scaling – Launch Configuration

Create Launch Configuration

AMI Details

 **Amazon Linux 2 AMI (HVM), SSD Volume Type - ami-0ded330691a314693**
Free tier eligible

Amazon Linux 2 comes with five years support. It provides Linux kernel 4.14 tuned for optimal performance on Amazon EC2, systemd 219, GCC 7.3, Glibc 2.26, Binutils 2.29.1, and the latest software packages through extras.
Root device type: ebs Virtualization Type: hvm

Instance Type

Instance Type	ECUs	vCPUs	Memory GiB	Instance Storage (GiB) GiB	EBS-Optimized Available	Network Performance
t2.micro	Variable	1	1	EBS only	-	Low to Moderate

Launch configuration details

Name LC4
Purchasing option On demand
EBS Optimized No
Monitoring No
IAM role None
Tenancy Shared tenancy (multi-tenant hardware)
Kernel ID Use default
RAM Disk ID Use default
User data
IP Address Type Only assign a public IP address to instances launched in the default VPC and subnet. (default)

Storage
Security Groups

Specifies the AMI and instance type

Roles, monitoring, tenancy etc.

And also storage and security groups

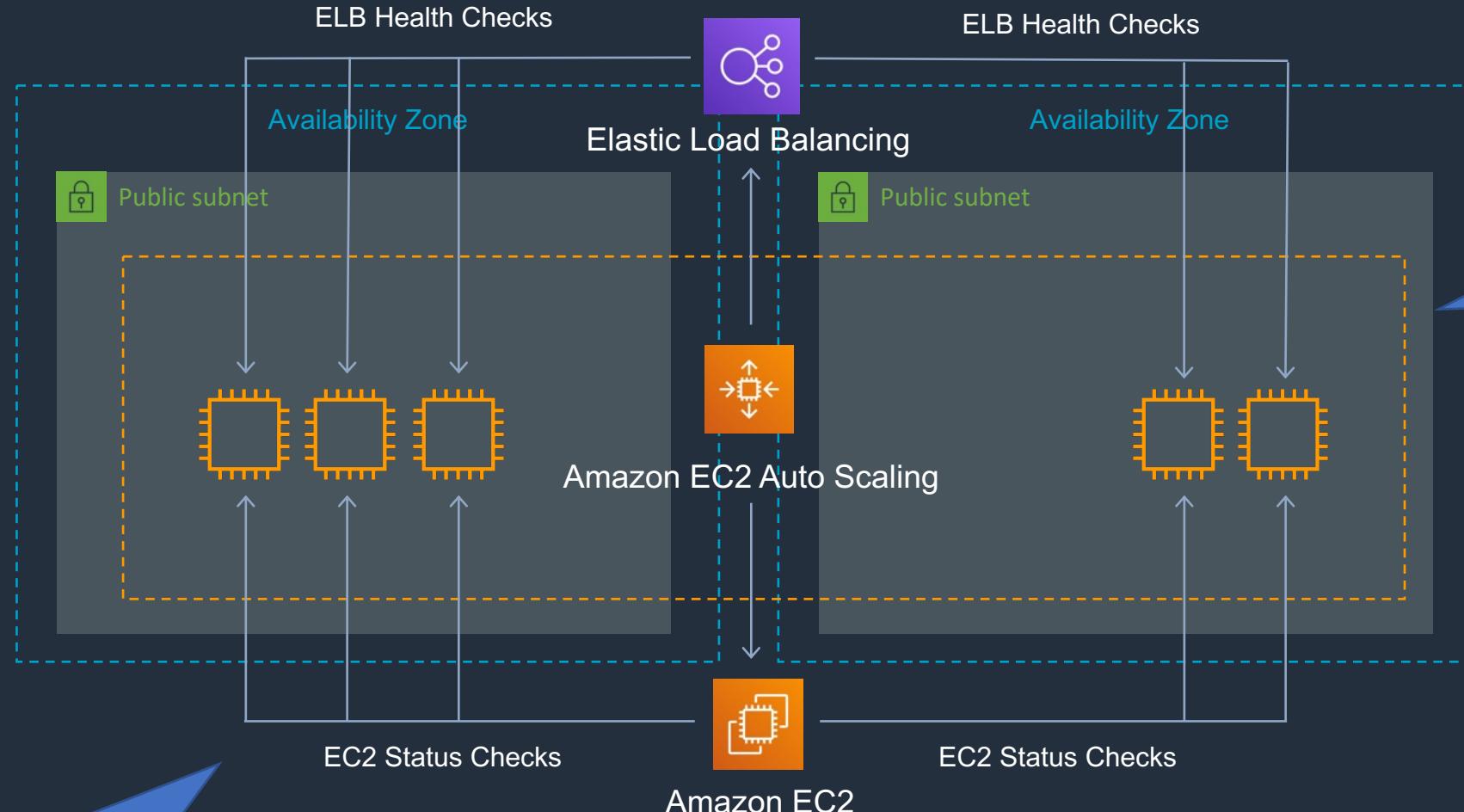
EC2 Auto Scaling – Launch Templates

Similar to a Launch Configuration but offers some additional features:

- Can have multiple versions of a template (launch configurations cannot be edited)
- Use dedicated hosts
- Use both Spot and On-demand instances
- Use multiple instance types
- Configure advanced settings such as termination protection, shutdown behavior, placement groups etc.
- And more.. We'll see in the console

EC2 and ELB Health Checks

ELB Health Checks are an optional (recommended) setting in ASG



By default, ASG uses EC2 Status Checks

If ELB Health Checks are enabled in ASG, both types are used

EC2 Auto Scaling – Types of Scaling

Scheduled Scaling

- Scaling based on a schedule allows you to scale your application ahead of known load changes
- For example, every week the traffic to your web application starts to increase on Wednesday, remains high on Thursday, and starts to decrease on Friday
- You can plan your scaling activities based on the known traffic patterns of your web application



EC2 Auto Scaling – Types of Scaling

Dynamic Scaling

- Amazon EC2 Auto Scaling enables you to follow the demand curve for your applications closely, reducing the need to manually provision Amazon EC2 capacity in advance
- Amazon EC2 Auto Scaling will then automatically adjust the number of EC2 instances as needed to maintain your target



EC2 Auto Scaling – Types of Scaling

Predictive Scaling

- Predictive Scaling, a feature of AWS Auto Scaling uses machine learning to schedule the right number of EC2 instances in anticipation of approaching traffic changes
- Predictive Scaling predicts future traffic, including regularly-occurring spikes, and provisions the right number of EC2 instances in advance
- Configured through AWS Auto Scaling – it's a layer on top of EC2 Auto Scaling
- Probably won't be on the exam yet



Auto Scaling Termination Policies – Default Policies

1. Determine which AZ has the most instances
2. Determine which instance to terminate so as to align the remaining instances to the allocation strategy for the On-Demand or Spot Instance that is terminating and your current selection of instance types
3. Determine whether any of the instances use the oldest launch template
4. Determine whether any of the instances use the oldest launch configuration
5. After applying all of the criteria in 2 through 4, if there are multiple unprotected instances to terminate, determine which instances are closest to the next billing hour



EC2 Auto Scaling – Types of Scaling

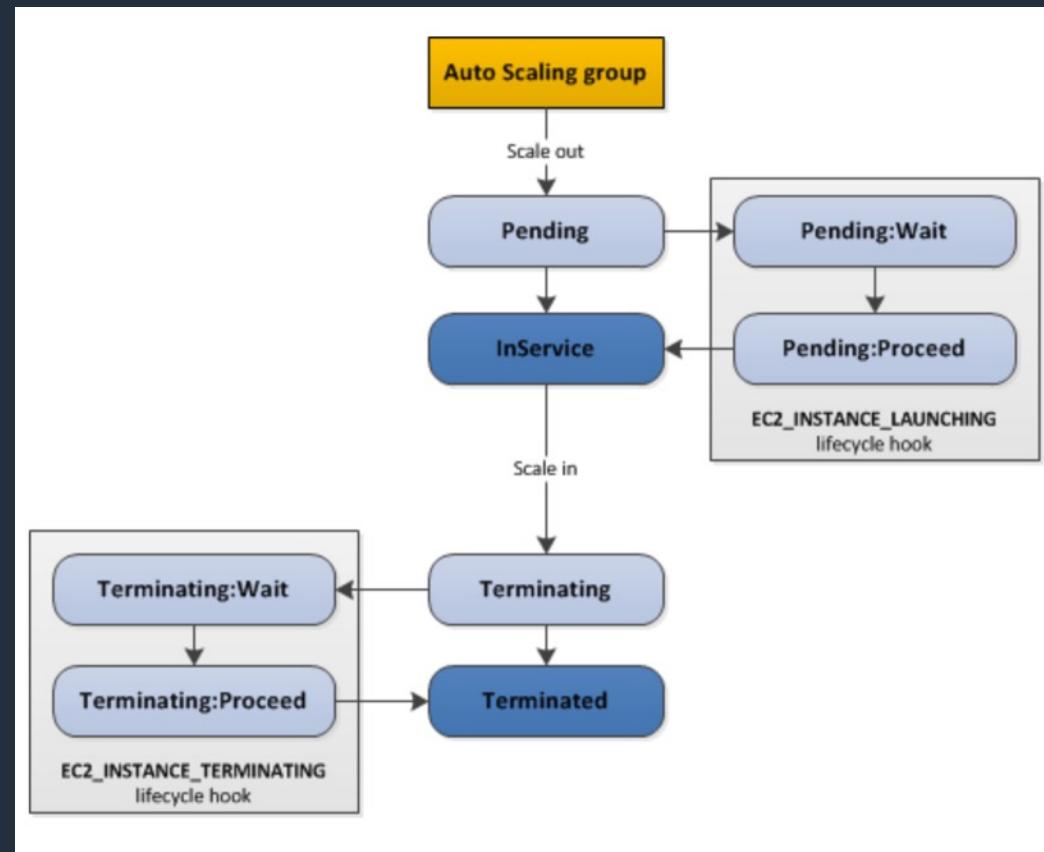
Scaling	What it is	When to use
Maintain	Ensures the required number of instances are running	Use when you always need a known number of instances running at all times
Manual	Manually change desired capacity via the console or CLI	Use when your needs change rarely enough that you're OK to make manual changes
Scheduled	Adjust min/max instances on specific dates/times or recurring time periods	Use when you know when your busy and quiet times are. Useful for ensuring enough instances are available before very busy times
Dynamic	Scale in response to system load or other triggers using metrics	Useful for changing capacity based on system utilization, e.g. CPU hits 80%

EC2 Auto Scaling – Types of Scaling

Scaling	What it is	When to use
Target Tracking Policy	The scaling policy adds or removes capacity as required to keep the metric at, or close to, the specified target value	A use case is that you want to keep the aggregate CPU usage of your ASG at 70%
Simple Scaling Policy	Waits until health check and cool down period expires before re-evaluating	This is a more conservative way to add/remove instances. Useful when load is erratic. AWS recommend step scaling instead of simple in most cases
Step Scaling Policy	Increase or decrease the current capacity of your Auto Scaling group based on a set of scaling adjustments, known as step adjustments	Useful when you want to vary adjustments based on the size of the alarm breach

Auto Scaling Lifecycle Hooks

A lifecycle hook puts a launching or terminating instance into a **Pending:Wait** or **Terminating:Wait** state



Auto Scaling Lifecycle Hooks

- You can perform a custom action using one or more of the following options:
 - Define an EventBridge target to invoke a Lambda function when a lifecycle action occurs
 - Define a notification target for the lifecycle hook.
 - Create a script that runs on the instance as the instance starts

Lifecycle Hook Name	PauseTermination
Auto Scaling Group	ASG1
Lifecycle Transition	Instance Terminate
Heartbeat Timeout	3600 seconds
Default Result	ABANDON
Notification Metadata	

Elastic Load Balancing – Monitoring and Logging

- CloudWatch – every 1 minute:
 - ELB service only sends information when requests are active

Elastic Load Balancing – Monitoring and Logging

- Some of the key metrics reported for load balancers are:
- BackendConnectionErrors
- HealthyHostCount / UnhealthyHostCount
- HTTPCode_Backend_2XX - Successful request
- HTTPCode_Backend_3XX - Redirected request
- HTTPCode_ELB_4XX client error
- HTTPCode_ELB_5XX server error (generated by ELB)
- Latency
- RequestCount
- SurgeQueueLength - the total number of requests (HTTP listener) or connections (TCP listener) that are pending routing to a healthy instance
- SpilloverCount - the total number of requests that were rejected because the surge queue is full

Elastic Load Balancing – Monitoring and Logging

- Access Logs:
 - Disabled by default
 - Includes information about the clients (not included in CloudWatch metrics):
 - Time
 - Client IP address
 - Latencies
 - Request paths
 - Server response
 - Trace ID
 - Can be optionally stored and retained in S3

Elastic Load Balancing – Monitoring and Logging

- CloudTrail:
 - Can be used to capture API calls to the ELB
 - Logs can be stored in an S3 bucket

Auto Scaling – Monitoring

EC2 Auto Scaling – Monitoring

- Basic monitoring sends EC2 metrics to CloudWatch about ASG instances every 5 minutes
- Detailed can be enabled and sends metrics every 1 minute (chargeable)
- When the launch configuration is created from the console basic monitoring of EC2 instances is enabled by default
- When the launch configuration is created from the CLI detailed monitoring of EC2 instances is enabled by default

Exam Scenarios

Exam Scenario	Solution
Design required for highly available and secure website on EC2 with ALB, and DB on EC2	Launch ALB in public subnets, web servers in private subnets and DB layer in private subnets – all layers across AZs
HealthyHostCount metrics for an ALB have dropped from 6 to 2. Need to determine the cause	The health checks on target EC2 instances are failing
An instance attached to an ALB exceeded the UnhealthyThresholdCount for consecutive health check failures. What will happen?	Health checks will continue and the ALB will take the instance out of service

Exam Scenarios

Exam Scenario	Solution
Requirement to track the source IP of clients and the instance that processes the request	Check the ALB access logs for this information
Requirement to trigger an alarm when all instances are unhealthy	Use Amazon CloudWatch with the condition: "AWS/ApplicationELB HealthyHostCount <= 0"
Need to check why users cannot connect to web server public IP and port (behind ALB)	Check the VPC Flow Logs

Exam Scenarios

Exam Scenario	Solution
HTTPCode_ELB_5XX_Count Amazon CloudWatch metrics are noticed for an ALB	The target group may not contain any healthy instances
CloudWatch shows 4XX errors for app with ALB but the Instances have already been terminated and need to analyze the root cause	Use ELB access logs to retrieve info from S3 bucket to find the originators of the requests
Need a load balancer where specific static public IP addresses can be whitelisted by clients	Use a Network Load Balancer (NLB)

Exam Scenarios

Exam Scenario	Solution
Poor performance has been experienced for an application running on Amazon EC2	Use EC2 Auto Scaling to dynamically scale
503 and 504 errors experienced and instances have high CPU utilization	Use EC2 Auto Scaling to dynamically scale
ASG does not launch instances during busy periods despite max capacity not being reached	Could be due to service limits (check Trusted Advisor) or check for RunInstances requests in CloudTrail in case they are failing
Need to analyze instances before they are terminated	Use Auto Scaling lifecycle hooks to pause termination

Exam Scenarios

Exam Scenario	Solution
Auto Scaling scales based on queue depth but at beginning of day app slows down	Create a scheduled scaling policy
Create highly available EC2 Auto Scaling group for a single instance app	Use at least 3 AZs, min size of 2, desired capacity of 2, and max of 2
Elastic Beanstalk worker node reads messages from SQS queue. Auto Scaling scales instances. App slows down when number of messages in queue increases	Update ASG to scale on queue depth
ALB is expecting a large spike in traffic and the application is memory heavy	Use the RequestCountPerTarget metric to control scaling

Exam Scenarios

Exam Scenario	Solution
New instances in an Auto Scaling group are not showing up in the aggregated metrics. Step scaling is used	Likely due to the warm-up period having not yet expired

SECTION 5

Storage: Amazon EBS, EFS, and AWS Storage Gateway

Amazon Elastic Block Store (EBS) - Block-based Storage

Hard drives are block-based storage systems



The Operating System (OS) sees a volume. A volume can be partitioned and formatted

Amazon EBS - HDDs and SSDs



Hard Disk Drive (HDD)

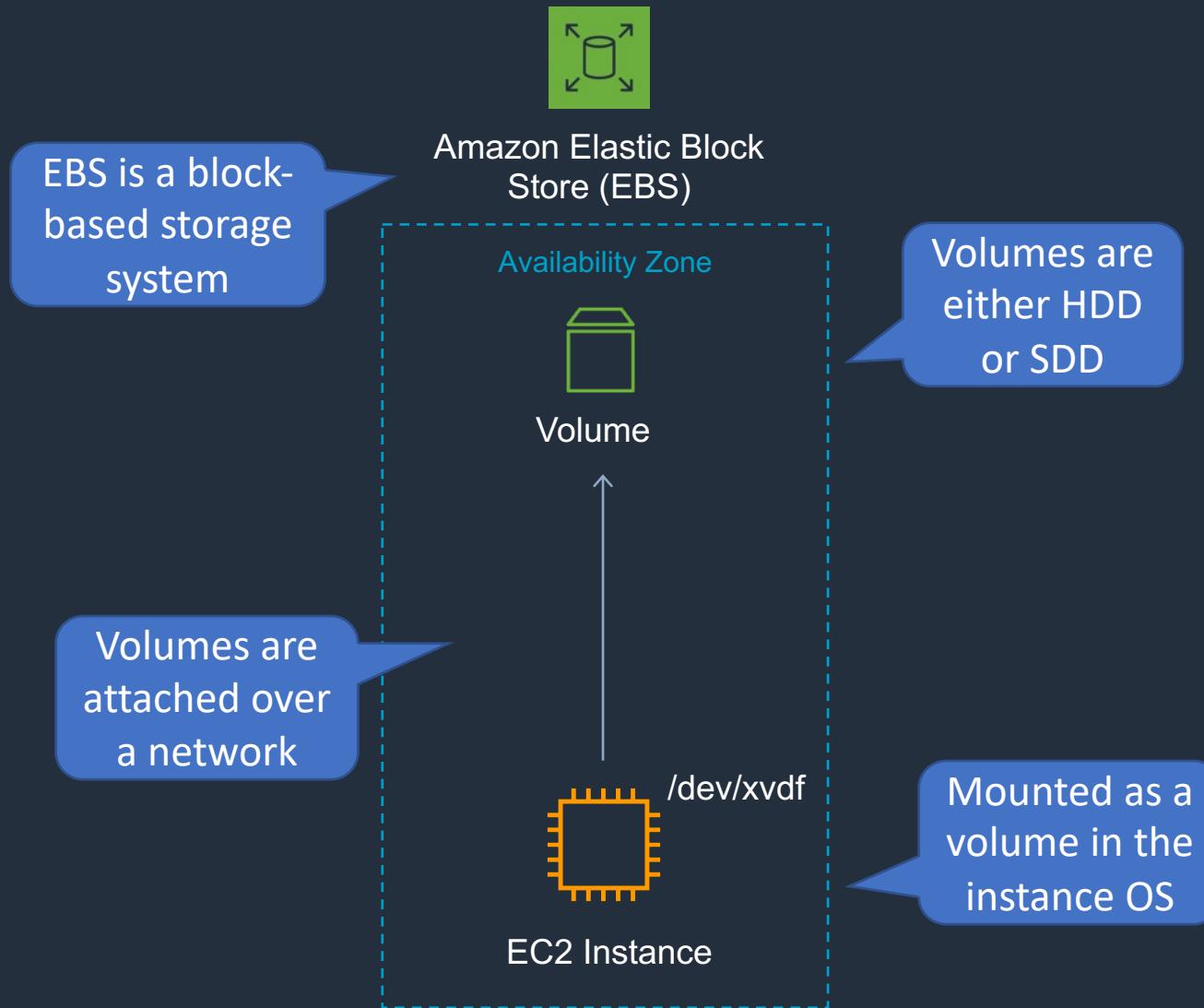
- Also known as magnetic drives
- Older technology
- Much slower than SSD
- Much cheaper than SSD



Solid State Drive (SSD)

- Uses flash memory
- Newer technology
- MUCH faster than HDD
- More expensive than HDD

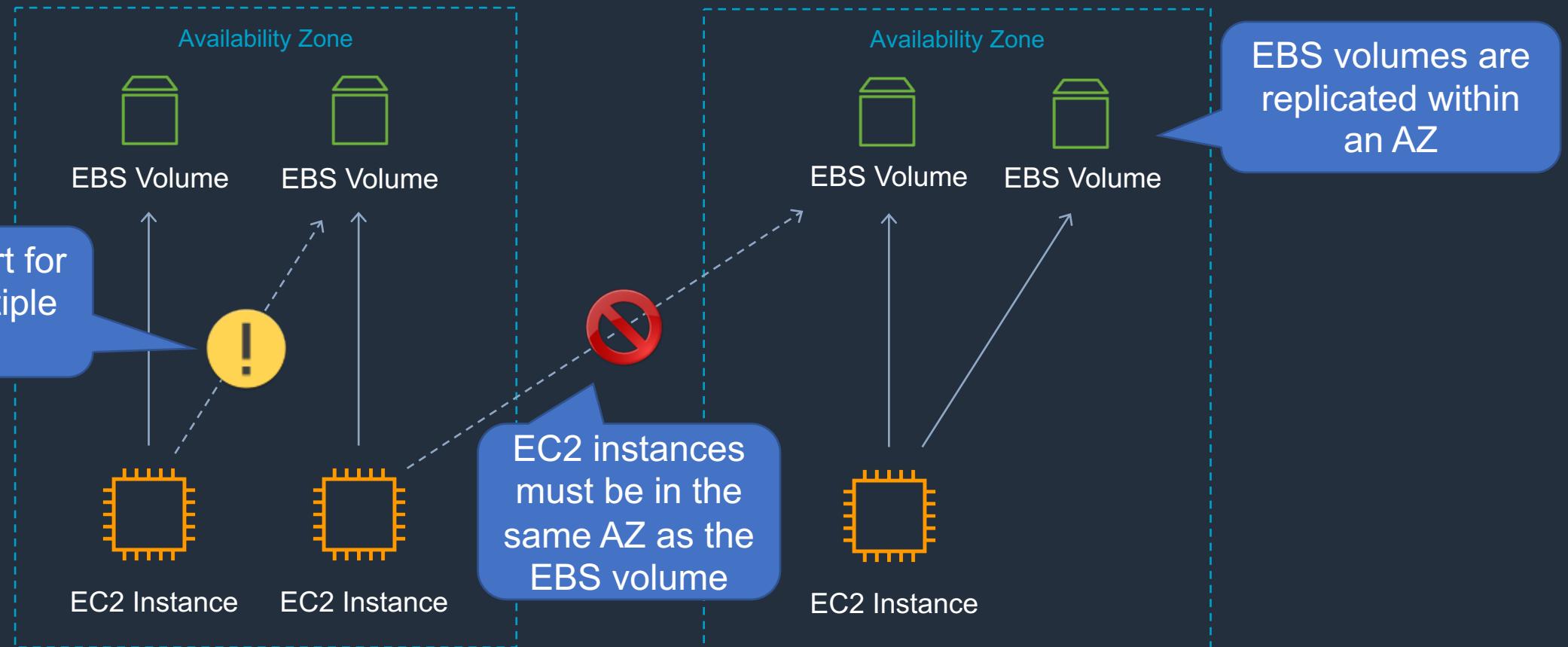
Amazon Elastic Block Store (EBS)



Amazon EBS Deployment



Amazon Elastic Block
Store (EBS)



Amazon EBS Multi-Attach



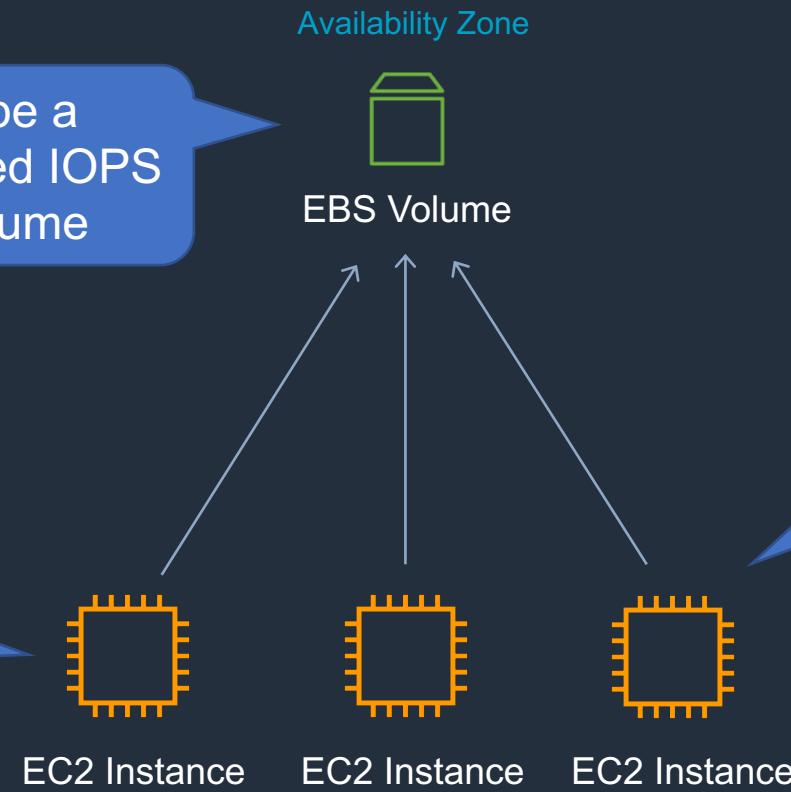
May not be on the exam yet

Available for Nitro system-based EC2 instances

Must be a Provisioned IOPS io1 volume

Must be within a single AZ

Up to 16 instances can be attached to a single volume



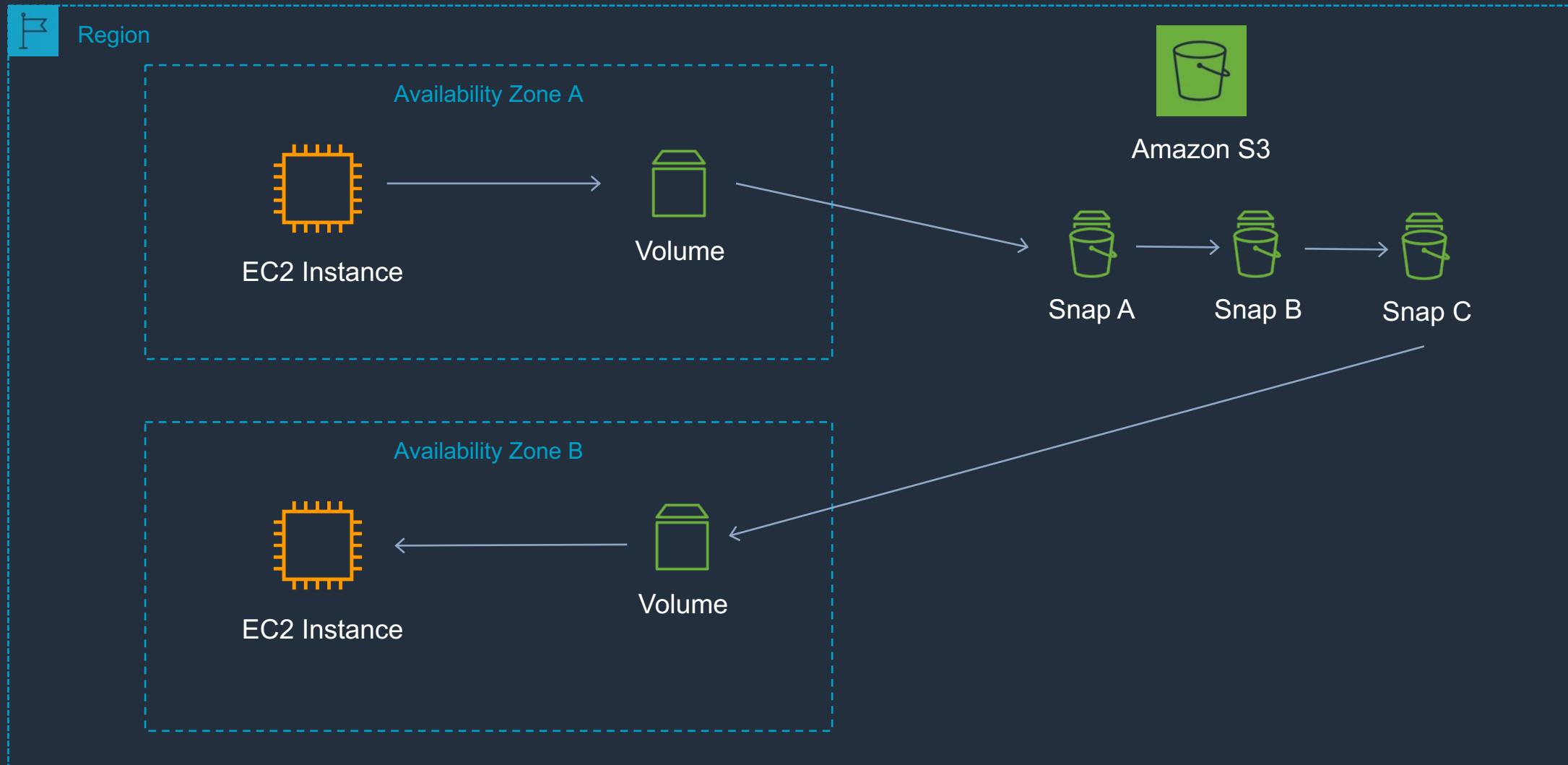
Amazon EBS

- Termination protection is turned off by default and must be manually enabled (keeps the volume/data when the instance is terminated)
- Root EBS volumes are deleted on termination by default
- Extra non-boot volumes are not deleted on termination by default
- The behaviour can be changed by altering the "DeleteOnTermination" attribute
- Volume sizes and types can be upgraded without downtime (except for magnetic standard)
- Elastic Volumes allow you to increase volume size, adjust performance, or change the volume type while the volume is in use

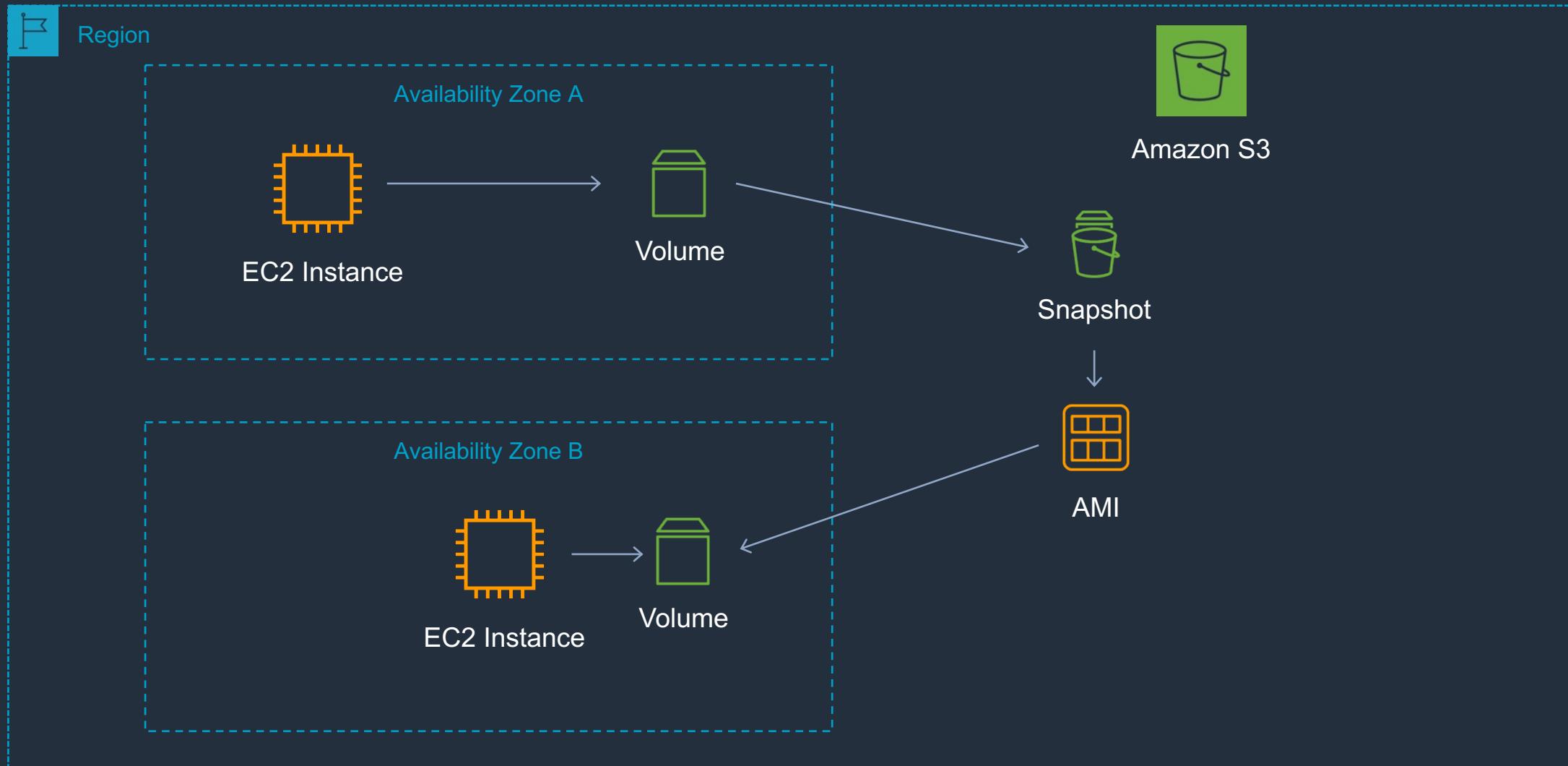
Amazon EBS Volume Types

	Solid State Drives (SSD)		Hard Disk Drives (HDD)	
Volume Type	EBS Provisioned IOPS SSD (io1)	EBS General Purpose SSD (gp2)	Throughput Optimized HDD (st1)	Cold HDD (sc1)
Short Description	Highest performance SSD volume designed for latency-sensitive transactional workloads	General Purpose SSD volume that balances price performance for a wide variety of transactional workloads	Low cost HDD volume designed for frequently accessed, throughput intensive workloads	Lowest cost HDD volume designed for less frequently accessed workloads
Use Cases	I/O-Intensive NoSQL and relational databases	Boot volumes, low-latency interactive apps, dev & test	Big data, data warehouses, log processing	Colder data requiring fewer scans per day
Volume Size	4GB – 16TB	1 GB – 16 TB	500 GB – 16 TB	500 GB – 16 TB
Max IOPS/Volume	64,000	16,000	500	250
Max Throughput/Volume	1,000 MB/s	250 MB/s	500 MB/s	250 MB/s

Amazon EBS Snapshots



Take Snapshot, Create AMI, Launch New Instance



Amazon EBS Copying, Sharing and Encryption



Volume



Snapshot

- Encryption state retained
- Same region



Unencrypted Snapshot



Encrypted Snapshot

- Can be encrypted
- Can change regions



Unencrypted Snapshot



Encrypted Volume

- Can be encrypted
- Can change AZ



Unencrypted Snapshot



AMI

- Cannot be encrypted
- Can be shared with other accounts
- Can be shared publicly



Encrypted Snapshot



Encrypted Snapshot

- Can change encryption key
- Can change regions



Encrypted Snapshot



Encrypted Volume

- Can change encryption key
- Can change AZ



Encrypted Snapshot



Encrypted AMI

- Block devices remain encrypted
- Cannot be shared with other accounts if using AWS CMK
- Cannot be shared publicly



Encrypted AMI



Encrypted AMI

- Block devices remain encrypted
- Can change regions



Encrypted AMI



EC2 Instance

- Can change encryption key
- Can change AZ



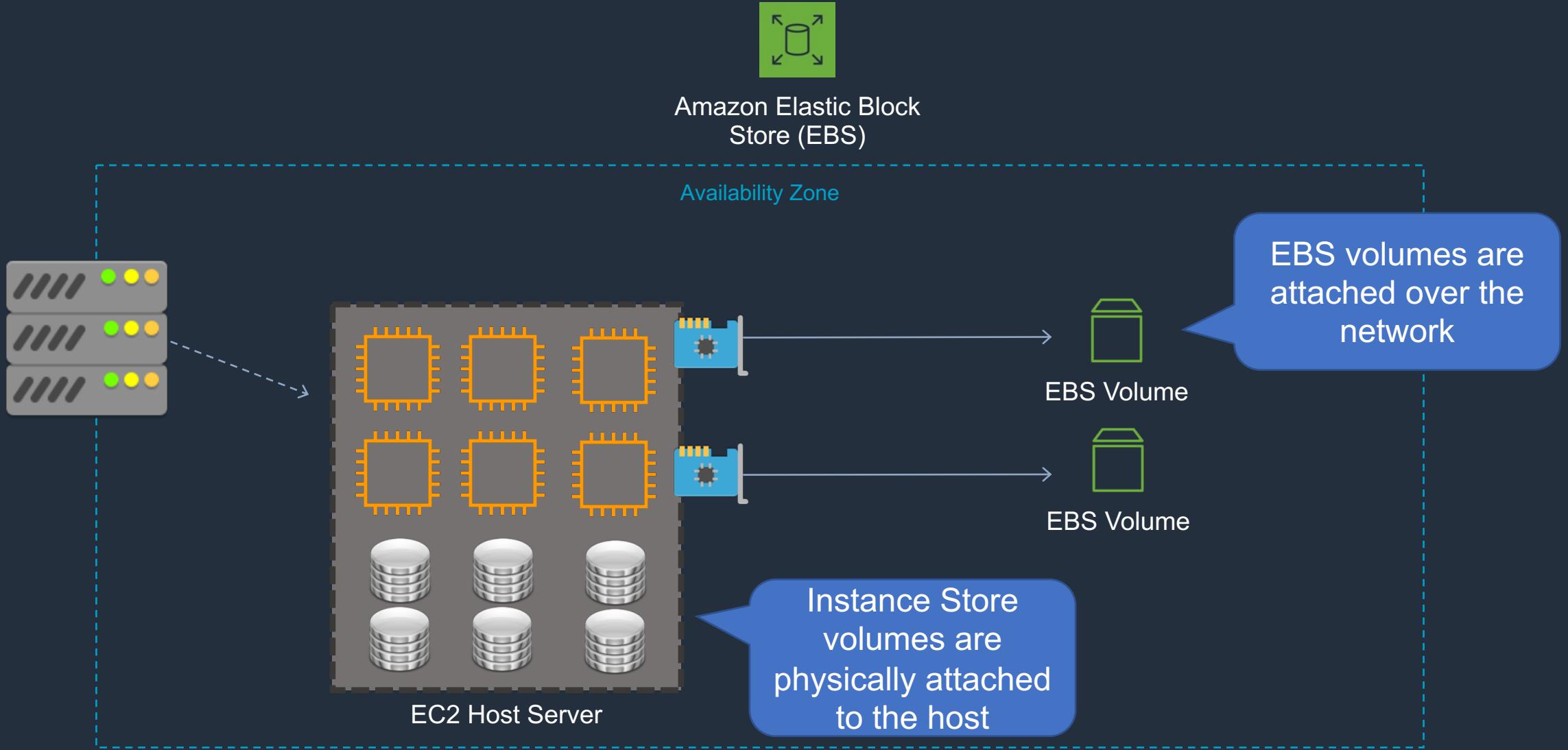
Unencrypted AMI



EC2 Instance

- Can change encryption state
- Can change AZ

Amazon EBS vs Instance Store



Amazon EBS Instance Stores

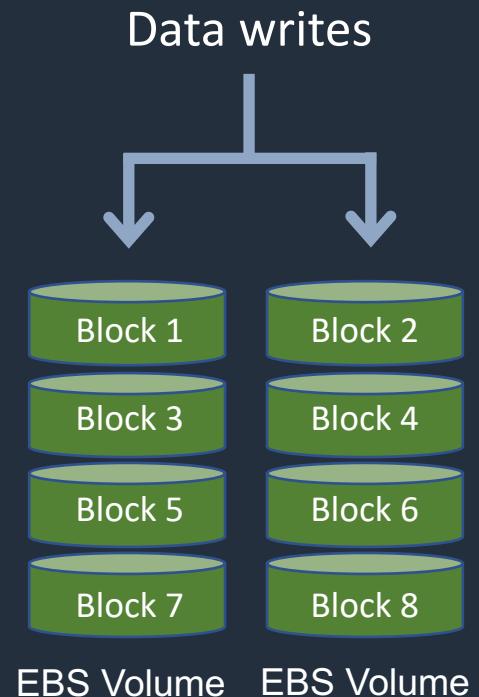
- Instance store volumes are high performance local disks that are physically attached to the host computer on which an EC2 instance runs
- Instance stores are ephemeral which means the data is lost when powered off (non-persistent)
- Instances stores are ideal for temporary storage of information that changes frequently, such as buffers, caches, or scratch data
- Instance store volume root devices are created from AMI templates stored on S3
- Instance store volumes cannot be detached/reattached

Using RAID with Amazon EBS

- RAID stands for Redundant Array of Independent disks
- Not provided by AWS, you must configure through your operating system
- RAID 0 and RAID 1 are potential options on EBS
- RAID 5 and RAID 6 are not recommended by AWS

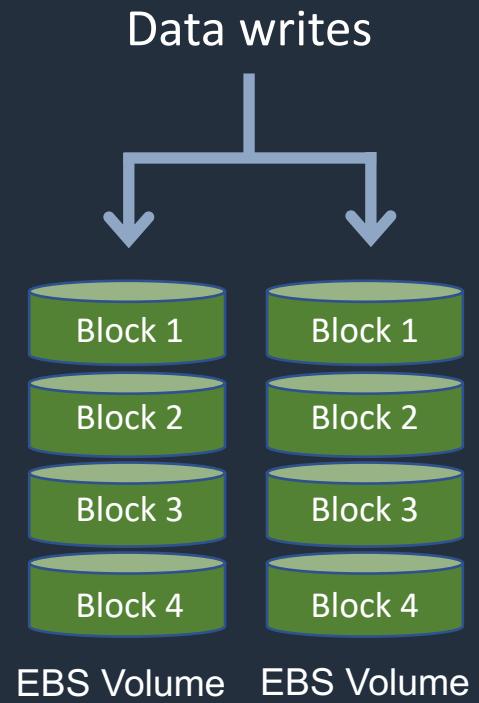
Using RAID with Amazon EBS

- RAID 0 is used for striping data across disks (performance)
 - Use 2 or more disks
 - If one disk fails, the entire RAID set fails

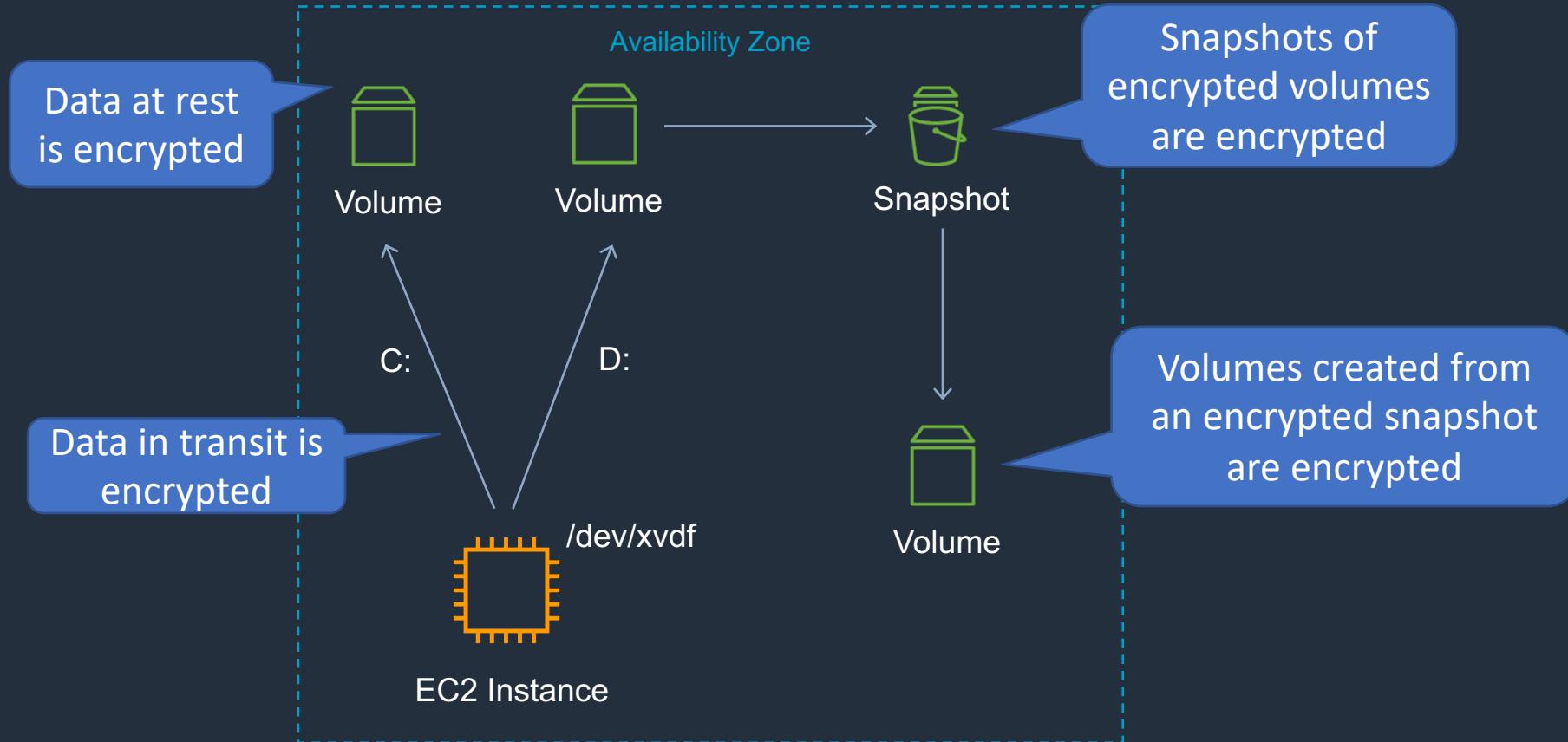


Using RAID with Amazon EBS

- RAID 1 is used for mirroring data across disks (redundancy / fault tolerance)
- If one disk fails, the other disk is still working
- Data gets sent to 2 EBS volumes at the same time



Amazon EBS Encryption



Amazon EBS Encryption

- Expect the same IOPS performance on encrypted volumes as on unencrypted volumes
- EBS encrypts your volume with a data key using the industry-standard AES-256 algorithm
- Your data key is stored on-disk with your encrypted data, but not before EBS encrypts it with your CMK. Your data key never appears on disk in plaintext
- The same data key is shared by snapshots of the volume and any subsequent volumes created from those snapshots
- You can share snapshots, but if they're encrypted it must be with a custom CMK key
- You can check the encryption status of your EBS volumes with AWS Config

CloudWatch Metrics for EBS

A few specific metrics to understand for the exam:

- **DiskReadBytes / DiskWriteBytes:**

- Relates to Instance Store volumes NOT to EBS
- Included in the **AWS/EC2** namespace

- **VolumeReadBytes / VolumeWriteBytes:**

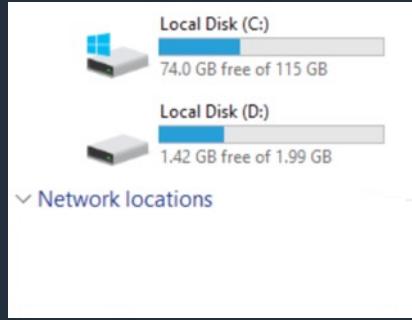
- Relates to the EBS volume
- Included in the **AWS/EBS** namespace

Amazon Data Lifecycle Manager

Automate the creation, retention, and deletion of EBS snapshots and EBS-backed AMIs

- Protect valuable data by enforcing a regular backup schedule
- Create standardized AMIs that can be refreshed at regular intervals
- Retain backups as required by auditors or internal compliance
- Reduce storage costs by deleting outdated backups
- Create disaster recovery backup policies that back up data to isolated accounts

Network Attached Storage



File Management

The Operating System (OS)
sees a filesystem that is
mapped to a local drive letter

The NAS “shares”
filesystems over the
network



NIC

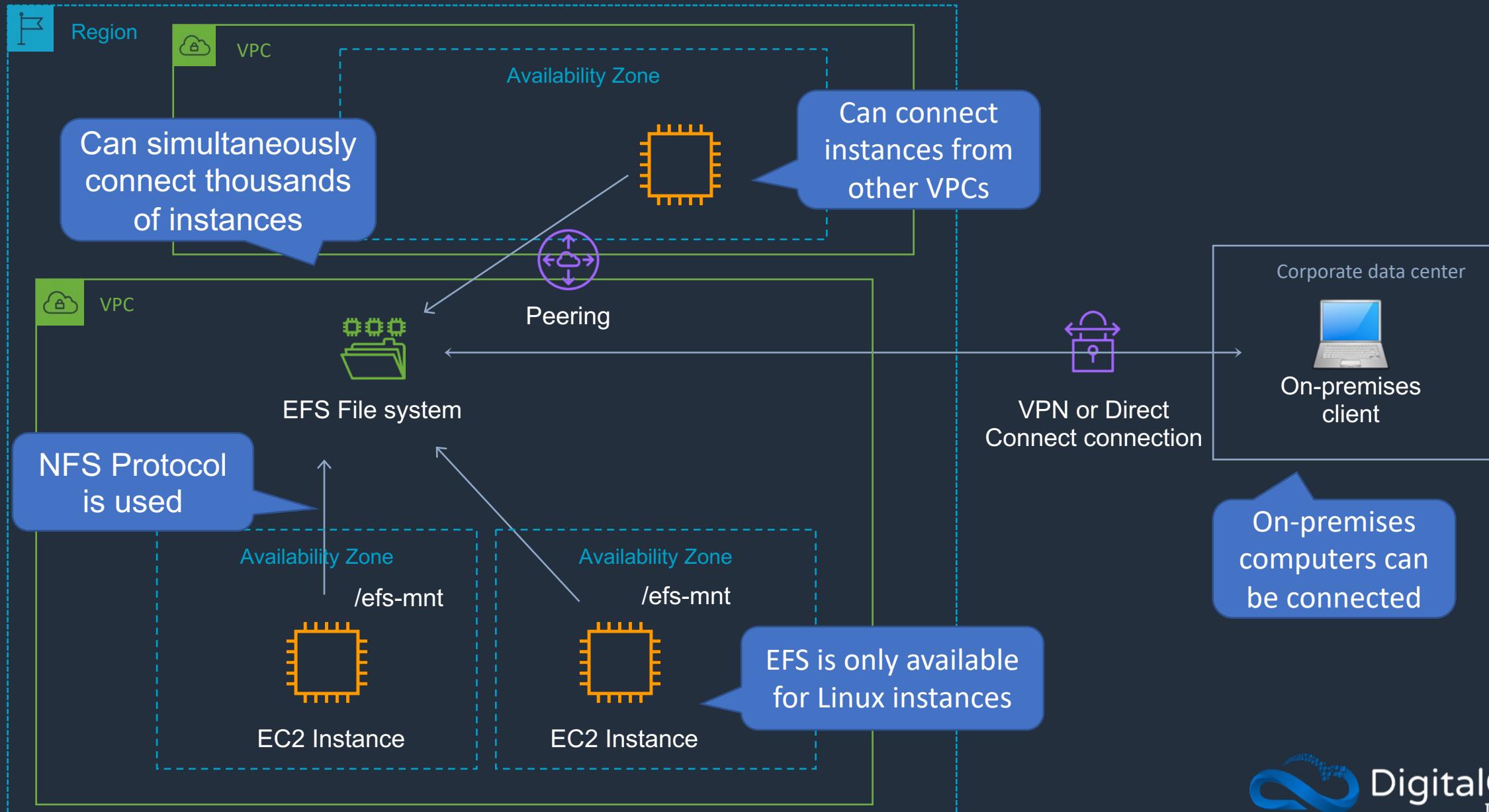


Network Attached
Storage Server (NAS)



NAS devices are file-based storage systems

Amazon Elastic File System (EFS) Overview



Amazon EFS Backups and Lifecycle Management

- Automatic backups are enabled by default and use AWS Backup
- Lifecycle management moves files that have not been accessed for a period of time to the EFS Infrequent Access Storage class

Lifecycle management

Automatically save money as access patterns change by moving files into the EFS Infrequent Access storage class. [Learn more](#) 

30 days since last access



None

7 days since last access

14 days since last access

30 days since last access

60 days since last access

90 days since last access

Amazon EFS Performance

- There are two performance modes:
 - “General purpose” – suitable for most use cases
 - “Max I/O” – Scales to higher levels of aggregate throughput and operations per second

Performance mode

Set your file system's performance mode based on IOPS required. [Learn more](#)

General Purpose
Ideal for latency-sensitive use cases, like web serving environments and content management systems

Max I/O
Scale to higher levels of aggregate throughput and operations per second

- There are two throughput modes:
 - “Bursting” – throughput scales with file system size
 - “Provisioned” – Throughput is fixed at the specified amount

Throughput mode

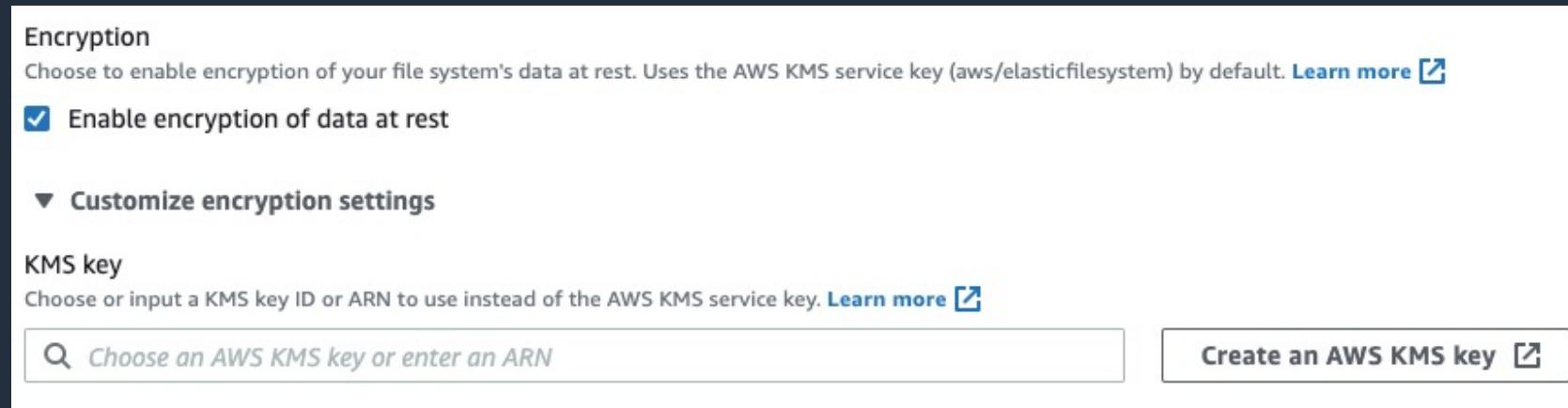
Set how your file system's throughput limits are determined. [Learn more](#)

Bursting
Throughput scales with file system size

Provisioned
Throughput fixed at specified amount

Amazon EFS Encryption

- EFS offers the ability to encrypt data at rest and in transit
- Encryption at rest is enabled by default and can be enabled in the EFS console or by using the AWS CLI or SDKs



- Encryption keys are managed by the AWS Key Management Service (KMS)
- Encryption of data at rest and of data in transit can be configured together or separately

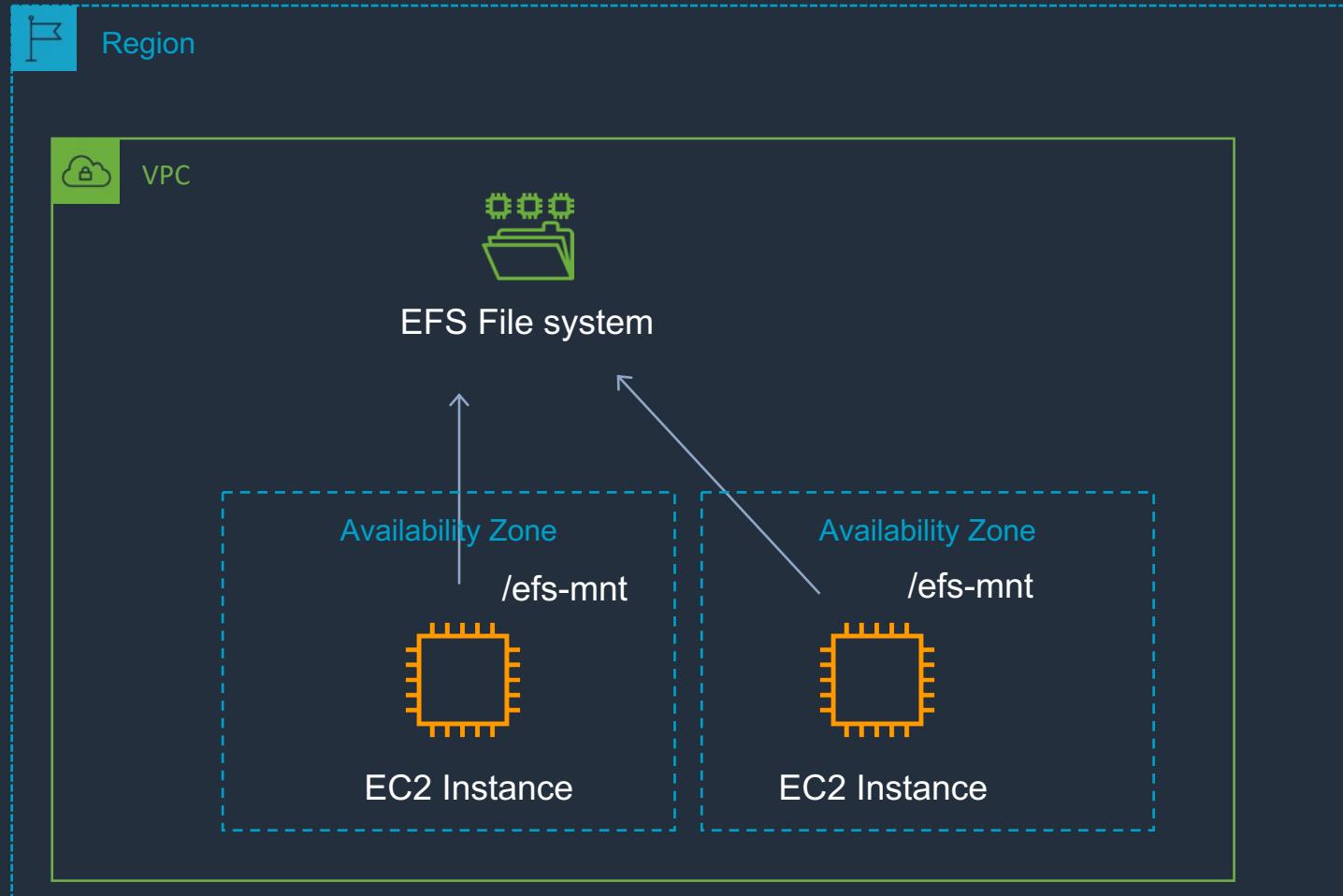
Amazon EFS Access Control

- When you create a file system, you create endpoints in your VPC called “mount targets”
- The file system’s DNS name resolves to a mount target’s IP address

Availability zone	Mount target ID	Subnet ID	Mount target state	IP address	Network interface ID
ap-southeast-2a	fsmt-7997e340	subnet-668f8301	Available	172.31.2.49	eni-014e304e143368dfb
ap-southeast-2b	fsmt-7a97e343	subnet-c9f8c180	Available	172.31.39.141	eni-0688b06d1be5aeee6
ap-southeast-2c	fsmt-7c97e345	subnet-fcdd63a4	Available	172.31.17.178	eni-0cb6f361323b5f4ed

- You can control file system admin using IAM (user-based and resource-based policies)
- You can control the NFS clients access to file systems (resource-based policies).
- You can control access to files and directories with POSIX-compliant user and group-level permissions

Amazon Elastic File System (EFS)



IAM Policy Example - Allow a User to Perform All Amazon EFS Actions

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid" : "Stmt1PermissionForAllEFSActions",  
            "Effect": "Allow",  
            "Action": "elasticfilesystem:*",  
            "Resource": "arn:aws:elasticfilesystem:us-west-2:123456789012:file-system/*"  
        },  
        {  
            "Sid" : "Stmt2RequiredEC2PermissionsForAllEFSActions",  
            "Effect": "Allow",  
            "Action": [  
                "ec2:DescribeSubnets",  
                "ec2>CreateNetworkInterface",  
                "ec2:DescribeNetworkInterfaces",  
                "ec2:DeleteNetworkInterface",  
                "ec2:ModifyNetworkInterfaceAttribute",  
                "ec2:DescribeNetworkInterfaceAttribute"  
            ],  
            "Resource": "*"  
        }  
    ]  
}
```

IAM Policy Example - Allow a User to Create a Mount Target and Tags on an Existing File System

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid" : "Stmt1CreateMountTargetAndTag",  
            "Effect": "Allow",  
            "Action": [  
                "elasticfilesystem>CreateMountTarget",  
                "elasticfilesystem>DescribeMountTargets",  
                "elasticfilesystem>CreateTags",  
                "elasticfilesystem>DescribeTags"  
            ],  
            "Resource": "arn:aws:elasticfilesystem:us-west-2:123456789012:file-system/file-system-ID"  
        },  
        {  
            "Sid" : "Stmt2AdditionalEC2PermissionsToCreateMountTarget",  
            "Effect": "Allow",  
            "Action": [  
                "ec2>DescribeSubnets",  
                "ec2>CreateNetworkInterface",  
                "ec2>DescribeNetworkInterfaces"  
            ],  
            "Resource": "*"  
        }  
    ]  
}
```

IAM Policy Example (resource-based) - Grant Read and Write Access to all IAM Principals

```
{  
    "Version": "2012-10-17",  
    "Id": "ExamplePolicy01",  
    "Statement": [  
        {  
            "Sid": "ExampleStatement01",  
            "Effect": "Allow",  
            "Principal": {  
                "AWS": "*"  
            },  
            "Action": [  
                "elasticfilesystem:ClientRootAccess",  
                "elasticfilesystem:ClientMount",  
                "elasticfilesystem:ClientWrite"  
            ],  
            "Condition": {  
                "Bool": {  
                    "aws:SecureTransport": "true"  
                }  
            }  
        }  
    ]  
}
```

Amazon EFS Encryption

Encryption In Transit



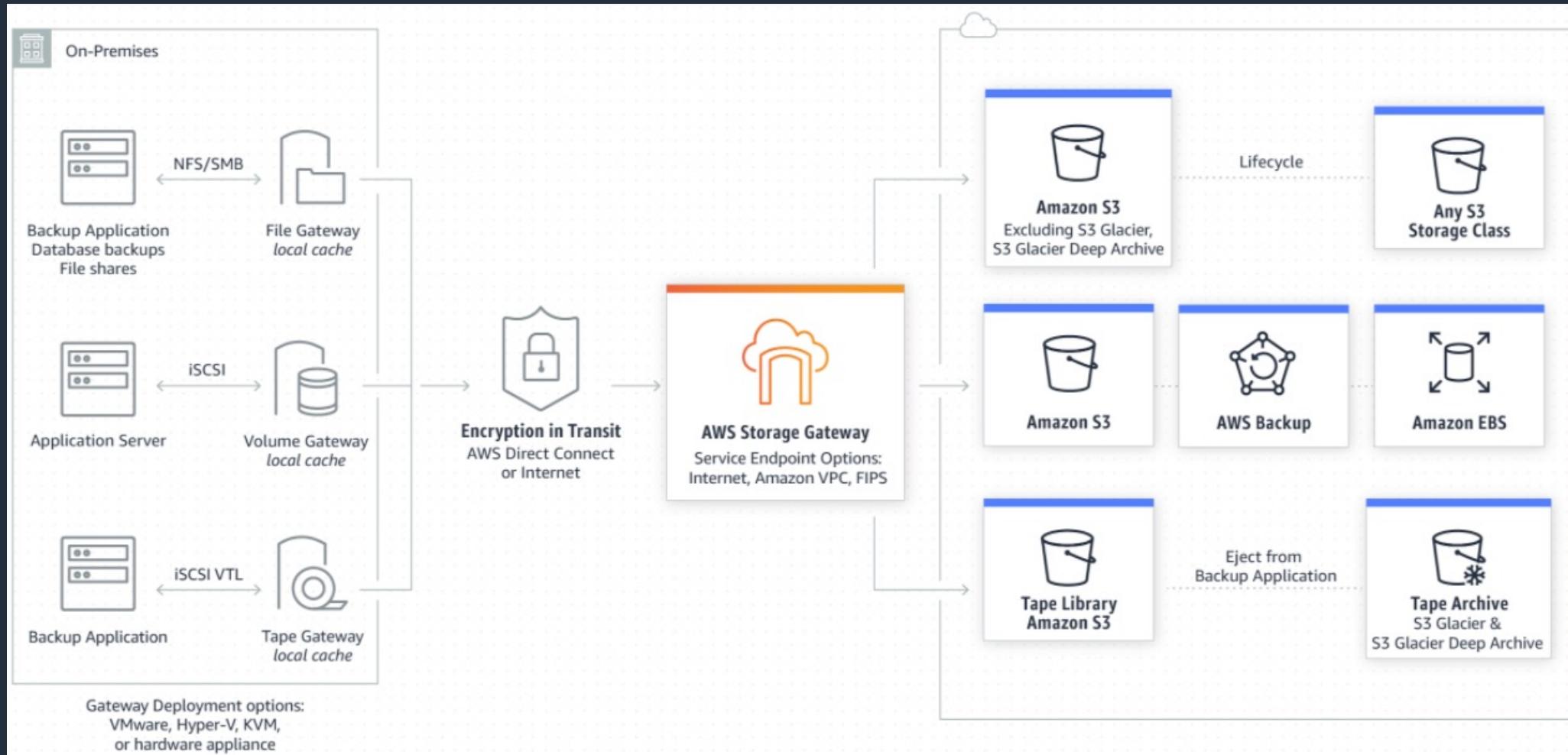
Encryption At Rest

Must be enabled at file system **creation time**

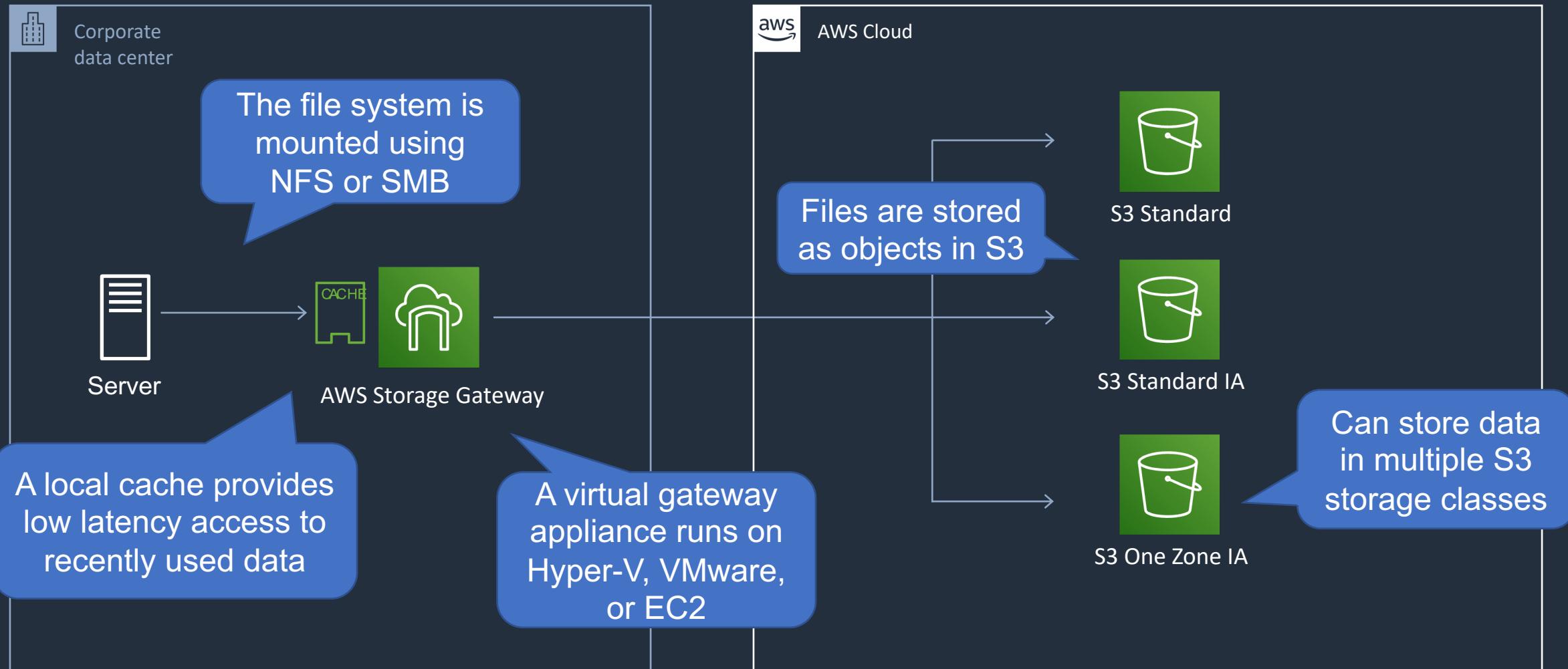


Can be combined with encryption in transit

AWS Storage Gateway



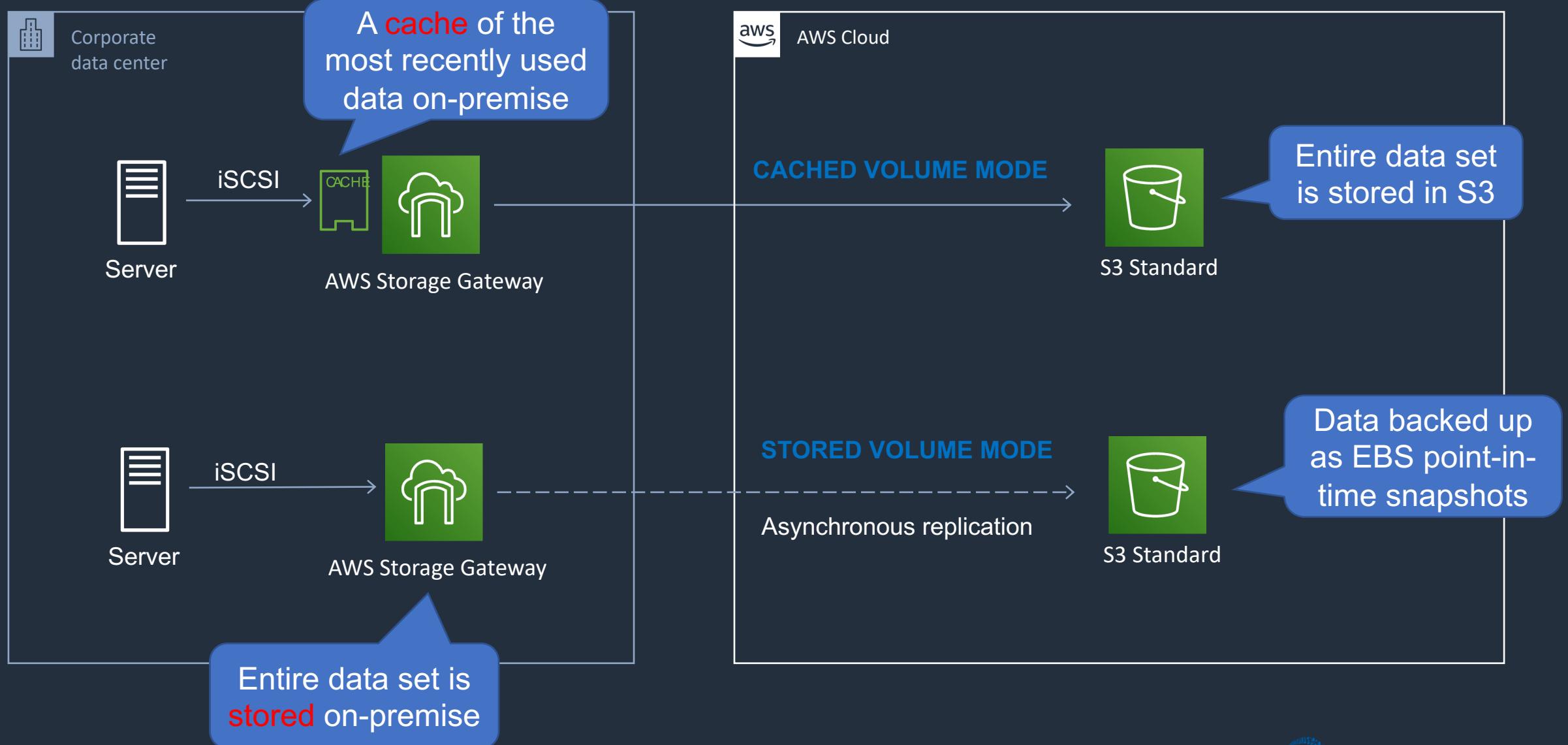
AWS Storage Gateway – File Gateway



AWS Storage Gateway – File Gateway

- File gateway provides a virtual on-premises file server, which enables you to store and retrieve files as objects in Amazon S3
- Can be used for on-premises applications, and for Amazon EC2-resident applications that need file storage in S3 for object based workloads
- Used for flat files only, stored directly on S3
- File gateway offers SMB or NFS-based access to data in Amazon S3 with local caching
- File gateway supports Amazon S3 Standard, S3 Standard – Infrequent Access (S3 Standard – IA) and S3 One Zone – IA

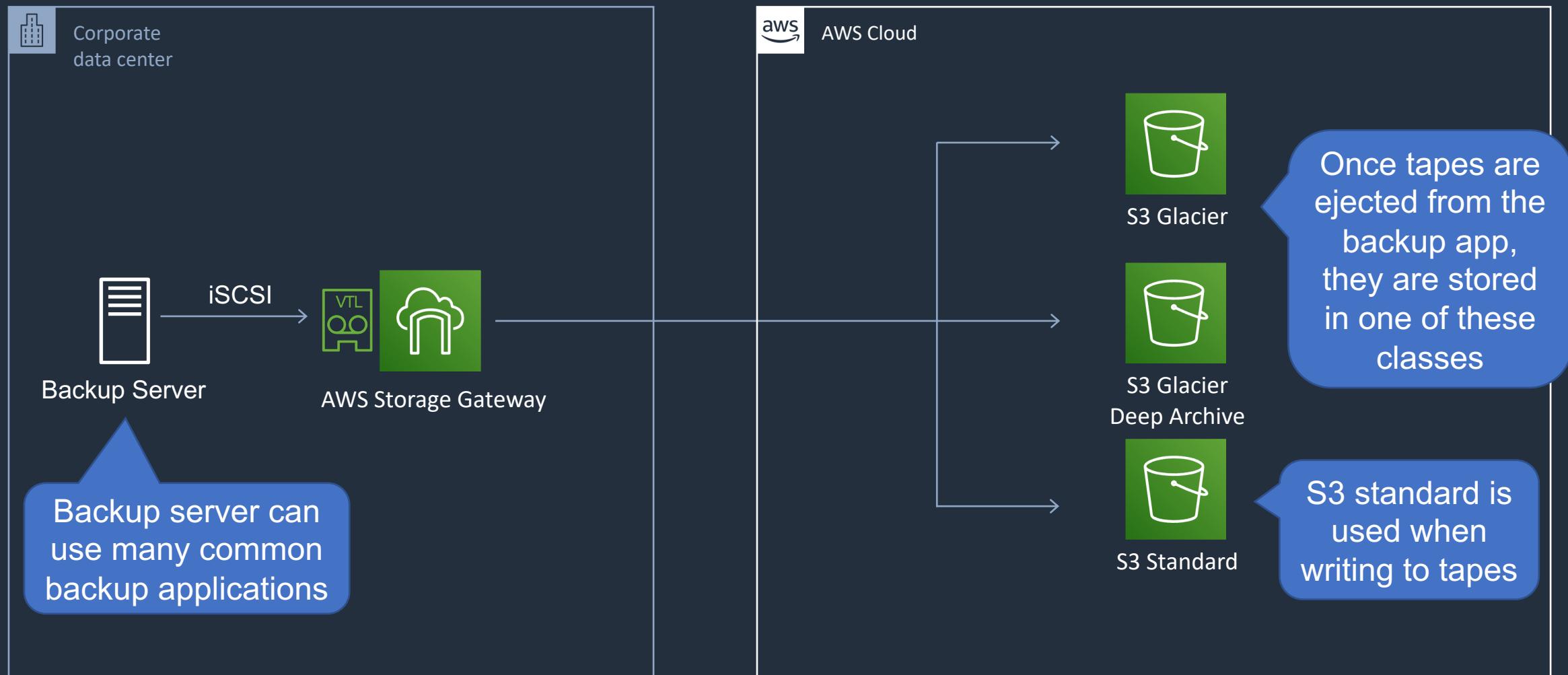
AWS Storage Gateway – Volume Gateway



AWS Storage Gateway – Volume Gateway

- The volume gateway represents the family of gateways that support block-based volumes, previously referred to as gateway-cached and gateway-stored modes
- Block storage – iSCSI based
- **Cached Volume mode** – the entire dataset is stored on S3 and a cache of the most frequently accessed data is cached on-site
- **Stored Volume mode** – the entire dataset is stored on-site and is asynchronously backed up to S3 (EBS point-in-time snapshots). Snapshots are incremental and compressed
- Each volume gateway can support up to 32 volumes
- In cached mode, each volume can be up to 32 TB for a maximum of 1 PB of data per gateway (32 volumes, each 32 TB in size)
- In stored mode, each volume can be up to 16 TB for a maximum of 512 TB of data per gateway (32 volumes, each 16 TB in size)

AWS Storage Gateway – Tape Gateway



AWS Storage Gateway – Tape Gateway

- Used for backup with popular backup software
- Each gateway is preconfigured with a media changer and tape drives. Supported by NetBackup, Backup Exec, Veeam etc.
- When creating virtual tapes, you select one of the following sizes: 100 GB, 200 GB, 400 GB, 800 GB, 1.5 TB, and 2.5 TB
- A tape gateway can have up to 1,500 virtual tapes with a maximum aggregate capacity of 1 PB
- All data transferred between the gateway and AWS storage is encrypted using SSL
- all data stored by tape gateway in S3 is encrypted server-side with Amazon S3-Managed Encryption Keys (SSE-S3)

Exam Scenarios

Exam Scenario	Solution
User deleted some data in an Amazon EBS volume and there's a recent snapshot	Can create a new EBS volume from the snapshot and attach it to an instance and copy the delete file across
EBS volume runs out of space and need to prevent it happening again	Use CloudWatch agent on EC2 and monitor disk metrics with CloudWatch alarm
Most cost-effective option for big data app that stores sequentially and infrequent access	Cold HDD (sc1)
EBS volume capacity is increased but cannot see the space	Need to extend the volume's file system to gain access to extra space

Exam Scenarios

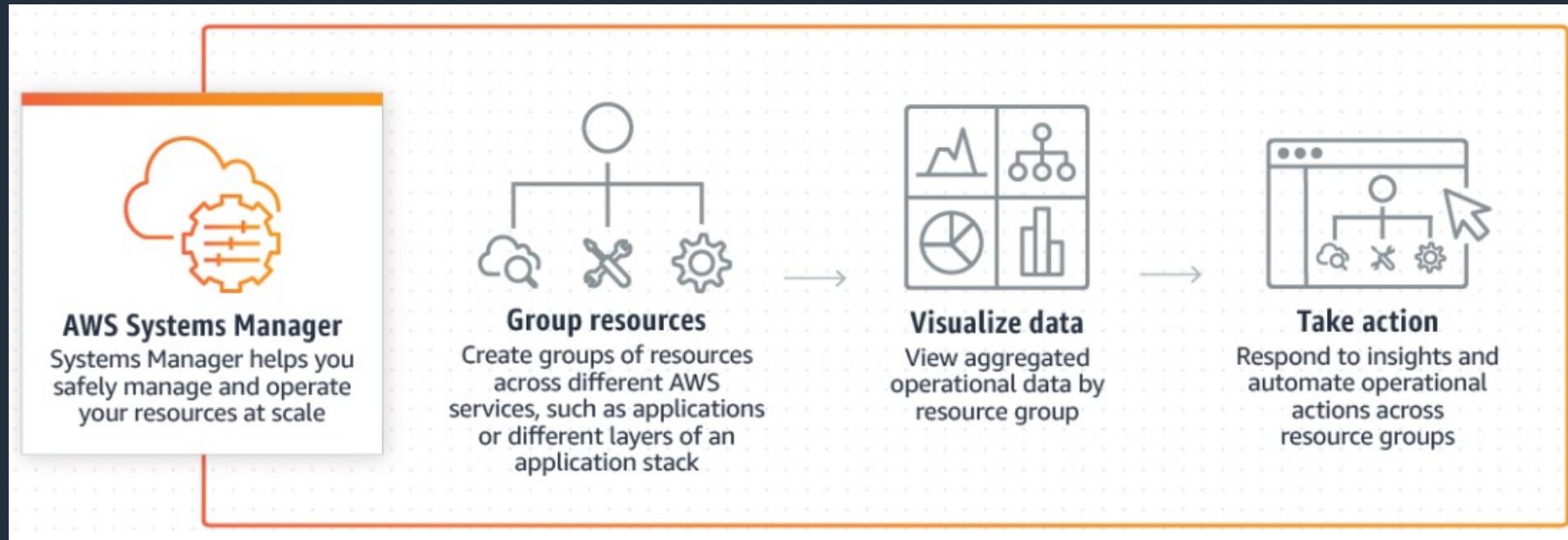
Exam Scenario	Solution
Need to replace user-shared drives. Must support POSIX permissions and NFS protocols and be accessible from on-premise servers and EC2	Use Amazon EFS
Low latency access required for image files in an office location with synchronized backup to offsite location. Local access required and disaster recovery	Use an AWS Storage Gateway volume gateway configured as a stored volume
Performance issues with iSCSI drives in volume gateway. CacheHitPercent metric is below 55% and CachePerecentUsed is above 95%	Create a larger disk for cached volume and select it in management console
Tape archival system needs replacement	Use an AWS Storage Gateway tape gateway

SECTION 6

Operations: AWS Systems Manager and OpsWorks

AWS Systems Manager

- AWS Systems Manager provides a unified interface through which you can view operational data from multiple AWS services



- With Systems Manager, you can group resources by application, view operational data for monitoring and troubleshooting, and take action on your groups of resources

AWS Systems Manager

- Manages many AWS resources including Amazon EC2, Amazon S3, Amazon RDS etc.
- You can create logical groups of resources such as applications, different layers of an application stack, or production vs development environments

AWS Systems Manager

- Systems Manager Components (in scope for the exam):

- Automation
- Run Command
- Inventory
- Patch Manager
- Session Manager
- Parameter Store

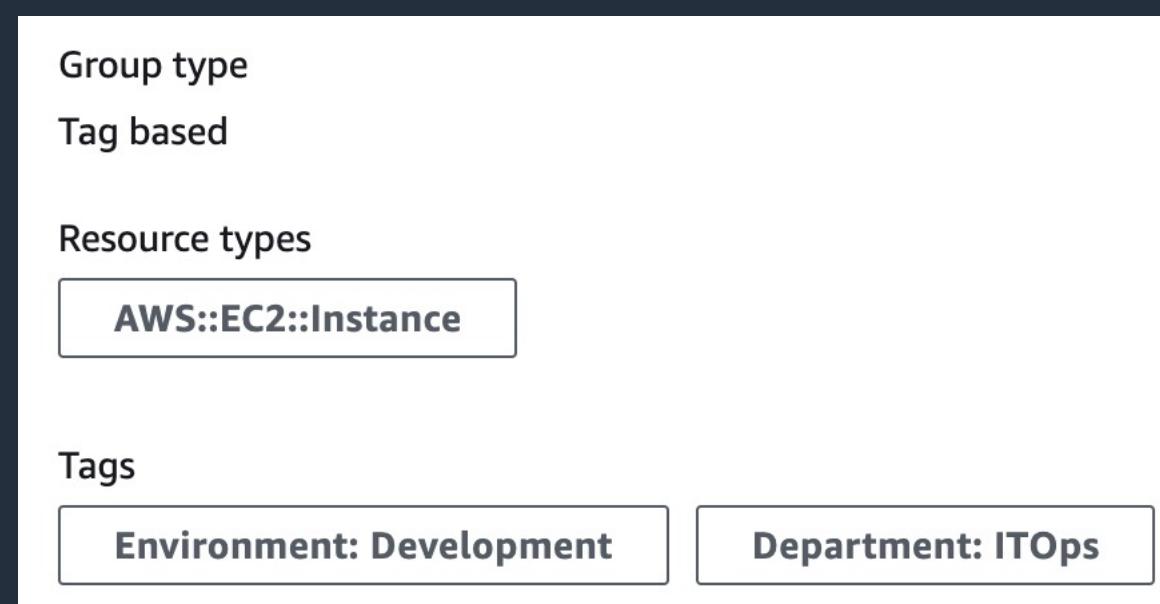
AWS Tags

- A tag is a label that you assign to an AWS resource
- Each tag consists of a key and an optional value
- Tags enable you to categorize your AWS resources in different ways, for example, by purpose, owner, or environment

Key	Value
Department	ITOps
Environment	Development
Name	DevelopmentServer01

AWS Resource Groups

- Resource groups can be used to organize AWS resources
- Resource groups make it easier to manage and automate tasks on large numbers of resources at one time



AWS Systems Manager – Automation

Documents define the actions to perform (YAML or JSON)

Automates IT operations and management tasks across AWS resources



Documents



Automation



Amazon RDS

```
description: Creates an RDS Snapshot for an RDS instance.  
schemaVersion: '0.3'  
assumeRole: "{{AutomationAssumeRole}}"  
parameters:  
  DBInstanceIdentifier:  
    type: String  
    description: (Required) The DBInstanceId ID of the RDS Instance.  
  DBSnapshotIdentifier:  
    type: String  
    description: (Optional) The DBSnapshotIdentifier ID.  
    default: ''  
  InstanceTags:  
    type: String  
    default: ''  
    description: (Optional) Tags to create for instance.  
  SnapshotTags:  
    type: String  
    default: ''  
    description: (Optional) Tags to create for snapshot
```

This automation, takes a snapshot of an RDS DB instance

AWS Systems Manager – Run Command

Document types include command, automation, package etc.

Runs commands on managed EC2 instances



Documents



Run Command

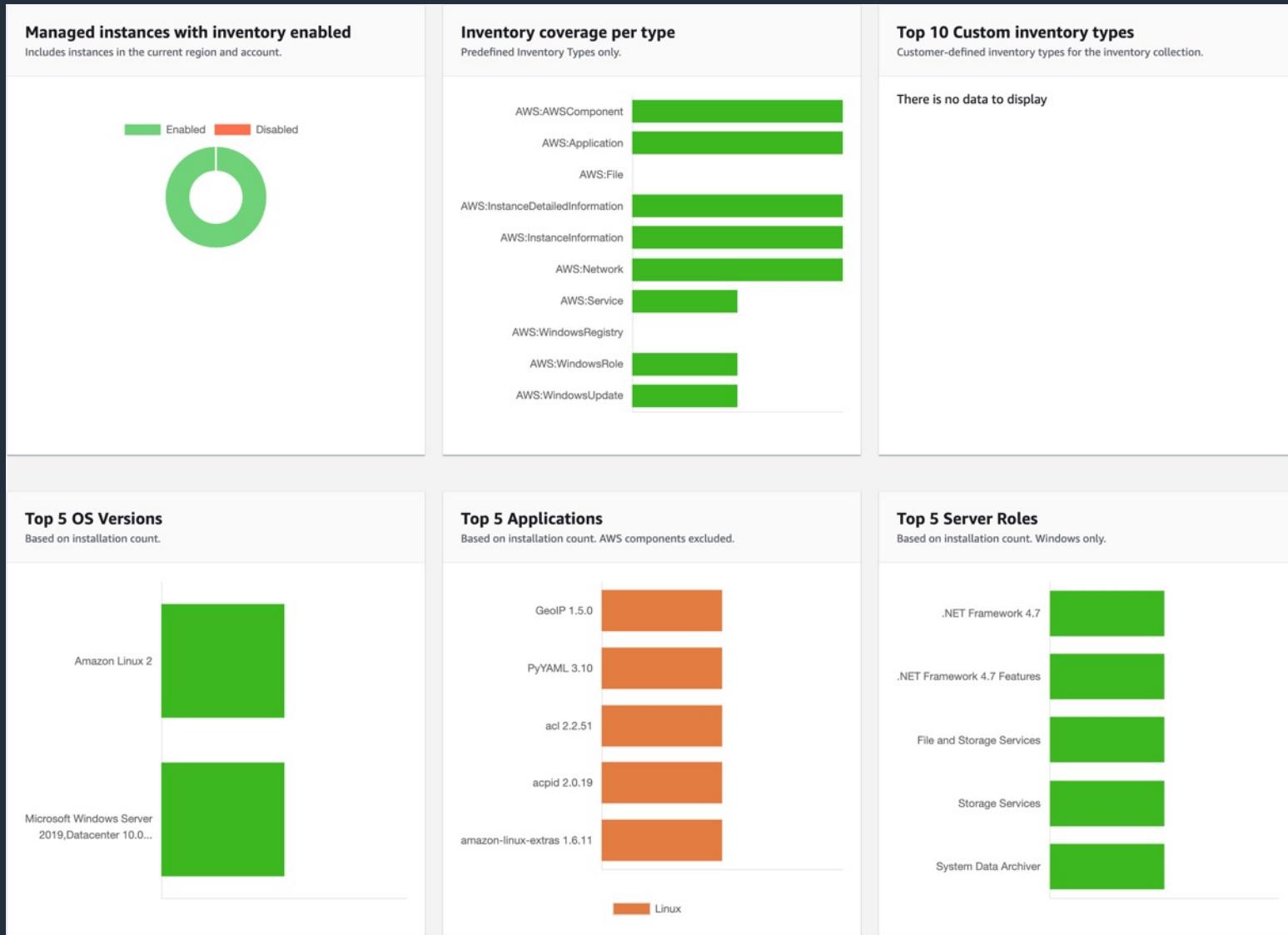


Amazon EC2

```
{  
  "schemaVersion": "1.2",  
  "description": "List missing Microsoft Windows updates.",  
  "parameters": {  
    "UpdateLevel": {  
      "type": "String",  
      "default": "Important",  
      "description": "Important: .....",  
      "allowedValues": [  
        "None",  
        "All",  
        "Important",  
        "Optional"  
      ]  
    }  
  }  
}
```

This command checks for missing updates

AWS Systems Manager – Inventory



Inventory

AWS Systems Manager – Patch Manager

- AWS Systems Manager helps you select and deploy operating system and software patches automatically across large groups of Amazon EC2 or on-premises instances
- Patch baselines:
 - Set rules to auto-approve select categories of patches to be installed
 - Specify a list of patches that override these rules and are automatically approved or rejected
- You can also schedule maintenance windows for your patches so that they are only applied during predefined times
- Systems Manager helps ensure that your software is up-to-date and meets your compliance policies



Patch Manager

AWS Systems Manager – Configuration Compliance

- AWS Systems Manager lets you scan your managed instances for patch compliance and configuration inconsistencies
- You can collect and aggregate data from multiple AWS accounts and Regions, and then drill down into specific resources that aren't compliant
- By default, AWS Systems Manager displays data about patching and associations
- You can also customize the service and create your own compliance types based on your requirements (must use the AWS CLI, AWS Tools for Windows PowerShell, or the SDKs)

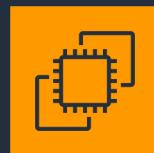
AWS Systems Manager – Session Manager

- Secure remote management of your instances at scale without logging into your servers
- Replaces the need for bastion hosts, SSH, or remote PowerShell
- Integrates with AWS Identity and Access Management (IAM) for granular permissions
- All actions taken with Systems Manager are recorded by AWS CloudTrail
- Can store session logs in an Amazon S3 bucket (optional encryption)
- Can send session output to CloudWatch Logs (optional encryption)

Doesn't require port 22,5985/5986



No need for bastion hosts



Amazon EC2
(Linux)



Amazon EC2
(Windows)

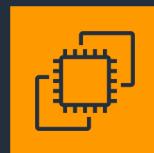
AWS Systems Manager – Session Manager

- Requires IAM permissions for EC2 instance to access SSM, S3, and CloudWatch Logs

Doesn't require port 22,5985/5986



No need for bastion hosts



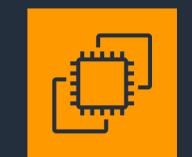
Amazon EC2
(Linux)



Amazon EC2
(Windows)

AWS Systems Manager Parameter Store

- AWS Systems Manager Parameter Store provides secure, hierarchical storage for configuration data management and secrets management
- It is highly scalable, available, and durable
- You can store data such as passwords, database strings, and license codes as parameter values
- You can store values as plaintext (unencrypted data) or ciphertext (encrypted data)
- You can then reference values by using the unique name that you specified when you created the parameter



Amazon EC2



Amazon RDS



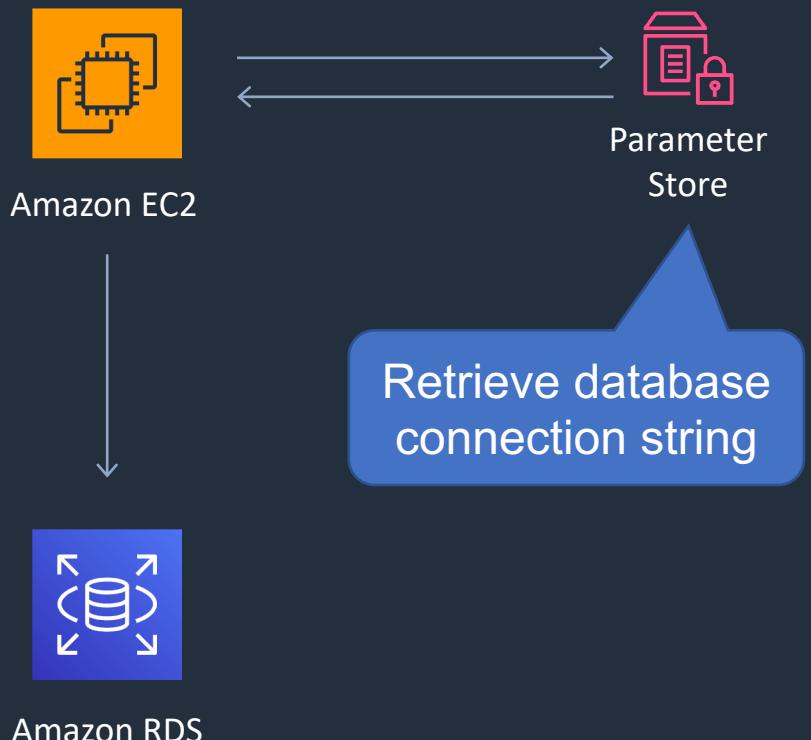
Parameter
Store



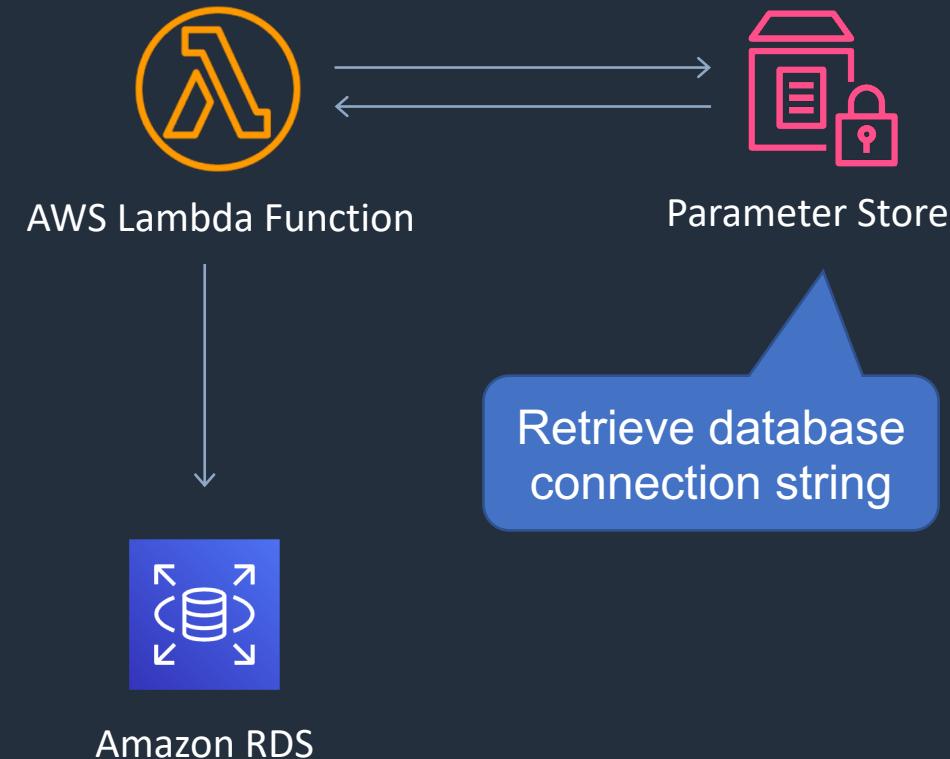
Retrieve database connection string

AWS Systems Manager Parameter Store

- No native rotation of keys (difference with AWS Secrets Manager which does it automatically)
- There are two tiers:
 - Standard – limit of 10,000 parameters, up to 4 KB, no additional charges
 - Advanced – more than 10,000 parameters, up to 8 KB, charges apply

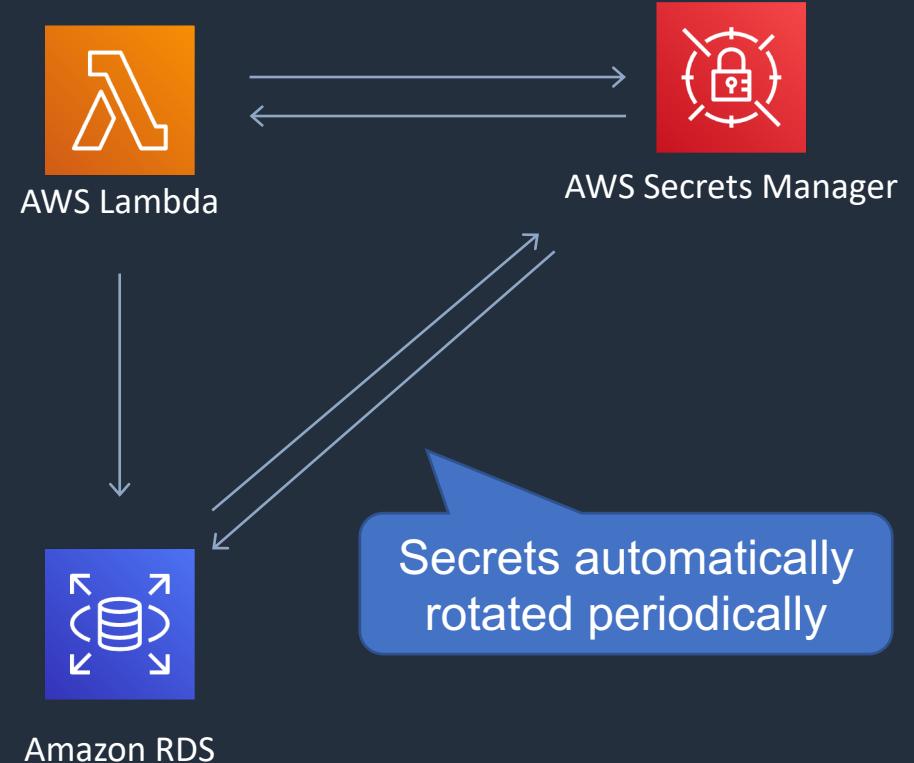


AWS Systems Manager Parameter Store



AWS Secrets Manager

- Stores and rotate secrets safely without the need for code deployments
- Secrets Manager offers automatic rotation of credentials (built-in) for:
 - Amazon RDS (MySQL, PostgreSQL, and Amazon Aurora)
 - Amazon Redshift
 - Amazon DocumentDB
- For other services you can write your own AWS Lambda function for automatic rotation

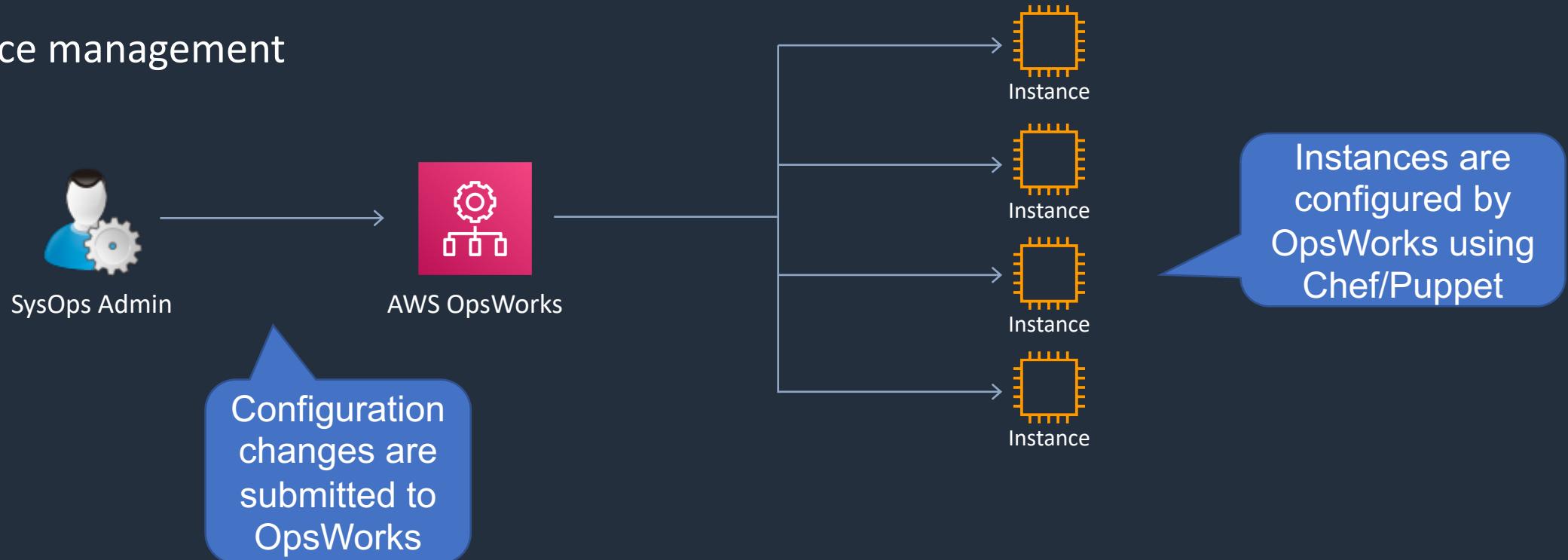


AWS Secrets Manager vs SSM Parameter Store

	Secrets Manager	SSM Parameter Store
Automatic Key Rotation	Yes, built-in for some services, use Lambda for others	No native key rotation
Key/Value Type	Encrypted only	String, StringList, SecureString (encrypted)
Change history	No	Yes
Price	Charges apply per secret	Free for standard, charges for advanced

AWS OpsWorks

- AWS OpsWorks is a configuration management service that provides managed instances of Chef and Puppet
- Updates include patching, updating, backup, configuration and compliance management



Exam Scenarios

Exam Scenario	Solution
Application running on EC2 needs login credentials for a DB that are stored as secure strings in SSM Parameter Store	Create an IAM role for the instance and grant permission to read the parameters
Linux instances are patched with Systems Manager Patch Manager. Application slows down whilst updates are happening	Change maintenance window to patch 10% of instances in the patch group at a time
Custom Linux AMI used with AWS Systems Manager. Can't find instances in Session Manager console	Need to add permissions to instance profile and install the SSM agent on the instances

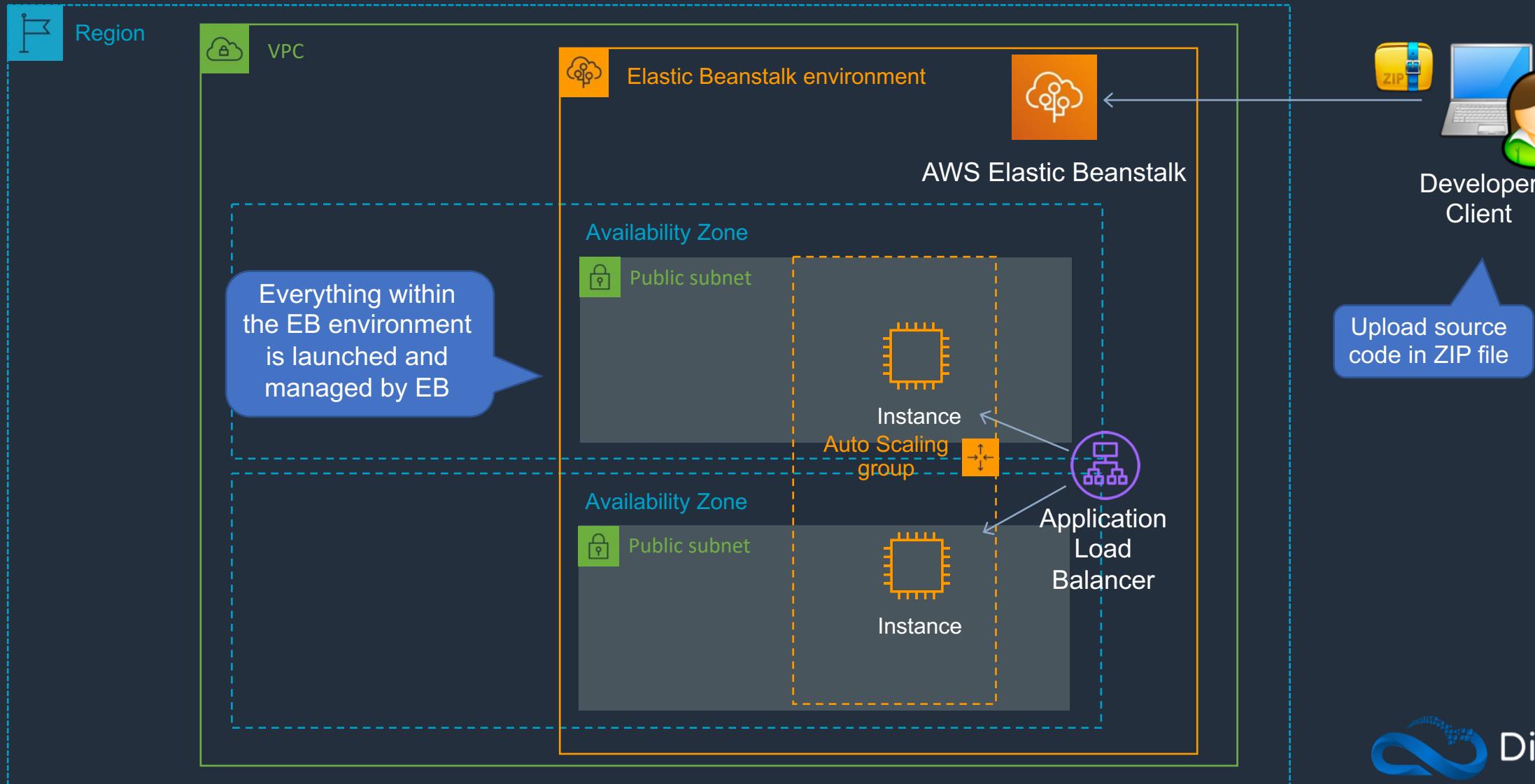
Exam Scenarios

Exam Scenario	Solution
Multiple environments require authentication credentials for external service. Deployed using CloudFormation	Store credentials in SSM Parameter Store and pass an environment tag as a parameter in CloudFormation template
IAM access keys used to manage EC2 instances using the CLI. Company policy mandates that access keys are automatically disabled after 60 days	Use an AWS Config rule to identify noncompliant keys. Create a custom AWS Systems Manager Automation document for remediation

SECTION 7

Automation: AWS Elastic Beanstalk

AWS Elastic Beanstalk



AWS Elastic Beanstalk

There are several layers:

Applications:

- Contain environments, environment configurations, and application versions
- You can have multiple application versions held within an application

APPLICATION

AWS Elastic Beanstalk

Application version

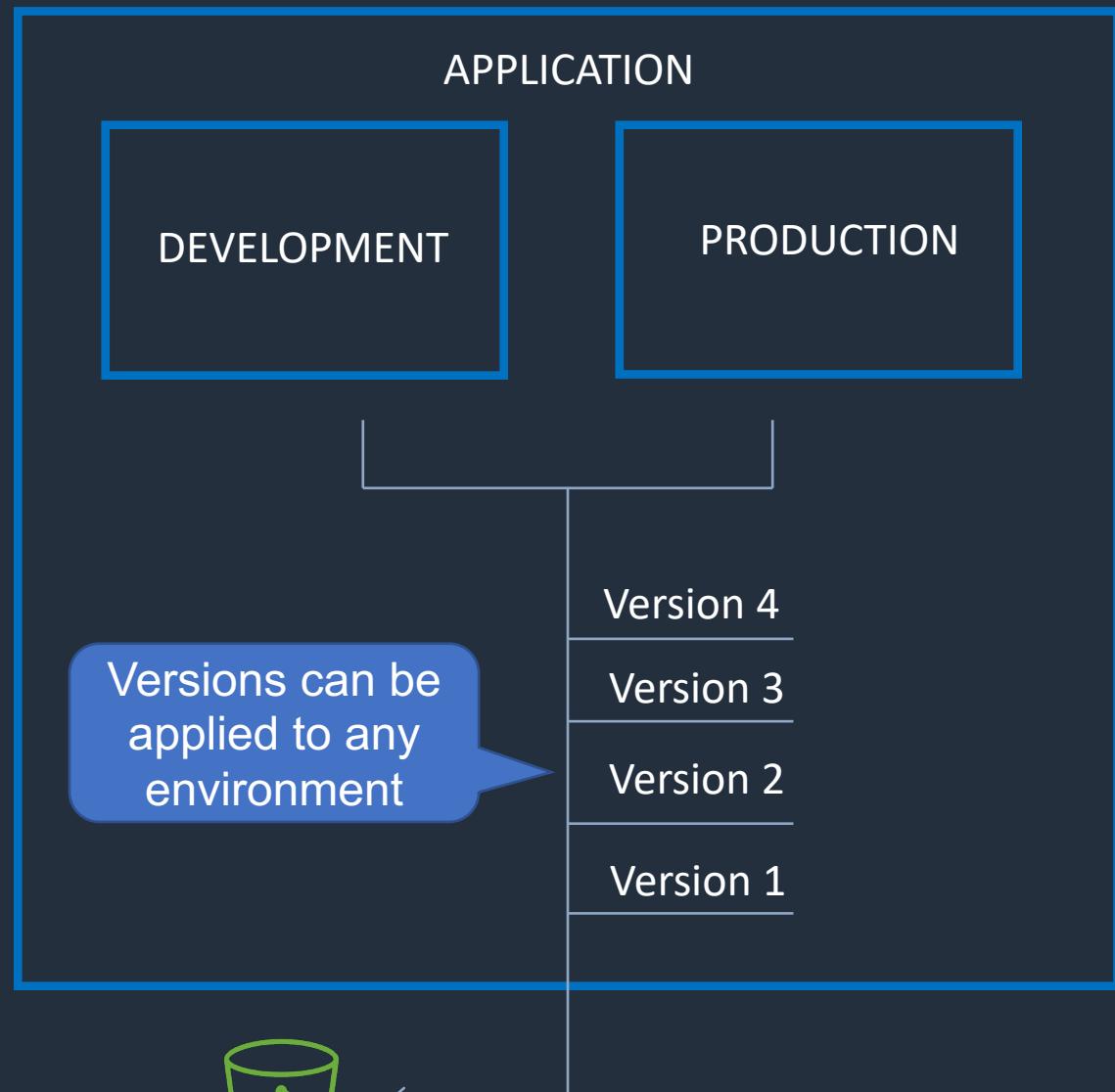
- A specific reference to a section of deployable code
- The application version will point typically to an Amazon S3 bucket containing the code



AWS Elastic Beanstalk

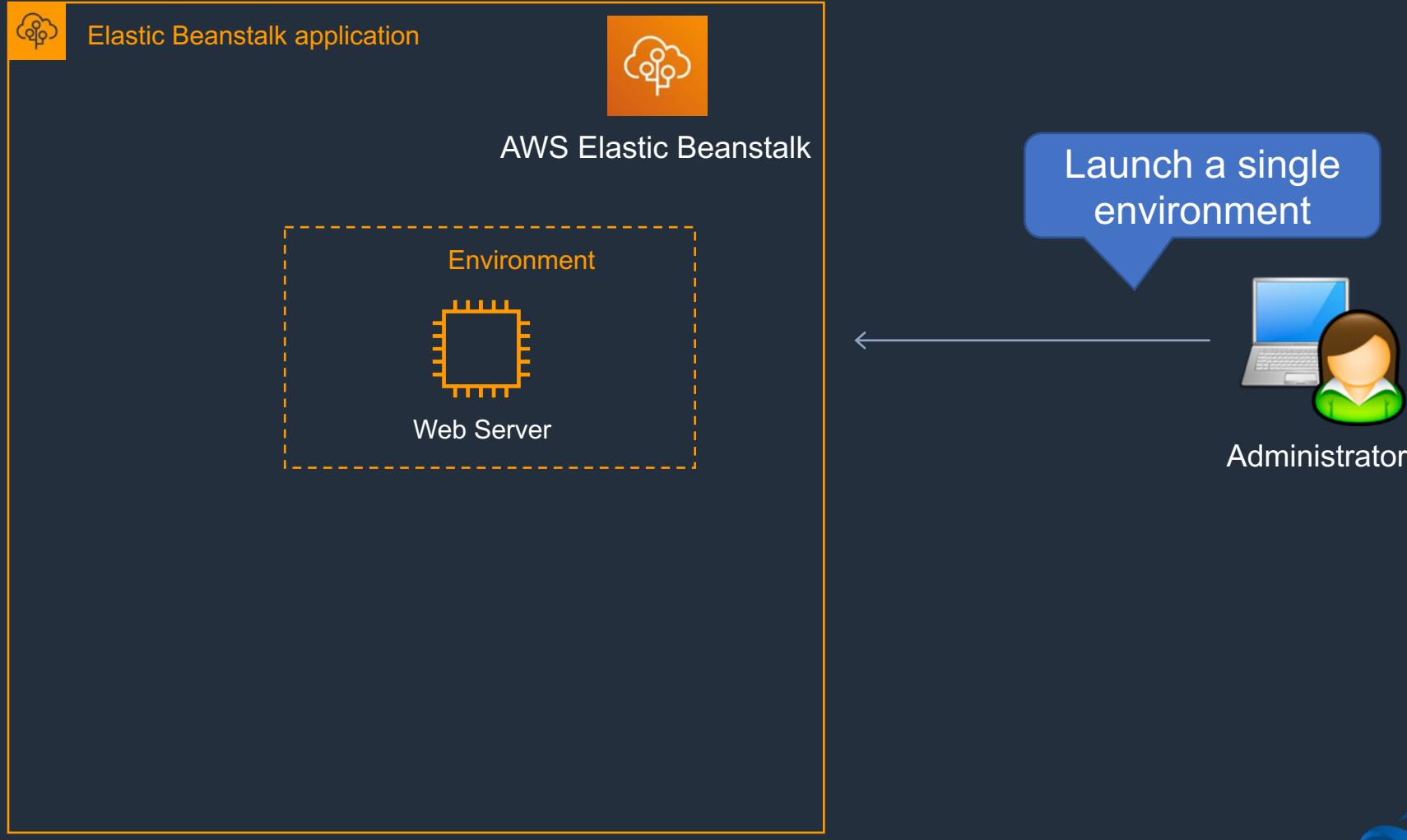
Environments:

- An application version that has been deployed on AWS resources
- The resources are configured and provisioned by AWS Elastic Beanstalk
- The environment is comprised of all the resources created by Elastic Beanstalk and not just an EC2 instance with your uploaded code

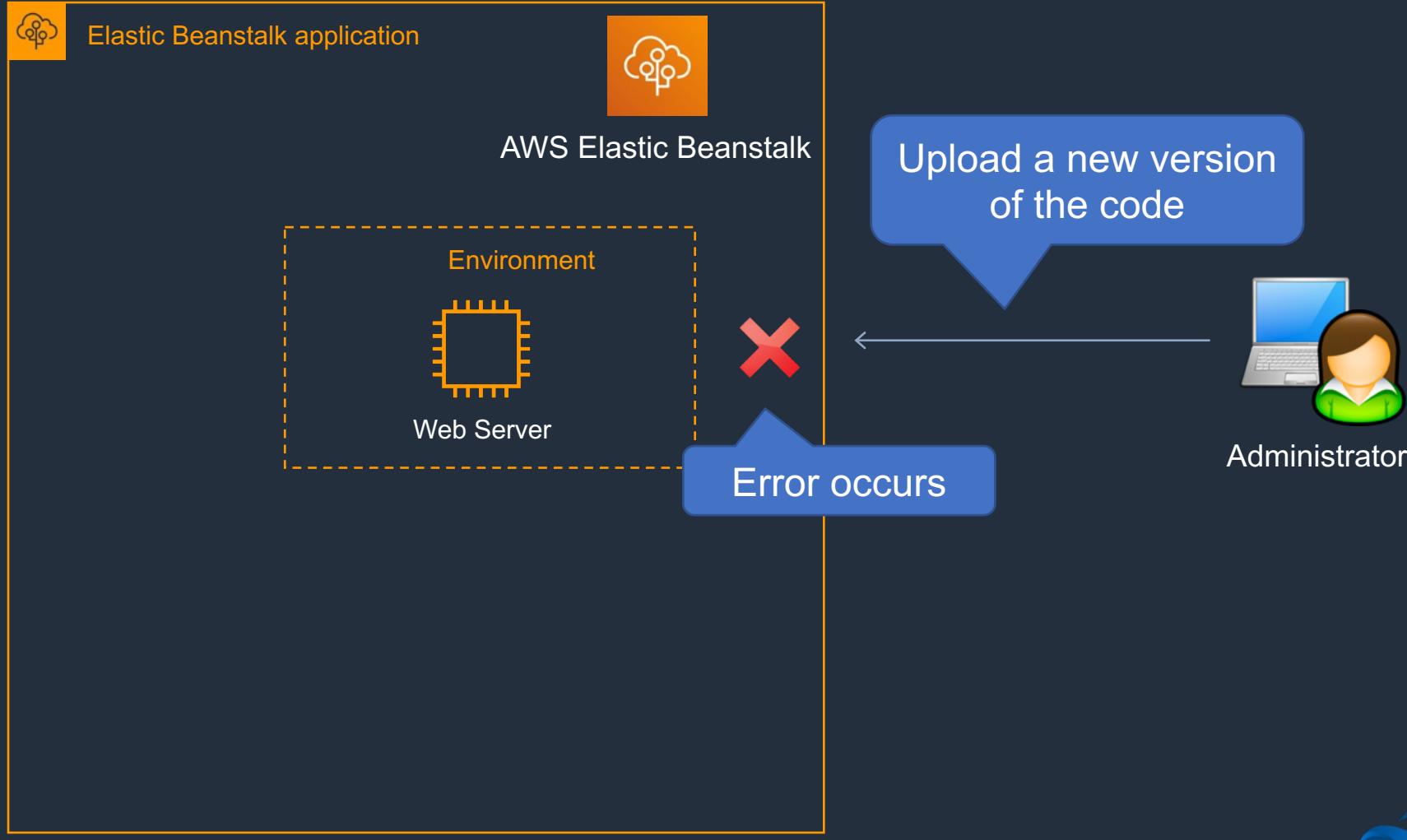


S3 Bucket

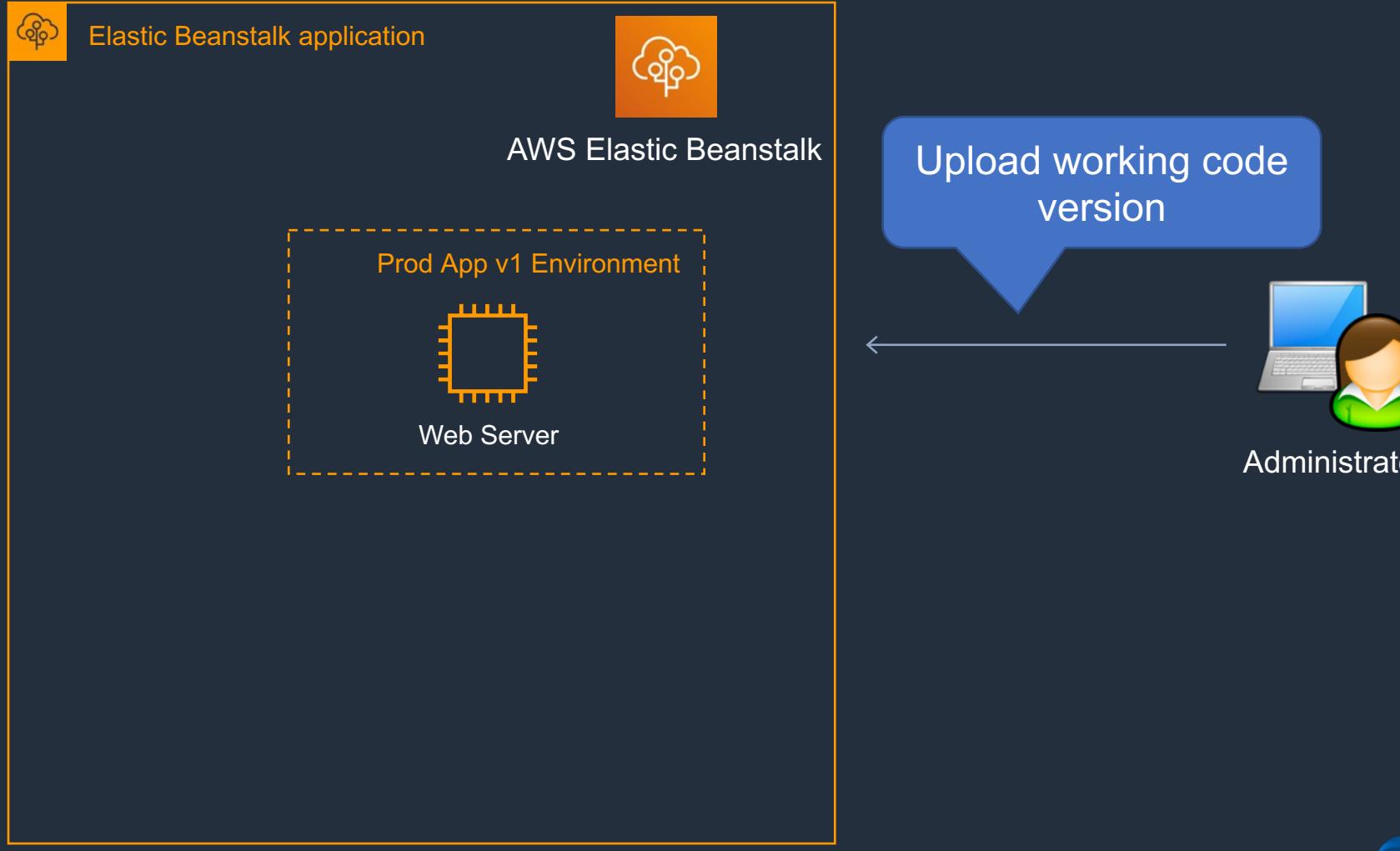
AWS Elastic Beanstalk – Create Single Environment



AWS Elastic Beanstalk – Troubleshooting errors



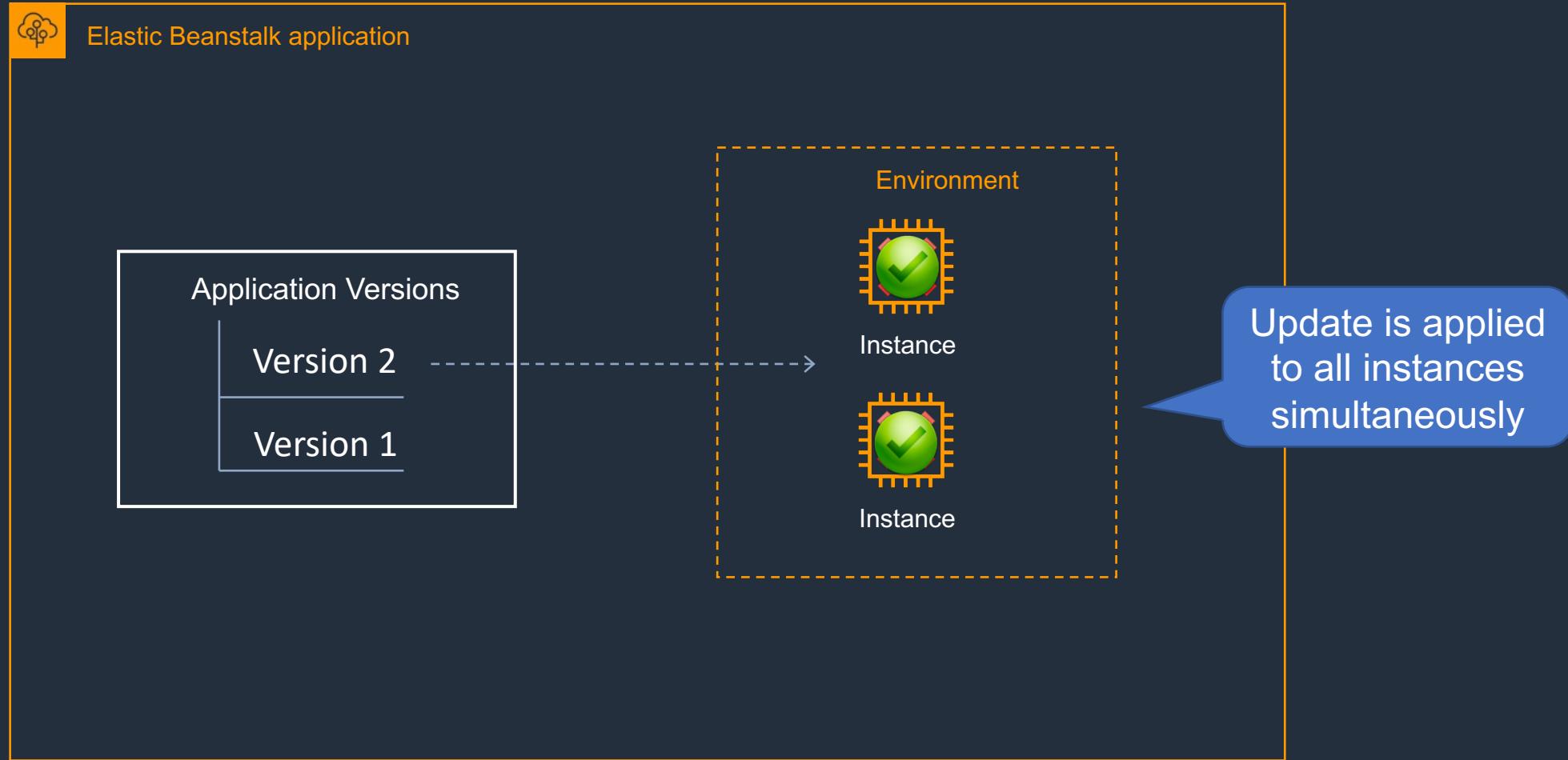
AWS Elastic Beanstalk – Upload Prod App v1



AWS Elastic Beanstalk Deployment Policies

- All at once:
 - Deploys the new version to all instances simultaneously
- Rolling:
 - Update a batch of instances, and then move onto the next batch once the first batch is healthy
- Rolling with additional batch:
 - Like Rolling but launches new instances in a batch ensuring that there is full availability
- Immutable:
 - Launches new instances in a new ASG and deploys the version update to these instances before swapping traffic to these instances once healthy
- Blue/green:
 - Create a new "stage" environment and deploy updates there

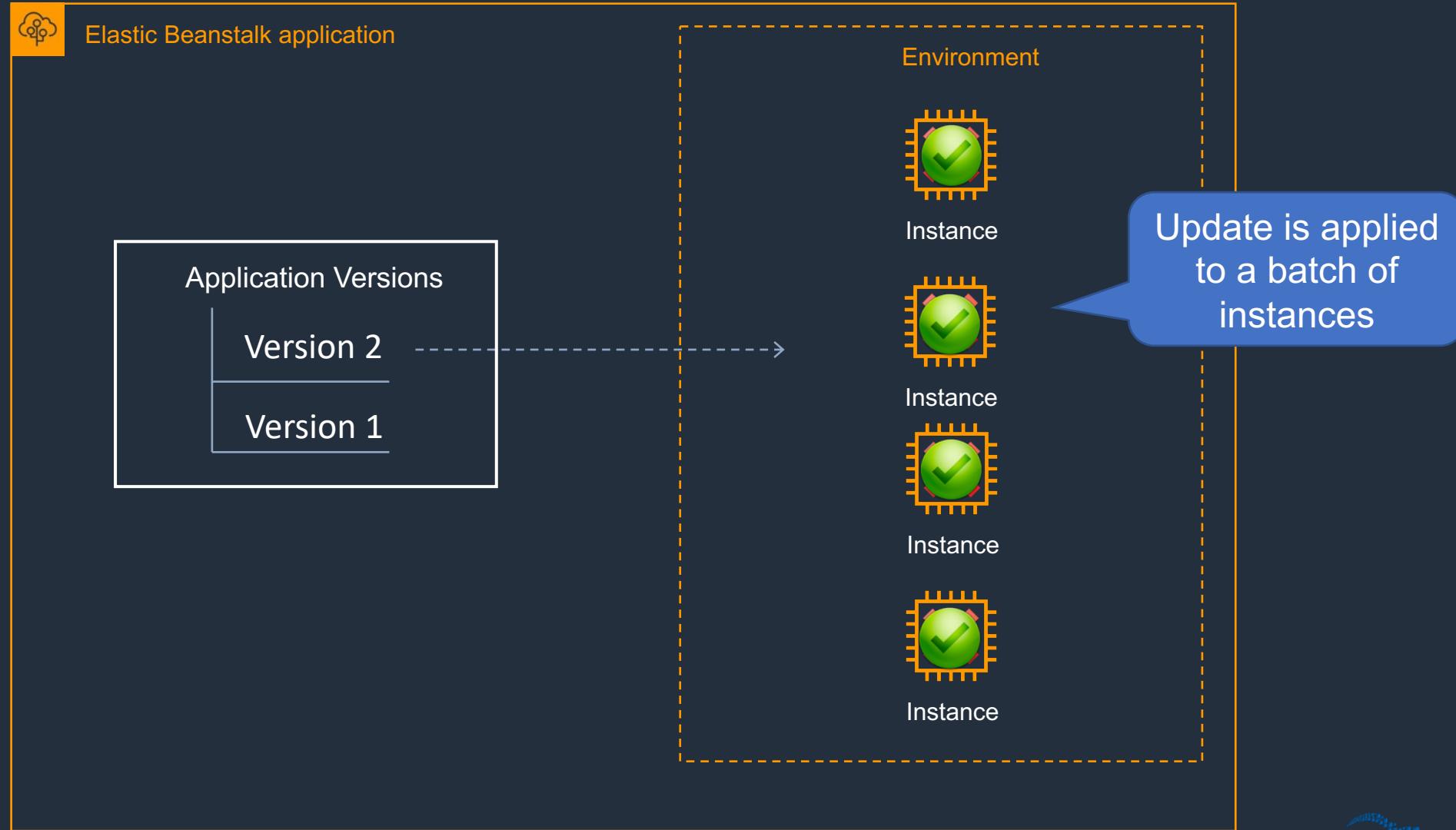
AWS Elastic Beanstalk – All at Once Update



AWS Elastic Beanstalk – All at Once Update

- Deploys the new version to all instances simultaneously
- All of your instances are out of service while the deployment takes place
- Fastest deployment
- Good for quick iterations in development environment
- You will experience an outage while the deployment is taking place - not ideal for mission-critical systems
- If the update fails, you need to roll back the changes by re-deploying the original version to all of your instances
- No additional cost

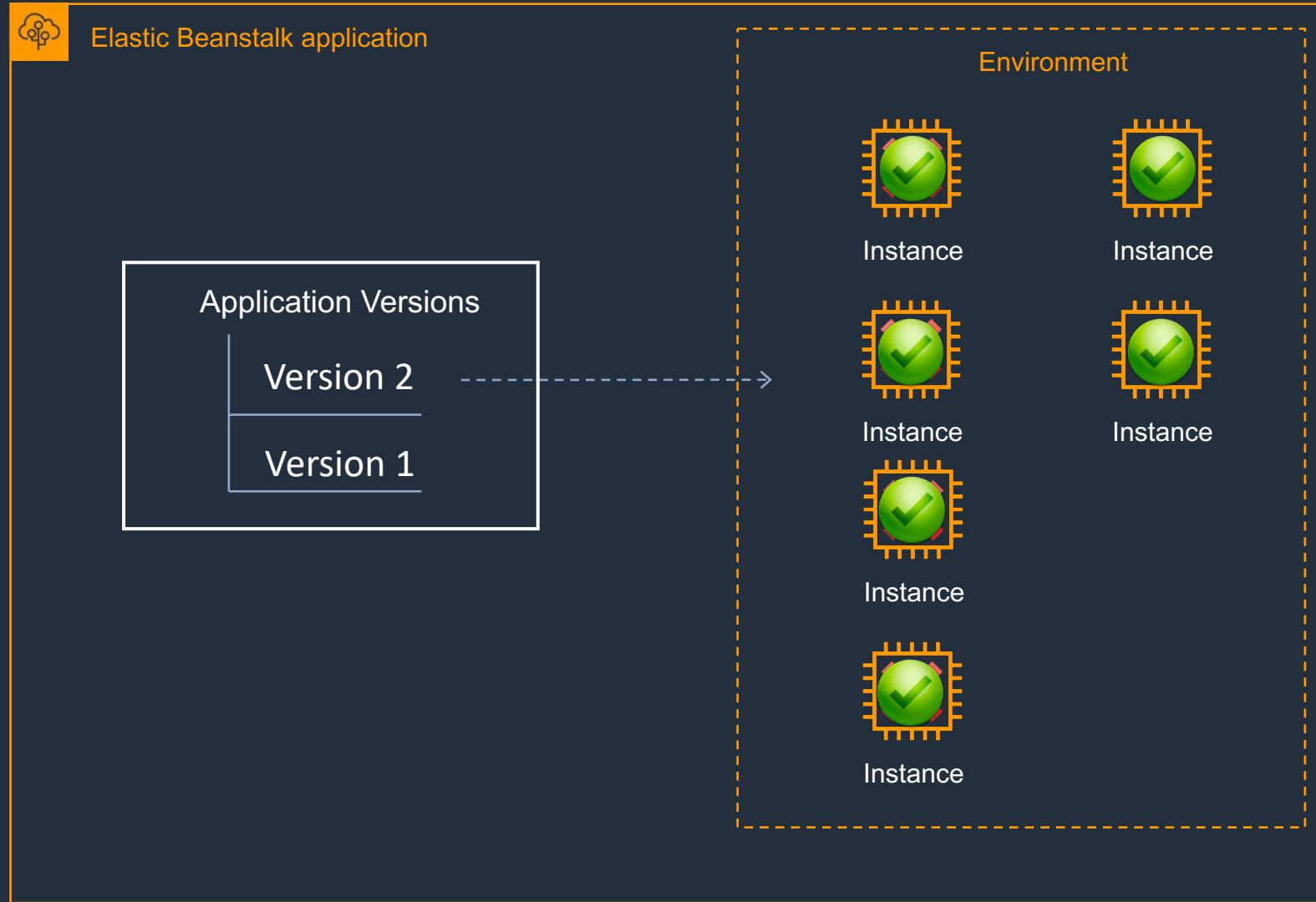
AWS Elastic Beanstalk – Rolling Update



AWS Elastic Beanstalk – Rolling Update

- Update a few instances at a time (batch), and then move onto the next batch once the first batch is healthy (downtime for 1 batch at a time)
- Application is running both versions simultaneously
- Each batch of instances is taken out of service while the deployment takes place
- Your environment capacity will be reduced by the number of instances in a batch while the deployment takes place
- Not ideal for performance-sensitive systems
- If the update fails, you need to perform an additional rolling update to roll back the changes.
- No additional cost
- Long deployment time

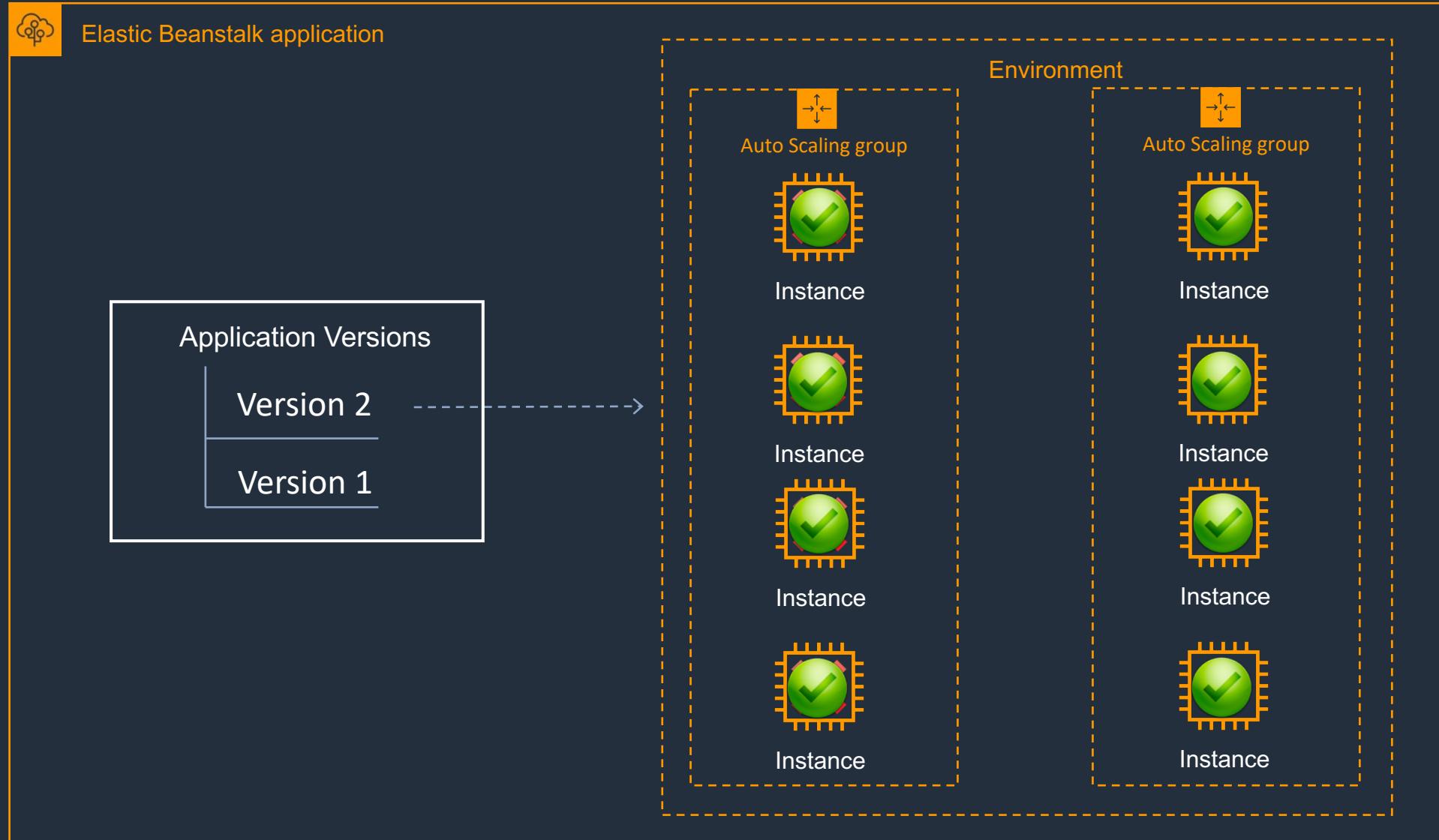
AWS Elastic Beanstalk – Rolling with Additional Batch Update



AWS Elastic Beanstalk – Rolling with Additional Batch Update

- Like Rolling but launches new instances in a batch ensuring that there is full availability.
- Application is running at capacity
- Can set the batch size
- Application is running both versions simultaneously
- Small additional cost
- Additional batch is removed at the end of the deployment
- Longer deployment
- Good for production environments

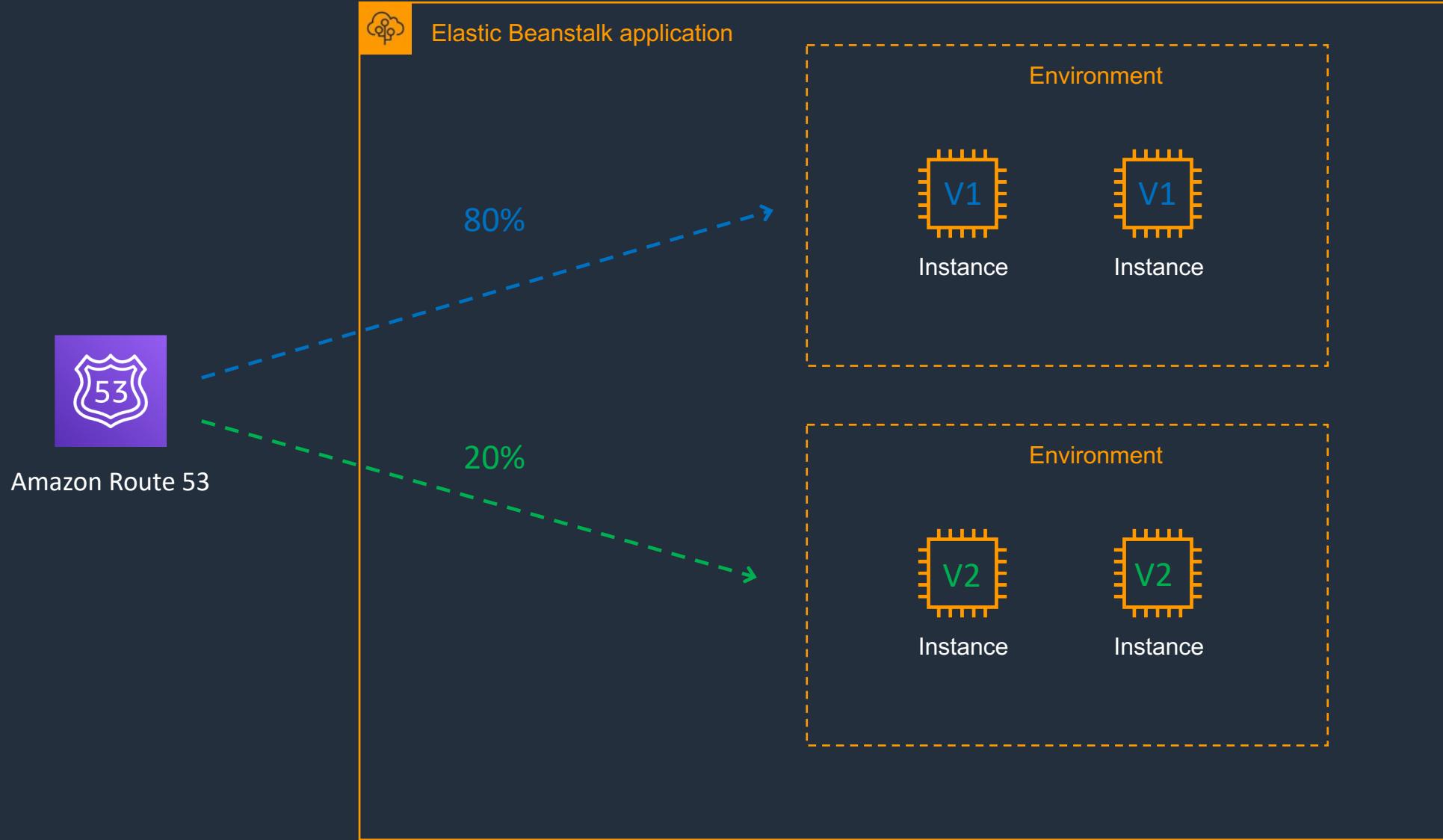
AWS Elastic Beanstalk – Immutable Update



AWS Elastic Beanstalk – Immutable Update

- Launches new instances in a new ASG and deploys the version update to these instances before swapping traffic to these instances once healthy
- Zero downtime
- New code is deployed to new instances using an ASG
- High cost as double the number of instances running during updates
- Longest deployment
- Quick rollback in case of failures
- Great for production environments

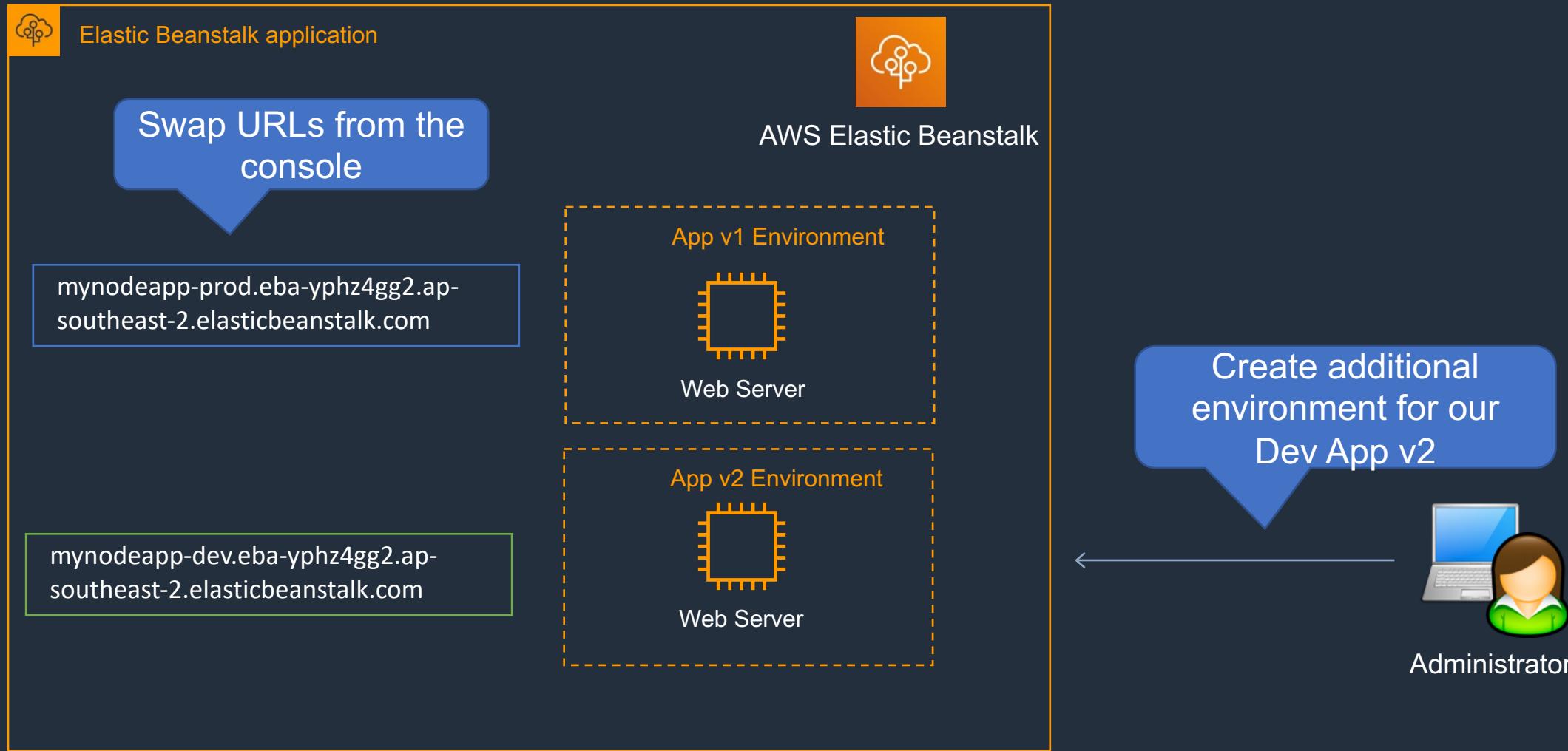
AWS Elastic Beanstalk – Blue/green



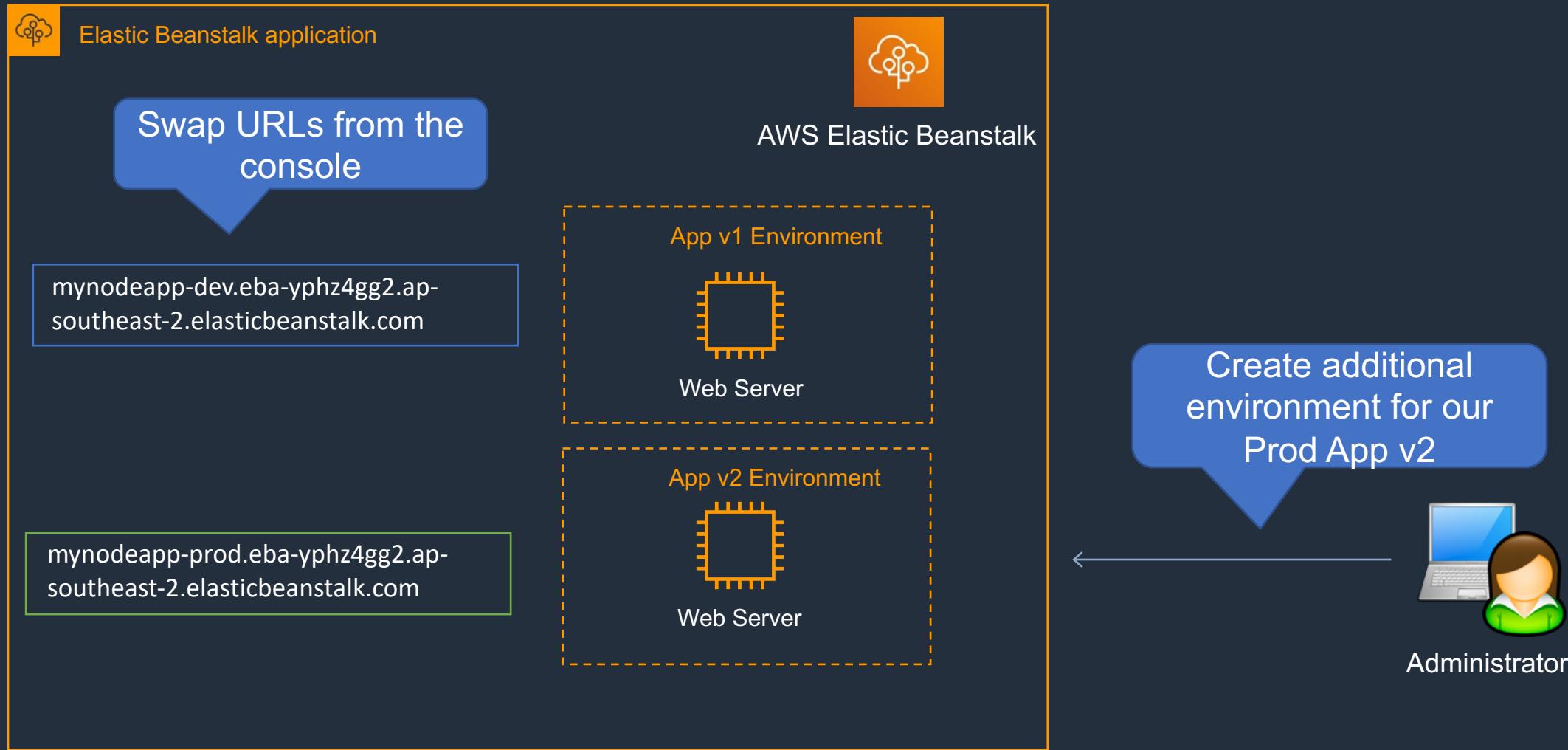
AWS Elastic Beanstalk – Blue/Green Update

- This is not a feature within Elastic Beanstalk
- You create a new "staging" environment and deploy updates there
- The new environment (green) can be validated independently and you can roll back if there are issues
- Route 53 can be setup using weighted policies to redirect a percentage of traffic to the staging environment
- Using Elastic Beanstalk, you can "swap URLs" when done with the environment test
- Zero downtime

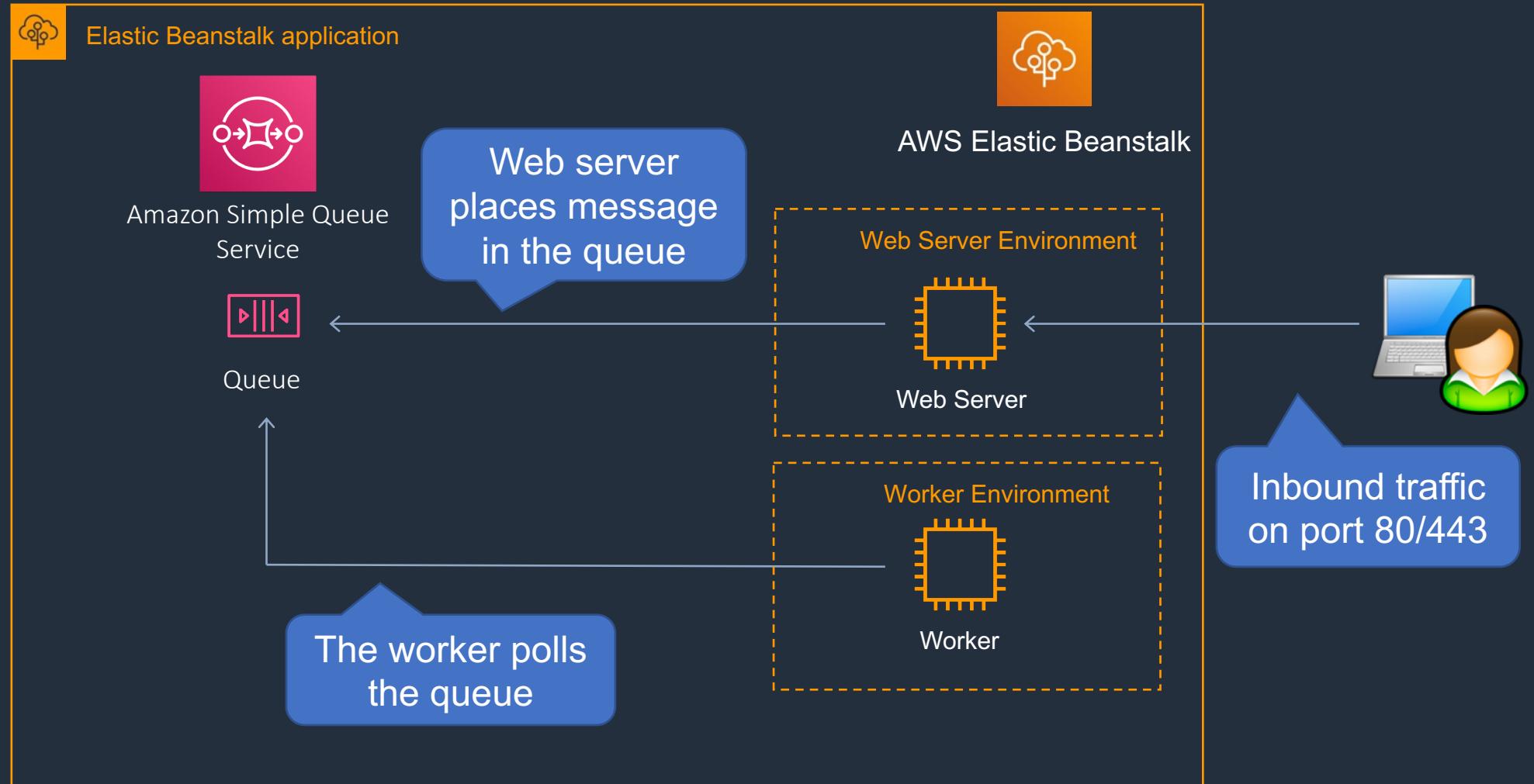
AWS Elastic Beanstalk - Multiple Environments



AWS Elastic Beanstalk - Multiple Environments



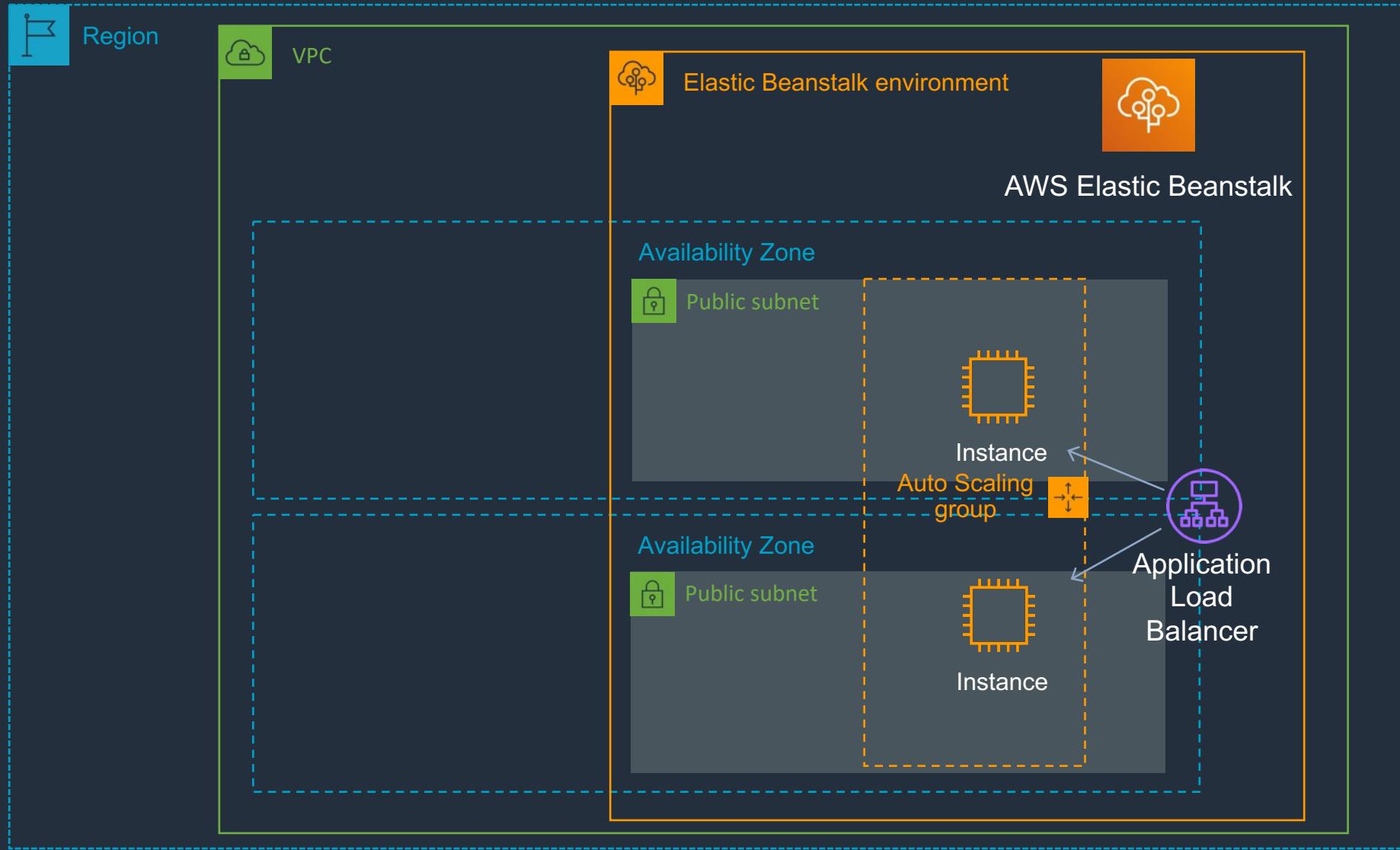
AWS Elastic Beanstalk Web Servers and Workers



AWS Elastic Beanstalk Environment Tiers

- Determines how Elastic Beanstalk provisions resources based on what the application is designed to do
- Consists of Web Servers and Workers:
 - Web servers are standard applications that listen for and then process HTTP requests, typically over port 80
 - Workers are specialized applications that have a background processing task that listens for messages on an Amazon SQS queue.
- Workers should be used for long-running tasks

AWS Elastic Beanstalk – High Availability



SECTION 8

Infrastructure Automation: AWS CloudFormation

AWS CloudFormation

- AWS CloudFormation is a service that allows you to manage, configure and provision your AWS infrastructure as code
- AWS CloudFormation provides a common language for you to describe and provision all the infrastructure resources in your cloud environment
- Resources are defined using a CloudFormation template

```
1  AWSTemplateFormatVersion: 2010-09-09
2  Description: >-
3  AWS CloudFormation Sample Template LAMP_Single_Instance: Create a LAMP stack
4  using a single EC2 instance and a local MySQL database for storage.
5  Parameters:
6    KeyName:
7      Description: Name of an existing EC2 KeyPair to enable SSH access to the instance
8      Type: 'AWS::EC2::KeyPair::KeyName'
9      ConstraintDescription: must be the name of an existing EC2 KeyPair.
10   DBName:
11     Default: MyDatabase
12     Description: MySQL database name
13     Type: String
14     MinLength: '1'
15     MaxLength: '64'
16     AllowedPattern: '[a-zA-Z][a-zA-Z0-9]*'
17     ConstraintDescription: must begin with a letter and contain only alphanumeric characters.
18   DBUser:
19     NoEcho: 'true'
20     Description: Username for MySQL database access
21     Type: String
22     MinLength: '1'
23     MaxLength: '16'
24     AllowedPattern: '[a-zA-Z][a-zA-Z0-9]*'
25     ConstraintDescription: must begin with a letter and contain only alphanumeric characters.
26   DBPassword:
27     NoEcho: 'true'
28     Description: Password for MySQL database access
29     Type: String
```

AWS CloudFormation

- CloudFormation can be used to provision a broad range of AWS resources
- Think of CloudFormation as deploying infrastructure as code

```
1 "AWSTemplateFormatVersion" : "2010-09-09",
2
3 "Description" : "AWS CloudFormation Sample Template WordPress_Multi_AZ: WordPress is web
4
5 "Parameters" : {
6     "VpcId" : {
7         "Type" : "AWS::EC2::VPC::Id",
8         "Description" : "VpcId of your existing Virtual Private Cloud (VPC)",
9         "ConstraintDescription" : "must be the VPC Id of an existing Virtual Private Cloud."
10    },
11
12 "Subnets" : {
13     "Type" : "List<AWS::EC2::Subnet::Id>",
14     "Description" : "The list of SubnetIds in your Virtual Private Cloud (VPC)",
15     "ConstraintDescription" : "must be a list of at least two existing subnets associated
16    },
17 }
```



AWS CloudFormation – Key Benefits

- Infrastructure is provisioned consistently, with fewer mistakes (human error)
- Less time and effort than configuring resources manually
- You can use version control and peer review for your CloudFormation templates

Free to use (you're only charged for the resources provisioned)

- Can be used to manage updates and dependencies
- Can be used to rollback and delete the entire stack as well

AWS CloudFormation – Key Concepts

Component	Description
Templates	The JSON or YAML text file that contains the instructions for building out the AWS environment
Stacks	The entire environment described by the template and created, updated, and deleted as a single unit
StackSets	AWS CloudFormation StackSets extends the functionality of stacks by enabling you to create, update, or delete stacks across multiple accounts and regions with a single operation
Change Sets	A summary of proposed changes to your stack that will allow you to see how those changes might impact your existing resources before implementing them

AWS CloudFormation – Templates

- A template is a YAML or JSON template used to describe the end-state of the infrastructure you are either provisioning or changing
- After creating the template, you upload it to CloudFormation directly or using Amazon S3
- CloudFormation reads the template and makes the API calls on your behalf.
- The resulting resources are called a "Stack"
- Logical IDs are used to reference resources within the template
- Physical IDs identify resources outside of AWS CloudFormation templates, but only after the resources have been created

AWS CloudFormation – Stacks

- Deployed resources based on templates
- Create, update and delete stacks using templates
- Deployed through the Management Console, CLI or APIs
- Stack creation errors:
 - Automatic rollback on error is enabled by default
 - You will be charged for resources provisioned even if there is an error

AWS CloudFormation – StackSets

- AWS CloudFormation StackSets extends the functionality of stacks by enabling you to create, update, or delete stacks across multiple accounts and regions with a single operation
- Using an administrator account, you define and manage an AWS CloudFormation template, and use the template as the basis for provisioning stacks into selected target accounts across specified regions
- An administrator account is the AWS account in which you create stack sets.
- A stack set is managed by signing into the AWS administrator account in which it was created
- A target account is the account into which you create, update, or delete one or more stacks in your stack set

Intrinsic Functions - Ref

- The intrinsic function Ref returns the value of the specified parameter or resource
- When you specify a parameter's logical name, it returns the value of the parameter
- When you specify a resource's logical name, it returns a value that you can typically use to refer to that resource, such as a physical ID
- The following resource declaration for an Elastic IP address needs the instance ID of an EC2 instance and uses the Ref function to specify the instance ID of the MyEC2Instance resource:

```
MyEIP:  
  Type: "AWS::EC2::EIP"  
  
  Properties:  
    InstanceId: !Ref MyEC2Instance
```

Intrinsic Functions - Fn::FindInMap

- The intrinsic function Fn::FindInMap returns the value corresponding to keys in a two-level map that is declared in the Mappings section
- Full syntax (YAML): Fn::FindInMap: [MapName, TopLevelKey, SecondLevelKey]
- Short form (YAML): !FindInMap [MapName, TopLevelKey, SecondLevelKey]

Intrinsic Functions - Fn::FindInMap

- The following example shows how to use Fn::FindInMap for a template with a Mappings section that contains a single map, RegionMap, that associates AMIs with AWS regions:

```
Mappings:  
  RegionMap:  
    us-east-1:  
      HVM64: "ami-0ff8a91507f77f867"  
      HVMG2: "ami-0a584ac55a7631c0c"  
    us-west-1:  
      HVM64: "ami-0bdb828fd58c52235"  
      HVMG2: "ami-066ee5fd4a9ef77f1"  
  
Resources:  
  myEC2Instance:  
    Type: "AWS::EC2::Instance"  
    Properties:  
      ImageId: !FindInMap  
        - RegionMap  
        - !Ref 'AWS::Region'  
        - HVM64  
      InstanceType: m1.small
```

AWS CloudFormation – Resources

- Resources - the required Resources section declares the AWS resources that you want to include in the stack, such as an Amazon EC2 instance or an Amazon S3 bucket
- Mandatory
- Resources are declared and can reference each other

```
Resources:  
MyEC2Instance:  
  Type: "AWS::EC2::Instance"  
Properties:  
  ImageId: "ami-0ff8a91507f77f867"
```

AWS CloudFormation – Parameters

- Parameters – use the optional Parameters section to customize your templates
- Parameters enable you to input custom values to your template each time you create or update a stack
- Useful for template reuse

```
Parameters:  
  InstanceTypeParameter:  
    Type: String  
    Default: t2.micro  
    AllowedValues:  
      - t2.micro  
      - m1.small  
      - m1.large  
    Description: Enter t2.micro, m1.small, or m1.large. Default is t2.micro.
```

AWS CloudFormation – Mappings

- Mappings – the optional Mappings section matches a key to a corresponding set of named values

```
RegionMap:  
  us-east-1:  
    HVM64: ami-0ff8a91507f77f867  
    HVMG2: ami-0a584ac55a7631c0c  
  us-west-1:  
    HVM64: ami-0bdb828fd58c52235  
    HVMG2: ami-066ee5fd4a9ef77f1
```

- Exam tip: with mappings you can, for example, set values based on a region You can create a mapping that uses the region name as a key and contains the values you want to specify for each specific region

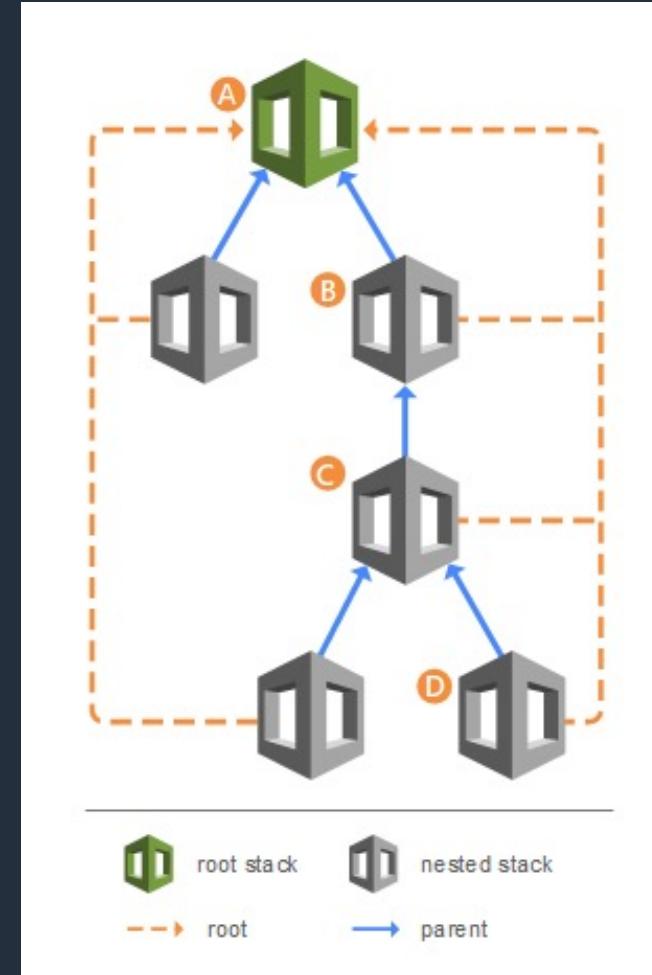
AWS CloudFormation – Outputs

- Outputs – the optional Outputs section declares output values that you can import into other stacks (to create cross-stack references), return in response (to describe stack calls), or view on the AWS CloudFormation console
- In the following example YAML code, the output named StackVPC returns the ID of a VPC, and then exports the value for cross-stack referencing with the name VPCID appended to the stack's name

```
Outputs:  
  StackVPC:  
    Description: The ID of the VPC  
    Value: !Ref MyVPC  
    Export:  
      Name: !Sub "${AWS::StackName}-VPCID"
```

AWS CloudFormation – Nested Stacks

- Nested stacks allow re-use of CloudFormation code for common use cases
- For example standard configuration for a load balancer, web server, application server etc.
- Instead of copying out the code each time, create a standard template for each common use case and reference from within your CloudFormation template



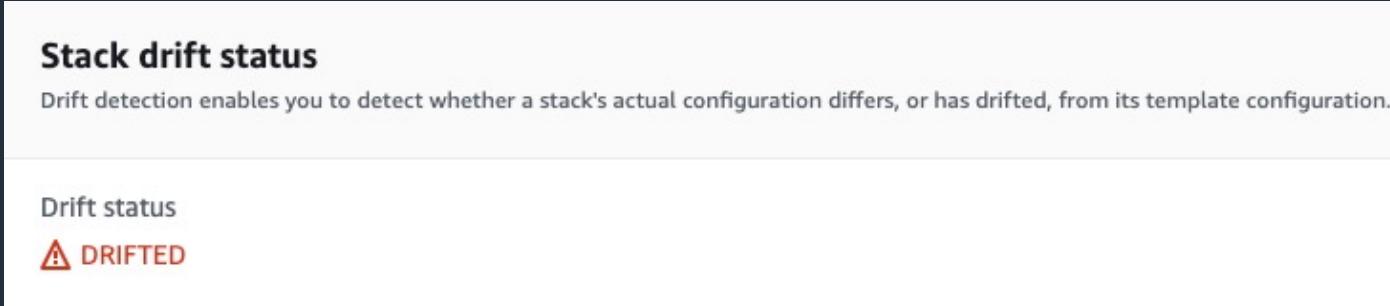
AWS CloudFormation – Change Sets

- AWS CloudFormation provides two methods for updating stacks:
direct update or creating and executing change sets
- When you directly update a stack, you submit changes and AWS CloudFormation immediately deploys them
- Use direct updates when you want to quickly deploy your updates
- With change sets, you can preview the changes AWS CloudFormation will make to your stack, and then decide whether to apply those changes

Action	Logical ID
Modify	MyEC2Instance
Remove	MyEIP
Remove	MySecurityGroup

AWS CloudFormation – Drift Detection

- Drift detection enables detects whether a stack's actual configuration differs, or has drifted, from its expected configuration



- You can perform drift detection on stacks with the following statuses: CREATE_COMPLETE, UPDATE_COMPLETE, UPDATE_ROLLBACK_COMPLETE, and UPDATE_ROLLBACK_FAILED

Type	Drift status
AWS::EC2::Instance	⚠ MODIFIED
AWS::EC2::EIP	✓ IN_SYNC
AWS::EC2::SecurityGroup	✓ IN_SYNC

AWS CloudFormation – UserData Property

- User data can be included in a CloudFormation template
- The script is passed into Fn::Base64
- The user data script logs are stored in /var/log/cloud-init-output.log

```
MyEC2Instance:  
  Type: AWS::EC2::Instance  
  Properties:  
    UserData:  
      Fn::Base64: |  
        #!/bin/bash  
        yum update -y  
        yum install httpd -y  
        systemctl start httpd  
        systemctl enable httpd  
        cd /var/www/html  
        echo "This EC2 instance is running successfully!" > index.html
```

AWS CloudFormation – Helper Scripts (cfn-init)

The cfn-init helper script reads template metadata from the AWS::CloudFormation::Init key and acts accordingly to:

- Fetch and parse metadata from AWS CloudFormation
- Install packages
- Write files to disk
- Enable/disable and start/stop services
- Logs go to /var/log/cfn-init.log

AWS CloudFormation – Helper Scripts (cfn-init)

- To install the applications the UserData property and Metadata property can be added to a template

```
"Properties": {  
    "ImageId"      : "ami-02354e95b39ca8dec",  
    "InstanceType"  : { "Ref" : "InstanceType" },  
    "SecurityGroups": [ { "Ref" : "WebServerSecurityGroup" } ],  
    "KeyName"       : { "Ref" : "KeyName" },  
    "UserData"      : { "Fn::Base64" : { "Fn::Join" : [ "", [  
        "#!/bin/bash -xe\n",  
        "yum install -y aws-cfn-bootstrap\n",  
  
        "# Install the files and packages from the metadata\n",  
        "/opt/aws/bin/cfn-init ",  
        "  --stack ", { "Ref" : "AWS::StackName" },  
        "  --resource WebServerInstance ",  
        "  --configsets InstallAndRun ",  
        "  --region ", { "Ref" : "AWS::Region" }, "\n" ] ] }  
}
```

AWS CloudFormation – Helper Scripts (cfn-signal)

- The cfn-signal helper script signals AWS CloudFormation to indicate whether Amazon EC2 instances have been successfully created or updated
- After installing software on EC2 instances, you can signal AWS CloudFormation when those software applications are ready
- You use the cfn-signal script in conjunction with a CreationPolicy or an Auto Scaling group with a WaitOnResourceSignals update policy

AWS CloudFormation – Helper Scripts (cfn-signal)

- In the UserData property, the template runs the cfn-signal script to send a success signal with an exit code if all the services are configured and started successfully

```
"# Install the files and packages from the metadata\n",
"/opt/aws/bin/cfn-init ",
"    --stack ", { "Ref" : "AWS::StackName" },
"    --resource WebServerInstance ",
"    --configsets InstallAndRun ",
"    --region ", { "Ref" : "AWS::Region" }, "\n"
"# Signal the status from cfn-init\n",
"/opt/aws/bin/cfn-signal -e $? ",
"    --stack ", { "Ref" : "AWS::StackName" },
"    --resource WebServerInstance ",
"    --region ", { "Ref" : "AWS::Region" }, "\n"
```

AWS CloudFormation – Helper Scripts (cfn-init and cfn-signal)

Troubleshooting errors:

- Make sure the AMI has the CloudFormation helper scripts included
- Check that the cfn-init and cfn-signal commands have run successfully

AWS CloudFormation – CreationPolicy

- Use the `CreationPolicy` attribute when you want to wait on resource configuration actions before stack creation proceeds
- You can associate the `CreationPolicy` attribute with a resource to prevent its status from reaching `create complete` until AWS CloudFormation receives a specified number of success signals or the timeout period is exceeded.
- To signal a resource, you can use the `cfn-signal` helper script or `SignalResource` API
- AWS CloudFormation publishes valid signals to the stack events so that you track the number of signals sent

AWS CloudFormation – CreationPolicy

The following CloudFormation resources support creation policies:

- AWS::AutoScaling::AutoScalingGroup
- AWS::EC2::Instance
- AWS::CloudFormation::WaitCondition

AWS CloudFormation –DeletionPolicy

- With the DeletionPolicy attribute you can preserve or (in some cases) backup a resource when its stack is deleted.
- You specify a DeletionPolicy attribute for each resource that you want to control
- If a resource has no DeletionPolicy attribute, AWS CloudFormation deletes the resource by default
- Deletion policies can be specified as:
 - DeletionPolicy=Retain – preserves the resources
 - DeletionPolicy=Snapshot – takes a snapshot (e.g. for EC2, ElastiCache, RDS)
 - DeletionPolicy=Delete – default, attempts to delete the resources

AWS CloudFormation – DependsOn

- With the DependsOn attribute you can specify that the creation of a specific resource follows another
- When you add a DependsOn attribute to a resource, that resource is created only after the creation of the resource specified in the DependsOn attribute

AWS CloudFormation – WaitCondition

- Use a WaitCondition to ensure resources are ready
- You can use a wait condition for situations like the following:
 - To coordinate stack resource creation with configuration actions that are external to the stack creation
 - To track the status of a configuration process
- Note: For Amazon EC2 and Auto Scaling resources, AWS recommends that you use a CreationPolicy attribute instead of wait conditions

AWS CloudFormation – UpdatePolicy and UpdateReplacePolicy

- Use the `UpdatePolicy` attribute to specify how AWS CloudFormation handles updates to the following resources:
 - `AWS::AutoScaling::AutoScalingGroup`,
 - `AWS::ElasticCache::ReplicationGroup`
 - `AWS::Elasticsearch::Domain`
 - `AWS::Lambda::Alias`
- Use the `UpdateReplacePolicy` attribute to retain or (in some cases) backup the existing physical instance of a resource when it is replaced during a stack update operation

AWS CloudFormation – Rollbacks and Stack Creation Failures

Stack creation failures:

- By default everything will be deleted
- Can modify the OnFailure attribute for a stack
- OnFailure must be one of:
 - DO_NOTHING – leaves the resources in place (good for troubleshooting)
 - ROLLBACK – rolls the stack back
 - DELETE – deletes the resources

AWS CloudFormation – Rollbacks and Stack Creation Failures

Stack update failures:

- A stack goes into the UPDATE_ROLLBACK_FAILED state when AWS CloudFormation cannot roll back all changes during an update
- The stack will automatically roll back to the previous known working state
- When a stack is in the UPDATE_ROLLBACK_FAILED state, you can continue to roll it back to a working state (UPDATE_ROLLBACK_COMPLETE)
- You can't update a stack that is in the UPDATE_ROLLBACK_FAILED state
- However, if you can continue to roll it back, you can return the stack to its original settings and then try to update it again

Exam Scenarios

Exam Scenario	Solution
Need to review updates to a CloudFormation stack before deploying them in production	Use change sets
Stack deployed and manual changes were made. Need to capture changes and update template	Use drift detection and use output to update template and redeploy the stack
Need to update new version of app on EC2 and ALB. Must avoid DNS changes and be able to rollback	Update template with AutoScalingReplacingUpdate policy and perform an update
Need to write a single template that can be deployed across several environments / Region	Use parameters to enter custom values and use Ref intrinsic function to reference the parameter
Tried to launch instance in a different region from a working template and it fails	Probably due to incorrect AMI ID

Exam Scenarios

Exam Scenario	Solution
CloudFormation stack created for first time and fails with ROLLBACK_COMPLETE status	To continue administrator must relaunch the template to create a new stack
Template for infrastructure in one region used to deploy in another and fails	Template likely referenced an AMI that doesn't exist in the new region and/or services that don't exist
CloudFormation stack fails and returns UPDATE_ROLLBACK_FAILED	Fix the error that caused the rollback to fail and then select "Continue update rollback" in the console
Need to deploy a single CloudFormation template across multiple accounts	Use StackSets
CloudFormation deploys stack with separate VPC for each app. Fails to deploy	May have reached the default limit for VPCs in the account

Exam Scenarios

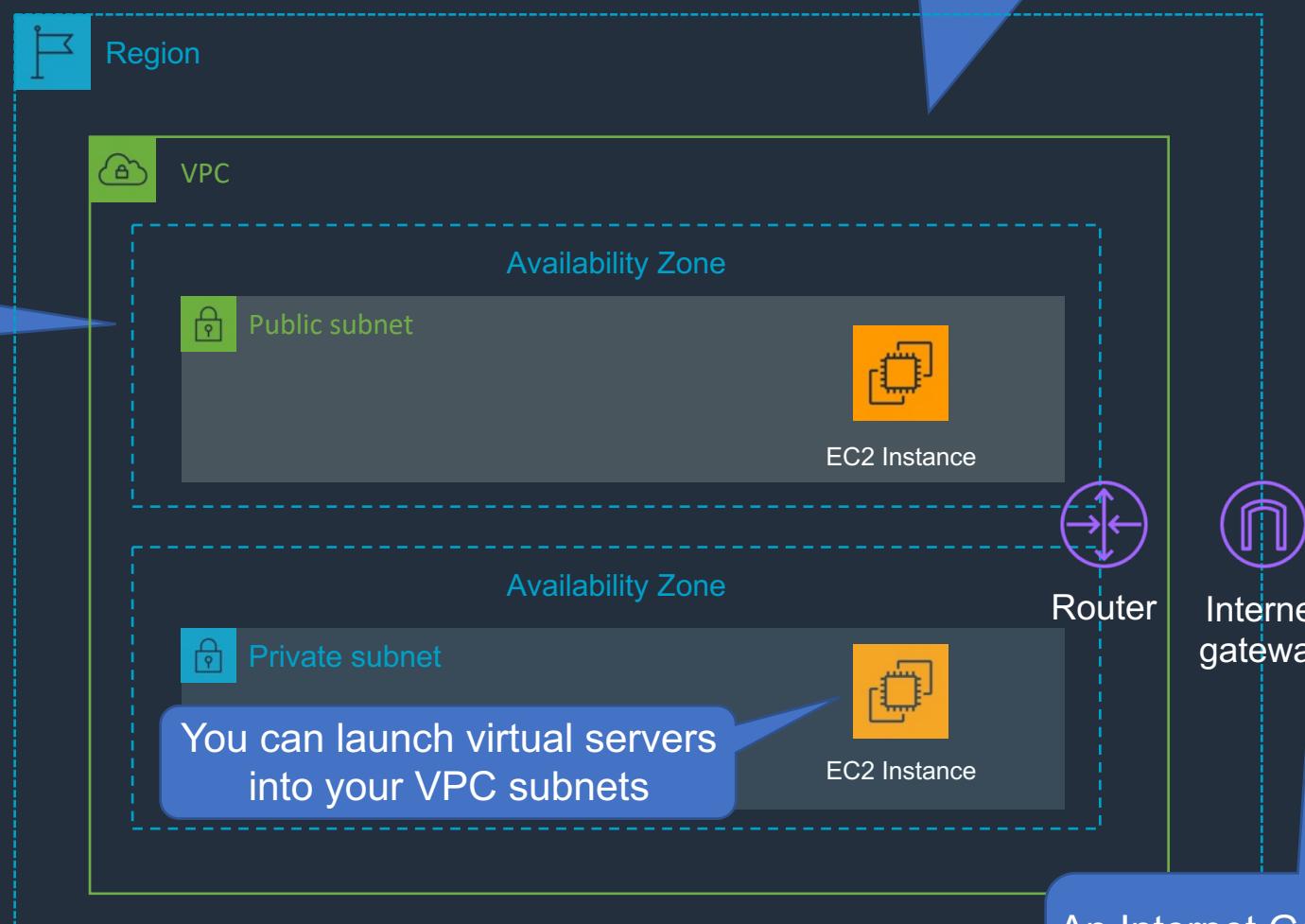
Exam Scenario	Solution
Would like to manually address any issues with CloudFormation stack creation	Set the OnFailure parameter to "DO NOTHING"
CloudFormation fails with "The image id '[ami-2a69aa47]' does not exist"	Most likely the template is being run in a different region where the AMI does not exist
When creating Stack a wait condition error is experienced: ""received 0 signals out of the 1 expected from the EC2 instance".	Check instance has a route through NAT device and in the cfn logs confirm that the cfn-signal command ran successfully

SECTION 9

Networking: Amazon Virtual Private Cloud (VPC)

Amazon Virtual Private Cloud (VPC)

A VPC is a logically isolated portion of the AWS cloud within a region

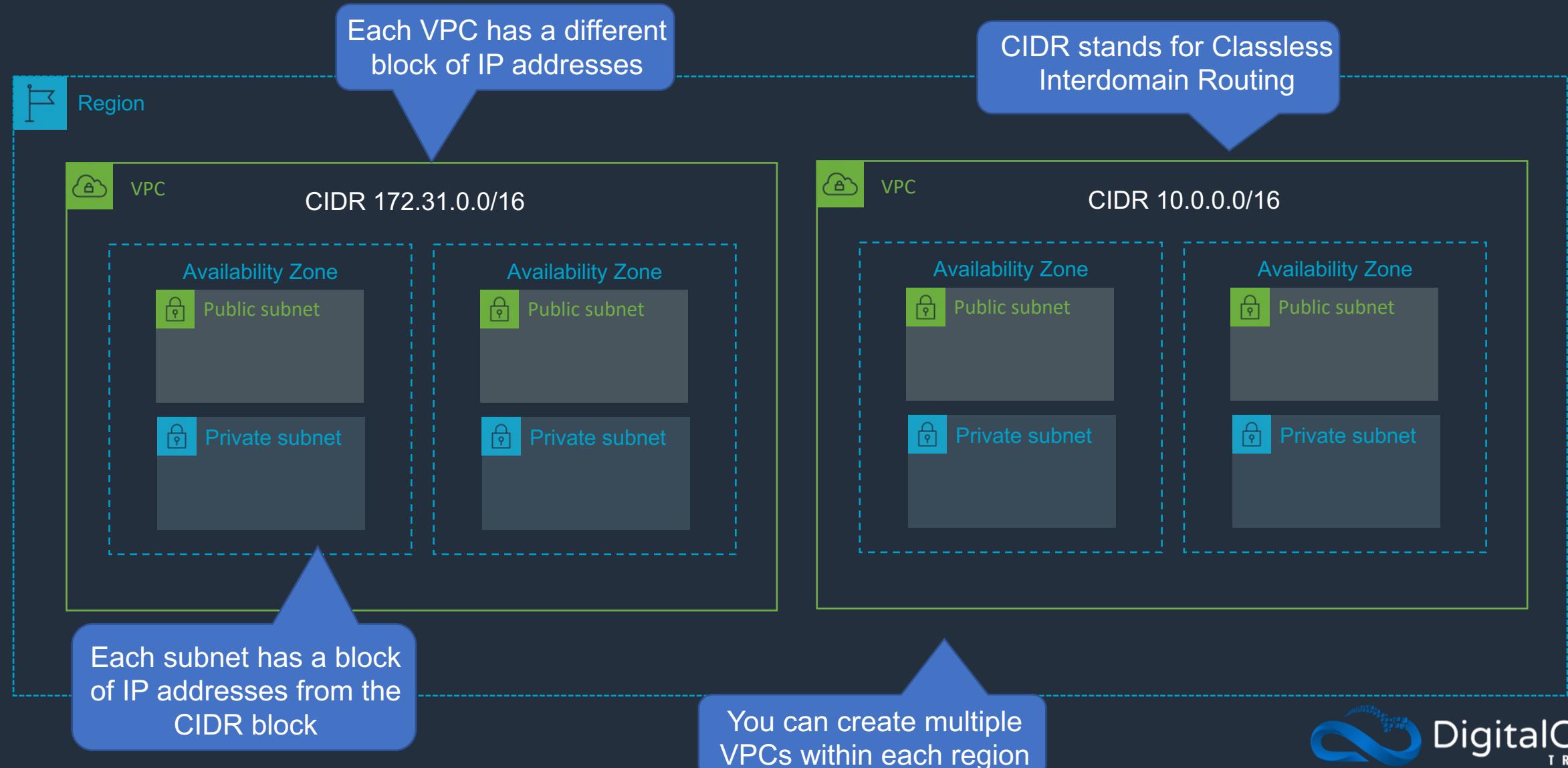


Main Route Table

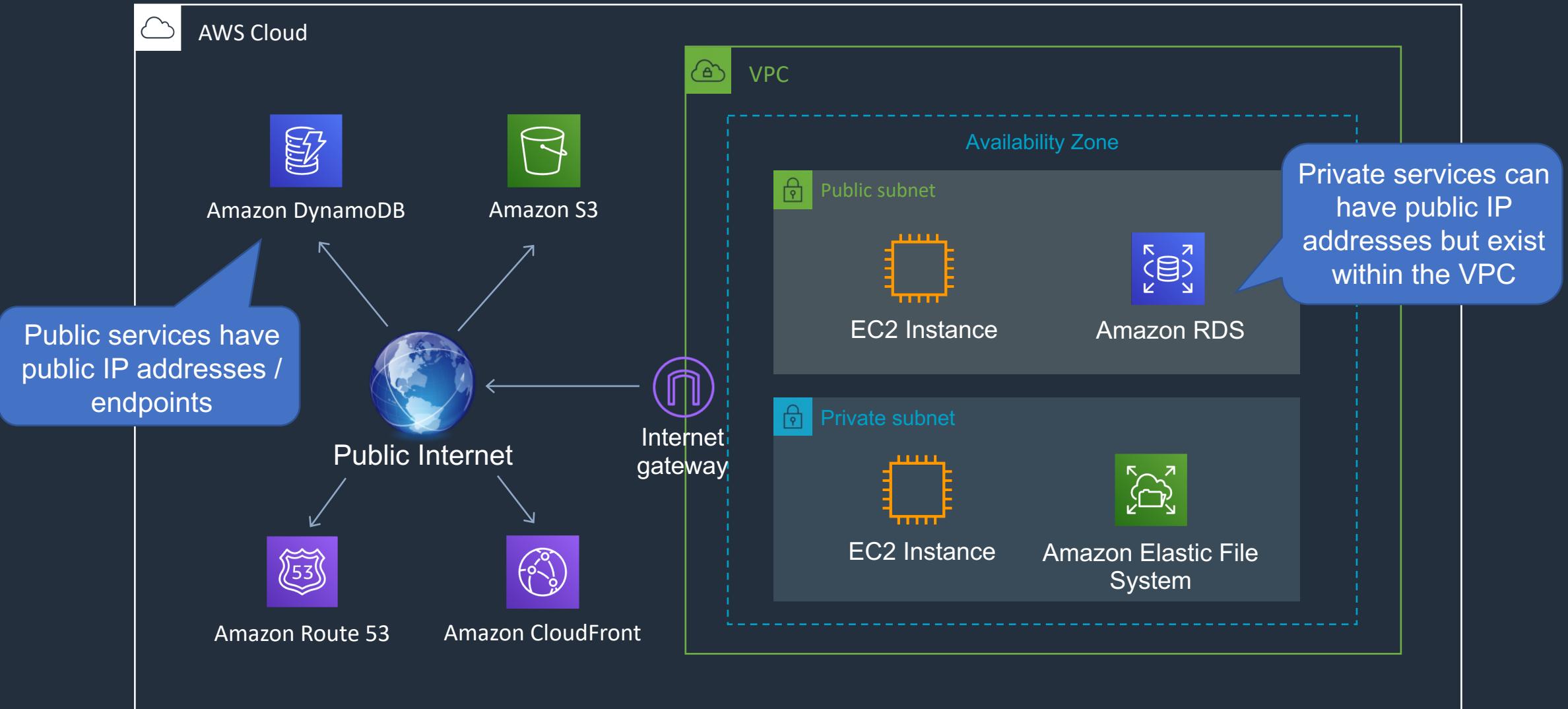
Destination	Target
172.31.0.0/16	Local
0.0.0.0/0	igw-id

The route table is used to configure the VPC router

Multiple VPCs



AWS Public and Private Services

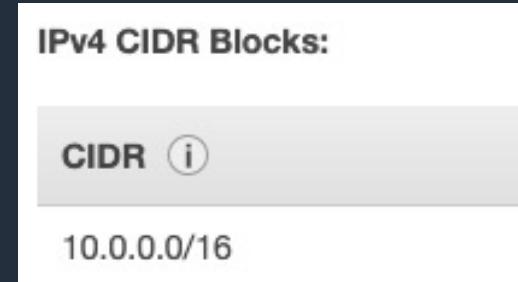


Availability Zone IDs (AZ ID)

- Availability Zones are mapped differently across accounts
- For example, **us-east-1a** may map to a different location across two different accounts
- To identify exactly where your resources are running, use the **AZ ID**
- For example, **us-east-1a** maps to the AZ ID **use1-az1**

Amazon VPC – CIDR Blocks and IP Subnets

- When you create a VPC, you must specify a range of IPv4 addresses for the VPC in the form of a Classless Inter-Domain Routing (CIDR) block



- You can then define ranges of IP addresses within the VPC CIDR that can be assigned to subnets. AWS resources obtain addresses from these IP ranges

IPv4 CIDR	Available IPv4 Addresses
10.0.1.0/24	251
10.0.2.0/24	251
10.0.3.0/24	251
10.0.4.0/24	251

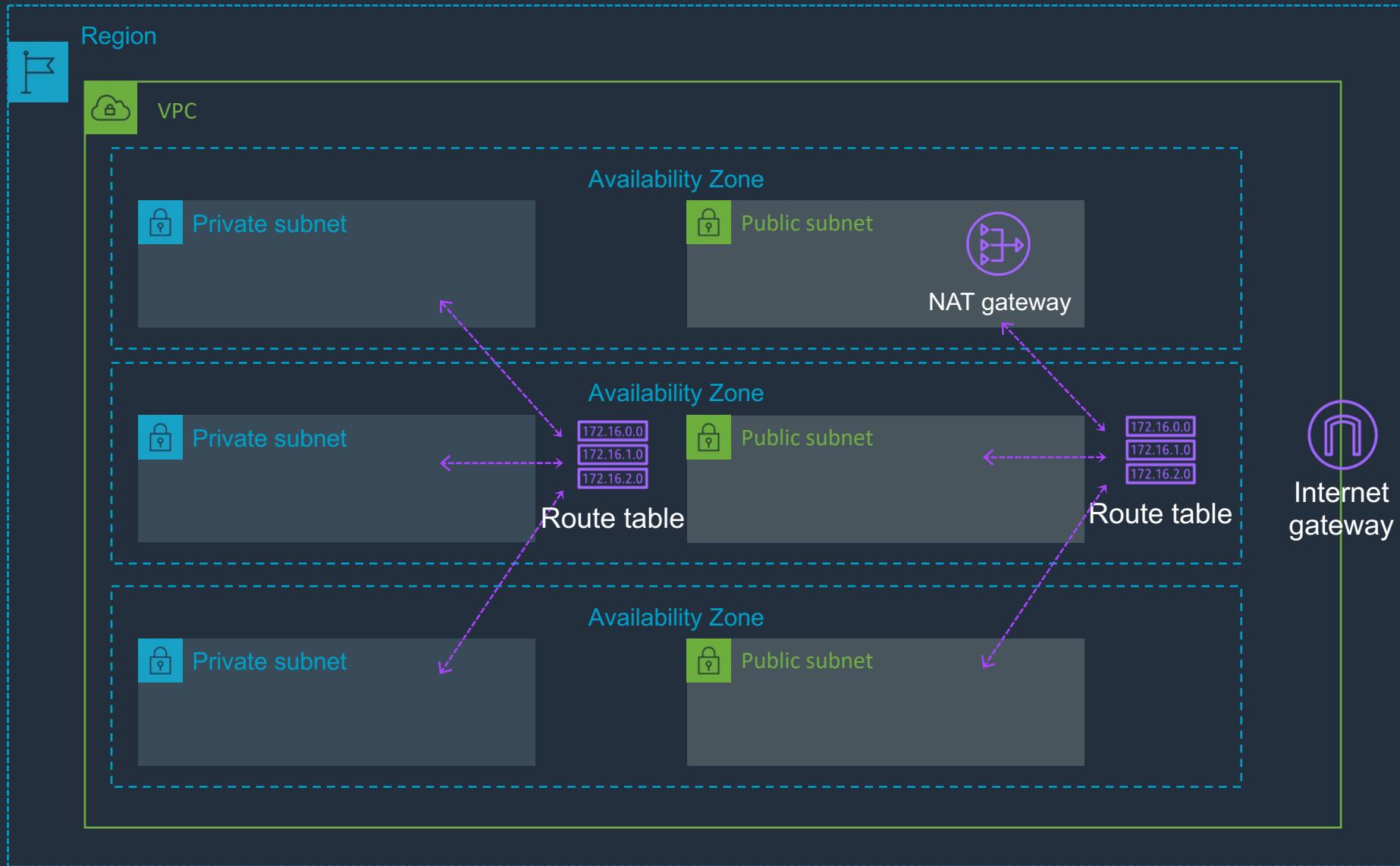
Amazon VPC – CIDR Blocks and IP Subnets

- AWS recommend that CIDR blocks of /16 or smaller are used
- It is recommended these come from the private IP ranges specified in RFC 1918
 - 10.0.0.0 - 10.255.255.255 (10/8 prefix)
 - 172.16.0.0 - 172.31.255.255 (172.16/12 prefix)
 - 192.168.0.0 - 192.168.255.255 (192.168/16 prefix)
- However, it is possible to create a VPC with publicly routable CIDR block
- The allowed block size is between a /28 netmask and /16 netmask
- The CIDR blocks of the subnets within a VPC cannot overlap

Amazon VPC – CIDR Blocks and IP Subnets

- The first four IP addresses and the last IP address in each subnet CIDR block are not available for you to use
- For example, in a subnet with CIDR block 10.0.0.0/24, the following five IP addresses are reserved:
 - 10.0.0.0: Network address
 - 10.0.0.1: Reserved by AWS for the VPC router
 - 10.0.0.2: Reserved by AWS
 - 10.0.0.3: Reserved by AWS for future use
 - 10.0.0.255: Network broadcast address (broadcast not supported)

Creating a Custom VPC



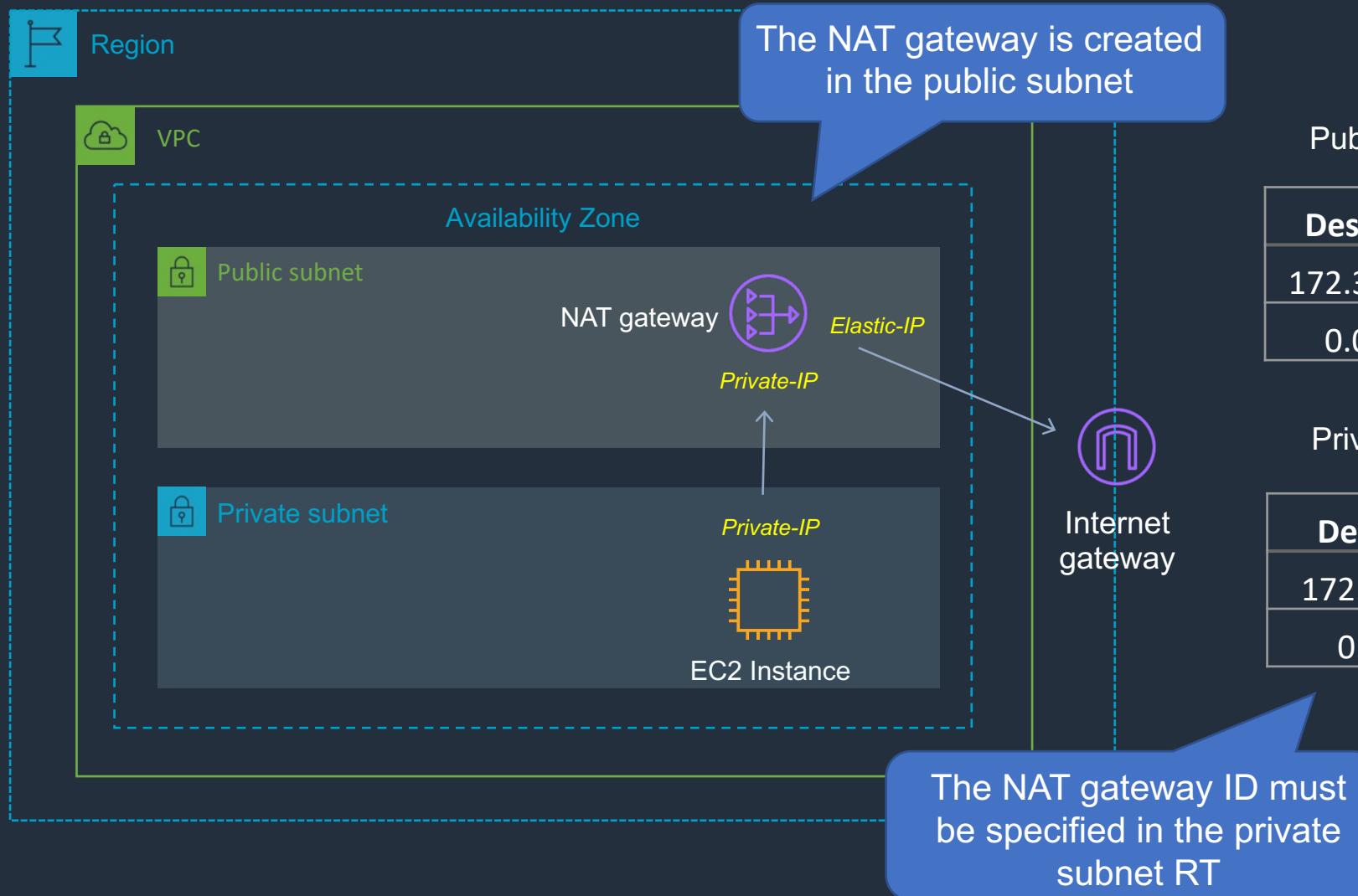
Public Route Table

Destination	Target
10.0.0.0/16	Local
0.0.0.0/0	<i>igw-id</i>

Private Route Table

Destination	Target
10.0.0.0/16	Local
0.0.0.0/0	<i>nat-gateway-id</i>

Private Subnet with NAT Gateway



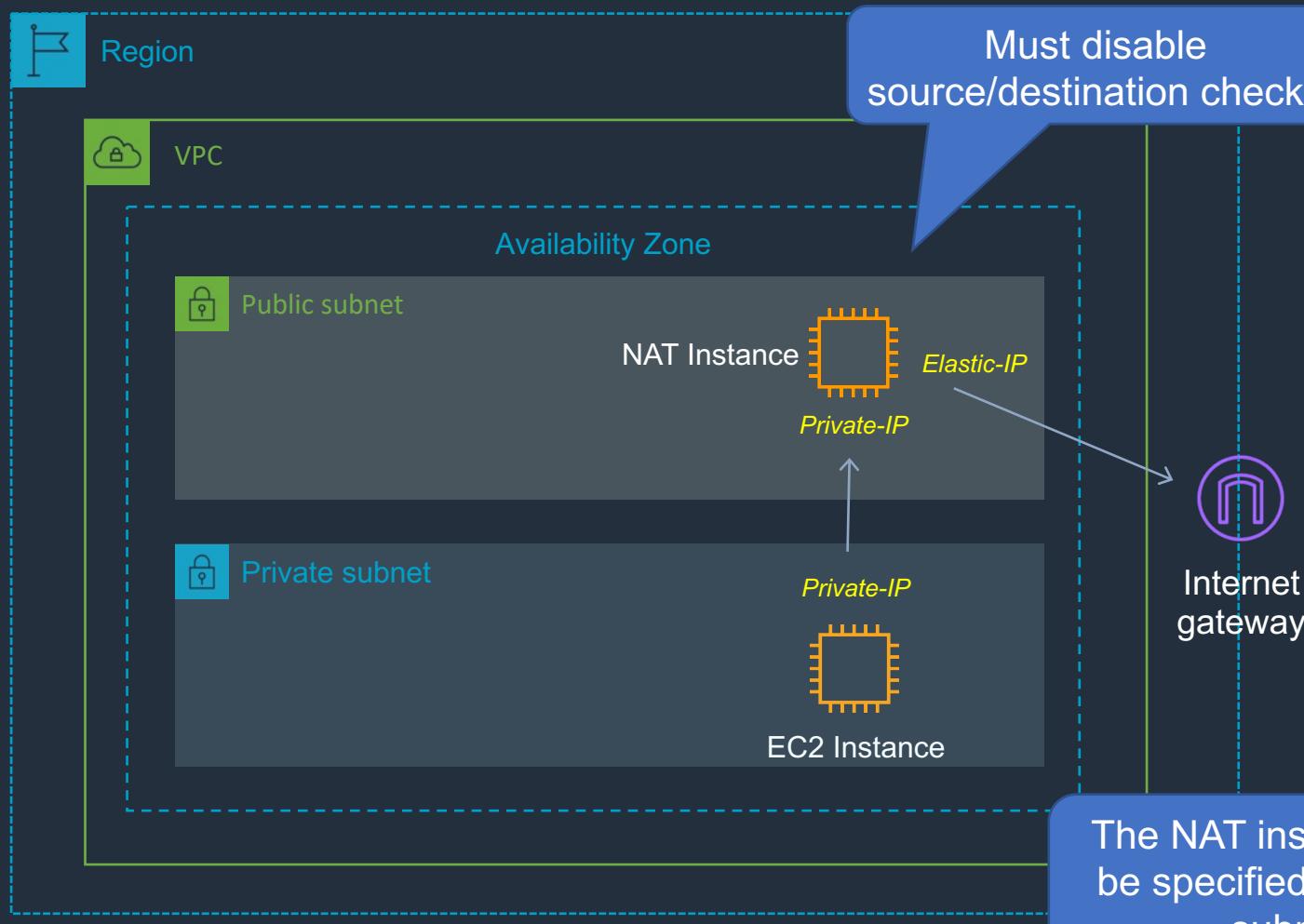
Public Subnet Route Table

Destination	Target
172.31.0.0/16	Local
0.0.0.0/0	<i>igw-id</i>

Private Subnet Route Table

Destination	Target
172.31.0.0/16	Local
0.0.0.0/0	<i>nat-gateway-id</i>

Private Subnet with NAT Instance



Public Subnet Route Table

Destination	Target
172.31.0.0/16	Local
0.0.0.0/0	<i>igw-id</i>

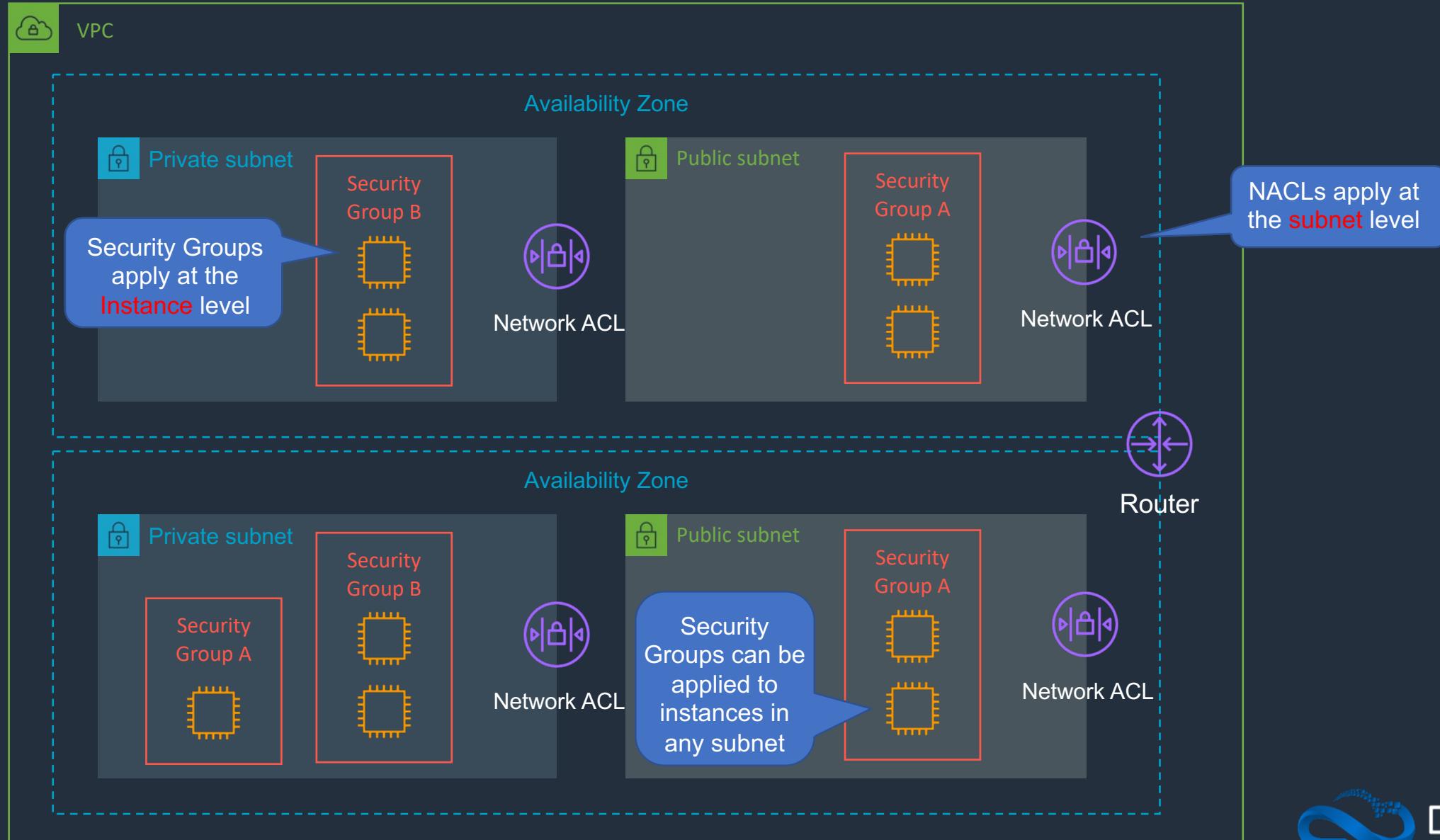
Private Subnet Route Table

Destination	Target
172.31.0.0/16	Local
0.0.0.0/0	<i>nat-instance-id</i>

NAT Instance vs NAT Gateway

NAT Instance	NAT Gateway
Managed by you (e.g. software updates)	Managed by AWS
Scale up (instance type) manually and use enhanced networking	Elastic scalability up to 45 Gbps
No high availability – scripted/auto-scaled HA possible using multiple NATs in multiple subnets	Provides automatic high availability within an AZ and can be placed in multiple AZs
Need to assign Security Group	No Security Groups
Can use as a bastion host	Cannot access through SSH
Use an Elastic IP address or a public IP address with a NAT instance	Choose the Elastic IP address to associate with a NAT gateway at creation
Can implement port forwarding through manual customisation	Does not support port forwarding

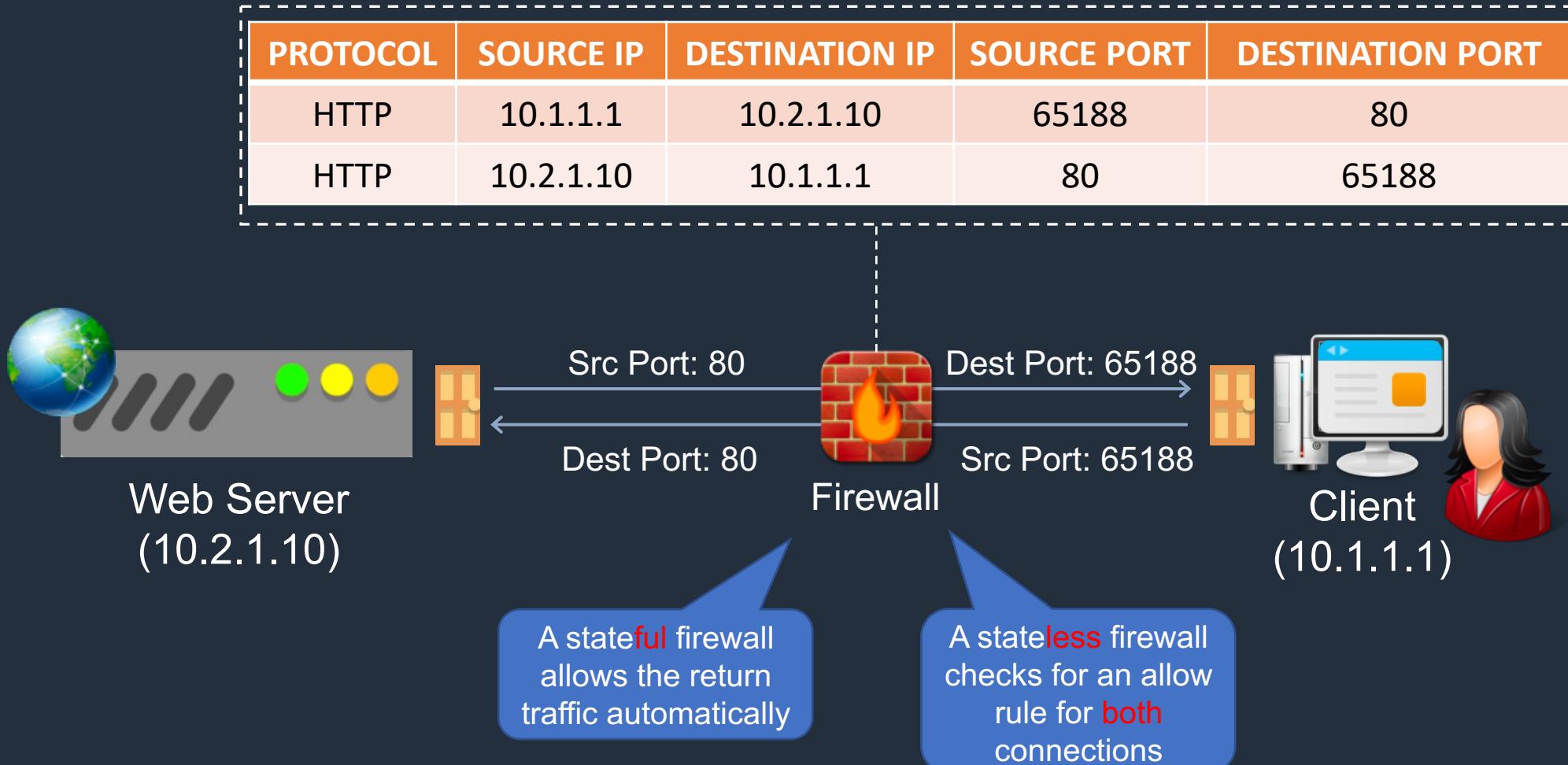
Security Groups & Network Access Control Lists (NACLs)



Security Groups

- Security groups act like a firewall at the instance level
- Specifically security groups operate at the network interface level
- Can only assign permit/allow rules in a security group,
- You cannot assign deny rules
- There is an implicit deny rule at the end of the security group
- All rules are evaluated until a permit is encountered or continues until the implicit deny
- Can control ingress and egress traffic with security groups
- Security groups are stateful

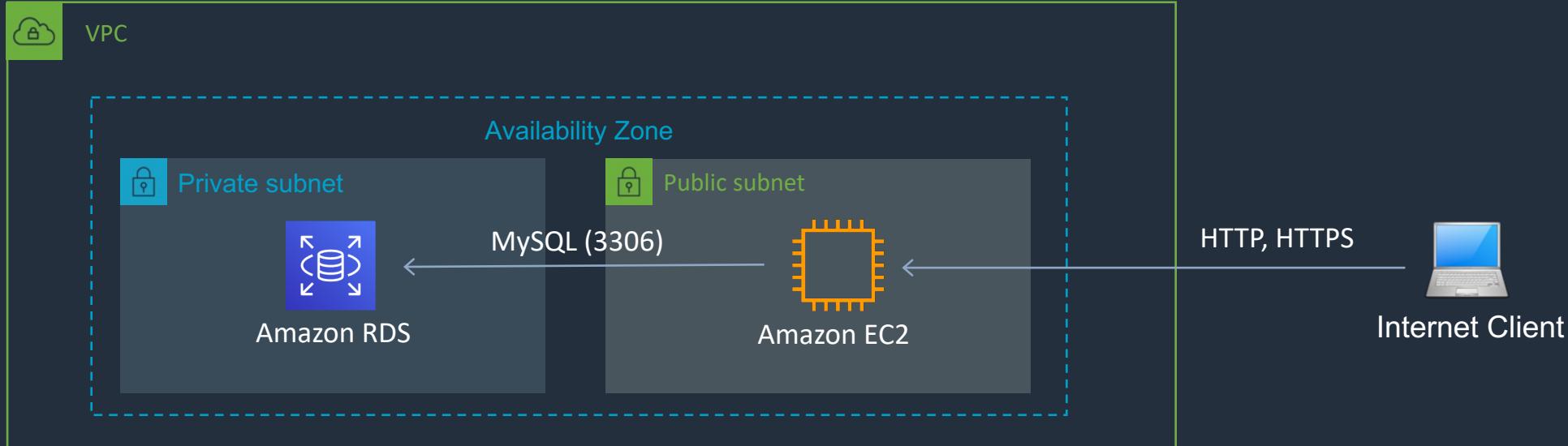
Stateful (Security Groups) vs Stateless (Network ACLs) Firewalls



Security Groups

- You can use security group names/IDs as the source or destination in other security groups
- You can use the security group name/ID as a source in its own inbound rules
- Security group members can be within any AZ or subnet within the VPC
- Security group membership can be changed whilst instances are running
- Any changes made will take effect immediately
- Up to 5 security groups can be added per EC2 instance interface.
- There is no limit on the number of EC2 instances within a security group.
- You cannot block specific IP addresses using security groups, use NACLs instead

Security Group Best Practice



Security group (DBSG)

Inbound

Type	Protocol	Port	Source
MySQL	TCP	3306	WebSG

Security group (WebSG)

Inbound

Type	Protocol	Port	Source
HTTP	TCP	80	0.0.0.0/0
HTTPS	TCP	443	0.0.0.0/0

Outbound

Type	Protocol	Port	Destination
MySQL	TCP	3306	DB-SG

Network Access Control Lists (NACLs)



Default NACL

Inbound:

Protocol	Port	Source	Action
All	All	0.0.0.0/0	ALLOW
All	All	::/0	ALLOW

Outbound:

Protocol	Port	Source	Action
All	All	0.0.0.0/0	ALLOW
All	All	::/0	ALLOW

Custom NACL

Inbound:

Protocol	Port	Source	Action
All	All	0.0.0.0/0	DENY
All	All	::/0	DENY

Outbound:

Protocol	Port	Source	Action
All	All	0.0.0.0/0	DENY
All	All	::/0	DENY

Network ACLs

- Network ACL's function at the subnet level
- With NACLs you can have permit and deny rules
- Network ACLs contain a numbered list of rules that are evaluated in order from the lowest number until the explicit deny
- Network ACLs have separate inbound and outbound rules and each rule can allow or deny traffic.
- Network ACLs are stateless so responses are subject to the rules for the direction of traffic.
- NACLs only apply to traffic that is ingress or egress to the subnet not to traffic within the subnet

Network ACLs

- Each subnet in your VPC must be associated with a network ACL. If you don't do this manually it will be associated with the default network ACL
- You can associate a network ACL with multiple subnets; however a subnet can only be associated with one network ACL at a time
- Network ACLs do not filter traffic between instances in the same subnet
- NACLs are the preferred option for blocking specific IPs or ranges
- Security groups cannot be used to block specific ranges of IPs
- NACL is the first line of defence, the security group is the second line

Security Groups & Network Access Control Lists (NACLs)

Security Group	Network ACL
Operates at the instance (interface) level	Operates at the subnet level
Supports allow rules only	Supports allow and deny rules
Stateful	Stateless
Evaluates all rules	Processes rules in order
Applies to an instance only if associated with a group	Automatically applies to all instances in the subnets its associated with

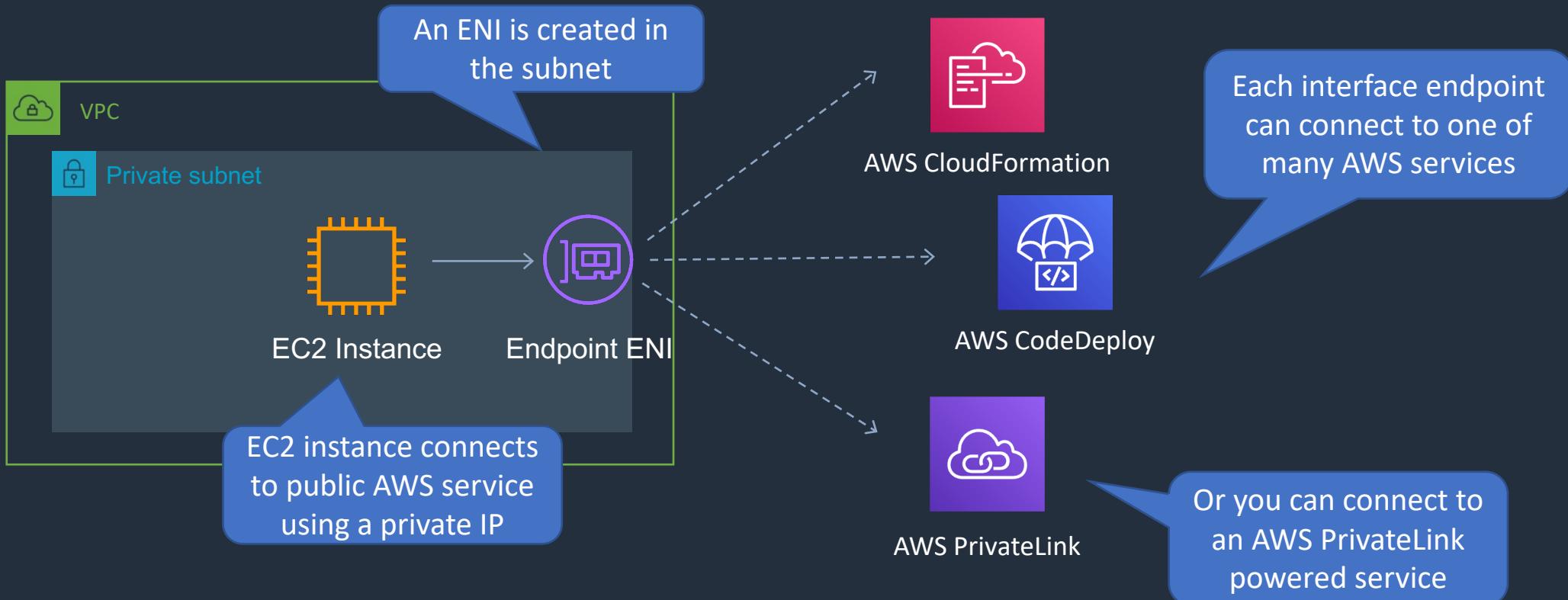
Amazon VPC Endpoints

- Enables private connectivity from a VPC to supported AWS services and VPC endpoint services powered by AWS PrivateLink
- Does not require an internet gateway, NAT device, VPN connection, or AWS Direct Connect connection
- Endpoints are virtual devices
- They are horizontally scaled, redundant, and highly available

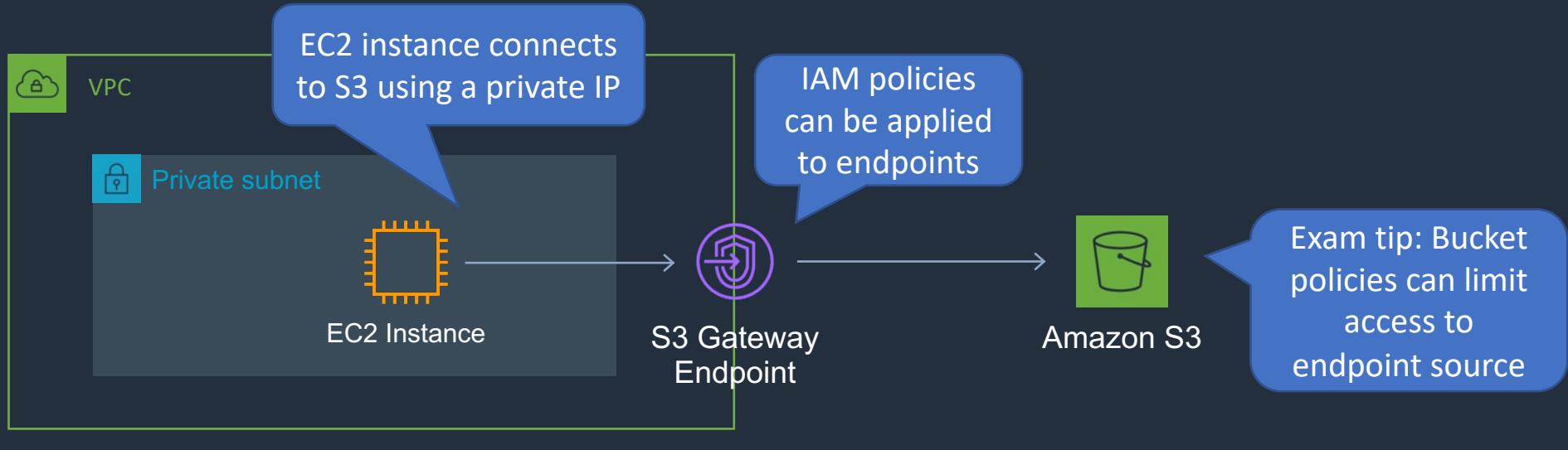
Amazon VPC Endpoints

- There are two types of VPC endpoints: interface endpoints and gateway endpoints.
- An interface endpoint is an elastic network interface with a private IP address that serves as an entry point for traffic destined to a supported service
- With an interface endpoint you remove the need for an internet gateway, NAT device, or virtual private gateway.
- A gateway endpoint is a gateway that you specify as a target for a route in your route table for traffic destined to a supported AWS service.
- The following AWS services are supported:
 - Amazon S3
 - DynamoDB

Amazon VPC Endpoint Services



Amazon S3 Gateway Endpoints



Route Table

Destination	Target
<code>pl-6ca54005 (com.amazonaws.ap-southeast-2.s3, 54.231.248.0/22, 54.231.252.0/24, 52.95.128.0/21)</code>	<code>vpc-e-ID</code>

A route table entry is required with the prefix list for S3 and the gateway ID

Amazon S3 Gateway Endpoint Policy Example

- Restricting access to a specific bucket

```
{  
  "Statement": [  
    {  
      "Sid": "Access-to-specific-bucket-only",  
      "Principal": "*",  
      "Action": [  
        "s3:GetObject",  
        "s3:PutObject"  
      ],  
      "Effect": "Allow",  
      "Resource": ["arn:aws:s3:::my_secure_bucket",  
                  "arn:aws:s3:::my_secure_bucket/*"]  
    }  
  ]  
}
```

Amazon S3 Bucket Policy Example

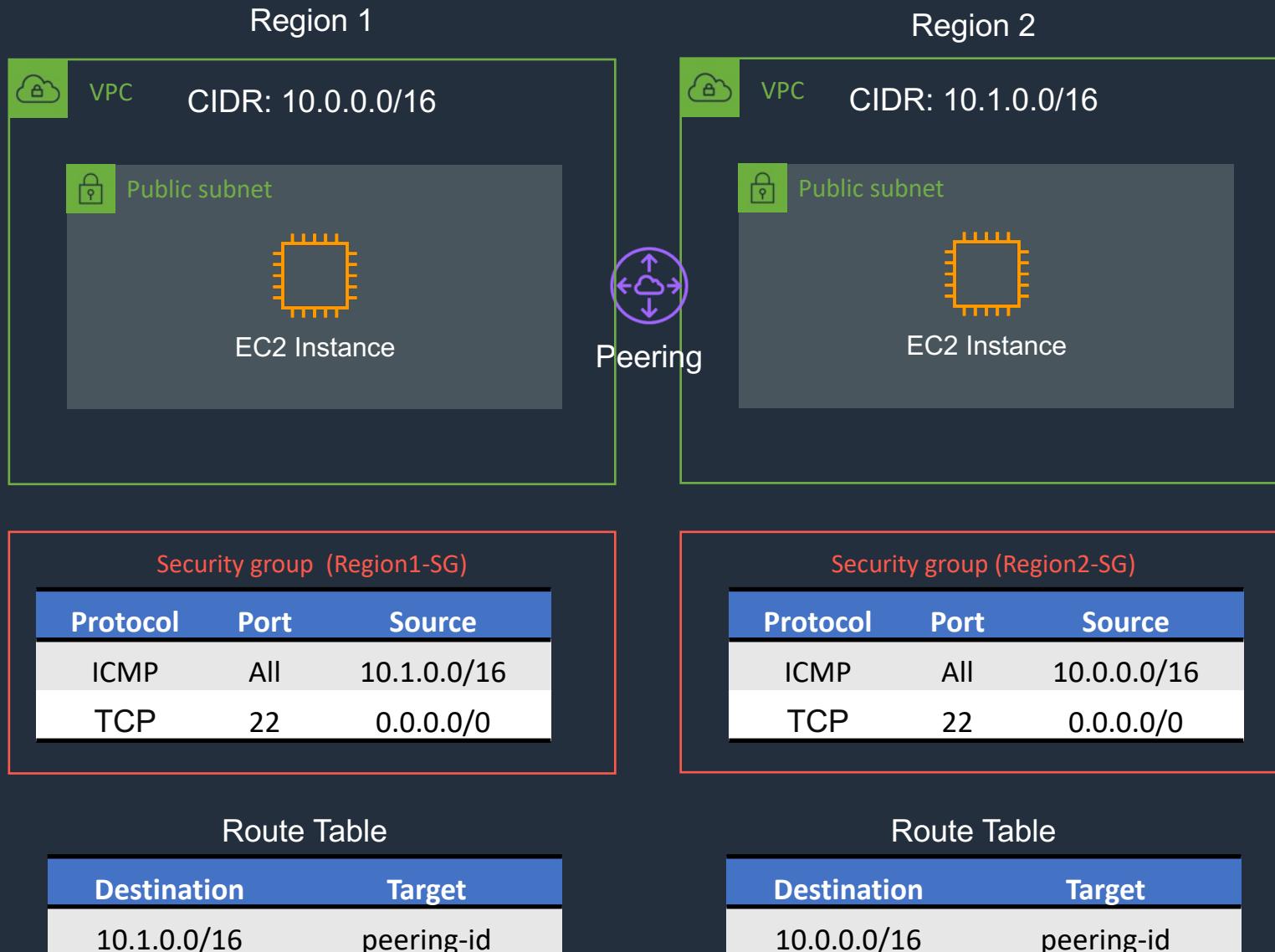
- Restricting access to a specific endpoint

```
{  
    "Version": "2012-10-17",  
    "Id": "Policy1415115909152",  
    "Statement": [  
        {  
            "Sid": "Access-to-specific-VPCE-only",  
            "Principal": "*",  
            "Action": "s3:*",  
            "Effect": "Deny",  
            "Resource": ["arn:aws:s3:::my_secure_bucket",  
                        "arn:aws:s3:::my_secure_bucket/*"],  
            "Condition": {  
                "StringNotEquals": {  
                    "aws:sourceVpce": "vpce-1a2b3c4d"  
                }  
            }  
        }  
    ]  
}
```

Amazon VPC Endpoints

	Interface Endpoint	Gateway Endpoint
What	Elastic Network Interface with a Private IP	A gateway that is a target for a specific route
How	Uses DNS entries to redirect traffic	Uses prefix lists in the route table to redirect traffic
Which services	API Gateway, CloudFormation, CloudWatch etc.	Amazon S3, DynamoDB
Security	Security Groups	VPC Endpoint Policies

Amazon VPC Peering



Amazon VPC Peering

- A VPC peering enables you to route traffic between VPCs using private IPv4 addresses or IPv6 addresses
- Instances in either VPC can communicate with each other as if they are within the same network
- You can create a VPC peering connection between your own VPCs, or with a VPC in another AWS account
- The VPCs can be in different regions (also known as an inter-region VPC peering connection)
- Data sent between VPCs in different regions is encrypted (traffic charges apply)

Amazon VPC Peering

- Cannot have overlapping CIDR ranges
- You can create multiple VPC peering connections for each VPC that you own, but transitive peering relationships are not supported
- Must update route tables to configure routing

Destination	Target
172.31.0.0/16	local
10.0.0.0/16	pcx-07a85f52b8e849b9e

- Must update the inbound and outbound rules for VPC security group to reference security groups in the peered VPC

Amazon VPC Peering

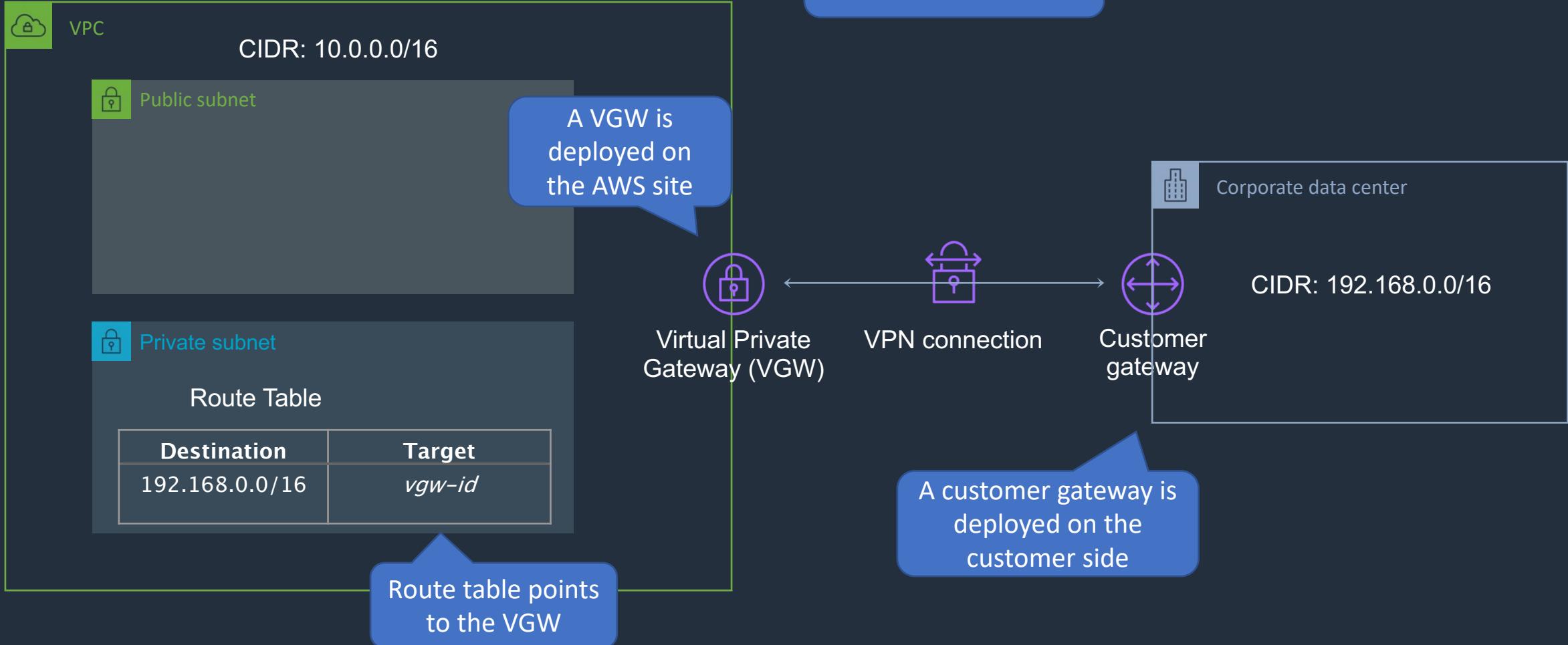
- When creating a VPC peering connection with another account you need to enter the account ID and VPC ID from the other account

Peering Connection	Status	Requester VPC	Acceptor VPC
pcx-07a85f52b8e849b9e	● Active	vpc-08b43399cff258a2f	vpc-ae3808c9

Amazon VPC Peering

What	AWS-provided network connectivity between two VPCs
When	Multiple VPCs need to communicate or access each other's resources
Pros	Uses AWS backbone without traversing the Internet
Cons	Transitive peering is not supported
How	VPC peering request made; accepter accepts request (either within or across accounts)

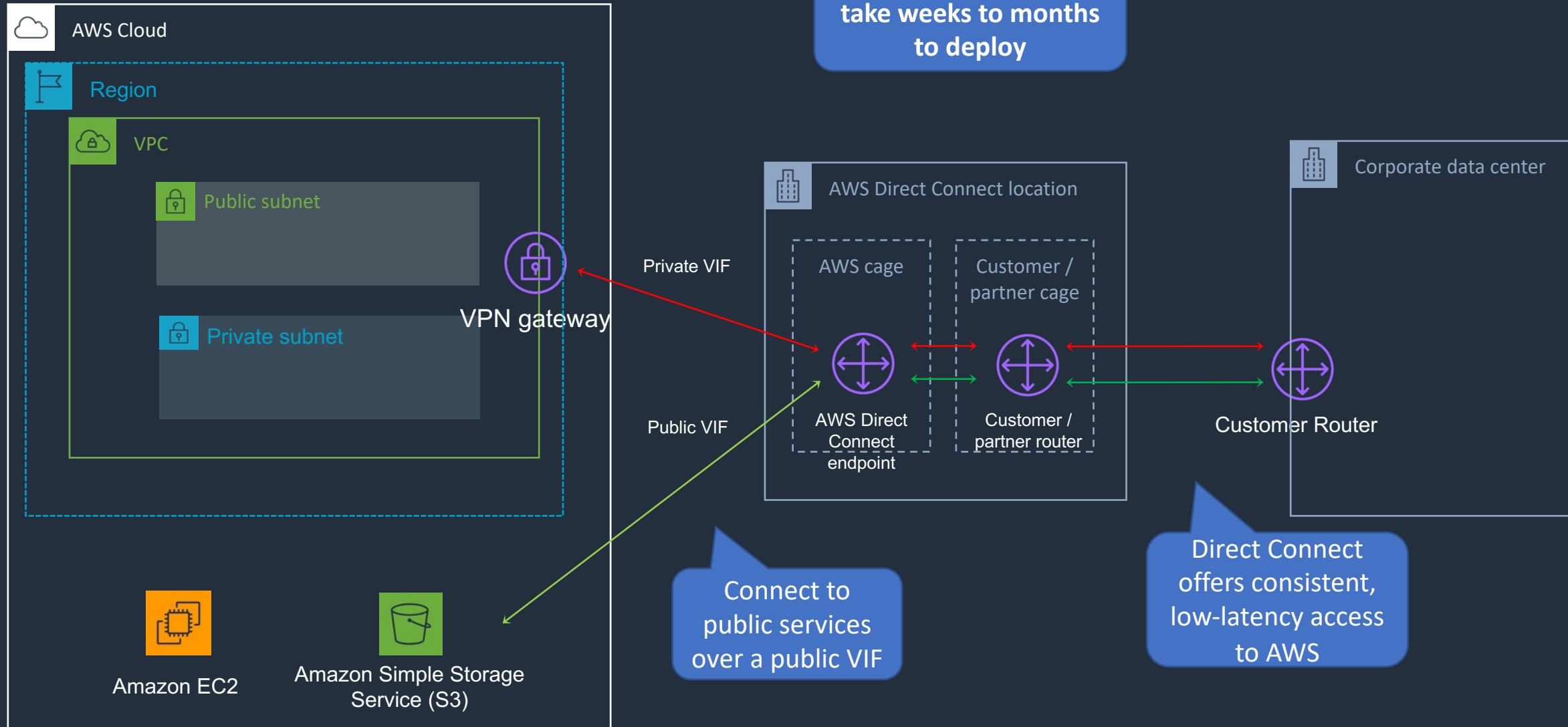
Amazon Virtual Private Networks (VPN)



AWS Managed VPN

What	AWS Managed IPSec VPN Connection over your existing Internet
When	Quick and usually simple way to establish a secure tunnelled connection to a VPC; redundant link for Direct Connect or other VPC VPN
Pros	Supports static routes or BGP peering and routing
Cons	Dependent on your Internet connection
How	Create a Virtual Private Gateway (VPG) on AWS, and a Customer Gateway on the on-premises side

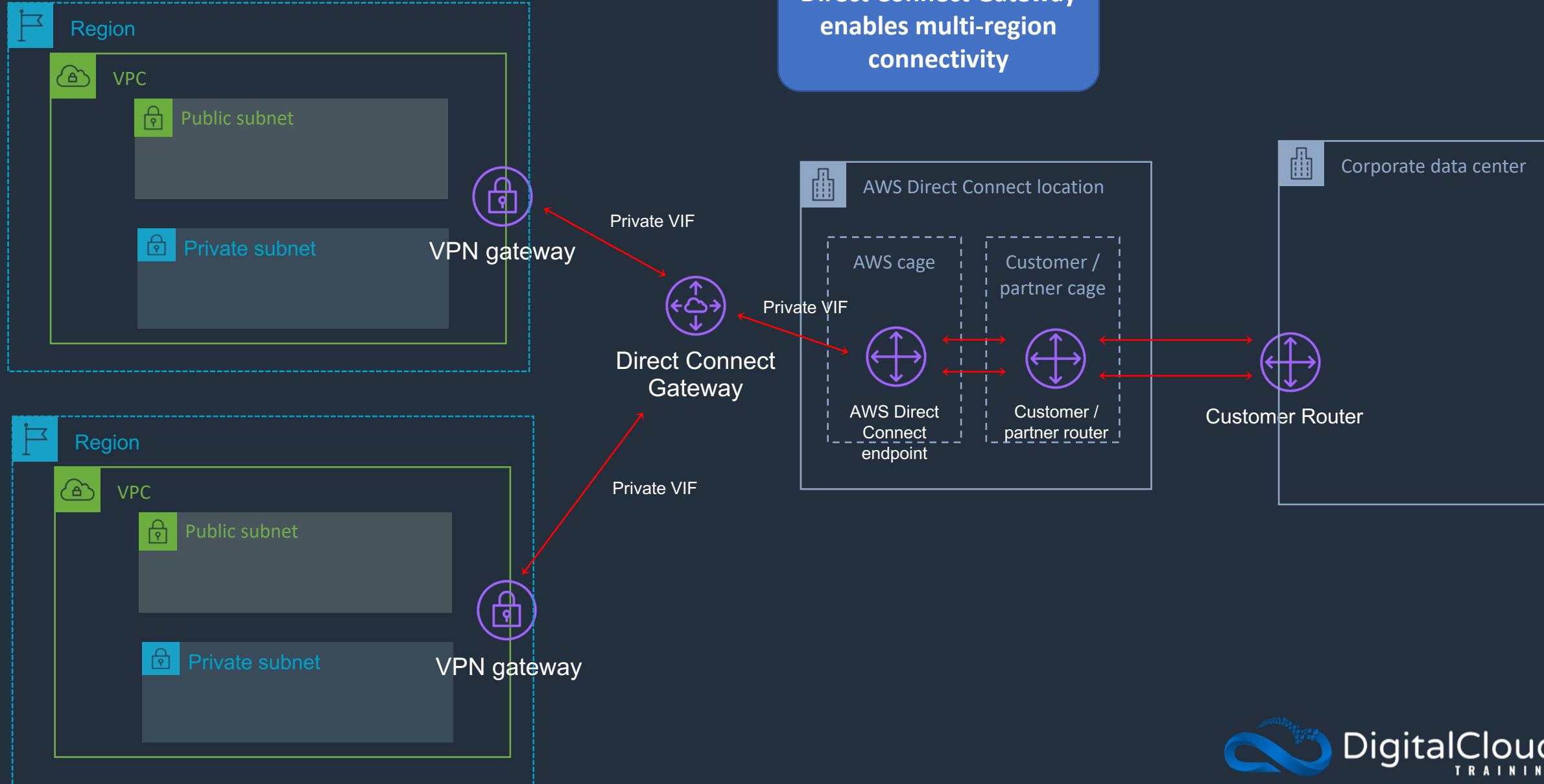
AWS Direct Connect



AWS Direct Connect

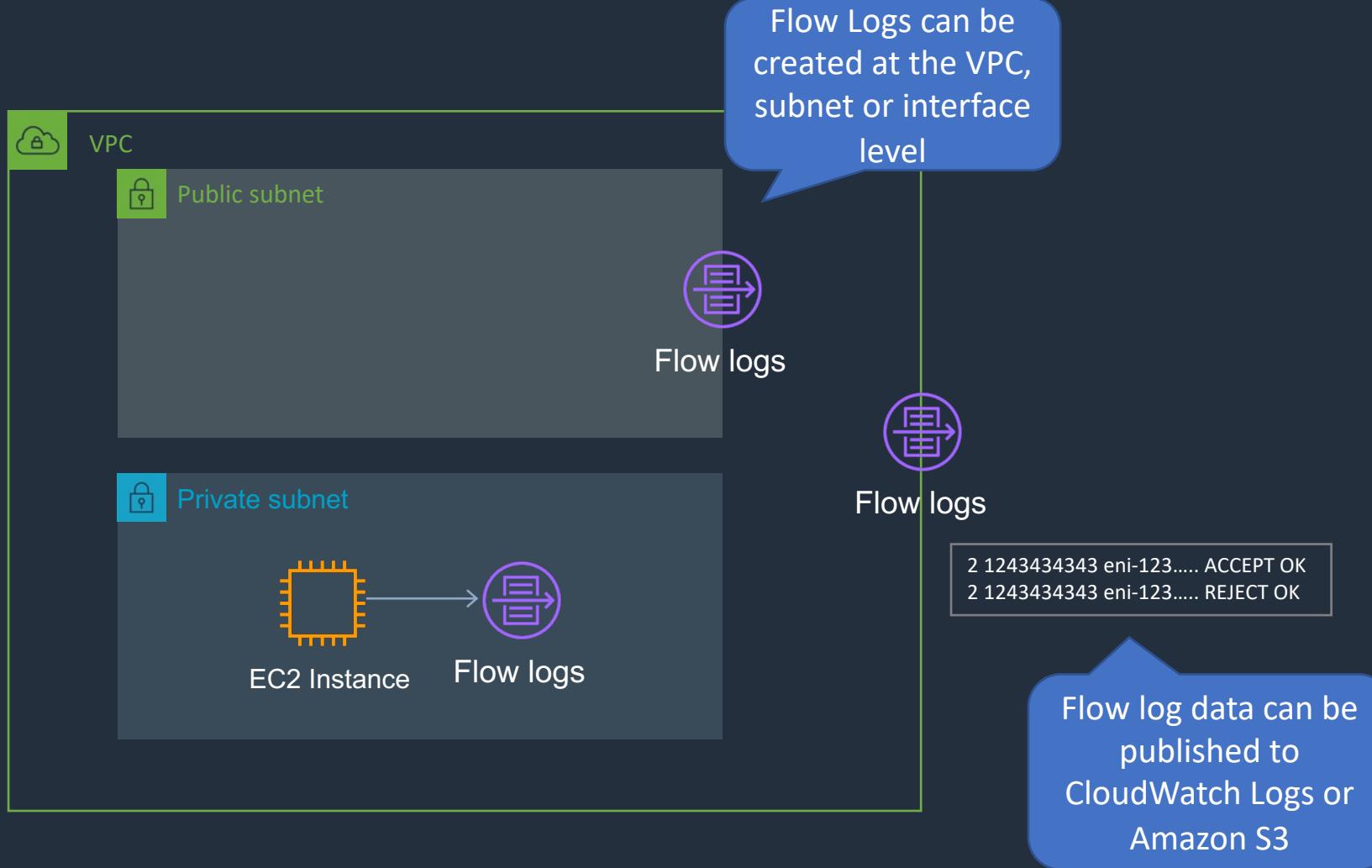
What	Dedicated network connection over private lines straight into the AWS backbone
When	Requires a large network link into AWS; lots of resources and services being provided on AWS to your corporate users
Pros	More predictable network performance; potential bandwidth cost reduction; up to 10 Gbps provisioned connections; supports BGP peering and routing
Cons	May require additional telecom and hosting provider relationships and/or network circuits; costly
How	Work with your existing data networking provider; create Virtual Interfaces (VIFs) to connect to VPCs (private VIFs) or other AWS services like S3 or Glacier (public VIFs)

AWS Direct Connect Gateway



Amazon VPC Flow Logs

Flow Logs capture data about IP traffic going to and from networking interfaces in a VPC



Amazon VPC Flow Logs vs ELB Access Logs

VPC Flow Log

version	account-id	interface-id	srcaddr	dstaddr	srcport	dstport	protocol	packets	bytes	start	end	action	log-status
2	55112233445	eni-0f5...	11.200.185.200	10.0.1.15	52933	22	6	1	401599...	1599...	ACCEPT	OK	
2	55112233445	eni-0f5...	10.0.1.15	11.200.185.200	22	52933	6	1	401599...	1599...	ACCEPT	OK	
2	55112233445	eni-0f5...	11.200.185.200	10.0.1.15	3624	80	6	1	441599...	1599...	REJECT	OK	
2	55112233445	eni-0f5...	11.200.185.200	10.0.1.15	3624	80	6	1	441599...	1599...	REJECT	OK	

ELB Access Log

```
http 2018-07-02T22:23:00.186641Z app/my-loadbalancer/50dc6c495c0c9188
192.168.131.39:2817 10.0.0.1:80 0.000 0.001 0.000 200 200 34 366
"GET http://www.example.com:80/ HTTP/1.1" "curl/7.46.0" --
arn:aws:elasticloadbalancing:us-east-2:123456789012:targetgroup/my-targets/73e2d6bc24d8a067
"Root=1-58337262-36d228ad5d99923122bbe354" "--" "--"
0 2018-07-02T22:22:48.364000Z "forward" "--" "--" 10.0.0.1:80 200 "--" "--"
```

Exam Scenarios

Exam Scenario	Solution
Need to identify the instances that are generating the most traffic using a NAT gateway	Use VPC flow logs on the NAT gateway ENI and use CloudWatch insights to filter based on source IP address
Latency on a NAT instance has increased, need a solution that scales with demand cost-efficiently	Swap with a NAT gateway
NAT gateway is NOT highly available across AZs, only within an AZ	Use multiple NAT gateways for HA across AZs
NAT instance deployed but not working	Make sure to disable source/destination checks
Need to enable access to S3 without the instances using public IP addresses	Use a NAT gateway or VPC endpoint

Exam Scenarios

Exam Scenario	Solution
EC2 instance in private subnet cannot reach the Internet. Route table has a route to a NAT gateway with a status of "Blackhole"	Indicates the NAT gateway has been deleted
Need to connect to S3 from EC2 using private network only. Must also ensure that only the instances can access the bucket	Create a VPC endpoint and a bucket policy with a Condition that limits S3 actions to the VPC endpoint as the source
VPC endpoint setup to allow private IP address connectivity to S3 bucket, permissions configured, but instances still can't connect	Make sure the subnet has a target in the route table for the VPC endpoint

Exam Scenarios

Exam Scenario	Solution
Need to manage EC2 instances in a private subnet from an office using SSH but instances cannot have internet access	Add a VGW and configure routing in the VPC and establish a VPN to the office
Need encryption in-transit and at-rest for hybrid environment	Use an AWS VPN and use KMS keys for data encryption
Network change was made that resulted in application to DB connection issues	Analyze using VPC Flow Logs
Inbound and outbound internet connectivity required for EC2 instances	Need to attach an internet gateway to the VPC and add an entry in the route table for the subnet that points to the internet gateway

Exam Scenarios

Exam Scenario	Solution
Web application has EC2 with public IPs behind an ALB. EC2 instances cannot connect to external service	Need to create an attach an IGW to the VPC and update the route table
VPC peering connection setup between two different VPCs. Instances in private subnets still can't communicate	Make sure the route tables are updated
A company has configured a VPC peering connection between two VPCs and needs to set up connectivity between instances in private subnets	Configure the VPC route tables with routes pointing to the address range of the other VPC
Company backing up one VPC to another in different region. All data must be private and encrypted	Use inter-region VPC peering which encrypts across the AWS global network

Exam Scenarios

Exam Scenario	Solution
Malicious IP identified and must be blocked from all ingress and egress connectivity	Add a rule to a network ACL for all affected subnets
VPC connected to data center by VPN. User pings private subnet instance from on-prem computer and fails. VPC Flow Logs show accept for inbound but reject for outbound traffic	Modify the network ACL to allow outbound traffic
Malicious traffic coming from a single IP address	Use a NACL for the web server subnet to deny IP address
Admin has setup instance for remote access and can SSH from internet but cannot ping	Most likely reason is that the instance's security group does not have a rule allowing ICMP

Exam Scenarios

Exam Scenario	Solution
Admin connecting to EC2 instance using SSH from office but gets connection timeout from home	Most likely doesn't have the home network IP range in the security group allow rule for SSH

SECTION 10

DNS: Amazon Route 53

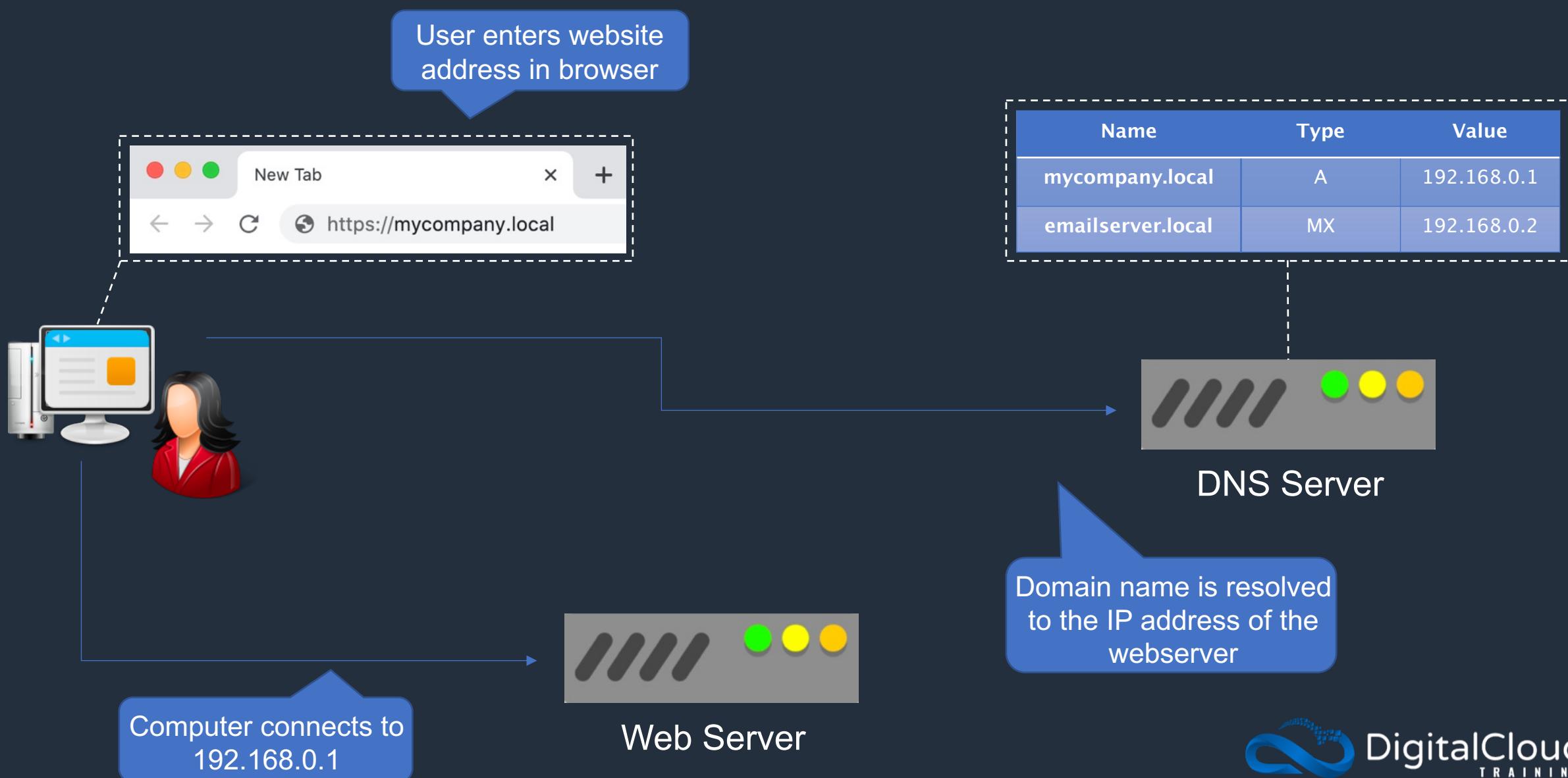
Amazon Route 53 Overview



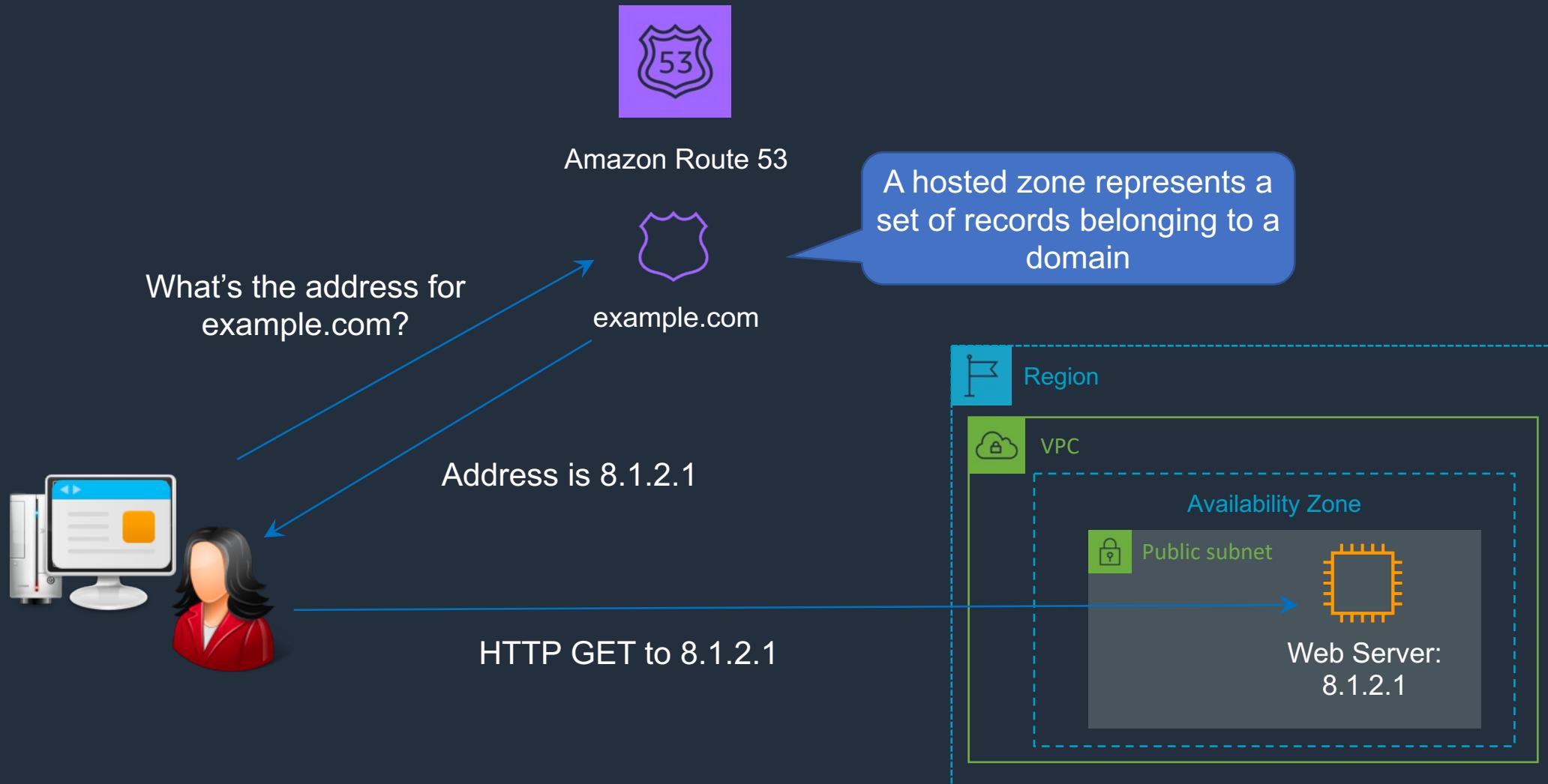
Amazon Route 53



DNS Resolution



DNS Resolution with AWS Route 53



Amazon Route 53 DNS Record Types

- Supported DNS records**
- A (address record)
 - AAAA (IPv6 address record)
 - CNAME (canonical name record)
 - Alias (an Amazon Route 53-specific virtual record)
 - CAA (certification authority authorization)
 - MX (mail exchange record)
 - NAPTR (name authority pointer record)
 - NS (name server record)
 - PTR (pointer record)
 - SOA (start of authority record)
 - SPF (sender policy framework)
 - SRV (service locator)
 - TXT (text record)

	CNAME	Alias
	Route 53 charges for CNAME queries	Route 53 doesn't charge for alias queries to AWS resources
	You can't create a CNAME record at the top node of a DNS namespace (zone apex)	You can create an alias record at the zone apex (however you can't route to a CNAME at the zone apex)
	A CNAME can point to any DNS record that is hosted anywhere	An alias record can only point to a CloudFront distribution, Elastic Beanstalk environment, ELB, S3 bucket as a static website, or to another record in the same hosted zone that you're creating the alias record in

Using Alias and CNAME Records

Alias to an Elastic Load Balancer

Record name	Type	Routing policy	Alias	Value/Route traffic to	TTL (seconds)	Health check	Evaluate target health
dctlabs.com	A	Simple	Yes	dualstack.myalb-219752408.ap-southeast-2.elb.amazonaws.com.	-	-	Yes

Zone apex can be used (dctlabs.com)

Alias to an Amazon S3 bucket (static website)

Record name	Type	Routing policy	Alias	Value/Route traffic to	TTL (seconds)	Health check	Evaluate target health
dctlabs.com	A	Simple	Yes	s3-website-ap-southeast-2.amazonaws.com.	-	-	Yes

CNAME of subdomain to another DNS name

Record name	Type	Routing policy	Alias	Value/Route traffic to	TTL (seconds)	Health check	Evaluate target health
app.dctlabs.com	CNAME	Simple	No	www.example.com	300	-	-

The CNAME must be a subdomain or you get this error



Error occurred

Bad request.

(InvalidChangeBatch 400: RRSet of type CNAME with DNS name dctlabs.com. is not permitted at apex in zone dctlabs.com.)

Route 53 DNS Routing Policies

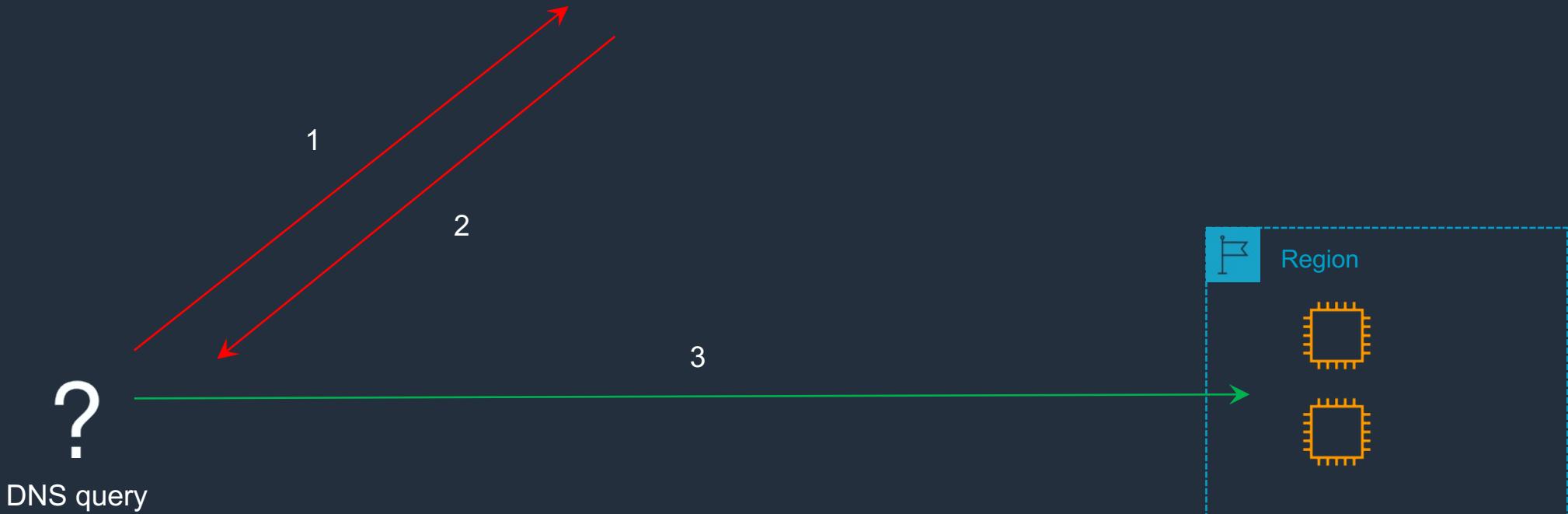
Routing Policy	What it does
Simple	Simple DNS response providing the IP address associated with a name
Failover	If primary is down (based on health checks), routes to secondary destination
Geolocation	Uses geographic location you're in (e.g. Europe) to route you to the closest region
Geoproximity	Routes you to the closest region within a geographic area
Latency	Directs you based on the lowest latency route to resources
Multivalue answer	Returns several IP addresses and functions as a basic load balancer
Weighted	Uses the relative weights assigned to resources to determine which to route to

Amazon Route 53 - Simple Routing Policy

Name	Type	Value	TTL
simple.dctlabs.com	A	1.1.1.1	60
		2.2.2.2	
simple2.dctlabs.com	A	3.3.3.3	60



Amazon Route 53

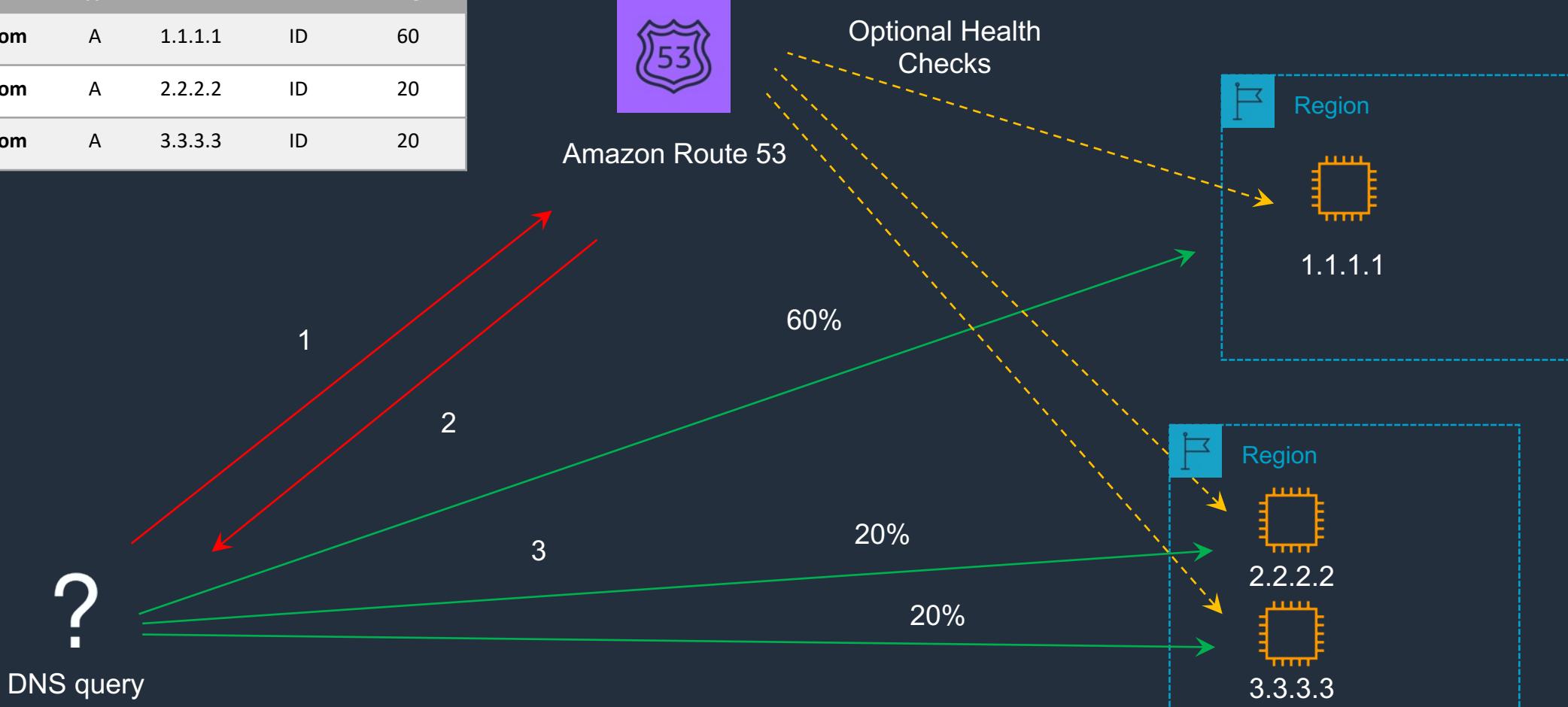


Amazon Route 53 - Simple Routing Policy

- With simple routing, you typically route traffic to a single resource such as a webserver
- You can't create multiple records that have the same name and type, but you can specify multiple values in the same record, such as multiple IP addresses
- When using multiple values in a record:
 - Route 53 returns all values to the recursive resolver in random order, and the resolver returns the values to the client
 - The client then chooses a value and resubmits the query

Amazon Route 53 - Weighted Routing Policy

Name	Type	Value	Health	Weight
weighted.dctlabs.com	A	1.1.1.1	ID	60
weighted.dctlabs.com	A	2.2.2.2	ID	20
weighted.dctlabs.com	A	3.3.3.3	ID	20

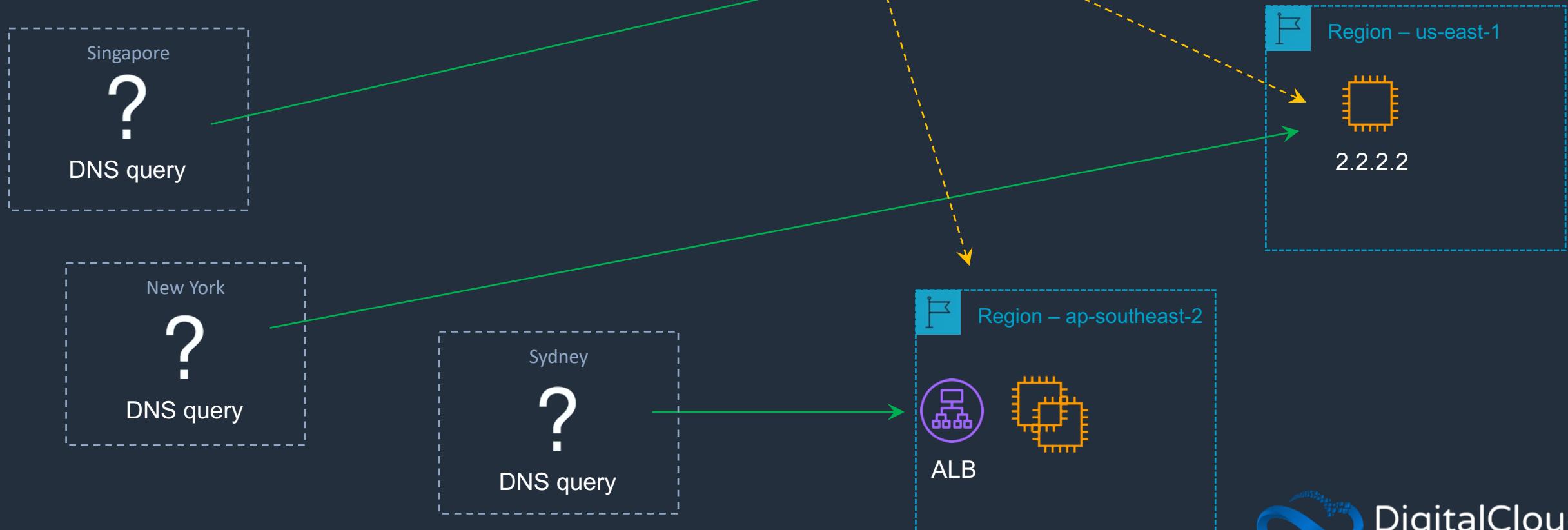


Amazon Route 53 - Weighted Routing Policy

- Create records that have the same name and type for each of your resources
- Assign each record a relative weight that corresponds with how much traffic you want to send to each resource
- Route 53 sends traffic to a resource based on the weight that you assign to the record as a proportion of the total weight for all records in the group
- Uses an integer between 0 and 255
- To disable routing to a resource, set Weight to 0
- If you set Weight to 0 for all of the records in the group, traffic is routed to all resources with equal probability

Amazon Route 53 - Latency Routing Policy

Name	Type	Value	Health	Region
latency.dctlabs.com	A	1.1.1.1	ID	ap-southeast-1
latency.dctlabs.com	A	2.2.2.2	ID	us-east-1
latency.dctlabs.com	A	alb-id	ID	ap-southeast-2



Amazon Route 53 - Latency Routing Policy

- When Route 53 receives a DNS query for a domain it determines which AWS Regions you've created latency records for, determines which region gives the user the lowest latency, and then selects a latency record for that region
- Route 53 responds with the value from the selected record, such as the IP address for a web server

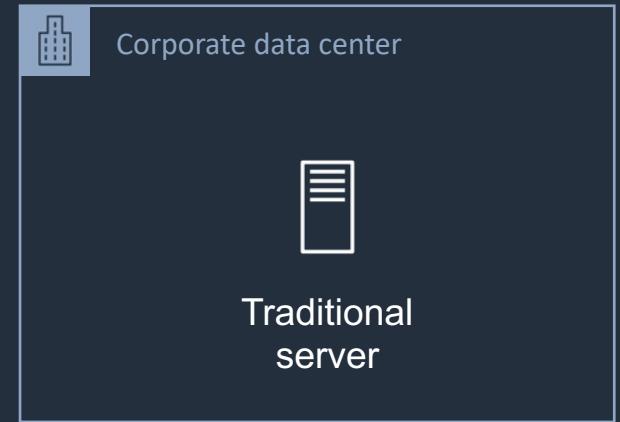
Amazon Route 53 - Failover Routing Policy

Name	Type	Value	Health	Record Type
failover.dctlabs.com	A	1.1.1.1	ID	Primary
failover.dctlabs.com	A	alb-id		Secondary

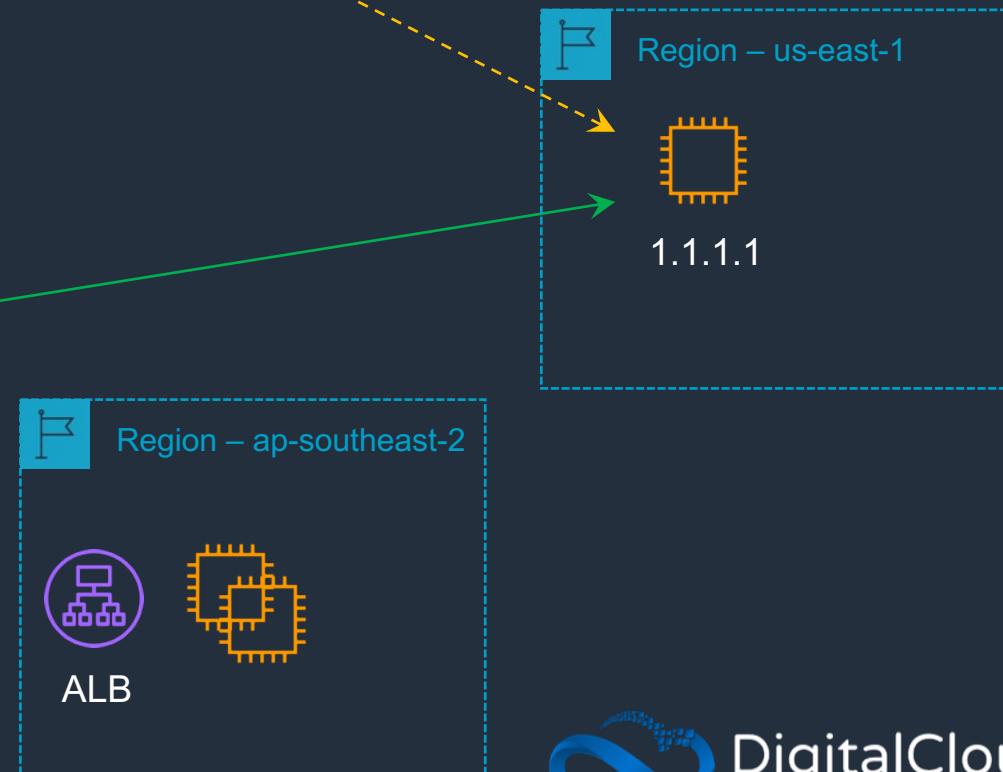


Amazon Route 53

Health Check
required on
Primary



DNS query

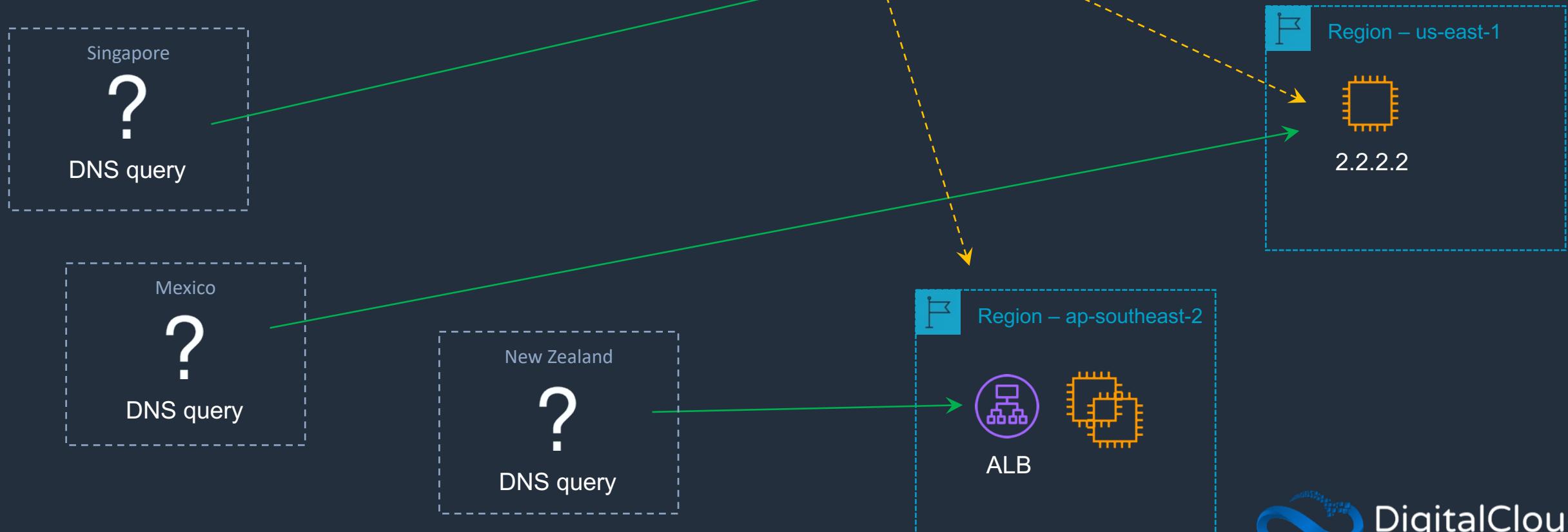


Amazon Route 53 - Failover Routing Policy

- When responding to queries, Route 53 includes only the healthy primary resources
- If all the primary resources are unhealthy, Route 53 begins to include only the healthy secondary resources in response to DNS queries
- If you're routing traffic to any AWS resources that you can create alias records for, don't create health checks for those resources. When you create the alias records, you set Evaluate Target Health to Yes instead

Amazon Route 53 - Geolocation Routing Policy

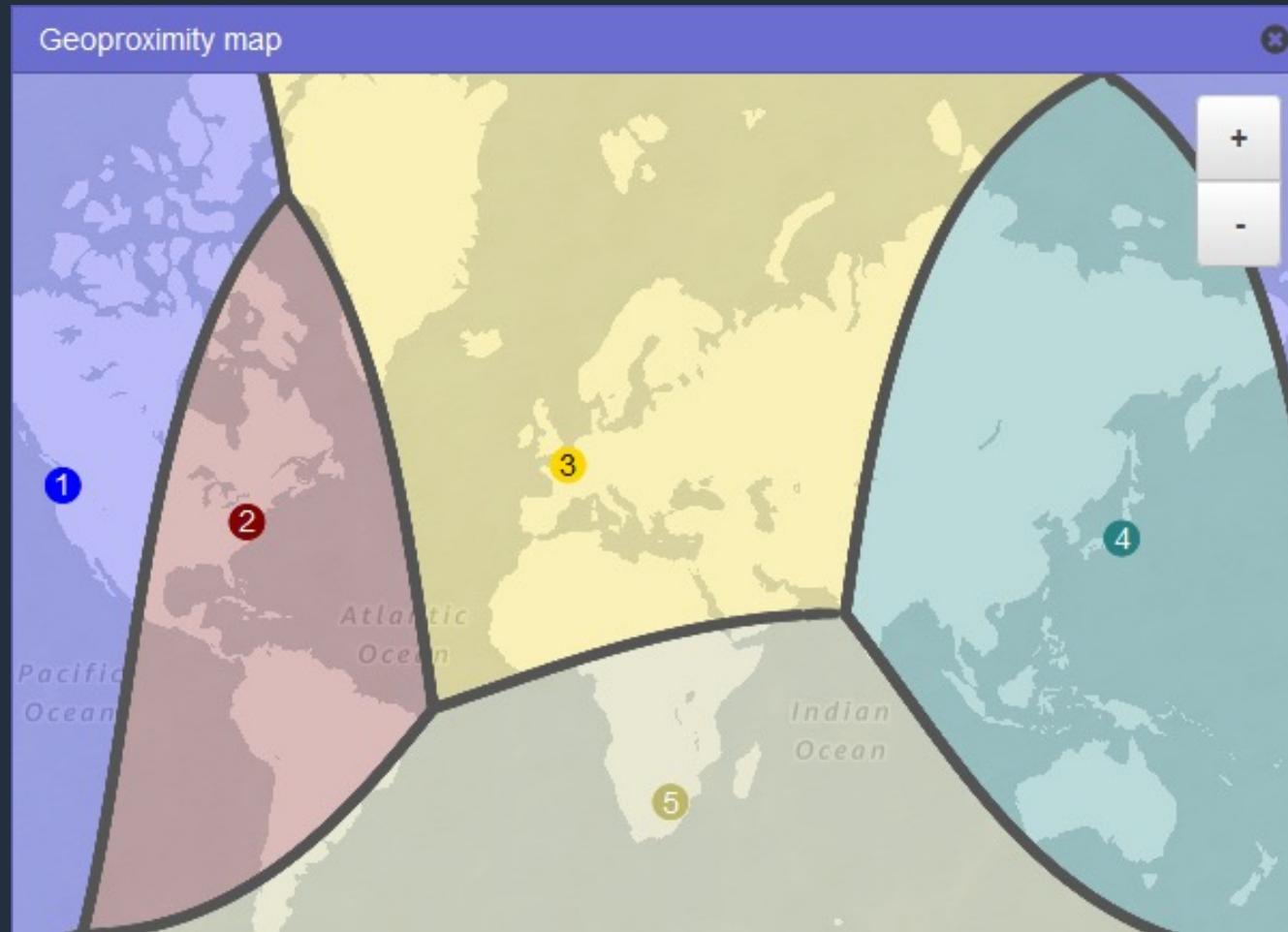
Name	Type	Value	Health	Geolocation
geolocation.dctlabs.com	A	1.1.1.1	ID	Singapore
geolocation.dctlabs.com	A	2.2.2.2	ID	Default
geolocation.dctlabs.com	A	alb-id	ID	Oceania



Amazon Route 53 - Geoproximity Routing Policy

- To use geoproximity routing, you must use Route 53 traffic flow
- You create geoproximity rules for your resources and specify one of the following values for each rule:
 - If you're using AWS resources, the AWS Region that you created the resource in
 - If you're using non-AWS resources, the latitude and longitude of the resource

Amazon Route 53 - Geoproximity Routing Policy



Amazon Route 53 - Multivalue Routing Policy

Name	Type	Value	Health	Multi Value
multivalue.dctlabs.com	A	1.1.1.1	ID	Yes
multivalue.dctlabs.com	A	2.2.2.2	ID	Yes
multivalue.dctlabs.com	A	3.3.3.3	ID	Yes



Amazon Route 53

Health Checks:
returns healthy
records only



Amazon Route 53 - Multivalue Routing Policy

- To route traffic approximately randomly to multiple resources, such as web servers, you create one multivalue answer record for each resource
- Can optionally associate a Route 53 health check with each record
- Route 53 responds to DNS queries with up to eight healthy records and gives different answers to different DNS resolvers

Amazon Route 53 – Health Checks

There are three types of Amazon Route 53 health checks:

- Health checks that monitor an endpoint
- Health checks that monitor other health checks (calculated health checks)
- Health checks that monitor CloudWatch alarms

Exam Scenarios

Exam Scenario	Solution
Use Route 53 to direct based on health checks with (2xx) traffic to primary and other responses to secondary	Need to create an A record for each server and a HTTP (not TCP) health check
Route 53 health check uses string matching for "/html". Alert shows health check fails	The search string must appear entirely within the first 5,120 bytes of the response body
Need to make a website promotion visible to users from a specific country only	Use Route 53 geolocation routing policy

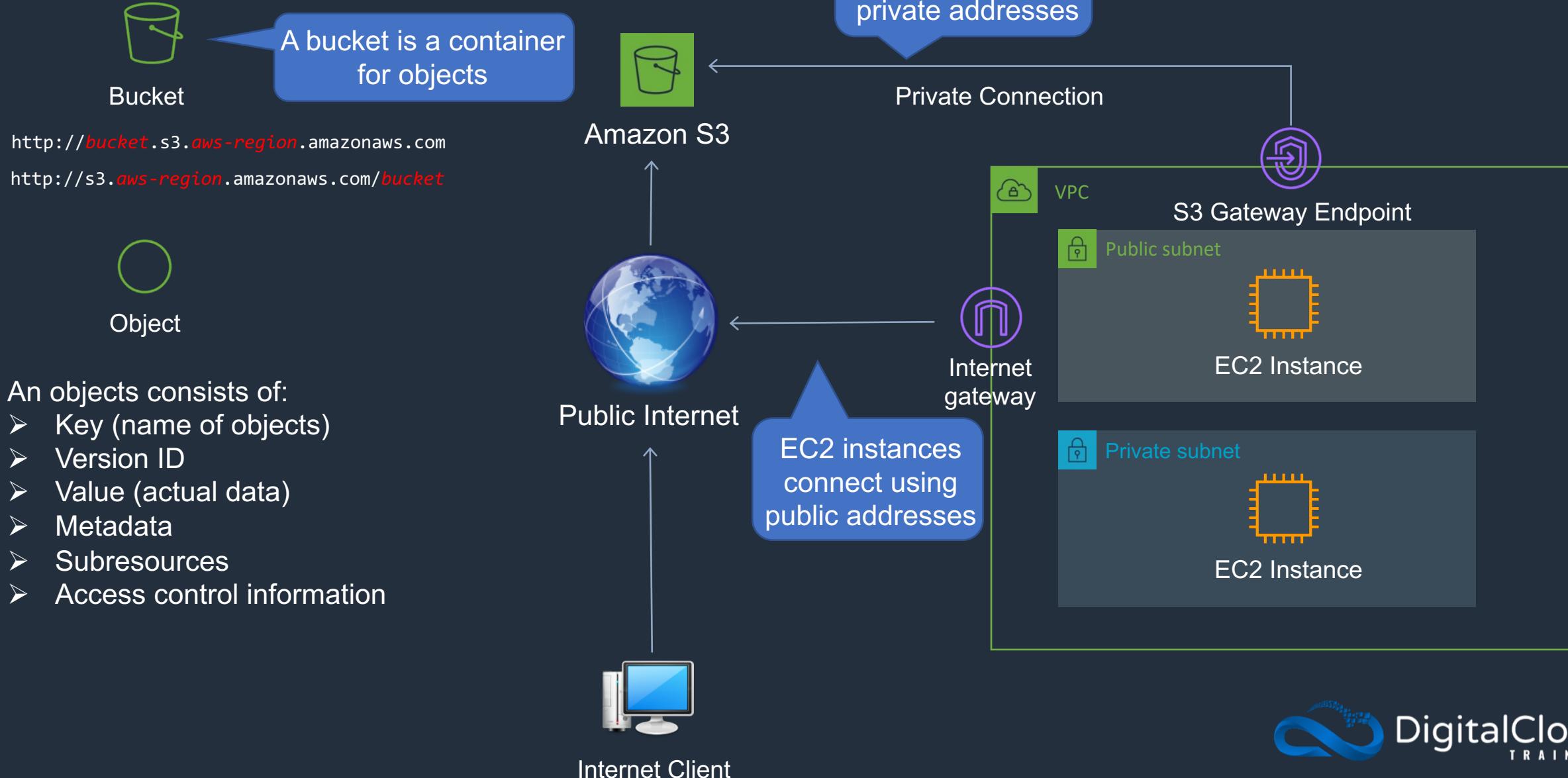
Exam Scenarios

Exam Scenario	Solution
New website runs on EC2 behind ALB. Need to create record in Route 53 to point to the domain apex (e.g. example.com)	Use an alias record
Hosted zone in Account A and ALB in Account B. Need the most cost-effective and efficient solution for pointing to the ALB	Create an Alias record in Account A that points to ALB in Account B

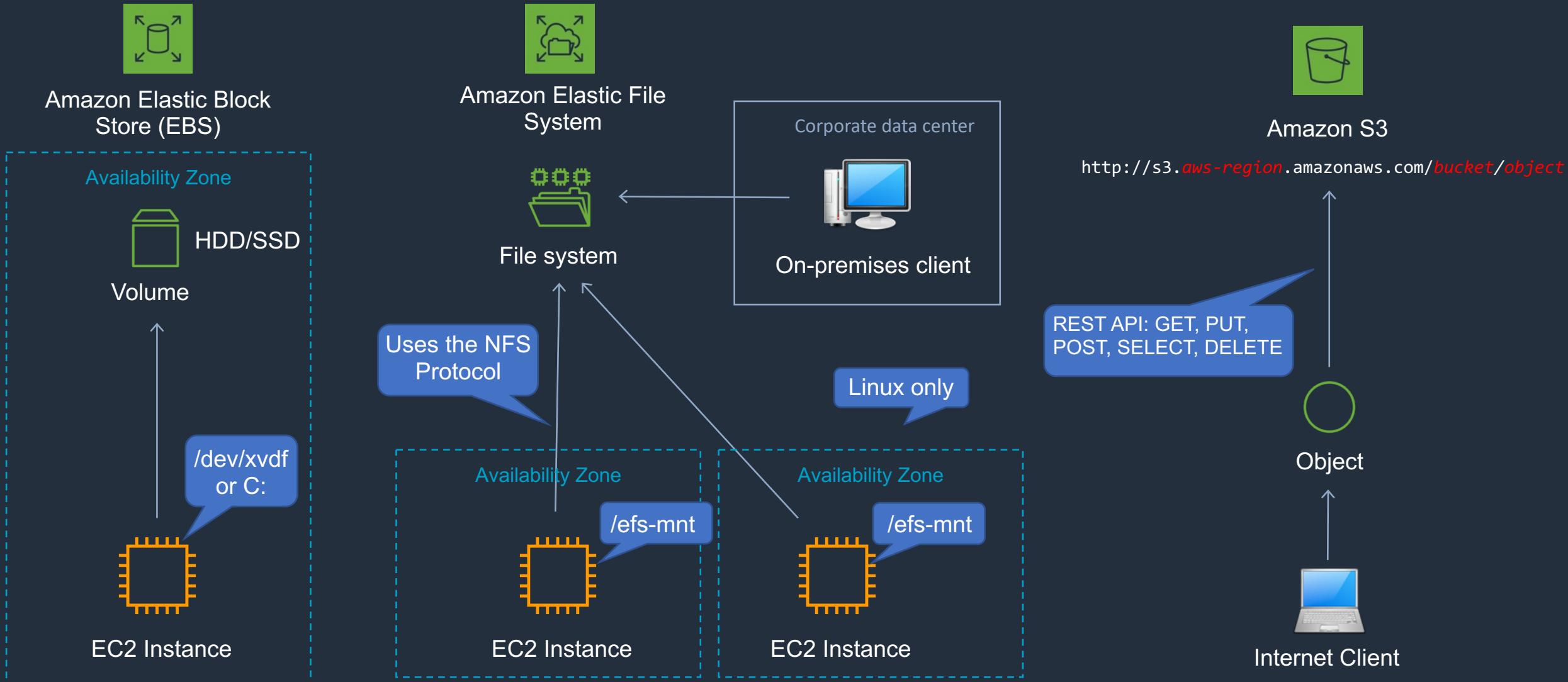
SECTION 11

Object Storage and Content Delivery: S3 and CloudFront

Amazon Simple Storage Service (S3)



Block, File, and Object Storage



S3 Storage Classes

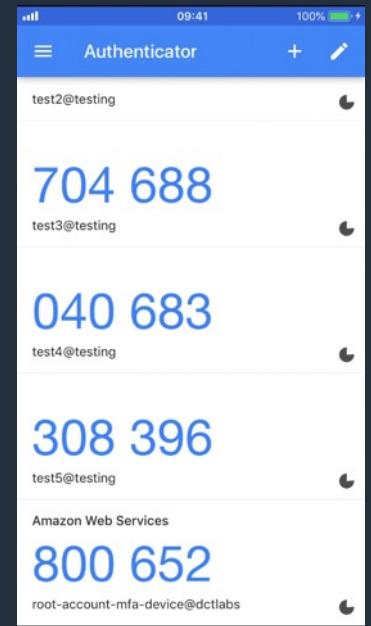
	S3 Standard	S3 Intelligent Tiering	S3 Standard-IA	S3 One Zone-IA	S3 Glacier	S3 Glacier Deep Archive
Designed for durability	99.99999999%	99.99999999%	99.99999999%	99.99999999%	99.99999999%	99.99999999%
Designed for availability	99.99%	99.9%	99.9%	99.5%	99.99%	99.99%
Availability SLA	99.9%	99%	99%	99%	99.9%	99.9%
Availability Zones	≥3	≥3	≥3	1	≥3	≥3
Minimum capacity charge per object	N/A	N/A	128KB	128KB	40KB	40KB
Minimum storage duration charge	N/A	30 days	30 days	30 days	90 days	180 days
Retrieval fee	N/A	N/A	Per GB retrieved	Per GB retrieved	Per GB retrieved	Per GB retrieved
First byte latency	milliseconds	milliseconds	milliseconds	milliseconds	select minutes or hours	select hours
Storage type	Object	Object	Object	Object	Object	Object
Lifecycle transitions	Yes	Yes	Yes	Yes	Yes	Yes

S3 Versioning

- Versioning is a means of keeping multiple variants of an object in the same bucket
- Use versioning to preserve, retrieve, and restore every version of every object stored in your Amazon S3 bucket
- Versioning-enabled buckets enable you to recover objects from accidental deletion or overwrite

S3 Multi-Factor Authentication Delete (MFA Delete)

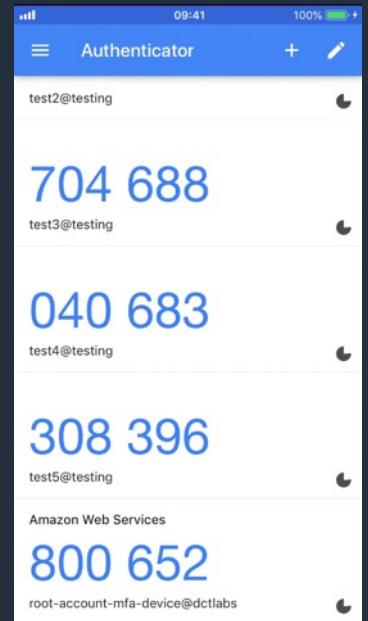
- Adds another layer of security by configuring a bucket to enable an additional authentication for the following operations:
 - Changing the versioning state of a bucket
 - Permanently deleting an object version
- The `x-amz-mfa` request header must be included in the above requests
- MFA Delete requires a second factor (in addition to security credentials)
- The second factor is a token generated by a hardware device or software program
- Requires versioning to be enabled on the bucket



S3 Multi-Factor Authentication Delete (MFA Delete)

- The bucket owner, the AWS account that created the bucket (root account), and all authorized IAM users can enable versioning
- Only the bucket owner (root account) can enable MFA Delete

MFA



MFA-Protected API Access

- Used to enforce another authentication factor (MFA code) when accessing sensitive Amazon S3 resources
- Enforce MFA using the aws:MultiFactorAuthAge key in a bucket policy:

```
{  
    "Version": "2012-10-17",  
    "Id": "123",  
    "Statement": [  
        {  
            "Sid": "",  
            "Effect": "Deny",  
            "Principal": "*",  
            "Action": "s3:*",  
            "Resource": "arn:aws:s3:::examplebucket/securedocuments/*",  
            "Condition": { "Null": { "aws:MultiFactorAuthAge": true } }  
        }  
    ]  
}
```



Denies any API operation that is not authenticated using MFA

S3 Lifecycle Management

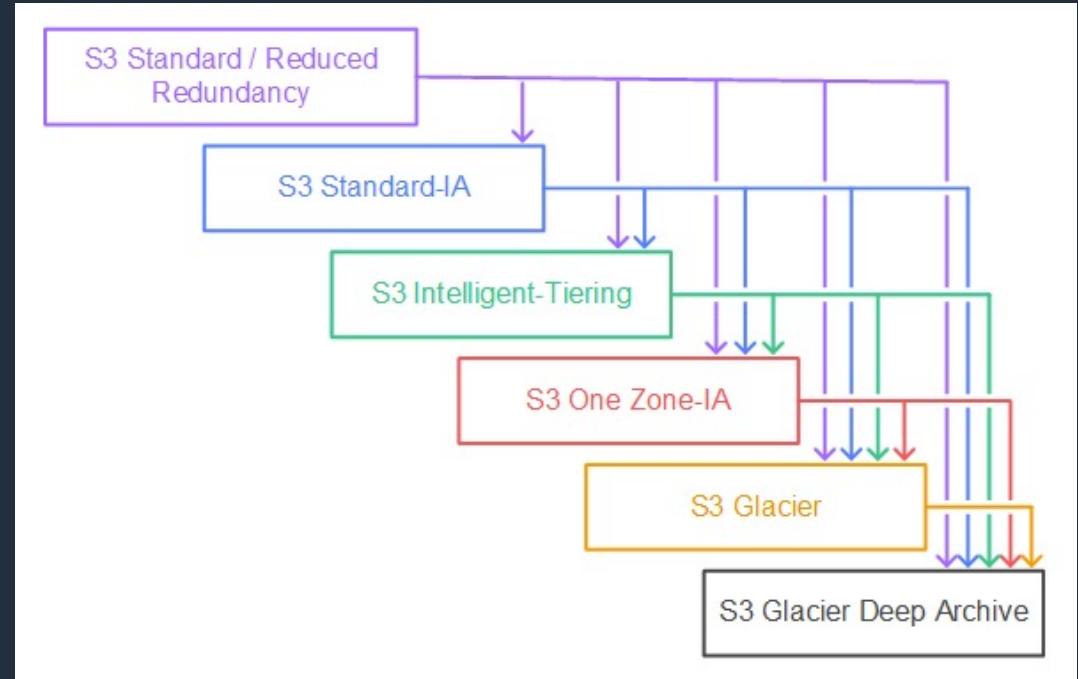
There are two types of actions:

- Transition actions - Define when objects transition to another storage class
- Expiration actions - Define when objects expire (get deleted by S3)

S3 Lifecycle Management: Supported Transitions

You can transition from the following:

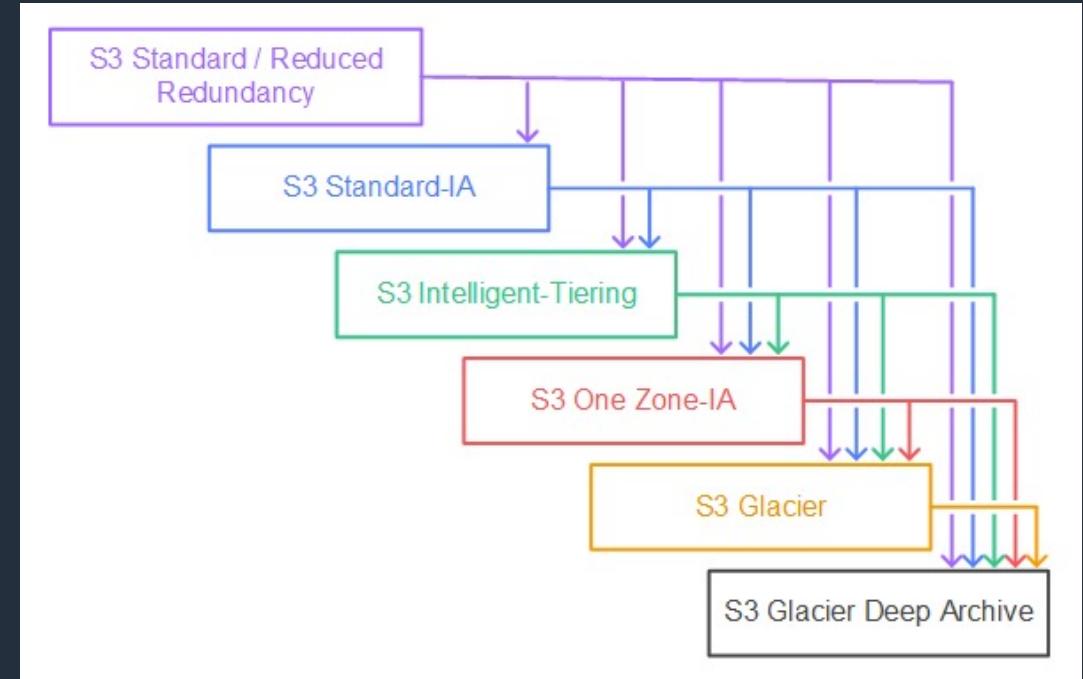
- The S3 Standard storage class to any other storage class
- Any storage class to the S3 Glacier or S3 Glacier Deep Archive storage classes
- The S3 Standard-IA storage class to the S3 Intelligent-Tiering or S3 One Zone-IA storage classes
- The S3 Intelligent-Tiering storage class to the S3 One Zone-IA storage class
- The S3 Glacier storage class to the S3 Glacier Deep Archive storage class



S3 Lifecycle Management: Unsupported Transitions

You can't transition from the following:

- Any storage class to the S3 Standard storage class
- Any storage class to the Reduced Redundancy storage class
- The S3 Intelligent-Tiering storage class to the S3 Standard-IA storage class
- The S3 One Zone-IA storage class to the S3 Standard-IA or S3 Intelligent-Tiering storage classes



S3 Lifecycle Management

- Can create a lifecycle policy through the console or CLI/API
- When configured using the CLI/API an XML or JSON file must be supplied
- API actions to create/update/delete lifecycle policies:
 - PutBucketLifecycleConfiguration - Creates a new lifecycle configuration for the bucket or replaces an existing lifecycle configuration
 - GetBucketLifecycleConfiguration - Returns the lifecycle configuration information set on the bucket
 - DeleteBucketLifecycle - Deletes the lifecycle configuration from the specified bucket

Example S3 Lifecycle Policy (XML)

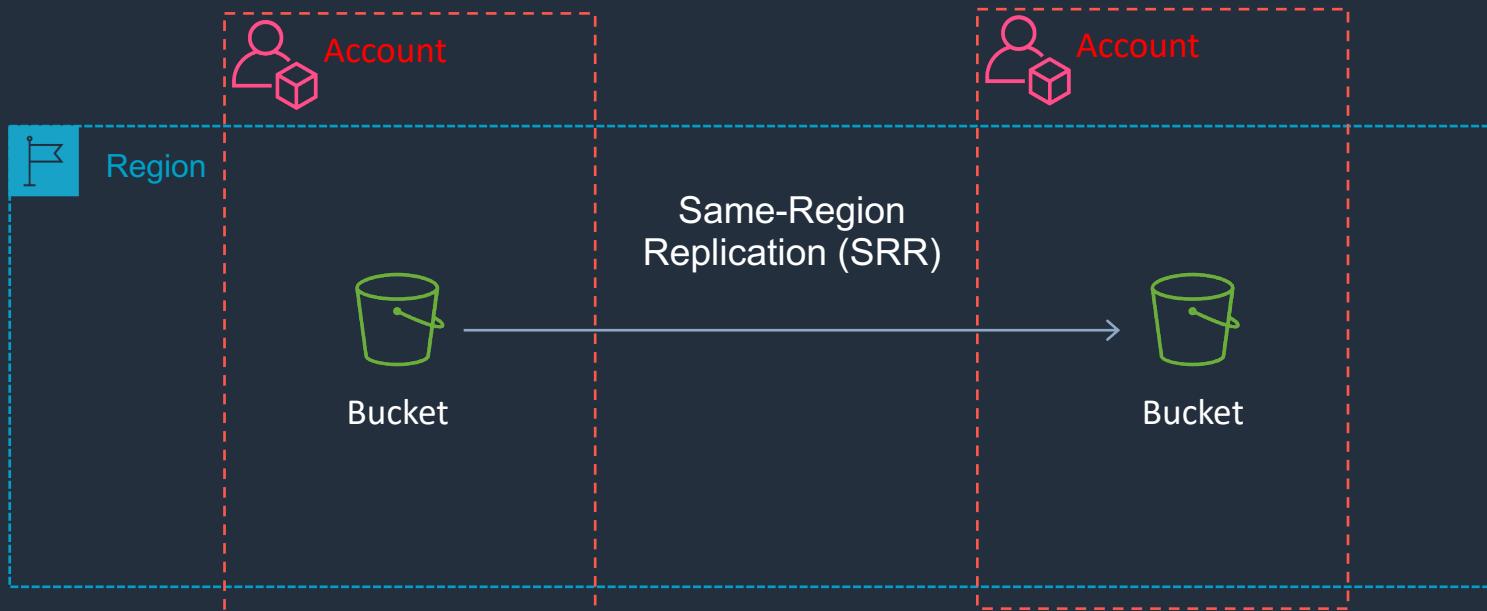
```
<LifecycleConfiguration>
    <Rule>
        <ID>ExampleRule</ID>
        <Filter>
            <Prefix>documents/</Prefix>
        </Filter>
        <Status>Enabled</Status>
        <Transition>
            <Days>365</Days>
            <StorageClass>GLACIER</StorageClass>
        </Transition>
        <Expiration>
            <Days>3650</Days>
        </Expiration>
    </Rule>
</LifecycleConfiguration>
```

Amazon S3 – Replication

Cross-Region Replication (CRR)



Same-Region Replication (SRR)



Amazon S3 –Replication

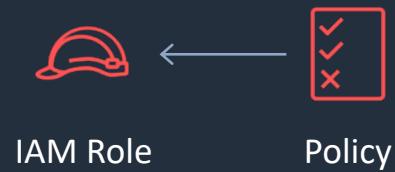
- You can replicate objects between different AWS Regions or within the same AWS Region
- Cross-Region replication (CRR) is used to copy objects across Amazon S3 buckets in different AWS Regions
- Same-Region replication (SRR) is used to copy objects across Amazon S3 buckets in the same AWS Region

How:

- Enable the AWS Region in the account
- Enable versioning on source and destination buckets
- Ensure S3 has permissions to both buckets
- Configure replication

Access Control Options

Identity-based policies



Resource-based policy



{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Sid": "AllowGroupToSeeBucketListInTheConsole",
 "Action": ["s3>ListAllMyBuckets"],
 "Effect": "Allow",
 "Resource": ["arn:aws:s3:::*"]
 }
]
}

Example identity-based policy

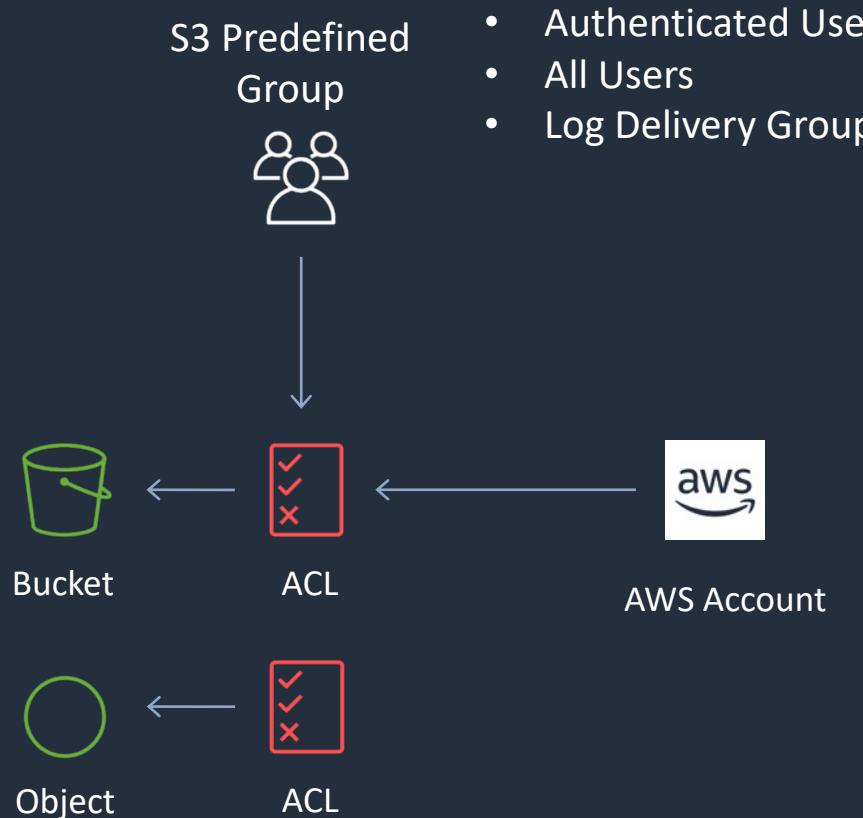
{
 "Version": "2012-10-17",
 "Id": "Policy1561964929358",
 "Statement": [
 {
 "Sid": "Stmt1561964454052",
 "Effect": "Allow",
 "Principal": {
 "AWS": "arn:aws:iam::515148227241:user/Paul"
 },
 "Action": "s3:*",
 "Resource": "arn:aws:s3:::dctcompany",
 "Condition": {
 "StringLike": {
 "s3:prefix": "Confidential/*"
 }
 }
 }
]
}

Example bucket policy

Example Policy - Allow IAM users access to their S3 home directory

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "s3>ListAllMyBuckets",  
                "s3:GetBucketLocation"  
            ],  
            "Resource": "*"  
        },  
        {  
            "Effect": "Allow",  
            "Action": "s3>ListBucket",  
            "Resource": "arn:aws:s3:::bucket-name",  
            "Condition": {  
                "StringLike": {  
                    "s3:prefix": [  
                        "",  
                        "home/",  
                        "home/${aws:username}/*"  
                    ]  
                }  
            }  
        },  
        {  
            "Effect": "Allow",  
            "Action": "s3:*",  
            "Resource": [  
                "arn:aws:s3:::bucket-name/home/${aws:username}",  
                "arn:aws:s3:::bucket-name/home/${aws:username}/*"  
            ]  
        }  
    ]  
}
```

Access Control Options



Example ACL

```
<?xml version="1.0" encoding="UTF-8"?>
<AccessControlPolicy xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Owner>
    <ID>*** Owner-Canonical-User-ID ***</ID>
    <DisplayName>owner-display-name</DisplayName>
  </Owner>
  <AccessControlList>
    <Grant>
      <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
               xsi:type="Canonical User">
        <ID>*** Owner-Canonical-User-ID ***</ID>
        <DisplayName>display-name</DisplayName>
      </Grantee>
      <Permission>FULL_CONTROL</Permission>
    </Grant>
  </AccessControlList>
</AccessControlPolicy>
```

Access Control List Permissions

Permissions	When granted on a bucket	When granted on an object
READ	Allows grantee to list the objects in the bucket	Allows grantee to read the object data and its metadata
WRITE	Allows grantee to create, overwrite, and delete any object in the bucket	Not applicable
READ_ACP	Allows grantee to read the bucket ACL	Allows grantee to read the object ACL
WRITE_ACP	Allows grantee to write the ACL for the applicable bucket	Allows grantee to write the ACL for the applicable object
FULL_CONTROL	Allows grantee the READ, WRITE, READ_ACP, and WRITE_ACP permissions on the bucket	Allows grantee the READ, READ_ACP, and WRITE_ACP permissions on the object

S3 Encryption

Server-side encryption with S3 managed keys (SSE-S3)

- S3 managed keys
- Unique object keys
- Master key
- AES 256



Encryption / decryption



Server-side encryption with AWS KMS managed keys (SSE-KMS)



- KMS managed keys
- Customer master keys
- CMK can be customer generated



Encryption / decryption



Server-side encryption with client provided keys (SSE-C)



Encryption / decryption



- Client managed keys
- Not stored on AWS



Client-side encryption



Encryption / decryption



- Client managed keys
- Not stored on AWS



S3 Default Encryption

- Amazon S3 default encryption provides a way to set the default encryption behavior for an S3 bucket
- You can set default encryption on a bucket so that all new objects are encrypted when they are stored in the bucket
- The objects are encrypted using server-side encryption
- Amazon S3 encrypts objects before saving them to disk and decrypts them when the objects are downloaded
- There is no change to the encryption of objects that existed in the bucket before default encryption was enabled

Prevent uploads of unencrypted objects

```
{  
    "Version": "2012-10-17",  
    "Id": "PutObjPolicy",  
    "Statement": [  
        {  
            "Sid": "DenyIncorrectEncryptionHeader",  
            "Effect": "Deny",  
            "Principal": "*",  
            "Action": "s3:PutObject",  
            "Resource": "arn:aws:s3:::<bucket_name>/*",  
            "Condition": {  
                "StringNotEquals": {  
                    "s3:x-amz-server-side-encryption": "AES256"  
                }  
            }  
        },  
        {  
            "Sid": "DenyUnEncryptedObjectUploads",  
            "Effect": "Deny",  
            "Principal": "*",  
            "Action": "s3:PutObject",  
            "Resource": "arn:aws:s3:::<bucket_name>/*",  
            "Condition": {  
                "Null": {  
                    "s3:x-amz-server-side-encryption": true  
                }  
            }  
        }  
    ]  
}
```

Enforces encryption using SSE-S3

For SSE-KMS use "aws:kms"

Example PUT request

```
PUT /example-object HTTP/1.1  
Host: myBucket.s3.amazonaws.com  
Date: Wed, 8 Jun 2016 17:50:00 GMT  
Authorization: authorization string  
Content-Type: text/plain  
Content-Length: 11434  
x-amz-meta-author: Janet  
Expect: 100-continue  
x-amz-server-side-encryption: AES256  
[11434 bytes of object data]
```

S3 Presigned URLs

AWS S3 CLI command to generate a presigned URL



```
aws s3 presign s3://dct-data-bucket/cool_image.jpeg
```

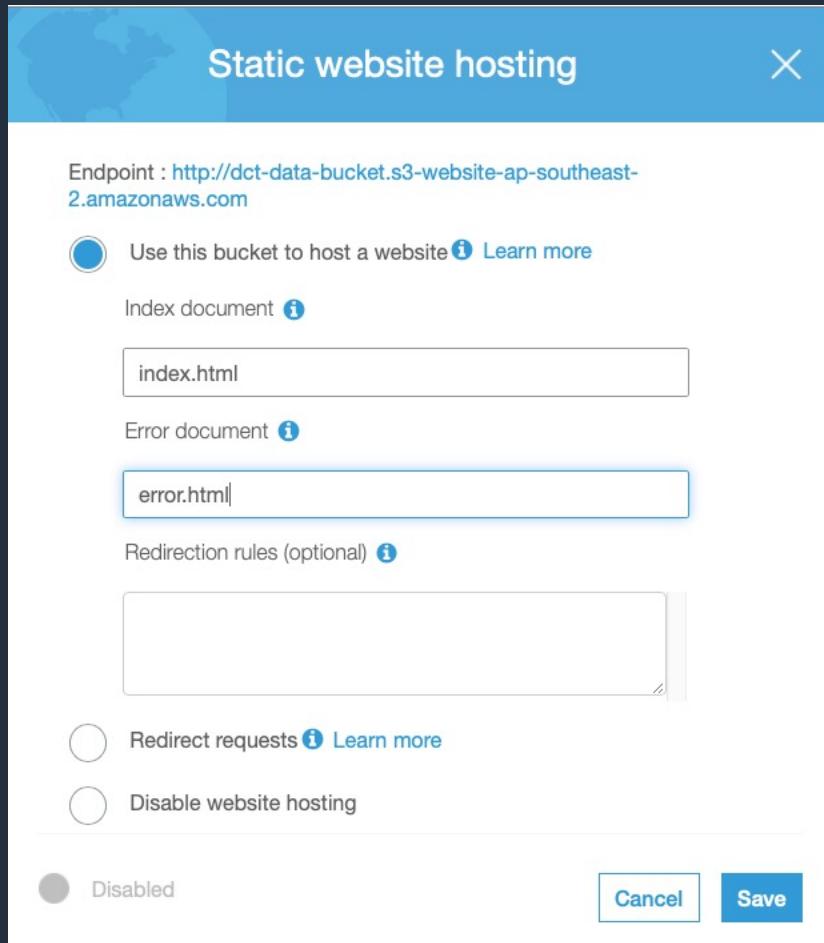


```
https://dct-data-bucket.s3.ap-southeast-2.amazonaws.com/cool_image.jpeg?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=AKIA3KSVPHP6MAHNW5YH%2F20200909%2Fap-southeast-2%2Fs3%2Faws4_request&X-Amz-Date=20200909T053538Z&X-Amz-Expires=3600&X-Amz-SignedHeaders=host&X-Amz-Signature=8b74653beee371da07a73dfdb4ff6883742383afa528aec5c95c326c97764db
```

This is the response; the URL expires after 1 hour

S3 Static Websites

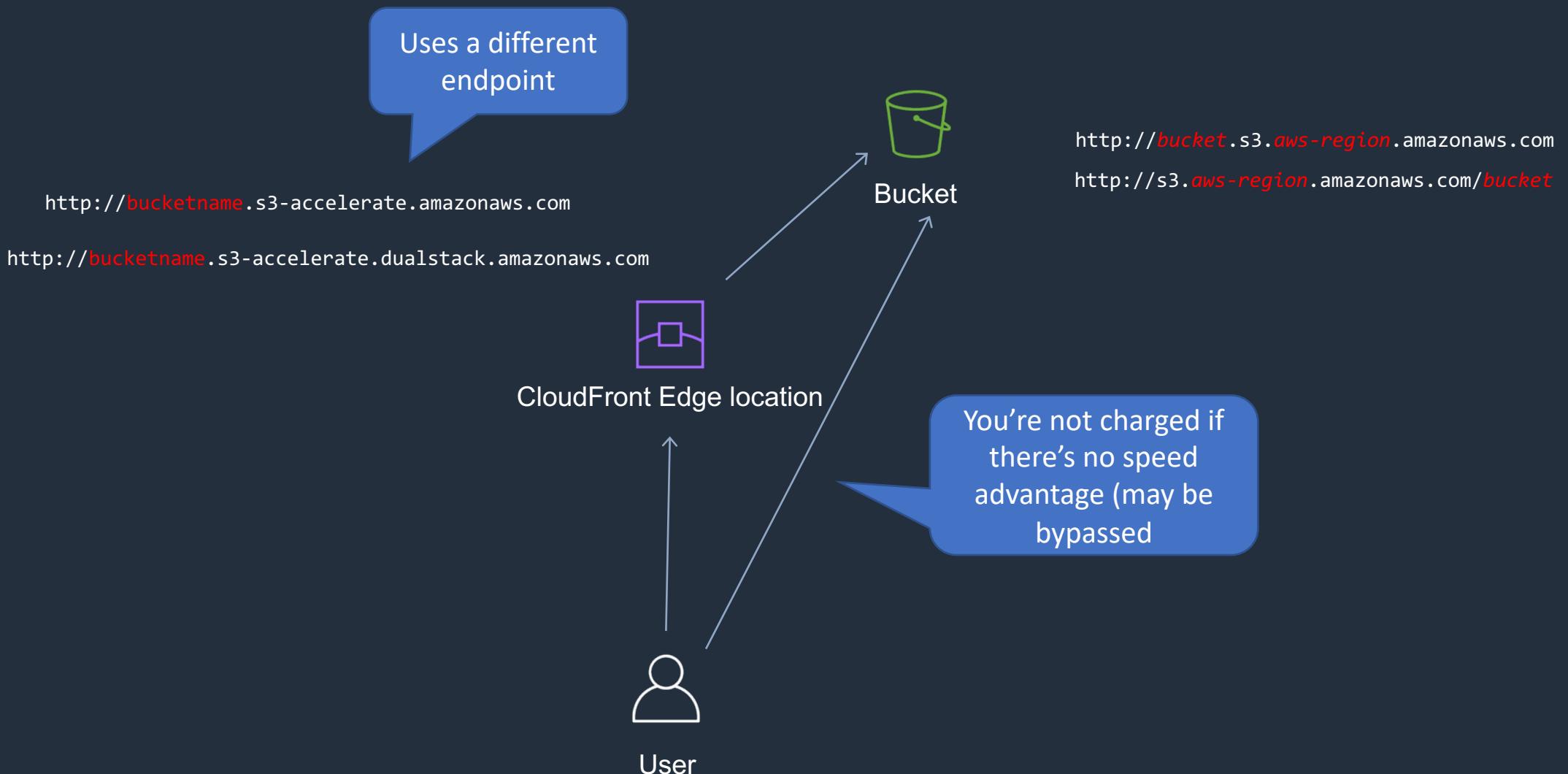
Webpages include static content and can include client-side scripts



Must enable public access and use a bucket policy that grants s3:GetObject

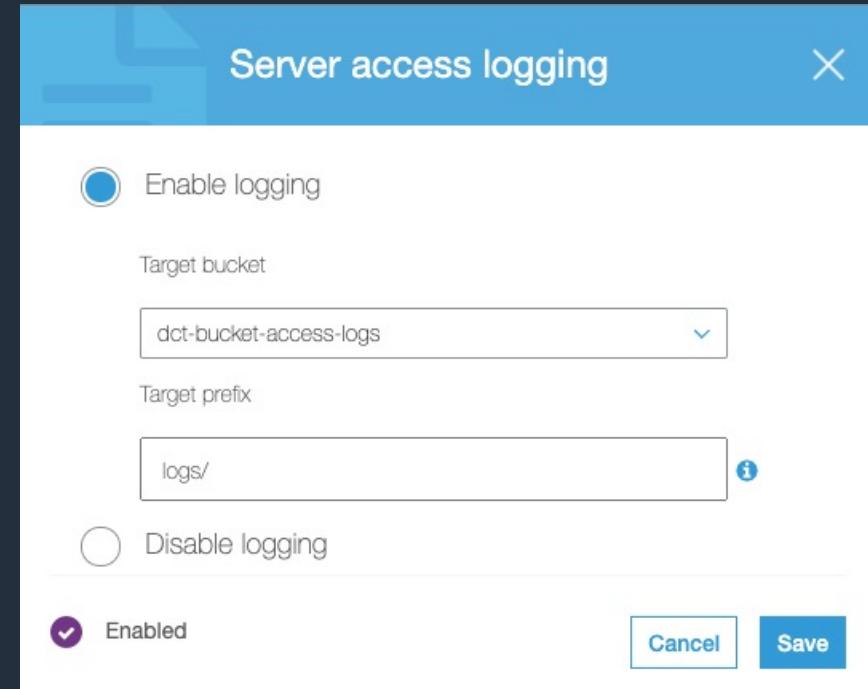
```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "PublicReadGetObject",  
            "Effect": "Allow",  
            "Principal": "*",  
            "Action": [  
                "s3:GetObject"  
            ],  
            "Resource": [  
                "arn:aws:s3:::example.com/*"  
            ]  
        }  
    ]  
}
```

S3 Transfer Acceleration



S3 Server Access Logging

- Provides detailed records for the requests that are made to a bucket
- Details include the requester, bucket name, request time, request action, response status, and error code (if applicable)
- Disabled by default
- Only pay for the storage space used
- Must configure a separate bucket as the destination (can specify a prefix)
- Must grant write permissions to the Amazon S3 Log Delivery group on destination bucket



S3 Event Notifications

- Sends notifications when events happen in buckets
- Destinations include:
 - Amazon Simple Notification Service (SNS) topics
 - Amazon Simple Queue Service (SQS) queues
 - AWS Lambda

The screenshot shows the AWS S3 Events configuration interface. At the top, there's a header with 'Events' and a close button. Below it is a toolbar with '+ Add notification', 'Delete', and 'Edit' buttons. A table header row has columns for 'Name', 'Events', 'Filter', and 'Type'. A single row is visible under 'New event' with the name 'Notify for Uploads'. The 'Events' section contains several checkboxes for different S3 events, with 'PUT' and 'POST' checked. Other options like 'All object delete events' and 'Restore initiated' are available but unchecked. The 'Prefix' field contains 'e.g. images/'. The 'Suffix' field contains 'e.g. .jpg'. Under 'Send to', a dropdown menu is set to 'SNS Topic', and another dropdown below it is set to 'NotifyMe'. The 'SNS' dropdown also has a 'NotifyMe' option selected.

S3 Glacier Vault Lock and Vault Access Policies

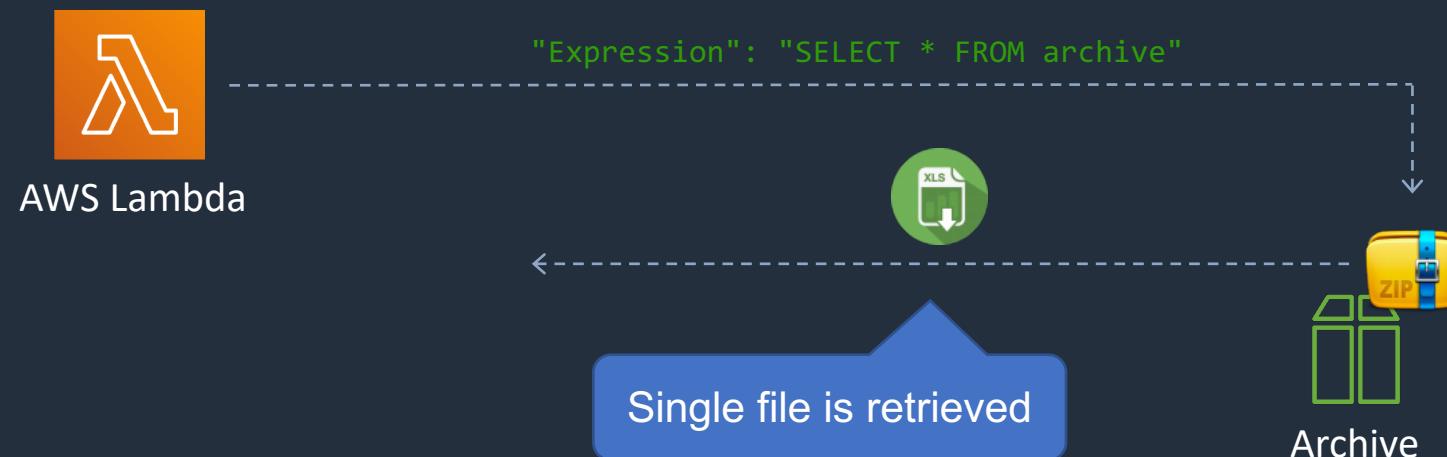
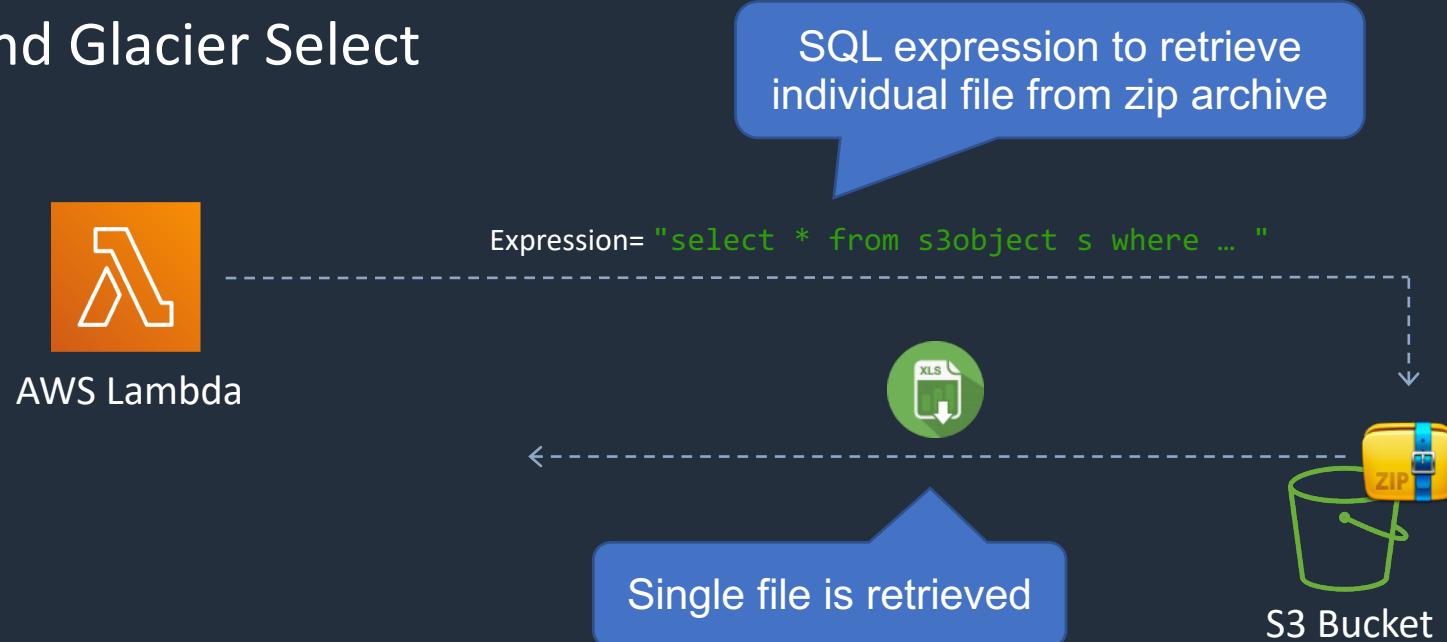
S3 Glacier Vault Lock:

- S3 Glacier Vault Lock enforces compliance controls for S3 Glacier vaults with a vault lock policy
- Can specify controls such as “write once read many” (WORM) in a vault lock policy and lock the policy from future edits
- Once locked, the policy can no longer be changed

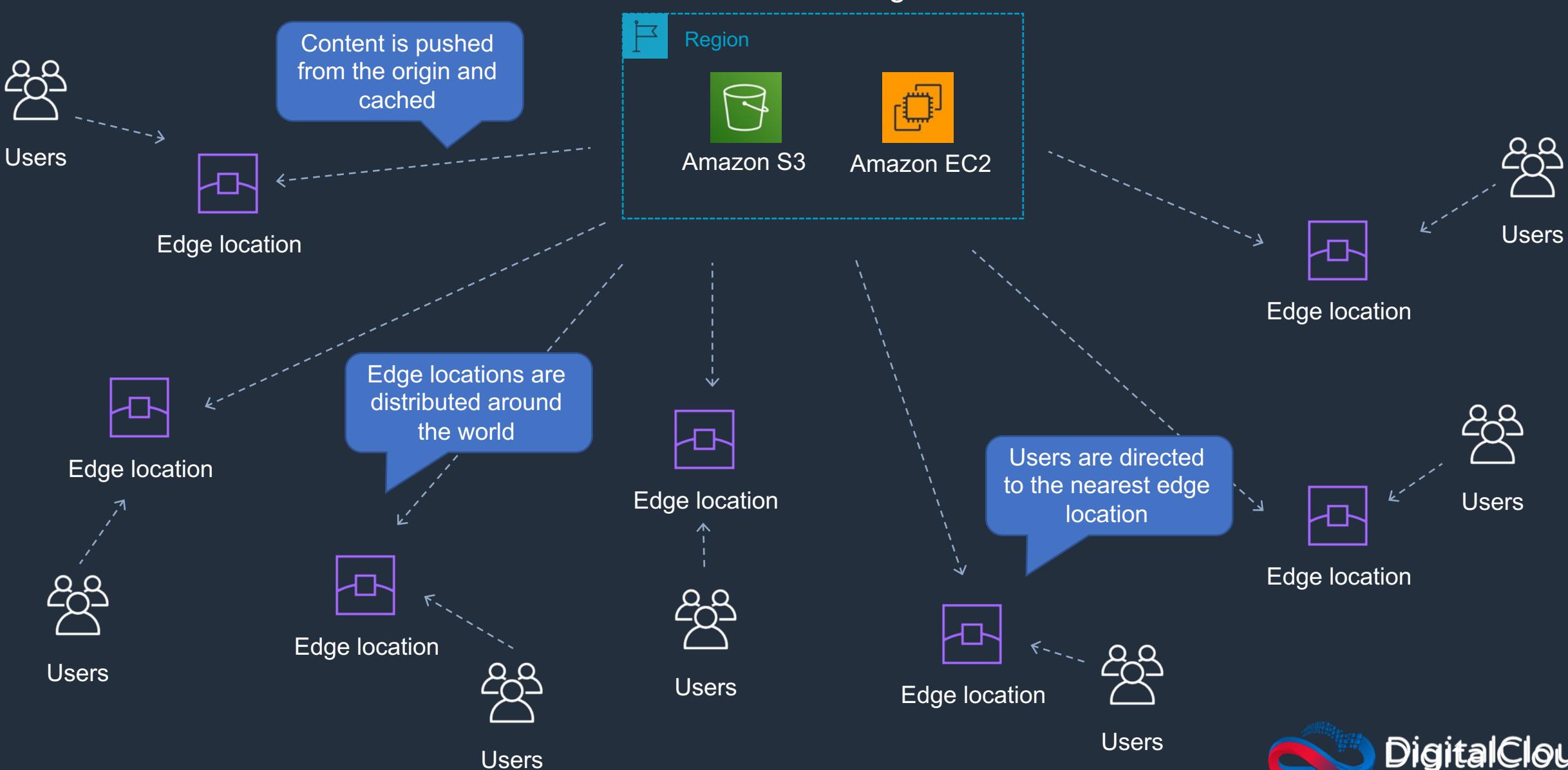
S3 Glacier Vault Access Policy:

- Cannot be locked to prevent future changes
- Use for access controls that are not compliance related, temporary, and subject to frequent modification
- Can be used with a vault lock policy

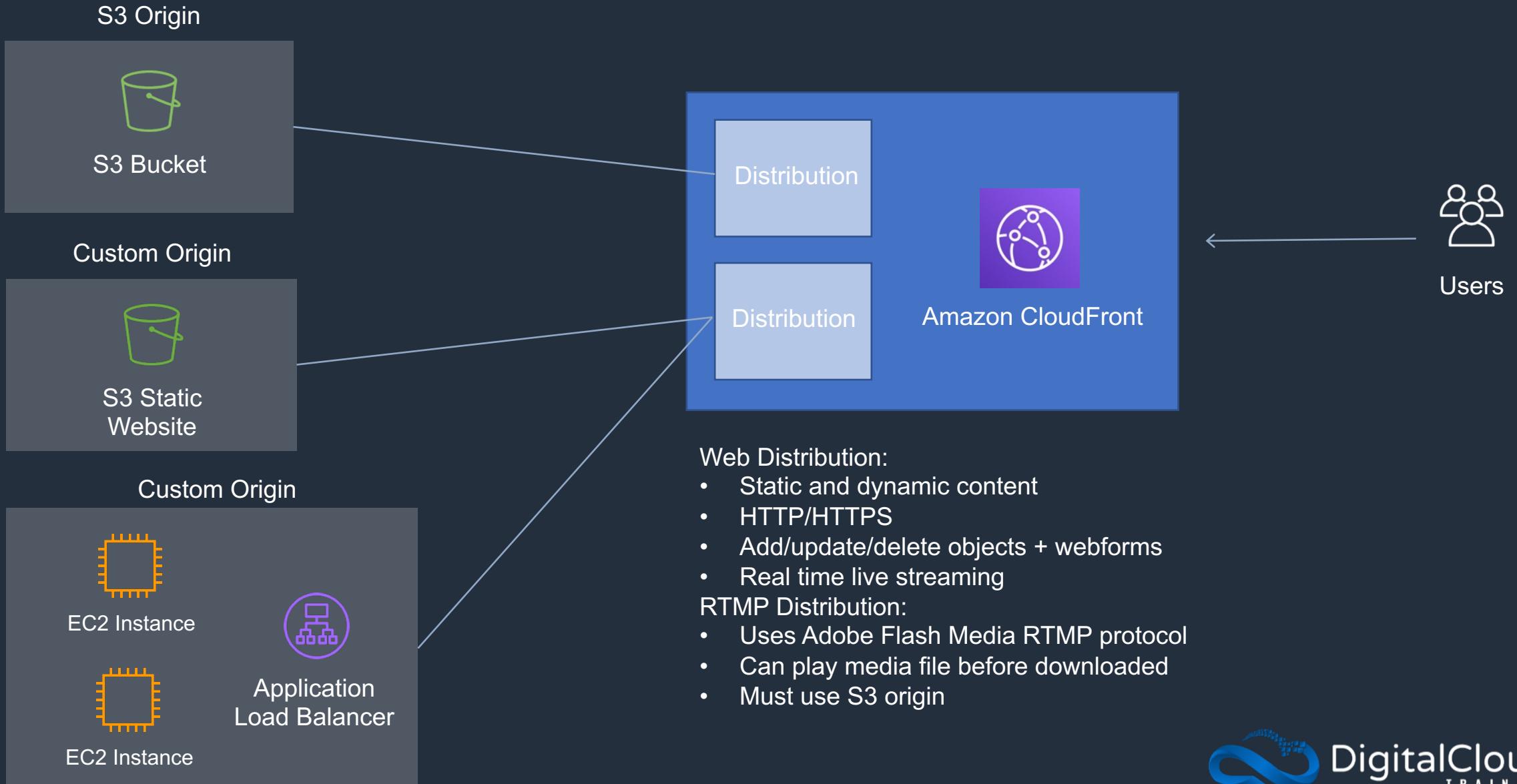
S3 Select and Glacier Select



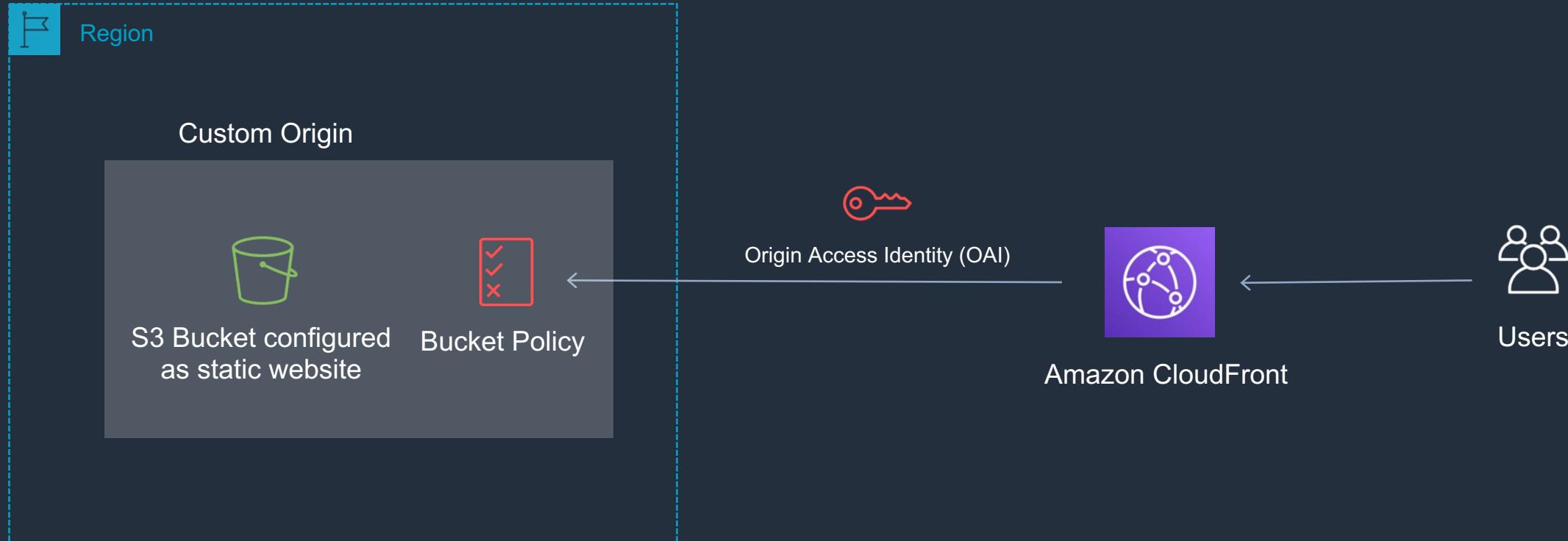
Amazon CloudFront



CloudFront Distribution and Origins



CloudFront with S3 Static Website



Improving the Cache Hit Ratio

- A good cache hit ratio means more requests are served from the cache
- Methods of improving the cache hit ratio include:
 - Use the Cache-Control max-age directive to increase the time objects remain in the cache
 - Use Origin Shield
 - Forward only the query string parameters for which your origin will return unique objects
 - Configure CloudFront to forward only specified cookies instead of forwarding all cookies
 - Configure CloudFront to forward and cache based on only specified headers instead of forwarding and caching based on all headers

Exam Scenarios

Exam Scenario	Solution
Static website on Amazon S3 with custom domain name	Requires that the bucket name matches the DNS name / record set name in Route 53
503 errors experienced with new site and thousands of user	Request rate is too high
Discrepancy with number of objects in bucket console vs CloudWatch	Use Amazon S3 Inventory to properly determine the number of objects in a bucket
Need to enforce encryption on all objects uploaded to bucket	Use a bucket policy with a "Condition": { "Bool": { "aws:SecureTransport": "false" } statement for PutObject and with the resource set to the bucket

Exam Scenarios

Exam Scenario	Solution
Unauthorized users tried to connect to S3 buckets. Need to know which buckets are targeted and who is trying to get access	Use S3 server access logs and Athena to query for HTTP 403 errors and look for IAM user or role making requests
Need to provide access to third-party to S3 bucket and must limit amount of access. List of users changes a lot	Use a pre-signed URL allowing access to the specific files
Need to protect S3 data from ransomware attacks that encrypt data	Enable S3 versioning
After enabling MFA on a bucket, what operations will require MFA authentication?	Permanently removing object versions and suspending versioning on the bucket

Exam Scenarios

Exam Scenario	Solution
Files are downloaded from S3, edited and uploaded with same file name. Sometimes they are accidentally modified or deleted	To allow recovery enable versioning on the bucket
Existing application uses EC2, RDS, EFS and S3. Need to enable encryption	Can enable encryption only on S3 (as already deployed)
Static website deployed but "HTTP 403 Forbidden" message received	Add bucket policy granting everyone read access to objects
Application on EC2 must save files to Amazon S3 and needs access	Create an IAM role for S3 access and attach to EC2 instance

Exam Scenarios

Exam Scenario	Solution
History of revisions to files stored in an S3 bucket must be maintained	Implement S3 versioning
Large volume of log files stored in S3 bucket and processed daily	Most cost-effective option is S3 standard
Need to restrict S3 bucket access to same account after previously shared with other account	Change ACL to restrict only to bucket owner
Static content is served from Amazon S3 with long loading times	Use CloudFront to cache for better performance
Need to use custom domain name with CloudFront	Create an alias record in Route 53 pointing to the distribution URL

Exam Scenarios

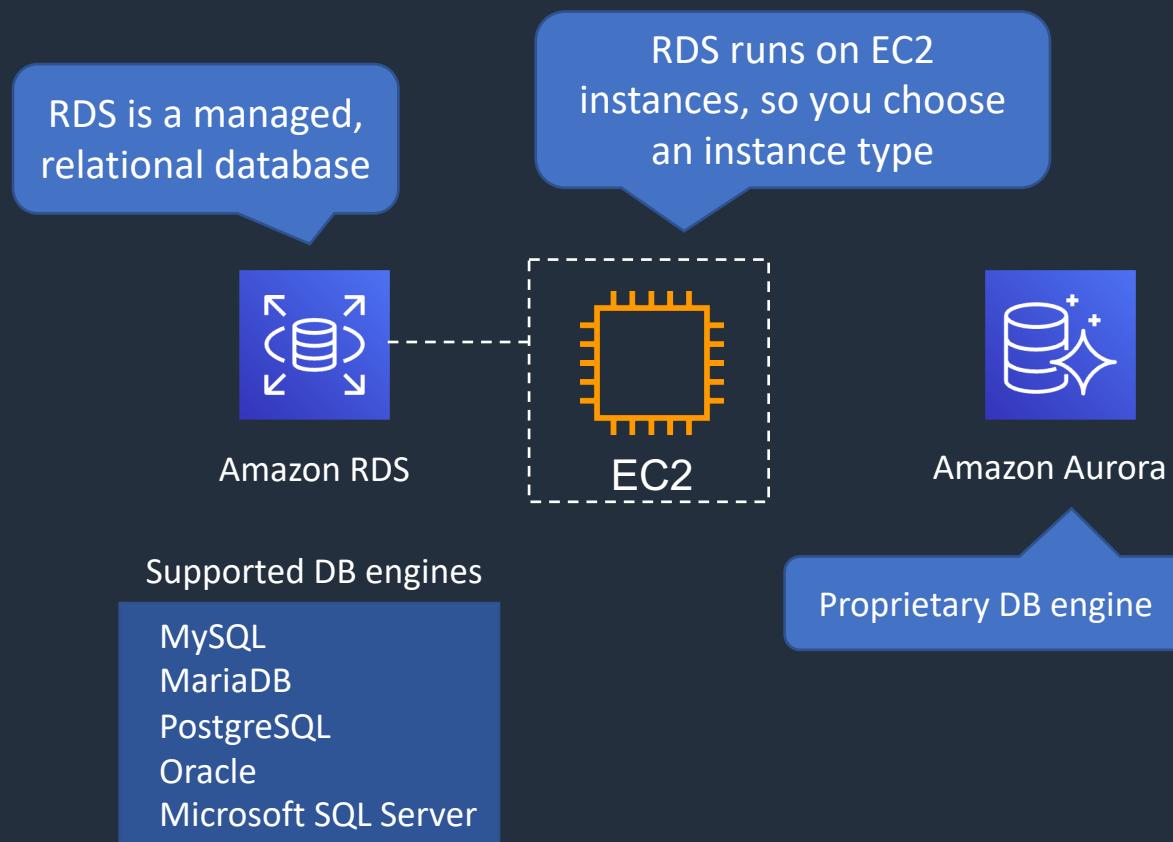
Exam Scenario	Solution
CloudFront in front of ALB and EC2 and logging enabled. Need to view logs for HTTP layer 7 status codes	Check ALB access logs and CloudFront access logs
App running on EC2 with RDS multi-AZ has static content on S3. Need to improve performance as load testing slowed it down	Use CloudFront to cache the content
Need to secure S3 bucket that is used with CloudFront	Use an OAI and grant permissions to read objects in the bucket
Website with dynamic content and need to restrict access from certain countries and regions	Use Amazon CloudFront geo-restriction and Amazon Route 53 geolocation routing

SECTION 12

Databases: Amazon RDS and ElastiCache

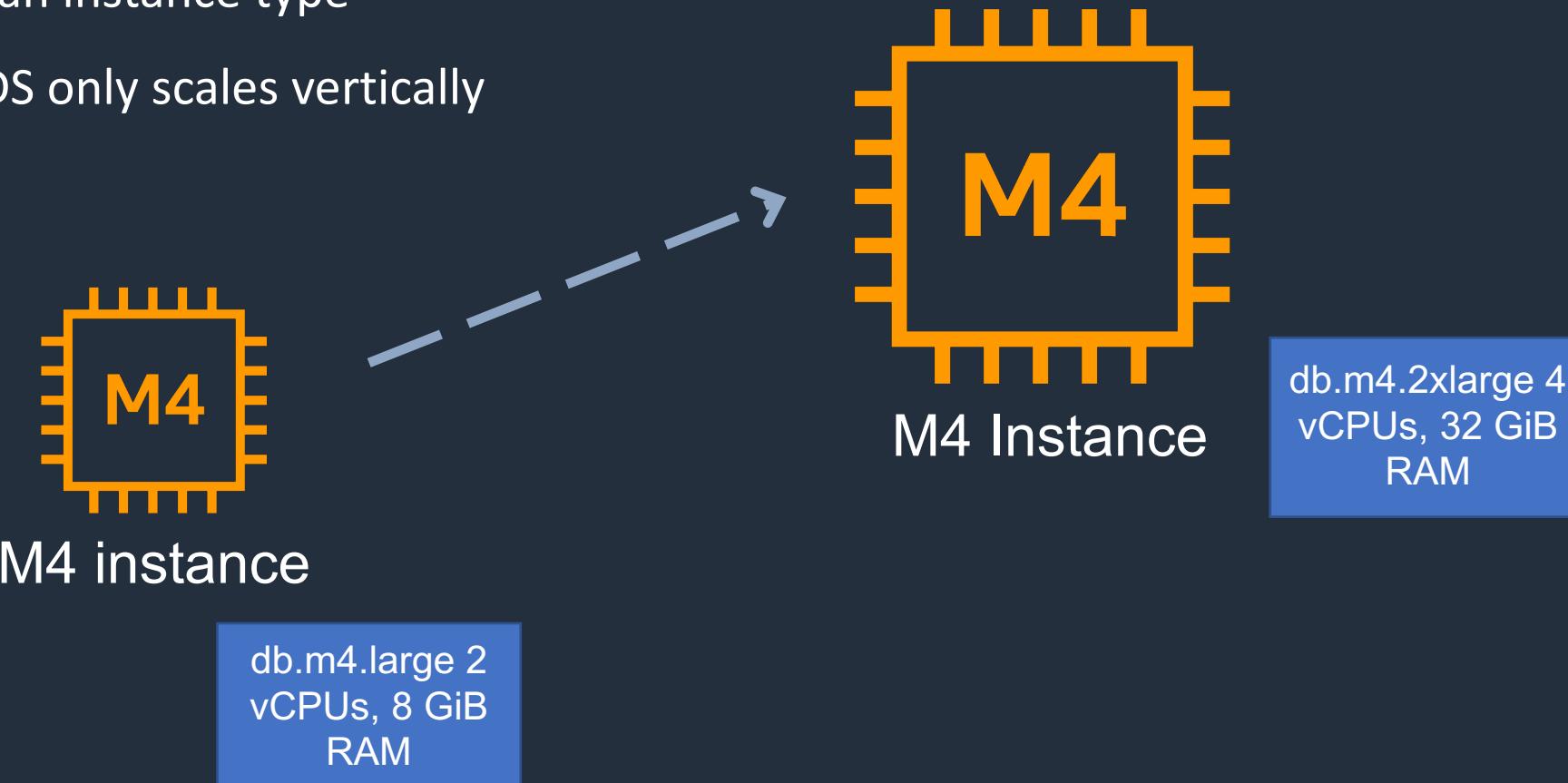
Amazon Relational Database Service (RDS)

- Managed relational database service including:
 - Backups
 - Software patching
 - Automatic failure detection
 - Recovery
- Backup options include automated backups and manual snapshots

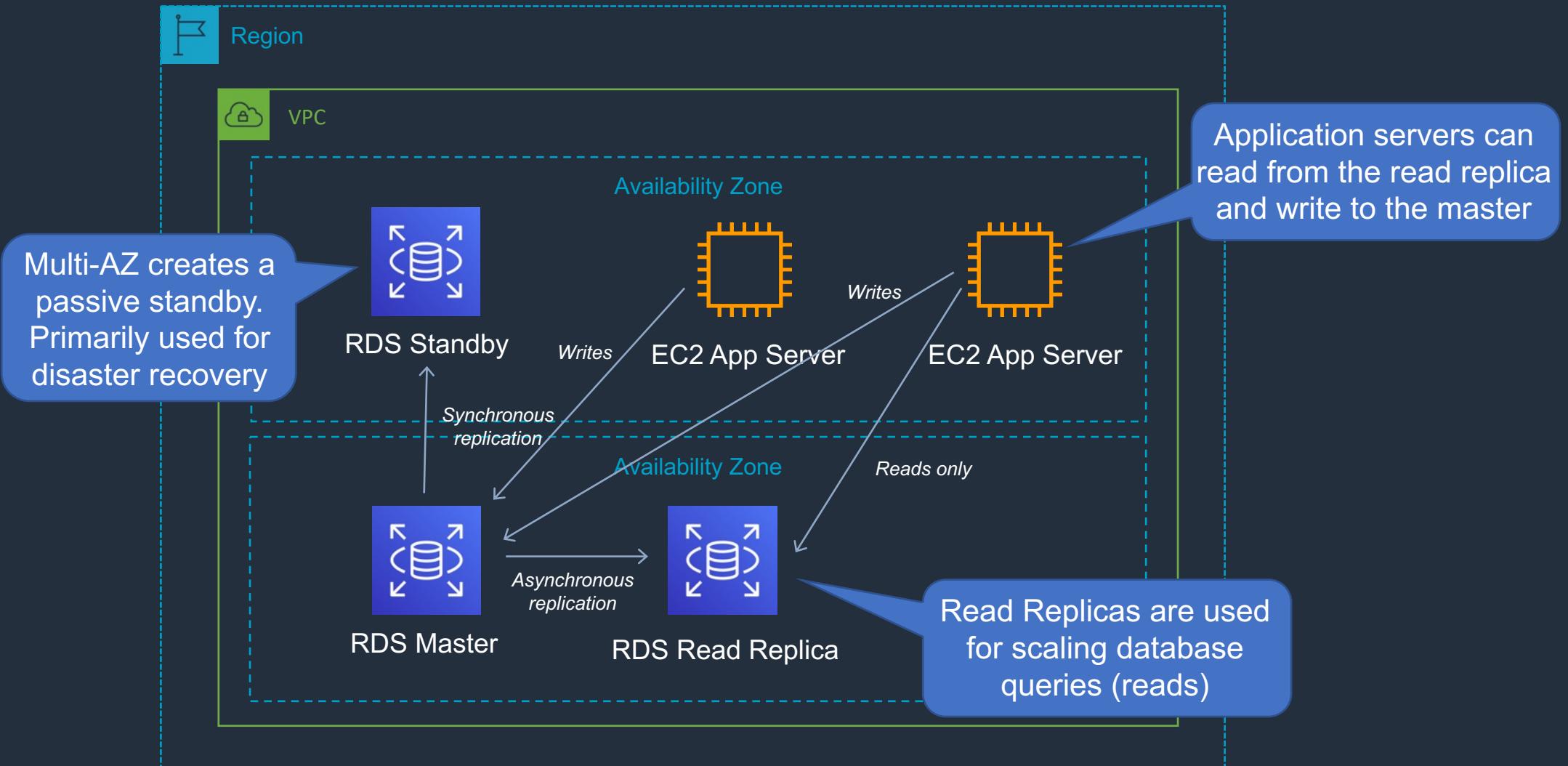


Scaling Vertically with Amazon RDS

- RDS runs on Amazon EC2 instances
- Must choose an instance type
- For writes, RDS only scales vertically



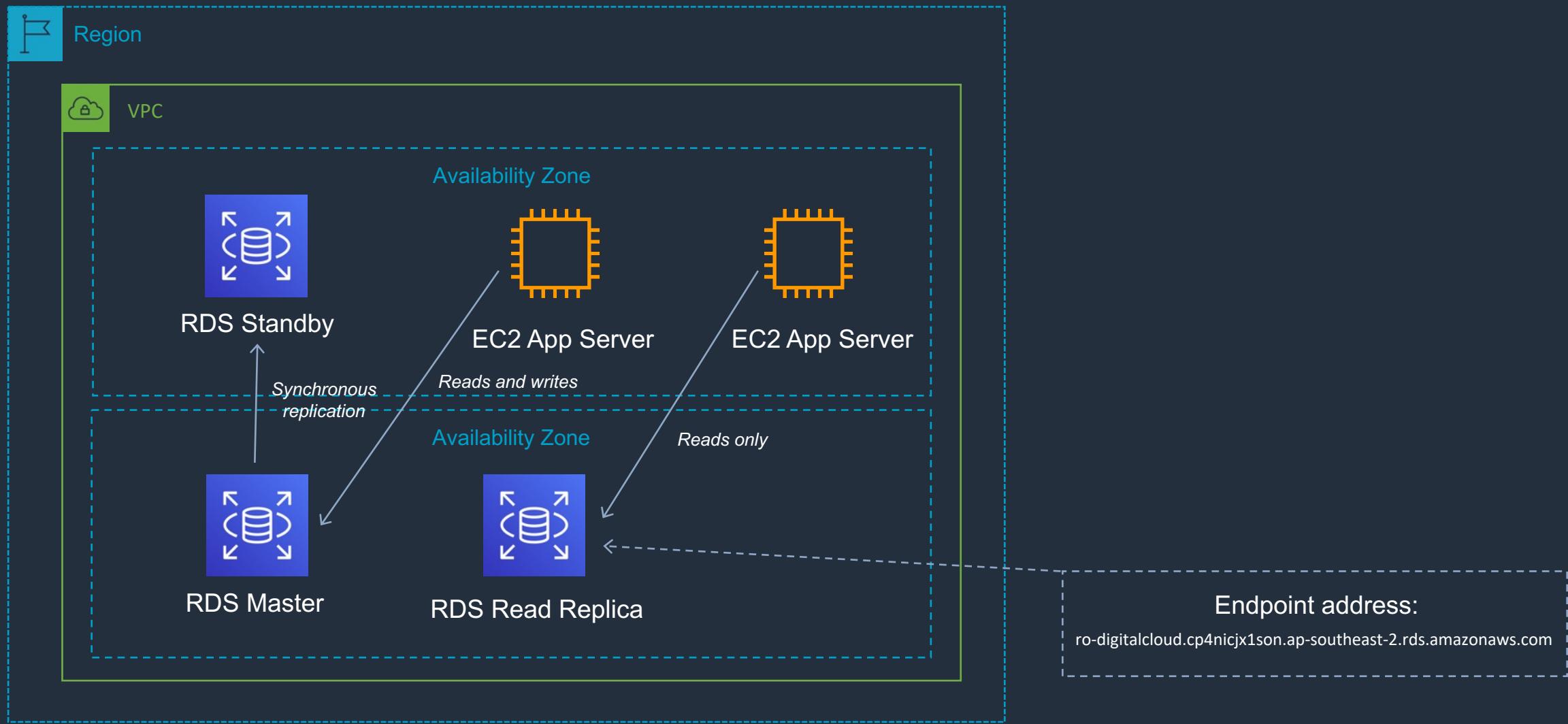
Amazon RDS – Disaster Recovery (DR) and Scaling Out (horizontally)



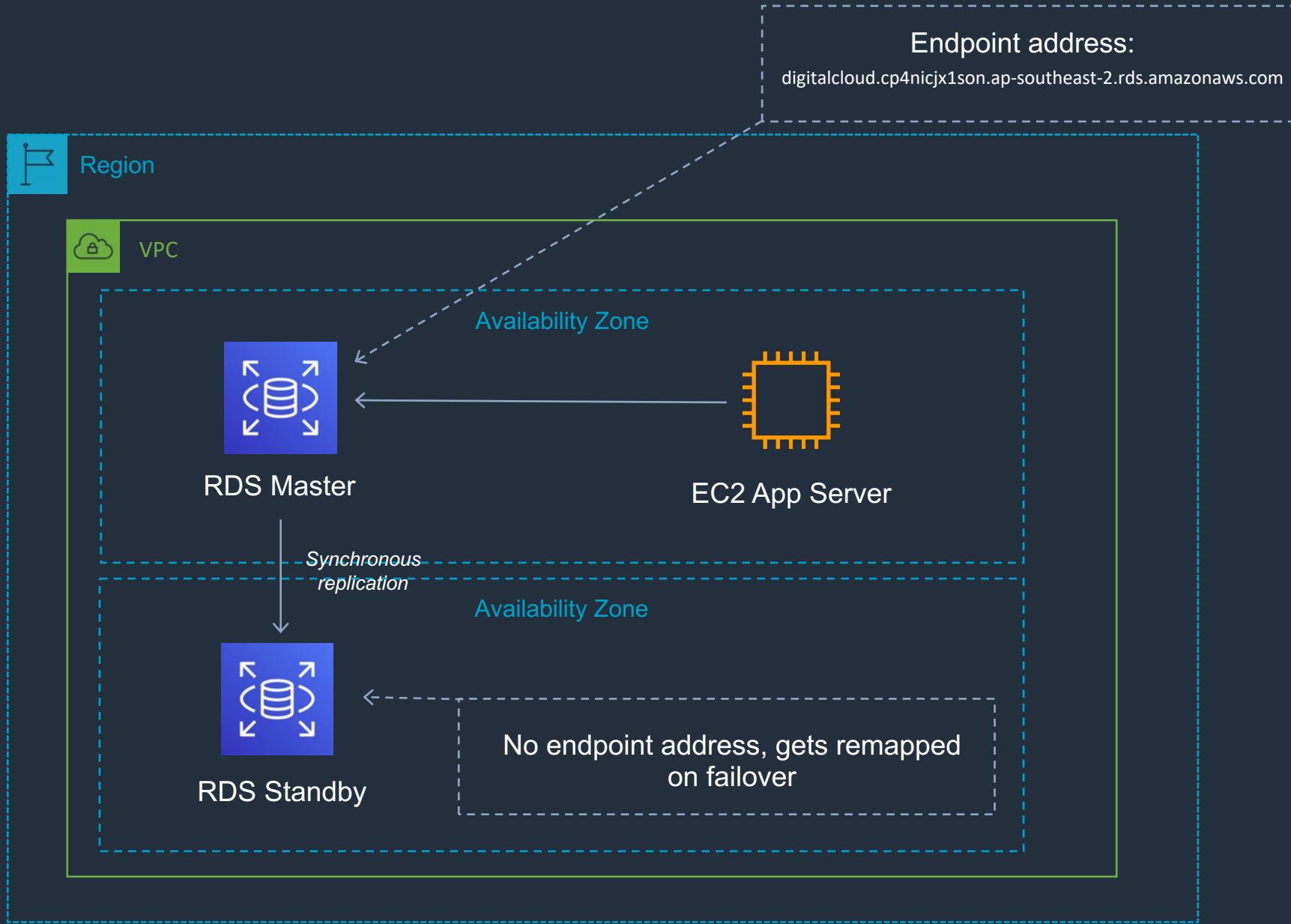
Amazon RDS – Multi-AZ and Read Replicas

Multi-AZ Deployments	Read Replicas
Synchronous replication – highly durable	Asynchronous replication – highly scalable
Only database engine on primary instance is active	All read replicas are accessible and can be used for read scaling
Automated backups are taken from standby	No backups configured by default
Always span two Availability Zones within a single Region	Can be within an Availability Zone, Cross-AZ, or Cross-Region
Database engine version upgrades happen on primary	Database engine version upgrade is independent from source instance
Automatic failover to standby when a problem is detected	Can be manually promoted to a standalone database instance

Amazon RDS Read Replicas



Amazon RDS Multi-AZ

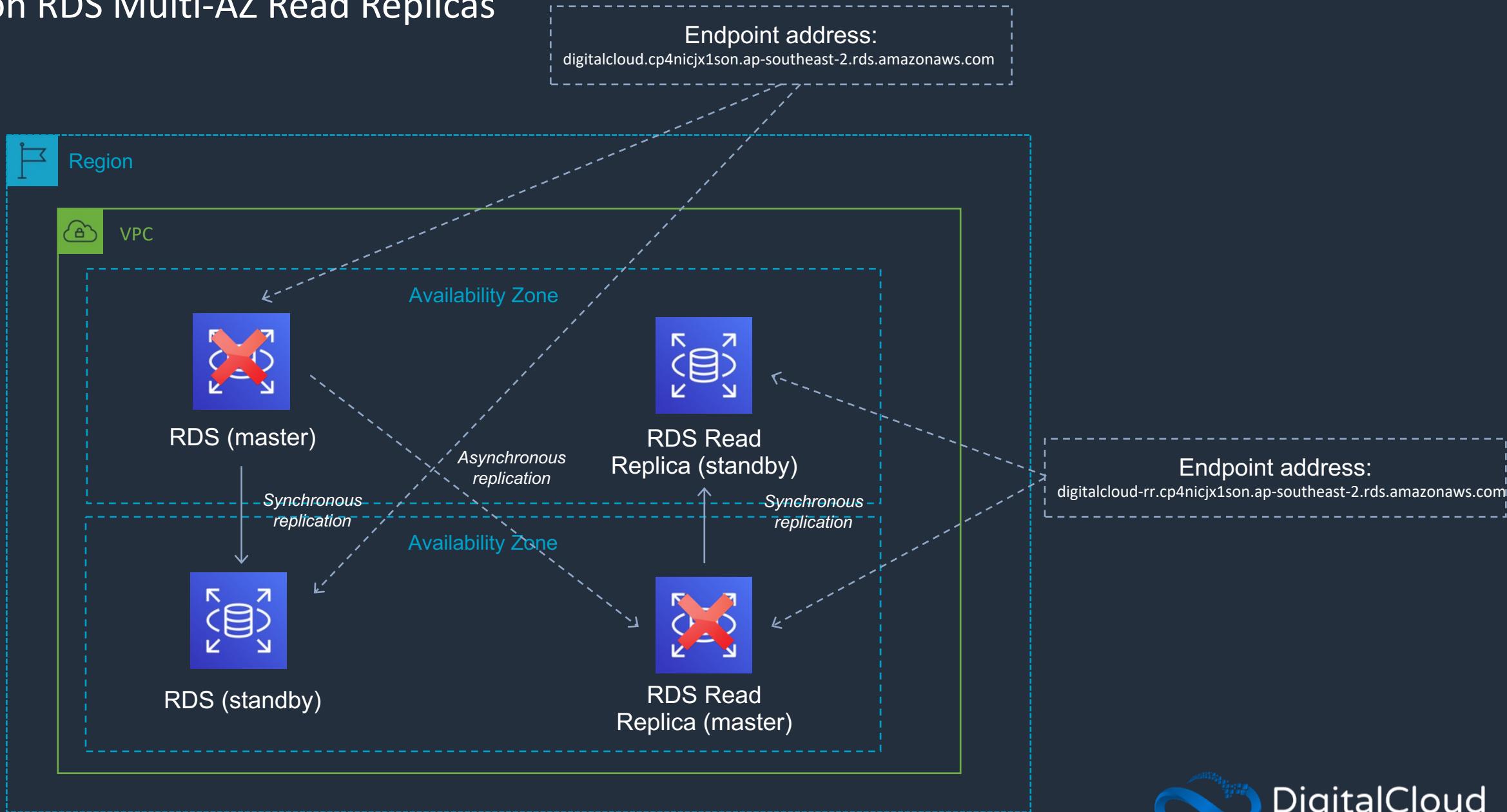


Amazon RDS – Multi-AZ Failover

Failover occurs in the following situations:

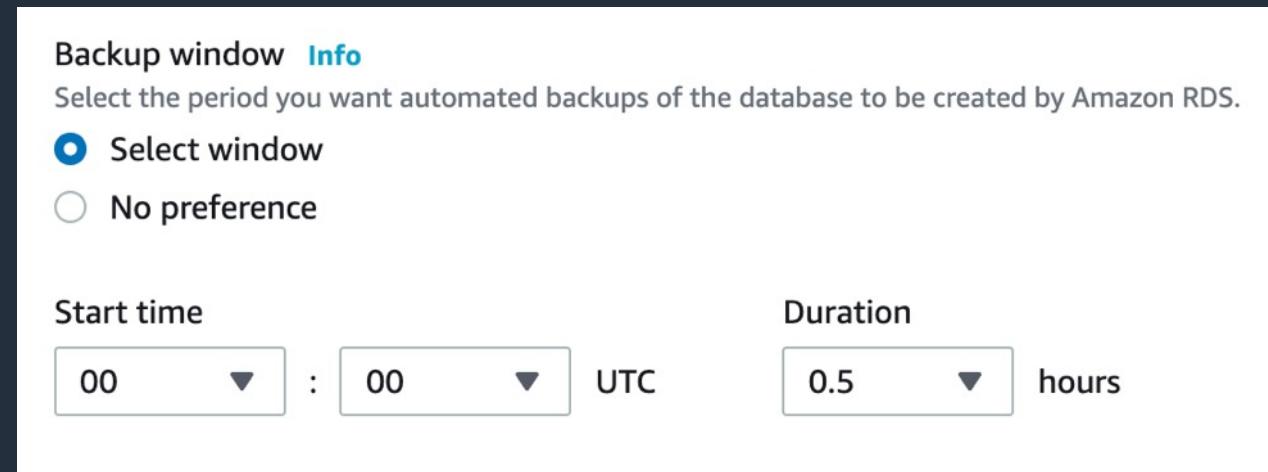
- An Availability Zone outage
- The primary DB instance fails
- The DB instance's server type is changed
- The operating system of the DB instance is undergoing software patching
- A manual failover of the DB instance was initiated using Reboot with failover

Amazon RDS Multi-AZ Read Replicas



Amazon RDS – Automated Backups

- Creates a point in time snapshot of the database
- Backup retention is 0 days to 35 days, default is 7 days
- 0 days switches automated backups off
- You can restore to any point in time during the retention period
- Restoring from backup creates a new DB instance
- You can configure the backup window (cannot overlap with the maintenance window)

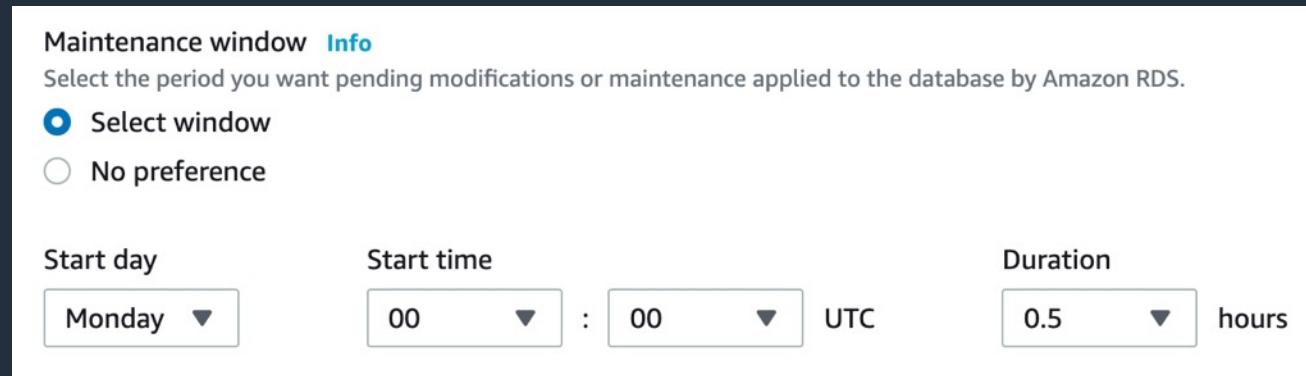


Amazon RDS – Snapshots

- Backs up the entire DB instance, not just individual databases
- For single-AZ DB instances there is a brief suspension of I/O
- For Multi-AZ SQL Server, I/O activity is briefly suspended on primary
- For Multi-AZ MariaDB, MySQL, Oracle and PostgreSQL the snapshot is taken from the standby
- Snapshots do not expire (no retention period)

Amazon RDS – Maintenance Windows

- Operating system and DB patching can require taking the database offline
- These tasks take place during a maintenance window
- By default a weekly maintenance window is configured
- You can choose your own maintenance window



- Required patching includes security and reliability updates
- Deferred DB instance modifications also take place during the maintenance window

Amazon RDS – Maintenance for Multi-AZ Deployments

- For Multi-AZ deployments maintenance uses the following process:
 1. Perform maintenance on standby
 2. Promote standby to primary
 3. Perform maintenance on the old primary (new standby)
- Modifying the DB engine affects both primary and secondary at the same time

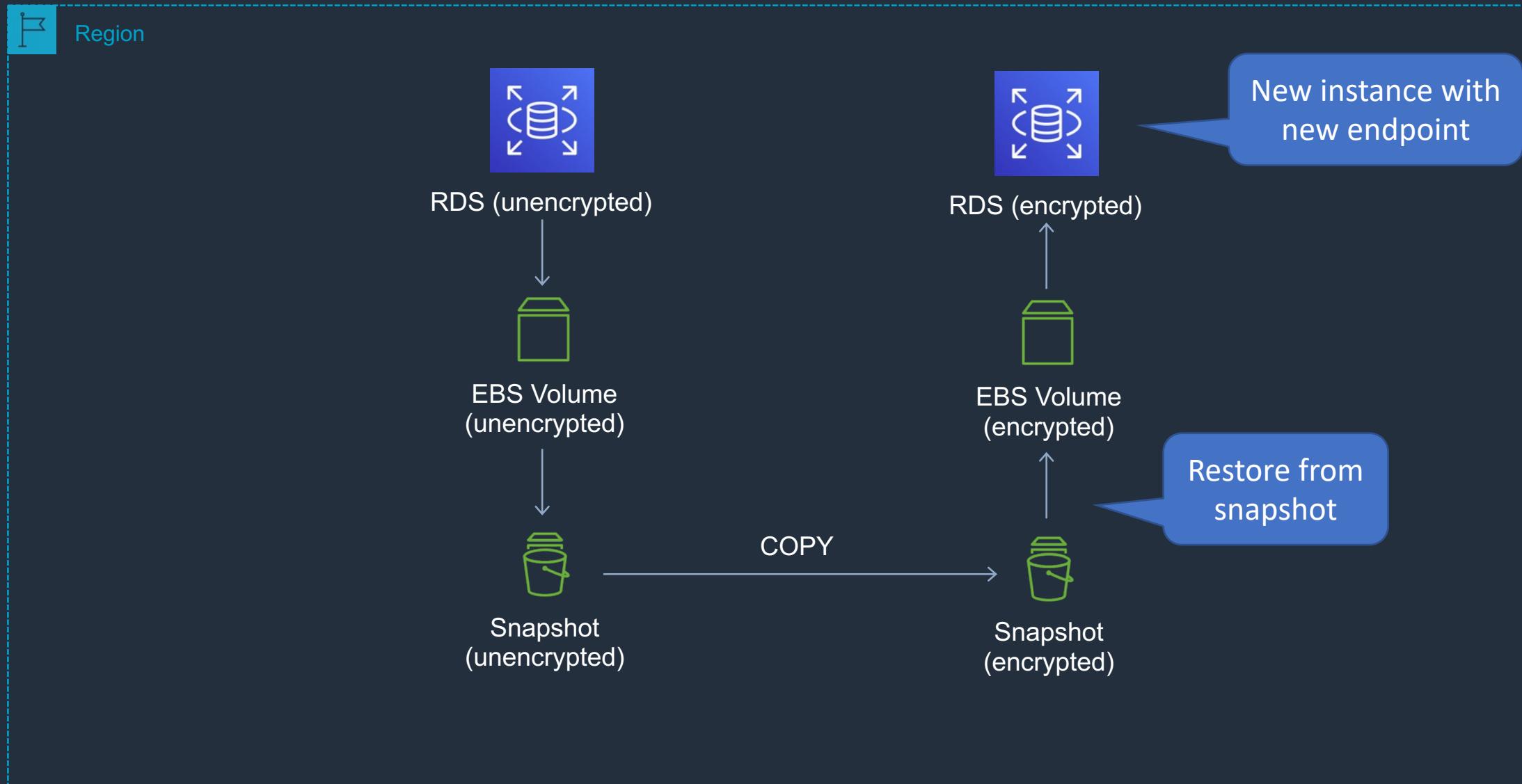
Amazon RDS Encryption

- Encryption at rest can be enabled – includes DB storage, backups, read replicas and snapshots
- You can only enable encryption for an Amazon RDS DB instance when you create it, not after the DB instance is created
- DB instances that are encrypted can't be modified to disable encryption
- Uses AES 256 encryption and encryption is transparent with minimal performance impact
- RDS for Oracle and SQL Server is also supported using Transparent Data Encryption (TDE) (may have performance impact)
- AWS KMS is used for managing encryption keys

Amazon RDS Encryption

- You can't have an encrypted read replica of an unencrypted DB instance or an unencrypted read replica of an encrypted DB instance
- Read replicas of encrypted master instances are encrypted
- The same key is used if in the same Region as the master
- If the read replica is in a different Region, a different key is used
- You can't restore an unencrypted backup or snapshot to an encrypted DB instance

Encrypting an unencrypted RDS DB instance



Amazon RDS Monitoring

- Amazon RDS monitoring tools include:
 - Amazon RDS Events – notifications for changes to DB instance, DB snapshot, DB parameter group, or DB security group
 - Database log files – View, download, or watch database log files using the Amazon RDS console or Amazon RDS API operations
 - Amazon RDS Enhanced Monitoring – Look at metrics in real time for the operating system
 - Amazon RDS Performance Insights – Assess the load on your database, and determine when and where to take action
 - Amazon RDS Recommendations – Look at automated recommendations for database resources, such as DB instances, read replicas, and DB parameter groups

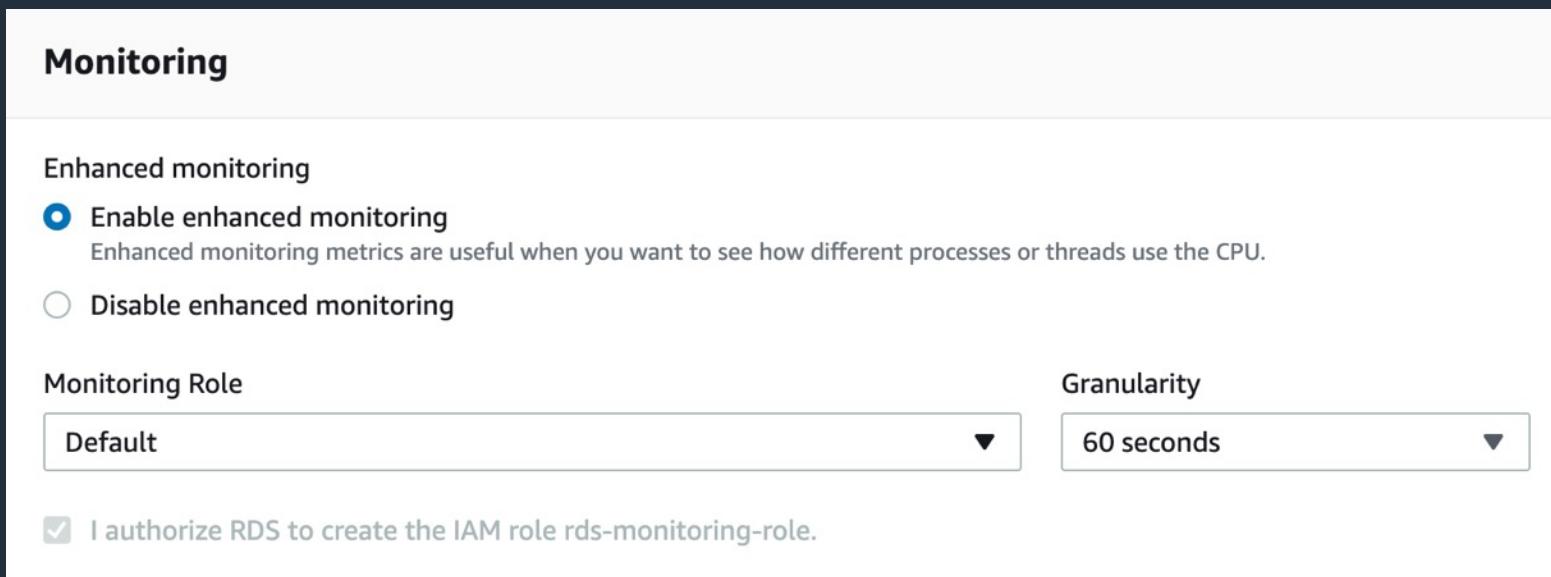
Amazon RDS Monitoring

- Additional monitoring tools include:
 - Amazon CloudWatch Metrics – Amazon RDS automatically sends metrics to CloudWatch every minute for each active database
 - Amazon CloudWatch Alarms – You can watch a single Amazon RDS metric over a specific time period
 - Amazon CloudWatch Logs – Most DB engines enable you to monitor, store, and access your database log files in CloudWatch Logs
 - Amazon CloudWatch Events and Amazon EventBridge – You can automate AWS services and respond to system events such as application availability issues or resource changes
 - AWS CloudTrail – You can view a record of actions taken by a user, role, or an AWS service in Amazon RDS

Amazon RDS Monitoring

➤ Enhanced Monitoring:

- Provides metrics in real time for the operating system (OS) that the DB instance runs on
- Installs an agent on the DB instance to collect the metrics
- Metrics can be viewed in the console



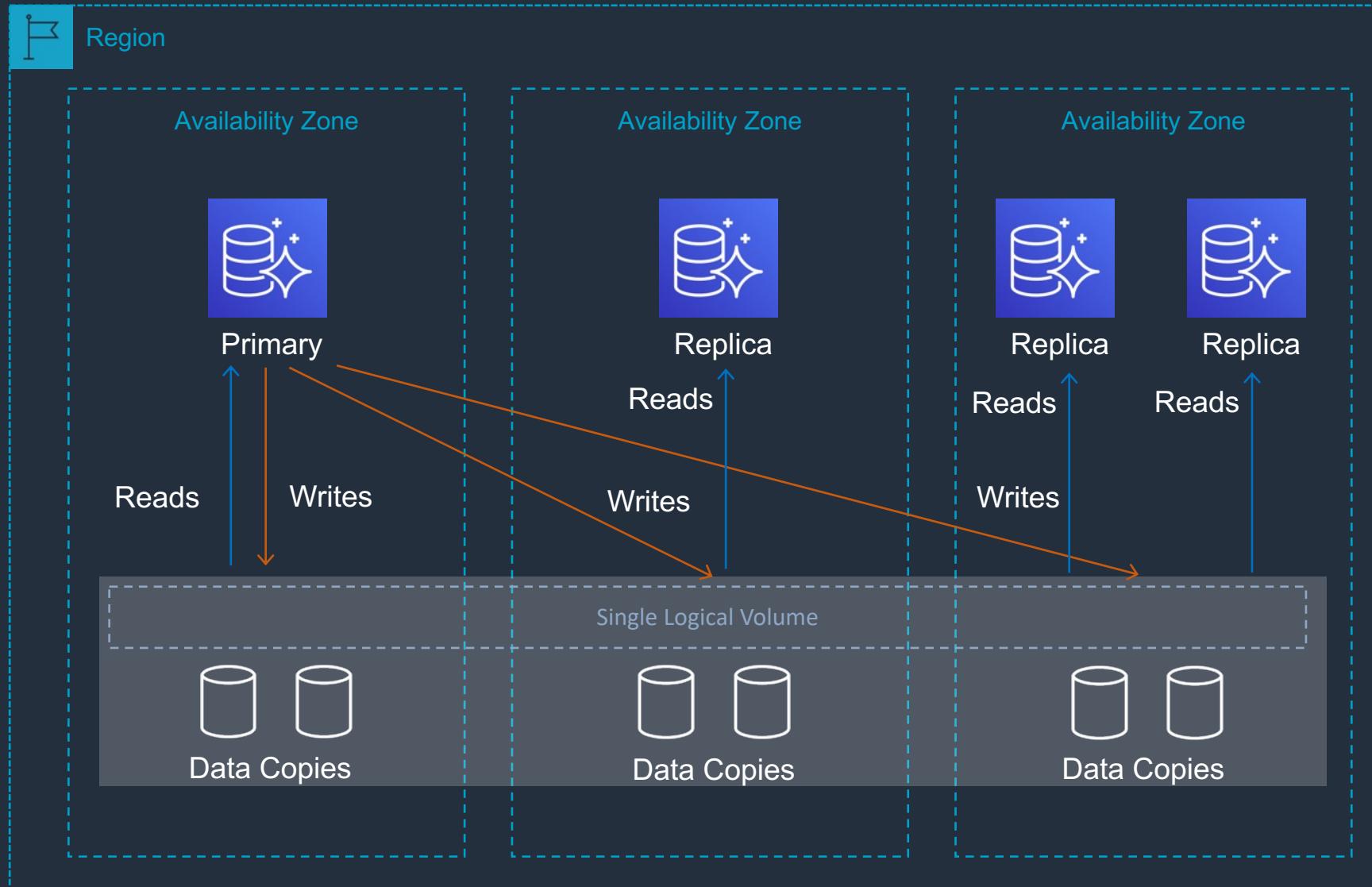
Amazon Aurora Key Features

Aurora Feature	Benefit
High performance and scalability	Offers high performance, self-healing storage that scales up to 64TB, point-in-time recovery and continuous backup to S3
DB compatibility	Compatible with existing MySQL and PostgreSQL open source databases
Aurora Replicas	In-region read scaling and failover target – up to 15 (can use Auto Scaling)
MySQL Read Replicas	Cross-region cluster with read scaling and failover target – up to 5 (each can have up to 15 Aurora Replicas)
Global Database	Cross-region cluster with read scaling (fast replication / low latency reads). Can remove secondary and promote
Multi-Master	Scales out writes within a region. In preview currently and will not appear on the exam
Serverless	On-demand, autoscaling configuration for Amazon Aurora - does not support read replicas or public IPs (can only access through VPC or Direct Connect - not VPN)

Amazon RDS Aurora Replicas

Feature	Aurora Replica	MySQL Replica
Number of replicas	Up to 15	Up to 5
Replication type	Asynchronous (milliseconds)	Asynchronous (seconds)
Performance impact on primary	Low	High
Replica location	In-region	Cross-region
Act as failover target	Yes (no data loss)	Yes (potentially minutes of data loss)
Automated failover	Yes	No
Support for user-defined replication delay	No	Yes
Support for different data or schema vs. primary	No	Yes

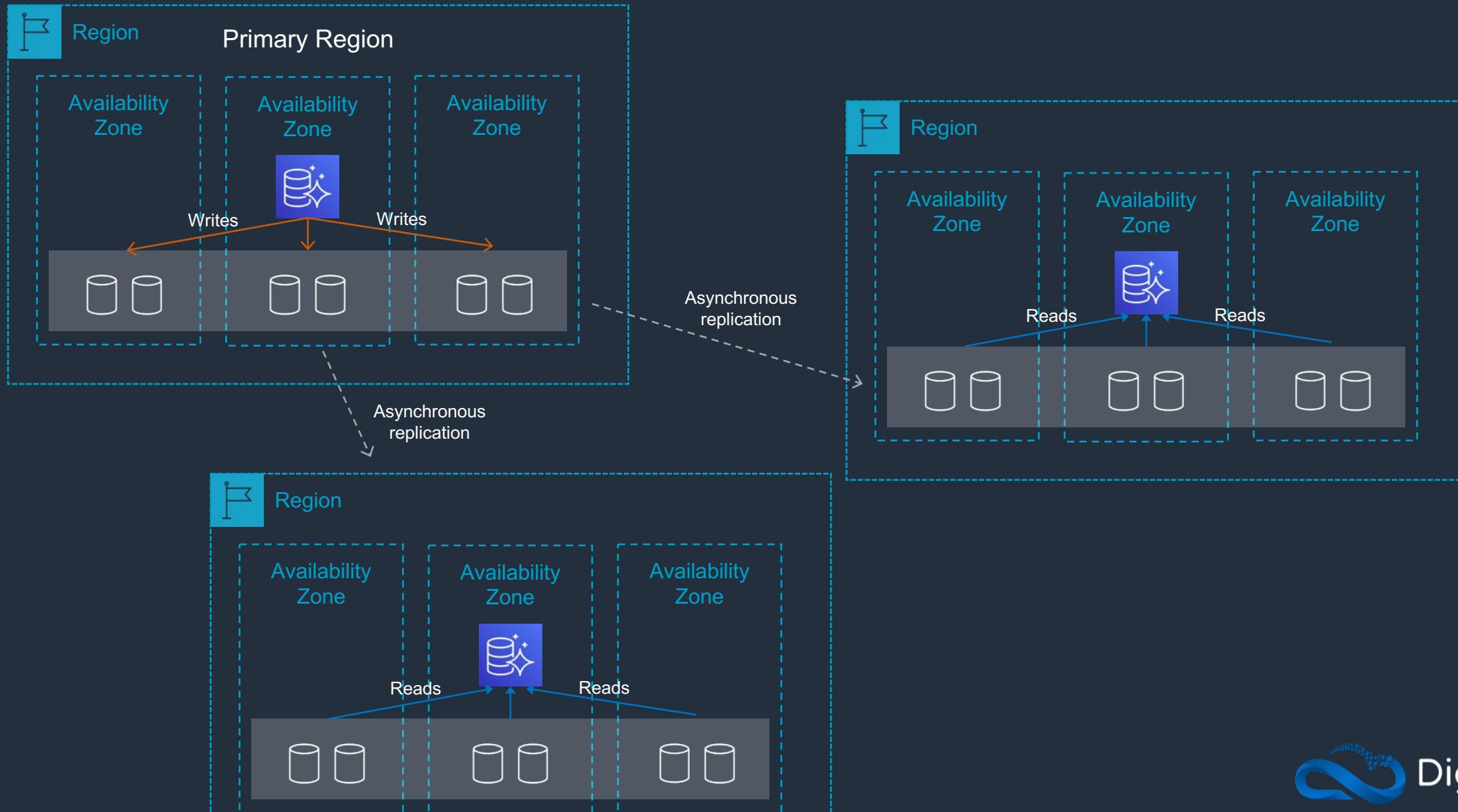
Aurora Fault Tolerance and Aurora Replicas



Aurora Fault Tolerance

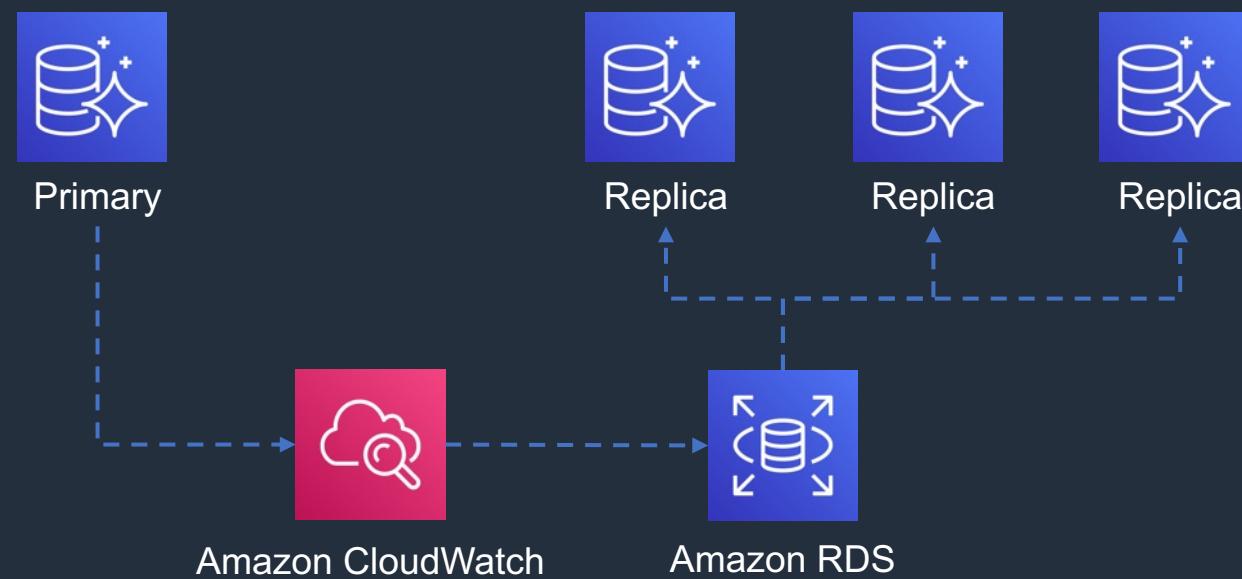
- Fault tolerance across 3 AZs
- Single logical volume
- Aurora Replicas scale-out read requests
- Up to 15 Aurora Replicas with sub-10ms replica lag
- Aurora Replicas are independent endpoints
- Can promote Aurora Replica to be a new primary or create new primary
- Set priority (tiers) on Aurora Replicas to control order of promotion
- Can use Auto Scaling to add replicas

Cross-Region Replica with Aurora MySQL



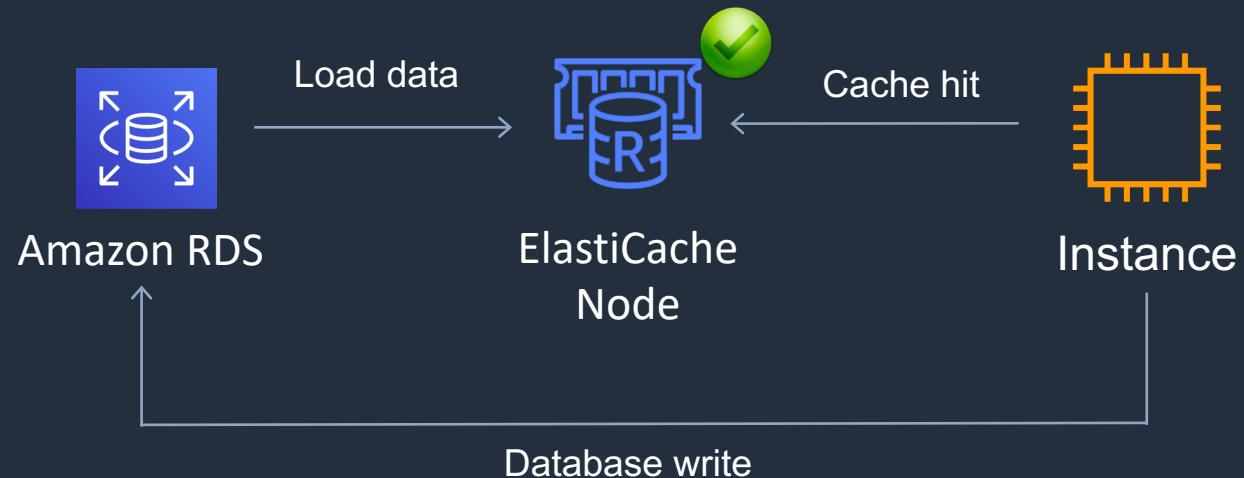
Aurora Auto Scaling

- Dynamically adjusts the number of Aurora Replicas provisioned
- Scaling policy defines min and max replicas
- Uses CloudWatch metrics to adjust number of replicas
- Application should use the Aurora reader endpoint



Amazon ElastiCache Overview

- Fully managed implementations Redis and Memcached
- ElastiCache is a key/value store
- In-memory database offering high performance and low latency
- Can be put in front of databases such as RDS and DynamoDB



Amazon ElastiCache Overview

- Good solution if your database is particularly read-heavy and the data does not change frequently
- ElastiCache can be used for storing session state
- Provides push-button scalability for memory, writes and reads
- Runs on Amazon EC2 instances
- ElastiCache EC2 nodes cannot be accessed from the Internet, nor can they be accessed by EC2 instances in other VPCs

Amazon ElastiCache Overview

Feature	Memcached	Redis (cluster mode disabled)	Redis (cluster mode enabled)
Data persistence	No	Yes	Yes
Data types	Simple	Complex	Complex
Data partitioning	Yes	No	Yes
Encryption	No	Yes	Yes
High availability (replication)	No	Yes	Yes
Multi-AZ	Yes, place nodes in multiple AZs. No failover or replication	Yes, with auto-failover. Uses read replicas (0-5 per shard)	Yes, with auto-failover. Uses read replicas (0-5 per shard)
Scaling	Up (node type); out (add nodes)	Up (node type); out (add replica)	Up (node type); out (add shards)
Multithreaded	Yes	No	No
Backup and restore	No (and no snapshots)	Yes, automatic and manual snapshots	Yes, automatic and manual snapshots

Amazon ElastiCache - Scalability

Scaling options are dependent on the database engine:

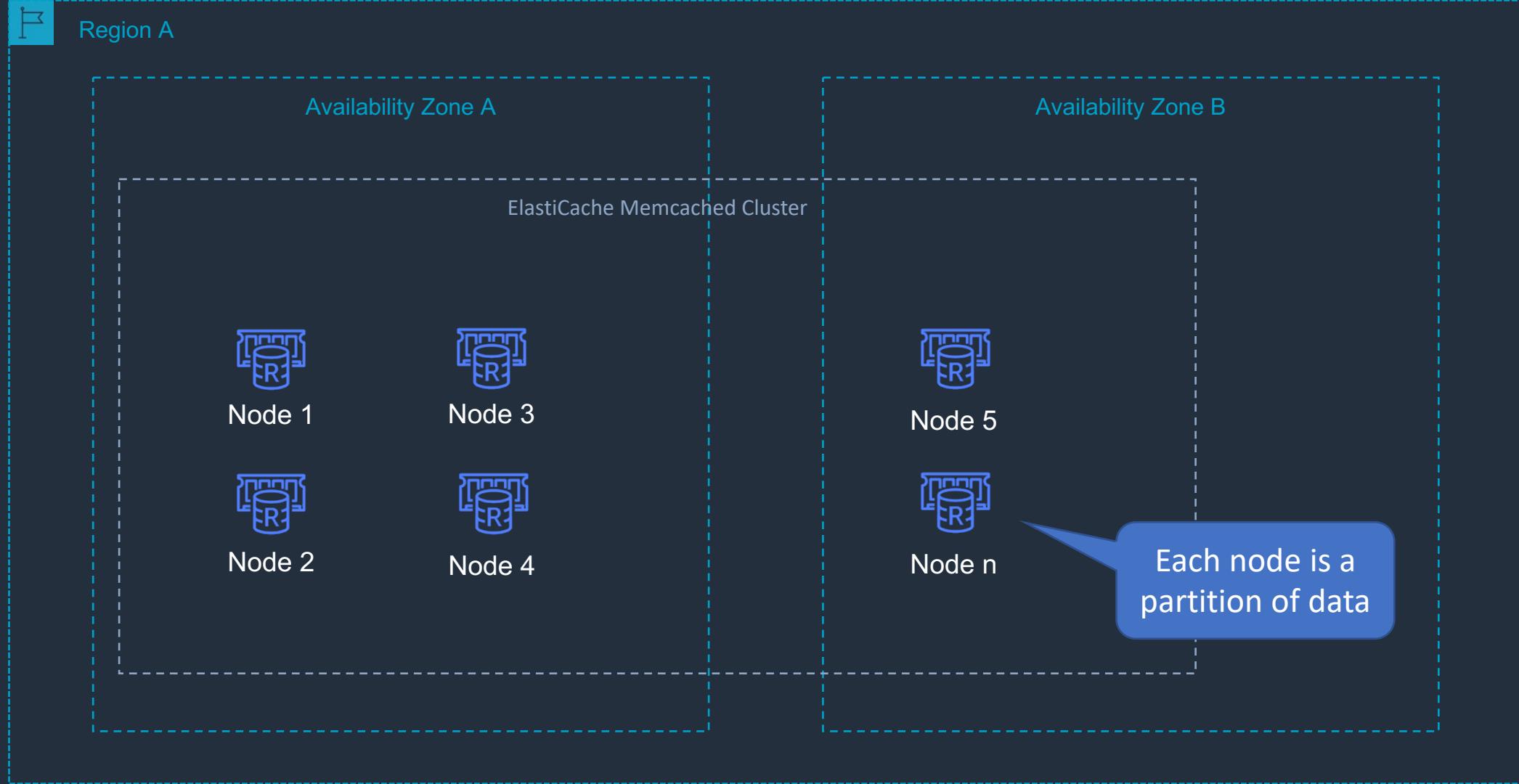
Memcached

- Add nodes to a cluster
- Scale vertically (node type) – must create a new cluster manually

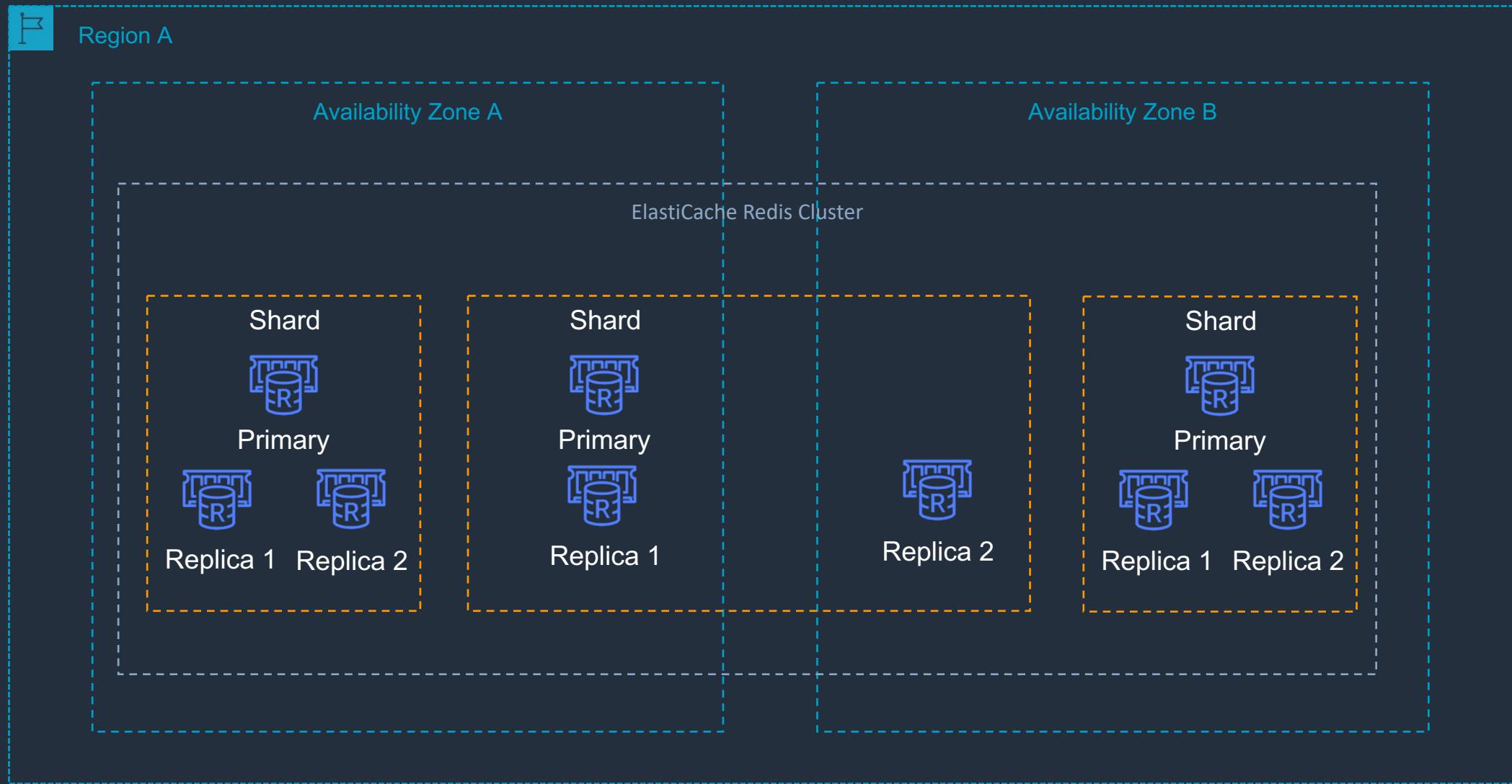
Redis

- Cluster mode disabled:
 - Add replica or change node type – creates a new cluster and migrates data
- Cluster mode enabled:
 - Online resharding to add or remove shards; vertical scaling to change node type
 - Offline resharding to add or remove shards change node type or upgrade engine (more flexible than online)

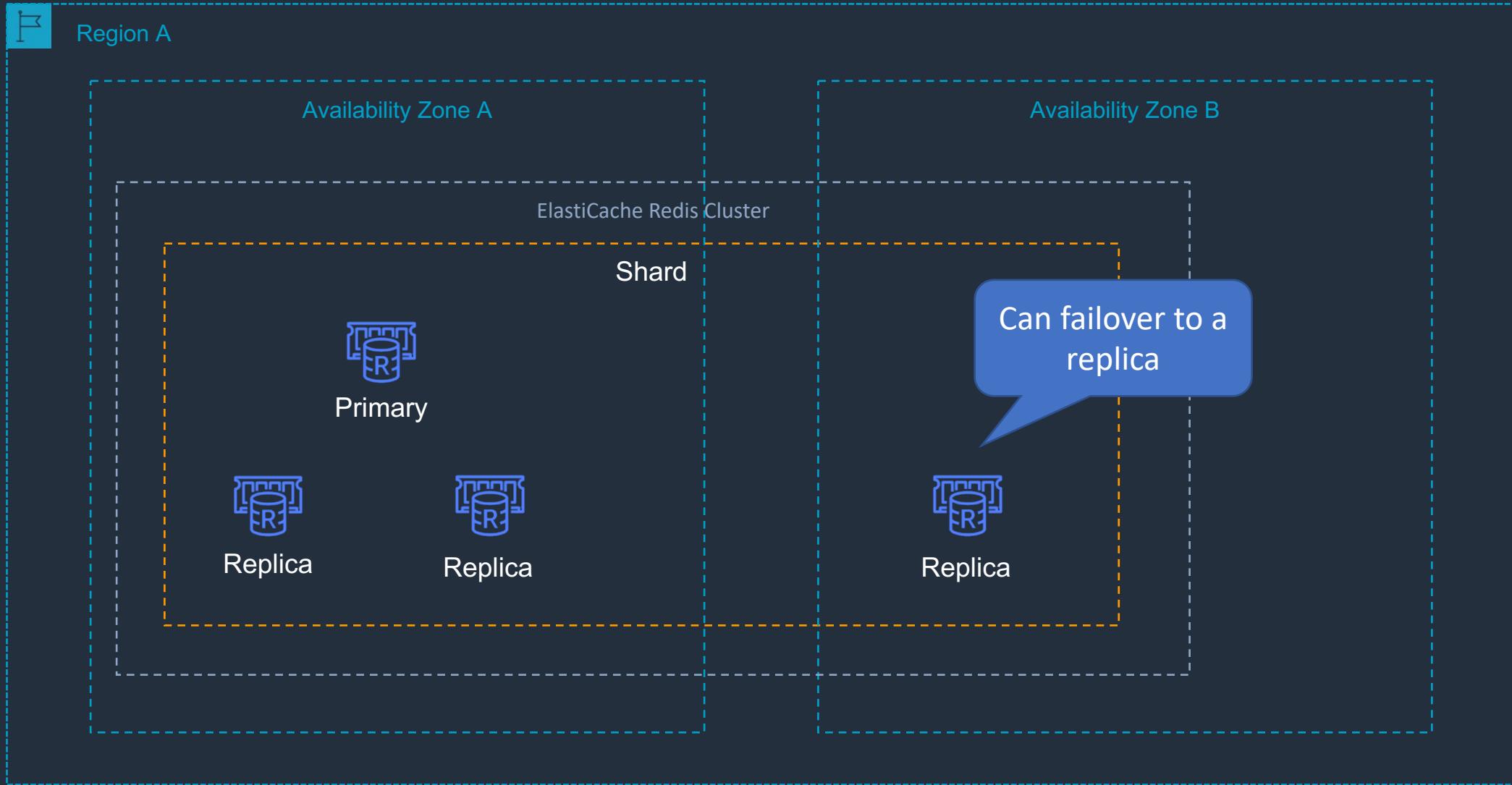
Amazon ElastiCache Memcached



Amazon ElastiCache Redis (Cluster mode enabled)



Amazon ElastiCache Redis (Cluster mode disabled)



Amazon ElastiCache – Backup

- Backup and restore is only supported for Redis
- ElastiCache does not offer backup/restore for Memcached
- Redis cluster mode enabled - supports cluster level backup only (not shard level)
- Redis cluster mode disabled – backup/restore not supported for cache.t1.micro nodes

Amazon ElastiCache – Monitoring

Useful metrics (Redis, slightly different for Memcached):

- CacheHits – The number of successful read-only key lookups
- CacheMisses – The number of unsuccessful read-only key lookups
- CacheHitRate - Indicates the usage efficiency of the Redis instance. If the cache ratio is lower than ~0.8, it means that a significant amount of keys are evicted, expired or do not exist
- DatabaseMemoryUsagePercentage – Percentage of the memory available for the cluster that is in use
- EngineCPUUtilization – Provides CPU utilization of the Redis engine thread
- Evictions – The number of keys that have been evicted due to the maxmemory limit

Amazon ElastiCache Metrics

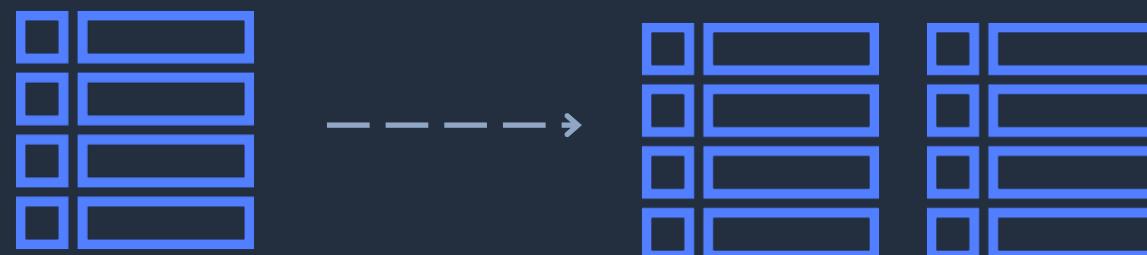
AWS recommend you monitor and set alarms for the following metrics

- **CPUUtilization** – Host-level metric reported as a percentage
- **SwapUsage** – Host-level metric reported in bytes
- **Evictions** – Cache engine metric; higher evictions indicates scaling may be required
- **CurrConnections** – Cache-engine metric; increasing connections could indicate application issues



Amazon DynamoDB

- Fully managed NoSQL database service
- Key/value store and document store
- It is a non-relational database
- Fully serverless service
- Push button scaling (horizontal)



DynamoDB Table



Amazon DynamoDB

- DynamoDB is made up of:

- Tables
- Items
- Attributes

userid	orderid	book	price	date
user001	1000092	ISBN100..	9.99	2020.04..
user002	1000102	ISBN100..	24.99	2020.03..
user003	1000168	ISBN2X0..	12.50	2020.04..



Amazon DynamoDB Key Features

DynamoDB Feature	Benefit
Serverless	Fully managed, fault tolerant, service
Highly available	99.99% availability SLA – 99.999% for Global Tables!
NoSQL type of database with Name / Value structure	Flexible schema, good for when data is not well structured or unpredictable
Horizontal scaling	Seamless scalability to any scale with push button scaling or Auto Scaling
DynamoDB Accelerator (DAX)	Fully managed in-memory cache for DynamoDB that increases performance (microsecond latency)
Backup	Point-in-time recovery down to the second in last 35 days; On-demand backup and restore
Global Tables	Fully managed multi-region, multi-master solution
DynamoDB Streams	Captures a time-ordered sequence of item-level modifications in any DynamoDB table and stores this information in a log for up to 24 hours

Amazon Elasticsearch

- Use Elasticsearch to search, analyze and visualize log data
- Fully managed service
- Supports queries using SQL syntax
- Integrates with open-source tools
- Built-in Kibana integration
- Up to 3 PB of data per cluster
- Scale by adding or removing instances
- Availability in up to three Availability Zones
- Backup using snapshots
- Encryption at-rest and in-transit

Amazon Elasticsearch

- Access controlled through:
 - **Resource-based policies** – often called a domain access policy
 - **Identity-based policies** – attached to users or roles (principals)
 - **IP-based policies** - Restrict access to one or more IP addresses or CIDR blocks; basically a condition in a resource-based policy

Exam Scenarios

Exam Scenario	Solution
Automated failover of a multi-AZ DB occurred	This may be due to storage failure on primary DB or the instance type could have been changed
Need to encrypt unencrypted RDS database	Take a snapshot, encrypt it, then restore a new encrypted instance from the snapshot
RDS DB query latency is high and CPU utilization is at 100%	Scale up with larger instance type
Need to share RDS DB snapshots across different accounts. Data must be encrypted	Use an AWS KMS key for encryption and update key policy to grant accounts with access then share snapshot

Exam Scenarios

Exam Scenario	Solution
DB needs to be made HA to protect against failure and updates cannot impact users in business hours	Change to Multi-AZ outside of business hours
Need to protect RDS databases against table corruption within a 30 day window of protection	Enable automated backups and set the appropriate retention period
Shared Responsibility Model	AWS is responsible for maintenance, patches and other updates for Aurora DB
AuroraReplicaLagMaximum is high for DB on eCommerce site. What affect could this have?	may result in cart not updating correctly (inconsistency)

Exam Scenarios

Exam Scenario	Solution
EC2 connects to RDS instance and fails with: "Error Establishing a Database Connection"	Web server may be using certificate validation and RDS does not trust the certificate. Or, the DB security group does not have the correct ingress rule
Aurora DB is hitting 100% CPU. Read-heavy app with many lookups	Add Aurora Replicas and use a Reader Endpoint for product table lookups
Database is running MySQL on Amazon EC2. Need to increase availability and durability without changing application	Use Aurora MySQL and configure an Aurora Replica in another AZ
Reporting job runs against RDS instance and is causing performance issues	Create a read replica and point the reporting job to the read replica endpoint

Exam Scenarios

Exam Scenario	Solution
Backup of RDS instance must be copied regularly to another account for testing	Create a snapshot with <code>create-db-snapshot</code> CLI, share with other account, then create a copy in that account
MySQL database on RDS must be patched due to a security vulnerability. Who is responsible?	AWS is responsible for patching Amazon RDS database instances
Reporting job runs against RDS instance and is causing performance issues	Create a read replica and point the reporting job to the read replica endpoint

Exam Scenarios

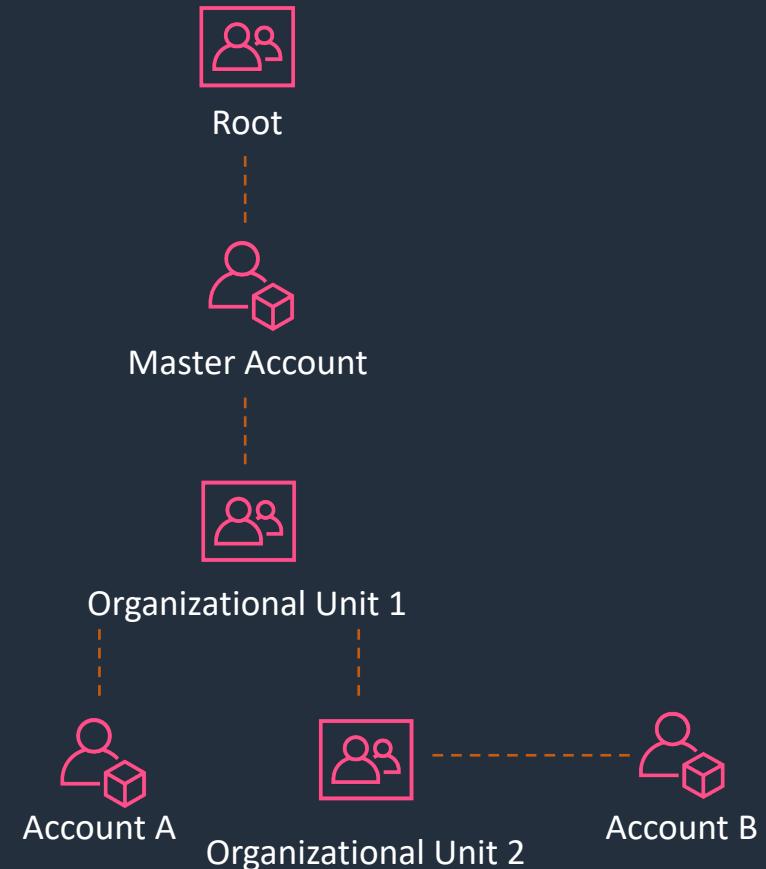
Exam Scenario	Solution
How can a Redis cluster be scaled to improve read times	Scale horizontally by adding shards
High CPU on a Memcached cluster	Options are to add additional nodes to cluster or vertically scale the node types
ElastiCache Memcached storing session state. Performance poor, eviction count metrics are high	Scale the cluster by adding additional nodes
A Memcached cluster is experiencing increased traffic, need to change to larger node type	Create a new cache cluster with the new node type using the CreateCacheCluster API

SECTION 13

Management, Governance and Billing

AWS Organizations

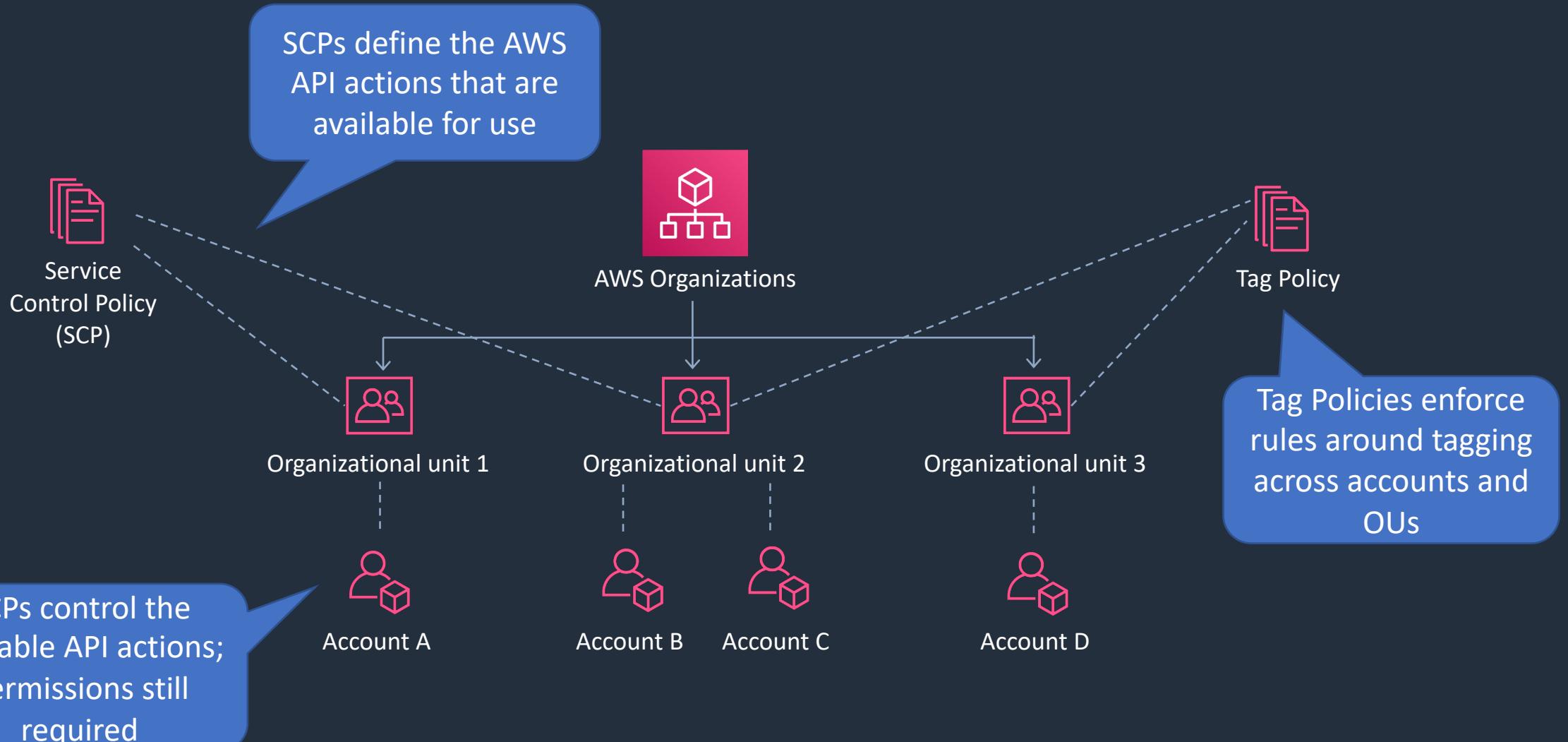
- AWS organizations allows you to consolidate multiple AWS accounts into an organization that you create and centrally manage
- Available in two feature sets:
 - Consolidated Billing
 - All features
- Consolidated billing includes:
 - Paying Account – independent and cannot access resources of other accounts; you get one bill for multiple accounts
 - Linked Accounts – all linked accounts are independent



AWS Organizations API

- The AWS Organizations API can be used to automate organization and account creation
- The following API actions are useful to know:
 - **CreateOrganization** – creates an organization
 - **CreateAccount** – creates an account that is a member of the organization
 - **CreatePolicy** – creates a policy that can be attached to a root, OU, or individual AWS account
 - **AttachPolicy** - Attaches a policy to a root, an organizational unit (OU), or an individual account
 - **InviteAccountToOrganization** - Sends an invitation to another account to join your organization as a member account

AWS Organizations – SCPs and Tag Policies



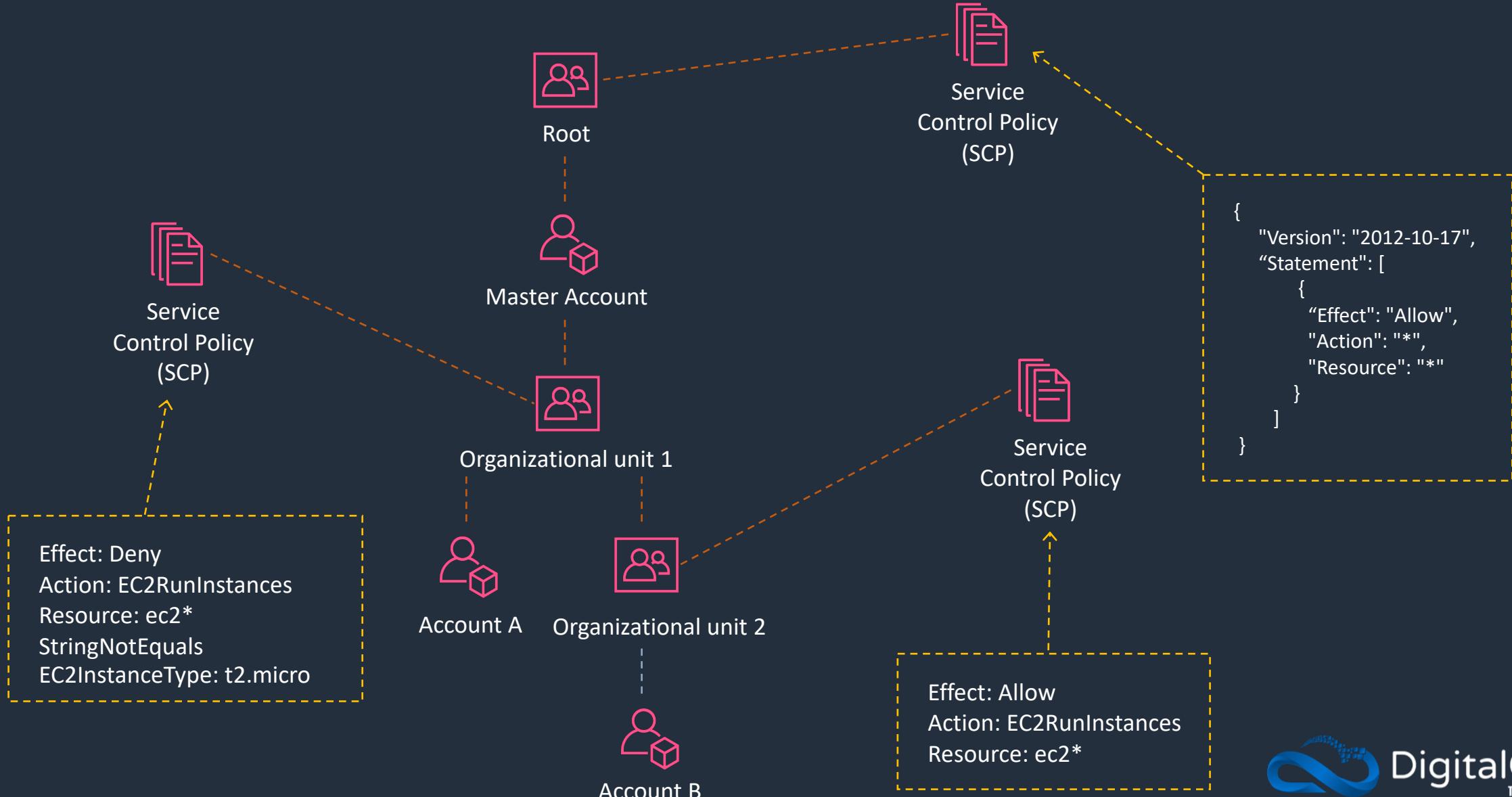
AWS Organizations – Service Control Policies

- Service Control Policies (SCPs) control the available permissions in accounts within an organization
- Specifically, they control the API actions that are available for use
- Must have all features enabled in the AWS Organization
- Examples of what you can control are:
 - Limit the **ec2:RunInstances** API action to allow launching **t2.micro** instances only
 - Deny the **s3:DeleteBucket** API action to prevent deletion of buckets
 - Users and roles must still be granted permissions with appropriate IAM permission policies

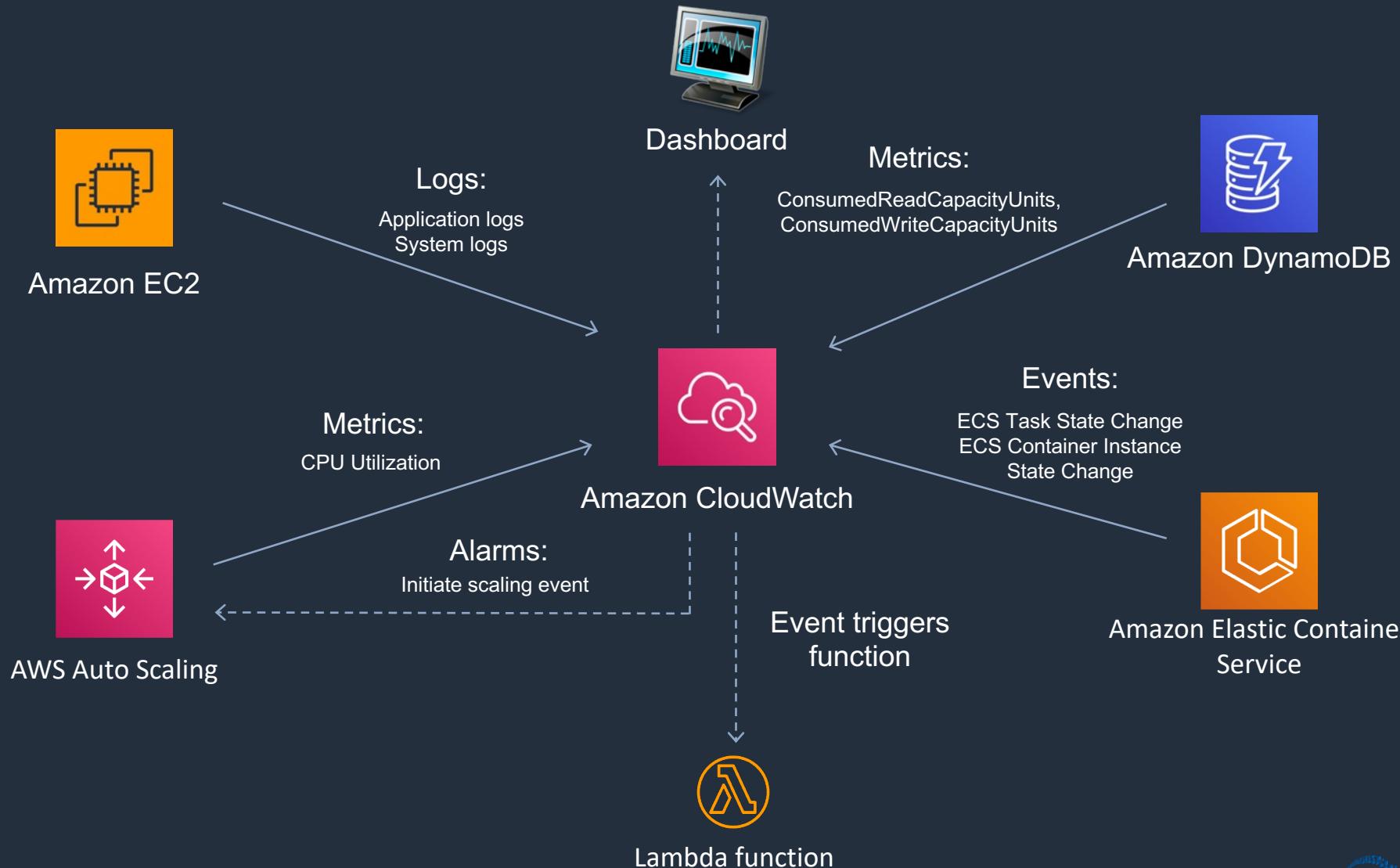
AWS Organizations – SCP Effects

- SCPs affect only principals that are managed by accounts that are part of the organization
- An SCP restricts permissions for principals in member accounts, including each AWS account root user (except in the master account)
- Users and roles must still be granted permissions with appropriate IAM permission policies
- Users / roles must have permissions through IAM and be allowed (or not denied) through an SCP to perform an action
- SCPs do not affect any service-linked roles

AWS Organizations – SCP Effects on Permissions

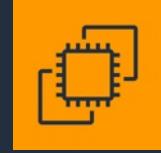


Amazon CloudWatch – Examples of Functionality



Amazon CloudWatch Overview

- Amazon CloudWatch monitors AWS resources and applications in real-time
- CloudWatch collects and tracks metrics
- Metrics are data points that are published to CloudWatch
- CloudWatch alarms monitor metrics and automatically initiate actions
- CloudWatch Logs centralizes logs from systems, applications and AWS services
- CloudWatch Events delivers a stream of system events that describe changes in AWS resources



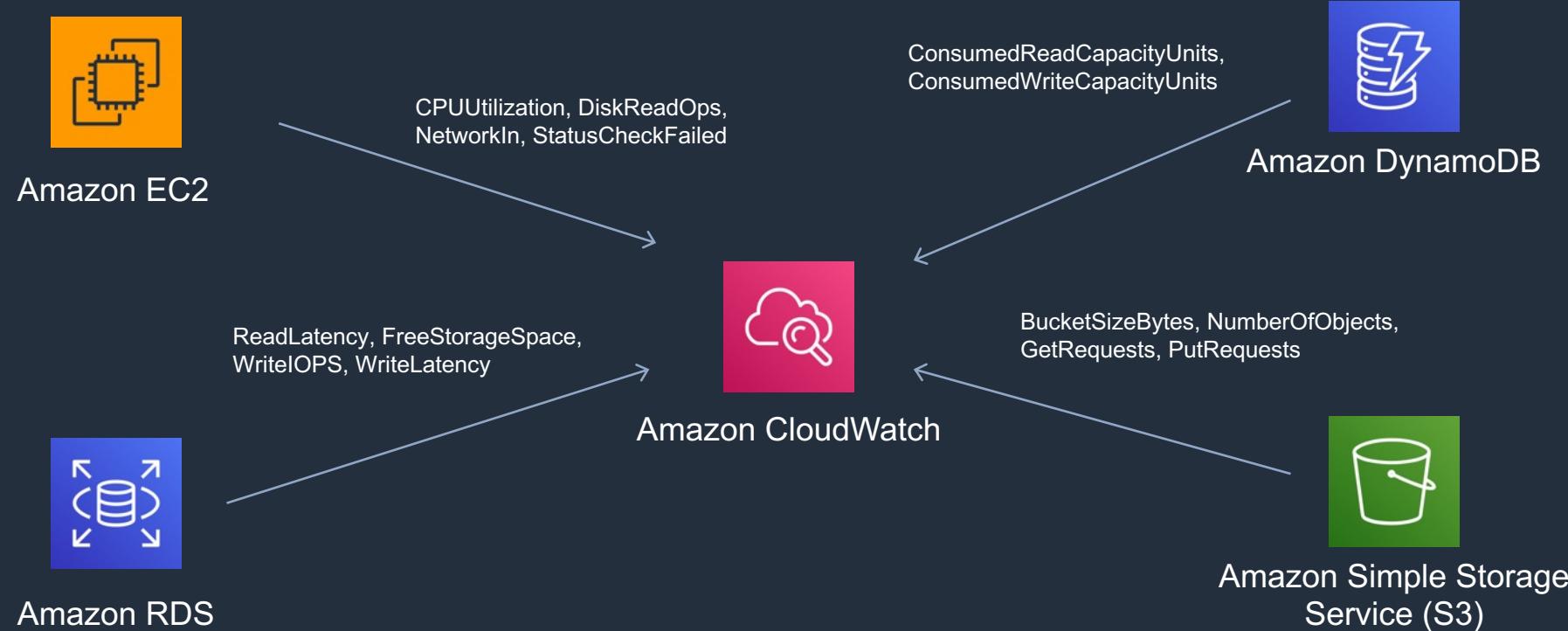
Amazon EC2

Metrics



Amazon CloudWatch

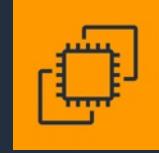
Amazon CloudWatch - Metrics



Amazon CloudWatch – Key Terminology and Concepts

Metrics:

- Metrics are the fundamental concept in CloudWatch
- A metric represents a time-ordered set of data points that are published to CloudWatch
- AWS services send metrics to CloudWatch
- You can also send your own custom metrics to CloudWatch
- Metrics exist within a region
- Metrics cannot be deleted but automatically expire after 15 months
- Metrics are uniquely defined by a name, a namespace, and zero or more dimensions
- Time stamps can be up to two weeks in the past and up to two hours into the future



Amazon EC2



Amazon CloudWatch

Amazon CloudWatch Metrics – Useful API Actions

- **GetMetricData**
 - Retrieve as many as 500 different metrics in a single request
- **PutMetricData**
 - Publishes metric data points to Amazon CloudWatch
 - CloudWatch associates the data points with the specified metric
 - If the specified metric does not exist, CloudWatch creates the metric
- **GetMetricStatistics**
 - Gets statistics for the specified metric
 - CloudWatch aggregates data points based on the length of the period that you specify
 - Maximum number of data points returned from a single call is 1,440

Amazon CloudWatch – Key Terminology and Concepts

Namespace:

- A namespace is a container for CloudWatch metrics
- Metrics in different namespaces are isolated from each other, so that metrics from different applications are not mistakenly aggregated into the same statistics

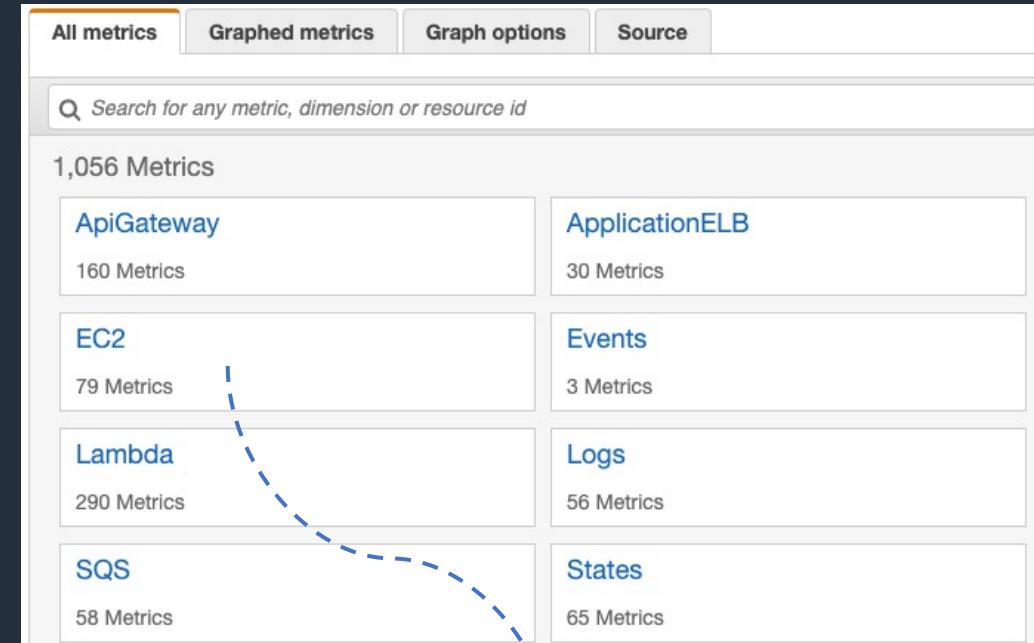
Service	Namespace
Amazon API Gateway	AWS/ApiGateway
Amazon CloudFront	AWS/CloudFront
AWS CloudHSM	AWS/CloudHSM
Amazon CloudWatch Logs	AWS/Logs
AWS CodeBuild	AWS/CodeBuild
Amazon Cognito	AWS/Cognito
Amazon DynamoDB	AWS/DynamoDB
Amazon EC2	AWS/EC2
AWS Elastic Beanstalk	AWS/ElasticBeanstalk

Amazon CloudWatch – Key Terminology and Concepts

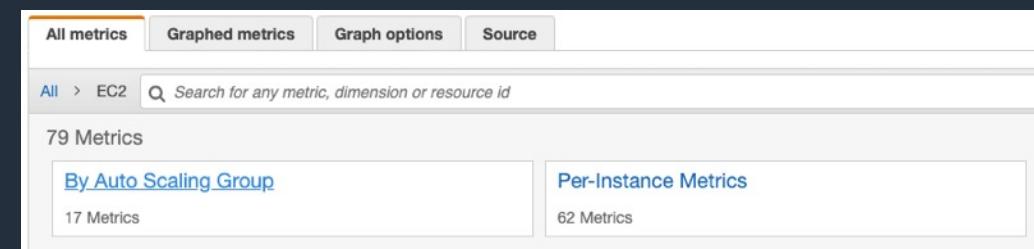
Dimensions:

- A dimension is a name/value pair that is part of the identity of a metric
- You can assign up to 10 dimensions to a metric
- Dimensions are categories for the characteristics of each metric

These are namespaces



These are dimensions



Amazon CloudWatch – Key Terminology and Concepts

Statistics:

- Statistics are metric data aggregations over specified periods of time
- CloudWatch provides statistics based on the metric data points provided by your custom data or provided by other AWS services to CloudWatch

Statistic	Description
Minimum	The lowest value observed during the specified period. You can use this value to determine low volumes of activity for your application.
Maximum	The highest value observed during the specified period. You can use this value to determine high volumes of activity for your application.
Sum	All values submitted for the matching metric added together. This statistic can be useful for determining the total volume of a metric.
Average	The value of Sum / SampleCount during the specified period. By comparing this statistic with the Minimum and Maximum, you can determine the full scope of a metric and how close the average use is to the Minimum and Maximum. This comparison helps you to know when to increase or decrease your resources as needed.
SampleCount	The count (number) of data points used for the statistical calculation.
pNN.NN	The value of the specified percentile. You can specify any percentile, using up to two decimal places (for example, p95.45). Percentile statistics are not available for metrics that include any negative values. For more information, see Percentiles .

Amazon CloudWatch – Key Terminology and Concepts

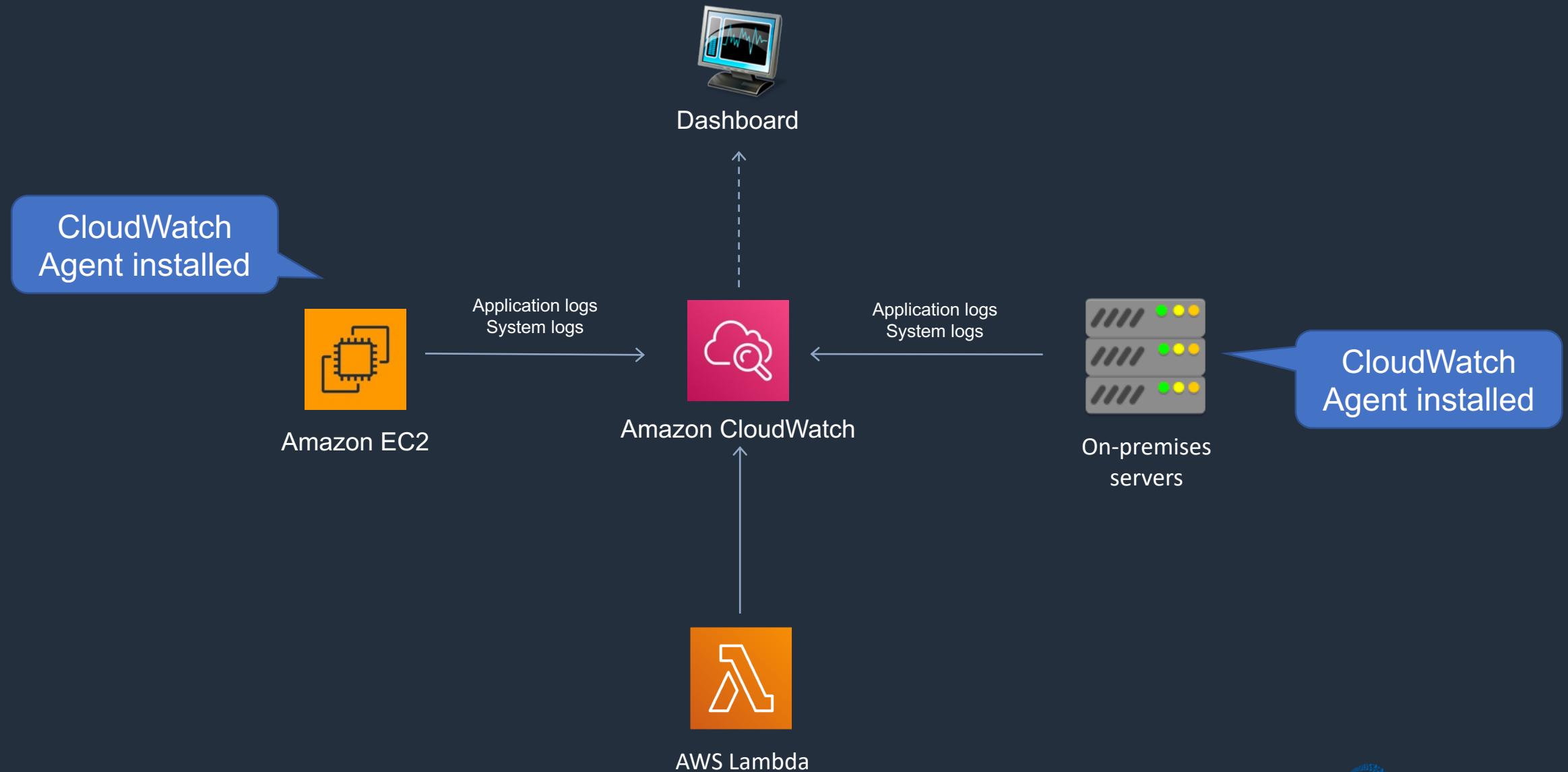
Alarms:

- You can use an alarm to automatically initiate actions on your behalf
- An alarm watches a single metric over a specified time period, and performs one or more specified actions, based on the value of the metric relative to a threshold over time
- The action is a notification sent to an Amazon SNS topic or an Auto Scaling policy
- Alarms invoke actions for sustained state changes only
- CloudWatch alarms do not invoke actions simply because they are in a particular state
- The state must have changed and be maintained for a specified period

Amazon CloudWatch Alarms – Useful API Actions

- **PutMetricAlarm**
 - Creates or updates an alarm and associates it with the specified metric, metric math expression, or anomaly detection model
 - Alarms based on anomaly detection models cannot have Auto Scaling actions
- **SetAlarmState**
 - Temporarily sets the state of an alarm for testing purposes

Amazon CloudWatch Logs



Amazon CloudWatch Logs

- CloudWatch Logs enables you to centralize the logs from all of your systems, applications, and AWS services.
- Features:
 - Monitor logs from Amazon EC2 instances - monitors application and system logs and can trigger notifications
 - Monitor CloudTrail Logged Events – alarms can be created in CloudWatch based on API activity captured by CloudTrail
 - Log retention – by default, logs are retained indefinitely. Configurable per log group from 1 day to 10 years

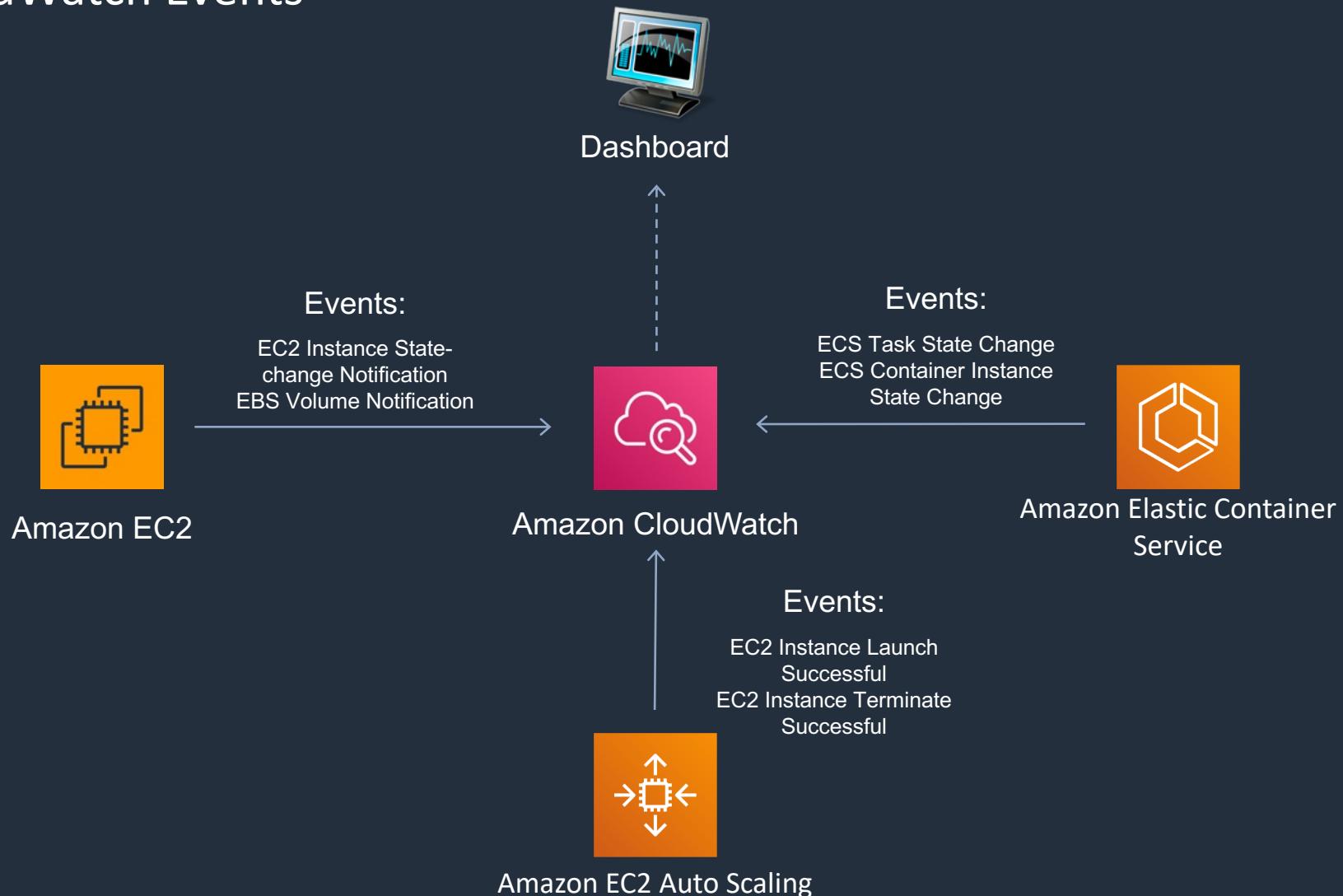
Amazon CloudWatch Logs Agent

- The CloudWatch Logs agent provides an automated way to send log data to CloudWatch Logs from Amazon EC2 instances
- There is now a unified CloudWatch agent that collects both logs and metrics
- The unified CloudWatch agent includes metrics such as memory and disk utilization

Amazon CloudWatch Agent

- The unified CloudWatch agent enables you to do the following:
 - Collect more system-level metrics from Amazon EC2 instances across operating systems. The metrics can include in-guest metrics, in addition to the metrics for EC2 instances
 - Collect system-level metrics from on-premises servers. These can include servers in a hybrid environment as well as servers not managed by AWS
 - Retrieve custom metrics from your applications or services using the StatsD and collectd protocols

Amazon CloudWatch Events



Amazon CloudWatch Events

- Amazon CloudWatch Events delivers a near real-time stream of system events that describe changes in AWS resources
- Can use CloudWatch Events to schedule automated actions that self-trigger at certain times using cron or rate expressions
- Can match events and route them to one or more target functions or streams

Amazon CloudWatch Events

- Targets include:
 - Amazon EC2 instances
 - AWS Lambda functions
 - Streams in Amazon Kinesis Data Streams
 - Delivery streams in Amazon Kinesis Data Firehose
 - Log groups in Amazon CloudWatch Logs
 - Amazon ECS tasks
 - Systems Manager Run Command
 - Systems Manager Automation
 - AWS Batch jobs
 - Step Functions state machines
 - Pipelines in CodePipeline
 - CodeBuild projects
 - Amazon Inspector assessment templates
 - Amazon SNS topics
 - Amazon SQS queues

Amazon CloudWatch Events

Specify event source:

Event Pattern i Schedule i

Build event pattern to match events by service

Service Name: EC2

Event Type: EC2 Instance State-change Notification

Any state Specific state(s)

Any instance Specific instance Id(s)

Event Pattern Preview

```
{  
  "source": [  
    "aws.ec2"  
  ],  
  "detail-type": [  
    "EC2 Instance State-change Notification"  
  ]  
}
```

[Copy to clipboard](#) [Edit](#)

Specify event target:

Targets

Select Target to invoke when an event matches your Event Pattern or when schedule is triggered.

Lambda function

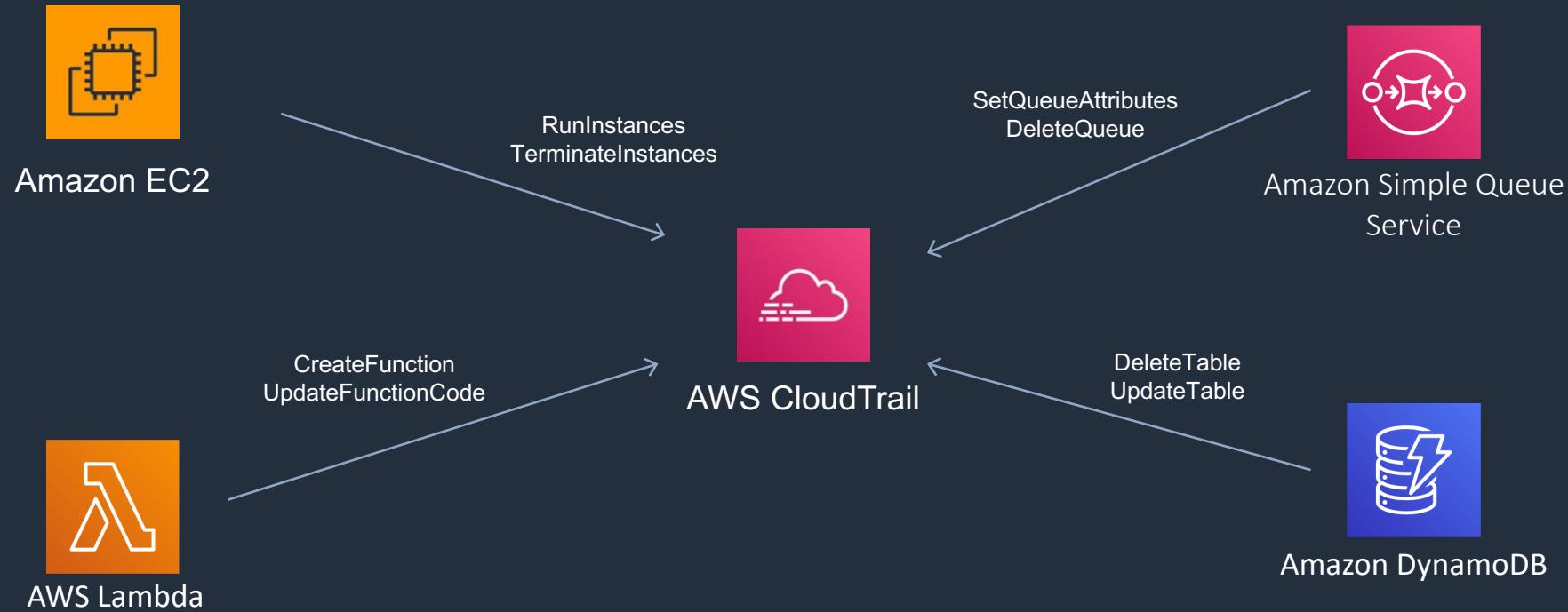
Function* [Select function](#)

▶ Configure version/alias
▶ Configure input

Amazon CloudWatch Events Example



Auditing with AWS CloudTrail



Amazon CloudTrail

- AWS CloudTrail is a web service that records API activity made on AWS accounts
- A CloudTrail trail can be created which delivers log files to an Amazon S3 bucket
- Enables governance, compliance, and operational and risk auditing of your AWS account
- Events include actions taken in the AWS Management Console, AWS Command Line Interface, and AWS SDKs and APIs
- CloudTrail is enabled on your AWS account when you create it
- Can use Athena to query logs

Amazon CloudTrail

You can create two types of trails for an AWS account:

- A trail that applies to all regions - records events in all regions and delivers to an S3 bucket
- A trail that applies to a single region – records events in a single region and delivers to an S3 bucket. Additional single trails can use the same or different bucket

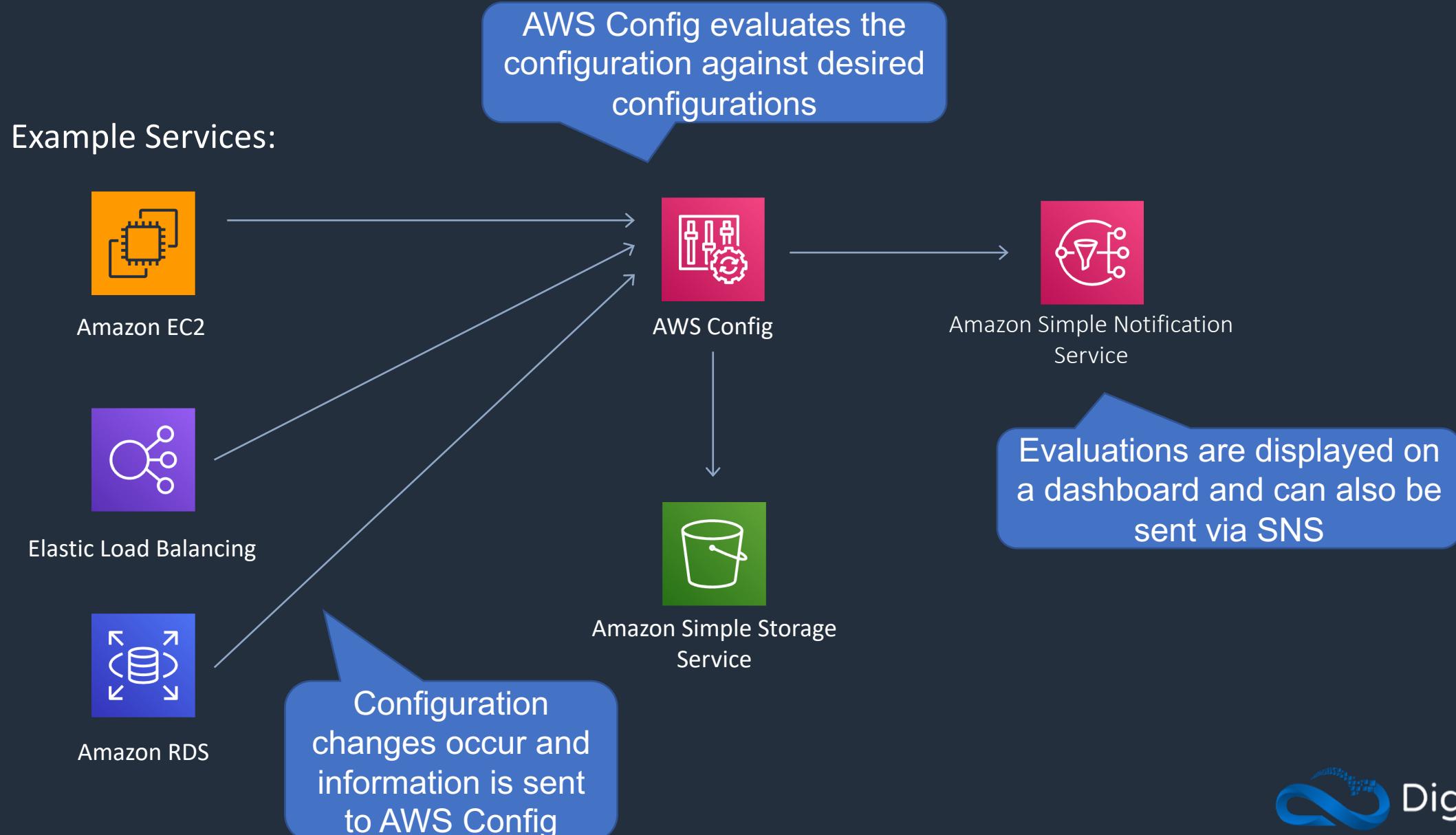
Amazon CloudTrail – Management Events

- Management events provide information about management operations that are performed on resources in your AWS account. These are also known as control plane operations
- Example management events include:
 - Configuring security (for example IAM `AttachRolePolicy` API operations)
 - Registering devices (for example, `CreateDefaultVpc` API operations)
 - Configuring rules for routing data (for example `CreateSubnet` API operations)
 - Setting up logging (for example, AWS CloudTrail `CreateTrail` API operations)

Amazon CloudTrail – Data Events

- Data events provide information about the resource operations performed on or in a resource
- These are also known as data plane operations
- Data events are often high-volume activities.
- Example data events include:
 - Amazon S3 object-level API activity (for example, `GetObject`, `DeleteObject`, and `PutObject` API operations)
 - AWS Lambda function execution activity (the `Invoke` API)

AWS Config



AWS Config

- AWS Config is a fully managed service that provides you with an AWS resource inventory, configuration history, and configuration change notifications to enable security and governance
- You can discover existing AWS resources, export a complete inventory of your AWS resources with all configuration details, and determine how a resource was configured at any point in time
- These capabilities enable compliance auditing, security analysis, resource change tracking, and troubleshooting
- Allow you to assess, audit and evaluate configurations of your AWS resources
- Very useful for Configuration Management as part of an ITIL program

AWS Service Catalog

- AWS Service Catalog allows organizations to create and manage catalogs of IT services that are approved for use on AWS
- These IT services can include everything from virtual machine images, servers, software, and databases to complete multi-tier application architectures
- AWS Service Catalog allows you to centrally manage commonly deployed IT services
- Helps to achieve consistent governance and meet compliance requirements
- Enables users to quickly deploy only the approved IT services they need

AWS Service Catalog- Sharing Portfolios

- You can share portfolios across accounts in AWS Organizations
- Either share a reference to the catalog or deploy a copy of the catalog
- With copies you must redeploy any updates
- CloudFormation StackSets can be used to deploy a catalog to multiple accounts at the same time
- Check reference link for more information on the behavior

AWS Trusted Advisor

- Trusted Advisor is an online resource that helps to reduce cost, increase performance and improve security by optimizing your AWS environment
- Trusted Advisor provides real time guidance to help you provision your resources following best practices
- Advisor will advise you on Cost Optimization, Performance, Security, and Fault Tolerance

AWS Personal Health Dashboard

- AWS Personal Health Dashboard provides alerts and remediation guidance when AWS is experiencing events that may impact you
- Personal Health Dashboard gives you a personalized view into the performance and availability of the AWS services underlying your AWS resources
- Provides a personalized view of AWS issues that may impact you
- The dashboard displays relevant and timely information to help you manage events in progress
- Also provides proactive notification to help you plan for scheduled activities
- Alerts are triggered by changes in the health of AWS resources, giving you event visibility, and guidance to help quickly diagnose and resolve issues

Service Health Dashboard

Not personalized information so may not be relevant to you

No proactive notification of scheduled activities

Current Status - Jun 7, 2020 PDT

Amazon Web Services publishes our most up-to-the-minute information on service availability in the table below. Check back here any time to get current status information, or subscribe to an RSS feed to be notified of interruptions to each individual service. If you are experiencing a real-time, operational issue with one of our services that is not described below, please inform us by clicking on the "Contact Us" link to submit a service issue report. All dates and times are Pacific Time (PST/PDT).

North America	South America	Europe	Africa	Asia Pacific	Middle East	Contact Us
Recent Events	Details				RSS	
No recent events.						
Remaining Services		Details			RSS	
Alexa for Business (N. Virginia)		Service is operating normally				
Amazon API Gateway (Montreal)		Service is operating normally				
Amazon API Gateway (N. California)		Service is operating normally				
Amazon API Gateway (N. Virginia)		Service is operating normally				
Amazon API Gateway (Ohio)		Service is operating normally				
Amazon API Gateway (Oregon)		Service is operating normally				
Amazon AppStream 2.0 (N. Virginia)		Service is operating normally				
Amazon AppStream 2.0 (Oregon)		Service is operating normally				
Amazon Athena (Montreal)		Service is operating normally				
Amazon Athena (N. Virginia)		Service is operating normally				
Amazon Athena (Ohio)		Service is operating normally				
Amazon Athena (Oregon)		Service is operating normally				

Shows current status information on service availability

AWS Cost Explorer

- The AWS Cost Explorer is a free tool that allows you to view charts of your costs
- You can view cost data for the past 13 months and forecast how much you are likely to spend over the next three months
- Cost Explorer can be used to discover patterns in how much you spend on AWS resources over time and to identify cost problem areas
- Cost Explorer can help you to identify service usage statistics such as:
- Which services you use the most
- View metrics for which AZ has the most traffic
- Which linked account is used the most



AWS Cost Allocation Tags

- A tag is a label that you or AWS assigns to an AWS resource
- Each tag consists of a key and a value
- You can use tags to organize your resources, and cost allocation tags to track your AWS costs on a detailed level
- Must activate the tags in the Billing and Cost Management console



AWS Cost and Usage Report

- AWS Cost and Usage reports provides a detailed data set about your AWS billing, delivered to an S3 bucket
- This is small excerpt:

F	G	H	I	J	K	L	M	N	O	P
bill/PayerAccountId	bill/BillingPeriod	bill/BillingPeriodEnd	lineitem/UsageAccountId	lineitem/LineItemUsageQuantity	lineitem/LineItemUsageType	lineitem/LineItemProvisionedCapacity	lineitem/LineItemOpType	lineitem/AvailableOpType		
5.15148E+11	2020-08-01T	2020-09-01T	3.33783E+11	Credit	2020-08-01T	2020-08-17T	AmazonRoute53 HostedZone			
5.15148E+11	2020-08-01T	2020-09-01T	3.33783E+11	Credit	2020-08-01T	2020-08-27T	AmazonS3 Requests-Tier2			
5.15148E+11	2020-08-01T	2020-09-01T	3.33783E+11	Credit	2020-08-01T	2020-08-22T	AWSDataTransfer Requests-Tier2	APS2-USE2-AWS-Out-Bytes		
5.15148E+11	2020-08-01T	2020-09-01T	3.33783E+11	Credit	2020-08-01T	2020-08-27T	AmazonS3 Requests-Tier1	APS2-Requests-Tier1		
5.15148E+11	2020-08-01T	2020-09-01T	3.33783E+11	Credit	2020-08-07T	2020-08-27T	AWSDataTransfer Requests-Tier2	APS2-USE1-AWS-Out-Bytes		
5.15148E+11	2020-08-01T	2020-09-01T	3.33783E+11	Credit	2020-08-01T	2020-08-29T	AmazonS3 Requests-Tier2	APS2-Requests-Tier2		
5.15148E+11	2020-08-01T	2020-09-01T	3.33783E+11	Credit	2020-08-01T	2020-08-31T	AmazonS3 TimedStorage-ByteHrs			
5.15148E+11	2020-08-01T	2020-09-01T	3.33783E+11	Credit	2020-08-01T	2020-09-01T	AmazonRoute53 DNS-Queries			
5.15148E+11	2020-08-01T	2020-09-01T	3.33783E+11	Credit	2020-08-01T	2020-08-17T	AmazonS3 Requests-Tier1	APS2-Requests-Tier1		
5.15148E+11	2020-08-01T	2020-09-01T	3.33783E+11	Credit	2020-08-01T	2020-08-31T	AmazonS3 TimedStorage-ByteHrs	APS2-TimedStorage-ByteHrs		
5.15148E+11	2020-08-01T	2020-09-01T	3.33783E+11	Credit	2020-08-01T	2020-08-06T	AWSDataTransfer Requests-Tier2	APS2-USE1-USE2-AWS-Out-Bytes		
5.15148E+11	2020-08-01T	2020-09-01T	8.3696E+11	Credit	2020-08-09T	2020-08-18T	AmazonS3 Requests-Tier1	APS2-Requests-Tier1		
5.15148E+11	2020-08-01T	2020-09-01T	8.3696E+11	Credit	2020-08-08T	2020-08-08T	AWSELB LoadBalancing:Application	APS2-LCUUS LoadBalancing:Application		
5.15148E+11	2020-08-01T	2020-09-01T	8.3696E+11	Credit	2020-08-09T	2020-08-18T	AmazonS3 Requests-Tier2	APS2-Requests-Tier2		
5.15148E+11	2020-08-01T	2020-09-01T	8.3696E+11	Credit	2020-08-08T	2020-08-10T	AmazonEC2 BoxUsage:RunInstances	APS2-BoxUsage:RunInstances		
5.15148E+11	2020-08-01T	2020-09-01T	8.3696E+11	Credit	2020-08-08T	2020-08-08T	AWSELB LoadBalancing:Application	APS2-LoadBalancing:Application		
5.15148E+11	2020-08-01T	2020-09-01T	8.3696E+11	Credit	2020-08-01T	2020-08-31T	AmazonS3 TimedStorage-ByteHrs	APS2-TimedStorage-ByteHrs		
5.15148E+11	2020-08-01T	2020-09-01T	8.3696E+11	Credit	2020-08-09T	2020-08-26T	AmazonAPIGateway Requests-Tier2	APS2-ApiGatewayRequest		
5.15148E+11	2020-08-01T	2020-09-01T	8.3696E+11	Credit	2020-08-08T	2020-08-10T	AmazonEC2 EBS:VolumeUsage	APS2-EBS:VolumeUsage		
5.15148E+11	2020-08-01T	2020-09-01T	8.3696E+11	Credit	2020-08-07T	2020-08-18T	AmazonDynamoDB Requests-Tier1	APS2-ReadRequestThroughput		
5.15148E+11	2020-08-01T	2020-09-01T	8.3696E+11	Credit	2020-08-09T	2020-08-18T	AmazonS3 Requests-Tier1	APS2-Requests-Tier1		
5.15148E+11	2020-08-01T	2020-09-01T	8.3696E+11	Credit	2020-08-10T	2020-08-10T	AmazonEC2 BoxUsage:RunInstances	APS2-BoxUsage:RunInstances		
5.15148E+11	2020-08-01T	2020-09-01T	8.3696E+11	Credit	2020-08-09T	2020-08-18T	AmazonS3 Requests-Tier2	APS2-Requests-Tier2		
5.15148E+11	2020-08-01T	2020-09-01T	5.15148E+11	Credit	2020-08-06T	2020-08-31T	AmazonS3 Requests-Tier1	APS2-Requests-Tier1		
5.15148E+11	2020-08-01T	2020-09-01T	5.15148E+11	Credit	2020-08-10T	2020-08-21T	AmazonGuard Requests-Tier1	APS2-PaidEventsAnalyzed-Bytes		
5.15148E+11	2020-08-01T	2020-09-01T	5.15148E+11	Credit	2020-08-01T	2020-08-01T	AmazonRoute53 HostedZone	APS2-Requests-Tier1		
5.15148E+11	2020-08-01T	2020-09-01T	5.15148E+11	Credit	2020-08-01T	2020-09-01T	AmazonS3 Requests-Tier1	APS2-Requests-Tier1		



AWS Budgets

- AWS Budgets gives you the ability to set custom budgets that alert you when your costs or usage exceed (or are forecasted to exceed) your budgeted amount

All budgets (1)	Cost budgets (1)	Usage budgets (0)	Reservation budgets (0)	Savings Plans budgets (0)				
Budget name	Type	Current	Budgeted	Forecasted	Current vs. budgeted	Forecasted vs. budgeted		
MyBudget	Cost	\$13.20	\$100.00	\$129.47	<div style="width: 13.2%;">13.2%</div>	<div style="width: 129.47%;">129.47%</div>		

Exam Scenarios

Exam Scenario	Solution
Audit requests to AWS Organizations for creating new accounts by federated users	use CloudTrail and look for the federated identity user name
Employees have created individual AWS accounts not under control. Security team need them in AWS Organizations	Send each account an invitation from the central organization
Need to restrict ability to launch specific instance types for a specific team/account	Use an organizations SCP to deny launches unless the instance type is T2, create an IAM group in the account granting access to T2 instances to the relevant users

Exam Scenarios

Exam Scenario	Solution
Need to ensure that S3 buckets are NEVER deleted in a production account	Use an SCP to deny the s3:DeleteBucket API action
Need to create user-defined cost allocation tags for new account	Use Tag Editor in new account to create user-defined tags and then use the billing and cost management console in the payer account to mark them as cost allocation tags
Separate departments must operate in isolation and only use pre-approved services	Use AWS Organizations to create accounts (Organizations API) and SCPs to control the services available for use

Exam Scenarios

Exam Scenario	Solution
Developers can manipulate IAM policies/roles and need to block them from some services	Use an SCP to block those services
AWS bill is increasing and unauthorized services are being used across accounts	Use AWS Organizations with an SCP to restrict the unauthorized services
Configuring AWS SSO for an Organizations master account. Directory created and full access enabled	Next step is to create a permission set and associate with directory users and groups
Process to create a custom dashboard in CloudWatch for custom metrics after installing agent on EC2	Create metric filters and select custom metrics

Exam Scenarios

Exam Scenario	Solution
Need to test notification settings for CloudWatch alarm with SNS	Use the set-alarm-state CLI command to test
App with EC2 and RDS is running slowly and suspected high CPU	Use CloudWatch metrics to examine resource usage
Site uses CloudFront and S3. Users accessing content that does not exist or they don't have access to	Check the 4XXErrorRate metric in CloudWatch to understand the extent of the issue
Script generates custom CloudWatch metrics from EC2 instance and clock is configured incorrectly by 30 mins	CloudWatch will accept the custom metric data and record it

Exam Scenarios

Exam Scenario	Solution
Need to collect logs from many EC2 instances	Use the unified CloudWatch Agent
External auditor needs to check for unauthorized changes to AWS account	Create an IAM user, assign an IAM policy with read access to CloudTrail logs on Amazon S3
Need to identify who is creating EIPs and not using them	Use CloudTrail and query logs using Athena to search for EIP address events
S3 bucket holds sensitive data. Must monitor object upload / download activity including AWS account and IAM user account of caller and time of API call	Use AWS CloudTrail and enable data event logging

Exam Scenarios

Exam Scenario	Solution
Need to record any modifications or deletions of CloudTrail logs in an S3 bucket	Enable CloudTrail log file integrity validation and enabled MFA delete on the bucket
Large increase in requests to SQS. Need to determine the source of the calls	Use CloudTrail to audit API calls
Need to ensure that S3 buckets have logging enabled without stopping users creating them	Auto remediate with AWS Config managed rule S3_BUCKET_LOGGING_ENABLE
Need to provide real-time compliance reporting for security groups to check that port 80 is not being used	Use the AWS Config restricted-common-ports rule and add port 80

Exam Scenarios

Exam Scenario	Solution
Company wants to limit the AMIs that are used. Need to review compliance with the policy	Create an AWS Config rule to check that only approved AMIs are used
Need to automatically disable access keys that are greater than 90 days old	Use Config rule to identify noncompliant keys and use Systems Manager Automation to remediate
Need to address concerns about exposing sensitive data in buckets without restricting ability to create them	Use AWS Config rules to identify public buckets and send SNS notification to security team
Need to ensure CloudFormation deployment changes are tracked for governance	Use AWS Config

Exam Scenarios

Exam Scenario	Solution
Company needs to verify that specific KMS CMK is used to encrypted EBS volumes	Use AWS Config with the encrypted-volumes managed rule and specify the key ID of the CMK
Need to create replica of existing infrastructure in new account. AWS Service Catalog is used	Most efficient option is to share the portfolio with the new accounts and import into those other accounts
Users have a specialized EC2 instance config and don't want to configure EC2 settings but need to launch/terminate instances. Special instance must only be available to them	Use CloudFormation template with AWS Service Catalog portfolio and grant permissions to users
Shared portfolio is imported into a second AWS account controlled by a different administrator	Admin can add products from the imported portfolio to a local portfolio

Exam Scenarios

Exam Scenario	Solution
Need to monitor costs per user in an account	Activate the createdBy tag and analyze with AWS Cost Explorer
How to check for underutilized EC2 instances?	Use AWS Cost Explorer to generate resource optimization recommendations
Bill is increasing over time, need to determine the cause of increased cost	Use AWS Cost Explorer
Need breakdown of costs per project in a single account using Cost Explorer	Do this by activating cost allocation tags and creating and applying resource tags

Exam Scenarios

Exam Scenario	Solution
Need to check that security best practices are being followed for the AWS account root user	Use AWS Trusted Advisor security checks to review configuration of root user
Costs rising and need to be alerted when a specific spending limit is forecast to be exceeded	Use AWS Budgets
Company needs to track the allocation of reserved instances in consolidated bill	Use the AWS Cost and Usage report
Company needs to integrate AWS maintenance events that may affect their resources into an operations dashboard	Use the AWS Health API

SECTION 14

Security and Compliance

Multi-Factor Authentication in AWS

Something you **know**:



IAM User

EJPx!*21p9%

Password

Something you **have**:



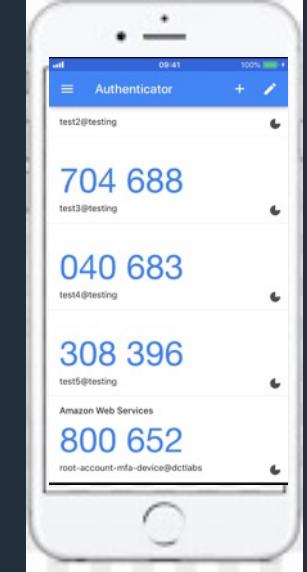
Virtual MFA



Physical MFA



e.g. Google Authenticator on
your smart phone



AWS Managed Policies

- An AWS managed policy is a standalone policy that is created and administered by AWS
- Standalone policy means that the policy has its own Amazon Resource Name (ARN) that includes the policy name
- AWS managed policies are designed to provide permissions for many common use cases
- You cannot change the permissions defined in AWS managed policies

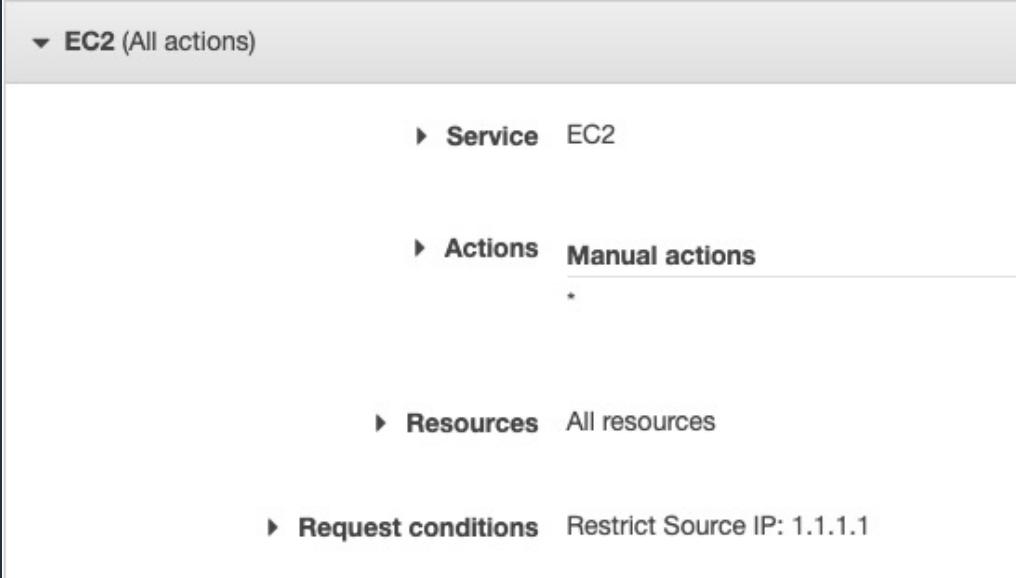
	Policy name ▾	Type	Used as	Description
○	▶ AmazonDynamoDBFullAccess	AWS managed	Permissions policy (3)	Provides full access to Amazon DynamoDB via the AWS Management Console.
○	▶ AmazonDynamoDBFullAccesswithDat...	AWS managed	<i>None</i>	Provides full access to Amazon DynamoDB including Export/Import using AWS Data P...
○	▶ AmazonDynamoDBReadOnlyAccess	AWS managed	<i>None</i>	Provides read only access to Amazon DynamoDB via the AWS Management Console.
○	▶ AmazonEC2ContainerRegistryFullAcc...	AWS managed	<i>None</i>	Provides administrative access to Amazon ECR resources
○	▶ AmazonEC2ContainerRegistryPower...	AWS managed	<i>None</i>	Provides full access to Amazon EC2 Container Registry repositories, but does not allo...
○	▶ AmazonEC2ContainerRegistryReadOnly	AWS managed	<i>None</i>	Provides read-only access to Amazon EC2 Container Registry repositories.
○	▶ AmazonEC2ContainerServiceAutoscal...	AWS managed	Permissions policy (1)	Policy to enable Task Autoscaling for Amazon EC2 Container Service
○	▶ AmazonEC2ContainerServiceEventsR...	AWS managed	<i>None</i>	Policy to enable CloudWatch Events for EC2 Container Service
○	▶ AmazonEC2ContainerServiceforEC2R...	AWS managed	Permissions policy (1)	Default policy for the Amazon EC2 Role for Amazon EC2 Container Service.

AWS Managed Policies

- Some AWS managed policies are designed for specific job functions
- The job-specific AWS managed policies include:
 - Administrator
 - Billing
 - Database Administrator
 - Data Scientist
 - Developer Power User
 - Network Administrator
 - Security Auditor
 - Support User
 - System Administrator
 - View-Only User

Customer Managed Policies

- You can create standalone policies that you administer in your own AWS account, which we refer to as customer managed policies
- You can then attach the policies to multiple principal entities in your AWS account
- When you attach a policy to a principal entity, you give the entity the permissions that are defined in the policy



The screenshot shows the AWS IAM Policy editor interface. A policy named "EC2 (All actions)" is selected. The policy document is displayed on the right, showing a single statement allowing EC2 actions on all resources from the IP address 1.1.1.1.

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Sid": "VisualEditor0",  
      "Effect": "Allow",  
      "Action": "ec2:*",  
      "Resource": "*",  
      "Condition": {  
        "IpAddress": {  
          "aws:SourceIp": "1.1.1.1"  
        }  
      }  
    }  
  ]  
}
```

Allowing access to an S3 bucket from IPv4 and IPv6 Addresses

```
{  
  "Id": "PolicyId2",  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Sid": "AllowIPmix",  
      "Effect": "Allow",  
      "Principal": "*",  
      "Action": "s3:*",  
      "Resource": "arn:aws:s3:::awsexamplebucket1/*",  
      "Condition": {  
        "IpAddress": {  
          "aws:SourceIp": [  
            "54.240.143.0/24",  
            "2001:DB8:1234:5678::/64"  
          ]  
        },  
        "NotIpAddress": {  
          "aws:SourceIp": [  
            "54.240.143.128/30",  
            "2001:DB8:1234:5678:ABCD::/80"  
          ]  
        }  
      }  
    }  
  ]  
}
```

Grant access to instances with a specific tag

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": "ec2:*",  
            "Resource": "*",  
            "Condition": {  
                "StringEquals": {  
                    "ec2:ResourceTag/UserName": "${aws:username}"  
                }  
            }  
        },  
        {  
            "Effect": "Allow",  
            "Action": "ec2:Describe*",  
            "Resource": "*"  
        },  
        {  
            "Effect": "Deny",  
            "Action": [  
                "ec2:CreateTags",  
                "ec2:DeleteTags"  
            ],  
            "Resource": "*"  
        }  
    ]  
}
```

Grant user permission to pass an IAM role

- To pass a role (and its permissions) to an AWS service, a user must have permissions to pass the role to the service.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {"Effect": "Allow",  
         "Action": [  
             "iam:GetRole",  
             "iam:PassRole"  
         ],  
         "Resource": "arn:aws:iam::<account-id>:role/EC2-roles-for-XYZ-*"  
     }]  
}
```

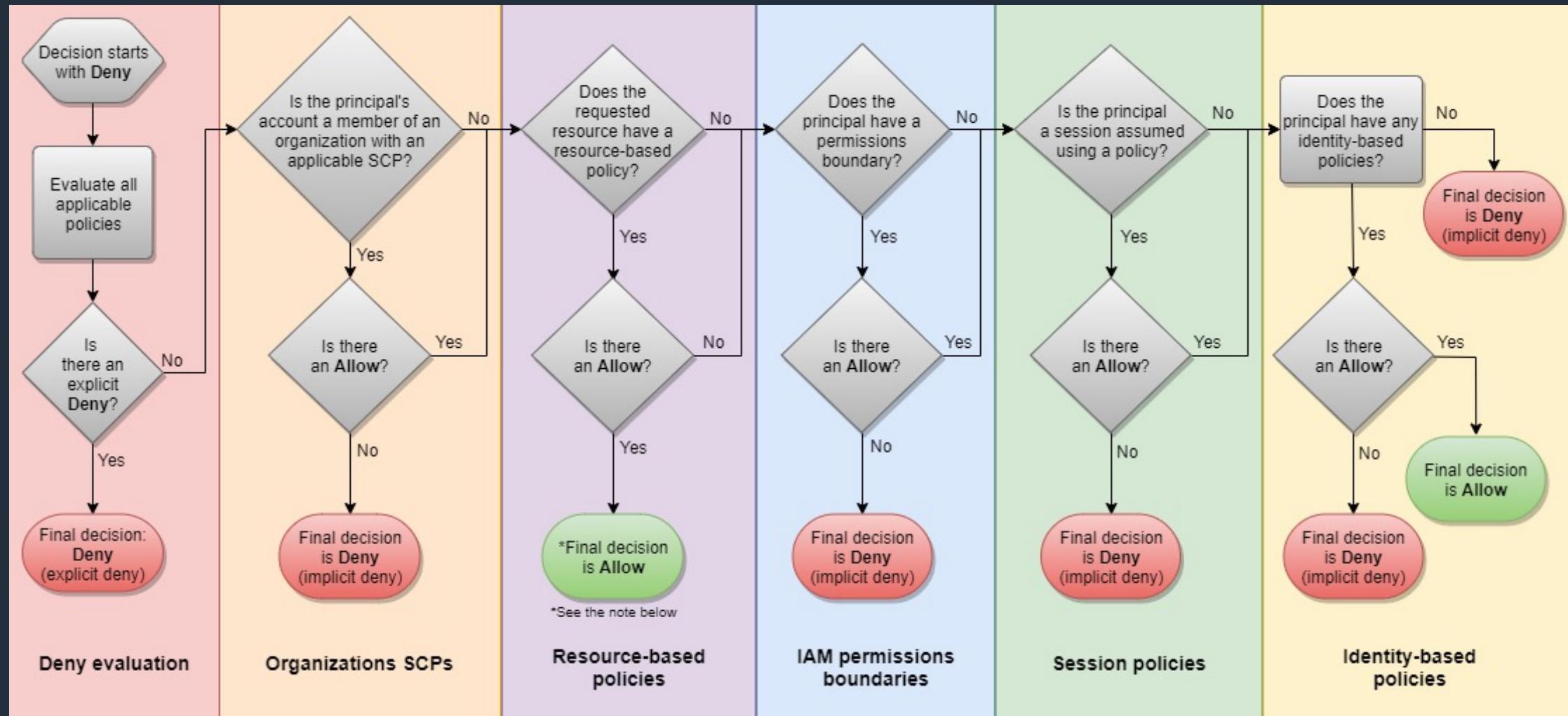
IAM Policy Evaluation Logic

- **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)
- **Resource-based policies** – Resource-based policies grant permissions to the principal (account, user, role, or federated user) specified as the principal
- **IAM permissions boundaries** – Permissions boundaries are an advanced feature that sets the maximum permissions that an identity-based policy can grant to an IAM entity (user or role)
- **AWS Organizations service control policies (SCPs)** – Organizations SCPs specify the maximum permissions for an organization or organizational unit (OU)
- **Session policies** – Session policies are advanced policies that you pass as parameters when you programmatically create a temporary session for a role or federated user

IAM Policy Evaluation Logic

- By default, all requests are implicitly denied. (Alternatively, by default, the AWS account root user has full access.)
- An explicit allow in an identity-based or resource-based policy overrides this default
- If a permissions boundary, Organizations SCP, or session policy is present, it might override the allow with an implicit deny
- An explicit deny in any policy overrides any allows

IAM Policy Evaluation Logic

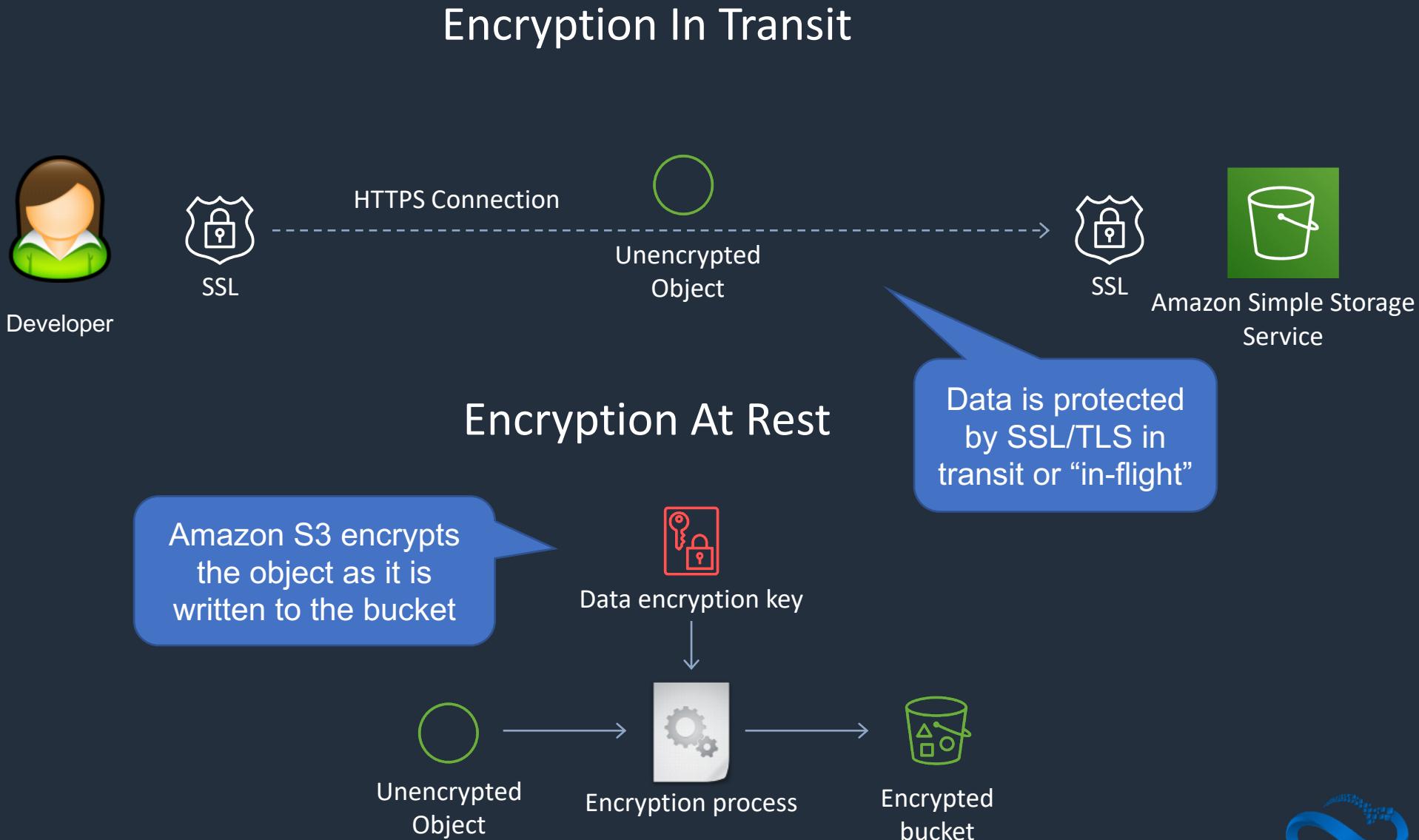


Amazon Inspector

- Inspector is an automated security assessment service that helps improve the security and compliance of applications deployed on AWS
- Inspector automatically assesses applications for vulnerabilities or deviations from best practices
- Uses an agent installed on EC2 instances
- Instances must be tagged

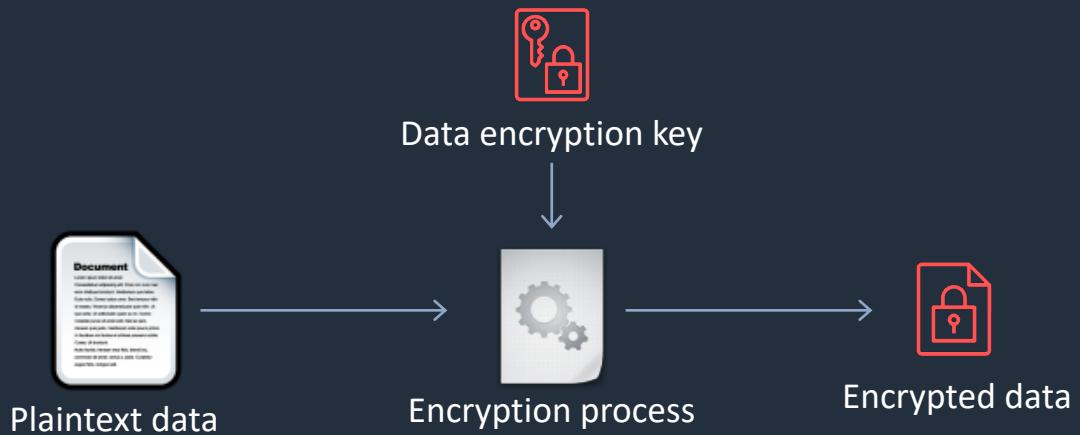
Severity ⓘ ▾	Date ▾	Finding	Target	Template	Rules Package
Medium	Today at 5:35 PM (GMT...)	On instance i-0279e0553d4a19a74, TCP port 22 w...	Assessment-Target-All-Instances-...	Assessment-Template-Def...	Network Reachability-1.1
Medium	Today at 5:35 PM (GMT...)	On instance i-0279e0553d4a19a74, TCP port 338...	Assessment-Target-All-Instances-...	Assessment-Template-Def...	Network Reachability-1.1
Low	Today at 5:35 PM (GMT...)	On instance i-0279e0553d4a19a74, TCP port 80 w...	Assessment-Target-All-Instances-...	Assessment-Template-Def...	Network Reachability-1.1
Informational	Today at 5:35 PM (GMT...)	Aggregate network exposure: On instance i-0279e...	Assessment-Target-All-Instances-...	Assessment-Template-Def...	Network Reachability-1.1

Encryption – In Transit vs At Rest

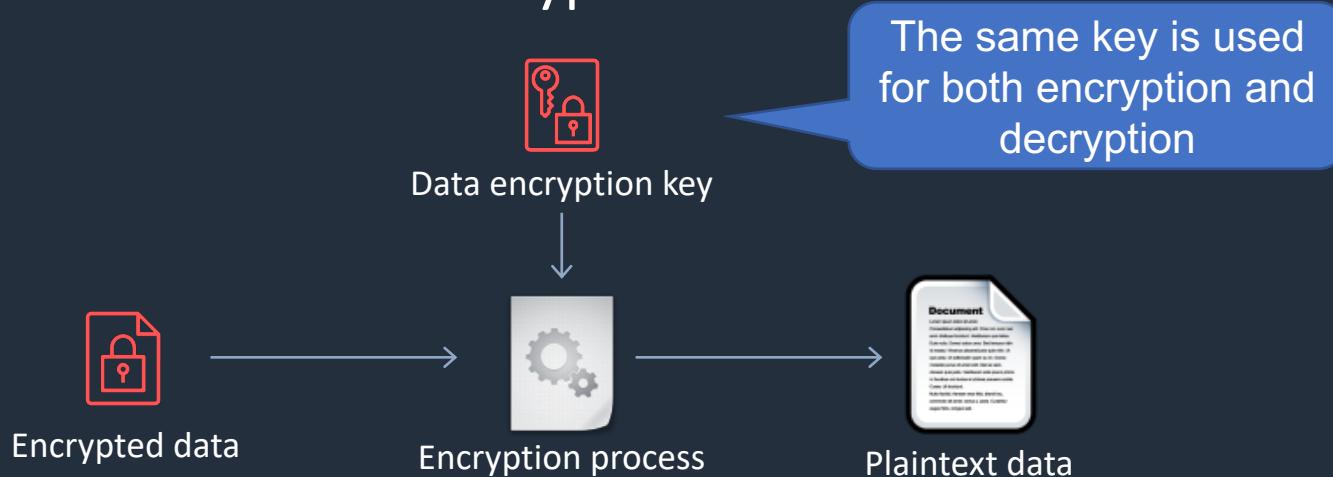


Symmetric Encryption

Encryption

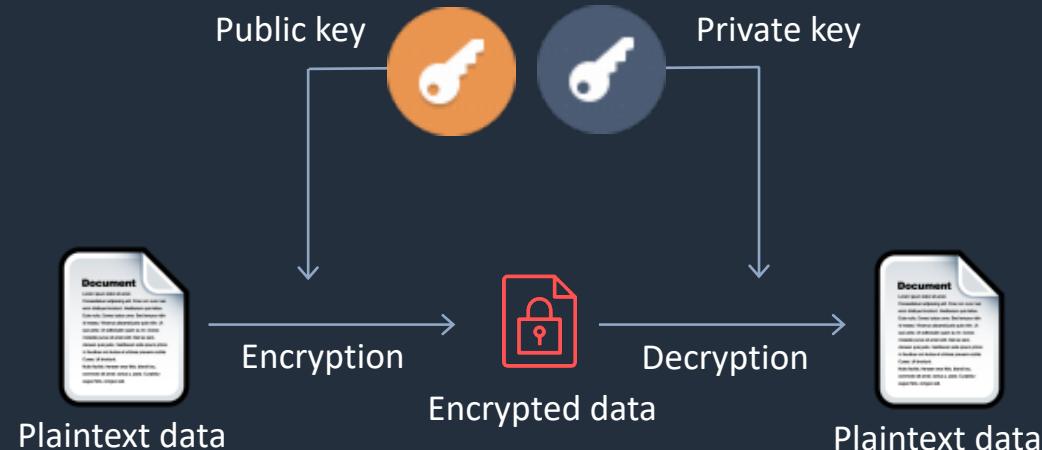


Decryption



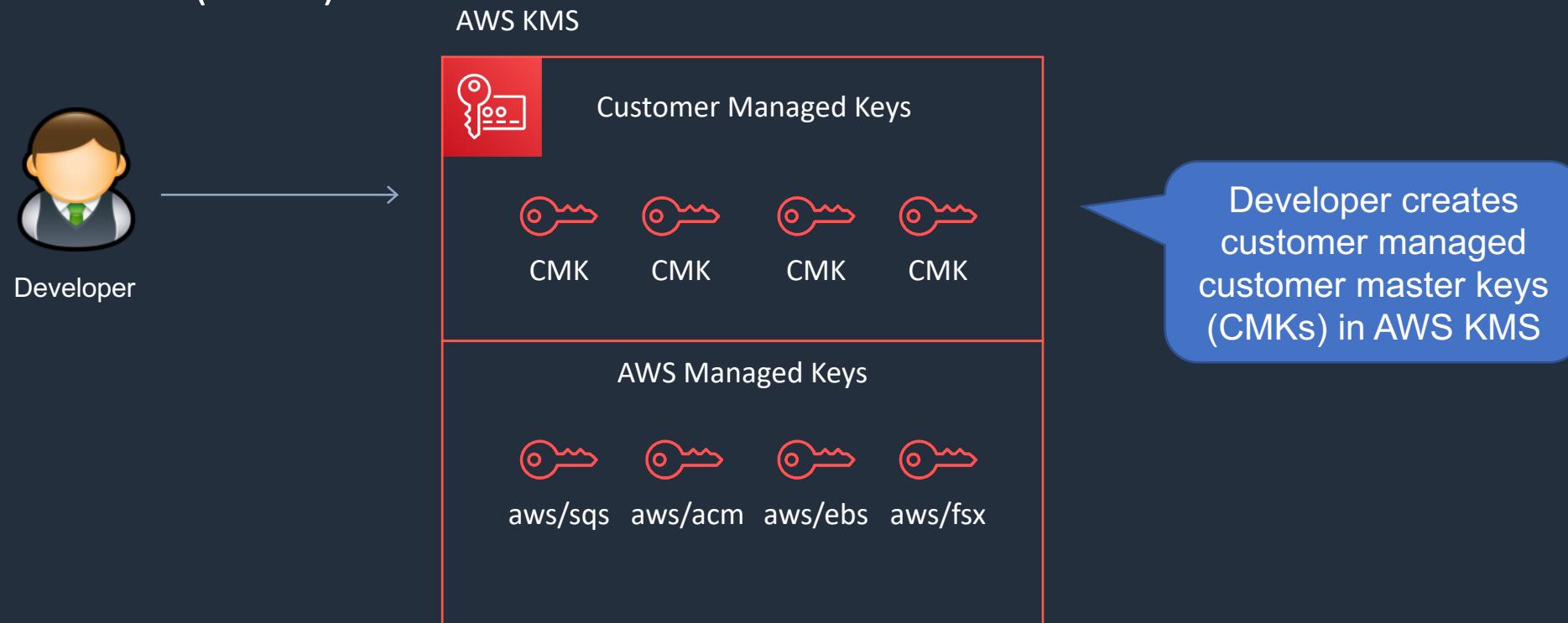
Asymmetric Encryption

- Asymmetric encryption is also known as public key cryptography
- Messages encrypted with the public key can only be decrypted with the private key
- Messages encrypted with the private key can be decrypted with the public key
- Examples include SSL/TLS and SSH



AWS Key Management Service (KMS)

- AWS KMS is a service for creating and controlling encryption keys
- The customer master keys (CMKs) are protected by hardware security modules (HSMs)



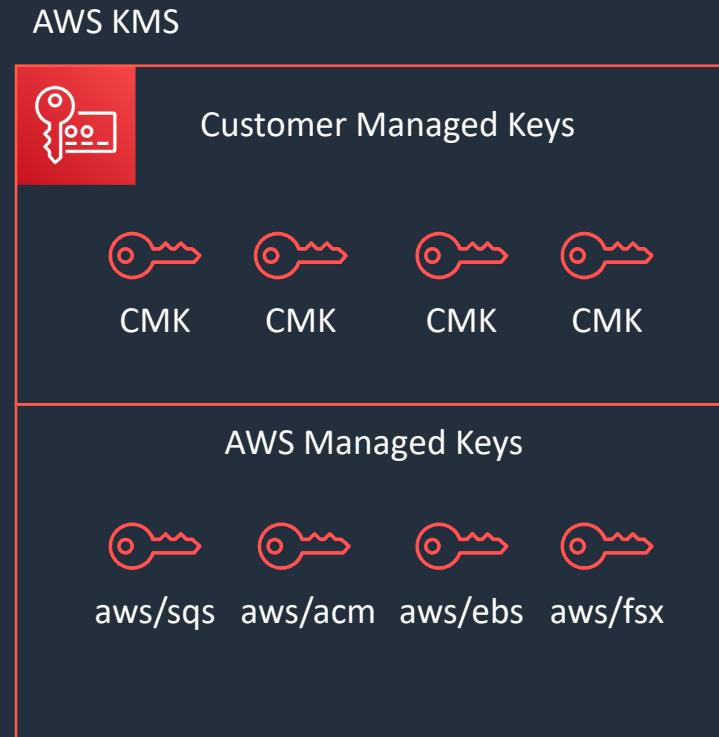
AWS Key Management Service (KMS)

With AWS KMS you can also perform the following cryptographic functions using master keys:

- Encrypt, decrypt, and re-encrypt data
- Generate data encryption keys that you can export from the service in plaintext or encrypted under a master key that doesn't leave the service

AWS KMS – Customer Master Keys (CMKs)

- Customer master keys are the primary resources in AWS KMS
- The CMK also contains the key material used to encrypt and decrypt data
- AWS KMS supports symmetric and asymmetric CMKs
- CMKs are created in AWS KMS. Symmetric CMKs and the private keys of asymmetric CMKs never leave AWS KMS unencrypted
- By default, AWS KMS creates the key material for a CMK
- Can also import your own key material
- A CMK can encrypt data up to 4KB in size
- A CMK can generate, encrypt and decrypt Data Encryption Keys (DEKs)



AWS KMS – AWS Managed CMKs

- These are created, managed, and used on your behalf by an AWS service that is integrated with AWS KMS
- You cannot manage these CMKs, rotate them, or change their key policies
- You also cannot use AWS managed CMKs in cryptographic operations directly; the service that creates them uses them on your behalf
- You do not pay a monthly fee for AWS managed CMKs. They can be subject to fees for use in excess of the free tier, but some AWS services cover these costs for you.

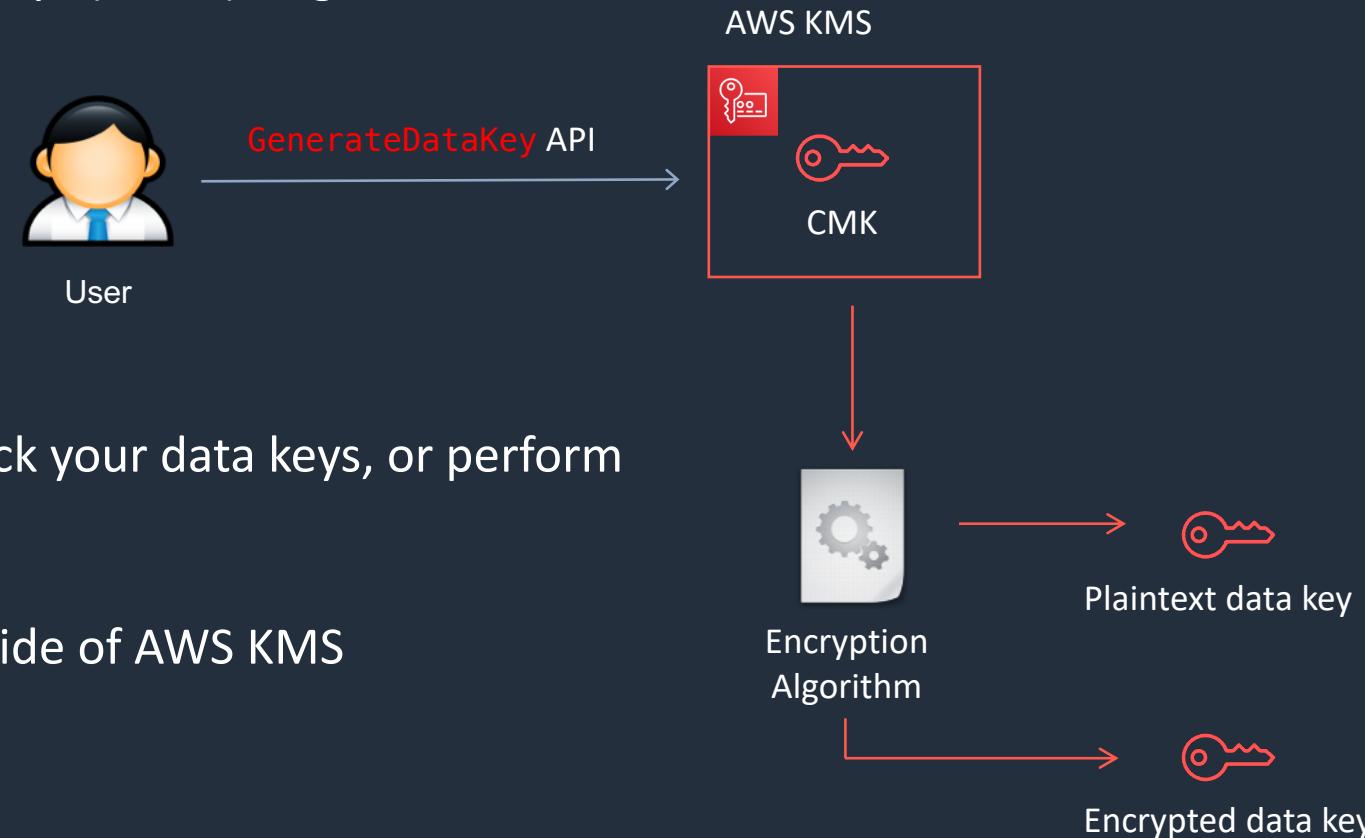
Alias	Key ID
aws/sqs	025b9386-b1f8-4fa9-84e2-ac3220b1de59
aws/acm	2d604e85-c2d4-42dc-ab0b-0b356f5fe26e
aws/codecommit	41fea9df-e447-4992-8af7-6ddec6d81175
aws/elasticfilesystem	460c4f05-fe98-4a35-b940-3e1992f04314
aws/glue	617516fe-bf19-4da4-a743-2b13c41973e1
aws/lambda	7f513d01-784b-41b9-9c51-51621db7b5e1
aws/ebs	b9baa4f6-3e87-4256-af6a-d181940df286
aws/lightsail	bc7ba666-8e17-444a-800c-d3d4be303a97
aws/fsx	cebc434c-ee2b-4a61-9b5a-f63be9fdb068
aws/kinesis	d99014b5-09d4-480d-9b2a-3a7d7e3e9c5b

AWS KMS – Customer Master Keys (CMKs)

Type of CMK	Can view	Can manage	Used only for my AWS account	Automatic rotation
Customer managed CMK	Yes	Yes	Yes	Optional. Every 365 days
AWS managed CMK	Yes	No	Yes	Required. Every 1095 days
AWS owned CMK	No	No	No	Varies

AWS KMS – Data Encryption Keys

- Data keys are encryption keys that you can use to encrypt data, including large amounts of data and other data encryption keys
- You can use AWS KMS customer master keys (CMKs) to generate, encrypt, and decrypt data keys



- AWS KMS does not store, manage, or track your data keys, or perform cryptographic operations with data keys
- You must use and manage data keys outside of AWS KMS

AWS CloudHSM

- AWS CloudHSM is a cloud-based hardware security module (HSM) that enables you to easily generate and use your own encryption keys on the AWS Cloud
- You can manage your own encryption keys using FIPS 140-2 Level 3 validated HSMs.
- CloudHSM runs in your VPC

	CloudHSM	AWS KMS
Tenancy	Single-tenant HSM	Multi-tenant AWS service
Availability	Customer-managed durability and available	Highly available and durable key storage and management
Root of Trust	Customer managed root of trust	AWS managed root of trust
FIPS 140-2	Level 3	Level 2 / Level 3 in some areas
3 rd Party Support	Broad 3 rd Party Support	Broad AWS service support

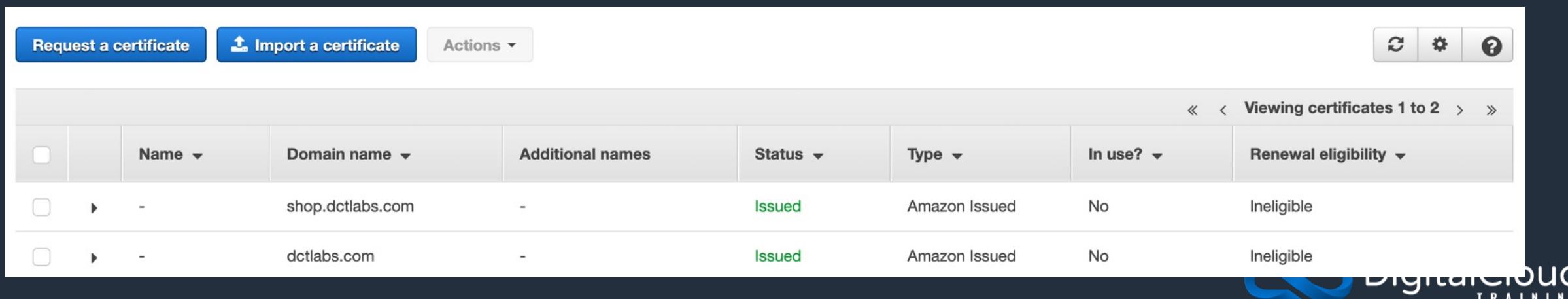
AWS CloudHSM

Benefits:

- FIPS 140-2 level 3 validated HSMs
- You can configure AWS Key Management Service (KMS) to use your AWS CloudHSM cluster as a custom key store rather than the default KMS key store
- Managed service and automatically scales
- Retain control of your encryption keys - you control access (and AWS has no visibility of your encryption keys)

AWS Certificate Manager (ACM)

- ACM is used for creating and managing public SSL/TLS certificates
- You can use public certificates provided by ACM (ACM certificates) or certificates that you import into ACM
- ACM certificates can secure multiple domain names and multiple names within a domain
- You can also use ACM to create wildcard SSL certificates that can protect an unlimited number of subdomains



The screenshot shows the AWS Certificate Manager (ACM) console interface. At the top, there are three buttons: "Request a certificate", "Import a certificate", and "Actions". To the right of these are icons for refresh, settings, and help. Below the buttons is a table header with columns: "Name", "Domain name", "Additional names", "Status", "Type", "In use?", and "Renewal eligibility". The table displays two rows of certificate information:

Name	Domain name	Additional names	Status	Type	In use?	Renewal eligibility	
<input type="checkbox"/>	-	shop.dctlabs.com	-	Issued	Amazon Issued	No	Ineligible
<input type="checkbox"/>	-	dctlabs.com	-	Issued	Amazon Issued	No	Ineligible

At the bottom right of the table, it says "Viewing certificates 1 to 2".

Certificate Renewal with ACM

- Managed renewal for SSL/TLS certificates
- Automatic if using DNS validation; email notification otherwise
- Provided for both public and private ACM certificates

Certificate Renewal with ACM

- At 60 days prior to expiration, ACM checks for the renewal criteria:
 - The certificate is currently in use by an AWS service
 - A valid DNS record for the apex domain exists
 - The required CNAME token is present and accessible in the DNS record
 - Each domain and subdomain that is named in the certificate is present in the DNS record
- If all of these criteria are met, ACM considers the domain names validated and renews the certificate

AWS Web Application Firewall (WAF)

- AWS WAF is a web application firewall
- WAF lets you create rules to filter web traffic based on conditions that include IP addresses, HTTP headers and body, or custom URIs
- WAF makes it easy to create rules that block common web exploits like SQL injection and cross site scripting
- WAF can be used to protect CloudFront distributions, ALBs (and the resources behind them), and API Gateway APIs

AWS Web Application Firewall (WAF)

- Web ACLs - You use a web access control list (ACL) to protect a set of AWS resources
- Rules - Each rule contains a statement that defines the inspection criteria, and an action to take if a web request meets the criteria
- Rules groups – You can use rules individually or in reusable rule groups



AWS Web Application Firewall (WAF)

- IP Sets - An IP set provides a collection of IP addresses and IP address ranges that you want to use together in a rule statement
- Regex pattern set - A regex pattern set provides a collection of regular expressions that you want to use together in a rule statement

AWS Web Application Firewall (WAF)

A rule action tells AWS WAF what to do with a web request when it *matches* the criteria defined in the rule:

- Count – AWS WAF counts the request but doesn't determine whether to allow it or block it. With this action, AWS WAF continues processing the remaining rules in the web ACL
- Allow – AWS WAF allows the request to be forwarded to the AWS resource for processing and response
- Block – AWS WAF blocks the request and the AWS resource responds with an HTTP 403 (Forbidden) status code

AWS Web Application Firewall (WAF)

Match statements compare the web request or its origin against conditions that you provide

Match Statement	Description
Geographic match	Inspects the request's country of origin
IP set match	Inspects the request against a set of IP addresses and address ranges
Regex pattern set	Compares regex patterns against a specified request component
Size constraint	Checks size constraints against a specified request component
SQLi attack	Inspects for malicious SQL code in a specified request component
String match	Compares a string to a specified request component.
XSS scripting attack	Inspects for cross-site scripting attacks in a specified request component

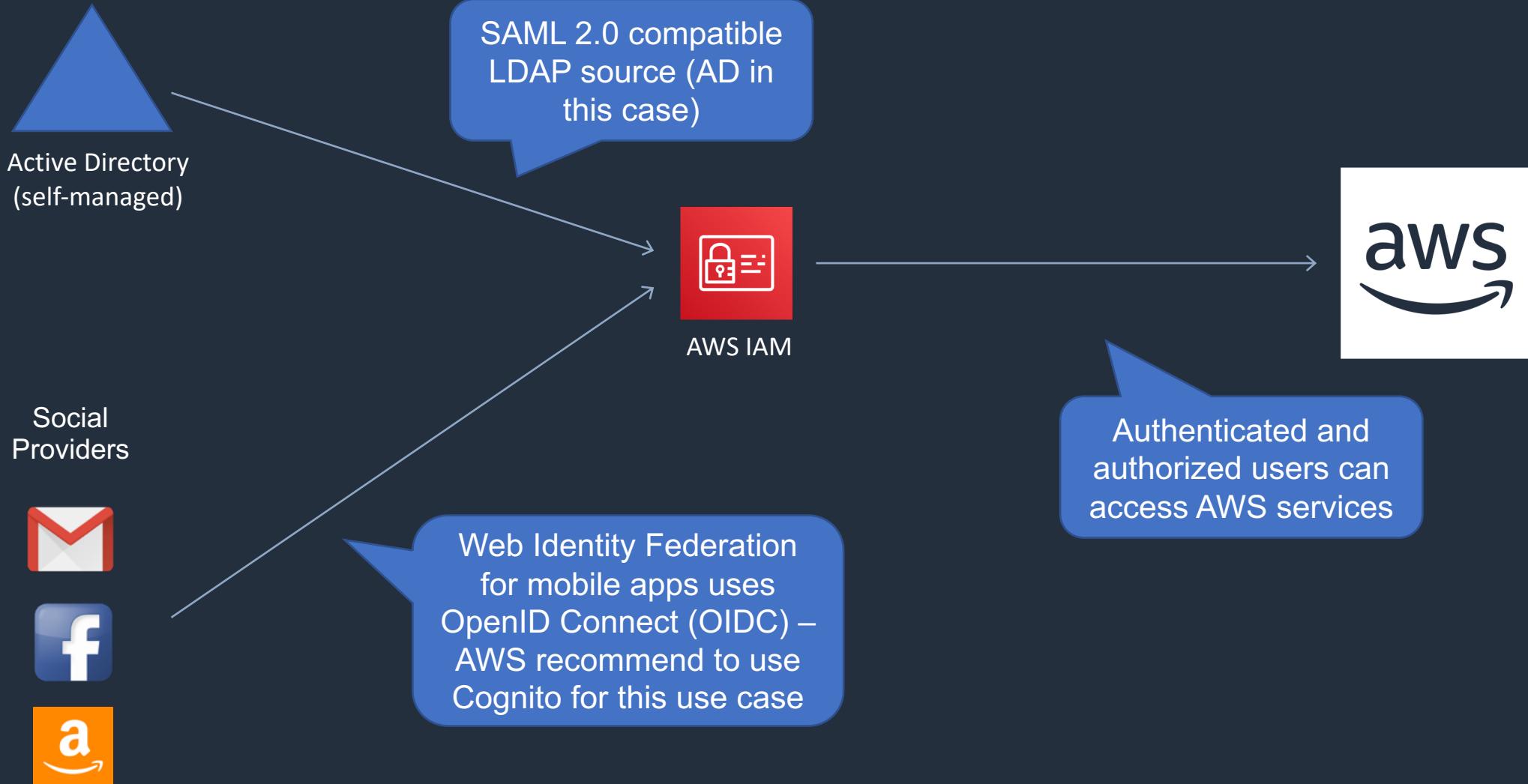
AWS Shield

- AWS Shield is a managed Distributed Denial of Service (DDoS) protection service
- Safeguards web application running on AWS with always-on detection and automatic inline mitigations
- Helps to minimize application downtime and latency
- Two tiers – Standard and Advanced
- Integrated with Amazon CloudFront

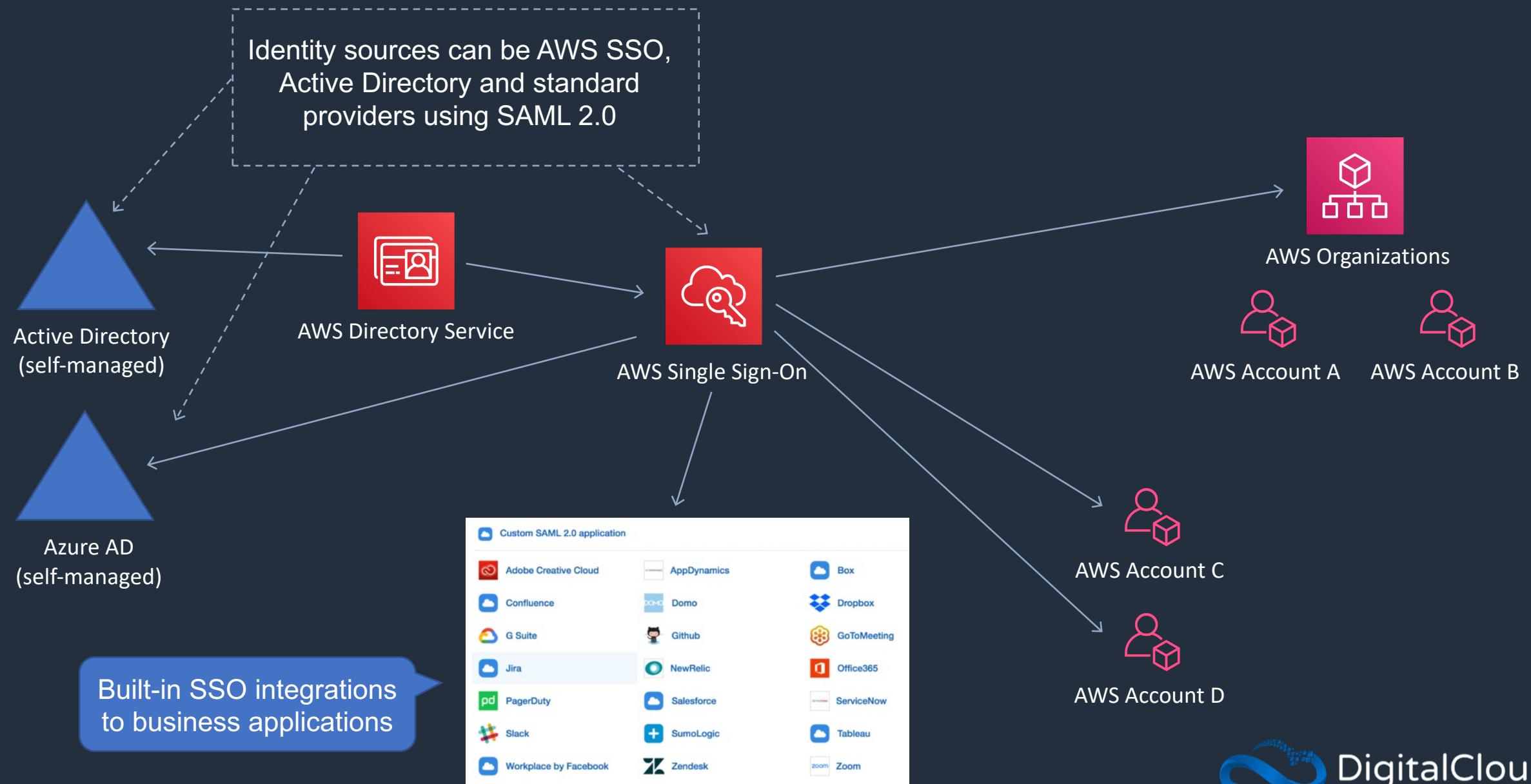
AWS Artifact

- AWS Artifact is your go-to, central resource for compliance-related information that matters to you
- It provides on-demand access to AWS' security and compliance reports and select online agreements
- Reports available in AWS Artifact include our Service Organization Control (SOC) reports, Payment Card Industry (PCI) reports, and certifications from accreditation bodies across geographies and compliance verticals that validate the implementation and operating effectiveness of AWS security controls
- Agreements available in AWS Artifact include the Business Associate Addendum (BAA) and the Nondisclosure Agreement (NDA)

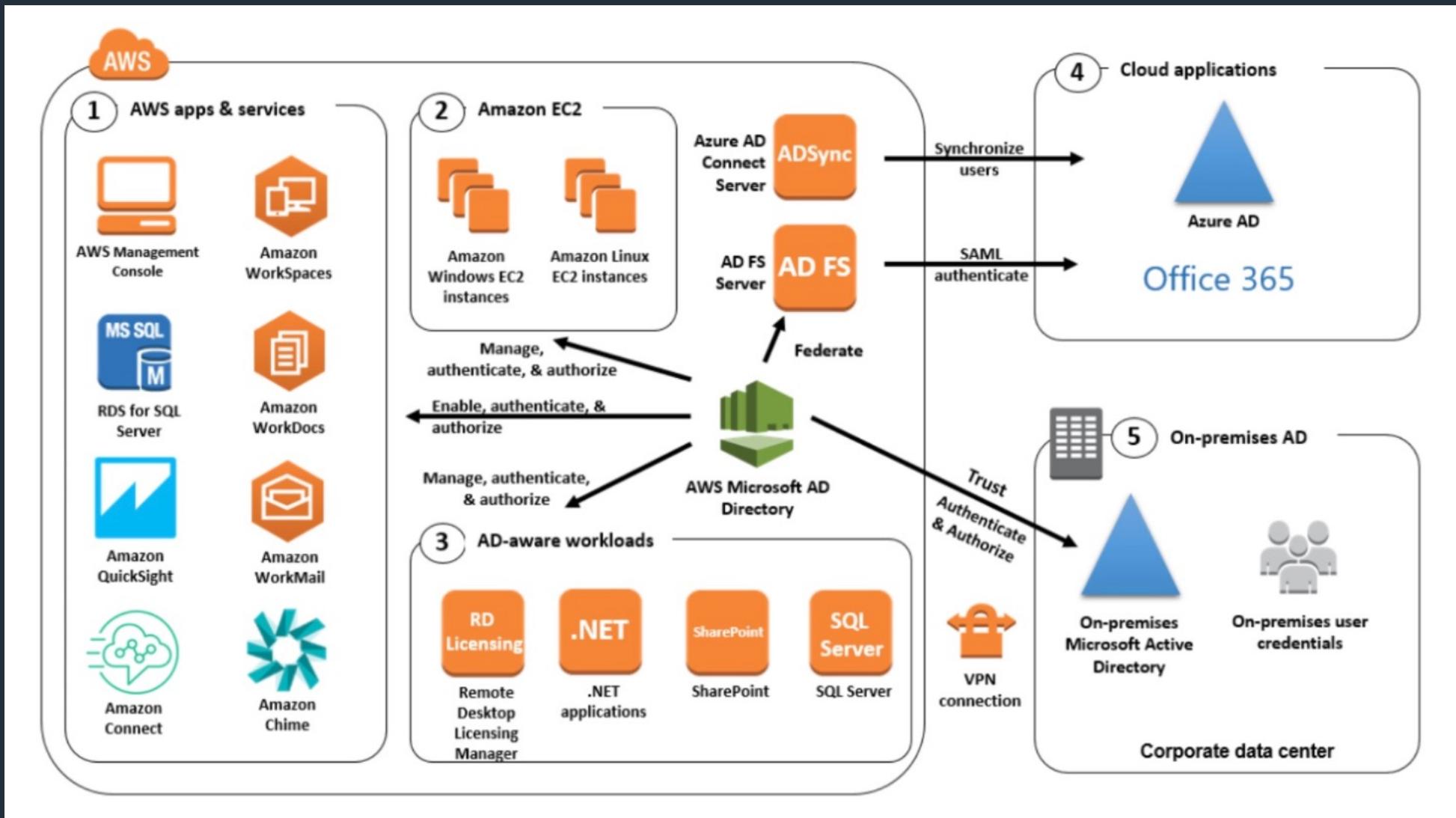
Identity Providers and Federation



AWS Single Sign-on (SSO)



AWS Directory Service - AWS Managed Microsoft AD



AWS Directory Service - AWS Managed Microsoft AD

- Fully managed AWS services on AWS infrastructure
- Best choice if you have more than 5000 users and/or need a trust relationship set up
- Runs on a Windows Server
- You can setup trust relationships to extend authentication from on-premises Active Directories into the AWS cloud
- On-premise users and groups can access resources in either domain using SSO
- Requires a VPN or Direct Connect connection
- Can be used as a standalone AD in the AWS cloud

AWS Directory Service - Simple AD

- An inexpensive Active Directory-compatible service with common directory features.
- Standalone, fully managed, directory on the AWS cloud
- Simple AD is generally the least expensive option
- Best choice for less than 5000 users and don't need advanced AD features

AWS Directory Service - AD Connector



AWS Directory Service – AD Connector

- AD Connector is a directory gateway for redirecting directory requests to your on-premise Active Directory
- Connects your existing on-premise AD to AWS
- Best choice when you want to use an existing Active Directory with AWS services
- You can also join EC2 instances to your on-premise AD through AD Connecto.
- You can also login to the AWS Management Console using your on-premise AD DCs for authentication

AWS Directory Service – AD Connector vs Simple AD

Directory Service	Service Description	Use Case
AWS Directory Service for Microsoft Active Directory	AWS-managed full Microsoft AD running on Windows Server 2012 R2	Enterprises that want hosted Microsoft AD or you need LDAP for Linux apps
AD Connector	Allows on-premises users to log into AWS services with their existing AD credentials. Also allows EC2 instances to join AD domain	Single sign-on for on-premises employees and for adding EC2 instances to the domain
Simple AD	Low scale, low cost, AD implementation based on Samba. Can also join EC2 instances to the domain	Simple user directory, or you need LDAP compatibility.

Exam Scenarios

Exam Scenario	Solution
Company wishes to force users to change their passwords regularly	Create an IAM password policy and enabled password expiration
Need to restrict access to a bucket based on source IP range	Use bucket policy with "Condition": "NotIpAddress": statement
Need to control access to group of EC2 instances with specific tags	Use an IAM policy with a condition element granting access based on the tag and attach an IAM policy to the user or groups that require access
IAM policy for SQS queue allows too much access. Who is responsible for correcting the issue?	According the AWS shared responsibility mode, this is a customer responsibility

Exam Scenarios

Exam Scenario	Solution
Data is encrypted with AWS KMS customer-managed CMKs. Need to enable rotation ensuring the data remains readable	Just enable key rotation in AWS KMS for the CMK (backing key is rotated, data key is not changed)
Company must rotate encryption keys once a year with least effort	Use customer-managed CMK and enabled automatic key rotation
App uses KMS CMK with imported key material and references the CMK by alias in the application. Must be rotated every 6 months	To rotate, create a new CMK with new imported material and update the key alias to point to new CMK

Exam Scenarios

Exam Scenario	Solution
Certificate request rejected by ACM	Submit a request for a certificate using the correct domain name NOT the ALB FQDN
Security findings are missing in Amazon Inspector	Verify agent installed on affected instances and restart agent
Security team need to verify vulnerabilities and exposures are addressed for EC2 instances regularly	Use Amazon Inspector and perform regular assessments
There may be a vulnerable version of software installed on EC2 instances and need to check	Create and run an Amazon Inspector assessment template

Exam Scenarios

Exam Scenario	Solution
Need to use information in request header to count requests from each front-end server	Use a string match statement
Large amount of suspicious HTTP requests hitting an ALB from various source IPs	Block the traffic using AWS WAF with a rate-based rule and a defined threshold
Many 404 errors being sent to one IP address every minute. Bot may be collecting info	Use AWS WAF to block the activity
Website has been deployed and penetration testing shows its vulnerable to cross-site scripting	Use AWS WAF to mitigate cross-site scripting attacks

Exam Scenarios

Exam Scenario	Solution
Application is under repeated DDoS attacks. Need to minimize downtime and require 24/7 support	Setup AWS Shield Advanced
Company needs to understand the PCI status of the AWS infrastructure	Use AWS Artifact to locate this information

Exam Scenarios

Exam Scenario	Solution
Company uses LDAP and needs to implement access control in AWS as part of an integration between internal and cloud	Need to configure SAM federation of IAM users and groups with the LDAP DB and map LDAP user and groups to IAM roles
Permissions policy for cross-account access must be created and attached. Who is responsible for doing this?	According to the AWS shared responsibility model, this is a customer responsibility
Company wishes to move from IAM user accounts to using on-premises Active Directory accounts for AWS management console access	Configure a VPN tunnel and use Active Directory Connector

THE END

Hope you enjoyed the
course!

