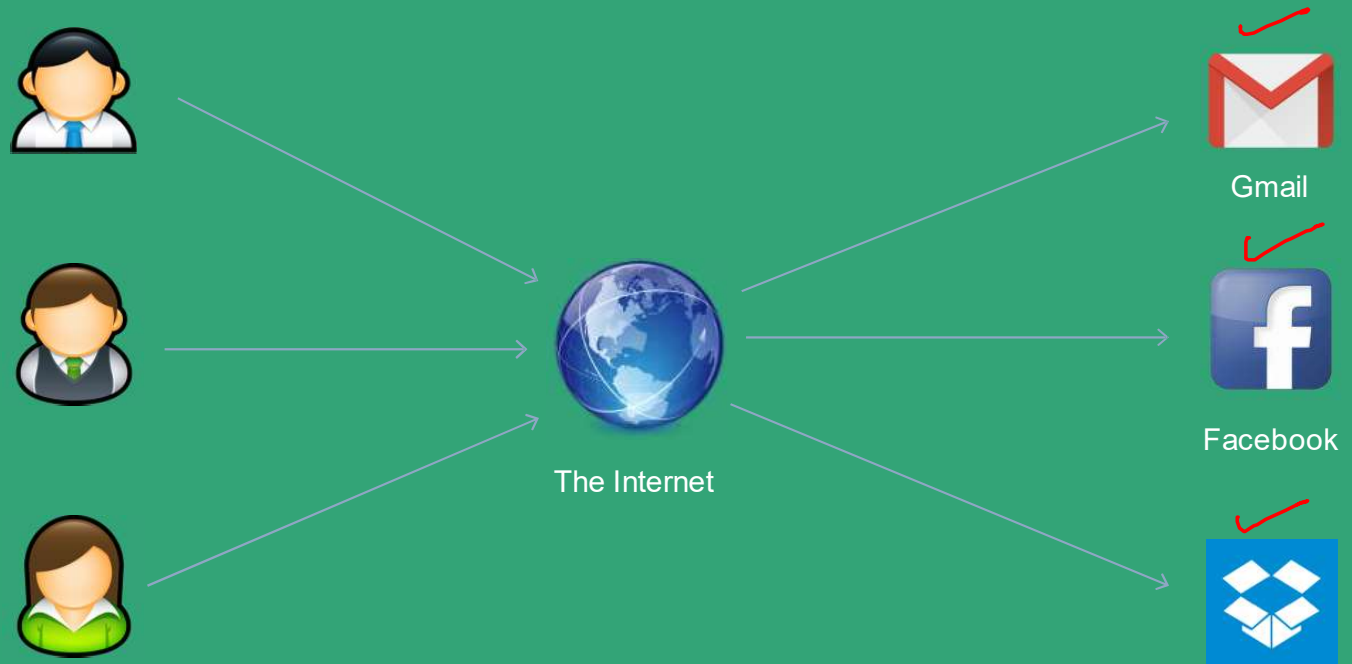
AWS Cloud Practitioner

Introduction

Overview of Cloud Computing

Section 2: What is Cloud? Consumer Examples



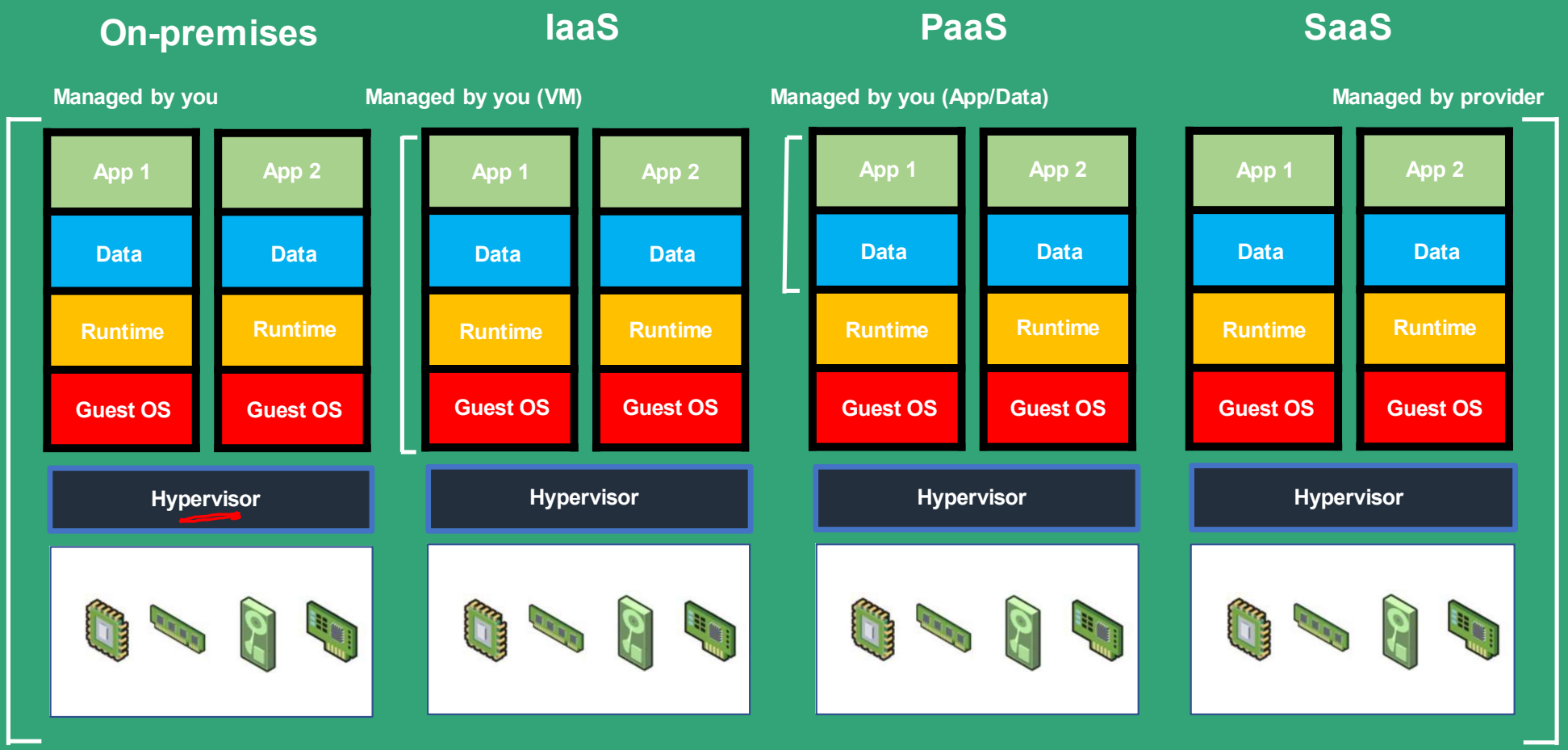
Section 2: Cloud Terminology

Term	Description
Cloud Computing	Cloud computing is the on-demand delivery of IT services from a third-party provider over the Internet
Cloud Service	The IT capability that is being provided by the cloud provider
Cloud Provider / Cloud Service Provider	A company that provides a cloud service to organizations and/or individuals
Consumer	The organization or individual who uses the cloud service
”Pay as you go” or “pay per use”	You are charged only for what you use. Analogous to a utility bill
Multi-tenant	Multiple customers consume services delivered using shared infrastructure
“x” as a service	Some cloud capability is delivered to consumers as a service

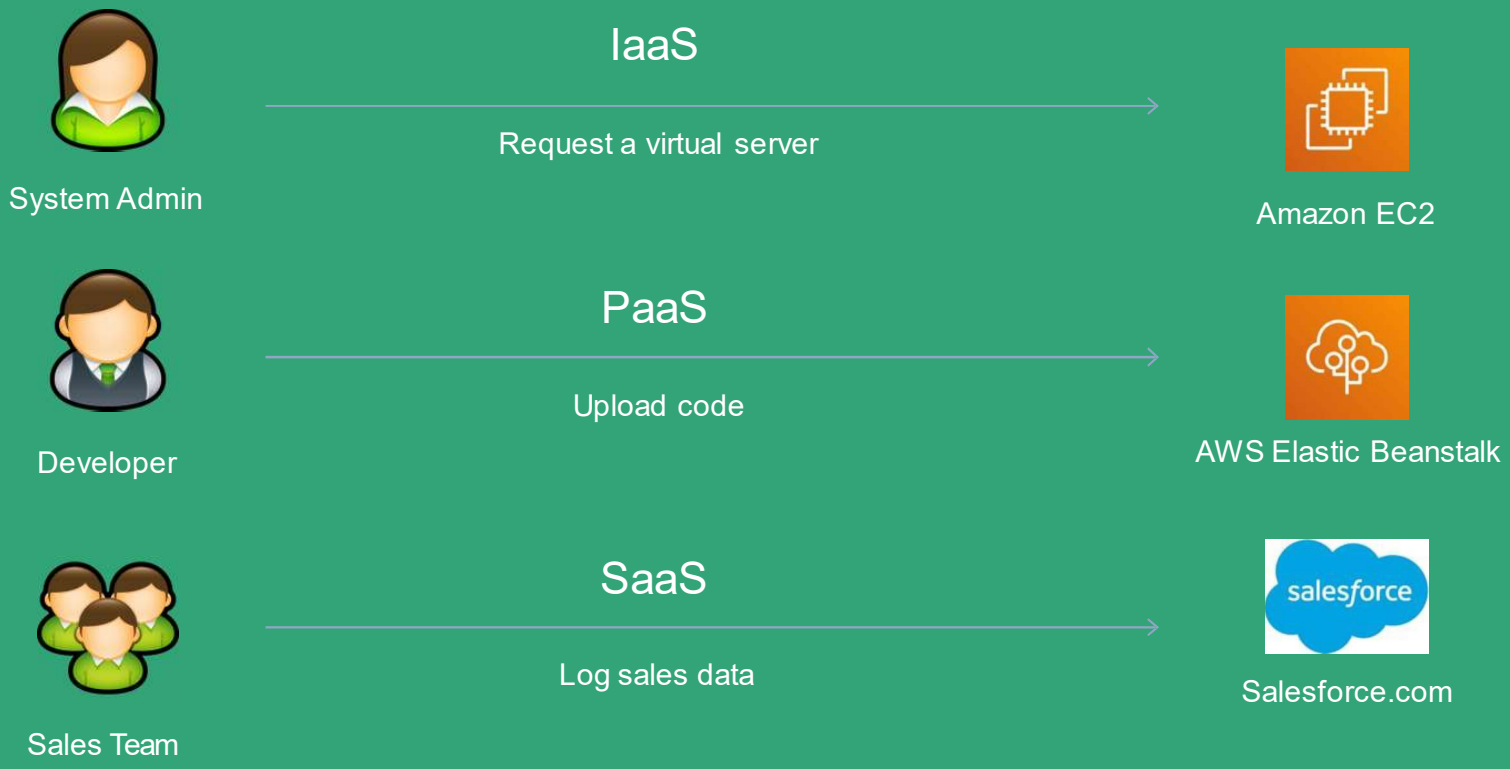
Section 2: Key Characteristics of Cloud Computing

Name	Description
On-demand, self-service	A user can consume cloud resources, as needed, automatically, and without human interaction
Broad network access	Capabilities are available over the network using standard mechanisms. Can be the Internet or a Wide Area Network (WAN)
Resource pooling	The providers resources are pooled and serve multiple consumers using a multi-tenant model
Rapid elasticity	Capabilities can scale “elastically” based on demand
Measured service	Resource usage is monitored and metered

Section 2: Cloud Computing Service Models



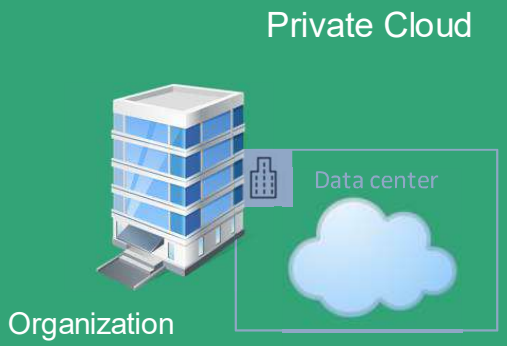
Section 2: IaaS, PaaS, and SaaS Examples



Section 2: Cloud Computing Deployment Models

Name	Description	Examples
Private Cloud	An enterprise deploys their own infrastructure and applications into their own data center	VMware, Microsoft, RedHat, OpenStack
Public Cloud	The IT services that you consume are hosted and delivered from a third-party and accessed over the Internet	AWS, Microsoft Azure, Google Cloud Platform
Hybrid Cloud	A combination of on-premises, private cloud, and public cloud services are consumed	
Multicloud	Usage of two or more public clouds at a time, and possibly multiple private clouds	

Section 2: Deployment Models – Private Cloud



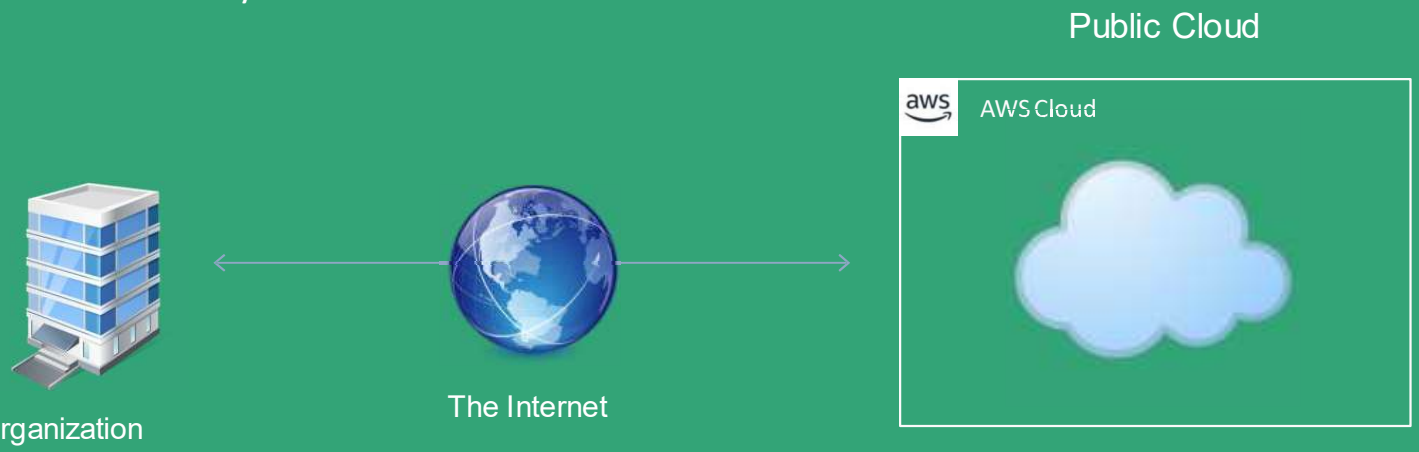
Benefits:

- Complete control of the entire stack
- Security – in a few cases, organizations may need to keep all or some of their applications and data in house

Section 2: Deployment Models – Public Cloud

Benefits:

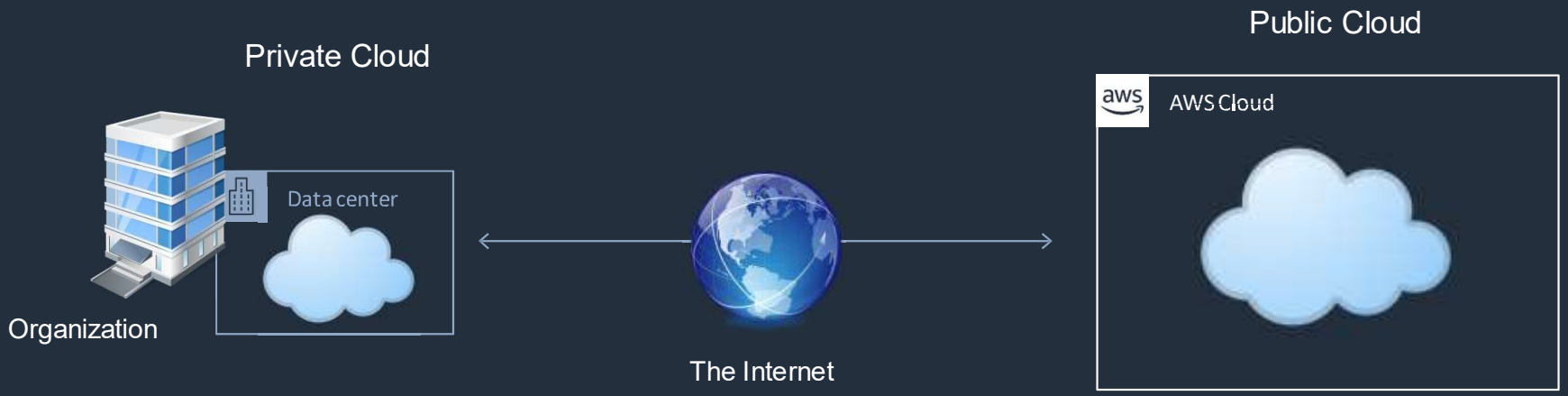
- Variable expense, instead of capital expense
- Economies of scale
- Massive elasticity



Section 2: Deployment Models – Hybrid Cloud

Benefits:

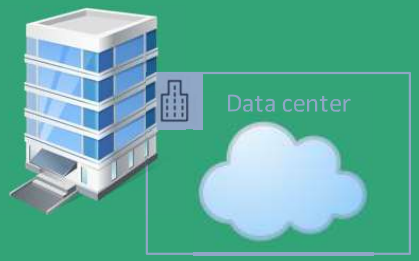
- Allows companies to keep the critical applications and sensitive data in a traditional data center environment or private cloud
- Take advantage of public cloud resources like SaaS, for the latest applications, and IaaS, for elastic virtual resources
- Facilitates portability of data, apps and services and more choices for deployment models



Section 2: Legacy IT

Costs:

- Data center
- Power
- Computing Equipment
- Software licenses
- Maintenance contracts
- Staff wages
- Etc.



Limitations:

- Very capital heavy (CAPEX) – need to purchase everything
- Operational overhead is high
- Scalability is very limited

Operations

- Data center operations and security
- Equipment installation and maintenance
- System monitoring and operations activities
- Backup/recovery
- Etc.

Section 2: The 6 Advantages of Cloud

	Name	Description
1	Trade capital expense for variable expense	Instead of investing in data centers before you know how you're going to use them, pay only when, and for how much, you consume
2	Benefit from massive economies of scale	Achieve a lower variable cost due to AWS' scale
3	Stop guessing about capacity	Eliminate guessing, scale as demand dictates
4	Increase speed and agility	Easily and quickly scale your usage
5	Stop spending money running and maintaining data centers	Focus on business growth and innovation instead!
6	Go global in minutes	Easily deploy applications in multiple regions around the world

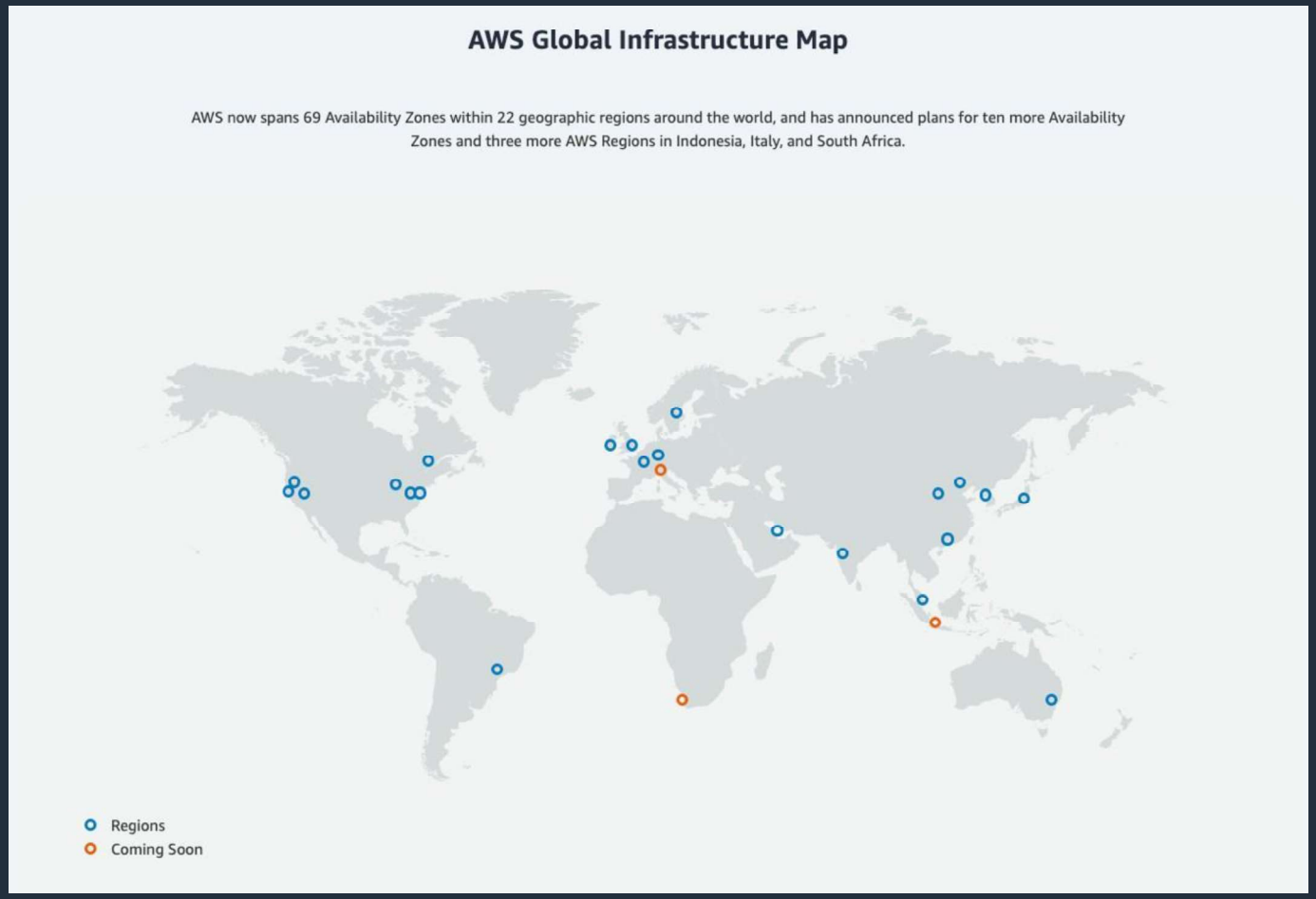
https://docs.aws.amazon.com/whitepapers/latest/aws-overview/six-advantages-of-cloud-computing.html

AWS Cloud Overview

Section 3: Amazon Web Services (AWS) Today

- Over 165 services including computing, storage, networking, database, analytics, media services, machine learning, management, mobile, and IoT
- 22 geographical regions of presence
- $25 billion in revenue in 2018

Section 3: AWS Global Infrastructure Map



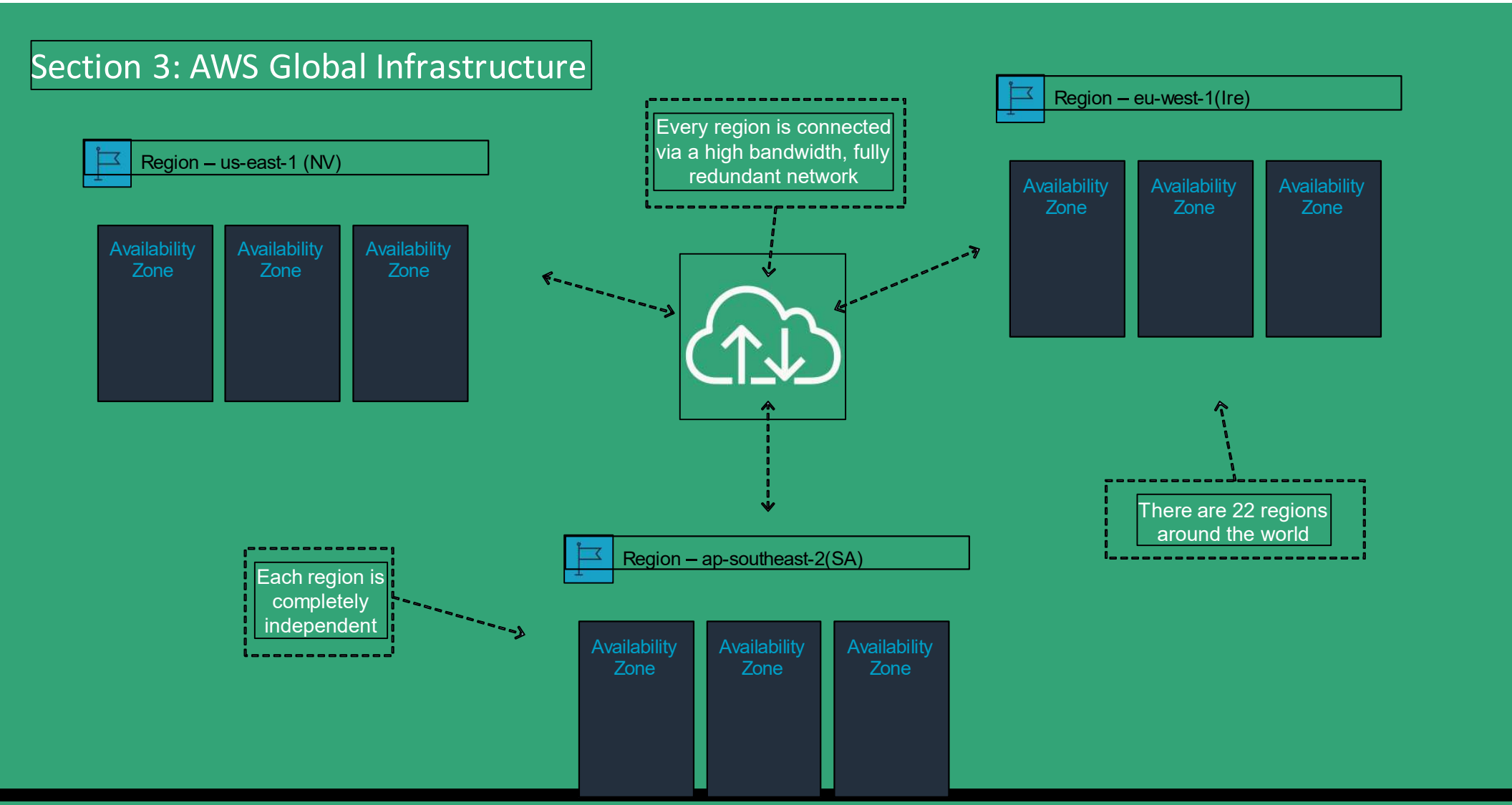
Section 3: AWS Regions

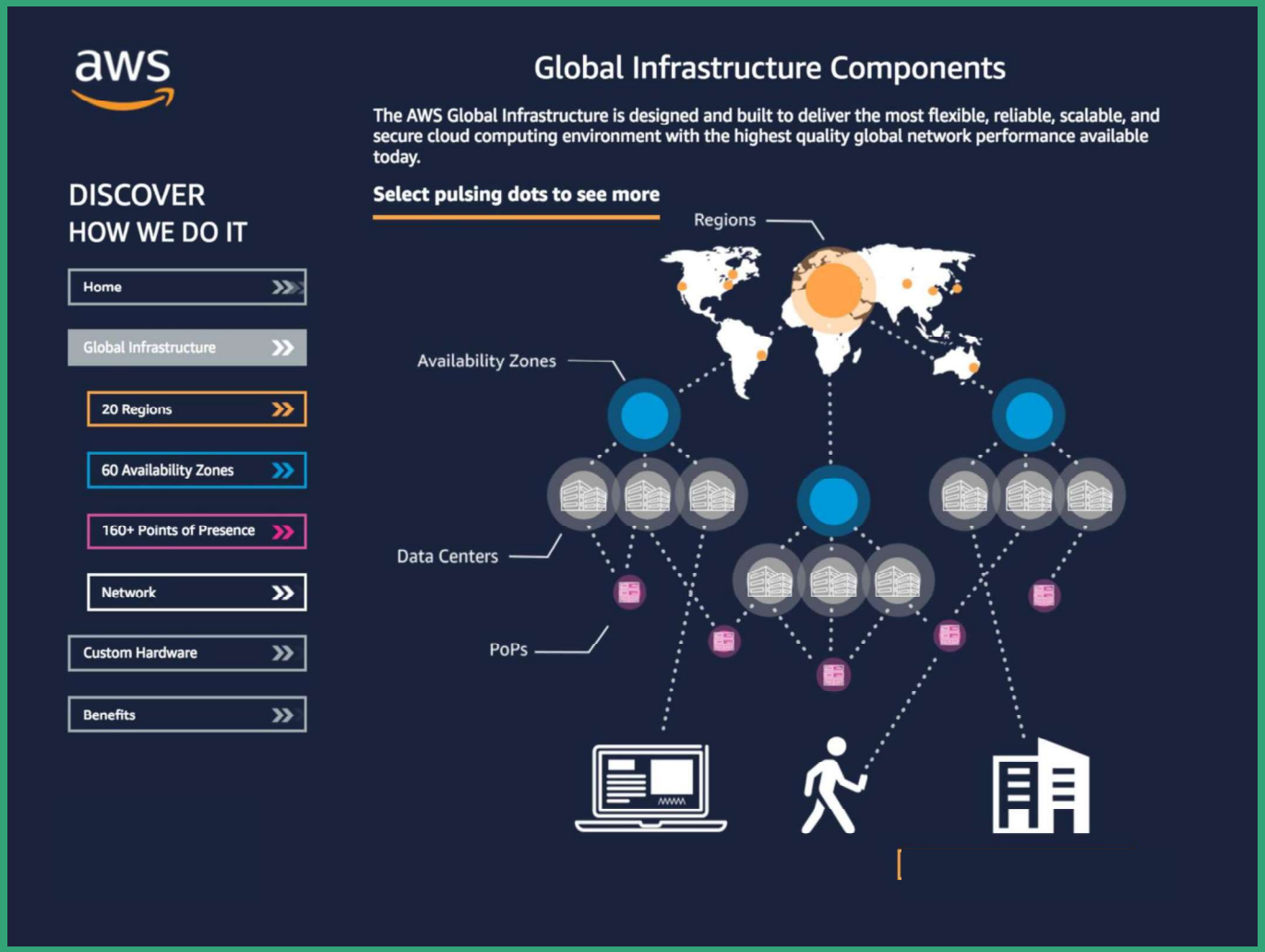
- An AWS region is a geographical area
- Each region consists of 2 or more availability zones
- Each Amazon Region is designed to be completely isolated from the other Amazon Regions

Section 3: AWS Availability Zones

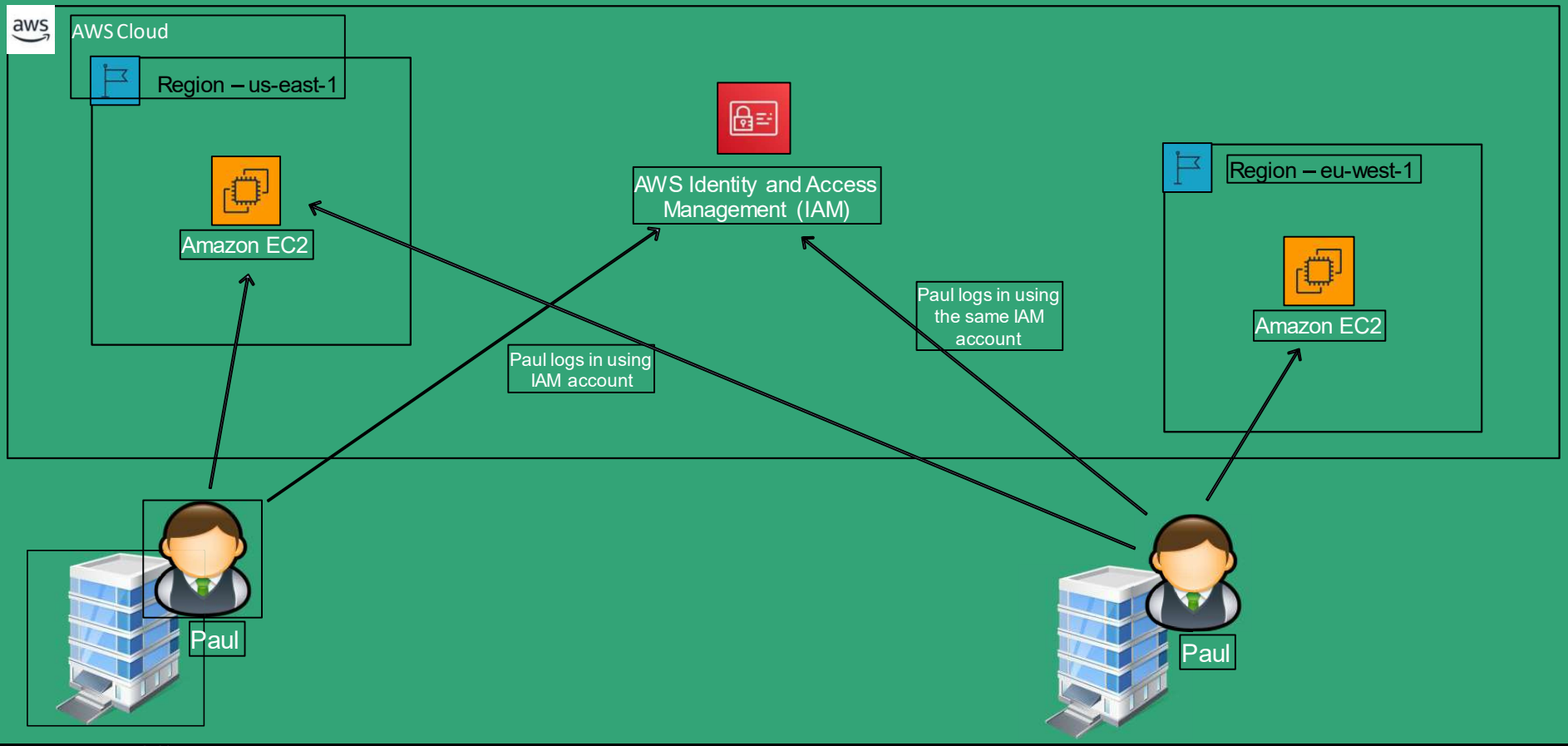
- Availability Zones (AZs) are locations into which you launch resources, such as Amazon EC2 instances
- AZs physically separate and isolated from each other
- AZs span one or more data centers and have direct, low-latency, high throughput and redundant network connections between each other
- Each AZ is designed as an independent failure zone
- AZs are physically separated within a typical metropolitan region, and use discrete power sources

Section 3: AWS Global Infrastructure

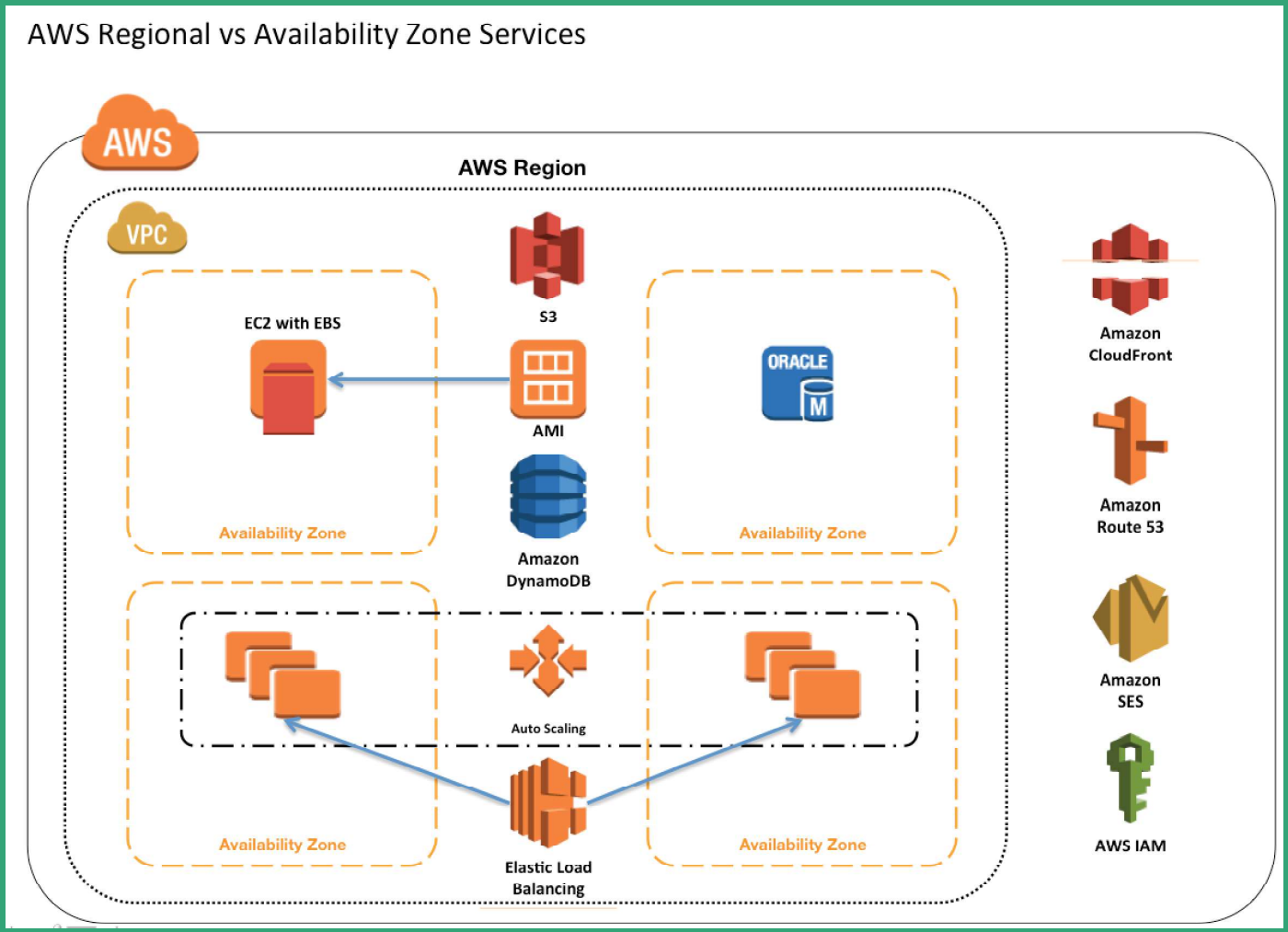




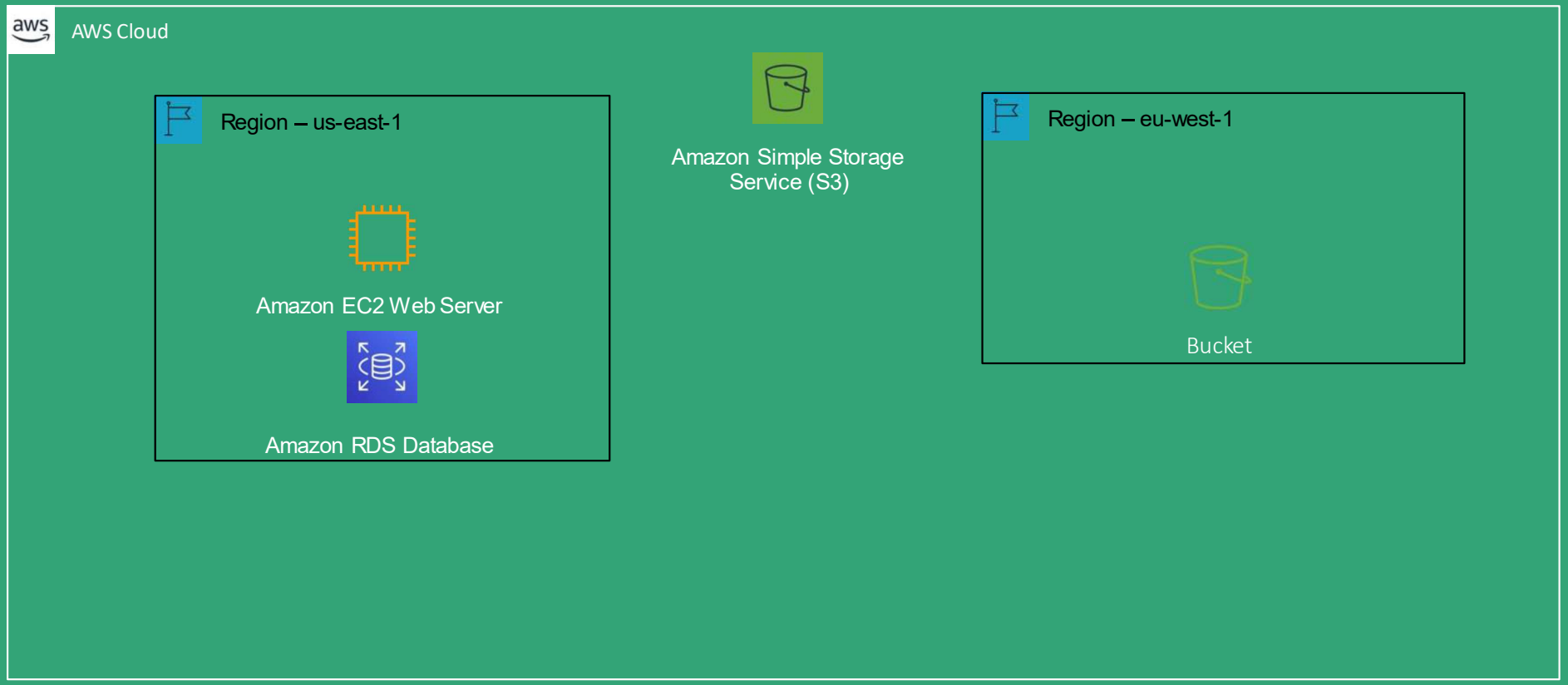
Section 3: Global Service Example - IAM



AWS Regional vs Availability Zone Services

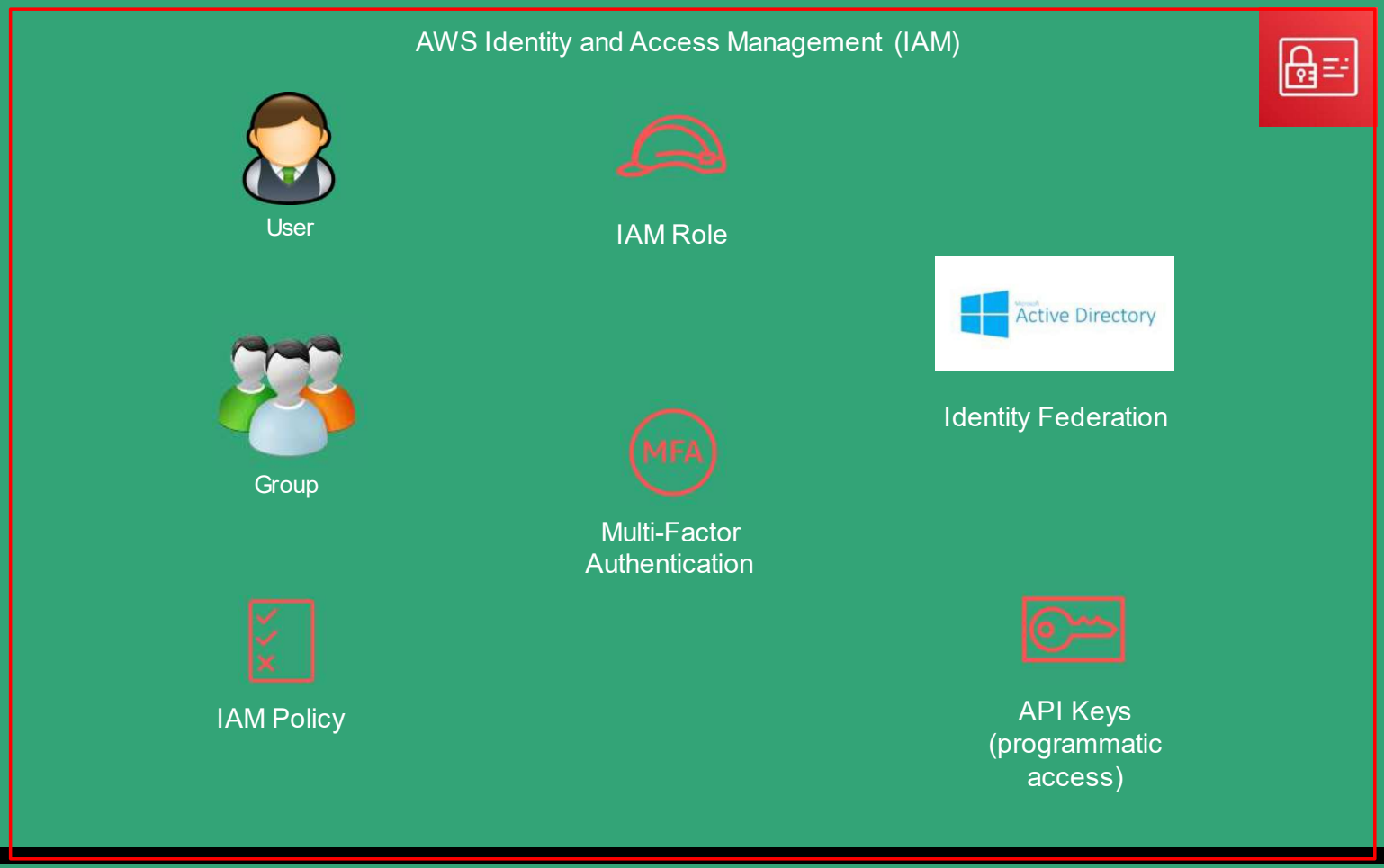


Section 3: Regional Service Examples

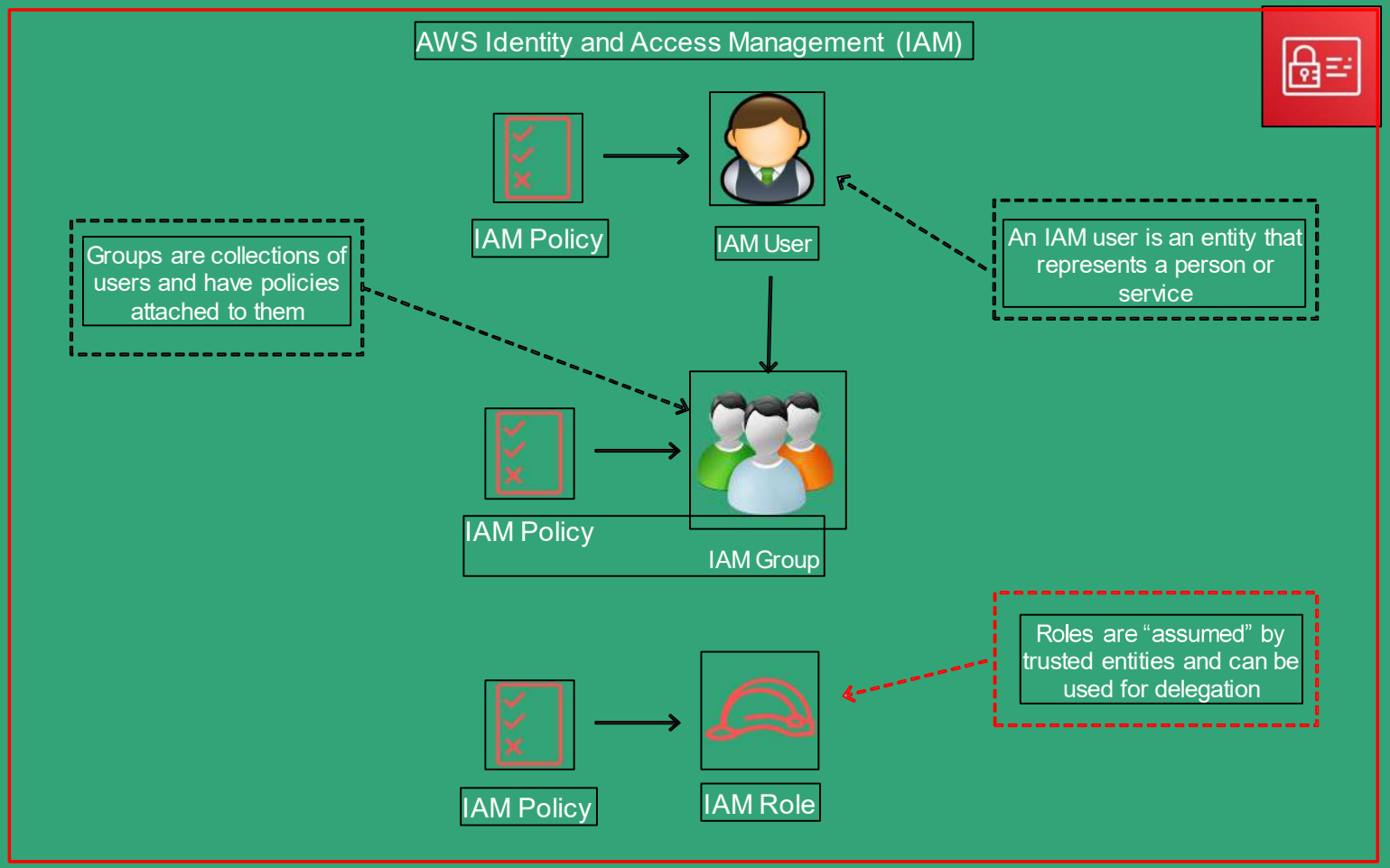


Identity and Access Management (IAM)

Section 4: Identity and Access Management (IAM) Overview



Section 4: IAM Users, Groups, Roles and Policies



Section 4: IAM Users

- An IAM user is an entity that represents a person or service
- Can be assigned:
  - An access key ID and secret access key for programmatic access to the AWS API, CLI, SDK, and other development tools
  - A password for access to the management console
- By default users cannot access anything in your account
- The account root user credentials are the email address used to create the account and a password
- The root account has full administrative permissions and these cannot be restricted
- Best practice for root accounts:
  - Don't use the root user credentials
  - Don't share the root user credentials
  - Create an IAM user and assign administrative permissions as required
  - Enable Multi-Factor Authentication (MFA)

Eric

Ethan

Andrea

Section 4: IAM Users

- IAM users can be created to represent applications and these are known as “service accounts”
- You can have up to 5000 users per AWS account
- Each user account has a friendly name and an Amazon Resource Name (ARN) which uniquely identifies the user across AWS
- You should create individual IAM accounts for users (best practice not to share accounts)
- A password policy can be defined for enforcing password length, complexity etc. (applies to all users)

Eric

Ethan

Andrea

Section 4: IAM Groups

- Groups are collections of users and have policies attached to them
- A group is not an identity and cannot be identified as a principal in an IAM policy
- Use groups to assign permissions to users
- Use the principal of least privilege when assigning permissions
- You cannot nest groups (groups within groups)

Developers

AWS Admins

Operations

Section 4: IAM Roles

- Roles are created and then “assumed” by trusted entities and define a set of permissions for making AWS service requests
- With IAM Roles you can delegate permissions to resources for users and services without using permanent credentials (e.g. user name and password)
- IAM users or AWS services can assume a role to obtain temporary security credentials that can be used to make AWS API calls
- You can delegate using roles
- There are no credentials associated with a role (password or access keys)

S3 Full Access

DynamoDB Read-Only

AWSLambdaBasicExecutionRole

Section 4: IAM Policies

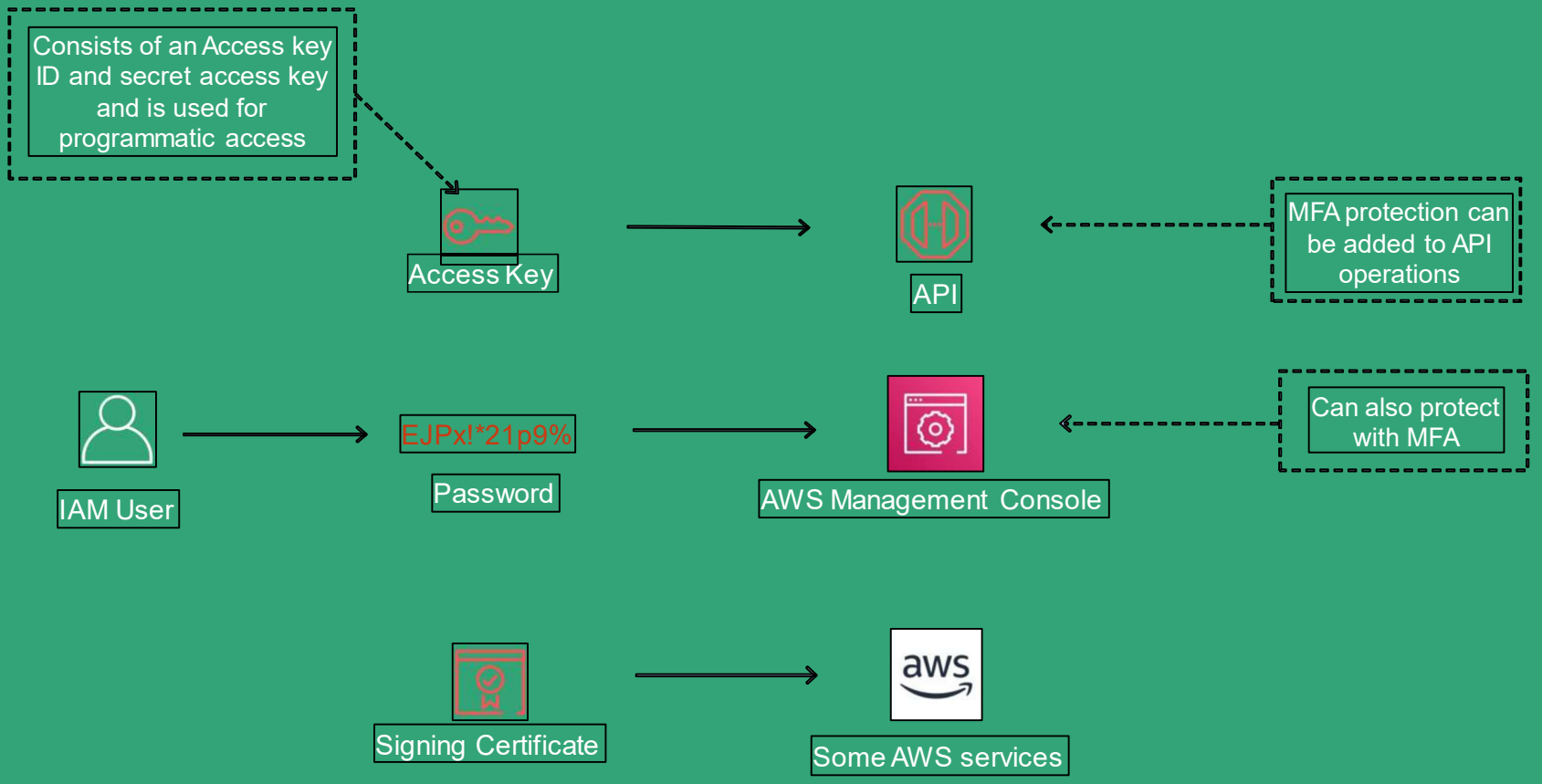
- Policies are documents that define permissions and can be applied to users, groups and roles
- Policy documents are written in JSON (key value pair that consists of an attribute and a value)
- All permissions are implicitly denied by default
- The most restrictive policy is applied
- The IAM policy simulator is a tool to help you understand, test, and validate the effects of access control policies
- The Condition element can be used to apply further conditional logic

S3 Full Access

DynamoDB Read-Only

AWSLambdaBasicExecutionRole

Section 4: Authentication Methods



Section 4: IAM Access Keys

- A combination of an access key ID and a secret access key

These can be used to make programmatic calls to AWS when using the API in program code or at a command prompt when using the AWS CLI or the AWS PowerShell tools

You can create, modify, view or rotate access keys

When created IAM returns the access key ID and secret access key

The secret access is returned only at creation time and if lost a new key must be created

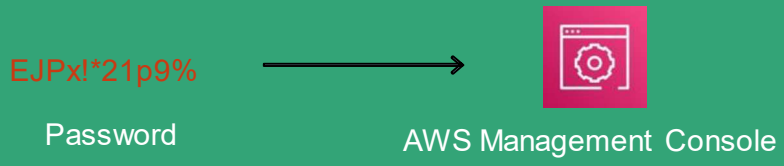
Ensure access keys and secret access keys are stored securely

Users can be given access to change their own keys through IAM policy (not from the console)

You can disable a user's access key which prevents it from being used for API calls

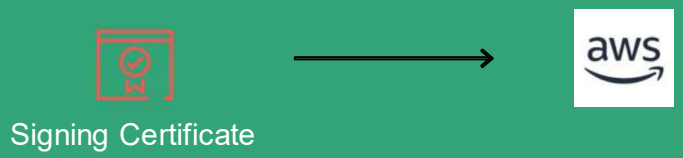
Section 4: IAM Console Password

- A password that the user can enter to sign into interactive sessions such as the AWS Management Console
- You can allow users to change their own passwords
- You can allow selected IAM users to change their passwords by disabling the option for all users and using an IAM policy to grant permissions for the selected users



Section 4: IAM Server Certificate / Signing Certificate

- SSL/TLS certificates that you can use to authenticate with some AWS services
- AWS recommends that you use the AWS Certificate Manager (ACM) to provision, manage and deploy your server certificates
- Use IAM only when you must support HTTPS connections in a region that is not supported by ACM



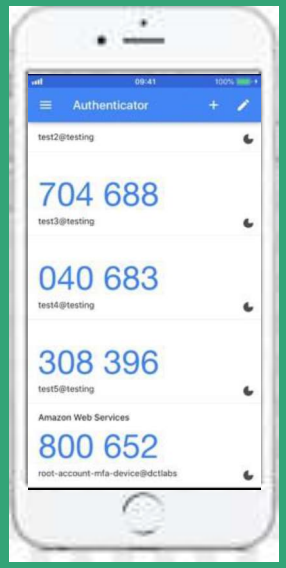
Section 4: Multi-Factor Authentication

Something you know:

EJPx!*21p9%

Password

Something you have:



Something you are:

Section 4: Multi-Factor Authentication in AWS

Something you know:

IAM User

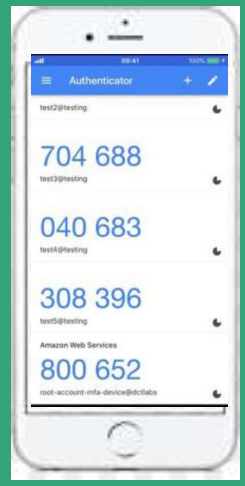
EJPx!*21p9%

Password

Something you have:

MFA Virtual MFA

e.g. Google Authenticator on your smart phone



MFA Physical MFA

Section 4: AWS Security Token Service (STS)

- The AWS Security Token Service (STS) is a web service that enables you to request temporary, limited-privilege credentials for IAM users or for users that you authenticate (federated users)
- By default, AWS STS is available as a global service, and all AWS STS requests go to a single endpoint at https://sts.amazonaws.com
- All regions are enabled for STS by default but can be disabled
- The region in which temporary credentials are requested must be enabled
- Credentials will always work globally

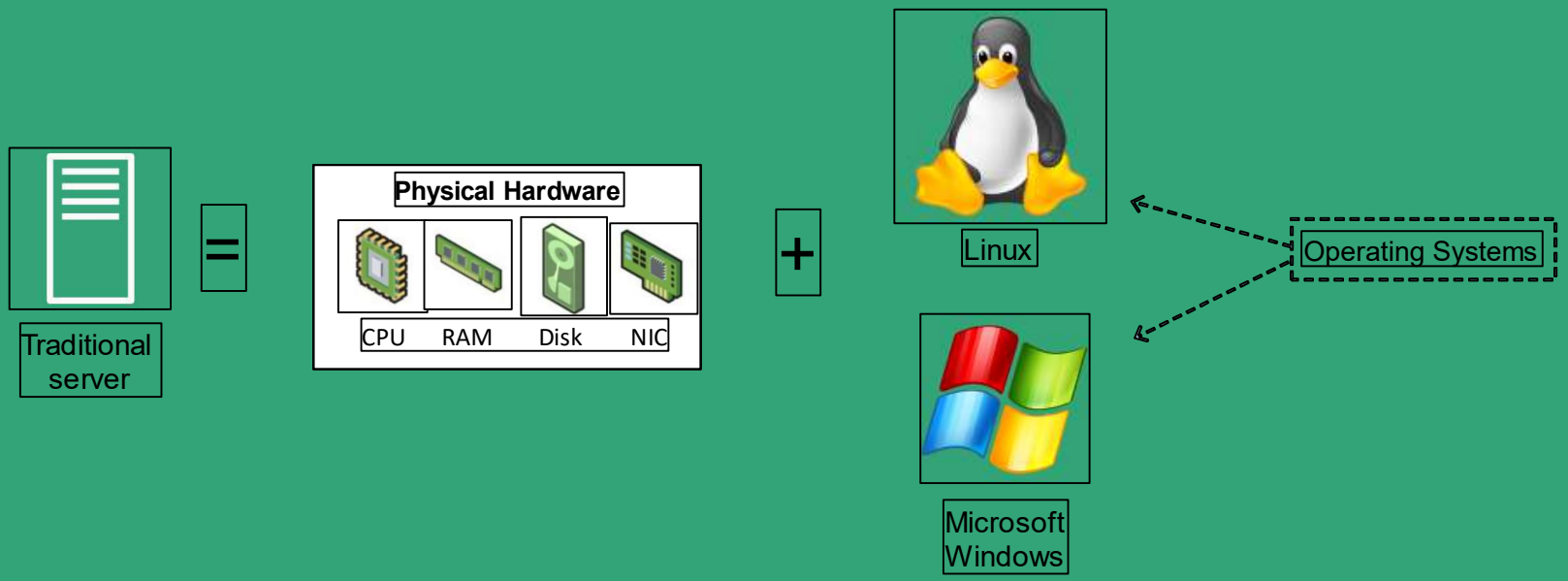
Temporary security credential

Section 4: IAM Best Practices

- Lock away the AWS root user access keys
- Create individual IAM users
- Use AWS defined policies to assign permissions whenever possible
- Use groups to assign permissions to IAM users
- Grant least privilege
- Use access levels to review IAM permissions
- Configure a strong password policy for users
- Enable MFA
- Use roles for applications that run on AWS EC2 instances
- Delegate by using roles instead of sharing credentials
- Rotate credentials regularly
- Remove unnecessary credentials
- Use policy conditions for extra security
- Monitor activity in your AWS account

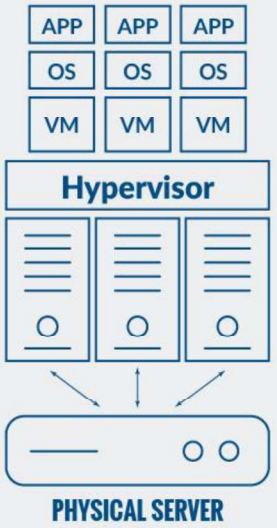
AWS Compute

Section 6: Traditional Servers



What is server virtualization?

Server Virtualization is a technology which enables the creation of a virtual instance of any operating system on a virtual platform. Before server virtualization became mainstream, each individual operating system required a physical platform, typically a server with CPU, Disk, Memory and other associated hardware to host the operating system. On a physical server, an operating system has access to all available computing resources of the host hardware. As servers have become more powerful, this approach has proven very wasteful on hardware resource availability.



While server virtualization has recently exploded in popularity, the technology of virtualization has been in development for well over 50 years

1960s

During the 1960s, IBM pioneered the first virtualization of system memory; this was the precursor to virtual hardware.

1970s

In the 1970s, IBM virtualized a proprietary operating system for the first time called VM/370

1990s

The popularity of virtualization increased greatly in the late 1990s, with VMware's release of VMware workstation

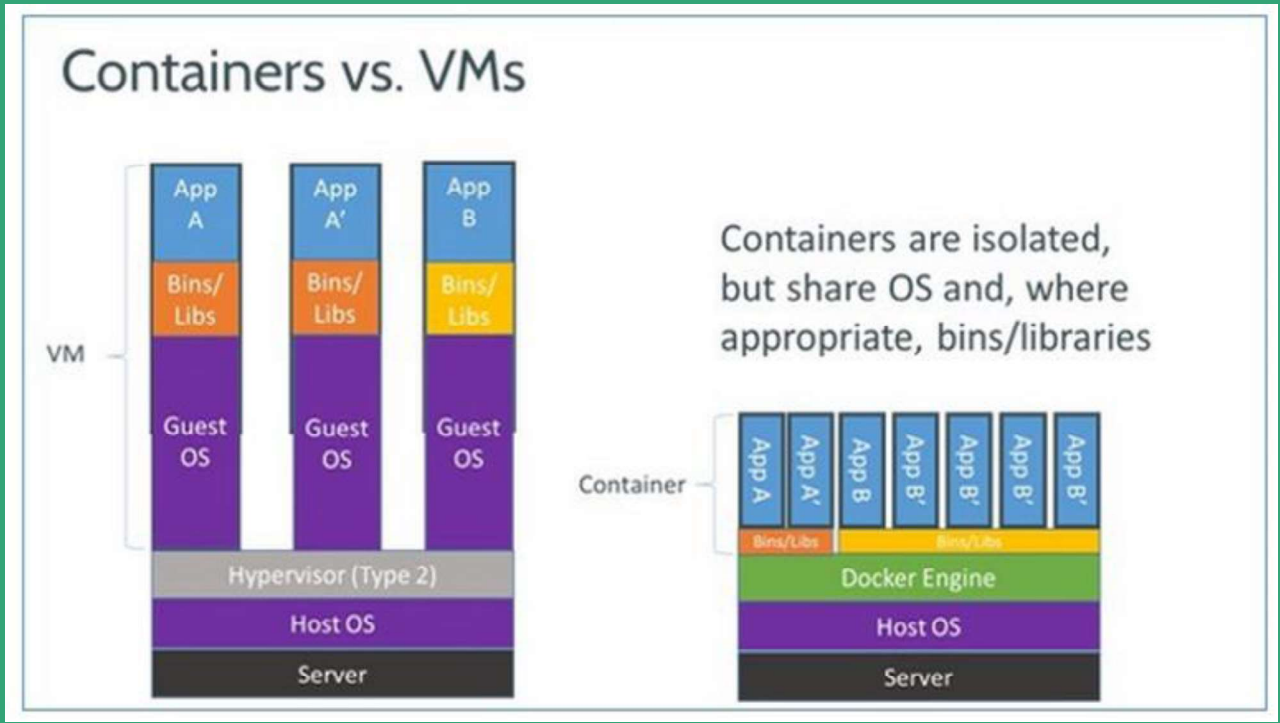
Virtualization

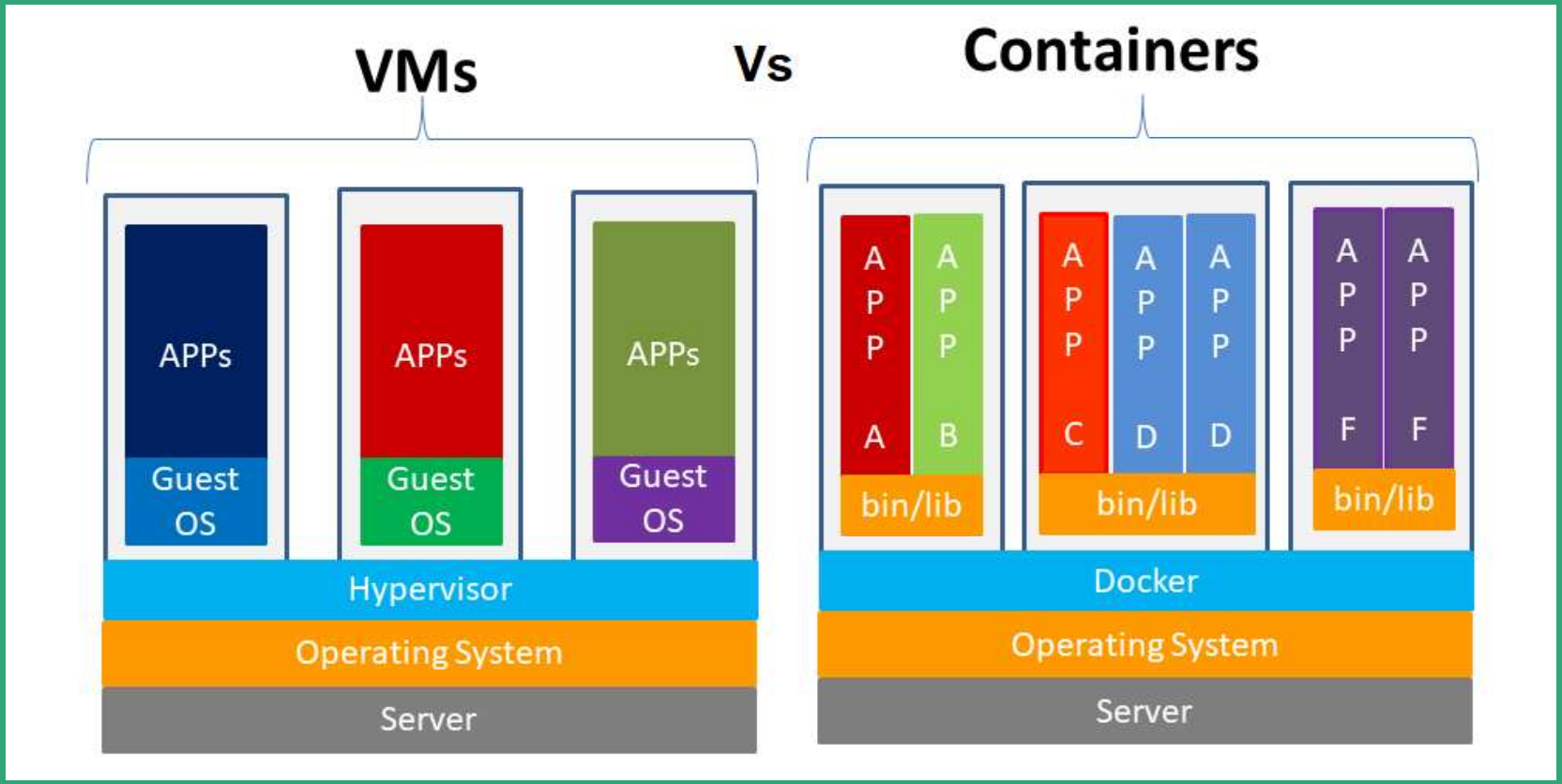
Virtualization is the process of running a virtual instance of a computer system in a layer abstracted from the actual hardware. Most commonly, it refers to running multiple operating systems on a computer system simultaneously. To the applications running on top of the virtualized machine, it can appear as if they are on their own dedicated machine, where the operating system, libraries, and other programs are unique to the guest virtualized system and unconnected to the host operating system which sits below it.

Types

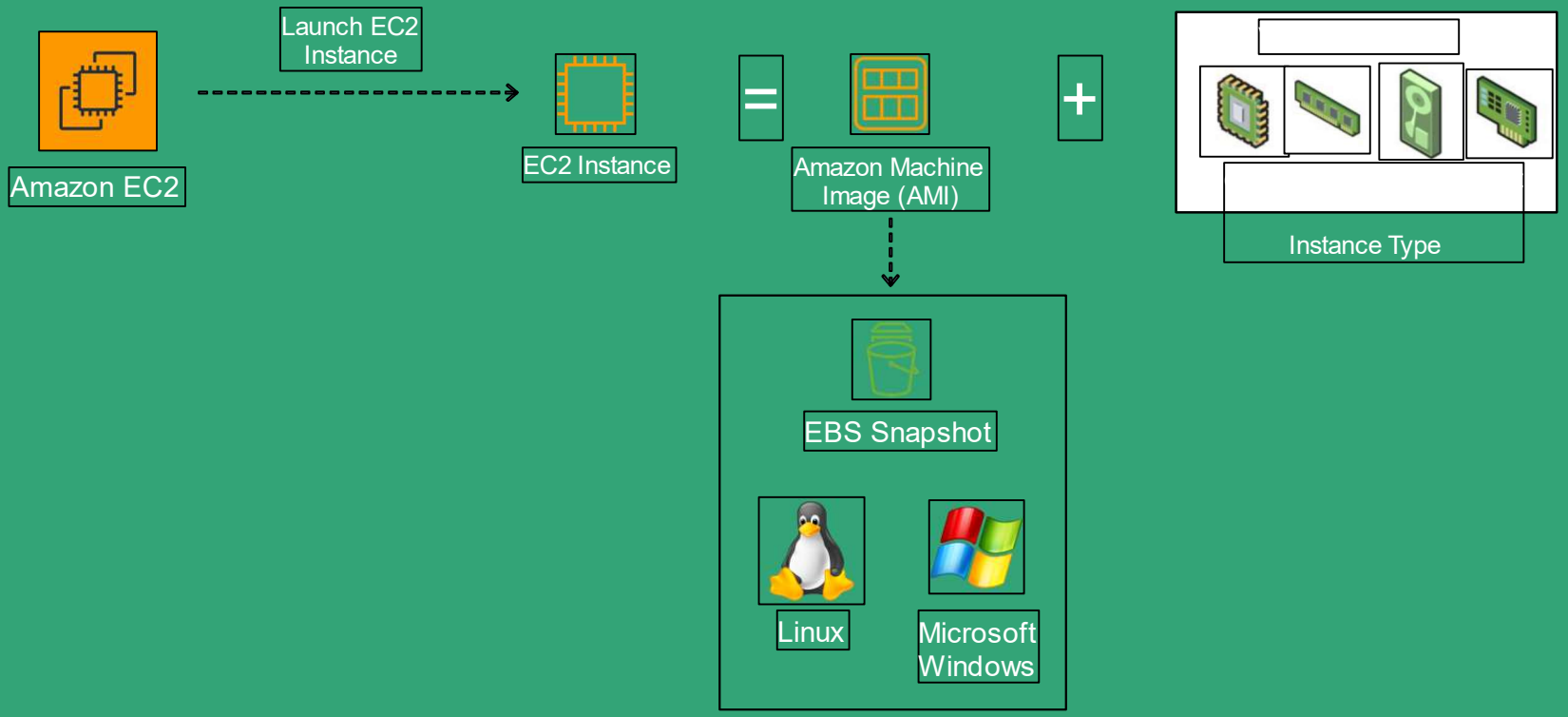
- Desktop Virtualization
- Application Virtualization
- Server Virtualization
- Storage Virtualization
- Network Virtualization

Virtual Machines VS Containers

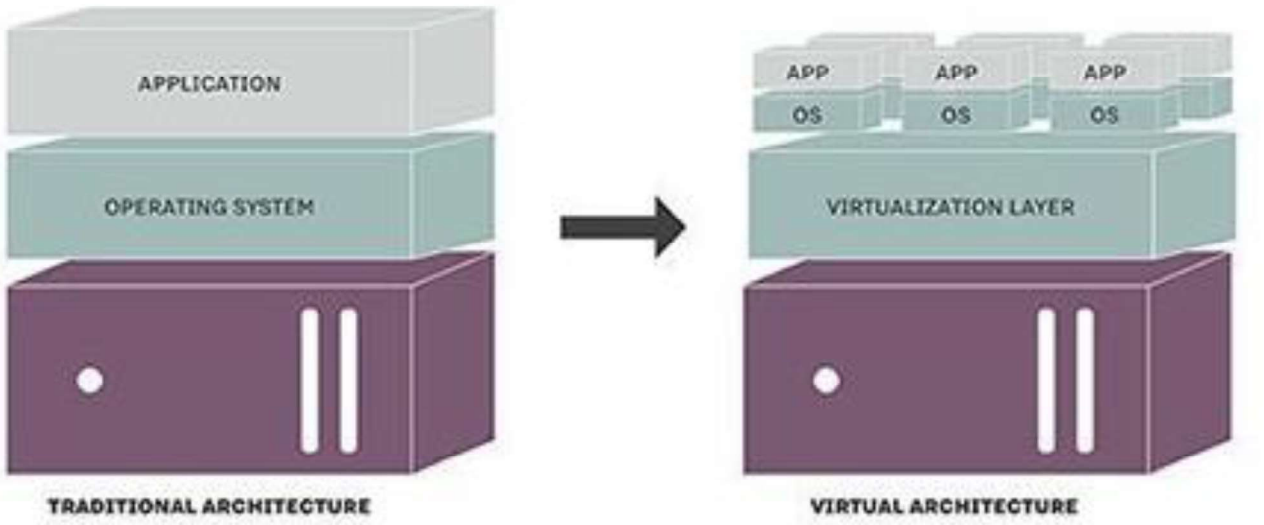


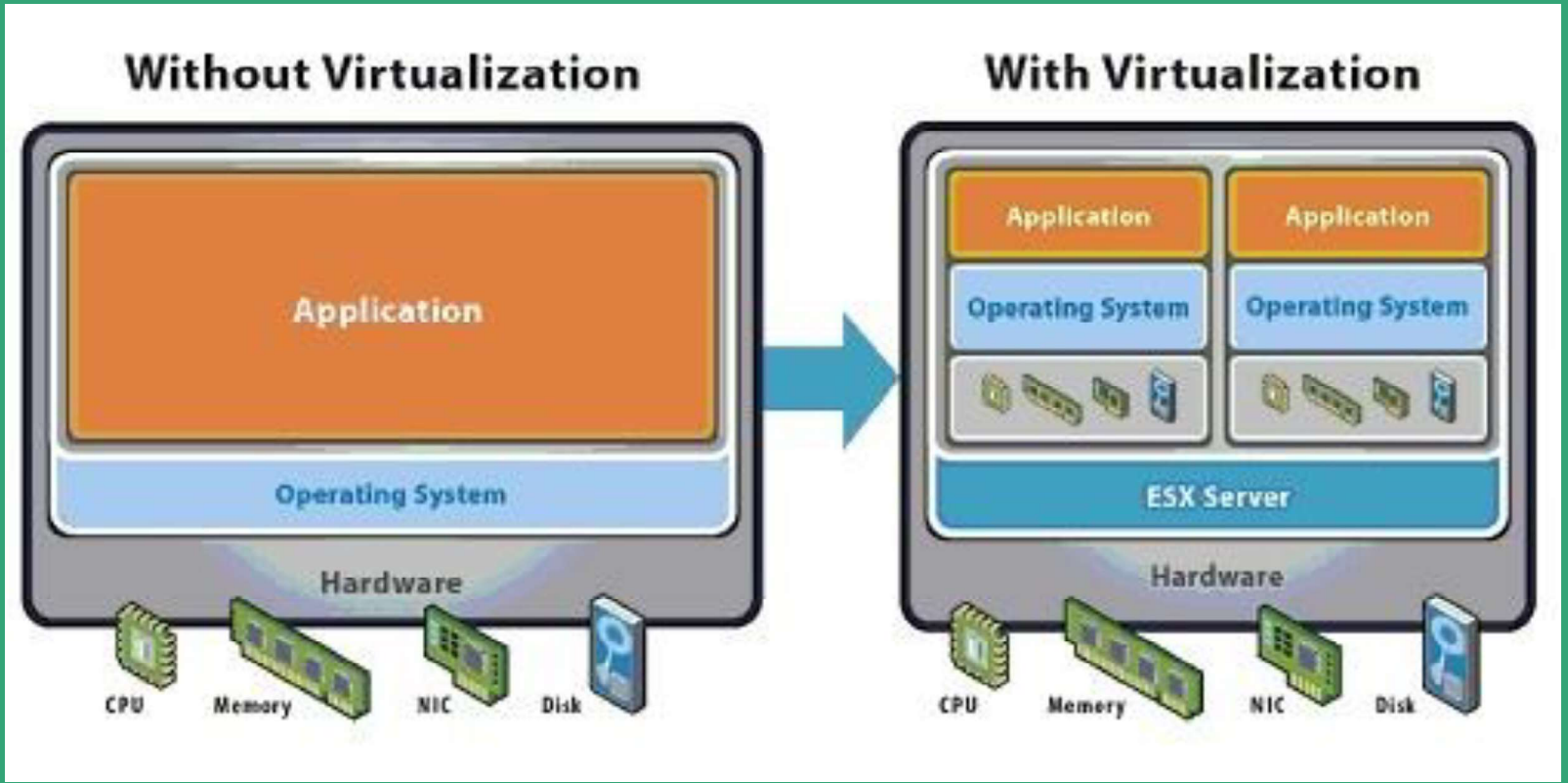


Section 6: Amazon Elastic Compute Cloud (EC2)



TRADITIONAL AND VIRTUAL ARCHITECTURE





Section 6: Amazon EC2 Overview

Amazon EC2

- Amazon Elastic Compute Cloud (Amazon EC2) is a web service in the AWS Compute suite of products that provides secure, resizable compute capacity in the cloud
- Elastic Web-Scale computing – you can increase or decrease capacity within minutes and commission one to thousands of instances simultaneously
- Completely controlled – You have complete control include root access to each instance and can stop and start instances without losing data and using web service APIs
- Flexible Cloud Hosting Services – you can choose from multiple instance types, operating systems, and software packages as well as instances with varying memory, CPU and storage configurations

Section 6: Amazon Machine Images (AMI)

Amazon EC2

- An Amazon Machine Image (AMI) provides the information required to launch an instance
- An AMI includes the following:
  - One or more EBS snapshots, or, for instance-store-backed AMIs, a template for the root volume of the instance (for example, an operating system, an application server, and applications).
  - Launch permissions that control which AWS accounts can use the AMI to launch instances.
  - A block device mapping that specifies the volumes to attach to the instance when it's launched
- AMIs come in three main categories:
  - Community AMIs - free to use, generally you just select the operating system you want
  - AWS Marketplace AMIs - pay to use, generally come packaged with additional, licensed software
  - My AMIs - AMIs that you create yourself

Section 6: Amazon EC2 Instance Types

Amazon EC2

Category	Families	Purpose/Design
General Purpose	A1, T3, T3a, T2, M5, M5a, M4	General purpose instances provide a balance of compute, memory and networking resources, and can be used for a variety of diverse workloads
Compute Optimized	C5, C5n, C4	Compute Optimized instances are ideal for compute bound applications that benefit from high performance processors
Memory Optimized	R5, R5a, R4, X1e, X1, High Memory, z1d	Memory optimized instances are designed to deliver fast performance for workloads that process large data sets in memory
Accelerated Computing	P3, P2, G4, G3, F1	Accelerated computing instances use hardware accelerators, or co-processors, to perform functions, such as floating-point number calculations, graphics processing, or data pattern matching
Storage Optimized	I3, I3en, D2, H1	This instance family provides Non-Volatile Memory Express (NVMe) SSD-backed instance storage optimized for low latency, very high random I/O performance, high sequential read throughput and provide high IOPS at a low cost

Section 6: Instance User Data and Instance Metadata

Amazon EC2

User Data

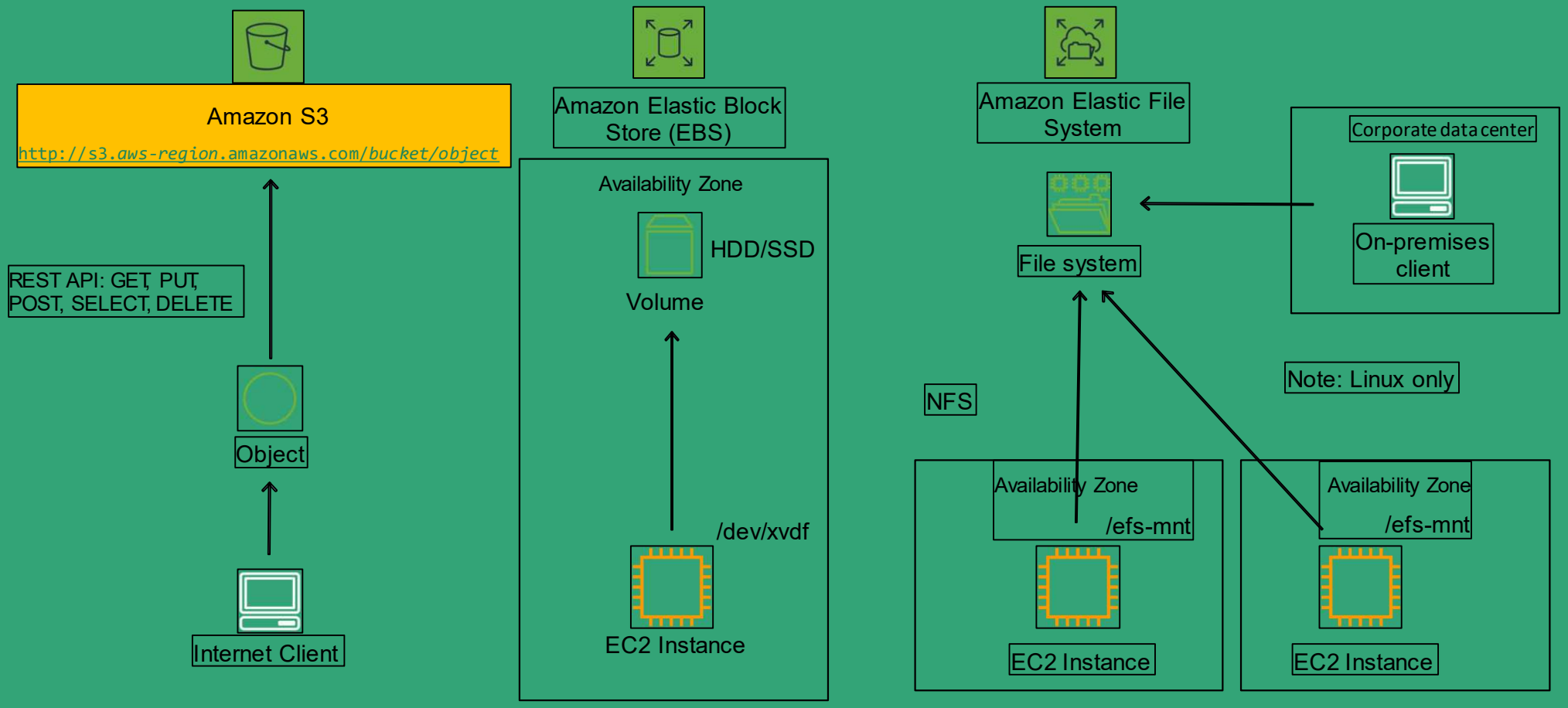
- User data is data that is supplied by the user at instance launch in the form of a script
- User data is limited to 16KB
- User data and metadata are not encrypted

Metadata

- Instance metadata is data about your instance that you can use to configure or manage the running instance
- Instance metadata is available at http://169.254.169.254/latest/meta-data
- The Instance Metadata Query tool allows you to query the instance metadata without having to type out the full URI or category names

Link

Section 7: Object, Block, and File Storage



Section 7: Object, Block, and File Storage Systems

Object Storage

- Object-based storage systems manage data as individual objects, rather than as blocks and sectors (block-based) or a file hierarchy (file-based)
- Object-based storage is accessed using a REST API (URL with HTTP methods, e.g. GET, PUT)
- With object storage data is managed as individual objects rather than a file hierarchy (as with a traditional file system)
- Each object includes the data itself, metadata (data about the data), and a globally unique identifier
- Due to its flat file structure, object storage has virtually unlimited scalability and allows the retention of massive amounts of unstructured data

Section 7: Object, Block, and File Storage Systems

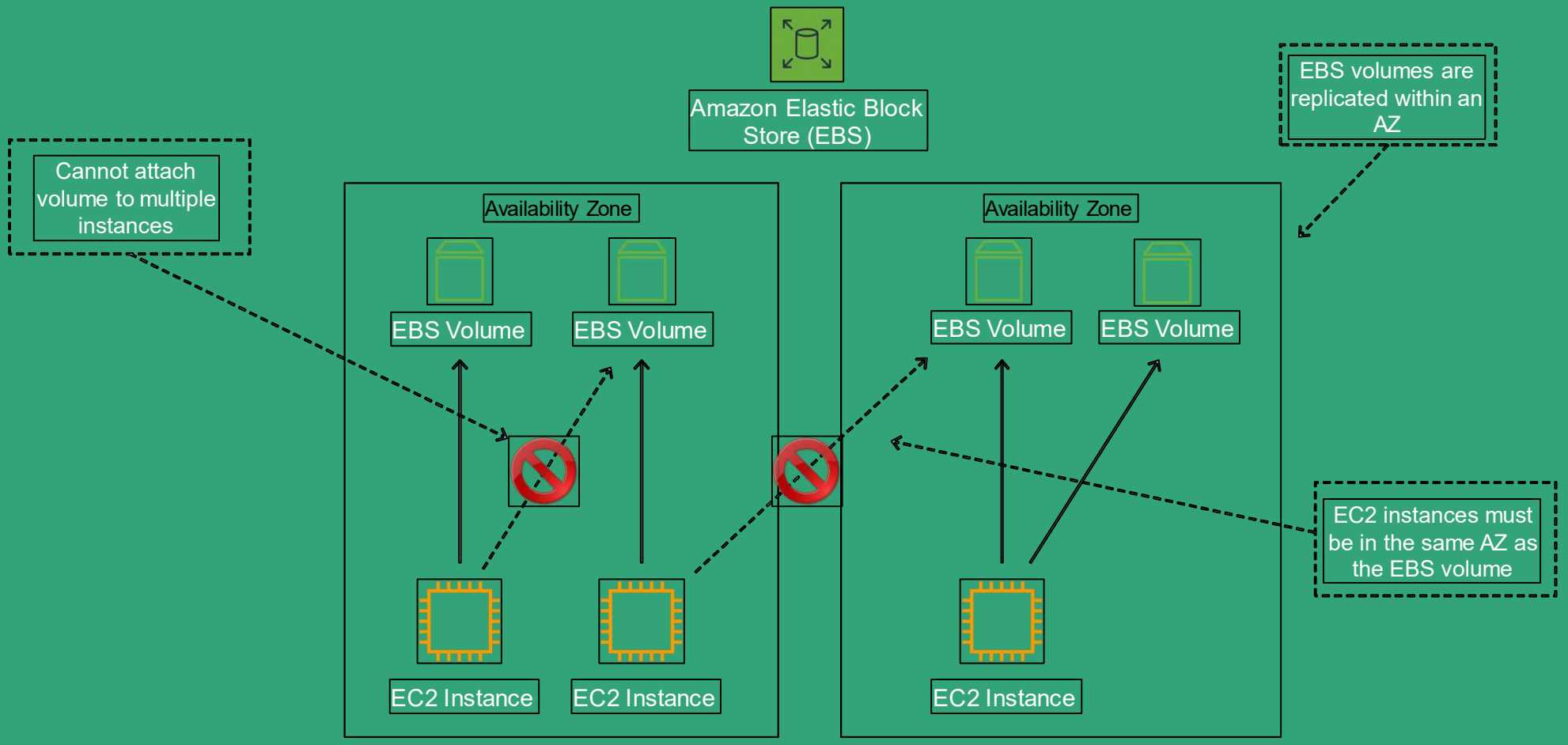
Block Storage

- Data is stored and managed in blocks within sectors and tracks and is controlled by a server-based operating system
- Block storage volumes appear as local disks to the operating system and can be partitioned and formatted
- You can use block storage devices as a boot volume
- Common use cases for block storage are structured information such as file systems, databases, transactional logs, SQL databases and virtual machines (VMs)

File Storage

- File-based storage systems manage data in a file hierarchy
- A file system is mounted via the network to a client computer where it then becomes accessible for reading and writing data
- Protocols used for accessing file systems include NFS or CIFS/SMB

Section 7: Amazon Elastic Block Store (EBS)



Section 7: Amazon Elastic Block Store (EBS)

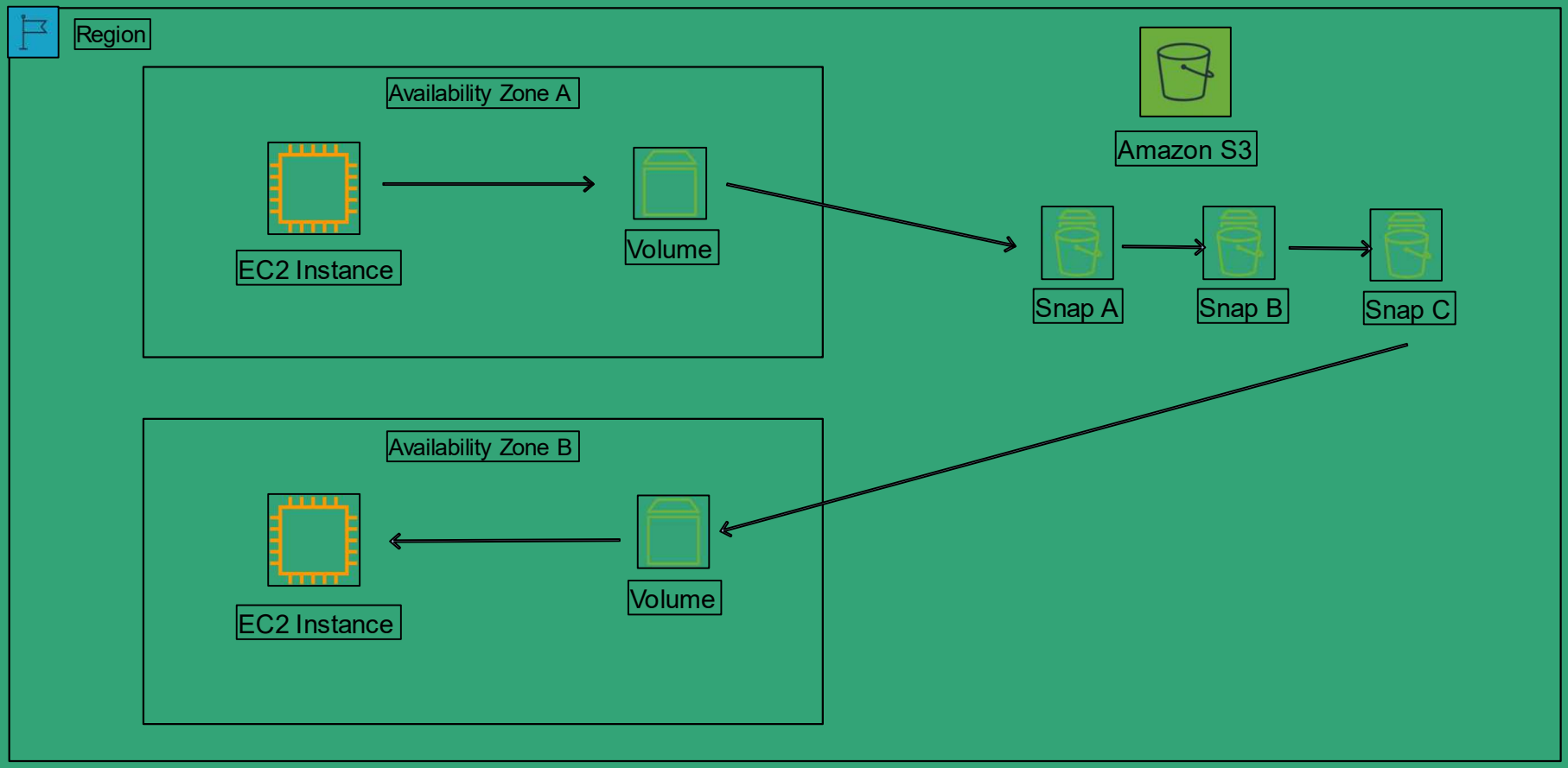
Amazon Elastic Block Store (EBS)

- Amazon Elastic Block Store (Amazon EBS) provides persistent block storage volumes for use with Amazon EC2 instances in the AWS Cloud
- Each Amazon EBS volume is automatically replicated within its Availability Zone to protect you from component failure, offering high availability and durability
- EBS volume data persists independently of the life of the instance
- EBS volumes do not need to be attached to an instance
- You can attach multiple EBS volumes to an instance
- You cannot attach an EBS volume to multiple instances (use Elastic File Store instead)
- EBS volumes must be in the same AZ as the instances they are attached to
- 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

Section 7: Amazon Elastic Block Store (EBS)

	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

Section 7: EBS Snapshots



Section 7: Amazon Elastic Block Store (EBS)

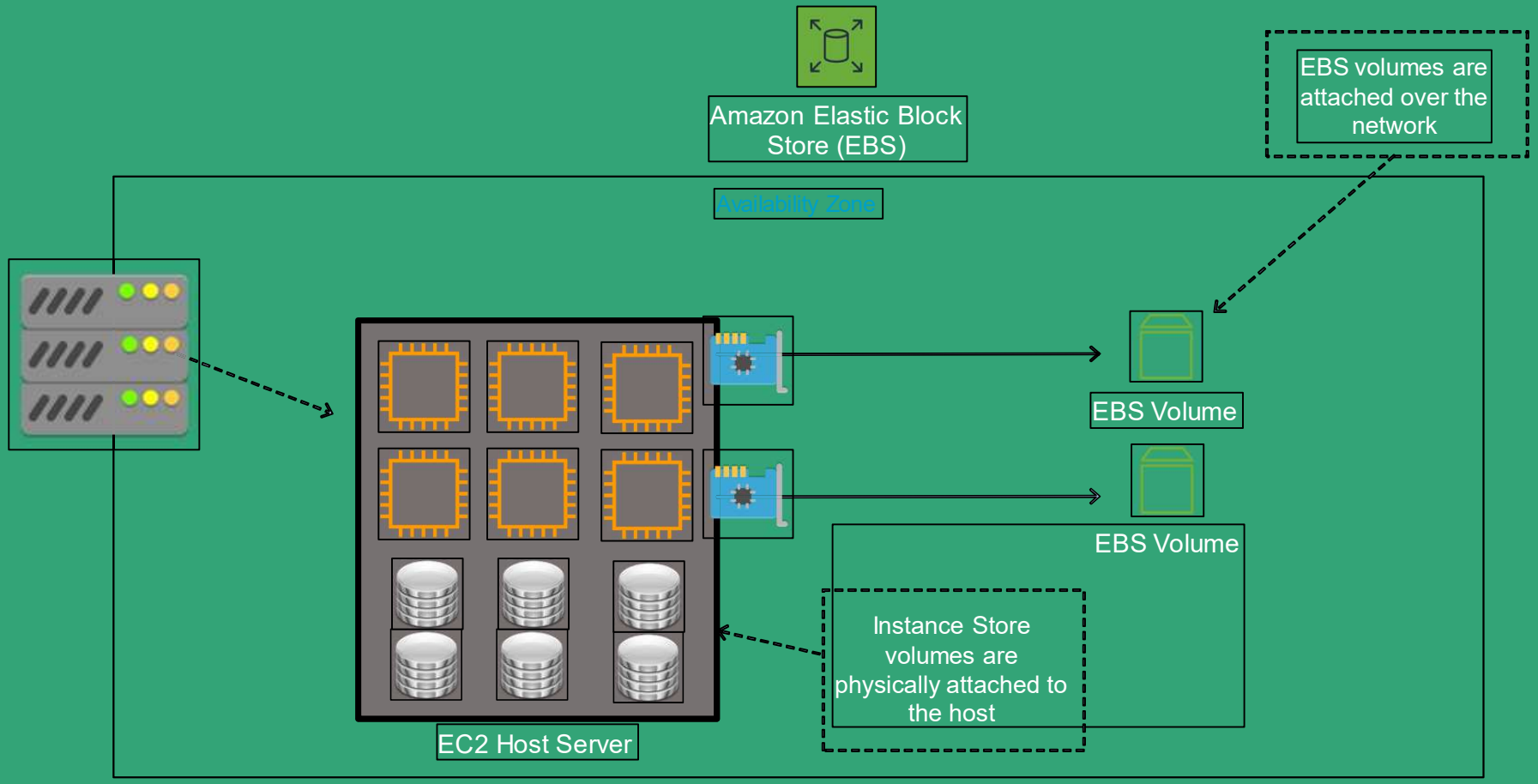
Amazon Elastic Block Store (EBS)

Snapshot

EBS Snapshots

- Snapshots capture a point-in-time state of an instance
- Snapshots are stored on S3
- Does not provide granular backup (not a replacement for backup software)
- If you make periodic snapshots of a volume, the snapshots are incremental, which means that only the blocks on the device that have changed after your last snapshot are saved in the new snapshot
- Even though snapshots are saved incrementally, the snapshot deletion process is designed so that you need to retain only the most recent snapshot in order to restore the volume
- Snapshots can only be accessed through the EC2 APIs
- EBS volumes are AZ specific but snapshots are region specific

Section 7: Amazon Elastic Block Store (EBS) vs Instance Store



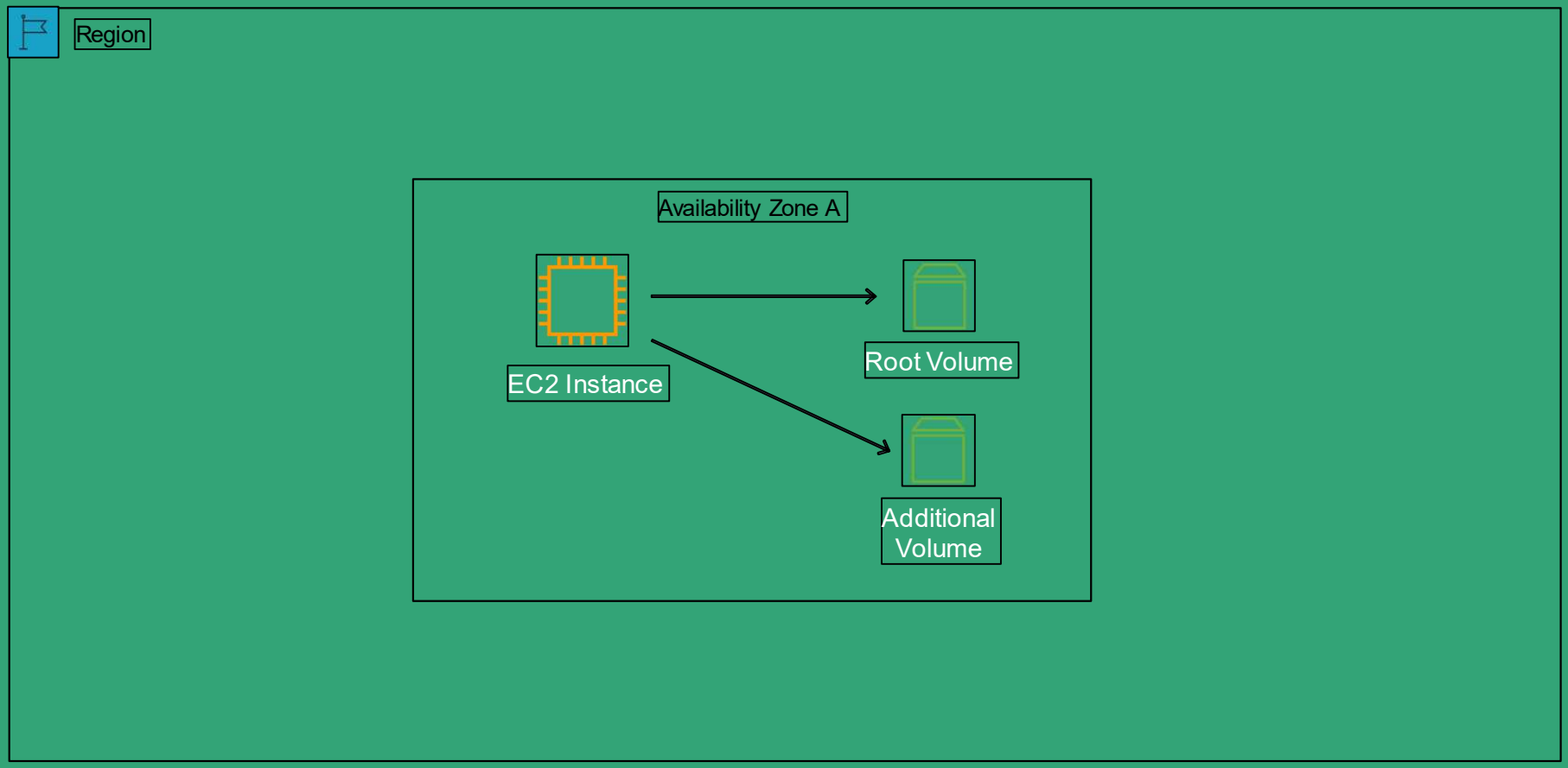
Section 7: Amazon Elastic Block Store (EBS)

Amazon Elastic Block Store (EBS)

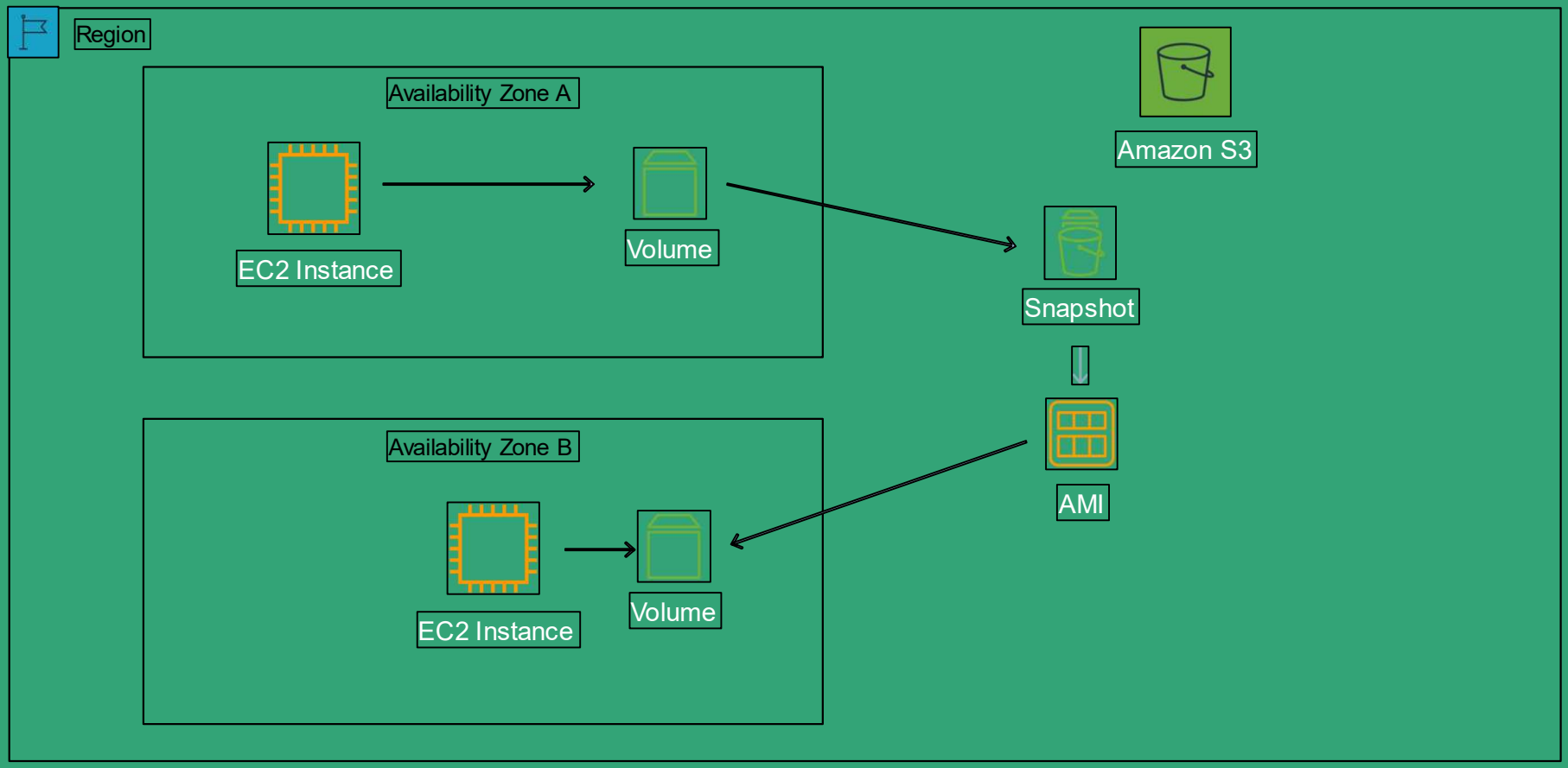
Instance Store Volumes

- 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

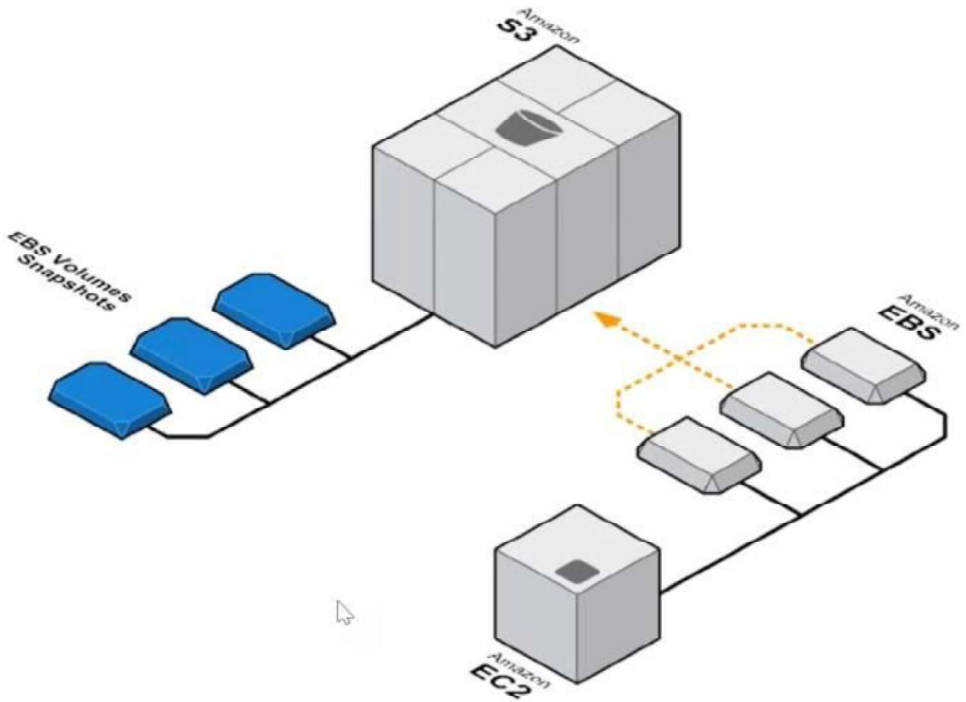
Section 7: Launch Instance, create and add new EBS Volume



Section 7: Take Snapshot, Create AMI, Launch New Instance



EBS: Snapshots



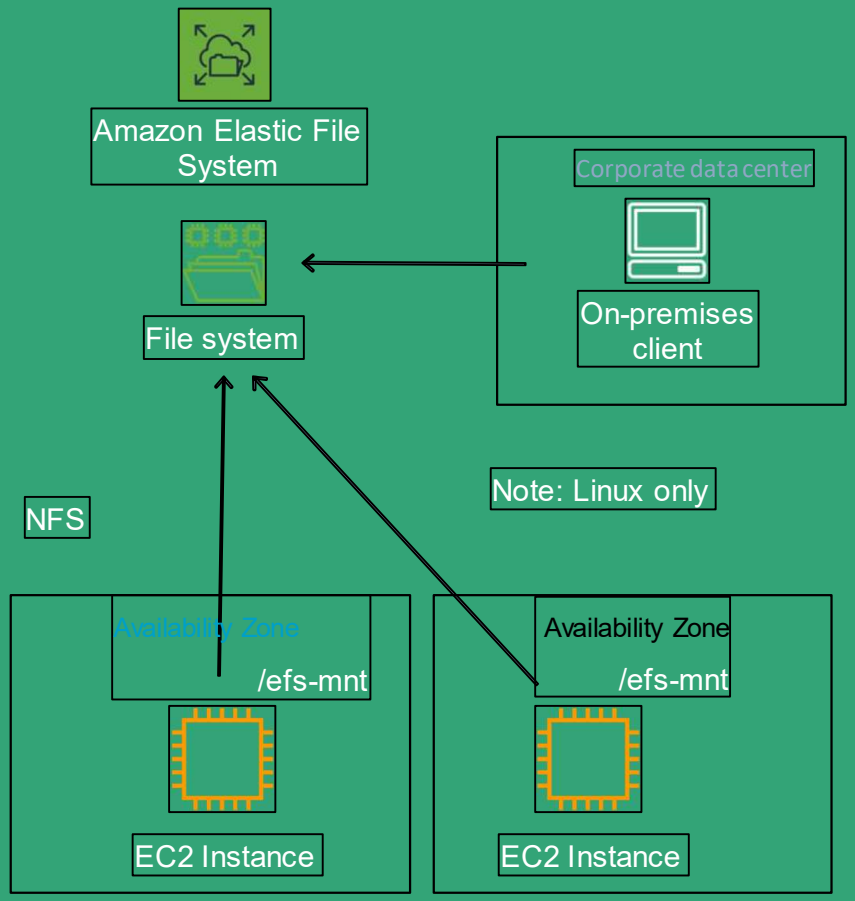
What are EBS Snapshots?

- Point-in-time backup of modified volume blocks
- Stored in S3, accessed via EBS APIs
- First snapshot is a clone, Subsequent snapshots are incremental
- Deleting snapshot will only remove data exclusive to that snapshot

What can you do with a Snapshot?

- Create a new instance from the AMI
- Create a new instance in another AZ or Region

Section 7: Amazon Elastic File System (EFS)



Section 7: Amazon Elastic File System (EFS)

Amazon Elastic File System

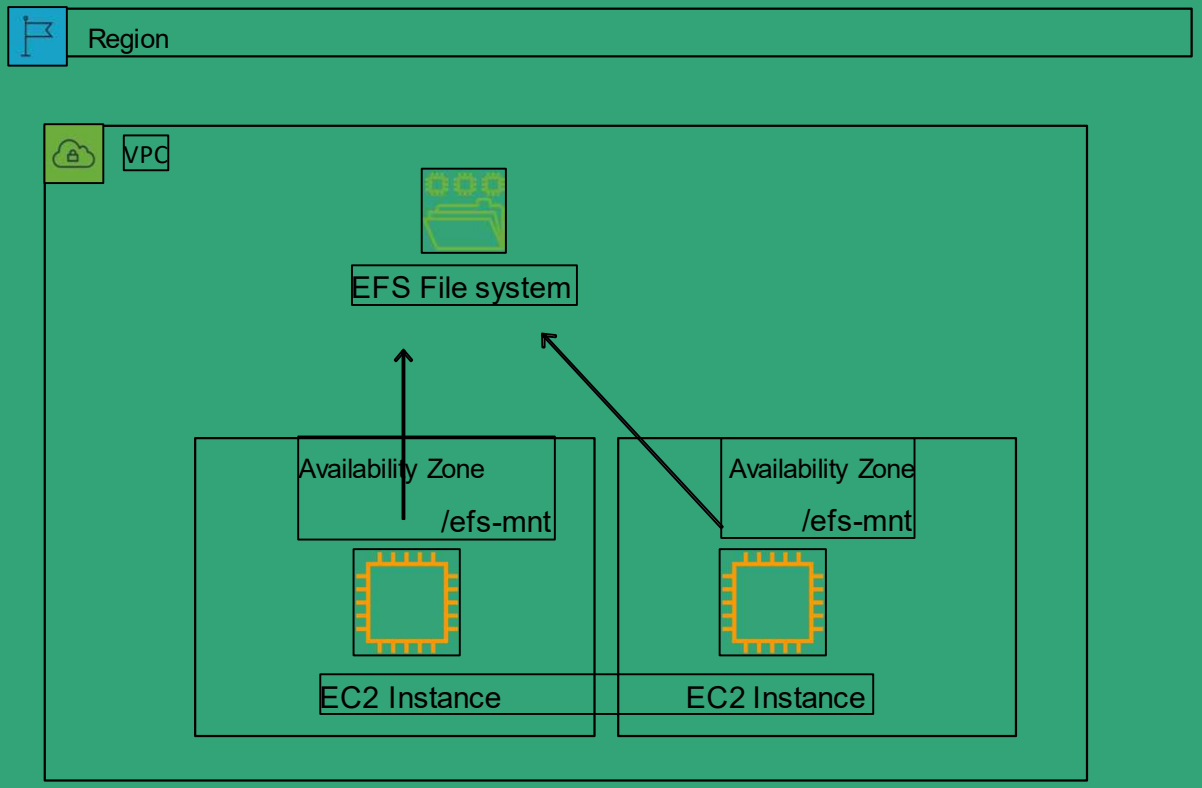
- EFS is a fully-managed service that makes it easy to set up and scale file storage in the Amazon Cloud
- EFS provides a file system interface and uses the NFS protocol
- Good for big data and analytics, media processing workflows, content management, web serving, home directories etc.
- Data is stored across multiple AZ's within a region
- Read after write consistency
- Pay for what you use (no pre-provisioning required)
- Can scale up to petabytes
- EFS is elastic and grows and shrinks as you add and remove data
- Can concurrently connect 1 to 1000s of EC2 instances, from multiple AZs
- A file system can be accessed concurrently from all AZs in the region where it is located
- By default you can create up to 10 file systems per account
- On-premises access can be enabled via Direct Connect or AWS VPN

Section 7: Amazon Elastic File System (EFS)

Amazon Elastic File System

- Instances can be behind an Amazon Elastic Load Balancer
- There are two performance modes:
  - "General Purpose" performance mode is appropriate for most file systems
  - "Max I/O" performance mode is optimized for applications where tens, hundreds, or thousands of EC2 instances are accessing the file system

Amazon Elastic File System (EFS)



Section 6: Amazon Data Life Cycle Manager

With Amazon Data Lifecycle Manager, you can manage the lifecycle of your AWS resources. You create lifecycle policies, which are used to automate operations on the specified resources. Amazon DLM supports Amazon EBS volumes and snapshots.

When you automate snapshot and AMI management, it helps you to:

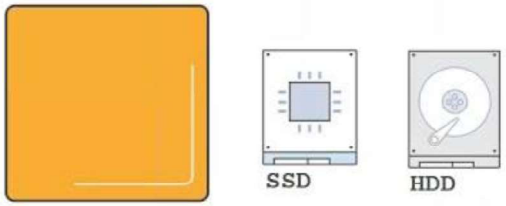
- Protect valuable data by enforcing a regular backup schedule.
- Create standardized AMIs that can be refreshed at regular intervals.
- Retain backups as required by auditors or internal compliance.
- Reduce storage costs by deleting outdated backups.
- Create disaster recovery backup policies that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

LINK

EC2 Instance Store *vs* EBS

EC2 Instance Store

- Local to instance
- Non-persistent data store
- Data not replicated (by default)
- No snapshot support
- SSD or HDD

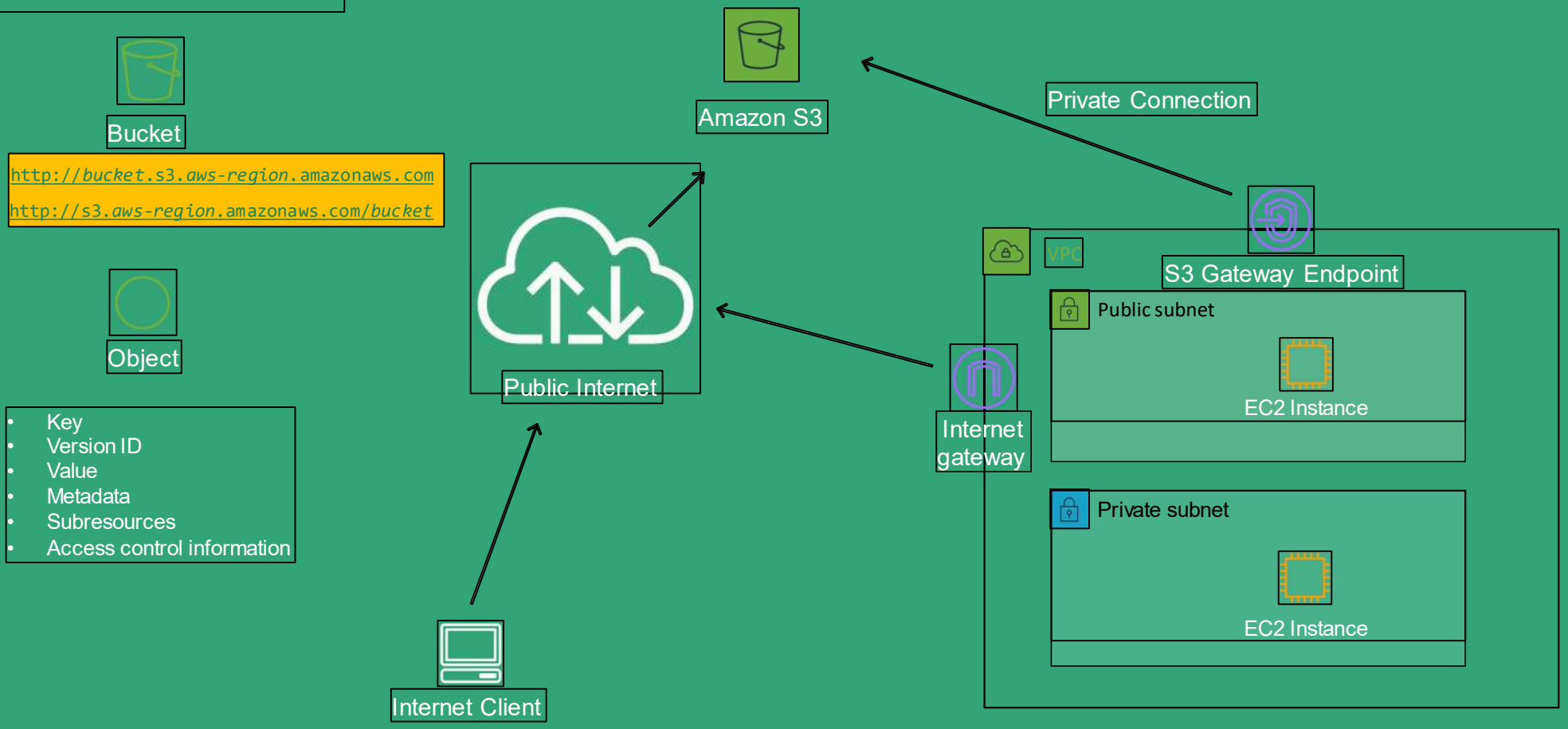


Elastic Block Store

- Persistent block storage volumes
- 99.999% availability
- Automatically replicated within its Availability Zone (AZ)
- Point-in-time snapshot support
- Modify volume type as needs change
- SSD or HDD
- Auto recovery

LINK

Section 7: Amazon S3



Section 7: Amazon Simple Storage Service (S3)

Amazon Simple Storage Service (S3)

- Amazon S3 is object storage built to store and retrieve any amount of data from anywhere – web sites and mobile apps, corporate applications, and data from IoT sensors or devices
- You can store any type of file in S3
- S3 is designed to deliver 99.999999999% durability
- Typical use cases include:
  - Backup and Storage – Provide data backup and storage services for others
  - Application Hosting – Provide services that deploy, install, and manage web applications
  - Media Hosting – Build a redundant, scalable, and highly available infrastructure that hosts video, photo, or music uploads and downloads
  - Software Delivery – Host your software applications that customers can download
  - Static Website – you can configure a static website to run from an S3 bucket

Section 7: Amazon Simple Storage Service (S3)

Amazon Simple Storage Service (S3)

- Files are stored in buckets
- Buckets are root level folders
- Files can be anywhere from 0 bytes to 5 TB
- There is unlimited storage available
- S3 is a universal namespace so bucket names must be unique globally
- However, you create your buckets within a REGION
- It is a best practice to create buckets in regions that are physically closest to your users to reduce latency
- Objects consist of:
  - Key (name of the object)
  - Value (data made up of a sequence of bytes)
  - Version ID (used for versioning)
  - Metadata (data about the data that is stored)

Section 7: Amazon Simple Storage Service (S3)

Amazon Simple Storage Service (S3)

- Pricing:
  - Storage
  - Requests
  - Storage management pricing
  - Data transfer pricing
  - Transfer acceleration

Section 7: Amazon Simple Storage Service (S3)

Amazon Simple Storage Service (S3)

- There are six S3 storage classes:
  - S3 Standard (durable, immediately available, frequently accessed)
  - S3 Intelligent-Tiering (automatically moves data to the most cost-effective tier)
  - S3 Standard-IA (durable, immediately available, infrequently accessed)
  - S3 One Zone-IA (lower cost for infrequently accessed data with less resilience)
  - S3 Glacier (archived data, retrieval times in minutes or hours)
  - S3 Glacier Deep Archive (lowest cost storage class for long term retention)

Section 7: Amazon Simple Storage Service (S3)

	S3 Standard	S3 Intelligent-Tiering*	S3 Standard-IA	S3 One Zone-IA†	S3 Glacier	S3 Glacier Deep Archive
Designed for durability	99.999999999% (11 9's)	99.999999999% (11 9's)	99.999999999% (11 9's)	99.999999999% (11 9's)	99.999999999% (11 9's)	99.999999999% (11 9's)
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	millseconds	milliseconds	milliseconds	select minutes or hours	select hours
Storage type	Object	Object	Object	Object	Object	Object
Lifecycle transitions	Yes	Yes	Yes	Yes	Yes	Yes

	Expedited	Standard	Bulk
Data access time (Glacier)	1-5 minutes	3-5 hours	5-12 hours
Data access time (Deep Archive)	N/A	12 hours	48 hours

Amazon Simple Storage Service (S3)

- S3 Glacier Vault Lock
- S3 Object lock write-once-read-many WORM

Section 7: Amazon Simple Storage Service (S3)

Amazon Simple Storage Service (S3)

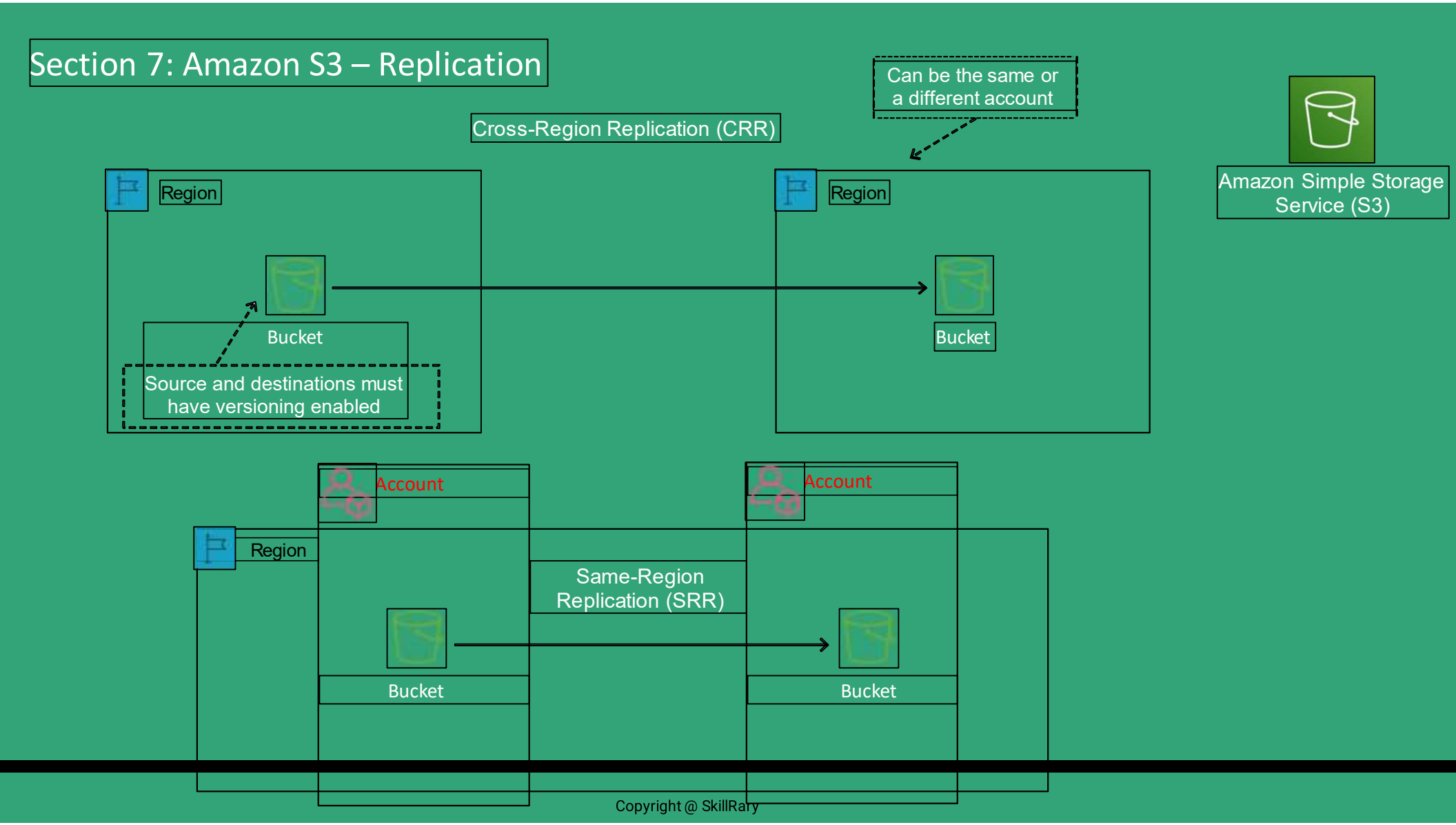
S3 Capability	How it works
Transfer Acceleration	Speed up data uploads using CloudFront in reverse
Requester Pays	The requester rather than the bucket owner pays for requests and data transfer
Tags	Assign tags to objects to use in costing, billing, security etc.
Events	Trigger notifications to SNS, SQS, or Lambda when certain events happen in your bucket
Static Web Hosting	Simple and massively scalable static website hosting
BitTorrent	Use the BitTorrent protocol to retrieve any publicly available object by automatically generating a .torrent file

Section 7: Amazon S3 –Versioning

Amazon Simple Storage Service (S3)

- Amazon S3 versioning maintains multiple variants of an object in the same bucket
- Can be used to preserve, retrieve, and restore every version of every object in an S3 bucket
- Can be enabled at any time
- Once enabled can be suspended

Section 7: Amazon S3 – Replication



Section 7: Amazon S3 –Replication

Amazon Simple Storage Service (S3)

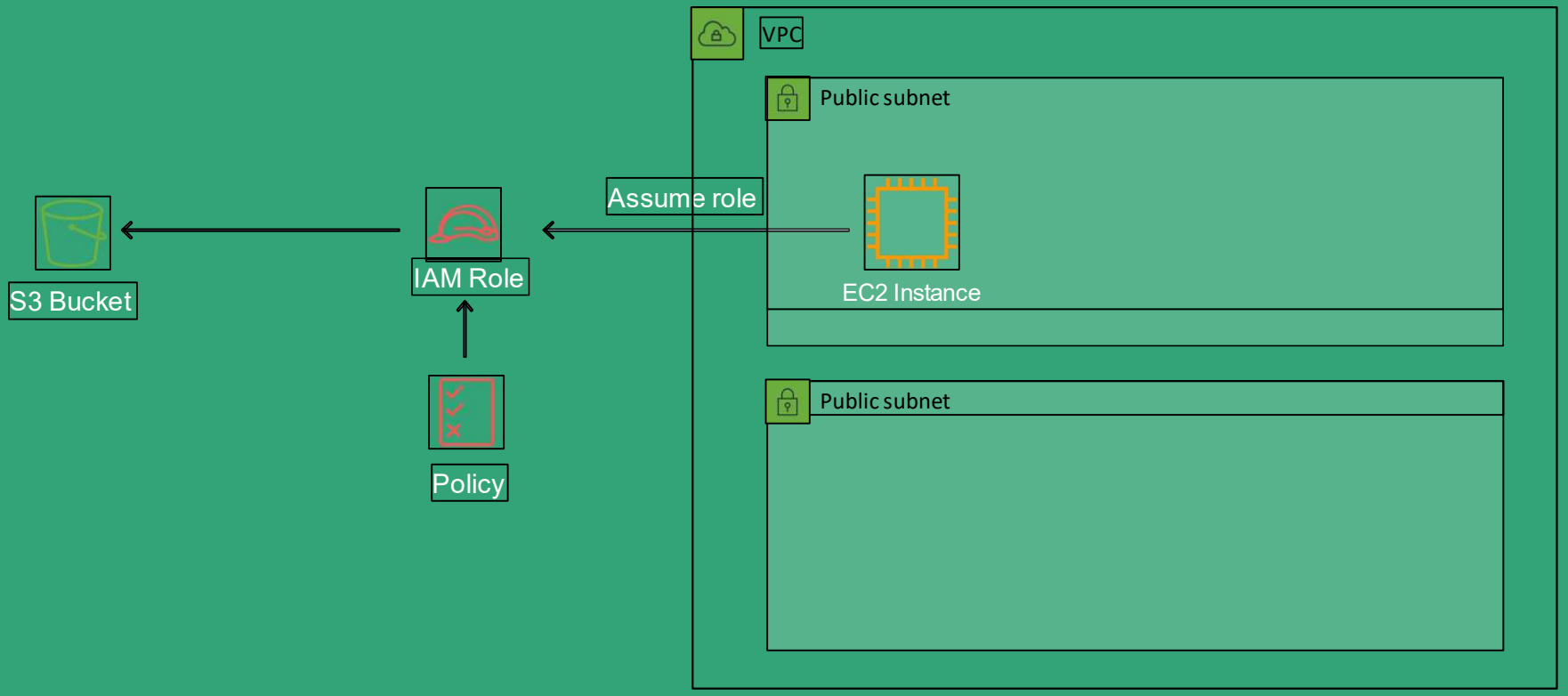
- You can replicate objects between different AWS Regions or within the same AWS Region
  - Cross-Region replication (CRR) is used to copy objects across Amazon S3 buckets in different AWS Regions
  - Same-Region replication (SRR) is used to copy objects across Amazon S3 buckets in the same AWS Region
- Why use replication?
  - Meet compliance requirements for storing data at greater distances (CRR)
  - Minimize latency for users who are closer to another AWS Region (CRR)
  - Backup copy of your data in another AWS Region (CRR)
  - Copy the objects to another S3 storage class
  - Aggregate logs into a single bucket

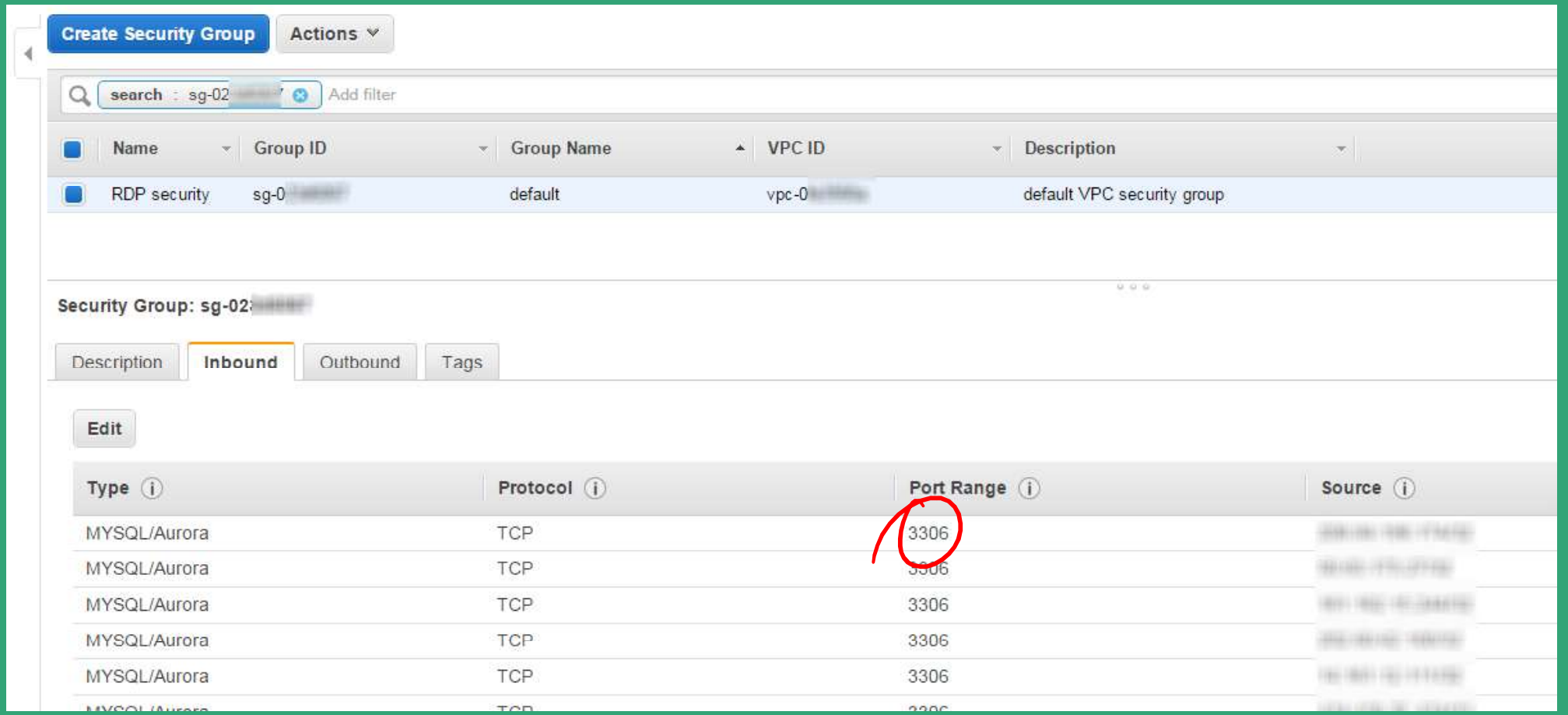
Section 7: Amazon S3 – Replication

Amazon Simple Storage Service (S3)

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

Section 7: Access Amazon S3 Bucket from EC2 with IAM Role





Summary | Inbound Rules | **Outbound Rules** | Subnet Associations | Tags

Allows outbound traffic. Because network ACLs are stateless, you must create inbound and outbound rules.

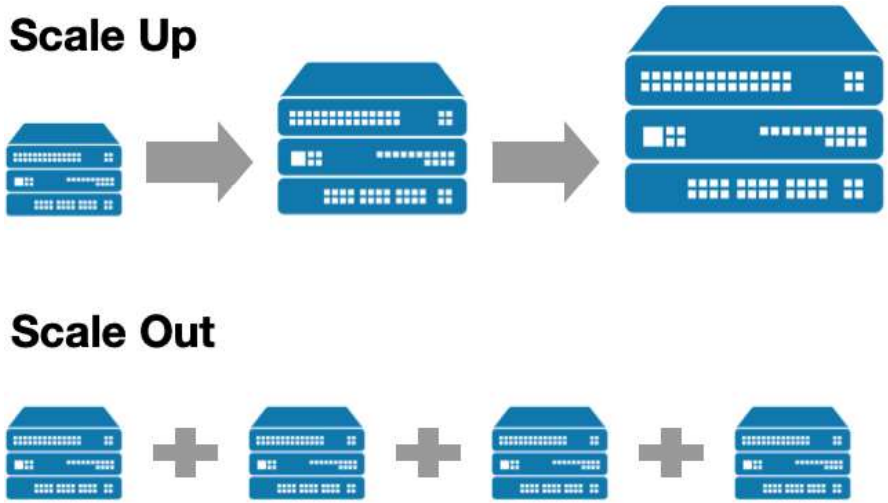
Edit

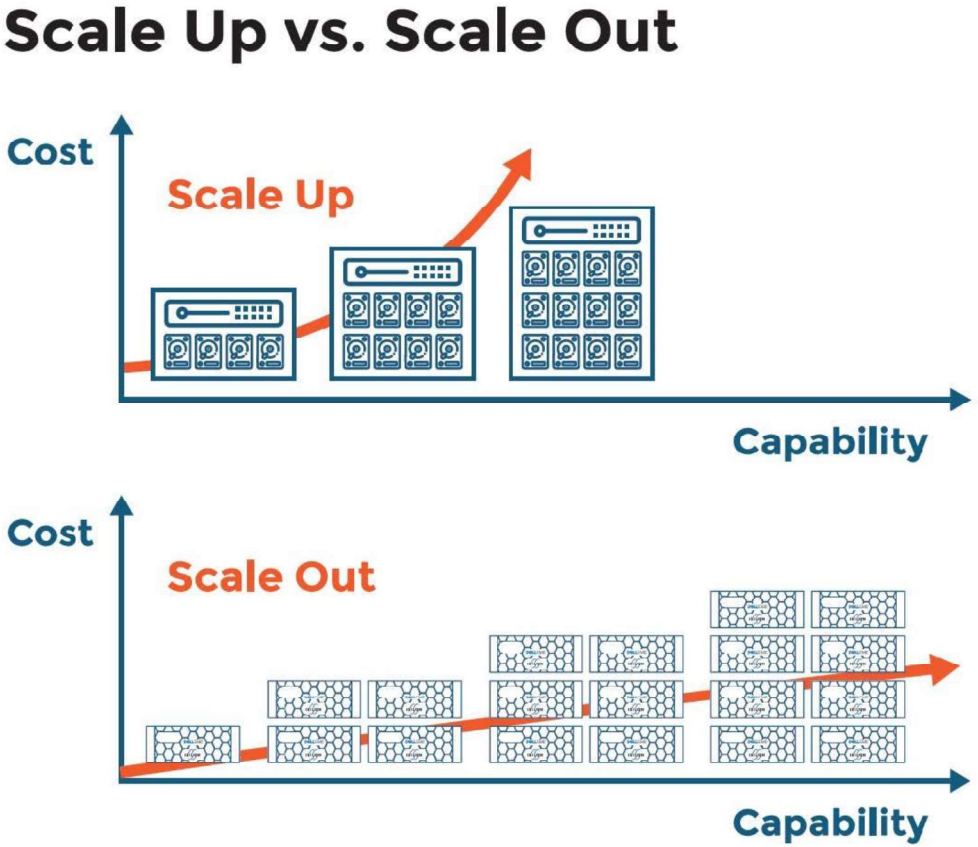
Rule #	Type	Protocol	Port Range	Destination	Allow / Deny
1	All ICMP	ICMP (1)	ALL	0.0.0.0/0	ALLOW
100	Custom TCP Rule	TCP (6)	1024-65535	0.0.0.0/0	ALLOW
200	HTTP (80)	TCP (6)	80	0.0.0.0/0	ALLOW
300	HTTPS (443)	TCP (6)	443	0.0.0.0/0	ALLOW
400	SSH (22)	TCP (6)	22	10.0.0.0/16	ALLOW
500	MySQL/Aurora (3306)	TCP (6)	3306	10.0.0.0/16	ALLOW
*	ALL Traffic	ALL	ALL	0.0.0.0/0	DENY

Section 8: Public, Private, and Elastic IP addresses

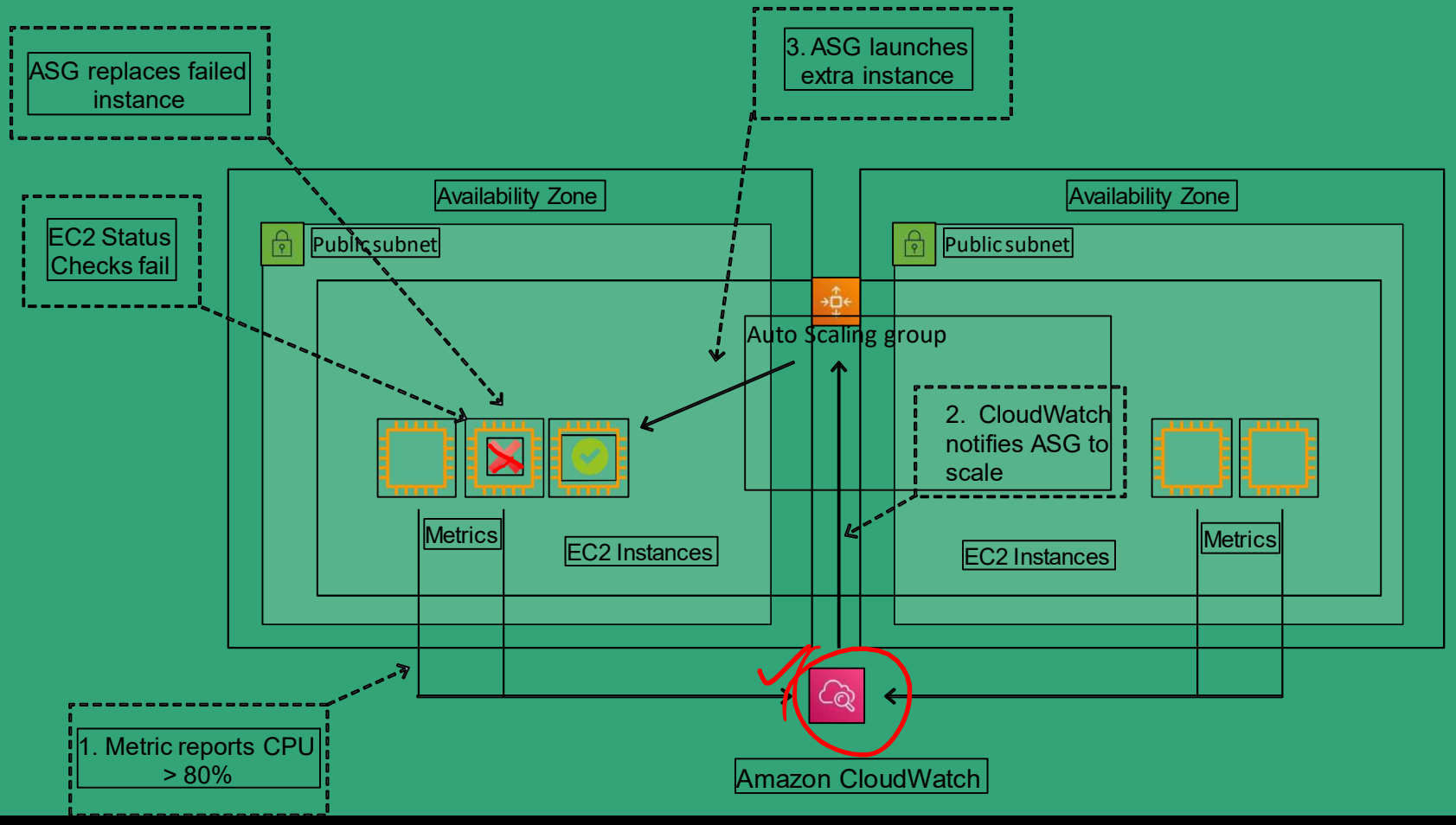
Name	Description
Public IP address	Lost when the instance is stopped Used in Public Subnets No charge Associated with a private IP address on the instance Cannot be moved between instances
Private IP address	Retained when the instance is stopped Used in Public and Private Subnets
Elastic IP address	Static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

Elastic Load Balancing and AutoScaling





Section 10: Amazon EC2 Auto Scaling



Section 10: Amazon EC2 Auto Scaling

Amazon EC2 Auto Scaling

Link

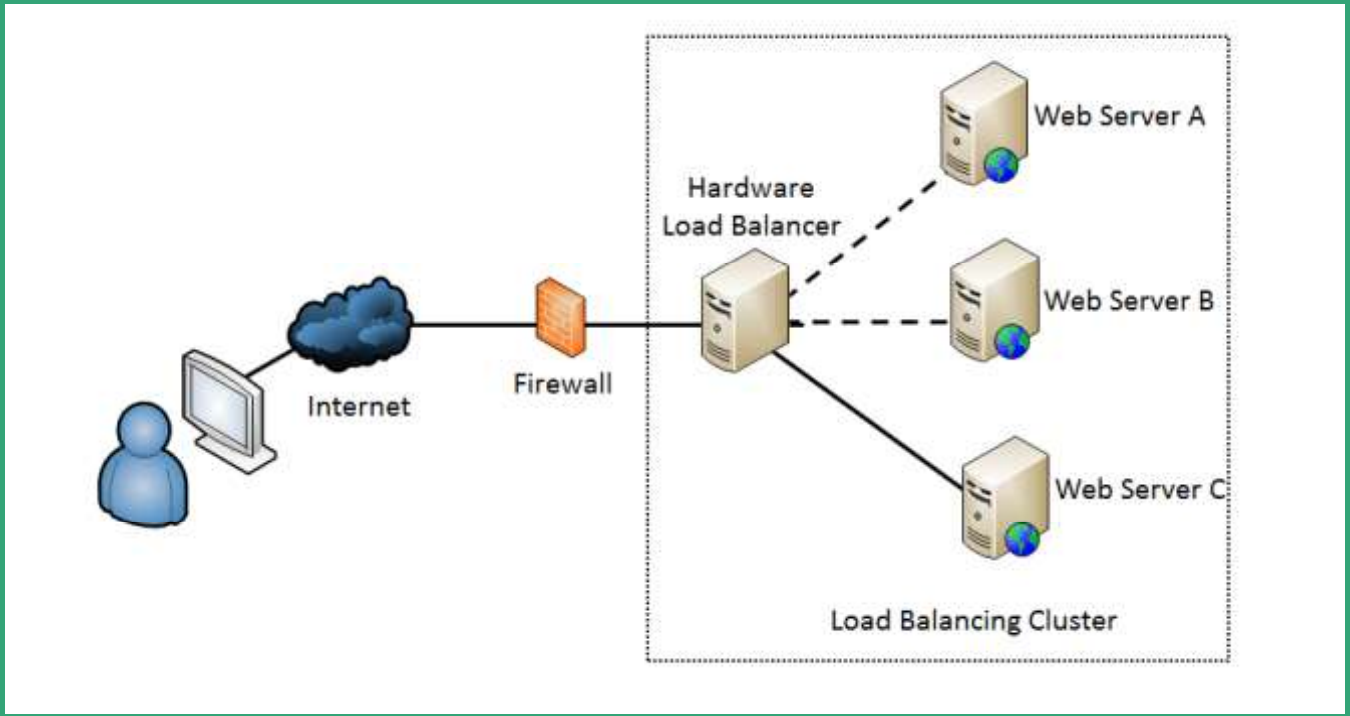
- Amazon EC2 Auto Scaling provides *horizontal* scaling
- Auto Scaling provides elasticity and scalability
- AWS Auto Scaling automates the process of adding (scaling up) OR removing (scaling down) EC2 instances based on the traffic demand for your application
- Auto Scaling helps to ensure that you have the correct number of EC2 instances available to handle the application load
- You create collections of EC2 instances, called Auto Scaling Group (ASG)
- You can specify the minimum number of instances in each ASG, and AWS Auto Scaling will ensure the group never goes beneath this size
- You can also specify the maximum number of instances in each ASG and the group will never go above this size
- A desired capacity can be configured and AWS Auto Scaling will ensure the group has this number of instances

Section 10: Amazon EC2 Auto Scaling

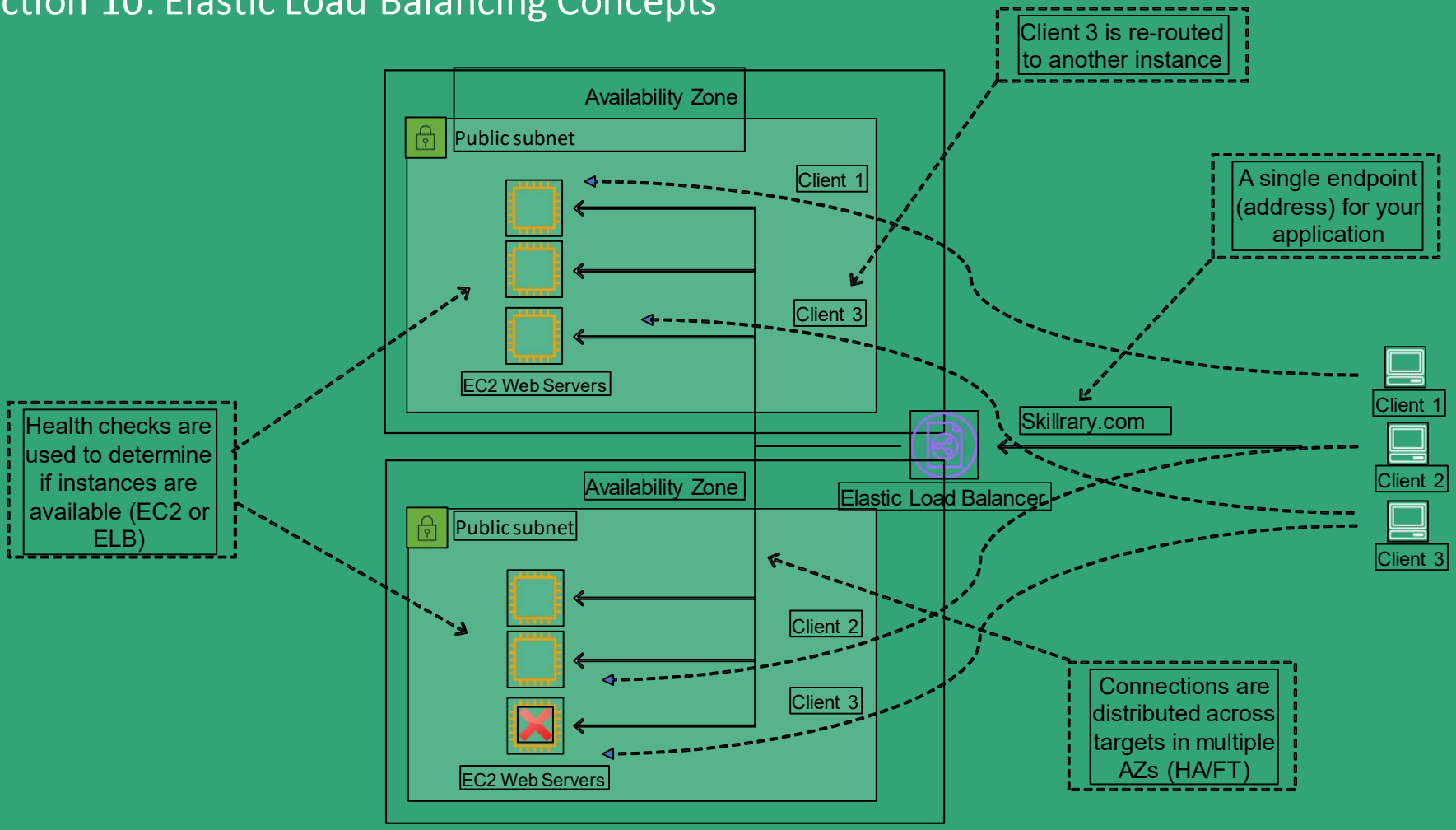

Amazon EC2 Auto Scaling

- You can also specify scaling policies that control when Auto Scaling launches or terminates instances
- Scaling policies determine when, if, and how the ASG scales and shrinks (on-demand/dynamic scaling, cyclic/scheduled scaling)
- Scaling Plans define the triggers and when instances should be provisioned/de-provisioned
- A launch configuration is the template used to create new EC2 instances and includes parameters such as instance family, instance type, AMI, key pair and security groups

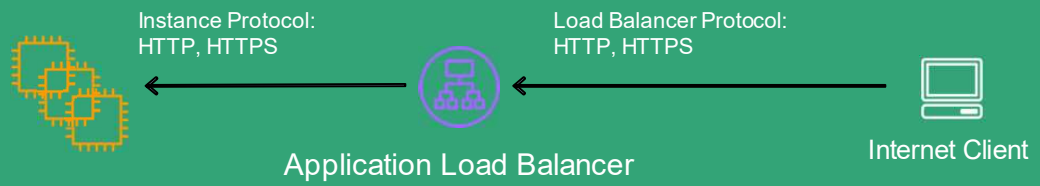
Section 10: Elastic Load Balancing Concepts



Section 10: Elastic Load Balancing Concepts

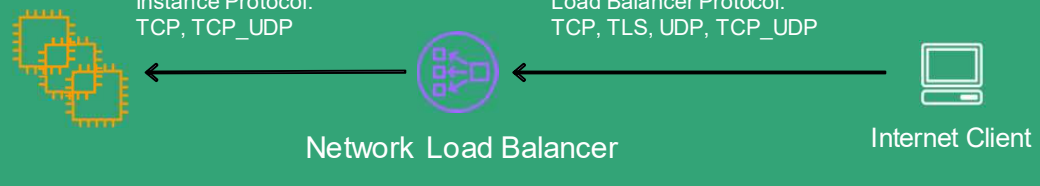


Section 10: Elastic Load Balancing (ELB) Types



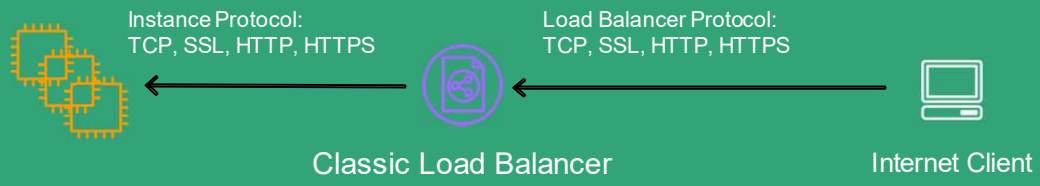
Application Load Balancer

- 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 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 static IP / Elastic IP
- Supports UDP and static IP addresses as targets



Classic Load Balancer

- Old generation; not recommended for new applications
- Performs routing at Layer 4 and Layer 7
- Use for existing applications running in EC2-Classic

Gateway Load Balancer

Section 10: Elastic Load Balancing

Elastic Load Balancing

Link

- ELB automatically distributes incoming application traffic across multiple targets, such as Amazon EC2 instances, containers, and IP addresses
- ELB can handle the varying load of your application traffic in a single Availability Zone or across multiple Availability Zones
- ELB features high availability, automatic scaling, and robust security necessary to make your applications fault tolerant
- There are three types of Elastic Load Balancer (ELB) on AWS:
  - Application Load Balancer (ALB) – layer 7 load balancer that routes connections based on the content of the request
  - Network Load Balancer (NLB) – layer 4 load balancer that routes connections based on IP protocol data
  - Classic Load Balancer (CLB) – this is the oldest of the three and provides basic load balancing at both layer 4 and layer 7

Section 10: Elastic Load Balancing

Elastic Load Balancing

Application Load Balancer (ALB)

- ALB is best suited for load balancing of HTTP and HTTPS traffic and provides advanced request routing targeted at the delivery of modern application architectures, including microservices and containers
- Operating at the individual request level (Layer 7), Application Load Balancer routes traffic to targets within Amazon Virtual Private Cloud (Amazon VPC) based on the content of the request

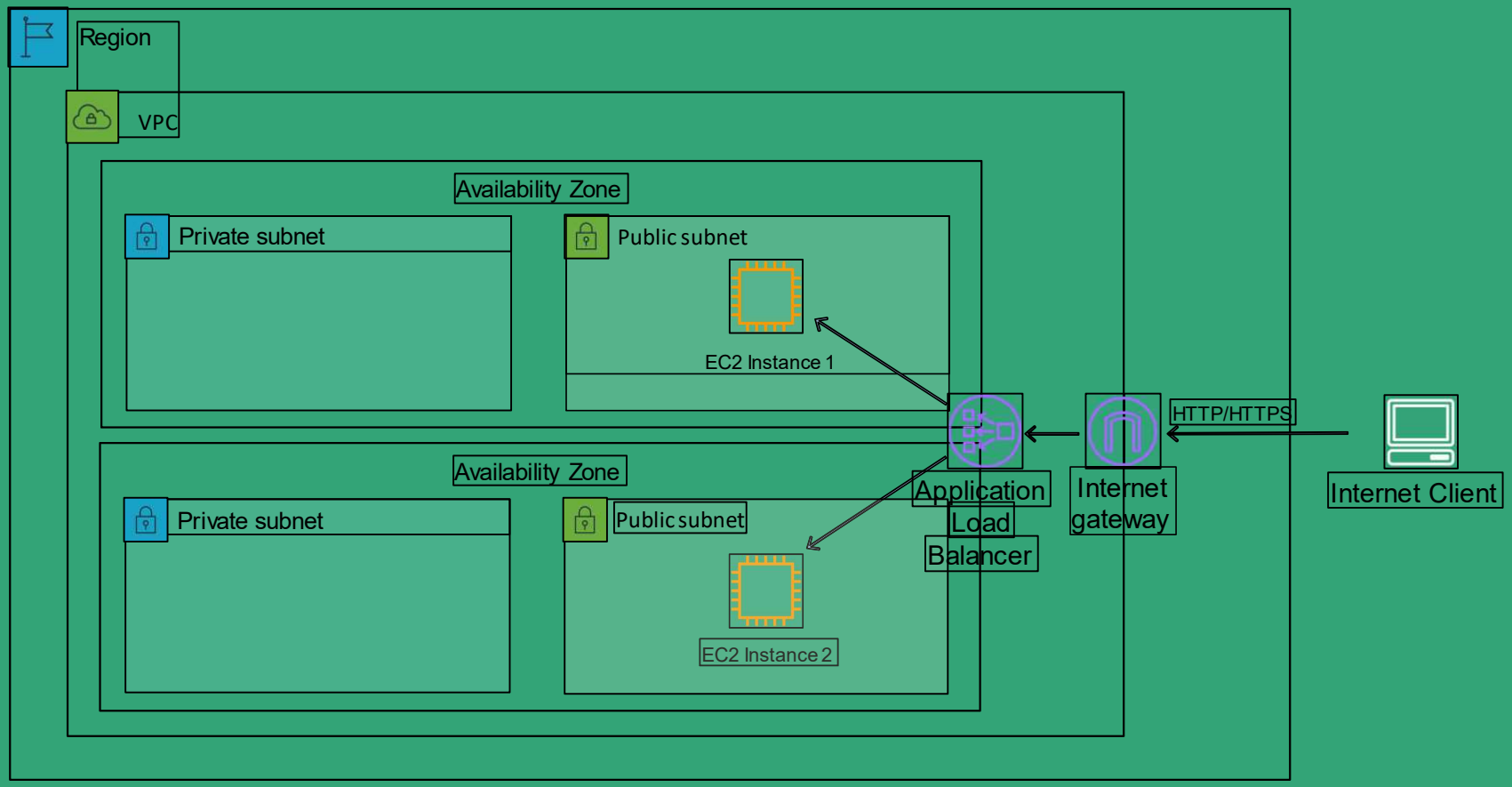
Network Load Balancer (NLB)

- NLB is best suited for load balancing of TCP traffic where extreme performance is required
- Operating at the connection level (Layer 4), NLB is capable of handling millions of requests per second while maintaining ultra-low latencies

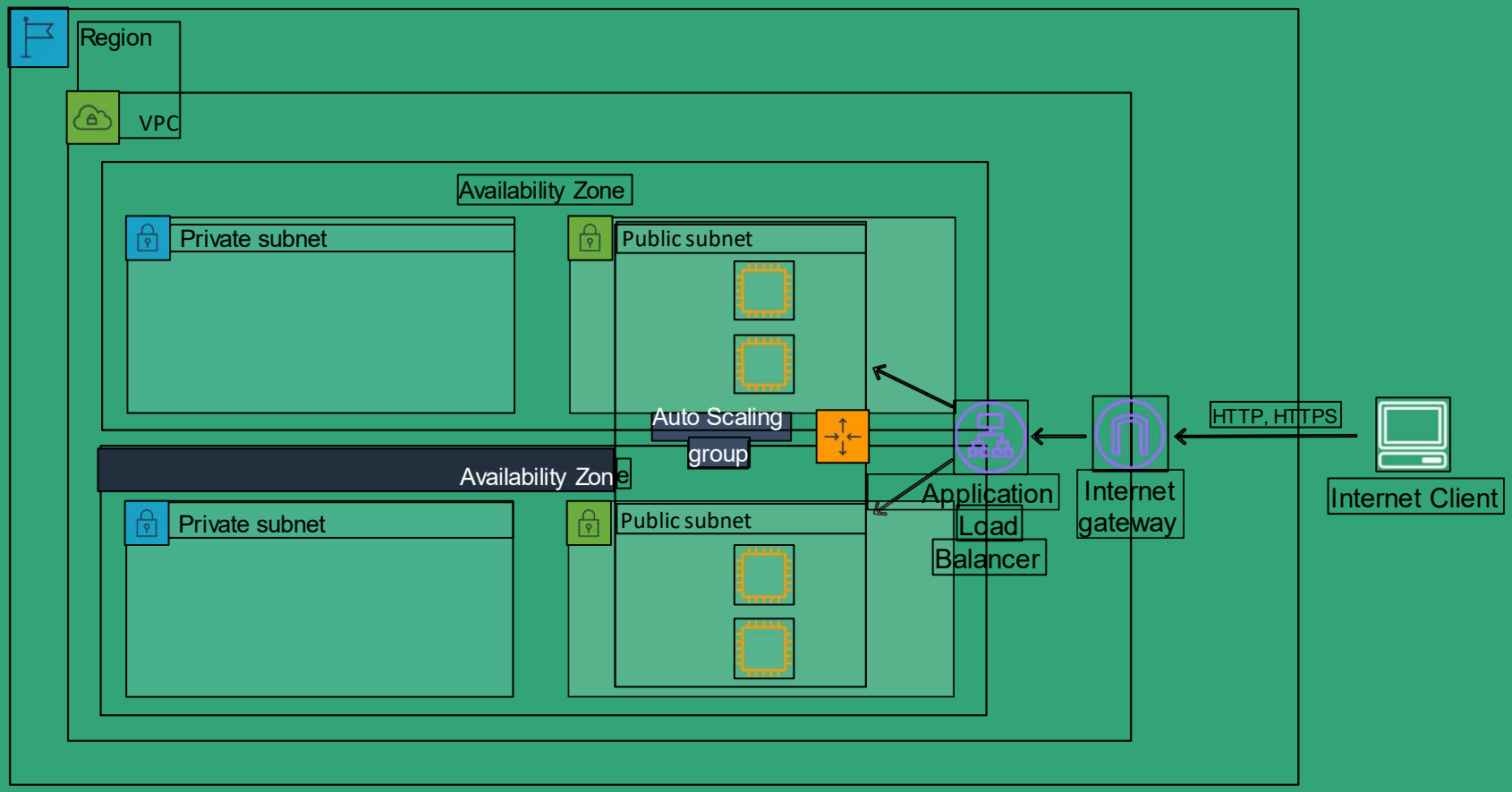
Classic Load Balancer (CLB)

- CLB provides basic load balancing across multiple Amazon EC2 instances and operates at both the request level and connection level

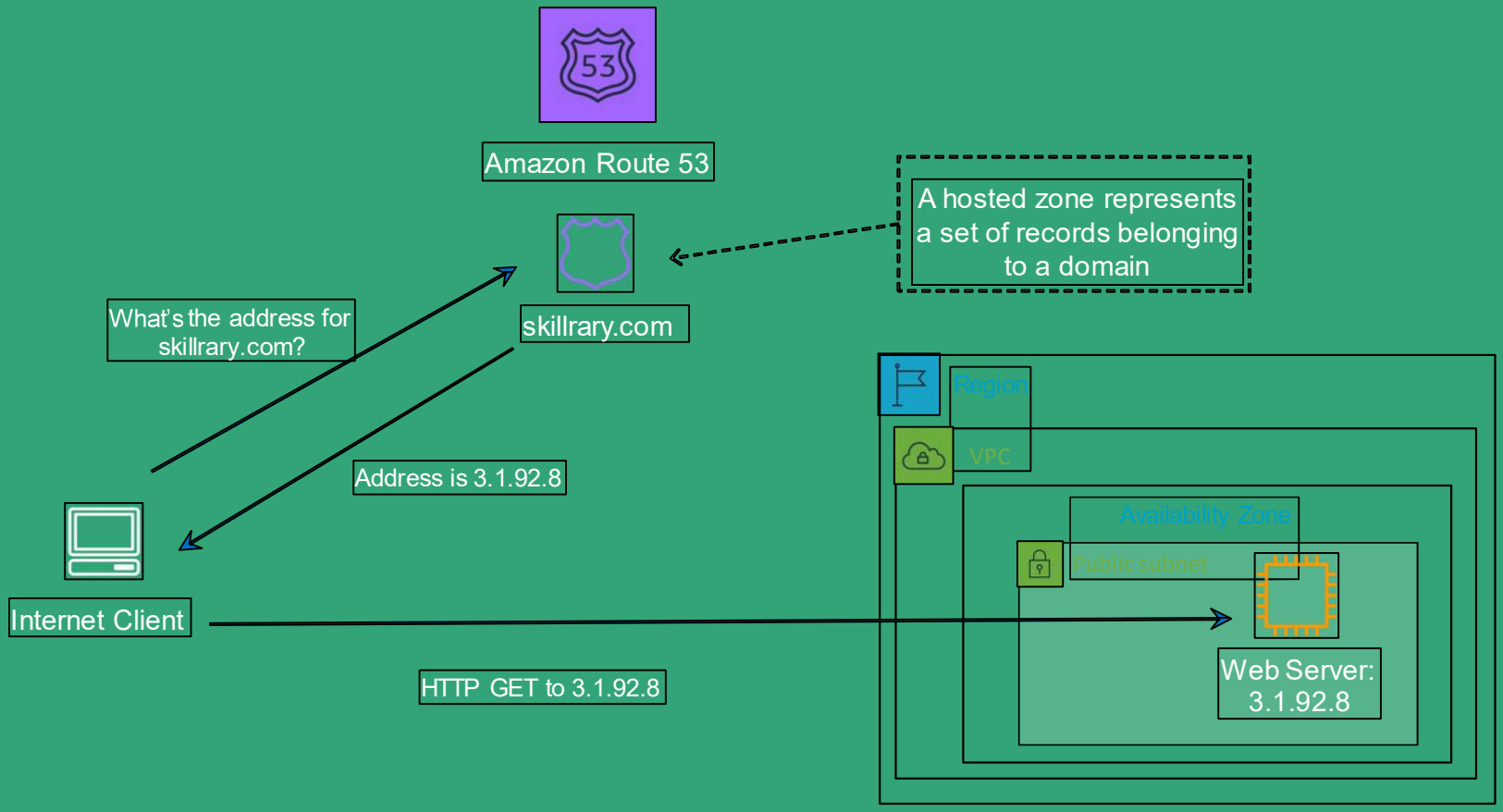
Application Load Balancer (Internet-Facing)



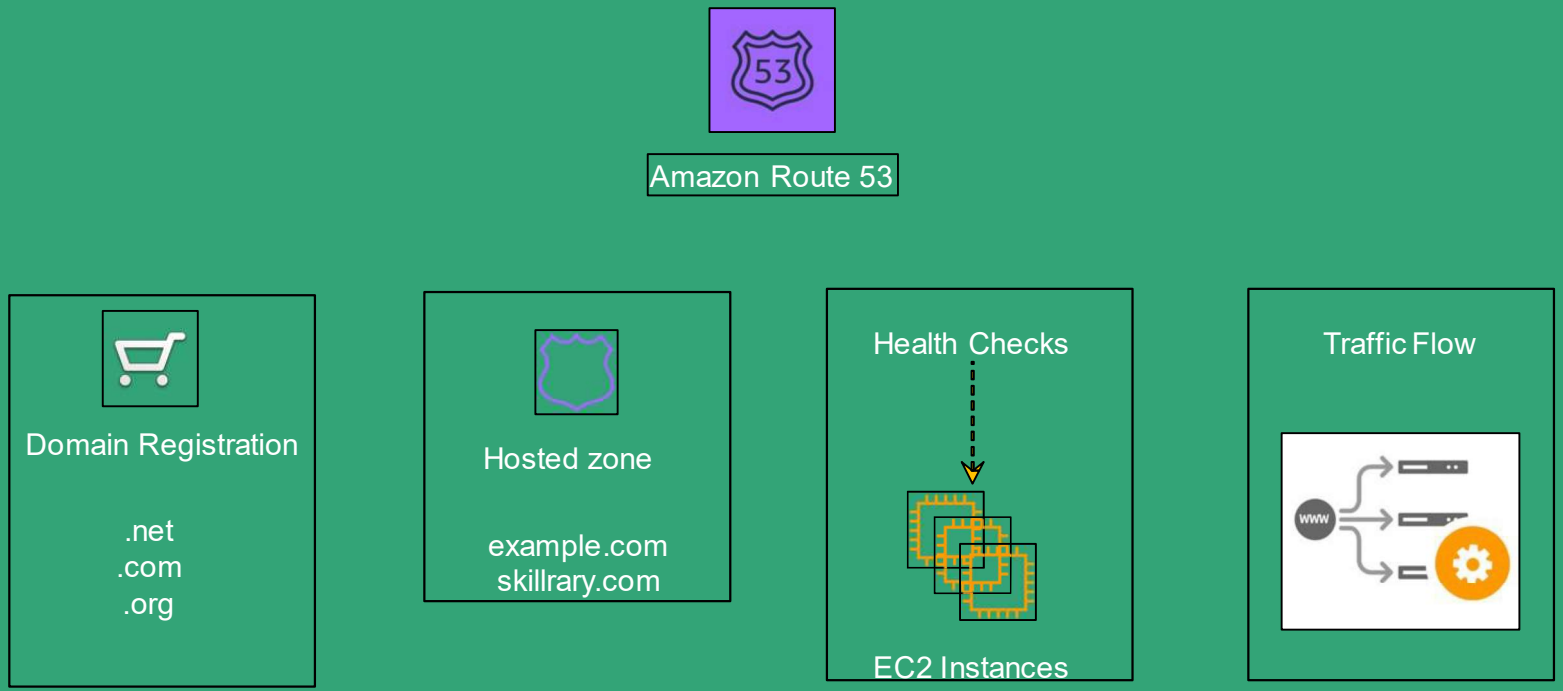
Auto Scaling Group with ALB



Section 11: DNS Resolution with AWS Route 53

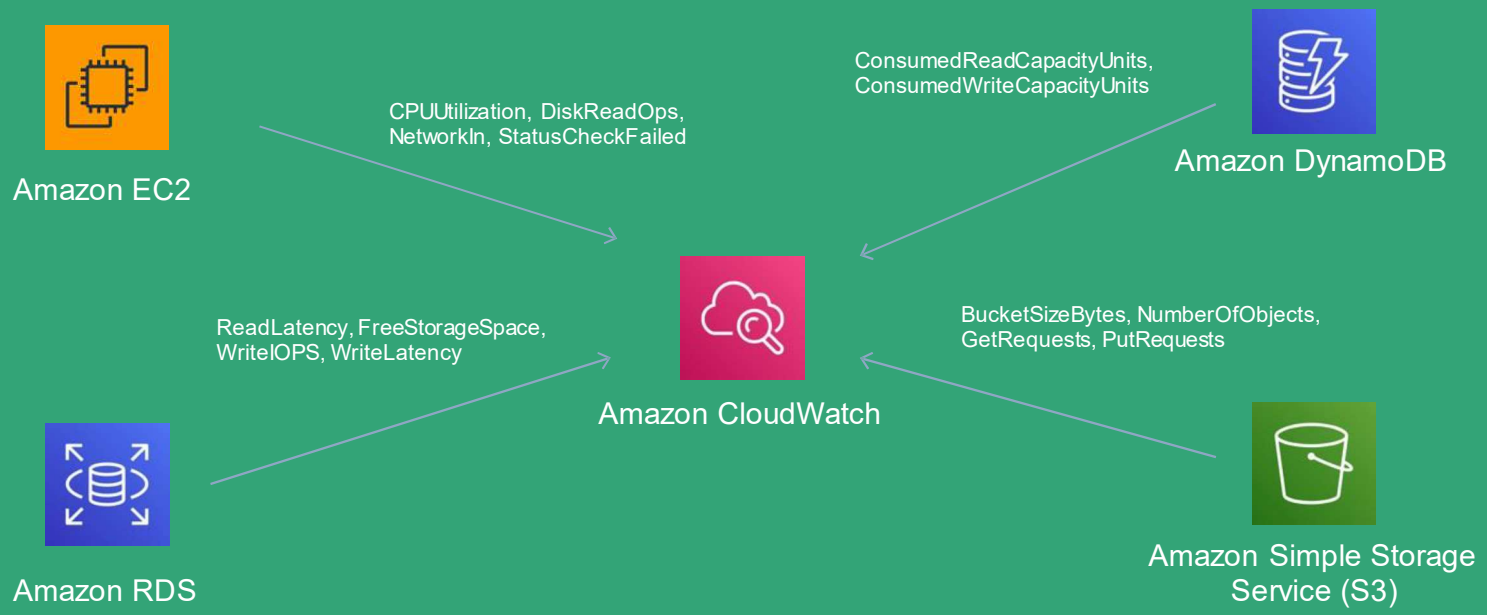


Section 11: Amazon Route 53



Monitoring and Logging Services

Section 12: Monitoring with Amazon CloudWatch



Section 12: Amazon CloudWatch

Amazon CloudWatch

Link

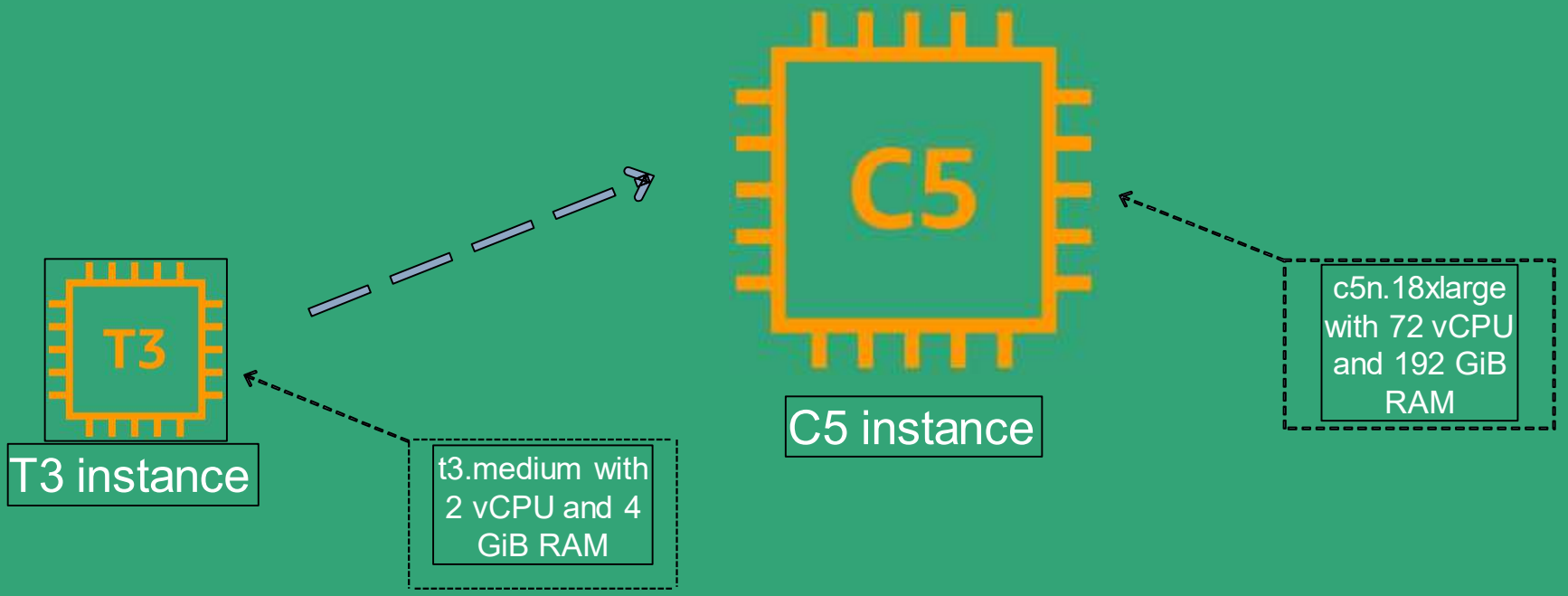
- Amazon CloudWatch is a monitoring service for AWS cloud resources and the applications you run on AWS
- CloudWatch is for performance monitoring (CloudTrail is for auditing)
- Used to collect and track metrics, collect and monitor log files, and set alarms
- CloudWatch is a regional service
- CloudWatch Alarms can be set to react to changes in your resources
- CloudWatch Events generates events when resource states change and delivers them to targets for processing
- CloudWatch Logs collects and centralizes logs from AWS resources
- Any log files generated by your applications
- Gain system-wide visibility into resource utilization
- CloudWatch monitoring includes application performance

Architecting for the Cloud

The Difference Between Traditional and Cloud Computing Environments

- IT Assets as Provisioned Resources
  - Provision as needed, and scale on-demand
- Global, Available, and Scalable Capacity
  - Deploy globally, easily, cost-effectively, and quickly
- Higher-Level Managed Services
  - Lower operational cost by leveraging managed storage, database, analytics, application and deployment services
- Built-in Security
  - Leverage AWS' significant investment in security, simplify security testing, and use native security and encryption features
- Architecting for Cost
  - Fine grained billing, transparent costs, budgets and alerting tools
- Operating on AWS
  - Tooling, processes and best practices to support operational transitions

Scaling Vertically



Scaling Vertically

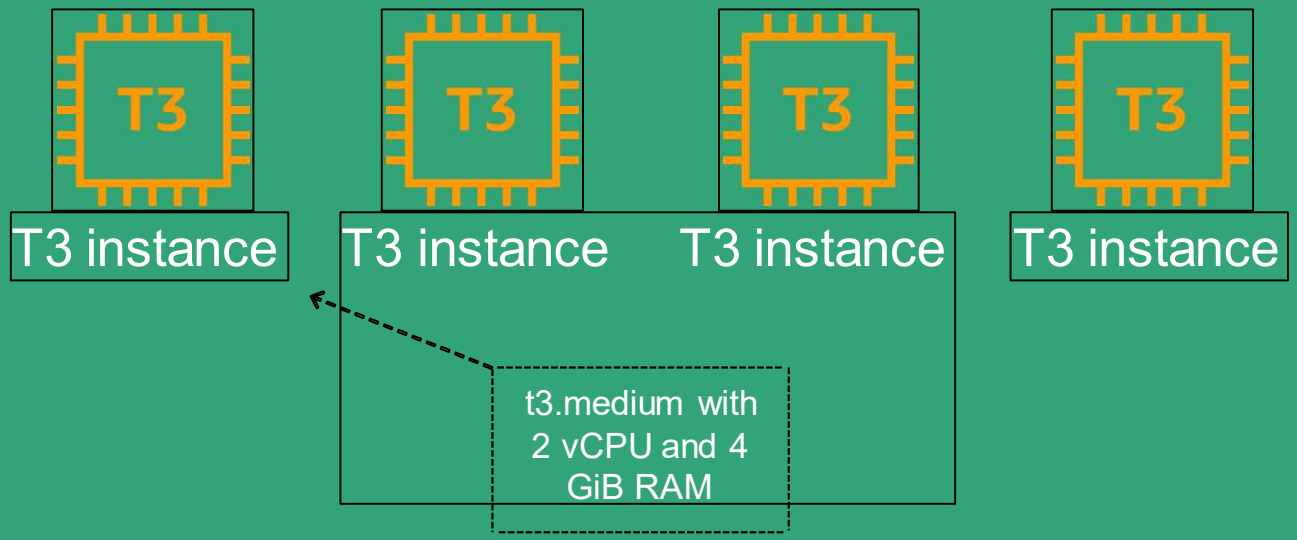
Examples of vertical scaling are:

- Amazon EC2 instances
- Amazon RDS Database instances

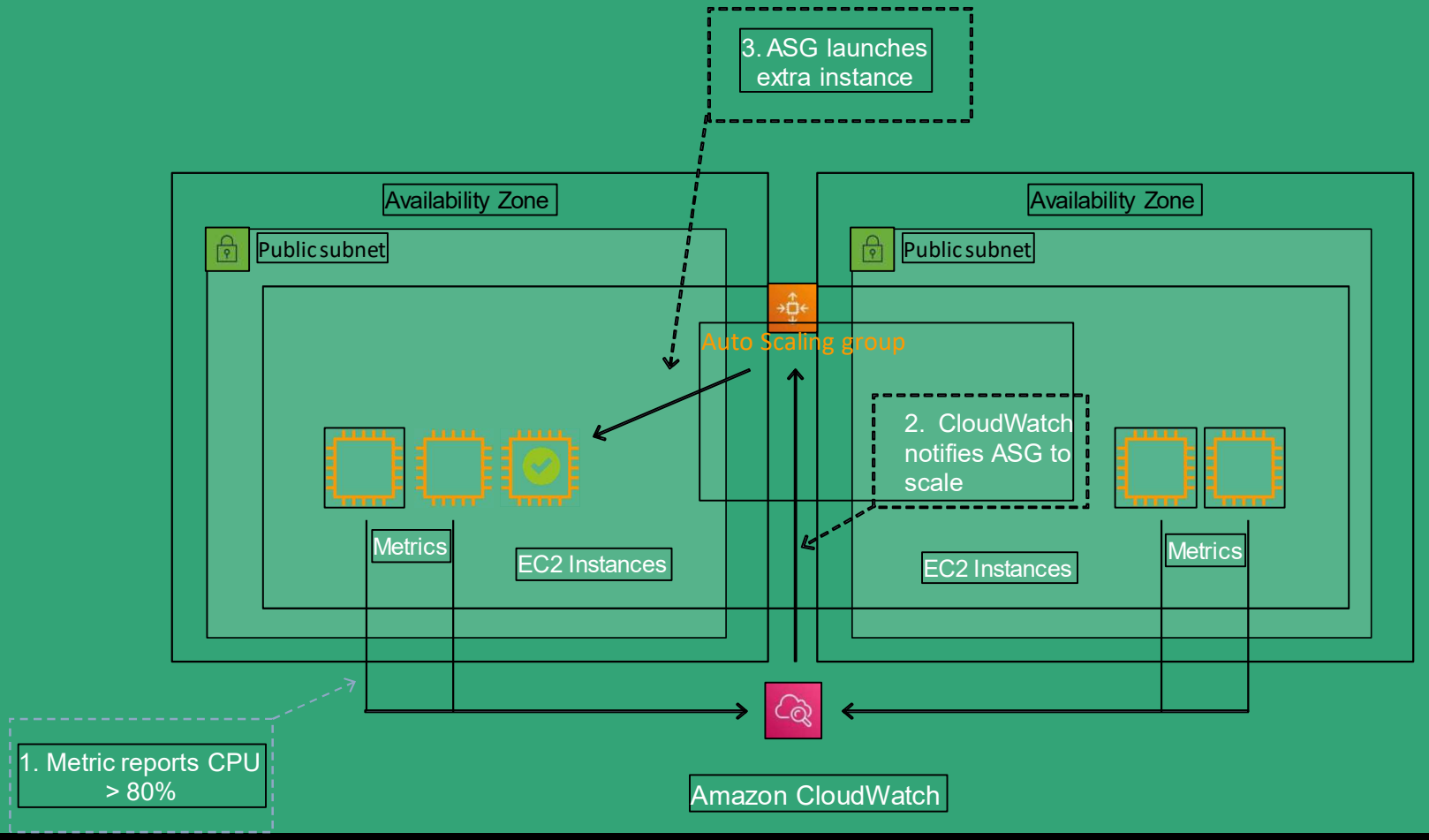
Limitations of scaling vertically:

- Often requires manual intervention (though can be scripted/automated)
- Typically requires downtime
- Can reach a limit of scalability

Scaling Horizontally



Example of Horizontal Scaling with EC2



Scaling Horizontally

Examples of horizontal scaling are:

- Amazon EC2 Auto Scaling
- Amazon DynamoDB

Benefits of scaling horizontally:

- Seamless scaling, without downtime
- Can scale almost limitlessly (in some cases)

The Five Pillars of Operational Excellence

1) Operational Excellence
   - The operational excellence pillar includes the ability to run and monitor systems to deliver business value and to continually improve supporting processes and procedures
2) Security
   - The security pillar includes the ability to protect information, systems, and assets while delivering business value through risk assessments and mitigation strategies
3) Reliability
   - The reliability pillar includes the ability of a system to recover from infrastructure or service disruptions, dynamically acquire computing resources to meet demand, and mitigate disruptions such as misconfigurations or transient network issues

The Five Pillars of Operational Excellence

4) Performance Efficiency

- The performance efficiency pillar includes the ability to use computing resources efficiently to meet system requirements and to maintain that efficiency as demand changes and technologies evolve

5) Cost Optimization

- The cost optimization pillar includes the ability to avoid or eliminate unneeded cost or suboptimal resources

Thank you