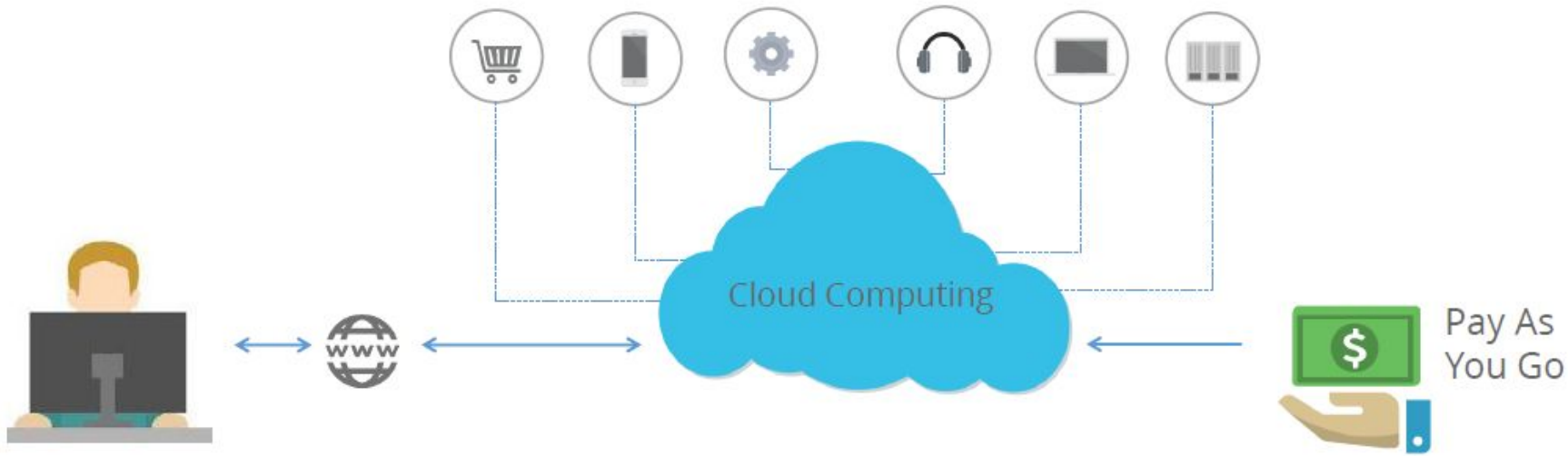
Introduction to Cloud Computing

On-demand provisioning of IT resources and applications through the Internet.



Cloud Computing facilitates:

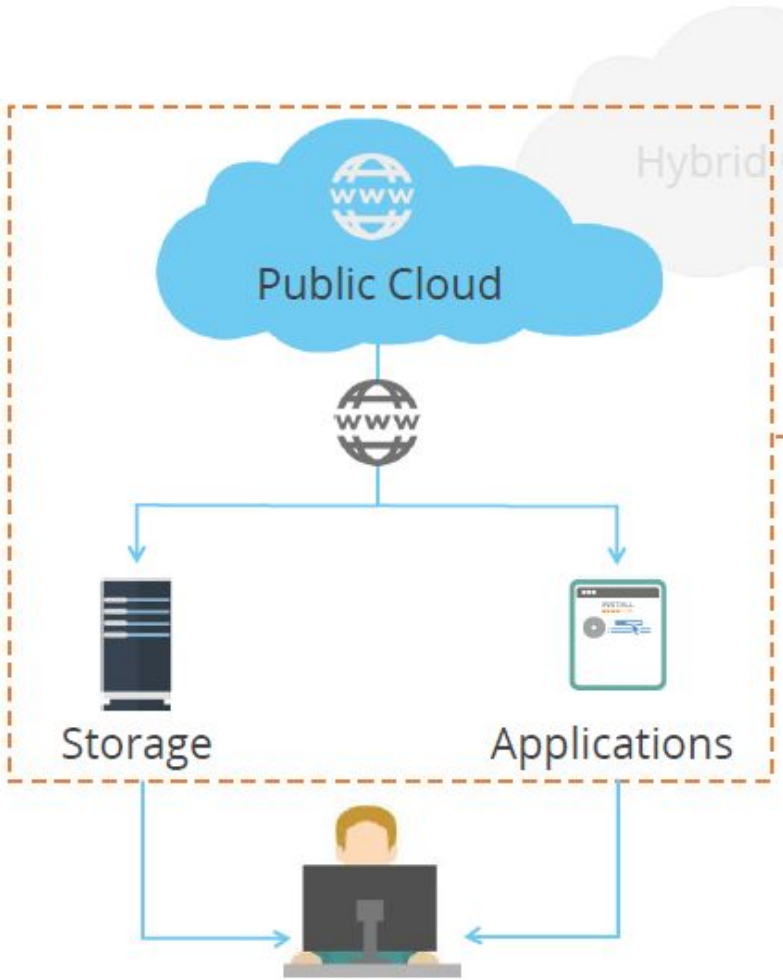
- Quick access to cost-efficient and flexible IT resources
- Accessing servers, databases, storage media, and a variety of application services on the World Wide Web

Six Key Benefits of Cloud Computing

1. You do not incur capital expenses 
2. You can take the advantage of economies of scale 
3. You can optimize the use of resources easily 
4. You can effortlessly boost speed and agility 
5. It becomes simpler to streamline processes 
6. You can enjoy the freedom of going global in a few minutes 

Three Forms of Cloud Computing

Three Forms of Cloud Computing

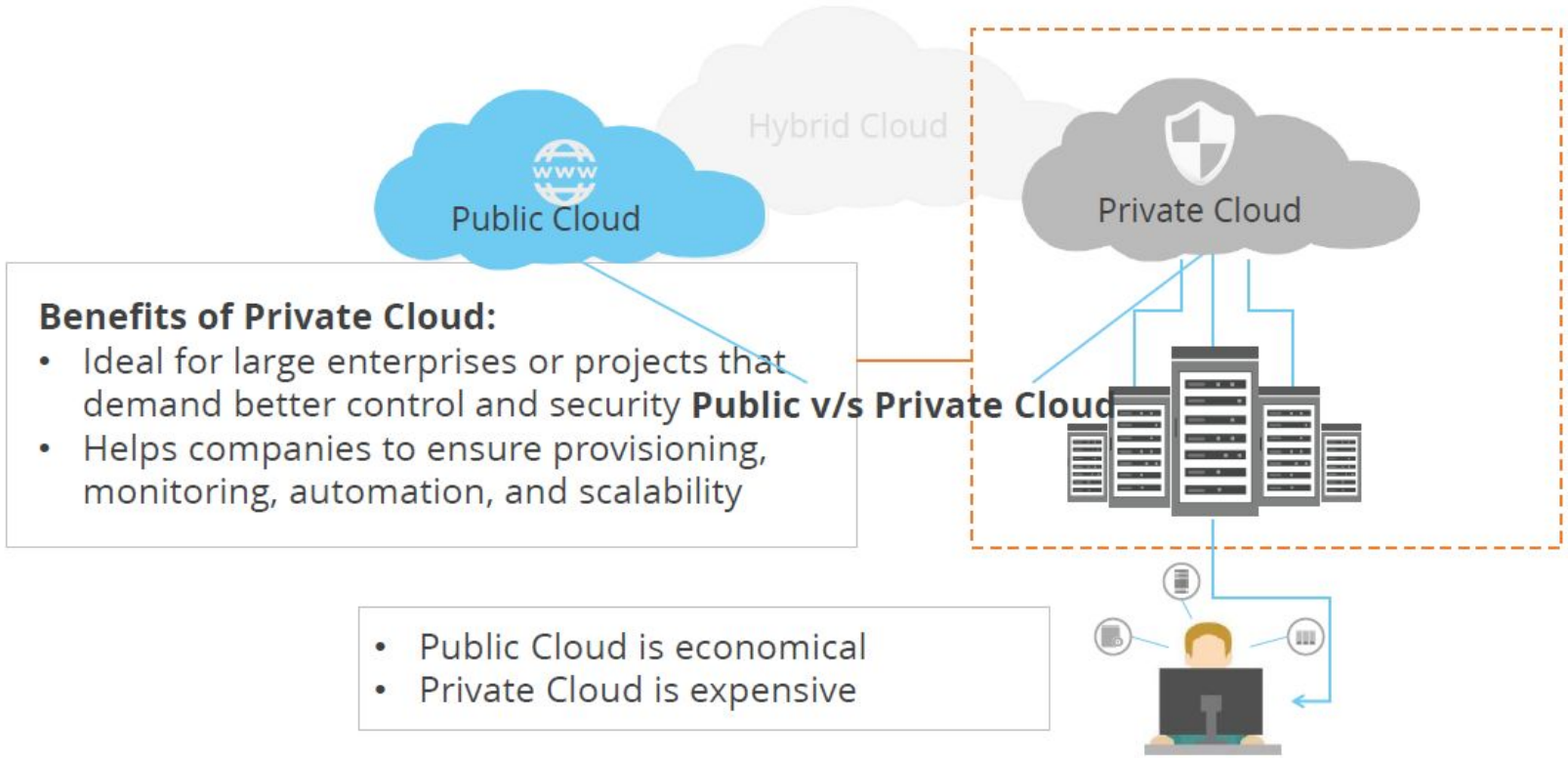


Benefits of Public Cloud:

- Ideal for sharing non-secured applications
- Testing application code

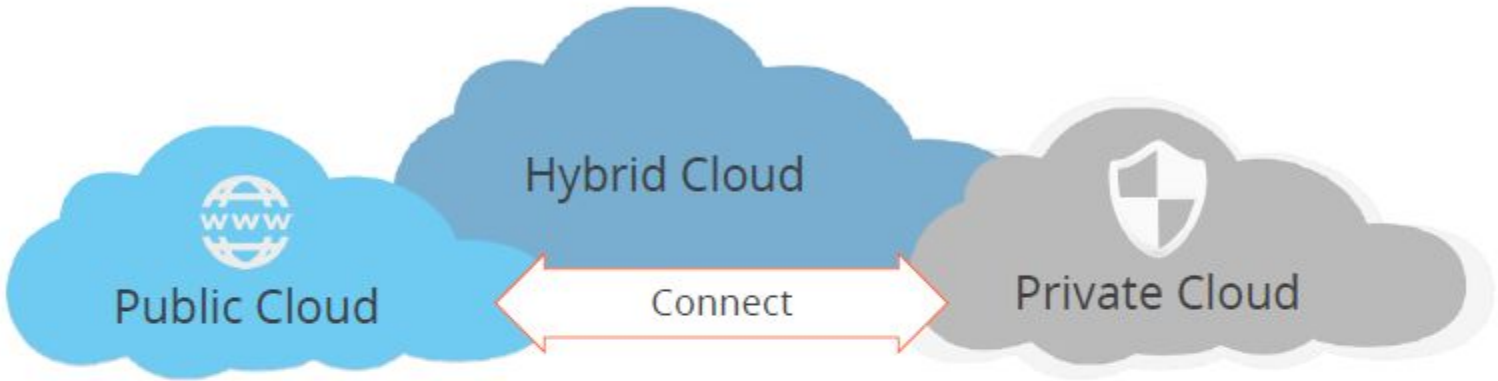
Public cloud solutions are readily available from Google, Amazon, Microsoft, and others

Three Forms of Cloud Computing



Private cloud solutions utilize infrastructure that is either owned and controlled by the organization, or they are able to contractually require those specific criteria be met by a vendor who manages the infrastructure.

Three Forms of Cloud Computing



Public Cloud + Private Cloud

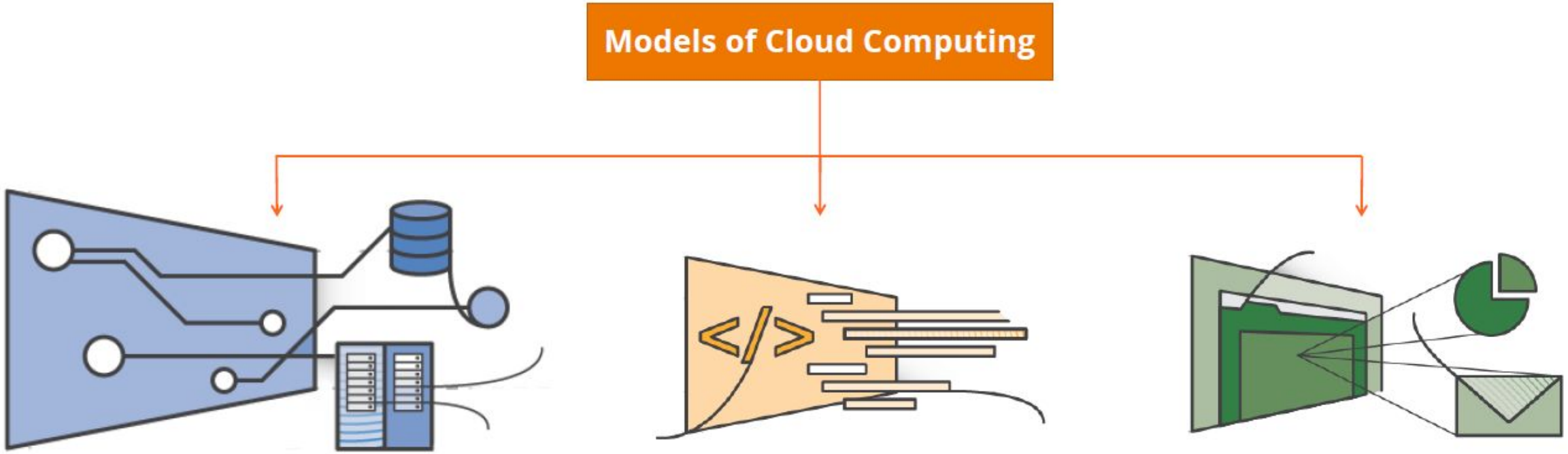
Benefits of Hybrid Cloud:

- Beneficial during forecasted unfavorable events such as scheduled Windows maintenance and hurricane warnings
- Cater to different market verticals

An example of a hybrid cloud solution is an organization that wants to keep confidential information secured on their private cloud, but make more general, customer-facing content on a public cloud.

Models of Cloud Computing

Cloud computing has three models to fulfil the needs of different users. Each of these models come with different levels of management, control, and flexibility.



Infrastructure as a Service

Infrastructure as a Service (IAAS) offers access to networking features, data storage space, and different computers.

Hardware as a Service

Highest Level of Control and Flexibility

Platforms as a Service

Platforms as a Service is responsible for allowing organizations to focus on managing and deploying applications.

Increases the Overall Efficiency

Capacity Planning

Resource Procurement

Tasks of running an application

Software as a Service

Software as a Service refers to end-user applications that are run and managed by service providers.

It eliminates the need to think:

How to Maintain a Specific Application or Service

How to Manage the Underlying Infrastructure

How to Use the Application or Software

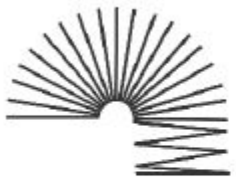
Examples of IaaS, Paas, and SaaS.

IaaS	PaaS	BaaS	SaaS
Amazon EC2	HEROKU	Firebase	zendesk
DigitalOcean	salesforce	Skygear	PayPal
rackspace the open cloud company		Parse Server	

What's AWS?

- AWS (Amazon Web Services) is a Cloud Provider
- They provide you with servers and services that you can use on demand and scale easily
- AWS has revolutionized IT over time
- AWS powers some of the biggest websites in the world
- Amazon.com
- Netflix

Benefits of using AWS



Flexibility

Cost-Effectiveness

Scalability/Elasticity

Security

and Reliable

Flexibility

You get more time for core business tasks through the instant availability of new features and services.

You get a choice in running services and applications. You can choose to run a part of your IT infrastructure in AWS and the remaining in your data centers.

You enjoy effortless hosting of legacy applications.

Cost-Effectiveness

No Upfront Investment

Long-term commitment

Minimum Expense

Scalability and Elasticity

Through Amazon Web Services, techniques such as auto scaling and elastic load balancing can automatically scale resources.

Scale up the required resources to fulfill a sudden demand

Scale them down when the demand falls without affecting speed and performance

Deal with unpredictable and variable loads

Benefits of reduced cost and increased user satisfaction

Security

AWS delivers end-to-end security and privacy to its customers. Its virtual infrastructure offers optimum availability while managing full privacy for customers and isolation of their operations.

Confidentiality

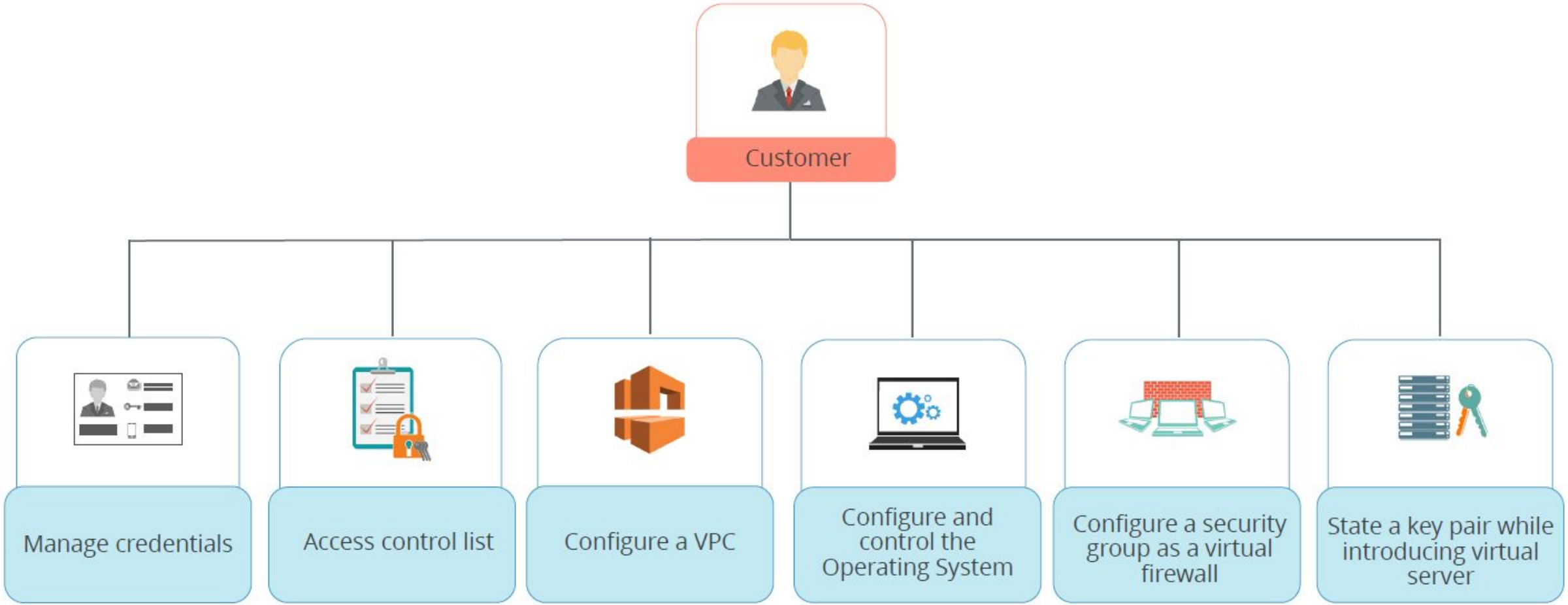
Integrity

Availability

Customers can expect high physical security, and this is due to Amazon's several years of experience in designing, developing, and running large-scale IT operation centers.

The purpose of AWS Compliance is to enable you to understand its powerful controls in action and maintain security and data protection.

Security


Customer
Manage credentials
Access control list
Configure a VPC
Configure and control the Operating System
Configure a security group as a virtual firewall
State a key pair while introducing virtual server

What we'll learn in the upcoming sessions


Amazon EC2


Amazon ECR


Amazon ECS


AWS Elastic Beanstalk


AWS Lambda


Elastic Load Balancing


Amazon CloudFront


Amazon Kinesis

Amazon Route 53


Amazon S3


Amazon RDS


Amazon Aurora


Amazon DynamoDB


Amazon ElastiCache


Amazon SQS


Amazon SNS


AWS Step Functions


Auto Scaling


Amazon API Gateway


Amazon SES


Amazon Cognito


IAM


Amazon CloudWatch


Amazon EC2 Systems Manager


AWS CloudFormation


AWS CloudTrail


AWS CodeCommit


AWS CodeBuild


AWS CodeDeploy


AWS CodePipeline


AWS X-Ray

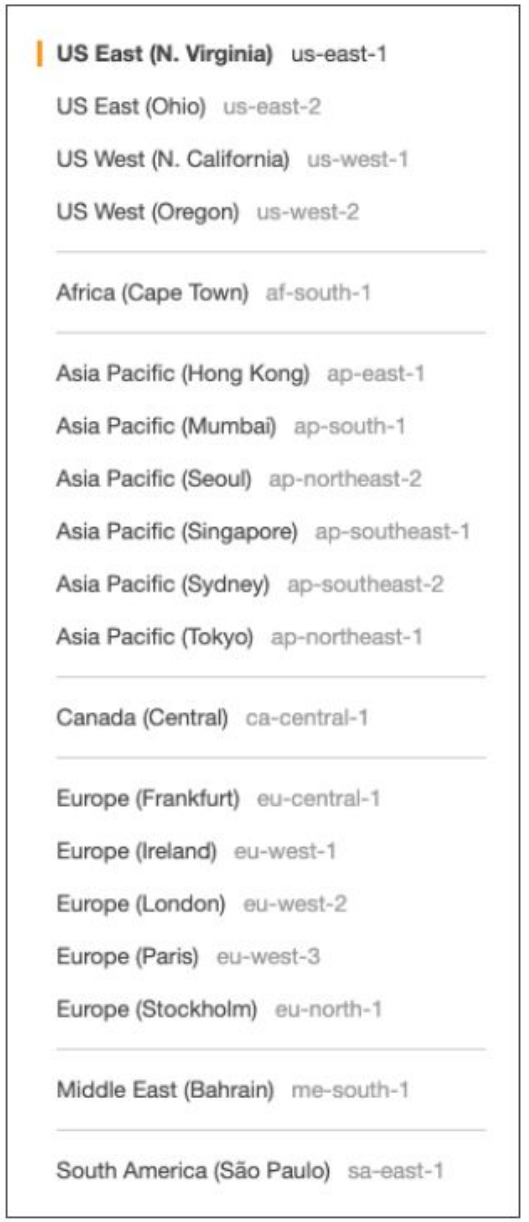

AWS KMS

Before we move on to the services and understanding them let us create
Free tier account and do a AWS Management Console walkthrough

AWS Regions

- AWS has Regions all around the world
- Names can be: us-east-1, eu-west-3…
- A region is a physical location around the world where AWS cluster data centers.
- Most AWS services are region-scoped

https://aws.amazon.com/about-aws/global-infrastructure/



AWS Availability Zones

- Each region has many availability zones

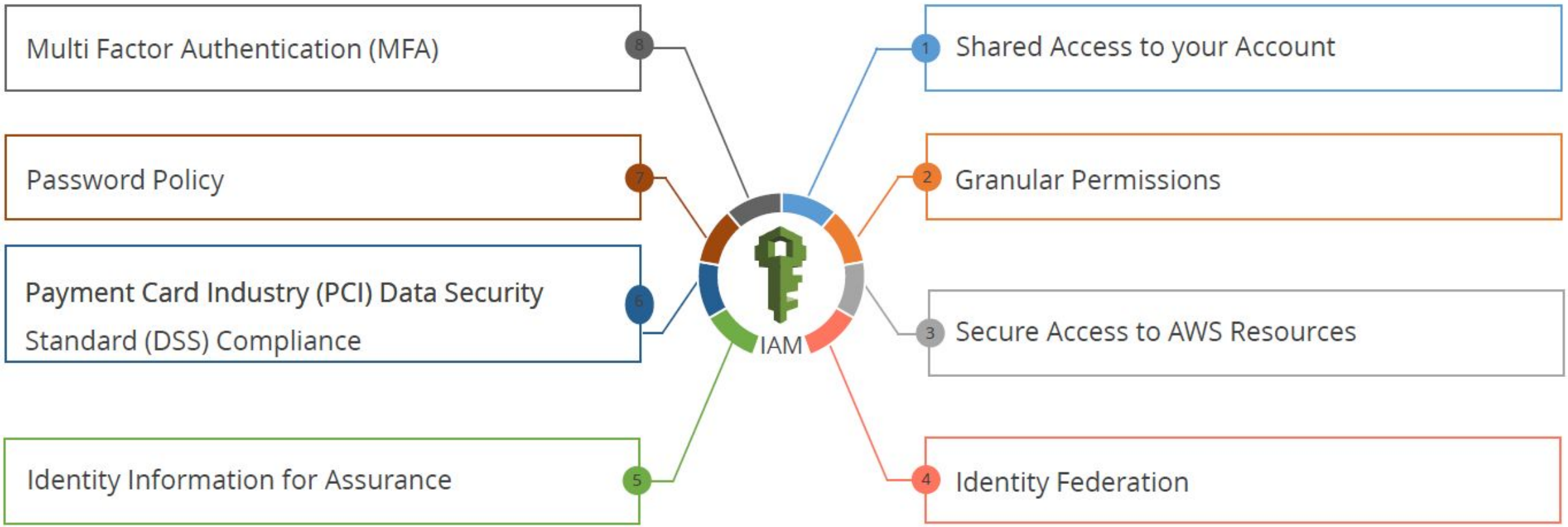
(usually 3, min is 2, max is 6). Example:

- ap-southeast-2a
- ap-southeast-2b
- ap-southeast-2c
- Each availability zone (AZ) is one or more discrete data centers with redundant power, networking, and connectivity
- They're separate from each other, so that they're isolated from disasters. All traffic between AZ's is encrypted
- They're connected with high bandwidth, ultra-low latency networking.

AZ's are physically separated by a meaningful distance, many kilometers, from any other AZ, although all are within 100 km (60 miles) of each other.

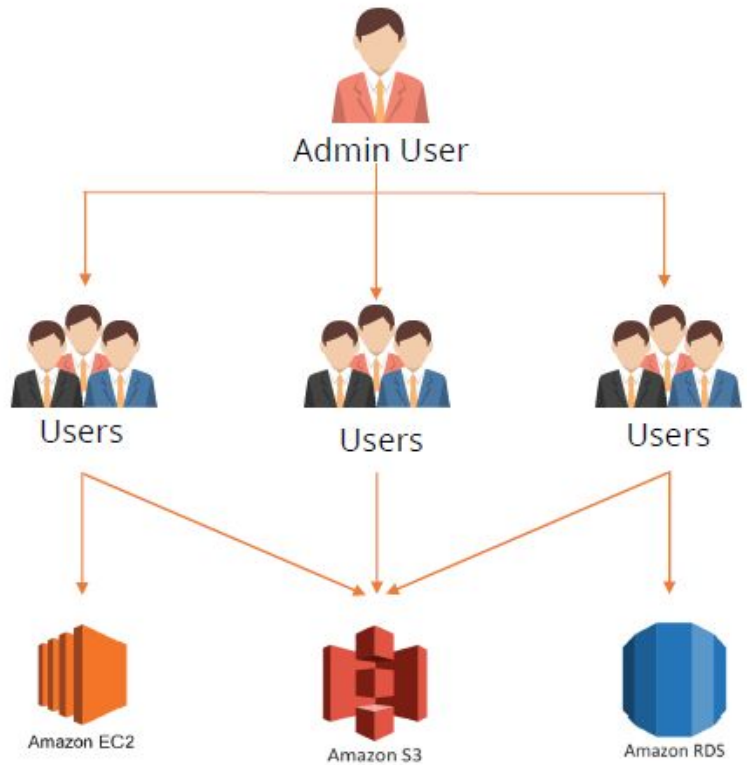
Identity and Access Management

The key features of IAM:



Shared Access

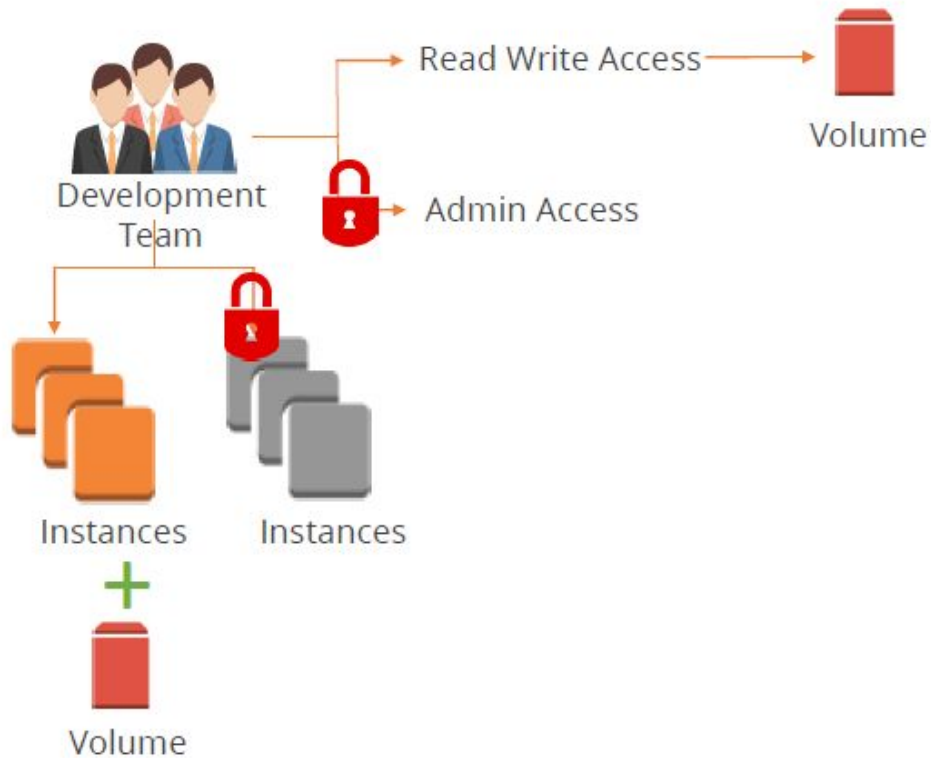
Grant permission to users to access and use resources in your AWS account without sharing your password.



Granular Permissions

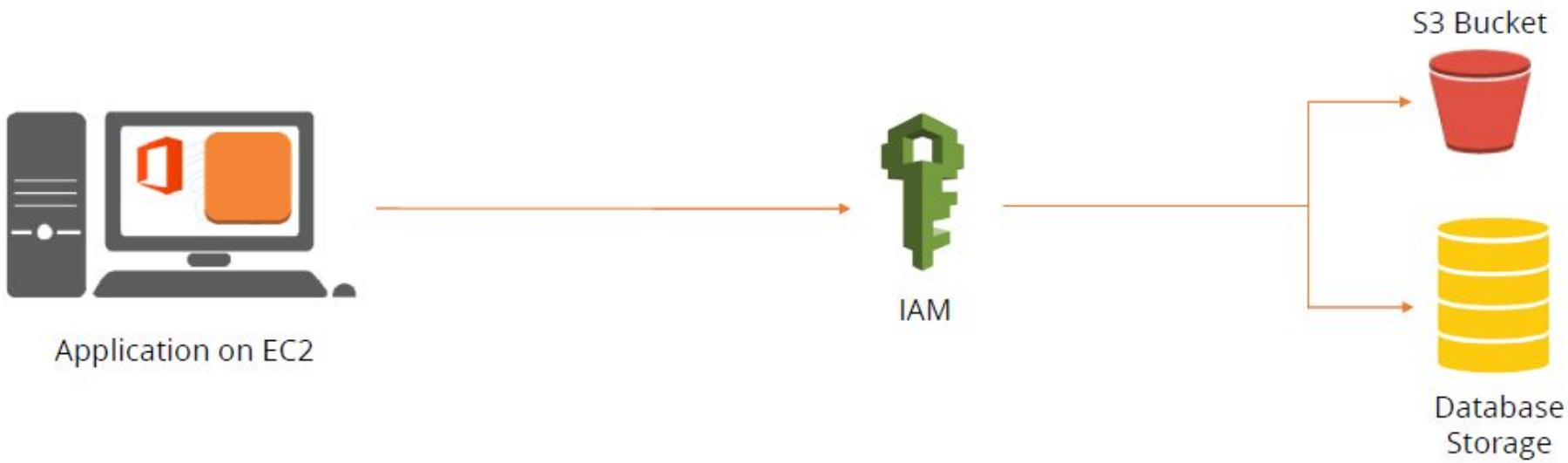
Granular permissions allow different permissions to various users to manage their access to AWS, such as:

- User access to specific services
- Specific permissions for actions
- Specific access to resources



Secure Access

Securely allocate credentials that applications on EC2 instances require to access other AWS resources.

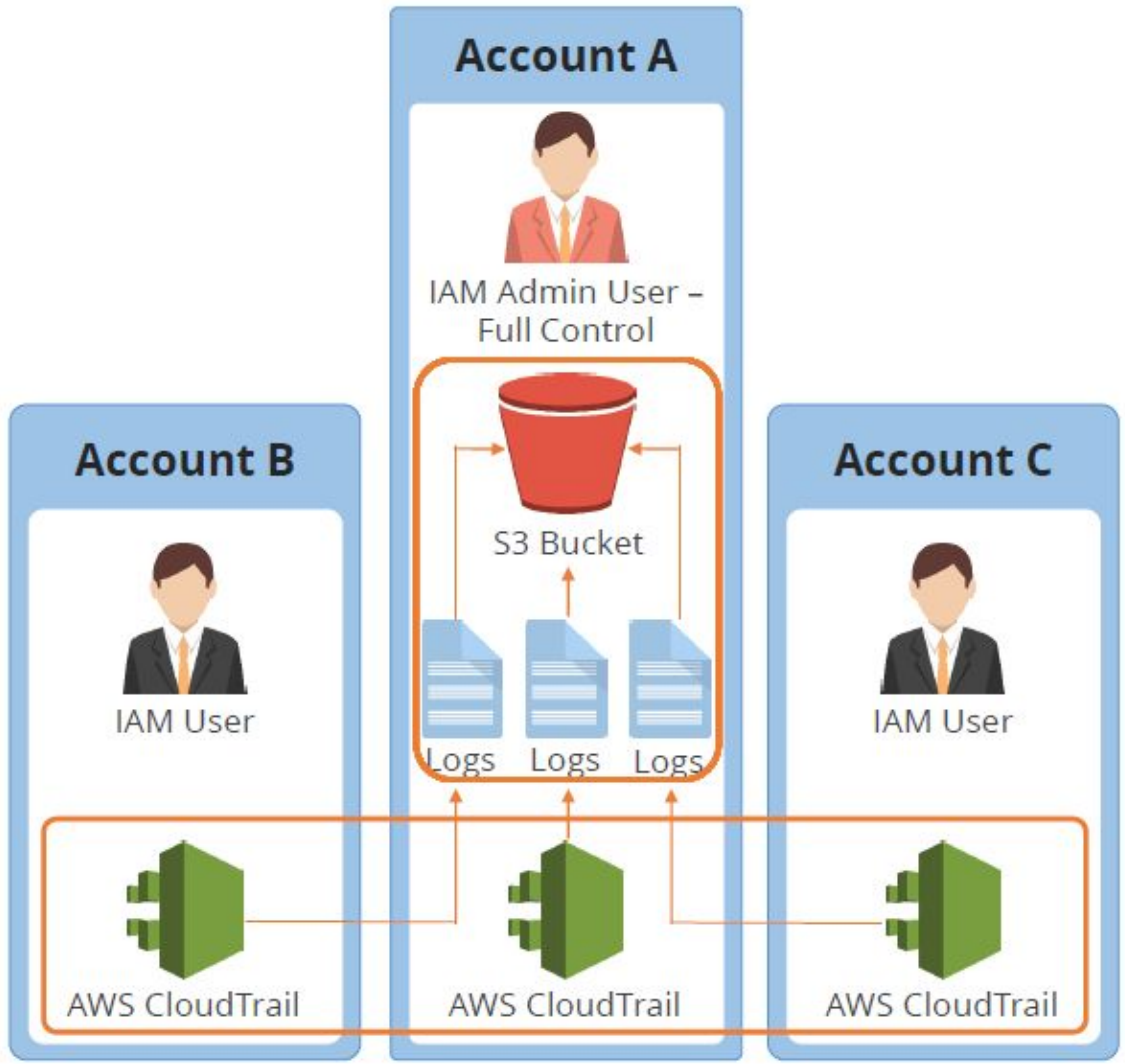


Identity Federation

Allows users with external accounts to get temporary access to AWS resources

Identity Information

Log, monitor, and track what users are doing with your AWS resources.



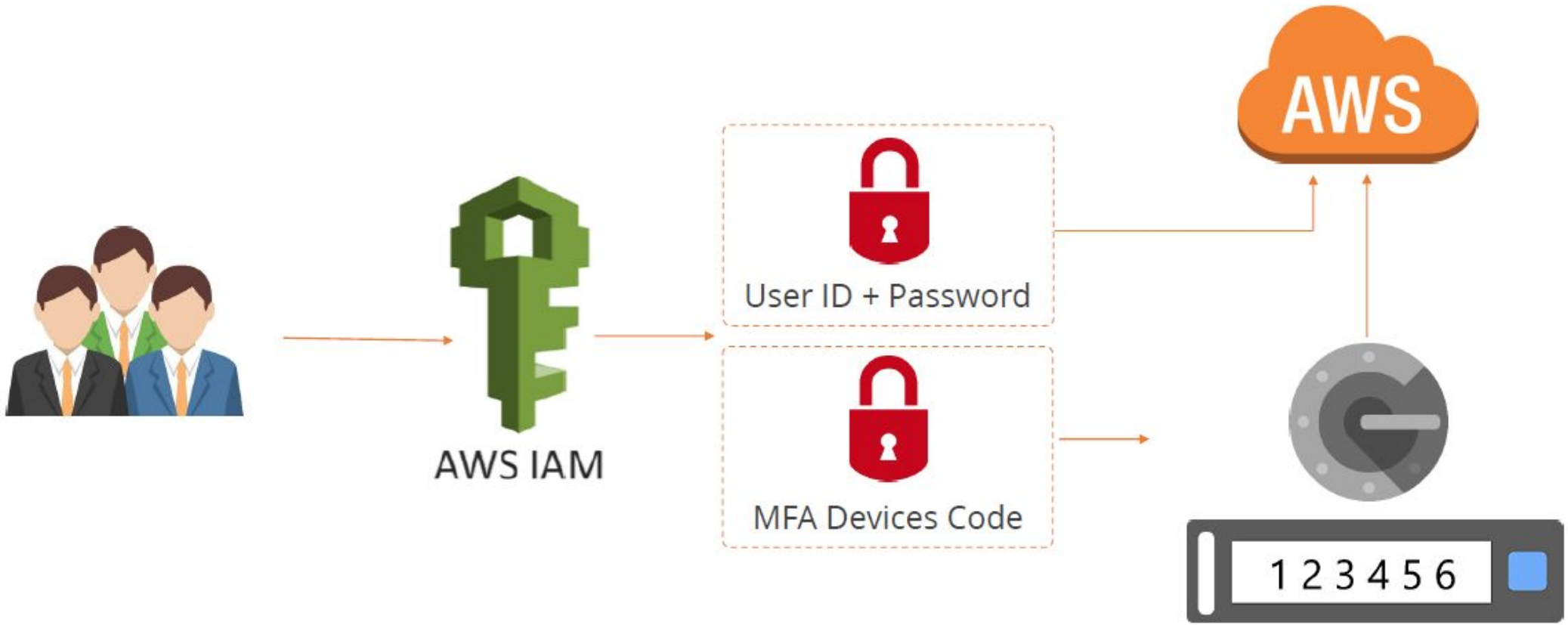
PCI DSS Compliance

Payment Card Industry (PCI) and Data Security Standard (DSS) compliant

PCI DSS COMPLIANT

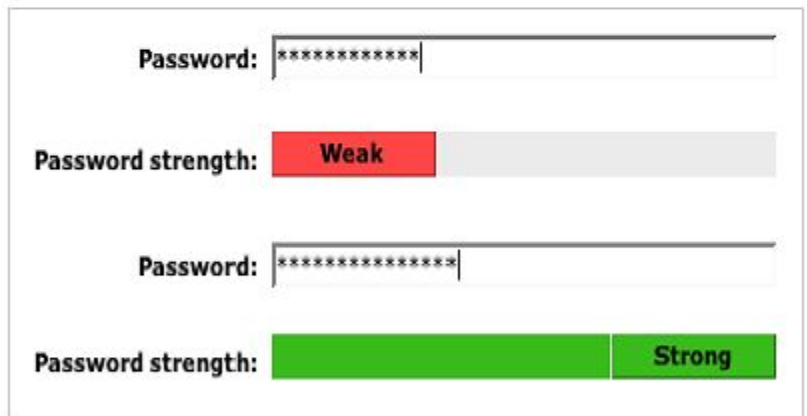
Multi-Factor Authentication

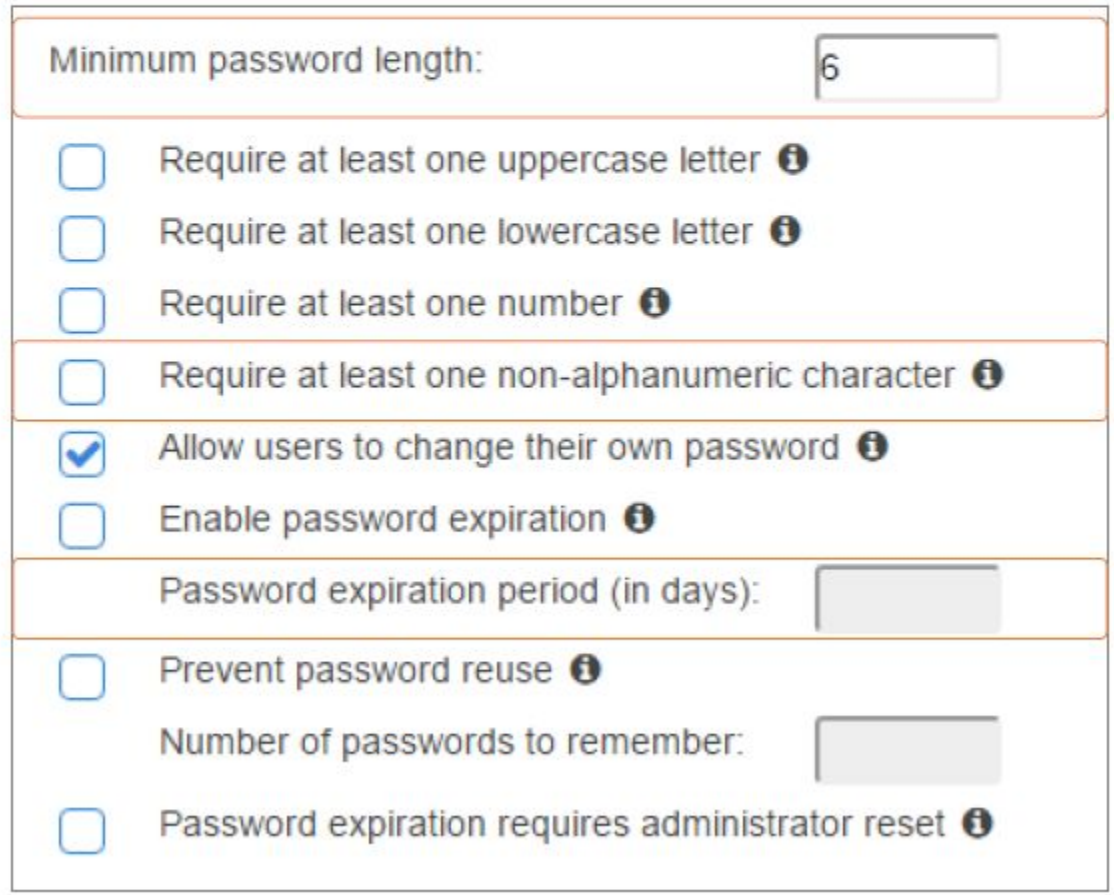
Two-Factor Authorization for users and resources to ensure absolute security using MFA devices



Password Policy

IAM allows you to define password strength and rotation policies.





LAB 1: Enabling MFA for root account and creating user, groups

Key take away from lab:
Root account should never be used (and shared)

- Users must be created with proper permissions
- IAM is at the center of AWS
- Policies are written in JSON (JavaScript Object Notation)
- IAM has a global view.
- It's best to give users the minimal amount of permissions they need to perform their job (least privilege principles)
- One can login into AWS using their company credentials. Identity Federation uses the SAML standard (Active Directory)

IAM credentials should NEVER BE SHARED

- Never, ever, ever, ever, write IAM credentials in code. EVER.
- NEVER EVER COMMIT YOUR IAM credentials
- Never use the ROOT account except for initial setup.
- Never use ROOT IAM Credentials

https://docs.aws.amazon.com/IAM/latest/UserGuide/best-practices.html

What is EC2?

Amazon Elastic Compute Cloud (Amazon EC2) is a web service that provides secure, resizable compute capacity in the cloud.

- EC2 is one of most popular of AWS offering
- It mainly consists in the capability of :
- Renting virtual machines (EC2)
- Storing data on virtual drives (EBS)
- Distributing load across machines (ELB)
- Scaling the services using an auto-scaling group (ASG)

Lab 2: Launching an EC2 Instance running Linux and SSH

Lab3 : Running Apache on EC2

SSH Summary Table

	SSH	Putty	EC2 Instance Connect
Mac	✓		✓
Linux	✓		✓
Windows < 10		✓	✓
Windows >= 10	✓	✓	✓

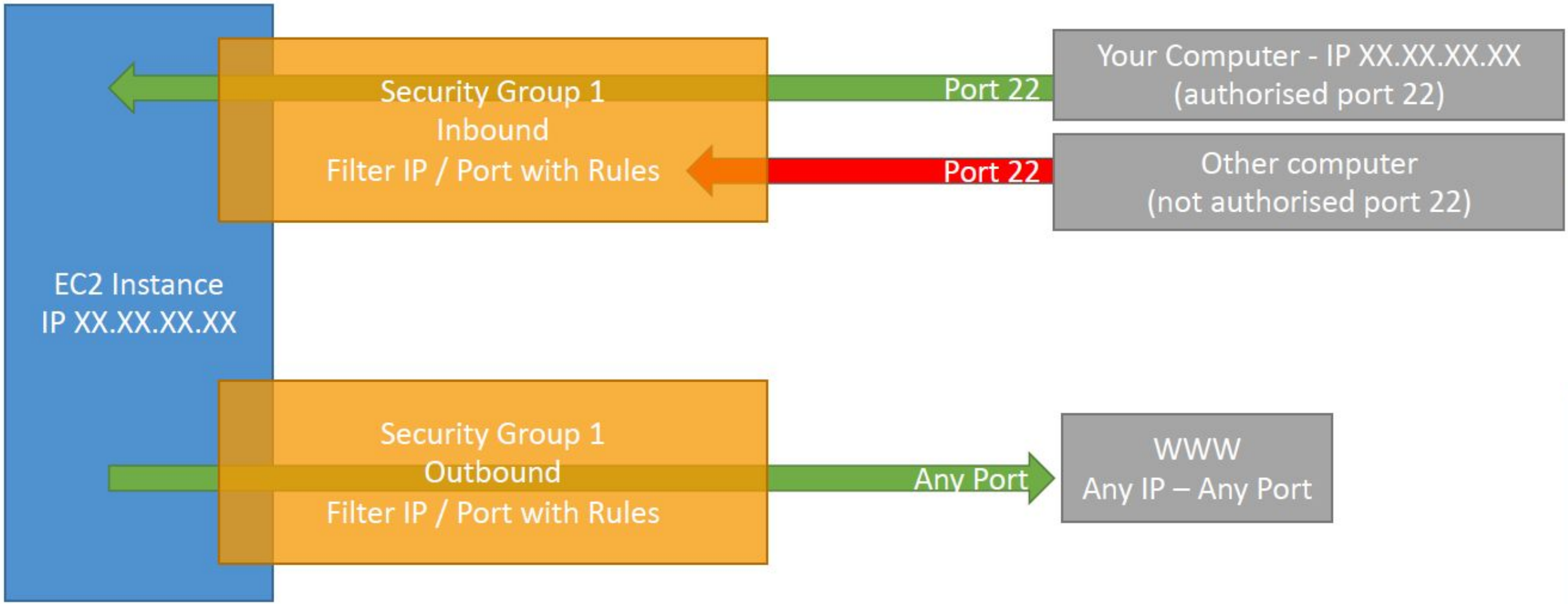
Introduction to Security Groups

- Security Groups are the fundamental of network security in AWS
- They control how traffic is allowed into or out of our EC2 Machines

- Security groups are acting as a "firewall" on EC2 instances
- They regulate:
- Access to Ports
- Authorized IP ranges – IPv4 and IPv6
- Control of inbound network (from other to the instance)
- Control of outbound network (from the instance to other)

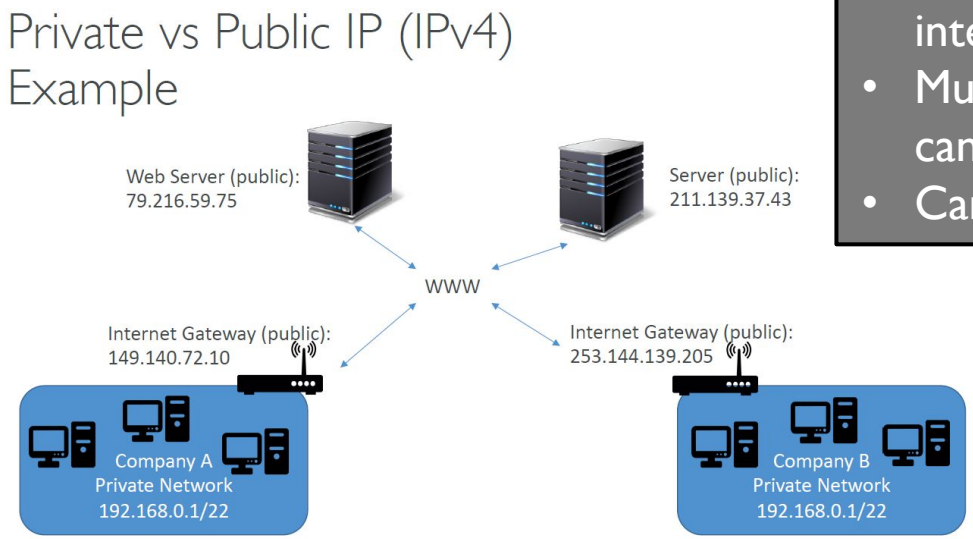
Security Groups Good to know

- Can be attached to multiple instances
- Locked down to a region / VPC combination
- It's good to maintain one separate security group for SSH access
- If your application is not accessible (time out), then it's a security group issue
- If your application gives a "connection refused" error, then it's an application error or it's not launched
- All inbound traffic is blocked by default
- All outbound traffic is authorized by default


EC2 Instance
IP XX.XX.XX.XX
Security Group 1
Inbound
Filter IP / Port with Rules
Port 22
Your Computer - IP XX.XX.XX.XX
(authorised port 22)
Port 22
Other computer
(not authorised port 22)
Security Group 1
Outbound
Filter IP / Port with Rules
Any Port
WWW
Any IP – Any Port

Private vs Public IP (IPv4)

- Networking has two sorts of IPs. IPv4 and IPv6:
- IPv4: 1.160.10.240 (common format used online)
- IPv6: 3ffe:1900:4545:3:200:f8ff:fe21:67cf(Internet of Things)



- Public IP means the machine can be identified on the internet (WWW)
- Must be unique across the whole web (not two machines can have the same public IP).
- Can be geo-located easily

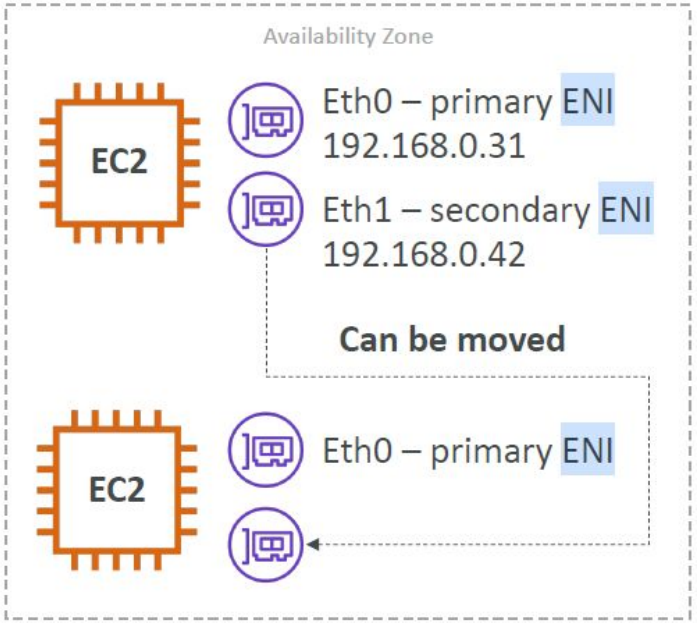
- Private IP means the machine can only be identified on a private network only
- The IP must be unique across the private network
- BUT two different private networks (two companies) can have the same IPs.
- Machines connect to WWW using a NAT + internet gateway (a proxy)
- Only a specified range of IPs can be used as private IP

Elastic IPs

- When you stop and then start an EC2 instance, it can change its public IP.
- If you need to have a fixed public IP for your instance, you need an Elastic IP
- An Elastic IP is a public IPv4 IP you own as long as you don’t delete it
- You can attach it to one instance at a time
- You can only have 5 Elastic IP in your account

Elastic Network Interfaces (ENI)

- Logical component in a VPC that represents a virtual network card
- The ENI can have the following attributes:
    - Primary private IPv4, one or more secondary IPv4
    - One Elastic IP (IPv4) per private IPv4
    - One Public IPv4
    - One or more security groups
    - A MAC address
- You can create ENI independently and attach them on the fly (move them) on EC2 instances for failover
- Bound to a specific availability zone (AZ)



EC2 User Data

- It is possible to bootstrap our instances using an EC2 User data script.
- Bootstrapping means launching commands when a machine starts
- That script is only run once at the instance first start
- EC2 user data is used to automate boot tasks such as:
- Installing updates
- Installing software
- Downloading common files from the internet
- The EC2 User Data Script runs with the root user

Lab 4 : EC2 User Data Hands-On. Install Apache HTTP server using user data

Knowledge Check

EC2 Instance Launch Types

EC2 On Demand

- Pay for what you use (billing per second, after the first minute)
- Has the highest cost but no upfront payment
- No long term commitment
- Recommended for short-term and un-interrupted workloads, where you can't predict how the application will behave.

EC2 Reserved Instances

- Up to 75% discount compared to On-demand
- Pay upfront for what you use with long term commitment
- Reservation period can be 1 or 3 years
- Reserve a specific instance type
- Recommended for steady state usage applications (database)
    - Convertible Reserved Instance
        - can change the EC2 instance type
        - Up to 54% discount
    - Scheduled Reserved Instances
        - launch within time window you reserve
        - When you require a fraction of day / week / month

EC2 Spot Instances

- Can get a discount of up to 90% compared to On-demand
- Instances that you can “lose” at any point of time if your max price is less than the current spot price
- The MOST cost-efficient instances in AWS
- Useful for workloads that are resilient to failure
- Batch jobs
- Data analysis
- Image processing
- Not great for critical jobs or databases

EC2 Dedicated Hosts

- Physical dedicated EC2 server for your use
- Full control of EC2 Instance placement
- Visibility into the underlying sockets / physical cores of the hardware
- Allocated for your account for a 3 year period reservation
- More expensive
- Useful for software that have complicated licensing model (BYOL – Bring Your Own License)
- Or for companies that have strong regulatory or compliance needs

EC2 Dedicated Instances

- Instances running on hardware that's dedicated to you
- May share hardware with other instances in same account
- No control over instance placement (can move hardware after Stop / Start)

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

EC2 Pricing Price : https://aws.amazon.com/ec2/pricing/on-demand/

Instances have 5 distinct characteristics advertised on the website:

- The RAM (type, amount, generation)
- The CPU (type, make, frequency, generation, number of cores)
- The I/O (disk performance, EBS optimisations)
- The Network (network bandwidth, network latency)
- The Graphical Processing Unit (GPU)

What's an AMI?

- As we saw, AWS comes with base images such as:
- Ubuntu
- Fedora
- RedHat
- Windows
- These images can be customized at runtime using EC2 User data

Custom AMI

- Using a custom built AMI can provide the following advantages:
- Pre-installed packages needed
- Faster boot time (no need for long ec2 user data at boot time)
- Machine comes configured with monitoring / enterprise software
- Security concerns – control over the machines in the network
- Control of maintenance and updates of AMIs over time
- Active Directory Integration out of the box
- Installing your app ahead of time (for faster deploys when auto-scaling)
- Using someone else's AMI that is optimized for running an app, DB, etc…
- AMI are built for a specific AWS region (!)

Load Balancing, Auto Scaling Groups and EBS Volumes

Scalability means that an application / system can handle greater loads by adapting.

- There are two kinds of scalability:
- Vertical Scalability : RDS, ElastiCache are services that can scale vertically.
- Horizontal Scalability : Amazon EC2

High Availability

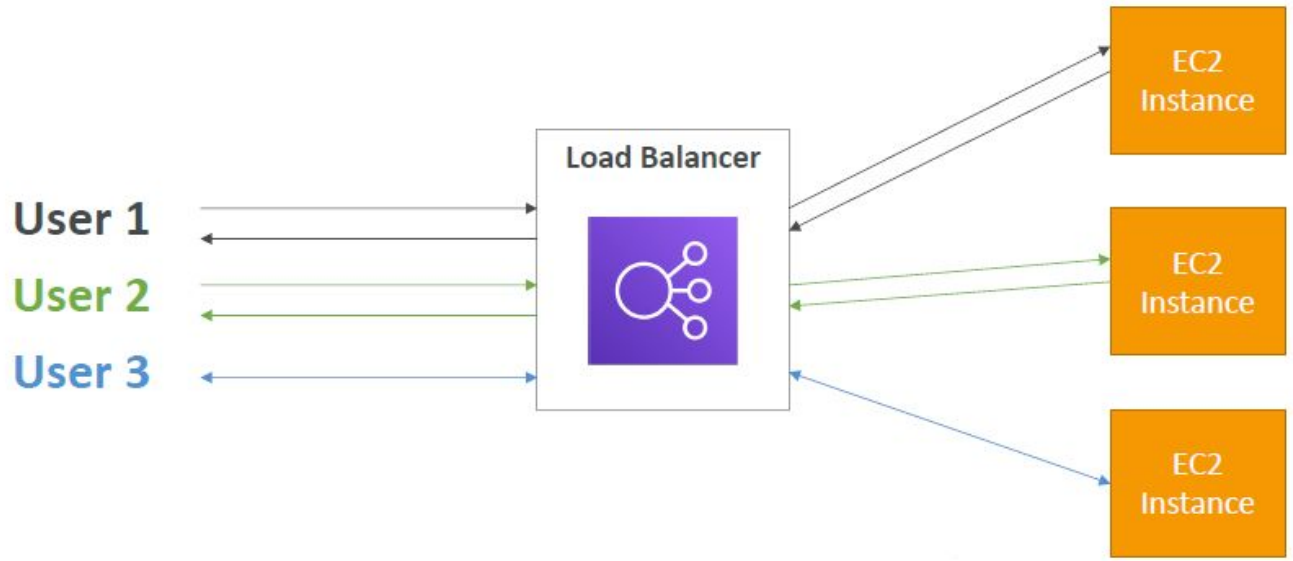
- High Availability usually goes hand in hand with horizontal scaling
- High availability means running your application / system in at least 2 data centers (== Availability Zones)
- The goal of high availability is to survive a data center loss.

High Availability & Scalability For EC2

- Vertical Scaling: Increase instance size (= scale up / down)
    - From: t2.nano - 0.5G of RAM, 1 vCPU
    - To: u-12tb1.metal – 12.3 TB of RAM, 448 vCPUs
- Horizontal Scaling: Increase number of instances (= scale out / in)
    - Auto Scaling Group
    - Load Balancer
- High Availability: Run instances for the same application across multi AZ
    - Auto Scaling Group multi AZ
    - Load Balancer multi AZ

What is load balancing?

- Load balancers are servers that forward internet traffic to multiple servers (EC2 Instances)



Why use a load balancer?

- Spread load across multiple downstream instances
- Expose a single point of access (DNS) to your application
- Seamlessly handle failures of downstream instances
- Do regular health checks to your instances
- Provide SSL termination (HTTPS) for your websites
- Enforce stickiness with cookies
- High availability across zones
- Separate public traffic from private traffic

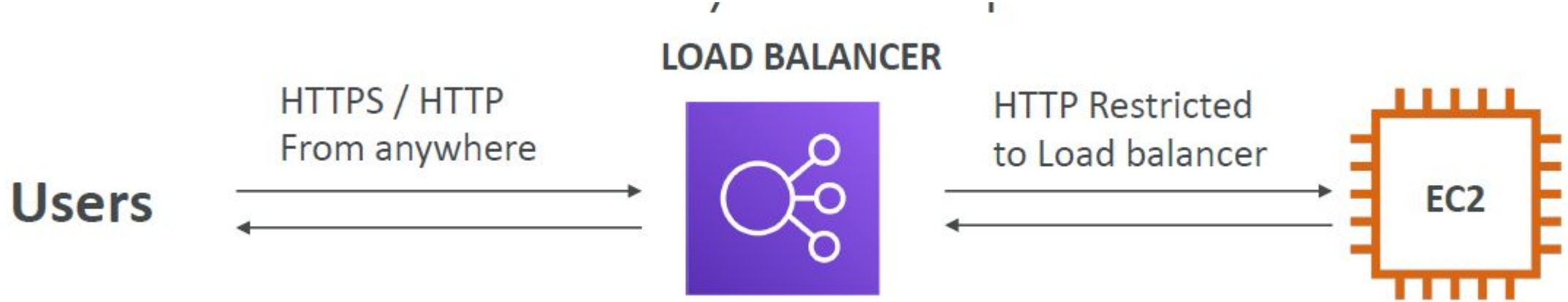
Types of load balancer on AWS

Classic Load Balancer (v1 - old generation) – 2009

- HTTP, HTTPS, TCP

- Application Load Balancer (v2 - new generation) – 2016
  - HTTP, HTTPS, WebSocket
- Network Load Balancer (v2 - new generation) – 2017
  - TCP, TLS (secure TCP) & UDP
- Overall, it is recommended to use the newer / v2 generation load balancers as they provide more features

Load Balancer Security Groups



Load Balancer Security Group:

Type	Protocol	Port Range	Source	Description
HTTP	TCP	80	0.0.0.0/0	Allow HTTP from an...
HTTPS	TCP	443	0.0.0.0/0	Allow HTTPS from a...

Application Security Group: Allow traffic only from Load Balancer

Type	Protocol	Port Range	Source	Description
HTTP	TCP	80	sg-054b5ff5ea02f2b6e (load-b	Allow Traffic only...

Health Checks

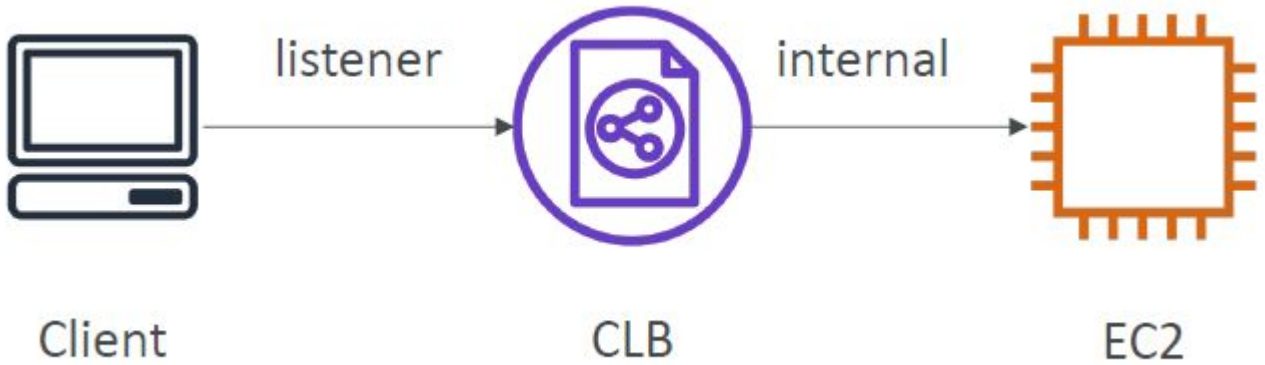
- Health Checks are crucial for Load Balancers
- They enable the load balancer to know if instances it forwards traffic to are available to reply to requests
- The health check is done on a port and a route (/health is common)
- If the response is not 200 (OK), then the instance is unhealthy

Load Balancer Good to Know

- LBs can scale but not instantaneously – contact AWS for a “warm-up”
- Troubleshooting
- 4xx errors are client induced errors
- 5xx errors are application induced errors
- Load Balancer Errors 503 means at capacity or no registered target
- If the LB can’t connect to your application, check your security groups!
- Monitoring
- ELB access logs will log all access requests (so you can debug per request)
- CloudWatch Metrics will give you aggregate statistics (ex: connections count)

Classic Load Balancers (v1)

- Supports TCP (Layer 4), HTTP & HTTPS (Layer 7)
- Health checks are TCP or HTTP based
- Fixed hostname
  XXX.region.elb.amazonaws.com



Lab 5 :Launch classic load balancer

Application Load Balancer (v2)

- Application load balancers is Layer 7 (HTTP)
- Load balancing to multiple HTTP applications across machines (target groups)
- Load balancing to multiple applications on the same machine (ex: containers)
- Support for HTTP/2 and WebSocket
- Support redirects (from HTTP to HTTPS for example)
- Routing tables to different target groups:
    - Routing based on path in URL (example.com/users & example.com/posts)
    - Routing based on hostname in URL (one.example.com & other.example.com)
    - Routing based on Query String, Headers(example.com/users?id=123&order=false)
- **ALB are a great fit for micro services & container-based application (example: Docker & Amazon ECS)**
- Has a port mapping feature to redirect to a dynamic port in ECS
- In comparison, we'd need multiple Classic Load Balancer per application

Application Load Balancer (v2)

Target Groups

- EC2 instances (can be managed by an Auto Scaling Group) – HTTP
- ECS tasks (managed by ECS itself) – HTTP
- Lambda functions – HTTP request is translated into a JSON event
- IP Addresses – must be private IPs
- ALB can route to multiple target groups
- Health checks are at the target group level

Good to Know

- Fixed hostname (XXX.region.elb.amazonaws.com)
- The application servers don't see the IP of the client directly
- The true IP of the client is inserted in the header X-Forwarded-For
- We can also get Port (X-Forwarded-Port) and proto (X-Forwarded-Proto)

Lab 6 : ALB hands on

Network Load Balancer (v2)

- Network load balancers (Layer 4) allow to:
- Forward TCP & UDP traffic to your instances
- Handle millions of request per seconds
- Less latency ~100 ms (vs 400 ms for ALB)
- NLB has one static IP per AZ, and supports assigning Elastic IP(helpful for whitelisting specific IP)
- NLB are used for extreme performance, TCP or UDP traffic
- Not included in the AWS free tier

Target Type:
Instance
IP

Load Balancer Stickiness

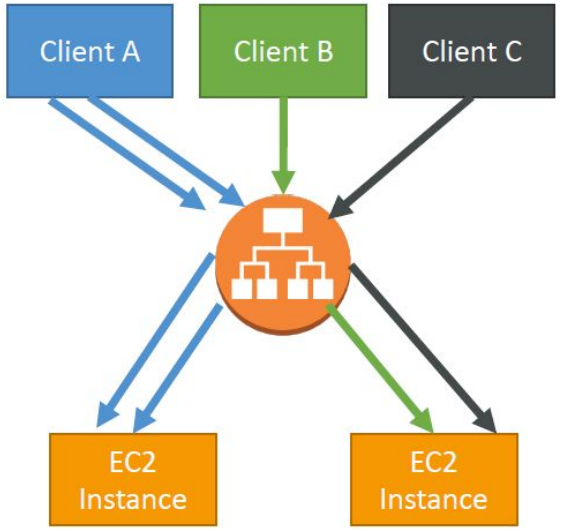
• It is possible to implement stickiness so that the same client is always redirected to the same instance behind a load balancer

• This works for Classic Load Balancers & Application Load Balancers

• The “cookie” used for stickiness has an expiration date you control

• Use case: make sure the user doesn’t lose his session data

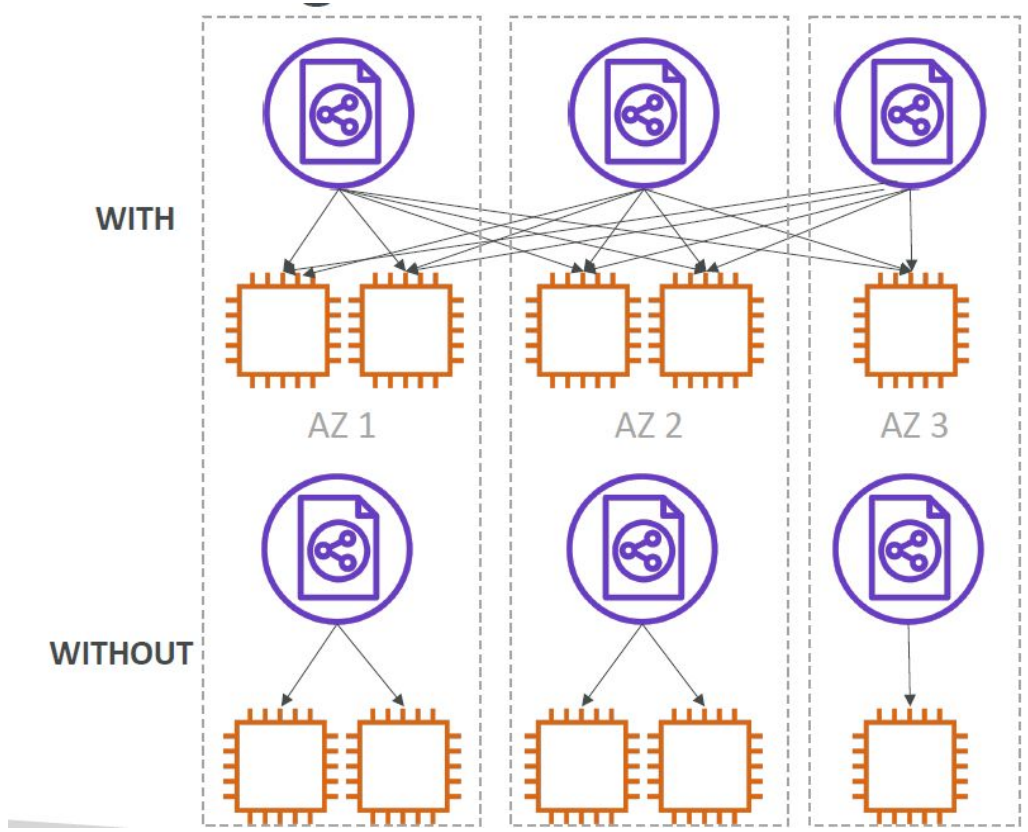
• Enabling stickiness may bring imbalance to the load over the backend EC2 instances



Cross-Zone Load Balancing

• With Cross Zone Load Balancing: each load balancer instance distributes evenly across all registered instances in all AZ

• Otherwise, each load balancer node distributes requests evenly across the registered instances in its Availability Zone only.



- ❑ Classic Load Balancer
  - • Disabled by default
  - • No charges for inter AZ data if enabled
- ❑ Application Load Balancer
  - • Always on (can't be disabled)
  - • No charges for inter AZ data
- ❑ Network Load Balancer
  - • Disabled by default
  - • You pay charges ($) for inter AZ data if enabled

SSL/TLS - Basics

- An SSL Certificate allows traffic between your clients and your load balancer to be encrypted in transit (in-flight encryption)
- SSL refers to Secure Sockets Layer, used to encrypt connections
- TLS refers to Transport Layer Security, which is a newer version
- Nowadays, TLS certificates are mainly used, but people still refer as SSL
- Public SSL certificates are issued by Certificate Authorities (CA)- **Comodo, Symantec, GoDaddy, GlobalSign, Digicert, Letsencrypt, etc**
- SSL certificates have an expiration date (you set) and must be renewed

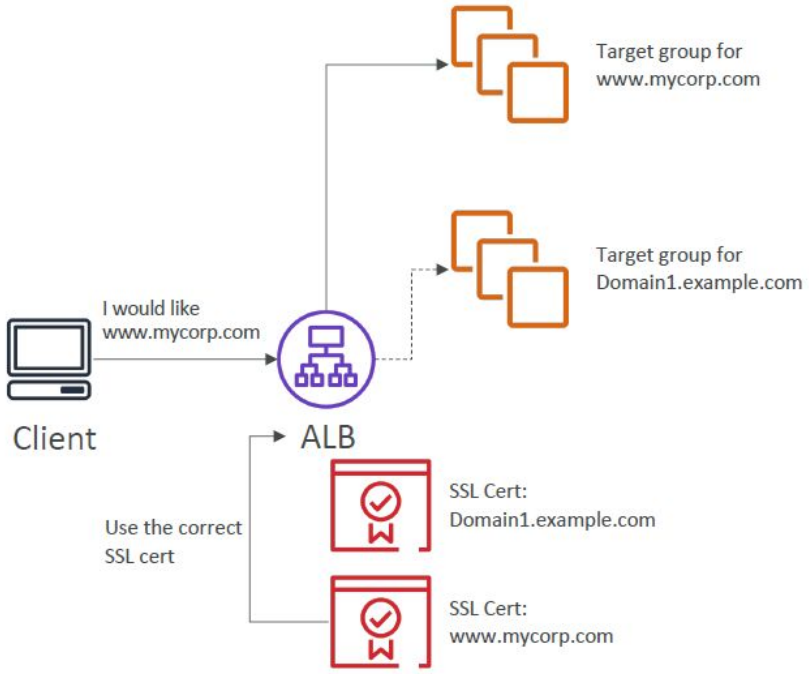
- The load balancer uses an X.509 certificate (SSL/TLS server certificate)
- You can manage certificates using ACM (AWS Certificate Manager)
- You can create upload your own certificates alternatively
- HTTPS listener:
    - You must specify a default certificate
    - You can add an optional list of certs to support multiple domains
- Clients can use SNI (Server Name Indication) to specify the hostname they reach

SSL – Server Name Indication (SNI)

- SNI solves the problem of loading multiple SSL certificates onto one web server (to serve multiple websites)
- It's a "newer" protocol, and requires the client to indicate the hostname of the target server in the initial SSL handshake and the server will then find the correct.

Note:

- Only works for ALB & NLB (newer generation), CloudFront
- Does not work for CLB (older gen)

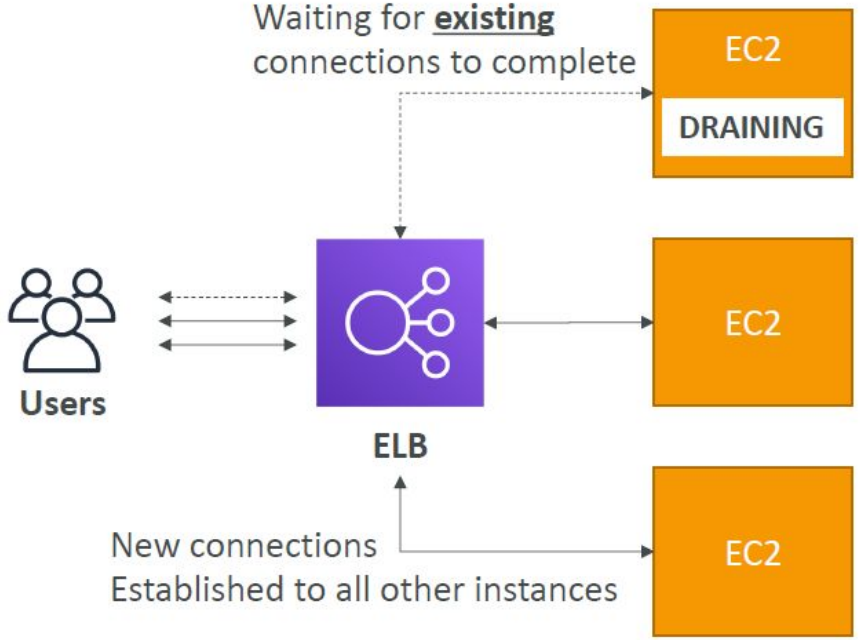


Elastic Load Balancers – SSL Certificates

- Classic Load Balancer (v1)
    - Support only one SSL certificate
    - Must use multiple CLB for multiple hostname with multiple SSL certificates
- Application Load Balancer (v2)
    - Supports multiple listeners with multiple SSL certificates
    - Uses Server Name Indication (SNI) to make it work
- Network Load Balancer (v2)
    - Supports multiple listeners with multiple SSL certificates
    - Uses Server Name Indication (SNI) to make it work

ELB – Connection Draining

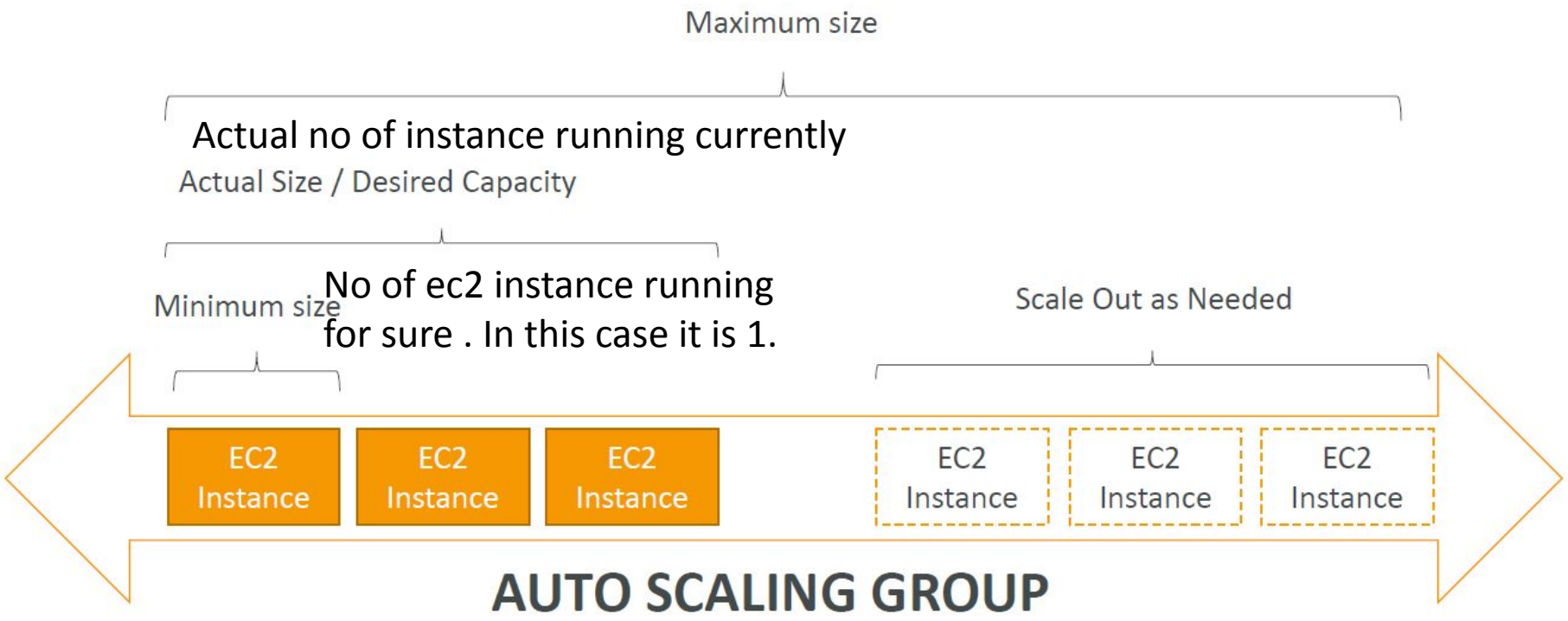
- Feature naming:
  - CLB: Connection Draining
  - Target Group: Deregistration Delay (for ALB & NLB)
- Time to complete “in-flight requests” while the instance is de-registering or unhealthy
- Stops sending new requests to the instance which is de-registering
- Between 1 to 3600 seconds, default is 300 seconds
- Can be disabled (set value to 0)
- Set to a low value if your requests are short



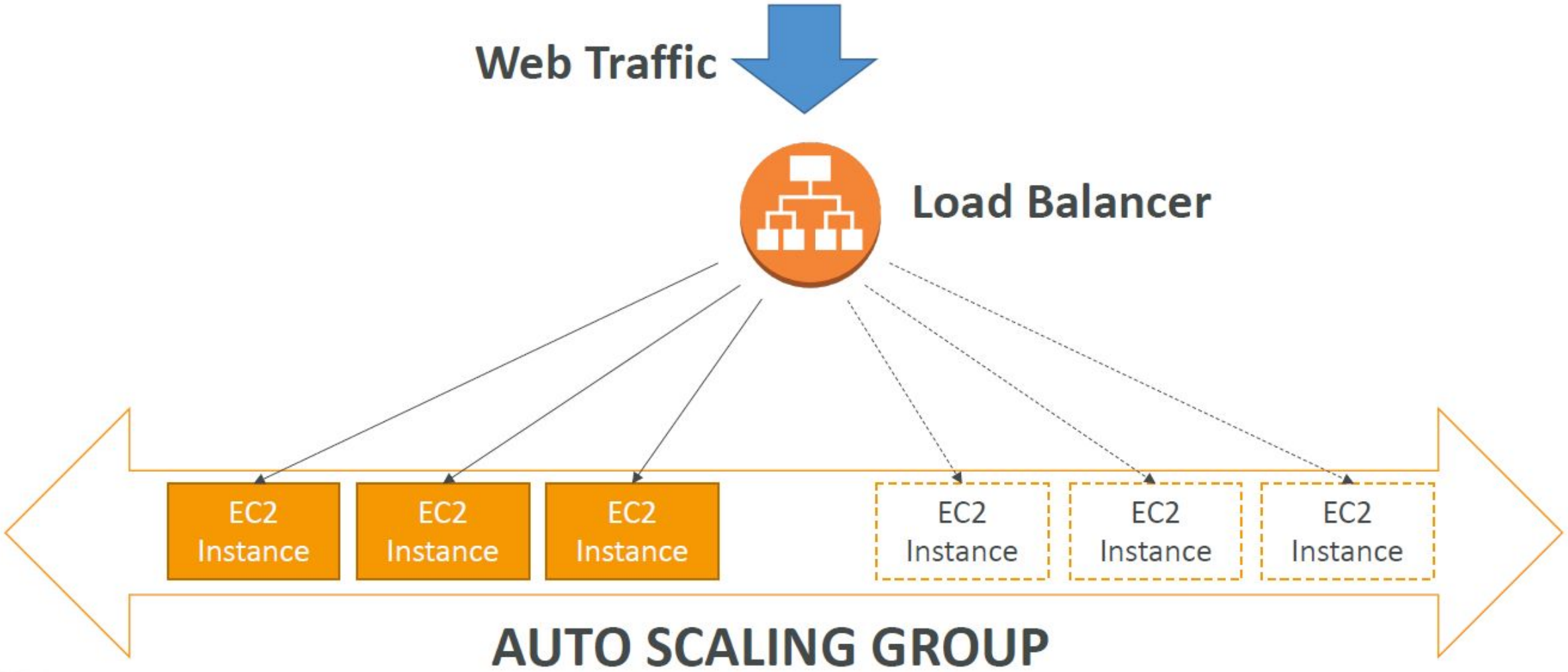
What's an Auto Scaling Group

The goal of an Auto Scaling Group (ASG) is to:

- Scale out (add EC2 instances) to match an increased load
- Scale in (remove EC2 instances) to match a decreased load
- Ensure we have a minimum and a maximum number of machines running
- Automatically Register new instances to a load balancer



Auto Scaling Group in AWS With Load Balancer



ASGs have the following attributes

- A launch configuration
- AMI + Instance Type
- EC2 User Data
- EBS Volumes
- Security Groups
- SSH Key Pair
- Min Size / Max Size / Initial Capacity
- Network + Subnets Information
- Load Balancer Information
- Scaling Policies

Auto Scaling Alarms

- It is possible to scale an ASG based on CloudWatch alarms
- An Alarm monitors a metric (such as Average CPU)
- Metrics are computed for the overall ASG instances
- Based on the alarm:
    - We can create scale-out policies (increase the number of instances)
    - We can create scale-in policies (decrease the number of instances)

Auto Scaling New Rules

- It is now possible to define ”better” auto scaling rules that are directly managed by EC2
- Target Average CPU Usage
- Number of requests on the ELB per instance
- Average Network In
- Average Network Out

Auto Scaling Custom Metric

- We can auto scale based on a custom metric (ex: number of connected users)
- 1. Send custom metric from application on EC2 to CloudWatch (PutMetric API)
- 2. Create CloudWatch alarm to react to low / high values
- 3. Use the CloudWatch alarm as the scaling policy for ASG

ASG Good to know

- Scaling policies can be on CPU, Network and can even be on custom metrics or based on a schedule
- ASGs use Launch configurations or Launch Templates (newer)
- To update an ASG, you must provide a new launch configuration / launch template
- IAM roles attached to an ASG will get assigned to EC2 instances
- ASG are free. You pay for the underlying resources being launched
- Having instances under an ASG means that if they get terminated for whatever reason, the ASG will automatically create new ones as a replacement. Extra safety!
- ASG can terminate instances marked as unhealthy by an LB (and hence replace them)

Auto Scaling Groups – Scaling Policies

Scaling Policy	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

Scheduled Actions

- Anticipate a scaling based on known usage patterns
- Example: increase the min capacity to 10 at 5 pm on Fridays

https://docs.aws.amazon.com/autoscaling/ec2/userguide/as-scaling-simple-step.html

Cooldown periods help to prevent the initiation of additional scaling activities before the effects of previous activities are visible.

Lab 7: ALB with scaling policy

https://docs.aws.amazon.com/autoscaling/ec2/userguide/common-scenarios-termination.html

EC2 Storage : EBS

What's an EBS Volume?

- An EC2 machine loses its root volume (main drive) when it is manually terminated.
- Unexpected terminations might happen from time to time.
- Sometimes, you need a way to store your instance data somewhere
- An EBS (Elastic Block Store) Volume is a network drive you can attach to your instances while they run
- It allows your instances to persist data

Good to know

- It's a network drive (i.e. not a physical drive)
    - It uses the network to communicate the instance, which means there might be a bit of latency
    - It can be detached from an EC2 instance and attached to another one quickly
- It's locked to an Availability Zone (AZ)
    - An EBS Volume in us-east-1a cannot be attached to us-east-1b
    - To move a volume across, you first need to snapshot it
- Have a provisioned capacity (size in GBs, and IOPS)
    - You get billed for all the provisioned capacity
    - You can increase the capacity of the drive over time

EBS Volume Types

- EBS Volumes come in 4 types

❑ GP2 (from AWS doc)

- Recommended for most workloads
- System boot volumes
- Virtual desktops
- Low-latency interactive apps
- Development and test environments
- 1 GiB - 16 TiB
- Small gp2 volumes can burst IOPS to 3000
- Max IOPS is 16,000…
- 3 IOPS per GB, means at 5,334GB we are at the max IOPS

- EBS Volumes are characterized in Size | Throughput | IOPS (I/O Ops Per Sec)

Only GP2 and IO1 can be used as boot volumes

❑ IO1 (from AWS doc)

- Critical business applications that require sustained IOPS performance, or more than 16,000 IOPS per volume (gp2 limit)
- Large database workloads, such as:
- MongoDB, Cassandra, Microsoft SQL Server, MySQL, PostgreSQL, Oracle
- 4 GiB - 16 TiB
- IOPS is provisioned (PIOPS) – MIN 100 - MAX 64,000 (Nitro instances) else MAX 32,000 (other instances)
- The maximum ratio of provisioned IOPS to requested volume size (in GiB) is 50:1

❑ ST1 (from AWS doc)

- Streaming workloads requiring consistent, fast throughput at a low price.
- Big data, Data warehouses, Log processing
- Apache Kafka
- Cannot be a boot volume
- 500 GiB - 16 TiB
- Max IOPS is 500
- Max throughput of 500 MiB/s – can burst

❑ SC1 (from AWS doc)

- Throughput-oriented storage for large volumes of data that is infrequently accessed
- Scenarios where the lowest storage cost is important
- Cannot be a boot volume
- 500 GiB - 16 TiB
- Max IOPS is 250
- Max throughput of 250 MiB/s – can burst

Instance store

An *instance store* provides temporary block-level storage for your instance.

Some instance do not come with Root EBS volumes

- Instead, they come with “Instance Store” (= ephemeral storage)
- Instance store is physically attached to the machine (EBS is a network drive)

- Pros:
  - Better I/O performance (EBS gp2 has an max IOPS of 16000, io1 of 64000)
  - Good for buffer / cache / scratch data / temporary content
  - Data survives reboots
- Cons:
  - On stop or termination, the instance store is lost
  - You can’t resize the instance store
  - Backups must be operated by the user

Very high IOPS

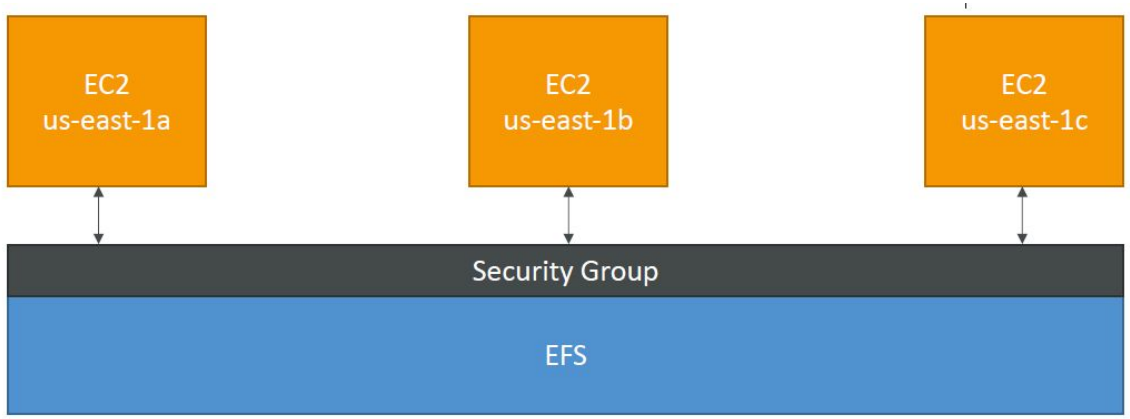
Instance Size	100% Random Read IOPS	Write IOPS
i3.large *	100,125	35,000
i3.xlarge *	206,250	70,000
i3.2xlarge	412,500	180,000
i3.4xlarge	825,000	360,000
i3.8xlarge	1.65 million	720,000
i3.16xlarge	3.3 million	1.4 million
i3.metal	3.3 million	1.4 million
i3en.large *	42,500	32,500
i3en.xlarge *	85,000	65,000
i3en.2xlarge *	170,000	130,000
i3en.3xlarge	250,000	200,000
i3en.6xlarge	500,000	400,000
i3en.12xlarge	1 million	800,000
i3en.24xlarge	2 million	1.6 million
i3en.metal	2 million	1.6 million

Good to know : Very High IOPS. Disks up to 7.5 TiB (can change over time), stripped to reach 30TiB

Only the following instance types support an instance store volume as the root device: C3, D2, G2, I2, M3, and R3.

EFS – Elastic File System

- Managed NFS (network file system) that can be mounted on many EC2
- EFS works with EC2 instances in multi-AZ
- Highly available, scalable, expensive (3x gp2), pay per use



Use cases: content management, web serving, data sharing, Wordpress

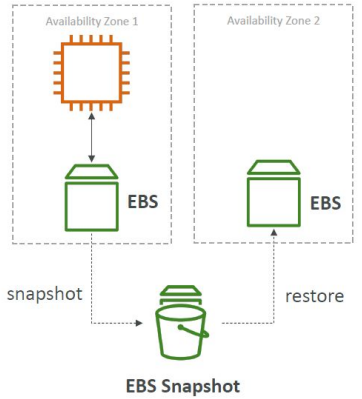
- Uses NFSv4.1 protocol
- Uses security group to control access to EFS
- **Compatible with Linux based AMI (not Windows)**
- Encryption at rest using KMS
- POSIX file system (~Linux) that has a standard file API
- File system scales automatically, pay-per-use, no capacity planning!

EFS – Performance & Storage Classes

- EFS Scale
  - 1000s of concurrent NFS clients, 10 GB+ /s throughput
  - Grow to Petabyte-scale network file system, automatically
  - Performance mode (set at EFS creation time)
- General purpose (default): latency-sensitive use cases (web server, CMS, etc…)
  - Max I/O – higher latency, throughput, highly parallel (big data, media processing)
- Storage Tiers (lifecycle management feature – move file after N days)
  - Standard: for frequently accessed files
  - Infrequent access (EFS-IA): cost to retrieve files, lower price to store

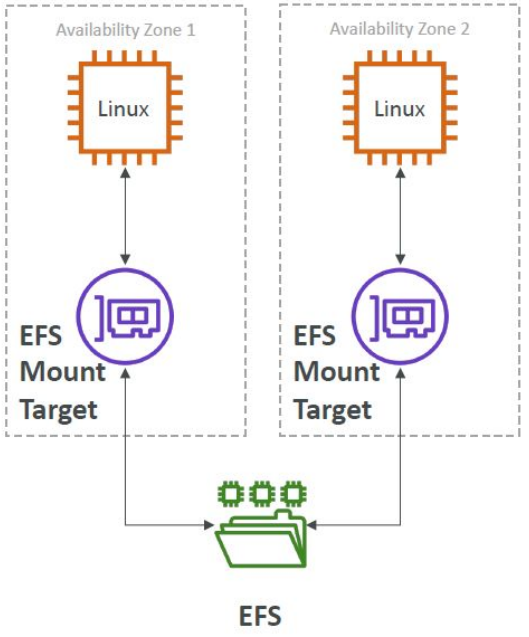
EBS vs EFS – Elastic Block Storage

- EBS volumes…
  - can be attached to only one instance at a time
  - are locked at the Availability Zone (AZ) level
  - gp2: IO increases if the disk size increases
  - io1: can increase IO independently
- To migrate an EBS volume across AZ
  - Take a snapshot
  - Restore the snapshot to another AZ
  - EBS backups use IO and you shouldn't run them while your application is handling a lot of traffic
- Root EBS Volumes of instances get terminated by default if the EC2 instance gets terminated.



EFS

- Mount 100s of instances across AZ
- EFS share website files (WordPress)
- Only for Linux Instances (POSIX)
- EFS has a higher price point than EBS
- Can leverage EFS-IA for cost savings
- **Remember: EFS vs EBS vs Instance Store**



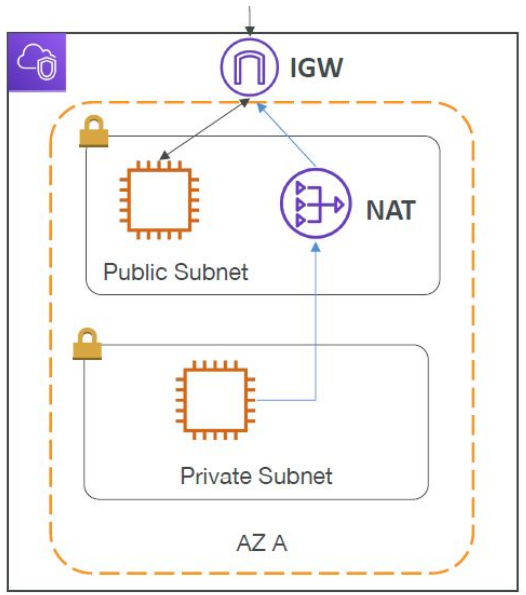
Lab 8 : Create an EBS volume, Snapshot

VPC & Subnets

- VPC: private network to deploy your resources (regional resource)
- Subnets allow you to partition your network inside your VPC(Availability Zone resource)
- A public subnet is a subnet that is accessible from the internet
- A private subnet is a subnet that is not accessible from the internet
- To define access to the internet and between subnets, we use Route Tables.

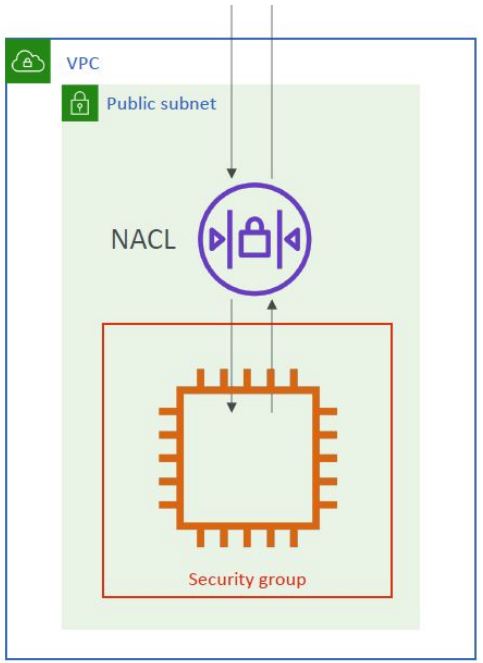
Internet Gateway & NAT Gateways

- Internet Gateways helps our VPC instances connect with the internet
- Public Subnets have a route to the internet gateway.
- NAT Gateways (AWS-managed) & NAT Instances (self-managed) allow your instances in your Private Subnets to access the internet while remaining private



Network ACL & Security Groups

- NACL (Network ACL)
  - A firewall which controls traffic from and to subnet
  - Can have ALLOW and DENY rules
  - Are attached at the Subnet level
  - Rules only include IP addresses
- Security Groups
  - A firewall that controls traffic to and from an ENI / an EC2 Instance
  - Can have only ALLOW rules
  - Rules include IP addresses and other security groups



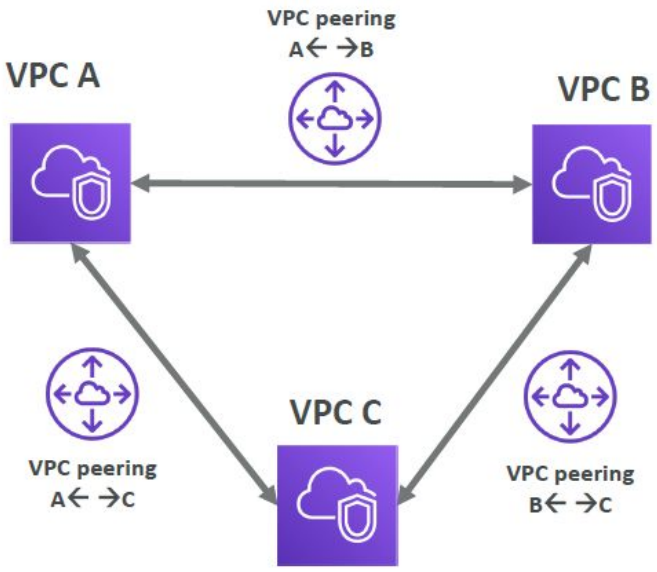
Security Group	Network ACL
Operates at the instance level	Operates at the subnet level
Supports allow rules only	Supports allow rules and deny rules
Is stateful: Return traffic is automatically allowed, regardless of any rules	Is stateless: Return traffic must be explicitly allowed by rules
We evaluate all rules before deciding whether to allow traffic	We process rules in number order when deciding whether to allow traffic
Applies to an instance only if someone specifies the security group when launching the instance, or associates the security group with the instance later on	Automatically applies to all instances in the subnets it's associated with (therefore, you don't have to rely on users to specify the security group)

VPC Flow Logs

- Capture information about IP traffic going into your interfaces:
    - VPC Flow Logs
    - Subnet Flow Logs
    - Elastic Network Interface Flow Logs
- Helps to monitor & troubleshoot connectivity issues. Example:
    - Subnets to internet
    - Subnets to subnets
    - Internet to subnets
- Captures network information from AWS managed interfaces too: Elastic Load Balancers, ElastiCache, RDS, Aurora, etc…
- VPC Flow logs data can go to S3 / CloudWatch Logs

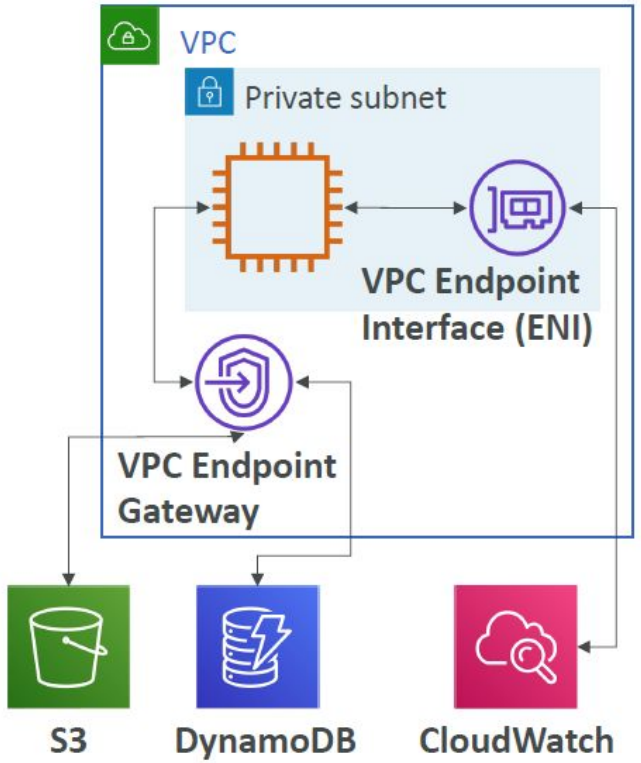
VPC Peering

- Connect two VPC, privately using AWS' network
- Make them behave as if they were in the same network
- Must not have overlapping CIDR (IP address range)
- VPC Peering connection is no transitive (must be established for each VPC that need to Communicate with one another)



VPC Endpoints

• Endpoints allow you to connect to AWS Services using a private network instead of the public www network
• This gives you enhanced security and lower latency to access AWS services
• VPC Endpoint Gateway: S3 & DynamoDB
• VPC Endpoint Interface: the rest
• Only used within your VPC



Site to Site VPN & Direct Connect

- Site to Site VPN
  - Connect an on-premises VPN to AWS
  - The connection is automatically encrypted
  - Goes over the public internet
- Direct Connect (DX)
  - Establish a physical connection between on premises and AWS
  - The connection is private, secure and fast
  - Goes over a private network
  - Takes at least a month to establish
- **Note: Site-to-site VPN and Direct Connect cannot access VPC endpoints**