



Department of Computer Science & Engineering

Cloud Computing

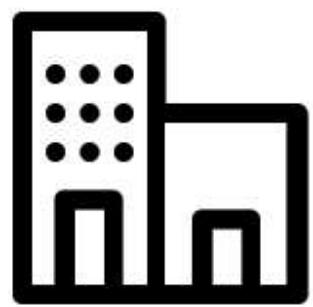
Lab

Manual

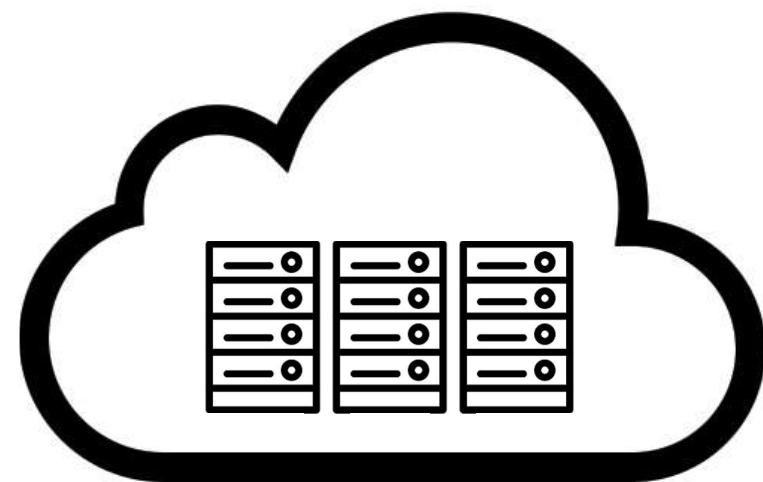
What is Cloud Computing?



- Cloud computing is the **on-demand delivery** of compute power, database storage, applications, and other IT resources
- Through a cloud services platform with **pay-as-you-go pricing**
- You can **provision exactly the right type and size of computing resources** you need
- You can access as many resources as you need, **almost instantly**
- Simple way to access **servers, storage, databases** and a set of **application services**
- Amazon Web Services owns and maintains the network-connected hardware required for these application services, while you provision and use what you need via a web application.



Office



The Cloud

You've been using some Cloud services



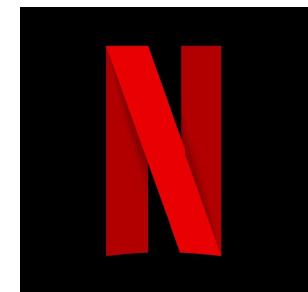
Gmail

- E-mail cloud service
- Pay for ONLY your emails stored (no infrastructure, etc.)



Dropbox

- Cloud Storage Service
- Originally built on AWS



Netflix

- Built on AWS
- Video on Demand

The Deployment Models of the Cloud

Private Cloud:

- Cloud services used by a single organization, not exposed to the public.
- Complete control
- Security for sensitive applications
- Meet specific business needs

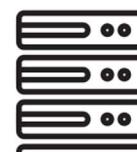
Public Cloud:

- Cloud resources owned and operated by a third-party cloud service provider delivered over the Internet.
- Six Advantages of Cloud Computing



Hybrid Cloud:

- Keep some servers on premises and extend some capabilities to the Cloud
- Control over sensitive assets in your private infrastructure
- Flexibility and cost-effectiveness of the public cloud



The Five Characteristics of Cloud Computing

- **On-demand self service:**
 - Users can provision resources and use them without human interaction from the service provider
- **Broad network access:**
 - Resources available over the network, and can be accessed by diverse client platforms
- **Multi-tenancy and resource pooling:**
 - Multiple customers can share the same infrastructure and applications with security and privacy
 - Multiple customers are serviced from the same physical resources
- **Rapid elasticity and scalability:**
 - Automatically and quickly acquire and dispose resources when needed
 - Quickly and easily scale based on demand
- **Measured service:**
 - Usage is measured, users pay correctly for what they have used

Six Advantages of Cloud Computing

- Trade capital expense (**CAPEX**) for operational expense (**OPEX**)
 - Pay On-Demand: don't own hardware
 - Reduced Total Cost of Ownership (TCO) & Operational Expense (OPEX)
- Benefit from massive economies of scale
 - Prices are reduced as AWS is more efficient due to large scale
- Stop guessing capacity
 - Scale based on actual measured usage
- Increase speed and agility
- Stop spending money running and maintaining data centers
- Go global in minutes: leverage the AWS global infrastructure

Problems solved by the Cloud

- **Flexibility:** change resource types when needed
- **Cost-Effectiveness:** pay as you go, for what you use
- **Scalability:** accommodate larger loads by making hardware stronger or adding additional nodes
- **Elasticity:** ability to scale out and scale-in when needed
- **High-availability and fault-tolerance:** build across data centers
- **Agility:** rapidly develop, test and launch software applications

Types of Cloud Computing

- **Infrastructure as a Service (IaaS)**

- Provide building blocks for cloud IT
- Provides networking, computers, data storage space
- Highest level of flexibility
- Easy parallel with traditional on-premises IT

- **Platform as a Service (PaaS)**

- Removes the need for your organization to manage the underlying infrastructure
- Focus on the deployment and management of your applications

- **Software as a Service (SaaS)**

- Completed product that is run and managed by the service provider

On-premises

Applications

Data

Runtime

Middleware

O/S

Virtualization

Servers

Storage

Networking

Infrastructure as a Service
(IaaS)

Applications

Data

Runtime

Middleware

O/S

Virtualization

Servers

Storage

Networking

Platform as a Service
(PaaS)

Applications

Data

Runtime

Middleware

O/S

Virtualization

Servers

Storage

Networking

Software as a Service
(SaaS)

Applications

Data

Runtime

Middleware

O/S

Virtualization

Servers

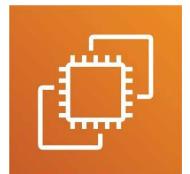
Storage

Networking

Example of Cloud Computing Types

- **Infrastructure as a Service:**

- Amazon EC2 (on AWS)
- GCP, Azure, Rackspace, Digital Ocean, Linode



- **Platform as a Service:**

- Elastic Beanstalk (on AWS)
- Heroku, Google App Engine (GCP), Windows Azure (Microsoft)



- **Software as a Service:**

- Many AWS services (ex: Rekognition for Machine Learning)
- Google Apps (Gmail), Dropbox, Zoom

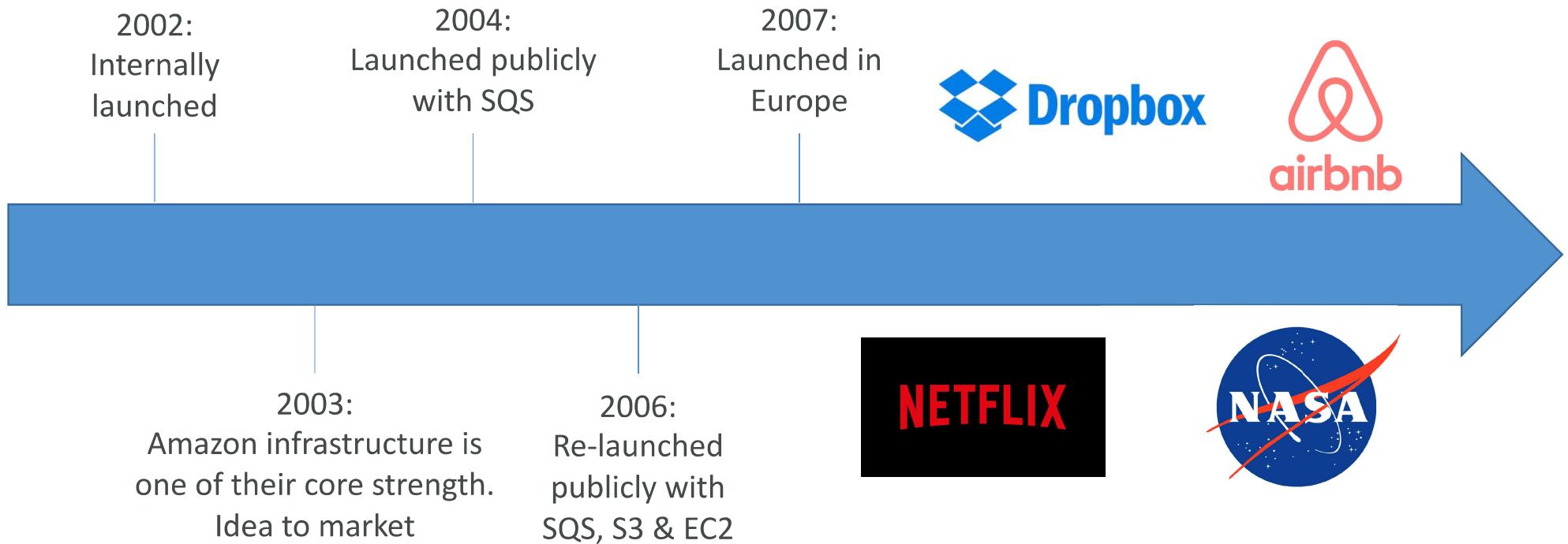


Pricing of the Cloud – Quick Overview

- AWS has 3 pricing fundamentals, following the pay-as-you-go pricing model
- **Compute:**
 - Pay for compute time
- **Storage:**
 - Pay for data stored in the Cloud
- **Data transfer OUT of the Cloud:**
 - Data transfer IN is free
- Solves the expensive issue of traditional IT



AWS Cloud History



AWS Cloud Number Facts

- In 2019, AWS had \$35.02 billion in annual revenue
- AWS accounts for 47% of the market in 2019 (Microsoft is 2nd with 22%)
- Pioneer and Leader of the AWS Cloud Market for the 9th consecutive year
- Over 1,000,000 active users

Figure 1. Magic Quadrant for Cloud Infrastructure as a Service, Worldwide



AWS Cloud Use Cases

- AWS enables you to build sophisticated, scalable applications
- Applicable to a diverse set of industries
- Use cases include
 - Enterprise IT, Backup & Storage, Big Data analytics
 - Website hosting, Mobile & Social Apps
 - Gaming



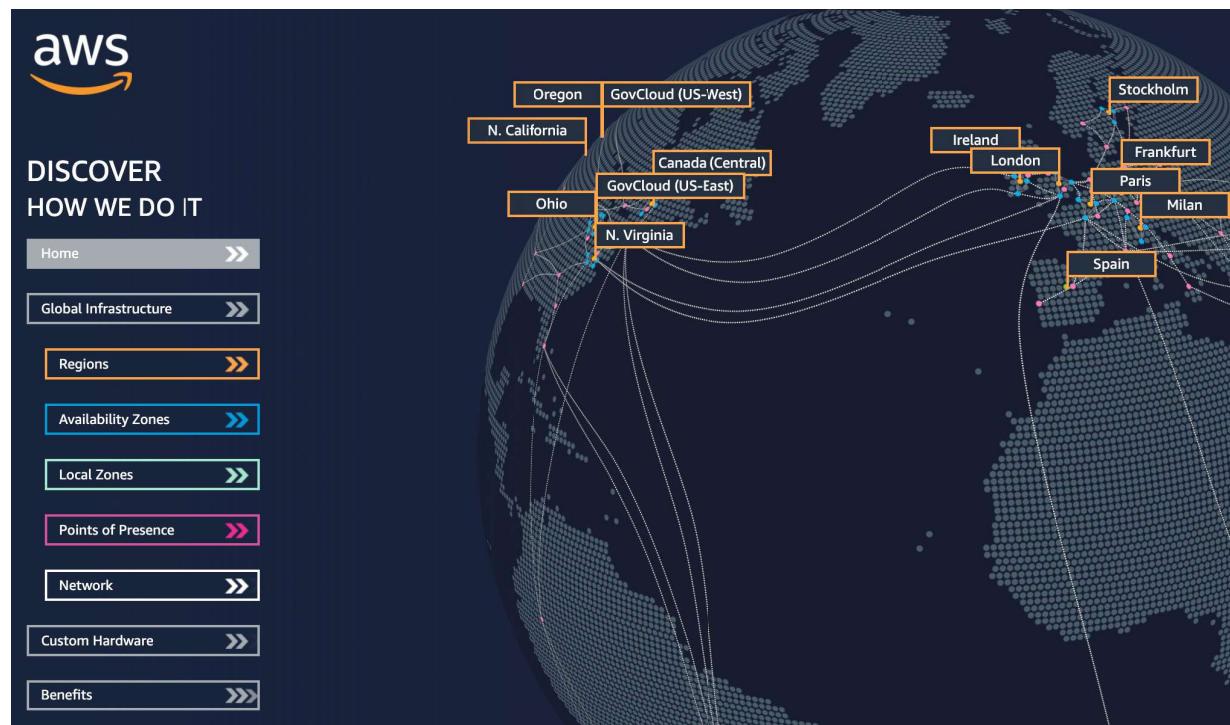
**21ST
CENTURY
FOX**

ACTIVISION

NETFLIX

AWS Global Infrastructure

- AWS Regions
- AWS Availability Zones
- AWS Data Centers
- AWS Edge Locations / Points of Presence
- <https://infrastructure.aws/>



AWS Regions

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



US East (N. Virginia)	<code>us-east-1</code>
US East (Ohio)	<code>us-east-2</code>
US West (N. California)	<code>us-west-1</code>
US West (Oregon)	<code>us-west-2</code>
<hr/>	
Africa (Cape Town)	<code>af-south-1</code>
<hr/>	
Asia Pacific (Hong Kong)	<code>ap-east-1</code>
Asia Pacific (Mumbai)	<code>ap-south-1</code>
Asia Pacific (Seoul)	<code>ap-northeast-2</code>
Asia Pacific (Singapore)	<code>ap-southeast-1</code>
Asia Pacific (Sydney)	<code>ap-southeast-2</code>
Asia Pacific (Tokyo)	<code>ap-northeast-1</code>
<hr/>	
Canada (Central)	<code>ca-central-1</code>
<hr/>	
Europe (Frankfurt)	<code>eu-central-1</code>
Europe (Ireland)	<code>eu-west-1</code>
Europe (London)	<code>eu-west-2</code>
Europe (Paris)	<code>eu-west-3</code>
Europe (Stockholm)	<code>eu-north-1</code>
<hr/>	
Middle East (Bahrain)	<code>me-south-1</code>

How to choose an AWS Region?

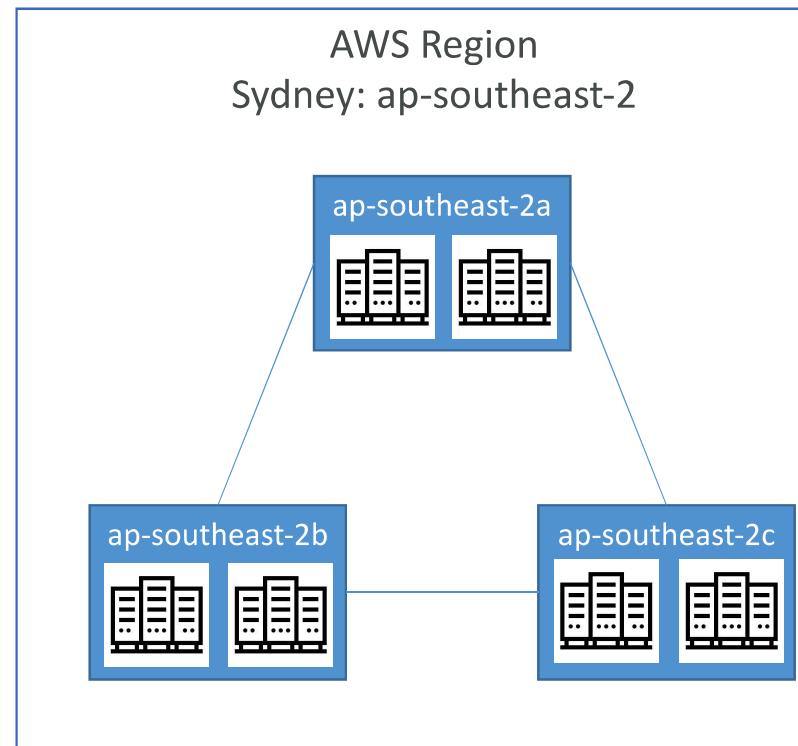
If you need to launch a new application,
where should you do it?



- **Compliance** with data governance and legal requirements: data never leaves a region without your explicit permission
- **Proximity** to customers: reduced latency
- **Available services** within a Region: new services and new features aren't available in every Region
- **Pricing**: pricing varies region to region and is transparent in the service pricing page

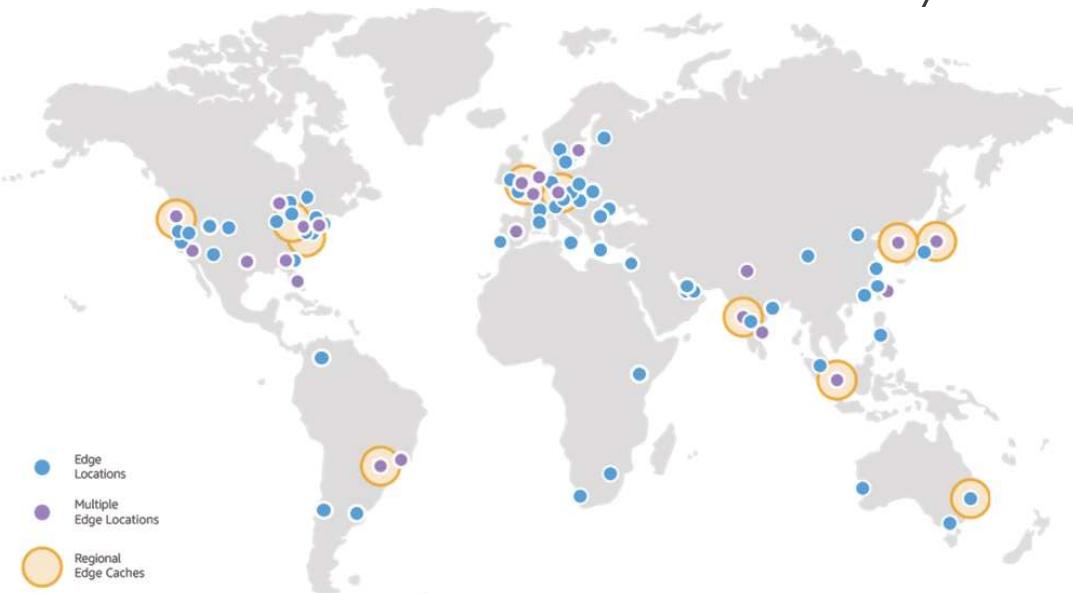
AWS Availability Zones

- Each region has many availability zones (usually 3, min is 3, 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
- They're connected with high bandwidth, ultra-low latency networking



AWS Points of Presence (Edge Locations)

- Amazon has 216 Points of Presence (205 Edge Locations & 11 Regional Caches) in 84 cities across 42 countries
- Content is delivered to end users with lower latency



IAM Section

IAM: Users & Groups



- IAM = Identity and Access Management, **Global** service
- Root account created by default, shouldn't be used or shared
- **Users** are people within your organization, and can be grouped
- **Groups** only contain users, not other groups
- Users don't have to belong to a group, and user can belong to multiple groups



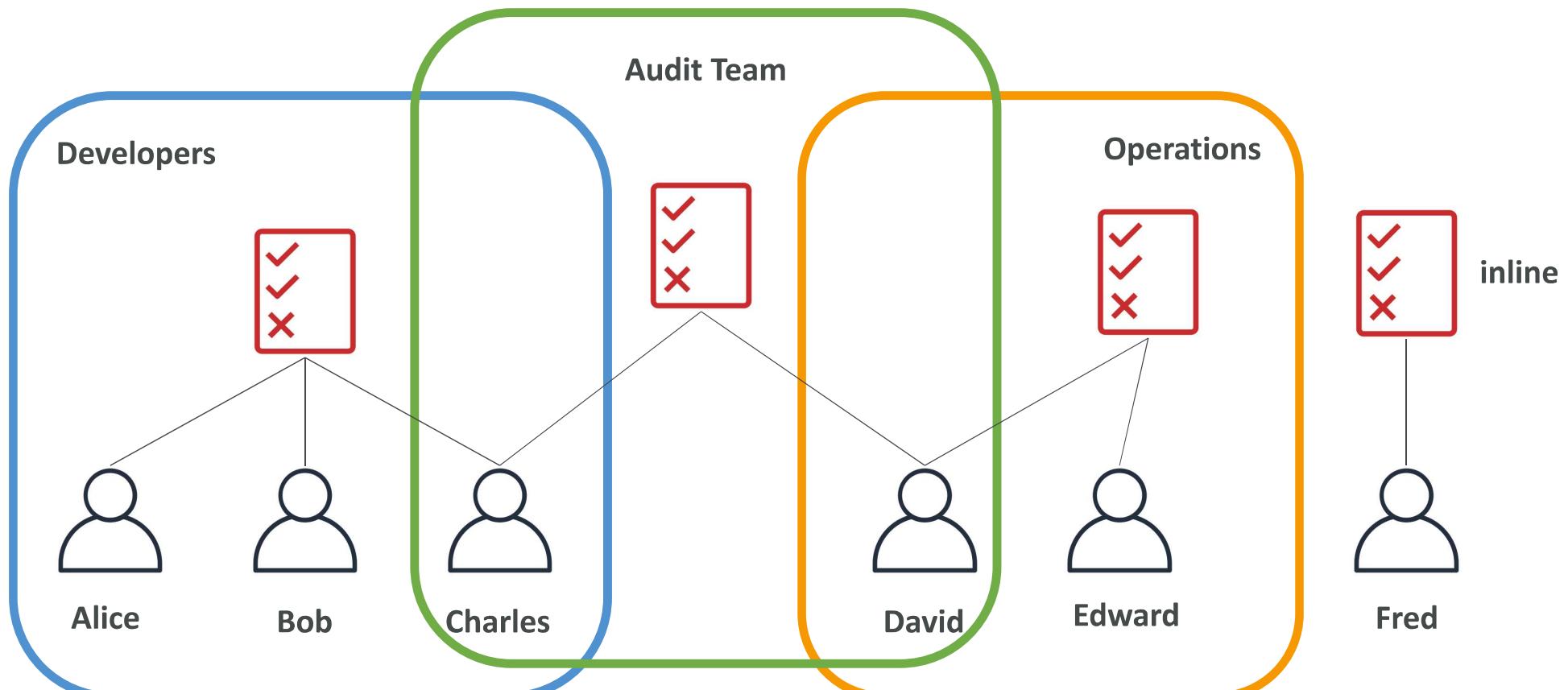
IAM: Permissions

- Users or Groups can be assigned JSON documents called policies
- These policies define the permissions of the users
- In AWS you apply the **least privilege principle**: don't give more permissions than a user needs

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Effect": "Allow",  
      "Action": "ec2:Describe*",  
      "Resource": "*"  
    },  
    {  
      "Effect": "Allow",  
      "Action": "elasticloadbalancing:Describe*",  
      "Resource": "*"  
    },  
    {  
      "Effect": "Allow",  
      "Action": [  
        "cloudwatch:ListMetrics",  
        "cloudwatch:GetMetricStatistics",  
        "cloudwatch:Describe"  
      ],  
      "Resource": "*"  
    }  
  ]  
}
```



IAM Policies inheritance



IAM Policies Structure

- Consists of
 - **Version:** policy language version, always include “2012-10-17”
 - **Id:** an identifier for the policy (optional)
 - **Statement:** one or more individual statements (required)
- Statements consists of
 - **Sid:** an identifier for the statement (optional)
 - **Effect:** whether the statement allows or denies access (Allow, Deny)
 - **Principal:** account/user/role to which this policy applied to
 - **Action:** list of actions this policy allows or denies
 - **Resource:** list of resources to which the actions applied to
 - **Condition:** conditions for when this policy is in effect (optional)

```
{  
    "Version": "2012-10-17",  
    "Id": "S3-Account-Permissions",  
    "Statement": [  
        {  
            "Sid": "1",  
            "Effect": "Allow",  
            "Principal": {  
                "AWS": ["arn:aws:iam::123456789012:root"]  
            },  
            "Action": [  
                "s3:GetObject",  
                "s3:PutObject"  
            ],  
            "Resource": ["arn:aws:s3:::mybucket/*"]  
        }  
    ]  
}
```

IAM – Password Policy

- Strong passwords = higher security for your account
- In AWS, you can setup a password policy:
 - Set a minimum password length
 - Require specific character types:
 - including uppercase letters
 - lowercase letters
 - numbers
 - non-alphanumeric characters
 - Allow all IAM users to change their own passwords
 - Require users to change their password after some time (password expiration)
 - Prevent password re-use

Multi Factor Authentication - MFA



- Users have access to your account and can possibly change configurations or delete resources in your AWS account
- You want to protect your Root Accounts and IAM users
- MFA = password you know + security device you own



Alice

Password

+



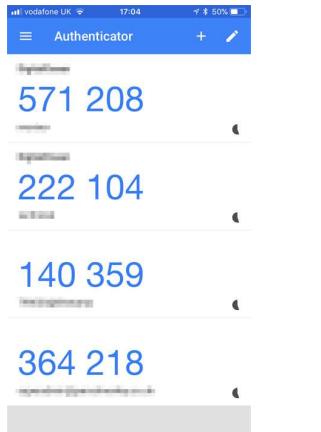
=>

Successful login

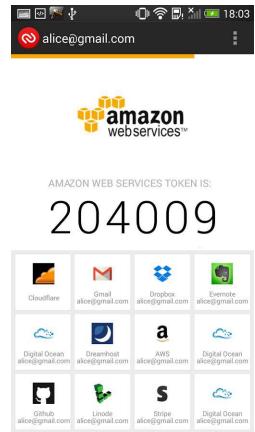
- Main benefit of MFA:
if a password is stolen or hacked, the account is not compromised

MFA devices options in AWS

Virtual MFA device



Google Authenticator
(phone only)



Authy
(multi-device)

Support for multiple tokens on a single device.

Universal 2nd Factor (U2F) Security Key

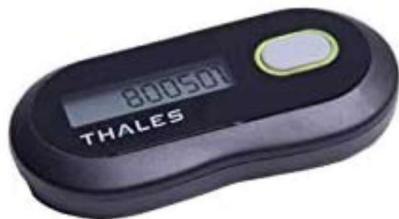


YubiKey by Yubico (3rd party)

Support for multiple root and IAM users using a single security key

MFA devices options in AWS

Hardware Key Fob MFA Device



Provided by Gemalto (3rd party)

Hardware Key Fob MFA Device for AWS GovCloud (US)



Provided by SurePassID (3rd party)

How can users access AWS ?



- To access AWS, you have three options:
 - AWS Management Console (protected by password + MFA)
 - AWS Command Line Interface (CLI): protected by access keys
 - AWS Software Developer Kit (SDK) - for code: protected by access keys
- Access Keys are generated through the AWS Console
- Users manage their own access keys
- Access Keys are secret, just like a password. Don't share them
- Access Key ID ~ = username
- Secret Access Key ~ = password

Example (Fake) Access Keys

Access keys

Use access keys to make secure REST or HTTP Query protocol requests to AWS service APIs. For your protection, you should never share your secret keys with anyone. As a best practice, we recommend frequent key rotation. [Learn more](#)

[Create access key](#)

Access key ID	Created	Last used	Status	
AKIASK4E37PV4TU3RD6C	2020-05-25 15:13 UTC+0100	N/A	Active	Make inactive X

- Access key ID: AKIASK4E37PV4983d6C
- Secret Access Key: AZPN3zojWozWCndljhB0Unh8239albzBzO5fqQkZq
- Remember: don't share your access keys

What's the AWS CLI?

- A tool that enables you to interact with AWS services using commands in your command-line shell
- Direct access to the public APIs of AWS services
- You can develop scripts to manage your resources
- It's open-source <https://github.com/aws/aws-cli>
- Alternative to using AWS Management Console

```
→ ~ aws s3 cp myfile.txt s3://ccp-mybucket/myfile.txt
upload: ./myfile.txt to s3://ccp-mybucket/myfile.txt
→ ~ aws s3 ls s3://ccp-mybucket
2021-05-14 03:22:52          0 myfile.txt
→ ~ |
```

What's the AWS SDK?



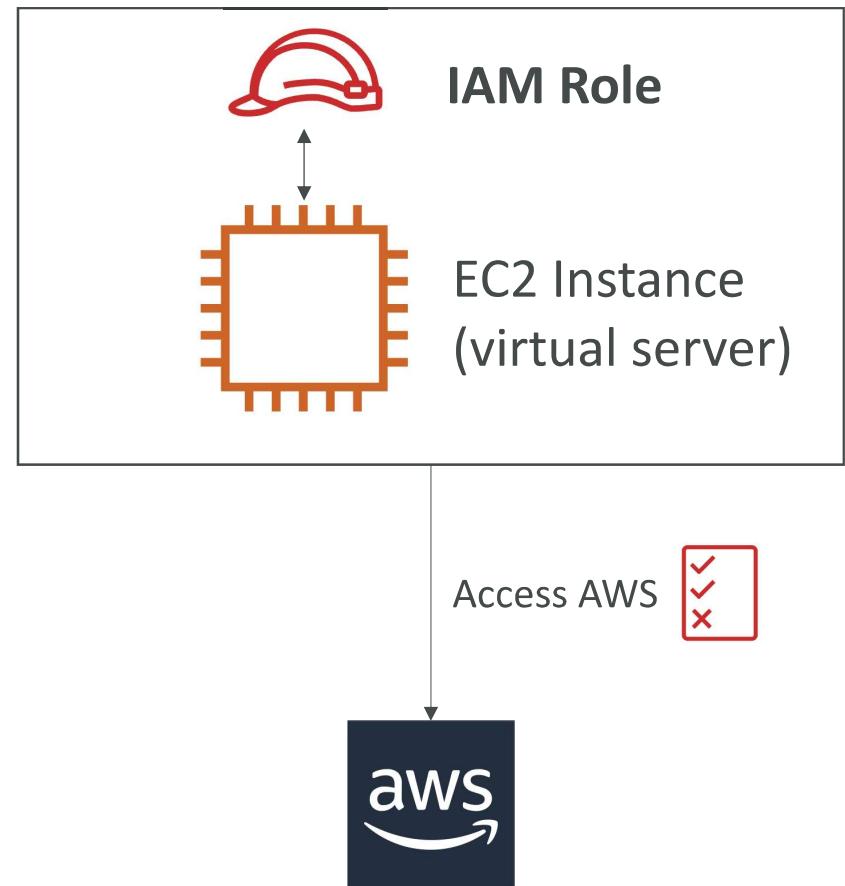
- AWS Software Development Kit (AWS SDK)
- Language-specific APIs (set of libraries)
- Enables you to access and manage AWS services programmatically
- Embedded within your application
- Supports
 - SDKs (JavaScript, Python, PHP, .NET, Ruby, Java, Go, Node.js, C++)
 - Mobile SDKs (Android, iOS, ...)
 - IoT Device SDKs (Embedded C, Arduino, ...)
- Example: AWS CLI is built on AWS SDK for Python



Your Application

IAM Roles for Services

- Some AWS service will need to perform actions on your behalf
- To do so, we will assign permissions to AWS services with IAM Roles
- Common roles:
 - EC2 Instance Roles
 - Lambda Function Roles
 - Roles for CloudFormation



IAM Security Tools

- IAM Credentials Report (account-level)
 - a report that lists all your account's users and the status of their various credentials
- IAM Access Advisor (user-level)
 - Access advisor shows the service permissions granted to a user and when those services were last accessed.
 - You can use this information to revise your policies.

IAM Guidelines & Best Practices



- Don't use the root account except for AWS account setup
- One physical user = One AWS user
- Assign users to groups and assign permissions to groups
- Create a strong password policy
- Use and enforce the use of Multi Factor Authentication (MFA)
- Create and use Roles for giving permissions to AWS services
- Use Access Keys for Programmatic Access (CLI / SDK)
- Audit permissions of your account with the IAM Credentials Report
- Never share IAM users & Access Keys

Shared Responsibility Model for IAM



- Infrastructure (global network security)
- Configuration and vulnerability analysis
- Compliance validation



You

- Users, Groups, Roles, Policies management and monitoring
- Enable MFA on all accounts
- Rotate all your keys often
- Use IAM tools to apply appropriate permissions
- Analyze access patterns & review permissions

IAM Section – Summary



- **Users:** mapped to a physical user, has a password for AWS Console
- **Groups:** contains users only
- **Policies:** JSON document that outlines permissions for users or groups
- **Roles:** for EC2 instances or AWS services
- **Security:** MFA + Password Policy
- **AWS CLI:** manage your AWS services using the command-line
- **AWS SDK:** manage your AWS services using a programming language
- **Access Keys:** access AWS using the CLI or SDK
- **Audit:** IAM Credential Reports & IAM Access Advisor

EC2 Section

Amazon EC2



- EC2 is one of the most popular of AWS' offering
- EC2 = Elastic Compute Cloud = Infrastructure as a Service
- 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)
- Knowing EC2 is fundamental to understand how the Cloud works

EC2 Section

Amazon EC2



- EC2 is one of the most popular of AWS' offering
- EC2 = Elastic Compute Cloud = Infrastructure as a Service
- 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)
- Knowing EC2 is fundamental to understand how the Cloud works

EC2 sizing & configuration options

- Operating System (**OS**): Linux, Windows or Mac OS
- How much compute power & cores (**CPU**)
- How much random-access memory (**RAM**)
- How much storage space:
 - Network-attached (**EBS & EFS**)
 - hardware (**EC2 Instance Store**)
- Network card: speed of the card, Public IP address
- Firewall rules: **security group**
- Bootstrap script (configure at first launch): **EC2 User Data**

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
 - Anything you can think of
- The EC2 User Data Script runs with the root user

Hands-On: Launching an EC2 Instance running Linux

- We'll be launching our first virtual server using the AWS Console
- We'll get a first high-level approach to the various parameters
- We'll see that our web server is launched using EC2 user data
- We'll learn how to start / stop / terminate our instance.

EC2 Instance Types - Overview

- You can use different types of EC2 instances that are optimised for different use cases (<https://aws.amazon.com/ec2/instance-types/>)
- AWS has the following naming convention:

m5.2xlarge

- m: instance class
- 5: generation (AWS improves them over time)
- 2xlarge: size within the instance class

General Purpose
Compute Optimized
Memory Optimized
Accelerated Computing
Storage Optimized
Instance Features
Measuring Instance Performance

EC2 Instance Types – General Purpose

- Great for a diversity of workloads such as web servers or code repositories
- Balance between:
 - Compute
 - Memory
 - Networking
- In the course, we will be using the t2.micro which is a General Purpose EC2 instance

General Purpose

General purpose instances provide a balance of compute, memory and networking resources, and can be used for a variety of diverse workloads. These instances are ideal for applications that use these resources in equal proportions such as web servers and code repositories.

Mac	T4g	T3	T3a	T2	M6g	M5	M5a	M5zn	M4	A1
-----	-----	----	-----	----	-----	----	-----	------	----	----

EC2 Instance Types – Compute Optimized

- Great for compute-intensive tasks that require high performance processors:
 - Batch processing workloads
 - Media transcoding
 - High performance web servers
 - High performance computing (HPC)
 - Scientific modeling & machine learning
 - Dedicated gaming servers

Compute Optimized

Compute Optimized instances are ideal for compute bound applications that benefit from high performance processors. Instances belonging to this family are well suited for batch processing workloads, media transcoding, high performance web servers, high performance computing (HPC), scientific modeling, dedicated gaming servers and ad server engines, machine learning inference and other compute intensive applications.

C6g

C6gn

C5

C5a

C5n

C4

EC2 Instance Types – Memory Optimized

- Fast performance for workloads that process large data sets in memory
- Use cases:
 - High performance, relational/non-relational databases
 - Distributed web scale cache stores
 - In-memory databases optimized for BI (business intelligence)
 - Applications performing real-time processing of big unstructured data

Memory Optimized

Memory optimized instances are designed to deliver fast performance for workloads that process large data sets in memory.

R6g	R5	R5a	R5b	R5n	R4	X1e	X1	High Memory	z1d
-----	----	-----	-----	-----	----	-----	----	-------------	-----

EC2 Instance Types – Storage Optimized

- Great for storage-intensive tasks that require high, sequential read and write access to large data sets on local storage
- Use cases:
 - High frequency online transaction processing (OLTP) systems
 - Relational & NoSQL databases
 - Cache for in-memory databases (for example, Redis)
 - Data warehousing applications
 - Distributed file systems

Storage Optimized

Storage optimized instances are designed for workloads that require high, sequential read and write access to very large data sets on local storage. They are optimized to deliver tens of thousands of low-latency, random I/O operations per second (IOPS) to applications.

I3	I3en	D2	D3	D3en	H1
----	------	----	----	------	----

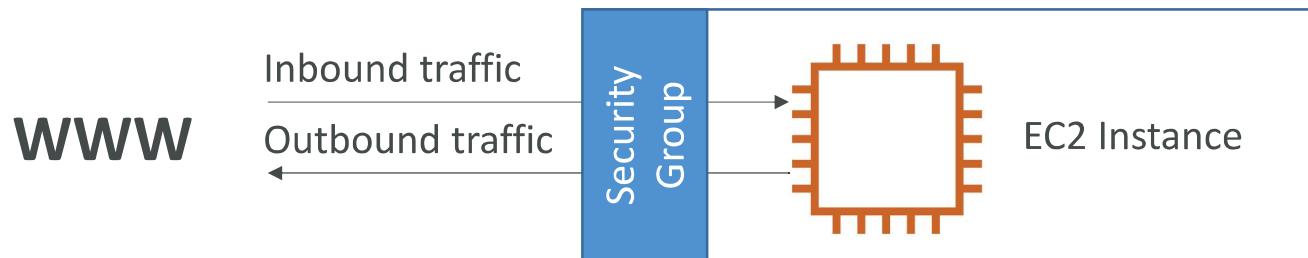
EC2 Instance Types: example

Instance	vCPU	Mem (GiB)	Storage	Network Performance	EBS Bandwidth (Mbps)
t2.micro	1	1	EBS-Only	Low to Moderate	
t2.xlarge	4	16	EBS-Only	Moderate	
c5d.4xlarge	16	32	1 x 400 NVMe SSD	Up to 10 Gbps	4,750
r5.16xlarge	64	512	EBS Only	20 Gbps	13,600
m5.8xlarge	32	128	EBS Only	10 Gbps	6,800

t2.micro is part of the AWS free tier (up to 750 hours per month)

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 Instances.



- Security groups only contain **allow** rules
- Security groups rules can reference by IP or by security group

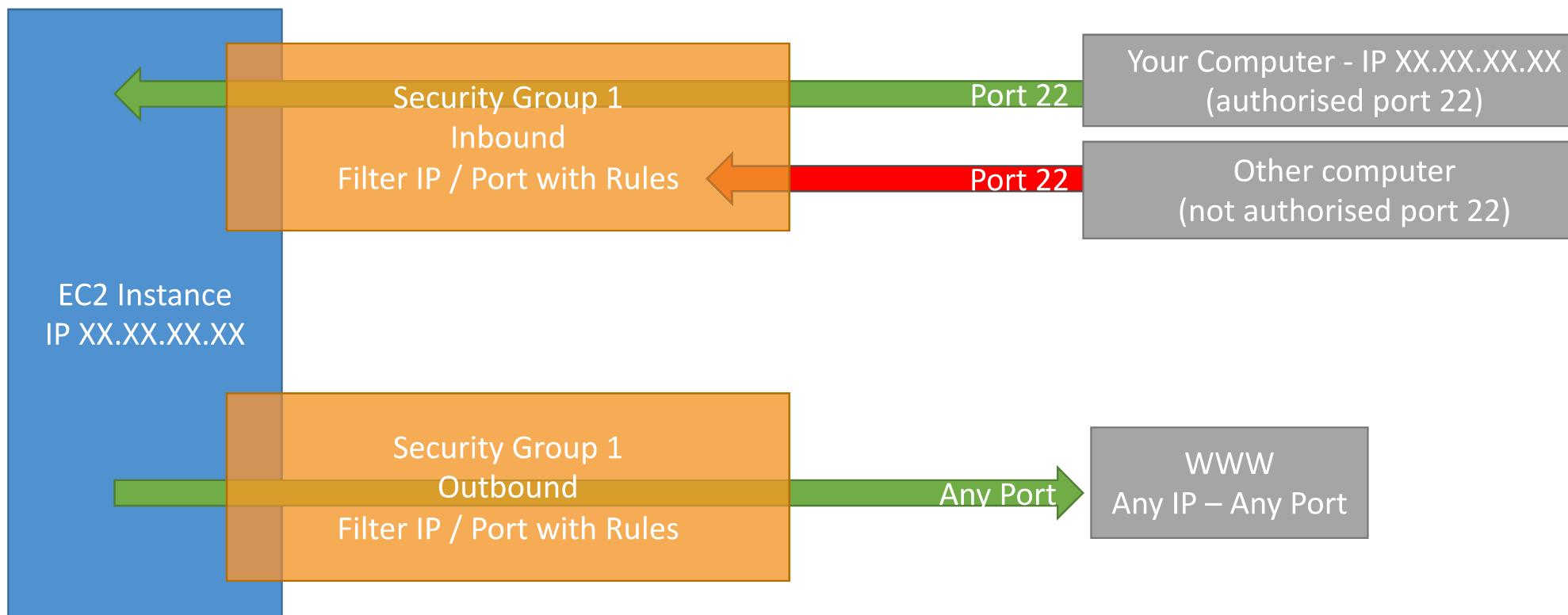
Security Groups

Deeper Dive

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

Type	Protocol	Port Range	Source	Description
HTTP	TCP	80	0.0.0.0/0	test http page
SSH	TCP	22	122.149.196.85/32	
Custom TCP Rule	TCP	4567	0.0.0.0/0	java app

Security Groups Diagram



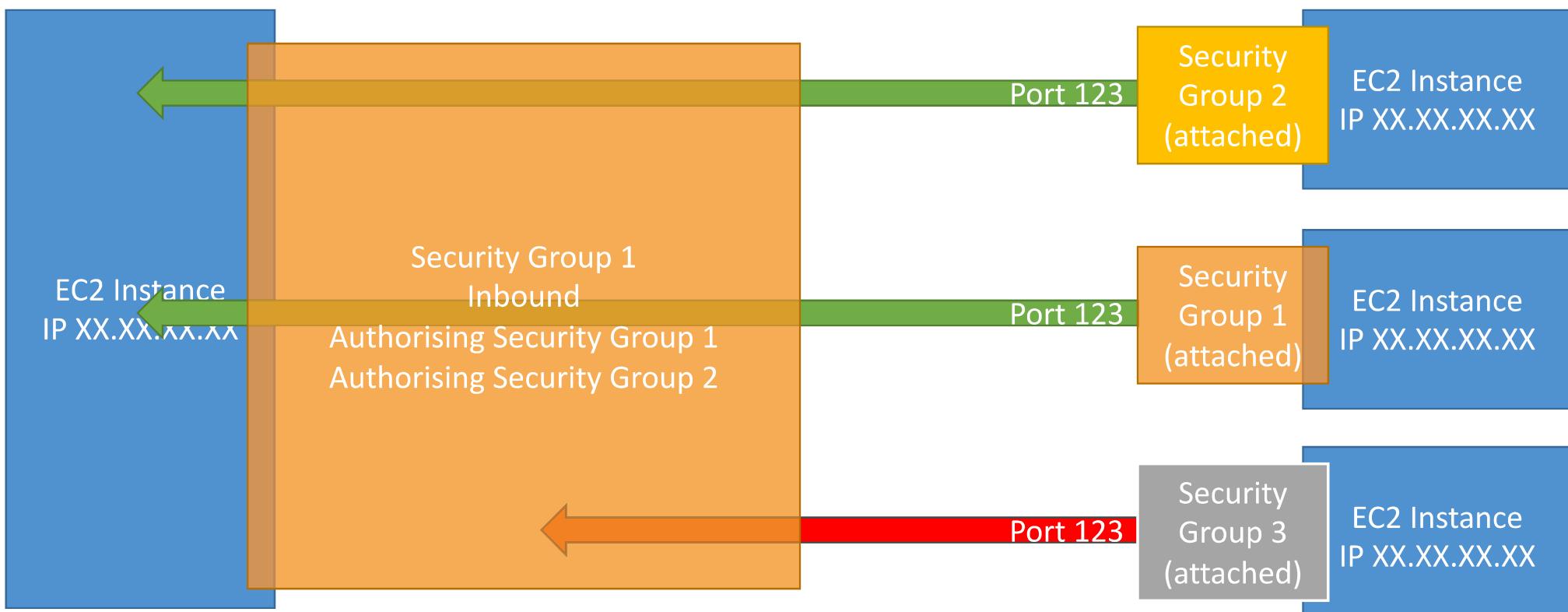
Security Groups

Good to know

- Can be attached to multiple instances
- Locked down to a region / VPC combination
- Does live “outside” the EC2 – if traffic is blocked the EC2 instance won’t see it
- 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 **authorised** by default

Referencing other security groups

Diagram



Classic Ports to know

- 22 = SSH (Secure Shell) - log into a Linux instance
- 21 = FTP (File Transfer Protocol) – upload files into a file share
- 22 = SFTP (Secure File Transfer Protocol) – upload files using SSH
- 80 = HTTP – access unsecured websites
- 443 = HTTPS – access secured websites
- 3389 = RDP (Remote Desktop Protocol) – log into a Windows instance

SSH Summary Table

	SSH	Putty	EC2 Instance Connect
Mac	✓		✓
Linux	✓		✓
Windows < 10		✓	✓
Windows ≥ 10	✓	✓	✓

Which Lectures to watch

- Mac / Linux:
 - SSH on Mac/Linux lecture
- Windows:
 - Putty Lecture
 - If Windows 10: SSH on Windows 10 lecture
- All:
 - EC2 Instance Connect lecture

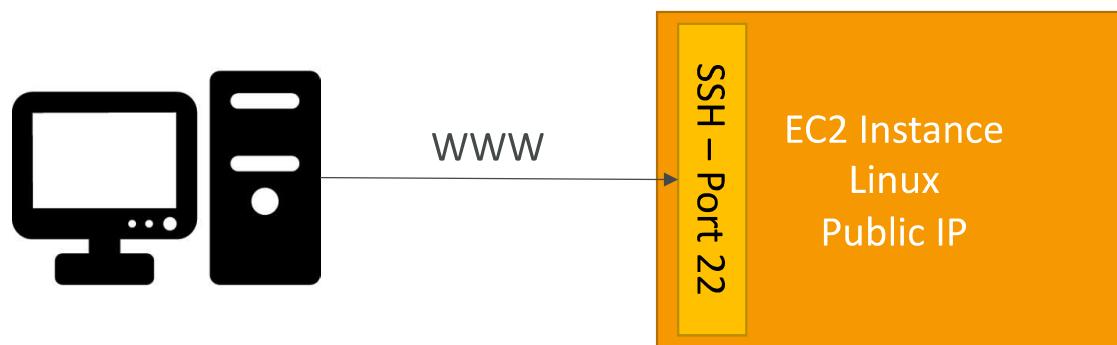
SSH troubleshooting

- Students have the most problems with SSH
- If things don't work...
 1. Re-watch the lecture. You may have missed something
 2. Read the troubleshooting guide
 3. Try EC2 Instance Connect
- If one method works (SSH, Putty or EC2 Instance Connect) you're good
- If no method works, that's okay, the course won't use SSH much

How to SSH into your EC2 Instance

Linux / Mac OS X

- We'll learn how to SSH into your EC2 instance using [Linux / Mac](#)
- SSH is one of the most important function. It allows you to control a remote machine, all using the command line.

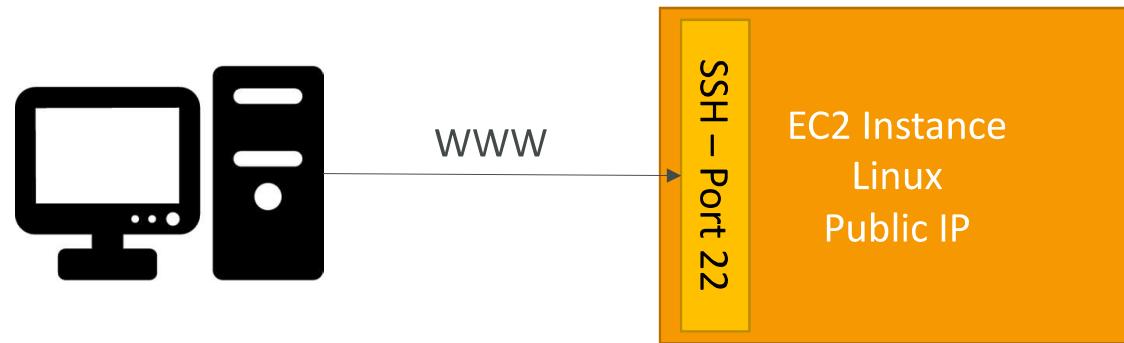


- We will see how we can configure OpenSSH [~/.ssh/config](#) to facilitate the SSH into our EC2 instances

How to SSH into your EC2 Instance

Windows

- We'll learn how to SSH into your EC2 instance using [Windows](#)
- SSH is one of the most important function. It allows you to control a remote machine, all using the command line.



- We will configure all the required parameters necessary for doing SSH on Windows using the free tool [Putty](#).

EC2 Instance Connect

- Connect to your EC2 instance within your browser
- No need to use your key file that was downloaded
- The “magic” is that a temporary key is uploaded onto EC2 by AWS
- Works only out-of-the-box with Amazon Linux 2
- Need to make sure the port 22 is still opened!

EC2 Instances Purchasing Options

- On-Demand Instances – short workload, predictable pricing, pay by second
- Reserved (1 & 3 years)
 - Reserved Instances – long workloads
 - Convertible Reserved Instances – long workloads with flexible instances
- Savings Plans (1 & 3 years) – commitment to an amount of usage, long workload
- Spot Instances – short workloads, cheap, can lose instances (less reliable)
- Dedicated Hosts – book an entire physical server, control instance placement
- Dedicated Instances – no other customers will share your hardware
- Capacity Reservations – reserve capacity in a specific AZ for any duration

EC2 On Demand

- Pay for what you use:
 - Linux or Windows - billing per second, after the first minute
 - All other operating systems - billing per hour
- 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 **72%** discount compared to On-demand
 - You reserve a specific instance attributes (Instance Type, Region, Tenancy, OS)
 - Reservation Period – 1 year (+discount) or 3 years (+++discount)
 - Payment Options – No Upfront (+), Partial Upfront (++) , All Upfront (+++)
 - Reserved Instance's Scope – Regional or Zonal (reserve capacity in an AZ)
 - Recommended for steady-state usage applications (think database)
 - You can buy and sell in the Reserved Instance Marketplace
-
- **Convertible Reserved Instance**
 - Can change the EC2 instance type, instance family, OS, scope and tenancy
 - Up to **66%** discount

Note: the % discounts are different from the video as AWS
change them over time – the exact numbers are not needed

EC2 Savings Plans

- Get a discount based on long-term usage (up to 72% - same as RIs)
- Commit to a certain type of usage (\$10/hour for 1 or 3 years)
- Usage beyond EC2 Savings Plans is billed at the On-Demand price
- Locked to a specific instance family & AWS region (e.g., M5 in us-east-1)
- Flexible across:
 - Instance Size (e.g., m5.xlarge, m5.2xlarge)
 - OS (e.g., Linux, Windows)
 - Tenancy (Host, Dedicated, Default)



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
 - Any **distributed** workloads
 - Workloads with a flexible start and end time
- Not suitable for critical jobs or databases

EC2 Dedicated Hosts

- A physical server with EC2 instance capacity fully dedicated to your use
- Allows you address **compliance requirements** and **use your existing server-bound software licenses** (per-socket, per-core, pe—VM software licenses)
- Purchasing Options:
 - On-demand – pay per second for active Dedicated Host
 - Reserved - 1 or 3 years (No Upfront, Partial Upfront, All Upfront)
- The most expensive option
- 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 run 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

EC2 Capacity Reservations

- Reserve **On-Demand** instances capacity in a specific AZ for any duration
- You always have access to EC2 capacity when you need it
- **No time commitment** (create/cancel anytime), **no billing discounts**
- Combine with Regional Reserved Instances and Savings Plans to benefit from billing discounts
- You're charged at On-Demand rate whether you run instances or not

- Suitable for short-term, uninterrupted workloads that needs to be in a specific AZ

Which purchasing option is right for me?



- **On demand:** coming and staying in resort whenever we like, we pay the full price
- **Reserved:** like planning ahead and if we plan to stay for a long time, we may get a good discount.
- **Savings Plans:** pay a certain amount per hour for certain period and stay in any room type (e.g., King, Suite, Sea View, ...)
- **Spot instances:** the hotel allows people to bid for the empty rooms and the highest bidder keeps the rooms. You can get kicked out at any time
- **Dedicated Hosts:** We book an entire building of the resort
- **Capacity Reservations:** you book a room for a period with full price even you don't stay in it

Price Comparison

Example – m4.large – us-east-1

Price Type	Price (per hour)
On-Demand	\$0.10
Spot Instance (Spot Price)	\$0.038 - \$0.039 (up to 61% off)
Reserved Instance (1 year)	\$0.062 (No Upfront) - \$0.058 (All Upfront)
Reserved Instance (3 years)	\$0.043 (No Upfront) - \$0.037 (All Upfront)
EC2 Savings Plan (1 year)	\$0.062 (No Upfront) - \$0.058 (All Upfront)
Reserved Convertible Instance (1 year)	\$0.071 (No Upfront) - \$0.066 (All Upfront)
Dedicated Host	On-Demand Price
Dedicated Host Reservation	Up to 70% off
Capacity Reservations	On-Demand Price

Shared Responsibility Model for EC2



- Infrastructure (global network security)
 - Isolation on physical hosts
 - Replacing faulty hardware
 - Compliance validation
-
- Security Groups rules
 - Operating-system patches and updates
 - Software and utilities installed on the EC2 instance
 - IAM Roles assigned to EC2 & IAM user access management
 - Data security on your instance

EC2 Section – Summary



- **EC2 Instance:** AMI (OS) + Instance Size (CPU + RAM) + Storage + security groups + EC2 User Data
- **Security Groups:** Firewall attached to the EC2 instance
- **EC2 User Data:** Script launched at the first start of an instance
- **SSH:** start a terminal into our EC2 Instances (port 22)
- **EC2 Instance Role:** link to IAM roles
- **Purchasing Options:** On-Demand, Spot, Reserved (Standard + Convertible + Scheduled), Dedicated Host, Dedicated Instance

Amazon S3 Section

Section introduction



- Amazon S3 is one of the main building blocks of AWS
- It's advertised as "infinitely scaling" storage
- Many websites use Amazon S3 as a backbone
- Many AWS services use Amazon S3 as an integration as well
- We'll have a step-by-step approach to S3

Amazon S3 Use cases

- Backup and storage
- Disaster Recovery
- Archive
- Hybrid Cloud storage
- Application hosting
- Media hosting
- Data lakes & big data analytics
- Software delivery
- Static website



Nasdaq stores 7 years of data into S3 Glacier



Sysco runs analytics on its data and gain business insights

Amazon S3 - Buckets

- Amazon S3 allows people to store objects (files) in “buckets” (directories)
- Buckets must have a **globally unique name** (across all regions all accounts)
- Buckets are defined at the region level
- S3 looks like a global service but buckets are created in a region
- Naming convention
 - No uppercase, No underscore
 - 3-63 characters long
 - Not an IP
 - Must start with lowercase letter or number
 - Must NOT start with the prefix `xn--`
 - Must NOT end with the suffix `-s3alias`



S3 Bucket

Amazon S3 - Objects

- Objects (files) have a Key
- The **key** is the **FULL** path:
 - s3://my-bucket/**my_file.txt**
 - s3://my-bucket/**my_folder1/another_folder/my_file.txt**
- The key is composed of **prefix** + **object name**
 - s3://my-bucket/**my_folder1/another_folder**/**my_file.txt**
- There's no concept of "directories" within buckets (although the UI will trick you to think otherwise)
- Just keys with very long names that contain slashes ("/")

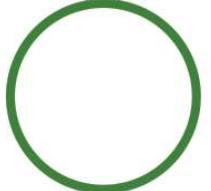


Object



S3 Bucket
with Objects

Amazon S3 – Objects (cont.)



- Object values are the content of the body:
 - Max. Object Size is 5TB (5000GB)
 - If uploading more than 5GB, must use “multi-part upload”
- Metadata (list of text key / value pairs – system or user metadata)
- Tags (Unicode key / value pair – up to 10) – useful for security / lifecycle
- Version ID (if versioning is enabled)

Amazon S3 – Security

- User-Based
 - IAM Policies – which API calls should be allowed for a specific user from IAM
- Resource-Based
 - Bucket Policies – bucket wide rules from the S3 console - allows cross account
 - Object Access Control List (ACL) – finer grain (can be disabled)
 - Bucket Access Control List (ACL) – less common (can be disabled)
- **Note:** an IAM principal can access an S3 object if
 - The user IAM permissions ALLOW it OR the resource policy ALLOWS it
 - AND there's no explicit DENY
- **Encryption:** encrypt objects in Amazon S3 using encryption keys

S3 Bucket Policies

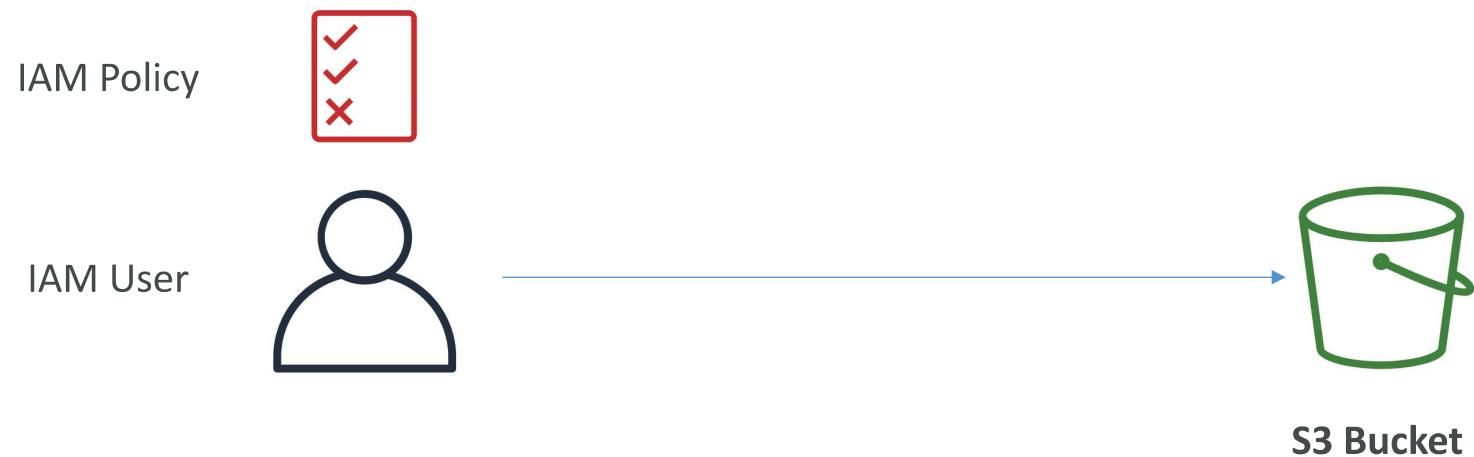
- JSON based policies
 - Resources: buckets and objects
 - Effect: Allow / Deny
 - Actions: Set of API to Allow or Deny
 - Principal: The account or user to apply the policy to
- Use S3 bucket for policy to:
 - Grant public access to the bucket
 - Force objects to be encrypted at upload
 - Grant access to another account (Cross Account)

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

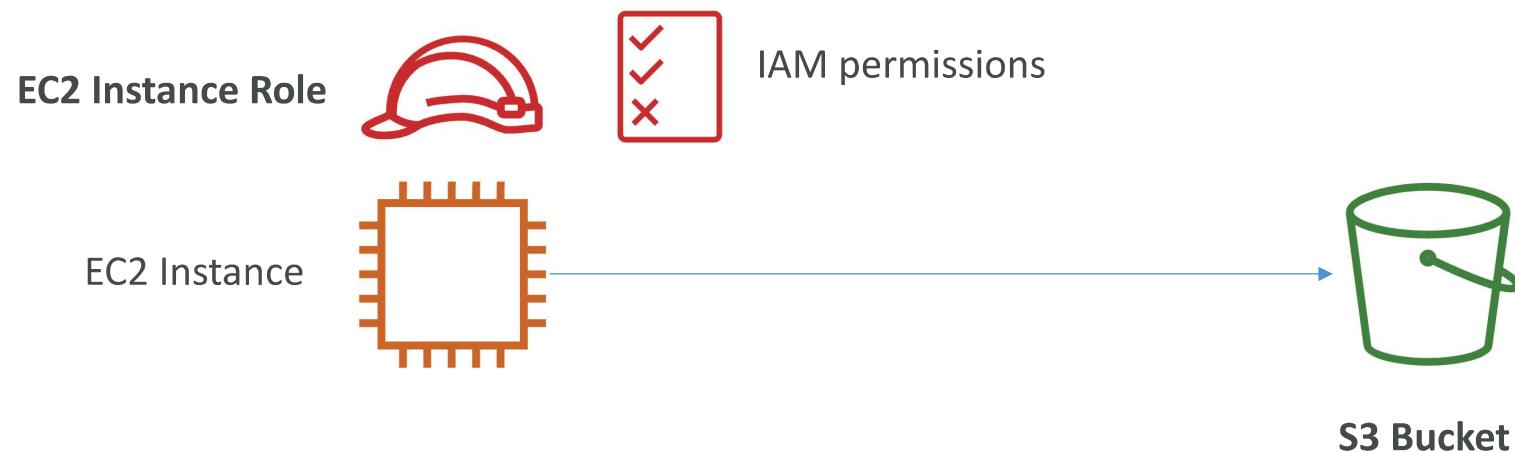
Example: Public Access - Use Bucket Policy



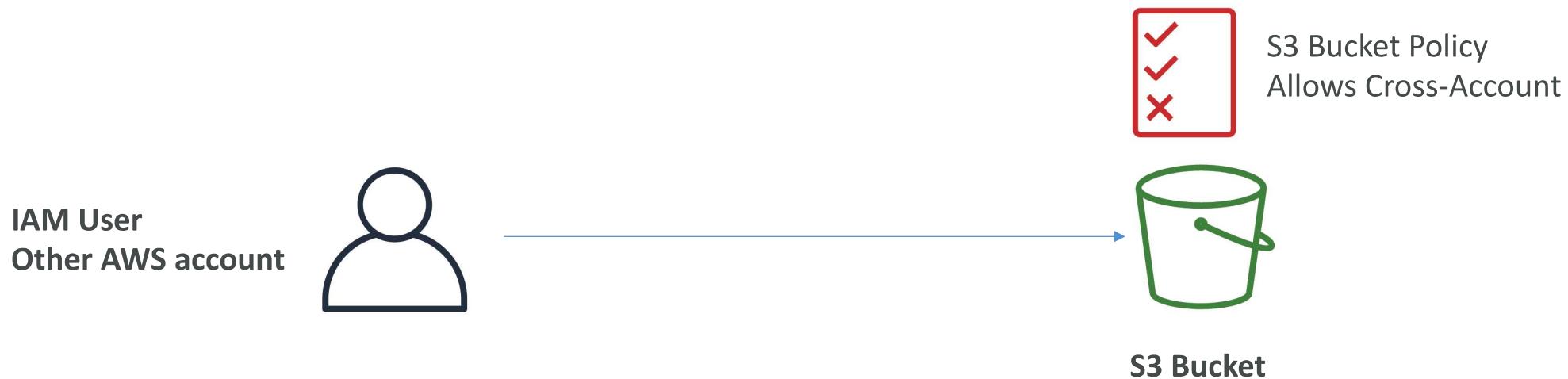
Example: User Access to S3 – IAM permissions



Example: EC2 instance access - Use IAM Roles



Advanced: Cross-Account Access – Use Bucket Policy



Bucket settings for Block Public Access

Block <i>all</i> public access	
On	Block public access to buckets and objects granted through <i>new</i> access control lists (ACLs)
On	Block public access to buckets and objects granted through <i>any</i> access control lists (ACLs)
On	Block public access to buckets and objects granted through <i>new</i> public bucket or access point policies
On	Block public and cross-account access to buckets and objects through <i>any</i> public bucket or access point policies

- These settings were created to prevent company data leaks
- If you know your bucket should never be public, leave these on
- Can be set at the account level

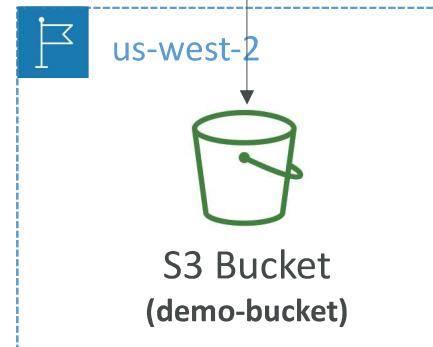
Amazon S3 – Static Website Hosting

- S3 can host static websites and have them accessible on the Internet
- The website URL will be (depending on the region)
 - `http://bucket-name.s3-website-aws-region.amazonaws.com`
 - OR
 - `http://bucket-name.s3-website.aws-region.amazonaws.com`
- If you get a **403 Forbidden** error; make sure the bucket policy allows public reads!



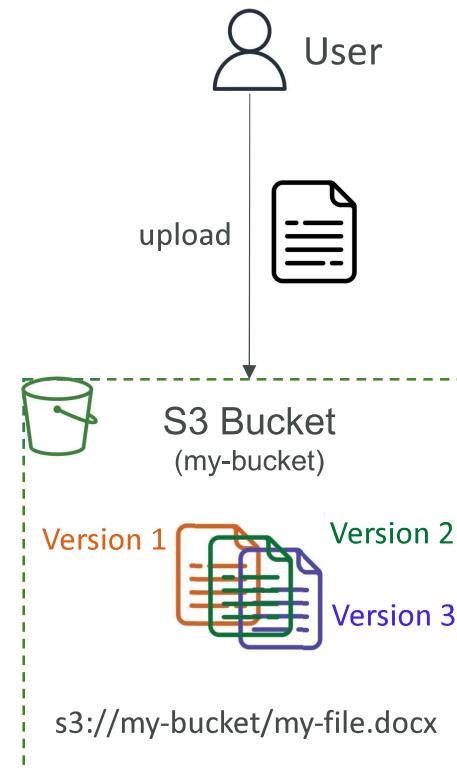
User

`http://demo-bucket.s3-website-us-west-2.amazonaws.com`
`http://demo-bucket.s3-website.us-west-2.amazonaws.com`



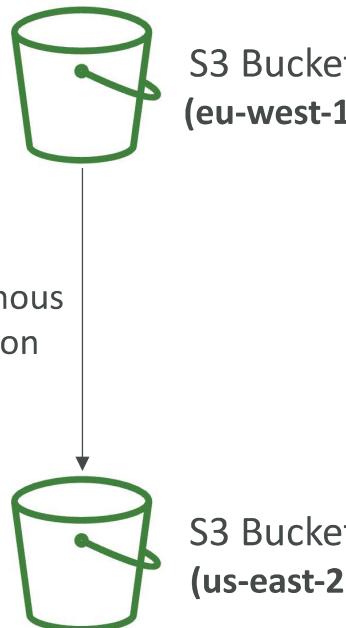
Amazon S3 - Versioning

- You can version your files in Amazon S3
- It is enabled at the **bucket level**
- Same key overwrite will change the “version”: 1, 2, 3....
- It is best practice to version your buckets
 - Protect against unintended deletes (ability to restore a version)
 - Easy roll back to previous version
- Notes:
 - Any file that is not versioned prior to enabling versioning will have version “null”
 - Suspending versioning does not delete the previous versions



Amazon S3 – Replication (CRR & SRR)

- Must enable Versioning in source and destination buckets
- Cross-Region Replication (CRR)
- Same-Region Replication (SRR)
- Buckets can be in different AWS accounts
- Copying is asynchronous
- Must give proper IAM permissions to S3
- Use cases:
 - CRR – compliance, lower latency access, replication across accounts
 - SRR – log aggregation, live replication between production and test accounts



S3 Storage Classes

- Amazon S3 Standard - General Purpose
- Amazon S3 Standard-Infrequent Access (IA)
- Amazon S3 One Zone-Infrequent Access
- Amazon S3 Glacier Instant Retrieval
- Amazon S3 Glacier Flexible Retrieval
- Amazon S3 Glacier Deep Archive
- Amazon S3 Intelligent Tiering
- Can move between classes manually or using S3 Lifecycle configurations

S3 Durability and Availability

- Durability:
 - High durability (99.99999999%, 11 9's) of objects across multiple AZ
 - If you store 10,000,000 objects with Amazon S3, you can on average expect to incur a loss of a single object once every 10,000 years
 - Same for all storage classes
- Availability:
 - Measures how readily available a service is
 - Varies depending on storage class
 - Example: S3 standard has 99.99% availability = not available 53 minutes a year

S3 Standard – General Purpose



- 99.99% Availability
 - Used for frequently accessed data
 - Low latency and high throughput
 - Sustain 2 concurrent facility failures
-
- Use Cases: Big Data analytics, mobile & gaming applications, content distribution...

S3 Storage Classes – Infrequent Access

- For data that is less frequently accessed, but requires rapid access when needed
- Lower cost than S3 Standard
- Amazon S3 Standard-Infrequent Access (S3 Standard-IA)
 - 99.9% Availability
 - Use cases: Disaster Recovery, backups
- Amazon S3 One Zone-Infrequent Access (S3 One Zone-IA)
 - High durability (99.99999999%) in a single AZ; data lost when AZ is destroyed
 - 99.5% Availability
 - Use Cases: Storing secondary backup copies of on-premise data, or data you can recreate



Amazon S3 Glacier Storage Classes

- Low-cost object storage meant for archiving / backup
- Pricing: price for storage + object retrieval cost
- **Amazon S3 Glacier Instant Retrieval**
 - Millisecond retrieval, great for data accessed once a quarter
 - Minimum storage duration of 90 days
- **Amazon S3 Glacier Flexible Retrieval** (formerly Amazon S3 Glacier):
 - Expedited (1 to 5 minutes), Standard (3 to 5 hours), Bulk (5 to 12 hours) – free
 - Minimum storage duration of 90 days
- **Amazon S3 Glacier Deep Archive** – for long term storage:
 - Standard (12 hours), Bulk (48 hours)
 - Minimum storage duration of 180 days



S3 Intelligent-Tiering



- Small monthly monitoring and auto-tiering fee
 - Moves objects automatically between Access Tiers based on usage
 - There are no retrieval charges in S3 Intelligent-Tiering
-
- *Frequent Access tier (automatic)*: default tier
 - *Infrequent Access tier (automatic)*: objects not accessed for 30 days
 - *Archive Instant Access tier (automatic)*: objects not accessed for 90 days
 - *Archive Access tier (optional)*: configurable from 90 days to 700+ days
 - *Deep Archive Access tier (optional)*: config. from 180 days to 700+ days

S3 Storage Classes Comparison

	Standard	Intelligent-Tiering	Standard-IA	One Zone-IA	Glacier Instant Retrieval	Glacier Flexible Retrieval	Glacier Deep Archive
Durability	99.999999999% == (11 9's)						
Availability	99.99%	99.9%	99.9%	99.5%	99.9%	99.99%	99.99%
Availability SLA	99.9%	99%	99%	99%	99%	99.9%	99.9%
Availability Zones	>= 3	>= 3	>= 3	1	>= 3	>= 3	>= 3
Min. Storage Duration Charge	None	None	30 Days	30 Days	90 Days	90 Days	180 Days
Min. Billable Object Size	None	None	128 KB	128 KB	128 KB	40 KB	40 KB
Retrieval Fee	None	None	Per GB retrieved	Per GB retrieved	Per GB retrieved	Per GB retrieved	Per GB retrieved

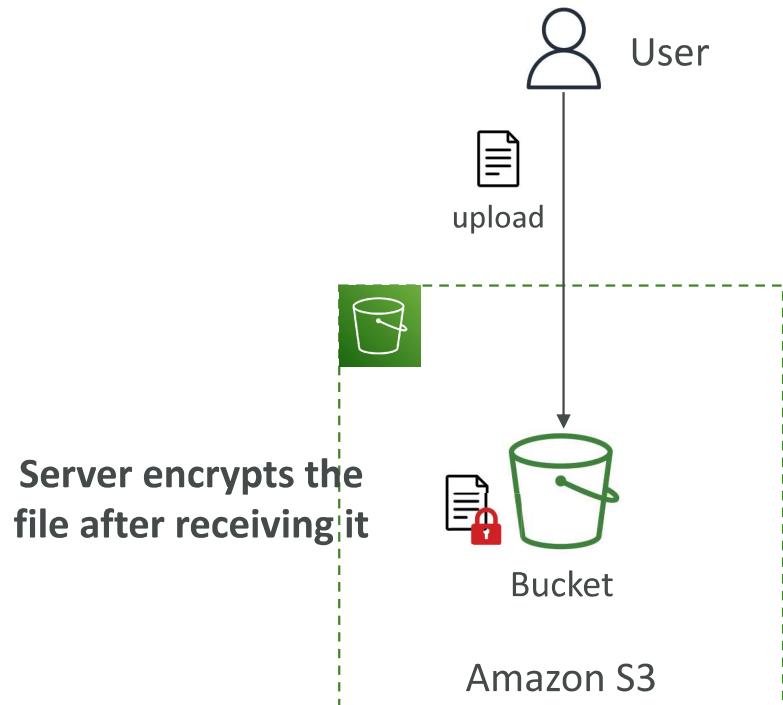
S3 Storage Classes – Price Comparison

Example: us-east-1

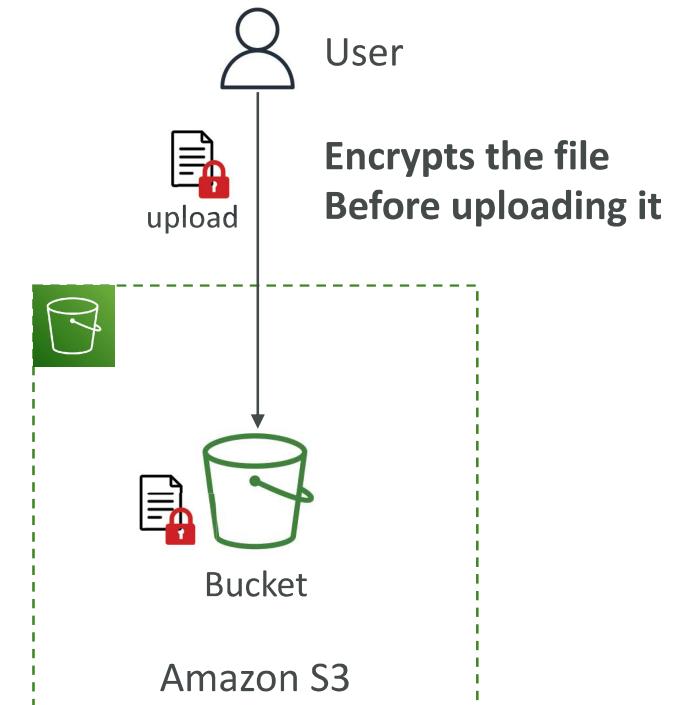
	Standard	Intelligent-Tiering	Standard-IA	One Zone-IA	Glacier Instant Retrieval	Glacier Flexible Retrieval	Glacier Deep Archive
Storage Cost (per GB per month)	\$0.023	\$0.0025 - \$0.023	%0.0125	\$0.01	\$0.004	\$0.0036	\$0.00099
Retrieval Cost (per 1000 request)	GET: \$0.0004 POST: \$0.005	GET: \$0.0004 POST: \$0.005	GET: \$0.001 POST: \$0.01	GET: \$0.001 POST: \$0.01	GET: \$0.01 POST: \$0.02	GET: \$0.0004 POST: \$0.03 Expedited: \$10 Standard: \$0.05 Bulk: free	GET: \$0.0004 POST: \$0.05 Standard: \$0.10 Bulk: \$0.025
Retrieval Time	Instantaneous						Expedited (1 – 5 mins) Standard (3 – 5 hours) Bulk (5 – 12 hours)
Monitoring Cost (per 1000 objects)		\$0.0025					

S3 Encryption

Server-Side Encryption (Default)



Client-Side Encryption



Shared Responsibility Model for S3



- Infrastructure (global security, durability, availability, sustain concurrent loss of data in two facilities)
- Configuration and vulnerability analysis
- Compliance validation
- S3 Versioning
- S3 Bucket Policies
- S3 Replication Setup
- Logging and Monitoring
- S3 Storage Classes
- Data encryption at rest and in transit

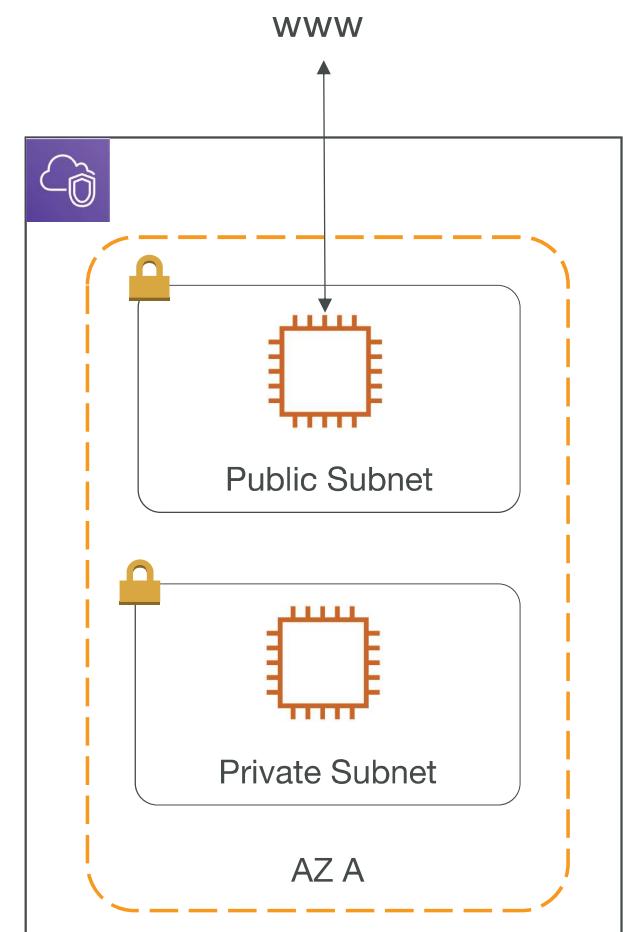
VPC Section

VPC – Crash Course

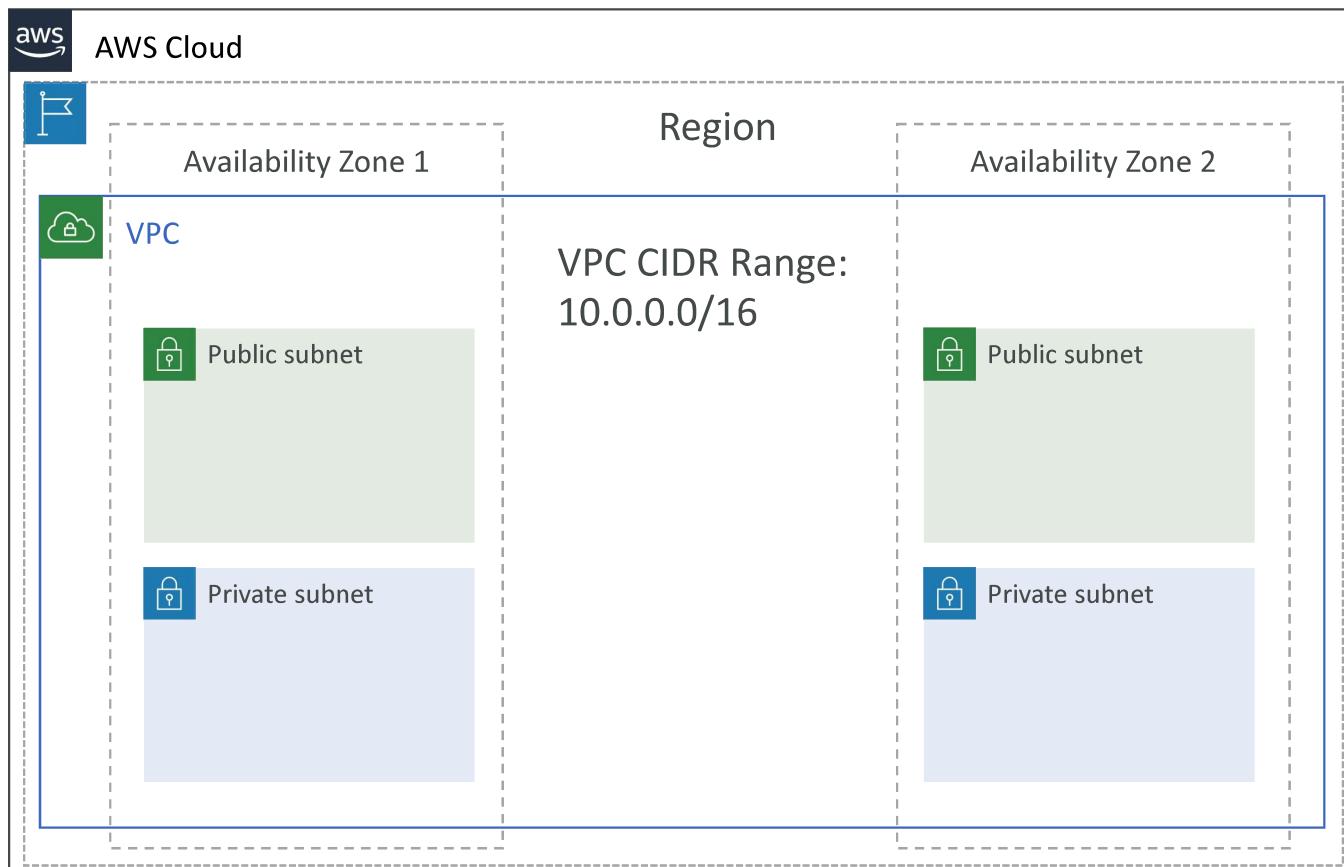
- VPC is something you should know in depth for the AWS Certified Solutions Architect Associate & AWS Certified SysOps Administrator
- At the AWS Certified Cloud Practitioner Level, you should know about:
 - VPC, Subnets, Internet Gateways & NAT Gateways
 - Security Groups, Network ACL (NACL), VPC Flow Logs
 - VPC Peering, VPC Endpoints
 - Site to Site VPN & Direct Connect
 - Transit Gateway
- I will just give you an overview, less than 1 or 2 questions at your exam.
- We'll have a look at the “default VPC” (created by default by AWS for you)
- There is a summary lecture at the end. It's okay if you don't understand it all

VPC & Subnets Primer

- **VPC - Virtual Private Cloud:** 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**.

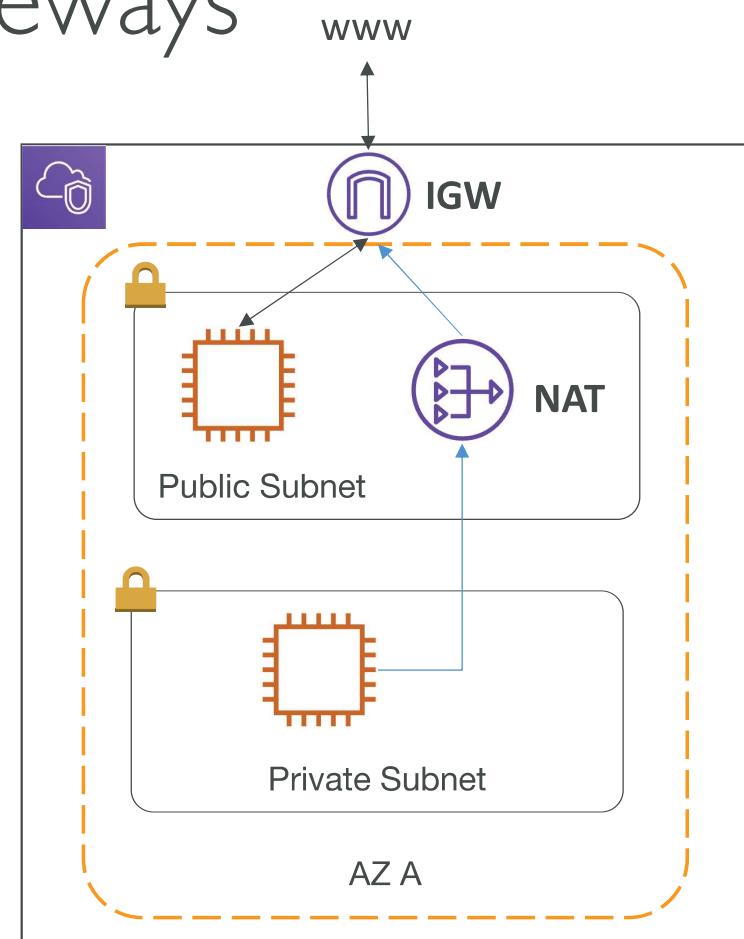


VPC Diagram



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



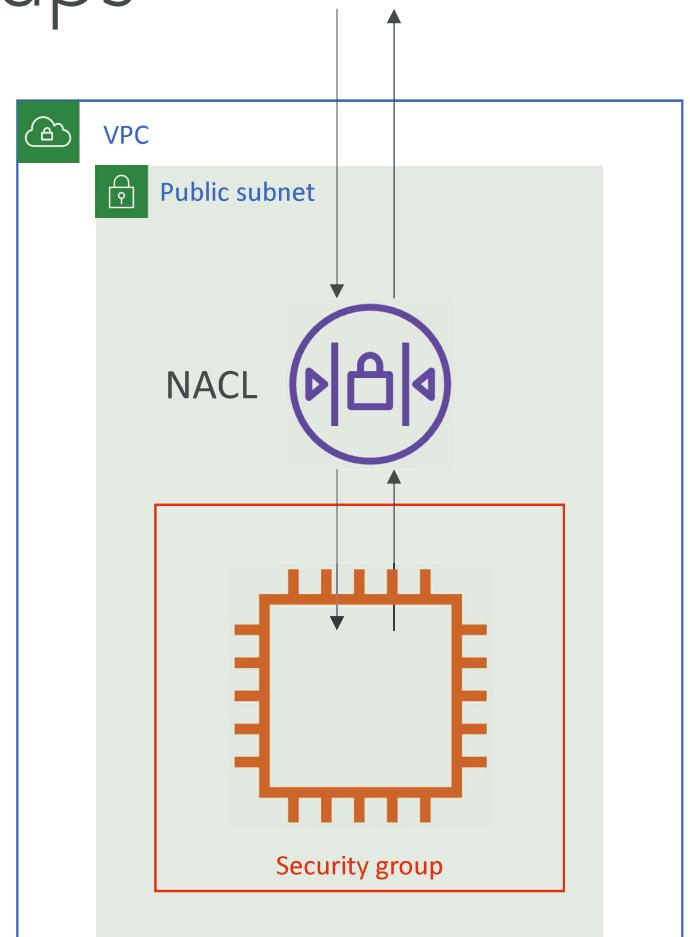
Network ACL & Security Groups

- **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



Network ACLs vs 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)

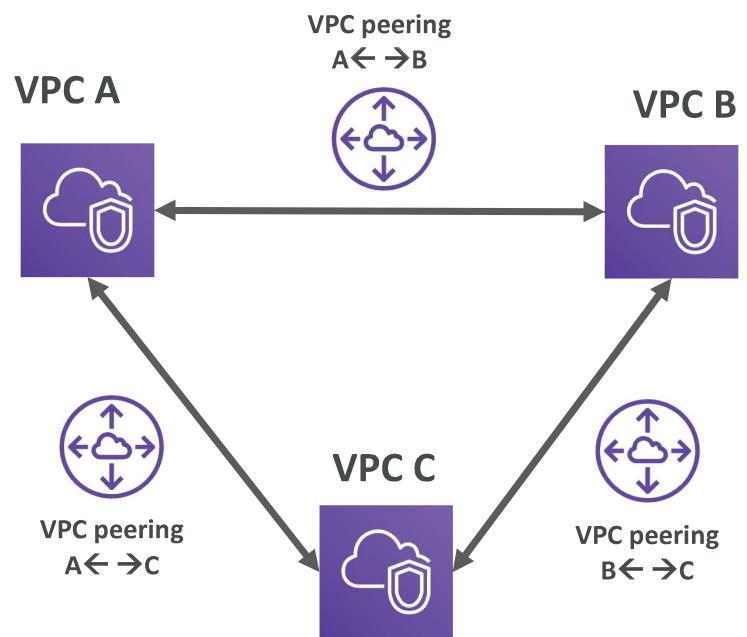
VPC Flow Logs



- 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

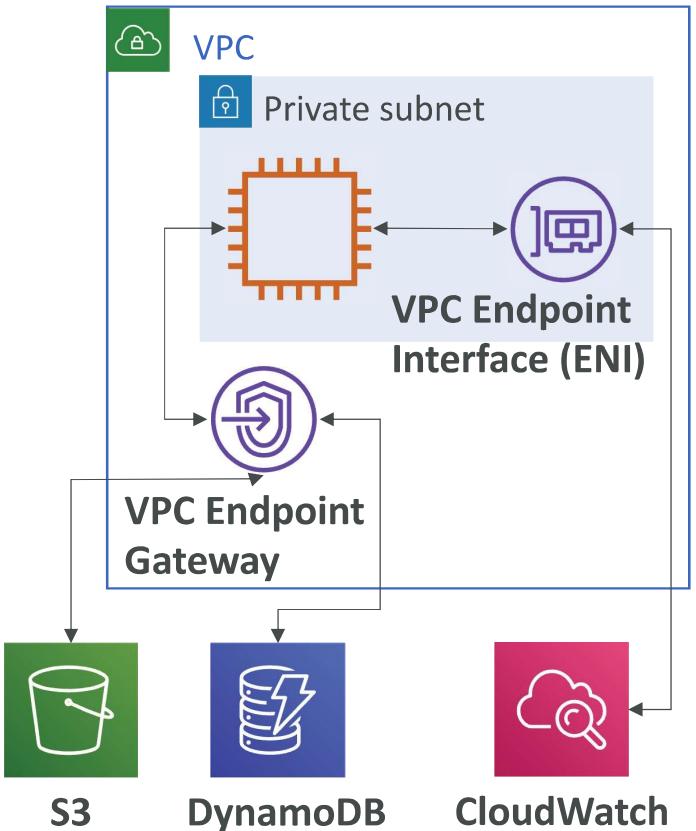
- Connect two VPC, privately using AWS' network
- Make them behave as if they were in the same network
- Must not have overlapping CIDR (IP address range)
- VPC Peering connection is **not transitive** (must be established for each VPC that need to communicate with one another)



VPC Endpoints



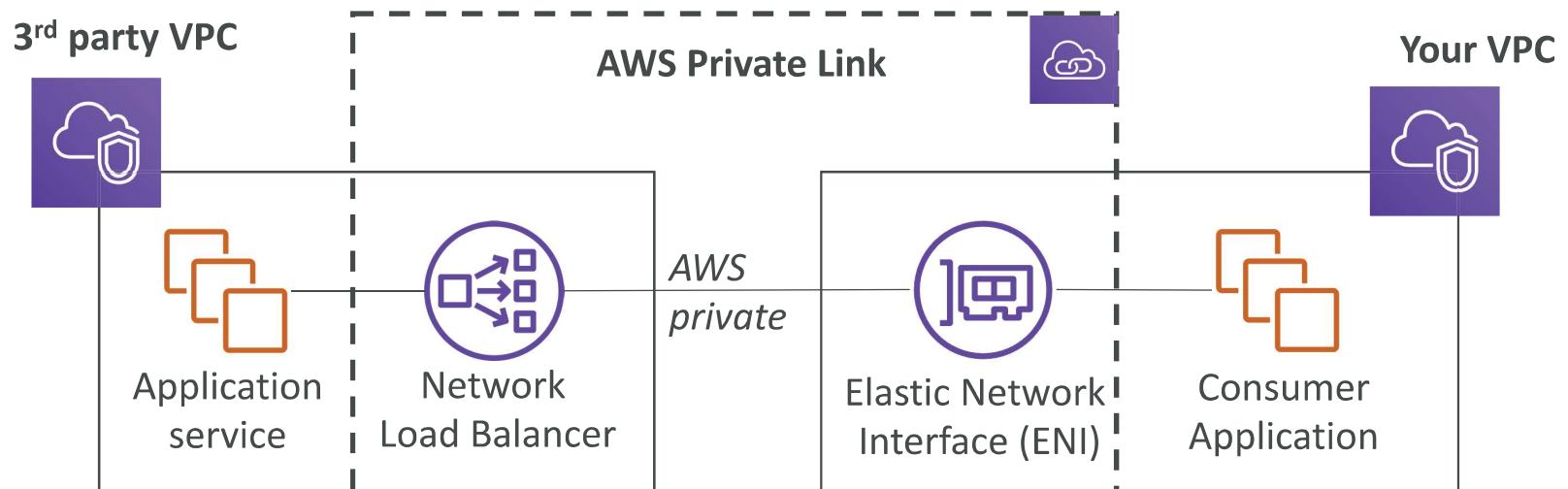
- Endpoints allow you to connect to AWS Services **using a private network** instead of the public www network
- This gives you enhanced security and lower latency to access AWS services
- VPC Endpoint **Gateway**: S3 & DynamoDB
- VPC Endpoint **Interface**: the rest



AWS PrivateLink (VPC Endpoint Services)



- Most secure & scalable way to expose a service to 1000s of VPCs
- Does not require VPC peering, internet gateway, NAT, route tables...
- Requires a network load balancer (Service VPC) and ENI (Customer VPC)



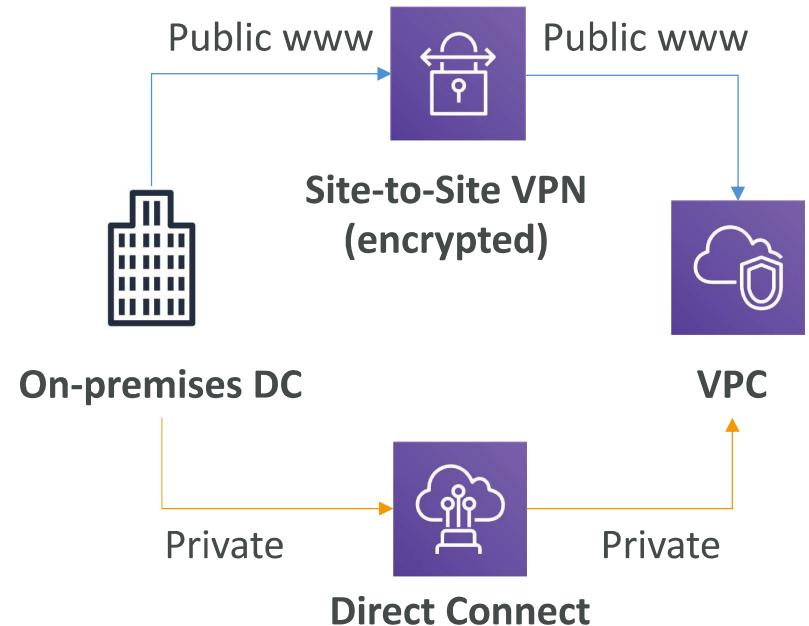
Site to Site VPN & Direct Connect

- Site to Site VPN

- Connect an on-premises VPN to AWS
- The connection is automatically encrypted
- Goes over the public internet

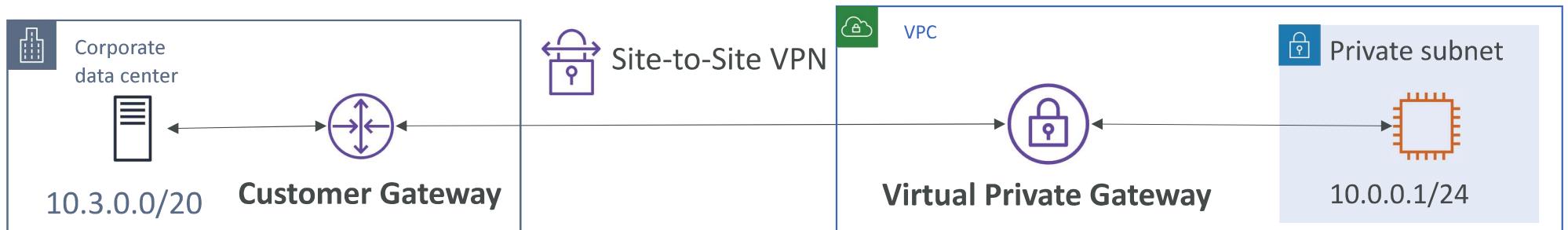
- Direct Connect (DX)

- 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



Site-to-Site VPN

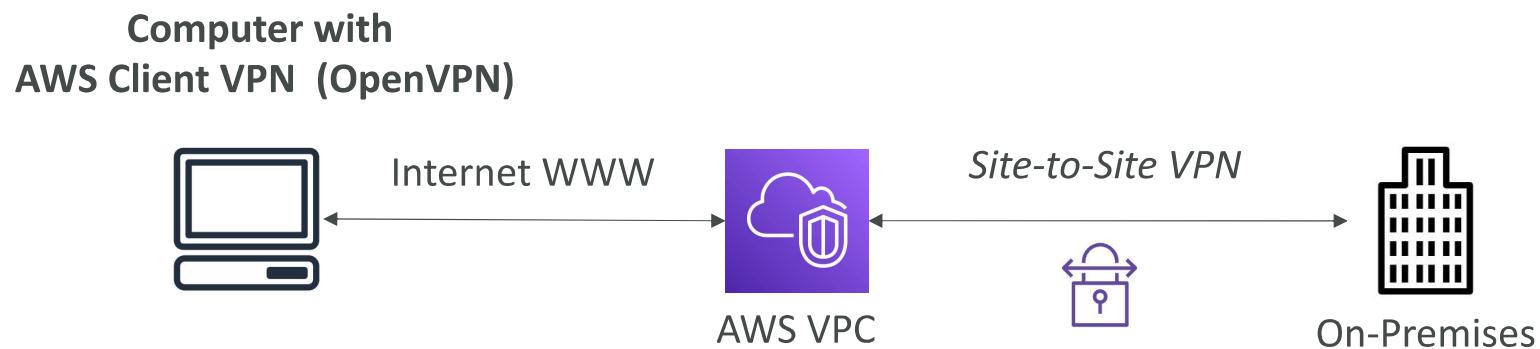
- On-premises: must use a **Customer Gateway** (CGW)
- AWS: must use a **Virtual Private Gateway** (VGW)



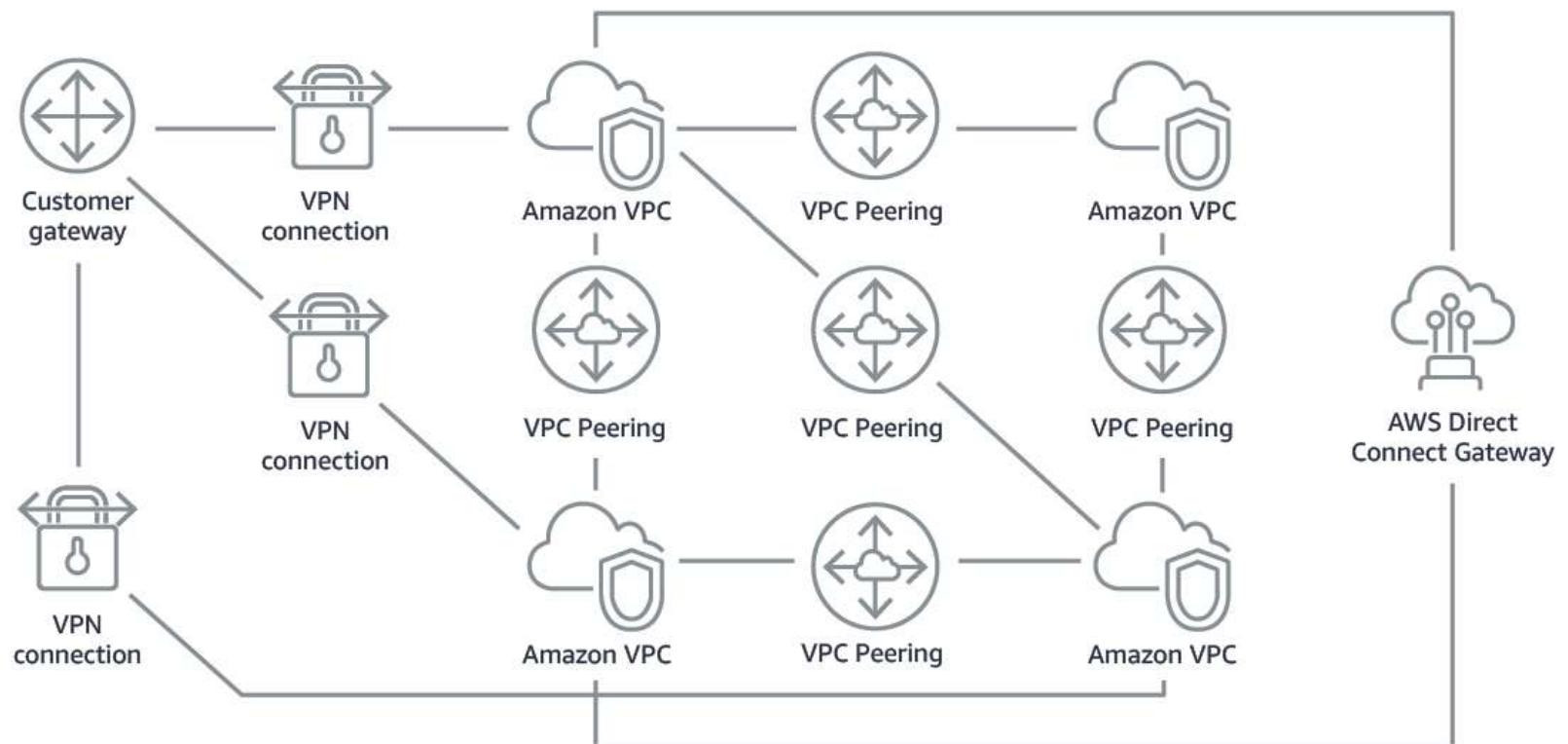
AWS Client VPN



- Connect from your computer using OpenVPN to your private network in AWS and on-premises
- Allow you to connect to your EC2 instances over a private IP (just as if you were in the private VPC network)
- Goes over public Internet



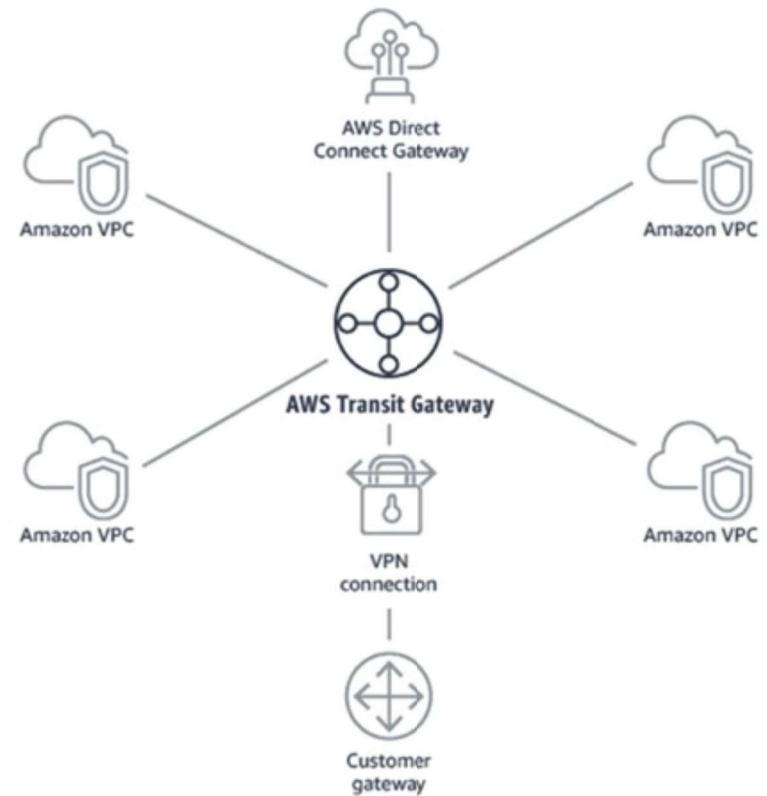
Network topologies can become complicated



Transit Gateway



- For having transitive peering between thousands of VPC and on-premises, hub-and-spoke (star) connection
- One single Gateway to provide this functionality
- Works with Direct Connect Gateway, VPN connections



VPC Closing Comments

- **VPC:** Virtual Private Cloud
- **Subnets:** Tied to an AZ, network partition of the VPC
- **Internet Gateway:** at the VPC level, provide Internet Access
- **NAT Gateway / Instances:** give internet access to private subnets
- **NACL:** Stateless, subnet rules for inbound and outbound
- **Security Groups:** Stateful, operate at the EC2 instance level or ENI
- **VPC Peering:** Connect two VPC with non overlapping IP ranges, nontransitive

VPC Closing Comments

- **VPC Endpoints:** Provide private access to AWS Services within VPC
- **PrivateLink:** Privately connect to a service in a 3rd party VPC
- **VPC Flow Logs:** network traffic logs
- **Site to Site VPN:** VPN over public internet between on-premises DC and AWS
- **Client VPN:** OpenVPN connection from your computer into your VPC
- **Direct Connect:** direct private connection to AWS
- **Transit Gateway:** Connect thousands of VPC and on-premises networks together