FPT Software | Corporate Training Center

Module 5 : Amazon Networking VPC & Compute Service EC2

OPEN MIND - OPEN DOORS

Topics

- What is Amazon VPC , EC2 ?
- Core Components
- Key Features

Virtual Private Cloud

VPC Introduction

amazon web services

Virtual Private Cloud (VPC)

Provision a **logically isolated section of the AWS Cloud** where you can launch AWS resources in a **virtual network** that you define

Virtual Private Cloud

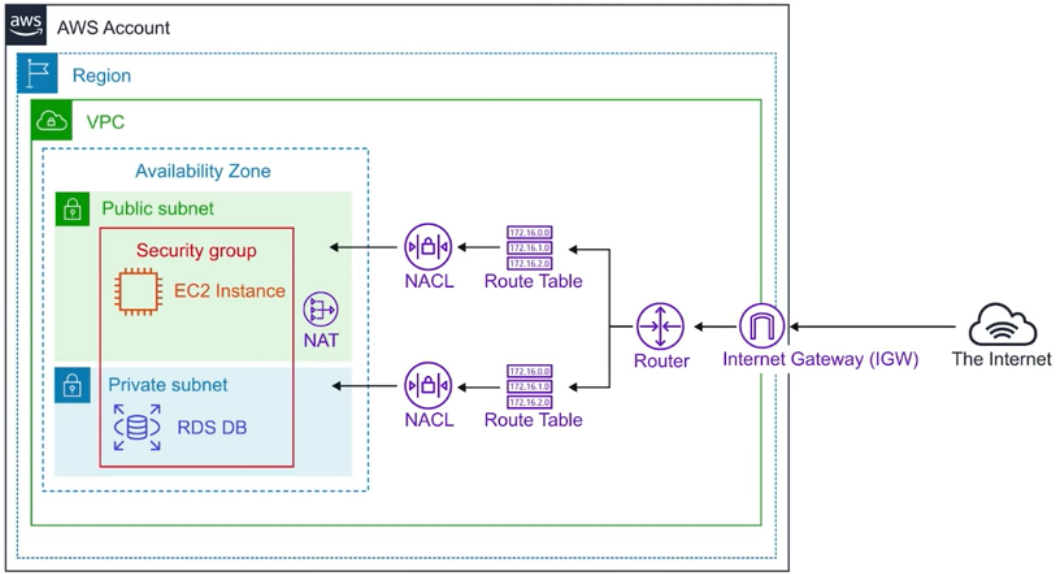
Core Components

amazon web services

Introduction to VPC

Think of a AWS VPC as your own **personal data centre.**

Gives you complete control over your virtual networking environment



amazon web services

Core Components

Combining these components and services is what makes up your VPC.

Internet Gateway (IGW)

Virtual Private Gateway (VPN Gateway)

Routing Tables

Network Access Control Lists (NACLs) - Stateless

Security Groups (SG) Stateful

Public Subnets

Private Subnets

Nat Gateway

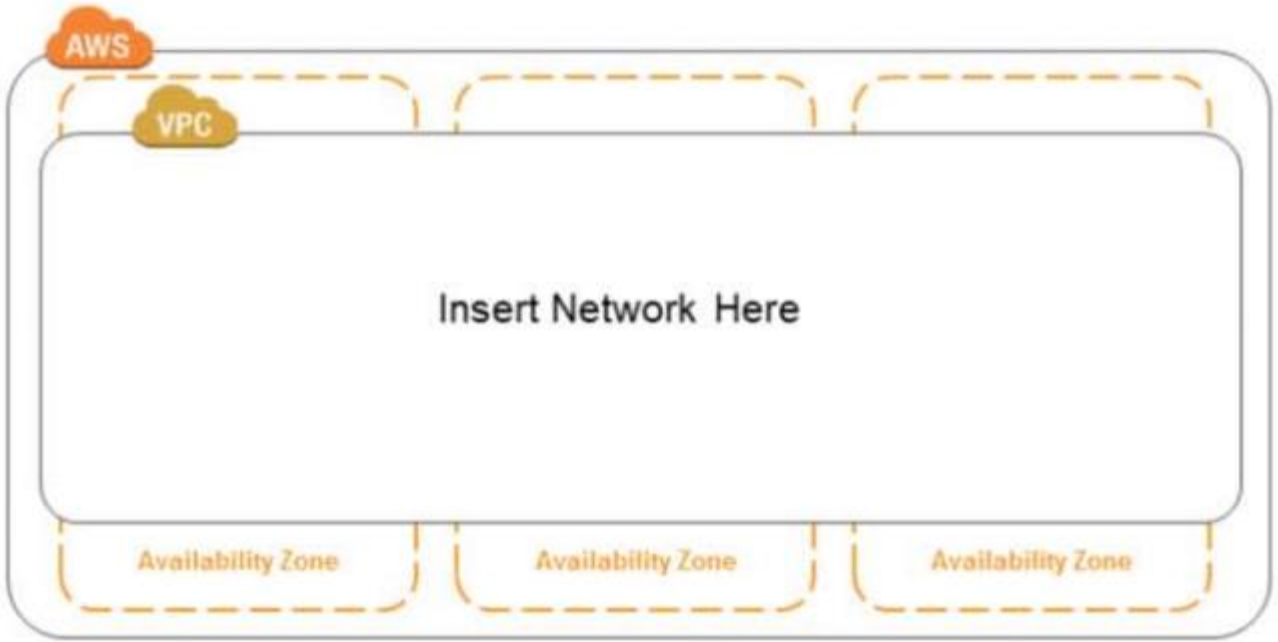
Customer Gateway

VPC Endpoints

VPC Peering

Core components

- Select an AWS region for your VPC

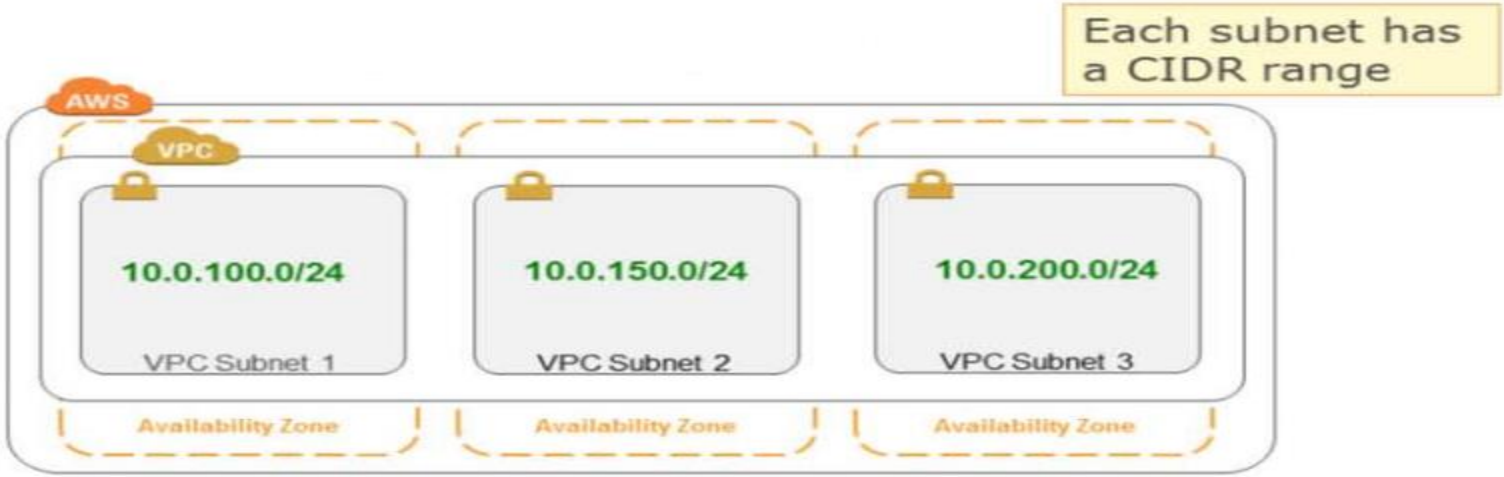


Core components

- Specify the size of our network with a CIDR block

Core components

- Create **subnets** inside the VPC
  - Subnets **do not** span Availability Zones



Core components

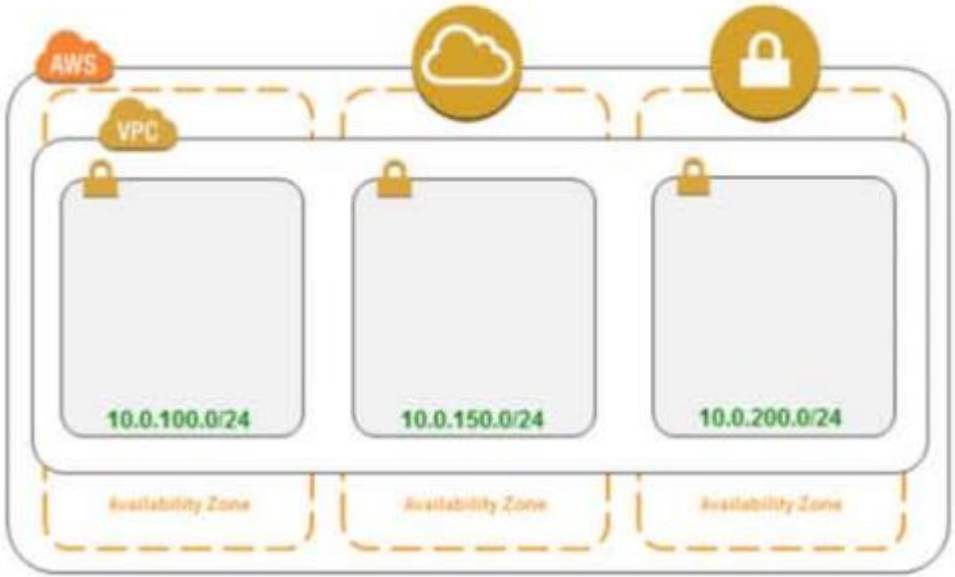
- Attach gateway devices to network

Core components

•VPN Gateway(VGW) allows subnet(s) to route to on-premises network over IPSEC VPN

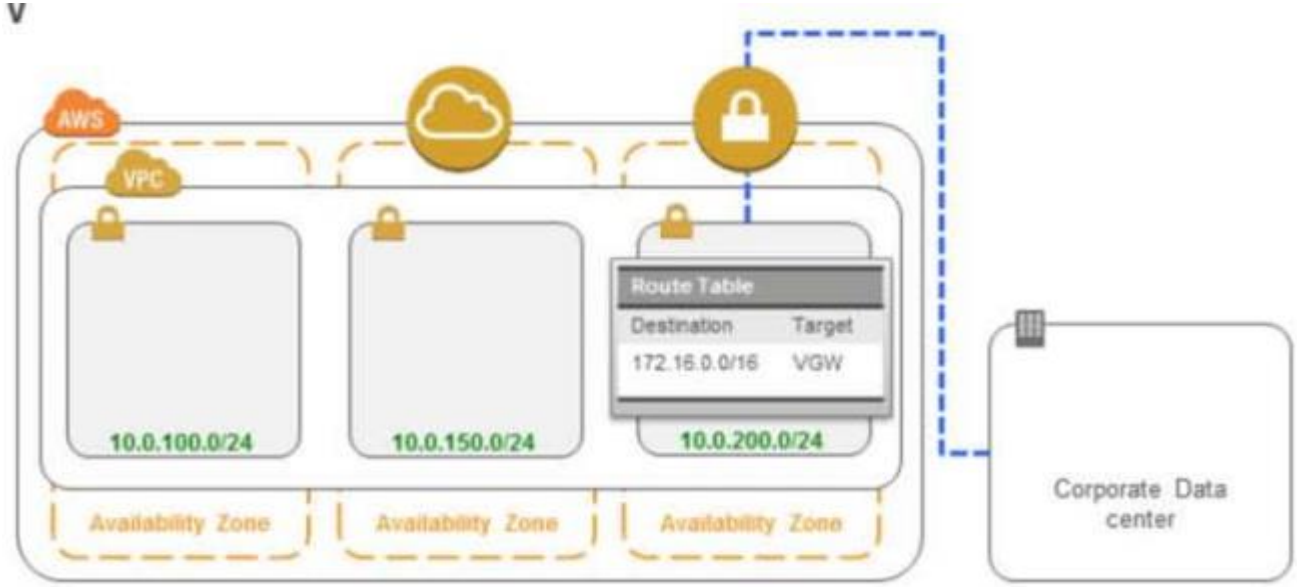
Core components

- Define custom routing rule(s) for each subnet



Core components

•Private subnet with IPSEC VPN to corporate datacenter via VGW



Core components

•Public subnet with inbound/outbound Internet connectivity via IGW

amazon web services

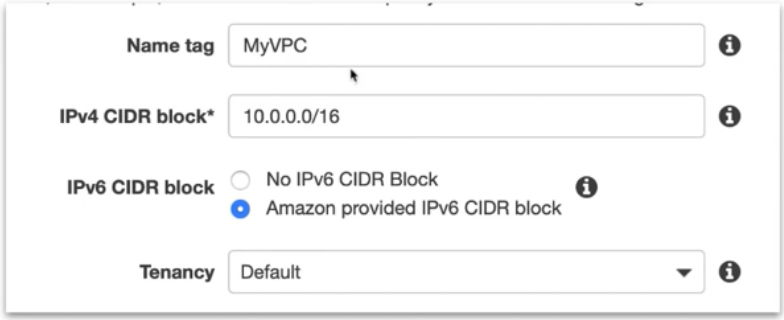
Virtual Private Cloud

Key Features

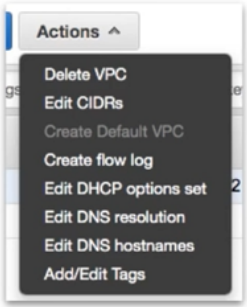
amazon web services

Key Features

- VPCs are **Region Specific** they do not span regions
- You can create upto **5 VPC** per region.
- Every region comes with a default VPC
- You can have **200 subnets** per VPC
- You can use **IPv4 Cidr Block** and in addition to a **IPv6 Cidr Blocks** (the address of the VPC)
- **Cost nothing:** VPC's, Route Tables, Nacls, Internet Gateways, Security Groups and Subnets, VPC Peering
- **Some things cost money:** eg. NAT Gateway, VPC Endpoints, VPN Gateway, Customer Gateway
- **DNS hostnames** (should your instance have domain name addresses)



IPv6 Cidr Block 2600:1f16:9e0:8d00::/56



Disabled by default, turn on **hostnames**

amazon web services

Virtual Private Cloud

Default VPC

amazon web services

Default VPC

AWS has a default VPC in every region so you can **immediately** deploy instances.

- Create a VPC with a size /16 IPv4 CIDR block (172.31.0.0/16).
- Create a size /20 **default subnet in each Availability Zone**.
- Create an **Internet Gateway** and connect it to your default VPC.
- Create a **default security group** and associate it with your default VPC.
- Create a **default network access control list (NACL)** and associate it with your default VPC.
- Associate the **default DHCP** options set for your AWS account with your default VPC.
- *When you create a VPC, it automatically has a main route table

Virtual Private Cloud

Default Everywhere IP

amazon web services

0.0.0.0/0

0.0.0.0/0 is also know as **default**

It represents **all possible IP addresses**

When we specify **0.0.0.0/0** in our route table for IGW we are allow internet access
When we specific **0.0.0.0/0** in our security groups inbound rules we are allowing all traffic from the internet access our public resources

When you see **0.0.0.0/0**, just *think* of giving access from anywhere or the internet.

Virtual Private Cloud

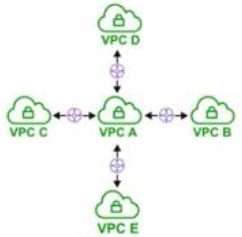
VPC Peering

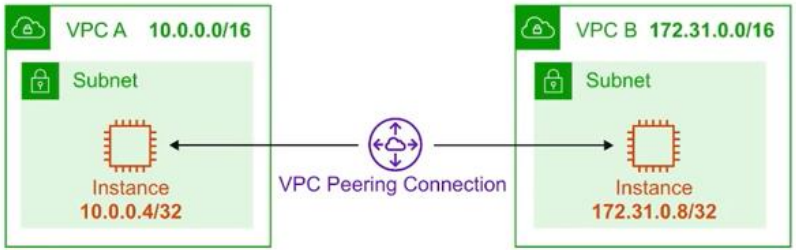
amazon web services

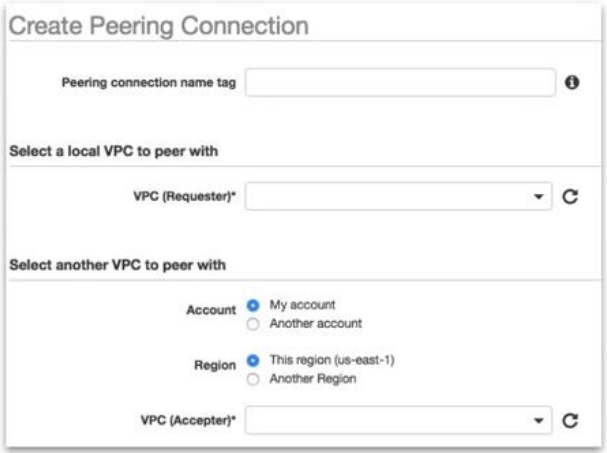
VPC Peering

VPC Peering allows you to connect one VPC with another over a **direct network route** using **private IP addresses.**

- Instances on peered VPCs **behave** just like they are on the **same network**
- Connect VPCs across **same** or **different AWS accounts** and **regions**
- Peering uses a **Star Configuration: 1 Central VPC - 4 other VPCs**
- **No Transitive Peering** (peering must take place directly between VPCs)
  - Needs a one to one connect to immediate VPC
- **No Overlapping CIDR Blocks**







amazon web services

amazon web services

Route Tables

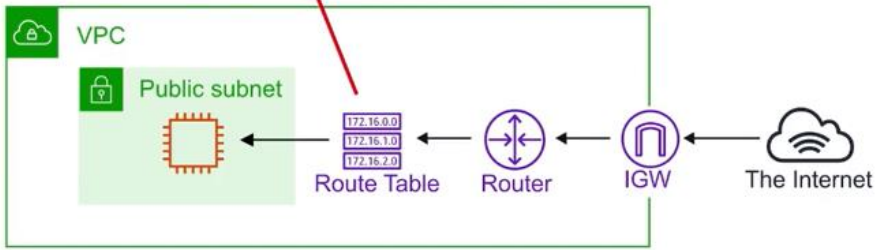
Route tables are used to determine where **network traffic is directed**

Each **subnet** in your VPC **must be associated** with a route table

Each record is called a “route”

A subnet can only be associated **with one route table at a time**, but you can associate multiple subnets with the same route table.

Destination	Target	Status	Propagated
10.0.0.0/16	local	active	No
0.0.0.0/0	igw-19e3a2e134fe086e2	active	No



Virtual Private Cloud

Internet Gateway

amazon web services

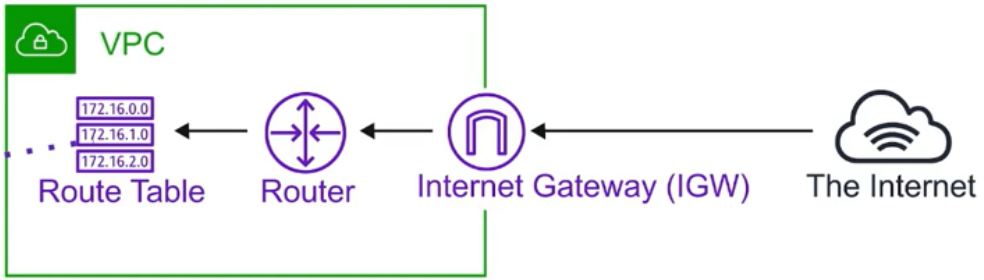
Internet Gateway (IGW)

The Internet Gateway allows **your VPC access to the internet**.

IGW does ✌️ two things:

1. provide a target in your VPC route tables for internet-routable traffic
2. perform network address translation (NAT) for instances that have been assigned **public IPv4 addresses**.

Destination	Target
10.0.0.0/16	local
0.0.0.0/0	**igw-id**



To route out to the internet you need to add in your route tables you need to add a route
To the internet gateway and set the Destination to be **0.0.0.0/0**

Virtual Private Cloud

Bastions / Jumpbox

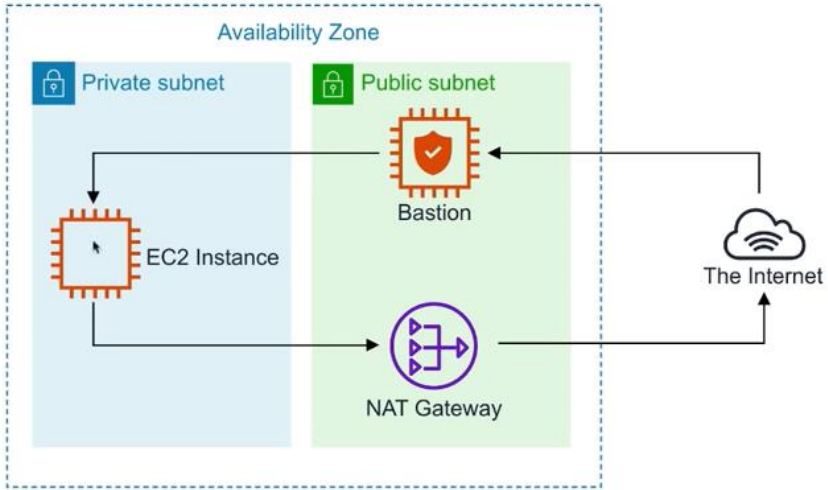
amazon web services

Bastion / Jumpbox

Bastions are EC2 instances which are security harden. They are designed to help you gain access to your EC2 Instances via SSH or RCP That are in a **private subnet**.

They are also known as Jump boxes because you are jumping from one box to access another.

NAT Gateways/Instances are only intended for EC2 instances to gain outbound access to the internet for things such as security updates. NATs cannot/should not be used as Bastions



System Manager's **Sessions Manager** replaces the need for Bastions

Virtual Private Cloud

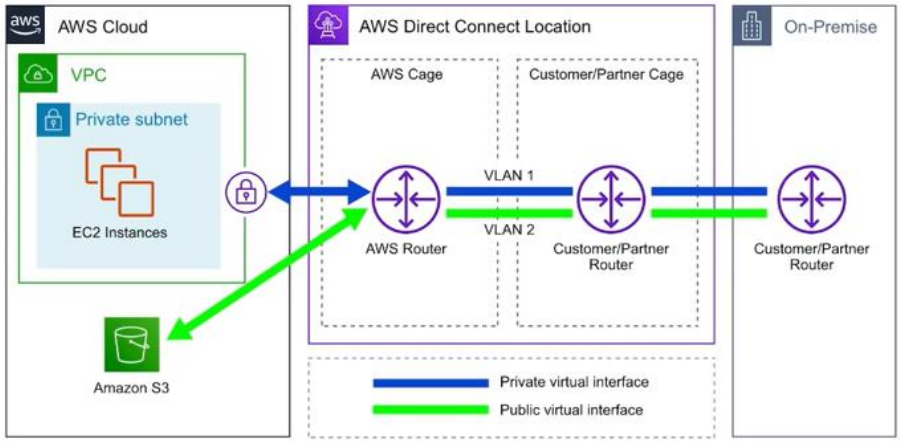
Direct Connect

amazon web services

Direct Connect

AWS Direct Connect is the AWS solution for establishing
dedicated network connections from on-premises locations to AWS.

Very fast network Lower Bandwidth **50M-500M** or Higher Bandwidth **1GB** or **10GB**



- Helps **reduce network costs** and **increase bandwidth throughput**. (great for high traffic networks)
- Provides a **more consistent network experience** than a typical internet-based connection. (reliable and secure)

amazon web services

Virtual Private Cloud

VPC Endpoints Introduction

amazon web services

VPC Endpoints

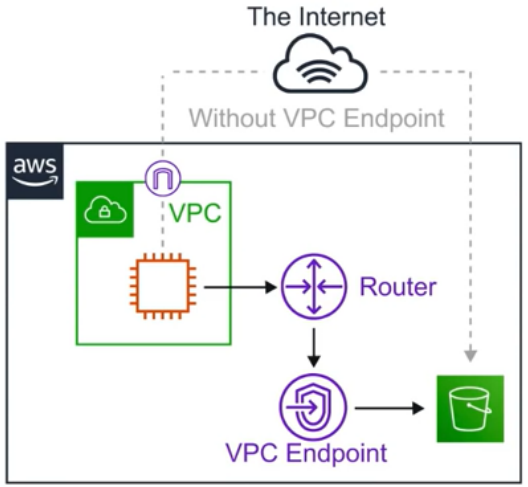
Think of a secret tunnel where you don't have to leave the AWS network

VPC Endpoints allow you to **privately connect your VPC to other AWS services**, and VPC endpoint services.

There are ✌️ **2 Types** of VPC Endpoints

1. Interface Endpoints
2. Gateway Endpoints

- **Eliminates** the need for an **Internet Gateway**, **NAT device**, **VPN connection**, or **AWS Direct Connect** connections.
- **Instances** in the VPC **do not require a public IP address** to communicate with service resources.
- **Traffic** between your VPC and other services **does not leave the AWS network**.
- **Horizontally scaled**, **redundant**, and **highly available** VPC component.
- Allows secure communication between instances and services - **without adding availability risks or bandwidth constraints** on your traffic.



amazon web services

Virtual Private Cloud

Interface Endpoints

amazon web services

Interface Endpoints

Interface Endpoints are **Elastic Network Interfaces (ENI)** with a **private IP address**.
They serve as an entry point for traffic going to a supported service.

Interface Endpoints are powered by **AWS PrivateLink**

Access services hosted on AWS easily and securely by keeping your network traffic within the AWS network

Pricing per VPC endpoint per AZ ($/hour)	**0.01**	~$7.5 / mo
Pricing per GB data processed ($)	**0.01**	

Interface Endpoints support the following AWS Services...

- API Gateway
- CloudFormation
- CloudWatch
- Kinesis
- SageMaker
- Codebuild
- AWS Config
- EC2 API
- ELB API
- AWS KMS
- Secrets Manager
- Security Token Service
- Service Catalog
- SNS
- SQS
- Systems Manager
- Marketplace Partner Services
- Endpoint Services in other AWS accounts

amazon web services

Virtual Private Cloud

Gateway Endpoints

amazon web services

VPC Gateway Endpoints

VPC Gateway Endpoints are **Free!**

A **Gateway Endpoint** is a gateway that is a target **for a specific route** in your **route table**, used for traffic destined for a supported AWS service.

To create a Gateway Endpoint, you must specify the VPC in which you want to create the endpoint, and the service to which you want to establish the connection.

AWS Gateway Endpoint currently only supports 2 services...

- Amazon S3
- DynamoDB

Virtual Private Cloud

Network Access Control List Introduction

amazon web services

Network Access Control List (NACLs)

An (optional) layer of security that acts
As a **firewall** for controlling traffic **in and out of subnet(s)**

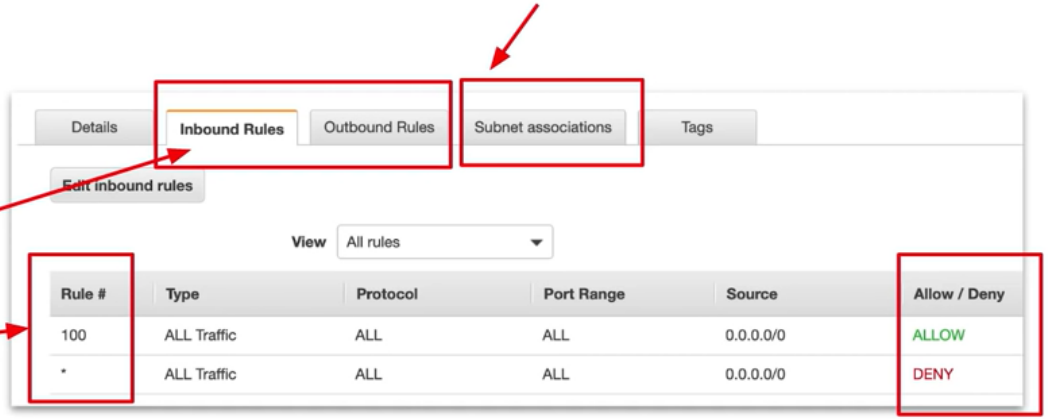
NACLs – Introduction

NACLs acts as a **virtual firewall** at the subnet level

VPCs automatically get a default NACL

Subnets are associated with NACLs. Subnets can only belong to a single NACL.

Each NACL contains a set of rules that can **allow** or **deny** traffic **into (inbound) and out of (outbound) subnets**



Rule # determines the **order of evaluation**. From lowest to highest. The highest rule # can be 32766 and its recommended to work in 10 or 100 increments.

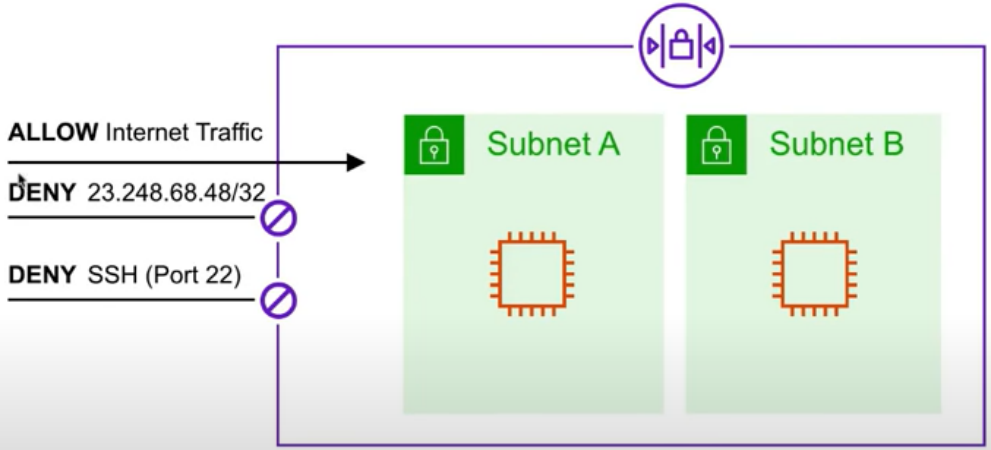
You can allow or deny traffic. You **could block a single IP address** (You can't do this with Security Groups)

amazon web services

NACLs – Use Case

We determine there is a malicious actor at a specific IP address is trying to access our instances so we block their IP

We never need to SSH into instances so we add a DENY for these subnets. This is just an additional measure in case our Security Groups SSH port was left open.



Virtual Private Cloud

SG

Security Groups Introduction

amazon web services

sg-exampro

Security Groups

A virtual firewall that controls the traffic to and from EC2 Instances

Security Groups – Introduction

Security Groups acts as a **virtual firewall** at the instance level

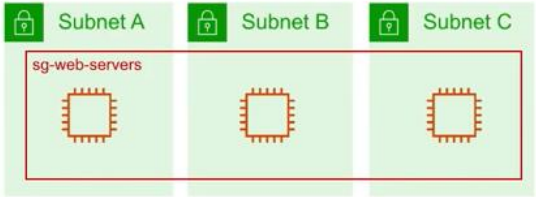
Security Groups are associated with EC2 instances

provide security at the **protocol** and **port** access level.

Each Security Group contains a set of rules that filter traffic coming **into (inbound) and out of (outbound)** EC2 instances.

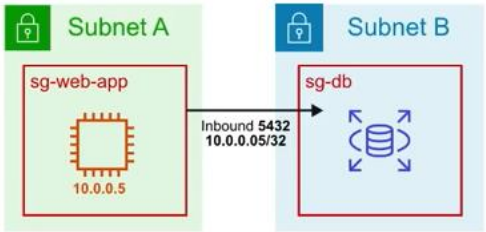
There are no 'Deny' rules. **All traffic is blocked by default** unless a rule specifically allows it.

Multiple Instances across multiple subnets can belong to a **Security Group.**

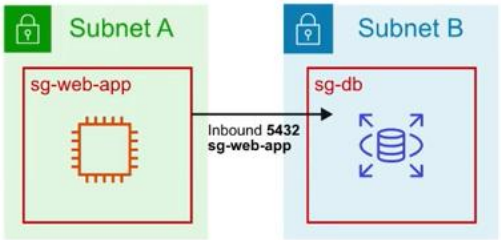


amazon web services

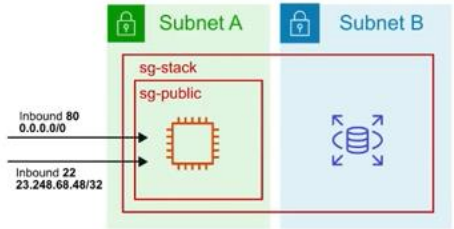
Security Groups – Use Case



You can specify the source to be an IP range or
A specific ip (/32 is a specific IP Address)



You can specify the source to be another security group



An instance can **belong to multiple Security Groups**, and rules are **permissive** (instead of restrictive). Meaning if you have one security group which has no Allow and you add an allow to another than it will Allow.

amazon web services

Security Groups – Limits

You can have **upto 10,000 Security Groups in a Region** (default is 2,500)

You can have **60 inbound rules** and **60 outbound rules** per security group

16 Security Groups per Elastic Network Interface (ENI) (default is 5)

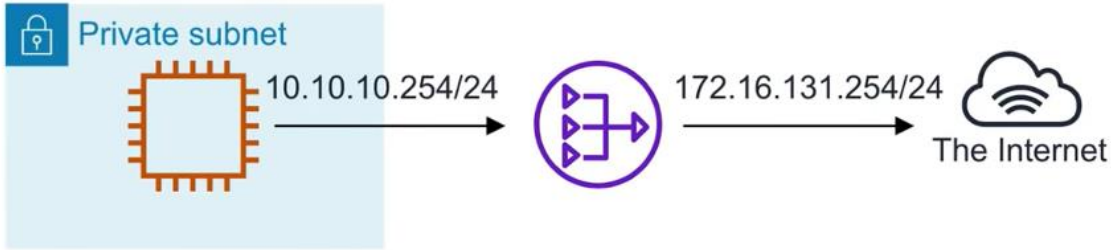
Virtual Private Cloud

Network Address Translation Introduction

amazon web services

Network Address Translation (NAT)

Network Address Translation (NAT) is the
method of **re-mapping** one IP address space into another.



If you have a private network and you need to help gain outbound access to the internet you would need to use a NAT gateway to remap the Private IPs

If you have two networks which have conflicting network addresses you can use a NAT to make the addresses more agreeable

Virtual Private Cloud

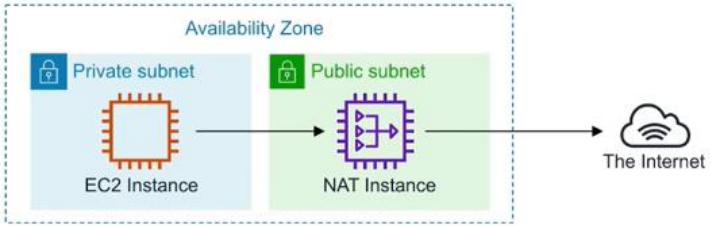
NAT Instances VS NAT Gateways

amazon web services

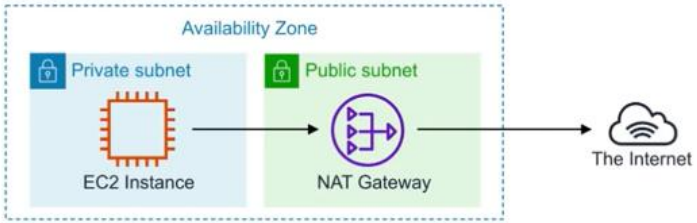
NAT Instances vs NAT Gateways

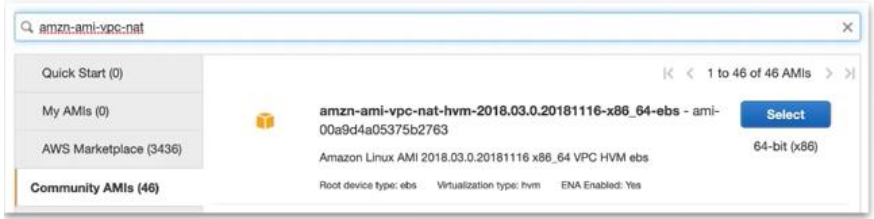
NATs have to run within a **Public Subnet**

NAT Instances (legacy) are individual EC2 instances. Community AMIs exist to launch NAT Instances.



NAT Gateways is a managed service which launches redundant instances within the selected AZ.





amazon web services

EC2

Elastic Cloud Compute Introduction

amazon web services

Elastic Compute Cloud (EC2)

Cloud Computing Service

Choose your **OS, Storage, Memory, Network Throughput.**

Launch and SSH into your server **within minutes**.

amazon web services

Introduction to EC2

Elastic Compute Cloud (EC2) is a **highly configurable server**.
EC2 is resizable **compute capacity.** It takes **minutes** to launch new instances.
Anything and everything on AWS uses EC2 Instance underneath.

Choose your OS **via**
Amazon Machine Image (AMI)

Red Hat ubuntu Amazon Linux SUSE

Choose you **Instance Type**

t2.nano
$0.0065/hour ($4.75/month)
1 vCPU 0.5GB Mem

C4.8xlarge
$1.591/hour ($1161.43/month)
36 vCPU 60GB Mem 10 Gigabit performance

Add Storage **(EBS, EFS)**

SSD HDD Virtual Magnetic Tape Multiple Volumes

Configure your Instance

Security Groups, Key Pairs, UserData, IAM Roles, Placement Groups

amazon web services

EC2

Instance Types

amazon web services

EC2 – Instance Types and Usage

General Purpose

A1 T3 T3a T2 M5 M5a M4

balance of compute, memory and networking resources

Use-cases web servers and code repositories

Compute Optimized

C5 C5n C4

Ideal for compute bound applications that benefit from high performance processor

Use-cases scientific modeling, dedicated gaming servers and ad server engines

Memory Optimized

R5 R5a X1e X1 High Memory z1d

fast performance for workloads that process large data sets in memory.

Use-cases in-memory caches, in-memory databases, real time big data analytics

Accelerated Optimized

P3 P2 G3 F1

hardware accelerators, or co-processors

Use-cases Machine learning, computational finance, seismic analysis, speech recognition

Storage Optimized

I3 I3en D2 H1

high, sequential read and write access to very large data sets on local storage

Use-cases NoSQL, in-memory or transactional databases, data warehousing

EC2

Instance Sizes

amazon web services

EC2 – Instance Sizes

EC2 Instance Sizes **generally double** in price and key attributes

Name	vCPU	RAM (GIB)	On-Demand per hour	On-Demand per month
t2.small	1	12	$0.023	$16.79
t2.medium	2	24	$0.0464	$33.87
t2.large	2	36	$0.0928	$67.74
t2.xlarge	4	54	$0.1856	$135.48

EC2

Instance Profiles

amazon web services

EC2 - Instance Profile

Instead of embedding your AWS credentials (Access Key and Secret) in your code so your Instance has permissions to access certain services you can **Attach a role to an instance** via an **Instance Profile**

You want to always avoid embedding your AWS credentials when possible.

An **Instance Profile** holds a reference to a role. The EC2 instance is associated with the Instance Profile. When you select an IAM role when Launching an EC2 instance, AWS will automatically create the Instance Profile for you. Instance Profiles are not easily viewed via the AWS Console.

EC2

Placement Groups

amazon web services

EC2 – Placement Groups

Placement Groups let you to choose **the logical placement** of your instances to optimize for **communication**, **performance** or **durability**. Placement groups are **free**.

Cluster

- packs instances close together inside an **AZ**
- low-latency network performance for tightly-coupled node-to-node communication
- well suited for High Performance Computing (HPC) applications
- Clusters cannot be multi-AZ

Partition

- spreads instances across logical partitions
- each partition do not share the underlying hardware with each other (rack per partition)
- well suite for large distributed and replicated workloads (Hadoop, Cassandra, Kafka)

Spread

- Each instance is placed on a different rack
- When critical instances should be keep separate from each other
- You can spread a max of 7 instances. Spreads can be multi-AZ

EC2

Userdata

amazon web services

EC2 – UserData

You can provide an EC2 with **UserData** which is a **script** that will be automatically run when launching an EC2 instance. You could install package, apply updates or anything you like.

This example sets up an apache web-server

From within the EC2 instance, if you were to SSH in and CURL this special URL you can see the UserData script eg. **curl** **http://169.254.169.254/latest/user-data**

amazon web services

EC2

Metadata

amazon web services

EC2 – MetaData

From within your EC2 instance you can access information about the EC2 via a special url endpoint at

169.254.169.254

You would SSH into your EC2 instance and can use the CURL command:

curl http://169.254.169.254/latest/meta-data

/public-ipv4 get the current public IPV4 address
/ami-id the AMI ID used to launch this EC2 instance
/instance-type the Instance Type of this EC2 instance

Combine metadata with userdata scripts to perform all sorts of advanced AWS staging automation

```
[ec2-user ~]$ curl
http://169.254.169.254/latest/meta-data/
ami-id
ami-launch-index
ami-manifest-path
block-device-mapping/
events/
hostname
iam/
instance-action
instance-id
instance-type
local-hostname
local-ipv4
mac
metrics/
network/
placement/
profile
public-hostname
public-ipv4
public-keys/
reservation-id
security-groups
services/
```

EC2 Pricing Models

EC2 Pricing Introduction

amazon web services

EC2 – Pricing Model

On-Demand — Least Commitment

- low cost and flexible
- only pay per hour
- short-term, spiky, unpredictable workloads
- cannot be interrupted
- For first time apps

Spot upto 90% — Biggest Savings

- request spare computing capacity
- flexible start and end times
- Can handle interruptions (server randomly stopping and starting)
- For non-critical background jobs

Reserved upto 75% off — Best Long-term

- steady state or predictable usage
- commit to EC2 over a 1 or 3 year term
- Can resell unused reserved instances

Dedicated — Most Expensive

- Dedicated servers
- Can be on-demand or reserved (upto 70% off)
- When you need a guarantee of isolate hardware (enterprise requirements)

EC2 Pricing Models

On-Demand Pricing

amazon web services

EC2 - On-Demand Instances

Least Commitment

When you launch an EC2 instance it is by default using **On-Demand** Pricing

On-demand has **no up-front payment** and **no long-term commitment**

You are charged by the **hour** or by the **minute** (varies based on EC2 Instance Types)

On-Demand is for applications where the workload is for **short-term**, **spikey** or **unpredictable**.
When you have a **new app** for development or you want to run experiment.

amazon web services

EC2 Pricing Models

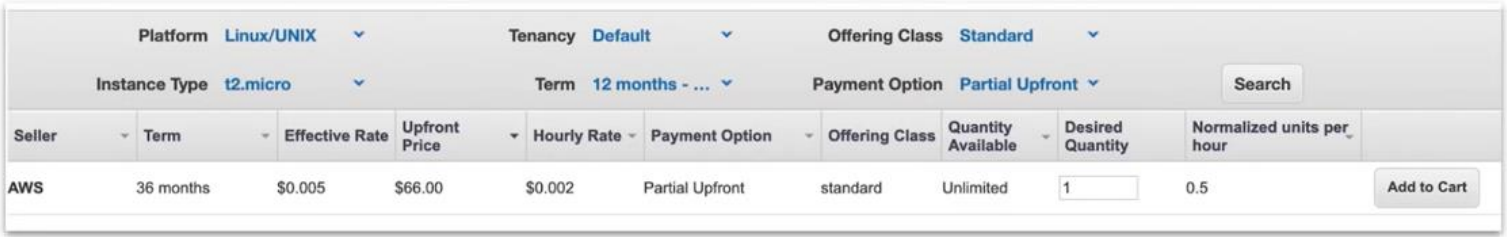
Reserved Instances (RI) Pricing

EC2 – Reserved Instances (RI)

Best Long-term

Designed for applications that have a **steady-state**, **predictable usage**, or require **reserved capacity**.

Reduced Pricing is based on **Term** x **Class Offering** x **Payment Option**



Standard Up to **75%** reduced pricing compared to on-demand.
Cannot change RI Attributes.

Convertible Up to **54%** reduced pricing compared to on-demand.
Allows you to change RI Attributes if greater or equal in value.

Scheduled You reserve instances for specific time periods eg. once a week for a few hours. Savings vary

Terms
You commit to a **1 Year** or **3 Year** contract.
The longer the term the greater savings.

Payment Options
All Upfront, **Partial Upfront**, and **No Upfront**
The greater upfront the great the savings

RIs can be **shared between multiple accounts** within an org

Unused RIs can be sold in the **Reserved Instance Marketplace**

amazon web services

EC2 Pricing Models

Spot Instances Pricing

amazon web services

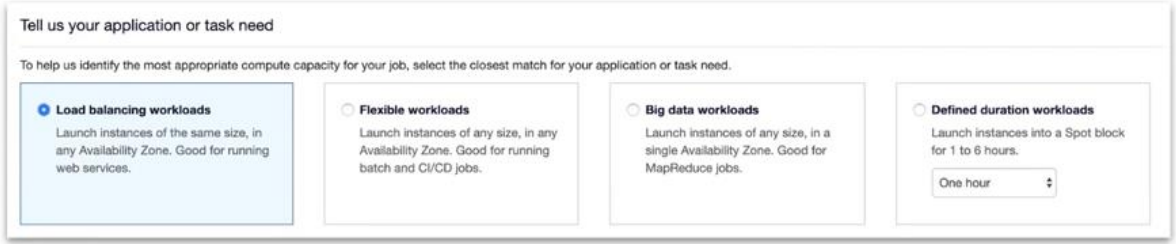
EC2 – Spot Instances

Biggest Savings

AWS has **unused compute capacity** that they want to maximize the utility of their idle servers.
It's like when a hotel offers discounts for to fill vacant suites or planes offer discount to fill vacant seats.

Spot Instances provide a discount of **90%** compared to On-Demand Pricing
Spot Instances can be terminated if the computing capacity is needed by on-demand customers.

Designed for applications that have flexible start and end times or applications that are only feasible at **very low** compute costs.



AWS Batch is an easy and convenient way to use Spot Pricing

Termination Conditions

Instances can be terminated by AWS **at anytime**
If your instance is **terminated by AWS**, **you don't get charged** for a partial hour of usage.
If **you terminate** an instance **you will still be charged** for any hour that it ran.

amazon web services

EC2 Pricing Models

Dedicated Host Instance Pricing

EC2 – Dedicated Host Instances

Most Expensive

Designed to meet regulatory requirements. When you have strict **server-bound licensing** that won't support multi-tenancy or cloud deployments.

Multi-Tenant vs Single Tenant

When multiple customers are running workloads on the same hardware. **Virtual Isolation** is what separate customers. (think apartment)

Multi-Tenant

When a single customer has dedicated hardware. **Physical Isolation** is what separates customers (think house)

Single-Tenant Single-Tenant Single-Tenant

Offered in both **On-demand** and **Reserved** (70% off on-demand pricing)

Enterprises and **Large Organizations** may have security concerns or obligations about against sharing the same hardware with other AWS Customers.

amazon web services

Amazon Machine Image

Amazon Machine Image Introduction

amazon web services

Amazon Machine Image (AMI)

A template to configure new instances

EC2 - AMI

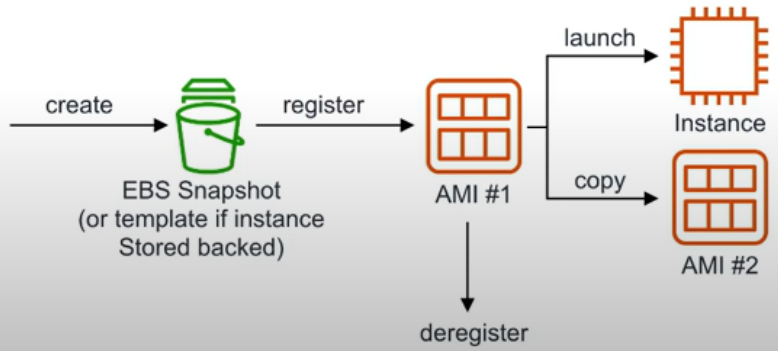
Amazon Machine Image **(AMI)** provides the information required to launch an instance.

You can **turn your EC2 instances into AMIs** so you can **create copies of your servers**

An AMI holds the following information:

- A template for the root volume for the instance (EBS Snapshot or Instance Store template) eg. an operating system, an application server, and applications
- Launch permissions that control which AWS accounts can use the AMI to launch instances.
- A block device mapping that specifies the volumes to attach to the instance when it's launched.

AMIs are **Region Specific!**



AMI – Use Cases

AMIs help you keep incremental changes to your OS, application code and system packages.

web-server-000
Ruby, Node, Postgres Client Installed

web-server-001
Redis for Sidekiq Installed

web-server-002
ImageMagick for Image Processing Installed

web-server-003
CloudWatch Agent Installed

Using **Systems Manager Automation** you can routinely patch your AMIs with security updates and bake those AMIS.

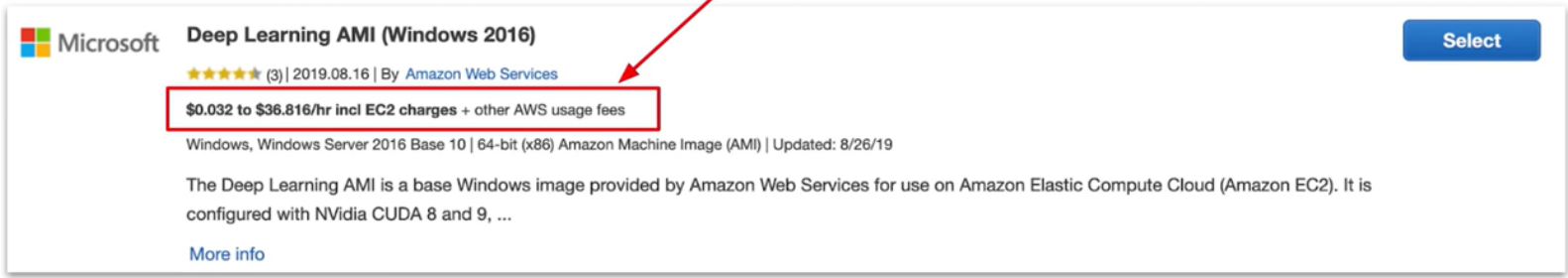
AMIs are used with **LaunchConfigurations**. When you want to roll out updates to multiple instances you make a copy of your LaunchConfiguration with new AMI

Amazon Machine Image

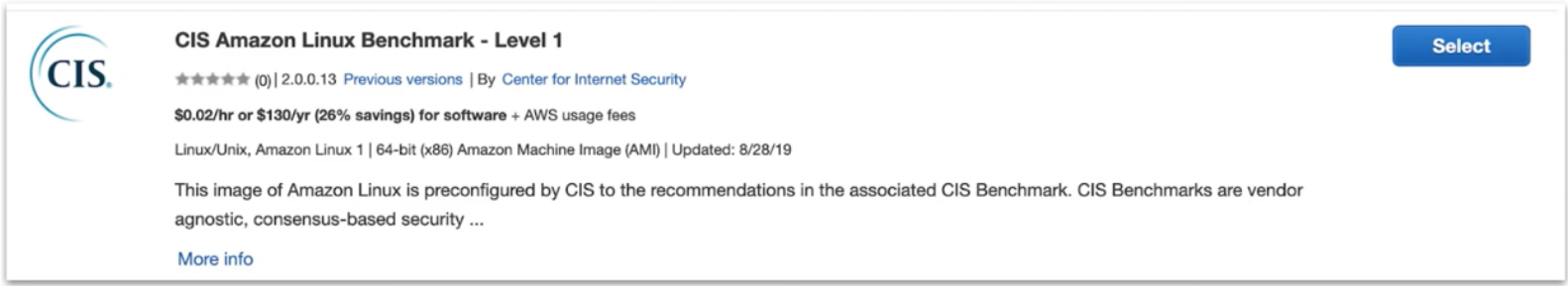
AMI Marketlpace

AWS Marketplace

The AWS Marketplace lets you **purchase subscriptions** to vendor maintained AMIs.



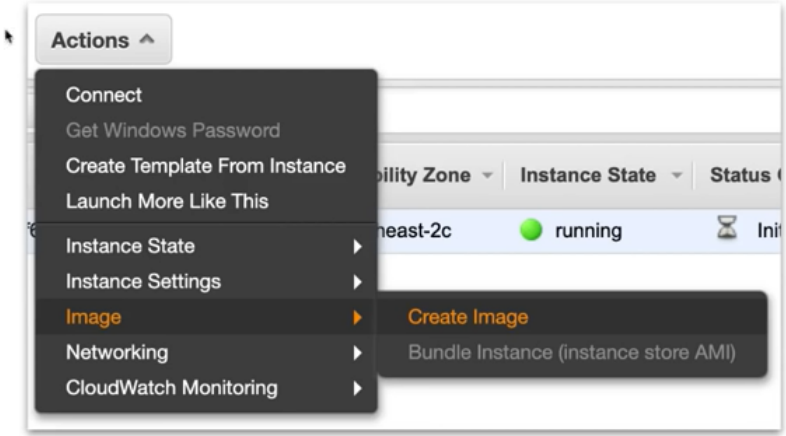
Security harden AMIs are very popular. eg. Center of Internet Security



amazon web services

AMI – Creating an AMI

You can **create an AMI** from an existing EC2 instance that's either **running** or **stopped**.



Choosing an AMI

AWS has hundreds of AMIs you can **search** and select from.

Community AMI are free AMIs maintained by the community
AWS Marketplace free or paid AMIs maintained by vendors

North Virginia

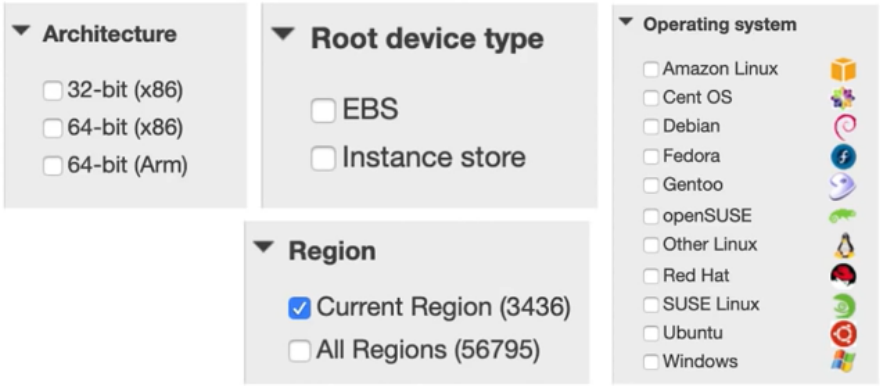
AMIs have an **AMI ID.** AMIs are **region specific**. Will have different AMI ID per region.

Canada Central

Choosing an AMI

Amazon Machine Images can be selected based on:

- Region
- Operating System
- Architecture (32-bit or 64-bit)
- Launch Permissions
- Root Device Volume
  - Instance Store (Ephemeral Storage)
  - EBS Backed Volumes



AMIs are categorized as either backed by Amazon EBS, or backed by Instance Store

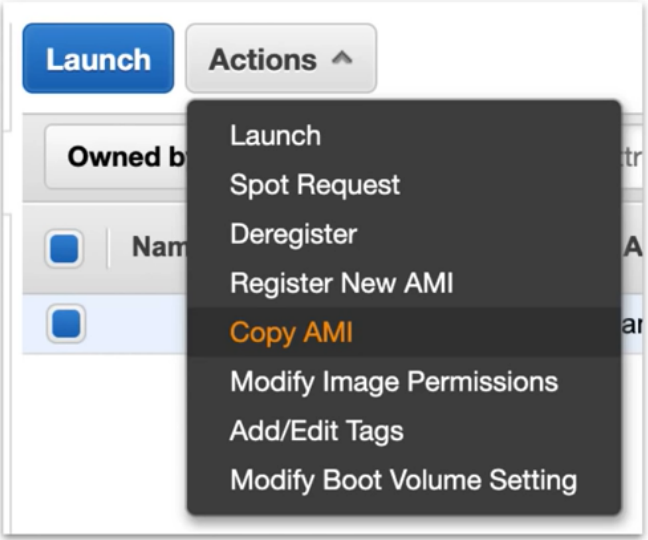
Amazon Machine Image

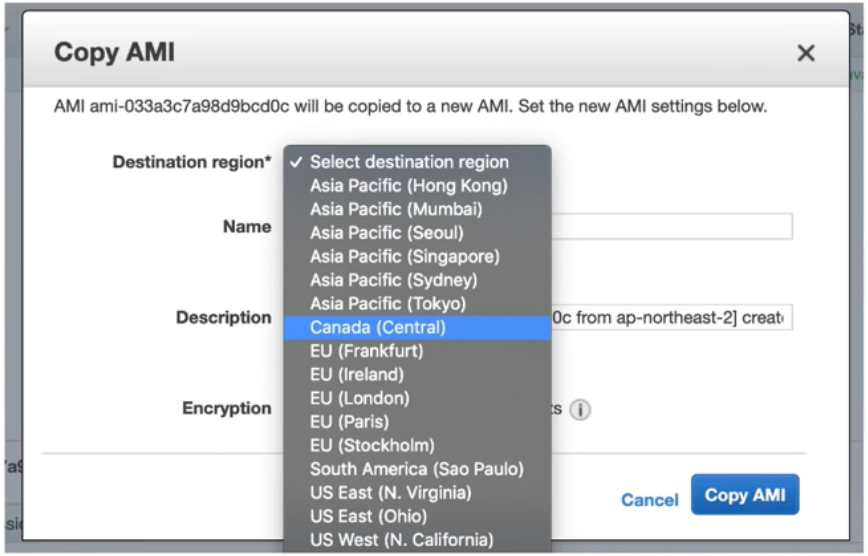
Copying an AMI

amazon web services

EC2 – Copying an AMI

AMIs are region specific. If you want to use an AMI from another region. You need to **Copy the AMI** and then select the destination region.





amazon web services

Auto Scaling Groups

Auto Scaling Groups Introduction

amazon web services

EC2 Auto Scaling Groups

Set scaling rules which will automatically launch additional EC2 instance or shutdown instances to meet current demand

Introduction to Auto Scaling Groups

Auto Scaling Groups (**ASG**) contains a collection of EC2 instances that are treated as a group for the purposes of automatic scaling and management.

Automatic scaling can occur via:

1. **Capacity Settings**
2. **Health Check Replacements**
3. **Scaling Policies.**

Auto Scaling Groups

Capacity Settings

amazon web services

ASG - Capacity Settings

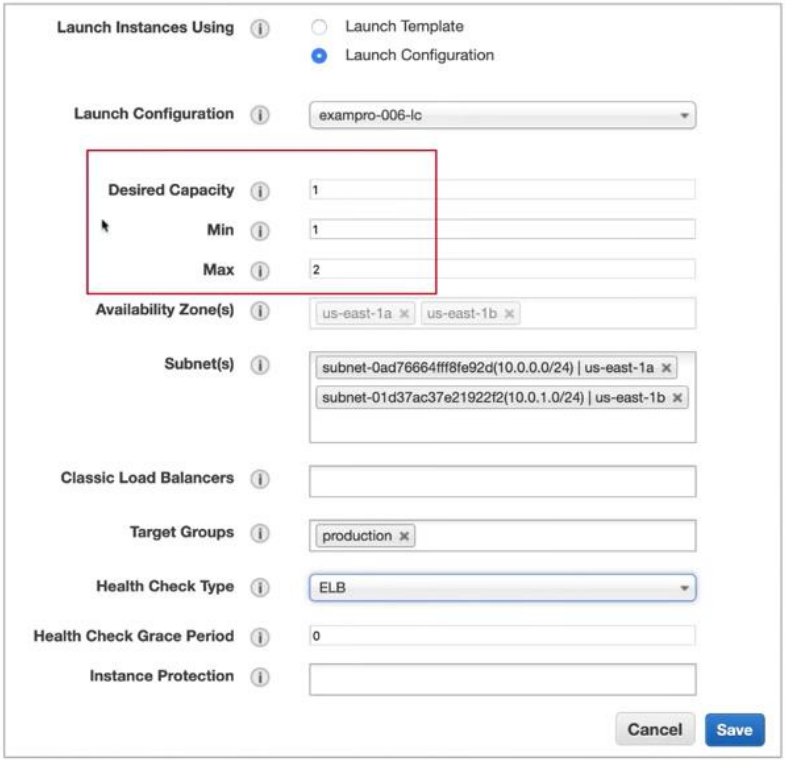
The size of an Auto Scaling Group is based on **Min**, **Max** and **Desired Capacity**.

Min is how many EC2 instances should at least be running.

Max is number EC2 instances allowed to be running.

Desired Capacity is how many EC2 instances you want to ideally run.

ASG will always launch instances to meet minimum capacity.



amazon web services

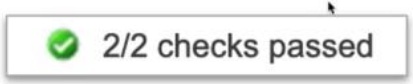
Auto Scaling Groups

Health Check Replacements

ASG – Health Check Replacements

EC2 Health Check Type

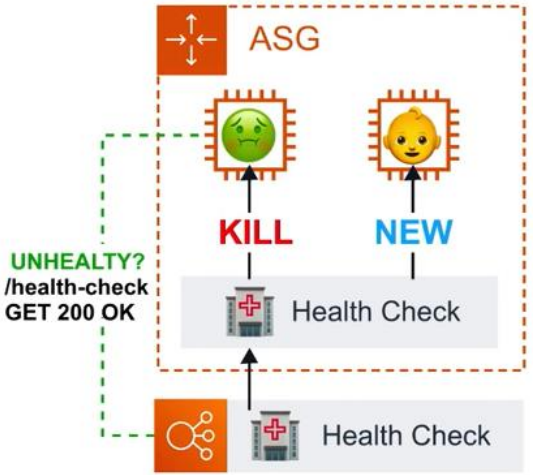
ASG will perform a health check on EC2 instances to determine if there is a software or hardware issue. This is based on the **EC2 Status Checks**. If an instance is considered unhealthy. ASG will terminate and launch a new instance.



ASG – Health Check Replacements

ELB Health Check Type

ASG will perform a health check based on the ELB health check. ELB can perform health checks by pinging an HTTP(S) endpoint with an expected response. If ELB determines a instance is unhealthy it forwards this information to ASG which will terminate the unhealthy instance.



Auto Scaling Groups

Scaling Policies

amazon web services

ASG – Scaling Policies

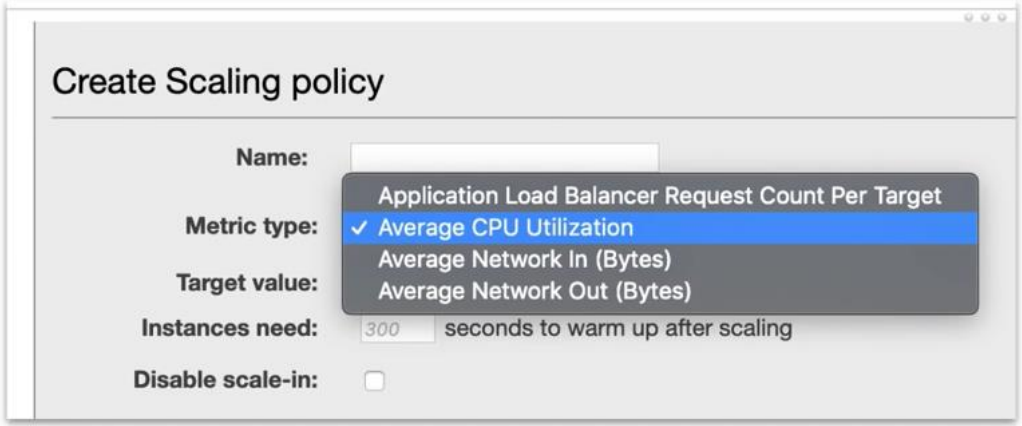
Scaling Out: Adding More Instances

Scaling In: Removing Instances

Target Tracking Scaling Policy

Maintains a specific metric at a target value.

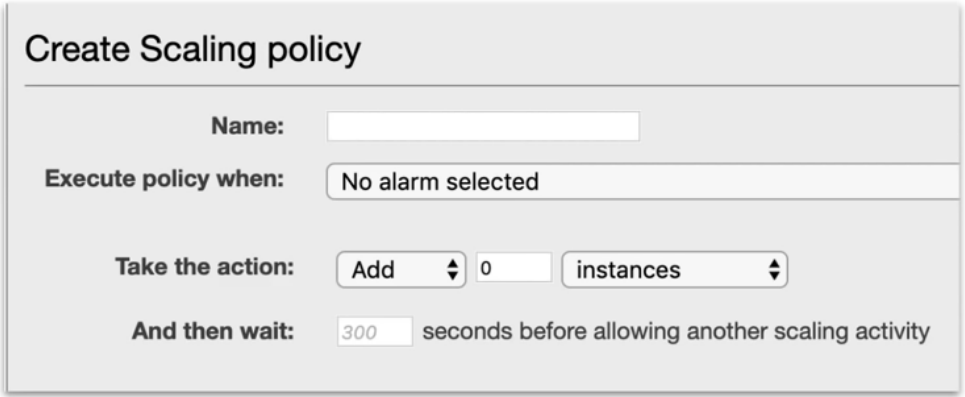
eg. If **Average CPU Utilization** exceeds 75% then add another server.



ASG – Scaling Policies

Simple Scaling Policy

Scales when an **alarm is breached**.

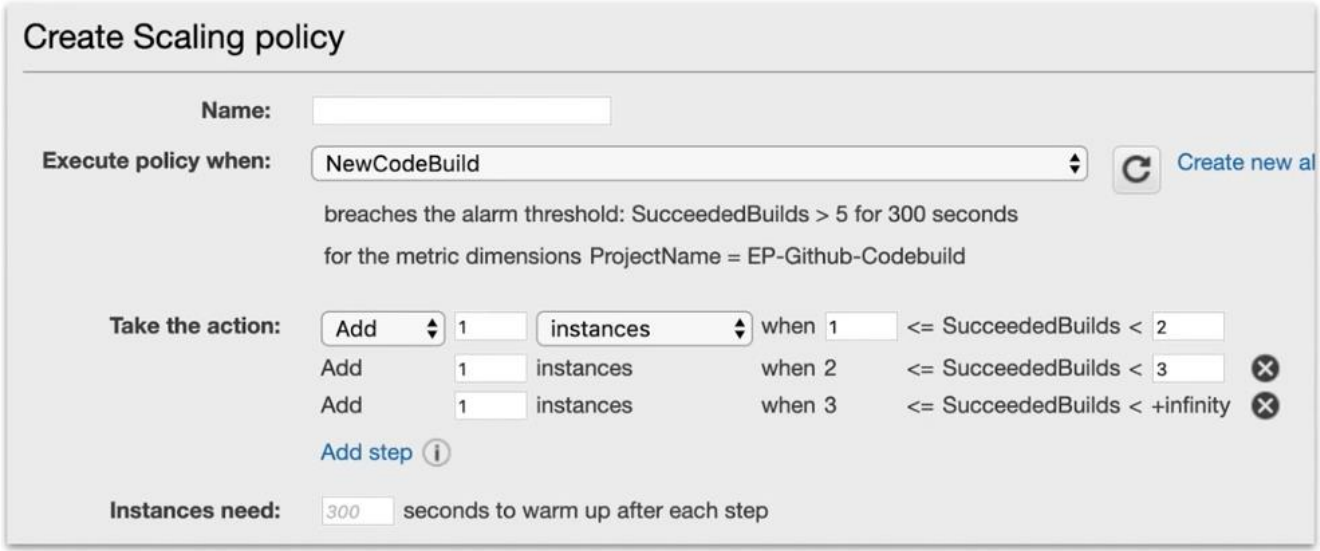


Not recommended, legacy scaling policy. Use scaling policies with steps now.

ASG – Scaling Policies

Scaling policies with steps

Scales when an **alarm is breached**, can **escalates based on alarm** value changing.



Auto Scaling Groups

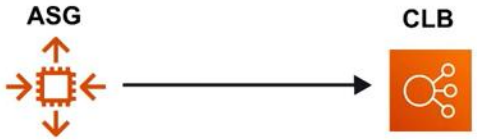
ELB Integration

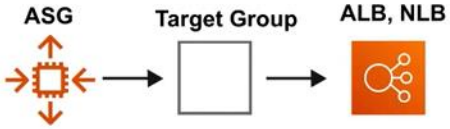
amazon web services

ASG – ELB Integration

ASG can be associated with Elastic Load Balancers (ELB). When ASG is associated with ELB richer health checks can be set.

Classic Load Balancers are associated **directly** to the ASG

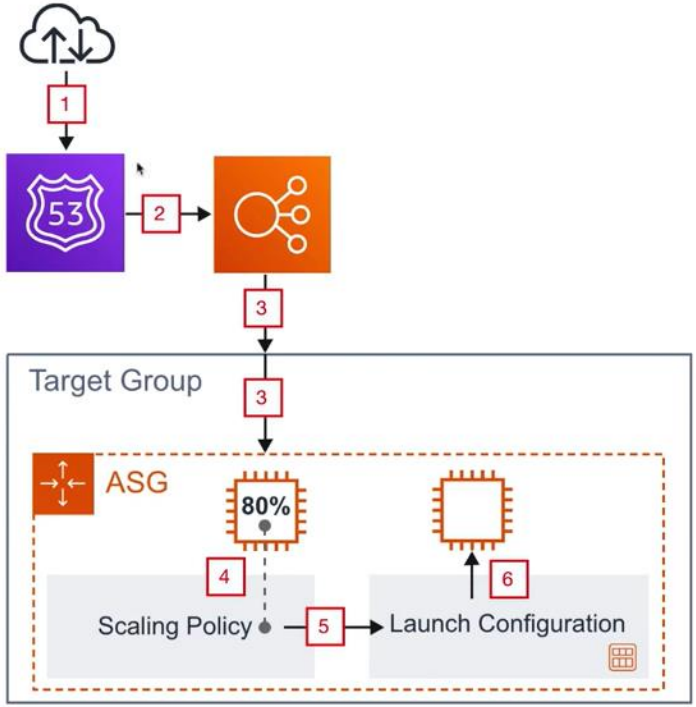




Application and Network Load Balancers are associated **indirectly** via their Target Groups.

ASG – Use Case

1. Burst of traffic from the internet hits our domain.
2. Route53 points that traffic to our load balancer.
3. Our load balancer passes the traffic to its target group.
4. The target group is associated with our ASG and sends the traffic to instances registered with our ASG
5. The ASG Scaling Policy will check if our instances are near capacity.
6. The Scaling Policy determines we need another instance, and it Launches an new EC2 instance with the associated Launch Configuration to our ASG



Auto Scaling Groups

Launch Configuration

amazon web services

Launch Configuration

A launch configuration is an instance configuration template that an Auto Scaling group uses to launch EC2 instances.

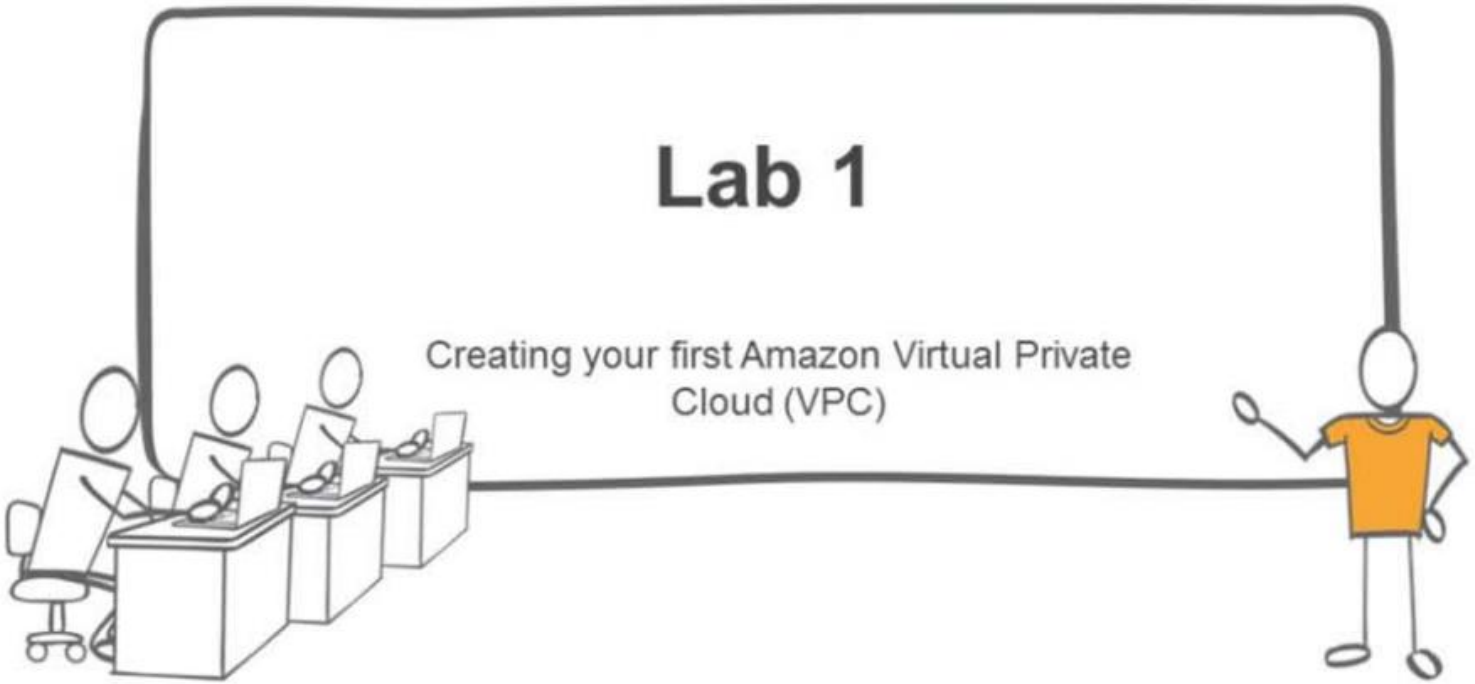
A Launch Configuration is the same process as Launching an EC2 instance except you are saving that configuration to Launch an Instance for later. Hence “Launch Configuration.

Launch Configurations **cannot be edited**, When you need to update your Launch Configuration you create a new one or clone the existing configuration and then manually associate that new Launch Configuration

Launch Templates are Launch Configurations with Versioning, Everyone appears to still use Launch Configurations

Lab 1

Creating your first Amazon Virtual Private Cloud (VPC)



amazon web services