# AWS CERTIFIED SOLUTION ARCHITECT ASSOCIATE

Study Guide V1.0

**Abdul Jaseem. VP**

Technical Consulting Engineer at Cisco

CCIE Collab 59174, CCNP Collab, CCNA,

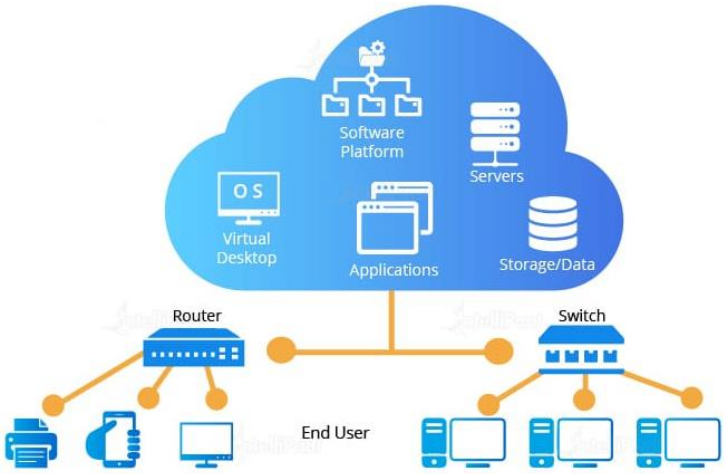
DevNet, vmware DCV, AWS SAA, CKA

30/August/2020

LinkedIn; YouTube

# Contents

## Introduction to Virtualization and Cloud Computing




- **What is Internet**: A global computer network using standardized communication protocols (e.g. UDP, TCP/IP) providing information and communication facilities.
- **Local Network**: Private network in LAN (Local Area Network)
- **Virtualization**: Run multiple OSs on a host machine (Type 1: BareMetal, Uses Hypervisor OS (e.g. ESXi) and Type 2: Application running on another base OS (e.g. vmware workstation)).
- **Virtual Machine**: Software representation of virtual computer as set of files! Easy to move, independent of hardware, Effective utilization of resources. We can do virtual networking between VMs.
- **Data Center**: Data centers are simply centralized locations where computing and networking equipment is concentrated for the purpose of collecting, storing, processing, distributing or allowing access to large amounts of data.
- **What is Cloud**? There is now Cloud, it is someone else computer accessible over Internet! Virtual Machine running on a cloud server is the most widely used way of hosting any applications online.
- **Cloud Computing**: On demand delivery of IT resources via the Interment, Instead of setting up Physical DCs, we can access Compute, Network, Storage on demand basis from Cloud providers. Use cases: Data backup, DRS, Email, Virtual Desktop, Software Development, Testing, Web Apps, Online Gaming, IoT, etc. Worldwide availability.

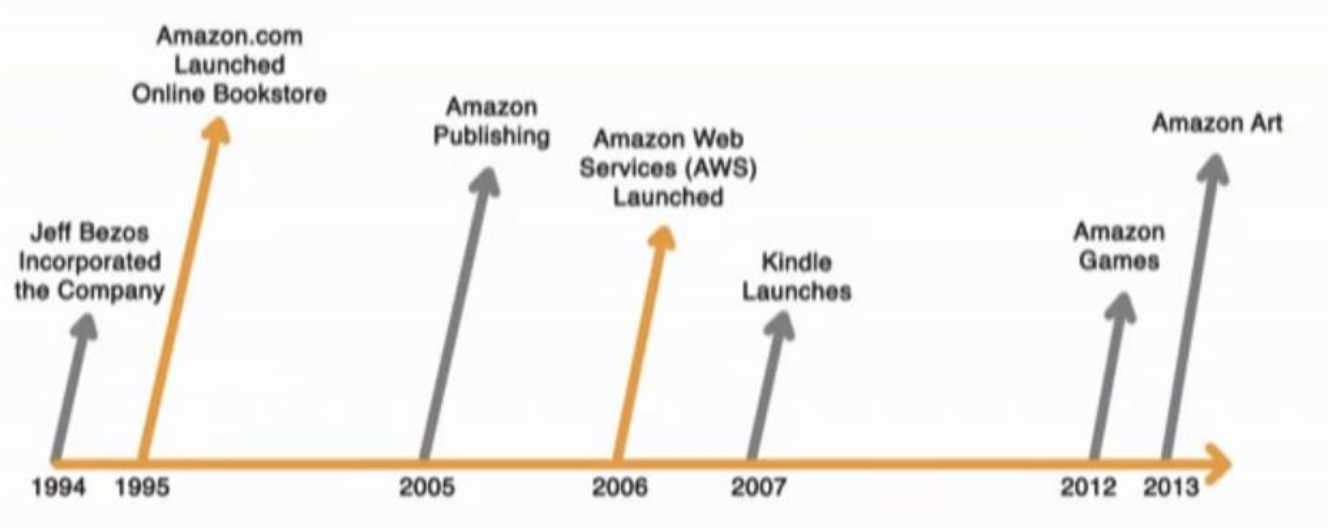
AWS Reference: [What is Cloud Computing?]

## What is Amazon Web Services (AWS)?


- Applications like YouTube, Gmail, Facebook are software that are hosted on cloud. Those are often called Software as a Services (SaaS).
- Amazon Web Services is a public cloud provider, a gigantic pool of configurable resources (Servers, Storage, Networks, etc.) spanned across multiple Data Centers in the globe where you can deploy your infrastructure/ applications. Hence it is also called Infrastructure as a Services (IaaS).
- IaaS – Entire Infrastructure provided to you as a fully managed service
- Rapid provisioning and release of the resources.
- Resources are elastic in nature; you can scale up and scale down the resources. (Scaling: Ability to change an implementation to respond to changing traffic patterns.). e.g. During Black Friday sale, services may require more compute power.
- You can build could apps/ software (SaaS) using AWS (IaaS). Services can be used On-demand and Pay as you Go fashion (Like our Electricity bill, pay for the usage).
- We can gradually move services to Cloud (On-Ramp), some of the services are moved to cloud, often called Hybrid Cloud.
- AWS follows strict regulations that allows Banking, Research sectors to move their infrastructure to cloud.
- The collective money that we pay to Amazon, enables them to maintain the Data Centers all over the world.
- Examples AWS Solutions: amazon.com, Netflix, etc. uses AWS in the backend.

AWS Reference: *What is AWS?*

## History of AWS




- 1994 – Jeff Bezos Started the company
- 1995 – amazon.com launched as an online bookstore
- 2006 – AWS Launched (Elastic Compute Cloud - EC2, S3)
- 2009 – Relational Database Services – RDS
- 2011 – Elastic Beanstalk
- 2012 – Glacier, DynamoDB, RedShift
- 2016 – Lambda

Over the years AWS has been diligent about expanding their offerings, making sure the features and products available inside the cloud that allow us to take enormous advantage of the resources available to us.

## AWS Global Infrastructure

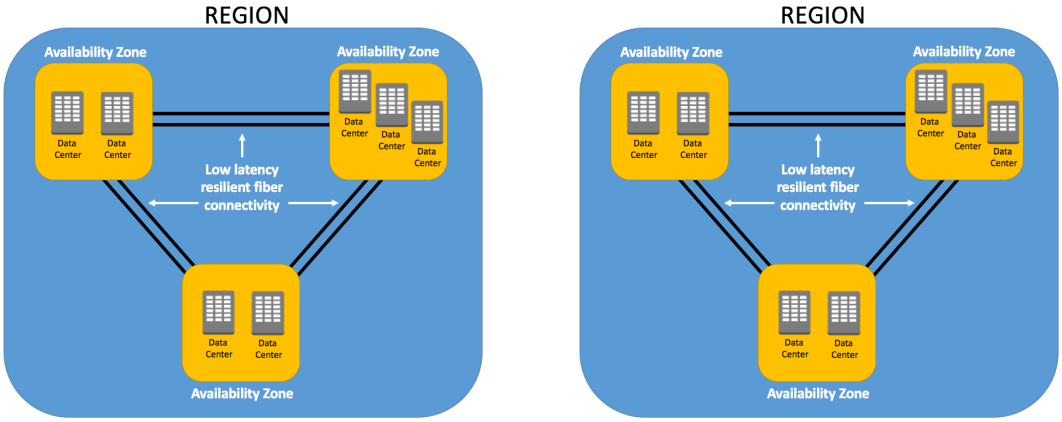
AWS infrastructure is geographical deployed in different Data Centers.

### 1. Regions

- Any geographic location where AWS has a Cluster of Data Center is called Region.
- AWS infrastructure Regions meet the highest levels of security, compliance, and data protection.
- 24 Launched and 3 announced regions (at the time of this article)

### 2. Availability Zones (AZ)

- Each group data centers called an Availability Zone (AZ). Each AZ has independent power, cooling, and physical security and is connected via redundant, ultra-low-latency networks.
- Inside a Region we can find multiple Availability Zone (1a, 1b, 1c), each Availability Zone comprised of multiple Data Centers.
- Inside an Availability Zone, our data/ infrastructure is split-up among multiple DCs. This gives built in reliability and redundancy.
- Also enables to push services close to your customer to reduce the latency.




- AZ's are physically separated by a meaningful distance, many kilometers, from any other AZ, although all are within 100 km (60 miles) of each other.
- Amazon recommends deploying apps/ solutions in at least 2 AZs
- 77 Availability Zones (at the time of this article)

### 3. AWS Edge Locations:

- Used to cache your data in order to get faster response time for customers. Like Content Delivery Network (CDN). e.g. Video Streaming, Netflix, Amazon Prime.
- CloudFront CDN available here
- 205 Edge Locations (at the time of this article)

## AWS Foundation Services

### 1. Compute

- EC2, Lambda, Elastic Beanstalk

### 2. Networking

- Load-balancing, Route53, VPC, Direct Connect

### 3. Storage

- S3, Block Storage, Glacier

## AWS Platform Services

More like Software as a Service (SaaS). The AWS Global Cloud Infrastructure is the most secure, extensive, and reliable cloud platform, offering over **175 fully featured services** from data centers globally.

### 1. Databases (Example. Pre-installed SQL)

- DynamoDB, RDS, Redshift

### 2. Analytics

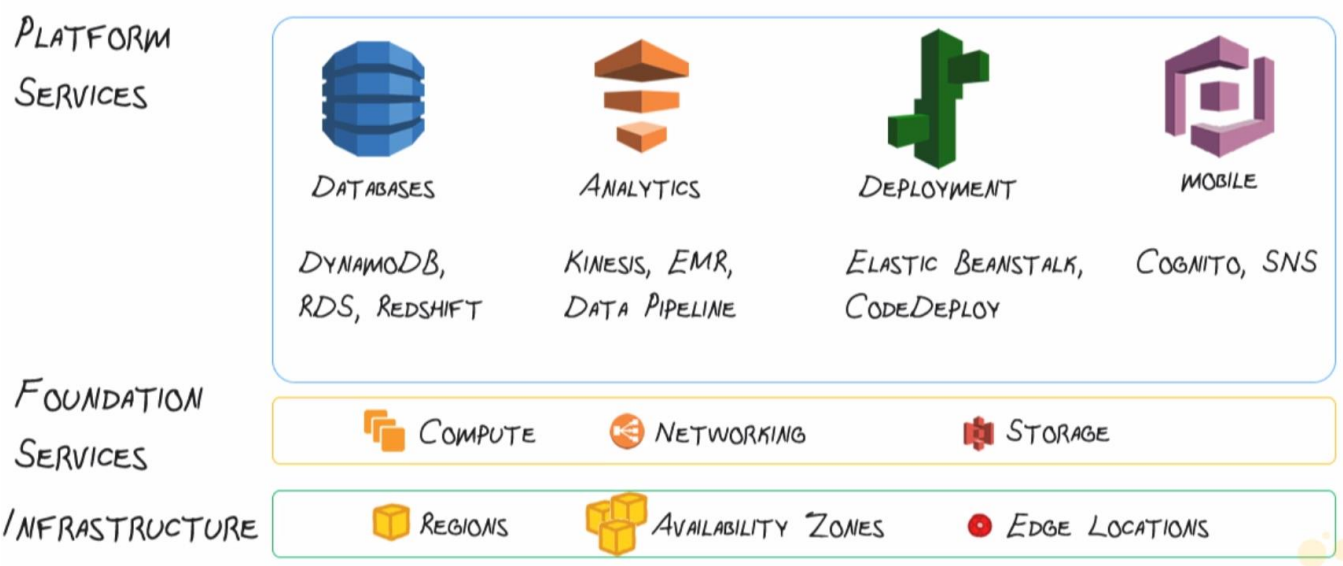
- Kinesis, EMR, Data Pipeline

### 3. Deployment

- Elastic Beanstalk (Platform as a Services - PaaS), Code Deploy

### 4. Mobile Services

- Cognito, SNS




AWS Reference: [Regions and Availability Zones]

# AWS Management Console

Three ways to connect AWS.

## 1. Command Line Interface (CLI)

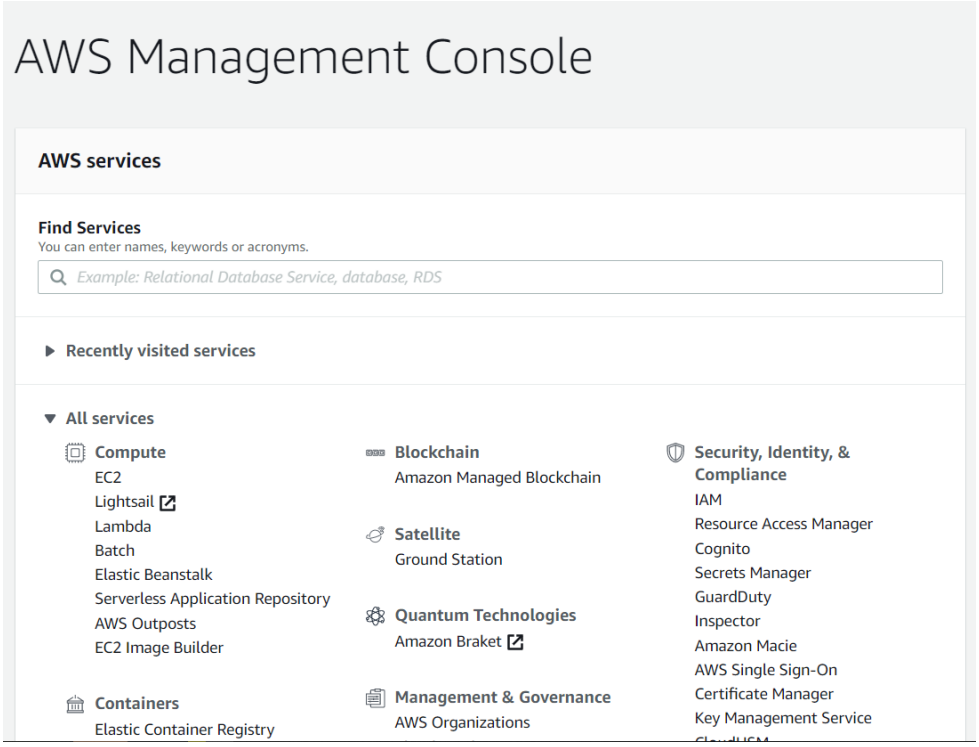
- Utility that you can download from AWS website.
- Integrate with your Shell (cmd, bash, etc.) and write AWS commands.

## 2. RESTFUL API

- HTTP Calls to AWS.
- Makes easier for automation.

## 3. Web Based Console

- Easiest way to connect AWS via Web browser, even AWS has mobile version!

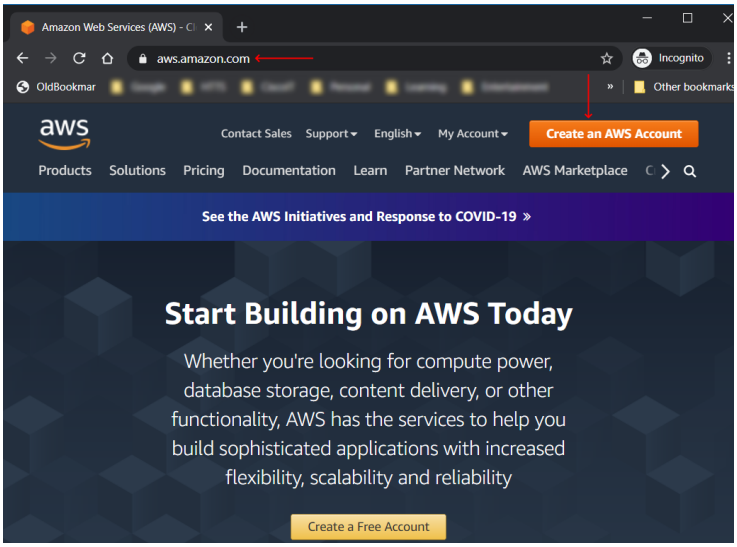


# Create an AWS Account

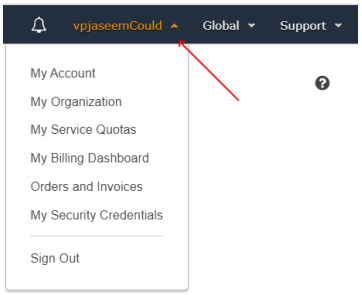
AWS Accounts Include 12 Months of Free Tier Access, including use of Amazon EC2, Amazon S3, and Amazon DynamoDB. Visit aws.amazon.com/free for more details. You need a Credit / Debit card to sign up. AWS will charge you RS.2/- and refund it back to the same card.

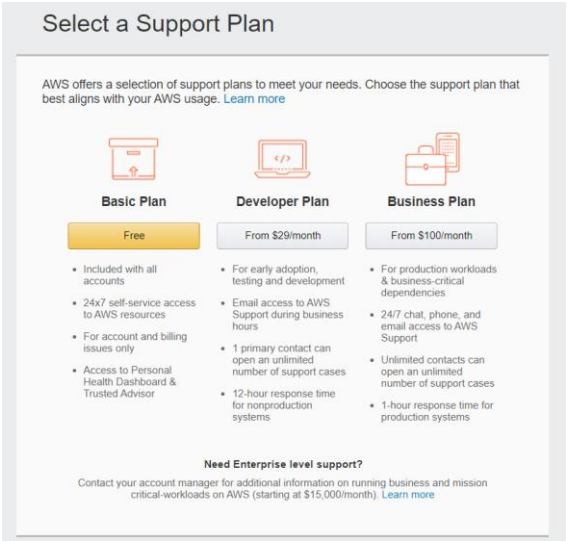
**Create an AWS Account**

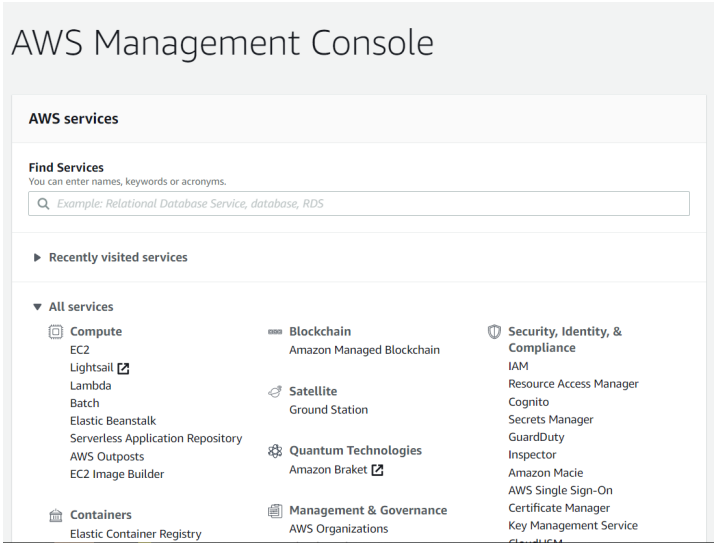
- Go to aws.amazon.com >> Create an AWS Account



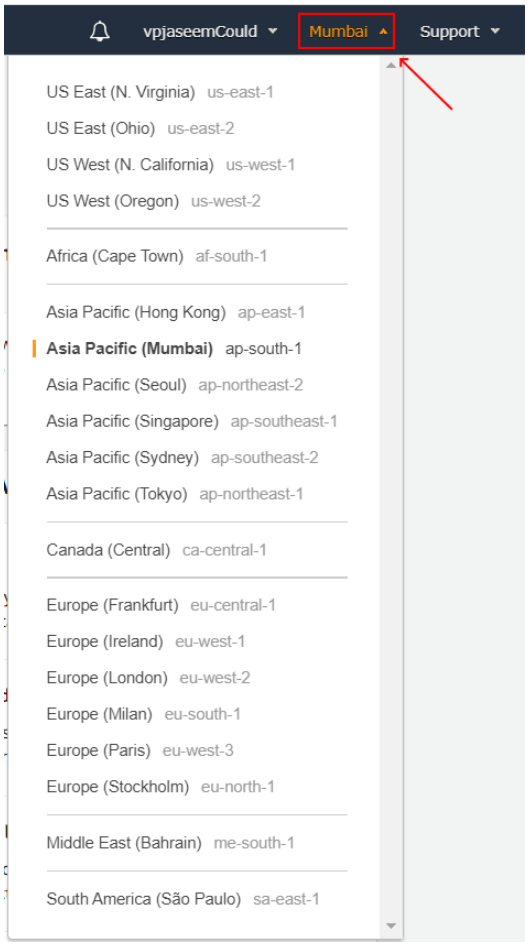

- Follow the Signup process. You can choose any friendly name as **AWS account name**. That will be displayed once you login (My AWS account name is 'vpjaseemCloud', this can be edited anytime)



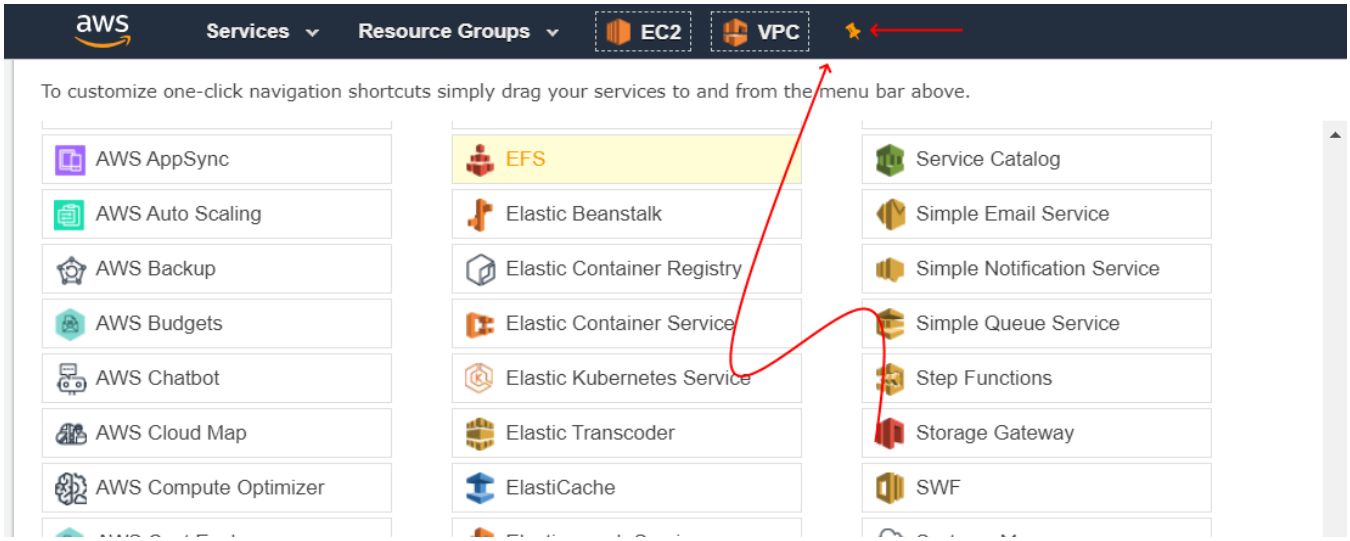



- Once you click the Region Name, you can see all available regions as shown below.




- You can pin some essential services as quick link in the top bar



# Amazon EC2 Instance

## What is Amazon Elastic Compute Cloud EC2?

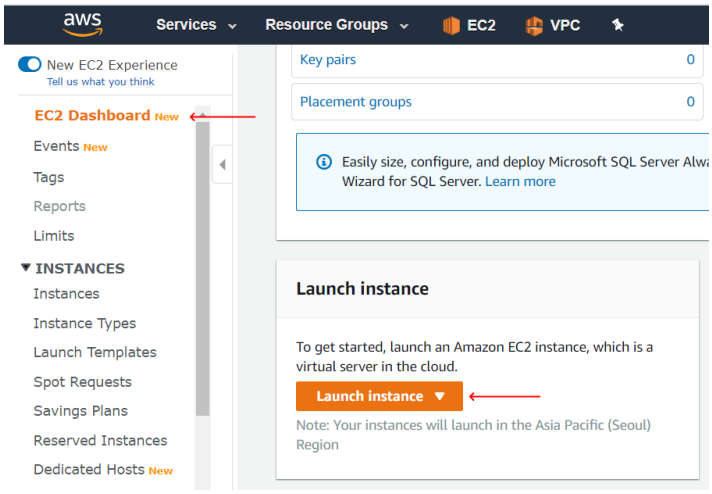
- Amazon Elastic Compute Cloud (Amazon EC2) is an IaaS offering that provides Custom Virtual Machines in the cloud. It's a fully managed infrastructure.
- Fast provisioning and Supports Elastic Scaling – Dynamically add instances to meet specific traffic requirements.
- The scope of EC2 is AZ
- **Amazon Machine Images (AMIs)**: Preconfigured Operating System templates with and additional software installed. We can build our own custom images as well.
- **Instance Types:** Various configurations of CPU, memory, storage, and networking capacity for your instances.
- Secure login information for your instances using key pairs (AWS stores the public key, and you store the private key in a secure place)
- Metadata, known as tags, that you can create and assign to your Amazon EC2 resources. This is helpful to identify the instances easily.
- Flexible payment options: Pay as You Go (Pay only for the machine in a running state, stop machine to save cost. Hourly pricing (partial hour count as full)), Reserved, Scheduled

AWS Reference: What is Amazon EC2?

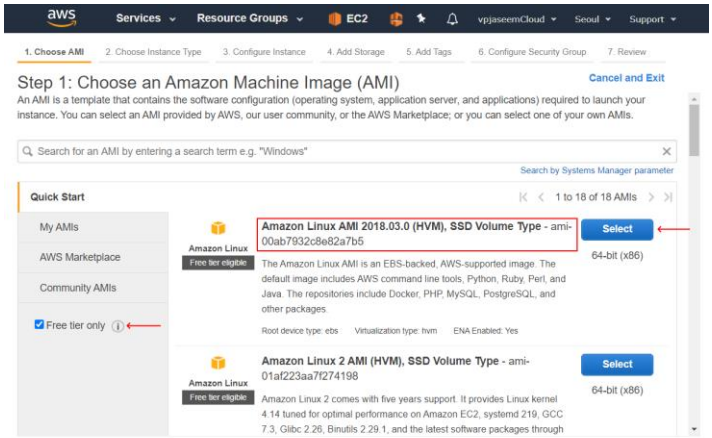
## How to Deploy Amazon Linux AMI EC2?

Now let's see how to deploy a 'Amazon Linux AMI' EC2 instance. There are 4 things to consider while deploying an EC2 instance.

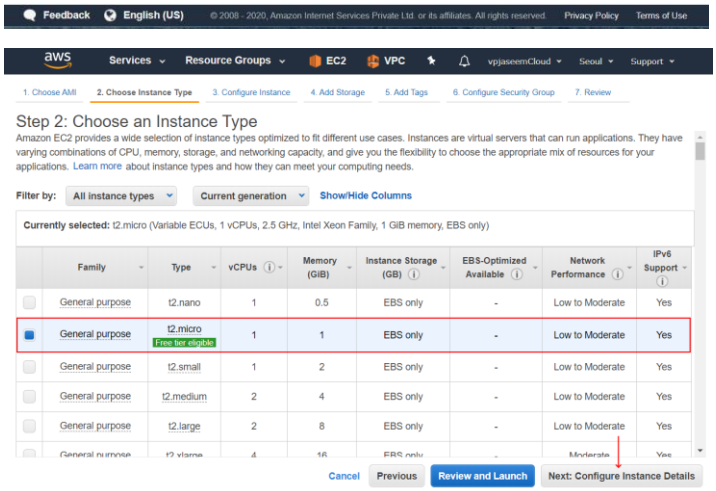
1. Region and Availability Zone
2. Choose an Image (AMI) to launch from
3. Machine Type (CPU, HDD, RAM, etc.)
4. Configure Network, Storage, Security Groups




Go to EC2 Dashboard >> Launch instance >> Launch instance >>

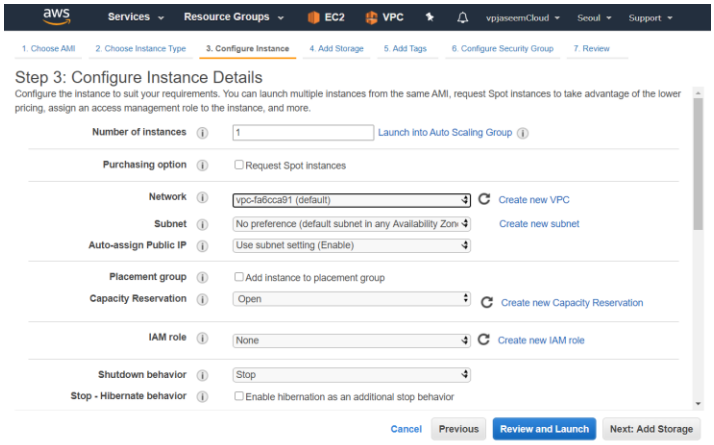



Free tier only >> Select 'Amazon Linux AMI' >>



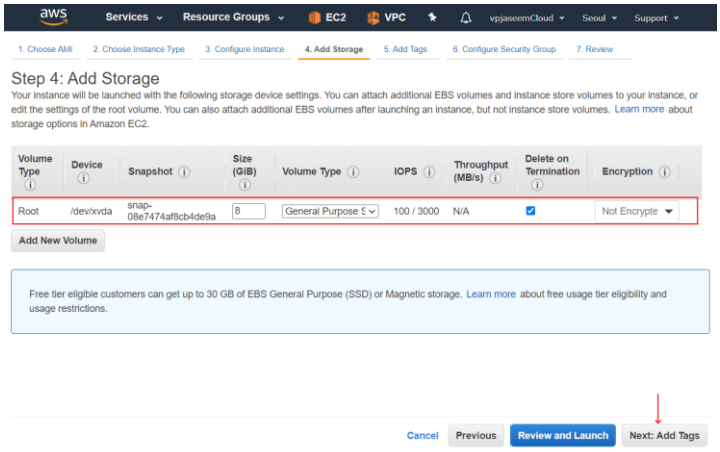

General purpose t2.micro >> Next: Configure Instance Details

AWS Reference: [EC2 Instance Types]



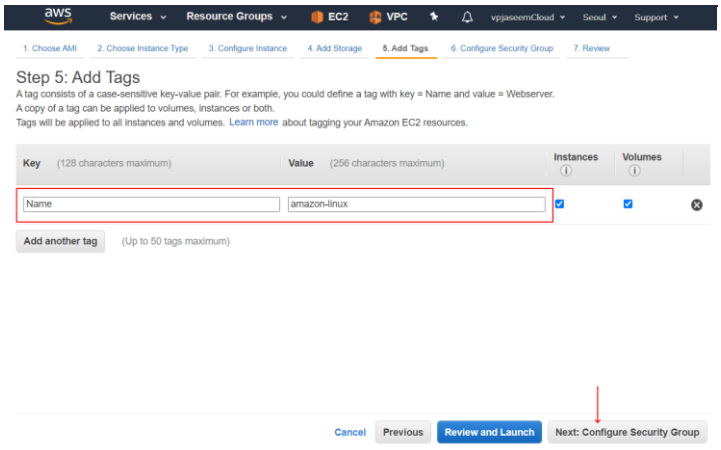

- Configure instance details to suit your requirements.
- Select all default options for now, we will cover every component in future sessions.

>> Next: Add Storage



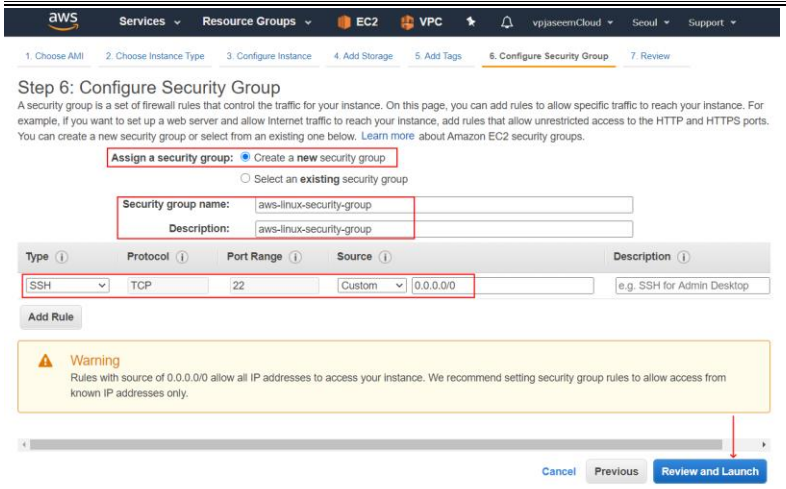

- An instance store provides temporary block-level storage for your instance. This storage is located on disks that are physically attached to the host computer.
- You can attach additional EBS (Elastic Block Storage) volumes and instance store volumes to your instance or edit the settings of the root volume.
- You can also attach additional EBS volumes after launching an instance, but not instance store volumes.

>> Next: Add Tags

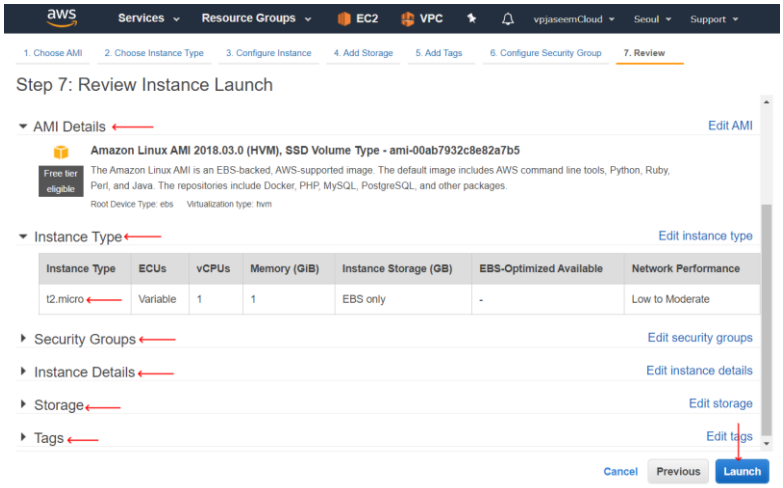



- Tags used to identify the EC2 instances as quickly as possible

>> Next: Configure Security Group



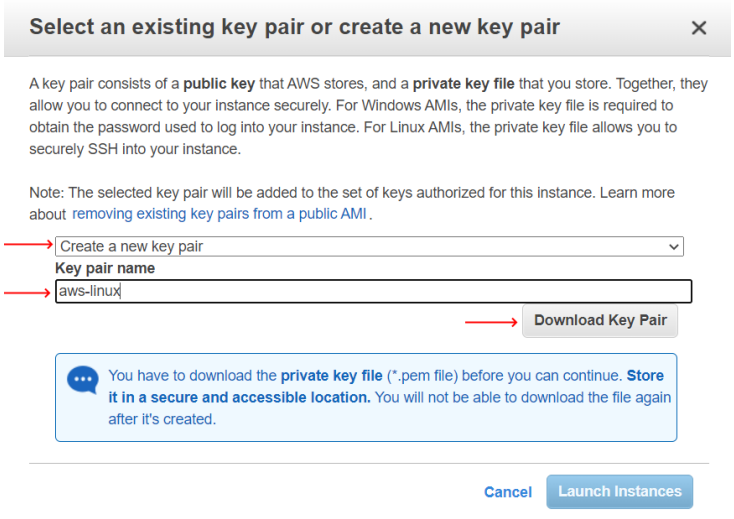

- Select Create a **new security group** >> provide a **Security group name**
- **Security Group**: A security group is a set of firewall rules that control the traffic for your instance
- You can add rules to allow specific traffic to reach your instance, Make sure you have SSH (22) allowed from all IP Address >> **Review and Launch**




- We can review all the settings here. Quickly verify those.

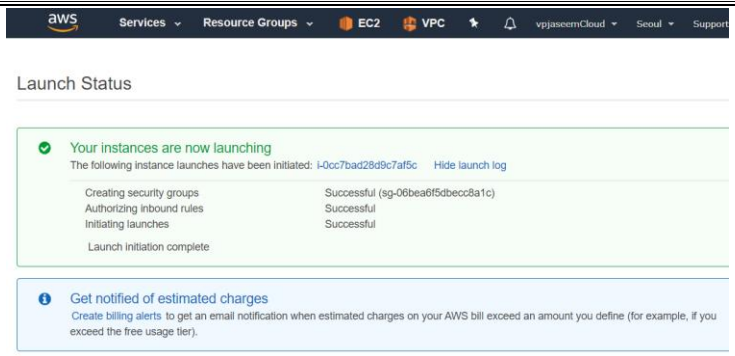
>> **Launch**

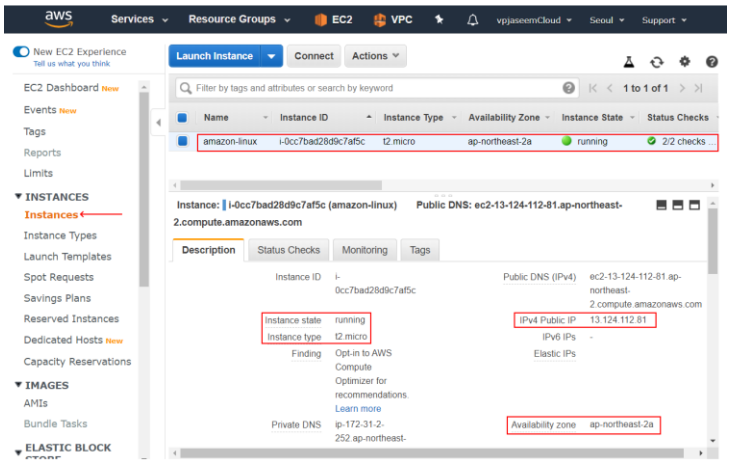
- A key pair consists of a public key that AWS stores, and a private key file that you store.
- Together, they allow you to connect to your instance securely.
- For Windows AMIs, the private key file is required to obtain the password used to log into your instance.
- For Linux AMIs, the private key file allows you to securely SSH into your instance.




- Select 'Create a new key pair' and enter a name for that.
- You can download the 'Private Key'. Keep that in a safe place

>> **Launch Instances**

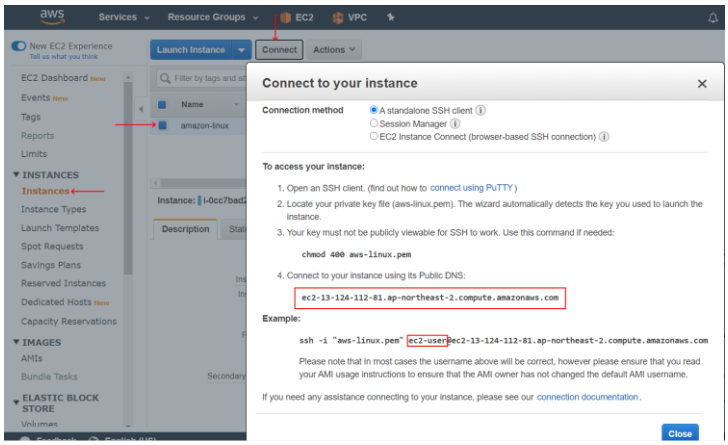


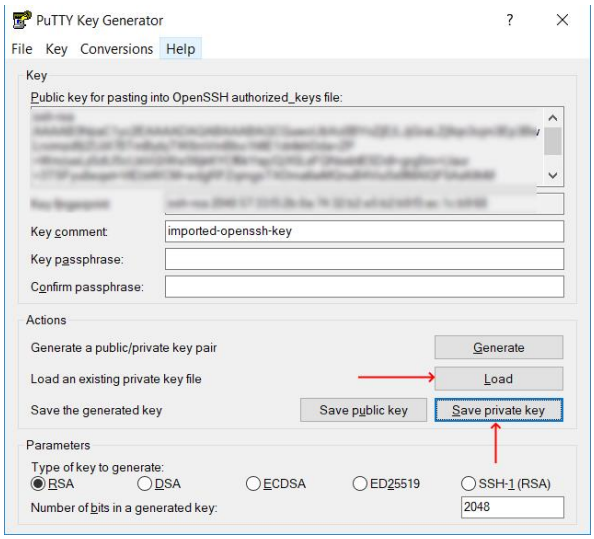

- Once the launching is completed, you can click the 'View Instances' button on the same page to see the newly created EC2 instance
- You could see all the details (Status, DNS, Public IP, Type, etc.) in this page

## How to Access Amazon Linux AMI EC2 Instance? Standalone SSH Client

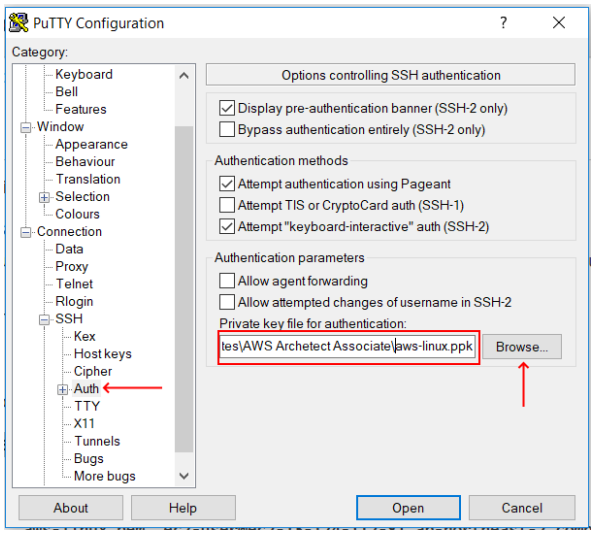
- Linux machines build from AMI can be accessed from anywhere over internet. You should need the .PEM file that downloaded after the creation of an EC2 instance.
- Windows uses Putty Gen to convert the .PEM file to .PPK file, and this file used to connect Linux over SSH. We do not use any password!



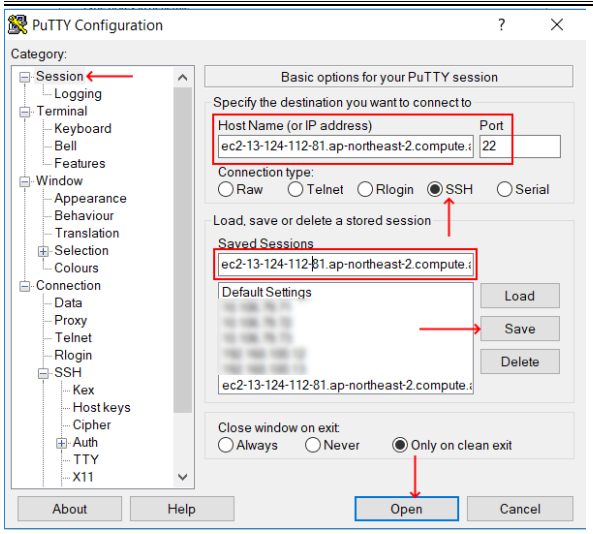

- Select the EC2 instance >> Connect >> Note the Public DNS (You can also use the Public IP of the EC2)
- Also note the username here is ec2-user



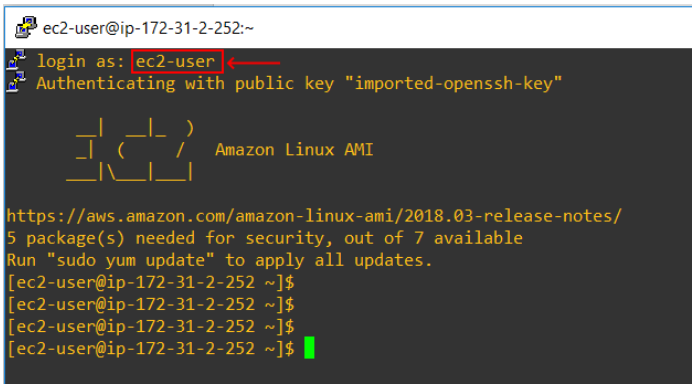

- Open PuttyGen and Load the .PEM file downloaded before. Click Save Private Key
- Keep the private key in a safe place




- Open Putty >> Auth >> Browse the Private Key (.PPK file created in previous step)



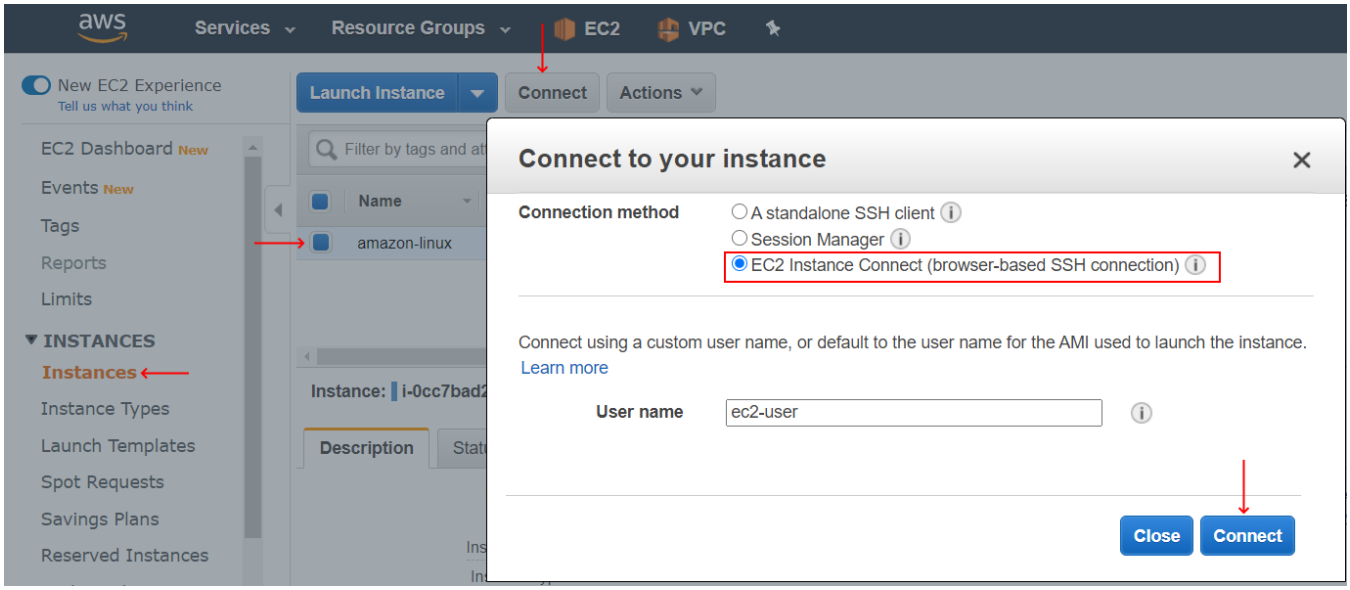

- Session >> Enter the FQDN or Public IP of the EC2 instance >> Select SSH >> Optionally you can Save the session >> Click Open




That's it, you are successfully connected!

## How to Access Amazon Linux AMI EC2 Instance? Browser-based SSH connection

- Select the EC2 instance >> Connect >> EC2 Instance Connect (browser-based SSH connection) >> Connect

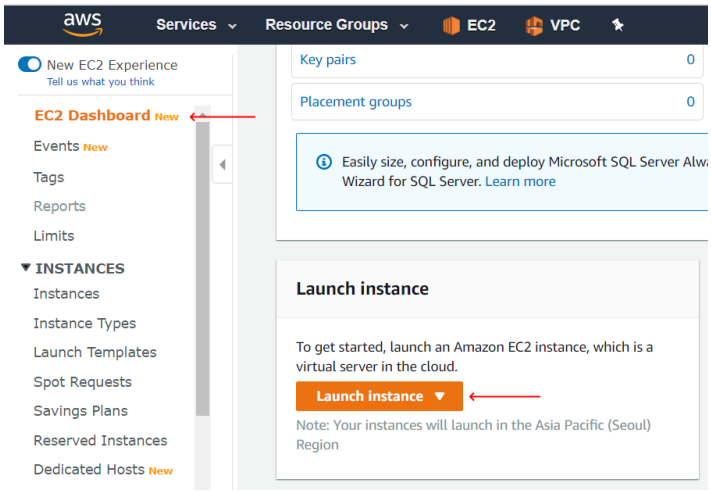



- This will open Linux CLI on new browser window. Please note that, some time, it fail

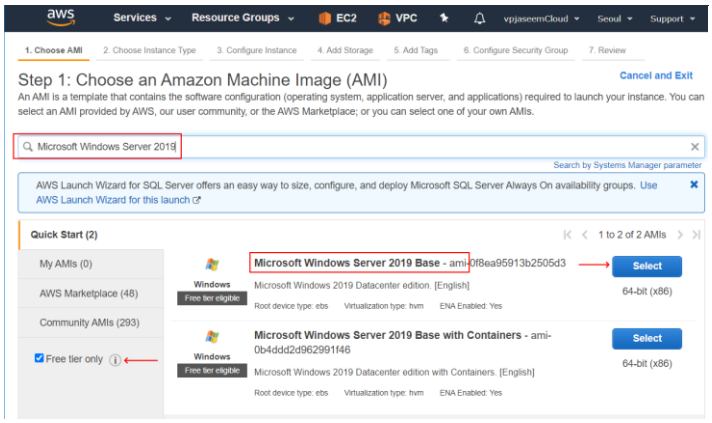
## How to Deploy Microsoft Windows Server 2019 Base AMI EC2?

Now let's see how to deploy a 'Microsoft Windows Server 2019 AMI' EC2 instance. There are 4 things to consider while deploying an EC2 instance.

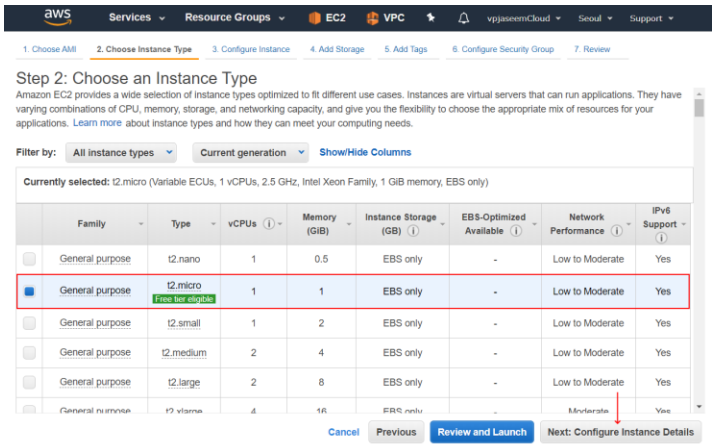
1. Region and Availability Zone
2. Choose an Image (AMI) to launch from
3. Machine Type (CPU, HDD, RAM, etc.)
4. Configure Network, Storage, Security Groups




Go to EC2 Dashboard >> Launch instance >> Launch instance >>

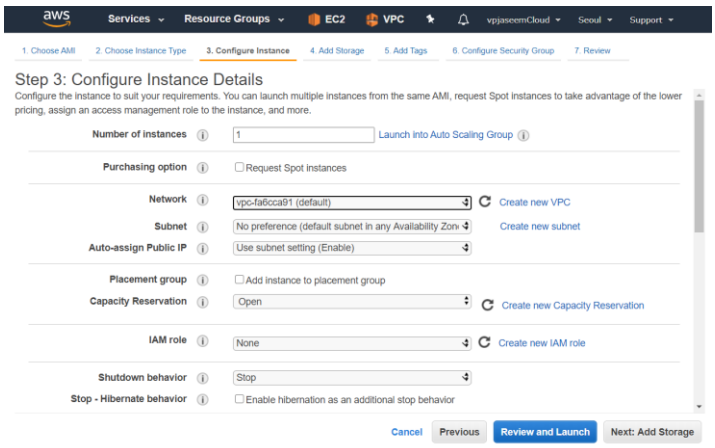



Free tier only >> Select 'Microsoft Windows Server 2019 AMI' >>



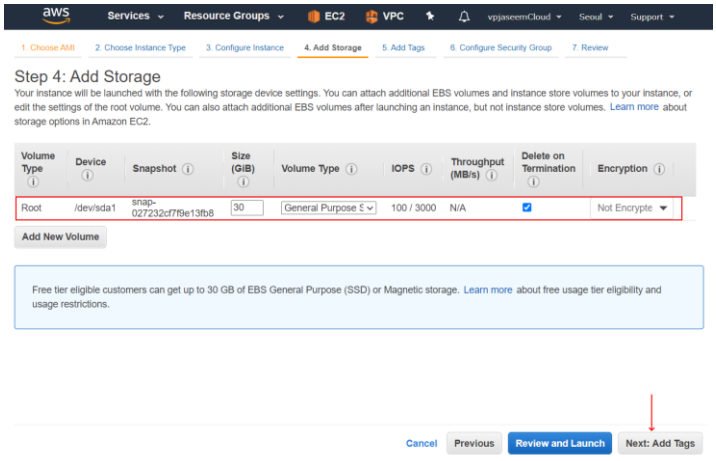

General purpose t2.micro >> Next: Configure Instance Details

AWS Reference: [EC2 Instance Types]



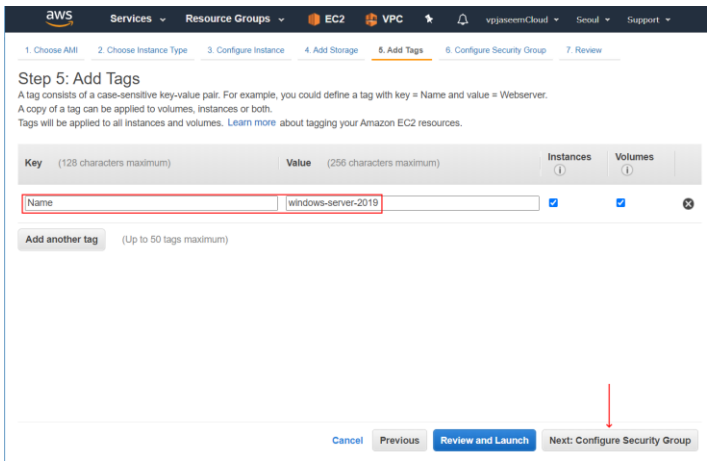

- Configure instance details to suit your requirements.
- Select all default options for now, we will cover every component in future sessions.

>> Next: Add Storage



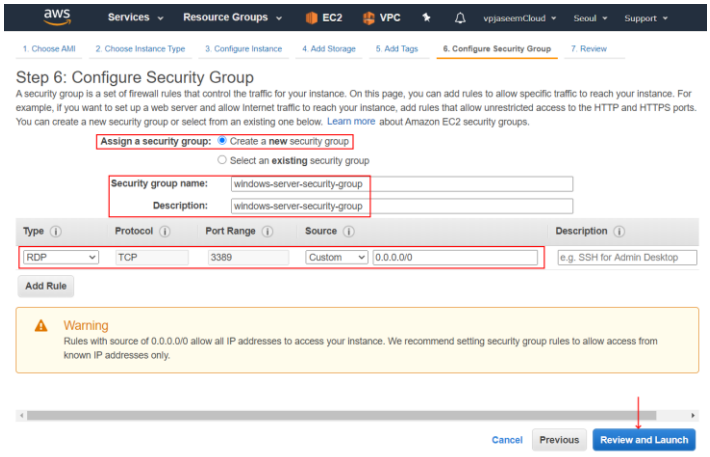

- Configure instance storage here.
- Default will get 30GB C:/ Drive Volume

>> Next: Add Tags

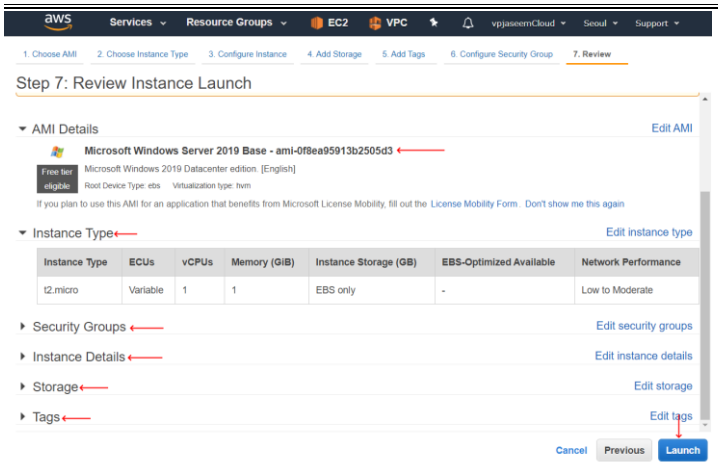



- Tags used to identify the EC2 instances as quickly as possible

>> Next: Configure Security Group

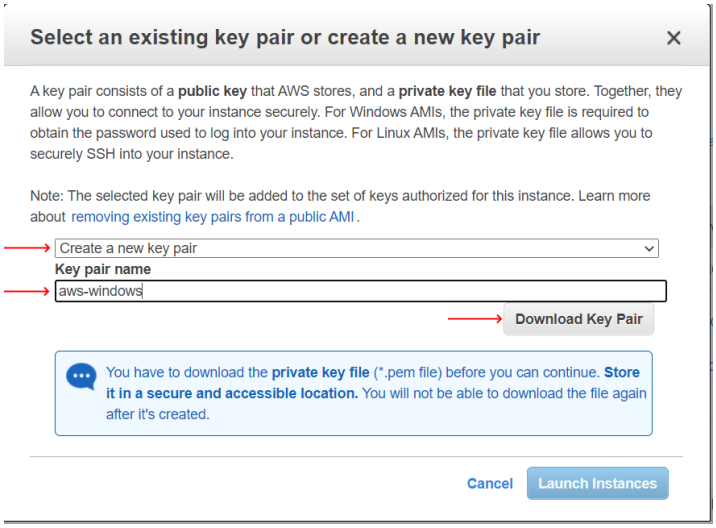



- Select Create a new security group >> provide a Security group name
- Make sure you have RDP (Port 3389) allowed from all IP Address
- >> Review and Launch




- We can review all the settings here. Quickly verify those.

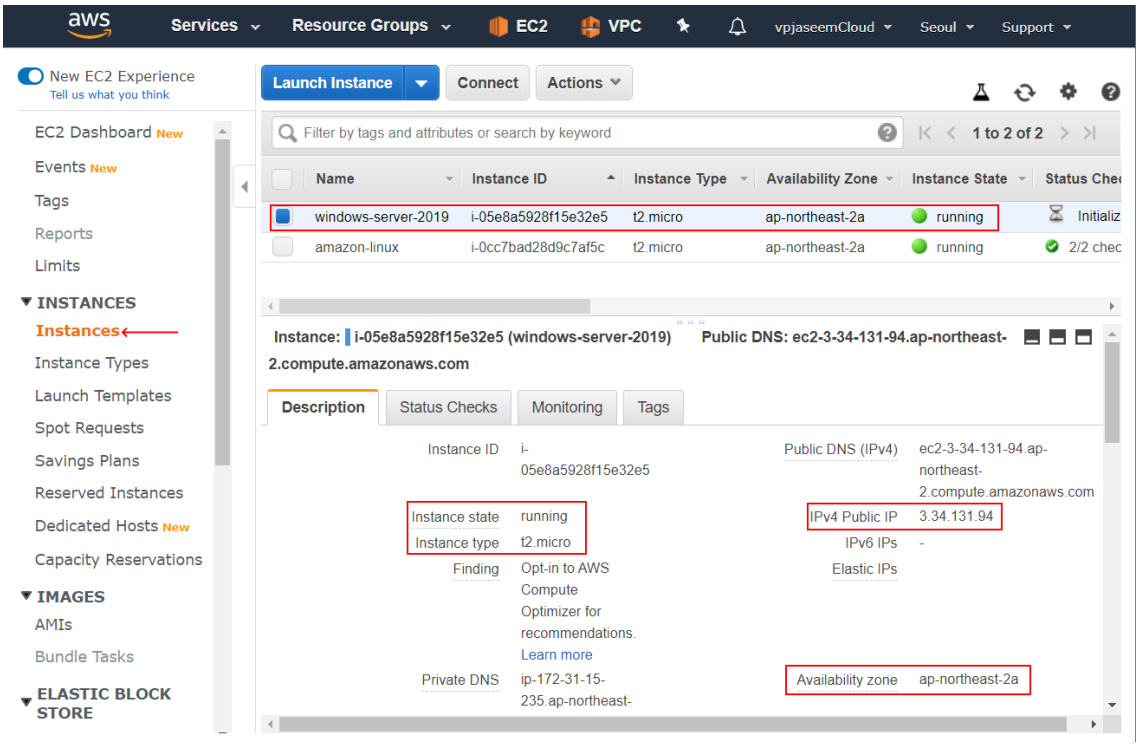
>> **Launch**




- Select 'Create a new key pair' and enter a name for that.
- You can download the 'Private Key'. Keep that in a safe place
- You can also use previous key (aws-linux) if you like to proceed without generating new key

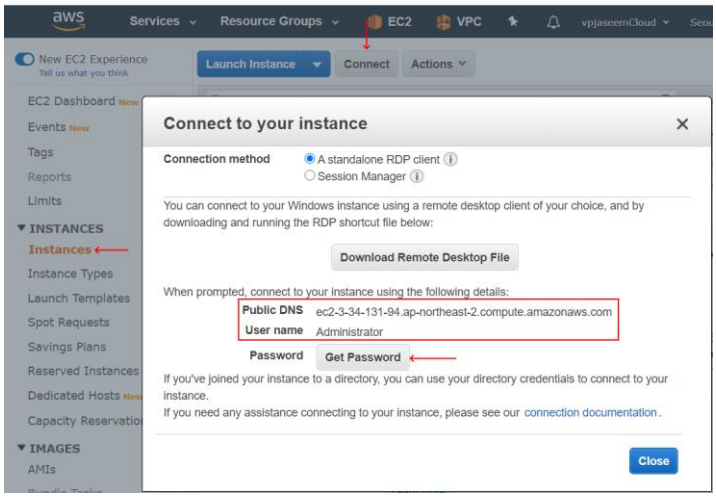
>> **Launch Instances**

- Once the launching is completed, you can click the 'View Instances' button on the same page to see the newly created EC2 instance
- You could see all the details (Status, DNS, Public IP, Type, etc.) in this page

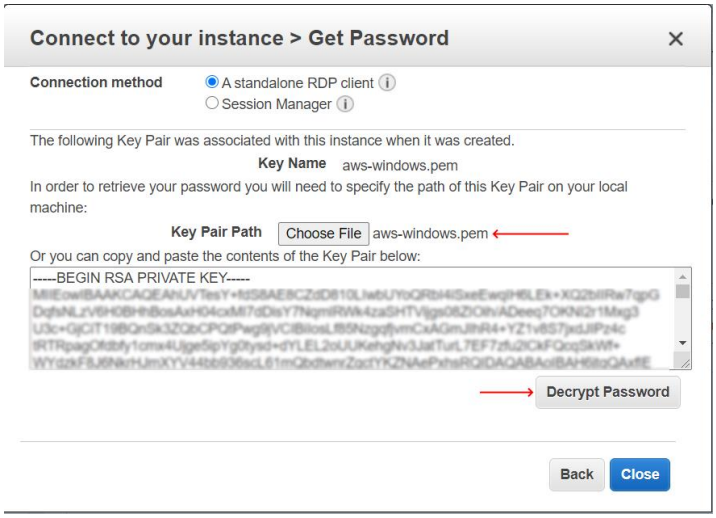


## How to Access Windows Server 2019 AMI EC2 Instance RDP?

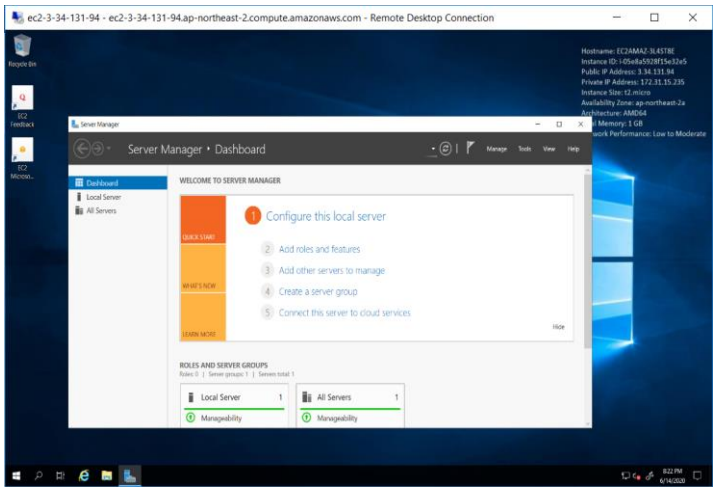
- Windows Server build from AMI can be accessed from anywhere over internet. You should need the .PEM file that downloaded after the creation of an EC2 instance.




- Select the EC2 instance >> Connect >> Note the Public DNS (You can also use the Public IP of the EC2)
- Also note the username here is Administrator
- Click on the Get Password button

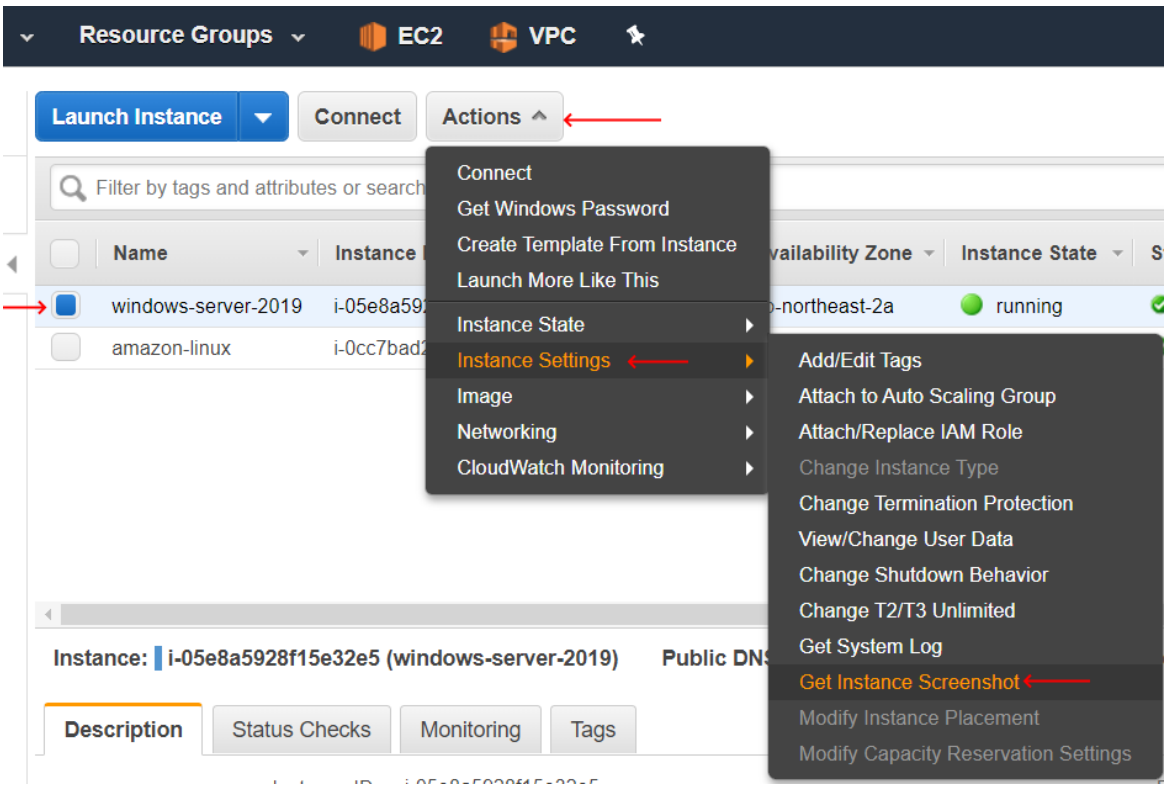


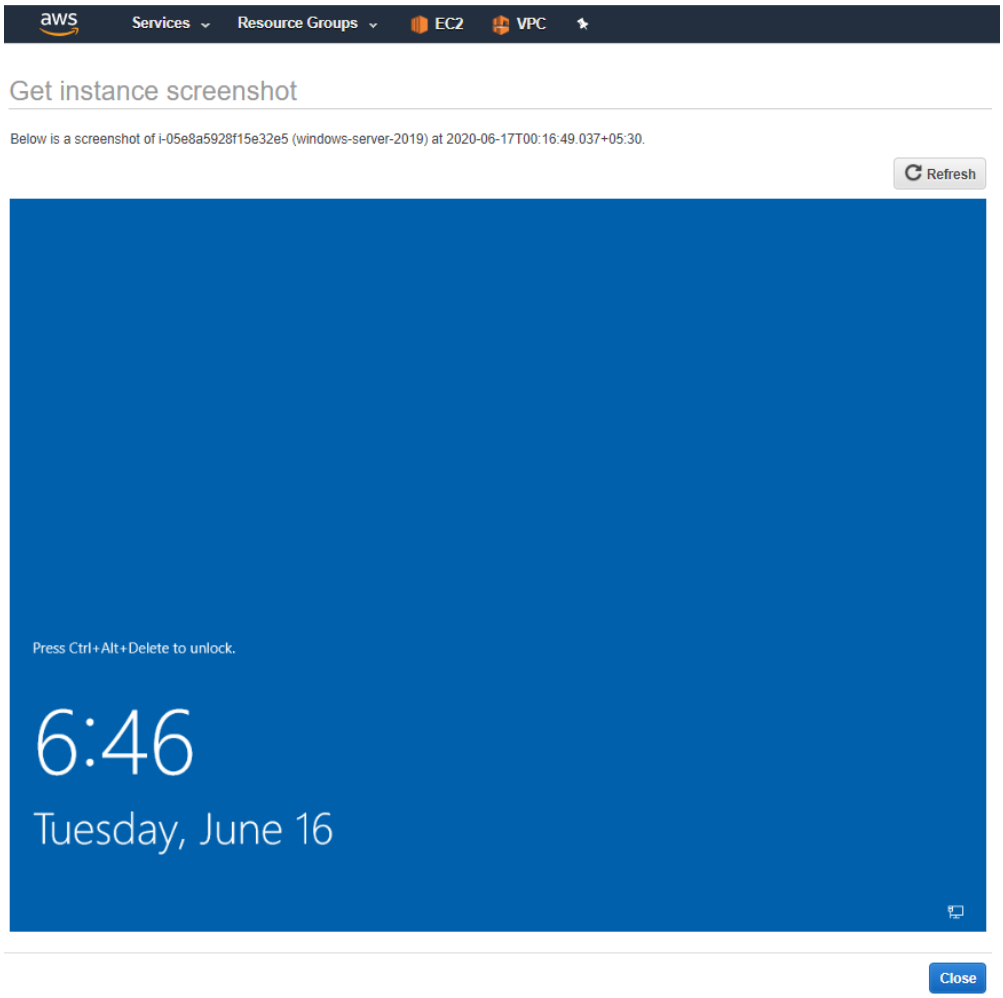

- Choose the .PEM file you downloaded at the last stage of EC2 creation process
- Click Decrypt Password button to get the password




- Once you get the password, use Windows Remote Desktop Connection (DRP) to login to Windows Server 2019

## EC2 Instance Screenshot



## Elastic Network Interface (ENI)

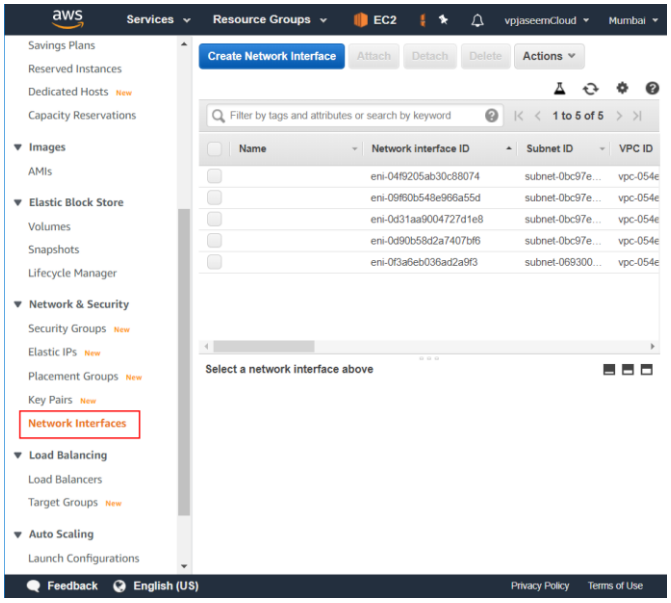
- An elastic network interface is a logical networking component (in a *VPC) that represents a virtual network card. Virtualized NIC for of EC2s.
- Primary ENI of an EC2 instance can not be detached.
- ENI contains below,
  - A primary private IPv4 address from the IPv4 address range of your VPC
  - One or more secondary private IPv4 addresses from the IPv4 address range of your VPC
  - One public IPv4 address (Dynamic Public IP)
  - One Elastic IP address (IPv4) per private IPv4 address (Static Public IP, manually created)
  - One or more IPv6 addresses
  - Security group
  - A MAC addresses
  - A source/destination check flag
  - A description

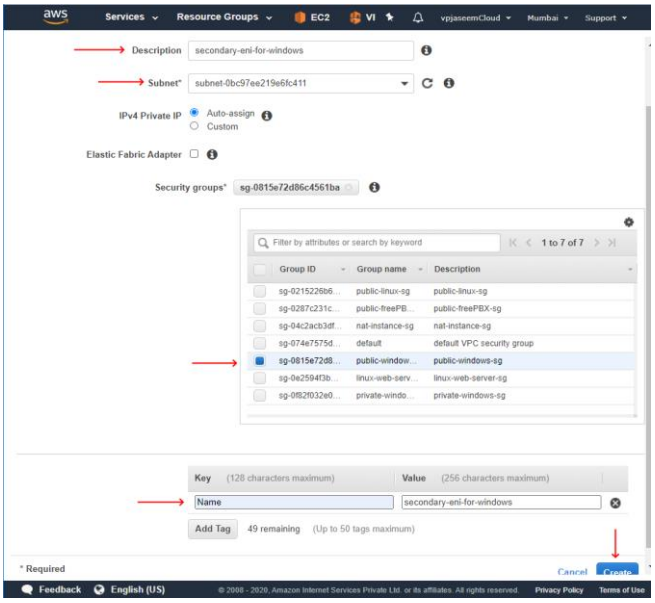
AWS Reference: Elastic network interfaces

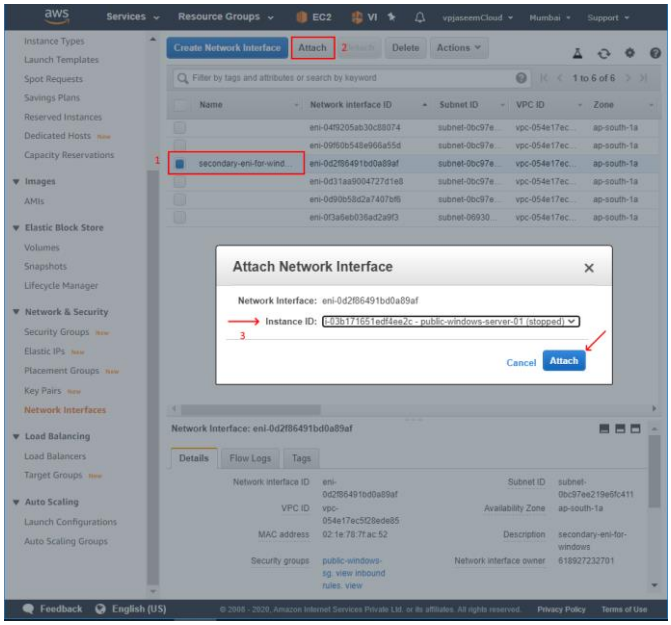
*VPC - Virtual Private Cloud, this will be discussed later.

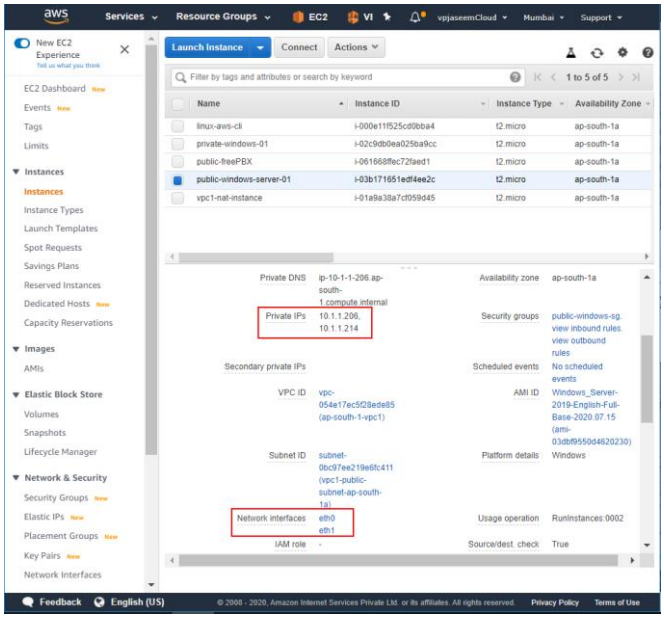
## [LAB] Create Secondary ENI and Add to Windows EC2

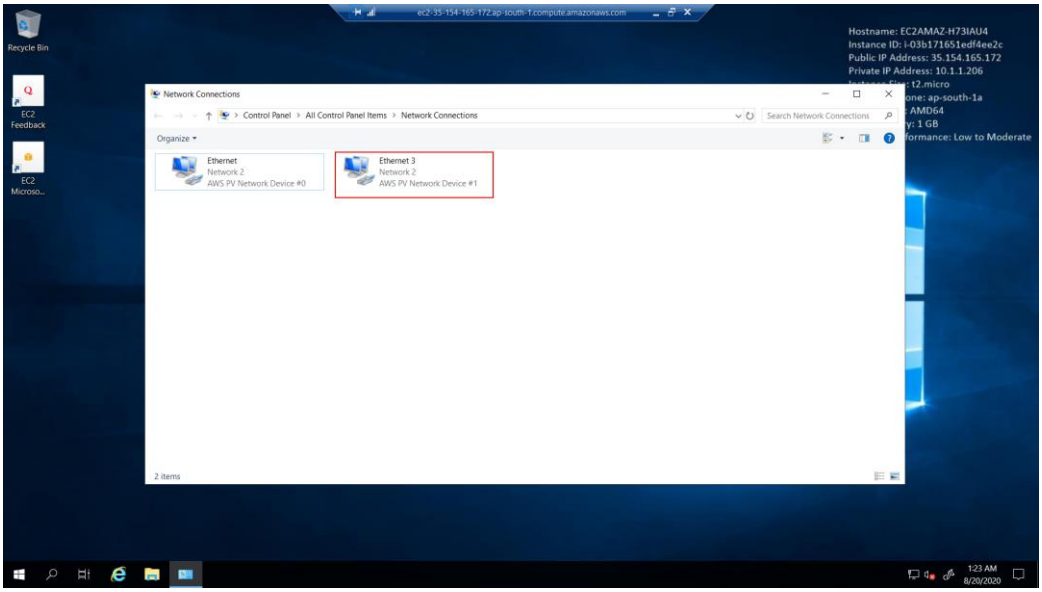
Note: It is recommended to do ENI lab after learning VPC Configurations.



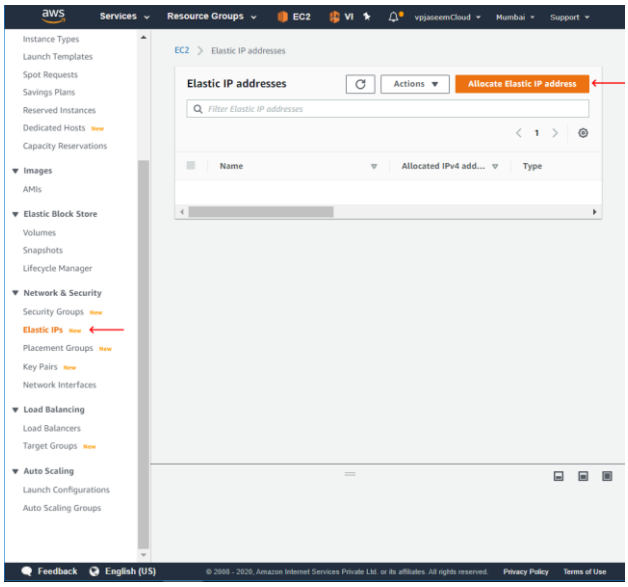
## Elastic IP Address

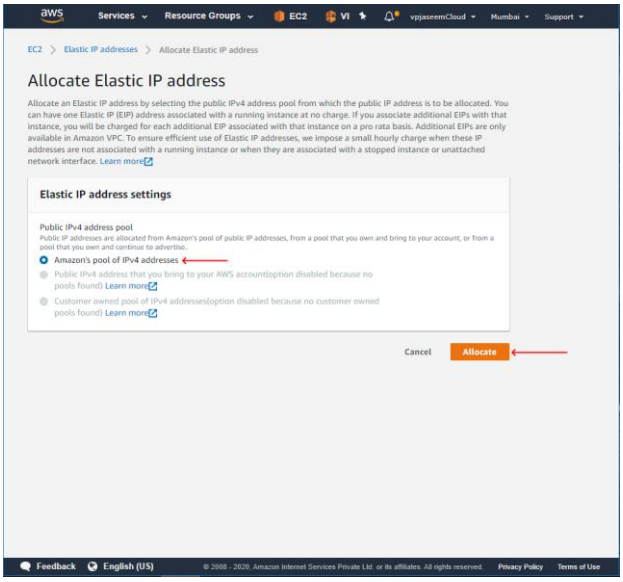
- An Elastic IP address is a static public IPv4 address, which is reachable from the internet.
- If your instance does not have a public IPv4 address, you can associate an Elastic IP address with your instance to enable communication with the internet.
- Elastic IP has to be assigned manually.
- To use an Elastic IP address, you first allocate one to your account, and then associate it with your instance or a network interface.
- The Public IP is dynamic in nature, each time you reboot the instance, it changes. Elastic Public IP is static assignment.
- While your instance is running, you are not charged for one Elastic IP address associated with the instance. If the Elastic IP is not active (Instance is nor running, detached from instance), there will be an hourly charge.

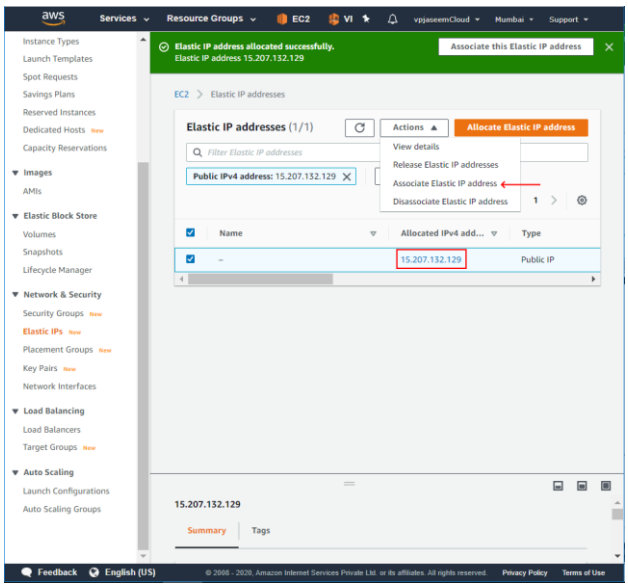
AWS Reference: Elastic IP addresses

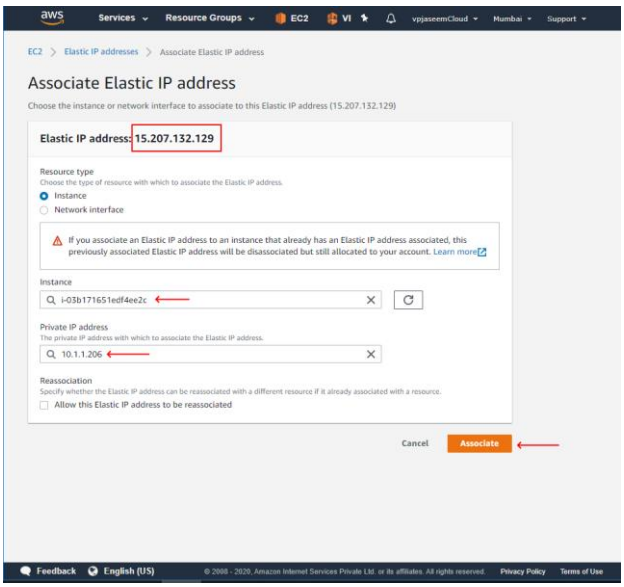
## [LAB] Create Elastic IP and Assign to Windows EC2

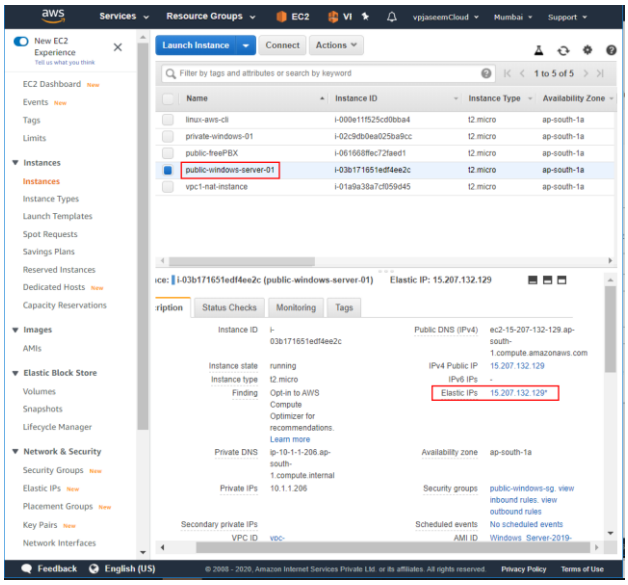
Note: It is recommended to do ENI lab after learning VPC Configurations, also release elastic IP once you complete the labs, elastic IPs are chargeable if not associated to any instance.

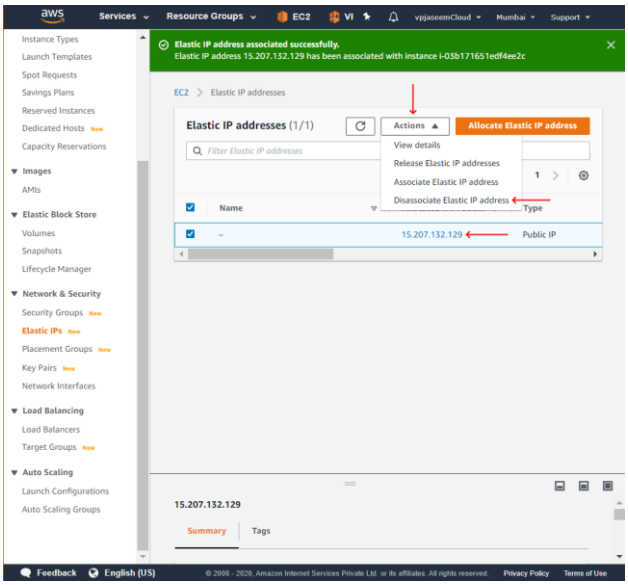


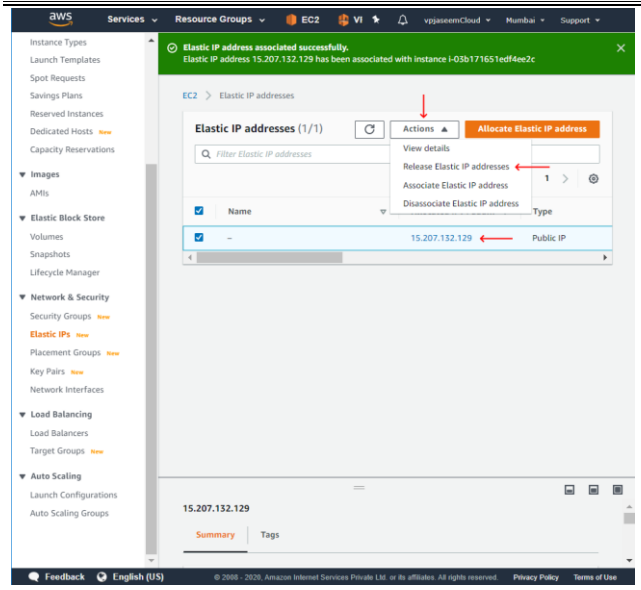



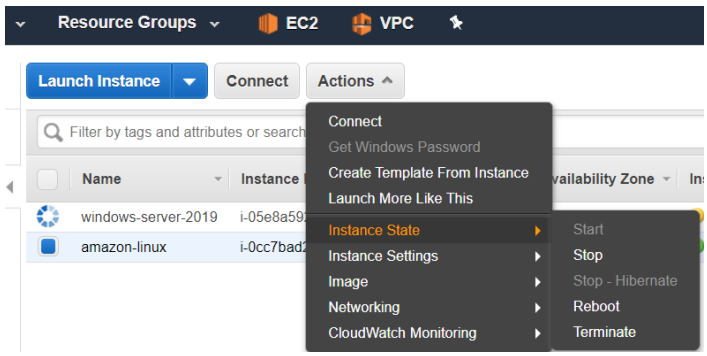



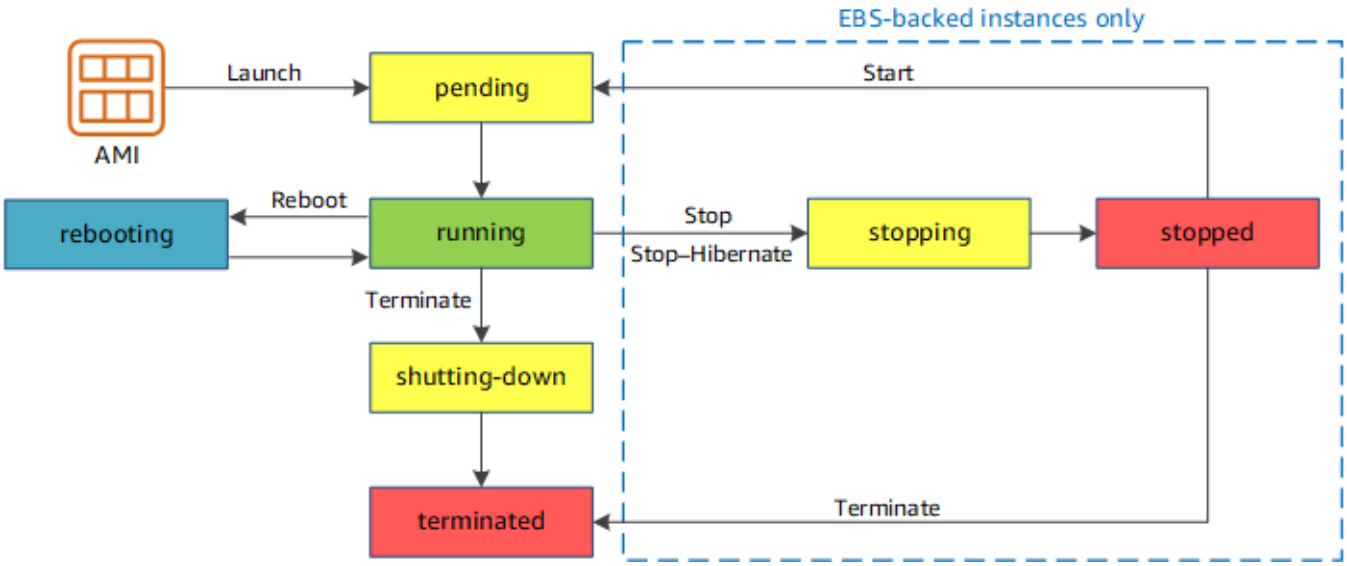




Services
Resource Groups
EC2
vpjaseemCloud
Mumbai
Support
Elastic IP address associated successfully.
Elastic IP address 15.207.132.129 has been associated with instance i-03b171651edf4ee2c
EC2 > Elastic IP addresses
Elastic IP addresses (1/1)
Actions
Allocate Elastic IP address
View details
Release Elastic IP addresses
Associate Elastic IP address
Disassociate Elastic IP address
Filter Elastic IP addresses
Name
Type
15.207.132.129
Public IP
Instance Types
Launch Templates
Spot Requests
Savings Plans
Reserved Instances
Dedicated Hosts
Capacity Reservations
Images
AMIs
Elastic Block Store
Volumes
Snapshots
Lifecycle Manager
Network & Security
Security Groups
Elastic IPs
Placement Groups
Key Pairs
Network Interfaces
Load Balancing
Load Balancers
Target Groups
Auto Scaling
Launch Configurations
Auto Scaling Groups
15.207.132.129
Summary
Tags
Feedback
English (US)
Privacy Policy
Terms of Use

## Life Cycle of an EC2 Instance




| Instance state | Description |
|---|---|
| pending | The instance is preparing to enter the running state. An instance enters the pending state when it launches for the first time, or when it is started after being in the stopped state. |
| running | The instance is running and ready for use. |
| stopping | The instance is preparing to be stopped or stop-hibernated. |
| stopped | The instance is shut down and cannot be used. The instance can be started at any time. |
| shutting-down | The instance is preparing to be terminated. |
| terminated | The instance has been permanently deleted and cannot be started. |

- Shutting down from the OS is equivalent to Stop option in AWS

AWS Reference: Instance lifecycle

## EC2 Instance Machine Types

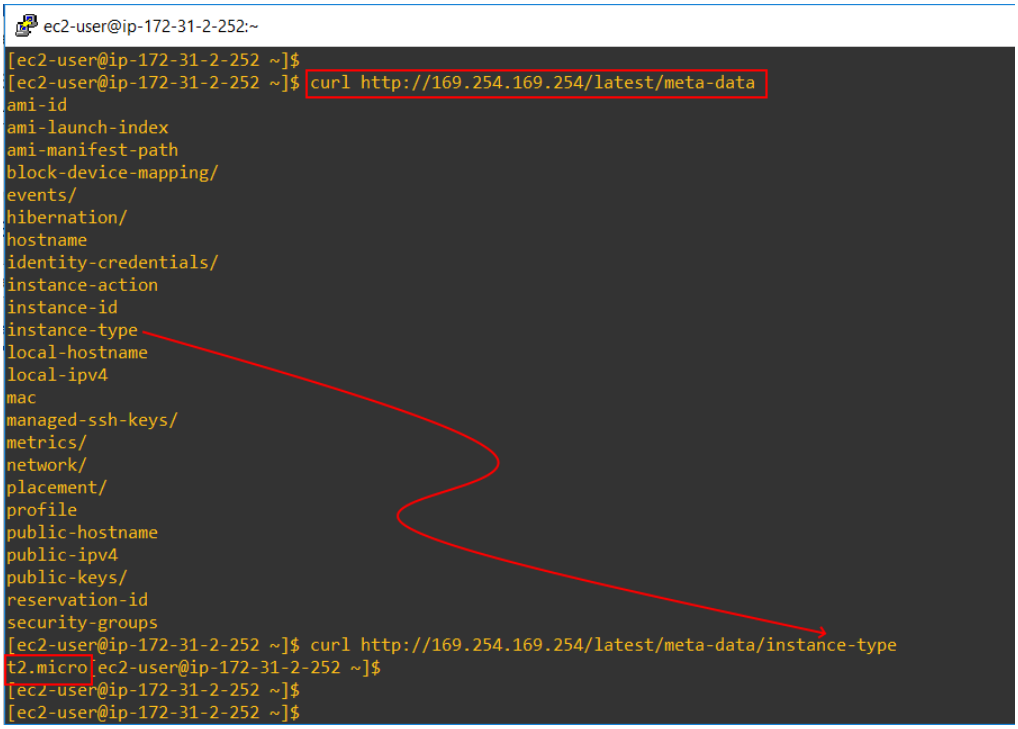
Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instance types comprise varying combinations of CPU, memory, storage, and networking capacity and give you the flexibility to choose the appropriate mix of resources for your applications. Each instance type includes one or more instance sizes, allowing you to scale your resources to the requirements of your target workload.

| Family | Instance sizes | Use Cases |
|---|---|---|
| General Purpose | A1, T3, T3a, T2, M6g, M5, M5a, M5n, M4 | • Low traffic web applications<br>• Small & Medium size database |
| Compute Optimized | C6g, C5, C5a, C5n, C4 | • Hight performance front-end<br>• Video encoding |
| Memory Optimized | R6g, R5, R5a, R5n, R4, X1e, X1, Hight Memory, z1d | • Hight performance Databases<br>• Distributed memory caches |
| Storage Optimized | I3, I3en, D2, H1 | • Data warehousing<br>• Log / Data processing applications |
| Accelerated Computing (GPU Optimized) | P3, P2, Inf1, G4, G3, F1 | • 3D Application streaming<br>• Machine learning |

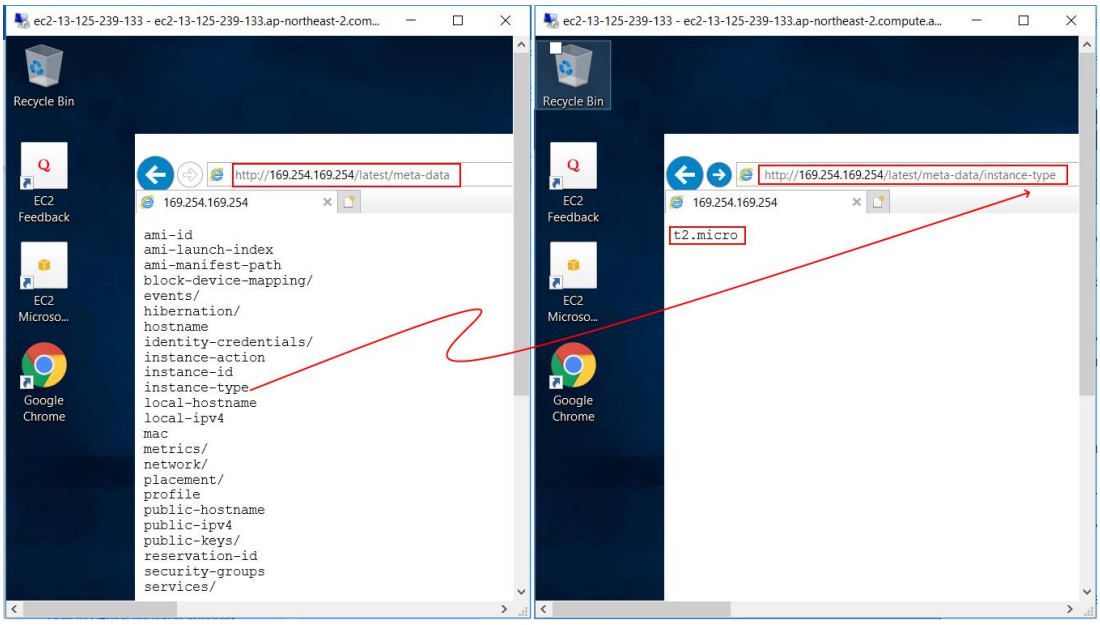
AWS reference: Amazon EC2 Instance Types

## Meta Data & User Data of EC2 Instances

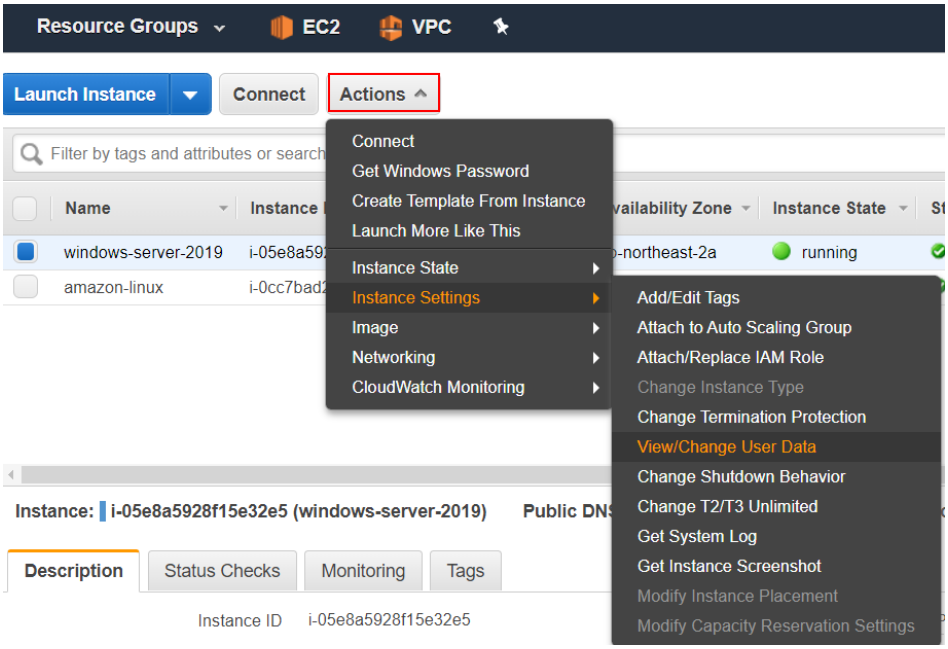
- In memory key value pairs that are accessible from the EC2 instance
- **Instance Meta Data:** Filled by AWS when you spin-up your EC2. It is the data about your instance. Can be used to configure and manage running instances via Scripts (e.g. Python Automated tasks)
- To view the meta data keys of a Linux EC2, use **curl** http://169.254.169.254/latest/meta-data then embed the 'key' name at the last of the URL as shown below to get the value. e.g. http://169.254.169.254/latest/meta-data/instance-type

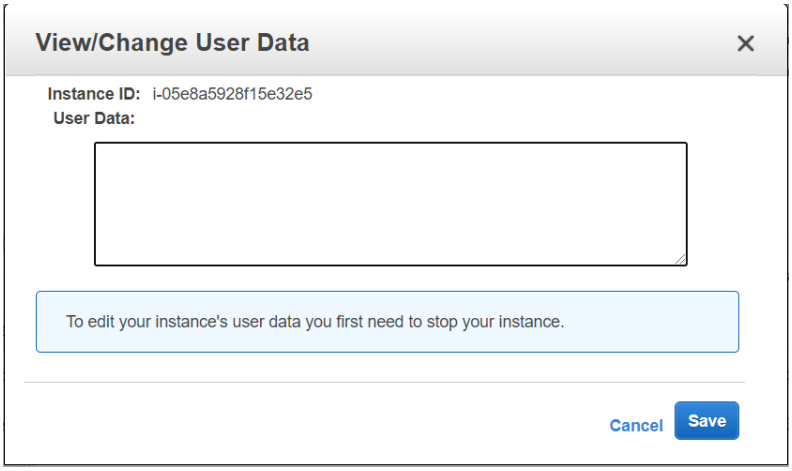



- To view the metadata of a Windows EC2, use browse http://169.254.169.254/latest/meta-data then embed the 'key' name at the last of the URL as shown below to get the value. e.g. http://169.254.169.254/latest/meta-data/instance-type



- **Instance User Data:** Can be passed to the instance at launch, used to run startup scripts, can be used to perform common automated tasks
- It can be Linux bash scripts or Windows batch / PowerShell scripts
- Use this URL to obtain User Data http://169.254.169.254/latest/user-data

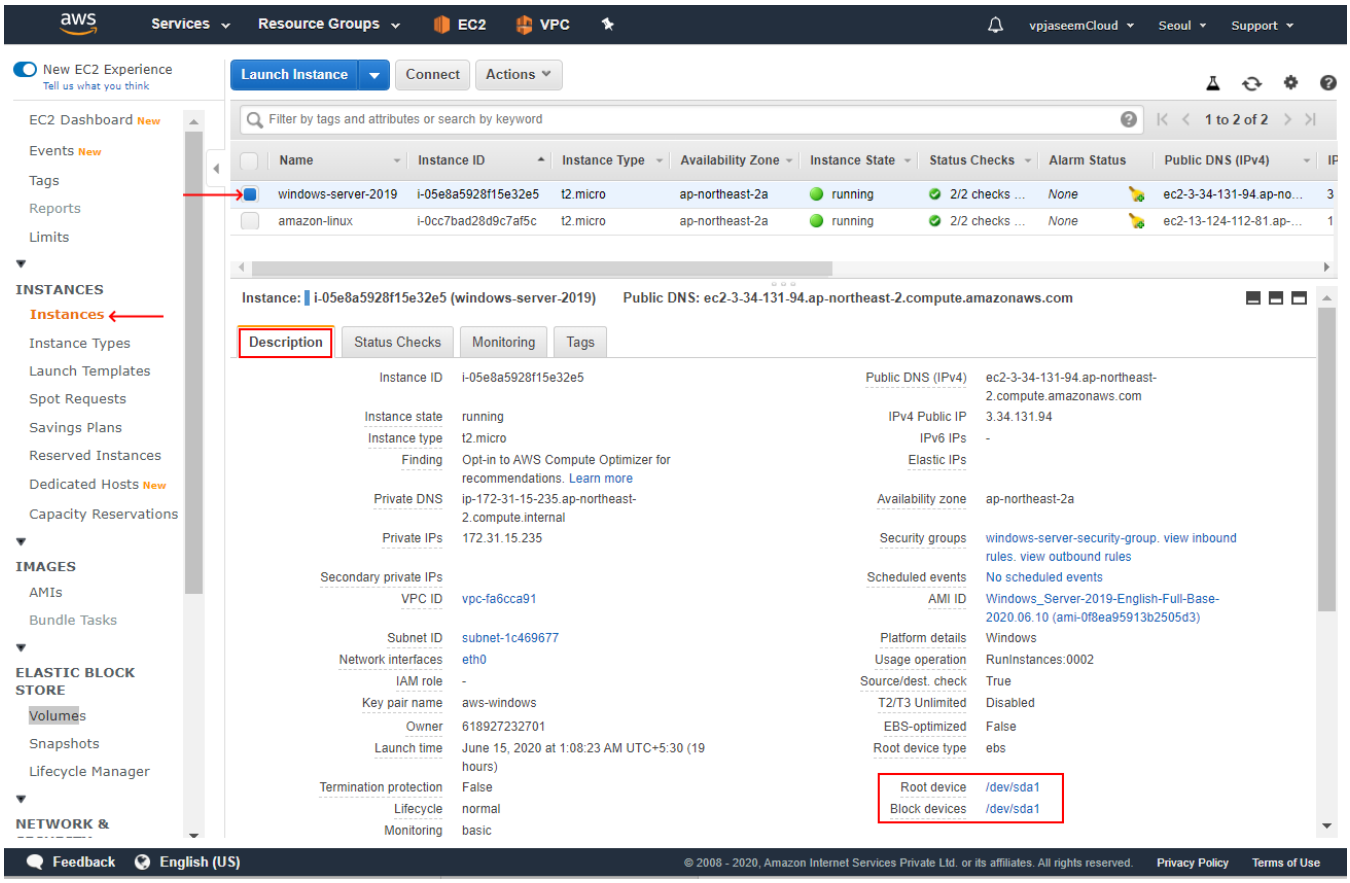



Note: Lab for User date is covered in the Web App hosting section.

## Storage Options for EC2 Instances

Now let's take a look at the Root Volume configurations of an EC2 Instance. When we start an EC2 instance, the fundamental things that you have to choose what sort of storage you want? specifically the root volume storage where you OS is located.

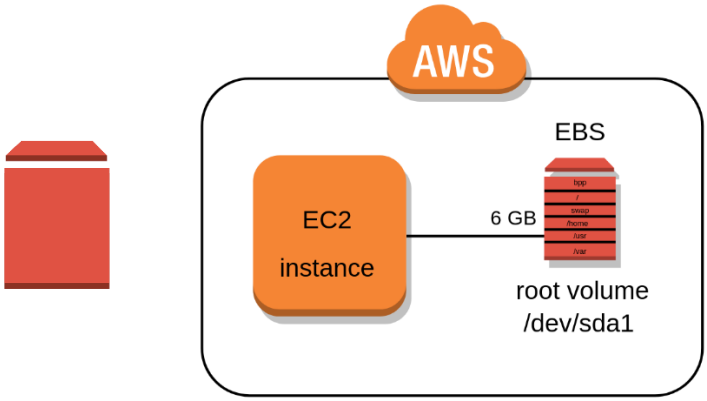



Where is the root volume stored in an EC2? We have two options available,

### 1. Instance Store Backed

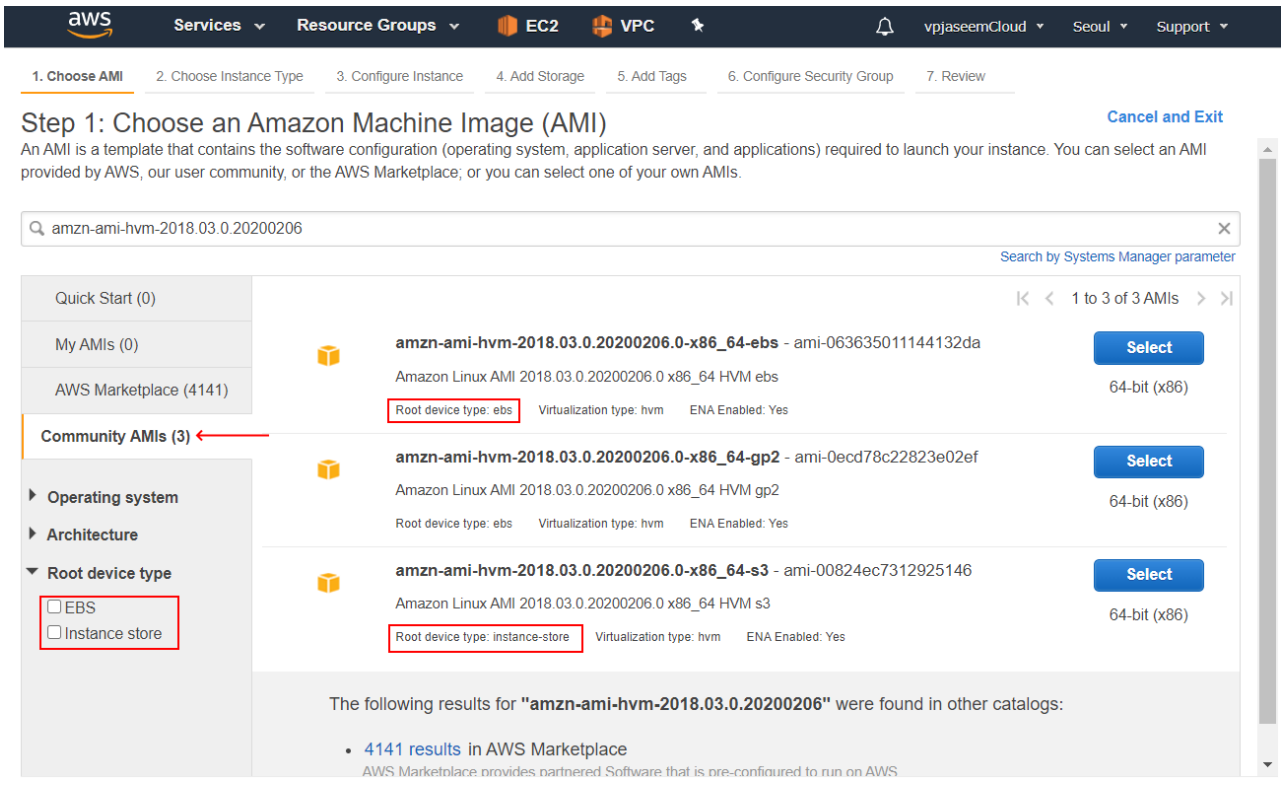
- It is a Physical Drive that is attached to the host computer that hosts the EC2 VM
- When you stop / terminate the EC2, the Instance Store loses its date, it evaporates
- Data on Instance Store is non-persistent
- Max size 5GB
- Can't be stopped

## 2. AWS Elastic Block Store (EBS)

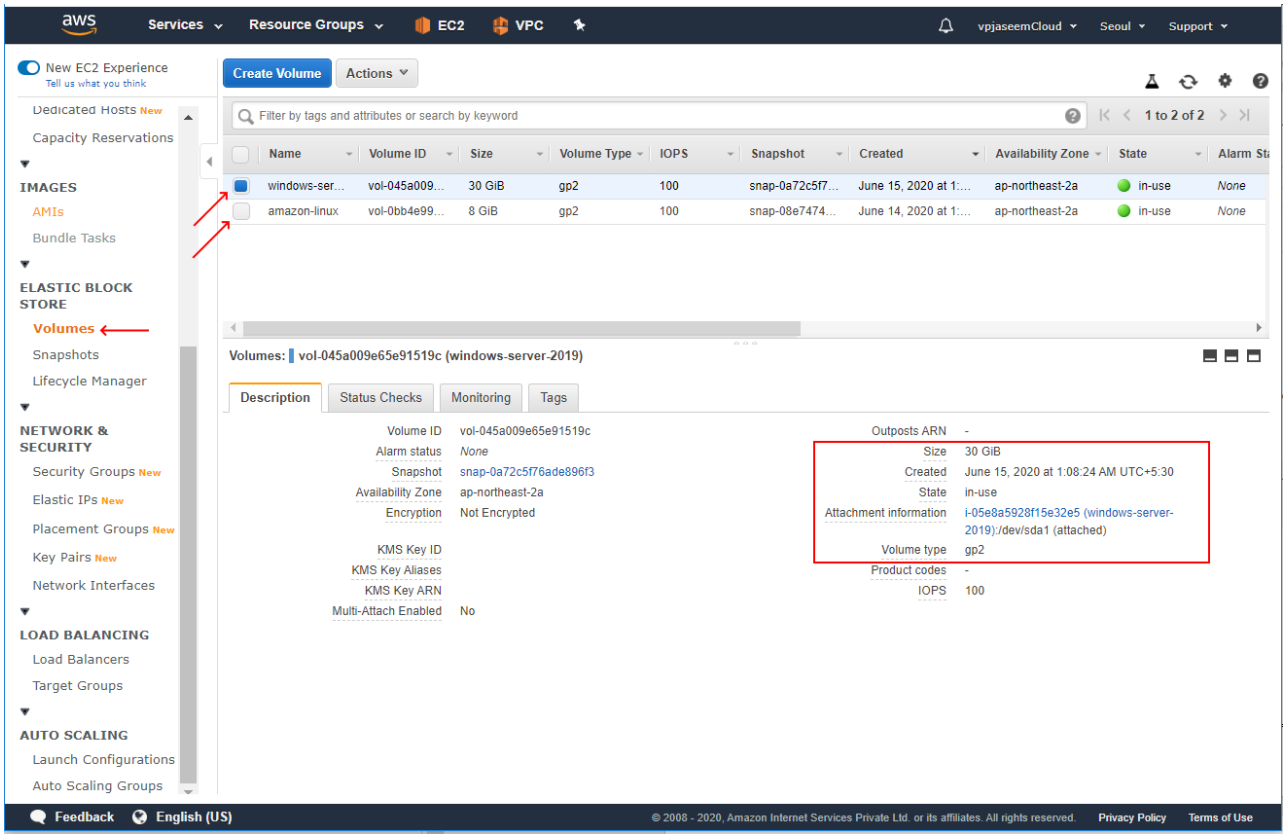


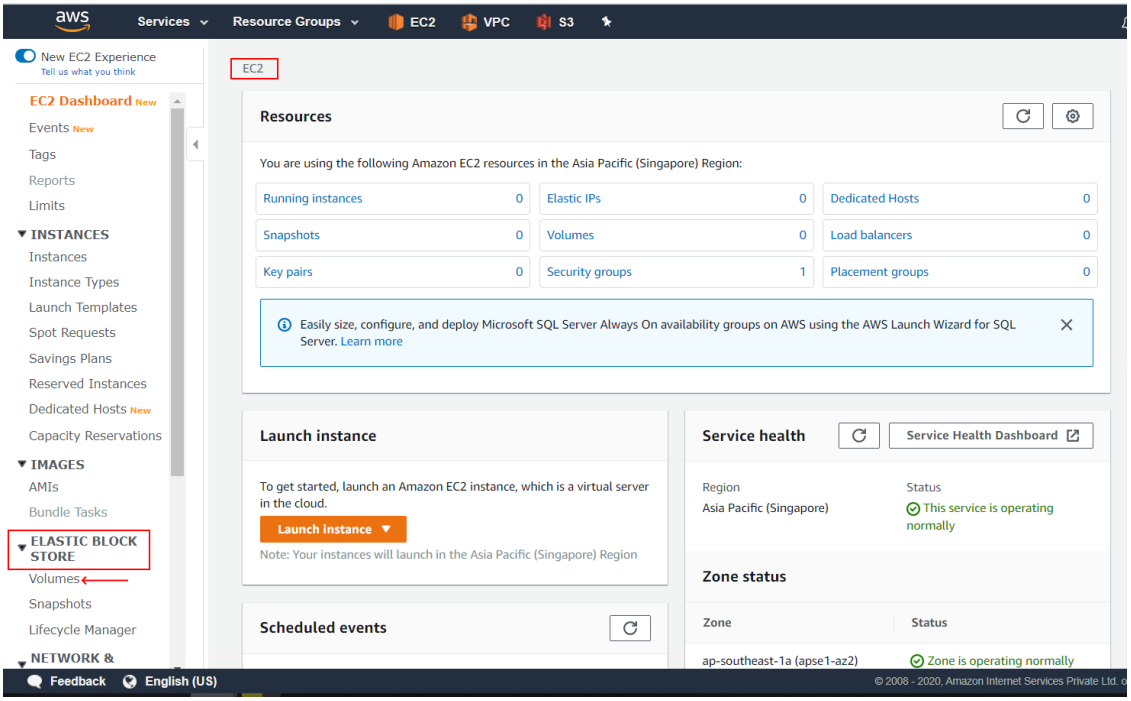

- We don't see any configuration menu / button for EBS, this is a storage mechanism tied closely to EC2 instance
- We can call it as Virtual Hard Disk in the cloud (like .VHD file in vmware)
- We could create EBS at the time of creation of EC2 or independently
- Based on SSD (1GB to 16TB capacity) or Magnetic drive (1GB to 1TB capacity)
- We can take snapshot of EBS and store in *S3
- Pay for what you provision, IOPS (Input Output Per Second) also charged. Also, charges are based on region.
- Default behavior is to delete the EBS volume along with the EC2 while terminating, but you can override that to keep data on EBS store.
- Data on Instance Store is persistent, Can be stopped and modified the configurations inside

While selecting the AMI, we can see the root device type.



| AWS EBS | AWS S3* |
|---|---|
| Cloud disk drive | Cloud based storage |
| Block storage file system based | Object Store |
| Redundancy - Availability Zone | Redundancy - Region |
| No access via Internet* | Access via Internet* |



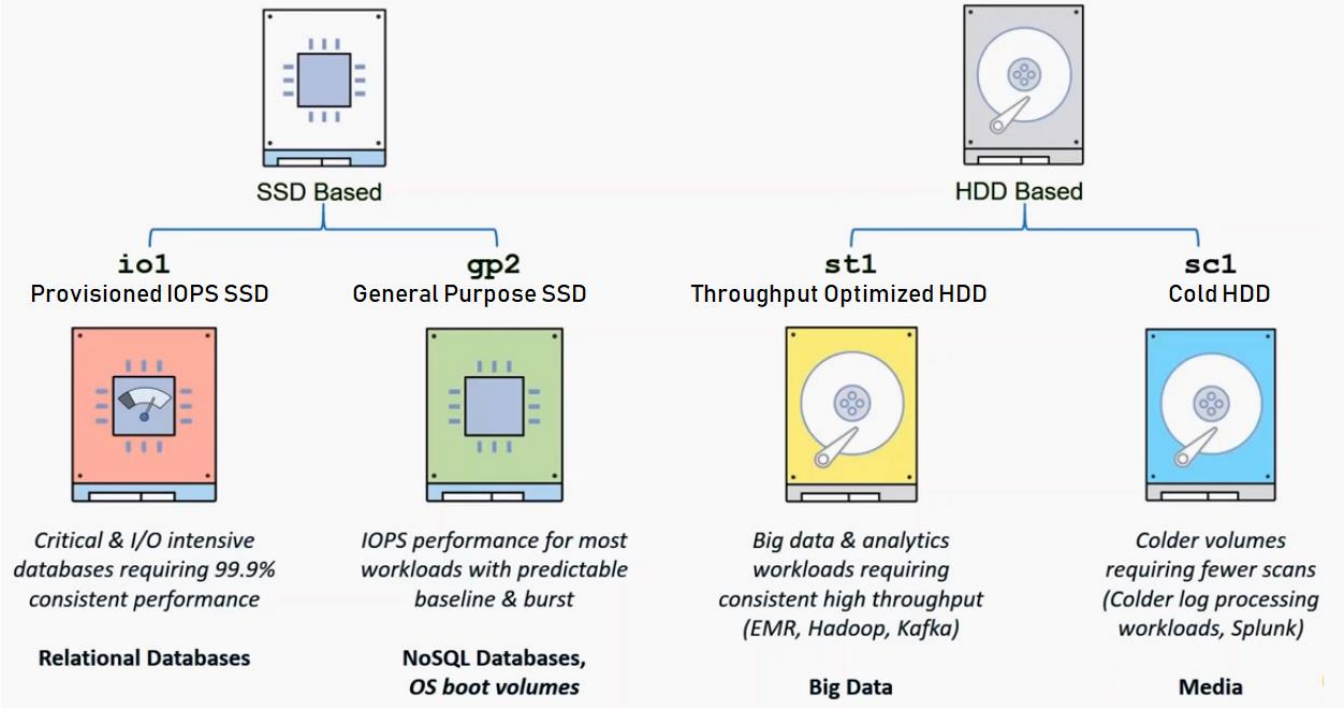




You can access / create new EBS drive from the Volumes options.

**S3: Simple Storage Service: This will be covered later*

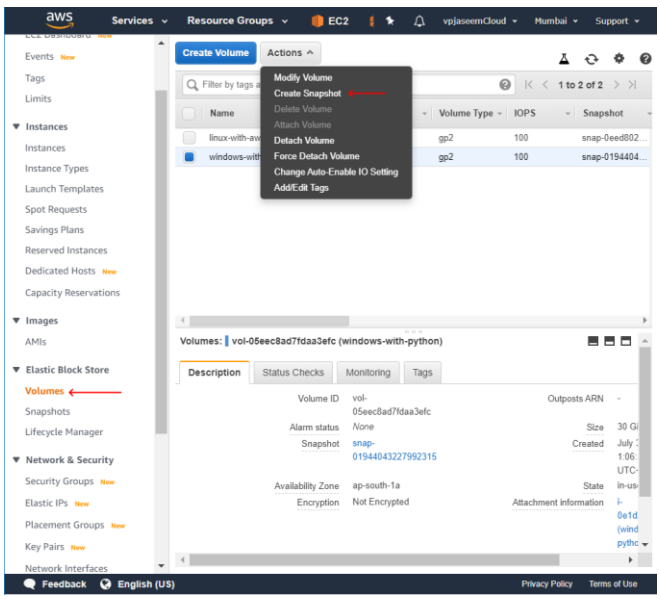
- Block storage for EC2 instance (root volume), high performance and low latency.
- We can attach and detach EBS to any one instance at a time
- Scope EBS is AZ
- To move EBS between AZs, take a snapshot and move it in to different AZs
- Fault tolerance is built in to EBS, AWS will create replicated copy in same AZ called Secondary volume. It is not accessible but automatically used in case of primary volume failure.
- We can alter the performance of EBS (IOPs, Latency, Throughput)
- **IOPS**: Input Output (Read/Write) per second
- **Latency**: Time in milli seconds between IO submission and complete. Consider in Disk level.
- **Throughput**: Read/Write transfer rate (MB/s); IOPS x Data Rate
- Which one to select? IOPS Optimized is preferred for Mobile Data (small data in transaction). Movie Streaming basically takes Throughput optimized volumes.
- We can increase the Disk Size but not decrease
- To reduce the size, create a snapshot of bigger volume and create a new small volume based on the snapshot.
- We can change the volume types (e.g. io1 to st1) but not all the combinations.
- We can increase or decrease IOPS

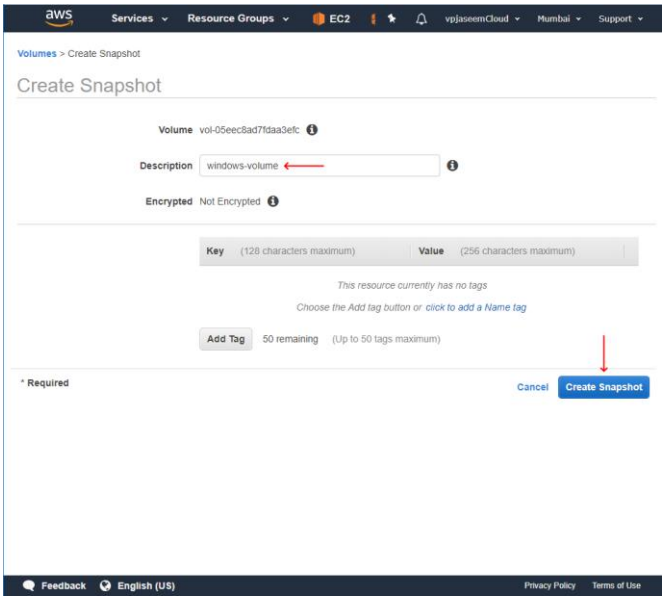
Types of EBS



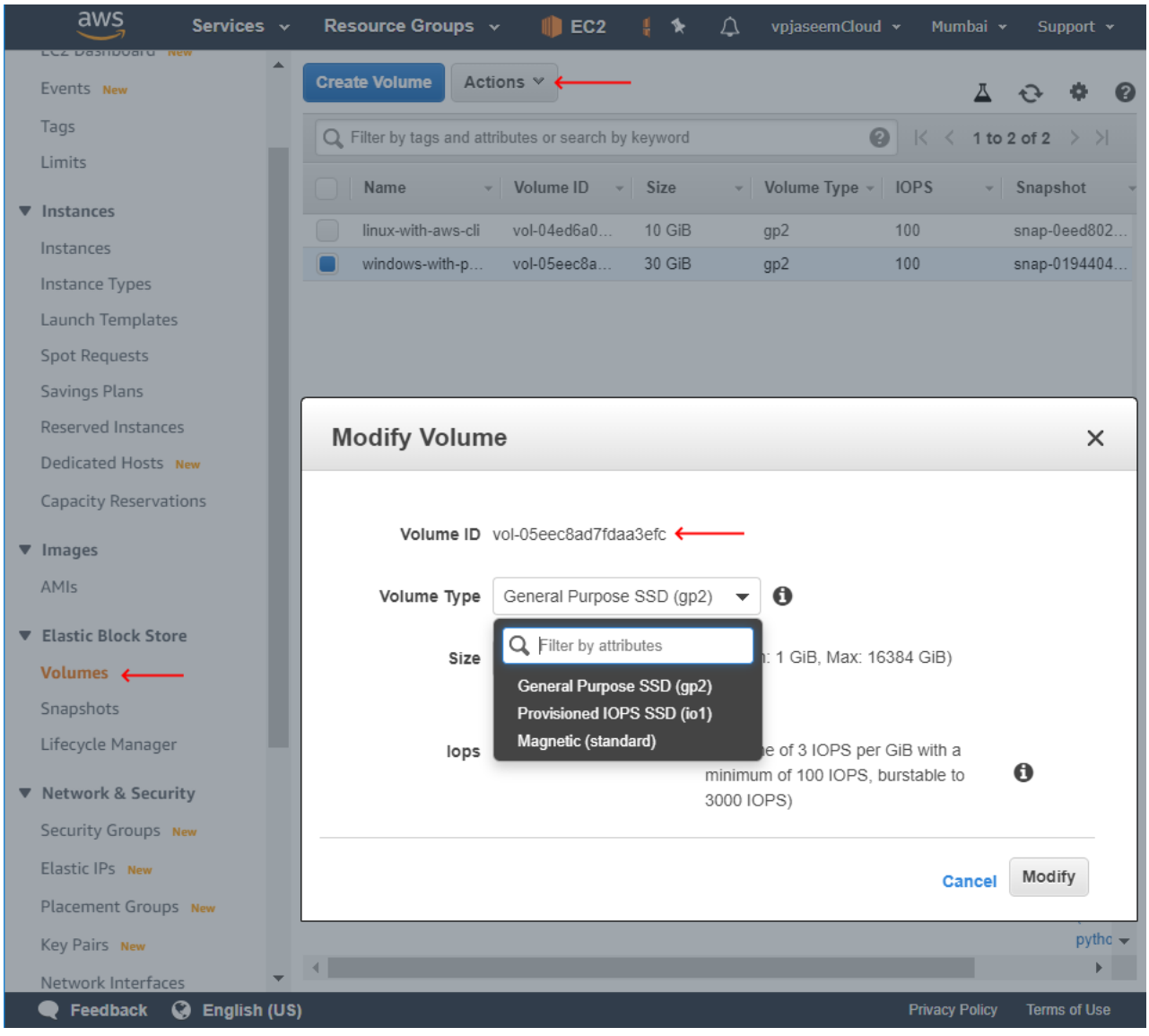

AWS Reference: Types of EBS

- Snapshots are the image of an EBS, it can be used to backup an EBS and create new from it

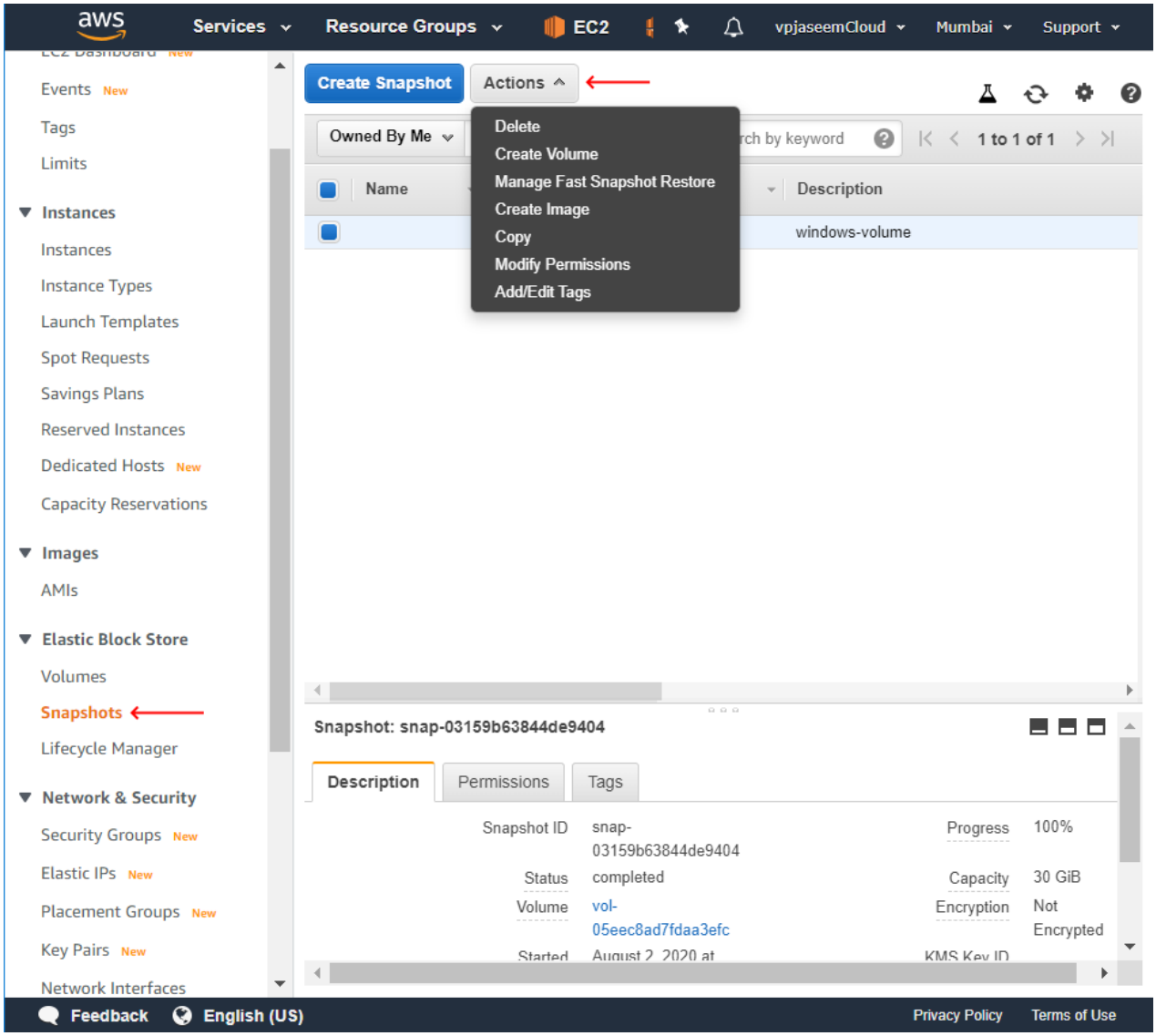



- We can modify the volume type whenever required



- From the snapshot, we can create new volume

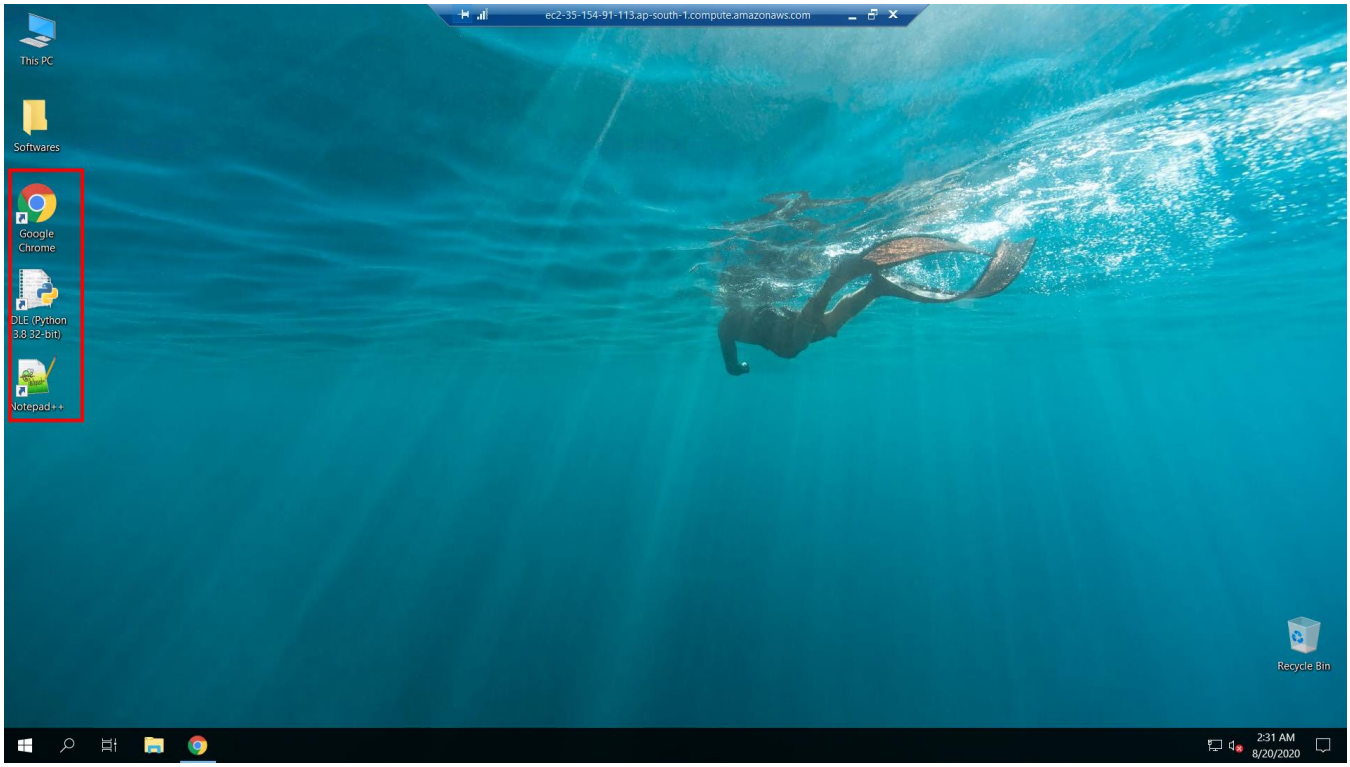


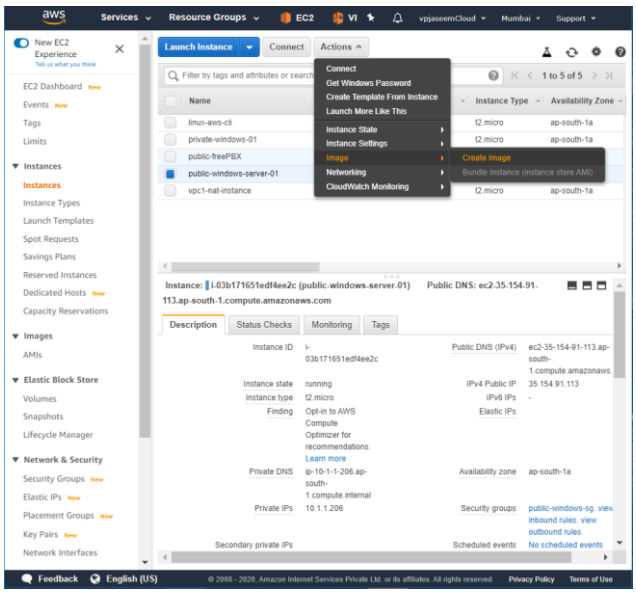
## EC2 Placement Group

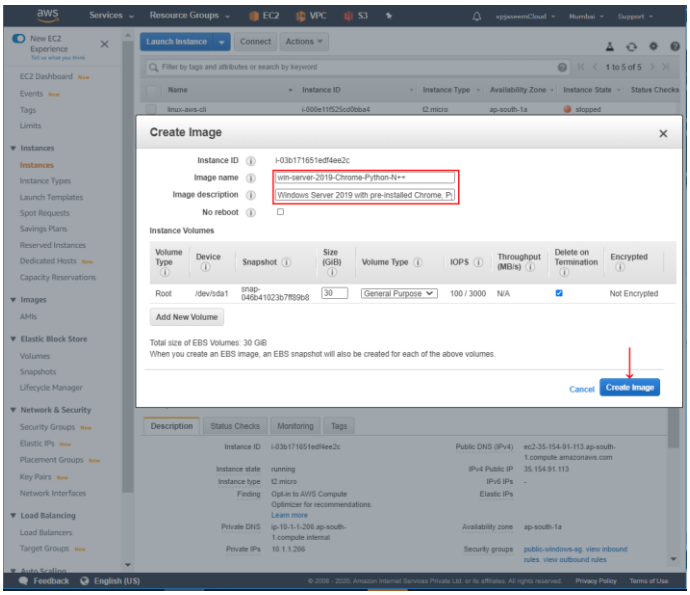
- When you launch a new EC2 instance, the EC2 service attempts to place the instance in such a way that all of your instances are spread out across underlying hardware to minimize correlated failures.
- You can use placement groups to influence the placement of a group of interdependent instances to meet the needs of your workload.
  - **Cluster** – packs instances close together inside an Availability Zone. This strategy enables workloads to achieve the low-latency network performance necessary for tightly-coupled node-to-node communication that is typical of HPC applications.
  - **Partition** – spreads your instances across logical partitions such that groups of instances in one partition do not share the underlying hardware with groups of instances in different partitions. This strategy is typically used by large distributed and replicated workloads like Hadoop.
  - **Spread** – strictly places a small group of instances across distinct underlying hardware to reduce correlated failures.

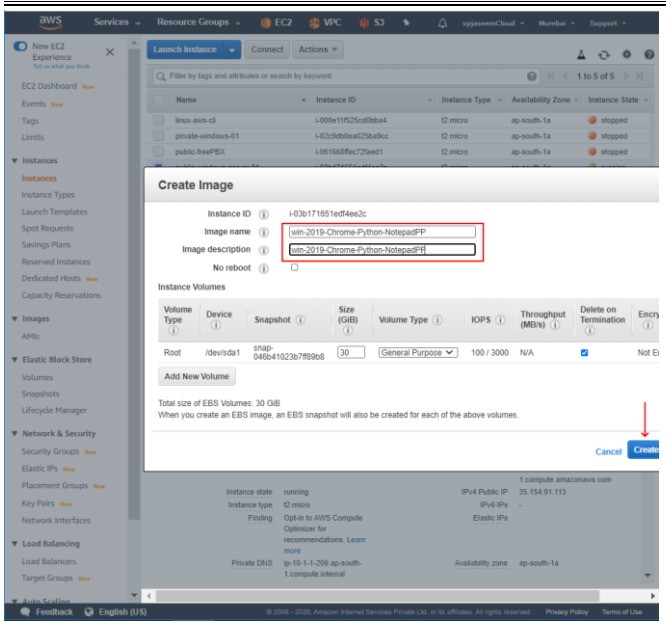
## Creating Custom AMI

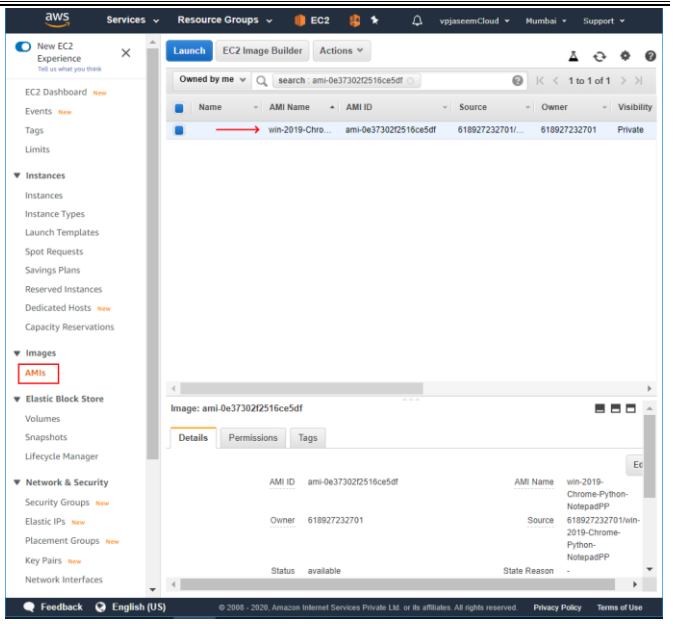
- AMI provides information required to launch an EC2 instance in cloud. AMI contains the root volume for the instance to keep the OS files.
- Once we have an instance, we can install software, patches, etc. and build another image from it. This will save some time in future. You will have the software pre-installed on new image.
- Select the instance >> Actions >> Image >> Create Image
- You can build new EC2 instances from this image in future.

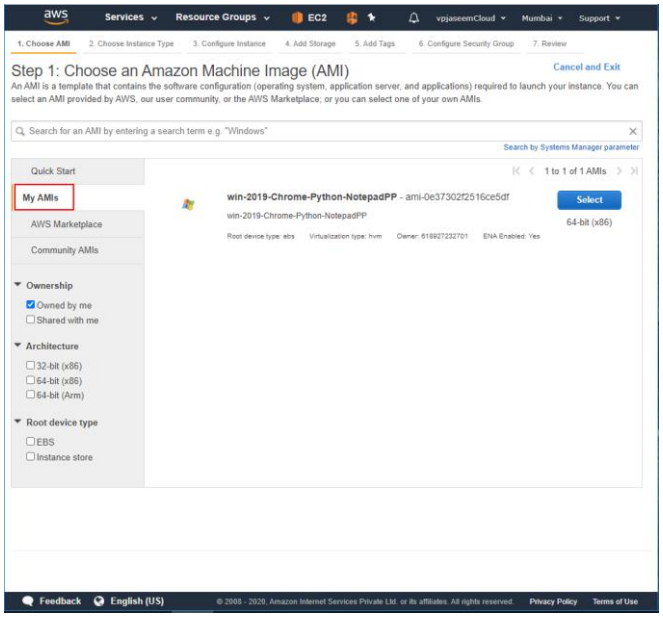



Create Image
Instance ID
i-03b171651edf4ee2c
Image name
win-2019-Chrome-Python-NotepadPP
Image description
win-2019-Chrome-Python-NotepadPP
No reboot
Instance Volumes
Add New Volume
Total size of EBS Volumes: 30 GiB
When you create an EBS image, an EBS snapshot will also be created for each of the above volumes.
Cancel
Create


Launch
EC2 Image Builder
Actions
win-2019-Chro...
ami-0e37302f2516ce5df
618927232701
Private
Image: ami-0e37302f2516ce5df
Details
Permissions
Tags
AMI ID
ami-0e37302f2516ce5df
AMI Name
win-2019-Chrome-Python-NotepadPP
Owner
618927232701
Source
618927232701/win-2019-Chrome-Python-NotepadPP
Status
available


Step 1: Choose an Amazon Machine Image (AMI)
Cancel and Exit
An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. You can select an AMI provided by AWS, our user community, or the AWS Marketplace; or you can select one of your own AMIs.
Search for an AMI by entering a search term e.g. "Windows"
Search by Systems Manager parameter
Quick Start
My AMIs
AWS Marketplace
Community AMIs
Ownership
Owned by me
Shared with me
Architecture
32-bit (x86)
64-bit (x86)
64-bit (Arm)
Root device type
EBS
Instance store
win-2019-Chrome-Python-NotepadPP - ami-0e37302f2516ce5df
win-2019-Chrome-Python-NotepadPP
Select
64-bit (x86)

## EC2 Pricing Model

### On-Demand / Pay as you Go

- With On-Demand instances, you pay for compute capacity by the hour or the second depending on which instances you run.
- No longer-term commitments or upfront payments are needed.

### Reserved

- Reserved Instances provide you with a significant discount (up to 75%) compared to On-Demand instance pricing.
- Cheap and long-term lock (1 to 3 years)

### Savings Plans

- Savings Plans is a flexible pricing model that provides savings of up to 72% on your AWS compute usage.
- For a known burst period (e.g. Black Friday sale)

### Spot

- Amazon EC2 Spot instances allow you to request spare Amazon EC2 computing capacity for up to 90% off the On-Demand price
- Other AWS users sell their Reserved or Savings Plans if they are not using it, its like bidding

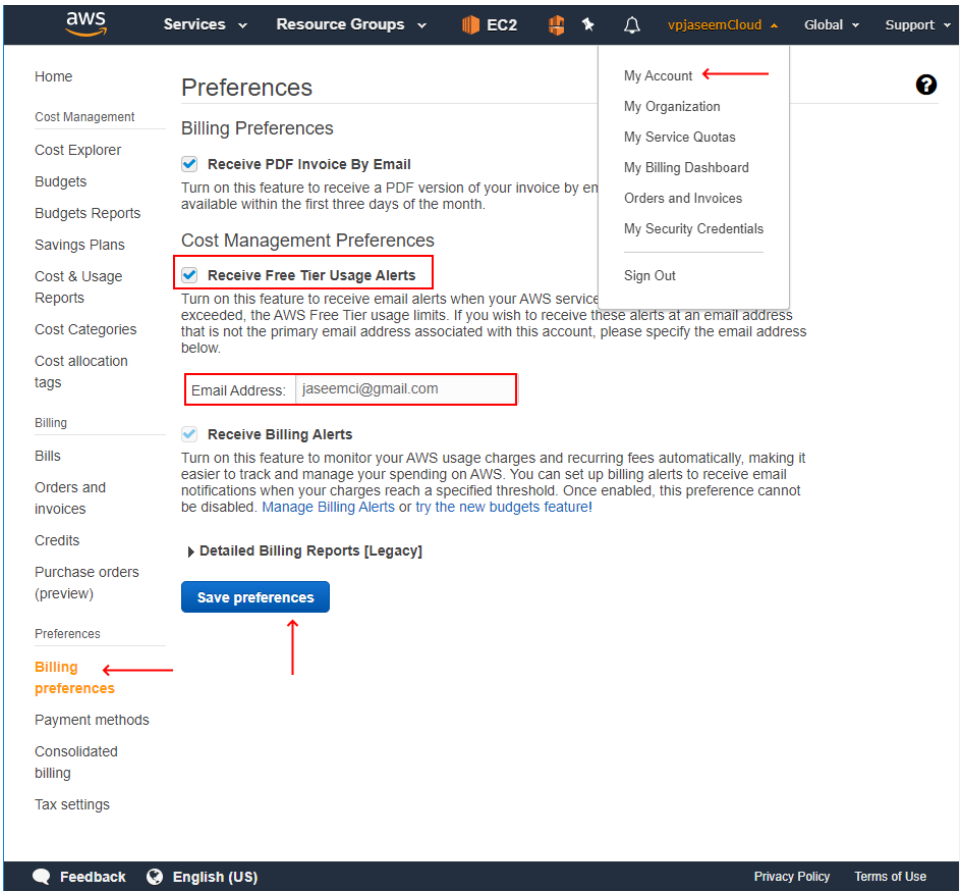
### Dedicated Host

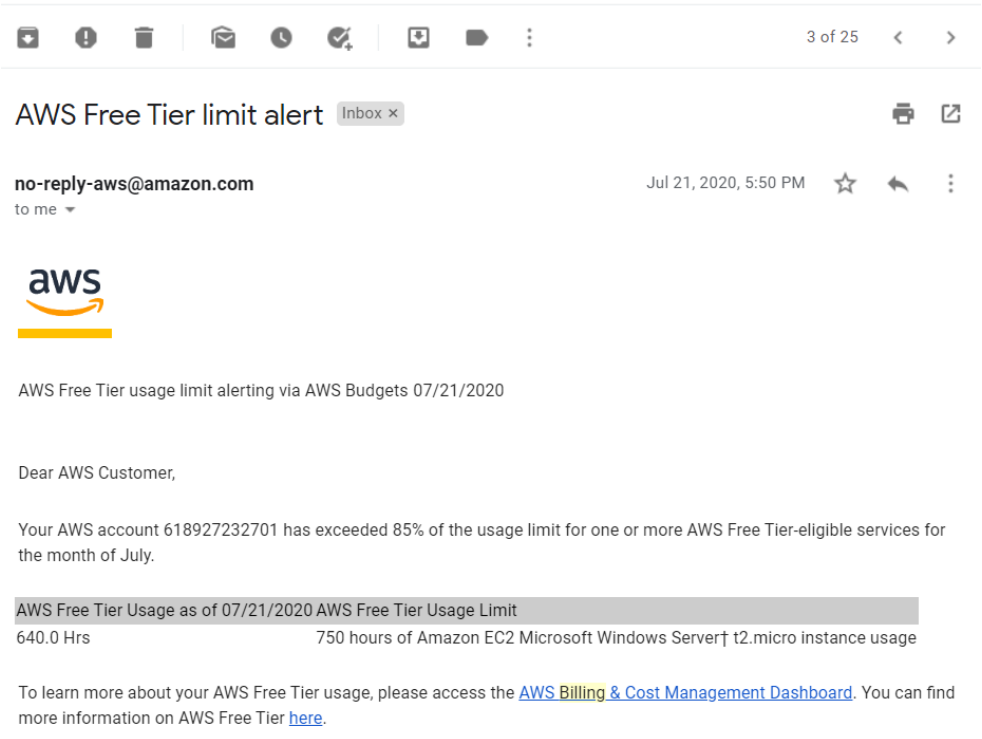
- A Dedicated Host is a physical EC2 server dedicated for your use.

AWS Reference: Amazon EC2 pricing

## Tracking your AWS Free Tier usage

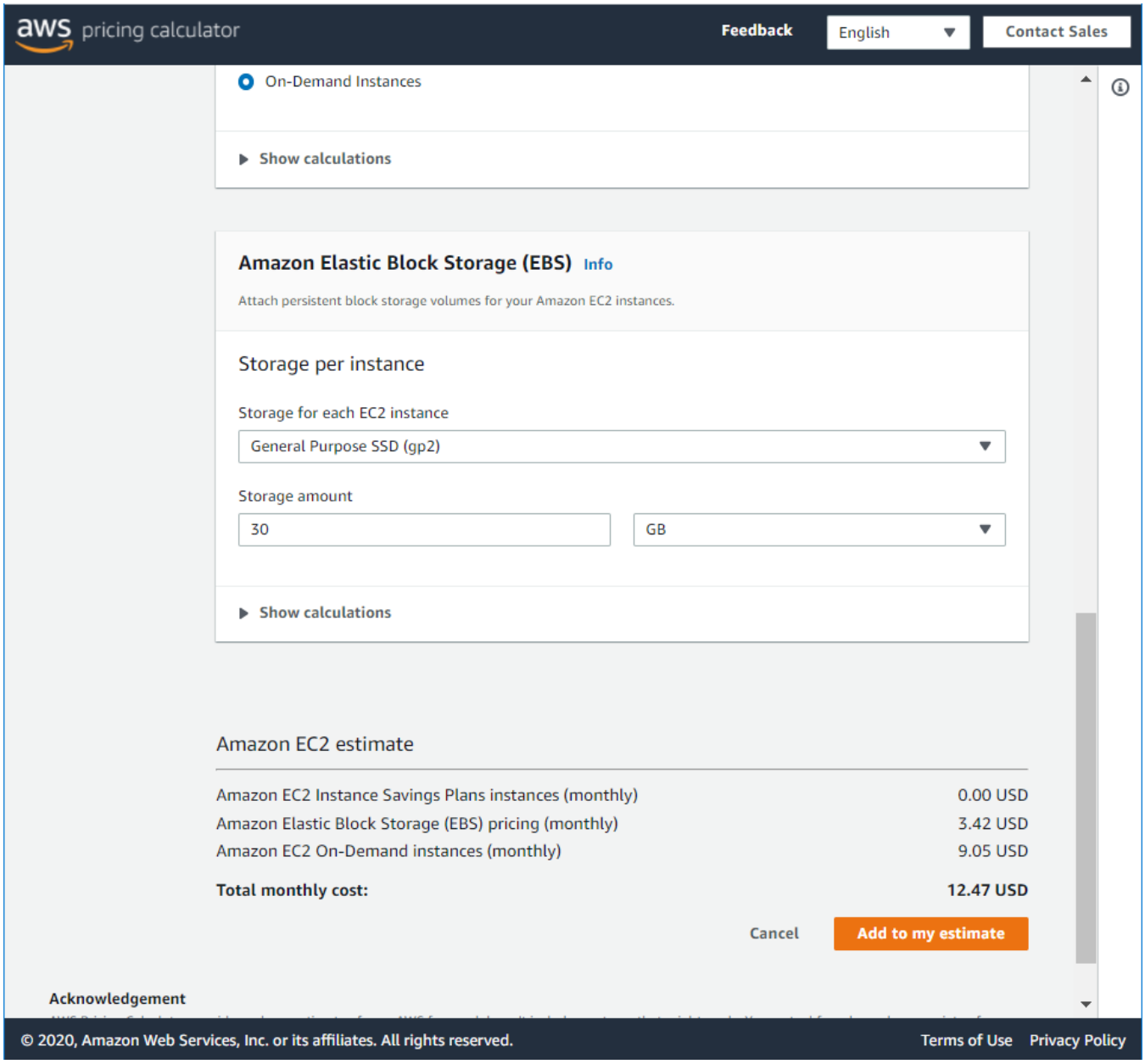
- You can track your AWS Free Tier usage to help you stay under the AWS Free Tier limits.
- AWS automatically provides alerts through AWS Budgets to notify you by email when you exceed 85 percent of your AWS Free Tier limits for each service.



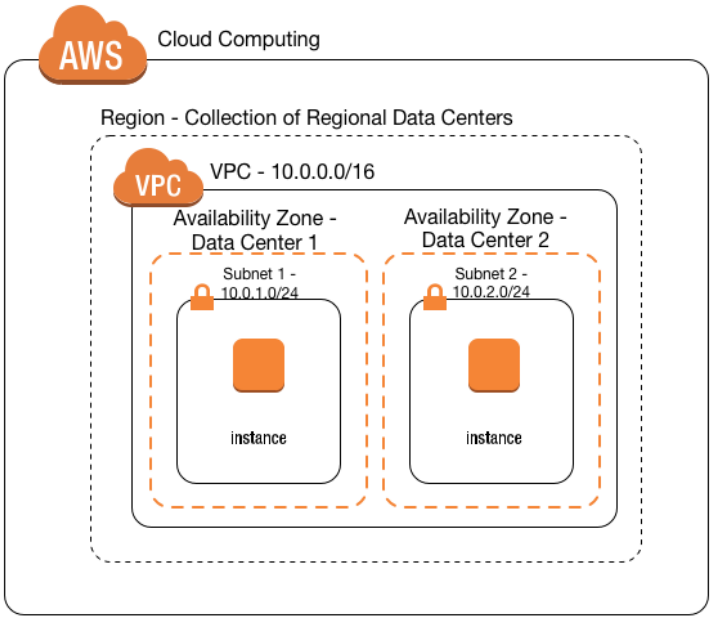
## AWS Pricing Calculator

- AWS Pricing Calculator gives estimate that fits your unique business or personal needs with AWS products and services.



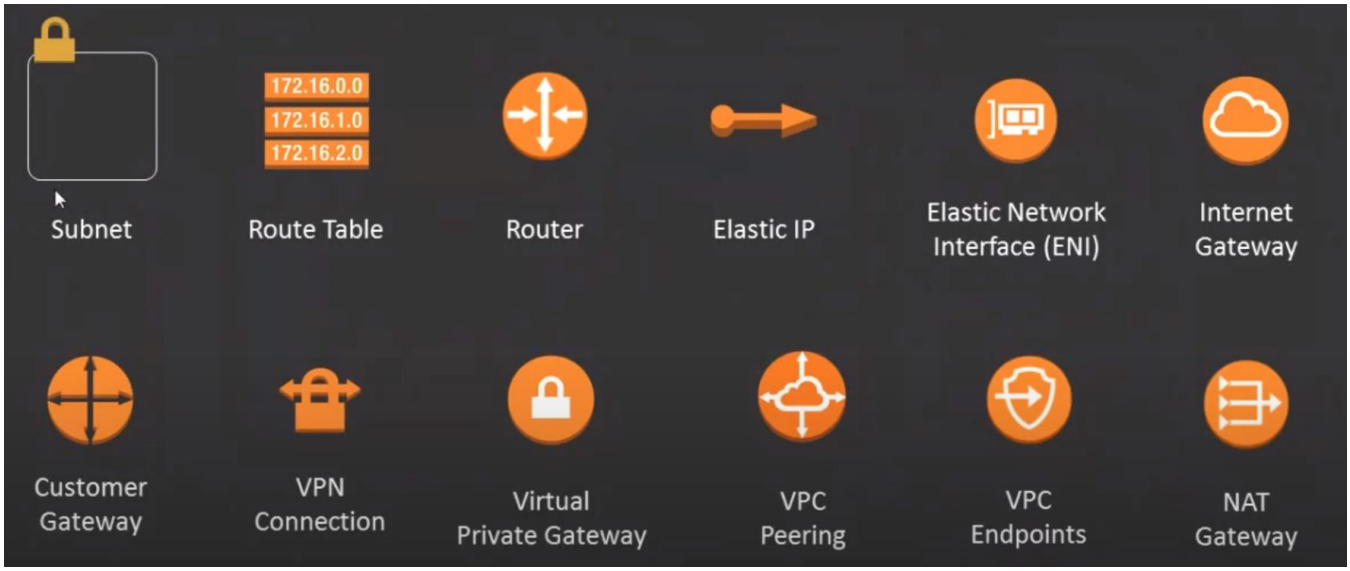
## AWS VPC (Virtual Private Cloud)


- Isolated user defined network inside AWS (like VLAN in switch)
- Logically isolated subnets
- Reserve IP Address ranges and assign to resources
- Setup Route Tables, Firewalls, Gateways
- Use Network ACL, IAM to control access




- Remotely connect on-prem DC network with VPN or Direct Connect
- Multiple VPCs can be peered across Region
- Subnet IPs are in Private range, once we assign Internet Gateway, we call it as Public Subnet
- VPC allocates a part of the cloud as your own network, we can control, manage IP, manage security, etc.
- Wen AWS launched initially the entire AWS cloud was a flat network.
- Forced to apply instance level security since the network is not managed / maintained by us
- We can connect to the VPC network and then access EC2s with private IPs
- We create Subnets in VPC so that we can provide access level
- Virtual Private Gateway (1.25Gbps) used to connect on-prem DC
- AWS has created Default VPCs in every Region of your account.

## Components of VPC (Overview)




- **Subnet**: We create Subnet inside VPC (later we can name it as Public or Private). VPC span across Region and Subnet spans within AZ.
- **Route Table**: Routing table holds information about where traffic to be sent. Router works based on the information in the route table. With in the VPC we use static routes. When it comes to VPN and Direct Connect, we go for Dynamic routing.
- **Router**: Router goes ahead and looks the route table and ask the route table that how do I go to the other system / device.
- **Elastic** IP: Static public IP, when we start EC2 in default VPC, dynamically we get public IP to the EC2 instance along with the private IP. If we restart EC2, the public IP may change. When we use Elastic IP, the public IP will be statically assigned to the EC2 until you release.
- **Elastic Network Interface (ENI)**: Virtual network card for EC2 instance. When we assign IP (Elastic or Dynamic), those IPs always get assigned to the ENI and the ENI gets assigned to EC2. We can always re-attach the ENI from one EC2 to another EC2 instance.
- **Internet Gateway (IGW)**: Gateway that allows you to connect to internet
- **Customer Gateway**: The router in on-prem data center for VPN
- **VPN Connection**: IPSec VPN connection between Customer Gateway and Virtual Private Gateway
- **Virtual Private Gateway**: Router in AWS side for VPN
- **VPC Peering**: To interconnect 2 VPCs together, 1st VPC sends a request and the other VPC accepts it.
- **VPC Endpoints**: Connectivity that allows you to talk to certain AWS cervices privately from VPC. For example, EC2 communicating to S3 go over via internet, VPC Endpoint gives direct connection to S3 privately. It doesn't go out of AWS network.
- **NAT Gateway**: Provides internet access to devices in private subnet of PVC. NAT Gateway sits in the Public subnet. Only outbound connection and response to that connections are allowed, no new inbound connections from internet.

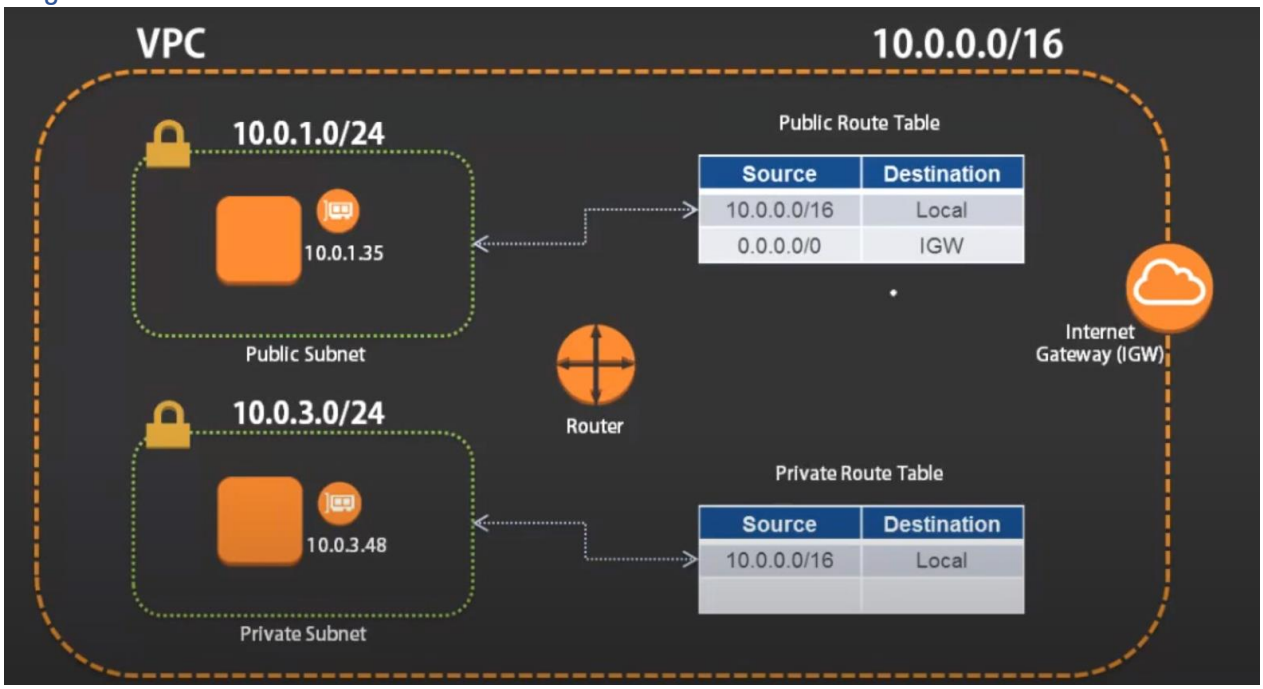
## IP Classless Inter-Domain Routing CIDR Block

- Notation that shows exactly how many IPs are there in a network, how large the network is, and additional information about the network. E.g. 10.0.0.0/16 (Network Prefix IP Addressing Method)
- Range: 10.0.0.0 to 10.0.255.255
- When we create VPC, we need to specify CIDER range of that VPC. We can add additional CIDER range to VPC, but existing one can't be deleted.
- 10.0.1.2/32 - Represents one IP using CIDER notation.
- Subnet calculator: http://jodies.de/ipcalc
- VPC capacity is between /28 (14 +2 IPs) to /16 (65534+2 IPs)
- We can bring our public IPs to AWS (Should talk to provider and AWS)
- **IP Address Reservations**:
    - .0 - Network Address
    - .1 - Reserved by AWS for the internal VPC Router
    - .2 - Reserved by AWS for the internal DNS Server
    - .3 - Reserved by AWS for future use
    - .255 - Broadcast Address. Broadcast is not supported with in VPC.

## Subnets


- Create subnets to divide the larger chunks of IPs
- Subnets are the Sub Networks of VPC, they have to fall in the IP range (CIDR) of VCP
- E.g.: CIDR 10.0.0.0/16
    - Subnet 1: 10.0.1.0/24 (10.0.1.0 to 10.0.1.255)
    - Subnet 2: 10.0.2.0/24 (10.0.2.0 to 10.0.2.255)
    - … … …
    - Subnet 255: 10.0.255.0/24 (10.0.1.0 to 10.0.255.255)

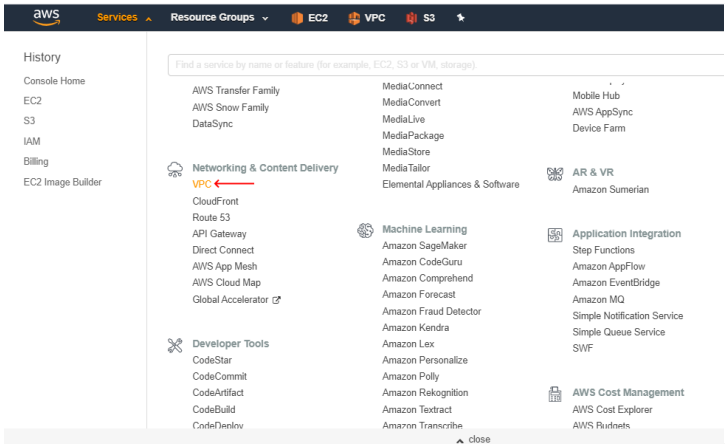
## Routing in VPC



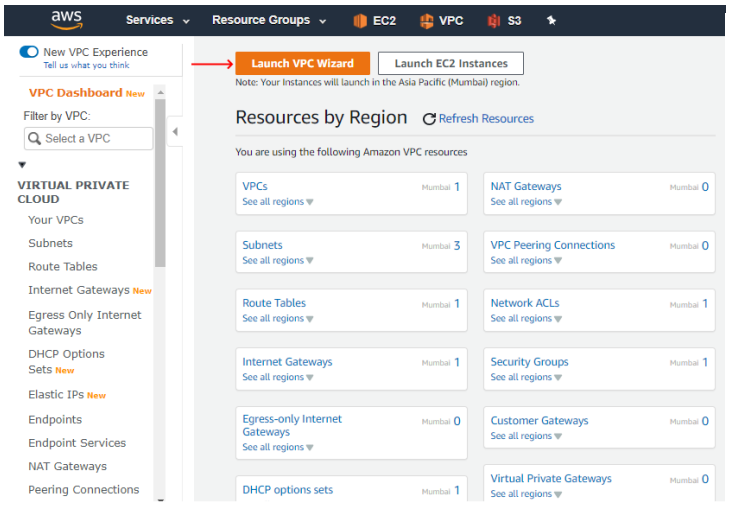

- When we create Subnets in VPC, it is just a smaller chunk of VPC CIDR, there is nothing called Public or Private.
- The moment we associate Route Table on subnet, based on the Route Table information, we call the subnet as Public or Private.
- Route Table is the one that decides whether the subnet is Public or Private.
- Apart from the Public Route Table, your EC2 instance should have a Dynamic Public IP or Elastic IP

# [LAB] VPC Configurations

## Step 1: Creating VPC

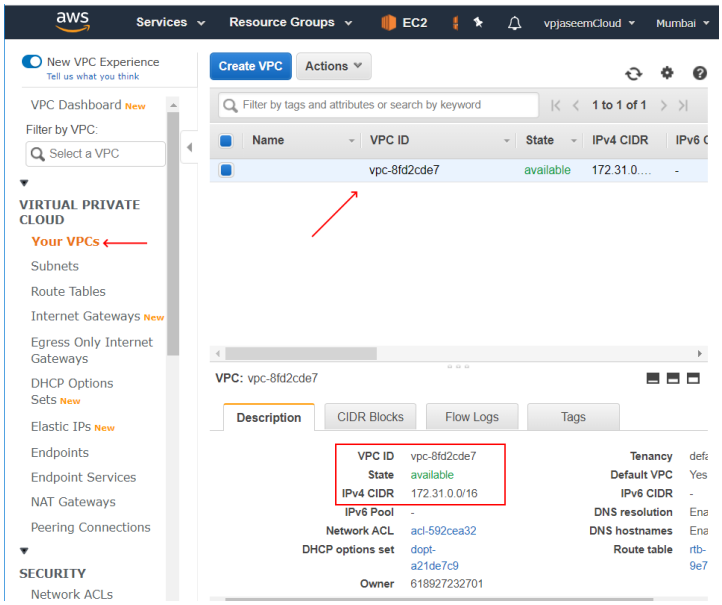



- Open AWS Console >> Services >> VPC

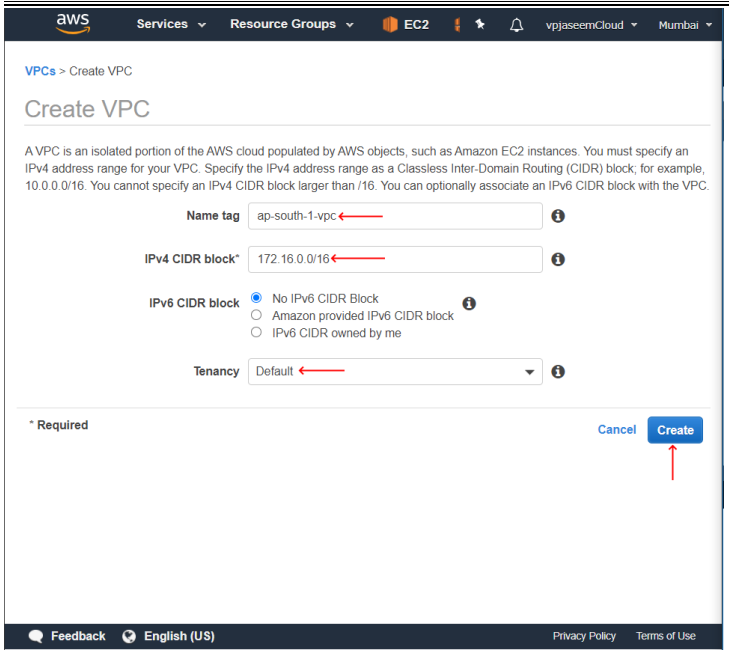



(Optional)

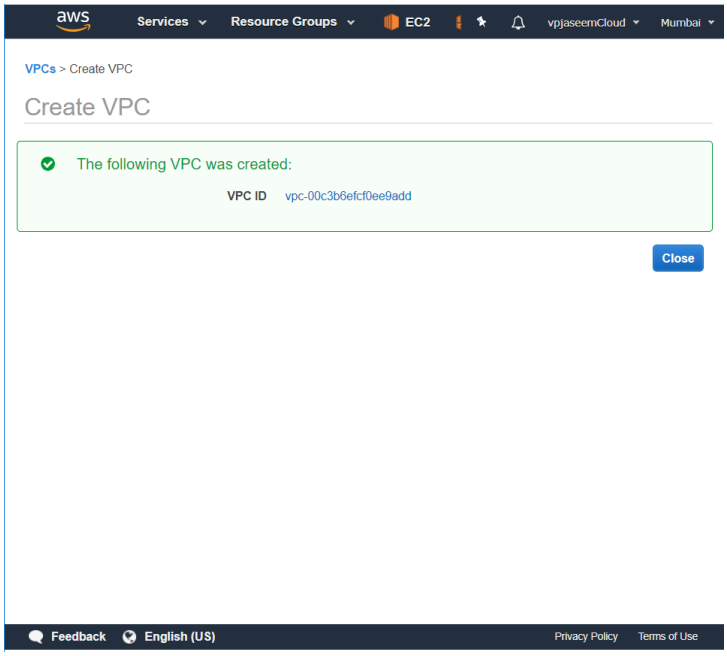
- We have an option to use Launch VPC Wizard to create a VPC, how ever that is too easy and you might not understand what's happening behind the scene.



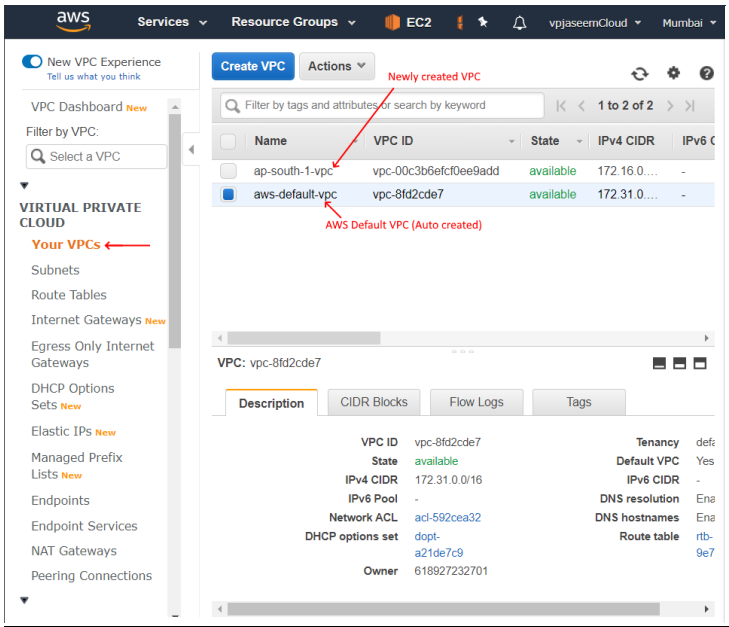

- Go to Your VPCs, you can see the default VPC created by AWS here. Every region will have default VPCs created.
- We can also see the IP CIDR block as well
- Go ahead and hit Create VPC button




- Go to Your VPCs, you can see the default VPC created by AWS here. Every region will have default VPCs created.
- We can also see the IP CIDR block as well
- Go ahead and hit Create VPC button




- Now we have successfully created VPC

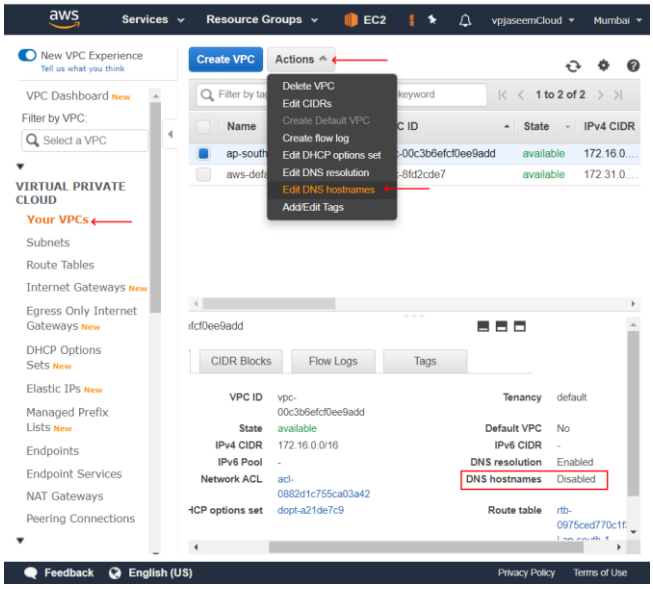


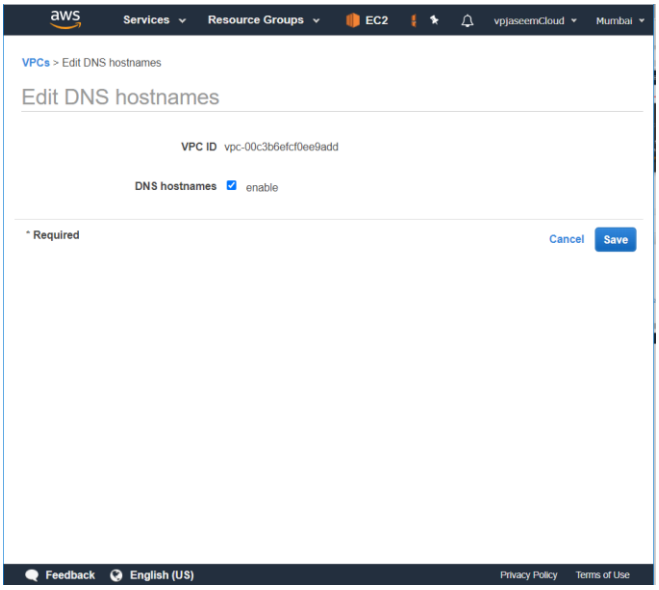

- In the VPC services page, you can see the newly created VPC and AWS Default VPC.

Note: There won't be any name tag for default VPC, it is recommended to provide a name tag for the default VPC.

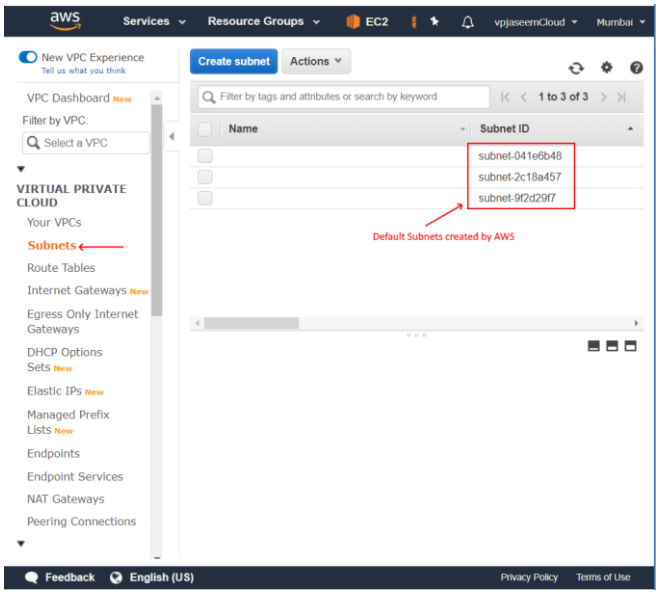
## Step 2: Enable Public DNS Hostnames

- When you create VPC, it doesn't give any Public DNS hostnames to EC2 instances by default.
- To enable this, select the VPC >> Actions >> Edit DNS Hostnames >> Yes >> Save



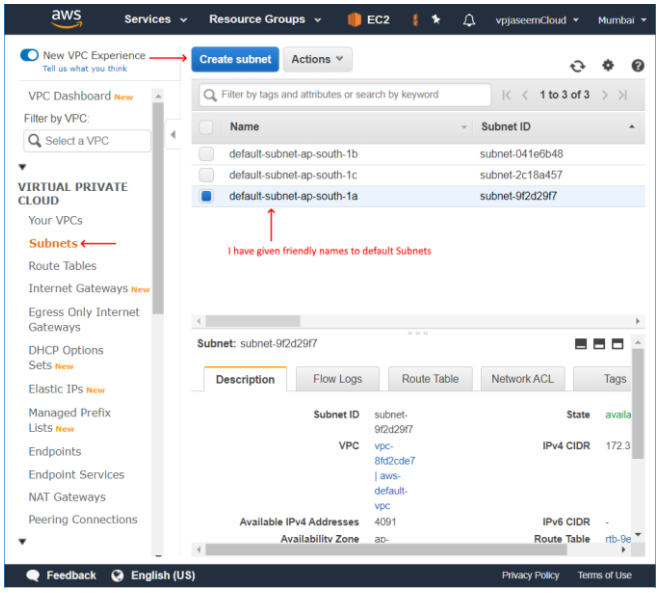



## Step 3: Creating Subnets

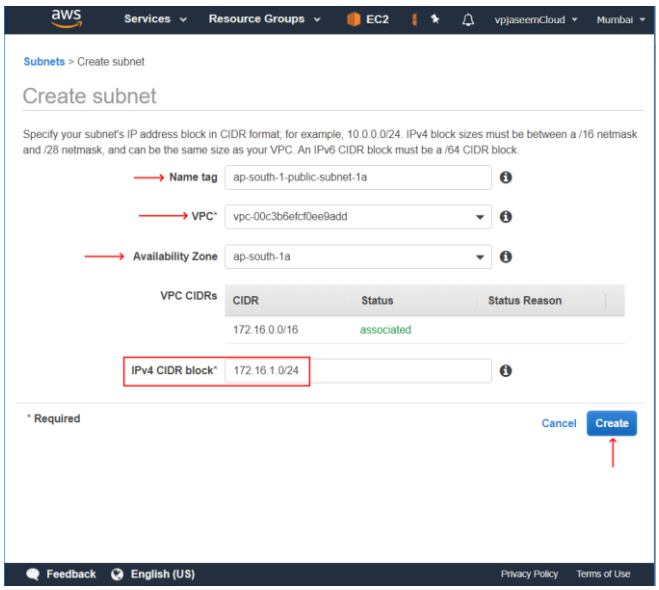



- Go to the Subnet options in VPC Dashboard >> You can observe the default Subnets created by AWS itself.

Note: It is recommended to provide a friendly name to the default subnets.



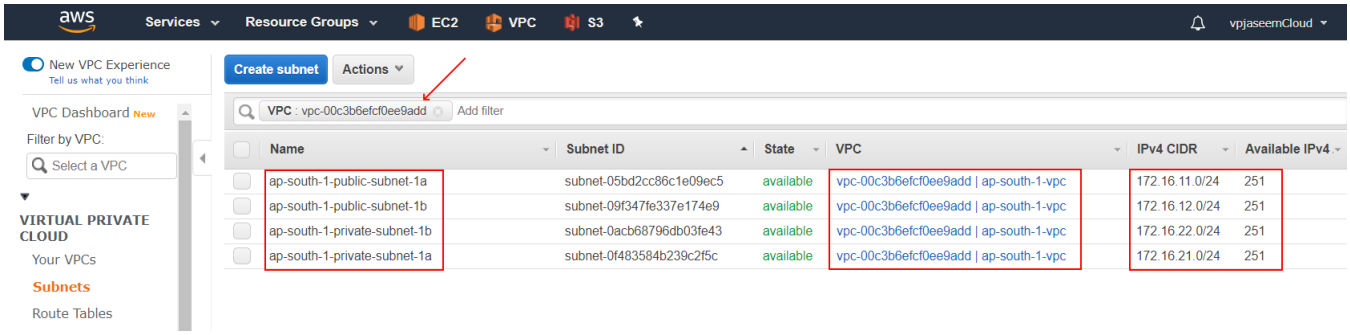

- I have given some friendly names to the default Subnets.
- Go ahead and click Create Subnet button
- In our lab practice we are creating 4 subnets (2 Private in 2 AZs and 2 Public in 2 AZs)
    1. ap-south-1-public-subnet-1a
    2. ap-south-1-public-subnet-1b
    3. ap-south-1-private-subnet-1a
    4. ap-south-1-private-subnet-1b




I have configured below subnets,

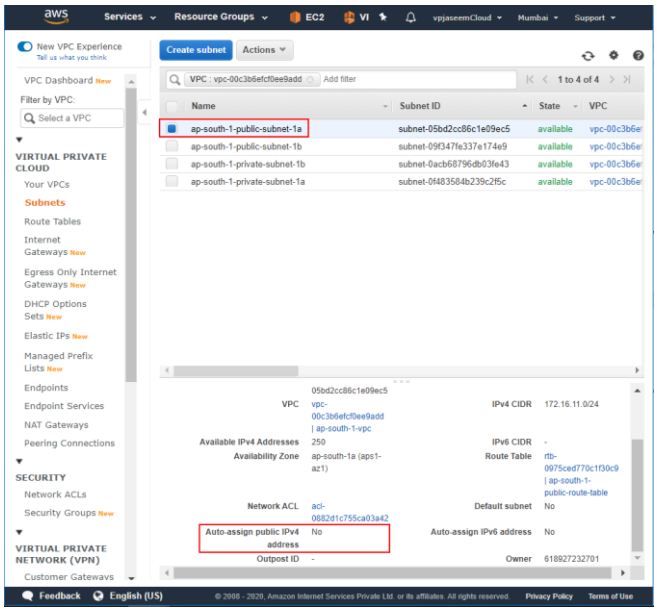
| SUBNET | CIDR |
|---|---|
| ap-south-1-public-subnet-1a | 172.16.11.0/24 |
| ap-south-1-public-subnet-1b | 172.16.12.0/24 |
| ap-south-1-private-subnet-1a | 172.16.21.0/24 |
| ap-south-1-private-subnet-1b | 172.16.22.0/24 |

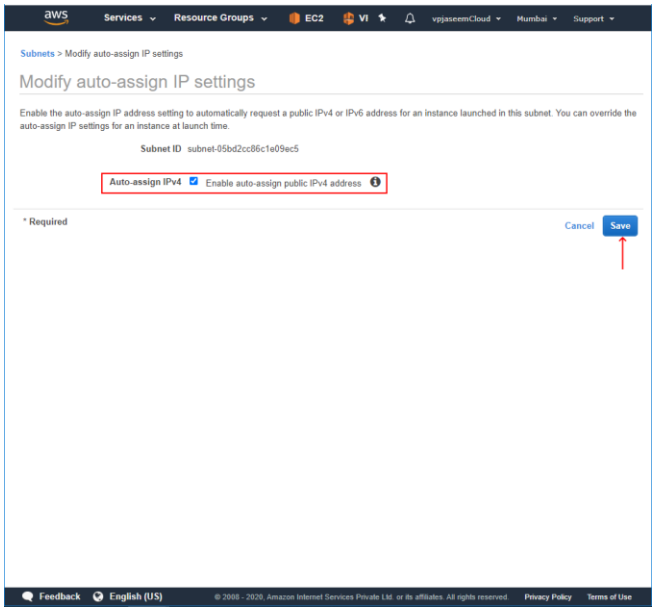
- We can apply a filter in the Subnets and see all the Subnets that we created.

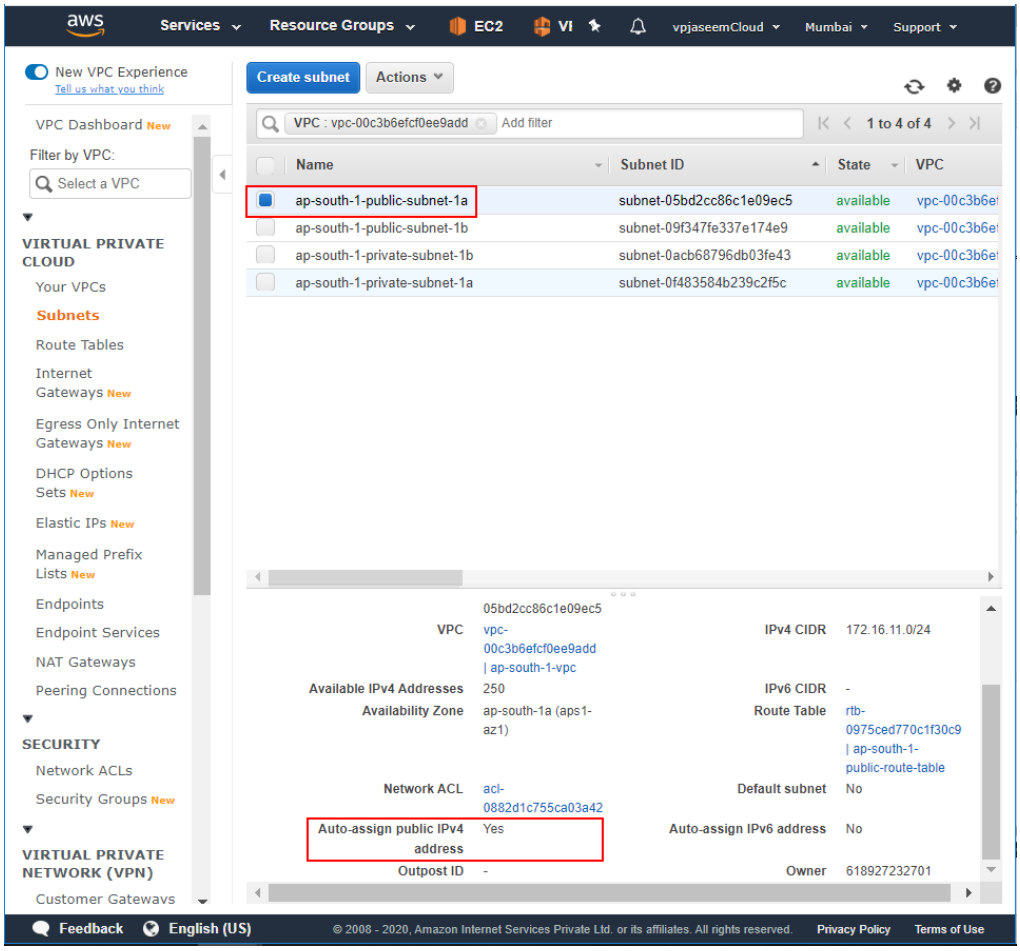


## Step 4: Enable Auto Assign Public IP for Public Subnet

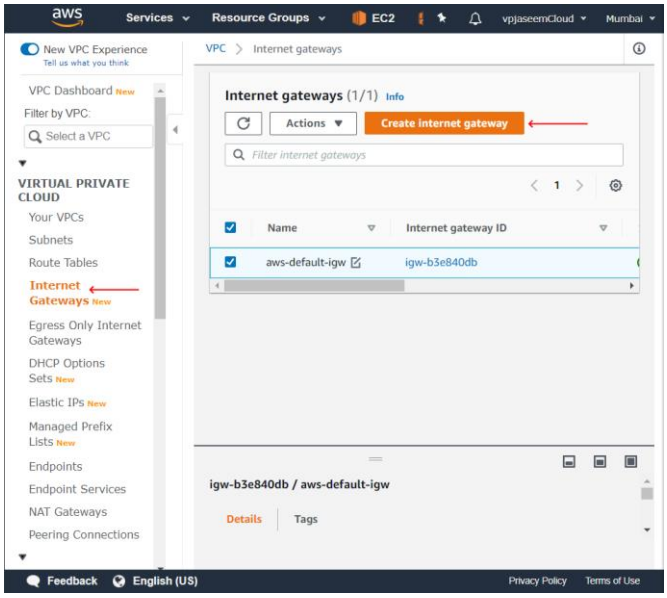
- Now we have Public and Private Subnets, lets enable Auto Assign Public IP settings on the two Public subnets.
- Select the Public Subnet >> Actions >> Modify auto-assign IP settings >> Enable auto-assign public IPv4 address.
- EC2 instances on the Public subnet will be automatically assigned with Public IP.







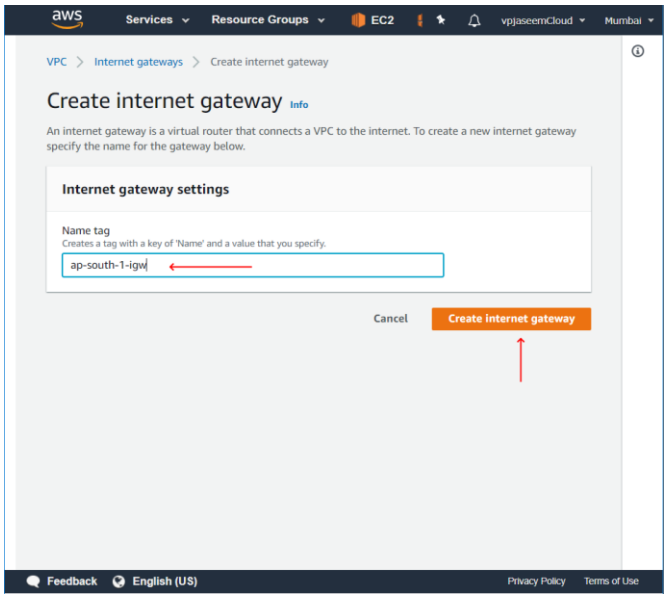
## Step 5: Add Internet Gateway and Associate to Public Subnet



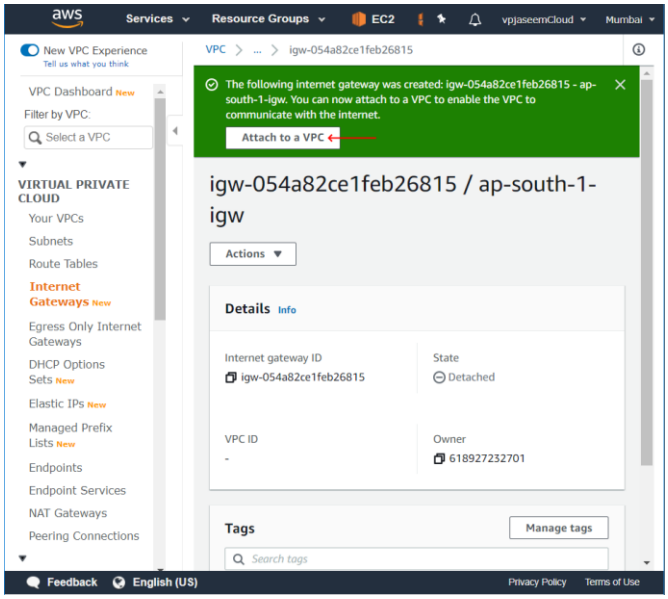

- Go to Internet Gateways >> Create internet gateway

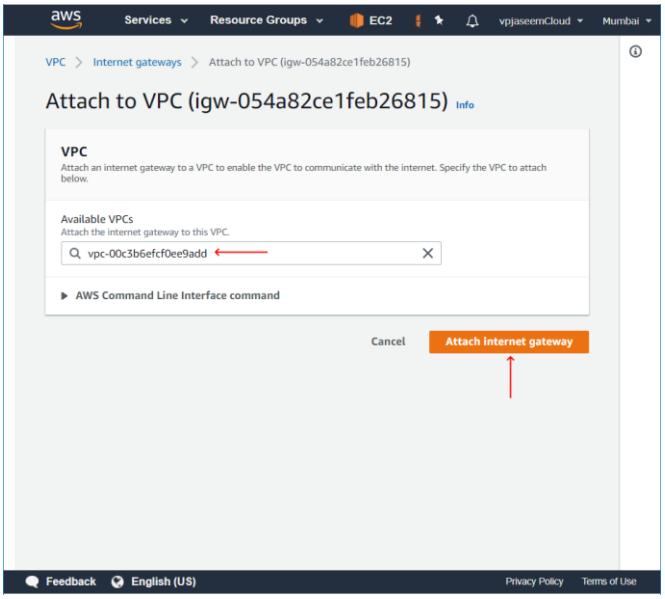
Note: You can see the default Internet Gateway here.

- This will provide connectivity to internet from VPC



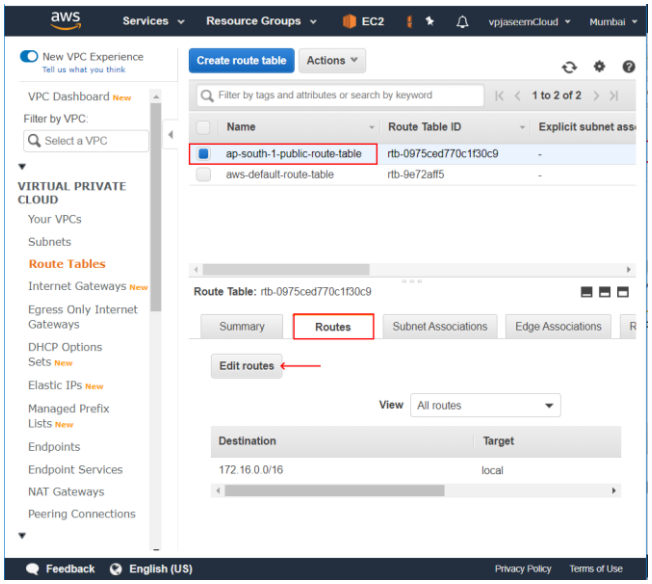

- Provide a Name tag >> Create internet gateway



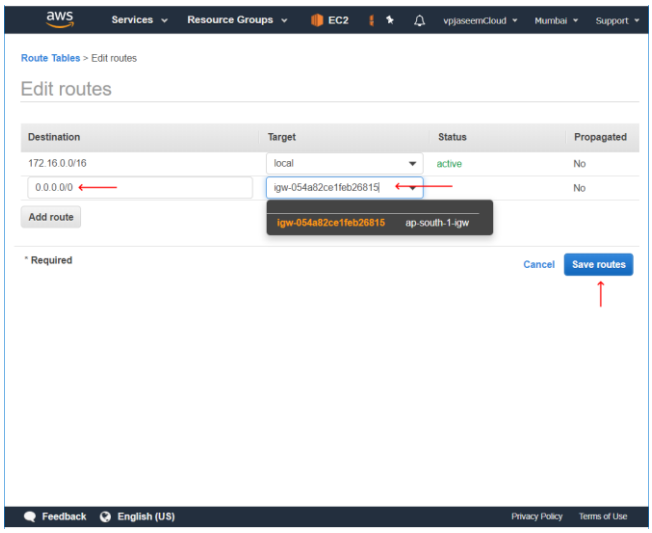



## Step 6: Route Table Configuration

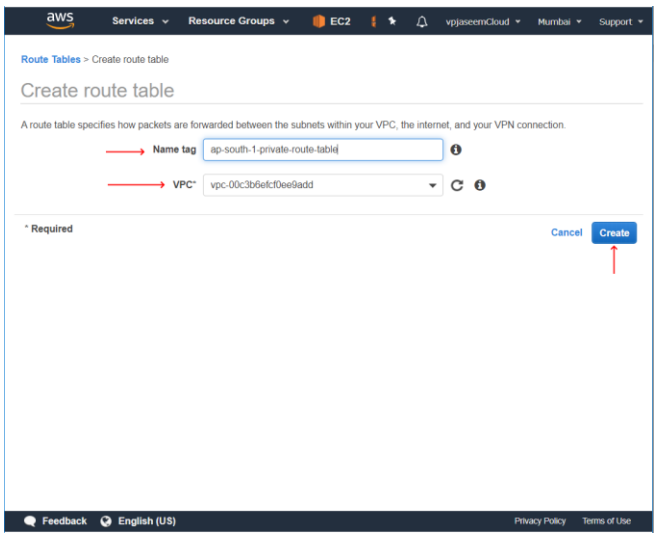
- For every VPC we create, AWS will add a specific Route Table for that VPC. You can see Main = Yes in those Route tables.
- We basically need 2 Route Tables now, either you can create 2 new or edit existing one and add another one. (ap-south-1-public-route-table, ap-south-1-private-route-table)




- Rename the Route Table that was created automatically for ap-south-1-vpc
- We are going to attach the Internet Gateway app-south-1-igw to this route table. Now this route table becomes public route table.
- Edit routes >>




- Add an internet route (0.0.0.0/0) and point to the app-south-1-igw

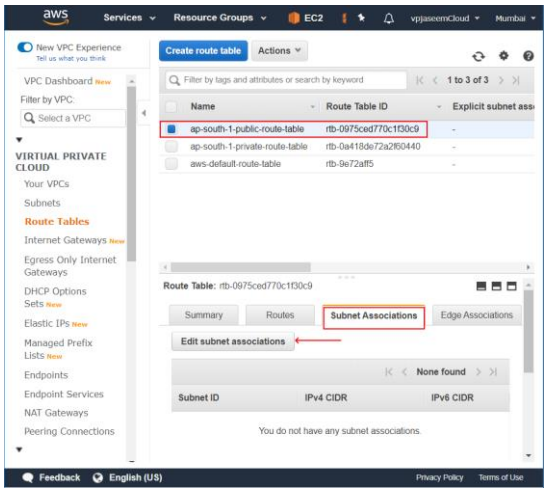



- Now add one more Route Table ap-south-1-private-route-table
- Do not associate Internet gateway to this route table so that this becomes always private.

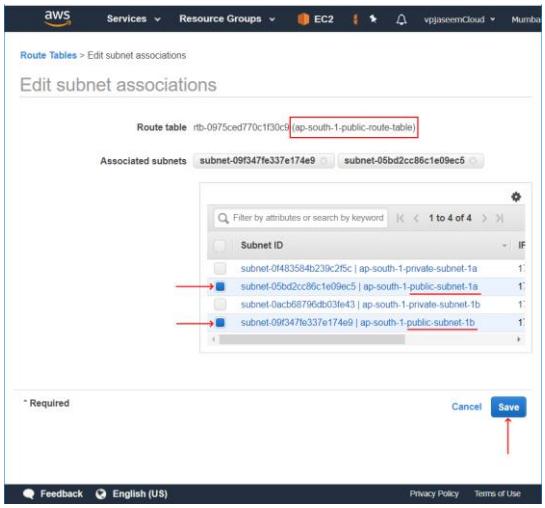
- Let's try understand how one is Public and other one is Private with below table.

| ap-south-1-public-route-table | | ap-south-1-private-route-table | |
|---|---|---|---|
| **Destination** | **Target** | **Destination** | **Target** |
| 172.16.0.0/16 | Local | 172.16.0.0/16 | Local |
| 0.0.0.0/0 | igw-054a82ce1feb26815 | - | - |

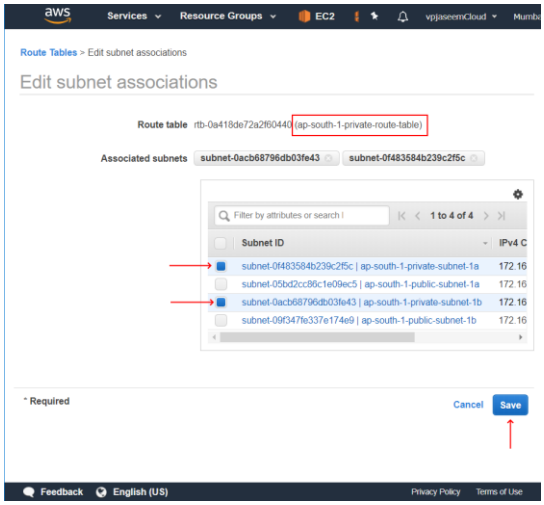
### Step 7: Associate Route Table with Subnets




- Select the ap-south-1-public-route-table >> Subnet Associations >> Edit subnet associations




- Associate ap-south-1-public-subnet-1a and ap-south-1-public-subnet-1b to the ap-south-1-public-route-table
- Now this particular subnet is actually a Public Subnet



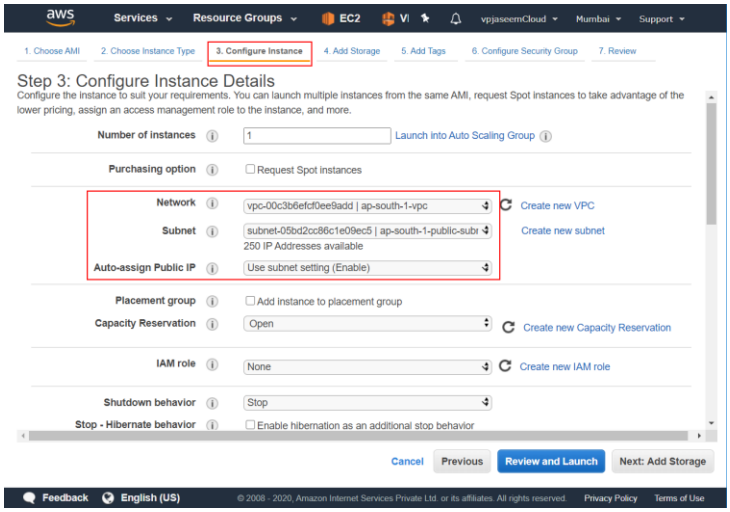

- Similarly, associate ap-south-1-private-subnet-1a and ap-south-1-private-subnet-1b to the ap-south-1-private-route-table
- Now this particular subnet is actually a Private Subnet

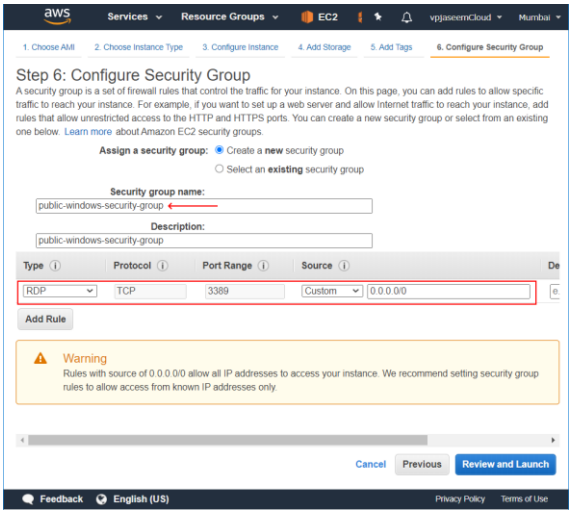
- Let's try understand how the entire routing has been configured for the VPC using below table.

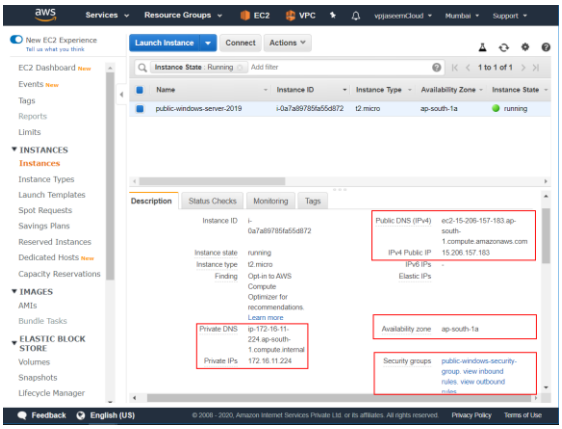
| TYPE | SUBNET | CIDR | ROUTE TABLE | ROUTES | |
|---|---|---|---|---|---|
| | | | | Destination | Target |
| Public subnet | **ap-south-1-public-subnet-1a** | 172.16.11.0/24 | ap-south-1-public-route-table | 172.16.0.0/16 | Local |
| | **ap-south-1-public-subnet-1b** | 172.16.12.0/24 | | 0.0.0.0/0 | igw-054a82ce1feb26815 |
| Private subnet | **ap-south-1-private-subnet-1a** | 172.16.21.0/24 | ap-south-1-private-route-table | 172.16.0.0/16 | Local |
| | **ap-south-1-private-subnet-1b** | 172.16.22.0/24 | | - | - |

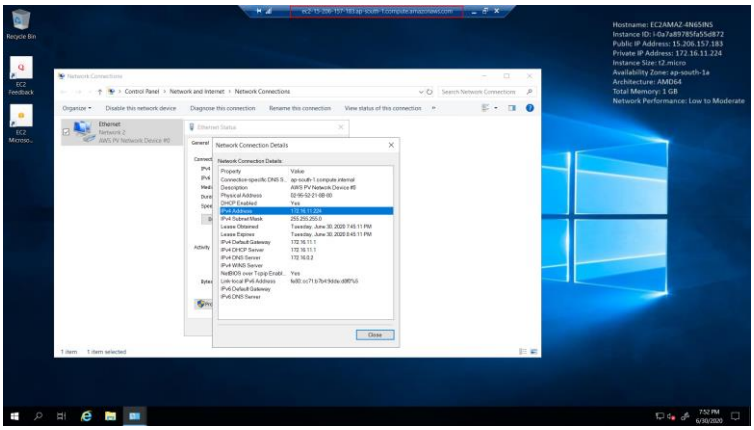
## Step 8: Launch EC2 Instances in the VPC

- Now launch an EC2 instance on your VPC with in Public Subnet, make sure you have a security group that allows access the EC2 from internet. Try to access the EC2 instance

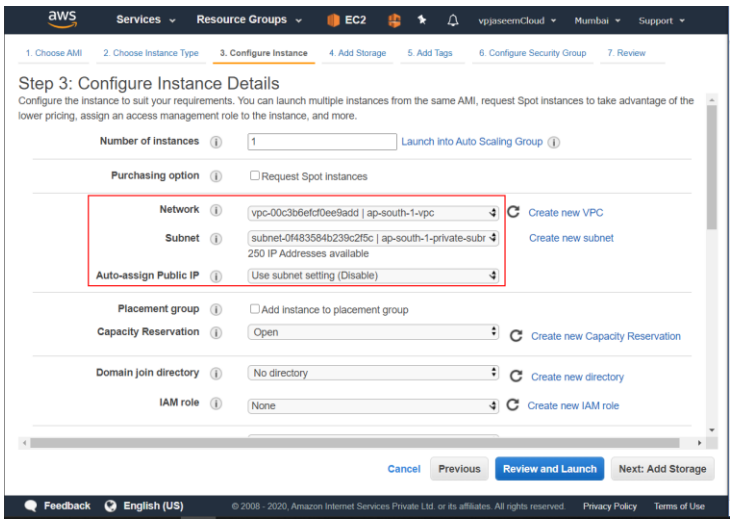


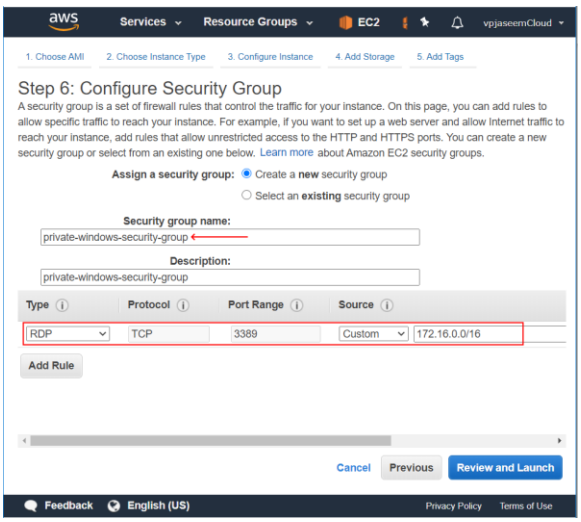



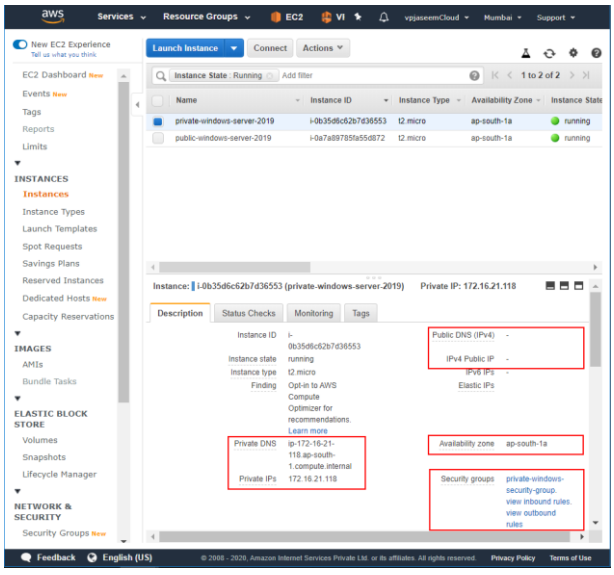



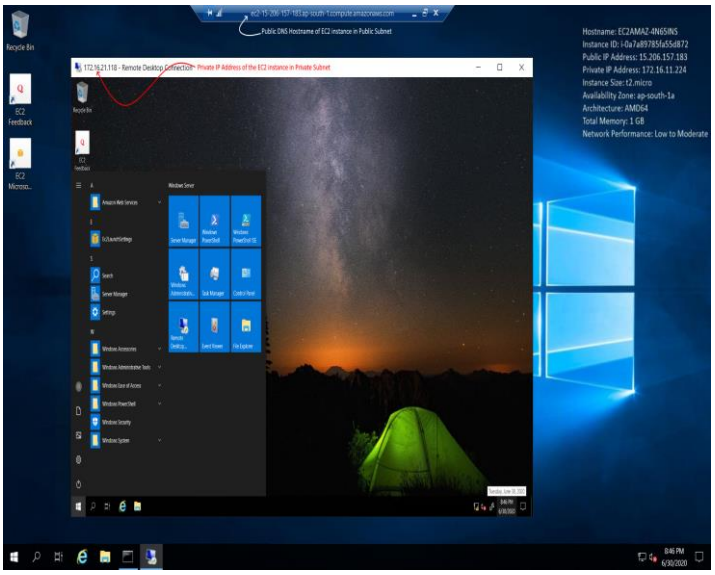



- Now launch an EC2 instance on your VPC with in Private Subnet, you won't get any Public IP top access it.
- Even if you enable auto assign public IP, and allow RDP access to all network, still the instance won't be accessible since it is in private subnet (without an internet gateway).
- The only way to access the EC2 instance is via another EC2 that is in Public subnet and then RDP to private subnet EC2 via private IP address.

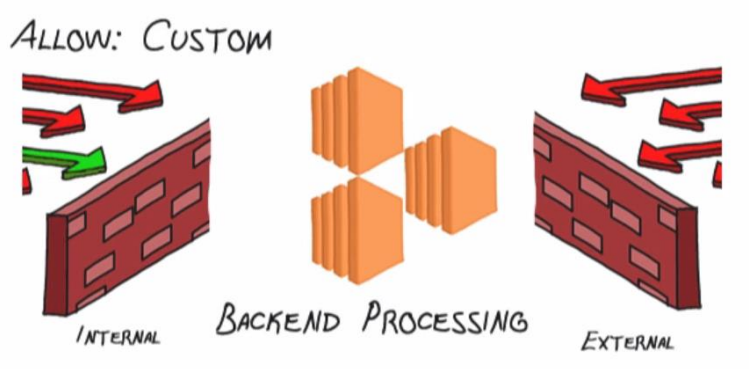


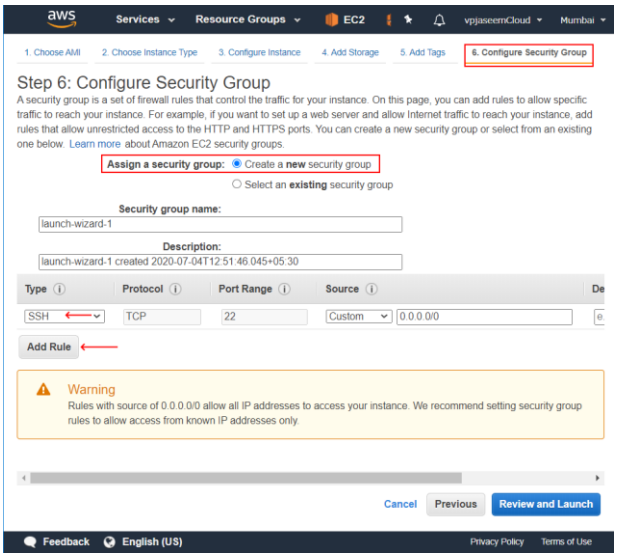
## Security Groups

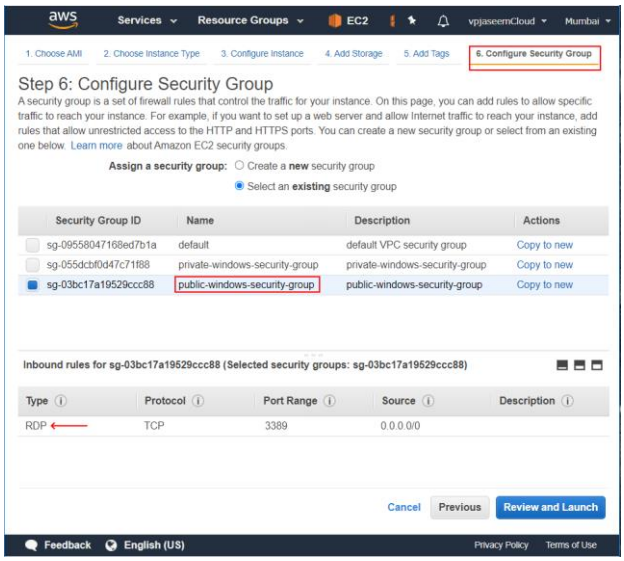



- Virtual firewalls for EC2 instances applied at instance NIC level.
- Created under VPC and applied at EC2
- Block / allow inbound and outbound traffic for an EC2 instances.
- One Security Group can be assigned to multiple EC2 instances, but one EC2 will have only one security group.
- Some other services (RDS, Redshift) also uses Security Groups to protect network attack.
- Each of the EC2 instances are protected by Security Group
- It is only protecting the individual resources assigned to an EC2
- If 2 instances are communicating in same Security group, traffic from one E2 goes out to the security group and enters to the same security group, then other EC2.
- Default is Allow all outbound and Deny all inbound.
- Response to the Security Group accepted automatically, no need to specify inbound rule for accepting response traffic.

## [LAB] Create or Select Security Group while Launching EC2

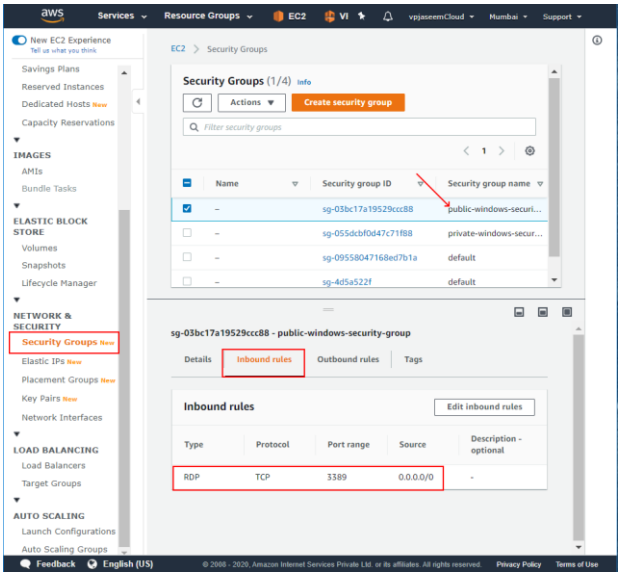
- You might have already seen, the way of assigning Security Group during the creation of EC2 instance.
- You can either cerate a new security group and define allow rules or select existing
- At this time we only add inbound rules, all outbound traffics allowed default.

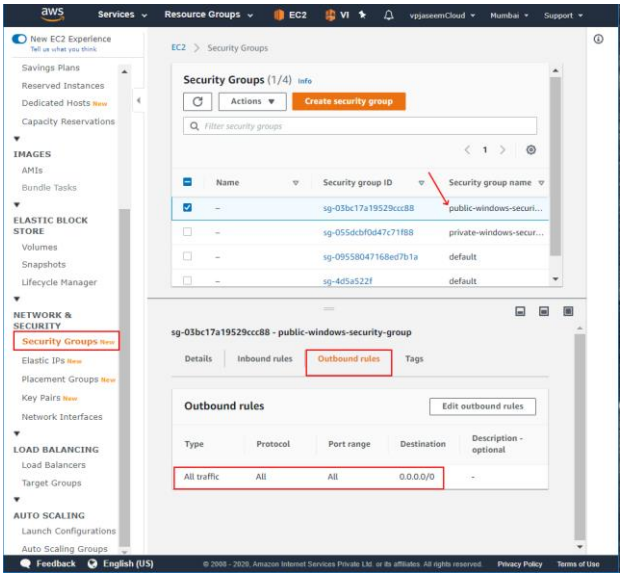


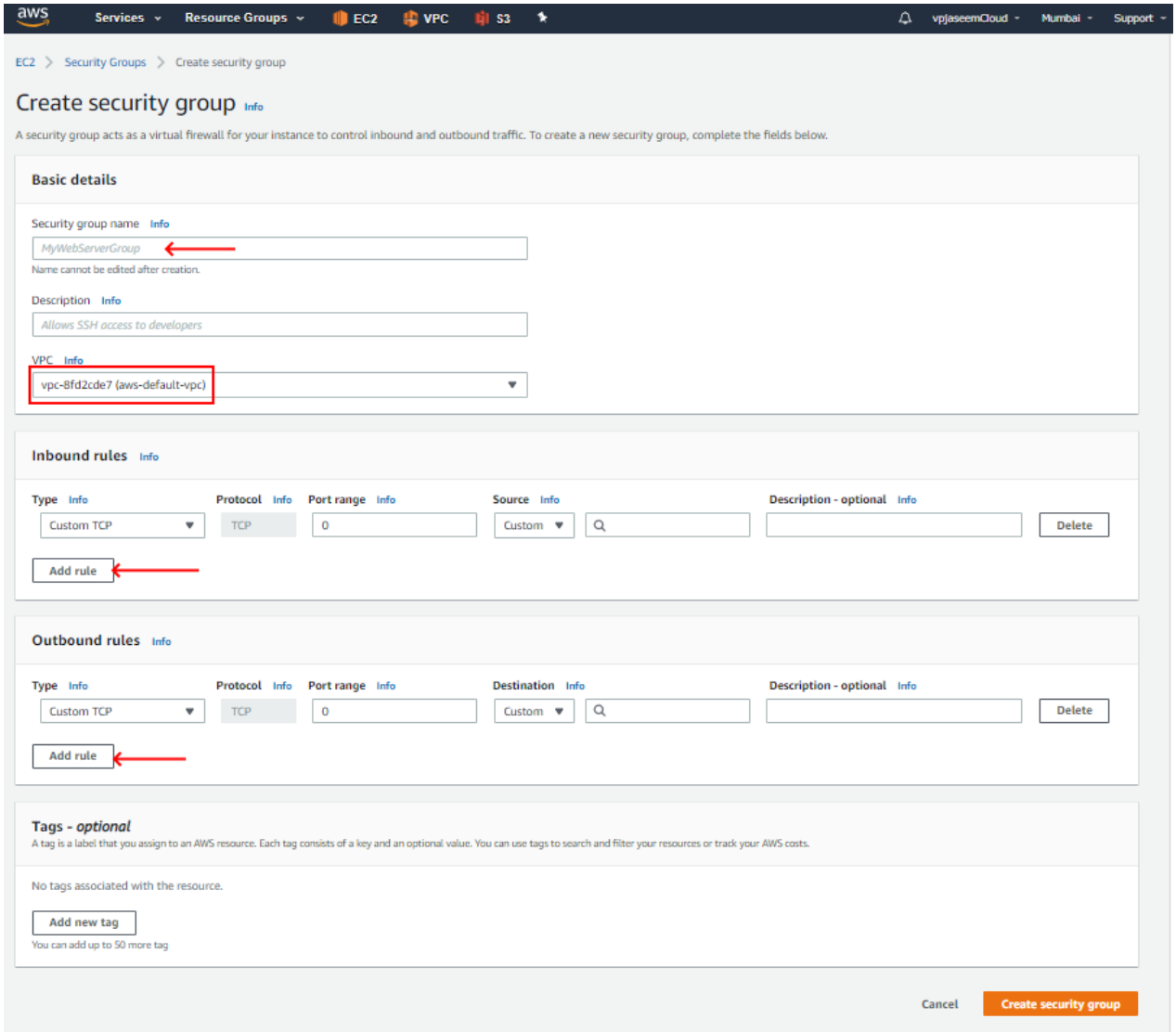
## [LAB] Modify Existing Security Group

- Modify inbound and outbound rules of Security group is available in EC2 Dashboard >> Security Groups



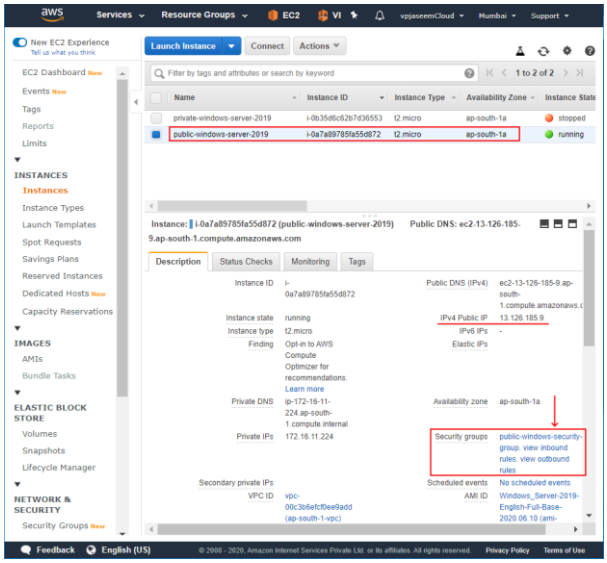




- You can also create new security group from this dashboard.

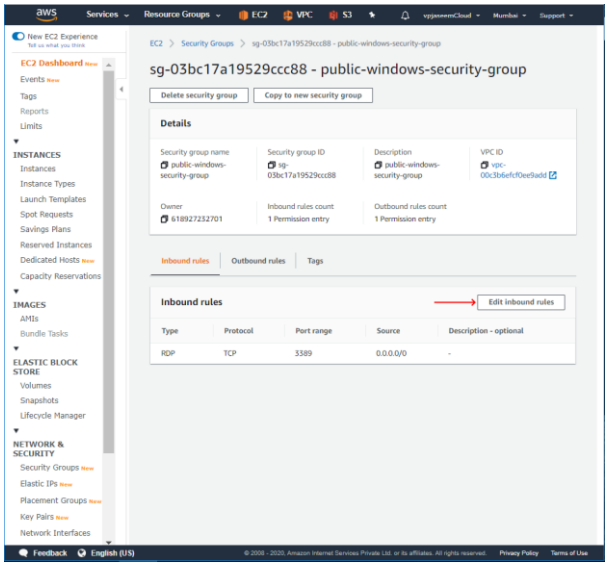


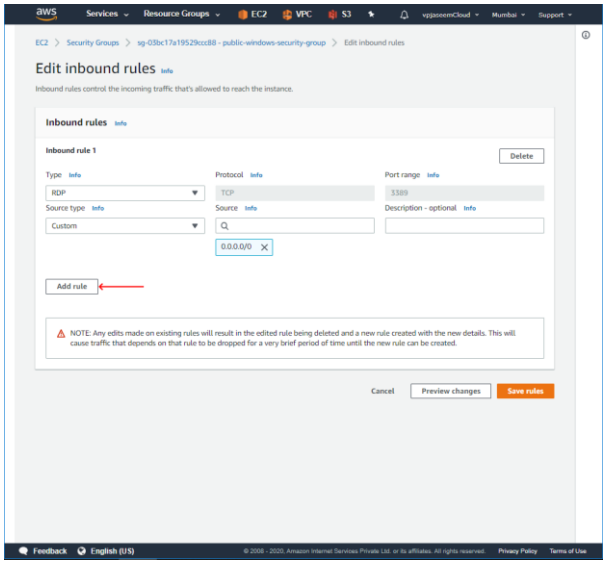
## [LAB] Allow ICMP from Internet to a Windows EC2 Instance

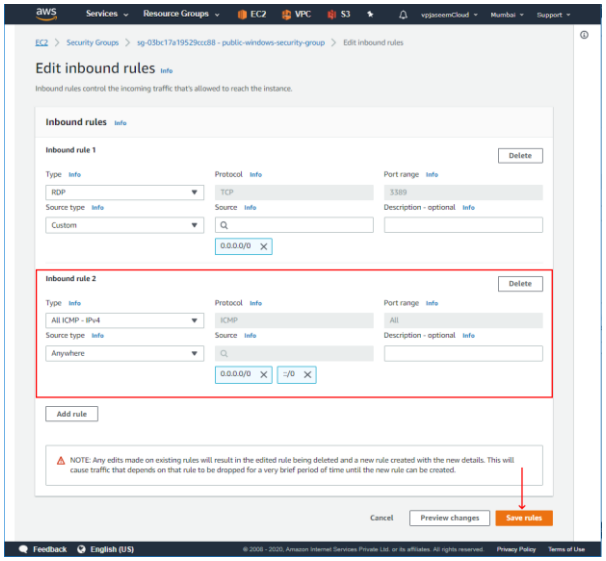
- When you spin up an EC2 in public subnet, the default Security Group blocks the ICMP traffic to the instance.
- In this lab, we modify the existing security group to allow ICMP (ping) traffic.

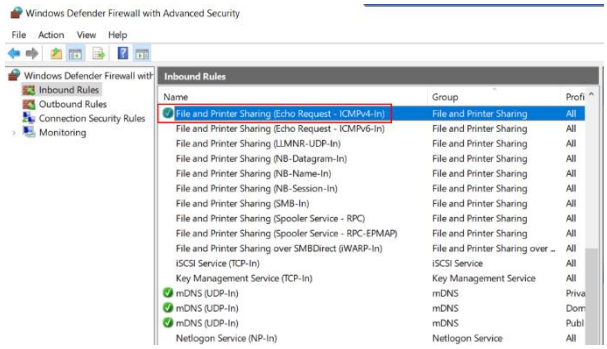


- You can either select the Security group from the EC2 instance or navigate to Security Group from the EC2 dashboard.
- Select the Security Groups >> Inbound Rules >> Edit inbound rules >> Add Rule (Inbound rule 2 will be crated)

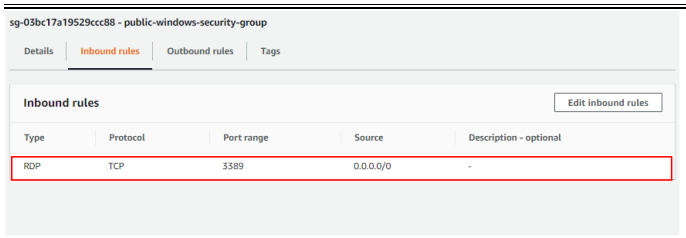


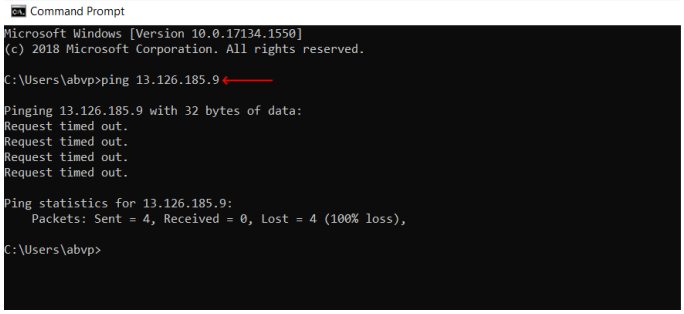


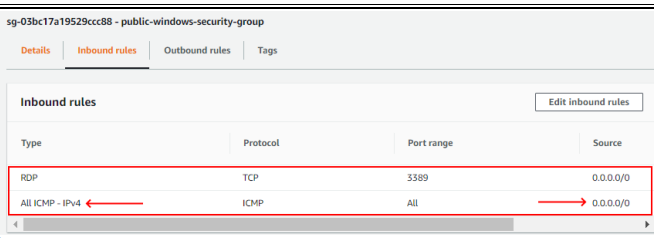


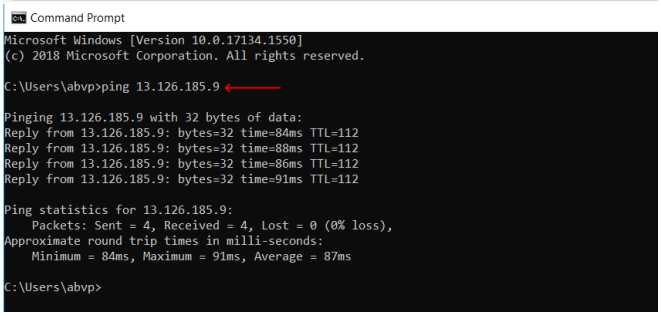


Since we are working on a Windows EC2, make sure either disable Windows native firewall or enable ICMP.


sg-03bc17a19529ccc88 - public-windows-security-group
Details
Inbound rules
Outbound rules
Tags
Inbound rules
Edit inbound rules
Type
Protocol
Port range
Source
Description - optional
RDP
TCP
3389
0.0.0.0/0
-

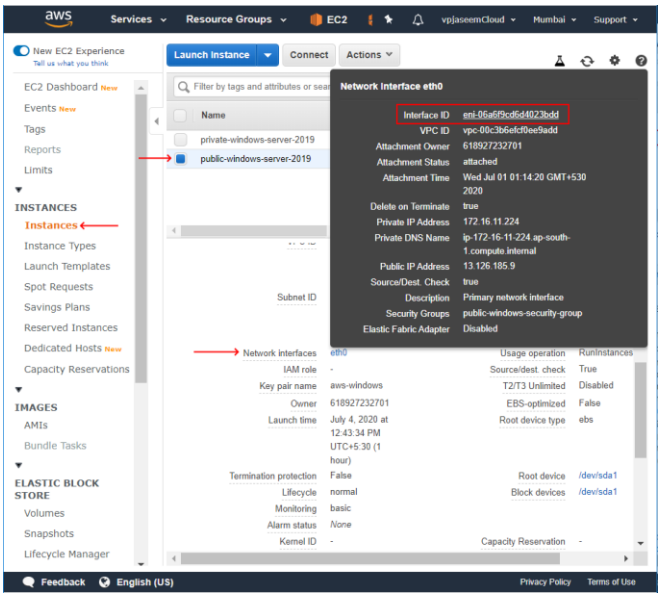

Command Prompt
Microsoft Windows [Version 10.0.17134.1550]
(c) 2018 Microsoft Corporation. All rights reserved.
C:\Users\abvp>ping 13.126.185.9
Pinging 13.126.185.9 with 32 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.
Ping statistics for 13.126.185.9:
Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
C:\Users\abvp>

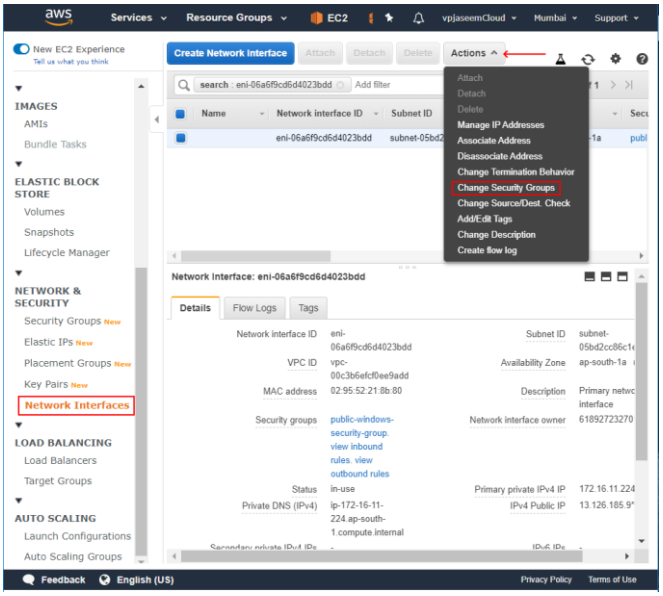

sg-03bc17a19529ccc88 - public-windows-security-group
Details
Inbound rules
Outbound rules
Tags
Inbound rules
Edit inbound rules
Type
Protocol
Port range
Source
RDP
TCP
3389
0.0.0.0/0
All ICMP - IPv4
ICMP
All
0.0.0.0/0

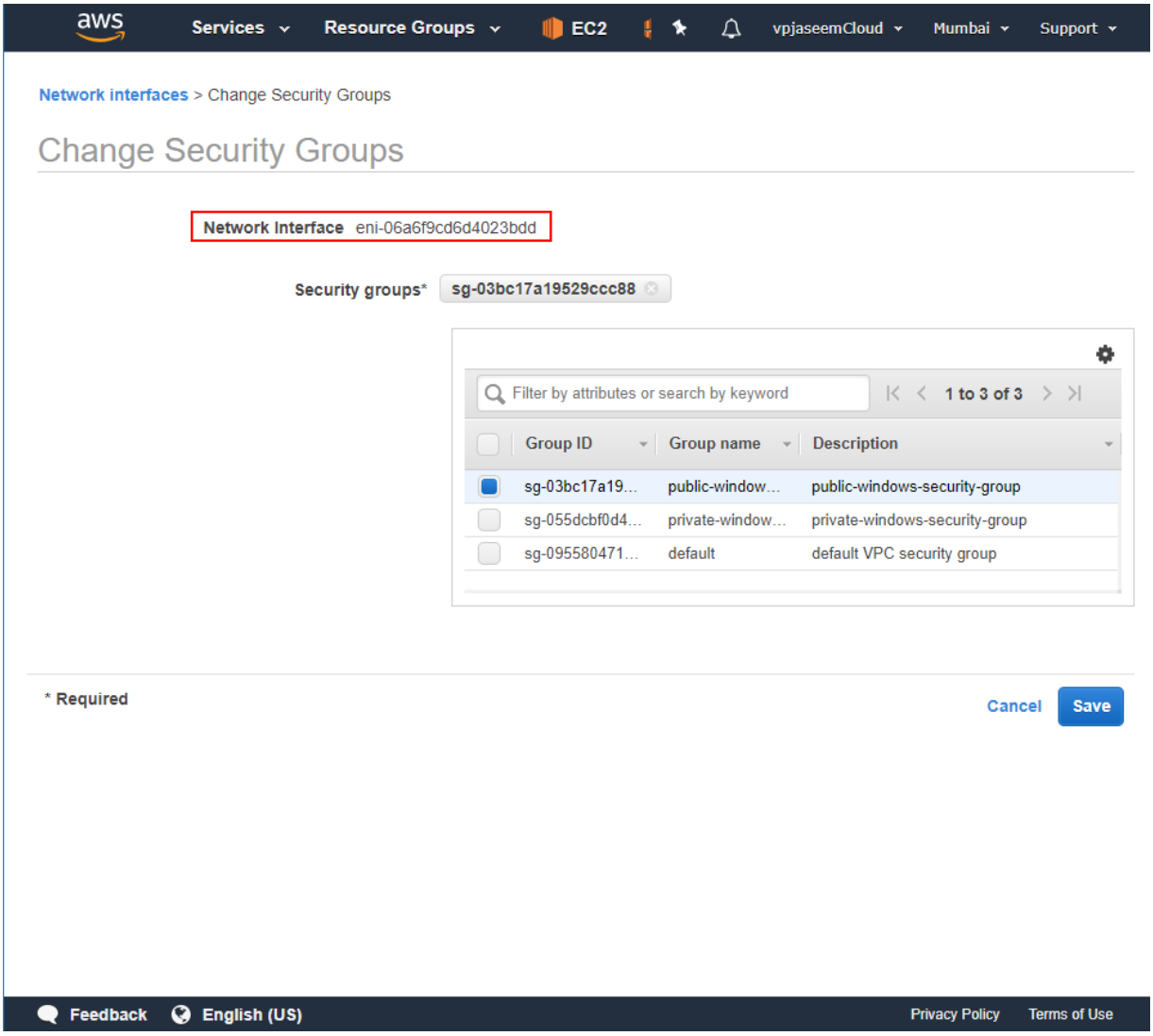

Command Prompt
Microsoft Windows [Version 10.0.17134.1550]
(c) 2018 Microsoft Corporation. All rights reserved.
C:\Users\abvp>ping 13.126.185.9
Pinging 13.126.185.9 with 32 bytes of data:
Reply from 13.126.185.9: bytes=32 time=84ms TTL=112
Reply from 13.126.185.9: bytes=32 time=88ms TTL=112
Reply from 13.126.185.9: bytes=32 time=86ms TTL=112
Reply from 13.126.185.9: bytes=32 time=91ms TTL=112
Ping statistics for 13.126.185.9:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 84ms, Maximum = 91ms, Average = 87ms
C:\Users\abvp>

## [LAB] Assign new Security Group to a Running EC2 Instance

- So far, we have seen modifying security group rules, what if we need to change the security group of a running EC2. That can be done by NIC level.
- Select the EC2 >> Network eth0 >> Interface ID >> Actions >> Change Security Groups



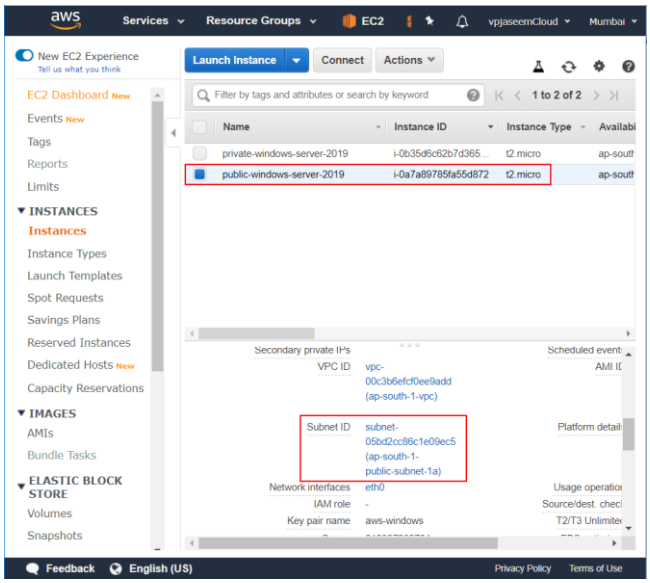



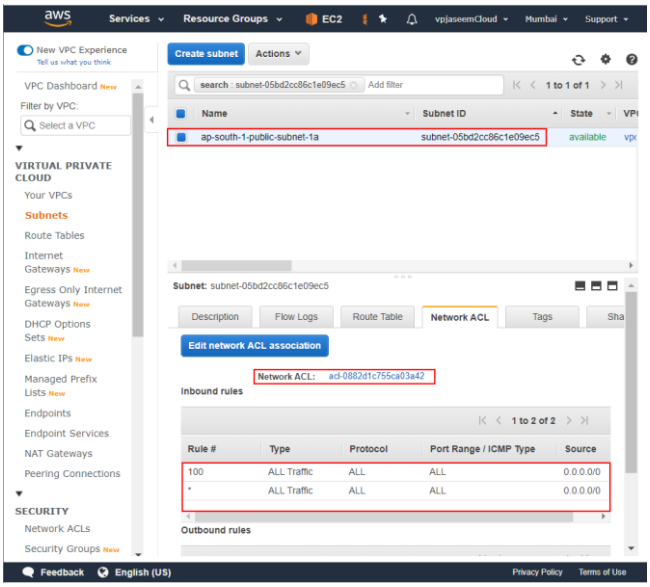
## Network Access Control Lists (NACLs)

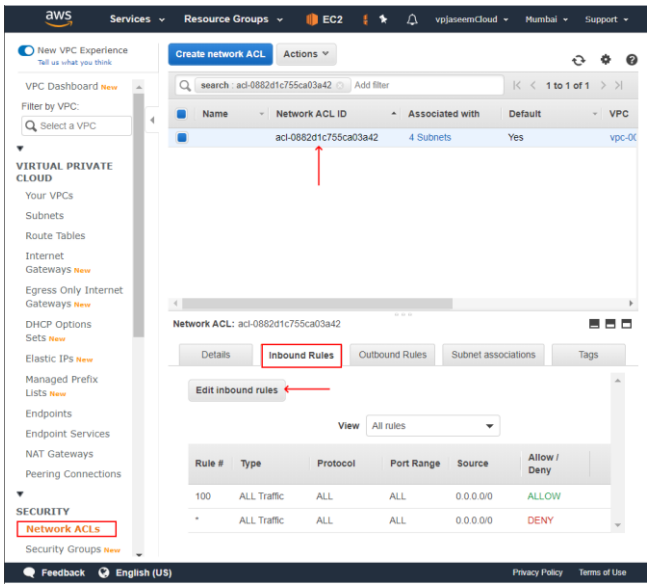
- Virtual firewalls for subnets, rules defined in CIDR range
- Created inside VPC and associated to Subnets
- We can define NACLs for the ports range as well
- Rules applied in order (1st rule applied always)
- One NACL can be allied to many subnets but one subnet will have only one NACL
- By default all inbound and all outbound traffic allowed

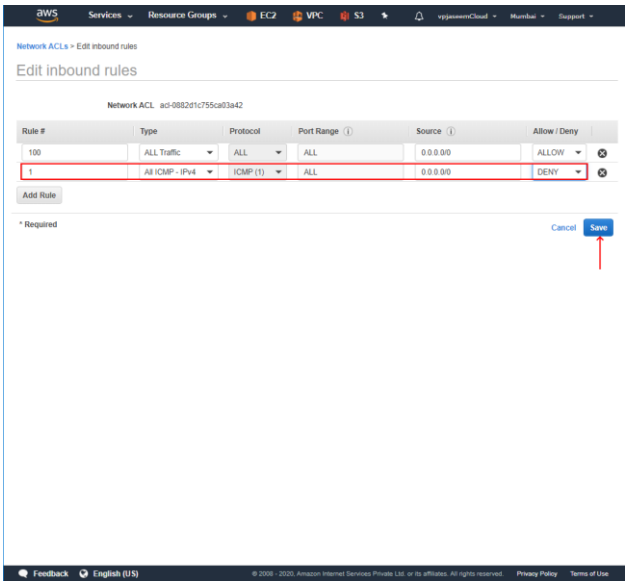
## [LAB] Modify NACLs to Block ICMP Traffic at the Subnet Level

- In the previous lab we have enabled ICMP traffic to our Windows EC2, you can ping the Windows machine from internet.
- Network ACLs allow all traffic (inbound and outbound) by default, in this lab, let's block the ICPM so the subnet (where our EC2 is running) using NACLs.
- You can access NACL from VPC dashboard >> Security >> Network ACLs or from EC2 >> Subnet ID >> NACL ID



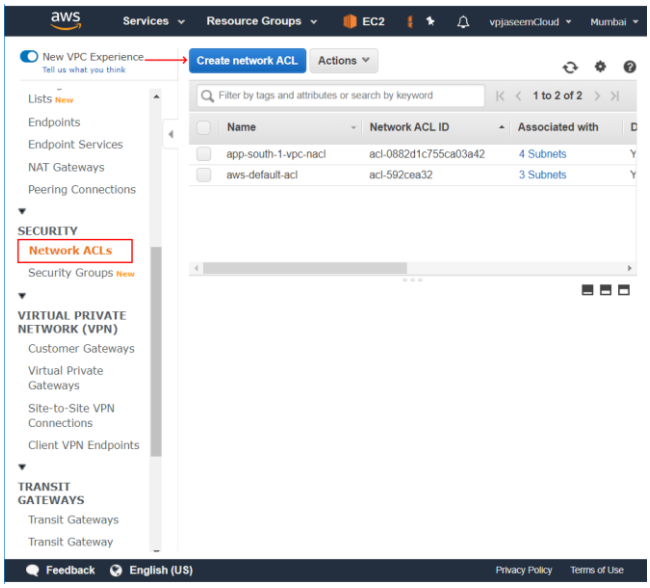



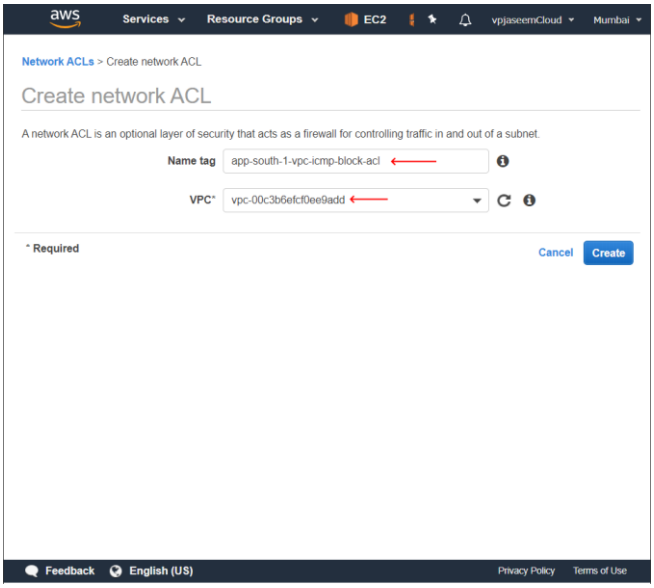



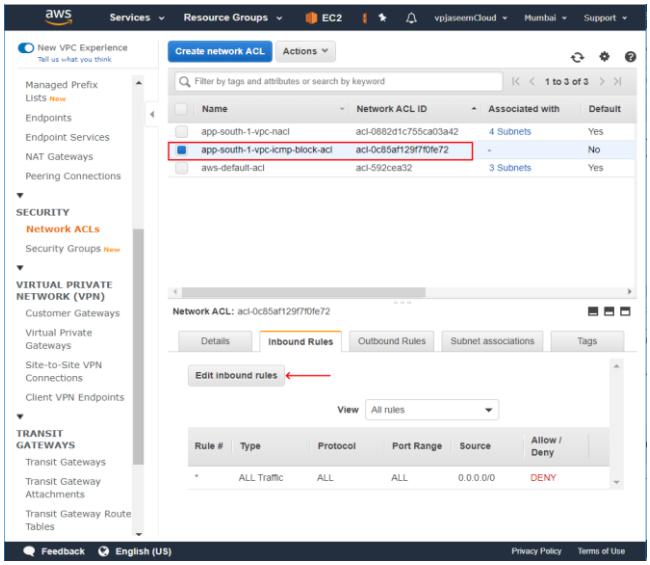



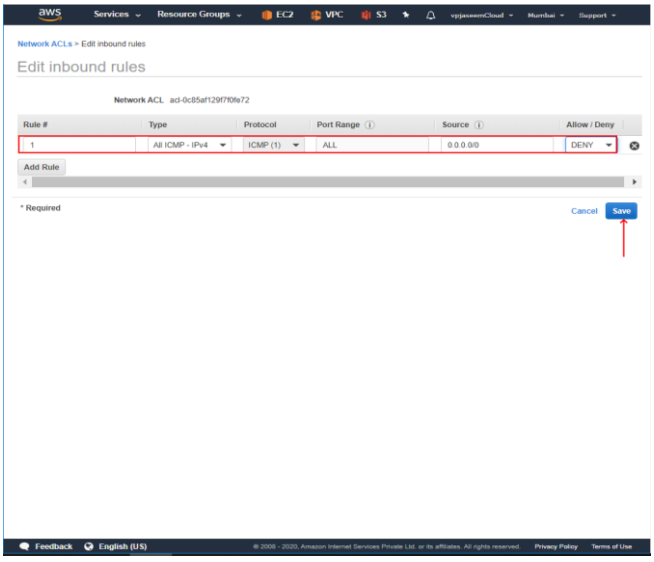
## [LAB] Create new NACL and Associate to Subnet

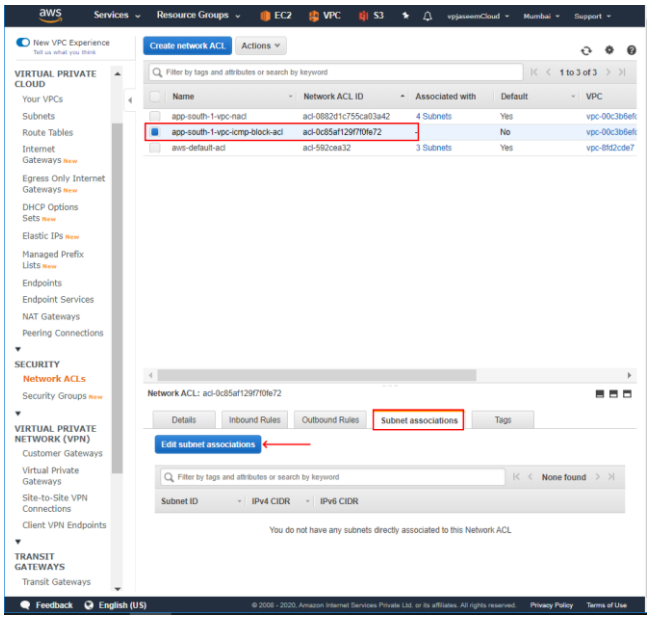
- If you are creating a new NACL, you need to associate to the required subnets.
- When you create a new NACL, all the traffic blocked by default

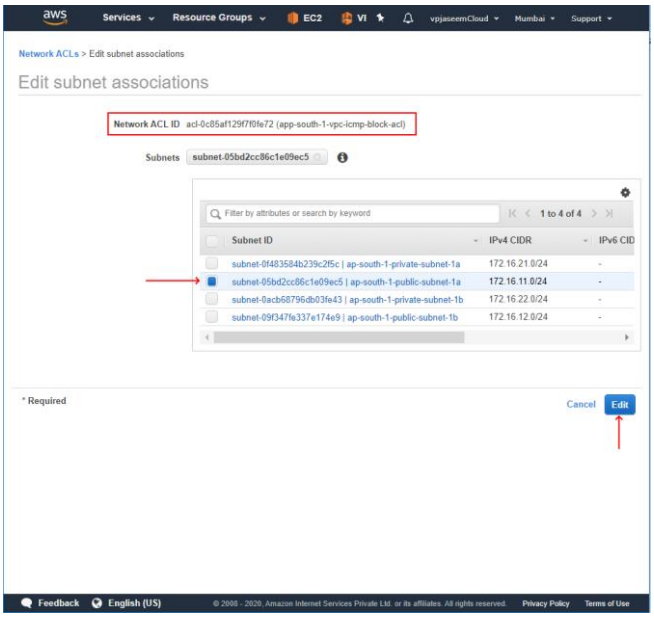


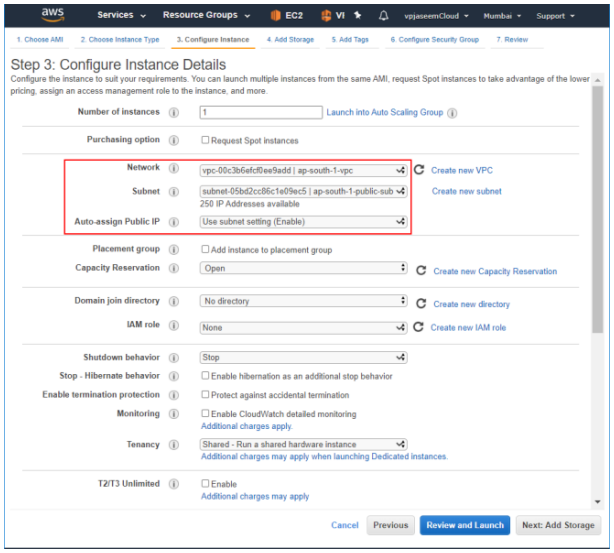



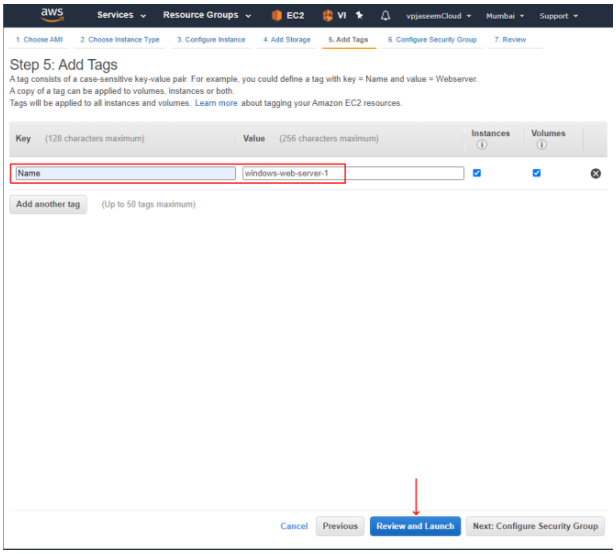



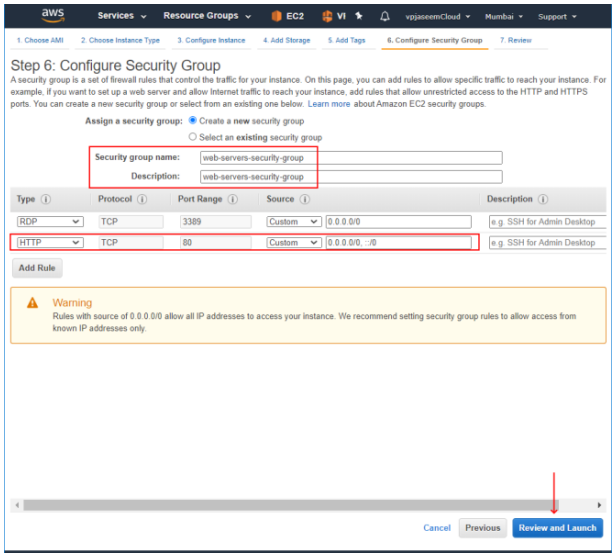
# ASSIGNMENT 1

## [LAB] Manually Host Sample HTML Website in AWS Windows Server 2019 EC2

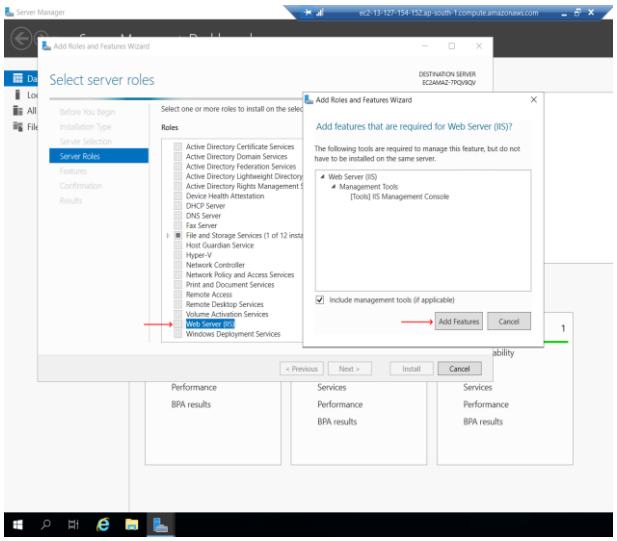
- Let see how to host a sample website in a Windows EC2 and access via public IP and DNS name.
- Launch a new Windows 2019 EC2 instance in Public Subnet



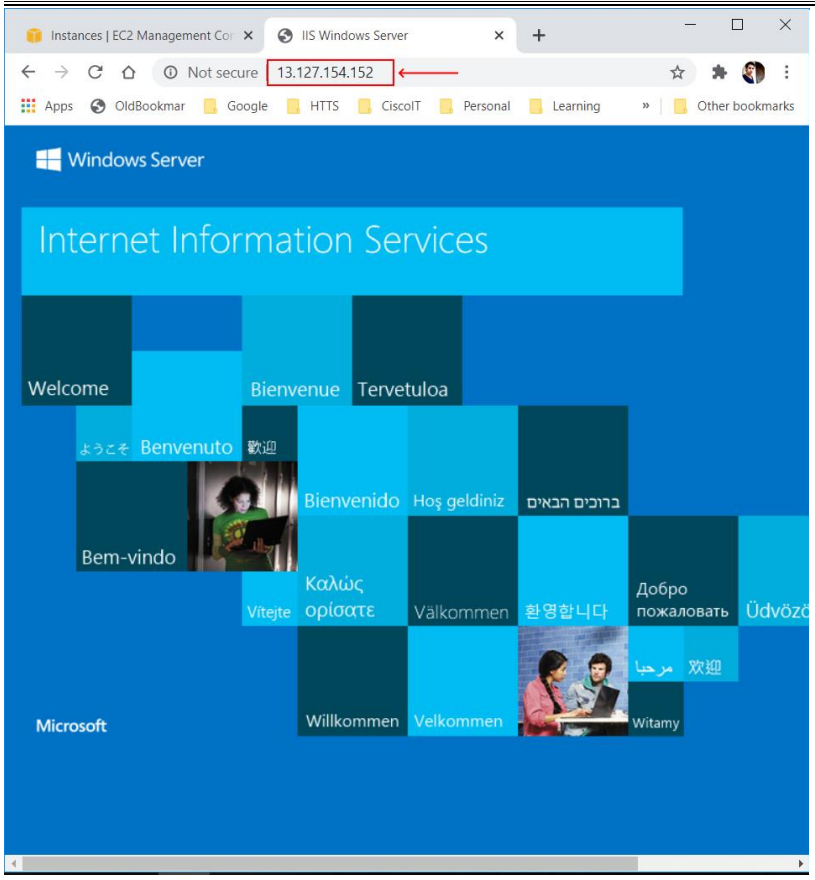



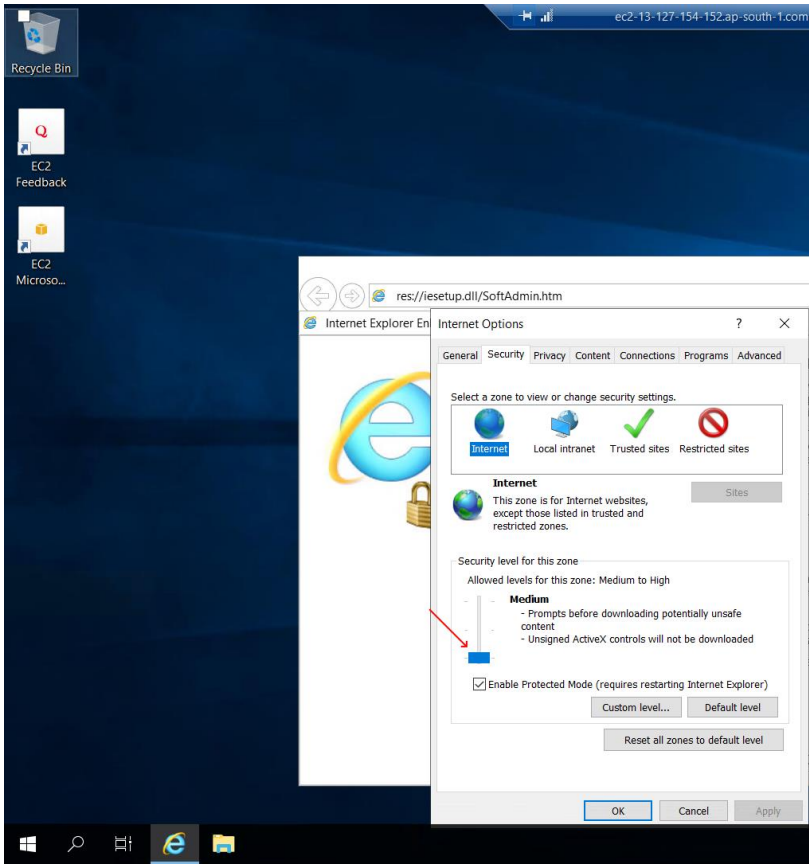

- Make sure your security group allows HTTP port 80 and RDP port 3389



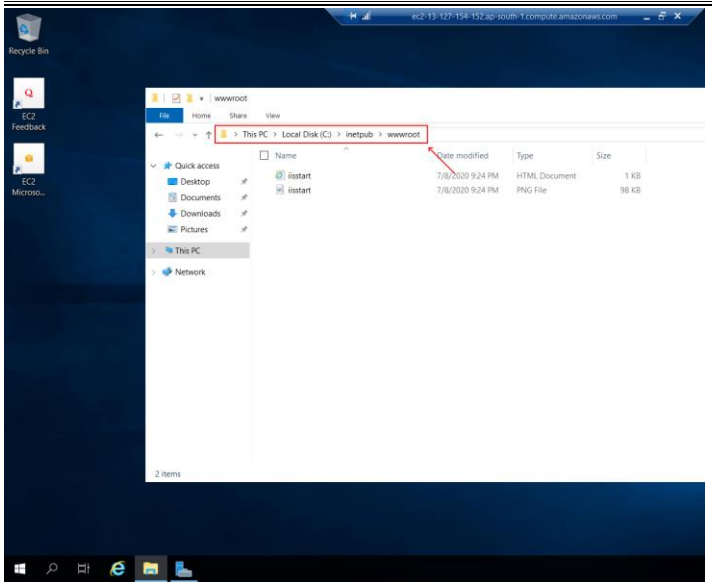

- Once the instance is ready, install Web Server IIS Role there.



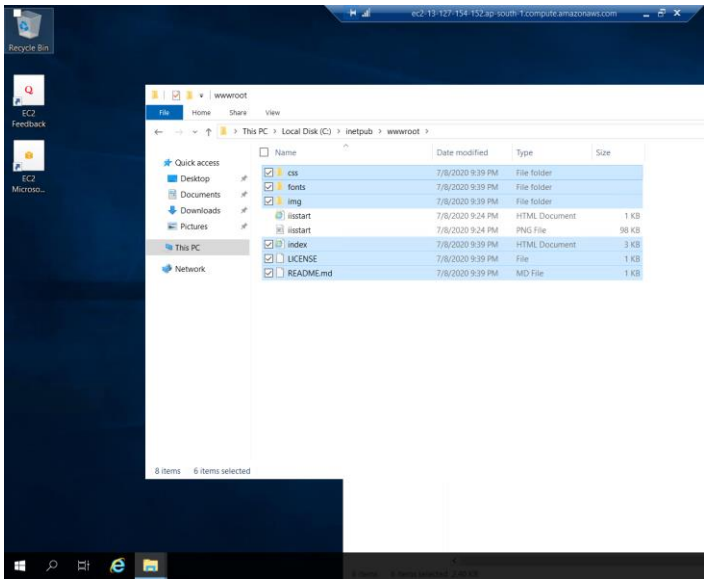

- Now try to access the public IP of the instance from a web browser, you will be able to see IIS default page.
- Now let's host our own website instead of the IIS default.



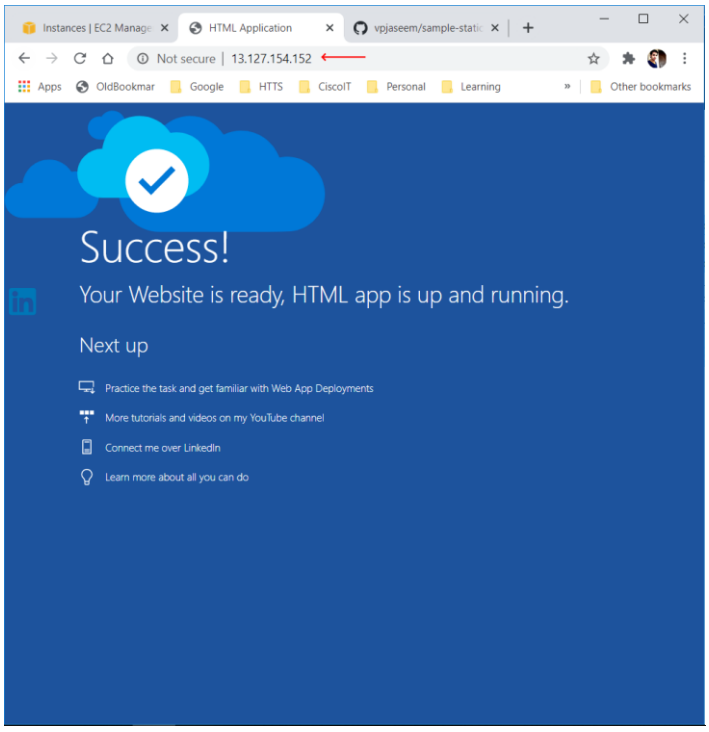

- On your Windows Server, turn off the IE Security Configurations.
- This is just to download a file, not a part of web server configuration



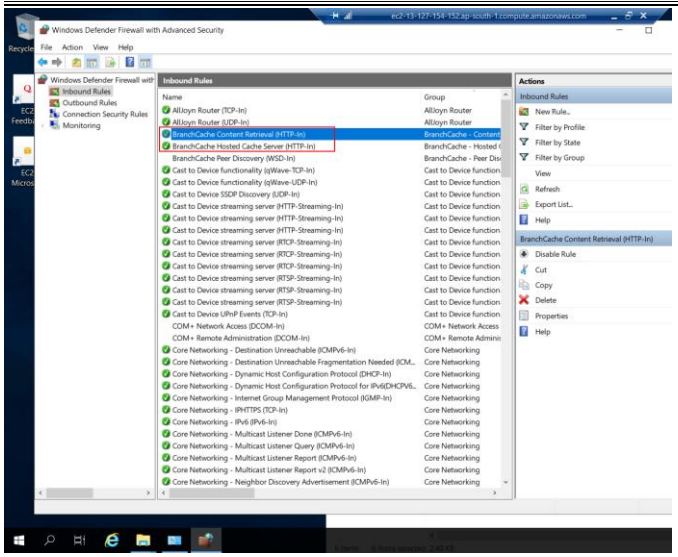
- Go to **C:\inetpub\wwwroot**
- I have created html website, those files can be downloaded from this link: [sample-html-web-app]



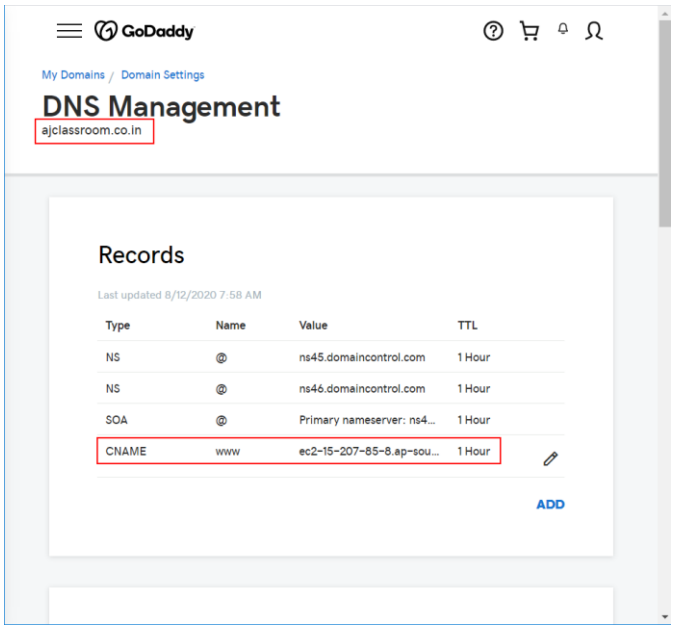
- Download those files and extract the content to **C:\inetpub\wwwroot**

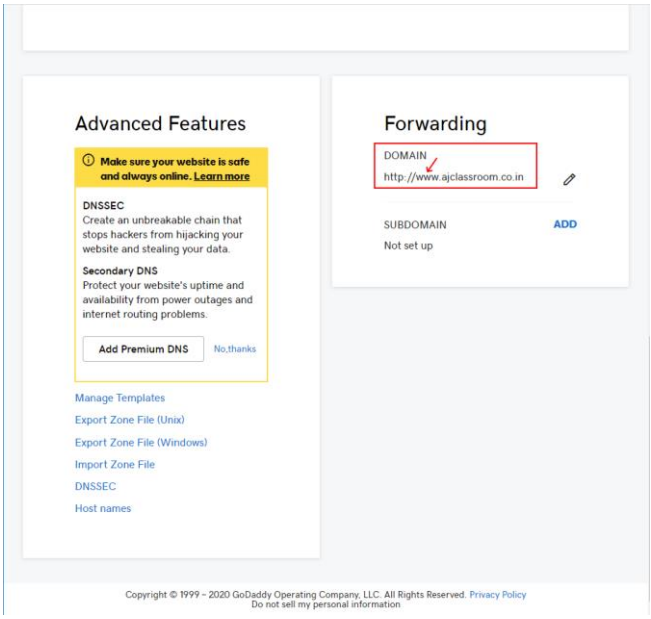


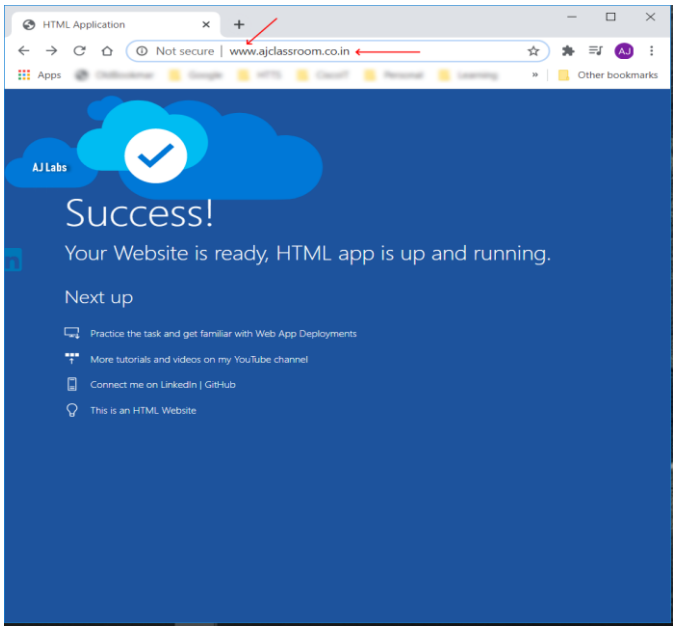

- Now try to access again the public IP or AWS FQDN of the instance from a web browser, you will be able to see your new website.
- We can access it via public domain name as well.
- I have a domain Godaddy name **ajclassroom.co.in** where I have added CNAME (public FQDN) of Windows EC2



- While accessing via FQDN, please make sure you have HTTP port allowed in Windows Firewall
- Note: You need to allow on Windows Firewall as well as AWS Security Group

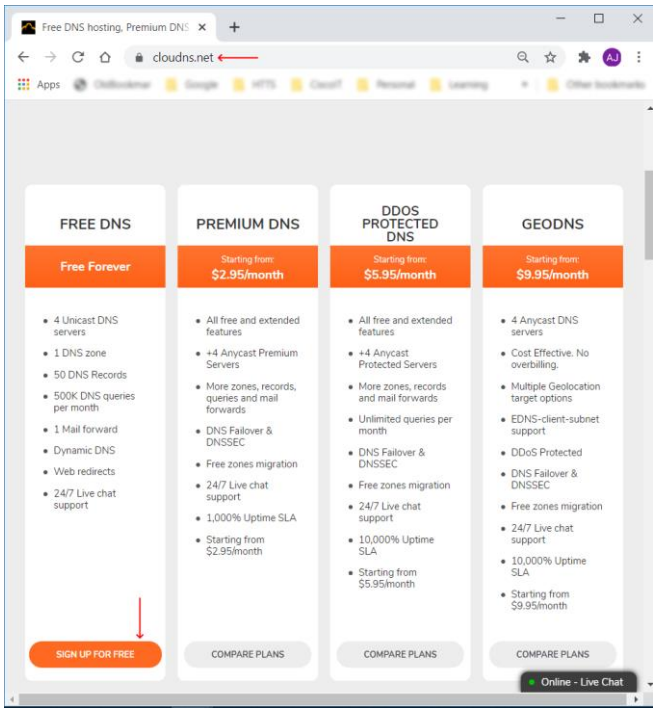


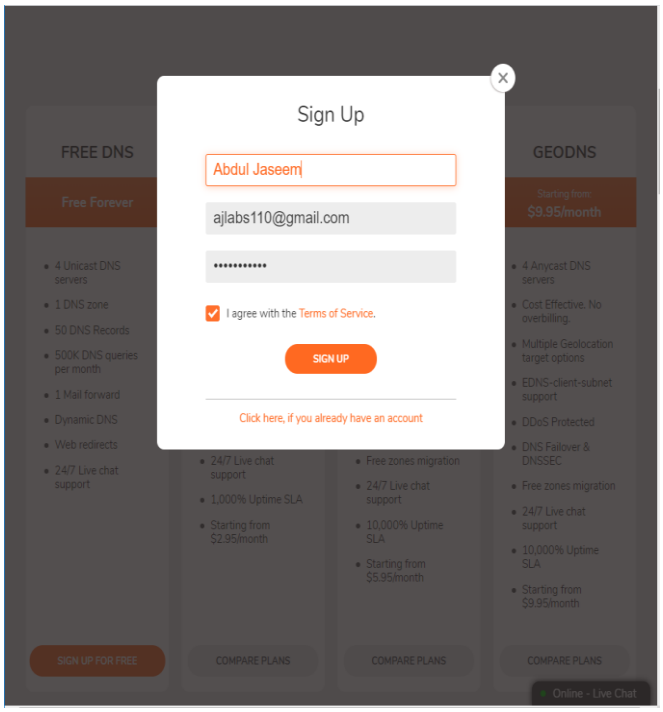



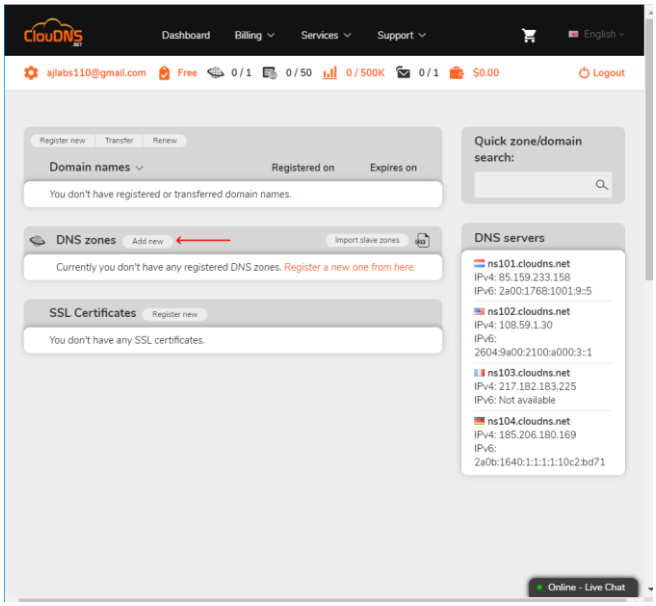



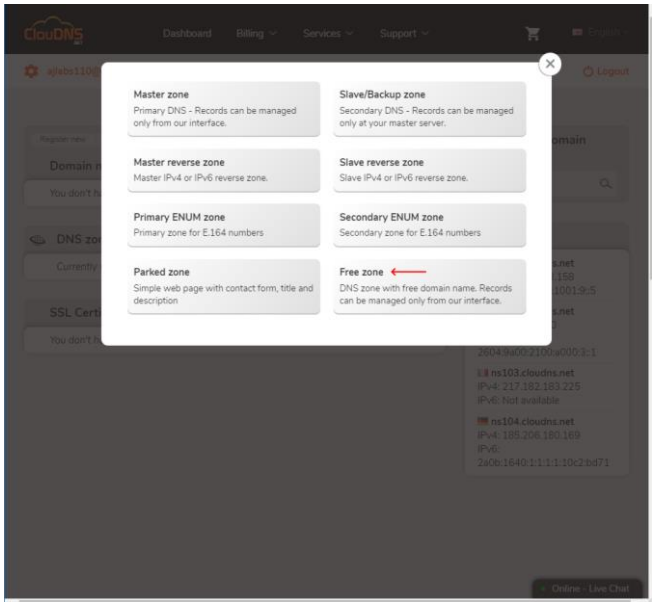
## [LAB] Get a free Domain Name

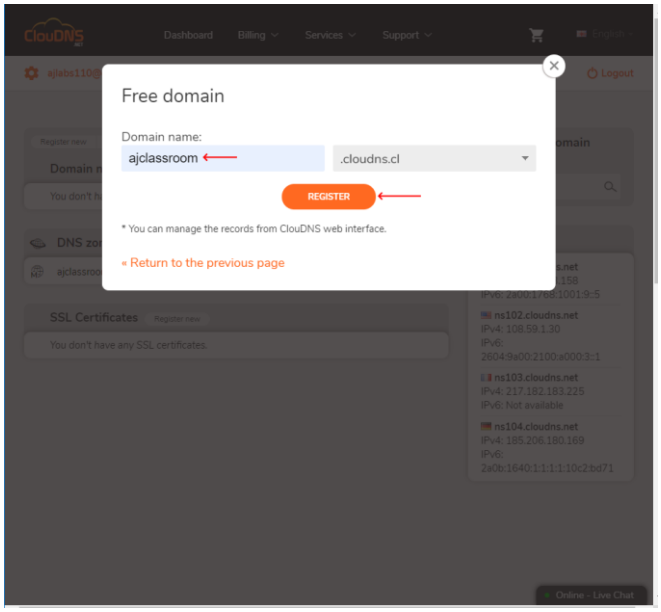
- Go to CloudNS and signup for a FREE account.

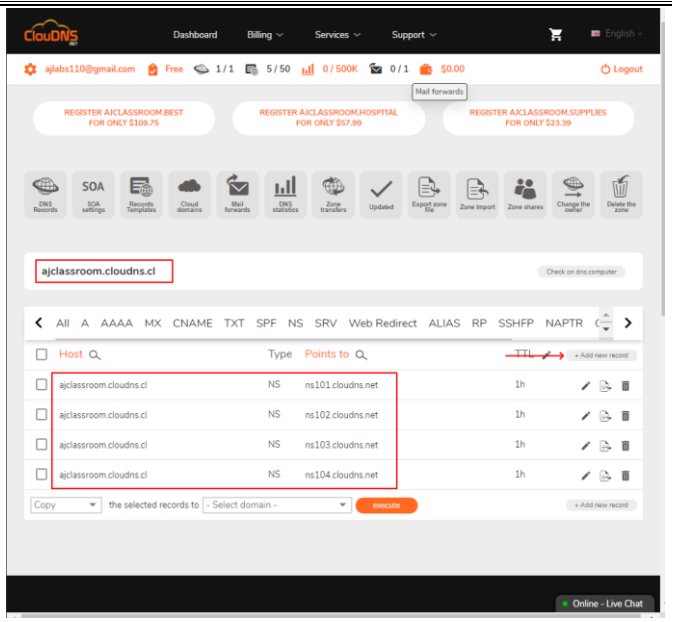


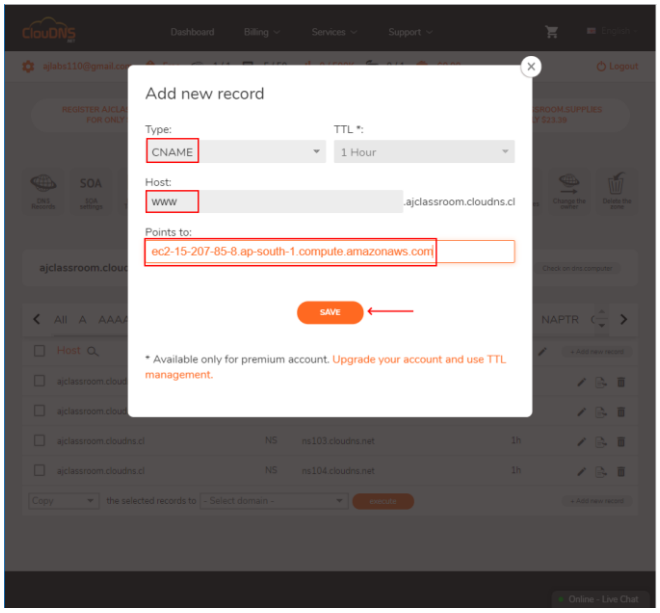



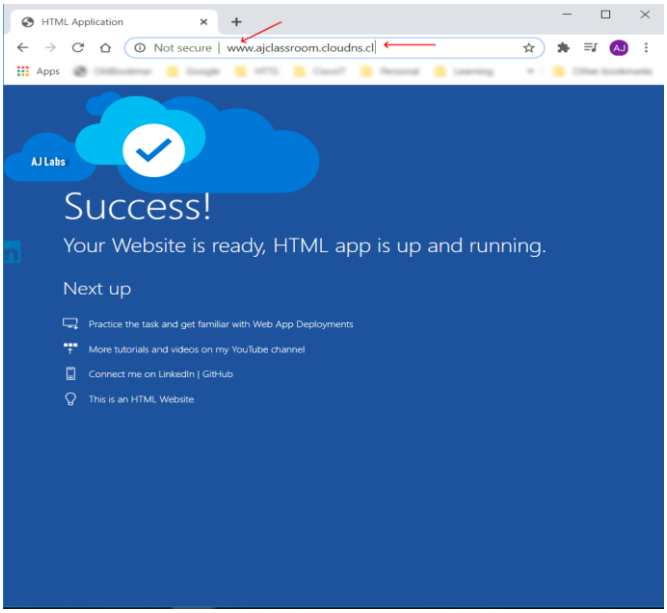




Free domain
Domain name:
ajclassroom
.cloudns.cl
REGISTER
* You can manage the records from ClouDNS web interface.
« Return to the previous page

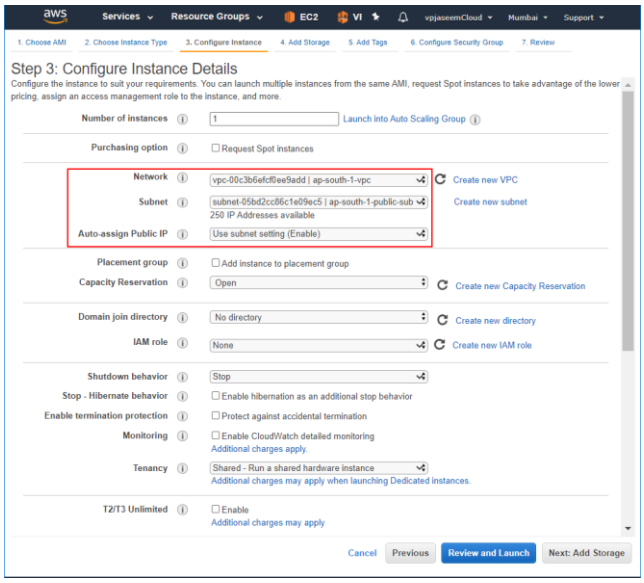

ajclassroom.cloudns.cl
Host
Type
Points to
TTL
ajclassroom.cloudns.cl NS ns101.cloudns.net 1h
ajclassroom.cloudns.cl NS ns102.cloudns.net 1h
ajclassroom.cloudns.cl NS ns103.cloudns.net 1h
ajclassroom.cloudns.cl NS ns104.cloudns.net 1h

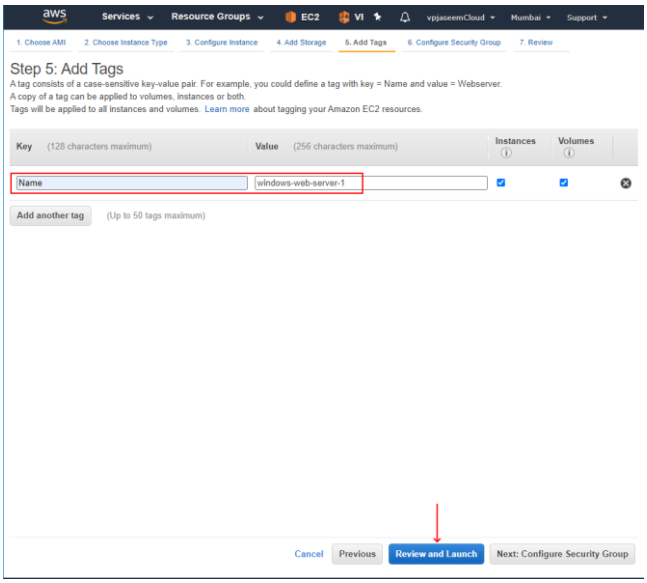

Add new record
Type:
CNAME
TTL *:
1 Hour
Host:
www
.ajclassroom.cloudns.cl
Points to:
ec2-15-207-85-8.ap-south-1.compute.amazonaws.com
SAVE
* Available only for premium account. Upgrade your account and use TTL management.

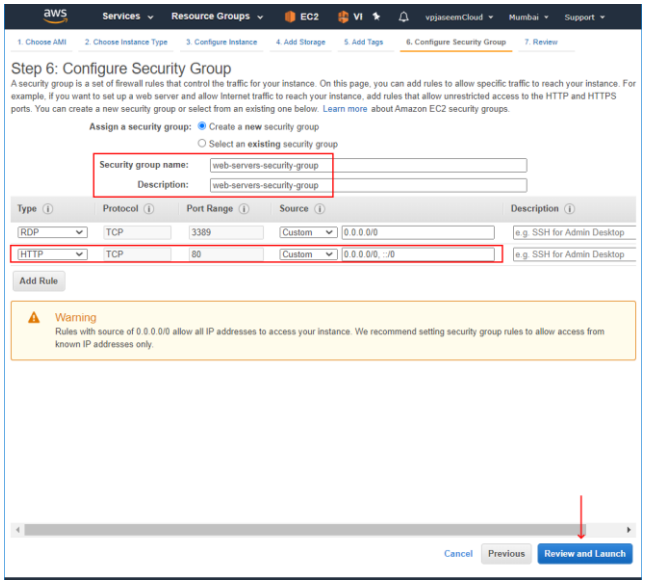

HTML Application
Not secure | www.ajclassroom.cloudns.cl
AJ Labs
Success!
Your Website is ready, HTML app is up and running.
Next up
Practice the task and get familiar with Web App Deployments
More tutorials and videos on my YouTube channel
Connect me on LinkedIn | GitHub
This is an HTML Website

## [LAB] PowerShell Script to Host Sample HTML Website in AWS Windows Server 2019 EC2

- Whatever steps we have performed above can be automated using PowerShell scripts. In this lab, let's do the same process using PowerShell.
- Launch another new Windows 2019 EC2 instance in Public Subnet



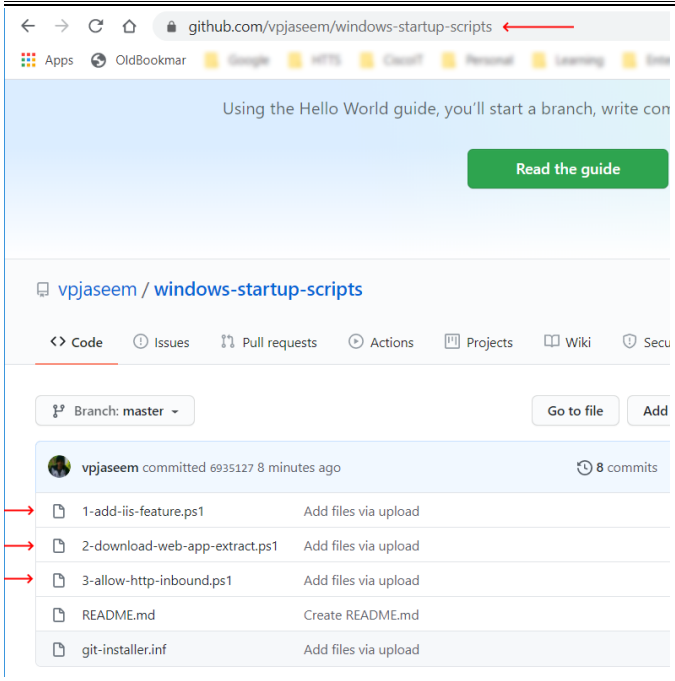




- Make sure your security group allows HTTP port 80 and RDP port 3389

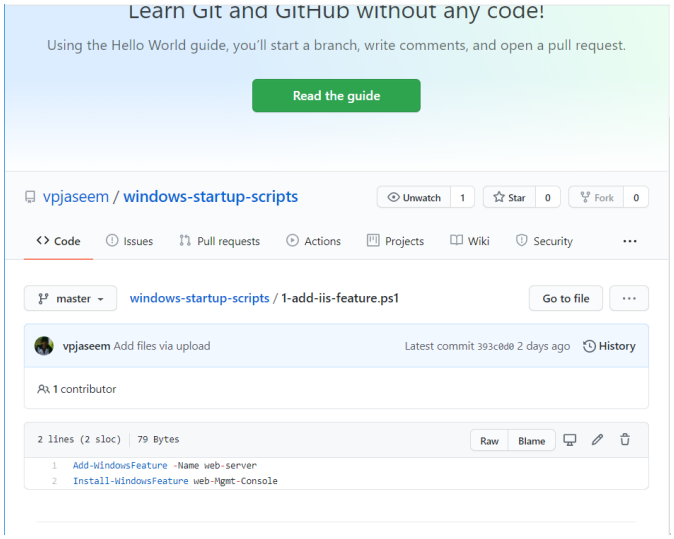
- We need to follow 3 steps,
  - Add Windows Feature IIS
  - Download and place the sample web app files to wwwroot folder
  - Allow HTTP port 80 in Windows firewall

I have 3 PowerShell scripts available in the this link: [PowerShell Scripts]

Download those and execute one by one

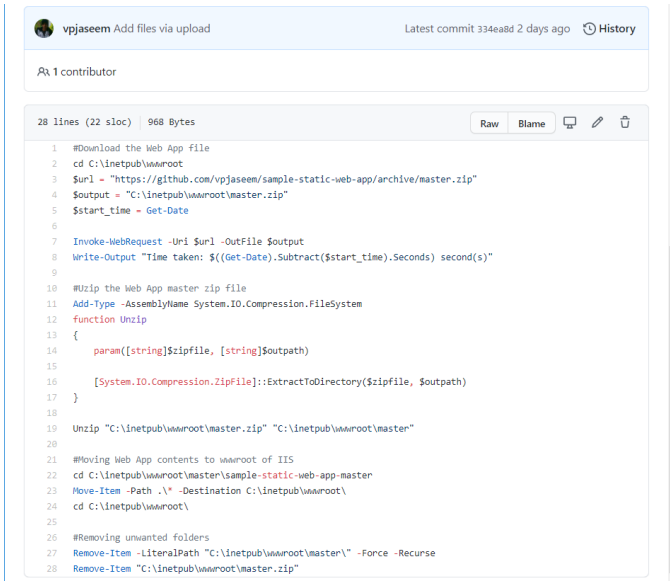



- I have 3 PowerShell scripts available in the this link: [PowerShell Scripts]



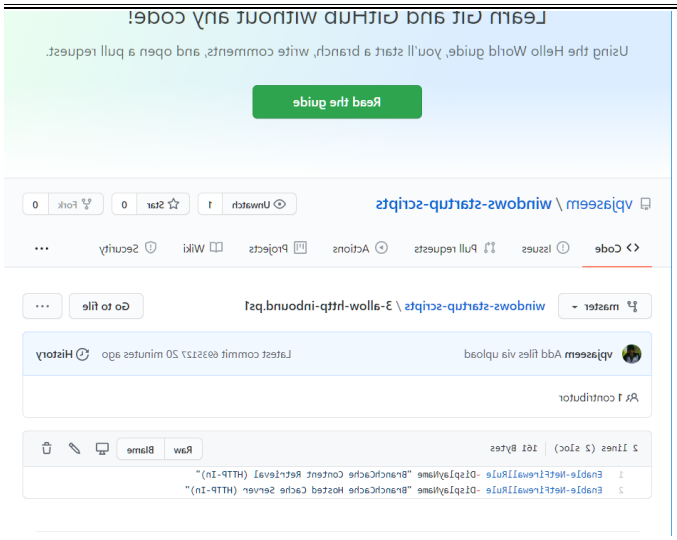

Script 1:

- This is used to add Windows Feature IIS




Script 2: Performs below tasks

- Download the Web App files
- Unzip the file content
- Place the contents to wwwroot folder of IIS
- Delete the unwanted files



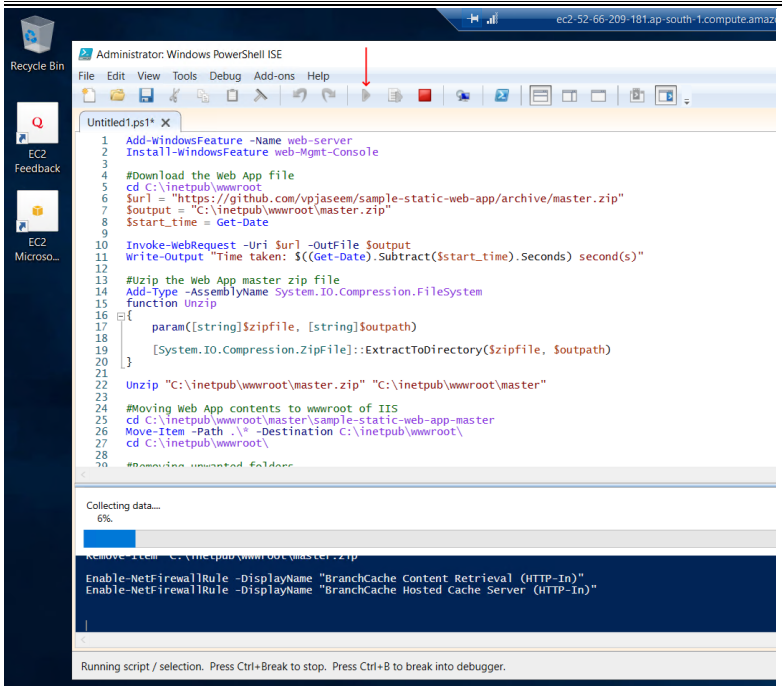
Script 3:

- Enable HTTP port on Windows firewall

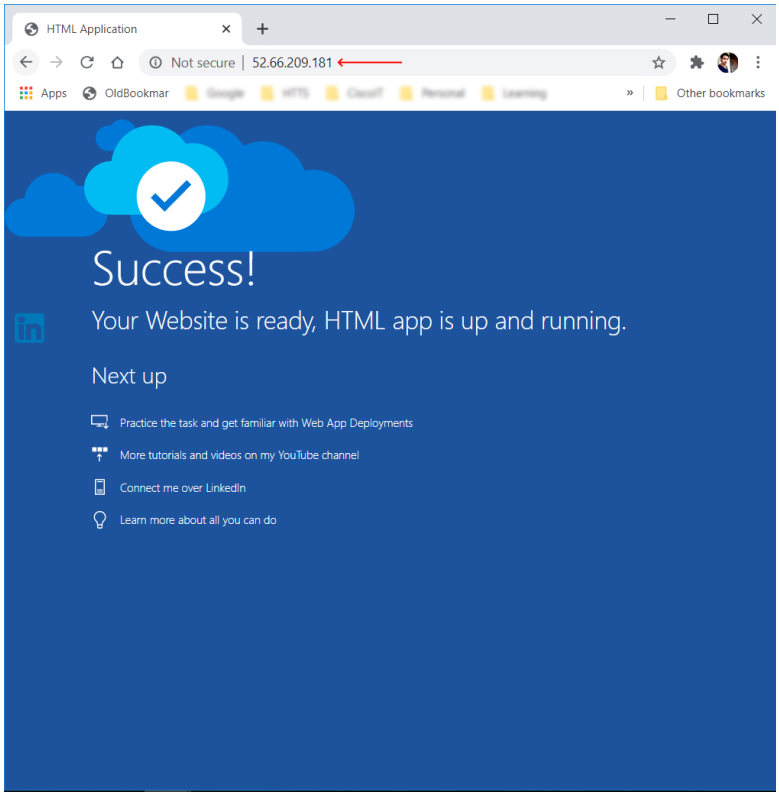
```
Add-WindowsFeature -Name web-server
Install-WindowsFeature web-Mgmt-Console
```

```
#Download the Web App file
cd C:\inetpub\wwwroot
$url = "https://github.com/vpjaseem/sample-static-html-web-
app/archive/master.zip"
$output = "C:\inetpub\wwwroot\master.zip"
$start_time = Get-Date

Invoke-WebRequest -Uri $url -OutFile $output
Write-Output "Time taken: $((Get-Date).Subtract($start_time).Seconds) second(s)"

#Uzip the Web App master zip file
Add-Type -AssemblyName System.IO.Compression.FileSystem
function Unzip
{
    param([string]$zipfile, [string]$outpath)

    [System.IO.Compression.ZipFile]::ExtractToDirectory($zipfile, $outpath)
}

Unzip "C:\inetpub\wwwroot\master.zip" "C:\inetpub\wwwroot\master"

#Moving Web App contents to wwwroot of IIS
cd C:\inetpub\wwwroot\master\sample-static-html-web-app-master
Move-Item -Path .\* -Destination C:\inetpub\wwwroot\
cd C:\inetpub\wwwroot\

#Removing unwanted folders
Remove-Item -LiteralPath "C:\inetpub\wwwroot\master\" -Force -Recurse
```

```
Enable-NetFirewallRule -DisplayName "BranchCache Content Retrieval (HTTP-In)"
Enable-NetFirewallRule -DisplayName "BranchCache Hosted Cache Server (HTTP-In)"
```




- Run all the 3 scripts using Windows PowerShell (one by one or all at once)




- Your Website is ready!

## [LAB] Hosted PowerShell Script to Host HTML Website in AWS Windows Server 2019 EC2

- We know that all the 2 scripts used for deploying simple website is available on GitHub
- Instead of copying and exciting it, you can create a master PowerShell script that can call multiple other scripts hosted in any cloud (GitHub, GoogleDrive, Drobox, AWS S3, etc.)

```
iex ((New-Object System.Net.WebClient).DownloadString('https://raw.githubusercontent.com/vpjaseem/windows-startup-scripts/master/1-add-iis-feature.ps1'))
iex ((New-Object System.Net.WebClient).DownloadString('https://raw.githubusercontent.com/vpjaseem/windows-startup-scripts/master/2-download-web-app-extract.ps1'))
iex ((New-Object System.Net.WebClient).DownloadString('https://github.com/vpjaseem/windows-startup-scripts/raw/master/3-allow-http-inbound.ps1'))
```

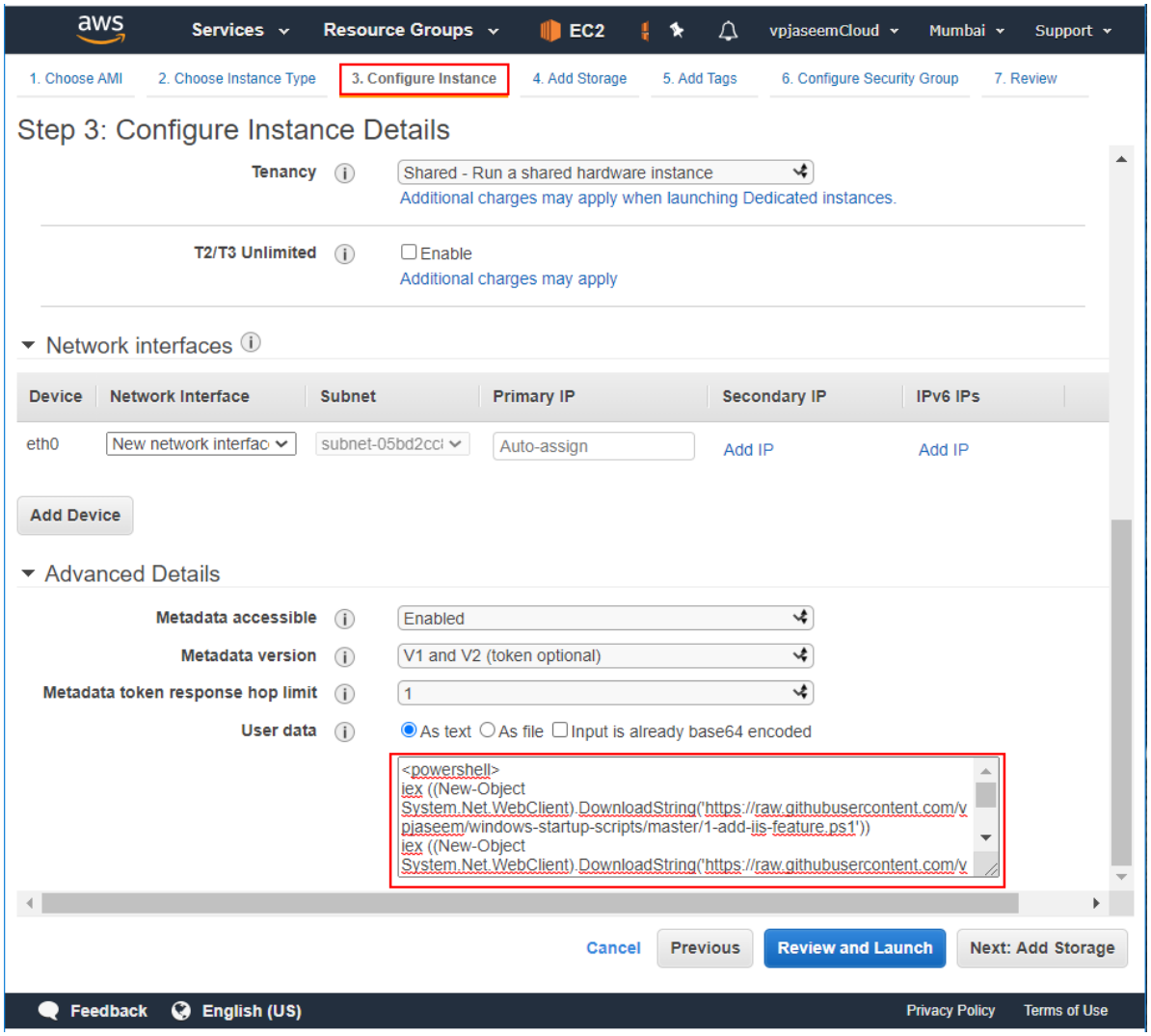
- In this case, all the 3 Scripts are hosted in GitHub cloud, using the above Script we are calling one by one.
- This method will make the script modification easy.
- Launch a new Windows Server 2019 EC2 instance in public subnet and allow port 80 on security group. Then just run above script to get your website deployed.

## [LAB] HTML Web Hosting Automation in AWS Windows Server 2019 EC2 Using User Data

- User Data is used to pass some startup scripts to EC2, when the EC2 spun up, it will execute those commands automatically.
- Out master PowerShell script can be supplied to a Windows Server 2019 EC2 instance via User Data, it will be executed automatically, and your web app deployed in minutes without you even logging in to windows.

```
<powershell>
iex ((New-Object System.Net.WebClient).DownloadString('https://raw.githubusercontent.com/vpjaseem/windows-startup-scripts/master/1-add-iis-feature.ps1'))
iex ((New-Object System.Net.WebClient).DownloadString('https://raw.githubusercontent.com/vpjaseem/windows-startup-scripts/master/2-download-web-app-extract.ps1'))
iex ((New-Object System.Net.WebClient).DownloadString('https://github.com/vpjaseem/windows-startup-scripts/raw/master/3-allow-http-inbound.ps1'))
</powershell>
```

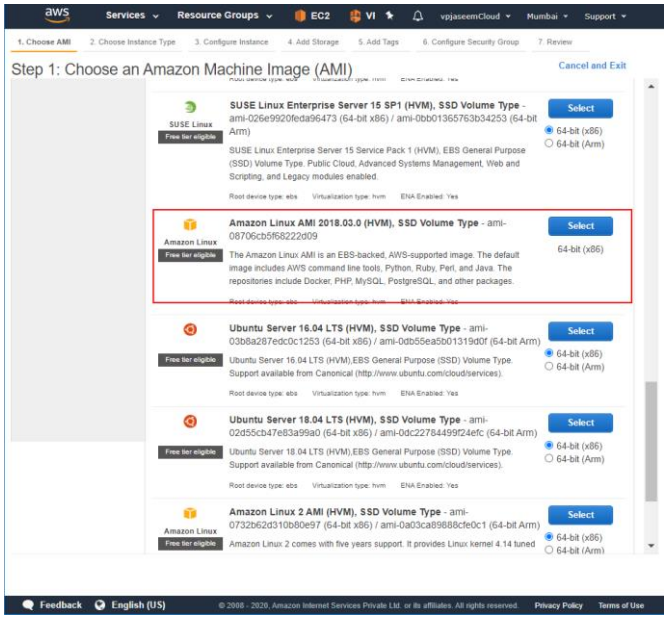
- You can pass any kind of PowerShell script in this way.
- Launch the EC2 like below User data, make sure you allow port 80 in security group.

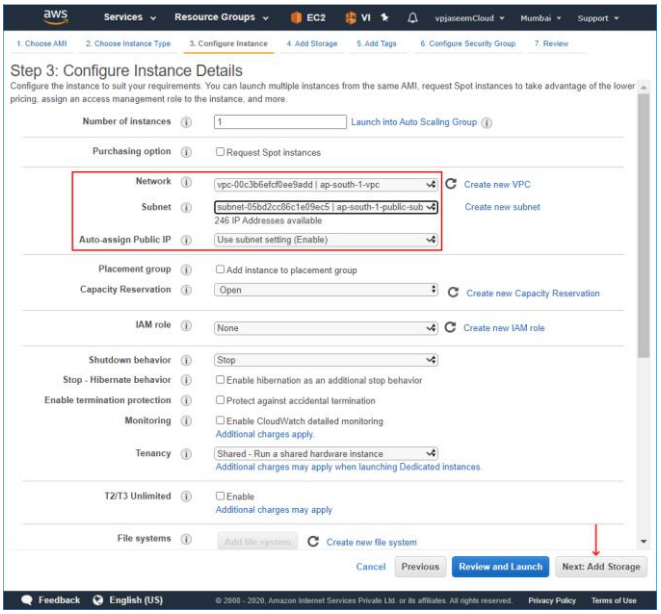


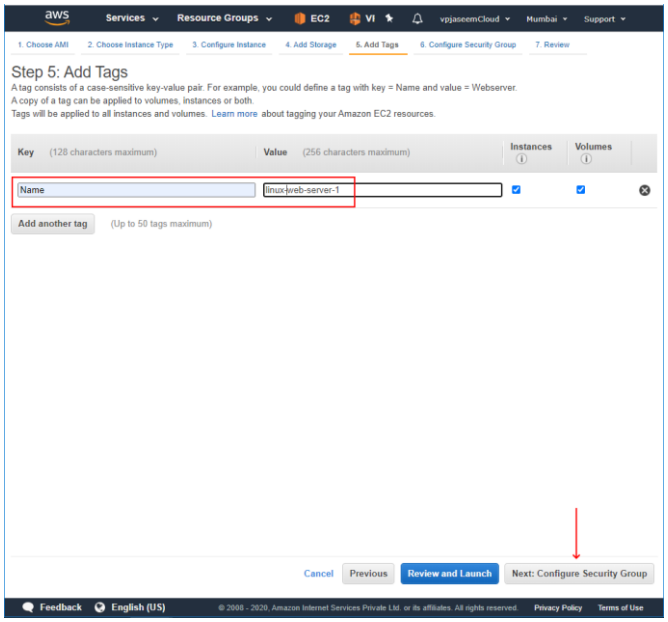
# ASSIGMENT 2

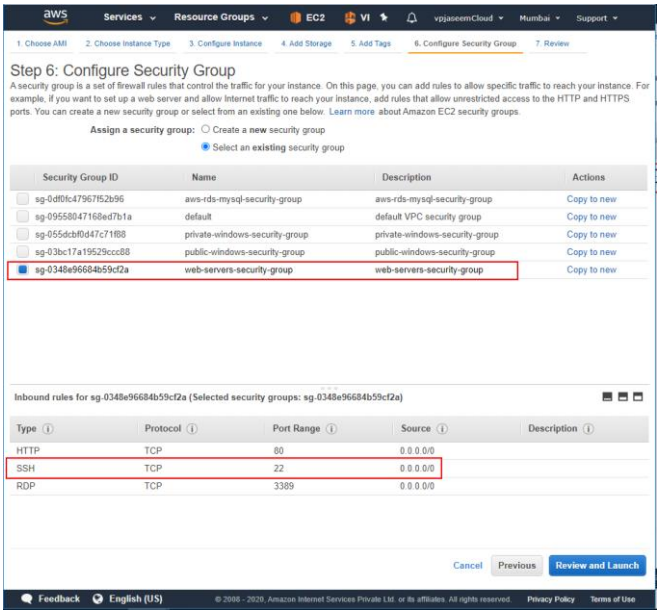
## [LAB] Manually Host Sample PHP Website in AWS Linux EC2

- Launch Amazon Linux AMI in public subnet and allow port 80 in security group




- SSH to the Linux and run below commands,
  - Install Apache HTTPD Web Server
  - Clone the web app files from GitHub to home/var/www/html/ directory

```
sudo yum update -y
sudo yum install httpd -y
sudo yum install php -y
sudo service httpd start
sudo chkconfig httpd on
sudo yum install git -y


cd /$home/var/www/html/
sudo git clone https://github.com/vpjaseem/sample-static-php-web-app.git
sudo cp -r /var/www/html/sample-static-php-web-app/* /var/www/html/
sudo service httpd restart
```

## [LAB] Bash Script to Host Sample PHP Website in AWS Linux EC2

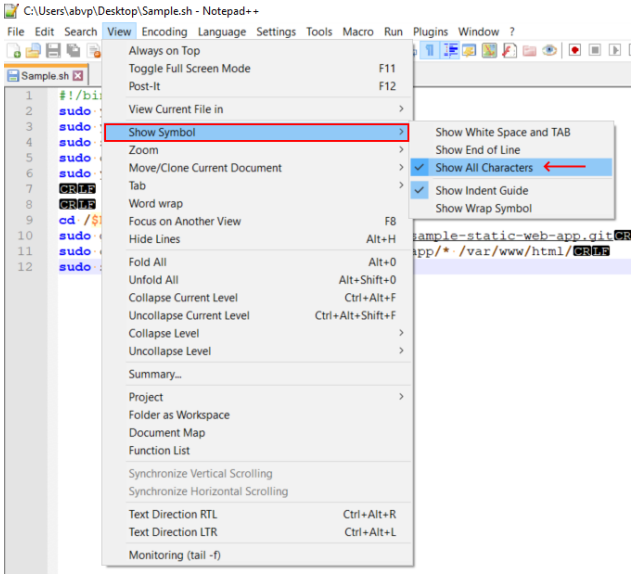
- The same above Linux commands can be grouped into bash script and execute once to deploy the web server.
- Save this as <FILE-NAME>.sh (*web-deploy.sh*)

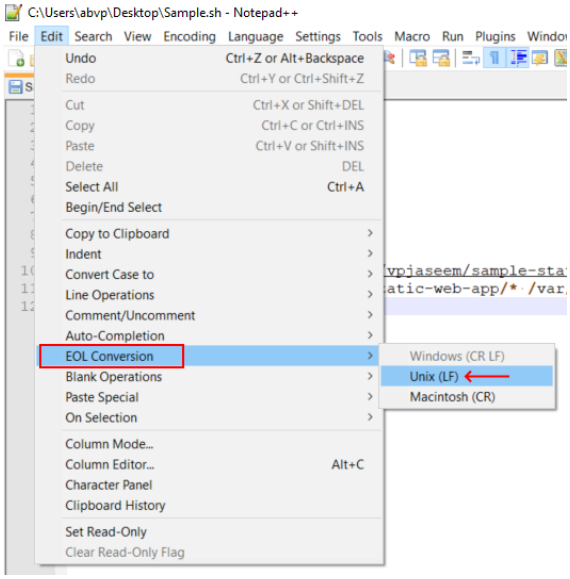
```
#!/bin/bash
sudo yum update -y
sudo yum install httpd -y
sudo service httpd start
sudo chkconfig httpd on
sudo yum install git -y


cd /$home/var/www/html/
sudo git clone https://github.com/vpjaseem/sample-static-web-app.git
sudo cp -r /var/www/html/sample-static-web-app/* /var/www/html/
sudo service httpd restart
```

- Grand permission using *chmod +x web-deploy.sh* then execute *./web-deploy.sh* or *sudo ./web-deploy.sh*
- The same script can also be passed as User Data to automate the process at the time of EC2 instance launching.

***Note:***

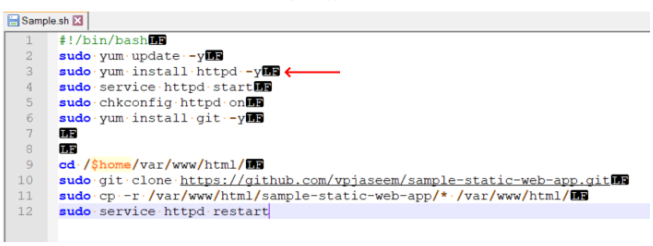
If you are using Windows Text editors for bash script, the new line will be CR+LF (Carriage Return + Line Feed) but Linux understands LF only.




*Windows EOL*


*Unix EOL*



## [LAB] Hosted Bash Script to Host Sample PHP Website in AWS Linux EC2

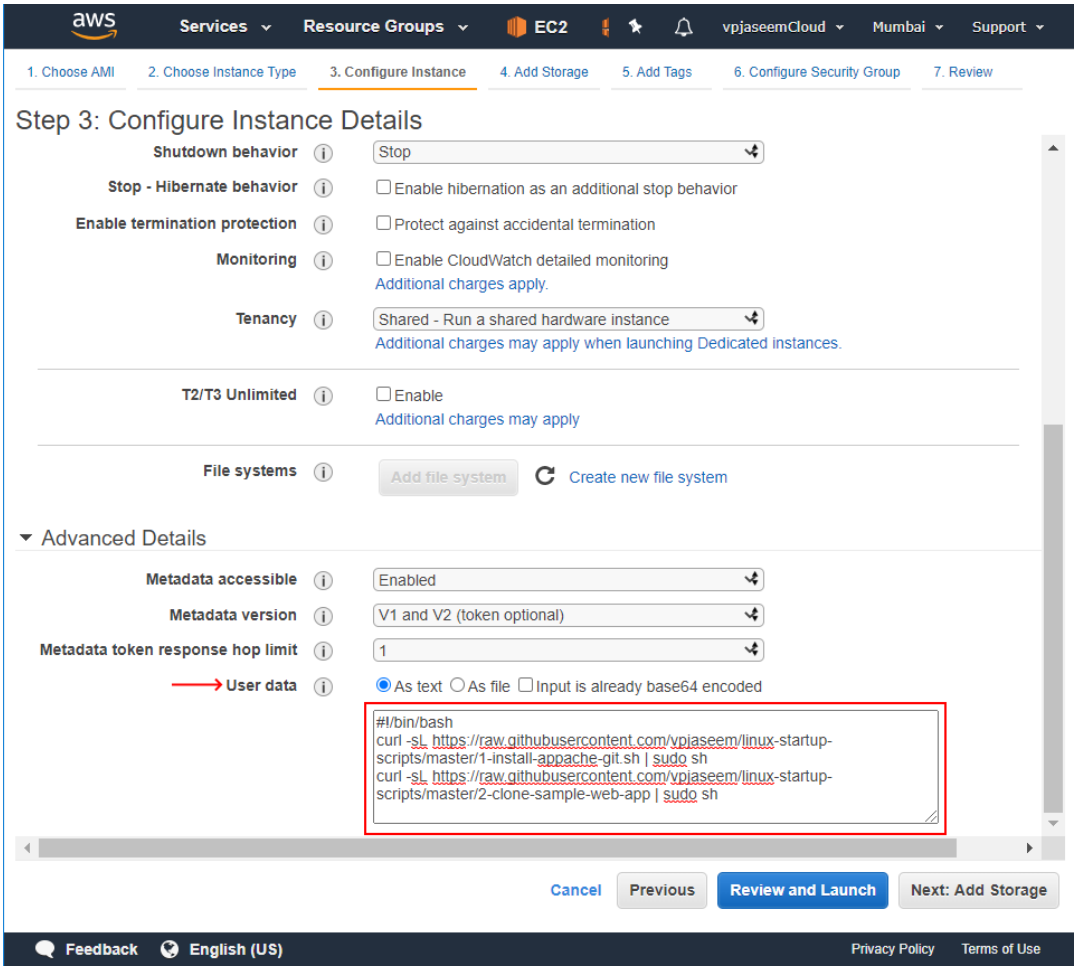
- Alternately, we can host the same bas script in GitHub (or GoogleDrive, DropBox, S3, etc.) and use another bash script to call all these.
- Below is the master bash script that calls the other 2 bash scripts to install httpd, PHP and clone web app files.

```
#!/bin/bash
curl -sL https://raw.githubusercontent.com/vpjaseem/linux-startup-scripts/master/1-install-appache-php-git.sh | sudo sh
curl -sL https://raw.githubusercontent.com/vpjaseem/linux-startup-scripts/master/3-clone-sample-php-web-app.sh | sudo sh
```

## [LAB] PHP Web Hosting Automation in AWS Linux EC2 Using User Data

- Above Script can be passed to User Data so that the complete process of Web app deployment will be automated end to end.

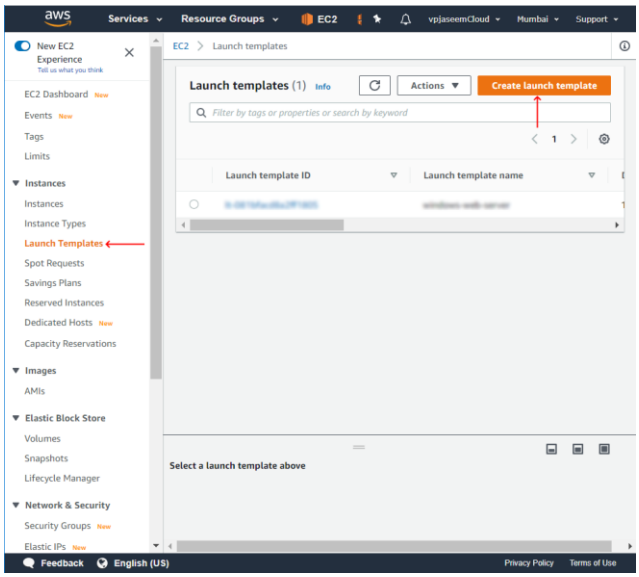
```
#!/bin/bash
curl -sL https://raw.githubusercontent.com/vpjaseem/linux-startup-scripts/master/1-install-appache-php-git.sh | sudo sh
curl -sL https://raw.githubusercontent.com/vpjaseem/linux-startup-scripts/master/3-clone-sample-php-web-app.sh | sudo sh
```

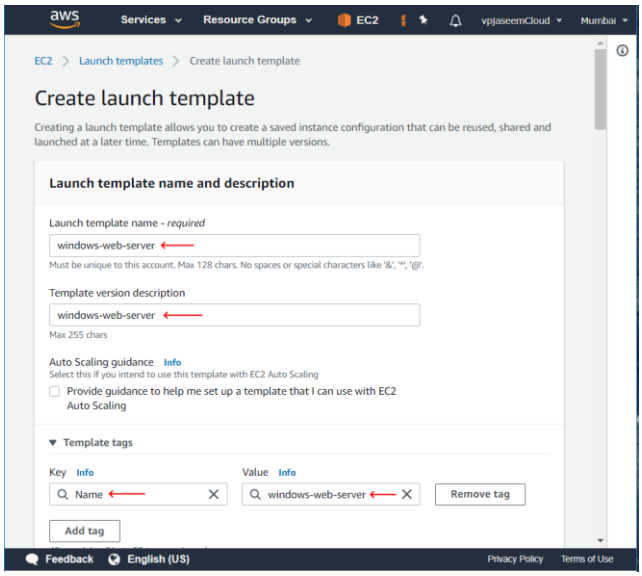


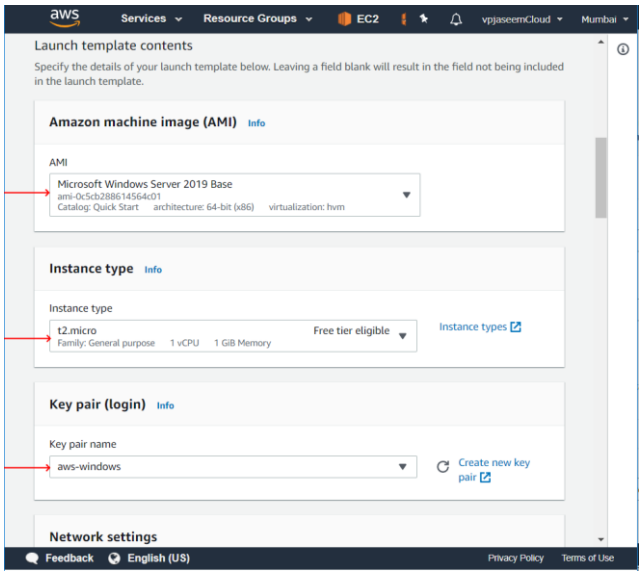

- You can put this all to a Launch Template and deploy web app in 1 click.

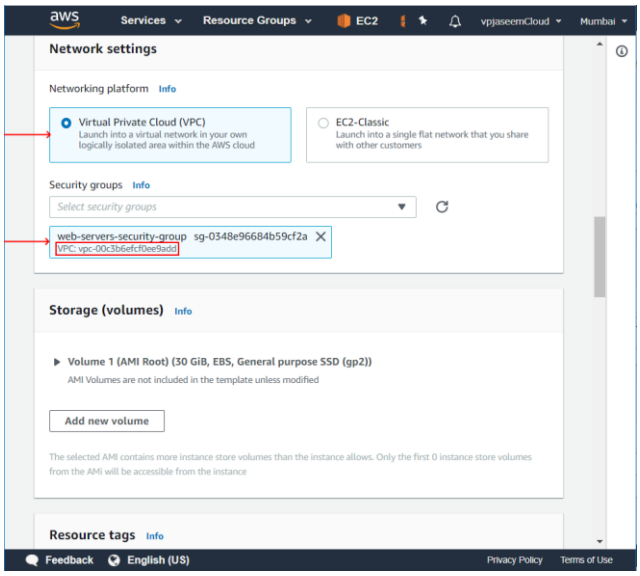
# EC2 Launch Template

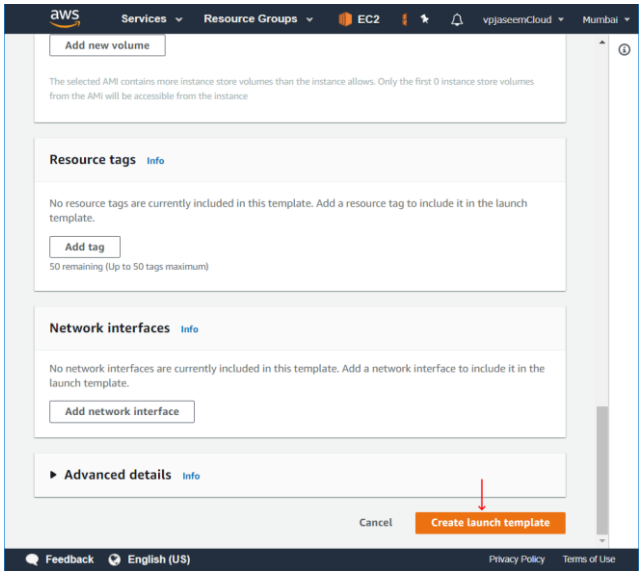
- Launch templates used to automate the launch configurations
- We create a template by pre-configuring all the necessary steps required for a particular EC2 instance

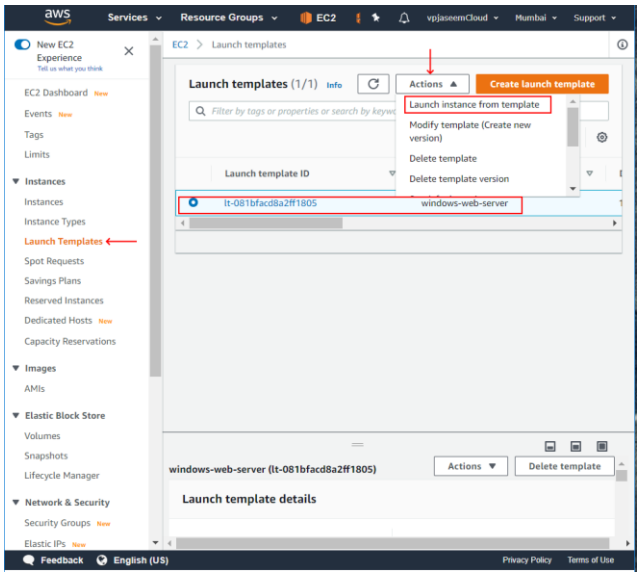


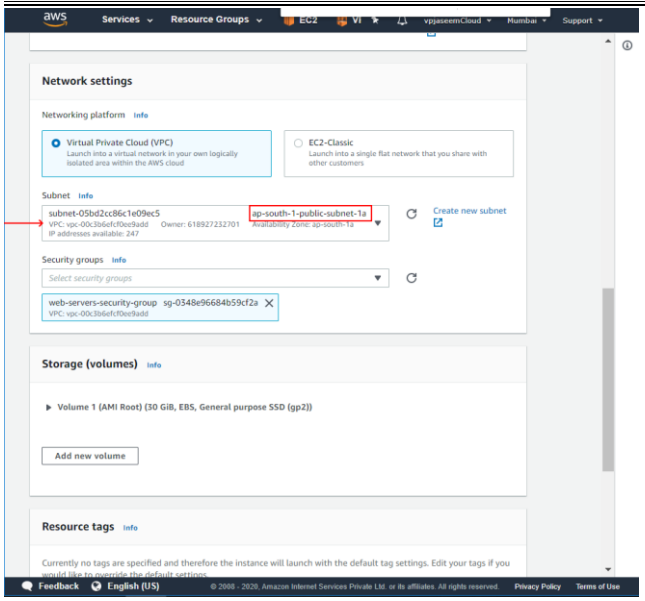



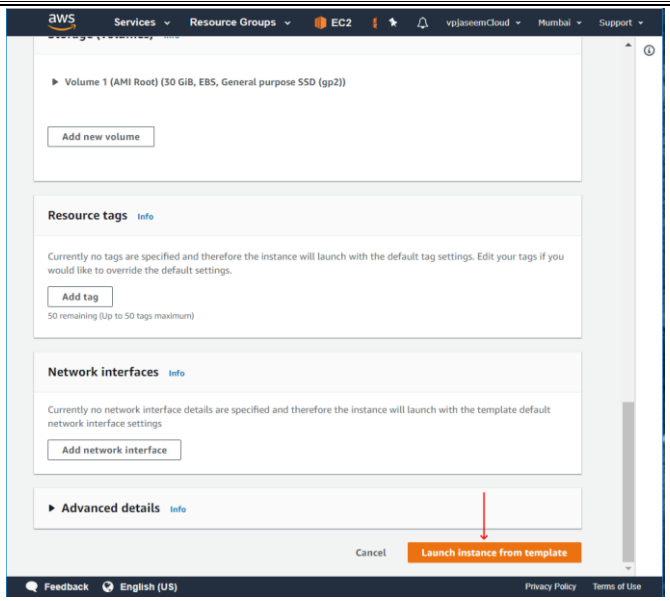




Network settings
Networking platform Info
Virtual Private Cloud (VPC)
Launch into a virtual network in your own logically isolated area within the AWS cloud
EC2-Classic
Launch into a single flat network that you share with other customers
Subnet Info
subnet-05bd2cc86c1e09ec5
ap-south-1-public-subnet-1a
Create new subnet
Security groups Info
Select security groups
web-servers-security-group sg-0348e96684b59cf2a
Storage (volumes) Info
Volume 1 (AMI Root) (30 GiB, EBS, General purpose SSD (gp2))
Add new volume
Resource tags Info
Currently no tags are specified and therefore the instance will launch with the default tag settings. Edit your tags if you
Feedback
English (US)
Privacy Policy
Terms of Use


Volume 1 (AMI Root) (30 GiB, EBS, General purpose SSD (gp2))
Add new volume
Resource tags Info
Currently no tags are specified and therefore the instance will launch with the default tag settings. Edit your tags if you would like to override the default settings.
Add tag
50 remaining (Up to 50 tags maximum)
Network interfaces Info
Currently no network interface details are specified and therefore the instance will launch with the template default network interface settings
Add network interface
Advanced details Info
Cancel
Launch instance from template
Feedback
English (US)
Privacy Policy
Terms of Use

# AWS Simple Storage Service (S3)

## What is AWS Simple Storage Service (S3)

- Amazon Simple Storage Service is storage for the Internet, A cloud storage (like Google Drive, Drobox, etc.)
- Amazon S3 has a simple web services interface that you can use to store and retrieve any amount of data, at any time, from anywhere on the web.
- To upload your data to Amazon S3, you must first create an S3 bucket in one of the AWS Regions. Cannot be nested (1 bucket inside another bucket)
- We can create 100 buckets, each bucket can store unlimited data
- Buckets are the top-level containers for data, bucket name has to be globally unique.
- It's an object-based storage, whatever file we upload to S3, we get a web link to access it via HTTP(s). Individual file size between 0 and 5TB
- Also available via REST or SOAP
- Build for unstructured binary data (basically files, large files), it is not a database.
- Objects can be versioned (same file upload twice, AWS will keep both)
- To access any object, bucket+object+version(optional) – https://bucket.s3.amazonaws.com/objectname.mp4
- Highly scalable, Durable and Unlimited.
- Secure with client side or AWS Server side encryption
- Object and bucket level logs
- We can store Application data, photos, videos, documents, AMI, a complete website in S3
- Cost of S3 is relatively cheap, pay for only wat you use, uploading data to S3 is free, download is chargeable (hourly charge)
- Access Control for the Bucket and Object to secure S3 (ACL - Access Control List: User to Permission mapping, IAM - Identify and Access Management policies)
- Bucket and Object policies can be defined with Access Policy Language (JSON Structure)
- Encrypt at REST with AES-256 or encrypt before sending.

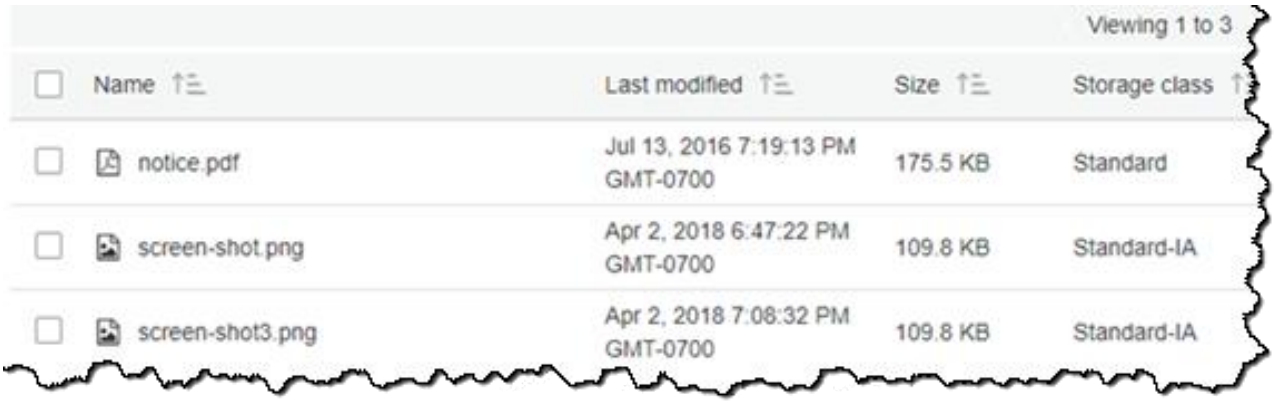
AWS Reference: What is Amazon S3?

## S3 Object Versioning

- Allows you to retain one or more versions of individual object (same file upload after some edits, S3 will keep latest as well as old)
- Versioning is enabled at the bucket level
- Existing objects have 'Null' version, new objects assigned with new version ID
- We can configure life cycle policy to automatically delete unwanted versions
- https://bucket.s3.amazonaws.com/objectname.mp4?versionId=XXXXXXXX

## Storage Classes

Each object in Amazon S3 has a storage class associated with it. For example, if you list the objects in an S3 bucket, the console shows the storage class for all the objects in the list.




| Class | Features |
|---|---|
| S3 Standard: | • Frequently accessed data<br>• 99.999999999% durability<br>• 99.99% availability,<br>• >= 3 AZs |
| S3 Standard-Infreq Access (IA) | • Long-lived, infrequently accessed data<br>• 99.999999999% durability<br>• 99.99% availability,<br>• >= 3 AZs<br>• 30 days minimum storage duration<br>• 128 KB minimum billable object size<br>• Per GB retrieval fees apply. |
| S3 Intelligent-Tiering | • Long-lived data with changing or unknown access patterns<br>• 99.999999999% durability<br>• 99.99% availability,<br>• >= 3 AZs<br>• 30 days minimum storage duration<br>• Monitoring and automation fees per object apply. No retrieval fees. |
| S3 One Zone-IA | • Long-lived, infrequently accessed, non-critical data<br>• 99.999999999% durability<br>• 99.5% availability,<br>• 1 AZs<br>• 30 days minimum storage duration<br>• 128 KB minimum billable object size<br>• Per GB retrieval fees apply. Not resilient to the loss of the Availability Zone. |

| | |
|---|---|
| S3 Glacier | • Long-term data archiving with retrieval times ranging from minutes to hours<br>• 99.999999999% durability<br>• 99.99% availability,<br>• >=3 AZs<br>• 90 days minimum storage duration<br>• 40 KB minimum billable object size<br>• Per GB retrieval fees apply.<br>• Glacier basically a different entity, to access data from Glacier, you need to restore the data to S3 and then use it |
| S3 Glacier Deep Archive | • Archiving rarely accessed data with a default retrieval time of 12 hours<br>• 99.999999999% durability<br>• 99.99% availability,<br>• >=3 AZs<br>• 180 days minimum storage duration<br>• 40 KB minimum billable object size<br>• Per GB retrieval fees apply. You must first restore archived objects before you can access them. |
| RRS – Reduce Redundancy (Not recommended) | • Frequently accessed, non-critical data<br>• 99.99% durability<br>• 99.99% availability,<br>• >=3 AZs |

AWS Reference: Amazon S3 Storage Classes

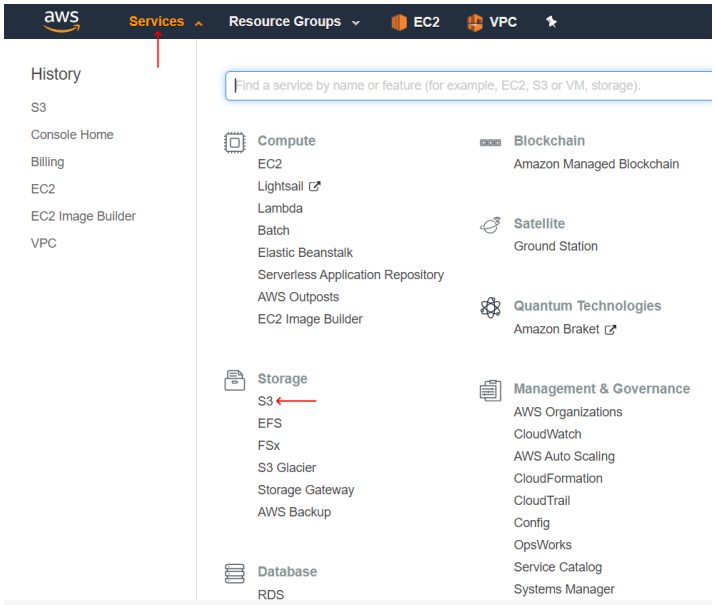
## S3 Life Cycle Management


- Versioning and storage classes are tied together using Life Cycle Management.
- Allows us to specify rules / policies to automatically manage the life cycle of object
- It happens in Object level or Bucket level
- An S3 Lifecycle configuration is a set of rules that define actions that Amazon S3 applies to a group of objects.
- **Transition actions**—Define when objects transition to another storage class. For example, you might choose to transition objects to the S3 Standard-IA storage class 30 days after you created them, or archive objects to the S3 Glacier storage class one year after creating them.
- **Expiration actions**—Define when objects expire. Amazon S3 deletes expired objects on your behalf.

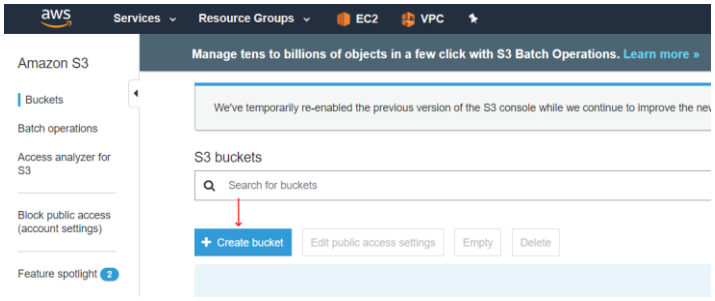
```
<LifecycleConfiguration>
    <Rule>
        <ID>sample-rule</ID>
        <Prefix>documents/</Prefix>
        <Status>Enabled</Status>
        <Transition>
           <Days>365</Days>
           <StorageClass>GLACIER</StorageClass>
        </Transition>
        <Expiration>
             <Days>3650</Days>
        </Expiration>
    </Rule>
</LifecycleConfiguration>
```

## [LAB] AWS Simple Storage Service S3 Configuration

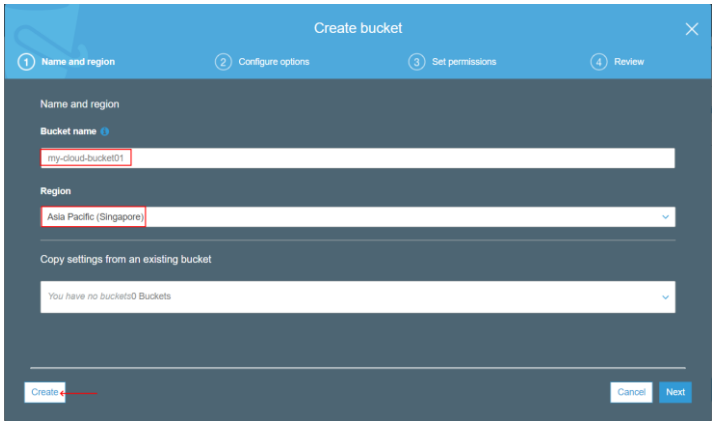
Now let's create a bucket in S3 and upload some files




- In AWS Console >> Services >> S3

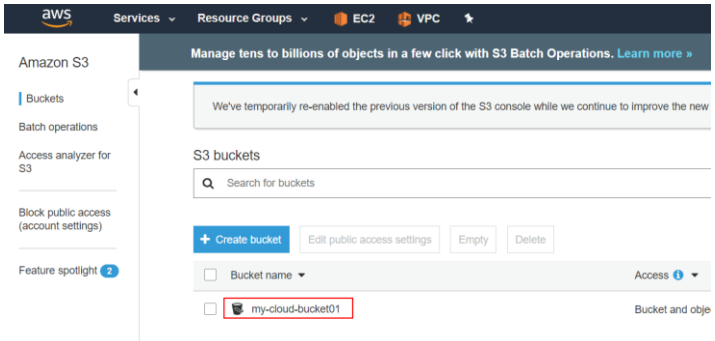



>> Create Bucket

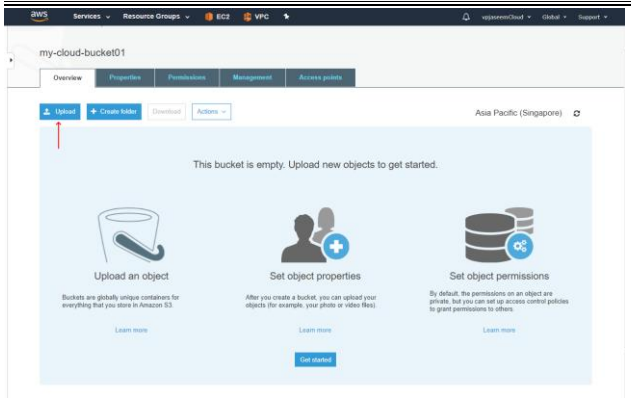



>> Bucket Name (This has to be unique) >> Region (Select) >> Create

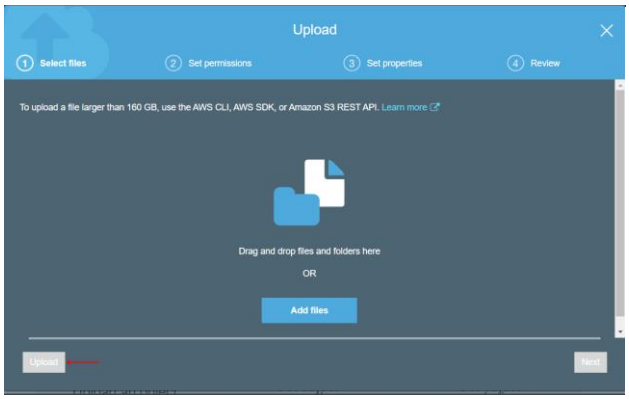
Note: We can do many customizations here (Config options, permissions, etc.), but we will deal with those later.



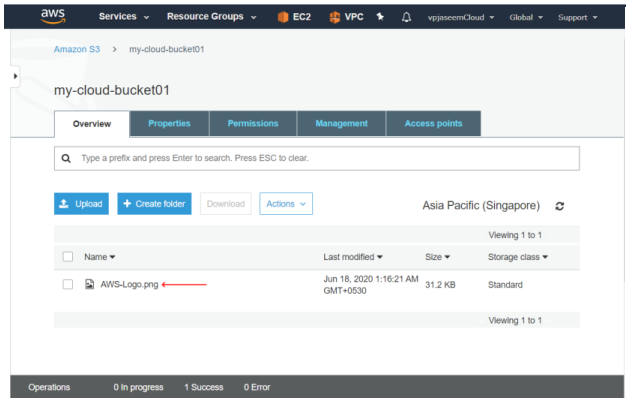

You can see the bucket that you crated. Go ahead and open it.



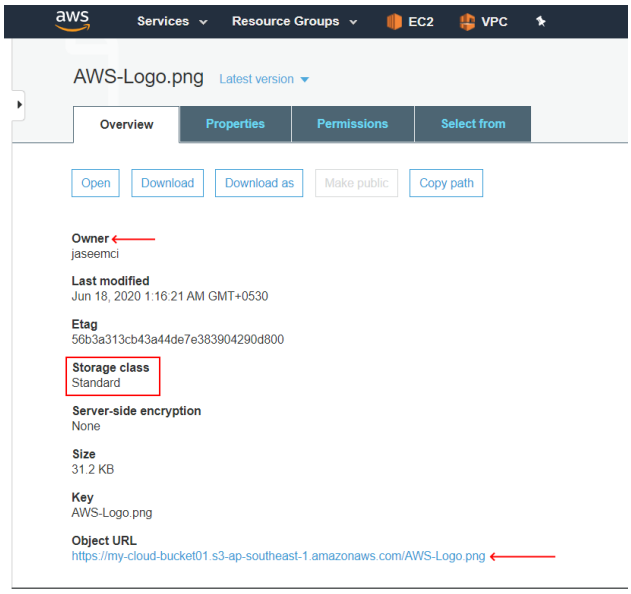

>> Upload button to upload new files (This is just like any other cloud storage e.g. Google Drive)



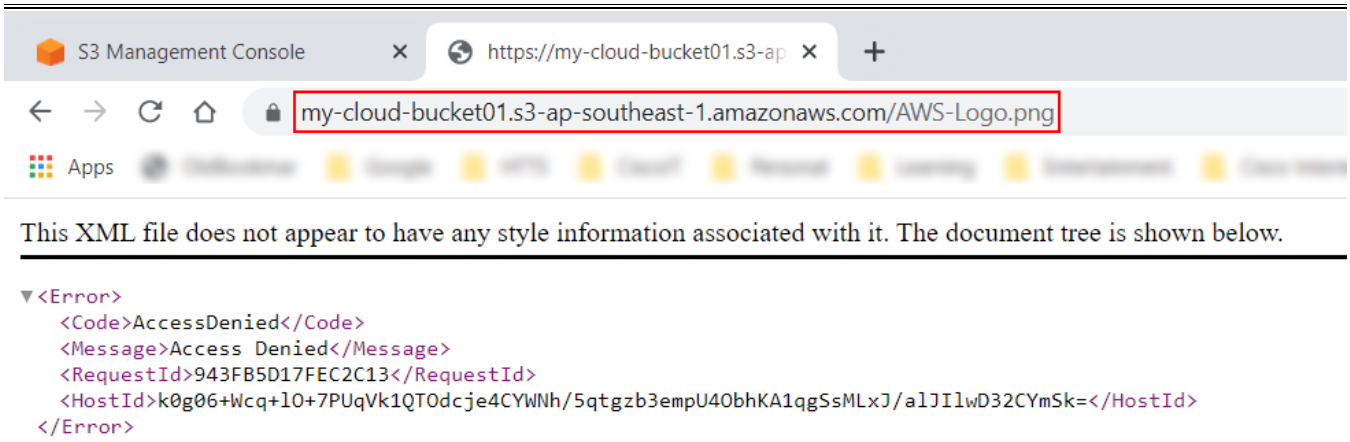

While uploading the file, you can do much more customization (Storage Class, Permissions, Properties, etc.), but we will deal with those later.




I have uploaded a file called ‘AWS-Logo.png’



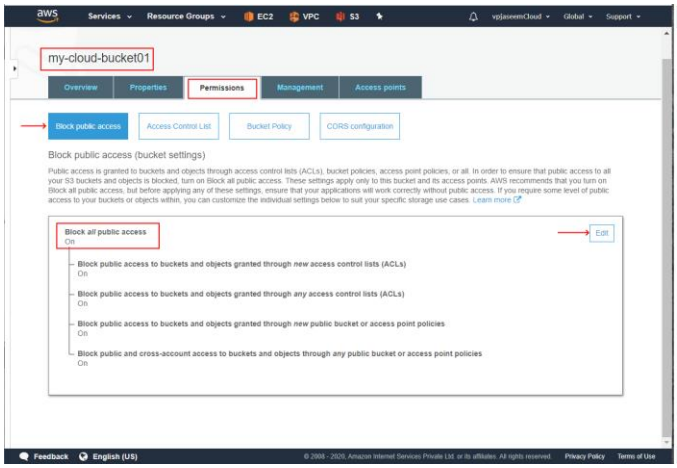

We can see the Storage Class, Public URL to access the object here.
You can only use ‘Open’ button to access the file.
We don’t get access to Object URL until you make it as public.

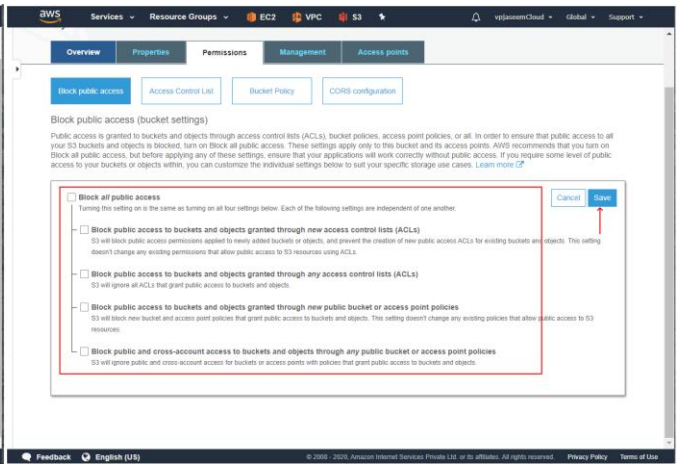



I tried to access the Object URL and above is the error that I got. This is common. How to get anonymous access to the Object URL? Will see in next topic.

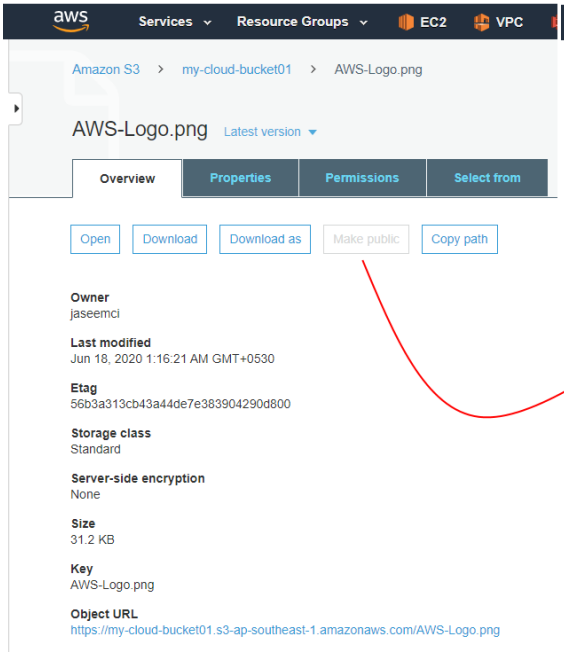
## [LAB] S3 Bucket and Object Permissions Method 1

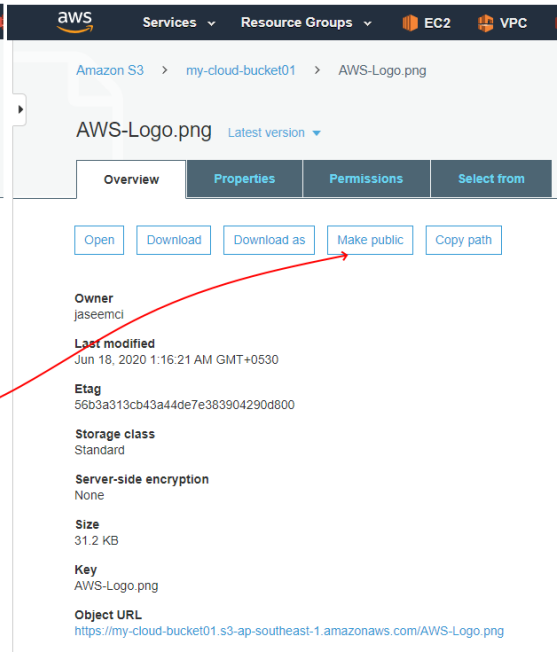
If we directly try to access an object URL, we may get the 'Access Denied' error. Well, how to set permissions and public access to S3 object?

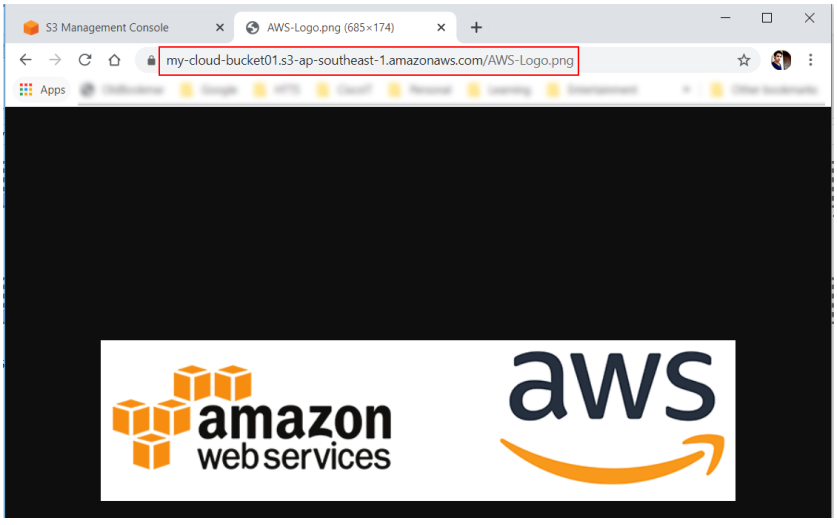


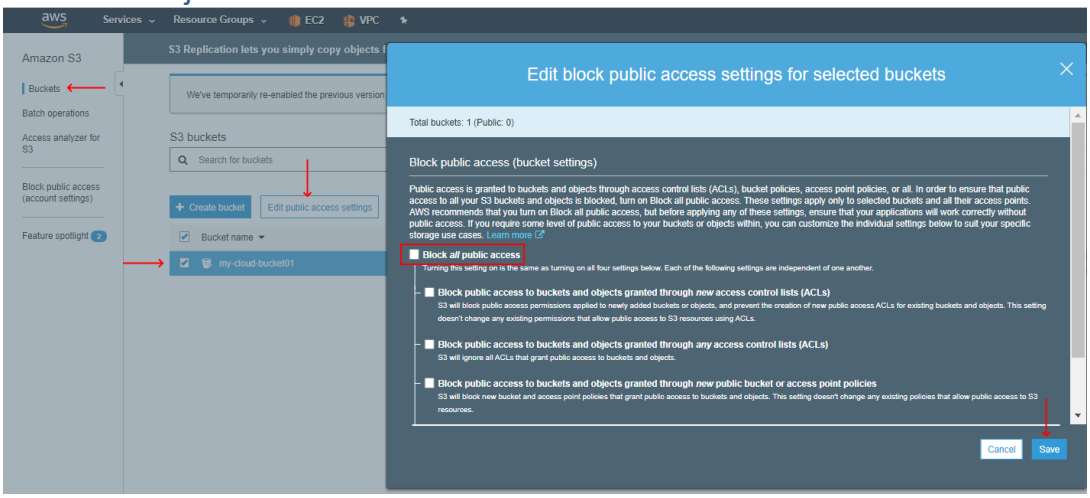

- Go to Bucket >> Permissions >> Block Public Access >> Edit >> Uncheck Block *all* public access
- Now we can enable public access to the objects in the bucket

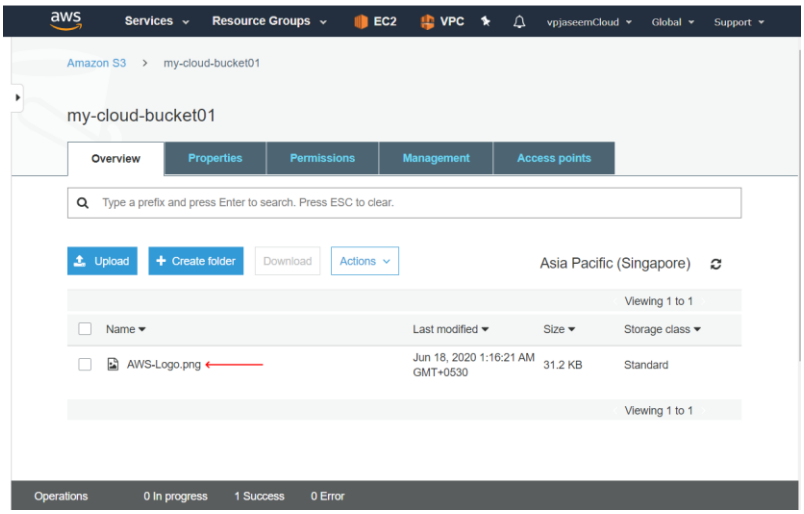


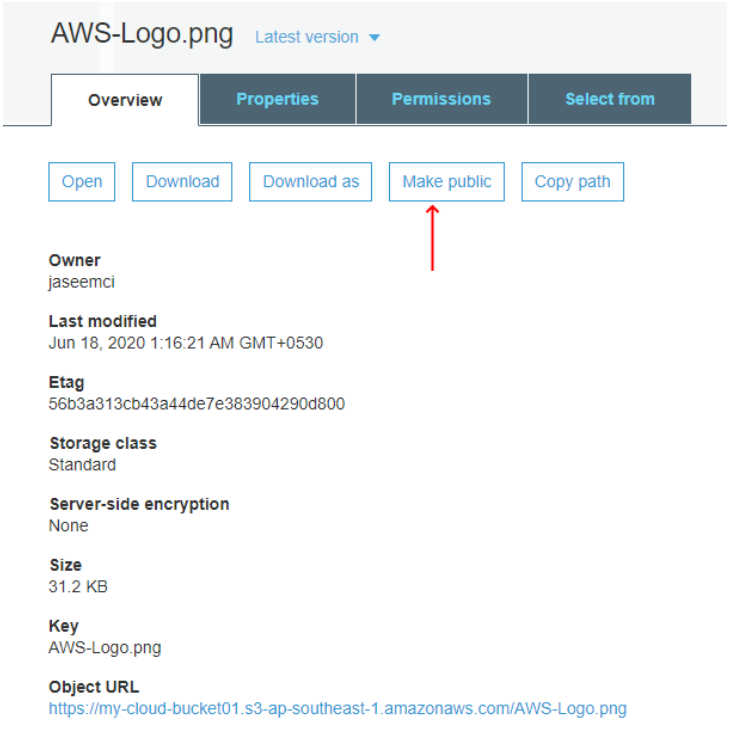



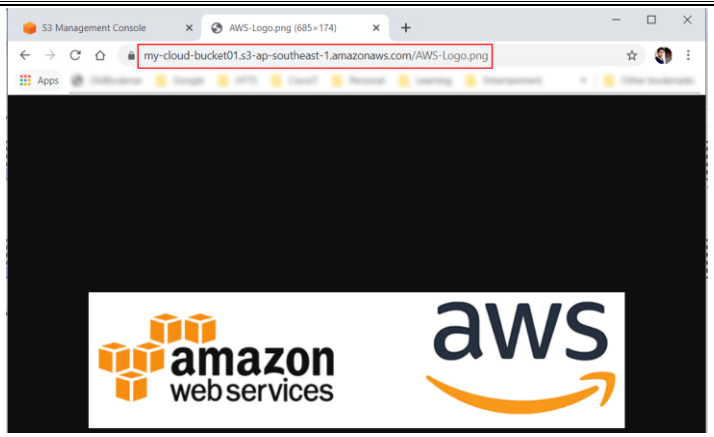



# [LAB] S3 Bucket and Object Permissions Method 2

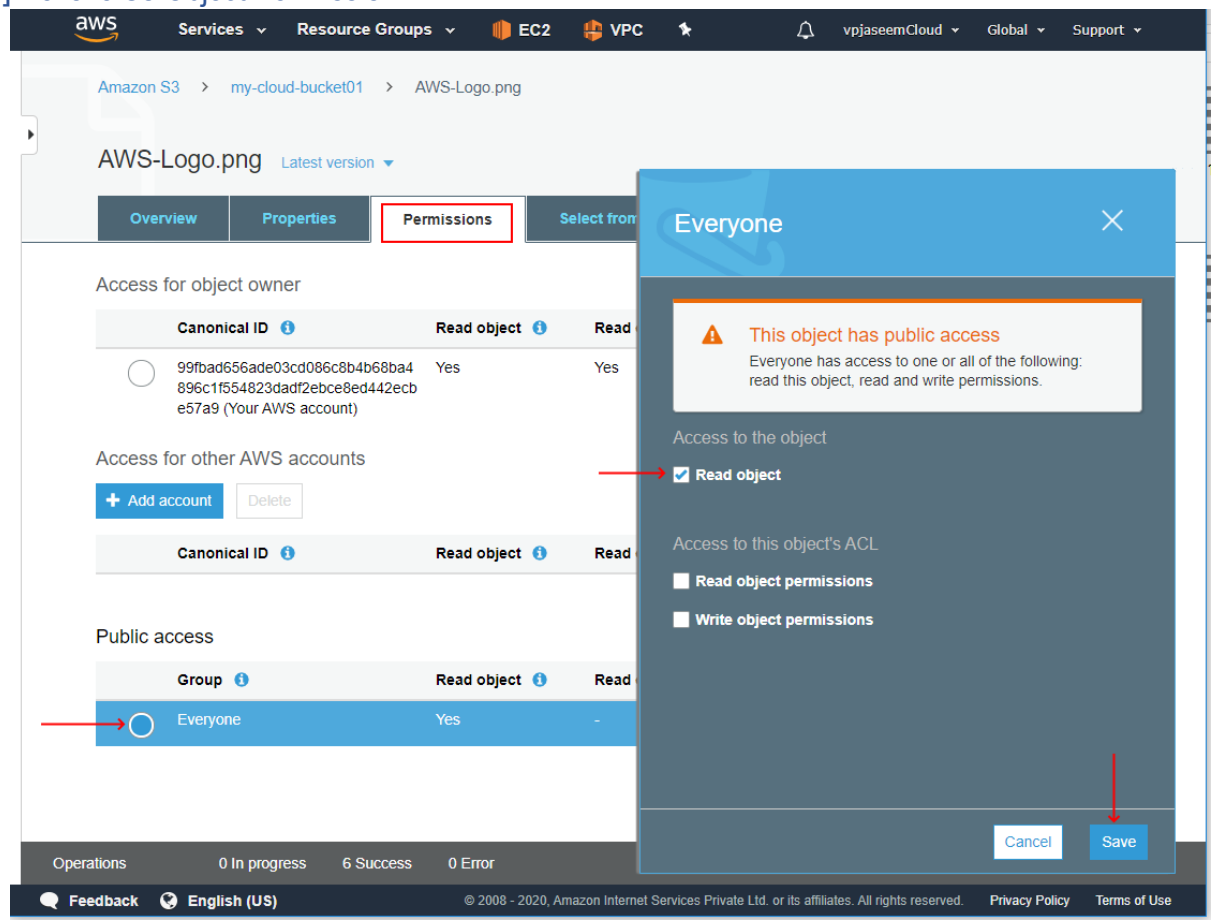



S3 Management Console
AWS-Logo.png (685 × 174)
my-cloud-bucket01.s3-ap-southeast-1.amazonaws.com/AWS-Logo.png
Apps
amazon
web services
aws

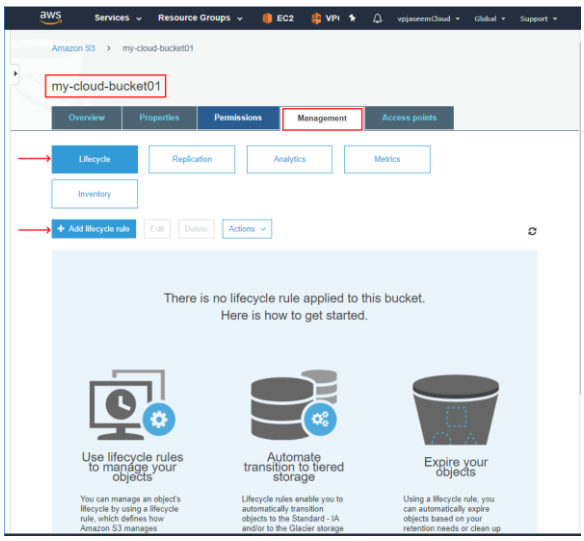
# [LAB] Revoke S3 Object Permission

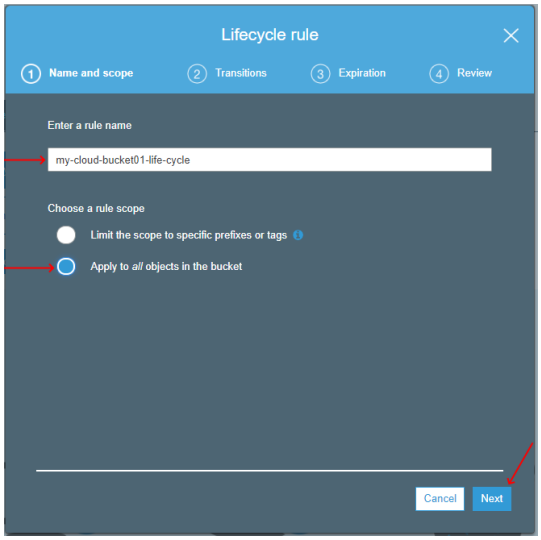


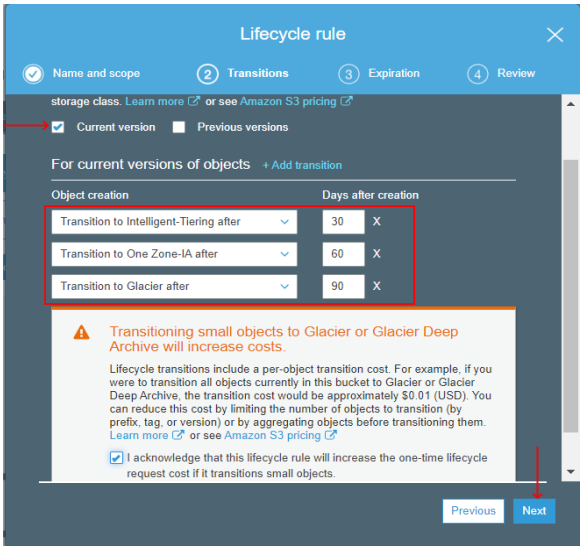

>> Select the Object >> Permissions >> Everyone >> Uncheck the Read Object

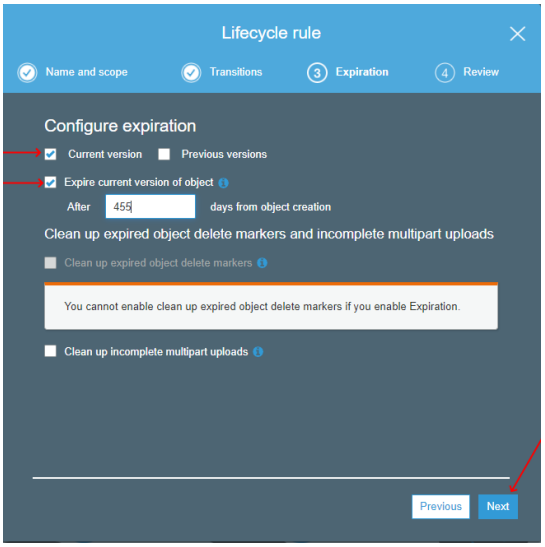
## [LAB] Configuring S3 Life Cycle Management

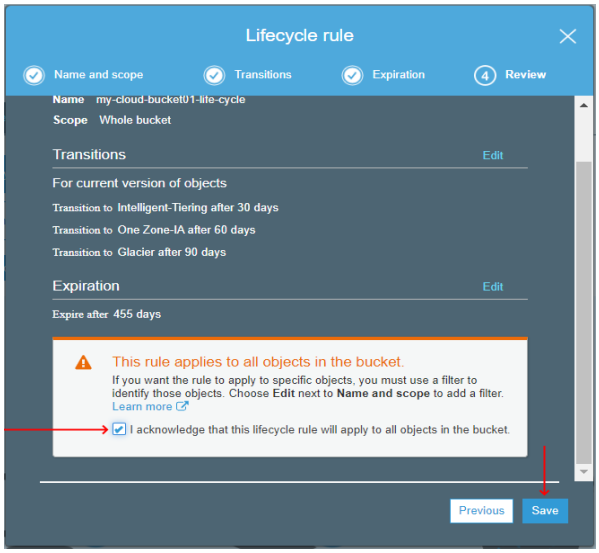
- To manage your objects so that they are stored cost effectively throughout their life cycle.
- Life Cycle configuration is a set of rules that define actions that AWS S3 applies to group of objects.
- **Transition Action**: Define when object transition to another class
- **Expiration Action**: Define when object expires, S3 delete the expired object on your behalf

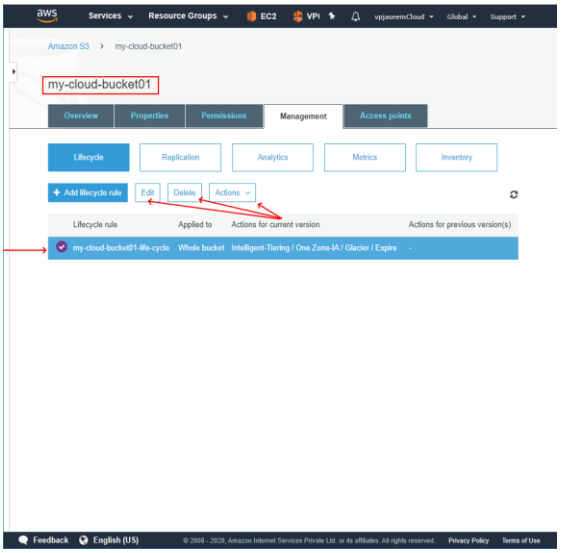


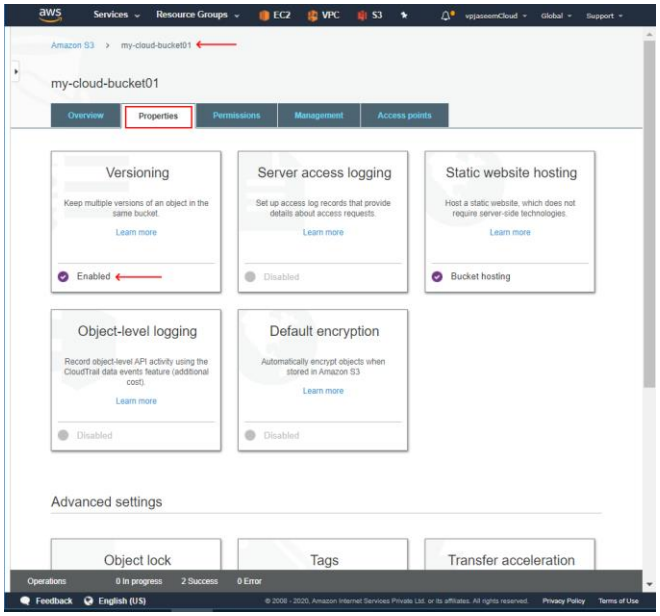



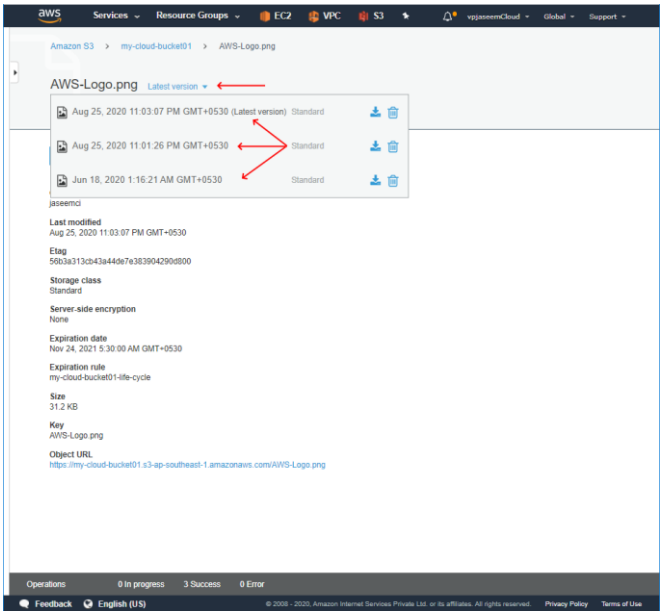



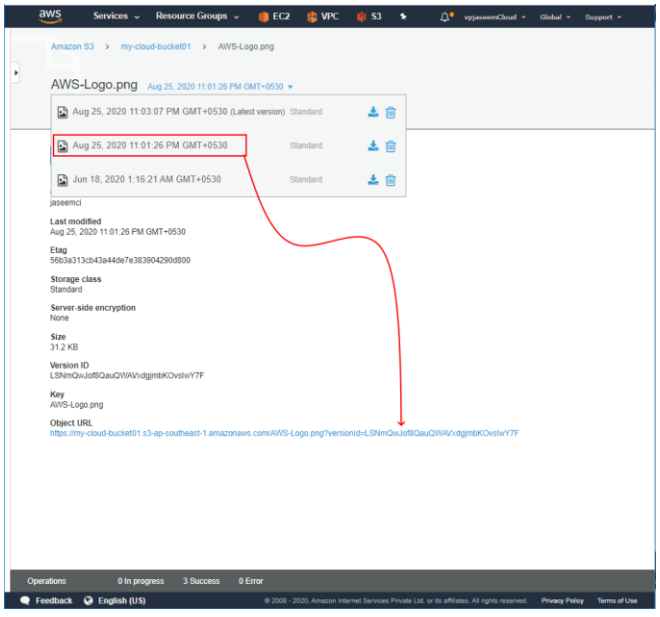



## [LAB] S3 Versioning

- Versioning allows to upload same file name multiple times, S3 will keep all the files with version ID.
- Versioning is a means of keeping multiple variants of an object in the same bucket.
- You can use versioning to preserve, retrieve, and restore every version of every object stored in your Amazon S3 bucket.

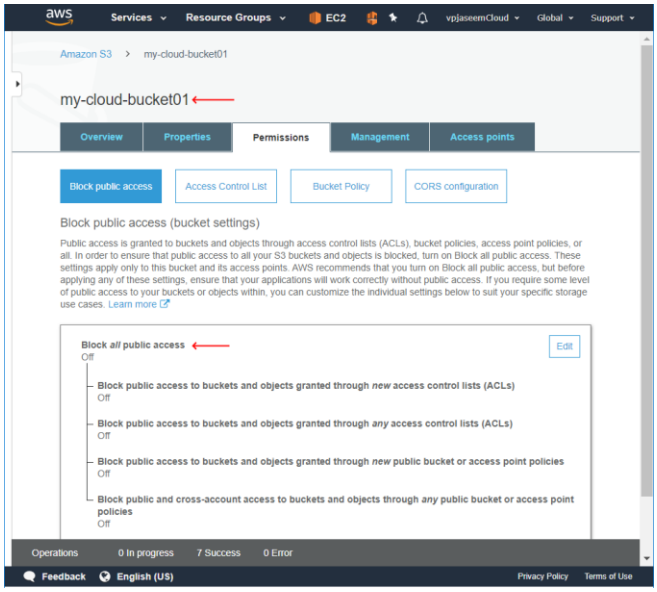


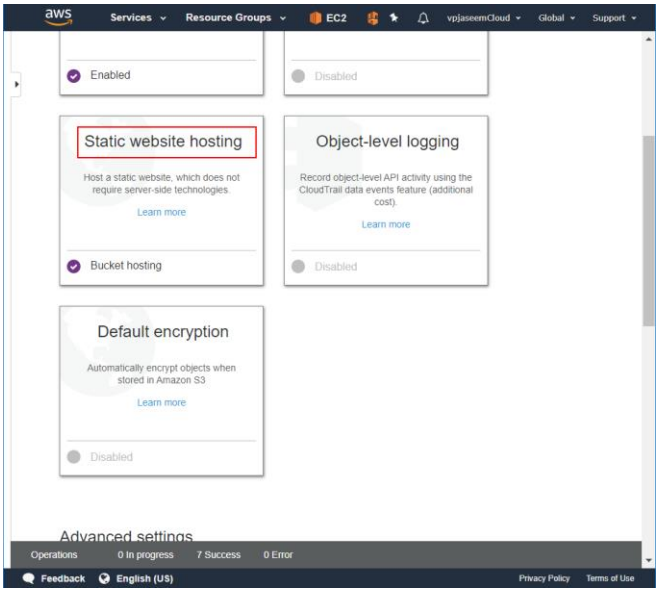



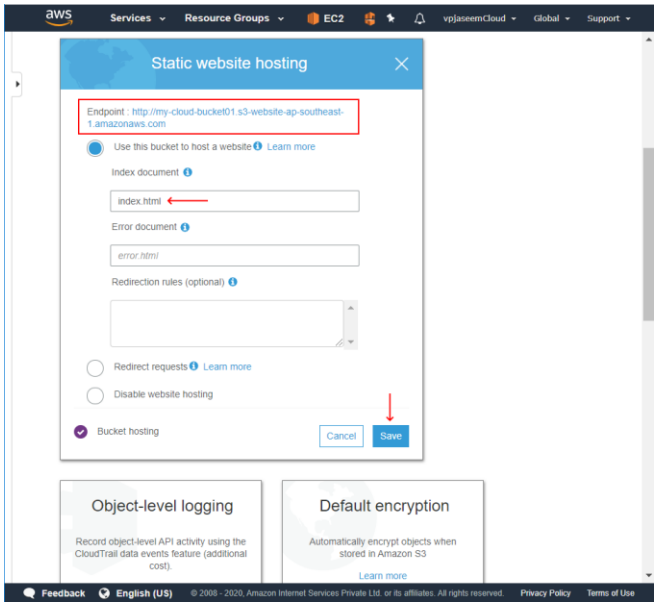



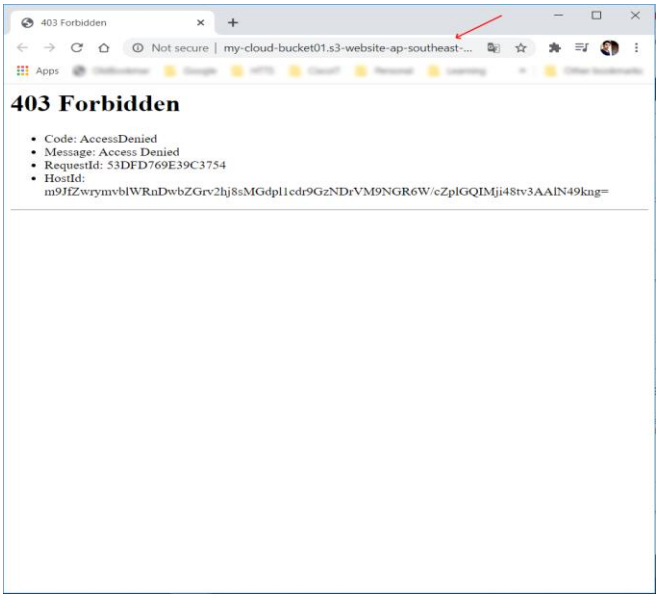

AWS Reference: S3 Versioning

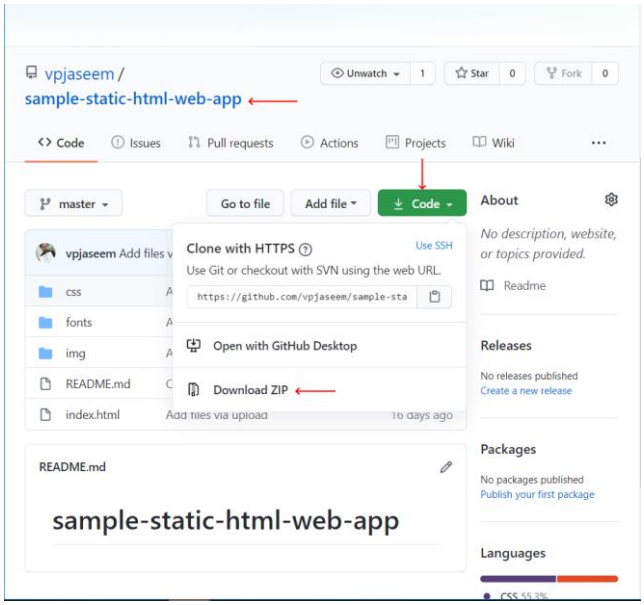
# [LAB] Host a Web Site in S3

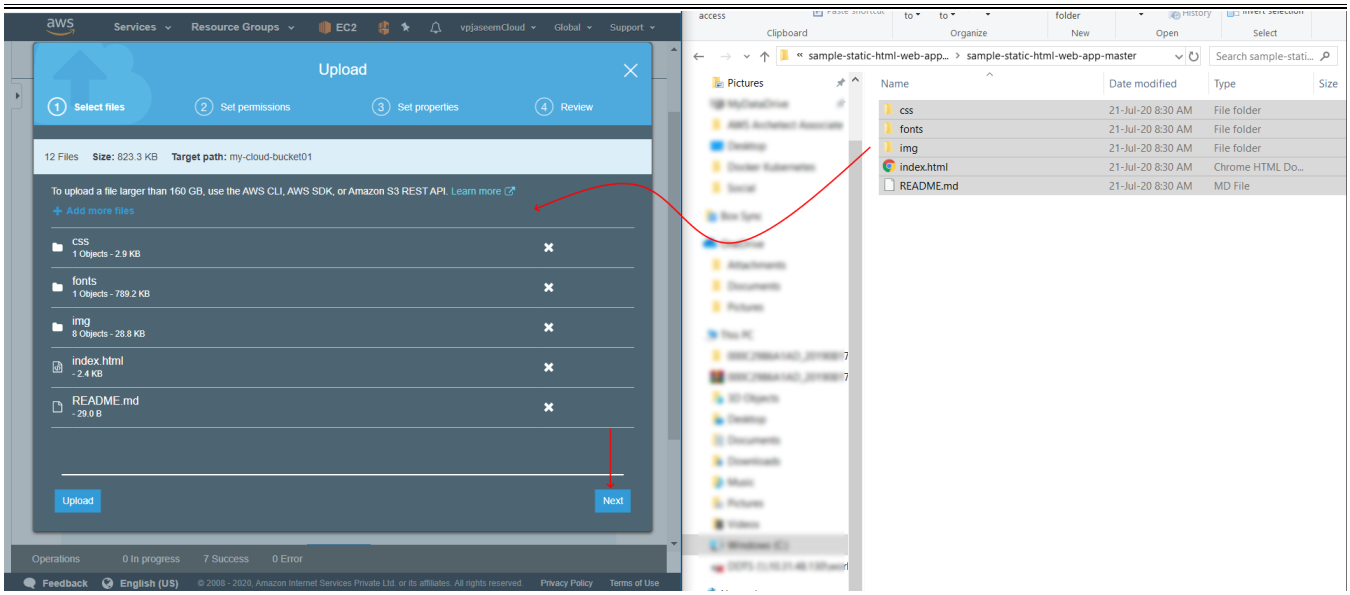


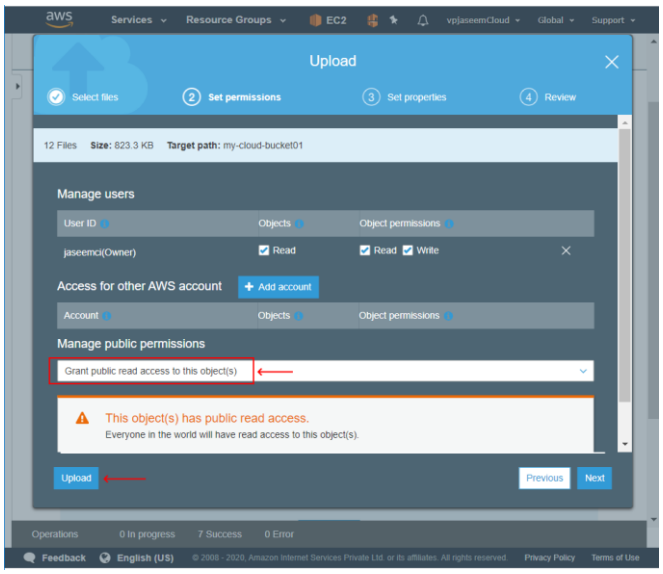



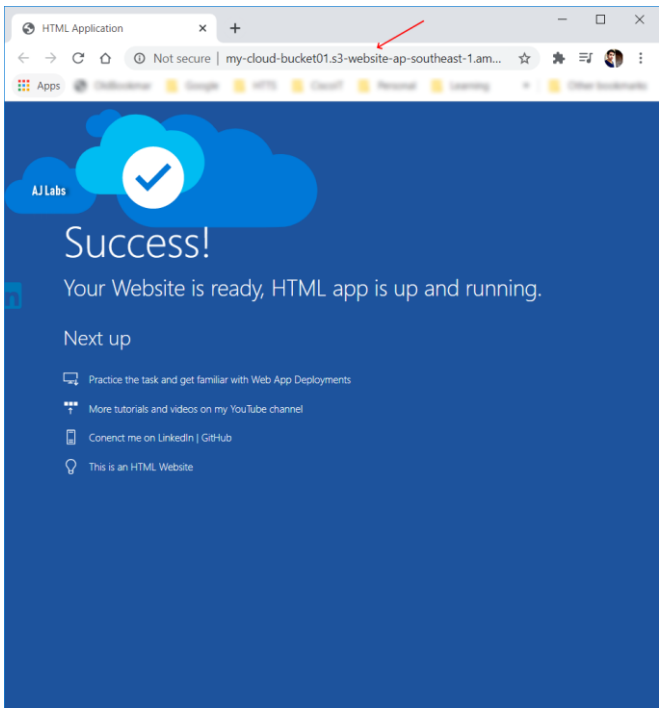




Upload
Select files
Set permissions
Set properties
Review
12 Files Size: 823.3 KB Target path: my-cloud-bucket01
To upload a file larger than 160 GB, use the AWS CLI, AWS SDK, or Amazon S3 REST API. Learn more
Add more files
css
1 Objects - 2.9 KB
fonts
1 Objects - 789.2 KB
img
8 Objects - 28.8 KB
index.html
- 2.4 KB
README.md
- 29.0 B
Upload
Next
Operations 0 In progress 7 Success 0 Error
Feedback English (US)
Privacy Policy Terms of Use
sample-static-html-web-app... > sample-static-html-web-app-master
Name Date modified Type Size
css 21-Jul-20 8:30 AM File folder
fonts 21-Jul-20 8:30 AM File folder
img 21-Jul-20 8:30 AM File folder
index.html 21-Jul-20 8:30 AM Chrome HTML Do...
README.md 21-Jul-20 8:30 AM MD File


Upload
Select files
Set permissions
Set properties
Review
12 Files Size: 823.3 KB Target path: my-cloud-bucket01
Manage users
User ID
Objects
Object permissions
jaseemci(Owner)
Read
Read Write
Access for other AWS account
Add account
Account
Objects
Object permissions
Manage public permissions
Grant public read access to this object(s)
This object(s) has public read access.
Everyone in the world will have read access to this object(s).
Upload
Previous
Next
Operations 0 In progress 7 Success 0 Error
Feedback English (US)
Privacy Policy Terms of Use


Make public
Selection: 2 Objects, 3 Folders
Total size: 823.3 KB
Total objects: 12
css/
1 Objects - 2.9 KB
fonts/
1 Objects - 789.2 KB
img/
8 Objects - 28.8 KB
README.md
- 29.0 B
index.html
- 2.4 KB
Everyone will have access to one or all of the following: read this object, read and write permissions.
Cancel
Make public


HTML Application
Not secure | my-cloud-bucket01.s3-website-ap-southeast-1.am...
Apps
AJ Labs
Success!
Your Website is ready, HTML app is up and running.
Next up
Practice the task and get familiar with Web App Deployments
More tutorials and videos on my YouTube channel
Conenct me on LinkedIn | GitHub
This is an HTML Website

## AWS Snow Family

- Deploying AWS managed hardware and software to locations outside AWS Regions and even beyond AWS Outposts.
- These services help physically transport up to exabytes of data into and out of AWS.
- Used to carry data locally from outside to AWS DC.

### AWS Snowcone

- Smallest member of the AWS Snow Family of edge computing and data transfer devices.
- Snowcone is portable, rugged, and secure. You can use Snowcone to collect, process, and move data to AWS, either offline by shipping the device, or online with AWS DataSync.

### AWS Snowball

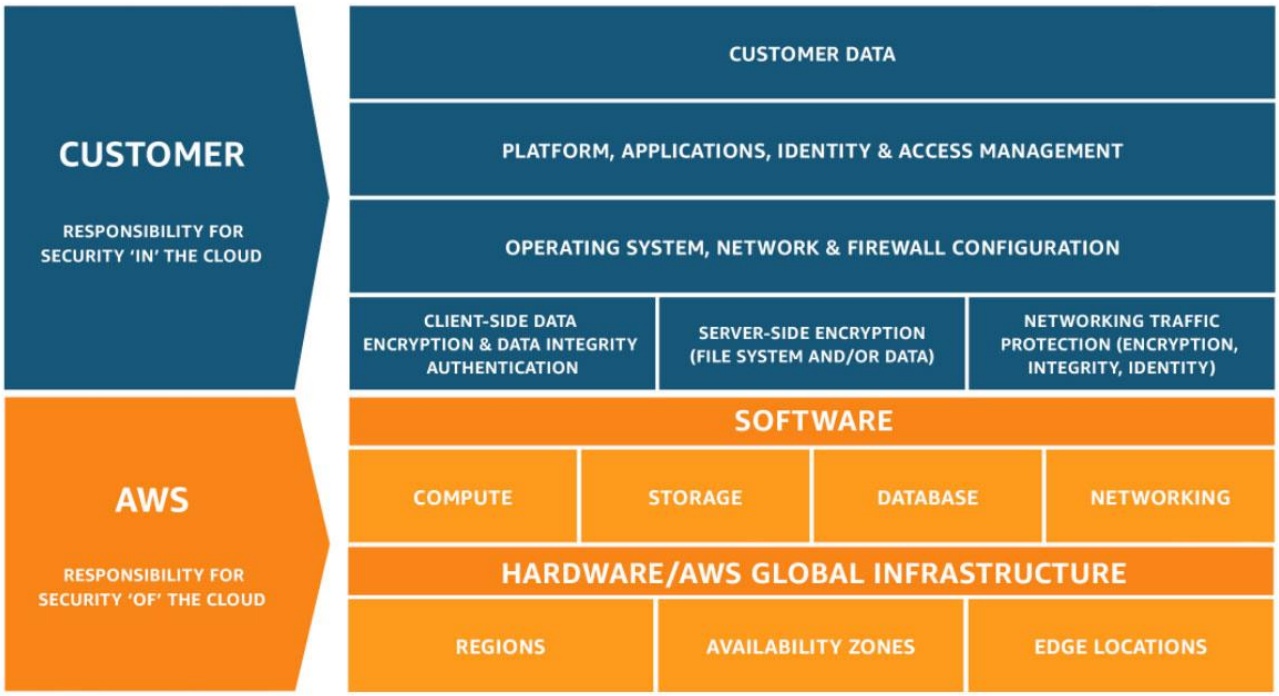

- AWS Snowball is a data migration and edge computing device that comes in two device options: Compute Optimized and Storage Optimized.

## AWS Snowmobile


- AWS Snowmobile moves up to 100 PB of data in a 45-foot long ruggedized shipping container and is ideal for multi-petabyte or Exabyte-scale digital media migrations and data center shutdowns.
- A Snowmobile arrives at the customer site and appears as a network-attached data store for more secure, high-speed data transfer.
- After data is transferred to Snowmobile, it is driven back to an AWS Region where the data is loaded into Amazon S3.

AWS Reference: Snow Family overview

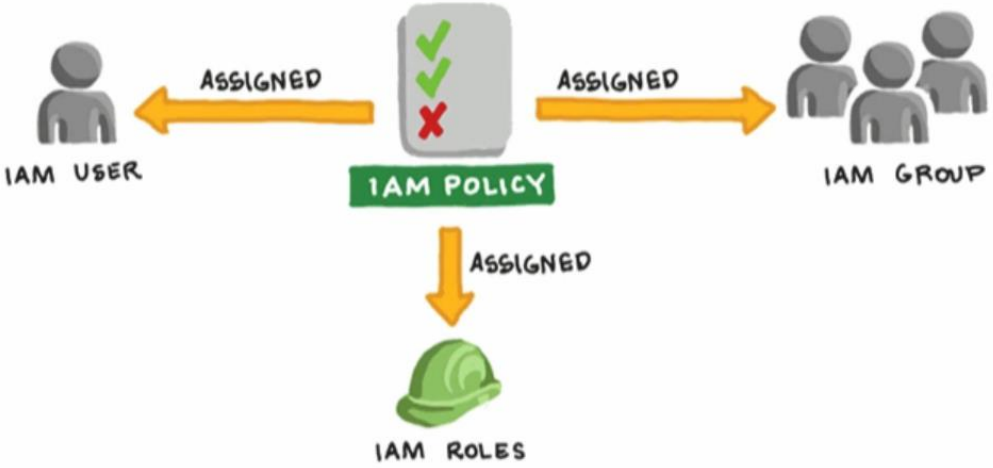
## AWS Shared Responsibility Model




- Who responsible for what?
- What AWS is responsible for securing and what we responsible for securing our resources
- AWS is responsible for Physical Security, CPU, Memory, Storage, Networking resources, Hypervisor, etc.
- We are responsible for Firewall, Security Groups, System updates
- AWS DCs are undisclosed locations, we can't identify it, 24/7 security guard, 2FA for entry,
- Automated Change Control, Bastion Server to configure any devices, Abnormal activity detection
- Regulatory compliance: https://aws.amazon.com/compliance/programs/

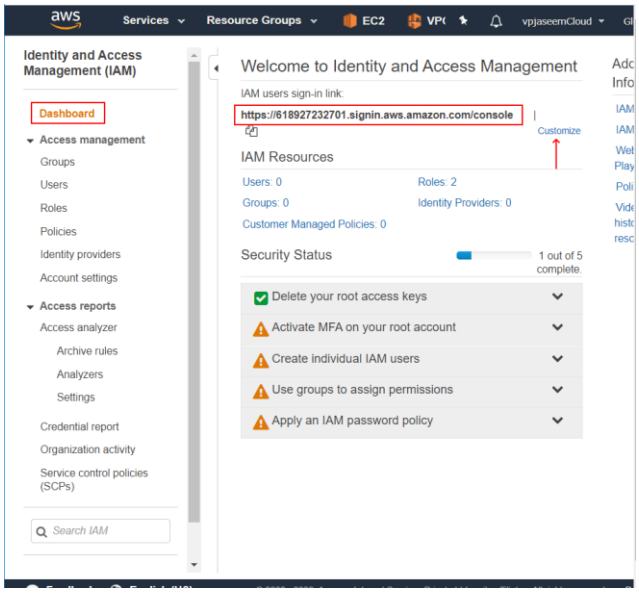
## Identity and Access Management (IAM)

- We do not use the 'root' user to perform day today activities.
- We create separate users with respective privileges.
- IAM allow us to create Users, assign permission to users, Groups and Roles, assign permission to groups and Roles.

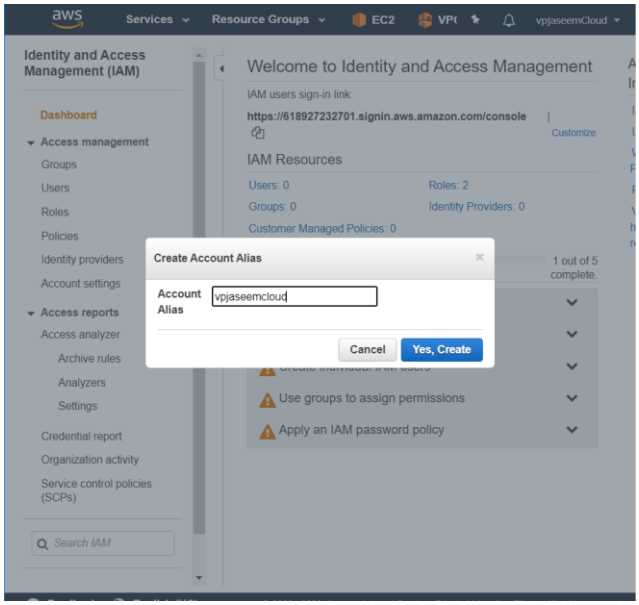



- Provide a separate URL for IAM users to login.
- Also allows Multi Factor Authentication (MFA)
- IAM gives 2 key pairs (Access Key and Secret Key) to access AWS via API
- IAM policies are applied to User / Group / Roles and policies are written in JSON format.
- AWS provides Policy Generator (JSON format generator)
- Resources are defined with ARN (Amazon Resource Name)
- There are three different policies,
- **Managed Policies**: AWS provides for us (e.g. S3 Administrator)
- **Custom Policies**: We can create
- **Inline Policies**: They are custom policies designed and tied with resources, if we delete the resource, policy goes away.

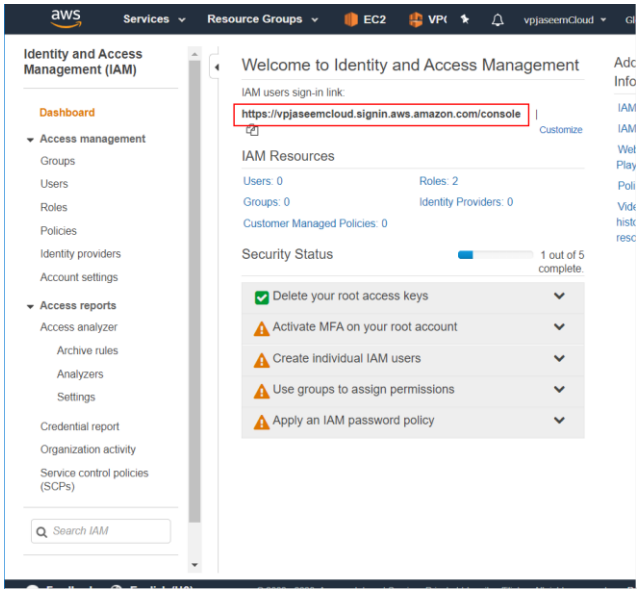
## [LAB] IAM Lab - Customize the IAM User Login URL (Account Alias)




- All the IAM users will use a separate URL to login to the AWS console.
- https://ACCOUNT_ID.sigin.aws.com/console
- We can customize the account ID part to something else.
- Click the Customize button

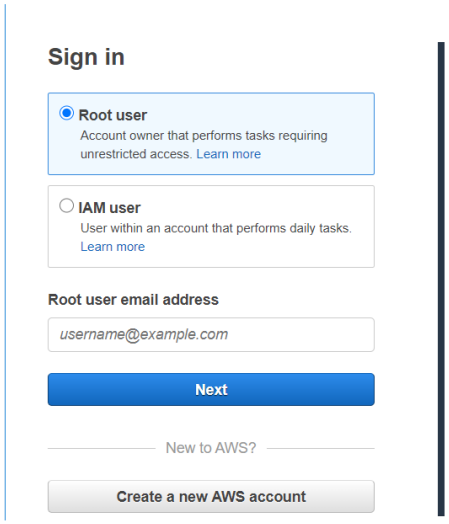



- Enter an Account Alias name

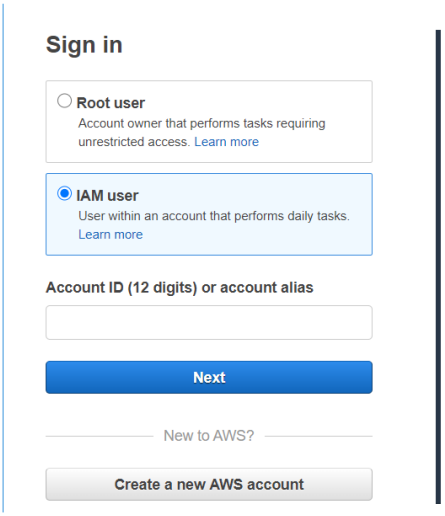



- My custom login URL is https://vpjaseemcloud.signin.aws.amazon.com/console

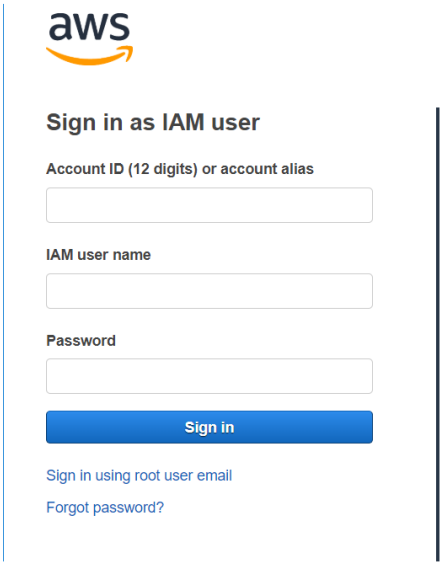
signin.aws.amazon.com



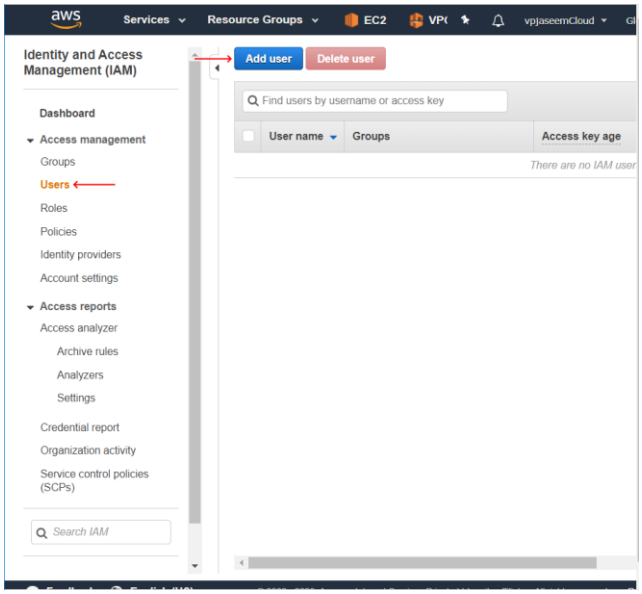

signin.aws.amazon.com

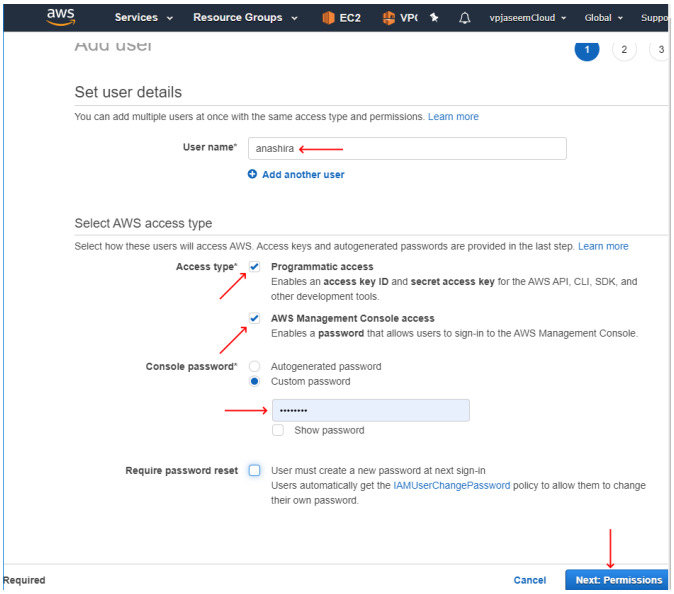


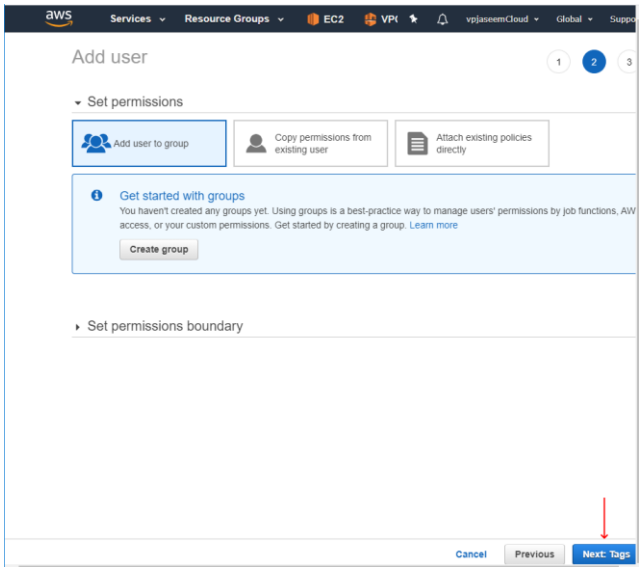

vpjaseemcloud.signin.aws.amazon.com/console

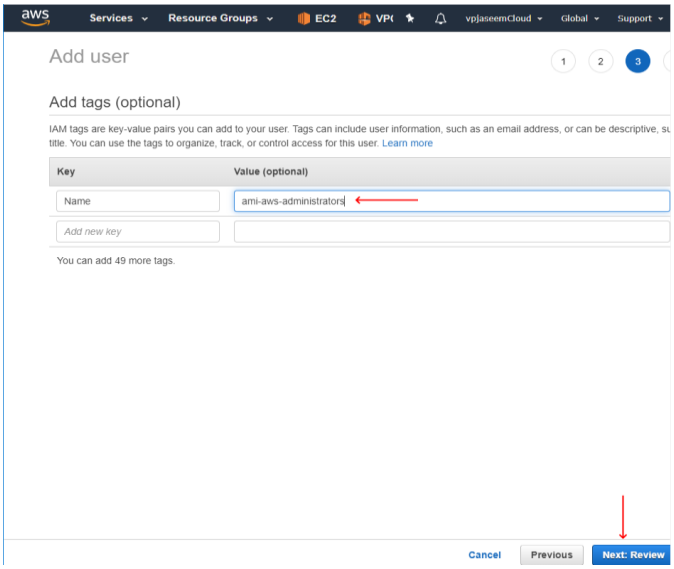


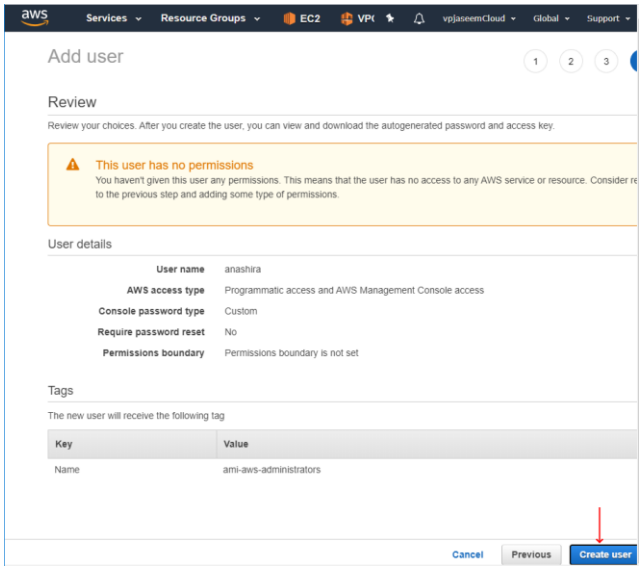
# [LAB] Configure IAM User

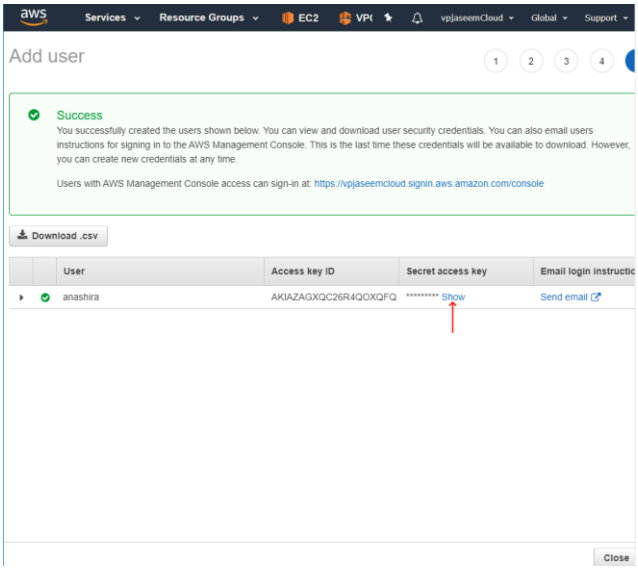


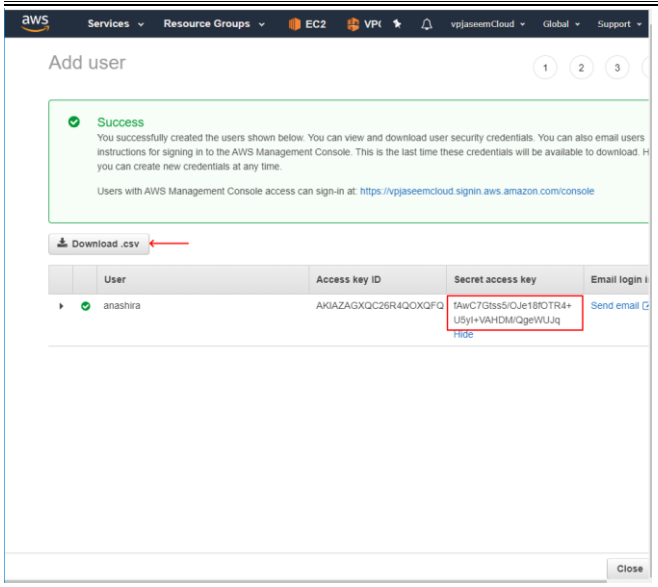



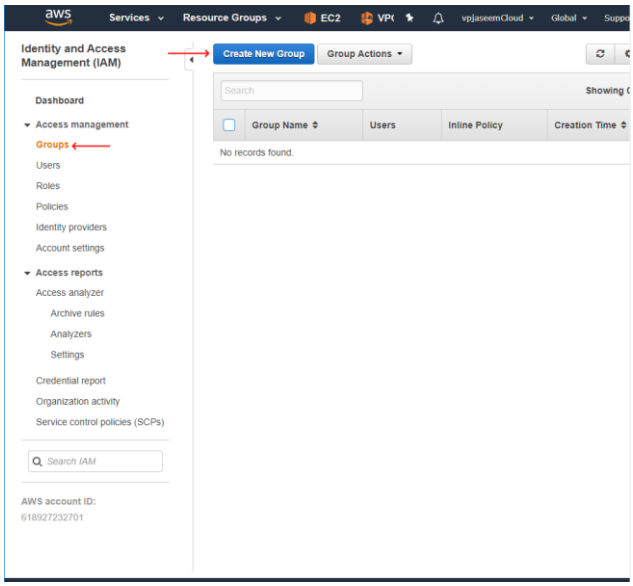



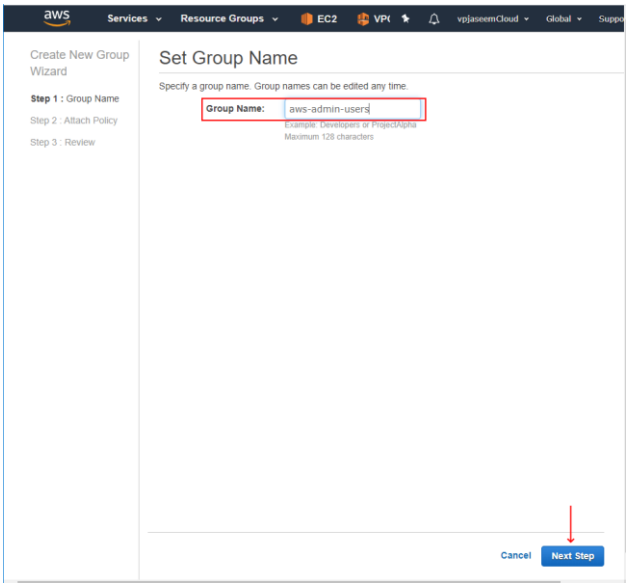



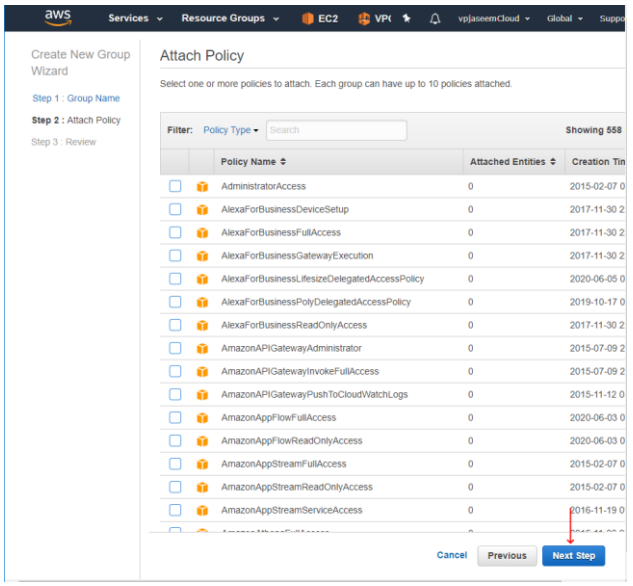


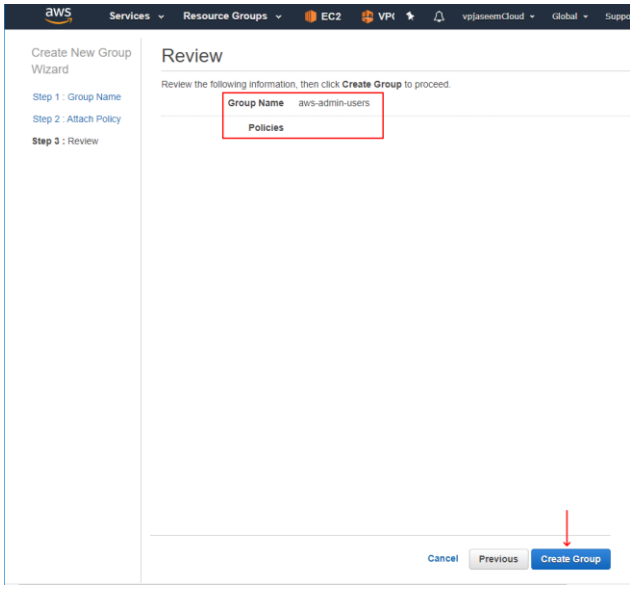

- We just created an IAM user
- Secret Access key is used to sign in to AWS via REST API or CLI (we will cover that later)

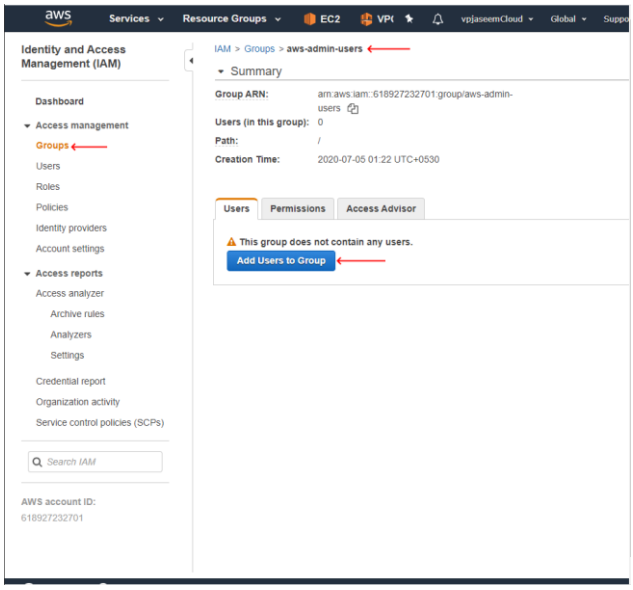
# [LAB] Cerate IAM Group

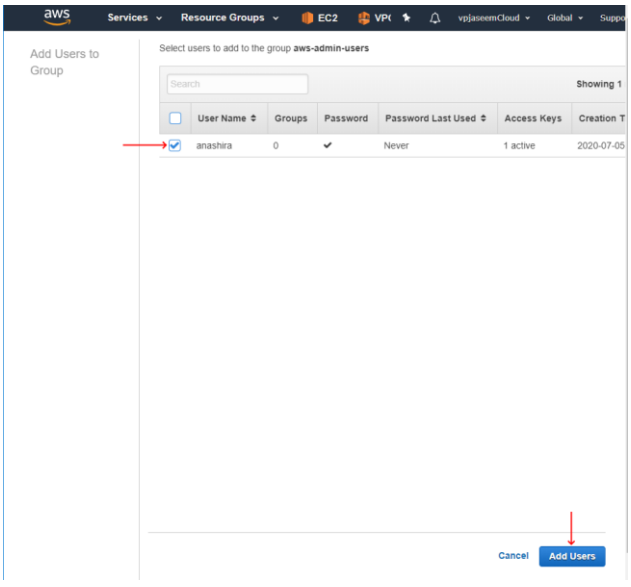


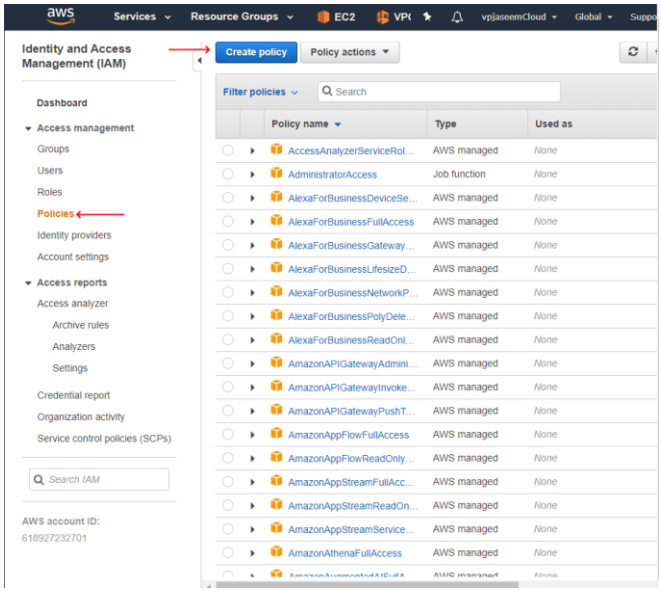



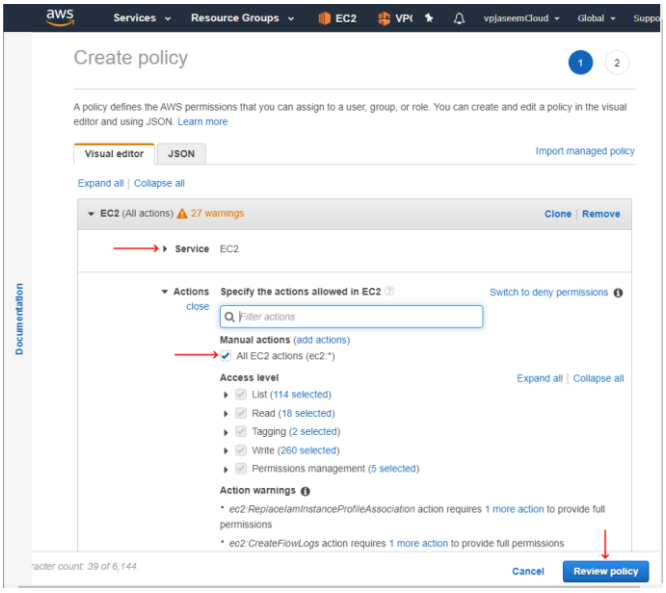



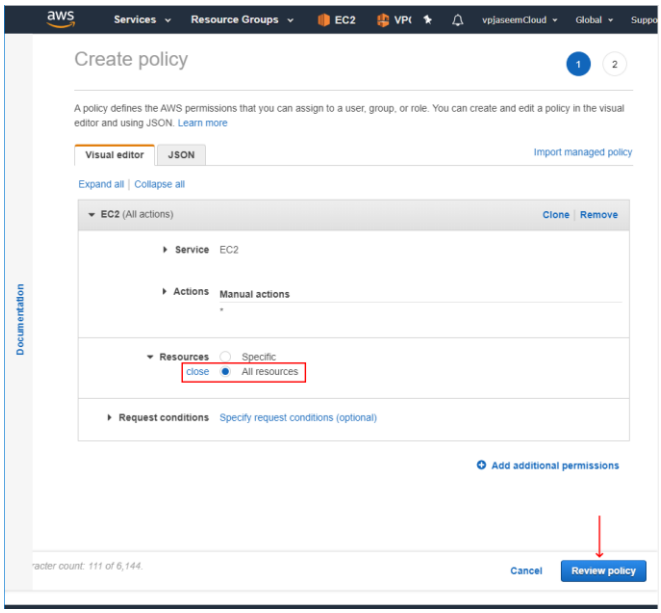



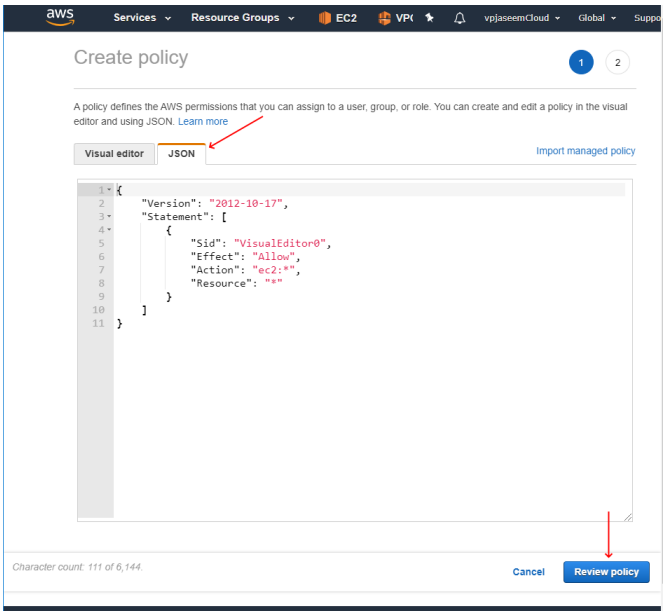



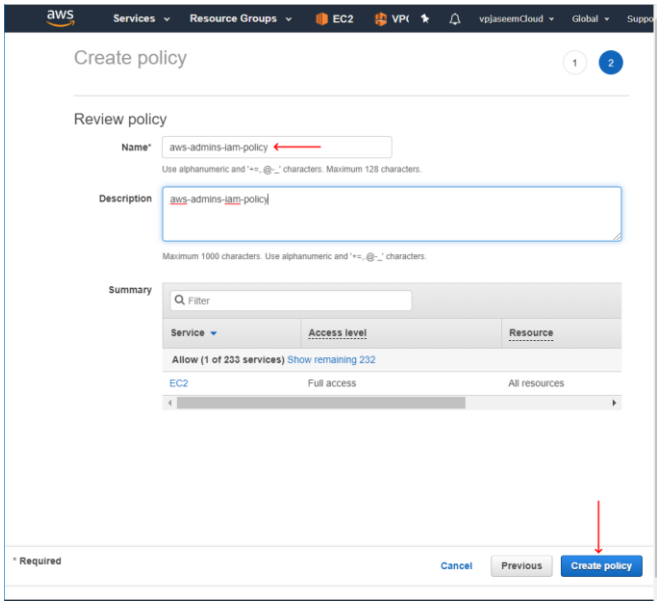
# [LAB] Create IAM Policy

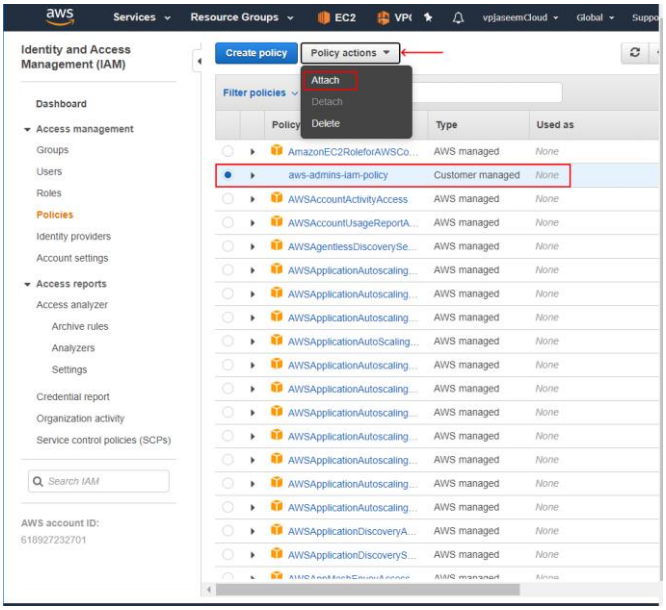


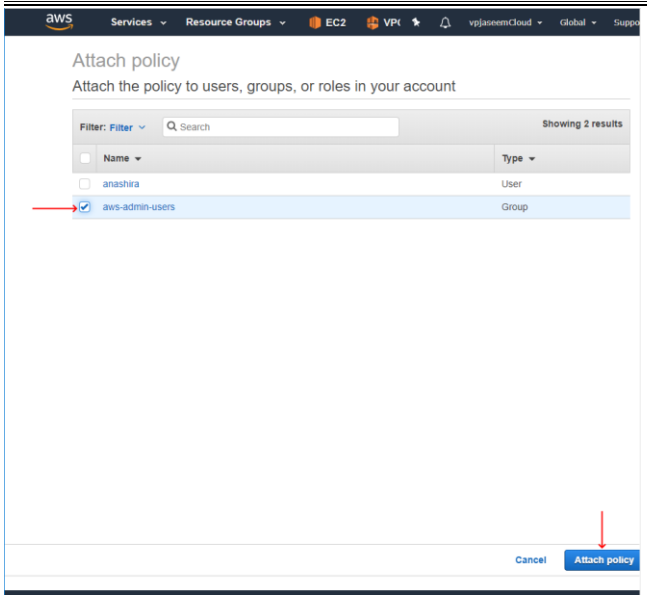



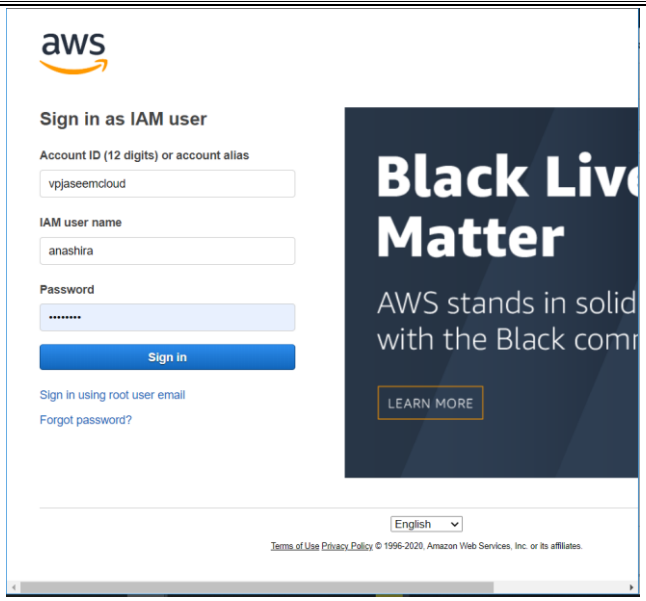



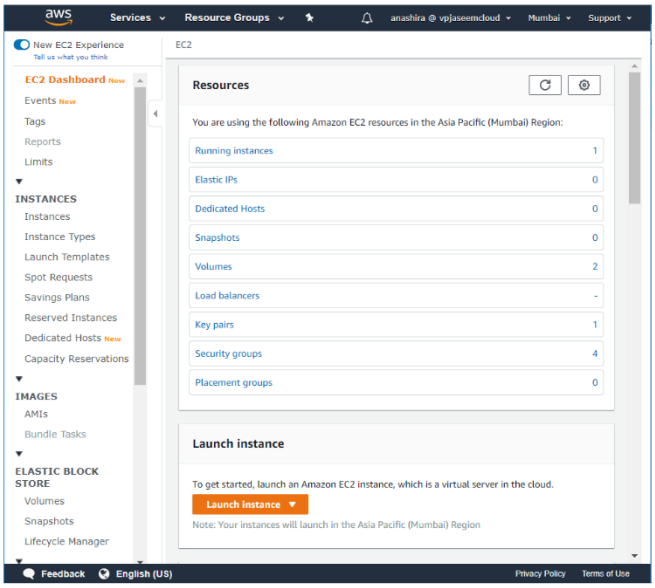



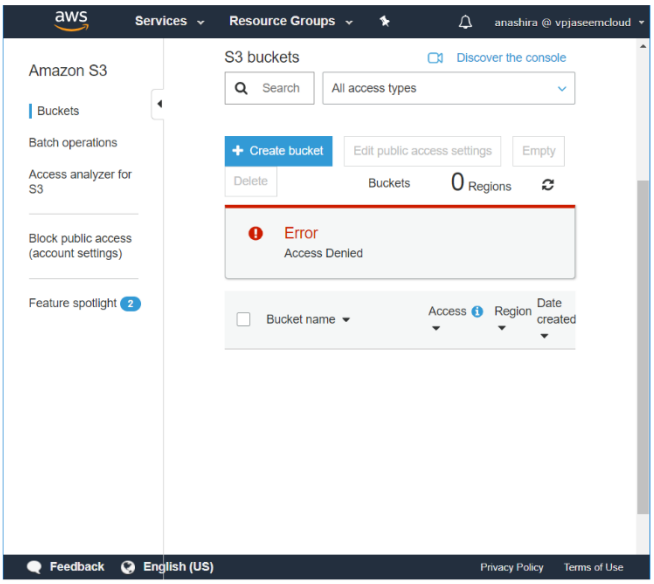



*IAM user accessing EC2*

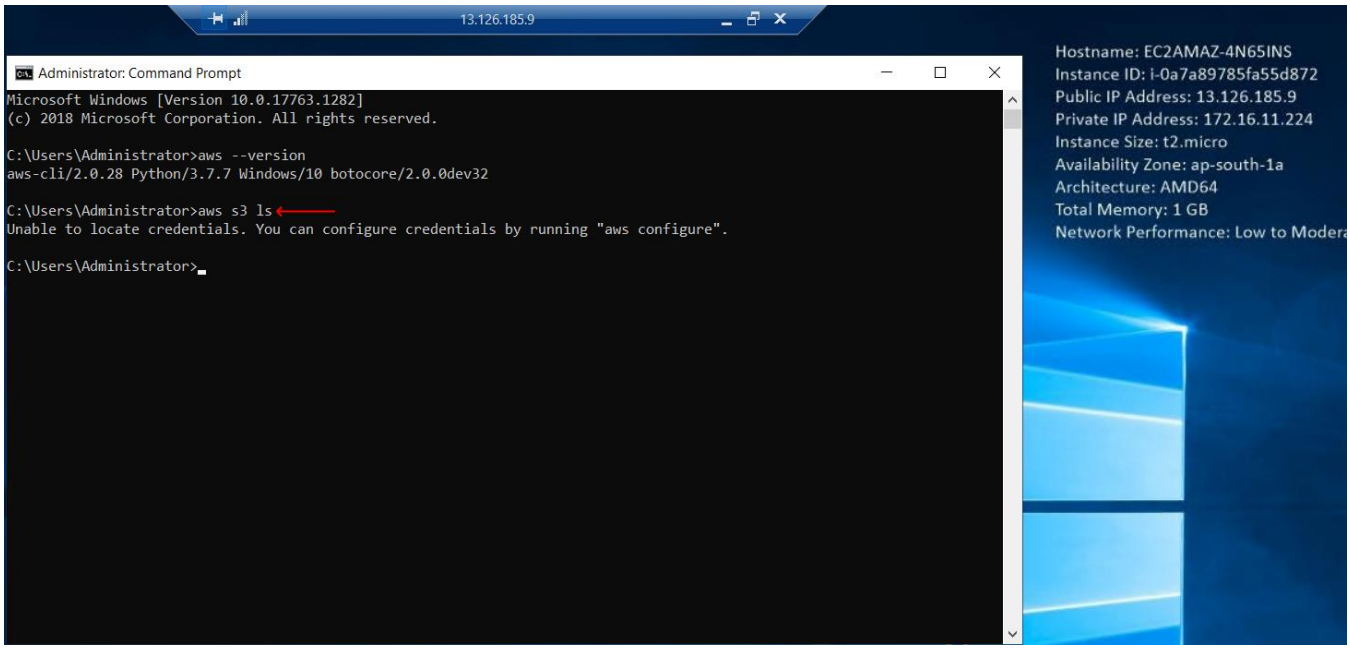



*IAM user accessing S3*

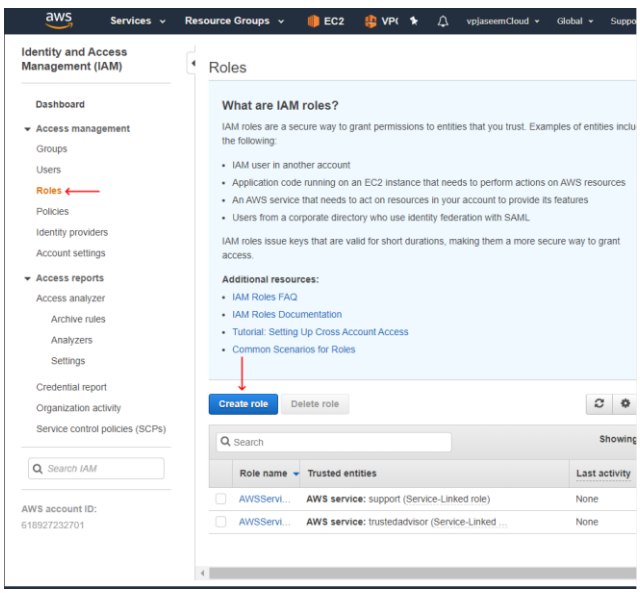


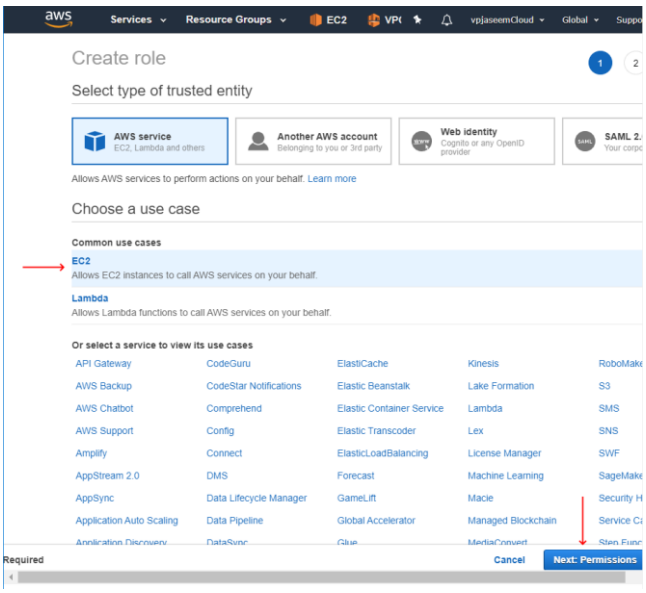
## [LAB] IAM Role Configuration

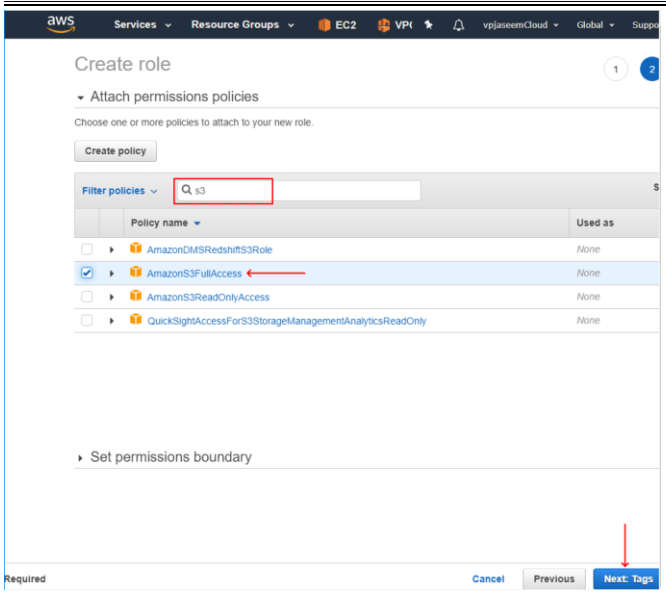
- IAM roles are a secure way to grant permissions to entities that you trust.
- If we want our EC2 resource to access S3 bucket (without user ID, Password, Secret key)
- Securely distribute your AWS credentials to the instances, enabling the applications on those instances to use your credentials to sign requests, while protecting your credentials from other users.
- Commonly used for API or CLI access to AWS.
- For this lab, either you need to have AWS CLI installed on your Windows EC2 or you should have an Amazon Linux EC2 (AWS CLI is pre-installed). I'm going with Windows Installer.
- Follow the link to download setup file for Windows and install it on your Windows EC2. The installation is out of the scope of this document.

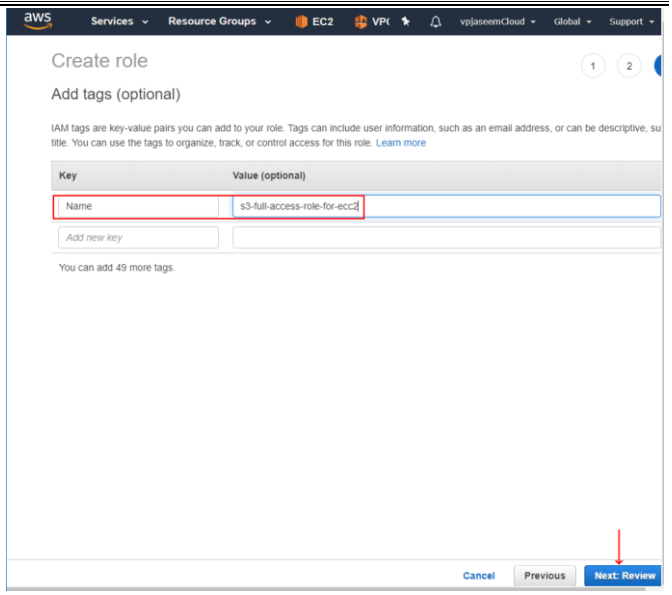


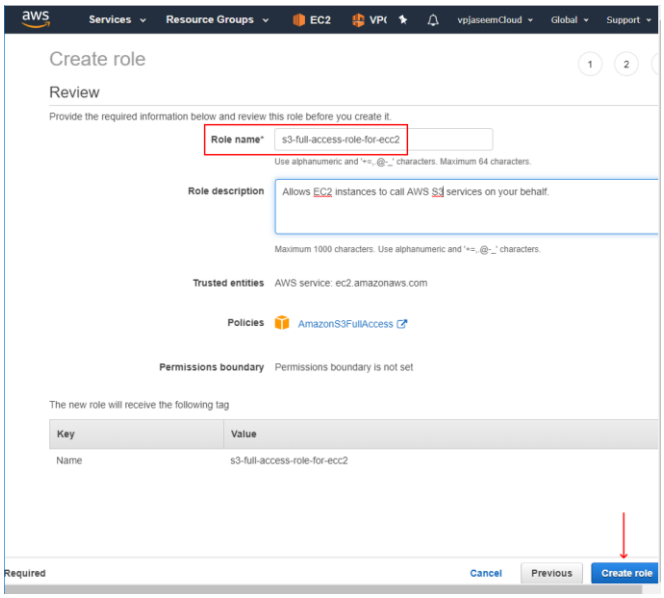

- I have used an AWS CLI command (aws s3 ls) to list the S3 bucket details, you can see that there is an error message 'Unable to locate credentials' because we did not configure anything.
- To provide access to S3 from this machine, without a username and password, we use IAM roles.

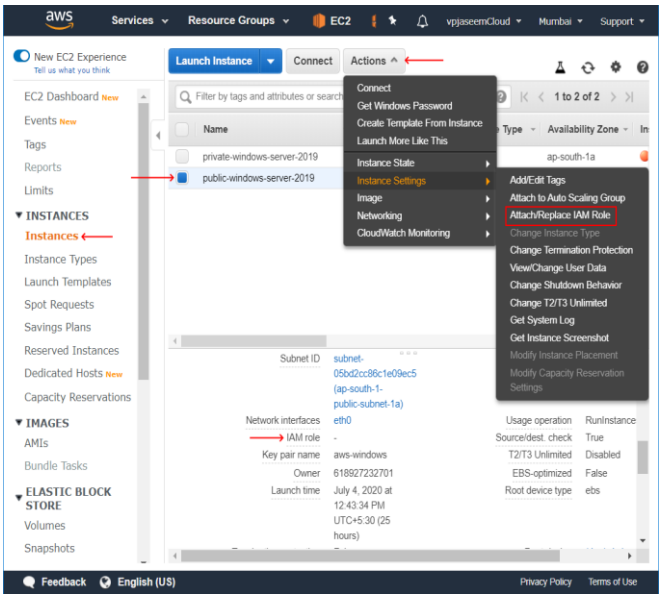


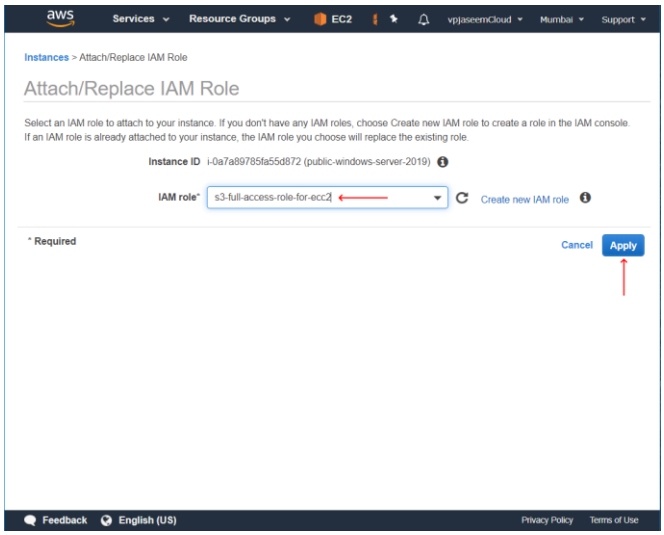



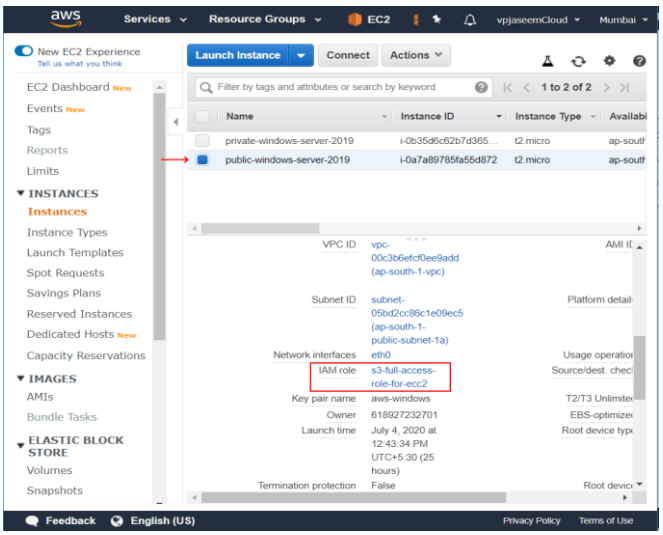


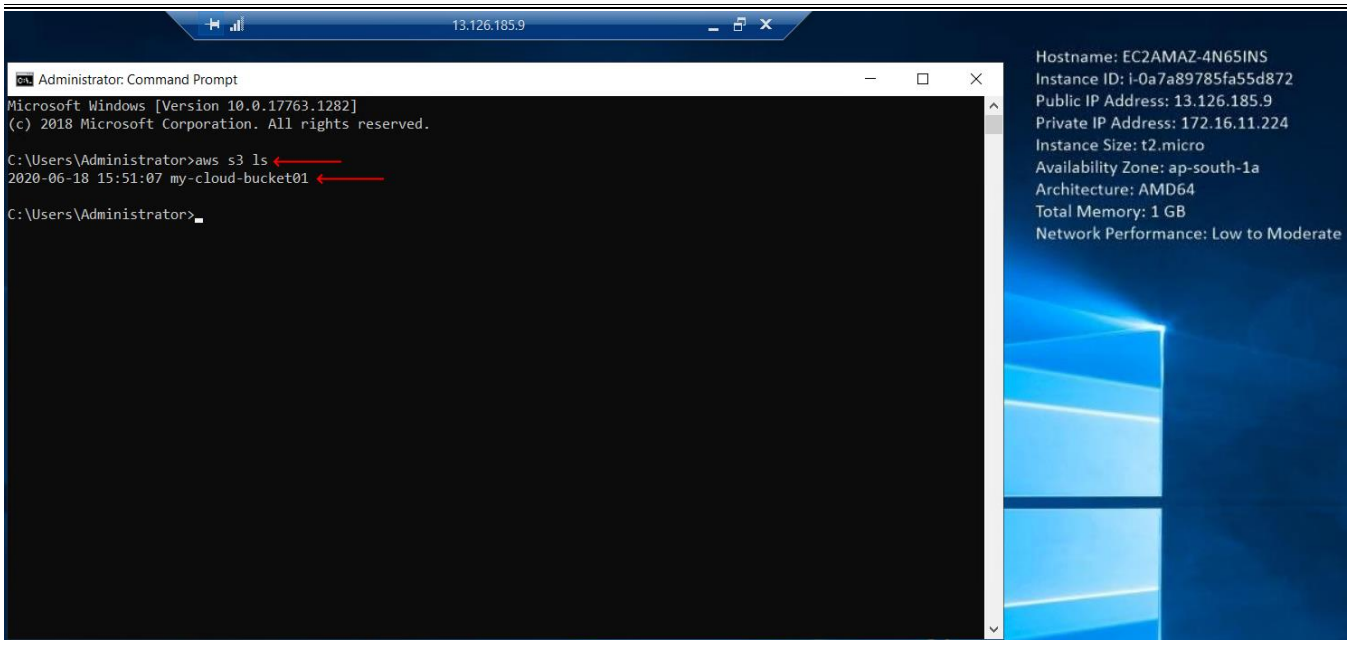



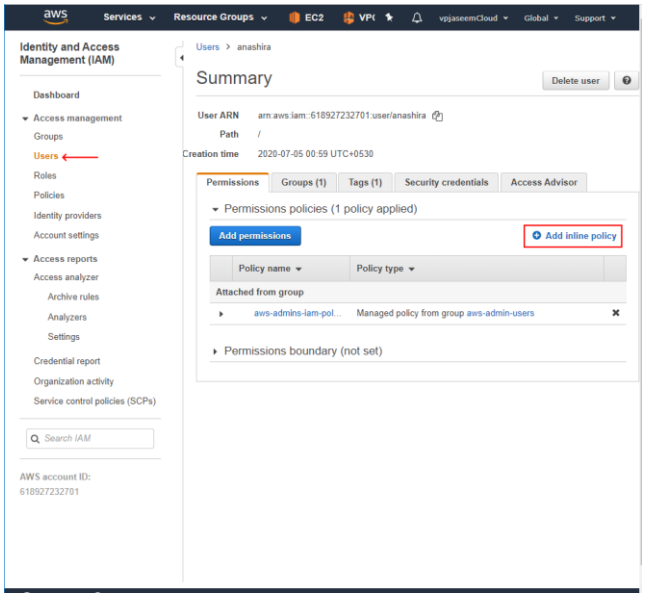



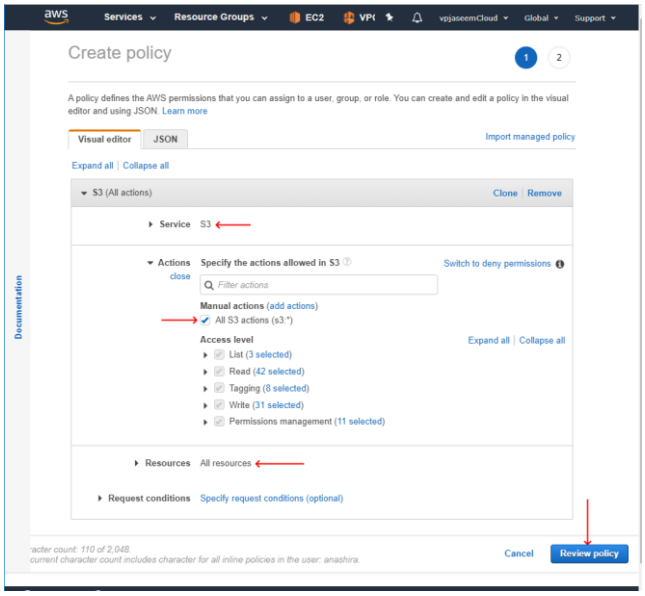

- Now the Windows EC2 can access S3 (AWS CLI commands) from Windows CMD

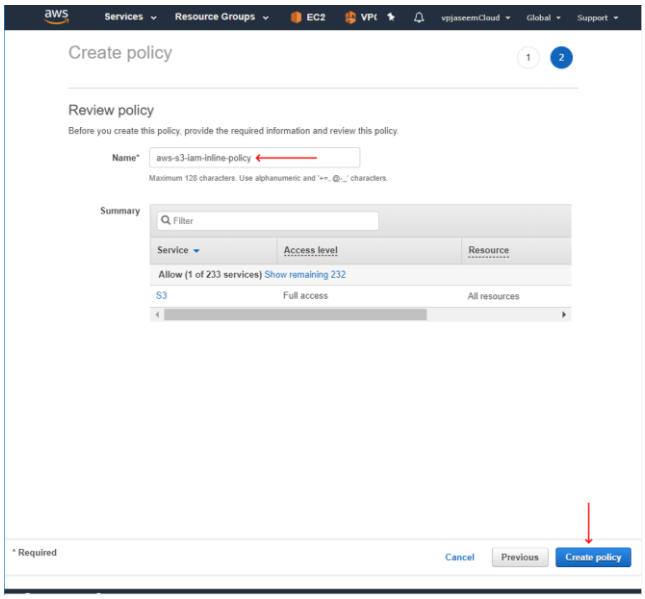

13.126.185.9
Administrator: Command Prompt
Microsoft Windows [Version 10.0.17763.1282]
(c) 2018 Microsoft Corporation. All rights reserved.
C:\Users\Administrator>aws s3 ls
2020-06-18 15:51:07 my-cloud-bucket01
C:\Users\Administrator>
Hostname: EC2AMAZ-4N65INS
Instance ID: i-0a7a89785fa55d872
Public IP Address: 13.126.185.9
Private IP Address: 172.16.11.224
Instance Size: t2.micro
Availability Zone: ap-south-1a
Architecture: AMD64
Total Memory: 1 GB
Network Performance: Low to Moderate

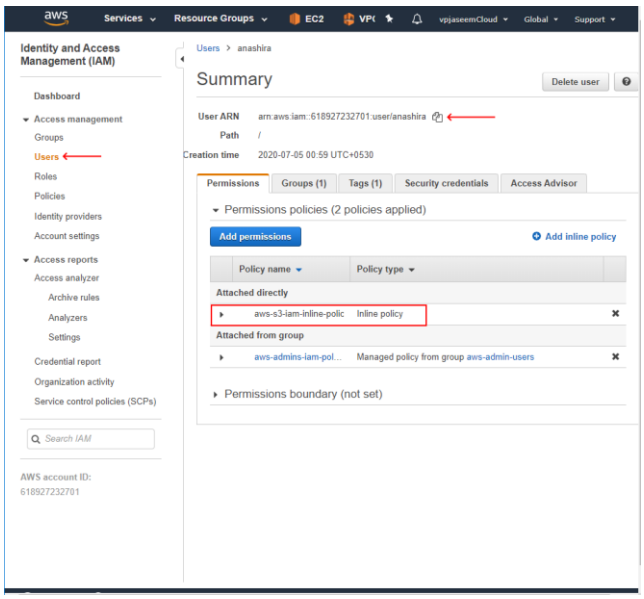
## [LAB] Inline Policy for IAM User

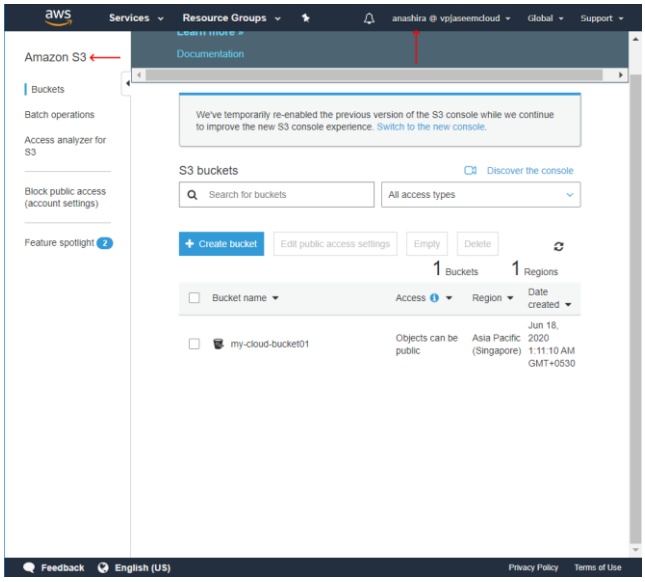
- These are the policies directly tied-up with Users, once you delete the user / group, the policy will get deleted.



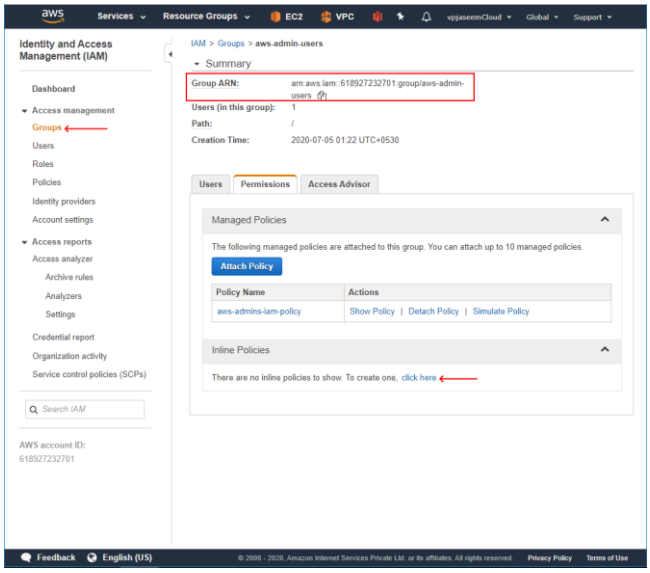



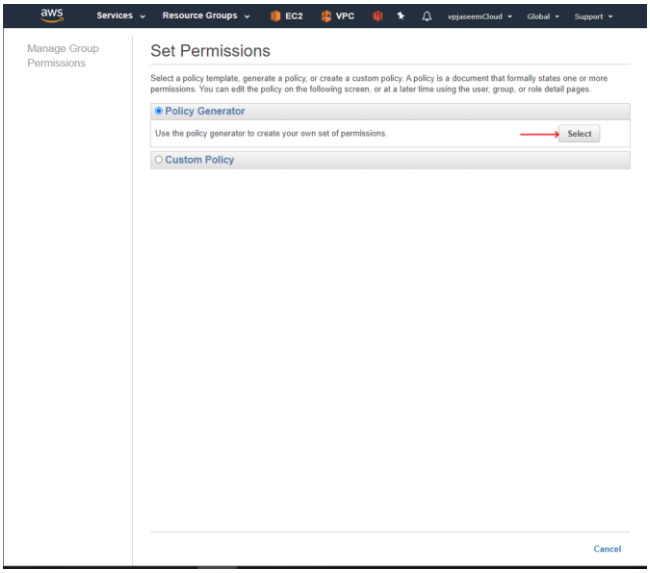



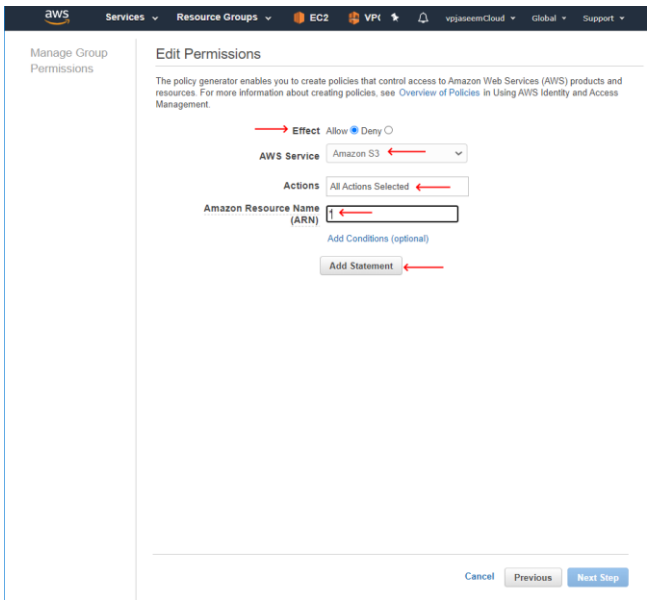



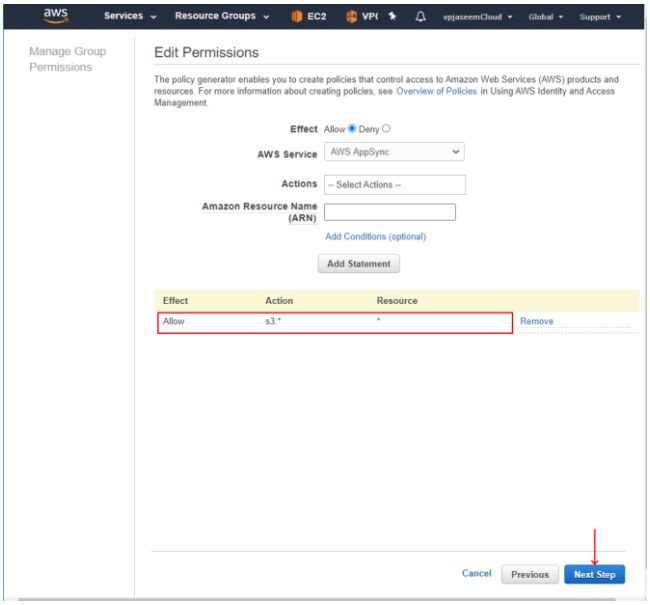
# [LAB] Inline Policy for IAM Group

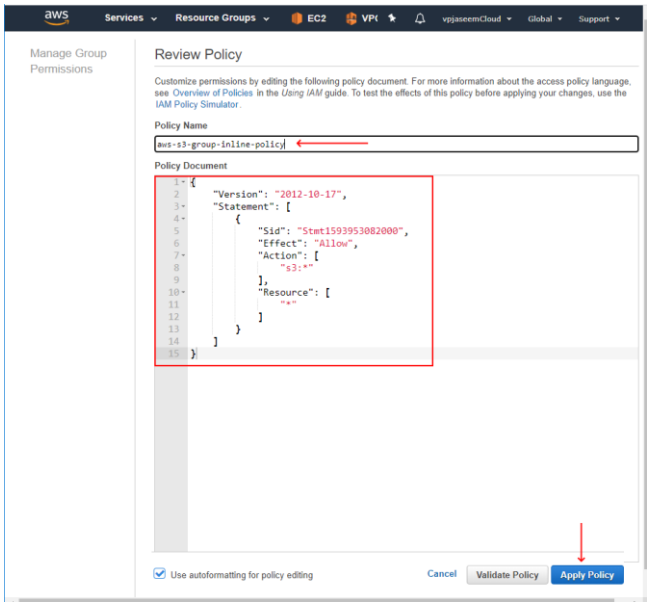
- Group specific policies can be assigned by this way, once you delete the group, the policy will get deleted.

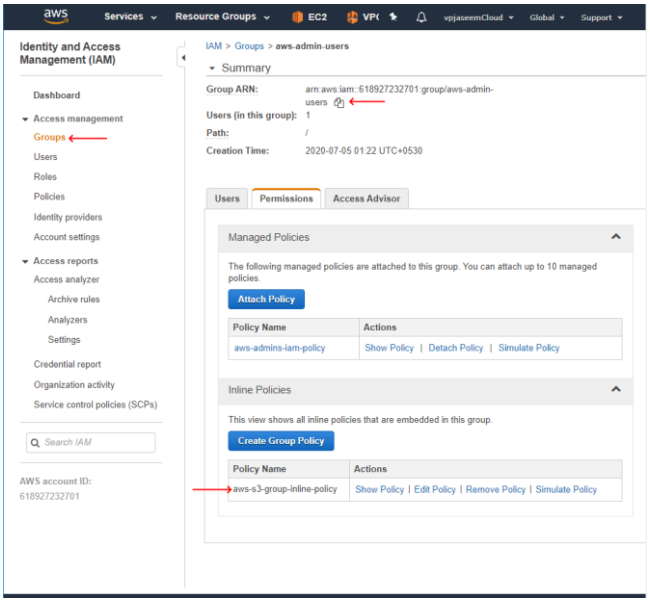


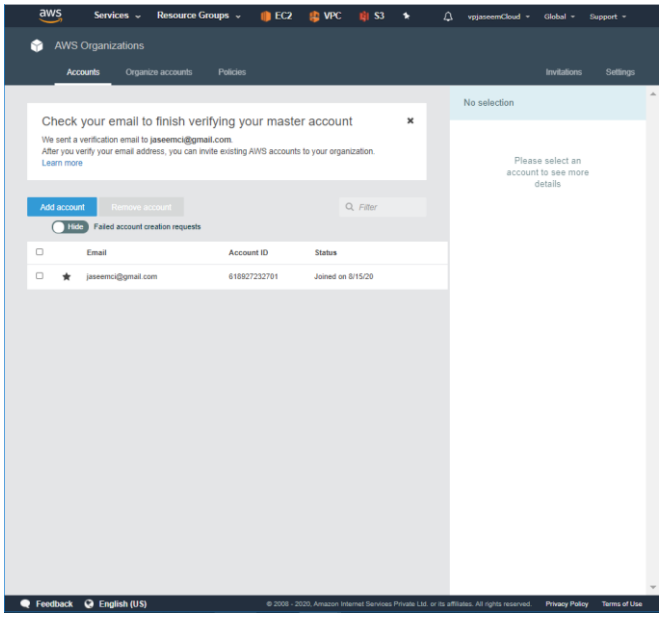



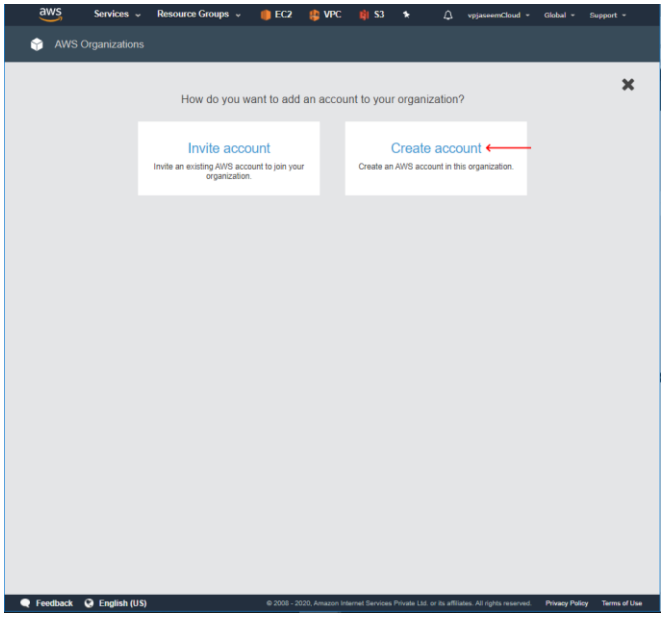



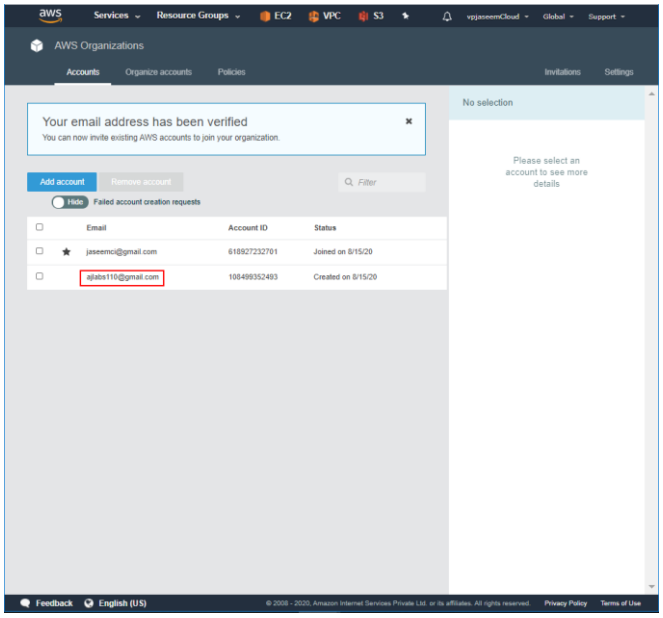
## AWS Organizations

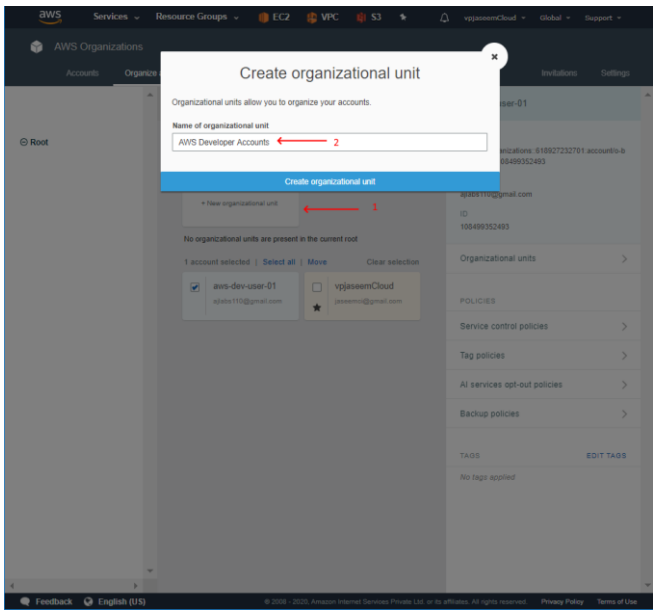
- Master and Child account concept
- Consolidated billing of multiple accounts
- Service Control Policy (Specific roles for child account)
- AWS Organizations enables you to centrally apply policy-based controls across multiple accounts in the AWS Cloud.
- You can consolidate all your AWS accounts into an organization, and arrange all AWS accounts into distinct organizational units.
- Invite existing AWS accounts to join your organization or create new accounts based on your needs.

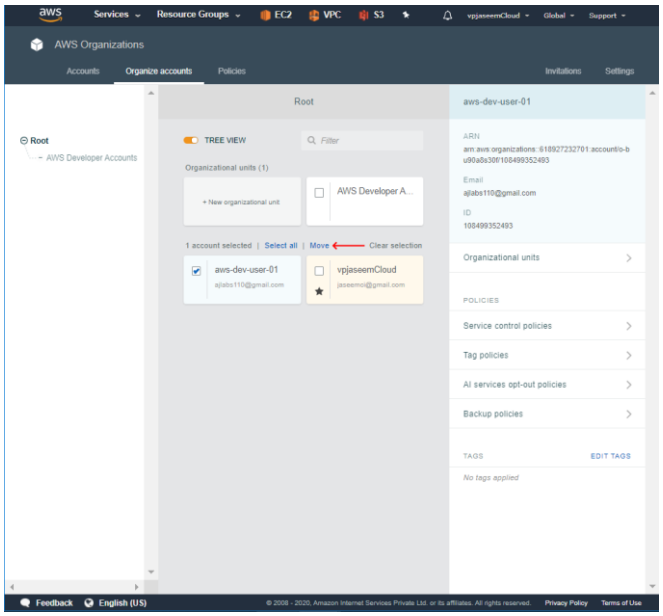
# [LAB] Creating a Child Account and Provide t2.micro Policy

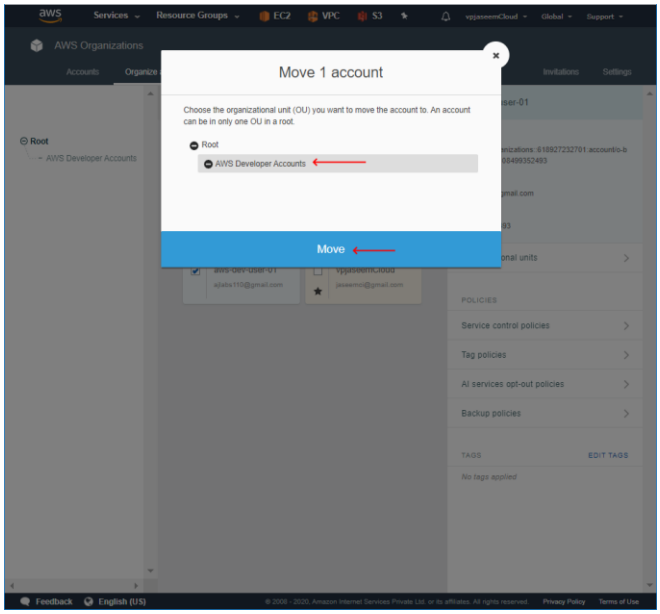


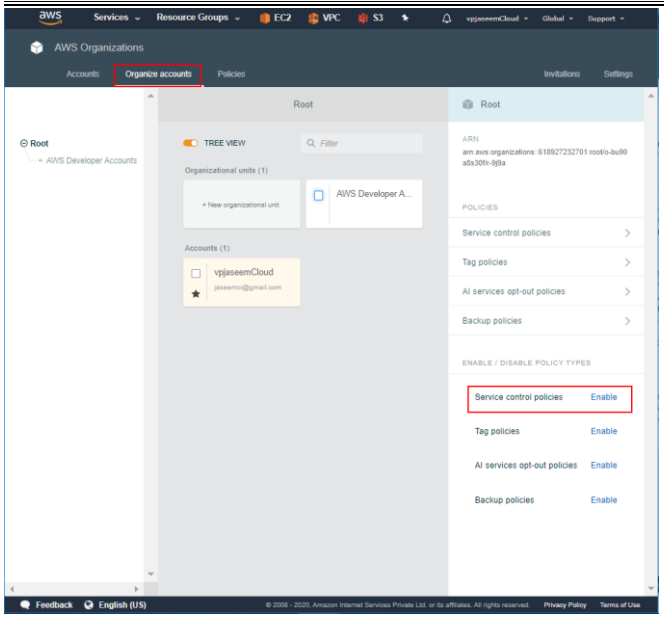



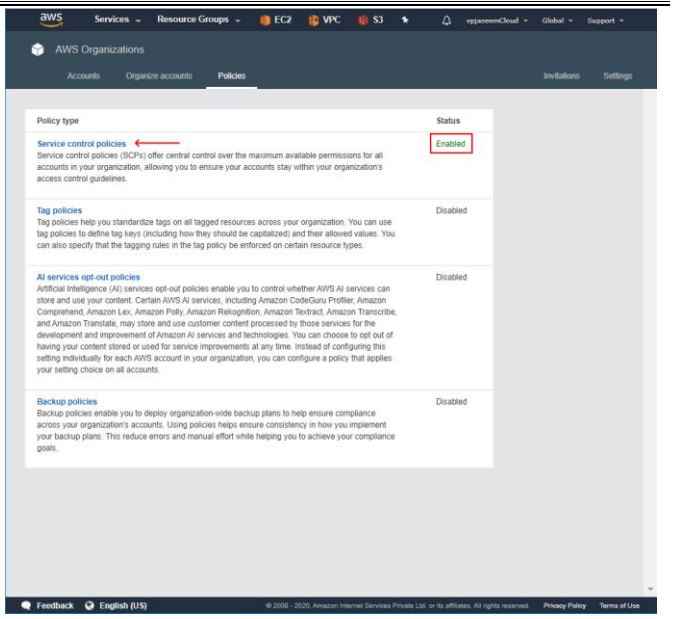



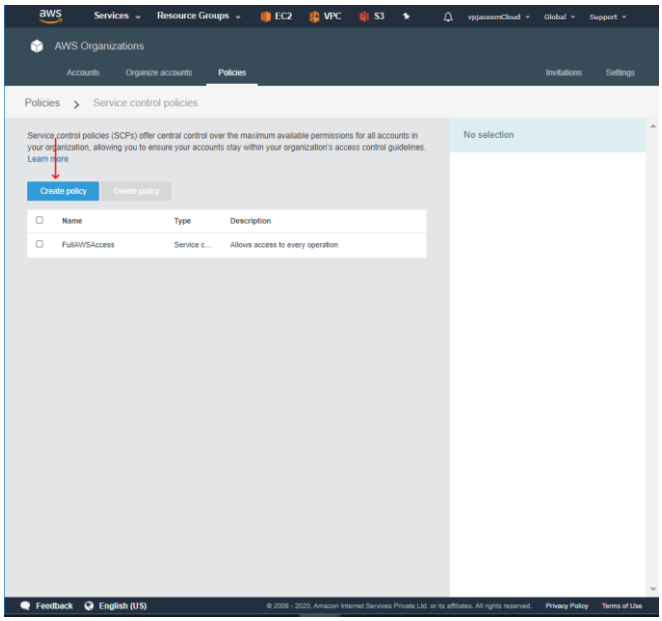



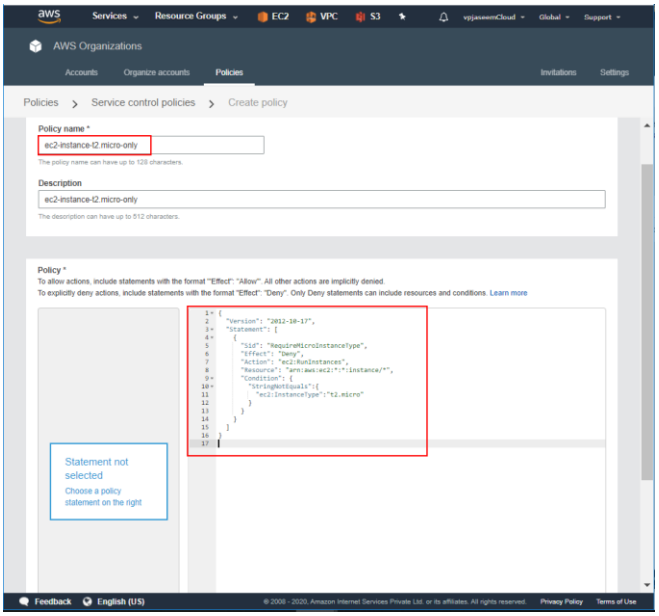



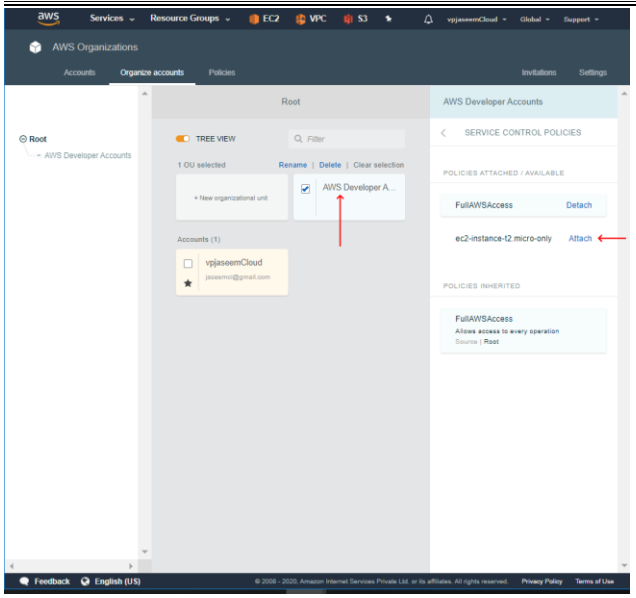


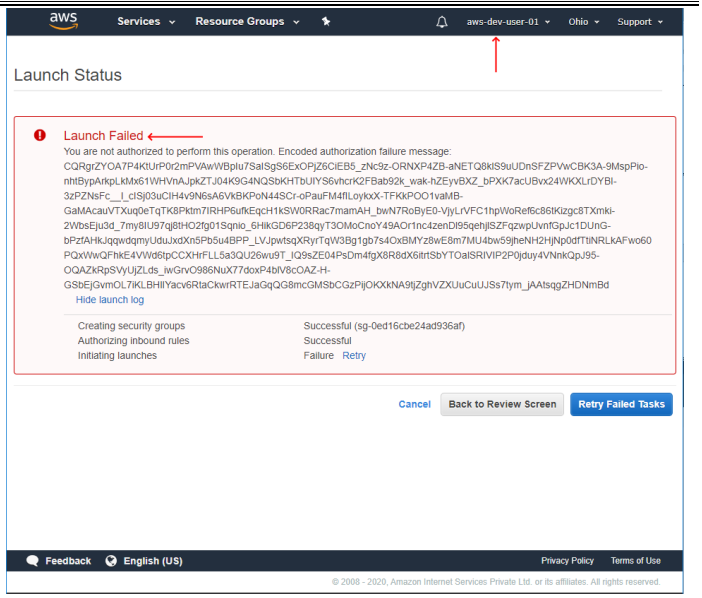




```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "RequireMicroInstanceType",
      "Effect": "Deny",
      "Action": "ec2:RunInstances",
      "Resource": "arn:aws:ec2:*:*:instance/*",
      "Condition": {
        "StringNotEquals":{
          "ec2:InstanceType":"t2.micro"
        }
      }
    }
  ]
}
```




AWS Reference: [What is AWS Organizations]

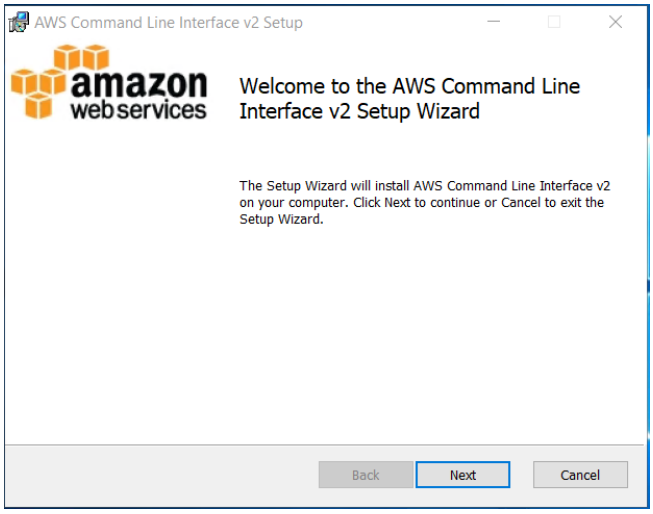
# AWS CLI

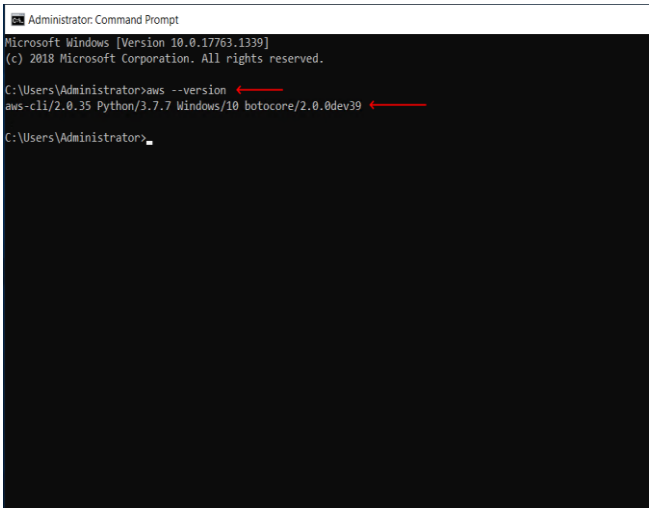
- The AWS Command Line Interface (AWS CLI) is an open source tool that enables you to interact with AWS services using commands in your command-line shell.
- With minimal configuration, the AWS CLI enables you to start running commands that implement functionality equivalent to that provided by the browser-based AWS Management Console from the command prompt in your favorite terminal program.

## Install AWS CLI on Windows

- Download AWS CLI Setup file from this link: https://awscli.amazonaws.com/AWSCLIV2.msi
- Go ahead and install the program (On your PC or AWS EC2)

AWS Reference: Installing the AWS CLI version 2 on Windows

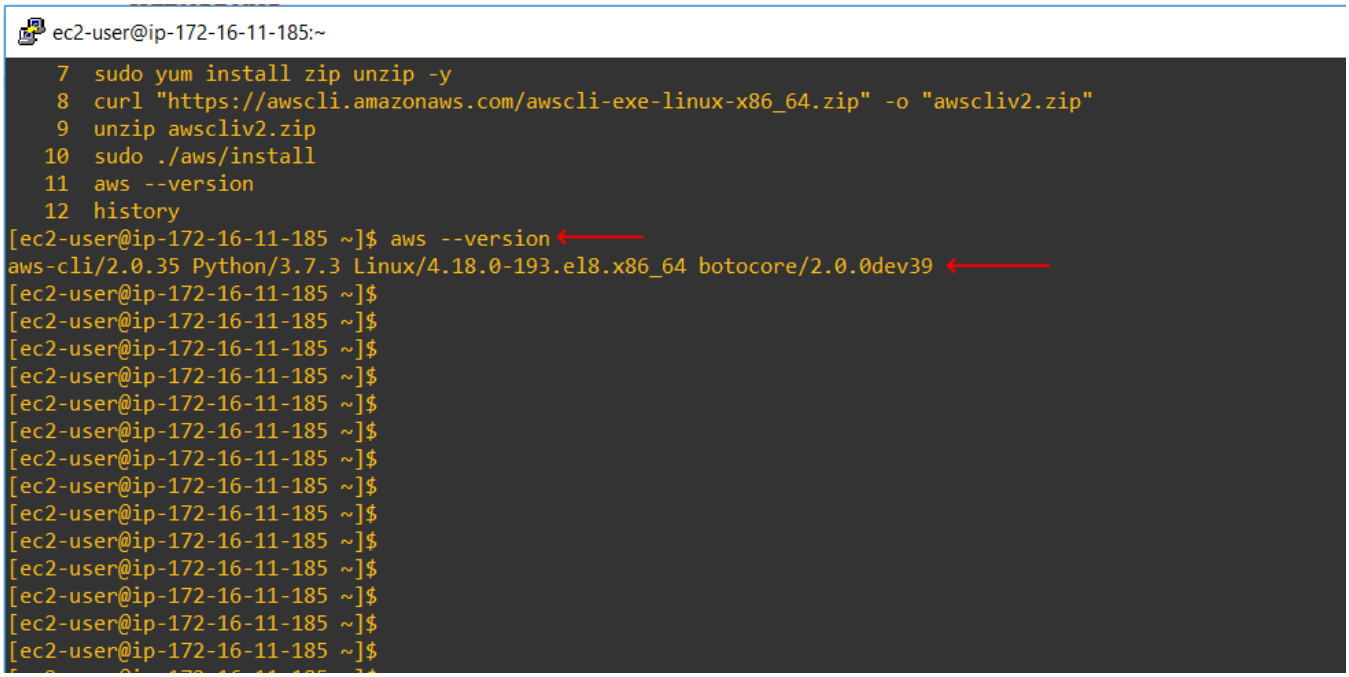


## Install AWS CLI on Linux

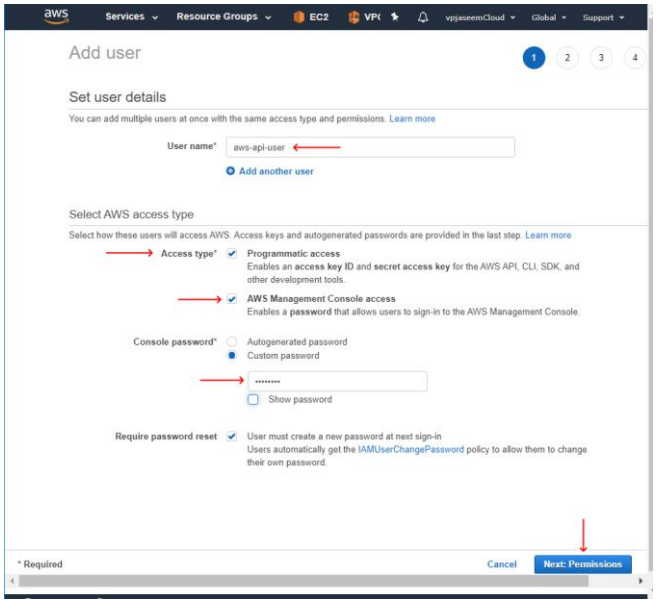
- The Amazon Linux AMI comes with pre-installed AWS CLI, If you have any other Linux distribution, follow below steps.

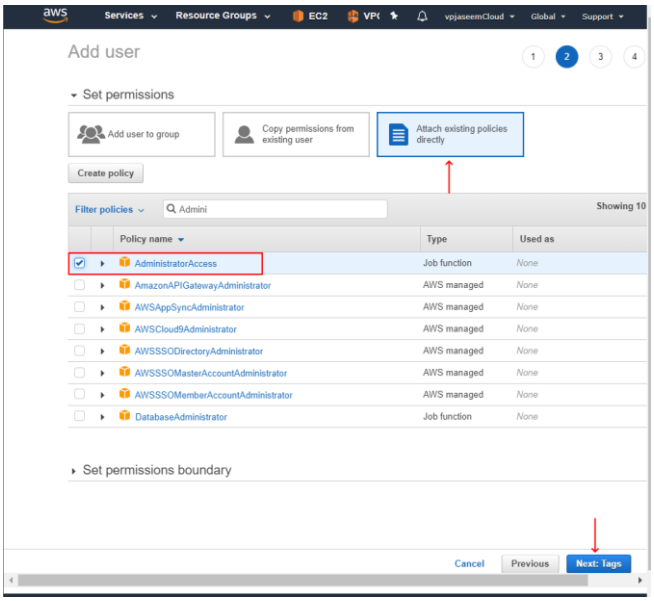
```
sudo yum install zip unzip -y
curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip"
unzip awscliv2.zip
sudo ./aws/install
aws --version
```

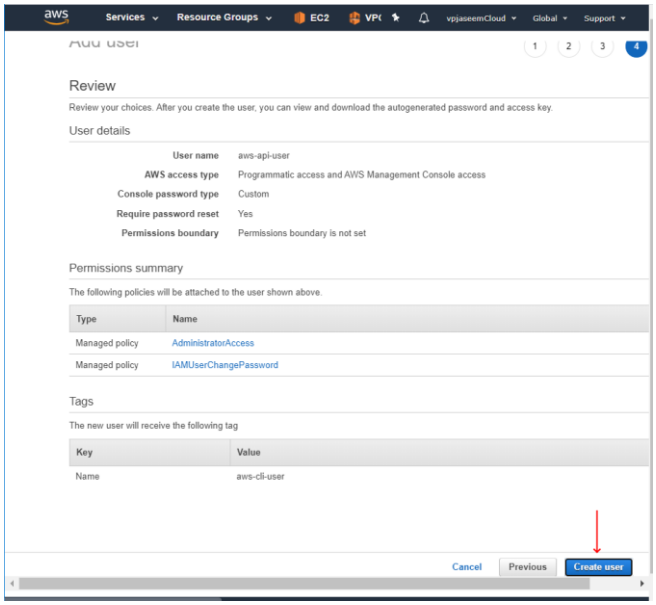


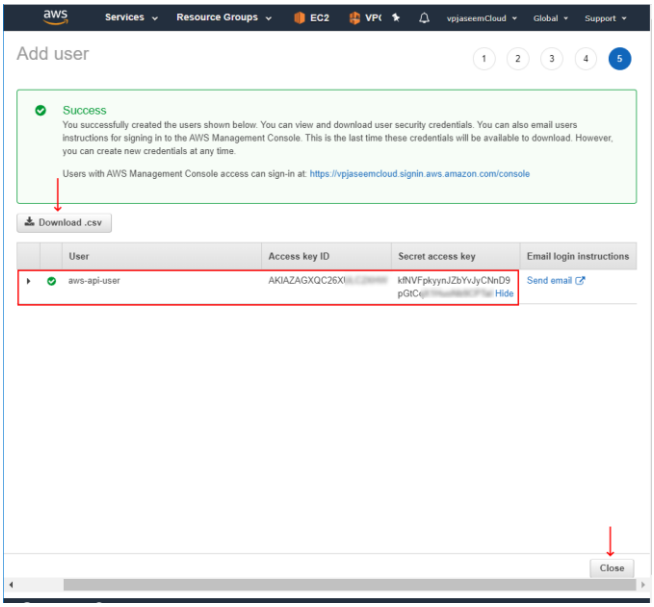
## AWS Access Key and Secret Key Authentication

- Access key and Secret key are used to access AWS via CLI (programmatic access), these are the credentials used to authenticate AWS from CLI.
- Go to IAM and create a user, make sure you copy Access key and Secret key.



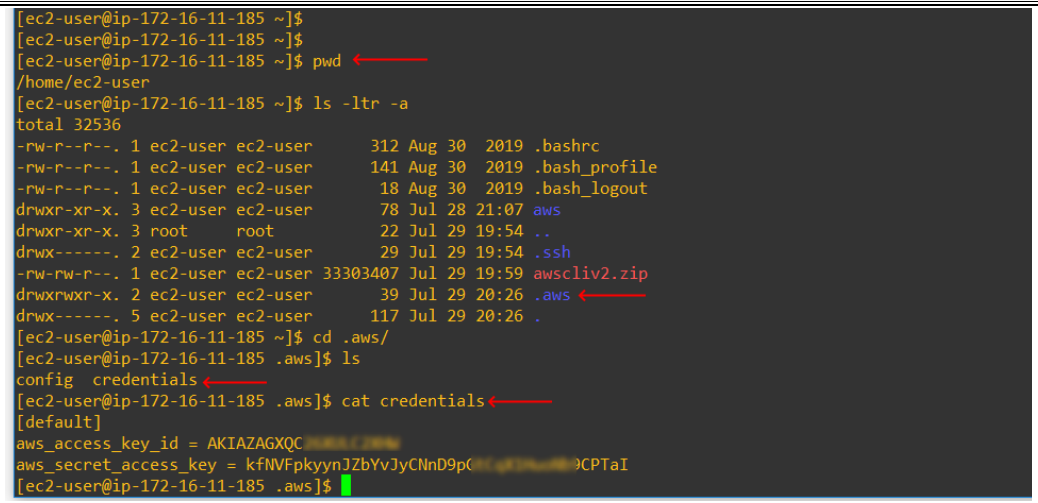




- Login to AWS CLI and enter below,

```
[ec2-user@ip-172-16-11-185 ~]$ aws configure
AWS Access Key ID [None]: AKIAZAGXQCXXXXXXXXHW
AWS Secret Access Key [None]: kfNVFpkyynXXXXXXXXXXD9pGtCqX1HuoNb9CPTaI
Default region name [None]: ap-south-1
Default output format [None]:
```

- These data will be saved in '/home/ec2-user/.aws'
- Now the AWS CLI authenticated with our AWS account


[ec2-user@ip-172-16-11-185 ~]$
[ec2-user@ip-172-16-11-185 ~]$
[ec2-user@ip-172-16-11-185 ~]$ pwd
/home/ec2-user
[ec2-user@ip-172-16-11-185 ~]$ ls -ltr -a
total 32536
-rw-r--r--. 1 ec2-user ec2-user 312 Aug 30 2019 .bashrc
-rw-r--r--. 1 ec2-user ec2-user 141 Aug 30 2019 .bash_profile
-rw-r--r--. 1 ec2-user ec2-user 18 Aug 30 2019 .bash_logout
drwxr-xr-x. 3 ec2-user ec2-user 78 Jul 28 21:07 aws
drwxr-xr-x. 3 root root 22 Jul 29 19:54 ..
drwx------. 2 ec2-user ec2-user 29 Jul 29 19:54 .ssh
-rw-rw-r--. 1 ec2-user ec2-user 33303407 Jul 29 19:59 awscliv2.zip
drwxrwxr-x. 2 ec2-user ec2-user 39 Jul 29 20:26 .aws
drwx------. 5 ec2-user ec2-user 117 Jul 29 20:26 .
[ec2-user@ip-172-16-11-185 ~]$ cd .aws/
[ec2-user@ip-172-16-11-185 .aws]$ ls
config credentials
[ec2-user@ip-172-16-11-185 .aws]$ cat credentials
[default]
aws_access_key_id = AKIAZAGXQC
aws_secret_access_key = kfNVFpkyynJZbYvJyCNnD9p
CPTaI
[ec2-user@ip-172-16-11-185 .aws]$

## Launch EC2 Instance Using AWS CLI

- You can use the AWS Command Line Interface (AWS CLI) to launch, list, and terminate Amazon Elastic Compute Cloud (Amazon EC2) instances.

### 1. Launch EC2 Instance

```
aws ec2 run-instances --image-id ami-03dbf9550d4620230 --count 1 --instance-type t2.micro --key-name aws-windows --security-group-ids sg-03bc17a19529ccc88 --subnet-id subnet-05bd2cc86c1e09ec5
```

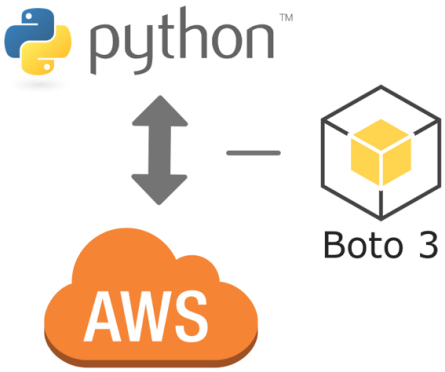

### 2. Stop EC2 Instance

```
aws ec2 stop-instances --instance-ids i-0a998993c7e1e334b
```

### 3. Terminate EC2 Instance

```
aws ec2 terminate-instances --instance-ids i-0a998993c7e1e334b
```

# AWS Python API / SDK - BOTO3

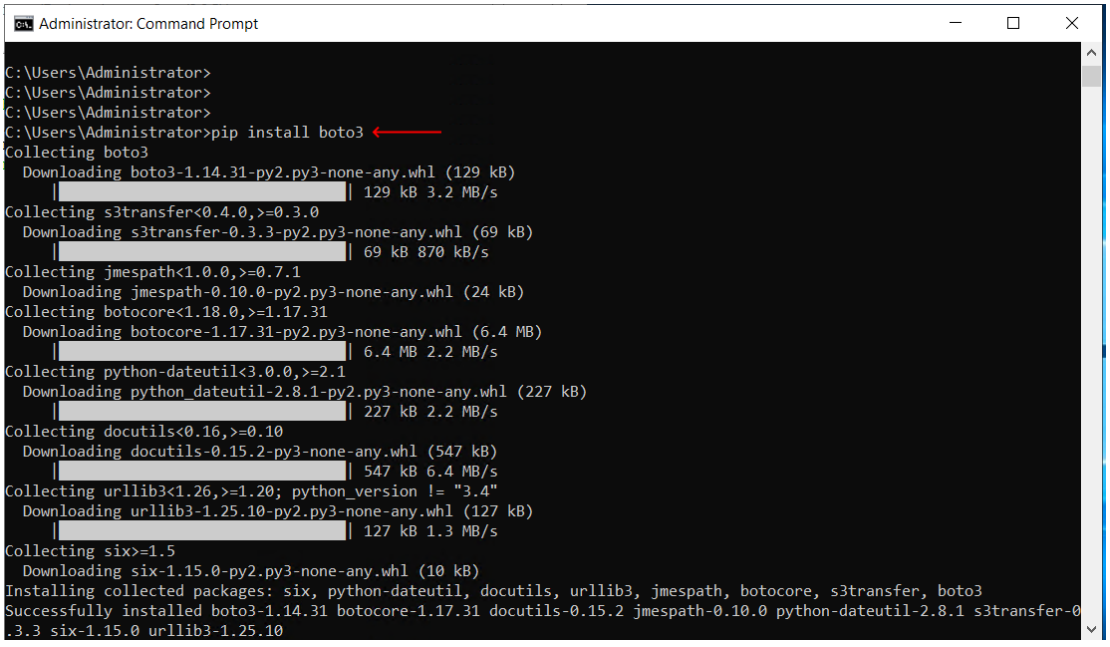



- Boto is the Amazon Web Services (AWS) SDK for Python.
- It enables Python developers to create, configure, and manage AWS services, such as EC2 and S3.
- Boto provides an easy to use, object-oriented API, as well as low-level access to AWS services.

Boto3 Reference: [Boto3 Documentation]

## Install BOTO3

- We use the popular Python Package Installer (PIP) for installing Boto3 (pip install boto3)



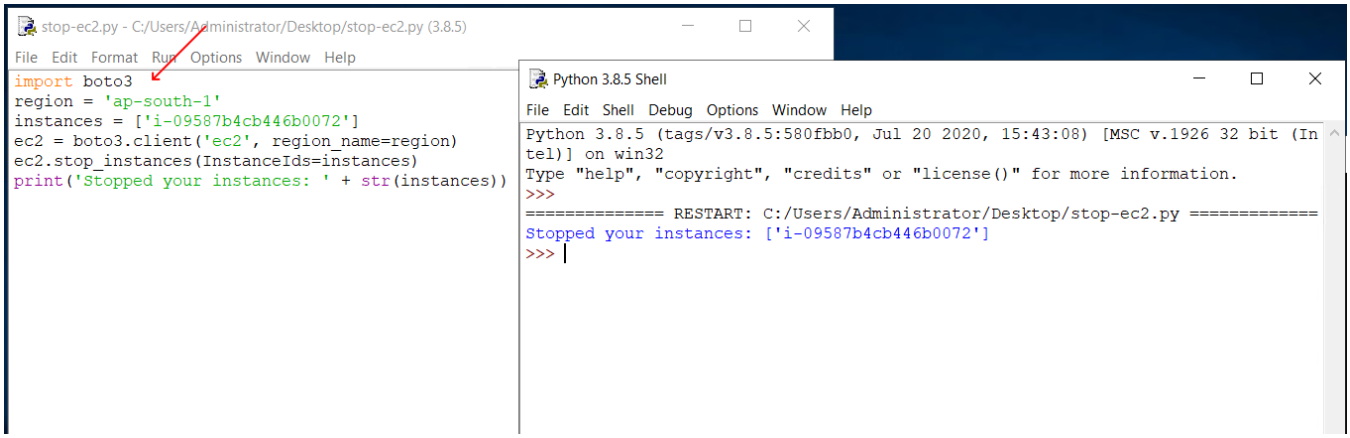
## Start / Stop EC2 Instance with Python Boto3 Module

- Run below Python Script to Start an EC2 Instance.

```
import boto3
region = 'ap-south-1'
instances = ['i-09587b4cb446b0072']
ec2 = boto3.client('ec2', region_name=region)
ec2.start_instances(InstanceIds=instances)
print('Started your instances: ' + str(instances))
```

- Run below Python Script to Stop an EC2 Instance.

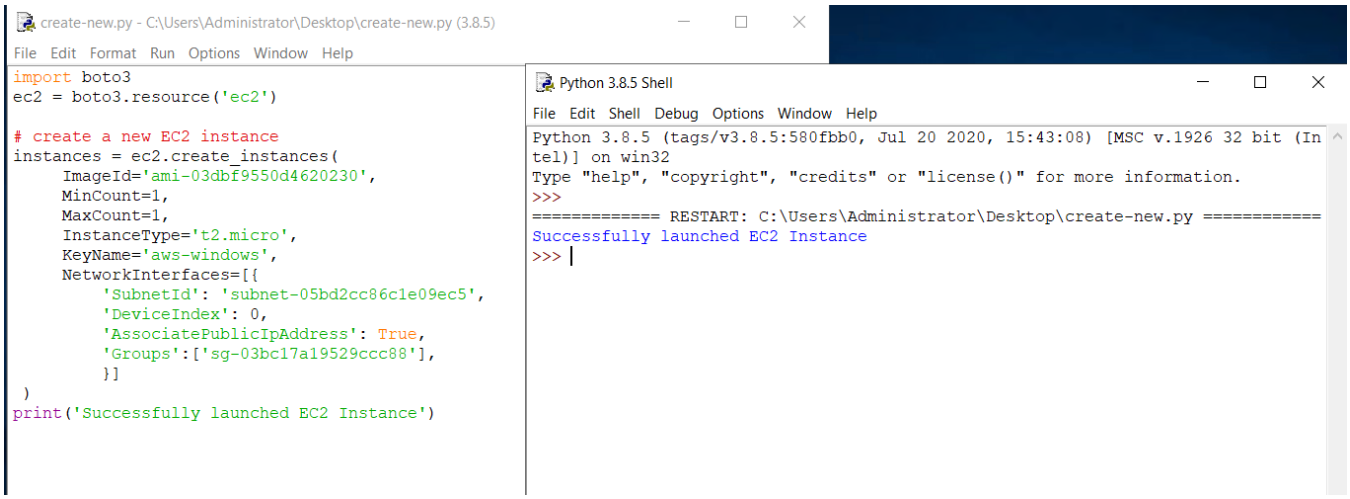
```
import boto3
region = 'ap-south-1'
instances = ['i-09587b4cb446b0072']
ec2 = boto3.client('ec2', region_name=region)
ec2.stop_instances(InstanceIds=instances)
print('Stopped your instances: ' + str(instances))
```



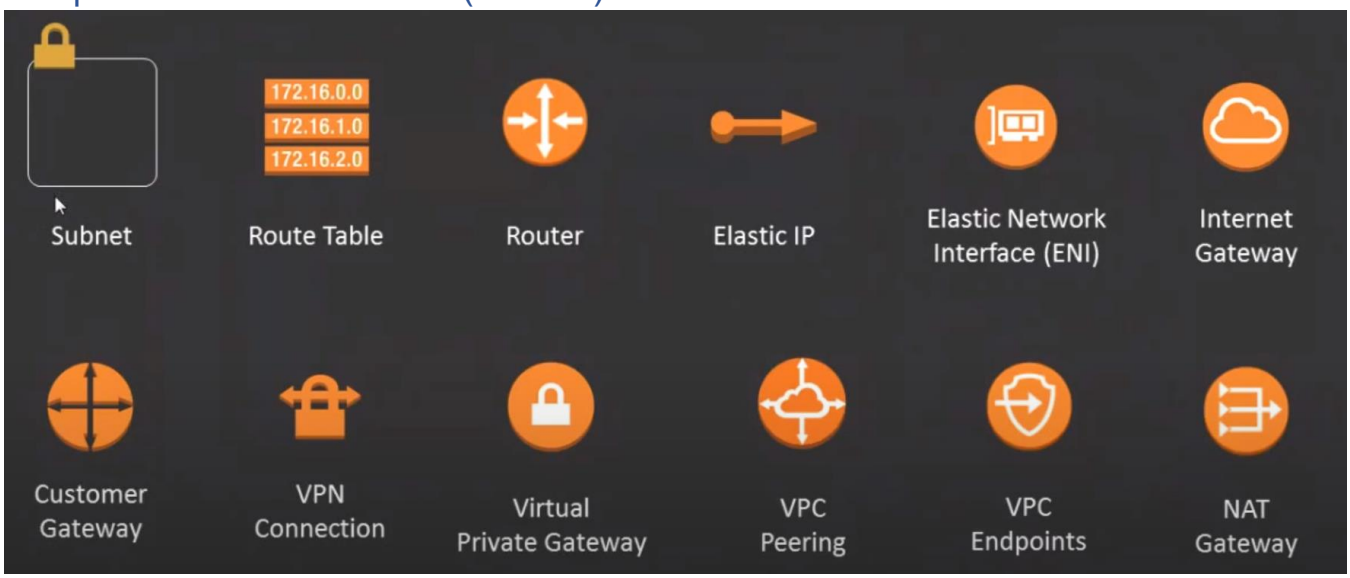
## Deploy EC2 using Python Boto3 API

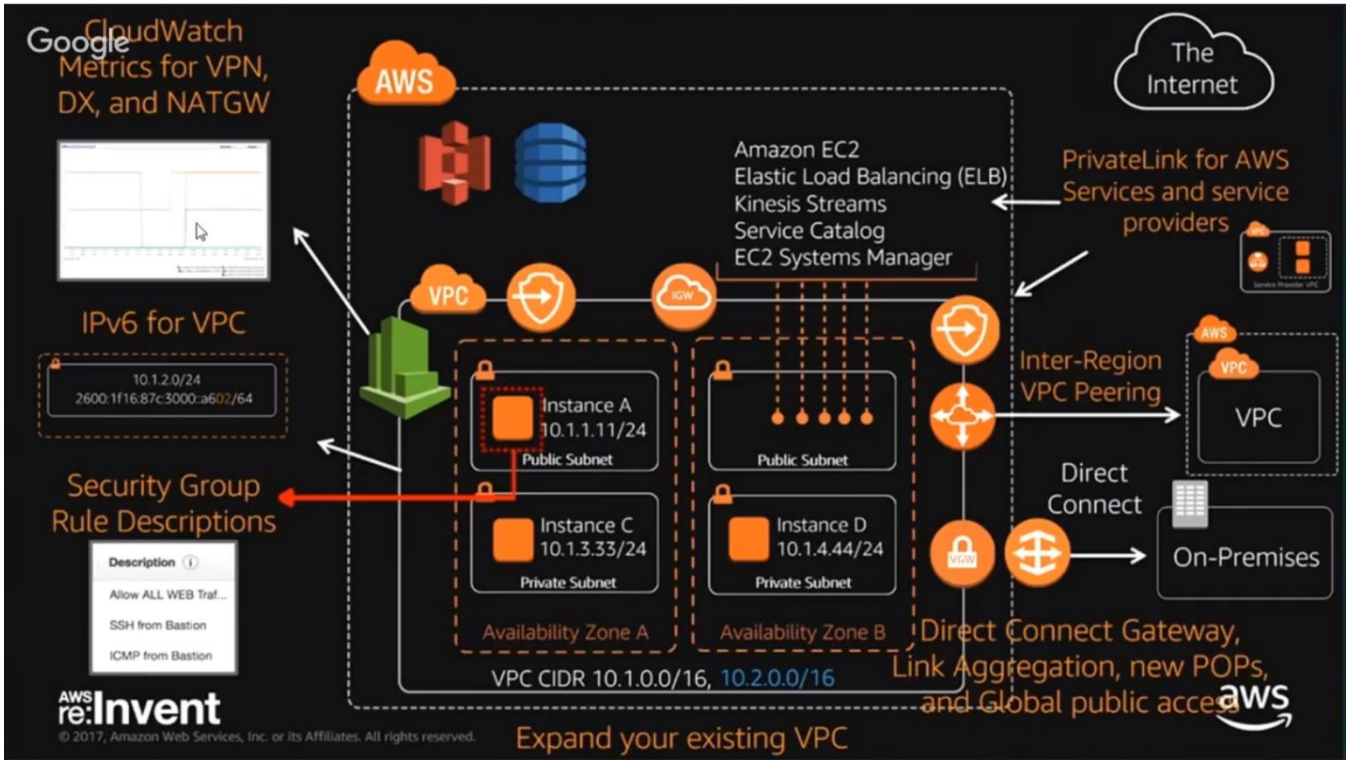
- This is the beauty of Automation, Boto3 has developed such a way that we can deploy an EC2 just in seconds.

```
import boto3
ec2 = boto3.resource('ec2', region_name='ap-south-1')

# create a new EC2 instance
instances = ec2.create_instances(
     ImageId='ami-03dbf9550d4620230',
     MinCount=1,
     MaxCount=1,
     InstanceType='t2.micro',
     KeyName='aws-windows',
     NetworkInterfaces=[{
          'SubnetId': 'subnet-05bd2cc86c1e09ec5',
          'DeviceIndex': 0,
          'AssociatePublicIpAddress': True,
          'Groups':['sg-03bc17a19529ccc88'],
          }]
 )
print('Successfully launched EC2 Instance')
```

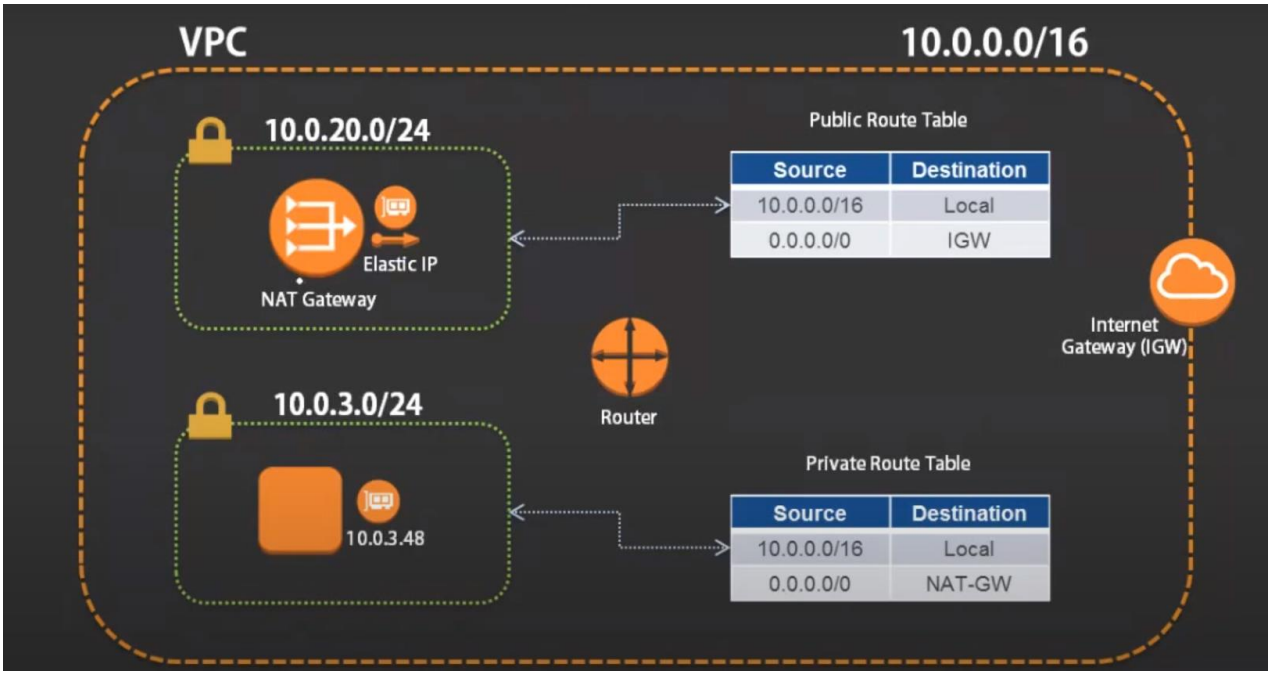


## Components of VPC Overview (Detailed)



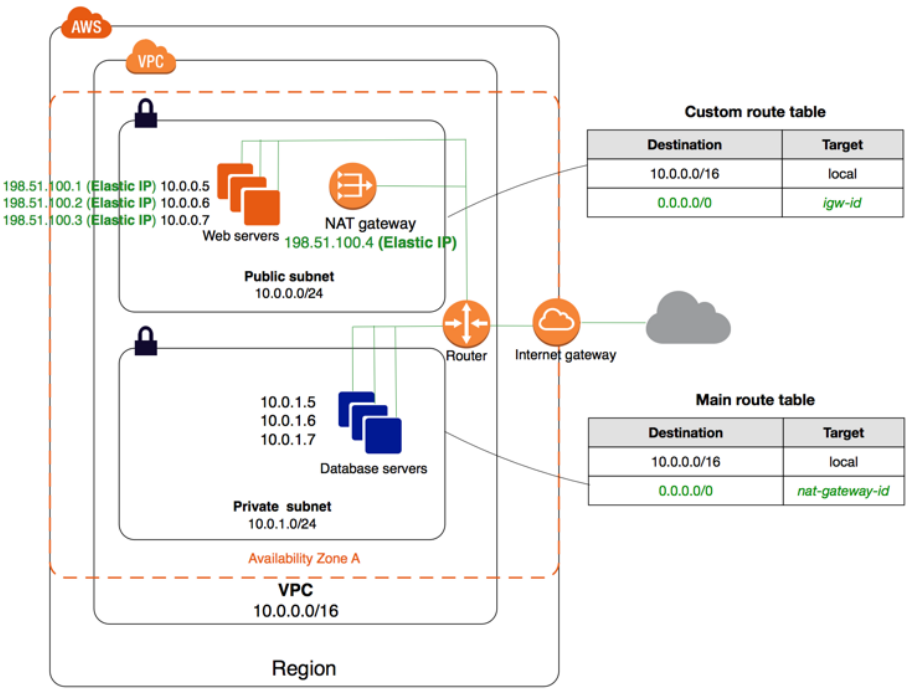



## NAT Gateway




- An AWS managed Network Address Translation (NAT) gateway to enable instances in Private Subnet to connect to internet or other AWS services, but prevent internet from initiating a connection to those instances.
- Allows your EC2 instances to perform updates/ patches etc. still being inside private subnet.
- Works only for IPV4. We use Egress-only Internet Gateway for IPV6
- Private Subnet checks Route Table and sees NAT-GW, the NAT-GW is in another subnet that has internet access via IGW and also a Public IP. (Private EC2 >> Route Table >> NAT Gateway >> IGW)
- Completely managed by AWS
- So far, we have configured EC2 instances in Public Subnet, what if we configure an EC2 on private subnet? Wo do we access them? Whether the EC2 gets internet?
- You can use a network address translation (NAT) gateway to enable instances in a private subnet to connect to the internet or other AWS services, but prevent the internet from initiating a connection with those instances.
- To access EC2 instances in Private Subnet, you need to have either VPN connection to VPC or access an EC2 in Public Subnet and from there connect to private subnet instances.
- You must also specify an Elastic IP address to associate with the NAT gateway when you create it.

- After you've created a NAT gateway, you must update the route table associated with one or more of your private subnets to point internet-bound traffic to the NAT gateway. This enables instances in your private subnets to communicate with the internet.

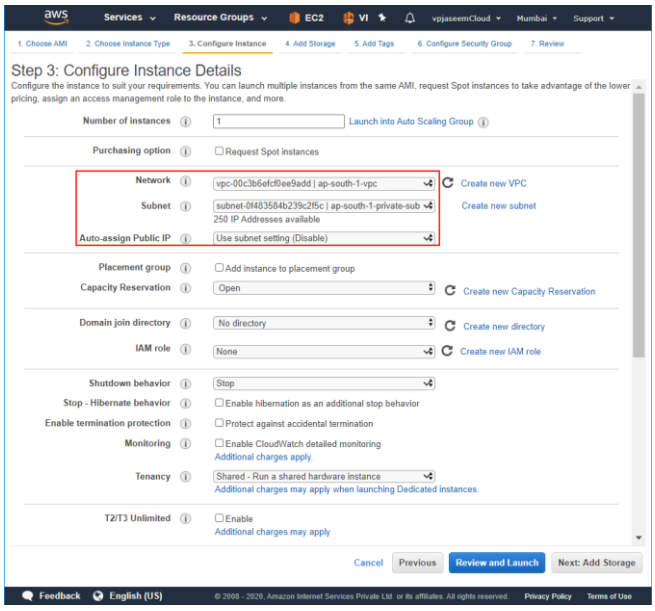


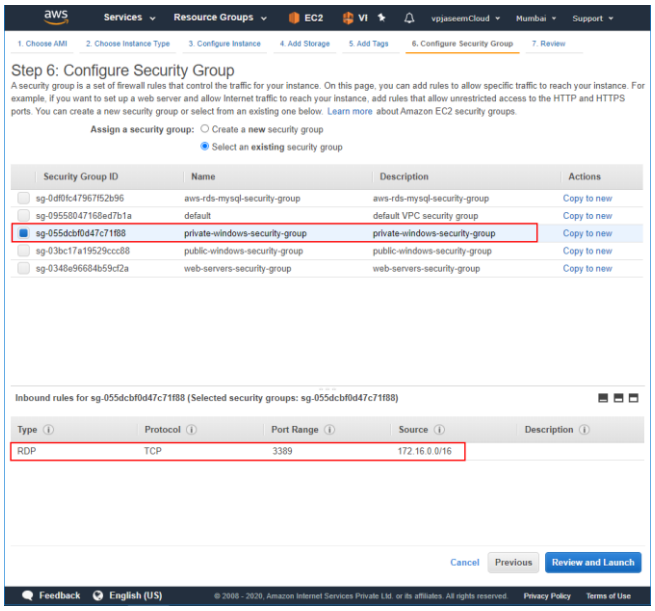

AWS Reference: NAT Gateway

- A NAT gateway supports 5 Gbps of bandwidth and automatically scales up to 45 Gbps.
- You can associate exactly one Elastic IP address with a NAT gateway. You cannot disassociate an Elastic IP address from a NAT gateway after it's created. To use a different Elastic IP address for your NAT gateway, you must create a new NAT gateway with the required address, update your route tables, and then delete the existing NAT gateway if it's no longer required.
- You cannot associate a security group with a NAT gateway. You can use security groups for your instances in the private subnets to control the traffic to and from those instances.
- You can use a network ACL to control the traffic to and from the subnet in which the NAT gateway is located. The network ACL applies to the NAT gateway's traffic. A NAT gateway uses ports 1024–65535.

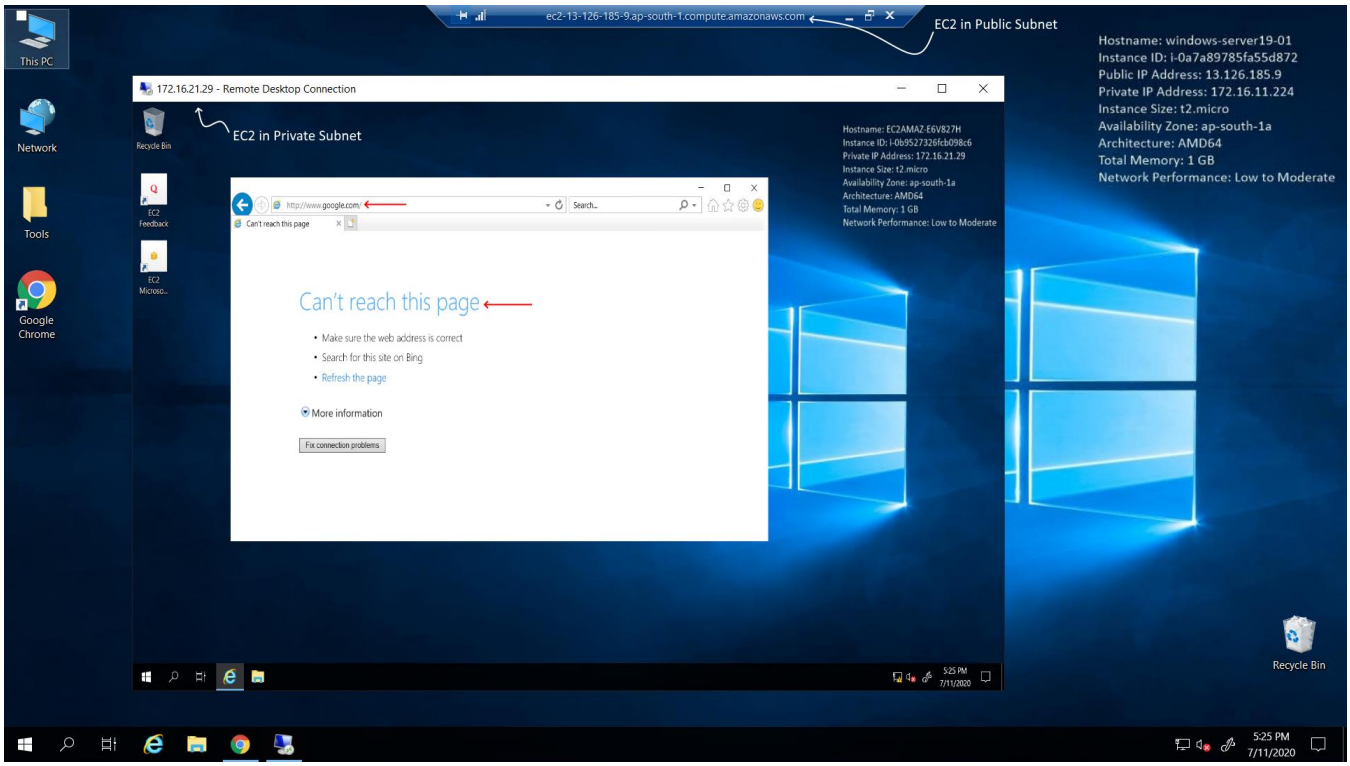
## [LAB] Provide Internet Access to Private Subnet Instances using NAT Gateway

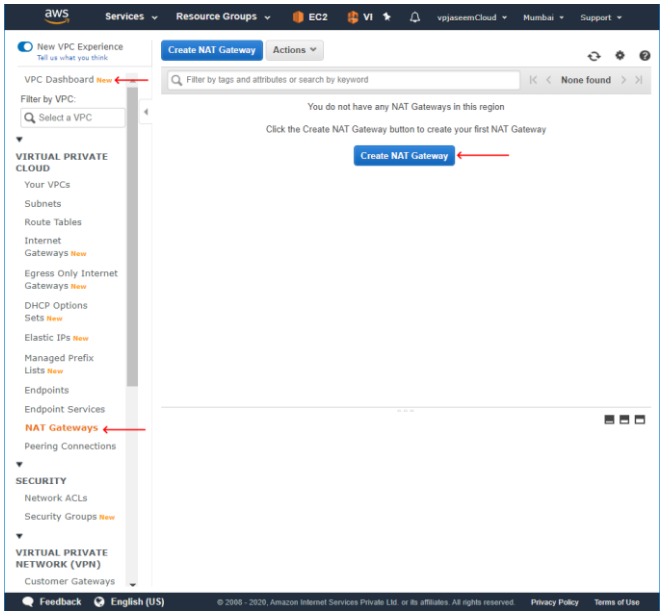
- Launch a Windows EC2 instance in Private Subnet, allow RDP from VPC CIDR (not from internet)

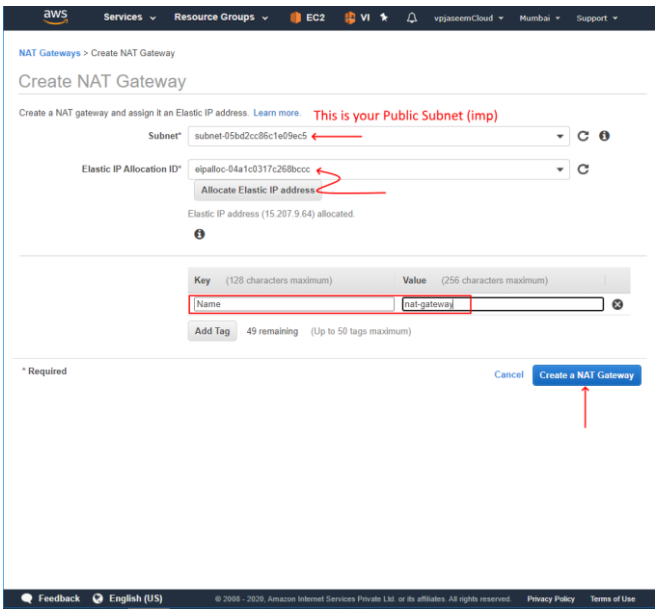


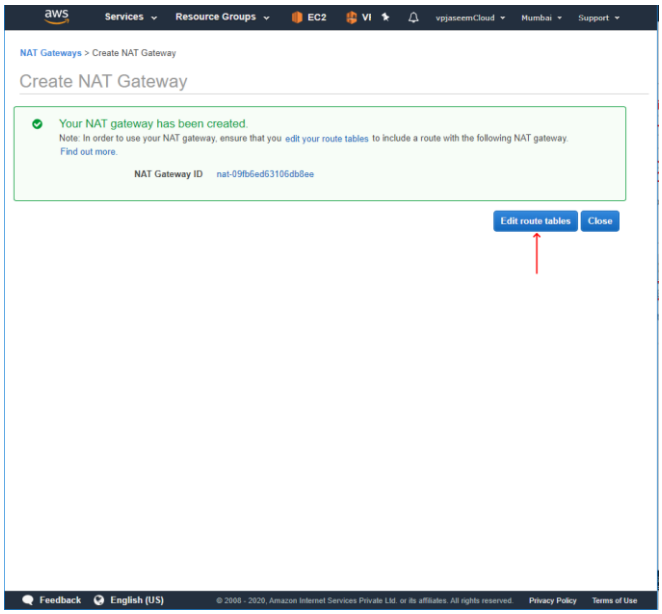



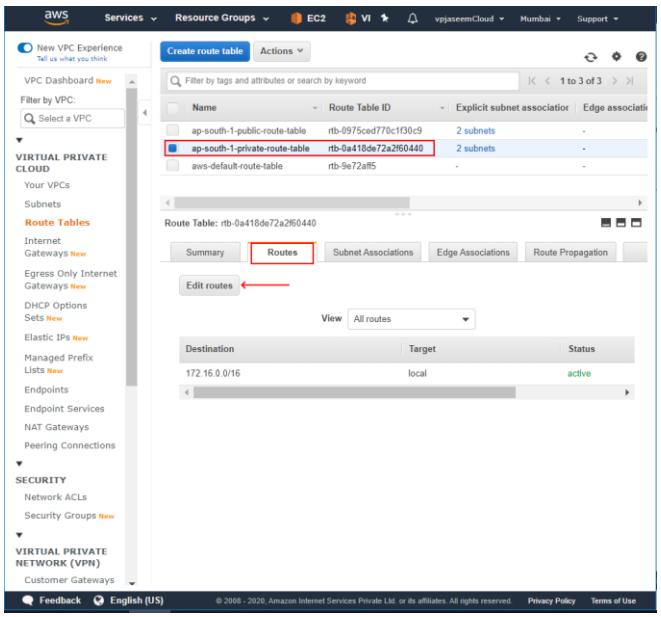

- RDP to the EC2 in Public Subnet and then RDP to EC2 in Private Subnet, then try to access internet. Yes, we don't have internet connectivity.

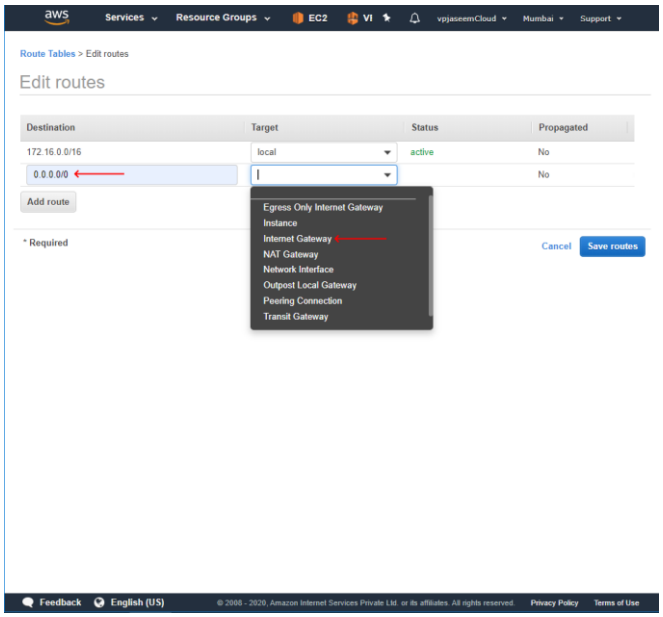


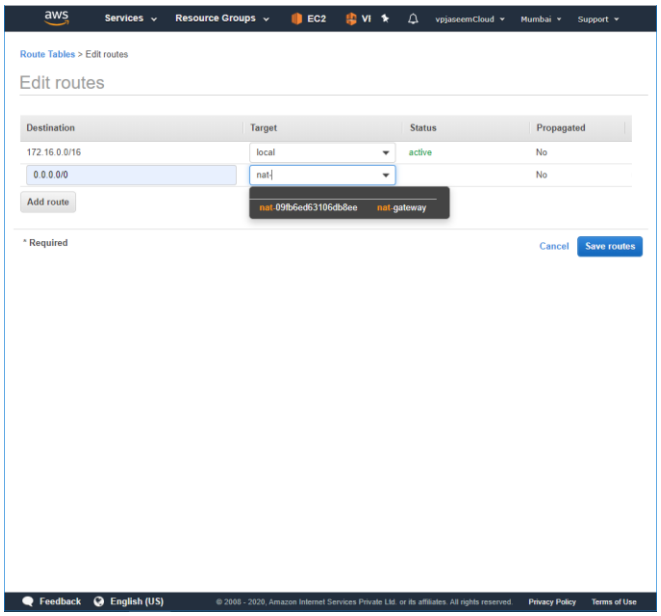

Create NAT Gateway
Actions
VPC Dashboard
You do not have any NAT Gateways in this region
Click the Create NAT Gateway button to create your first NAT Gateway
Create NAT Gateway
NAT Gateways
Peering Connections

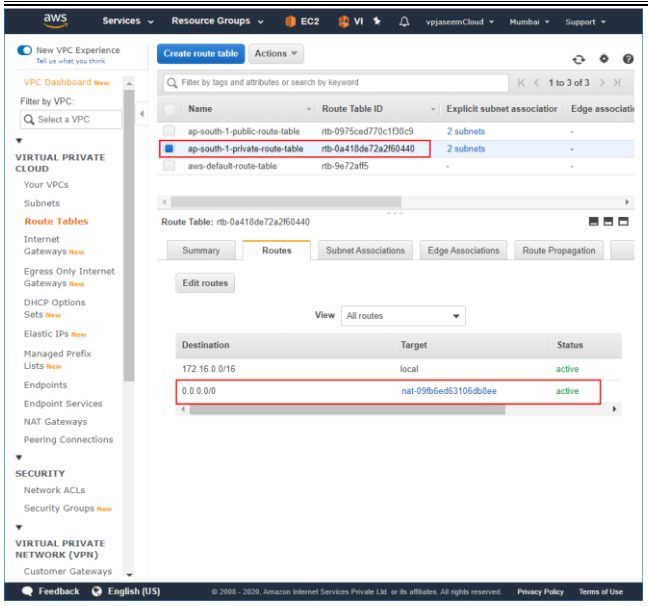

NAT Gateways > Create NAT Gateway
Create NAT Gateway
This is your Public Subnet (imp)
Subnet*
Elastic IP Allocation ID*
Allocate Elastic IP address
Key
Value
Name
nat-gateway
Add Tag
Cancel
Create a NAT Gateway


NAT Gateways > Create NAT Gateway
Create NAT Gateway
Your NAT gateway has been created.
Edit route tables
Close

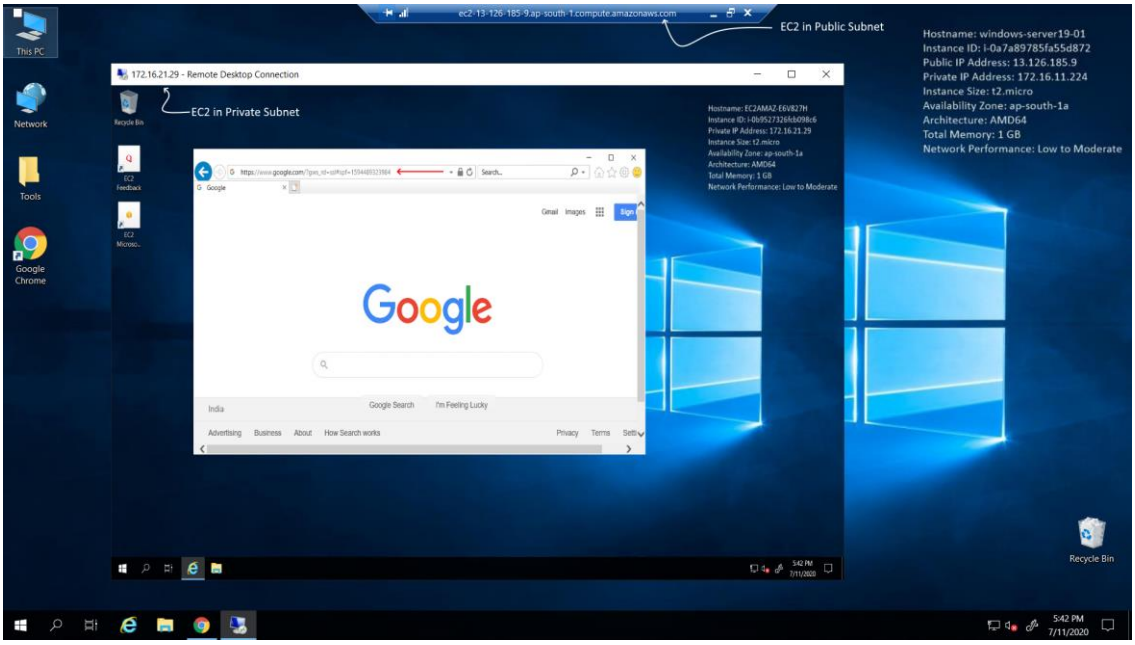

Create route table
Actions
Route Tables
Route Table: rtb-0a418de72a2f60440
Summary
Routes
Subnet Associations
Edge Associations
Route Propagation
Edit routes
172.16.0.0/16
local
active

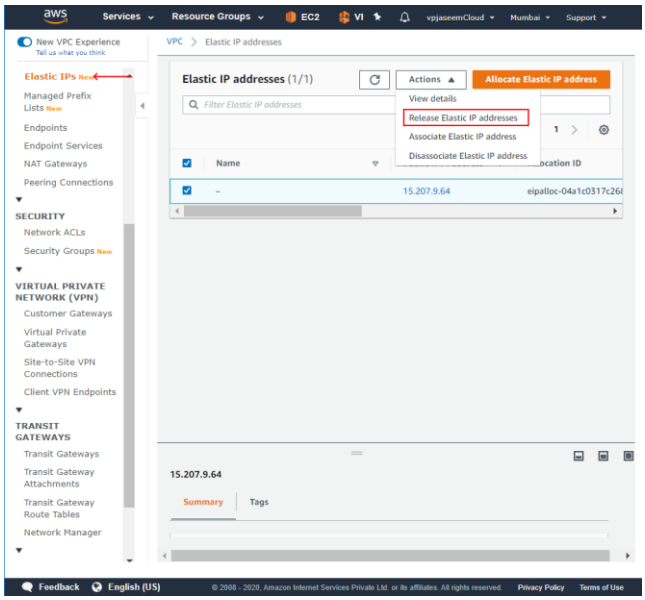

Route Tables > Edit routes
Edit routes
Destination
Target
Status
Propagated
172.16.0.0/16
local
active
No
0.0.0.0/0
Add route
Egress Only Internet Gateway
Instance
Internet Gateway
NAT Gateway
Network Interface
Outpost Local Gateway
Peering Connection
Transit Gateway
Cancel
Save routes


Route Tables > Edit routes
Edit routes
Destination
Target
Status
Propagated
172.16.0.0/16
local
active
No
0.0.0.0/0
nat
nat-09fb6ed63106db8ee
nat-gateway
Add route
Cancel
Save routes




Note: NAT Gateways are chargeable, hence delete immediately once the lab is completed



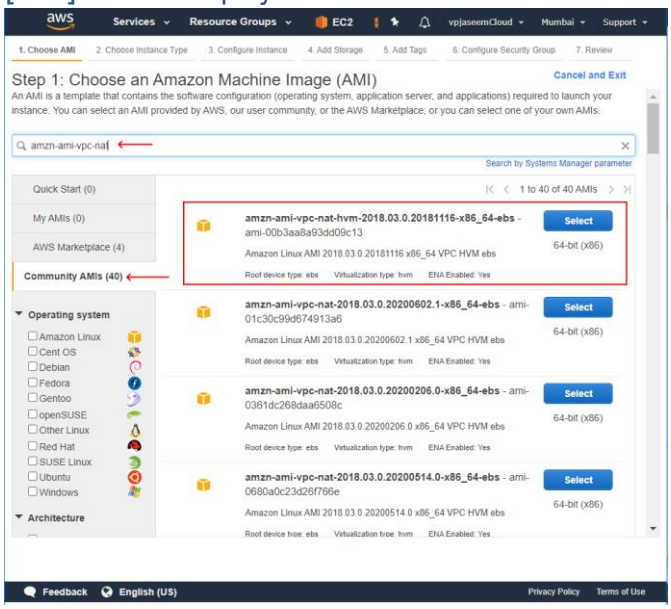



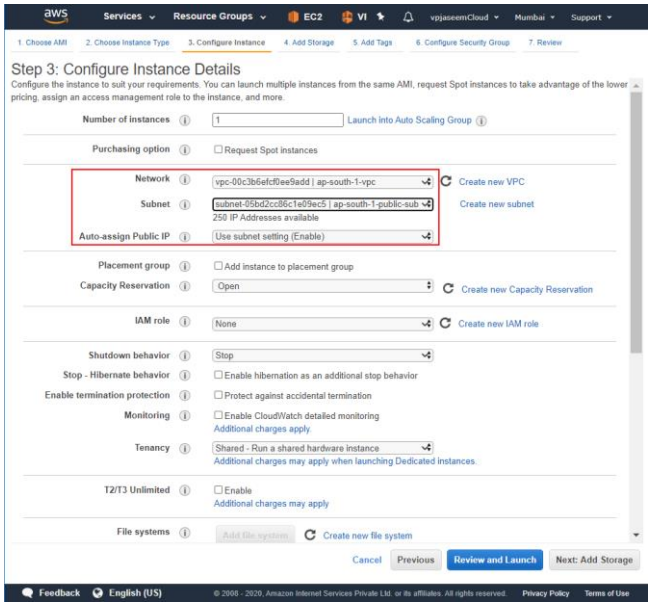

- NAT Gateway is assigned with an Elastic IP address, when an Elastic IP is not being used, it is chargeable, hence release the Elastic IP once lab is done.
- Deleting NAT Gateway won't release elastic IP, we need to release it manually.

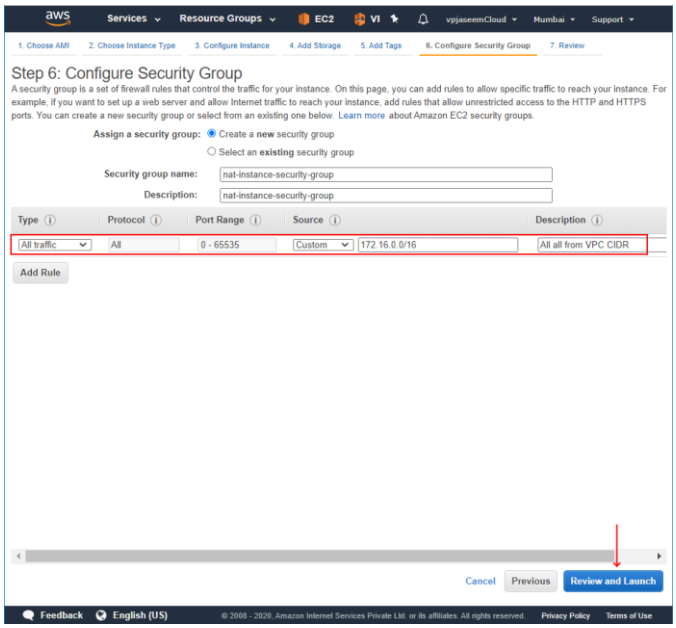
## NAT Instance

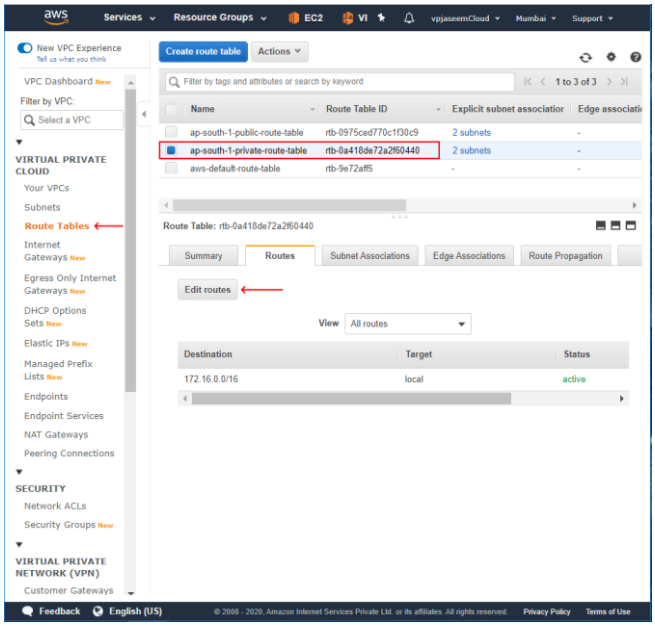
- It an EC2 instance, same functionality what NAT Gateway does, but is not AWS managed
- It is an Amazon customized Linux AMI with preconfigured NAT rules.
- AMIs include the string **amzn-ami-vpc-nat** in their names, so that you can identify them in the Amazon EC2 console or search for them using the AWS CLI.
- NAT instance should be launched in Public Subnet.
- Since t2.micro NAT instance is free, it is a better alternative for NAT Gatreway.

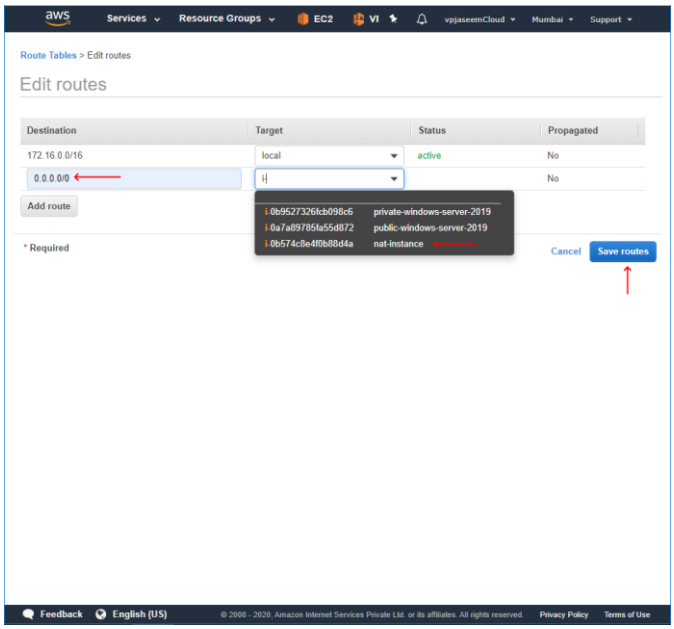
# [LAB] How to Deploy NAT Instance

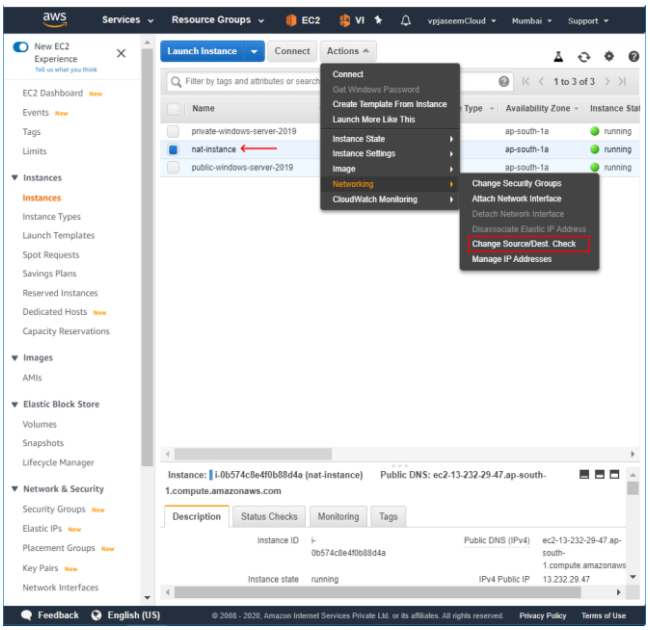


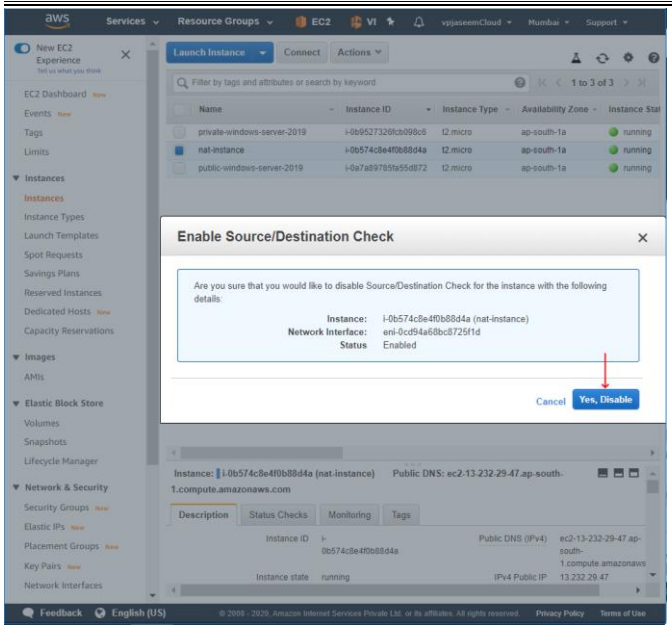



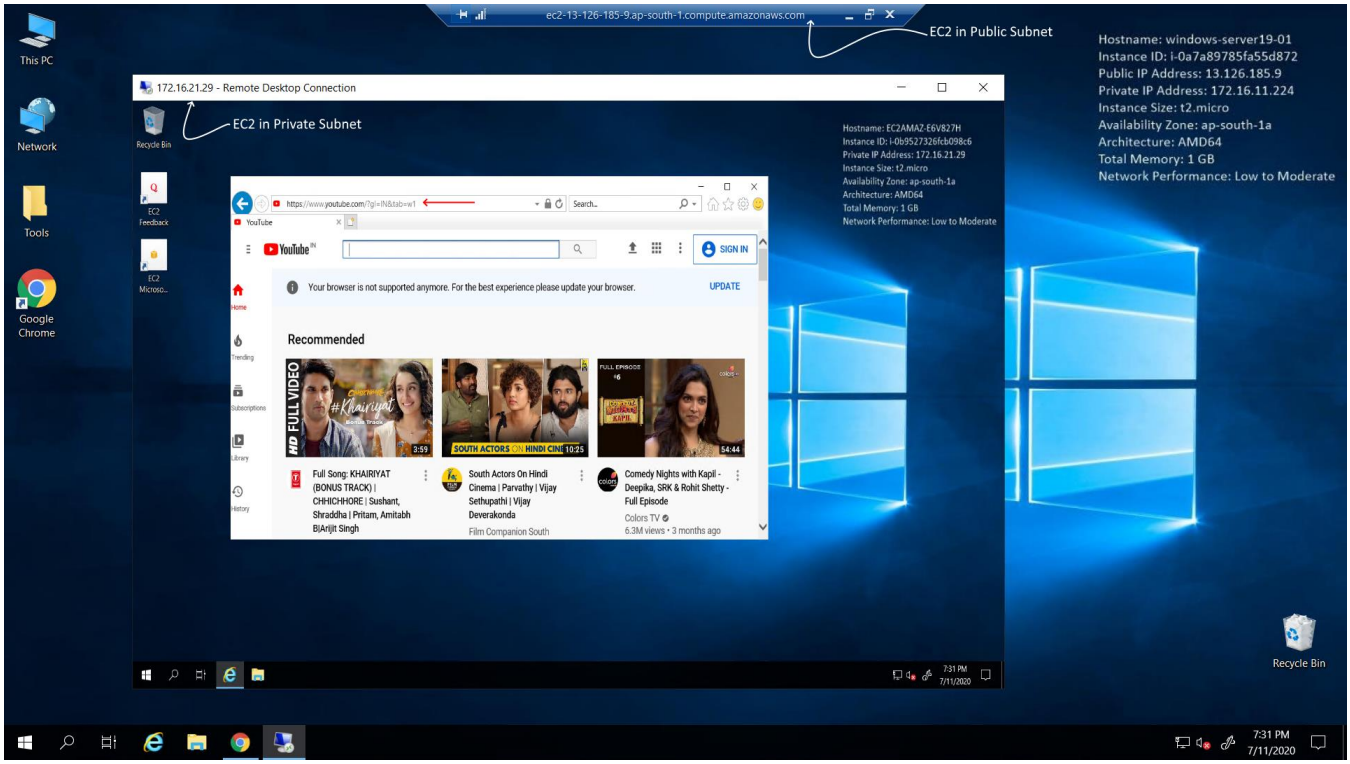



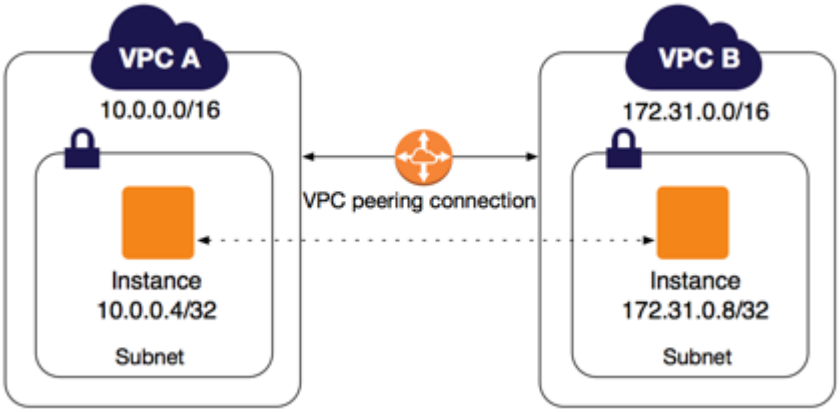




Enable Source/Destination Check
Are you sure that you would like to disable Source/Destination Check for the instance with the following details:
Instance: i-0b574c8e4f0b88d4a (nat-instance)
Network Interface: eni-0cd94a68bc8725f1d
Status Enabled
Cancel
Yes, Disable


ec2-13-126-185-9.ap-south-1.compute.amazonaws.com
EC2 in Public Subnet
172.16.21.29 - Remote Desktop Connection
EC2 in Private Subnet
Hostname: windows-server19-01
Instance ID: i-0a7a89785fa55d872
Public IP Address: 13.126.185.9
Private IP Address: 172.16.11.224
Instance Size: t2.micro
Availability Zone: ap-south-1a
Architecture: AMD64
Total Memory: 1 GB
Network Performance: Low to Moderate
Hostname: EC2AMAZ-E6V827H
Instance ID: i-0b9527326fcb098c6
Private IP Address: 172.16.21.29
Instance Size: t2.micro
Availability Zone: ap-south-1a
Architecture: AMD64
Total Memory: 1 GB
Network Performance: Low to Moderate
Your browser is not supported anymore. For the best experience please update your browser.
UPDATE
Recommended
Full Song: KHAIRIYAT (BONUS TRACK) | CHHICHHORE | Sushant, Shraddha | Pritam, Amitabh B|Arijit Singh
South Actors On Hindi Cinema | Parvathy | Vijay Sethupathi | Vijay Deverakonda
Film Companion South
Comedy Nights with Kapil - Deepika, SRK & Rohit Shetty - Full Episode
Colors TV
6.3M views • 3 months ago
This PC
Network
Tools
Google Chrome
Recycle Bin
7:31 PM
7/11/2020

## VPC Peering

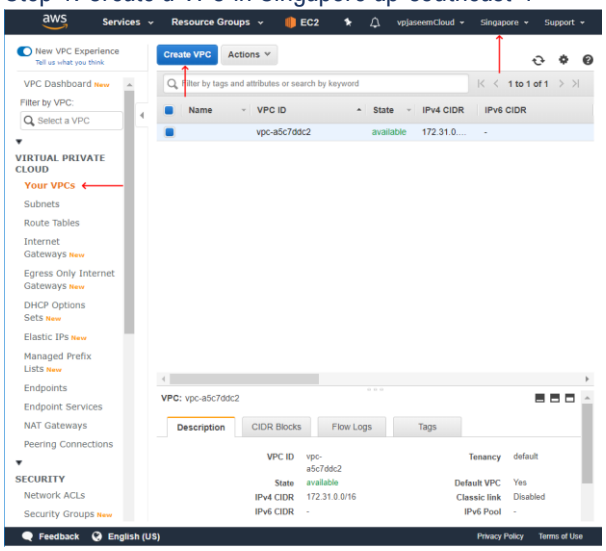


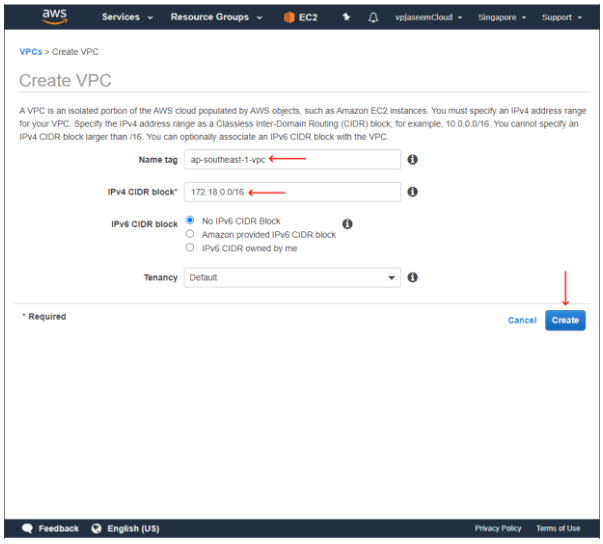

- We interconnect 2 VPCs together (e.g. when one company acquire another one)
- Without VPC, any given AWS resources need to travel though internet and talk to other AWS resource.
- Initially VPC peering works within the region but as of today we can do cross region VPC peering.
- There shouldn't be any overlapping CIDR block, no transitive peering (A peer to B and B peer to C, that doesn't mean A can talk to C)
- IGW access across VPC is not possible, no NAT routing between VPCs.

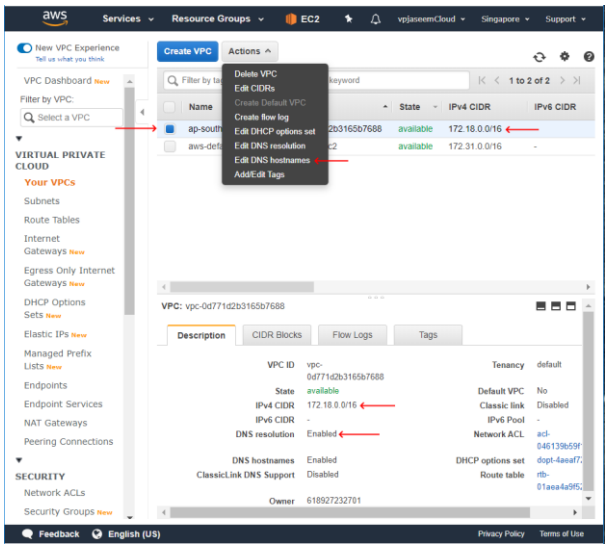
## [LAB] VPC Peering Cross Region

- We already have a VPC with CIDR range 172.16.0.0/16 belongs to ap-south-1 region, go ahead and create another new VPC with CIDR 172.18.0.0/16 on another region (e.g. Singapore ap-southeast-1)

### Step 1: Create a VPC in Singapore ap-southeast-1

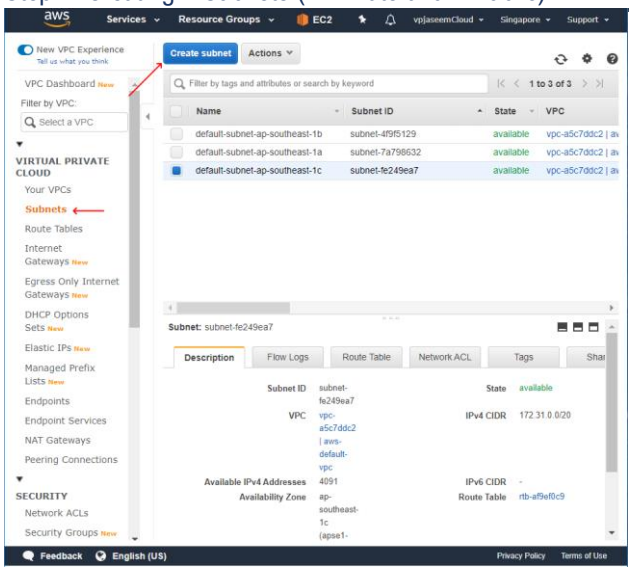


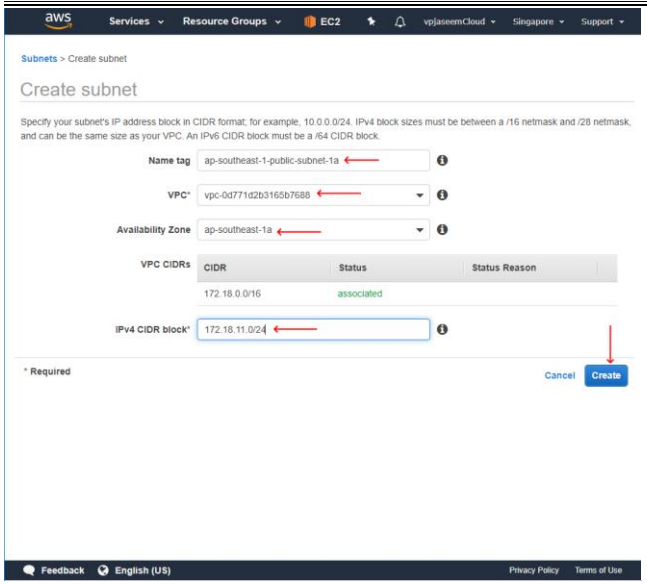




- Enable Public DNS Hostnames assignments

### Step 2: Creating 4 Subnets (2 Private and 2 Public)



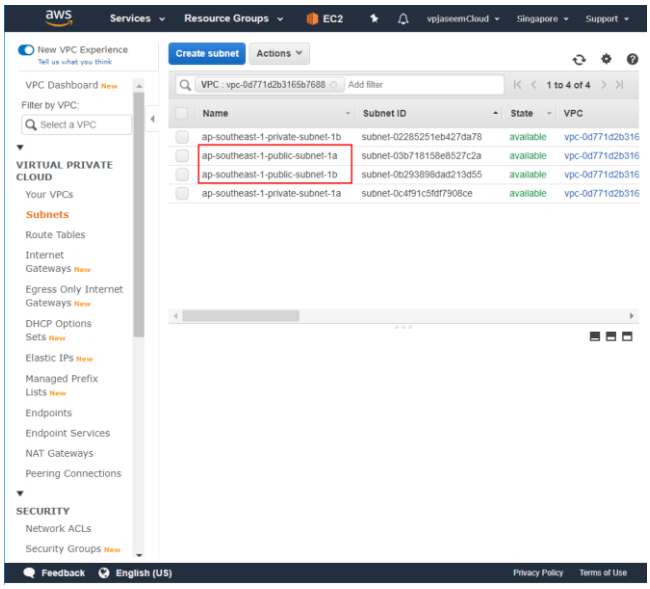

- I have given some friendly names to the default Subnets.
- Go ahead and click Create Subnet button
- In our lab practice we are creating 4 subnets (2 Private in 2 AZs and 2 Public in 2 AZs)
  - ap-southeast-1-public-subnet-1a
  - ap-southeast-1-public-subnet-1b
  - ap-southeast-1-private-subnet-1a
  - ap-southeast-1-private-subnet-1b

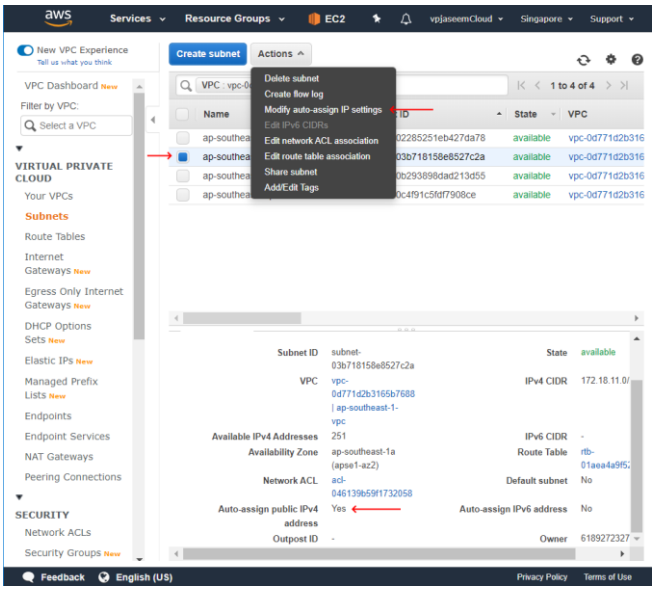



I have configured below subnets,

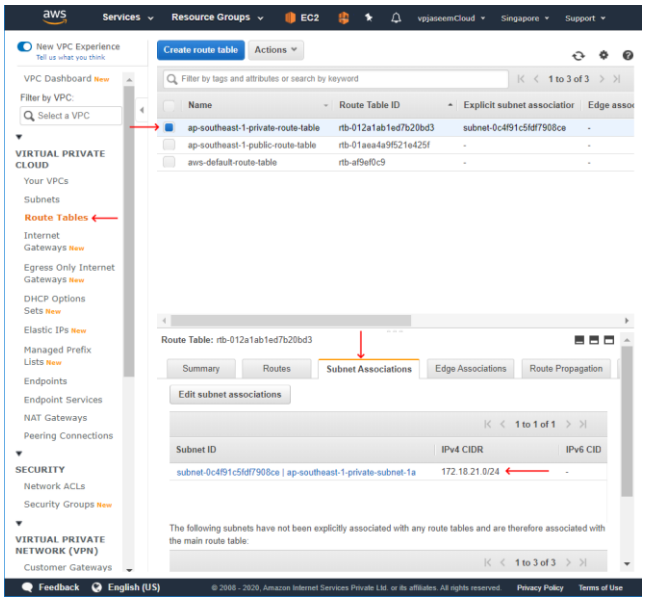
| SUBNET | CIDR |
|---|---|
| ap-southeast-1-public-subnet-1a | 172.18.11.0/24 |
| ap-southeast-1-public-subnet-1b | 172.18.12.0/24 |
| ap-southeast-1-private-subnet-1a | 172.18.21.0/24 |
| ap-southeast-1-private-subnet-1b | 172.18.22.0/24 |

## Step 3: Auto-assign public IPv4 address for Public Subnets



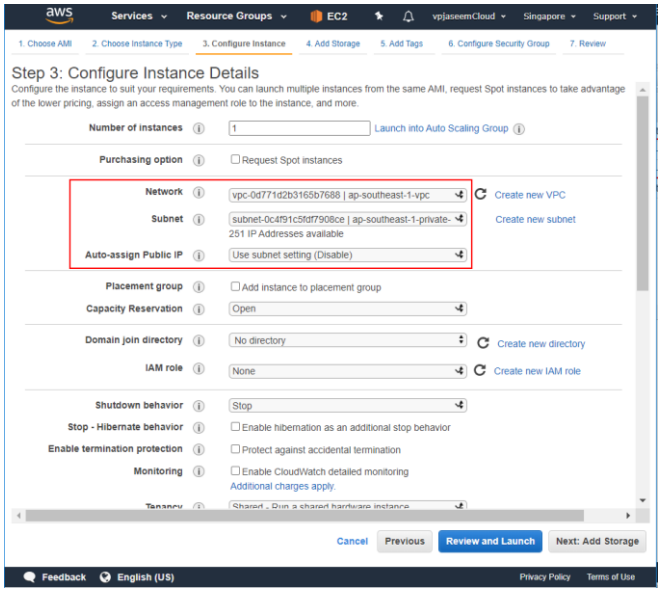



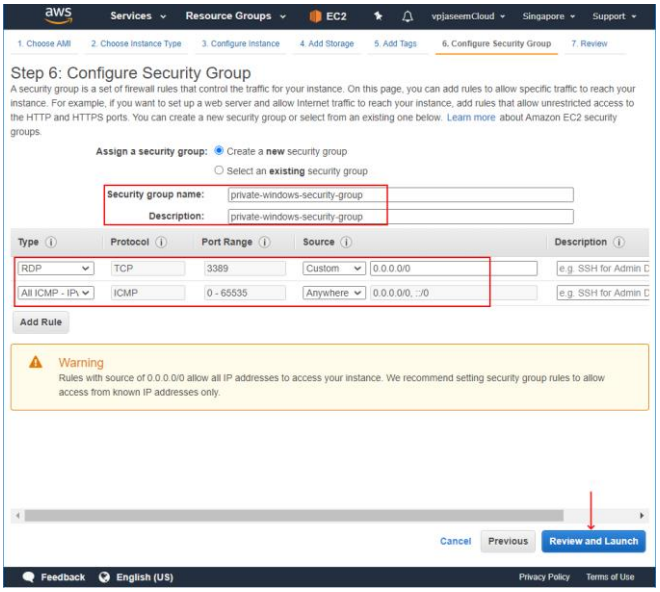

## Step 4: Create 2 Route Tables (Public and Private) in Singapore VPC



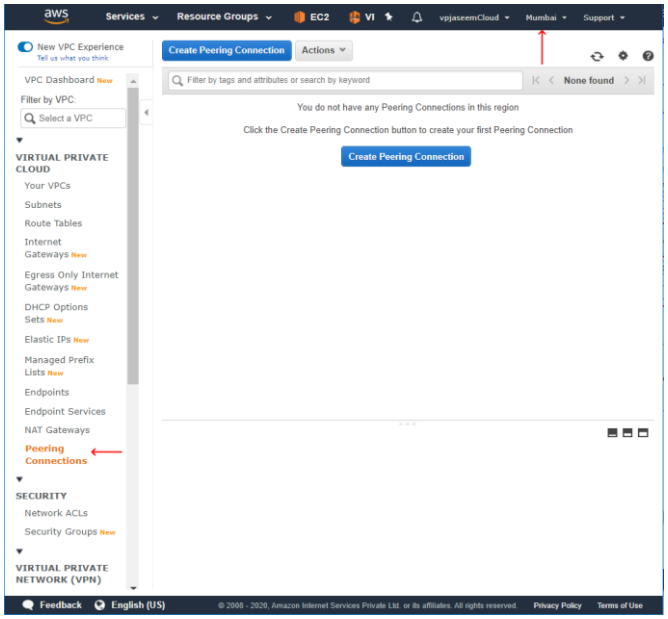

- I just interested in one Route Table (Private) because I'm going to launch my instance in Singapore Private Subnet.
- Make sure in the Private Subnet is attached to the Private Route Table.

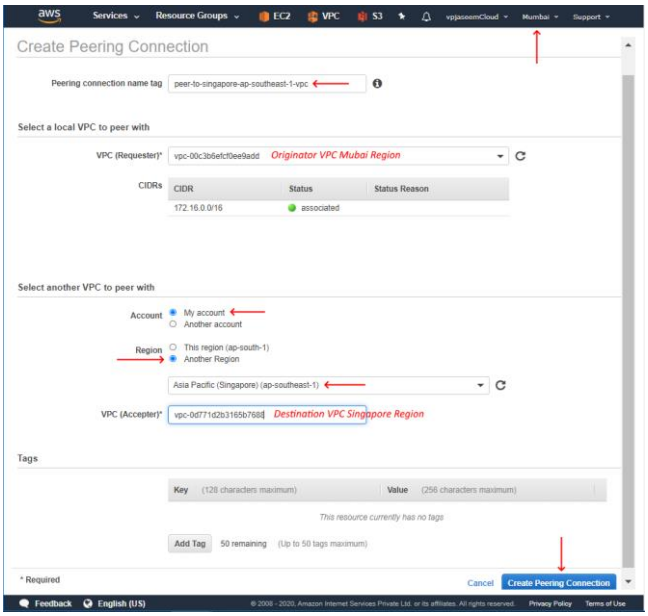
## Step 5: Launch a Windows EC2 in Private Subnet

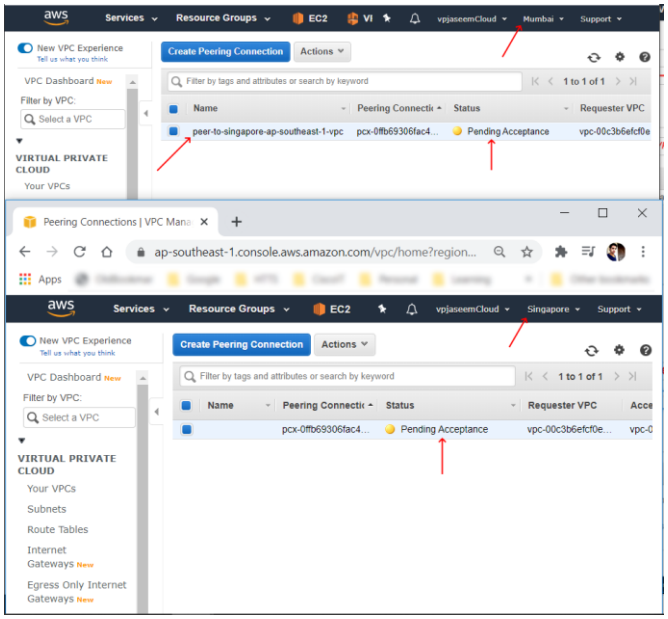


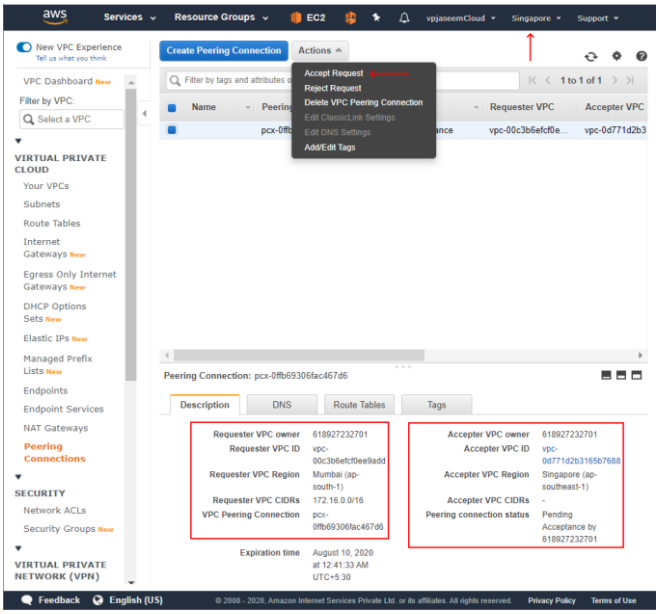



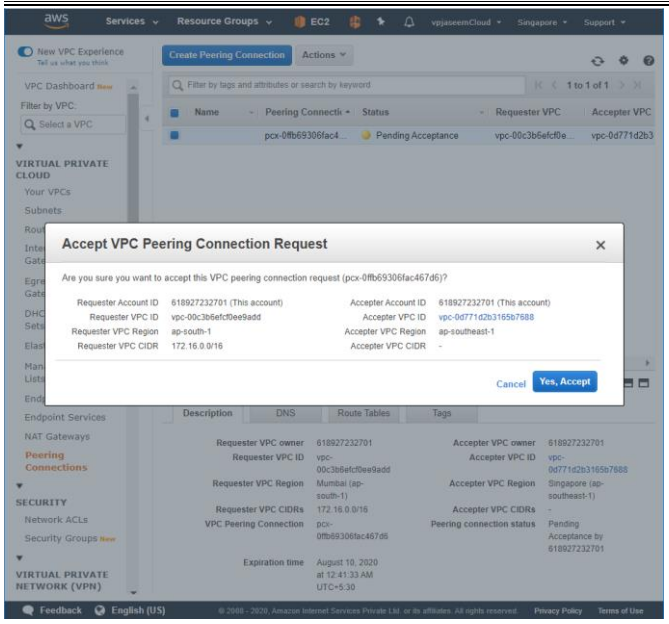

## Step 6: Create a Peering Connection from Mumbai to Singapore

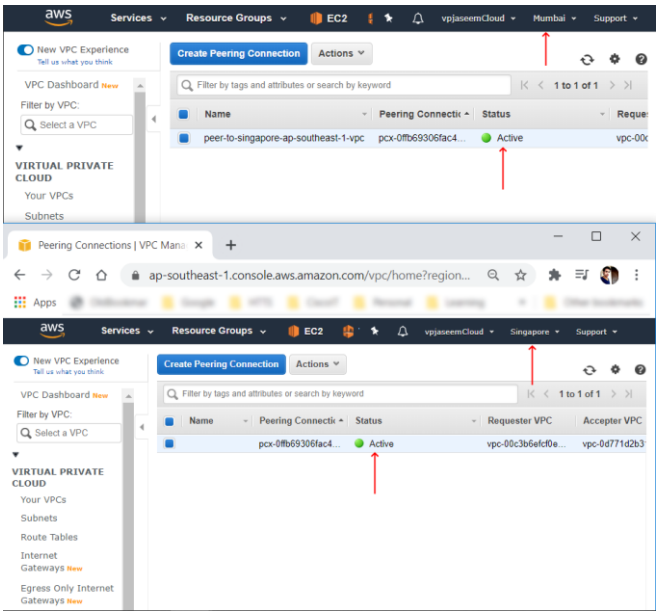


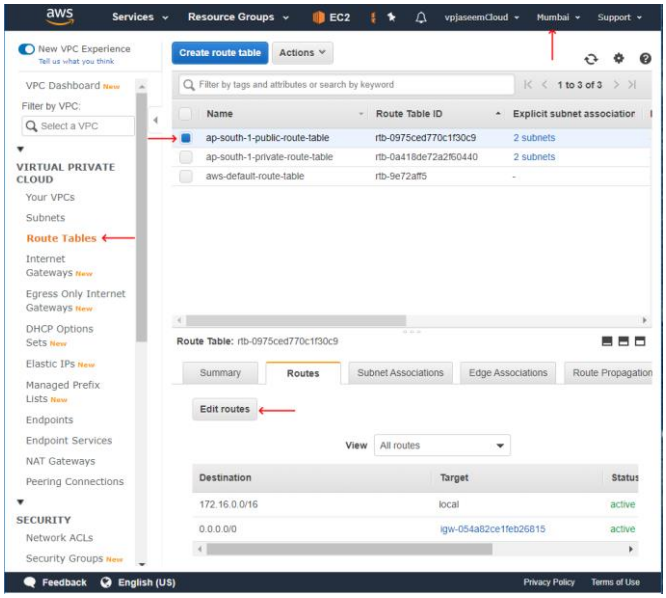



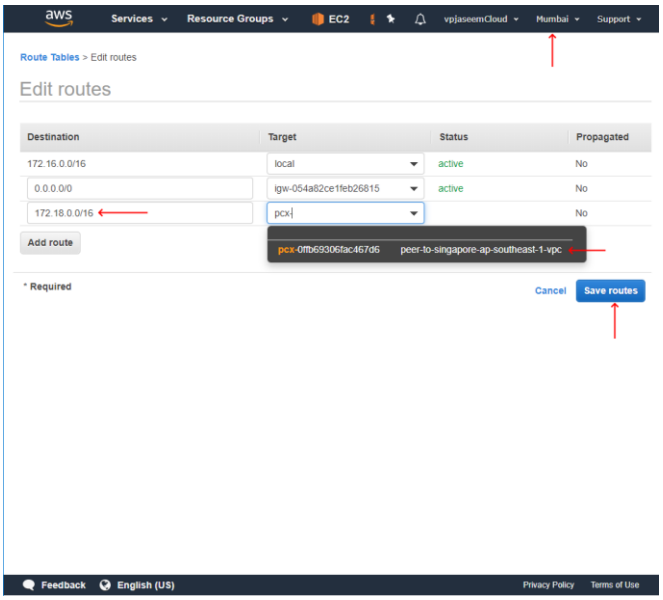





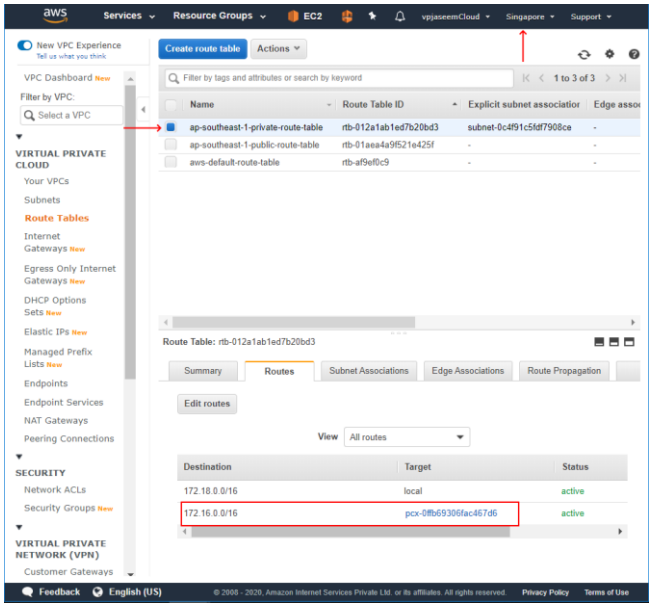




## Step 7: Update Mumbai Public and Private Routables



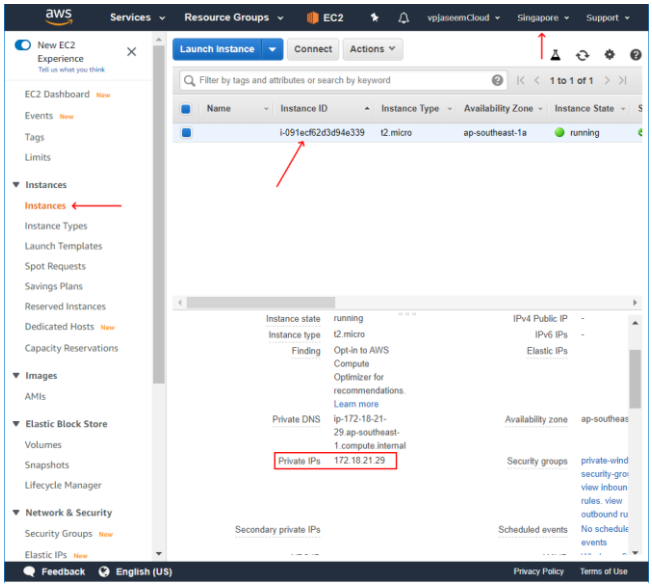




## Step 8: Update Singapore Private Route Table

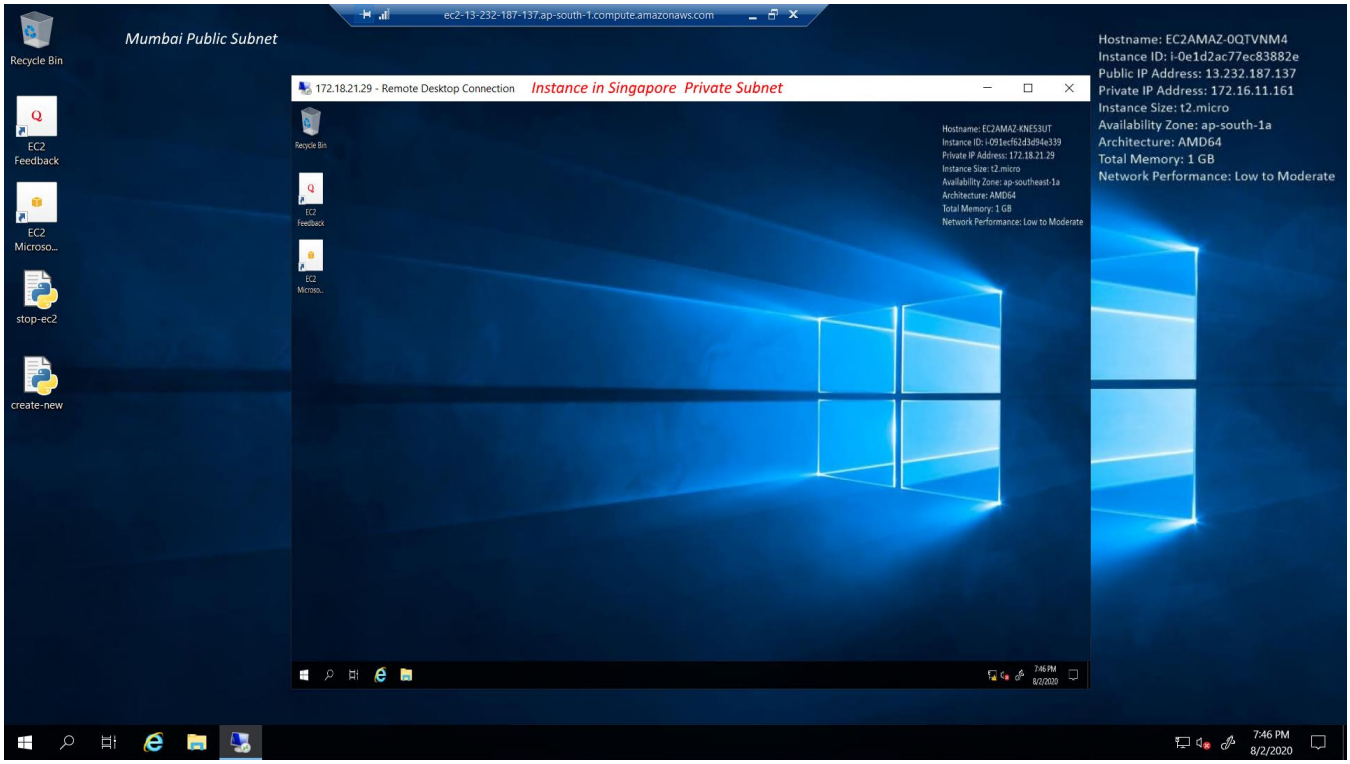



- In Singapore Private Route table, add the CIDR of Mumbai VPC and point to Peering Connection.

## Step 9: Access Singapore EC2 instance from Mumbai Instance




- We may not get ICPM response since ICMP is blocked in Windows firewall by default, hence try RDP to Singapore VPC instance.



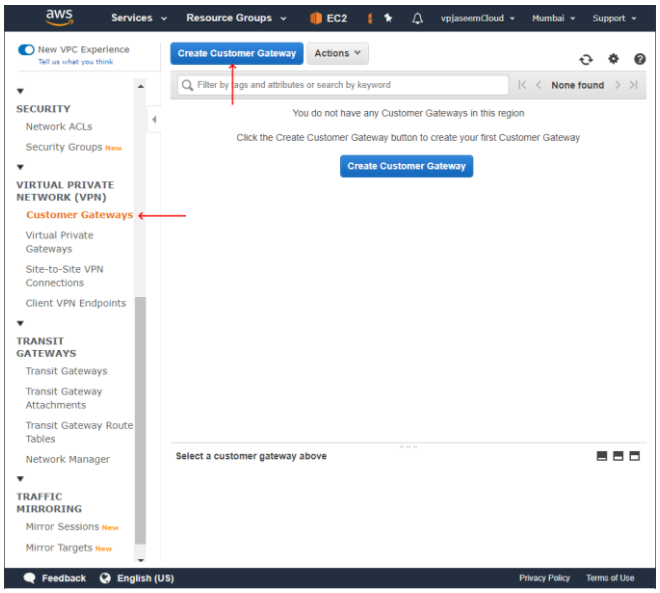
## VPC IPSec Virtual Private Network (VPN)

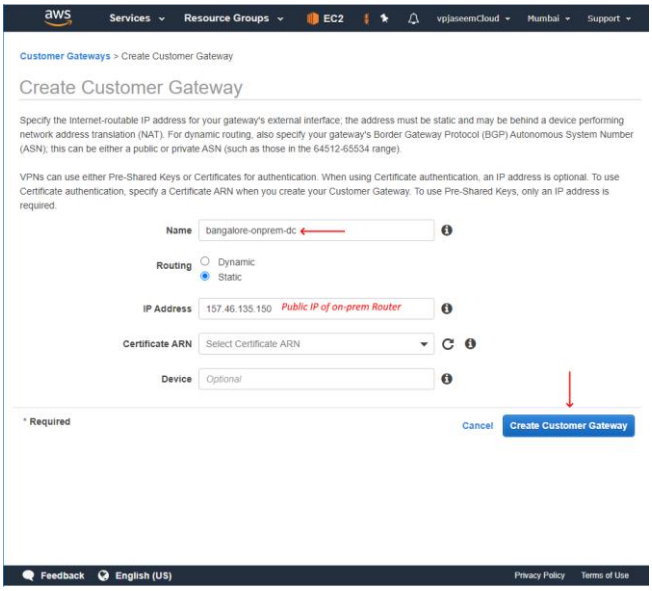
- Create secure private link between AWS VPC and on-prem data center.
- Access VCP resources with its Private IP address from the on-prem data center.
- AWS VPN is comprised of two services: AWS Site-to-Site VPN and AWS Client VPN. Together, they deliver a highly-available, managed, and elastic cloud VPN solution to protect your network traffic.
- Customer Gateway: Logical representation of on-premise end of VPN
- AWS VPN gateway: AWS end of VPN connection

AWS Reference: Site to Site VPN

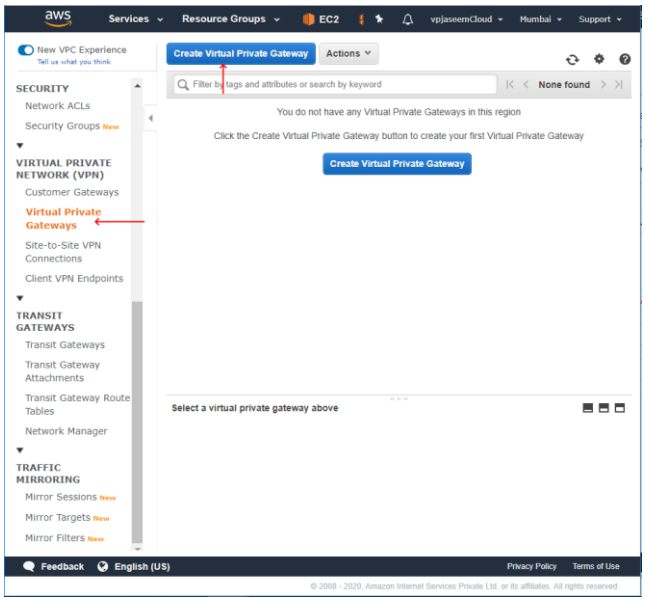
# [LAB] VPC IPSec VPN Configuration

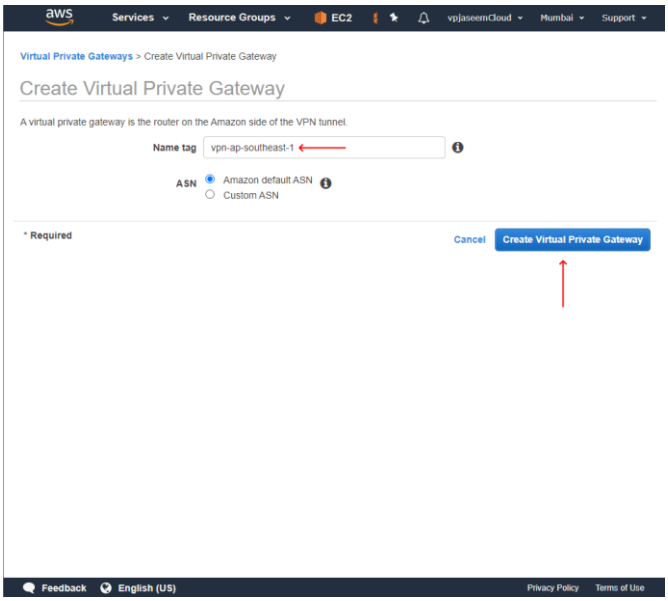
## Step 1: Create Customer gateway

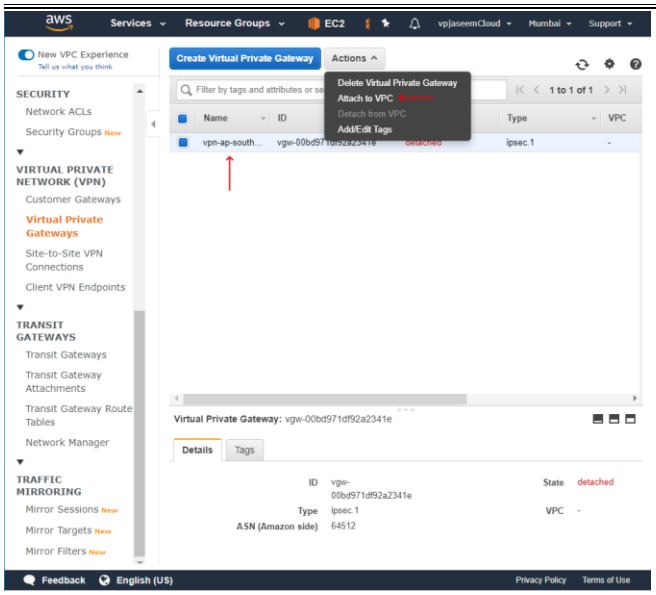


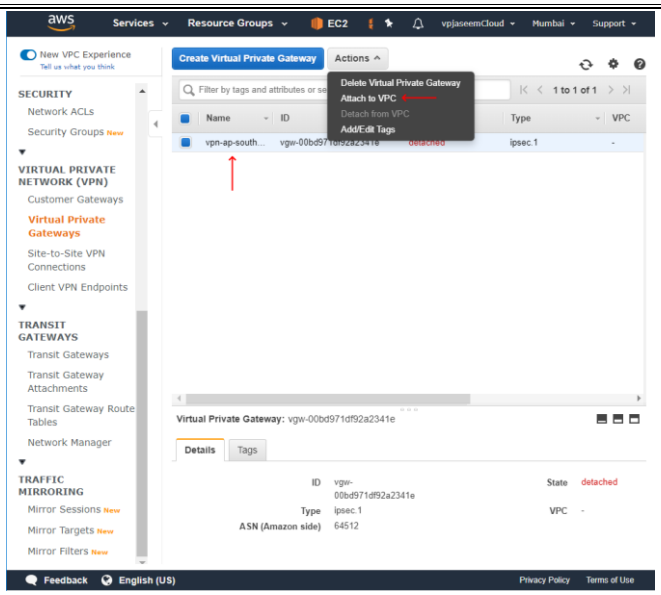



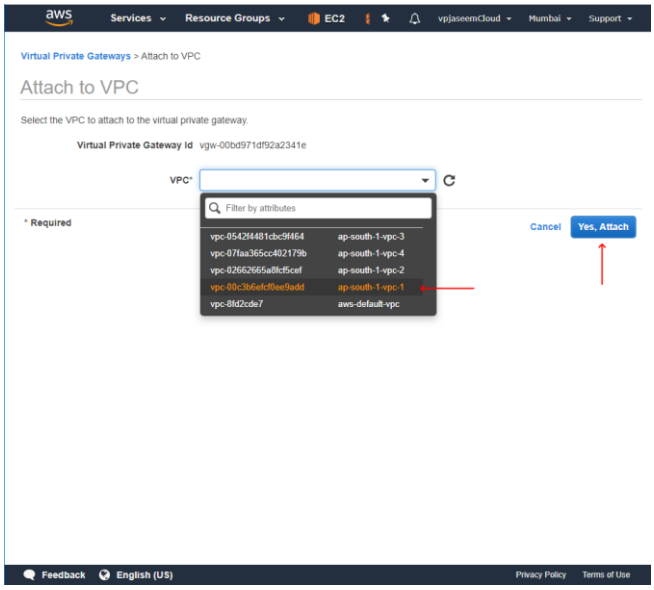

## Step 2: Create VPN Gateway & Attach to VPC



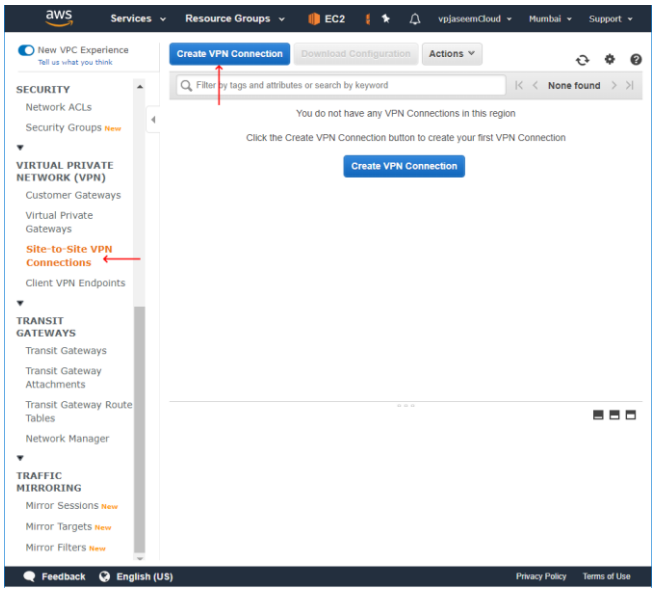



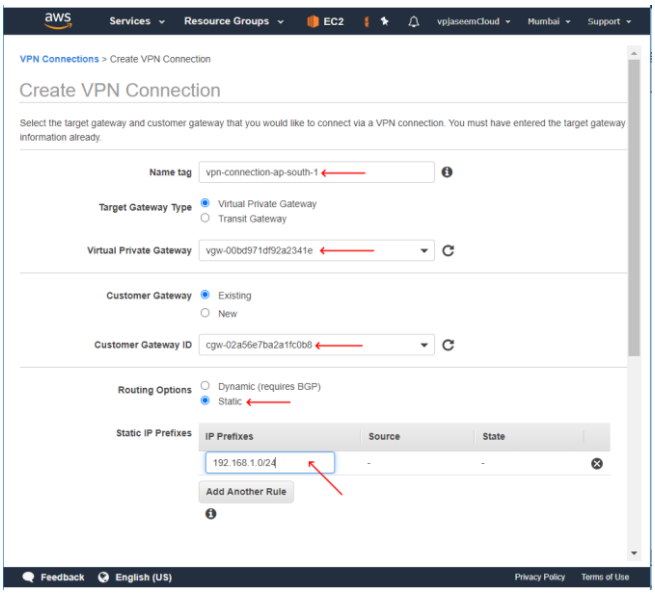


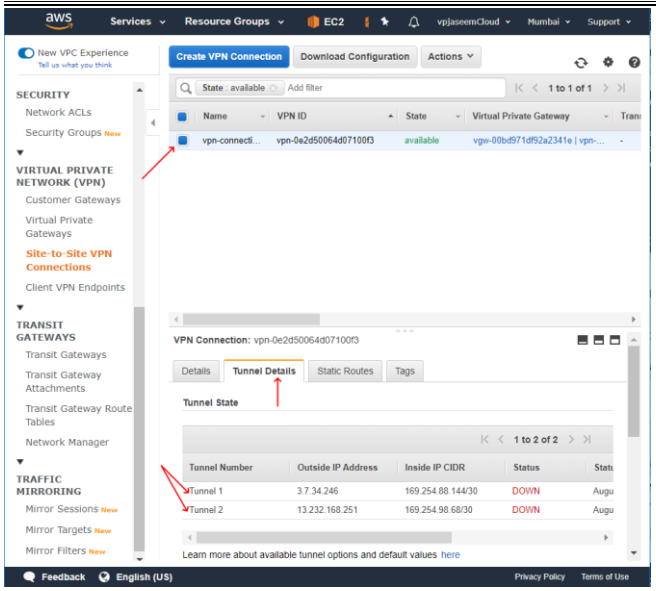



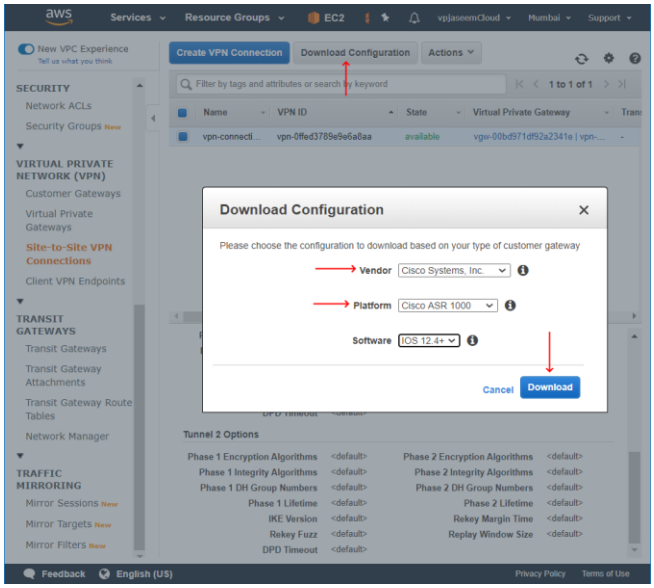

## Step 3: Create Site to Site VPN Connection

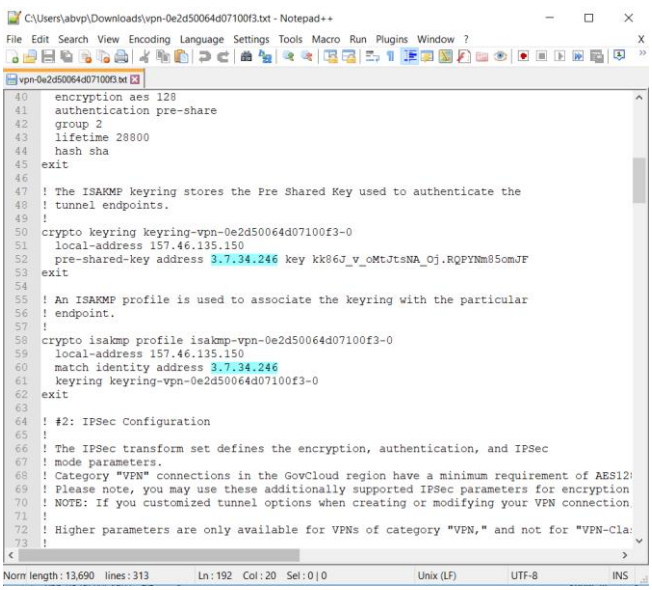





## Step 4: Download the Vendor Specific Configuration



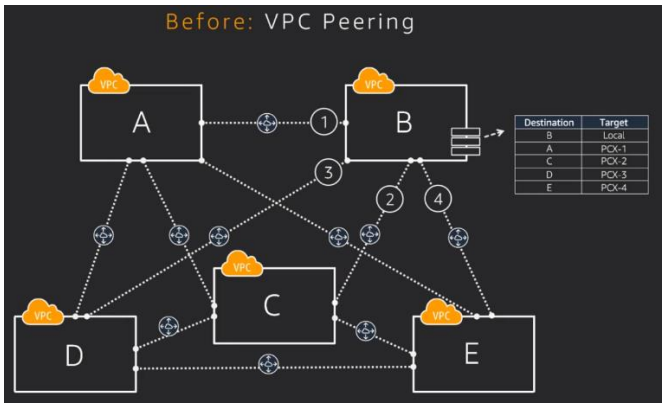
## VPC Direct Connect

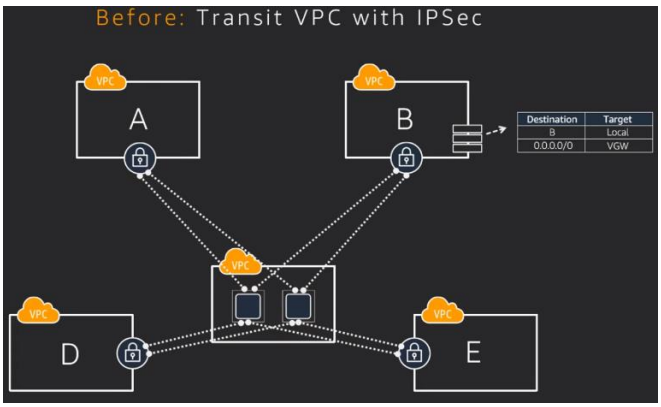
- Dedicated network connection from on-prem DC to AWS, better speed, and bandwidth than VPN.
- sing AWS Direct Connect, you can establish private connectivity between AWS and your datacenter, office, or colocation environment, which in many cases can reduce your network costs, increase bandwidth throughput, and provide a more consistent network experience than Internet-based connections.

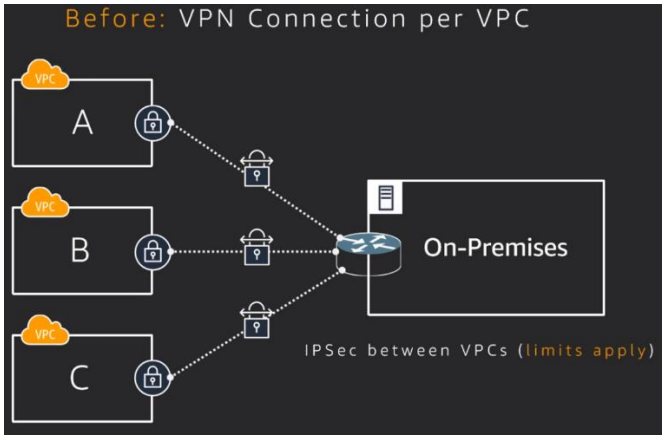
AWS Reference: AWS Direct Connect

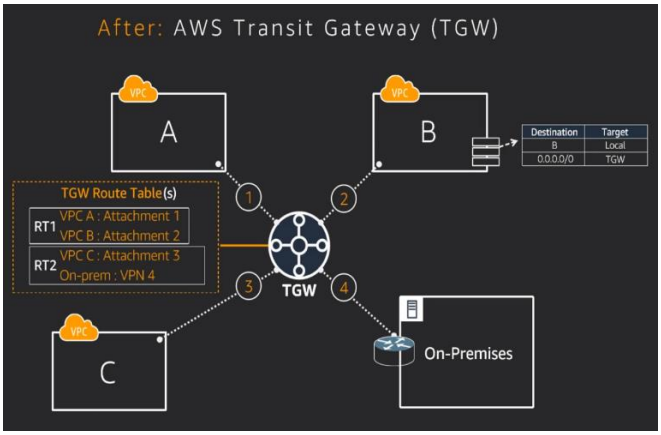
## AWS Transit Gateway (TGW)

- A Transit Gateway (TGW) is a network transit hub that interconnects attachments (VPCs and VPNs) within the same account or across accounts.
- When we have multiple VPCs, peering becomes challenging, we end up building a complete peer full mesh.
- TGW scope is Region
- Consider we have n VPCs, then, Total number of Peering connections = $\frac{n(n-1)}{2}$
- Limit for Static Routes per VPC route table = 100; and Peering connection per VPC = 125



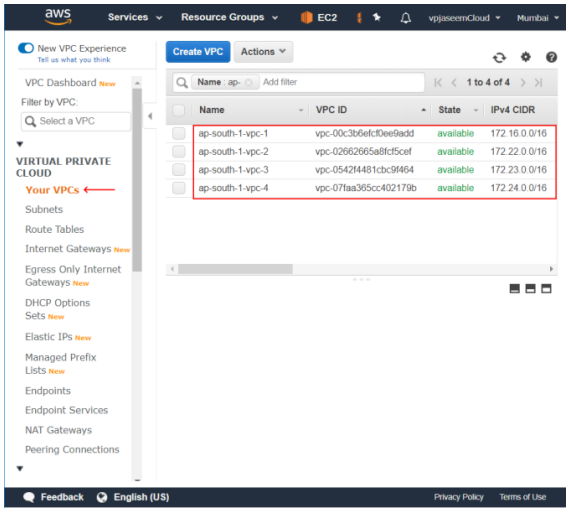



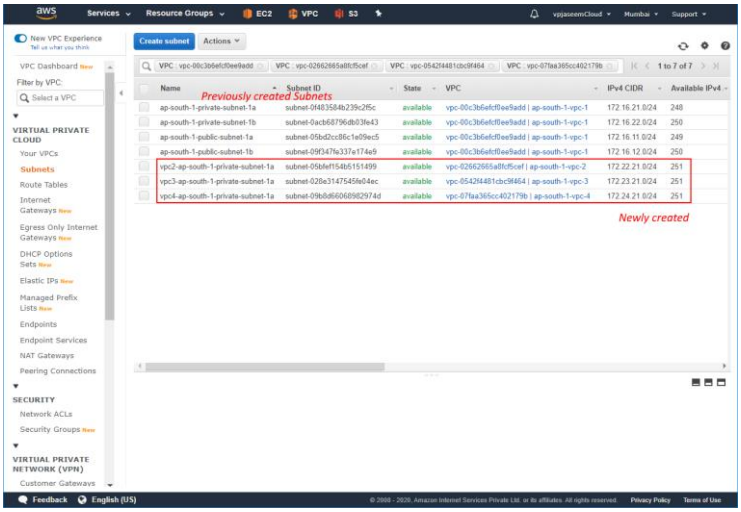

- Transit gateway works as Hub and Spoke model, all the VPCs are attached to the Transit Gateway
- On-Prem DC can also be attached to Transit Gateway
- We can define rules Transit Gateway regarding the routes
- **Attachment**: Connection between VPC and Transit Gateway (VPN to TGW). Attach one Subnet per AZ in VPC
- **Association**: Route table used to route traffic coming from an attachment. One attachment uses one Route table.
- **Propagation**: Route table where the attachment's routes are installed.

## [LAB] Transit Gateway

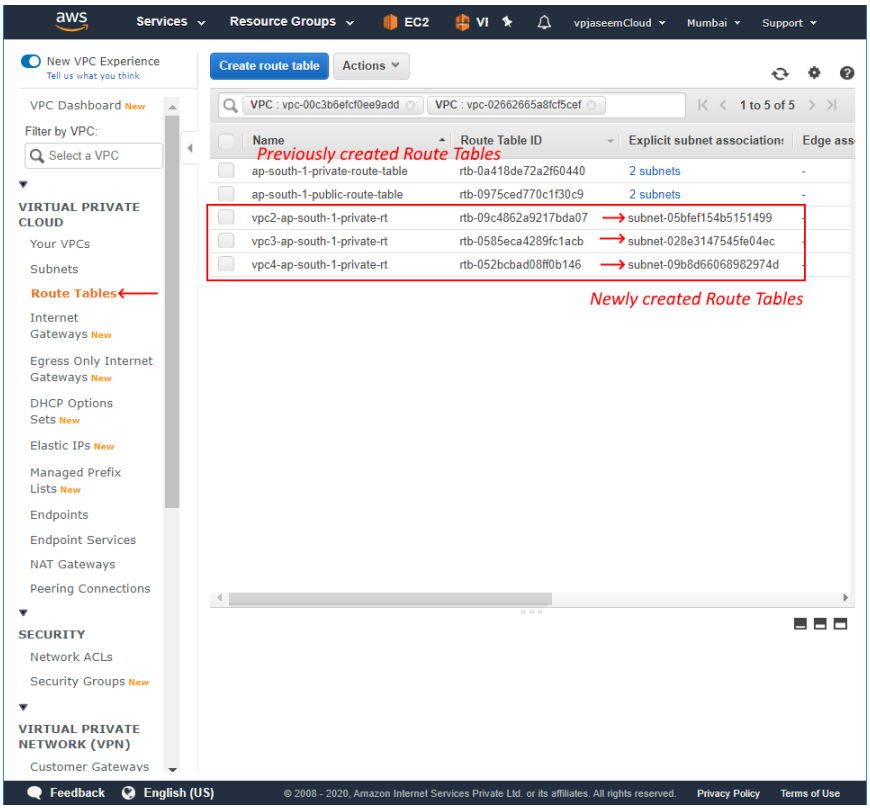
- Let's create 4 VPCs in same region (ap-south-1), one has Public and Private Subnets, the others have only 1 private subnet.
- Assuming we have 1 VPC already and a Windows EC2 instance running on it with Public IP.

### Step 1: Create 3 VPCs, 3 Subnets, edit 3 default route table



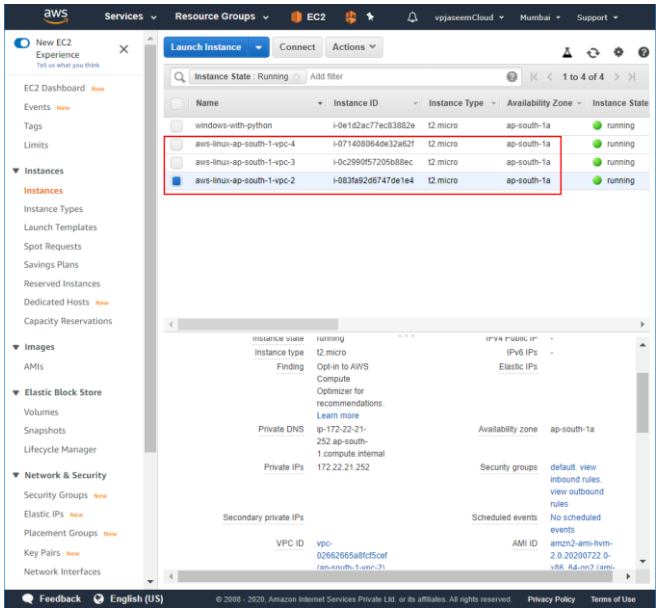

| VPC Name | VPC CIDR | Subnet CIDR | Name of Subnet | Route Table |
|---|---|---|---|---|
| ap-south-1-vpc-1 | 172.16.0.0/16 | 172.16.11.0/24 | ap-south-1-public-subnet-1a | ap-south-1-public-route-table |
| | | 172.16.12.0/24 | ap-south-1-public-subnet-1b | |
| | | 172.16.21.0/24 | ap-south-1-private-subnet-1a | ap-south-1-private-route-table |
| | | 172.16.22.0/24 | ap-south-1-private-subnet-1b | |
| ap-south-1-vpc-2 | 172.22.0.0/16 | 172.22.21.0/24 | vpc2-ap-south-1-private-subnet-1a | vpc2-ap-south-1-private-rt |
| ap-south-1-vpc-3 | 172.23.0.0/16 | 172.23.21.0/24 | vpc3-ap-south-1-private-subnet-1a | vpc3-ap-south-1-private-rt |
| ap-south-1-vpc-4 | 172.24.0.0/16 | 172.24.21.0/24 | vpc4-ap-south-1-private-subnet-1a | vpc4-ap-south-1-private-rt |



## Step 2: Launch Amazon Linux EC2 on 3 VPC's Private Subnets

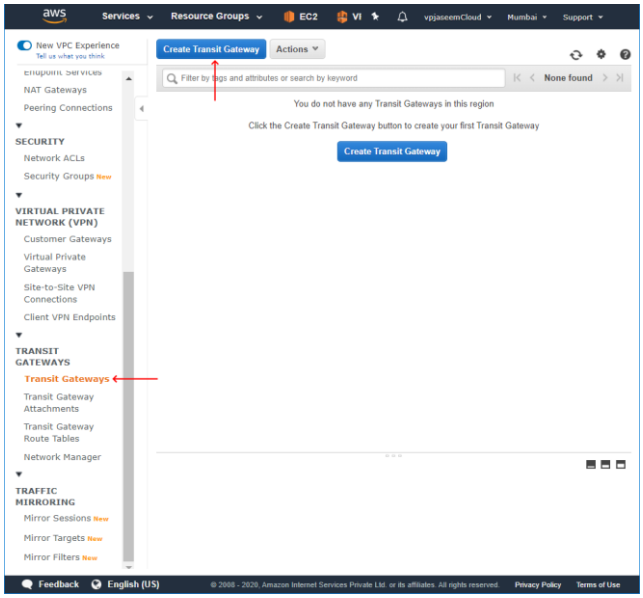
- Deploy Amazon Linux on each of the 3 VPCs, make sure to allow all traffic in Security Group

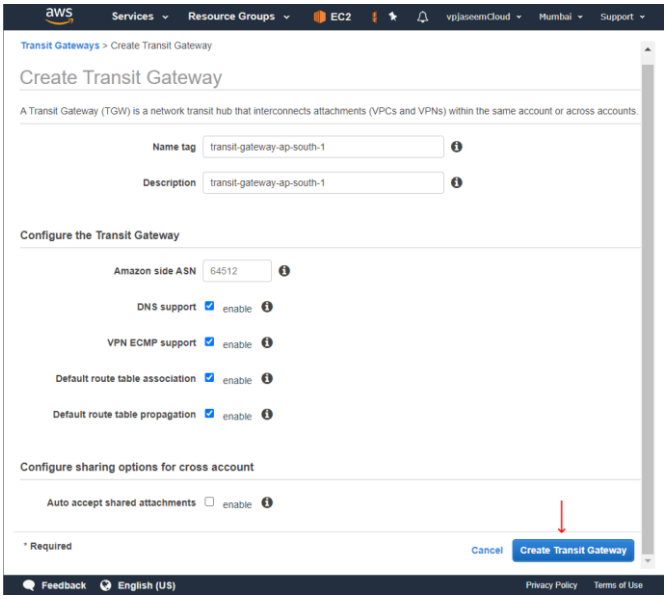
| VPC Name | EC2 Instance Name |
|---|---|
| ap-south-1-vpc-2 | aws-linux-ap-south-1-vpc-2 |
| ap-south-1-vpc-3 | aws-linux-ap-south-1-vpc-3 |
| ap-south-1-vpc-4 | aws-linux-ap-south-1-vpc-4 |



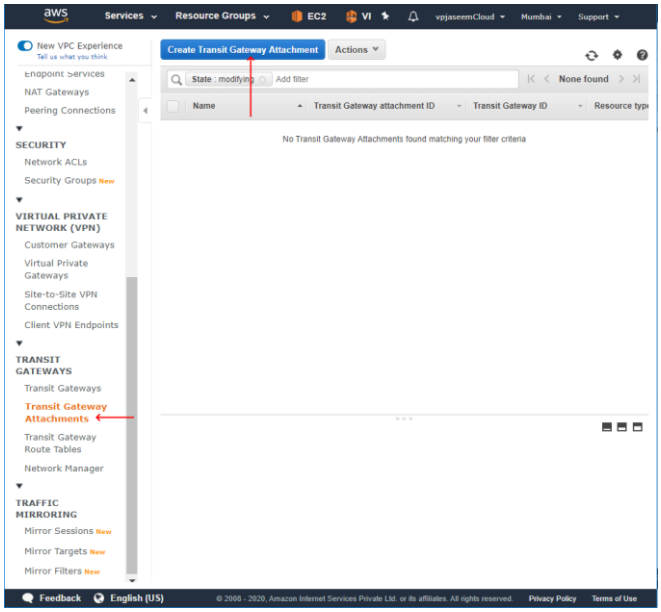

- We have a Windows Instance running on the Public Subnet (VPC1)
- As of now we are not able to ping any other Linux instances from the Windows since they are on a different VPCs
- Once we create Transit Gateway, we will be able to connect all of them.

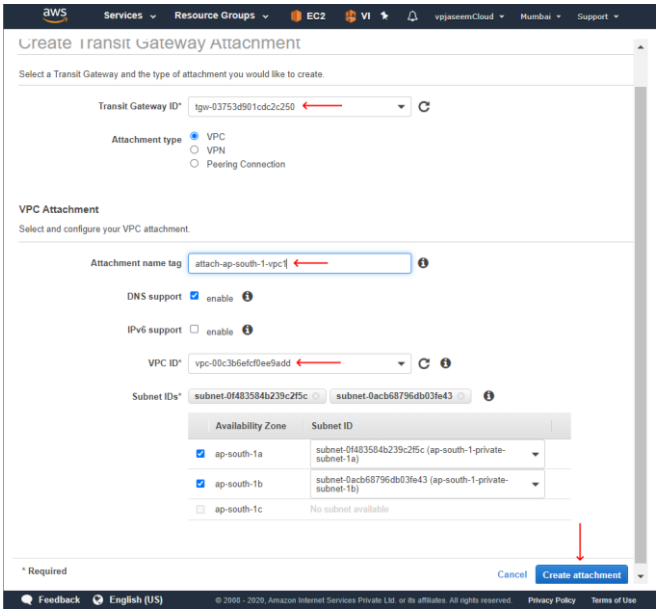
## Step 3: Creating Transit Gateway

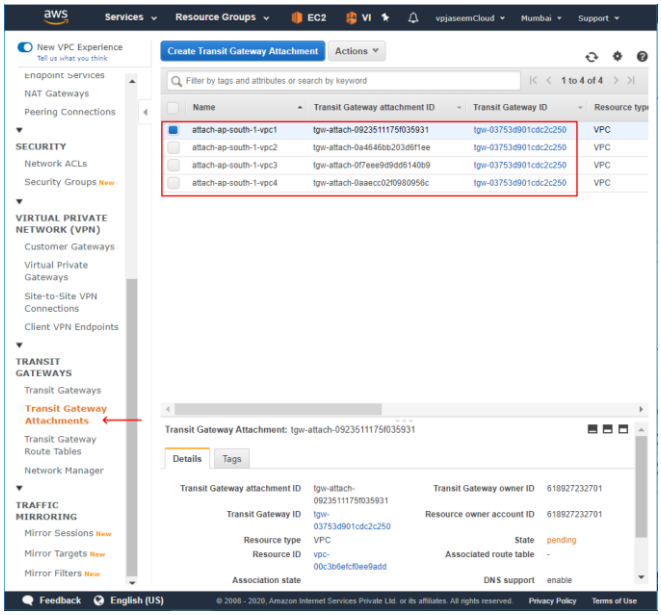


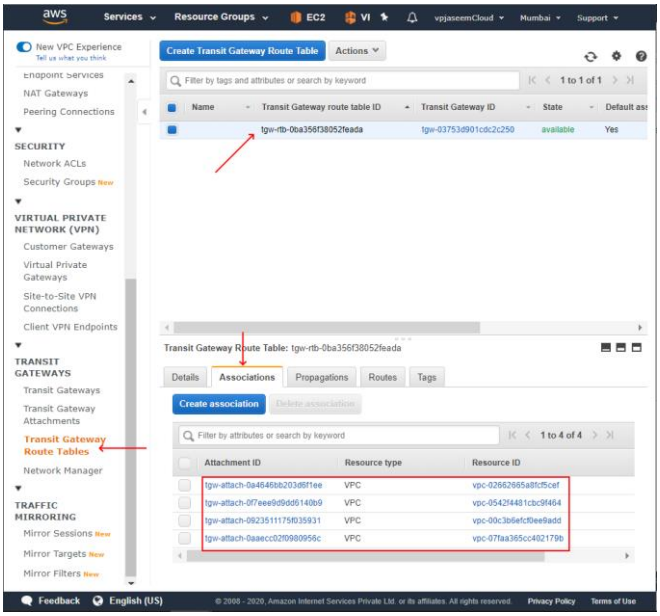



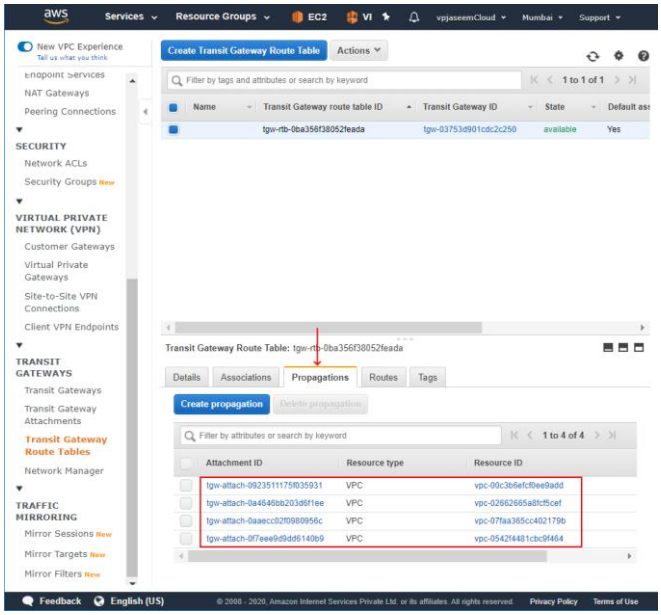
## Step 4: Attaching VPCs to the Transit Gateway

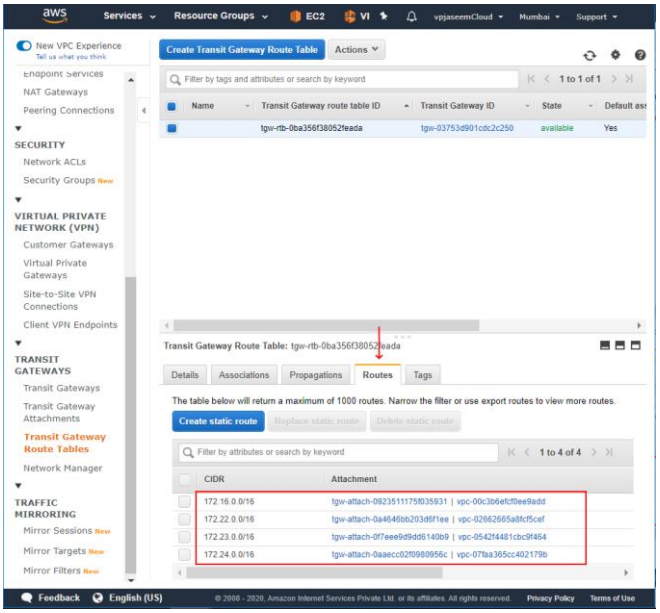


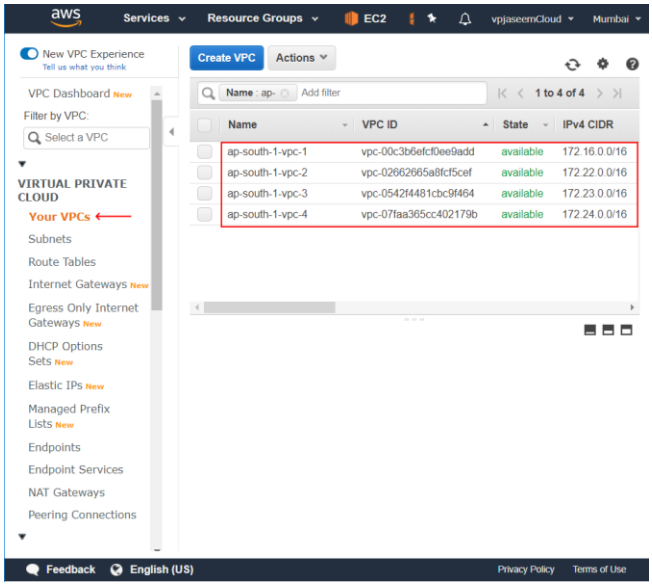



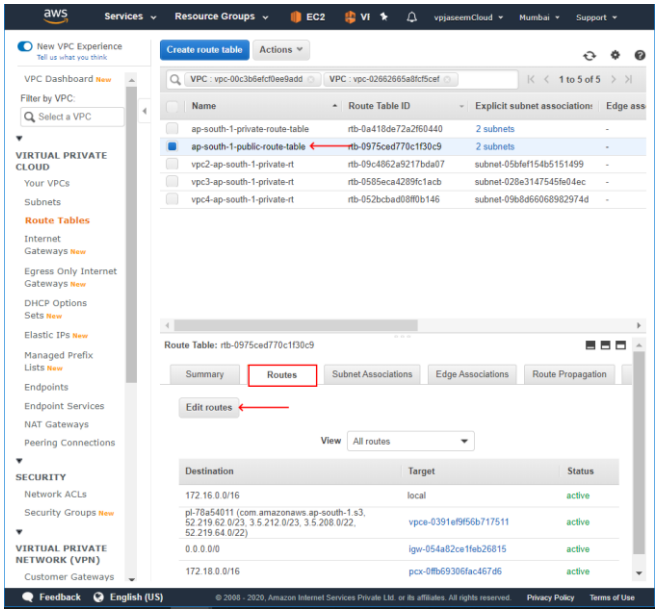



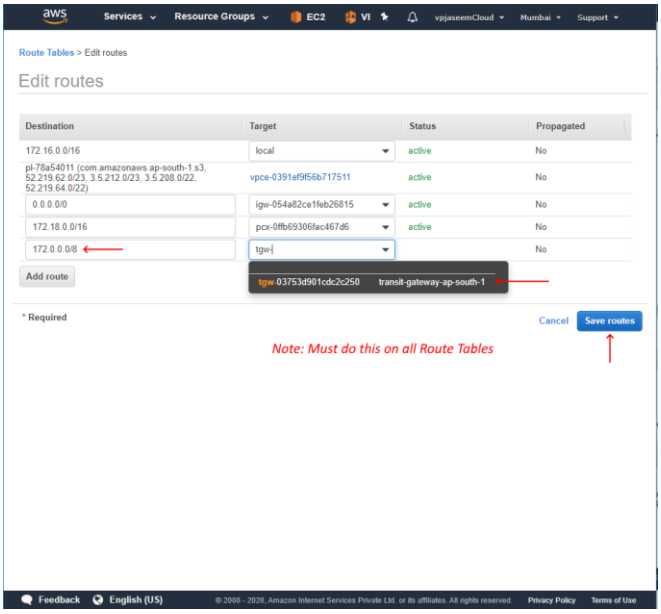



## Step 5: Update Each VPC Route Table with Transit Gateway

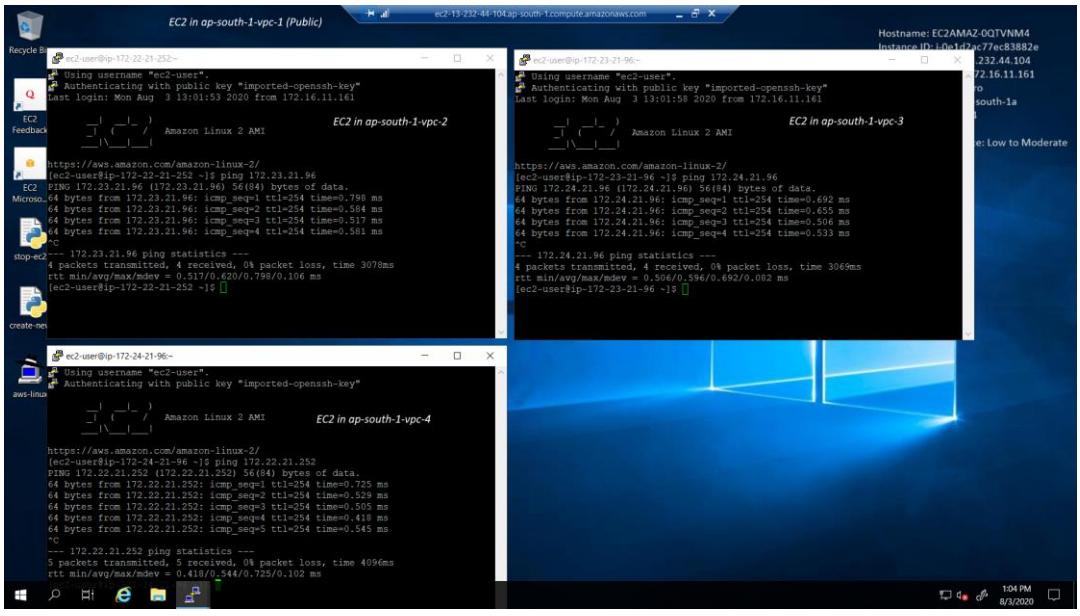



- My VPC CIDR blocks are 172.16.0.0/16, 172.22.0.0/16, 172.23.0.0/16, 172.24.0.0/16
- To match all these CIDR, let me select something like this, 172.0.0.0/8



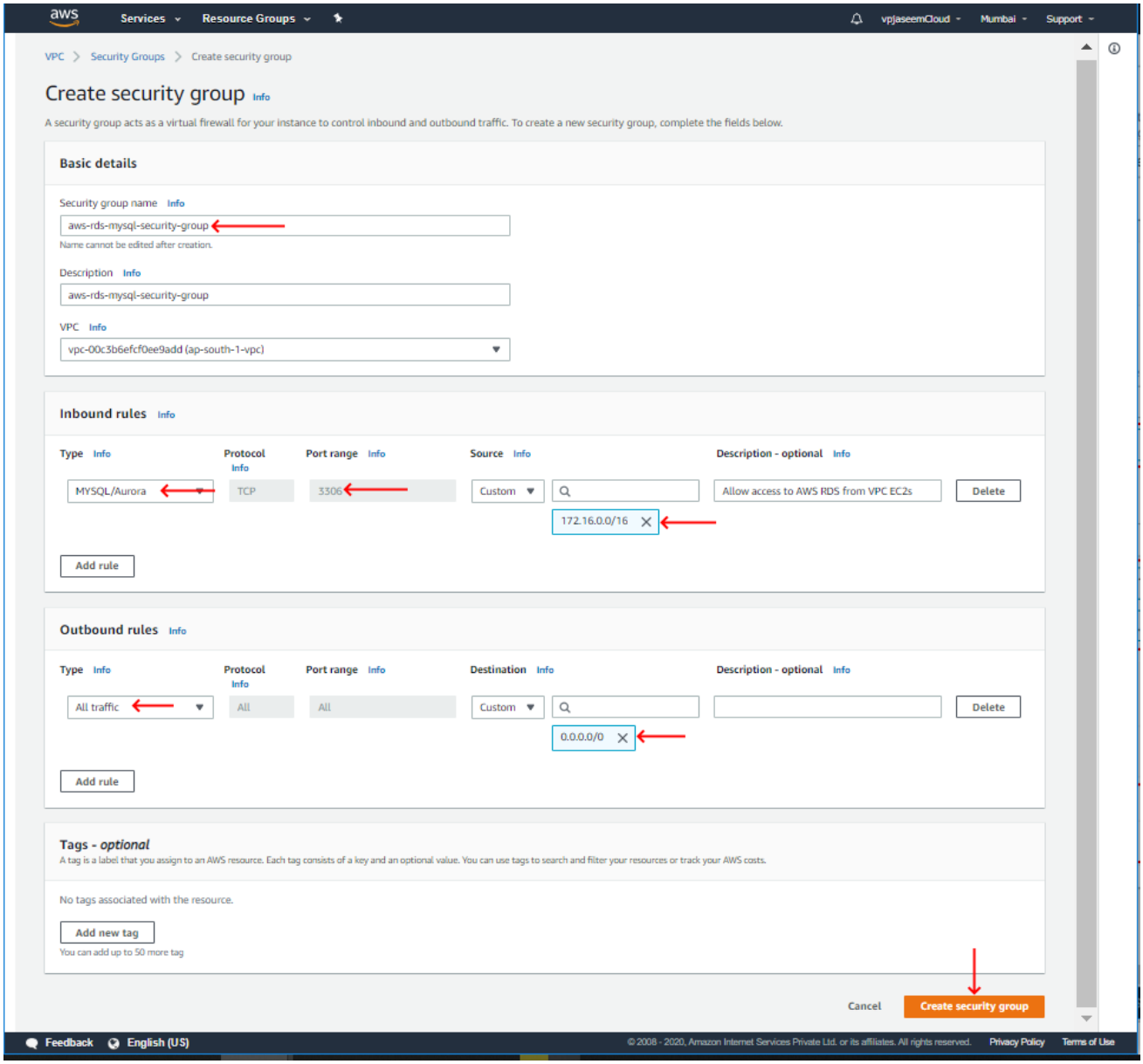

## Step 6: Login to Public Windows EC2 and Access other Linux EC2s

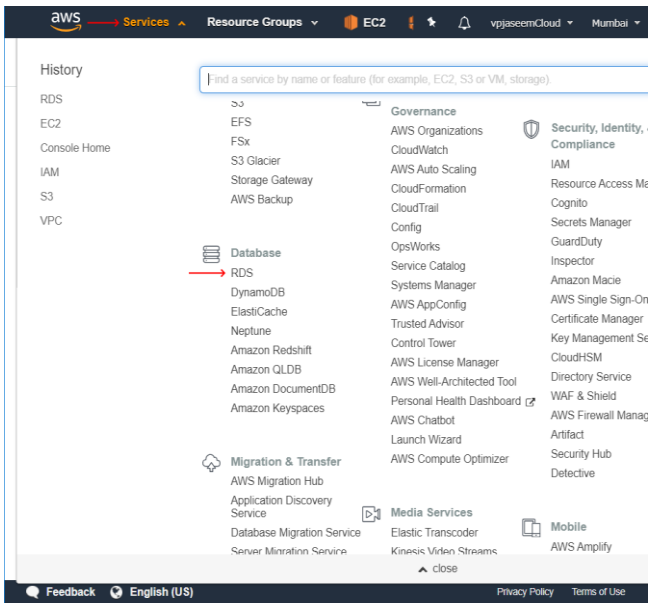


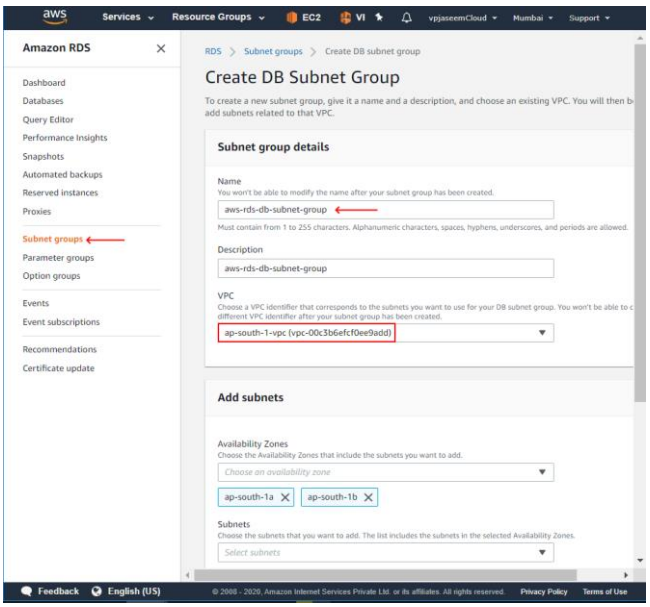
## AWS Relational Database Service

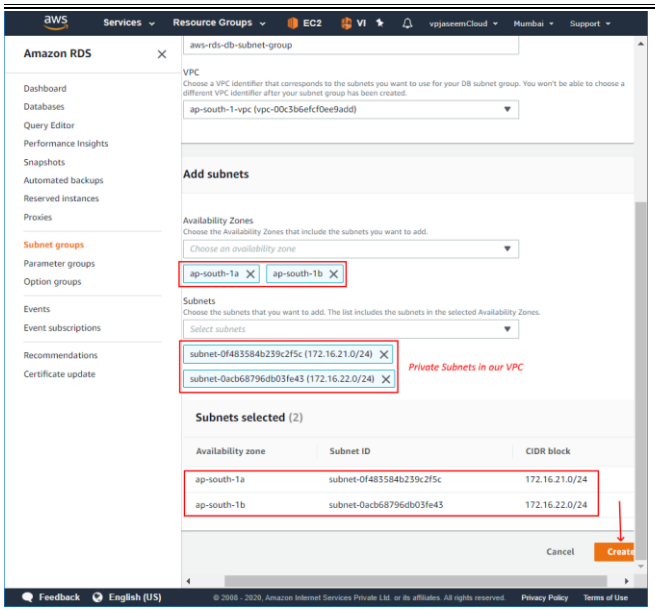
- RDS is the way to store structured, quarriable, indexable data inside the cloud
- Unstructured Data stored in EBS or S3 whereas Structured Data stored in databases (Redshift, RDS, Dynamo DB) SQL Style
- These services are called DBaaS (Database As A Service)
- We can choose any Database engine from PostgreSQL, MySQL, ORACLE, Microsoft SQL, MariaDB, Amazon Aurora)
- We can resize any time if required
- Auto backup is enabled by default, 0 to 35 days data backup and point in time recovery.
- Geo replicated
- Also have manual database snapshot to S3 feature available
- RDS Encryptions (Encryption at rest), uses AWS KMS (Key Management Store)
- Also, DB engine will have its own encryption.

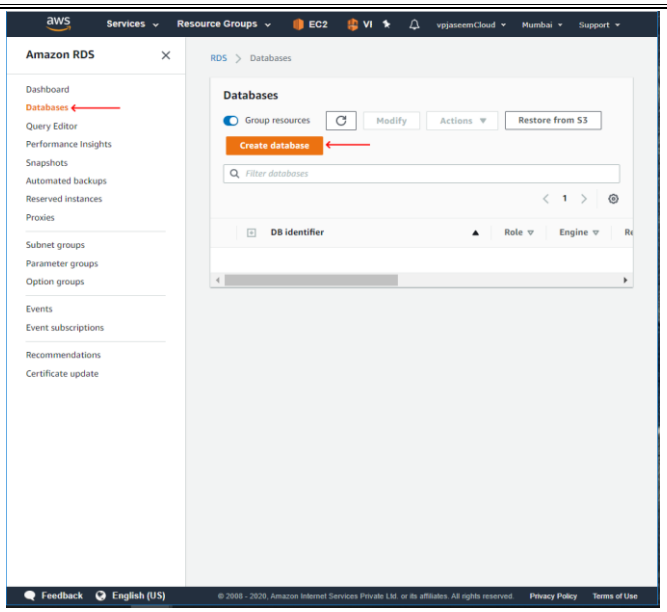
# [LAB] RDS Configuration

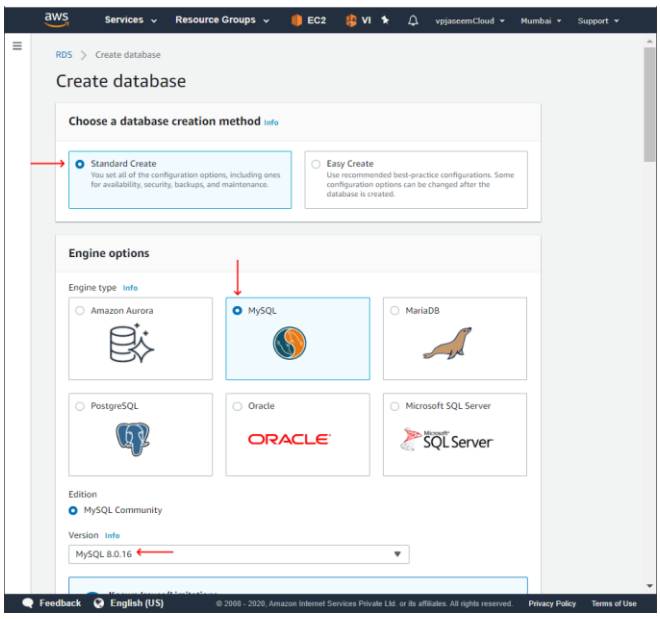


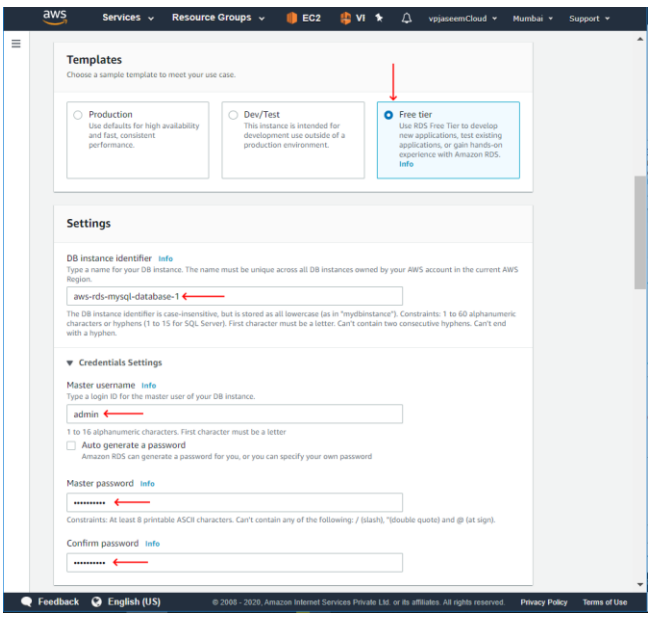



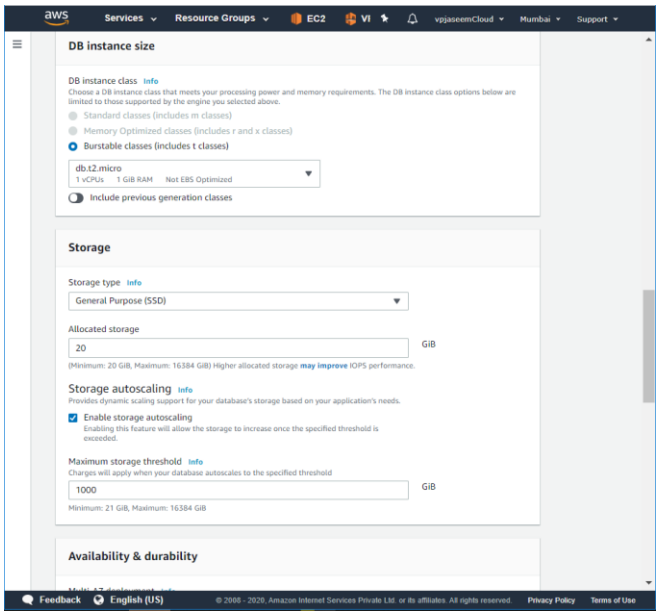



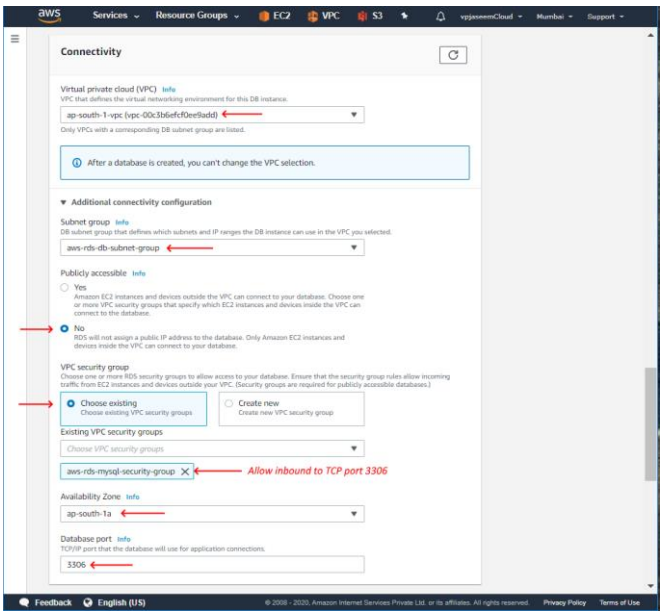

Amazon RDS
aws-rds-db-subnet-group
VPC
ap-south-1-vpc (vpc-00c3b6efcf0ee9add)
Add subnets
Availability Zones
ap-south-1a
ap-south-1b
Subnets
subnet-0f483584b239c2f5c (172.16.21.0/24)
subnet-0acb68796db03fe43 (172.16.22.0/24)
Private Subnets in our VPC
Subnets selected (2)
Availability zone
Subnet ID
CIDR block
ap-south-1a
subnet-0f483584b239c2f5c
172.16.21.0/24
ap-south-1b
subnet-0acb68796db03fe43
172.16.22.0/24
Cancel
Create

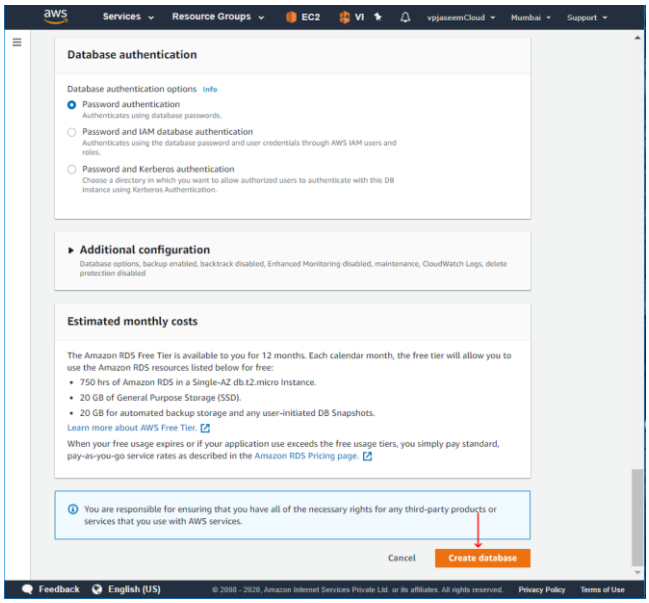

Amazon RDS
Databases
Group resources
Modify
Actions
Restore from S3
Create database
Filter databases
DB identifier
Role
Engine

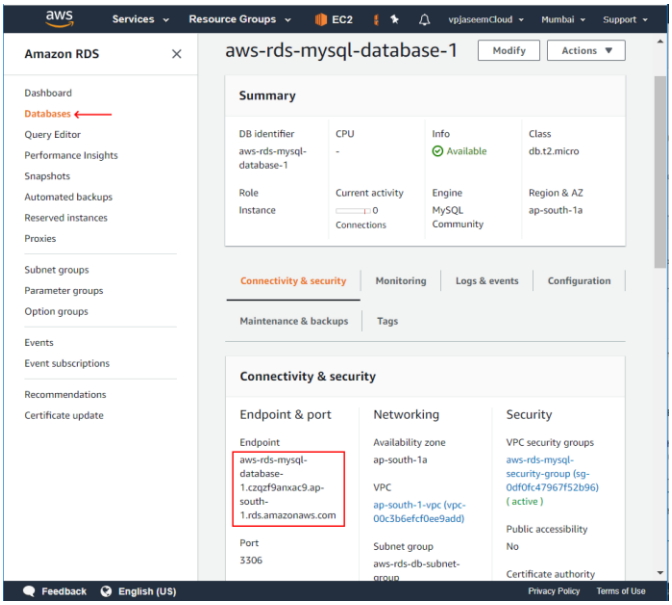

RDS > Create database
Create database
Choose a database creation method
Standard Create
Easy Create
Engine options
Engine type
Amazon Aurora
MySQL
MariaDB
PostgreSQL
Oracle
Microsoft SQL Server
Edition
MySQL Community
Version
MySQL 8.0.16

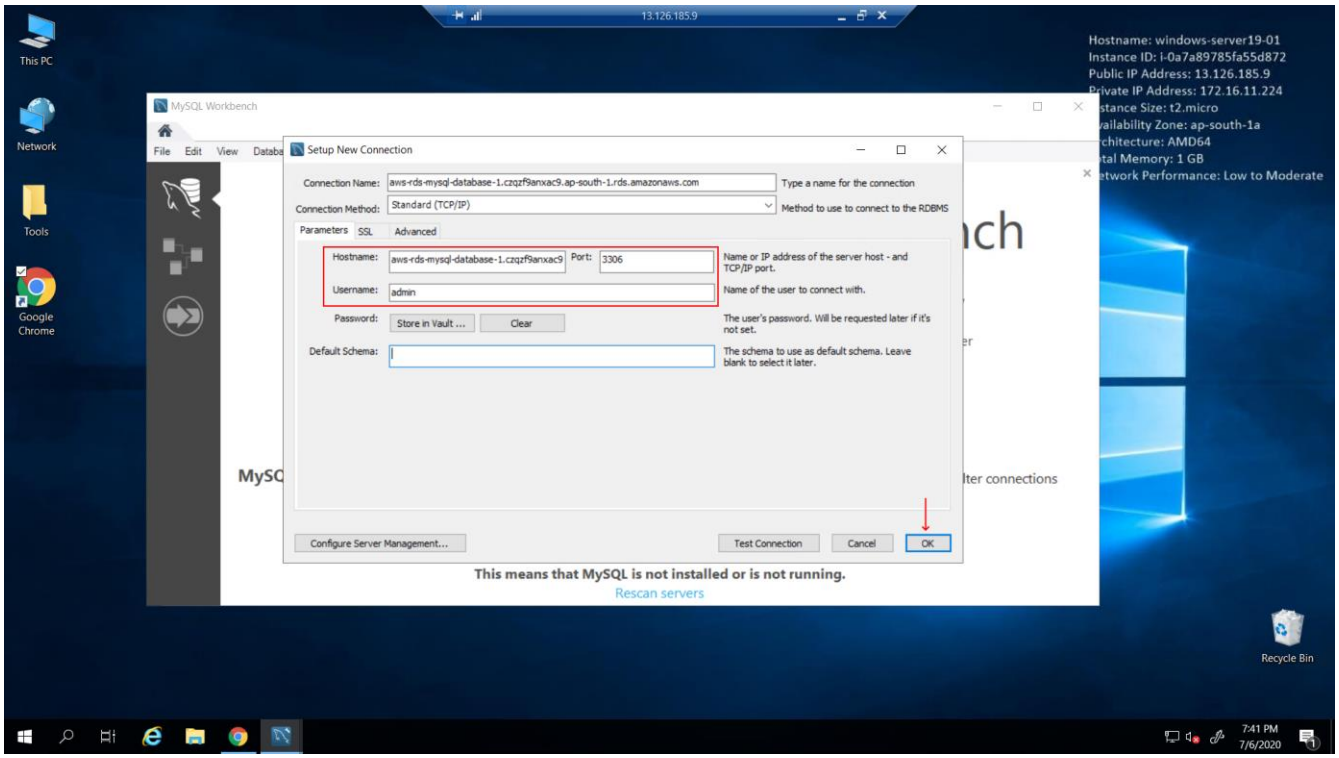

Templates
Production
Dev/Test
Free tier
Settings
DB instance identifier
aws-rds-mysql-database-1
Credentials Settings
Master username
admin
Auto generate a password
Master password
Confirm password

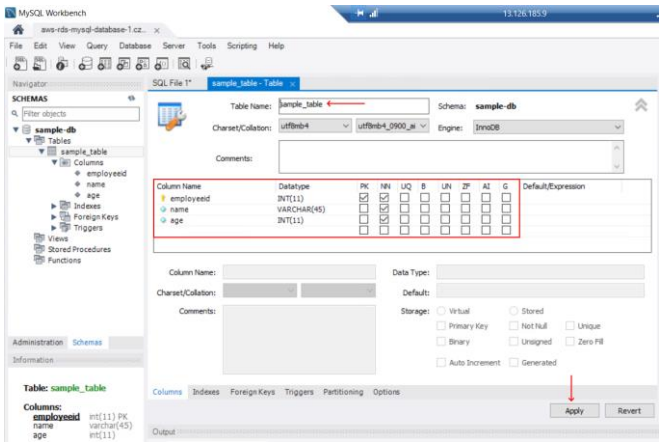

DB instance size
DB instance class
Burstable classes (includes t classes)
db.t2.micro
Storage
Storage type
General Purpose (SSD)
Allocated storage
20
GiB
Storage autoscaling
Enable storage autoscaling
Maximum storage threshold
1000
Availability & durability

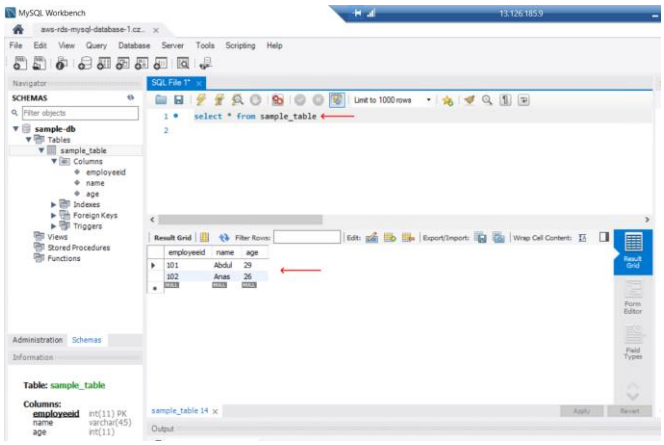

Connectivity
Virtual private cloud (VPC)
ap-south-1-vpc (vpc-00c3b6efcf0ee9add)
After a database is created, you can't change the VPC selection.
Additional connectivity configuration
Subnet group
aws-rds-db-subnet-group
Publicly accessible
No
VPC security group
Choose existing
Create new
Existing VPC security groups
aws-rds-mysql-security-group
Allow inbound to TCP port 3306
Availability Zone
ap-south-1a
Database port
3306


Database authentication
Database authentication options Info
Password authentication
Authenticates using database passwords.
Password and IAM database authentication
Authenticates using the database password and user credentials through AWS IAM users and roles.
Password and Kerberos authentication
Choose a directory in which you want to allow authorized users to authenticate with this DB instance using Kerberos Authentication.
Additional configuration
Database options, backup enabled, backtrack disabled, Enhanced Monitoring disabled, maintenance, CloudWatch Logs, delete protection disabled
Estimated monthly costs
The Amazon RDS Free Tier is available to you for 12 months. Each calendar month, the free tier will allow you to use the Amazon RDS resources listed below for free:
750 hrs of Amazon RDS in a Single-AZ db.t2.micro Instance.
20 GB of General Purpose Storage (SSD).
20 GB for automated backup storage and any user-initiated DB Snapshots.
Learn more about AWS Free Tier.
When your free usage expires or if your application use exceeds the free usage tiers, you simply pay standard, pay-as-you-go service rates as described in the Amazon RDS Pricing page.
You are responsible for ensuring that you have all of the necessary rights for any third-party products or services that you use with AWS services.
Cancel
Create database


Amazon RDS
Dashboard
Databases
Query Editor
Performance Insights
Snapshots
Automated backups
Reserved instances
Proxies
Subnet groups
Parameter groups
Option groups
Events
Event subscriptions
Recommendations
Certificate update
aws-rds-mysql-database-1
Modify
Actions
Summary
DB identifier
aws-rds-mysql-database-1
CPU
Info
Available
Class
db.t2.micro
Role
Instance
Current activity
0 Connections
Engine
MySQL Community
Region & AZ
ap-south-1a
Connectivity & security
Monitoring
Logs & events
Configuration
Maintenance & backups
Tags
Connectivity & security
Endpoint & port
Endpoint
aws-rds-mysql-database-1.czqzf9anxac9.ap-south-1.rds.amazonaws.com
Port
3306
Networking
Availability zone
ap-south-1a
VPC
ap-south-1-vpc (vpc-00c3b6efcf0ee9add)
Subnet group
aws-rds-db-subnet-group
Security
VPC security groups
aws-rds-mysql-security-group (sg-0df0fc47967f52b96) ( active )
Public accessibility
No
Certificate authority


13.126.185.9
This PC
Network
Tools
Google Chrome
Recycle Bin
Hostname: windows-server19-01
Instance ID: i-0a7a89785fa55d872
Public IP Address: 13.126.185.9
Private IP Address: 172.16.11.224
Instance Size: t2.micro
Availability Zone: ap-south-1a
Architecture: AMD64
Total Memory: 1 GB
Network Performance: Low to Moderate
MySQL Workbench
File Edit View Database
Setup New Connection
Connection Name: aws-rds-mysql-database-1.czqzf9anxac9.ap-south-1.rds.amazonaws.com
Type a name for the connection
Connection Method: Standard (TCP/IP)
Method to use to connect to the RDBMS
Parameters SSL Advanced
Hostname: aws-rds-mysql-database-1.czqzf9anxac9 Port: 3306
Name or IP address of the server host - and TCP/IP port.
Username: admin
Name of the user to connect with.
Password: Store in Vault ... Clear
The user's password. Will be requested later if it's not set.
Default Schema:
The schema to use as default schema. Leave blank to select it later.
Configure Server Management...
Test Connection
Cancel
OK
This means that MySQL is not installed or is not running.
Rescan servers
7:41 PM
7/6/2020


MySQL Workbench
SCHEMAS
sample-db
Tables
sample_table
Table Name: sample_table
Schema: sample-db
Charset/Collation:
Engine: InnoDB
Comments:
Column Name
Datatype
PK
NN
UQ
B
UN
ZF
AI
G
Default/Expression
employeeid INT(11)
name VARCHAR(45)
age INT(11)
Apply
Revert

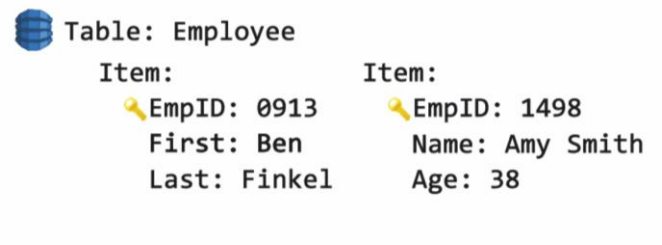

MySQL Workbench
SCHEMAS
sample-db
Tables
sample_table
SQL File 1*
select * from sample_table
employeeid name age
101 Abdul 29
102 Anas 26
Table: sample_table
Columns:
employeeid int(11) PK
name varchar(45)
age int(11)

## DynamoDB (AWS NoSQL)

- NoSQL database getting popular day by day.

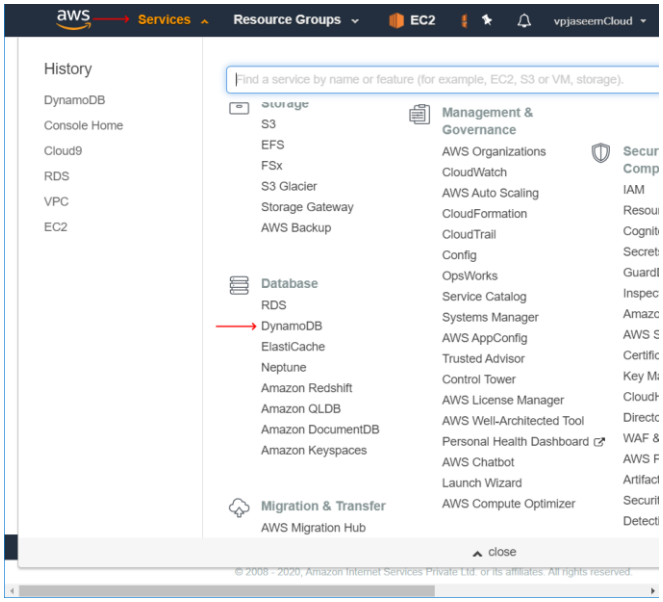
| | SQL | NoSQL |
|---|---|---|
| **Data Storage** | Rows and Columns | Key-Value |
| **Schemas** | Fixed | Dynamic |
| **Querying** | Using SQL | Focused on collection of documents |
| **Scalability** | Vertical (Add more CPU/ RMA) | Horizontal (Add more machines) |

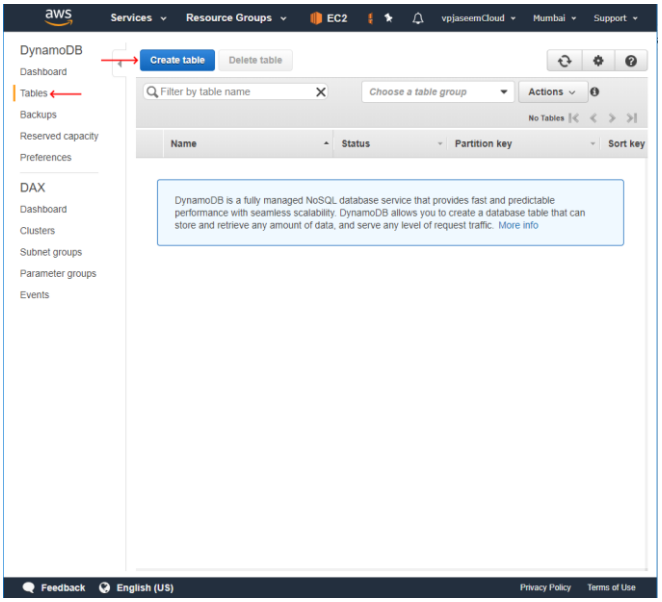
- Top level Organized into TABLES
- Table contain ITEMS
- ITEMS contain ATTRIBUTES (Key-value pair)
- Every TABLE requires a PRIMARY KEY
- Optionally include a SORT KEY and up to five LOCAL SECONDARY INDEX

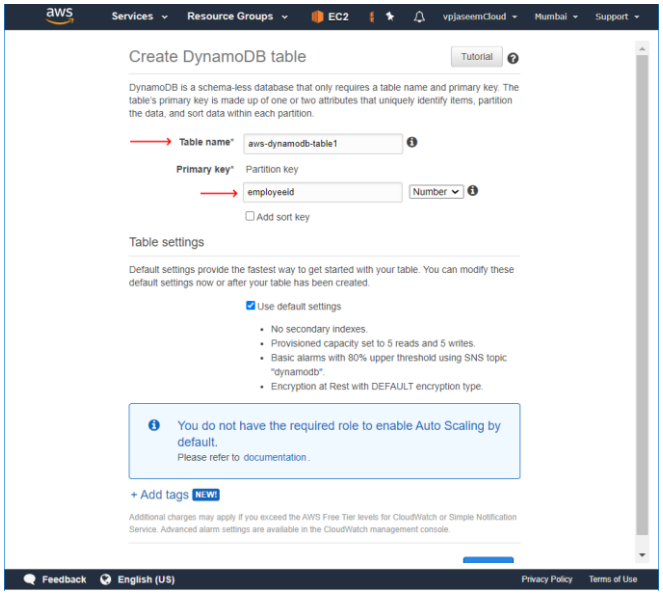


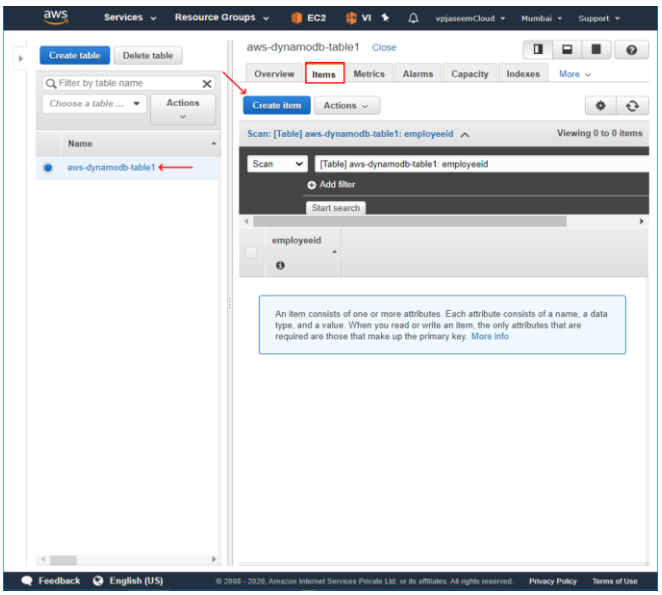

- Charged for Read and Write capacity throughput.
- Performing operations against AWS DynamoDB is either Management Console or REST API or Program Codes or SDKs.
- On demand auto scaling based on the application requirement

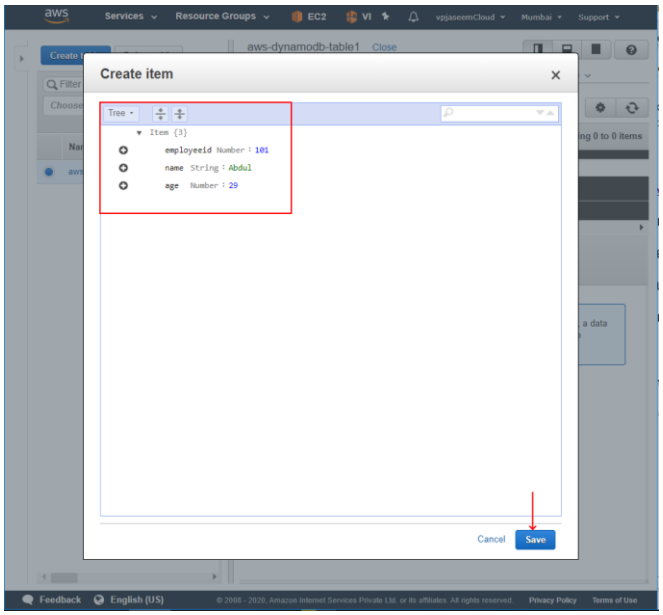
# [LAB] Dynamo DB Configurations

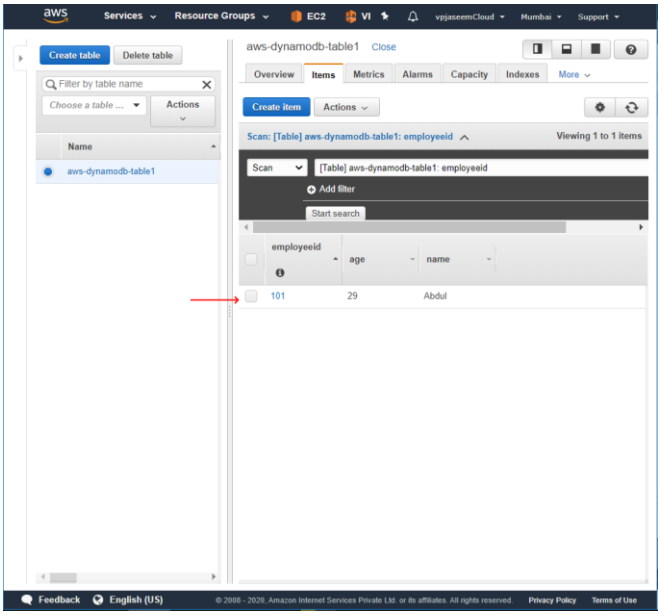


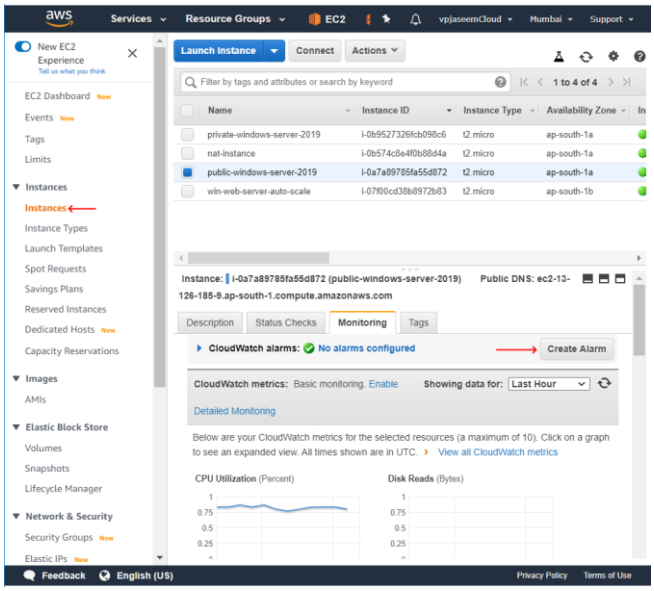



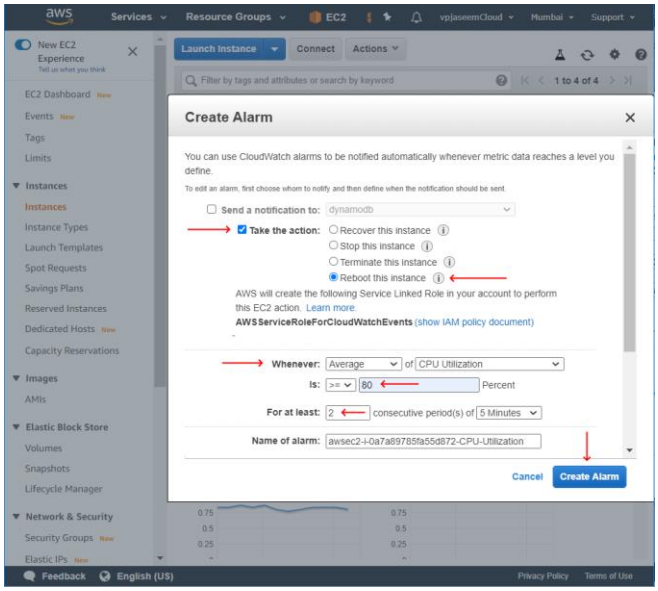



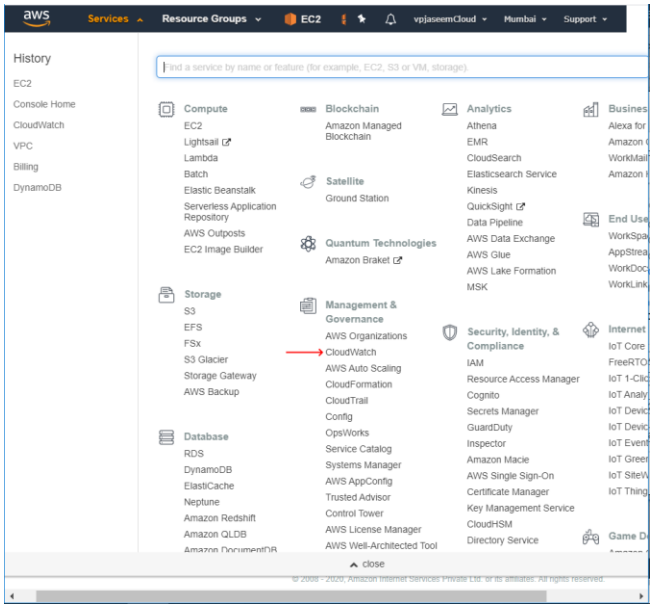
## Amazon CloudWatch

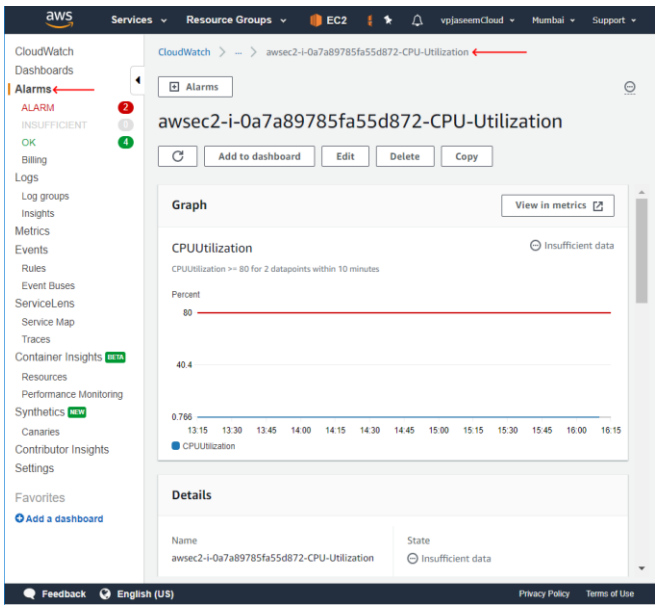
- CloudWatch collects monitoring and operational data in the form of logs, metrics, and events, providing you with a unified view of AWS resources, applications, and services that run on AWS and on-premises servers.
- By default, basic metrices are monitored every 5 minutes
- Detailed monitoring is chargeable, and the sampling period will be 1 minute.

# [LAB] Configure EC2 Alarm for CPU Utilization

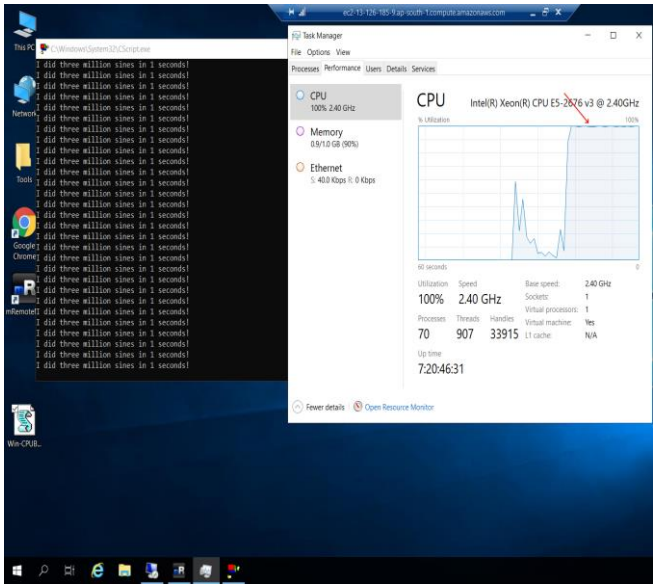


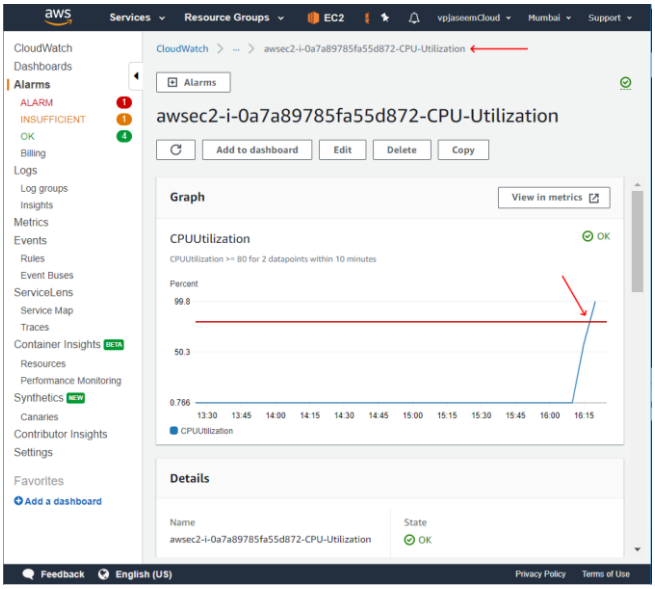


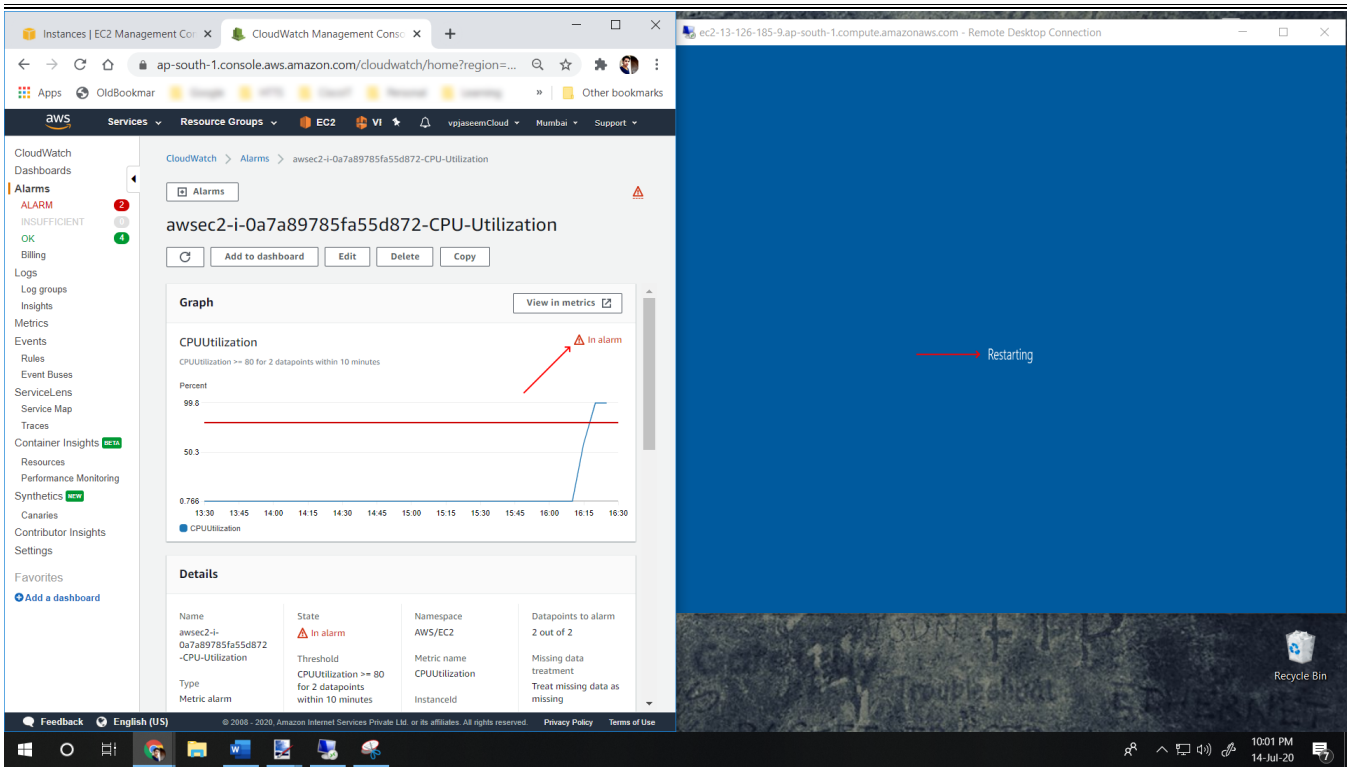



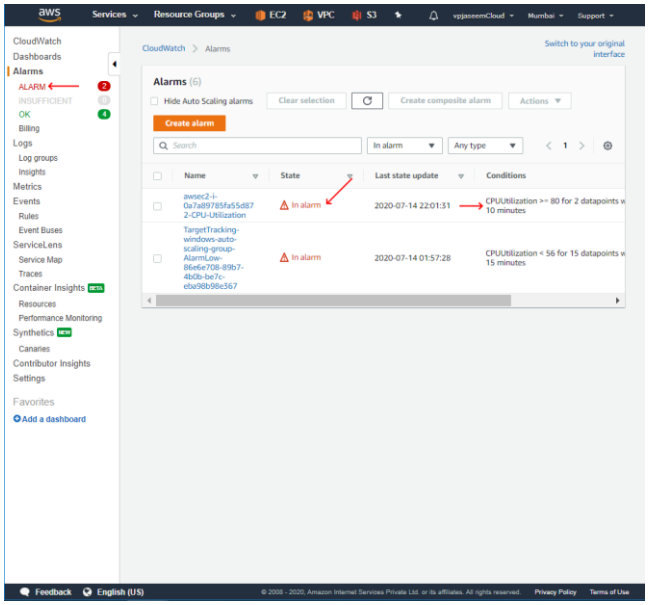



Download [CPU Busy Script], Right click and Run, do not double click.

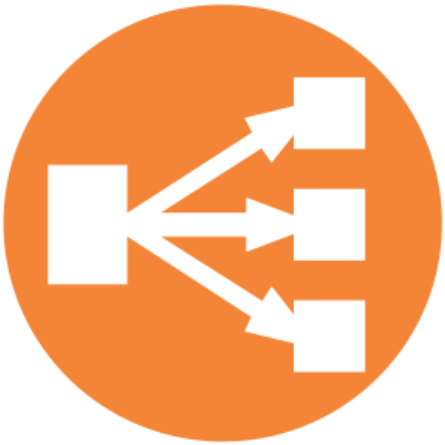





Instances | EC2 Management Con
CloudWatch Management Consc
ap-south-1.console.aws.amazon.com/cloudwatch/home?region=...
Apps
OldBookmar
Other bookmarks
Services
Resource Groups
EC2
VI
vpjaseemCloud
Mumbai
Support
CloudWatch
Dashboards
Alarms
ALARM
INSUFFICIENT
OK
Billing
Logs
Log groups
Insights
Metrics
Events
Rules
Event Buses
ServiceLens
Service Map
Traces
Container Insights
Resources
Performance Monitoring
Synthetics
Canaries
Contributor Insights
Settings
Favorites
Add a dashboard
CloudWatch > Alarms > awsec2-i-0a7a89785fa55d872-CPU-Utilization
Alarms
awsec2-i-0a7a89785fa55d872-CPU-Utilization
Add to dashboard
Edit
Delete
Copy
Graph
View in metrics
CPUUtilization
In alarm
CPUUtilization >= 80 for 2 datapoints within 10 minutes
Percent
99.8
50.3
0.766
13:30
13:45
14:00
14:15
14:30
14:45
15:00
15:15
15:30
15:45
16:00
16:15
16:30
CPUUtilization
Details
Name
awsec2-i-0a7a89785fa55d872-CPU-Utilization
State
In alarm
Threshold
CPUUtilization >= 80 for 2 datapoints within 10 minutes
Namespace
AWS/EC2
Metric name
CPUUtilization
InstanceId
Datapoints to alarm
2 out of 2
Missing data treatment
Treat missing data as missing
Type
Metric alarm
Feedback
English (US)
© 2008 - 2020, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use
ec2-13-126-185-9.ap-south-1.compute.amazonaws.com - Remote Desktop Connection
Restarting
Recycle Bin
10:01 PM
14-Jul-20


Services
Resource Groups
EC2
VPC
S3
vpjaseemCloud
Mumbai
Support
CloudWatch
Dashboards
Alarms
ALARM
INSUFFICIENT
OK
Billing
Logs
Log groups
Insights
Metrics
Events
Rules
Event Buses
ServiceLens
Service Map
Traces
Container Insights
Resources
Performance Monitoring
Synthetics
Canaries
Contributor Insights
Settings
Favorites
Add a dashboard
CloudWatch > Alarms
Switch to your original interface
Alarms (6)
Hide Auto Scaling alarms
Clear selection
Create composite alarm
Actions
Create alarm
Search
In alarm
Any type
Name
State
Last state update
Conditions
awsec2-i-0a7a89785fa55d872-CPU-Utilization
In alarm
2020-07-14 22:01:31
CPUUtilization >= 80 for 2 datapoints w 10 minutes
TargetTracking-windows-auto-scaling-group-AlarmLow-86e6e708-89b7-4b0b-be7c-eba98b98e367
In alarm
2020-07-14 01:57:28
CPUUtilization < 56 for 15 datapoints w 15 minutes
Feedback
English (US)
© 2008 - 2020, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use

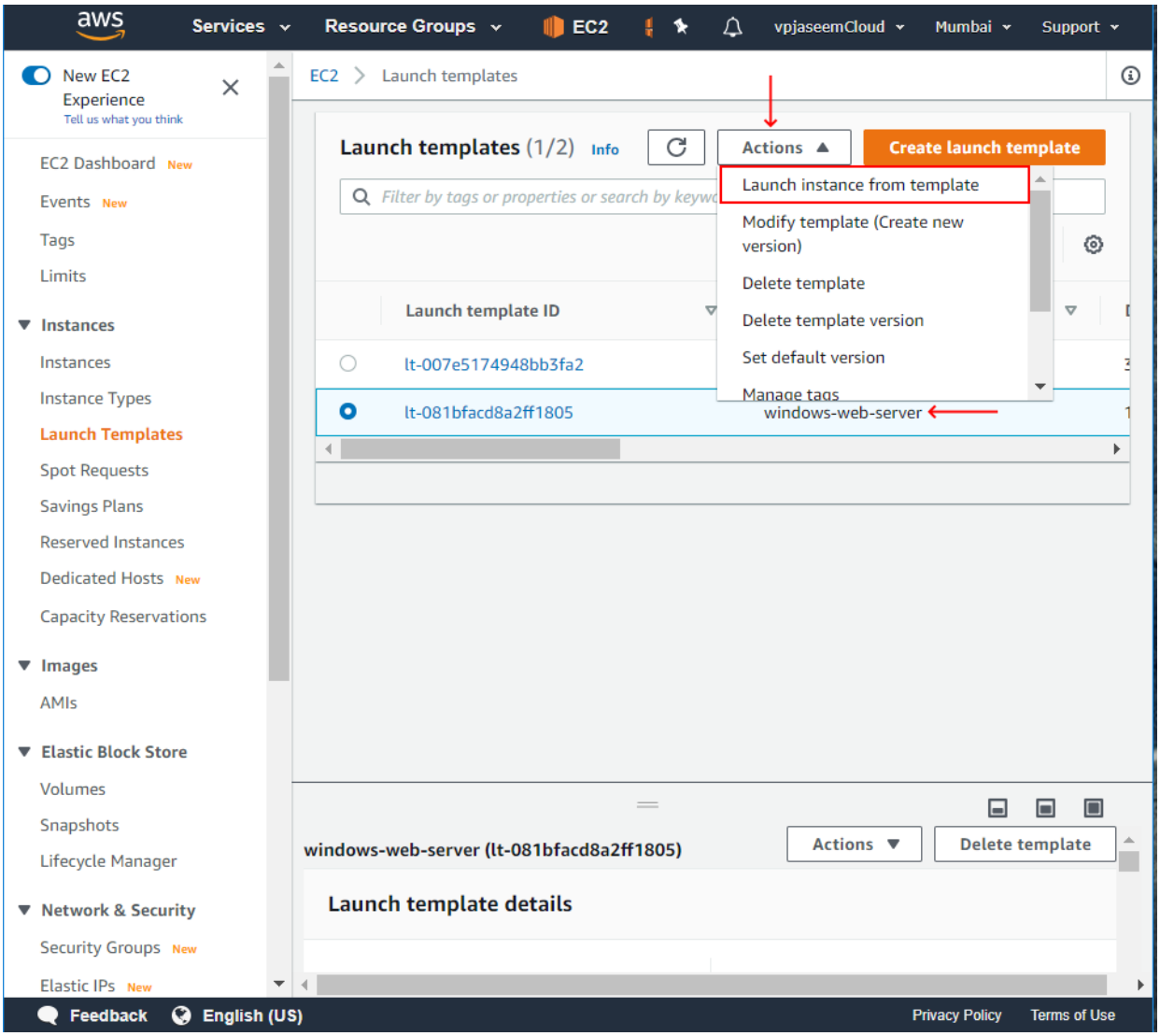
## Elastic Load Balances (ELB)

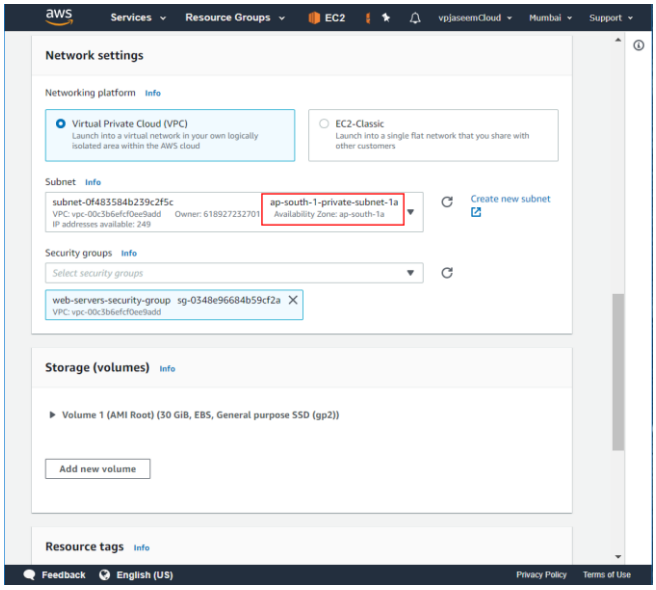


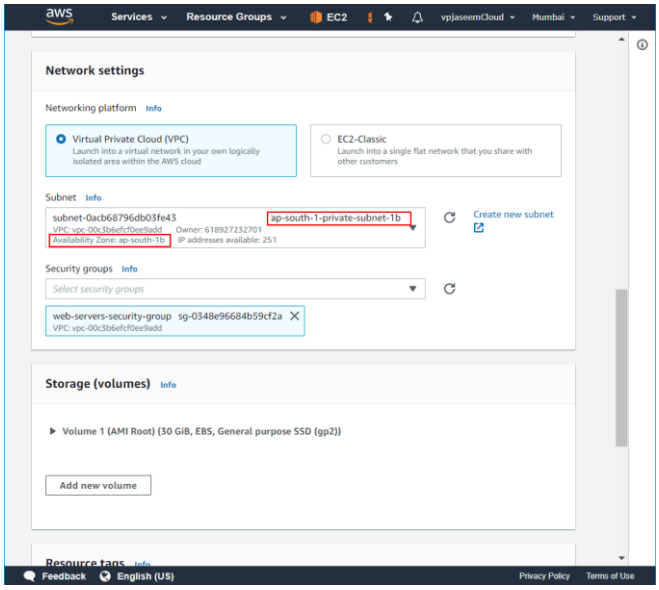
- ELB can be Public or Private (depends on the subnets you choose)
- It is managed by AWS (we don't need to worry about scalability or availability)
- Used to distribute incoming traffic to multiple of instances those are registered to it and does the health checks of instances.
- If an instance is unhealthy, ELB stops sending traffic to that.
- We can move our Web Servers to Private subnet, only keep ELB in public and send the traffic to private instance.
- Traffic is distributed between 2 AZs in round-robin fashion
- Traffic is distributed within AZs based on the instance having least number of connections.
- Recommended to access ELB via DNS name only not based on IP

## [LAB] Elastic Load Balances

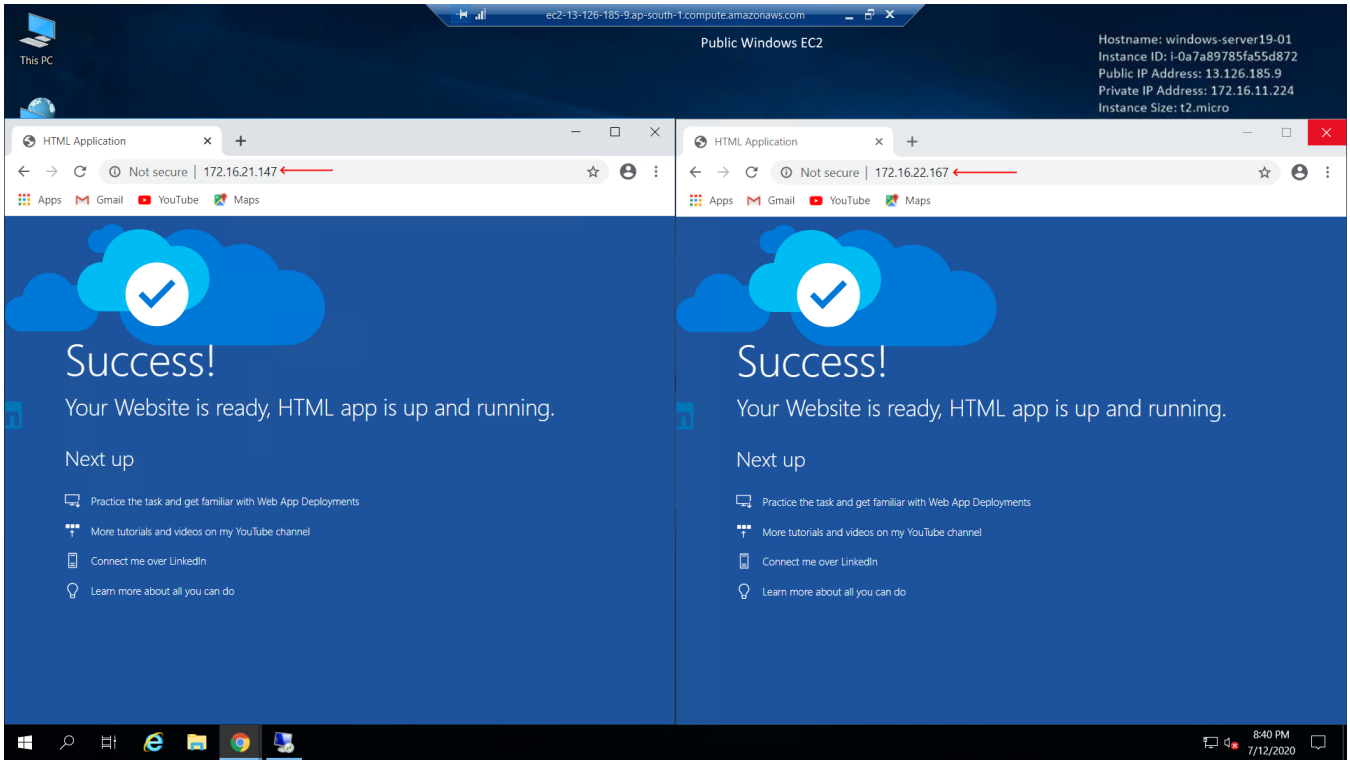
- Deploy 2 web servers (either Windows or Linux) in private subnet. Use the launch templates with User Data to do so.
- In this example, I have deployed 2 Windows IIS Web Servers running same copy of website files.



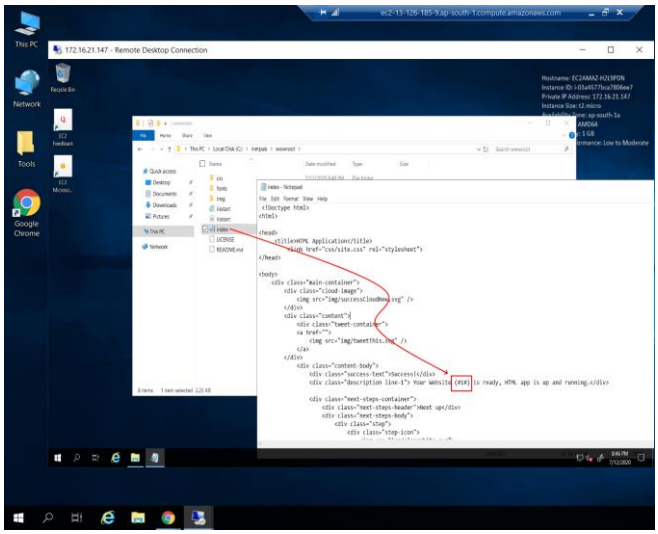



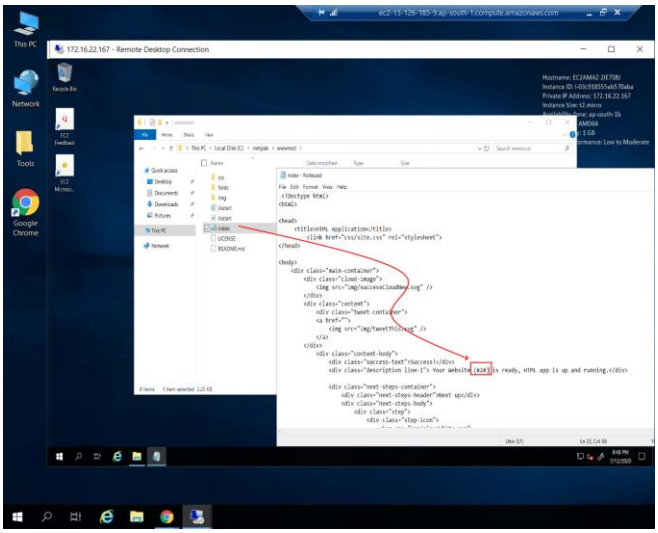



- These 2 websites are not accessible from internet since they are deployed in Private Subnet. Use Public Windows Server (Jump Host) to access these two servers with local IP.

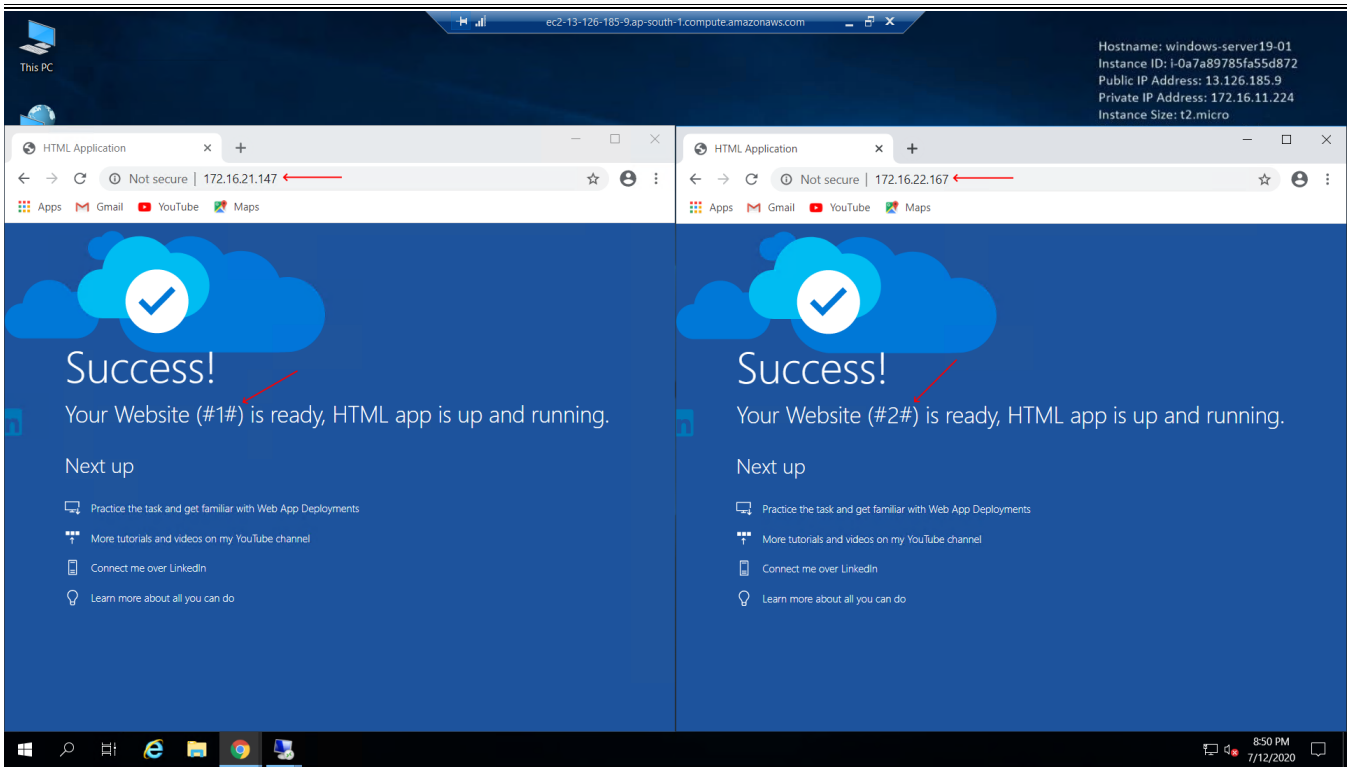



- Let's modify these sites a bit so that we can identify which is which. Go to the C:\inetpub\wwwroot and add some characters in the index.html file.

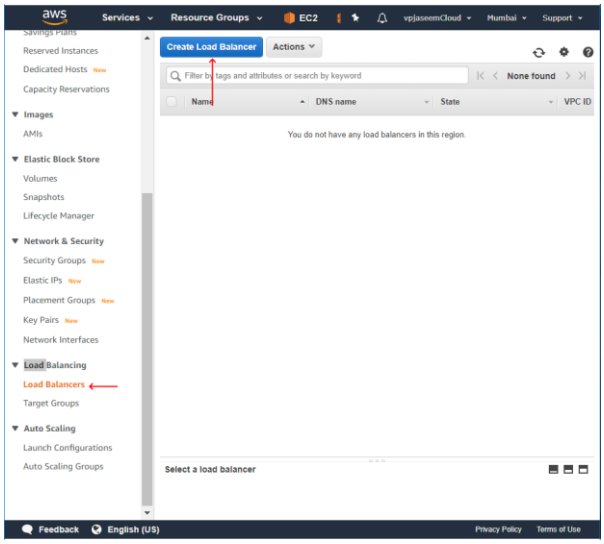


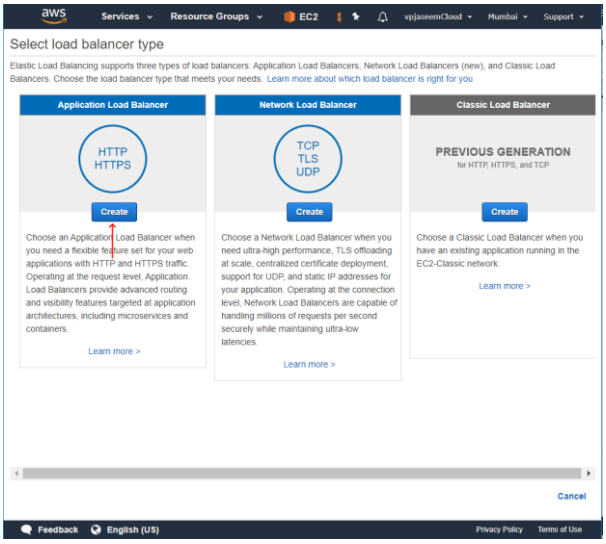


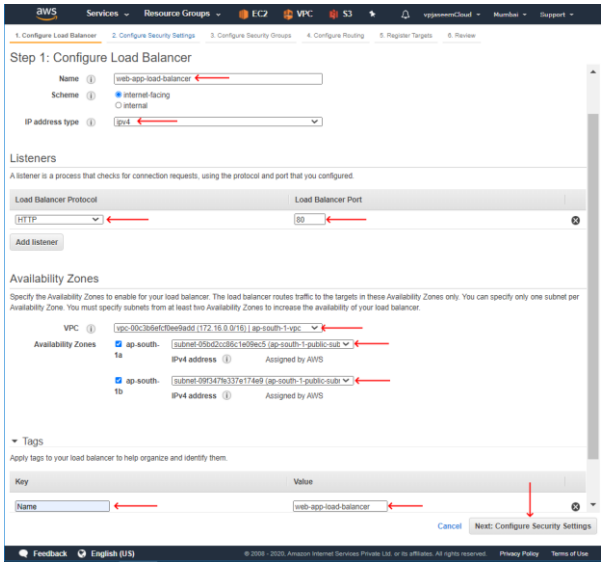
- Now we have a small difference in the Websites, this is just for our understandings.

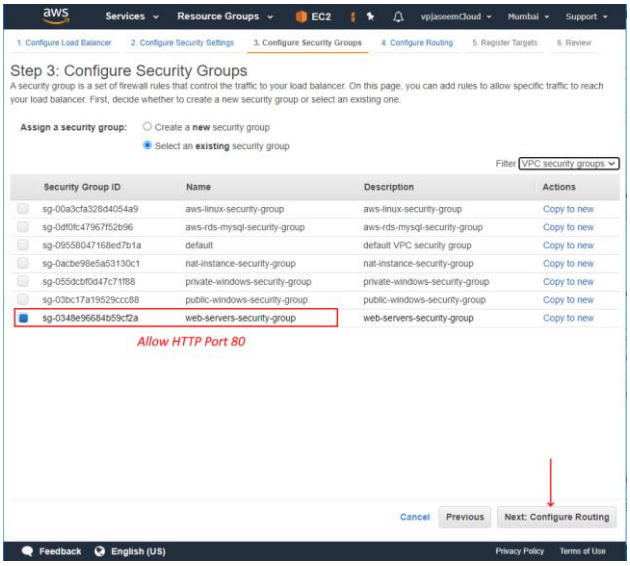


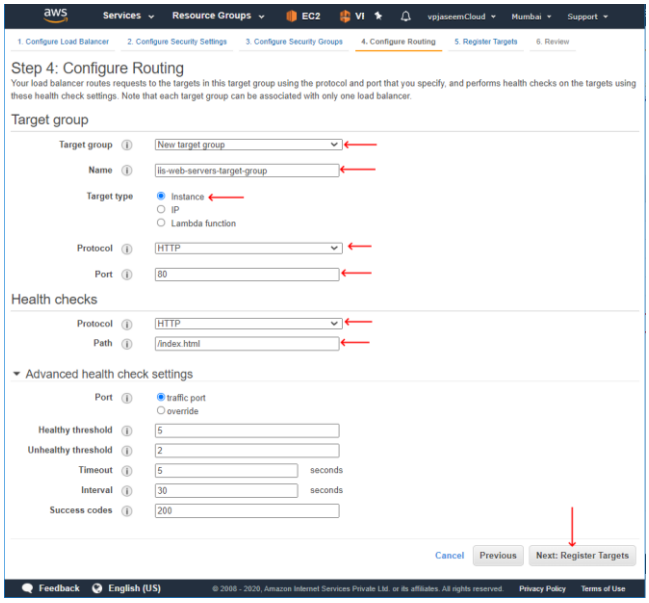

- Now we are going to deploy ELB so that it can distribute traffic between these two.

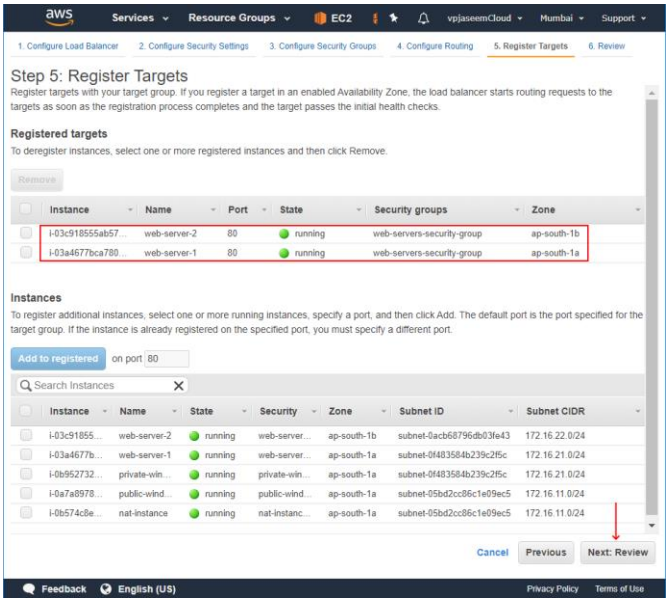


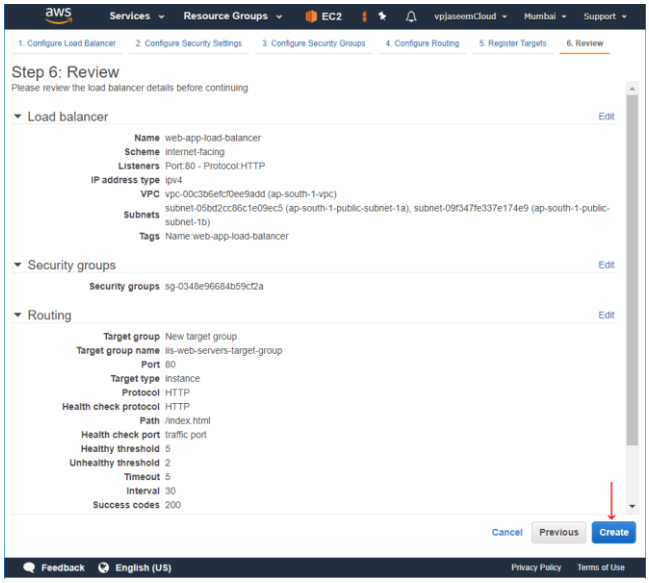



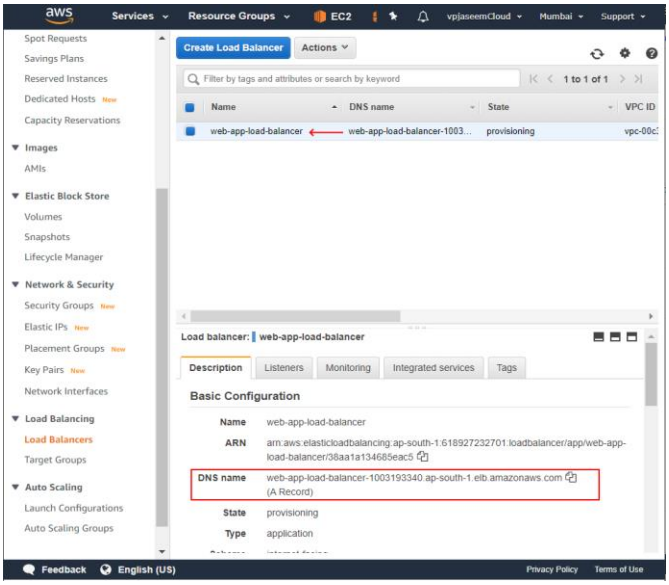



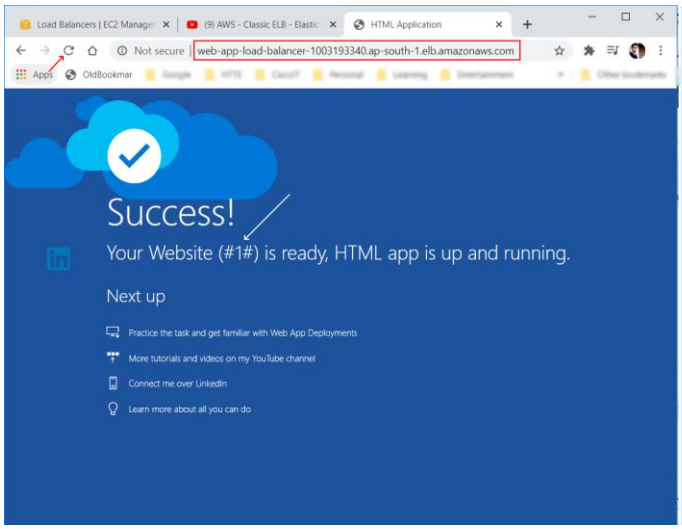



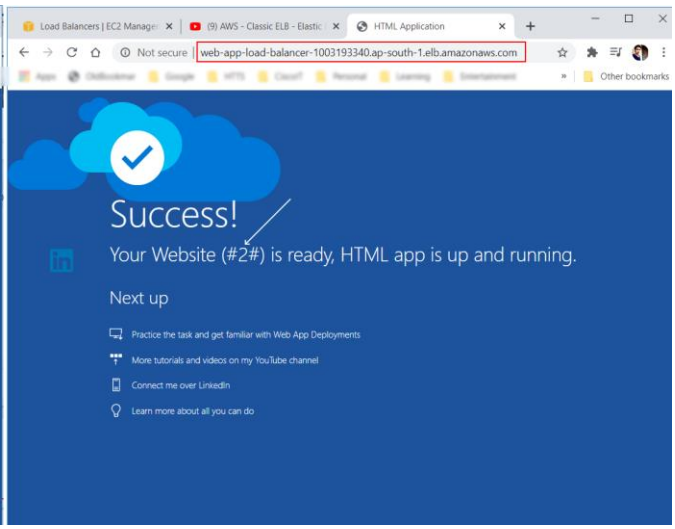

Step 4: Configure Routing
Your load balancer routes requests to the targets in this target group using the protocol and port that you specify, and performs health checks on the targets using these health check settings. Note that each target group can be associated with only one load balancer.
Target group
Target group: New target group
Name: iis-web-servers-target-group
Target type: Instance / IP / Lambda function
Protocol: HTTP
Port: 80
Health checks
Protocol: HTTP
Path: /index.html
Advanced health check settings
Port: traffic port / override
Healthy threshold: 5
Unhealthy threshold: 2
Timeout: 5 seconds
Interval: 30 seconds
Success codes: 200
Cancel
Previous
Next: Register Targets

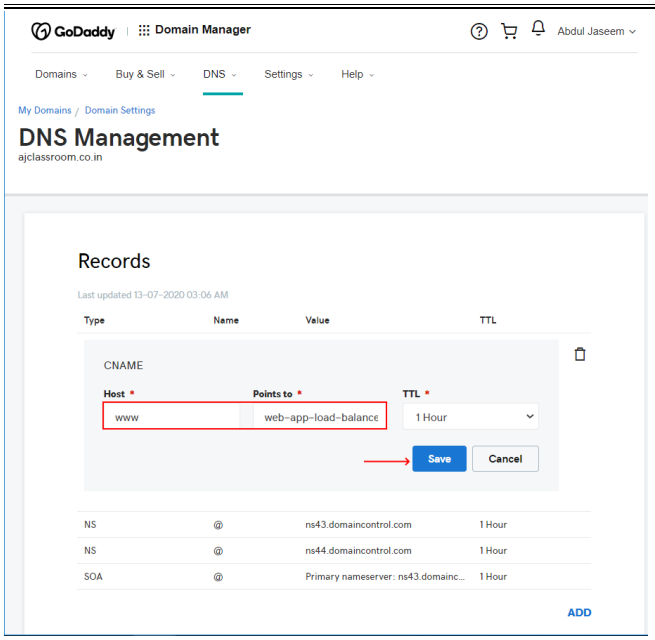

Step 5: Register Targets
Register targets with your target group. If you register a target in an enabled Availability Zone, the load balancer starts routing requests to the targets as soon as the registration process completes and the target passes the initial health checks.
Registered targets
To deregister instances, select one or more registered instances and then click Remove.
Remove
Instance | Name | Port | State | Security groups | Zone
i-03c918555ab57... | web-server-2 | 80 | running | web-servers-security-group | ap-south-1b
i-03a4677bca780... | web-server-1 | 80 | running | web-servers-security-group | ap-south-1a
Instances
To register additional instances, select one or more running instances, specify a port, and then click Add. The default port is the port specified for the target group. If the instance is already registered on the specified port, you must specify a different port.
Add to registered
on port 80
Search Instances
Instance | Name | State | Security | Zone | Subnet ID | Subnet CIDR
i-03c91855... | web-server-2 | running | web-server... | ap-south-1b | subnet-0acb68796db03fe43 | 172.16.22.0/24
i-03a4677b... | web-server-1 | running | web-server... | ap-south-1a | subnet-0f483584b239c2f5c | 172.16.21.0/24
i-0b952732... | private-win... | running | private-win... | ap-south-1a | subnet-0f483584b239c2f5c | 172.16.21.0/24
i-0a7a8978... | public-wind... | running | public-wind... | ap-south-1a | subnet-05bd2cc86c1e09ec5 | 172.16.11.0/24
i-0b574c8e... | nat-instance | running | nat-instanc... | ap-south-1a | subnet-05bd2cc86c1e09ec5 | 172.16.11.0/24
Cancel
Previous
Next: Review

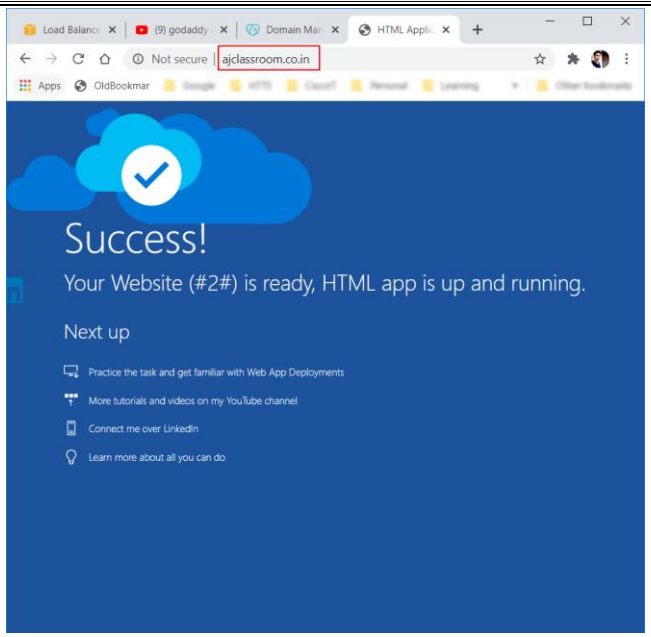

Step 6: Review
Please review the load balancer details before continuing
Load balancer
Name: web-app-load-balancer
Scheme: internet-facing
Listeners: Port:80 - Protocol:HTTP
IP address type: ipv4
VPC: vpc-00c3b6efcf0ee9add (ap-south-1-vpc)
Subnets: subnet-05bd2cc86c1e09ec5 (ap-south-1-public-subnet-1a), subnet-09f347fe337e174e9 (ap-south-1-public-subnet-1b)
Tags: Name:web-app-load-balancer
Security groups
Security groups: sg-0348e96684b59cf2a
Routing
Target group: New target group
Target group name: iis-web-servers-target-group
Port: 80
Target type: instance
Protocol: HTTP
Health check protocol: HTTP
Path: /index.html
Health check port: traffic port
Healthy threshold: 5
Unhealthy threshold: 2
Timeout: 5
Interval: 30
Success codes: 200
Cancel
Previous
Create


Create Load Balancer
Actions
Name | DNS name | State | VPC ID
web-app-load-balancer | web-app-load-balancer-1003... | provisioning
Load balancer: web-app-load-balancer
Description | Listeners | Monitoring | Integrated services | Tags
Basic Configuration
Name: web-app-load-balancer
ARN: arn:aws:elasticloadbalancing:ap-south-1:618927232701:loadbalancer/app/web-app-load-balancer/38aa1a134685eac5
DNS name: web-app-load-balancer-1003193340.ap-south-1.elb.amazonaws.com (A Record)
State: provisioning
Type: application

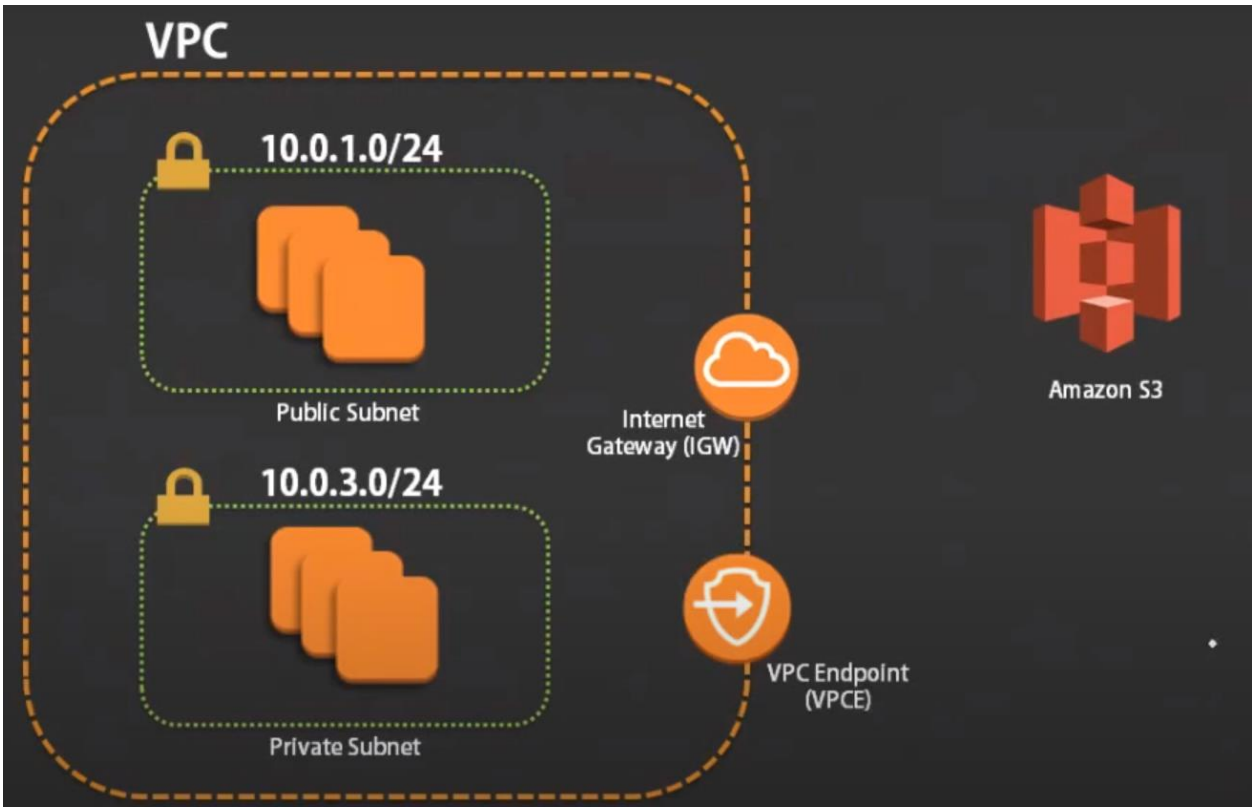

web-app-load-balancer-1003193340.ap-south-1.elb.amazonaws.com
Success!
Your Website (#1#) is ready, HTML app is up and running.
Next up
Practice the task and get familiar with Web App Deployments
More tutorials and videos on my YouTube channel
Connect me over LinkedIn
Learn more about all you can do


web-app-load-balancer-1003193340.ap-south-1.elb.amazonaws.com
Success!
Your Website (#2#) is ready, HTML app is up and running.
Next up
Practice the task and get familiar with Web App Deployments
More tutorials and videos on my YouTube channel
Connect me over LinkedIn
Learn more about all you can do


GoDaddy
Domain Manager
Abdul Jaseem
Domains
Buy & Sell
DNS
Settings
Help
My Domains / Domain Settings
DNS Management
ajclassroom.co.in
Records
Last updated 13-07-2020 03:06 AM
Type
Name
Value
TTL
CNAME
Host *
Points to *
TTL *
www
web-app-load-balance
1 Hour
Save
Cancel
NS
@
ns43.domaincontrol.com
1 Hour
NS
@
ns44.domaincontrol.com
1 Hour
SOA
@
Primary nameserver: ns43.domainc...
1 Hour
ADD

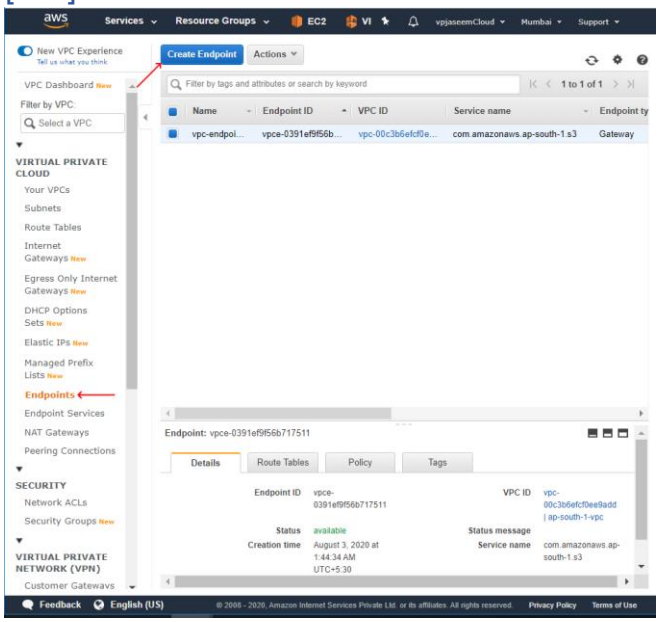

Not secure
ajclassroom.co.in
Apps
OldBookmar
Success!
Your Website (#2#) is ready, HTML app is up and running.
Next up
Practice the task and get familiar with Web App Deployments
More tutorials and videos on my YouTube channel
Connect me over LinkedIn
Learn more about all you can do

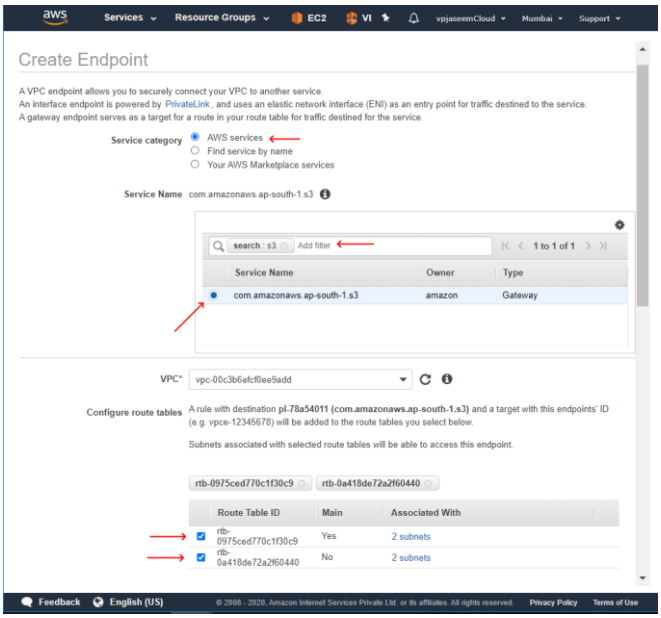
## VPC Endpoints

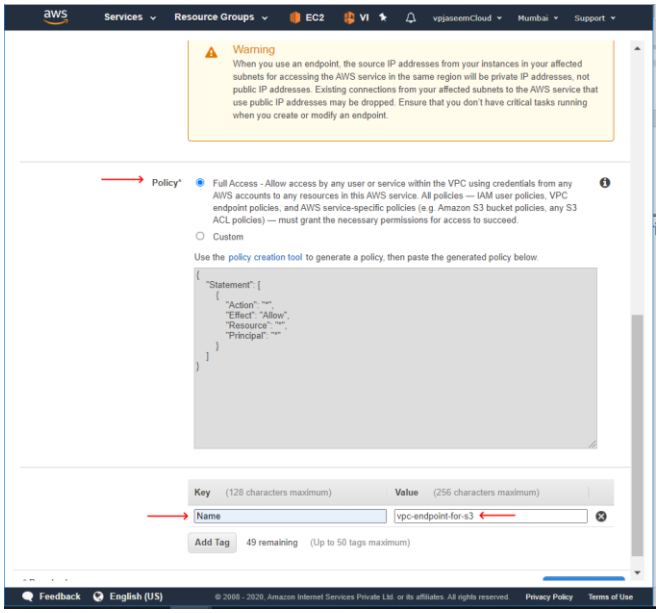


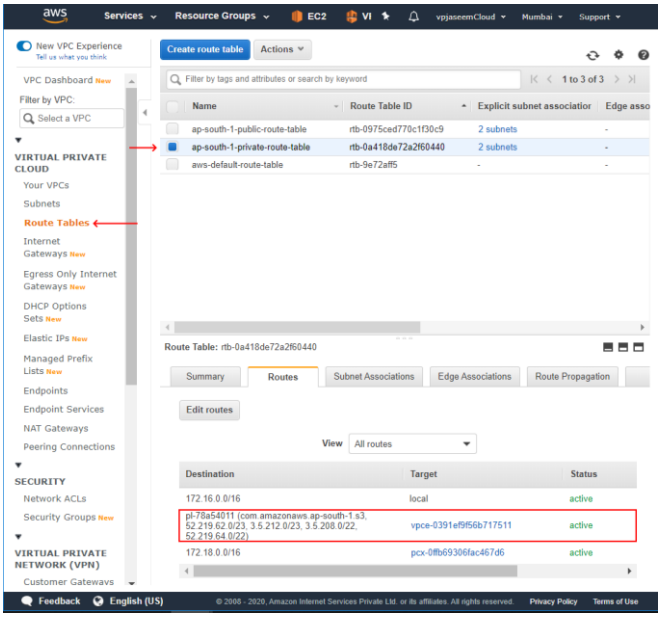

- AWS services like S3, DynamoDB, etc. are connected via Internet Gateway by default.
- Traffic has to go via IGW and come back although the services are still on AWS
- VPC Endpoint is like an IGW but it interconnects connects AWS Services via AWS internal network rather that traversing over internet.
- Originating and Destination resources should be in same region.

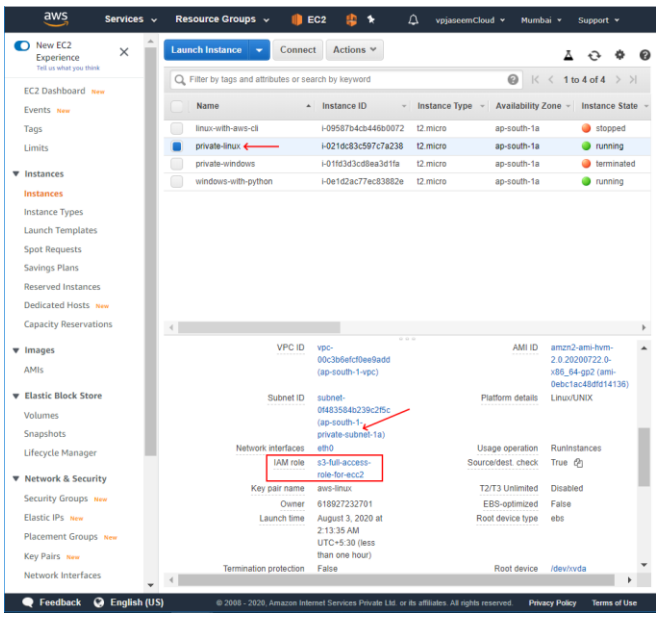
# [LAB] Get Access to S3 from Private Subnet EC2 Instance

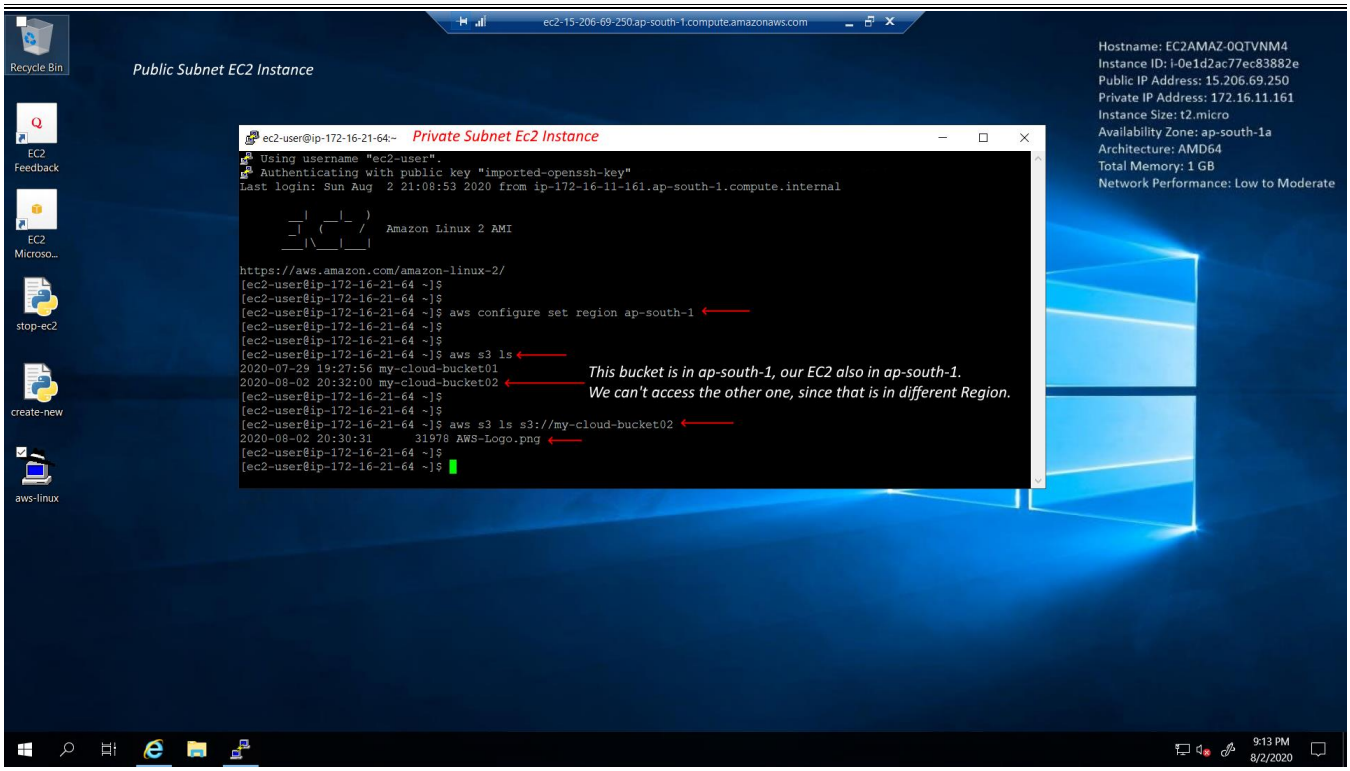


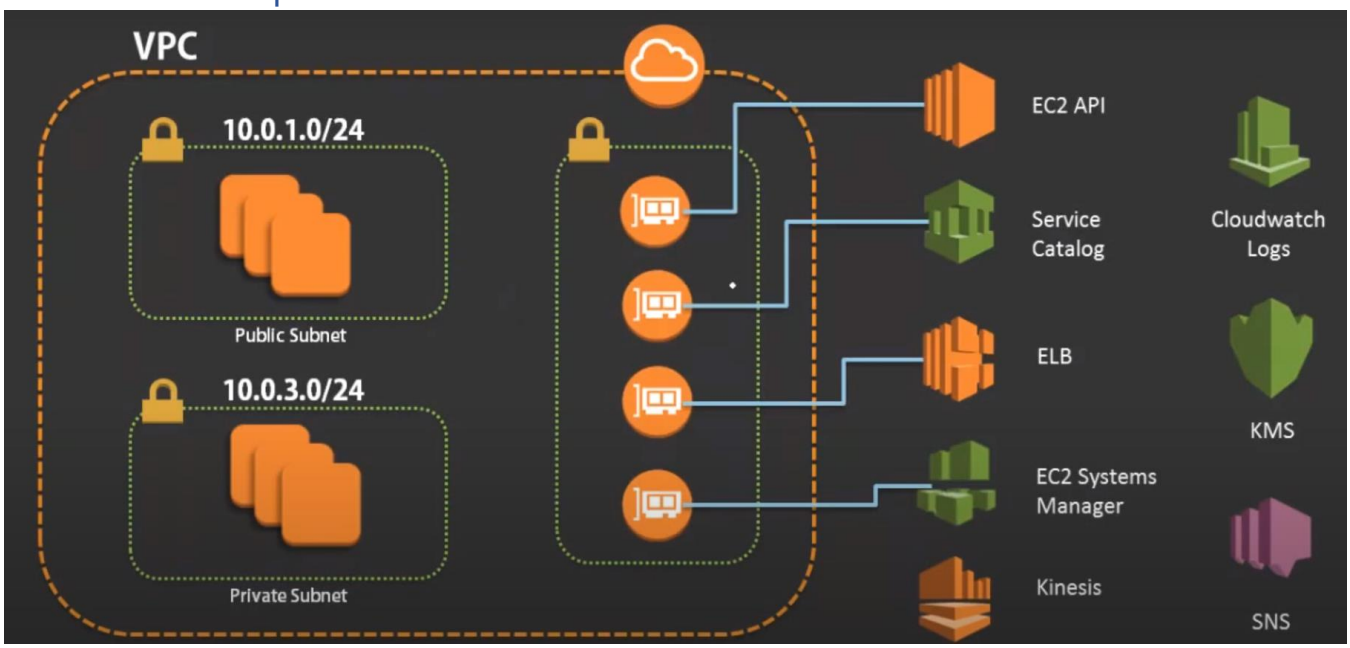



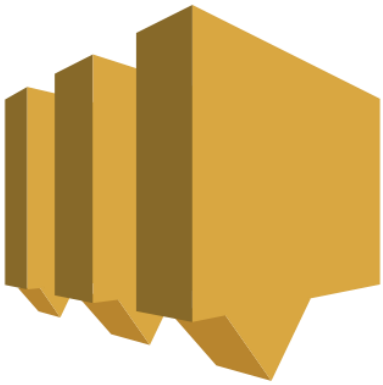

ec2-15-206-69-250.ap-south-1.compute.amazonaws.com
Public Subnet EC2 Instance
Hostname: EC2AMAZ-0QTVNM4
Instance ID: i-0e1d2ac77ec83882e
Public IP Address: 15.206.69.250
Private IP Address: 172.16.11.161
Instance Size: t2.micro
Availability Zone: ap-south-1a
Architecture: AMD64
Total Memory: 1 GB
Network Performance: Low to Moderate
Recycle Bin
EC2 Feedback
EC2 Microso...
stop-ec2
create-new
aws-linux
ec2-user@ip-172-16-21-64:~
Private Subnet Ec2 Instance
Using username "ec2-user".
Authenticating with public key "imported-openssh-key"
Last login: Sun Aug 2 21:08:53 2020 from ip-172-16-11-161.ap-south-1.compute.internal
Amazon Linux 2 AMI
https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-172-16-21-64 ~]$
[ec2-user@ip-172-16-21-64 ~]$
[ec2-user@ip-172-16-21-64 ~]$ aws configure set region ap-south-1
[ec2-user@ip-172-16-21-64 ~]$
[ec2-user@ip-172-16-21-64 ~]$
[ec2-user@ip-172-16-21-64 ~]$ aws s3 ls
2020-07-29 19:27:56 my-cloud-bucket01
2020-08-02 20:32:00 my-cloud-bucket02
[ec2-user@ip-172-16-21-64 ~]$
[ec2-user@ip-172-16-21-64 ~]$
[ec2-user@ip-172-16-21-64 ~]$ aws s3 ls s3://my-cloud-bucket02
2020-08-02 20:30:31 31978 AWS-Logo.png
[ec2-user@ip-172-16-21-64 ~]$
[ec2-user@ip-172-16-21-64 ~]$
This bucket is in ap-south-1, our EC2 also in ap-south-1.
We can't access the other one, since that is in different Region.
9:13 PM
8/2/2020

## VPC Interface Endpoints



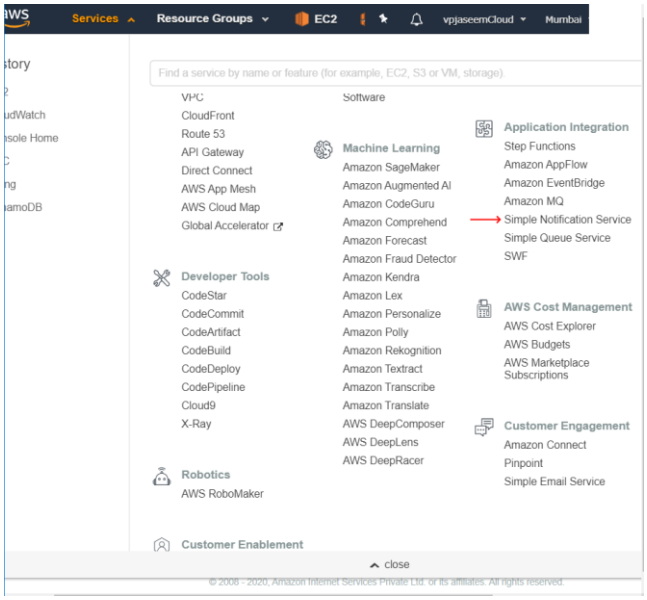

- Some other AWS services are accessed using VPC Interface Endpoint locally via AWS network rather than IGW.
- It's an Interface mapped to specific AWS service, once you deploy a load balancer, there will be an interface created automatically.

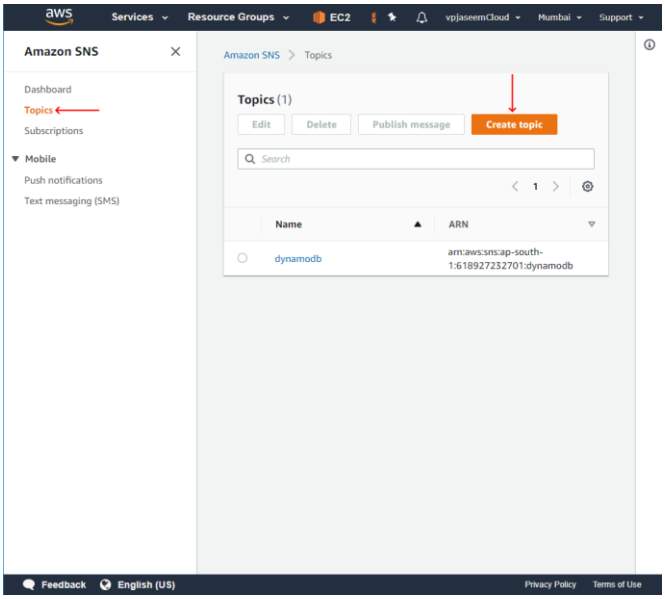
## SNS - Simple Notification Service

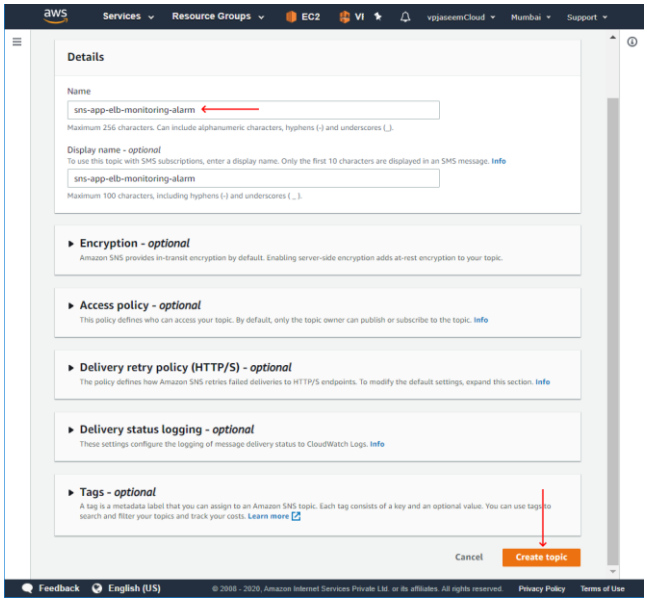


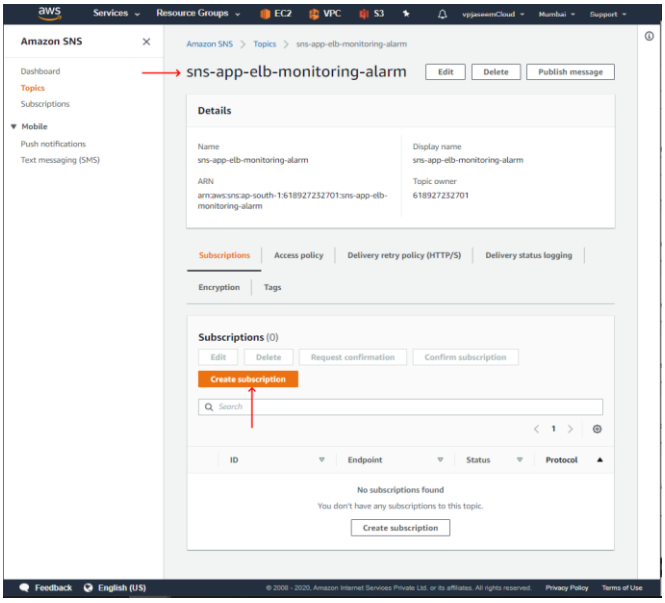
- Used to get notification for AWS resources
- Amazon Simple Notification Service (SNS) is a fully managed messaging service for both system-to-system and app-to-person (A2P) communication
- Amazon SNS enables you to send messages or notifications directly to users with SMS text messages to over 200 countries, mobile push on Apple, Android, and other platforms or email (SMTP)
- Amazon SNS leverages the proven AWS cloud to dynamically scale with your application.

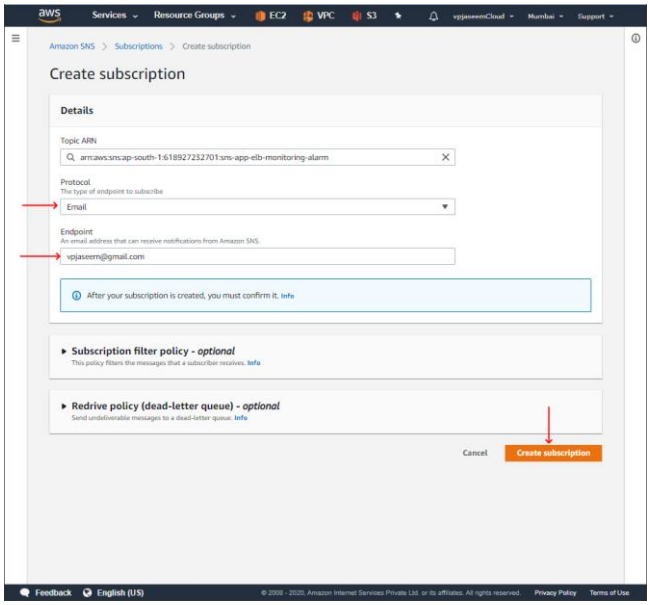
# [LAB] SNS Configuration Notification email for ELB Number of Connections

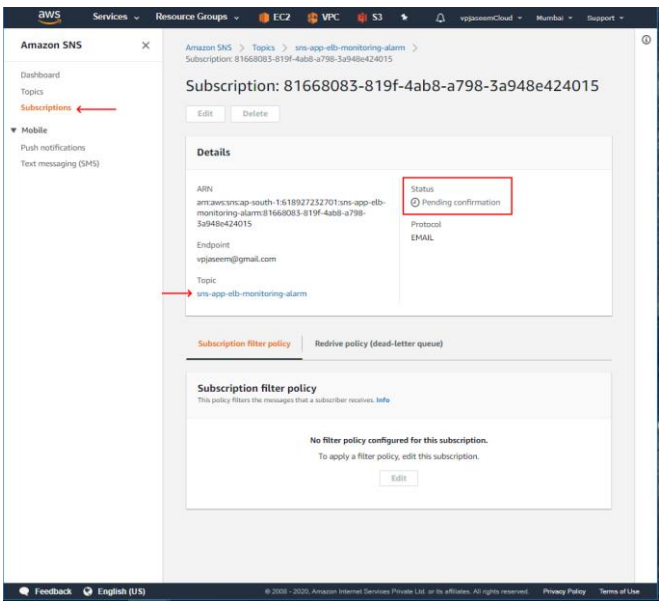


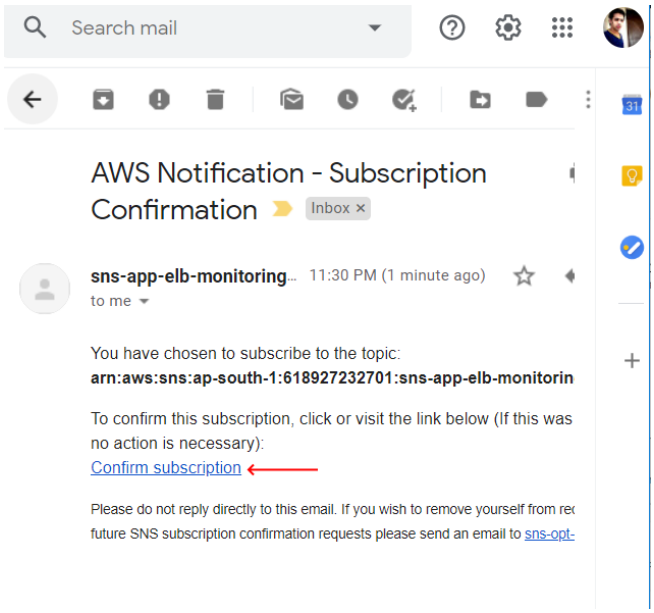



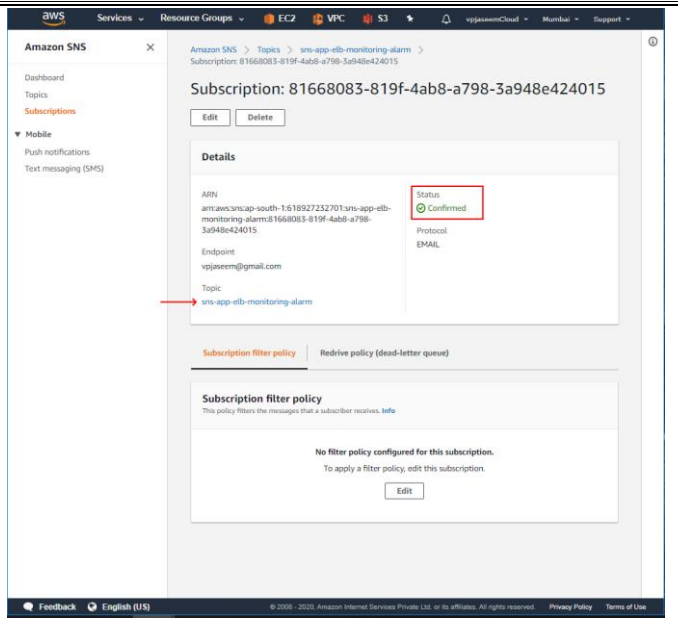



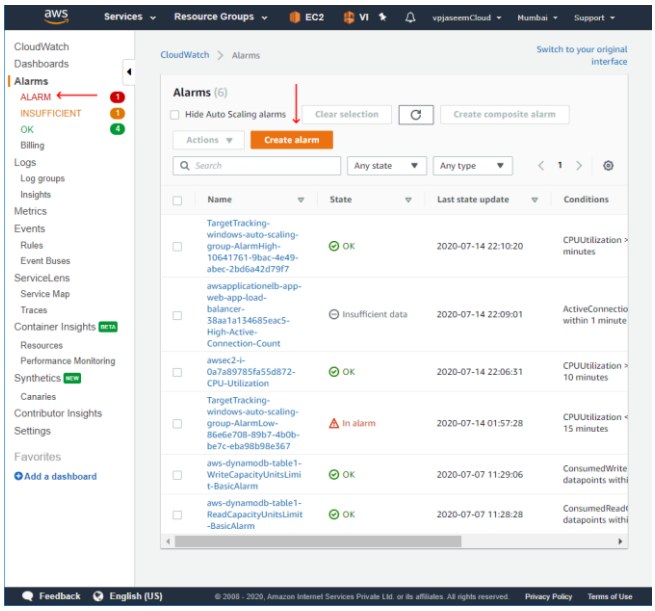



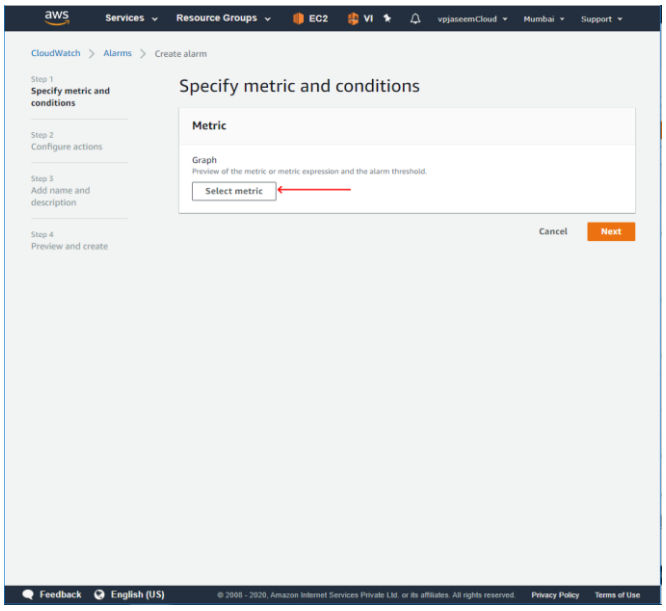



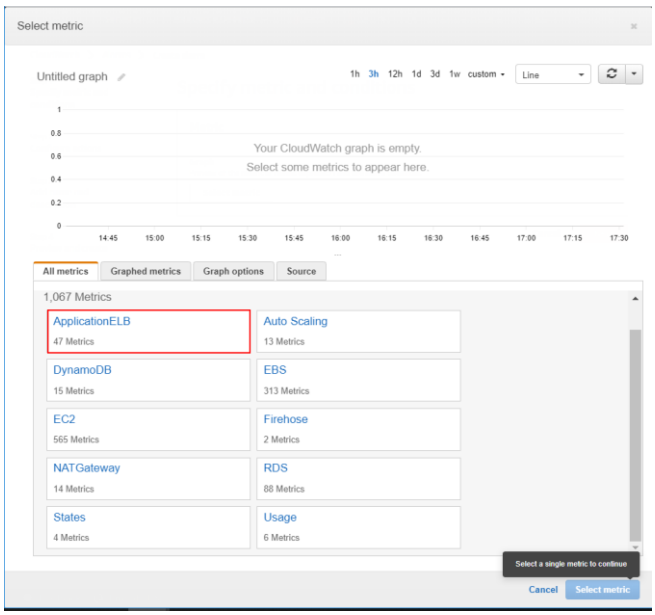



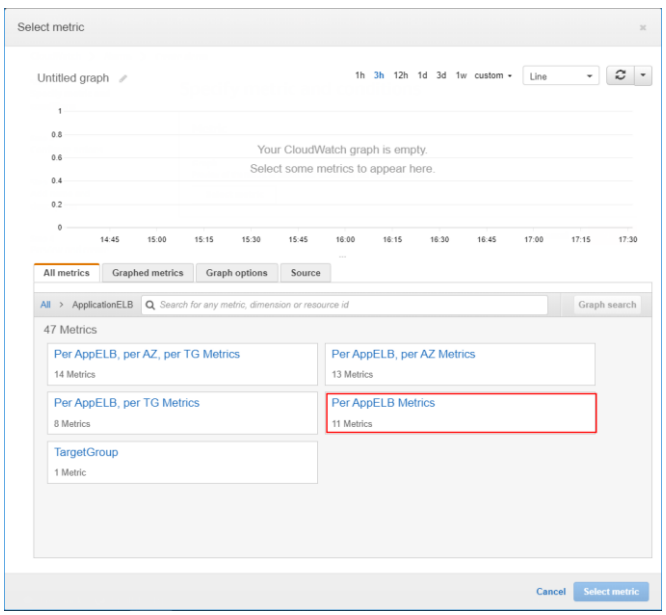

Search mail
AWS Notification - Subscription Confirmation
Inbox ×
sns-app-elb-monitoring... 11:30 PM (1 minute ago)
to me
You have chosen to subscribe to the topic:
arn:aws:sns:ap-south-1:618927232701:sns-app-elb-monitorin
To confirm this subscription, click or visit the link below (If this was no action is necessary):
Confirm subscription
Please do not reply directly to this email. If you wish to remove yourself from rec
future SNS subscription confirmation requests please send an email to sns-opt-

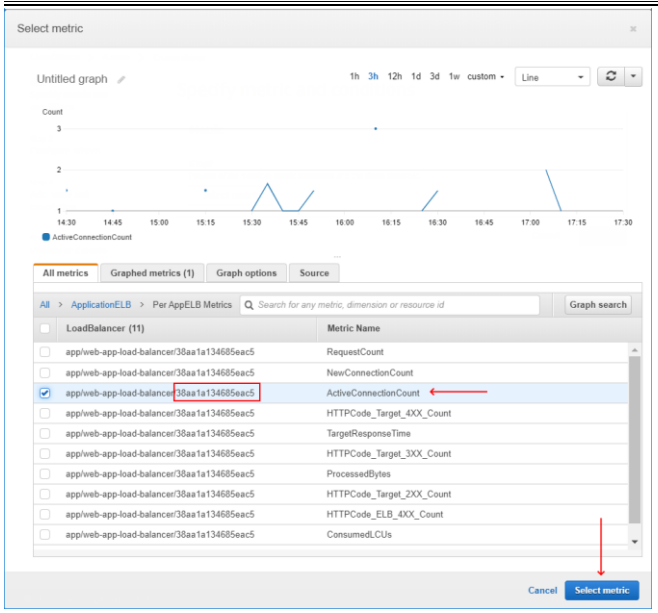

Amazon SNS
Subscription: 81668083-819f-4ab8-a798-3a948e424015
Edit
Delete
Details
Status
Confirmed
Protocol
EMAIL
Endpoint
vpjaseem@gmail.com
Topic
sns-app-elb-monitoring-alarm
Subscription filter policy
Redrive policy (dead-letter queue)
Subscription filter policy
No filter policy configured for this subscription.
To apply a filter policy, edit this subscription.
Edit

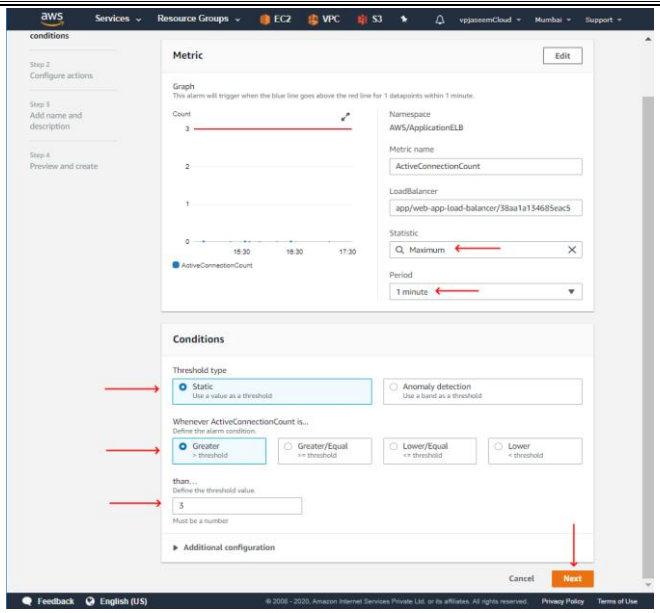

CloudWatch
Alarms (6)
Create alarm
Actions
Hide Auto Scaling alarms
Clear selection
Create composite alarm
Any state
Any type
Name
State
Last state update
Conditions

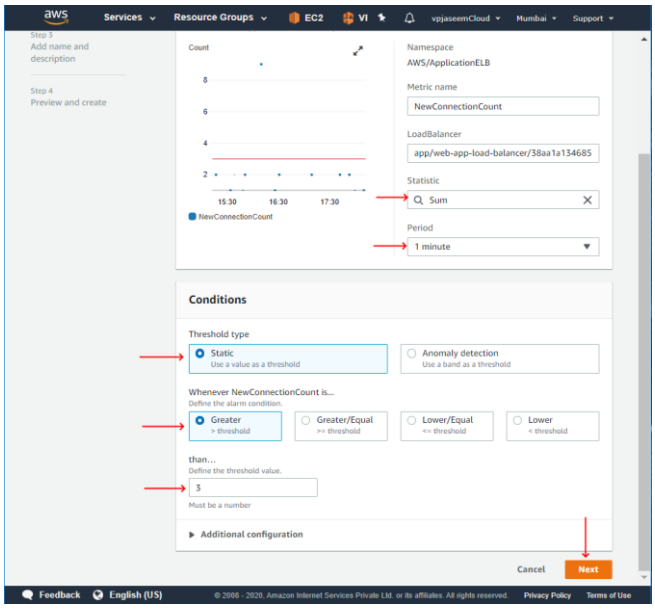

Specify metric and conditions
Metric
Graph
Select metric
Cancel
Next

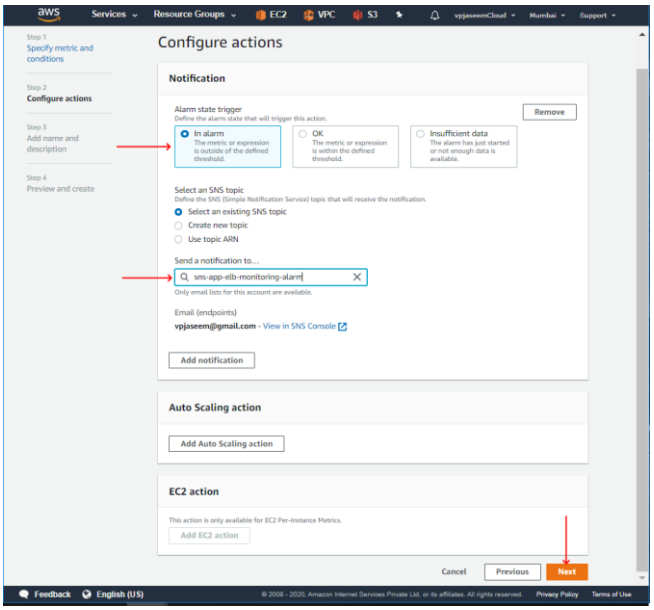

Select metric
Untitled graph
Your CloudWatch graph is empty.
Select some metrics to appear here.
All metrics
Graphed metrics
Graph options
Source
1,067 Metrics
ApplicationELB
47 Metrics
Auto Scaling
13 Metrics
DynamoDB
15 Metrics
EBS
313 Metrics
EC2
565 Metrics
Firehose
2 Metrics
NATGateway
14 Metrics
RDS
88 Metrics
States
4 Metrics
Usage
6 Metrics
Select a single metric to continue
Cancel
Select metric

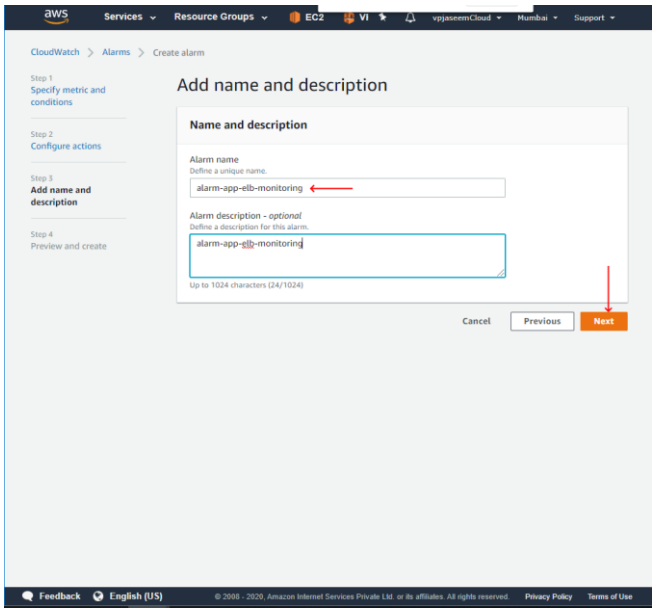

Select metric
Untitled graph
Your CloudWatch graph is empty.
Select some metrics to appear here.
All metrics
Graphed metrics
Graph options
Source
47 Metrics
Per AppELB, per AZ, per TG Metrics
14 Metrics
Per AppELB, per AZ Metrics
13 Metrics
Per AppELB, per TG Metrics
8 Metrics
Per AppELB Metrics
11 Metrics
TargetGroup
1 Metric
Cancel
Select metric

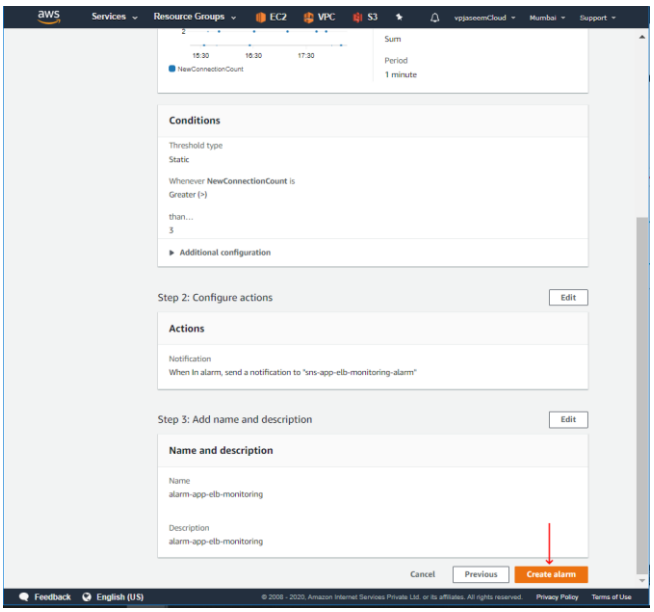

Select metric
Untitled graph
1h 3h 12h 1d 3d 1w custom
Line
Count
ActiveConnectionCount
All metrics
Graphed metrics (1)
Graph options
Source
All > ApplicationELB > Per AppELB Metrics
Search for any metric, dimension or resource id
Graph search
LoadBalancer (11)
Metric Name
app/web-app-load-balancer/38aa1a134685eac5
RequestCount
NewConnectionCount
ActiveConnectionCount
HTTPCode_Target_4XX_Count
TargetResponseTime
HTTPCode_Target_3XX_Count
ProcessedBytes
HTTPCode_Target_2XX_Count
HTTPCode_ELB_4XX_Count
ConsumedLCUs
Cancel
Select metric

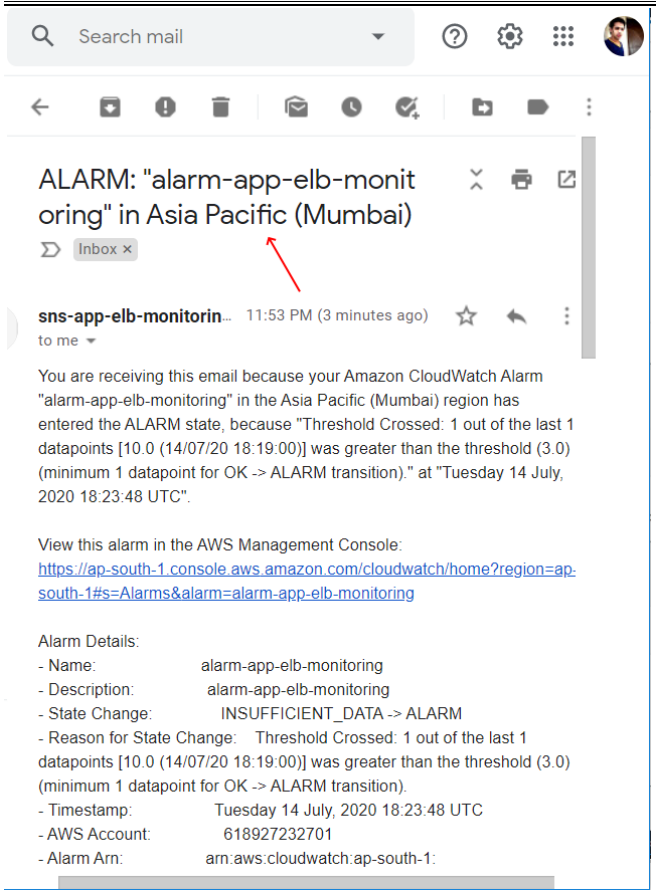

Metric
Edit
Graph
Namespace
AWS/ApplicationELB
Metric name
ActiveConnectionCount
LoadBalancer
app/web-app-load-balancer/38aa1a134685eac5
Statistic
Maximum
Period
1 minute
Conditions
Threshold type
Static
Use a value as a threshold
Anomaly detection
Use a band as a threshold
Whenever ActiveConnectionCount is...
Define the alarm condition.
Greater
> threshold
Greater/Equal
>= threshold
Lower/Equal
<= threshold
Lower
< threshold
than...
Define the threshold value.
3
Must be a number
Additional configuration
Cancel
Next
Feedback
English (US)
Privacy Policy
Terms of Use


Step 3
Add name and description
Step 4
Preview and create
Count
NewConnectionCount
Namespace
AWS/ApplicationELB
Metric name
NewConnectionCount
LoadBalancer
app/web-app-load-balancer/38aa1a134685
Statistic
Sum
Period
1 minute
Conditions
Threshold type
Static
Use a value as a threshold
Anomaly detection
Use a band as a threshold
Whenever NewConnectionCount is...
Define the alarm condition.
Greater
> threshold
Greater/Equal
>= threshold
Lower/Equal
<= threshold
Lower
< threshold
than...
Define the threshold value.
3
Must be a number
Additional configuration
Cancel
Next
Feedback
English (US)
Privacy Policy
Terms of Use

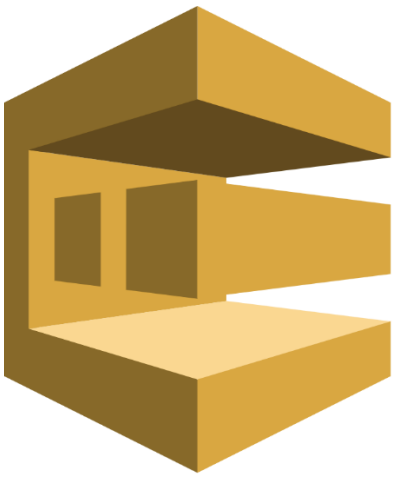

Step 1
Specify metric and conditions
Step 2
Configure actions
Step 3
Add name and description
Step 4
Preview and create
Configure actions
Notification
Alarm state trigger
Define the alarm state that will trigger this action.
Remove
In alarm
The metric or expression is outside of the defined threshold.
OK
The metric or expression is within the defined threshold.
Insufficient data
The alarm has just started or not enough data is available.
Select an SNS topic
Define the SNS (Simple Notification Service) topic that will receive the notification.
Select an existing SNS topic
Create new topic
Use topic ARN
Send a notification to...
sns-app-elb-monitoring-alarm
Only email lists for this account are available.
Email (endpoints)
vpjaseem@gmail.com - View in SNS Console
Add notification
Auto Scaling action
Add Auto Scaling action
EC2 action
This action is only available for EC2 Per-Instance Metrics.
Add EC2 action
Cancel
Previous
Next
Feedback
English (US)
Privacy Policy
Terms of Use


CloudWatch > Alarms > Create alarm
Step 1
Specify metric and conditions
Step 2
Configure actions
Step 3
Add name and description
Step 4
Preview and create
Add name and description
Name and description
Alarm name
Define a unique name.
alarm-app-elb-monitoring
Alarm description - optional
Define a description for this alarm.
alarm-app-elb-monitoring
Up to 1024 characters (24/1024)
Cancel
Previous
Next
Feedback
English (US)
Privacy Policy
Terms of Use


Sum
Period
1 minute
Conditions
Threshold type
Static
Whenever NewConnectionCount is
Greater (>)
than...
3
Additional configuration
Step 2: Configure actions
Edit
Actions
Notification
When In alarm, send a notification to "sns-app-elb-monitoring-alarm"
Step 3: Add name and description
Edit
Name and description
Name
alarm-app-elb-monitoring
Description
alarm-app-elb-monitoring
Cancel
Previous
Create alarm
Feedback
English (US)
Privacy Policy
Terms of Use


Search mail
ALARM: "alarm-app-elb-monit
oring" in Asia Pacific (Mumbai)
Inbox ×
sns-app-elb-monitorin... 11:53 PM (3 minutes ago)
to me
You are receiving this email because your Amazon CloudWatch Alarm "alarm-app-elb-monitoring" in the Asia Pacific (Mumbai) region has entered the ALARM state, because "Threshold Crossed: 1 out of the last 1 datapoints [10.0 (14/07/20 18:19:00)] was greater than the threshold (3.0) (minimum 1 datapoint for OK -> ALARM transition)." at "Tuesday 14 July, 2020 18:23:48 UTC".
View this alarm in the AWS Management Console:
https://ap-south-1.console.aws.amazon.com/cloudwatch/home?region=ap-south-1#s=Alarms&alarm=alarm-app-elb-monitoring
Alarm Details:
- Name: alarm-app-elb-monitoring
- Description: alarm-app-elb-monitoring
- State Change: INSUFFICIENT_DATA -> ALARM
- Reason for State Change: Threshold Crossed: 1 out of the last 1 datapoints [10.0 (14/07/20 18:19:00)] was greater than the threshold (3.0) (minimum 1 datapoint for OK -> ALARM transition).
- Timestamp: Tuesday 14 July, 2020 18:23:48 UTC
- AWS Account: 618927232701
- Alarm Arn: arn:aws:cloudwatch:ap-south-1:

## SQS - Simple Queuing Service



- Amazon Simple Queue Service (SQS) is a fully managed message queuing service that enables you to decouple and scale microservices, distributed systems, and serverless applications.

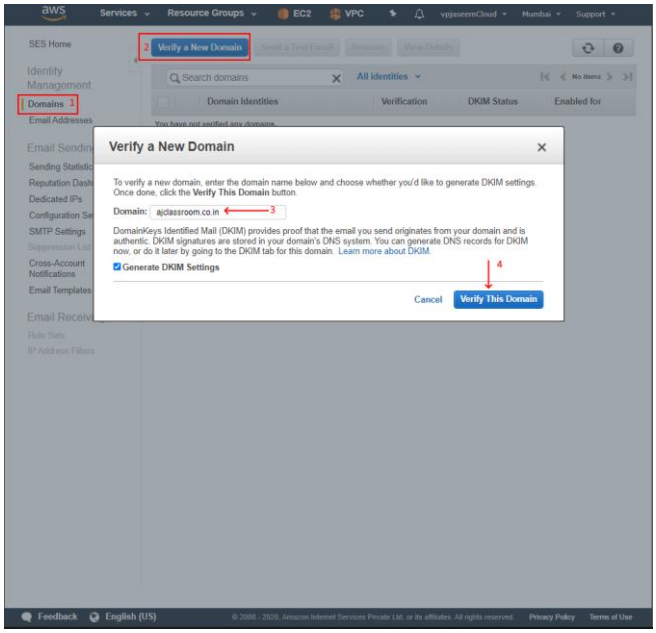
| SNS | SQS |
|---|---|
| Publisher / Subscriber System. Topic is subscribed by other party | Queuing service for message processing. SNS can send message to SQS and SQS process later. |
| Message deliver to many subscribers | Messages lies in the queue, messages in the que are processed by single consumer. (Poll) |
| Use Case: Do other systems care about the event? | Use Case: Does your system care about an event? |

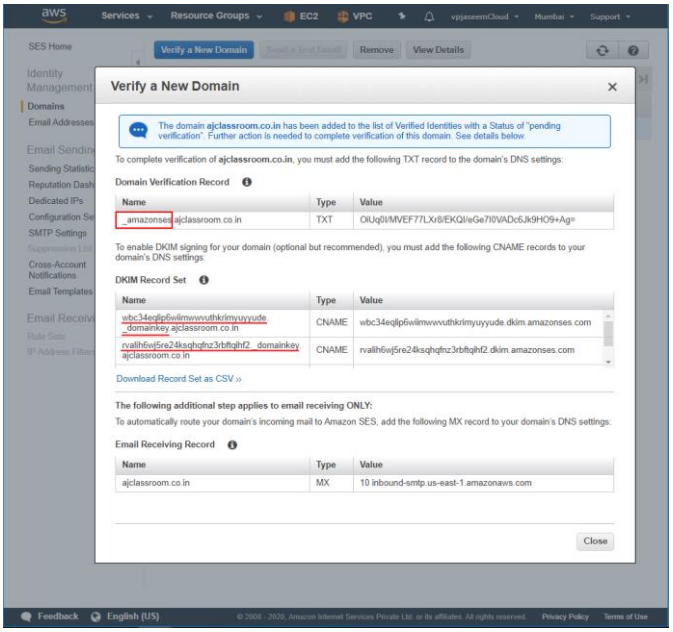
# Simple Email Service (SES)

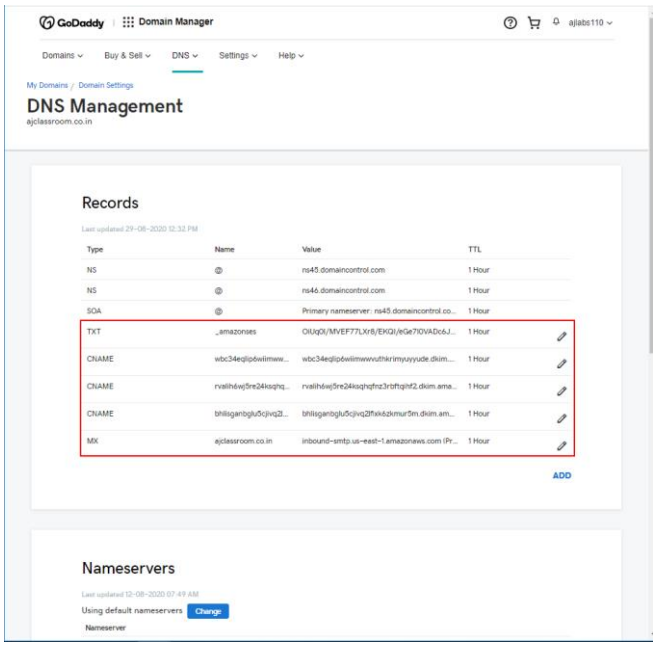
- Amazon managed SMTP server, we must have a domain name to go with SES.
- Amazon Simple Email Service (SES) is a cost-effective, flexible, and scalable email service that enables developers to send mail from within any application.

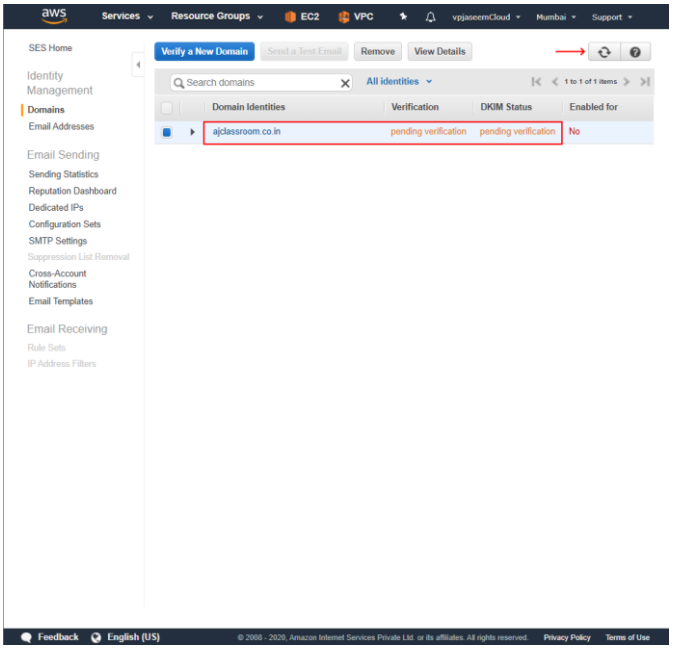
## [LAB] Simple Email Service

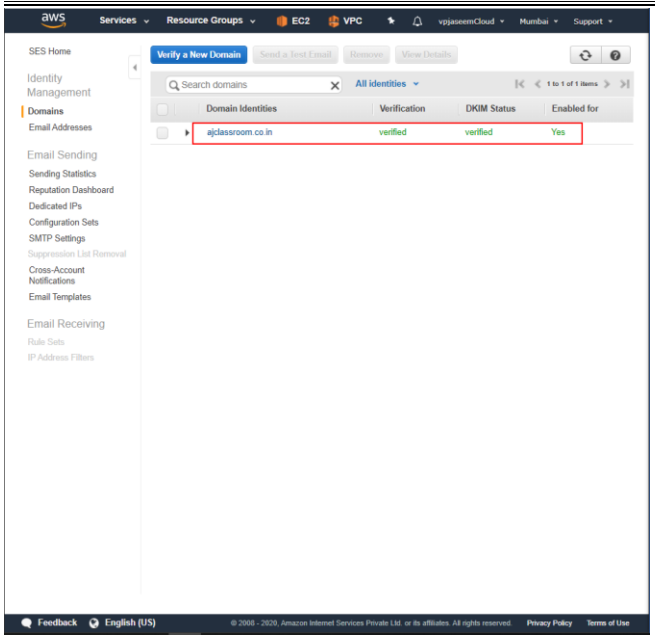
- I'm using Sandbox feature, can send emails to only to verified users. If you are in production, you can enable production grade.
- This is again used for API or Programs to send emails (e.g. Email marketing programs)

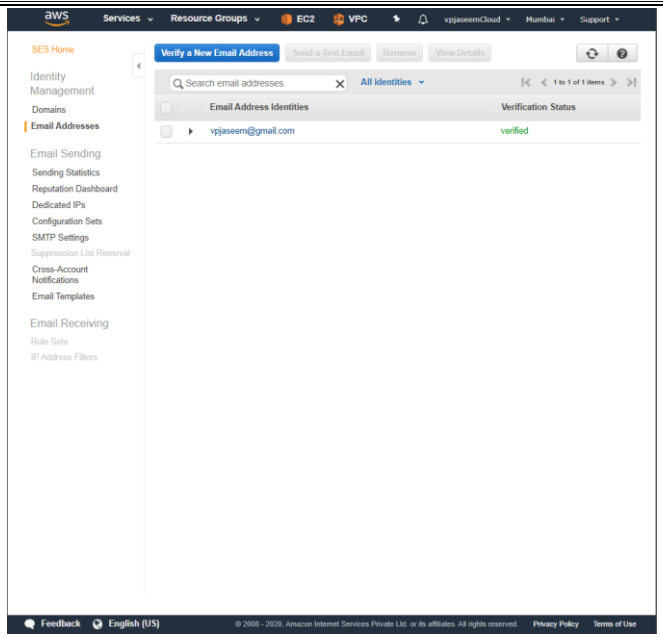


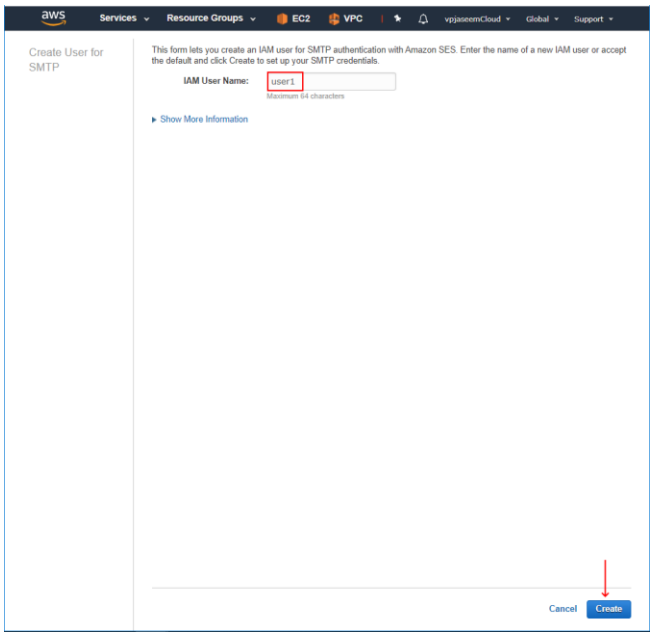



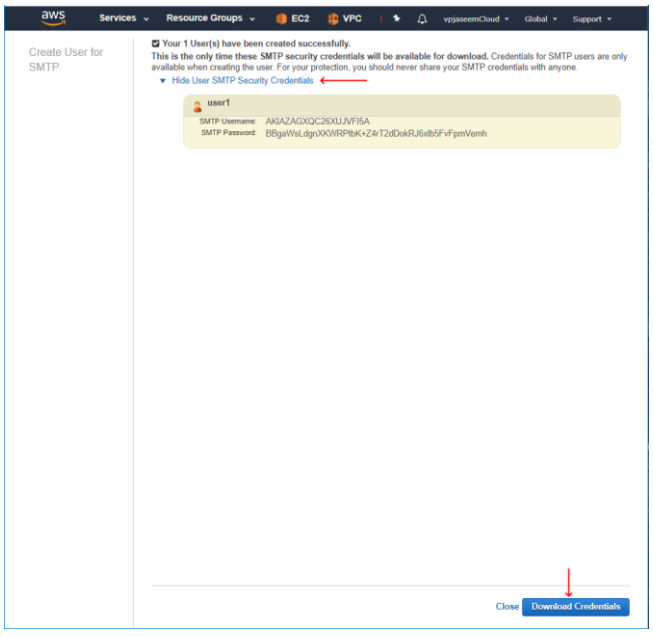




Services
Resource Groups
EC2
VPC
vpjaseemCloud
Mumbai
Support
SES Home
Identity Management
Domains
Email Addresses
Email Sending
Sending Statistics
Reputation Dashboard
Dedicated IPs
Configuration Sets
SMTP Settings
Suppression List Removal
Cross-Account Notifications
Email Templates
Email Receiving
Rule Sets
IP Address Filters
Verify a New Domain
Send a Test Email
Remove
View Details
Search domains
All identities
1 to 1 of 1 items
Domain Identities
Verification
DKIM Status
Enabled for
ajclassroom.co.in
verified
verified
Yes
Feedback
English (US)
© 2008 - 2020, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use


Services
Resource Groups
EC2
VPC
vpjaseemCloud
Mumbai
Support
SES Home
Identity Management
Domains
Email Addresses
Email Sending
Sending Statistics
Reputation Dashboard
Dedicated IPs
Configuration Sets
SMTP Settings
Suppression List Removal
Cross-Account Notifications
Email Templates
Email Receiving
Rule Sets
IP Address Filters
Verify a New Email Address
Send a Test Email
Remove
View Details
Search email addresses
All identities
1 to 1 of 1 items
Email Address Identities
Verification Status
vpjaseem@gmail.com
verified
Feedback
English (US)
© 2008 - 2020, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use

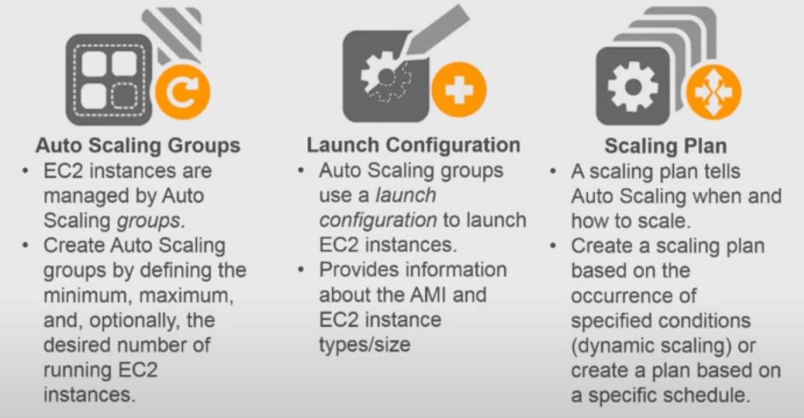

Services
Resource Groups
EC2
VPC
vpjaseemCloud
Mumbai
Support
SES Home
Identity Management
Domains
Email Addresses
Email Sending
Sending Statistics
Reputation Dashboard
Dedicated IPs
Configuration Sets
SMTP Settings
Suppression List Removal
Cross-Account Notifications
Email Templates
Email Receiving
Rule Sets
IP Address Filters
Using SMTP to Send Email with Amazon SES
You can send email through Amazon SES by using applications or programming languages that use the SMTP protocol. To learn more about the Amazon SES SMTP interface, see Using the Amazon SES SMTP Interface to Send Email.
To send email by using the SMTP interface, you need the following information:
Server Name: email-smtp.ap-south-1.amazonaws.com
Port: 25, 465 or 587
Use Transport Layer Security (TLS): Yes
Authentication: Your SMTP credentials. See below for more information.
To send email using the Amazon SES SMTP interface, you first have to create a set of SMTP credentials. Your SMTP credentials are the user name and password that you use to connect to an Amazon SES SMTP endpoint. SMTP credentials are unique to each AWS Region. If you want to send email in more than one Region, you have to generate a unique set of credentials for each Region.
To obtain SMTP credentials for the current Region, choose Create My SMTP Credentials. For more information about SMTP credentials, see Obtaining Your Amazon SES SMTP Credentials.
Create My SMTP Credentials
Note: Your SMTP credentials are different from your AWS access keys. The two types of credentials aren't interchangeable. For more information, see Using Credentials With Amazon SES.
Feedback
English (US)
© 2008 - 2020, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use


Services
Resource Groups
EC2
VPC
vpjaseemCloud
Global
Support
Create User for SMTP
This form lets you create an IAM user for SMTP authentication with Amazon SES. Enter the name of a new IAM user or accept the default and click Create to set up your SMTP credentials.
IAM User Name:
user1
Maximum 64 characters
Show More Information
Cancel
Create


Services
Resource Groups
EC2
VPC
vpjaseemCloud
Global
Support
Create User for SMTP
Your 1 User(s) have been created successfully.
This is the only time these SMTP security credentials will be available for download. Credentials for SMTP users are only available when creating the user. For your protection, you should never share your SMTP credentials with anyone.
Hide User SMTP Security Credentials
user1
SMTP Username: AKIAZAGXQC26XUJVFI5A
SMTP Password: BBgaWsLdgnXKWRPtbK+Z4rT2dDokRJ6xlb5FvFpmVemh
Close
Download Credentials

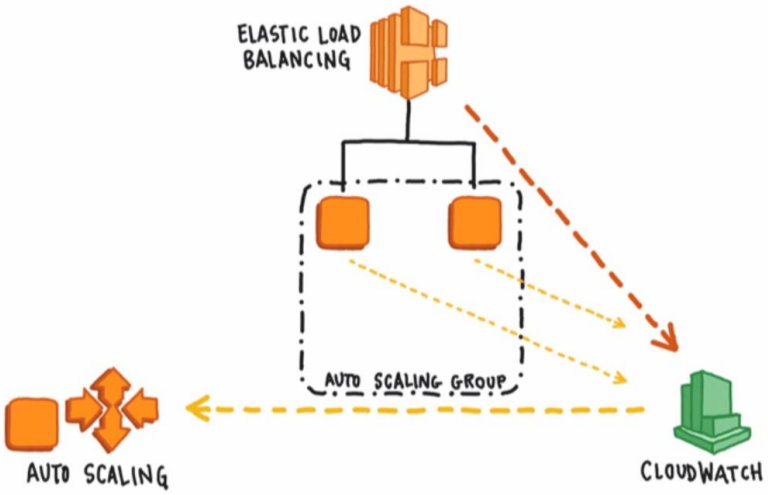
## Auto Scaling

- Add more EC2 instances automatically when demand is more.
- Always keep minimum number of instances running.
- Works across AZs and multiple subnets.
- Elastic Load Balances (ELB) tied with all the instances
- Based on the Launch configuration, Autoscaling determine EC2 Instance Type, AMI, Security Group, etc.
- Installation and setup need to be fully automatic. Tuning the server for after the launch is called Bootstrapping - Done via User Data scripts.
- Use amazon Golden Image (Create an EC2 instance, install everything you need on that, convert that as a AMI). Auto Scaling will use this image to launch new EC2s
- If Using Golden Image is not practical, use Base AMI + User data to inject configurations (or Chef/Puppet/Ansible/AWS CodeDeploy)
- If the load is going down, Auto Scaling will terminate instance (wait for no active session)
- Termination Policy used to determine which instances to be terminated (Longest Running, Oldest launch configuration, Closest to full billing hour)
- Scaling Plan: When to do scale up and down
  - Default: Based on the launch configuration
  - Manual: Modify desired capacity via Console, API or CLI (Monitoring tool output fed to some sort of script to perform API call to increase capacity)
  - Scheduled: Scale in / out based on time
  - Dynamic Scaling: Based AWS CloudWatch metrices




- Commonly used for Web based workload with an ELB
- Elastic Load Balances (ELB): Distribute incoming web traffic to automatically, single point of entry to our application. It sends data about ELB and EC2 to CloudWatch.
- Autoscaling uses health check,
  - EC2 Instance Status: Instance is unhealth if it is not running
  - ELB Health Checks: ELB sends some HTTP request to see if it is getting responses
  - Manual: Mark an instance unhealthy via API

- CloudWatch: Is a web service that enables to monitor and manage various metrices and configure alarms based on those data (e.g. CPU usage / Incoming traffic)

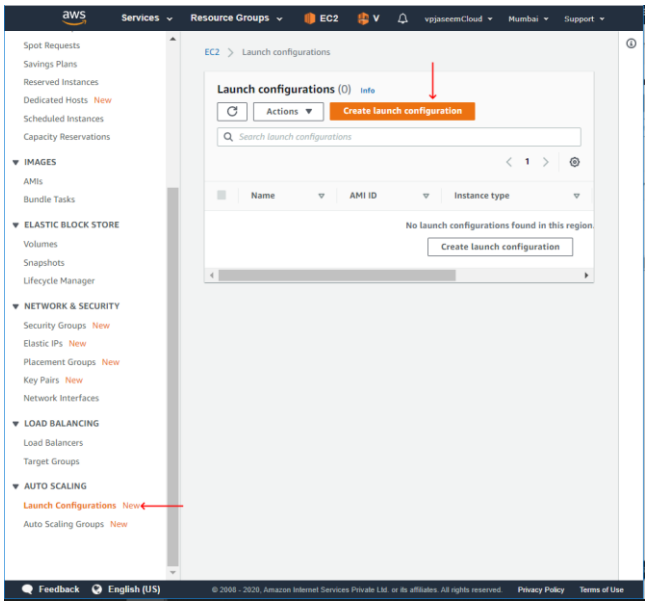


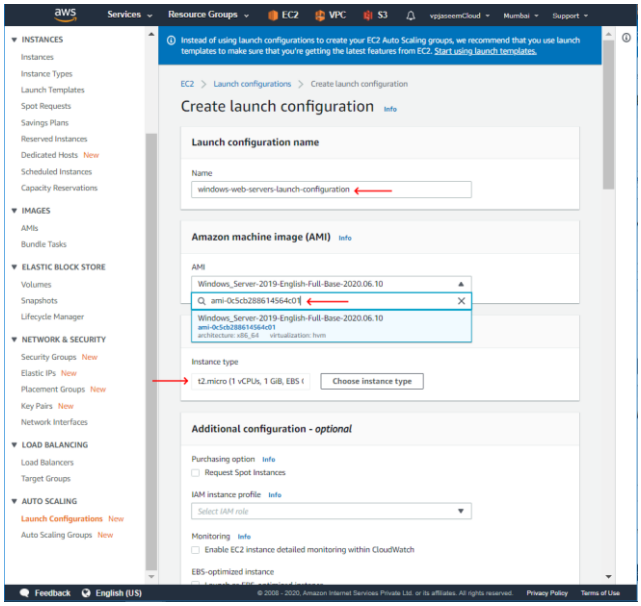

- Based on the load / demand, the entire system growing or shrinking
- Auto Scaler manages the Auto Scaling Group, Load balances (ELB)
- Scale out: Adding more instances
- Scale in: Removing instances

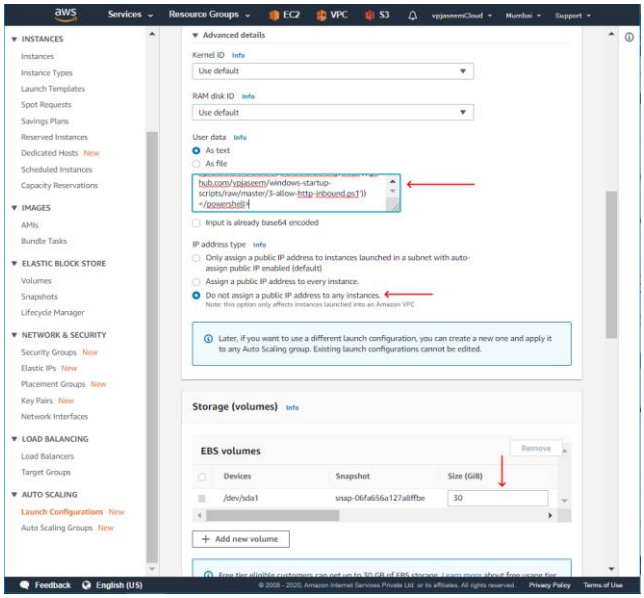
- Simple Scaling Policy: Use any CloudWatch Matric as load matric, specify single threshold beyond which you want to scale.
- Step Scaling Policy: Use any CloudWatch Matric as load matric, specify different threshold which you want to scale.
- Target Tracking Scaling: Specify CPU or Memory

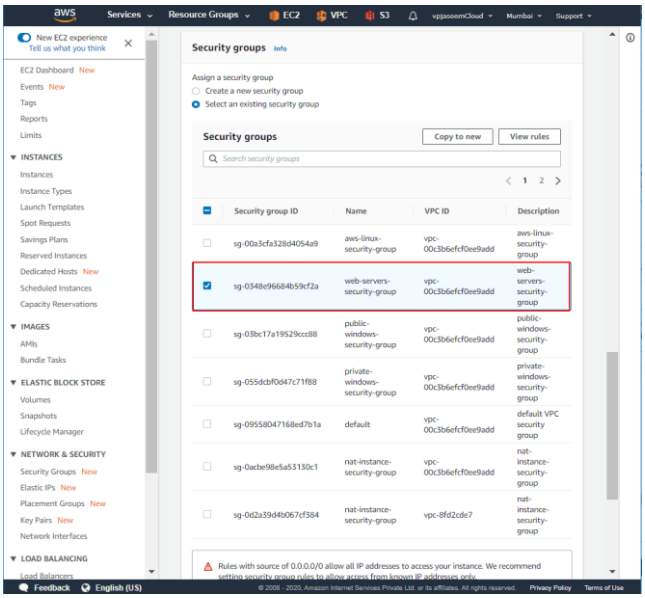
# [LAB] Auto Scaling Configuration

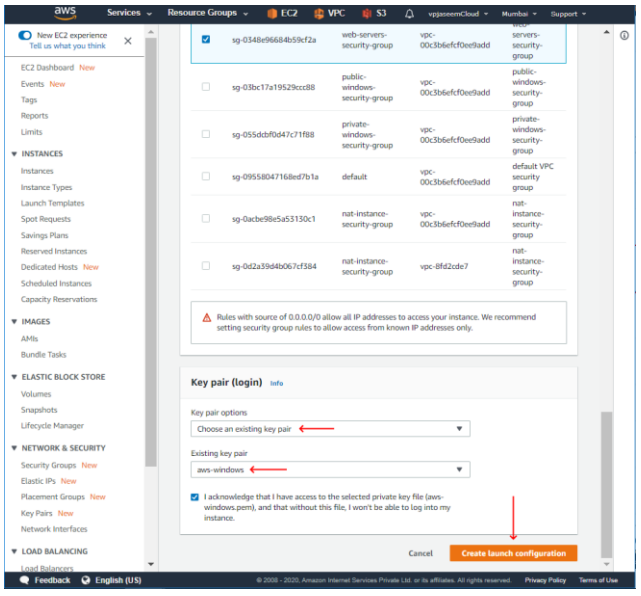
- Launch configuration: Collection of all the attributes those are required to launch an EC2

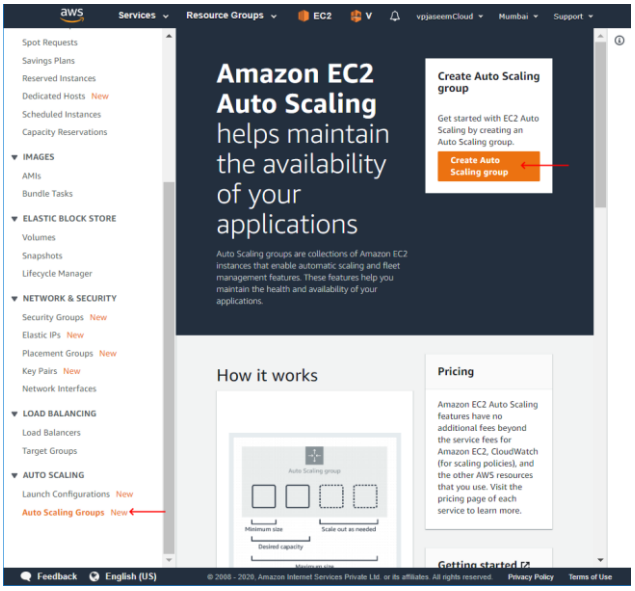


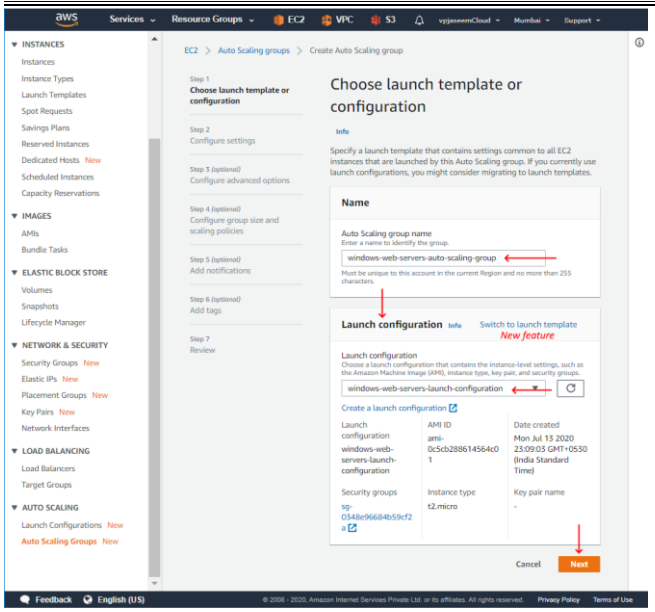



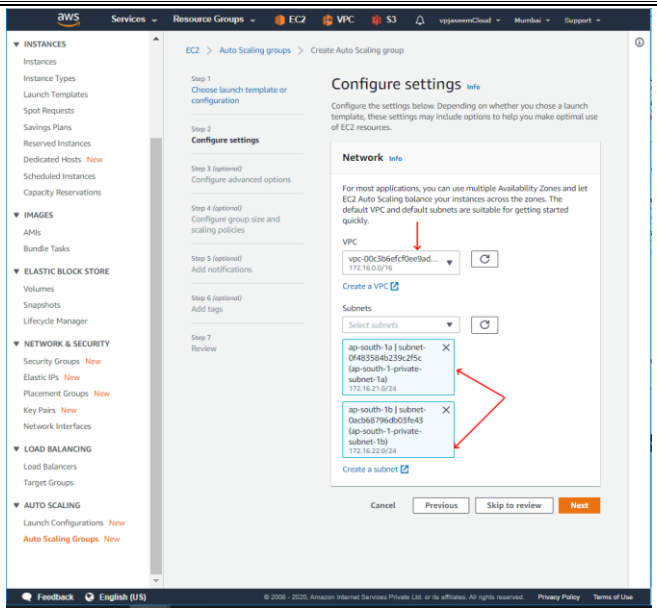



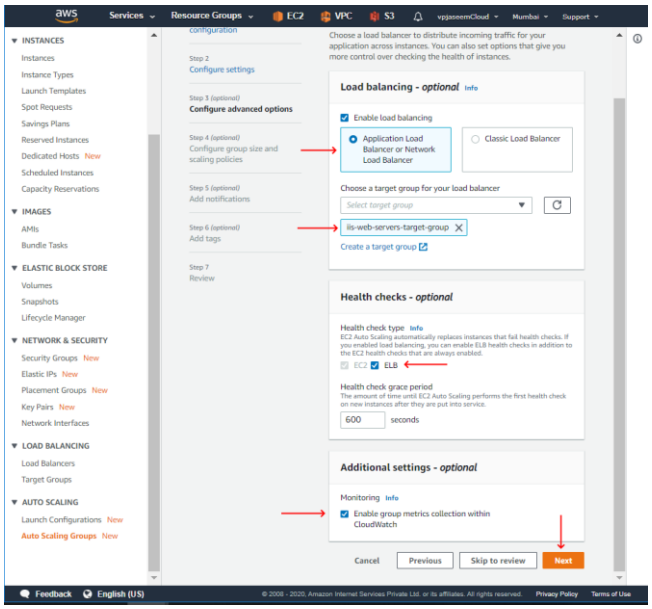



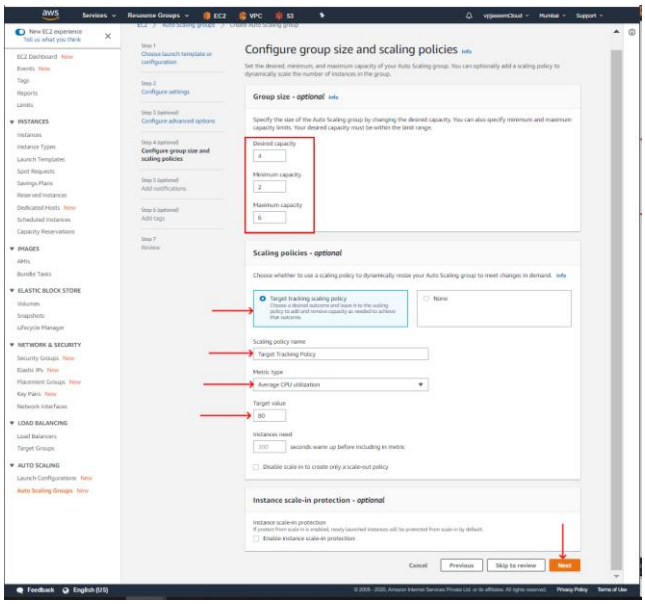



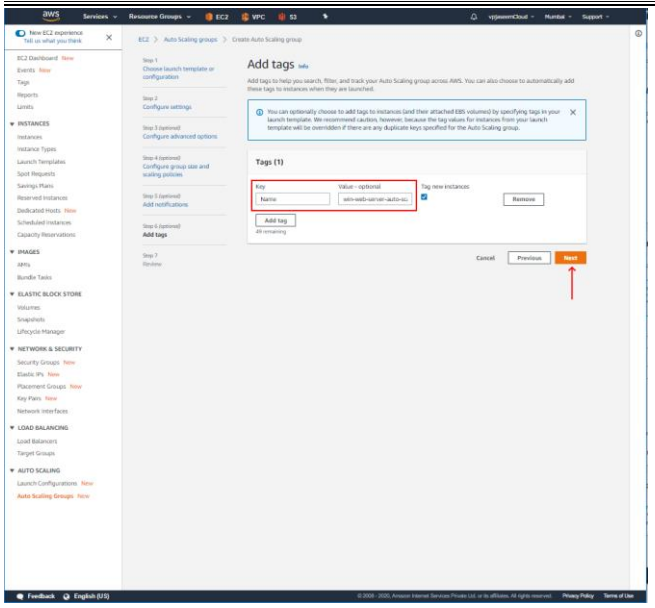


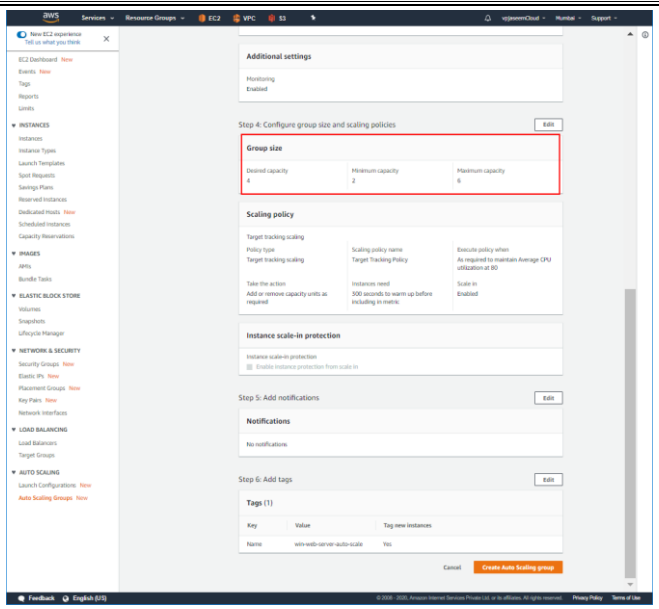



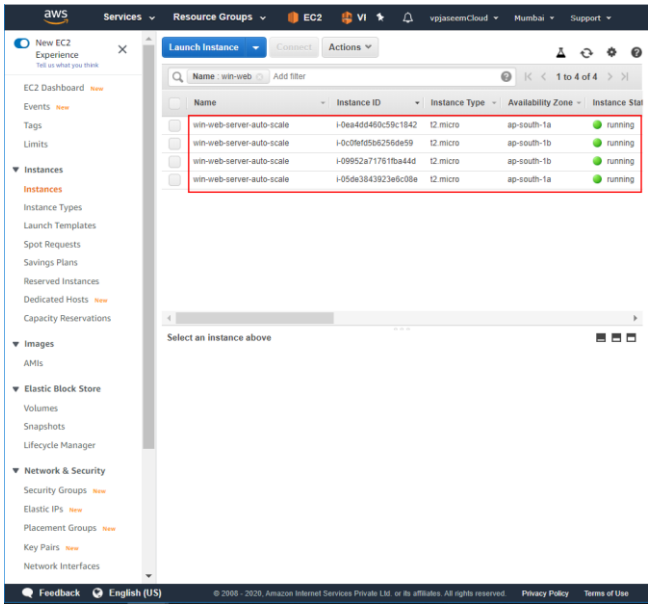



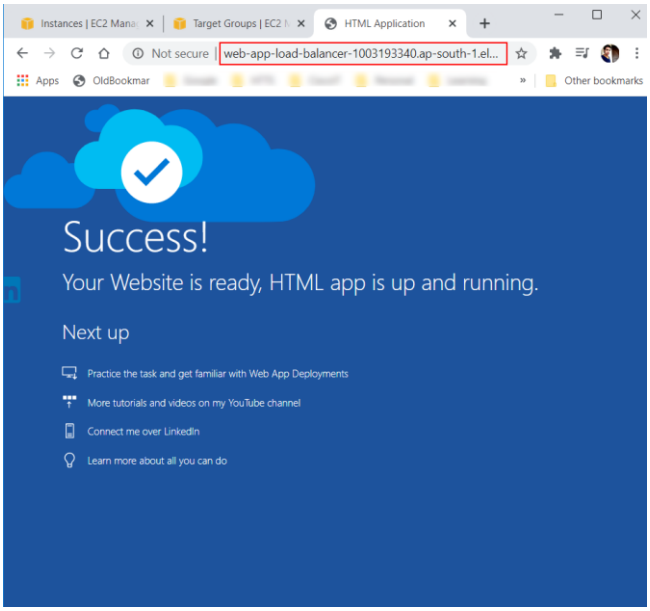

- Make sure you remove other 2 EC2 instances from the load balancer that we configured in previous lab (ELB Lab)
- For that edit the target group and remove instances.
- Otherwise create a new load balancer after the Auto Scaling configuration and attach to this.

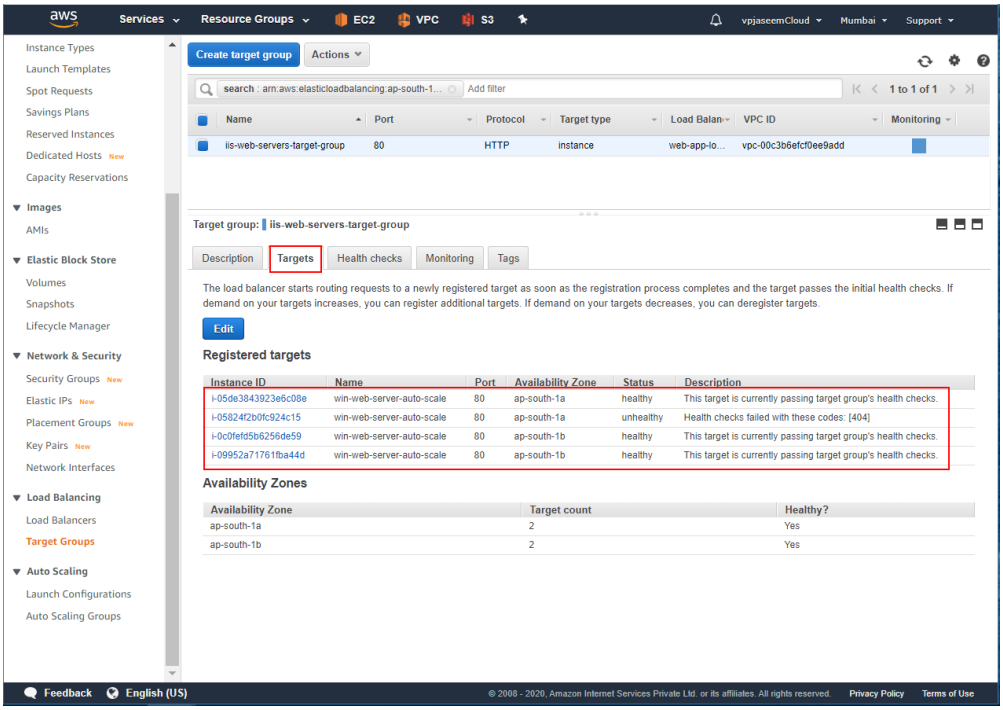






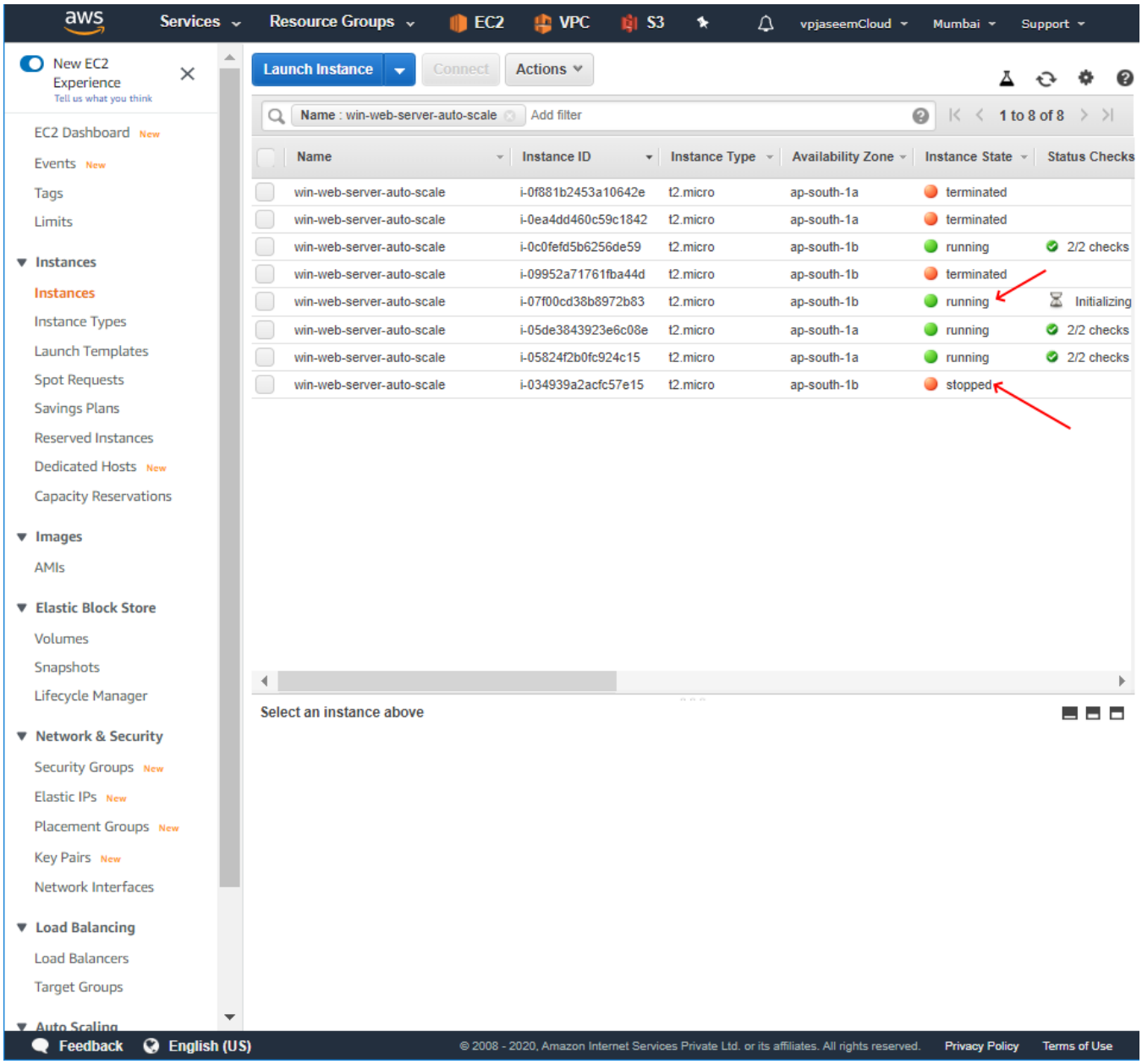



Success!
Your Website is ready, HTML app is up and running.
Next up
Practice the task and get familiar with Web App Deployments
More tutorials and videos on my YouTube channel
Connect me over LinkedIn
Learn more about all you can do


Create target group
Actions
Target group: iis-web-servers-target-group
Description
Targets
Health checks
Monitoring
Tags
The load balancer starts routing requests to a newly registered target as soon as the registration process completes and the target passes the initial health checks. If demand on your targets increases, you can register additional targets. If demand on your targets decreases, you can deregister targets.
Edit
Registered targets
Availability Zones

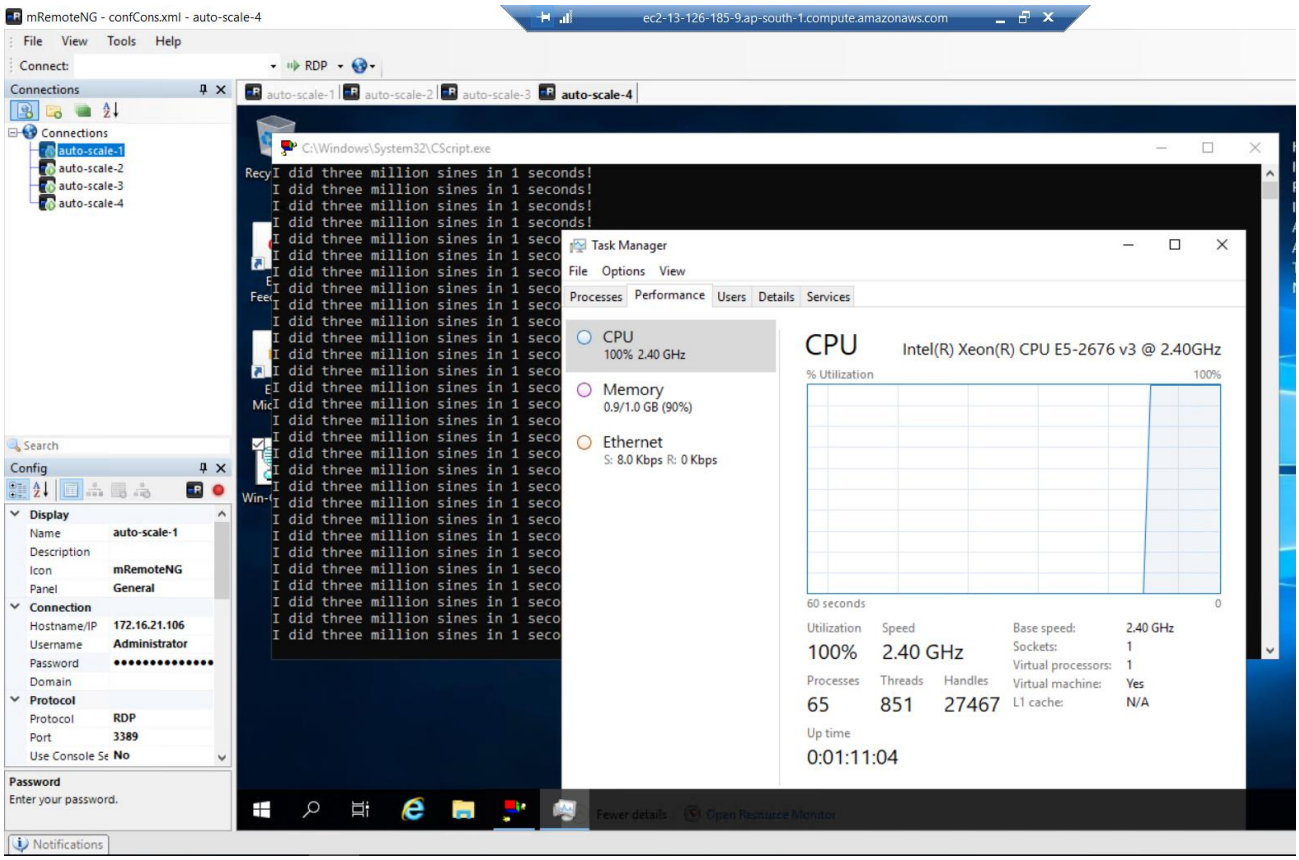
## [LAB] Test Auto Scaling Lab: Unhealthy Instance

- Auto Scaling replaces unhealthy instances as well as scale out (add more) and scale in (terminate unwanted)
- Let's test what if the health check fails? So far, we have 4 desired instances, if I shutdown an instance manually, the health check fails (get request to /index/html return 404).
- Auto Scaler will launch new instance and terminate the other one

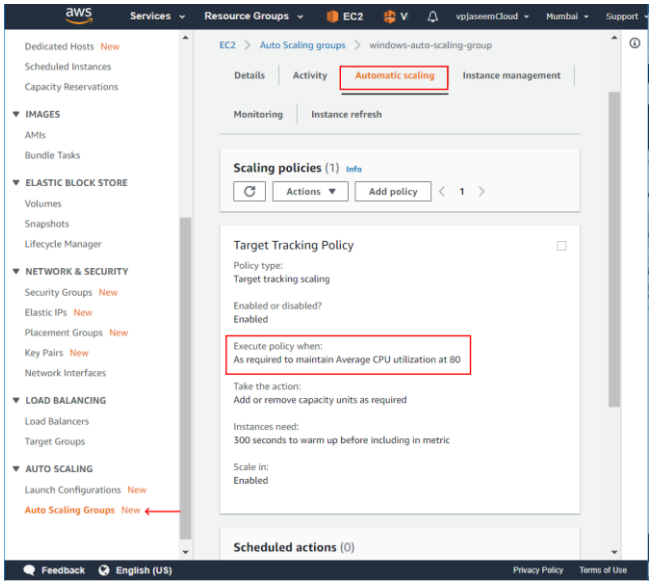


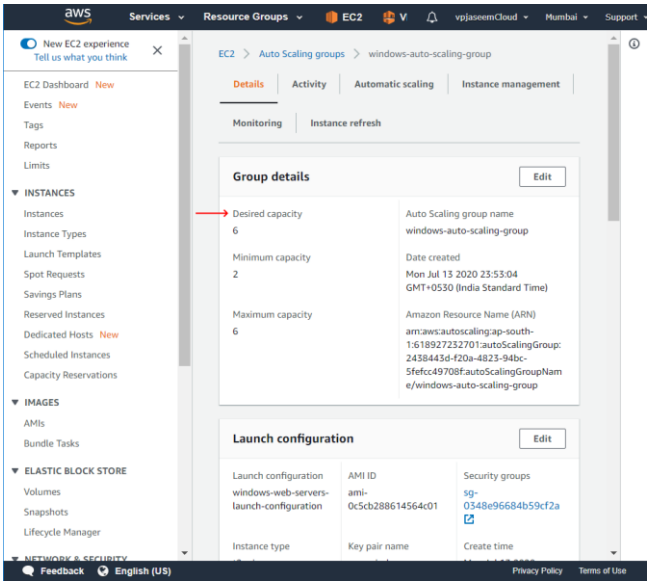
## [LAB] Test Auto Scaling Lab: CPU Utilization Scale Out

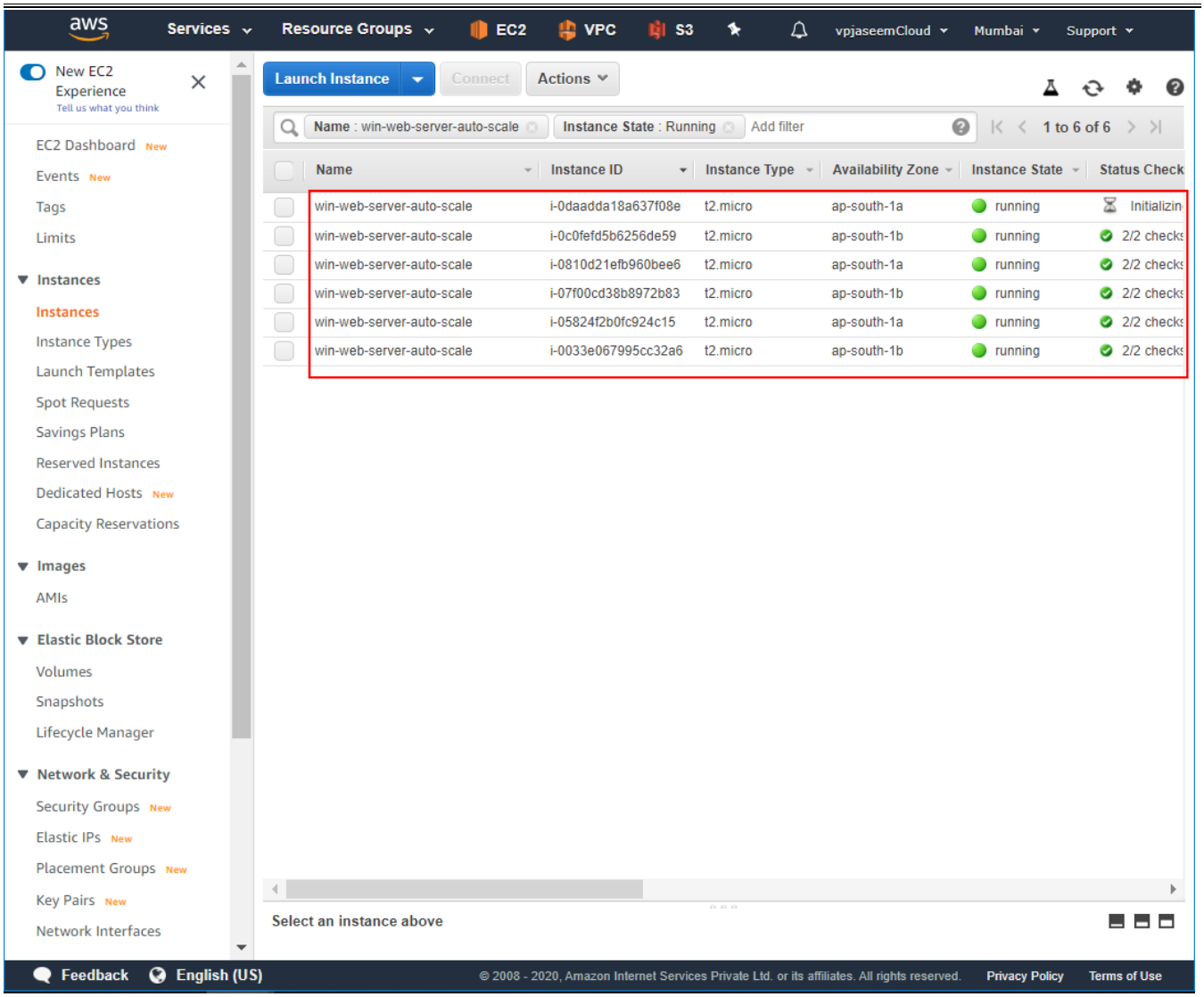
- I'm going to run a script on 2 EC2s and that will over utilize the CPU. This activity kicks Scale Out. Download the script: Win-CPUBUSY.VBS
- I have used mRemoteNG software to connect all the instances and run the Script. You right click on the script and use 'Run with command prompt'




- At this stage the desired capacity changes automatically to 6




aws
Services
Resource Groups
EC2
VPC
S3
vpjaseemCloud
Mumbai
Support
New EC2 Experience
Tell us what you think
EC2 Dashboard New
Events New
Tags
Limits
Instances
Instances
Instance Types
Launch Templates
Spot Requests
Savings Plans
Reserved Instances
Dedicated Hosts New
Capacity Reservations
Images
AMIs
Elastic Block Store
Volumes
Snapshots
Lifecycle Manager
Network & Security
Security Groups New
Elastic IPs New
Placement Groups New
Key Pairs New
Network Interfaces
Launch Instance
Connect
Actions
Name : win-web-server-auto-scale
Instance State : Running
Add filter
1 to 6 of 6
Name
Instance ID
Instance Type
Availability Zone
Instance State
Status Check
win-web-server-auto-scale i-0daadda18a637f08e t2.micro ap-south-1a running Initializin
win-web-server-auto-scale i-0c0fefd5b6256de59 t2.micro ap-south-1b running 2/2 checks
win-web-server-auto-scale i-0810d21efb960bee6 t2.micro ap-south-1a running 2/2 checks
win-web-server-auto-scale i-07f00cd38b8972b83 t2.micro ap-south-1b running 2/2 checks
win-web-server-auto-scale i-05824f2b0fc924c15 t2.micro ap-south-1a running 2/2 checks
win-web-server-auto-scale i-0033e067995cc32a6 t2.micro ap-south-1b running 2/2 checks
Select an instance above
Feedback
English (US)
© 2008 - 2020, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use

## [LAB] Test Auto Scaling Lab: CPU Utilization Scale In

- When you step the Script, the instance CPU utilization goes below 80% (threshold), that changes the desired instances value to go down.
- This will eventually terminate extra instances

## Route 53

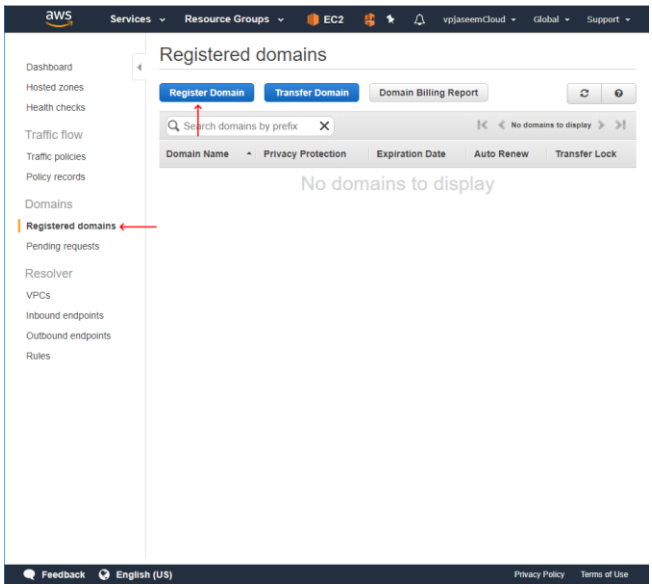
- Highly available and scalable DNS web service
- Route 53 registers domain name, acts as a registrar server
- Points Domain name to AWS Service (ELB, EC2)
- Health checks of resources
- It works on edge locations
- Alias: All AWS Services DNS names (ELD, EC2, S3, etc.) mapped to friendly Alias
- CNAME: One DNS name maps to another DNS name

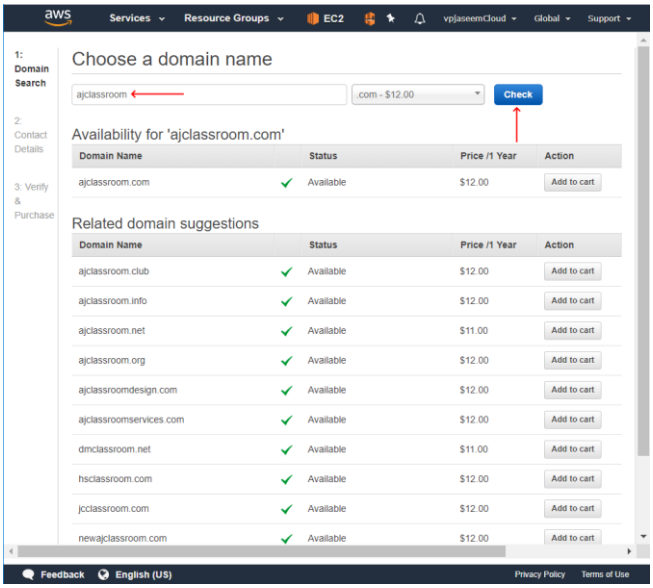
### Routing Policies

- **Simple Routing Policy**: One to One mapping, one domain name map to one resource
- **Failover Routing Policy**: Route the traffic to healthy instance, if the instance is unhealth, route traffic to redundant instance.
- **Geolocation Routing Policy**: Routing based on geographic location of the user.
- **Geoproximity Routing Policy**: Routing based on geographic location of the resource, optimally shift traffic from one resource to another resource in different location.
- **Latency Based Routing**: When website is deployed in multiple regions, this routing is used.
- **Multivalue Answer Routing**: 8 servers get the request in random way.
- **Weighted Routing Policy**: Control the traffic routed to each resource, used for load balancing and testing (20% to some new instances and 80% to old instances)

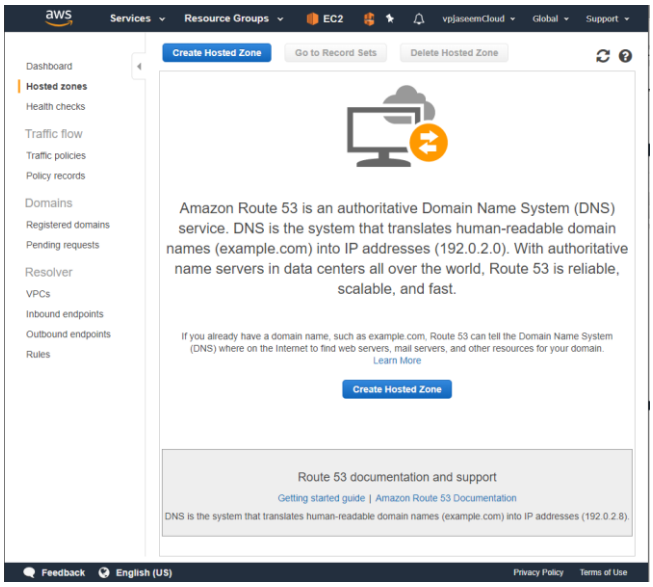
# [LAB] Route 53

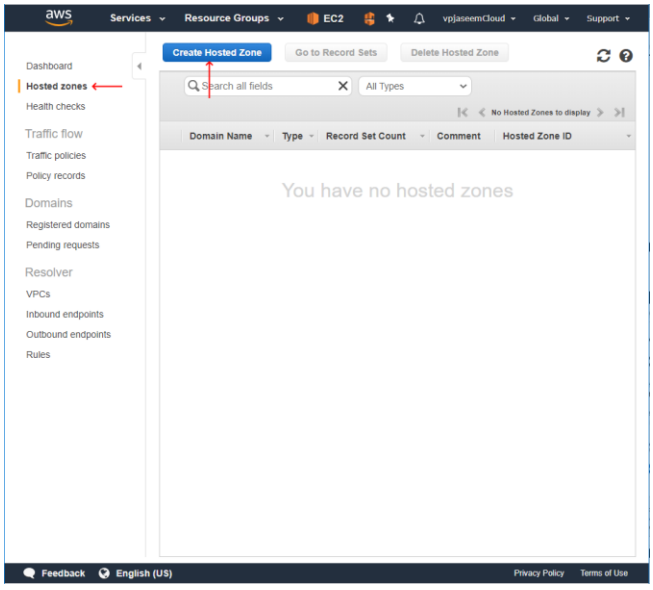
## Domain Registration

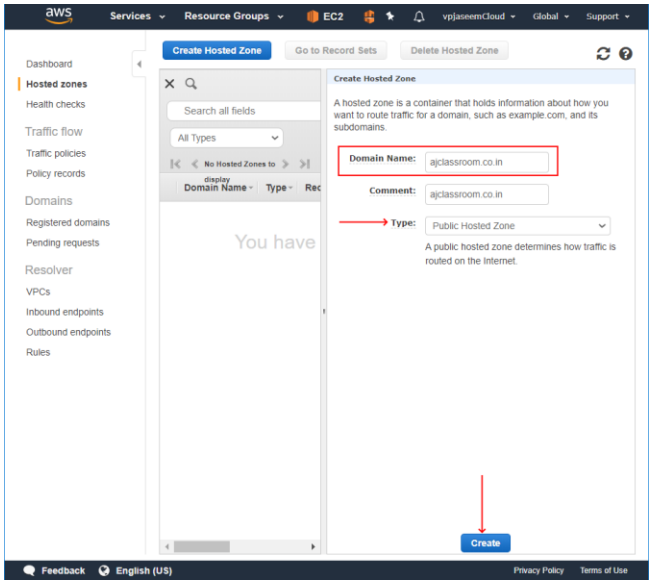


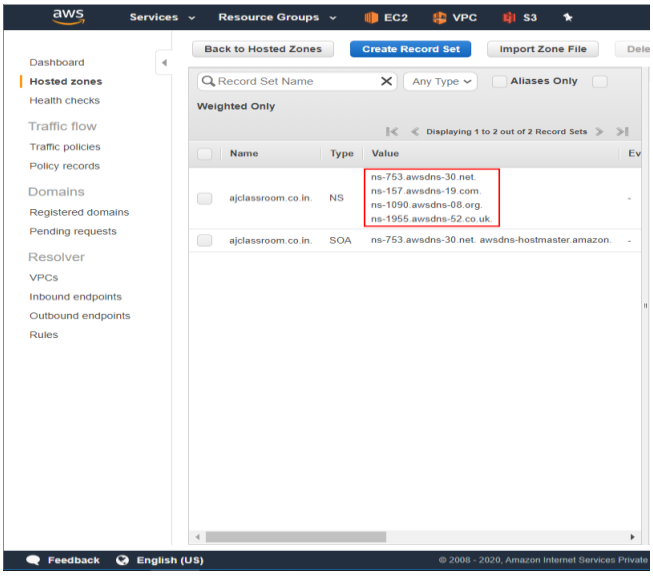



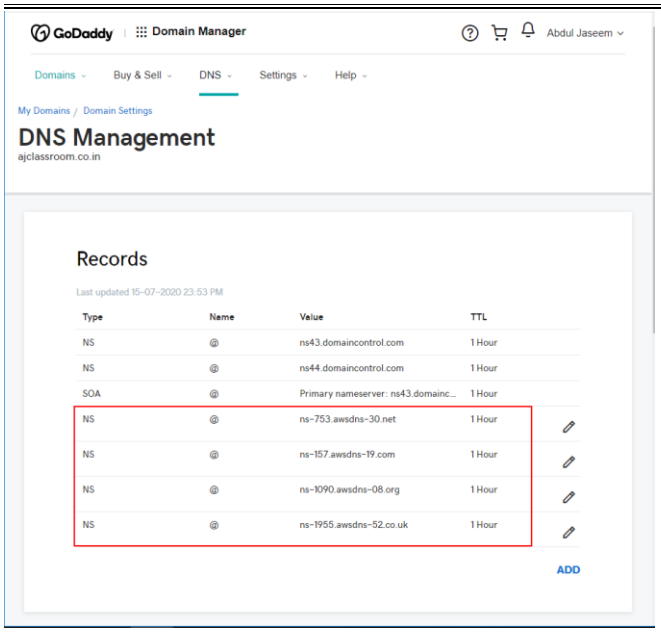

## Hosted Zone

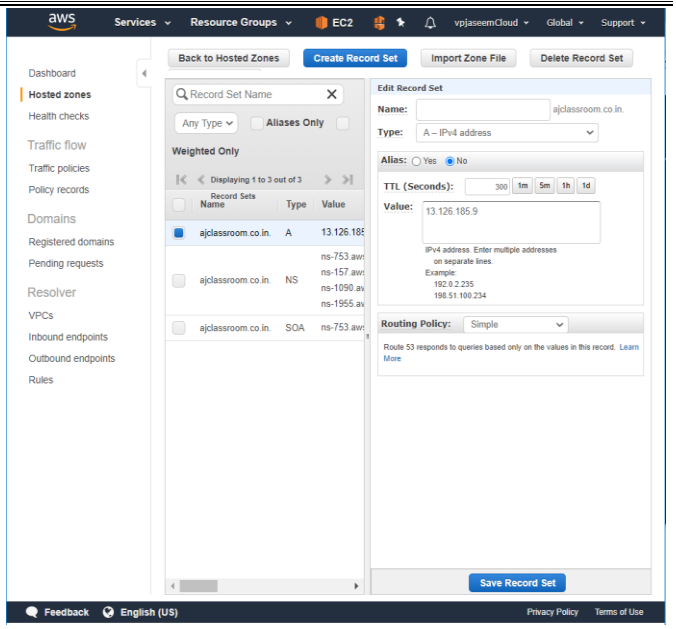


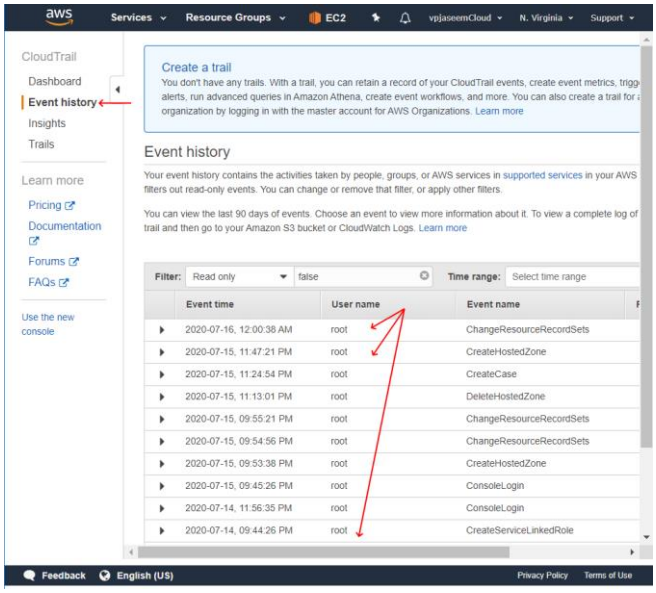

GoDaddy
Domain Manager
Abdul Jaseem
Domains
Buy & Sell
DNS
Settings
Help
My Domains / Domain Settings
DNS Management
ajclassroom.co.in
Records
Last updated 15-07-2020 23:53 PM
Type
Name
Value
TTL
NS @ ns43.domaincontrol.com 1 Hour
NS @ ns44.domaincontrol.com 1 Hour
SOA @ Primary nameserver: ns43.domainc... 1 Hour
NS @ ns-753.awsdns-30.net 1 Hour
NS @ ns-157.awsdns-19.com 1 Hour
NS @ ns-1090.awsdns-08.org 1 Hour
NS @ ns-1955.awsdns-52.co.uk 1 Hour
ADD

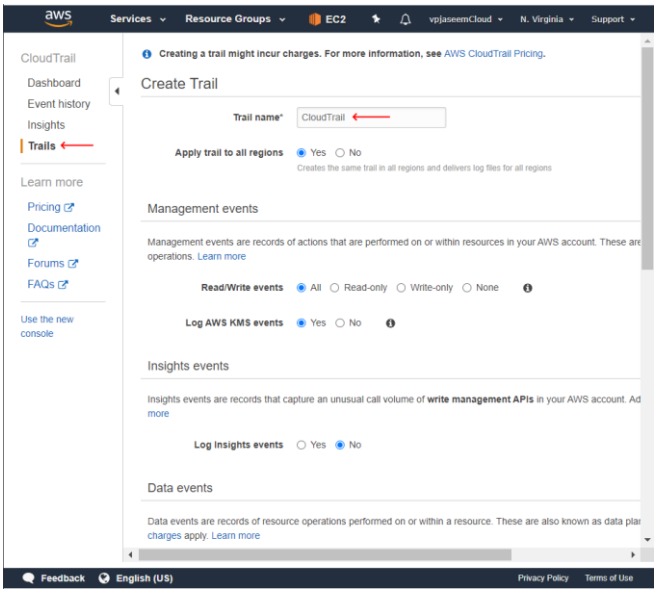

aws
Services
Resource Groups
EC2
vpjaseemCloud
Global
Support
Dashboard
Hosted zones
Health checks
Traffic flow
Traffic policies
Policy records
Domains
Registered domains
Pending requests
Resolver
VPCs
Inbound endpoints
Outbound endpoints
Rules
Back to Hosted Zones
Create Record Set
Import Zone File
Delete Record Set
Record Set Name
Any Type
Aliases Only
Weighted Only
Displaying 1 to 3 out of 3
Record Sets
Name
Type
Value
ajclassroom.co.in. A 13.126.185
ajclassroom.co.in. NS ns-753.aws ns-157.aws ns-1090.a ns-1955.a
ajclassroom.co.in. SOA ns-753.aws
Edit Record Set
Name:
ajclassroom.co.in.
Type:
A – IPv4 address
Alias: Yes No
TTL (Seconds):
300
1m
5m
1h
1d
Value:
13.126.185.9
IPv4 address. Enter multiple addresses on separate lines.
Example:
192.0.2.235
198.51.100.234
Routing Policy:
Simple
Route 53 responds to queries based only on the values in this record. Learn More
Save Record Set
Feedback
English (US)
Privacy Policy
Terms of Use

## AWS CloudTrail

- With CloudTrail, you can view events for your AWS account. Create a trail to retain a record of these events. With a trail, you can also create event metrics, trigger alerts, and create event workflows.
- User Activity reordered in CloudTrail, Available in 90 days (event viewer logs, audit logs)
- If you want to retain the logs after 90 days, you can create a Trail to send the logs to S3

# [LAB] Cloud Trail



## AWS Lambda

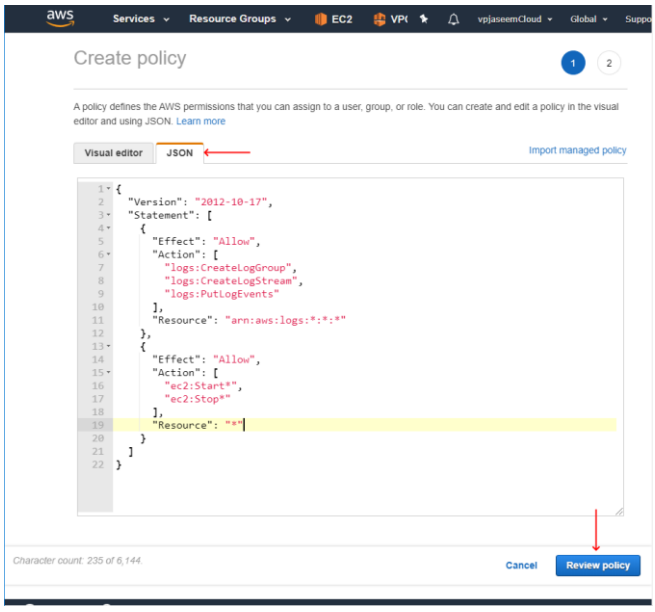
- Allows to execute our code without server (also called serverless computing). We don't need to manage any servers (EC2s) to run the code. AWS will provide managed environment to run the codes.
- Supports code C#, Java, Python, Node.js and Go languages
- 128MB Memory to 3008 MB memory, we can choose while creating Lambda
- Functions can run between 100ms to 5 minutes
- We should assign IAM role to Lambda so that it can access other AWS services
- We can run Lambda function manually or scheduled or based on some events
- Commonly used to run standalone scripts

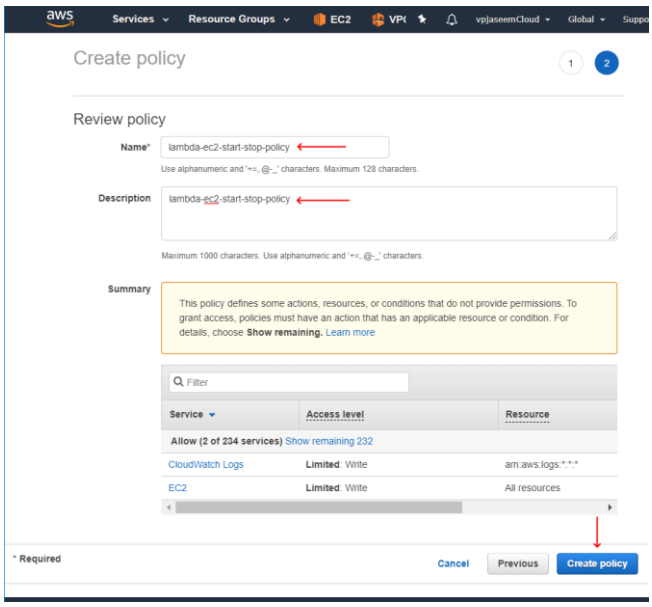
## [LAB] AWS Lambda Example

- Start and stop EC2 instances with Lambda, Python and CloudWatch to schedule
- Aim: Run EC2 instances only Mon-Fri 9Am to 6PM

### Step 1: Create a Policy with below JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "logs:CreateLogGroup",
        "logs:CreateLogStream",
        "logs:PutLogEvents"
      ],
      "Resource": "arn:aws:logs:*:*:*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "ec2:Start*",
        "ec2:Stop*"
      ],
      "Resource": "*"
    }
  ]
}
```

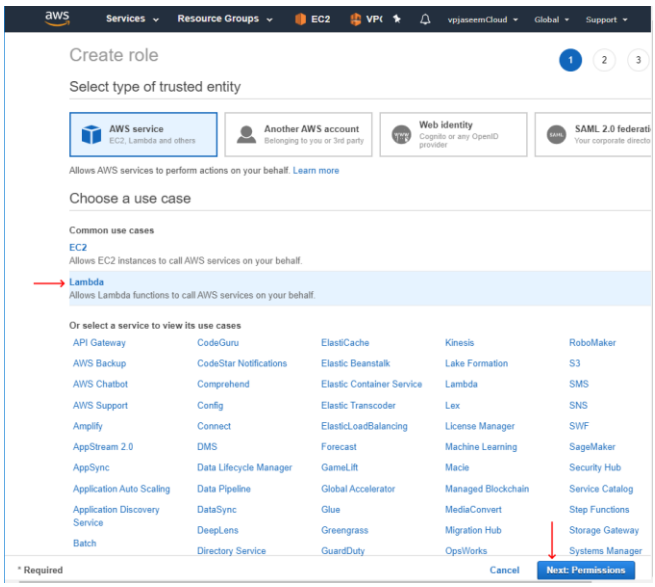


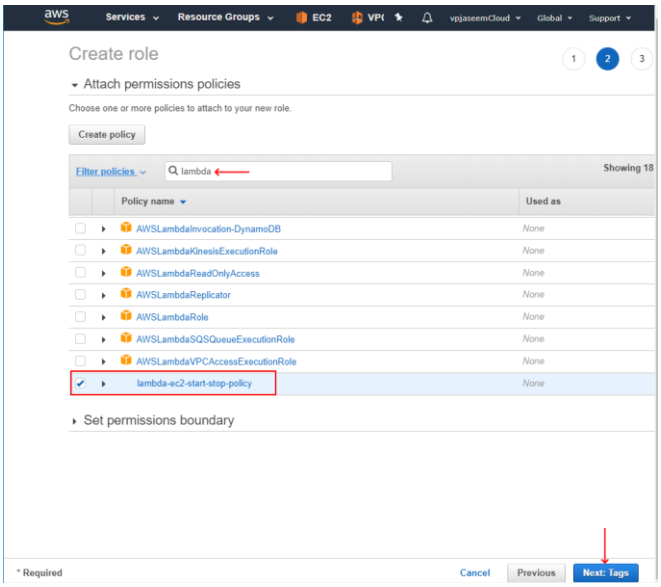



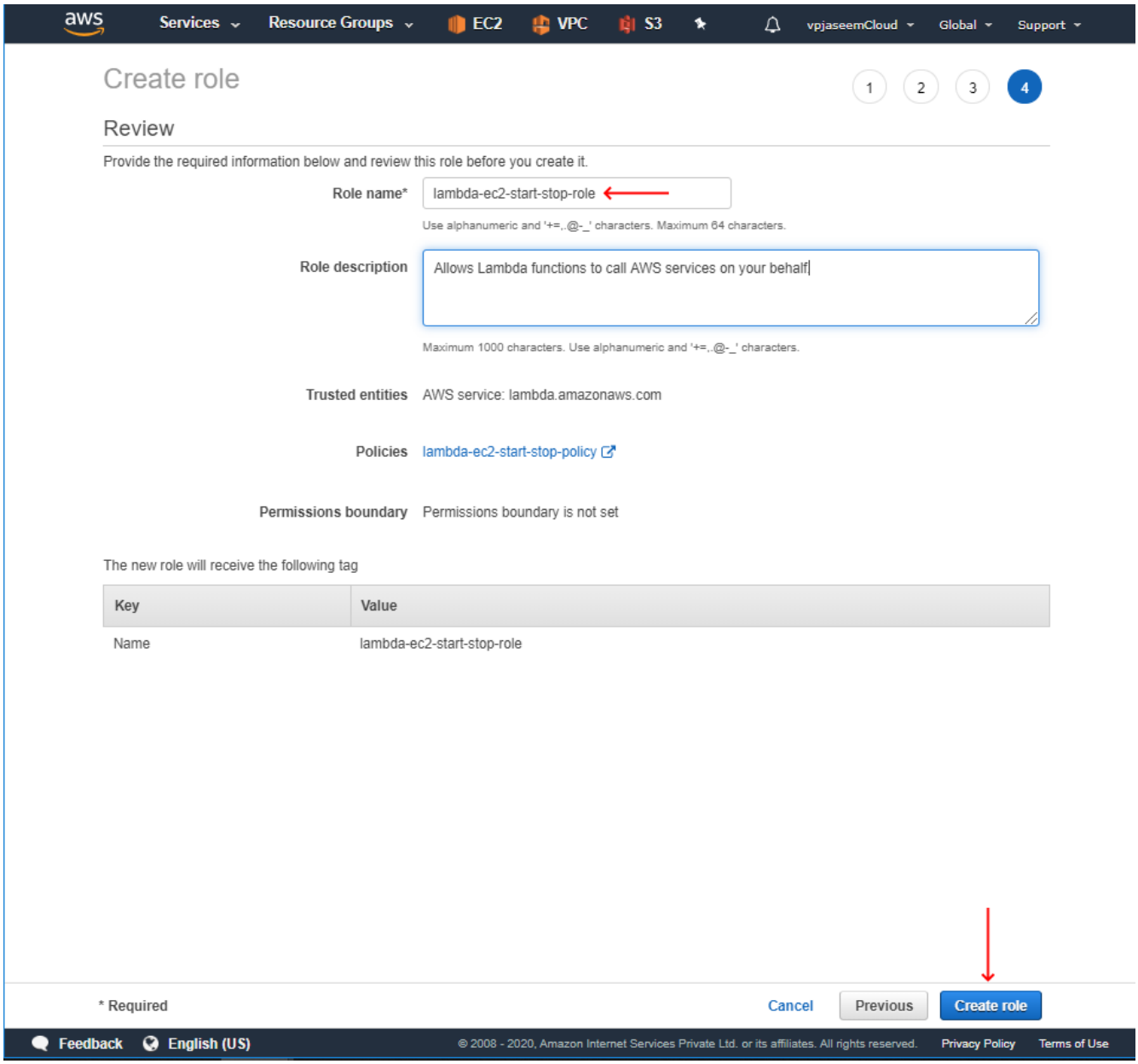

- This will allow EC2 Start and Stop and Upload logs to CloudWatch
- Next step, we create an IAM Role and attach this policy

## Step 2: IAM Role for Lambda

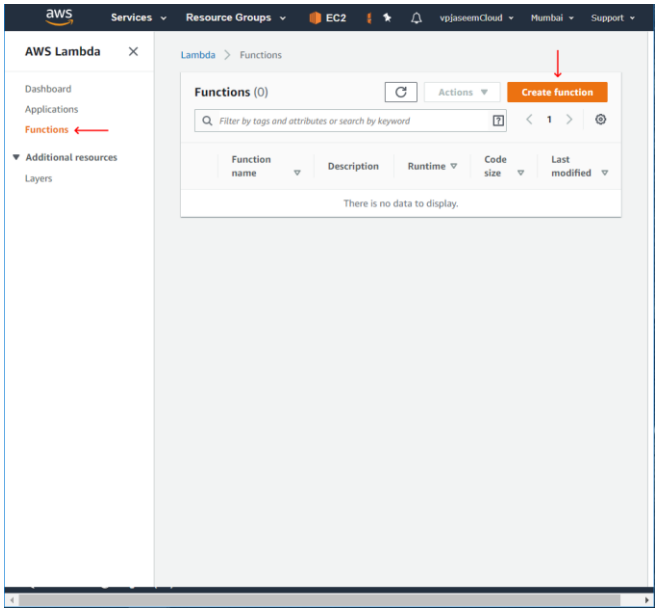
- When attaching a permissions policy, search for and choose the IAM policy that you created.

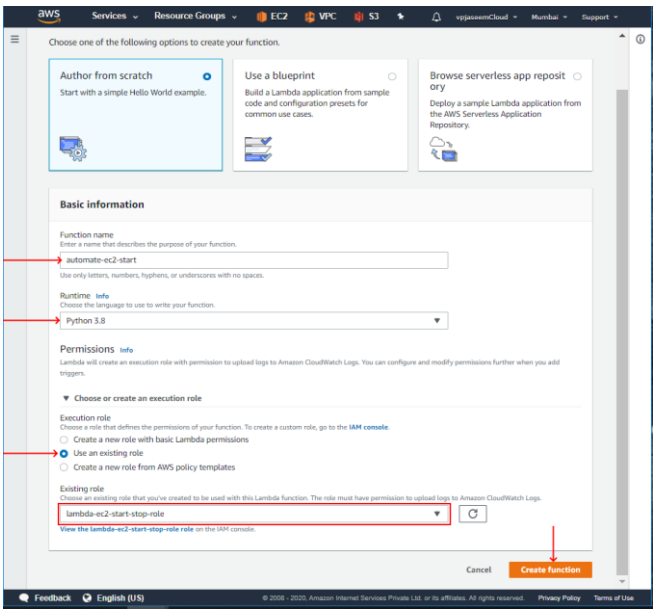


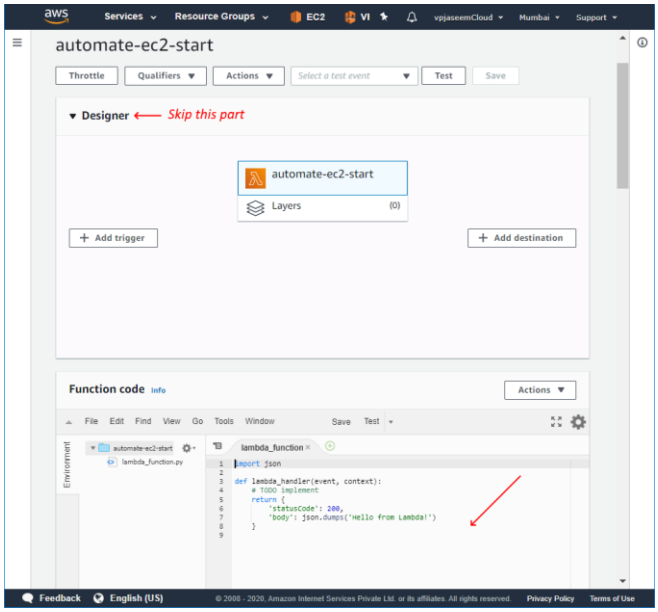



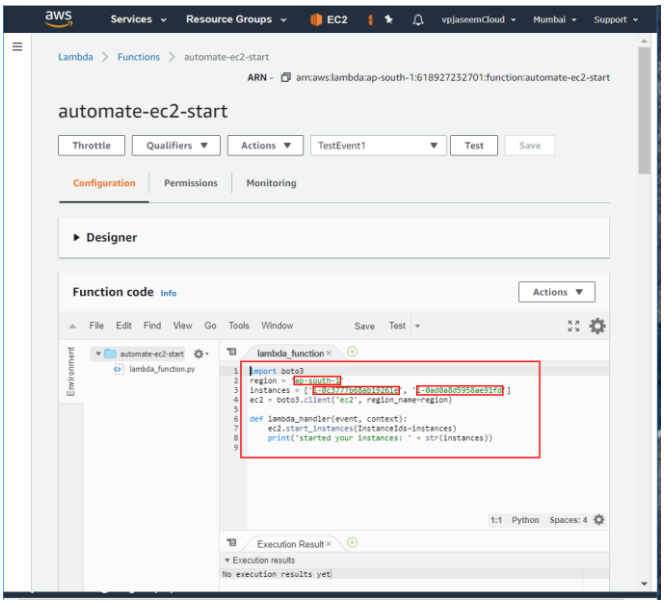



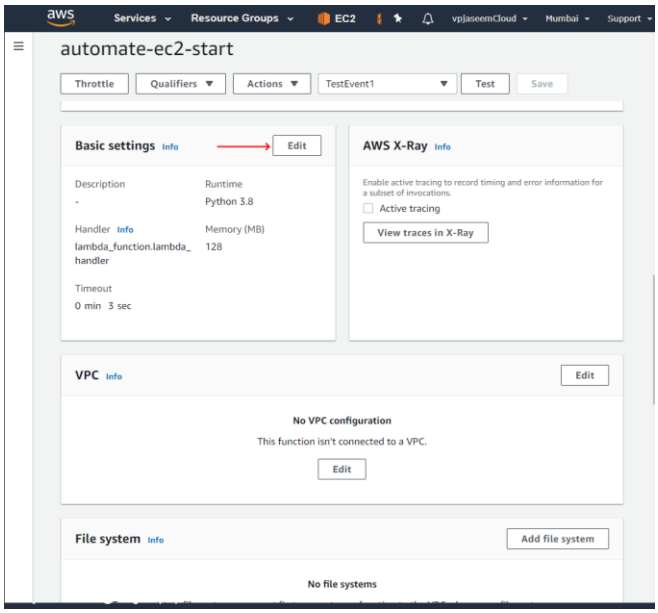
## Step 3: Create Lambda Function

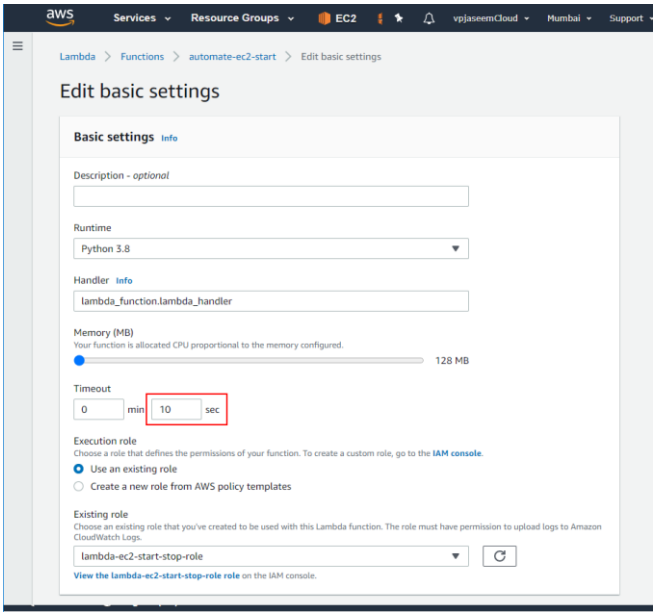


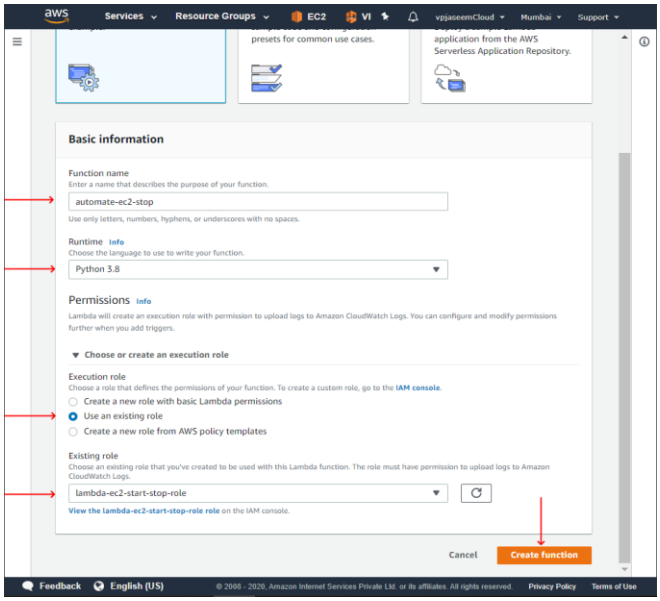



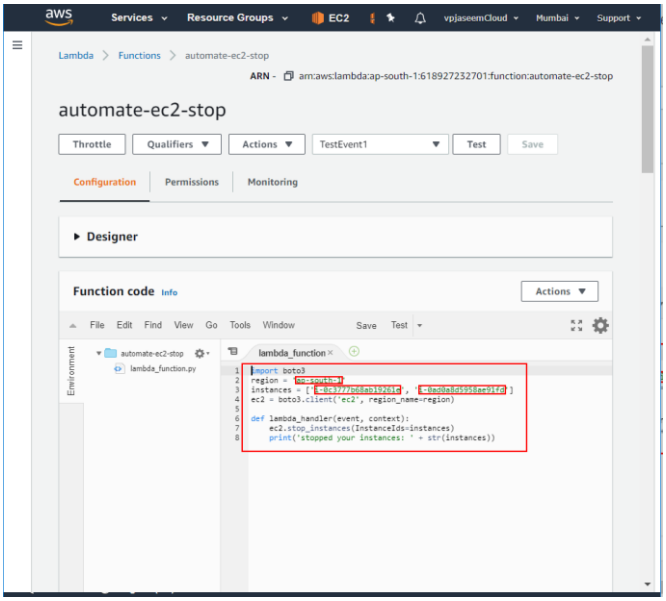



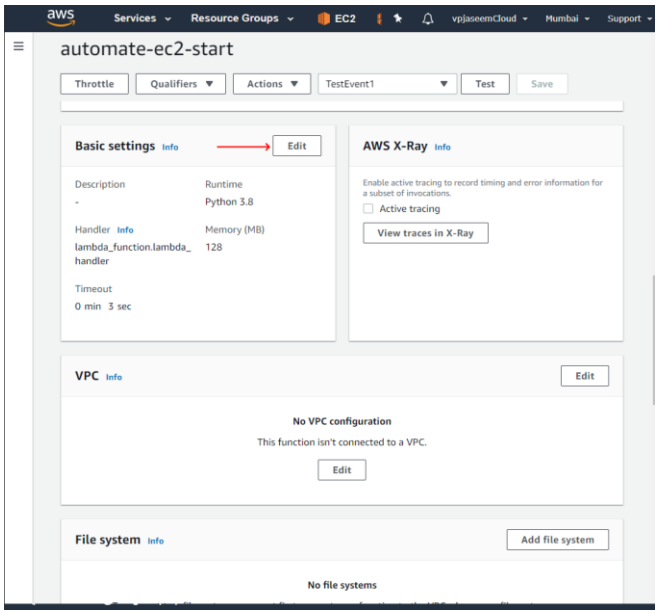
```
import boto3
region = 'us-west-1'
instances = ['i-12345cb6de4f78g9h', 'i-08ce9b2d7eccf6d26']
ec2 = boto3.client('ec2', region_name=region)

def lambda_handler(event, context):
    ec2.start_instances(InstanceIds=instances)
    print('started your instances: ' + str(instances))
```

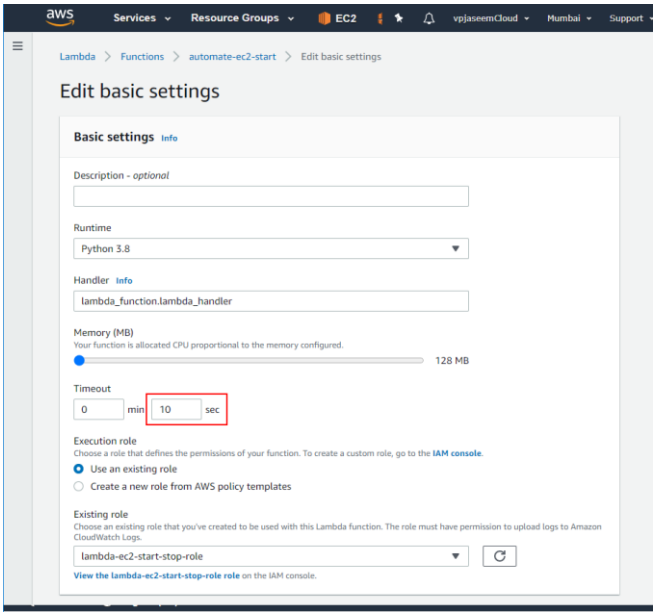
- Create another Lambda Function to stop EC2 instances

```
import boto3
region = 'us-west-1'
instances = ['i-12345cb6de4f78g9h', 'i-08ce9b2d7eccf6d26']
ec2 = boto3.client('ec2', region_name=region)

def lambda_handler(event, context):
    ec2.stop_instances(InstanceIds=instances)
    print('stopped your instances: ' + str(instances))
```

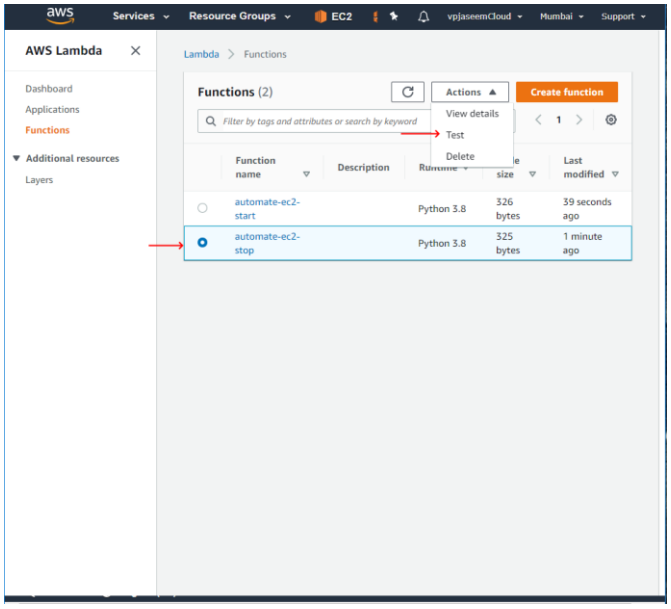


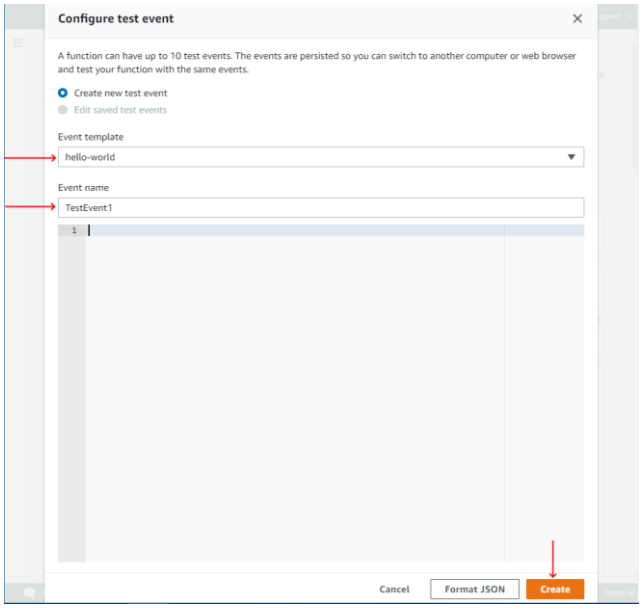



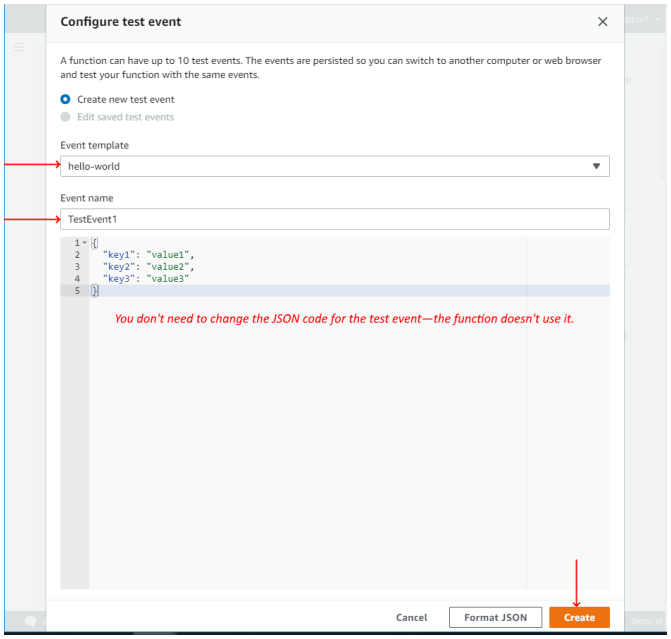



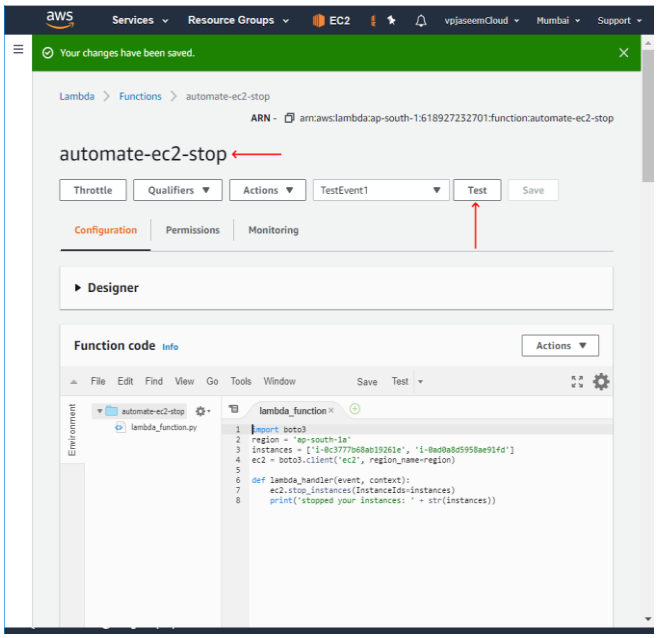



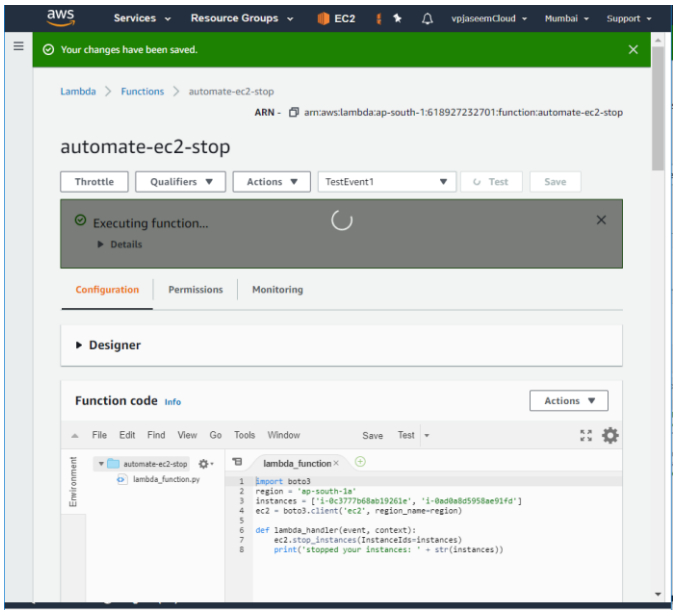
## Step 3: Test Lambda functions

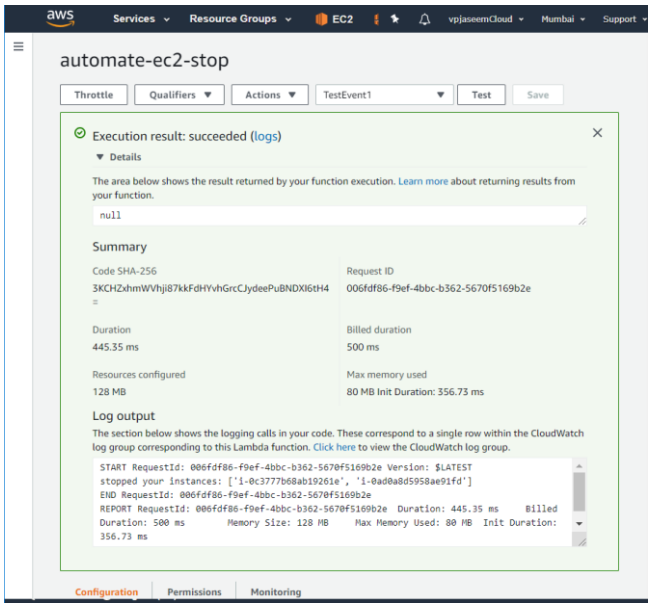


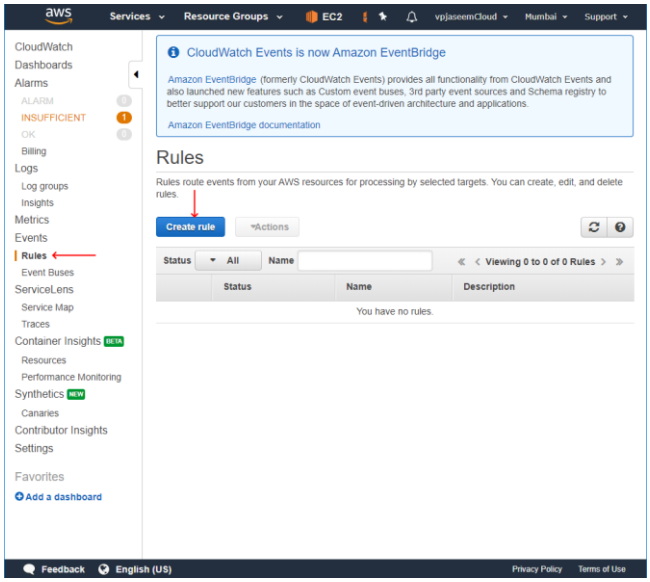



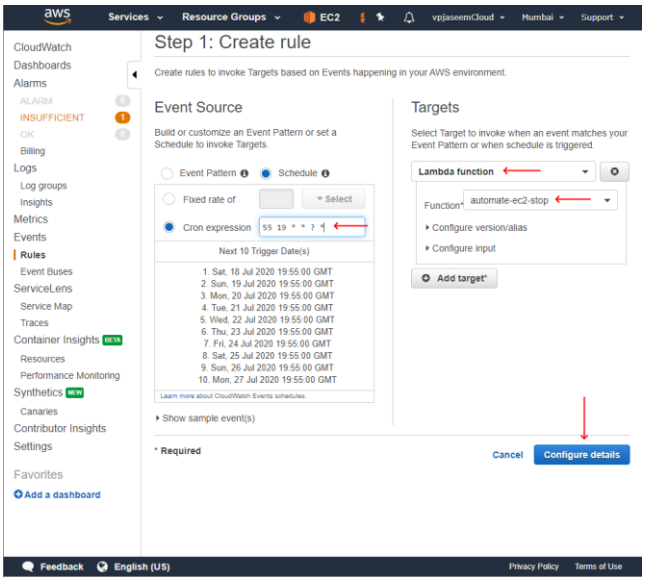



## Step 4: Create CloudWatch Rule to Trigger Lambda Functions



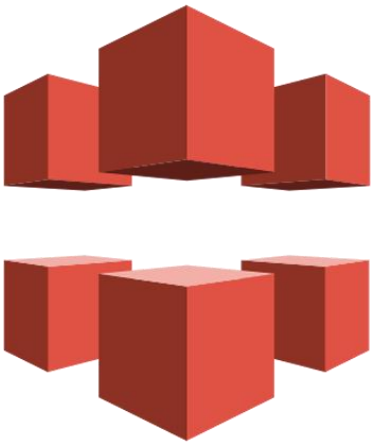




AWS Docs: [Cron Schedule Strings]

- Create another Rule for starting the instances

Read More about the LAB: [AWS Docs]

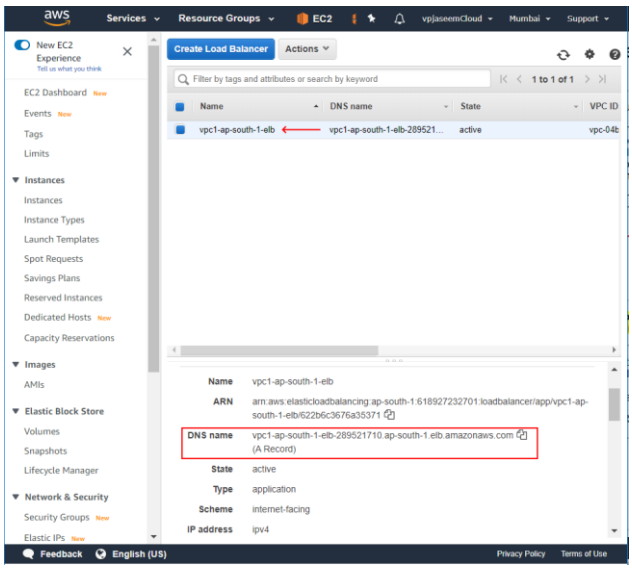
## AWS CloudFront

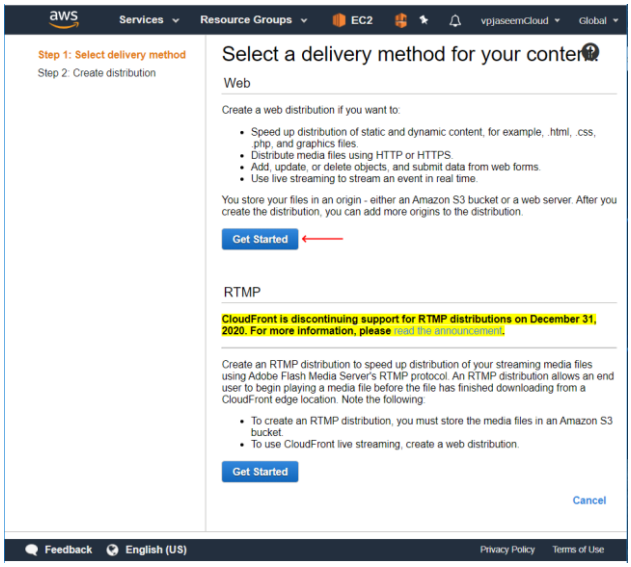


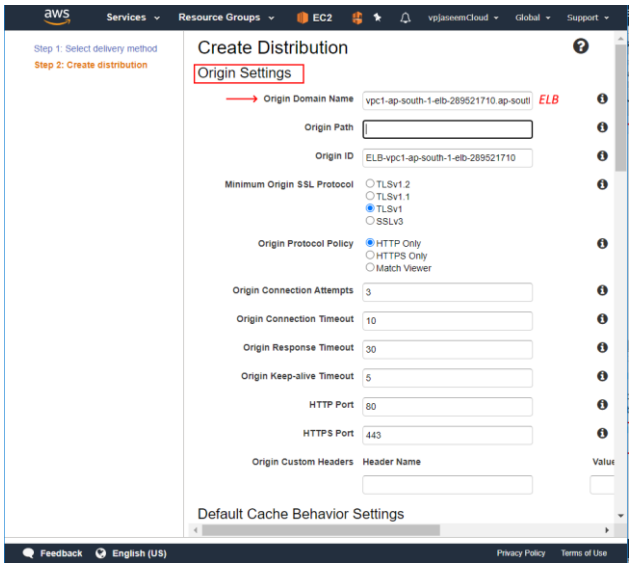
- While we access content from far region, it may take some time
- CloudFont will cache the data in nearest location and made available to users faster
- Speeds up the content distribution
- Works on AWS Edge Location
- It's a CDN (Content Delivery Network)
- Provide HTTPS by default

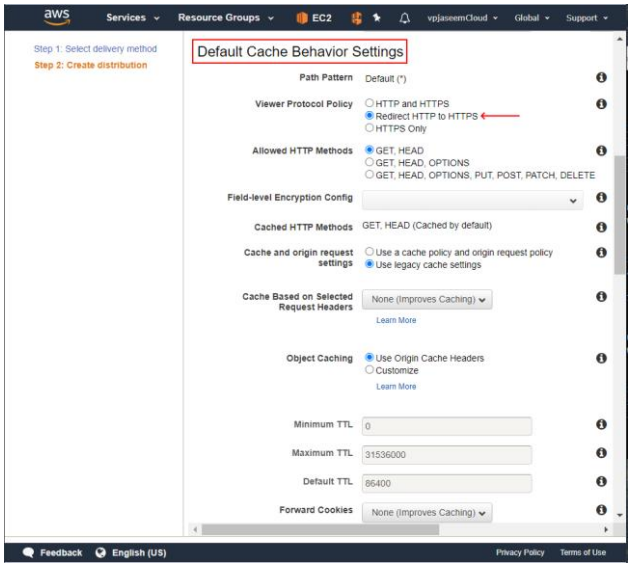
## [LAB] AWS CloudFront Distribution for a load balanced Web Site

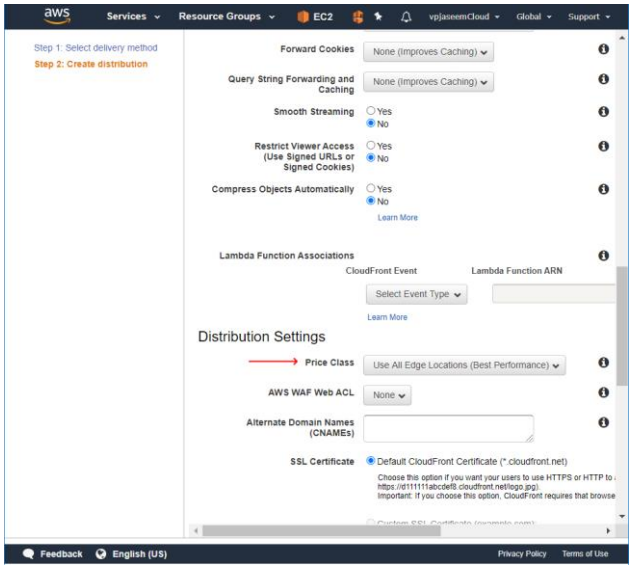
- Host a PHP website in 2 Linux EC2 instance, Connect to Load balancer to these instances via target group.
- Make sure your Web app is accessible via ELB DNS name.

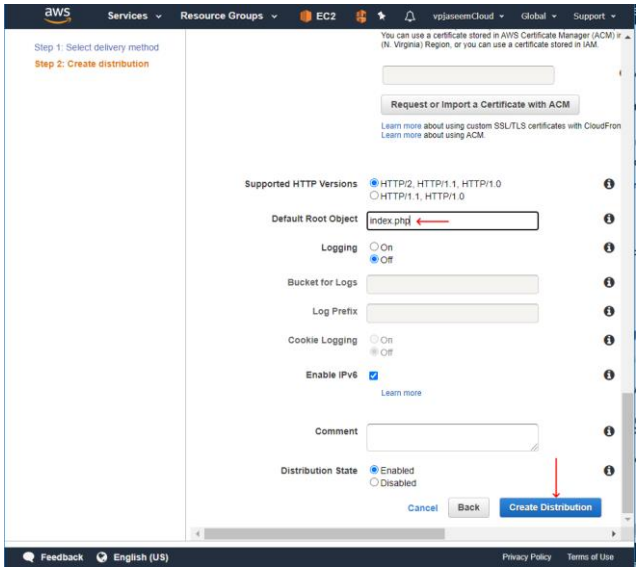


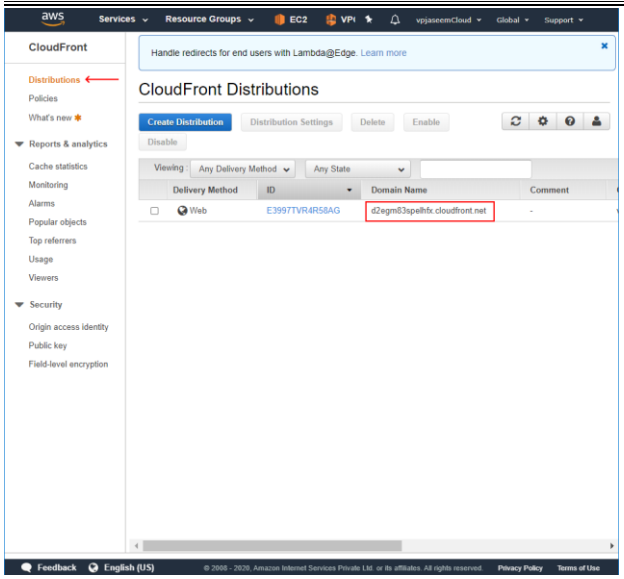



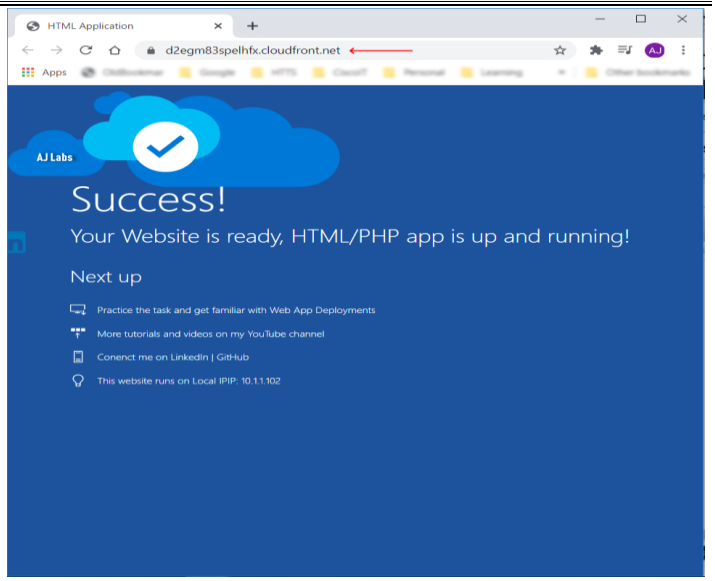






Note: You can map the CloudFront URL as a CNAME to existing custom domain

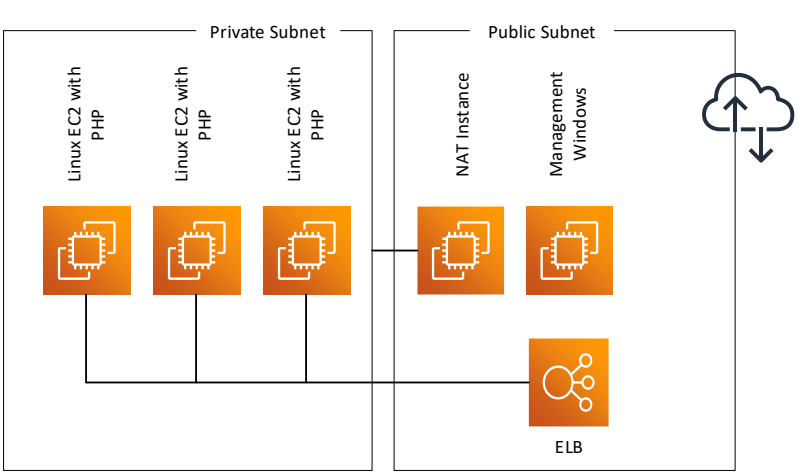
## AWS Cloud Formation

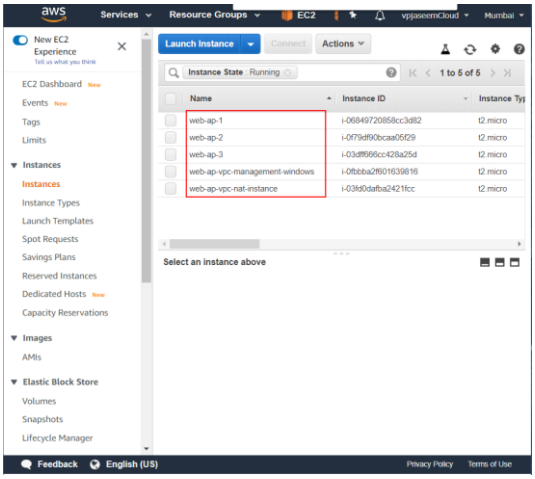
- CloudFormation gives the ability to create the complete AWS environment using a template
- Describe the complete environment in JSON template
- Template: Collection of resources, attributes
- Stack: Once you feed this to AWS Cloud Formation, the resulting structure is called Stack. Once we delete the stack, the entire resources get deleted.
- For example, if you want to setup a RHCSA lab, at least you need 3 Linux machines, you planned to deploy that on AWS with proper VPC, Security Group, NACLs, etc. After all the configuration, you can create template out of this lab and share it with your friend so that they can deploy the same. It is a JSON export of existing configuration.
- We can create new template from Scratch using Designer or Export existing infrastructure as a Template.

AWS Reference Video: CloudFormation

## [LAB] AWS Could Formation - Exporting Existing Infa to Template (CloudFormer)

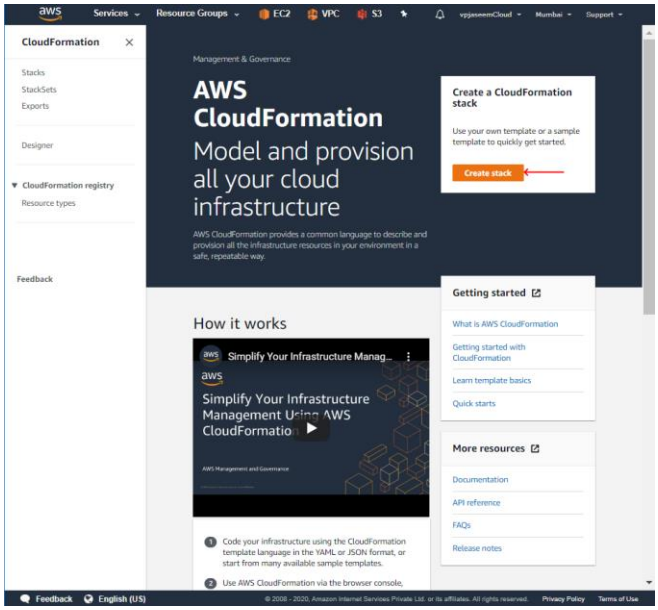
- Below is a small test infrastructure consisting of 3 EC2 Linux instances with PHP Running, 1 Load Balancer, NAT Instance, Management Windows.
- I have already deployed this in a new VPC

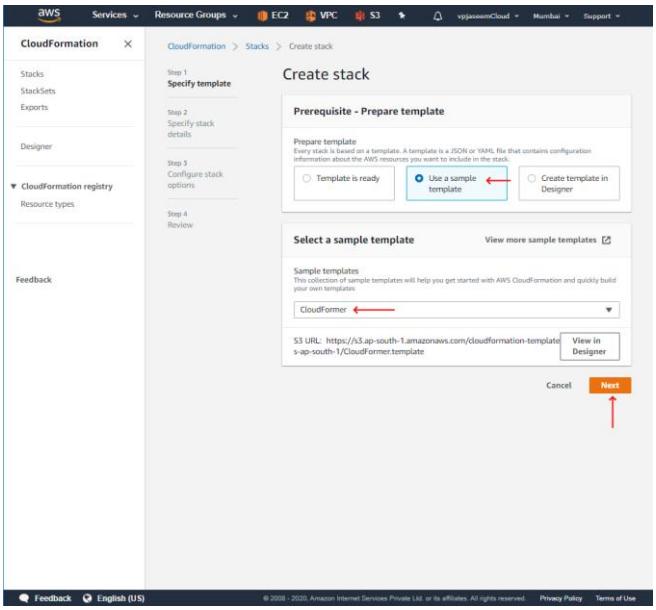


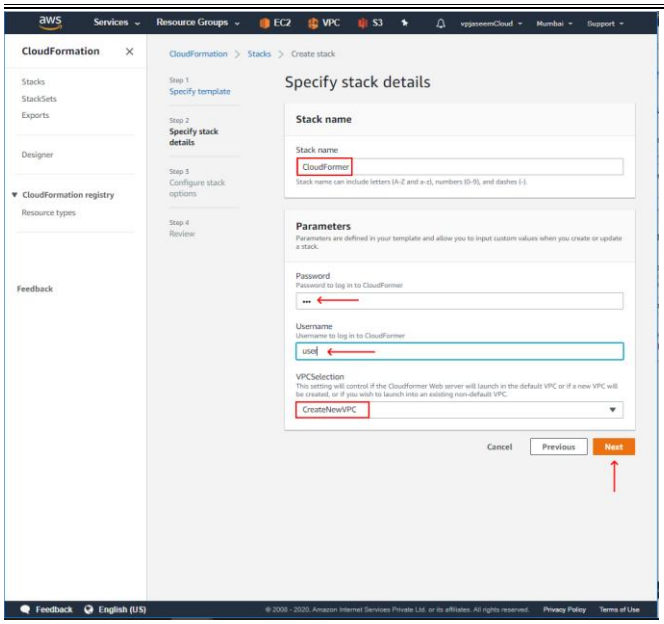



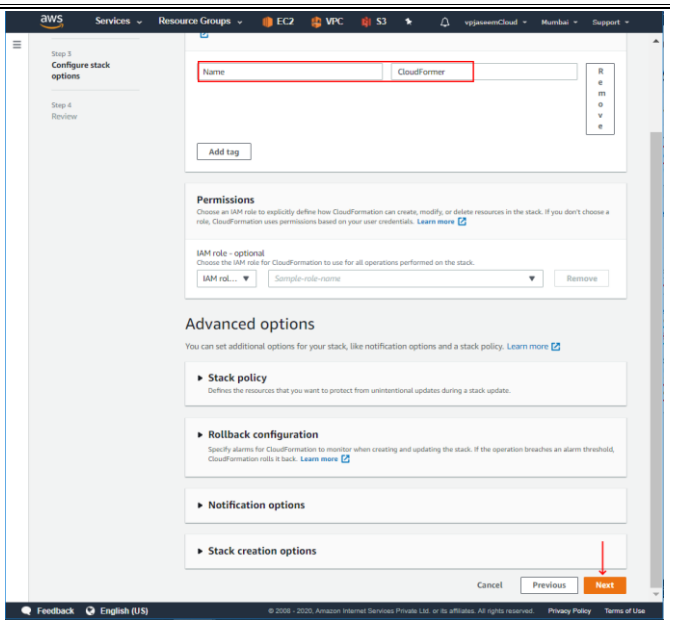
| | VPC NAME | VPC CIDR | | SUBNET NAME | SUBNET CIDR |
|---|---|---|---|---|---|
| FloudFormation | web-ap-vpc | 10.57.0.0/16 | Public | web-ap-vpc-public-subnet-ap-south-1a | 10.57.1.0/24 |
| | | | | web-ap-vpc-public-subnet-ap-south-1b | 10.57.2.0/24 |
| | | | Private | web-ap-vpc-private-subnet-ap-south-1a | 10.57.11.0/24 |

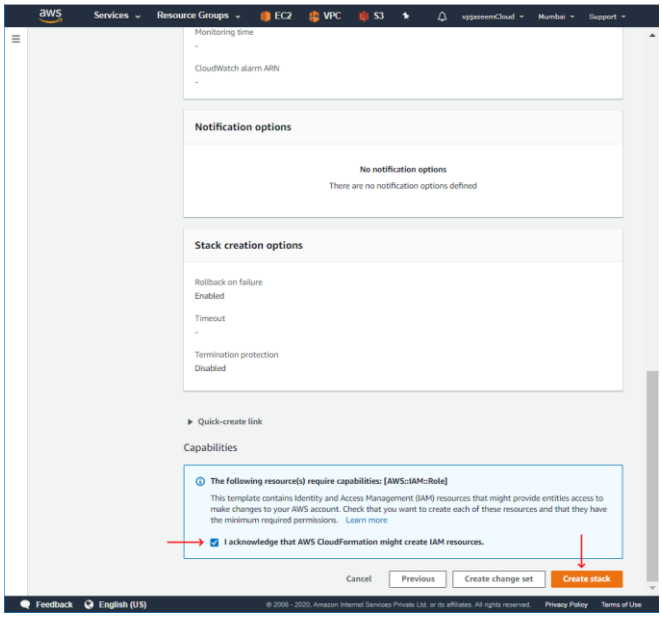
- Now I will create a template of the above setup using CloudFormer. Basically a JSON export of the entire setup.
- As part of the process, AWS will launch a CloudFormer EC2 instance with its own VPC and Subnets.
- By accessing that instance over internet, we can generate our template.

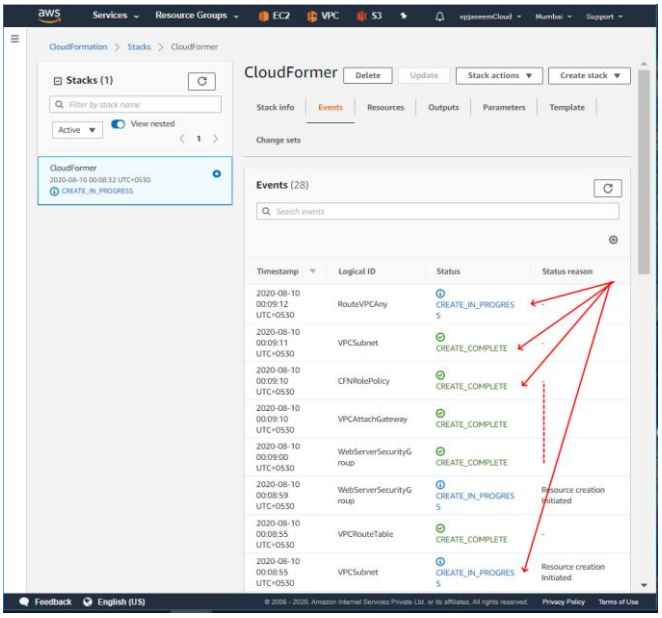


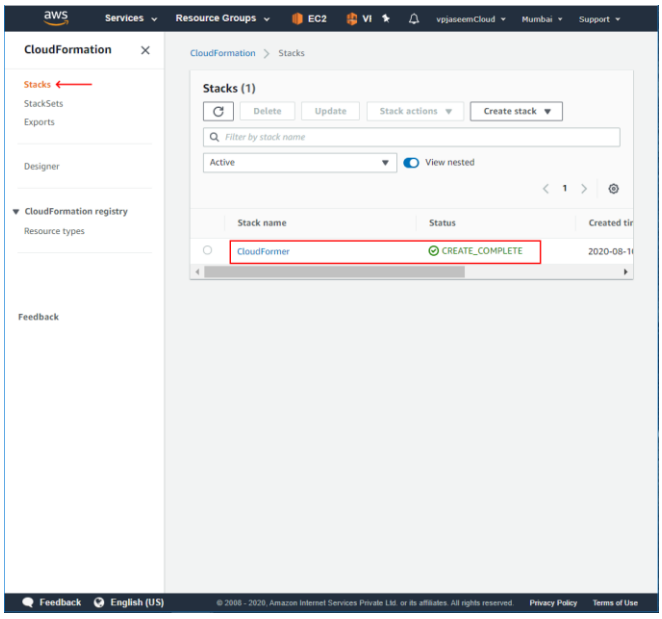



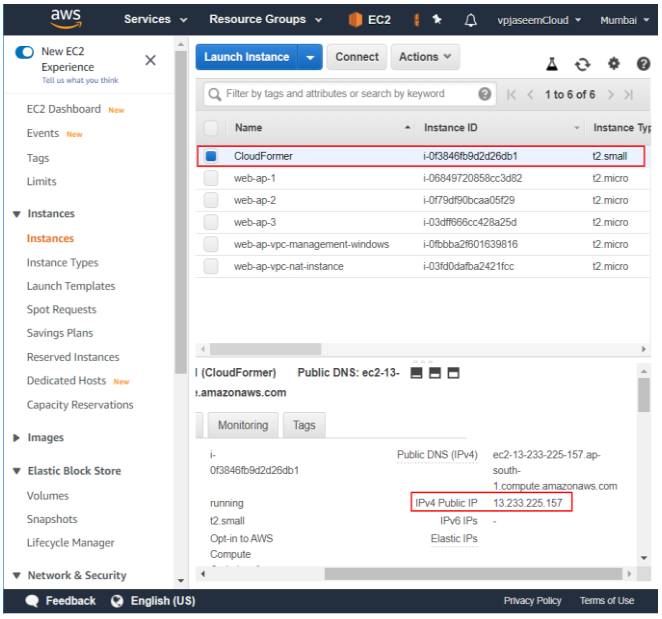

CloudFormation
Specify stack details
Stack name
CloudFormer
Parameters
Password
Username
user
VPCSelection
CreateNewVPC
Cancel
Previous
Next

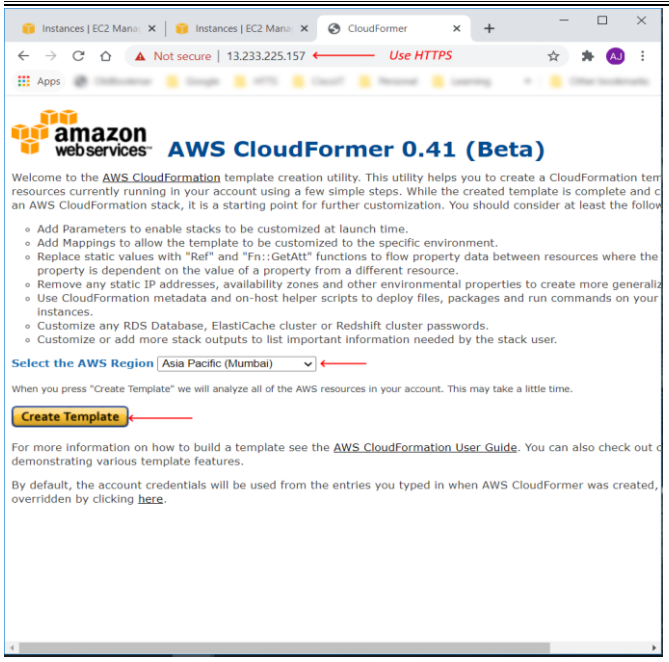

Name
CloudFormer
Add tag
Permissions
Advanced options
Stack policy
Rollback configuration
Notification options
Stack creation options
Cancel
Previous
Next

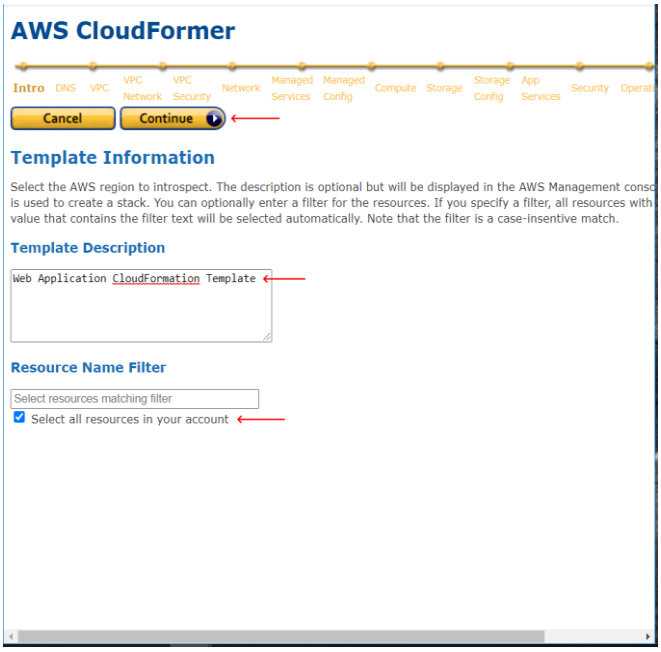

Notification options
No notification options
There are no notification options defined
Stack creation options
Rollback on failure
Enabled
Timeout
Termination protection
Disabled
Capabilities
The following resource(s) require capabilities: [AWS::IAM::Role]
I acknowledge that AWS CloudFormation might create IAM resources.
Cancel
Previous
Create change set
Create stack


CloudFormer
Stacks (1)
Delete
Update
Stack actions
Create stack
Stack info
Events
Resources
Outputs
Parameters
Template
Change sets
Events (28)
Timestamp
Logical ID
Status
Status reason
CREATE_IN_PROGRESS
CREATE_COMPLETE
Resource creation Initiated

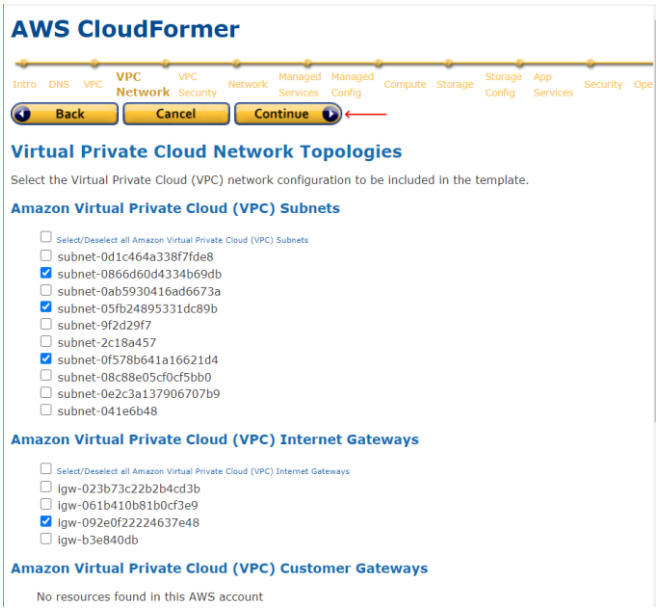

CloudFormation
Stacks
Stacks (1)
Delete
Update
Stack actions
Create stack
Stack name
Status
CloudFormer
CREATE_COMPLETE

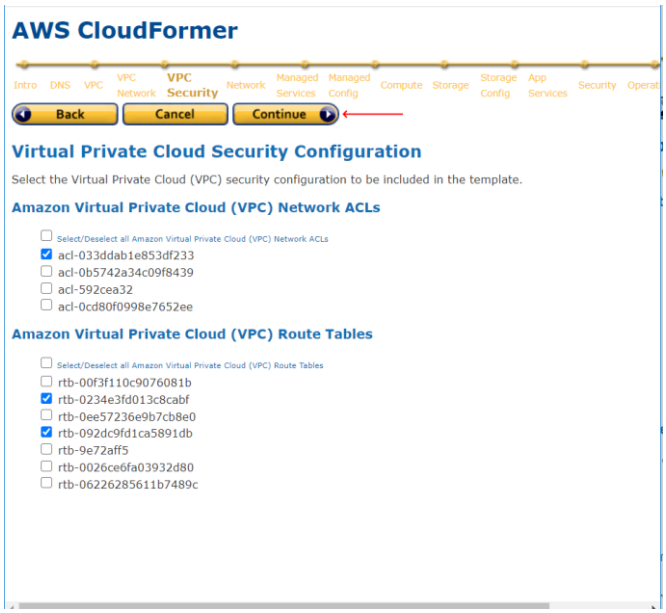

Launch instance
Connect
Actions
Name
Instance ID
CloudFormer
i-0f3846fb9d2d26db1
t2.small
web-ap-1
web-ap-2
web-ap-3
web-ap-vpc-management-windows
web-ap-vpc-nat-instance
Monitoring
Tags
Public DNS (IPv4)
IPv4 Public IP
13.233.225.157
running
t2.small

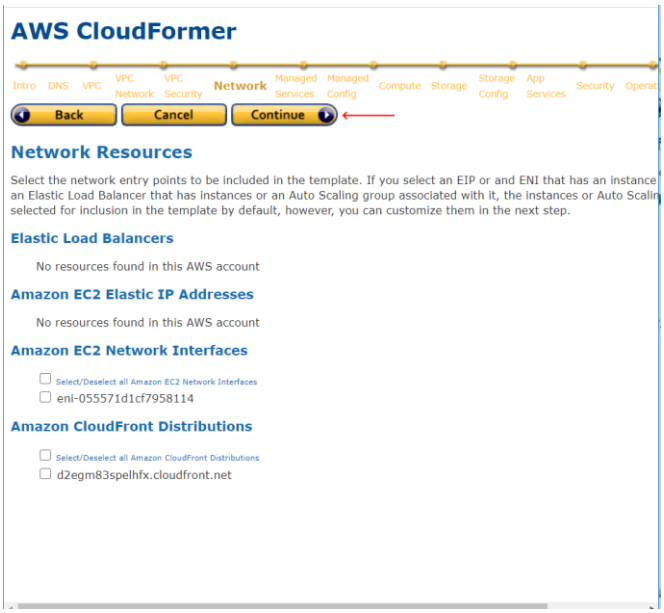

Instances | EC2 Mana
Instances | EC2 Mana
CloudFormer
Not secure | 13.233.225.157
Use HTTPS
Apps
amazon web services
AWS CloudFormer 0.41 (Beta)
Welcome to the AWS CloudFormation template creation utility. This utility helps you to create a CloudFormation tem
resources currently running in your account using a few simple steps. While the created template is complete and c
an AWS CloudFormation stack, it is a starting point for further customization. You should consider at least the follow
Add Parameters to enable stacks to be customized at launch time.
Add Mappings to allow the template to be customized to the specific environment.
Replace static values with "Ref" and "Fn::GetAtt" functions to flow property data between resources where the
property is dependent on the value of a property from a different resource.
Remove any static IP addresses, availability zones and other environmental properties to create more generaliz
Use CloudFormation metadata and on-host helper scripts to deploy files, packages and run commands on your
instances.
Customize any RDS Database, ElastiCache cluster or Redshift cluster passwords.
Customize or add more stack outputs to list important information needed by the stack user.
Select the AWS Region
Asia Pacific (Mumbai)
When you press "Create Template" we will analyze all of the AWS resources in your account. This may take a little time.
Create Template
For more information on how to build a template see the AWS CloudFormation User Guide. You can also check out
demonstrating various template features.
By default, the account credentials will be used from the entries you typed in when AWS CloudFormer was created,
overridden by clicking here.


AWS CloudFormer
Intro DNS VPC VPC Network VPC Security Network Managed Services Managed Config Compute Storage Storage Config App Services Security Operati
Cancel
Continue
Template Information
Select the AWS region to introspect. The description is optional but will be displayed in the AWS Management consc
is used to create a stack. You can optionally enter a filter for the resources. If you specify a filter, all resources with
value that contains the filter text will be selected automatically. Note that the filter is a case-insentive match.
Template Description
Web Application CloudFormation Template
Resource Name Filter
Select resources matching filter
Select all resources in your account

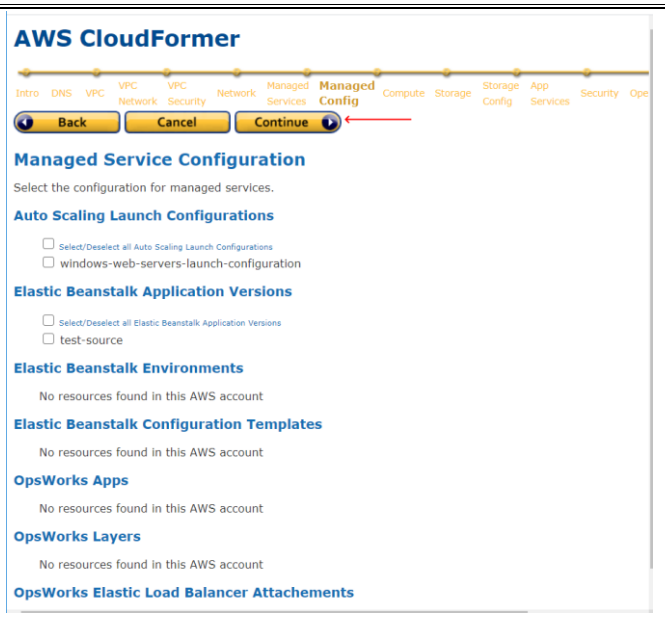

AWS CloudFormer
Intro DNS VPC VPC Network VPC Security Network Managed Services Managed Config Compute Storage Storage Config App Services Security Operati
Back
Cancel
Continue
Virtual Private Clouds
Select the Virtual Private Clouds (VPCs) to be included in the template. If you select a VPC, all VPC network and sec
selected for inclusion in the template by default, however, you can customize them in the next steps.
Amazon Virtual Private Clouds (VPCs)
Select/Deselect all Amazon Virtual Private Clouds (VPCs)
vpc-0a54928c8327672a8
vpc-033c496ef1786eff5
vpc-04bdc71ef654d09fa
vpc-8fd2cde7


AWS CloudFormer
Intro DNS VPC VPC Network VPC Security Network Managed Services Managed Config Compute Storage Storage Config App Services Security Ope
Back
Cancel
Continue
Virtual Private Cloud Network Topologies
Select the Virtual Private Cloud (VPC) network configuration to be included in the template.
Amazon Virtual Private Cloud (VPC) Subnets
Select/Deselect all Amazon Virtual Private Cloud (VPC) Subnets
subnet-0d1c464a338f7fde8
subnet-0866d60d4334b69db
subnet-0ab5930416ad6673a
subnet-05fb24895331dc89b
subnet-9f2d29f7
subnet-2c18a457
subnet-0f578b641a16621d4
subnet-08c88e05cf0cf5bb0
subnet-0e2c3a137906707b9
subnet-041e6b48
Amazon Virtual Private Cloud (VPC) Internet Gateways
Select/Deselect all Amazon Virtual Private Cloud (VPC) Internet Gateways
igw-023b73c22b2b4cd3b
igw-061b410b81b0cf3e9
igw-092e0f22224637e48
igw-b3e840db
Amazon Virtual Private Cloud (VPC) Customer Gateways
No resources found in this AWS account

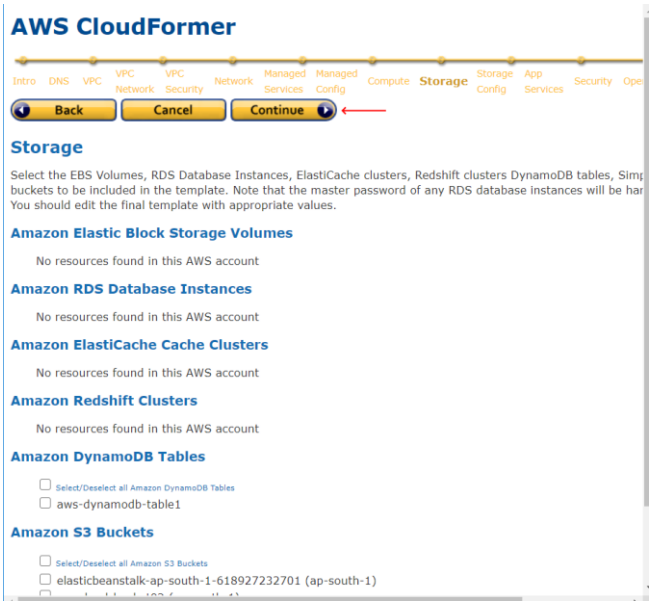

AWS CloudFormer
Intro DNS VPC VPC Network VPC Security Network Managed Services Managed Config Compute Storage Storage Config App Services Security Operati
Back
Cancel
Continue
Virtual Private Cloud Security Configuration
Select the Virtual Private Cloud (VPC) security configuration to be included in the template.
Amazon Virtual Private Cloud (VPC) Network ACLs
Select/Deselect all Amazon Virtual Private Cloud (VPC) Network ACLs
acl-033ddab1e853df233
acl-0b5742a34c09f8439
acl-592cea32
acl-0cd80f0998e7652ee
Amazon Virtual Private Cloud (VPC) Route Tables
Select/Deselect all Amazon Virtual Private Cloud (VPC) Route Tables
rtb-00f3f110c9076081b
rtb-0234e3fd013c8cabf
rtb-0ee57236e9b7cb8e0
rtb-092dc9fd1ca5891db
rtb-9e72aff5
rtb-0026ce6fa03932d80
rtb-06226285611b7489c

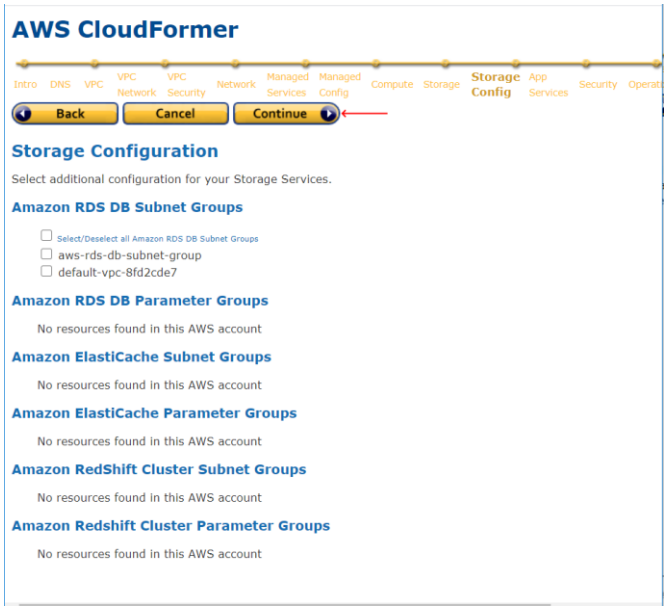

AWS CloudFormer
Intro DNS VPC VPC Network VPC Security Network Managed Services Managed Config Compute Storage Storage Config App Services Security Operati
Back
Cancel
Continue
Network Resources
Select the network entry points to be included in the template. If you select an EIP or and ENI that has an instance
an Elastic Load Balancer that has instances or an Auto Scaling group associated with it, the instances or Auto Scalin
selected for inclusion in the template by default, however, you can customize them in the next step.
Elastic Load Balancers
No resources found in this AWS account
Amazon EC2 Elastic IP Addresses
No resources found in this AWS account
Amazon EC2 Network Interfaces
Select/Deselect all Amazon EC2 Network Interfaces
eni-055571d1cf7958114
Amazon CloudFront Distributions
Select/Deselect all Amazon CloudFront Distributions
d2egm83spelhfx.cloudfront.net

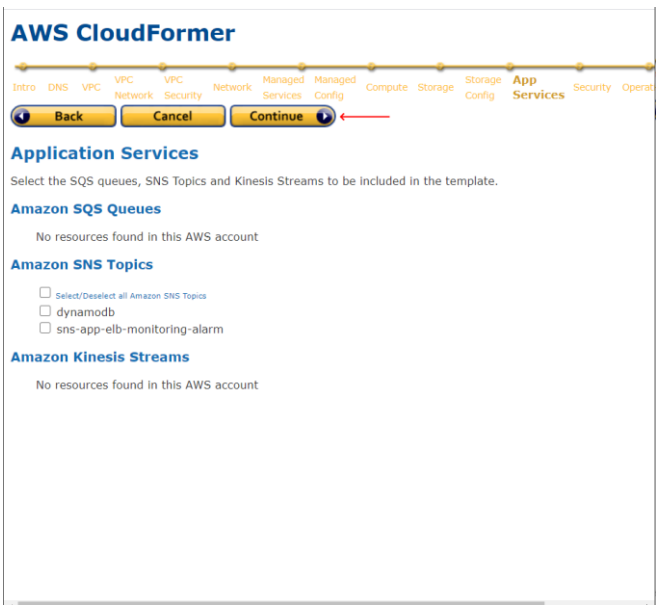

AWS CloudFormer
Intro DNS VPC VPC Network VPC Security Network Managed Services Managed Config Compute Storage Storage Config App Services Security Operati
Back
Cancel
Continue
Managed Services
Select the managed services to include in the template. You can customize the content of the managed services in t
Auto Scaling Groups
No resources found in this AWS account
Elastic Beanstalk Applications
Select/Deselect all Elastic Beanstalk Applications
test
OpsWorks Stacks
No resources found in this AWS account


AWS CloudFormer
Intro DNS VPC VPC Network VPC Security Network Managed Services Managed Config Compute Storage Storage Config App Services Security Ope
Back
Cancel
Continue
Managed Service Configuration
Select the configuration for managed services.
Auto Scaling Launch Configurations
Select/Deselect all Auto Scaling Launch Configurations
windows-web-servers-launch-configuration
Elastic Beanstalk Application Versions
Select/Deselect all Elastic Beanstalk Application Versions
test-source
Elastic Beanstalk Environments
No resources found in this AWS account
Elastic Beanstalk Configuration Templates
No resources found in this AWS account
OpsWorks Apps
No resources found in this AWS account
OpsWorks Layers
No resources found in this AWS account
OpsWorks Elastic Load Balancer Attachements

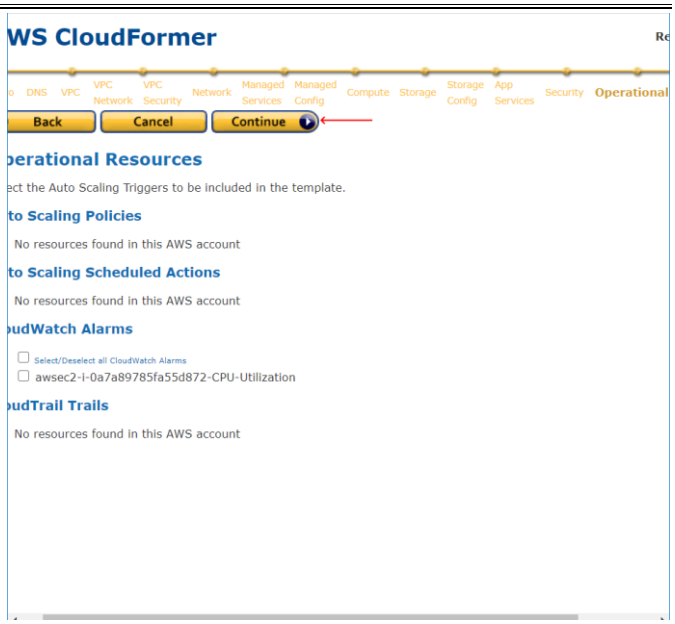

AWS CloudFormer
Intro DNS VPC VPC Network VPC Security Network Managed Services Managed Config Compute Storage Storage Config App Services Security Operati
Back
Cancel
Continue
Compute Resources
Select the EC2 instances and OpsWorks instances to be included in the template.If you select instances that are ass
volumes, the volumes will be selected for inclusion in the template by default, however, you can customize them in
Amazon EC2 Instances
Select/Deselect all Amazon EC2 Instances
i-03fd0dafba2421fcc
i-06849720858cc3d82
i-0f79df90bcaa05f29
i-0fbbba2f601639816
i-03dff666cc428a25d
OpsWorks Instances
No resources found in this AWS account

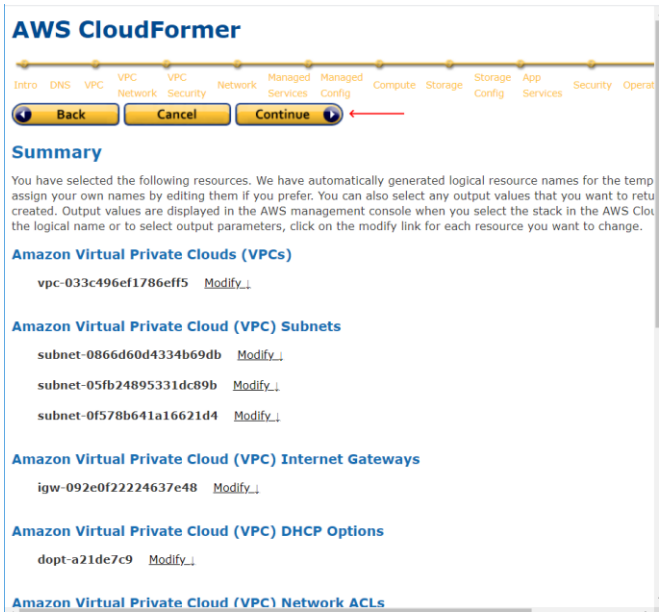

AWS CloudFormer
Intro DNS VPC VPC Network VPC Security Network Managed Services Managed Config Compute Storage Storage Config App Services Security Ope
Back
Cancel
Continue
Storage
Select the EBS Volumes, RDS Database Instances, ElastiCache clusters, Redshift clusters DynamoDB tables, Simp
buckets to be included in the template. Note that the master password of any RDS database instances will be har
You should edit the final template with appropriate values.
Amazon Elastic Block Storage Volumes
No resources found in this AWS account
Amazon RDS Database Instances
No resources found in this AWS account
Amazon ElastiCache Cache Clusters
No resources found in this AWS account
Amazon Redshift Clusters
No resources found in this AWS account
Amazon DynamoDB Tables
Select/Deselect all Amazon DynamoDB Tables
aws-dynamodb-table1
Amazon S3 Buckets
Select/Deselect all Amazon S3 Buckets
elasticbeanstalk-ap-south-1-618927232701 (ap-south-1)

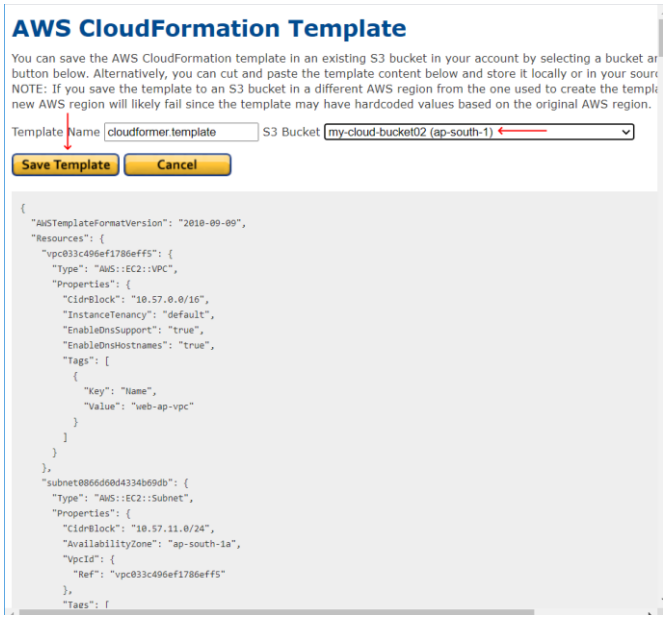

AWS CloudFormer
Intro DNS VPC VPC Network VPC Security Network Managed Services Managed Config Compute Storage Storage Config App Services Security Operati
Back
Cancel
Continue
Storage Configuration
Select additional configuration for your Storage Services.
Amazon RDS DB Subnet Groups
Select/Deselect all Amazon RDS DB Subnet Groups
aws-rds-db-subnet-group
default-vpc-8fd2cde7
Amazon RDS DB Parameter Groups
No resources found in this AWS account
Amazon ElastiCache Subnet Groups
No resources found in this AWS account
Amazon ElastiCache Parameter Groups
No resources found in this AWS account
Amazon RedShift Cluster Subnet Groups
No resources found in this AWS account
Amazon Redshift Cluster Parameter Groups
No resources found in this AWS account

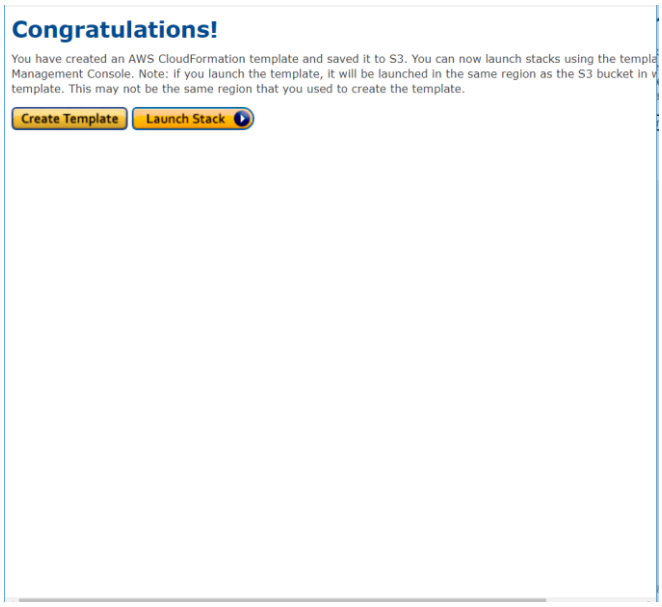

AWS CloudFormer
Intro DNS VPC VPC Network VPC Security Network Managed Services Managed Config Compute Storage Storage Config App Services Security Operat
Back
Cancel
Continue
Application Services
Select the SQS queues, SNS Topics and Kinesis Streams to be included in the template.
Amazon SQS Queues
No resources found in this AWS account
Amazon SNS Topics
Select/Deselect all Amazon SNS Topics
dynamodb
sns-app-elb-monitoring-alarm
Amazon Kinesis Streams
No resources found in this AWS account

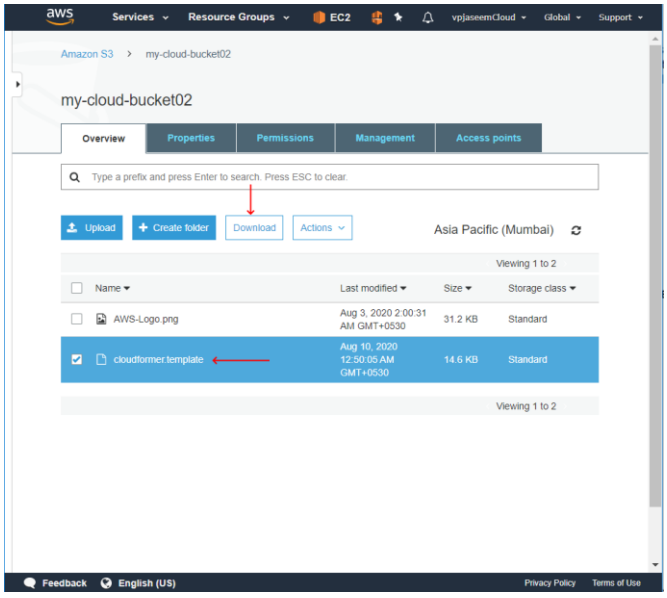

AWS CloudFormer
Intro DNS VPC VPC Network VPC Security Network Managed Services Managed Config Compute Storage Storage Config App Services Security
Back
Cancel
Continue
Security Groups
Select the Security Groups and Policies to be included in the template. The default security groups in the account they have been modified.
Amazon EC2 Security Groups
Select/Deselect all Amazon EC2 Security Groups
default
web-ap-vpc-nat-security-group
public-windows-security-group
web-ap-vpc-windows-security-group
web-ap-vpc-instance-security-group
default
web-ap-vpc-nat-instance-security-group
default
CloudFormer-WebServerSecurityGroup-1AVANL41MXNA3
web-server-security-group
default
Amazon RDS Security Groups
Select/Deselect all Amazon RDS Security Groups
default
Amazon ElastiCache Security Groups
Error enumerating resources: Use of cache security groups is not permitted in this API version for your accou

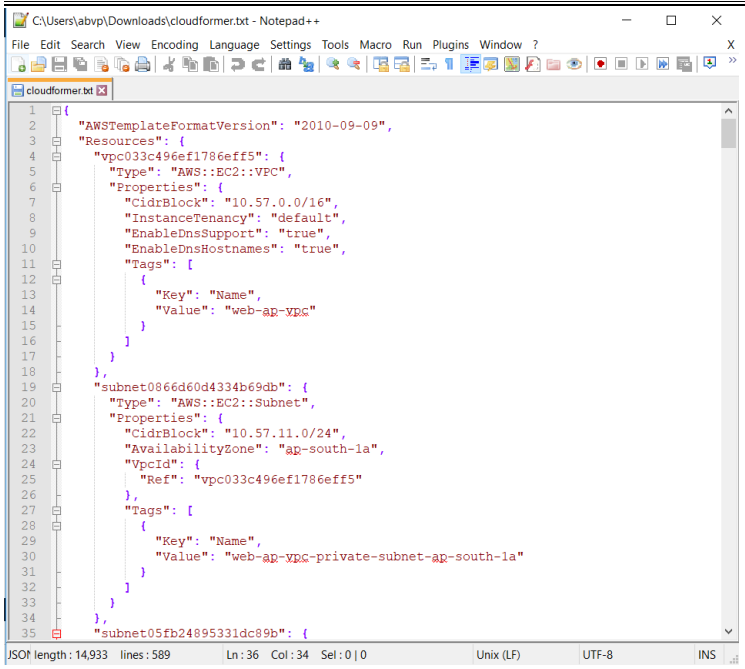

Back
Cancel
Continue
Operational Resources
Auto Scaling Policies
No resources found in this AWS account
Auto Scaling Scheduled Actions
No resources found in this AWS account
CloudWatch Alarms
Select/Deselect all CloudWatch Alarms
awsec2-i-0a7a89785fa55d872-CPU-Utilization
CloudTrail Trails
No resources found in this AWS account


AWS CloudFormer
Back
Cancel
Continue
Summary
You have selected the following resources. We have automatically generated logical resource names for the temp
assign your own names by editing them if you prefer. You can also select any output values that you want to retu
created. Output values are displayed in the AWS management console when you select the stack in the AWS Clou
the logical name or to select output parameters, click on the modify link for each resource you want to change.
Amazon Virtual Private Clouds (VPCs)
vpc-033c496ef1786eff5 Modify
Amazon Virtual Private Cloud (VPC) Subnets
subnet-0866d60d4334b69db Modify
subnet-05fb24895331dc89b Modify
subnet-0f578b641a16621d4 Modify
Amazon Virtual Private Cloud (VPC) Internet Gateways
igw-092e0f22224637e48 Modify
Amazon Virtual Private Cloud (VPC) DHCP Options
dopt-a21de7c9 Modify
Amazon Virtual Private Cloud (VPC) Network ACLs


AWS CloudFormation Template
You can save the AWS CloudFormation template in an existing S3 bucket in your account by selecting a bucket ar
button below. Alternatively, you can cut and paste the template content below and store it locally or in your sour
NOTE: If you save the template to an S3 bucket in a different AWS region from the one used to create the templa
new AWS region will likely fail since the template may have hardcoded values based on the original AWS region.
Template Name cloudformer.template
S3 Bucket my-cloud-bucket02 (ap-south-1)
Save Template
Cancel
{
"AWSTemplateFormatVersion": "2010-09-09",
"Resources": {
"vpc033c496ef1786eff5": {
"Type": "AWS::EC2::VPC",
"Properties": {
"CidrBlock": "10.57.0.0/16",
"InstanceTenancy": "default",
"EnableDnsSupport": "true",
"EnableDnsHostnames": "true",
"Tags": [
{
"Key": "Name",
"Value": "web-ap-vpc"
}
]
}
},
"subnet0866d60d4334b69db": {
"Type": "AWS::EC2::Subnet",
"Properties": {
"CidrBlock": "10.57.11.0/24",
"AvailabilityZone": "ap-south-1a",
"VpcId": {
"Ref": "vpc033c496ef1786eff5"
},
"Tags": [

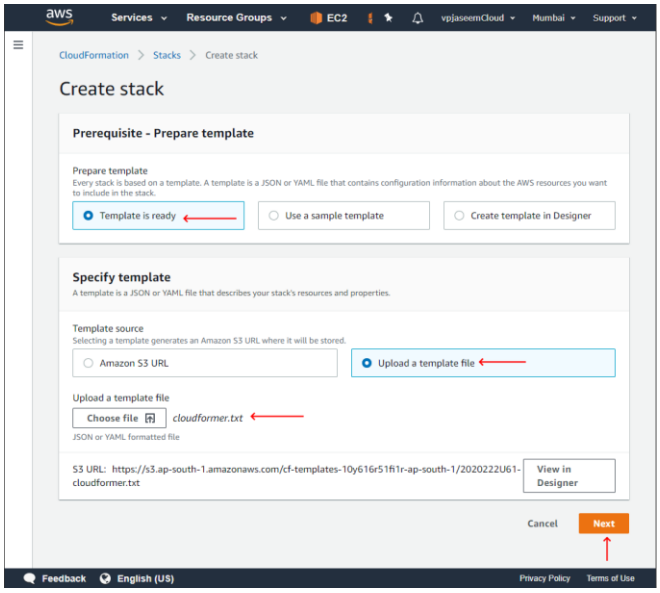

Congratulations!
You have created an AWS CloudFormation template and saved it to S3. You can now launch stacks using the templa
Management Console. Note: if you launch the template, it will be launched in the same region as the S3 bucket in w
template. This may not be the same region that you used to create the template.
Create Template
Launch Stack

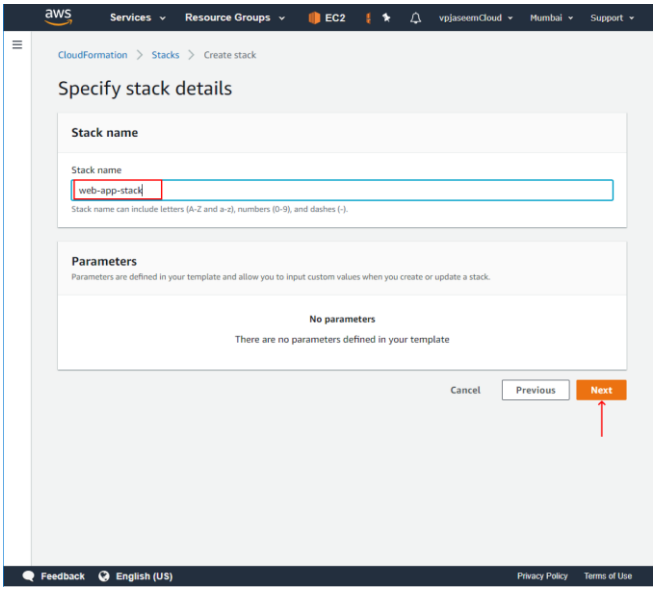

Services
Resource Groups
EC2
vpJaseemCloud
Global
Support
Amazon S3 > my-cloud-bucket02
my-cloud-bucket02
Overview
Properties
Permissions
Management
Access points
Type a prefix and press Enter to search. Press ESC to clear.
Upload
Create folder
Download
Actions
Asia Pacific (Mumbai)
Viewing 1 to 2
Name
Last modified
Size
Storage class
AWS-Logo.png
Aug 3, 2020 2:00:31 AM GMT+0530
31.2 KB
Standard
cloudformer.template
Aug 10, 2020 12:50:05 AM GMT+0530
14.6 KB
Standard
Viewing 1 to 2
Feedback
English (US)
Privacy Policy
Terms of Use

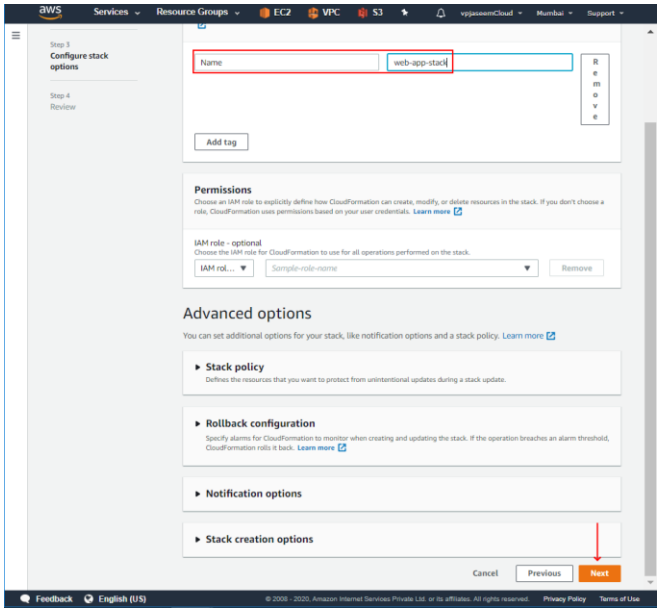


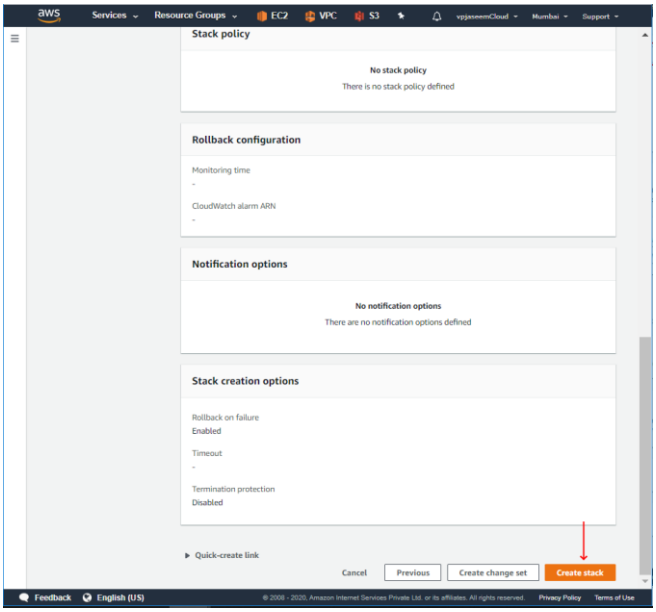

- Delete the associated components in web-ap-vpc (VPC, Subnets, Security Groups, EC2s, ELB, Target Group, Internet Gateway).

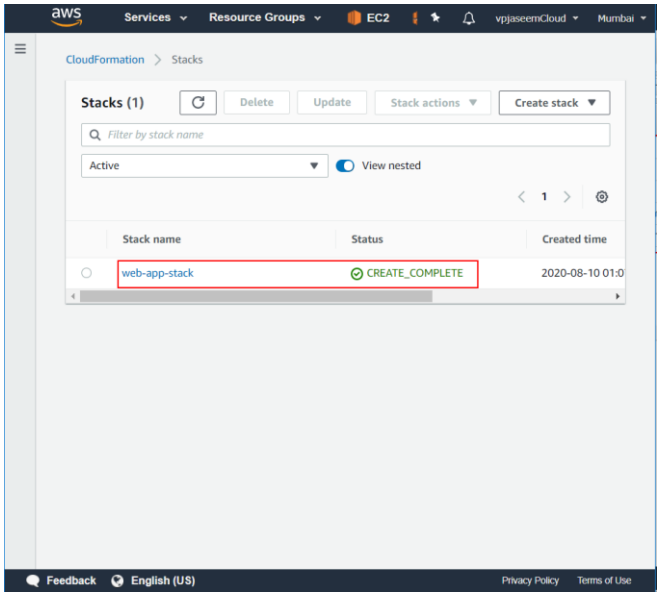
# [LAB] AWS Could Formation - Deploy from Template

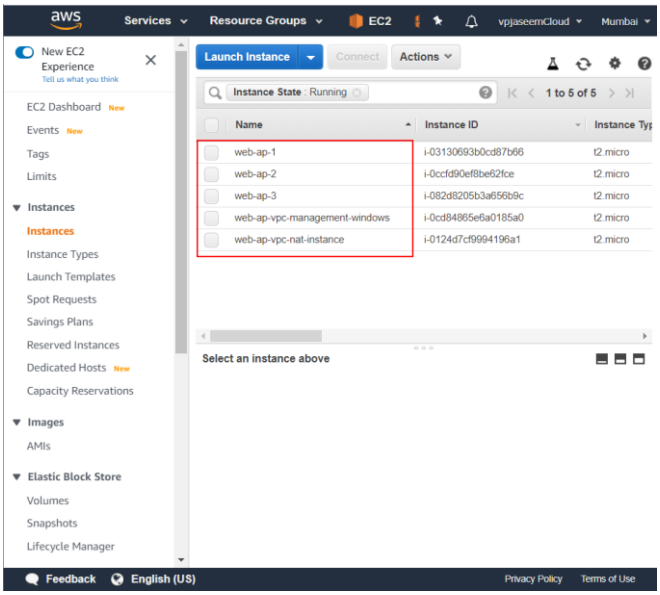


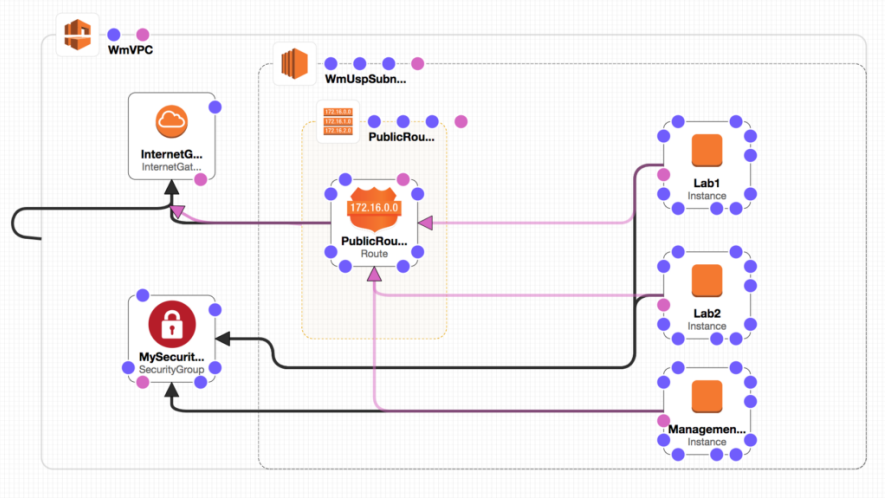




*Note: I was able to get the EC2 instanced from the template, but for some reason the ELB not created.*

## AWS Could Formation - Designer

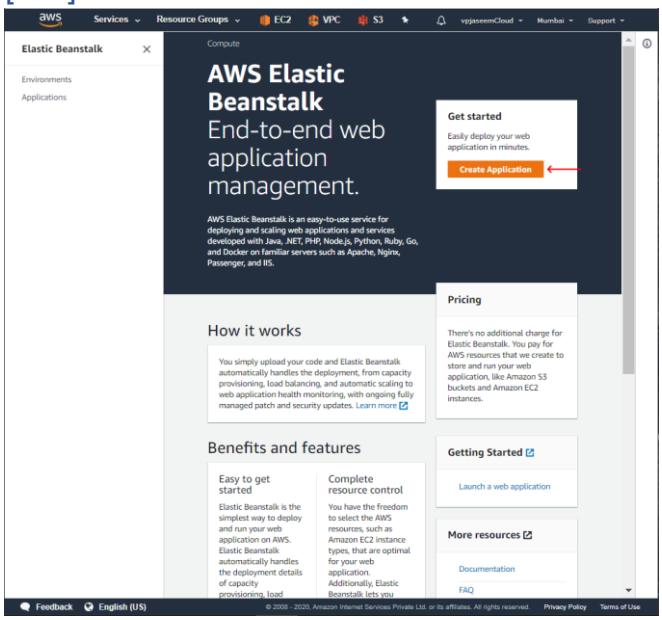


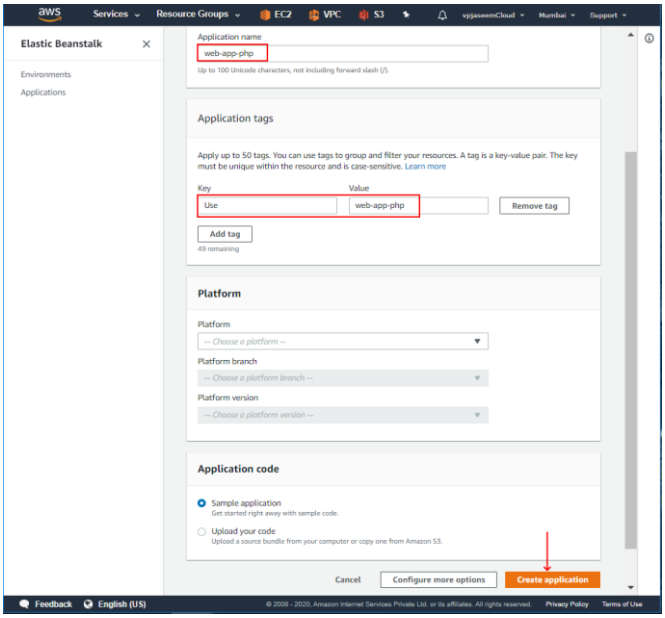

- AWS CloudFormation Designer is a graphic tool for creating, viewing, and modifying AWS CloudFormation templates.
- With Designer, you can diagram your template resources using a drag-and-drop interface, and then edit their details using the integrated JSON and YAML editor.
- Whether you are a new or an experienced AWS CloudFormation user, AWS CloudFormation Designer can help you quickly see the interrelationship between a template's resources and easily modify templates.

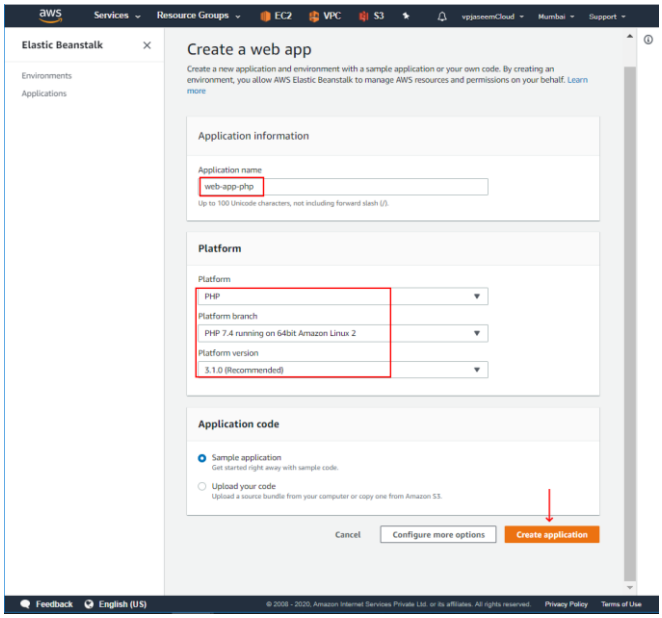
## Elastic Beanstalk

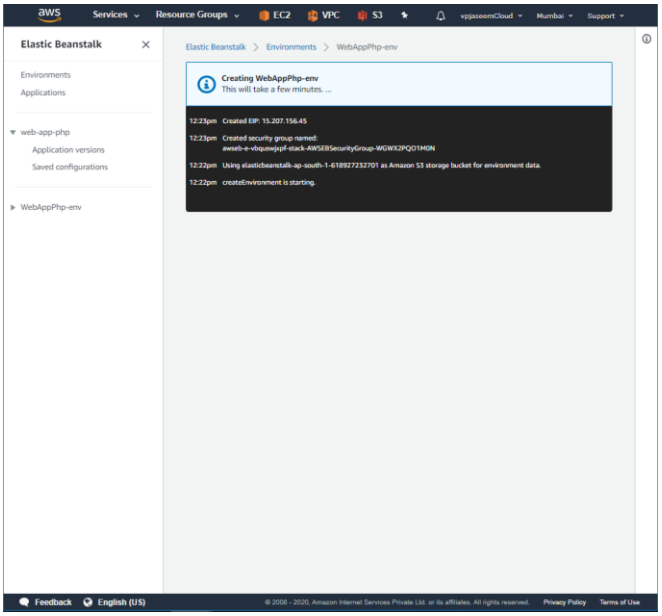
- Fully managed PaaS Web application hosting platform in AWS
- Built-in Monitoring tools and Security
- Automatically scale-up
- You have access to the underlying AWS resources at any time
- Support versioning
- Environment (EC2s, Security Group, etc.)
- Environment Tier: Web Server Environment or Worker Environment
- Environment Health: Color coded indication of the health
- Beanstalk Web Server Environment contains,
  - Elastic Load Balancer
  - Autoscaling Group
  - EC2 Instances
  - Host Manager: Runs on every EC2s, provides performance statistics, events, application logs, etc. in CloudWatch
  - Security Group: Allows HTTP traffic
- Beanstalk Worker Environment contains,
  - Worker handles background tasks which are time intensive and processor intensive, send email notification, etc.
- Web server tier Create SQS message and place in SQS queue, in Worked environment has Daemon that pulls the SQL queue.

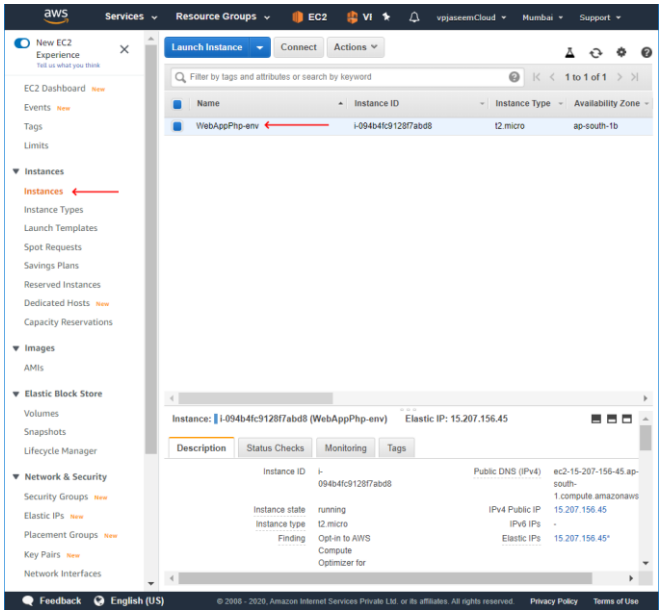
# [LAB] Elastic Beanstalk

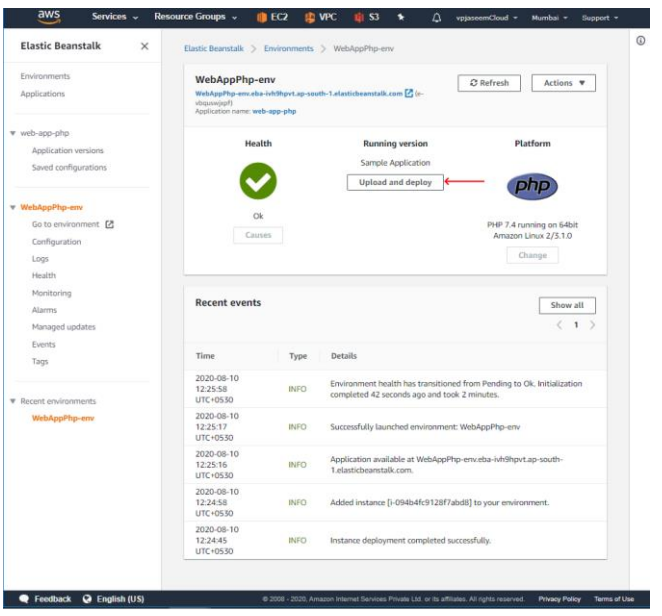


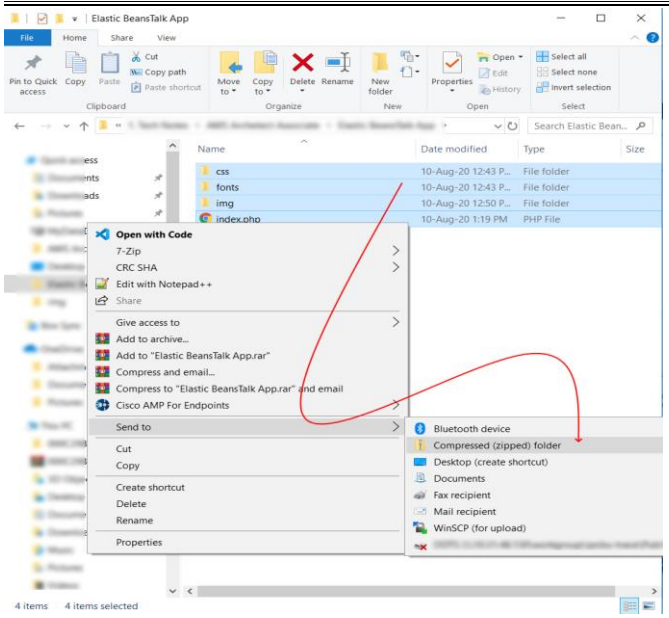



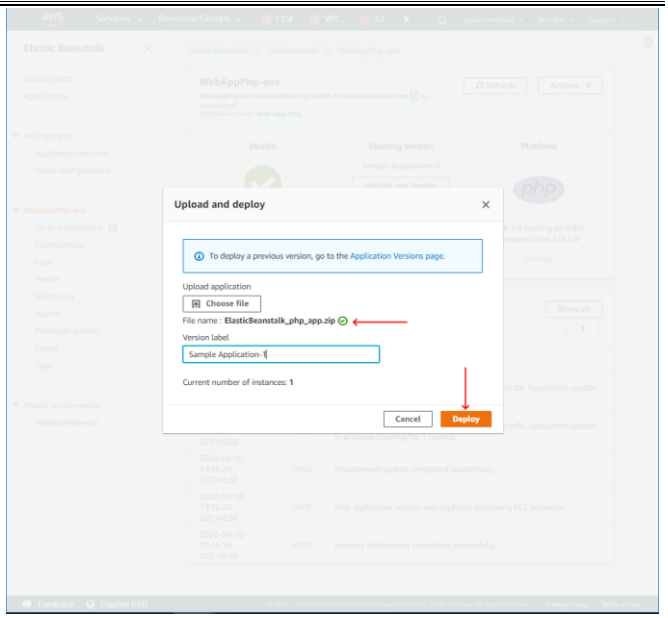



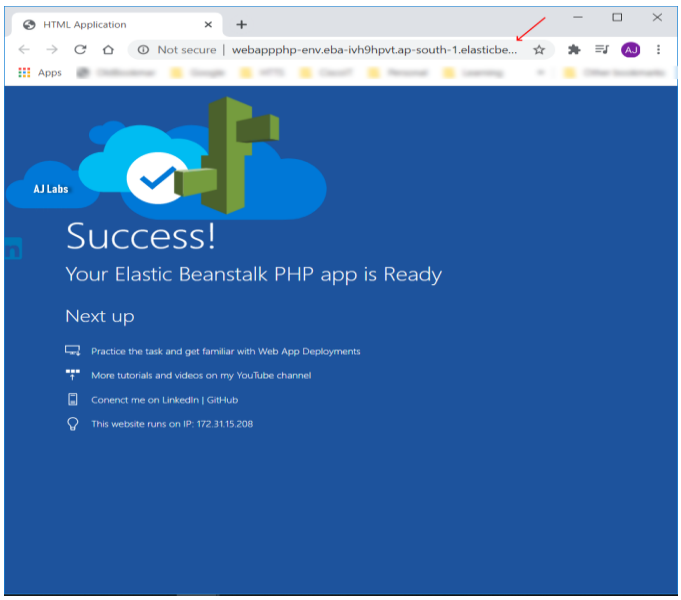



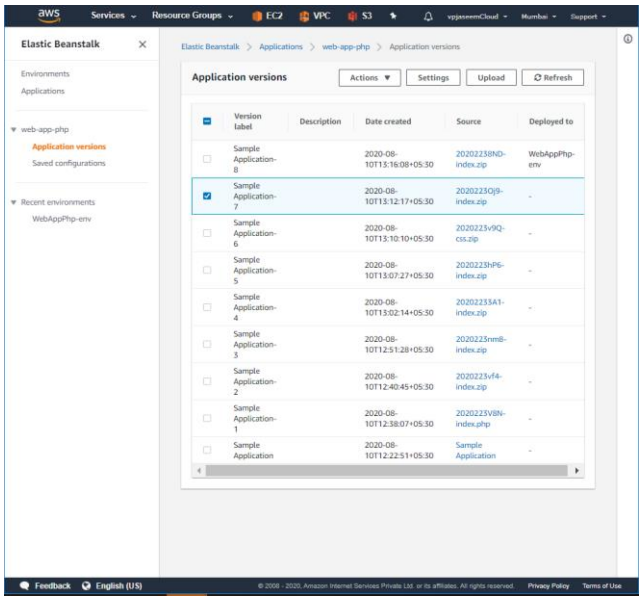



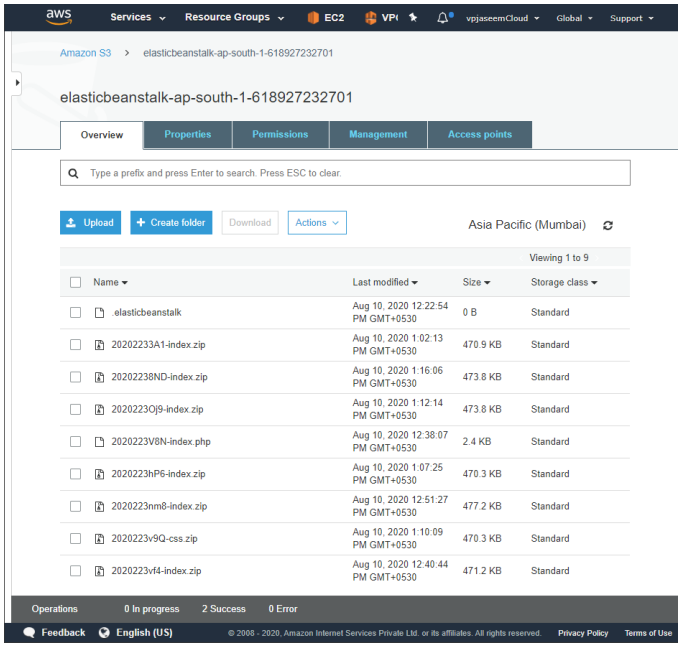



App Download Link: [Elastic Beanstalk Sample PHP App]



## Other AWS Services (Theory Only)


- In this section, we will discuss about some AWS Feature Services, no labs are included since those require some prerequisites (mainly on coding section).
- These services / features are intended for developers.

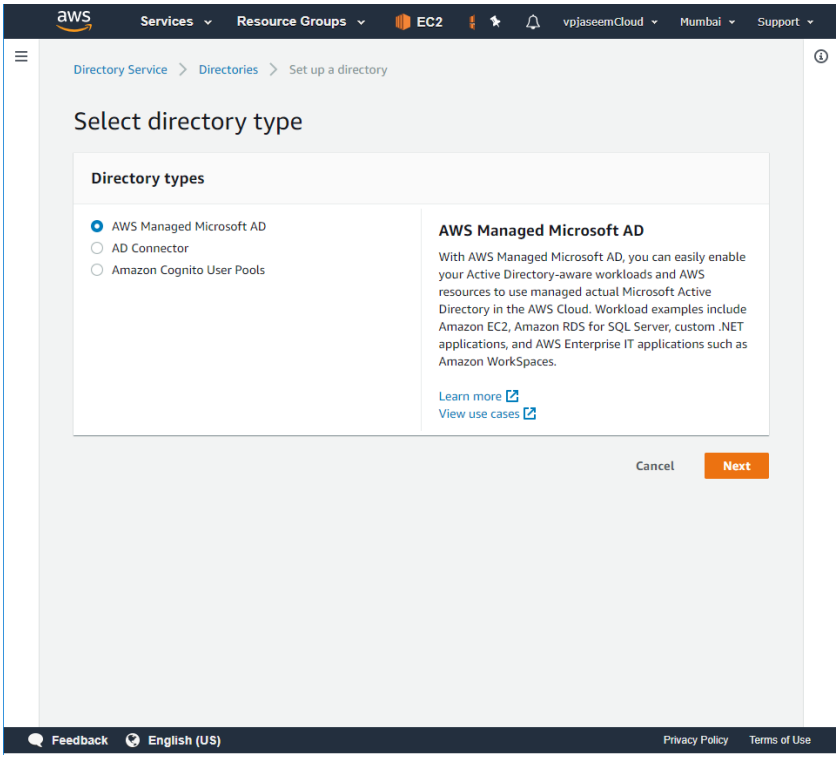
## AWS Cognito

- For user management in web app (User signup, notification, etc.)
- Amazon Cognito lets you add user sign-up, sign-in, and access control to your web and mobile apps quickly and easily.
- Amazon Cognito scales to millions of users and supports sign-in with social identity providers, such as Facebook, Google, and Amazon, and enterprise identity providers via SAML 2.0.

AWS Reference: Amazon Cognito

## AWS Directory Services

- Fully Managed Microsoft AD
- AD Connector to authenticate via on-prem AD
- Authentication from AD (AWS Managed or AD Connector to on-prem) and authorization via IAM
- Use Cognito to authenticate




AWS Reference: AWS Directory Service

## Single Sign-On (SSO)

- Grant access to your AWS accounts and any business applications that support SSO using SAML 2.0.
- Users simply sign in once to a personalized user portal using their directory credentials and can then access all their assigned applications without additional sign-in prompts.
- Multiple AWS Accounts can be authenticated.

AWS Reference: AWS Single Sign-On

## Storage Gateway

- The AWS Storage Gateway is a service connecting an on-premises software appliance with cloud-based storage to provide seamless and secure integration between an organization's on-premises IT environment and AWS's storage infrastructure.
- File Gateway, Volume gateway, Tape Gateway are available.
- You can deploy an on-premise local VM that acts as a gateway to S3 storage.
- AWS Storage Gateway is a hybrid cloud storage service that gives you on-premises access to virtually unlimited cloud storage.
- It is like a DropBox software installed on local computer.

AWS reference: AWS Storage Gateway

## API Gateway

- Amazon API Gateway is a fully managed service that makes it easy for developers to create, publish, maintain, monitor, and secure APIs at any scale.
- APIs act as the "front door" for applications to access data, business logic, or functionality from your backend services.

Amazon Reference: API Gateway

YouTube: AWS API gateway to Lambda Function

## AWS OpsWorks

- Fully managed configuration management service uses Chef and Puppet.
- Configuration management service helps to operate and change configurations instantly.
- For example, you want to change properties of 1000 servers running at a time, OpsWorks can be used.

Amazon Reference: Aws OpsWorks

YouTube: AWS OpsWorks Demo

## RedShift

- Used for OLAP (Online Analytical Processing) systems or Data warehousing.
- Columnar databases, we usually select columns in data warehousing scenarios.

Amazon Reference: What is Amazon Redshift?

## Kinesis

- Amazon Kinesis makes it easy to collect, process, and analyze real-time, streaming data so you can get timely insights and react quickly to new information.
- Used for AI applications.
- CCTV Camera devices can stream videos to Kinesis directly and can be used for Machine Learning.

AWS Reference: Kinesis

YouTube: AWS Kinesis Tutorial

## Elastic File System (EFS)

- Fully managed elastic NFS file system for use with AWS Cloud services and on-premises resources.
- Replicated across AZs in a region
- Once EFS Volume can be mounter to N number of EC2 instances like a shared drive
- Could be mounted to on premise server as well (Over VPN or Direct Connect)
- No sizing to be done
- Using Amazon EFS with Microsoft Windows–based Amazon EC2 instances is not supported.

AWS Reference: What is EFS?

## FXs

- Amazon FSx for Windows File Server provides fully managed Microsoft Windows file servers, backed by a fully native Windows file system.
- This the NFS for Windows.

AWS Reference: Amazon FSx

## Terra Forms

- We can templatize your infrastructure, compatible with AWS, Google Cloud and Microsoft Azure.
- This is like Cloud Formation template.
- Use Infrastructure as Code to provision and manage any cloud, infrastructure, or service.

## About the Author:

**Abdul Jaseem VP**, a Network Consulting Engineer who has worked with complex IT infrastructure in multiple enterprise environments. He is currently employed with Cisco Systems India Pvt. LTD. TAC Team. He has seven plus years of industry experience in Cisco Collaboration, Networking, Cloud Computing, Virtualization and Security.

Presently lives in Bangalore and basically from a small village called Karuvarakundu in the state of Kerala, India. He loves conducting industry standard IT trainings for people across the globe.

You can connect him via Email: vpjaseem@gmail.com; LinkedIn; YouTube