

DyaSaaTech

AMAZON WEB SERVICES

By

Mr. RG (IT Expert)

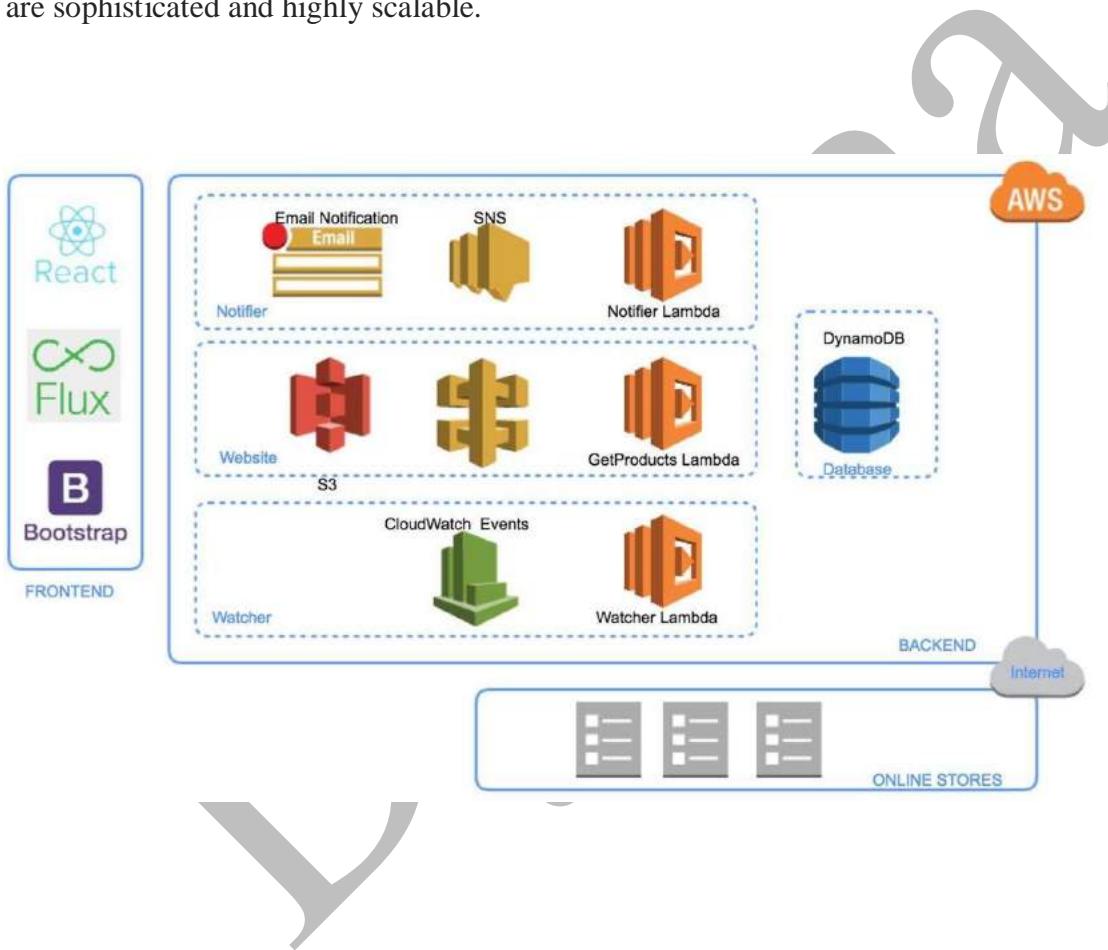
Content Index

1.	Introduction	3
2.	Amazon Elastic Compute Cloud (EC2).....	5
3.	Elastic Block Storage (EBS)	26
4.	Elastic File System (EFS)	42
5.	Elastic Load Balancing (ELB).....	53
6.	Auto Scaling	99
7.	Amazon Simple Storage Service (S3).....	116
8.	Identity and Access Management (IAM).....	141
9.	Virtual Private Cloud (VPC)	178
10.	Route-53	232
11.	Amazon Relational Database Service (RDS)	243
12.	Simple Notification Services (SNS)	254
13.	Simple Email Services (SES)	264
14.	Simple Queue Services (SQS).....	272
15.	Cloud Formation	286
16.	Cloud Watch	302
17.	Cloud Trail	318
18.	AWS CLI.....	326

Introduction

What is AWS? – **Amazon Web Services(AWS)** is a cloud service from Amazon, which provides services in the form of building blocks, these building blocks can be used to create and deploy any type of application in the cloud.

These services or building blocks are designed to work with each other, and result in applications which are sophisticated and highly scalable.



Each type of service in this “What is AWS” blog, is categorized under a domain, the few domains which are widely used are:

- Compute
- Storage
- Database
- Network and Content Delivery
- Management Tools
- Security & Identity Compliance

- Messaging

The **Compute** domain includes services related to compute workloads, it includes the following services:

- EC2 (Elastic Compute Cloud)

The **Storage** domain includes services related data storage, it includes the following services:

- S3 (Simple Storage Service)
- Elastic Block Store
- Amazon Glacier

The **Database** domain is used for database related workloads, it includes the following services:

- Amazon RDS

The **Networking and Content Delivery** domain is used for isolating your network infrastructure, and content delivery is used for faster delivery of content. It includes the following services:

- Amazon VPC
- Amazon Route 53

The **Management Tools** domain consists of services which are used to manage other services in AWS, it includes the following services:

- AWS CloudWatch
- AWS CloudFormation
- AWS CloudTrail

The **Security & Identity, Compliance** domain consist of services which are used to manage to authenticate and provide security to your AWS resources. It consists of the following services:

- AWS IAM

The **Messaging** domain consists of services which are used for queuing, notifying or emailing messages. It consists of the following domains:

- Amazon SQS
- Amazon SNS
- Amazon SES

Amazon Elastic Compute Cloud (EC2)

Amazon Elastic Compute Cloud (Amazon EC2) provides scalable computing capacity in the Amazon Web Services (AWS) cloud. Using Amazon EC2 eliminates your need to invest in hardware up front, so you can develop and deploy applications faster. You can use Amazon EC2 to launch as many or as few virtual servers as you need, configure security and networking, and manage storage. Amazon EC2 enables you to scale up or down to handle changes in requirements or spikes in popularity, reducing your need to forecast traffic.

Features of Amazon EC2

Amazon EC2 provides the following features:

- Virtual computing environments, known as *instances*
- Preconfigured templates for your instances, known as *Amazon Machine Images (AMIs)*, that package the bits you need for your server (including the operating system and additional software)
- Various configurations of CPU, memory, storage, and networking capacity for your instances, known as *instance types*
- Secure login information for your instances using *key pairs* (AWS stores the public key, and you store the private key in a secure place)
- Storage volumes for temporary data that's deleted when you stop or terminate your instance, known as *instance store volumes*
- Persistent storage volumes for your data using Amazon Elastic Block Store (Amazon EBS), known as *Amazon EBS volumes*
- Multiple physical locations for your resources, such as instances and Amazon EBS volumes, known as *regions* and *Availability Zones*
- A firewall that enables you to specify the protocols, ports, and source IP ranges that can reach your instances using *security groups*
- Static IPv4 addresses for dynamic cloud computing, known as *Elastic IP addresses*
- Metadata, known as *tags*, that you can create and assign to your Amazon EC2 resources

- Virtual networks you can create that are logically isolated from the rest of the AWS cloud, and that you can optionally connect to your own network, known as *virtual private clouds*(VPCs)

Sign Up for AWS

When you sign up for Amazon Web Services (AWS), your AWS account is automatically signed up for all services in AWS, including Amazon EC2. You are charged only for the services that you use.

With Amazon EC2, you pay only for what you use. If you are a new AWS customer, you can get started with Amazon EC2 for free. For more information, see [AWS Free Tier](#).

If you have an AWS account already, skip to the next task. If you don't have an AWS account, use the following procedure to create one.

To create an AWS account

1. Open <https://aws.amazon.com/>, and then choose **Create an AWS Account**.

Note

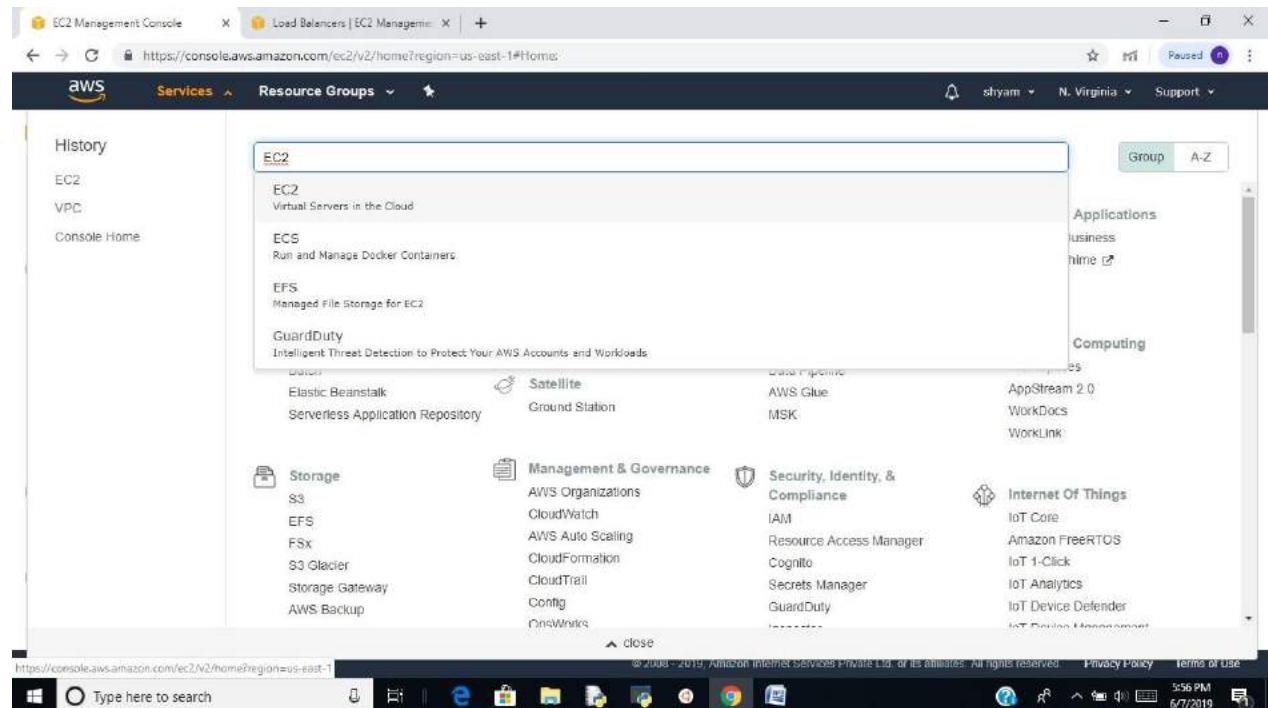
If you previously signed in to the AWS Management Console using AWS account root user credentials, choose **Sign in to a different account**. If you previously signed in to the console using IAM credentials, choose **Sign-in using root account credentials**. Then choose **Create a new AWS account**.

2. Follow the online instructions.

Part of the sign-up procedure involves receiving a phone call and entering a verification code using the phone keypad.

Launch Instances

- Signin into aws console by using aws account credentials that is account id and password in aws console signin page
- Select EC2 service and click on that



- Click on Launch Instance

The screenshot shows the AWS EC2 Management Console Home page. The sidebar on the left includes links for EC2 Dashboard, Instances, Images, and Elastic Block Store. The main content area displays various Amazon EC2 resources in the US East (N. Virginia) region, with counts for Running Instances, Dedicated Hosts, Volumes, Key Pairs, Placement Groups, Elastic IPs, Snapshots, Load Balancers, and Security Groups. A 'Create Instance' section with a 'Launch Instance' button is visible. On the right, there are sections for Account Attributes (listing Supported Platforms like VPC and Default VPC), Additional Information (links to Getting Started Guide, Documentation, etc.), and AWS Marketplace (listing free software trial products). The bottom of the screen shows a Windows taskbar with the date and time (5:58 PM, 6/7/2019).

- Choose AMI: select any AMI from list of AMIs. Here I select ubuntu 16 version. AMI is simply called OS for our launching machine
- Click on Next

Amazon Web Services

The screenshot shows the AWS Launch Instance Wizard Step 1: Choose an Amazon Machine Image (AMI). The page displays a list of available AMIs:

- Ubuntu Server 16.04 LTS (HVM), SSD Volume Type** - ami-07b4156579ea1d7ba (64-bit x86) / ami-036ede09922dadc9b (64-bit Arm)
Status: Free tier eligible
Root device type: ebs Virtualization type: hvm ENA Enabled: Yes
Select button (64-bit (x86))
Select button (64-bit (Arm))
- Microsoft Windows Server 2019 Base** - ami-0a9ca0496f746e6e0
Status: Free tier eligible
Root device type: ebs Virtualization type: hvm ENA Enabled: Yes
Select button (64-bit (x86))



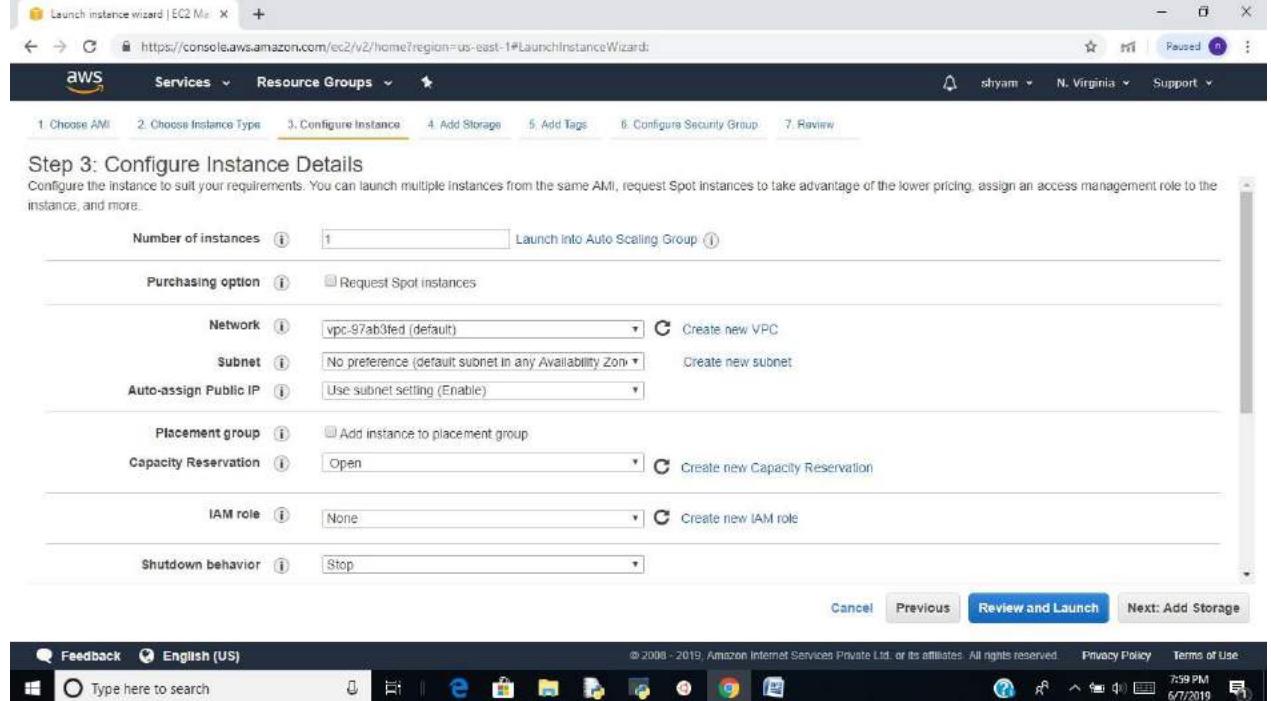
- Choose Instance Type: select Instance type t2.micro for free tier usage
- Click on Next Configure details

The screenshot shows the AWS Launch Instance Wizard Step 2: Choose an Instance Type. The page displays a table of instance types:

Family	Type	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance	IPv6 Support
General purpose	t2.nano	1	0.5	EBS only	-	Low to Moderate	Yes
General purpose	t2.micro	1	1	EBS only	-	Low to Moderate	Yes
General purpose	t2.small	1	2	EBS only	-	Low to Moderate	Yes
General purpose	t2.medium	2	4	EBS only	-	Low to Moderate	Yes
General purpose	t2.large	2	8	EBS only	-	Low to Moderate	Yes

Buttons at the bottom: Cancel, Previous, Review and Launch, Next: Configure Instance Details.

- Number of Instances: enter the number for how many instances are launched
- Network: select the VPC to launch those instances
- Subnet: select the subnet to launch those instances
- Auto-assign Public IP: use subnet setting Enable for assign public IP address to that instance
- IAM Role: select IAM role for that instance
- Shutdown behaviour: stop
- Click on next Add storage



- Enter the required memory size, volume type is gp2
- Click on Next Add tags

Step 4: Add Storage

Your instance will be launched with the following storage device settings. You can attach additional EBS 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. Learn more about storage options in Amazon EC2.

Volume Type	Device	Snapshot	Size (GiB)	Volume Type	IOPS	Throughput (MB/s)	Delete on Termination	Encryption
Root	/dev/sda1	snap-0e384451033d3767e	8	General Purpose SSD (gp2)	100 / 3000	N/A	<input checked="" type="checkbox"/>	Not Encrypted

Add New Volume

Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage. Learn more about free usage tier eligibility and usage restrictions.

- Add the tags: Enter key and value pair

Step 5: Add Tags

A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = Webserver.
A copy of a tag can be applied to volumes, instances or both.
Tags will be applied to all instances and volumes. Learn more about tagging your Amazon EC2 resources.

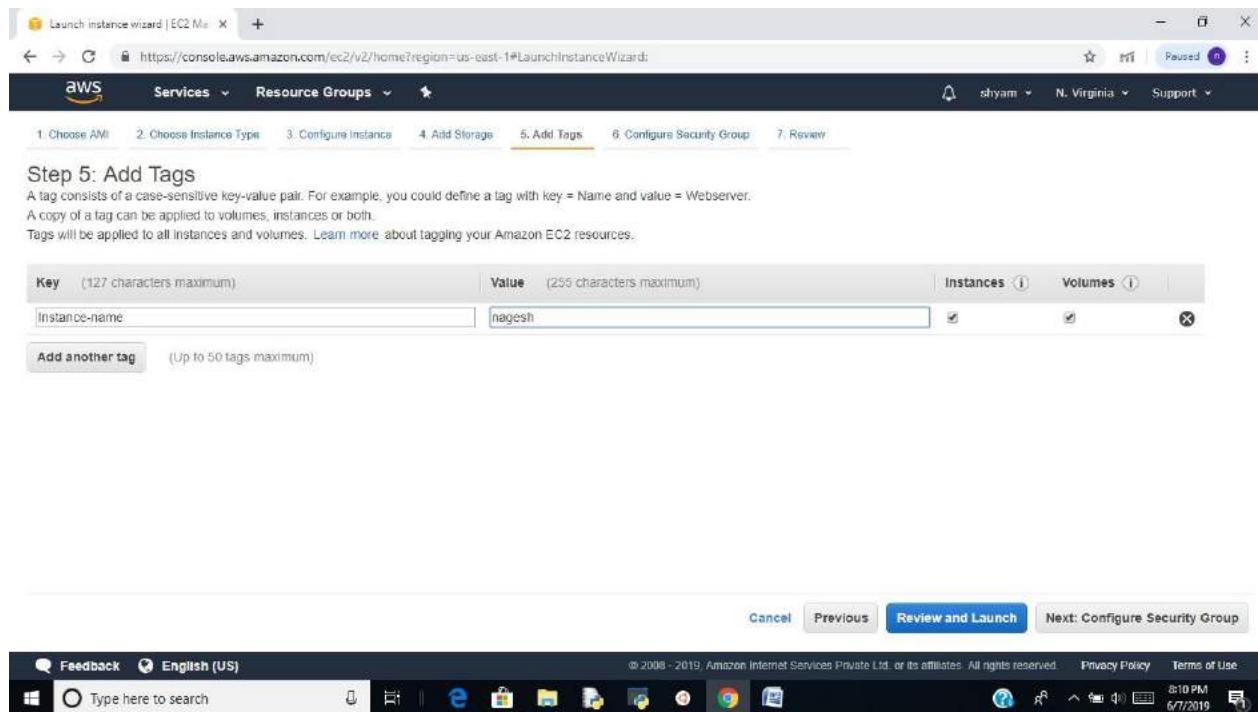
Key	Value	Instances	Volumes
Instance-name	nagesh	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Add another tag (Up to 50 tags maximum)

Cancel Previous Review and Launch Next: Configure Security Group

Feedback English (US) © 2008 - 2019, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. Privacy Policy Terms of Use 8:04 PM 6/7/2019

□ Next: configure security group



- By default ssh protocol is included in that security group
- Add required Protocols by click on Add Rule
- Type: select protocol
- Port Range: enter port number or range
- Source: select from any where option
- Click on review and lasunch

Amazon Web Services

Step 6: Configure Security Group

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. Learn more about Amazon EC2 security groups.

Assign a security group:

- Create a new security group
- Select an existing security group

Security group name: launch-wizard-14

Description: launch-wizard-14 created 2019-06-07T20:09:50.577+05:30

Type	Protocol	Port Range	Source	Description
SSH	TCP	22	Custom 0.0.0.0/0	e.g. SSH for Admin Desktop
HTTP	TCP	80	Anywhere 0.0.0.0/0, /0	e.g. SSH for Admin Desktop

Add Rule

Warning: Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

Cancel Previous Review and Launch

- Click on launch

Step 7: Review Instance Launch

Please review your instance launch details. You can go back to edit changes for each section. Click **Launch** to assign a key pair to your instance and complete the launch process.

AMI Details

Ubuntu Server 16.04 LTS (HVM), SSD Volume Type - ami-07b4156679ea1d7ba

Free tier eligible

Instance Type

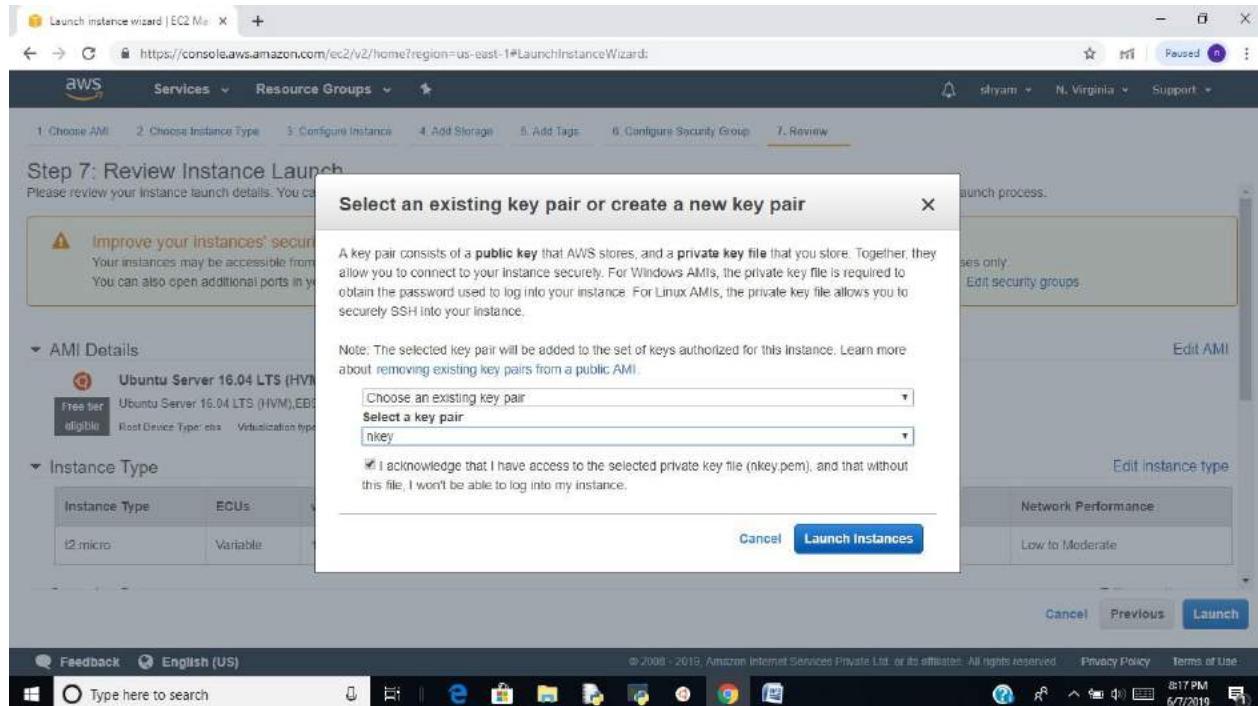
Instance Type	ECUs	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance
t2.micro	Variable	1	1	EBS only	-	Low to Moderate

Launch

Feedback English (US)

Cancel Previous Launch

- Key Pair: select existing key pair or choose new key pair option
- Enable acknowledgement
- Enter key pair name and click download key pair
- Click on Launch Instances

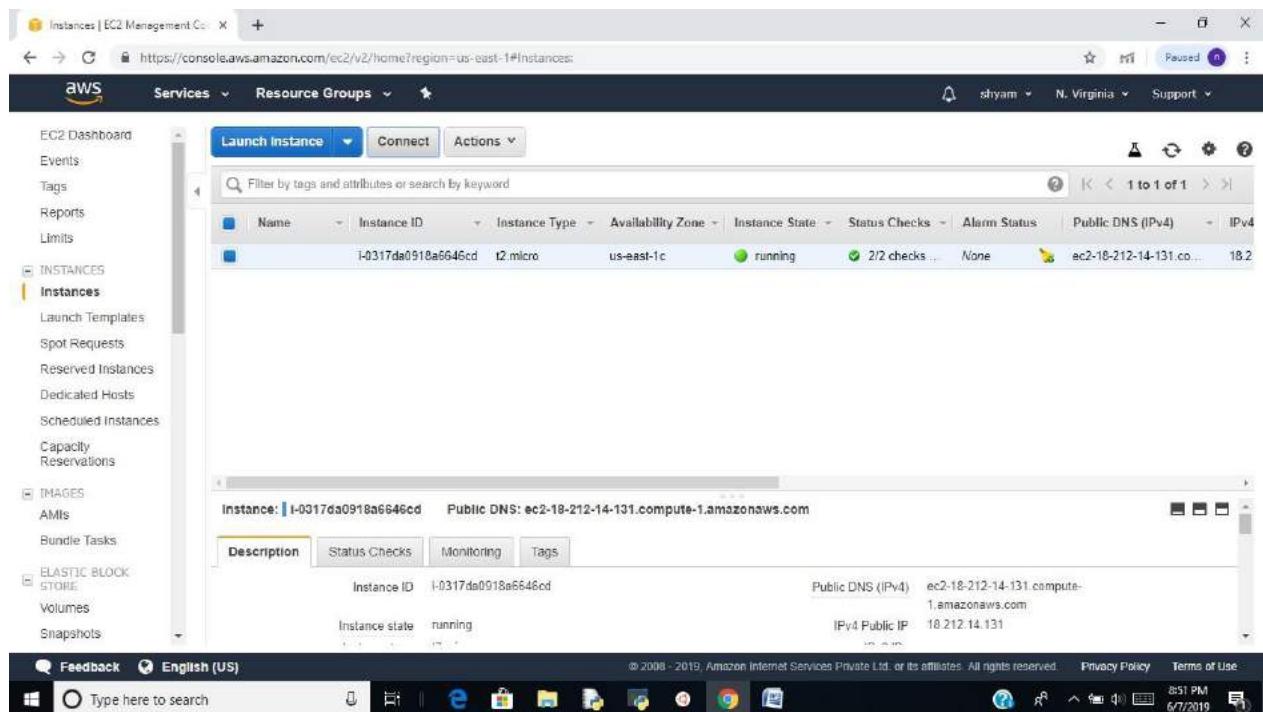


- Click on view instances

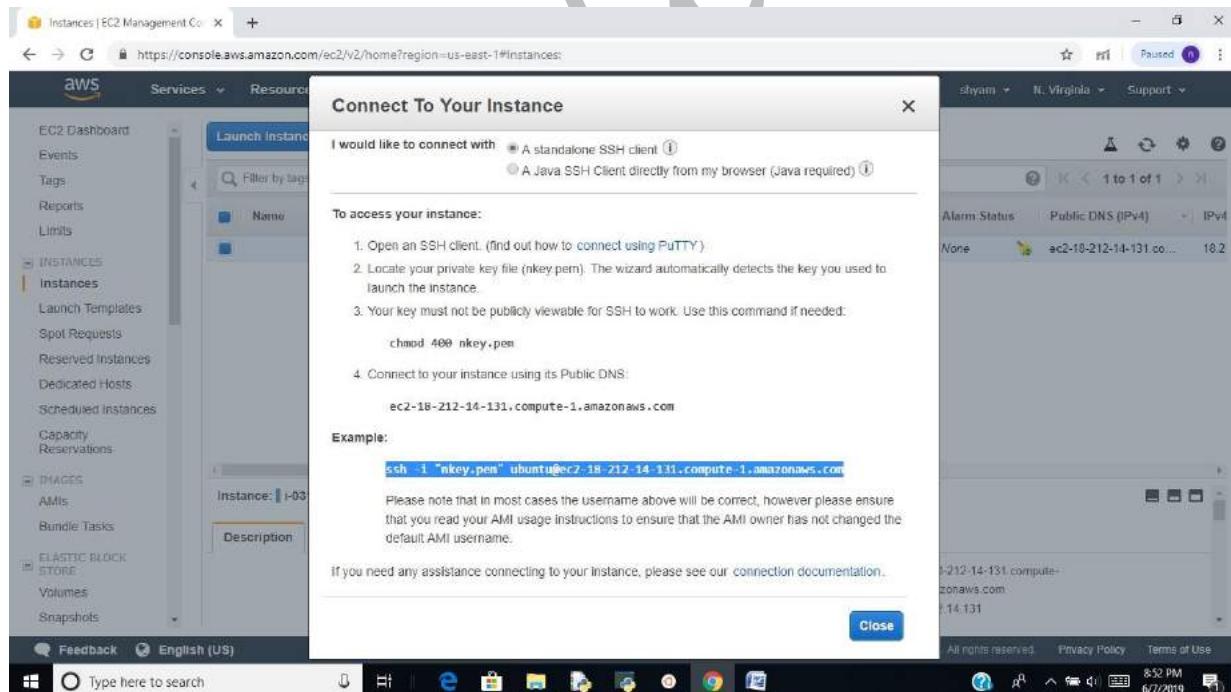
The screenshot shows the AWS Launch Instance Wizard interface. At the top, there's a navigation bar with tabs like 'Services' and 'Resource Groups'. Below the navigation bar, the main content area has a title 'Launch Status'. A callout box titled 'Get notified of estimated charges' provides instructions on creating billing alerts. Underneath, a section titled 'How to connect to your instances' explains that instances are launching and provides a link to 'View Instances' for monitoring. It also lists helpful resources like the User Guide and Discussion Forum. Further down, it suggests creating status check alarms, attaching EBS volumes, and managing security groups. A 'View Instances' button is located at the bottom right of the main content area.

Connect to Instance

- Open gitbash in private key download location that is downloads or open putty
- Select particular EC2 instance which instance do you want to connect and click connect option



Copy Example ssh command and click on close



- Paste in gitbash and make enter

The screenshot shows a Windows desktop environment. In the foreground, a Git Bash terminal window is open, displaying the command:

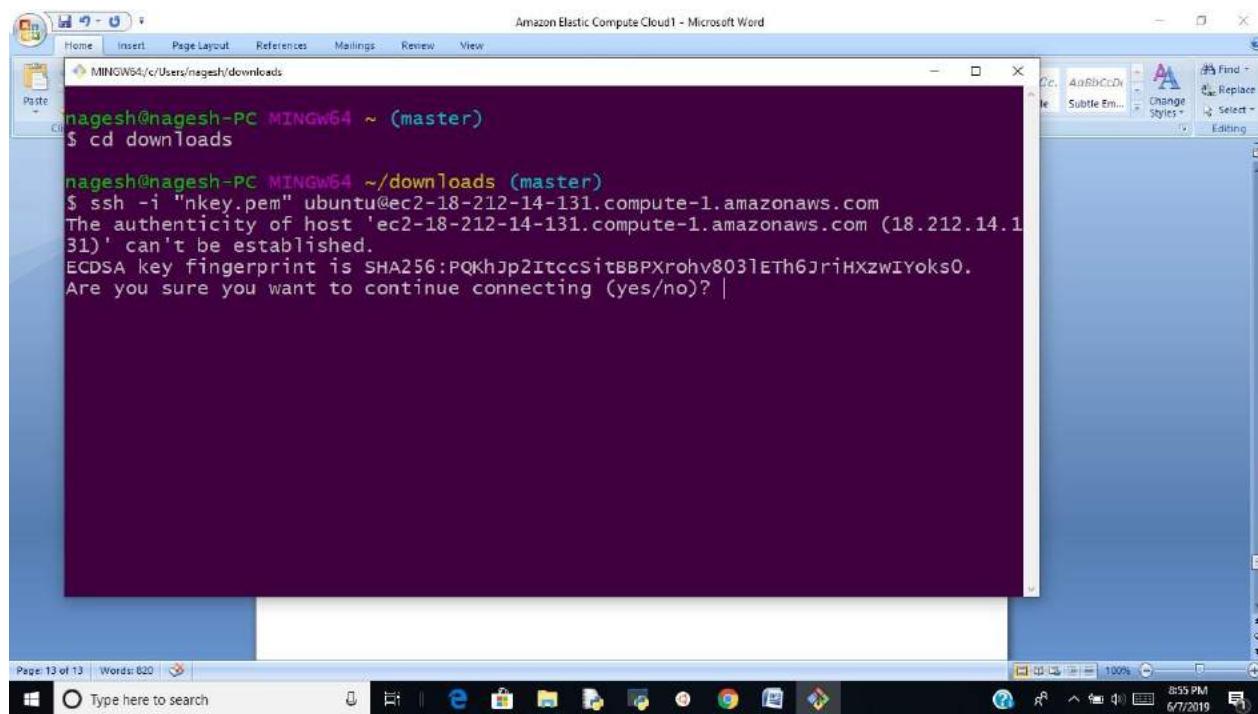
```
nagesh@nagesh-PC MINGW64 ~ (master)
$ cd downloads
$ ssh -i "nkey.pem" ubuntu@ec2-18-212-14-131.compute-1.amazonaws.com
```

In the background, the AWS Management Console is visible, showing the 'Instances' section with one instance listed:

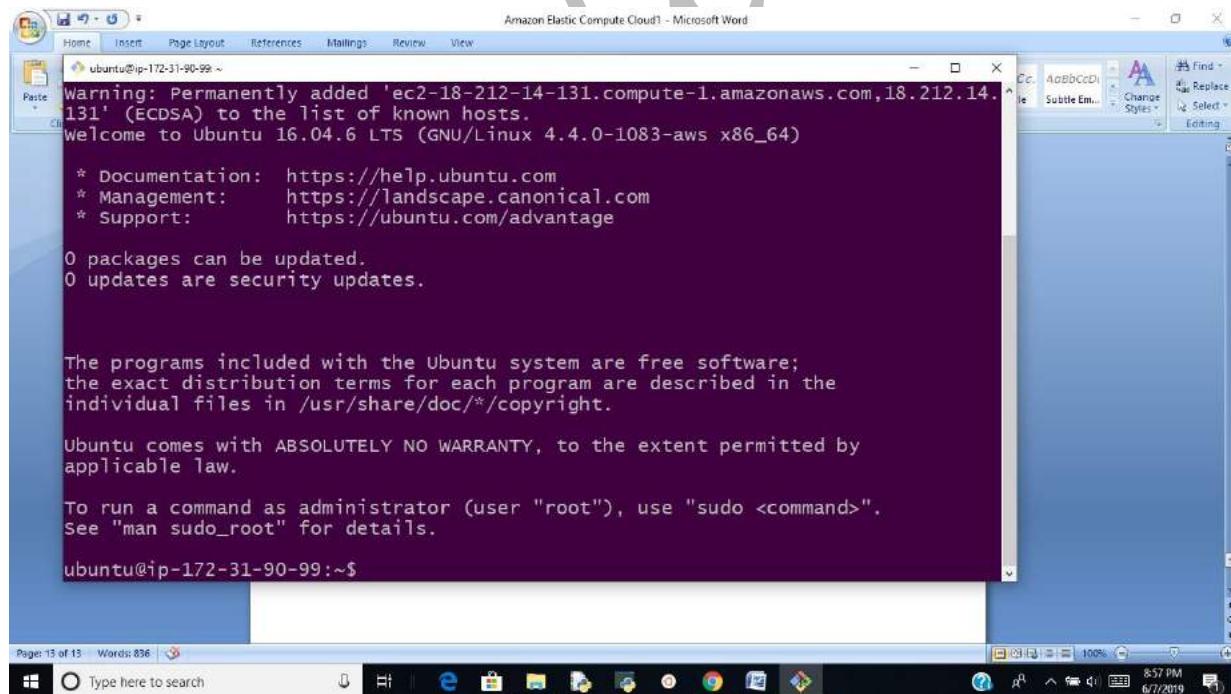
Public DNS (IPv4)	IPv4
ec2-18-212-14-131.co...	18.2...

The taskbar at the bottom shows various application icons.

- Enter yes

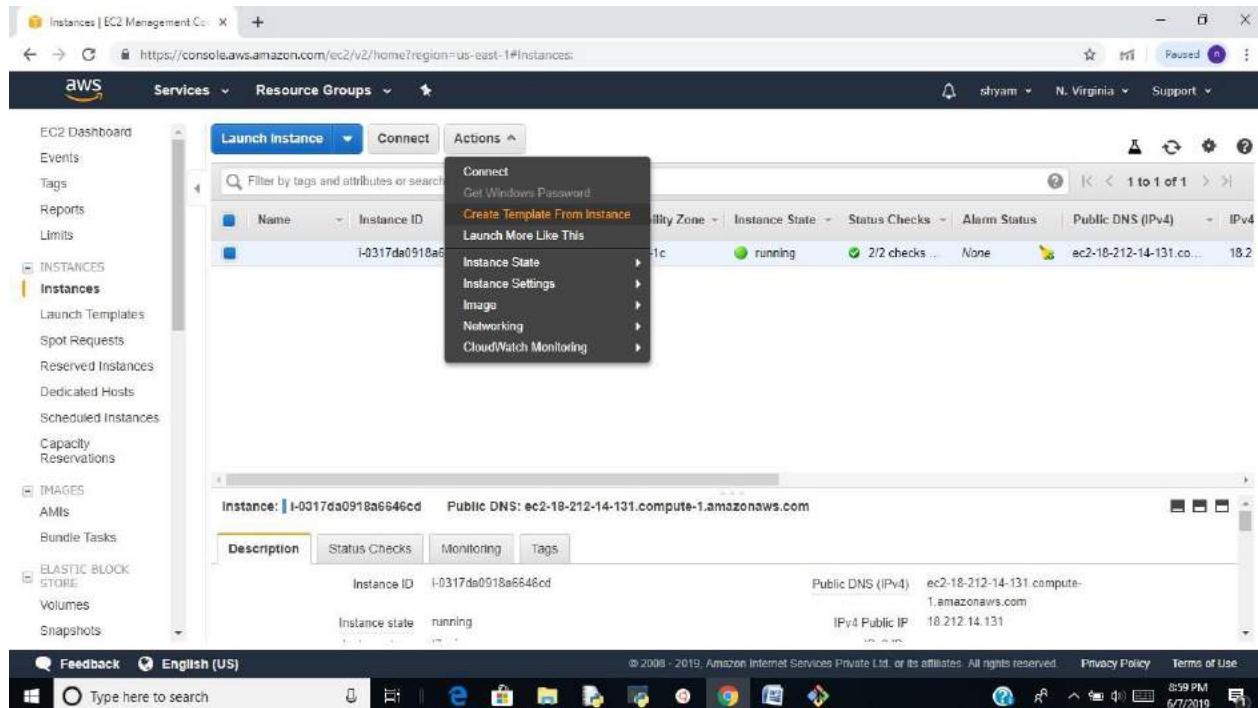


- Connected to ec2 instance and you can do any operations on that ec2 instance terminal



Create Template from Instance

- Select EC2 Instance and goto actions and select Create template from Instance option



- Launch Template: enter name for that template
- Click on create template from instance

The screenshot shows the AWS Create Template From Instance wizard. The first section, "Create Template From Instance," asks if you want to create a new template from an instance or a new template version from an existing one. The second section, "Launch template contents," allows setting the AMI ID (ami-07b4156579ea1d7ba) and instance type (t2.micro). The third section, "Network interfaces," shows a table with one row for eth0, which has an auto-assigned public IP of 123.123.123.1. The fourth section, "Storage (Volumes)," is currently empty.

Create Template From Instance

Creating a launch template allows you to create a saved instance configuration that can be reused, shared and launched at a later time. Templates can have multiple versions. You can either create a new template or create a new version of an existing template. When you create a new template you are creating a template and the first version of that template.

What would you like to do? Create a new template from an instance Create a new template version from an instance

Source instance I-0317da0918a6646cd

Launch template* e.g. MyTemplate (Max 125 chars. No spaces)

Template version description e.g. A prod webserver for MyApp (Max 255 chars)

Launch template contents

AMI ID ami-07b4156579ea1d7ba Search for AMI

Instance type t2.micro

Feedback English (US) © 2008 - 2019, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. Privacy Policy Terms of Use 9:02 PM 6/7/2019

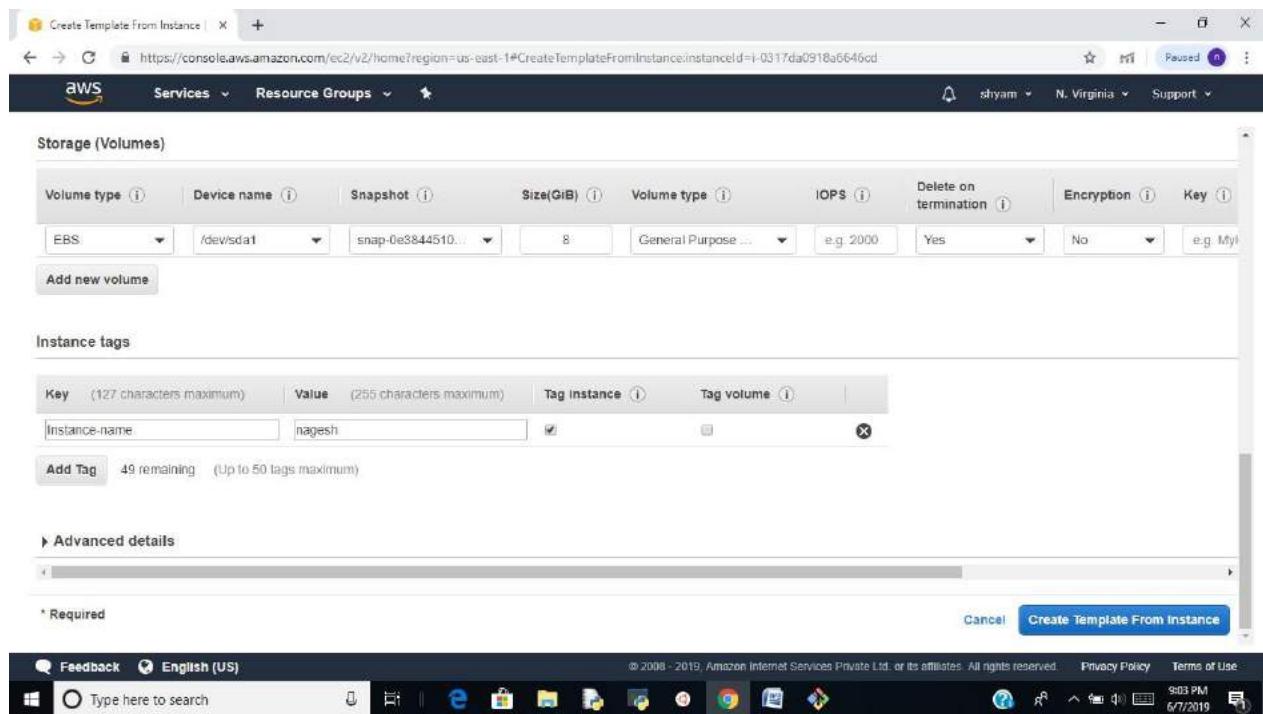
Network interfaces

Device	Network interface	Description	Subnet	Auto-assign public IP	Primary IP	Secondary IP	IPv6 IPs	Security groups
eth0	e.g. eni-12345678	e.g. My Primary ENI	subnet-8df4b3a3	Enable	e.g. 123.123.123.1	e.g. 123.123.123.1	e.g. 2001:0db8:85::	sg-00

Add network interface

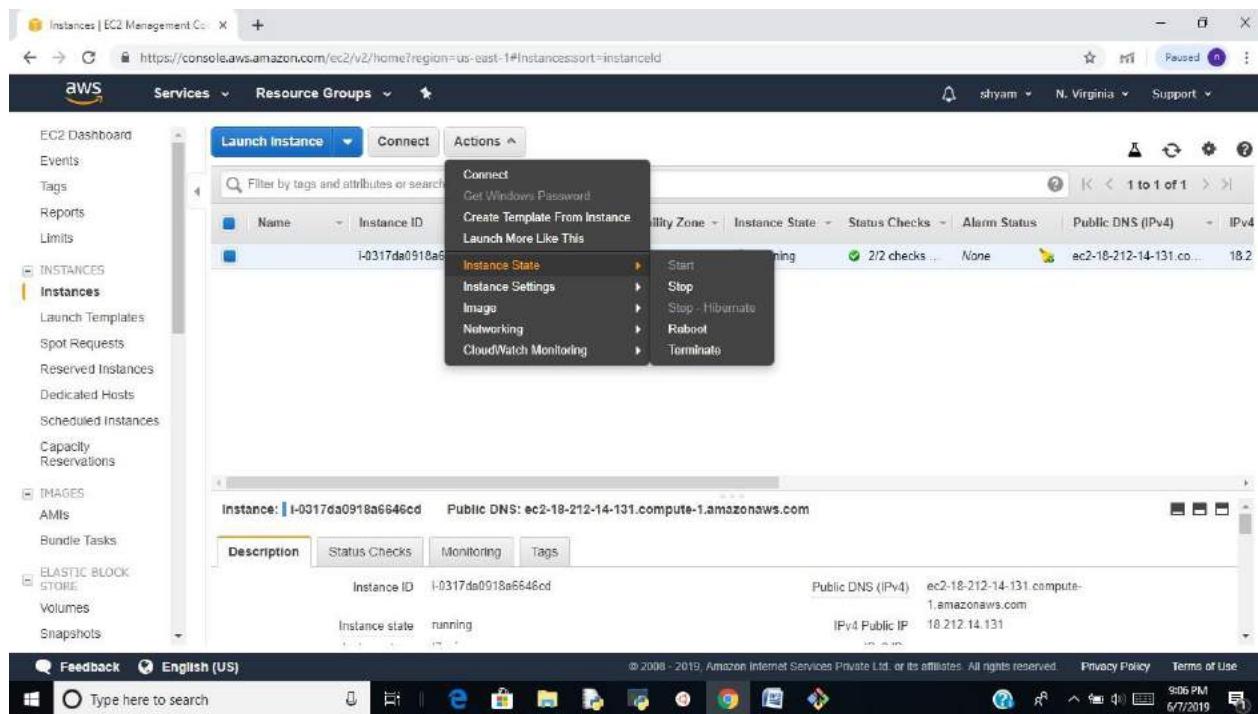
Storage (Volumes)

Feedback English (US) © 2008 - 2019, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. Privacy Policy Terms of Use 9:03 PM 6/7/2019



Instance State

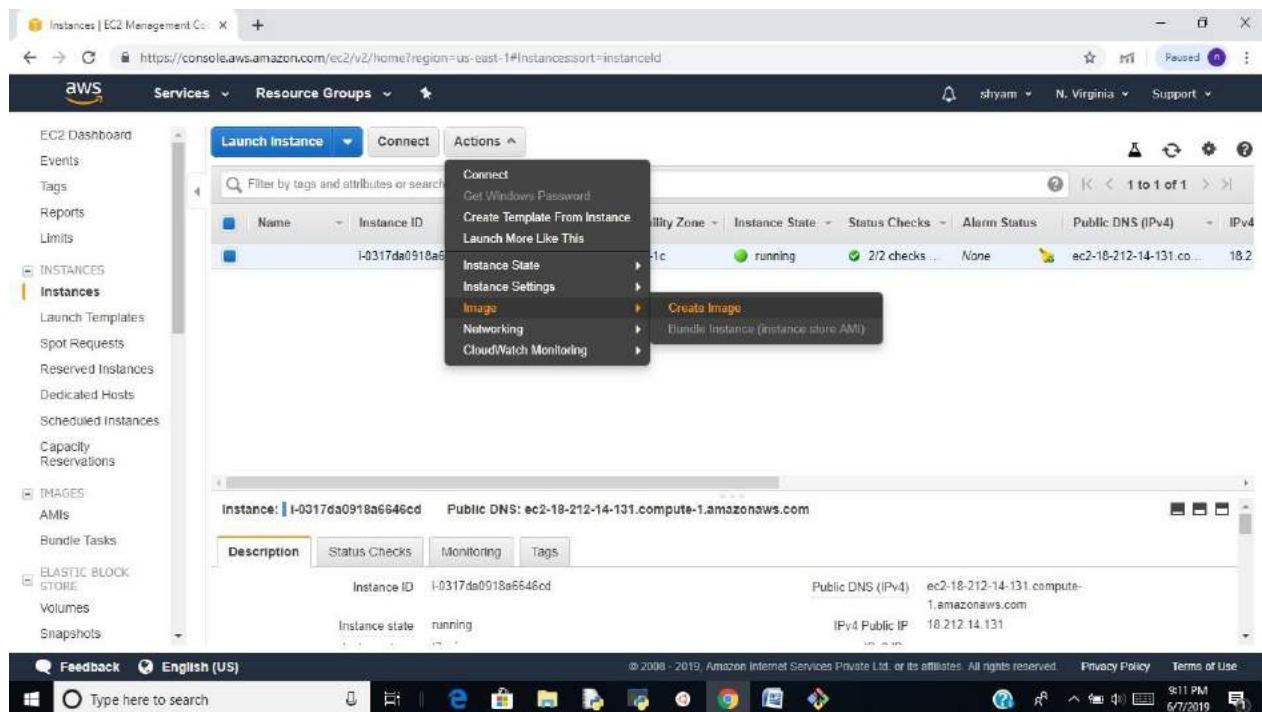
- Select EC2 instance and goto actions and select instance state option



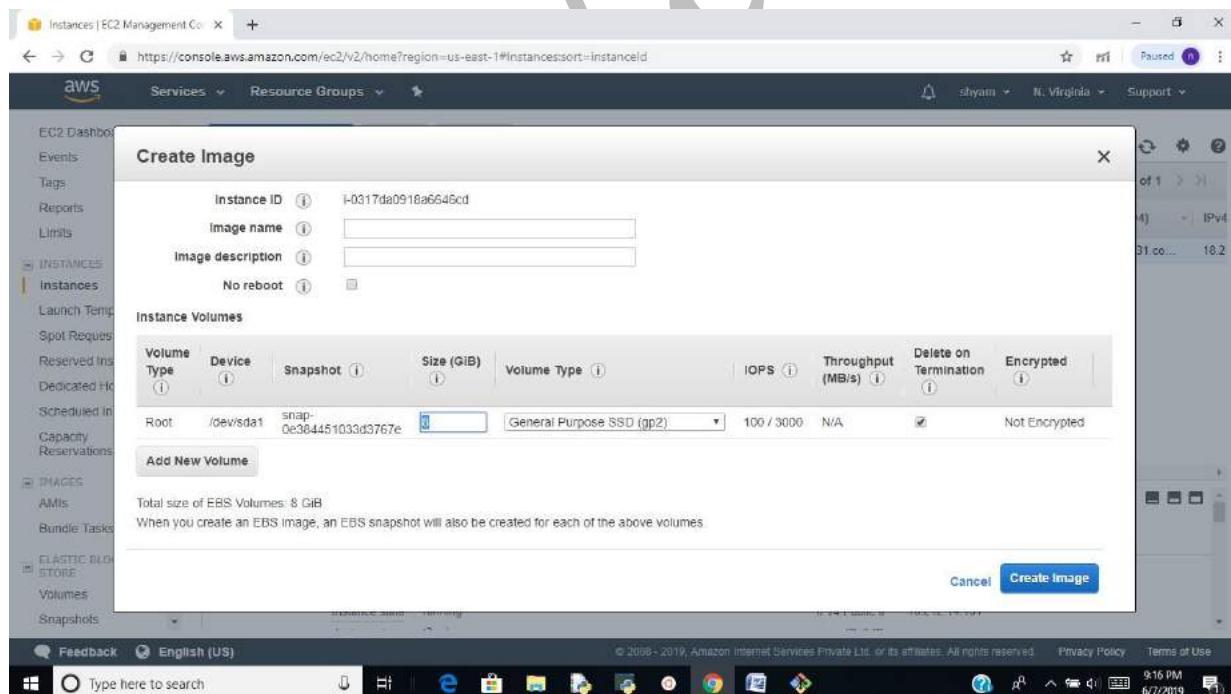
- Select start if instance is stopped and do you want to start
- Select stop if do you want to stop the instance
- Select reboot option for reboot your machine
- Select terminate for terminate your instance or machine

Create Image from Instance

- Select instance and goto actions and click on image and select create image option

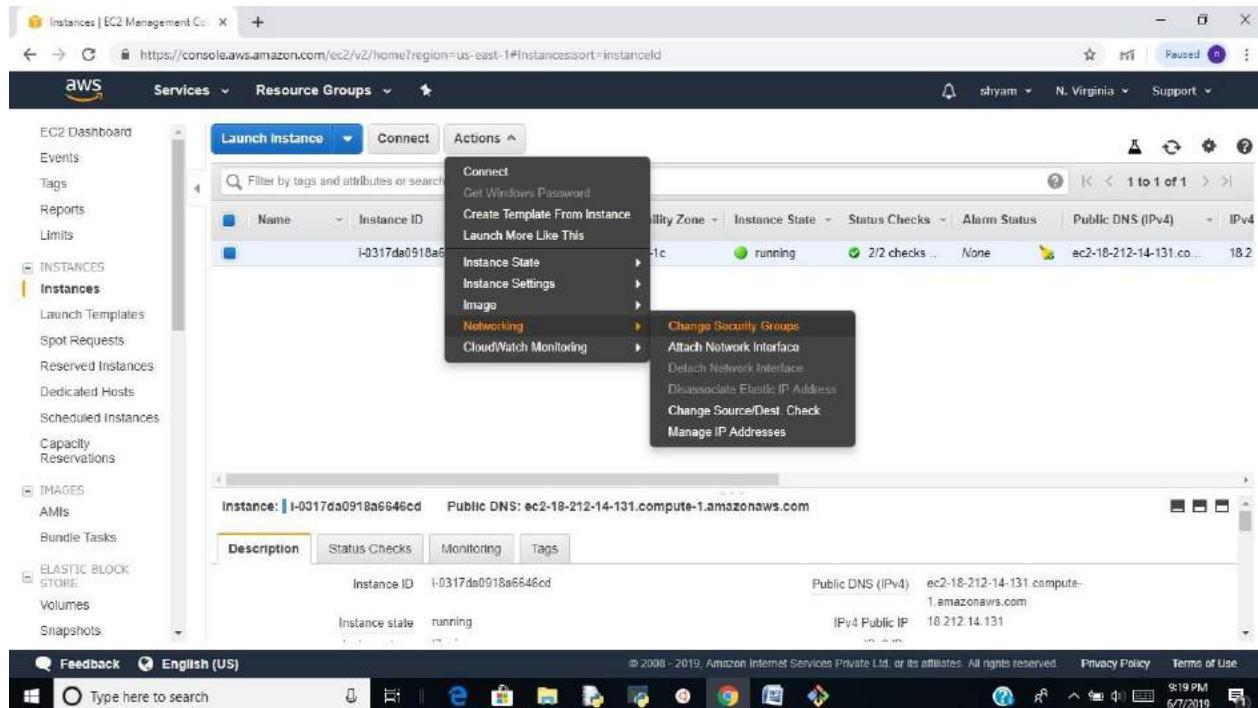


- Enter image name and click on Create Image



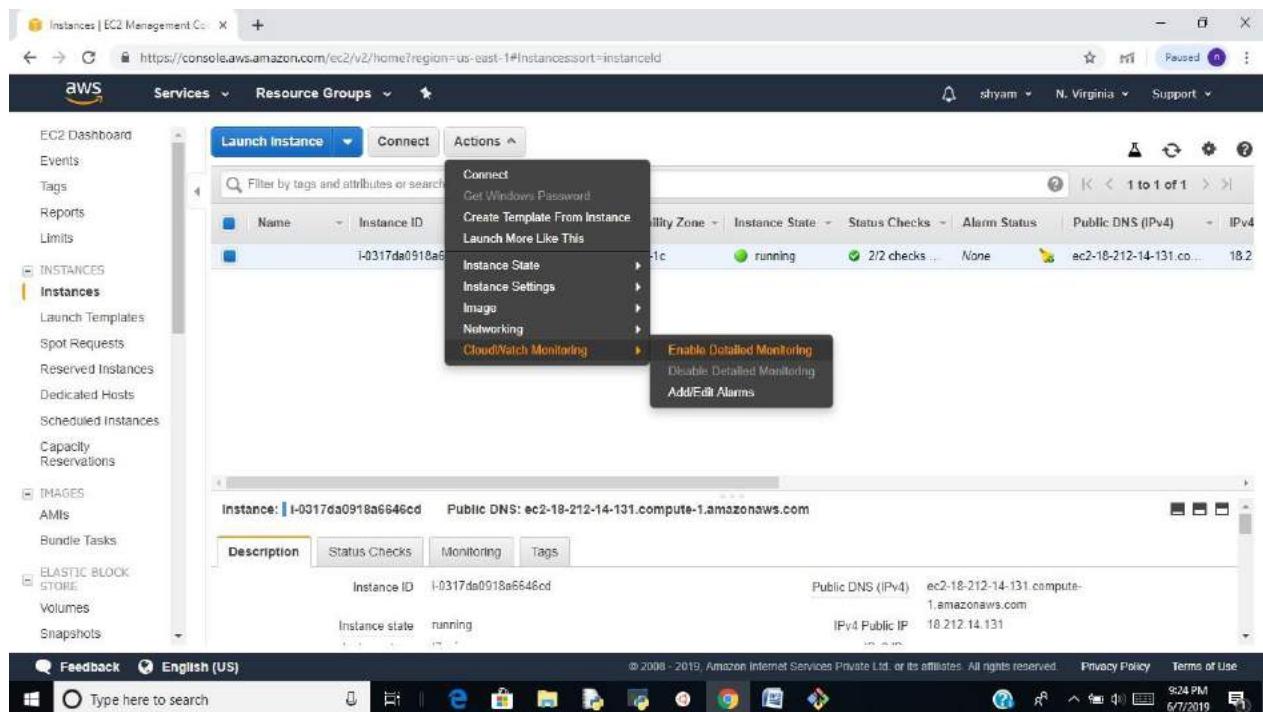
Change security group

- Select instance goto actions and click on Network and select change security group option

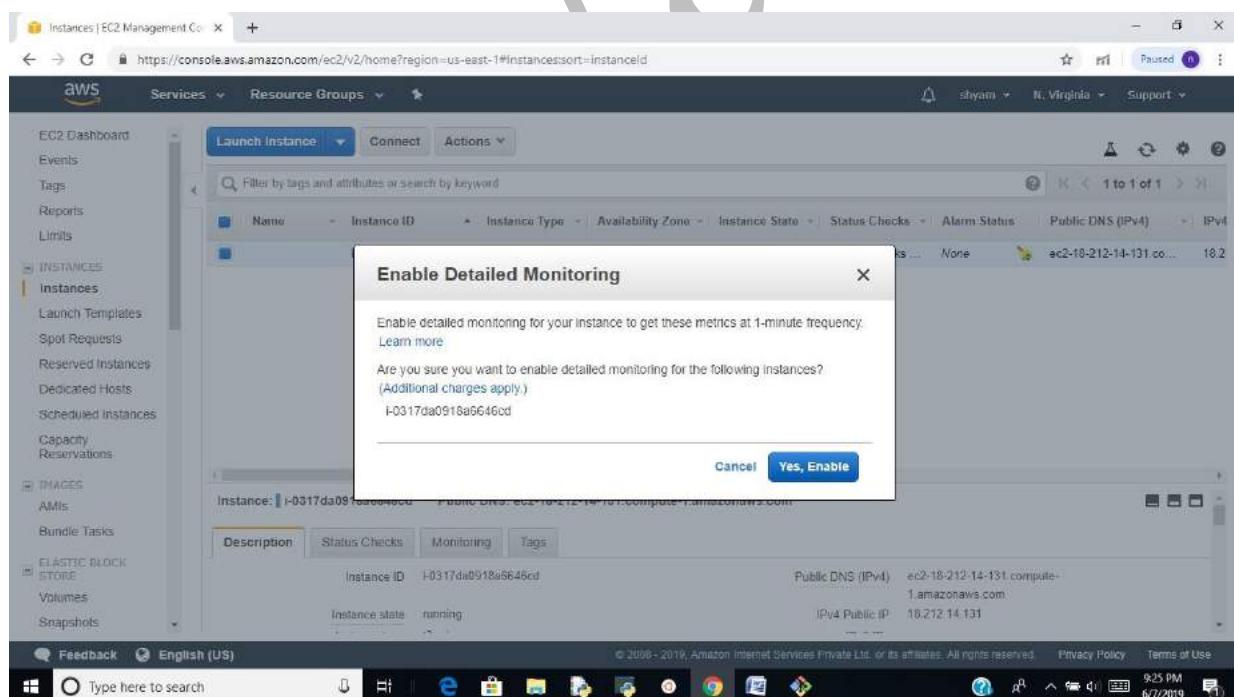


Cloud watch monitoring

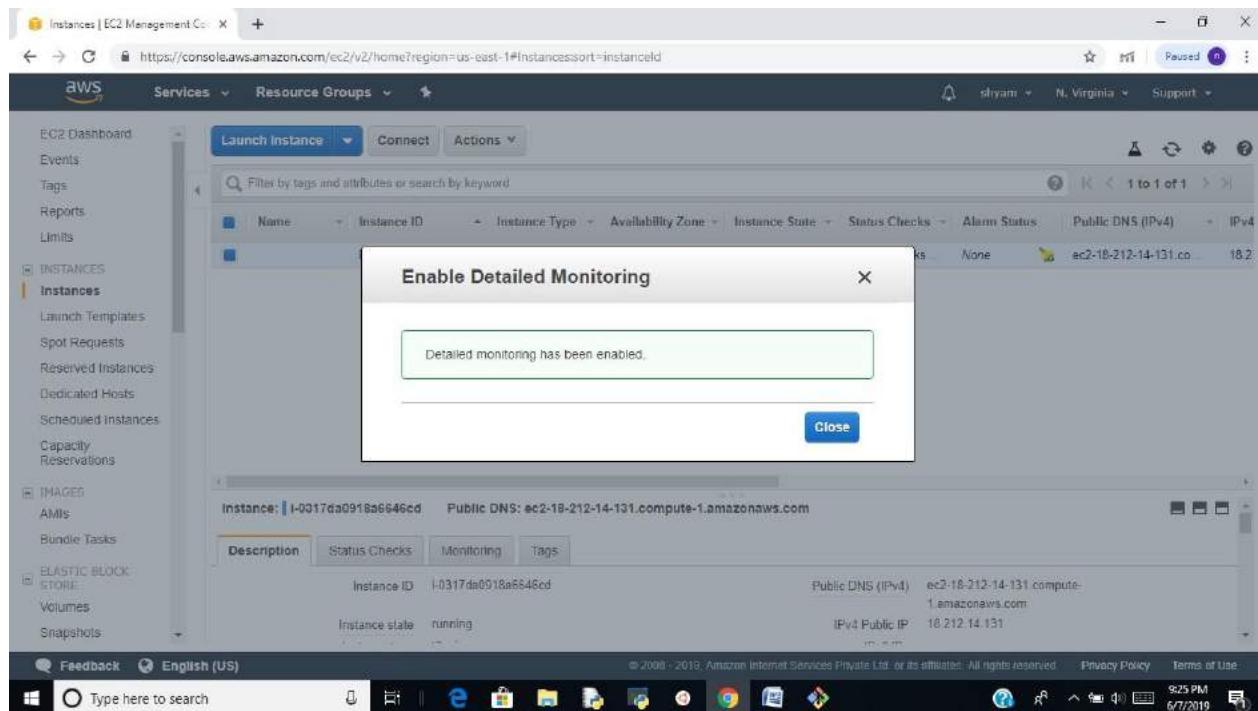
- Select Instance goto actions and click on cloud watch monitoring option and select Enable detailed monitoring



- Click on yes Enable

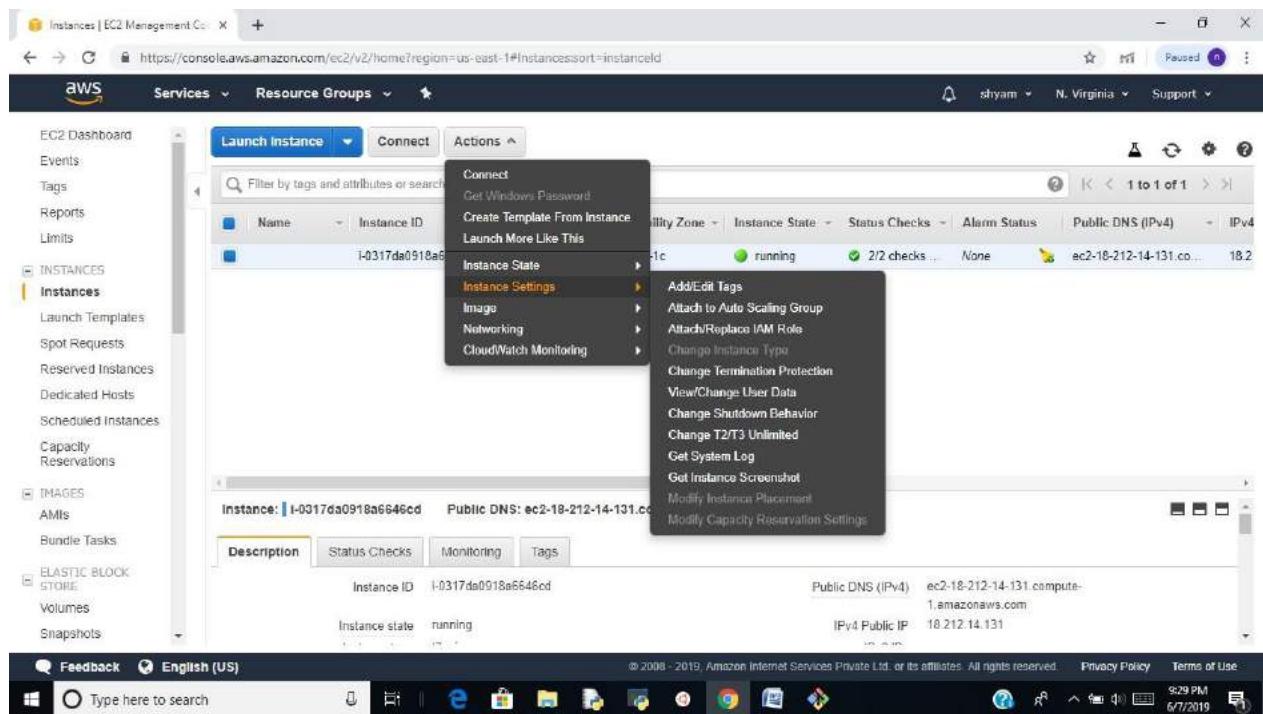


- Click on close



Instance settings

- Instance settings used for attach to auto scaling group, attach/replace IAM role, change termination protection, view or change user data, change shutdown behaviour...etc



- After completion of work terminate the instance by using above terminate option

Elastic Block Storage (EBS)

Amazon Elastic Block Store (Amazon EBS) provides block level storage volumes for use with EC2 instances. EBS volumes are highly available and reliable storage volumes that can be

attached to any running instance that is in the same Availability Zone. EBS volumes that are attached to an EC2 instance are exposed as storage volumes that persist independently from the life of the instance.

Amazon EBS is recommended when data must be quickly accessible and requires long-term persistence. EBS volumes are particularly well-suited for use as the primary storage for file systems, databases, or for any applications that require fine granular updates and access to raw, unformatted, block-level storage. Amazon EBS is well suited to both database-style applications that rely on random reads and writes, and to throughput-intensive applications that perform long, continuous reads and writes.

You can attach multiple volumes to the same instance within the limits specified by your AWS account. Your account has a limit on the number of EBS volumes that you can use, and the total storage available to you.

Amazon EBS Volumes

An Amazon EBS volume is a durable, block-level storage device that you can attach to a single EC2 instance. You can use EBS volumes as primary storage for data that requires frequent updates, such as the system drive for an instance or storage for a database application. You can also use them for throughput-intensive applications that perform continuous disk scans. EBS volumes persist independently from the running life of an EC2 instance.

Amazon EBS Volume Types

Amazon EBS provides the following volume types, which differ in performance characteristics and price, so that you can tailor your storage performance and cost to the needs of your applications.

Solid-State Drives (SSD)

Hard Disk Drives (HDD)

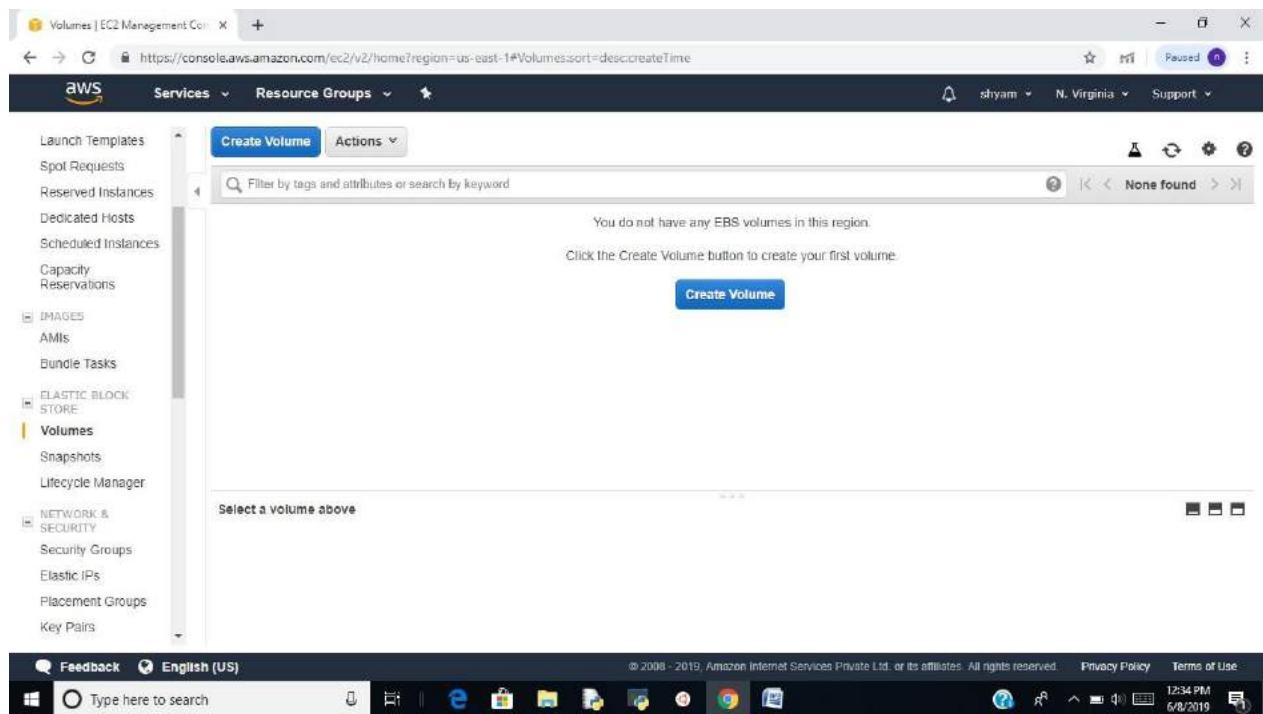
Volume Type	General Purpose SSD (gp2)*	Provisioned IOPS SSD (io1)	Throughput Optimized HDD (st1)	Cold HDD (sc1)
Description	General purpose SSD volume that balances price and performance for a wide variety of workloads	Highest-performance SSD volume for mission-critical low-latency or high-throughput workloads	Low-cost HDD volume designed for frequently accessed, throughput-intensive workloads	Lowest cost HDD volume designed for less frequently accessed workloads
Use Cases	<ul style="list-style-type: none"> • Recommended for most workloads • System boot volumes • Virtual desktops • Low-latency interactive apps • Development and test environments 	<ul style="list-style-type: none"> • Critical business applications that require sustained IOPS performance, or more than 16,000 IOPS or 250 MiB/s of throughput per volume • Large database workloads, such as: <ul style="list-style-type: none"> ◦ MongoDB ◦ Cassandra ◦ Microsoft SQL Server ◦ MySQL ◦ PostgreSQL ◦ Oracle 	<ul style="list-style-type: none"> • Streaming workloads requiring consistent, fast throughput at a low price • Big data warehouses • Log processing • Cannot be a boot volume 	<ul style="list-style-type: none"> • Throughput-oriented storage for large volumes of data that is infrequently accessed • Scenarios where the lowest storage cost is important • Cannot be a boot volume

API Name	gp2	io1	st1	sc1
Volume Size	1 GiB - 16 TiB	4 GiB - 16 TiB	500 GiB - 16 TiB	500 GiB - 16 TiB
Max. IOPS**/Volume	16,000***	64,000****	500	250
Max. Throughput/Volume	250 MiB/s***	1,000 MiB/s†	500 MiB/s	250 MiB/s
Max. IOPS/Instance††	80,000	80,000	80,000	80,000
Max. Throughput/Instance††	1,750 MiB/s	1,750 MiB/s	1,750 MiB/s	1,750 MiB/s
Dominant Performance Attribute	IOPS	IOPS	MiB/s	MiB/s

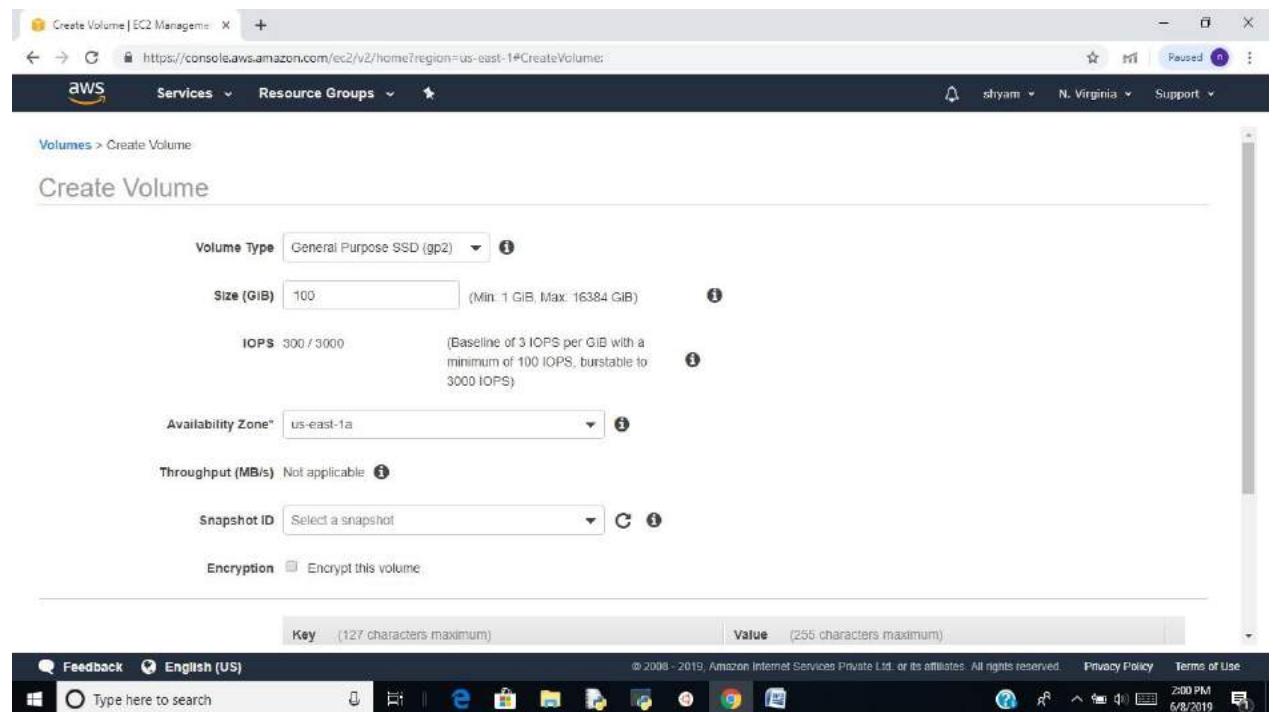
* Default volume type for EBS volumes created from the console is gp2. Volumes created using the CreateVolume API without a volume-type argument default to either gp2 or standard according to region.

Create volume

- Select Elastic Block Storage section on left side panel of EC2 service and click on volumes
- Click on create volume



- Volume Type: select general purpose SSD (gp2)
- Size: enter size for volume
- IOPS: 100/300
- Availability Zone: select availability zone
- Snapshot ID: enter snapshot id for create volume
- Encrypt this volume: use Encryption for security you don't was disable
- Add key value pair tag to identification and it is optional
- Click on create volume



The screenshot shows the 'Create Volume' page in the AWS EC2 Management console. The volume type is set to 'General Purpose SSD (gp2)'. The size is 100 GiB. IOPS is 300 / 3000. Availability Zone is us-east-1a. Throughput (MB/s) is not applicable. A snapshot ID dropdown is set to 'Select a snapshot'. An encryption checkbox is checked. Below the form is a key-value pair section for tags.

Key: (127 characters maximum) **Value:** (255 characters maximum)

IOPS: 300 / 3000 (Baseline of 3 IOPS per GiB with a minimum of 100 IOPS, burstable to 3000 IOPS)

Availability Zone*: us-east-1a

Throughput (MB/s): Not applicable

Snapshot ID: Select a snapshot

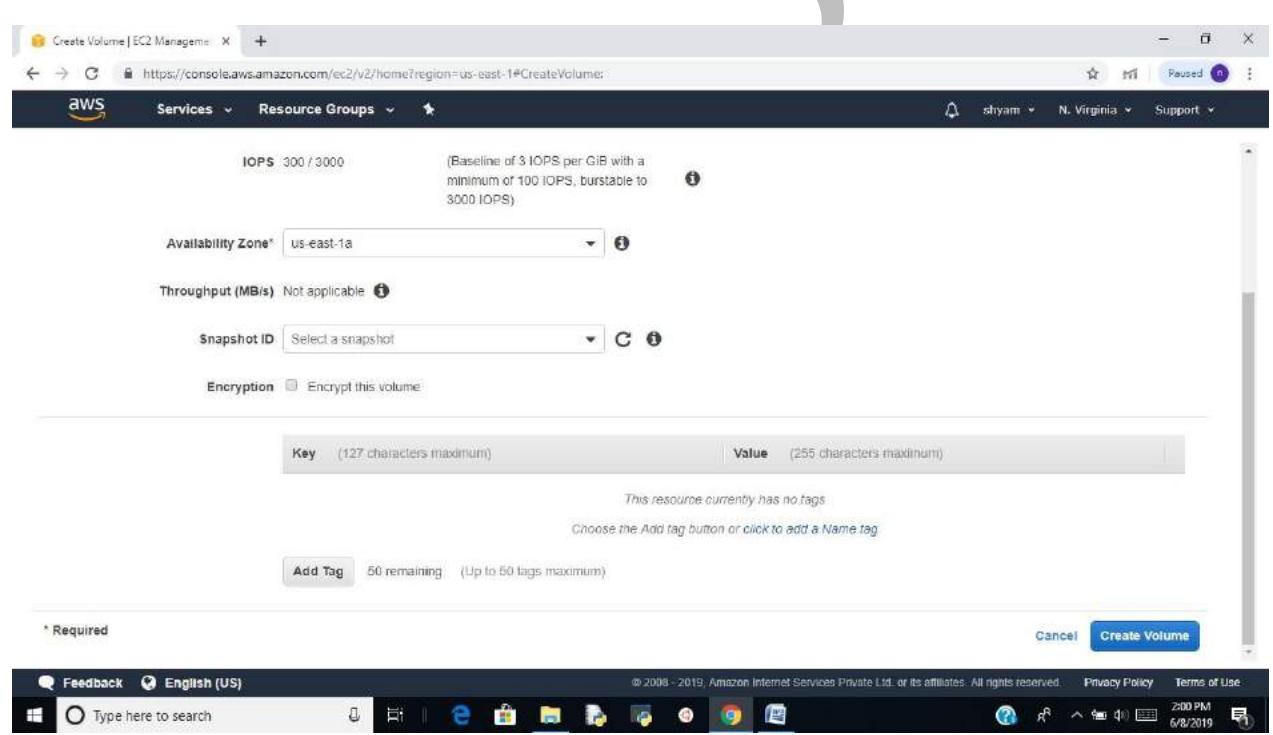
Encryption: Encrypt this volume

Tags

Key	(127 characters maximum)	Value	(255 characters maximum)
This resource currently has no tags.			
Choose the Add tag button or click to add a Name tag.			
Add Tag	50 remaining (Up to 50 tags maximum)		

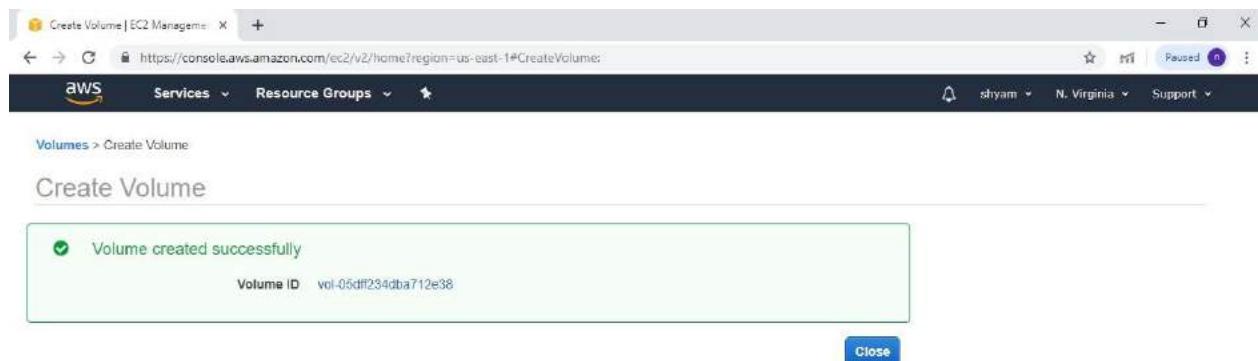
Required

Create Volume



This screenshot shows the same 'Create Volume' page, but the 'Tags' section is more prominent. It displays a message stating 'This resource currently has no tags.' and 'Choose the Add tag button or click to add a Name tag.' Below this is a 'Add Tag' button and a note about the tag limit. The 'Create Volume' button is at the bottom right.

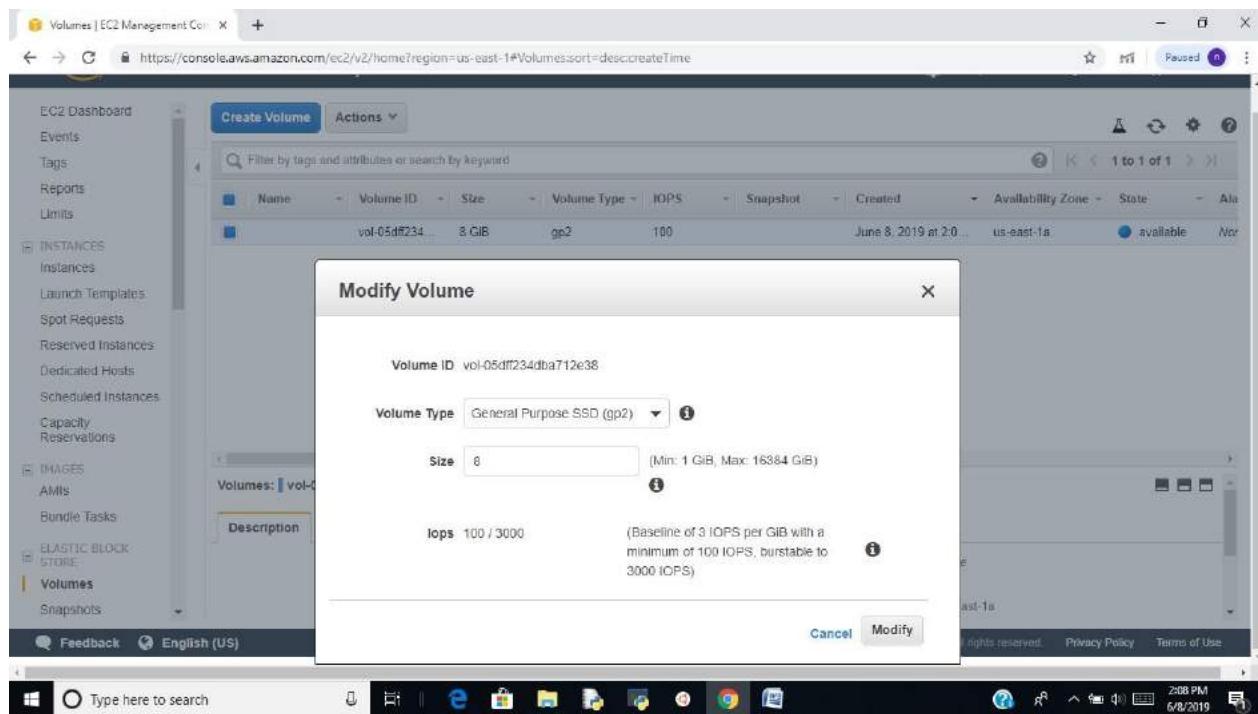
Click on close



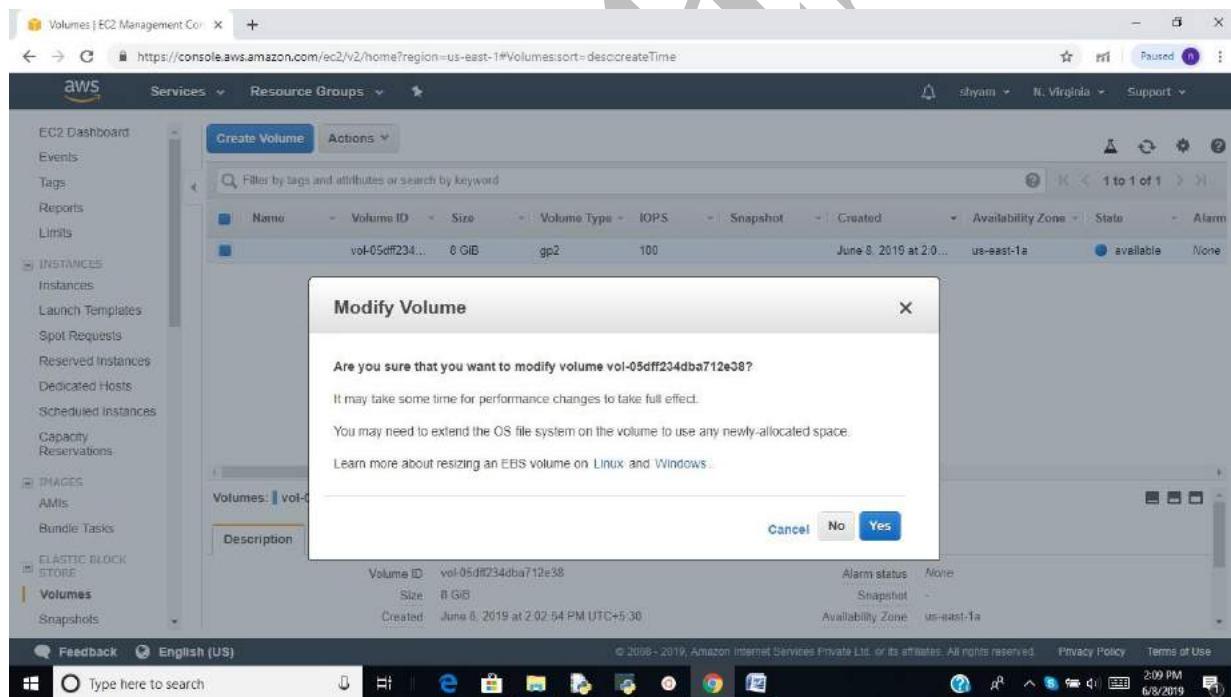
Modify Volume

- Select particular volume in volumes section and go to actions and click on modify volume
- Change volume type and size by required volumes and click on Modify

Amazon Web Services

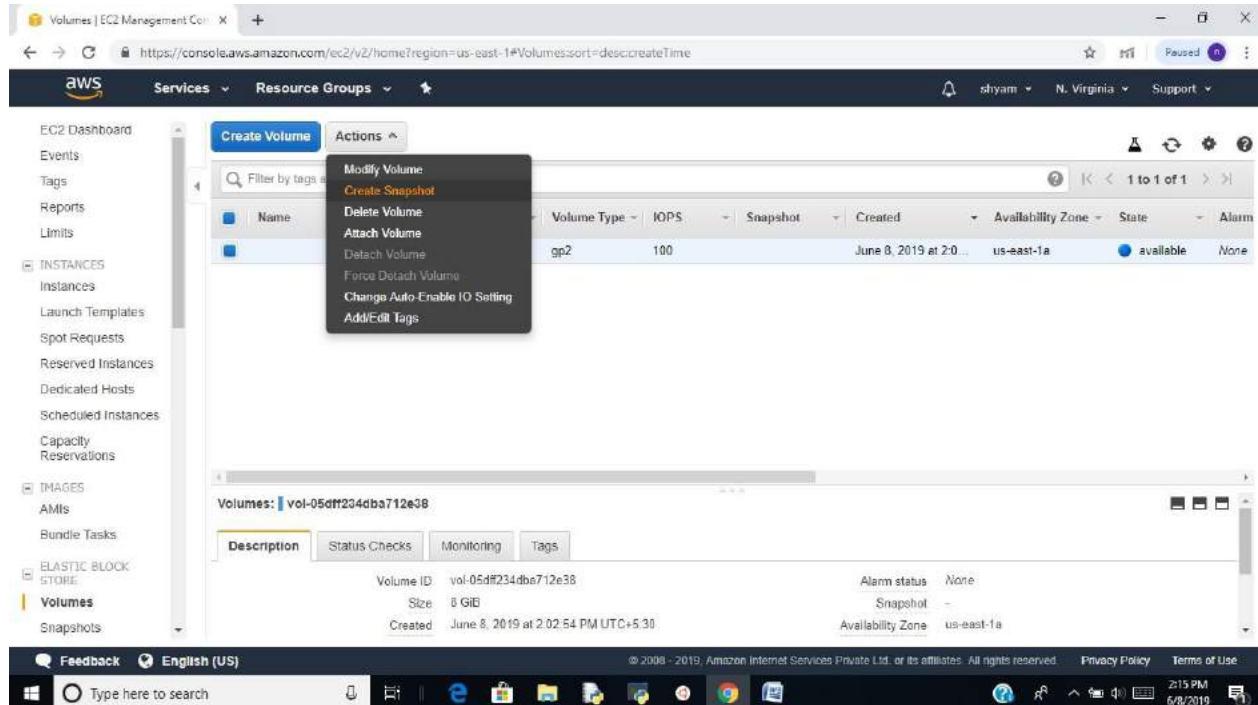


Click on yes

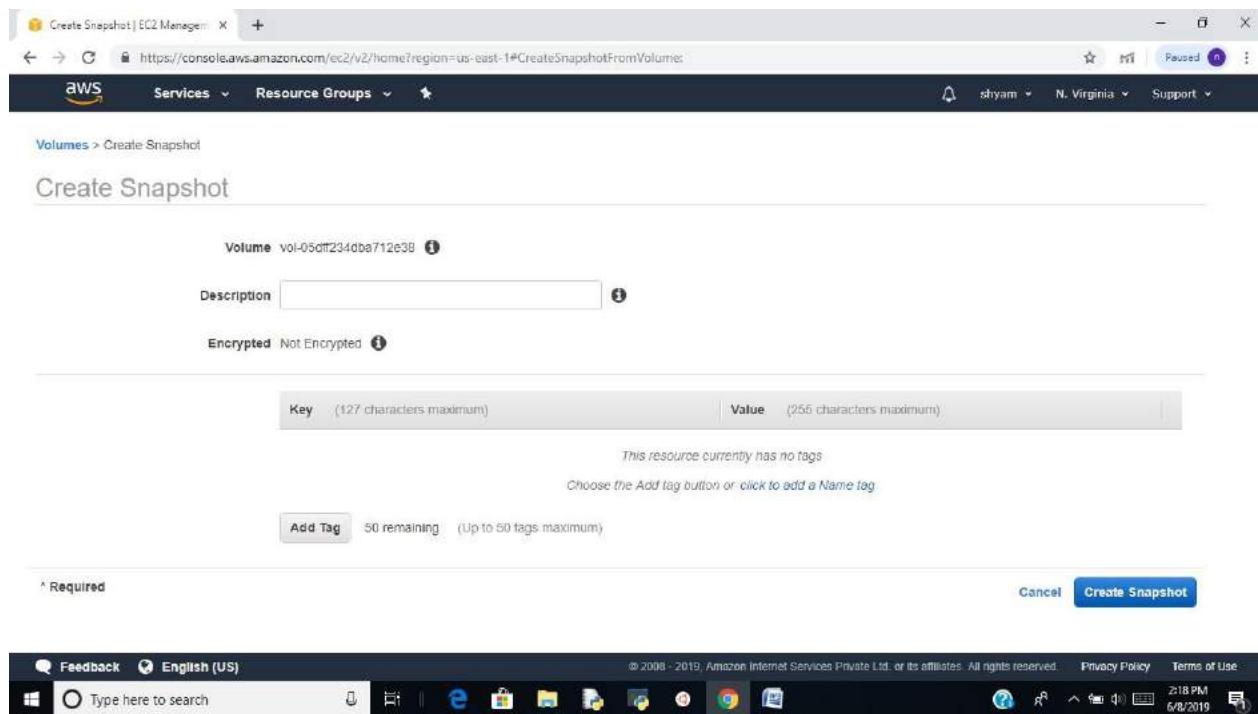


Create snapshot

- Select particular volume on volume section and goto actions and click on create snapshot option`

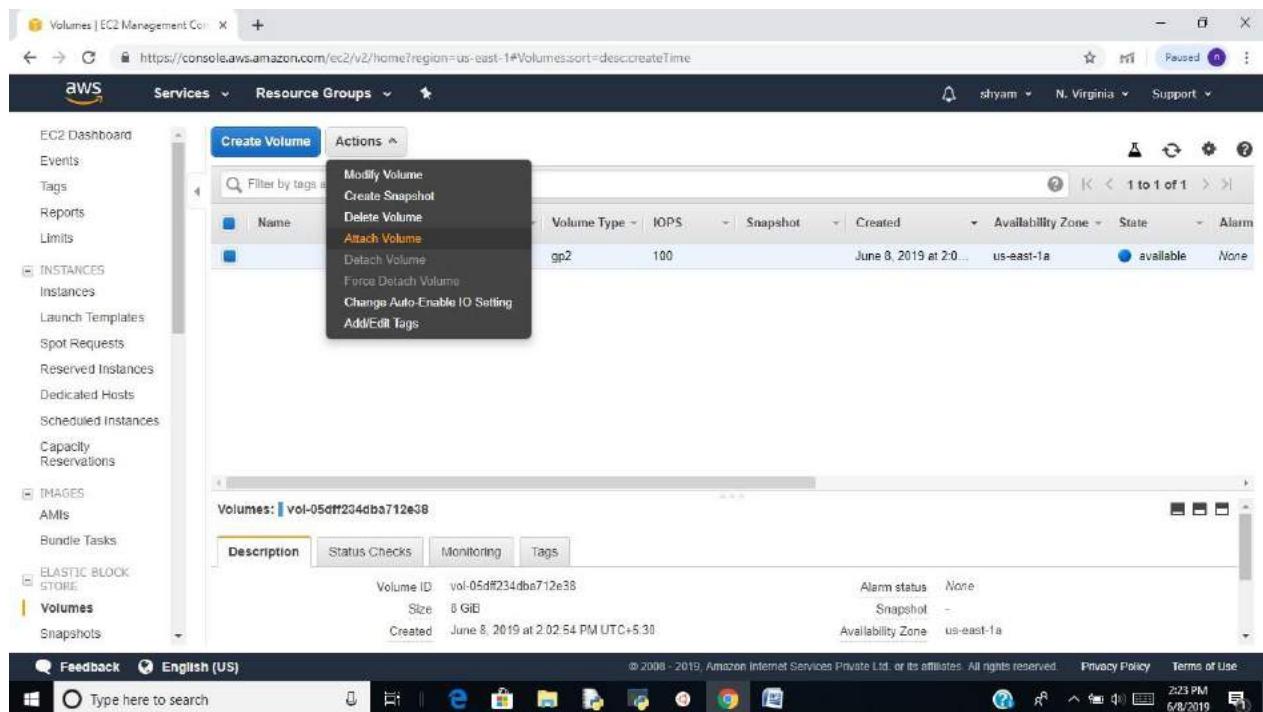


- Description: Enter description and it is the optional value
- Click on create snapshot



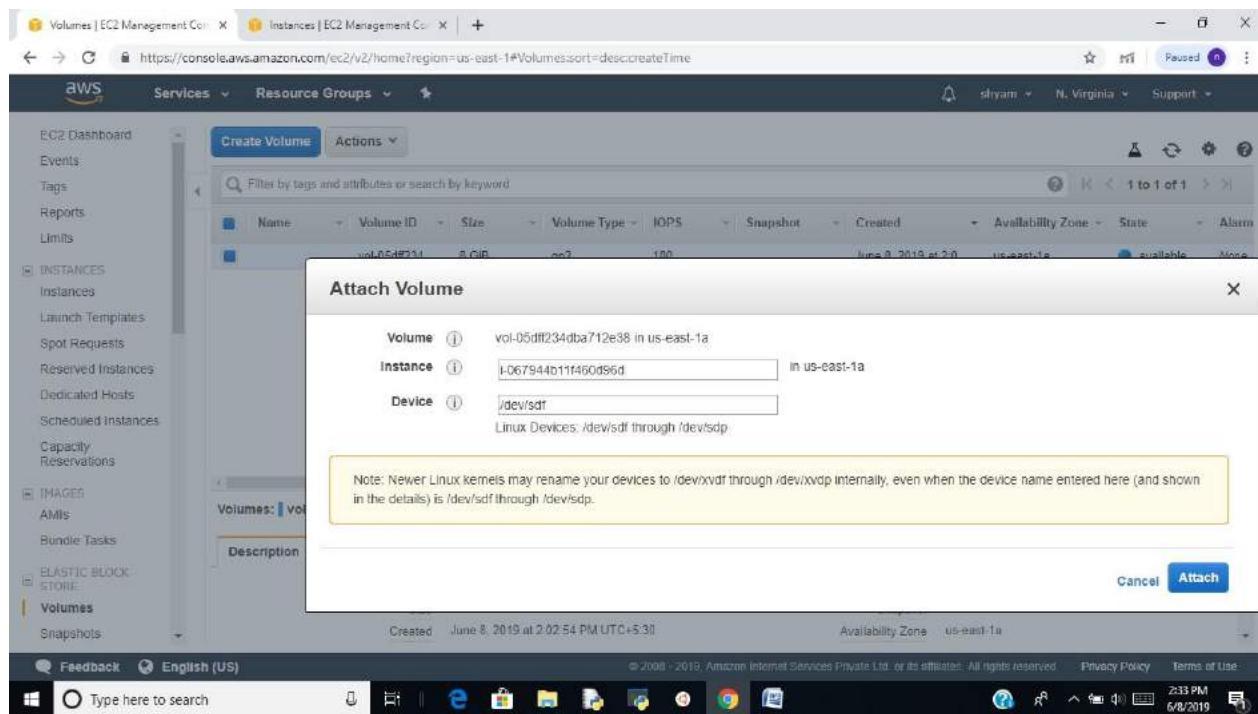
Attach Volume

- Select volume and goto actions and click on attach volume option



- Instance: select instance id to attach this volume
- Device : enter device that is /dev/sdf or /dev/sdp
- Click on attach

Note: Both instance and volume must be in same availability zone to attach volume to instance



Mount the volume

- Connect to EC2 instance and check this volume using **lsblk** command

```

Elastic Block Storage (Autosaved) - Microsoft Word

0 packages can be updated.
0 updates are security updates.

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-42-68:~$ lsblk
NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
xvda 202:0 0 8G 0 disk
└─xvda1 202:1 0 8G 0 part /
xvdf 202:80 0 8G 0 disk
loop0 7:0 0 89.4M 1 loop /snap/core/6818
loop1 7:1 0 18M 1 loop /snap/amazon-ssm-agent/1335
ubuntu@ip-172-31-42-68:~$

Page: 10 of 11 Words: 795
2:38 PM 6/8/2019

```

- Convert the volume into ext4 file system using below command
- Sudo mkfs -t ext4 /dev/xvdf

```

Elastic Block Storage (Autosaved) - Microsoft Word

Get:19 http://security.ubuntu.com/ubuntu xenial-security/main amd64 Packages [66
9 kB]
Get:20 http://security.ubuntu.com/ubuntu xenial-security/main Translation-en [27
0 kB]
Get:21 http://security.ubuntu.com/ubuntu xenial-security/universe amd64 Packages
[438 kB]
Get:22 http://security.ubuntu.com/ubuntu xenial-security/universe Translation-en
[178 kB]
Get:23 http://security.ubuntu.com/ubuntu xenial-security/multiverse amd64 Packag
es [5,600 B]
Get:24 http://security.ubuntu.com/ubuntu xenial-security/multiverse Translation-
en [2,676 B]
Fetched 16.5 MB in 3s (5,003 kB/s)
Reading package lists... Done
ubuntu@ip-172-31-42-68:~$ lsblk
NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
xvda 202:0 0 8G 0 disk
└─xvda1 202:1 0 8G 0 part /
xvdf 202:80 0 8G 0 disk
loop0 7:0 0 89.4M 1 loop /snap/core/6818
loop1 7:1 0 18M 1 loop /snap/amazon-ssm-agent/1335
ubuntu@ip-172-31-42-68:~$ sudo file -s /dev/xvdf
/dev/xvdf: data
ubuntu@ip-172-31-42-68:~$ sudo mkfs -t ext4 /dev/xvdf
ubuntu@ip-172-31-42-68:~$ 

Page: 11 of 16 Words: 925
3:19 PM 6/8/2019

```

- Create one directory using below command

Mkdir ev

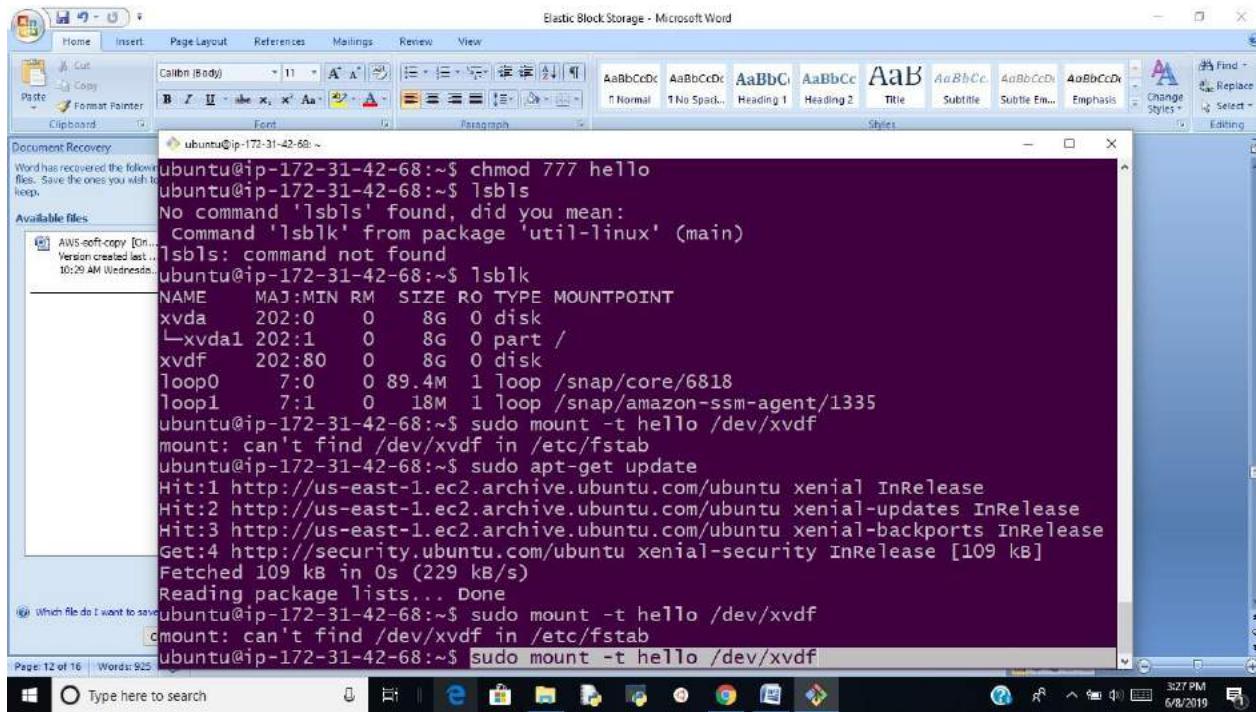


A screenshot of a Microsoft Word document titled "Elastic Block Storage (Autosaved) - Microsoft Word". The document contains a terminal session window with the following content:

```
ubuntu@ip-172-31-42-68:~$ 0 packages can be updated.  
0 updates are security updates.  
  
The programs included with the Ubuntu system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*copyright.  
  
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by  
applicable law.  
  
To run a command as administrator (user "root"), use "sudo <command>".  
See "man sudo_root" for details.  
  
ubuntu@ip-172-31-42-68:~$ lsblk  
NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINT  
xvda 202:0 0 8G 0 disk  
└─xvda1 202:1 0 8G 0 part /  
xvdf 202:80 0 8G 0 disk  
loop0 7:0 0 89.4M 1 loop /snap/core/6818  
loop1 7:1 0 18M 1 loop /snap/amazon-ssm-agent/1335  
ubuntu@ip-172-31-42-68:~$ mkdir ev
```

- Mount this volume with this directory using below command

Sudo mount –t ev /dev/xvdf1



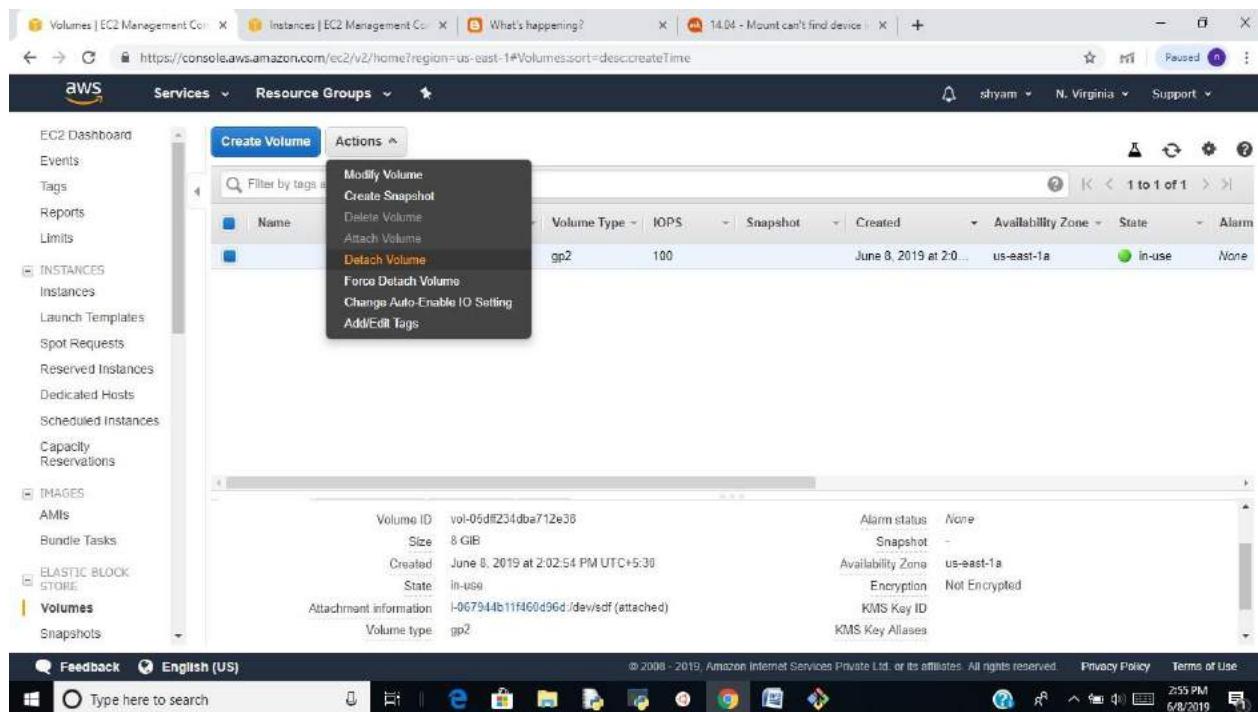
The screenshot shows a Microsoft Word document titled "Elastic Block Storage - Microsoft Word". The document contains a terminal session from an Ubuntu system. The terminal output is as follows:

```
ubuntu@ip-172-31-42-68:~$ chmod 777 hello
ubuntu@ip-172-31-42-68:~$ lsblk
No command 'lsblk' found, did you mean:
Command 'lsblk' from package 'util-linux' (main)
lsblk: command not found
ubuntu@ip-172-31-42-68:~$ lsblk
NAME   MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
xvda   202:0    0   8G  0 disk
└─xvda1 202:1    0   8G  0 part /
xvdf   202:80   0   8G  0 disk
loop0   7:0     0  89.4M 1 loop /snap/core/6818
loop1   7:1     0  18M  1 loop /snap/amazon-ssm-agent/1335
ubuntu@ip-172-31-42-68:~$ sudo mount -t hello /dev/xvdf
mount: can't find /dev/xvdf in /etc/fstab
ubuntu@ip-172-31-42-68:~$ sudo apt-get update
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu xenial InRelease
Hit:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu xenial-updates InRelease
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu xenial-backports InRelease
Get:4 http://security.ubuntu.com/ubuntu xenial-security InRelease [109 kB]
Fetched 109 kB in 0s (229 kB/s)
Reading package lists... Done
@ Which file do I want to save?
ubuntu@ip-172-31-42-68:~$ sudo mount -t hello /dev/xvdf
mount: can't find /dev/xvdf in /etc/fstab
ubuntu@ip-172-31-42-68:~$ sudo mount -t hello /dev/xvdf
```

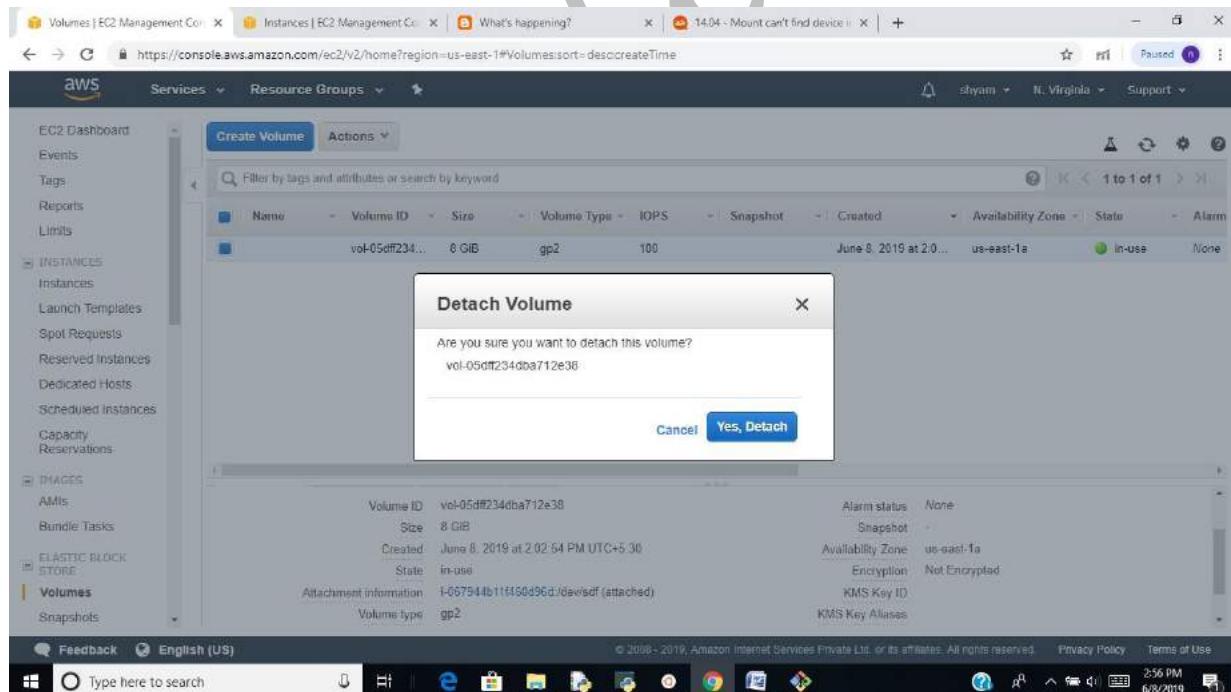
The Word document interface includes a ribbon bar with Home, Insert, Page Layout, References, Mailings, Review, and View tabs. The ribbon bar also includes various font and paragraph styles. A status bar at the bottom shows "Page: 12 of 16" and "Words: 925". The taskbar at the bottom of the screen shows several icons, including the Start button, a search bar, and icons for File Explorer, Edge browser, File Explorer, Task View, and others.

Detach volume

- Select the volume and goto actions and click on detach volume option



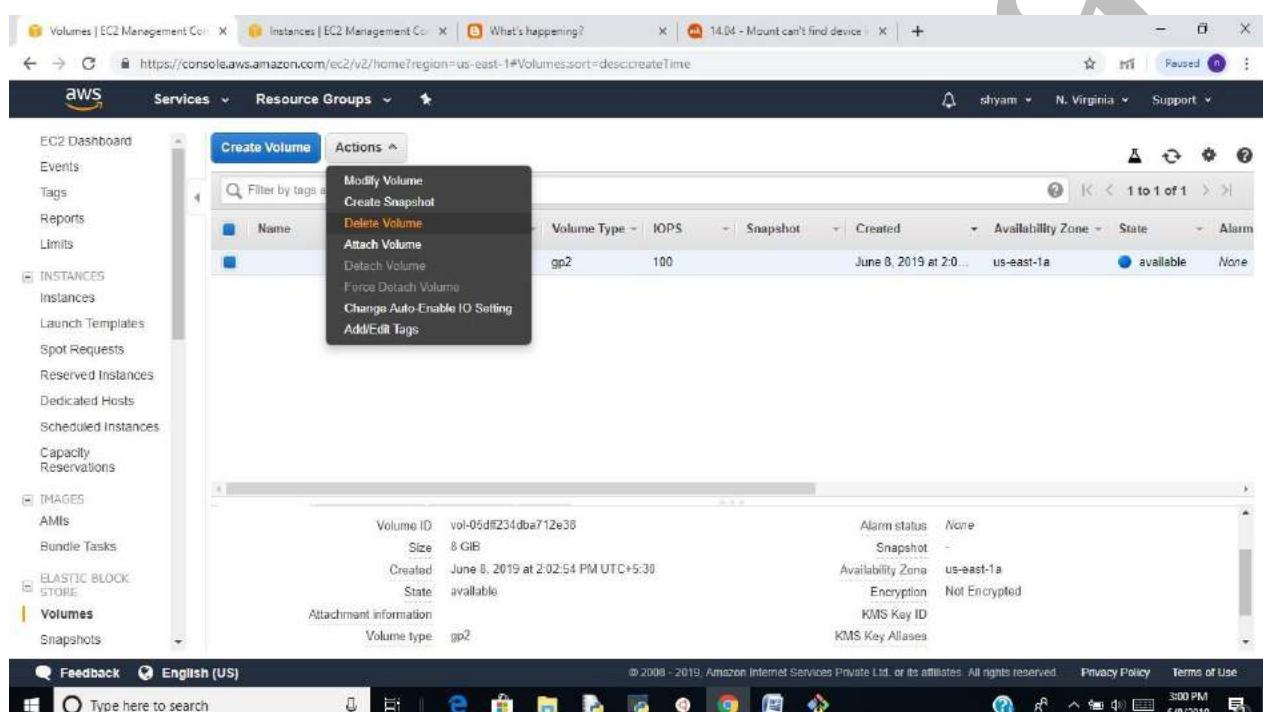
Click on yes, detach



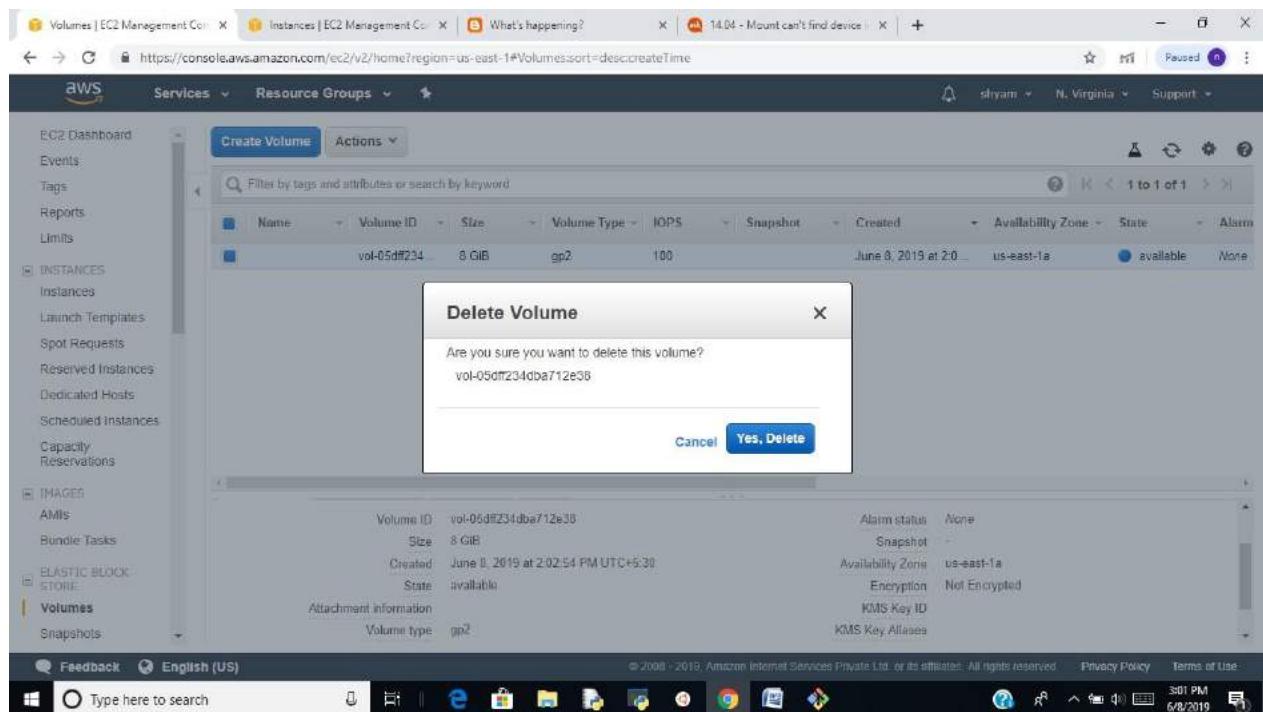
- Before detach umount the volume from ec2 instance terminal
- Connect to EC2 instance and use below command to umount
Sudo umount /dev/xvdf

Delete Volume

- Select volume and goto actions and click on delete volume option



- Click on yes delete



Note: root volume you not delete directly when instance terminated it is automatically deleted or first stop the instance and after you detach and delete the volume

Elastic File System (EFS)

Amazon Elastic File System (Amazon EFS) provides simple, scalable file storage for use with Amazon EC2. With Amazon EFS, storage capacity is elastic, growing and shrinking automatically as you add and remove files, so your applications have the storage they need, when they need it. Amazon EFS has a simple web services interface that allows you to create and configure file systems quickly and easily. The service manages all the file storage infrastructure for you, meaning that you can avoid the complexity of deploying, patching, and maintaining complex file system configurations.

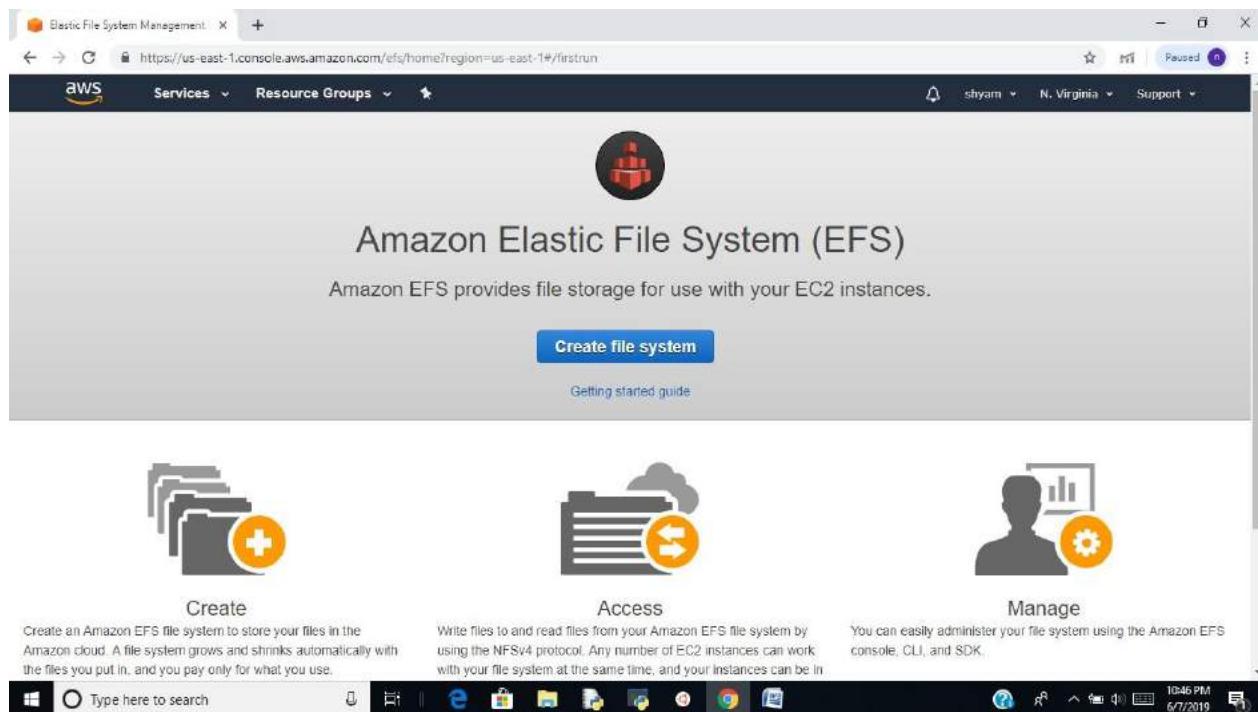
Amazon EFS supports the Network File System version 4 (NFSv4.1 and NFSv4.0) protocol, so the applications and tools that you use today work seamlessly with Amazon EFS. Multiple Amazon EC2 instances can access an Amazon EFS file system at the same time, providing a common data source for workloads and applications running on more than one instance or server.

Launch one EC2 instance

- Launch the ec2 instance by choosing AMI, instance type, vpc, subnet, security group and key pair as like above launch ec2 instance in EC2 section

Create EFS

- Select EFS service on AWS console and click on create file system



- Vpc: select vpc in which vpc above ec2 instance is launched
- Create mount Targets: select all availability zones to access file system from entire vpc
- Click on Next step

An Amazon EFS file system is accessed by EC2 instances running inside one of your VPCs. Instances connect to a file system by using a network interface called a mount target. Each mount target has an IP address, which we assign automatically or you can specify.

VPC: vpc-97ab3fed (default)

	Availability Zone	Subnet	IP address	Security groups
✓	us-east-1a	subnet-10e9ac4c (default)	Automatic	sg-32f6679 - default
✓	us-east-1b	subnet-1c1b97b (default)	Automatic	sg-32f6679 - default
✓	us-east-1c	subnet-0d74b3a3 (default)	Automatic	sg-32f6679 - default
✓	us-east-1d	subnet-09ebcb443 (default)	Automatic	sg-32f6679 - default
✓	us-east-1e	subnet-a3e35c9d (default)	Automatic	sg-32f6679 - default
✓	us-east-1f	subnet-aa92a0a5 (default)	Automatic	sg-32f6679 - default

Next Step

- Add tags, life cycle management, throughput mode and encryption all are the optional values do want use enable this options for this EFS
- Click on Next step

Create file system

Step 1: Configure file system access

Step 2: Configure optional settings (selected)

Step 3: Review and create

Configure optional settings

Add tags

You can add tags to describe your file system. A tag consists of a case-sensitive key-value pair. (For example, you can define a tag with key-value pair with key = Corporate Department and value = Sales and Marketing.) At a minimum, we recommend a tag with key = Name.

Key	Value	Remove
Name	Add New Value	X
Add New Key		

Enable lifecycle management NEW!

Automatically save up to 85% on your EFS bill as your access patterns change by enabling **Lifecycle Management** for your file system. Using a predefined lifecycle policy, any files in your file system that are not accessed for thirty (30) days will automatically move to the EFS Infrequent Access (EFS IA) storage class. EFS IA provides price/performance that's cost-optimized for files not accessed every day. Learn more

Enable Lifecycle Management

- Check all configuration details for EFS and click create EFS

Step 1: Configure file system access

Step 2: Configure optional settings

Step 3: Review and create

Review and create

Review the configuration below before proceeding to create your file system.

File system access

VPC	Availability Zone	Subnet	IP address	Security groups
vpc-97ab3fed (default)	us-east-1a	subnet-10e9ac4c (default)	Automatic	sg-3f2f6679 - default
	us-east-1b	subnet-1cf1897b (default)	Automatic	sg-3f2f6679 - default
	us-east-1c	subnet-8df4b3a3 (default)	Automatic	sg-3f2f6679 - default
	us-east-1d	subnet-09bcb443 (default)	Automatic	sg-3f2f6679 - default
	us-east-1e	subnet-a3e36c9d (default)	Automatic	sg-3f2f6679 - default
	us-east-1f	subnet-aa92a0a5 (default)	Automatic	sg-3f2f6679 - default

Optional settings

Tags: No tags added

VPC	Availability Zone	Subnet	IP address	Security groups
vpc-97ab3fed (default)	us-east-1a	subnet-10e9ac4c (default)	Automatic	sg-3f2f6679 - default
	us-east-1b	subnet-1cf1897b (default)	Automatic	sg-3f2f6679 - default
	us-east-1c	subnet-8df4b3a3 (default)	Automatic	sg-3f2f6679 - default
	us-east-1d	subnet-09bcb443 (default)	Automatic	sg-3f2f6679 - default
	us-east-1e	subnet-a3e36c9d (default)	Automatic	sg-3f2f6679 - default
	us-east-1f	subnet-aa92a0a5 (default)	Automatic	sg-3f2f6679 - default

Optional settings

Tags: No tags added

Performance mode: General Purpose

Throughput mode: Bursting

Encrypted: No

Lifecycle policy: None

Create File System

- EFS is created

The screenshot shows the AWS Elastic File System Management console. The URL in the address bar is <https://us-east-1.console.aws.amazon.com/efs/home?region=us-east-1#/filesystems/fs-17e779f4>. The page displays a success message: "You have created a file system. You can mount your file system from an EC2 instance with an NFSv4.1 client installed. You can also mount your file system from an on-premises server over an AWS Direct Connect or AWS VPN connection. Click here for EC2 mount instructions, and here for on-premises mount instructions." Below this, there is a table with one row showing the newly created file system details:

	Name	File system ID	Metered size	Number of mount targets	Creation date
fs-17e779f4	fs-17e779f4	6.0 KB	6	06/07/2019, 17:52:46 UTC	

Below the table, there is a section for "Other details" with the following information:

- Owner ID: 814927698004
- File system state: Available
- Performance mode: General Purpose
- Throughput mode: Bursting
- Encrypted: No
- Lifecycle policy: None

The browser's status bar at the bottom right shows the time as 11:27 PM and the date as 6/7/2019.

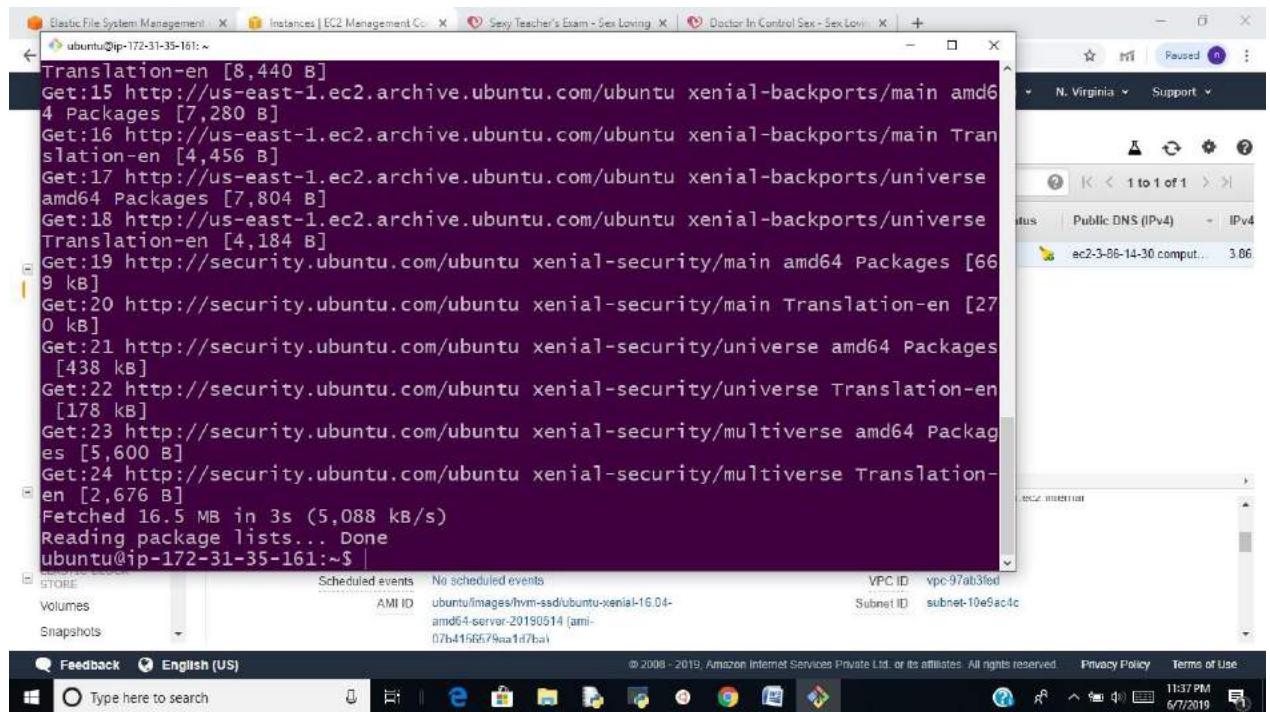
DNS name: fs-17e779f4.efs.us-east-1.amazonaws.com

Amazon EC2 mount instructions (from local VPC)
Amazon EC2 mount instructions (across VPC peering connection)
On-premises mount instructions

VPC	Availability Zone	Subnet	IP address	Mount target ID	Network interface ID	Security groups	Mount target state
vpc-97ab3fed (default)	us-east-1c	subnet-0d74b0a3 (default)	172.31.67.57	fsmt-413e07a0	eni-0e3f667c7fd4bf85e		Creating
	us-east-1e	subnet-a3e36c9d (default)	172.31.79.218	fsmt-423e07a3	eni-01ff2b846c43fc1b8		Creating
	us-east-1f	subnet-aa92a0a6 (default)	172.31.67.247	fsmt-443e07a5	eni-0e859e8565800b812		Creating
	us-east-1d	subnet-09bcb443 (default)	172.31.27.81	fsmt-463e07a7	eni-0debb343b919ddff90		Creating
	us-east-1a	subnet-10e9ac4c (default)	172.31.45.125	fsmt-7c3e079d	eni-0ff7fa13e6543067		Creating
	us-east-1b	subnet-1cf1897b (default)	172.31.1.62	fsmt-7e3e079f	eni-0f30e2f6facc3cd29		Creating

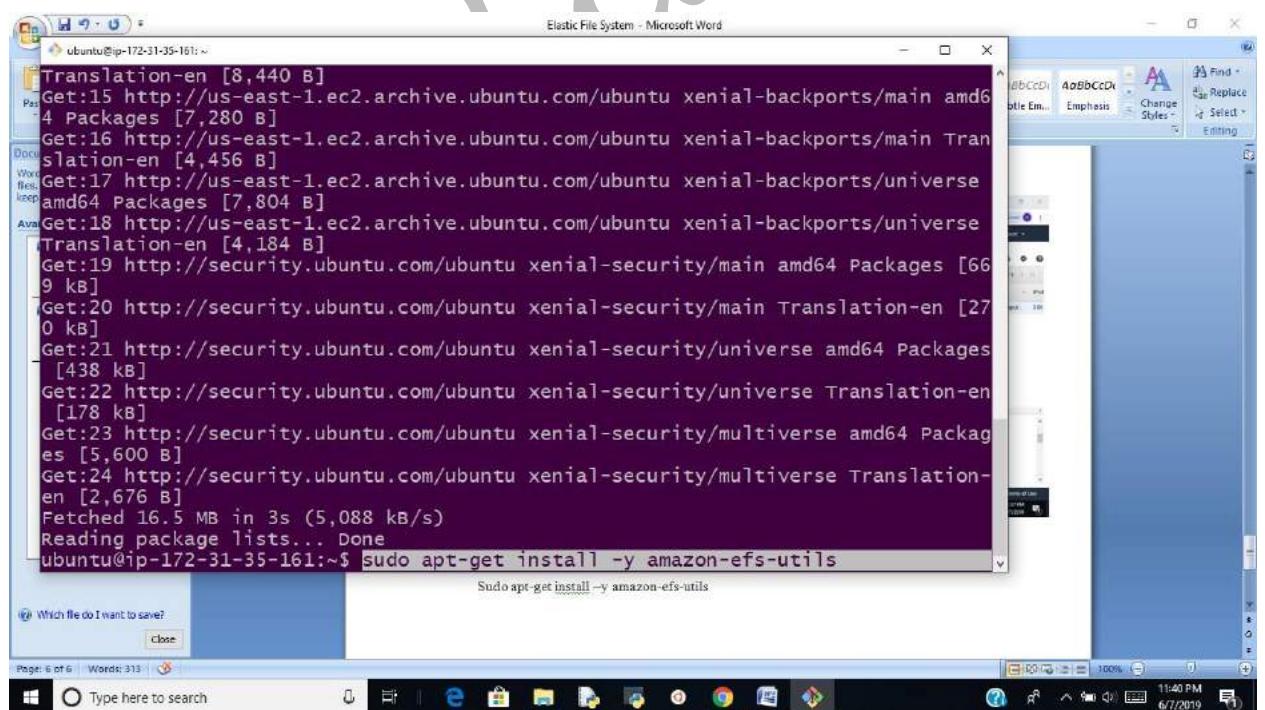
To connect to your Amazon EC2 instance and mount the Amazon EFS file system

- Connect to EC2 instance using gitbash or putty with ssh protocol like above



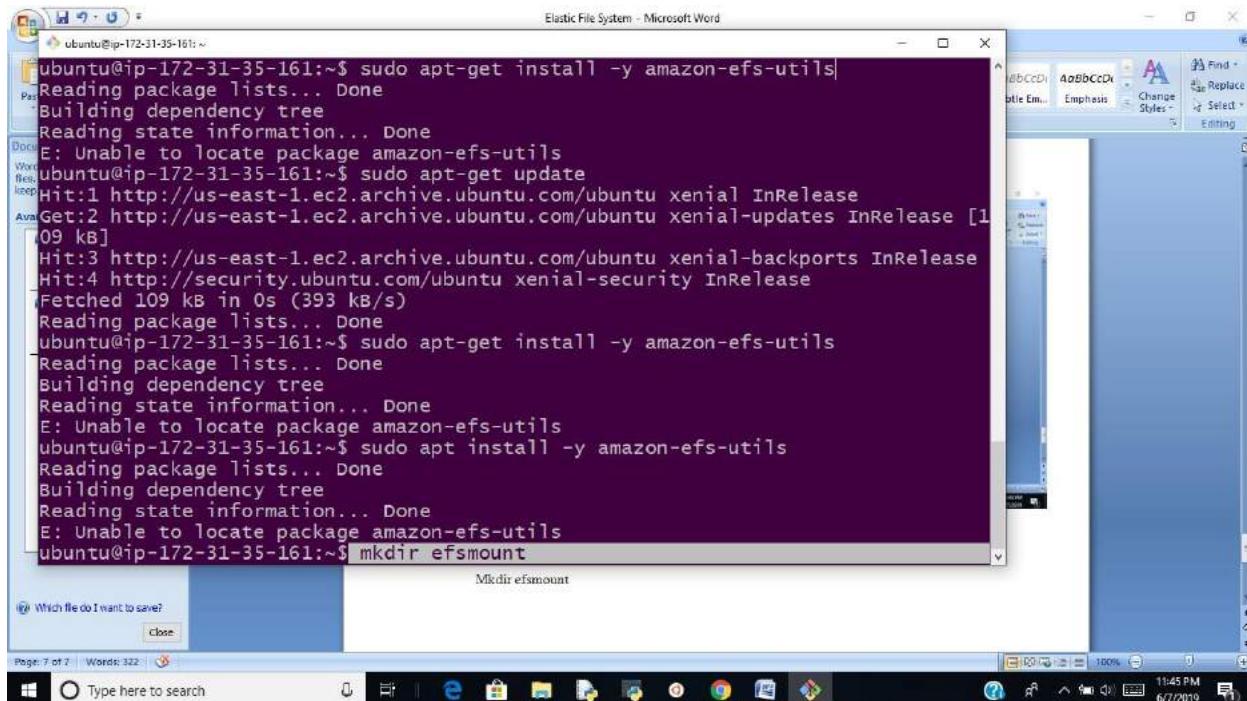
- Install amazon-efs-utils with below command

Sudo apt-get install -y amazon-efs-utils



- Create one directory using below command

Mkdir efsmount



A screenshot of a Windows desktop environment. In the center, there is a terminal window titled "Elastic File System - Microsoft Word" showing the output of a terminal session. The session starts with commands to install the Amazon EFS utilities, which fail because the package is not found. It then shows an attempt to update the package lists and install the utilities again, which also fails. Finally, the command "mkdir efsmount" is run successfully. To the right of the terminal is a Microsoft Word document window. At the bottom of the screen is a taskbar with various icons and a search bar.

```
ubuntu@ip-172-31-35-161:~$ sudo apt-get install -y amazon-efs-utils
Reading package lists... Done
Building dependency tree
Reading state information... Done
E: Unable to locate package amazon-efs-utils
ubuntu@ip-172-31-35-161:~$ sudo apt-get update
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu xenial InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu xenial-updates InRelease [109 kB]
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu xenial-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu xenial-security InRelease
Fetched 109 kB in 0s (393 kB/s)
Reading package lists... Done
ubuntu@ip-172-31-35-161:~$ sudo apt-get install -y amazon-efs-utils
Reading package lists... Done
Building dependency tree
Reading state information... Done
E: Unable to locate package amazon-efs-utils
ubuntu@ip-172-31-35-161:~$ sudo apt install -y amazon-efs-utils
Reading package lists... Done
Building dependency tree
Reading state information... Done
E: Unable to locate package amazon-efs-utils
ubuntu@ip-172-31-35-161:~$ mkdir efsmount
```

- Mount amazon file system to that directory using below command

Sudo mount -t directory-name efs-filesystem-id

- Copy the efs-filesystem-id from EFS

AWS-soft-copy (Last saved by user) - Microsoft Word

```
ubuntu@ip-172-31-35-161:~$ Building dependency tree
ubuntu@ip-172-31-35-161:~$ Reading state information... Done
ubuntu@ip-172-31-35-161:~$ E: Unable to locate package amazon-efs-utils
ubuntu@ip-172-31-35-161:~$ sudo apt install -y amazon-efs-utils
ubuntu@ip-172-31-35-161:~$ Reading package lists... Done
ubuntu@ip-172-31-35-161:~$ Building dependency tree
ubuntu@ip-172-31-35-161:~$ Reading state information... Done
ubuntu@ip-172-31-35-161:~$ E: Unable to locate package amazon-efs-utils
ubuntu@ip-172-31-35-161:~$ mkdir efsmount
ubuntu@ip-172-31-35-161:~$ sudo mount -t efsmount fs-17e779f4 /mnt/efs
mount: can't find fs-17e779f4 in /etc/fstab
ubuntu@ip-172-31-35-161:~$ sudo mount -t efsmount fs-17e779f4:/ /mnt/efs
mount: mount point /mnt/efs does not exist
ubuntu@ip-172-31-35-161:~$ sudo mount -t efsmount fs-17e779f4:/
mount: can't find fs-17e779f4:/ in /etc/fstab
ubuntu@ip-172-31-35-161:~$ sudo apt-get update
Hit:1 http://security.ubuntu.com/ubuntu xenial-security InRelease
Hit:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu xenial InRelease
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu xenial-updates InRelease [109 kB]
Hit:4 http://us-east-1.ec2.archive.ubuntu.com/ubuntu xenial-backports InRelease
Fetched 109 kB in 0s (347 kB/s)
Reading package lists... Done
ubuntu@ip-172-31-35-161:~$ sudo mount -t efsmount fs-17e779f4
$ cd efs
```

5. Make a subdirectory and change the ownership of that subdirectory to your EC2 instance user. Then navigate to that new directory with the following commands.

Page: 48 of 496 | Words: 92,753 | 11:52 PM | 6/7/2019

- Go to that mount direcory using below command
Cd efsmount

Elastic File System - Microsoft Word

```
ubuntu@ip-172-31-35-161:~$ Building dependency tree
ubuntu@ip-172-31-35-161:~$ Reading state information... Done
ubuntu@ip-172-31-35-161:~$ E: Unable to locate package amazon-efs-utils
ubuntu@ip-172-31-35-161:~$ sudo apt install -y amazon-efs-utils
ubuntu@ip-172-31-35-161:~$ Reading package lists... Done
ubuntu@ip-172-31-35-161:~$ Building dependency tree
ubuntu@ip-172-31-35-161:~$ Reading state information... Done
ubuntu@ip-172-31-35-161:~$ E: Unable to locate package amazon-efs-utils
ubuntu@ip-172-31-35-161:~$ mkdir efsmount
ubuntu@ip-172-31-35-161:~$ sudo mount -t efsmount fs-17e779f4 /mnt/efs
mount: can't find fs-17e779f4 in /etc/fstab
ubuntu@ip-172-31-35-161:~$ sudo mount -t efsmount fs-17e779f4:/ /mnt/efs
mount: mount point /mnt/efs does not exist
ubuntu@ip-172-31-35-161:~$ sudo mount -t efsmount fs-17e779f4:/
mount: can't find fs-17e779f4:/ in /etc/fstab
ubuntu@ip-172-31-35-161:~$ sudo apt-get update
Hit:1 http://security.ubuntu.com/ubuntu xenial-security InRelease
Hit:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu xenial InRelease
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu xenial-updates InRelease [109 kB]
Hit:4 http://us-east-1.ec2.archive.ubuntu.com/ubuntu xenial-backports InRelease
Fetched 109 kB in 0s (347 kB/s)
Reading package lists... Done
ubuntu@ip-172-31-35-161:~$ cd efsmount
```

• Go to that mount direcory using below command
Cd efsmount

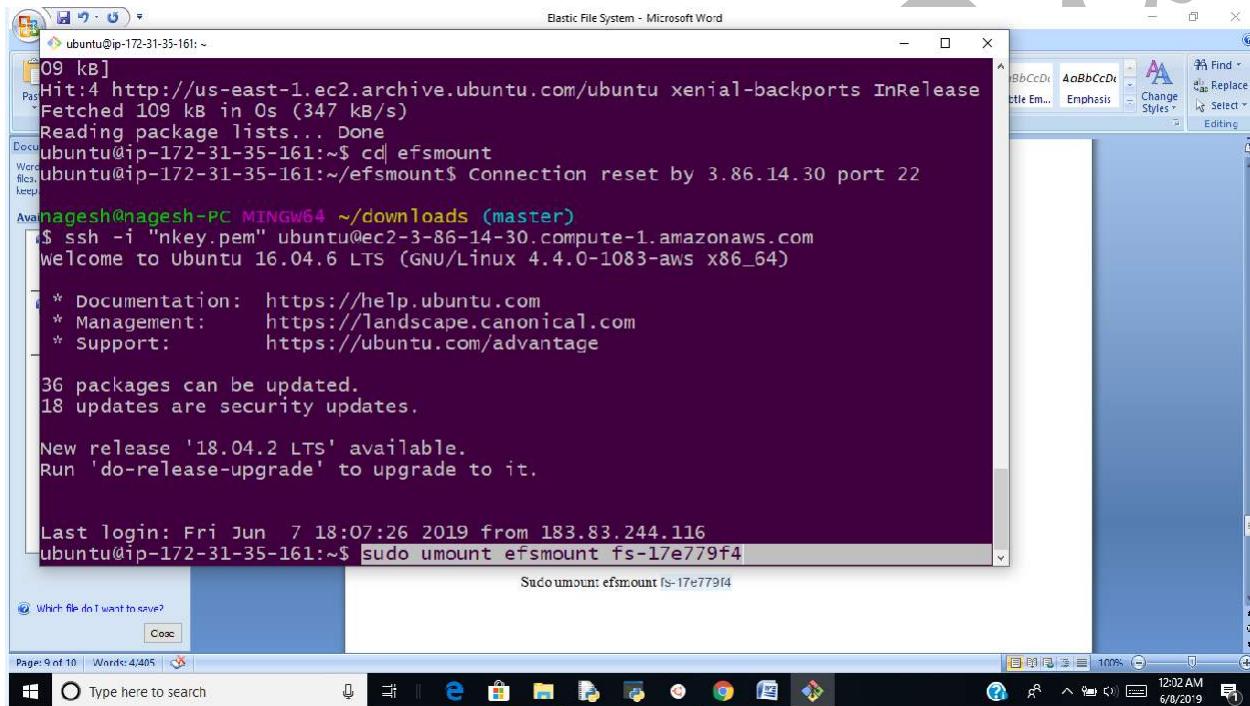
Page: 8 of 9 | Words: 355 | 11:55 PM | 6/7/2019

- Create files that are stored in EFS
- Unmount EFS and mount another EFS then we access first instance data that is called file transferring using EFS

Detach EFS from ec2 instance

- Connect to that ec2 instance and umount the EFS using below command

Sudo umount efsmount fs-17e779f4



Delete File system

- Select file system and goto actions and click delete file system option

The screenshot shows the AWS EFS console interface. In the center, there's a table titled 'File systems' with columns for Name, Metered size, Number of mount targets, and Creation date. The first row in the table has a context menu open, with the 'Delete file system' option highlighted in orange. The AWS navigation bar at the top includes 'Services' and 'Resource Groups'. The browser address bar shows the URL: https://us-east-1.console.aws.amazon.com/efs/home?region=us-east-1#/filesystems/fs-17e779f4.

- Enter file system id and click delete file system

The screenshot shows the 'Permanently delete file system' dialog box. It contains a warning message: 'This is a destructive action that cannot be undone.' Below it, another message states: 'This action will permanently delete the file system. The file system's mount targets will also be deleted.' A text input field asks for confirmation with the file system ID 'fs-17e779f4'. The 'Delete File System' button is visible at the bottom right of the dialog. The background shows the EFS file system list with the same file system entry as the previous screenshot.

Elastic Load Balancer (ELB)

Elastic Load Balancing distributes incoming application or network traffic across multiple targets, such as Amazon EC2 instances, containers, and IP addresses, in multiple Availability Zones. Elastic Load Balancing scales your load balancer as traffic to your application changes over time, and can scale to the vast majority of workloads automatically.

Elastic Load Balancing Works

A load balancer accepts incoming traffic from clients and routes requests to its registered targets (such as EC2 instances) in one or more Availability Zones. The load balancer also monitors the health of its registered targets and ensures that it routes traffic only to healthy targets. When the load balancer detects an unhealthy target, it stops routing traffic to that target, and then resumes routing traffic to that target when it detects that the target is healthy again.

You configure your load balancer to accept incoming traffic by specifying one or more *listeners*. A listener is a process that checks for connection requests. It is configured with a protocol and port number for connections from clients to the load balancer and a protocol and port number for connections from the load balancer to the targets.

Elastic Load Balancing supports three types of load balancers: Application Load Balancers, Network Load Balancers, and Classic Load Balancers. There is a key difference between the way

you configure these load balancers. With Application Load Balancers and Network Load Balancers, you register targets in target groups, and route traffic to the target groups. With Classic Load Balancers, you register instances with the load balancer.

Classic Load Balancer (CLB) Use Cases

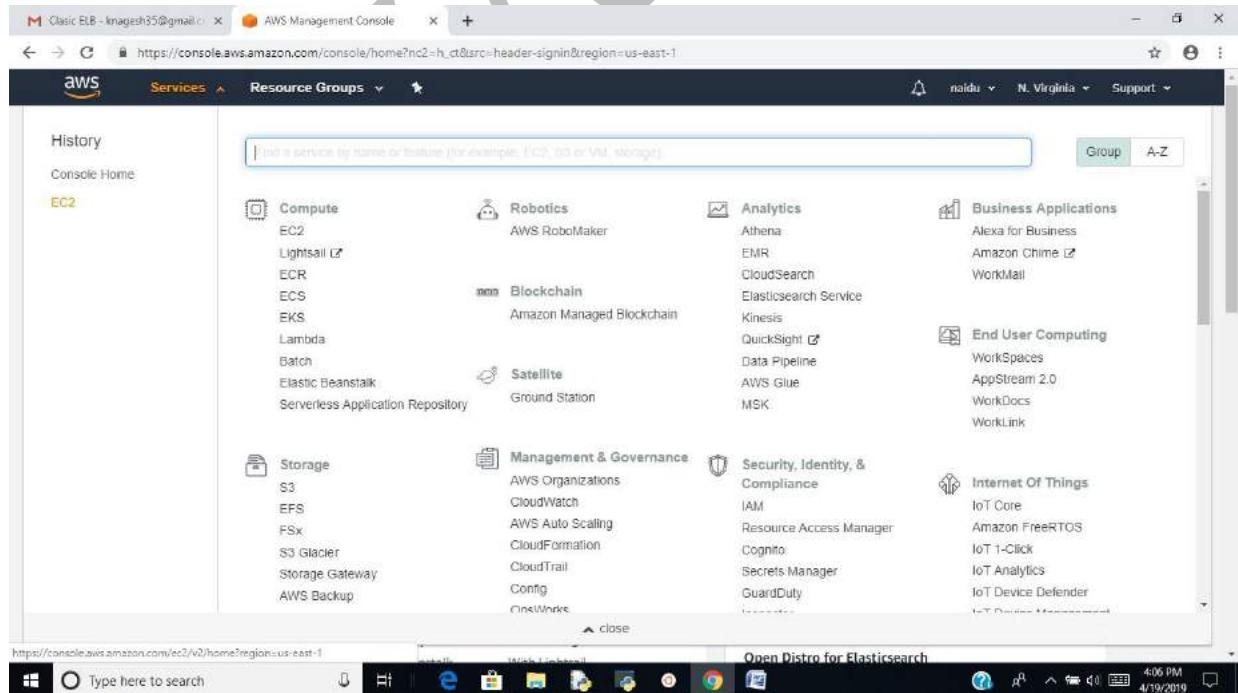
Uc1) Create ELB and Attach Instances to ELB

1. Create 3 ec2 instances with below user data

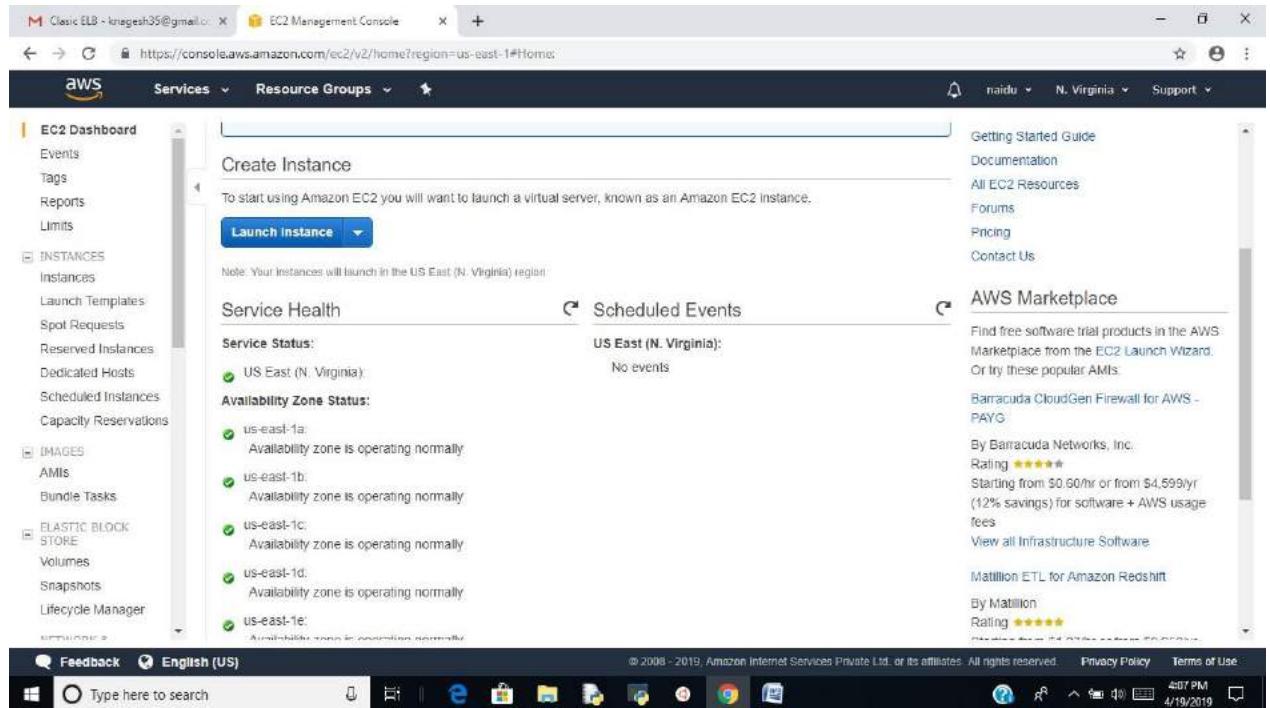
```
#!/bin/bash
apt-get update
apt-get install apache2 -y
mkdir /var/www/html/prepaid
echo This is prepaid page $(date) > /var/www/html/prepaid/test.html
```

login into aws console

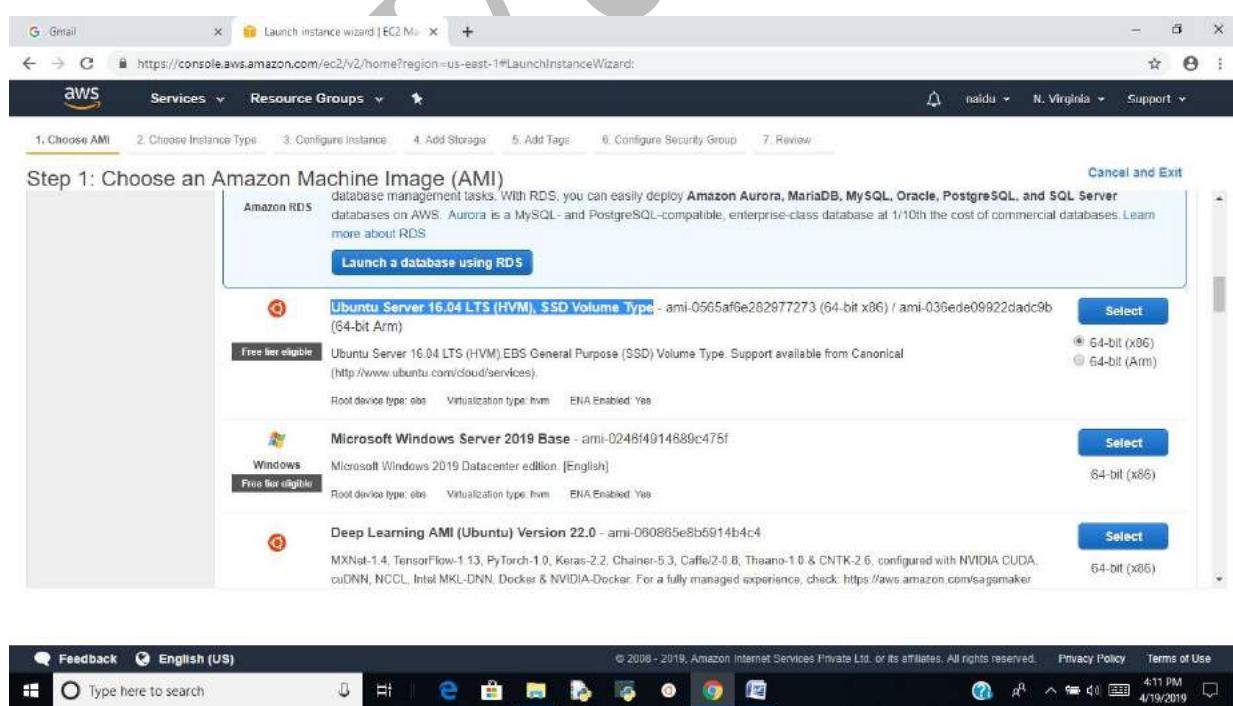
select services click on ec2



Click on launch instances



Choose ami : click on Ubuntu Server 16.04 LTS (HVM), SSD Volume Type



Choose Instance Type: select *t2.micro* and click on Next:*Configure instance Details*

The screenshot shows the AWS Launch Instance Wizard interface. The top navigation bar includes 'Gmail', 'Launch instance wizard | EC2 Man...', and a link to 'https://console.aws.amazon.com/ec2/v2/home?region=us-east-1#LaunchInstanceWizard;'. The AWS logo is at the top left. Below it, the 'Services' dropdown is set to 'Resource Groups'. The main content area is titled 'Step 2: Choose an Instance Type'. It displays a table of instance types with the following columns: Family, Type, vCPUs, Memory (GiB), Instance Storage (GiB), EBS-Optimized Available, Network Performance, and IPv6 Support. The 'Currently selected' row is highlighted in green and shows 't2.micro (Variable ECUs, 1 vCPU, 2.5 GHz, Intel Xeon Family, 1 GiB memory, EBS only)'. The 't2.micro' row has a green background and a green border around the 'Type' column. The 'Free tier eligible' badge is also visible in this row. Other rows include 't2.nano', 't2.small', 't2.medium', and 't2.large'. At the bottom of the table are buttons for 'Cancel', 'Previous', 'Review and Launch' (which is highlighted in blue), and 'Next: Configure Instance Details'.

In configure instance details give the value 3 in *nuber of instances* section and click on *Advanced Details* section and paste below userdata in that box and click *Review and Launch*

Userdata:

```
#!/bin/bash
apt-get update
apt-get install apache2 -y
mkdir /var/www/html/prepaid
echo This is prepaid page $(date) > /var/www/html/prepaid/test.html
```

Step 3: Configure Instance Details

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot Instances to take advantage of the lower pricing, assign an access management role to the instance, and more.

Purchasing option: Request Spot instances

Network: vpc-1aa11160 (default) | Create new VPC

Subnet: No preference (default subnet in any Availability Zone) | Create new subnet

Auto-assign Public IP: Use subnet setting (Elastic)

Placement group: Add instance to placement group

Capacity Reservation: Open | Create new Capacity Reservation

Monitoring: Enable CloudWatch detailed monitoring
Additional charges apply

Tenancy: Shared - Run a shared hardware instance
Additional charges will apply for dedicated tenancy

Elastic Inference: Add an Elastic Inference accelerator
Additional charges apply

T2/T3 Unlimited: Enable
Additional charges may apply

Advanced Details

User data:

As text: As file: Input is already base64 encoded

```
#!/bin/bash
apt-get update
apt-get install apache2 -y
mkdir /var/www/html/prepaid
echo This is prepaid page $date > /var/www/html/prepaid/test.html
```

Review and Launch

Click on launch

Step 7: Review Instance Launch

Instance Type	ECUs	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance
t2.micro	Variable	1	1	EBS only	-	Low to Moderate

Security Groups

Security group name: launch-wizard-4
Description: launch-wizard-4 created 2019-04-19T16:38:10.994+05:30

Type	Protocol	Port Range	Source	Description
This security group has no rules				

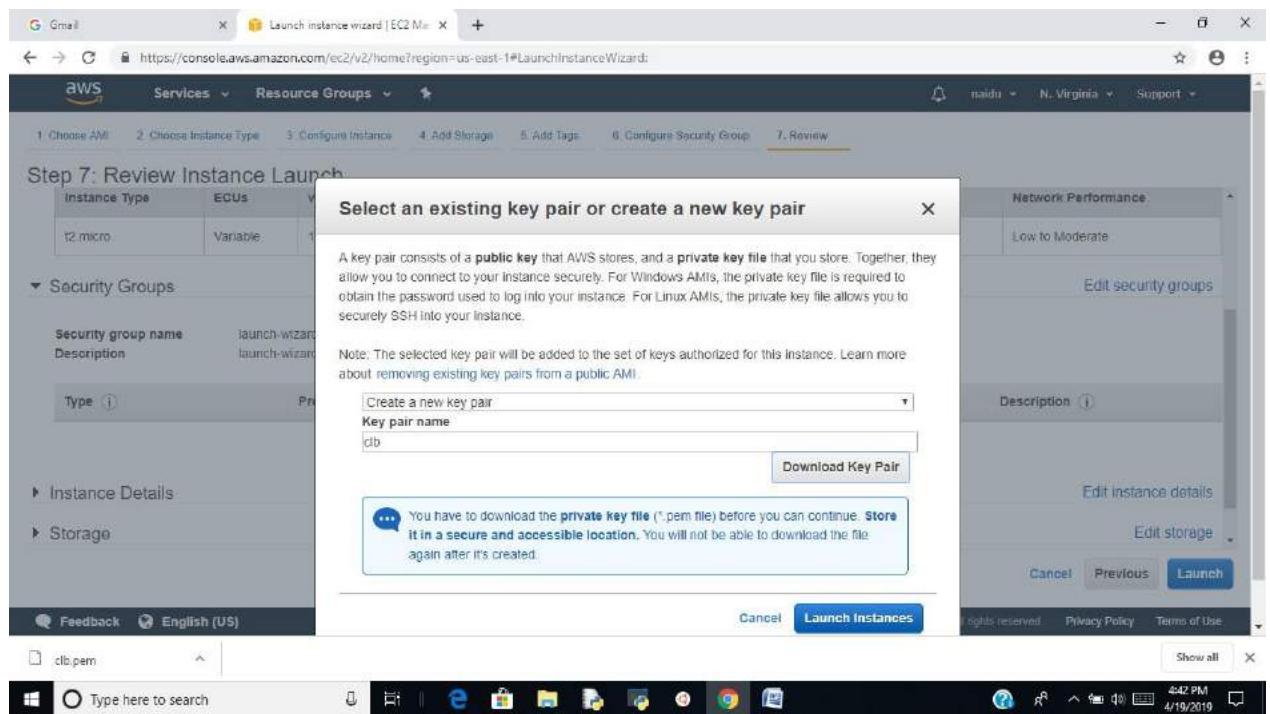
Instance Details | Storage | Tags

Edit instance details | Edit storage | Edit tags

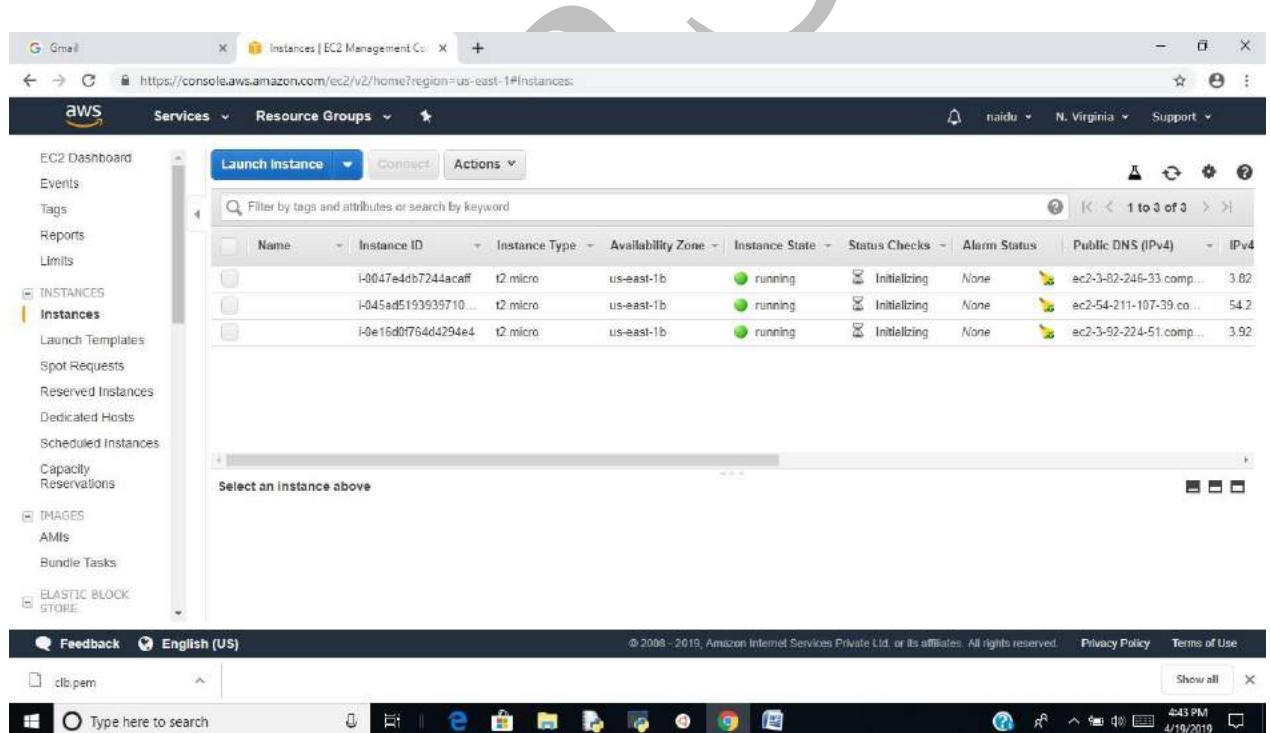
Cancel | Previous | Launch

In keypair section select *create new key pair* option and write some name to that key pair and click on *download*

Then click on Launch instance



Next click on View instance to see the instances in dash board



- Add port 80 in above created ec2 instances security group inbound rules

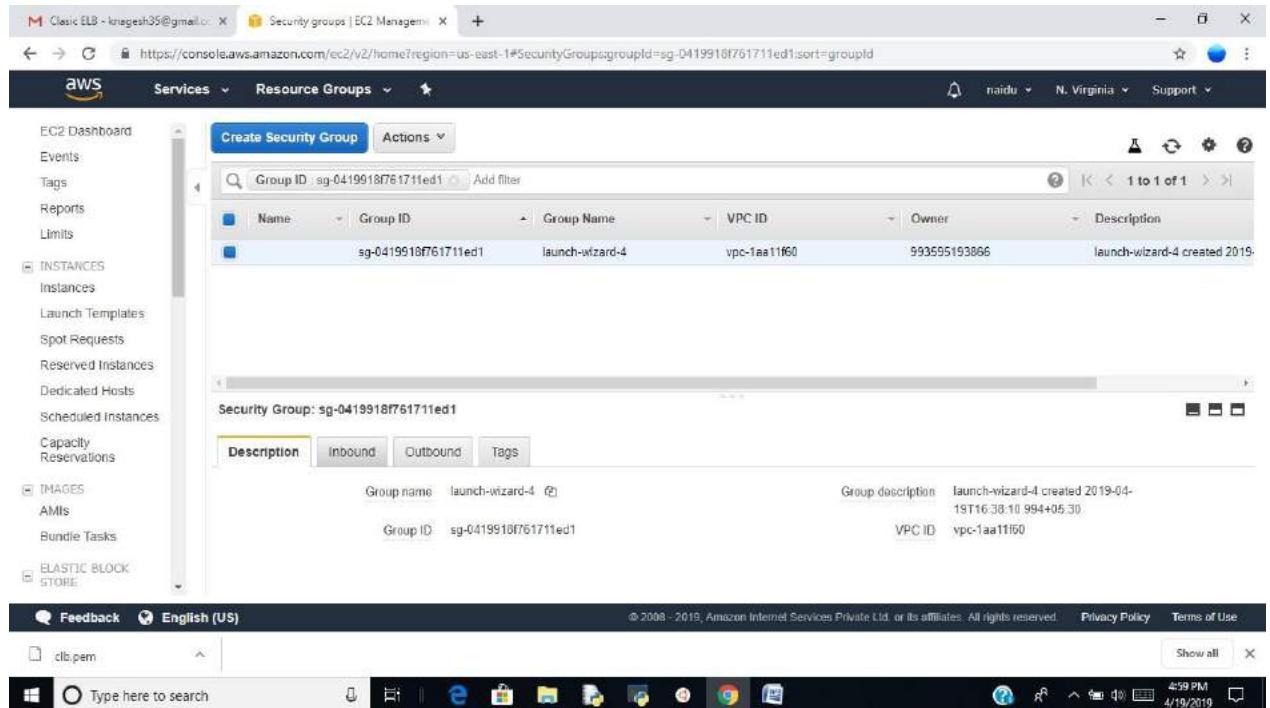
Select any one of the instance in above three instances

The screenshot shows the AWS EC2 Management Console interface. On the left, there's a navigation sidebar with options like EC2 Dashboard, Events, Tags, Reports, Limits, Instances (selected), Launch Templates, Spot Requests, Reserved Instances, Dedicated Hosts, Scheduled Instances, Capacity Reservations, Images (AMIs), Bundle Tasks, and Elastic Block Store. The main area displays a table of instances. The first instance, with the ID i-0047e4db7244acaff, is highlighted and selected. A modal window for this instance is open, showing its description: "Instance: i-0047e4db7244acaff Public DNS: ec2-3-82-246-33.compute-1.amazonaws.com". Below the modal, the instance details are listed: Instance ID (i-0047e4db7244acaff), Instance state (running), Instance type (t2.micro), Public DNS (IPv4) (ec2-3-82-246-33.compute-1.amazonaws.com), IPv4 Public IP (3.82.246.33), and IPv6 IPs (-). At the bottom of the page, there are links for Feedback, English (US), Privacy Policy, and Terms of Use.

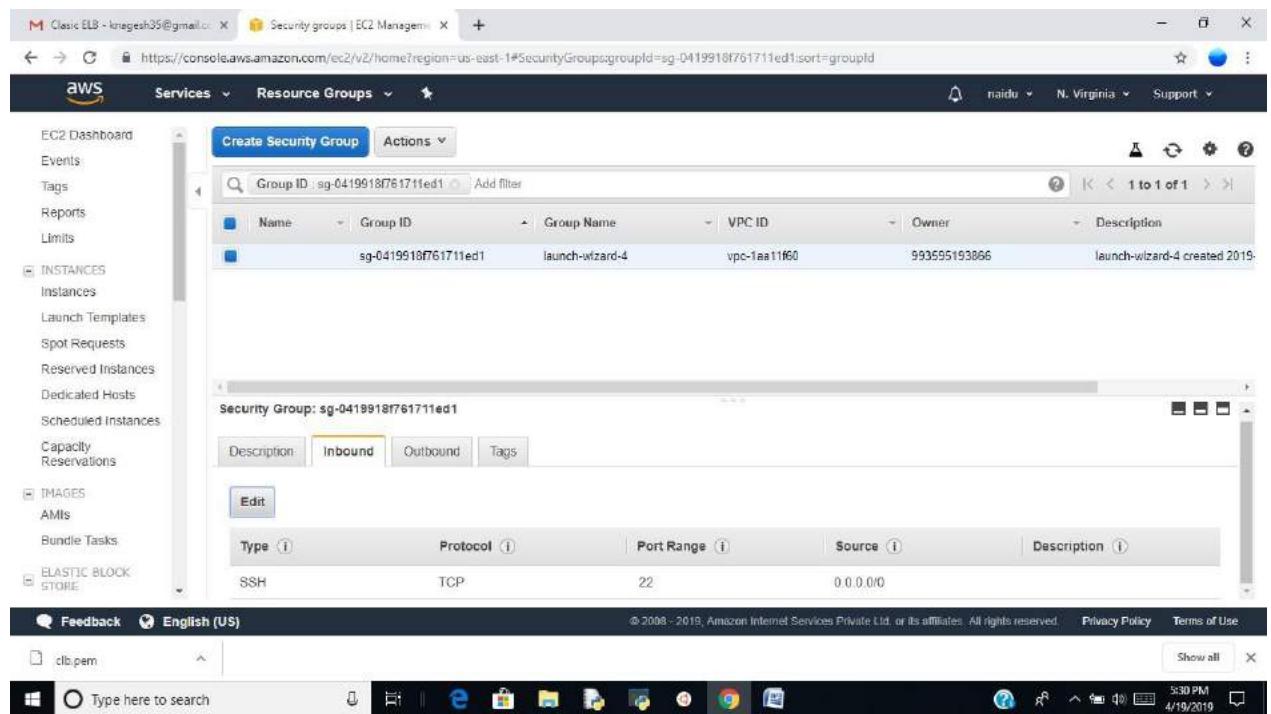
Then you see the below *Description* section here you find security groups option and click on that *security group name (launch-wizard)*

This screenshot is similar to the previous one but focuses on the security group information for the selected instance. The 'Security groups' field in the modal is now highlighted, showing 'launch-wizard-4'. Other visible fields include Instance type (t2.micro), Availability zone (us-east-1b), and AMI ID (ubuntu/images/hvm-ssd/ubuntu-xenial-16.04-amd64-server-20190212 (ami-0565af5e202977273)). The rest of the interface and the Windows taskbar at the bottom remain the same.

Then click on **Inbound** option



Then click on **Edit** option



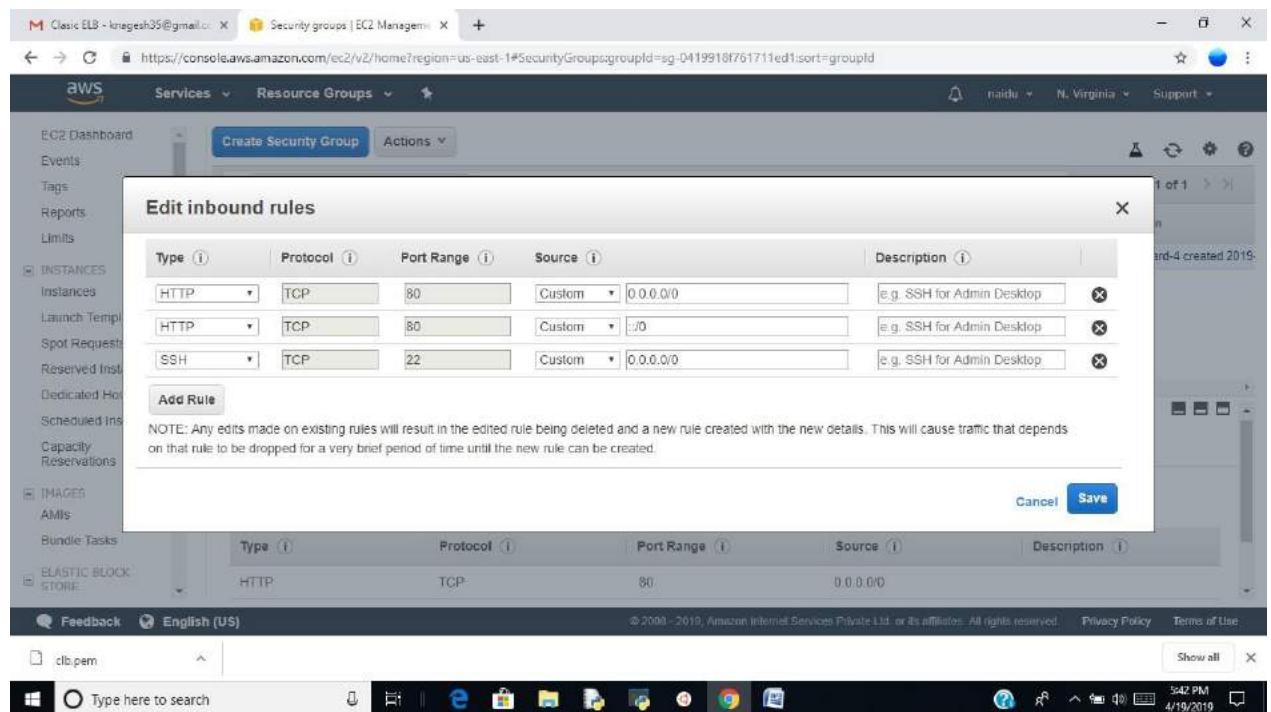
Click on Add Rule and enter Inbound Rule protocol attributes

Type: select *HTTP*

Protocol and port range assigned automatically

Source: select *anywhere* option

And click on save



Note: no need change for ramining two instances security groups protocol since three instances have same security group

3. Create Clasic ELB and Attach above three Instances

Click on load balancers in the left side panel of ec2 service

The screenshot shows the AWS EC2 Instances Management Console. The left sidebar navigation includes: 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), and SYSTEMS MANAGER SERVICES (Run Command). The main content area displays a table of instances:

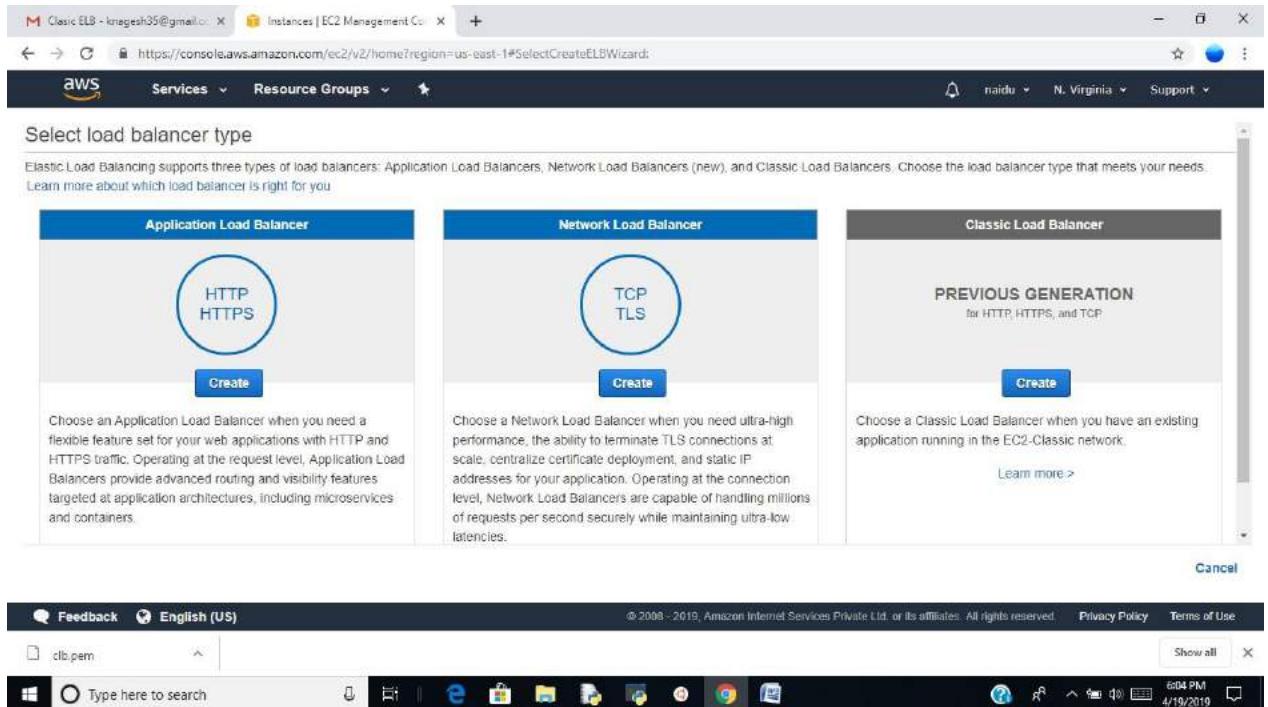
Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)	IPv4
	i-0047e4db7244acaff	t2.micro	us-east-1b	running	2/2 checks ...	None	ec2-3-82-246-33.comp...	3.82
	i-045ad5193939710...	t2.micro	us-east-1b	running	2/2 checks ...	None	ec2-54-211-107-39.co...	54.2
	i-0e16d0f0764d4294e4	t2.micro	us-east-1b	running	2/2 checks ...	None	ec2-3-92-224-51.comp...	3.92

A message at the bottom says "Select an instance above". The browser address bar shows "https://console.aws.amazon.com/ec2/v2/home?region=us-east-1#Instances:".

Click on Create Load Balancer

The screenshot shows the AWS Load Balancers Management Console. The left sidebar navigation includes: 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), and SYSTEMS MANAGER SERVICES (Run Command). The main content area displays a table with the message "You do not have any load balancers in this region." A message at the bottom says "Select a load balancer". The browser address bar shows "https://console.aws.amazon.com/ec2/v2/home?region=us-east-1#LoadBalancers:".

Select the Load Balancer Type: Classic Load Balancer (click on *Create*)



Define Load Balancer:

Load Balancer Name: enter name for Load Balancer

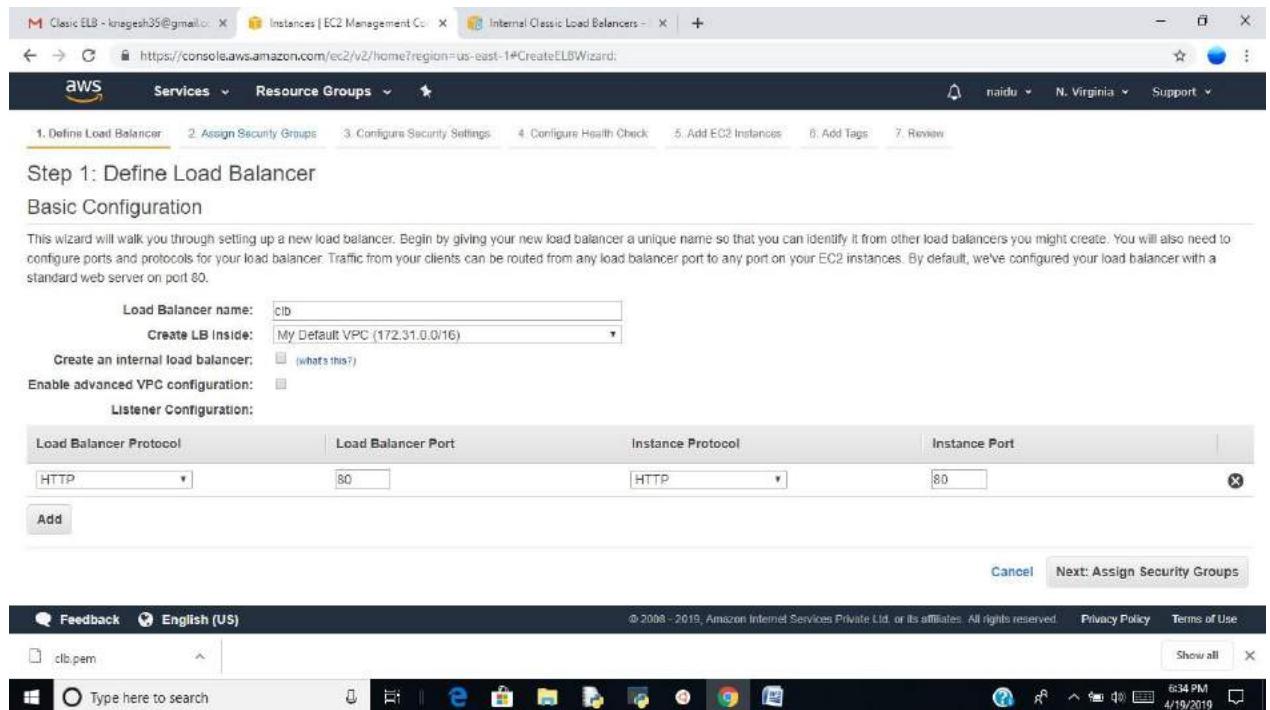
Load Balancer protocol: HTTP

Load Balancer port: 80

Instance protocol: HTTP

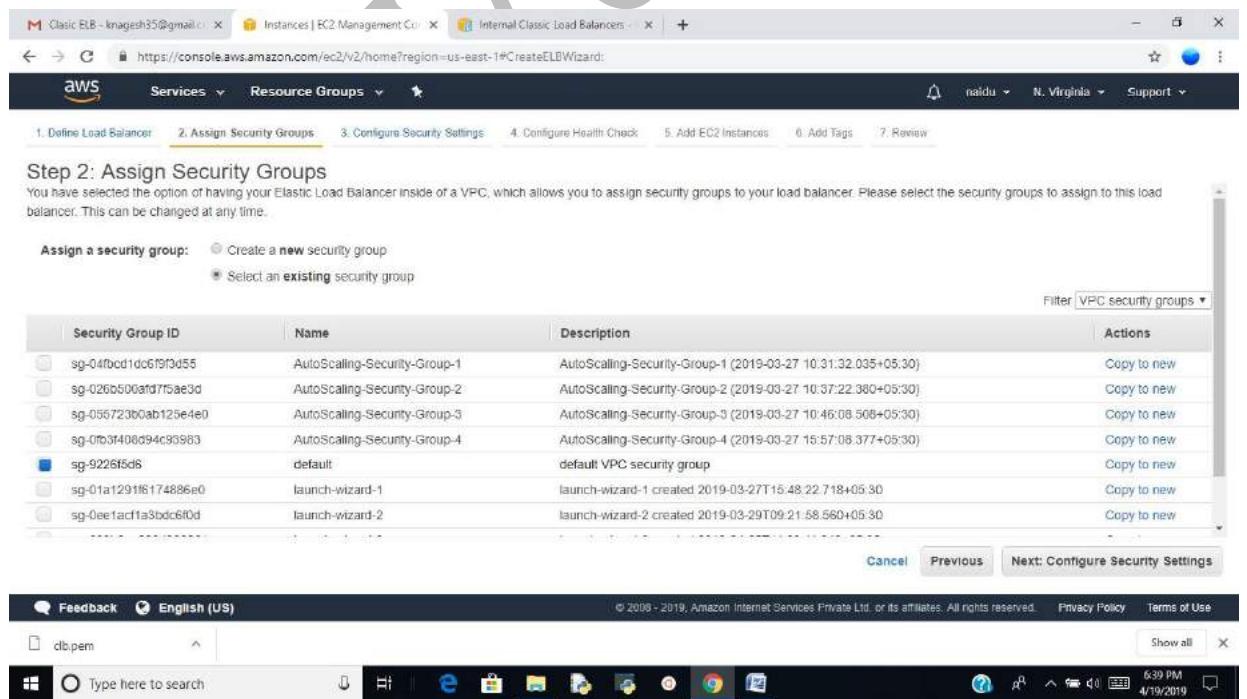
Instance port: 80

Note: By default these ports assigned

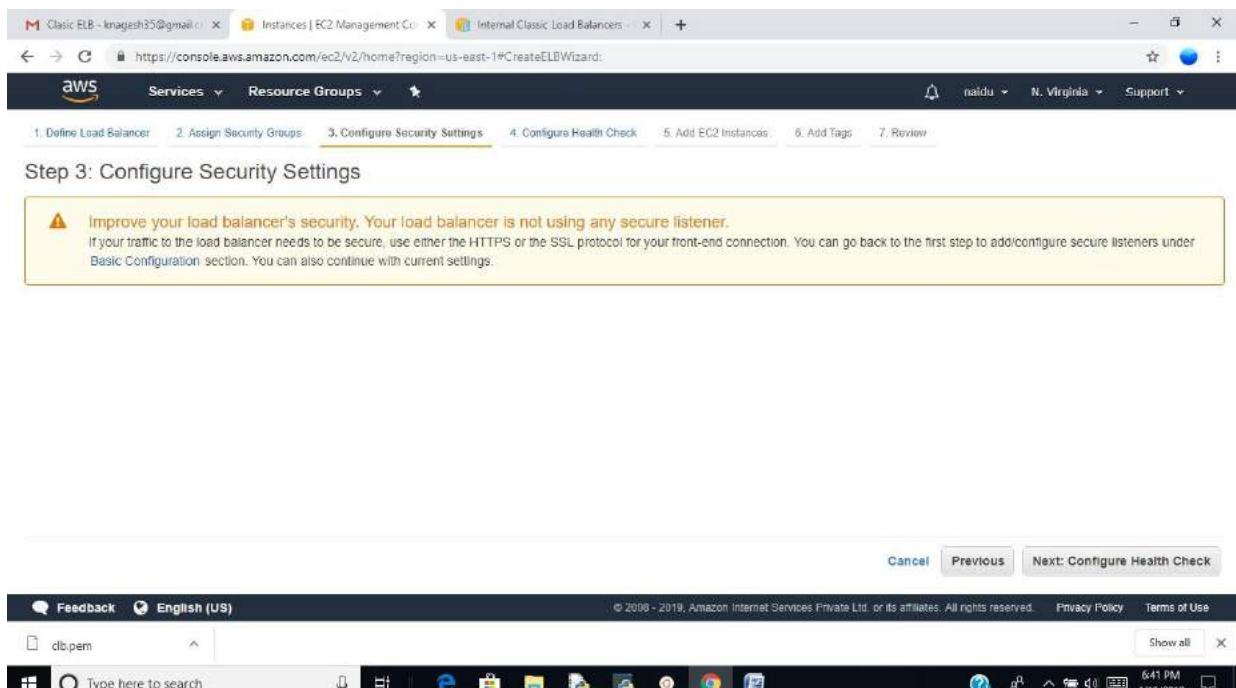


Click on next assign security groups

Selecting security group
Bydefault it is assigned with Default Security group



Click on next configure security settings



Click on Configure Health check

Ping Protocol: HTTP

Ping port: 80

Ping path: /index.html

Advanced Details:

Response Timeout: 5 seconds

Interval: 10 seconds (By default 30)

Unhealthy Threshold: 2

Healthy Threshold: 5 (By default 10)

Note: By default these values are assigned do you want modify then change

Click on Next: Add Ec2 Instances

Step 4: Configure Health Check

Your load balancer will automatically perform health checks on your EC2 instances and only route traffic to instances that pass the health check. If an instance fails the health check, it is automatically removed from the load balancer. Customize the health check to meet your specific needs.

Ping Protocol	HTTP
Ping Port	80
Ping Path	/index.html

Advanced Details

Response Timeout	5	seconds
Interval	30	seconds
Unhealthy threshold	2	
Healthy threshold	10	

Cancel Previous Next: Add EC2 Instances

It is showing list of EC2 instances then you select above created three instances

Step 5: Add EC2 Instances

The table below lists all your running EC2 instances. Check the boxes in the Select column to add those instances to this load balancer.

VPC vpc-1aa11f60 (172.31.0.0/16)

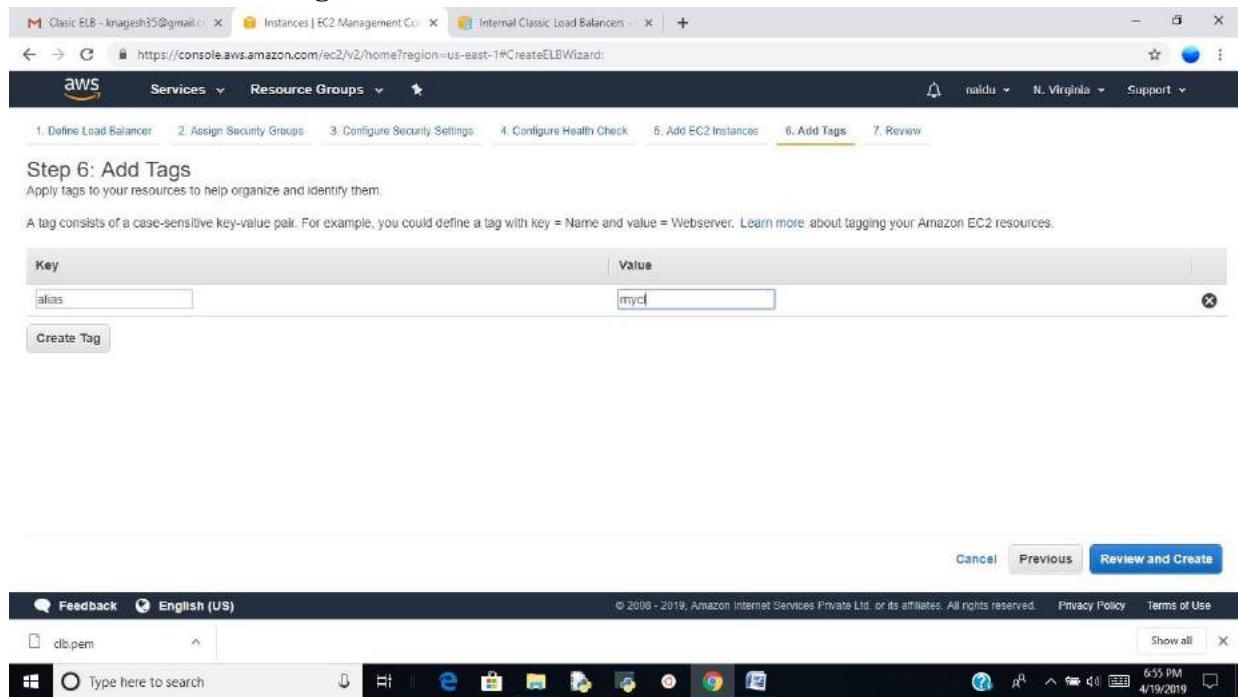
Select	Instance	Name	State	Security groups	Zone	Subnet ID	Subnet CIDR
<input checked="" type="checkbox"/>	i-0047e4db7244acaf		running	launch-wizard-4	us-east-1b	subnet-9d96fb3	172.31.80.0/20
<input checked="" type="checkbox"/>	i-0a16d0f764d4294e4		running	launch-wizard-4	us-east-1b	subnet-9d96fb3	172.31.80.0/20
<input checked="" type="checkbox"/>	i-045ad519393971075		running	launch-wizard-4	us-east-1b	subnet-9d96fb3	172.31.80.0/20

Availability Zone Distribution
3 instances in us-east-1b

Enable Cross-Zone Load Balancing

Cancel Previous Next: Add Tags

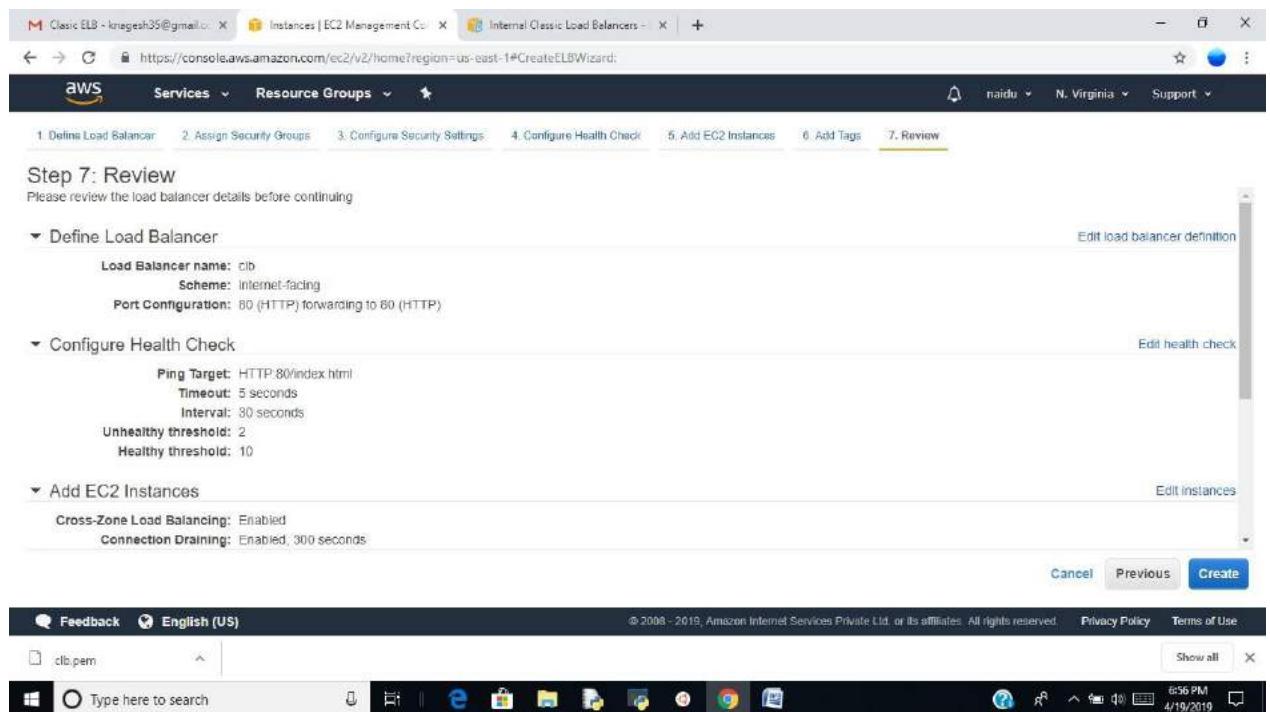
Click on Next: Add Tags



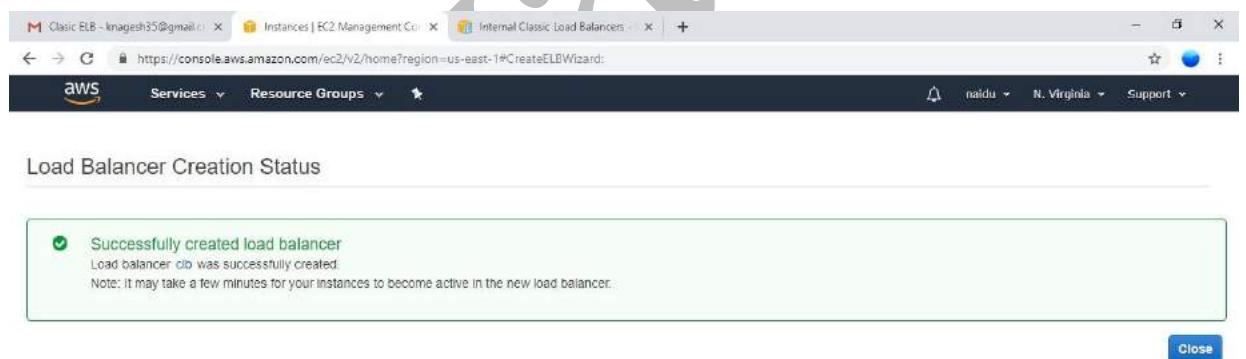
Note: tags is the optional

Click on review and create

Then you see the all configuration options to create ELB

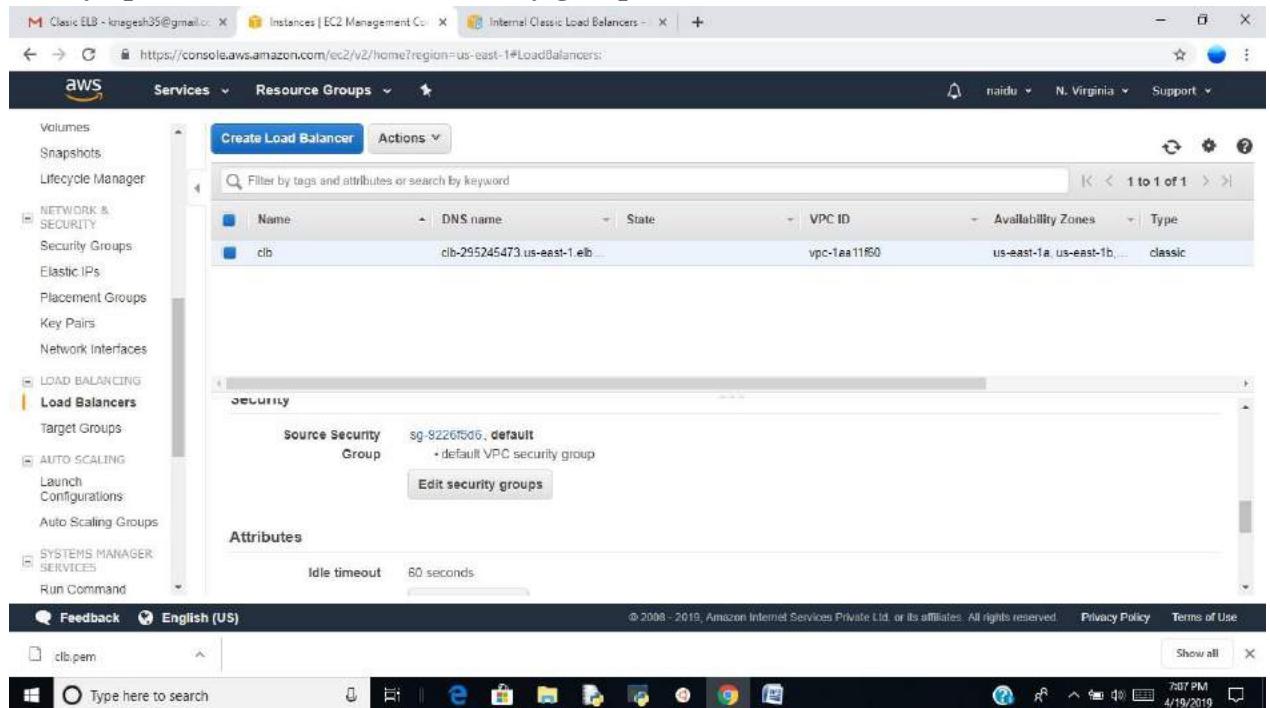


Then click on create
click on close



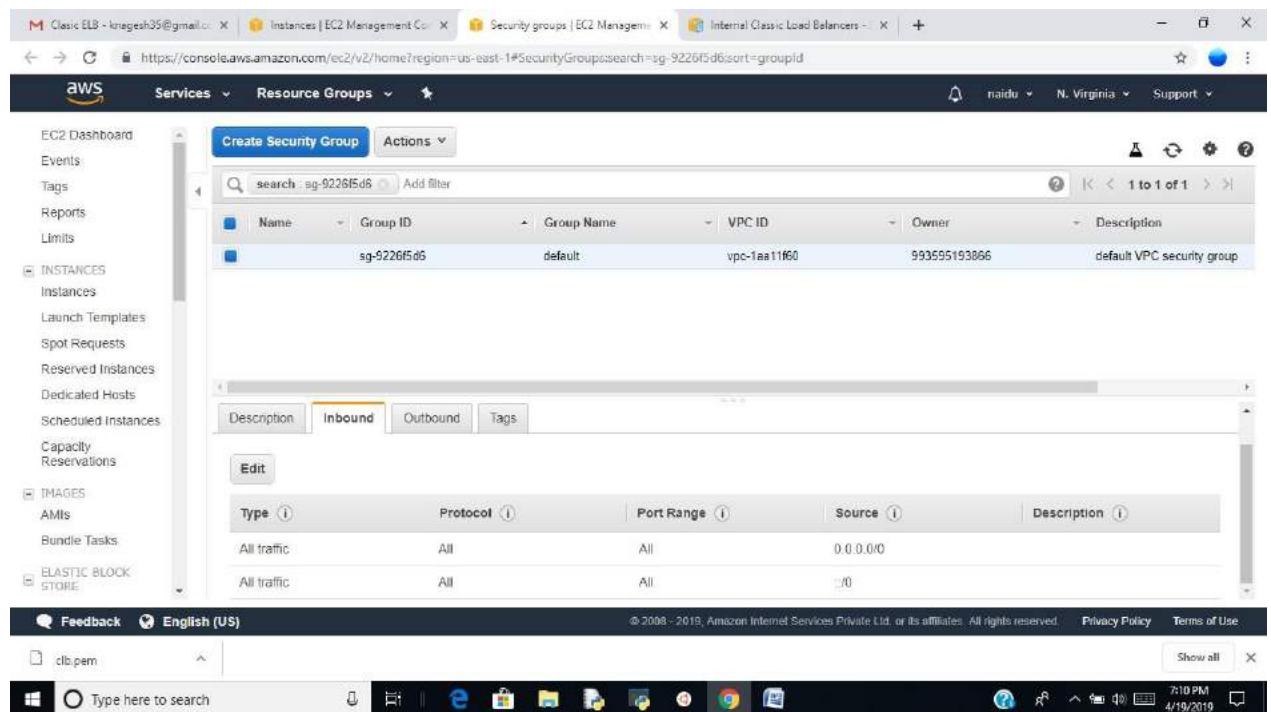
4. Make sure the ELB security group Inbound Rule is allowing port 80

Select load Balancer and click description section below and goto security option and click on that security group name



The screenshot shows the AWS Management Console with the URL <https://console.aws.amazon.com/ec2/v2/home?region=us-east-1#LoadBalancers>. The left sidebar navigation includes Volumes, Snapshots, Lifecycle Manager, NETWORK & SECURITY (Security Groups selected), Elastic IPs, Placement Groups, Key Pairs, Network Interfaces, LOAD BALANCING (Load Balancers selected), Target Groups, AUTO-SCALING, Launch Configurations, Auto Scaling Groups, and SYSTEMS MANAGER SERVICES. The main content area displays a table of Load Balancers. One row is selected, showing details: Name (clb), DNS name (clb-295245473.us-east-1.elb.amazonaws.com), State (Active), VPC ID (vpc-1ea11f60), Availability Zones (us-east-1a, us-east-1b, us-east-1c), and Type (classic). Below the table, the 'Security' section shows the Source Security Group as sg-8226f5d6, default (a default VPC security group). There is a link to 'Edit security groups'. The 'Attributes' section shows an Idle timeout of 60 seconds. The bottom of the screen shows the Windows taskbar with various pinned icons.

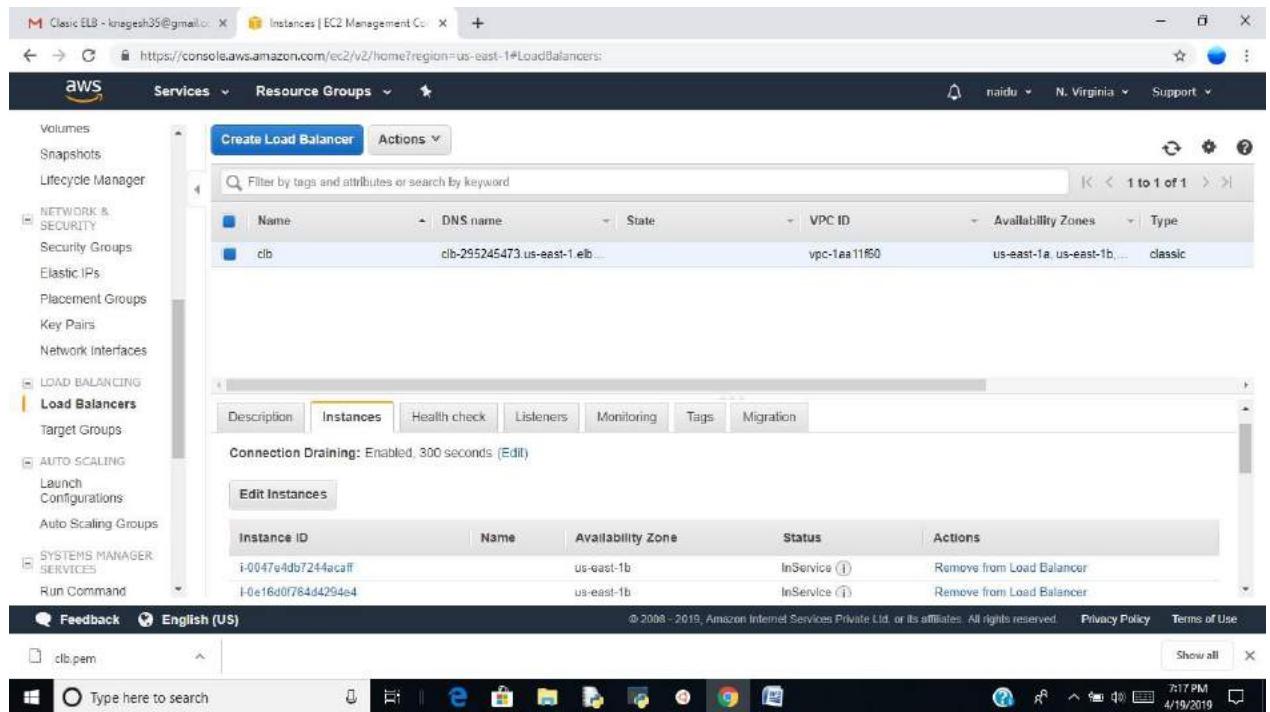
In that security group click on *inbound* option then you see list of protocols



Then check port 80 exist or not. Here *all traffic* protocol is existed so port 80 also belongs to this rule no need to add. If it is not existed add by click on Edit option

5. Check ELB instances Health status is inservice or not

Click on Load Balancer and select Instances Section below then you see the list of instances and it's status

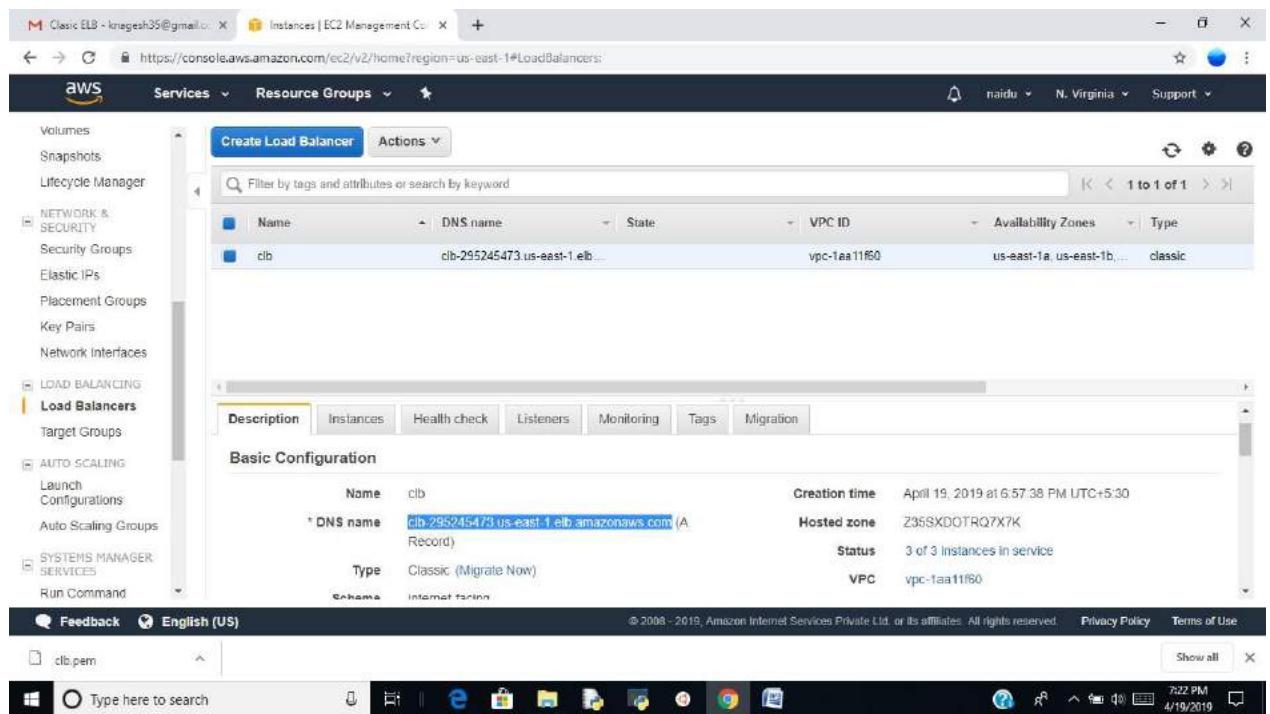


Here all Instances have Inservice Status.

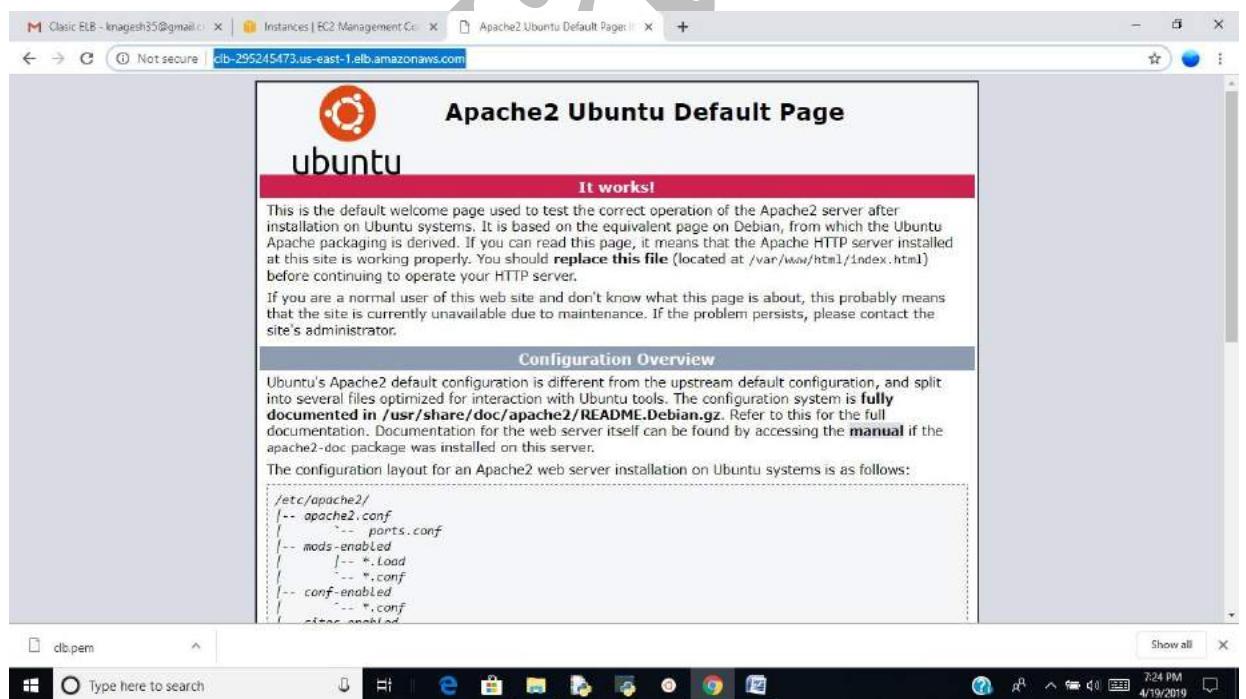
If it have out of service status then check where the problem is occurred

- Once instances is in "inservice" copy the ELB DNS NAmE

Click on ELB then go to description section here we find DNS name attribute and value copy the DNS name value



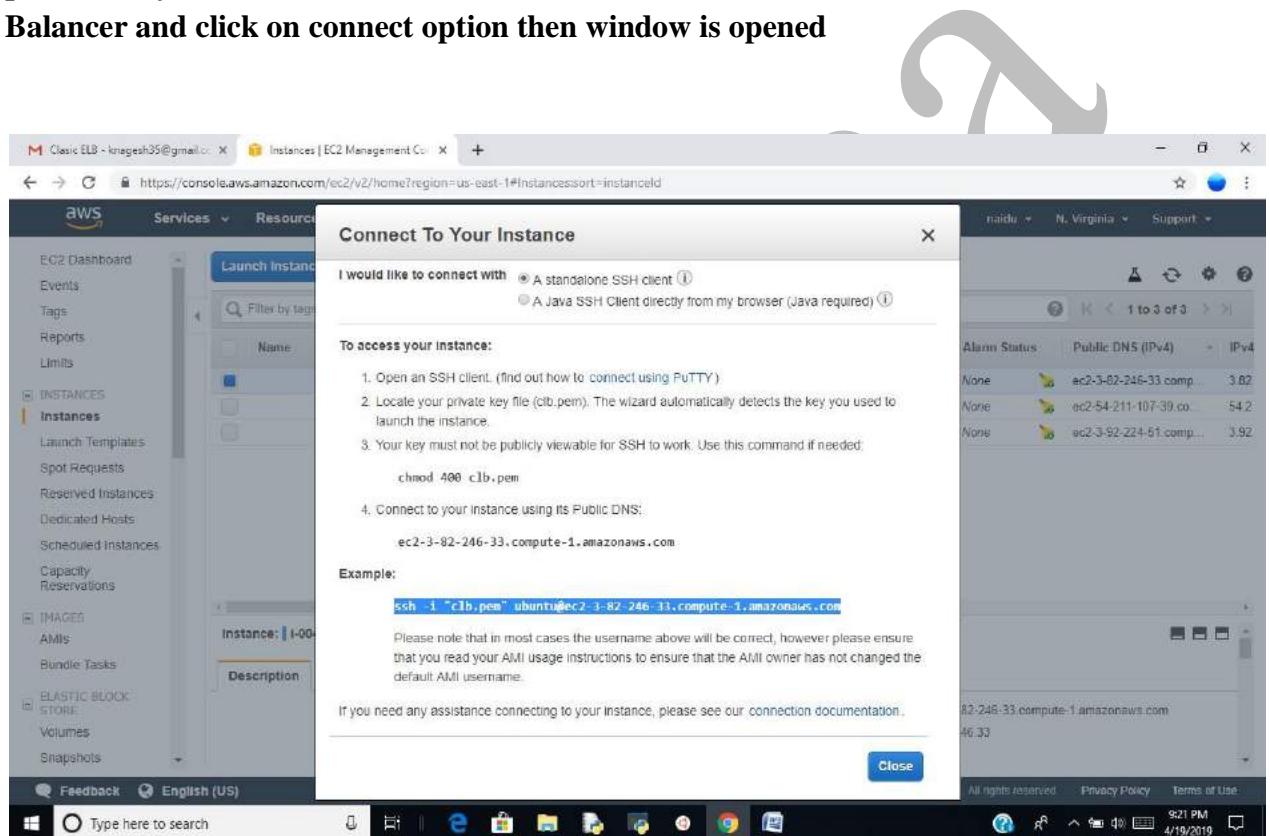
7. Open the browser paste ELB DNS Name



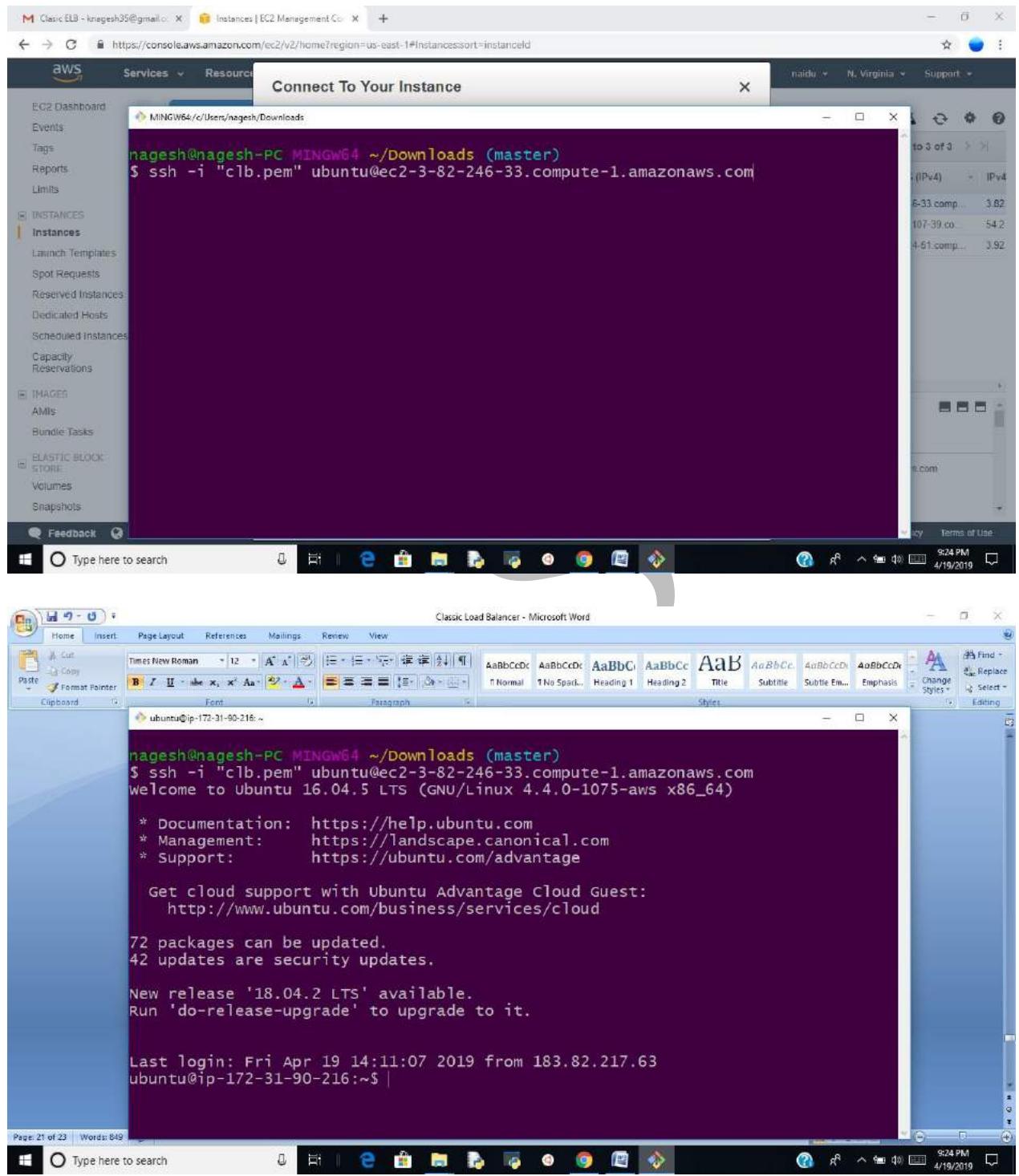
8. Refresh the page multiple time, You will Observe that load is distributing Between Multiple Instances

To identify easily this Load Balancer distribubute the among server you edit the apache index.html page in server1 as server1 and in server2 edit as server2 and server3 edit as server3 by fallowing procedure

Go to the services in aws console and select EC2 and click on instace on left side panel then you see list of instances click one instance which is attached to the Load Balancer and click on connect option then window is opened



Copy the ssh command that is highlighted in above screen shot. Open the gitbash in downloads location and paste above code and enter then you connected to EC2 Instance



Create one file and write data to that file using cat command (cat > s1)

Write server1-----first machine

Write server2-----second machine

Write server3-----third machine

Move this file content to apache index.html page

Sudo mv s1 /var/www/html/index.html

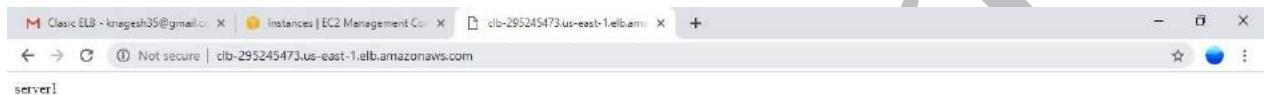
Enter

Restart apache server

Sudo service apache2 restart

Then copy paste the DNS name in Browser and enter, then observe

First time coming server1 page



Refresh then coming server2 page





Again Refresh then coming Server3 page



So finally work isw distibuted equally among all Instances that is called Load Balancer

Uc2) ELB Always send request to Healthy instances only

1. Stop the apache2 service in one of the EC2 instances which we have launched in usecase1

Connect the first instance through gitbash

Stop the apache2 service in that machine using below command screen shot

The screenshot shows a Windows desktop environment with a browser window open to the AWS EC2 Management Console. The left sidebar shows 'Instances' selected. A terminal window is open with the following command history:

```
ubuntu@ip-172-31-90-216:~$ Connection reset by 3.82.246.33 port 22
nagesh@nagesh-PC MINGW64 ~/Downloads (master)
$ ssh -i "clb.pem" ubuntu@ec2-3-82-246-33.compute-1.amazonaws.com
Welcome to Ubuntu 16.04.5 LTS (GNU/Linux 4.4.0-1075-aws x86_64)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/advantage

 Get cloud support with Ubuntu Advantage Cloud Guest:
 http://www.ubuntu.com/business/services/cloud

72 packages can be updated.
42 updates are security updates.

New release '18.04.2 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Fri Apr 19 15:54:30 2019 from 183.83.246.150
ubuntu@ip-172-31-90-216:~$ sudo service apache2 stop
ubuntu@ip-172-31-90-216:~$
```

2. Check ELB instances health status

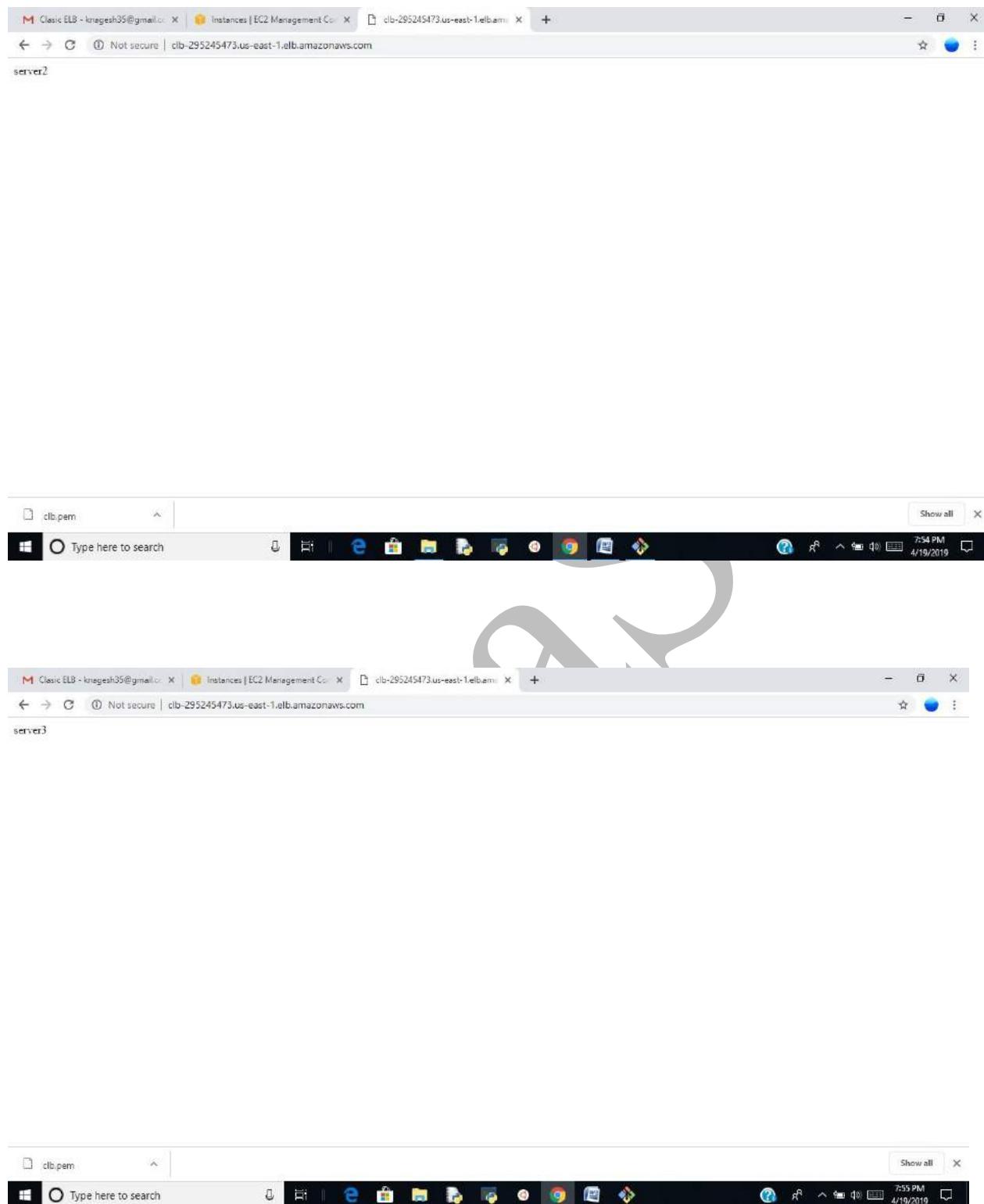
Click on Load Balances in left side panel of EC2 console and select Load Balancer and click on instances on below panel

The screenshot shows the AWS EC2 Management Console with the 'Instances' tab selected. A search bar at the top has 'clb' entered. Below it is a table with columns: Name, DNS name, State, VPC ID, Availability Zones, and Type. One row is highlighted for 'clb'. Below the table are tabs for Description, Instances, Health check, Listeners, Monitoring, Tags, and Migration. Under the Instances tab, there's a section titled 'Connection Draining: Enabled, 300 seconds (Edit)'. Below this is a table titled 'Edit Instances' with columns: Instance ID, Name, Availability Zone, Status, and Actions. Three instances are listed: 'i-0047e4db7244acaff' (Status: OutOfService), 'i-0a16d0f764d4294e4' (Status: InService), and 'i-045ad519393971075' (Status: InService). The status 'OutOfService' is highlighted.

You can observe above here one instance have outofservice status

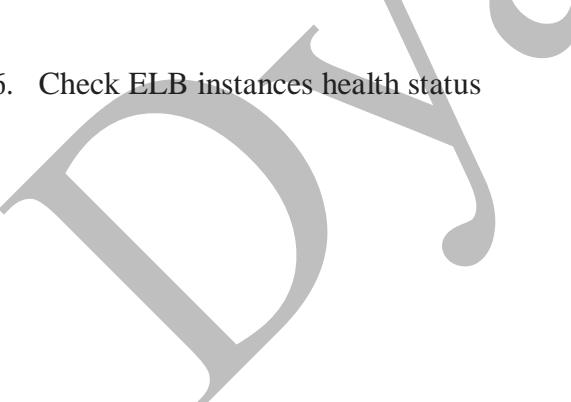
3. Open the browser paste ELB DNS Name
4. Refresh the page multiple time, You will Observe that load is not distributing to Unhealth instance

Amazon Web Services



You can observe from above screen shots the load is distributed to only server2 and server3 not server1 since it is unhealthy status

5. Start the apache2 service in EC2 instance



A screenshot of a Microsoft Word document window titled "Classic Load Balancer - Microsoft Word". The document contains a terminal session log from an Ubuntu 16.04.5 LTS system. The log shows the user logging in via SSH, updating packages, and then running the command "sudo service apache2 start". The terminal window has a blue background and white text.

```
nagesh@nagesh-PC MINGW64 ~/Downloads (master)
$ ssh -i "c1b.pem" ubuntu@ec2-3-82-246-33.compute-1.amazonaws.com
Welcome to Ubuntu 16.04.5 LTS (GNU/Linux 4.4.0-1075-aws x86_64)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/advantage

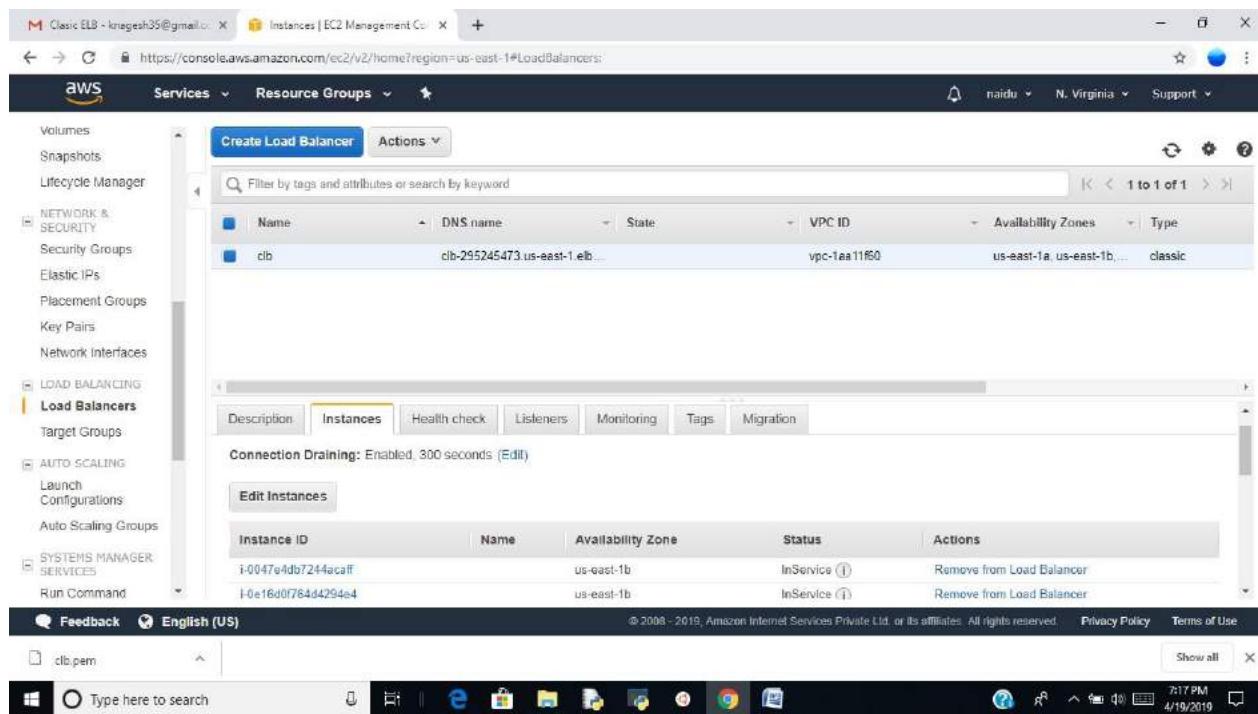
Get cloud support with Ubuntu Advantage Cloud Guest:
http://www.ubuntu.com/business/services/cloud

72 packages can be updated.
42 updates are security updates.

New release '18.04.2 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Fri Apr 19 16:15:24 2019 from 183.83.246.150
ubuntu@ip-172-31-90-216:~$ sudo service apache2 start
```

6. Check ELB instances health status

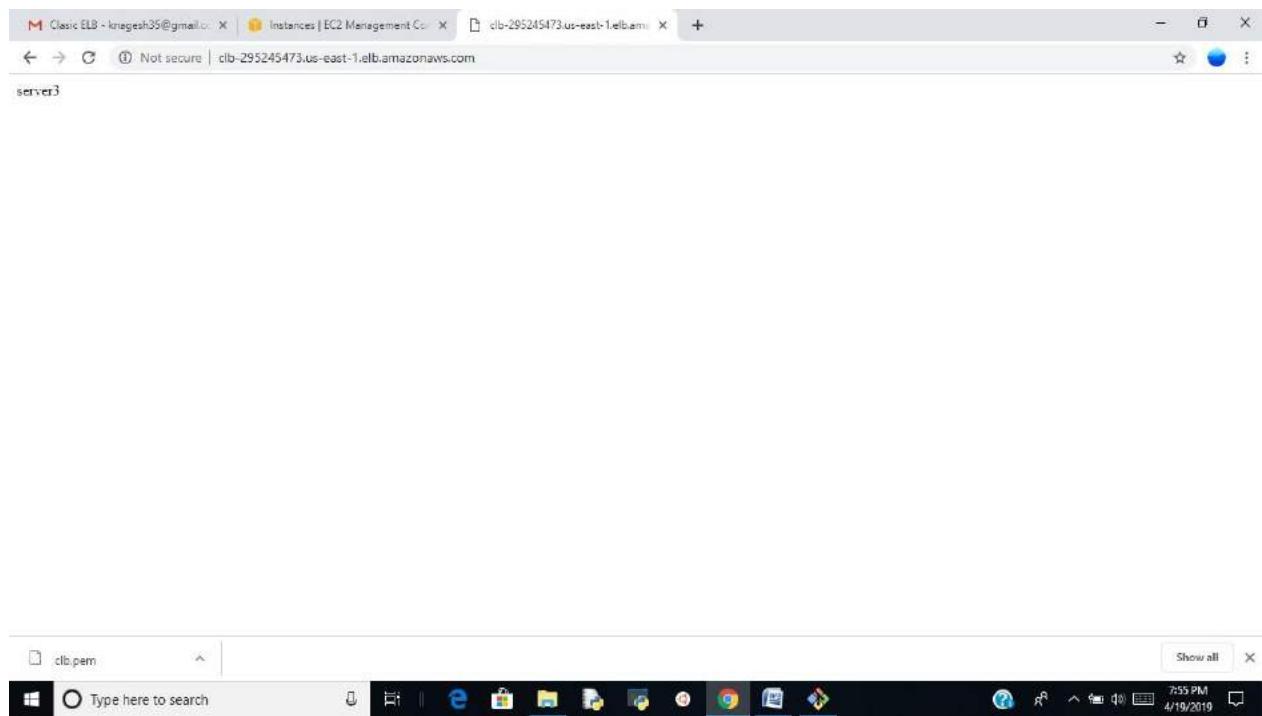


Then three Instances are inservice that is healthy machines

7. Refresh the page multiple time, You will Observe that load is distributing to three instance equally since first instance also come to healthy status

Amazon Web Services



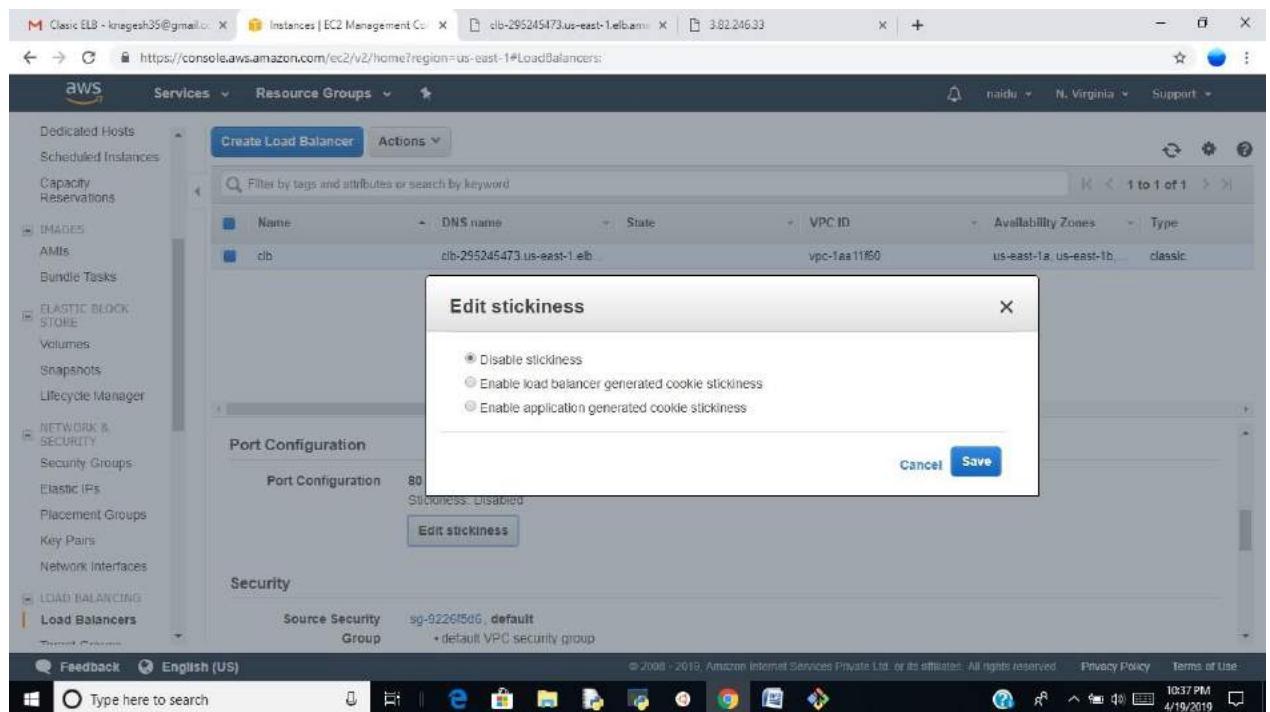


Uc3) Enable Stickiness in ELB

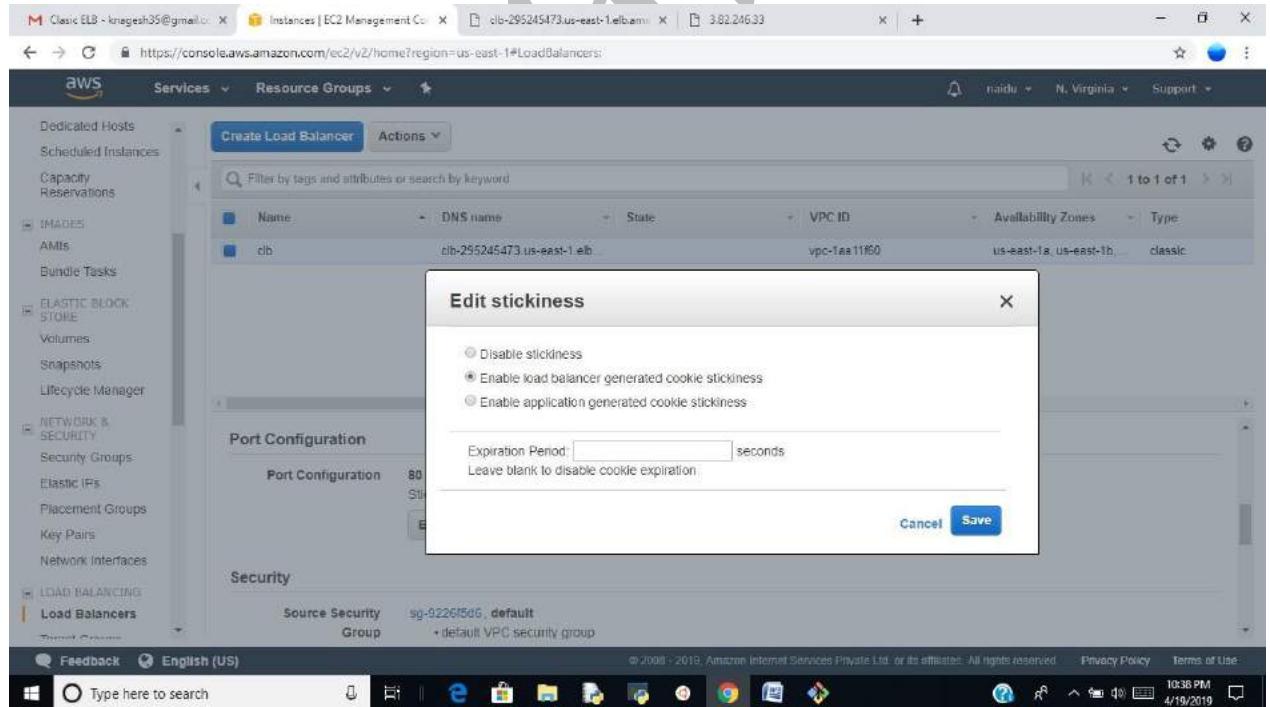
If Stickiness is enabled the load balancer to send a user's session request to a specific instance. This ensures that all requests from the user during the session are sent to the same instance.

1. Goto ELB Dash Board and Enable Stickiness

Click on ELB and goto elb description here go to port configuration then click on Edit stickiness

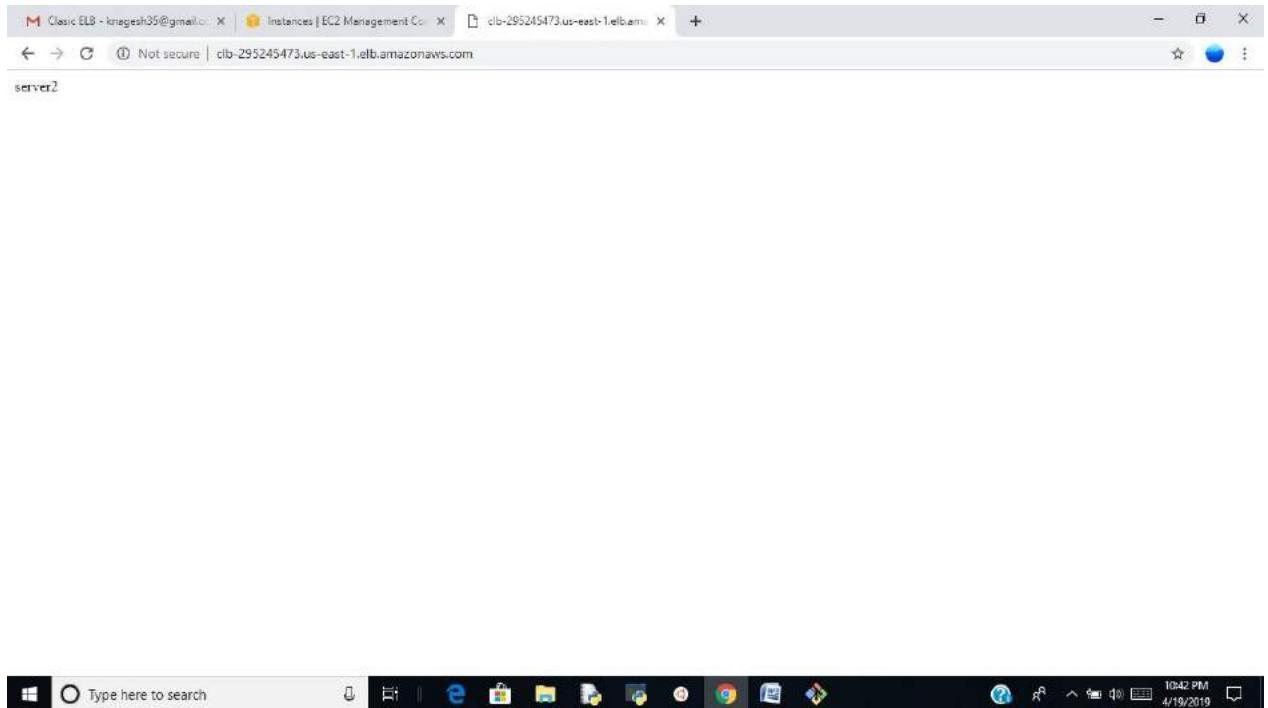


Then click on second option to enable stickiness



**Expiration Period: no give any value
Click on save**

2. Open the browser paste ELB DNS Name

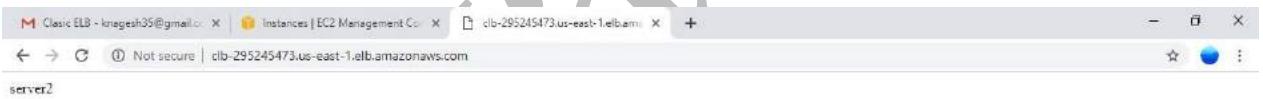


Request to server2

3. Refresh the page multiple time, You will Observe that request is going to specific server only that is server2



Again refresh

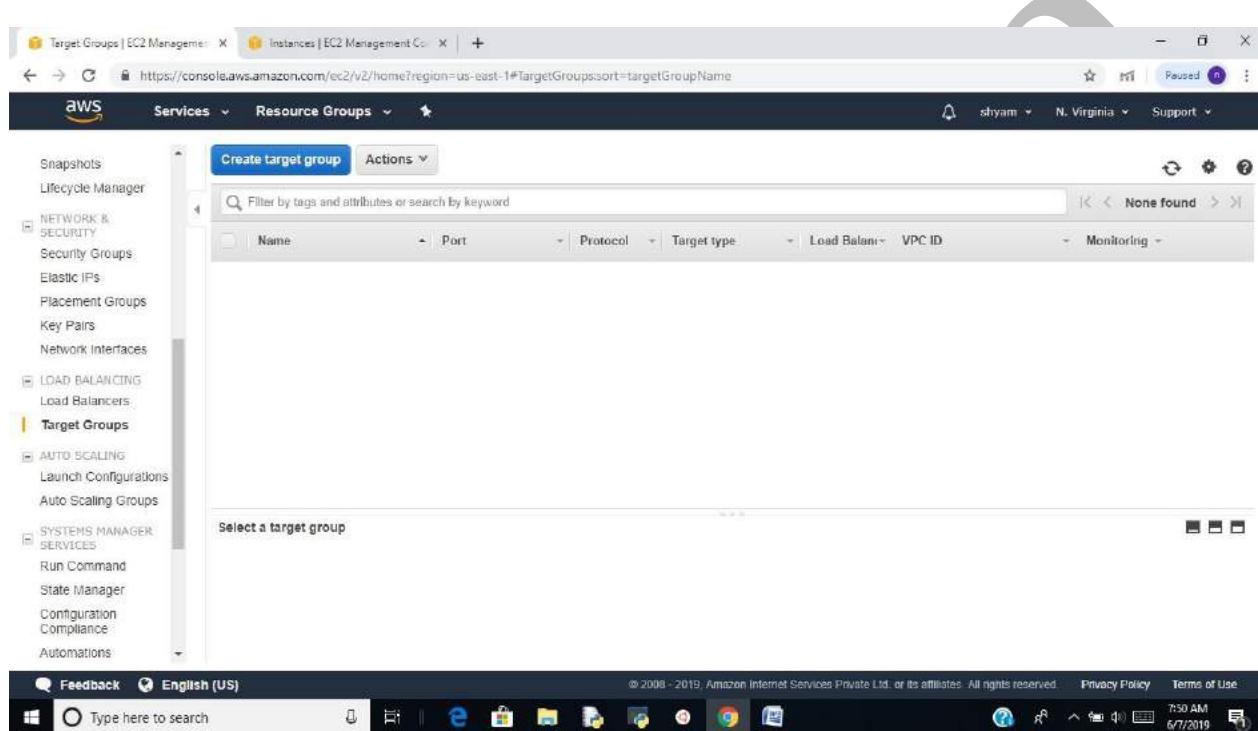


So when stickiness is enabled request send to particular server only

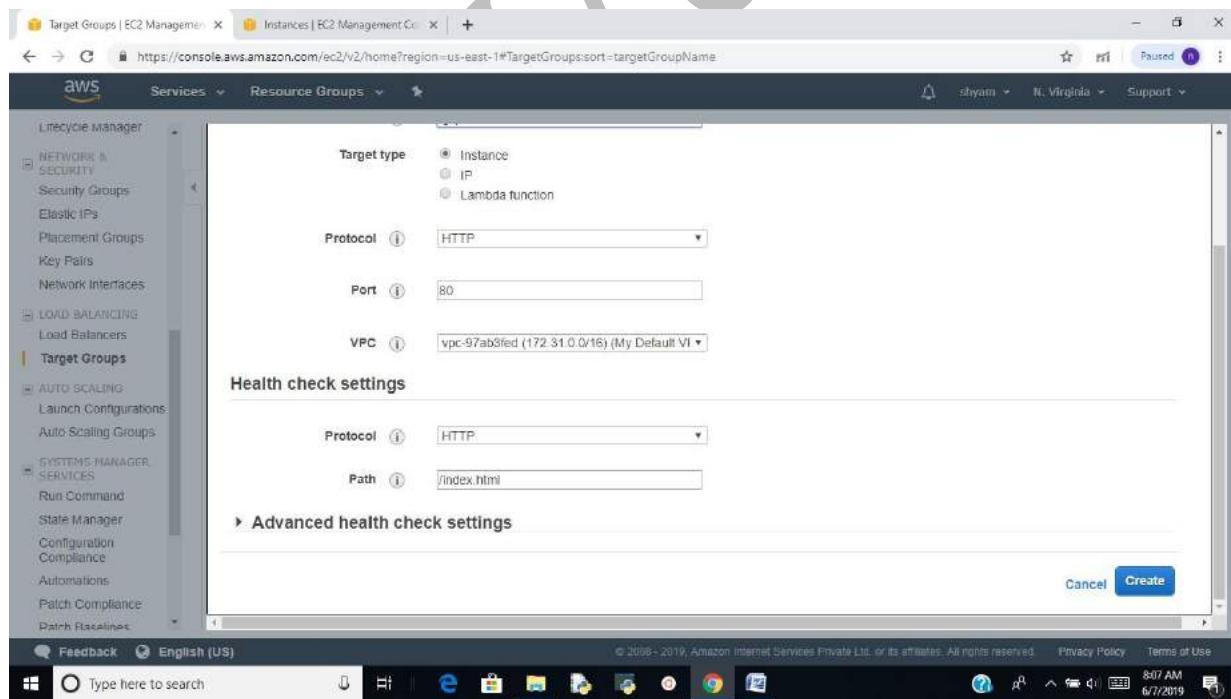
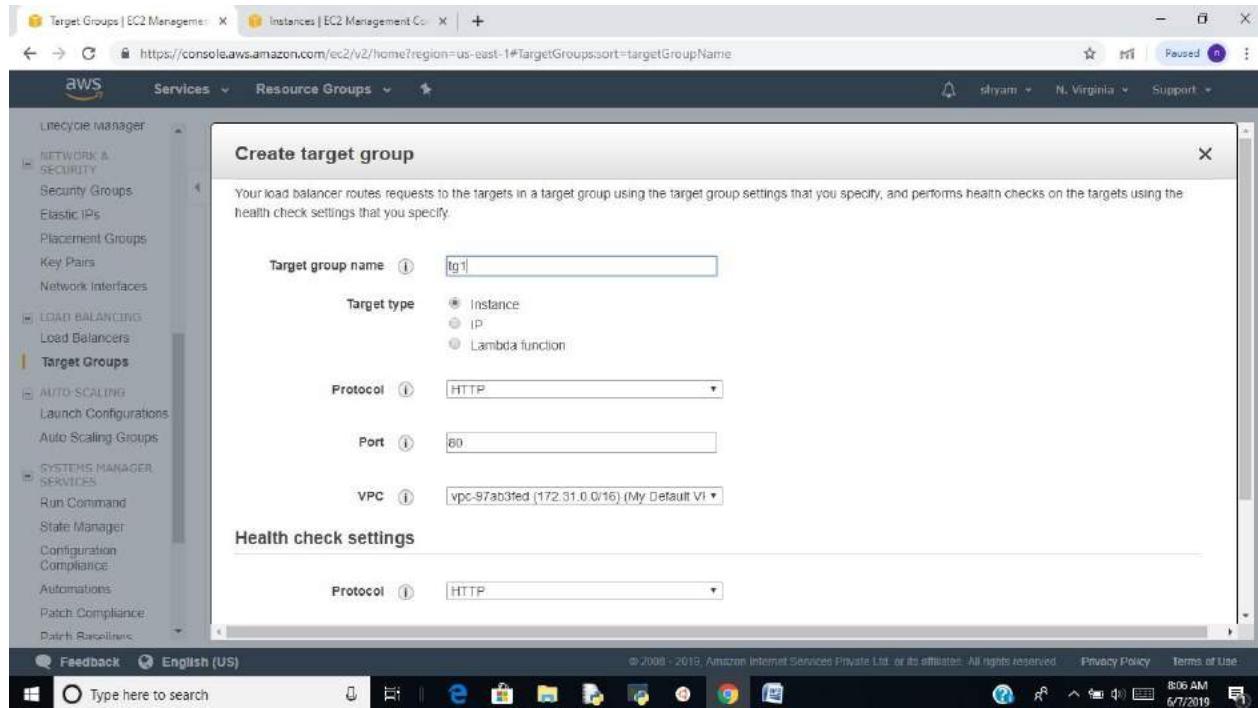
Application Load Balancer

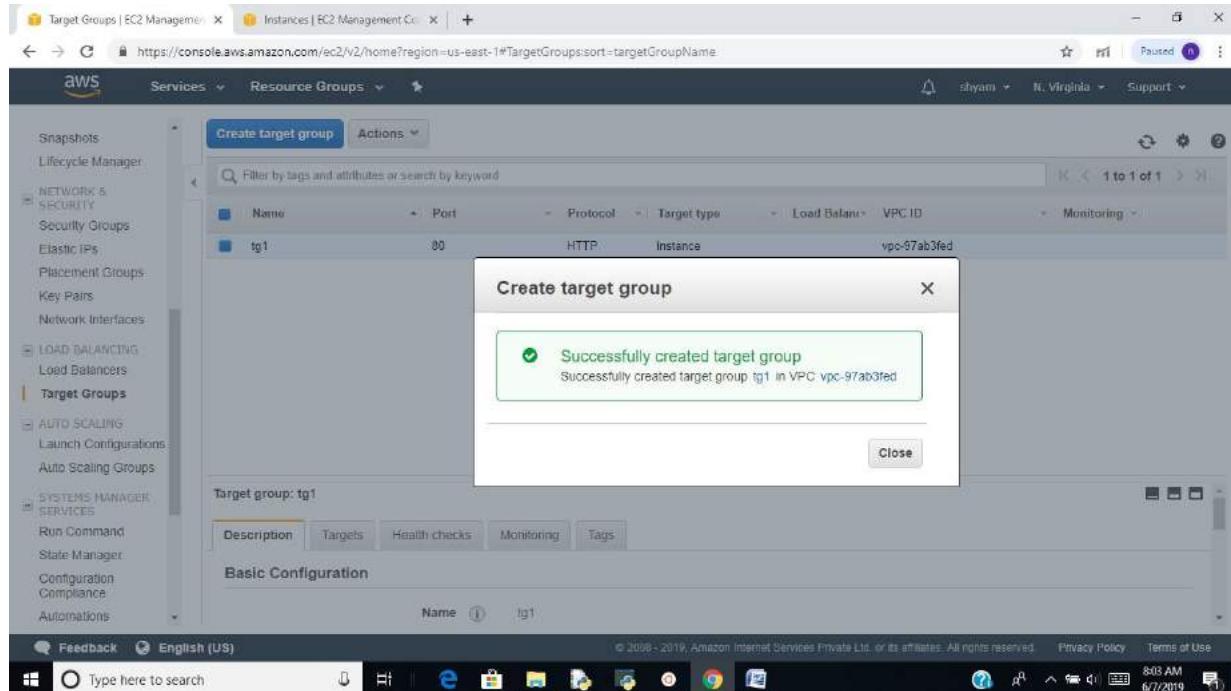
Creating Target Groups

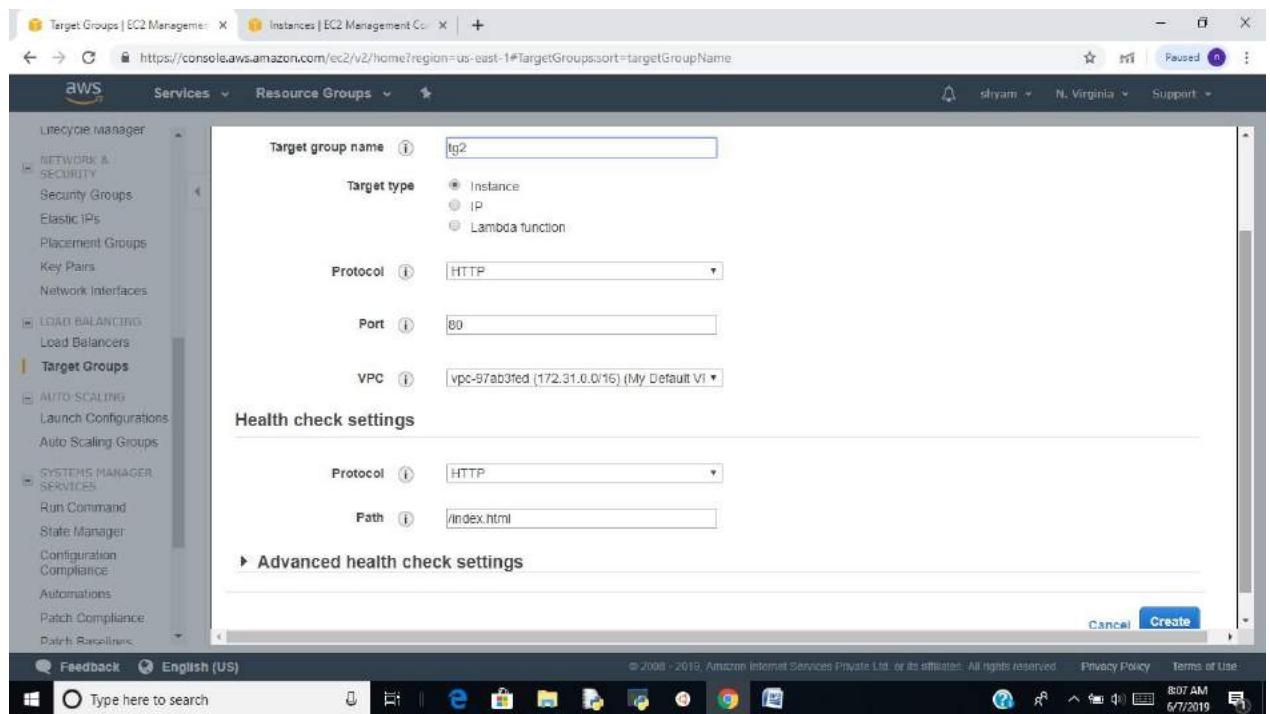
- Select EC2 service and click on Target groups in Load Balancer section of left side panel of EC2
- Click on create Target Group



- Target group name: enter name for target group
- Target type: select instance
- Protocol: HTTP
- Port: 80
- VPC: select vpc
- Health check settings: for health checks use HTTP protocol and path is /index.html
- Click on create

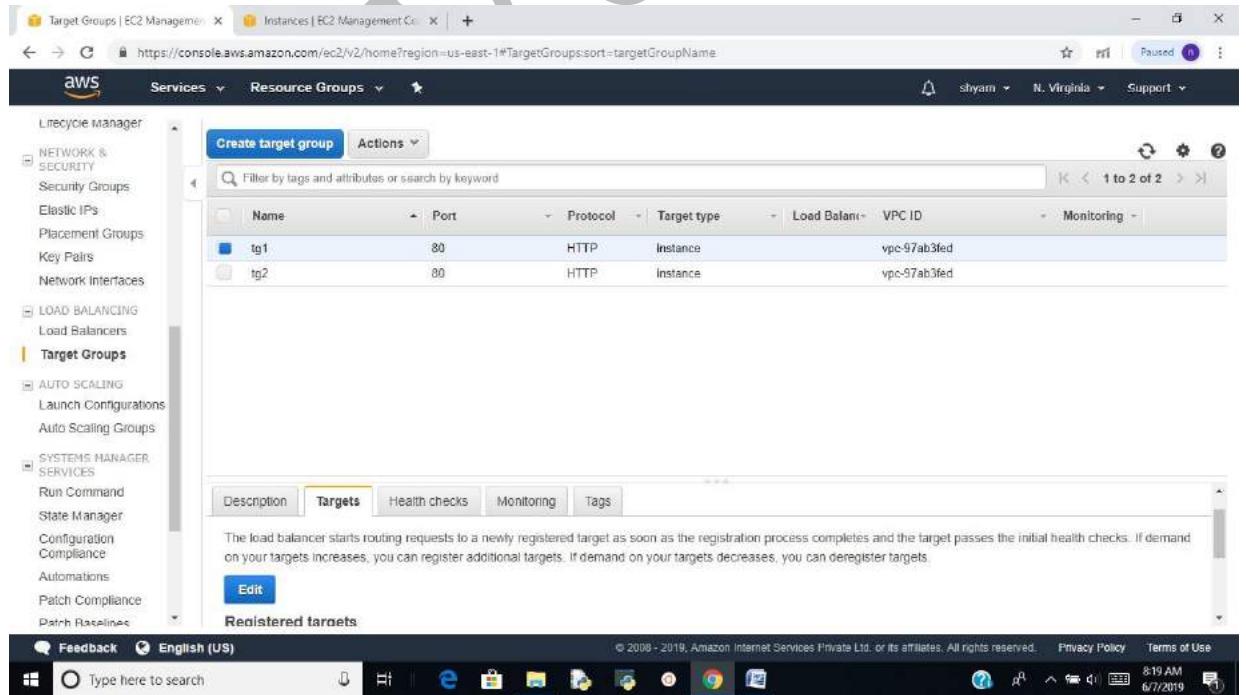






Add instances to Target group

- Select target group and click on Targets on below panel and click on Edit



- Select two Instances to add this target group and this instances have service that is running on 80 port. Ex apache2
- One instant you create one applicatin and another instance you create another application. Ex prepaid and postpaid
- Click on Add to register and click on save

The screenshot shows the AWS Management Console with the URL <https://console.aws.amazon.com/ec2/v2/home?region=us-east-1#targetGroups:sort=targetGroupName>. The left sidebar navigation includes 'Lifecycle Manager', 'NETWORK & SECURITY', 'Elastic IPs', 'Placement Groups', 'Key Pairs', 'Network Interfaces', 'LOAD BALANCING' (selected), 'Target Groups' (highlighted in yellow), 'AUTO SCALING', 'SYSTEMS MANAGER SERVICES', and 'Feedback'. The main content area displays a table of registered instances:

	Instance	Name	Port	State	Security groups	Zone
<input type="checkbox"/>	i-084873720d00fcf2		80	running	launch-wizard-13	us-east-1c

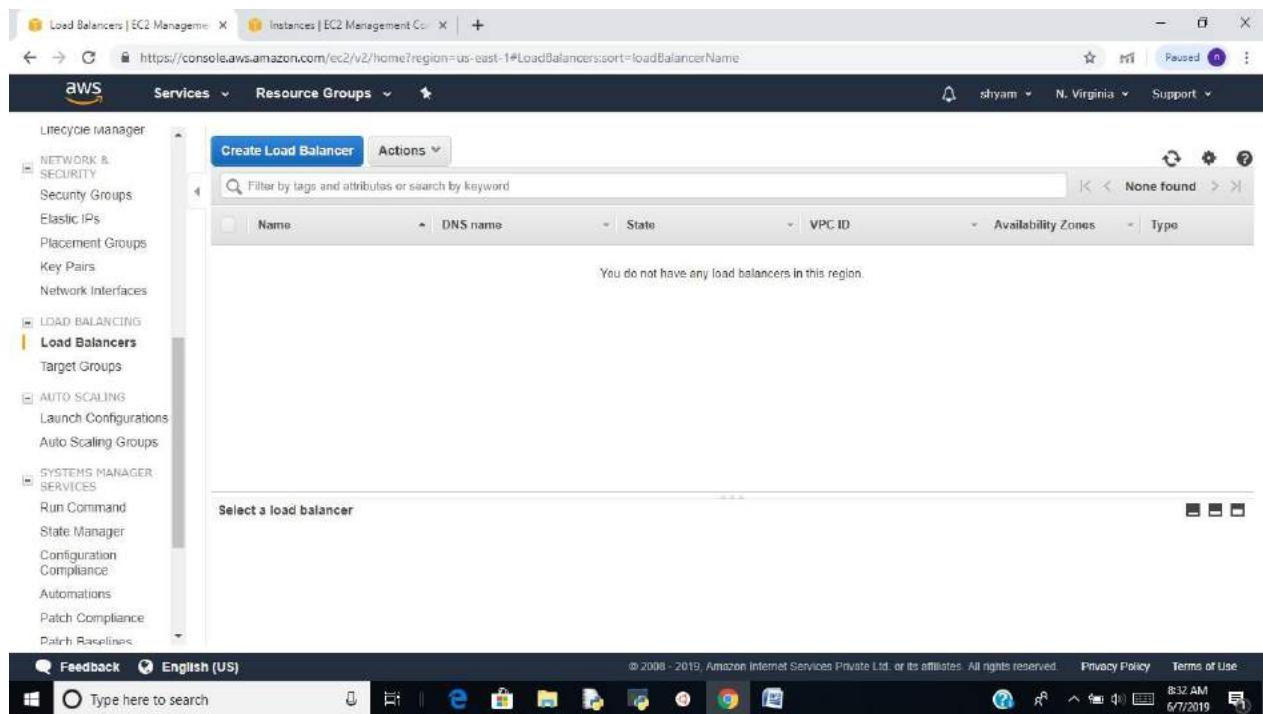
Below this is a section titled 'Instances' with instructions: 'To register additional instances, select one or more running instances, specify a port, and then click Add. The default port is the port specified for the target group. If the instance is already registered on the specified port, you must specify a different port.' A modal dialog box is open, titled 'Add to registered on port 80'. It contains a search bar 'Search Instances' and a table of available instances:

	Instance	Name	State	Security	Zone	Subnet ID	Subnet CIDR
<input checked="" type="checkbox"/>	i-084873720d...		running	launch-wizard...	us-east-1c	subnet-8df4b3a3	172.31.80.0/20
<input type="checkbox"/>	i-0e3ff4beab0c...		running	launch-wizard...	us-east-1c	subnet-8df4b3a3	172.31.80.0/20

At the bottom right of the dialog are 'Cancel' and 'Save' buttons.

Create Application Load Balancer

- Click on Load Balancers on left side panel of EC2 section and click on create Load Balancers



- Select Application Load Balancer and click on create
- Name: name for application load balancer
- Scheme: select internet-facing
- Ip address type: ipv4
- Listeners: Load Balancer Protocol: HTTP and Load Balancer port: 80
- Availability zones: select atleast two subnets on different availability zones
- Click on Next configure security settings

Step 1: Configure Load Balancer

Basic Configuration

To configure your load balancer, provide a name, select a scheme, specify one or more listeners, and select a network. The default configuration is an Internet-facing load balancer in the selected network with a listener that receives HTTP traffic on port 80.

Name	Only a-z, A-Z, 0-9 and hyphens are allowed
Scheme	<input checked="" type="radio"/> Internet-facing <input type="radio"/> Internal
IP address type	ipv4

Listeners

A listener is a process that checks for connection requests, using the protocol and port that you configured.

Load Balancer Protocol	Load Balancer Port
HTTP	80

Availability Zones

Specify the Availability Zones to enable for your load balancer. The load balancer routes traffic to the targets in these Availability Zones only. You can specify only one subnet per Availability Zone. You must specify subnets from at least two Availability Zones to increase the availability of your load balancer.

VPC	vpc-97ab3fed (172.31.0.0/16) (default)
Availability Zones	<input checked="" type="checkbox"/> us-east-1a subnet-10e9acdc <input checked="" type="checkbox"/> us-east-1b subnet-1cf1897b <input type="checkbox"/> us-east-1c subnet-8df4b3a3 <input type="checkbox"/> us-east-1d subnet-090cb443 <input type="checkbox"/> us-east-1e subnet-a3e36c9d <input type="checkbox"/> us-east-1f subnet-aa92a0a5

- Click on Next configure security groups

Step 2: Configure Security Settings

⚠ Improve your load balancer's security. Your load balancer is not using any secure listener.
If your traffic to the load balancer needs to be secure, use the HTTPS protocol for your front-end connection. You can go back to the first step to add/configure secure listeners under Basic Configuration section. You can also continue with current settings.

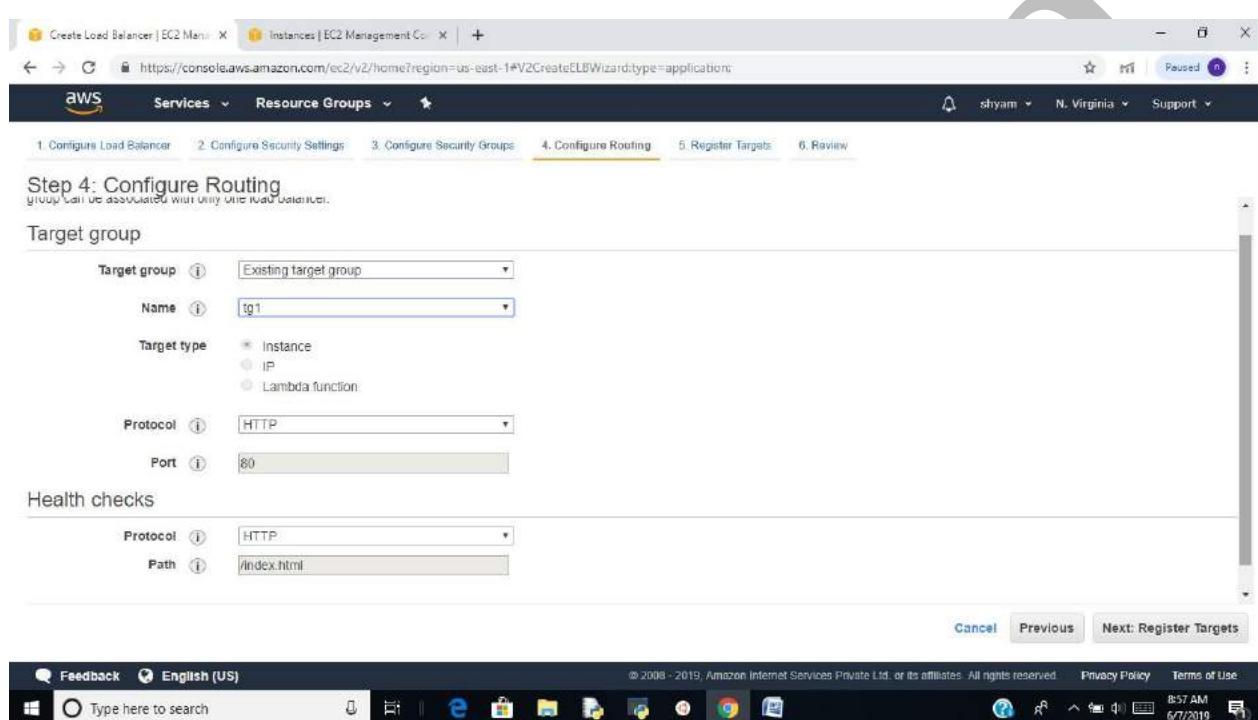
Step 3: Configure Security Groups
A security group is a set of firewall rules that control the traffic to your load balancer. On this page, you can add rules to allow specific traffic to reach your load balancer. First, decide whether to create a new security group or select an existing one.

Assign a security group:

- Create a new security group
- Select an existing security group

Security Group ID	Name	Description	Actions
sg-3f2f5679	default	default VPC security group	Copy to new
sg-04313f505ead7e28	launch-wizard-1	launch-wizard-1 created 2019-04-20T19:33:58.363+05:30	Copy to new
sg-0492ed76c0e29109a	launch-wizard-10	launch-wizard-10 created 2019-05-22T17:37:33.527+05:30	Copy to new
sg-0f1efdf22007a9174	launch-wizard-11	launch-wizard-11 created 2019-06-02T12:33:44.102+05:30	Copy to new
sg-03e2687ecfffe1957e	launch-wizard-12	launch-wizard-12 created 2019-06-03T13:39:24.197+05:30	Copy to new
sg-07df357b308fb099	launch-wizard-13	launch-wizard-13 created 2019-06-07T07:49:43.661+05:30	Copy to new
sg-003aa344ab328dbb6	launch-wizard-2	launch-wizard-2 created 2019-05-03T12:05:15.997+05:30	Copy to new
sg-0d29bf81dd65e777e	launch-wizard-3	launch-wizard-3 created 2019-05-06T10:41:46.673+05:30	Copy to new
sg-0611f37a0d735c372	launch-wizard-4	launch-wizard-4 created 2019-05-15T15:42:47.597+05:30	Copy to new

- Target group: select one target group previously created
- Name: Target Group name
- Target Type: Instance
- Protocol: HTTP
- Port: 80
- Health checks: protocol HTTP and path is /index.html
- Click on next register targets



- Click on next review

Amazon Web Services

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

The following targets are registered with the target group that you selected. You can only modify this list after you create the load balancer.

Instance	Port
i-084879720d0001cf2	80

Cancel Previous Next: Review

- Click on create

Step 6: Review

Please review the load balancer details before continuing.

Load balancer

- Name: myal
- Scheme: Internet-facing
- Listeners: Port 80 - Protocol HTTP
- IP address type: IPv4
- VPC: vpc-97abc0fd
- Subnets: subnet-10e9ac4c, subnet-1cf1897b
- Tags:

Security groups

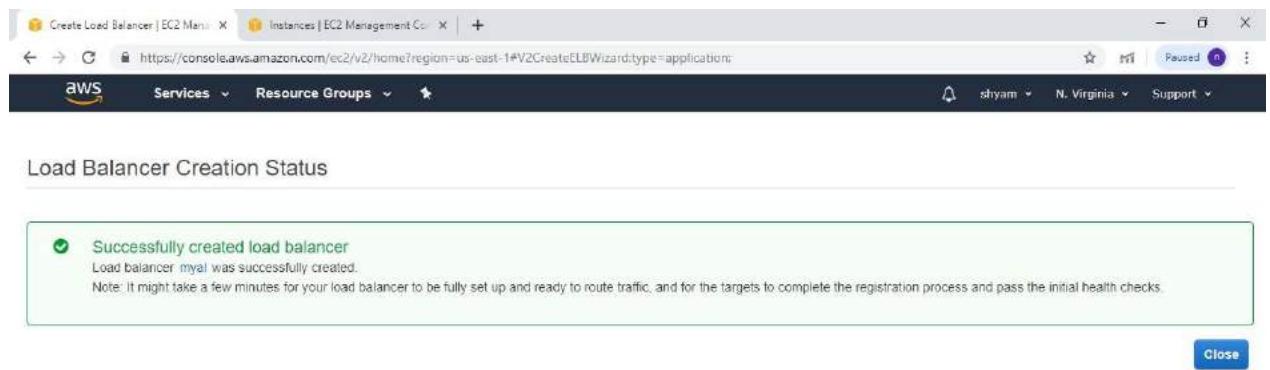
- Security groups: sg-3f2f5679

Routing

- Target group: Existing target group
- Target group name: tg1
- Port: 80
- Target type: Instance

Cancel Previous Create

- Click on close

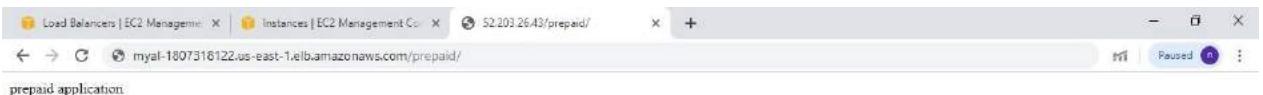


Test Application Load Balancer

- Select the application load balancer and goto description in below panel and copy the dns url and paste in browser

Name	DNS name	State	VPC ID	Availability Zones	Type
myal	myal-1807318122.us-east-1.elb.amazonaws.com	active	vpc-97ab3fed	us-east-1a, us-east-1b	application

- Based on client request that which applicatin name in include in request that is forward to particular application
- Client first include prepaid word in request then redirect to prepaid application



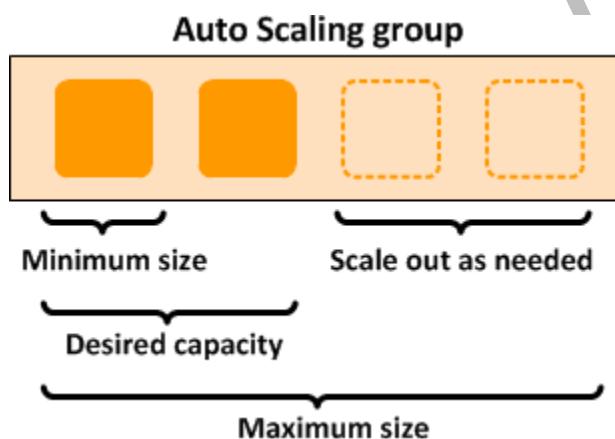
- Second time client include postpaid keyword in his request then redirected to post paid application



Auto Scaling

Amazon EC2 Auto Scaling helps you ensure that you have the correct number of Amazon EC2 instances available to handle the load for your application. You create collections of EC2 instances, called *Auto Scaling groups*. You can specify the minimum number of instances in each Auto Scaling group, and Amazon EC2 Auto Scaling ensures that your group never goes below this size. You can specify the maximum number of instances in each Auto Scaling group, and Amazon EC2 Auto Scaling ensures that your group never goes above this size. If you specify the desired capacity, either when you create the group or at any time thereafter, Amazon EC2 Auto Scaling ensures that your group has this many instances. If you specify scaling policies, then Amazon EC2 Auto Scaling can launch or terminate instances as demand on your application increases or decreases.

For example, the following Auto Scaling group has a minimum size of one instance, a desired capacity of two instances, and a maximum size of four instances. The scaling policies that you define adjust the number of instances, within your minimum and maximum number of instances, based on the criteria that you specify.



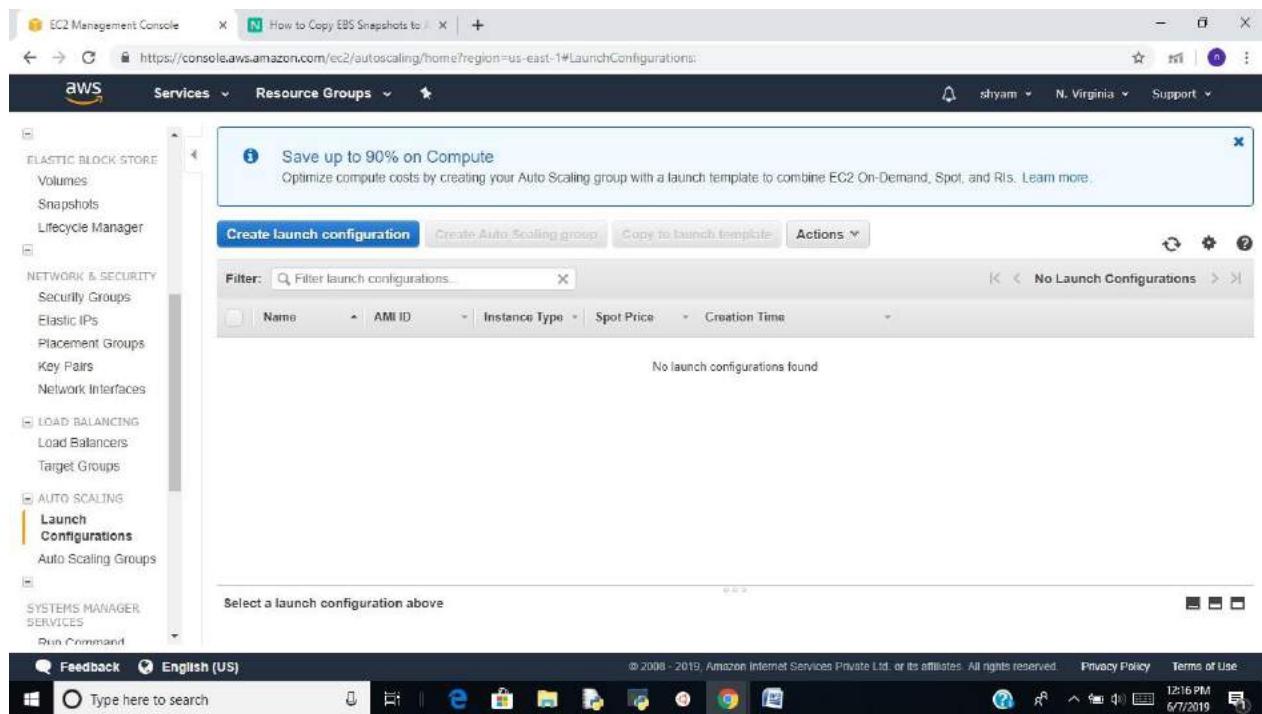
Auto Scaling Components

The following table describes the key components of Amazon EC2 Auto Scaling.

	<h3>Groups</h3> <p>Your EC2 instances are organized into <i>groups</i> so that they can be treated as a logical unit for the purposes of scaling and management. When you create a group, you can specify its minimum, maximum, and, desired number of EC2 instances.</p>
	<h3>Configuration templates</h3> <p>Your group uses a <i>launch template</i> or a <i>launch configuration</i> as a configuration template for its EC2 instances. You can specify information such as the AMI ID, instance type, key pair, security groups, and block device mapping for your instances.</p>
	<h3>Scaling options</h3> <p>Amazon EC2 Auto Scaling provides several ways for you to scale your Auto Scaling groups. For example, you can configure a group to scale based on the occurrence of specified conditions (dynamic scaling) or on a schedule.</p>

Create Launch Configuration

- Select EC2 service and click on Launch configurations on left side panel EC2 in auto scaling section
- Click on Create Launch Configuration



- Choose Ami: select amazon machine image for launching EC2 (I select ubuntu 16)
- Click on next
- Choose Instance Type: select t2.micro
- Click on next
- Configure details: Launch configuration name, purchasing option is disable, IAM role select IAM role if you want to apply, enable cloud watch monitoring if do you want to monitor
- Click on Next Add storage

Create Launch Configuration

Name:

Purchasing option: Request Spot Instances

IAM role:

Monitoring: Enable CloudWatch detailed monitoring

Later, if you want to use a different launch configuration, you can create a new one and apply it to any Auto Scaling group. Existing launch configurations cannot be edited.

Cancel Previous Skip to review Next: Add Storage

- Add storage: By default root volume is attached it is 8GB. If do want to increase the size then increase or do you want to external volume you can add
- Click on Next configure security group

Create Launch Configuration

Your instance will be launched with the following storage device settings. You can attach additional EBS 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.

<https://docs.aws.amazon.com/console/ec2/launchinstance/storage> about storage options in Amazon EC2.

Volume Type	Device	Snapshot	Size (GiB)	Volume Type	IOPS	Throughput	Delete on Termination	Encrypted
Root	/dev/sda1	snap-0e364461033d3767e	8	General Purpose (SSD)	100 / 3000	N/A	<input checked="" type="checkbox"/>	No

Add New Volume

Free tier eligible customers can get up to 30 GB of EBS storage. [Learn more](#) about free usage tier eligibility and usage restrictions.

Cancel Previous Skip to review Next: Configure Security Group

- Configure security group: by default ssh protocol only include this security group do you want add another rule you can add and you can also select existing security group
- Click on review

Create Launch Configuration

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. [Learn more](#) about Amazon EC2 security groups.

Assign a security group:

- Create a new security group
- Select an existing security group

Security group name: AutoScaling-Security-Group-1

Description: AutoScaling-Security-Group-1 (2019-06-07 13:03:36.898+05:30)

Type	Protocol	Port Range	Source
SSH	TCP	22	Anywhere (0.0.0.0/0)

Add Rule

Warning

Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

Review

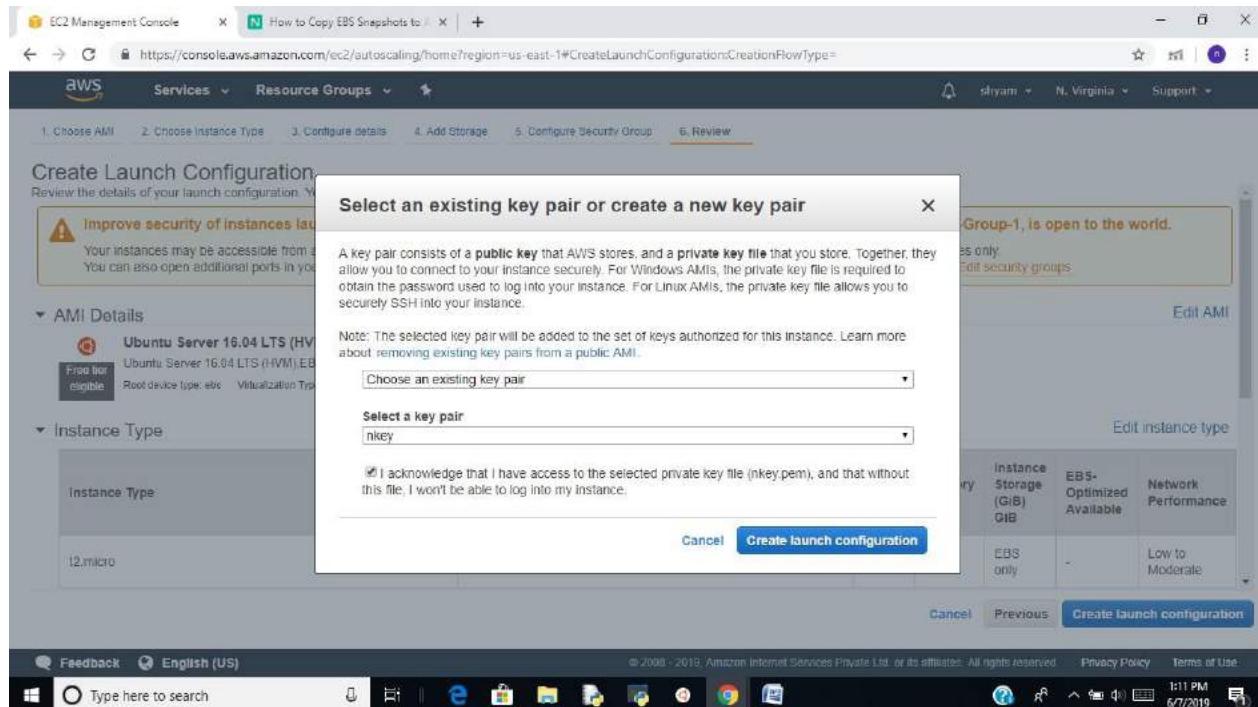
Launch Configuration Summary

AMI: Amazon Linux 2019.03.1 (HVM, SSD) - Latest Version
 Instance Type: t2.micro
 Key Name: DyaSaa-Test-1
 Security Groups: AutoScaling-Security-Group-1
 Root Volume: 8 GiB (General Purpose SSD)
 Block Device Mapping: /dev/sda1 (8 GiB, General Purpose SSD)
 Network Interfaces: eth0 (Private IP: 172.31.10.11, Public IP: 52.16.111.11, Subnet: 172.31.10.0/24, VPC: vpc-00000000, AZ: us-east-1a)
 IAM Role: Lambda@Edge (AWS Lambda Role)
 Tags: None

Next Step

Launch instance

- Click on create launch configuration
- Key Pair: select existing key pair or create new key pair
- Enable acknowledgement and click on create launch configuration

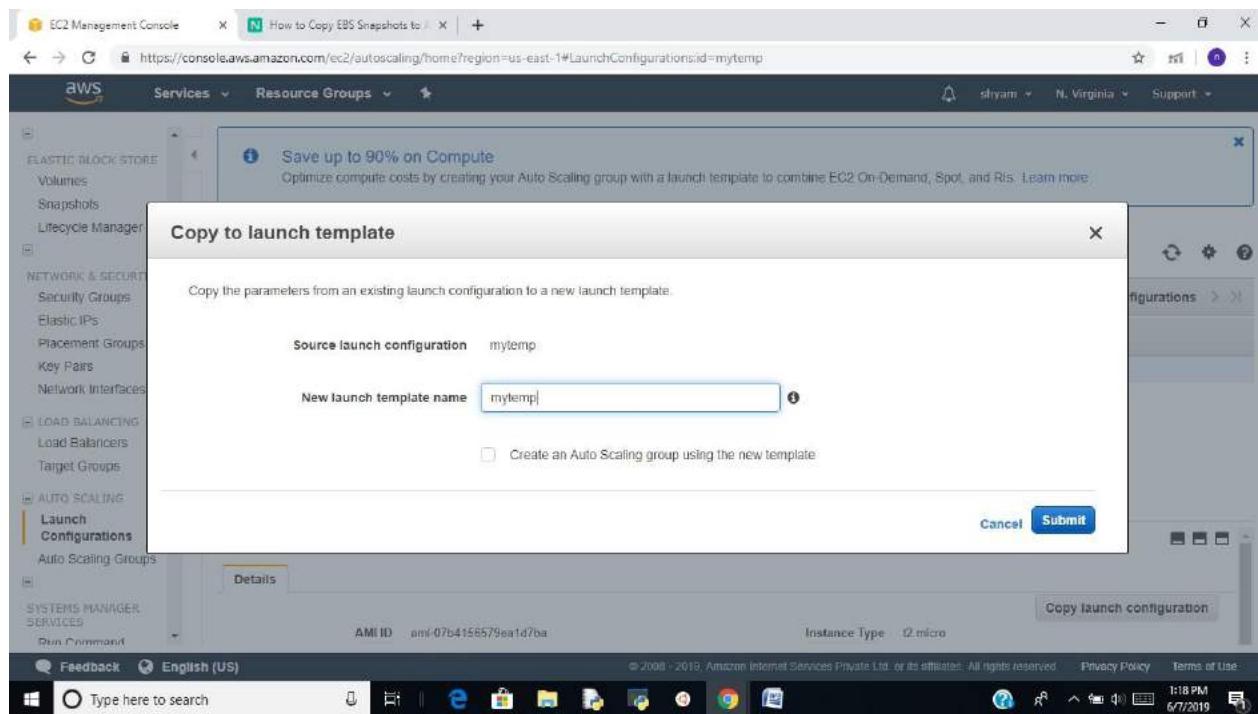


- Click on close

The screenshot shows a browser window with the AWS EC2 Management Console. The URL is <https://console.aws.amazon.com/ec2/autoscaling/home?region=us-east-1#CreateLaunchConfigurationCreationFlowType=copy>. The page title is "How to Copy EBS Snapshots to..." and the sub-page title is "Launch configuration creation status". A green success message box contains the text "Successfully created launch configuration: mytemp" and a link "View creation log". Below this, there are sections for "View" (with links to "View your launch configurations" and "View your Auto Scaling groups"), "Helpful resources" (with a link to "Here are some helpful resources to get you started"), and a button "Create an Auto Scaling group using this launch configuration" with a "Close" button.

Copy to launch template

- This option is used to create new template from existing template
- Click on copy to launch template
- Enter new template name
- Click on submit



Delete launch configuration

- Select template and goto actions and click on delete launch configuration

Amazon Web Services

The screenshot shows the AWS Management Console for the EC2 service. The left sidebar has 'LAUNCH CONFIGURATIONS' selected under 'AUTO SCALING'. The main area displays a table of launch configurations. A context menu is open over the first row, with 'Delete launch configuration' highlighted. The table shows one entry:

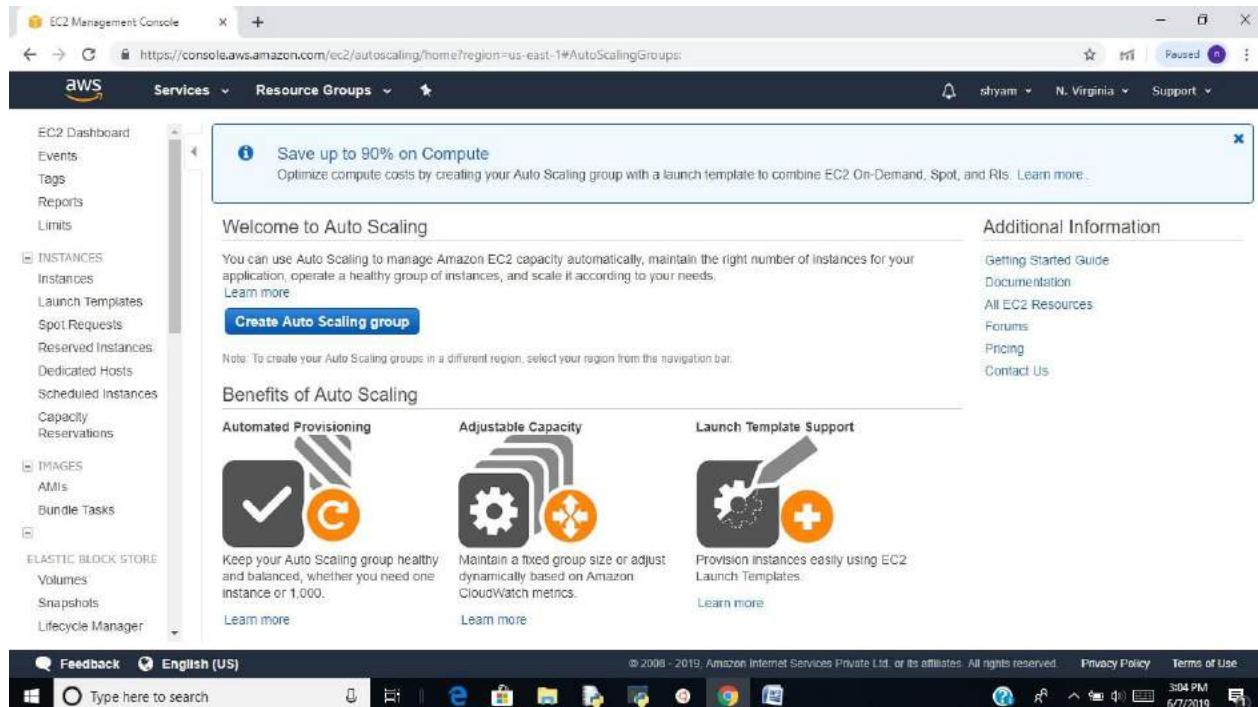
Name	AMI ID	Instance Type	Creation Time
mytemp	ami-07b41565	t2.micro	June 7, 2019 at 1:12:18 PM UT...

□ Click on yes delete

The screenshot shows the same AWS Management Console interface. A modal dialog box titled 'Delete launch configuration' is centered on the screen. It contains the message 'Are you sure you want to delete this resource? mytemp' and two buttons: 'Cancel' and 'Yes, Delete'. The background shows the same launch configuration list as the previous screenshot.

Create Auto scaling Group

- Click on auto scaling groups on left side panel of EC2 section
- Click on create auto scaling group



- Select launch configuration ad choose one template
- Click on next step

The screenshot shows the AWS EC2 Management Console interface for creating an Auto Scaling Group. It consists of three main sections:

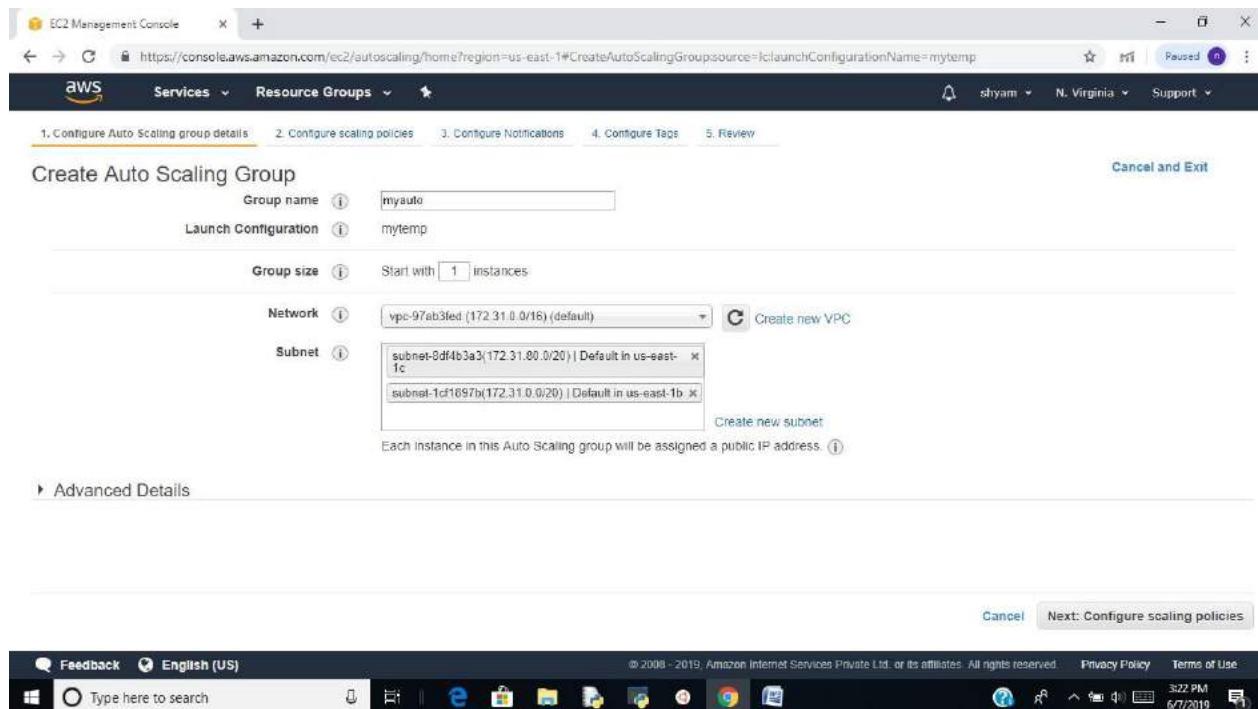
- Step 1: Launch Configuration**: The user has selected this option. A note says: "You can continue to use your launch configurations if they support the Amazon EC2 features you need. [Learn more](#)". A button "Create a new launch configuration" is present.
- Step 2: Launch Template New**: This option is also available. A note says: "Launch templates give you the option of launching one type of instance, or a combination of instance types and purchase options. Launch templates include the latest Amazon EC2 features and can be updated and versioned. [Learn more](#)". A button "Create new launch template" is present.
- Step 3: Create Auto Scaling Group**: This is the final step where the user can review and submit the configuration. It shows a summary of the selected launch configuration "mytemp".

At the bottom, there are "Cancel" and "Next Step" buttons. The browser status bar indicates the URL is <https://console.aws.amazon.com/ec2/autoscaling/home?region=us-east-1#CreateAutoScalingGroupsource=wizard>.

Configure auto scaling group details:

- Group name: auto scaling group name
- Launch configuration: selected in previous step

- Group size: it is the desired value. How many instances are maintained initially
- Network: vpc to launch EC2 instances
- Subnet: select subnet in above VPC
- Click on next configure scaling policies



Configure Scaling Policies:

- Select keep this at its group initial size
- Use scaling policies to adjust the capacity of this size
- Scales between here enter maximum and minimum number of instances
- Name: enter policy name
- Metric Type: select Metric (Average CPU Utilization)
- Target Value: enter cpu utilization percentage value
- Instances need: enter instance scale warm up period
- Disable scale-in: not select
- Here we select first option only
- Click on Next Configure Notifications

Create Auto Scaling Group

You can optionally add scaling policies if you want to adjust the size (number of instances) of your group automatically. A scaling policy is a set of instructions for making such adjustments in response to an Amazon CloudWatch alarm that you assign to it. In each policy, you can choose to add or remove a specific number of instances or a percentage of the existing group size, or you can set the group to an exact size. When the alarm triggers, it will execute the policy and adjust the size of your group accordingly. Learn more about scaling policies.

Keep this group at its initial size
 Use scaling policies to adjust the capacity of this group

Cancel Previous Review Next: Configure Notifications

Create Auto Scaling Group

You can optionally add scaling policies if you want to adjust the size (number of instances) of your group automatically. A scaling policy is a set of instructions for making such adjustments in response to an Amazon CloudWatch alarm that you assign to it. In each policy, you can choose to add or remove a specific number of instances or a percentage of the existing group size, or you can set the group to an exact size. When the alarm triggers, it will execute the policy and adjust the size of your group accordingly. Learn more about scaling policies.

Keep this group at its initial size
 Use scaling policies to adjust the capacity of this group

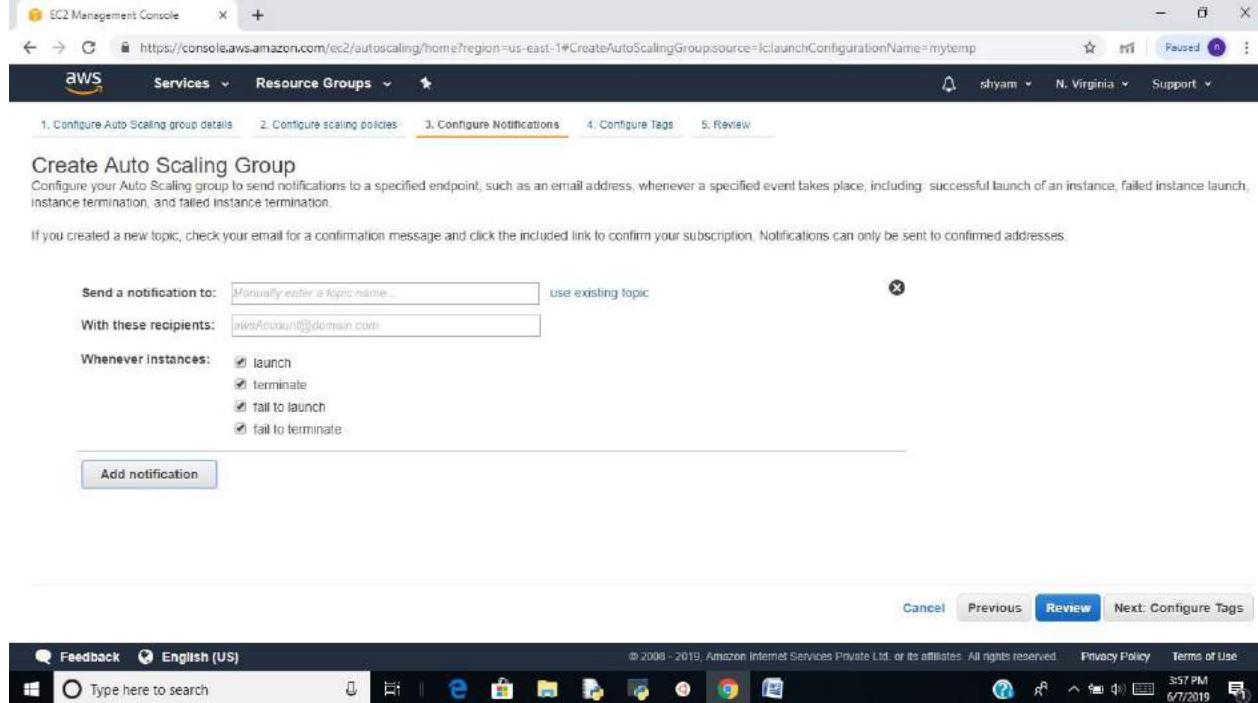
Scale between and instances. These will be the minimum and maximum size of your group.

Scale Group Size

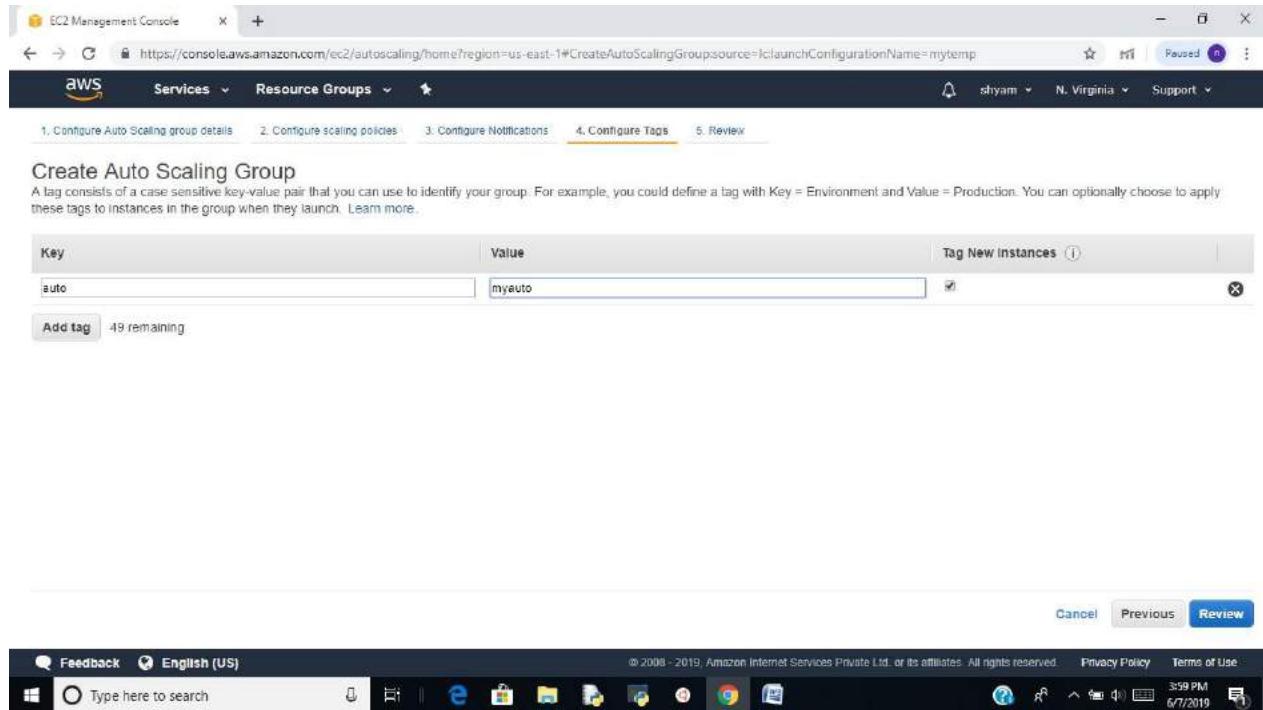
Name: Scale Group Size
Metric type: Average CPU Utilization
Target value:
Instances need: 1000 seconds to warm up after scaling
Disable scale-in:

Cancel Previous Review Next: Configure Notifications

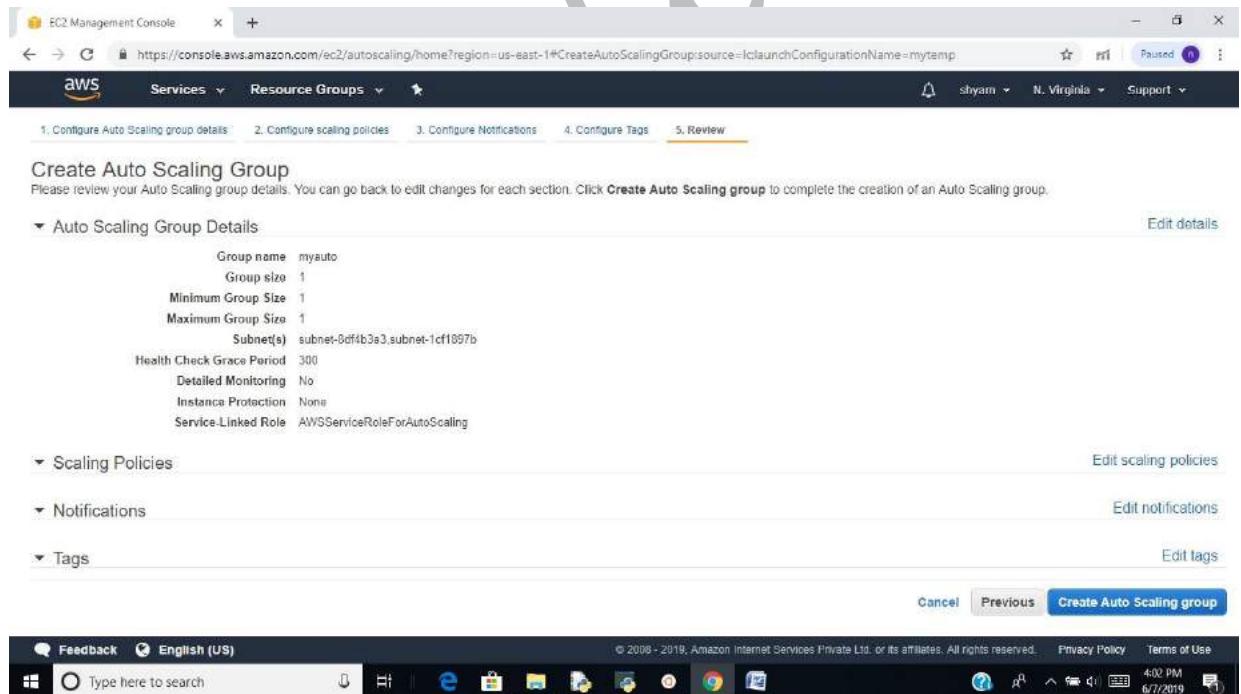
- Add Notification: add notification for this auto scaling activities that is send notification when scale in or scale out
- This is the optional so you can skip also
- If do you want add notification then add by using endpoint option
- Send notification to: SNS topic
- With these recipients: receivers email address
- Whenever instances: select actions that is launch, terminate, fail to launch, fail to terminate....etc
- Click on next configuration tags



- Configure Tags: enter key value pair for Tags and it is the optional
- Click on Review



- Check all configuration details and click on create auto scaling group



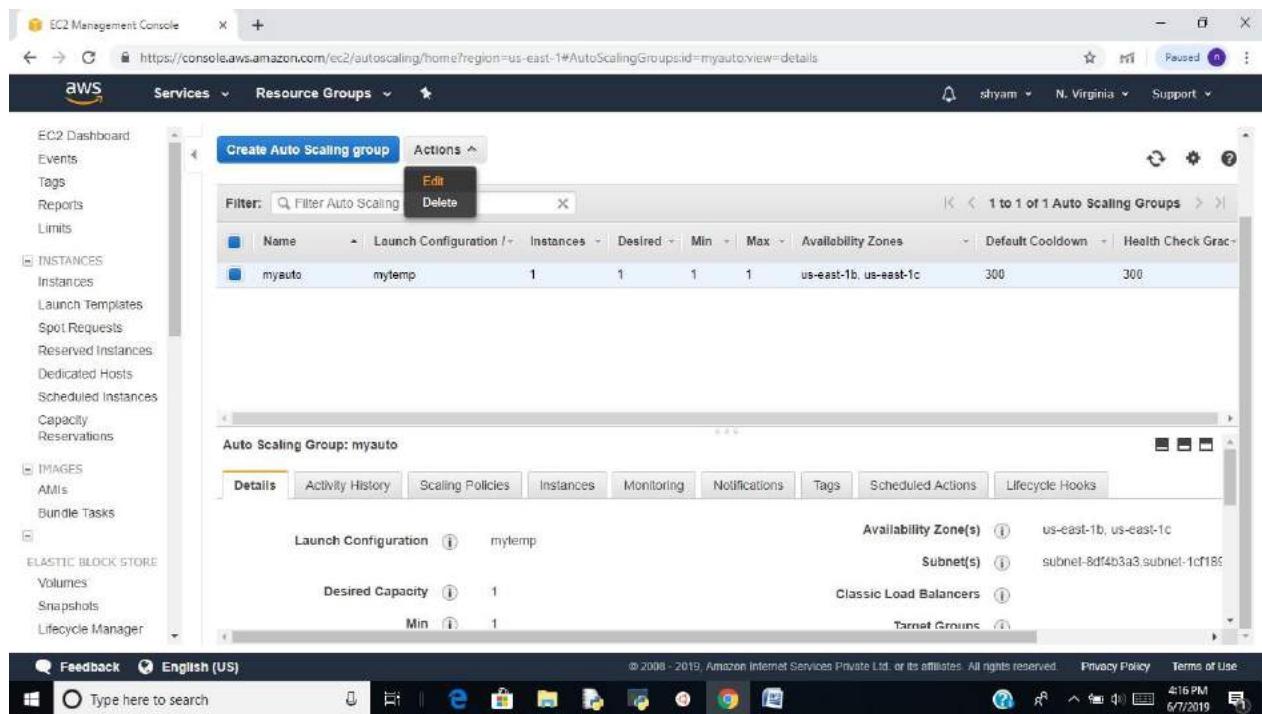
- Click on close

The screenshot shows the AWS EC2 Management Console with the URL <https://console.aws.amazon.com/ec2/autoscaling/home?region=us-east-1#CreateAutoScalingGroupsource=lc-launchConfigurationName=mytemp>. The top navigation bar includes 'Services' (selected), 'Resource Groups', and user information 'shyam N. Virginia Support'. A green success message box contains the text 'Successfully created Auto Scaling group' and a link 'View creation log'. Below the message, there are sections for 'View' (links to 'View your Auto Scaling groups' and 'View your launch configurations') and 'Helpful resources' (link to 'Here are some helpful resources to get you started'). A 'Close' button is located in the bottom right corner of the message box.

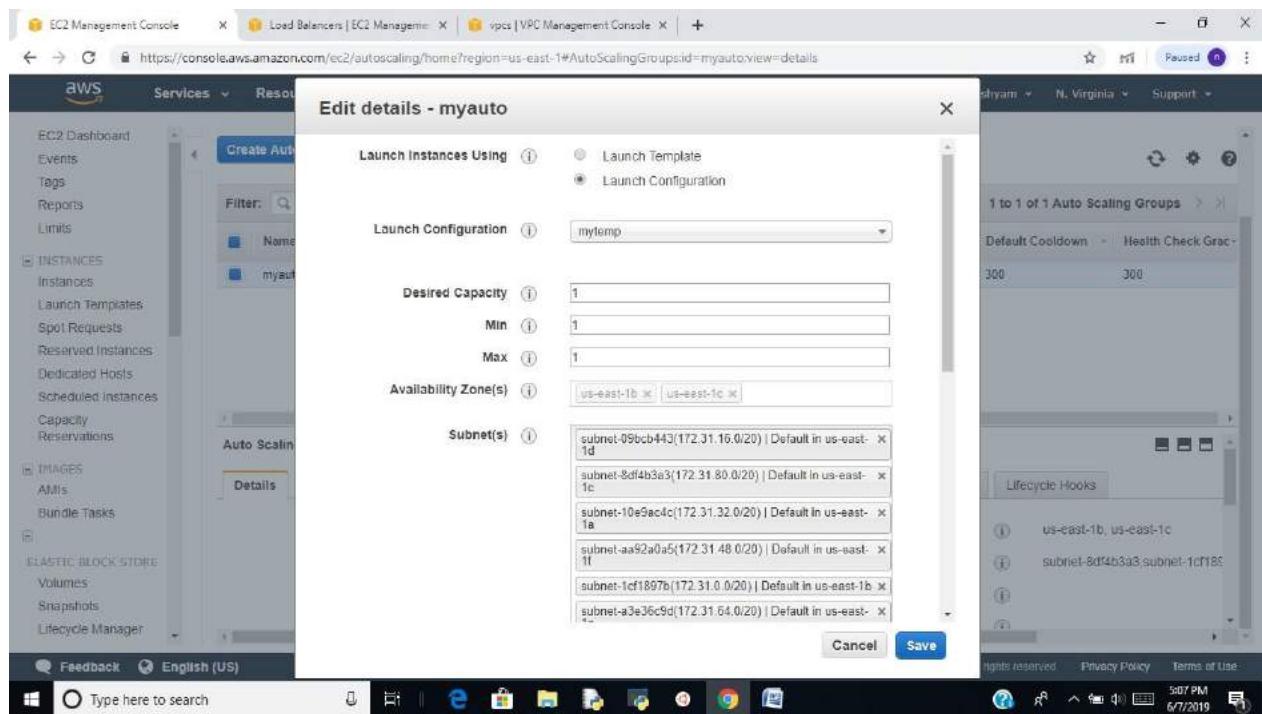


Edit Auto scaling Group

- Select auto scaling group and goto actions and click on Edit



- Desired capacity: you can change this desired capacity but the value with in minimum and maximum values
- Min: edit this minimum value
- Max: edit this Max value
- You can also add and remove availability zones and subnets in that vpc
- Classical Load Balancers: you can add load balancer to this auto scaling group
- Target Groups: attach target group
- Health check type: EC2
- Health check grace period: 300
- Edit all required details and click on save



Delete auto scaling group

- Select auto scaling group and goto actions and click on delete

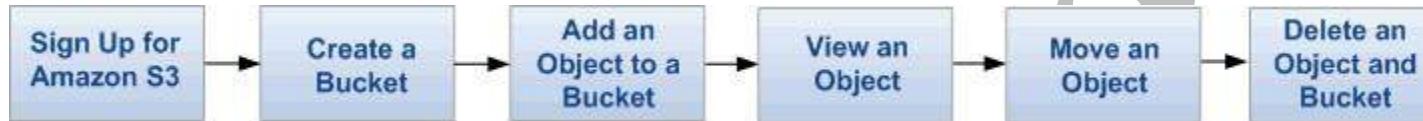
The screenshot shows the AWS Auto Scaling Groups page. On the left, there's a sidebar with navigation links like EC2 Dashboard, Instances, Launch Templates, and so on. The main content area displays a table for the 'myauto' Auto Scaling group. The table has columns for Name, Launch Configuration / Template, Instances, Desired, Min, Max, Availability Zones, Default Cooldown, and Health Check Grace Period. One row is shown with the values: myauto, mytemp, 1, 1, 1, us-east-1b, us-east-1c, 300, and 300. Below the table, there's a detailed view of the single instance, showing its Instance ID (I-090c552ab131dd871), Lifecycle (InService), Launch Configuration / Template (mytemp), Availability Zone (us-east-1c), and Health Status (Healthy). A large watermark 'Dyasa' is overlaid across the center of the screen.

- Click on yes, delete

This screenshot is similar to the previous one but includes a modal dialog box titled 'Delete Auto Scaling group'. The dialog asks 'Are you sure you want to delete this resource?' with 'myauto' listed below it. At the bottom of the dialog are 'Cancel' and 'Yes, Delete' buttons. The background shows the same Auto Scaling group details as the first screenshot. A large watermark 'Dyasa' is overlaid across the center of the screen.

Amazon Simple Storage Service (S3)

Amazon Simple Storage Service (Amazon S3) is storage for the Internet. You can use Amazon S3 to store and retrieve any amount of data at any time, from anywhere on the web. You can accomplish these tasks using the AWS Management Console, which is a simple and intuitive web interface. This guide introduces you to Amazon S3 and how to use the AWS Management Console to complete the tasks shown in the following figure.

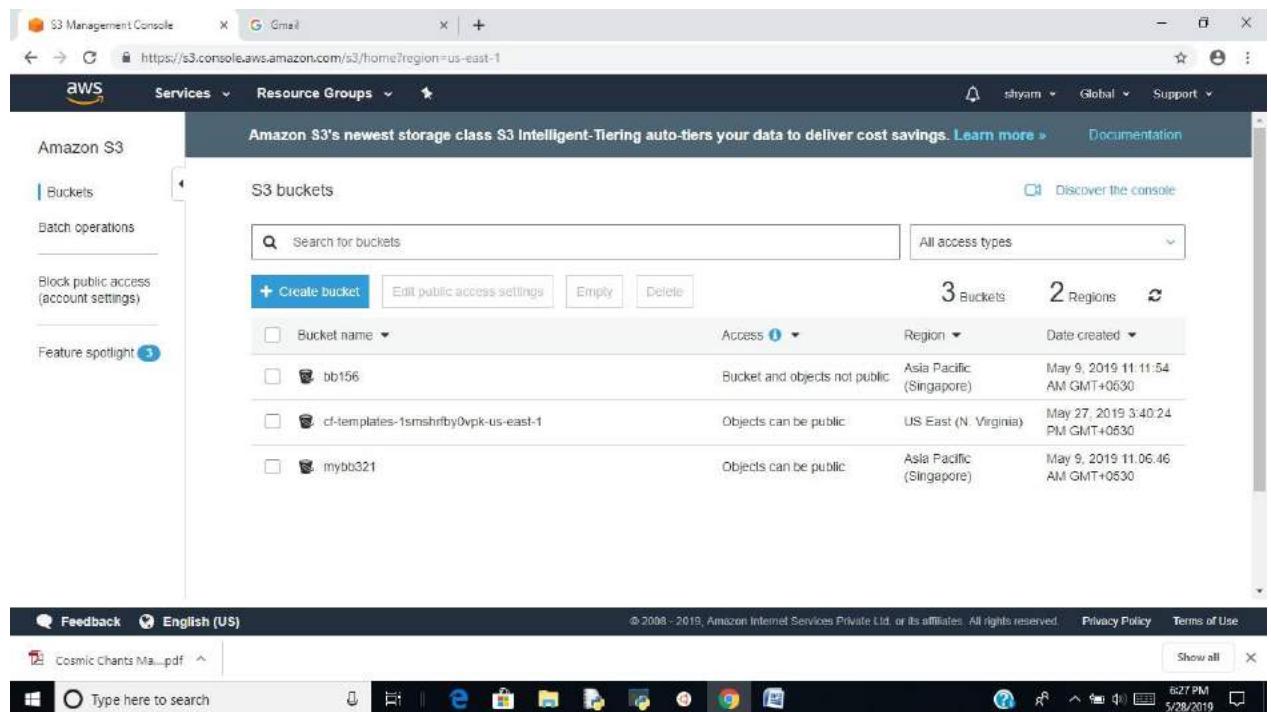


Amazon S3 Console

Amazon S3 provides virtually limitless storage on the internet. This guide explains how you can manage buckets, objects, and folders in Amazon S3 by using the AWS Management Console, a browser-based graphical user interface for interacting with AWS services.

Creatins S3 bucket

- Signin into aws console and choose s3 service and click on create bucket



Name and region

- Bucket name: enter name for bucket this bucket name should be unique across all existing buckets in Amazon S3
- Region: enter region name in which region do you want to create this s3 bucket
- Copy settings from an existing bucket: this is the optional. If do you want to copy settings from another bucket then enter bucket name
- Click on next

Configure options

Properties:

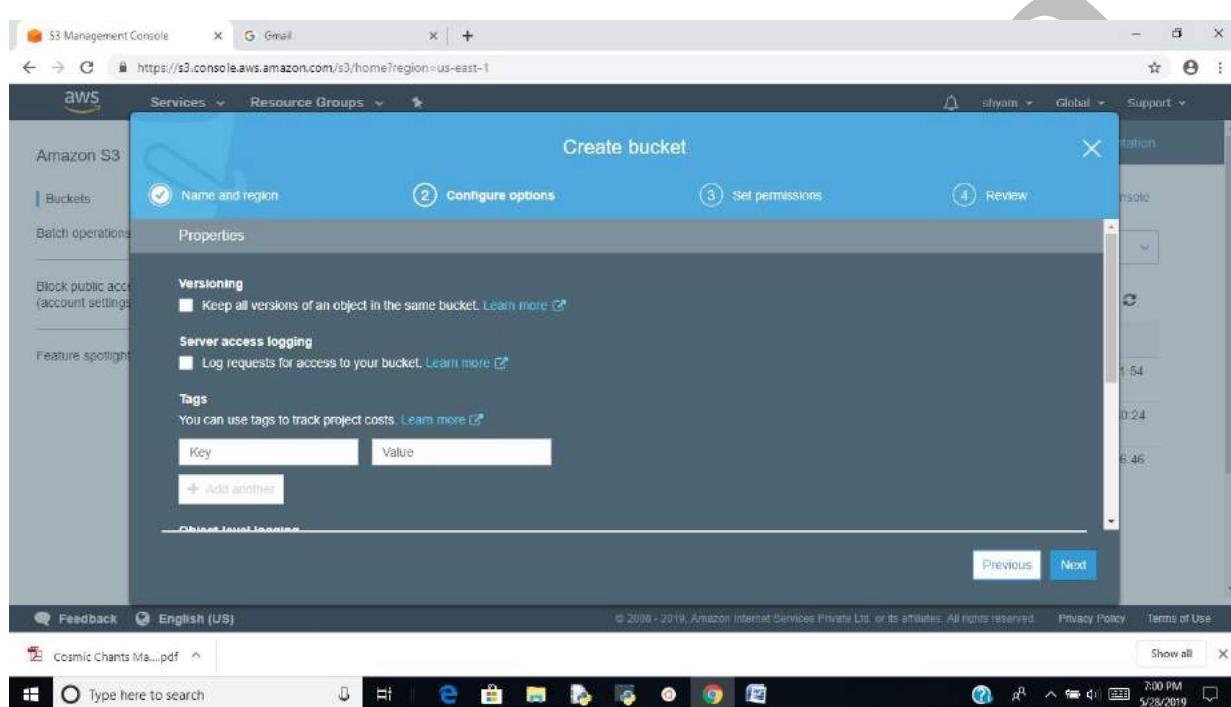
- Versioning: Enable versioning if keep all versions of an object in same bucket
- Server access logging: Enable this option if log requests for access to your bucket
- Tags: you can use tags to track project costs
- Object-level logging: Enable if Record object-level API activity using AWS cloud trail for an additional cost
- Default Encryption: enable this option if do you want encrypt objects stored in s3

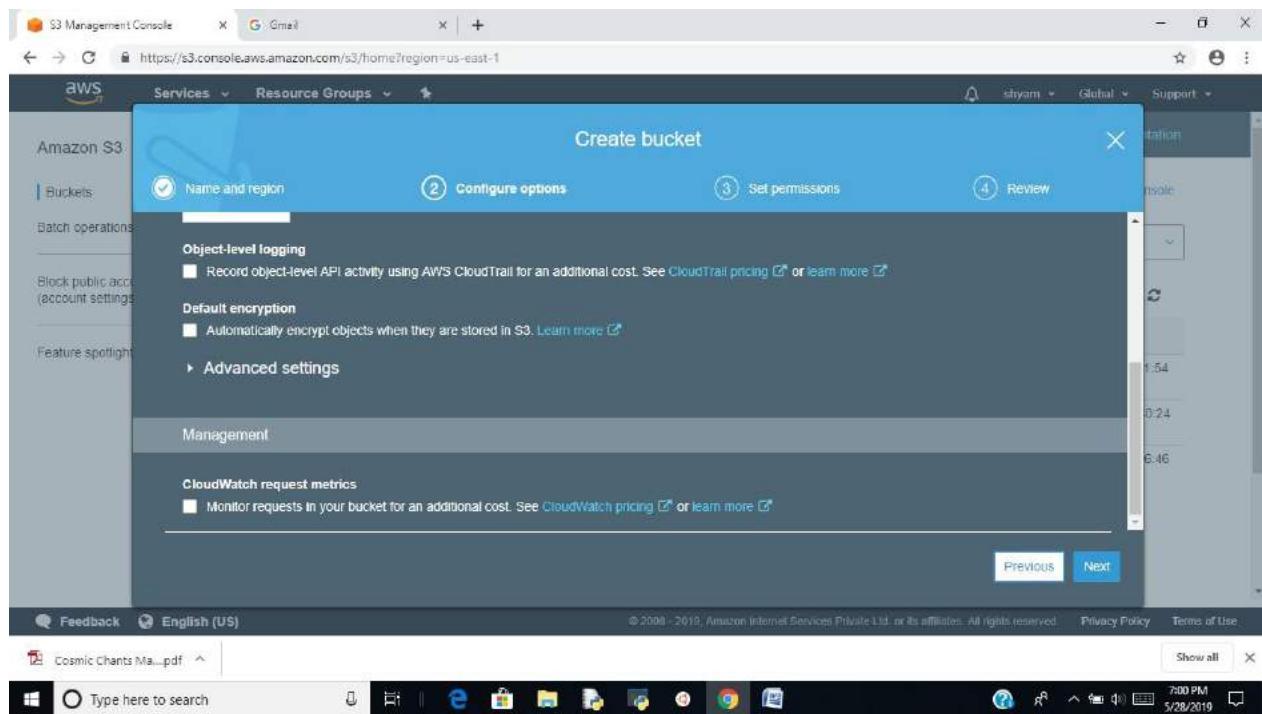
Advanced settings

- Object lock: Enable this option if permanently allow objects in this bucket to be locked

Management

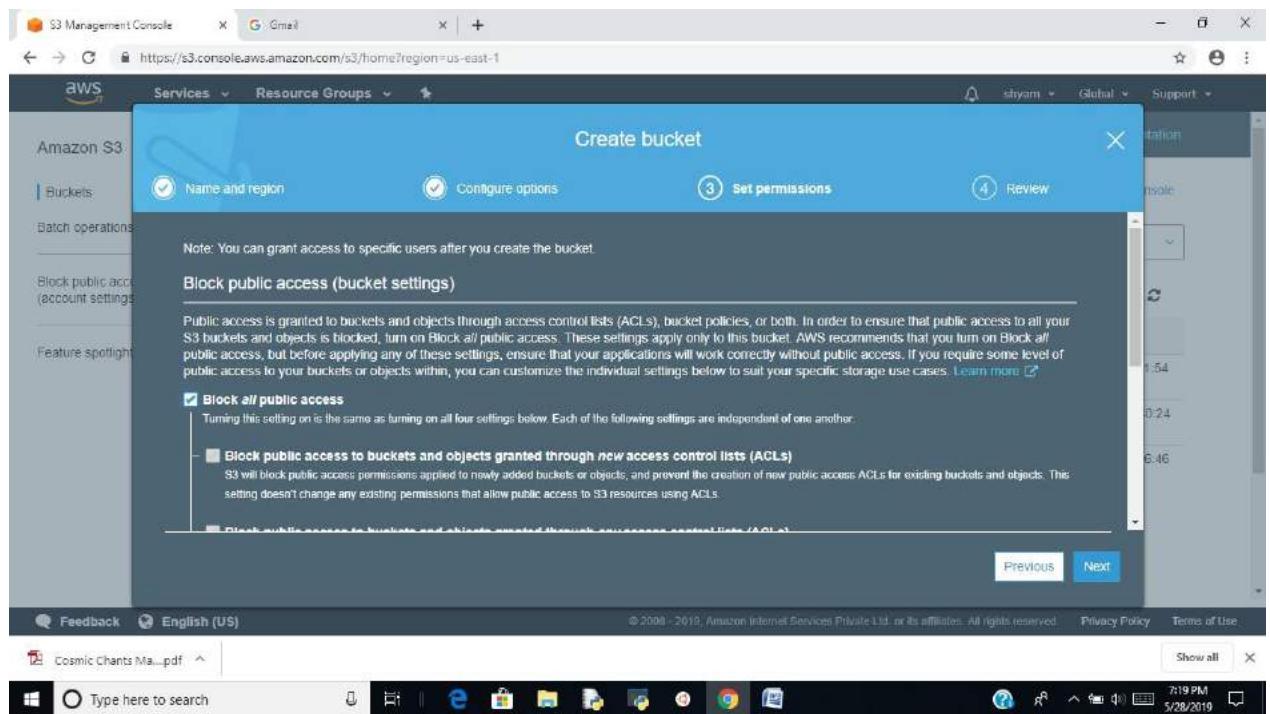
- CloudWatch request metrics: Enable this option if monitor requests in your bucket for additional cost.





Set permissions

- Aws s3 is recommended to Block Public Access
- Disable this Block Public Access for given public access to this bucket
- You can set the permissions to this bucket by using bucket policy and Access control List
- Click on next



Review

- You can check all attributes and options to create these bucket and click on create bucket

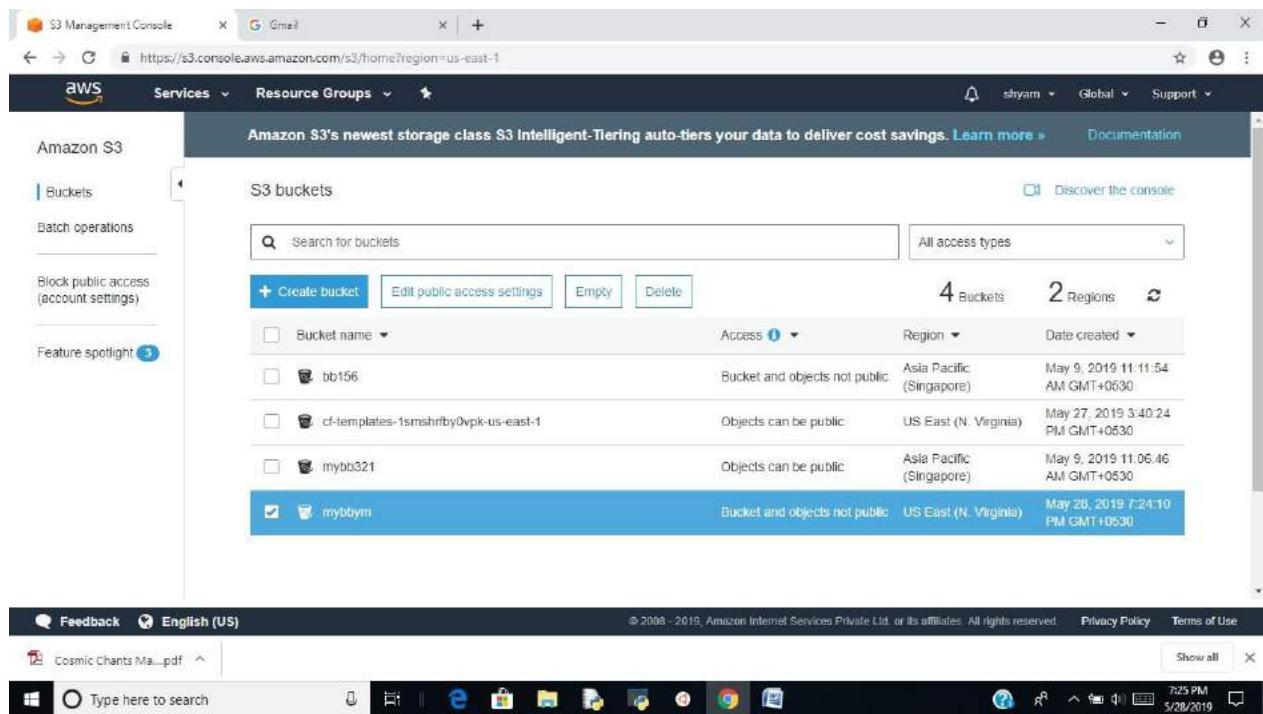
Amazon Web Services

The screenshot shows the AWS S3 Management Console with the 'Create bucket' wizard open. The wizard has three steps: 'Name and region', 'Configure options', and 'Set permissions'. The 'Name and region' step is selected. In this step, the bucket name is set to 'mybbym' and the region is 'US East (N. Virginia)'. The 'Configure options' step is also visible, showing various AWS services like Versioning, Server access logging, Tagging, Object-level logging, Default encryption, CloudWatch request metrics, and Object lock, all currently disabled. The 'Set permissions' step is shown below. At the bottom right of the wizard are 'Previous' and 'Create bucket' buttons.

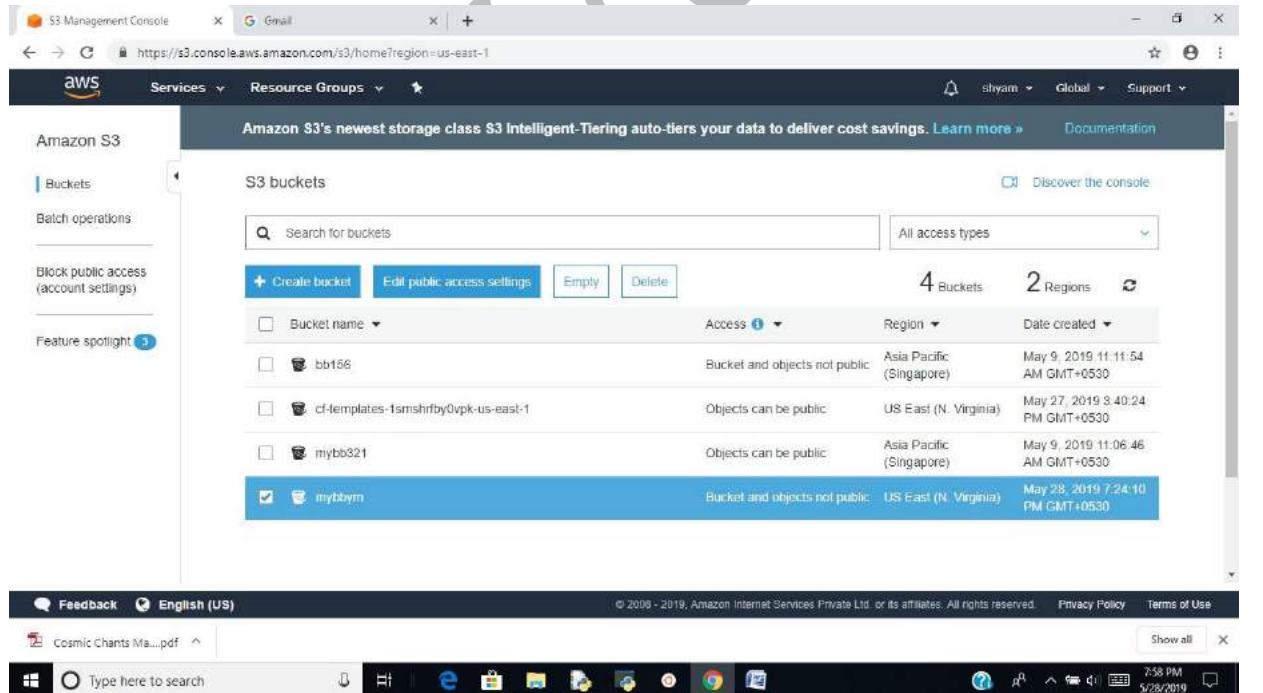
The browser window title is 'S3 Management Console'. The URL is 'https://s3.console.aws.amazon.com/s3/home?region=us-east-1'. The page footer includes links for 'Feedback', 'English (US)', 'Privacy Policy', and 'Terms of Use'. The operating system taskbar at the bottom shows the date as 5/28/2019 and the time as 7:22 PM.

This screenshot shows the 'Set permissions' step of the 'Create bucket' wizard. It lists several permission options under 'Block all public access': 'Block public access to buckets and objects granted through new access control lists (ACLs)', 'Block public access to buckets and objects granted through any access control lists (ACLs)', 'Block public access to buckets and objects granted through new public bucket policies', and 'Block public and cross-account access to buckets and objects through any public bucket policies'. All these options are currently set to 'On'. The other two steps of the wizard ('Name and region' and 'Configure options') are also visible above this step. The bottom right of the wizard has 'Previous' and 'Create bucket' buttons.

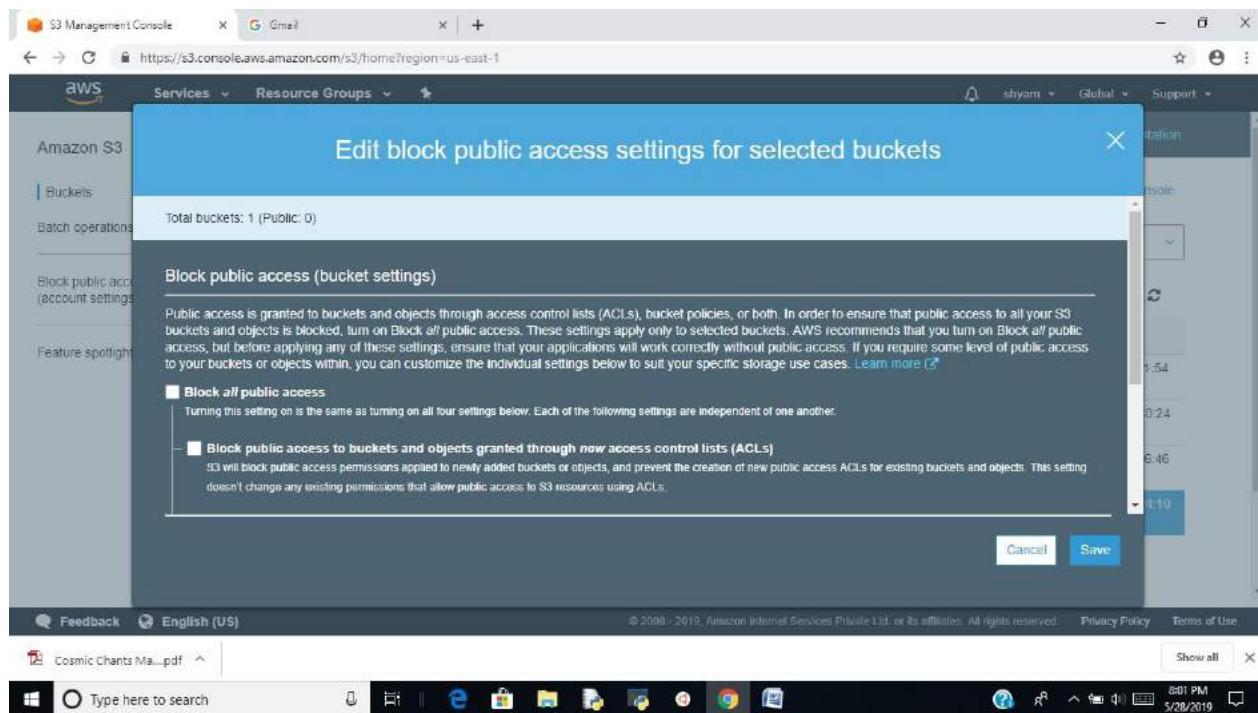
The browser window title is 'S3 Management Console'. The URL is 'https://s3.console.aws.amazon.com/s3/home?region=us-east-1'. The page footer includes links for 'Feedback', 'English (US)', 'Privacy Policy', and 'Terms of Use'. The operating system taskbar at the bottom shows the date as 5/28/2019 and the time as 7:23 PM.



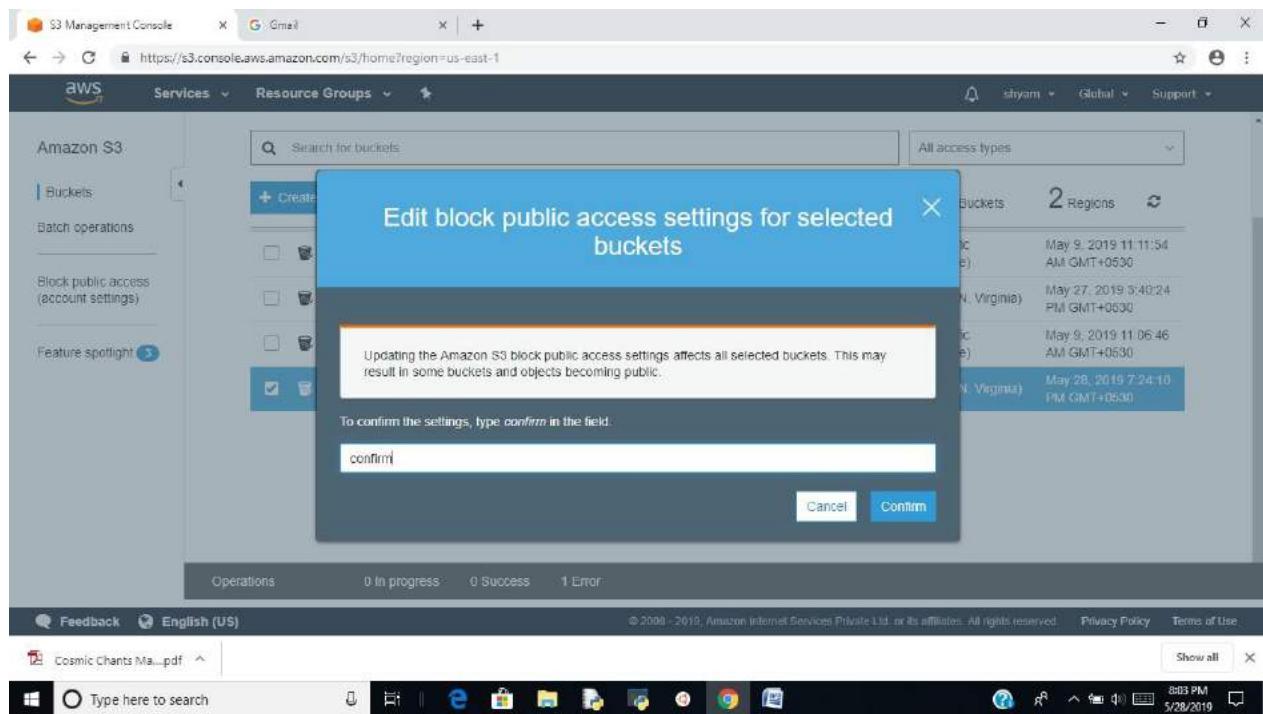
- Do you want to edit the public access settings of existing bucket then first select the bucket and after click on edit public access settings



- Enable required permissions and click on save

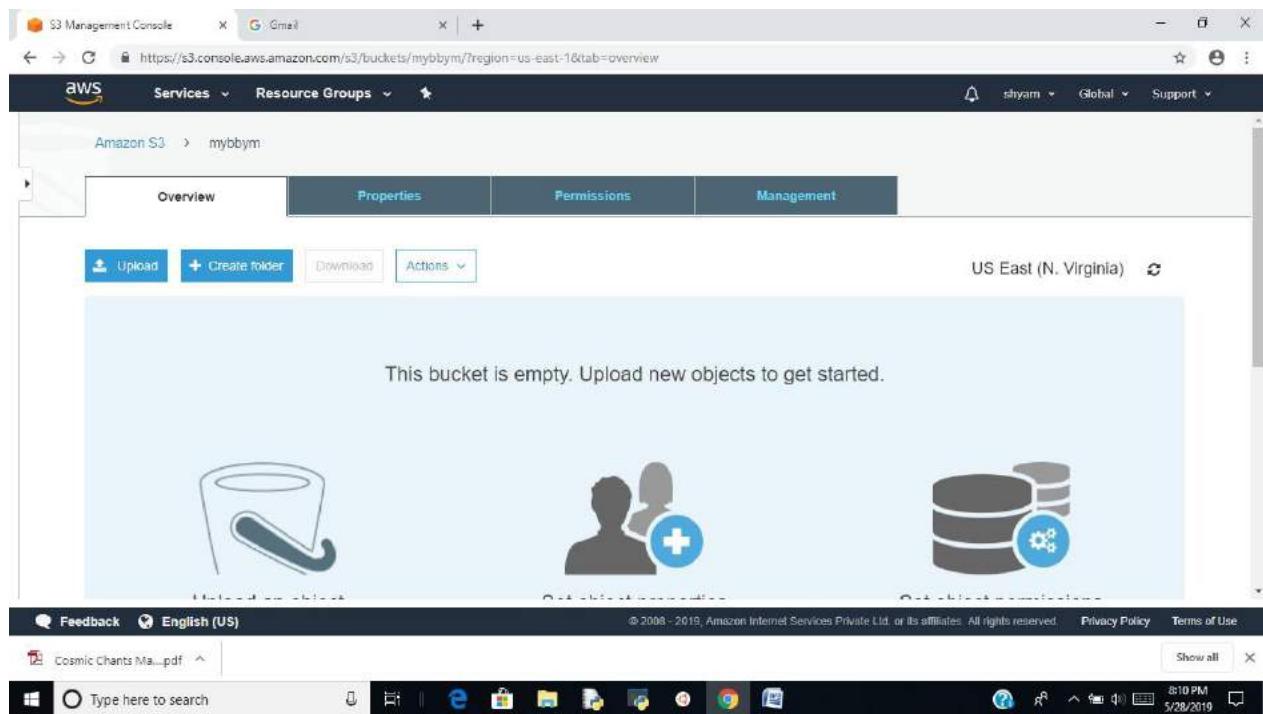


- To confirm the settings type confirm in the field and click on confirm



Create folder in bucket

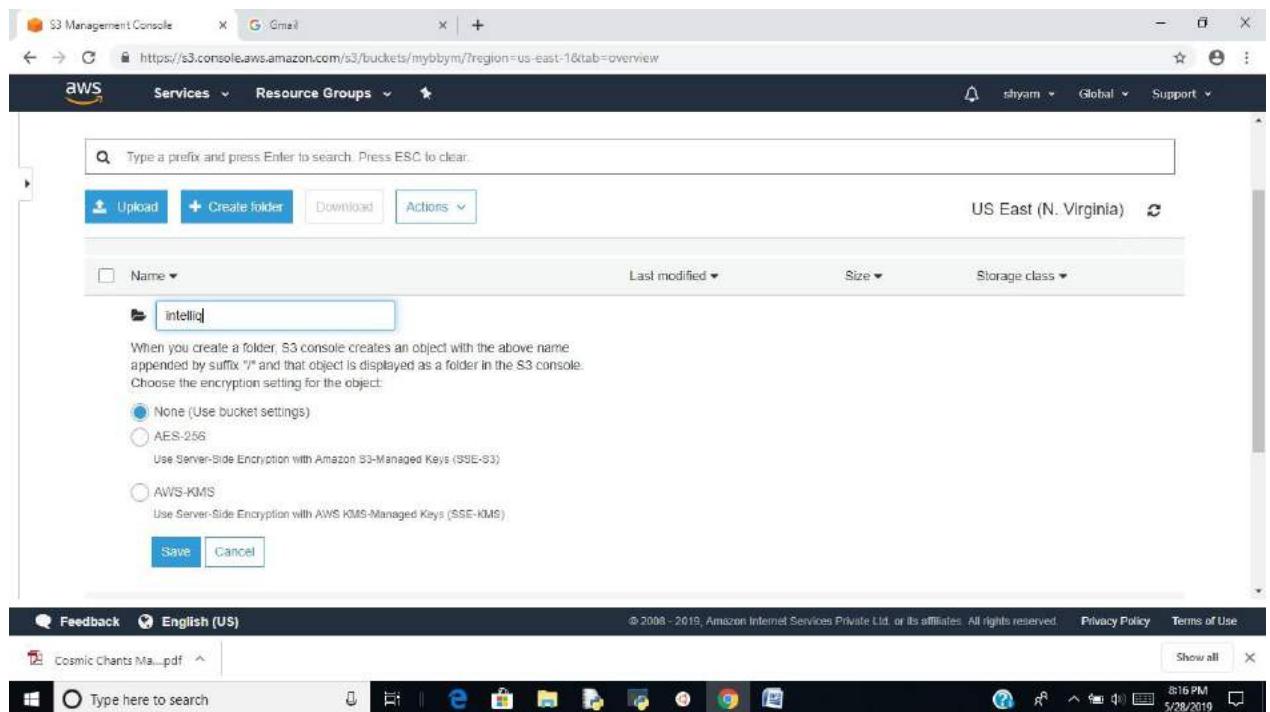
- Click on bucket name and click on create folder



- Enter folder name
- Encryption Settings: select None (Use bucket settings)

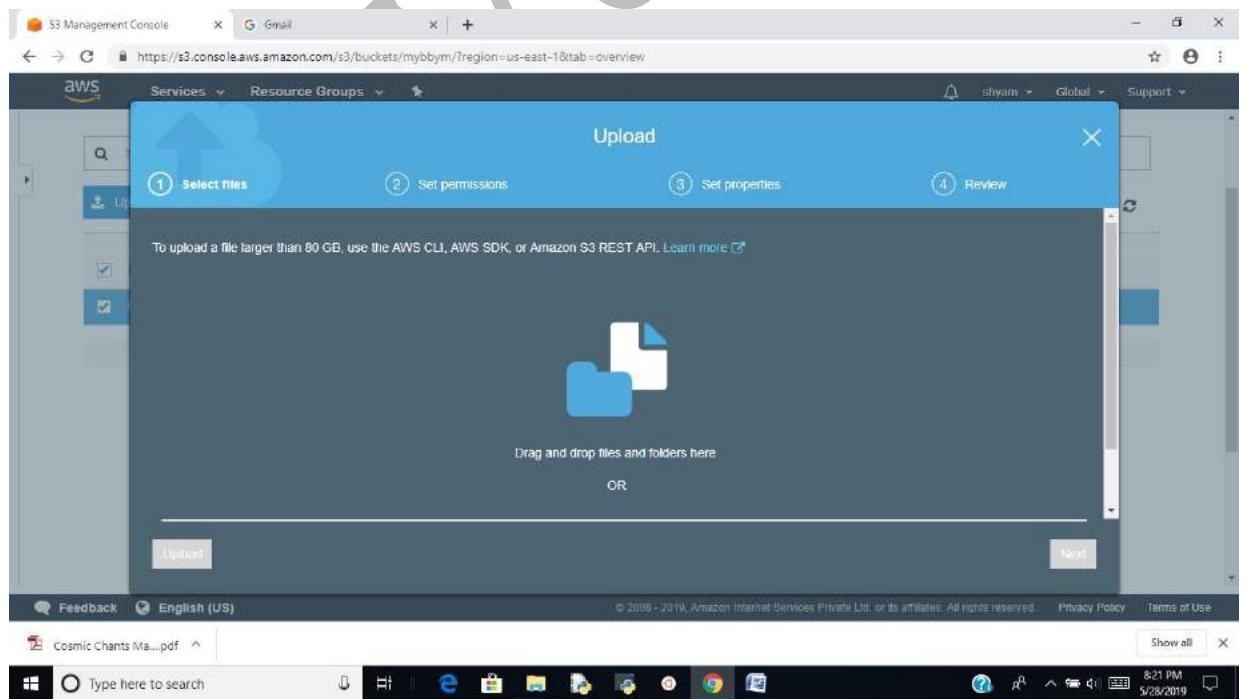
Note: do you want to apply AES-256 or AWS-KMS you can select particular option

- Click on save

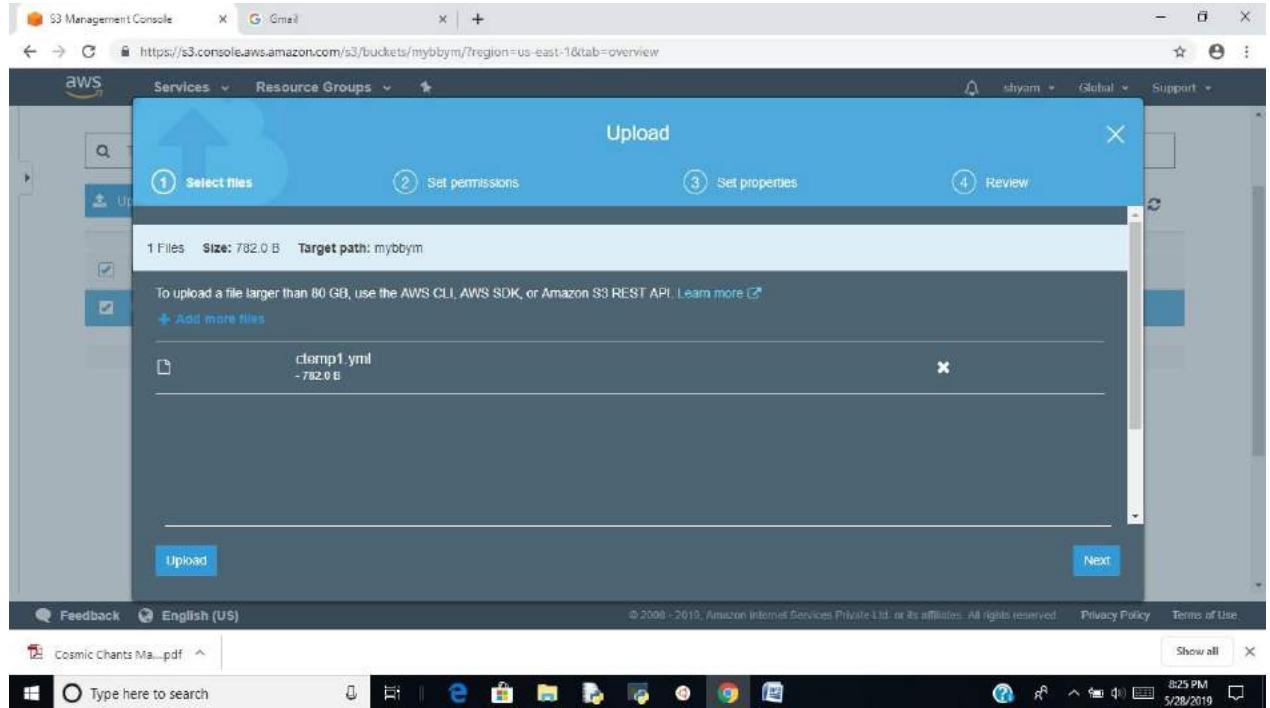


Upload Objects

- Click on upload and select object or file to upload from your local host

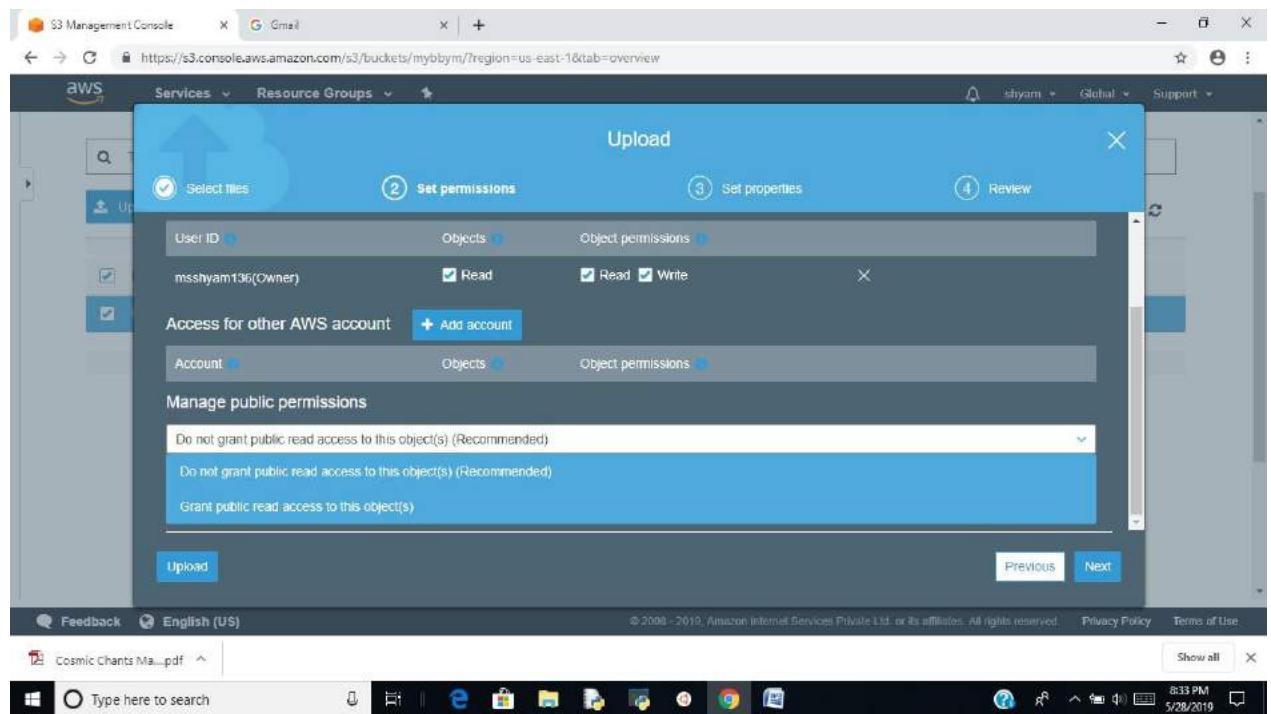


- Select files: drag and drop your uploaded files and click on next



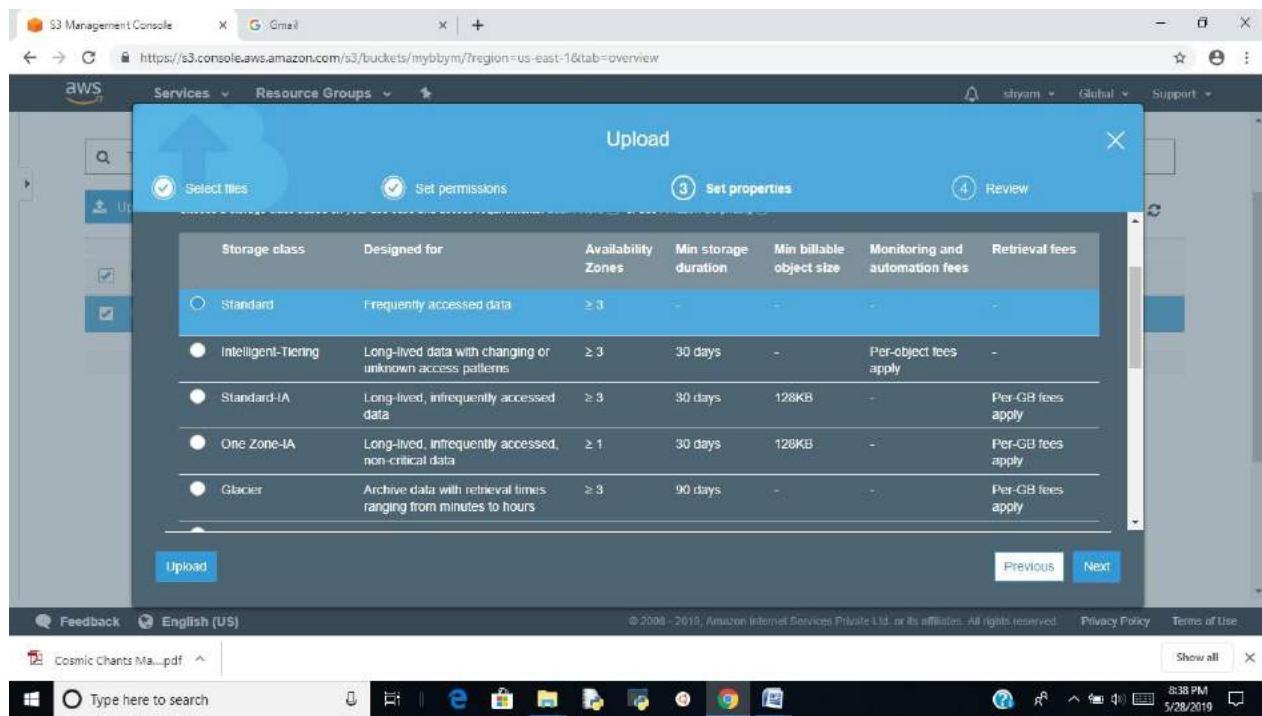
Set permissions:

- Manage Users: Here you can enable or disable read and write permissions for user
- Access for another AWS account: you can add another AWS account by click on Add account
- Manage Public Permissions: Here you can add public read access to this object or not add
- Click on next



Set properties

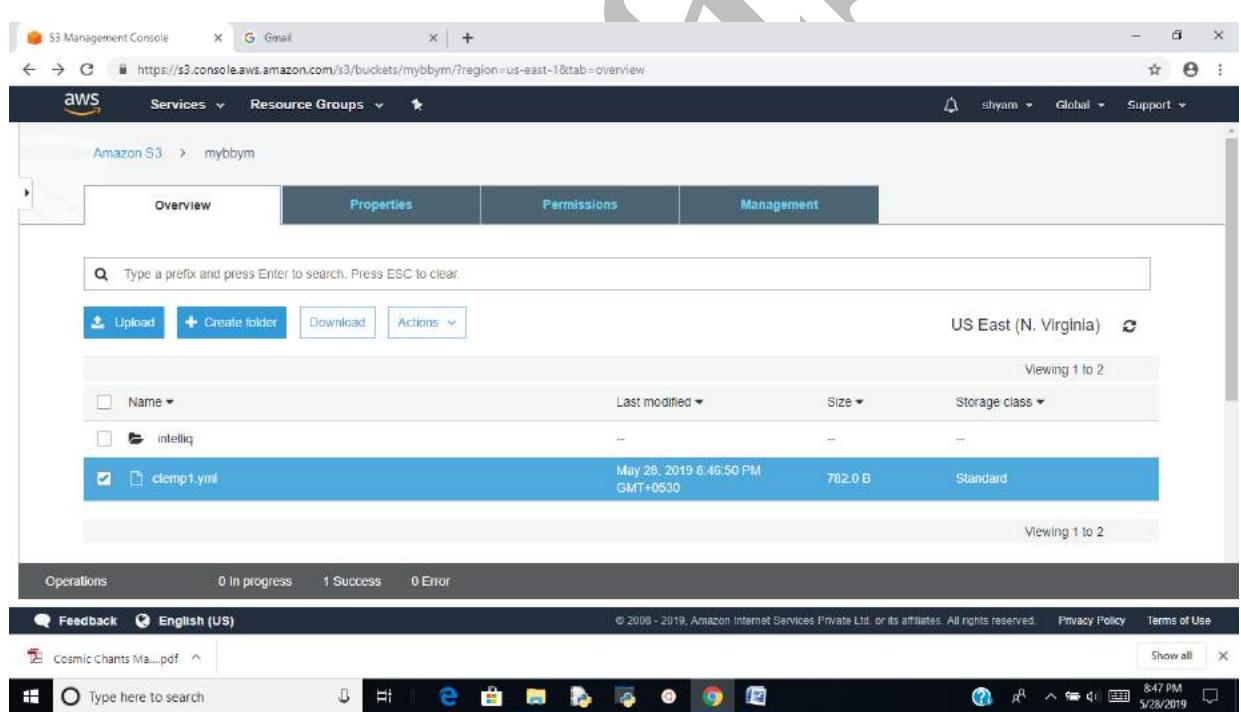
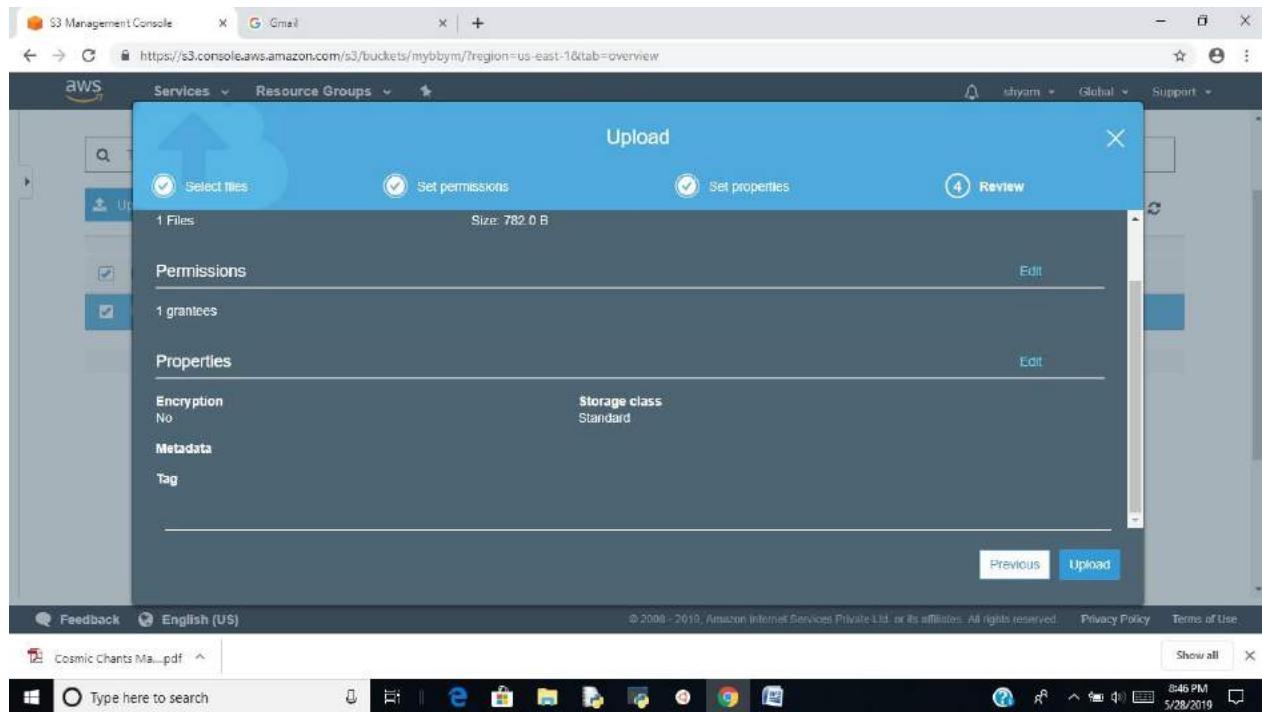
- Here we select storage class for this object based on accessibility and object type you can choose storage class
- Select the Standard storage class and click on next



Review

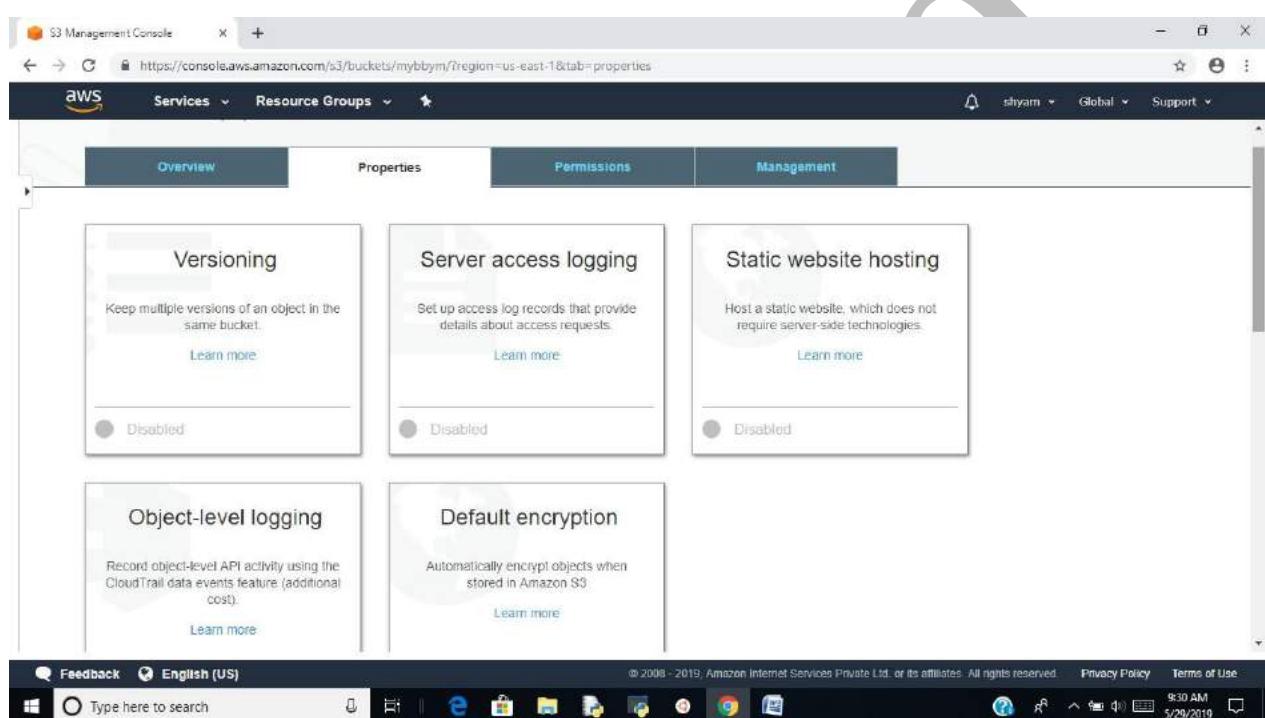
- Check the all object properties and do you want make any changes then click on edit option and save changes
- Click on upload

Amazon Web Services



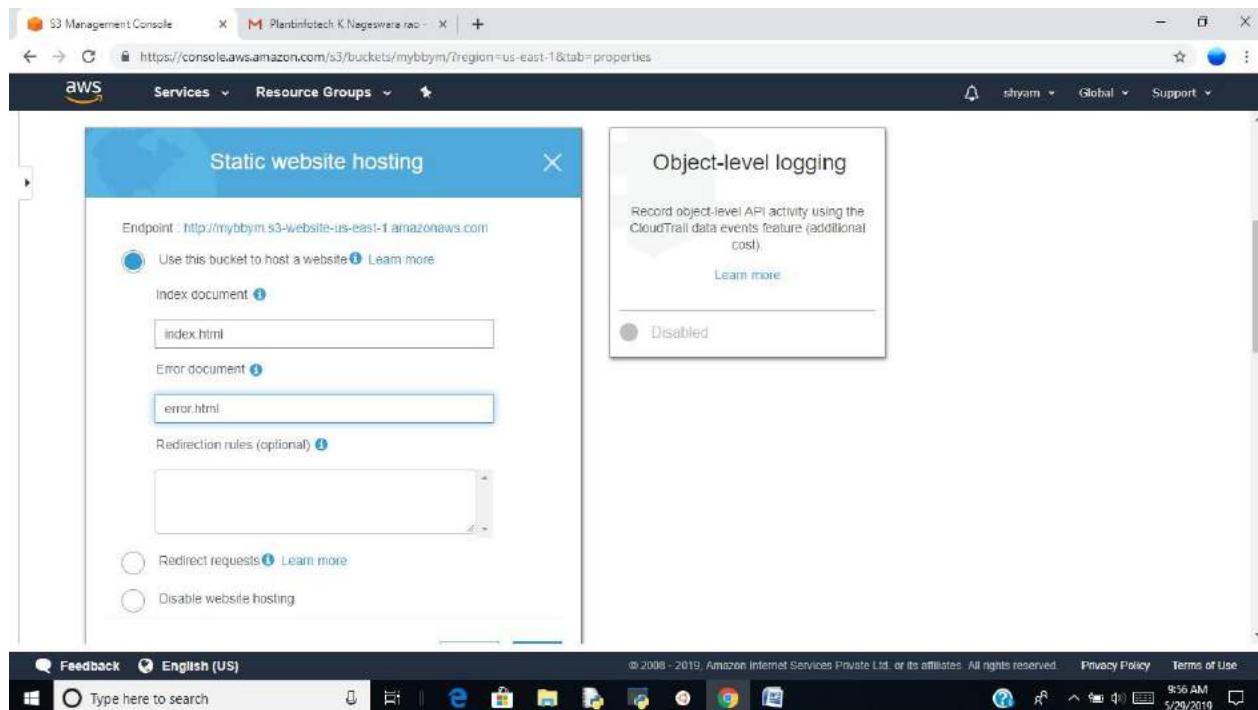
Changing bucket properties

- Click on bucketname then we see the properties section then click on properties
- There are several bucket properties that is Versioning, server access logging, static website hosting, object-level logging and default encryption all properties disable by default



- To change the properties of bucket then click on particular property and enable this property
- Enable Static website hosting : this option is used hosting you applicatin from this bucket
- There two options one is use this bucket to host a website and another is redirect requests to another host
- Here click on first option
- Index Document: enter file name which file do you want make as index starting page (ex: index.html)
- Error Document: enter file name for error redirection

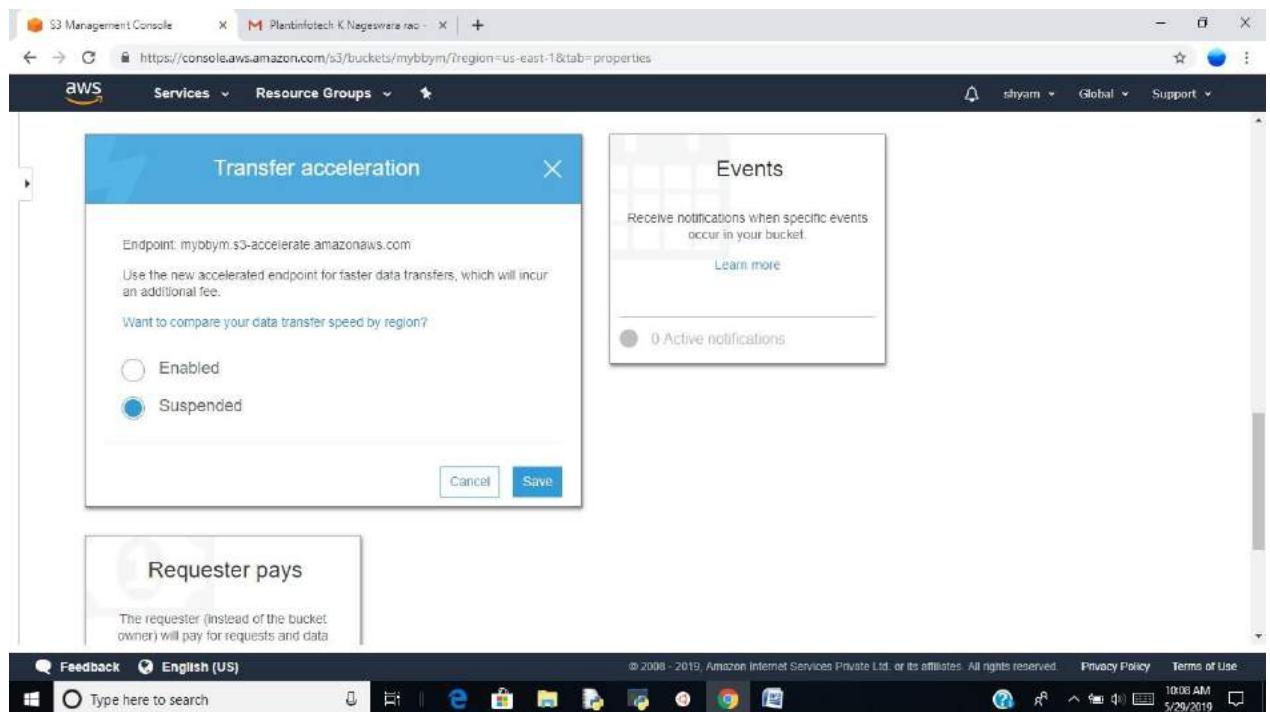
- Redirection rules: redirection rules is the optional



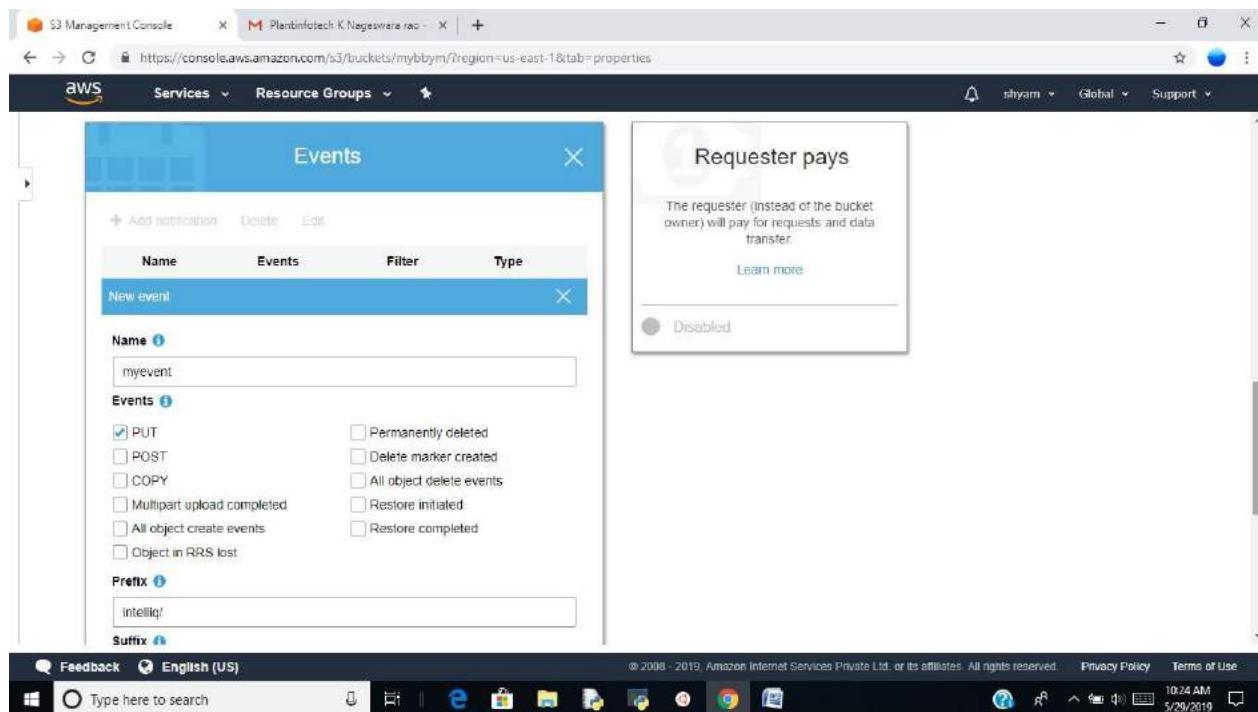
- Enable Versioning: click on enable versioning do you want to make version controlling

Advanced Settings

- Object lock: Enable this option to prevent objects from being deleted
- Transfer acceleration: Enable fast, easy and secure transfer of files to and from your bucket.

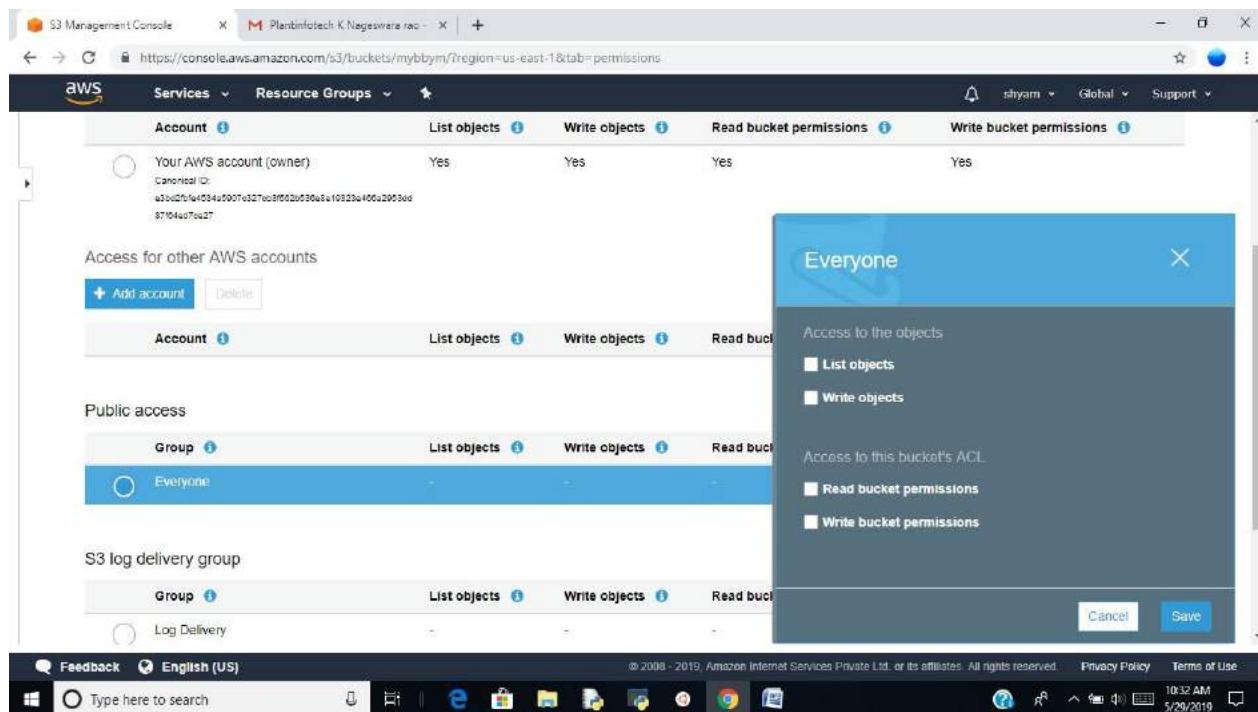


- Events: Receive notifications when specific events occur in your bucket
- Click on add notification
- Name: name for event
- Event: for which action do you want to make notification that is PUT, POST, COPY etc
- Prefix: it is the optional, notifications to object with keys starting with matching characters (ex: dyasa...: make the notifications for objects whose object name starts with dyasa)
- Suffix: with the keys ending with matching characters
- Send To: select notification destination that is SNS Topic, SQS Queue etc
- If select SNS Topic then enter Topic name
- Click on save



Permissions

- Block public access: click on Block public access to Block public access
- Access Control List: access control list (ACLs) are used to grant basic read/write permissions to another aws account
- Click on Add another AWS account and enter account id
- Public access: click on every one and add permissions and save



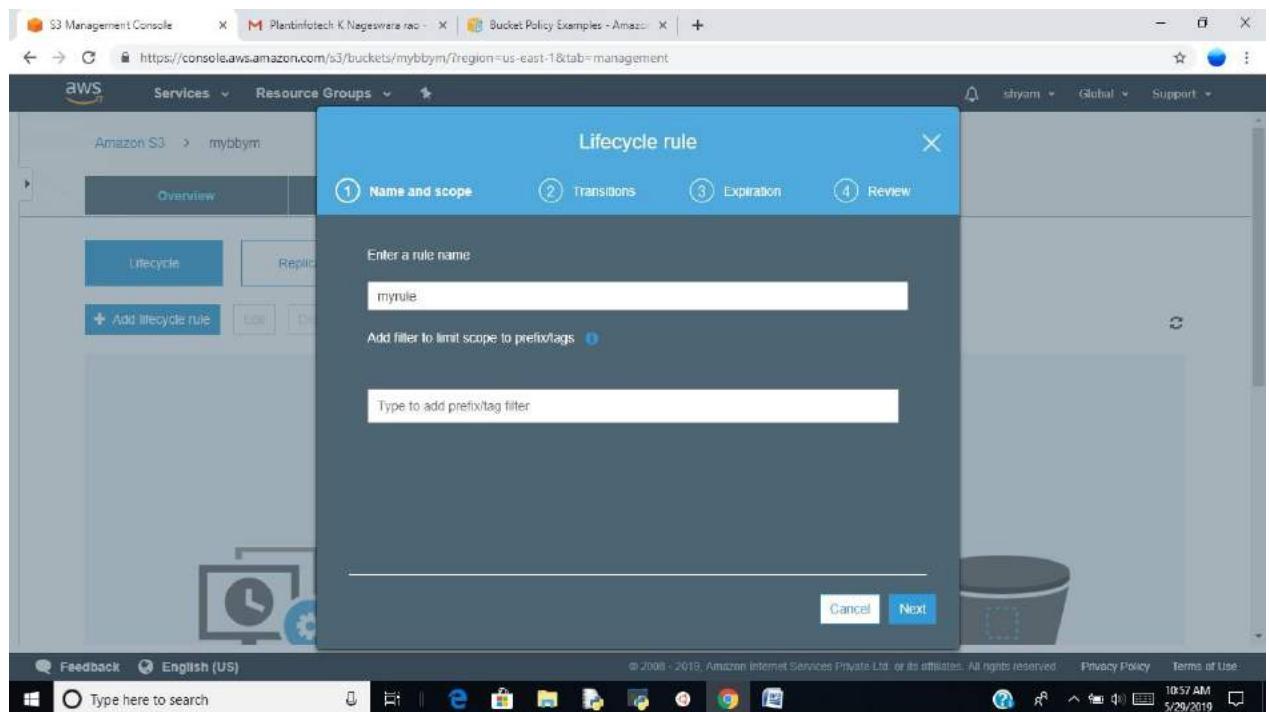
- S3 Log Delivery Group: add the group to deliver the s3 log files to group

Bucket Policy: bucket policy is used to manage advanced permissions on Amazon S3 Resources

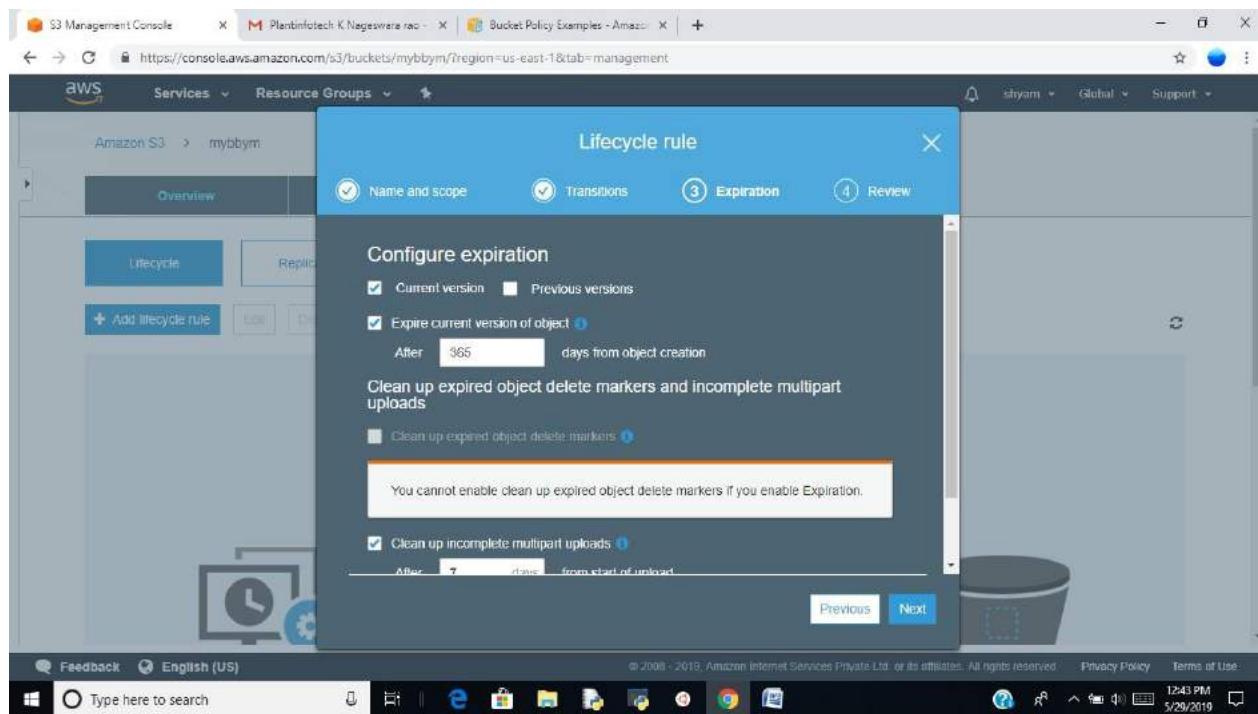
CORS Configuration: Cross-origin resource sharing (CORS) defines way for client web applications that are loaded in one domain that interact with resources in different domain

Management

- Lifecycle: configure the lifecycle for this bucket by click on add lifecycle rule
- Enter a rule name and click on next



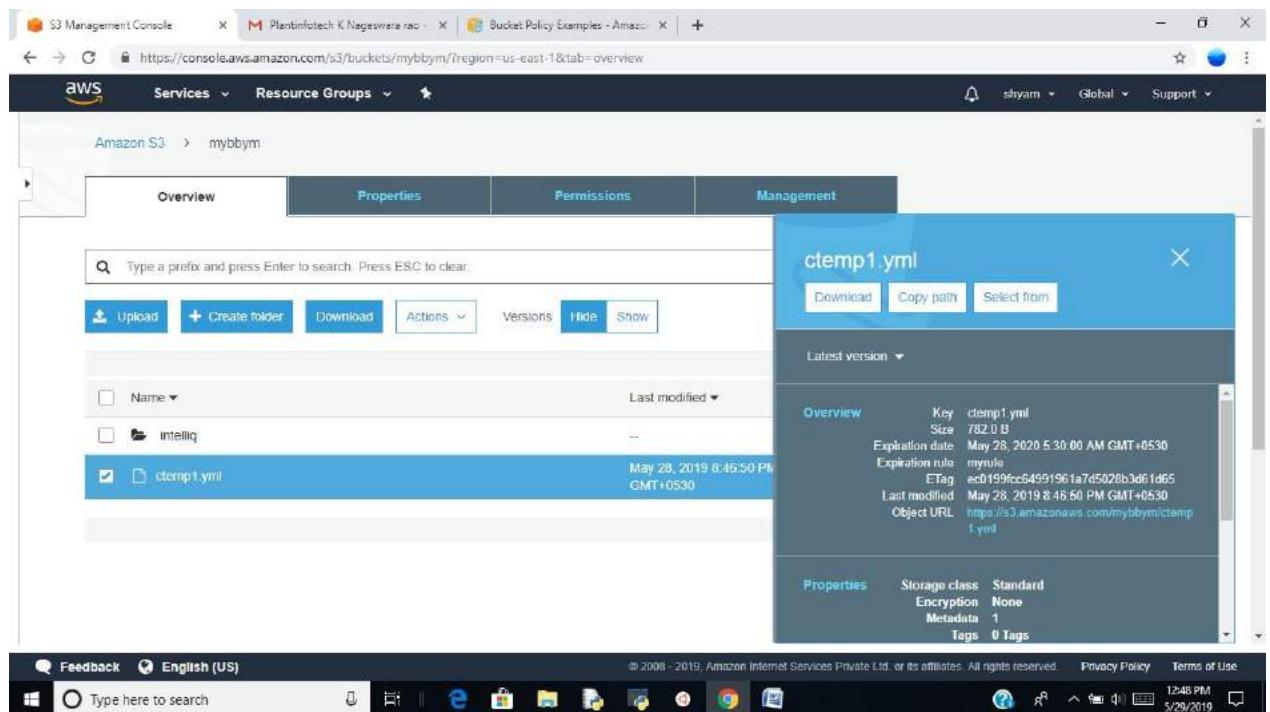
- Storage class transition : you can add rules in a lifecycle configuration to tell Amazon S3 to transition objects to another storage class
- Click on Current version and click on next
- Configure expiration:
- Enable current version
- Expire current version of object and enter number of days for expiration
- Click on next



- See the review and click save

Download objects from S3 bucket

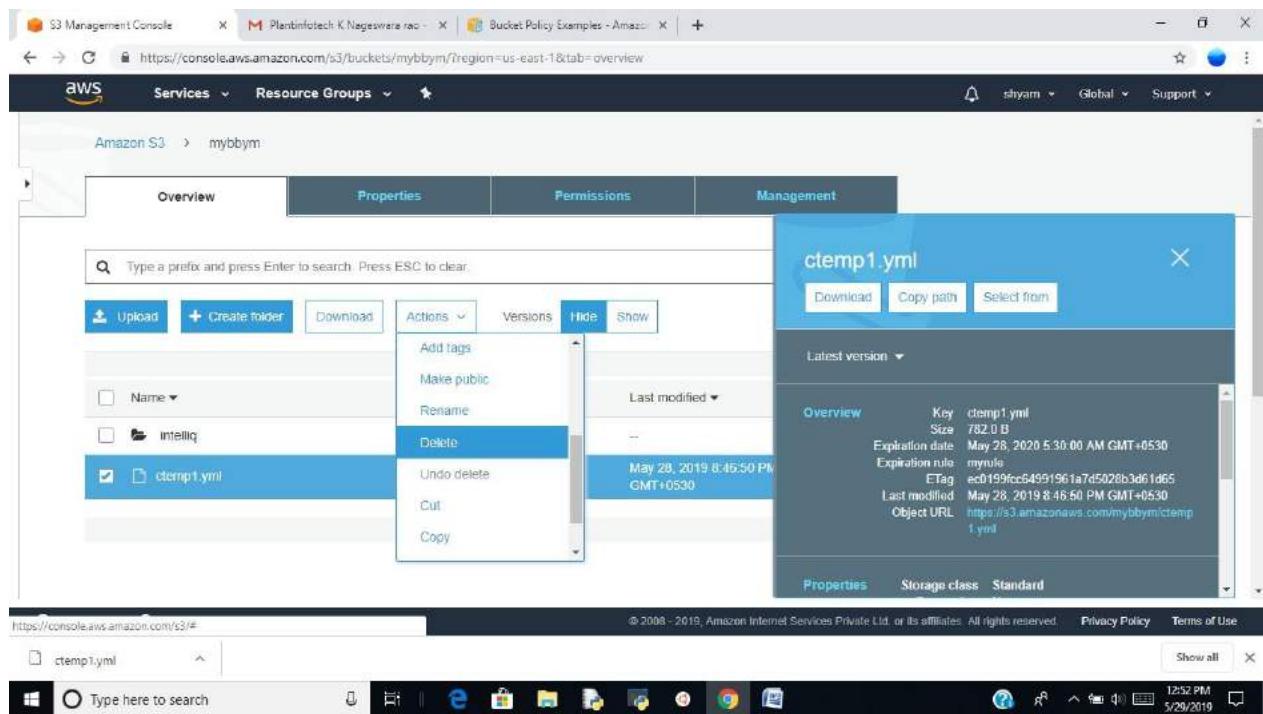
- Open the bucket in s3 and select object which object do you want download
- Click on download option



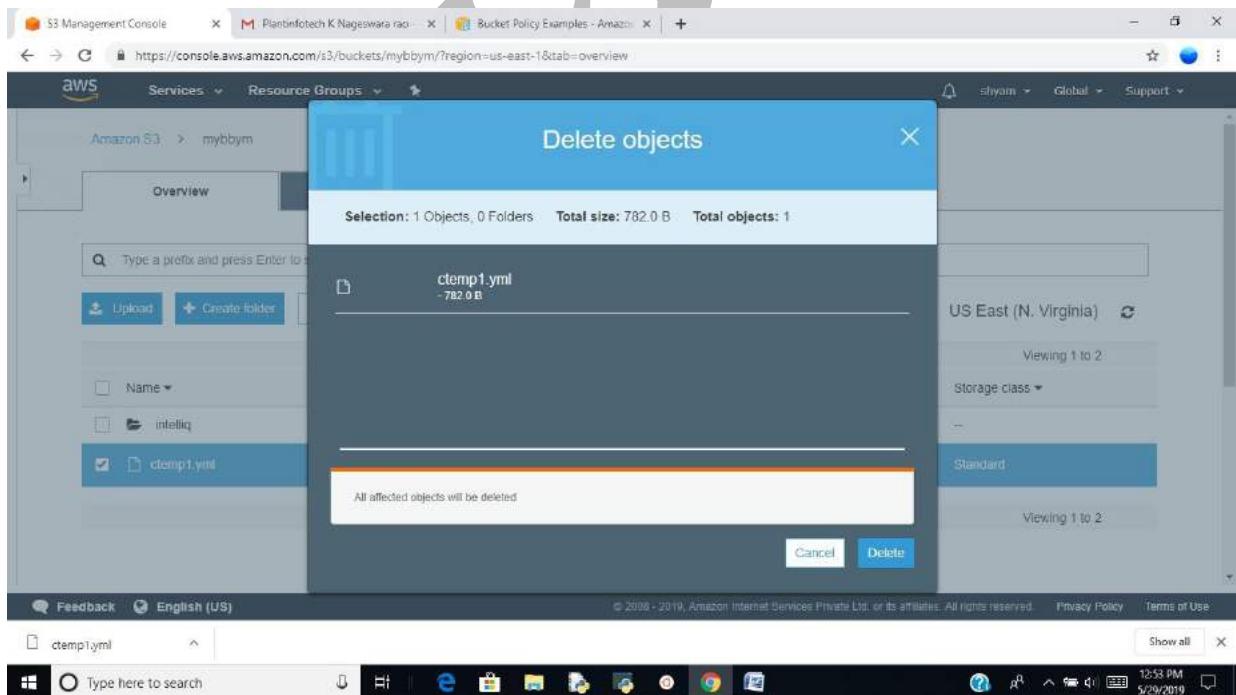
- You can change the version status that is Hide object version or Show object version

Delete the object

- Open the bucket and select object
- Goto actions and select delete objecvt option

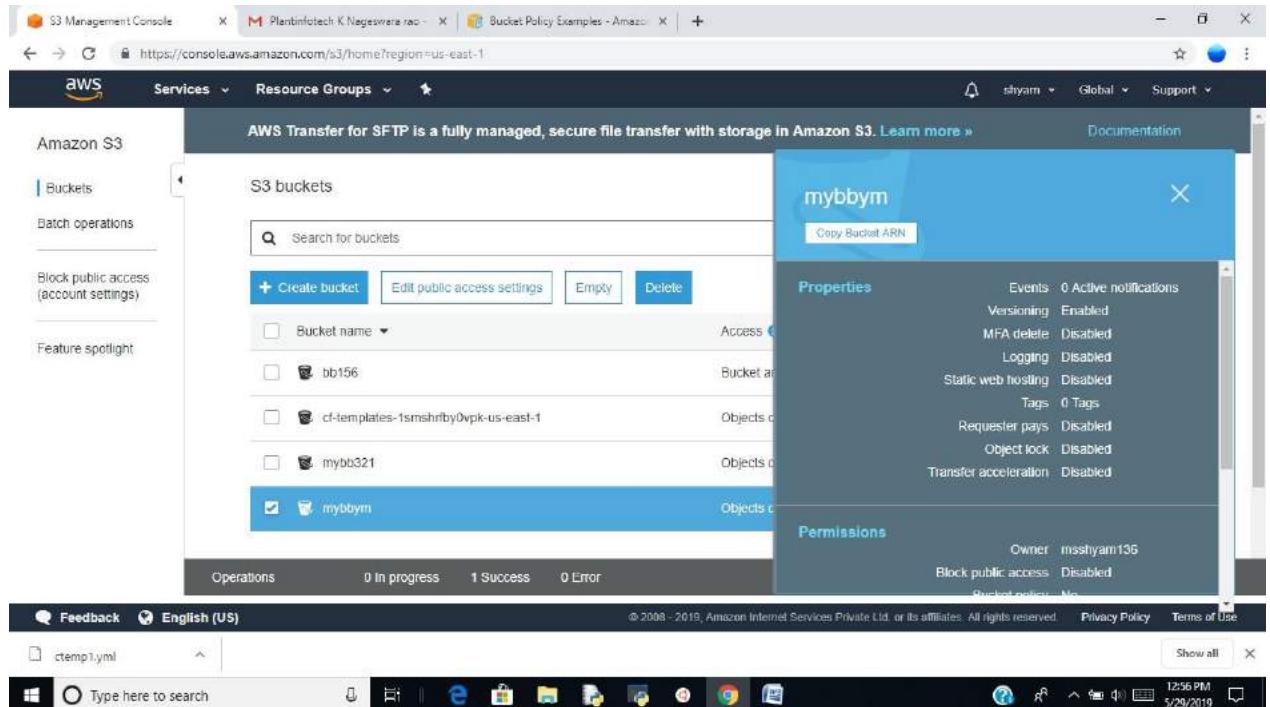


- Click on Delete

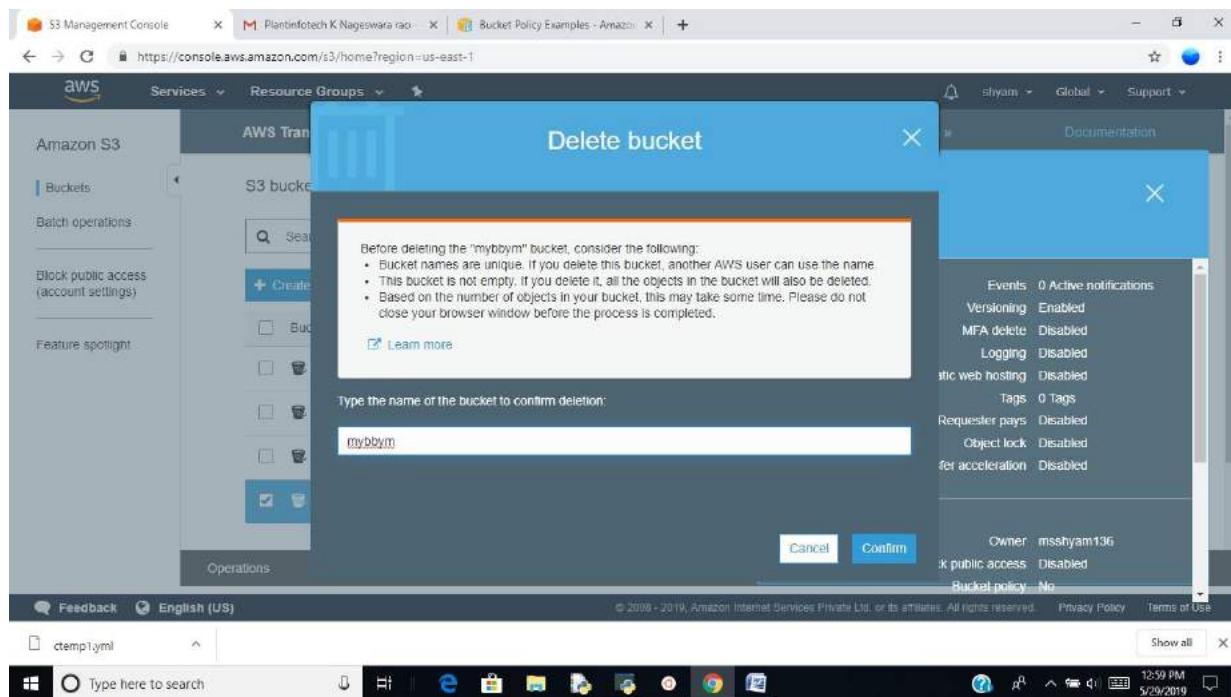


Delete Bucket

- Signin into AWS Console and select S3 service
- Select bucket which you want to delete and click on delete option



- Type the name of bucket to confirm deletion and click on confirm



Identity and Access management (IAM)

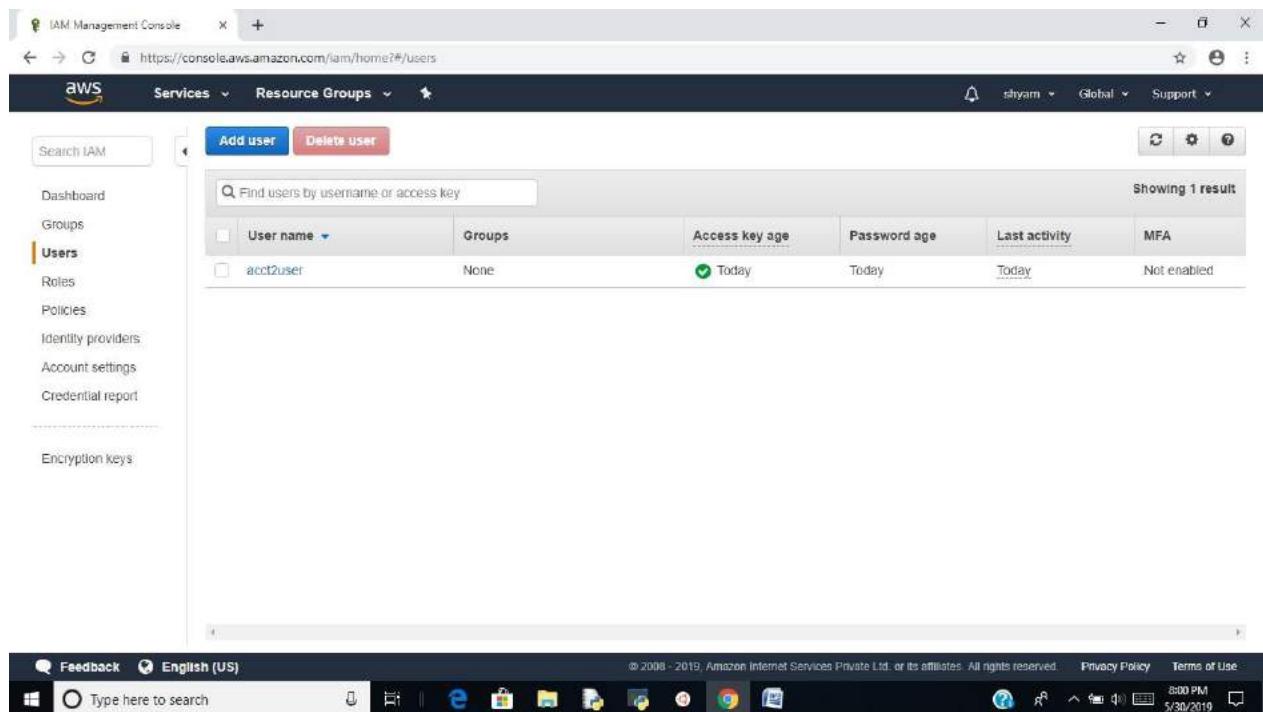
AWS Identity and Access Management (IAM) is a web service that helps you securely control access to AWS resources. You use IAM to control who is authenticated (signed in) and authorized (has permissions) to use resources.

When you first create an AWS account, you begin with a single sign-in identity that has complete access to all AWS services and resources in the account. This identity is called the AWS account *root user* and is accessed by signing in with the email address and password that you used to create the account. We strongly recommend that you **do not use the root user for your everyday tasks**, even the administrative ones. Instead, adhere to the best practice of using the root user only to create your first IAM user. Then securely lock away the root user credentials and use them to perform only a few account and service management tasks.

Users

Create users

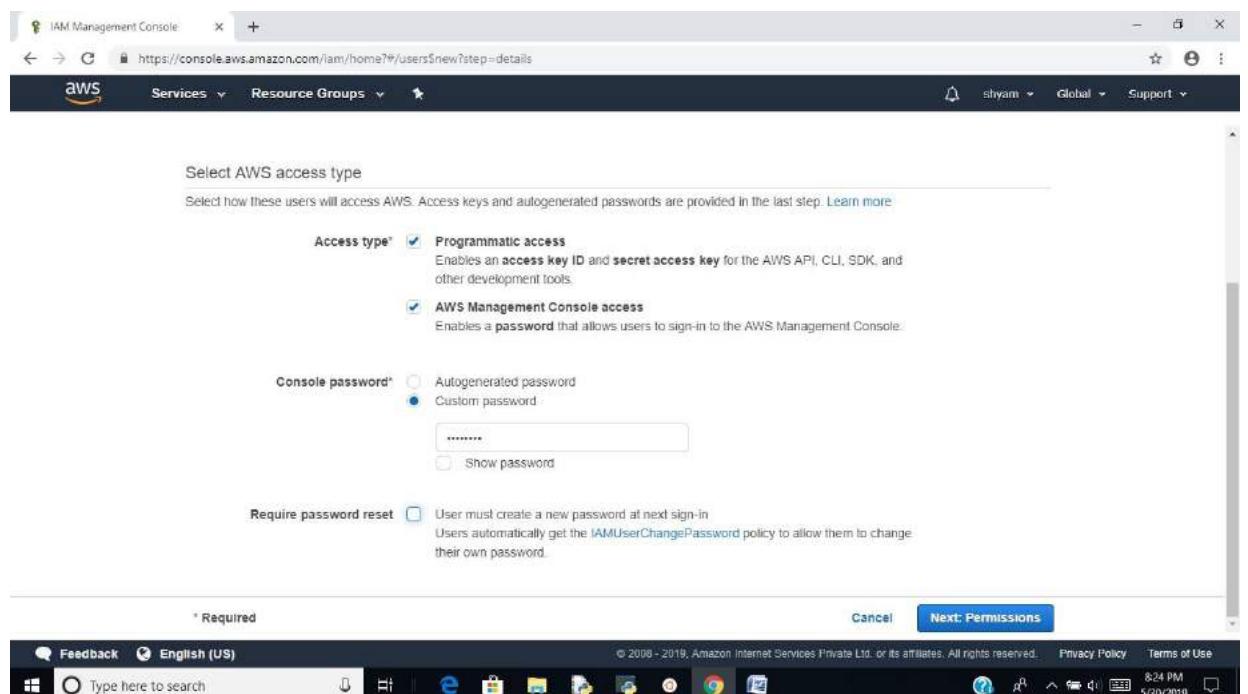
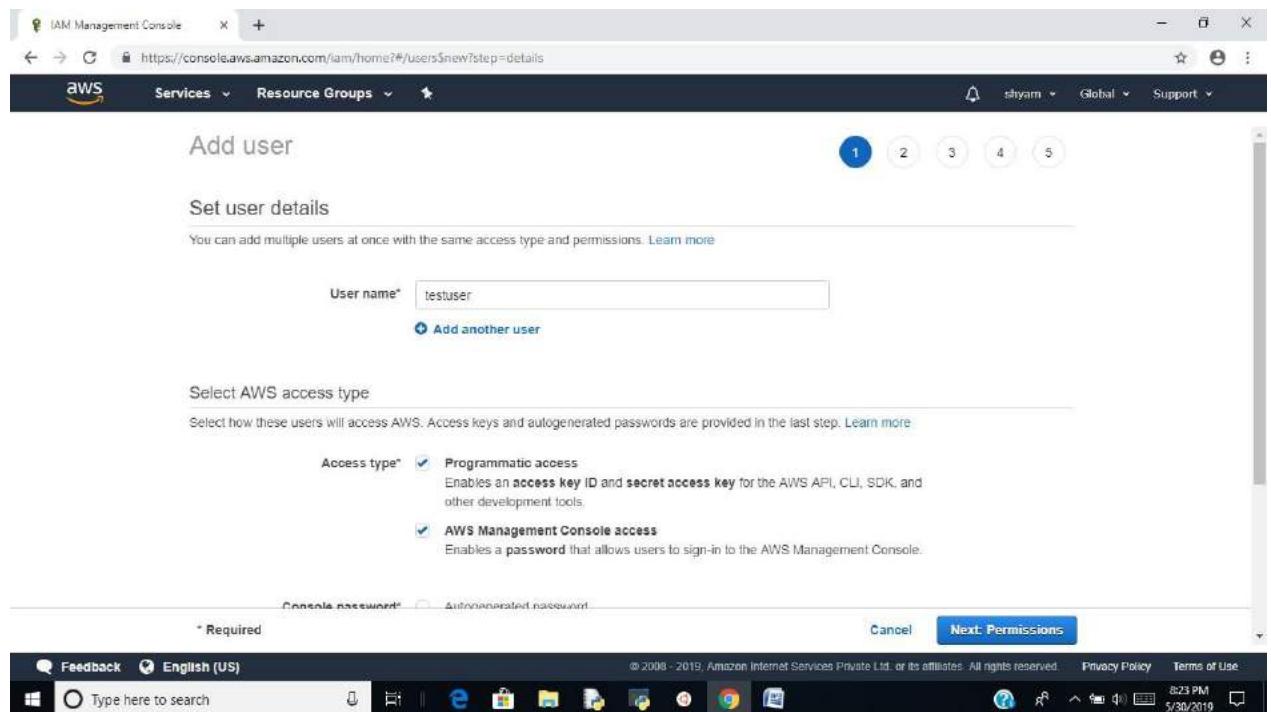
- Signin into the aws console and select IAM service
- Click on users on leftside panel and click on Add user



The screenshot shows the AWS IAM Management Console. The left sidebar has a 'Users' section selected. The main area displays a table with one user entry:

User name	Groups	Access key age	Password age	Last activity	MFA
acct2user	None	Today	Today	Today	Not enabled

- Username: enter username
- You can add multiple users by click on add another user
- Select AWS access type: There are two types one is programmatic access it is used for accessing through aws cli, aws API, aws SDK using accesskey and secret accesskey. Another one is AWS Management Console access it is used for setting user custom password and it's provide user authentication through user console
- Consloe password: there are two options autogenerated password and custom password. Do you want to make your own custom password use custom password
- Require password reset: enable this option if do you want to reset the password for every signin
- Click on next permissions



Set permissions: there are three types

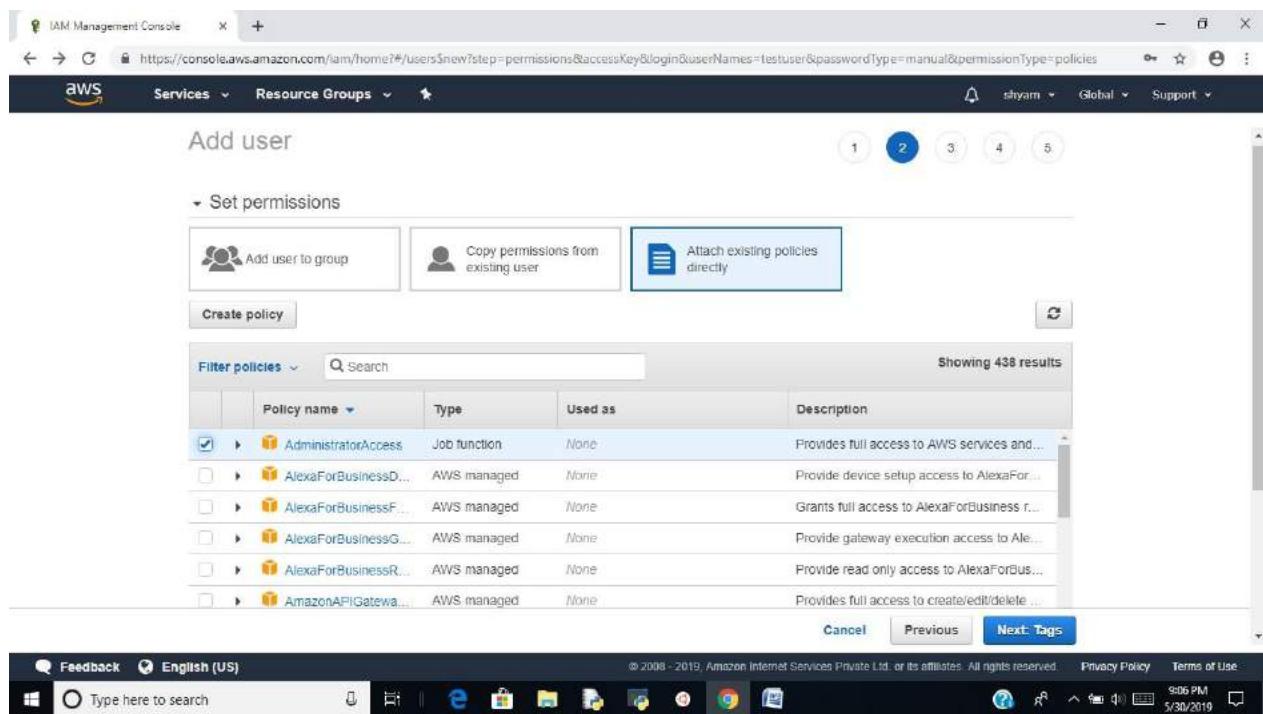
Add user to group: add this user to existing group then this user have all permissions of that group

Copy permissions from existing user: you copy the all permissions of existing user by selecting particular user from the list of users

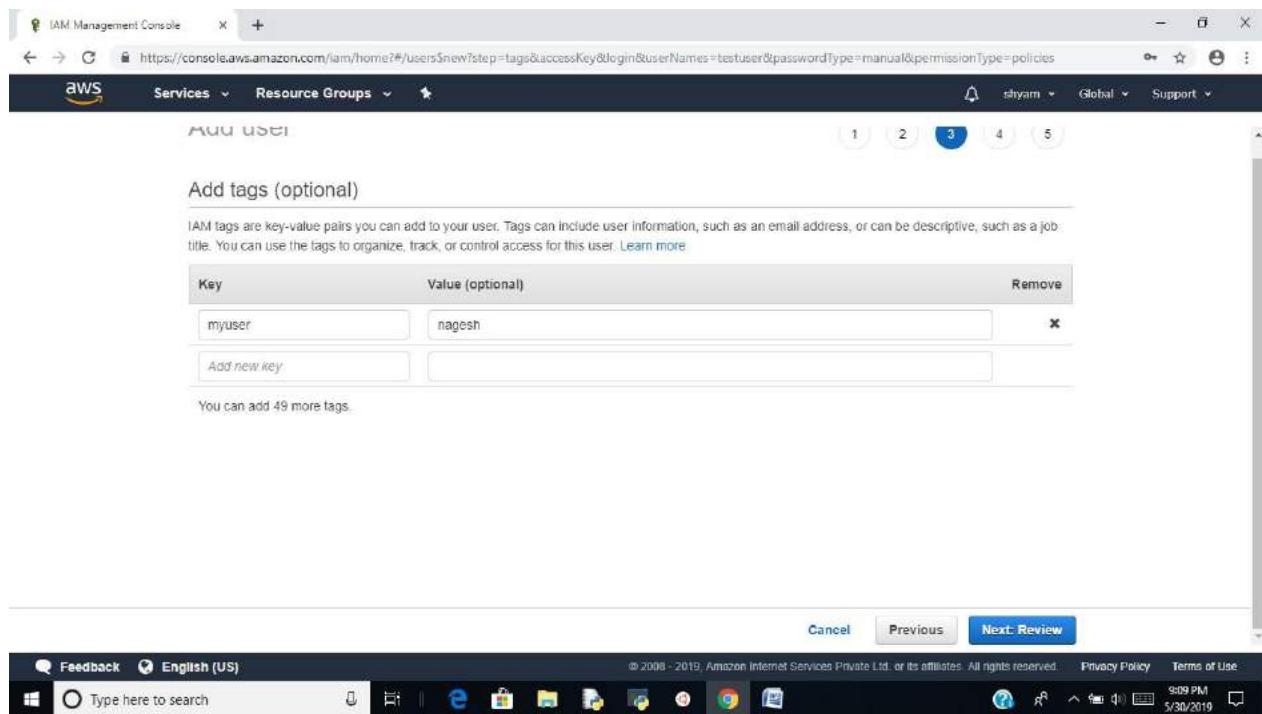
Attach existing policies directly: we can directly attach the existing policies to this user

Note: you can choose the one of the above option to attach policies to user. Here we select the attach existing policies directly

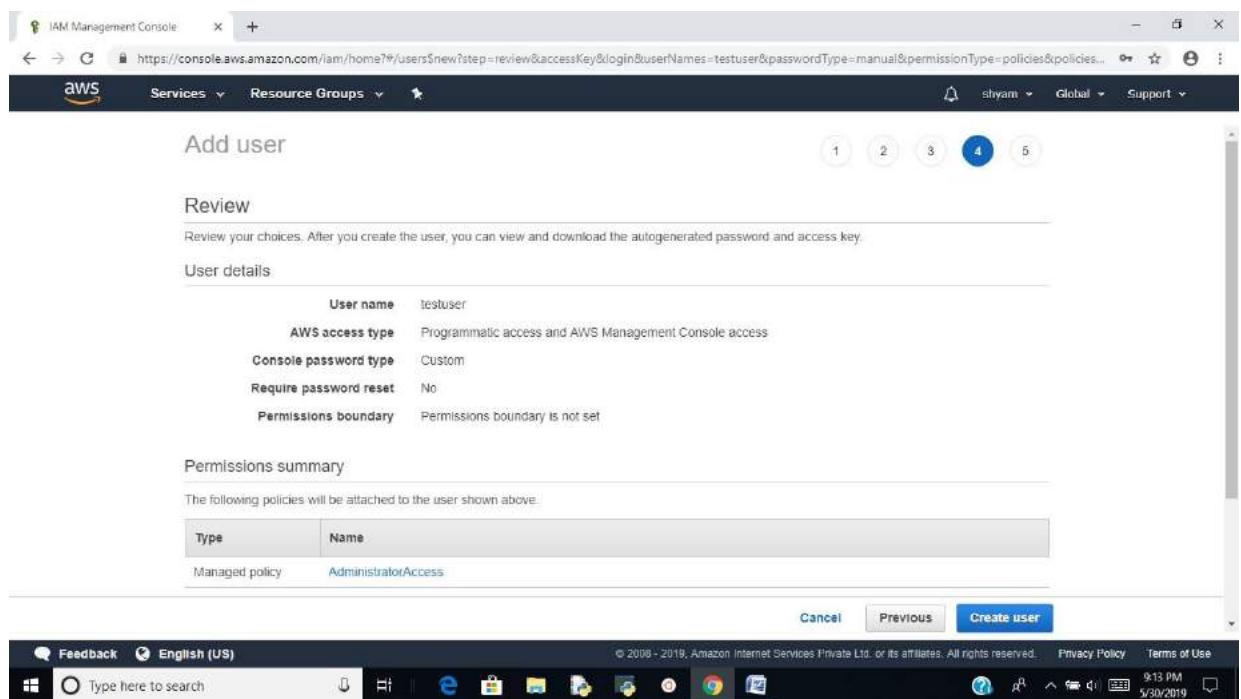
- Select required policy from list of policies and click on next



- Add tags: tags is the optional if do you want then enter key and value and click on next



- Review: see all details given by you and check these details and finally click on create user



- User was created

The screenshot shows the AWS IAM Management Console. At the top, there's a success message: "Success: You successfully created the users shown below. You can view and download user security credentials. You can also email users instructions for signing in to the AWS Management Console. This is the last time these credentials will be available to download. However, you can create new credentials at any time." Below this, a table lists a single user named "testuser". The table has columns for User, Access key ID, Secret access key, and Email login instructions. The Access key ID is listed as "AKIA33PLZ3BKNL6D47BD". There are "Show" and "Send email" buttons next to the secret access key column. A "Download .CSV" button is located above the table. The browser's address bar shows the URL: [https://console.aws.amazon.com/iam/home?#/users\\$new?step=final&accessKey&login&userNames=testuser&passwordType=manual&permissionType=policies&policies=...](https://console.aws.amazon.com/iam/home?#/users$new?step=final&accessKey&login&userNames=testuser&passwordType=manual&permissionType=policies&policies=...). The browser interface includes a toolbar with icons for Back, Forward, Stop, Refresh, Home, and others, along with a search bar and system status indicators.

- Click on close

Delete User

- Click on users on left side panel and select particular user which user do you want to delete and click on delete

The screenshot shows the AWS IAM Management Console. On the left, there's a sidebar with options like Dashboard, Groups, Users (which is selected), Roles, Policies, Identity providers, Account settings, Credential report, and Encryption keys. The main area has tabs for 'Add user' and 'Delete user'. A search bar at the top says 'Find users by username or access key'. Below it is a table titled 'Showing 2 results' with columns: User name, Groups, Access key age, Password age, Last activity, and MFA. Two users are listed: 'acct2user' (selected) and 'testuser'. The 'acct2user' row shows 'None' under Groups, 'Yesterday' under Access key age, 'Yesterday' under Password age, 'Today' under Last activity, and 'Not enabled' under MFA.

- Enable confirmation box and click on delete

The screenshot shows a 'Delete user' confirmation dialog box overlying the IAM console. The dialog box contains a message: 'The following users will be permanently deleted, including all user data, user security credentials, and user inline policies. Deleted user data cannot be recovered. Are you sure that you want to delete the following users?'. It lists 'User name' and 'Last activity' for the selected user 'acct2user' ('1 day ago'). Below this is a note: 'Recent activity usually appears within 4 hours. Data is stored for a maximum of 365 days, depending when your region began supporting this feature.' A checkbox is checked with the note: 'One or more of these users have recently accessed AWS. Deleting them could affect running systems. Check the box to confirm that you want to delete these users.' At the bottom are 'Cancel' and 'Yes, delete' buttons.

Edit permissions

- Click on user and click on permissions and click on Add permissions

The screenshot shows the AWS IAM Management Console interface. The URL in the browser is <https://console.aws.amazon.com/iam/home?region=us-east-1#users/testuser>. The left sidebar is collapsed. The main area shows the 'Summary' for the user 'testuser'. The 'Permissions' tab is active. Under 'Attached directly', there is one policy listed: 'AdministratorAccess' (AWS managed policy). Other tabs include 'Groups', 'Tags (1)', 'Security credentials', and 'Access Advisor'.

- Grant permissions: attaching the permissions by selecting from group, user or directly form existing policies
- Select the particular policy from list of policies and click on next

Add permissions to testuser

Grant permissions

Use IAM policies to grant permissions. You can assign an existing policy or create a new one.

Create policy

Filter policies Search Showing 439 results

	Policy name	Type	Used as	Description
<input type="checkbox"/>	AlexaForBusinessDeviceSetup	AWS managed	None	Provide device setup access to AlexaForBusiness services
<input type="checkbox"/>	AlexaForBusinessFullAccess	AWS managed	None	Grants full access to AlexaForBusiness resources and access to ...
<input type="checkbox"/>	AlexaForBusinessGateway	AWS managed	None	Provide gateway execution access to AlexaForBusiness services
<input type="checkbox"/>	AlexaForBusinessReadOnly	AWS managed	None	Provide read only access to AlexaForBusiness services
<input type="checkbox"/>	AmazonAPIGatewayAdmin	AWS managed	None	Provides full access to create/edit/delete APIs in Amazon API Ga...

Cancel **Next: Review**

- Click on Add permissions

Add permissions to testuser

Permissions summary

The following policies will be attached to the user shown above.

Type	Name
Managed policy	AlexaForBusinessDeviceSetup

Cancel **Previous** **Add permissions**

- Adding existing user to group: click on group and click on Add user to group by mentioning group name

The screenshot shows the AWS IAM Management Console interface. The top navigation bar includes the AWS logo, services dropdown, resource groups dropdown, user 'shyam', global dropdown, and support dropdown. The main menu on the left has options like Dashboard, Groups, Users (which is selected), Roles, Policies, Identity providers, Account settings, Credential report, and Encryption keys. The current page is 'Users > testuser'. The 'Summary' section displays User ARN: am:aws:iam:814927698004:user/testuser, Path: /, and Creation time: 2019-05-30 21:13 UTC+0530. Below this, the 'Groups' tab is selected, showing an 'Add user to groups' button. A table titled 'Attached permissions' shows 'Group name' and 'Attached permissions' columns, with 'No results' listed.

Security Credentials:

- Console sign-in-link: copy this url for access IAM user through AWS console
- Access keys: click on create access key to create access key and secret access key to use these IAM user in programmatic manner

The screenshot shows the AWS IAM Management Console. The URL is https://console.aws.amazon.com/iam/home?region=us-east-1#users/testuser)section=security_credentials. The left sidebar shows 'Users' selected. The top navigation bar has 'Services', 'Resource Groups', and 'Support'. The main content area has tabs: 'Permissions', 'Groups', 'Tags (1)', 'Security credentials' (selected), and 'Access Advisor'. Under 'Sign-in credentials', there's a 'Summary' section with a 'Console sign-in link' (https://signin.aws.amazon.com/console). Below it are sections for 'Console password' (Enabled, never signed in), 'Assigned MFA device' (Not assigned), and 'Signing certificates' (None). Under 'Access keys', there's a 'Create access key' button. A table lists one access key: Access key ID AKIA33PLZ3BKNL6D47BD, Created 2019-05-30 21:13 UTC+0530, Last used N/A, Status Active. The bottom of the screen shows a Windows taskbar with various icons and the date/time 10:09 AM 5/31/2019.

Signin into IAM user

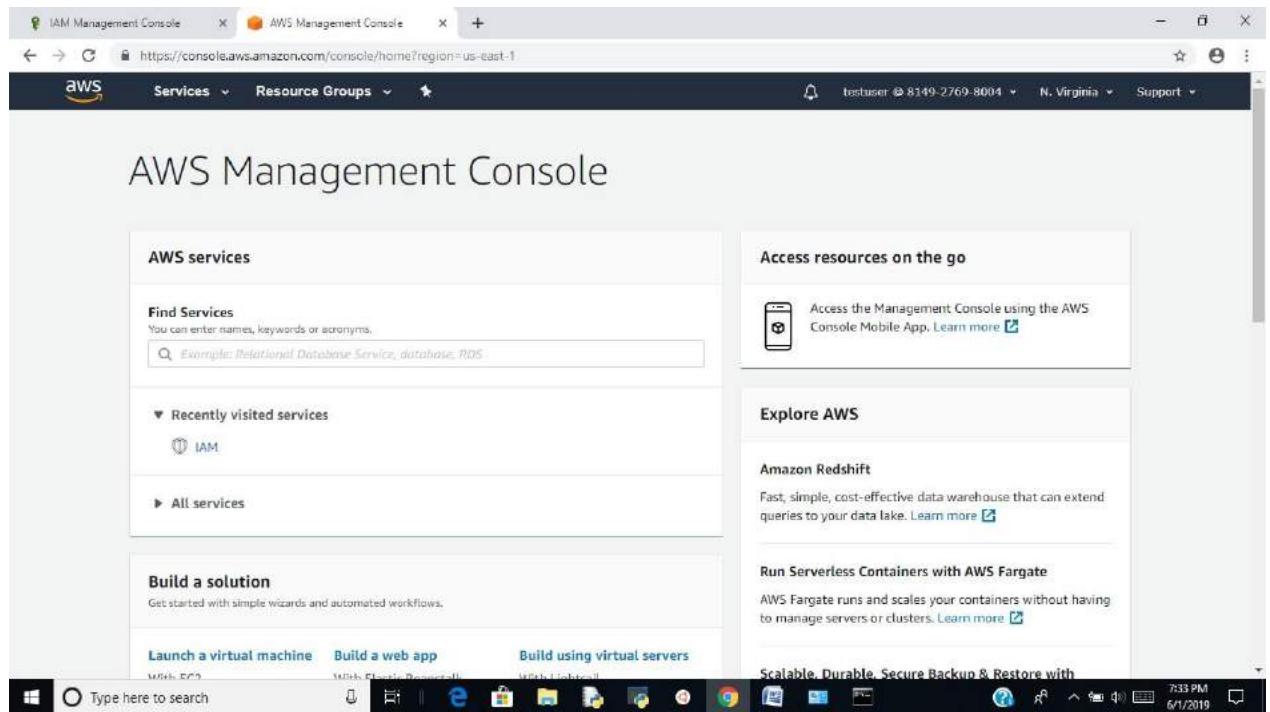
- Copy the sign-in url of particular user and paste in browser

The screenshot shows the AWS IAM Management Console interface. On the left, there's a navigation sidebar with options like Dashboard, Groups, Users (which is selected), Roles, Policies, Identity providers, Account settings, and Credential report. The main area is titled 'Summary' for the user 'testuser'. It shows the User ARN (arn:aws:iam:814927698004:user/testuser), Path (/), and Creation time (2019-05-30 21:13 UTC+0530). Below this, there are tabs for Permissions, Groups, Tags (1), Security credentials (which is active), and Access Advisor. Under 'Sign-in credentials', it shows a 'Console sign-in link' (https://814927698004.signin.aws.amazon.com/console), a 'Console password' status (Enabled, never signed in), an 'Assigned MFA device' status (Not assigned), and 'Signing certificates' (None). There's also a section for 'Access keys' with a note about frequent key rotation.

- Enter IAM username and password and click on login

The screenshot shows the AWS sign-in page. The URL is https://us-east-1signin.aws.amazon.com/oauth?SignatureVersion=4&X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=AKIAJMOATPLHVSJS563XQ8X-Amz-Date... The page has fields for 'Account ID or alias' (814927698004), 'IAM user name' (testuser), and 'Password'. Below these is a 'Sign In' button. To the right, there's an advertisement for 'Open Distro for Elasticsearch' featuring the text '100% open-source, community driven distribution of Elasticsearch' and an illustration of a search interface. The bottom of the screen shows a Windows taskbar with various icons and the date/time (7:29 PM 6/1/2019).

- Check the user permissions in this IAM user account

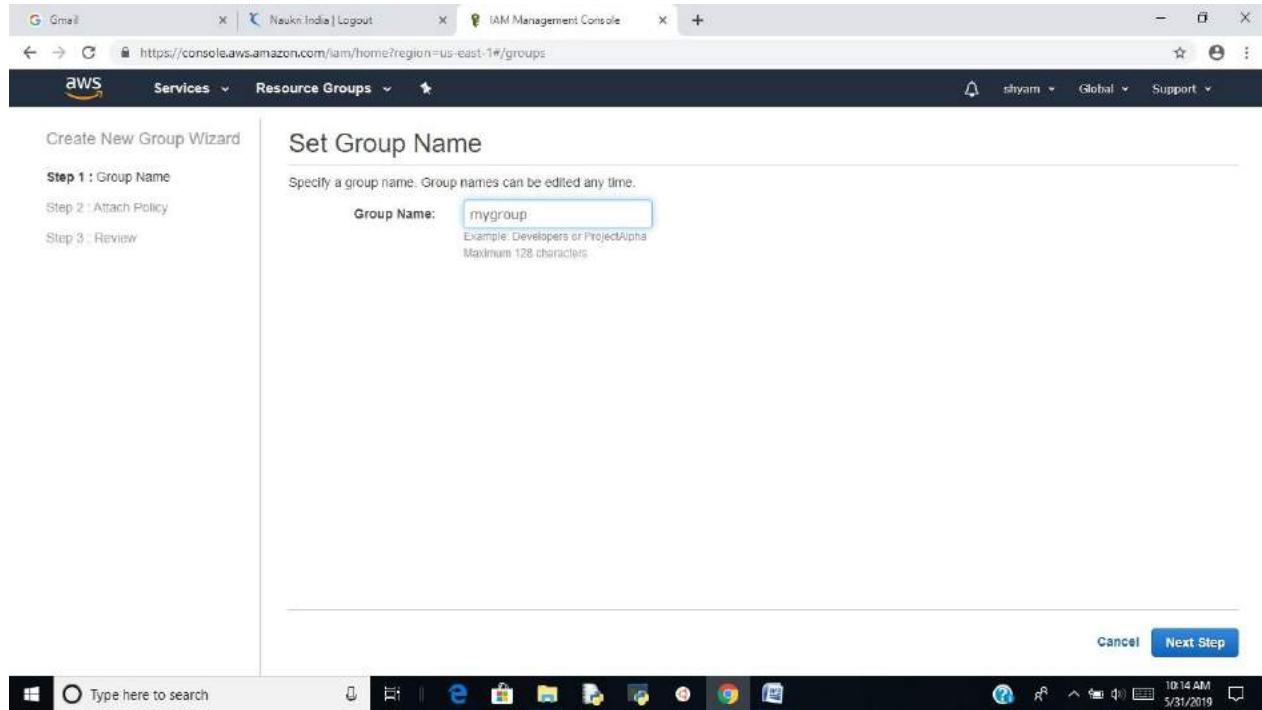


Groups

- Click on groups on left side panel of IAM service
- Click on create new group

A screenshot of the IAM Management Console Groups page. The top navigation bar shows tabs for 'Gmail' and 'Naukri India | Logout'. The main title 'IAM Management Console' is centered above a sidebar and content area. The sidebar on the left lists 'Dashboard', 'Groups' (which is selected and highlighted in orange), 'Users', 'Roles', 'Policies', 'Identity providers', 'Account settings', and 'Credential report'. The main content area shows a table with columns for 'Group Name', 'Users', 'Inline Policy', and 'Creation Time'. A search bar at the top of the table allows filtering by 'Group Name'. A message at the bottom of the table states 'No records found.' A Windows taskbar is visible at the bottom.

- Set Group Name: specify name for this group and click on next step



- Attach Policy: select one particular policy from list of policy and click on next step

The screenshot shows the 'Attach Policy' step of the 'Create New Group Wizard' in the AWS IAM Management Console. The left sidebar shows the steps: Step 1: Group Name, Step 2: Attach Policy, and Step 3: Review. The main area is titled 'Attach Policy' with the sub-instruction 'Select one or more policies to attach. Each group can have up to 10 policies attached.' Below this is a table listing 440 policies, filtered by 'Policy Type'. The table columns are 'Policy Name', 'Attached Entities', 'Creation Time', and 'Edited Time'. A checkbox next to each policy name allows selection. The policies listed include 'AdministratorAccess', 'AlexaForBusinessDeviceSetup', 'AlexaForBusinessFullAccess', 'AlexaForBusinessGatewayExe...', 'AlexaForBusinessReadOnlyAc...', 'AmazonAPIGatewayAdministr...', 'AmazonAPIGatewayInvokeFull...', 'AmazonAPIGatewayPushToCli...', and 'AmazonAppStreamFullAccess'. At the bottom right of the table are 'Cancel', 'Previous', and 'Next Step' buttons.

Click on create group

Add user to group: select group and goto group actions and select Add user to Group option

The screenshot shows the 'Group Actions' dropdown menu for a group named 'mygroup' in the AWS IAM Management Console. The menu options are 'Add Users to Group', 'Delete Group', 'Edit Group Name', and 'Remove Users from Group'. The 'Add Users to Group' option is highlighted with a black box. The main pane shows a table with one result for 'mygroup', with columns for 'Group Name', 'Users', 'Inline Policy', and 'Creation Time'. The table shows a single entry for 'mygroup' created on '2019-05-31 10:17 UTC+0530'. The left sidebar shows the navigation menu with 'Groups' selected. The bottom of the screen shows the Windows taskbar with various pinned icons.

- Select users from list of user and click on Add users
- Remove users from group: select group and goto group actions and click on Remove users from group

The screenshot shows the AWS IAM Management Console with the Groups page selected. A context menu is open over a group named 'mygroup'. The menu includes options like 'Create New Group', 'Group Actions', 'Add Users to Group', 'Delete Group', 'Edit Group Name', and 'Remove Users from Group'. The 'Remove Users from Group' option is highlighted.

- Select user which user do want to remove from group and click on remove users

Select users to remove from the group **mygroup**

User Name	Groups	Password	Password Last Used	Access Keys	Creation Time
<input checked="" type="checkbox"/> testuser	1	✓	Never	1 active	2019-05-30 21:13 U...

Cancel **Remove Users**

- Delete Group: select group and goto Group Actions and click on delete group

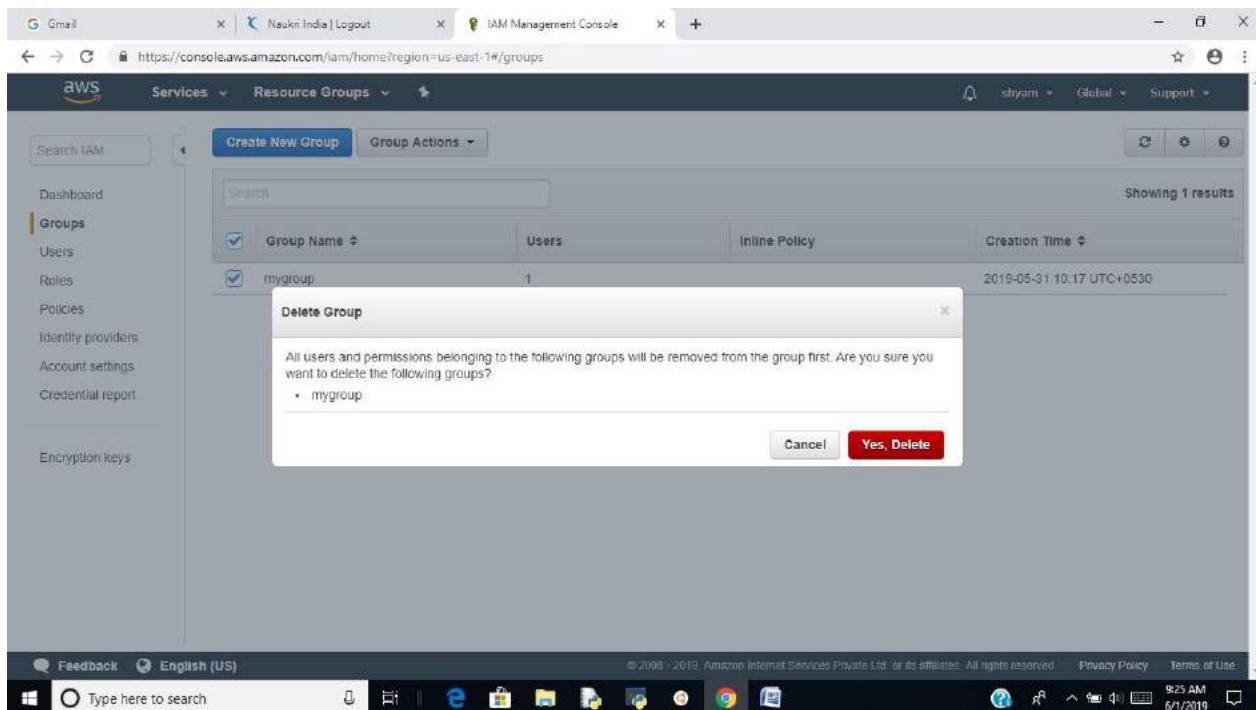
Create New Group **Group Actions** ▾

Delete Group

Group Name	Users	Inline Policy	Creation Time
<input checked="" type="checkbox"/> mygroup	1	Never	2019-05-31 10:17 UTC+0530

Feedback English (US) © 2006 - 2019, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. Privacy Policy Terms of Use

- Click on yes delete

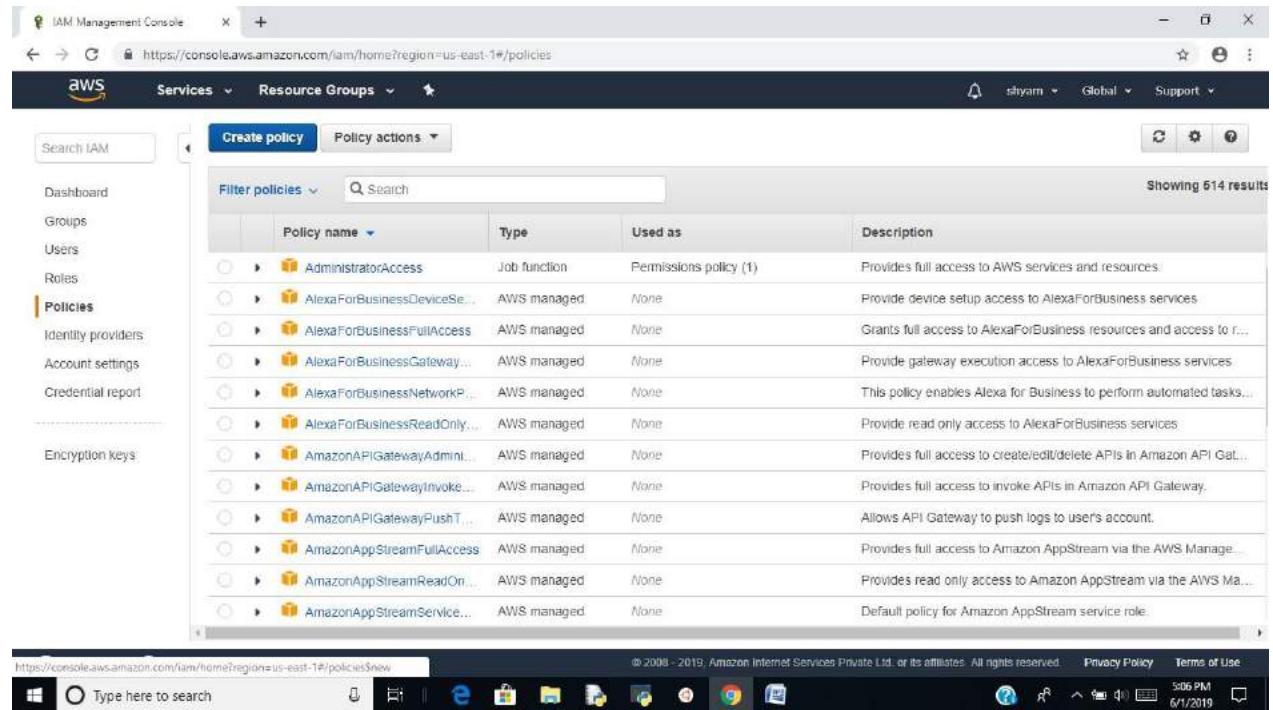


Policies

- There are two types of policies one is Identity-based policy and another one is resource-based policy
- Identity-Based Policies: These policies attached to Identifiers like users, groups and roles to give restricted access on resources to these identifiers
- Resource-Based Policies: These policies attached to resources to give
- Before writing policies we need to know some attributes
- Effect: This parameter is used to provide permission type that is allow or deny the particular service
- Action: level of accessibility in a particular service here we use wild cards (*)
- (*) indicates all actions on particular service
- Resource: include the particular resource here * indicates all resources on AWS
- Condition: you can include the condition section in policies so the policy or single permission from list of permissions is applied only condition is true.

Create a policy for administration access to IAM users

- Click on create policy



The screenshot shows the AWS IAM Management Console interface. The left sidebar has a 'Policies' section selected. The main area displays a table of existing policies with columns for Policy name, Type, Used as, and Description. A 'Create policy' button is visible at the top of the main content area.

Policy name	Type	Used as	Description
AdministratorAccess	Job function	Permissions policy (1)	Provides full access to AWS services and resources.
AlexaForBusinessDeviceSe...	AWS managed	None	Provide device setup access to AlexaForBusiness services
AlexaForBusinessFullAccess	AWS managed	None	Grants full access to AlexaForBusiness resources and access to r...
AlexaForBusinessGateway...	AWS managed	None	Provide gateway execution access to AlexaForBusiness services
AlexaForBusinessNetworkP...	AWS managed	None	This policy enables Alexa for Business to perform automated tasks...
AlexaForBusinessReadOnly...	AWS managed	None	Provide read only access to AlexaForBusiness services
AmazonAPIGatewayAdmini...	AWS managed	None	Provides full access to create/edit/delete APIs in Amazon API Gat...
AmazonAPIGatewayInvoke...	AWS managed	None	Provides full access to invoke APIs in Amazon API Gateway
AmazonAPIGatewayPushT...	AWS managed	None	Allows API Gateway to push logs to users account.
AmazonAppStreamFullAccess	AWS managed	None	Provides full access to Amazon AppStream via the AWS Manage...
AmazonAppStreamReadOn...	AWS managed	None	Provides read only access to Amazon AppStream via the AWS Ma...
AmazonAppStreamService...	AWS managed	None	Default policy for Amazon AppStream service role.

- There are two ways to create a policy one is visual editor and another one is JSON
- Visual editor is the graphical mode and JSON is the programming mode or script format

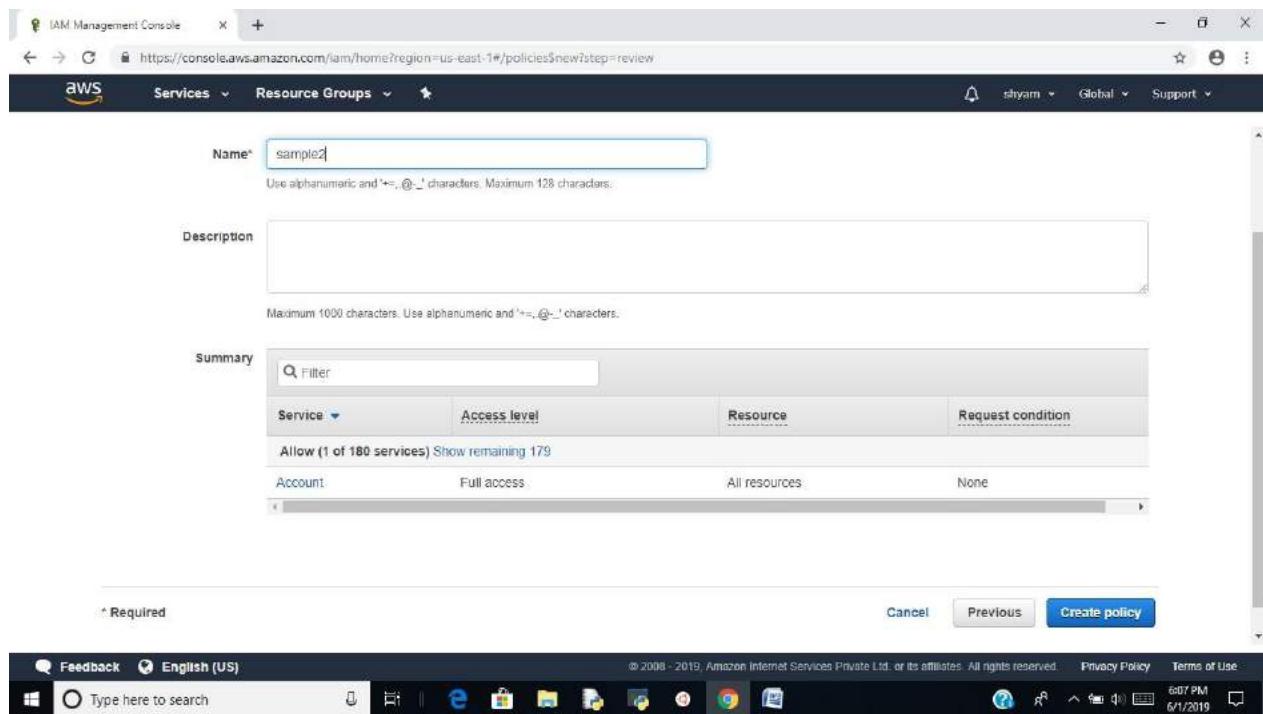
Visual editor

- Service: click on choose a service and select one service from list of services
- Enter service name and click on that service and select Account
- Actions: specify the actions allowed into account that is read action or write action or both or all actions (*)
- Here we select All account actions
- Resources: Resource section depend on Action section. Here we choose all actions so the resource is selected as all resources
- Request Conditions: it is the condition applied when client make request to access the service
- MFA required: require users to authenticate with MFA device to perform the specified actions

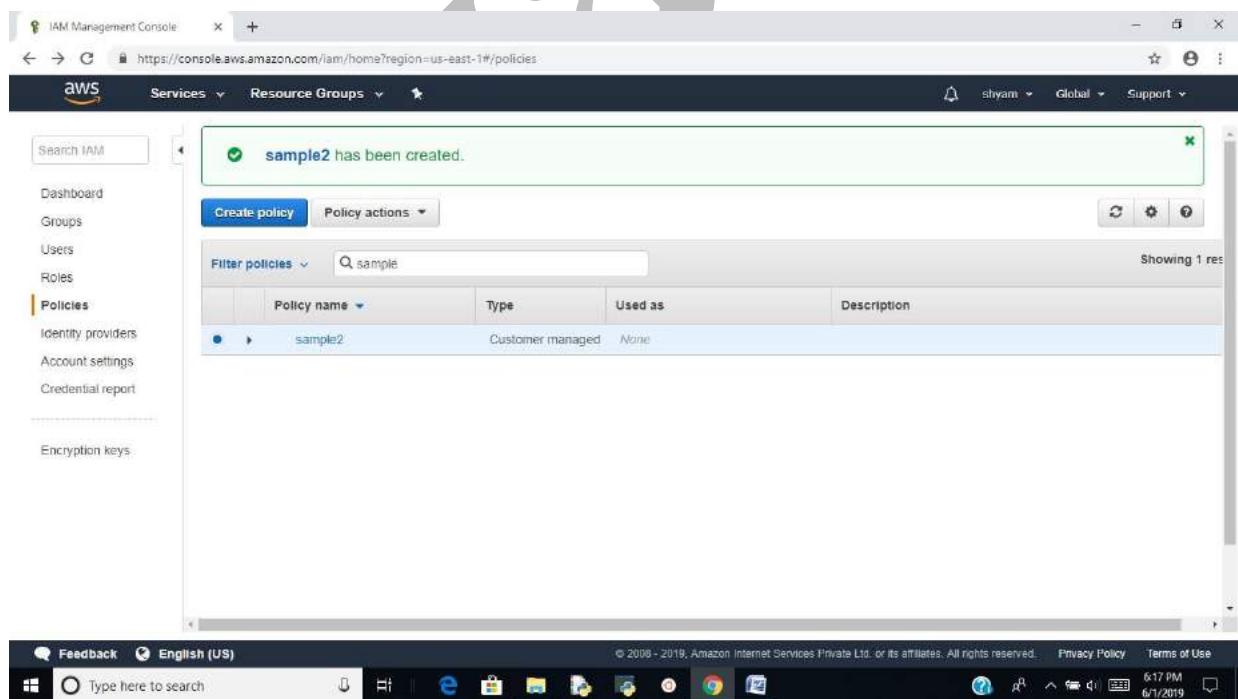
- Source IP: allow access to the specified actions only when the request comes from the specified IP address range

The screenshot shows the AWS IAM Management Console interface. A large watermark 'DyaSaa' is overlaid on the page. At the top, the URL is https://console.aws.amazon.com/iam/home?region=us-east-1#/policies\$new?step=edit. The main area displays a policy editor with tabs for 'Visual editor' and 'JSON'. The 'Visual editor' tab is active. Under the 'Account' section, there are two conditions: 'MFA required' (unchecked) and 'Source IP' (unchecked). The 'Source IP' condition is described as allowing access to specified actions only when the request comes from a specified IP address range. There are buttons for 'Clone' and 'Remove' at the top right of the condition list.

- Click on review policy
- Name: give name for policy
- Click on create policy



- Do you want see that policy then search with that name in policies



JSON

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": "*",  
            "Resource": "*"  
        }  
    ]  
}
```

Create a policy allow for ec2 full access

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

Create a policy for deny ec2 full access

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Deny",  
            "Action": "ec2:*",  
            "Resource": "*"  
        }  
    ]  
}
```

}

Create a policy for ec2 read only access

{

"Version": "2012-10-17",

"Statement": [

{

"Effect": "Allow",

"Action": "ec2:Describe*",

"Resource": "*"

},

{

"Effect": "Allow",

"Action": "elasticloadbalancing:Describe*",

"Resource": "*"

},

{

"Effect": "Allow",

"Action": [

"cloudwatch:ListMetrics",

"cloudwatch:GetMetricStatistics",

"cloudwatch:Describe*"

],

"Resource": "*"

```

    },
    {
        "Effect": "Allow",
        "Action": "autoscaling:Describe*",
        "Resource": "*"
    }
]
}

```

permissions

- Click on policy and goto permissions section then we see the list of permission to attached to that policy

The screenshot shows the AWS IAM Management Console with the URL [https://console.aws.amazon.com/iam/home?region=us-east-1#/policies/arn:aws:iam::aws:policy/AmazonEC2FullAccess\\$serviceLevelSummary?section=permissions](https://console.aws.amazon.com/iam/home?region=us-east-1#/policies/arn:aws:iam::aws:policy/AmazonEC2FullAccess$serviceLevelSummary?section=permissions). The left sidebar is collapsed. The main area shows the 'AmazonEC2FullAccess' policy summary. The 'Permissions' tab is active, showing a table of service access levels and resources:

Service	Access level	Resource	Request condition
CloudWatch	Full access	All resources	None
EC2	Full access	All resources	None
EC2 Auto Scaling	Full access	All resources	None
ELB	Full access	All resources	None
ELB v2	Full access	All resources	None
IAM	Limited: Write	All resources	Multiple

Policy usage

- Click on the policy and select policy usage

- Here two options one is attach policy and another one in detach policy by selecting particular users

The screenshot shows the AWS IAM Management Console with the URL <https://console.aws.amazon.com/iam/home?region=us-east-1#/policies/arn:aws:iam::aws:policy/AmazonEC2FullAccess>. The left sidebar has 'Policies' selected. The main page displays the 'AmazonEC2FullAccess' policy summary. The 'Permissions' tab is active, showing an 'Attach' button and a search/filter bar. The results table is empty, showing 'Showing 0 results'. The top navigation bar includes the AWS logo, services dropdown, resource groups dropdown, user dropdown (shyam), global dropdown, and support dropdown.

Policy Versions

- To see the all versions of policy click on policy versions

Policy ARN: arn:aws:iam::aws:policy/AmazonEC2FullAccess

Description: Provides full access to Amazon EC2 via the AWS Management Console.

Version	Creation time
Version 5 (Default)	2018-11-27 07:46 UTC+0530
Version 4	2018-02-08 23:41 UTC+0530
Version 3	2018-01-12 01:46 UTC+0530
Version 2	2017-10-31 04:06 UTC+0530
Version 1	2015-02-07 00:10 UTC+0530

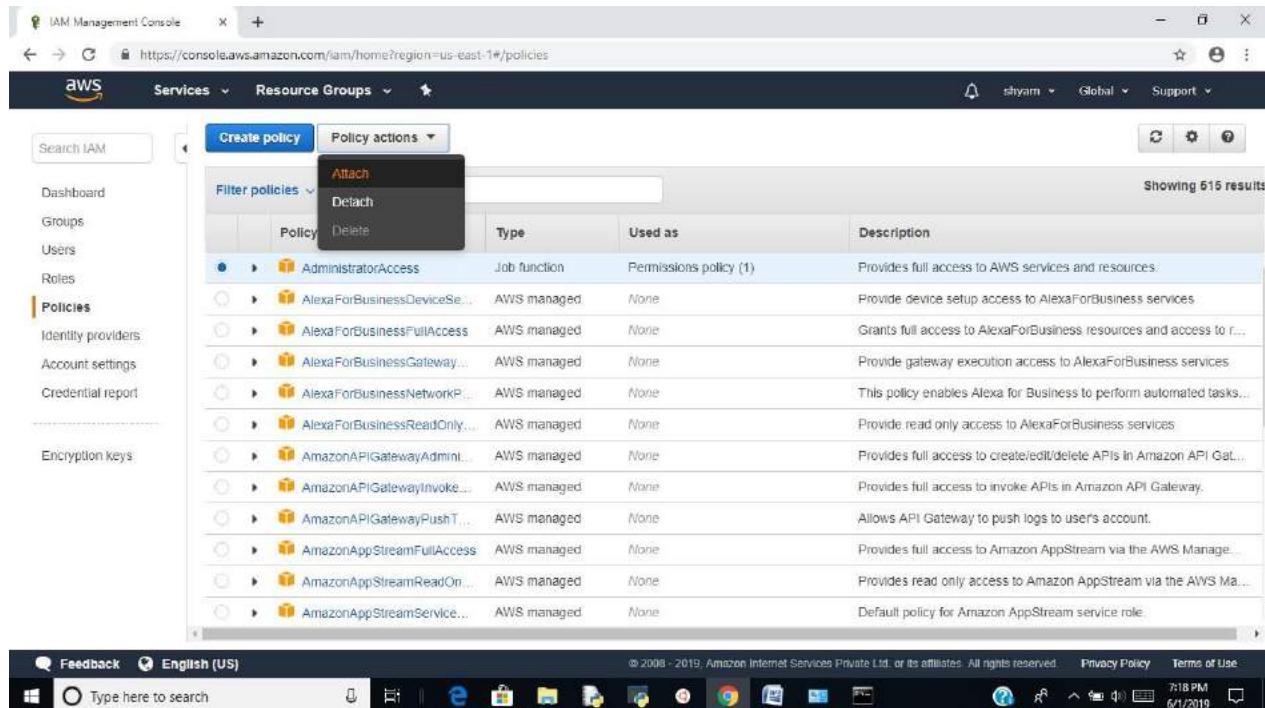


- We can also attach and detach a policy by selecting policy and goto policy actions and select attach or detach

Policy	Type	Used as	Description
AdministratorAccess	Job function	Permissions policy (1)	Provides full access to AWS services and resources.
AlexaForBusinessDeviceSe...	AWS managed	None	Provide device setup access to AlexaForBusiness services.
AlexaForBusinessFullAccess	AWS managed	None	Grants full access to AlexaForBusiness resources and access to r...
AlexaForBusinessGateway...	AWS managed	None	Provide gateway execution access to AlexaForBusiness services.
AlexaForBusinessNetworkP...	AWS managed	None	This policy enables Alexa for Business to perform automated tasks...
AlexaForBusinessReadOnly...	AWS managed	None	Provide read only access to AlexaForBusiness services.
AmazonAPIGatewayAdmini...	AWS managed	None	Provides full access to create/edit/delete APIs in Amazon API Gat...
AmazonAPIGatewayInvoke...	AWS managed	None	Provides full access to invoke APIs in Amazon API Gateway.
AmazonAPIGatewayPushT...	AWS managed	None	Allows API Gateway to push logs to user's account.
AmazonAppStreamFullAccess	AWS managed	None	Provides full access to Amazon AppStream via the AWS Manage...
AmazonAppStreamReadOn...	AWS managed	None	Provides read only access to Amazon AppStream via the AWS Ma...
AmazonAppStreamService...	AWS managed	None	Default policy for Amazon AppStream service role.

Delete policy

- Select policy and goto policy actions and click on delete



The screenshot shows the AWS IAM Management Console. On the left, there's a navigation sidebar with options like Dashboard, Groups, Users, Roles, Policies (which is selected), Identity providers, Account settings, Credential report, and Encryption keys. The main area displays a table of policies with columns for Policy, Type, Used as, and Description. A context menu is open over one of the rows, with 'Delete' being the option highlighted. The URL in the browser is https://console.aws.amazon.com/iam/home?region=us-east-1#/policies.

Policy	Type	Used as	Description
AdministratorAccess	Job function	Permissions policy (1)	Provides full access to AWS services and resources.
AlexaForBusinessDeviceSe...	AWS managed	None	Provide device setup access to AlexaForBusiness services.
AlexaForBusinessFullAccess	AWS managed	None	Grants full access to AlexaForBusiness resources and access to r...
AlexaForBusinessGateway...	AWS managed	None	Provide gateway execution access to AlexaForBusiness services.
AlexaForBusinessNetworkP...	AWS managed	None	This policy enables Alexa for Business to perform automated tasks...
AlexaForBusinessReadOnly...	AWS managed	None	Provide read only access to AlexaForBusiness services.
AmazonAPIGatewayAdmini...	AWS managed	None	Provides full access to create/edit/delete APIs in Amazon API Gat...
AmazonAPIGatewayInvoke...	AWS managed	None	Provides full access to invoke APIs in Amazon API Gateway.
AmazonAPIGatewayPushT...	AWS managed	None	Allows API Gateway to push logs to users account.
AmazonAppStreamFullAccess	AWS managed	None	Provides full access to Amazon AppStream via the AWS Manage...
AmazonAppStreamReadOnly...	AWS managed	None	Provides read only access to Amazon AppStream via the AWS Ma...
AmazonAppStreamService...	AWS managed	None	Default policy for Amazon AppStream service role.

Roles

- IAM roles are secure way to grant permissions to entities that you trust. Example of entities include the following.
- IAM users in another account
- Application code running on Ec2 instance that needs to perform actions on AWS resource

IAM users in another account

- Select roles in IAM service and click on roles

The screenshot shows the AWS IAM Management Console with the URL <https://console.aws.amazon.com/iam/home?region=us-east-1#roles>. The left sidebar has a 'Roles' link selected. The main content area is titled 'Roles' and contains information about IAM roles, a list of additional resources, and buttons for 'Create role' and 'Delete role'. A search bar at the bottom is set to 'Showing 2 results'.

- Select type of trusted entity: Here we choose another AWS account entity
- Specify accounts that can use this role: give another aws account id and click on Next:permissions

Create role

Select type of trusted entity

AWS service EC2, Lambda and others

Another AWS account Belonging to you or 3rd party

Web identity Cognito or any OpenID provider

SAML 2.0 federation Your corporate directory

Allows entities in other accounts to perform actions in this account. [Learn more](#)

Specify accounts that can use this role

Account ID* 170831088255

Options Require external ID (Best practice when a third party will assume this role)
 Require MFA

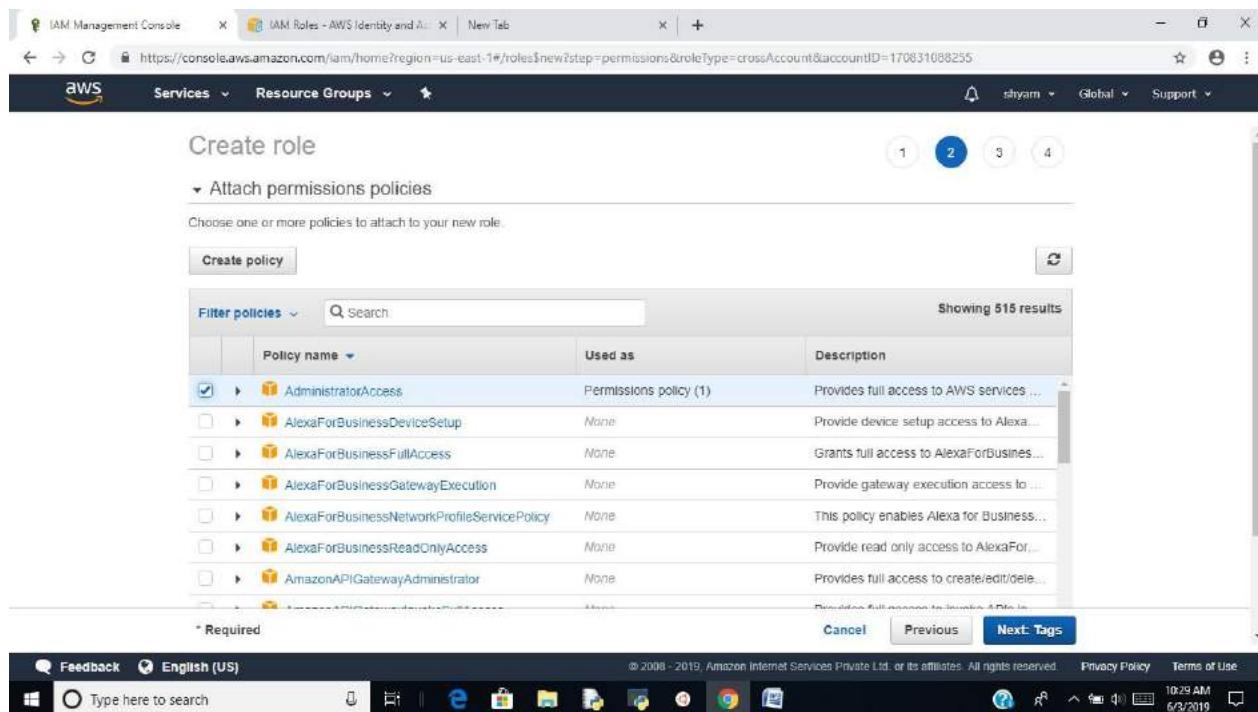
* Required

Feedback English (US)

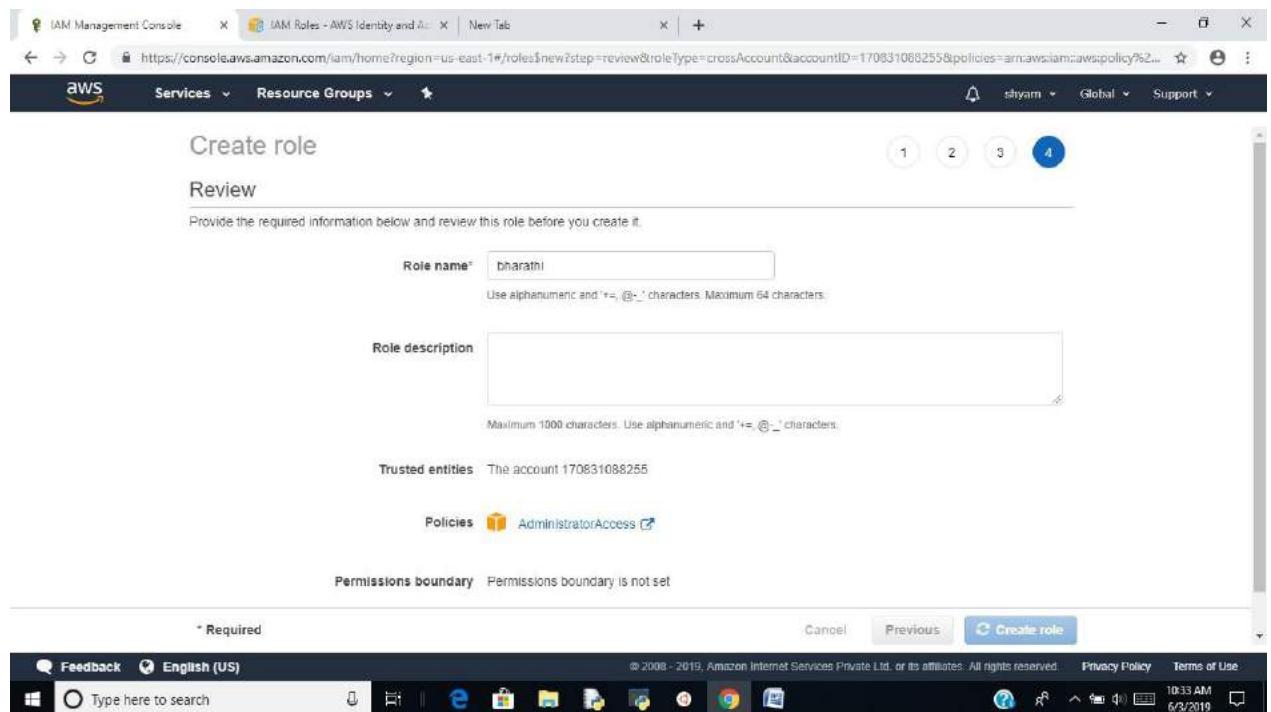
Type here to search

Cancel Next: Permissions

- Attaching permissions policies: here we attach existing policies previously created or attach new policy by click on create policy.
- Here I want give my AWS account full permissions to another aws account IAM user so I select ec2 full permissions policy
- You select any policy which type of permissions do want to apply
- Click on next tags



- Tags is the optional if you want to mention key and value pair and click on next review
- Rolename: enter name to this role
- Role Description: write some description to this role and it is the optional
- Click on create role



Assume role

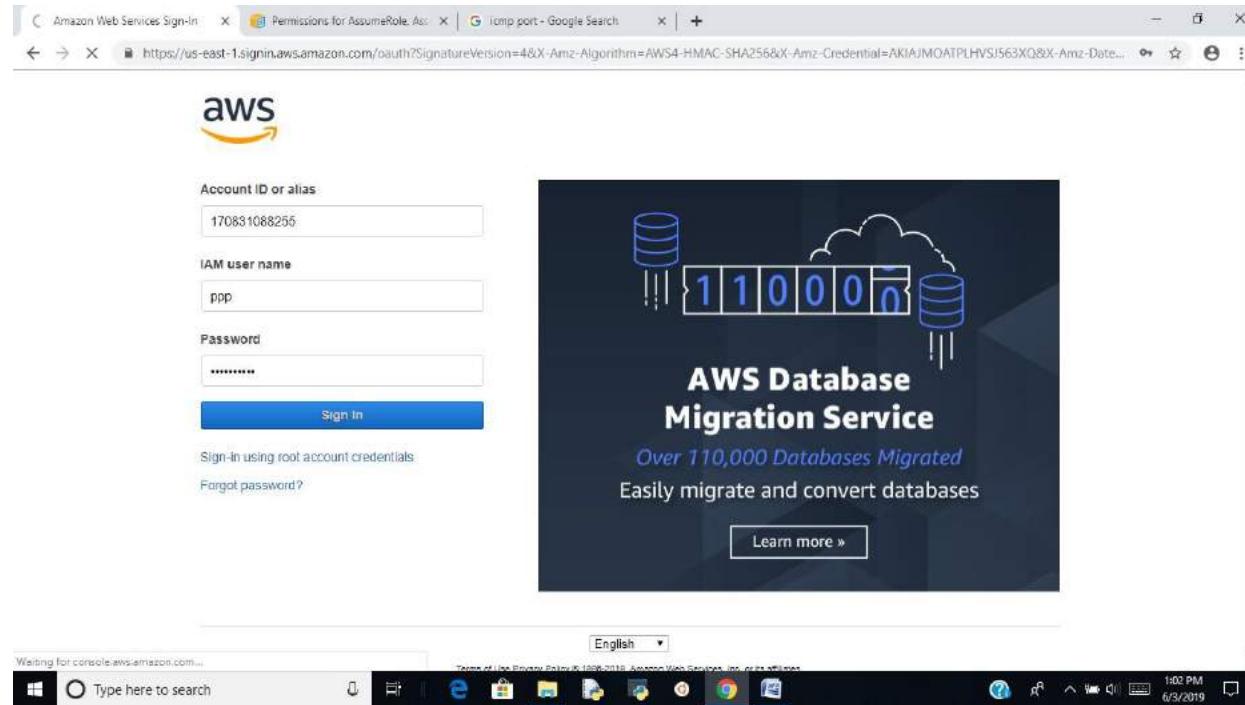
- Login into another AWS root account and create policy for assume role by selecting STS service
- Create policy using json or visual editor

```
{
  "Version": 2012-10-17,
  "Statement": [
    {
      "Sid": "VisualEditor0",
      "Effect": "Allow",
      "Action": "sts:*",
      "Resource": "*"
    }
  ]
}
```

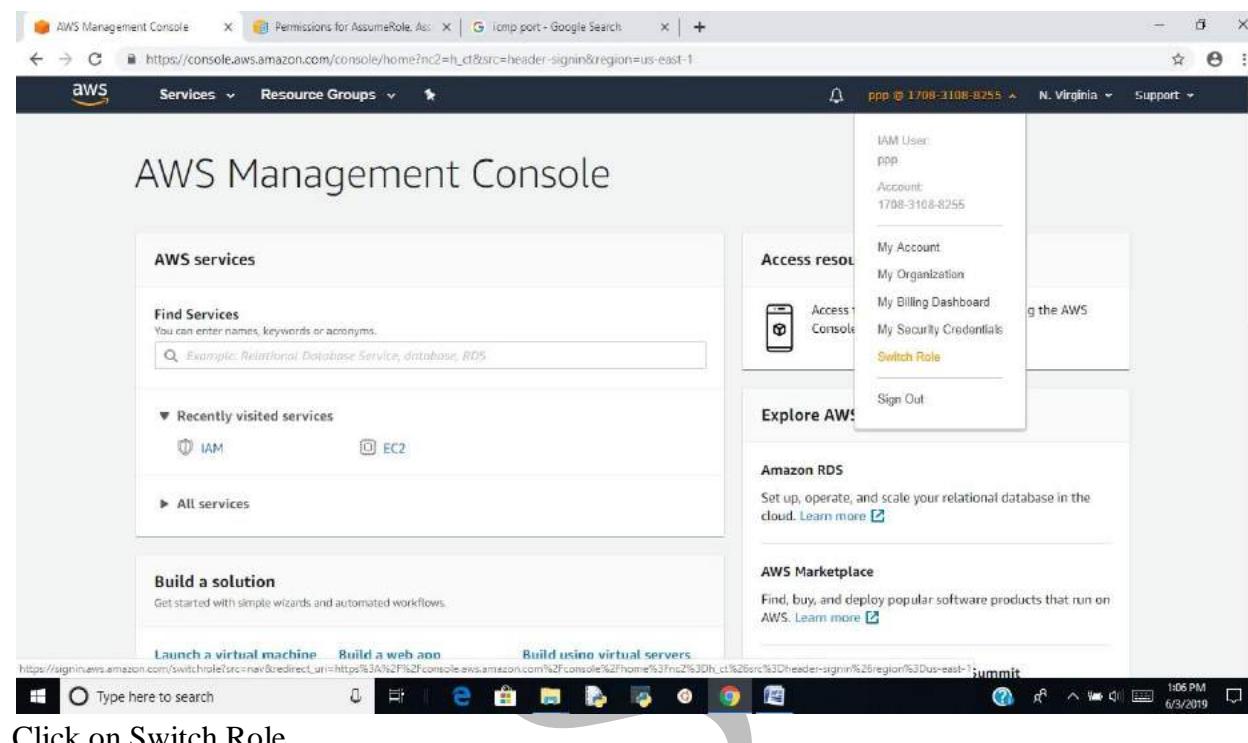
- Attach this policy to IAM user

Switch Role:

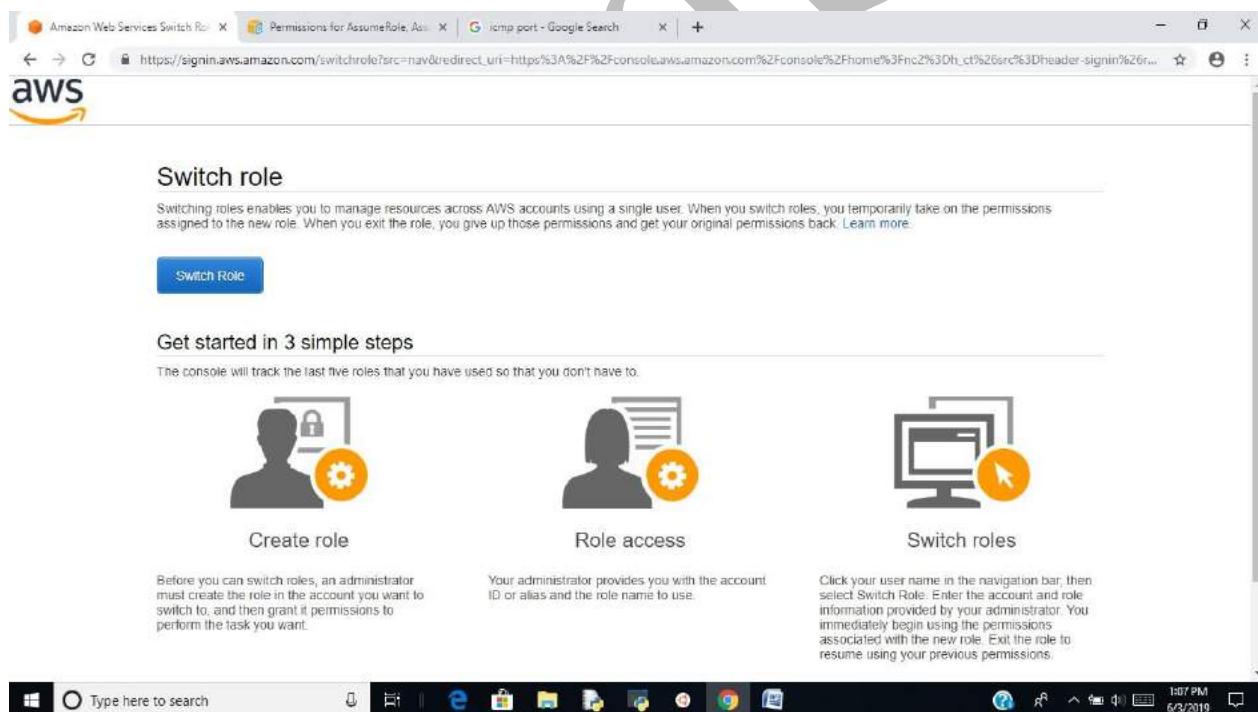
- Signin into another aws account IAM user using root account id, IAM user name and IAM user password



- Click on IAM user id on top right corner and select Switch Role option and click on that



- Click on Switch Role

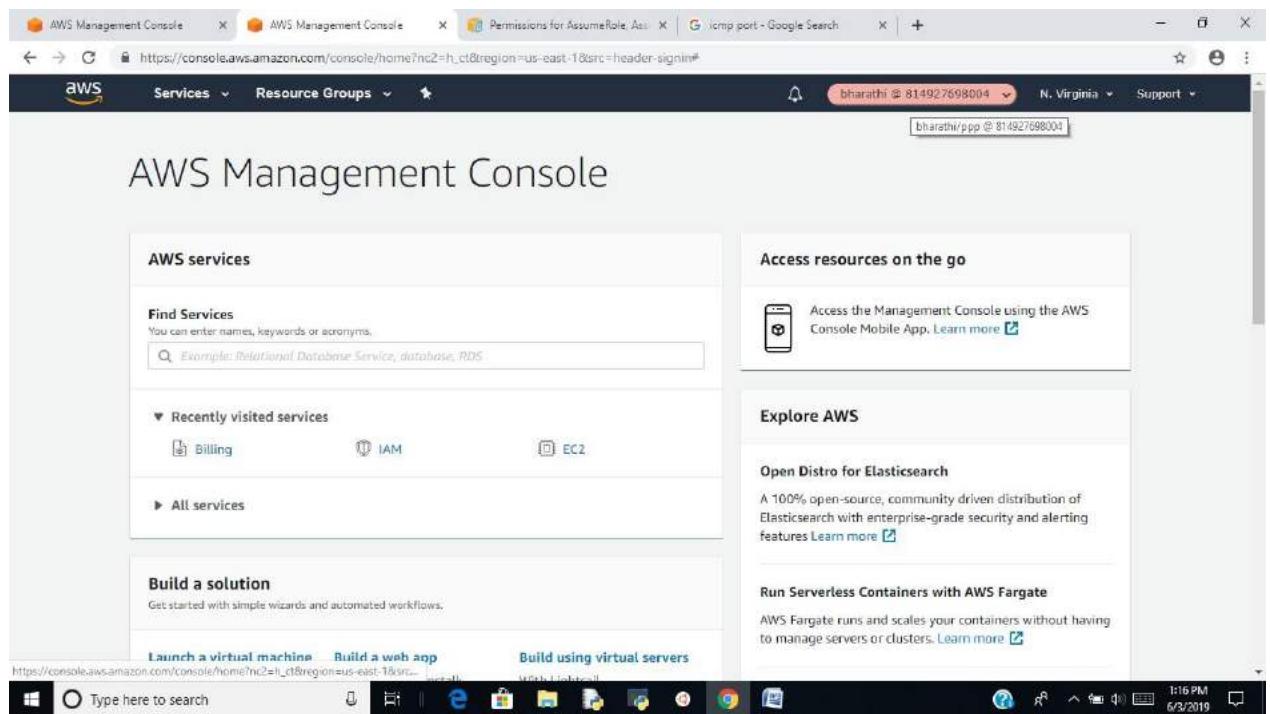


- Account: enter aws account number in which account the role is created

- Role: Enter role name
- Display Name: enter Display name
- Click on switch role

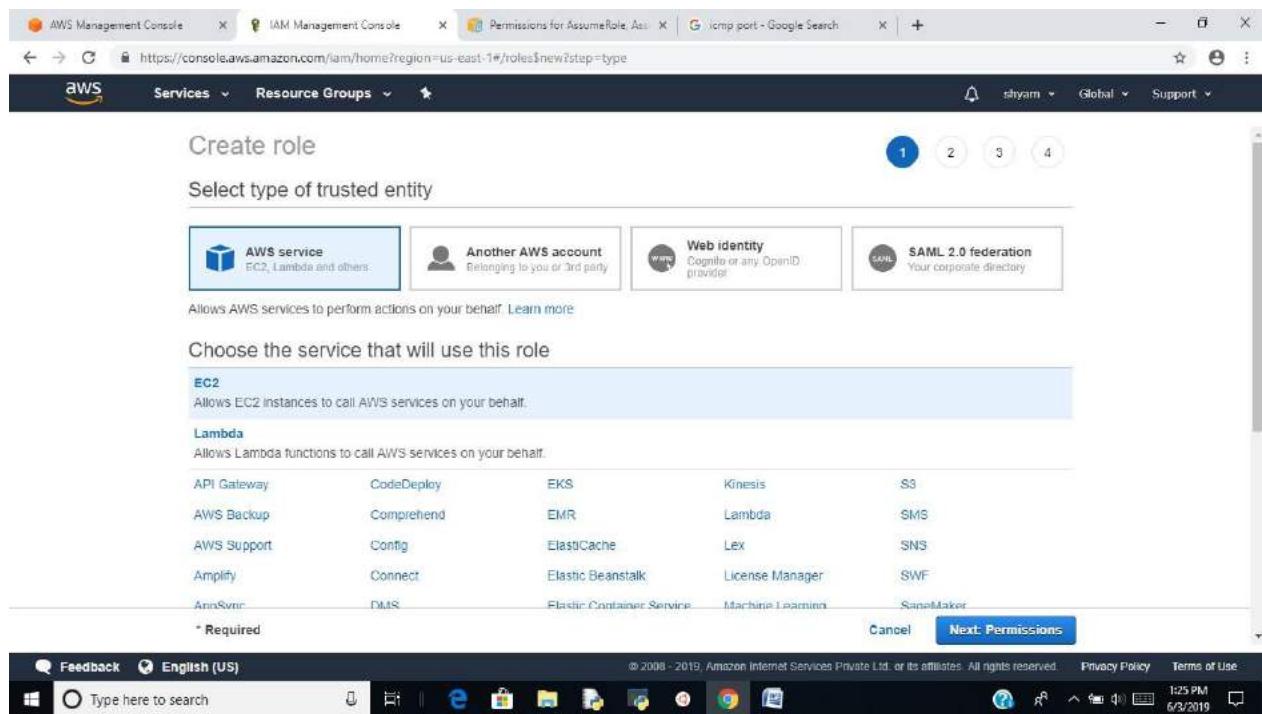
The screenshot shows the 'Switch Role' page in the AWS Management Console. The URL in the address bar is https://signin.aws.amazon.com/switchrole?src=nav&redirect_uri=https%3A%2F%2Fconsole.aws.amazon.com%2Fconsole%2Fhome%3Fnc2%3Dh_ct%26src%3Dheader-signin%26r.... The page includes fields for Account (814927690004), Role (bharathi), and Display Name (bharathi @ 814927690004). A color picker shows a red square selected. Buttons for 'Required' and 'Switch Role' are present. The background features a large, semi-transparent watermark of the letters 'Dya'.

- Then you can use another aws services which services are included in that role

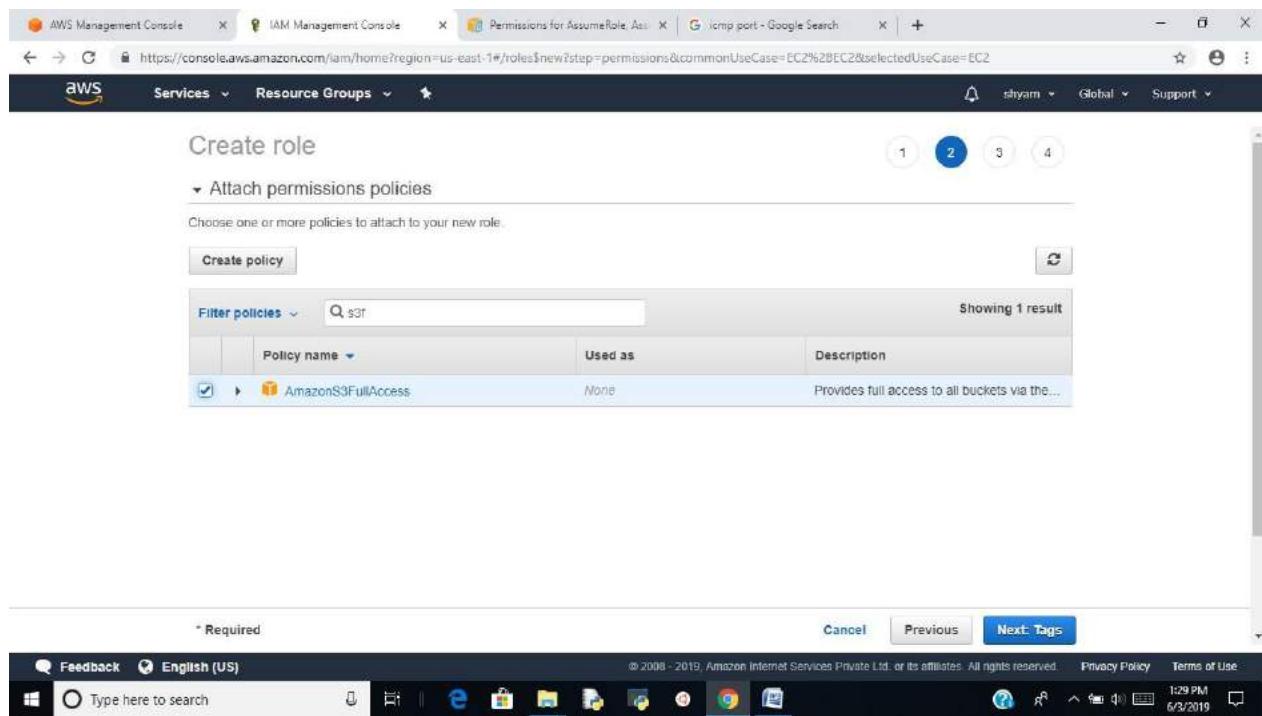


Role for aws Resources

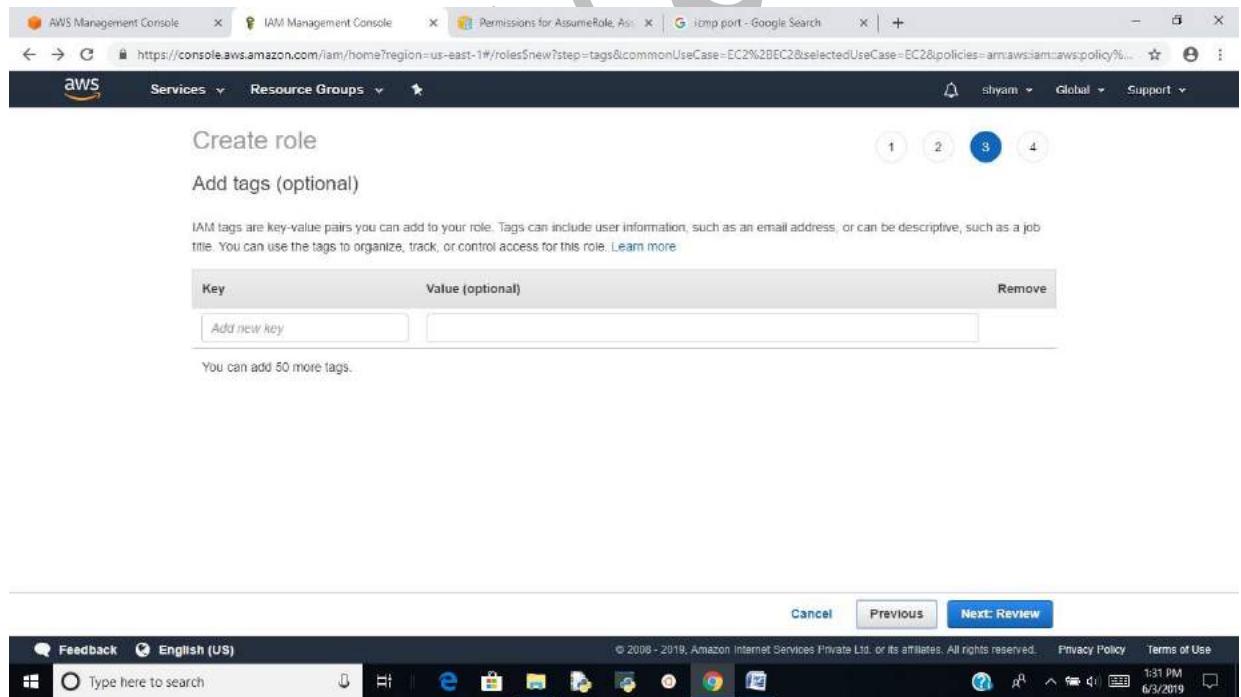
- Select IAM service and click on roles
- Click on create role
- Select type of trusted entity: choose AWS Service
- Choose the service that will use this role: select EC2
- Click on Next: permissions



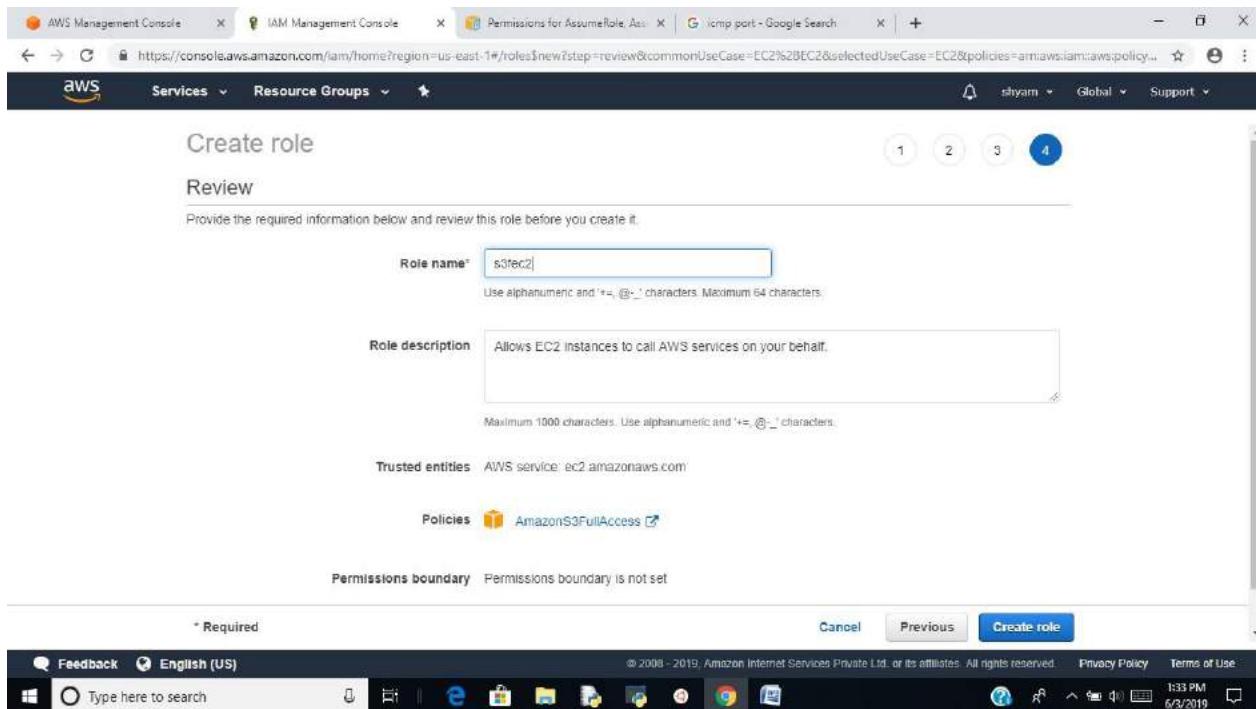
- Choose the policy for this role from list of policies or create new policy
- Here I want to give S3 full permissions to my ec2 instance so I choose policy for s3 full permissions and click on next



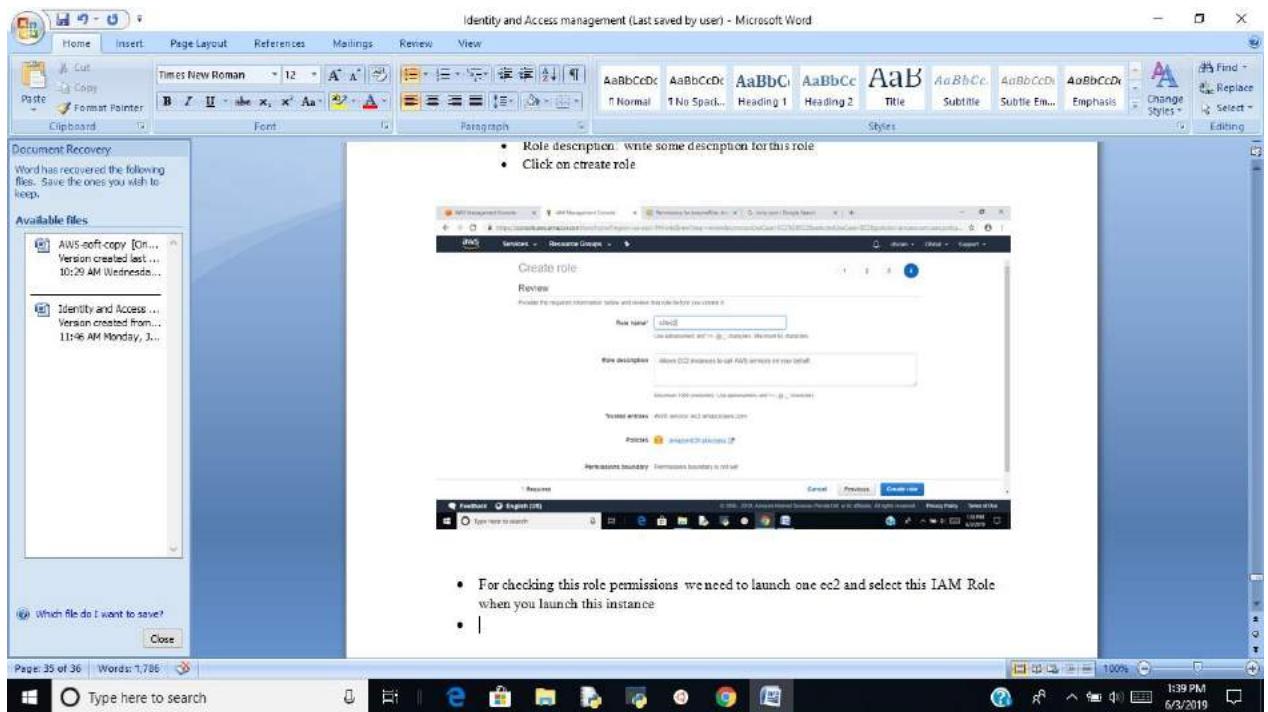
- Tags is the optional do you want to add tags enter key and value pair and click on next



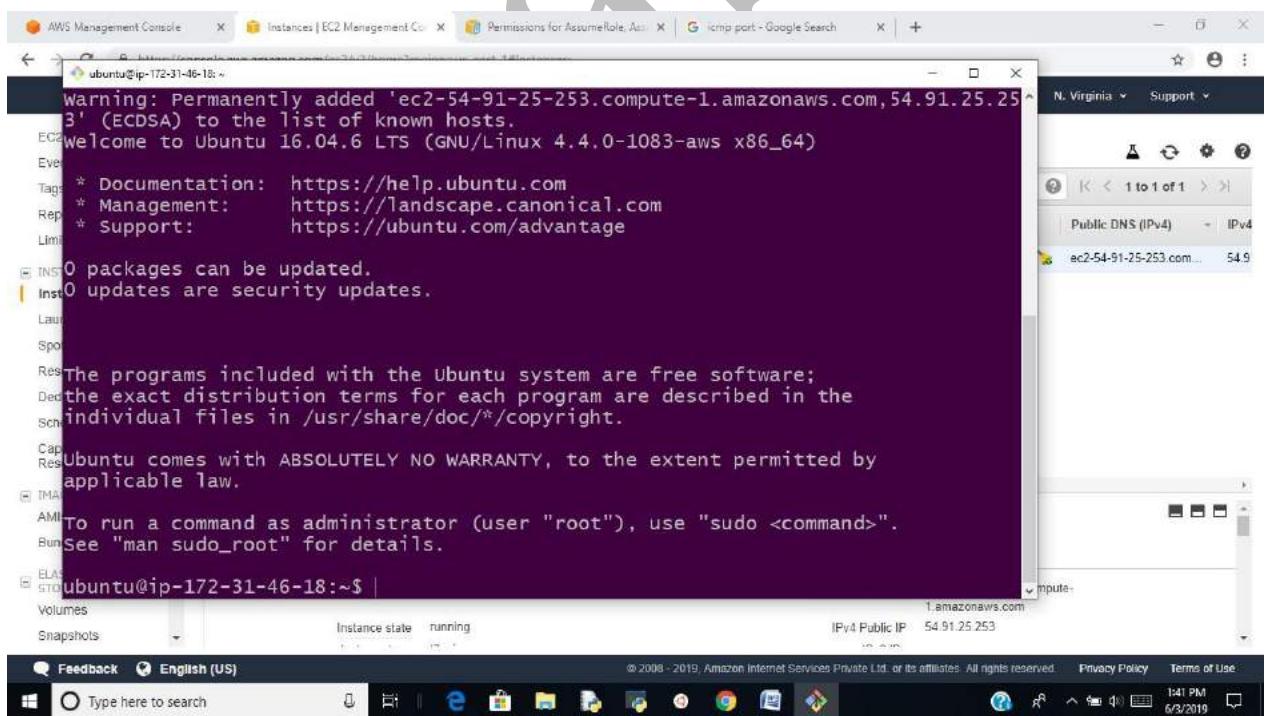
- Role name: enter some name for this role
- Role description: write some description for this role
- Click on create role



- For checking this role permissions we need to launch one ec2 and select this IAM Role when you launch this instance

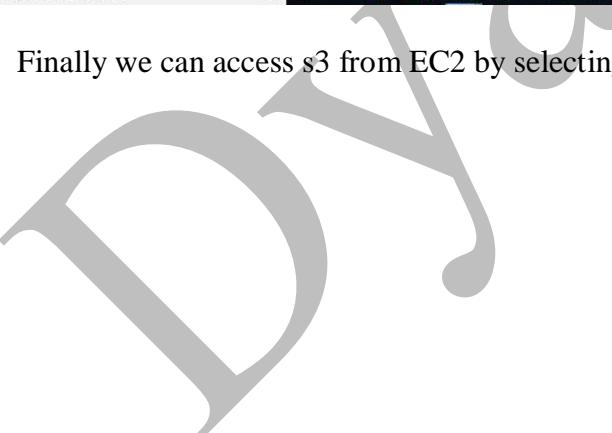


- Connected to that ec2 instance terminal by using git bash or putty



- Then we check s3 permissions is applied or not by seeing list of buckets in s3 from these EC2 instance

- First we install awscli on ec2 machine and check number of buckets



A screenshot of a Microsoft Word document window titled "Identity and Access management (Last saved by user) - Microsoft Word". The document contains a terminal session from an Ubuntu EC2 instance. The terminal output shows the installation of AWS CLI and lists several S3 buckets:

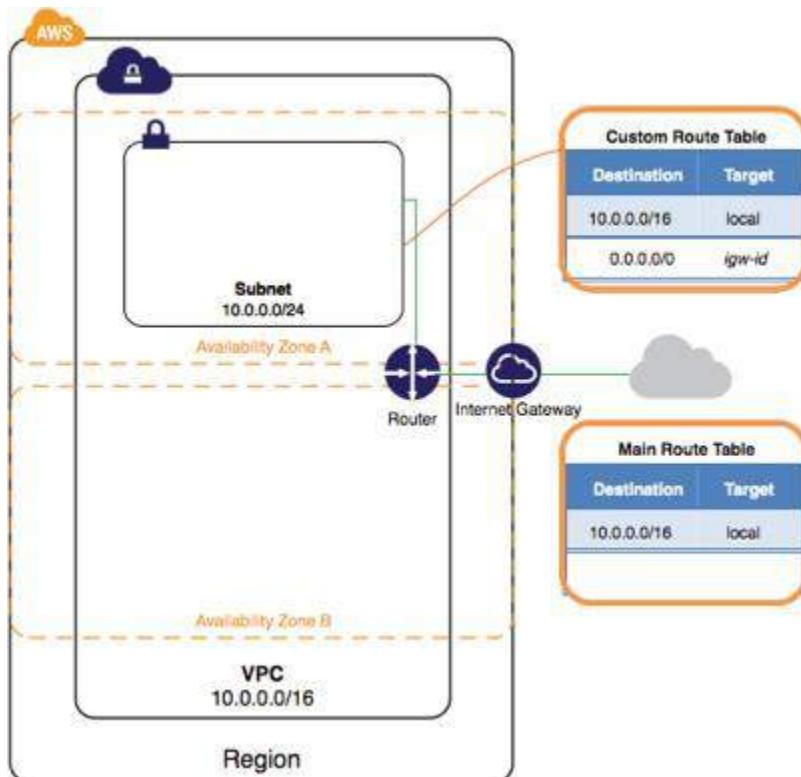
```
ubuntu@ip-172-31-46-18:~$ aws s3 ls
2019-05-09 05:41:54 bb156
2019-05-27 10:10:24 cf-templates-1smshrfby0vpk-us-east-1
2019-05-09 05:36:46 mybb321
ubuntu@ip-172-31-46-18:~$
```

- Finally we can access s3 from EC2 by selecting role

Amazon VPC

Amazon Virtual Private Cloud (Amazon VPC) enables you to launch AWS resources into a virtual network that you've defined. This virtual network closely resembles a traditional network that you'd operate in your own data center, with the benefits of using the scalable infrastructure of AWS.

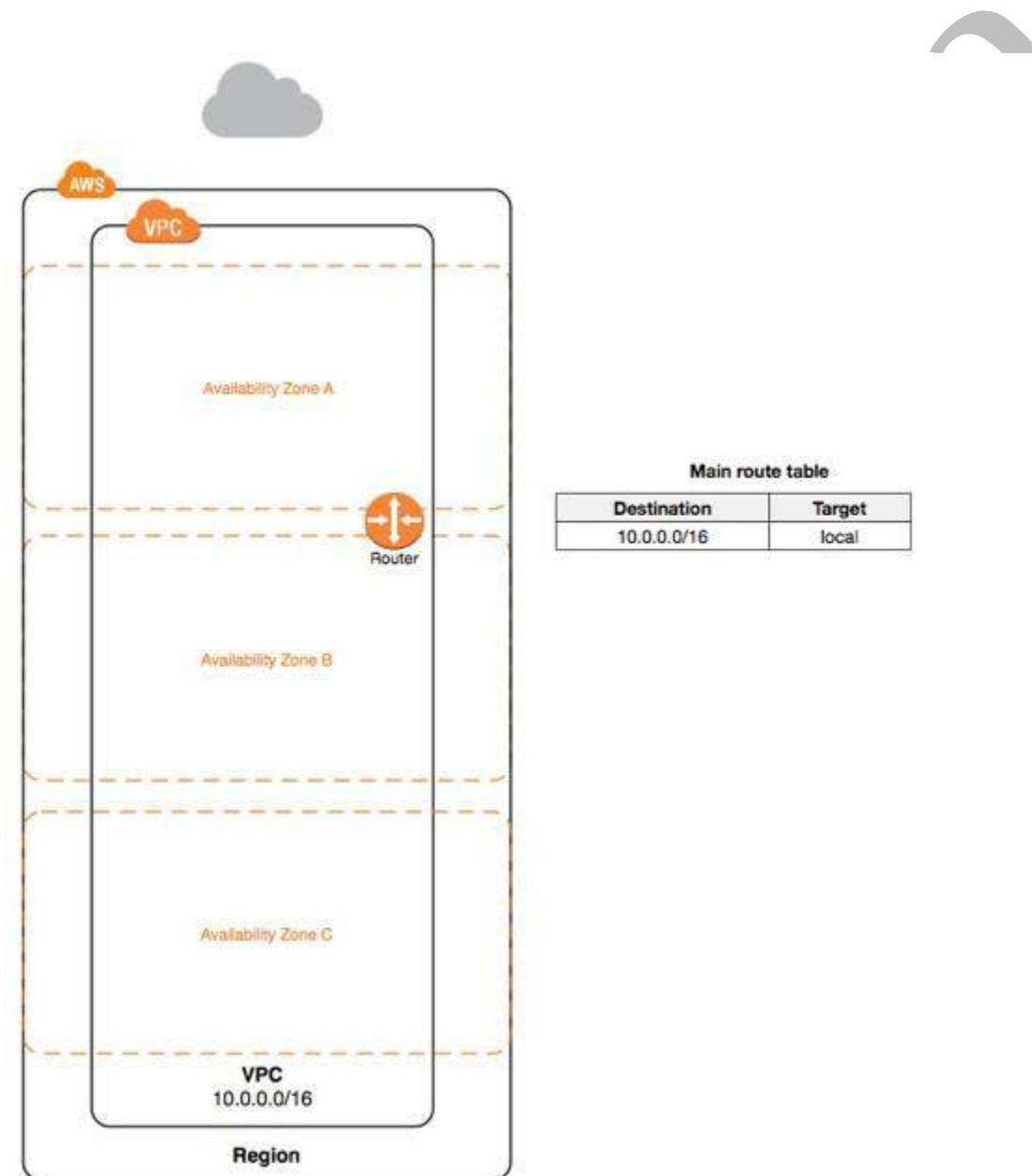
The following diagram represents the architecture of your VPC after you've completed this step.



VPCs and Subnets

A virtual private cloud (VPC) is a virtual network dedicated to your AWS account. It is logically isolated from other virtual networks in the AWS Cloud. You can launch your AWS resources, such as Amazon EC2 instances, into your VPC.

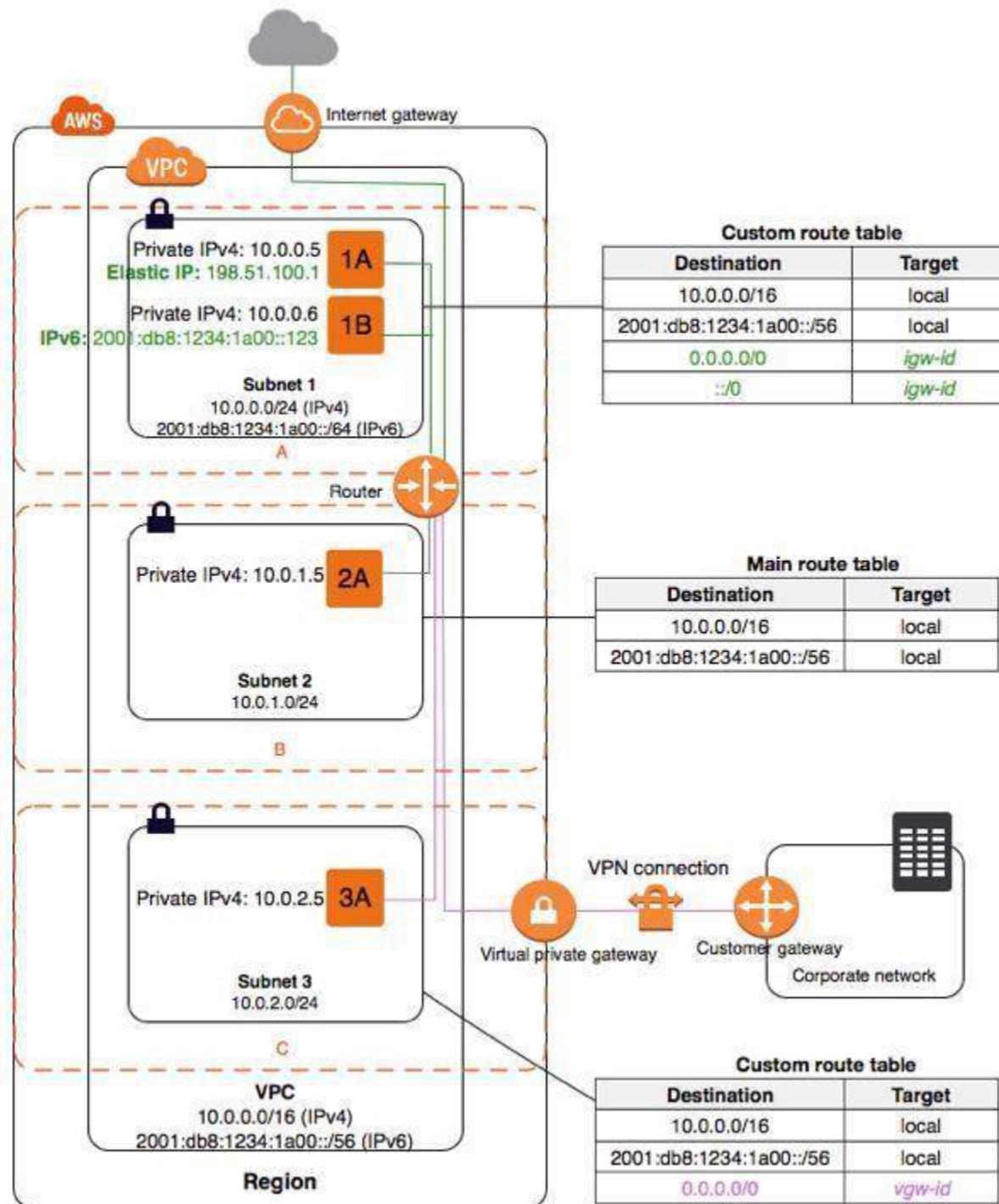
When you create a VPC, you must specify a range of IPv4 addresses for the VPC in the form of a Classless Inter-Domain Routing (CIDR) block; for example, 10.0.0.0/16. This is the primary CIDR block for your VPC.



Address: # 502, 5th Floor, Mahalaxmi Square Building, Opp to GTM Hospital, Andheri East Park Road,

A VPC spans all the Availability Zones in the region. After creating a VPC, you can add one or more subnets in each Availability Zone. When you create a subnet, you specify the CIDR block for the subnet, which is a subset of the VPC CIDR block. Each subnet must reside entirely within one Availability Zone and cannot span zones. Availability Zones are distinct locations that are engineered to be isolated from failures in other Availability Zones. By launching instances in separate Availability Zones, you can protect your applications from the failure of a single location. We assign a unique ID to each subnet.

The following diagram shows a VPC that has been configured with subnets in multiple Availability Zones. 1A, 1B, 2A, and 3A are instances in your VPC. An IPv6 CIDR block is associated with the VPC, and an IPv6 CIDR block is associated with subnet 1. An internet gateway enables communication over the internet, and a virtual private network (VPN) connection enables communication with your corporate network.



VPC and Subnet Sizing for IPv4

When you create a VPC, you must specify an IPv4 CIDR block for the VPC. The allowed block size is between a /16 netmask (65,536 IP addresses) and /28 netmask (16 IP addresses). After you've created your VPC, you can associate secondary CIDR blocks with the VPC.

When you create a VPC, we recommend that you specify a CIDR block (of /16 or smaller) from the private IPv4 address ranges as specified in [RFC 1918](#):

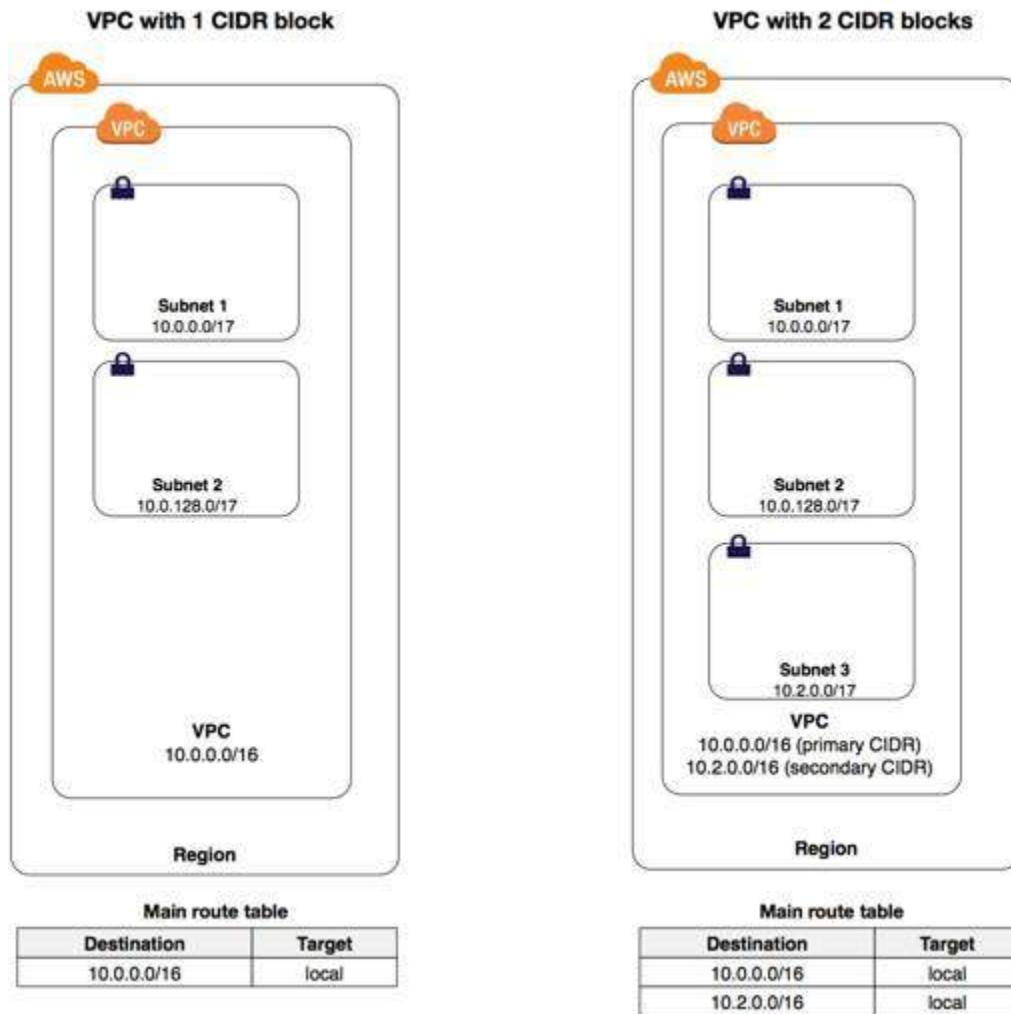
- 10.0.0.0 - 10.255.255.255 (10/8 prefix)
- 172.16.0.0 - 172.31.255.255 (172.16/12 prefix)
- 192.168.0.0 - 192.168.255.255 (192.168/16 prefix)

You can create a VPC with a publicly routable CIDR block that falls outside of the private IPv4 address ranges specified in RFC 1918; however, for the purposes of this documentation, we refer to *private IP addresses* as the IPv4 addresses that are within the CIDR range of your VPC.

Adding IPv4 CIDR Blocks to a VPC

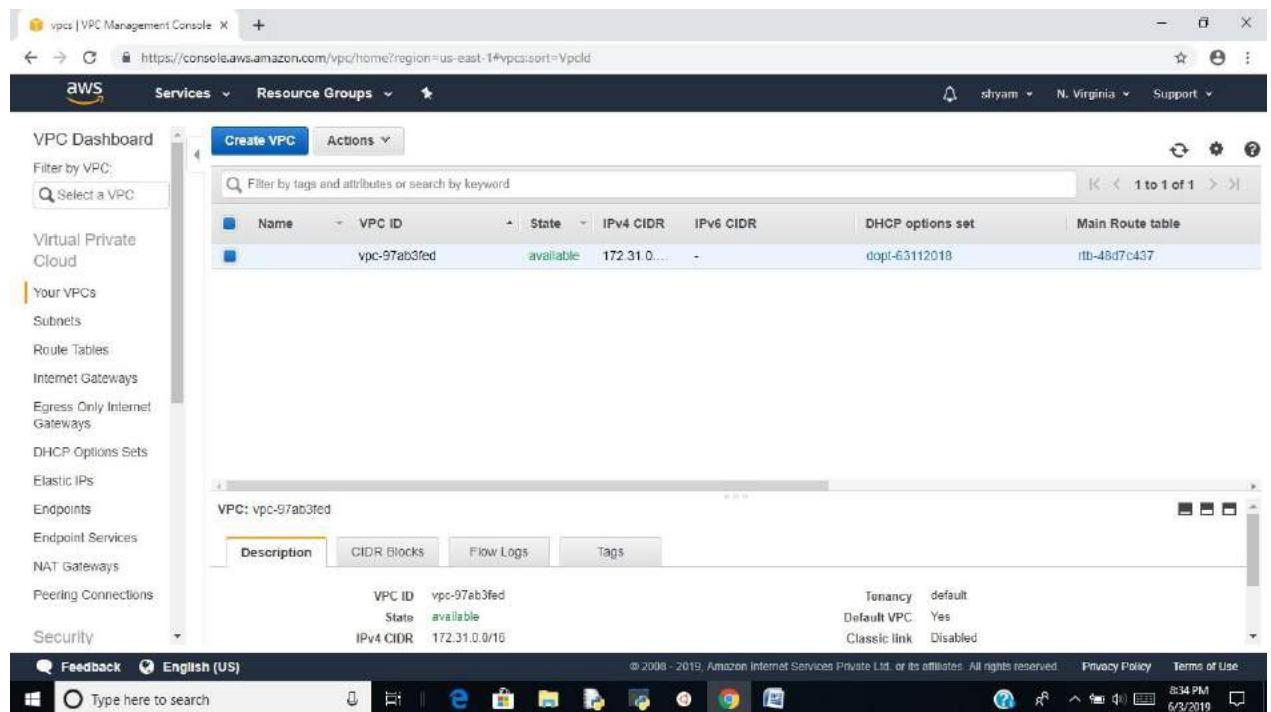
You can associate secondary IPv4 CIDR blocks with your VPC. When you associate a CIDR block with your VPC, a route is automatically added to your VPC route tables to enable routing within the VPC (the destination is the CIDR block and the target is `local`).

In the following example, the VPC on the left has a single CIDR block (10.0.0.0/16) and two subnets. The VPC on the right represents the architecture of the same VPC after you've added a second CIDR block (10.2.0.0/16) and created a new subnet from the range of the second CIDR.

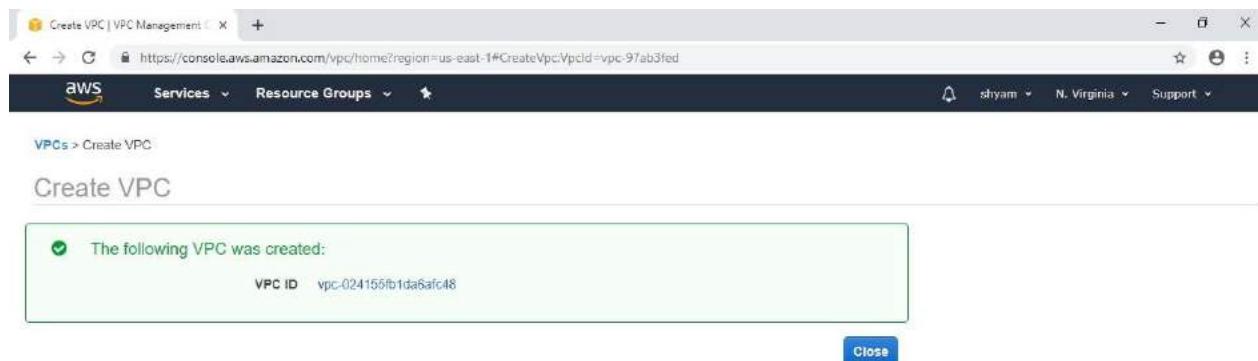


Create vpc

- Signin into aws account and select VPC section
- Click on create VPC



- Name tag: name for your VPC
- ipv4 cidr block: enter CIDR block
- CIDR block determines the vpc range that is how many IP address are allocated for this network
- Example: 10.0.0.0/24
- It is the IPV4 address block this format have total four digits each digit have 8 bits
- Final range is $4 \times 8 = 32$ that is $2^4 \times 2^8 = 2^{12} = 4096$ ipv4 addresses
- Based on subnet mask the range is allocated that is
- Total range - subnetmask = $32 - 24 = 8$ so $2^8 = 256$
- 256 ipv4 addresses are allocated for this VPC
- Ipv4 cidr block: select no ipv4 cidr block
- Tenancy: Default
- Click on create



- When the vpc is created then automatically Route Table, Network ACL and DHCP options set is created.

Create subnet

- click on subnets on leftside panel of your vpc section and click on create subnet

The screenshot shows the AWS VPC Management Console with the URL <https://console.aws.amazon.com/vpc/home?region=us-east-1#subnetssort=SubnetId>. The left sidebar lists various VPC-related services. The main content area shows a table of existing subnets:

Name	Subnet ID	State	VPC	IPv4 CIDR	Available IPv4	IPv6 CIDR	Availability Zone
subnet-09bcb443	available	vpc-97ab3fed	172.31.16.0/20	4091	-	-	us-east-1
subnet-10e9ac4c	available	vpc-97ab3fed	172.31.32.0/20	4091	-	-	us-east-1
subnet-1cf1697b	available	vpc-97ab3fed	172.31.0.0/20	4091	-	-	us-east-1
subnet-8df4b3a3	available	vpc-97ab3fed	172.31.80.0/20	4091	-	-	us-east-1
subnet-a3e36c9d	available	vpc-97ab3fed	172.31.64.0/20	4091	-	-	us-east-1
subnet-aa92a0a5	available	vpc-97ab3fed	172.31.48.0/20	4091	-	-	us-east-1

- Name tag: enter name for subnet
- VPC: select vpc in which vpc do want to create this subnet
- Availability Zone: select availability zone in which Availability Zone do you want to create subnet
- IPV4 CIDR Block: CIDR block represents the subnet range with in the vpc. The subnet CIDR Block is must less than VPC Range (30.0.0.0/25)
- Click on create

Amazon Web Services

The screenshot shows the 'Create subnet' wizard in the AWS VPC Management console. The 'Name tag' field contains 'mysubnet'. The 'VPC' dropdown is set to 'vpc-024155fb1da6afc48'. The 'Availability Zone' is 'us-east-1c'. The 'IPv4 CIDR block' is '30.0.0.0/24'. A table titled 'VPC CIDRs' shows one entry: '30.0.0.0/24' with status 'associated'. At the bottom, there are 'Cancel' and 'Create' buttons.



The screenshot shows the 'Create subnet' wizard with a green checkmark indicating success: 'The following Subnet was created:'. The 'Subnet ID' is 'subnet-0439e593df2e0f5ab'. There is a 'Close' button at the bottom right.



- When created the vpc it has the route table that route is vpc route table and Network ACL which is also coming from vpc
- We created number of subnets in a single VPC based VPC CIDR Block

Route Table

A *route table* contains a set of rules, called *routes*, that are used to determine where network traffic is directed.

Each subnet in your VPC must be associated with a route table; the table controls the routing for the subnet. A subnet can only be associated with one route table at a time, but you can associate multiple subnets with the same route table.

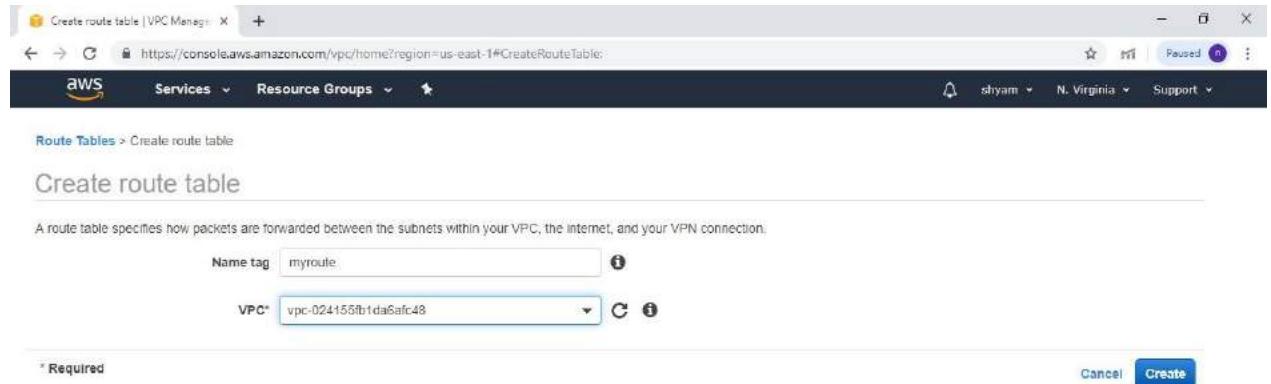
Create route Table

- Click on Route Table on left side panel your vpc section and click on create Route Table

Name	Route Table ID	Explicitly Associated with	Main	VPC ID	Owner
	rtb-06bf9f0dd1d859c32c	-	Yes	vpc-024155fb1da5afc48	814927698004
	rtb-48d7c437	-	Yes	vpc-97ab8fed	814927698004

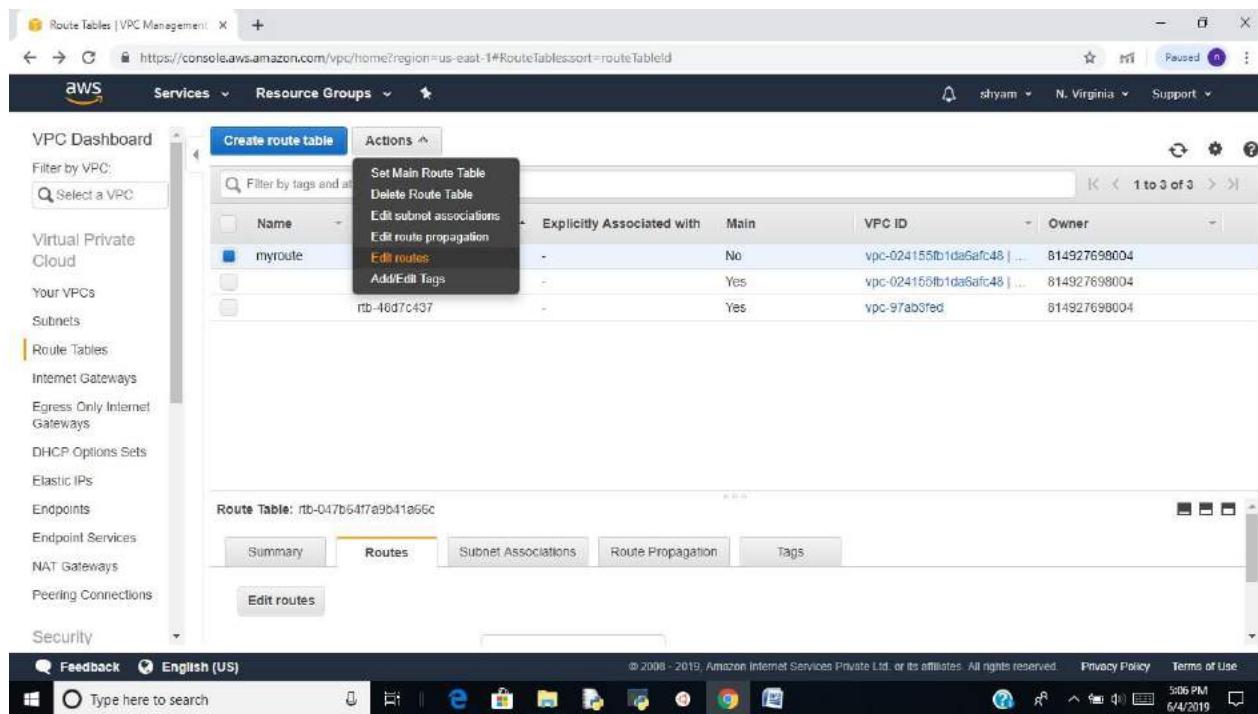
- Name Tag: name for Route Table
- VPC: select vpc in which vpc do you want to create the route table

- Click on create



Edit Routes

- Click on route table on left side panel vpc and select one route table
- Goto actions and select edit routes or click on routes on below panel and click on edit Routes



- Click on Add route: enter the CIDR block of subnet in destination field which subnet do you want to connected from this route table
- Target field: select type of target that is Internet Gateway or NAT Gateway or peering connection or instance it is the Network Interface
- Click on Save Routes

Destination	Target	Status	Propagated
30.0.0.0/24	local	active	No
1.0.0.0/32	eigwi		No

Add route Cancel Save routes



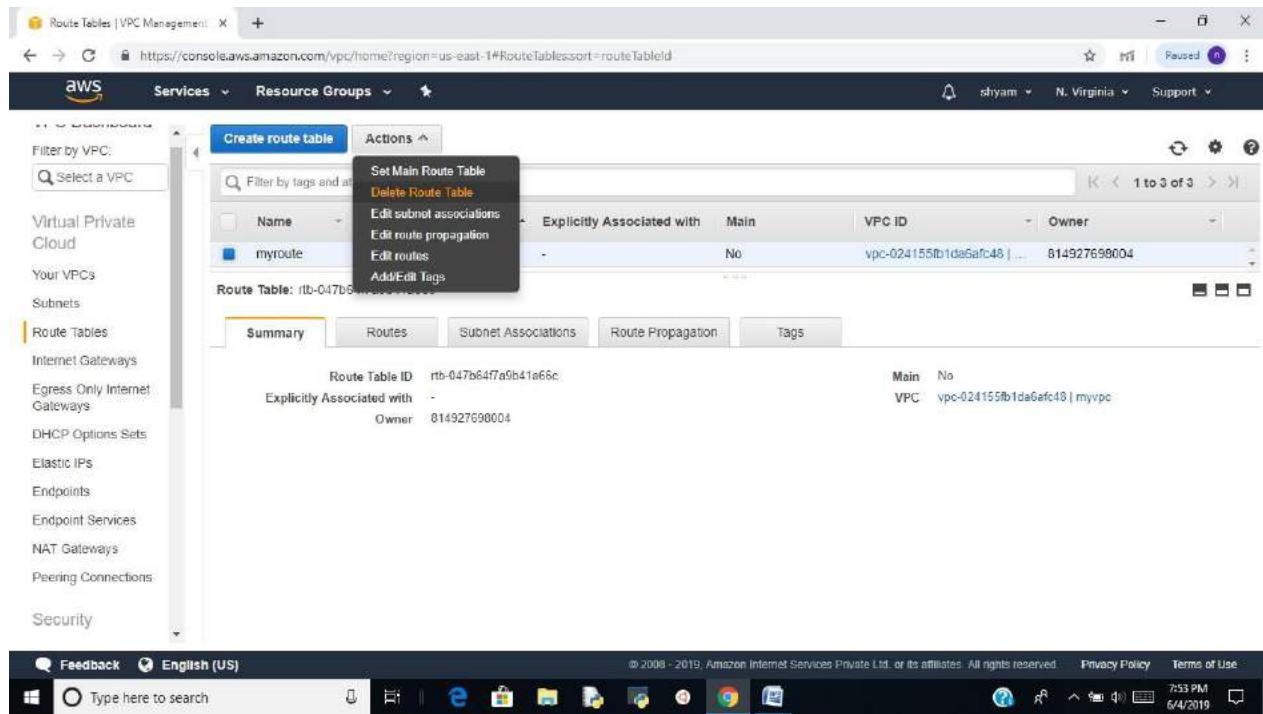
Subnet association

- This section is used to attach route table to particular subnet
- Click on Edit subnet association
- Select particular subnet and click on save

The screenshot shows the AWS VPC console with the URL <https://console.aws.amazon.com/vpc/home?region=us-east-1#EditRouteTableSubnetAssociations:routeTableId=rtb-047b64ff7a9b41a66c>. The page title is "Edit subnet associations". The route table selected is "rtb-047b64ff7a9b41a66c (myroute)". A table lists one associated subnet: "subnet-0439e593df2e8f5ab | mysubnet" with IPv4 CIDR "30.0.0.0/25" and Current Route Table "Main". The interface includes a search bar, pagination (1 to 1 of 1), and buttons for "Cancel" and "Save". The bottom of the screen shows a Windows taskbar with various icons.

Delete Route Table

- Click on route table on leftside panel of VPC section and goto actions and click delete Route Table



Note: if it is attached with any subnets then you can't delete this route table

Network ACL

- NACL is the Network Access Control List
- When you created vpc one NACL is created that is attached to subnet that is created in same VPC
- It is also called as network level security group
- All incoming services are first checked with NACL Inbound Rule if it is allow then only give access to this VPC
- All outcome services are checked with that VPC NACL outbound Rules
- Do you want block any ports or service from particular subnet or VPC then make deny in NACL inbound or outbound rules

Create NACL

- Click NACL on left side panel of VPC click on create Network ACL

- Name Tag: Name for NACL
- VPC: select vpc for NACL
- Click on create

Network ACLs > Create network ACL

Create network ACL

A network ACL is an optional layer of security that acts as a firewall for controlling traffic in and out of a subnet.

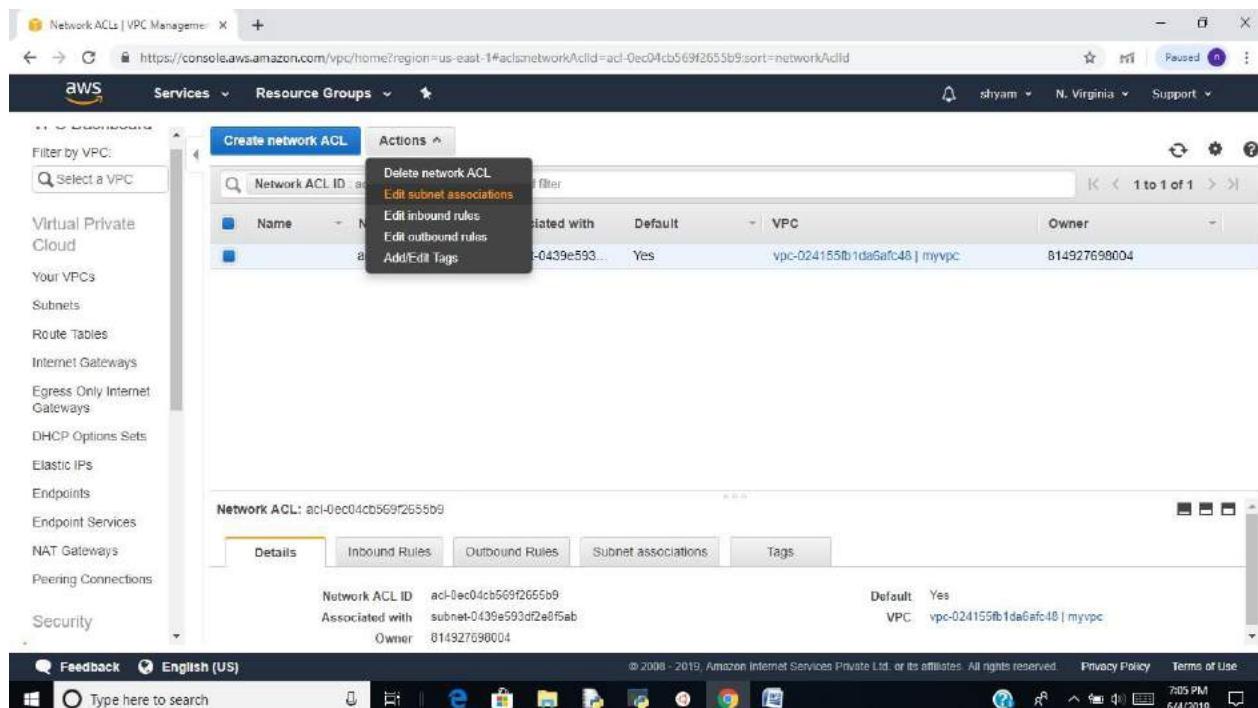
Name tag	mynacl	<small>i</small>
VPC*	vpc-024155fb1da6afc48	<small>C i</small>

* Required

© 2008 - 2019, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. Privacy Policy Terms of Use

Edit subnet association

- Click on NACL left side panel of VPC and goto actions and select Edit subnet association



- Select subnet and click on Edit

The screenshot shows the AWS VPC Network ACLs interface. The URL is https://console.aws.amazon.com/vpc/home?region=us-east-1#EditNetworkAclSubnetAssociations:networkAclId=acl-0ec04cb569f2655b9. The page title is "Edit subnet associations". A sub-header says "Network ACL ID: acl-0ec04cb569f2655b9". Below this, a table lists a single subnet association:

Subnet ID	IPv4 CIDR	IPv6 CIDR	Associated with
subnet-0439e593df2e8f5ab mysubnet	30.0.0.0/25	-	acl-0ec04cb569f2655b9

At the bottom right are "Cancel" and "Edit" buttons. The browser's address bar shows "Type here to search" and the system tray indicates it's 7:06 PM on 6/4/2019.

Add Inbound Rules

- Select NACL click on inbound rules in below panel of NACL and click on edit inbound rules

Network ACL ID: acl-0ec04cb569f2655b9

Name	Network ACL ID	Associated with	Default	VPC	Owner
acl-0ec04cb569f2655b9	acl-0ec04cb569f2655b9	subnet-0439e593...	Yes	vpc-024155fb1da6afc48 myvpc	814927698004

Network ACL: acl-0ec04cb569f2655b9

- Details
- Inbound Rules**
- Outbound Rules
- Subnet associations
- Tags

Edit inbound rules

View: All rules

Rule #	Type	Protocol	Port Range	Source	Allow / Deny
100	All Traffic	ALL	ALL	0.0.0.0/0	ALLOW

- Click on add rule
- Enter rule number and select protocol type which protocol do you want allow or deny and enter port range that is single protocol or range and enter source CIDE Block select ALLOW or DENY and click on save

Network ACL: acl-0ec04cb569f2655b9

Rule #	Type	Protocol	Port Range	Source	Allow / Deny
100	All Traffic	ALL	ALL	0.0.0.0/0	ALLOW
101	Custom TCP Rule	TCP (6)	8080	0.0.0.0/0	ALLOW

Add Rule

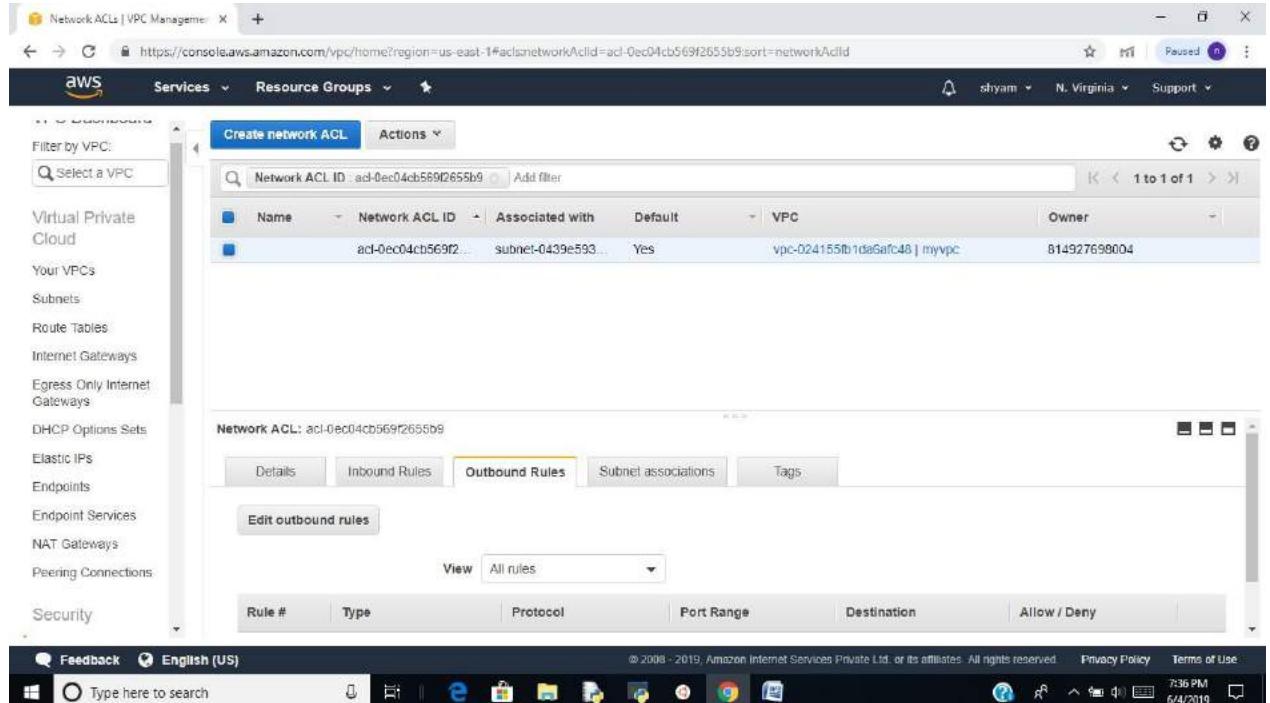
* Required

Cancel Save



Add Outbound Rules

- Click on VPC and select NACL and click outbound rules and click on Add outbound Rules



The screenshot shows the AWS VPC Network ACL Management interface. On the left, there's a sidebar with various VPC-related options like Virtual Private Cloud, Your VPCs, Subnets, Route Tables, Internet Gateways, Egress Only Internet Gateways, DHCP Options Sets, Elastic IPs, Endpoints, Endpoint Services, NAT Gateways, and Peering Connections. The main area has tabs for 'Create network ACL' and 'Actions'. Below that is a search bar and a table listing Network ACLs. One row is selected, showing details: Name (acl-0ec04cb569f2655b9), Network ACL ID (acl-0ec04cb569f2655b9), Associated with (subnet-0439e593...), Default (Yes), VPC (vpc-024155fb1da5afc48 | myvpc), and Owner (814927698004). At the bottom, there's a detailed view for 'Network ACL: acl-0ec04cb569f2655b9' with tabs for Details, Inbound Rules, Outbound Rules (which is selected), Subnet associations, and Tags. Under Outbound Rules, there's a button for 'Edit outbound rules' and a table with columns: Rule #, Type, Protocol, Port Range, Destination, and Allow / Deny. The table currently shows one row with 'All rules' selected. The status bar at the bottom indicates the browser version (IE 11) and the date/time (6/4/2019, 7:36 PM).

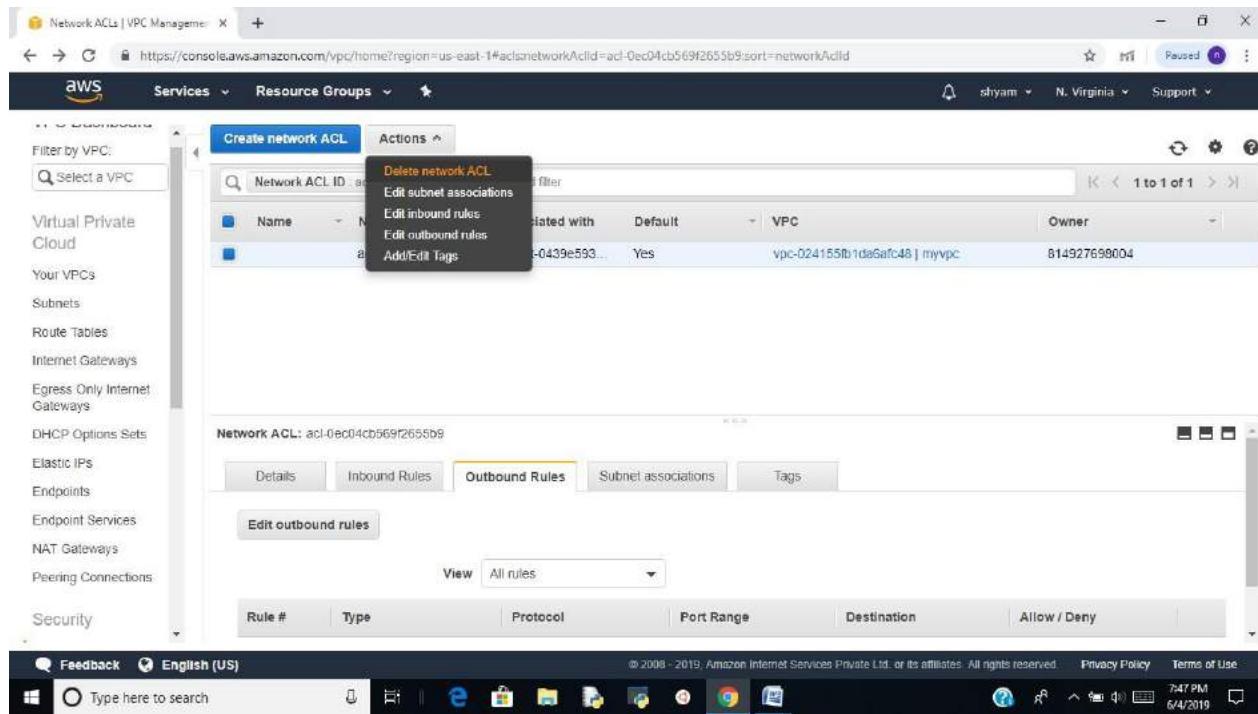
- Click on Edit outbound Rules and click on Add rule
- Enter rule number, protocol Type, port number or range, Desination CIDR Block and select ALLOW or DENY
- Click on save

The screenshot shows the AWS VPC Manager Network ACLs page. The URL is https://console.aws.amazon.com/vpc/home?region=us-east-1#EditOutboundRules?networkAclId=acl-0ec04cb569f2655b9. The page title is "Edit outbound rules". The main content is a table titled "Network ACL acl-0ec04cb569f2655b9" with columns: Rule #, Type, Protocol, Port Range, Destination, and Allow / Deny. There are two rules: Rule 100 (All Traffic, All, ALL, 0.0.0.0/0, ALLOW) and Rule 102 (Custom TCP Rule, TCP (6), 8080, 0.0.0.0/0, DENY). A large "Add Rule" button is visible. At the bottom are "Cancel" and "Save" buttons.



Delete NACL

- Select NACL which NACL do you want to delete and goto actions and click on delete NACL



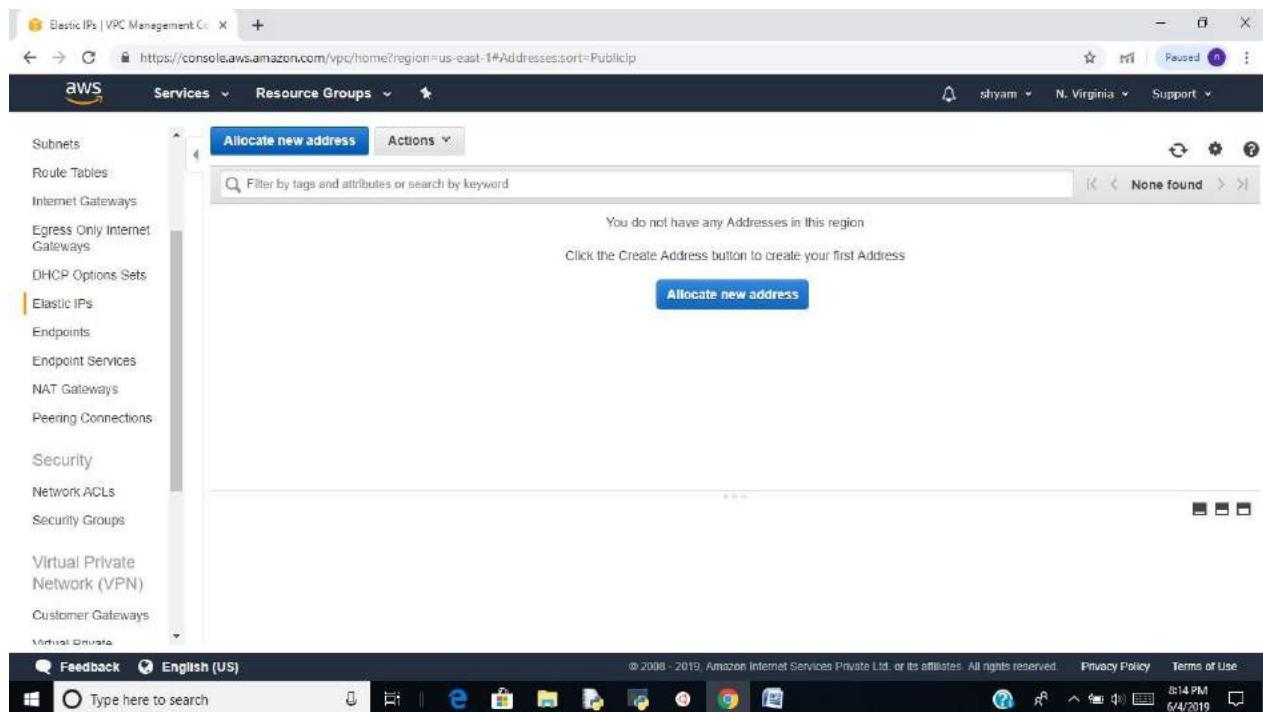
Note: if it is attached with any subnet you can't delete this NACL

ELASTIC IP'S

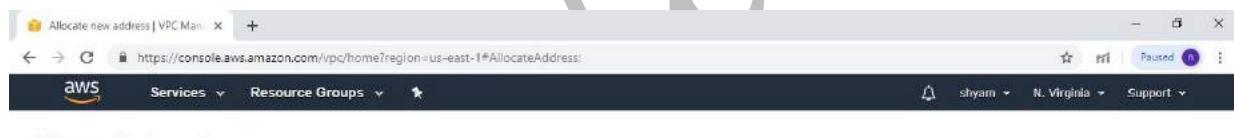
- Elastic IP address is the static IP address it is allocated to instances, NAT Gateways and Network Interfaces to identify and access this is not changed dynamically

Create Elastic IP

- Click on Elastic IPs on left side panel of vpc section and click on Allocate new address



- IPV4 address pool: select Amazon pool and click on allocate



Addresses > Allocate new address

Allocate new address

Allocate a new Elastic IP address by selecting the scope in which it will be used

Scope: VPC

IPv4 address pool:

- Amazon pool
- Owned by me

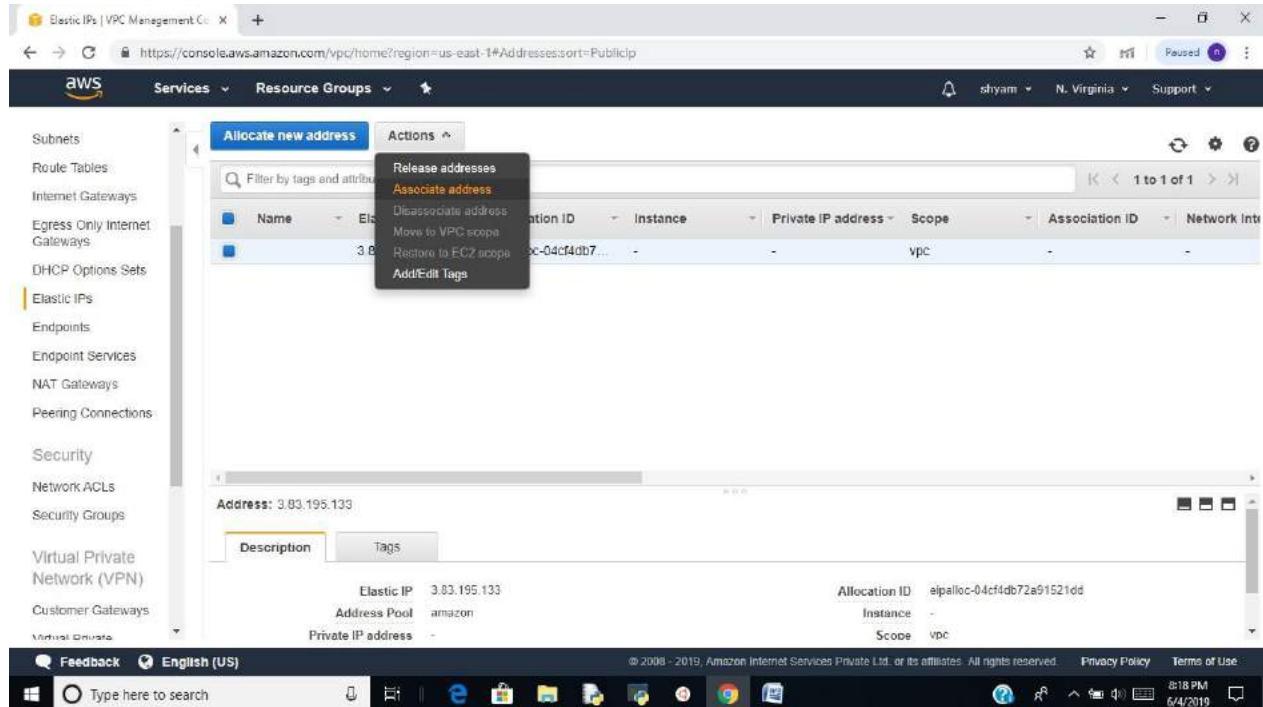
* Required

[Cancel](#) [Allocate](#)

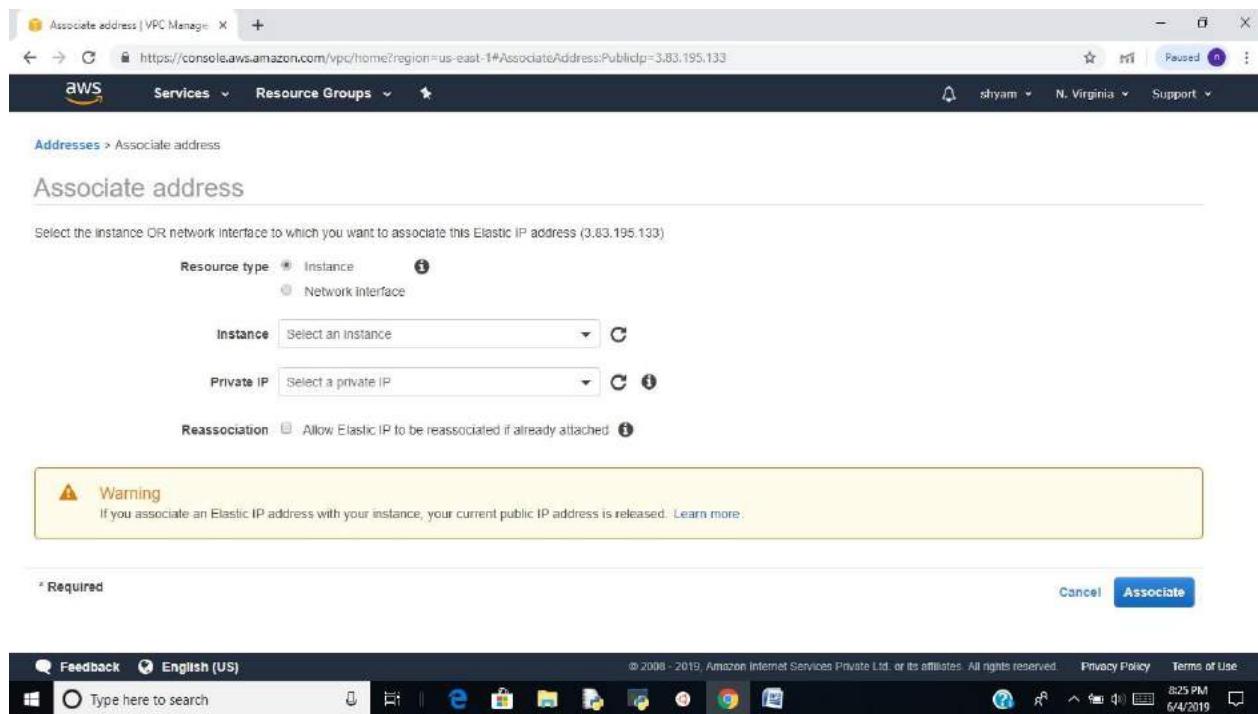


Associate address

- Select Elastic IP address and goto actions and click on associate address

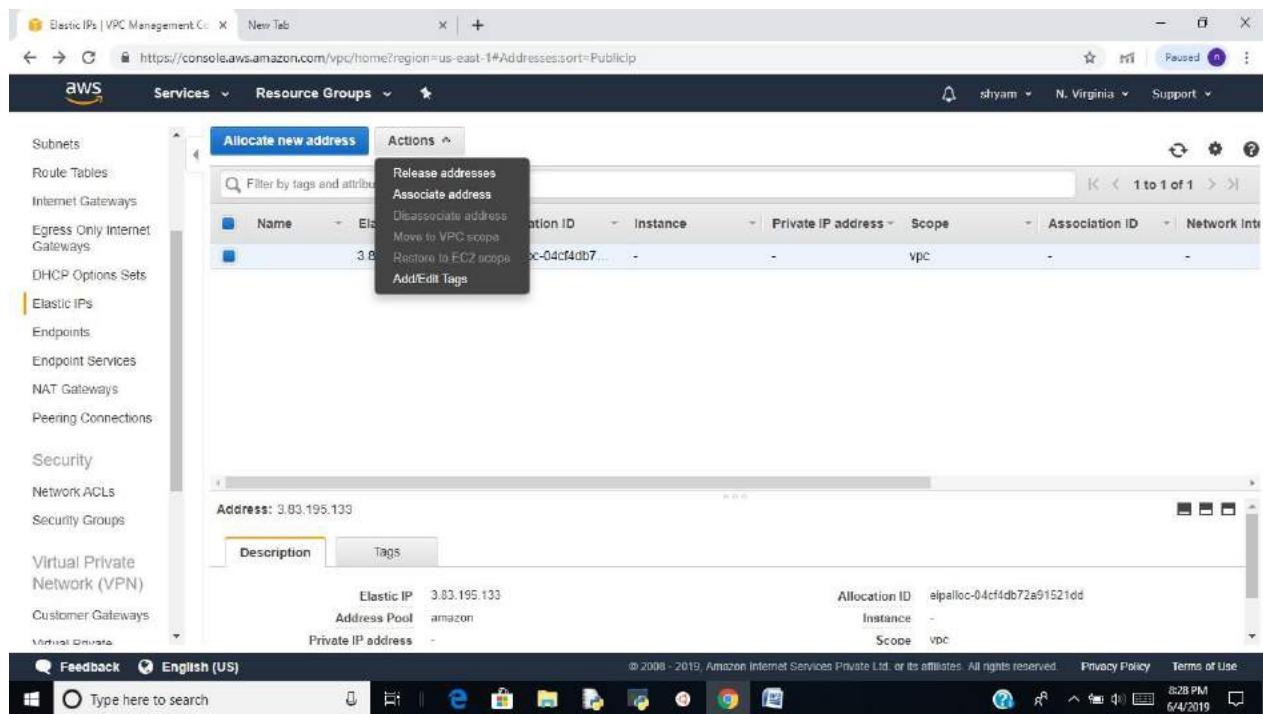


- Resource Type: select Instance or network interface to attach this Elastic IP (here I select instance)
- Instance: select instance id which instance do you want attach this Elastic IP if we choose network interface then select network interface id
- Private Ip: Elastic IP is the public address so you attach private ip to that instance or network interface
- Reassociation: allow Elastic IP to be reassigned if already attached
- Click on associate



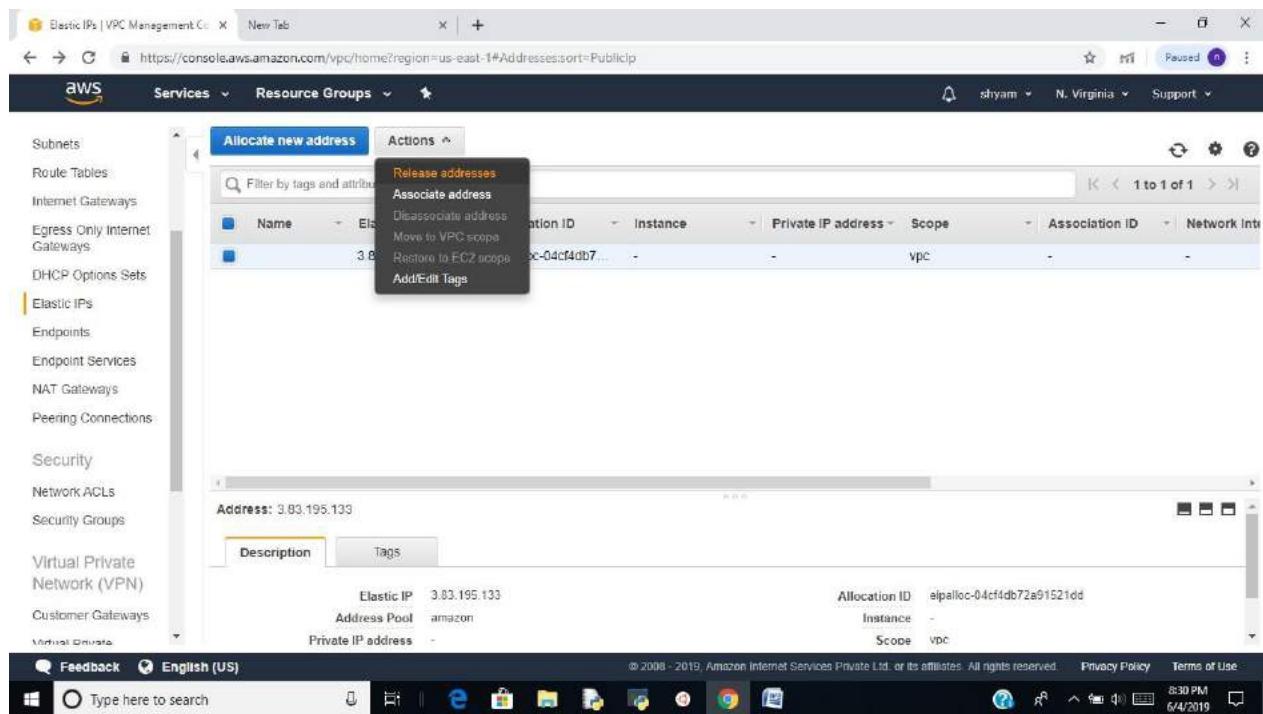
Diassociate address

- Do want detach or diassociate this Elastic IP then select Elastic IP and goto actions and click on Diassociate address

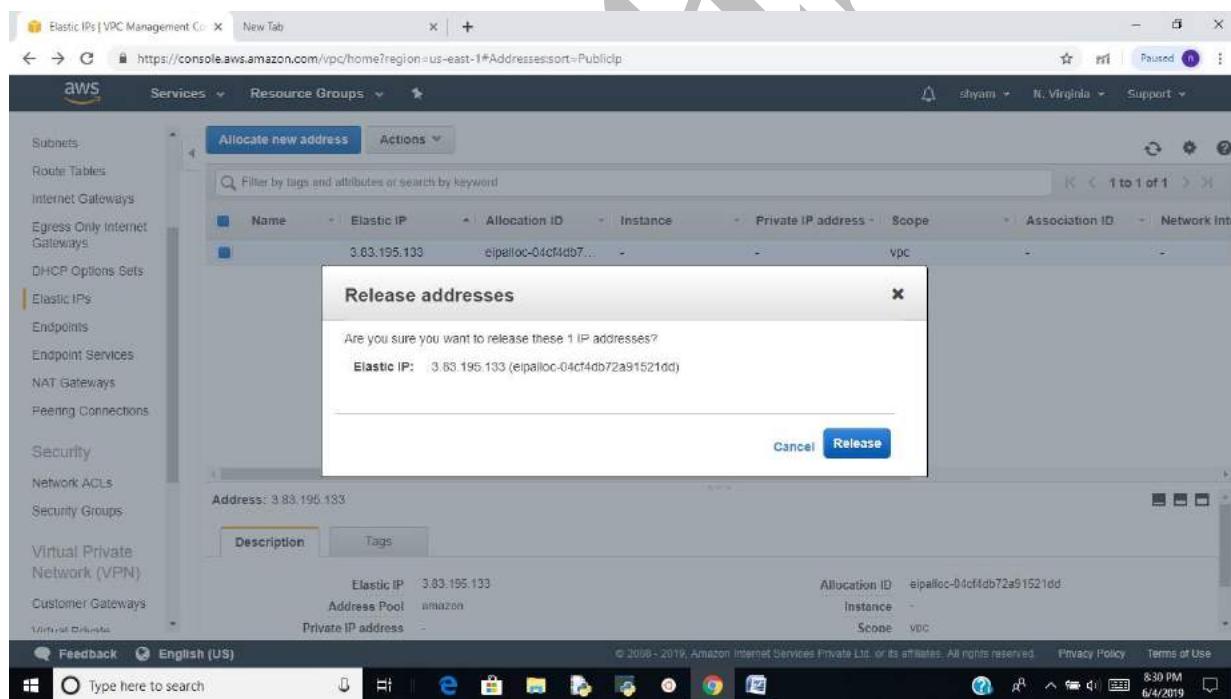


Delete Elastic IP

- Select Elastic IP and goto actions and select Release address



Click on release



Security Groups

- Security Group acts as a fire wall on instance level it is also called instance ACL
- It has set of inbound and outbound rules to access the instance

Create Security Group

- Click on security group on left side panel of VPC and click on create security group
- The Security Group is created within the vpc so do you want to attach any Security Group to that instance the instance must be in same VPC

The screenshot shows the AWS VPC Management Console. On the left, there's a sidebar with various VPC-related options like Subnets, Route Tables, Internet Gateways, etc., and 'Security Groups' is selected. The main area has a table titled 'Create security group' with columns: Name, Group ID, Group Name, VPC ID, Type, Description, and Owner. There are 14 rows listed, all starting with 'sg-' and ending with 'launch-wizard-' followed by a number from 2 to 10. The 'Actions' column contains a single link for each row.

Name	Group ID	Group Name	VPC ID	Type	Description	Owner
sg-003aa344ab32...	vpc-97ab3fed	EC2-VPC	launch-wizard-2 cr...	814927698004		
sg-03317f72b9eca...	vpc-97ab3fed	EC2-VPC	launch-wizard-6 cr...	814927698004		
sg-03e2667ecfe1...	vpc-97ab3fed	EC2-VPC	launch-wizard-12 ...	814927698004		
sg-043131605ead...	vpc-97ab3fed	EC2-VPC	launch-wizard-1 cr...	814927698004		
sg-0492ed76c8e2...	vpc-97ab3fed	EC2-VPC	launch-wizard-10 ...	814927698004		
sg-0611f37ad0d735...	vpc-97ab3fed	EC2-VPC	launch-wizard-4 cr...	814927698004		
sg-074924ea48a7...	vpc-024155fb1da6...	EC2-VPC	default VPC securi...	814927698004		
sg-0b981d7bc3e0...	vpc-97ab3fed	EC2-VPC	launch-wizard-7 cr...	814927698004		
sg-0c7asa308c32...	vpc-97ab3fed	EC2-VPC	launch-wizard-8 cr...	814927698004		
sg-0e49f1341npah4	vpc-97ab3fed	EC2-VPC	launch-wizard-5 cr...	814927698004		

- Security group Name: enter name for security group
- Description: enter some Description for Security Group
- VPC: select VPC for security group.
- Click on create

The screenshot shows the 'Create security group' page in the AWS VPC console. The URL is https://console.aws.amazon.com/vpc/home?region=us-east-1#CreateSecurityGroup. The form fields are as follows:

- Security group name*: mysg
- Description*: test sg
- VPC: vpc-024155fb1da6afc48

Below the form, there is a note: "A security group acts as a virtual firewall for your instance to control inbound and outbound traffic. To create a new security group fill in the fields below." At the bottom right are 'Cancel' and 'Create' buttons.



Edit Inbound Rules

- Select Security Group and goto Actions and click on Edit inbound rules

The screenshot shows the AWS VPC Management Console with the 'Security Groups' page. On the left, there's a sidebar with various network-related services. The main area displays a table of security groups, with one specific group selected ('sg-003aa344ab328dbb6'). A context menu is open over this group, showing options such as 'Delete security group' and 'Edit inbound rules'. Below the table, there are tabs for 'Description', 'Inbound Rules', 'Outbound Rules', and 'Tags'. At the bottom of the page, there are details for the selected group: Group ID (sg-003aa344ab328dbb6), VPC ID (vpc-97ab3fed), Group Name (launch-wizard-2), and Description (launch-wizard-2 created 2019-05-03T12:05:15.997+05:30). The browser status bar at the bottom indicates it's from June 4, 2019, at 8:46 PM.

- Click on Add Rule
- Select protocol, port number or range and source (select any where for global access) and click on Save rules

The screenshot shows the AWS VPC Management Console with the URL <https://console.aws.amazon.com/vpc/home?region=us-east-1#ModifyInboundSecurityGroupRules?groupId=sg-003aa344ab328dbb6>. The page title is "Edit inbound rules". It displays three existing inbound rules:

Type	Protocol	Port Range	Source	Description
All traffic	All	All	Custom 0.0.0.0/0	e.g. SSH for Admin Desktop
All traffic	All	All	Custom ::/0	e.g. SSH for Admin Desktop
SSH	TCP	22	Custom 0.0.0.0/0	e.g. SSH for Admin Desktop

A "Add Rule" button is visible at the bottom left. A note at the bottom center states: "NOTE: Any edits made on existing rules will result in the edited rule being deleted and a new rule created with the new details. This will cause traffic that depends on that rule to be dropped for a very brief period of time until the new rule can be created." At the bottom right are "Cancel" and "Save rules" buttons.



Edit outbound Rules

- Select security group and goto actions and click on outbound rules
- Click on Add Rule
- Enter protocol, port and Destination addresses and click Save Rules

Outbound rules control the outgoing traffic that's allowed to leave the instance.

Type	Protocol	Port Range	Destination	Description
All traffic	All	All	Custom 0.0.0.0/0	e.g. SSH for Admin Desktop
Custom TCP Rule	TCP	8080	Anywhere 0.0.0.0/0	e.g. SSH for Admin Desktop

NOTE: Any edits made on existing rules will result in the edited rule being deleted and a new rule created with the new details. This will cause traffic that depends on that rule to be dropped for a very brief period of time until the new rule can be created.

* Required

Add Rule Cancel Save rules



- Note:** in inbound rules by default ssh protocol with 22 port is enabled and outbound rules all traffic is enabled

Delete Security Group

- Select Security Group and goto actions and click on Delete security group

The screenshot shows the AWS VPC Manager Security Groups page. On the left, there's a sidebar with links to Subnets, Route Tables, Internet Gateways, Egress Only Internet Gateways, DHCP Options Sets, Elastic IPs, Endpoints, Endpoint Services, NAT Gateways, Peering Connections, Security, Network ACLs, and Security Groups. The Security Groups link is highlighted. The main area displays a table of security groups with columns for Name, Group ID, Group Name, VPC ID, Type, Description, and Owner. One row is selected, showing details below the table: Group ID sg-003aa344ab328dbb6, VPC ID vpc-97ab3fed, Owner 814927698004, Inbound rule count 3, Outbound rule count 1. A context menu is open over the selected row, with 'Delete security group' highlighted.

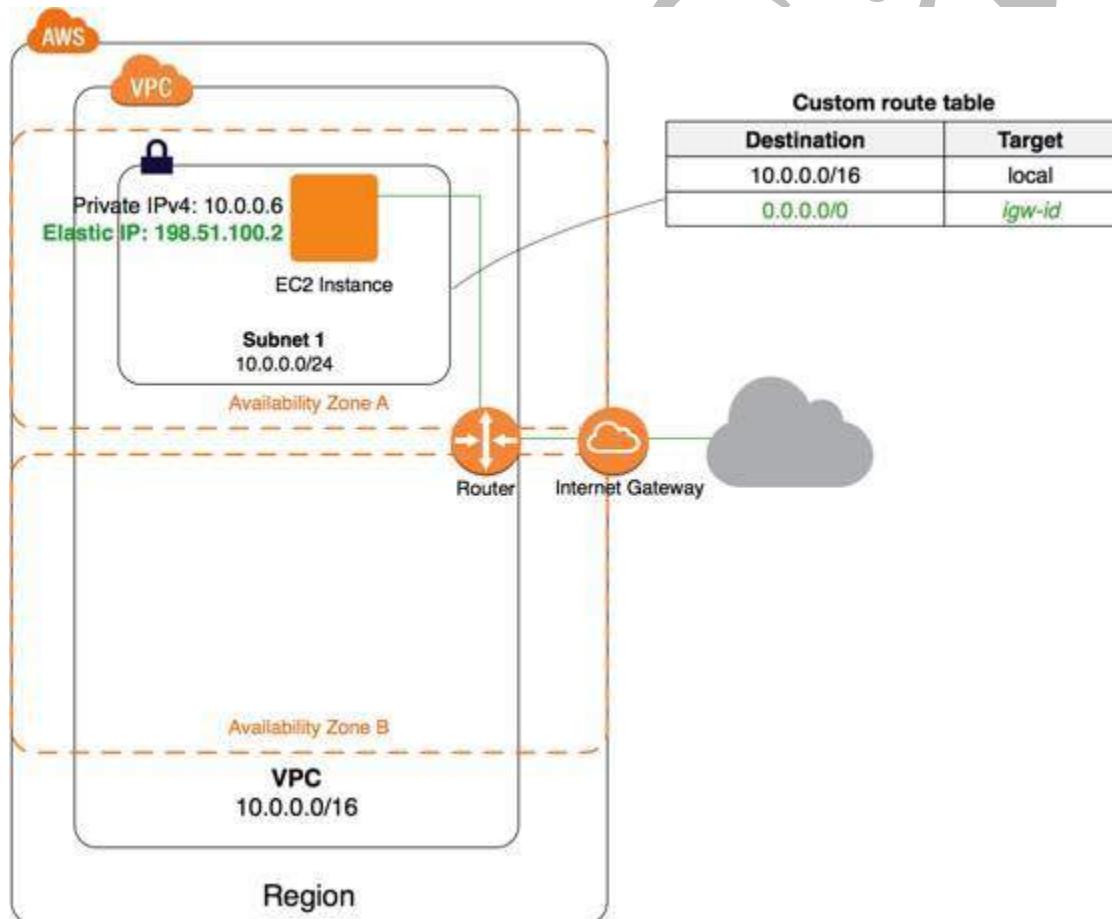
- Click on Delete security group

This screenshot shows the same AWS VPC Manager Security Groups page as the previous one, but with a modal dialog box in the center. The dialog is titled 'Delete security group' and contains the message 'Are you sure that you want to delete this security group (sg-003aa344ab328dbb6)?'. It has two buttons at the bottom: 'Cancel' and 'Delete security group'. The background table of security groups is visible but dimmed.

Internet Gateways

An internet gateway is a horizontally scaled, redundant, and highly available VPC component that allows communication between instances in your VPC and the internet. It therefore imposes no availability risks or bandwidth constraints on your network traffic.

An internet gateway serves two purposes: to provide a target in your VPC route tables for internet-routable traffic, and to perform network address translation (NAT) for instances that have been assigned public IPv4 addresses.

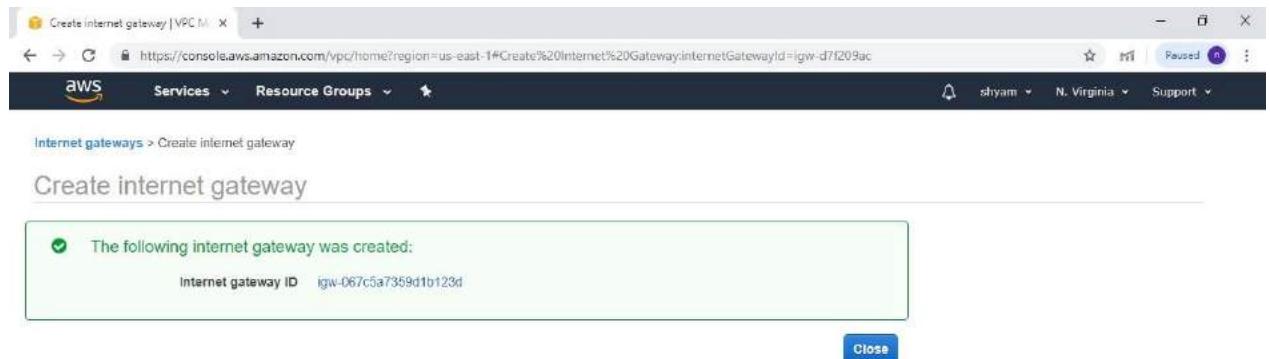


Creating Internet Gateway

- Select VPC section and click on Internet Gateways on left side panel and click on Create internet gateway

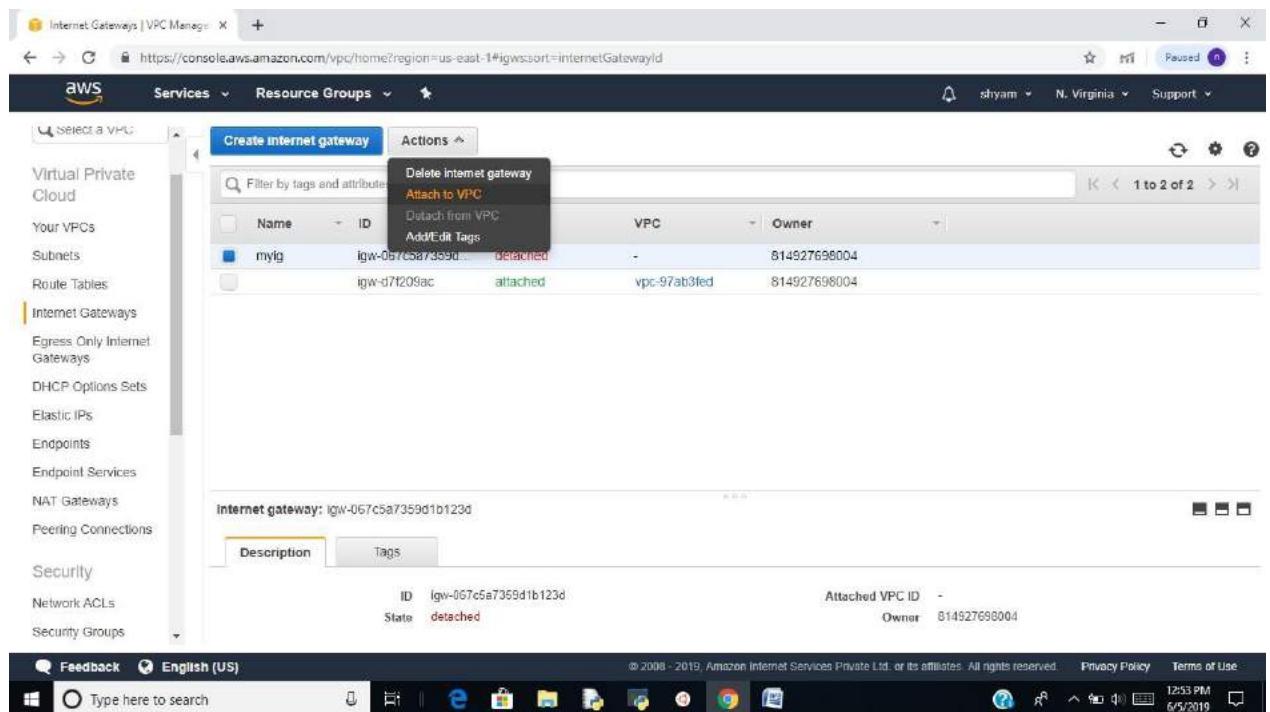
- Name tag: Enter some name for internet gateway and click on create

- Click on close

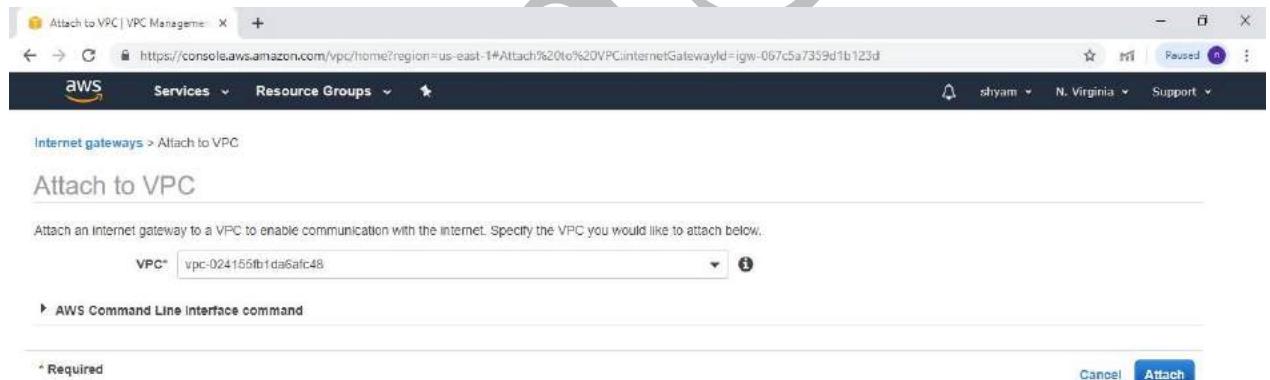


Attach to VPC

- Select Internet Gateway and goto actions and click on attach to VPC



- Select vpc id and click on attach



Add internet gateway to subnet

- Select subnet in which subnet do you want to attach internet gateway and click on Route table on below panel of subnet and click on Route table id

Subnets | VPC Management Console

https://console.aws.amazon.com/vpc/home?region=us-east-1#subnetssort=SubnetId

Services Resource Groups

Virtual Private Cloud

Your VPCs

Subnets

Route Tables

Internet Gateways

Egress Only Internet Gateways

DHCP Options Sets

Elastic IPs

Endpoints

Endpoint Services

NAT Gateways

Peering Connections

Security

Network ACLs

Security Groups

Create subnet Actions

Filter by tags and attributes or search by keyword

Name	Subnet ID	State	VPC	IPv4 CIDR	Available IPv4	IPv6 CIDR	Available IPv6
mysubnet	subnet-0439e593df2e0f5ab	available	vpc-024155fb1da6af48	30.0.0.0/25	123	-	-

Subnet: subnet-0439e593df2e0f5ab

Description Flow Logs Route Table Network ACL Tags Sharing

Edit route table association

Route Table: rtb-06bf9f0df0d859c32c

Destination Target

30.0.0.0/24 local

https://console.aws.amazon.com/vpc/home?region=us-east-1#routetableslist=rtb-06bf9f0df0d859c32c

© 2006 - 2019, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. Privacy Policy Terms of Use

- Click on Routes on below panel and click on Edit Routes

Route Tables | VPC Management

https://console.aws.amazon.com/vpc/home?region=us-east-1#RouteTablessearch=rtb-06bf9f0df0d859c32c;sort=routeTableId

Services Resource Groups

Virtual Private Cloud

Your VPCs

Subnets

Route Tables

Internet Gateways

Egress Only Internet Gateways

DHCP Options Sets

Elastic IPs

Endpoints

Endpoint Services

NAT Gateways

Peering Connections

Security

Network ACLs

Security Groups

Create route table Actions

search rtb-06bf9f0df0d859c32c Add filter

Name	Route Table ID	Explicitly Associated with	Main	VPC ID	Owner
	rtb-06bf9f0df0d859c32c	-	Yes	vpc-024155fb1da6af48 ...	814927698004

Route Table: rtb-06bf9f0df0d859c32c

Summary Routes Subnet Associations Route Propagation Tags

Edit routes

Feedback English (US)

https://dyasaatech.com

© 2006 - 2019, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. Privacy Policy Terms of Use

- Click on Add route
- Destination: enter global CID (0.0.0.0/0)
- Target: select previously created Internet Gateway
- Click on save Routes

Route Tables > Edit routes

Destination	Target	Status	Propagated
30.0.0.0/24	local	active	No
0.0.0.0/0	igw-067c5a7359d1b123d	active	No

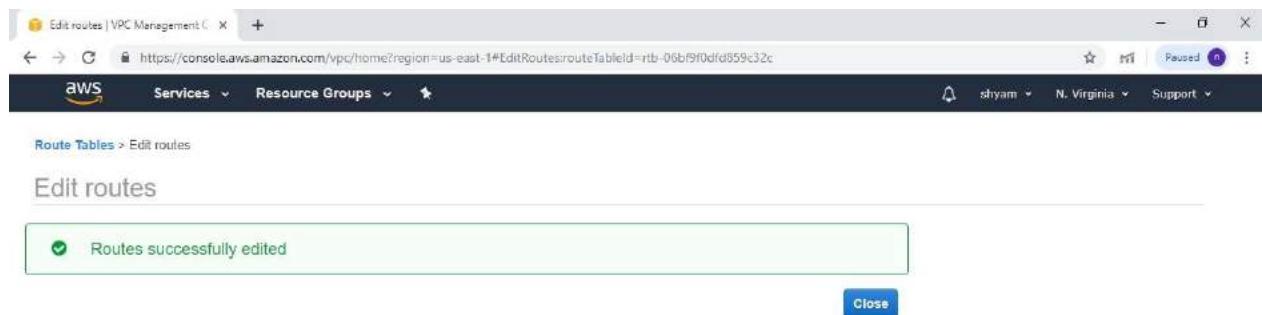
Add route

* Required

Cancel Save routes



- Click on close



- The Instances in that subnet are connected to public using Internet Gateway

Detach Internet Gateway

- Select Internet Gateway and goto actions and click on Detach from vpc

Amazon Web Services

The screenshot shows the AWS VPC Manager interface. On the left, a sidebar lists various VPC components: Virtual Private Cloud, Your VPCs, Subnets, Route Tables, Internet Gateways (selected), Egress Only Internet Gateways, DHCP Options Sets, Elastic IPs, Endpoints, Endpoint Services, NAT Gateways, Peering Connections, Security, Network ACLs, and Security Groups. The main pane displays a table of Internet Gateways. One row for 'myig' is selected, and a context menu is open over it, showing options like 'Delete internet gateway', 'Attach to VPC', and 'Detach from VPC'. The 'Detach from VPC' option is highlighted with a yellow background. Below the table, a detailed view of the selected 'myig' internet gateway is shown, including its ID (igw-067c5a7359d1b123d), state (attached), Attached VPC ID (vpc-024155fb1da6afc48 | myvpc), and Owner (814927698004). The bottom of the screen shows the Windows taskbar.

- Click on detach

The screenshot shows the AWS VPC Manager interface again. A confirmation dialog box titled 'Detach from VPC' is centered on the screen. It asks, 'Are you sure that you want to detach this internet gateway (igw-067c5a7359d1b123d | myig) from this VPC (vpc-024155fb1da6afc48 | myvpc)?'. There are 'Cancel' and 'Detach' buttons at the bottom. The background shows the same VPC Manager interface with the 'myig' internet gateway listed in the table.

Delete Internet Gateway

- Select Internet Gateway and goto actions and select Delete internet gateway

The screenshot shows the AWS VPC Management Console with the 'Internet Gateways' section selected. A context menu is open over the first internet gateway listed, which is named 'myig'. The menu options include 'Delete internet gateway' (which is highlighted), 'Attach to VPC', 'Detach from VPC', and 'Add/Edit Tags'. Below the menu, a table lists two internet gateways. The first one, 'myig', is detached from a VPC. The second one, 'igw-d7f209ac', is attached to a VPC with ID 'vpc-97ab3fed'. The table includes columns for Name, ID, State, VPC, and Owner.

Name	ID	VPC	Owner
myig	igw-067c5a7359d1b123d	-	814927698004
igw-d7f209ac		attached	vpc-97ab3fed

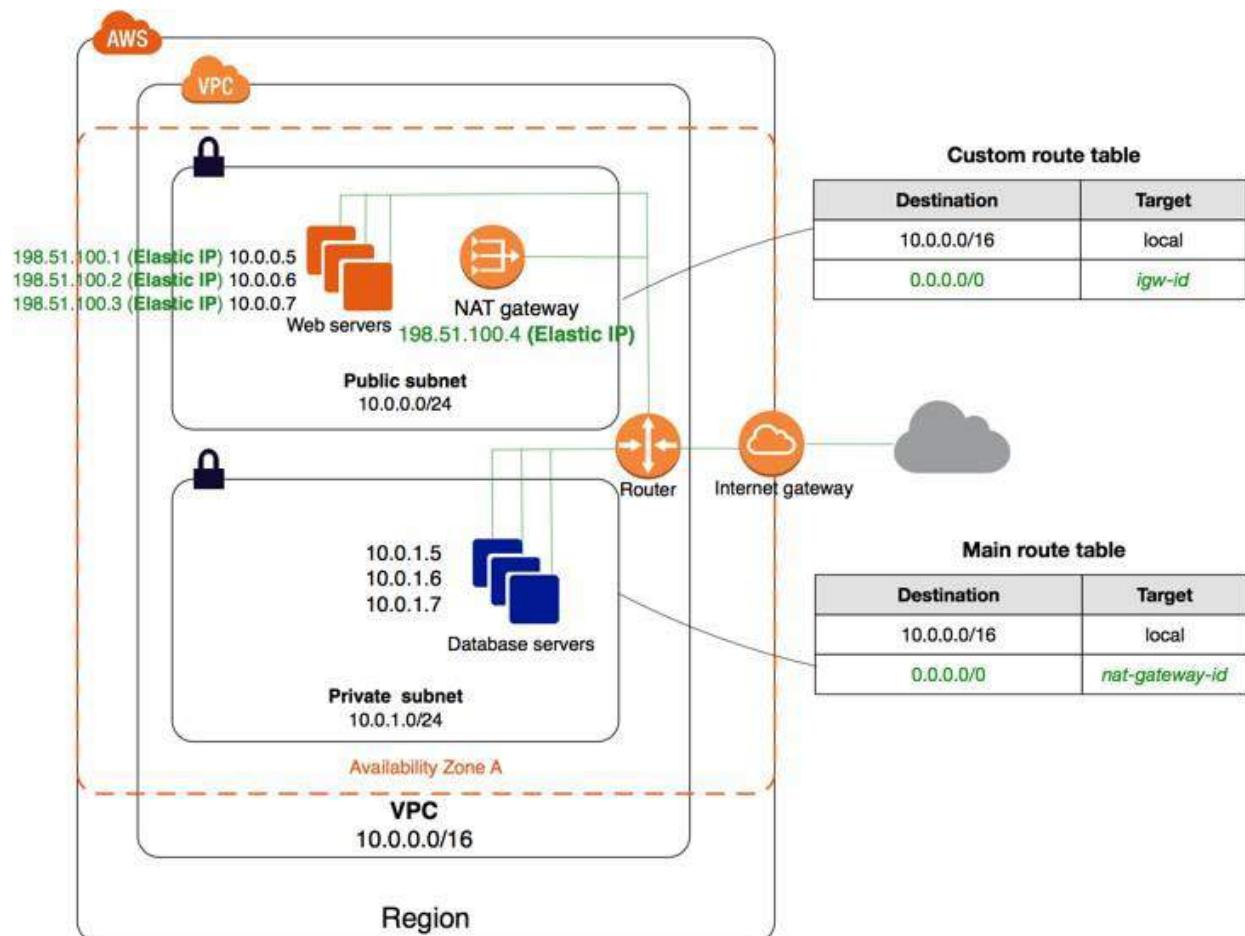
Click on delete

The screenshot shows the AWS VPC Management Console with the 'Internet Gateways' section selected. A confirmation dialog box titled 'Delete internet gateway' is displayed, asking 'Are you sure that you want to delete this internet gateway (igw-067c5a7359d1b123d | myig)?'. The dialog has 'Cancel' and 'Delete' buttons. In the background, the internet gateway table shows the 'myig' entry with its details: ID 'igw-067c5a7359d1b123d', State 'detached', Attached VPC ID 'vpc-97ab3fed', and Owner '814927698004'.

Nat gateways

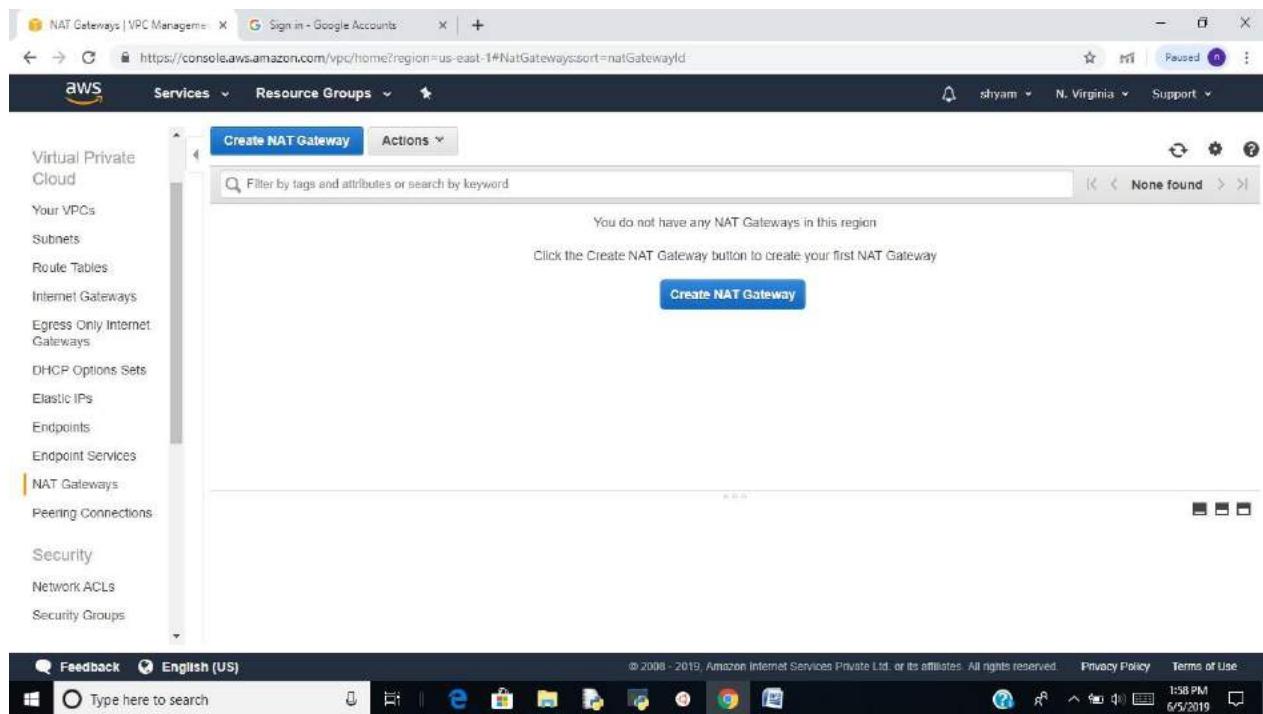
To create a NAT gateway, you must specify the public subnet in which the NAT gateway should reside. For more information about public and private subnets, see [Subnet Routing](#). You must also specify an [Elastic IP address](#) to associate with the NAT gateway when you create it. The Elastic IP address cannot be changed once you associate it with the NAT Gateway. 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.

Each NAT gateway is created in a specific Availability Zone and implemented with redundancy in that zone. You have a limit on the number of NAT gateways you can create in an Availability Zone

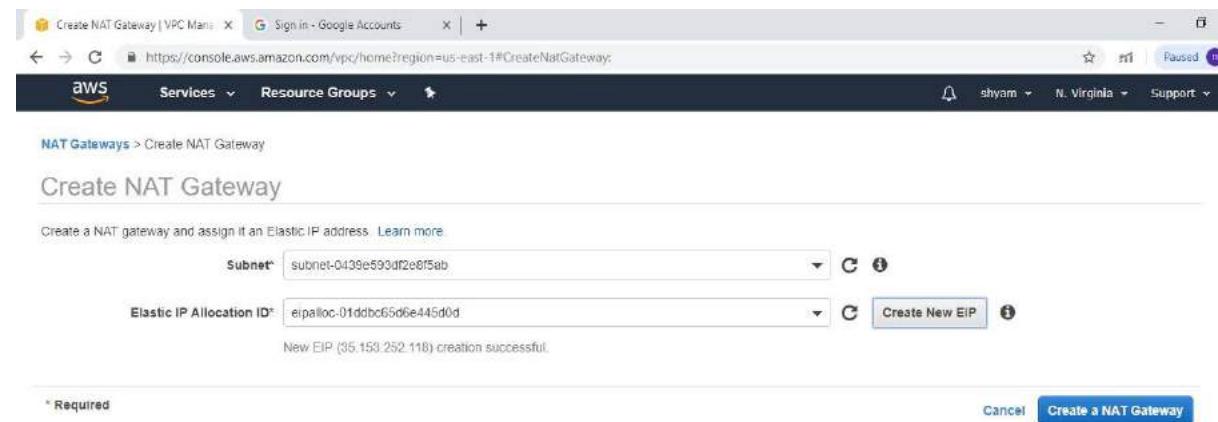


Creating NAT Gateway

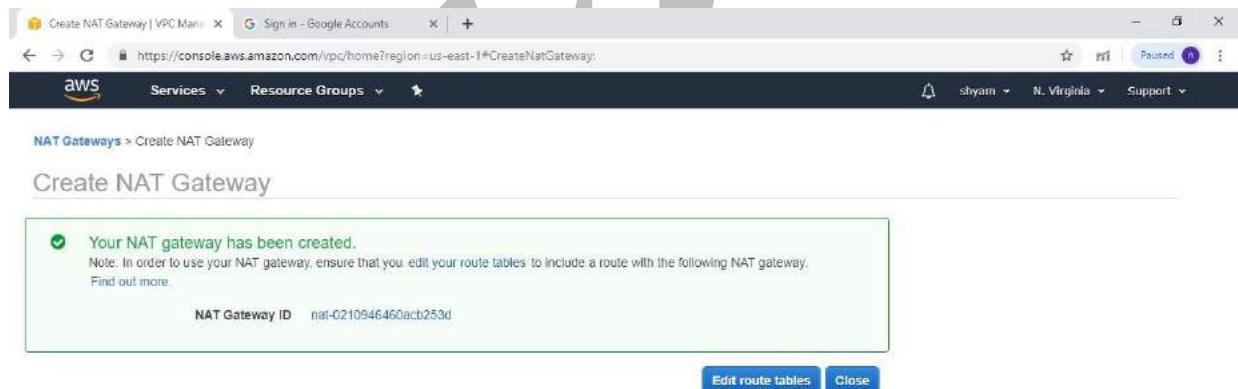
- Select NAT gateway on leftside panel of your vpc section and click on create NAT Gateway



- Subnet: select public subnet to create NAT gateway
- Elastic IP Allocation ID: select Elastic IP if you don't have Elastic IP then click on create NEW EIP then new Elastic IP is allocated
- Click on create NAT Gateway

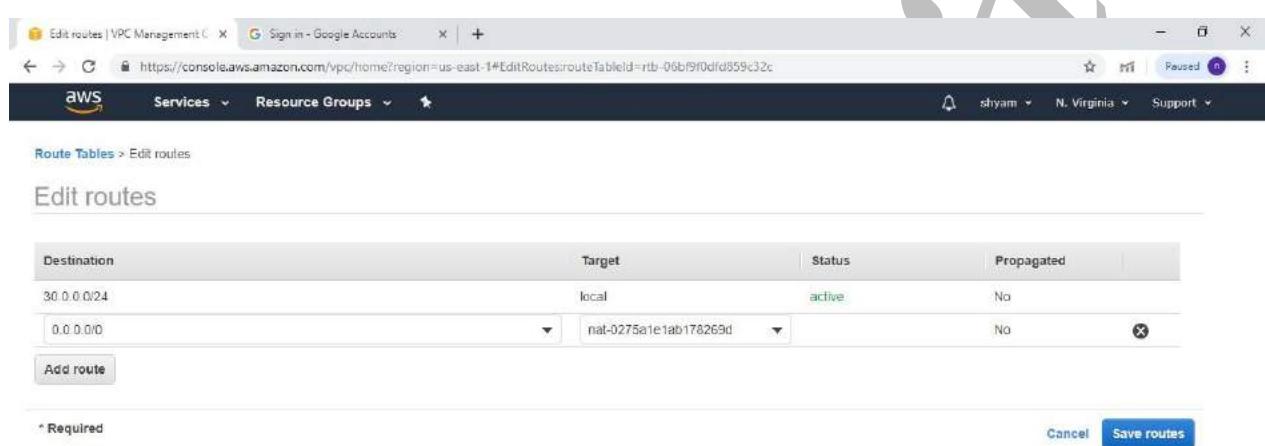


- Click on close



Attach NAT gateway in private subnet

- Select private subnet in same vpc and edit the route table and add NAT gateway
- Destination: 0.0.0.0/0
- Target: select NAT gateway
- Click on save routes



Delete NAT gateway

- Select NAT gateway and goto actions and click on Delete NAT gateway

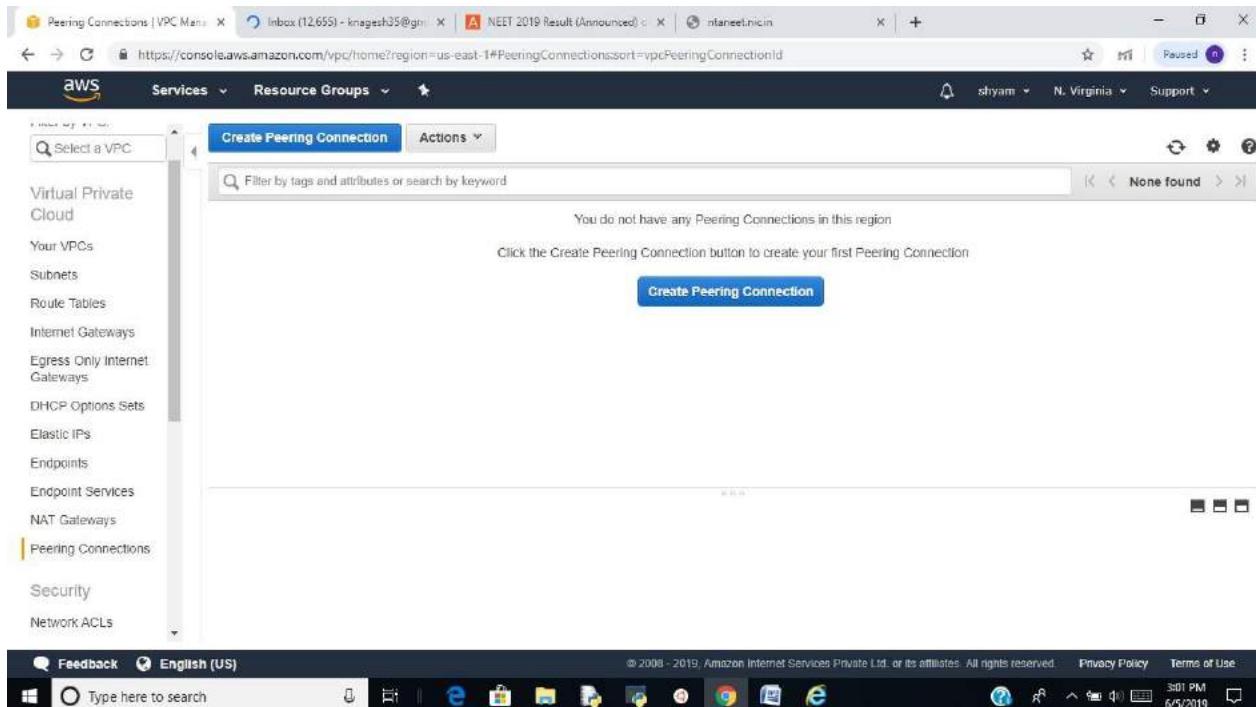
Peering connections

- Peering Connection is used to communicate from one vpc to another from private cluster or subnet to another vpc private subnet

- It is the one way communication only

Create Peering Connection

- Click on peering connection on left side panel of VPC and click on create peering connection

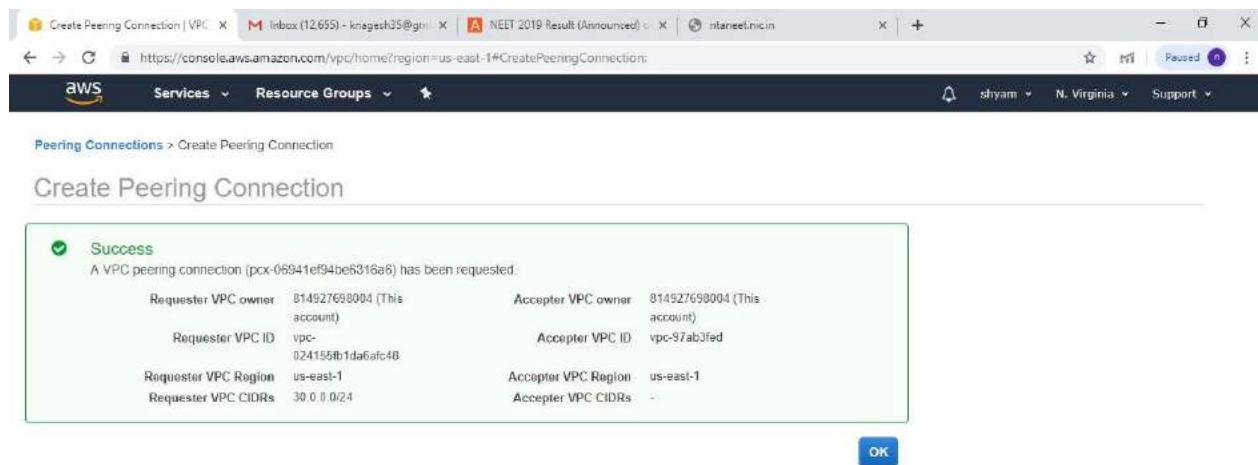


- Peering connection name tag: enter some name for peering connection
- VPC(Requester) : Enter source or sender VPC id
- Account: select my account if do want to communicate vpc in another account then select another account option
- Region: Enter the region of destination vpc
- VPC(Acceptor): enter the Acceptor or receiver VPC id
- Click on Create peering connection

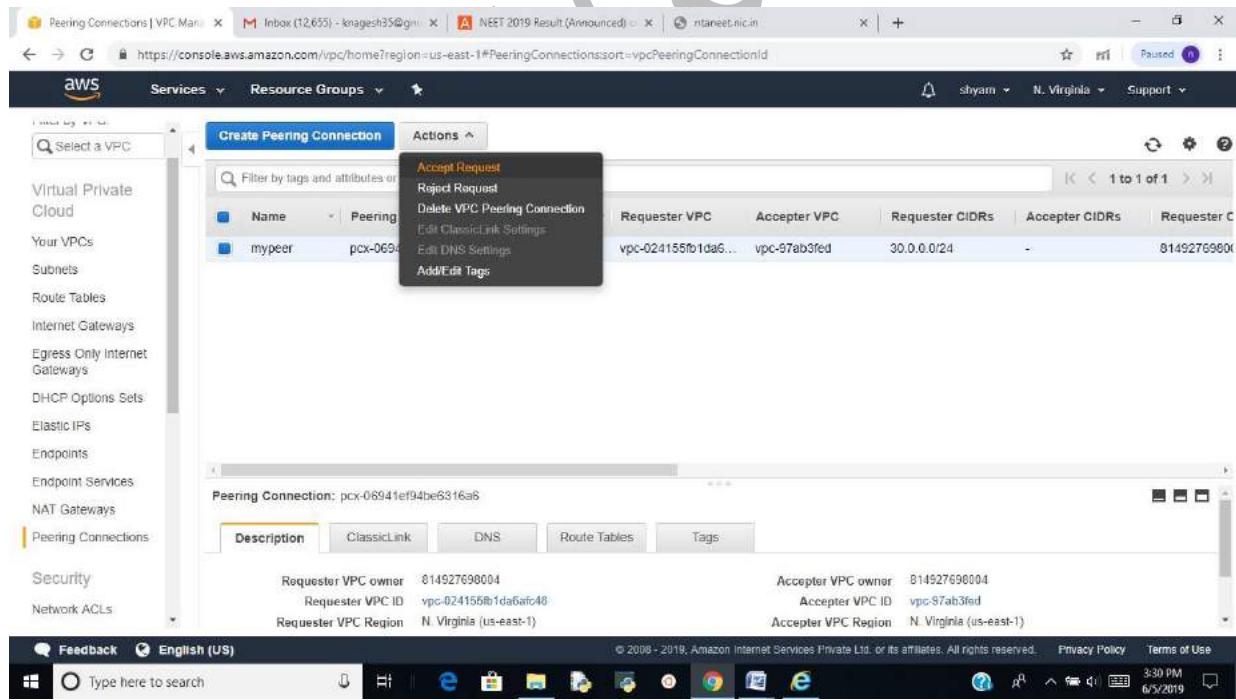
Amazon Web Services

The screenshot shows the 'Create Peering Connection' page in the AWS VPC console. At the top, there are tabs for 'Peering Connections > Create Peering Connection'. Below this, a 'Peering connection name tag' input field contains 'mypeer'. A section titled 'Select a local VPC to peer with' shows a dropdown menu set to 'vpc-024155fb1da6afc48'. A table lists one CIDR range: '30.0.0.0/24' with status 'associated'. Another section titled 'Select another VPC to peer with' includes 'Account' and 'Region' dropdowns, both set to 'My account' and 'This region (us-east-1)'. A 'VPC (Acceptor)' dropdown is set to 'vpc-97ab3fed'. A table below shows a single CIDR range '172.31.0.0/16' with status 'associated'. At the bottom right, there are 'Cancel' and 'Create Peering Connection' buttons.

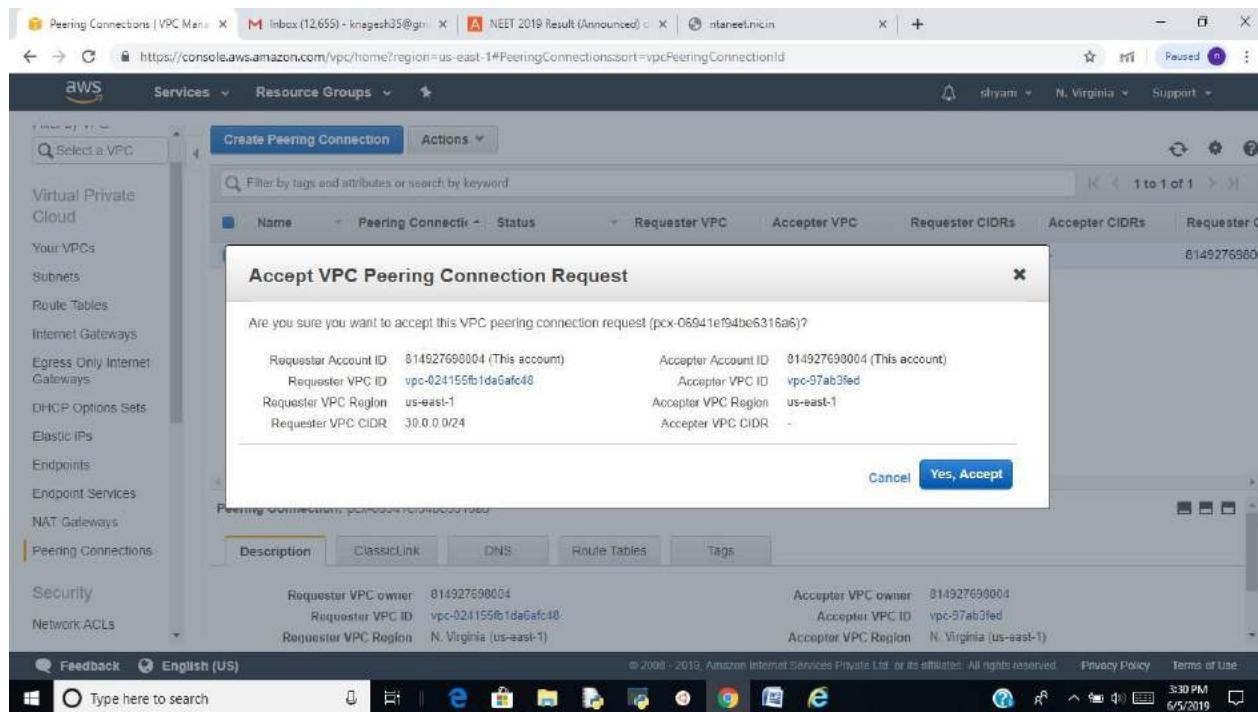
□ Click on ok



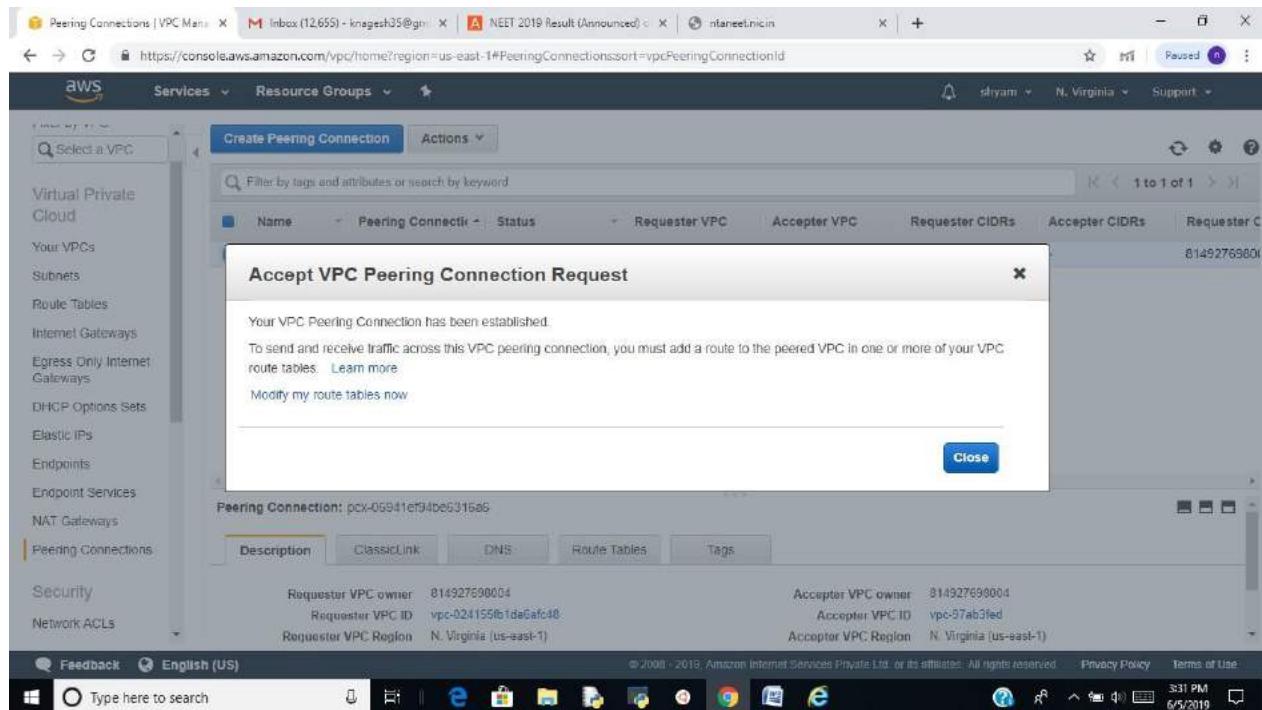
- Select this peering connection in Peering Connection section and goto actions and click on accept request to activate this peering connection



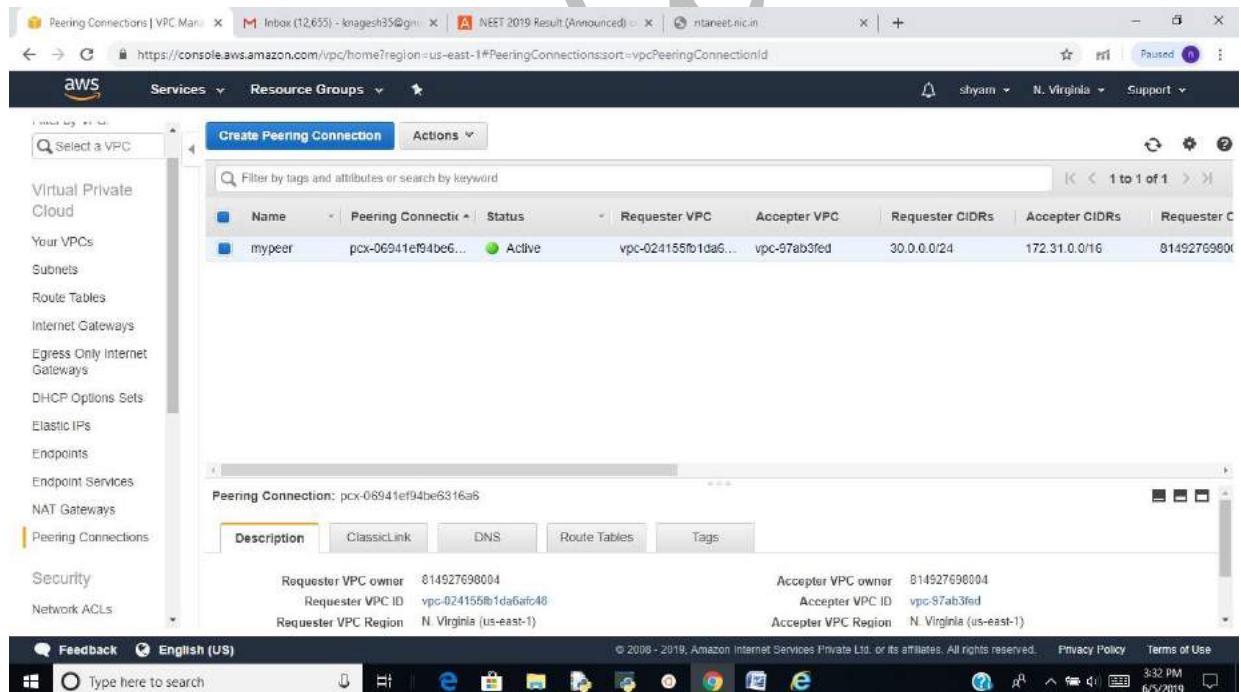
- Click on yes, accept



- Click on close



- Peering connection is activated now



Peering Connection Mapping

- Select requester vpc any subnet route table and click Edit Routes
- Click on Add route
- Desination: Enter the CIDR of accepter or receiver subnet
- Target: select Peering Connection
- Click Save routes

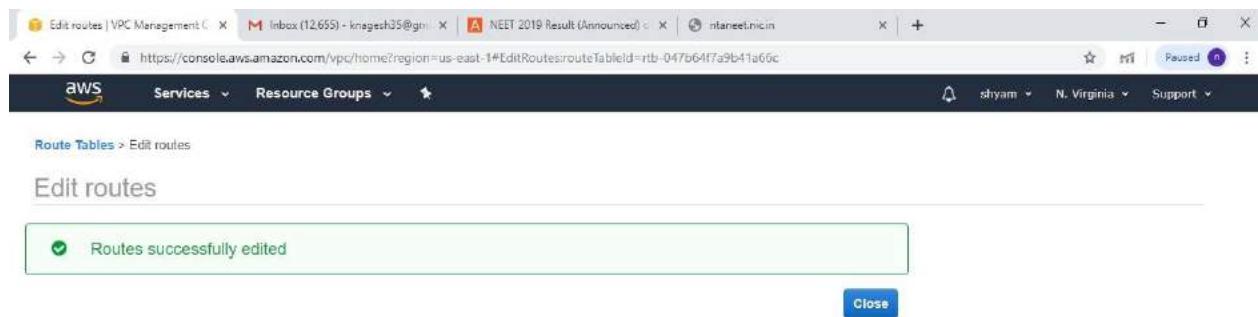
Destination	Target	Status	Propagated
30.0.0.0/24	local	active	No
172.31.0.0/16	pex-06941ef94be6315a8		No

Add route

* Required

Cancel Save routes

- Click on close



- Select accepter or reciver vpc subnet route table and click edit route table
- Click on Edit routes
- Destination: Enter Requester or source subnet CIDR block address
- Target: select Peering Connection that is previously created
- Click on Save Routes

The screenshot shows the AWS VPC Management console with the URL <https://console.aws.amazon.com/vpc/home?region=us-east-1#EditRoutes:routeTableId=rtb-48d7c437>. The page title is "Edit routes". It displays a table of routes:

Destination	Target	Status	Propagated
172.31.0.0/16	local	active	No
0.0.0.0/0	lgw-d7f209ac	active	No
30.0.0.129/25	pox-06941ef94be631ba6		No

Buttons at the bottom include "Add route", "Required", "Cancel", and "Save routes".



- Click on close

The screenshot shows the same AWS VPC Management console page as before, but now with a success message: "Routes successfully edited". A "Close" button is visible at the bottom right.



- Launch two EC2 instances one is Requester subnet that is called source machine and another one launch in accepter subnet that is called receiver
- Then try connect from source machine to receiver machine and you sucessfully connected by using this peering connection

DyaSaa

Amazon Route 53

Amazon Route 53 is a highly available and scalable Domain Name System (DNS) web service. You can use Route 53 to perform three main functions in any combination: domain registration, DNS routing, and health checking. If you choose to use Route 53 for all three functions, perform the steps in this order:

Register domain names

Route internet traffic to the resources for your domain

Check the health of your resources

How Domain Registration Works

If you want to create a website or a web application, you start by registering the name of your website, known as a domain name. Your domain name is the name, such as example.com, that your users enter in a browser to display your website.

Here's an overview of how you register a domain name with Amazon Route 53:

1. You choose a domain name and confirm that it's available, meaning that no one else has registered the domain name that you want.

If the domain name you want is already in use, you can try other names or try changing only the *top-level domain*, such as .com, to another top-level domain, such as .ninja or .hockey. For a list of the top-level domains that Route 53 supports,

2. You register the domain name with Route 53. When you register a domain, you provide names and contact information for the domain owner and other contacts.

Configure Amazon Route 53 to Route Internet Traffic for Your Domain

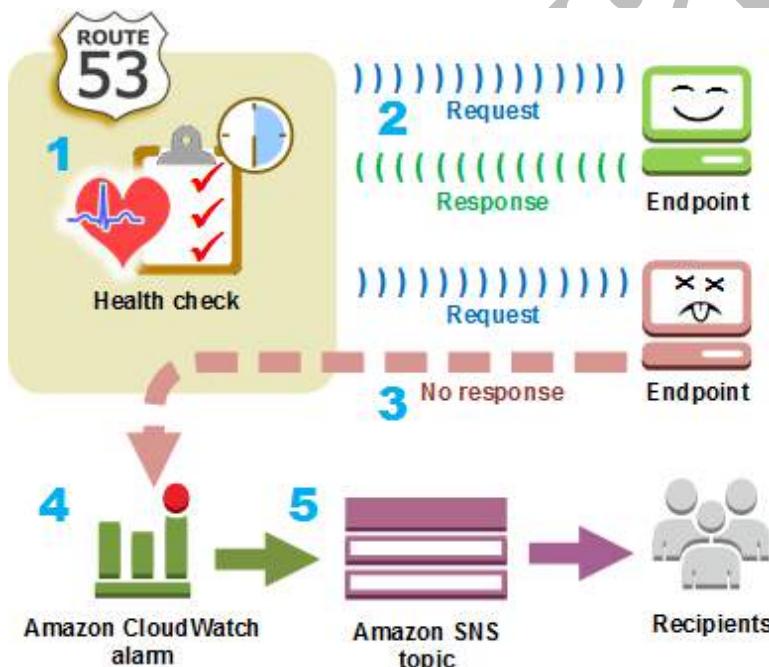
Here's an overview of how to use the Amazon Route 53 console to register a domain name and configure Route 53 to route internet traffic to your website or web application.

1. You register the domain name that you want your users to use to access your content.
2. After you register your domain name, Route 53 automatically creates a public hosted zone that has the same name as the domain.
3. To route traffic to your resources, you create *records*, also known as *resource record sets*, in your hosted zone. Each record includes information about how you want to route traffic for your domain, such as the following:

Amazon Route 53 Checks the Health of Your Resources

Amazon Route 53 health checks monitor the health of your resources such as web servers and email servers. You can optionally configure Amazon CloudWatch alarms for your health checks, so that you receive notification when a resource becomes unavailable.

Here's an overview of how health checking works if you want to be notified when a resource becomes unavailable:



1. You create a health check and specify values that define how you want the health check to work, such as the following:
 - The IP address or domain name of the endpoint, such as a web server, that you want Route 53 to monitor. (You can also monitor the status of other health checks, or the state of a CloudWatch alarm.)
 - The protocol that you want Amazon Route 53 to use to perform the check: HTTP, HTTPS, or TCP.
 - How often you want Route 53 to send a request to the endpoint. This is the *request interval*.
 - How many consecutive times the endpoint must fail to respond to requests before Route 53 considers it unhealthy. This is the *failure threshold*.
 - Optionally, how you want to be notified when Route 53 detects that the endpoint is unhealthy. When you configure notification, Route 53 automatically sets a CloudWatch alarm. CloudWatch uses Amazon SNS to notify users that an endpoint is unhealthy.
2. Route 53 starts to send requests to the endpoint at the interval that you specified in the health check.

If the endpoint responds to the requests, Route 53 considers the endpoint to be healthy and takes no action.

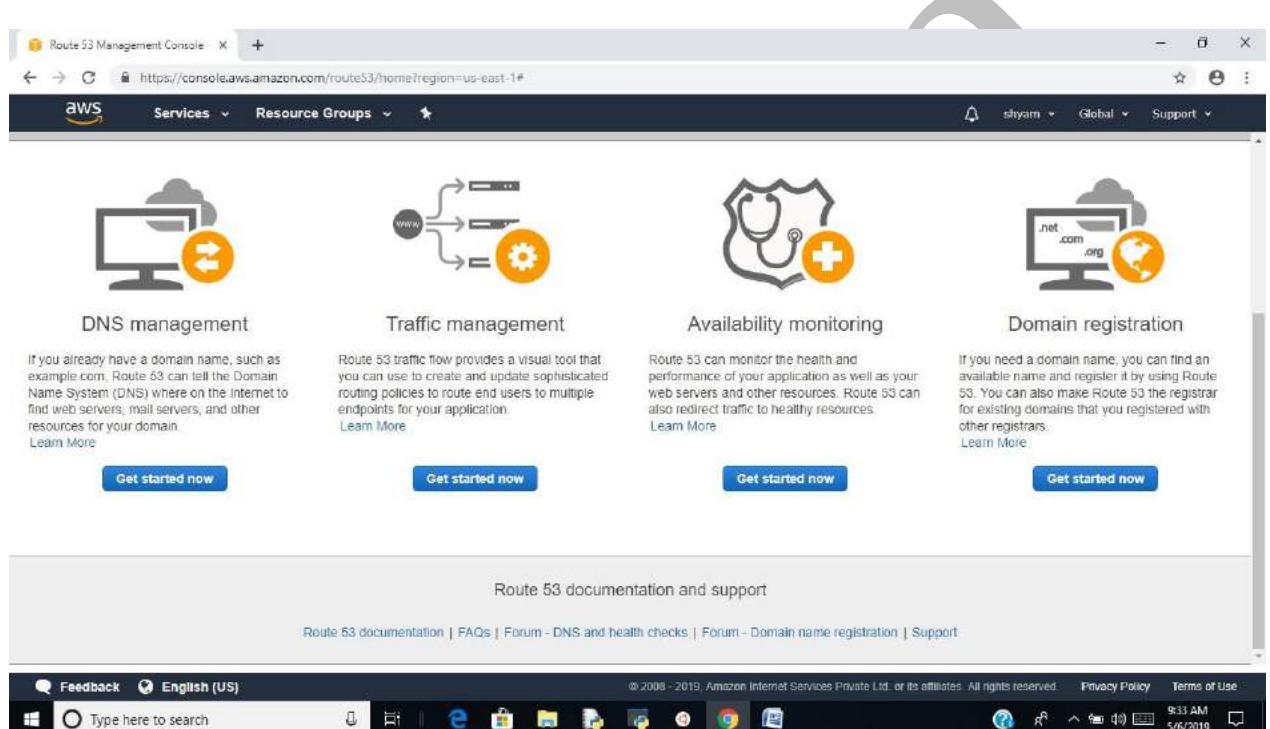
3. If the endpoint doesn't respond to a request, Route 53 starts to count the number of consecutive requests that the endpoint doesn't respond to:
 - If the count reaches the value that you specified for the failure threshold, Route 53 considers the endpoint unhealthy.
 - If the endpoint starts to respond again before the count reaches the failure threshold, Route 53 resets the count to 0, and CloudWatch doesn't contact you.
4. If Route 53 considers the endpoint unhealthy and if you configured notification for the health check, Route 53 notifies CloudWatch.

If you didn't configure notification, you can still see the status of your Route 53 health checks in the Route 53 console.

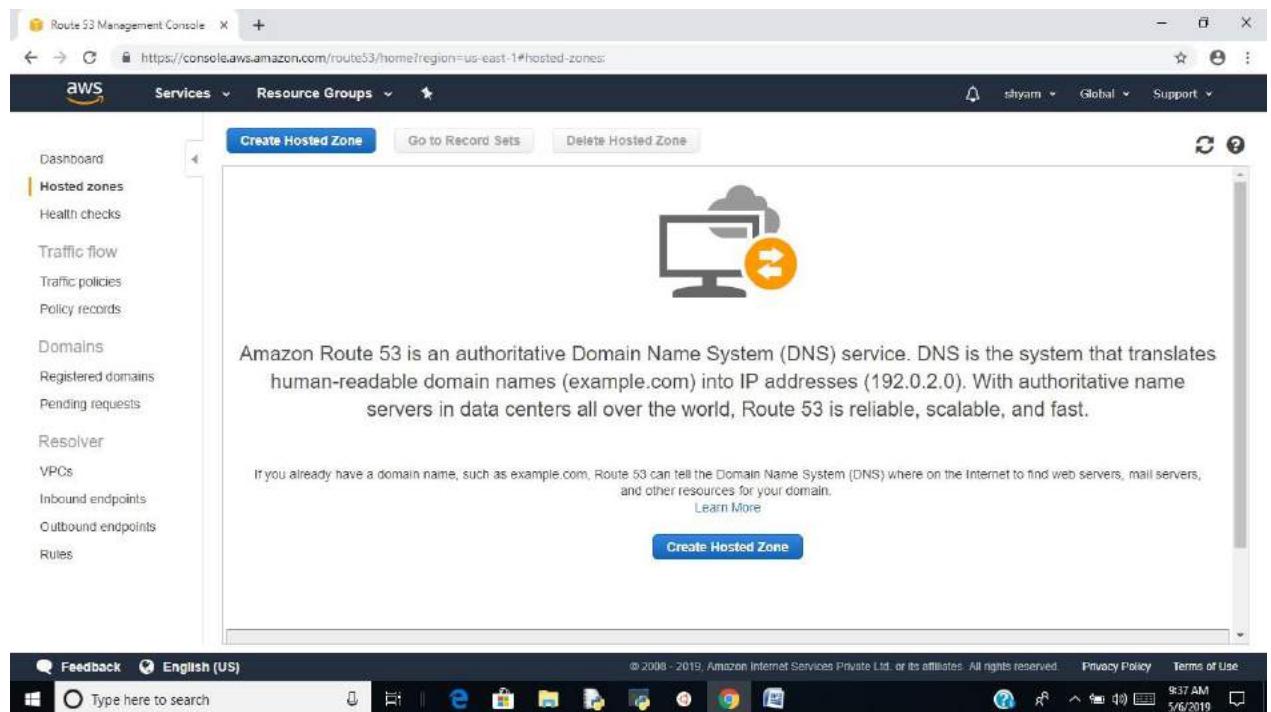
5. If you configured notification for the health check, CloudWatch triggers an alarm and uses Amazon SNS to send notification to the specified recipients.

UseCase

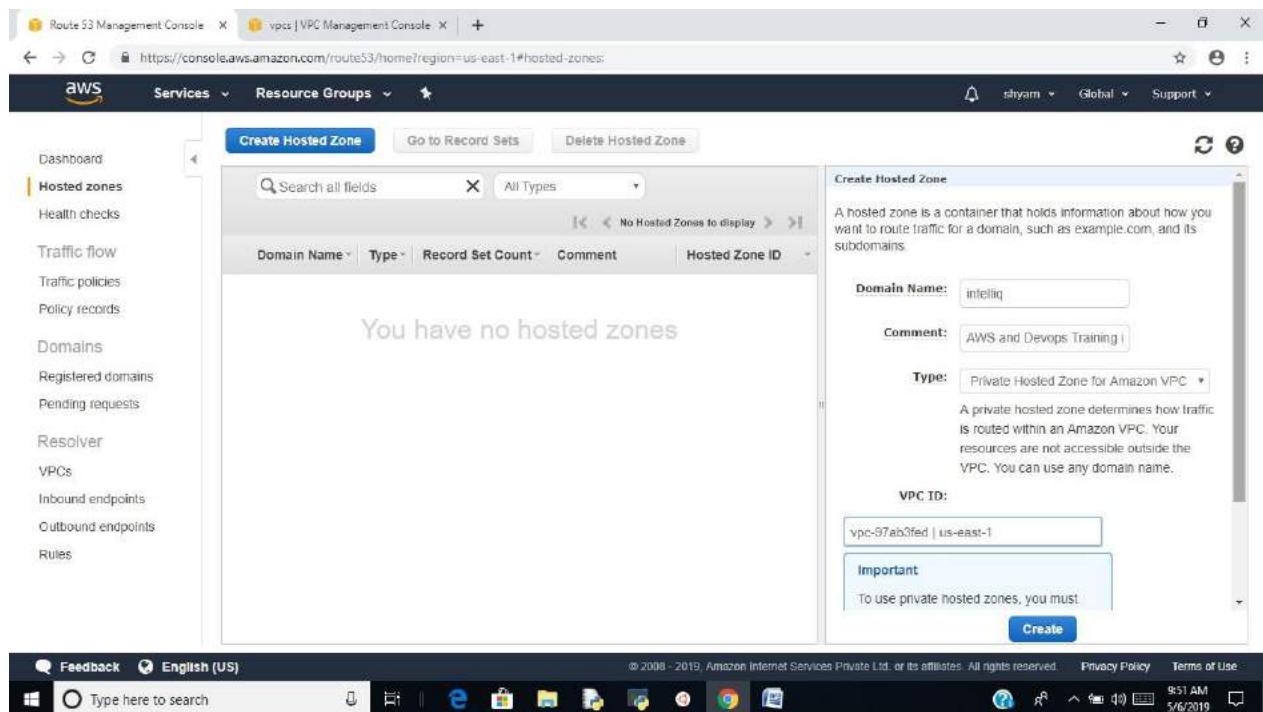
- Select Route53 Service in AWS console
- Then you can see the topics of DNS Management, Traffic Management, Availability Monitoring and Domain Registration.
- Click on Domain Registration
- If you already have a domain name, such as example.com, Route 53 can tell the Domain Name System (DNS) where on the Internet to find web servers, mail servers, and other resources for your domain.



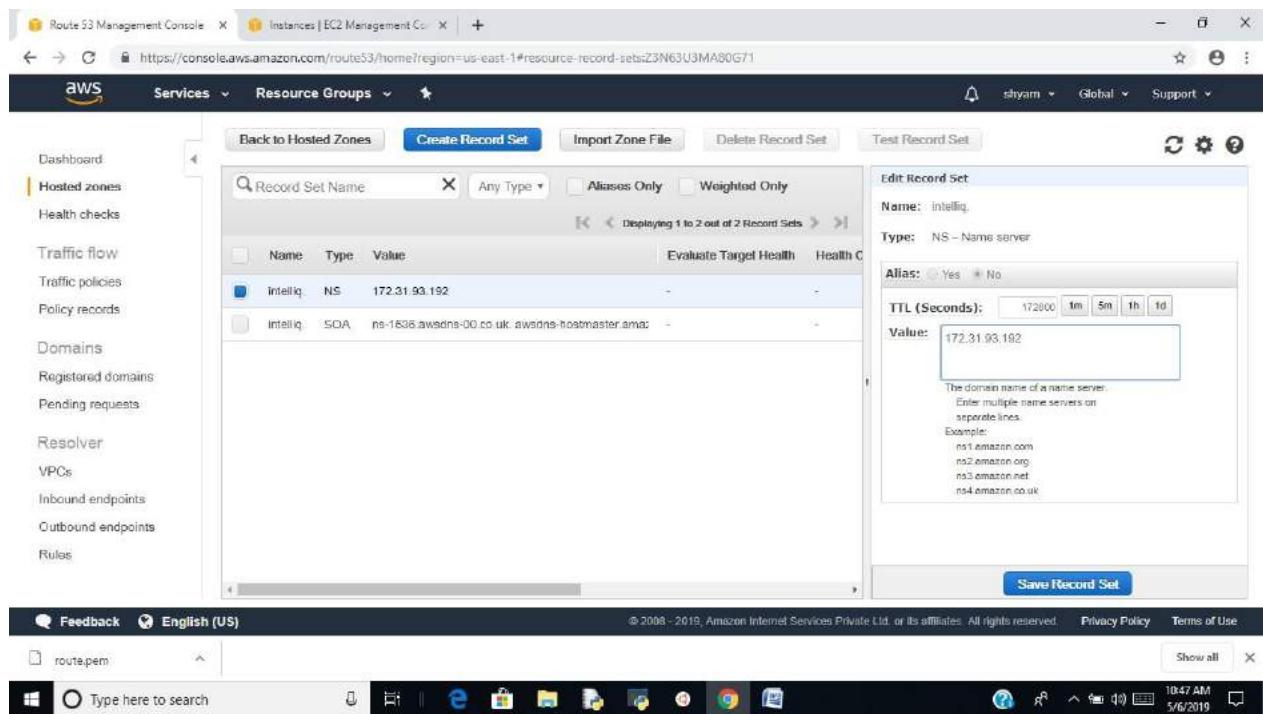
- Amazon Route 53 is an authoritative Domain Name System (DNS) service. DNS is the system that translates human-readable domain names (example.com) into IP addresses (192.0.2.0). With authoritative name servers in data centers all over the world, Route 53 is reliable, scalable, and fast.
- Click on create Hosted Zone



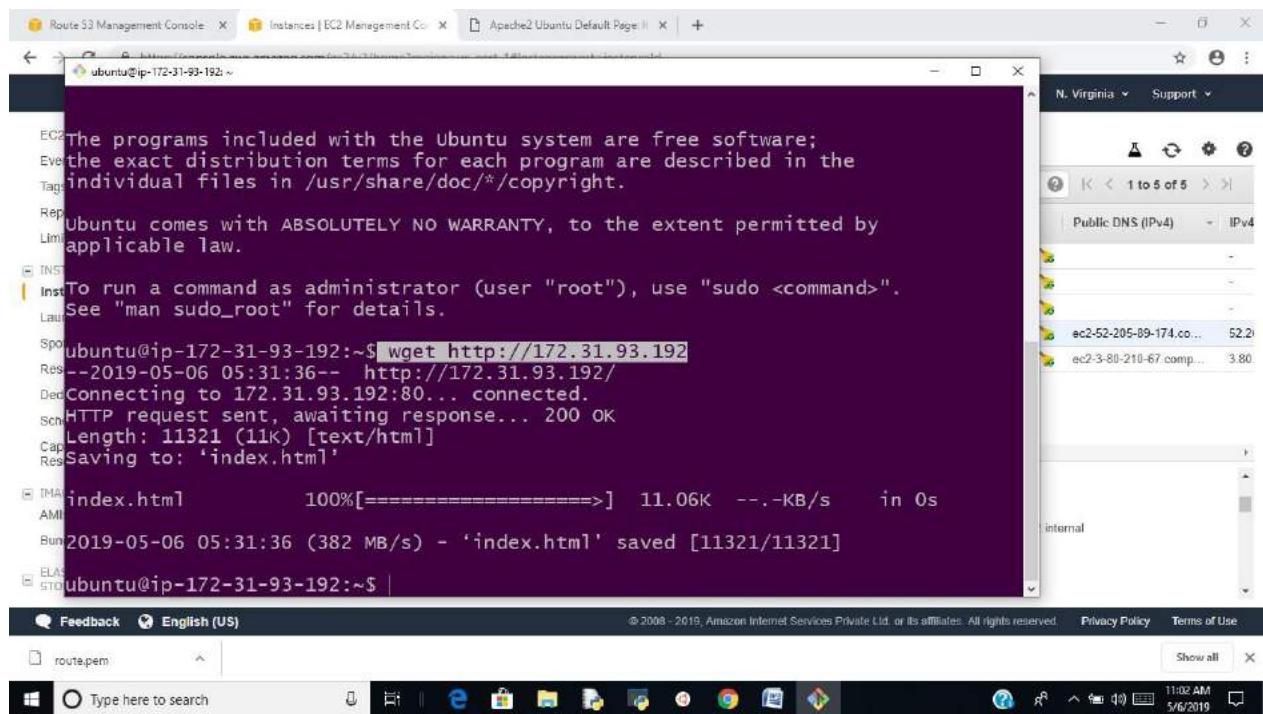
- A hosted zone is a container that holds information about how you want to route traffic for a domain.
- Domain Name: enter your Domain Name that is already registered. If you select private hosted zone type you give any domain name there is no need to register
- Comments: write any comments about your domain and it is the optional
- Type: It is the type of hosted zone that is public or private hosted zone. If you select public you can route your traffic through internet and if you select private hosted zone you can route your traffic in with that VPC only. So select private
- Then you enter the vpc id and region
- Click on create.



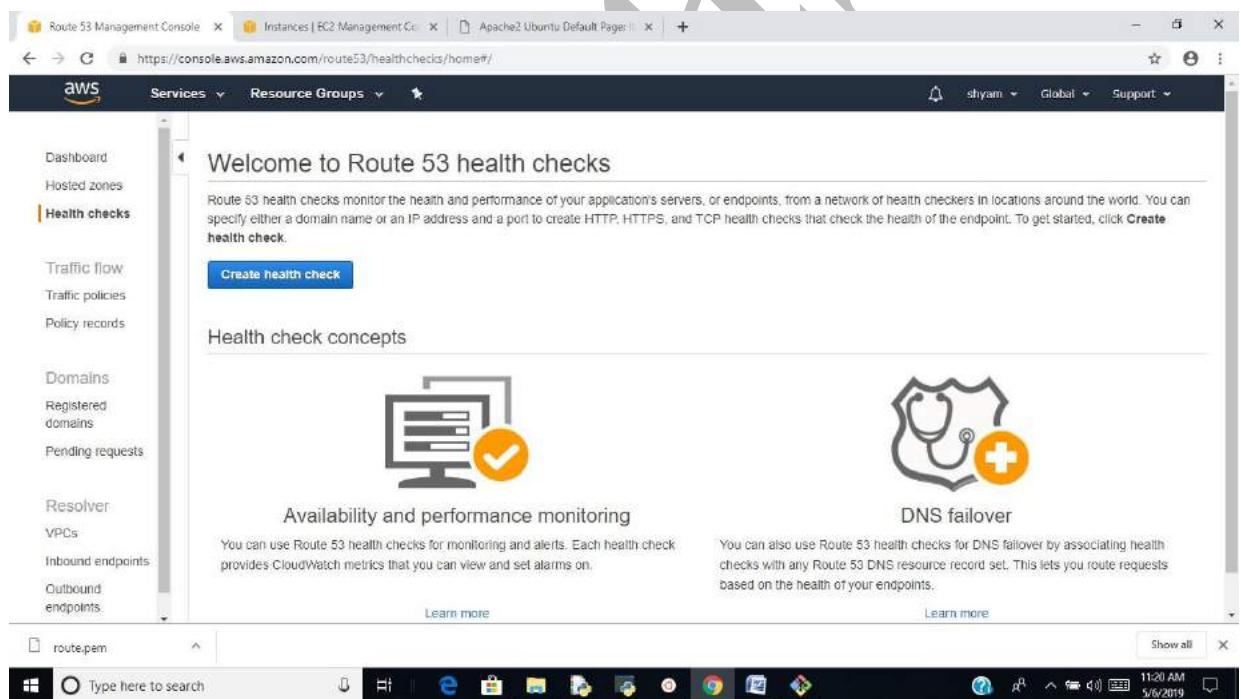
- Click on Create Record Set and Select previously created records in the list
 Name: DNS Name
 Type: NS – Name Server
 Value: enter the private ip address of your server in which server your application is existed.
- So we need to Launch Two EC2 Instances and install sample apache2 application on it. Both Instances are Launched in single VPC that is selected in private Hosted Zone.
- Enter private ip address of server of the application in which server do you want to register with DNS.
- Click on save Record



- You can check these services are running with DNS or not.
- Connected to one ec2 instace and check our previously registered application is running or not by using wget command.



- Click on create Helth check



Configure Health check

Name: name for helthcheck(webhealthcheck)

What to monitor: select Endpoint

- Monitor an endpoint

Specify an end point by : select ip address

Protocol: select HTTP

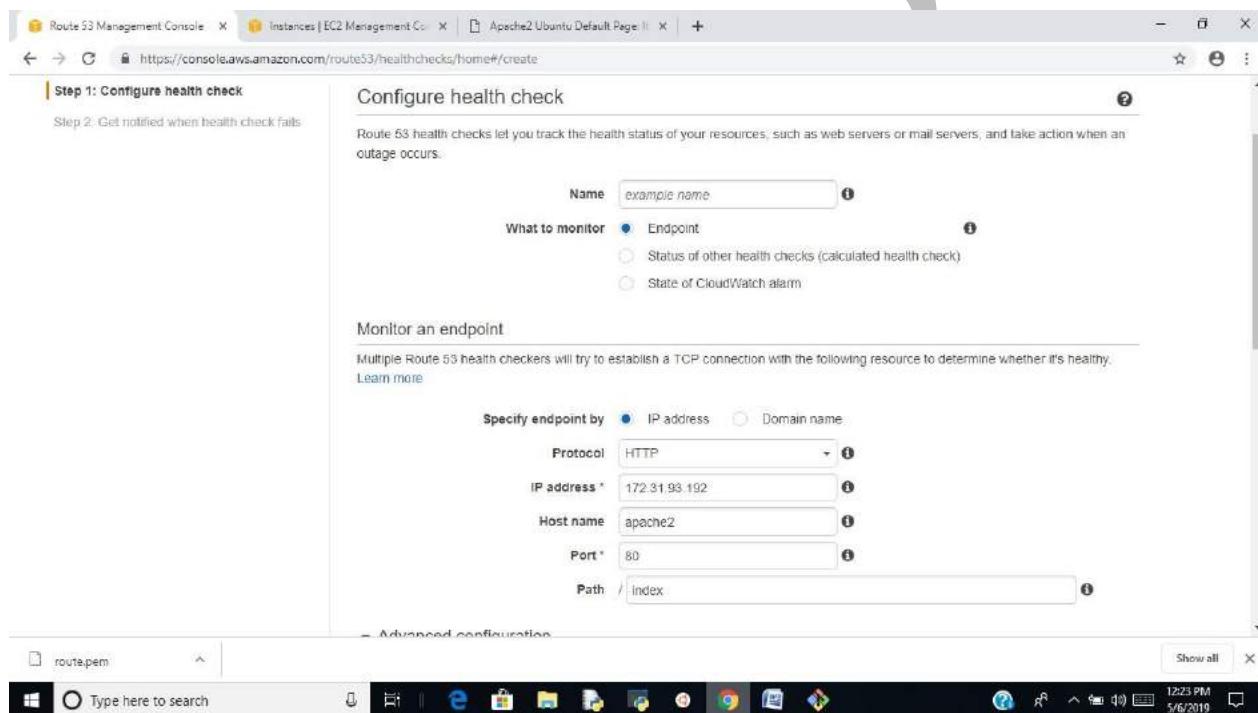
Ip address: enter ip address

Host Name: enter some host name and it is the optional value

Port: enter port number in which port number the service is running (80)

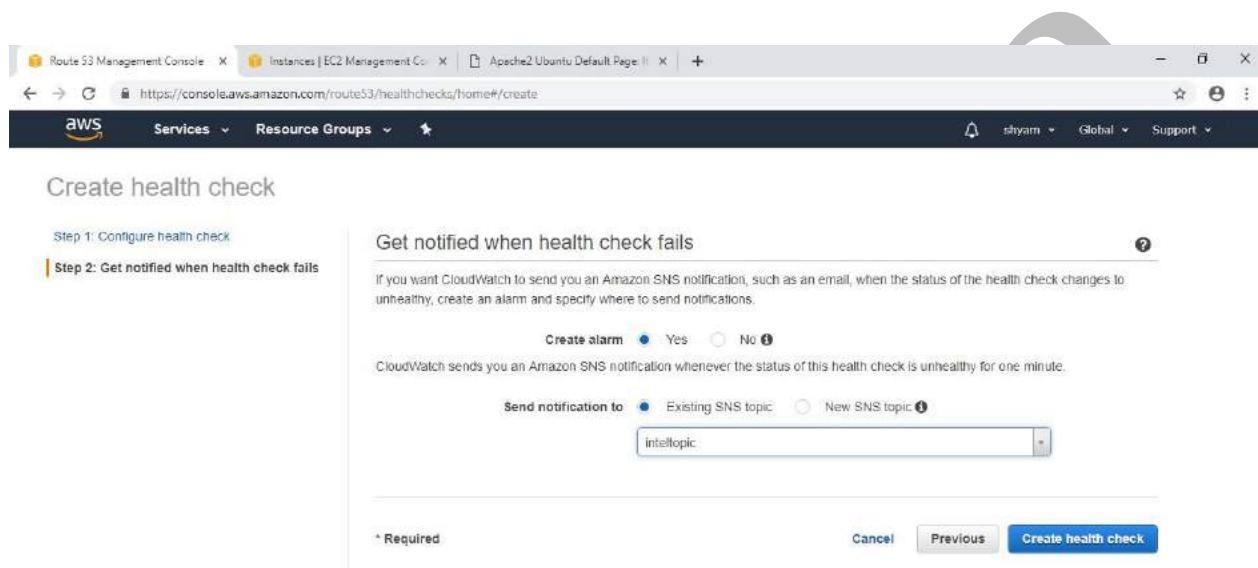
Path: path for checking service (index)

- Advanced Configuration: give default settings and click on next

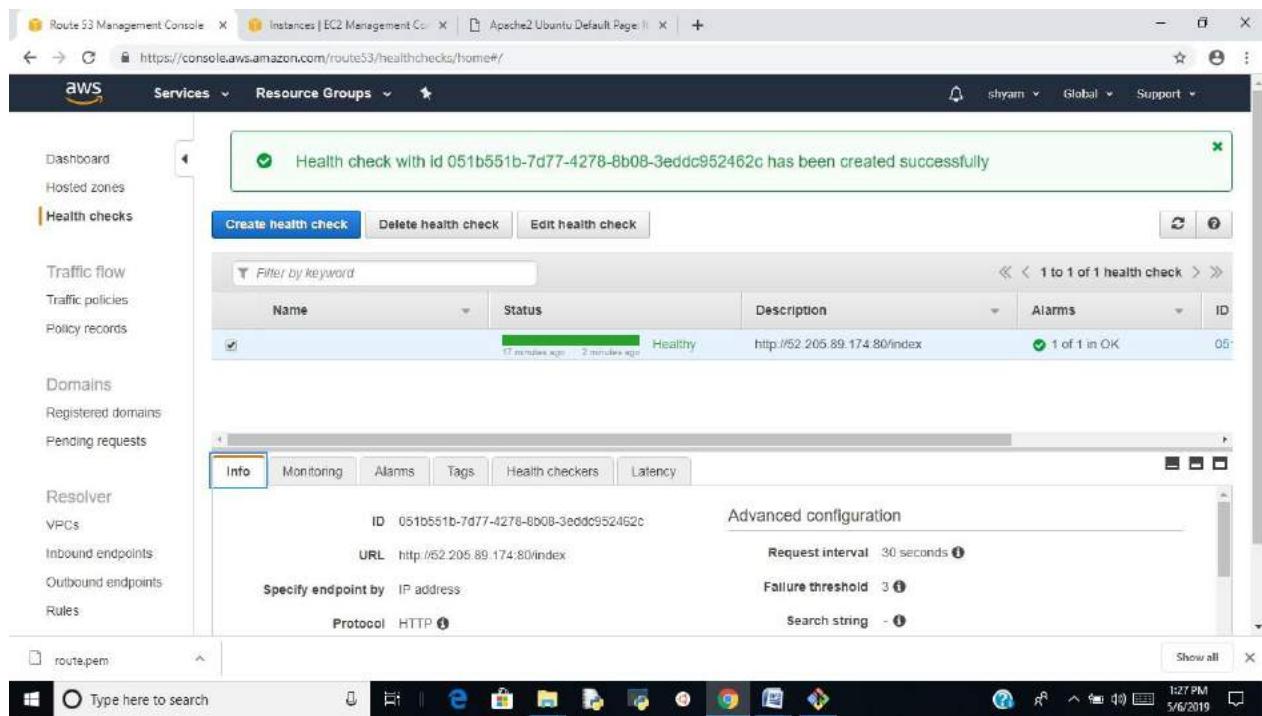


- Get notified when helth check fails : If you want CloudWatch to send you an Amazon SNS notification, such as an email, when the status of the health check changes to unhealthy, create an alarm and specify where to send notifications.

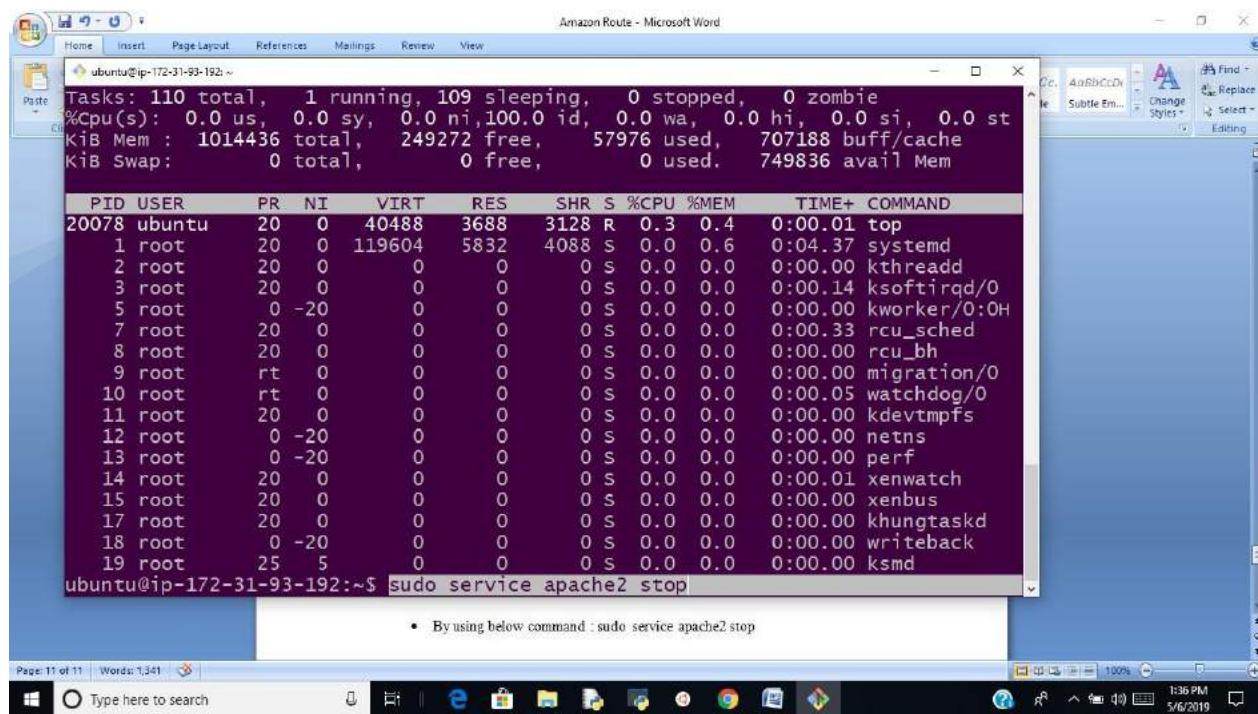
- Create alarm: select no or do you want make any alarms you can select yes. Here select yes
- Send notification to: select Existing SNS Topic, if you do not have Existing SNS then select New SNS Topic. Here select Existing SNS Topic and enter topic name
- Click on Create Health Check



- You can see the Helth Checks



- Stop the apache2 service and check the alarm activate or not. If alarm activate you receive the SNS notification to your mail
- By using below command : sudo service apache2 stop



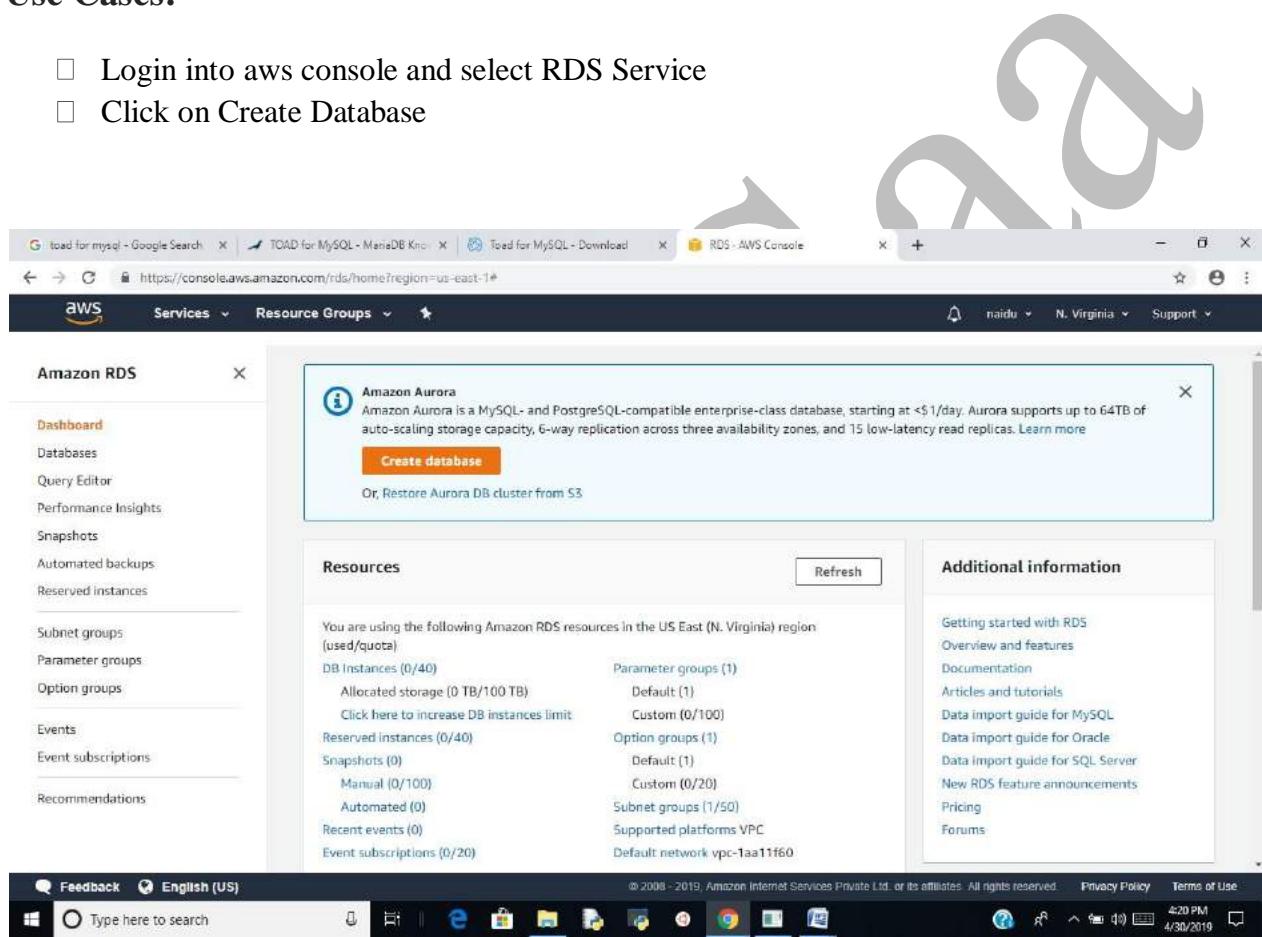
- The Helth Check is going to state unhealthy and Alarm Activated
- To see the SNS Notification is received or not by checking your mail

Amazon Relational Database Service (Amazon RDS)

Amazon Relational Database Service (Amazon RDS) is a web service that makes it easier to set up, operate, and scale a relational database in the cloud. It provides cost-efficient, resizable capacity for an industry-standard relational database and manages common database administration tasks.

Use Cases:

- Login into aws console and select RDS Service
- Click on Create Database



Select Engine mysql and click on Next

The screenshot shows the 'Select engine' step of the AWS RDS setup wizard. On the left, a sidebar lists steps: Step 2 (Choose use case), Step 3 (Specify DB details), and Step 4 (Configure advanced settings). The main area is titled 'Select engine' and contains a grid of engine options:

Engine options		
<input type="radio"/> Amazon Aurora Amazon Aurora	<input checked="" type="radio"/> MySQL 	<input type="radio"/> MariaDB 
<input type="radio"/> PostgreSQL 	<input type="radio"/> Oracle 	<input type="radio"/> Microsoft SQL Server 

MySQL

MySQL is the most popular open source database in the world. MySQL on RDS offers the rich features of the MySQL community edition with the flexibility to easily scale compute resources or storage capacity for your database.

- Supports database size up to 32 TiB.
- Supports General Purpose, Memory Optimized, and Burstable Performance instance classes.

At the bottom, there is a standard Windows taskbar with icons for File Explorer, Task View, Start, Taskbar settings, and system status.

- Choose use case : use this database for production purposes
- Select Production – Amazon Aurora click on Next

The screenshot shows the 'Choose use case' step of the AWS RDS setup wizard. The sidebar shows steps: Step 1 (Select engine), Step 2 (Choose use case), Step 3 (Specify DB details), and Step 4 (Configure advanced settings). The main area is titled 'Choose use case' and contains a 'Use case' section:

Do you plan to use this database for production purposes?

Use case

Production - Amazon Aurora Recommended
MySQL-compatible, enterprise-class database at 1/10th the cost of commercial databases.

Production - MySQL
Use Multi-AZ Deployment and Provisioned IOPS Storage as defaults for high availability and fast, consistent performance.

Dev/Test - MySQL
This instance is intended for use outside of production or under the RDS Free Usage Tier.

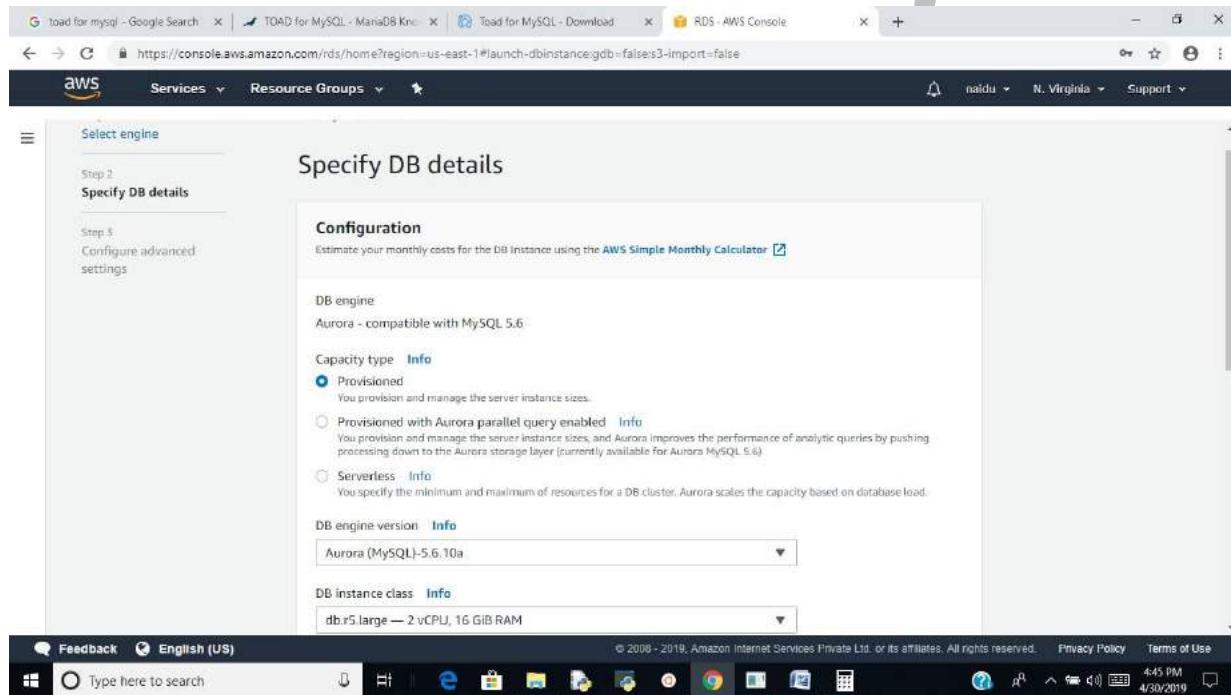
Billing is based on RDS pricing .

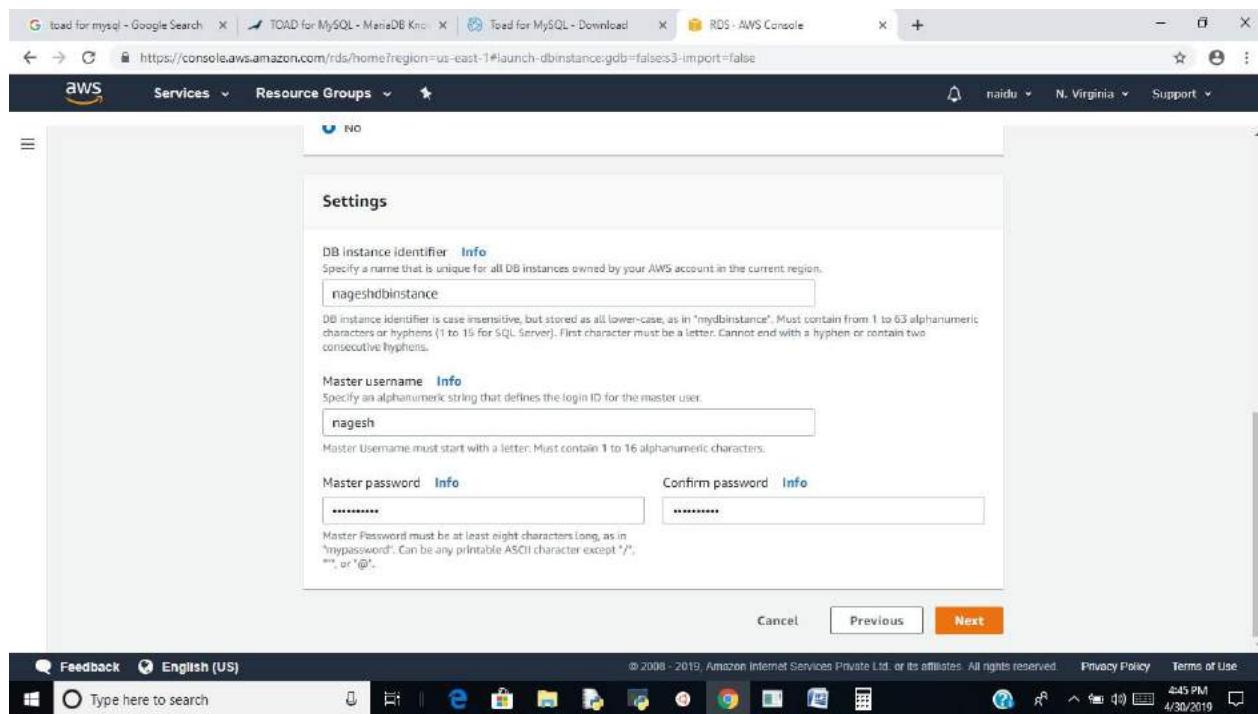
At the bottom are 'Cancel', 'Previous', and 'Next' buttons. The bottom of the screen shows a Windows taskbar.

- Specify DB details: it your DB Instance Configuration Settings
- Capacity Type : select Provisioned
- DB Engine Version: select Aurora (MYSQL)-5.6.10a
- DB Instance class: db.r5.large – 2vCPU , 16 gib RAM
- Multi-AZ deployment: it is used for Deploy your DB Instance Replica into Multiple Availability Zones. Here we select no do you want maintain replicas in different Availability Zones then select first option

Settings:

- DB instance identifier: This is the DB Instance name for unique identification
- Master username: this is the master username for DB instace Connection
- Master Password: password for DB Instance Master user and Click on Next



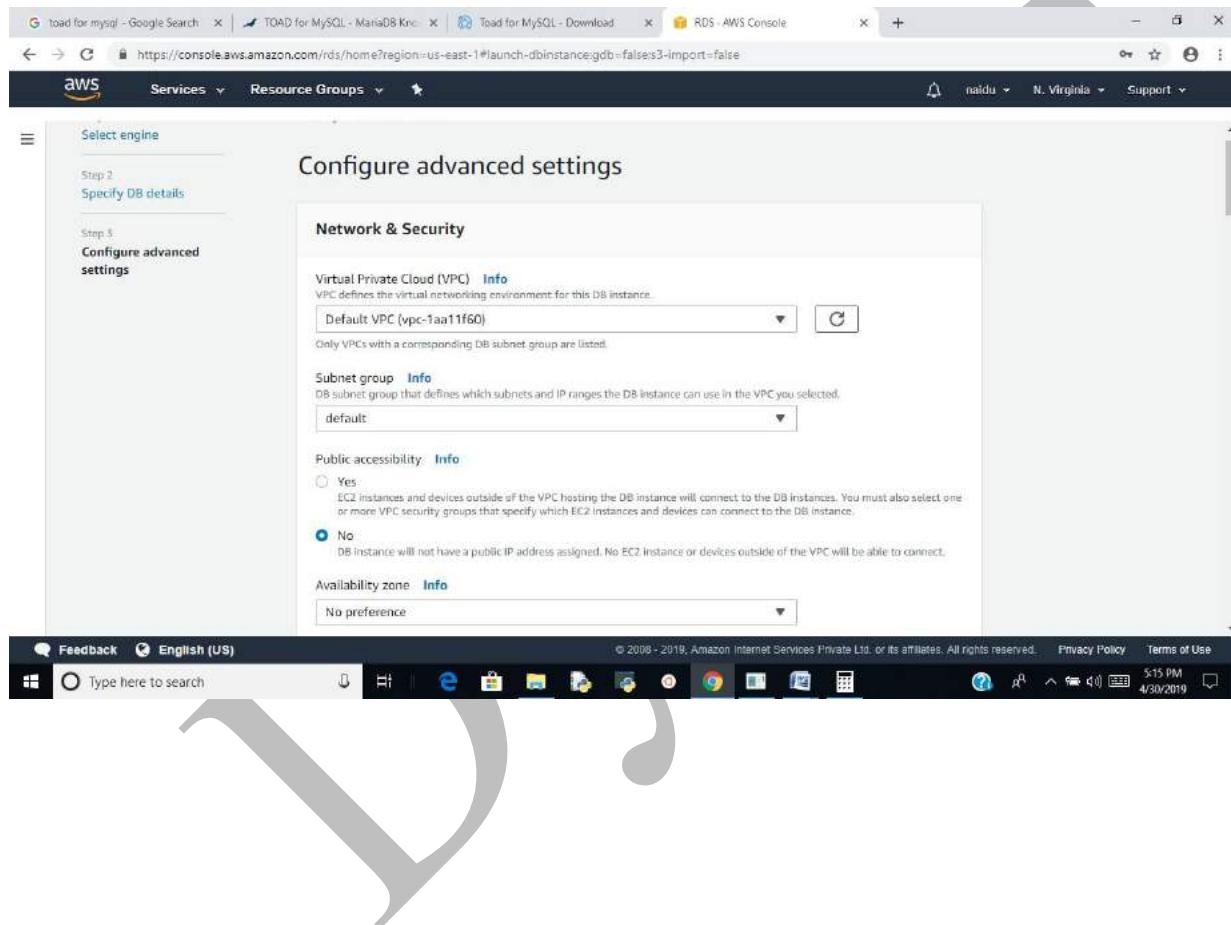


Configure Advanced Settings

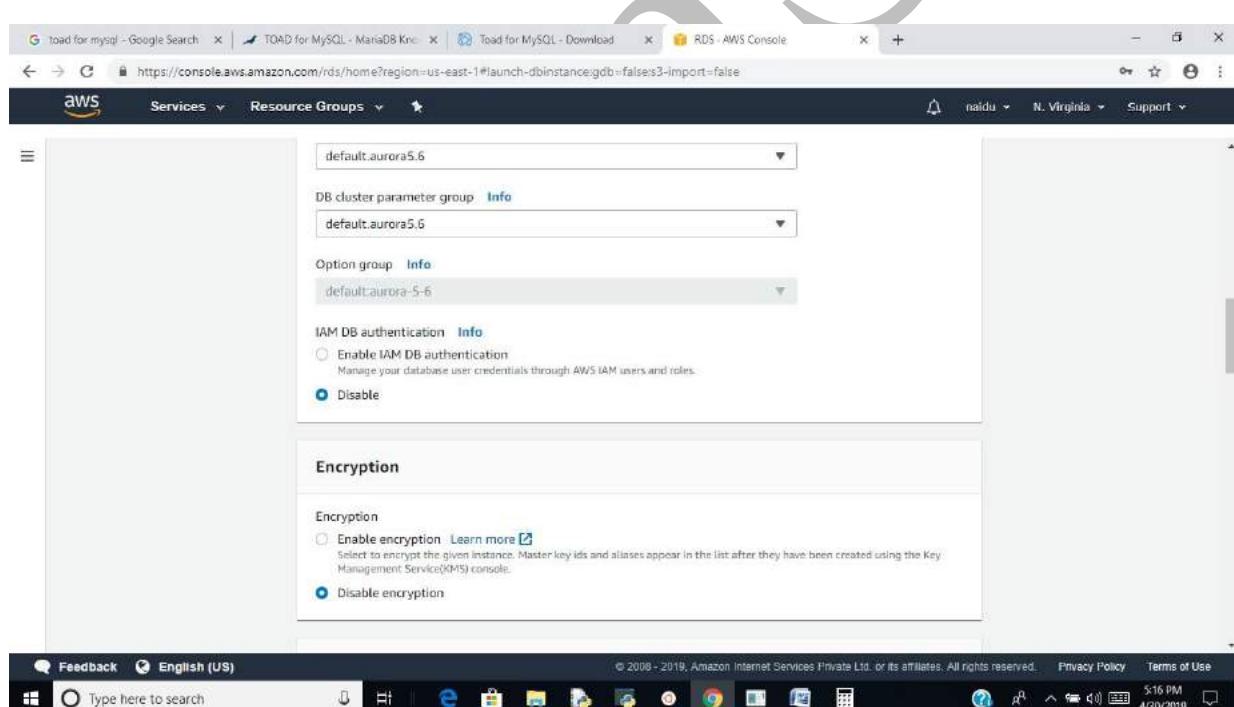
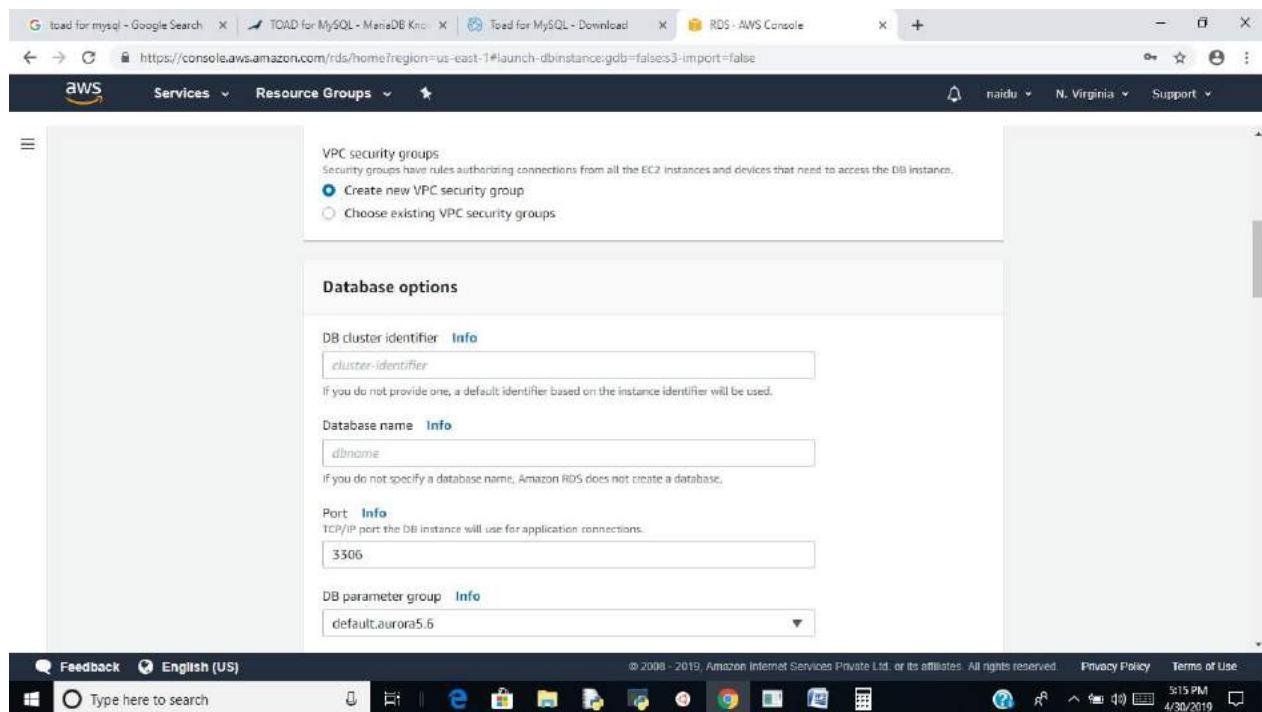
Network & Security

- Virtual Private Cloud (VPC): select The default VPC
- Subnet group: select default subnet
- Public accessibility: do you want to connect to your DB Instance from outside of selected VPC then click YES otherwise click no. if you select NO then you connect from within that VPC only.
- DB cluster identifier: It is The DB Instance Identifier used for unique identification of your DB Instance in the DB Cluster
- Database name: enter mysql Database name
- Port: it is the mysql DB Engine Port Default Port is 3306
- DB parameter group: default.aurora5.6
- DB Cluster parameter group: default.aurora5.6
- IAM DB authentication: select Disable Option
- Encryption: click on Disable
- Failover: select no preference
- Backup: after how many days do you want to backup from DB Instance
- Backtrack: Disable Backtrack option

- Monitoring: Do you want monitor logs from these DB Instance then select enable otherwise select Disable option
- Performance Insights: click on Enable Performance Insights it is increase the performance of your DB Instance
- Maintenance: select Enable auto minor version upgrade
- Deletion protection: unselect Enable deletion protection
- Click on create database



Amazon Web Services



Amazon Web Services

The screenshot shows the AWS RDS console with the URL <https://console.aws.amazon.com/rds/home?region=us-east-1#launch-dbinstance:gdb=false:s3-import=false>. The main content area is titled "Encryption". It contains two options: "Enable encryption" (radio button) and "Disable encryption" (radio button, selected). A note below says: "Select to encrypt the given instance. Master key IDs and aliases appear in the list after they have been created using the Key Management Service (KMS) console." Below this is the "Failover" section, which includes a "Priority" dropdown set to "No preference".

The screenshot shows the AWS RDS console with the same URL as the previous screenshot. The main content area has sections for "Backtrack" and "Monitoring". In the "Backtrack" section, there is a checkbox "Copy tags to snapshots" and two options: "Enable Backtrack" (radio button) and "Disable Backtrack" (radio button, selected). In the "Monitoring" section, there is a checkbox "I authorize RDS to create the IAM role rds-monitoring-role". Below this are "Monitoring Role" (dropdown set to "Default") and "Granularity" (dropdown set to "60 seconds").

The screenshot shows the AWS RDS console with the URL <https://console.aws.amazon.com/rds/home?region=us-east-1#launch-dbinstance:gdb=false;s3-import=false>. The main content area is titled "Performance Insights" and contains two radio button options: "Enable Performance Insights" (selected) and "Disable Performance Insights". Below this is a section titled "Log exports" with four checkbox options: "Audit log", "Error log", "General log", and "Slow query log". A note about the IAM role is present, stating: "The following service-linked role is used for publishing logs to CloudWatch Logs." A callout box at the bottom right of this section says: "Ensure that General, Slow Query, and Audit Logs are turned on. Error logs are enabled by default." The status bar at the bottom indicates the date and time as 4/30/2019 5:18 PM.

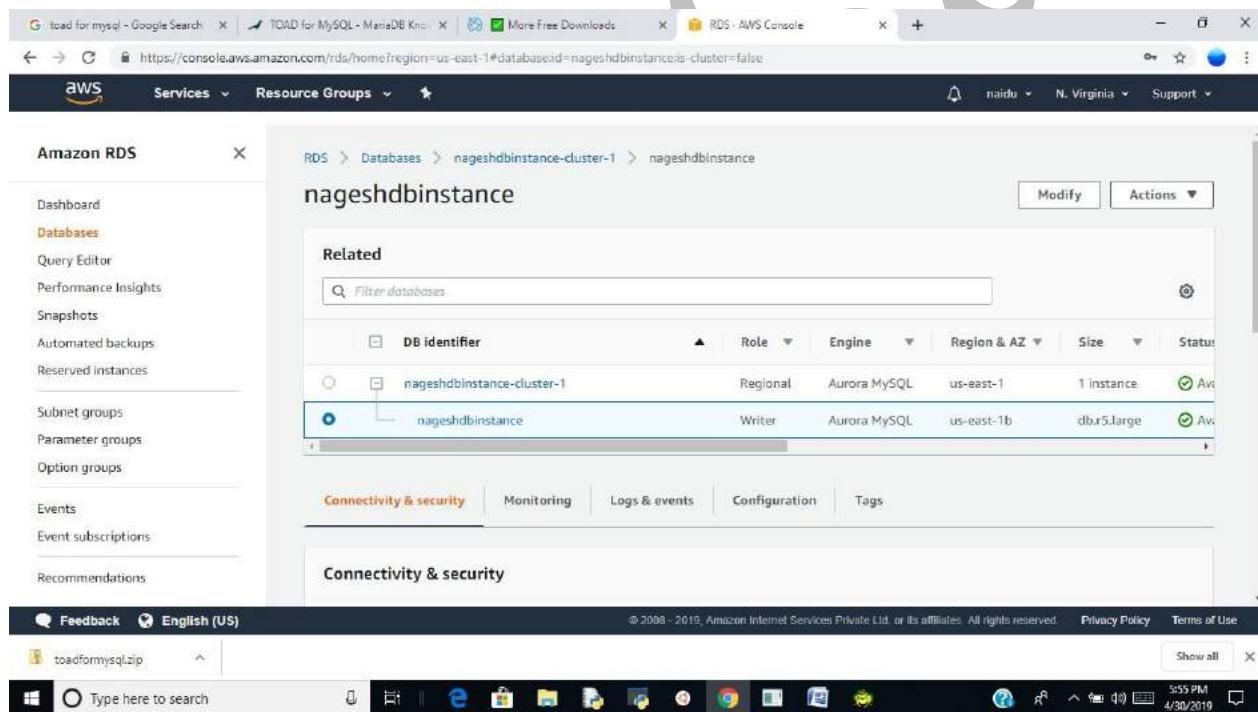
The screenshot shows the AWS RDS console with the same URL as the previous screenshot. The main content area is titled "Maintenance" and contains two sections: "Auto minor version upgrade" and "Maintenance window". Under "Auto minor version upgrade", the "Enable auto minor version upgrade" option is selected. Under "Maintenance window", the "No preference" option is selected. Below these sections is a "Deletion protection" section with a checkbox for "Enable deletion protection". At the bottom right of the page are "Cancel", "Previous", and "Create database" buttons. The status bar at the bottom indicates the date and time as 4/30/2019 5:18 PM.

Connection with DataBase

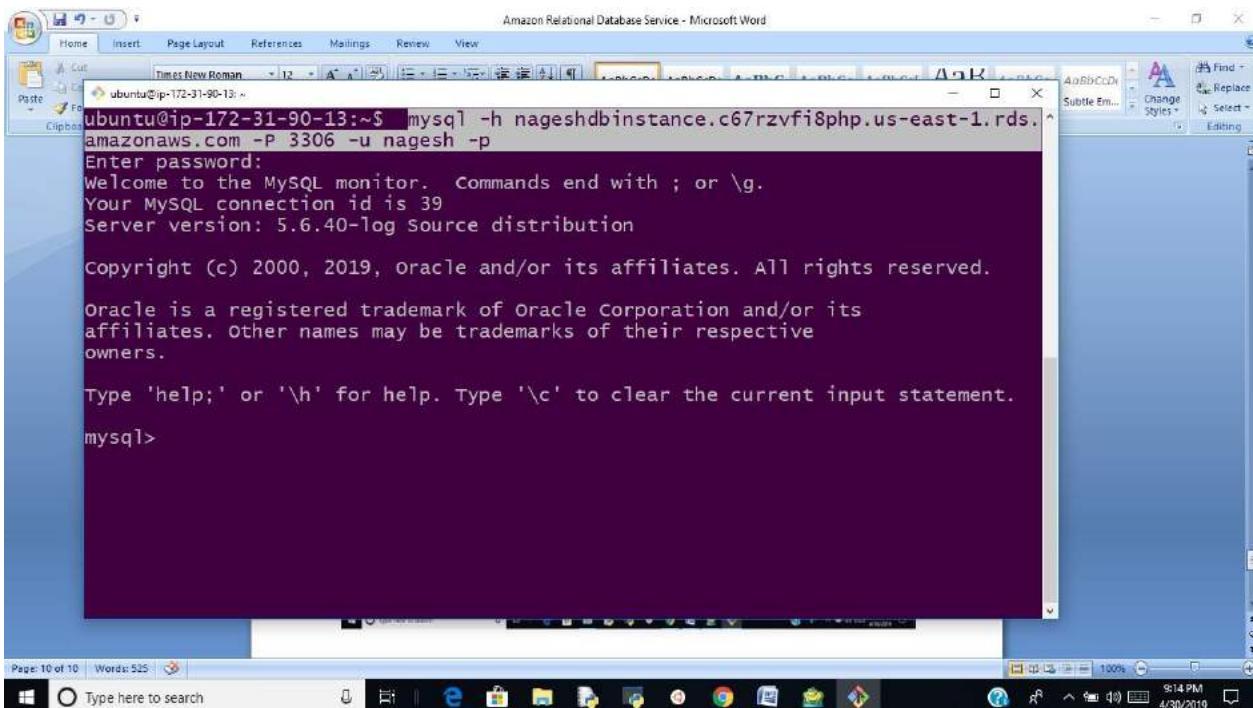
- Connecting to RDS MYSQL server you need mysql-client in your instance
- Launch one ec2 instance and install fallowing mysql client application

Sudo apt-get install mysql-client mysql-server

- Run the below command to connect to RDS mysq-server
mysql -h <hostname/Endpoint> -P <port> -u <master-username> -p
- To know the endpoint, port and database name information follow below steps
- Goto your RDS Console and select Databases option in left side panel
- Click on Database name



- Click on Connectivity & Security option yhen you find End point and port
- User name is the master username and password is for that user that is given by you when you create the RDS mysql Database
- Database name given by you when you created DB Engine.



- Then you are connected to MYSQL server
- For creating tables and see previous data first you switch your database (use databasename)



A screenshot of a Microsoft Word document window titled "Amazon Relational Database Service - Microsoft Word". The document contains a terminal session from a MySQL monitor on an Ubuntu system. The session shows connecting to a database instance, changing the database to "nagesh", and creating a table named "student" with two columns: "id" (integer) and "name" (varchar(10)). It also shows the description of the "student" table.

```
ubuntu@ip-172-31-90-13:~$ mysql -h nageshdbinstance.c67rzvfi8php.us-east-1.rds.amazonaws.com -P 3306 -u nagesh -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 39
Server version: 5.6.40-log Source distribution

Copyright (c) 2000, 2019, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> use nagesh
Database changed
mysql>
```

- Then you create tables and modify tables and retrieving data all mysql related tasks



A screenshot of a Microsoft Word document window titled "Amazon Relational Database Service - Microsoft Word". The document contains a terminal session from a MySQL monitor on an Ubuntu system. The session shows connecting to a database instance, changing the database to "nagesh", creating a table named "student" with two columns: "id" (integer) and "name" (varchar(10)), and then describing the "student" table.

```
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> use nagesh
Database changed
mysql> create table student(id integer,name varchar(10));
Query OK, 0 rows affected (0.02 sec)

mysql> desc table student;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that
corresponds to your MySQL server version for the right syntax to use near 'table
student' at line 1
mysql> desc student;
+-----+-----+-----+-----+-----+
| Field | Type   | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| id   | int(11) | YES  |     | NULL    |       |
| name | varchar(10)| YES |     | NULL    |       |
+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)

mysql>
```



Amazon Relational Database Service - Microsoft Word

```

Field | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| id   | int(11)  | YES  |     | NULL    |       |
| name | varchar(10)| YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)

mysql> insert into student values (535,'nagesh');
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that
corresponds to your MySQL server version for the right syntax to use near 'inser
into student values (535,'nagesh')' at line 1
mysql> insert into student values (535,'nagesh');
Query OK, 1 row affected (0.00 sec)

mysql> select * from student;
+----+-----+
| id | name |
+----+-----+
| 535 | nagesh |
+----+-----+
1 row in set (0.00 sec)

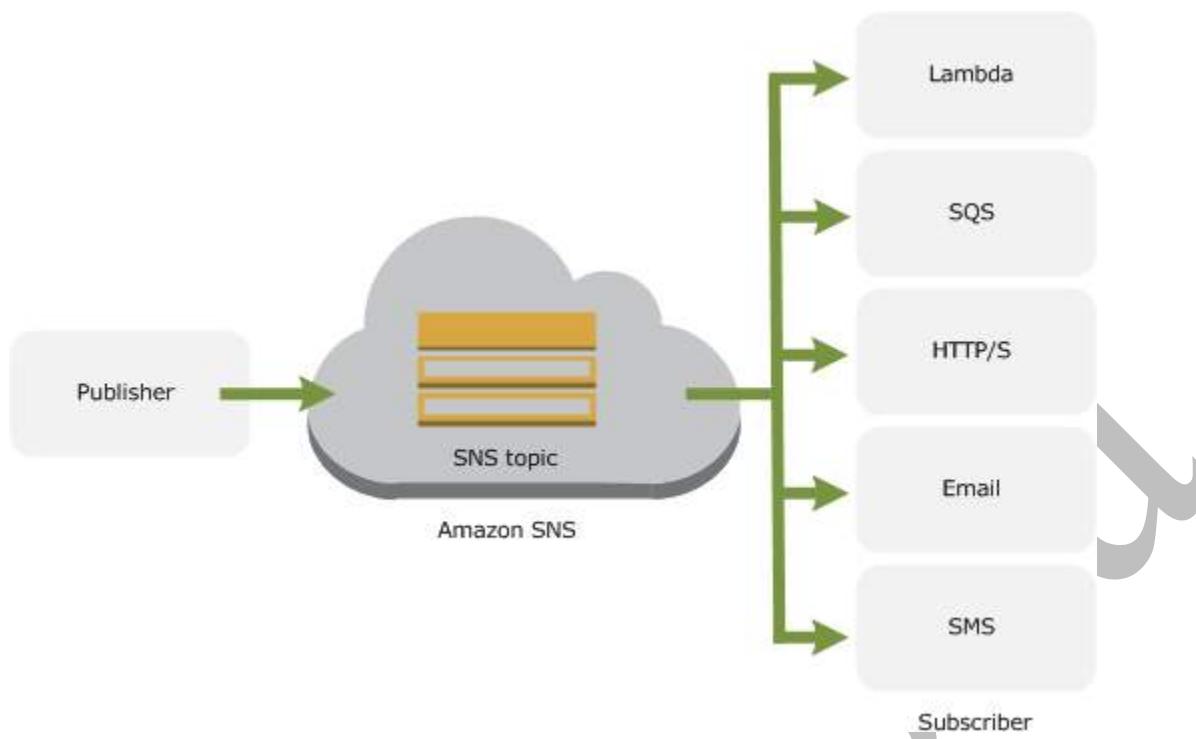
mysql> |

```

Page: 11 of 11 Words: 563

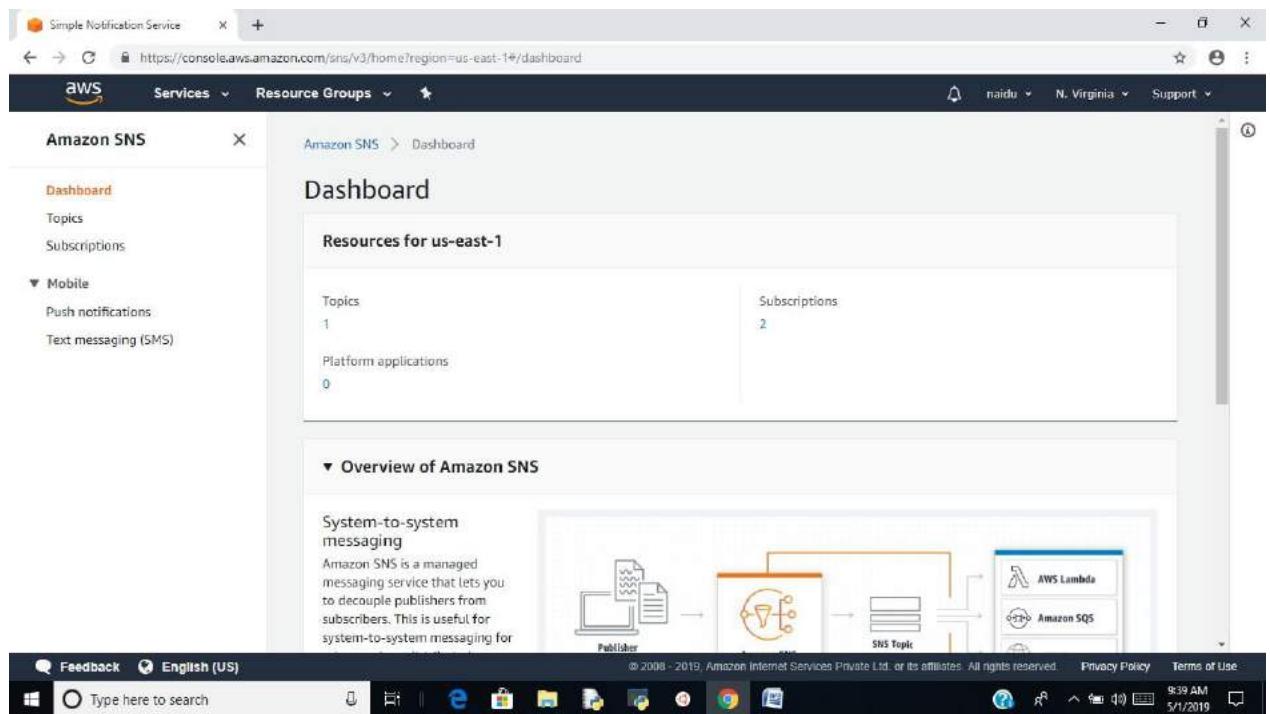
Amazon Simple Notification Service

Amazon Simple Notification Service (Amazon SNS) is a web service that coordinates and manages the delivery or sending of messages to subscribing endpoints or clients. In Amazon SNS, there are two types of clients—publishers and subscribers—also referred to as producers and consumers. Publishers communicate asynchronously with subscribers by producing and sending a message to a topic, which is a logical access point and communication channel. Subscribers (i.e., web servers, email addresses, Amazon SQS queues, AWS Lambda functions) consume or receive the message or notification over one of the supported protocols (i.e., Amazon SQS, HTTP/S, email, SMS, Lambda) when they are subscribed to the topic.

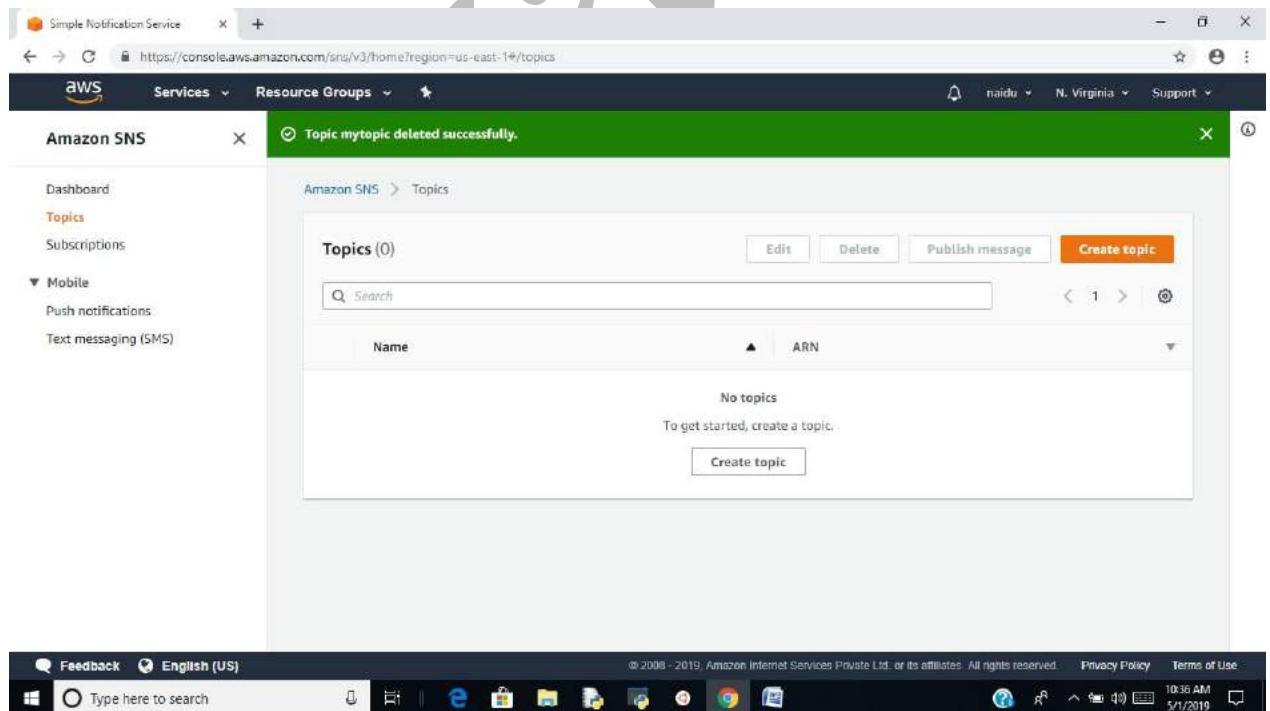


UseCase:

- Signin into the aws console and select the SNS service

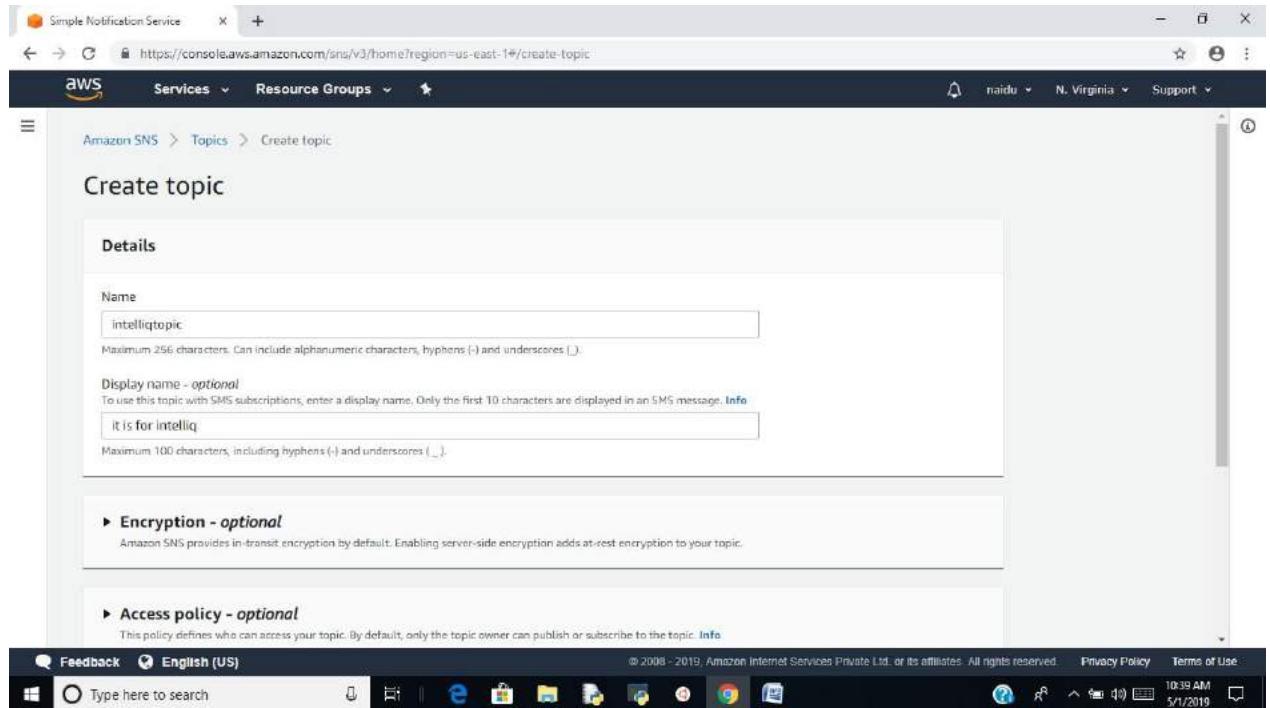


- Click on Topics on left side panel and click on create topic



- In Details section

- Name: enter name for topic
- Display Name: the name is displayed in subscription it is the optional



- Encryption: it is used to encrypt the notification messages and it is the optional
- Access Policy: it is used to assign policy or permissions to users whose access the SNS Topic. It is the Optional
- Delivery retry policy (HTTP/S) : It is used to deliver the Failure messages by using HTTP protocol. It is the optional
- Delivery status logging – *optional*: it is used to monitor the SNS Service with CloudWatch. It is the optional
- Click on create topic

Display name - optional
To use this topic with SMS subscriptions, enter a display name. Only the first 10 characters are displayed in an SMS message. [Info](#)
it is for intelliq
Maximum 100 characters, including hyphens (-) and underscores (_).

Encryption - optional
Amazon SNS provides in-transit encryption by default. Enabling server-side encryption adds at-rest encryption to your topic.

Access policy - optional
This policy defines who can access your topic. By default, only the topic owner can publish or subscribe to the topic. [Info](#)

Delivery retry policy (HTTP/S) - optional
The policy defines how Amazon SNS retries failed deliveries to HTTP/S endpoints. To modify the default settings, expand this section. [Info](#)

Delivery status logging - optional
These settings configure the logging of message delivery status to CloudWatch Logs. [Info](#)

[Cancel](#) [Create topic](#)

Topic intelliqtopic created successfully.
You can create subscriptions and send messages to them from this topic.

Name	intelliqtopic	Display name	it is for intelliq
ARN	arn:aws:sns:us-east-1:993595193866:intelliqtopic	Topic owner	993595193866

[Edit](#) [Delete](#) [Publish message](#)

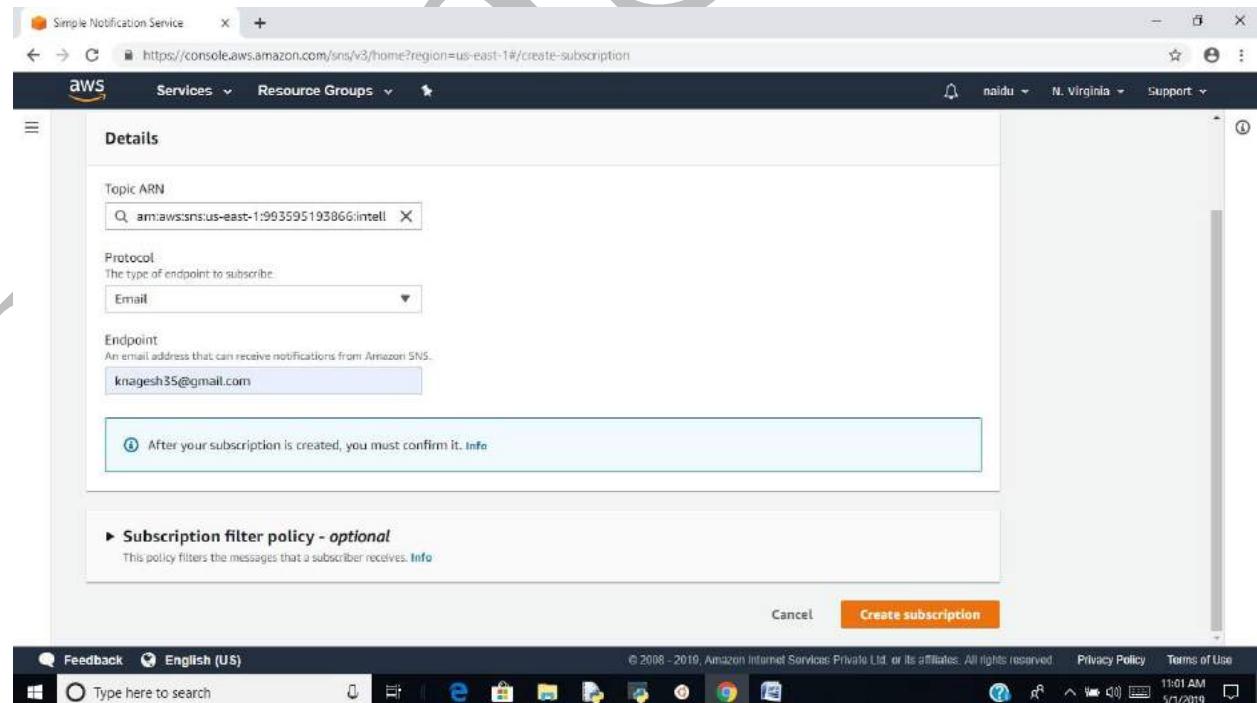
[Subscriptions](#) [Access policy](#) [Delivery retry policy \(HTTP/S\)](#) [Delivery status logging](#) [Encryption](#)

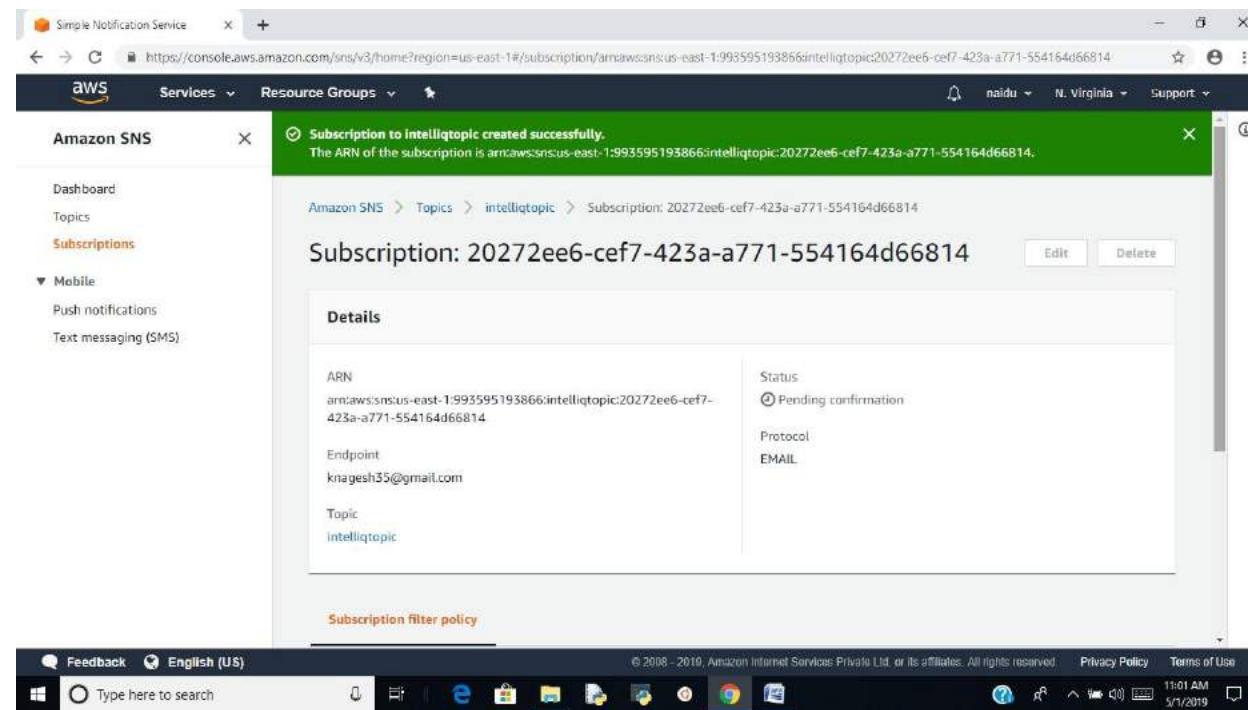
Subscriptions (0) [Edit](#) [Delete](#) [Request confirmation](#) [Confirm subscription](#) [Create subscription](#)

□ Click Create subscription

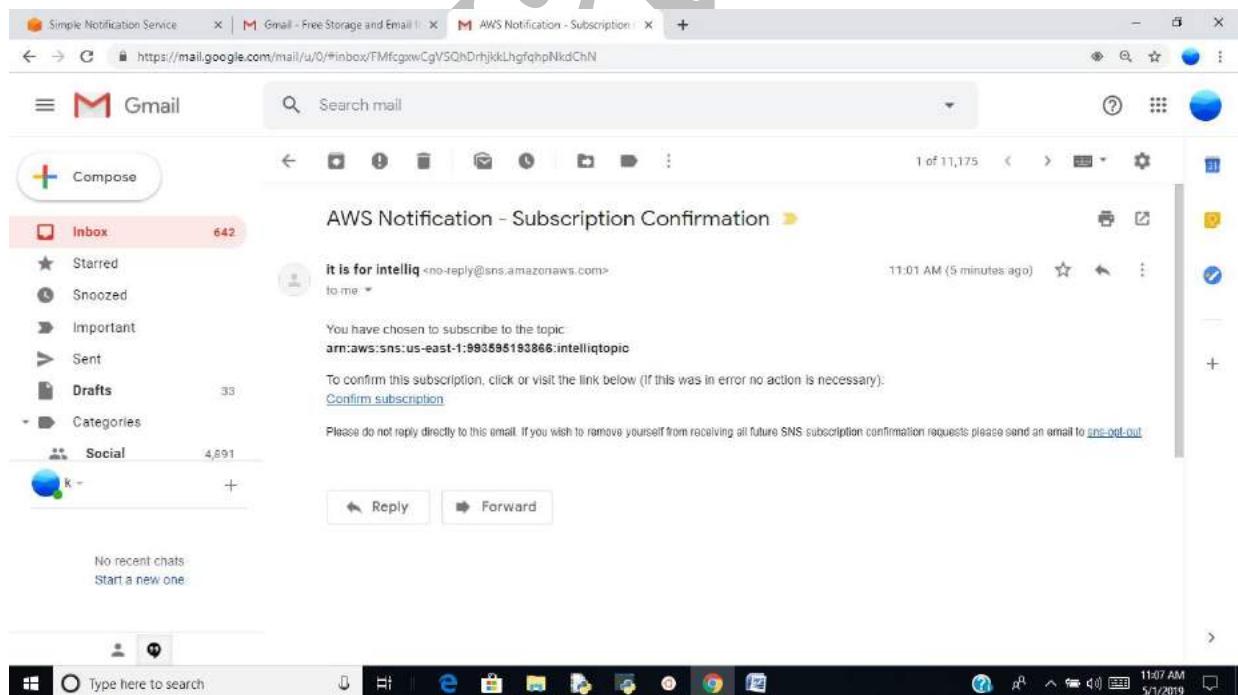
- Topic ARN: it is unique id of your SNS Topic. It is the optional
- Protocol: it is the type of End point Subscription

- HTTP
- HTTPS
- Email
- Email-JSON
- Amazon SQS
- AWS Lambda
- Platform application endpoint
- SMS
- Here we select Email or SMS anything which Endpoint do you want use. I select Email
- Endpoint: enter your email address
- Then click on create subscription

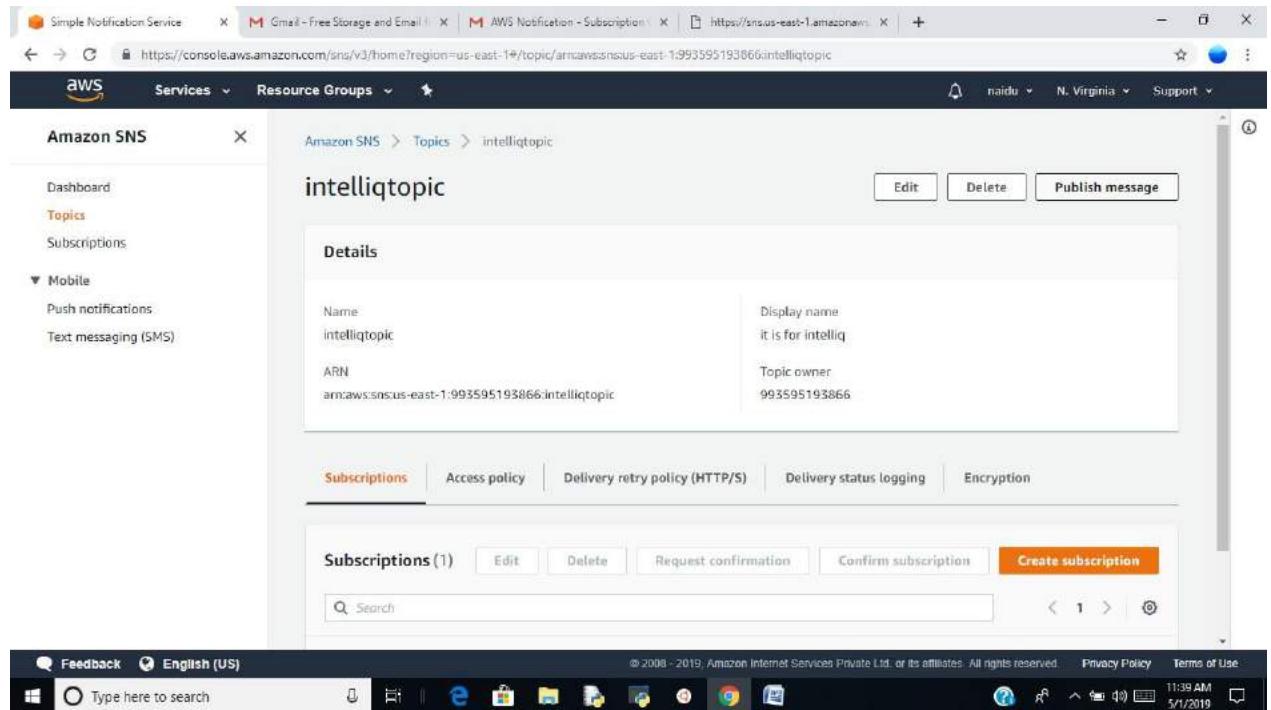




- Open our Email and Confirm SNS subscription



- Do you want to check the SNS notifications then first select topic and click on publish message on upper right side panel topic



- Enter Subject: subject is the message subject. It is the optional
- Time to Live : This setting applies only to mobile application endpoints. The number of seconds that the push notification service has to deliver the message to the endpoint. It is the optional
- In Message Body click on Identical payload for all delivery protocol to use same protocol for all Endpoints.
- Write message body and click on publish message

The screenshot shows the AWS Simple Notification Service (SNS) interface for publishing a message to a topic. The top navigation bar includes tabs for Simple Notification Service, Gmail - Free Storage and Email, AWS Notification - Subscription, and the current page, https://sns.us-east-1.amazonaws.com. The main content area is titled "Publish message to topic".
Message details:
Topic ARN: arn:aws:sns:us-east-1:993595193866:intelijtopic
Subject - optional: Enter message subject (Maximum 100 printable ASCII characters)
Time to Live (TTL) - optional: This setting applies only to mobile application endpoints. The number of seconds that the push notification service has to deliver the message to the endpoint. Info
Message body:
Message structure: A dropdown menu showing "Message structure" is selected.
Payload type:
 Identical payload for all delivery protocols. The same payload is sent to endpoints subscribed to the topic, regardless of their delivery protocol.
 Custom payload for each delivery protocol. Different payloads are sent to endpoints subscribed to the topic, based on their delivery protocol.
Message body to send to the endpoint: A text input field containing "1 Enter new message".
The bottom of the window shows the Windows taskbar with various pinned icons and the system clock indicating 11:45 AM on 5/1/2019.

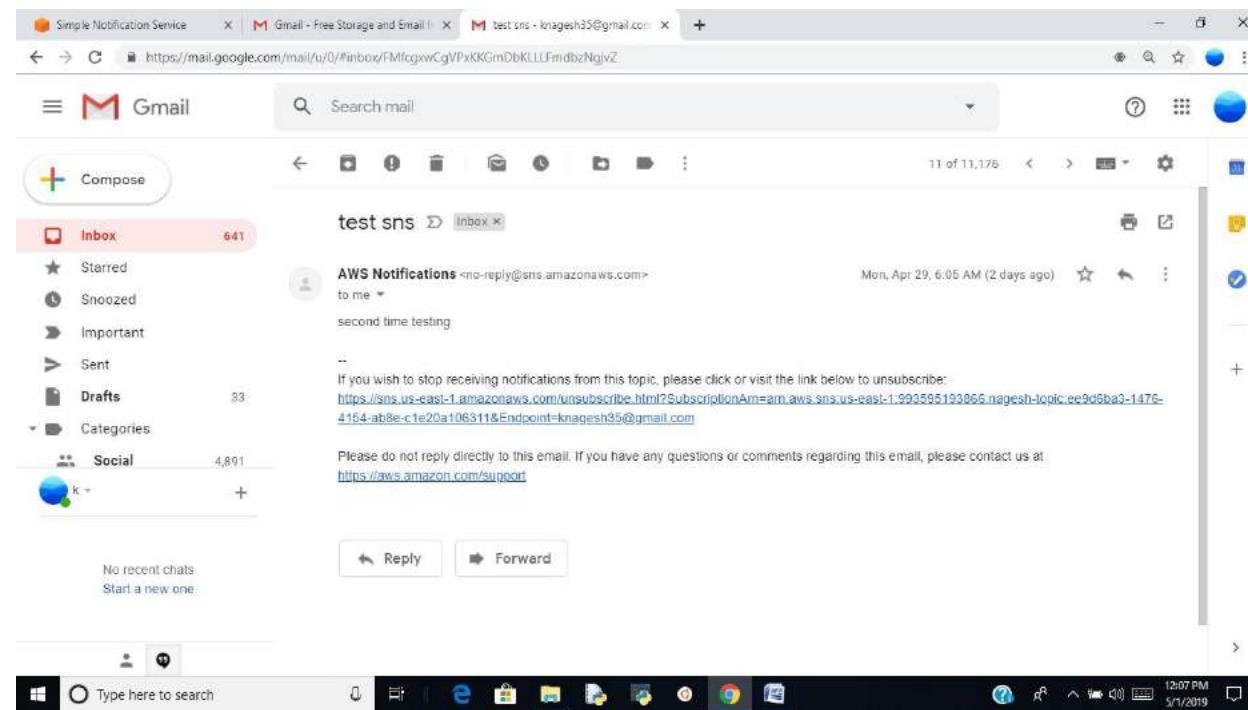
The screenshot shows the AWS SNS console interface. At the top, there are tabs for Simple Notification Service, Gmail - Free Storage and Email, AWS Notification - Subscription, and a specific topic URL. The main area is titled "Message attributes". It includes fields for "Type" (dropdown menu), "Name" (text input), and "Value" (text input containing "value or [value1, value2]"). Below these are buttons for "Remove" and "Add another attribute". At the bottom right are "Cancel" and "Publish message" buttons.

Message published to topic intelliqtopic successfully.
Message ID: 787e5486-c278-5328-a1aa-01c5f98f817f

The browser address bar shows the URL: https://sns.us-east-1.amazonaws.com/?region=us-east-1#/topic/armaws sns us-east-1:993595193866:intelliqtopic

The operating system taskbar at the bottom shows various open applications like File Explorer, Edge, and Google Chrome.

□ Check your Email



- You can use these SNS Service in where you applications and another AWS services like CloudWatch, Autoscaling....etc by selecting SNS Topic.

Amazon SES

Welcome to the Amazon Simple Email Service (Amazon SES) Developer Guide. Amazon SES is an email platform that provides an easy, cost-effective way for you to send and receive email using your own email addresses and domains.

Step 1: Sign up for AWS

Before you can use Amazon SES, you need to sign up for AWS. When you sign up for AWS, your account is automatically signed up for all AWS services.

Step 2: Verify your email address

Before you can send email from your email address through Amazon SES, you need to show Amazon SES that you own the email address by verifying it.

Step 3: Send your first email

You can send an email simply by using the Amazon SES console. As a new user, your account is in a test environment called the sandbox, so you can only send email to and from email addresses that you have verified.

Step 4: Consider how you will handle bounces and complaints

Before the next step, you need to think about how you will handle bounces and complaints. If you are sending to a small number of recipients, your process can be as simple as examining the bounce and complaint feedback that you receive by email, and then removing those recipients from your mailing list.

Step 5: Move out of the Amazon SES sandbox

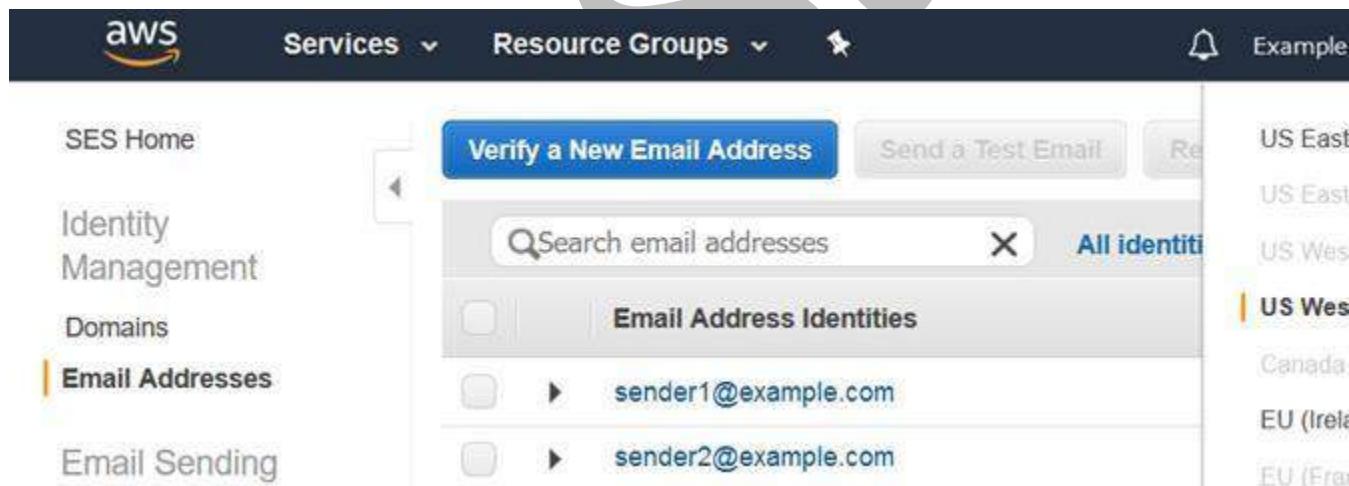
To be able to send emails to unverified email addresses and to raise the number of emails you can send per day and how fast you can send them, your account needs to be moved out of the sandbox. This process involves opening an SES Sending Limits Increase case in Support Center.

Verifying an Email Address Using the Amazon SES Console

Complete the procedure in this section to verify an email address using the Amazon SES console.

To verify an email address using the Amazon SES console

1. Sign in to the AWS Management Console and open the Amazon SES console at <https://console.aws.amazon.com/ses/>.
2. In the console, use the region selector to choose the AWS Region where want to verify the email address, as shown in the following image.



Note

To verify an email address for use in more than one region, repeat the procedure in this section for each region.

3. In the navigation pane, under **Identity Management**, choose **Email Addresses**.
4. Choose **Verify a New Email Address**.
5. In the **Verify a New Email Address** dialog box, type your email address in the **Email Address** field, and then choose **Verify This Email Address**.

6. Check the inbox for the email address that you're verifying. You'll receive a message with the following subject line: "Amazon Web Services - Email Address Verification Request in region `RegionName`," where `RegionName` is the name of the AWS Region you selected in step 2.

Click the link in the message.

Note

The link in the verification message expires 24 hours after the message was sent. If 24 hours have passed since you received the verification email, repeat steps 1–5 to receive a verification email with a valid link.

7. In the Amazon SES console, under **Identity Management**, choose **Email Addresses**. In the list of email addresses, locate the email address you're verifying. If the email address was verified, the value in the **Status** column is "verified".

Send an Email Using the Amazon SES Console

The easiest way to send an email with Amazon SES is to use the Amazon SES console. Because the console requires you to manually enter information, you typically only use it to send test emails. After you get started with Amazon SES, you will most likely send your emails using either the Amazon SES SMTP interface or API, but the console is useful for monitoring your sending activity.

To send an email message from the Amazon SES console

1. Sign in to the AWS Management Console and open the Amazon SES console at <https://console.aws.amazon.com/ses/>.

Note

If you are not currently signed in to your AWS account, this link takes you to a sign-in page. After you sign in, you are directed to the Amazon SES console.

2. In the navigation pane on the left side of the Amazon SES console, under **Identity Management**, choose **Email Addresses** to view the email address that you verified in [Verifying Email Addresses in Amazon SES](#).

3. In the list of identities, check the box next to email address that you have verified.
4. Choose **Send a Test Email**.
5. For **Send Test Email**, choose the **Email Format**. The two choices are as follows:
 - Formatted—This is the simplest option. Choose this option if you simply want to type the text of your message into the **Body** text box. When you send the email, Amazon SES puts the text into email format for you.
 - Raw—Choose this option if you want to send a more complex message, such as a message that includes HTML or an attachment. Because of this flexibility, you need to format the message, as described in [Sending Raw Email Using the Amazon SES API](#), yourself, and then paste the entire formatted message, including the headers, into the **Body** text box. You can use the following example, which contains HTML, to send a test email using the **Raw** email format. Copy and paste this message in its entirety into the **Body** text box. Ensure that there is not a blank line between the `MIME-Version` header and the `Content-Type` header; a blank line between these two lines causes the email to be formatted as plain text instead of HTML.

```
Subject: Amazon SES Raw Email Test
MIME-Version: 1.0
Content-Type: text/html

<!DOCTYPE html>
<html>
<body>
<h1>This text should be large, because it is formatted as a header in HTML.</h1>
<p>Here is a formatted link: <a href="https://docs.aws.amazon.com/ses/latest/DeveloperGuide>Welcome.html">Amazon Simple Email Service Developer Guide</a>.</p>
</body>
</html>
```

6. For **Send Test Email**, fill out the rest of the fields. If you are still in the Amazon SES sandbox, make sure that the address in the **To** field is a verified email address.
7. Choose **Send Test Email**.
8. Sign in to the email client of the address you sent the email to. You will find the message that you sent.

Email-Receiving Process

When Amazon SES receives an email for your domain, the following events occur:

1. Amazon SES first looks at the IP address of the sender. Amazon SES allows the mail to pass this stage unless:
 - The IP address is in your block list.
 - The IP address is in the Amazon SES block list and not on your allow list.
2. Amazon SES examines your active receipt rule set to determine whether any of your receipt rules contain a condition that matches any of the incoming email's recipients.
3. If there aren't any matches, Amazon SES rejects the mail. Otherwise, Amazon SES accepts the mail.
4. If Amazon SES accepts the mail, it evaluates your active receipt rule set. All of the receipt rules that match at least one of the recipient conditions are applied in the order that they are defined, unless an action or a receipt rule explicitly terminates evaluation of the receipt rule set.

Getting Started Receiving Email with Amazon SES

In this tutorial, you'll create an AWS account, register a domain using Amazon Route 53, and configure Amazon Simple Email Service to deliver all email sent to your domain to an Amazon Simple Storage Service bucket.

Step 1: Before You Begin

Before you start this tutorial, sign up for an AWS account (if you don't already have one), and use [Amazon Route 53](#) to register the domain you want to use to receive email.

Sign Up

If you already have an AWS account, you can skip this section.

To create an AWS account

1. Go to <https://console.aws.amazon.com/ses/>, and then choose **Get Started with Amazon SES**.

2. On the **Create an AWS Account** page, complete the required fields and follow the on-screen instructions to create a new account.

Register a Domain using Route 53

This tutorial assumes that you're using a domain that you registered using Route 53. You can also use a domain that you registered using another service, but the procedures for verifying your domain will differ from those shown in this tutorial.

Step 2: Verify Your Domain

Before you can configure Amazon SES to receive email for your domain, you must prove that you own the domain. You can verify any domain that you own, but it is easier to verify domains that you registered using Route 53.

To verify a domain with Amazon SES

1. Open the Amazon SES console at <https://console.aws.amazon.com/ses/>.

Note

To complete the procedure in this section, sign in to the AWS Management Console using the same AWS account you used when you registered your domain with Route 53.

2. In the navigation pane, under **Identity Management**, choose **Domains**.
3. Choose **Verify a New Domain**.
4. On the **Verify a New Domain** dialog box, for **Domain**, type the name of the domain that you registered using Route 53, and then choose **Verify This Domain**.
5. On the **Verify a New Domain** dialog box, choose **Use Route 53**
6. On the **Use Route 53** dialog box, select **Domain Verification Record** and **Email Receiving Record**. Then, under **Hosted Zones**, select the name of the Hosted Zone you want to use. If you haven't made any changes to the domain you registered using Route 53, there should only be one option available in the **Hosted Zones** section.
7. Choose **Create Record Sets**. You'll return to the list of domains.
8. Wait five minutes, and then choose the refresh () button. Confirm that the value in the **Status** column is **verified**. If the status is **pending verification**, wait a few more

minutes, and then refresh the list again. Repeat this process until the domain's status is **verified**.

Step 4: Send a Test Email

Now that you've verified and configured your domain, you can send an email to test your domain's ability to receive email.

To send a test email, use an email account that you know is capable of sending email, such as your personal email address. Send a test message to any email address on your verified domain. For example, if your domain is *example.com*, you can send an email to *test@example.com* or *abc123@example.com* (or any other address on the *example.com* domain).

Step 5: View the Received Email

After you send a test message to an address on your domain, you can retrieve it from your Amazon S3 bucket and view its contents.

To view a message that you received through Amazon SES

1. Open the Amazon S3 console at <https://console.aws.amazon.com/s3/>.
2. In the Amazon S3 console, choose the bucket you created in [Step 3: Set up a Receipt Rule](#).
3. In the Amazon S3 bucket, find the email you received. The name of the email is a unique string of letters and numbers.

Note

The bucket may also contain a file named `AMAZON_SES_SETUP_NOTIFICATION`. You can ignore or delete this file.

4. Select the check box next to the name of the file. On the **Actions** menu, choose **Download**.
5. Open the folder on your computer that contains the file you downloaded in the preceding step. There are several ways to view the downloaded message, including the following:

- Open the file in a text editor and read its contents directly. Depending on the method you used to send the email, part of the message may be encoded. If part of the message is encoded, you'll need to decode them manually (for example, by using a base64 decoder).
- Add the `.eml` extension to the end of the file name, and then open the file using an email client such as Microsoft Outlook or Mozilla Thunderbird. Most email clients will automatically decode the encoded parts of a message, and will display things like HTML formatting and file attachments.

Step 6: Clean Up

After you complete this tutorial, you can clean up the resources you created to avoid incurring additional charges.

Amazon Simple Queue Service

Amazon Simple Queue Service (Amazon SQS) offers a secure, durable, and available hosted queue that lets you integrate and decouple distributed software systems and components.

This section helps you become more familiar with Amazon SQS by showing you how to manage queues and messages using the AWS Management Console

Creating an Amazon SQS Queue

The first and most common Amazon SQS task is creating queues. In this tutorial you'll learn how to create and configure a queue.

AWS Management Console

1. Sign in to the [Amazon SQS console](#).
2. Choose **Create New Queue**.
3. On the **Create New Queue** page, ensure that you're in the correct region and then type the **Queue Name**.

Note

The name of a FIFO queue must end with the `.fifo` suffix.

4. **Standard** is selected by default. Choose **FIFO**.
5. Create your queue.
 - To create your queue with the default parameters, choose **Quick-Create Queue**.
 - To configure your queue's parameters, choose **Configure Queue**. When you finish configuring the parameters, choose **Create Queue**. For more information about creating a queue with SSE.

The following example shows the **Content-Based Deduplication** parameter specific to FIFO queues.

Queue Attributes	
Default Visibility Timeout	30 <input type="button" value="seconds"/> Value must be between 0 seconds and 3,155,692,600 seconds.
Message Retention Period	4 <input type="button" value="days"/> Value must be between 1 minute and 1,209,600 minutes.
Maximum Message Size	256 <input type="button" value="KB"/> Value must be between 1 and 256,000 KB.
Delivery Delay	0 <input type="button" value="seconds"/> Value must be between 0 seconds and 3,155,692,600 seconds.
Receive Message Wait Time	0 <input type="button" value="seconds"/> Value must be between 0 and 20 seconds.
Content-Based Deduplication <input type="checkbox"/>	
Dead Letter Queue Settings	
Use Redrive Policy	<input type="checkbox"/>
Dead Letter Queue	<input type="text"/> Value must be an existing queue name.
Maximum Receives	<input type="text"/> Value must be between 1 and 1000.

Your new queue is created and selected in the queue list.

Note

When you create a queue, it can take a short time for the queue to propagate throughout Amazon SQS.

The **Queue Type** column helps you distinguish standard queues from FIFO queues at a glance. For a FIFO queue, the **Content-Based Deduplication** column displays whether you have enabled exactly-once processing.

Name	Queue Type	Content-Based Deduplication	Messages Available
MyQueue	Standard Queue	N/A	0
MyQueue.fifo	FIFO Queue	Disabled	0

Your queue's **Name**, **URL**, and **ARN** are displayed on the **Details** tab.

Name: MyQueue.fifo

URL: https://sqs.us-west-2.amazonaws.com/XXXXXXXXXX/MyQueue.fifo

ARN: arn:aws:sqs:us-west-2:XXXXXXXXXX:MyQueue.fifo

Listing All Amazon SQS Queues in a Region

When you create a queue, it can take a short time for the queue to propagate throughout Amazon SQS. In this tutorial you learn how to confirm your queue's existence by listing all queues in the current region.

AWS Management Console

1. Sign in to the [Amazon SQS console](#).
2. Your queues in the current region are listed.

The **Queue Type** column helps you distinguish standard queues from FIFO queues at a glance. For a FIFO queue, the **Content-Based Deduplication** column displays whether you have enabled exactly-once processing.

Name	Queue Type	Content-Based Deduplication	Messages Available
MyQueue	Standard Queue	N/A	0
MyQueue fifo	FIFO Queue	Disabled	0

Your queue's **Name**, **URL**, and **ARN** are displayed on the **Details** tab.

Name: MyQueue fifo

URL: <https://sqs.us-west-2.amazonaws.com/123456789012/MyQueue fifo>

ARN: arn:aws:sqs:us-west-2:123456789012:MyQueue fifo

Adding Permissions to an Amazon SQS Queue

You can specify to whom you allow (or explicitly deny) the ability to interact with your queue in specific ways by adding permissions to a queue. The following example shows how to add the permission for anyone to get a queue's URL.

Note

An Amazon SQS policy can have a maximum of 7 actions.

AWS Management Console

1. Sign in to the [Amazon SQS console](#).
2. From the queue list, select a queue.

Name	Queue Type
MyQueue	Standard
MyQueue fifo	FIFO

3. From **Queue Actions**, select **Add a Permission**.



The **Add a Permission** dialog box is displayed.

4. In this example, you allow anyone to get the queue's URL:

Add a Permission to MyQueue.fifo

Permissions enable you to control which operations a user can perform on a queue. [Click here](#) to learn about access control concepts.

1 Effect Allow Deny

2 Principal aws account number(s) Everybody (*)

Use commas between multiple values.

3 Actions All SQS Actions (SQS:*)

- AddPermission
- ChangeMessageVisibility
- DeleteMessage
- DeleteQueue
- GetQueueAttributes
- GetQueueUrl
- ListDeadLetterSourceQueues
- PurgeQueue
- ReceiveMessage
- RemovePermission
- SendMessage
- SetQueueAttributes

4 Cancel

- ① Ensure that next to **Effect**, **Allow** is selected.
- ② Next to **Principal**, check the **Everybody** box.
- ③ From the **Actions** drop-down list, select **GetQueueUrl** box.

④ Choose **Add Permission**.

The permission is added to the queue.

Your queue's policy **Effect**, **Principals**, **Actions**, and **Conditions** are displayed on your queue's **Permissions** tab.

Effect	Principals	Actions	Conditions	
Allow	• Everybody (*)	• SQS:GetQueueUrl	None	

Sending a Message to an Amazon SQS Queue

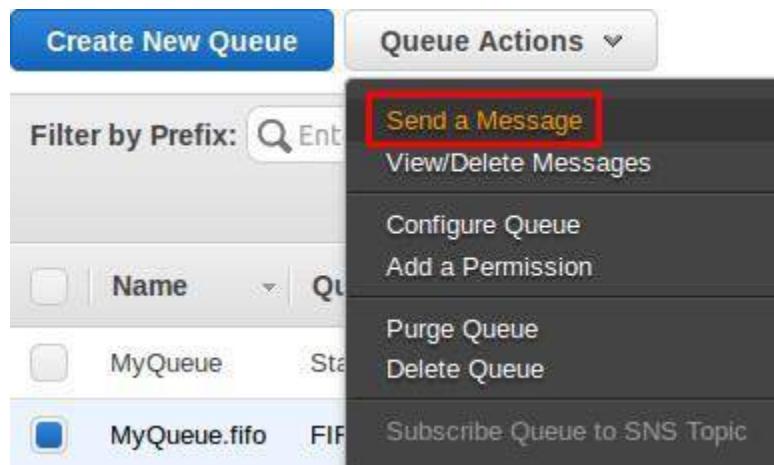
After you create your queue, you can send a message to it. The following example shows sending a message to an existing queue.

AWS Management Console

1. Sign in to the [Amazon SQS console](#).
2. From the queue list, select a queue.



3. From **Queue Actions**, select **Send a Message**.



The **Send a Message to *QueueName*** dialog box is displayed.

The following example shows the **Message Group ID** and **Message Deduplication ID** parameters specific to FIFO queues (content-based deduplication is disabled).

Send a Message to MyQueue fifo

Message Body **Message Attributes**

Enter the text of a message you want to send.

This is my message text.

Message Group ID Type a FIFO message group (required).

Message Deduplication ID Type a deduplication token (required).

Cancel **Send Message**

4. To send a message to a FIFO queue, type the **Message Body**, the **Message Group ID** `MyMessageGroupId1234567890`, and the **Message Deduplication ID** `MyMessageDeduplicationId1234567890`, and then choose **Send Message**.

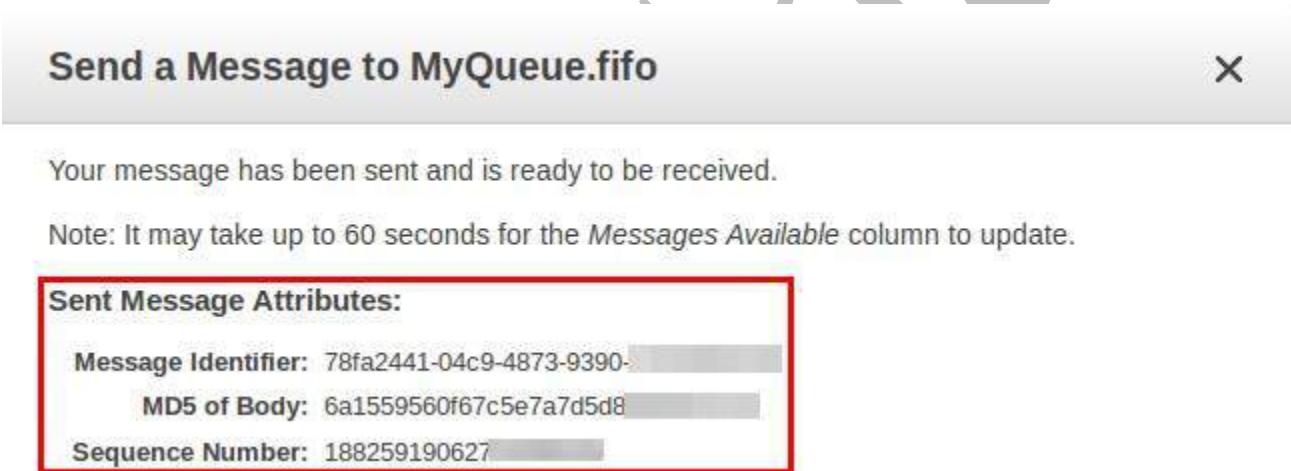
Note

The message group ID is always required. However, if content-based deduplication is enabled, the message deduplication ID is optional.



Your message is sent and the **Send a Message to *QueueName*** dialog box is displayed, showing the attributes of the sent message.

The following example shows the **Sequence Number** attribute specific to FIFO queues.



5. Choose **Close**.

Receiving and Deleting a Message from an Amazon SQS Queue

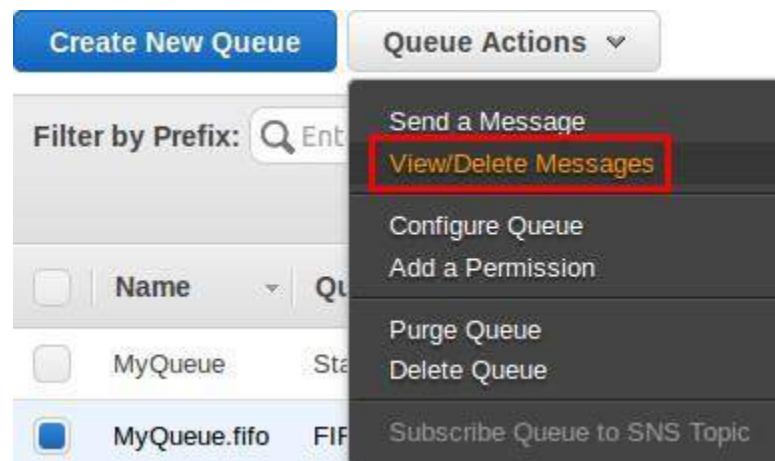
After you send a message into a queue, you can consume it from the queue. When you request a message from a queue, you can't specify which message to get. Instead, you specify the maximum number of messages (up to 10) that you want to get.

AWS Management Console

1. Sign in to the [Amazon SQS console](#).
2. From the queue list, select a queue.



3. From **Queue Actions**, select **View/Delete Messages**.

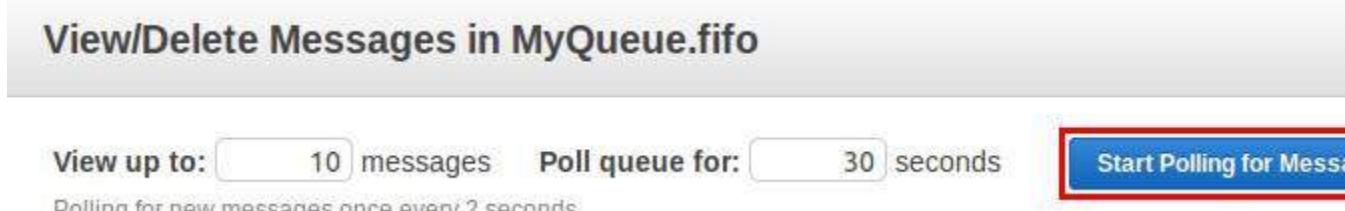


The **View/Delete Messages in QueueName** dialog box is displayed.

Note

The first time you take this action, an information screen is displayed. To hide the screen, check the **Don't show this again** checkbox.

4. Choose **Start Polling for messages**.



Amazon SQS begins to poll the messages in the queue. The dialog box displays a message from the queue. A progress bar at the bottom of the dialog box displays the status of the message's visibility timeout.

The following example shows the **Message Group ID**, **Message Deduplication ID**, and **Sequence Number** columns specific to FIFO queues.

View/Delete Messages in MyQueue.fifo

View up to: messages **Poll queue for:** seconds Polling for Message

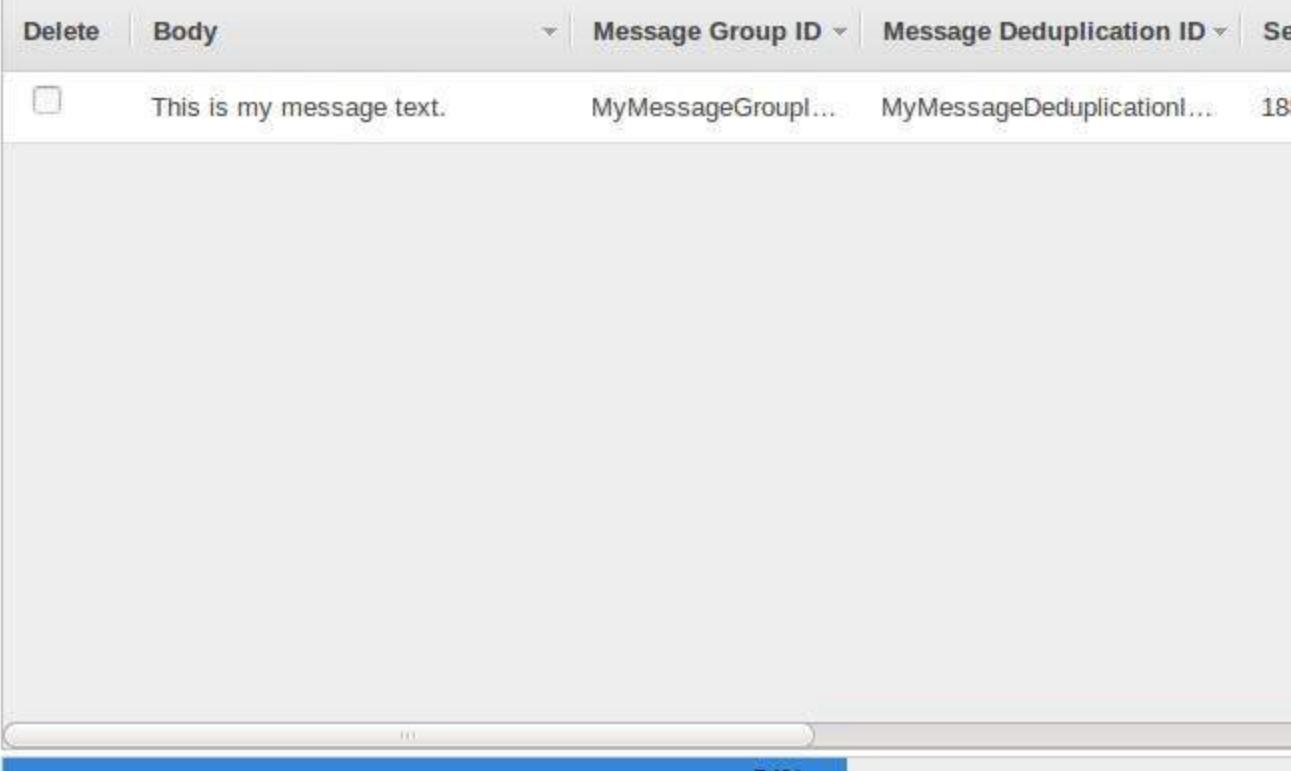
Polling for new messages once every 2 seconds.

Delete	Body	Message Group ID	Message Deduplication ID	Sequence Number
<input type="checkbox"/>	This is my message text.	MyMessageGroup1...	MyMessageDuplication1...	18

54%

Polling the queue at 0.6 receives/second. Stopping in 13.7 seconds. Messages shown above are currently hidden.

Close



Note

When the progress bar is filled in, the visibility timeout expires and the message becomes visible to consumers.

5. *Before* the visibility timeout expires, select the message that you want to delete and then choose **Delete 1 Message**.

View/Delete Messages in MyQueue.fifo

View up to: messages **Poll queue for:** seconds **Start Polling for Messages**

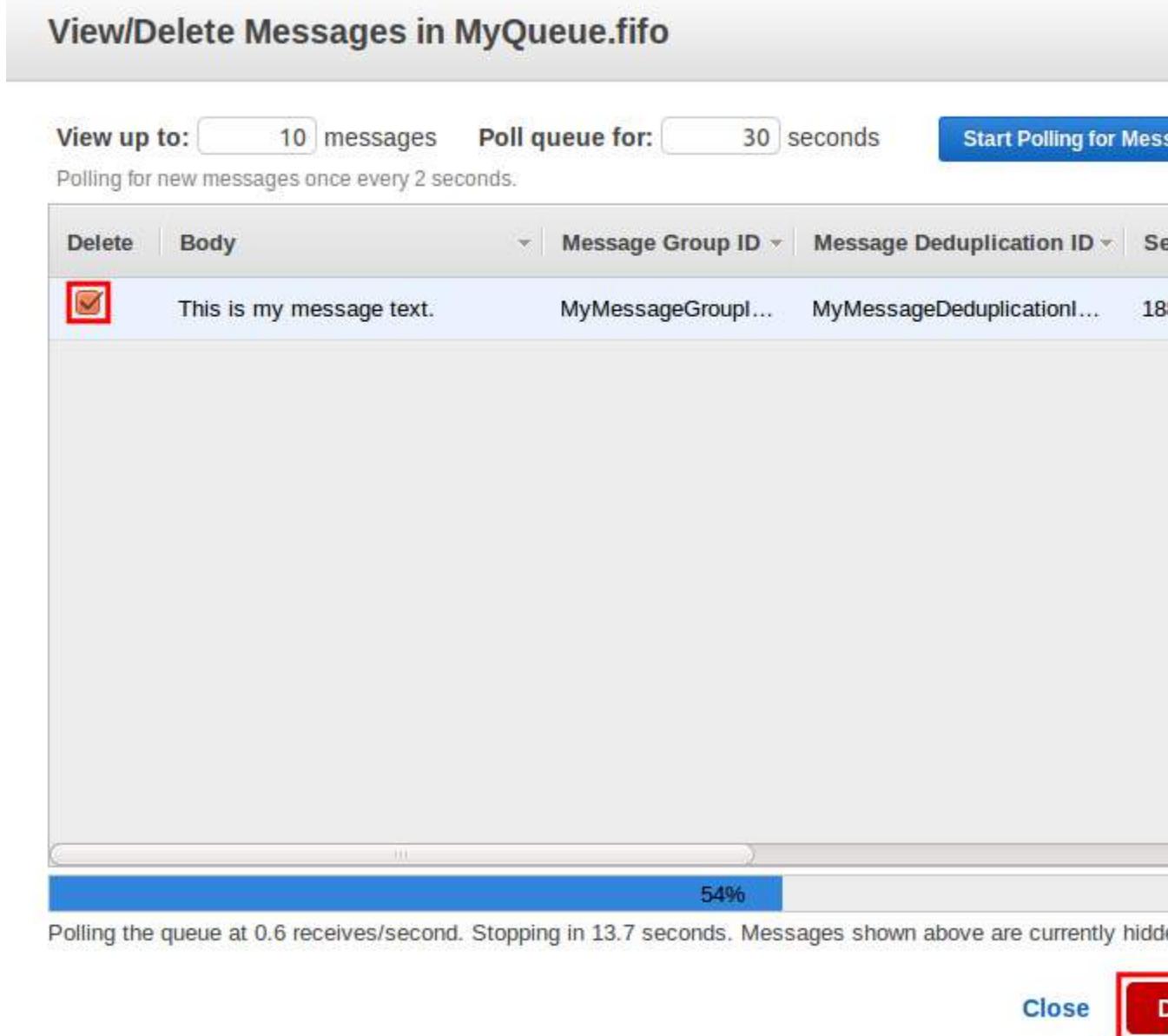
Polling for new messages once every 2 seconds.

Delete	Body	Message Group ID	Message Deduplication ID	Sequence Number
<input checked="" type="checkbox"/>	This is my message text.	MyMessageGroup1...	MyMessageDeduplication1...	18

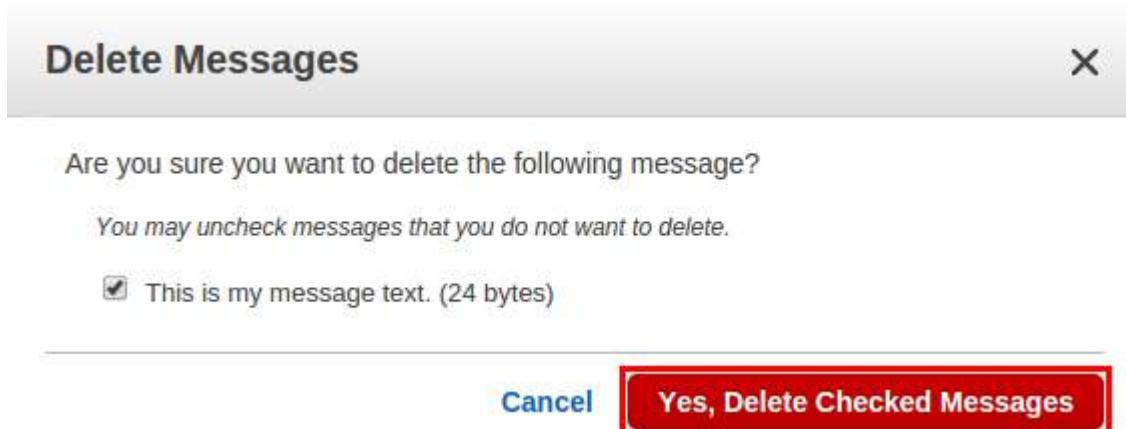
54%

Polling the queue at 0.6 receives/second. Stopping in 13.7 seconds. Messages shown above are currently hidden.

Close



6. In the **Delete Messages** dialog box, confirm that the message you want to delete is checked and choose **Yes, Delete Checked Messages**.



The selected message is deleted.

7. Select **Close**.

Deleting an Amazon SQS Queue

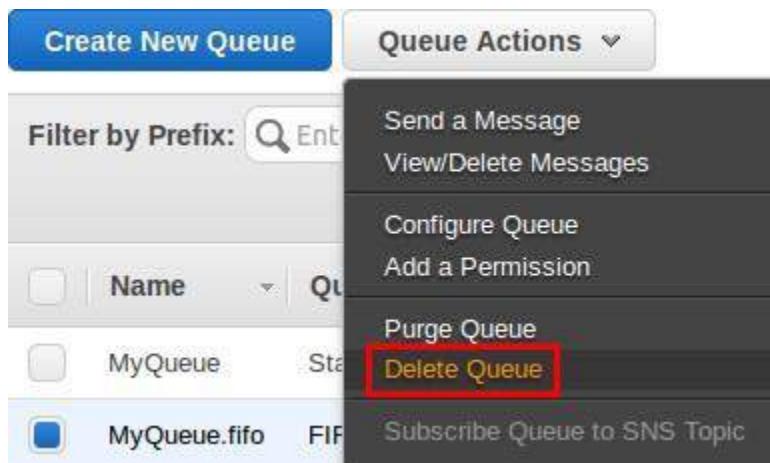
If you don't use an Amazon SQS queue (and don't foresee using it in the near future), it is a best practice to delete it from Amazon SQS. In this tutorial you'll learn how to delete a queue.

AWS Management Console

1. Sign in to the [Amazon SQS console](#).
2. From the queue list, select a queue.



3. From **Queue Actions**, select **Delete Queue**.



The **Delete Queues** dialog box is displayed.

Delete Queues

Are you **sure** you want to delete the following queue, and any messages left in it?

- MyQueue fifo - contains no messages.

Cancel

Yes, Delete

4. Choose **Yes, Delete Queue**.

The queue is deleted.

AWS Cloud Formation

AWS CloudFormation is a service that helps you model and set up your Amazon Web Services resources so that you can spend less time managing those resources and more time focusing on your applications that run in AWS. You create a template that describes all the AWS resources that you want (like Amazon EC2 instances or Amazon RDS DB instances), and AWS CloudFormation takes care of provisioning and configuring those resources for you. You don't need to individually create and configure AWS resources and figure out what's dependent on what; AWS CloudFormation handles all of that.

When you use AWS CloudFormation, you work with *templates* and *stacks*. You create templates to describe your AWS resources and their properties. Whenever you create a stack, AWS CloudFormation provisions the resources that are described in your template.

Templates

An AWS CloudFormation template is a JSON or YAML formatted text file. You can save these files with any extension, such as .json, .yaml, .template, or .txt. AWS CloudFormation uses these templates as blueprints for building your AWS resources. For example, in a template, you can describe an Amazon EC2 instance, such as the instance type, the AMI ID, block device mappings, and its Amazon EC2 key pair name. Whenever you create a stack, you also specify a template that AWS CloudFormation uses to create whatever you described in the template.

Example JSON

```
{  
  "AWSTemplateFormatVersion" : "2010-09-09",  
  "Description" : "A sample template",  
  "Resources" : {  
    "MyEC2Instance" : {  
      "Type" : "AWS::EC2::Instance",  
      "Properties" : {  
        "ImageId" : "ami-0ff8a91507f77f867",  
        "InstanceType" : "t2.micro",  
        "KeyName" : "testkey",  
        "BlockDeviceMappings" : [  
          {
```

```
        "DeviceName" : "/dev/sdm",
        "Ebs" : {
            "VolumeType" : "io1",
            "Iops" : "200",
            "DeleteOnTermination" : "false",
            "VolumeSize" : "20"
        }
    }
}
]
```

Example YAML

```
AWS::TemplateFormatVersion: "2010-09-09"
Description: A sample template
Resources:
  MyEC2Instance:
    Type: "AWS::EC2::Instance"
    Properties:
      ImageId: "ami-0ff8a91507f77f867"
      InstanceType: t2.micro
      KeyName: testkey
      BlockDeviceMappings:
        -
          DeviceName: /dev/sdm
          Ebs:
            VolumeType: io1
            Iops: 200
            DeleteOnTermination: false
            VolumeSize: 20
```

Stacks

When you use AWS CloudFormation, you manage related resources as a single unit called a stack. You create, update, and delete a collection of resources by creating, updating, and deleting stacks. All the resources in a stack are defined by the stack's AWS CloudFormation template. Suppose you created a template that includes an Auto Scaling group, Elastic Load Balancing load balancer, and an Amazon Relational Database Service (Amazon RDS) database instance. To

create those resources, you create a stack by submitting the template that you created, and AWS CloudFormation provisions all those resources for you.

Change Sets

If you need to make changes to the running resources in a stack, you update the stack. Before making changes to your resources, you can generate a change set, which is a summary of your proposed changes. Change sets allow you to see how your changes might impact your running resources, especially for critical resources, before implementing them.



Updating a Stack with Change Sets

When you need to update your stack's resources, you can modify the stack's template. You don't need to create a new stack and delete the old one. To update a stack, create a change set by submitting a modified version of the original stack template, different input parameter values, or

both. AWS CloudFormation compares the modified template with the original template and generates a change set. The change set lists the proposed changes. After reviewing the changes, you can execute the change set to update your stack or you can create a new change set. The following diagram summarizes the workflow for updating a stack.

1. You can modify an AWS CloudFormation stack template by using [AWS CloudFormation Designer](#) or a text editor. For example, if you want to change the instance type for an EC2 instance, you would change the value of the `InstanceType` property in the original stack's template.
2. Save the AWS CloudFormation template locally or in an S3 bucket.
3. Create a change set by specifying the stack that you want to update and the location of the modified template, such as a path on your local computer or an Amazon S3 URL. If the template contains parameters, you can specify values
4. When you create the change set, view the change set to check that AWS CloudFormation will perform the changes that you expect. For example, check whether AWS CloudFormation will replace any critical stack resources. You can create as many change sets as you need until you have included the changes that you want.
5. Execute the change set that you want to apply to your stack. AWS CloudFormation updates your stack by updating only the resources that you modified and signals that your stack has been successfully updated. If the stack update fails, AWS CloudFormation rolls back changes to restore the stack to the last known working state.

Deleting a Stack

When you delete a stack, you specify the stack to delete, and AWS CloudFormation deletes the stack and all the resources in that stack.

If you want to delete a stack but want to retain some resources in that stack, you can use a [deletion policy](#) to retain those resources.

After all the resources have been deleted, AWS CloudFormation signals that your stack has been successfully deleted. If AWS CloudFormation cannot delete a resource, the stack will not be deleted. Any resources that haven't been deleted will remain until you can successfully delete the stack.

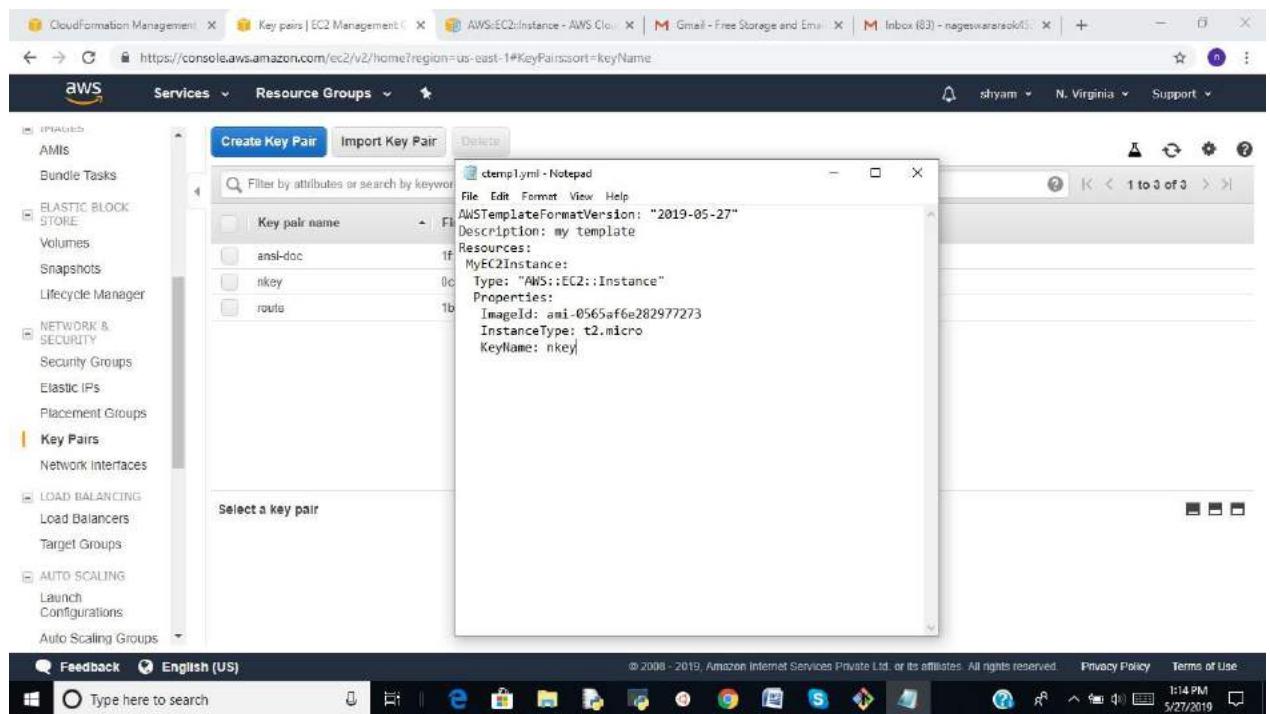
UseCase:

Create template

- Open text editor and write code by using json or yaml format by following template conditions and rules
- Below mention the example template for launching one ec2 instance

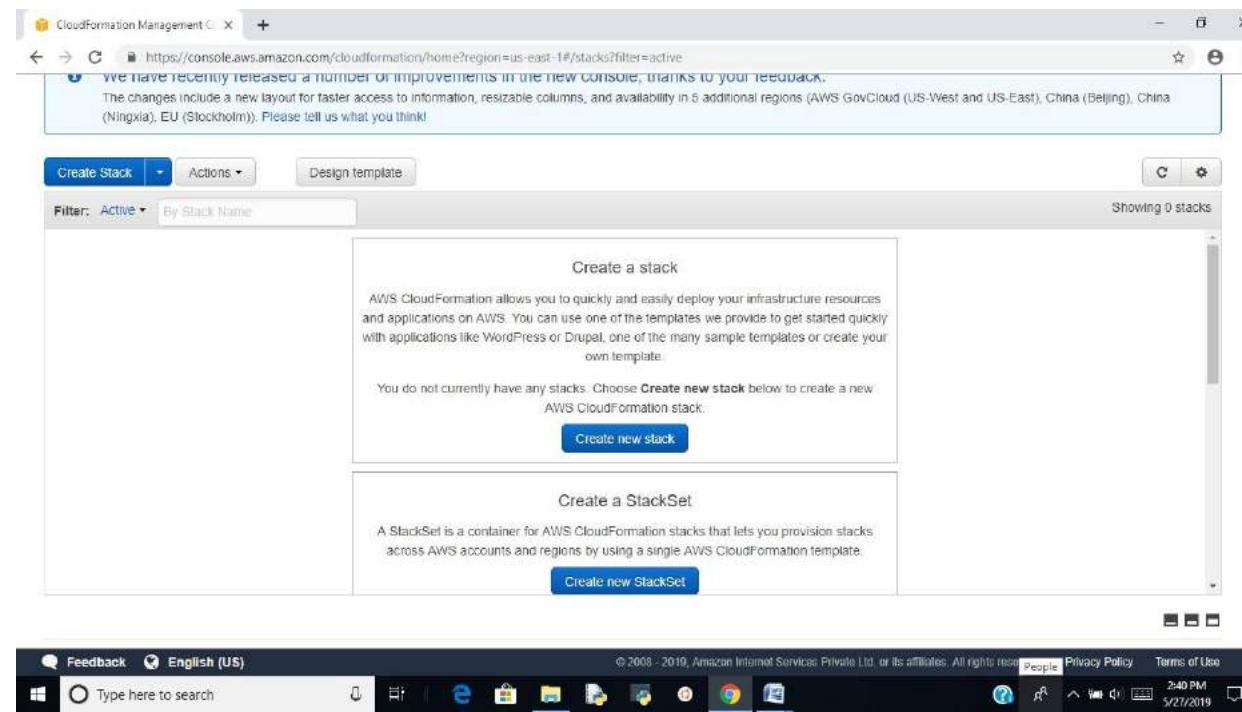
```
AWSTemplateFormatVersion: "2010-09-09"
Description: A sample template
Resources:
  MyEC2Instance:
    Type: "AWS::EC2::Instance"
    Properties:
      ImageId: "ami-0ff8a91507f77f867"
      InstanceType: t2.micro
      KeyName: testkey
```

- First line indicates the template version you can give your version for that template
- Second line indicate description for that template
- Third line is the Resource section, in this resource section you mention resource type and properties for that resource. In this section you can include any resource and properties one by one by following json or yaml syntax which resources you are needed.
- First statement in resource section is the resource name
 - Within the resource section you mention resource type and its properties
 - In properties section you write minimum properties to implement that service
 - Optional parameters are not mandatory, do you want to use you can pass
 - You not mention optional parameters then assigned to default values
- In above case we mention the imageid, InstanceType and keyname these all are the mandatory properties for EC2. Another properties are assigned with default values that is vpc id, subnet id, security group id..etc



Create Stack

- Signin with aws console and select cloudforamtion as service
- Click on create stack and click on create new stack



Select Template

- Here mainly two options is there one is design template and another one is choose a template
- We can design template with help of template designer tools by simply drag and drop

Choose a Template

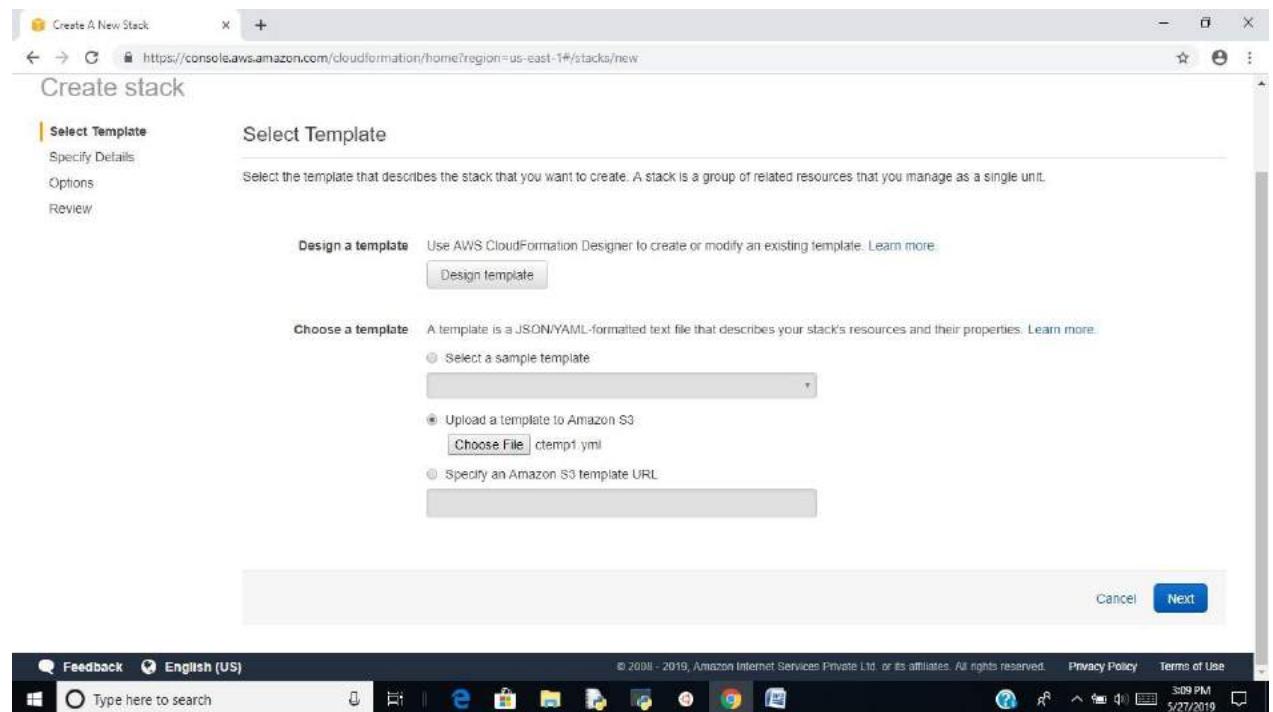
In this section we have 3 types to choose a file.

Select a sample template: choosing already existing sample template

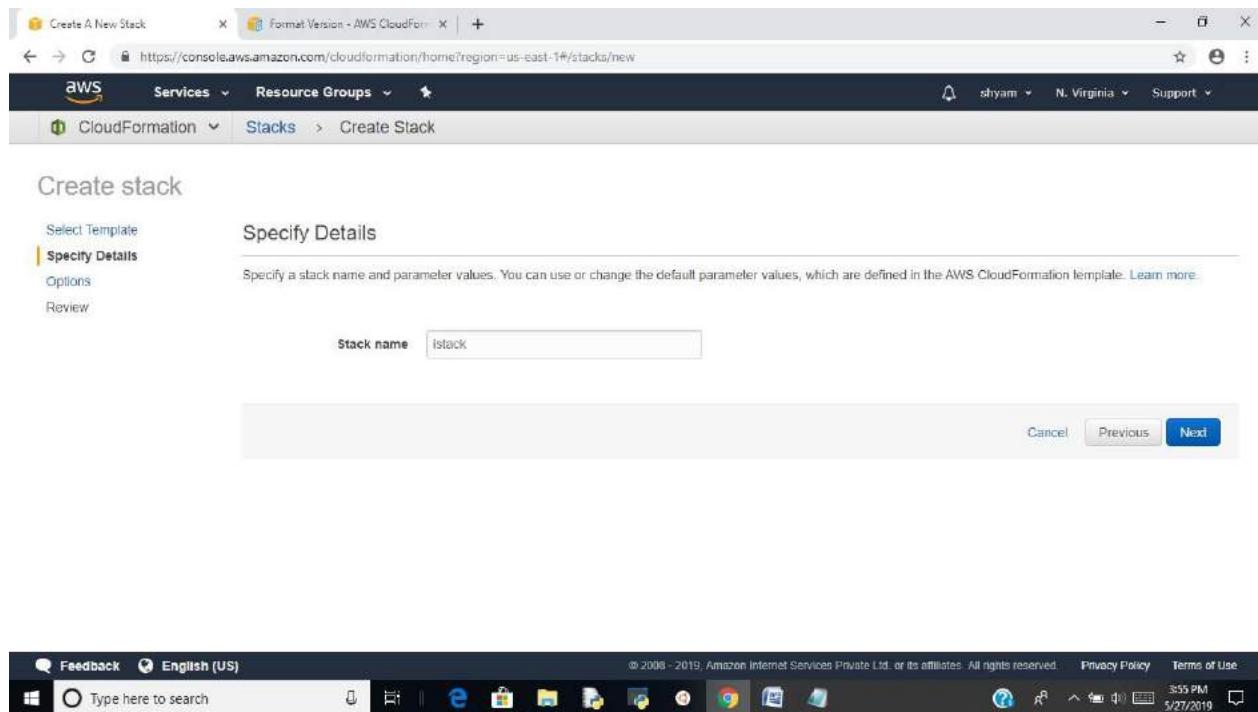
Upload a template to amazon s3: choose your own template and upload to amazon s3 bucket

Specify an Amazon s3 template URL: choose template from Amazon s3 bucket

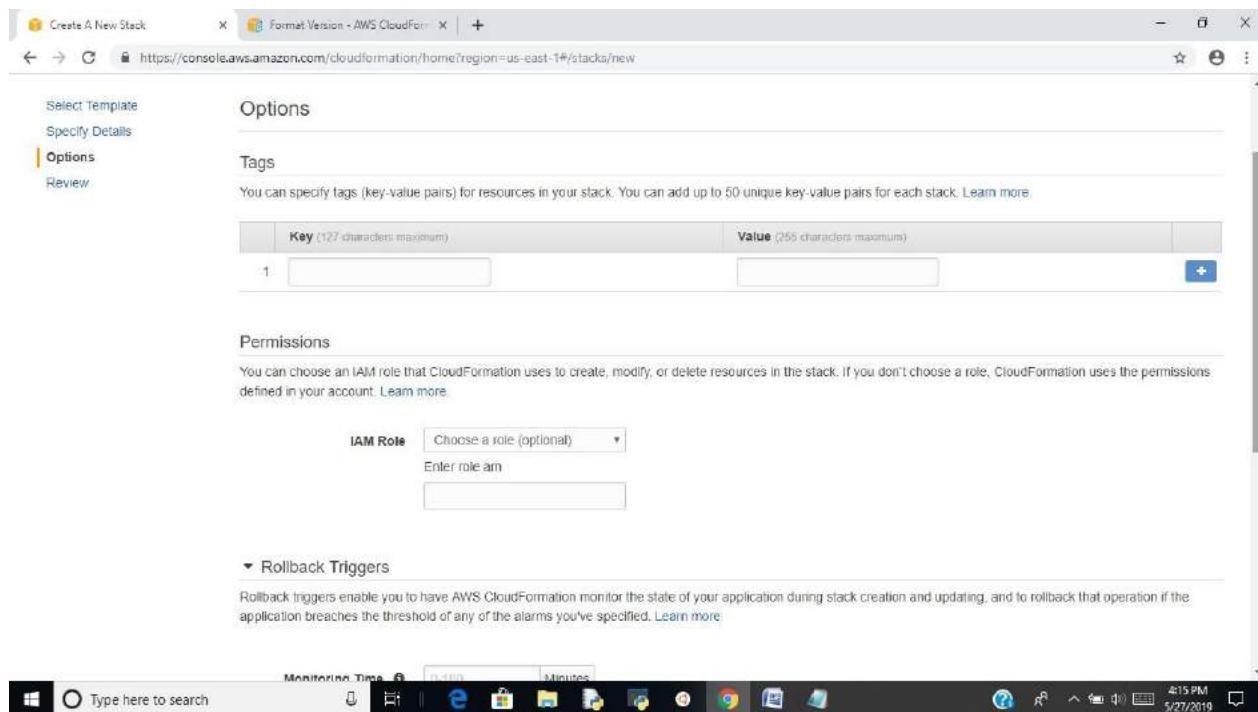
Note: in our case we use upload a template to Amazon s3 option to choose file from our local machine



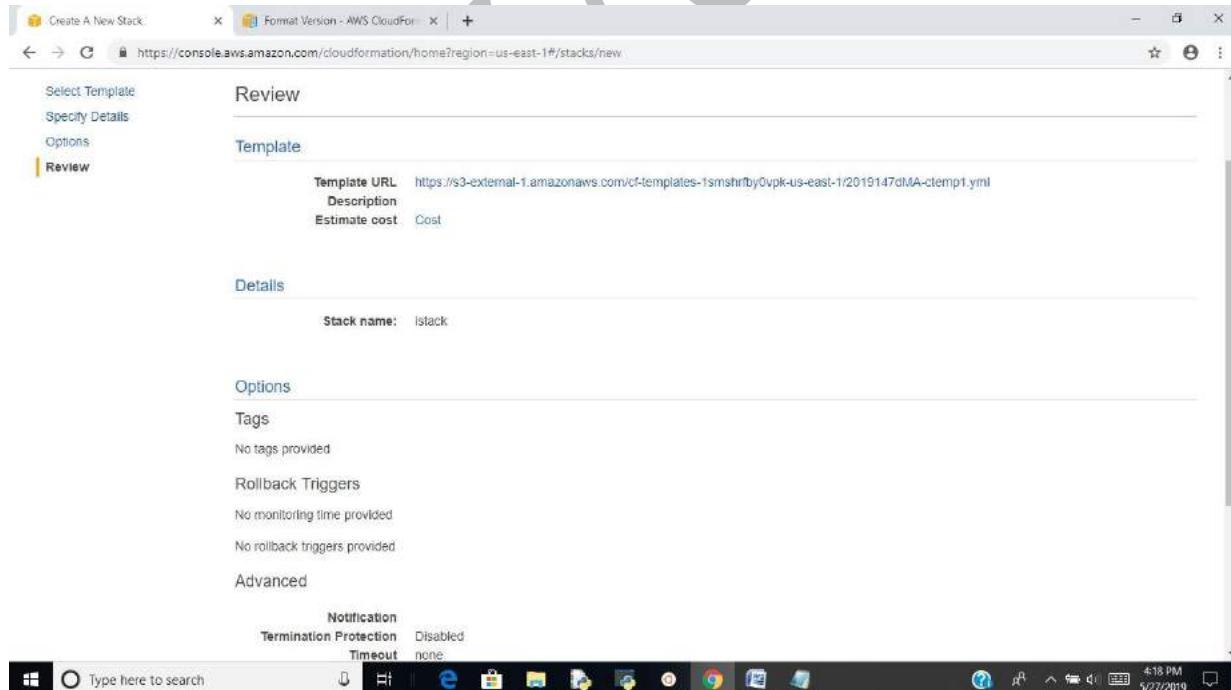
- Click on next, if template is correct then goto specify details section
- Enter stack name and click on next



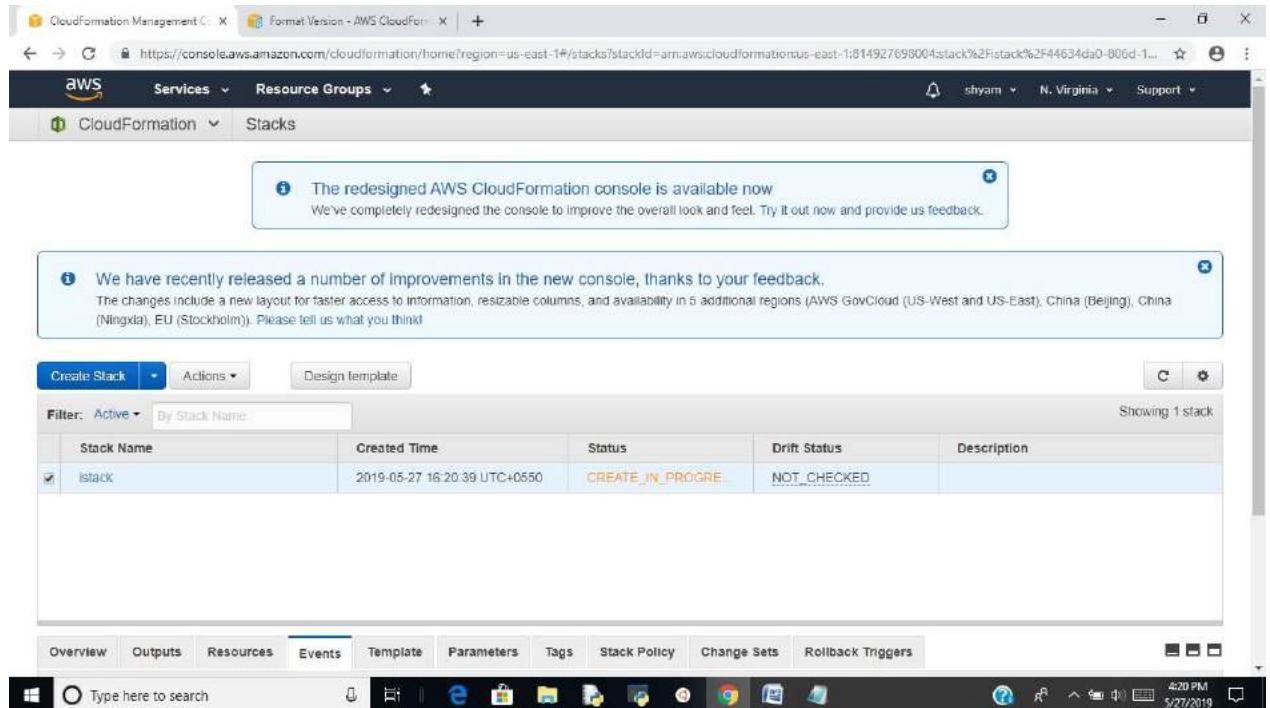
- Options section is not mandatory but do you want mention you can mention these details
- In tags section you can enter key name and value for this stack
- Permissions: You can choose an IAM role that CloudFormation uses to create, modify, or delete resources in the stack. If you don't choose a role, CloudFormation uses the permissions defined in your account.
- RollbackTriggers: Rollback triggers enable you to have AWS CloudFormation monitor the state of your application during stack creation and updating, and to rollback that operation if the application breaches the threshold of any of the alarms you've specified.
- Click on next



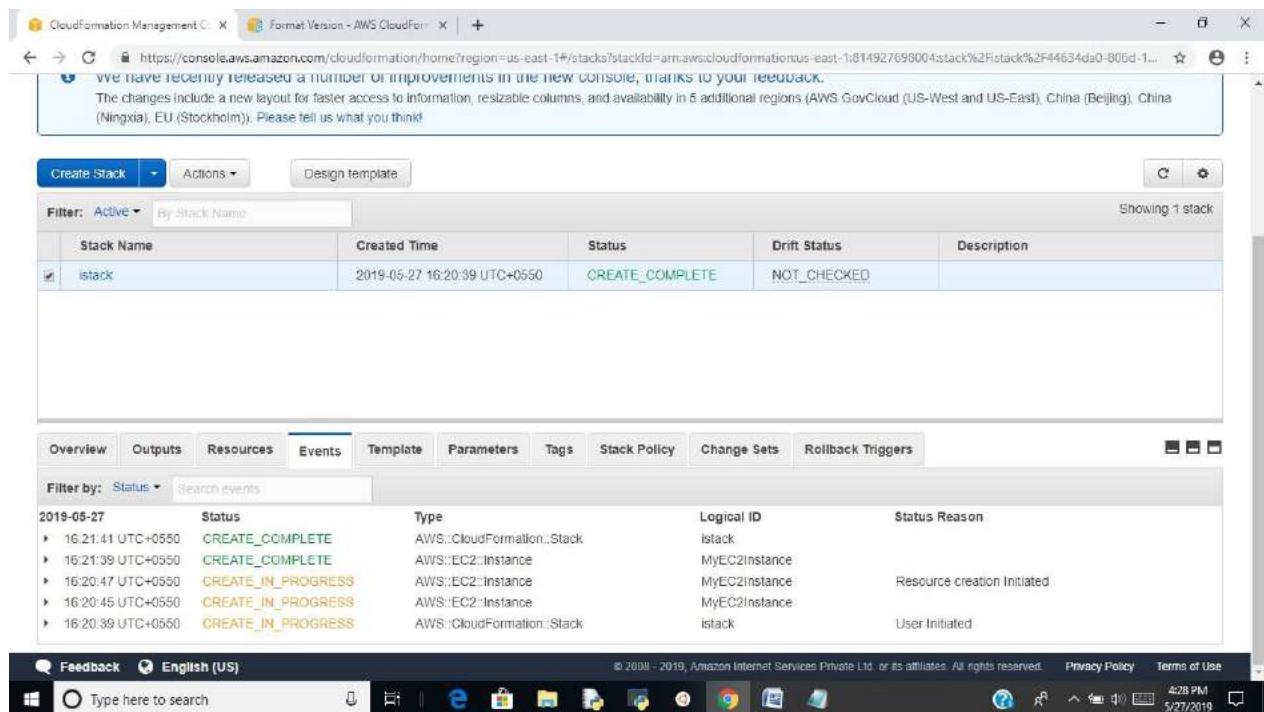
- You can see all attribute used to create a stack and these values given by you



- Click on create



- Creation of stack is initiated
- If any issues find in resource creation then completely rollback by deleting previously created resources through same stack
- Stack is the collection of resources as a single unit.
- Stack is created successfully



Create Change Set for Current Stack

- Do you want to add some changes and modify existing resources then you can go change set
- You can create one template with changes of existing resources or template
- Upload this template to stack
- Add 80 port to above created ec2 instance security group and launch one new instance and install apache2 on that instance
- You can absorb modified template below

Resources:

MyEC2Instance:

Type: "AWS::EC2::Instance"

Properties:

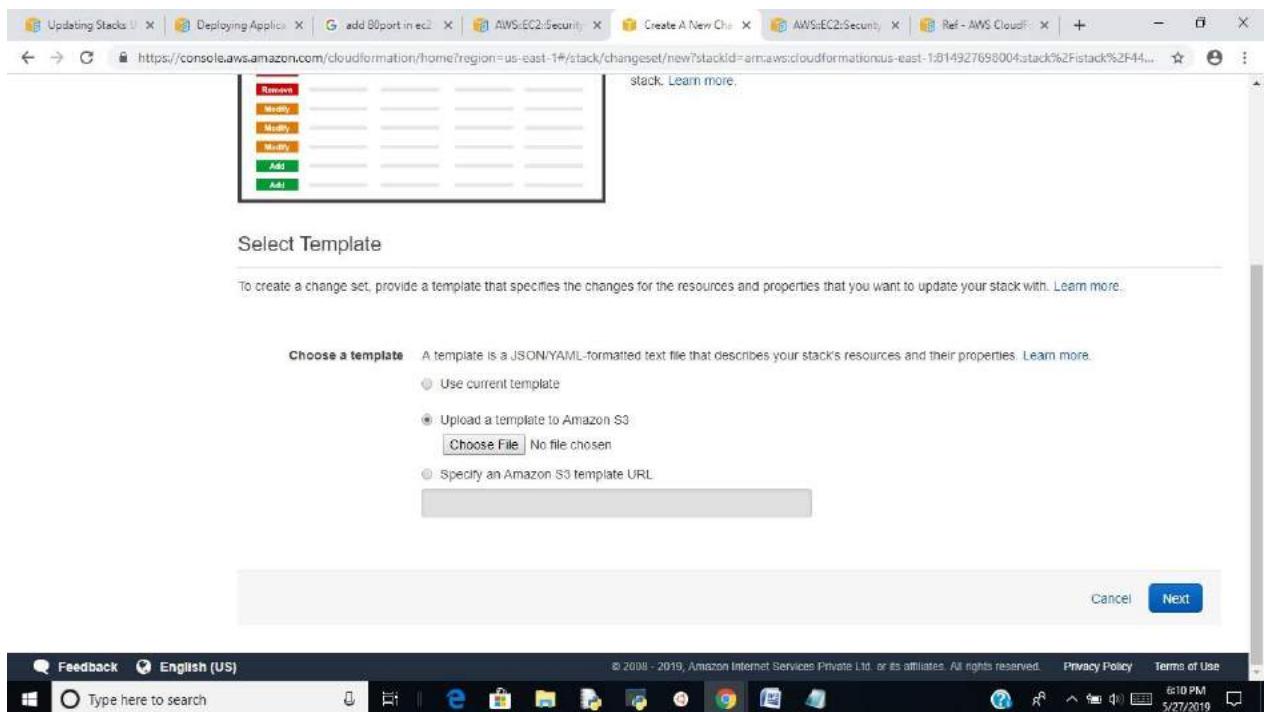
ImageId: ami-0565af6e282977273

InstanceType: t2.micro

KeyName: nkey

```
SecurityGroupId: !Ref mysg
mysg:
  Type: 'AWS::EC2::SecurityGroup'
  Properties:
    GroupDescription: Base Security Group
    SecurityGroupIngress:
      - IpProtocol: tcp
        CidrIp: 0.0.0.0/0
        FromPort: 80
        ToPort: 80
  chresource:
    Type: "AWS::EC2::Instance"
    Properties:
      ImageId: ami-0565af6e282977273
      InstanceType: t2.micro
      KeyName: nkey
      SecurityGroupId: !Ref mysg
      UserData:
        #!/bin/bash
        sudo apt-get update
        sudo apt-get install -y apache2
```

- goto cloudformation service and select previously created stack and goto actions here we select create change set for current stack



- Here we choose modified template file and click on next
- Enter name for changeset and click on next
- Do you want mention option field fill that or skip by click on next

Amazon Web Services

The screenshot shows the AWS CloudFormation Options page. It includes sections for:

- Select Template
- Specify Details
- options** (highlighted)
- Review

Options: You can update additional options for your stack, like notification options and a stack policy.

Tags: You can specify tags (key-value pairs) for resources in your stack. You can add up to 50 unique key-value pairs for each stack. Learn more.

Permissions: You can choose an IAM role that CloudFormation uses to create, modify, or delete resources in the stack. If you don't choose a role, CloudFormation uses the permissions defined in your account. Learn more.

IAM Role: Choose a role (optional). Enter role arn.

Rollback Triggers: Rollback triggers enable you to have AWS CloudFormation monitor the state of your application during stack creation and updating, and to rollback that operation if the application breaches the threshold of any of the alarms you've specified. Learn more.

- See the review section and click on create change set

The screenshot shows the AWS CloudFormation Create Change Set page. It includes sections for:

Description

Details

- Stack name:** istack
- Change set name:** ocs
- Change set description:**

Options

Tags: No tags provided

Rollback Triggers: No monitoring time provided
No rollback triggers provided

Advanced

Notification

Buttons at the bottom: Cancel, Previous, **Create change set**.

- You carefully absorb below screen three changes is there

Created time: 2019-05-27 18:17:21 UTC+0500

Status: CREATE_COMPLETE

Stack name: lstack

Change set input

Changes

The changes CloudFormation will make if you execute this change set.

Action	Logical ID	Physical ID	Resource Type	Replacement
Modify	MyEC2Instance	i-061937f26b1448c85	AWS::EC2::Instance	False
Add	chresource		AWS::EC2::Instance	
Add	mysg		AWS::EC2::SecurityGroup	

Details

Template

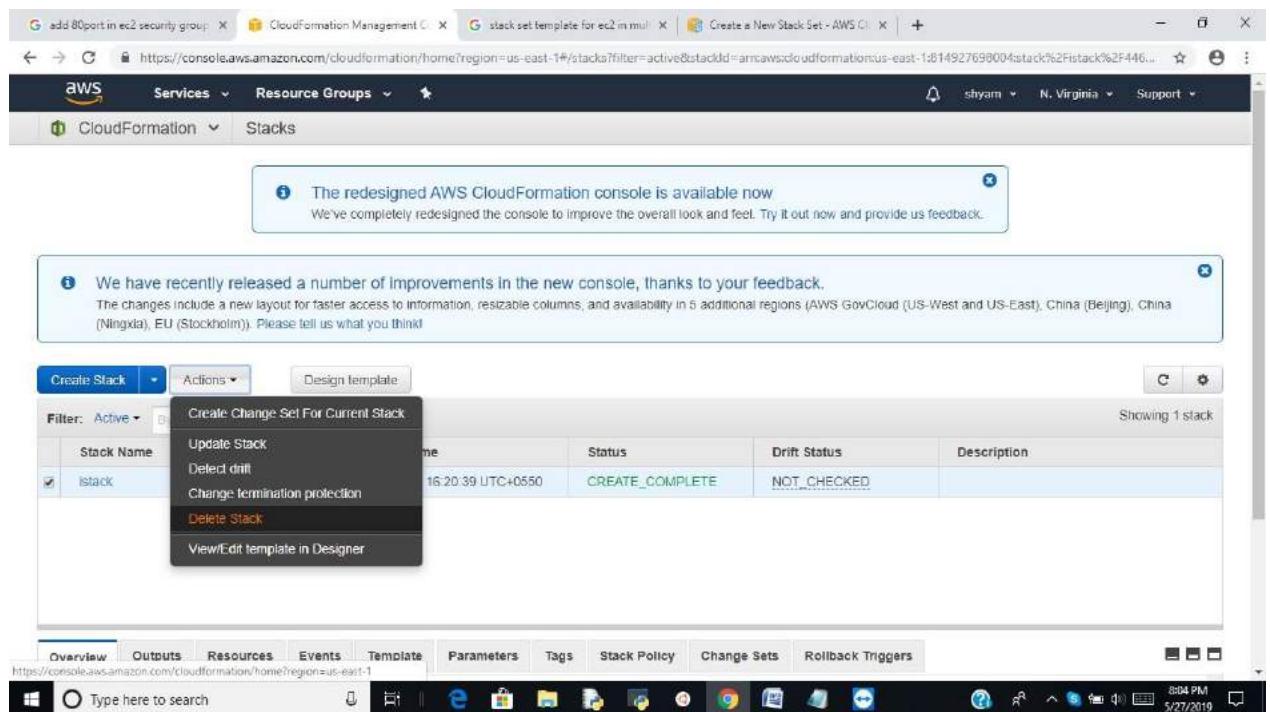
Feedback English (US) © 2008 - 2019, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. Privacy Policy Terms of Use

Type here to search

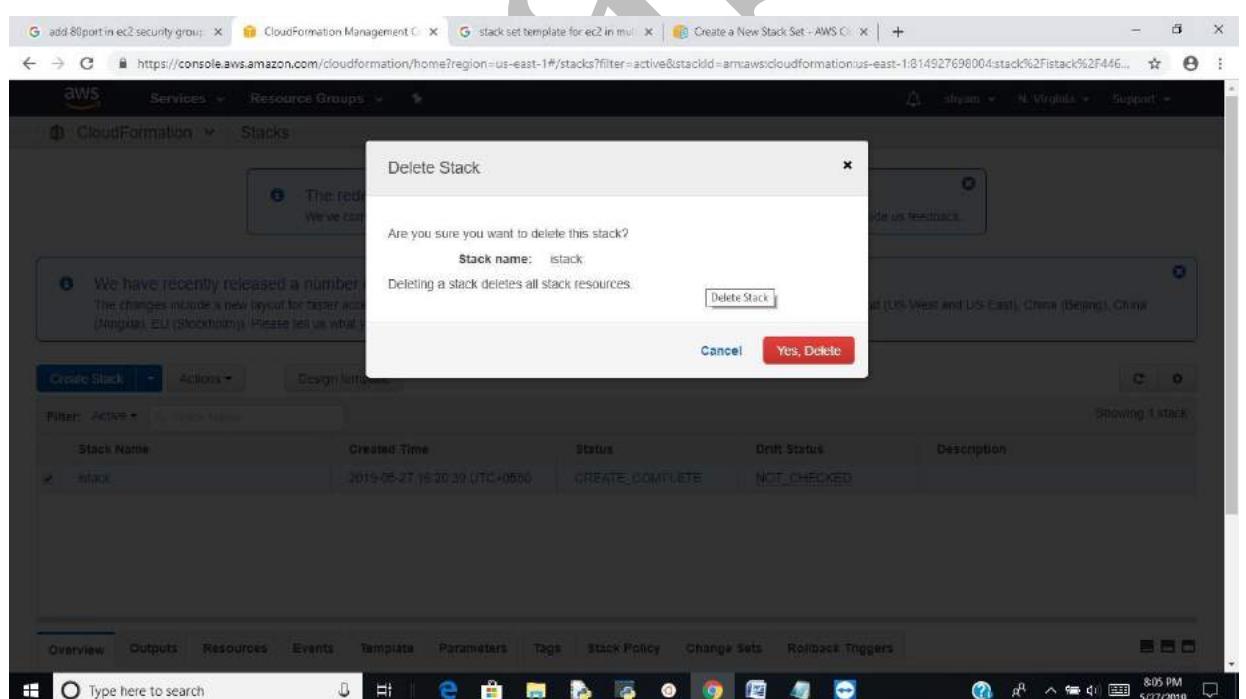
- Here Myec2Instance that is previously created is modified and chresource, mysg are added
- Update the stack by deploying new template or modified template into the existing stack as like above change set scenario.

Deleting Stack

- Select stack which stack do you want to deleted and goto actions here click on delete stack option



- Click on yes, delete



- The all resources with in that stack are deleted

Amazon CloudWatch

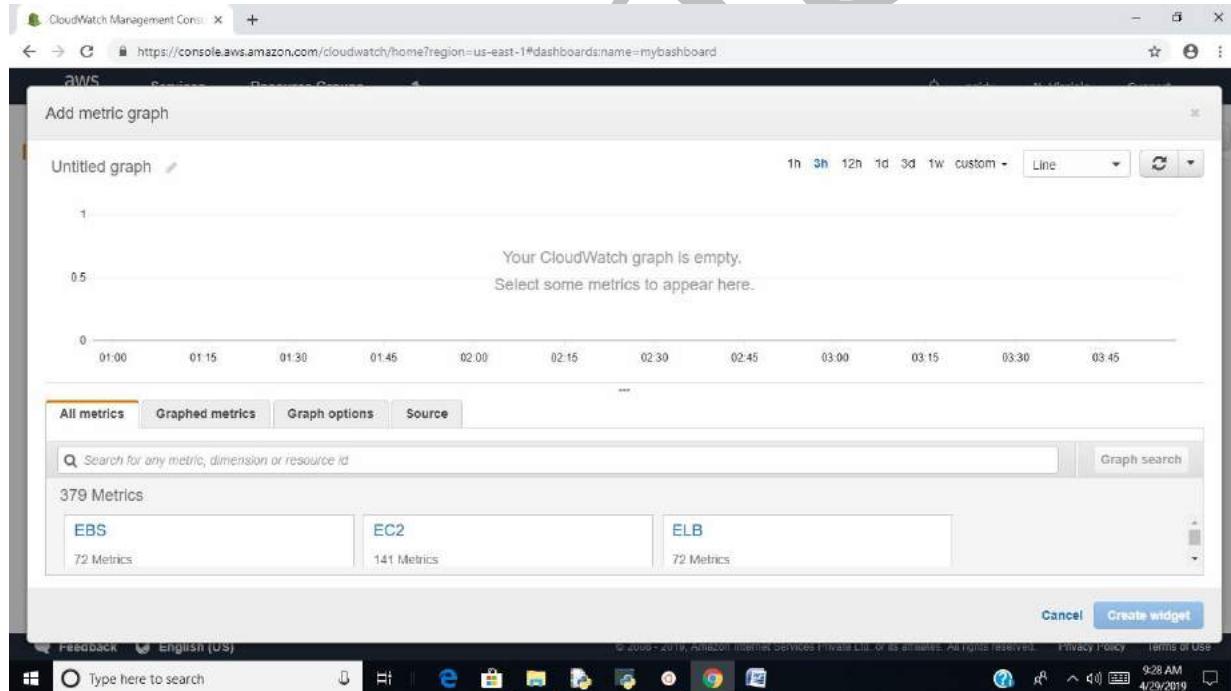
Amazon CloudWatch monitors your Amazon Web Services (AWS) resources and the applications you run on AWS in real time. You can use CloudWatch to collect and track metrics, which are variables you can measure for your resources and applications.

Use Case

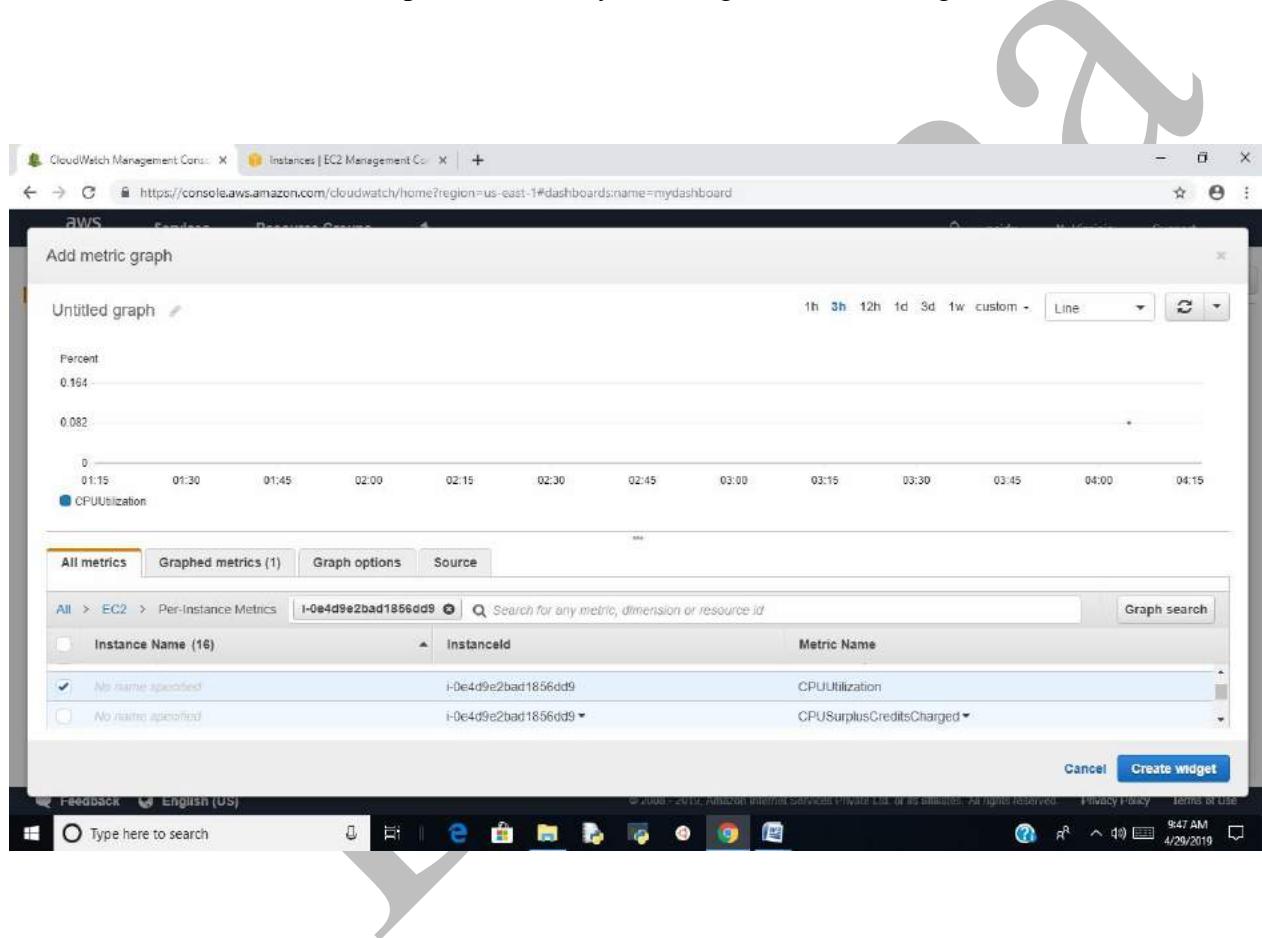
- Login into AWS console and choose cloudwatch service
- In the left side panel we can see the cloud watch Events..(Dashboards, Logs...etc)

Dashboards

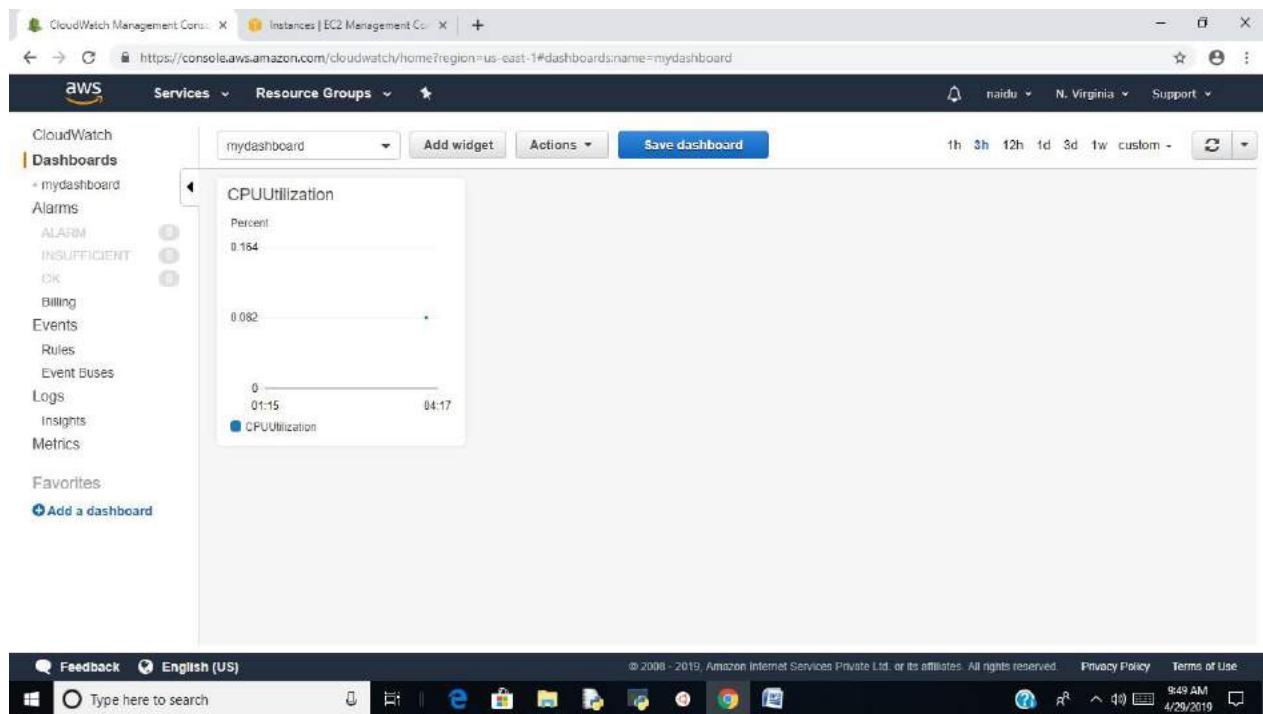
- Click on Create Dashboard and enter Dashboard Name and select widget type (lines or graphs)



- In the metric section select the service which service do you want to monitor (click on EC2)
- Select instanceid with particular metric (ex.. cp utilizati on, disk input and access and network)
- So we need to Launch one EC2 instance
- Go to EC2 service section and click on Launch instance and select all necessary attributes to create these EC2 instance that is AMI, instance type, vpc id and subnet id and key pair name
- Then we select metric cpu utilization by selecting above launching ec2 instance id



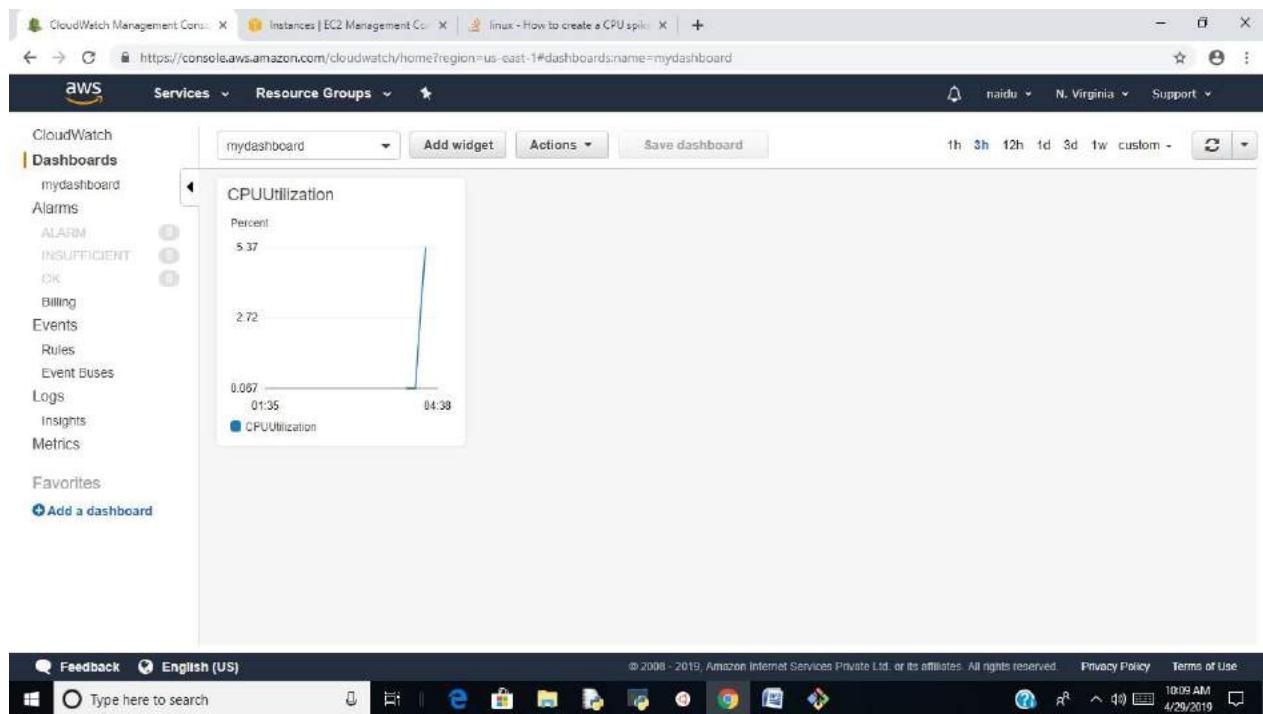
- Click on create Widget then you see selected instance cup utilization on dash board



- In above graph the horizontal line indicate time and vertical line indicates percentage of cpu utilization and click on Save DashBoard
- You can increasee cpu utilization of your ec2 instance by applying more load by executing fallowing command in your terminal

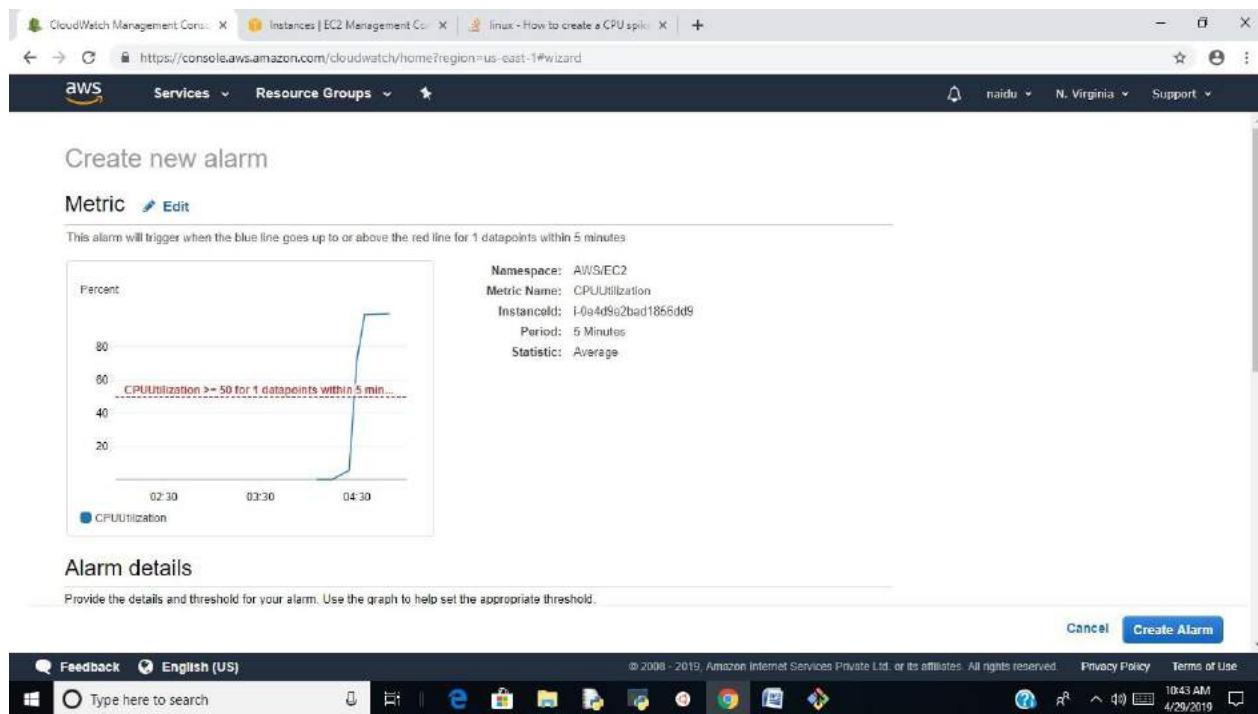
dd if=/dev/zero of=/dev/null

- Refresh your DashBoard and absorve modified graph



Alarms

- Alarms are used for put conditions on Events or Actions and make some more Actions Depending on condition. As like Prediction
- Click on create alarm and select metric here metric is EC2 Instance cpu utilization
- Give some name to this alarm



- Go to Whenever (it is the condition section). Here you mention the cpu utilization value based on this value triggering is happened. (Here Notification is Triggering Action)
- Go to Actions section that is which action do you want perform when condition is true
- Here we can select any action like Notification, Auto scaling Action and EC2 Action
- Select the alarm state this indicate which state do you want to perform the alarm. Here select STATE is ALARM

Second option is select SNS topic for send Notification

Whenever: CPUUtilization
is: \geq 50
for: 1 datapoint

Additional settings
Provide additional configuration for your alarm.
Treat missing data as: missing

Actions
Define what actions are taken when your alarm changes state.

Notification	Delete
Whenever this alarm: State is ALARM	
Send notification to: Select a notification list	New list Enter list

+ Notification + AutoScaling Action + EC2 Action

Create Alarm

- So first we create SNS Topic
- Goto SNS Service section and goto Topics Section Here Click on Create topic
- Give topic Name and click create topic

CloudWatch Management Console | Instances | EC2 Management Con... | linux - How to create a CPU spike | +

https://console.aws.amazon.com/cloudwatch/home?region=us-east-1#wizard

aws Services Resource Groups

Whenever: CPUUtilization
is: \geq 50
for: 1 datapoint

Additional settings

Treat missing data as: missing

Actions

Define what actions are taken when your alarm changes state.

Notification	Delete
Whenever this alarm: State is ALARM	
Send notification to: Select a notification list	New list Enter list

+ Notification + AutoScaling Action + EC2 Action

Create Alarm

Feedback English (US)

Type here to search

11:33 AM 4/29/2019

CloudWatch Management Console | Simple Notification Service | linux - How to create a CPU spike | +

https://console.aws.amazon.com/sns/v3/home?region=us-east-1#/create-topic

aws Services Resource Groups

Amazon SNS > Topics > Create topic

Create topic

Details

Name: mytopic
Maximum 256 characters. Can include alphanumeric characters, hyphens (-) and underscores (_).

Display name - optional
To use this topic with SMS subscriptions, enter a display name. Only the first 10 characters are displayed in an SMS message. Info
My Topic
Maximum 100 characters, including hyphens (-) and underscores (_).

► Encryption - optional
Amazon SNS provides in-transit encryption by default. Enabling server-side encryption adds at-rest encryption to your topic.

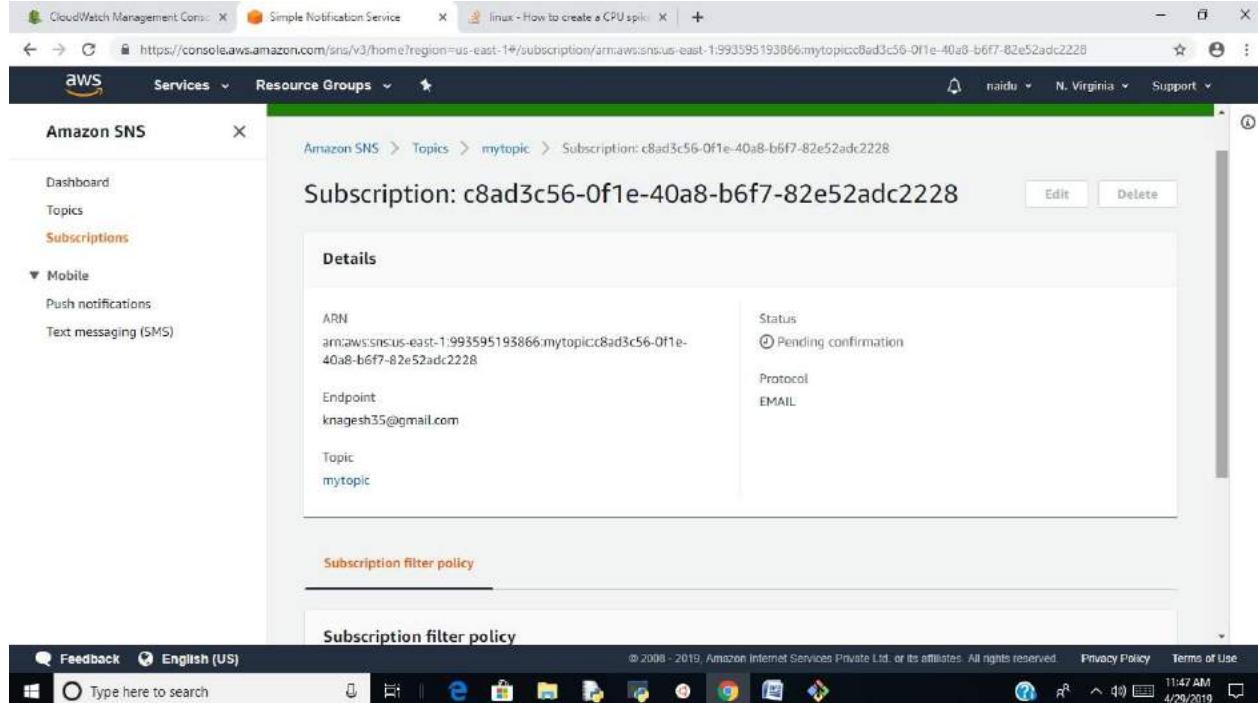
► Access policy - optional
This policy defines who can access your topic. By default, only the topic owner can publish or subscribe to the topic. Info

Feedback English (US)

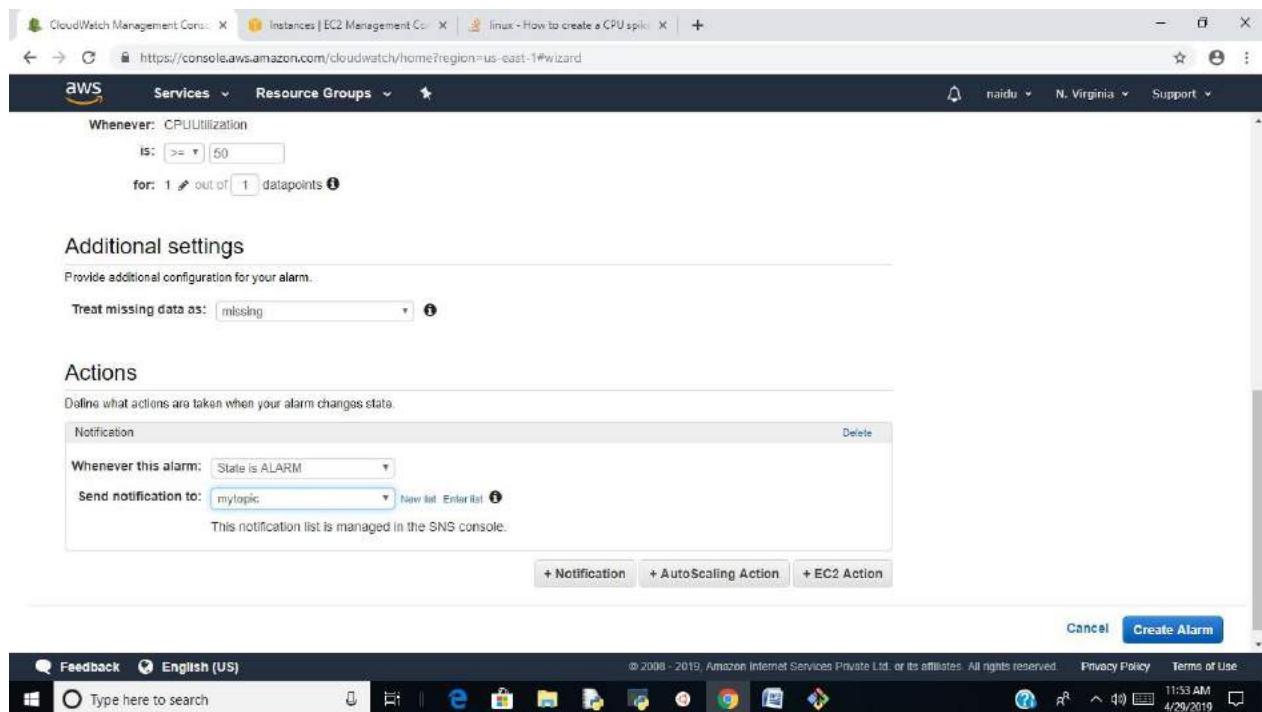
Type here to search

11:33 AM 4/29/2019

- Click on create subscription
- Select protocol as Email and enter your Email address in Endpoint section
- Then click on create subscription



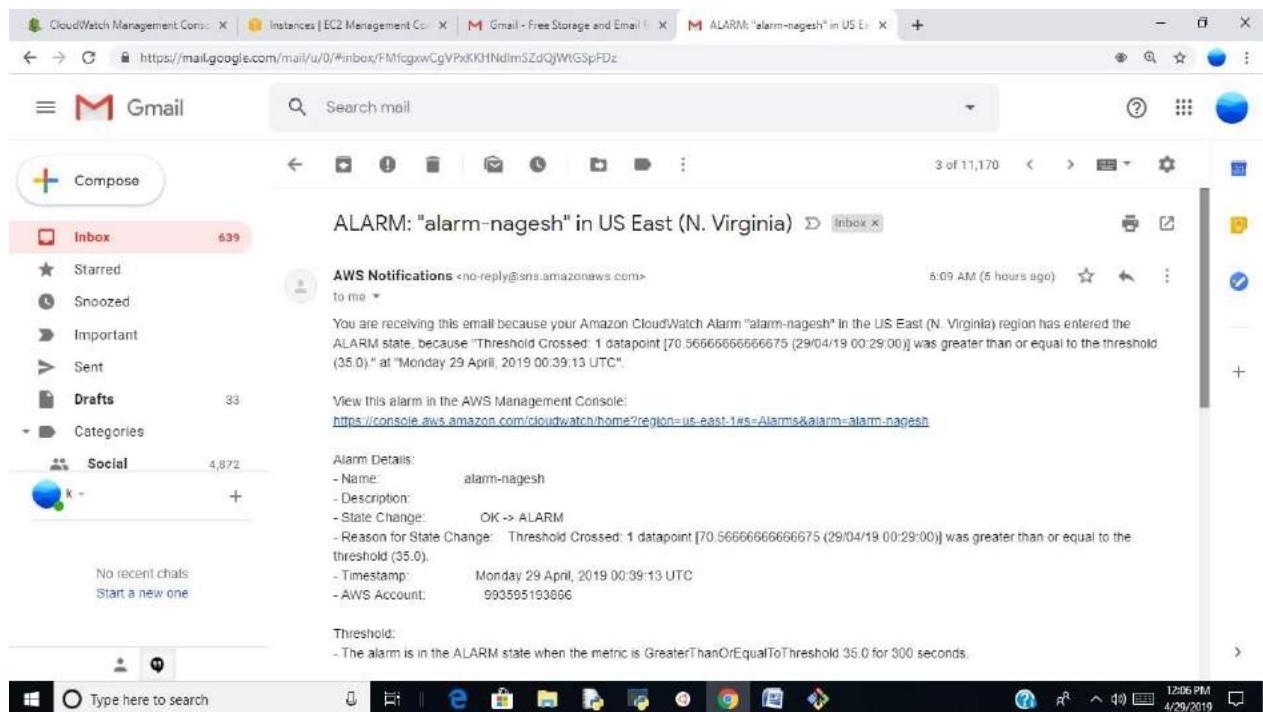
- Then come cloud watch again and refresh and select these above created SNS topic Name
- Notification section
- Click on create Alarm



- In above Case I give the Cpu utilization value in condiction section is 50%. Present cpu utilization of EC2 Instance is more than 50% since I was running

dd if=/dev/zero of=/dev/null command in that instance

- So State is Alarm since it's reach the condition then send notification action is performed
- Open your SNS subscription mail and check it



Logs

- By using logs in CloudWatch we can monitor both system Logs and Application Logs
- Install the python on aws ec2 instance (**sudo apt-get install -y python**)
- Install the aws CloudWatch Logs agent by following commands

curl https://s3.amazonaws.com/awscloudwatch/downloads/latest/awslogs-agent-setup.py -O

Note: the above code Download the awslogs agent python script from s3

- Run the above Download python Script by specifying region and Configure settings needed for CloudWatch Logs agent

sudo python ./awslogs-agent-setup.py --region us-east-1

- Then it is asking accesskey and secretaccesskey for aws cli configuration

The screenshot shows a Windows desktop environment with a terminal window open. The terminal window title is "Intelliq-softcopy" and the command prompt is "ubuntu@ip-172-31-88-57:~". The terminal output shows the following steps:

```
<?xml version="1.0" encoding="UTF-8"?>
^
SyntaxError: invalid syntax
ubuntu@ip-172-31-88-57:~$ rm awslogs-agent-setup.py
ubuntu@ip-172-31-88-57:~$ ls
ubuntu@ip-172-31-88-57:~$ curl https://s3.amazonaws.com/aws-cloudwatch/downloads/latest/awslogs-agent-setup.py -O
% Total % Received % Xferd Average Speed Time Time Time Current
          Dload Upload Total Spent Left Speed
100 56093  100 56093    0     0  676k      0 --:--:--:--:--:-- 684k
ubuntu@ip-172-31-88-57:~$ ls
awslogs-agent-setup.py
ubuntu@ip-172-31-88-57:~$ python --version
Python 2.7.12
ubuntu@ip-172-31-88-57:~$ sudo python ./awslogs-agent-setup.py --region us-east-1
Launching interactive setup of CloudWatch Logs agent ...
Step 1 of 5: Installing pip ...libyaml-dev does not exist in system DONE
Step 2 of 5: Downloading the latest CloudWatch Logs agent bits ... DONE
Step 3 of 5: Configuring AWS CLI ...
AWS Access Key ID [None]: |
```

The terminal window is part of a larger desktop interface with a taskbar at the bottom showing various application icons.

- If you don't know accesskey and secretaccesskey then goto IAM section and select
 - Goto root access keys section click on manage security credentials
 - Then click accesskey and secretaccesskey then click on create new accesskey

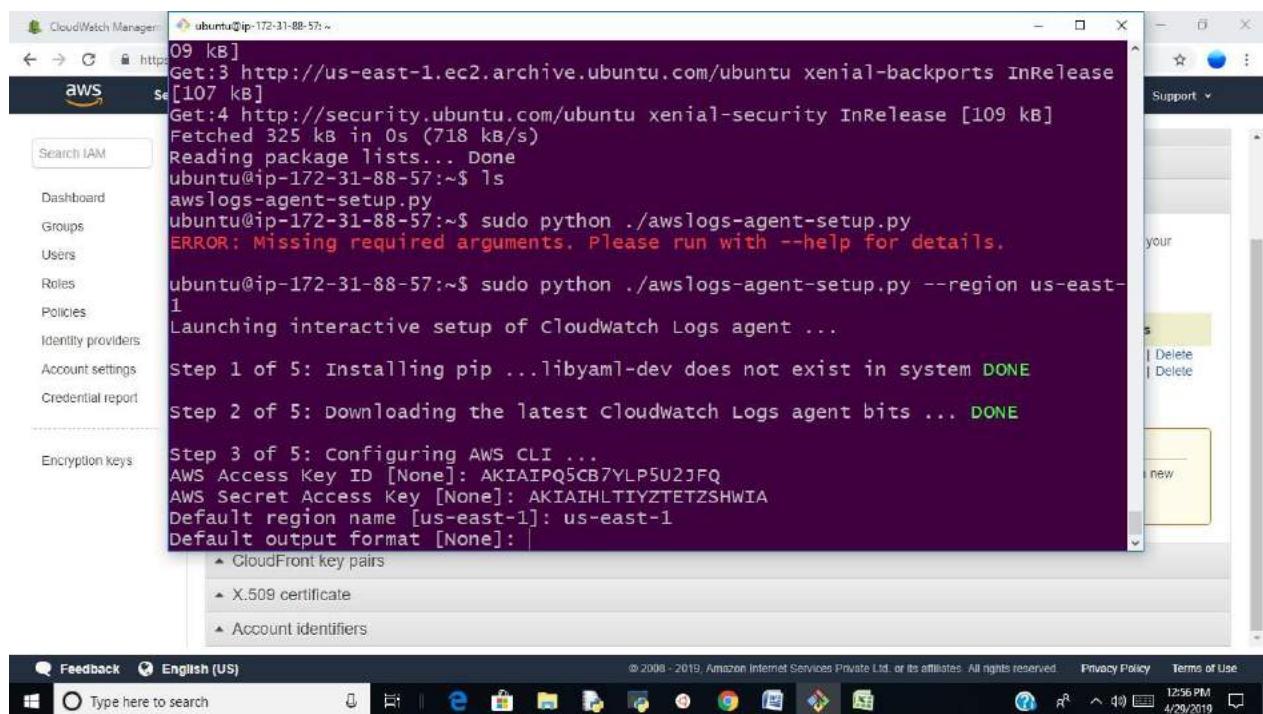
The screenshot shows the AWS IAM Management Console with the 'Access keys (access key ID and secret access key)' section selected. The table lists two access keys:

Created	Deleted	Access Key ID	Last Used	Last Used Region	Last Used Service	Status	Actions
Apr 29th 2019		AKIAIPQSCB7YLP5U2JFQ	2019-04-29 07:10 UTC+0530	us-east-1	logs	Active	Make Inactive Delete
Mar 8th 2019		AKIAIHLTIYZTETZSHWIA	2019-03-08 19:07 UTC+0530	ap-southeast-1	cloudformation	Active	Make Inactive Delete

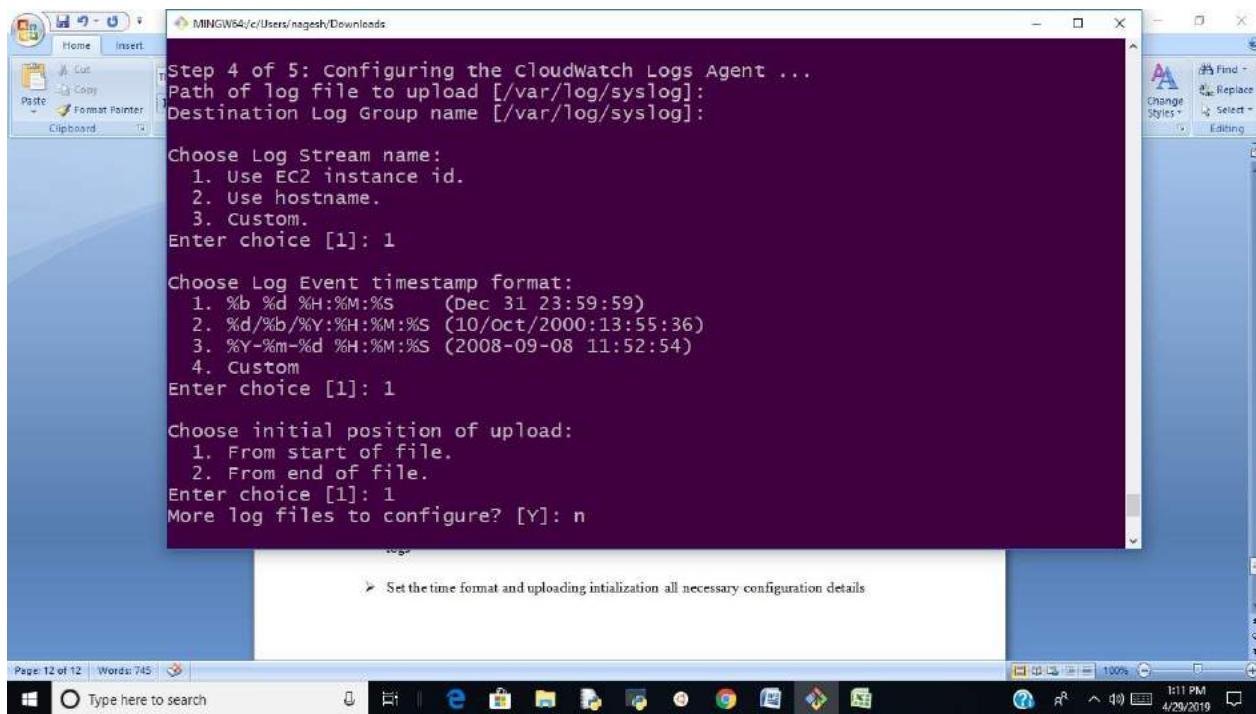
Important Change - Managing Your AWS Secret Access Keys

As described in a previous announcement, you cannot retrieve the existing secret access keys for your AWS root account, though you can still create a new root access key at any time. As a best practice, we recommend creating an IAM user that has access keys rather than relying on root access keys.

- Copy both keys
- Enter these accesskey and secret accesskey and region when you configure Cloudwatch Logs agent



- In configuration it is log file path then you enter log file location which log file do you want monitor
- By default it is assigned with /var/logs/syslog it is the system log file
- You can also monitor application log files like apache, tomcat..etc by specifying that log file location instead os /var/logs/syslog
- You can also configure destination log group names which is appeared in cloud watch logs
- Set the time format and uploading initialization all necessary configuration details

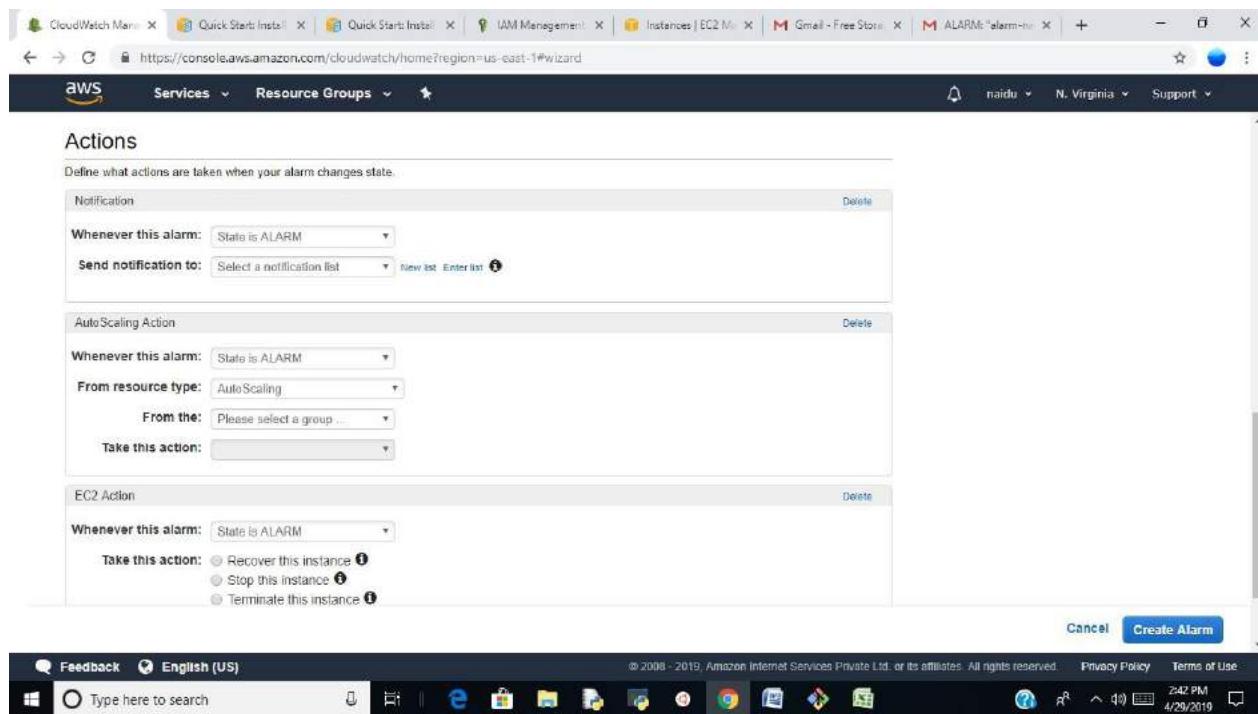


- You can goto CloudWatch dashboard then goto Logs Section then you find log group that is you previously configured destination log group name
- Click on that log group then you find all your system logs

The screenshot shows the AWS CloudWatch Management Console with the CloudWatch Logs Insights interface open. The URL in the address bar is <https://console.aws.amazon.com/cloudwatch/home?region=us-east-1#logEventViewer:group=/var/log/syslog;start=2019-04-28T09:38:13Z>. The left sidebar shows navigation links for CloudWatch Dashboards, Alarms, Billing, Events, Rules, Event Buses, Logs (selected), Insights, Metrics, Favorites, and Add a dashboard. The main content area displays a modal window titled "Try CloudWatch Logs Insights" with a message about the purpose-built query language. Below the modal is a table titled "Filter events" with columns "Time (UTC +00:00)" and "Message". The table lists log entries from April 29, 2019, at 09:30:36 UTC. The messages describe various system boot and configuration events, such as kernel initialization and cgroup subsystems being initialized.

Time (UTC +00:00)	Message
2019-04-29	Apr 29 09:30:36 ip-172-31-80-58 rsyslogd: [origin software="rsyslogd" swVersion="8.16.0" x-pid="1071" x-info="http://www.rsyslog.com"] start
09:30:36	Apr 29 09:30:36 ip-172-31-80-58 rsyslogd-2222: command 'KLogPermitNonKernelFacility' is currently not permitted - did you already set it via
09:30:36	Apr 29 09:30:36 ip-172-31-80-58 rsyslogd: rsyslogd's groupid changed to 108
09:30:36	Apr 29 09:30:36 ip-172-31-80-58 rsyslogd: rsyslogd's usend changed to 104
09:30:36	Apr 29 09:30:36 ip-172-31-80-58 rsyslogd-2039: Could not open output pipe 'dev/xconsole': No such file or directory [v8.16.0 try http://www.rs
09:30:36	Apr 29 09:30:36 ip-172-31-80-58 rsyslogd-2007: action 'action 11' suspended, next retry is Mon Apr 29 09:31:06 2019 [v8.16.0 try http://www.rs
09:30:36	Apr 29 09:30:36 ip-172-31-80-58 kernel [0.000000] Initializing cgroup subsys cpuset
09:30:36	Apr 29 09:30:36 ip-172-31-80-58 kernel [0.000000] Initializing cgroup subsys cpu
09:30:36	Apr 29 09:30:36 ip-172-31-80-58 kernel [0.000000] Initializing cgroup subsys cpacct
09:30:36	Apr 29 09:30:36 ip-172-31-80-58 kernel [0.000000] Linux version 4.4.0-1075-aws (buildd@lgw01-amd64-035) (gcc version 5.4.0 20160609 (L
09:30:36	Apr 29 09:30:36 ip-172-31-80-58 kernel [0.000000] Command line: BOOT_IMAGE=/boot/vmlinuz-4.4.0-1075-aws root=LABEL=cloudimg-root
09:30:36	Apr 29 09:30:36 ip-172-31-80-58 kernel [0.000000] KERNEL supported cores:

- For each alarm we can implement multiple actions that is Notifications, AutoScaling and EC2 instance actions

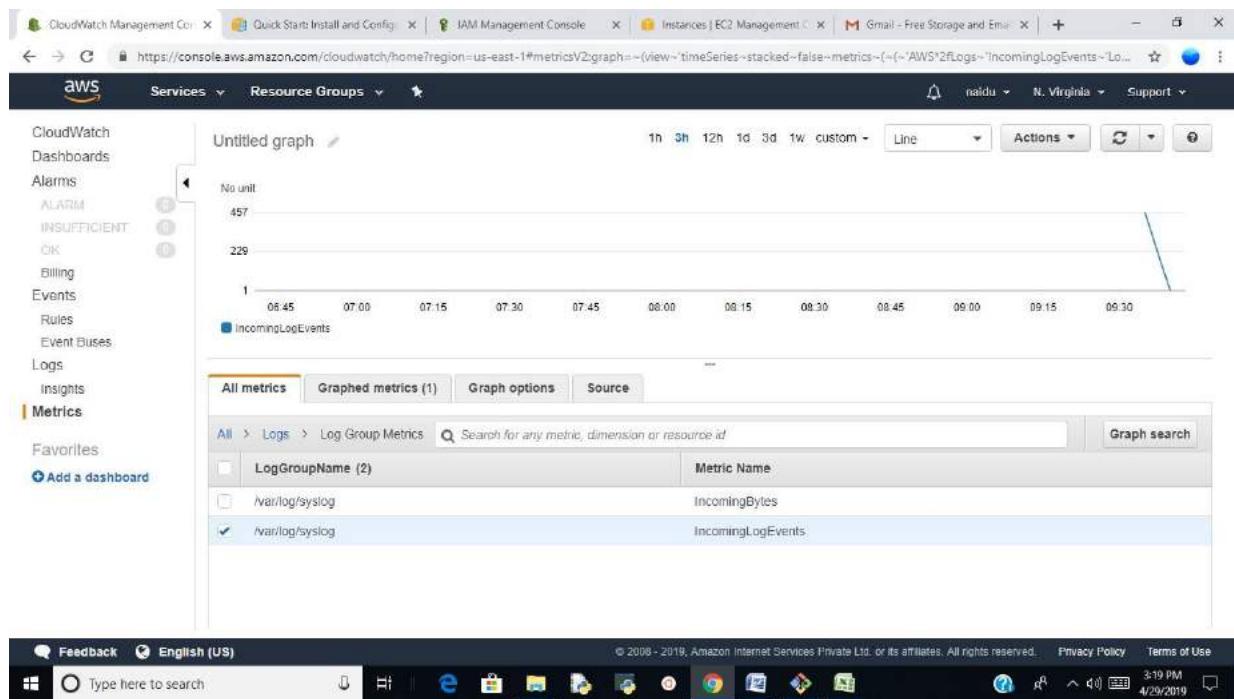


Note: we can implement multiple actions for each metrics.

Ex. we can make alarm for EBS, EC2, ELB and AutoScaling metrics with actions of Notifications, Auto Scaling Scalein and Scale out and EC2 instance stop, terminate.

Note: we can also taking metricks from logs and make alarms on this log files

Log Metrics: IncomingBytes and IncomingLogEvents



AWS CloudTrail

AWS CloudTrail is an AWS service that helps you enable governance, compliance, and operational and risk auditing of your AWS account. Actions taken by a user, role, or an AWS service are recorded as events in CloudTrail. Events include actions taken in the AWS Management Console, AWS Command Line Interface, and AWS SDKs and APIs.

CloudTrail is enabled on your AWS account when you create it. When activity occurs in your AWS account, that activity is recorded in a CloudTrail event. You can easily view recent events in the CloudTrail console by going to Event history.

CloudTrail Workflow

View event history for your AWS account

You can view and search the last 90 days of events recorded by CloudTrail in the CloudTrail console or by using the AWS CLI.

Download events

You can download a CSV or JSON file containing up to the past 90 days of CloudTrail events for your AWS account.

Create a trail

A trail enables CloudTrail to deliver log files to your Amazon S3 bucket. By default, when you create a trail in the console, the trail applies to all regions. The trail logs events from all regions in the AWS partition and delivers the log files to the S3 bucket that you specify.

Create and subscribe to an Amazon SNS topic

Subscribe to a topic to receive notifications about log file delivery to your bucket. Amazon SNS can notify you in multiple ways, including programmatically with Amazon Simple Queue Service.

View your log files

Use Amazon S3 to retrieve log files.

Manage user permissions

Use AWS Identity and Access Management (IAM) to manage which users have permissions to create, configure, or delete trails; start and stop logging; and access buckets that have log files.

Monitor events with CloudWatch Logs

You can configure your trail to send events to CloudWatch Logs. You can then use CloudWatch Logs to monitor your account for specific API calls and events.

Log management and data events

Configure your trails to log read-only, write-only, or all management and data events. By default, trails log manage

Enable log encryption

Log file encryption provides an extra layer of security for your log files.

Enable log file integrity

Log file integrity validation helps you verify that log files have remained unchanged since CloudTrail delivered them.

Share log files with other AWS accounts

You can share log files between accounts.

Aggregate logs from multiple accounts

You can aggregate log files from multiple accounts to a single bucket.

Work with partner solutions

Analyze your CloudTrail output with a partner solution that integrates with CloudTrail. Partner solutions offer a broad set of capabilities, such as change tracking, troubleshooting, and security analysis.

Viewing CloudTrail Events in the CloudTrail Console

You can use the CloudTrail console to view the last 90 days of recorded API activity and events in an AWS Region. You can also download a file with that information, or a subset of information based on the filter and time range you choose. You can customize your view of **Event history** by selecting which columns are displayed in the console. You can also look up and filter events by the resource types available for a particular service.

To view CloudTrail events

1. Sign in to the AWS Management Console and open the CloudTrail console at <https://console.aws.amazon.com/cloudtrail/home/>.
2. In the navigation pane, choose **Event history**.

A filtered list of events appears in the content pane with the latest event first. Scroll down to see more events.

Displaying CloudTrail Events

You can customize the display of **Event history** by selecting which columns to display in the CloudTrail console. By default, the following columns are displayed:

- Event time**
- User name**
- Event name**
- Resource type**
- Resource name**

You cannot change the order of the columns.

To customize the columns displayed in Event history

1. Sign in to the AWS Management Console and open the CloudTrail console at <https://console.aws.amazon.com/cloudtrail/home/>.
2. In the navigation pane, choose **Event history**.
3. Choose the gear icon.
4. In **Show/Hide Columns**, select the columns you want to display. Clear the columns you do not want to display. When you have finished, choose **Save**.

Viewing Details for an Event

1. Choose an event in the results list to show its details.
2. If the event referenced more than one resource, the additional resources are listed at the bottom of the details pane.
3. Some referenced resources have links. Choose the link to open the console for that resource.
4. Choose **View Event** in the details pane to view the event in JSON format.
5. Choose the event again to close the details pane.

Downloading Events

You can download recorded event history as a file in CSV or JSON format. Use filters and time ranges to reduce the size of the file you download.

1. Specify the filter and time range for events you want to download. For example, you can specify the event name, `StartInstances`, and specify a time range for the last three days of activity.
2. Choose the  and then choose **Export to CSV** or **Export to JSON**. The download starts immediately.
3. After your download is complete, open the file to view the events that you specified.
4. To cancel your download, choose **Cancel download**.

Creating and Updating a Trail with the Console

You can create, update, or delete your trails with the CloudTrail console. You can create up to five trails for each region. After you create a trail, CloudTrail automatically starts logging API

calls and related events in your account to the Amazon S3 bucket that you specify. To stop logging, you can turn off logging for the trail or delete it.

Creating a Trail

Follow the procedure to create a trail that applies to all regions. A trail that applies to all regions delivers log files from all regions to an S3 bucket. After you create the trail, CloudTrail automatically starts logging the events that you specified.

To create a CloudTrail trail with the AWS Management Console

1. Sign in to the AWS Management Console and open the CloudTrail console at <https://console.aws.amazon.com/cloudtrail/>.
2. Choose the region where you want the trail to be created.
3. Choose **Get Started Now**.

Tip

If you do not see **Get Started Now**, choose **Trails**, and then choose **Create trail**.

4. On the **Create Trail** page, for **Trail name**, type a name for your trail.
5. For **Apply trail to all regions**, choose **Yes** to receive log files from all regions. This is the default and recommended setting. If you choose **No**, the trail logs files only from the region in which you create the trail.
6. For **Management events**, for **Read/Write events**, choose if you want your trail to log **All**, **Read-only**, **Write-only**, or **None**, and then choose **Save**. By default, trails log all management events.
7. For **Data events**, you can specify logging data events for Amazon S3 buckets, for AWS Lambda functions, or both. By default, trails don't log data events. Additional charges apply for logging data events.
8. For **Storage location**, for **Create a new S3 bucket**, choose **Yes** to create a bucket. When you create a bucket, CloudTrail creates and applies the required bucket policies.
9. For **S3 bucket**, type a name for the bucket you want to designate for log file storage. The name must be globally unique.
10. To configure advanced settings, see [Configuring Advanced Settings for Your Trail](#). Otherwise, choose **Create**.

11. The new trail appears on the **Trails** page. The **Trails** page shows the trails in your account from all regions. In about 15 minutes, CloudTrail publishes log files that show the AWS API calls made in your account. You can see the log files in the S3 bucket that you specified.

Updating a Trail

To change trail settings, use the following procedure.

To update a trail with the AWS Management Console

1. Sign in to the AWS Management Console and open the CloudTrail console at <https://console.aws.amazon.com/cloudtrail/>.
2. Choose **Trails** and then choose a trail.
3. To make updates for **Trail settings**, choose the pencil icon, specify if you want your trail to apply to a single region or all regions, and then choose **Save**.
4. For **Management events**, choose the pencil icon, make your changes, and then choose **Save**. Your trail can log **All**, **Read-only**, **Write-only**, or **None**. By default, trails log **All** management events.
5. For **Data events**, choose the pencil icon or **Configure**, make your changes, and then choose **Save**. By default, trails don't log data events.
6. For **Storage location**, choose the pencil icon to update the settings for the following:
 - The S3 bucket (with optional prefix) that is receiving your log files.
 - Log file encryption with AWS KMS.
 - Log file validation for logs.
 - The Amazon SNS topic to notify you when log files are delivered.
7. Choose **Save**.

To configure CloudWatch Logs and tags for your trail

1. To configure CloudTrail to deliver events to CloudWatch Logs for monitoring, for **CloudWatch Logs**, choose **Configure**.

2. To configure tags (custom key-value pairs) for your trail, for **Tags**, click the pencil icon. You can add up to 50 key-value pairs per trail. Trail tags must be configured from the region in which the trail was created.
3. When finished, choose **Apply**.

Deleting a Trail

You can delete trails with the CloudTrail console. If you want to delete a trail that receives log files from all regions, you must choose the region where you originally created the trail.

To delete a trail with the CloudTrail console

1. Sign in to the AWS Management Console and open the CloudTrail console at <https://console.aws.amazon.com/cloudtrail/>.
2. Navigate to the **Trails** page of the CloudTrail console for the region in which the trail was created.
3. Choose the trail name.
4. At the top of the configuration page, click the trash icon.
5. Choose **Delete** to delete the trail permanently. The trail will be removed from the list of trails for the region. Log files that were already delivered to the Amazon S3 bucket will not be deleted.

Finding Your CloudTrail Log Files

CloudTrail publishes log files to your S3 bucket in a gzip archive. In the S3 bucket, the log file has a formatted name that includes the following elements:

- The bucket name that you specified when you created trail (found on the Trails page of the CloudTrail console)
- The (optional) prefix you specified when you created your trail
- The string "AWSLogs"
- The account number
- The string "CloudTrail"
- A region identifier such as us-west-1
- The year the log file was published in YYYY format
- The month the log file was published in MM format

- The day the log file was published in DD format
- An alphanumeric string that disambiguates the file from others that cover the same time period

The following example shows a complete log file object name:

```
bucket_name/prefix_name/AWSLogs/Account  
ID/CloudTrail/region/YYYY/MM/DD/file_name.json.gz
```

To retrieve a log file, you can use the Amazon S3 console, the Amazon S3 command line interface (CLI), or the API.

To find your log files with the Amazon S3 console

1. Open the Amazon S3 console.
2. Choose the bucket you specified.
3. Navigate through the object hierarchy until you find the log file you want.

All log files have a .gz extension.

You will navigate through an object hierarchy that is similar to the following example, but with a different bucket name, account ID, region, and date.

```
All Buckets  
Bucket_Name  
AWSLogs  
123456789012  
CloudTrail  
us-west-1  
2014  
06  
20
```

A log file for the preceding object hierarchy will look like the following:

```
123456789012_CloudTrail_us-west-1_20140620T1255ZDkvFTXOA3Vnhbc.json.gz
```

To download and read a log file

1. Open the Amazon S3 console at <https://console.aws.amazon.com/s3/>.
2. Choose the bucket and choose the log file that you want to download.
3. Choose **Download** or **Download as** and follow the prompts to save the file. This saves the file in compressed format.
4. Use a product such as 7-Zip to extract the log file.
5. Open the log file in a text editor such as Notepad++.

AWS Command Line Interface

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, you can start using functionality equivalent to that provided by the browser-based AWS Management Console from the command prompt in your favorite terminal program:

- **Linux shells** – Use common shell programs such as `bash`, `zsh`, and `tsch` to run commands in Linux, macOS, or Unix.
- **Windows command line** – On Windows, run commands in PowerShell or at the Windows command prompt.
- **Remotely** – Run commands on Amazon Elastic Compute Cloud (Amazon EC2) instances through a remote terminal such as PuTTY or SSH, or with AWS Systems Manager.

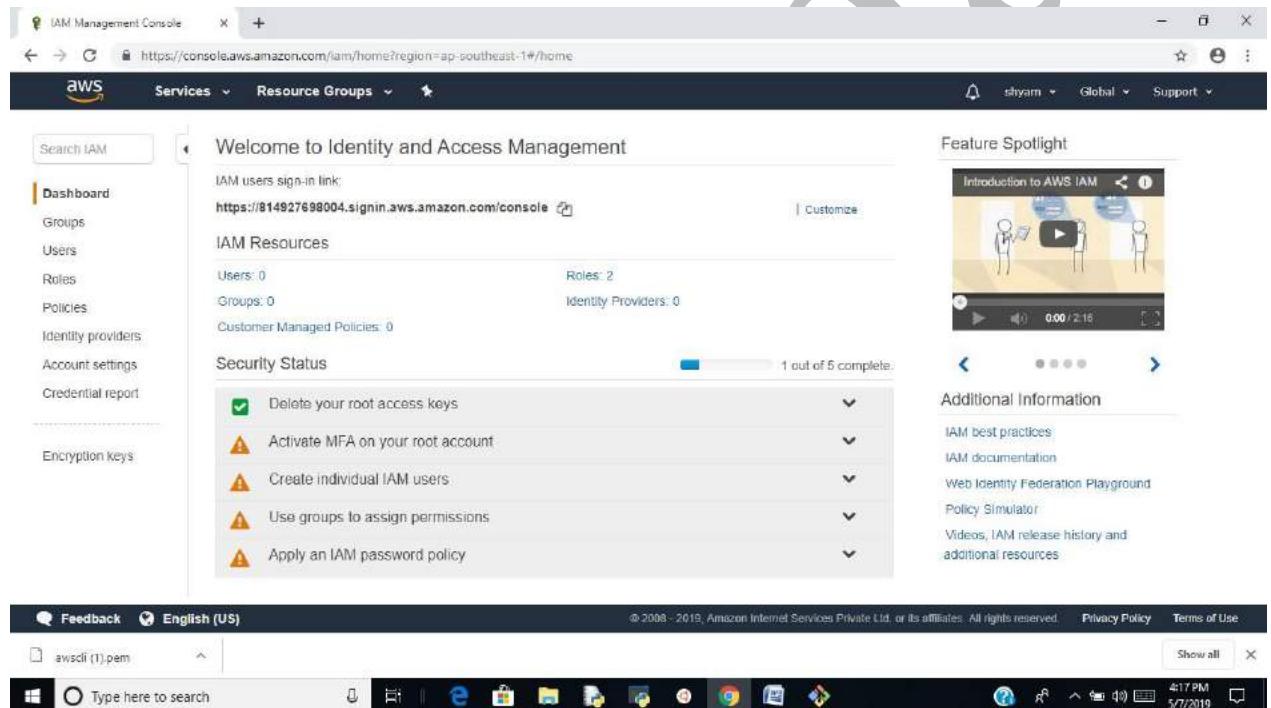
All IaaS (infrastructure as a service) AWS administration, management, and access functions in the AWS Management Console are available in the AWS API and CLI. New AWS IaaS features and services provide full AWS Management Console functionality through the API and CLI at launch or within 180 days of launch.

Install the AWS CLI on Amazon Linux

- Launch EC2 Instance and connect to that EC2 Instance using putty or git bash
- Install the awscli on your EC2 Instance by using below command

sudo apt-get install -y awscli

- Configure your awscli using (**aws configure**) command
- We need aws accesskey and aws secretaccesskey and region to configure
- First choose IAM service from aws console



- Click on delete your root access keys and click on manage security credentials

Welcome to Identity and Access Management

IAM users sign-in link:
https://B14927698004.signin.aws.amazon.com/console

IAM Resources

- Users: 0
- Groups: 0
- Roles: 2
- Policies: 0
- Identity providers: 0
- Customer Managed Policies: 0

Security Status

- Delete your root access keys
- Activate MFA on your root account
- Create individual IAM users

1 out of 6 complete.

Feature Spotlight

Introduction to AWS IAM

Additional Information

- IAM best practices
- IAM documentation
- Web Identity Federation Playground
- Policy Simulator
- Videos, IAM release history and additional resources

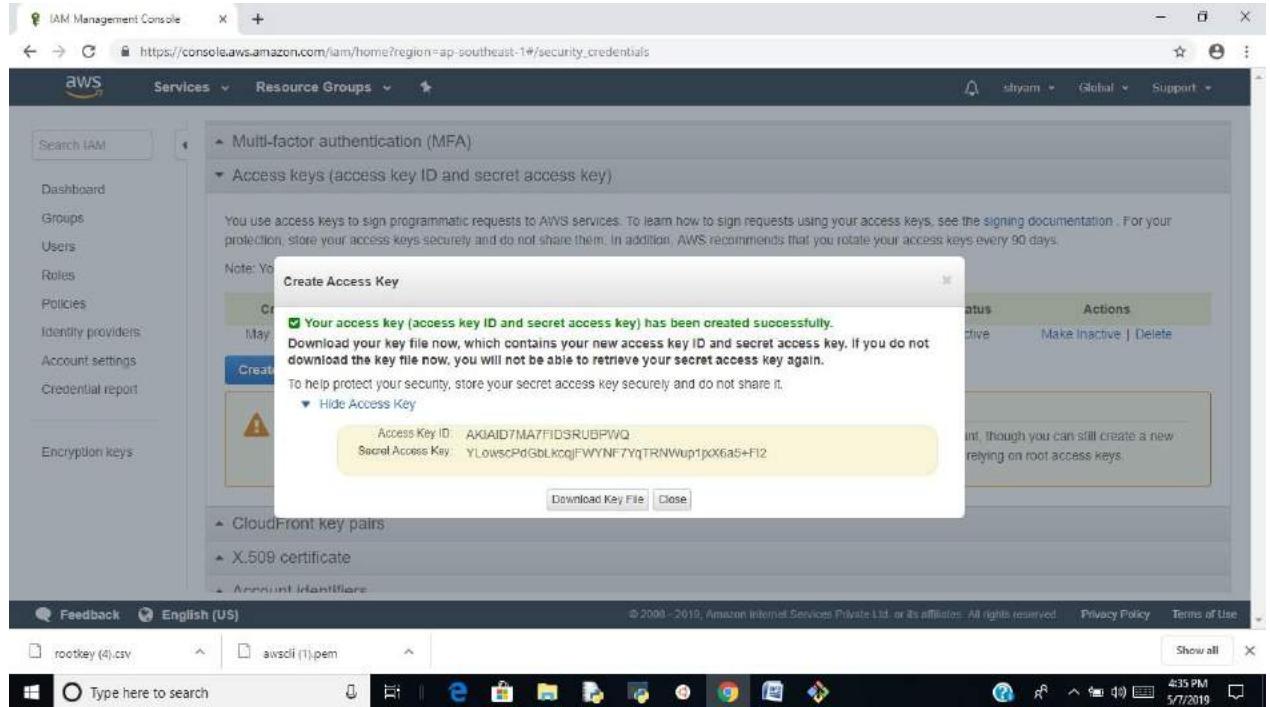
- Click on continue security credentials and click on access keys and click on create new access keys

Created	Deleted	Access Key ID	Last Used	Last Used Region	Last Used Service	Status	Actions
May 7th 2019		AKIAJLQKD9IVMACI7AA	N/A	N/A	N/A	Active	Make Inactive Delete

Important Change - Managing Your AWS Secret Access Keys

As described in a previous announcement, you cannot retrieve the existing secret access keys for your AWS root account, though you can still create a new root access key at any time. As a best practice, we recommend creating an IAM user that has access keys rather than relying on root access keys.

- Click on show keys and copy that keys



- Then fire command (aws configure) on your EC2 Instance to configure AWS
- AWS Access Key Id: enter your aws accesskey
- AWS Secret Access Key Id: enter your aws secret acess key
- Default region Name: enter your region name
- Default output format : enter JSON or YAML.....etc which format do you want.

```

ubuntu@ip-172-31-6-20:~$ top
PID USER PR NI VIRT RES SHR S %CPU %MEM TIME+ COMMAND
2565 ubuntu 20 0 23540 1604 1172 R 0.3 0.2 0:00.39 top
1 root 20 0 33512 2840 1480 S 0.0 0.3 0:01.42 init
2 root 20 0 0 0 0 S 0.0 0.0 0:00.00 kthreadd
3 root 20 0 0 0 0 S 0.0 0.0 0:00.00 ksoftirqd/0
5 root 0 -20 0 0 0 S 0.0 0.0 0:00.00 kworker/0:0H
6 root 20 0 0 0 0 S 0.0 0.0 0:00.02 kworker/u30+
7 root 20 0 0 0 0 S 0.0 0.0 0:00.08 rcu_sched
8 root 20 0 0 0 0 S 0.0 0.0 0:00.13 rcuos/0
9 root 20 0 0 0 0 S 0.0 0.0 0:00.00 rcuos/1
10 root 20 0 0 0 0 S 0.0 0.0 0:00.00 rcuos/2
11 root 20 0 0 0 0 S 0.0 0.0 0:00.00 rcuos/3
12 root 20 0 0 0 0 S 0.0 0.0 0:00.00 rcuos/4
13 root 20 0 0 0 0 S 0.0 0.0 0:00.00 rcuos/5
14 root 20 0 0 0 0 S 0.0 0.0 0:00.00 rcuos/6
15 root 20 0 0 0 0 S 0.0 0.0 0:00.00 rcuos/7
16 root 20 0 0 0 0 S 0.0 0.0 0:00.00 rcuos/8
17 root 20 0 0 0 0 S 0.0 0.0 0:00.00 rcuos/9

ubuntu@ip-172-31-6-20:~$ aws configure
AWS Access Key ID [None]: AKIAID7MA7FIDSRRUPWQ
AWS Secret Access Key [None]: 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
Default region name [None]: ap-southeast-1
Default output format [None]: JSON
ubuntu@ip-172-31-6-20:~$ |
```

The screenshot shows an AWS EC2 Management Console window with a terminal session. The terminal displays the output of the 'top' command, which lists various system processes and their resource usage. Below 'top', the user runs the 'aws configure' command, which prompts for AWS Access Key ID and AWS Secret Access Key, setting the default region to 'ap-southeast-1' and the output format to 'JSON'. The AWS Management Console sidebar is visible on the left, and the Windows taskbar at the bottom shows other open applications like 'rootkey (4).csv' and 'awscli (1).pem'.

Usecase for S3 service

- To see the list of buckets in S3 service use below command

`aws s3 ls`

AWS Command Line Interface - Microsoft Word

```

ubuntu@ip-172-31-211:~$ free -m
              total        free       used      avail Mem
KiB Swap:      0 total,      0 free,      0 used.  786844 avail Mem

  PID USER      PR  NI    VIRT    RES    SHR S %CPU %MEM   TIME+ COMMAND
  1 root      20   0 37668  5640 3932 S  0.0  0.6  0:03.03 systemd
  2 root      20   0      0     0     0 S  0.0  0.0  0:00.00 kthreadd
  3 root      20   0      0     0     0 S  0.0  0.0  0:00.05 ksoftirqd/0
  5 root      0 -20      0     0     0 S  0.0  0.0  0:00.00 kworker/0:0H
  7 root      20   0      0     0     0 S  0.0  0.0  0:00.24 rcu_sched
  8 root      20   0      0     0     0 S  0.0  0.0  0:00.00 rcu_bh
  9 root      rt   0      0     0     0 S  0.0  0.0  0:00.00 migration/0
 10 root      rt   0      0     0     0 S  0.0  0.0  0:00.03 watchdog/0
 11 root      20   0      0     0     0 S  0.0  0.0  0:00.00 kdevtmpfs
 12 root      0 -20      0     0     0 S  0.0  0.0  0:00.00 netns
 13 root      0 -20      0     0     0 S  0.0  0.0  0:00.00 perf
 14 root      20   0      0     0     0 S  0.0  0.0  0:00.01 xenwatch
 15 root      20   0      0     0     0 S  0.0  0.0  0:00.00 xenbus
 16 root      20   0      0     0     0 S  0.0  0.0  0:00.14 kworker/0:1
 17 root      20   0      0     0     0 S  0.0  0.0  0:00.00 khungtaskd
 18 root      0 -20      0     0     0 S  0.0  0.0  0:00.00 writeback
 19 root      25   5      0     0     0 S  0.0  0.0  0:00.00 ksmd
ubuntu@ip-172-31-211:~$ aws s3 ls
2019-05-09 05:41:55 bb156
2019-05-09 07:19:19 mybb321
ubuntu@ip-172-31-211:~$ 
```

- You can also check list of buckets through s3 console

S3 Management Console

AWS Transfer for SFTP is a fully managed, secure file transfer with storage in Amazon S3. Learn more » Documentation

Buckets

S3 buckets

S3 buckets			
Actions		Access	Region
<input type="checkbox"/>	Bucket name	Bucket and objects not public	Asia Pacific (Singapore) May 9, 2019 11:11:54 AM GMT+0630
<input type="checkbox"/>	bb156	Objects can be public	Asia Pacific (Singapore) May 9, 2019 11:06:46 AM GMT+0630
<input type="checkbox"/>	mybb321		

- To create buckets in s3 use the mb (make bucket) command

Aws s3 mb s3://bucketname

Here bucket name shoud be unique



AWS-soft-copy - Microsoft Word

ubuntu@ip-172-31-31-211:~\$

Kib Swap:	0 total,	0 free,	0 used.	783180 avail Mem
PID USER	PR NI	VIRT RES	SHR S %CPU %MEM	TIME+ COMMAND
1 root	20 0	37668 5640	3932 S 0.0 0.6	0:03.03 systemd
2 root	20 0	0 0	0 S 0.0 0.0	0:00.00 kthreadd
3 root	20 0	0 0	0 S 0.0 0.0	0:00.06 ksoftirqd/0
5 root	0 -20	0 0	0 S 0.0 0.0	0:00.00 kworker/0:0H
7 root	20 0	0 0	0 S 0.0 0.0	0:00.24 rcu_sched
8 root	20 0	0 0	0 S 0.0 0.0	0:00.00 rcu_bh
9 root	rt 0	0 0	0 S 0.0 0.0	0:00.00 migration/0
10 root	rt 0	0 0	0 S 0.0 0.0	0:00.04 watchdog/0
11 root	20 0	0 0	0 S 0.0 0.0	0:00.00 kdevtmpfs
12 root	0 -20	0 0	0 S 0.0 0.0	0:00.00 netns
13 root	0 -20	0 0	0 S 0.0 0.0	0:00.01 perf
14 root	20 0	0 0	0 S 0.0 0.0	0:00.01 xenwatch
15 root	20 0	0 0	0 S 0.0 0.0	0:00.00 xenbus
16 root	20 0	0 0	0 S 0.0 0.0	0:00.16 kworker/0:1
17 root	20 0	0 0	0 S 0.0 0.0	0:00.00 khungtaskd
18 root	0 -20	0 0	0 S 0.0 0.0	0:00.00 writeback
19 root	25 5	0 0	0 S 0.0 0.0	0:00.00 ksmd

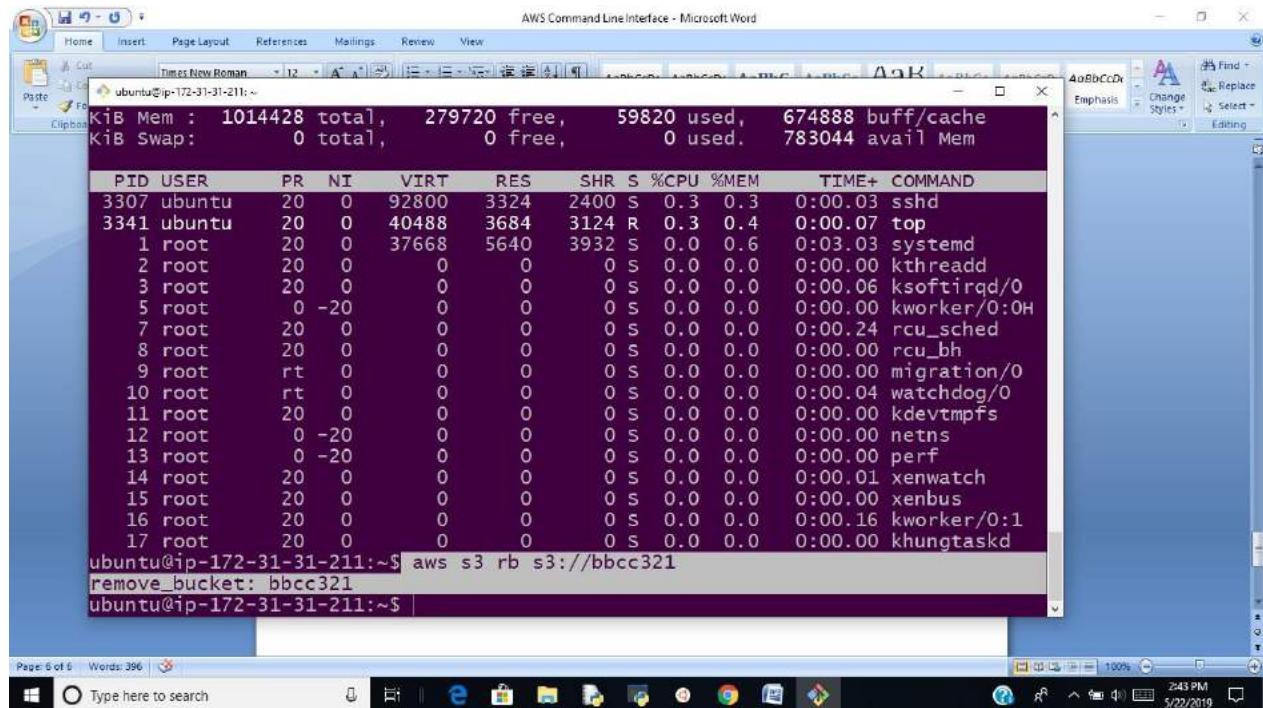
ubuntu@ip-172-31-31-211:~\$ aws s3 mb s3://bbcc321
make_bucket: bbcc321
ubuntu@ip-172-31-31-211:~\$

aws cli/1.16.116 Python/3.6.8 Linux/4.14.77-81.59.amzn2.x86_64 botocore/1.12.106

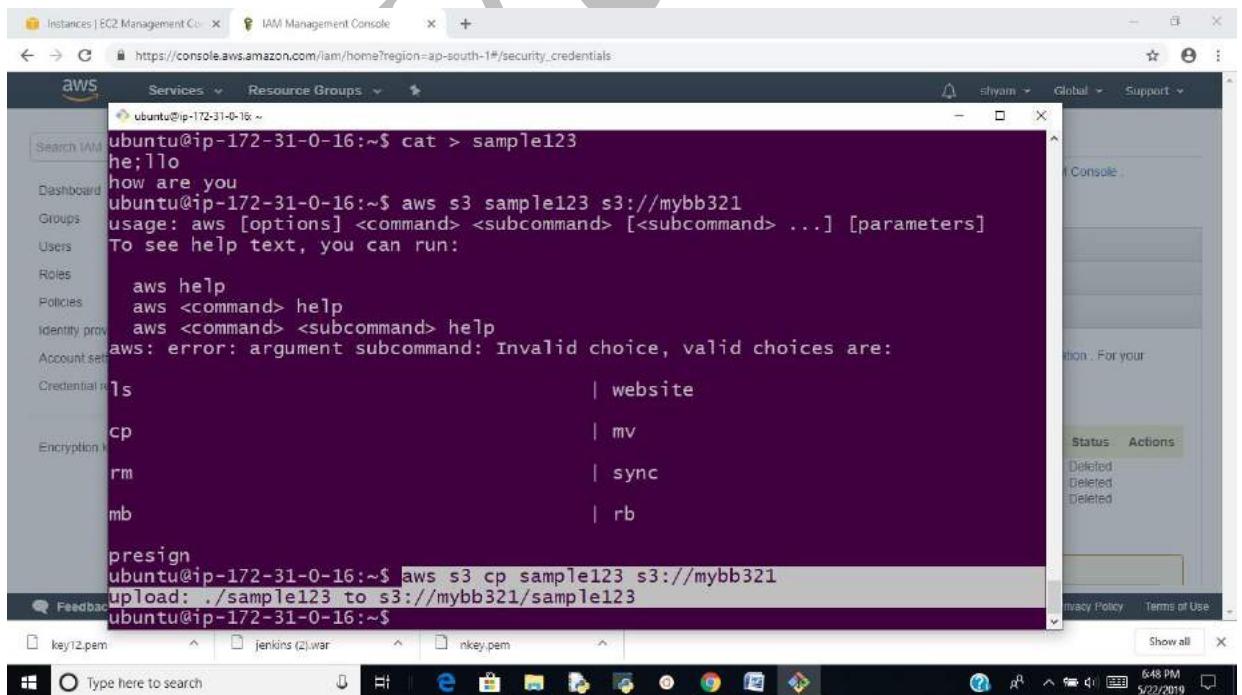
Type here to search

To remove bucket from s3

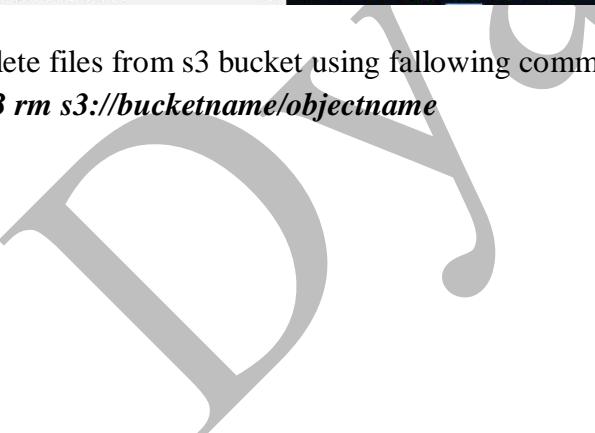
aws s3 rb s3://bucketname



- To upload files to s3 bucket
`aws s3 cp sourcefile/upload_data s3://bucketname/destination`



- To download files from s3 bucket use cp command in reverse order
aws s3 cp s3://bucketname/file destination/hostlocation



AWS Command Line Interface - Microsoft Word

```
ubuntu@ip-172-31-0-16:~$ Connection reset by 13.127.31.49 port 22
nagesh@nagesh-PC MINGW64 ~/Downloads (master)
$ ssh -i "key12.pem" ubuntu@ec2-13-127-31-49.ap-south-1.compute.amazonaws.com
Welcome to Ubuntu 16.04.5 LTS (GNU/Linux 4.4.0-1075-aws x86_64)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/advantage

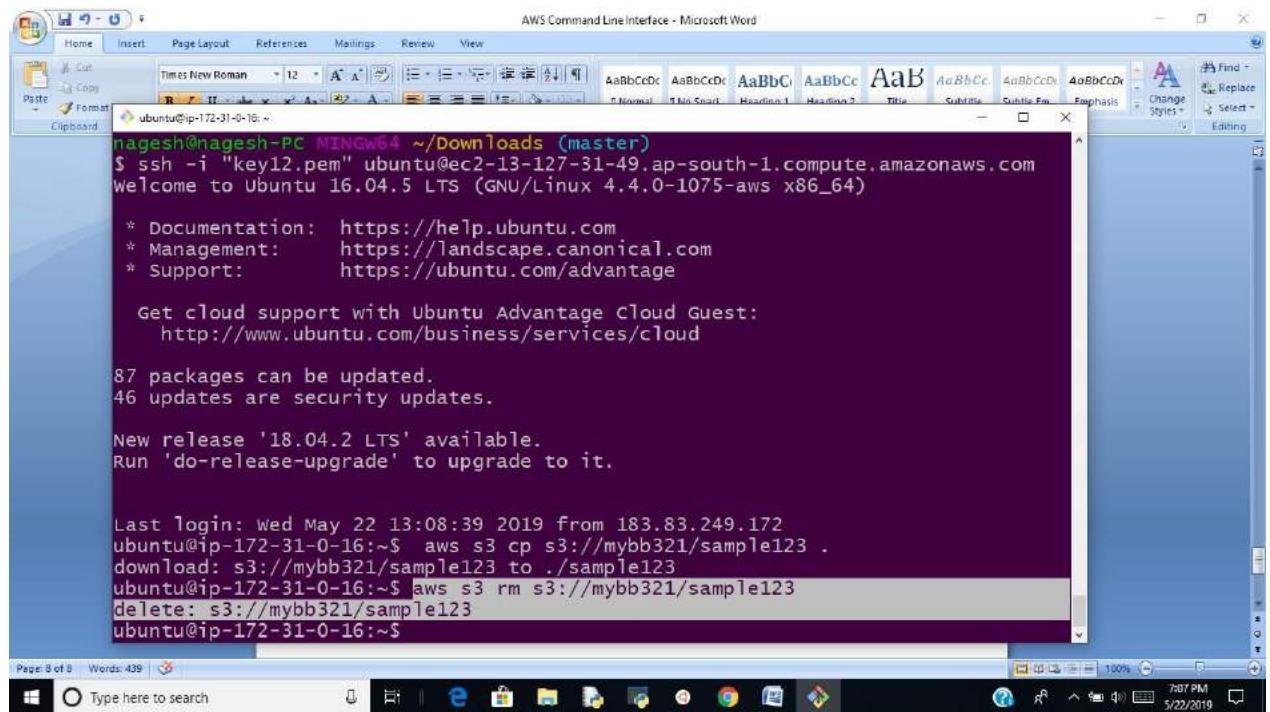
 Get cloud support with Ubuntu Advantage Cloud Guest:
 http://www.ubuntu.com/business/services/cloud

87 packages can be updated.
46 updates are security updates.

New release '18.04.2 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: wed May 22 13:08:39 2019 from 183.83.249.172
ubuntu@ip-172-31-0-16:~$ aws s3 cp s3://mybb321/sample123 .
download: s3://mybb321/sample123 to ./sample123
ubuntu@ip-172-31-0-16:~$
```

- To delete files from s3 bucket using fallowing command
aws s3 rm s3://bucketname/objectname



A screenshot of a Microsoft Word document titled "AWS Command Line Interface - Microsoft Word". The document contains a terminal session from an Ubuntu 16.04.5 LTS instance. The session shows the user logging in via SSH, updating packages, and interacting with AWS S3. The terminal output is as follows:

```
nagesh@nagesh-PC MINGW64 ~/Downloads (master)
$ ssh -i "key12.pem" ubuntu@ec2-13-127-31-49.ap-south-1.compute.amazonaws.com
Welcome to Ubuntu 16.04.5 LTS (GNU/Linux 4.4.0-1075-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

 Get cloud support with Ubuntu Advantage Cloud Guest:
 http://www.ubuntu.com/business/services/cloud

87 packages can be updated.
46 updates are security updates.

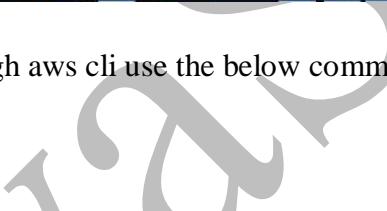
New release '18.04.2 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: wed May 22 13:08:39 2019 from 183.83.249.172
ubuntu@ip-172-31-0-16:~$ aws s3 cp s3://mybb321/sample123 .
download: s3://mybb321/sample123 to ./sample123
ubuntu@ip-172-31-0-16:~$ aws s3 rm s3://mybb321/sample123
delete: s3://mybb321/sample123
ubuntu@ip-172-31-0-16:~$
```

Usecase for EC2 Service

- Launch EC2 Instance by using command line interface

```
aws ec2 run-instances --image-id ami-xxxxxxxx --count 1 --instance-type t2.micro --
key-name MyKeyPair --security-group-ids sg-903004f8 --subnet-id subnet-6e7f829e
```

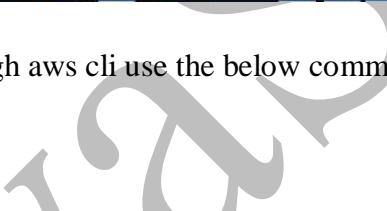


```
ubuntu@ip-172-31-0-16:~$ aws help
aws <command> help
aws <command> <subcommand> help

Unknown options: --availability-zone, ap-south-1a
ubuntu@ip-172-31-0-16:~$ aws ec2 run-instances --image-id ami-0a574895390037a62
--instance-type t2.micro --count 1 --security-group-ids sg-676ff30b --key-name key12 --subnet-id subnet-de0b59b6
{
    "Groups": [],
    "Instances": [
        {
            "InstanceType": "t2.micro",
            "StateTransitionReason": "",
            "ImageId": "ami-0a574895390037a62",
            "PrivateIpAddress": "172.31.29.103",
            "KeyName": "key12",
            "EbsOptimized": false,
            "SourceDestCheck": true,
            "Placement": {
                "Tenancy": "default",
                "AvailabilityZone": "ap-south-1a",
                "GroupName": ""
            },
            "Architecture": "x86_64",
            "VpcId": "vpc-1637027e",
            "SecurityGroups": [
                {
                    "GroupId": "sg-676ff30b",
                    "Status": "active"
                }
            ],
            "RootDeviceType": "Amazon EBS",
            "RootDeviceName": "/dev/sda1",
            "BlockDeviceMappings": [
                {
                    "DeviceName": "/dev/sda1",
                    "VirtualName": "/dev/xvda",
                    "Ebs": {
                        "VolumeSize": 8,
                        "DeleteOnTermination": true,
                        "VolumeType": "standard"
                    }
                }
            ],
            "NetworkInterfaces": [
                {
                    "Description": "Primary network interface",
                    "AssociatePublicIpAddress": false,
                    "DeleteOnTermination": true,
                    "DeviceIndex": 0,
                    "MacAddress": "56:84:7A:4B:9C:0D",
                    "SubnetId": "subnet-de0b59b6",
                    "VpcId": "vpc-1637027e",
                    "OwnerId": "814927698004",
                    "InterfaceType": "primary",
                    "PrivateIpAddress": "172.31.29.103",
                    "PrivateIpAddresses": [
                        {
                            "Primary": true
                        }
                    ],
                    "Attachment": {
                        "AttachmentId": "eni-attach-00000000000000000000000000000000",
                        "DeviceIndex": 0,
                        "DeleteOnTermination": true,
                        "DeviceName": "/dev/xvda",
                        "NetworkCardType": "Amazon VPC Network Adapter"
                    }
                }
            ],
            "Monitoring": {
                "Enabled": true
            },
            "EnclaveOptions": {
                "Enabled": false
            },
            "HibernationOptions": {
                "Config": {
                    "Hibernation": true
                }
            },
            "MetadataOptions": {
                "HttpTokens": "optional",
                "HttpProtocolVersion": "1.1"
            },
            "CapacityReservationPreference": "auto"
        }
    ],
    "ResponseMetadata": {
        "RequestId": "43f34a2d-4a2c-4a2c-8a2c-4a2c4a2c4a2c",
        "HTTPStatusCode": 200,
        "HTTPHeaders": {
            "Content-Type": "application/json",
            "Content-Length": "1024",
            "Date": "Tue, 28 May 2019 12:38:19 GMT"
        },
        "RetryAttempts": 0
    }
}
```

- To see the list of instances through aws cli use the below command

aws ec2 describe-instances

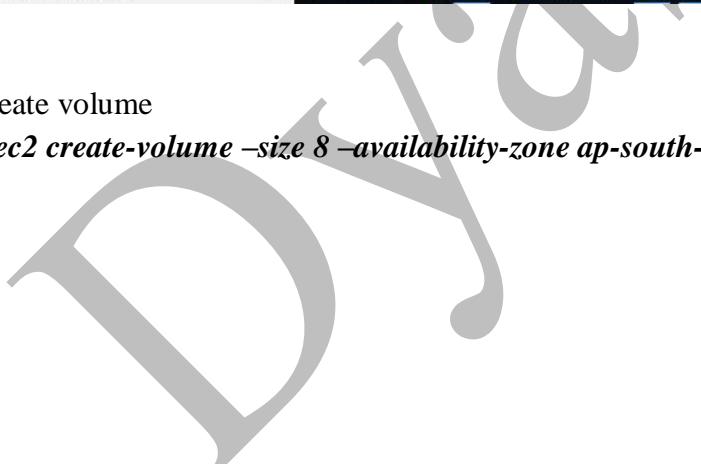


```
ubuntu@ip-172-31-0-16:~$ 0 updates are security updates.

New release '18.04.2 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

*** System restart required ***
Last Login: Thu May 23 07:14:30 2019 from 183.83.249.172
ubuntu@ip-172-31-0-16:~$ aws ec2 describe-instances
{
    "Reservations": [
        {
            "OwnerId": "814927698004",
            "Instances": [
                {
                    "State": {
                        "Code": 16,
                        "Name": "running"
                    },
                    "InstanceType": "t2.micro",
                    "VirtualizationType": "hvm",
                    "VpcId": "vpc-1637027e",
                    "BlockDeviceMappings": [
                        {
                            "DeviceName": "/dev/sda1",
                            "VirtualName": "/dev/xvda",
                            "Ebs": {
                                "VolumeSize": 8,
                                "DeleteOnTermination": true,
                                "VolumeType": "standard"
                            }
                        }
                    ],
                    "NetworkInterfaces": [
                        {
                            "Description": "Primary network interface",
                            "AssociatePublicIpAddress": false,
                            "DeleteOnTermination": true,
                            "DeviceIndex": 0,
                            "MacAddress": "56:84:7A:4B:9C:0D",
                            "SubnetId": "subnet-de0b59b6",
                            "VpcId": "vpc-1637027e",
                            "OwnerId": "814927698004",
                            "InterfaceType": "primary",
                            "PrivateIpAddress": "172.31.29.103",
                            "PrivateIpAddresses": [
                                {
                                    "Primary": true
                                }
                            ],
                            "Attachment": {
                                "AttachmentId": "eni-attach-00000000000000000000000000000000",
                                "DeviceIndex": 0,
                                "DeleteOnTermination": true,
                                "DeviceName": "/dev/xvda",
                                "NetworkCardType": "Amazon VPC Network Adapter"
                            }
                        }
                    ],
                    "Monitoring": {
                        "Enabled": true
                    },
                    "EnclaveOptions": {
                        "Enabled": false
                    },
                    "HibernationOptions": {
                        "Config": {
                            "Hibernation": true
                        }
                    },
                    "MetadataOptions": {
                        "HttpTokens": "optional",
                        "HttpProtocolVersion": "1.1"
                    },
                    "CapacityReservationPreference": "auto"
                }
            ],
            "ResponseMetadata": {
                "RequestId": "43f34a2d-4a2c-4a2c-8a2c-4a2c4a2c4a2c",
                "HTTPStatusCode": 200,
                "HTTPHeaders": {
                    "Content-Type": "application/json",
                    "Content-Length": "1024",
                    "Date": "Tue, 28 May 2019 12:54:59 GMT"
                },
                "RetryAttempts": 0
            }
        }
    ],
    "ResponseMetadata": {
        "RequestId": "43f34a2d-4a2c-4a2c-8a2c-4a2c4a2c4a2c",
        "HTTPStatusCode": 200,
        "HTTPHeaders": {
            "Content-Type": "application/json",
            "Content-Length": "1024",
            "Date": "Tue, 28 May 2019 12:54:59 GMT"
        },
        "RetryAttempts": 0
    }
}
```

- To terminate the instance through aws cli use below command
aws ec2 terminate-instances --instance-ids instanceid

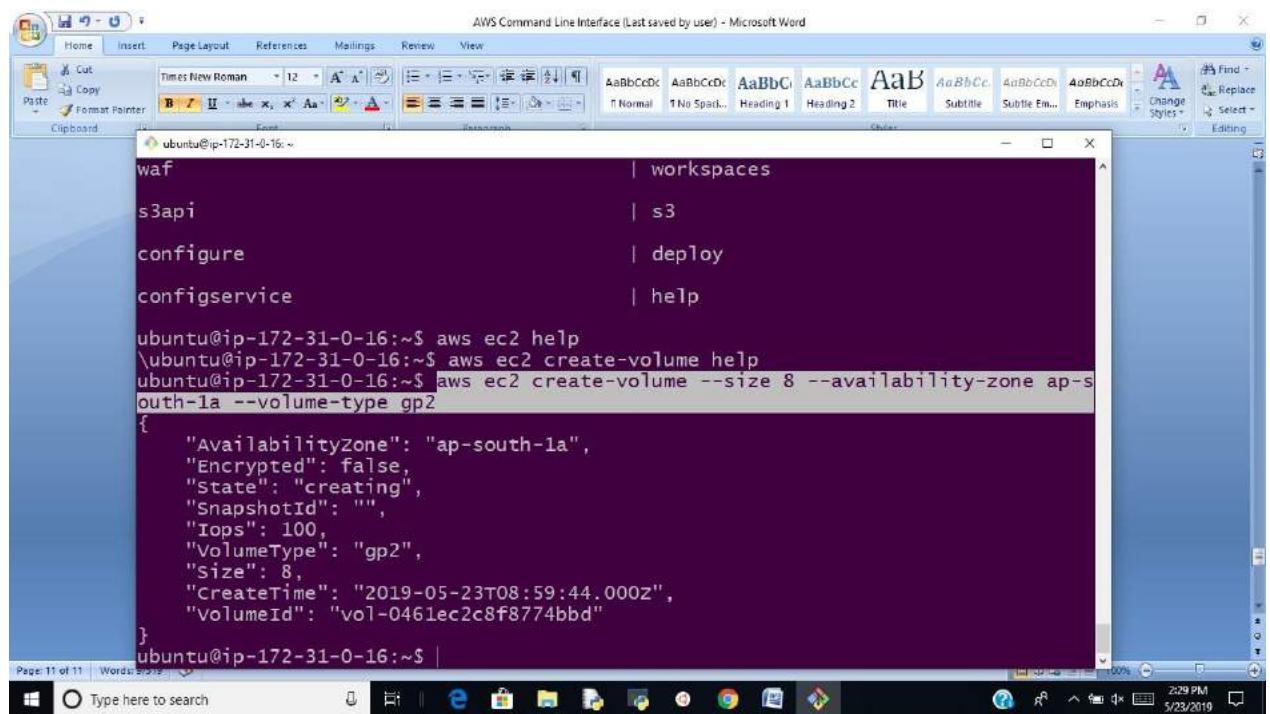


AWS Command Line Interface (Last saved by user) - Microsoft Word

```
ubuntu@ip-172-31-0-16:~$ To see help text, you can run:
aws help
aws <command> help
aws <command> <subcommand> help
aws: error: the following arguments are required: --instance-ids
ubuntu@ip-172-31-0-16:~$ aws ec2 terminate-instances --instance-ids i-0ee70b09c7fad1902
{
    "TerminatingInstances": [
        {
            "CurrentState": {
                "Code": 32,
                "Name": "shutting-down"
            },
            "PreviousState": {
                "Code": 16,
                "Name": "running"
            },
            "InstanceId": "i-0ee70b09c7fad1902"
        }
    ]
}
ubuntu@ip-172-31-0-16:~$
```

Page 10 of 10 | Word document | 100% | 1:55 PM | 5/23/2019 | Type here to search | Windows Taskbar

- To create volume
aws ec2 create-volume --size 8 --availability-zone ap-south-1a --volume-type gp2

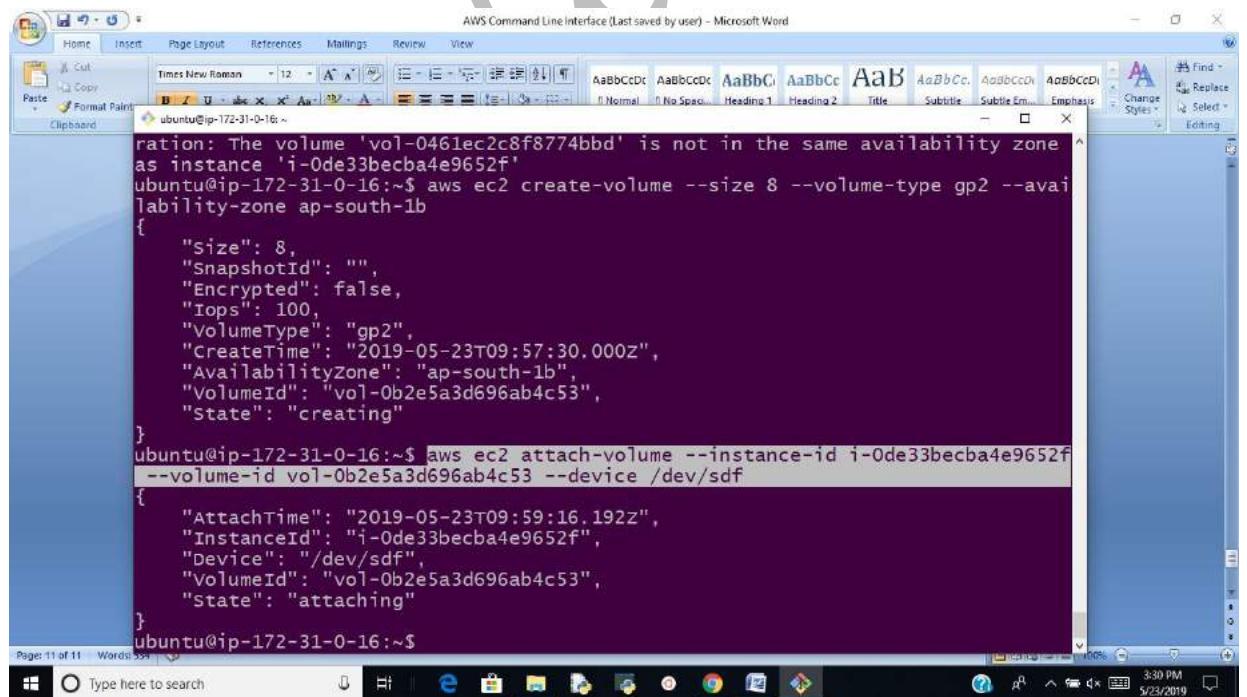


```
waf          | workspaces
s3api       | s3
configure   | deploy
configservice | help

ubuntu@ip-172-31-0-16:~$ aws ec2 help
ubuntu@ip-172-31-0-16:~$ aws ec2 create-volume help
ubuntu@ip-172-31-0-16:~$ aws ec2 create-volume --size 8 --availability-zone ap-south-1a --volume-type gp2
{
    "AvailabilityZone": "ap-south-1a",
    "Encrypted": false,
    "State": "creating",
    "SnapshotId": "",
    "Iops": 100,
    "VolumeType": "gp2",
    "Size": 8,
    "CreateTime": "2019-05-23T08:59:44.000Z",
    "VolumeId": "vol-0461ec2c8f8774bbd"
}
ubuntu@ip-172-31-0-16:~$
```

Attach volume to EC2 Instance

aws ec2 attach-volume –volume-id volumeid –instance-id instanceid –device /dev/sdf



```
ration: The volume 'vol-0461ec2c8f8774bbd' is not in the same availability zone as instance 'i-0de33becba4e9652f'
ubuntu@ip-172-31-0-16:~$ aws ec2 create-volume --size 8 --volume-type gp2 --availability-zone ap-south-1b
{
    "Size": 8,
    "SnapshotId": "",
    "Encrypted": false,
    "Iops": 100,
    "VolumeType": "gp2",
    "CreateTime": "2019-05-23T09:57:30.000Z",
    "AvailabilityZone": "ap-south-1b",
    "VolumeId": "vol-0b2e5a3d696ab4c53",
    "State": "creating"
}
ubuntu@ip-172-31-0-16:~$ aws ec2 attach-volume --instance-id i-0de33becba4e9652f --volume-id vol-0b2e5a3d696ab4c53 --device /dev/sdf
{
    "AttachTime": "2019-05-23T09:59:16.192Z",
    "InstanceId": "i-0de33becba4e9652f",
    "Device": "/dev/sdf",
    "VolumeId": "vol-0b2e5a3d696ab4c53",
    "State": "attaching"
}
ubuntu@ip-172-31-0-16:~$
```

Note: do you want to attach volume then both volume and instance are in same availability zone

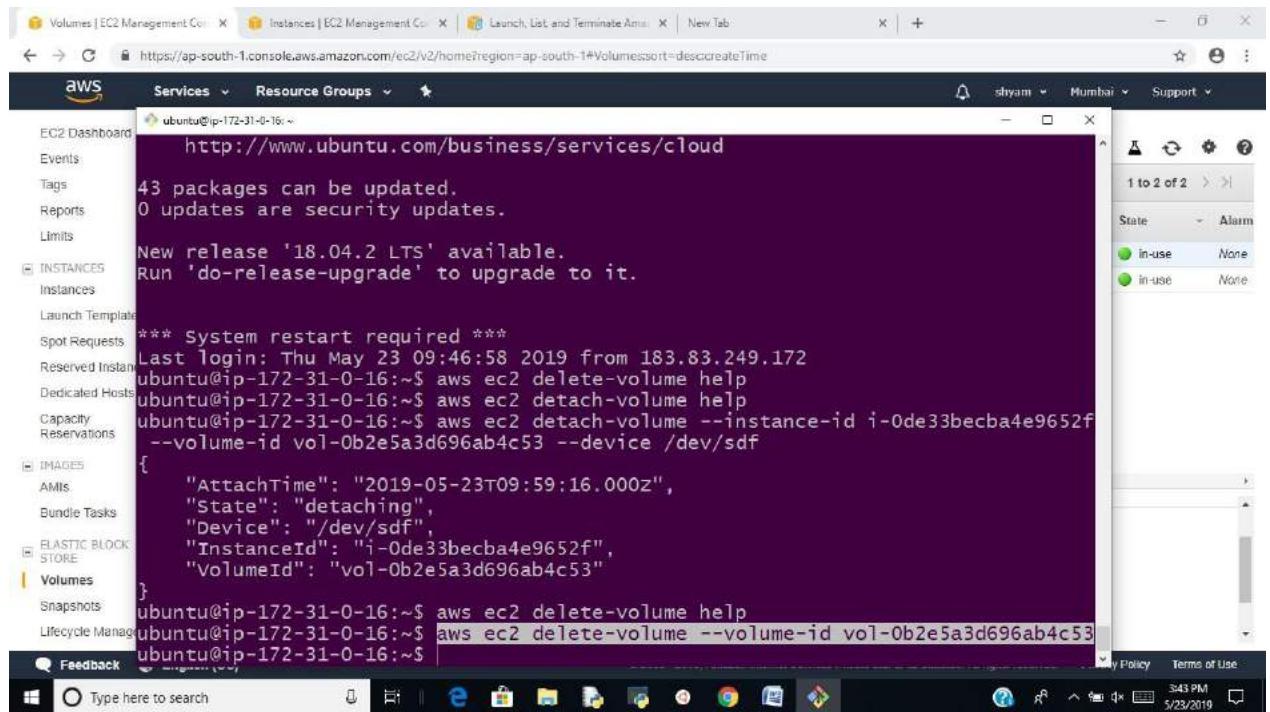
- To detach volume

aws ec2 detach-volume –instance-id instanceid –volume-id volumeid –device devicetype

```
ubuntu@ip-172-31-0-16:~$ Get cloud support with Ubuntu Advantage Cloud Guest:  
http://www.ubuntu.com/business/services/cloud  
43 packages can be updated.  
0 updates are security updates.  
New release '18.04.2 LTS' available.  
Run 'do-release-upgrade' to upgrade to it.  
*** System restart required ***  
Last login: Thu May 23 09:46:58 2019 from 183.83.249.172  
ubuntu@ip-172-31-0-16:~$ aws ec2 delete-volume help  
ubuntu@ip-172-31-0-16:~$ aws ec2 detach-volume help  
ubuntu@ip-172-31-0-16:~$ aws ec2 detach-volume --instance-id i-0de33becba4e9652f  
--volume-id vol-0b2e5a3d696ab4c53 --device /dev/sdf  
{  
    "AttachTime": "2019-05-23T09:59:16.000Z",  
    "State": "detaching",  
    "Device": "/dev/sdf",  
    "InstanceId": "i-0de33becba4e9652f",  
    "VolumeId": "vol-0b2e5a3d696ab4c53"  
}  
ubuntu@ip-172-31-0-16:~$ |
```

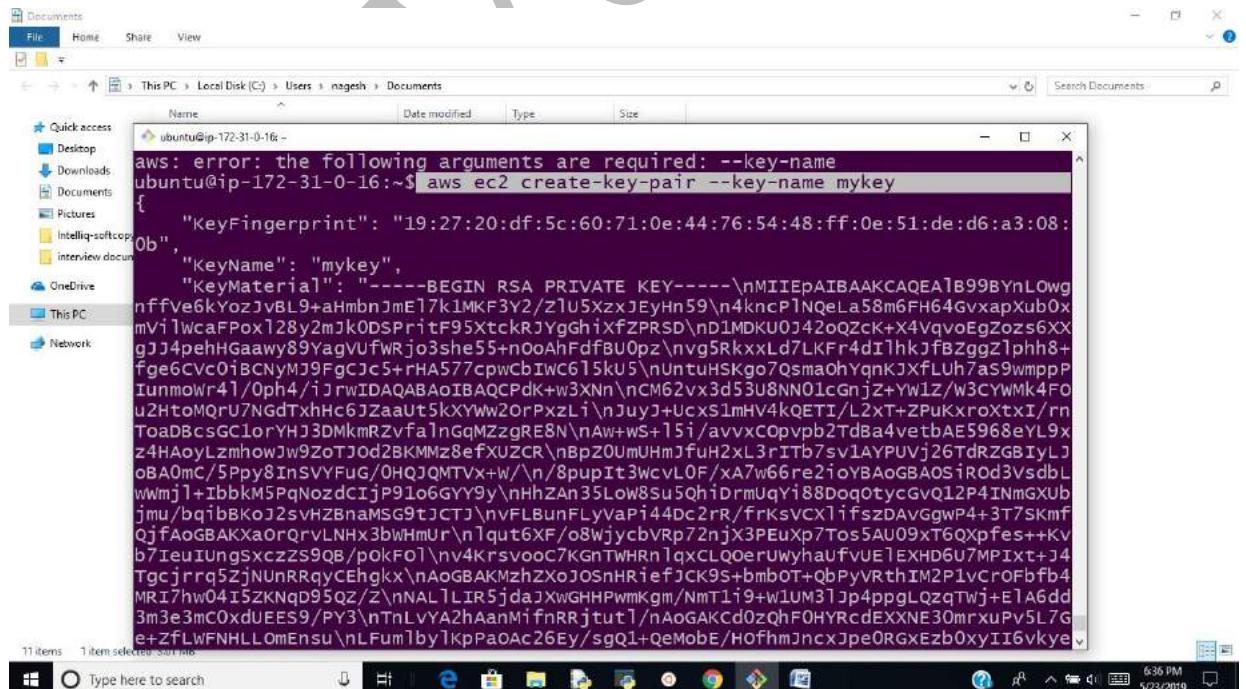
- To delete volume

aws ec2 delete-volume –volume-id volumeid



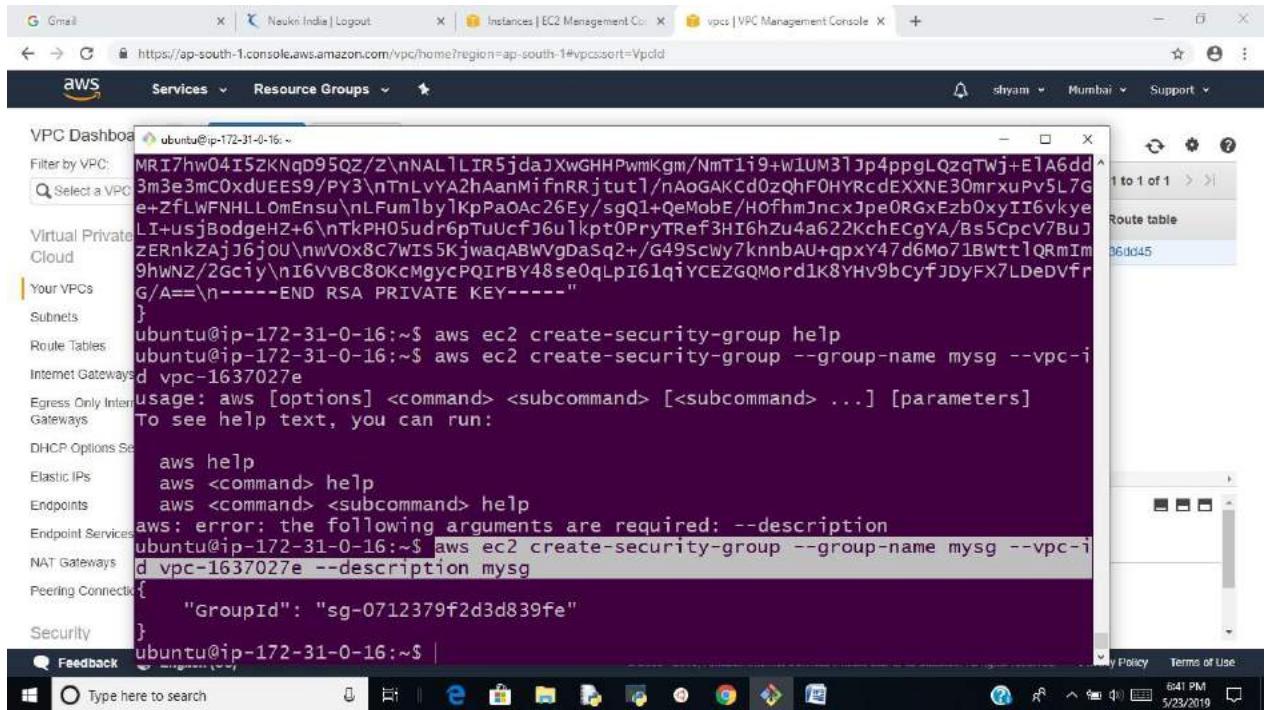
- To create key pair

aws ec2 create-key-pair –key-name keypairname



- To create securitygroup

aws ec2 create-security-group --group-name groupname --vpc-id vpcid --description message



A screenshot of a Linux terminal window titled "ubuntu@ip-172-31-0-16:~\$". The terminal shows the following command being run:

```
aws ec2 create-security-group --group-name mysg --vpc-id vpc-1637027e
```

The terminal then displays the usage information for the command:

```
usage: aws [options] <command> <subcommand> [<subcommand> ...] [parameters]
```

To see help text, you can run:

```
aws help
aws <command> help
aws <command> <subcommand> help
```

The terminal then shows an error message:

```
aws: error: the following arguments are required: --description
```

Finally, the command is run again with the --description option:

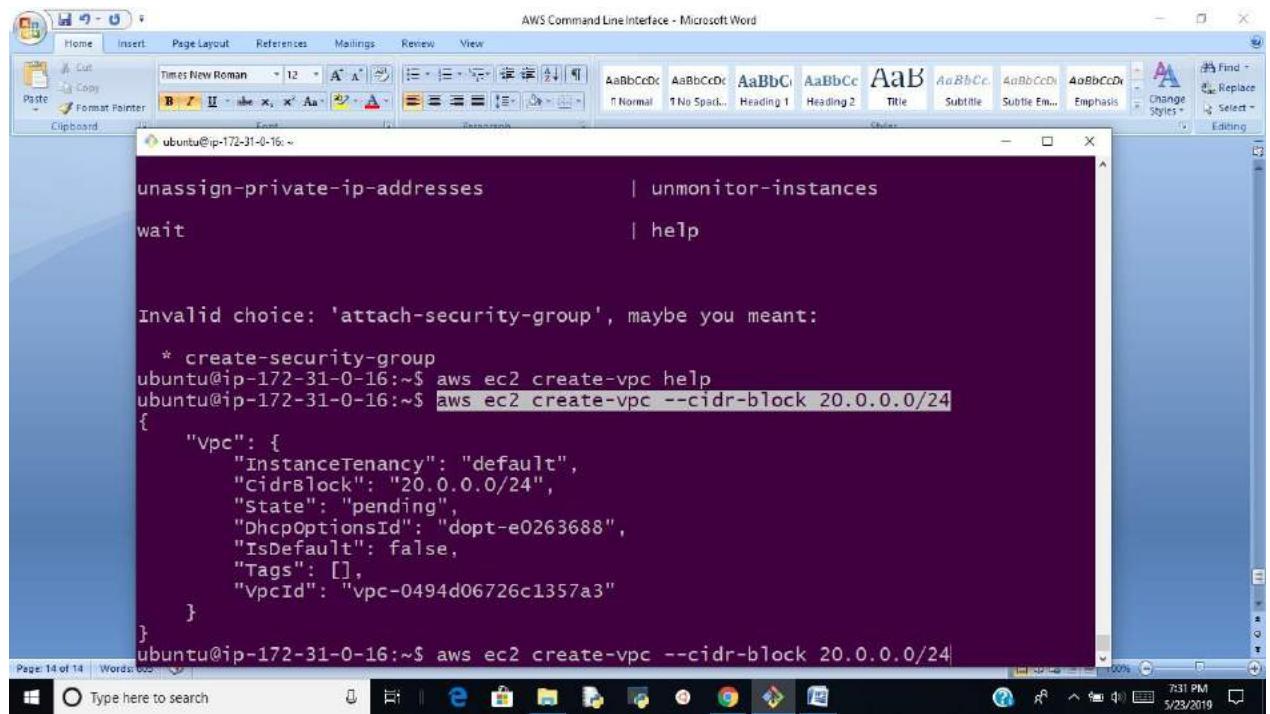
```
ubuntu@ip-172-31-0-16:~$ aws ec2 create-security-group --group-name mysg --vpc-id vpc-1637027e --description mysg
```

The terminal then displays the response from the AWS API:

```
{"GroupId": "sg-0712379f2d3d839fe"}
```

- To create vpc

aws ec2 create-vpc --cidr-block cidrblockrange(10.0.0.0/24)



AWS Command Line Interface - Microsoft Word

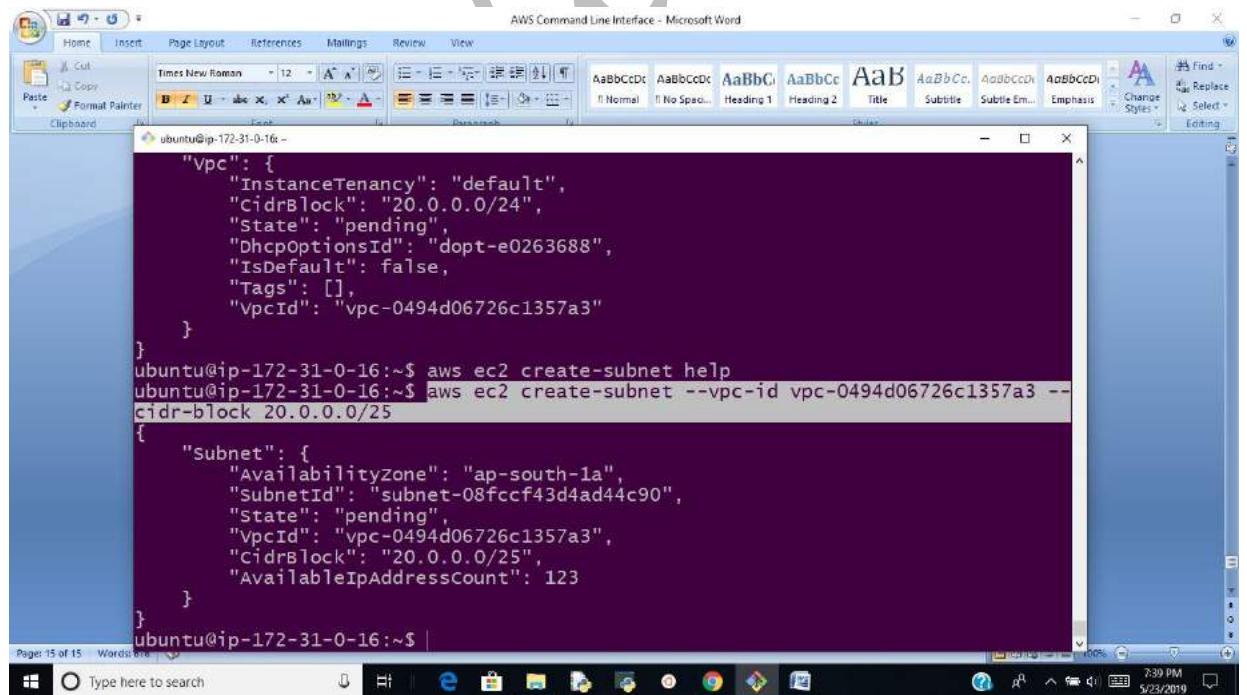
```
ubuntu@ip-172-31-0-16:~$ unassign-private-ip-addresses | unmonitor-instances
wait | help

Invalid choice: 'attach-security-group', maybe you meant:
* create-security-group
ubuntu@ip-172-31-0-16:~$ aws ec2 create-vpc help
ubuntu@ip-172-31-0-16:~$ aws ec2 create-vpc --cidr-block 20.0.0.0/24
{
    "Vpc": {
        "InstanceTenancy": "default",
        "CidrBlock": "20.0.0.0/24",
        "State": "pending",
        "DhcpOptionsId": "dopt-e0263688",
        "IsDefault": false,
        "Tags": [],
        "VpcId": "vpc-0494d06726c1357a3"
    }
}
ubuntu@ip-172-31-0-16:~$ aws ec2 create-vpc --cidr-block 20.0.0.0/24
```

Type here to search

- To create subnet

aws ec2 create-subnet –vpc-id vpcid –cidr-block subnetcidrblock



AWS Command Line Interface - Microsoft Word

```
ubuntu@ip-172-31-0-16:~$ "vpc": {
    "InstanceTenancy": "default",
    "CidrBlock": "20.0.0.0/24",
    "State": "pending",
    "DhcpOptionsId": "dopt-e0263688",
    "IsDefault": false,
    "Tags": [],
    "VpcId": "vpc-0494d06726c1357a3"
}
ubuntu@ip-172-31-0-16:~$ aws ec2 create-subnet help
ubuntu@ip-172-31-0-16:~$ aws ec2 create-subnet --vpc-id vpc-0494d06726c1357a3 --cidr-block 20.0.0.0/25
{
    "subnet": {
        "AvailabilityZone": "ap-south-1a",
        "SubnetId": "subnet-08fccf43d4ad44c90",
        "State": "pending",
        "VpcId": "vpc-0494d06726c1357a3",
        "CidrBlock": "20.0.0.0/25",
        "AvailableIpAddressCount": 123
    }
}
ubuntu@ip-172-31-0-16:~$
```

Type here to search

Thanks and All the best!!!

From

RG(DyaSaa)

DyaSaa