

Amazon Web Services(AWS) SourceFuse-Training Material

AWS-CONTENTS

S.No	Topic Name	Slide No.
1	AWS-Introduction	4
2	Elastic Compute Cloud (EC2)	10
3	Identity and Access Management (IAM)	38
4	Virtual Private Cloud (VPC)	62
5	Load Balancing (LB)	98
6	Auto Scaling Group (ASG)	124
7	Elastic File System (EFS)	137
8	Simple Storage Service (S3)	148
9	Cloud Trail	169
10	Cloud Watch	174
11	Simple Email Service (SES)	183
12	Simple Notification Service (SNS)	191

AWS-CONTENTS

S.No	Topic Name	Slide No.
13	Simple Queuing Service (SQS)	200
14	Lambda	211
15	CloudFront	218
16	Elastic Beanstalk	230
17	Relational Database Service(RDS)	237
18	DynamoDB	249
19	Route53	259
20	Cloud Formation	272
21	AWS-CLI	286
22	ElastiCache	302

AWS Introduction

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)
- Elastic Beanstalk
- Lambda
- Lightsail
- Batch
- AWS outposts
- EC2 Image Builder
- Server less Application repository

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

- S3 (Simple Storage Service)
- Elastic Block Store
- S3 Glacier
- EFS
- Storage Gateway
- AWS backup

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

- Amazon RDS
- Dynamo DB
- Elastic Cache
- Amazon DocumentDB
- Neptune

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
- CDN(Content Delivery Network)
- API gateway
- Direct Connect

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 and provide security to your AWS resources. It consists of the following services:

- AWS IAM
- WAF and Shield
- GuardDuty
- Inspector
- Amazon Macie
- Cognito
- Single Signon
- CloudHSM

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

- Amazon SQS(Simple Queuing Service)
- Amazon SNS (Simple Notification Service)
- Amazon SES (Simple Email Service)

The **Container** domain consists of services which are used for Creating Private registry. It consists of the following domains:

- Elastic Container Registry(ECR)
- Elastic Container Service(ECS)
- Elastic Kubernetes Service (EKS)

The **Analytics** domain consists of services which are used for performing analytical operations. It consists of the following domains:

- Athena
- EMR (Elastic Map Reduce)
- AWS Glue
- Kinesis
- Data Pipeline

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:

- 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

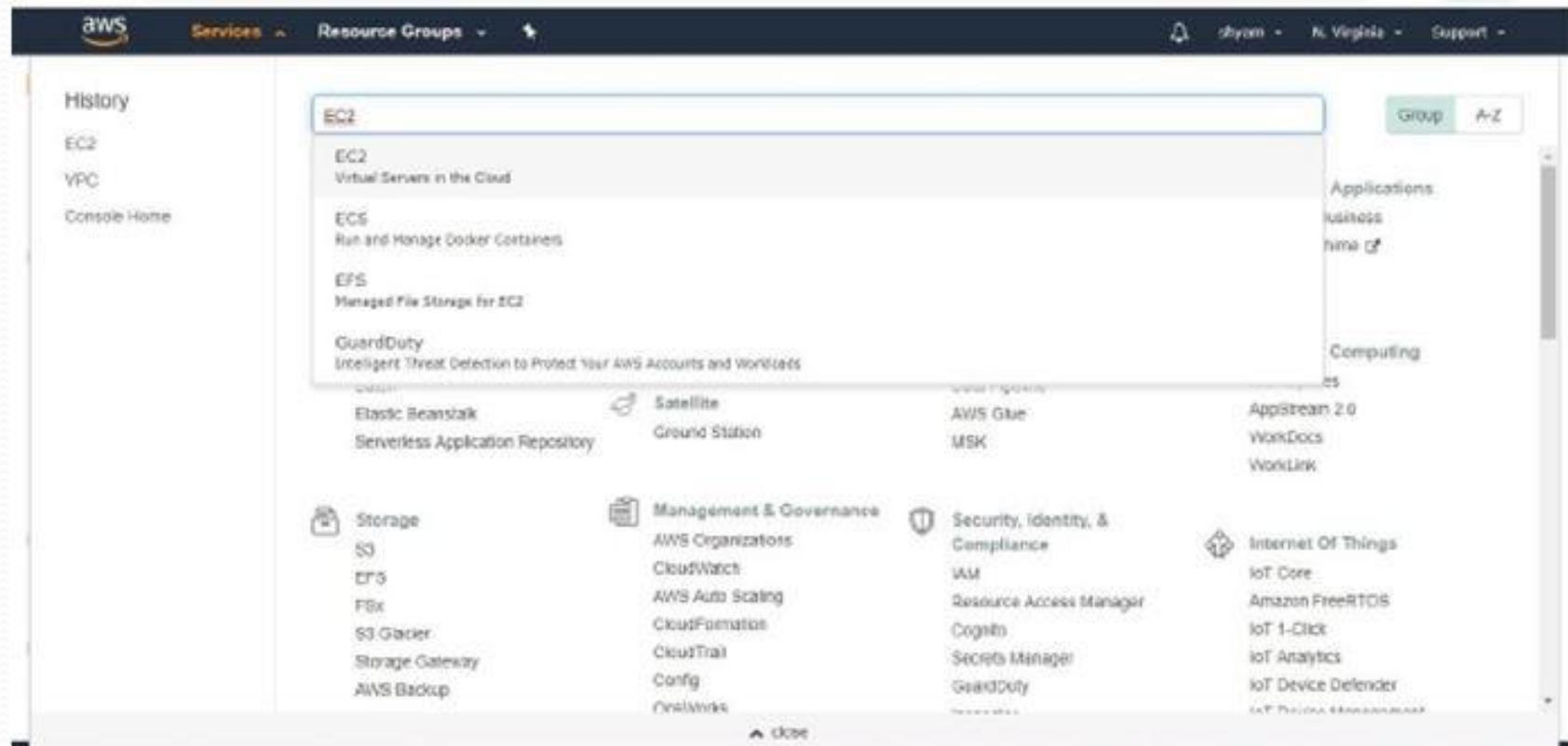
If you previously signed in to the AWS Management Console using AWS account root user credentials, choose Sign in to a different account. If you previously signed in to the console using IAM credentials, choose Sign-in using root account credentials. Then choose Create a new AWS account.

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.

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



Click on Launch Instance

The screenshot shows the AWS EC2 Dashboard. On the left, there's a sidebar with navigation links for EC2 Dashboard, Events, Tags, Reports, Limits, Instances (selected), Launch Templates, Spot Requests, Reserved Instances, Dedicated Hosts, Scheduled Instances, Capacity Reservations, Images (AMIs), Bundle Tasks, Elastic Block Store (Volumes, Snapshots, Lifecycle Manager), and Support.

The main content area has several sections:

- Resources:** You are using the following Amazon EC2 resources in the US East (N. Virginia) region:

0 Running Instances	0 Elastic IPs
0 Dedicated Hosts	1 Snapshots
0 Volumes	0 Load Balancers
4 Key Pairs	16 Security Groups
0 Placement Groups	
- Create Instance:** To start using Amazon EC2 you will want to launch a virtual server, known as an Amazon EC2 instance.
[Launch Instance](#)
- Note:** Your instances will launch in the US East (N. Virginia) region.
- Service Health:** Service Status: US East (N. Virginia) (green dot, indicating healthy). Availability Zone Status: (no status shown).
- Scheduled Events:** US East (N. Virginia): No events.
- Account Attributes:** Supported Platforms: VPC. Default VPC: vpc-97a039ed. Resource ID length management: Console experiments. Settings.
- Additional Information:** Getting Started Guide, Documentation, All EC2 Resources, Points, Pricing, Contact Us.
- AWS Marketplace:** Find free software trial products in the AWS Marketplace from the EC2 Launch Wizard. Or try these popular AMIs.

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

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

Step 1: Choose an Amazon Machine Image (AMI)

Cancel and Exit

Root device type: ebs Virtualization type: hvm ENA Enabled: Yes

Are you launching a database instance? Try Amazon RDS.

Amazon Relational Database Service (RDS) makes it easy to set up, operate, and scale your database on AWS by automating time-consuming database management tasks. With RDS, you can easily deploy Amazon Aurora, MySQL, Oracle, PostgreSQL, and SQL Server databases on AWS. Aurora is a MySQL- and PostgreSQL-compatible, enterprise-class database at 1/10th the cost of commercial databases. Learn more about RDS.

[Launch a database using RDS](#)

 **Ubuntu Server 16.04 LTS (HVM), SSD Volume Type - ami-07b4156579ea1d7ba (64-bit x86) / ami-036ede09522dadcb** [Select](#)

(64-bit Arm)

64-bit (x86)
 64-bit (Arm)

Free tier eligible

 **Microsoft Windows Server 2019 Base - ami-0a5ca0496f746e6e0** [Select](#)

Microsoft Windows 2019 Datacenter edition, [English]

Root device type: ebs Virtualization type: hvm ENA Enabled: Yes

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 EC2 instance creation wizard at Step 2: Choose an Instance Type. The user has selected the t2.micro instance type, which is highlighted with a green border. The table below lists various instance types with their details.

Family	Type	vCPUs	Memory (GB)	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 <small>(Free tier eligible)</small>	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

At the bottom, there are navigation buttons: Cancel, Previous, Review and Launch (which is highlighted in blue), and Next: Configure Instance Details.

Number of Instances: Enter the number for how many instances are launched

- Network: select the VPC to launch that instances
- Subnet: select the subnet to launch that 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

The screenshot shows the AWS EC2 Launch Wizard at Step 4: Add Storage. The top navigation bar includes 'Services', 'Resource Groups', and tabs for '1. Choose AMI', '2. Choose Instance Type', '3. Configure Instance', '4. Add Storage' (which is highlighted), '5. Add Tags', '6. Configure Security Group', and '7. Review'. The main content area is titled 'Step 4: Add Storage' with the sub-instruction: '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.' Below this, a table lists the storage configuration:

Volume Type	Device	Snapshot	Size (GiB)	Volume Type	IOPS	Throughput (Mbps)	Delete on Termination	Encryption
Root	/dev/sda1	snap-0e84451033d576e	8	General Purpose SSD (gp2)	100 / 3000	N/A	No	Not Encrypted

A button 'Add New Volume' is located below the table. A note at the bottom states: 'Free-tier eligible customers can get up to 30 GiB of EBS General Purpose (SSD) or Magnetic storage. Learn more about free usage tier eligibility and usage restrictions.' At the bottom right are buttons for 'Cancel', 'Previous', 'Review and Launch' (which is blue and bolded), and 'Next: Add Tags'.

Add the tags: Enter key and value pair

The screenshot shows the AWS EC2 instance creation process at Step 5: Add Tags. The top navigation bar includes the AWS logo, Services dropdown, Resource Groups dropdown, a user icon, 'ryan' (selected), 'N. Virginia', and 'Support'. Below the navigation is a horizontal progress bar with steps 1 through 7, where step 5 is highlighted. The main content area is titled 'Step 5: Add Tags' and contains instructions: 'A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = Webserver'. It also states: 'A copy of a tag can be applied to volumes, instances or both.' and 'Tags will be applied to all instances and volumes. Learn more about tagging your Amazon EC2 resources.' Below this, there's a table for adding tags. The first row has columns for 'Key' (with placeholder 'instance-name') and 'Value' (with placeholder 'tagish'). To the right are buttons for 'Instances' (0) and 'Volumes' (0). Below this row is a button labeled 'Add another tag (Up to 50 tags maximum)'. The entire interface is set against a light blue background.

Key (127 characters maximum)	Value (256 characters maximum)	Instances (0)	Volumes (0)
instance-name	tagish	0	0

Add another tag (Up to 50 tags maximum)

Next: configure Security group

The screenshot shows the AWS EC2 instance creation wizard at Step 5: Add Tags. The top navigation bar includes 'Services' and 'Resource Groups'. The breadcrumb trail shows the current step: 1. Choose AMI, 2. Choose Instance Type, 3. Configure instance, 4. Add storage, 5. Add Tags, 6. Configure Security Group, 7. Review.

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 = Websvc0.

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 (127 characters maximum)	Value (255 characters maximum)	Instances (i)	Volumes (v)
instance-name	nayesh	1	0

Add another tag (Up to 50 tags maximum)

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 anywhere option
- Click on review and launch

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: MUNCHI-wQ2mf0-14

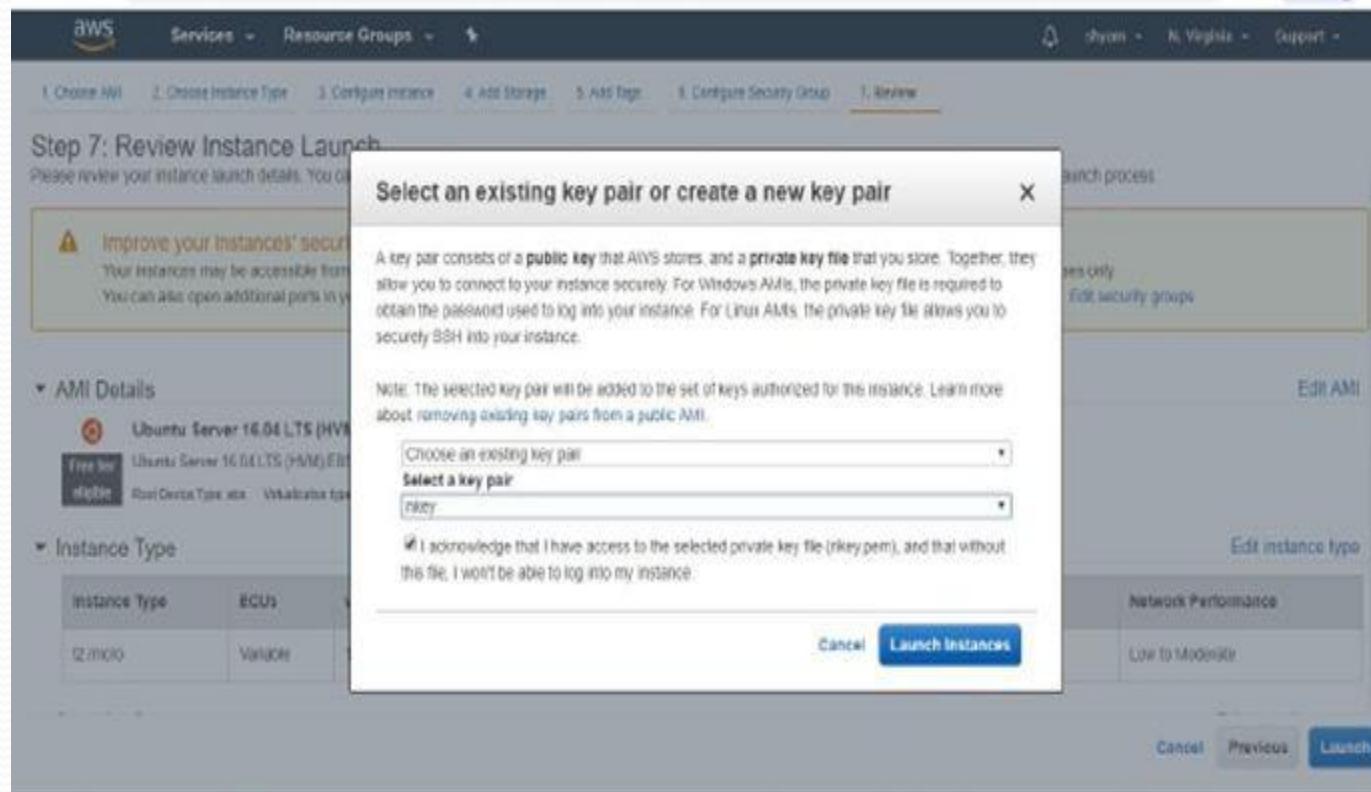
Description: launch-wizard-14 created 2019-06-07T20:09:50Z 577+06:30

Type	Protocol	Port Range	Source	Description
SSH	TCP	22	Custom <input type="button" value="..."/> 0.0.0.0/0	e.g. SSH for Admin Desktop
HTTP	TCP	80	Anywhere <input type="button" value="..."/> 0.0.0.0/0, ::0	e.g. SSH for Admin Desktop

⚠ 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.

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



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

The screenshot shows the AWS EC2 Instances page. On the left, there's a sidebar with navigation links: EC2 Dashboard, Events, Tags, Reports, Limits, INSTANCES (with Instances selected), Launch Templates, Spot Requests, Reserved Instances, Dedicated Hosts, Scheduled Instances, Capacity Reservations, STACES (with Axis selected), and ELASTIC BLOCK STORE (with Volumes selected). The main content area has tabs for Launch Instance, Connect (which is highlighted in blue), and Actions. Below these are filters for Name, Instance ID, Instance Type, Availability Zone, Instance State, Status Checks, Alarm Status, Public DNS (IPv4), and IPv6. A table lists one instance: i-0317da0918a6646cd, t2.micro, us-east-1c, running, 0/2 checks, None, ec2-18-212-14-131.compute-1.amazonaws.com, and 182. At the bottom, there's a detailed view for the selected instance, showing Instance ID: i-0317da0918a6646cd, Public DNS: ec2-18-212-14-131.compute-1.amazonaws.com, Description tab selected, Status Checks, Monitoring, Tags, Instance ID: i-0317da0918a6646cd, Public DNS (IPv4): ec2-18-212-14-131.compute-1.amazonaws.com, Instance state: running, IPv4 Private IP: 18.212.14.131.

Copy Example SSH command and click on close

The screenshot shows the AWS EC2 console interface. On the left, the navigation pane includes links for EC2 Dashboard, Events, Tags, Reports, Units, Instances (selected), Launch Templates, Spot Requests, Reserved Instances, Dedicated Instances, Scheduled Instances, Capacity Reservations, AMIs, Bundle Tasks, and Elastic Block Store Volumes Snapshots. At the bottom, there are Feedback and Language (English (US)) links.

The main content area displays a modal dialog titled "Connect To Your Instance". It contains instructions for connecting:

- I would like to connect with:
 - A standalone SSH client (1)
 - A Java SSH Client directly from my browser (Java required) (1)
- To access your instance:
 - Open an SSH client. (Find out how to connect using PuTTY.)
 - Locate your private key file (`mykey.pem`). The wizard automatically detects the key you used to launch the instance.
 - Your key must not be publicly viewable for SSH-I to work. Use this command if needed:
`chmod 400 mykey.pem`
 - Connect to your instance using its PUBLIC DNS:
`ec2-18-212-14-131.compute-1.amazonaws.com`
- Example:
`ssh -i "mykey.pem" ubuntu@ec2-18-212-14-131.compute-1.amazonaws.com`

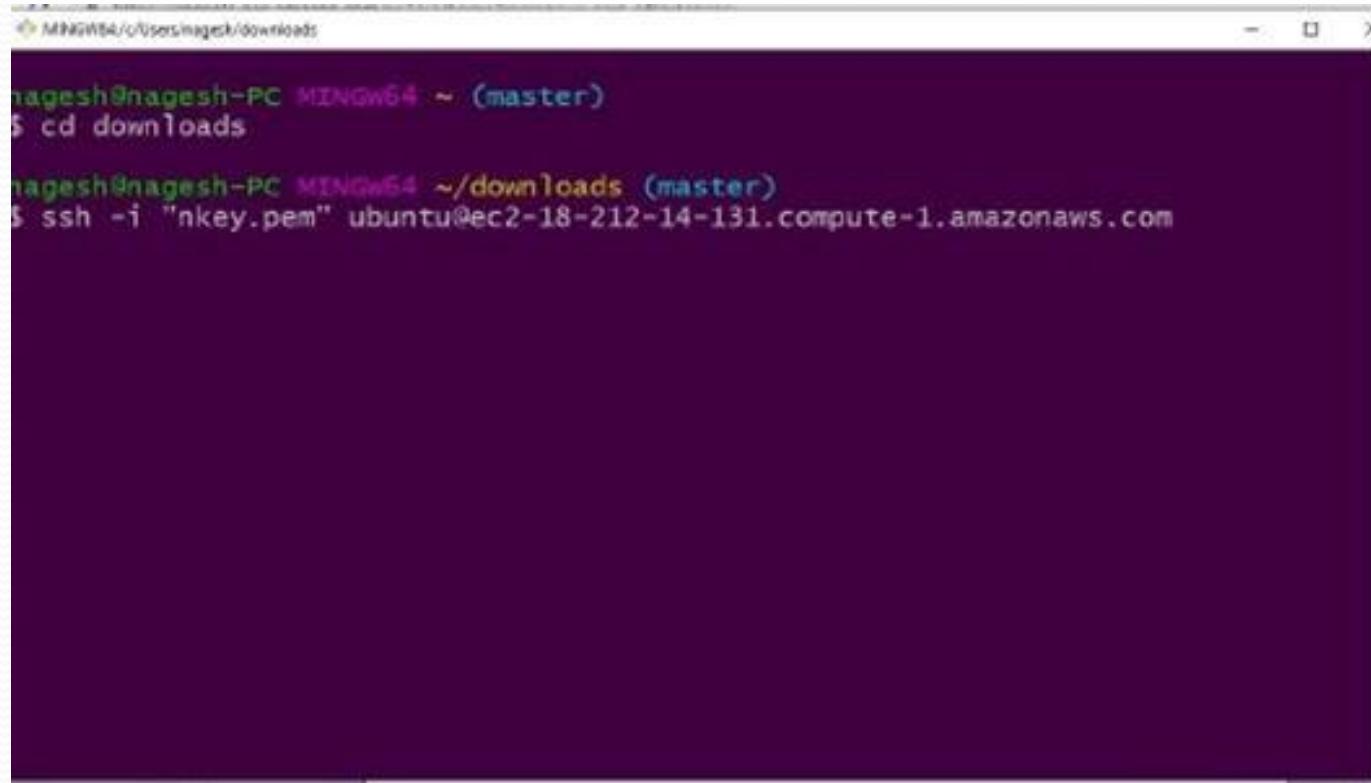
Please note that in most cases the username above will be correct, however please ensure that you read your AMI usage instructions to ensure that the AMI owner has not changed the default AMI username.

If you need any assistance connecting to your instance, please see our connection documentation.

A "Close" button is located at the bottom right of the dialog.

On the right side of the screen, the "Instances" section of the EC2 dashboard is visible, showing a single instance named "ec2-18-212-14-131" with a Public DNS of "ec2-18-212-14-131.compute-1.amazonaws.com" and an IP address of "18.212.14.131".

Paste in gitbash and make enter



```
MINGW64 /c/Users/nagesh/downloads
nagesh@nagesh-PC MINGW64 ~ (master)
$ cd downloads

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

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.

Amazon EBS Volume Types

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

Amazon EBS Volume Types

Solid State Drives (SSD) — Optimized for transactional workloads involving frequent read/write operations with small I/O size, where the dominant performance attribute is IOPS.

Hard Disk Drives (HDD) — Optimized for large streaming workloads where the dominant performance attribute is throughput.

Previous generation — Hard disk drives that can be used for workloads with small datasets where data is accessed infrequently and performance is not of primary importance.

Solid State Drives (SSD)

The SSD-backed volumes provided by Amazon EBS fall into these categories:

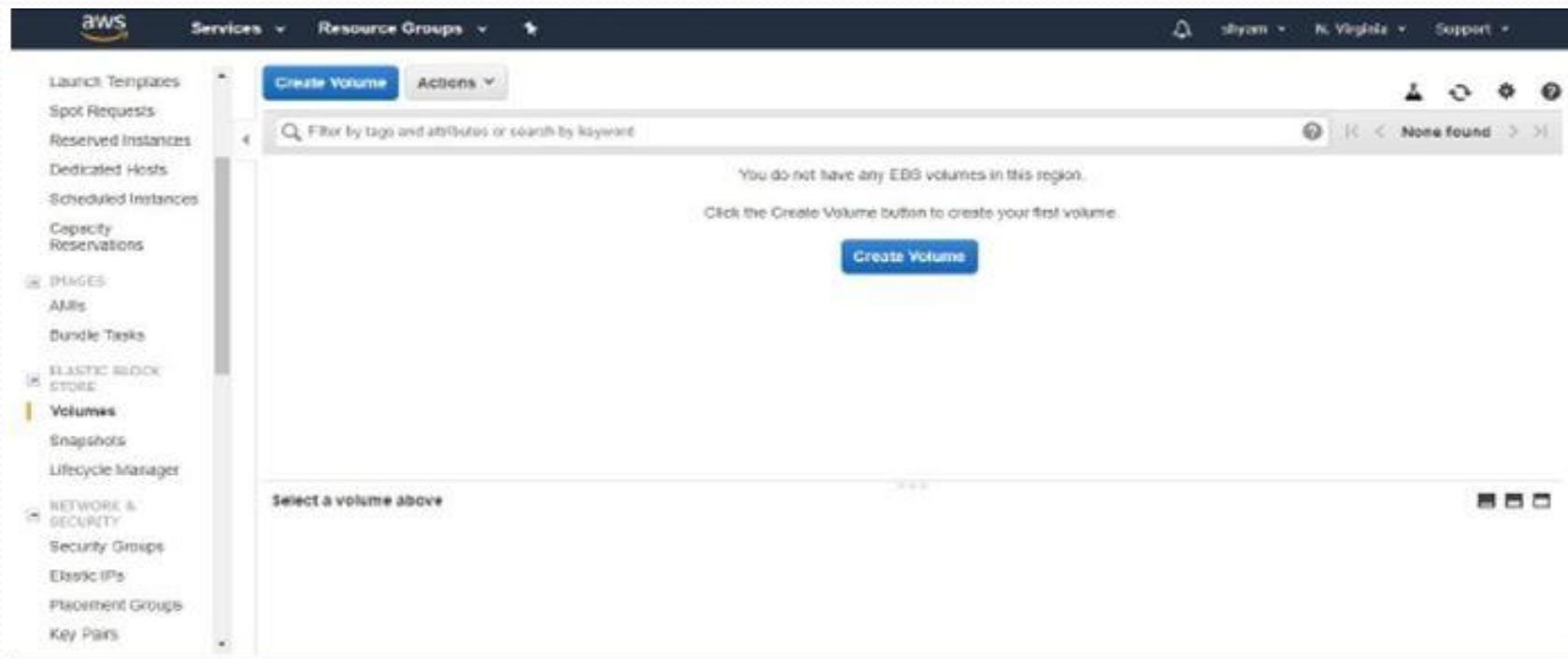
General Purpose SSD — Provides a balance of price and performance. We recommend these volumes for most workloads.

Provisioned IOPS SSD — Provides high performance for mission-critical, low-latency, or high-throughput workloads.

	Throughput Optimized HDD	Cold HDD
Volume type	st1	sc1
Durability	99.8% - 99.9% durability (0.1% - 0.2% annual failure rate)	99.8% - 99.9% durability (0.1% - 0.2% annual failure rate)
Use cases		Throughput-oriented storage for data that is infrequently accessed Scenarios where the lowest storage cost is important
Volume size	125 GiB - 16 TiB	125 GiB - 16 TiB
Max IOPS per volume (1 MiB I/O)	500	250
Max throughput per volume	500 MiB/s	250 MiB/s
Amazon EBS Multi-attach	Not supported	Not supported
Boot volume	Not supported	Not supported

Create volume

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



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

Volumes > Create Volume

Create Volume

Volume Type: General Purpose SSD (gp2) (i)

Size (GB): 100 (Min: 1 GB, Max: 16384 GB) (i)

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

Availability Zone: us-east-1a (i)

Throughput (MB/s): Not applicable (i)

Snapshot ID: Select a snapshot (i)

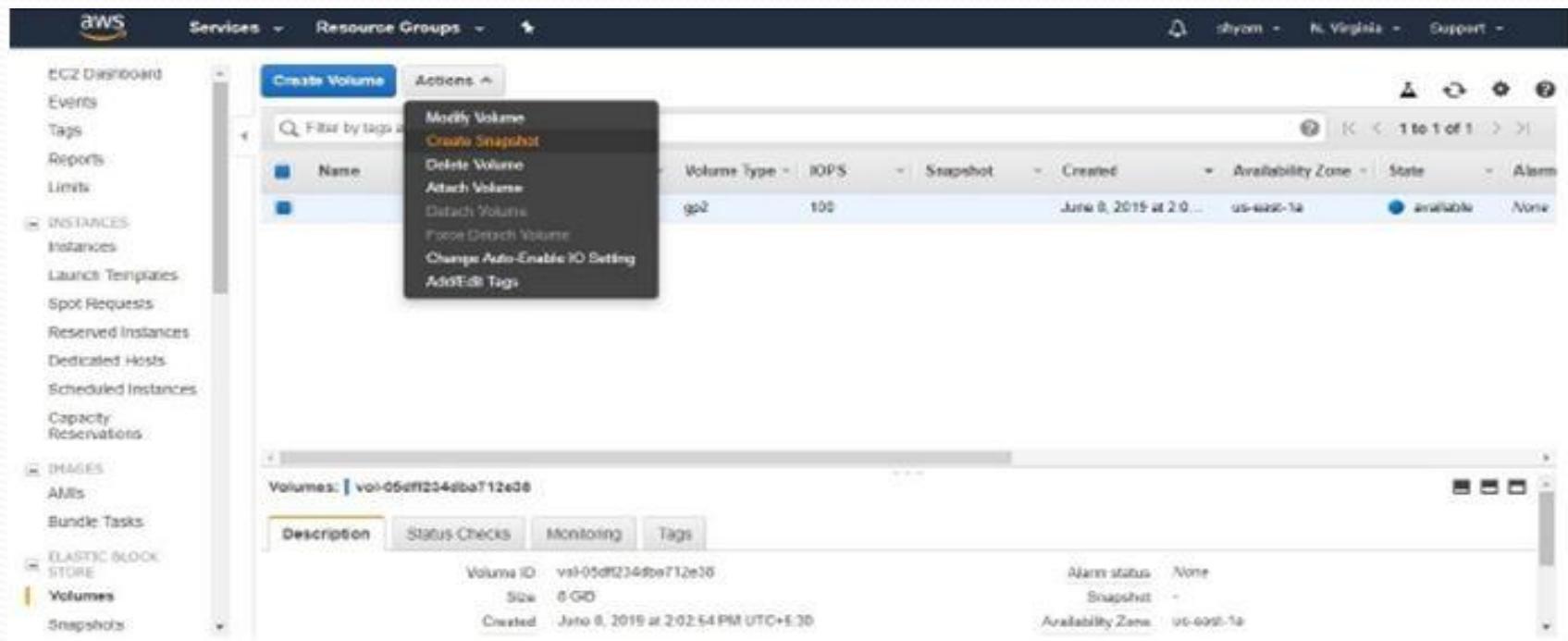
Encryption: Encrypt this volume

Key: (127 characters maximum) (i)

Value: (256 characters maximum) (i)

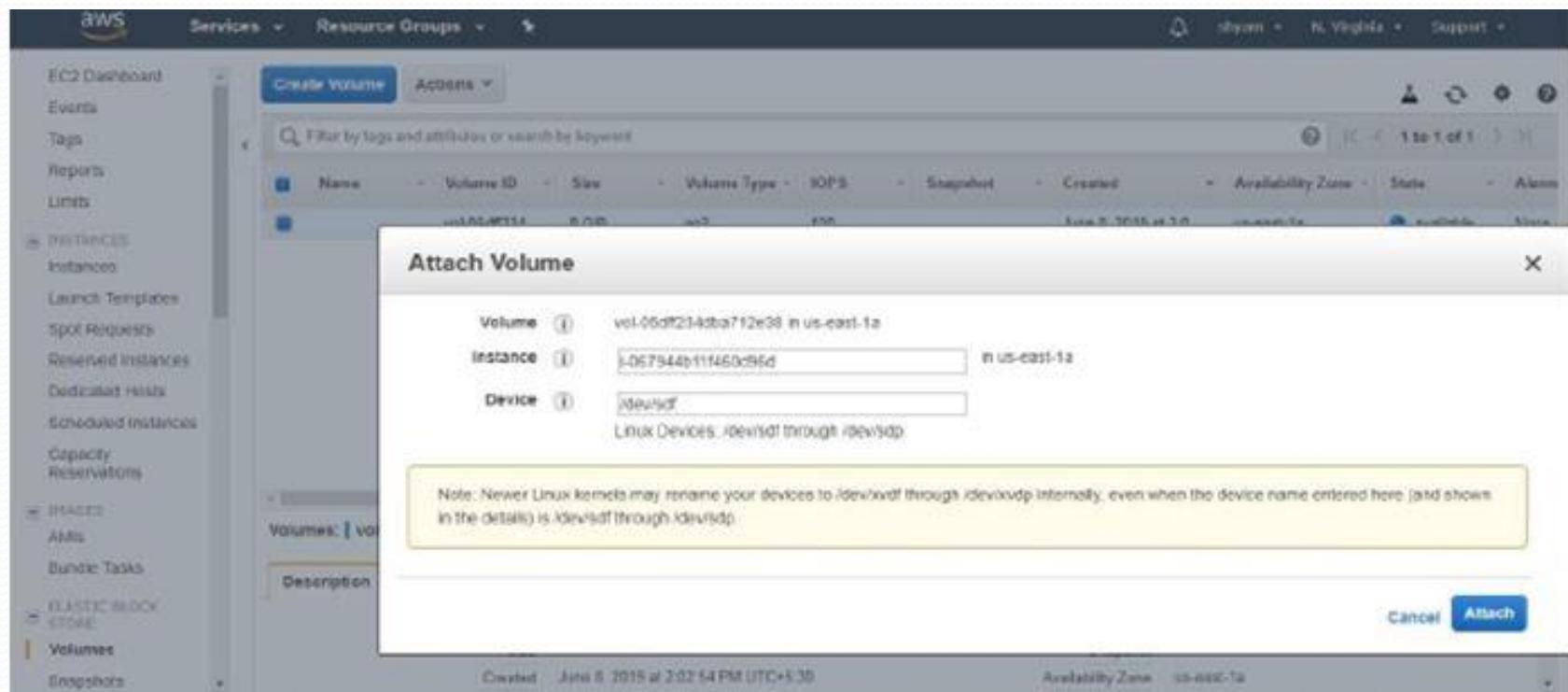
Create snapshot

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



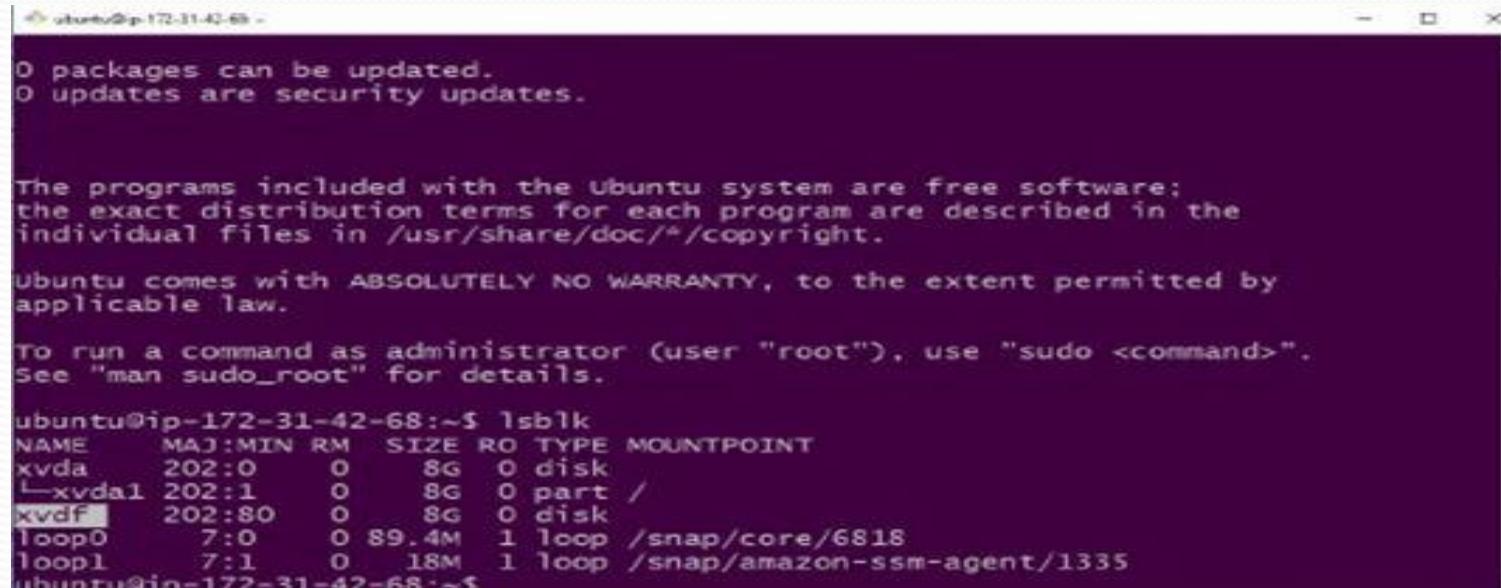
Attach Volume

Select volume and goto actions and click on attach volume option



Mount the volume

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



The screenshot shows a terminal window with the following content:

```
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:~$
```

The terminal window also displays standard Ubuntu system messages at the top:

```
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.
```

Convert the volume into ext4 file system using below command

Sudo mkfs -t ext4 /dev/xvdf

```
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
```

Create one directory using below command

Mkdir ev

```
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.

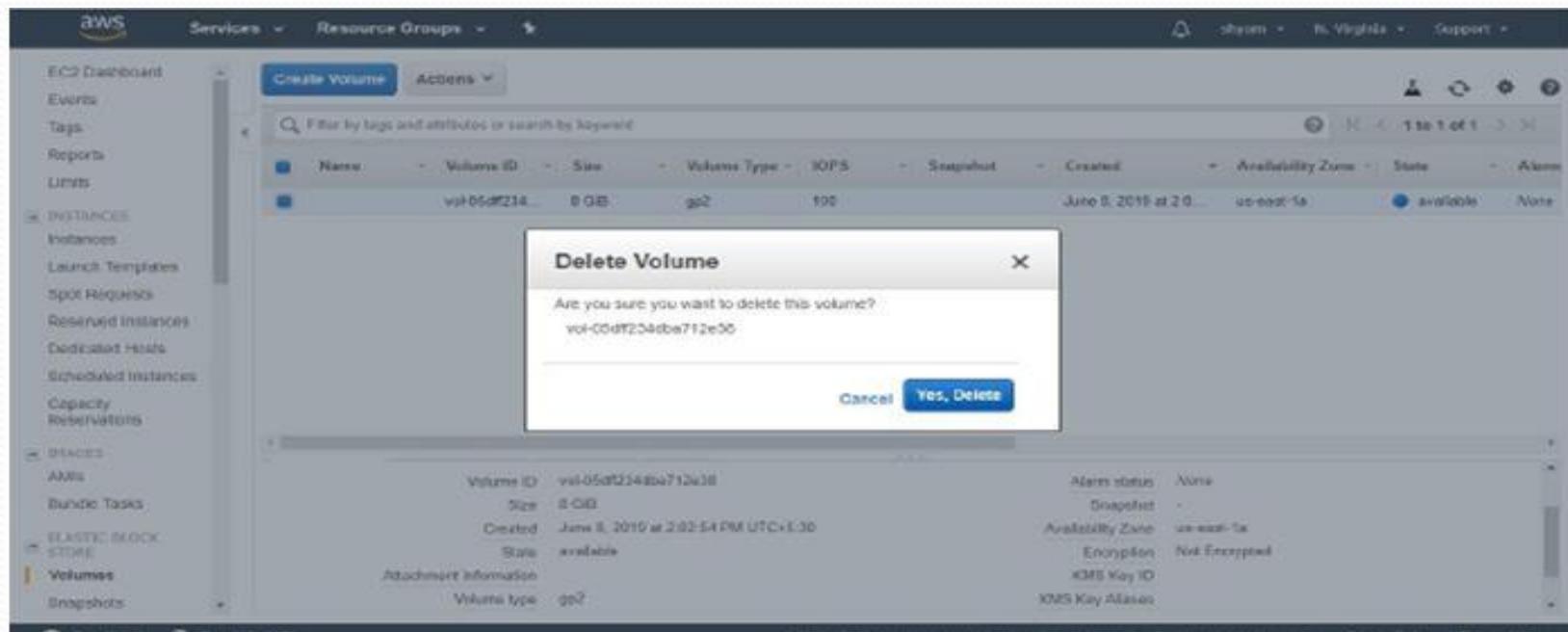
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

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
```

Delete Volume

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



IAM (Identity and Access Management)

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 left side panel and click on Add user

The screenshot shows the AWS IAM service interface. The top navigation bar includes the AWS logo, Services dropdown, Resource Groups dropdown, a user icon for 'shyam', Global dropdown, and Support link. On the far right are three small circular icons. The left sidebar has a 'Search SAM' input field and a list of navigation items: Dashboard, Groups, **Users** (which is selected and highlighted in orange), Roles, Policies, Identity providers, Account settings, and Credential report. Below the sidebar is an 'Encryption keys' section. The main content area has a header with 'Add user' and 'Delete user' buttons. A search bar says 'Find users by username or access key'. A table displays one user account: 'User name' is 'acctUser', 'Groups' is 'None', 'Access key age' is 'Today' (indicated by a green checkmark), 'Password age' is 'Today', 'Last activity' is 'Today', and 'MFA' status is 'Not enabled'. The table has columns for User name, Groups, Access key age, Password age, Last activity, and MFA.

User name	Groups	Access key age	Password age	Last activity	MFA
acctUser	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 key

Another one is AWS Management Console access it is used for setting user

custom password and it's provide user authentication through user console

• Console 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 sign in.

• Click on next permissions

Add user

1 2 3 4 5

Set user details

You can add multiple users at once with the same access type and permissions. [Learn more](#)

User name*

Add another user

Select AWS access type

Select how these users will access AWS. Access keys and autogenerated passwords are provided in the last step. [Learn more](#)

Access type* Programmatic access

Enables an [access key ID](#) and [secret access key](#) for the AWS API, CLI, SDK, and other development tools.

AWS Management Console access

Enables a password that allows users to sign in to the AWS Management Console.

[Programmatic access](#) [Autogenerated password](#)

* Required

[Cancel](#)

[Next: Permissions](#)

Add user

1 2 3 4 5

Set user details

You can add multiple users at once with the same access type and permissions. [Learn more](#)

User name*

testuser

[Add another user](#)

Select AWS access type

Select how these users will access AWS. Access keys and autogenerated passwords are provided in the last step. [Learn more](#)

Access type*

Programmatic access

Enables an **access key ID** and **secret access key** for the AWS API, CLI, SDK, and other development tools.

AWS Management Console access

Enables a **password** that allows users to sign-in to the AWS Management Console.

Console password

Autogenerated password

Custom password

Console password*

Custom password

.....

Show password

Require password reset

User must create a new password at next sign-in

Users automatically get the IAMUserChangePassword policy to allow them to change their own password.

* Required

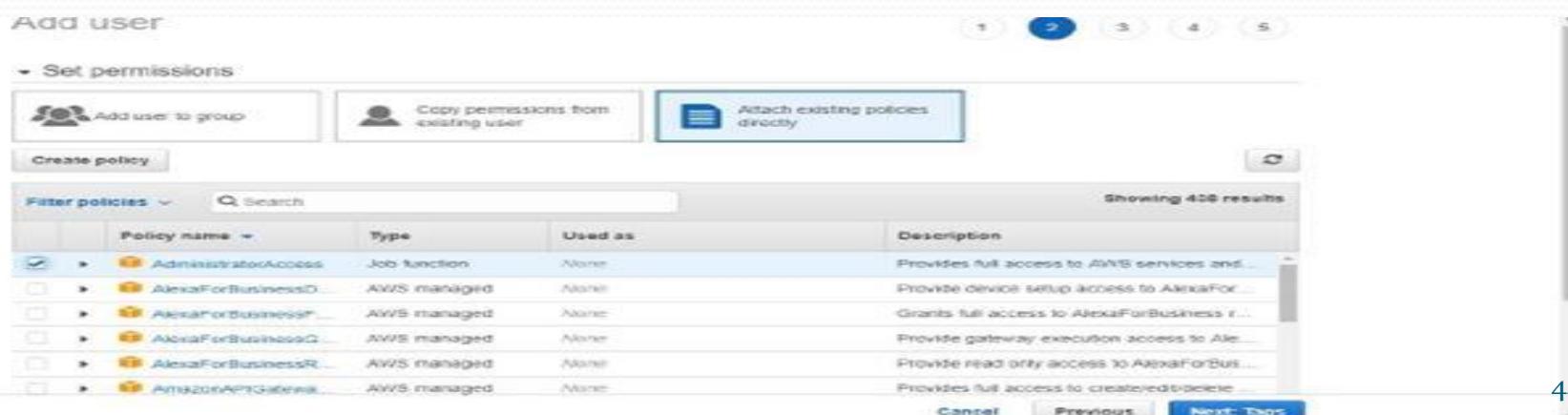
[Cancel](#)

[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

The screenshot shows the 'Add tags (optional)' step of creating a new IAM user. At the top, there's a navigation bar with 'Services', 'Resource Groups', and other account-related links. Below the navigation is a progress bar with five steps, where step 3 is highlighted. The main area contains a table for adding tags:

Key	Value (optional)	Remove
myuser	nagesh	X
Add new key		

A note below the table says, "You can add 49 more tags." At the bottom of the screen are three buttons: 'Cancel', 'Previous', and 'Next: Review'.

Sign in into IAM user

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

The screenshot shows the AWS IAM User Summary page for a user named 'testuser'. The top navigation bar includes 'Services', 'Resource Groups', and 'Global' dropdowns, along with a search bar and user profile icons.

The left sidebar menu is visible, showing options like 'Dashboard', 'Groups', 'Users' (which is selected), 'Roles', 'Policies', 'Identity providers', 'Account settings', and 'Credential report'. Below this is another section for 'Encryption keys'.

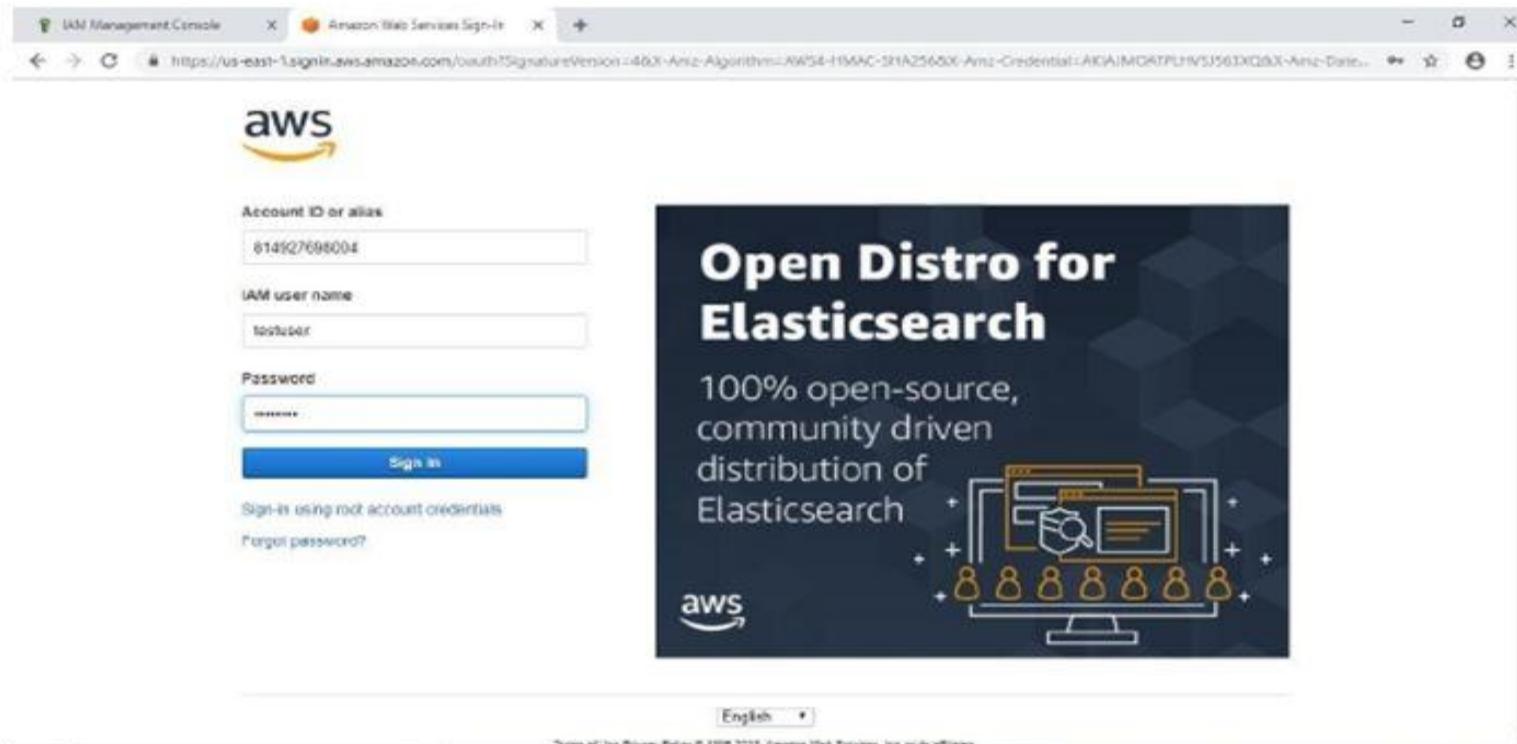
The main content area displays the 'Summary' for the 'testuser'. Key details shown include:

- User ARN: `arn:aws:iam::614927690004:user/testuser`
- Path: `/`
- Creation time: `2019-05-30 21:13 UTC+0000`

The 'Security credentials' tab is active, showing the following information:

- Sign-in credentials:**
 - Summary:** Includes a 'Console sign-in link' (<https://signin.aws.amazon.com/console>)
 - Console password:** Enabled (never signed in) | Manage
 - Assigned MFA device:** Not assigned | Manage
 - Signing certificates:** None
- Access keys:** A note stating: "Use access keys to make secure REST or HTTP Query protocol requests to AWS service APIs. For your protection, you should never share your secret key with anyone. As a best practice, we recommend frequent key rotation." It also links to "Learn more".

Enter IAM username and password and click on login



Check the user permissions in this IAM user account

The screenshot shows the AWS Management Console interface. At the top, there's a navigation bar with the AWS logo, 'Services' dropdown, 'Resource Groups' dropdown, a bell icon, 'testuser @ 8149-2709-80XH', 'N. Virginia', and 'Support'. Below the navigation is the title 'AWS Management Console'.

The main content area is divided into several sections:

- AWS services**: A search bar labeled 'Find Services' with placeholder text 'You can enter names, keywords or acronyms.' and a suggestion 'Example: Relational Database Service, database, RDS'. Below it is a 'Recently visited services' section with 'IAM' listed under the 'Recently visited services' heading.
- Access resources on the go**: A section with a smartphone icon and text: 'Access the Management Console using the AWS Console Mobile App. Learn more'.
- Explore AWS**: A section featuring 'Amazon Redshift' (described as a fast, simple, cost-effective data warehouse) and 'Run Serverless Containers with AWS Fargate' (described as running and scaling containers without managing servers or clusters).
- Build a solution**: A section with the sub-headline 'Get started with simple wizards and automated workflows.' and three buttons: 'Launch a virtual machine', 'Build a web app', and 'Build using virtual servers'.
- Footer**: A horizontal bar with the text 'Scalable. Durable. Secure. Backup & Restore with' followed by a dropdown arrow.

Groups

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

The screenshot shows the AWS IAM Groups page. The URL in the browser is <https://console.aws.amazon.com/iam/home?region=us-east-1#groups>. The left sidebar has a 'Groups' item selected under the 'IAM' section. The main content area has a search bar and a 'Create New Group' button. A table header includes columns for 'Group Name', 'Users', 'Inline Policy', and 'Creation Time'. Below the table, a message says 'No records found.'

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

The screenshot shows the 'Create New Group Wizard' interface for 'Step 1: Group Name'. The top navigation bar includes the AWS logo, 'Services' dropdown, 'Resource Groups' dropdown, user 'shyam', 'Global' dropdown, and 'Support' dropdown. On the left, a sidebar lists 'Create New Group Wizard' steps: 'Step 1: Group Name' (selected), 'Step 2: Attach Policy', and 'Step 3: Review'. The main content area is titled 'Set Group Name' with the sub-instruction 'Specify a group name. Group names can be edited any time.' Below this is a 'Group Name:' input field containing 'mygroup'. A note below the input field says 'Example: Developers or ProdData' and 'Maximum 128 characters'. At the bottom right are 'Cancel' and 'Next Step' buttons.

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

AWS Services Resource Groups Step 2 : Attach Policy Create New Group Wizard Step 1 : Group Name Step 2 : Attach Policy Step 3 : Review

Attach Policy

Select one or more policies to attach. Each group can have up to 10 policies attached.

Policy Name	Attached Entries	Creation Time	Edited Time
AdministratorAccess	1	2015-02-07 00:09 UTC+0530	2015-02-07 00:09 UTC+0530
AlexaForBusinessDeviceSetup	0	2017-11-30 22:17 UTC+0530	2019-06-21 02:56 UTC+0530
AlexaForBusinessFullAccess	0	2017-11-30 22:17 UTC+0530	2019-06-21 03:02 UTC+0530
AlexaForBusinessGatewayExecute	0	2017-11-30 22:17 UTC+0530	2017-11-30 22:17 UTC+0530
AlexaForBusinessReadOnlyAccess	0	2017-11-30 22:17 UTC+0530	2018-06-26 05:22 UTC+0530
AmazonAPIGatewayAdministrator	0	2015-07-09 23:04 UTC+0530	2015-07-09 23:04 UTC+0530
AmazonAPIGatewayInvokeFull	0	2015-07-09 23:06 UTC+0530	2019-12-18 23:55 UTC+0530
AmazonAPIGatewayPushToCloudWatchLogs	0	2015-11-12 00:11 UTC+0530	2019-11-12 00:11 UTC+0530
AmazonAppStreamFullAccess	0	2015-02-07 00:10 UTC+0530	2018-09-10 22:09 UTC+0530

Showing 440 results

Cancel Previous Next Step

Click on create group

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

The screenshot shows the AWS Resource Groups console. On the left, there's a sidebar with navigation links: Dashboard, Groups (which is selected and highlighted in yellow), Users, Roles, Policies, Identity providers, Account settings, and Credential report. Below that is an Encryption keys section. The main content area has a search bar and a 'Create New Group' button. A dropdown menu titled 'Group Actions' is open over a table row for a group named 'mygroup'. The 'Add Users to Group' option is highlighted in orange. Other options in the menu include 'Delete Group', 'Edit Group Name', and 'Remove Users from Group'. The table below shows one result: 'mygroup' with an 'Edit' icon, 'Inline Policy', and a creation time of '2019-05-31 10:17 UTC+0000'. At the bottom of the page, there are links for Feedback, English (US), Copyright notice (© 2006 - 2019, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved.), Privacy Policy, and Terms of Use.

Delete Group: select group and go to Group Actions and click on delete

The screenshot shows the AWS IAM Groups page. In the top navigation bar, 'Services' is set to 'Resource Groups'. The left sidebar includes links for 'Dashboard', 'Groups' (which is selected), 'Users', 'Roles', 'Policies', 'Identity providers', 'Account settings', and 'Credential report'. Below the sidebar is an 'Encryption keys' section. The main content area has a 'Create New Group' button and a 'Group Actions' dropdown menu. The 'Group Actions' menu is open, showing options: 'Add Users to Group', 'Delete Group' (which is highlighted with a yellow box), 'Edit Group Name', and 'Remove Users from Group'. A search bar above the table says 'Search...'. The table displays one result: 'Showing 1 results'. The row contains a checkbox checked, a 'Group Name' column with 'mygroup', an 'Edit Group Name' link, a 'Users' column, an 'Inline Policy' column, and a 'Creation Time' column with '2019-05-31 10:17 UTC+0000'. At the bottom of the page are links for 'Feedback', 'English (US)', and legal notices: '© 2006 - 2019, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved.', 'Privacy Policy', and 'Terms of Use'.

Policies:

- There are two types of policies one is Identity-based policies and another one is resource-based policies
- **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 attribute\
- 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

Create a policies for administration access to IAM users

- Click on create policy

The screenshot shows the AWS IAM Policies page. The left sidebar has a 'Policies' section selected, which is highlighted with a yellow bar. The main content area displays a table of managed policies. The columns are 'Policy name', 'Type', 'Used as', and 'Description'. There are 514 results shown. The first few policies listed are:

Policy name	Type	Used as	Description
AdministratorAccess	Job function	Permissions policy (1)	Provides full access to AWS services and resources.
AlexaForBusinessDeviceSe...	AIWS managed	None	Provide device setup access to AlexaForBusiness services.
AlexaForBusinessPutAccess	AIWS managed	None	Grants full access to AlexaForBusiness resources and access to T...
AlexaForBusinessGateway...	AIWS managed	None	Provide gateway execution access to AlexaForBusiness services.
AlexaForBusinessNetworkP...	AIWS managed	None	This policy enables Alexa for Business to perform automated tasks...
AlexaForBusinessReadOnly...	AIWS managed	None	Provide read only access to AlexaForBusiness services.
AmazonAPIGatewayAdmin...	AIWS managed	None	Provides full access to create/delete APIs in Amazon API Gat...
AmazonAPIGatewayInvoke...	AIWS managed	None	Provides full access to Invoke APIs in Amazon API Gateway.
AmazonAPIGatewayPushT...	AIWS managed	None	Allows API Gateway to push logs to user's account.
AmazonAppStreamFullAcces...	AIWS managed	None	Provides full access to Amazon AppStream via the AWS Manag...
AmazonAppStreamReadOnl...	AIWS managed	None	Provides read only access to Amazon AppStream via the AWS Ma...
AmazonAppStreamService...	AIWS managed	None	Default policy for Amazon AppStream service role.

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": "*"  
}]}
```

We can also attach and detach a policy by selecting policy and go to policy actions and select attach or detach

The screenshot shows the AWS IAM Policies page. On the left, there's a navigation sidebar with links like Dashboard, Groups, Users, Roles, Policies (which is selected), Identity providers, Account settings, and Credential report. Below that is an Encryption keys section. The main content area has tabs for Create policy and Policy actions, with Policy actions being the active tab. A context menu is open over a policy named 'AdministratorAccess', showing options to Attach and Detach. The table lists 515 results, showing columns for Policy, Type, Used as, and Description. Policies listed include AdministratorAccess, AlexaForBusinessDeviceAdmin, AlexaForBusinessFullAccess, AlexaForBusinessGateway, AlexaForBusinessNetworkP, AlexaForBusinessReadOnly, AlexaForBusinessReadonly, AmazonAPIGatewayAdmin, AmazonAPIGatewayInvoke, AmazonAPIGatewayPushTo, AmazonAppStreamFullAccess, AmazonAppStreamReadOnly, and AmazonAppStreamService.

Policy	Type	Used as	Description
AdministratorAccess	Job function	Permissions policy (1)	Provides full access to AWS services and resources.
AlexaForBusinessDeviceAdmin	AWS managed	None	Provide device setup access to AlexaForBusiness services.
AlexaForBusinessFullAccess	AWS managed	None	Grants full access to AlexaForBusiness resources and access to its...
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	Provides read only access to AlexaForBusiness services.
AlexaForBusinessReadonly	AWS managed	None	Provides read only access to AlexaForBusiness services.
AmazonAPIGatewayAdmin	AWS managed	None	Provides full access to create/edit/delete APIs in Amazon API G...
AmazonAPIGatewayInvoke	AWS managed	None	Provides full access to invoke APIs in Amazon API Gateway.
AmazonAPIGatewayPushTo	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...
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.

Delete policy

Select policy and goto policy actions and click on delete

The screenshot shows the AWS IAM Policies page. On the left, there's a sidebar with navigation links: Dashboard, Groups, Users, Roles, Policies (which is selected), Identity providers, Account settings, Credential report, and Encryption keys. The main area has tabs for 'Create policy' and 'Policy actions'. A context menu is open over a row in the table, with 'Detach' being the highlighted option. The table lists 616 results, showing columns for Policy, Type, Used as, and Description. Some policies listed include 'AdministratorAccess', 'AlexaForBusinessDeviceSetupAccess', 'AlexaForBusinessFullAccess', 'AlexaForBusinessGatewayExecutionAccess', 'AlexaForBusinessNetworkAccess', 'AlexaForBusinessReadOnlyAccess', 'AmazonAPIGatewayAdministrator', 'AmazonAPIGatewayInvoke', 'AmazonAPIGatewayPushToCloudWatchLogs', 'AmazonAppStreamFullAccess', 'AmazonAppStreamReadOnlyAccess', and 'AmazonAppStreamServiceRole'. The 'Description' column provides a brief summary of each policy's purpose.

Policy	Type	Used as	Description
AdministratorAccess	Job function	Permissions policy (1)	Provides full access to AWS services and resources.
AlexaForBusinessDeviceSetupAccess	AWS managed	None	Provide device setup access to AlexaForBusiness services
AlexaForBusinessFullAccess	AWS managed	None	Grants full access to AlexaForBusiness resources and access to r...
AlexaForBusinessGatewayExecutionAccess	AWS managed	None	Provide gateway execution access to AlexaForBusiness services
AlexaForBusinessNetworkAccess	AWS managed	None	This policy enables Alexa for Business to perform automated tasks...
AlexaForBusinessReadOnlyAccess	AWS managed	None	Provide read only access to AlexaForBusiness services
AmazonAPIGatewayAdministrator	AWS managed	None	Provides full access to create/delete APIs in Amazon API G...
AmazonAPIGatewayInvoke	AWS managed	None	Provides full access to invoke APIs in Amazon API Gateway
AmazonAPIGatewayPushToCloudWatchLogs	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...
AmazonAppStreamReadOnlyAccess	AWS managed	None	Provides read only access to Amazon AppStream via the AWS Ma...
AmazonAppStreamServiceRole	AWS managed	None	Default policy for Amazon AppStream service role.

Roles

IAM roles are secure way to grant permissions to entities that you trust. Examples of entities include the following.

- IAM users in another account
- Application code running on EC2 instance that needs to perform actions on AWS Resource.

The screenshot shows the AWS IAM Roles page. On the left, there's a sidebar with links like Dashboard, Groups, Users, Roles (which is selected), Policies, Identity providers, Account settings, Credential report, and Encryption keys. The main content area has a title 'Roles' and a sub-section titled 'What are IAM roles?'. It explains that IAM roles are a secure way to grant permissions to entities and lists examples: 'IAM user in another account', 'Application code running on an EC2 instance that needs to perform actions on AWS resources', 'An AWS service that needs to act on resources in your account to provide its features', and 'Users from a corporate directory who use identity federation with SAML'. It also notes that IAM roles issue short-lived keys. Below this, there's a section for 'Additional resources' with links to 'IAM Roles FAQ', 'IAM Roles Documentation', 'Tutorial: Setting Up Cross-Account Access', and 'Common Scenarios for Roles'. At the bottom, there are buttons for 'Create role' and 'Delete role', and a search bar.

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

The screenshot shows the 'Create role' wizard in the AWS IAM console. The top navigation bar includes 'Services', 'Resource Groups', and 'Support'. Below the title 'Create role' is a step indicator showing '1' (highlighted in blue) and '2 3 4'. The first step, 'Select type of trusted entity', has four options: 'AWS service' (EC2, Lambda and others), 'Another AWS account' (Belongs to you or 3rd party, currently selected), 'Web Identity' (Cognito or any OpenID provider), and 'SAML 2.0 federation' (Your corporate directory). A note below says 'Allows entities in other accounts to perform actions in this account. Learn more'. The second step, 'Specify accounts that can use this role', contains an 'Account ID' input field with '170831086255' and an 'Options' section with checkboxes for 'Require external ID' (unchecked) and 'Require MFA' (unchecked). At the bottom are 'Cancel' and 'Next: Permissions' buttons.

Create role

1 2 3 4

Select type of trusted entity

AWS service EC2, Lambda and others

Another AWS account Belongs 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: 170831086255

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

* Required Cancel Next: Permissions

Tags is the optional if you want mention give key and value pair and click on next review

• **Role name:** enter name to this role

• **Role Description:** write some description to this role and it is the optional

• Click on create role

The screenshot shows the 'Create role' review step in the AWS IAM console. The top navigation bar includes 'Services', 'Resource Groups', a search bar, and links for 'shyam', 'Global', and 'Support'. The main section is titled 'Create role' and 'Review'. It displays the following information:

- Role name:** bharathi (highlighted in a red box)
- Role description:** (empty text area)
- Trusted entities:** The account: 170831066265
- Policies:** AdministrationAccess (highlighted in a red box)
- Permissions boundary:** Permissions boundary is not set.

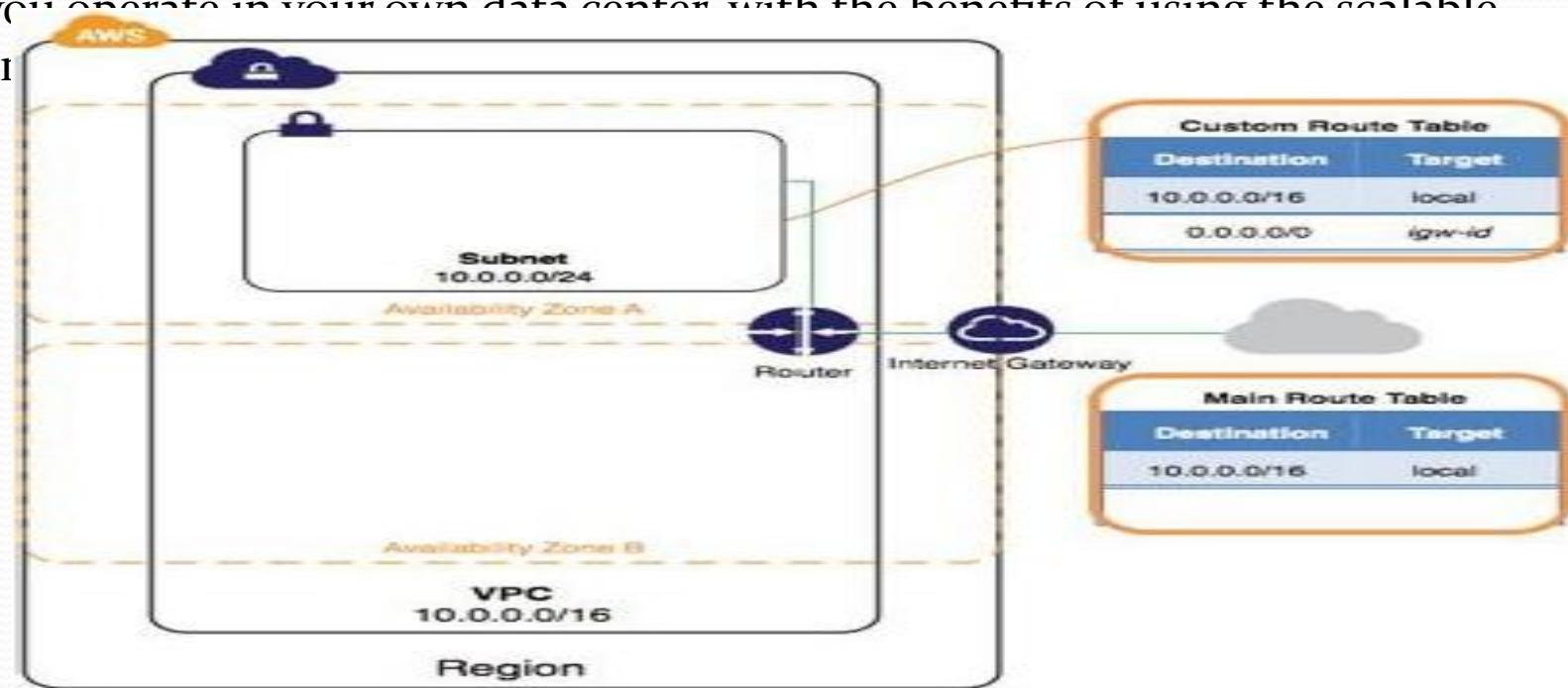
At the bottom, there are buttons for 'Cancel', 'Previous', and a blue 'Create role' button. A note at the bottom left indicates that the 'Required' field is mandatory. The footer contains links for 'Feedback', 'English (US)', '© 2006 - 2019, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved.', 'Privacy Policy', and 'Terms of Use'.

VPC (Virtual Private Cloud)

Amazon VPC:

Amazon Virtual Private Cloud (Amazon VPC) enables you to launch AWS resources into virtual network that you've defined. This virtual network closely resembles traditional

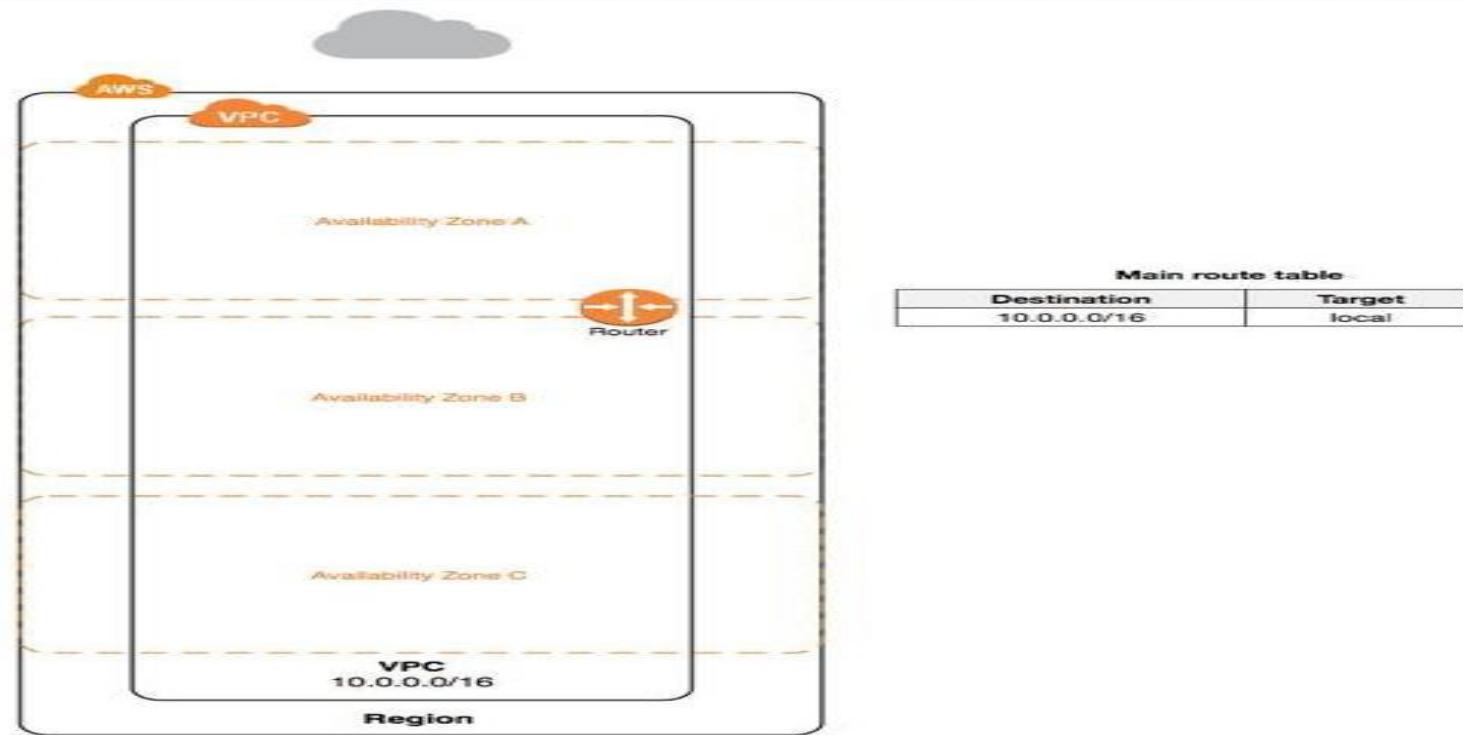
N/W you operate in your own data center with the benefits of using the scalable infrastructure.



VPC and Subnets:

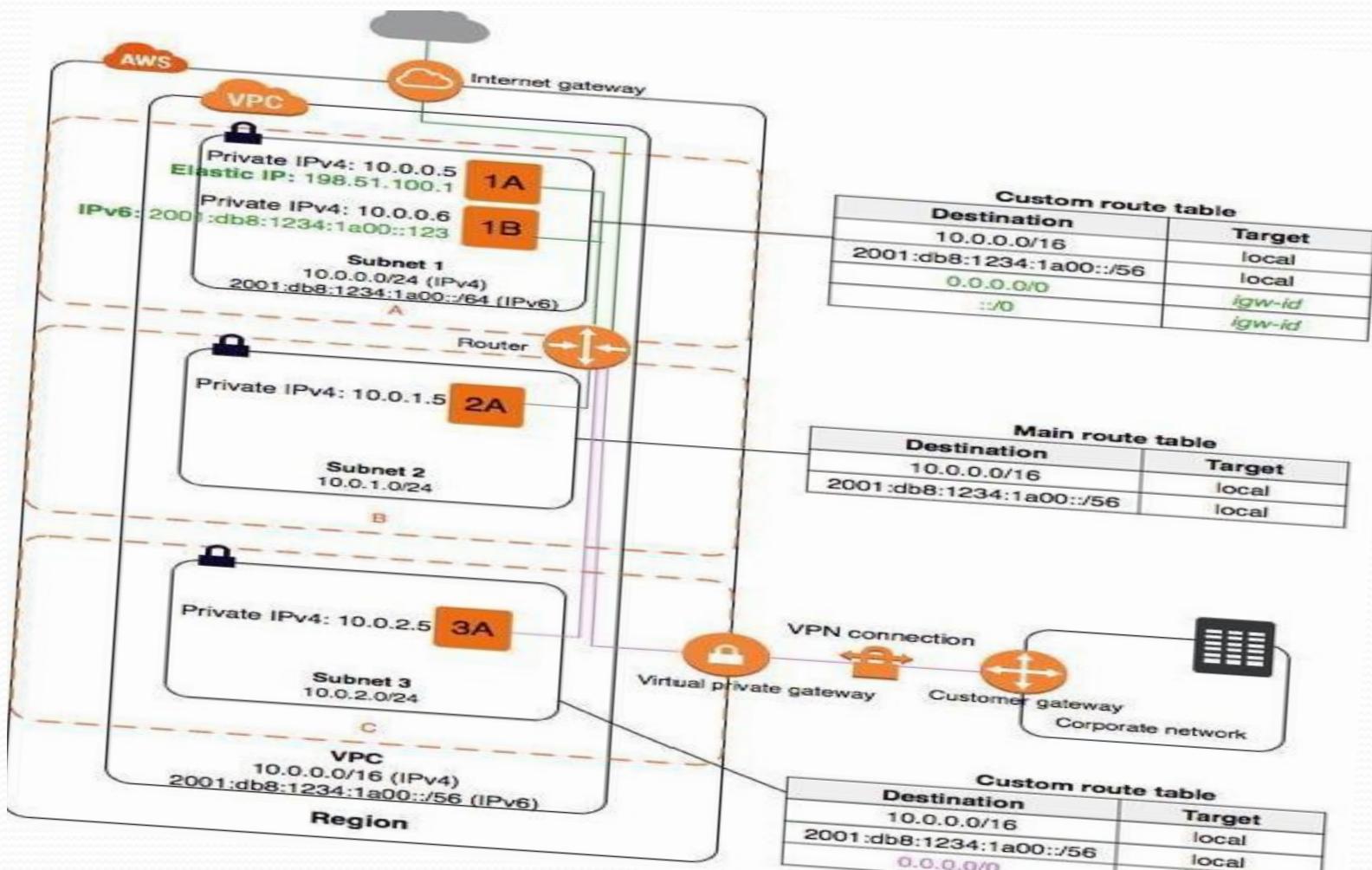
A virtual private cloud (VPC) is a virtual network dedicated to your AWS account. It is logically isolated from other virtual networks in the AWS Cloud. You can launch your AWS resources, such as Amazon EC2 instances, into your VPC.

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.



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

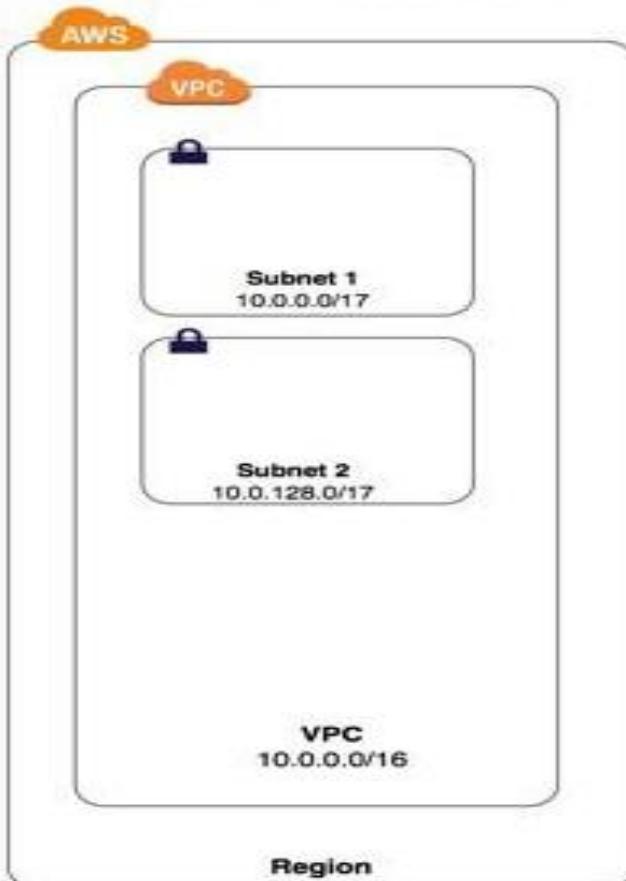
When you create a VPC, you must specify an IPv4 CIDR block for the VPC. Allowed block size is between a /16 netmask (65,536 IP addresses) and /28 netmask (16 IP addresses). After you've created your VPC, you can associate secondary CIDR blocks with the VPC.

Adding IPv4 CIDR Blocks to a VPC

We can associate secondary IPv4 CIDR blocks with your VPC. When you associate 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.

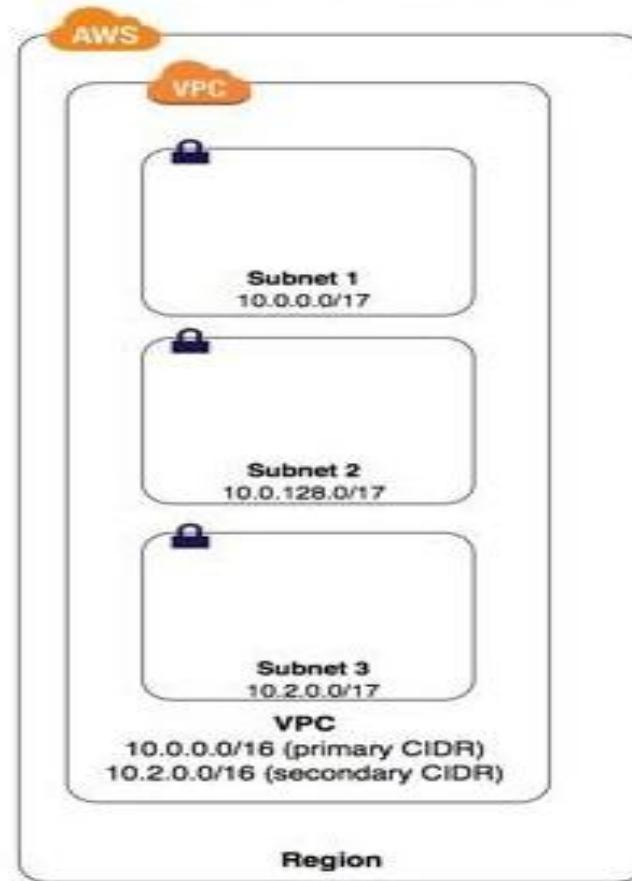
VPC with 1 CIDR block



Main route table

Destination	Target
10.0.0.0/16	local

VPC with 2 CIDR blocks



Main route table

Destination	Target
10.0.0.0/16	local
10.2.0.0/16	local

Create VPC:

Sign in into AWS account and select VPC from Network and Content Delivery section
Click on create VPC

The screenshot shows the AWS VPC Dashboard. On the left, there's a sidebar with navigation links: VPC Dashboard, Filter by VPC (with a dropdown menu), Virtual Private Cloud, Your VPCs (which is selected and highlighted in yellow), Subnets, Route Tables, Internet Gateways, Egress Only Internet Gateways, DHCP Options Sets, Elastic IPs, Endpoints, Endpoint Services, NAT Gateways, Peering Connections, and Security. At the top, there are tabs for Services, Resource Groups, and Actions, along with a search bar and pagination controls (1 to 1 of 1). The main content area displays a table with one row of data. The columns are: Name, VPC ID, State, IPv4 CIDR, IPv6 CIDR, DHCP options set, and Main Route table. The data row shows: Name - 'vpc-97ab3fed', VPC ID - 'vpc-97ab3fed', State - 'available', IPv4 CIDR - '172.31.0.0/16', IPv6 CIDR - 'None', DHCP options set - 'dopt-63112018', and Main Route table - 'rtb-4bd7c437'. Below the table, there's a summary card for the VPC 'vpc-97ab3fed' with tabs for Description, CIDR Blocks, Flow Logs, and Tags. The Description tab is active, showing the VPC ID, State, IPv4 CIDR, and other details like Tenancy (default), Default VPC (Yes), and Classic link (Disabled).

Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR	DHCP options set	Main Route table
vpc-97ab3fed	vpc-97ab3fed	available	172.31.0.0/16	None	dopt-63112018	rtb-4bd7c437

Steps for Creating 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 $4^8=32$ that is 2^{32} ipv4 addresses
- Based on subnet mask the range is allocated that is
- Total range-subnet mask = $32-24=8$ 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 left side panel of your VPC section and click on create subnet

The screenshot shows the AWS VPC Dashboard. On the left sidebar, under the 'Subnets' section, there is a 'Create subnet' button. The main area displays a table of existing subnets:

Name	Subnet ID	State	VPC	IPv4 CIDR	Available IPv4	IPv6 CIDR	Available
subnet-09bcb443	subnet-09bcb443	available	vpc-97ab3fed	172.31.16.0/20	4091	-	us-east-1
subnet-10e9ec4c	subnet-10e9ec4c	available	vpc-97ab3fed	172.31.32.0/20	4091	-	us-east-1
subnet-1cf16970	subnet-1cf16970	available	vpc-97ab3fed	172.31.0.0/20	4091	-	us-east-1
subnet-5d14b383	subnet-5d14b383	available	vpc-97ab3fed	172.31.80.0/20	4091	-	us-east-1
subnet-a5a3fc9d	subnet-a5a3fc9d	available	vpc-97ab3fed	172.31.64.0/20	4091	-	us-east-1
subnet-ea920a05	subnet-ea920a05	available	vpc-97ab3fed	172.31.48.0/20	4091	-	us-east-1

Name tag: Enter name for subnet

VPC: select VPC in which VPC 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

The screenshot shows the AWS Management Console interface for creating a new subnet. The top navigation bar includes 'Services', 'Resource Groups', and 'Support'. On the right, there are links for 'System' (dropdown), 'N. Virginia' (region dropdown), and 'Support'. Below the navigation is a breadcrumb trail: 'Subnets > Create subnet'. The main title is 'Create subnet'. A descriptive note says: 'Specify your subnet's IP address block in CIDR format, for example, 10.0.0.0/24. IPv4 block sizes must be between a /16 netmask and /28 netmask, and can be the same size as your VPC. An IPv6 CIDR block must be a /64 CIDR block.' The form fields are as follows:

Name tag	mysubnet		
VPC*	vpc-024155fb1da5afc48		
VPC CIDRs	CIDR 30.0.0.0/24	Status associated	Status Reason
Availability Zone	us-east-1c		
IPv4 CIDR block*	30.0.0.0/25		

At the bottom left is a note: "Required" with an asterisk. At the bottom right are 'Cancel' and 'Create' buttons.

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

The screenshot shows the AWS VPC Dashboard. On the left sidebar, under the 'Route Tables' section, the 'Route Tables' link is highlighted. The main content area displays a table of existing route tables. The table has columns for Name, Route Table ID, Explicitly Associated with, Main, VPC ID, and Owner. Two route tables are listed:

Name	Route Table ID	Explicitly Associated with	Main	VPC ID	Owner
rtb-0cb19fd0d1855c02c	rtb-0cb19fd0d1855c02c	-	Yes	vpc-024153fb1dollar40	014927606004
rtb-40d7c437	rtb-40d7c437	-	Yes	vpc-07ab0fed	014927606004

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 Lambda console with the title "Edit subnet associations". The URL in the address bar is "Route Tables > Edit subnet associations". The main content area displays a table titled "Route table: ID-047064779EB41A60C (myvourl)". The table has one row, which is highlighted with a blue background. The row contains the following information:

Associated subnets	subnet-0439e593d0ef5ab
Subnet ID	subnet-0439e593d0ef5ab
IPv4 CIDR	0.0.0.0/25
IPv6 CIDR	-
Current Route Table	Main

At the bottom of the table, there is a "Save" button. A note at the bottom left says "* Required".

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

Screenshot of the AWS Network ACLs page:

Left sidebar navigation:

- Endpoints
- Endpoint Services
- NAT Gateways
- Peering Connections
- Security**
- Network ACLs
- Security Groups
- Virtual Private Network (VPN)
- Customer Gateways
- Virtual Private Gateways
- Site-to-Site VPN Connections
- Client VPN Endpoints
- Transit Gateways
- Transit Gateways
- Transit Gateway

Main content area:

Create network ACL Actions

Filter by tags and attributes, or search by keyword

Name	Network ACL ID	Associated with	Default	VPC	Owner
acl-0ec04cb56f82	subnet-0439e693	Yes	vpc-02415fb1da5af048 myvpc	014927690004	
acl-9a2b04e7	6 Subnets	Yes	vpc-97eb0fed	014927690004	

Edit subnet association:

Screenshot of the AWS Network ACLs page showing the edit subnet association dialog:

Left sidebar navigation:

- Filter by VPC: Select a VPC
- 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

Main content area:

Create network ACL Actions

Deletes network ACL

Edit subnet associations

Filter

Name	Associated with	Default	VPC	Owner
acl-0ec04cb56f82	subnet-0439e693	Yes	vpc-02415fb1da5af048 myvpc	014927690004

Network ACL: acl-0ec04cb56f82

Details Inbound Rules Outbound Rules Subnet associations Tags

Network ACL ID	Associated with	Default	VPC
acl-0ec04cb56f82	subnet-0439e693 02415fb1da5af048	Yes	vpc-02415fb1da5af048 myvpc

Select subnet and click on Edit

The screenshot shows the AWS Management Console interface for managing Network ACLs. The top navigation bar includes the AWS logo, Services dropdown, Resource Groups dropdown, a notification bell icon, user name 'shyam', region 'N. Virginia', and Support link.

The current page is 'Network ACLs > Edit subnet associations' for a specific Network ACL with ID 'n0-0e04c000f0263509'. The main content area displays a table of subnet associations:

Subnet ID	IPv4 CIDR	IPv6 CIDR	Associated with
subnet-0439e583df2e8fbab mysubnet	10.0.0.0/25	-	acl-0ec04cb609f263509

Below the table, there is a note: '* Required' and two buttons: 'Cancel' and 'Edit' (highlighted in blue).

Add Inbound Rules

- Select NACL click on inbound rules in below panel of NACL and click on edit inbound rules
- 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 CIDR Block select **ALLOW** or **DENY** and click on save.

Rule #	Type	Protocol	Port Range	Source	Allow / Deny
100	All Traffic	All	All	0.0.0.0	Allow
101	Custom TCP Rule	TCP (5)	8081	0.0.0.0	Allow

Add Rule

* Required

Cancel

Save

Add Outbound Rules

- Select NACL click on inbound rules in below panel of NACL and click on edit outbound rules
- 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 CIDR Block select ALLOW or DENY and click on save.

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

Add Rule

* Required

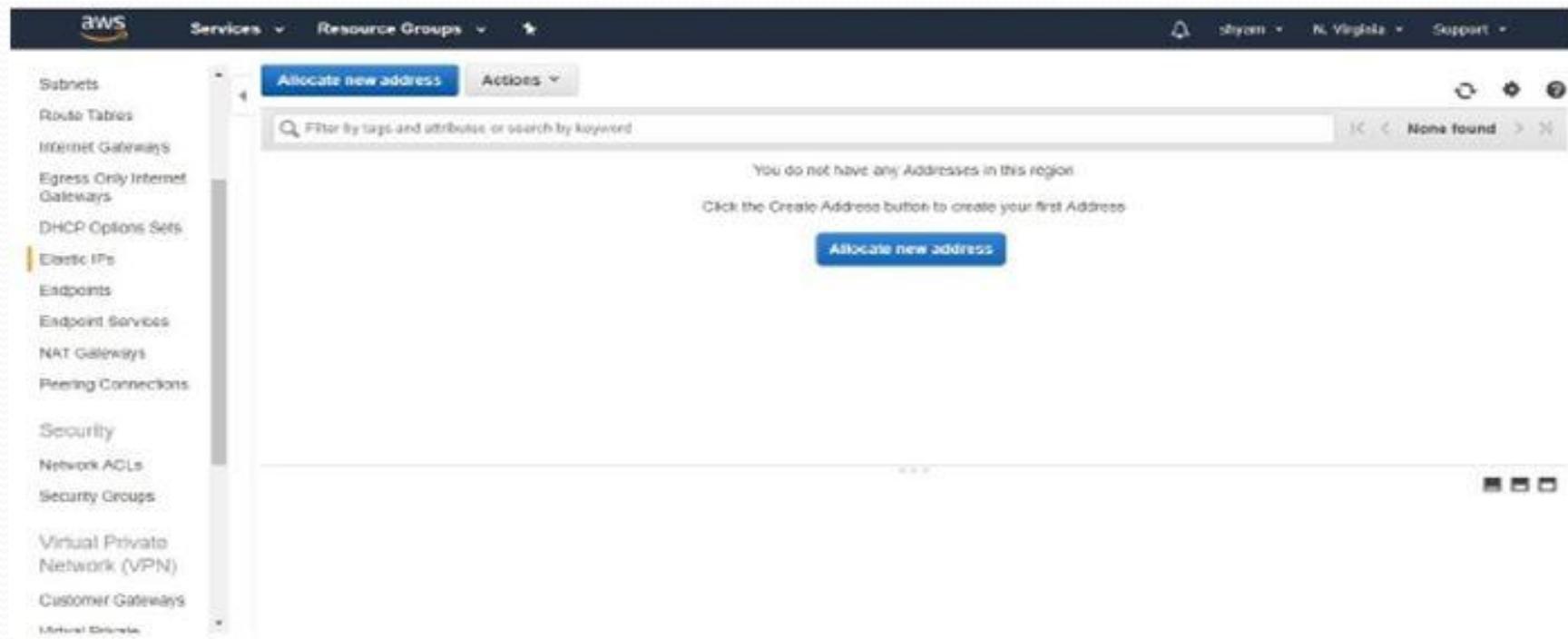
Cancel Save

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.



Associate Elastic IP address:

Select Elastic IP address and go to actions and click on associate address

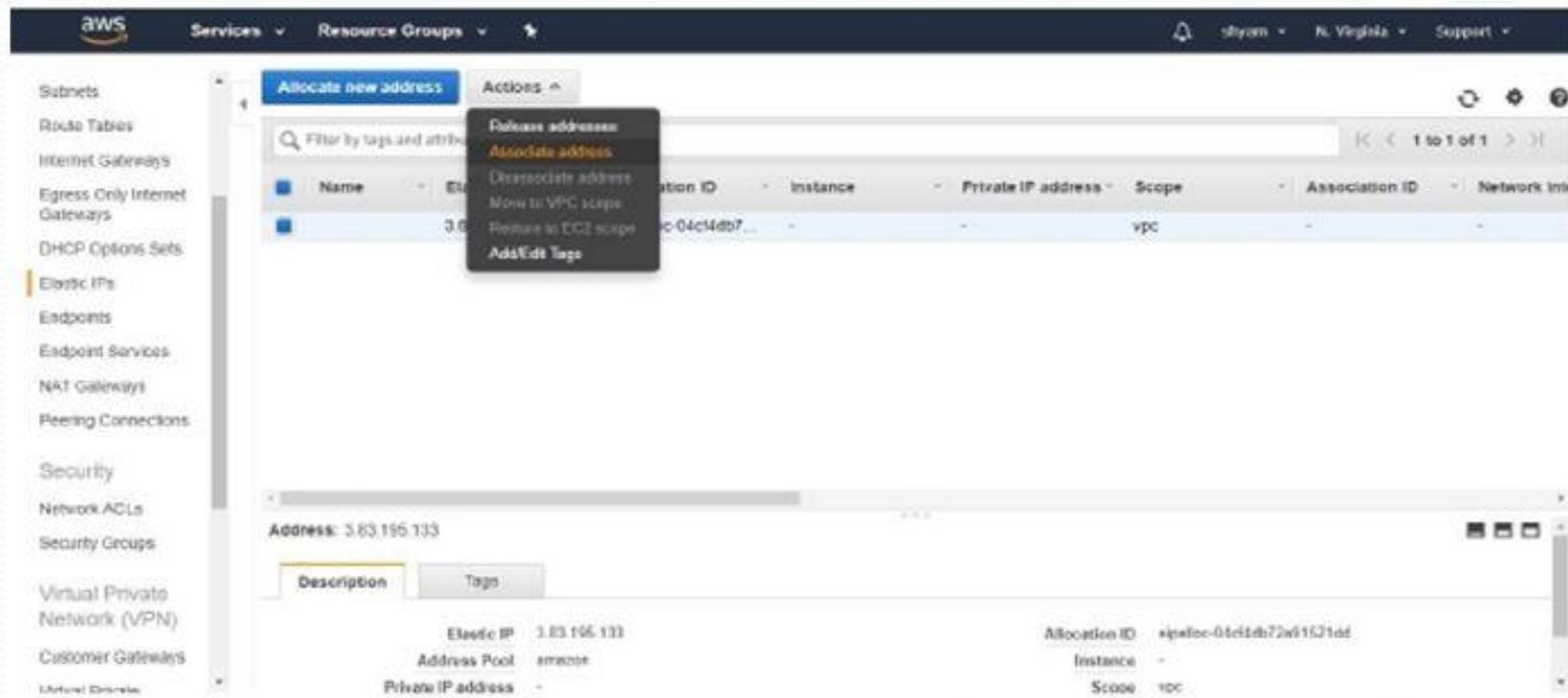
Resource Type: select Instance or network interface to attach this Elastic IP (here select instance)

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

Re association: allow Elastic IP to be re associated if already attached

Click on associate



aws Services Resource Groups

Addresses > Associate address

Associate address

Select the instance or network interface to which you want to associate this Elastic IP address (3.83.196.133)

Resource type Instance [?](#)

Network Interface

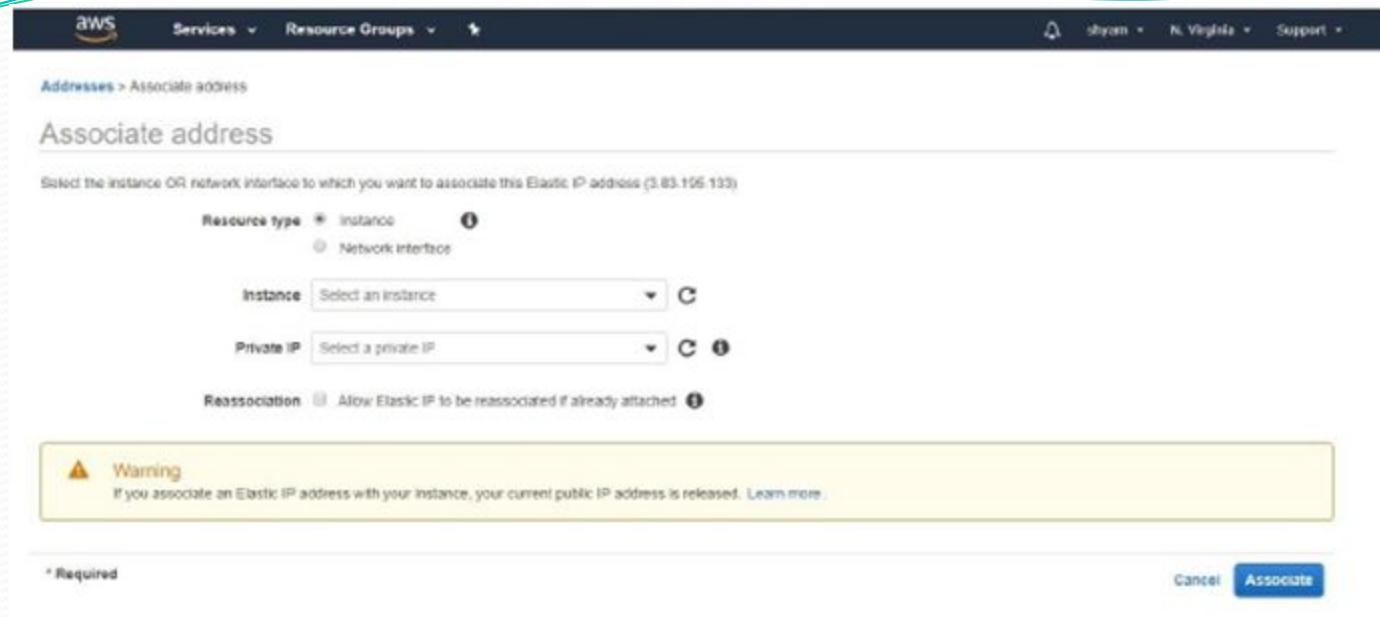
Instance [Select an Instance](#) [C](#)

Private IP [Select a private IP](#) [C](#) [?](#)

Reassociation Allow Elastic IP to be reassigned if already attached [?](#)

Warning
If you associate an Elastic IP address with your instance, your current public IP address is released. [Learn more](#).

* Required [Cancel](#) [Associate](#)



Delete Elastic IP:

aws Services Resource Groups

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 Virtual Private Network (VPN) Customer Gateways

Allocate new address Actions

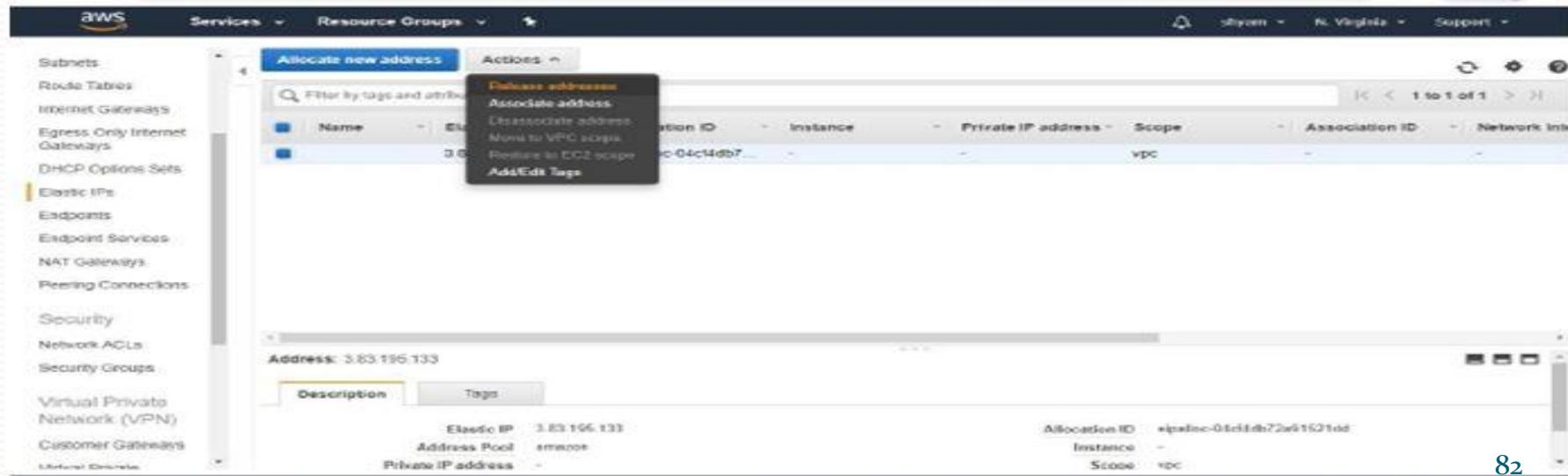
Associate address Disassociate address Move to VPC scope Reassign to EC2 scope Add/Edit Tags

Name	Allocation ID	Instance	Private IP address	Scope	Association ID	Network Intf
3.83.196.133	ipadme-04c14db7	-	-	vpc	-	-

Address: 3.83.196.133

Description Tags

Elastic IP: 3.83.196.133 Allocation ID: ipadme-04c14db72a91521dd
Address Pool: amazone Instance: -
Private IP address: - Scope: vpc



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

Security group Name: enter name for security group

Description: enter some Description for Security Group

VPC: select VPC for security group.

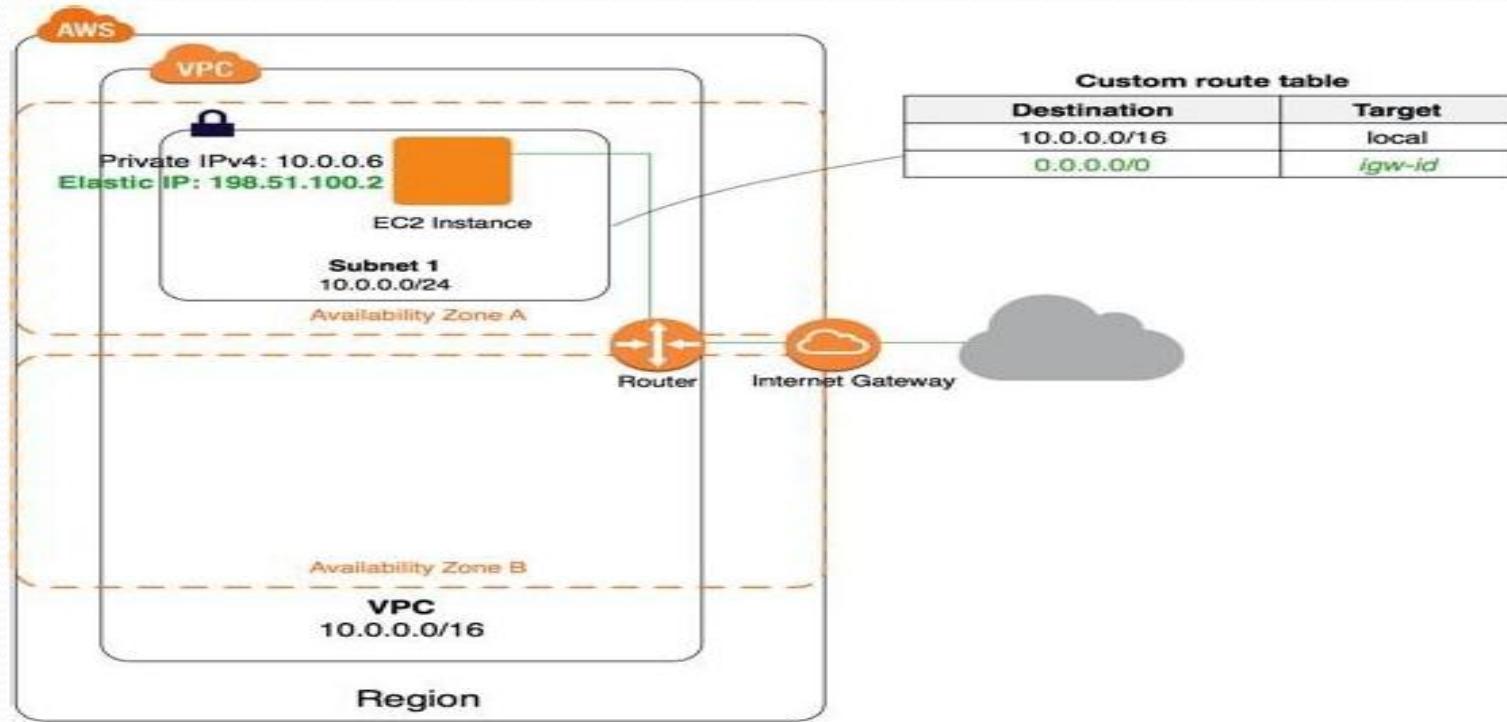
Click on create

The screenshot shows the AWS Management Console with the VPC service selected. On the left, a sidebar lists various network components: Subnets, Route Tables, Internet Gateways, Egress-Only Internet Gateways, DHCP Options Sets, Elastic IPs, Endpoints, Endpoint Services, NAT Gateways, Peering Connections, Security (which is highlighted), Network ACLs, and Security Groups. The main content area is titled 'Create security group' and displays a table of existing security groups. The table columns are: Name, Group ID, Group Name, VPC ID, Type, Description, and Owner. There are 14 entries listed, all of which are EC2-VPC type and owned by the user 'B14927690004'. The first few entries are 'launch-wizard-2', 'launch-wizard-5', 'launch-wizard-12', 'launch-wizard-1', 'launch-wizard-10', 'launch-wizard-4', 'default', 'launch-wizard-7', 'launch-wizard-6', and 'launch-wizard-8'. The search bar at the top of the table says 'Filter by tags and attributes or search by keyword'.

Internet Gateways (IGW)

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

The screenshot shows the AWS VPC Internet Gateways page. On the left sidebar, under the 'Virtual Private Cloud' section, 'Internet Gateways' is selected. The main content area displays a table with one row of data:

Name	ID	State	VPC	Owner
igw-d7f209ac	attached	vpc-97ab3fed	614927698004	

Below the table, there is a detailed view for the internet gateway 'igw-d7f209ac'. It shows the following information:

Internet gateway: igw-d7f209ac

Description	Tags
ID: igw-d7f209ac State: attached	Attached VPC ID: vpc-97ab3fed Owner: 614927698004

Attach to VPC

Select Internet Gateway and go to actions and click on attach to VPC

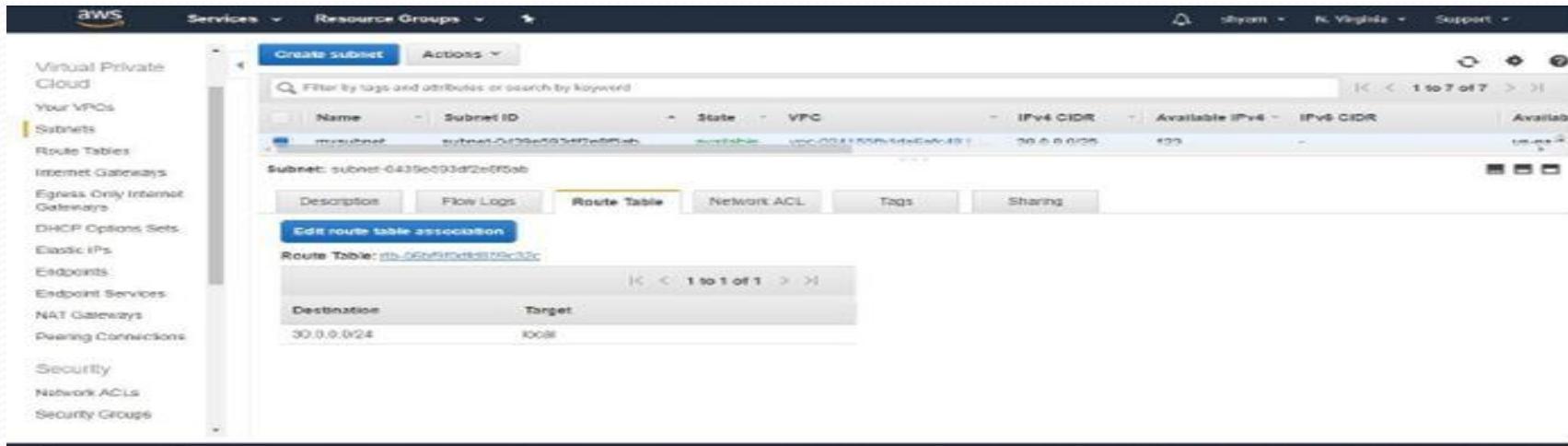
The screenshot shows the AWS Management Console interface for managing Internet Gateways. On the left, there's a navigation pane with various services like Virtual Private Cloud, Your VPCs, Subnets, Route Tables, and Internet Gateways. The 'Internet Gateways' section is currently selected. The main content area displays a table of existing Internet Gateways. One row is selected, showing details for an Internet Gateway named 'myig' with ID 'igw-067c5a73598fb123d'. Below the table, there's a summary card for the selected gateway. At the top of the main content area, there's a 'Actions' dropdown menu with several options, including 'Delete Internet gateway', 'Attach to VPC', 'Detach from VPC', 'Associate Tags', and 'Disassociate Tags'. The 'Attach to VPC' option is highlighted with a blue background.

Select VPC id and click on attach

This screenshot shows the 'Attach to VPC' dialog box. At the top, it says 'Internet gateways > Attach to VPC'. Below that, there's a descriptive text: 'Attach an internet gateway to a VPC to enable communication with the internet. Specify the VPC you would like to attach below.' A dropdown menu labeled 'VPC*' contains the value 'vpc-024155fbfdafaf48'. To the right of the dropdown is a small info icon. At the bottom of the dialog, there are two buttons: 'Cancel' and 'Attach'. There's also a note at the bottom left: '† AWS Command Line Interface command'. A small note at the very bottom left indicates that the VPC selection is required.

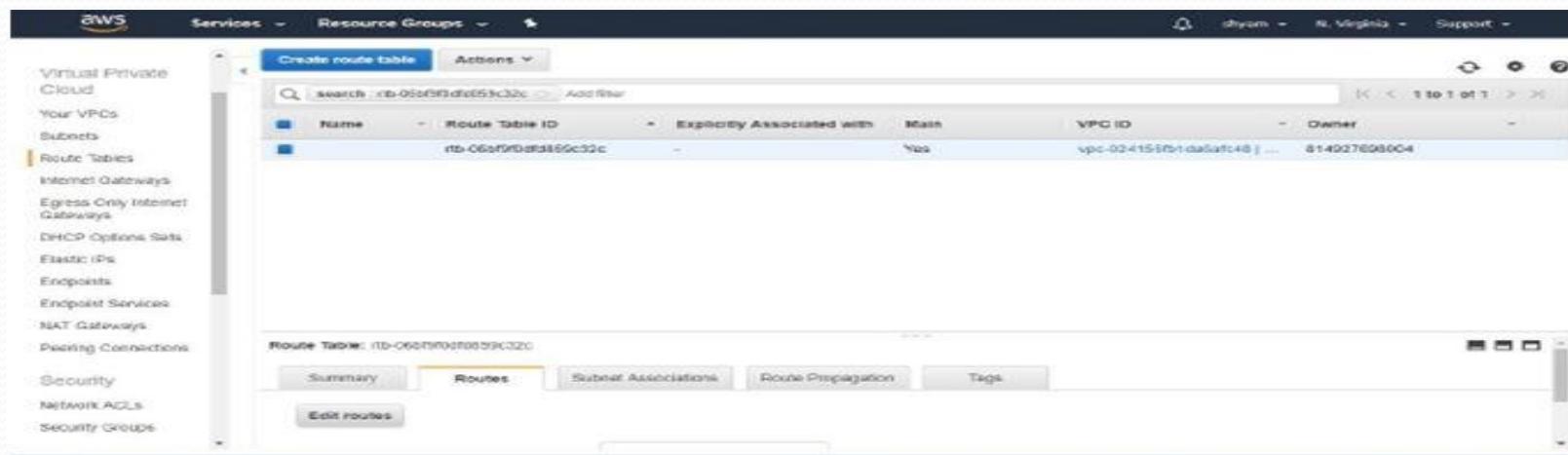
Add Internet gateway to Subnet

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



This screenshot shows the AWS VPC console interface for managing subnets. On the left sidebar, 'Subnets' is selected under the 'Virtual Private Cloud' section. In the main content area, a table lists subnets, including one named 'intsubnet'. Below the table, a 'Route Table' tab is selected, showing a single route entry for 'Destination: 0.0.0.0/24' and 'Target: local'. This indicates that traffic from this subnet will be sent directly to the local VPC without going through an internet gateway.

Click on Routes on below panel and click on Edit Routes



This screenshot shows the AWS VPC console interface for managing route tables. On the left sidebar, 'Route Tables' is selected under the 'Virtual Private Cloud' section. In the main content area, a table lists route tables, including one named 'rtb-06af0fdff855c32c'. Below the table, a 'Routes' tab is selected, showing a single route entry for 'Destination: 0.0.0.0/24' and 'Target: local'. At the bottom of the screen, there is a prominent 'Edit routes' button, which is used to modify the existing route or add new ones.

Click on Add route

Destination: enter global ID (o.o.o.o/o)

Target: select previously created Internet Gateway

Click on save Routes

Destination	Target	Status	Propagated
0.0.0.0/0	local	active	No

Add route

* Required

Cancel Save routes

Detach Internet Gateway

Internet Gateway and go to actions and click on Detach from VPC

Name	ID	VPC	Owner
myig	igw-067c5a735981b123d	vpc-004150fb1da...	014927650004
	igw-d7f209ec	attached	vpc-03d166b31a1a1d1 myvpc

Create internet gateway Actions

Filter by tags and attributes

1 to 2 of 2

Internet gateway: igw-067c5a735981b123d

Description Tags

ID: igw-067c5a735981b123d State: attached Attached VPC ID: vpc-03d166b31a1a1d1 | myvpc Owner: 014927650004

NAT(Network Address Translation) Gateways

NAT gateway, you must specify the public subnet in which the NAT gateway should reside. 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.

Creating NAT Gateway

- Select NAT gateway on left side panel of your VPC section and click on create NAT Gateway
- Subnet: select public subnet to create NAT gate way
- 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

NAT Gateways > Create NAT Gateway

Create NAT Gateway

Create a NAT gateway and assign it an elastic IP address. [Learn more](#)

Subnet: subnet-0439e0530726bf94b

Elastic IP Allocation ID: elipalloc-01ddc65d64d45d0c

New EIP (35.153.252.118) creation successful.

* Required

Create a NAT Gateway

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

Route tables > Edit routes

Edit routes

Destination	Target	Status	Propagated
0.0.0.0/24 0.0.0.0/0	nat-0275a161ab1782690	active	No

Add route

* Required

Cancel Save routes

VPC Peering:

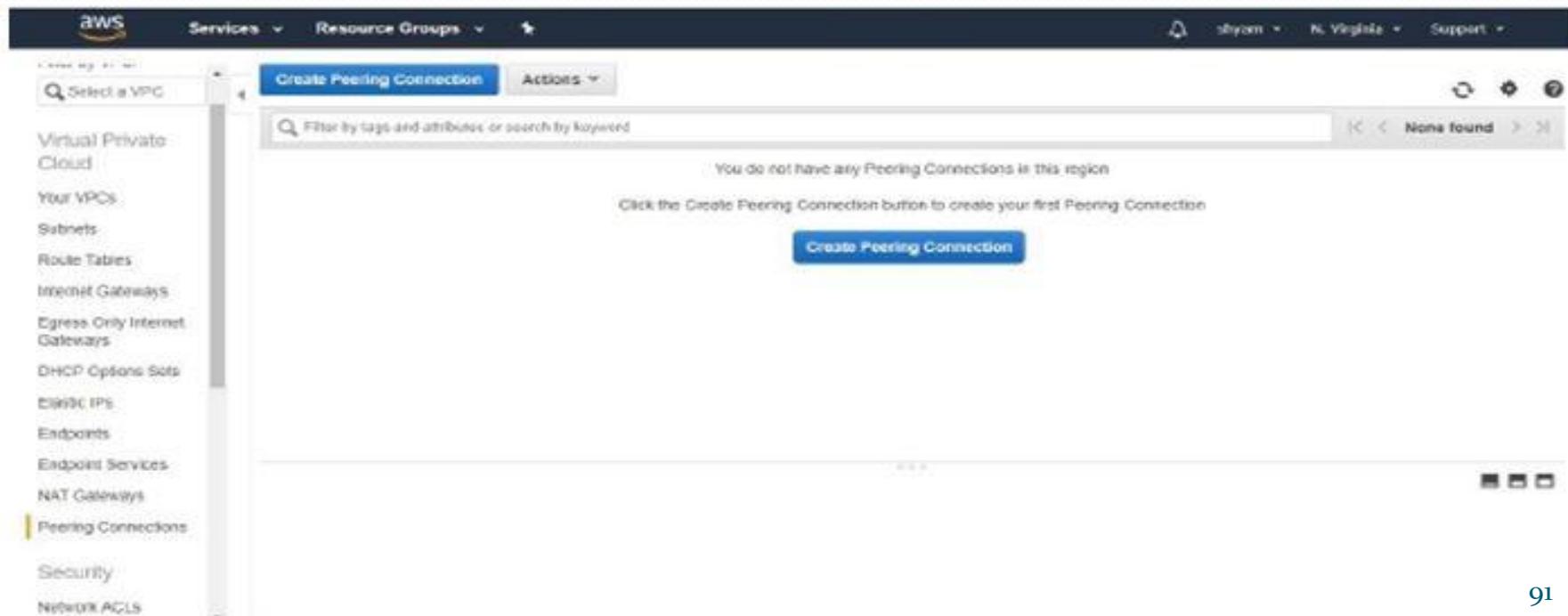
Peering Connection is used to communicate from one VPC to another from private cluster or subnet to another VPC private subnet.

Constraints for VPC Peering:

- No overlapping IP
- No Transitive Peering
- No NAT routing between two VPC's

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 our 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

The screenshot shows the 'Create Peering Connection' page in the AWS Management Console. At the top, there's a navigation bar with 'Services', 'Resource Groups', and other account information. The main section is titled 'Create Peering Connection'. It has two main sections: 'Select a local VPC to peer with' and 'Select another VPC to peer with'. In the first section, a 'Peering connection name tag' input field contains 'mypeer'. Below it, a dropdown menu labeled 'VPC (Requester)' shows 'VPC-C24155D10898C46'. A table lists CIDR ranges: '30.0.0.0/24' with status 'ASSOCIATED'. In the second section, the 'Account' dropdown is set to 'My account'.

CIDRs	CIDR	Status	Status Reason
	30.0.0.0/24	● ASSOCIATED	

Select another VPC to peer with

Account My account Another account

Region This region (us-east-1) Another Region

VPC (Acceptor)*

CIDRs	CIDR	Status	Status Reason
	172.31.0.0/16	●	355008M0

* Required Cancel **Create Peering Connection**

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

Services AWS CloudWatch Metrics Resource Groups AWS Lambda

Actions AWS Lambda Metrics

Peering Connection AWS Lambda Metrics

Accept Request Reject Request Delete VPC Peering Connection Edit CloudWatch Settings Edit DNS Settings Add/Edit Tags

Name	Requester VPC	Acceptor VPC	Requester CIDRs	Acceptor CIDRs	Requester Region
mypeer	vpc-024155fb1da26...	vpc-97ab0fd...	30.0.0.0/24	-	us-east-1

Peering Connection: pcc-06941e174000d37040

Description ClassicLink DNS Route Tables Tags

Requester VPC owner: 014927098604
Requester VPC ID: vpc-024155fb1da2645
Requester VPC Region: N. Virginia (us-east-1)

Acceptor VPC owner: 014927098604
Acceptor VPC ID: vpc-97ab0fd...

Acceptor VPC Region: N. Virginia (us-east-1)

VPC Flow logs:

VPC Flow Logs is a feature that enables you to capture information about the IP traffic going to and from network interfaces in your VPC. Flow log data can be published to Amazon Cloud Watch Logs or Amazon S3.

Creating VPC Flow logs:

Select VPC for you want applying flow logs then Choose Create Flow logs Option

The screenshot shows the AWS VPC Details page for a specific VPC. The top navigation bar includes the AWS logo, Services dropdown, search bar, and user information (Sundarraj, N. Virginia, Support). The left sidebar has a 'New VPC Experience' section and lists VPC components like Subnets, Route Tables, Internet Gateways, Egress Only Internet Gateways, Carrier Gateways, DHCP Options Sets, Elastic IPs, and Managed Prefix Lists. The main content area shows the VPC ID (vpc-0c70a998da8fe165c), State (Available), and various configuration details such as DHCP options set, Main route table, and DNS resolution. At the bottom of the main content, there are tabs for CIDRs, Flow logs (which is currently selected), and Tags. Below this, a detailed view of the Flow logs section shows one entry (1/1) with an 'Info' link, a delete button, an Actions dropdown, and a 'Create flow log' button. A search bar and pagination controls are also present.

Provide the Following Information

Name-optional

Filter- Choose any one from Accept, Reject and All

Maximum Aggregation Interval- 1 minutes or 10 minutes

Destination-Send to CloudWatch Logs or send to an Amazon S3 bucket

Log record Format- AWS default format or Custom format

Tags-Label that you assign to an AWS resource (maximum 50 tags)

Flow log settings

Name - optional

my-flow-log-01

Filter

The type of traffic to capture (accepted traffic only, rejected traffic only, or all traffic).

- Accept
- Reject
- All

Maximum aggregation interval [Info](#)

The maximum interval of time during which a flow of packets is captured and aggregated into a flow log record.

- 10 minutes
- 1 minute

Destination

The destination to which to publish the flow log data.

- Send to CloudWatch Logs
- Send to an Amazon S3 bucket

Destination log group [Info](#)

The name of the Amazon CloudWatch log group to which the flow log is published. A new log stream is created for each monitored network interface.

▼ 

Fields in VPC Flow logs

Version- VPC Flow Logs version, If you use the default format, the version is 2. If you use a custom format, the version is the highest version among the specified fields

account-id- the AWS account ID of the owner of the source network interface for which traffic is recorded.

interface-id- the ID of the network interface for which the traffic is recorded.

Srcaddr- The source address for incoming traffic

Dstaddr- The destination address for outgoing traffic

Srcport- The source port of the traffic.

Dstport- The destination port of the traffic.

Protocol- The IANA protocol number of the traffic

Packets- The number of packets transferred during the flow

Bytes- The number of bytes transferred during the flow.

Start- The time, in Unix seconds, when the first packet of the flow was received within the aggregation interval.

End- The time, in Unix seconds, when the last packet of the flow was received within the aggregation interval.

Action- The action that is associated with the traffic Accept or Reject

log-status- The logging status of the flow log 1) Ok 2)No data 3) Skip Data

Event Date	account-id	interface-id	srcaddr	dstaddr	srcport	dstport	protocol	packets	bytes	start	end	action	log-status
# 2 49304625987015	eni-30076469	107.170.242.27	172.31.8.238	123	123	17	1	76	1433806982	1433807038	ACCEPT OK		
# 2 49304625987015	eni-30076469	172.31.8.238	107.170.242.27	123	123	17	1	76	1433806982	1433807038	ACCEPT OK		
# 2 49304625987015	eni-30076469	79.39.7.58	172.31.8.238	54517	23	6	3	180	1433807174	1433807218	REJECT OK		
# 2 49304625987015	eni-30076469	71.6.135.131	172.31.8.238	15324	21379	6	1	40	1433807224	1433807278	REJECT OK		
# 2 49304625987015	eni-30076469	172.31.8.238	106.63.36.35	323	123	17	1	76	1433807261	1433807338	ACCEPT OK		
# 2 49304625987015	eni-30076469	106.61.56.35	172.31.8.238	123	123	17	1	76	1433807281	1433807338	ACCEPT OK		
# 2 49304625987015	eni-30076469	172.31.8.238	23.226.142.216	123	123	17	1	76	1433807250	1433807396	ACCEPT OK		
# 2 49304625987015	eni-30076469	23.226.142.216	172.31.8.238	40022	24	6	1	40	1433807350	1433807396	REJECT OK		
# 2 49304625987015	eni-30076469	23.226.142.216	172.31.8.238	123	123	17	1	76	1433807350	1433807396	ACCEPT OK		
# 2 49304625987015	eni-30076469	50.133.6.35	172.31.8.238	123	123	17	1	76	1433807431	1433807458	ACCEPT OK		
# 2 49304625987015	eni-30076469	172.31.8.238	50.133.6.38	167	123	123	17	1	76	1433807431	1433807458	ACCEPT OK	
# 2 49304625987015	eni-30076469	107.170.242.27	172.31.8.238	123	123	17	1	76	1433807529	1433807579	ACCEPT OK		
# 2 49304625987015	eni-30076469	172.31.8.238	107.170.242.27	123	123	17	1	76	1433807529	1433807579	ACCEPT OK		
# 2 49304625987015	eni-30076469	316.211.0.90	172.31.8.238	46373	5080	6	1	40	1433807649	1433807699	REJECT OK		
# 2 49304625987015	eni-30076469	222.322.52.157	172.31.8.238	56354	22	6	3	180	1433807694	1433807999	REJECT OK		
# 2 49304625987015	eni-30076469	199.217.137.85	172.31.8.238	5064	5060	17	1	443	1433807944	1433807999	REJECT OK		
# 2 49304625987015	eni-30076469	107.170.242.27	172.31.8.238	123	123	17	1	76	1433808069	1433808119	ACCEPT OK		
# 2 49304625987015	eni-30076469	172.31.8.238	107.170.242.27	123	123	17	1	76	1433808069	1433808119	ACCEPT OK		

Transit Gateway

A *transit gateway* is a network transit hub that you can use to interconnect your virtual private clouds (VPCs) and on-premises networks.

Key Components of Transit gateway:

Attachments : One or more VPCs , Connect SD-WAN/third-party network appliance, AWS Direct Connect gateway, peering connection with another transit gateway, VPN connection to a transit gateway

Transit gateway Maximum Transmission Unit (MTU): The maximum transmission unit (MTU) of a network connection is the size, in bytes, of the largest permissible packet that can be passed over the connection.

Transit gateway route table: A transit gateway has a default route table and can optionally have additional route tables. A route table includes dynamic and static routes that decide the next hop based on the destination IP address of the packet.

Associations : Each attachment is associated with exactly one route table. Each route table can be associated with zero to many attachments.

Route propagation : VPC, VPN connection, or Direct Connect gateway can dynamically propagate routes to a transit gateway route table. With a Connect attachment, the routes are propagated to a transit gateway route table by default. With a VPC, you must create static routes to send traffic to the transit gateway .

Load Balancing (LB)

Elastic Load Balancer:

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

Working Mechanism of Load Balancer:

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.



Load Balancing supports Three types of load balancers:

1. Application Load Balancer
2. Network Load Balancer
3. Gateway Load Balancer

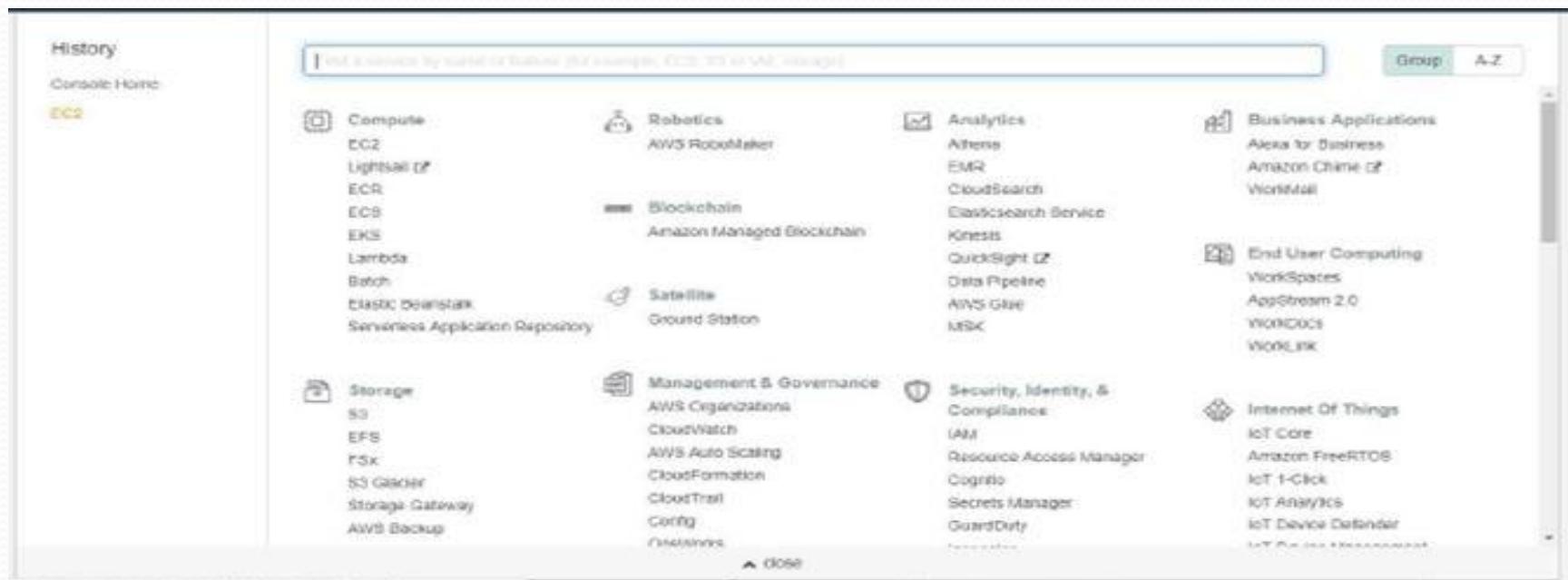
Configure these load balancers with Application Load Balancers and Network Load Balancers, you register targets in target groups, and route traffic to the target groups. With Classic Load Balancers, you register instances with the load balancer.

Classic Load Balancer (CLB) Use Cases:

Use Case : Create ELB and Attach Instances to ELB: Create 3 ec2 instances

Login into AWS console:

Select services EC2



Click on launch instances:

The screenshot shows the AWS EC2 Dashboard. On the left, there's a sidebar with navigation links for EC2 Dashboard, Events, Tags, Reports, Limits, Instances (selected), Launch Templates, Spot Requests, Reserved Instances, Dedicated Hosts, Scheduled Instances, Capacity Reservations, Images, AMIs, Bundle Tasks, Elastic Block Store (selected), Volumes, Snapshots, and Lifecycle Manager. The main content area has a title 'Create Instance' with a sub-instruction: 'To start using Amazon EC2 you will want to launch a virtual server, known as an Amazon EC2 instance.' Below this is a large blue button labeled 'Launch Instance'. A note below the button says 'Note: Your instances will launch in the US East (N. Virginia) region.' To the right of the note are two sections: 'Service Health' and 'Scheduled Events'. The 'Service Health' section shows 'Service Status: US East (N. Virginia)' with a green status icon and 'Availability Zone Status:' for us-east-1a through us-east-1e, each also with a green status icon and the text 'Availability zone is operating normally'. The 'Scheduled Events' section shows 'US East (N. Virginia):' followed by 'No events'. On the far right, there's a sidebar with links to 'Getting Started Guide', 'Documentation', 'All EC2 Resources', 'Forums', 'Pricing', and 'Contact Us'. Below that is a section titled 'AWS Marketplace' with a note about finding software trial products, a product listing for Barracuda CloudGen Firewall for AWS - PAYG, and links to view all infrastructure software and another product listing for Matillion ETL for Amazon Redshift.

Choose AMI : Click on Ubuntu Server 16.04 LTS (HVM), SSD Volume Type

The screenshot shows the AWS Step 1: Choose an Amazon Machine Image (AMI) interface. At the top, there are tabs for 1. Choose AMI, 2. Choose Instance Type, 3. Configure Instance, 4. Add Storage, 5. Add Tags, 6. Configure Security Group, and 7. Review. The 1. Choose AMI tab is selected. On the left, there's a sidebar with 'Amazon RDS' and a 'Launch a database using RDS' button. The main area lists three AMI options:

- Ubuntu Server 16.04 LTS (HVM), SSD Volume Type - ami-0565af0e262977273 (64-bit x86) / ami-006ede09622dedc9b (64-bit Arm)**: Selected. Options: 64-bit (x86) (radio button selected), 64-bit (Arm). Root device type: hba. Visualization type: t2n. ENA Enabled: Yes. Buttons: Select, Next Step.
- Microsoft Windows Server 2019 Base - ami-02404914069c475f**: Options: 64-bit (x86). Root device type: hba. Visualization type: t2n. ENA Enabled: Yes. Buttons: Select, Next Step.
- Deep Learning AMI (Ubuntu) Version 22.0 - ami-000665e00601404c4**: Options: 64-bit (x86). Root device type: hba. Visualization type: t2n. ENA Enabled: Yes. Buttons: Select, Next Step.

At the bottom right, there are 'Cancel and Exit' and 'Next Step' buttons.

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

The screenshot shows the AWS Step 2: Choose an Instance Type interface. At the top, there are tabs for 1. Choose AMI, 2. Choose Instance Type, 3. Configure Instance, 4. Add Storage, 5. Add Tags, 6. Configure Security Group, and 7. Review. The 2. Choose Instance Type tab is selected. On the left, there's a sidebar with 'All instance types' and 'Current generation'. The main area lists instance types:

Family	Type	vCPUs	Memory (GB)	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 (Selected)	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

At the bottom right, there are 'Cancel', 'Previous', 'Review and Launch', and 'Next: Configure instance Details' buttons.

In configure instance details give the value 3 in *number of instances* section &click on *Advanced Details* section and paste below user data in that box and click *Review and Launch*.

Userdata:

```
#!/bin/bash
```

```
Sudo apt-get update -y
```

```
Sudo apt-get install apache2 -y
```

```
echo "This is prepaid page $(date)" > /var/www/html/prepaid/test.html
```

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

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.

Number of instances Launch into Auto Scaling Group

You may want to consider launching these instances into an Auto Scaling Group to help you maintain application availability and for easy scaling in the future. Learn how Auto Scaling can help your application stay healthy and cost effective.

Purchasing option Request Spot Instances

Network Create new VPC

Subnet Create new subnet

Auto-assign Public IP

Placement group Add instance to placement group

Capacity Reservation Create new Capacity Reservation

[Cancel](#) [Previous](#) [Review and Launch](#) [Next: Add Storage](#)

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

Step 3: Configure Instance Details

Monitoring (1) Enable CloudWatch detailed monitoring
Additional charges apply.

Tenancy (1) Shared - Run a shared hardware instance (2)
Additional charges will apply for dedicated tenancy.

Elastic Inference (1) Add an Elastic Inference accelerator
Additional charges apply.

T2/T3 Unlimited (1) Enable
Additional charges may apply.

Advanced Details

User data (1) 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
```

Cancel **Previous** **Review and Launch** **Next: Add Storage**

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

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 Edit security groups

Security group name launch-wizard-4
Description launch-wizard-4 created 2019-04-19T10:36:10.994+00:00

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

Instance Details Edit instance details

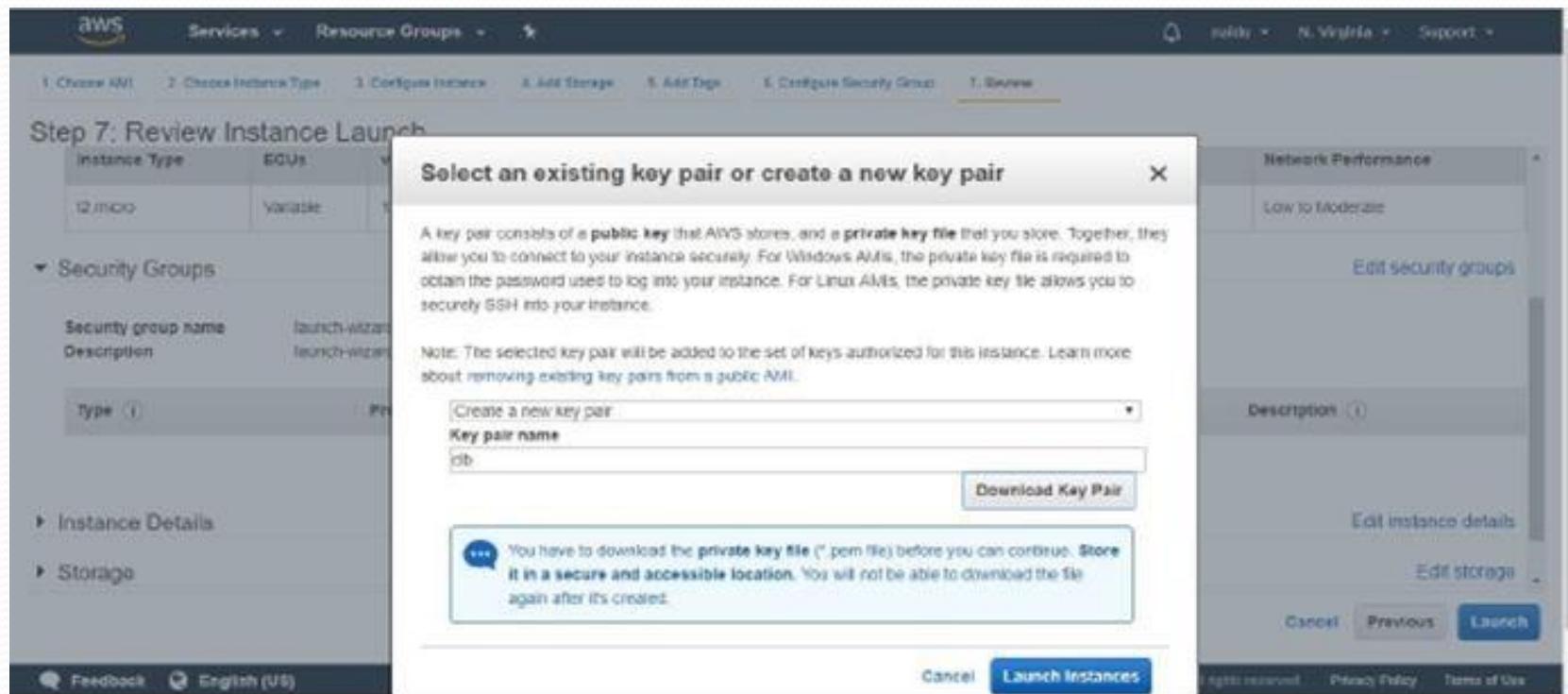
Storage Edit storage

Tags Edit tags

Cancel **Previous** **Launch**

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

Then click on Launch instance



Next click on View instance to see the instances in dashboard

The screenshot shows the AWS EC2 Instances dashboard. On the left, there's a sidebar with navigation links: EC2 Dashboard, Events, Tags, Reports, Limits, INSTANCES (with 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 has tabs for Launch Instance, CONSOLE, and ACTIONS. A search bar says "Filter by tags and attributes or search by keyword". Below it is a table with columns: Name, Instance ID, Instance Type, Availability Zone, Instance State, Status Checks, Alarm Status, Public DNS (IPv4), and IPv4. There are three rows of data:

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)	IPv4
	i-0047e4d87244ec0ff	t2.micro	us-east-1b	running	initializing	none	ec2-3-62-246-30.compl...	3.32
	i-045ad519309710...	t2.micro	us-east-1b	running	initializing	none	ec2-54-211-107-31.co...	54.2
	i-0e1fd3f76dd4294e4	t2.micro	us-east-1b	running	initializing	none	ec2-3-52-224-51.compl...	3.32

Select an instance above

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

AWS Services Resource Groups Actions

EC2 Dashboard Events Tags Reports Instances Instances Launch Temp Spot Requests Reserved Inst Dedicated Host Scheduled Inst Capacity Reservations Images AMIs Bundle Tasks

Create Security Group

Edit inbound rules

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

Add Rule

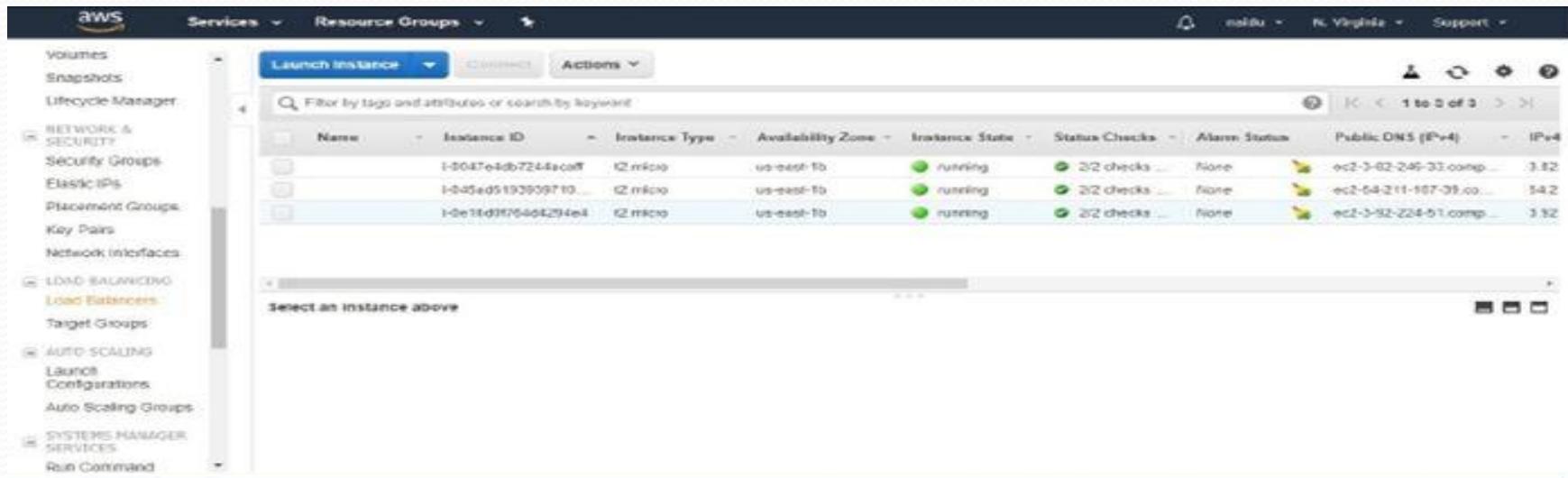
NOTE: Any edits made on existing rules will result in the edited rule being deleted and a new rule created with the new details. This will cause traffic that depends on that rule to be dropped for a very brief period of time until the new rule can be created.

Cancel Save

Type Protocol Port Range Source Description

Create Classic ELB and Attach above three Instances:

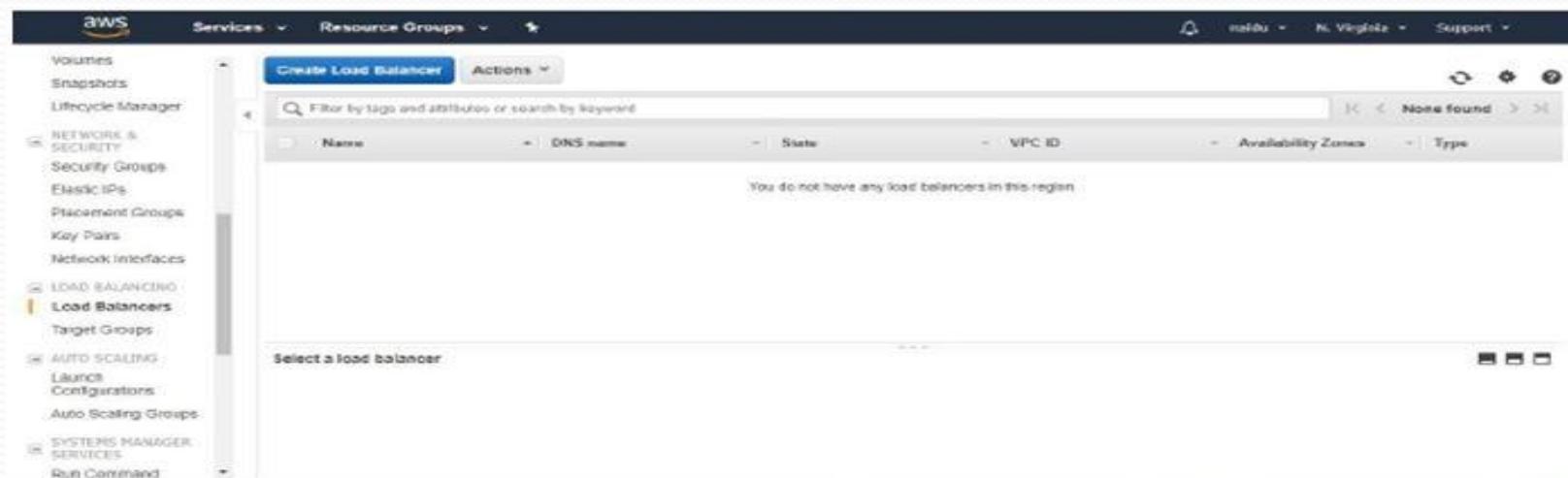
Click on load balancers in the left side panel of ec2 service:



The screenshot shows the AWS Management Console with the EC2 service selected. The left sidebar lists various services like Volumes, Snapshots, and Load Balancers. Under Load Balancers, 'Load Balancers' is selected. The main pane displays a table of three EC2 instances. The columns include Name, Instance ID, Instance Type, Availability Zone, Instance State, Status Checks, Alarm Status, Public DNS (IPv4), and IPv6. All three instances are listed as 'running'.

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)	IPv6
I-0047e4db7244ec0ff	i2.micro	us-east-1b	running	2/2 checks	None	ec2-3-82-245-33.compute-1.amazonaws.com	3.82	
I-025ed5133939710...	i2.micro	us-east-1b	running	2/2 checks	None	ec2-54-211-107-09.compute-1.amazonaws.com	54.2	
I-0e11d3f764d4294e4	i2.micro	us-east-1b	running	2/2 checks	None	ec2-52-224-51.compute-1.amazonaws.com	52.2	

Click on Create Load Balancer



The screenshot shows the AWS Management Console with the Load Balancers service selected. The left sidebar lists various services like Volumes, Snapshots, and Load Balancers. Under Load Balancers, 'Load Balancers' is selected. The main pane displays a table with one row, indicating 'None found'. A message below the table states 'You do not have any load balancers in this region.' Below the table, there is a section labeled 'Select a load balancer'.

Name	DNS name	Status	VPC ID	Availability Zones	Type

Application Load Balancer	Network Load Balancer	Gateway Load Balancer	Classic Load Balancer
 Create <p>Choose an Application Load Balancer when you need a flexible feature set for your web applications with HTTP and HTTPS traffic. Operating at the request level, Application Load Balancers provide advanced routing and visibility features targeted at application architectures, including microservices and containers.</p> <p>Learn more ></p>	 Create <p>Choose a Network Load Balancer when you need ultra-high performance, TLS offloading at scale, centralized certificate deployment, support for UDP, and static IP addresses for your application. Operating at the connection level, Network Load Balancers are capable of handling millions of requests per second securely while maintaining ultra-low latencies.</p> <p>Learn more ></p>	 Create <p>Choose a Gateway Load Balancer when you need to deploy and manage a fleet of third-party virtual appliances that support GENEVE. These appliances enable you to improve security, compliance, and policy controls.</p> <p>Learn more ></p>	<p>PREVIOUS GENERATION for HTTP, HTTPS, and TCP</p> Create <p>Choose a Classic Load Balancer when you have an existing application running in the EC2-Classic network.</p> <p>Learn more ></p>

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

Step 1: Define Load Balancer

Basic Configuration

This wizard will walk you through setting up a new load balancer. Begin by giving your new load balancer a unique name so that you can identify it from other load balancers you might create. You will also need to configure ports and protocols for your load balancer. Traffic from your clients can be routed from any load balancer port to any port on your EC2 instances. By default, we've configured your load balancer with a standard web server on port 80.

Load Balancer name:

Create LB inside:

Create an internal load balancer: (what's this?)

Enable advanced VPC configuration:

Listener Configuration:

Load Balancer Protocol	Load Balancer Port	Instance Protocol	Instance Port
HTTP	80	HTTP	80

Add

Cancel **Next: Assign Security Groups**

Click on next assign security groups
 Selecting security group
 By default it is assigned with Default Security group

Step 2: Assign Security Groups

You have selected the option of having your Elastic Load Balancer inside of a VPC, which allows you to assign security groups to your load balancer. Please select the security groups to assign to this load balancer. This can be changed at any time.

Assign a security group: Create a new security group
 Select an existing security group

Filter: VPC security groups

Security Group ID	Name	Description	Actions
sg-04fbcd1dd0f0f3d55	AutoScaling-Security-Group-1	AutoScaling-Security-Group-1 (2019-03-27 10:31:32.056+06:30)	Copy to new
sg-02fb50ca0d775ae3d	AutoScaling-Security-Group-2	AutoScaling-Security-Group-2 (2019-03-27 10:37:22.380+06:30)	Copy to new
sg-00c72000b122e4b0	AutoScaling-Security-Group-3	AutoScaling-Security-Group-3 (2019-03-27 10:46:06.509+06:30)	Copy to new
sg-0fb3400bd54c05983	AutoScaling-Security-Group-4	AutoScaling-Security-Group-4 (2019-03-27 15:57:08.377+06:30)	Copy to new
<input checked="" type="checkbox"/> sg-922615d6	default	default VPC security group	Copy to new
sg-01a1291b174800e0	Launch-wizard-1	Launch-wizard-1 created 2019-03-27T15:48:22.716+05:30	Copy to new
sg-0ec11acff4d0d60d	Launch-wizard-2	Launch-wizard-2 created 2019-03-25T09:21:56.660+05:30	Copy to new

Cancel **Previous** **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 you can 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	

Navigation: Cancel | Previous | Next: Add EC2 Instances

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

The screenshot shows the AWS Step 5: Add EC2 Instances configuration page. At the top, there is a navigation bar with links: 1. Define Load Balancer, 2. Assign Security Groups, 3. Configure Security Settings, 4. Configure Health Check, 5. Add EC2 Instances (which is underlined in green), 6. Add Tags, and 7. Review. Below the navigation bar, the title "Step 5: Add EC2 Instances" is displayed, followed by a sub-instruction: "The table below lists all your running EC2 instances. Check the boxes in the Select column to add those instances to this load balancer." A VPC section shows "VPC vpc-1aa011f50 (172.31.0.0/16)". A table lists four EC2 instances:

Instance	Name	Status	Security groups	Zone	Subnet ID	Subnet CIDR
H047e4db7244ecff		running	launch-wizard-4	us-east-1b	subnet-9d56fb3	172.31.0.0/20
Hw1id0794c4254e4		running	launch-wizard-4	us-east-1b	subnet-9d56fb3	172.31.0.0/20
I045ad516393571875		running	launch-wizard-4	us-east-1b	subnet-9d56fb3	172.31.0.0/20

Below the table, the "Availability Zone Distribution" section indicates "3 instances in us-east-1b". There is a checkbox for "Enable Cross Zone Load Balancing" which is checked. At the bottom right, there are buttons for "Cancel", "Previous", and "Next: Add Tags".

Click on Next: Add Tags

The screenshot shows the AWS Step 6: Add Tags configuration page. At the top, there's a navigation bar with the AWS logo, Services dropdown, Resource Groups dropdown, and other account-related links. Below the navigation is a horizontal progress bar with steps 1 through 7. Step 6, "Add Tags", is highlighted. A sub-section titled "Step 6: Add Tags" instructs users to apply tags to their resources to help organize and identify them. It includes a note about tag key-value pairs and a link to learn more about tagging. The main area contains a table with two columns: "Key" and "Value". A row is shown with the key "alias" and the value "myelb". There's also a "Create Tag" button.

**Note: Tags is the optional
Click on review and create
Then you see the all configuration options to create ELB**

The screenshot shows the AWS Step 7: Review configuration page. At the top, there's a navigation bar with the AWS logo, Services dropdown, Resource Groups dropdown, and other account-related links. Below the navigation is a horizontal progress bar with steps 1 through 7. Step 7, "Review", is highlighted. A sub-section titled "Step 7: Review" asks users to review the load balancer details before continuing. It lists three configuration sections: "Define Load Balancer", "Configure Health Check", and "Add EC2 instances". Each section has expandable details. The "Edit load balancer definition" link is visible next to the "Define Load Balancer" section. The "Edit health check" link is visible next to the "Configure Health Check" section. The "Edit instances" link is visible next to the "Add EC2 instances" section. At the bottom right are "Cancel", "Previous", and "Create" buttons.

- Make sure the ELB security group Inbound Rule is allowing port 80
- Select load Balancer and click description section below and go to security option in security group in security option and click on that security group name

The screenshot shows the AWS Load Balancers console. On the left, there's a navigation pane with various services like Volumes, Snapshots, and Load Balancers. The 'Load Balancers' section is selected. The main area displays a table of load balancers. One row is selected, showing details: Name (elb), DNS name (elb-255245473.us-east-1.elb.amazonaws.com), State (Active), VPC ID (vpc-1aa111e0), Availability Zones (us-east-1a, us-east-1b), and Type (classic). Below the table, there's a 'Description' section with a 'Source Security Group' dropdown set to 'sg-522650d6, default - default VPC security group' and an 'Edit security groups' button. Under 'Attributes', there's a 'Idle timeout' field set to '60 seconds'.

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

This screenshot shows the 'Instances' tab for the 'elb' load balancer. At the top, there are tabs for Description, Instances, Health check, Listeners, Monitoring, Tags, and Migration. The Instances tab is selected. Below it, a message says 'Connection Draining: Enabled, 300 seconds (Edit)'. There's a 'Edit Instances' button. The main table lists two instances:

Instance ID	Name	Availability Zone	Status	Actions
i-067edbc724e8c0f		us-east-1b	InService (1)	Remove from Load Balancer
i-0619d0754d429e4		us-east-1b	InService (1)	Remove from Load Balancer

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

Name	DNS name	State	VPC ID	Availability Zones	Type
elb	elb-295245473.us-east-1.elb.amazonaws.com (A)	Active	vpc-1aa11f50	us-east-1a, us-east-1b...	classic

Open the browser paste ELB DNS Name



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



Move this file content to apache index.html page

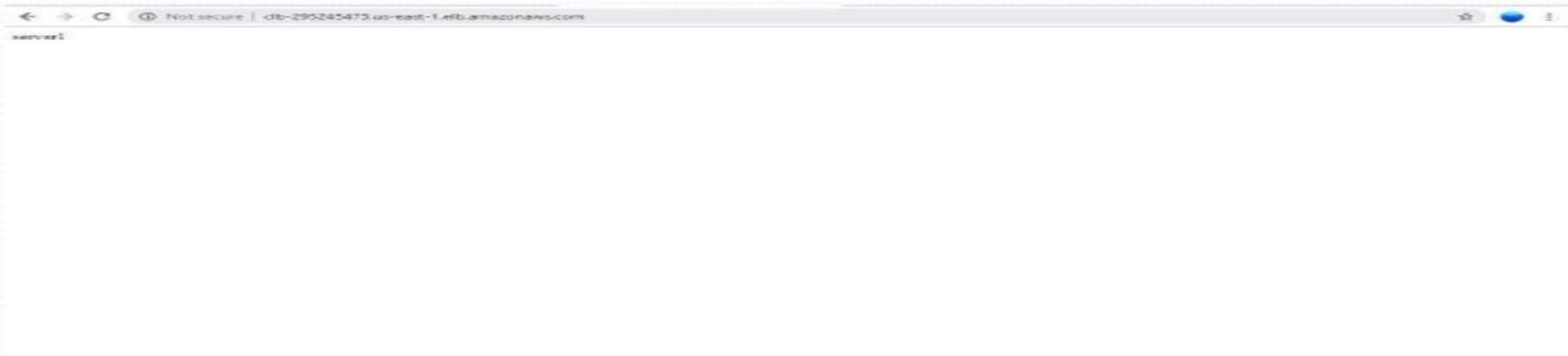
Sudo mv s1 /var/www/html/index.html

Restart apache server

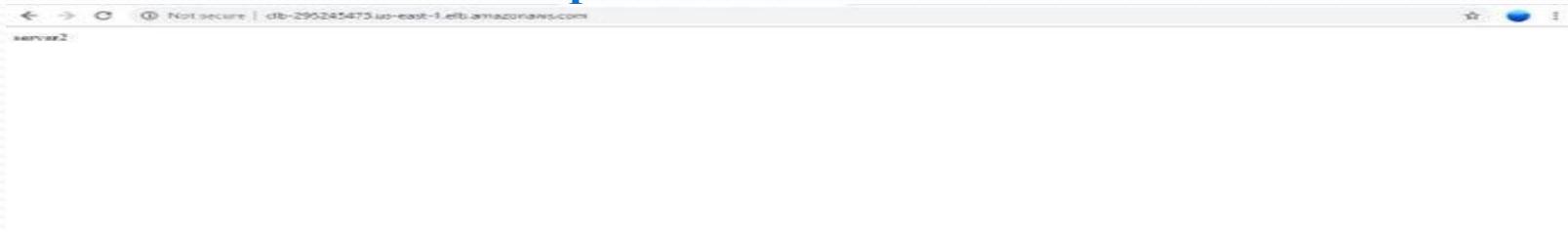
Sudo service apache2 restart

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

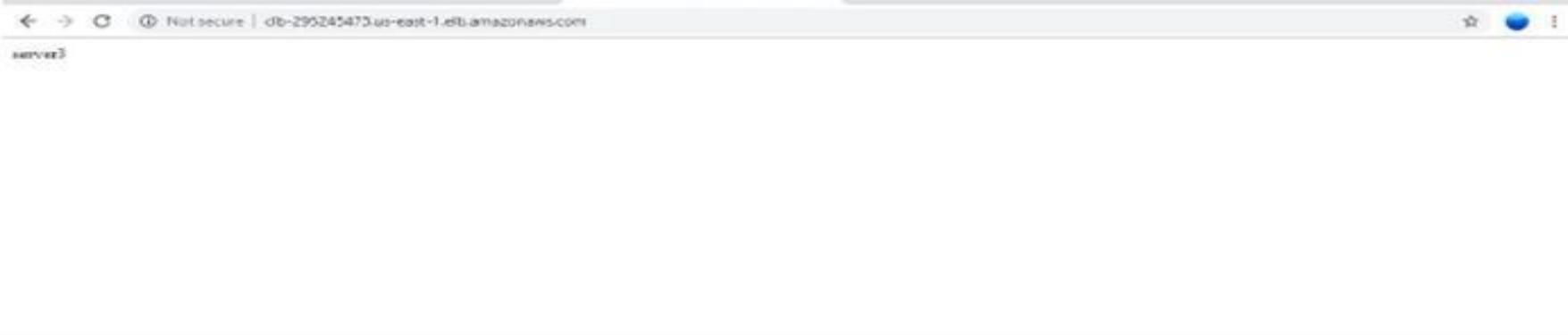
Refresh the browser: Response receive from Server1



Refresh the browser: Response receive from Server2



Refresh the browser: Response receive from Server3



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.

Go to ELB Dashboard and Enable Stickiness

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

The screenshot shows the AWS Management Console with the AWS logo at the top left. The navigation bar includes 'Services', 'Resource Groups', and a search bar. On the left, a sidebar lists various AWS services: Dedicated Hosts, Scheduled Instances, Capacity Reservations, IMAGES, AMIs, Bundle Tasks, ELASTIC BLOCK STORE, Volumes, Snapshots, Lifecycle Manager, NETWORK & SECURITY, Security Groups, Elastic IPs, Placement Groups, Key Pairs, Network Interfaces, and LOAD BALANCING. Under LOAD BALANCING, 'Load Balancers' is selected. The main content area displays a table of load balancers with columns: Name, DNS name, State, VPC ID, Availability Zones, and Type. Two entries are shown: 'db' (dnsname: elb-296245473.us-east-1.elb, state: active, vpc-id: vpc-fac1195, availability-zones: us-east-1a, us-east-1b, type: classic). Below the table, a 'Port Configuration' section shows port 80 and a 'Stickiness' link. A modal dialog box titled 'Edit stickiness' is open, containing three radio button options: 'Disable stickiness' (selected), 'Enable load balancer generated cookie stickiness' (highlighted in blue), and 'Enable application generated cookie stickiness'. At the bottom right of the dialog are 'Cancel' and 'Save' buttons.

Then click on second option to enable stickiness

This screenshot is identical to the one above, but the 'Edit stickiness' dialog has been modified. The 'Enable load balancer generated cookie stickiness' option is now selected, and the 'Expiration Period' field contains the value '300 seconds'. The other options ('Disable stickiness' and 'Enable application generated cookie stickiness') are still available. The 'Cancel' and 'Save' buttons are visible at the bottom right of the dialog.

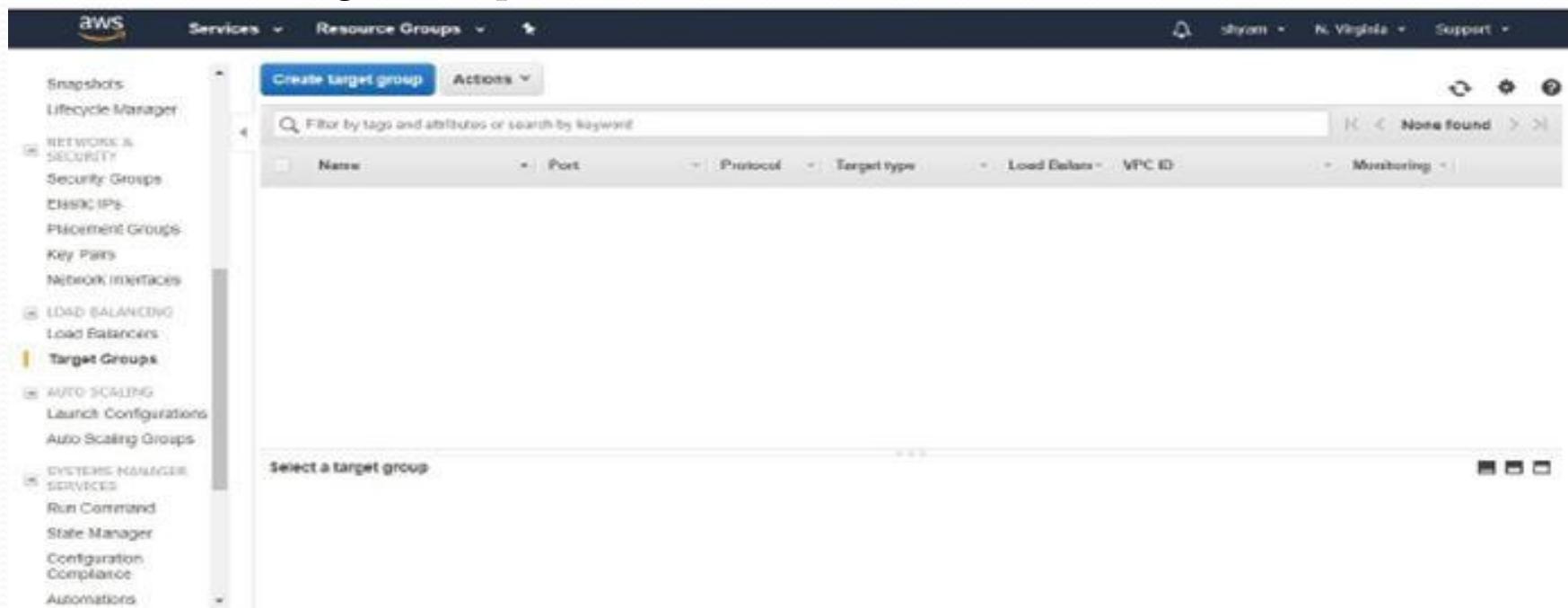
Refresh the page multiple time, You will Observe that request going to specific server only that is server2



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

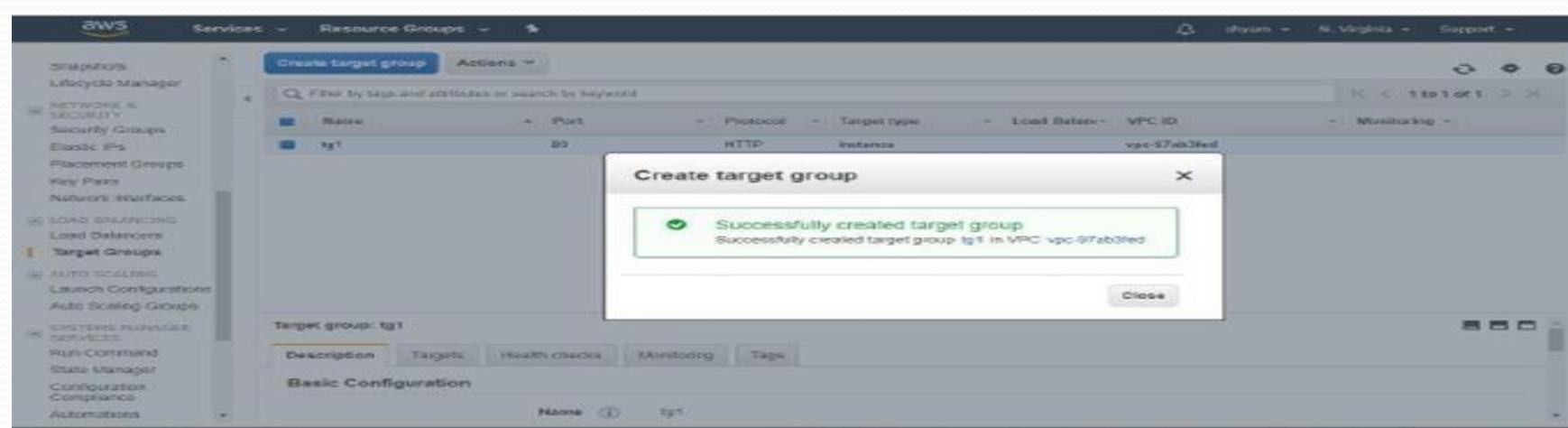
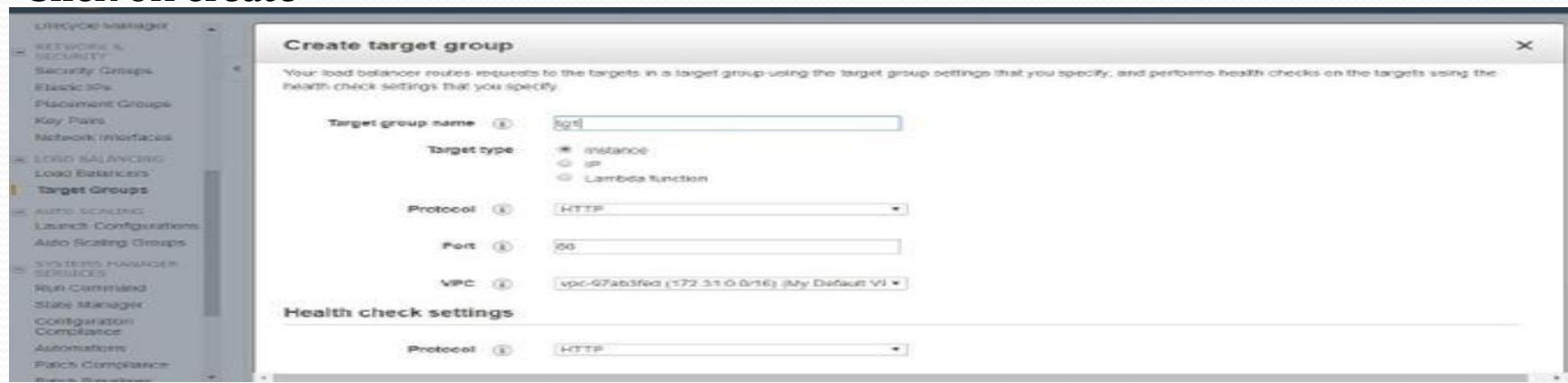
· Protocol: HTTP

· Port: 80

· VPC: select VPC

· Health check settings: for health checks use HTTP protocol and path is /index.html

· Click on create

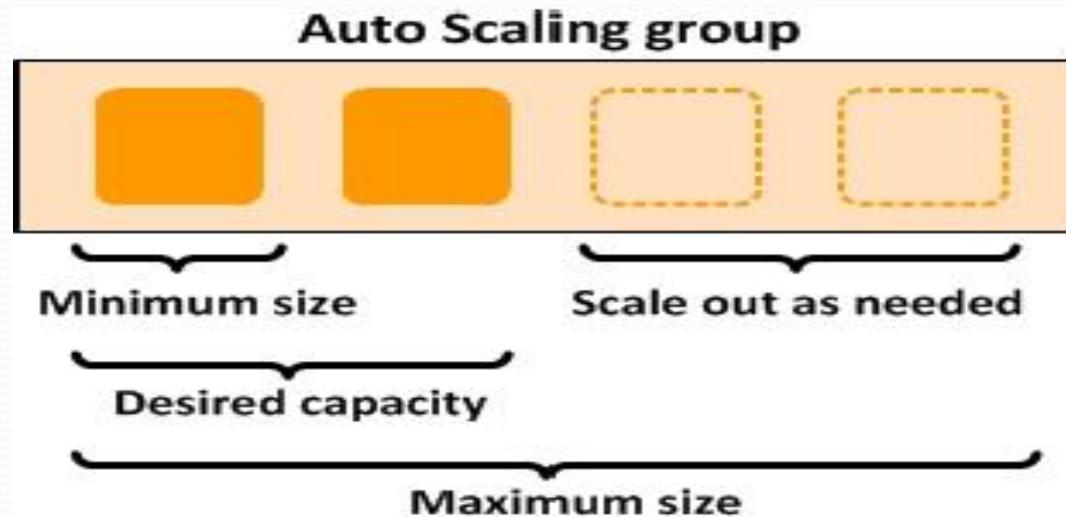


Auto Scaling Groups (ASG)

Auto Scaling

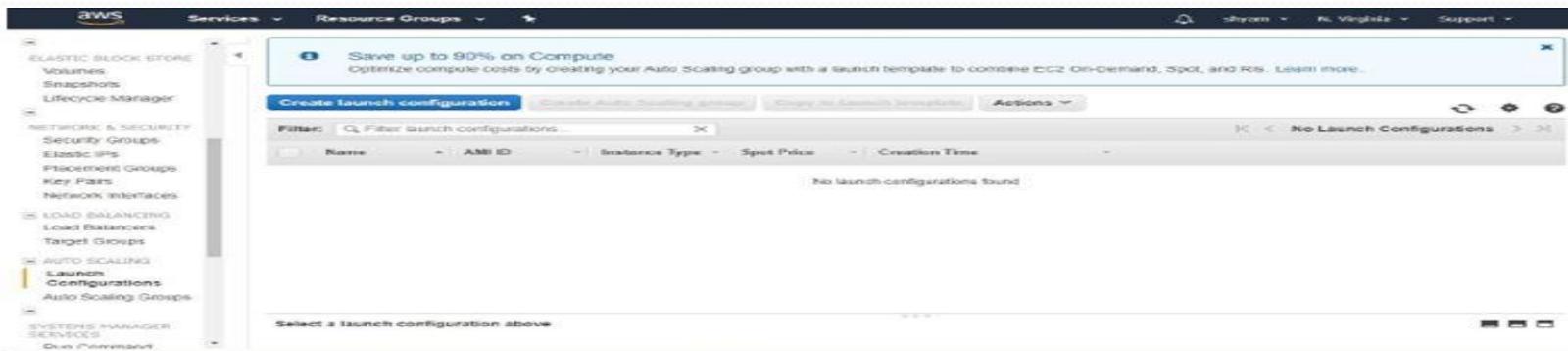
Auto Scaling helps you ensure that you have the correct number of Amazon EC2 instances available to handle the load for your application. You create collections of EC2 instances, called Auto Scaling groups. You can specify the minimum number of instances in each Auto Scaling group, and Amazon EC2 Auto 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.



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 (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

Services ▾ Resource Groups ▾

1. Choose AMI 2. Choose Instance Type 3. Configure details 4. Add Storage 5. Configure Security Group 6. Review

Create Launch Configuration

Name

Purchasing option Request Spot Instances

IAM role None

Monitoring Enable CloudWatch detailed monitoring
Learn more

Advanced Details

Note: 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

Services ▾ Resource Groups ▾

1. Choose AMI 2. Choose Instance Type 3. Configure details 4. Add Storage 5. Configure Security Group 6. Review

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/launchinstancelstorage/about-storage-options-in-amazon-ec2>

Volume Type	Device	Snapshot	Size (GiB)	Volume Type	IOPS	Throughput	Delete on Termination	Encrypted
Root	/dev/sda1	snap-0e34451003d3767e	8	General Purpose (SSD)	100 / 1000	MIA	No	No

Add New Volume

Note: Free tier eligible customers can get up to 30 GB of EBS storage. Learn more about free usage tier eligibility and usage restrictions.

- 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

1. Choose AMI 2. Choose Instance Type 3. Configure details 4. Add storage 5. Configure Security Group 6. 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 13:03:36.859+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.

Click on create launch configuration

[Cancel](#) [Previous](#) [Review](#)

1. Choose AMI 2. Choose Instance Type 3. Configure details 4. Add storage 5. Configure Security Group 6. Review

Create Launch Configuration

Review the details of your launch configuration. You can change any of these settings later.

Improve security of instances launched from this AMI
Your instances may be accessible from anywhere. You can also open additional ports in your security group.

AMI Details
Ubuntu Server 16.04 LTS (HVM)
Ubuntu Server 16.04 LTS (PVHVM)
Root device type: ebs Instance type: t2.micro

Instance Type
t2.micro

Select an existing key pair or create a new key pair

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

Note: The selected key pair will be added to the set of keys authorized for this instance. Learn more about removing existing key pairs from a public AMI.

Choose an existing key pair
[key]

I acknowledge that I have access to the selected private key file (key.pem), and that without this file, I won't be able to log into my instance.

[Cancel](#) [Create launch configuration](#)

Auto Scaling Group-1 is open to the world.
Edit AMI

AMI **Instance Storage (0.08 GB)** **EBS-Optimized Available** **Network Performance**

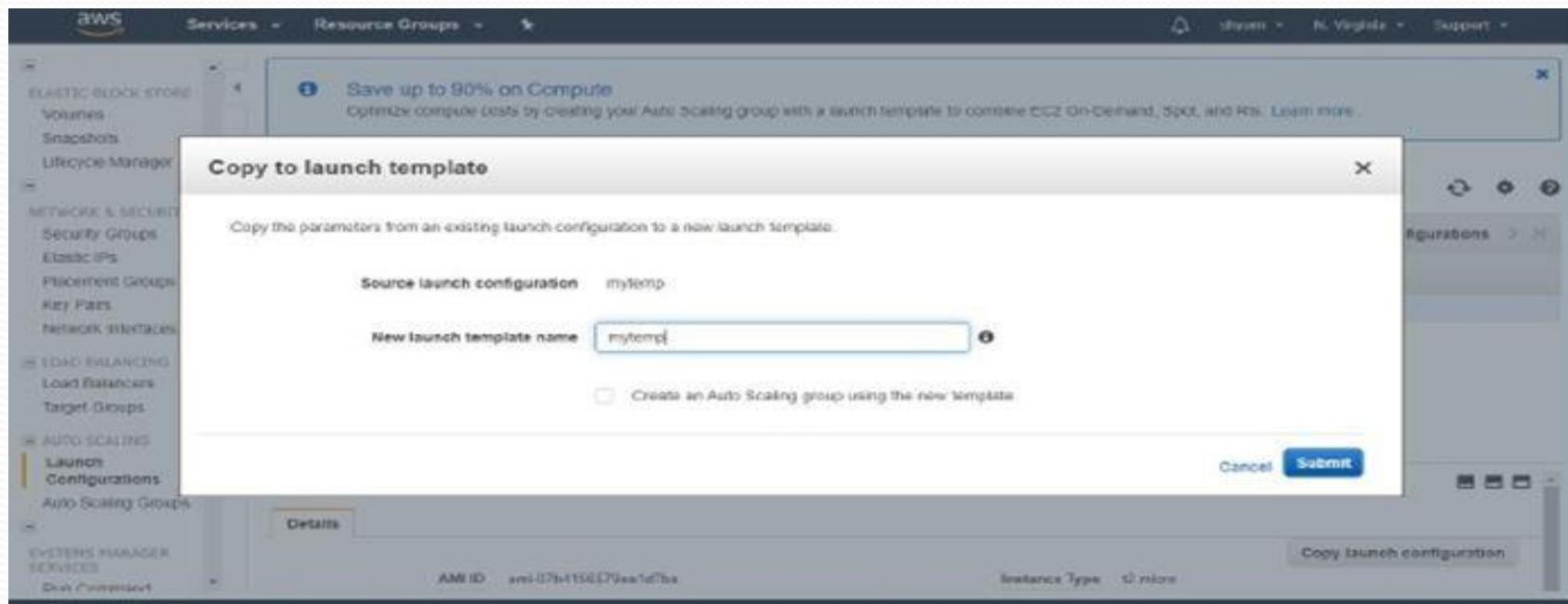
EBS only **Low to Moderate**

[Cancel](#) [Previous](#) [Create launch configuration](#)

Copy to launch template

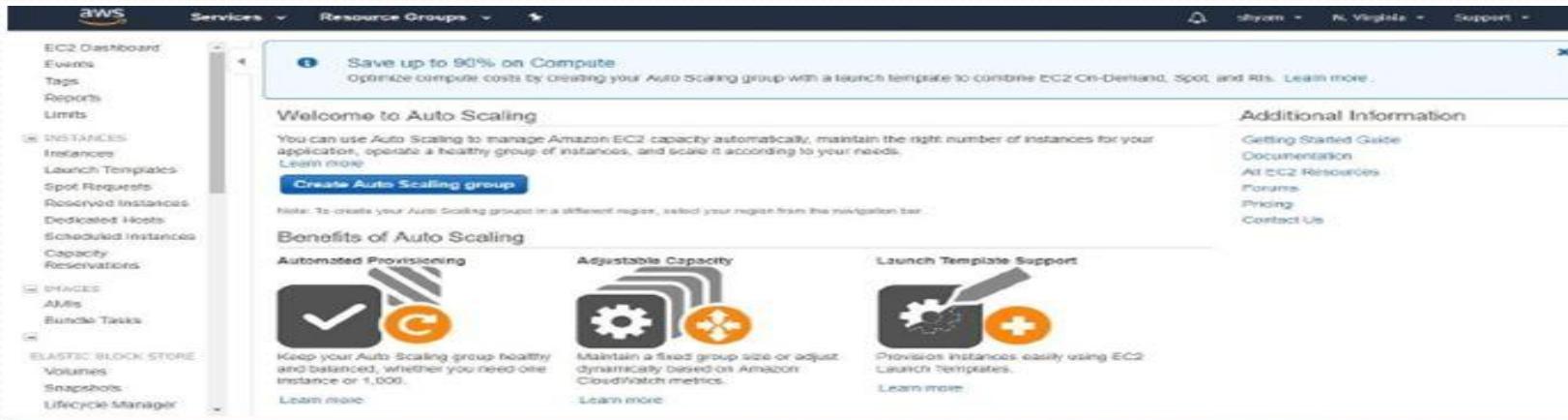
This option is used to create new template from existing template

- Click on copy to launch template
- Enter new template name
- Click on submit



Create Auto scaling Group

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



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

Services - Resource Groups -

1. Configure Auto Scaling group details 2. Configure scaling policies 3. Configure Notifications 4. Configure Tags 5. Review

Create Auto Scaling Group

Cancel and Exit

Group name: myauto

Launch Configuration: mylomp

Group size: Start with 1 instances

Network: vpc-97ab3fed (172.31.0.0/16) (default)

Create new VPC

Subnet:

- subnet-3df4b3a3 (172.31.0.0/20) | Default in us-east-1c
- subnet-1cf1897b (172.31.0.0/20) | Default in us-east-1b

Create new subnet

Each instance in this Auto Scaling group will be assigned a public IP address.

Advanced Details

Configure Scaling Policies:

Cancel Next: 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
- Click on Next Configure Notifications

AWS Services Resource Groups

1. Configure Auto Scaling group details 2. Configure scaling policies 3. Configure Notifications 4. Configure Tags 5. Review

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

AWS Services Resource Groups

1. Configure Auto Scaling group details 2. Configure scaling policies 3. Configure notifications 4. Configure Tags 5. Review

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: 300 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

The screenshot shows the AWS Auto Scaling 'Create Auto Scaling Group' wizard, specifically step 3: Configure Notifications. The top navigation bar includes 'Services' (dropdown), 'Resource Groups' (dropdown), and tabs for '1. Configure Auto Scaling group details', '2. Configure scaling policies', '3. Configure Notifications' (highlighted in yellow), '4. Configure Tags', and '5. Review'. The main content area is titled 'Create Auto Scaling Group' and contains the following fields:

- Send a notification to:**
- With these recipients:**
- Whenever instances:** launch
 terminate
 fail to launch
 fail to terminate

A large 'Add notification' button is located at the bottom left. At the bottom right, there are buttons for 'Cancel', 'Previous', 'Review' (highlighted in blue), and 'Next: Configure Tags'.

The screenshot shows the AWS Management Console with the EC2 service selected. In the left sidebar, under 'INSTANCES', 'Launch Templates' is expanded, showing 'mytemp'. The main content area displays the 'Create Auto Scaling group' interface. A table lists one Auto Scaling Group: 'myauto' (Launch Configuration: 'mytemp', Instances: 1, Desired: 1, Min: 1, Max: 1, Availability Zones: 'us-east-1a, us-east-1c', DefaultCooldown: 300, HealthCheckGracePeriod: 300). Below the table, the 'Auto Scaling Group: myauto' details page is shown with tabs for 'Details', 'Activity History', 'Scaling Policies', 'Instances', 'Monitoring', 'Notifications', 'Tags', 'Scheduled Actions', and 'Lifecycle Hooks'. Under 'Launch Configuration', it shows 'mytemp'. Under 'Availability Zone(s)', it shows 'us-east-1a, us-east-1c'. Under 'Subnet(s)', it shows 'subnet-8cf4b0a3, subnet-1cf18c'. Under 'Classic Load Balancers', there is a link to 'Attach load balancer...'. Under 'Target Groups', there is a link to 'Attach target group...'. Under 'Health check type', it says 'EC2'. Under 'Health check grace period', it says '300'. Under 'Edit', it says '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 Maximum 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

Launch Instances Using: Launch Template: mytemp

Launch Configuration: mytemp

Desired Capacity: 1

Min: 1

Max: 1

Availability Zones: us-east-1a, us-east-1c

Subnet(s):

- subnet-09ccb443 (172.31.16.0/20) | Default in us-east-1a
- subnet-0d4b3a3 (172.31.20.0/20) | Default in us-east-1c
- subnet-10a5ac4c (172.31.32.0/20) | Default in us-east-1a
- subnet-0a92a0e5 (172.31.48.0/20) | Default in us-east-1a
- subnet-1cf1097b (172.31.0.0/20) | Default in us-east-1b
- subnet-e3e30c8d (172.31.64.0/20) | Default in us-east-1a

Cancel Save

1 to 1 of 1 Auto Scaling Groups

Default Condition: 300

Health Check Grace: 300

Lifecycle Hooks:

- On-start-1a, us-east-1c
- subnet-0d4b3a3, subnet-1cf1097b

Delete Auto scaling group

Select auto scaling group and go to actions and click on delete

Create Auto Scaling group Actions Edit Delete

Name	Launch Configuration	Instances	Desired	Min	Max	Availability Zones	Default Condition	Health Check Grace
myauto	mytemp	1	1	1	1	us-east-1a, us-east-1c	300	300

Details Activity History Scaling Policies Instances Monitoring Notifications Tags Scheduled Actions Lifecycle Hooks Actions

Filter: Any Health Status Any Lifecycle State

Instance ID	Lifecycle	Launch Configuration / Template	Availability Zone	Health Status	Protected from
i-000c652ab31de8271	InService	mytemp	us-east-1c	Healthy	

Note: Until you are deleting Auto scaling group, Instances in EC2 will be Running State.

Best Practice: Delete the Auto scaling group, Once complete the task.

Elastic File System (EFS)

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.

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

The screenshot shows the AWS Elastic File System (EFS) Management console. At the top, there's a navigation bar with tabs for 'Services' and 'Resource Groups'. Below the navigation bar, there's a large central area with a red circular icon containing a stylized orange 'EFS' logo. The text 'Amazon Elastic File System (EFS)' is displayed prominently. Below the title, a subtitle reads 'Amazon EFS provides file storage for use with your EC2 instances.' A blue 'Create file system' button is centered below the subtitle. To the right of the button, there's a link to a 'Getting started guide'. At the bottom of this main section, there are three cards with icons and descriptions: 'Create' (with a folder icon), 'Access' (with a cloud and drive icon), and 'Manage' (with a person icon). Each card has a brief explanatory text below it.



Create

Create an Amazon EFS file system to store your files in the Amazon cloud. A file system grows and shrinks automatically with the files you put in, and you pay only for what you use.



Access

Upload files to and read files from your Amazon EFS file system by using the NFSv4 protocol. Any number of EC2 instances can work with your file system at the same time, and your instances can be in



Manage

You can easily administer your file system using the Amazon EFS console, CLI, and SDK.

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-917ab3cd (default)

Create mount targets:

Instances connect to a file system by using mount targets you create. We recommend creating a mount target in each of your VPC's Availability Zones so that EC2 instances across your VPC can access the file system.

Availability zone	Subnet	IP address	Security groups
us-east-1a	subnet-10ef8ac4 (default)	Automatic	sg-3121679 - default
us-east-1b	subnet-fcf10897b (default)	Automatic	sg-3121679 - default
us-east-1c	subnet-b04bd3a3 (default)	Automatic	sg-3121679 - default
us-east-1d	subnet-09bcbb443 (default)	Automatic	sg-3121679 - default
us-east-1e	subnet-a5e56c5d (default)	Automatic	sg-3121679 - default
us-east-1f	subnet-1a92e0a5 (default)	Automatic	sg-3121679 - default

Cancel **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

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	
Add New Key		

Enable lifecycle management

Automatically save up to 86% 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



Services

Resource Groups



shyam

N. Virginia

Support

Create file system

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-07ab0fed (default)	us-east-1a	subnet-10ef9ac4c (default)	Automatic	sg-3f2f6679 - default
	us-east-1b	subnet-1cf1ab97b (default)	Automatic	sg-3f2f6679 - default
	us-east-1c	subnet-6df4b2e03 (default)	Automatic	sg-3f2f6679 - default
	us-east-1d	subnet-09icb443 (default)	Automatic	sg-3f2f6679 - default
	us-east-1e	subnet-a3e06c1d (default)	Automatic	sg-3f2f6679 - default
	us-east-1f	subnet-aa92a0a5 (default)	Automatic	sg-3f2f6679 - default

Optional settings

Tags No tags added

The screenshot shows the AWS File Systems console. In the top navigation bar, 'File systems' is selected under 'AWS Storage'. A success message box is displayed, stating: 'Success! You have created a file system. You can mount your file system from an EC2 instance with an NFSv4.1 (CIFS) mount. 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 the message, there are 'Create file system' and 'Actions' buttons. A table lists the created file system details:

Name	File system ID	Metered size	Number of mount targets	Creation date
*	f-17e773f8	0.0 KB	0	06/07/2019, 17:52:46 UTC

Below the table, 'Other details' and 'Tags' sections are shown. The 'Other details' section includes fields: Owner ID (81492768004), File system state (Available), Performance mode (General Purpose), Throughput mode (Bursting), Encrypted (No), and Lifecycle policy (None). The 'Tags' section shows 'no tags added' and a 'Manage tags' button.

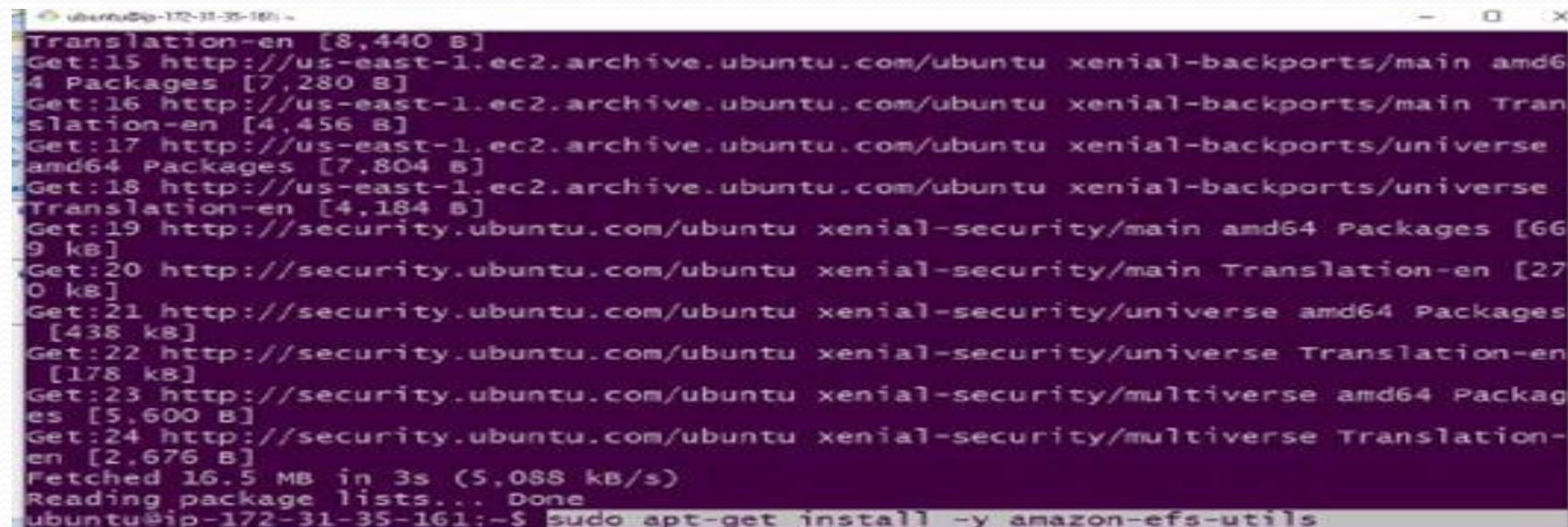
To connect to your Amazon EC2 instance and mount the Amazon EFS file system
Connect to EC2 instance using git bash or putty with SSH protocol like below.

The terminal window shows the output of a package manager (likely apt-get) as it fetches and lists packages. The output includes:

```
Translation-en [8,440 B]
Get:15 http://us-east-1.ec2.archive.ubuntu.com/ubuntu xenial-backports/main amd64 Packages [7,280 B]
Get:16 http://us-east-1.ec2.archive.ubuntu.com/ubuntu xenial-backports/main Translation-en [4,456 B]
Get:17 http://us-east-1.ec2.archive.ubuntu.com/ubuntu xenial-backports/universe amd64 Packages [7,804 B]
Get:18 http://us-east-1.ec2.archive.ubuntu.com/ubuntu xenial-backports/universe Translation-en [4,184 B]
Get:19 http://security.ubuntu.com/ubuntu xenial-security/main amd64 Packages [669 kB]
Get:20 http://security.ubuntu.com/ubuntu xenial-security/main Translation-en [270 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 Packages [5,600 B]
Get:24 http://security.ubuntu.com/ubuntu xenial-security/multiverse Translation-en [2,676 B]
Fetched 16.5 MB in 3s (5,088 kB/s)
Reading package lists... Done
```

Install amazon-efs-utils with below command:

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

A screenshot of a terminal window titled "ubuntu@ip-172-31-35-161:~". The window shows the output of an "apt-get update" command followed by an "apt-get install -y amazon-efs-utils" command. The output includes various package download details and a final message indicating the installation was successful.

```
Translation-en [8,440 B]
Get:15 http://us-east-1.ec2.archive.ubuntu.com/ubuntu xenial-backports/main amd64 Packages [7,280 B]
Get:16 http://us-east-1.ec2.archive.ubuntu.com/ubuntu xenial-backports/main Translation-en [4,456 B]
Get:17 http://us-east-1.ec2.archive.ubuntu.com/ubuntu xenial-backports/universe amd64 Packages [7,804 B]
Get:18 http://us-east-1.ec2.archive.ubuntu.com/ubuntu xenial-backports/universe Translation-en [4,184 B]
Get:19 http://security.ubuntu.com/ubuntu xenial-security/main amd64 Packages [669 kB]
Get:20 http://security.ubuntu.com/ubuntu xenial-security/main Translation-en [270 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 Packages [5,600 B]
Get:24 http://security.ubuntu.com/ubuntu xenial-security/multiverse Translation-en [2,676 B]
Fetched 16.5 MB in 3s (5,088 kB/s)
Reading package lists... Done
ubuntu@ip-172-31-35-161:~$ sudo apt-get install -y amazon-efs-utils
```

Create one directory using below command

Mkdir efsmount

```
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 EFS to that directory using below command

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

Copy the efs-filesystem-id from EFS

```
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
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 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:/ /mnt/efs/
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
```

Go to that mount directory using below command

Cd efsmount

```
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
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 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
```

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

```
09 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
ubuntu@ip-172-31-35-161:~/efsmount$ Connection reset by 3.86.14.30 port 22

nagesh@nagesh-PC: MINGW64 ~/downloads (master)
$ ssh -i "nkey.pem" ubuntu@ec2-3-86-14-30.compute-1.amazonaws.com
Welcome to Ubuntu 16.04.6 LTS (GNU/Linux 4.4.0-1083-aws x86_64)

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

36 packages can be updated.
18 updates are security updates.

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

Last login: Fri Jun  7 18:07:26 2019 from 183.83.244.116
ubuntu@ip-172-31-35-161:~$ sudo umount efsmount fs-17e779f4
```

Delete File system:

Select file system and go
to actions and click delete file system option

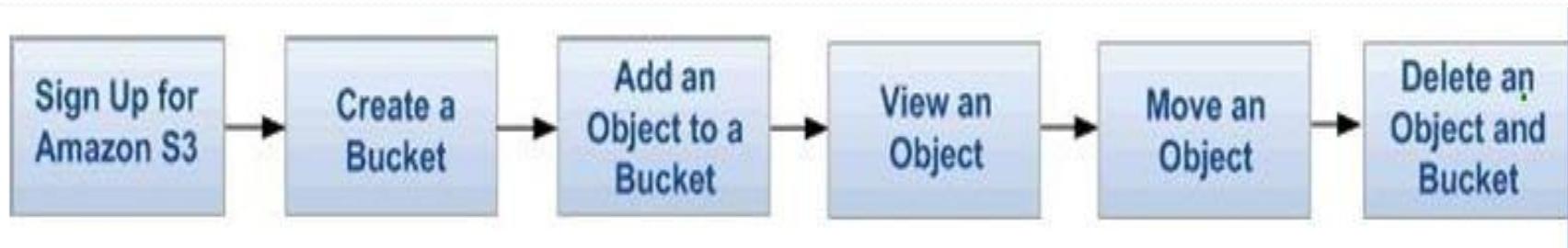
The screenshot shows the AWS File Systems console. In the top navigation bar, 'File systems' is selected under 'Services'. The main area displays a table of file systems. A context menu is open over the first row, listing options: 'Manage file system access', 'Manage tags', 'Manage throughput mode', and 'Delete file system'. The 'Delete file system' option is highlighted with a black box.

Enter file system id & click delete file system

The screenshot shows the AWS File Systems console with a confirmation dialog box in the foreground. The dialog title is 'Permanently delete file system'. It contains a warning message: 'This is a destructive action that cannot be undone.' Below this, it states: 'This action will permanently delete the file system. The file system's mount targets will also be deleted.' A text input field asks 'Confirm the deletion by entering the file system's ID: fs-17e77961' with the value 'fs-17e77961' entered. At the bottom right of the dialog are 'Cancel' and 'Delete File System' buttons.

S3 (Simple Storage Service)

Amazon Simple Storage Service (Amazon S3) is storage Service provided by AWS. We 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, CLI and SDK which is a simple and intuitive web interface.



Amazon S3 provides virtually limitless storage on the cloud. We 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.

Creating S3 bucket:

1. Bucket name must have a globally unique name.
2. Buckets are defined at region level.
3. No uppercase, Special Symbols and IP address.
4. Bucket name must start with lowercase letters and numbers.
5. Maximum object size is 5TB.
6. If uploading more than 5GB must use “**multi-part upload**”.

The screenshot shows the AWS S3 console interface. The top navigation bar includes the AWS logo, Services dropdown, Resource Groups dropdown, a user icon for 'shyam', a Global dropdown, and Support links. The main header reads "Amazon S3's newest storage class: S3 Intelligent-Tiering auto-tiers your data to deliver cost savings. Learn more >" and "Documentation". On the left, a sidebar menu has "Buckets" selected. The main content area is titled "S3 buckets" and features a search bar with "Search for buckets" and a dropdown for "All access types". Below the search bar are buttons for "+ Create bucket", "Edit public access settings", "Empty", and "Delete". To the right, it displays "3 Buckets" and "2 Regions". A table lists three buckets: "cb156" (Bucket and objects not public, Asia Pacific (Singapore), May 9, 2019 11:11:54 AM GMT+0530), "cf-templates-farnashifly2vpl-us-east-1" (Objects can be public, US East (N. Virginia), May 27, 2019 3:40:24 PM GMT+0530), and "myobj21" (Objects can be public, Asia Pacific (Singapore), May 9, 2019 11:06:46 AM GMT+0530). There is also a "Discover the console" link.

Bucket Name	Access	Region	Last Modified
cb156	Bucket and objects not public	Asia Pacific (Singapore)	May 9, 2019 11:11:54 AM GMT+0530
cf-templates-farnashifly2vpl-us-east-1	Objects can be public	US East (N. Virginia)	May 27, 2019 3:40:24 PM GMT+0530
myobj21	Objects can be public	Asia Pacific (Singapore)	May 9, 2019 11:06:46 AM GMT+0530

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

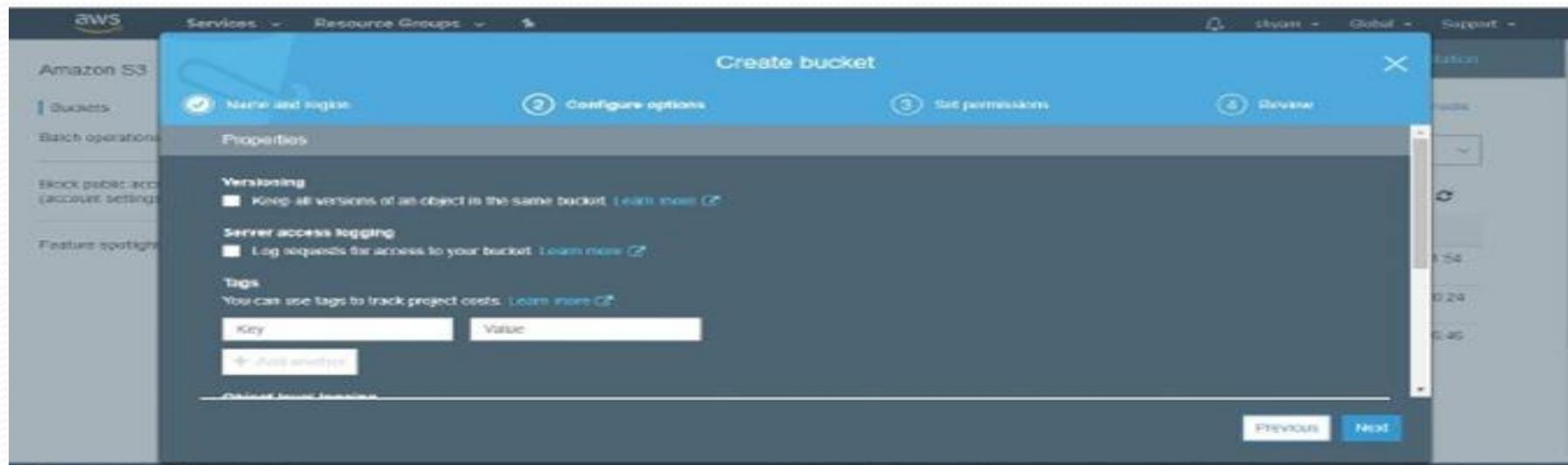
Default Encryption: enable this option if do you want encrypt objects stored in s3

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 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 permissions to this bucket by using bucket policy and Access control List

Click on next



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.

The screenshot shows the AWS S3 service dashboard. On the left, there's a sidebar with options like 'Amazon S3', 'Create bucket', 'Edit public access settings', 'Empty', and 'Delete'. The main area is titled 'S3 buckets' and contains a search bar and a dropdown for 'All access types'. It lists five buckets:

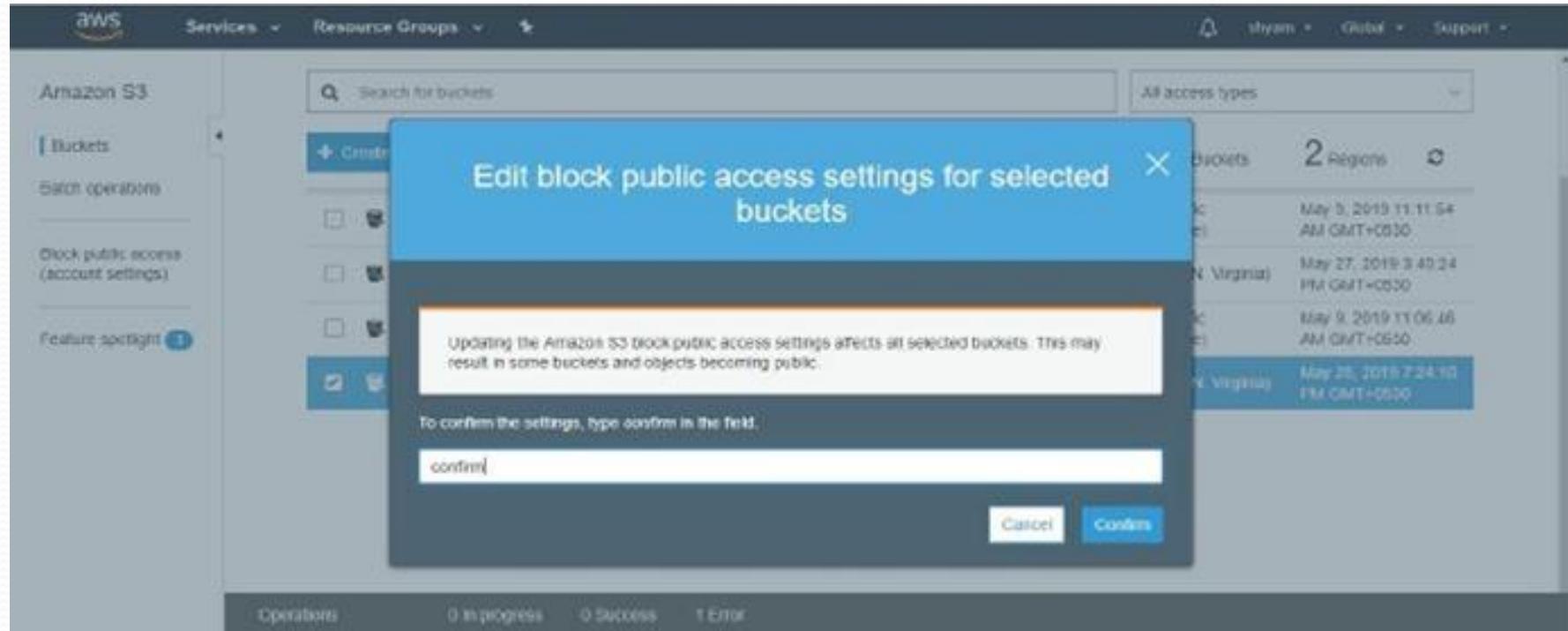
Bucket name	Access	Region	Date created
tb136	Bucket and objects not public	Asia Pacific (Singapore)	May 5, 2019 11:11:54 AM GMT+0530
cf-templates-1smstirby3vpk-us-east-1	Objects can be public	US East (N. Virginia)	May 27, 2019 3:40:24 PM GMT+0530
myco321	Objects can be public	Asia Pacific (Singapore)	May 5, 2019 11:05:45 AM GMT+0530
rstb3m	Bucket and objects not public	US East (N. Virginia)	May 26, 2019 7:54:30 PM GMT+0530

Advanced Configuration: Give default settings and click on next

The screenshot shows the 'Configure health check' step in the AWS Route 53 console. The 'Name' field is set to 'example name'. Under 'What to monitor', the 'Endpoint' option is selected. In the 'Monitor an endpoint' section, the 'Specify endpoint by' dropdown is set to 'IP address', and the IP address is '172.31.93.192'. The protocol is 'HTTP', host name is 'apache2', port is '80', and the path is '/index'. A note at the bottom states: 'Amazon SNS notifications will be sent to the specified SNS topic when the status of the health check changes to unhealthy.' There are 'Next Step' and 'Show all' buttons at the bottom.

Get notified when health check fails : If you want Cloud Watch 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.

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

The screenshot shows the AWS S3 Management console for a bucket named "mytbyn". The "Management" tab is selected. At the top, there are tabs for "Overview", "Properties", "Permissions", and "Management". Below the tabs are buttons for "Upload", "Create folder" (which is highlighted in blue), "Download", and "Actions". The main content area displays the message "This bucket is empty. Upload new objects to get started." with three corresponding icons: a bucket, users, and databases.

This bucket is empty. Upload new objects to get started.

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

Enter folder name

Encryption Settings: Select None (Use bucket settings)

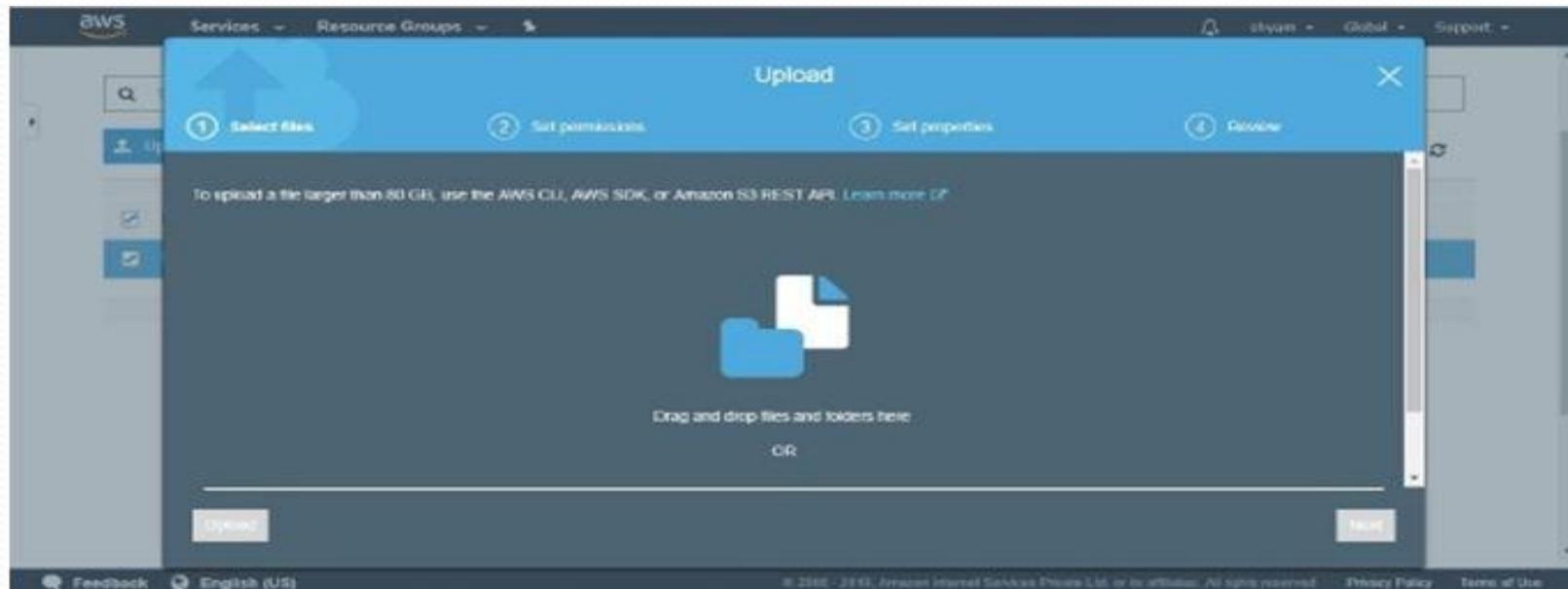
Note: If you want to apply AES-256 or AWS-KMS you can select particular option

Click on save

The screenshot shows the AWS S3 console interface. At the top, there is a navigation bar with the AWS logo, 'Services' dropdown, 'Resource Groups' dropdown, and other global navigation links like 'Stream', 'Global', and 'Support'. Below the navigation bar is a search bar with placeholder text 'Type a prefix and press Enter to search. Press ESC to clear.' and several action buttons: 'Upload' (blue), 'Create folder' (light blue), 'Download', and 'Actions' (dropdown). To the right of these buttons, it shows the region 'US East (N. Virginia)' with a dropdown arrow. The main area is a table for managing objects. The columns are 'Name' (with a checkbox), 'Last modified', 'Size', and 'Storage class'. A row is currently selected, highlighted with a blue border, and contains the folder name 'Intelig'. Below the table, there is a note: 'When you create a folder, S3 console creates an object with the above name appended by suffix "/" and that object is displayed as a folder in the S3 console. Choose the encryption setting for the object:'. Three radio button options are listed: 'None (Use bucket settings)' (selected), 'AES-256' (disabled), and 'AWS-KMS' (disabled). Below each option is a small explanatory text: 'Use Server-Side Encryption with Amazon S3-Managed Keys (SSE-S3)' for AES-256 and 'Use Server-Side Encryption with AWS KMS-Managed Keys (SSE-KMS)' for AWS-KMS. At the bottom of the form are two buttons: 'Save' (blue) and 'Cancel'.

Upload Objects

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



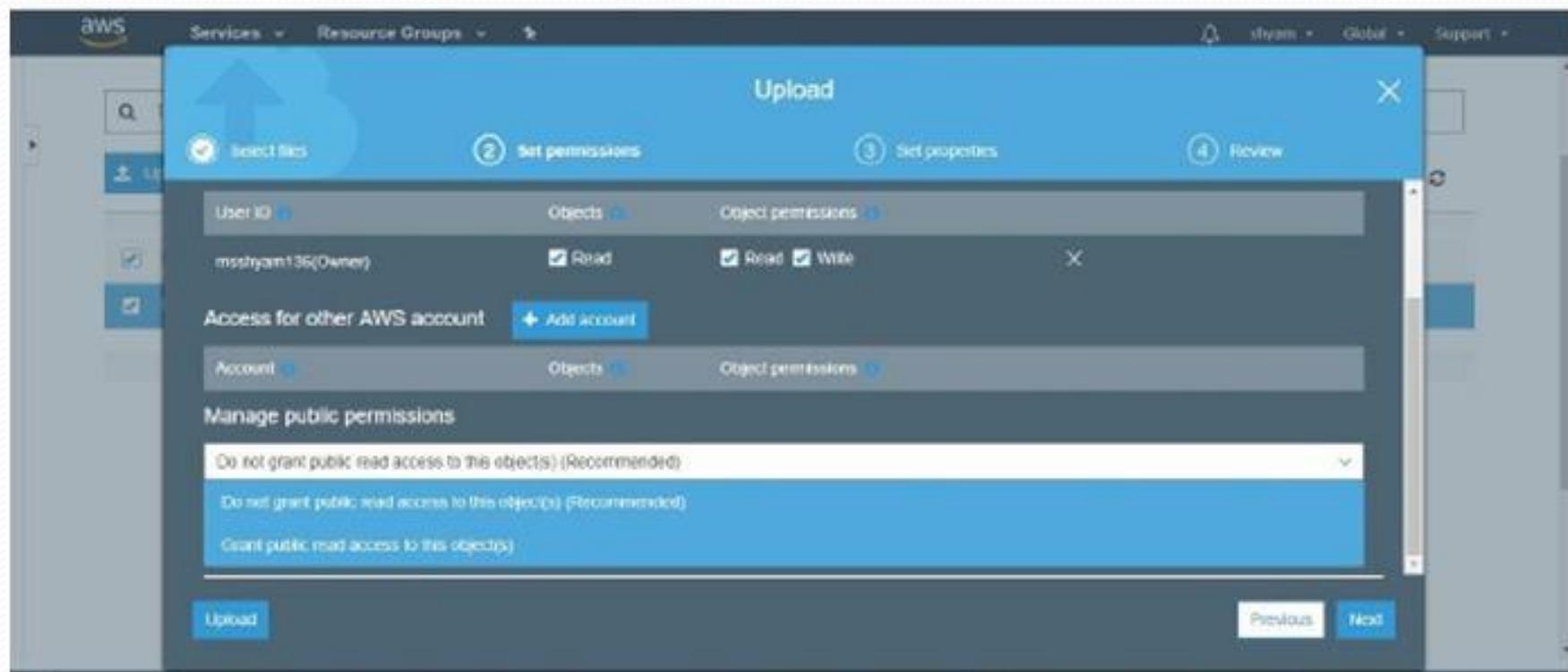
Set permissions:

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 any one of the below storage class and click on next

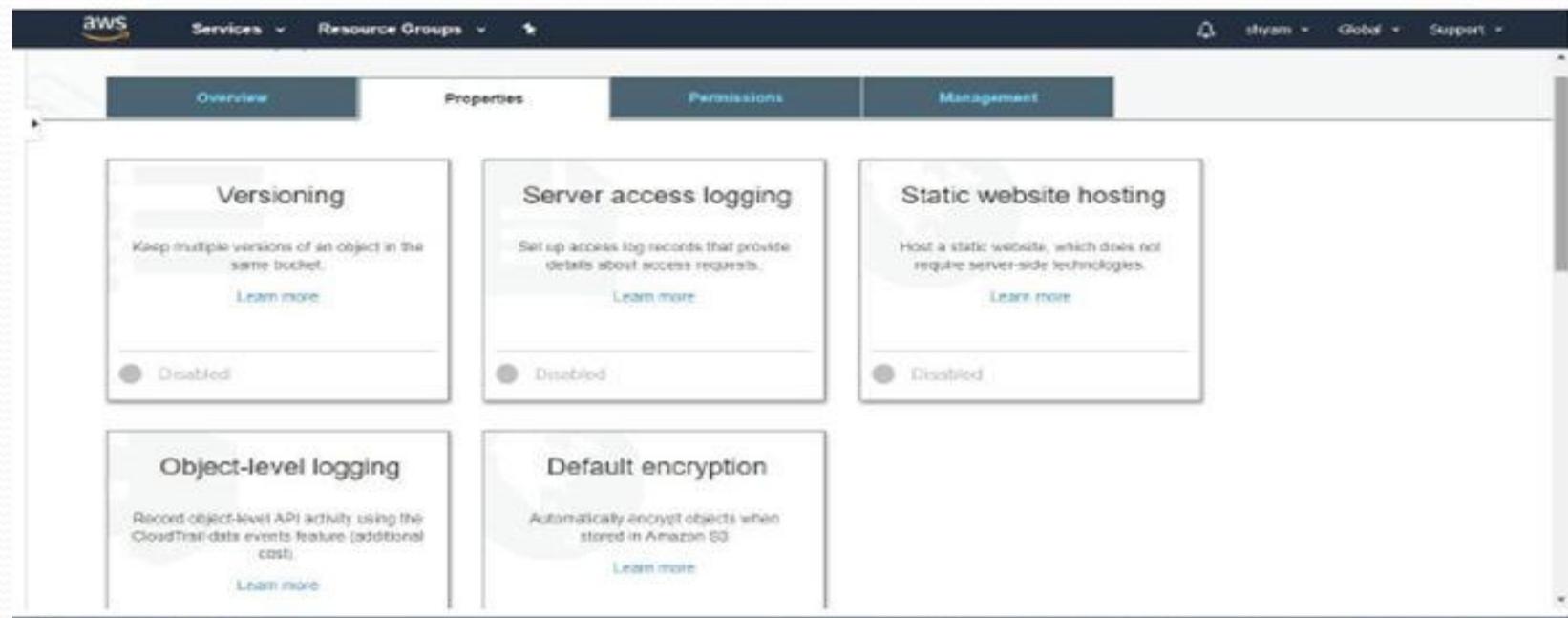
The screenshot shows the 'Set properties' step of an AWS S3 upload wizard. The top navigation bar includes 'Select files' (with a checkmark), 'Set permissions' (with a checkmark), 'Set properties' (selected, indicated by a blue background and a checkmark), and 'Review'. The main content area displays a table comparing six storage classes:

Storage class	Designed for	Availability Zones	Min storage duration	Min billable object size	Monitoring and automation fees	Retrieval fees
Standard	Frequently accessed data	≥ 3	-	-	-	-
Intelligent-Tiering	Long-lived data with changing or unknown access patterns	≥ 3	30 days	-	Per-object fees apply	-
Standard-IA	Long-lived, infrequently accessed data	≥ 3	30 days	128KB	-	Per-GB fees apply
One Zone-IA	Long-lived, infrequently accessed, non-critical data	≥ 1	30 days	128KB	-	Per-GB fees apply
Glacier	Archive data with retrieval times ranging from minutes to hours	≥ 3	90 days	-	-	Per-GB fees apply

At the bottom are 'Upload' and 'Next' buttons.

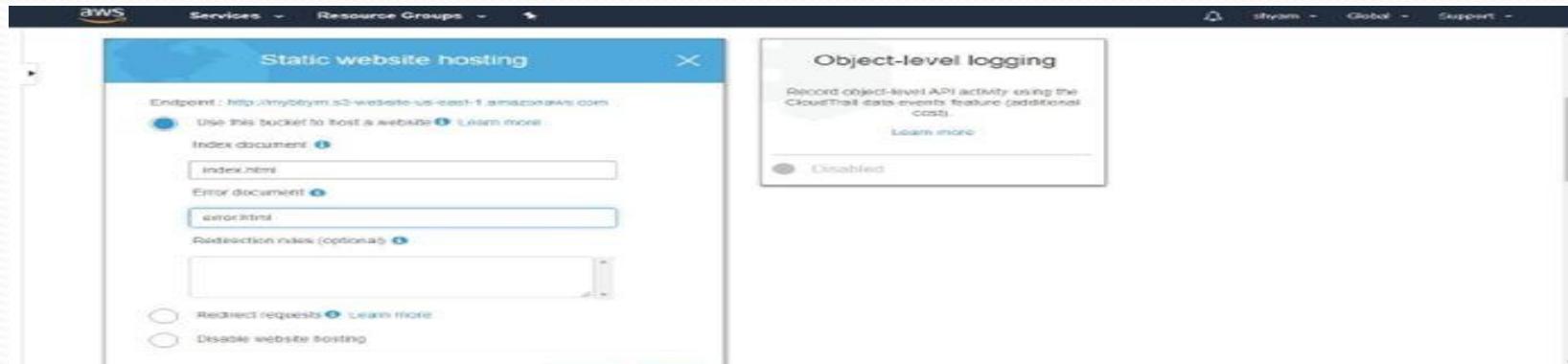
Changing bucket properties:

Click on bucket name 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

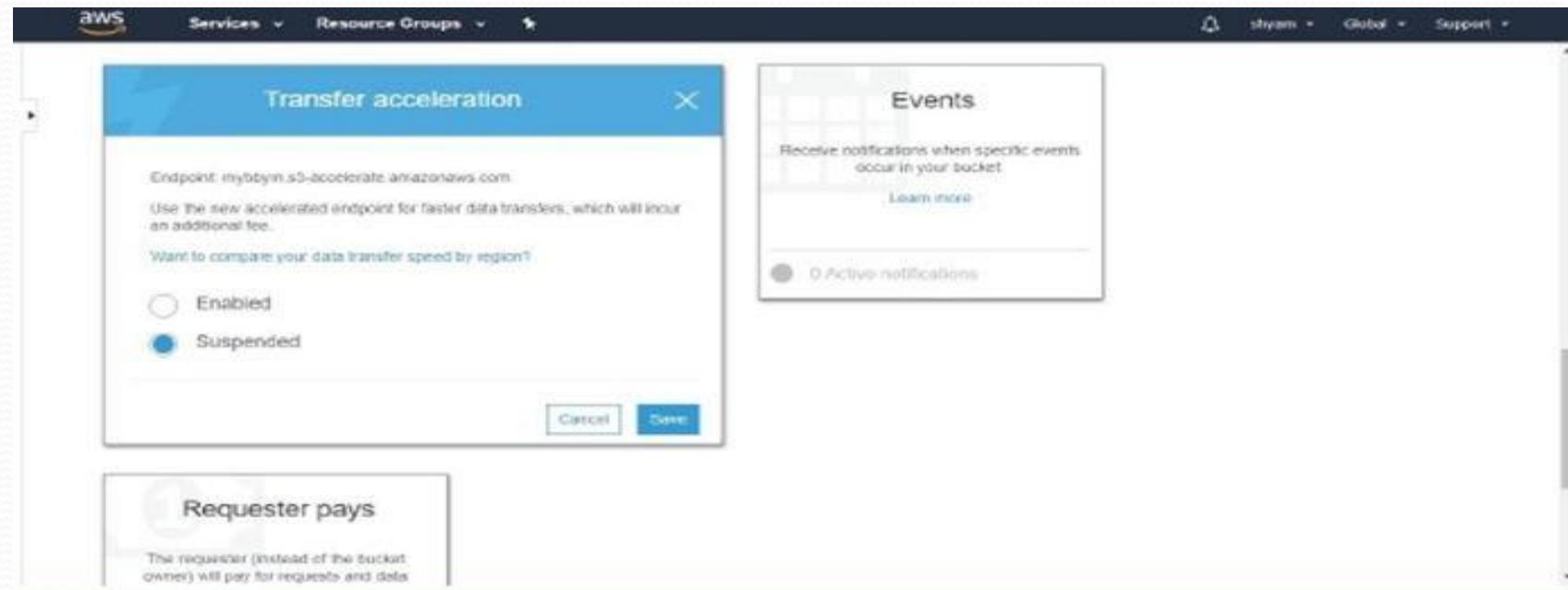
- Enable Static website hosting : this option is used hosting your application from bucket.
- Enable Versioning: click on enable versioning do you want to make version controlling
- There are two options one is use this bucket to host a website and another is redirect requests to another host
- Here click on first option
- Index Document: enter file name which file do you want to make as index starting page (ex: index.html)
- Error Document: enter file name for error redirection (ex: error.html)



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 want make notification that is PUT, POST, COPY.etc

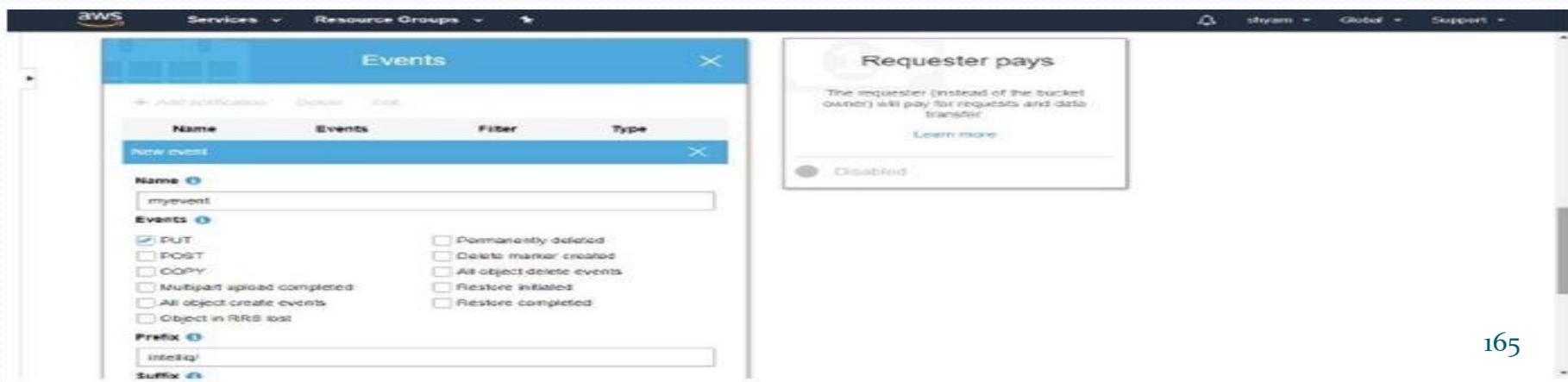
Prefix: it is the optional, notifications to object with keys starting with matching characters (ex: sourcefuse...: make the notifications for objects which object name is starting with sourcefuse)

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

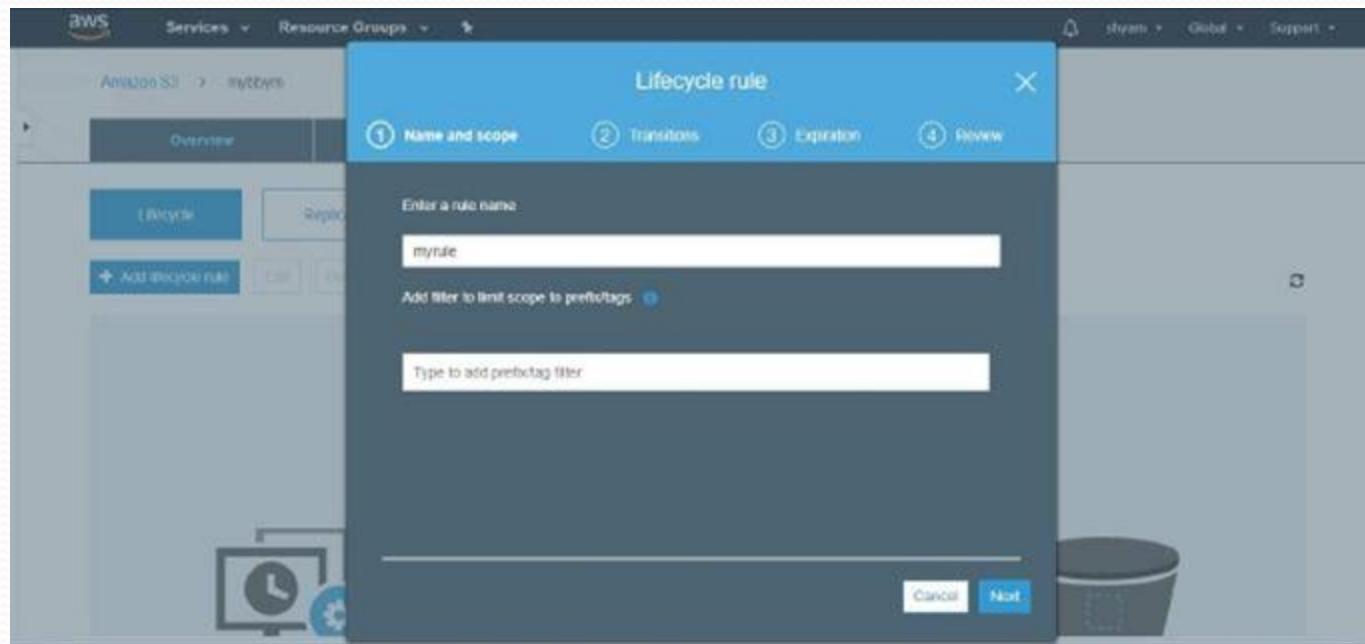


Bucket Policy: bucket policy is used to manage advanced permissions on Amazon S3 Resources.

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.

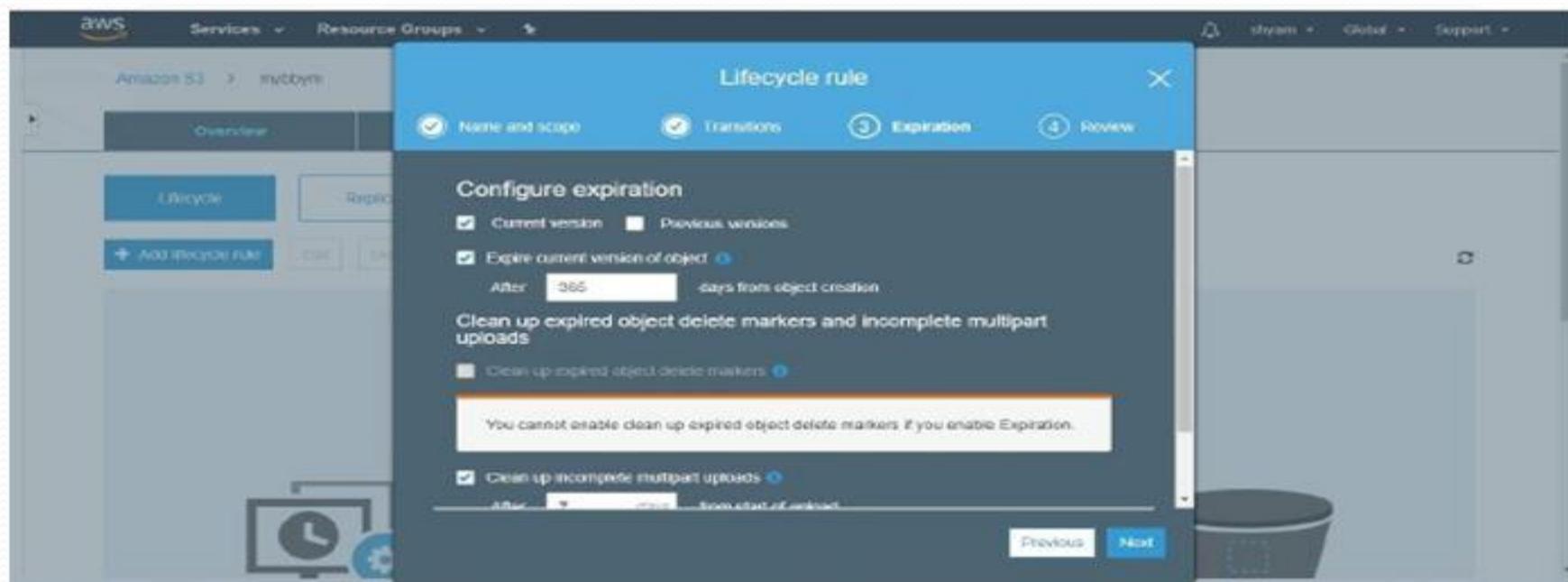
Click on Current version and click on next

Configure expiration

Enable current version

Expire current version of object and enter number of days for expiration

Click on next



Download objects from S3 bucket

Open the bucket in s3 and select object which object do you want download

- Click on download option

The screenshot shows the AWS S3 console interface. At the top, there's a navigation bar with 'Services', 'Resource Groups', and other account-related links. Below the navigation bar, the path 'Amazon S3 > mybbiny' is shown. The main area has tabs for 'Overview', 'Properties', 'Permissions', and 'Management'. A search bar at the top of the list area contains the placeholder 'Type a prefix and press Enter to search. Press Esc to clear.' Below the search bar are buttons for 'Upload', 'Create folder', 'Download' (which is highlighted in blue), 'Actions', 'Versions', 'Hide', and 'Show'. The list of objects in the bucket includes 'intelliq' and 'ctemp1.yml'. The 'ctemp1.yml' object is selected, as indicated by a blue border around its row. A modal dialog box is open over the list, titled 'ctemp1.yml'. It contains three buttons: 'Download' (highlighted in blue), 'Copy path', and 'Select from'. The 'Download' button is currently active. The dialog also shows the 'Latest version' of the object. The 'Overview' section displays details such as Key (ctemp1.yml), Size (702.9 B), and Last modified (May 20, 2019 8:46:50 PM GMT+0530). Other sections like Properties, Storage class, Encryption, Metadata, and Tags are also visible.

Delete Bucket:

Sign in 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

The screenshot shows the AWS S3 console interface. On the left, there's a sidebar with options like 'Buckets' (which is selected), 'Batch operations', 'Block public access (account settings)', and 'Feature spotlight'. The main area is titled 'S3 buckets' and contains a search bar and buttons for 'Create bucket', 'Edit public access settings', 'Empty', and 'Delete'. Below these are four bucket entries: 'cb106', 'cf-templates-1smehfrfy0vpk-us-east-1', 'mybb321', and 'mybbym'. The 'mybbym' entry is highlighted with a blue selection bar at the bottom. A modal dialog box is open over the interface, centered on the 'mybbym' bucket. The dialog has a title 'mybbym' and a 'Copy bucket ARN' button. It is divided into two sections: 'Properties' and 'Permissions'. The 'Properties' section lists various settings: Events (0 Active notifications), Versioning (Enabled), MFA delete (Disabled), Logging (Disabled), Static web hosting (Disabled), Tags (0 Tags), Requester pays (Disabled), Object lock (Disabled), and Transfer acceleration (Disabled). The 'Permissions' section shows the Owner as 'meethym126' and the Block public access setting as 'Disabled'. At the bottom right of the dialog is a large red 'Delete' button.

Cloud Trail

AWS Cloud Trail 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 Cloud Trail. Events include actions taken in the AWS Management Console, AWS Command Line Interface, and AWS SDKs and APIs.

Cloud Trail is enabled on your AWS account when you create it. When activity occurs in your AWS account, that activity is recorded in a Cloud Trail event. You can easily view recent events in the Cloud Trail console by going to Event history.

Cloud Trail Workflow:

[View event history for your AWS account](#)

You can view and search the last 90 days of events recorded by Cloud Trail 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 Cloud Trail events for your AWS account.

Create a trail

A trail enables Cloud Trail 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 from all regions in the AWS partition and delivers the log files to the S3 bucket that we 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 management events.
- 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 Cloud Trail delivered them.
- Share log files with other AWS accounts
- You can share log files between accounts.
- Aggregate logs from multiple accounts
- We can aggregate log files from multiple accounts to a single bucket.

Finding our Cloud Trail Log Files

Cloud Trail 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 Cloud Trail console)
- The (optional) prefix you specified when you created your trail
- The string "AWS Logs"
- The account number
- The string "Cloud Trail"
- 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

Deleting a Trail

You can delete trails with the Cloud Trail 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.

Cloud Watch

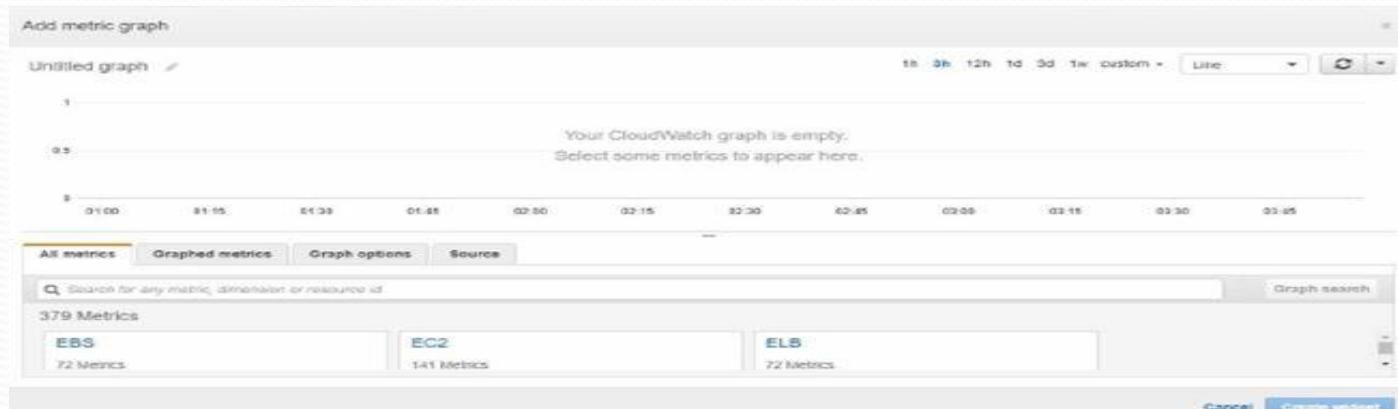
Amazon Cloud Watch monitors your Amazon Web Services (AWS) resources and the applications you run on AWS in real time. We can use Cloud Watch to collect and track metrics, which are variables you can measure for your resources and applications.

Use Case:

- Login into AWS console and choose cloud watch service
- In the left side panel we can see the cloud watch Events..(Dashboards, Logs...etc)

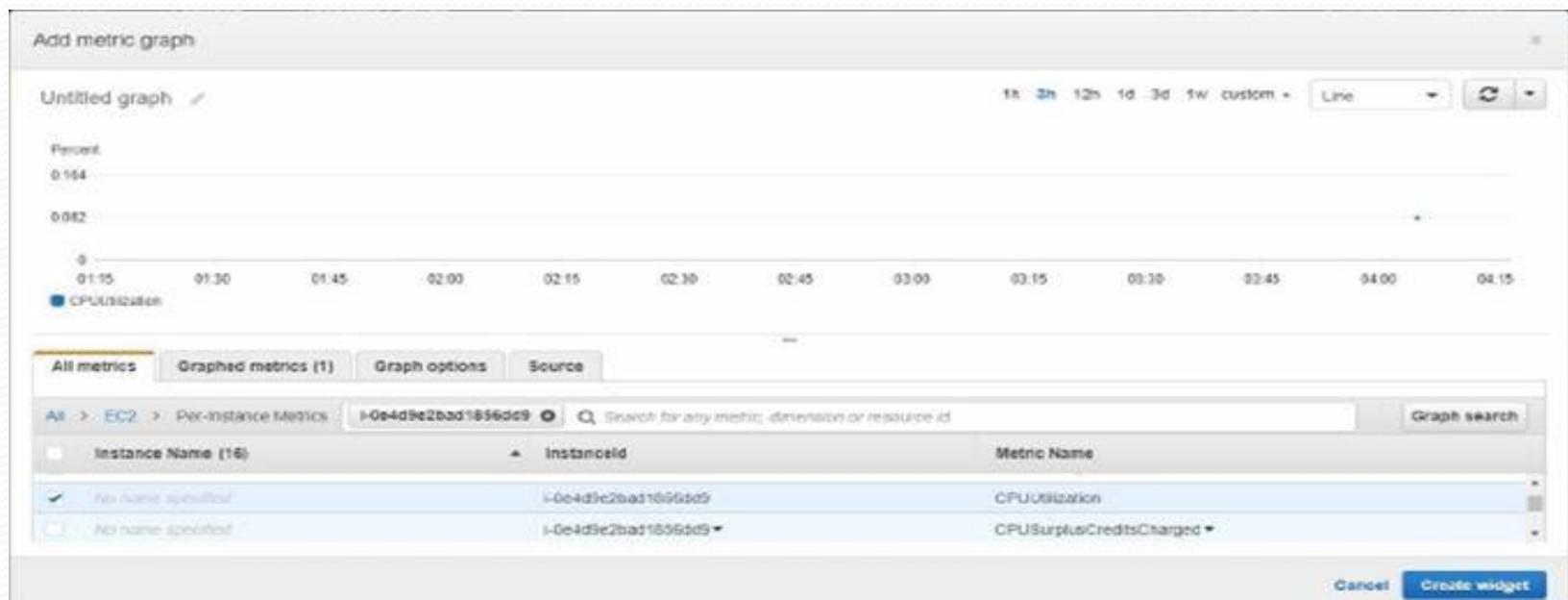
Dashboards

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 instance id with particular metric (ex.. CPU utilization, 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 in this 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 selected instance CPU utilization on dashboard



In above graph the horizontal line indicate time and vertical line indicates percentage of CPU utilization and click on Save Dashboard.

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: >= 50
 for: 1 ✓ out of 1 datapoints

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		
Whenever this alarm:	State is ALARM	
Send notification to:	Select a notification list	New list Enter list
<input type="button" value="+ Notification"/> <input type="button" value="+ AutoScaling Action"/> <input type="button" value="+ EC2 Action"/>		

[Cancel](#)

[Create Alarm](#)

- We create SNS Topic
- Go to SNS Service section and go to Topics Section Here Click on Create topic
- Give topic Name and click create topic

AWS Lambda > 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 SNS subscriptions, enter a display name. Only the first 10 characters are displayed in an SNS message. Limit: My Topic
 Maximum 100 characters, including hyphens (-) and underscores (_).

► **Encryption - optional**
Amazon SNS provides in-transit encryption by default. Enabling server-side encryption adds an extra layer of 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. Refer to the [Amazon SNS Access Policy Reference](#) for more information.

- Click on create subscription
- Select protocol as Email and enter your Email address in Endpoint section
- Then click on create subscription

The screenshot shows the Amazon SNS console. On the left, there's a sidebar with links: Dashboard, Topics, Subscriptions (which is selected and highlighted in orange), and Mobile (with sub-links Push notifications and Text messaging (SMS)). The main content area has a title bar: 'Amazon SNS > Topics > mytopic > Subscription: c8ad3c56-0f1e-40a8-b6f7-82e52adc2228'. Below this is a 'Subscription: c8ad3c56-0f1e-40a8-b6f7-82e52adc2228' card with tabs for 'Details' and 'Subscription filter policy'. The 'Details' tab shows the ARN (arn:aws:sns:us-east-1:993595193866:mytopic:c8ad3c56-0f1e-40a8-b6f7-82e52adc2228), Endpoint (knagesh35@gmail.com), Topic (mytopic), Status (Pending confirmation), and Protocol (EMAIL). There are 'Edit' and 'Delete' buttons at the top right of the card.

- Then go to cloud watch and refresh and select these above created SNS topic Name Notification section
- Click on create Alarm

The screenshot shows the AWS CloudWatch Metrics Alarm configuration page. At the top, it says "Whenever: CPUUtilization" with a threshold of "50". Below this, there's a note "for: 1 out of 1 datapoints". Under "Additional settings", "treat missing data as: missing". In the "Actions" section, there's a notification rule for "State is ALARM" sent to "mytopic". Buttons at the bottom include "Cancel" and "Create Alarm".

In above Case give the CPU utilization value in condition section is 50%. So State is Alarm since it's reach the condition then send notification action performed Open your SNS subscription mail and check it.

The screenshot shows a Gmail inbox with an incoming email from "aws-rep1@sns.amazonaws.com" titled "ALARM: "alarm-nagesh" in US East (N. Virginia)". The email body contains the following text:

You are receiving this email because your Amazon CloudWatch Alarm "alarm-nagesh" in the US East (N. Virginia) region has entered the ALARM state, because "Threshold Crossed: 1 datapoint [70.56566666666675 (29/04/19 00:29:00)] was greater than or equal to the threshold (50.0)" at "Monday 29 April, 2019 00:35:13 UTC".

View this alarm in the AWS Management Console:
<https://console.aws.amazon.com/cloudwatchmetrics/home?region=us-east-1&metricName=alarm-nagesh>

Alarm Details:

- Name: alarm-nagesh
- Description:
- State Change: OK -> ALARM
- Reason for State Change: Threshold Crossed: 1 datapoint [70.56566666666675 (29/04/19 00:29:00)] was greater than or equal to the threshold (50.0).
- Timestamp: Monday 29 April, 2019 00:35:13 UTC
- AWS Account: 933545193866

Threshold:

- The alarm is in the ALARM state when the metric is GreaterThanOrEqualToThreshold 50.0 for 300 seconds.

Logs:

By using logs in Cloud Watch we can monitor both system Logs and Application Logs

The screenshot shows the AWS CloudWatch service dashboard. On the left, there's a sidebar with navigation links: CloudWatch Dashboards, Alarms, Billing, Events, Rules, Event Buses, Log Insights (which is selected), Metrics, Favorites, and a link to Add a dashboard. The main content area shows a breadcrumb trail: CloudWatch > Log Groups > /var/log/syslog > 1.0f99fc99570c86c92. A modal window titled "Try CloudWatch Logs Insights" provides information about the feature, including a link to the AWS blog and documentation. Below the modal, there are buttons for "Expand all", "Row", "Text", and other filtering options. A date range selector shows "All" and "2019-04-29 (09:38:13)". A "Filter events" input field is present. The main table lists log events with columns for "Time (UTC +00:00)" and "Message". The "Time" column shows entries from April 29, 2019, at 09:30:36. The "Message" column contains detailed log entries, such as kernel initialization messages and errors related to syslogd and rsyslogd.

Time (UTC +00:00)	Message
2019-04-29 09:30:36	Apr 29 09:30:36 ip-172-31-80-58 rsyslogd: [origin software="rsyslogd" swVersion="5.16.0" x-pid="1071" x-info="http://www.rsyslog.com/] start
2019-04-29 09:30:36	Apr 29 09:30:36 ip-172-31-80-58 rsyslogd-2222: command 'logPermitInKernelFacility' is currently not permitted - did you already set it via :/etc/rsyslog.conf? [origin software="rsyslogd" swVersion="5.16.0" x-pid="1071" x-info="http://www.rsyslog.com/"]
2019-04-29 09:30:36	Apr 29 09:30:36 ip-172-31-80-58 rsyslogd: rsyslogd's groupid changed to 108
2019-04-29 09:30:36	Apr 29 09:30:36 ip-172-31-80-58 rsyslogd: rsyslogd's uid/gid changed to 104
2019-04-29 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.rsyslog.com/]
2019-04-29 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.rsyslog.com/]
2019-04-29 09:30:36	Apr 29 09:30:36 ip-172-31-80-58 kernel: [0:000000] Initializing cgroup subsys cpuset
2019-04-29 09:30:36	Apr 29 09:30:36 ip-172-31-80-58 kernel: [0:000000] Initializing cgroup subsys cpu
2019-04-29 09:30:36	Apr 29 09:30:36 ip-172-31-80-58 kernel: [0:000000] Initializing cgroup subsys cpufreq
2019-04-29 09:30:36	Apr 29 09:30:36 ip-172-31-80-58 kernel: [0:000000] Linux version 4.4.0-1073-aws (builder@tgw01-amd64-030) (gcc version 5.4.0 20160609 (L)
2019-04-29 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
2019-04-29 09:30:36	Apr 29 09:30:36 ip-172-31-80-58 kernel: [0:000000] KERNEL supported cores

Simple Email Service (SES)

SES

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 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 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

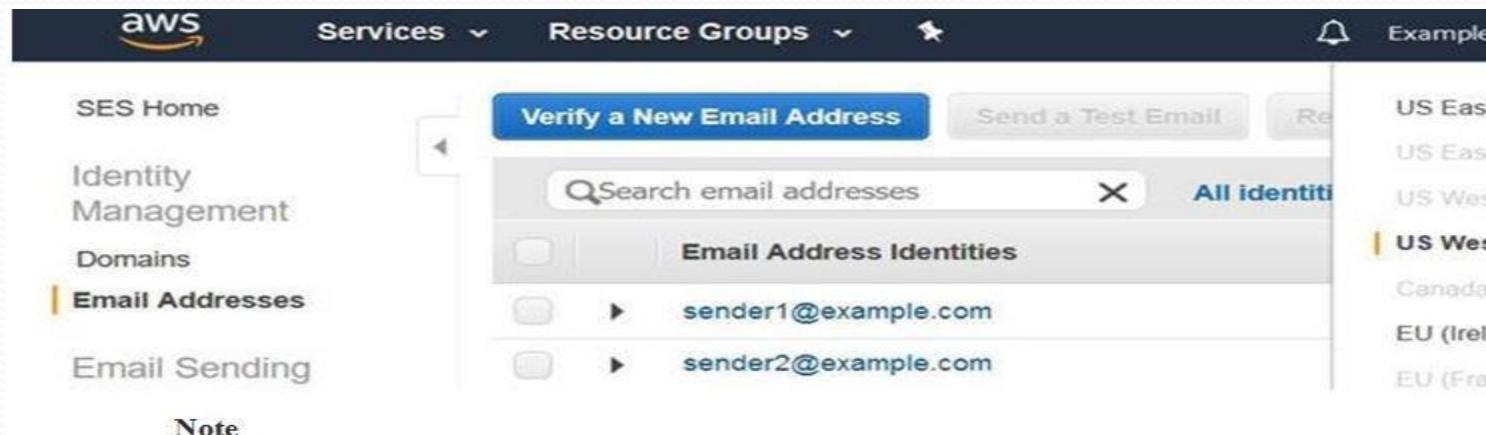
To be able to send emails to unverified email addresses and to raise the number of emails you can send per day and how fast you can send them, your account needs to be moved out of the sandbox. This process involves opening an SES Sending Limits Increase case in Support Center.

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 Region Name," where Region Name 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.

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

Note:

1. 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.
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:

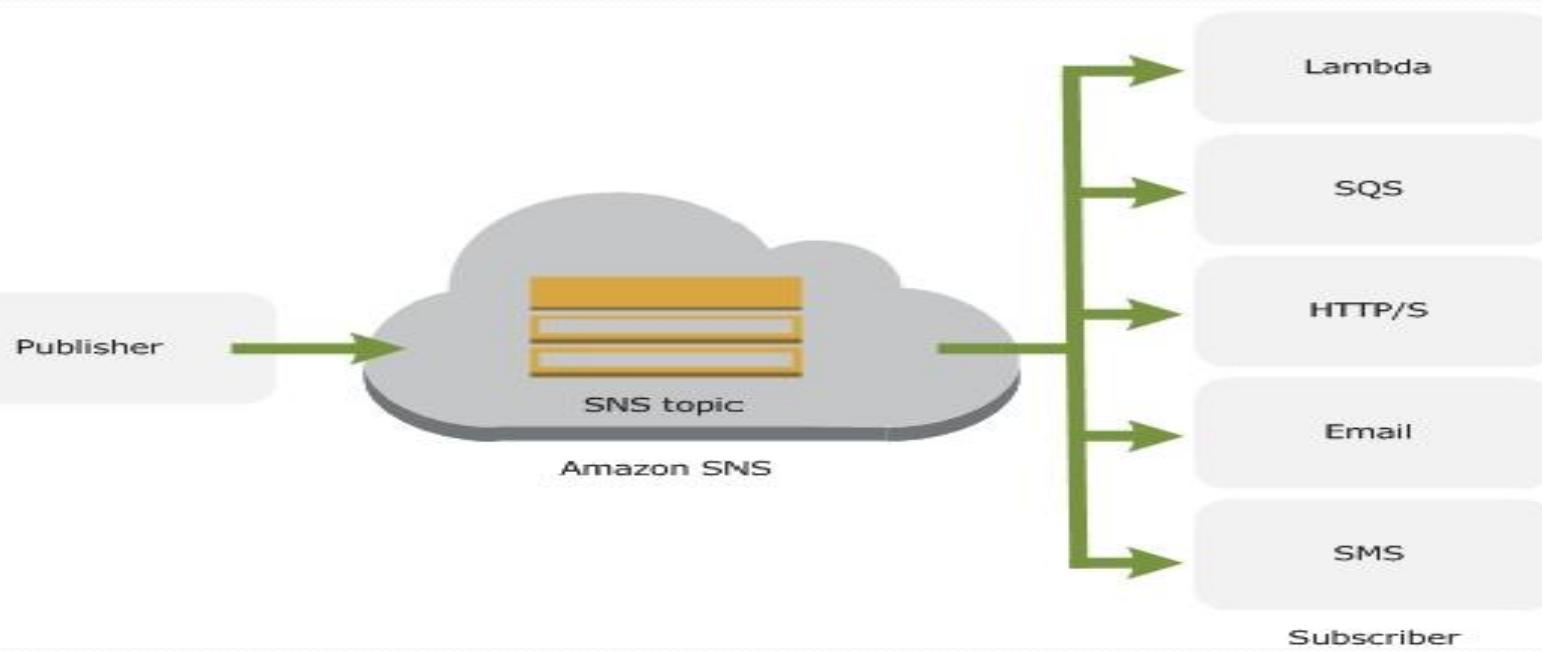
1. Amazon SES first looks at the IP address of the sender. Amazon SES allows the mail to pass this stage unless:
 - The IP address is in your block list.
 - The IP address is in the Amazon SES block list and not on your allow list.
2. Amazon SES examines your active receipt rule set to determine whether 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.

Simple Notification Service (SNS)

SNS

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.

SNS



Use Case:

Sign in into the AWS console and select the **SNS service**

The screenshot shows the AWS SNS 'Create Topic' configuration page. At the top, it says 'Topic type cannot be modified after topic is created'. Below this, there are two radio button options: 'FIFO (first-in, first-out)' and 'Standard'. The 'Standard' option is selected. Under 'FIFO (first-in, first-out)', there is a bulleted list: Strictly-preserved message ordering, Exactly-once message delivery, High throughput, up to 300 publishes/second, and Subscription protocols: SQS. Under 'Standard', there is another bulleted list: Best-effort message ordering, At-least once message delivery, Highest throughput in publishes/second, and Subscription protocols: SQS, Lambda, HTTP, SMS, email, mobile application endpoints. The 'Name' field contains 'mysampletopic' with a note: Maximum 256 characters. Can include alphanumeric characters, hyphens (-) and underscores (_). The 'Display name - optional' field also contains 'mysampletopic' with a note: To use this topic with SMS subscriptions, enter a display name. Only the first 10 characters are displayed in an SMS message. The 'Encryption - optional' section notes that Amazon SNS provides in-transit encryption by default and enabling server-side encryption adds at-rest encryption to your topic.

Type [Info](#)
Topic type cannot be modified after topic is created

FIFO (first-in, first-out)

- Strictly-preserved message ordering
- Exactly-once message delivery
- High throughput, up to 300 publishes/second
- Subscription protocols: SQS

Standard

- Best-effort message ordering
- At-least once message delivery
- Highest throughput in publishes/second
- Subscription protocols: SQS, Lambda, HTTP, SMS, email, mobile application endpoints

Name

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](#)

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.

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

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 delivery the Failure message by using HTTP protocol. It is the optional

Delivery status logging – optional: it is used to monitor the SNS Service with Cloud Watch. It is the optional

Click on create topic

aws Services Resource Groups

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

Minimum: 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 handles 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](#)

Amazon SNS > Topics > mysampletopic

mysampletopic

[Edit](#)

[Delete](#)

[Publish message](#)

Details

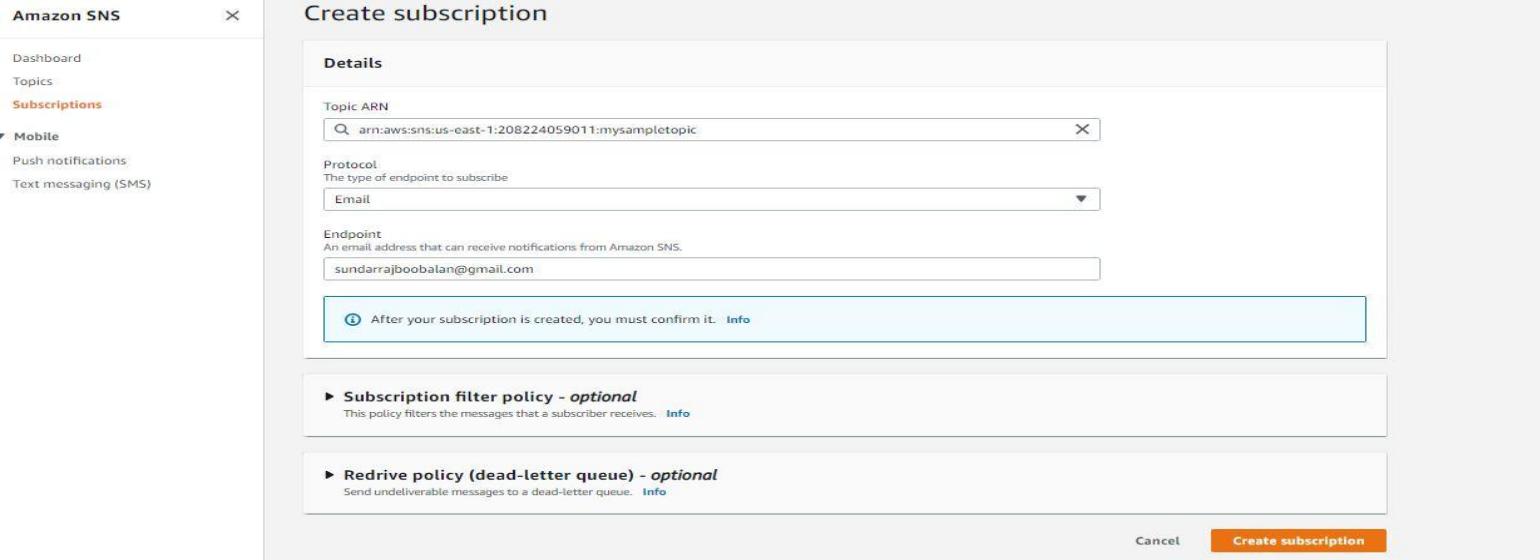
Name
mysampletopic

ARN
arn:aws:sns:us-east-1:208224059011:mysampletopic

Type
Standard

Display name
mysampletopic

Topic owner
208224059011



The screenshot shows the 'Create subscription' page in the Amazon SNS console. The left sidebar includes links for Dashboard, Topics, Subscriptions (selected), Mobile, Push notifications, and Text messaging (SMS). The main area has a 'Details' section with fields for Topic ARN (arn:aws:sns:us-east-1:208224059011:mysampletopic), Protocol (Email selected), and Endpoint (sundarrajboobalan@gmail.com). A note says 'After your subscription is created, you must confirm it.' Below are sections for 'Subscription filter policy - optional' and 'Redrive policy (dead-letter queue) - optional'. At the bottom are 'Cancel' and 'Create subscription' buttons.

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 ,Select Email

•**Endpoint:** Enter your email address

•Then click on create subscription

Amazon SNS

Important changes for sending text messages (SMS) to US destinations
Effective April 1, 2021, US telecom providers no longer support person-to-person (P2P) long codes for sending SMS messages to US destinations. To continue to send SMS messages to US destinations, register and use a valid origination ID. [Learn more](#)

Subscription to mysampletopic created successfully.
The ARN of the subscription is arn:aws:sns:us-east-1:208224059011:mysampletopic:ed31b94b-8aba-4626-92c2-3932b83e6b82.

Amazon SNS > Topics > mysampletopic > Subscription: ed31b94b-8aba-4626-92c2-3932b83e6b82

Subscription: ed31b94b-8aba-4626-92c2-3932b83e6b82

[Edit](#) [Delete](#)

Details	
ARN	arn:aws:sns:us-east-1:208224059011:mysampletopic:ed31b94b-8aba-4626-92c2-3932b83e6b82
Status	Pending confirmation
Endpoint	sundarrajboobalan@gmail.com
Protocol	EMAIL
Topic	mysampletopic

Amazon SNS

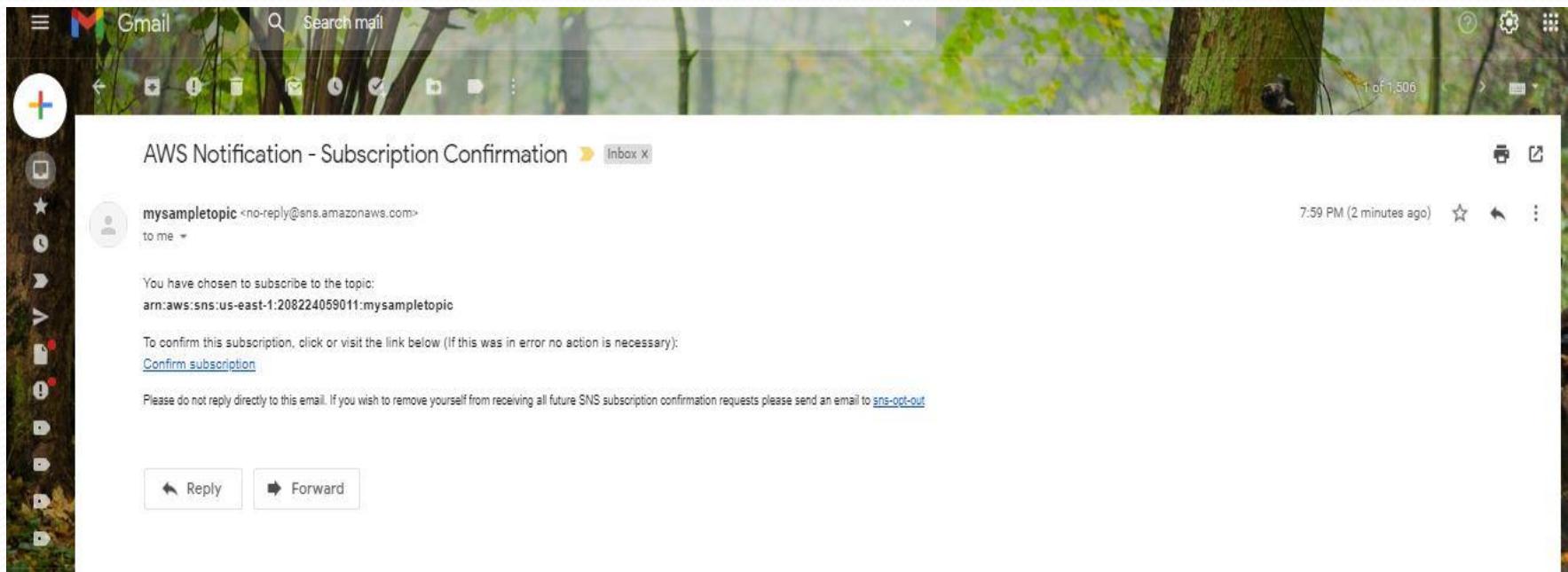
Amazon SNS > Subscriptions

Subscriptions (4)

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

ID	Endpoint	Status	Protocol	Topic
94d056e2-bda5-40ad-8ab4-6dc587cf3f32	sundarrajboobalan@gmail.com	Confirmed	EMAIL	MySNStopic
Pending confirmation	sundarrajboobalan@mailinator.com	Pending confirmation	EMAIL	MySNStopic
1b716853-68de-495a-a9aa-faf25dc818a9	sundarraj.b@sourcefuse.com	Confirmed	EMAIL	mysampletopic
Pending confirmation	sundarrajboobalan@gmail.com	Pending confirmation	EMAIL	mysampletopic

Open our Email and Confirm SNS Subscription



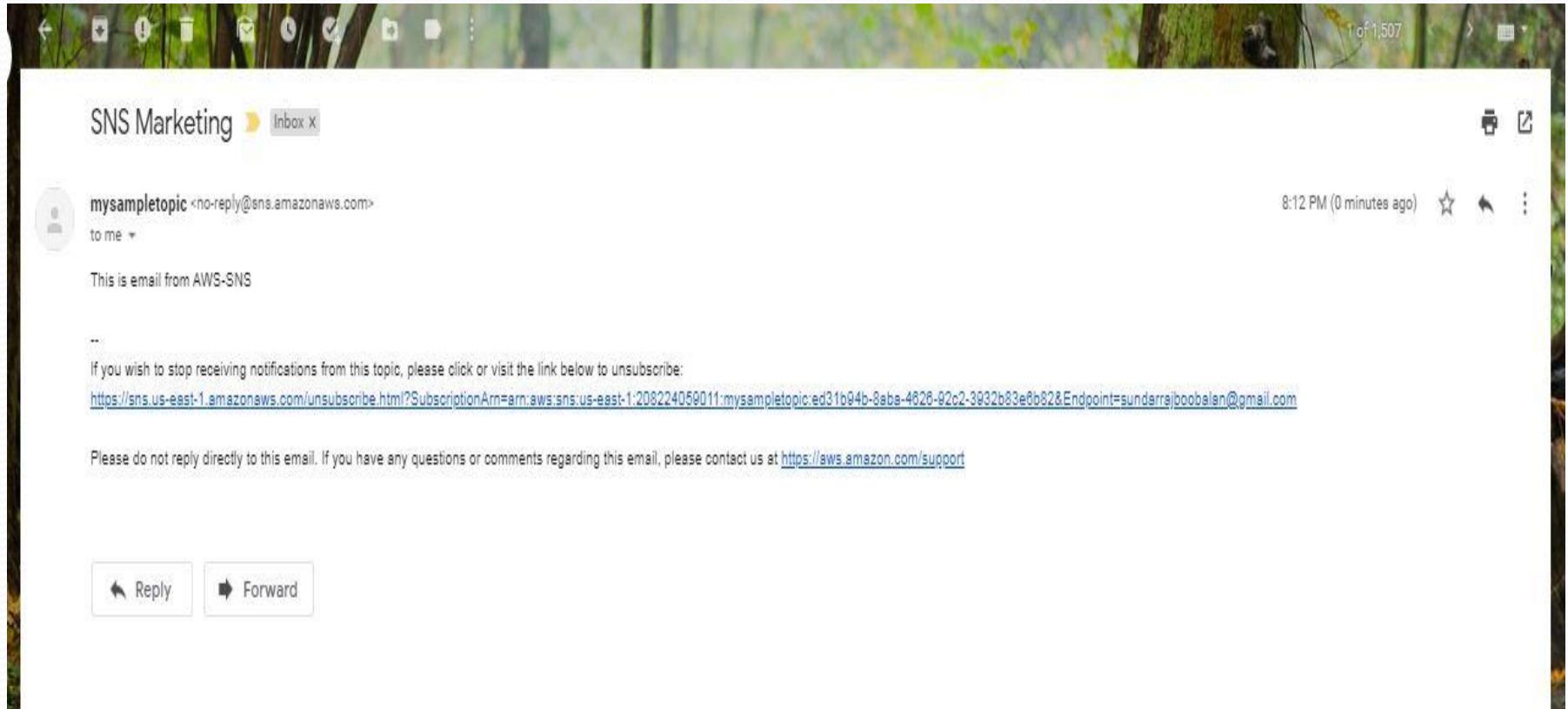
Do you want to check the SNS notifications then first select topic and click on publish message on upper right side panel topic

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



We can use these SNS Service in where you applications and another AWS services like Cloud Watch, Auto scaling....etc by selecting SNS Topic.

Simple Queuing Service (SQS)

SQS

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

AWS Management Console

1. Sign in to the Amazon SQS console.
2. Choose SQS and 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**.

Content-Based Deduplication parameter specific to FIFO queues for Avoiding duplication messages.

Queue Attributes

Default Visibility Timeout	<input type="text" value="30"/>	<input type="button" value="seconds"/> ▼	Value must be between 0 seconds and 1200000 seconds.
Message Retention Period	<input type="text" value="4"/>	<input type="button" value="days"/> ▼	Value must be between 1 minute and 120 days.
Maximum Message Size	<input type="text" value="256"/>	KB	Value must be between 1 and 256 KB.
Delivery Delay	<input type="text" value="0"/>	<input type="button" value="seconds"/> ▼	Value must be between 0 seconds and 1200000 seconds.
Receive Message Wait Time	<input type="text" value="0"/>	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.

***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, Content- Based Deduplication column displays whether you have enabled exactly-once processing

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

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

Adding Permissions to an Amazon SQS Queue

We 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

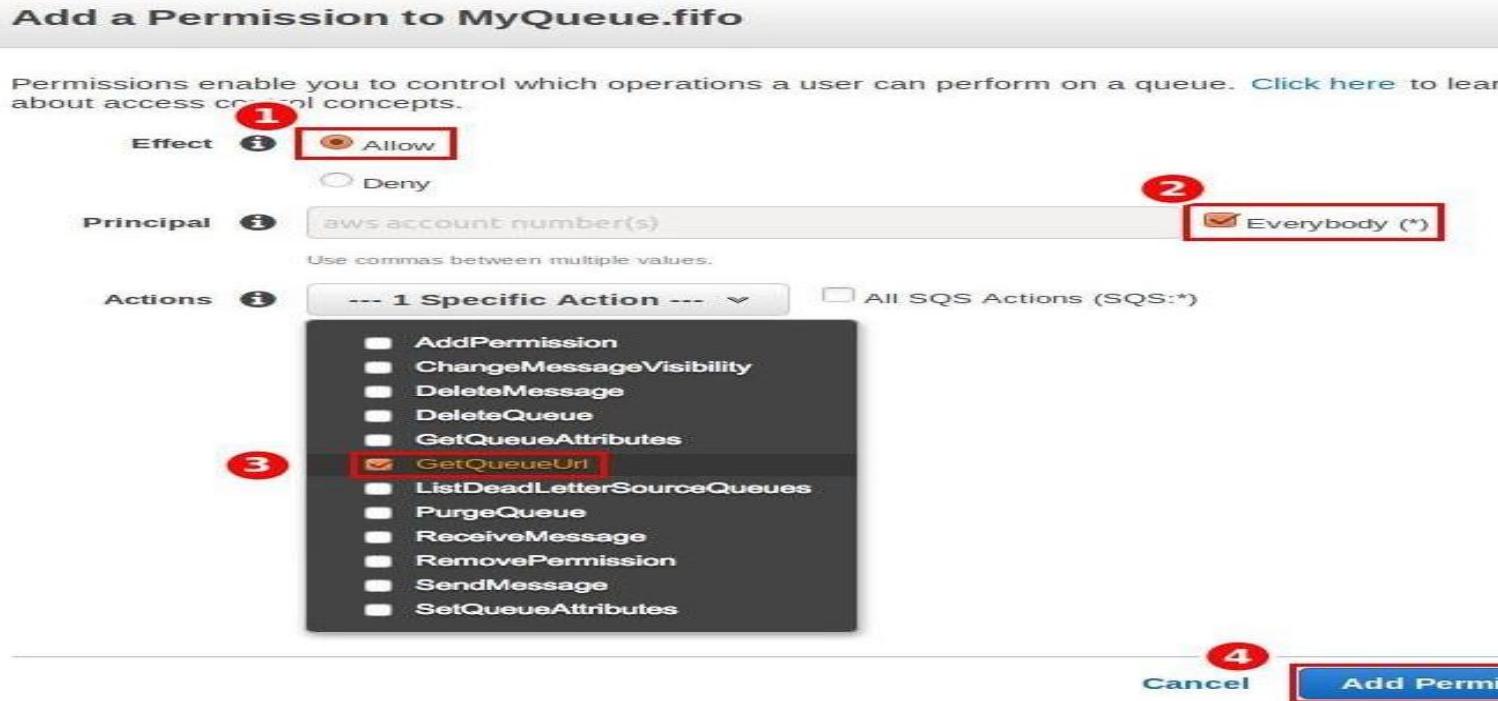


3. From Queue Actions, select Add a Permission.



The Add a Permission dialog box is displayed.

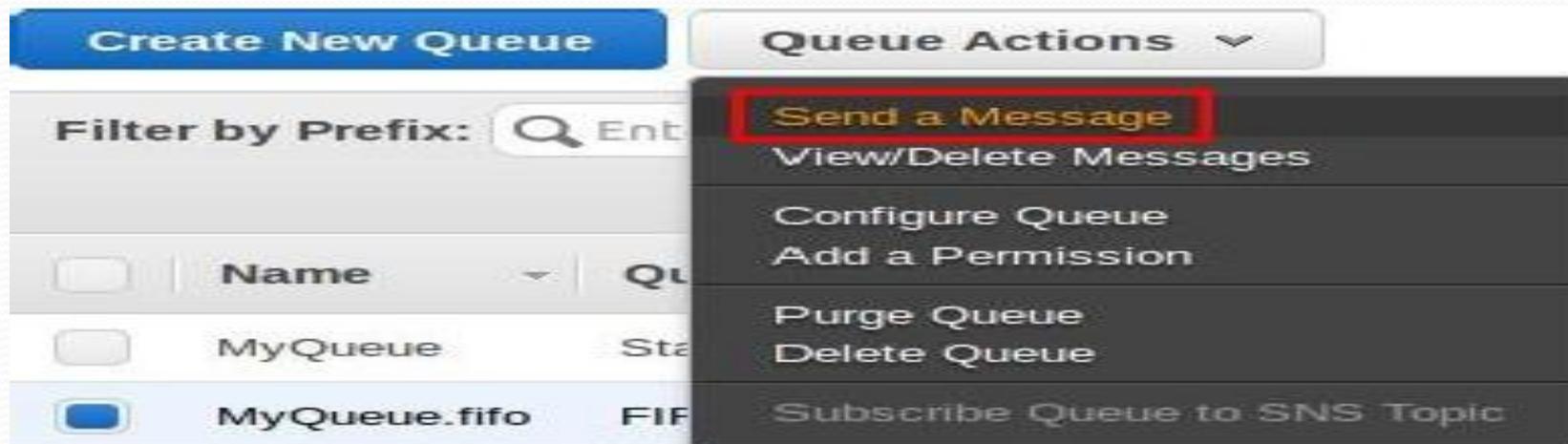
4. In this example, you allow anyone to get the queue's URL:



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 Get QueueUrl box.

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.



The Send a Message to Queue Name 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

X

Message Body Message Attributes

Enter the text of a message you want to send.

This is my message text.

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

Message Deduplication ID i 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.

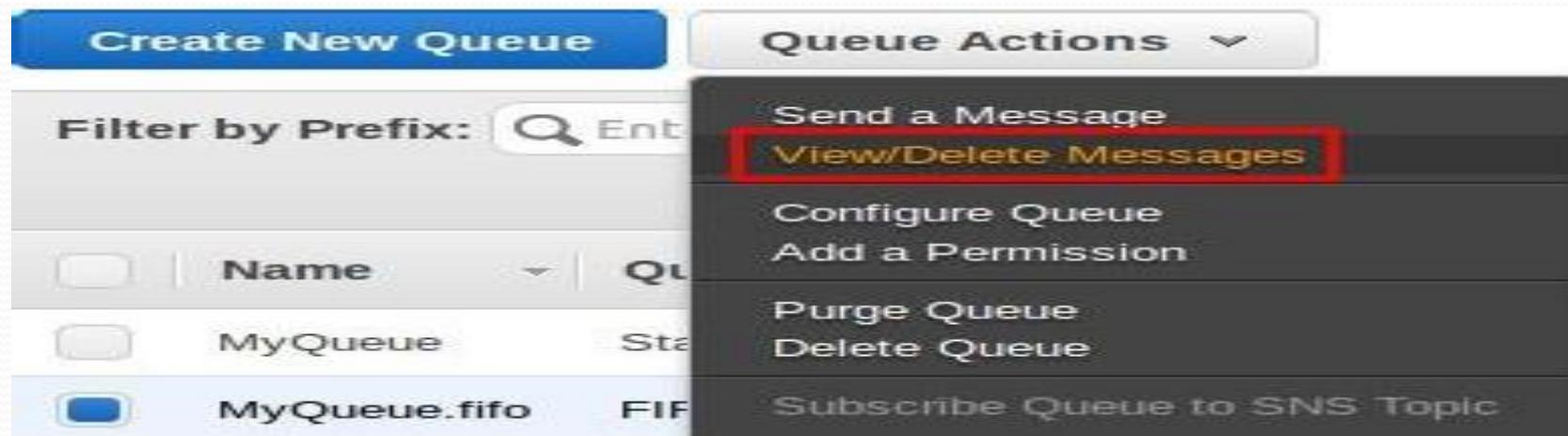
Note

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

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 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.

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



The View/Delete Messages in Queue Name 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.

Choose **Start Polling for messages**.

View/Delete Messages in MyQueue fifo

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

Polling for new messages once every 2 seconds.

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...	MyMessageDeduplication1...	181

54%

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

Deleting an Amazon SQS Queue

If you don't use an Amazon SQS queue , it is a best practice to delete it from Amazon SQS.

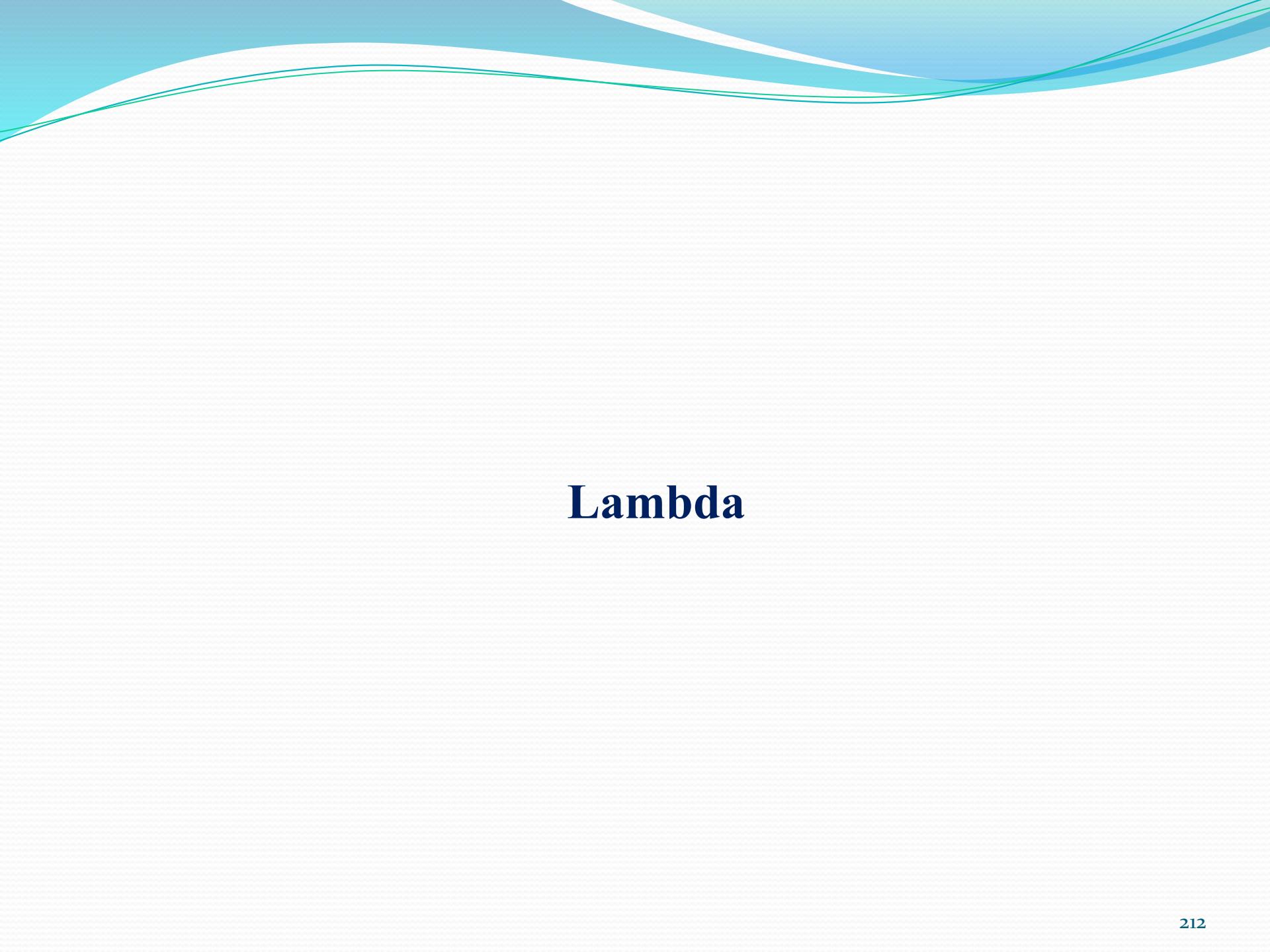
AWS Management Console

1. Sign in to the Amazon SQS console.
2. From the queue list, select a queue

The screenshot shows the AWS Management Console interface for Amazon SQS. At the top, there are filters for 'Name' and 'Queue Type'. Below the filters, a list of queues is displayed. The first queue is 'MyQueue', which is a Standard queue. The second queue, 'MyQueue.fifo', is highlighted with a red border, indicating it is selected. It is a FIFO queue.

3. From Queue Actions, select Delete Queue.

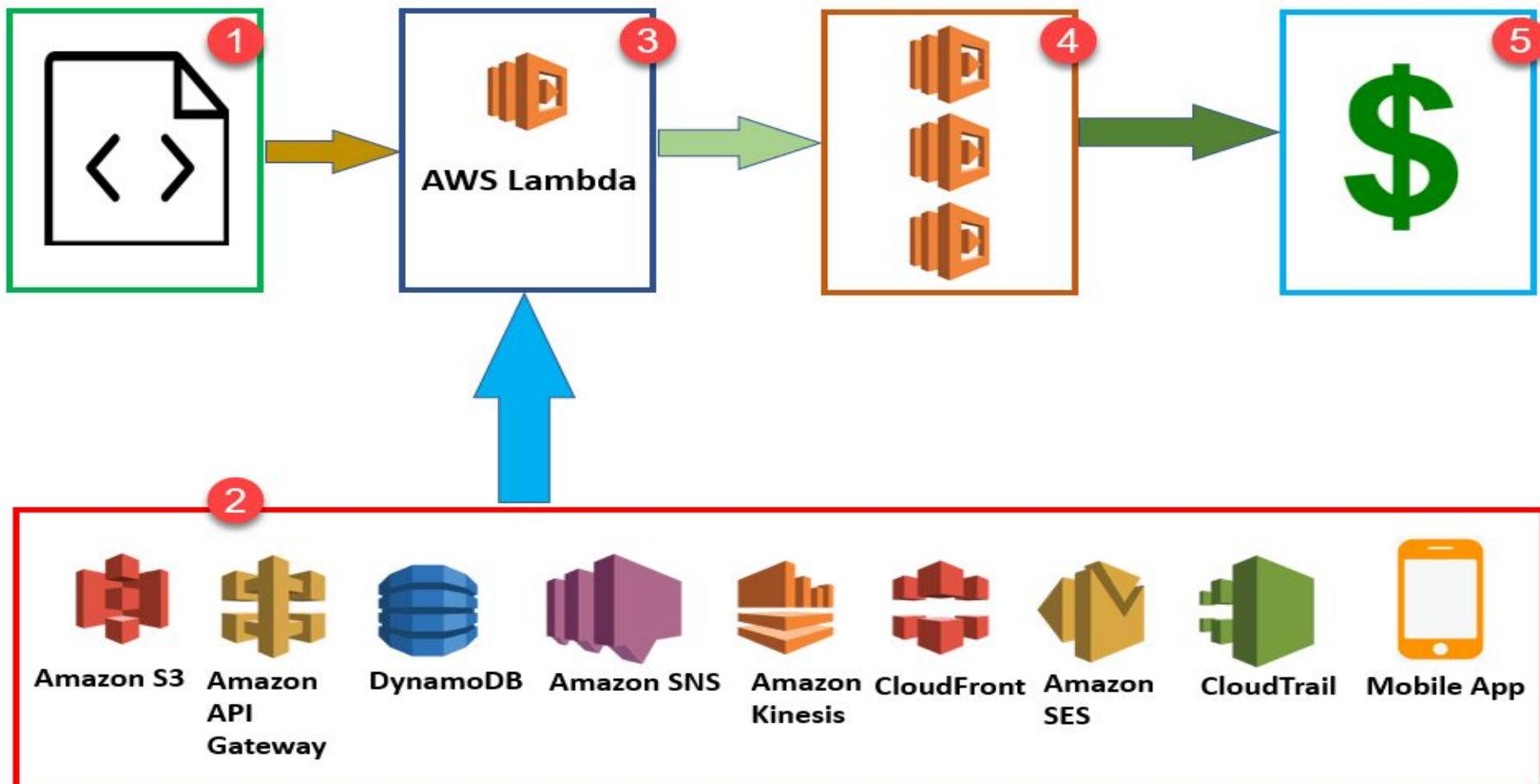
The screenshot shows the AWS Management Console interface for Amazon SQS. On the left, there is a 'Create New Queue' button and a 'Filter by Prefix:' search bar. On the right, there is a 'Queue Actions' dropdown menu. The menu items are: Send a Message, View/Delete Messages, Configure Queue, Add a Permission, Purge Queue, and Delete Queue. The 'Delete Queue' option is highlighted with a red border.



Lambda

Serverless:

The serverless computing model allows you to build and run applications and services without having to worry about infrastructure or servers. It eliminates infrastructure management tasks such as server provisioning, patching, operating system maintenance, scaling, and capacity provisioning.



Step 1: First upload your AWS Lambda code in any language supported by AWS Lambda. Java, Python, Go, and C# are some of the languages that are supported by AWS Lambda function.

Step 2: These are some AWS services which allow you to trigger AWS Lambda.

Step 3: AWS Lambda helps you to upload code and the event details on which it should be triggered.

Step 4: Executes AWS Lambda Code when it is triggered by AWS services:

Step 5: AWS charges only when the AWS lambda code executes, and not otherwise.

Upload files in an S3 bucket

- When HTTP get/post endpoint URL is hit
- For adding/modifying and deleting Dynamo DB tables
- In the process of data streams collection
- Push notification
- Hosting of website
- Email sending

Create function Info

Choose one of the following options to create your function.

Author from scratch

Start with a simple Hello World example.

Use a blueprint

Build a Lambda application from sample code and configuration presets for common use cases.

Container image

Select a container image to deploy for your function.

Browse serverless app repository

Deploy a sample Lambda application from the AWS Serverless Application Repository.

Basic information

Function name

Enter a name that describes the purpose of your function.

Use only letters, numbers, hyphens, or underscores with no spaces.

Runtime Info

Choose the language to use to write your function. Note that the console code editor supports only Node.js, Python, and Ruby.



Permissions Info

By default, Lambda will create an execution role with permissions to upload logs to Amazon CloudWatch Logs. You can customize this default role later when adding triggers.

Change default execution role

Click Create Function In Lambda Dashboard

We can choose any one from following options for Creating Function

1.Author from scratch

2.Use a Blueprint

3.Container Image

4.Browse Serverless app repository

Lambda Supports Following Runtime

- Node.js
- Python
- Ruby
- Java
- Go
- .NET Core
- Custom Runtime

The screenshot shows the AWS Lambda function code editor interface. At the top, there's a navigation bar with the AWS logo, 'Services' dropdown, and 'Resource Groups' dropdown. Below the header, the title 'Lambda function code' is displayed, followed by a note: 'Code is preconfigured by the chosen blueprint. You can configure it after you create the function. [Learn more about deploying Lambda functions.](#)' Under the 'Runtime' section, 'Python 3.7' is selected. The code editor contains the following Python script:

```
1 import json
2
3 print('Loading function')
4
5
6 def lambda_handler(event, context):
7     #print("Received event: " + json.dumps(event, indent=2))
8     print("value1 = " + event["key1"])
9     print("value2 = " + event["key2"])
10    print("value3 = " + event["key3"])
11    return event["key1"] # Echo back the first key value
12    #raise Exception('Something went wrong')
13
```

Function name
Lambdafunction

Execution role:
Choose a role that defines the permissions of your function. To create a custom role, go to the [IAM console](#).

- Create a new role with basic Lambda permissions
- Use an existing role
- Create a new role from AWS policy templates

i Role creation might take a few minutes. Please do not delete the role or edit the trust or permissions policies in this role.

Lambda will create an execution role named Lambdafunction-role-ylkct6yw, with permission to upload logs to Amazon CloudWatch Logs.

Lambda function code
Code is preconfigured by the chosen blueprint. You can configure it after you create the function. [Learn more](#) about deploying Lambda functions.

Runtime
Python 3.7

```

1 import json
2 print("Loading function")
3
4
5 - def lambda_handler(event, context):
6     #print("Received event: " + json.dumps(event, indent=2))
7     print("value1 = " + event['key1'])
8     print("value2 = " + event['key2'])
9     print("value3 = " + event['key3'])
10    return event['key1'] # Echo back the first key value
11    #raise Exception('something went wrong')
12
13

```

i Successfully created the function hello-world. You can now change its code and configuration. To invoke your function with a test event, choose "Test". X

Lambda > Functions > hello-world ARN - arn:aws:lambda:eu-west-2:387124123361:function:hello-world

hello-world

Throttle Qualifiers Actions Select a test event Test Save

Configuration Permissions Monitoring

▼ Designer

 **hello-world**
Layers (0)

Blue Green Deployment

Another traffic shifting pattern is enabling blue/green deployments. This near zero-downtime release enables traffic to shift to the new live environment (green) while still keeping the old production environment (blue) warm in case a rollback is necessary. Since API Gateway allows you to define what percentage of traffic is shifted to a particular environment; this style of deployment can be an effective technique. Since blue/green deployments are designed to reduce downtime, many customers adopt this pattern for production changes.

CloudFront

Amazon CloudFront is a fast Content Delivery Network (CDN) service that securely delivers data, videos, applications, and APIs to customers globally with low latency, high transfer speeds, all within a developer-friendly environment.

Content Delivery Network (CDN) improves read performance content is cached at the edge.

225 point of presence is globally.

CDN protect against multiple types of attacks including network and application layer DDoS attacks. These services co-reside at edge networking locations – globally scaled and connected via the AWS network backbone – providing a more secure, performance, and available experience for your users.

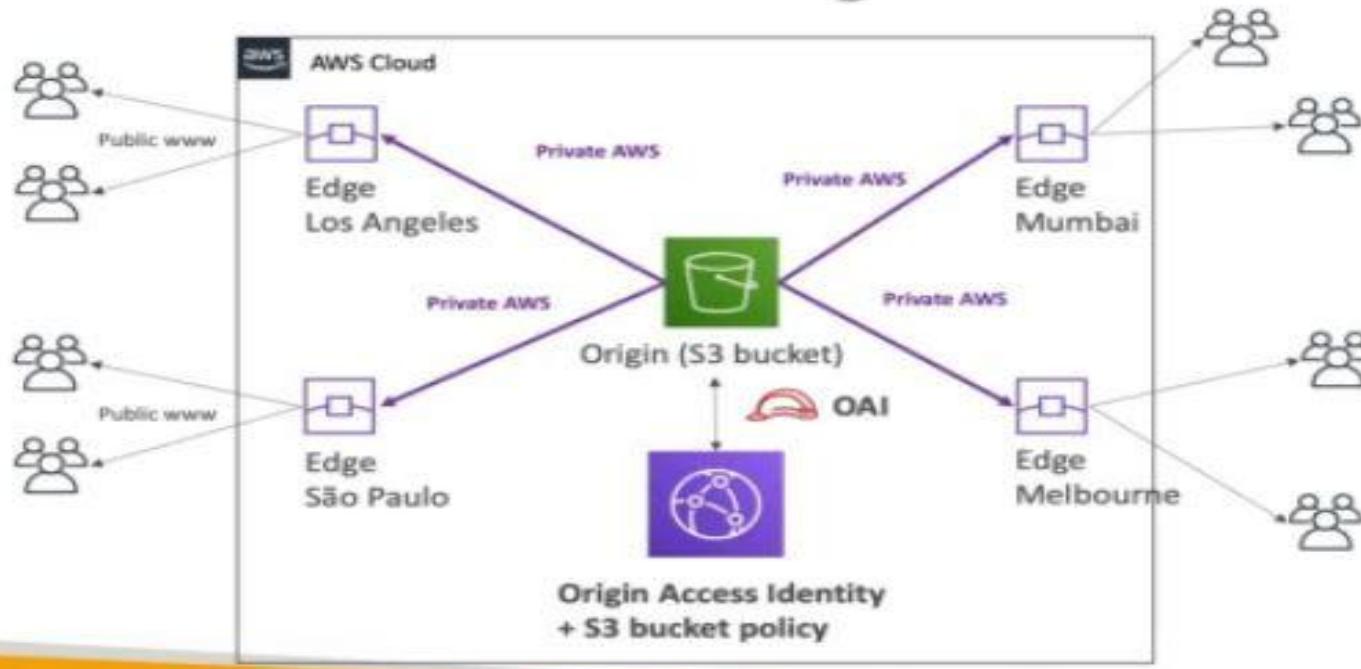
CloudFront works seamlessly with any AWS origin, such as Amazon S3, Amazon EC2, Elastic Load Balancing, or with any custom HTTP origin. We can customize your content delivery through CloudFront using the secure and programmable edge computing features.

Using Amazon S3 Buckets for Your Origin:

When you use Amazon S3 as an origin for your distribution, you place any objects that you want CloudFront to deliver in an Amazon S3 bucket. You can use any method that is supported by Amazon S3 to get your objects into Amazon S3, for example, the Amazon S3 console or API, or a third-party tool. You can create a hierarchy in your bucket to store the objects, just as you would with any other Amazon S3 bucket.

Using an existing Amazon S3 bucket as your CloudFront origin server doesn't change the bucket in any way; you can still use it as you normally would to store and access Amazon S3 objects at the standard Amazon S3 price. AWS incur regular Amazon S3 charges for storing the objects in the bucket.

CloudFront – S3 as an Origin



Origin Access Identity (OAI) is used for sharing private content via CloudFront. The OAI is a virtual user identity that will be used to give your CloudFront distribution permission to fetch a private object from your origin server (e.g. S3 bucket).

To work with Origin Access Identities, navigate to **Clouds > AWS Global > CF Origin Access Identities**.

Actions

The following action buttons are available on the page.

New Origin Access Identity - Create a new OAI object. In order to share private content through CF, you must have an OAI.

Below is more detailed information about your CloudFront Origin Access Identity.

Name - A RightScale-specific name for the OAI.

ID - A unique virtual identity that's used to give your distribution permission to fetch a private object from your origin server S3 bucket.

S3 Canonical User ID - The User ID that's used to grant your OAI permission to access the objects you want to deliver as private content.

Caller Reference - This is a unique value that's designed to prevent accidental replays of your request.

Comment - Provide a description or notes about the OAI.

Geo Restriction, a new feature that allows you to use Amazon CloudFront to restrict access to your content based on the geographic location of your viewers. With Geo Restriction you can choose the countries where you want Amazon CloudFront to deliver your content.

CloudFront Geo Restriction:

When a user requests your content, CloudFront typically serves the requested content regardless of where the user is located. If you need to prevent users in specific countries from accessing your content, you can use the CloudFront geo restriction feature to do one of the following:

- Allow your users to access your content only if they're in one of the countries on a **Whitelist** of approved countries.
- Prevent your users from accessing your content if they're in one of the countries on a **Blacklist** of banned countries.

Search CloudFront in AWS Console

CloudFront

Distributions

Policies

What's new *

Telemetry

Monitoring

Alarms

Logs. NEW

Reports & analytics

Cache statistics

Popular objects

Top referrers

Usage

Viewers

Security

Origin access identity

Field-level encryption

Key management

Public keys

Key groups

Savings Bundle

Amazon CloudFront - Get started

Either your search returned no results, or you do not have any distributions. Click the button below to create a new CloudFront distribution. A distribution allows you to distribute content using a worldwide network of edge locations that provide low latency and high data transfer speeds (learn more)

Create Distribution

Click Create Distribution

Step 1: Select delivery method
Step 2: Create distribution

Create Distribution

Origin Settings

Origin Domain Name: elasticbeanstalk-us-east-1-20822405901

Origin Path:

Enable Origin Shield: No

Origin ID: S3-elasticbeanstalk-us-east-1-20822405

Restrict Bucket Access: No

Origin Connection Attempts: 3

Origin Connection Timeout: 10

Origin Custom Headers: Header Name: Value:

Default Cache Behavior Settings

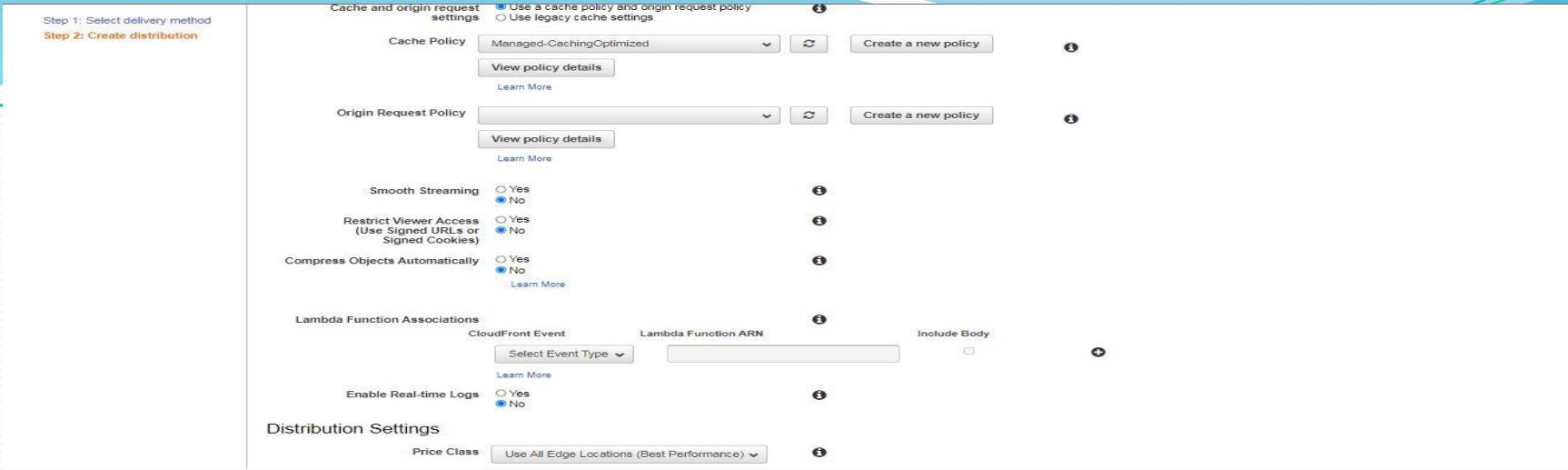
Path Pattern: Default (*)

Viewer Protocol Policy: HTTP and HTTPS

Allowed HTTP Methods: GET, HEAD

Field-level Encryption Config:

Cached HTTP Methods: GET, HEAD (Cached by default)



Aliases

A complex type that contains information about CNAMEs (alternate domain names), if any, for this distribution.

Required: No

Type: List of String

Update requires: No Interruption

CacheBehaviors A complex type that contains zero or More CacheBehavior elements.

Required: No

Type: List of CacheBehavior

Update requires: No Interruption

Comment

- Any comments you want to include about the distribution.
- If you don't want to specify a comment, include an empty Comment element.

DefaultCacheBehavior

A complex type that describes the default cache behavior if you don't specify a CacheBehavior element or if files don't match any of the values of PathPattern in Cache Behavior elements. You must create exactly one default cache behavior.

Enabled

From this field, you can enable or disable the selected distribution

Required: Yes

Type: Boolean

InValidating Files

If you need to remove a file from CloudFront edge caches before it expires, you can do one of the following:

- Invalidate the file from edge caches. The next time a viewer requests the file, CloudFront returns to the origin to fetch the latest version of the file.
- Use file versioning to serve a different version of the file that has a different name.

To invalidate files, you can specify either the path for individual files or a path that ends with the * wildcard, which might apply to one file or to many, as shown in the following examples:

- /images/image1.jpg
- /images/image*
- /images/*

InValidating Files

If you need to remove a file from CloudFront edge caches before it expires, you can do one of the following:

- Invalidate the file from edge caches. The next time a viewer requests the file, CloudFront returns to the origin to fetch the latest version of the file.
- Use file versioning to serve a different version of the file that has a different name.

To invalidate files, you can specify either the path for individual files or a path that ends with the * wildcard, which might apply to one file or to many, as shown in the following examples:

- /images/image1.jpg
- /images/image*
- /images/*

CloudFront signed URLs and signed cookies provide the same basic functionality: they allow you to control who can access your content. If you want to serve private content through CloudFront and you're trying to decide whether to use signed URLs or signed cookies, consider the following.

- Use signed URLs in the following cases:
- You want to restrict access to individual files, for example, an installation download for your application.
- Your users are using a client (for example, a custom HTTP client) that doesn't support cookies.

Use signed cookies in the following cases:

- You want to provide access to multiple restricted files, for example, all of the files for a video in HLS format or all of the files in the subscribers' area of website.
- You don't want to change your current URLs.

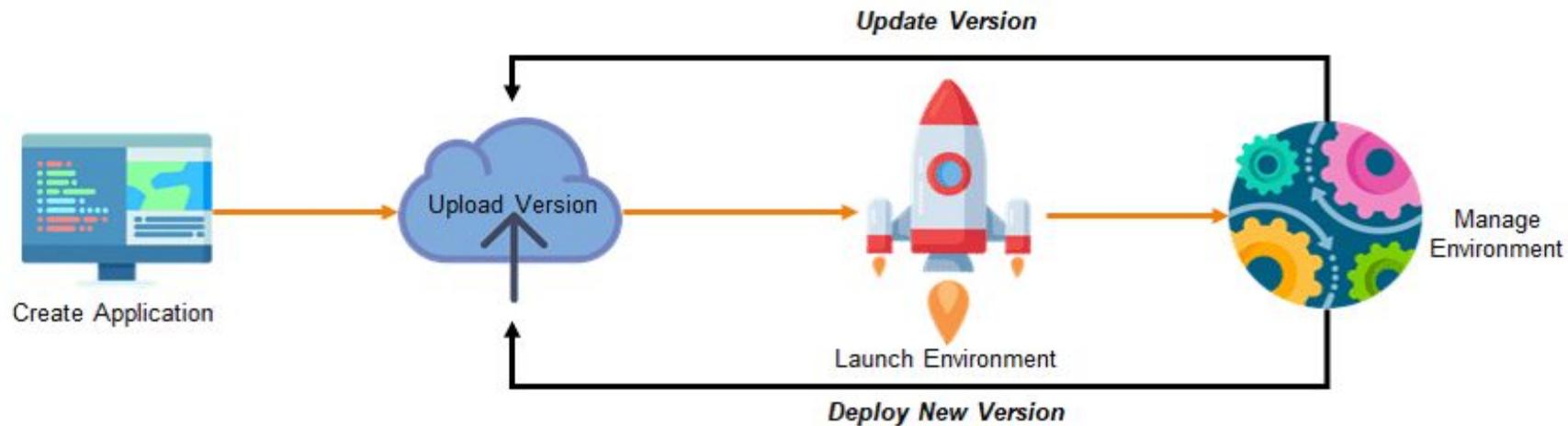
If you are not currently using signed URLs, and if your (unsigned) URLs contain any of the following query string parameters, you cannot use either signed URLs or signed cookies:

- Expires
- Policy
- Signature
- Key-Pair-Id

CloudFront assumes that URLs that contain any of those query string parameters are signed URLs, and therefore won't look at signed cookies.

Elastic Beanstalk

AWS Elastic Beanstalk is a Compute service which makes it easier for the developers to quickly deploy and manage applications which you upload to the AWS cloud. Developers simply upload their application to the AWS cloud, and then let the AWS Beanstalk provision and handle the configuration for you. Your application will be provided with capacity provisioning, load balancing, auto-scaling, and health monitoring.



Before using Amazon elastic beanstalk service, you have to create a local application of any platform. It can be Python, PHP, Node.js ,Ruby, Go ,Docker , .Net ,Java etc. After that you have to create an application in Elastic Beanstalk with an environment where you can upload your local application. Then you deploy it and use the URL provided for it to launch it.

AWS Elastic Beanstalk Benefits

Now that we understand what is Elastic Beanstalk in AWS and how does Elastic beanstalk work, let now understand what are the benefits of using Elastic Beanstalk. So, Elastic Beanstalk provides the user with several benefits and they are:

- Auto scaling options
- Developer productivity
- Customization
- Cost-effective
- Management and updates

Open The Elastic beanstalk from AWS Console and Give the Application Name

The screenshot shows the 'Create a web app' step of the AWS Elastic Beanstalk wizard. At the top, there is a blue info icon with the text: 'Managed updates are now enabled by default for new environments on supporting platforms.' Below this, the heading 'Create a web app' is displayed. A descriptive text follows: 'Create a new application and environment with a sample application or your own code. By creating an environment, you allow AWS Elastic Beanstalk to manage AWS resources and permissions on your behalf. [Learn more](#)'.

The main form area is titled 'Application information'. Under 'Application name', the value 'my-first-webapp-bea' is entered into a text input field. Below the input field, a note states: 'Up to 1000 Unicode characters, not including newline (less than 1%).'

Provide the Following information:

Platform: NodeJs

Platform Branch: NodeJs running on 64 bit Amazon Linux2

Application Code: Choose Sample Code or Upload your Code

Platform

Platform: Node.js

Platform branch: Node.js 12 running on 64bit Amazon Linux 2

Platform Version: --- Choose a platform version ---

Application code

Sample application: Get started right away with sample code.

Uploaded your code

Elastic Beanstalk

Environments Applications

my-first-webapp-beanstalk Application versions Saved configurations

MyFirstWebappBeanstalk-env

Elastic Beanstalk > Environments > MyFirstWebappBeanstalk-env

MyFirstWebappBeanstalk-env.geba.uqqscsmq.eu-west-2.elasticbeanstalk.com (gmtgognitz) Refresh Environment actions

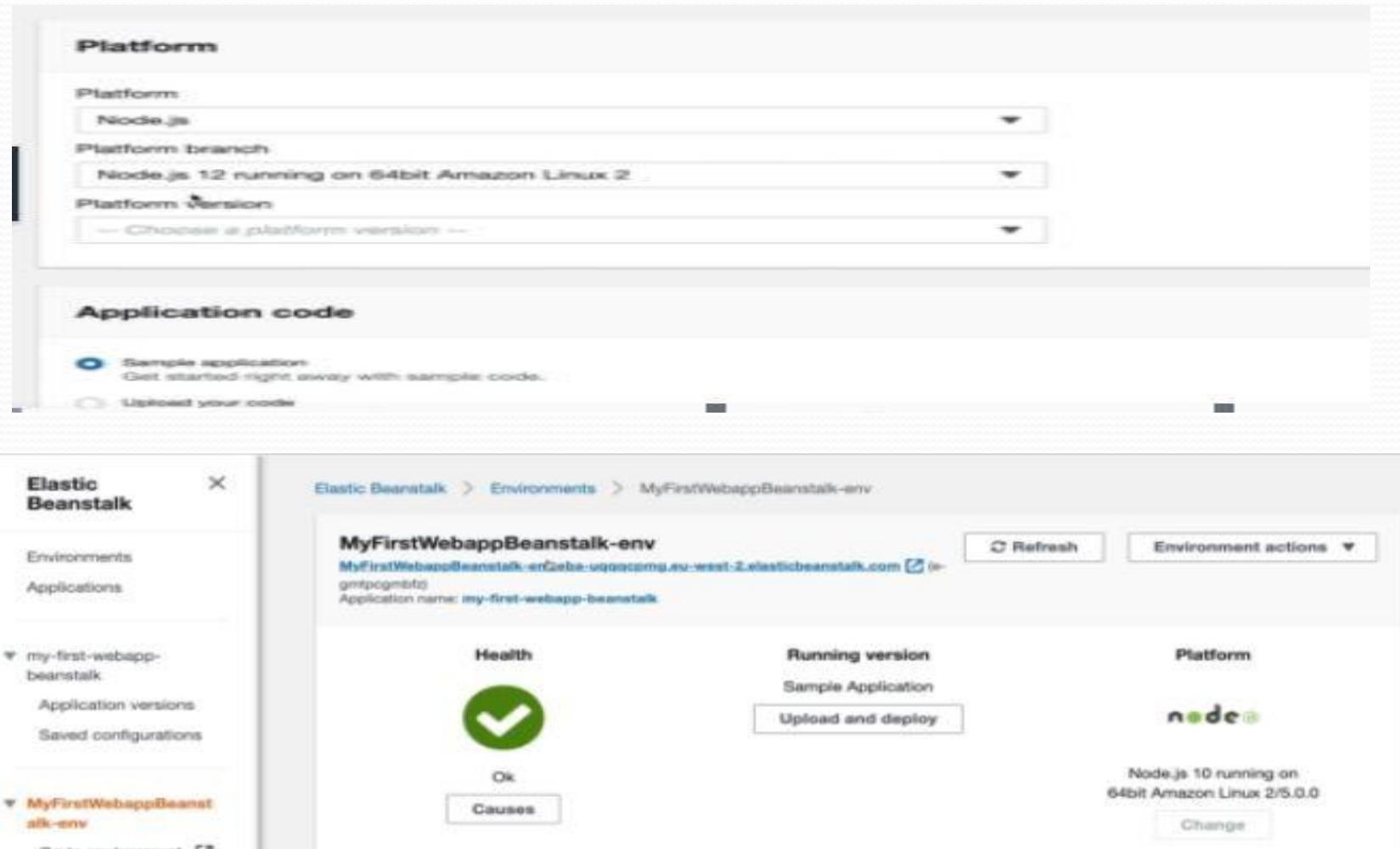
Application name: my-first-webapp-beanstalk

Health: Ok Causes

Running version: Sample Application Upload and deploy

Platform: node.js

Node.js 10 running on 64bit Amazon Linux 2/5.0.0 Change



Open the Logs from left side Panel

It Shows all the Steps follows Creating Beanstalk.

The screenshot shows the AWS Elastic Beanstalk console. On the left, a sidebar lists environment-related options like Application versions, Saved configurations, and the current environment, MyFirstWebappBeanstalk-env. The main content area is titled 'Events' and displays a table of log entries. The table has columns for Time, Type, and Details. The details column contains messages such as 'Successfully launched environment: MyFirstWebappBeanstalk-env' and 'Application available at MyFirstWebappBeanstalk-env.eba-uqqqcprng.eu-west-2.elasticbeanstalk.com'. The logs show the progression from launching the environment to it becoming available.

Time	Type	Details
2020-05-06 13:19:53 UTC+0100	INFO	Successfully launched environment: MyFirstWebappBeanstalk-env
2020-05-06 13:19:52 UTC+0100	INFO	Application available at MyFirstWebappBeanstalk-env.eba-uqqqcprng.eu-west-2.elasticbeanstalk.com,
2020-05-06 13:19:48 UTC+0100	INFO	Added instance [i-0e360c973e2ec569c] to your environment.
2020-05-06 13:19:48 UTC+0100	INFO	Environment health has transitioned from Pending to Ok. Initialization completed 5 seconds ago and took 2 minutes.
2020-05-06 13:19:15 UTC+0100	INFO	Waiting for EC2 instances to launch. This may take a few minutes.
2020-05-06 13:18:48 UTC+0100	INFO	Environment health has transitioned to Pending. Initialization in progress (running for 55 seconds). There are no instances.

Open the Health from left side Panel

It Shows Health Status for Application.

The screenshot shows the 'Enhanced health overview' for the environment 'MyFirstWebappBeanstalk-env'. It displays one instance, 'i-0e360c973e2ec569c', which is marked as 'OK'. The status is listed as 'Running' with a deployment ID of 'N/A' and a request count of '0'. The overall status is also 'OK'. There are buttons for 'Filter by', 'Instance actions', and a refresh icon.

Open the Monitoring Properties

It Shows information about CPU Utilization ,Network in and Network out.

The screenshot shows the 'Monitoring' overview for the environment 'MyFirstWebappBeanstalk-env'. Key metrics displayed are CPU Utilization at 4.2%, Max Network In at 15MB, and Max Network Out at 153KB. The monitoring section includes a timeline for environment health and a chart for CPU utilization over a 3-hour period. The CPU utilization chart shows a single data point at approximately 4.2% at 13:18.

Deploying Application

The screenshot shows the AWS Elastic Beanstalk console. On the left, a sidebar menu includes 'Environments' and 'Applications'. Under 'my-first-webapp-beanstalk', there are 'Application versions' and 'Saved configurations'. A section for 'MyFirstWebappBeanstalk-prod' is expanded, showing 'Go to environment' (with a link), 'Configuration', 'Logs', and 'Health'. The main content area displays the environment details for 'MyFirstWebappBeanstalk-prod'. It shows a green 'Health' status with a checkmark icon and the text 'Ok'. The 'Running version' is 'Sample Application' with a 'Upload and deploy' button. The 'Platform' is 'node.js' with 'Node.js 10 running on 64bit Amazon Linux 2/5.0.0' and a 'Change' button. Below this, a 'Recent events' section has a 'Show all' button. At the top right, there are 'Refresh' and 'Environment actions' buttons.

Congratulations

Your first AWS Elastic Beanstalk Node.js application is now running on your own dedicated environment in the AWS Cloud.

This environment is launched with Elastic Beanstalk Node.js Platform

What's Next?

- [AWS Elastic Beanstalk overview](#)
- [AWS Elastic Beanstalk concepts](#)
- [Deploy an Express Application to AWS Elastic Beanstalk](#)
- [Deploy an Express Application with Amazon ElastiCache to AWS Elastic Beanstalk](#)
- [Deploy a Derby Application with Amazon ElastiCache to AWS Elastic Beanstalk](#)
- [Customizing and Configuring a Node.js Container](#)
- [Working with Logs](#)

Relational Database Service (RDS)

Amazon Relational Database Service (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

The screenshot shows the Amazon RDS console interface. At the top, there's a navigation bar with 'Services', 'Resource Groups', and other account information. Below the navigation bar, a modal window titled 'Amazon Aurora' is open, providing information about Aurora and a 'Create database' button. The main content area is divided into two sections: 'Resources' and 'Additional information'. The 'Resources' section lists various RDS resources: DB Instances (0/40), Parameter groups (1), Allocated storage (0 TB/100 TB), Reserved instances (0/40), Option groups (1), Snapshots (0), Events (0), Recent events (0), and Event subscriptions (0/20). The 'Additional information' section links to 'Getting started with RDS', 'Overview and features', 'Documentation', 'Articles and tutorials', 'Data import guide for MySQL', 'Data import guide for Oracle', 'Data import guide for SQL Server', 'New RDS feature announcements', 'Pricing', and 'Forums'.

Select Engine MYSQL and click on Next

The screenshot shows the 'Select engine' step of an 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 section titled 'Engine options'. It lists six database engines with their logos: Amazon Aurora (orange), MySQL (blue), MariaDB (yellow), PostgreSQL (blue), Oracle (red), and Microsoft SQL Server (purple). The MySQL option is selected, indicated by a blue circle and highlighted with a light blue border. Below the MySQL section, there is a detailed description of MySQL and a bulleted list of its features.

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.

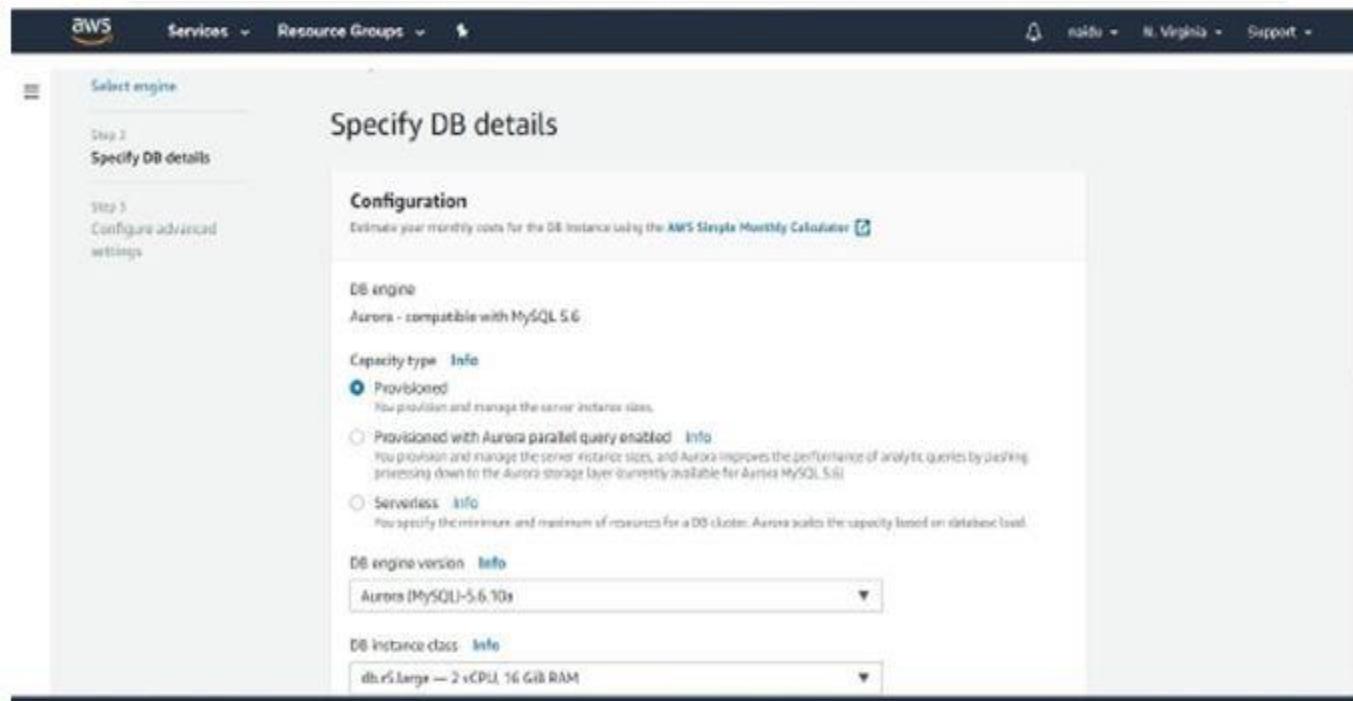
- Supports database size up to 32 TiB.
- Supports General Purpose, Memory Optimized, and InnoDB Performance instance classes.

Specify DB details: it your DB Instance Configuration Settings

- Capacity Type : select Provisioned
- DB Engine Version: MYSQL5.6.10a
- DB Instance class: db.r5.large – 2vCPU , 16 gib RAM
- Multi-A 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 instance Connection
- Master Password:** password for DB Instance Master user and Click on Next



The screenshot shows the AWS Aurora DB instance creation wizard at Step 2: Specify DB details. The configuration section includes:

- DB engine:** Aurora - compatible with MySQL 5.6
- Capacity type:** Provisioned. You provision and manage the server instance sizes.
- Provisioned with Aurora parallel query enabled:** You provision and manage the server instance sizes, and Aurora improves the performance of analytic queries by pushing processing down to the Aurora storage layer (currently available for Aurora MySQL 5.6).
- Serversless:** You specify the minimum and maximum of resources for a DB cluster. Aurora scales the capacity based on database load.
- DB engine version:** Aurora (MySQL)-5.6.10x
- DB instance class:** db.r5.large — 2 vCPUs, 16 GiB RAM

Configure Advanced Settings

Network & Security

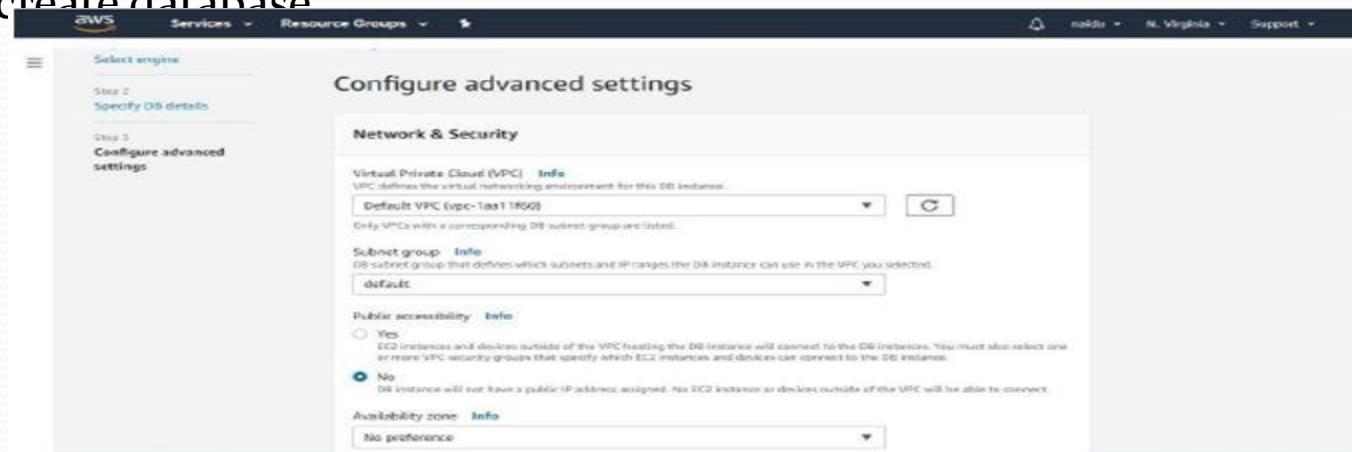
- Virtual Private Cloud (VPC): select The default VPC
- Subnet group: select default subnet
- Public accessibility: do you want to connect to your DB Instance from outside of selected VPC then click YES otherwise click no. if you select NO then you connect from within that VPC only.
- DB cluster identifier: 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
- 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: Un select Enable deletion protection
- Click on create database



aws Services Resource Groups

Encryption

Encryption:

Enable encryption [Learn more](#) 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.

Disable encryption

Failover

Priority: Info

No preference

Backup

Backup retention period [Info](#) Select the number of days that Amazon RDS should retain automatic backups of this DB instance.

1 day

This screenshot shows the 'Encryption' section of the AWS RDS configuration interface. It includes a radio button for enabling encryption (with a link to learn more), a radio button for disabling encryption (which is selected), and a note about master key IDs appearing in the list after creation via KMS.

aws Services Resource Groups

Copy tags to snapshots

Backtrack

Backtrack lets you quickly move an Aurora database to a prior point in time without needing to restore data from a backup. [Info](#)

Enable Backtrack

Disable Backtrack

Monitoring

Enhanced monitoring

Enable enhanced monitoring Enhanced monitoring metrics are useful when you want to see how different processes or threads use the CPU.

Disable enhanced monitoring

Monitoring Role

Default

Granularity

60 seconds

I authorize RDS to create the IAM role rds-monitoring-role.

This screenshot shows the 'Monitoring' section of the AWS RDS configuration interface. It includes a checkbox for copying tags to snapshots, a section for 'Backtrack' (which is disabled), and a section for 'Enhanced monitoring'. The 'Enable enhanced monitoring' option is selected, with a note explaining its usefulness for CPU usage analysis. It also includes fields for 'Monitoring Role' (set to 'Default') and 'Granularity' (set to '60 seconds'), and a checkbox for authorizing RDS to create an IAM role for monitoring.

AWS Services Resource Groups

Performance Insights

Enable Performance Insights
 Disable Performance Insights

Log exports

Select the log types to publish to Amazon CloudWatch Logs

Audit log
 Error log
 General log
 Slow query log

IAM role:
The following service-linked role is used for publishing logs to CloudWatch Logs.

RDS Service Linked Role

Ensure that General, Slow Query, and Audit Logs are turned on. Error logs are enabled by default.
[Learn more](#)

Cancel Previous Create database

AWS Services Resource Groups

Maintenance

Auto minor version upgrade [Info](#)
 Enable auto minor version upgrade
Enables automatic upgrades to new minor versions as they are released. The automatic upgrades occur during the maintenance window for the DB instance.

Disable auto minor version upgrade

Maintenance window [Info](#)
Select the period in which you want pending modifications or patches applied to this DB instance by Amazon RDS.

Select window
 No preference

Deletion protection

Enable deletion protection
Protects the database from being deleted accidentally. While this option is enabled, you can't delete the database.

Cancel Previous **Create database**

Connecting with Database

- Connecting to RDS MYSQL server you need mysql-client in your instance
 - Launch one ec2 instance and install following mysql client application
 - Sudo apt-get install mysql-client mysql-server
 - Run the below command to connect to RDS mysql-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 then 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.

```
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>
```

Then you are connected to MySQL server

· For creating tables and see previous data first you switch your database
(use database name)

```
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

```
ubuntu@ip-172-31-90-13:~$ 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)
```

```
ubuntu@ip-172-31-90-13: ~
+-----+-----+-----+-----+-----+
| Field | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| id   | int(11)   | YES  |     | NULL    |       |
| name | varchar(10) | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)

mysql> inser 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> |
```

DynamoDB

DynamoDB is a fully managed NoSQL database service that provides fast and predictable performance with seamless scalability. DynamoDB lets you offload the administrative burdens of operating and scaling a distributed database so that you don't have to worry about hardware provisioning, setup and configuration, replication, software patching, or cluster scaling.

- NoSQL databases are non relational databases and are distributed.
- NoSQL databases are MongoDB, DynamoDB etc..
- NoSQL databases do not support join
- All the data that is needed for a query is present in one row.
- NoSQL databases don't perform aggregation operations like SUM.
- NoSQL database scale Horizontally

DynamoDB:

Fully managed with High available replication across 3AZ

- NoSQL database not a relational database.

- Scales to Massive workloads ,distributed database.
- Millions of request per second.
 - Fast and consistent in performance(low latency on retrieval)
- Enable event driven programming with Dynamodb streams.
- Dynamodb is made up of tables.
 - Each table as a primary key must be decided at a creation time.
 - Table can have an infinite number of items each item attributes can be added over time (can be null)
- Maximum size of the item is 400 KB

Data types supported are

String

Boolean, boolean

Byte, byte

Date

Long, long

Integer, int

Double, double

Float, float

BigDecimal

BigInteger

- Dynamodb primary keys must be unique for each item.
- Partition key must be unique for each item.
- Example: **user-id** for a user table.

Type DynamoDB in AWS Console

Screenshot of the Amazon DynamoDB console homepage:

The page features a central logo and title: **Amazon DynamoDB**. Below the title is a brief description: "Amazon DynamoDB is a fast and flexible NoSQL database service for all applications that need consistent, single-digit millisecond latency at any scale. Its flexible data model and reliable performance make it a great fit for mobile, web, gaming, ad-tech, IoT, and many other applications." A prominent blue "Create table" button is located below the description. To the right of the main content area, there are three sections with icons and links:

- Create tables**: Shows an icon of two databases with a plus sign. Description: "Create DynamoDB tables with a few clicks. Just specify the desired read and write throughput for your table, and DynamoDB handles the rest." Link: [More about DynamoDB throughput](#)
- Add and query items**: Shows an icon of a database with a magnifying glass. Description: "Once you have created a DynamoDB table, use the AWS SDKs to write, read, modify, and query items in DynamoDB." Link: [DynamoDB API reference](#)
- Monitor and manage tables**: Shows an icon of a monitor displaying a graph with a checkmark. Description: "Using the AWS Management Console, you can monitor performance and adjust the throughput of your tables, enabling you to scale seamlessly." Link: [Monitoring tables](#)

- **Tables, attributes, and other objects in DynamoDB must have names.** Names should be meaningful and concise—for example, names such as *Employee*, *Payroll*
- You Must Specify at least one Primary Key and Data type.

Screenshot of the AWS DynamoDB "Create DynamoDB table" wizard:

Create DynamoDB table

DynamoDB is a schema-less database that only requires a table name and primary key. The table's primary key is made up of one or two attributes that uniquely identify items, partition the data, and sort data within each partition.

Table name*: Employee

Primary key*: Partition key

Emplid

String

Add sort key

Table settings

Default settings provide the fastest way to get started with your table. You can modify these default settings now or after your table has been created.

Use default settings

- No secondary indexes.
- Provisioned capacity set to 5 reads and 5 writes.
- Basic alarms with 80% upper threshold using SNS topic "dynamodb".
- Encryption at Rest with DEFAULT encryption type.

Info You do not have the required role to enable Auto Scaling by default.
Please refer to documentation.

- Add tags (1) NEW!

Key	Value
Employee	Ename
Add key	Empty value

Tags can contain unicode letters, spaces, digits, and these special characters _ ./+=-@. Max 50 tags. Learn more

Additional charges may apply if you exceed the AWS Free Tier levels for CloudWatch or Simple Notification Service. Advanced alarm settings are available in the CloudWatch management console.

•Specify Read and Write Capacity mode

The screenshot illustrates the process of creating a DynamoDB table and interacting with an existing one.

Create DynamoDB table (Left Panel):

- Table name***: Employee
- Primary key***: Partition key
Empid (String)
- Add sort key

Table settings:

Default settings provide the fastest way to get started with your table. You can modify these default settings now or after your table has been created.

Use default settings

Secondary indexes:

Name	Type	Partition key	Sort key	Projected Attributes
+ Add index				

Read/write capacity mode:

Select on-demand if you want to pay only for the read and writes you perform, with no capacity planning required. Select provisioned to save on throughput costs if you can reliably estimate your application's throughput requirements. See the [DynamoDB pricing page](#) and [DynamoDB Developer Guide](#) to learn more.

Read/write capacity mode can be changed later.

Provisioned (free-tier eligible)
 On-demand

DynamoDB Dashboard (Bottom Left):

- Tables
- Backup
- Reserved capacity
- Preferences

Users Table Overview (Bottom Right):

- Overview
- Items
- Metrics
- Alarms
- Capacity
- Indexes
- Global Tables
- Backups
- Triggers
- Access control
- Tags

Scan: [Table] Users: user_id

An item consists of one or more attributes. Each attribute consists of a name, a data type, and a value. When you read or write an item, the only attributes that are required are those that make up the primary key. [More info](#)

The screenshot shows the AWS DynamoDB console. On the left, the navigation menu includes 'DynamoDB', 'Dashboard', 'Tables', 'Backups', 'Reserved capacity', 'Preferences', 'DAX', 'Dashboard', 'Clusters', 'Subnet groups', 'Parameter groups', and 'Events'. The main area displays the 'Users' table. At the top, there are 'Create table' and 'Delete table' buttons, and a search bar labeled 'Filter by table name' with 'Users' selected. Below this is the 'Stream details' section, which shows 'Stream enabled' is 'No', 'View type' is 'Latest stream ARN', and a 'Manage Stream' button. The 'Table details' section provides comprehensive information about the table, including:

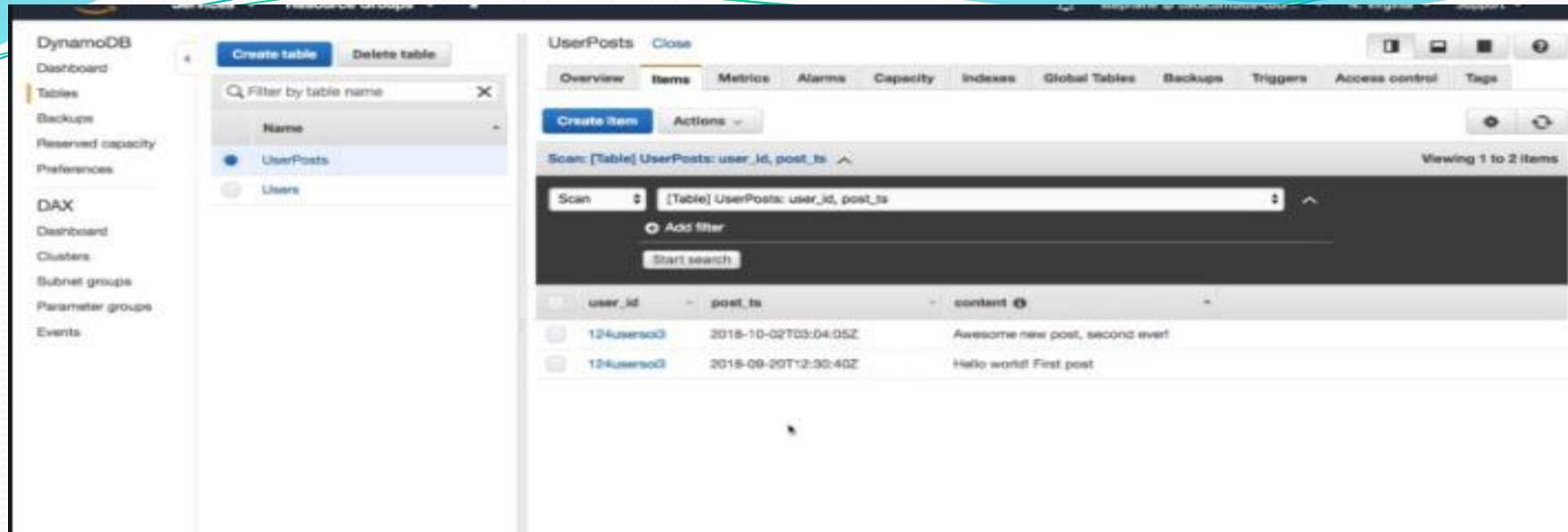
- Table name:** Users
- Primary partition key:** user_id (String)
- Primary sort key:** -
- Point-in-time recovery:** DISABLED
- Encryption:** Enabled
- Time to live attribute:** -
- Table status:** Active
- Creation date:** October 4, 2018 at 9:54:01 AM UTC+2
- Provisioned read capacity units:** 2 (Auto Scaling Disabled)
- Provisioned write capacity units:** 2 (Auto Scaling Disabled)
- Last decrease time:** -
- Last increase time:** -
- Storage size (in bytes):** 0
- Item count:** 0
- Region:** US East (N. Virginia)
- Amazon Resource Name (ARN):** arn:aws:dynamodb:us-east-1:387124123361:table/Users

A note at the bottom states: "Storage size and item count are not updated in real-time. They are updated periodically, roughly every six hours."

- Click Create Item tab and Select the Data type and Provide Attribute Name.

The screenshot shows the 'Create item' dialog box. The left sidebar of the main console is visible, showing the same navigation menu as the previous screenshot. The dialog box has a title 'Create item' and contains a tree view of item attributes:

- * Item (3)
 - user_id String + `12345678901234567890123456789012`
 - first_name String + `Stephanie`
 - last_name String + `Haasak`



RCU(Read Capacity Unit): unit of read capacity has been defined as 1 strongly consistent read per second for an item as large as 4KB.

WCU (Write Capacity Unit): unit of write capacity represents 1 write per second for an item as large as 1KB. The read operations will be rounded up to the next 4KB

DynamoDB

Dashboard

Tables

Backup

Reserved capacity

Preferences

DAX

Dashboard

Clusters

Subnet groups

Parameter groups

Events

UserPosts

Create table

Delete table

Filter by table name

Name

UserPosts

Users

Overview

Items

Metrics

Alarms

Capacity

Indexes

Global Tables

Backup

Triggers

Access control

Tags

Scaling activities

Provisioned capacity

Read capacity units

Table 3

Write capacity units

2

Estimated cost \$1.17 / month (Capacity calculator)

Auto Scaling

Read capacity

Write capacity

Save Cancel

Capacity calculator

Avg. item size 1 KB

Item read/sec 1 Eventually consistent

Item write/sec 1

Read capacity 1

Write capacity 1

Estimated cost \$0.59 / month per item/index

Update

This screenshot shows the AWS DynamoDB console for the UserPosts table. The Capacity tab is selected. The provisioned capacity is set to 3 read and 2 write capacity units. The auto-scaling section has 'Write capacity' selected. A modal window titled 'Capacity calculator' provides detailed breakdowns for different item sizes and usage metrics, showing estimated monthly costs.

DynamoDB

Dashboard

Tables

Backup

Reserved capacity

Preferences

DAX

Dashboard

Clusters

Subnet groups

Parameter groups

Events

UserPosts

Create table

Delete table

Filter by table name

Name

UserPosts

Users

Overview

Items

Metrics

Alarms

Capacity

Indexes

Global Tables

Backup

Triggers

Access control

Tags

Provisioned capacity

Read capacity units

Table 8

Write capacity units

8

Estimated cost \$1.17 / month (Capacity calculator)

Auto Scaling

Read capacity

Write capacity

Same settings as read

Target utilization 70 %

Minimum provisioned capacity 8 units

Maximum provisioned capacity 40000 units

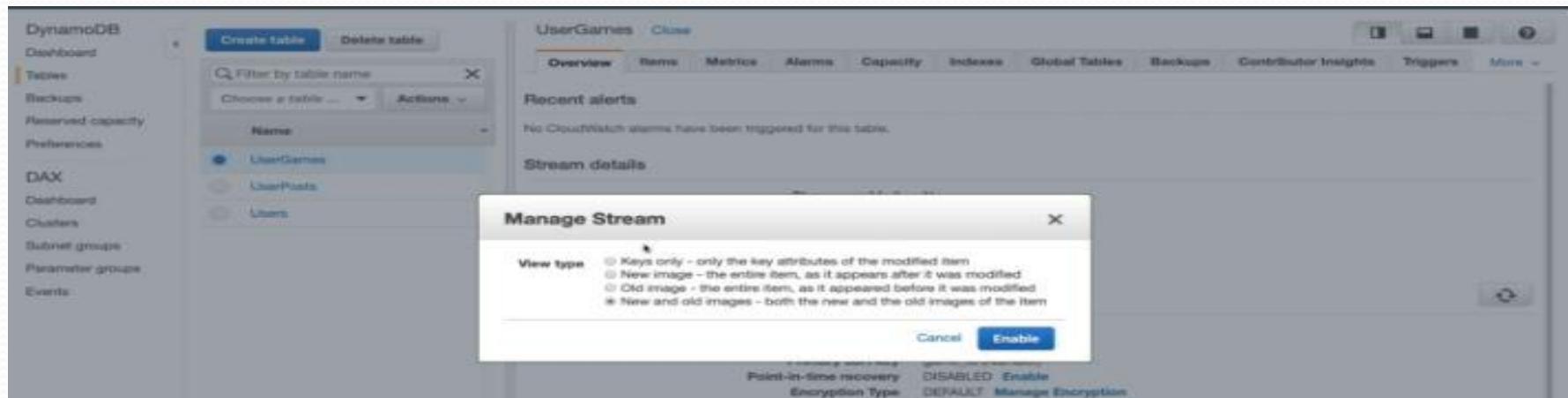
Apply same settings to global secondary indexes

Apply same settings to global secondary indexes

Please check your IAM permissions to create new service linked role for enabling Auto Scaling. See permissions.

This screenshot shows the AWS DynamoDB console for the UserPosts table. The Capacity tab is selected. The provisioned capacity is set to 8 units each for read and write. The auto-scaling section has 'Read capacity' selected. A note at the bottom states: 'Please check your IAM permissions to create new service linked role for enabling Auto Scaling. See permissions.'

DynamoDB Streams captures a time-ordered sequence of item-level modifications in any DynamoDB table and stores this information in a log for up to 24 hours. Applications can access this log and view the data items as they appeared before and after they were modified, in near-real time.

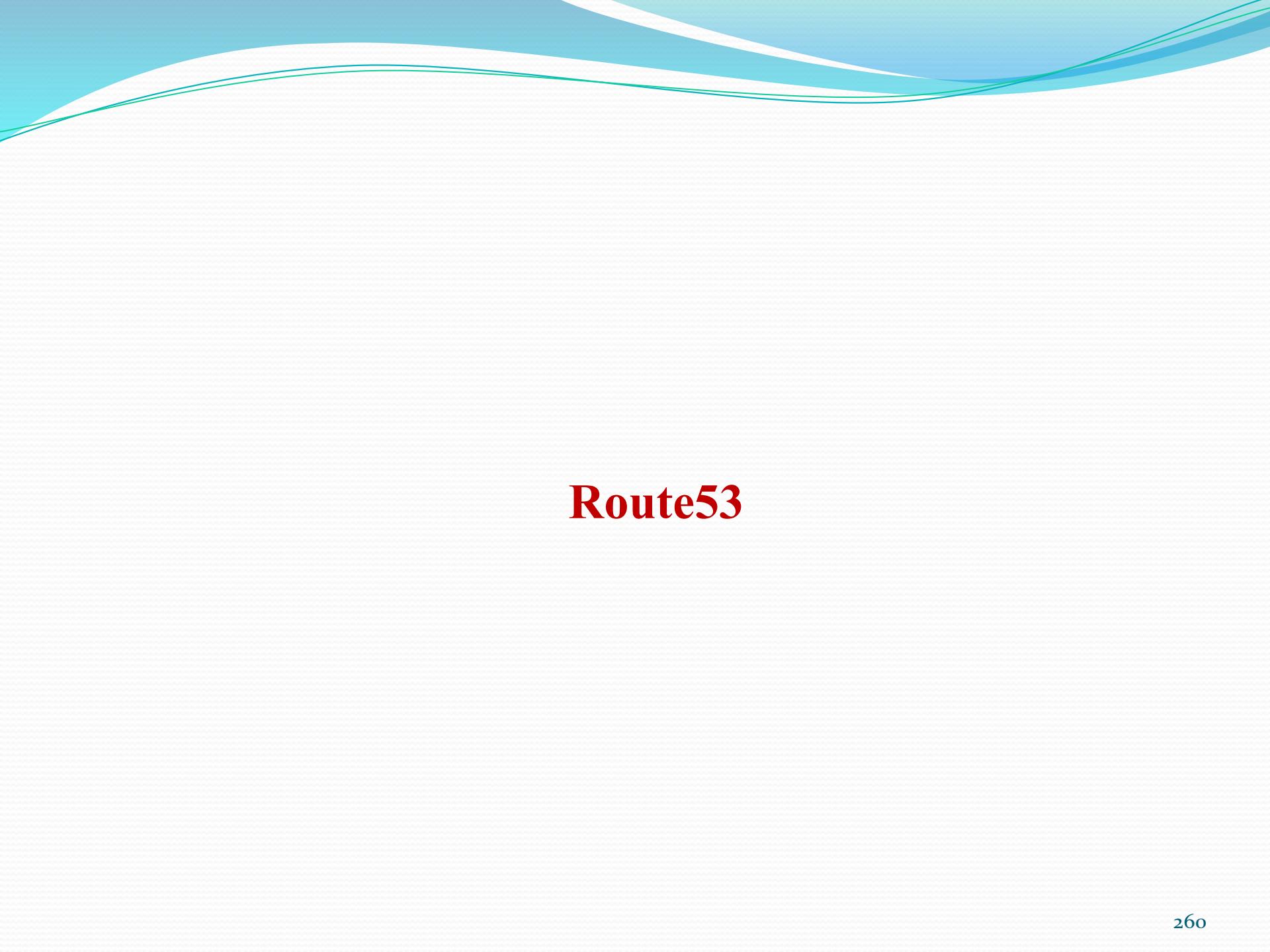


Keys only — Only the key attributes of the modified item.

New image — The entire item, as it appears after it was modified.

Old image — The entire item, as it appeared before it was modified.

New and old images — Both the new and the old images of the item.



Route53

Route53

Amazon Route 53 is a highly available and scalable Domain Name System (DNS) web service.

We 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 website, known as a domain name. our 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 .cricket. 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.

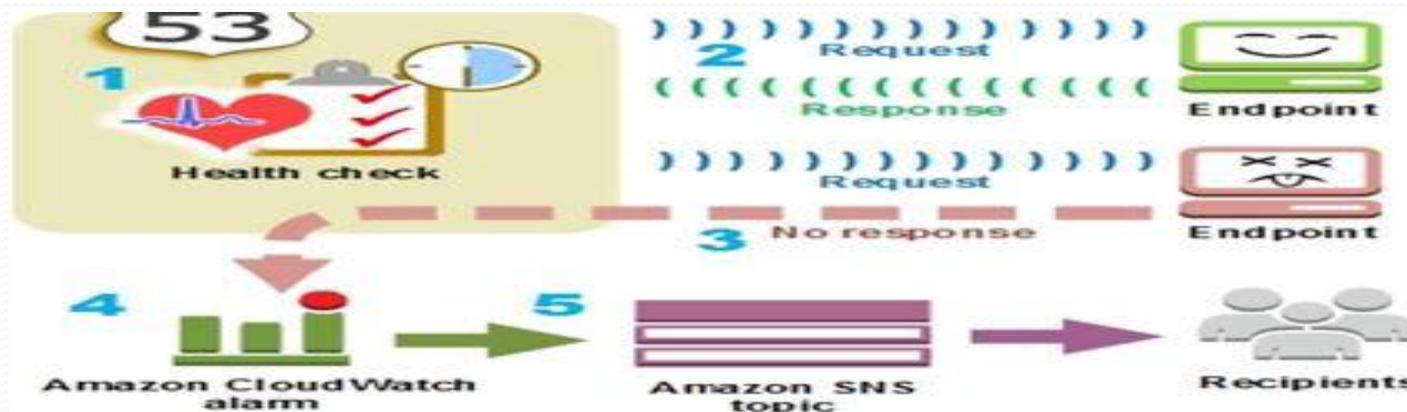
Here's an overview of how to use the Amazon Route 53 console to register a domain name and configure Route 53 to route internet traffic to your website or web application.

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.

Amazon Route 53 Checks the Health of Your Resources

Amazon Route 53 health checks monitor the health of your resources such as web servers or mail servers. You can optionally configure Amazon Cloud Watch 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:



Use Case:

- 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.

aws Services Resource Groups

shyam Global Support

DNS management: 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. [Learn More](#) [Get started now](#)

Traffic management: Route 53 traffic flow provides a visual tool that you can use to create and update sophisticated routing policies to route end users to multiple endpoints for your application. [Learn More](#) [Get started now](#)

Availability monitoring: Route 53 can monitor the health and performance of your application as well as your web servers and other resources. Route 53 can also redirect traffic to healthy resources. [Learn More](#) [Get started now](#)

Domain registration: If you need a domain name, you can find an available name and register it by using Route 53. You can also make Route 53 the registrar for existing domains that you registered with other registrars. [Learn More](#) [Get started now](#)

Route 53 documentation and support
[Route 53 documentation](#) | [FAQs](#) | [Forum - DNS and health checks](#) | [Forum - Domain name registration](#) | [Support](#)

Amazon Route 53 is an authoritative Domain Name System (DNS) service. DNS is 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

The screenshot shows the AWS Route 53 service page. The left sidebar has a 'Hosted zones' section selected. The main area features a large icon of a computer monitor with a cloud above it. Below the icon, text explains what Amazon Route 53 is: "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." A note below states: "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." There is a "Learn More" link and a "Create Hosted Zone" button.

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 rote 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.

The screenshot shows the AWS Route 53 service interface. On the left, a sidebar lists various services: Dashboard, Hosted zones (which is selected and highlighted in yellow), Health checks, Traffic flow, Traffic policies, Policy records, Domains, Registered domains, Pending requests, Resolver, VPCs, Inbound endpoints, Outbound endpoints, and Rules. The main area displays a search bar with 'Search all fields' and a dropdown menu set to 'All Types'. Below the search bar are filters for 'Domain Name', 'Type', 'Record Set Count', 'Comment', and 'Hosted Zone ID'. A message states 'You have no hosted zones'. To the right, a modal window titled 'Create Hosted Zone' is open. It contains fields for 'Domain Name' (with placeholder 'example.com'), 'Comment' (empty), 'Type' (set to 'Private Hosted Zone for Amazon VPC'), a description of what a private hosted zone is, 'VPC ID' (set to 'vpc-0f7ab0fd | us-east-1'), and an 'Important' note: 'To use private hosted zones, you must'. A blue 'Create' button is at the bottom right of the modal.

Click on Create Record Set and Select previously created record sets 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.

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

The screenshot shows the AWS Route 53 service interface. On the left, there's a sidebar with various navigation options. The main area displays a list of record sets for a specific hosted zone. The right side has a detailed view of one record set, allowing for editing its TTL and value.

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

```
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-172-31-93-192:~$ wget http://172.31.93.192
--2019-05-06 05:31:36-- http://172.31.93.192/
Connecting to 172.31.93.192:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 11321 (11K) [text/html]
Saving to: 'index.html'

index.html          100%[=====]  11.06K  --.-KB/s   in 0s

2019-05-06 05:31:36 (382 MB/s) - 'index.html' saved [11321/11321]
ubuntu@ip-172-31-93-192:~$ |
```

The screenshot shows the AWS Route 53 Health Checks console. The left sidebar includes links for Dashboard, Hosted zones, Health checks (which is selected), Traffic flow, Traffic policies, Policy records, Domains, Registered domains, Pending requests, Resolver, VPCs, Inbound endpoints, and Outbound endpoints. The main content area has a title 'Welcome to Route 53 health checks' and a sub-section 'Health check concepts'. It features two cards: 'Availability and performance monitoring' (with an icon of a computer monitor showing a checkmark) and 'DNS failover' (with an icon of a shield containing a stethoscope and a plus sign). Both cards have a 'Learn more' link below them.

Click on create Health check
Configure Health check

Name: name for health check(Web health check)

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)

Create alarm: select no or do you want make any alarms you can 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

The screenshot shows a user interface for managing health checks. On the left, there's a sidebar with various navigation options like Dashboard, Hosted zones, Health checks (which is selected), Traffic flow, Traffic policies, Policy records, Domains, Registered domains, Pending requests, Resolver, vPCs, Inbound endpoints, Outbound endpoints, and Rules. The main area has a success message: "Health check with id 051b551b-7d77-4278-8b08-3eddc952462c has been created successfully". Below this, there are buttons for "Create health check", "Delete health check", and "Edit health check". A table lists one health check entry:

Name	Status	Description	Alarms	ID
051b551b-7d77-4278-8b08-3eddc952462c	Healthy	http://52.205.89.174:80/Index	1 of 1 in OK	051b551b-7d77-4278-8b08-3eddc952462c

At the bottom, there's a detailed view of the health check configuration with tabs for Info, Monitoring, Alarms, Tags, Health checkers, and Uptime. The Info tab shows the ID (051b551b-7d77-4278-8b08-3eddc952462c), URL (http://52.205.89.174:80/Index), and Protocol (HTTP). The Advanced configuration section includes Request interval (30 seconds), Failure threshold (3), and Search string.

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

System Resource Utilization Report												
Process Details		Memory Usage			CPU Activity		Disk I/O				Network Status	
PID	User	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND	
20078	ubuntu	20	0	40488	3688	3128	R	0.3	0.4	0:00.01	top	
1	root	20	0	119604	5832	4088	S	0.0	0.6	0:04.37	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	0:00.14	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	0:00.33	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	0:00.05	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	
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-172-31-93-192:~$ sudo service apache2 stop
```

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

Routing Policy in Route53:

Simple routing policy – Use for a single resource that performs a given function for your domain, for example, a web server that serves content for the example.com website.

Failover routing policy – Use when you want to configure active-passive failover.

Geolocation routing policy – Use when you want to route traffic based on the location of your users.

Geoproximity routing policy – Use when you want to route traffic based on the location of your resources and, optionally, shift traffic from resources in one location to resources in another.

Latency routing policy – Use when you have resources in multiple AWS Regions and you want to route traffic to the region that provides the best latency.

Multivalue answer routing policy – Use when you want Route 53 to respond to DNS queries with up to eight healthy records selected at random.

Weighted routing policy – Use to route traffic to multiple resources in proportions that you specify.

AWS- CloudFormation

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 resources.

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**. 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.

JSON:

```
{  
  "AWSTemplateFormatVersion" : "2010-09-09",  
  "Description" : "A sample template",  
  "Resources" : {  
    "MyEC2Instance" : {  
      "Type" : "AWS::EC2::Instance",  
      "Properties" : {  
        "ImageId" : "ami-off8a91507f77f867",  
        "InstanceType" : "t2.micro",  
        "KeyName" : "testkey",  
        "BlockDeviceMappings" : [  
          {  
            "DeviceName" : "/dev/sdm",
```

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

YAML

```
AWSTemplateFormatVersion: "2010-09-09"
```

```
Description: A sample template
```

```
Resources:
```

```
MyEC2Instance:
```

```
Type: "AWS::EC2::Instance"
```

```
Properties:
```

```
ImageId: "ami-off8a91507f77f867"
```

```
InstanceType: t2.micro
```

```
KeyName: testkey
```

```
BlockDeviceMappings:
```

```
-
```

```
    DeviceName: /dev/sdm
```

```
    Ebs:
```

```
        VolumeType: io1
```

Iops: 200

DeleteOnTermination: false

Volume Size: 20

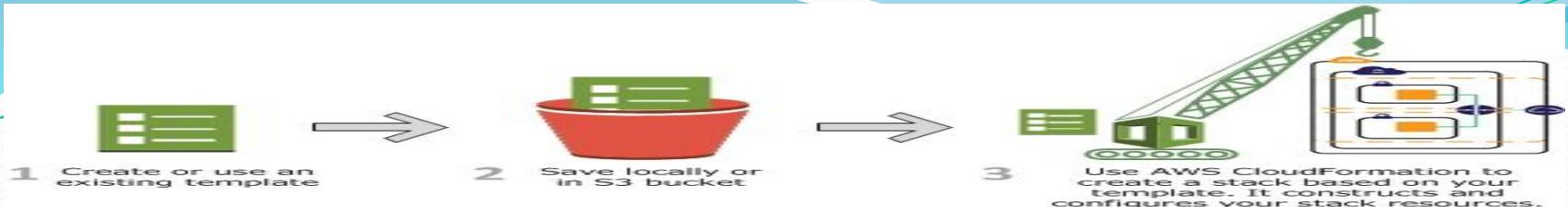
Stacks

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.



You can modify an AWS CloudFormation stack template by using AWS

1. 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.

Use case:

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:

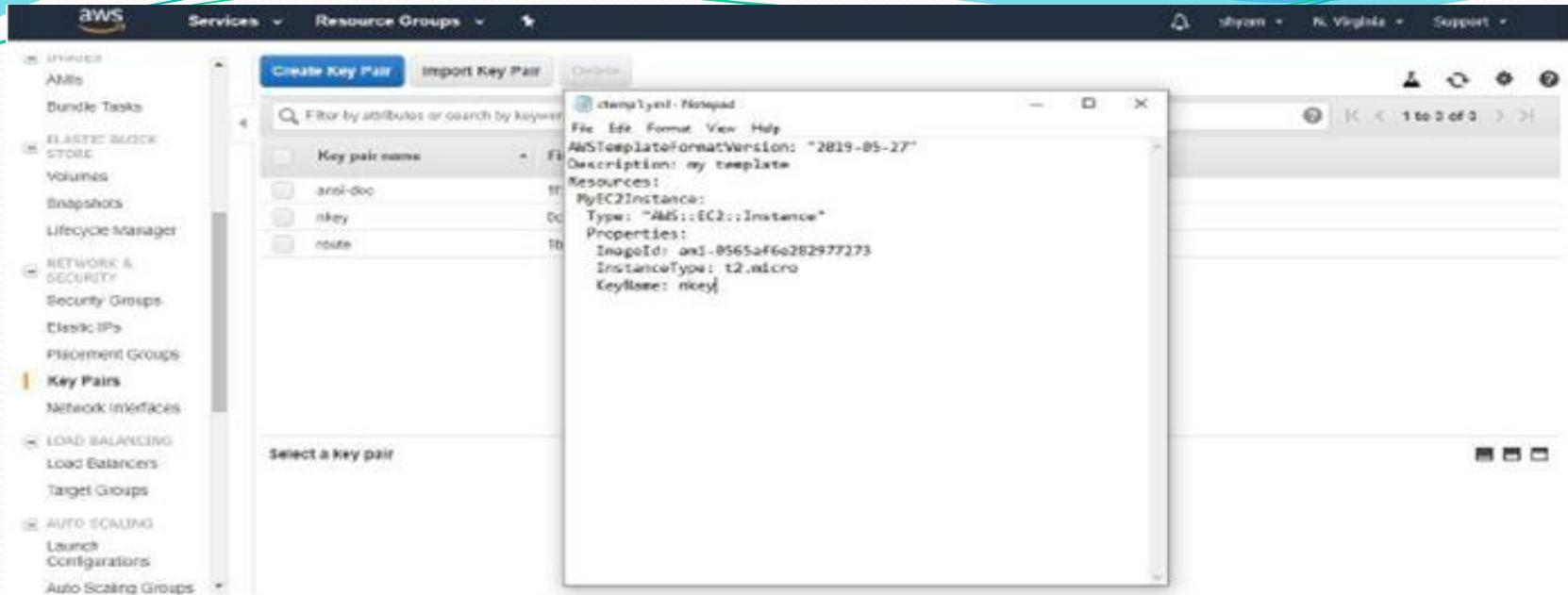
AvailabilityZone:us-east-1a

ImageId: "ami-off8a91507f77f867"

InstanceType: t2.micro

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
- With in the resource section you mention resource type and it's 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 cloudformation 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

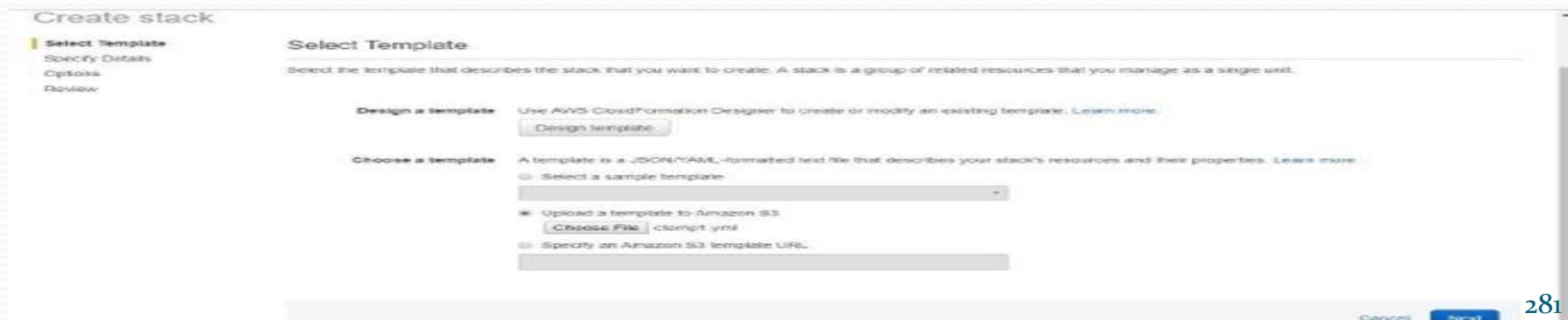
In this section we have 3 types to choose a file.

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.

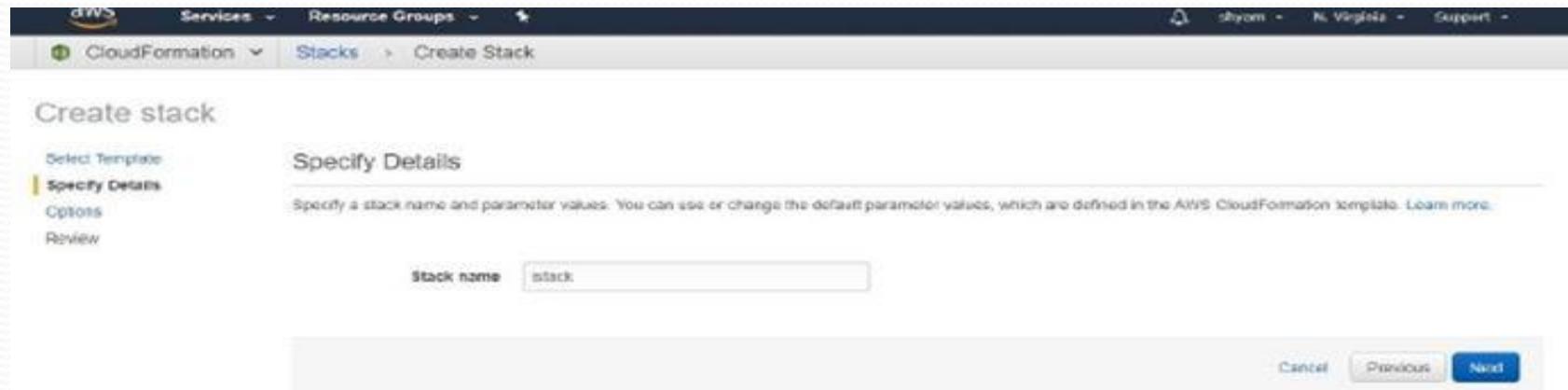
Specify an Amazon s3 template URL: choose template from Amazon s3 bucket

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 go to 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

Select Template
Specify Details
Options
Review

Options

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.

Key	(1-27 characters, maximum)	Value	(255 characters maximum)
1			

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)
Either 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.

Select Template
Specify Details
Options
Review

Review

Template

Template URL: <https://s3.amazonaws.com/cf-templates-registry/us-east-1/2019147d8A-cwmp1.yaml>
Description:
Estimate cost: Cost

Details

Stack name: istack

Options

Tags

No tags provided

Rollback Triggers

No monitoring time provided

No rollback triggers provided

Advanced

Notification: Disabled
Termination Protection: Enabled
Timeout: 60s

Click on create

The screenshot shows the AWS CloudFormation console interface. At the top, there are navigation links for Services, Resource Groups, and a dropdown for regions (Oregon - N. Virginia). Below the header, a banner announces the availability of the redesigned console. Another message highlights recent improvements and new regions available. The main area features a table listing stacks. A single row is selected, showing details: Stack Name (iStack), Created Time (2019-05-27 16:20:39 UTC+0550), Status (CREATE_IN_PROGRESS), Drift Status (NOT_CHECKED), and Description (empty). The table has columns for Stack Name, Created Time, Status, Drift Status, and Description.

Stack Name	Created Time	Status	Drift Status	Description
iStack	2019-05-27 16:20:39 UTC+0550	CREATE_IN_PROGRESS	NOT_CHECKED	

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

Over 100,000 AWS users have already tried out CloudFormation's new visual editor. We're excited to share the results of your feedback. The changes include a new layout for faster access to information, resizable columns, and availability in 5 additional regions (AWS GovCloud (US-West and US-East), China (Beijing), China (Ningbo), EU (Stockholm)). Please tell us what you think!

Create StackActionsDesign template

Filter: Activeby Stack NameShowing 1 stack

Stack Name	Created Time	Status	Drift Status	Description
istack	2019-05-27 16:20:39 UTC+0550	CREATE_COMPLETE	NOT_CHECKED	

OverviewOutputsResourcesEventsTemplateParametersTagsStack PolicyChange SetsRollback Triggers

Filter by:StatusSearch events

2019-05-27	Status	Type	Logical ID	Status Reason
16:21:41 UTC+0550	CREATE_COMPLETE	AWS::CloudFormation::Stack	istack	
16:21:39 UTC+0550	CREATE_COMPLETE	AWS::EC2::Instance	MyEC2Instance	
16:20:47 UTC+0550	CREATE_IN_PROGRESS	AWS::EC2::Instance	MyEC2Instance	Resource creation initiated
16:20:45 UTC+0550	CREATE_IN_PROGRESS	AWS::EC2::Instance	MyEC2Instance	
16:20:39 UTC+0550	CREATE_IN_PROGRESS	AWS::CloudFormation::Stack	istack	User Initiated

AWS-CLI(Command Line Interface)

CLI

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

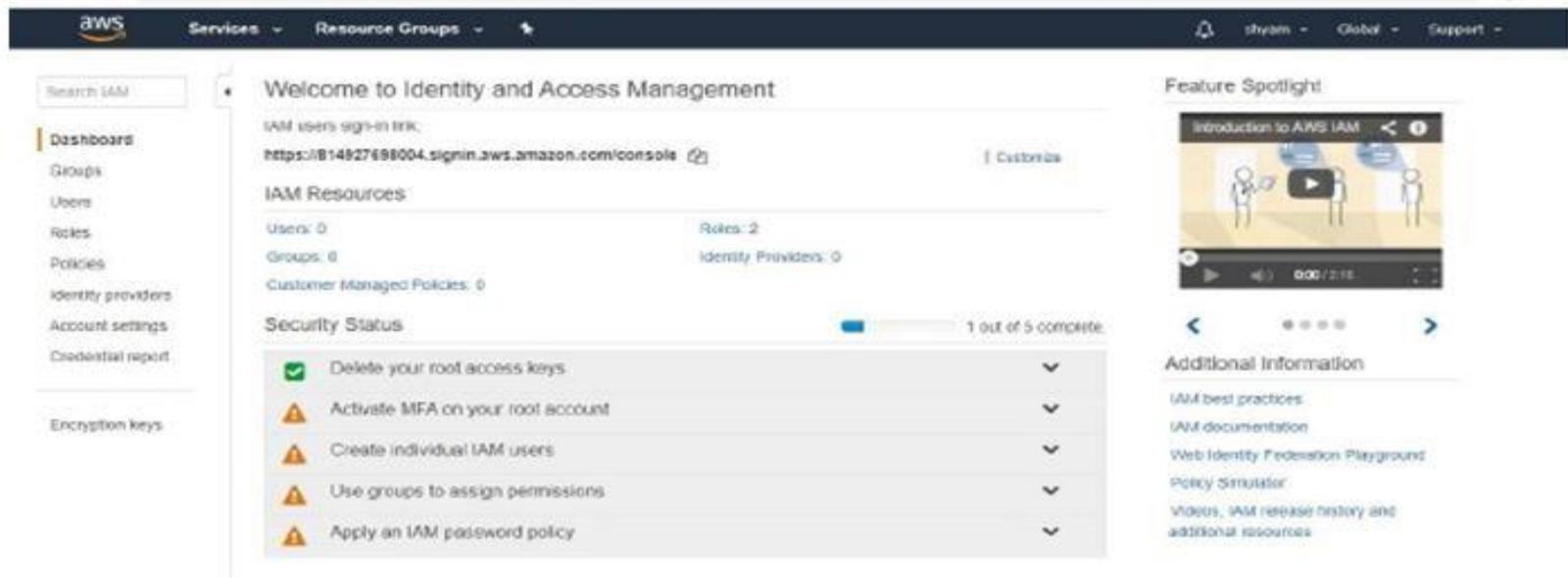
Remotely – Run commands on Amazon Elastic Compute Cloud (Amazon EC2) instances through a remote terminal such as PuTTY or SSH, or with AWS Systems Manager.

Install the AWS CLI on Amazon Linux:

- Launch EC2 Instance and connect to that EC2 Instance using putty or git bash
- Install the AWS CLI on our EC2 Instance by using below command.

sudo apt-get install -y awscli

- Configure your aws cli using (**aws configure**) command
- We need aws access key and aws secret access key and region to configure
- First choose IAM service from aws console.



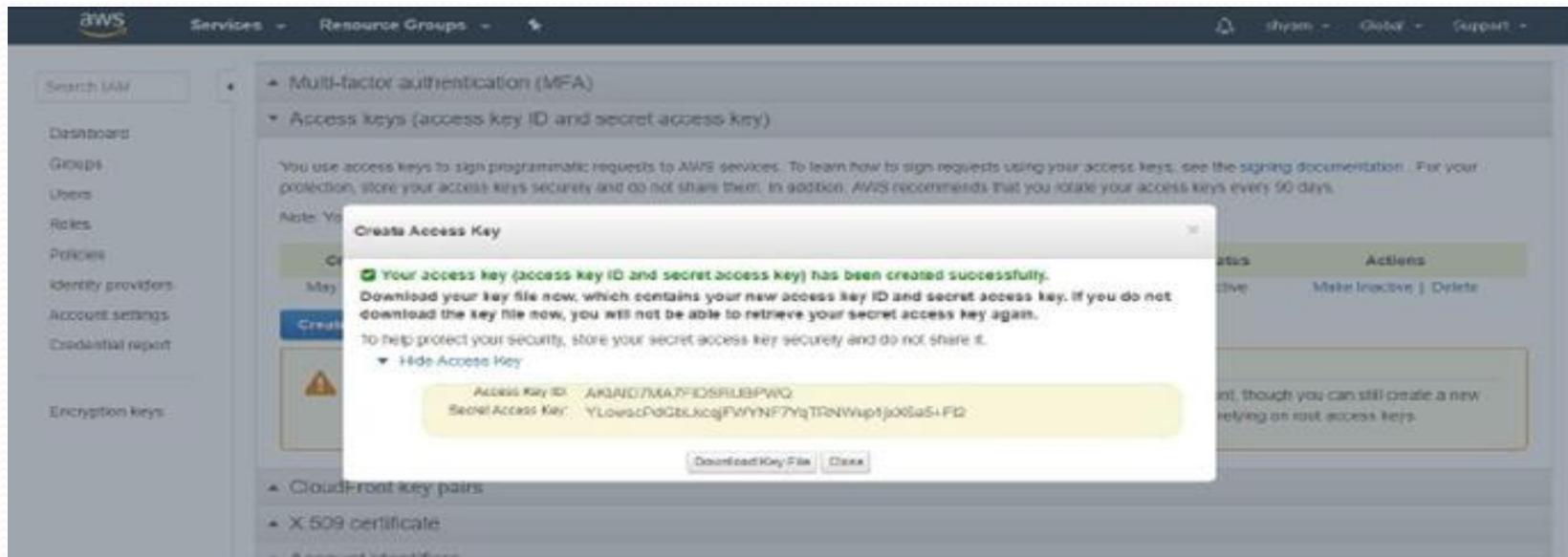
Click on delete your root access keys and click on manage security credentials

The screenshot shows the AWS Identity and Access Management (IAM) dashboard. On the left, there's a sidebar with links like 'Dashboard', 'Groups', 'Users', 'Roles', 'Policies', 'Identity providers', 'Account settings', 'Credential report', and 'Encryption keys'. The main area has a heading 'Welcome to Identity and Access Management' and a 'Feature Spotlight' section with a video thumbnail. Below these are sections for 'IAM Resources' (User: 0, Roles: 2, Groups: 0, Identity Providers: 0), 'Customer Managed Policies: 0', and 'Security Status' (1 out of 5 complete). A prominent call-to-action box contains a checked checkbox labeled 'Delete your root access keys', followed by a note about the risks of using root keys and a 'Manage Security Credentials' button. Below this are two dropdown menus: one for 'Activate MFA on your root account.' and another for 'Create individual IAM users.'

Click on continue security credentials and click on access keys and click on create New access keys

The screenshot shows the 'Access Keys' section of the AWS IAM service. The sidebar on the left includes 'Dashboard', 'Groups', 'Users', 'Roles', 'Policies', 'Identity providers', 'Account settings', 'Credential report', and 'Encryption keys'. The main content area has a heading 'Multi-factor authentication (MFA)' and 'Access keys (access key ID and secret access key)'. It includes a note about using access keys for programmatic requests and a warning about the maximum of two active access keys. A table lists existing access keys with columns: Created (May 26, 2019), Deleted (N/A), Access Key ID (AKIAJULQKQDSINMACI7AA), Last Used (N/A), Last Used Region (N/A), Last Used Service (N/A), Status (Active), and Actions (Edit, Inactive | Delete). Below the table is a box titled 'Important Change - Managing Your AWS Secret Access Keys' with a note about the inability to retrieve existing secret access keys for the root account. At the bottom, there are links for 'CloudFront key pairs', 'X.509 certificate', and 'Amazon VPC endpoints'.

Click on show keys and copy that key



Then use the following 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 acesskey
- **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 ~
  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]: AKIAID7MA7FIDSRUBPWQ
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:~|

```

Use case for S3 service:

To see the list of buckets in S3 service use below command

aws s3 ls

Bucket	Region	Date Created
CD106	Asia Pacific (Singapore)	May 9, 2019 11:05:46 AM (GMT+0530)
my08021	Asia Pacific (Singapore)	May 9, 2019 11:05:46 AM (GMT+0530)

To create buckets in s3 use the mb (make bucket) command

aws s3 mb s3://bucketname

*Here bucket name should be unique

A screenshot of a Microsoft Word document titled "AWSsoft-copy - Microsoft Word". The document contains a table titled "Kib Swap" showing system memory usage. Below the table, two terminal command-line outputs are shown:

```
ubuntu@ip-172-31-31-211:~$ aws s3 mb help
ubuntu@ip-172-31-31-211:~$ aws s3 mb s3://bbcc321
make_bucket: bbcc321
```

To remove bucket from s3

aws s3 rb s3://bucketname

A screenshot of a terminal window showing system memory usage and AWS CLI command output. The terminal shows the following:

```
ubuntu@ip-172-31-31-211:~$ aws s3 rb s3://bbcc321
remove_bucket: bbcc321
ubuntu@ip-172-31-31-211:~$
```

To upload files to s3 bucket

aws s3 cp sourcefile/upload_data s3://bucketname/destination

```
ubuntu@ip-172-31-0-16:~$ cat > sample123
he;ll
how are you
ubuntu@ip-172-31-0-16:~$ aws s3 sample123 s3://mybb321
usage: aws [options] <command> <subcommand> [<subcommand> ...] [parameters]
To see help text, you can run:
aws help
aws <command> help
aws <command> <subcommand> help
aws: error: argument subcommand: invalid choice, valid choices are:
ls                                | website
cp                                | mv
rm                                | sync
mb                                | rb
presign
ubuntu@ip-172-31-0-16:~$ aws s3 cp sample123 s3://mybb321
upload: ./sample123 to s3://mybb321/sample123
```

To download files from s3 bucket use cp command in reverse order

aws s3 cp s3://bucketname/file destination/hostlocation

```
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
```

To delete files from s3 bucket using following command

`aws s3 rm s3://bucketname/objectname`

```
nagesh@nagesh-PC:~$ 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:~$
```

Use case 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`

```

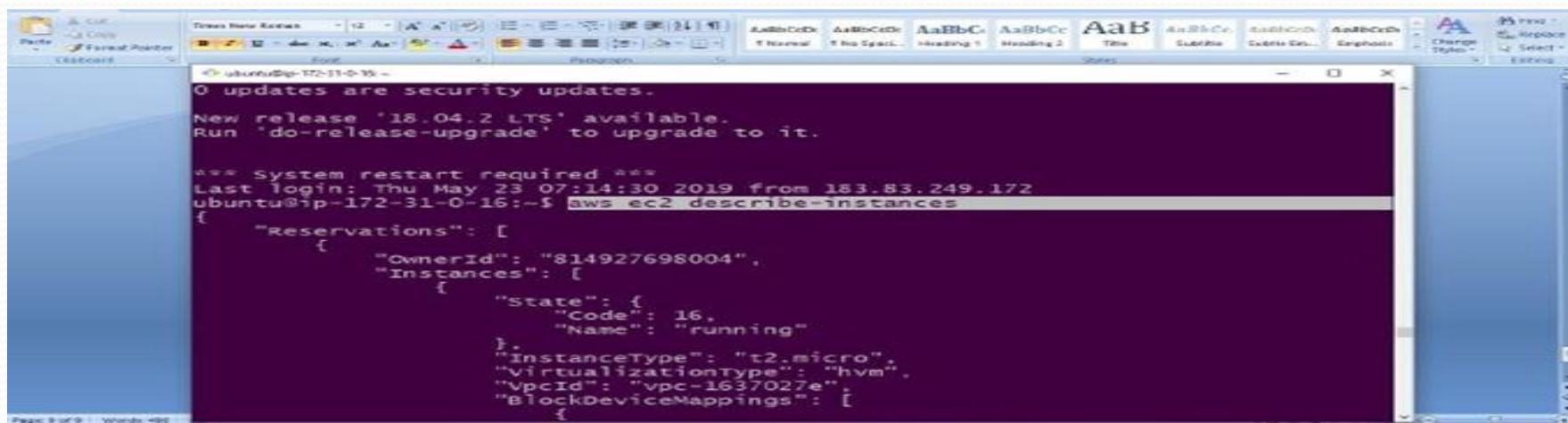
aws help
aws <command> help
aws <command> <subcommand> help

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",

```

To see the list of instances through aws cli use the below command

`aws ec2 describe-instances`



The screenshot shows a Windows terminal window with the title bar 'Ubuntu@ip-172-31-0-16 ~'. The window contains the following text:

```

0 updates are security updates.

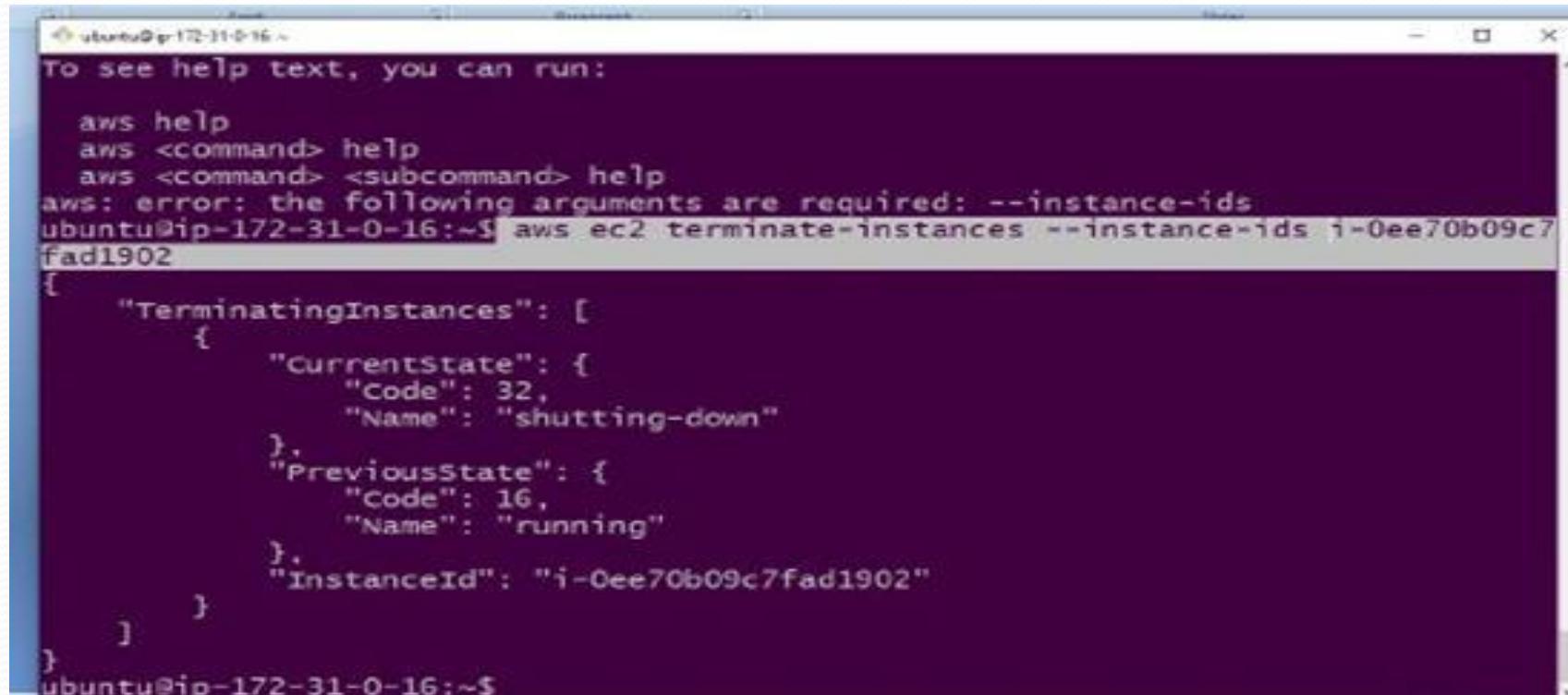
New release '18.04.2 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

*** System restart required ***
Last login: Thu May 23 07:14:30 2019 from 163.83.249.172
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": [

```

To terminate the instance through aws cli use below command

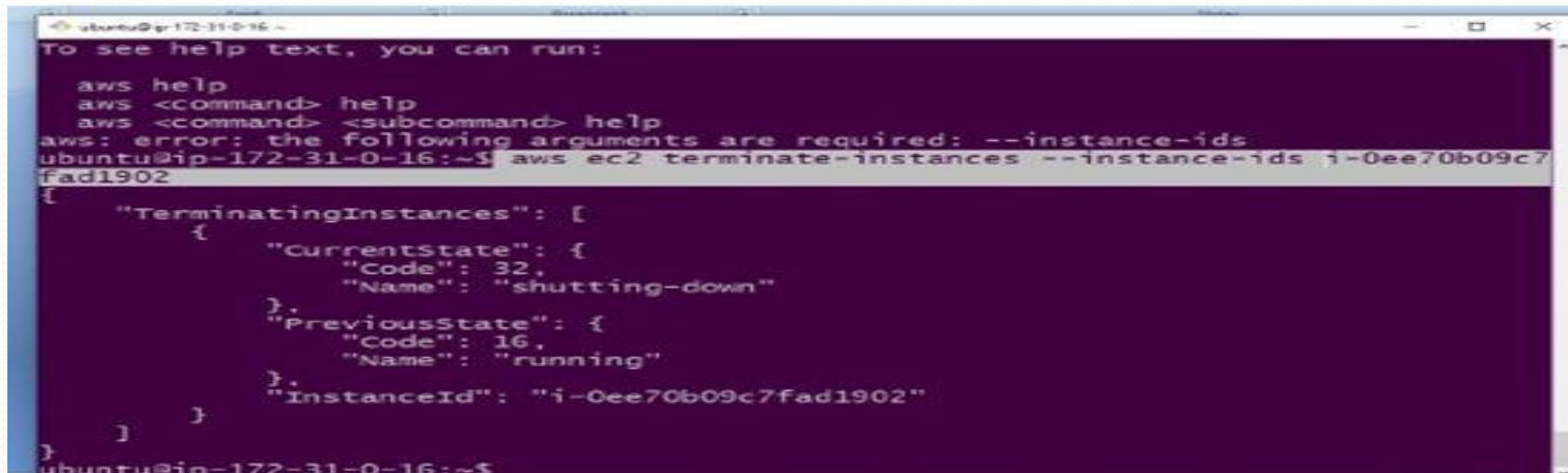
aws ec2 terminate-instances --instance-ids instanceid



```
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:~$
```

To terminate the instance through aws cli use below command

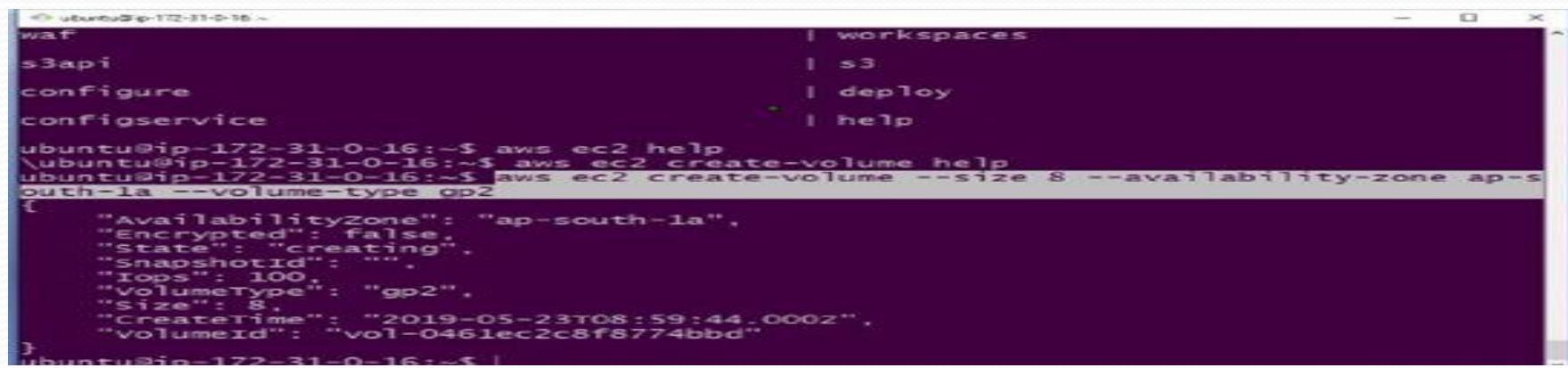
aws ec2 terminate-instances --instance-ids instanceid



```
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:~$
```

To create volume

aws ec2 create-volume --size 8 --availability-zone ap-south-1a --volume-type gp2



```
ubuntu@ip-172-31-0-16:~$ waf
| workspaces
s3api
configure
configservice
ubuntu@ip-172-31-0-16:~$ aws ec2 help
| s3
| deploy
| 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 *instanc eid* --device /dev/sdf~~

```
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 create key pair

aws ec2 create-key-pair --key-name keypairname

```
ubuntu@ip-172-31-0-16:~$ aws ec2 create-key-pair --key-name mykey
aws: error: the following arguments are required: --key-name
ubuntu@ip-172-31-0-16:~$ {
    "KeyFingerprint": "19:27:20:df:5c:60:71:0e:44:76:54:48:ff:0e:s1:de:d6:a3:08:0b",
    "KeyName": "mykey",
    "KeyMaterial": "-----BEGIN RSA PRIVATE KEY-----\nMIIEpaIBAAKCAQEATB99BynL0wg\nnffve6kyozJvBL9+aHmbn3mE17k1MKF3Y2/z1u5xxz3EyHn59/n4knCP1NQeLa58m6FH64GvxapXubOx\nmVi1wcaFPox128y2m0k00SPritF95xtckR0YgGhixfzPRSD\n/nD1MDKU0j42oQzck+X4VqvoEgZozs6XX\n9J34pehHGaawy89YagVUFwRjo3she55+n0oAhFdFBu0pz\n/nvg5RKxxLd7LKFr4di1hkjfb2gg2lphh8+\nfg6cVc0iBCNyM9FgCJC5+rHA577cpwCbIwC615ku5\n/nUntuH5Kgo7qsmaOhYqnKJxfLuH7a59wmppP\nIunmoWr41/Oph4/i3rwIDAQABoIBAQCPdk+w3XNN\n/nCM62vx3d53u8NN01cGnjz+YwIZ/w3CYWmk4FD\nu2HtoMqrU7NgdTxhHe63zaaut5kXYWw2OrPxZL1\n/nJuY3+ucxs1mHV4kQETI/L2xT+2PuKxroxtxi/rn\nToaDBcsGC1orYHJ3DMkmRzvfa1ngQMzzgRE8N\n/nAw+w5+15i/avvxCOpvpb2TdBa4vetbAE5968eYL9x\nz4HAoyLzmhowJw9ZoT3od2BKMMz8efXUZCR\n/nBpZ0uuiuHmJfuH2xL3riTb7sv1AYPUVvj26TdRZGBIyLJ\noBAOmC/SPPy8InSVYFUG/0HQJQMTVx+w/\n/nBpuPiT3wcvL0F/xA7w66re2ioYBAoGBAOSiROd3VsdbL\nwamj1+ibbkM5PqNozdciJP91o6GYY9y\n/nHH2An35Low8Su5Qh1DrMuQy188DoqotycgvQ12P4INmGxUb\njmu/bq1bBko32svhZBnamsG9t3CT3\n/nvFL8unFLyvaPi44dc2rR/frKsVCX11fszDAvGgwP4+3T7SKmf\nQjfAoGBAKXaOrQrvLNhx3bwHmUr\n/n1qut6XF/o8WjycbVRp72njx3PEuxp7Tos5AU09xt6Qxpfe++Ky\nb7ieuIUngSxczzS9QB/pokFO1\n/nv4krsvoooc7KGnTuHrn1qxCLQoeruwyhaufVUE1EXHD6U7MPIxt+34\nTgcJrrq5ZjNUnRRqyCEhgkx\n/nAOGBAKMzh2Xo3OSnHR1efJCK9s+bmbot+QbPyVRthIM2P1vCrOFbfba4\nMRIT7hw04i5ZKNqd95QZ/z\n/nNAL1LIR5jdaJxwGHHPwmKgm/NmT1i9+w1uM31jp4ppgLQzqTWj+e1A6dd\n3m3e3mc0xdUEES9/PY3\n/nTalvYA2haanMifnRRjtut1/nAOGAKcd0zqhFOHYRcdEXXNE30mr xuPv5L7G\ne+zFLWFNHLL0mEns\n/nLFum1by1kpPaoAc26Ey/sgQJ+QeMobE/HofhmJncx3peORGxExbzOxyII6vkyc\n-----END RSA PRIVATE KEY-----"
```

To create security group

aws ec2 create-security-group --group-name groupname --vpc-id vpcid --description message

```
ubuntu@ip-172-31-0-16:~$ aws ec2 create-security-group help
ubuntu@ip-172-31-0-16:~$ aws ec2 create-security-group --group-name mysg --vpc-id vpc-1637027e
usage: aws [options] <command> <subcommand> [<subcommand> ...] [parameters]
To see help text, you can run:
aws help
aws <command> help
aws <command> <subcommand> help
aws: error: the following arguments are required: --description
ubuntu@ip-172-31-0-16:~$ aws ec2 create-security-group --group-name mysg --vpc-id vpc-1637027e --description mysg
{
    "GroupId": "sg-0712379f2d3d839fe"
}
ubuntu@ip-172-31-0-16:~$ |
```

To create VPC

aws ec2 create-vpc --cidr-block cidrblockrange(10.0.0.0/24)

```
ubuntu@ip-172-31-0-16:~$ unassign-private-ip-addresses | unmonitor-instances
ubuntu@ip-172-31-0-16:~$ 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
```

To create subnet

aws ec2 create-subnet -vpc-id vpcid -cidr-block subnetcidrblock

```
ubuntu@ip-172-31-0-16:~$ unassign-private-ip-addresses | unmonitor-instances
ubuntu@ip-172-31-0-16:~$ 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
```

To create VPC

aws ec2 create-vpc --cidr-block cidrblockrange(10.0.0.0/24)

```
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
```

To create subnet

aws ec2 create-subnet -vpc-id vpcid -cidr-block subnetcidrblock

```
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
```

ElastiCache

ElastiCache

AWS ElastiCache is a Web Service used to deploy, run and scale popular open source compatible in-memory data stores. It improves the performance of the existing apps by retrieving data from high throughput and low latency in-memory data stores. Amazon ElastiCache is a popular choice for Gaming, Ad-Tech, Financial Services, Healthcare, and IoT apps.

ElastiCache is a distributed cache environment for providing faster access to data by using cloud-based caching. Querying for data directly from databases or through remote API calls is much slower than querying the data from cache. AWS provides ElastiCache service which has high performance, scalability and cost-effectiveness.

Components of ElastiCache:

The ElastiCache provisioned by AWS has the following important components. We will see their configuration and use in the subsequent chapters.

- **Node** – A node is a fixed-size chunk of secure, network-attached RAM. Each cache node has its own Domain Name Service (DNS) name and port.
- **ElastiCache for Redis Shards** – It is a group of one to six related nodes. A cluster can have one to 90 shards.
- **ElastiCache for Redis Clusters** – A Redis cluster is a logical grouping of one or more ElastiCache for Redis Shards. Data is partitioned across the shards in a Redis (cluster mode enabled) cluster.
- **ElastiCache for Redis Endpoints** –It is a unique address your application uses to connect to an ElastiCache node or cluster.

Features of ElastiCache:

The various features of ElastiCache helps us to plan for the proper configuration and cost estimation. The important features are as below.

- **ElastiCache for Redis Replication** – Replication is implemented by grouping from two to six nodes in a shard. One of these nodes is the read/write primary node. All the other nodes are read-only replica nodes.
- **Regions and Availability Zones** – The cache can be built and used in any locations that meet your business requirements. The AWS ElastiCache service is available in multiple AWS regions worldwide.
- **ElastiCache Parameter Groups** – An ElastiCache parameter group is a named collection of engine-specific parameters that you can apply to a cluster. They are used to control memory usage, eviction policies and item sizes etc.
- **ElastiCache for Redis Security** – You can control the Amazon EC2 instances that can access your cluster by using subnet groups or security groups.

Installing and Accessing Memcached :

yum update -y

yum install telnet

Connecting Memcached: telnet <endpoint name> port number[11211]

set name o o 4

Ex:pune

get name

Installing and Accessing Redis:

yum install redis -y

yum install redis-cli

sudo yum install -y gcc

wget http://download.redis.io/redis-stable.tar.gz && tar xvzf redis-stable.tar.gz && cd

Connecting Redis: redis-cli -h <endpoint name> -p 6389

sudo cp src/redis-cli /usr/bin/

Ex: set name sourcefuse

get name