Abstract

Amazon Web Services is a collection of remote computing services or web services that together make up a cloud computing platform, offered over the Internet by Amazon. Many companies are moving away from traditional datacenters and toward AWS because of its reliability, service offerings, low costs, and high rate of innovation. Because of its versatility and flexible design, AWS can be used to accomplish a variety of simple and complicated tasks such as hosting multitier websites, running large scale parallel processing, content delivery, petabyte storage and archival, and lot more.

This AWS Administration Guide is prepared by **Mr. Avinash Reddy Thipparthi**, Who is a AWS Certified Security specialist, Solutions Architect - Professional, AWS SysOps Administrator, Red Hat Certified Engineer and Microsoft Certified Professional with over 8 years of experience in IT Infrastructure Production support.

This AWS Administration Guide will help you to gain knowledge on the below concepts:

- Cloud Computing and AWS accompanied by steps to sign up for AWS account.
- Create and manage users, groups, and permissions using AWS Identity and Access Management services.
- Deploying and accessing EC2 instances, working with EBS Volumes and Snapshots.
- Customizing and creating own AMI's.
- Effectively monitor AWS using custom monitoring metrics.
- Exploring the various Database-as-a-Service offerings and leverage those using Amazon RDS and Amazon DynamoDB.
- Design and deploying instances on a highly secured, network isolated environment using Amazon VPCs, ELB and Auto Scaling groups.
- Hosting options and routing policies with Amazon Web Services.
- Security options with Amazon Web Services.

INDEX

Introduction To Cloud Computing

What is Cloud Computing?

Cloud computing is the on-demand delivery of compute power, database storage, applications, and other IT resources through a cloud services platform via the internet with pay-as-you-go pricing.

Cloud Computing Basics

Whether you are running applications that share photos to millions of mobile users or you're supporting the critical operations of your business, a cloud services platform provides rapid access to flexible and low cost IT resources. With cloud computing, you don't need to make large upfront investments in hardware and spend a lot of time on the heavy lifting of managing that hardware. Instead, you can provision exactly the right type and size of computing resources you need to power your newest bright idea or operate your IT department. You can access as many resources as you need, almost instantly, and only pay for what you use.

Six Advantages and Benefits of Cloud Computing by Amazon:

Trade capital expense for variable expense

Instead of having to invest heavily in data centers and servers before you know how you're going to use them, you can only pay when you consume computing resources, and only pay for how much you consume.

Benefit from massive economies of scale

By using cloud computing, you can achieve a lower variable cost than you can get on your own. Because usage from hundreds of thousands of customers are aggregated in the cloud, providers such as Amazon Web Services can achieve higher economies of scale which translates into lower pay as you go prices.

Stop guessing capacity

Eliminate guessing on your infrastructure capacity needs. When you make a capacity decision prior to deploying an application, you often either end up sitting on expensive idle resources or dealing with limited capacity. With cloud computing, these problems go away. You can access as much or as little as you need, and scale up and down as required with only a few minutes notice.

Increase speed and agility

In a cloud computing environment, new IT resources are only ever a click away, which means you reduce the time it takes to make those resources available to your developers from weeks to just minutes. This results in a dramatic increase in agility for the organization, since the cost and time it takes to experiment and develop is significantly lower.

Stop spending money on running and maintaining data centers

Focus on projects that differentiate your business, not the infrastructure. Cloud computing lets you focus on your own customers, rather than on the heavy lifting of racking, stacking and powering servers.

Go global in minutes

Easily deploy your application in multiple regions around the world with just a few clicks. This means you can provide a lower latency and better experience for your customers simply and at minimal cost.

The NIST Definition of Cloud Computing

NIST is responsible for developing standards and guidelines, including minimum requirements, for providing adequate information security for all agency operations and assets.

Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction.

Essential Characteristics:

1. **On-demand self-service**

   A consumer can unilaterally provision computing capabilities, such as server time and network storage, as needed automatically without requiring human interaction with each service provider.

2. **Broad network access.**

   Capabilities are available over the network and accessed through standard mechanisms that promote use by heterogeneous thin or thick client platforms (e.g., mobile phones, tablets, laptops, and workstations)

3. **Resource pooling**

   The provider's computing resources are pooled to serve multiple consumers using a multi-tenant model, with different physical and virtual resources dynamically assigned and reassigned according to consumer demand. There is a sense of location independence in that the customer generally has no control or knowledge over the exact location of the provided resources but may be able to specify location at a higher level of abstraction (e.g., country, state, or datacenter). Examples of resources include storage, processing, memory, and network bandwidth.

4. **Rapid elasticity**

   Capabilities can be elastically provisioned and released, in some cases automatically, to scale rapidly outward and inward commensurate with demand. To the consumer, the capabilities available for provisioning often appear to be unlimited and can be appropriated in any quantity at any time.

5. **Measured service**

   Cloud systems automatically control and optimize resource use by leveraging a metering capability at some level of abstraction appropriate to the type of service (e.g., storage, processing, bandwidth, and active user accounts). Resource usage can be monitored, controlled, and reported, providing transparency for both the provider and consumer of the utilized service.

Service Models:

1. **Software as a Service (SaaS)**

   The capability provided to the consumer is to use the provider's applications running on a cloud infrastructure. The applications are accessible from various client devices through either a thin client interface, such as a web browser (e.g., web-based email), or a program interface. The consumer does not manage or control the underlying cloud infrastructure including network, servers, operating systems, storage, or even individual application capabilities, with the possible exception of limited user specific application configuration settings.

   SaaS providers host an application and make it available to users through the internet, usually a browser-based interface. As the most familiar category of cloud computing, users most commonly interact with SaaS applications such as Gmail, Dropbox, Salesforce, or Netflix.

2. **Platform as a Service (PaaS)**

   The capability provided to the consumer is to deploy onto the cloud infrastructure consumer-created or acquired applications created using programming languages, libraries, services, and tools supported by the provider.3 The consumer does not manage or control the underlying cloud infrastructure including network, servers, operating systems, or storage, but has control over the deployed applications and possibly configuration settings for the application-hosting environment.

   PaaS solutions appeal to developers who want to spend more time coding, testing, and deploying their applications instead of dealing with hardware-oriented tasks such as managing security patches and operating system updates.

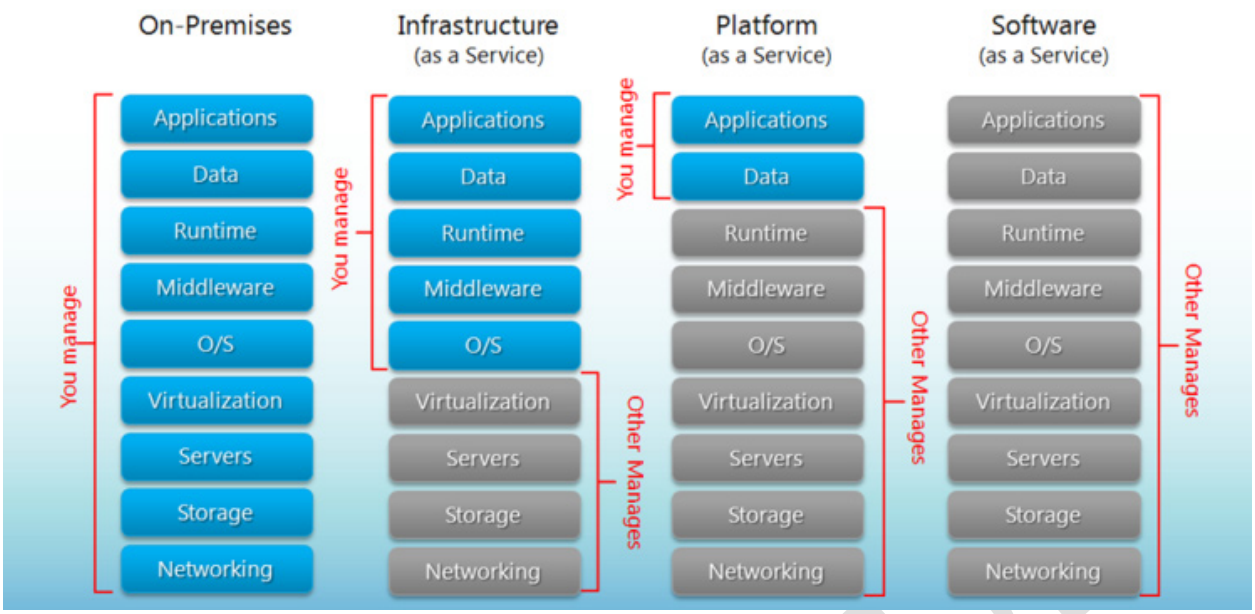
3. **Infrastructure as a Service (IaaS)**

   The capability provided to the consumer is to provision

   Processing, storage, networks, and other fundamental computing resources where the consumer is able to deploy and run arbitrary software, which can include operating systems and applications. The consumer does not manage or control the underlying cloud infrastructure but has control over operating systems, storage, and deployed applications; and possibly limited control of select networking components.

   IaaS providers deploy and manage pre-configured and virtualized hardware and enable users to spin up virtual machines or computing power without the labor-intensive server management or hardware investments.

   Amazon Web Services, for example, offers IaaS through the Elastic Compute Cloud, or EC2. Most IaaS packages cover the storage, networking, servers, and virtualization components, while IaaS customers are usually responsible for installing and maintaining the operating system, databases, security components, and applications.



Deployment Models:

1. **Private cloud**

   The cloud infrastructure is provisioned for exclusive use by a single organization comprising multiple consumers. It may be owned, managed, and operated by the organization, a third party, or some combination of them, and it may exist on or off premises.

   - A private cloud is dedicated to a single organization.
   - Private cloud offers hosted services to a limited number of people behind a firewall, so it minimizes the security concerns some organizations have around cloud. Private cloud also gives companies direct control over their data.

2. **Community Cloud**

   The cloud infrastructure is provisioned for exclusive use by a specific community of consumers from organizations that have shared concerns (e.g., mission, security requirements, policy, and compliance considerations). It may be owned, managed, and operated by one or more of the organizations in the community, a third party, or some combination of them, and it may exist on or off premises

   - A community cloud is a multi-tenant infrastructure that is shared among several organizations from a specific group with common computing concerns.
   - The community cloud can be either on-premises or off-premises, and can be governed by the participating organizations or by a third-party managed service provider.

3. **Public Cloud**

   The cloud infrastructure is provisioned for open use by the general public. It may be owned, managed, and operated by a business, academic, or government organization, or some combination of them. It exists on the premises of the cloud provider.

   - Computing resources, such as virtual machines (VMs), applications or storage, available to the general public over the internet.

- It reduces the need for organizations to invest in and maintain their own on-premises IT resources.
- It enables scalability to meet workload and user demands.

4. **Hybrid Cloud**

   The cloud infrastructure is a composition of two or more distinct cloud infrastructures (private, community, or public) that remain unique entities, but are bound together by standardized or proprietary technology that enables data and application portability (e.g., cloud bursting for load balancing between clouds).

   - Hybrid cloud is a combination of public and private cloud services, with orchestration between the two.

Introduction to AWS

What is Amazon Web Services?

Amazon Web Services (AWS) is a secure cloud services platform, offering compute power, database storage, content delivery and other functionality to help businesses scale and grow. Explore how millions of customers are currently leveraging AWS cloud products and solutions to build sophisticated applications with increased flexibility, scalability and reliability.

Amazon Web Services (AWS) is a subsidiary of Amazon.com that provides on-demand cloud computing platforms to individuals, companies and governments, on a paid subscription basis with a free-tier option available for 12 months. Amazon Web Services was officially launched on March 14, 2006, combining the three initial service offerings of Amazon S3 cloud storage, SQS, and EC2. AWS has more than 70 services including computing, storage, networking, database, analytics, application services, deployment, management, mobile, developer tools, and tools for the Internet of Things. The most popular include Amazon Elastic Compute Cloud (EC2) and Amazon Simple Storage Service (S3).

AWS Global infrastructure

The AWS Cloud operates 54 Availability Zones within 18 geographic Regions around the world. (Till Feb, 2018).

Region: Region is a collection of availability zones that are geographically located close to one other. Each region is a separate geographic area. There is no technical definition for AWS Region. Each region has multiple, isolated locations known as Availability Zones.

Availability Zone: These are essentially the physical data centers of AWS. This is the place where actual compute, storage, network, and database resources are hosted. A single availability zone is equal to a single data center. Each region will contains minimum of two Availability Zones.

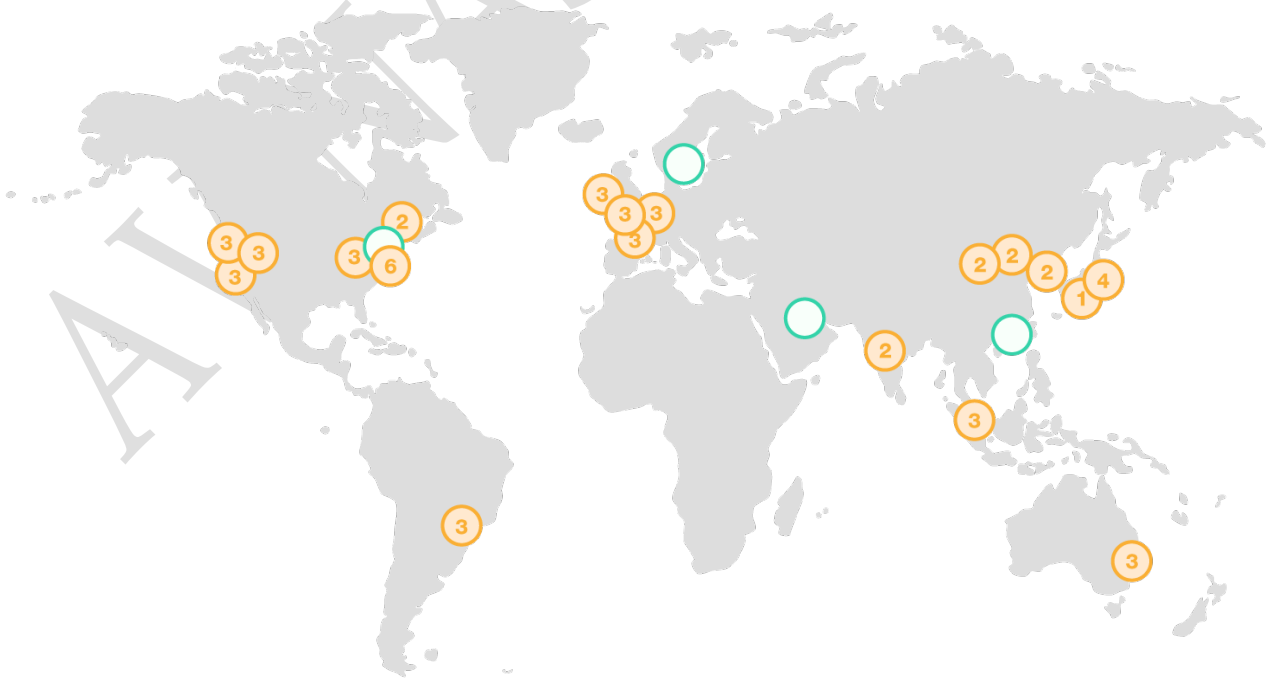
Edge Locations: Edge locations are CDN endpoints. Edge locations are located in most of the major cities around the world and are specifically used by CloudFront (CDN) to distribute content to end user to reduce latency. We have 98 Edge locations in 50 cities across 23 countries.

Amazon CloudFront is a web service that gives businesses and web application developers an easy and cost effective way to distribute content with low latency and high data transfer speeds.

Regions and Codes:

Region Code	Region Name
us-east-1	US East (N. Virginia)
us-east-2	US East (Ohio)
us-west-1	US West (N. California)
us-west-2	US West (Oregon)
ca-central-1	Canada (Central)
eu-west-1	EU (Ireland)
eu-central-1	EU (Frankfurt)
eu-west-2	EU (London)
ap-northeast-1	Asia Pacific (Tokyo)
ap-northeast-2	Asia Pacific (Seoul)
ap-southeast-1	Asia Pacific (Singapore)
ap-southeast-2	Asia Pacific (Sydney)
ap-south-1	Asia Pacific (Mumbai)
sa-east-1	South America (São Paulo)
us-gov-west-1	AWS GovCloud (US)
cn-north-1	China (Beijing)

- AWS GovCloud (US) account provides access to the AWS GovCloud (US) region only.
- AWS (China) account provides access to the China (Beijing) region only.



Region & Number of Availability Zones

US East
N. Virginia (6), Ohio (3)

US West
N. California (3), Oregon (3)

Asia Pacific
Mumbai (2), Seoul (2), Singapore (2), Sydney (3), Tokyo (3)

Canada
Central (2)

China
Beijing (2)

Europe
Frankfurt (3), Ireland (3), London (2)

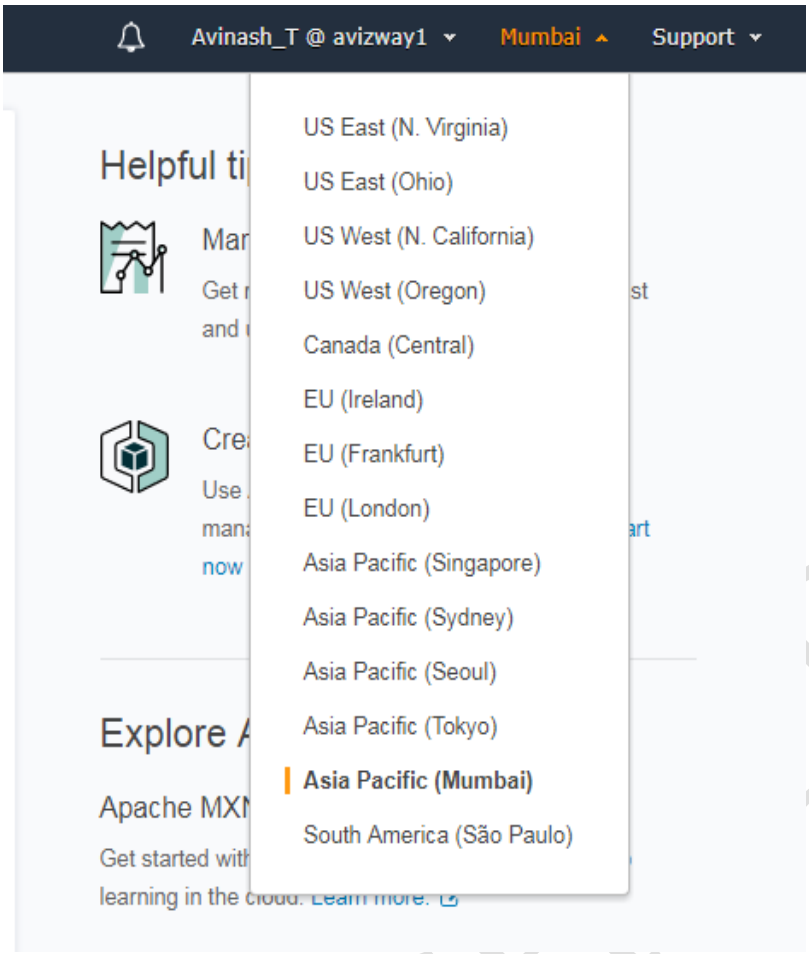
South America
São Paulo (3)

AWS GovCloud (US-West) (2)

New Region (coming soon)
Bahrain
Hong Kong SAR, China
Sweden
AWS GovCloud (US-East)

How to find regions and Availability Zones using the console

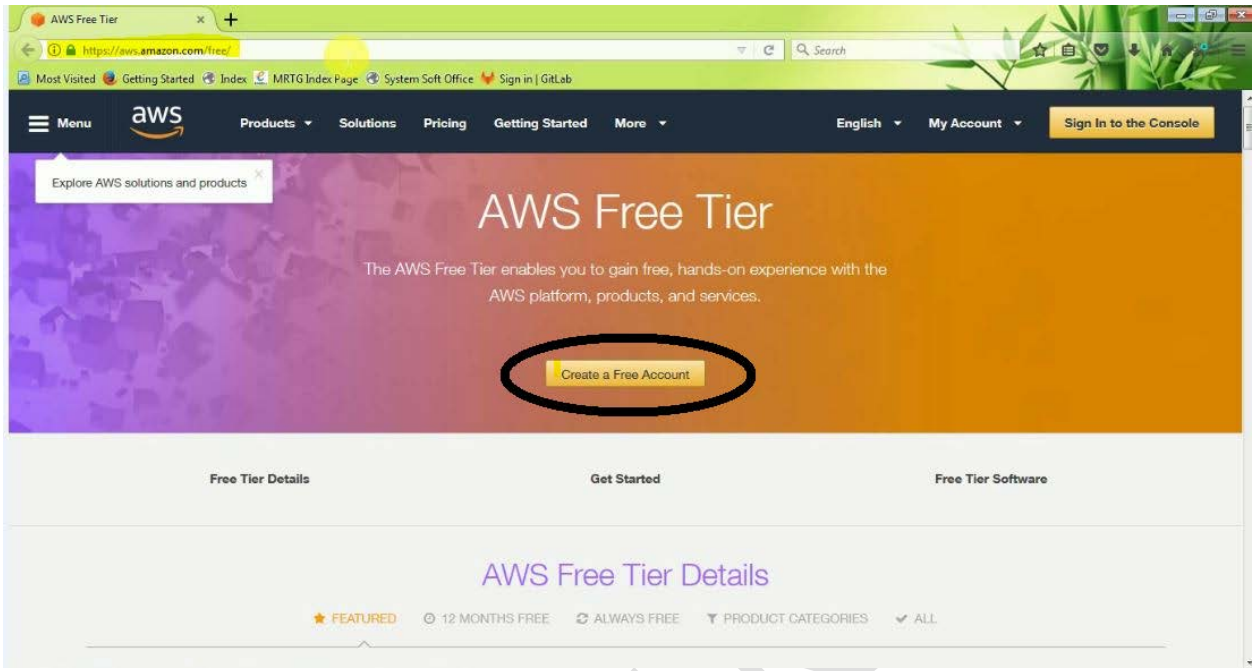
1. Open the Amazon EC2 console
2. From the navigation bar, view the options in the region selector.



3. You can switch between the regions and some services are region specific and some are global.

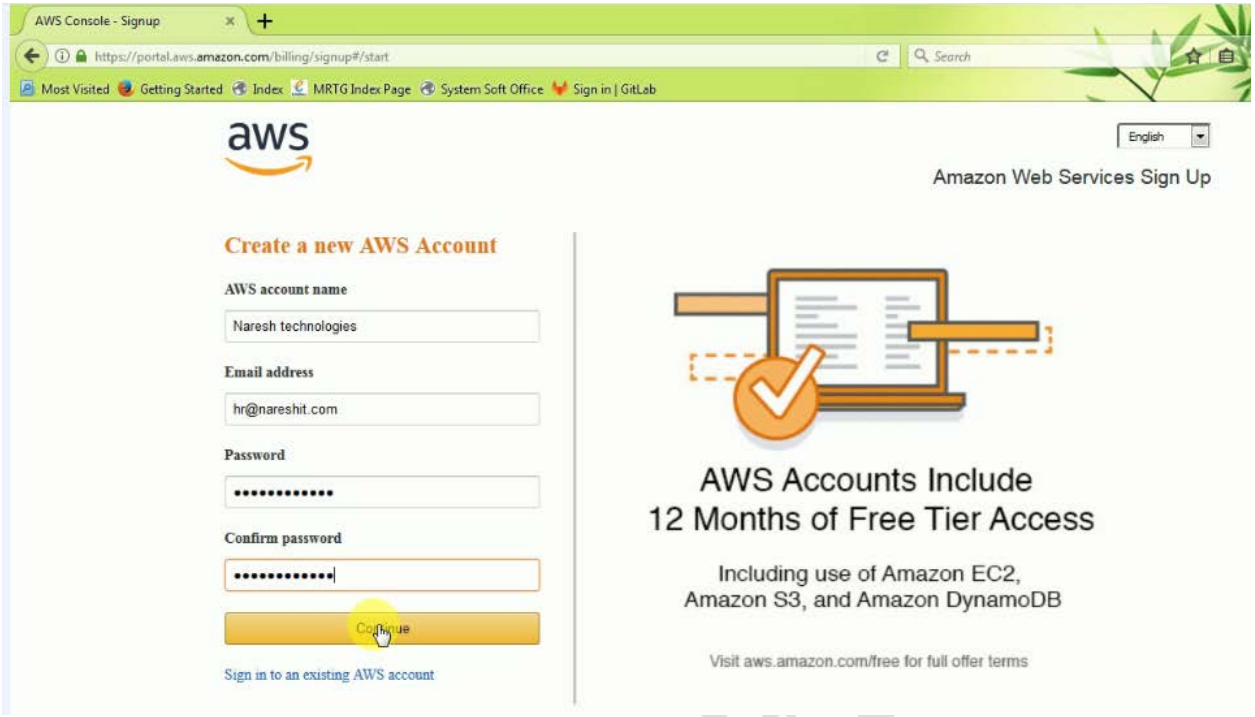
AWS Account Creation

1. Open **https://aws.amazon.com/free**, and verify the free tier limitations then choose "Create a Free Account".



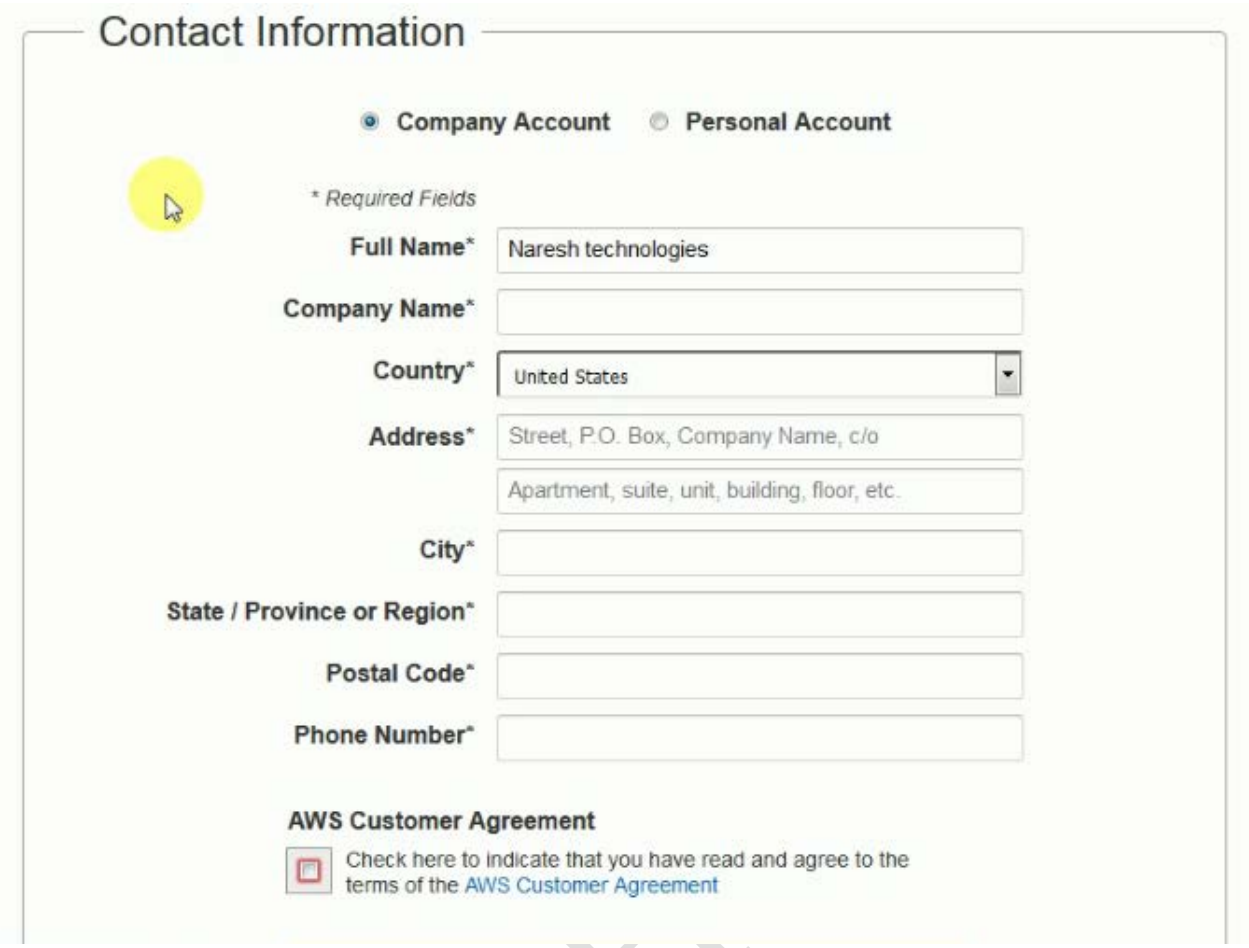
2. And Select "Create a new AWS account" option if you want to create a new account, or enter your Email ID if you are an existing user.

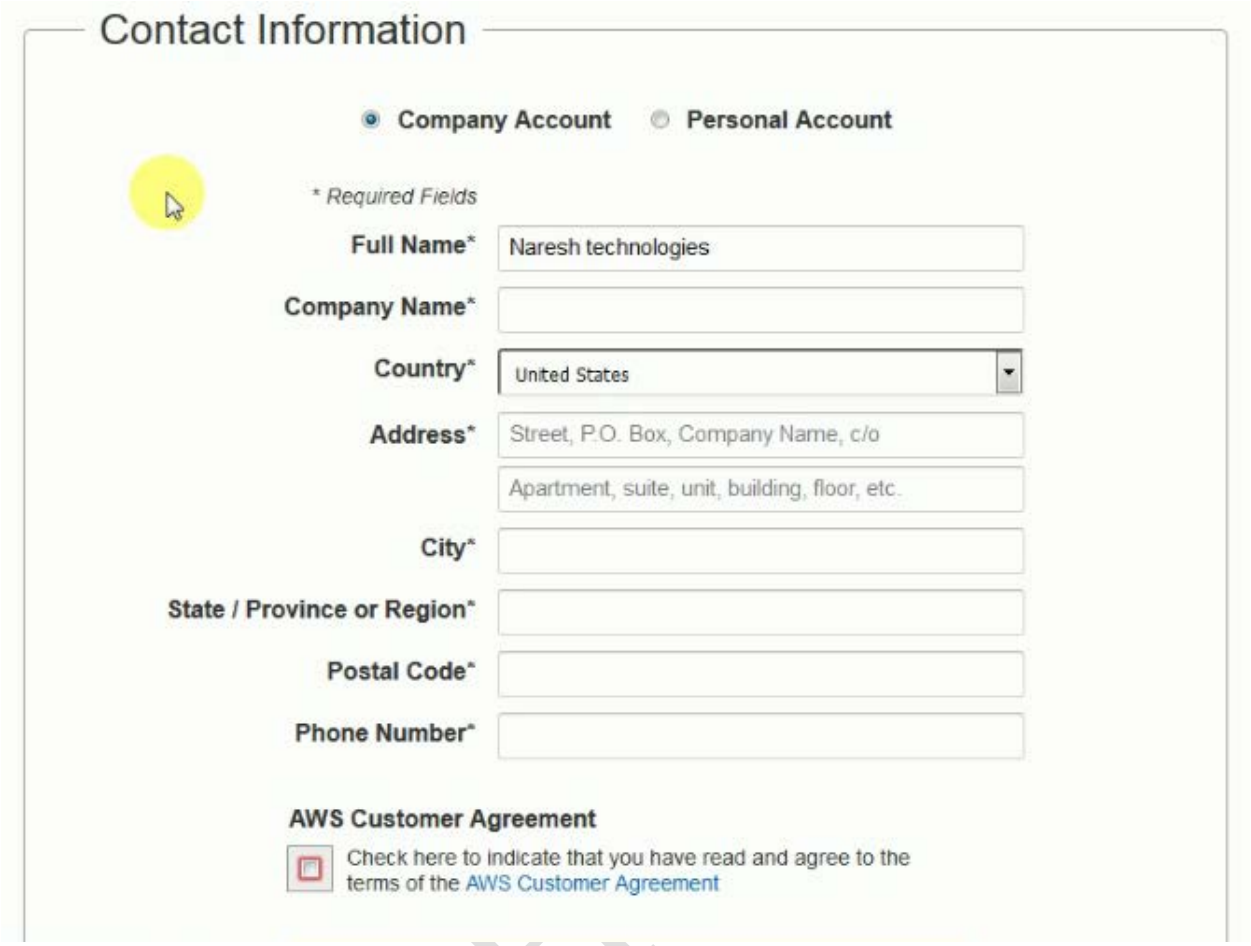
3. Enter the required details; AWS Account Name (You can give your name), Email Address and Choose a Password. Whatever the email ID you are using here is called as "**Root"** user and this user will have highest privileges on your AWS account.



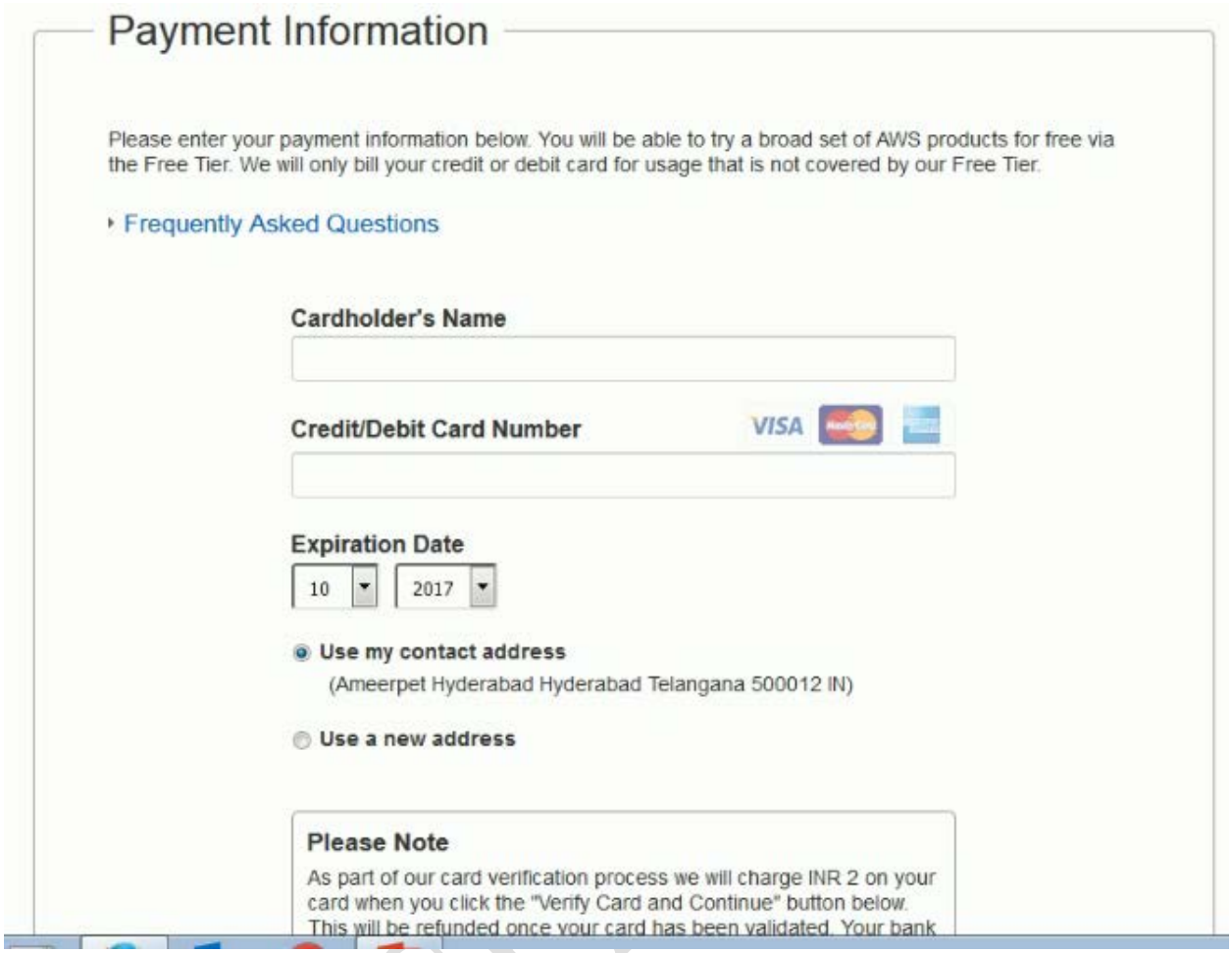
4. In this step we have to select “Account type” and need to provide the “Contact information”.
   - **a.** You can select “Personal Account” as your AWS account type, if you are an individual user.
   - **b.** You can select “Company Account” if you are creating this account for your organization.
   - **c.** You have to provide the required contact Information (i.e; Full Name, Country, Address, City, State, Postal code and Phone Number)
   - **d.** Click on checkbox for Agree the terms and conditions defined by Amazon.

   Then select **“Create account and continue”** button.

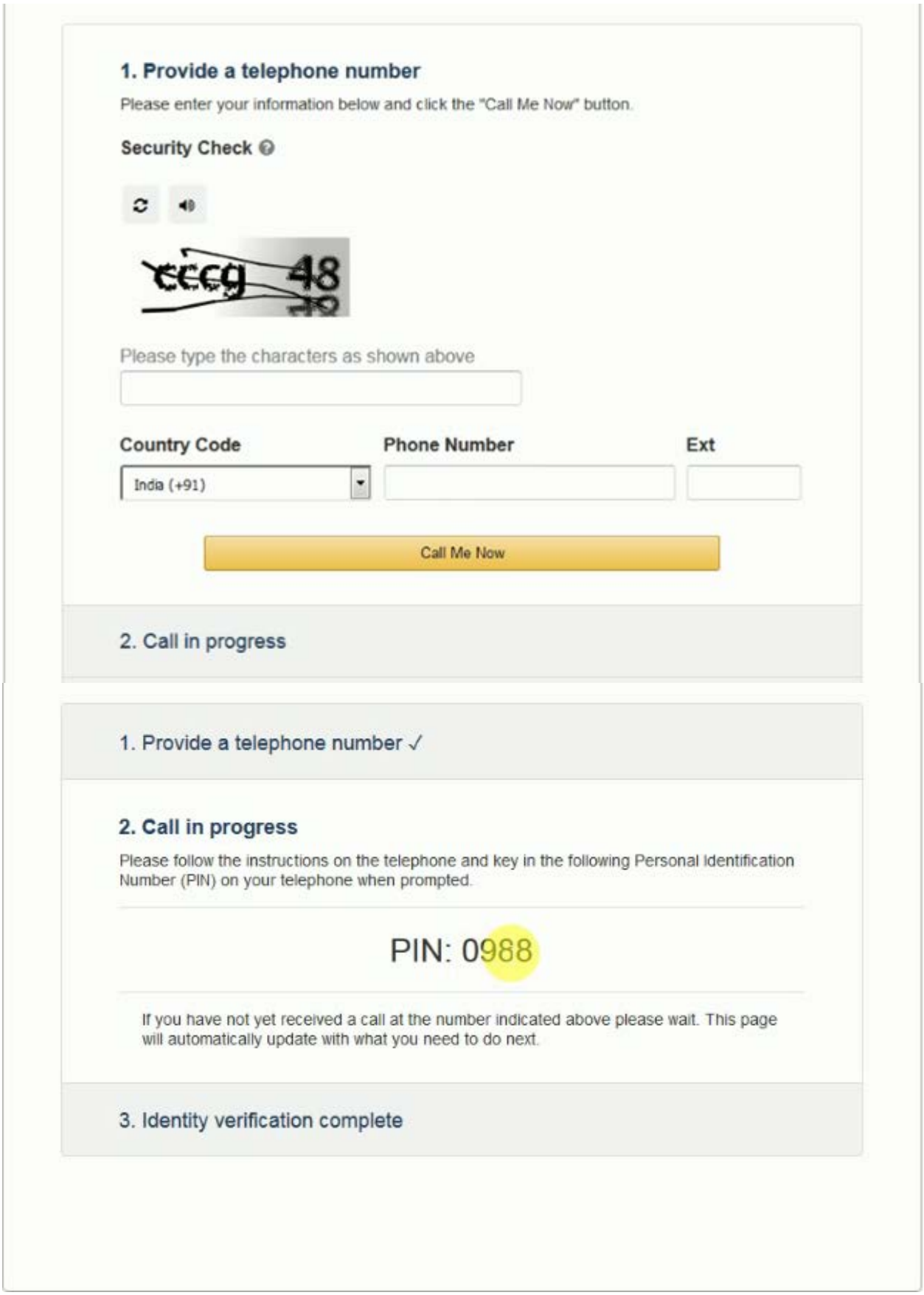

Contact Information
Company Account
Personal Account
* Required Fields
Full Name*
Naresh technologies
Company Name*
Country*
United States
Address*
Street, P.O. Box, Company Name, c/o
Apartment, suite, unit, building, floor, etc.
City*
State / Province or Region*
Postal Code*
Phone Number*
AWS Customer Agreement
Check here to indicate that you have read and agree to the terms of the AWS Customer Agreement



5. You have to enter your payment information. AWS will accept Credit/Debit Card (Visa /Mastercard /Americal express).
   As part of payment details verification process amazon will deduct INR 2 from your account. However this amount will refunded once your card has been validated.



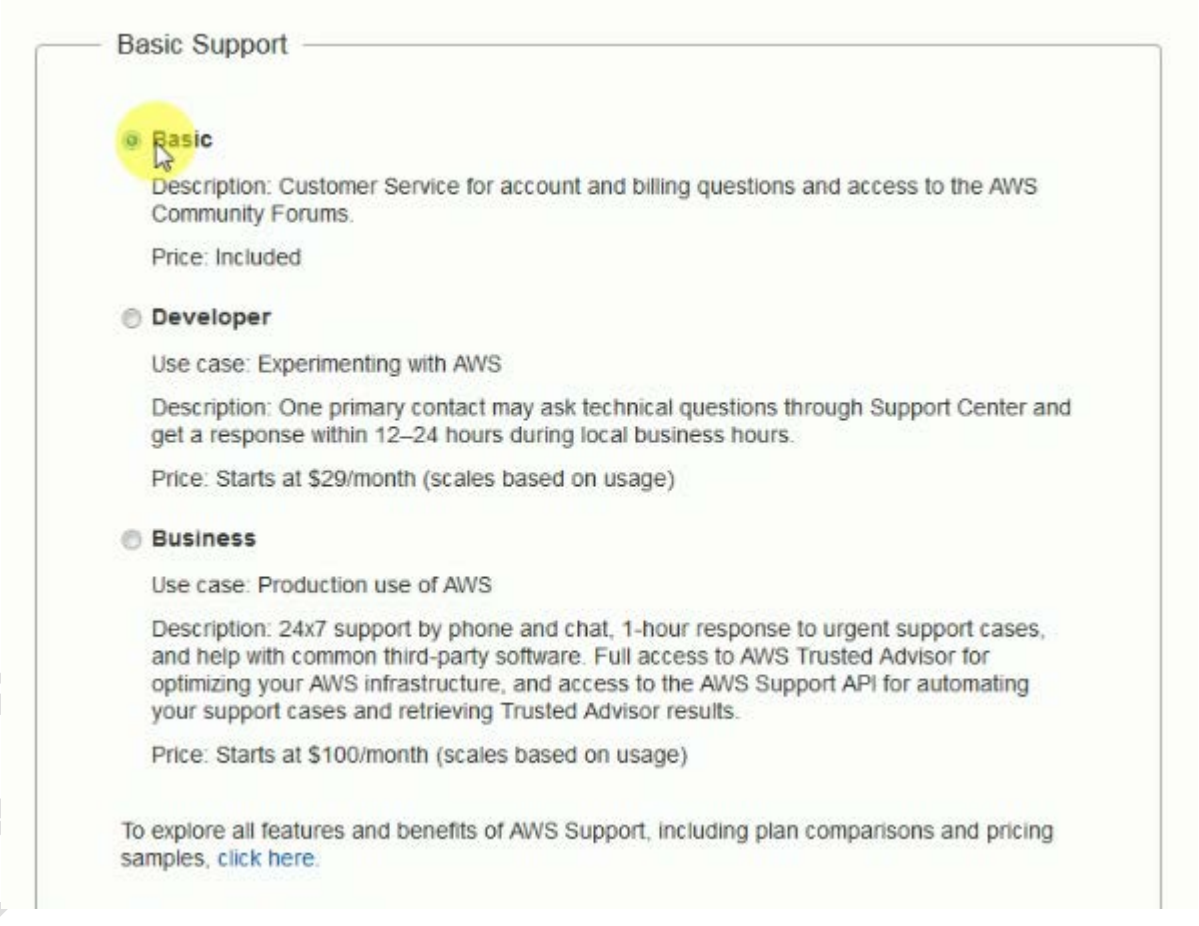
6. In Step 6, we have to perform “Identity verification” and to complete this step you need to have a valid Phone number with you.
    a. Enter the valid phone number, captcha and press “Call me now” button.
    b. When you click on call me now option, you will get a 4 digit PIN on your phone and simultaneously you will get a phone call from AWS to the mentioned phone number.
    c. You have to enter the 4 digit pin number on the IVR call, then your Identity verification is going to complete.



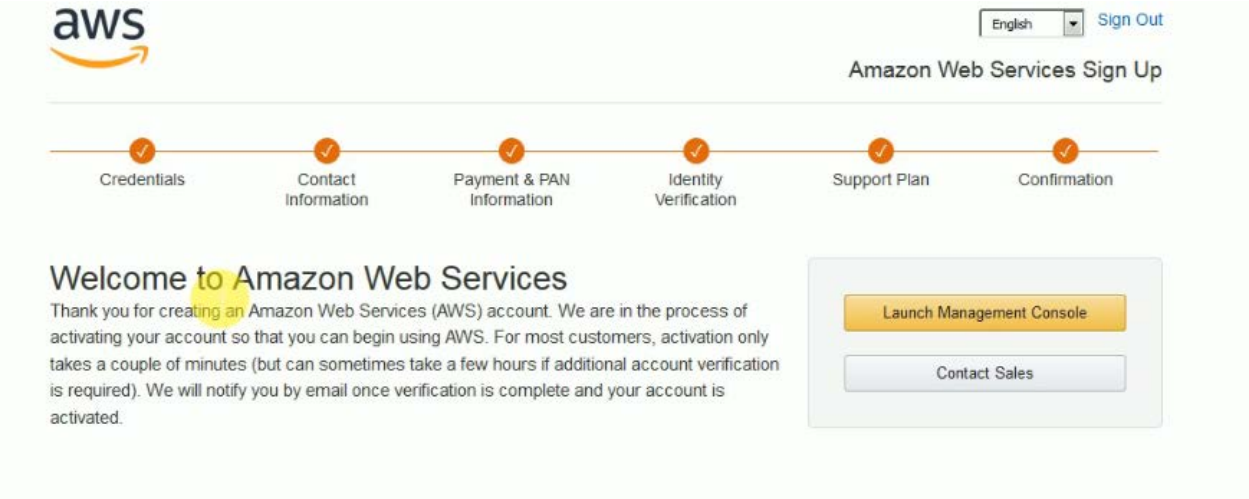
7. After completing the Identity verification, we have to select the “Support Plan” and click on “Continue”.
   Amazon have 4 support plans, those are

a. **Basic:** No Monthly Pricing for Basic support plan and no option to get technical support from Amazon if you are facing any.
b. **Developer:** Starting at $29/month and **one primary contact** may ask technical questions through support center and your issue will address within 12-24 hours during local business hours.
c. **Business:** Starting at $100/month and 24x7 access to Cloud Support Engineers via email, chat, and phone. 1 hour response to urgent support cases.
d. **Enterprise:** Starting at $15,000/month and you will get thee business support plan benefits along with Operational reviews, recommendations, and reporting, Designated Technical Account Manager, Access to online self-paced labs and Assigned Support Concierge.

Note: You can change this support plan at any time by logging in with Root account. You can "Support Center" under "support" navigation pane. Then click on change button and select the required support plan. We can use "Basic Support Plan" to explore the AWS features.



8. We have completed the AWS Account creation process select the **"Launch Management Console"**and Select **"Sign in to the console"**



9. Now you can enter the Email id and Password to login to your AWS account.

AWS basically offers usage of certain of its products at no charge for a period of 12months from the date of the actual signup.

AWS Product	What's free?
Amazon EC2	750 hours per month of Linux micro instance usage 750 hours per month of Windows micro instance usage
Amazon S3	5 GB of standard storage 20,000 get requests 2,000 put requests
Amazon RDS	750 Hours of Amazon RDS Single-AZ micro instance usage 20 GB of DB Storage: any combination of general purpose (SSD) or magnetic 20 GB for backups 10,000,000 I/Os
Amazon ELB	750 hours per month 15 GB of data processing

For complete list of free tier eligibility products, please refer https://aws.amazon.com/free/

IAM

(IDENTITY AND ACCESS MANAGEMENT)

Root User

When you first create an Amazon Web Services (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.

- The "root account" is simply the account created when first setup your AWS account. It has complete Admin access on your account.

AWS strongly recommend that you do not use the root user for your everyday tasks, even the administrative ones. Instead of using the root user we can create IAM user and allocates the appropriate permissions for the IAM user.

IAM:

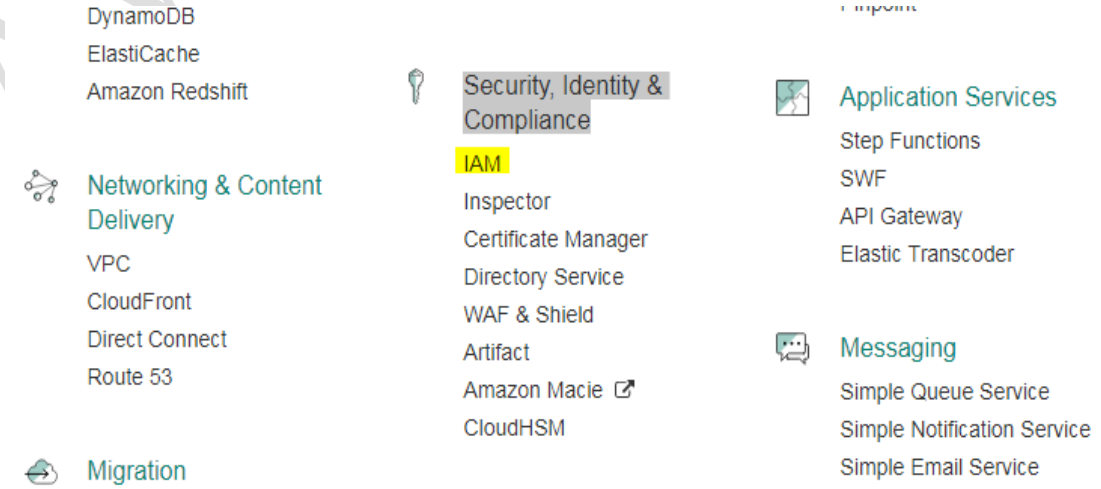
IAM stands for Identity and Access Management (IAM). IAM is a web service that helps you securely control access to AWS resources for your users. We can use IAM to control who can use our AWS resources and how they can use resources.

IAM Features:

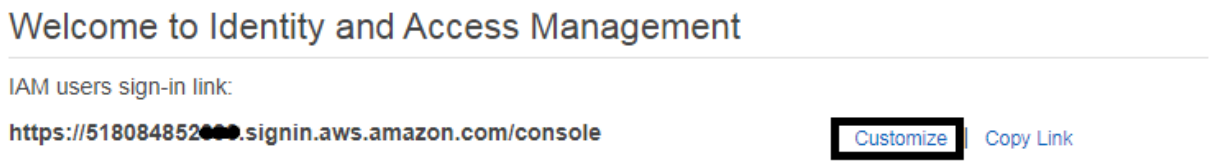
- You can provide Shared Access to your AWS account
- You can grant different permissions to different people for different resources.
- IAM allows you to manage users and their level of access to AWS console.
- IAM is universal. It does not apply to regions.
- You can enable Multi-factor authentication (MFA) for your AWS account
- IAM allows you to set up your own password rotation policy
- Integrates with many different AWS services

Steps to Create an IAM user:

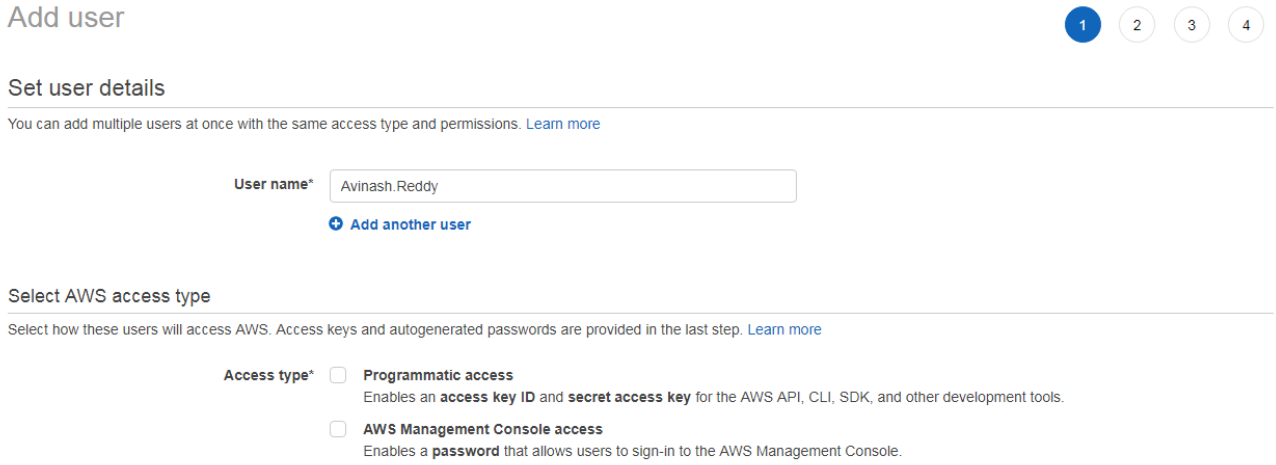
1. Login with the root Account credentials and find the **"IAM"** under **"Security, Identity & Compliance"**



2. IAM users have to sign-in using a dedicated Sign-In link. Every AWS account user will get a 12 Digit account number, that 12 digit number will be displayed on the Sign-In link, if you don't want to expose the account Number you can give an Alias name. For that select the **"customize"** option in IAM dashboard.



   - Alias name must be unique over the globe.
3. To create a new IAM user, Please select **"Users"** option under IAM Resources and Select **"Add User"** option.

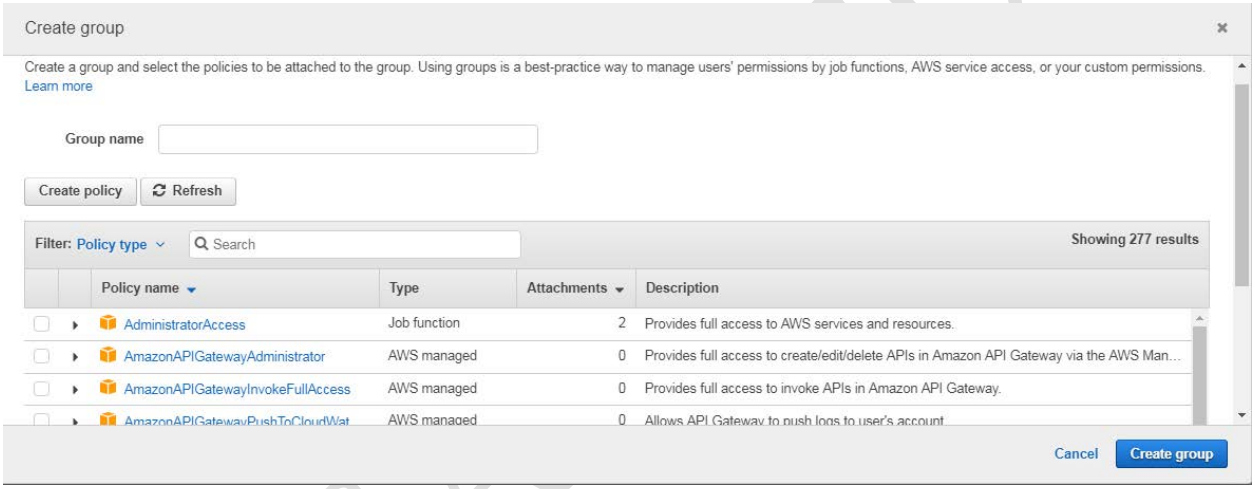


   - We need to provide a "user name" for the newly creating IAM user. This username must be unique with-in your AWS account.
   - Then you have to select AWS access type. We have two types of the access types
     - **Programmatic access:** This Enables the access to your AWS account by AWS API, CLI, SDK, and other development tools. You will get an access key ID and secret access key if you select this access type.
     - **AWS Management Console access:** This enables users to sign-in to the AWS Management Console i.e; Web Browser. You will get a username and password to login.
   - If you select "**AWS Management Console access"** you have to get a password by **"Auto generated password"** or **"Custom password"** option.
   - You can select the **"Require password reset option"** tick box if you want IAM user to create a new password at next sign-in.
4. By default IAM users will create with **NO Permissions**. If you want to allocate certain level of permission on any of the AWS resource, you have to attach/apply policy to the user.
   - You can directly Attach one or more existing policies directly to the users or create a new policy

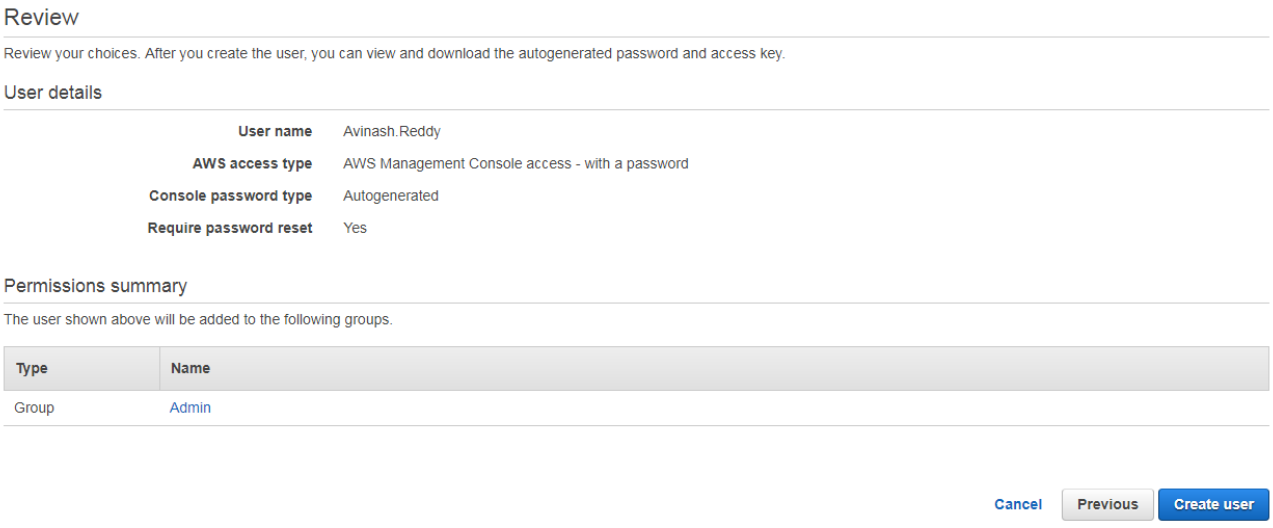
- If you have any existing user with policies you can select the user, same permissions will apply for the newly created user also.
- Or, you can create a group allocate the policy on top of the group, then you can add this IAM user to that group. Creating group will eases the administration.

5. To create a group, select the **"Create a Group"** option and you will get a pop-up to select the policy. You can filter the policies based on your requirement and select.
   Here is some key policies, you have to remember

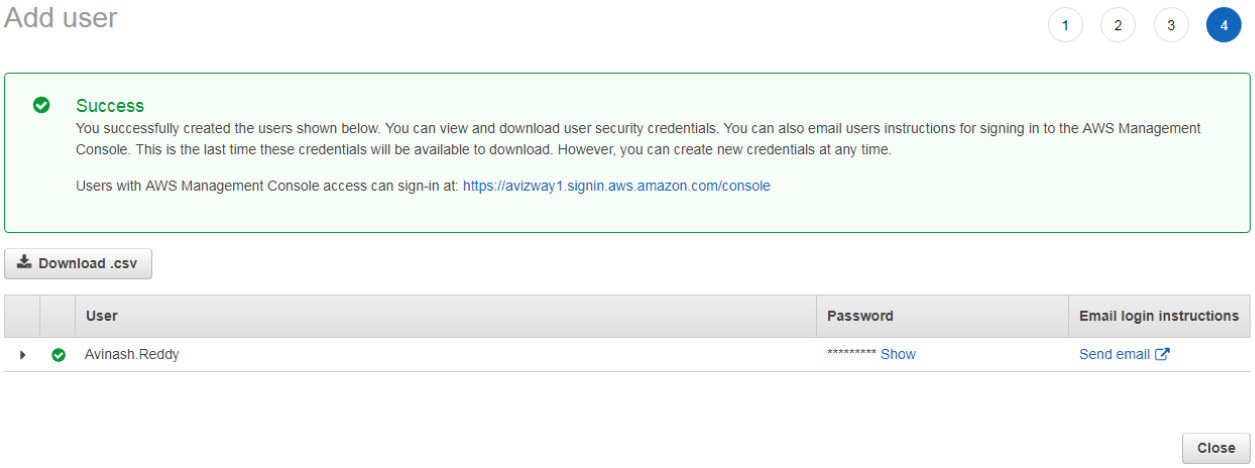
   - **AdministratorAccess:** Provides full access to AWS services and resources Except Billing and Account management. He can create/delete an IAM user or Groups.
   - **PowerUserAccess:** Provides full access to AWS services and resources, but does not allow management of Users and groups. He can launch any resource but doesn't have any permission to create a new user, group or deleting an existing user.
   - **ReadonlyAccess:** Provides Read Only access on all AWS services and resources.



6. Review the screen and click on **"Create User"** option. New IAM user will create and you can send the credentials directly to the user by using "Send Email" option.



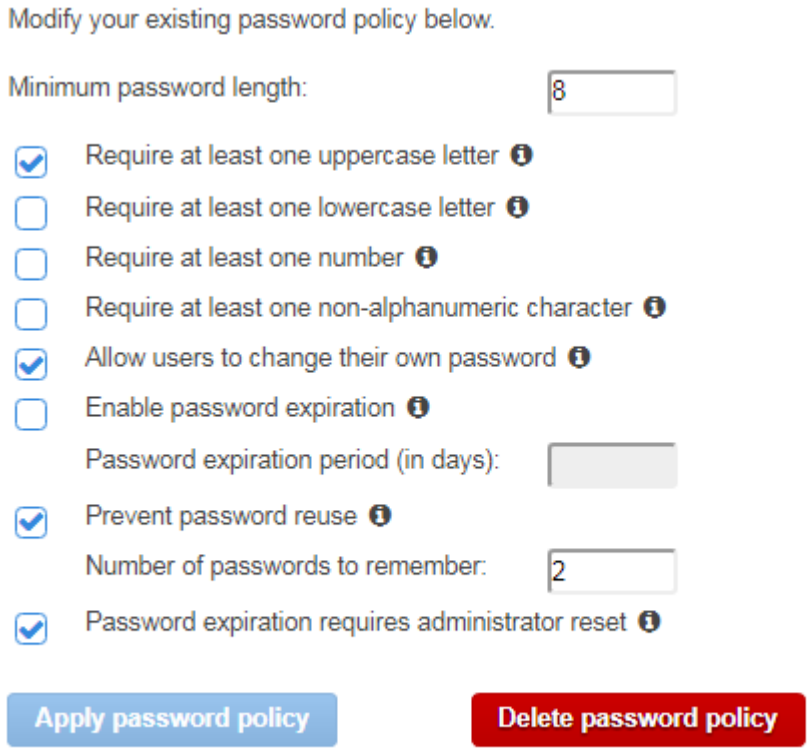
7. You can download the Credentials.csv file and keep it in a secured location.



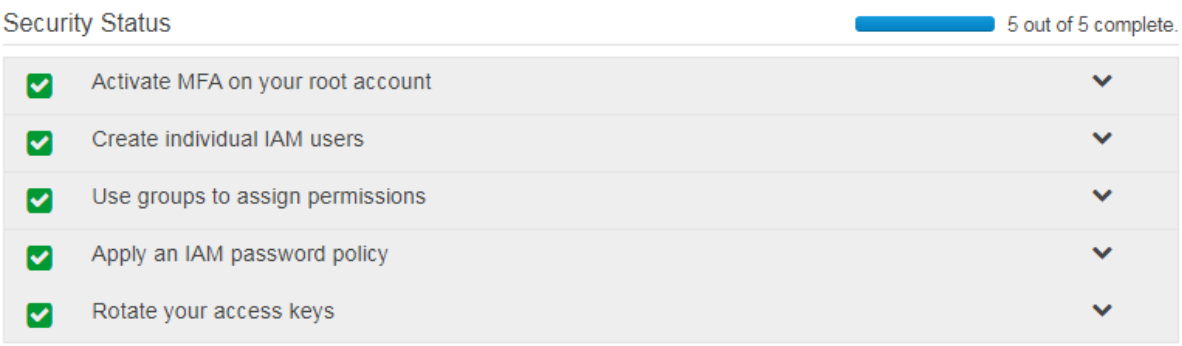
8. By using the mentioned IAM sig-in URL, this newly created IAM user can login to AWS console.

Setup own password policy:

A password policy is a set of rules that define the type of password an IAM user can set. You can set the password complexity to secure your AWS account from easily guessable passwords. You can modify the password policy based on the requirement.



9. You need to get all the tick marks in IAM dashboard, then you can consider you are good to go with other services.



EXERCISE 1

Create an IAM Group

Create a group for all IAM administrator users and assign the proper permissions to the new group. This will allow you to avoid assigning policies directly to a user later in these exercises.

1. Log in as the root user.
2. Create an IAM group called Administrators.
3. Attach the managed policy, IAM Full Access, to the Administrators group.

EXERCISE 2

Create a Customized Sign-In Link and Password Policy

In this exercise, you will set up your account with some basic IAM safeguards. The password policy is a recommended security practice, and the sign-in link makes it easier for your users to log in to the AWS Management Console.

1. Customize a sign-in link, and write down the new link name in full.
2. Create a password policy for your account.

EXERCISE 3

Create an IAM User

In this exercise, you will create an IAM user who can perform all administrative IAM functions. Then you will log in as that user so that you no longer need to use the root user login. Using the root user login only when explicitly required is a recommended security practice (along with adding MFA to your root user).

1. While logged in as the root user, create a new IAM user called Administrator.
2. Add your new user to the Administrators group.
3. On the Details page for the administrator user, create a password.
4. Log out as the root user.
5. Use the customized sign-in link to sign in as Administrator.

EXERCISE 4

Set Up MFA

In this exercise, you will add MFA to your IAM administrator. You will use a virtual MFA application for your phone. MFA is a security recommendation on powerful accounts such as IAM administrators.

1. Download the AWS Virtual MFA app to your phone.
2. Select the administrator user, and manage the MFA device.
3. Go through the steps to activate a Virtual MFA device.
4. Log off as administrator.
5. Log in as administrator, and enter the MFA value to complete the authentication process.

S3 (SIMPLE STORAGE SERVICE)

Introduction to S3

Amazon S3 is one of first services introduced by AWS. Amazon S3 provides developers and IT teams with secure, durable, and highly-scalable cloud storage. Amazon S3 is easy-to-use object storage with a simple web service interface that you can use to store and retrieve any amount of data from anywhere on the web. Amazon S3 also allows you to pay only for the storage you actually use, which eliminates the capacity planning and capacity constraints associated with traditional storage. Block storage operates at a lower level, the raw storage device level and manages data as a set of numbered, fixed-size blocks. Object storage or File storage operates at a higher level, the operating system level, and manages data as a named hierarchy of files and folders.

- S3 is Object based i.e. allows you to upload, Download, Share files.
- All our Objects reside in containers called **buckets.**
- S3 is a universal namespace that means **name of your bucket must be unique globally**.
- Amazon S3 is cloud object storage. Instead of being closely associated with a server, Amazon S3 storage is independent of a server and is accessed over the Internet.
- You can create and use multiple buckets; you can have up to **100 per account by default,** this is a soft limit, you can increase this at any time by creating a service limit increase ticket with AWS.
- File Size can be from 0 Byte to 5TB
- Single bucket can store an unlimited number of files.
- You can create buckets in your nearby region which is located close to a particular set of end users or customers in order to minimize latency.
- Or, Create bucket and store data far away from your primary facilities in order to satisfy disaster recovery and compliance needs
- Amazon S3 objects are automatically replicated on multiple devices in multiple facilities within a region
- Every Amazon S3 object can be addressed by a unique URL i.e; http://mybucket.s3.amazonaws.com/document.doc
- You can access using this URL also https://s3-region.amazonaws.com/uniquebucketName/objectname

- Bucket names must be at least 3 and no more than 63 characters long
- Bucket names must not be formatted as an IP address (e.g., 192.168.32.1).

Invalid Bucket Name	Comment
.myawsbucket	Bucket name cannot start with a period (.).
myawsbucket.	Bucket name cannot end with a period (.).
my..examplebucket	There can be only one period between labels

S3 Storage classes:

S3-Standard – Amazon S3 Standard offers high durability, high availability, low latency, and high performance object storage for general purpose use. 99.99% availability, 99.999999999% durability, stored redundantly across multiple devices in multiple facilities and is designed to sustain the loss of 2 facilities concurrently.

S3 - IA (Infrequently Accessed) For data that is accessed less frequently, but requires rapid access when needed. Lower fee than S3, but you are charged a retrieval fee. Min Obj Size is 128Kb.

- Designed for durability of 99.999999999% of objects
- Designed for 99.9% availability over a given year
- Lower Price than S3 Standard
- Designed for storing less frequently accessed data.
- Minimum duration 30 days
- Retrieval charges applicable

Reduced Redundancy Storage - Designed to provide 99.99% durability and 99.99% availability of objects over a given year. It is most appropriate for derived data that can be easily reproduced, such as image thumbnails.

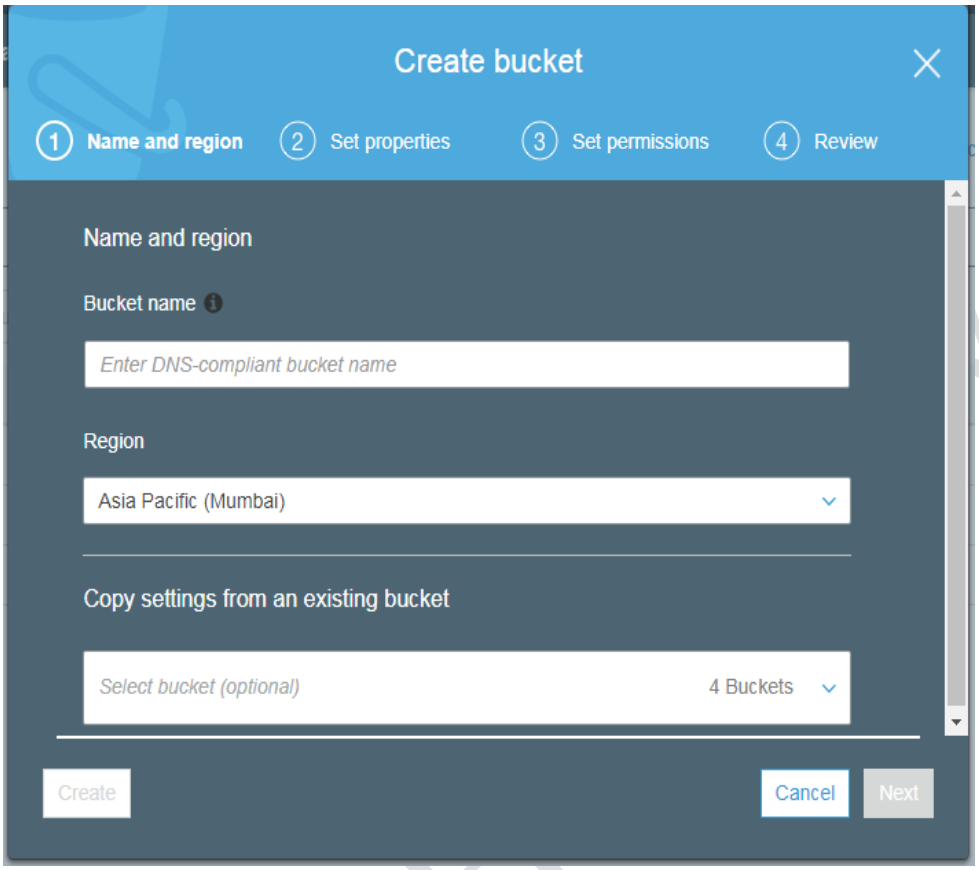
Glacier - Amazon Glacier is an extremely low-cost storage service that provides durable, secure, and flexible storage for data archiving and online backup. Storage class offers secure, durable, and extremely low-cost cloud storage for data that does not require real-time access, such as archives and long-term backups.

- **Archives:** In Amazon Glacier, data is stored in archives. An archive can contain up to **40TB** of data, andyou can have an unlimited number of archives.
- **Vaults:**Vaults are containers for archives. Each AWS account can have up to 1,000 vaults.
- After requesting for data three to five hours later, the Amazon Glacier object is copied to Amazon S3 RRS.
- Amazon Glacier allows you to retrieve up to 5% of the Amazon S3 data stored in Amazon Glacier for free each month.

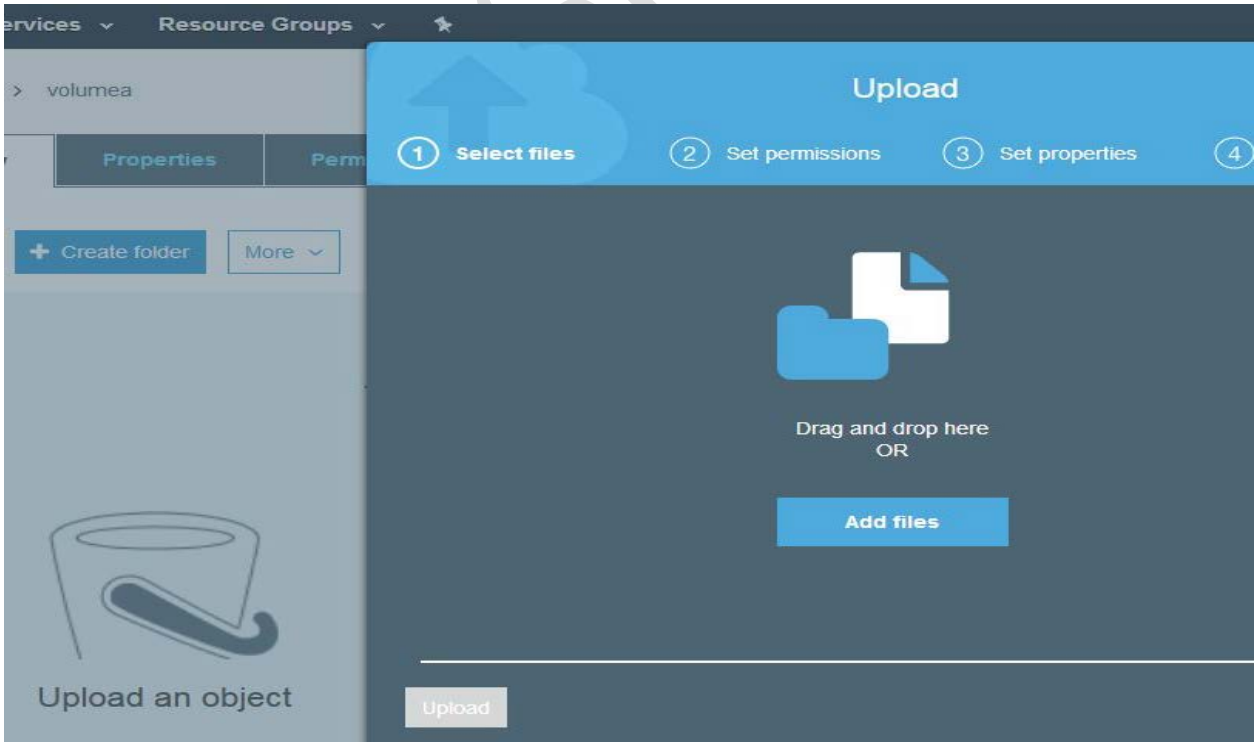
Availability and Durability chart

Storage Class	Durability (designed for)	Availability (designed for)	Other Considerations
STANDARD	99.999999999%	99.99%	None
STANDARD_IA	99.999999999%	99.9%	There is a retrieval fee associated with STANDARD_IA objects which makes it most suitable for infrequently accessed data.
GLACIER	99.999999999%	99.99% (after you restore objects)	GLACIER objects are not available for real-time access. You must first restore archived objects before you can access them.
RRS	99.99%	99.99%	None

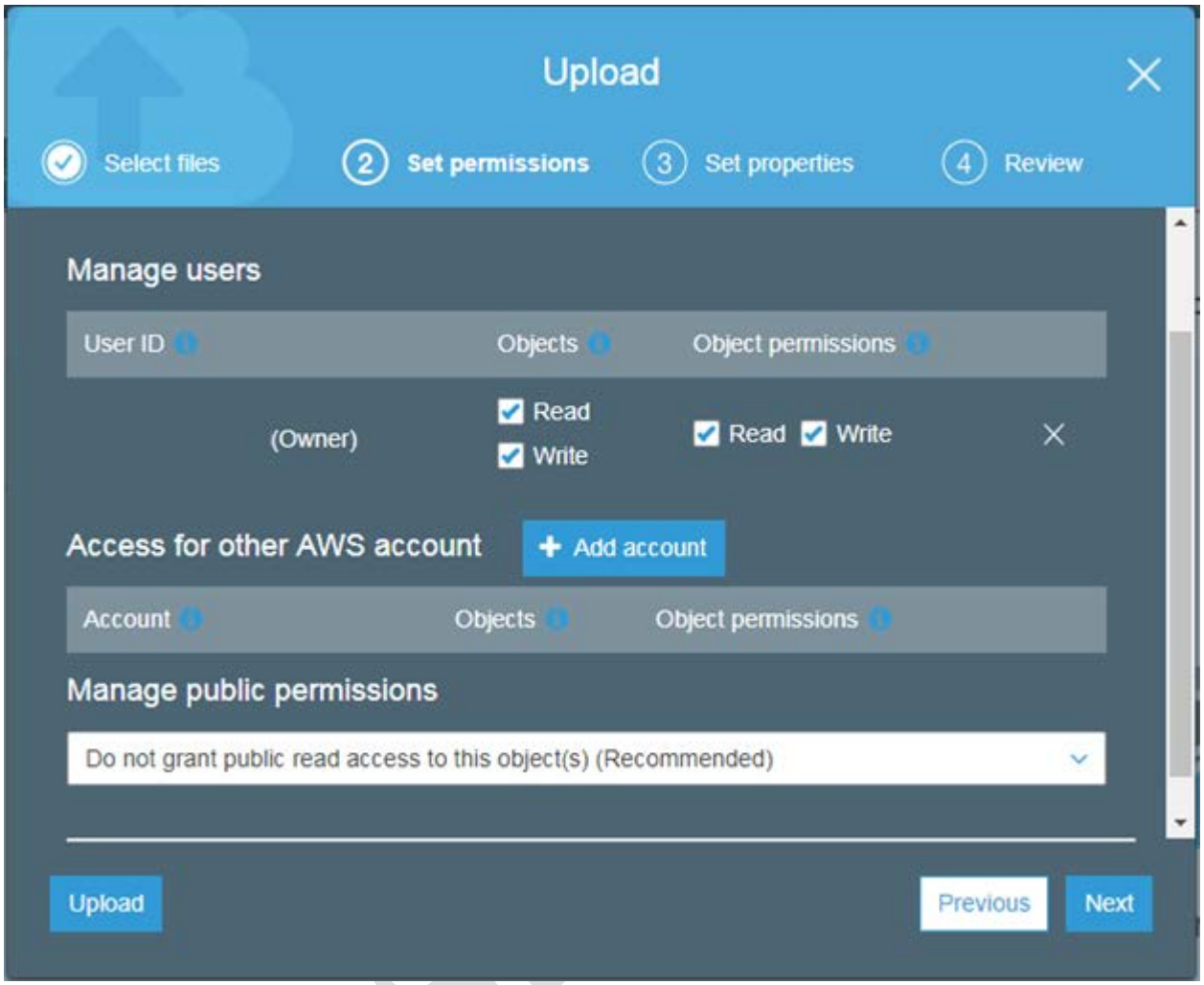
S3 Bucket Creation:



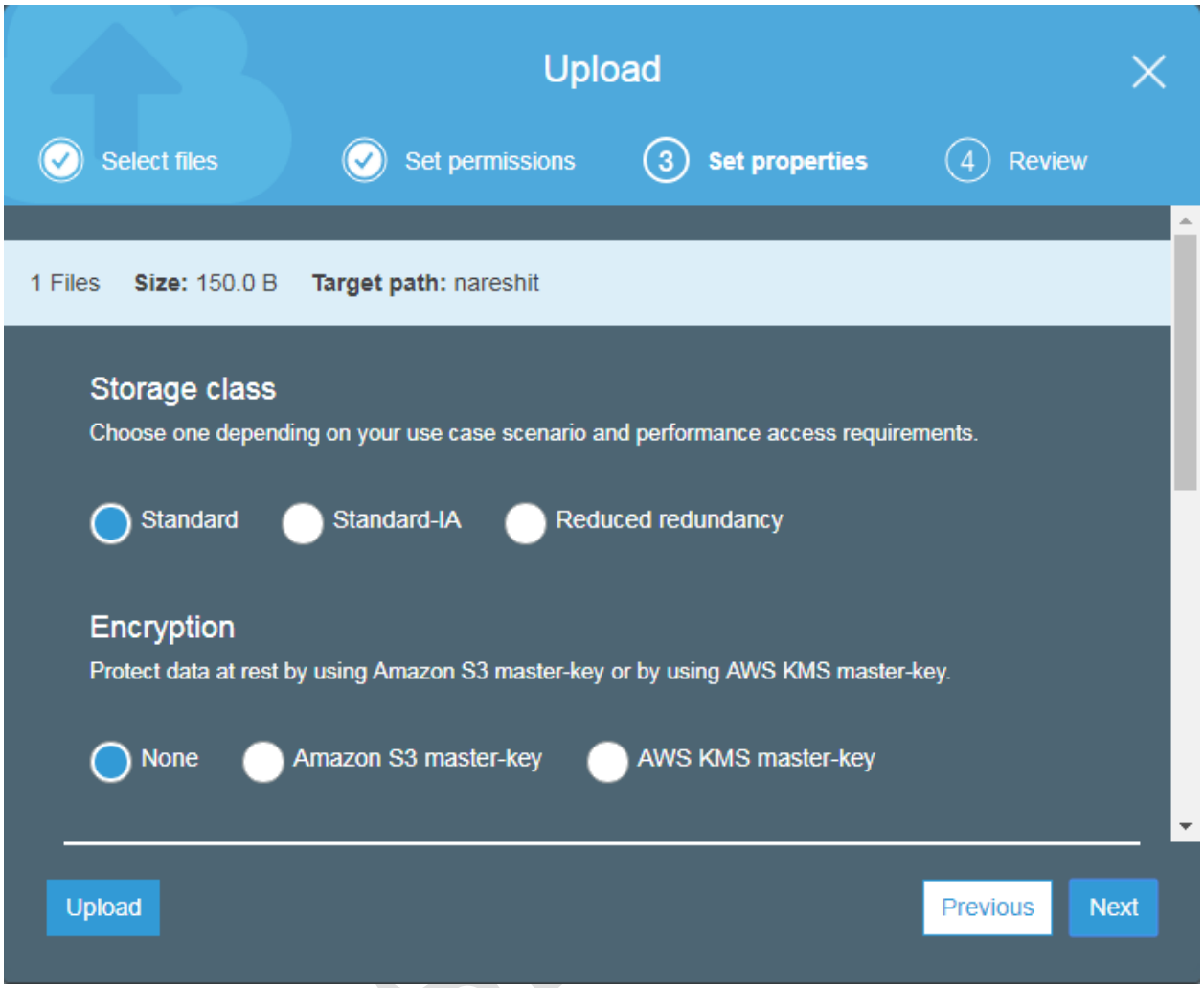
- We can Drag & Dropobjects to upload the objects.



- After selection of files, we can give access to other users who requiredpermissions.
- We can Manage Public Permissions or give permissions for other AWS account users.



- Here we can select the object Properties, We can select the Object storage class of the object, Encryption methods, Metadata and tags for the object.



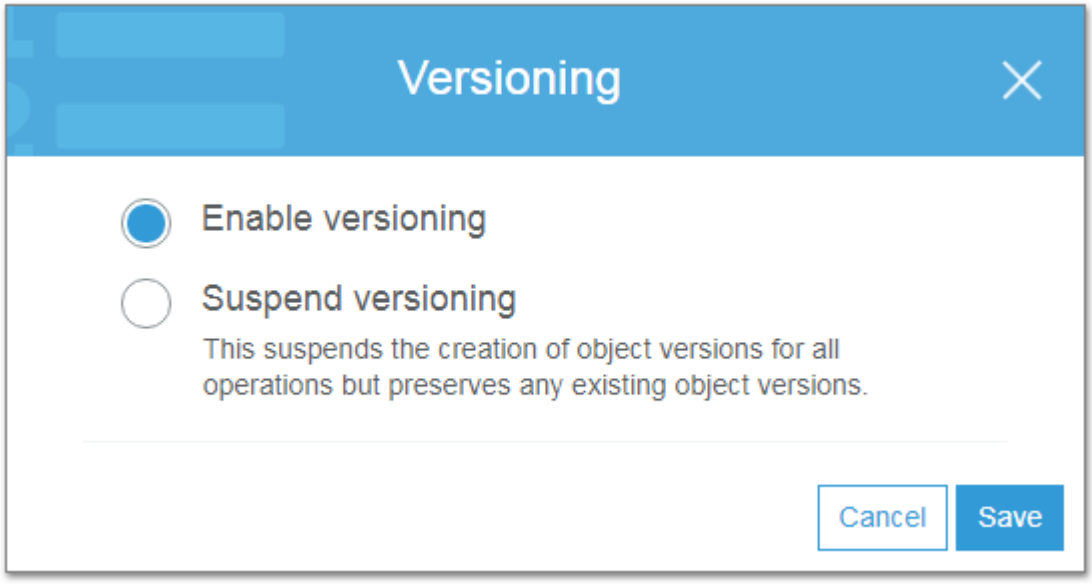
- Then we can review and click on upload option to upload the object into S3 bucket.

Versioning

Versioning helps protects your data against accidental or malicious deletion by keeping multiple versions of each object in the bucket, identified by a unique version ID.

- Versioning is turned on at the bucket level.
- Once enabled, versioning cannot be removed from a bucket; it can only be suspended.
- If you enable versioning you will get Current version files and previous version files in your bucket.
- If you delete current version file, if will overwrite with a Delete Marker, if you want to get that object back to you S3 bucket, you can delete the delete marker.

To enable versioning on bucket, navigate to properties of the respective bucket and select versioning and select “Enable versioning” option.



Lifecycle Management

By using Life cycle management we can automate the storage tiers in s3 buckets.

We can move objects from one storage class/tier to another storage class/tier based on our business requirements.

Here is the possible scenarios:

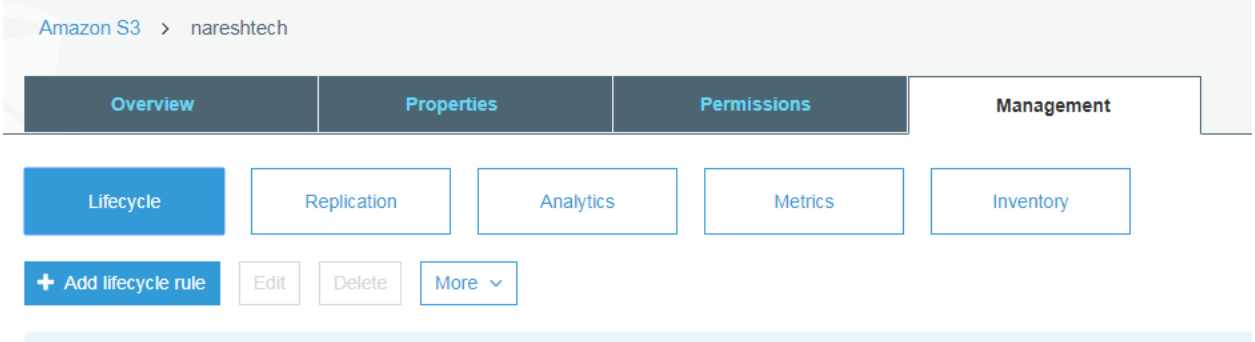
S3-Standard → S3-IA → Glacier → Delete

S3-Standard → Glacier → Delete

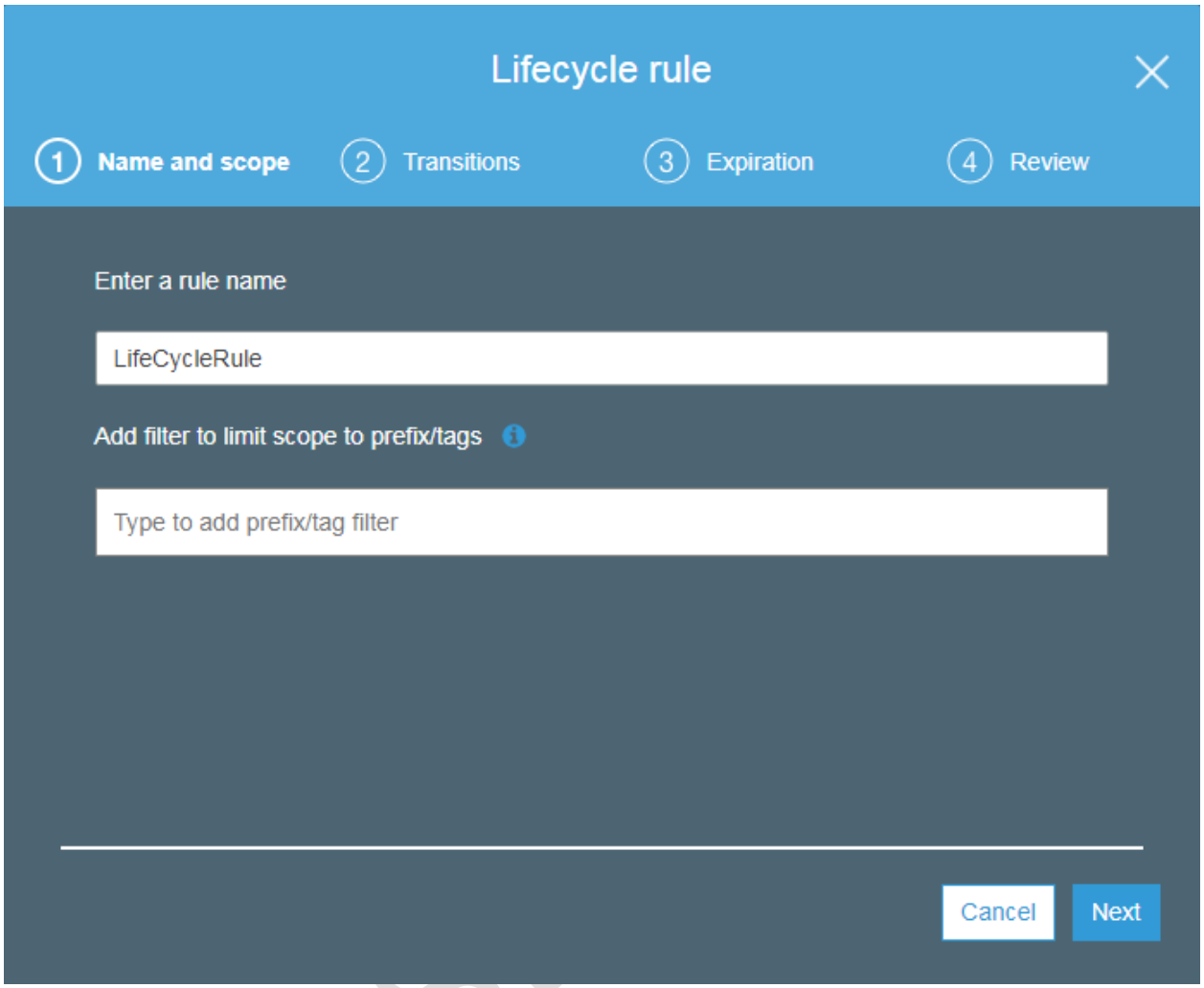
S3-Standard → Delete

Steps to enable lifecycle management rules:

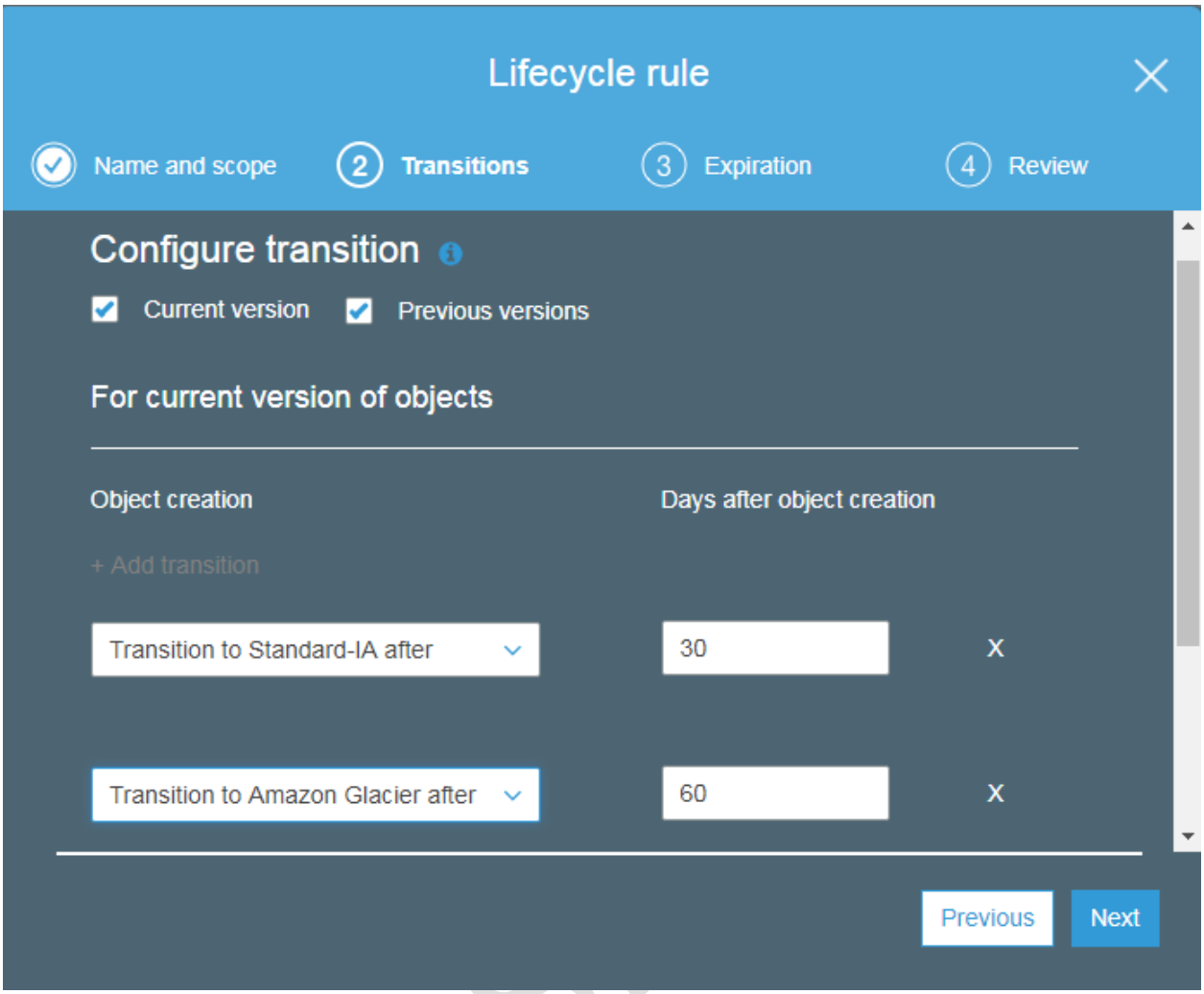
- Select the S3 bucket which we want to add life cycle rule.
- Go to management option after selecting the bucket.



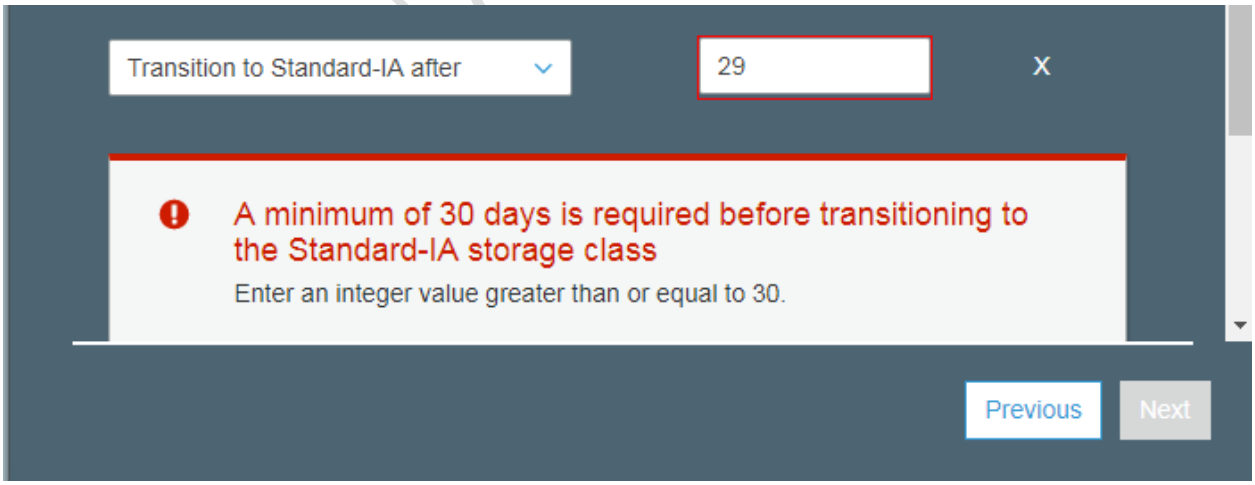
- Select Add Lifecycle rule and then give a valid name for the life cycle rule. We can add prefix, If LC rule will apply to the entire buckets objects.



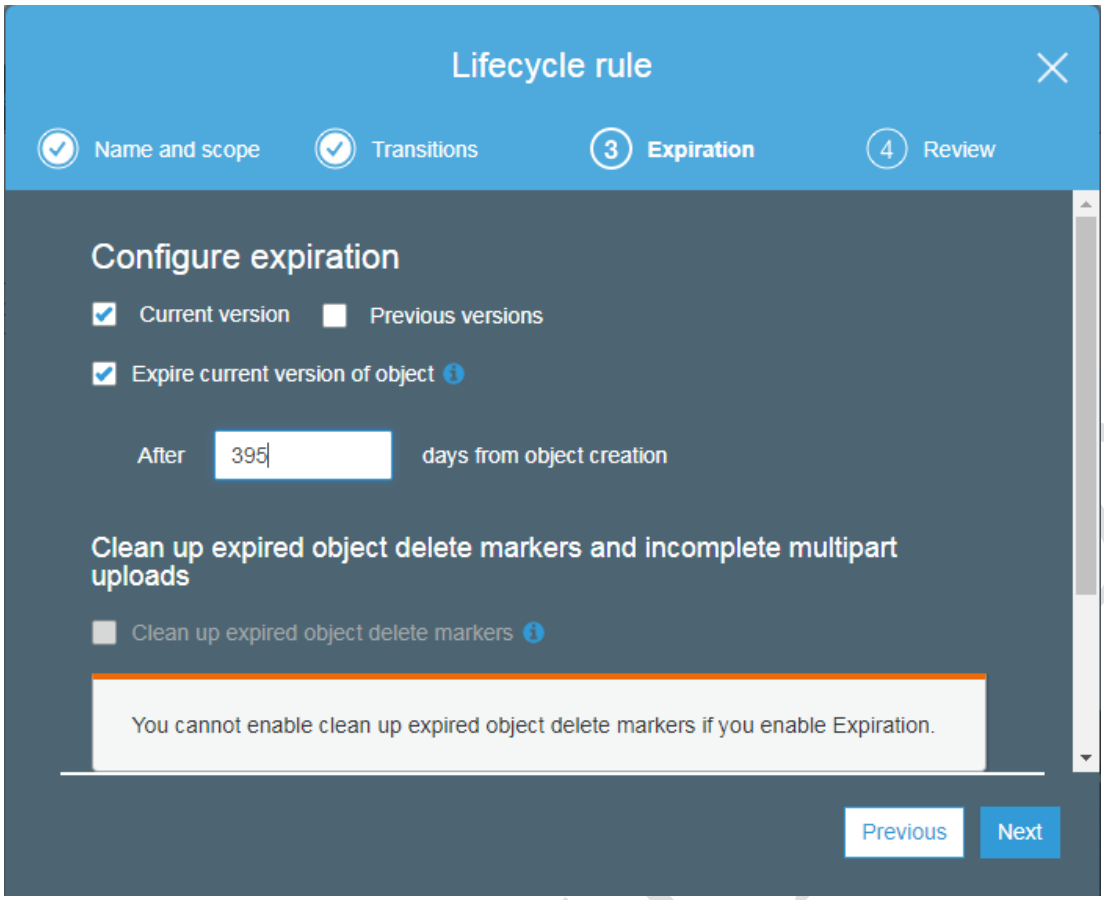
- After entering “name and scope” we need to configure the transitions. We can configure transitions for current version and previous versions. Click “add transition” and enter the days count from “Object creation”.



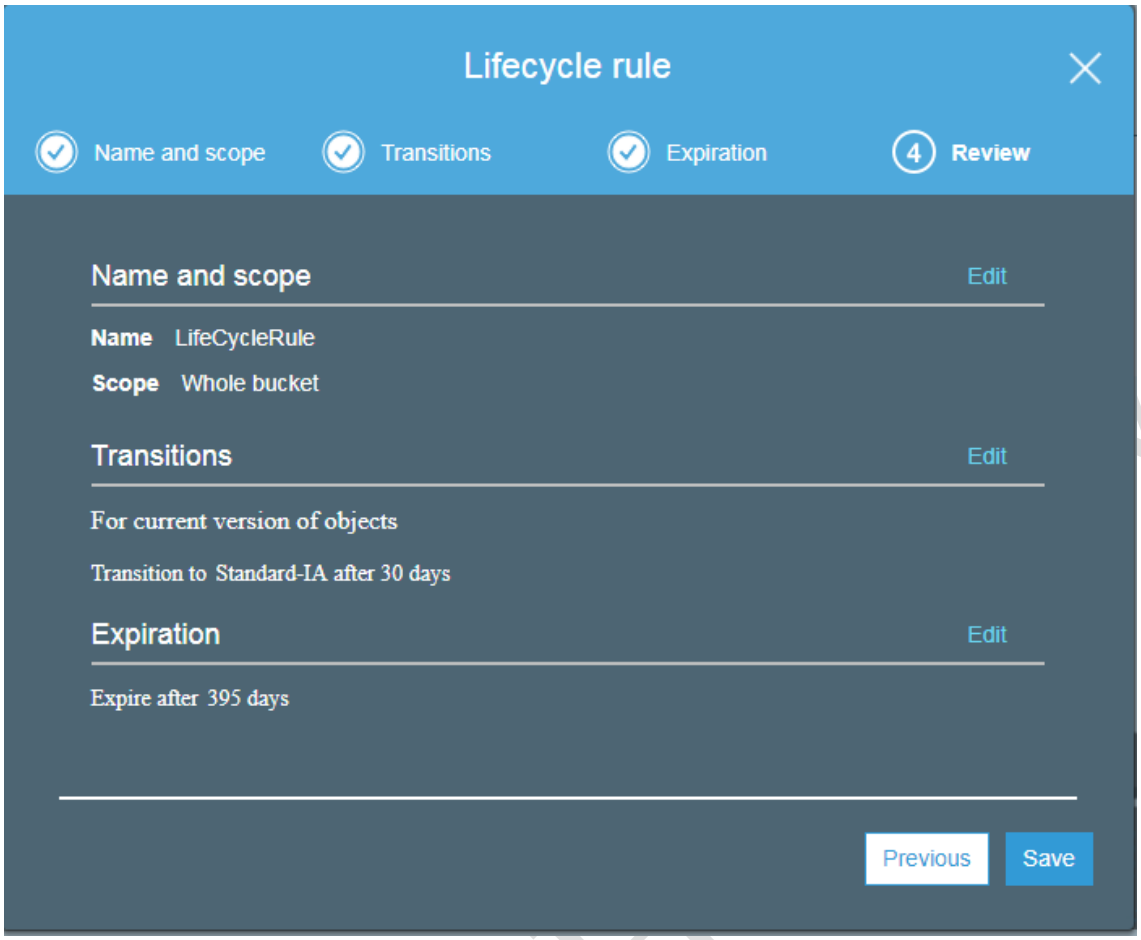
- For S3-IA We need to store the object for minimum of 30 days and for Glacier 60 days from object creation date.



- In Next step we can configure object expirations.
    - For current version Expiration creates a Delete Marker if Versioning is enabled on this bucket.
    - For Previous version object will delete permanently.



- This is the review status for the lifecycle rule that we have created. Review the Lifecycle rule and click on "Save", Created lifecycle rule will apply on bucket.

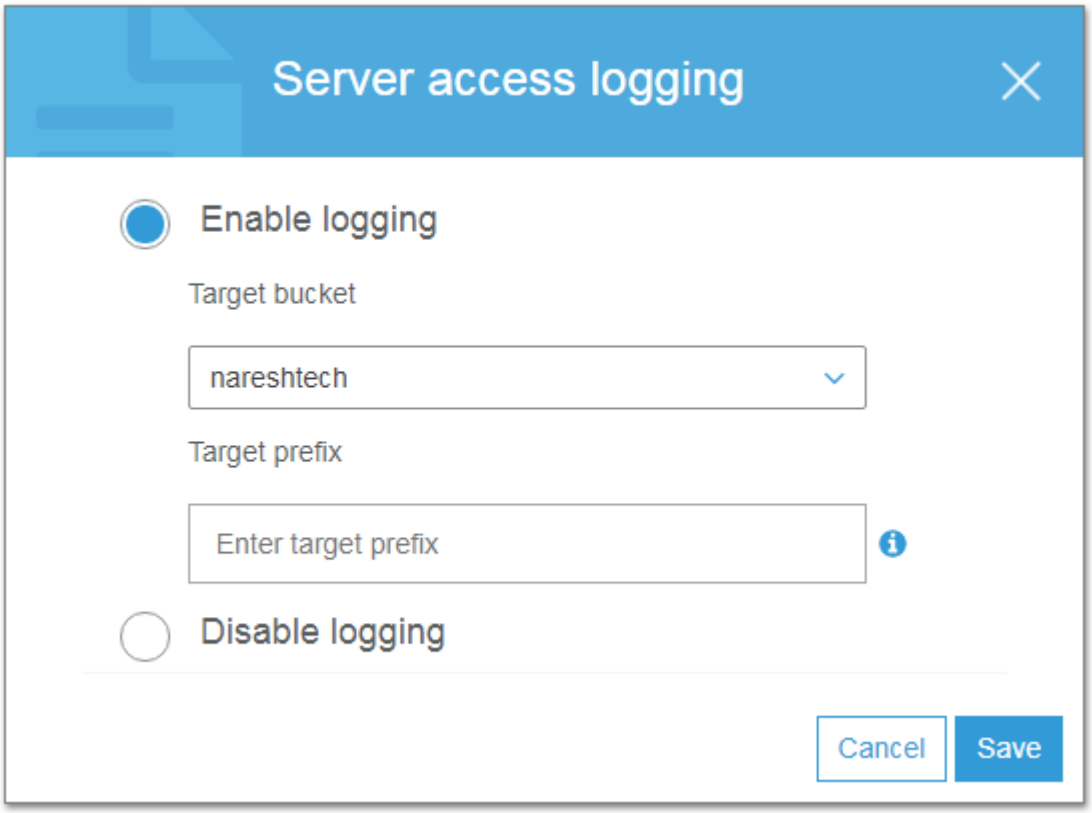


Logging

By enabling logs we can track requests on our Amazon S3 bucket. Logging is off by default. You can enable it from bucket properties.

Every log will contains the below information

- Requestor account and IP address
- Bucket name
- Request time
- Action (GET, PUT, LIST, and so forth)
- Response status or error code



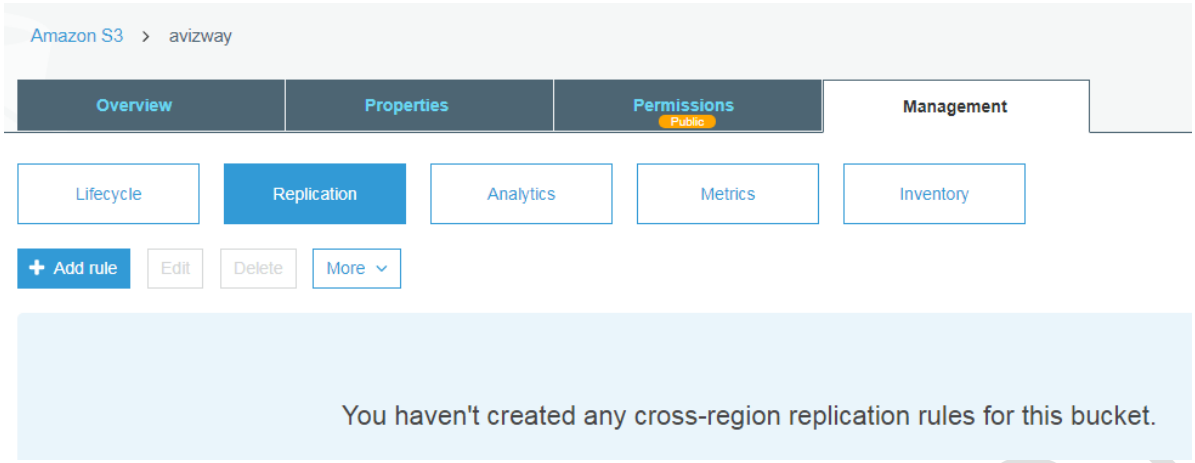
Cross-Region Replication:

With Cross-region replication Amazon S3 allows you to asynchronously replicate all new objects in the source bucket in one AWS region to a target bucket in another region.

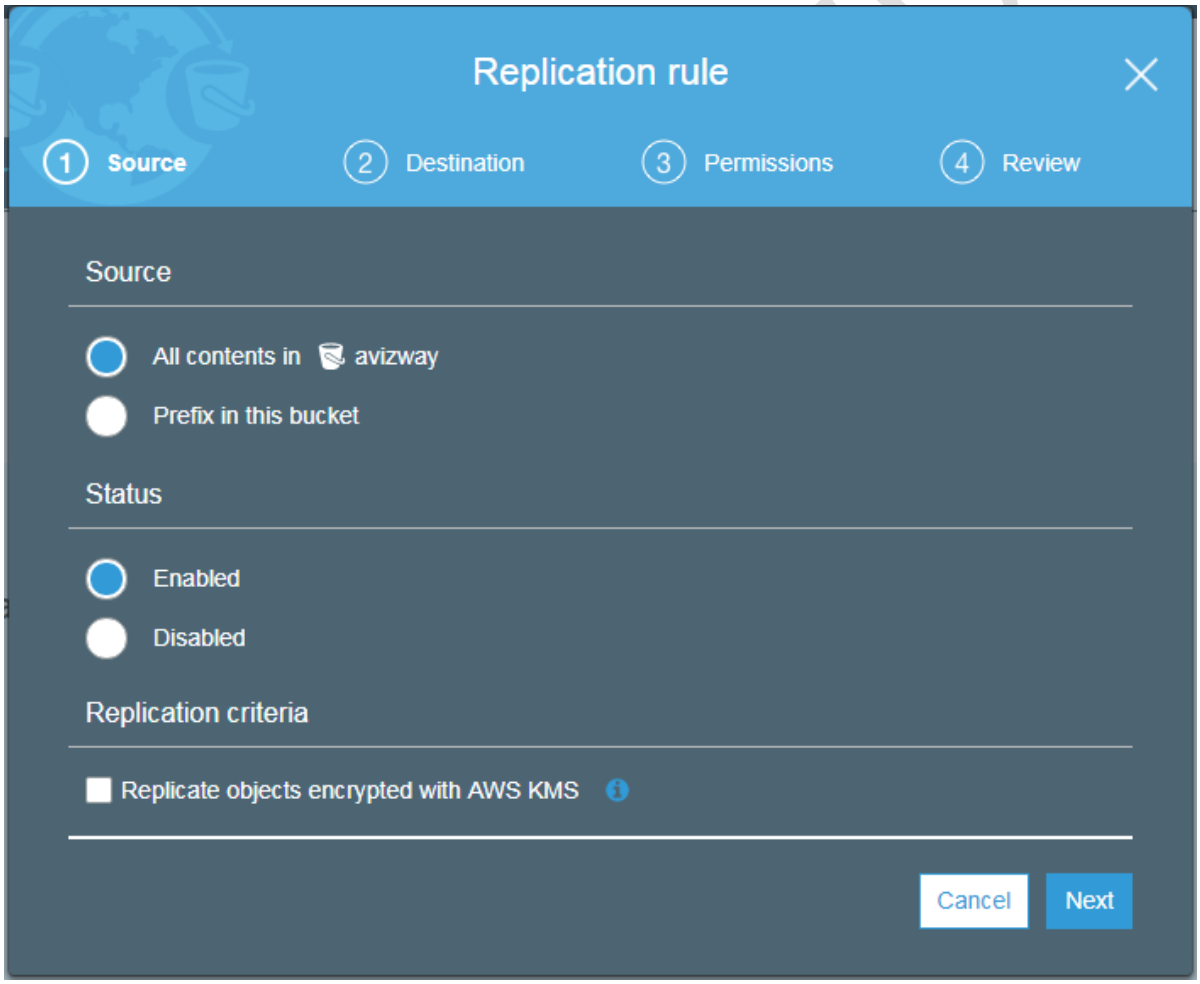
- Versioning must be enabled on both the source and destination buckets.
- Regions must be unique
- Files in an existing bucket are not replicated automatically. All subsequent/future updated files will be replicated automatically.
- You cannot replicate to multiple buckets or use daisy chaining (at this time).
- Delete markers are replicated.
- Deleting individual versions or delete markers will not be replicated.
- Cross-region replication is used to reduce the latency required to access objects in Amazon S3 by placing objects closer to a set of users or to meet requirements to store backup data at a certain distance from the original source data.
- Amazon S3 must have permission to replicate objects from that source bucket to the destination bucket on your behalf.
    - You can grant these permissions by creating an IAM role that Amazon S3 can assume.

Steps to enable cross region replication:

- Select S3 bucket that you want to replicate, Select Replication option under Management.

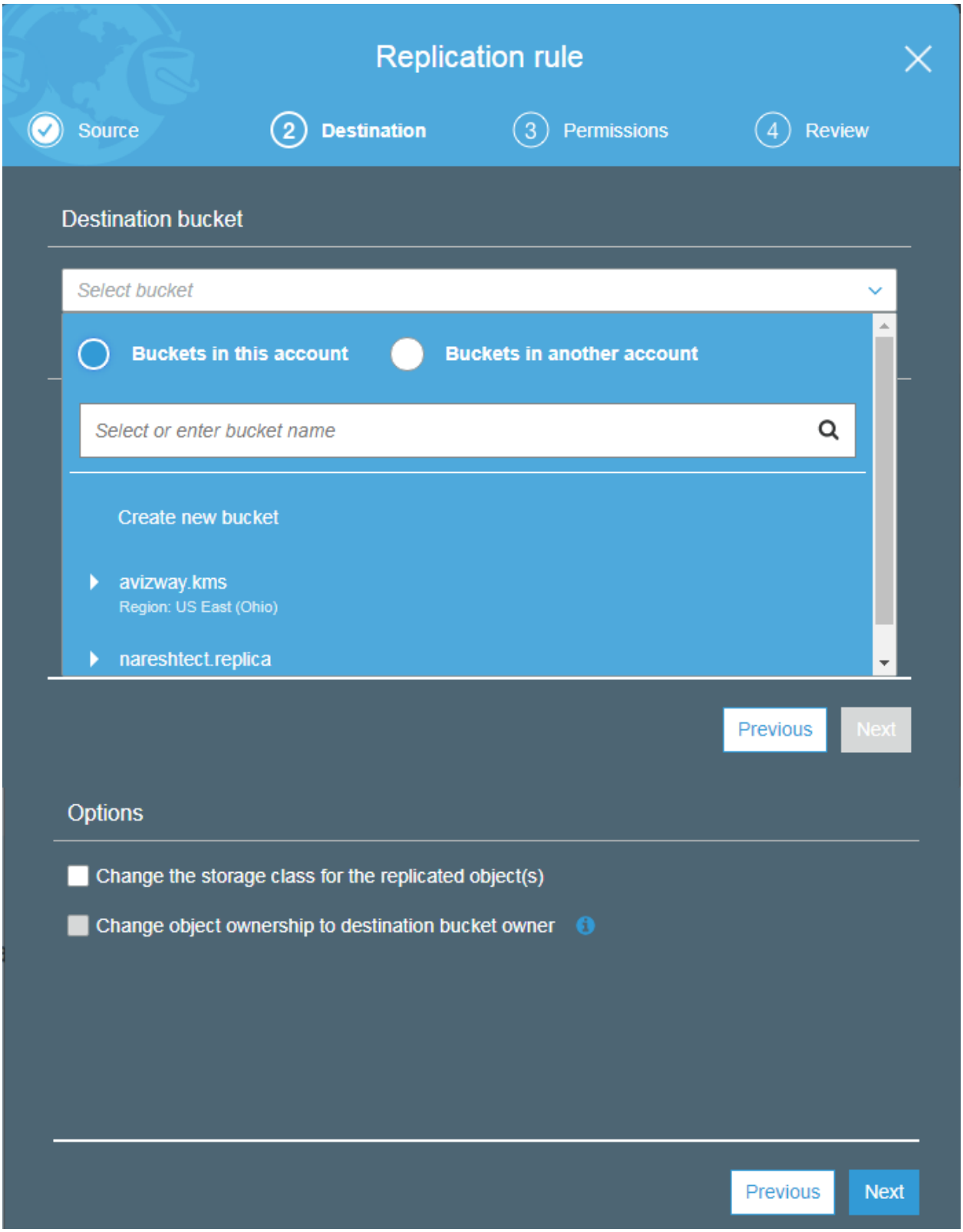


- We can replicate the entire bucket or we can use particular prefixes (i.e; all objects that have names that begin with the string pictures)

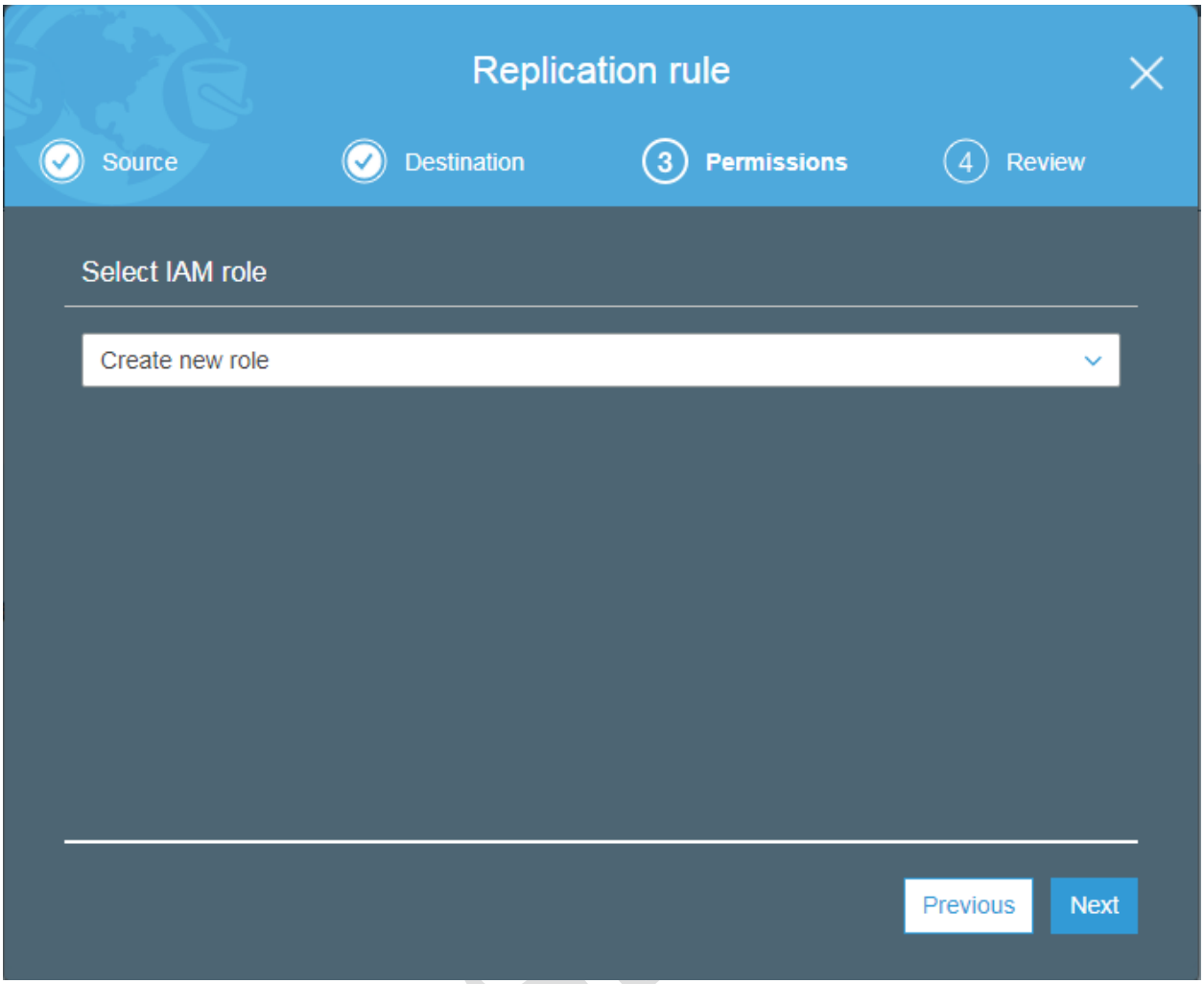


- On the **Destination** tab, under **Destination bucket**, select destination bucket for the replication. You can choose a destination bucket from same account or we can choose to create new bucket, or else we can replicate the data to a destination bucket from a different AWS account.
- Give a valid name for the replication rule

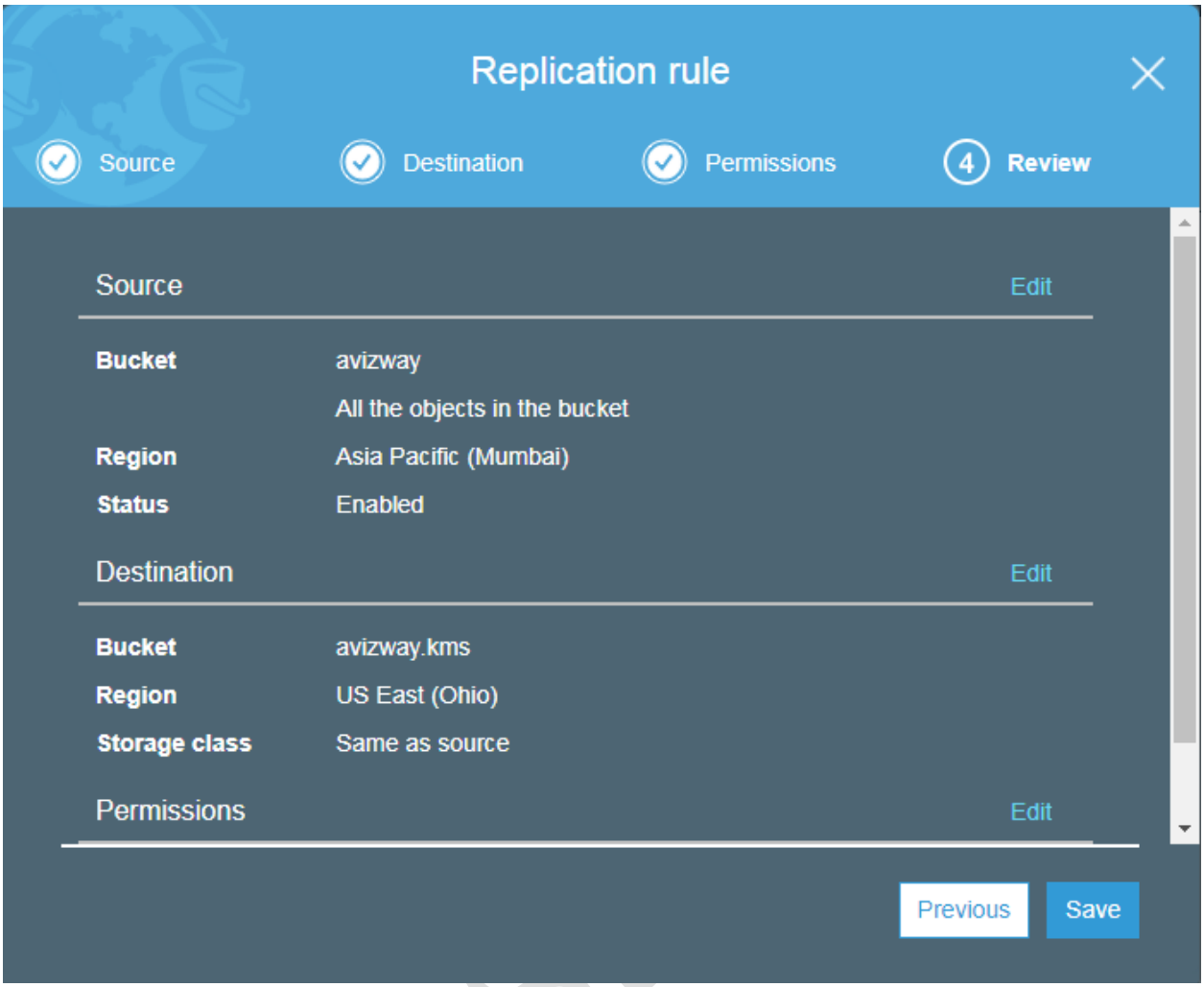
- We can change the object storage class for the destination bucket, if required.



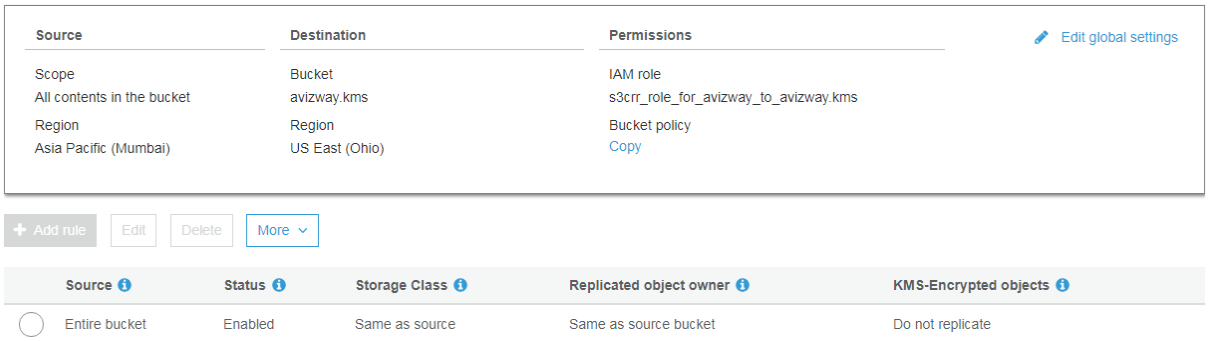
- We have to create an IAM role for replication. Role is "s3crr_role_for_source_to _destination"



- Review and click on save to activate the cross region replication on the bucket.



- After you save your rule, you can edit, enable, disable, or delete your rule on the **Replication** page.

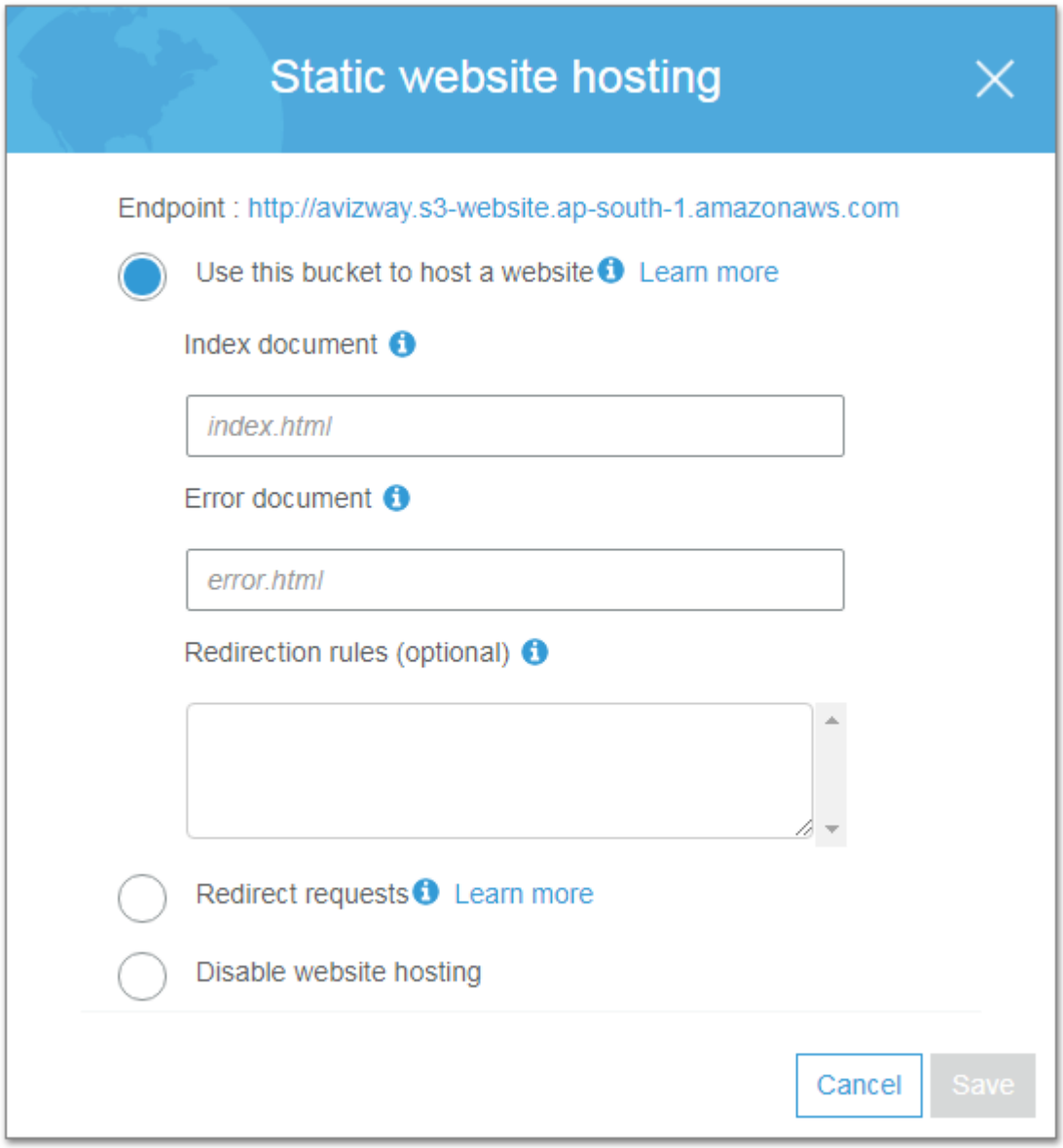


Static Website Hosting

We can host a static website on Amazon Simple Storage Service.

- We need to create a bucket with the same name as the desired website hostname.
- Upload the static files to the bucket (Index.html and error.html).
- Make all the files public, then only website will be readable for all the world.
- Go to Properties of the bucket and Enable static website hosting for the bucket. And mention the specifying an Index.html and an Error.html.

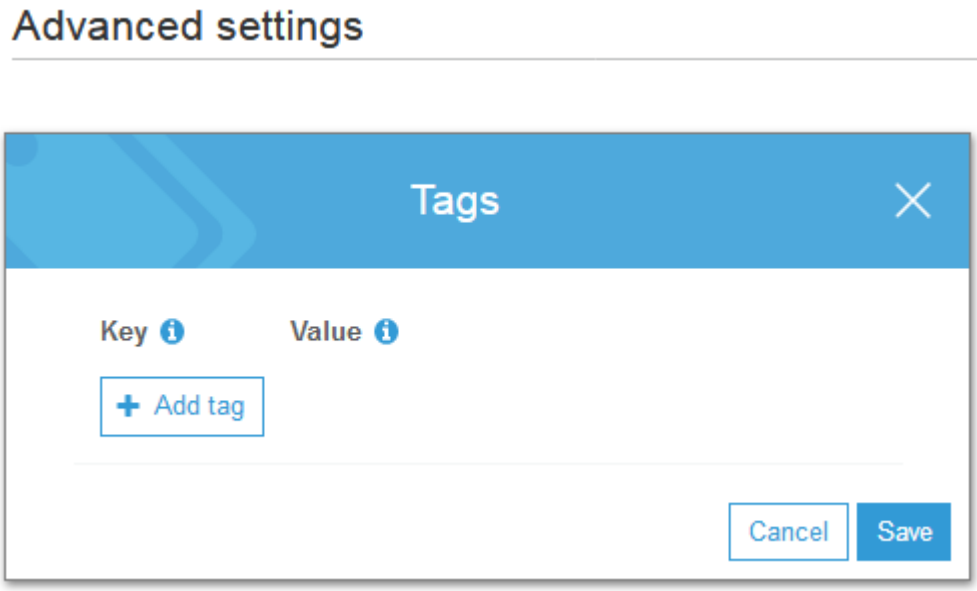
- The website will now be available at the S3 website URL: **<bucket-name>.s3-website-<AWS-region>.amazonaws.com**.
- We have to create a DNS record in Route53 with purchased Domain name, then all the requests to the domain name will point to S3 bucket.
- If required, we can redirect the requests to another bucket also.



Tags:

Tags are combination of keys & values. Each tag is a simple label consisting of a customer-defined key and an optional value that can make it easier to manage, search for, and filter resources.

We can add tags under S3 bucket properties tab.

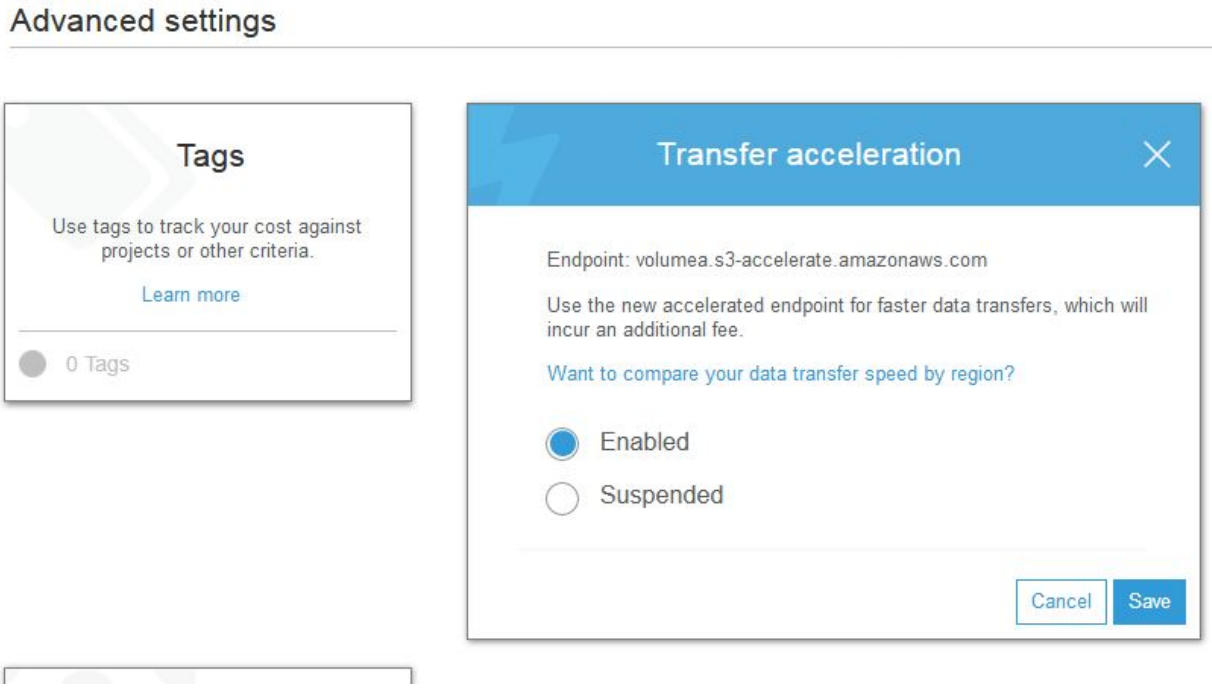


Amazon S3 Transfer Acceleration:

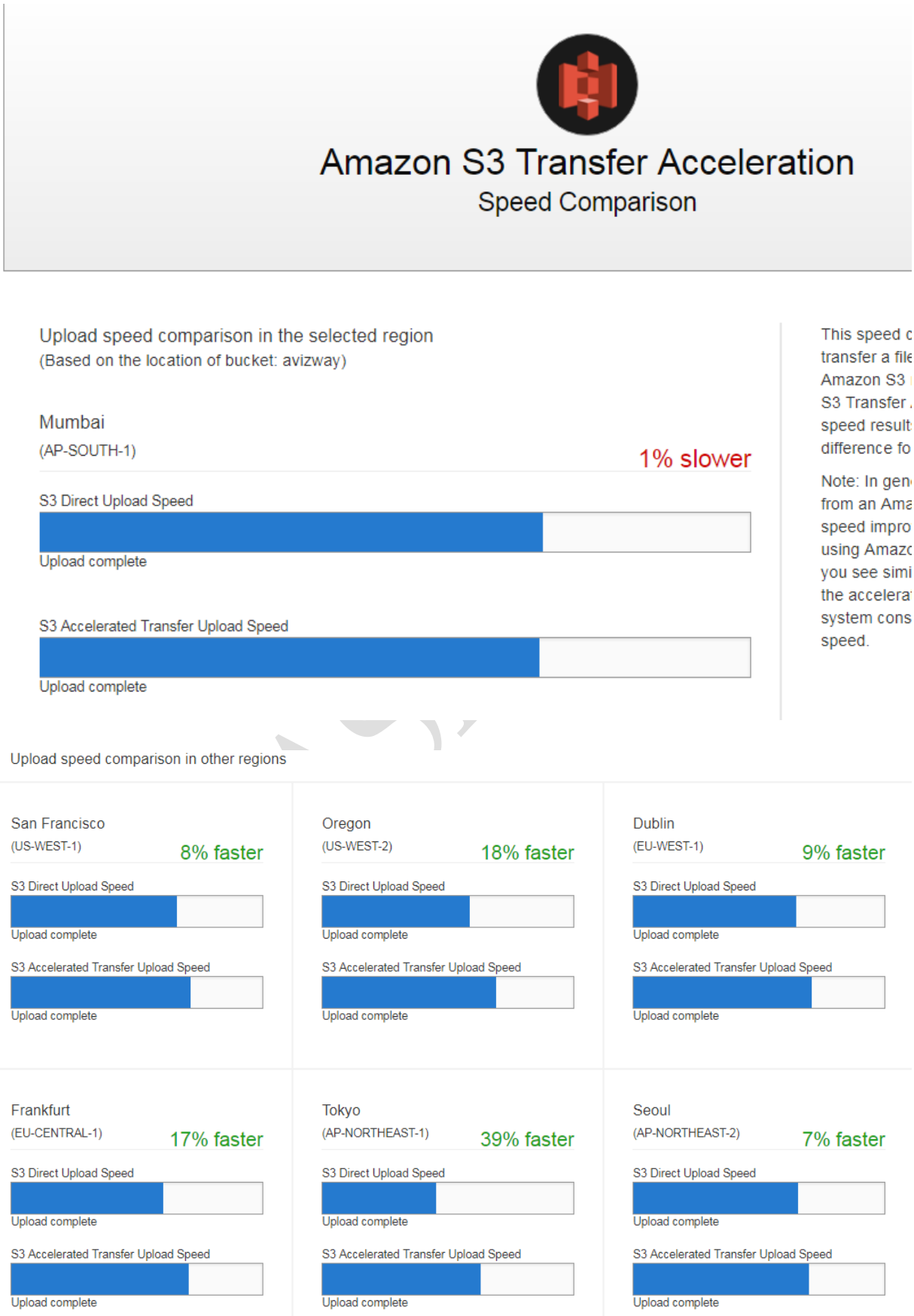
Amazon S3 Transfer Acceleration enables fast, easy, and secure transfers of files over long distances between your client and an S3 bucket. Transfer Acceleration takes advantage of Amazon CloudFront's globally distributed edge locations. As the data arrives at an edge location, data is routed to Amazon S3 over an optimized network path. Additional data transfer charges will apply for this tool.

- **By Using the Amazon S3 Transfer Acceleration Speed Comparison Tool we can compare the** accelerated and non-accelerated upload speeds across Amazon S3 regions.
- The Speed Comparison tool uses multipart uploads to transfer a file from your browser to various Amazon S3 regions with and without using Transfer Acceleration.

You can enable the Transfer acceleration option under S3 Bucket Properties.



Here is a sample result for Transfer acceleration result.



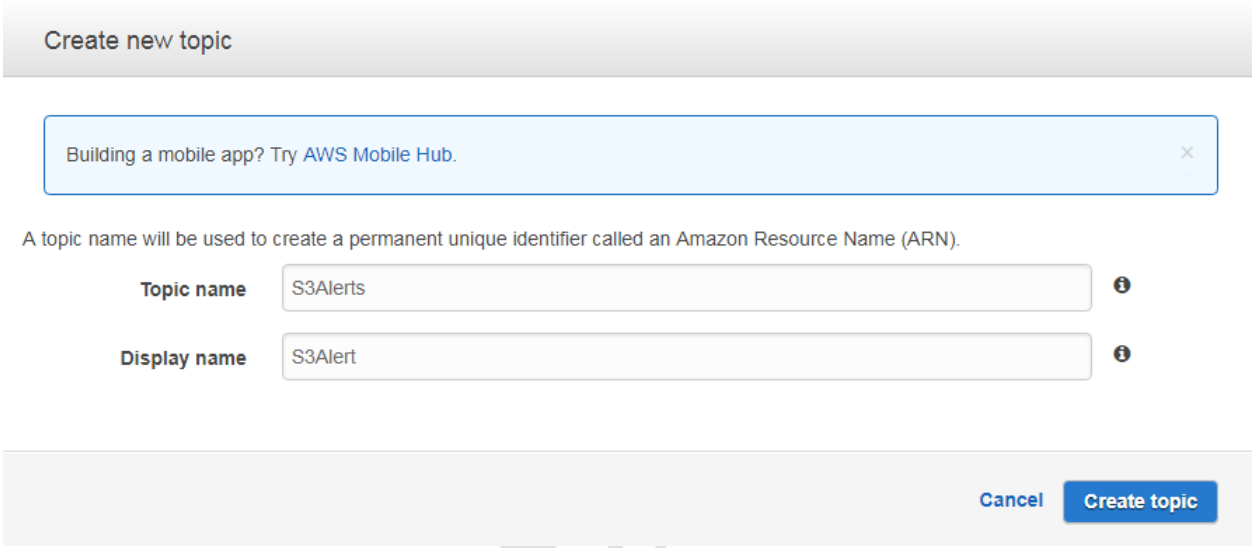
Events

Amazon S3 event notifications can be sent in response to actions taken on objects uploaded or stored in Amazon S3. The Amazon S3 notification feature enables you to receive notifications when certain events happen in your bucket.

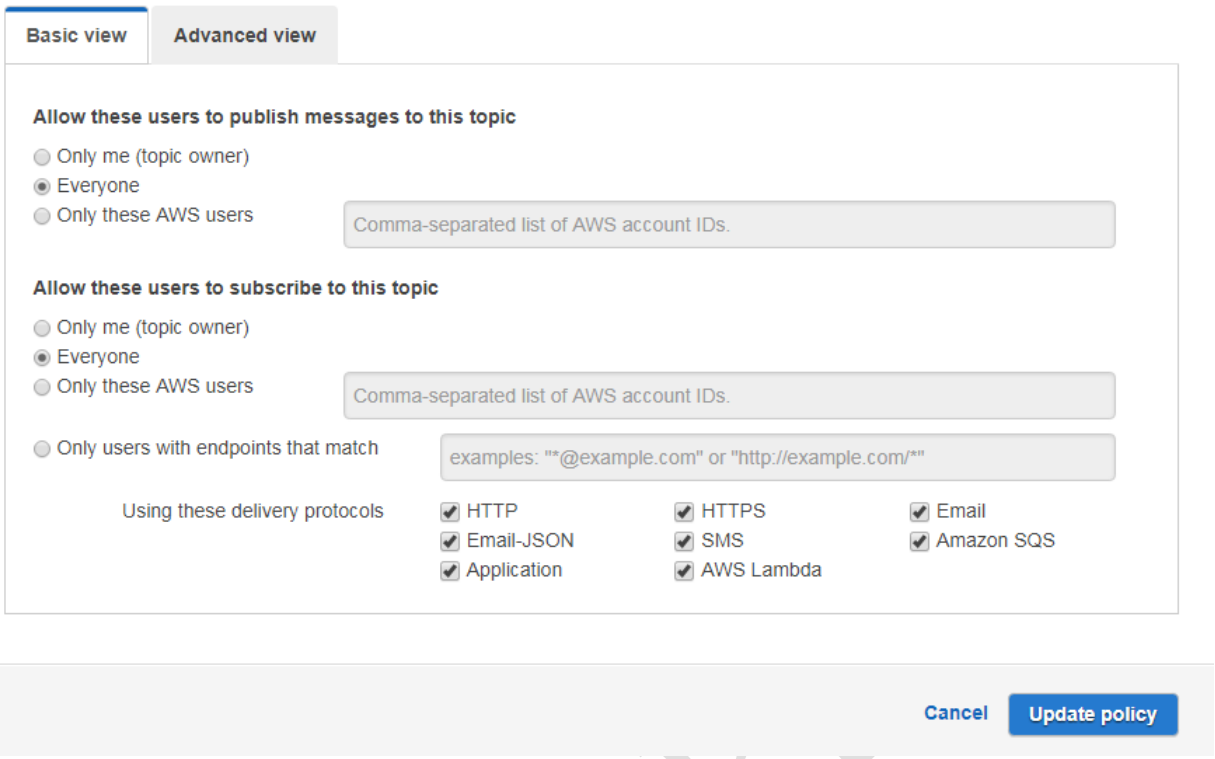
- Notification messages can be sent through either Amazon Simple Notification Service or Amazon Simple Queue Service or delivered directly to AWS Lambda to invoke AWS Lambda functions.

Here is an example to enable Notifications through SNS

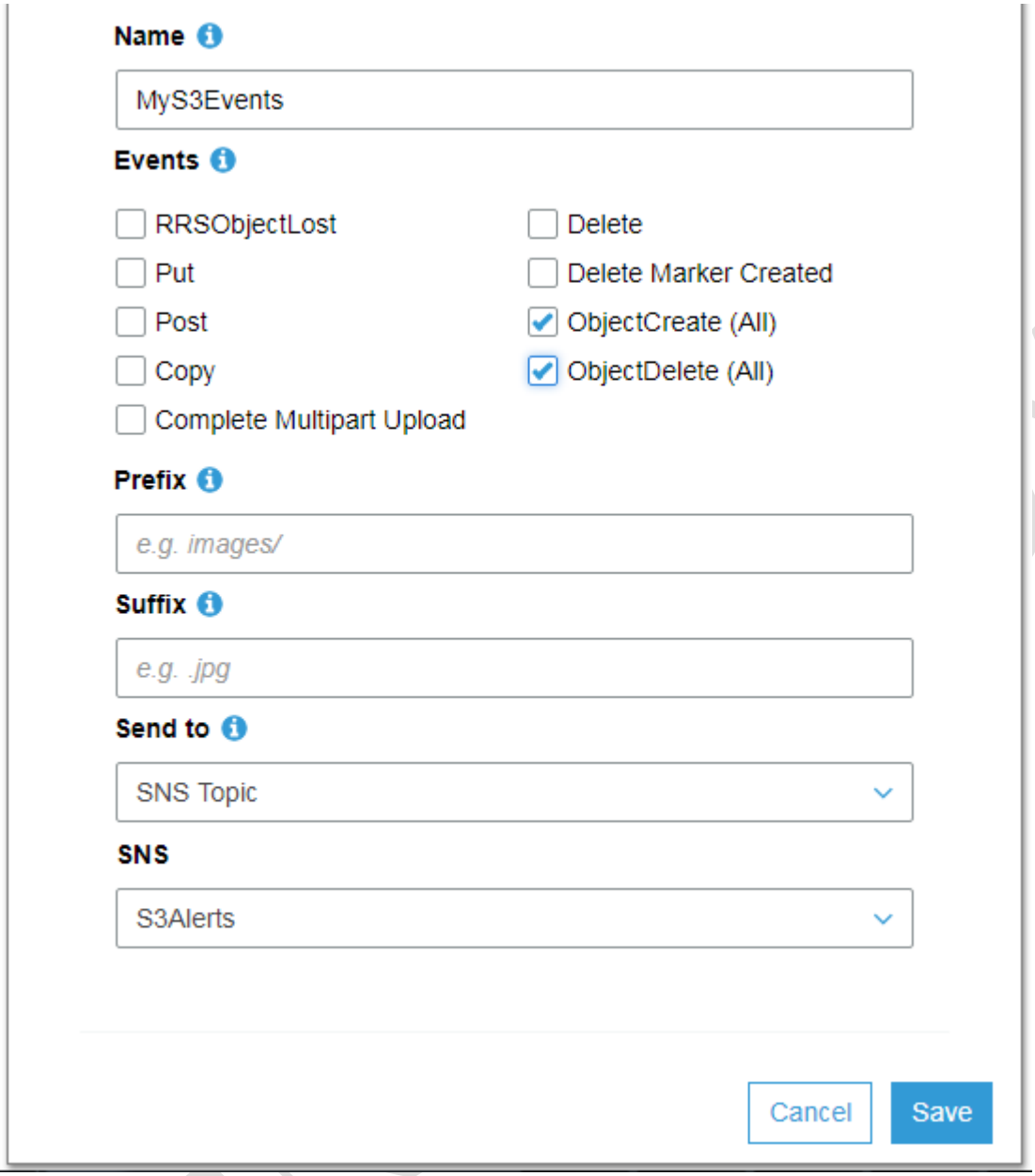
- To set event notifications via SNS, Go to services →Messaging→SNS. In SNS dashboard, we have to create topic in SNS service and edit the Topic Policy to publish through S3.



After creating topic, we have to update the topic policy. Next we can give email id for subscription of notifications. Once we select confirm option from email id then that email got subscribed for event notifications



- Now Go to Properties of S3 bucket and select **Events** →Add notification→give event name→select Events→select SNS topic and select save option.
- We can select the Event type to get notified through the Email.



- When the selected action performed on S3 bucket, Subscribed users to that topic will get a notification.

Inventory:

Amazon S3 inventory is one of the tools Amazon S3 provides to help manage your storage. Amazon S3 inventory provides a comma-separated values (CSV) flat-file output of your objects and their corresponding metadata on a daily or weekly basis for an S3 bucket or a shared prefix.

Requester pays

Generally, bucket owners pay for all Amazon S3 storage and data transfer costs associated with their bucket. If you enable Requester pays on the bucket, instead of bucket owner requested user will pay.

- Anonymous access to that bucket is not allowed, if we want to enable the requester pays on bucket.

Encryption:

We have three types of encryptions available in S3

1. Server-Side Encryption: All SSE performed by Amazon S3 and AWS Key Management Service (Amazon KMS) uses the 256-bit Advanced Encryption Standard (AES).
    - SSE-S3 (AWS-Managed Keys)
    - SSE-KMS (AWS KMS Keys)
    - SSE-C (Customer-Provided Keys)
2. Client-Side Encryption: We can encrypt the data on the client before sending it to Amazon S3. We have to take care about the encryption and Decryption process.
3. In-Transit Encryption
    - We can use SSL API endpoints, this ensures that all data sent to and from Amazon S3 is encrypted while in transit using the HTTPS protocol.

AWS Import/Export:

AWS Import/Export is a service that accelerates transferring large amounts of data into and out of AWS using physical storage appliances, bypassing the Internet. AWS Import/Export supports transfers data directly onto and off of storage devices you own using the Amazon high-speed internal network.

We can ship our own device to AWS by creating a Import/Export job or we can get AWS own hardware appliances.

Here is the three devices available from AWS to transit large set of data from On-premise to AWS environment.

If we are Import/Export our own Disk we can
- Import to EBS
- Import to S3
- Import to Glacier
- Export from S3

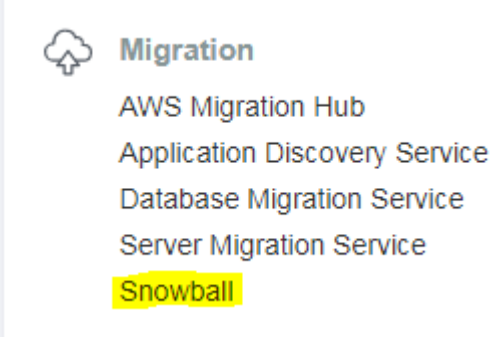
If using Snowball/snowball edge/snow mobile we can
- Import to S3
- Export to S3

AWS SNOWBALL

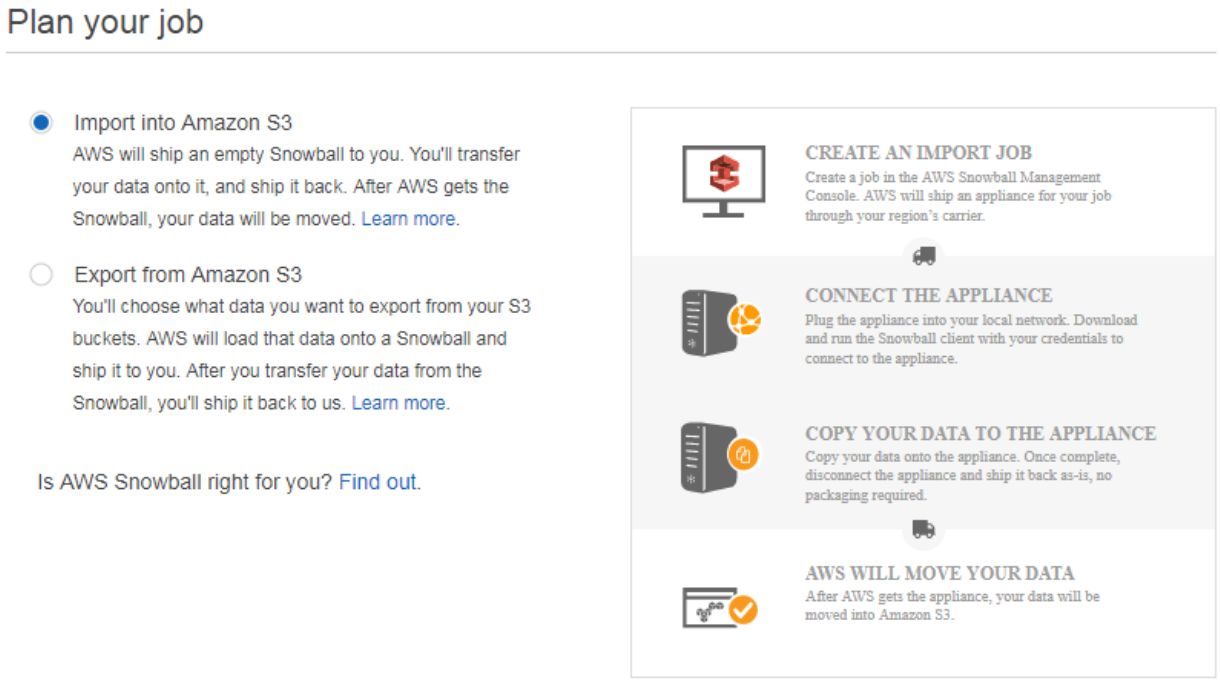
Snowball is a petabyte-scale data transport solution that uses secure appliances to transfer large amounts of data into and out of the AWS cloud.

We don't need to write any code or purchase any hardware to transfer your data. Simply create a job in the AWS Management Console and a Snowball appliance will be automatically shipped to you*. Once it arrives, attach the appliance to your local network, download and run the Snowball client to establish a connection, and then use the client to select the file directories that you want to transfer to the appliance. The client will then encrypt and transfer the files to the appliance at high speed. Once the transfer is complete and the appliance is ready to be returned, the E Ink shipping label will automatically update and you can track the job status via Amazon Simple Notification Service (SNS), text messages, or directly in the Console.

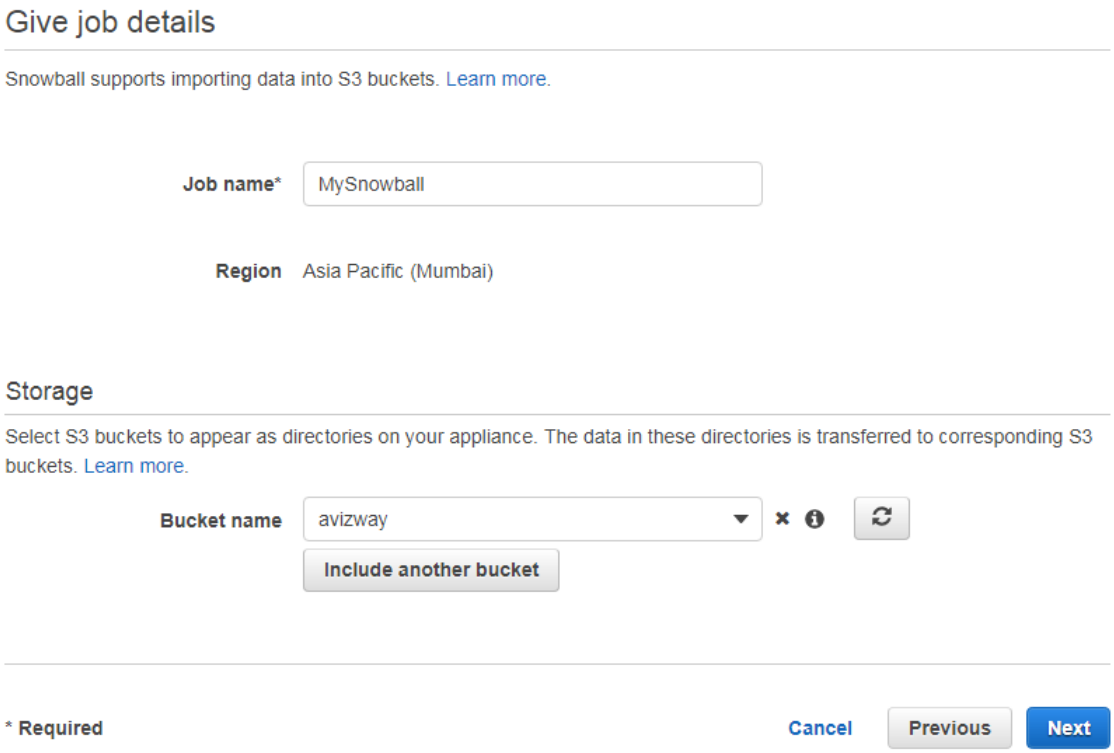
You can find the AWS Snowball under Migration category:



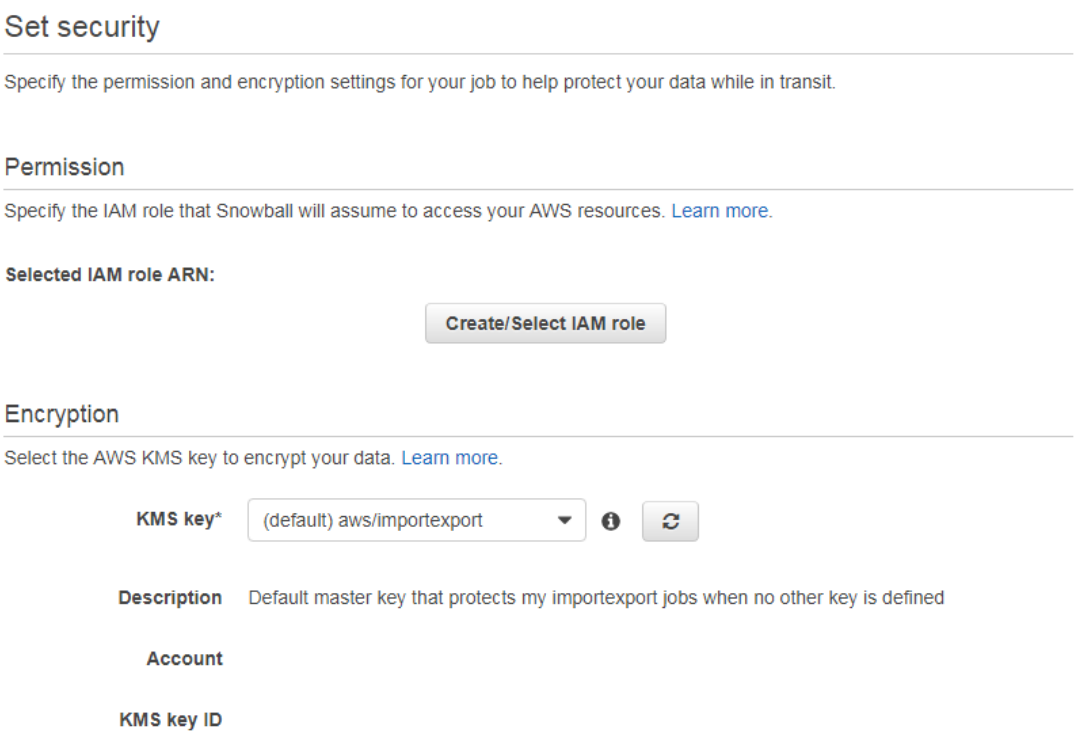
Select the Job type (Import into S3 / Export from S3)



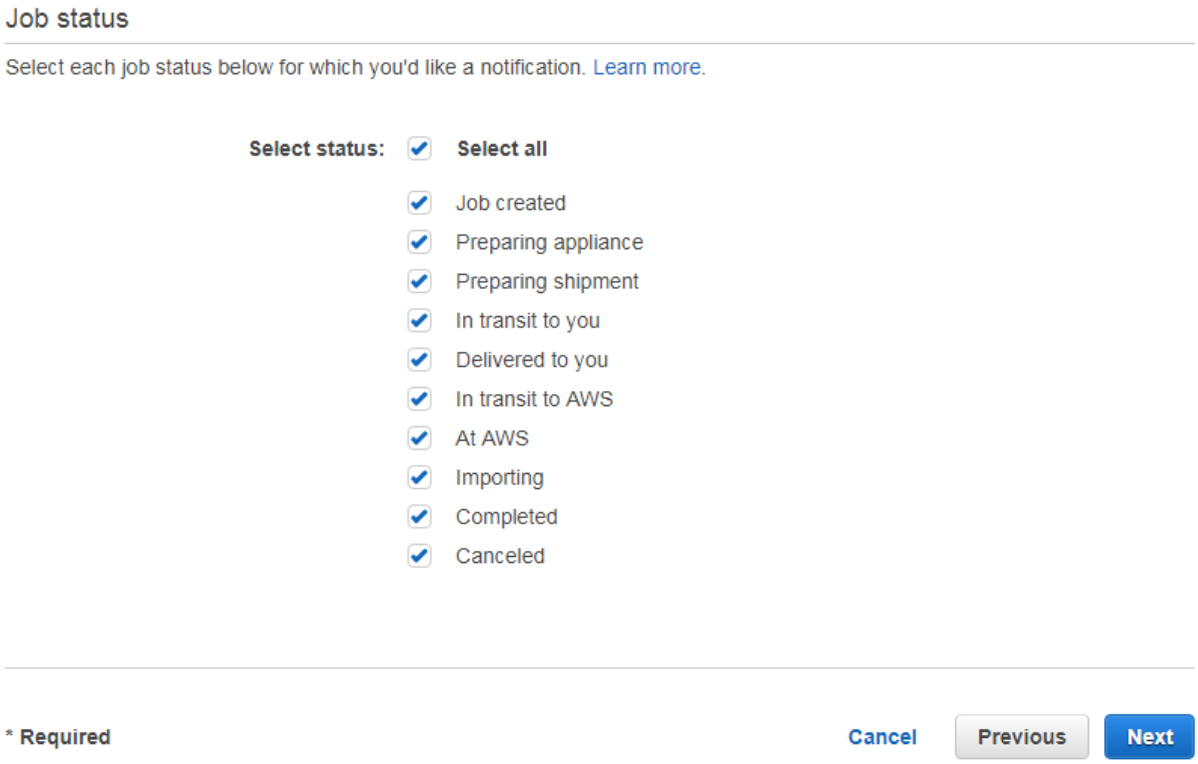
Give the address to ship the snowball device and give a name for the Job and select the S3 bucket to Import/Export the data.



By default all the data will be encrypted by KMS service. And need to create a IAM role to perform the copy operation to our S3 bucket.



We can configure the SNS topics to get notifications about the Snowball device status.



In next step, Review the screen and create the Job. Amazon will send you the snowball device on given address.

Here is the pricing details for snowball device: Service Fee per Job is based on the appliance capacity. We have 50 TB device and 80 TB device. First 10 days of onsite usage are free* and each extra onsite day is $15

Snowball 50 TB: $200

Snowball 80 TB: $250

Snowball Edge:AWS Snowball Edge is a 100TB data transfer device with on-board storage and compute capabilities. It also have compute capability that is approximately the equivalent of an EC2 m4.4xlarge instance. 16 vCPU & 64 GB RAM

AWS Snowmobile:Snowmobile is a Exabyte-Scale Data transfer service used to move extremely large amount of data to AWS.Capacity : 100 PB

With Snowmobile, we can move 100 petabytes of data in as little as a few weeks, plus transport time. If you transfer same with 1Gbps connection, it may take more than 20 years.

We need to request the amazon with the given url to get the snowmobile https://aws.amazon.com/contact-us/aws-sales/

AWS Direct Connect

AWS Direct Connect makes it easy to establish a dedicated network connection from your premises to AWS. Using AWS Direct Connect, you can establish private connectivity between AWS and your datacenter, office, or colocation environment, which in many cases can reduce your network costs,

increase bandwidth throughput, and provide a more consistent network experience than Internet-based connections.

AWS Direct Connect lets you establish a dedicated network connection between your network and one of the AWS Direct Connect locations. Using industry standard 802.1q VLANs, this dedicated connection can be partitioned into multiple virtual interfaces. This allows you to use the same connection to access public resources such as objects stored in Amazon S3 using public IP address space, and private resources such as Amazon EC2 instances running within an Amazon Virtual Private Cloud (VPC) using private IP space, while maintaining network separation between the public and private environments. Virtual interfaces can be reconfigured at any time to meet your changing needs.

Service Advantages:

1. Reduces Your Bandwidth Costs
2. Consistent Network Performance
3. Compatible with all AWS Services
4. Private Connectivity to your Amazon VPC
5. Elastic

EC2 (ELASCTIC COMPUTE CLOUD)

Amazon Elastic Compute Cloud (Amazon EC2)

Amazon EC2 is AWS primary web service that provides resizable compute capacity in the cloud. Amazon EC2 allows you to acquire compute through the launching of virtual servers called **instances.** Instance is nothing but a Virtual Server.

Instance Types:

The instance type defines the virtual hardware supporting an Amazon EC2 instance. There are many instance types available, based on the following dimensions:

- General purpose
- Compute Optimized (vCPUs)
- GPU Compute
- Memory Optimized
- Storage Optimized

General Purpose: General purpose instance family provides a balance of compute, memory, and network resources, and it is a good choice for many applications.

Compute Optimized (vCPUs): Compute Optimized instances are optimized for compute-intensive workloads and delivers high performance computing, batch processing.

GPU Compute: GPU Compute instances are next generation of general purpose GPU computing instances. We can use GPU instances for 3D visualizations, graphics-intensive remote workstation, 3D rendering, application streaming, video encoding, Machine/Deep learning, high performance computing and other server-side graphics workloads.

Memory Optimized: Memory Optimized category instances are most suitable for high performance databases, distributed memory caches, in-memory analytics, large-scale, enterprise-class, and In-memory applications.

Storage Optimized:

Optimized category instances are most suitable for low latency, very high random I/O performance, high sequential read throughput and provide high IOPS and NoSQL databases like Cassandra, MongoDB, Redis and In-memory databases.

Compute optimized	For workloads requiring significant processing
Memory optimized	For memory-intensive workloads
Storage optimized	For workloads requiring high amounts of fast SSD storage
GPU-based instances	Intended for graphics and general-purpose GPU compute workloads

Instance launch pricing Options:

- On-Demand Instances
- Reserved Instances
- Spot Instances

On-Demand Instances:

The price **per hour** for each instance type published on the AWS website represents the price for On-Demand Instances.

- On-Demand is most flexible pricing option, as it doesn't requires up-front commitment.
- We will have control over when the instance is launched and when it is terminated.
- Suitable for unpredictable workloads.

Reserved Instances:

When purchasing a reserved instance we have to specify the instance type and Availability Zone for that Reserved Instance and achieves a lower effective hourly price for that instance for the duration of the reservation. You can select duration from 1 Yr to 3 yrs.

We have two offering classes in RI: Convertible or Standard

- We have three payment options for Reserved Instances.
  - **All Upfront**—Pay for the entire reservation up front. There is no monthly charge for the customer during the term.
  - **Partial Upfront**—Pay a portion of the reservation charge up front and the rest in monthly installments for the duration of the term.
  - **No Upfront—**Pay the entire reservation charge in monthly installments for the duration of the term.
- We can save up to 75 percent over on-demand hourly rate if we reserve instance through Reserved Option.

Spot Instances:

For workloads that are not time critical and are tolerant of interruption, Spot Instances offer the greatest discount.

- We can specify the price they are willing to pay for a certain instance type.
- When the bid price is above the current Spot price, we'll get the requested instance.
- These instances will operate like all other Amazon EC2 instances, and the customer will only pay the Spot price for the hours that instance(s) run.

The instances will run until:

- Till we terminate them manually.
- The Spot price goes above our bid price.
- There is not enough unused capacity to meet the demand for Spot Instances.
- If Amazon EC2 needs to terminate a Spot Instance, the instance will receive a termination notice providing a **two-minute warning prior to termination**.
- If we terminate Instance manually we have to pay for Partial hours, if amazon terminates we will not get charged for partial hours.

Tenancy Options:

Shared Tenancy: Shared tenancy is the default tenancy model for all Amazon EC2 instances. A single host machine may house instances from different customers. (One host may share with multiple customers).

Dedicated Instances: Dedicated Instances run on hardware that's dedicated to a single customer. As a customer runs more Dedicated Instances, more underlying hardware may be dedicated to their account.

Dedicated Host: An Amazon EC2 Dedicated Host is a physical server with Amazon EC2 instance capacity fully dedicated to a single customer's use. We will get complete control over which specific host runs an instance at launch.

Placement Groups: A placement group is a logical grouping of instances within a single Availability Zone.

- Placement groups enable applications to participate in a low-latency, 10 Gbps network.
- Recommended for applications that benefit from low network latency, high network throughput, or both.
- Only certain types of instances can be launched in a placement group.
- A placement group can't span multiple Availability Zones.
- The name you specify for a placement group must be unique within your AWS account.
- AWS recommend homogenous instances within placement groups.
- You can't merge placement groups.
- You can't move an existing instance into a placement group.

Amazon Machine Images (AMIs)

The Amazon Machine Image (AMI) defines the initial software that will be on an instance when it is launched.

- The Operating System (OS) and its configuration
- The initial state of any patches
- Application or system software

All AMIs are based on x86 OSs, either Linux or Windows.
We can launch instances from four options

1. Published by AWS
2. AWS Marketplace
3. Generated from existing Instance (Custom AMIs)
4. Uploaded Virtual Servers

Accessing an Instance: We can access our Instances by Using Public DNS, Public IP address and Elastic IP addresses.

Public DNS: When we launch instance, we will get one Public DNS associated for that instance.

- Public DNS will generate automatically. We can't specify
- We can found this information in Instance description
- We cannot transfer this Public DNS to another instance.
- We will get public DNS when the instance is in running state.

Public IP:

- When we launch instance, we will get one Public IP address also.
- AWS will allocate this address, no option to select specific IP.
- This is unique on the Internet.

Elastic IP

- An Elastic IP address is a static IPv4 address designed for dynamic cloud computing. An Elastic IP address is associated with your AWS account.

- To use an EIP address, we have to generate one to our AWS account, and then associate it with your instance or a network interface.
- We can disassociate an EIP address from a resource, and reassociate it with a different resource.
- A disassociated EIP address remains allocated to your account until you manually release it.
- By Default, we are limited to 5 Elastic IP addresses per region.

Steps to get EIP Address:

1. Login to AWS account and navigate to Amazon EC2 console.
2. In the navigation pane, choose **Elastic IPs**.
3. Choose Allocate new address.
4. Select Allocate. Close the confirmation screen.

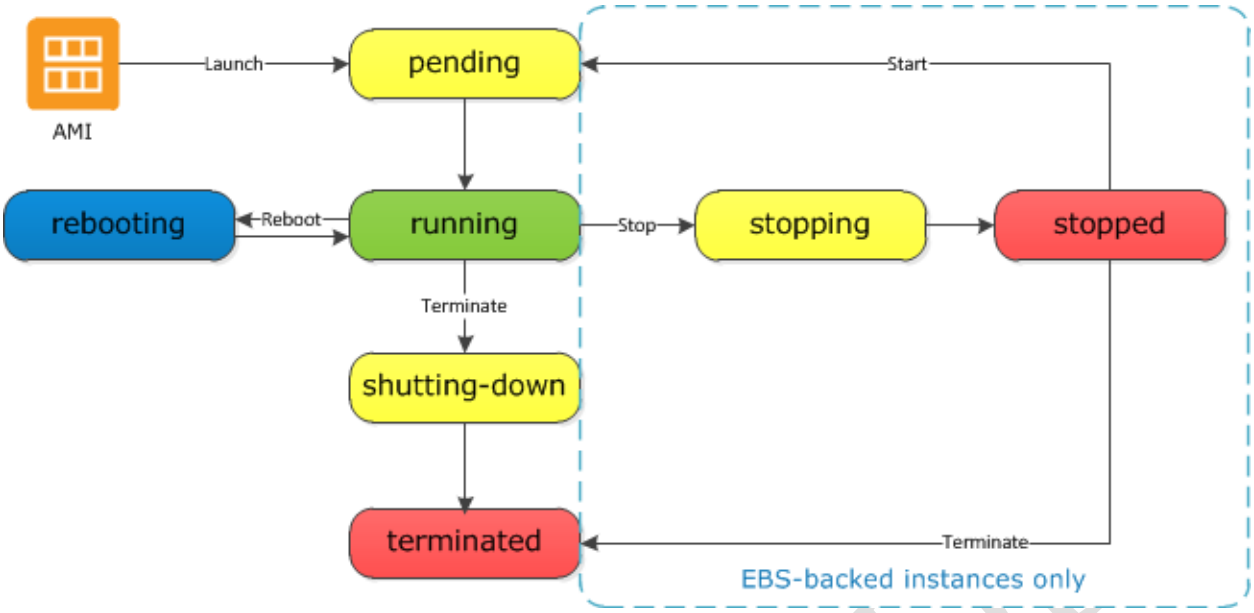
Enhanced networking: reduces the impact of virtualization on network performance by enabling a capability called Single Root I/O Virtualization (SR-IOV). This results in more Packets per Second, lower latency, and less jitter.

Current Generation Instance Types:

Instance Family	Current Generation Instance Types
General purpose	t2.nano | t2.micro | t2.small | t2.medium | t2.large | t2.xlarge | t2.2xlarge | m4.large | m4.xlarge | m4.2xlarge | m4.4xlarge | m4.10xlarge | m4.16xlarge | m5.large | m5.xlarge | m5.2xlarge | m5.4xlarge | m5.12xlarge | m5.24xlarge
Compute optimized	c4.large | c4.xlarge | c4.2xlarge | c4.4xlarge | c4.8xlarge | c5.large | c5.xlarge | c5.2xlarge | c5.4xlarge | c5.9xlarge | c5.18xlarge
Memory optimized	r4.large | r4.xlarge | r4.2xlarge | r4.4xlarge | r4.8xlarge | r4.16xlarge | x1.16xlarge | x1.32xlarge | x1e.xlarge | x1e.2xlarge | x1e.4xlarge | x1e.8xlarge | x1e.16xlarge | x1e.32xlarge
Storage optimized	d2.xlarge | d2.2xlarge | d2.4xlarge | d2.8xlarge | h1.2xlarge | h1.4xlarge | h1.8xlarge | h1.16xlarge | i3.large | i3.xlarge | i3.2xlarge | i3.4xlarge | i3.8xlarge | i3.16xlarge
Accelerated computing	f1.2xlarge | f1.16xlarge | g3.4xlarge | g3.8xlarge | g3.16xlarge | p2.xlarge | p2.8xlarge | p2.16xlarge | p3.2xlarge | p3.8xlarge | p3.16xlarge

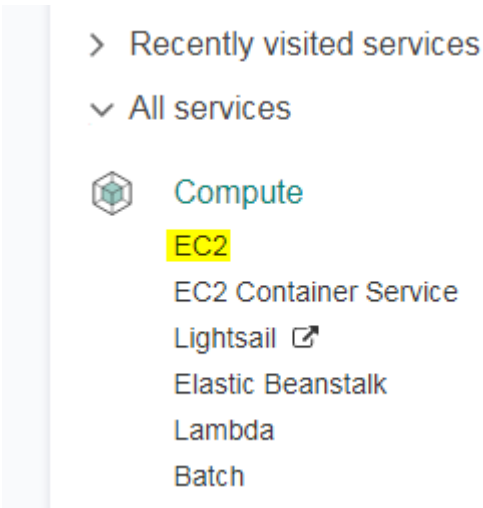
Instance Lifecycle

Here is a diagram that represents the transitions between instance states. **Note:**We can't stop and start an instance store-backed instance

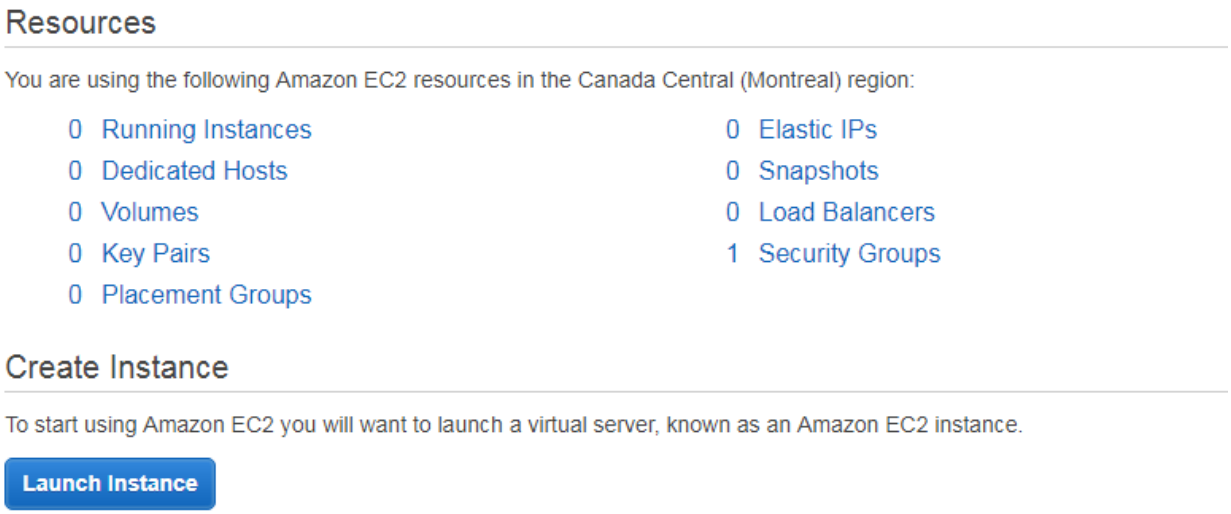


Instance launch process:

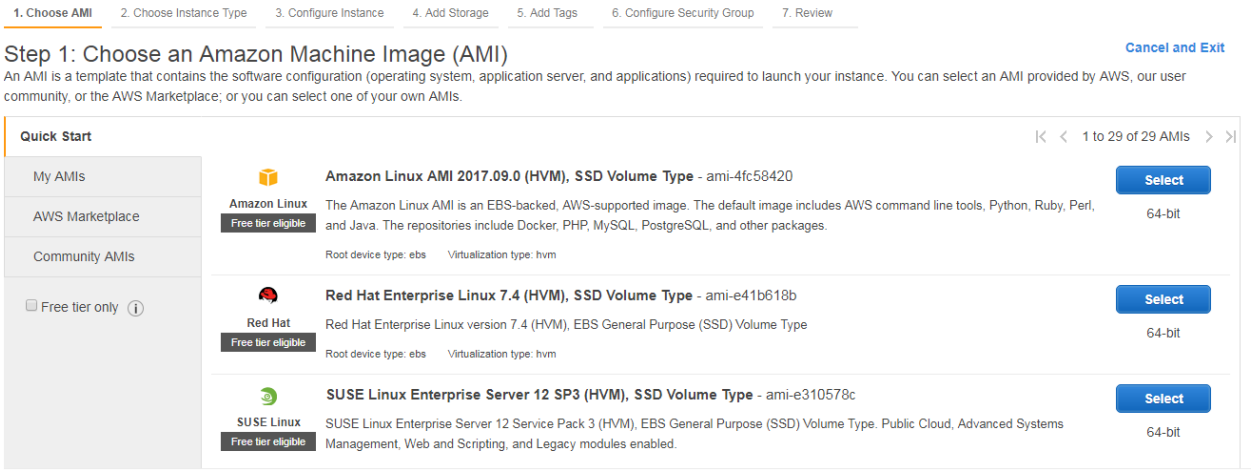
Login to Your AWS Account, Select and switch to the required Region and find **EC2 under Compute Section**.



Select the Launch instance option and it will launch an instance launch wizard.



I want to launch an Amazon Linux AMI, so selecting Amazon Linux AMI from the Quick Start menu.

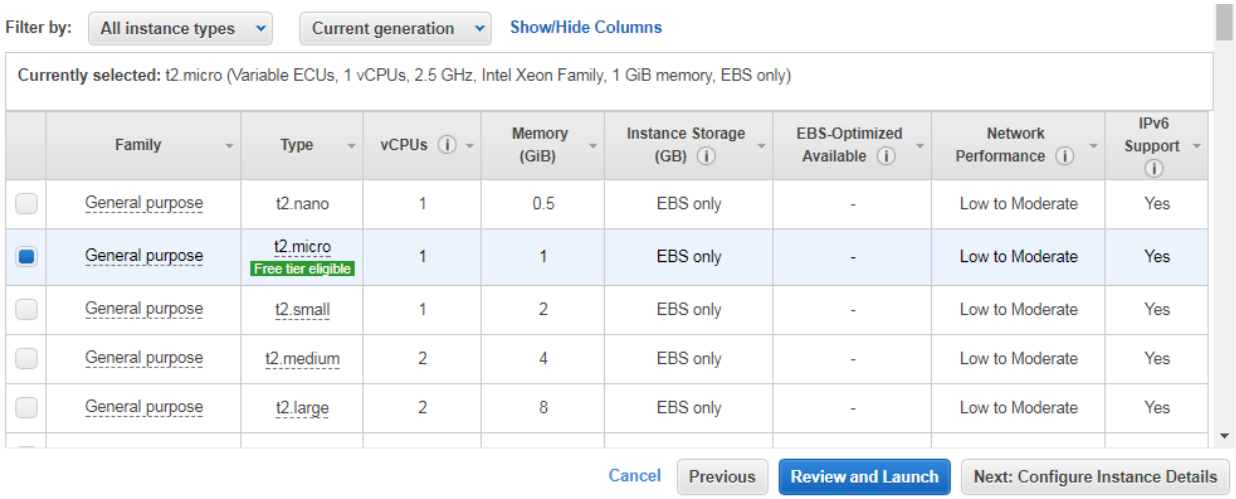


- We have Windows and Linux operating systems available here in Quick start option
- Along with the Quick Start option, you can also spin up your instances using the AWS Marketplace and the Community AMIs section. Both these options contains list of customized AMIs that have been created by either third-party companies or by developers and can be used for a variety of purposes.

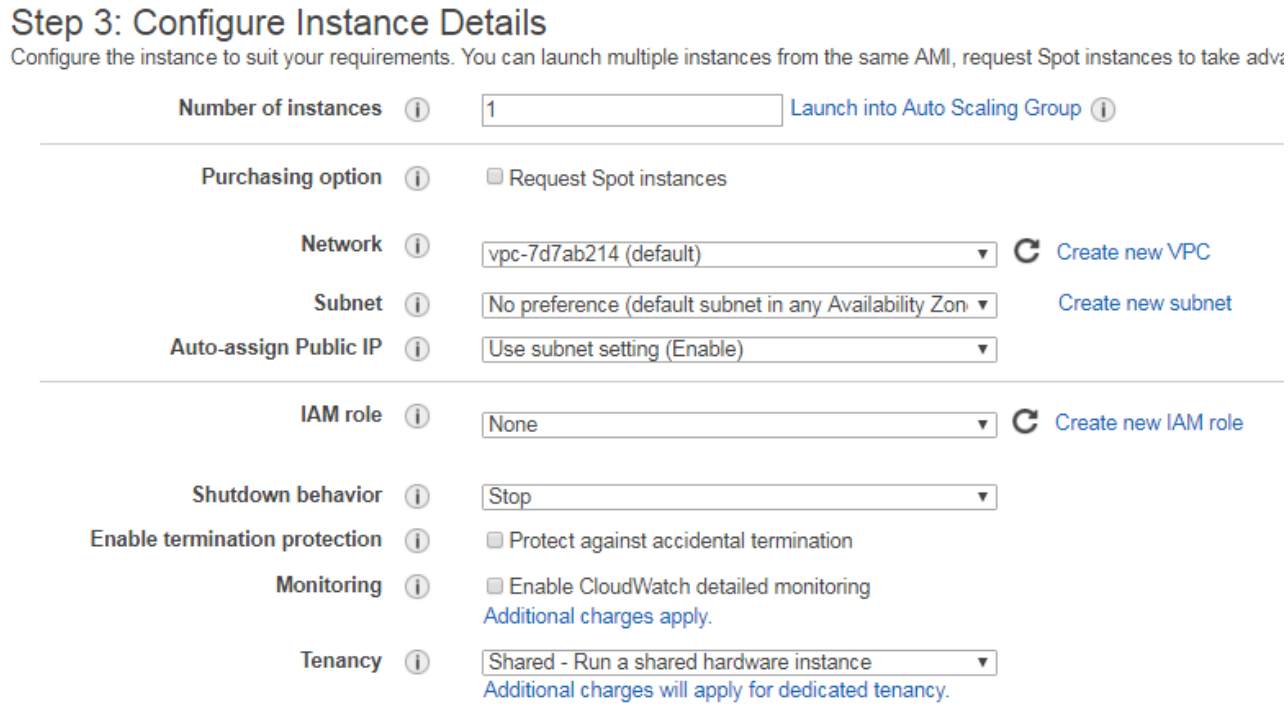
Choose an instance type

In the next step, we have to select the instance type as per our requirements. You can filter instances according to their families.

We can use the general purpose t2.micro instance type, which is comes under the free tier eligibility and configuration is 1 vCPU and 1 GB of RAM.



Configure instance details



Here is Step 3, we have multiple options,

Number of instances: You can specify how many instances the wizard should launch using this field. By default, the value is always set to one single instance.

Purchasing option: We can this instance under spot instances request. For now let's leave this option.

Network: Select the default **Virtual Private Cloud** (**VPC**) network that is displayed in the dropdown list. We can even go ahead and create a new VPC network for this instance, but we will leave and will see VPC in later chapters.

Subnet: select the **Subnet** in which you wish to deploy your new instance.
You can either choose to have AWS select and deploy your instance in a particular subnet from an available list or you can select a particular choice of subnet on your own.

Auto-assign Public IP: Each instance that you launch will be assigned a Public IP. We are going to use this public IP to connect to our Instance over Internet.

IAM role: You can additionally select a particular IAM role to be associated with your instance.

Shutdown behavior: This option allows us to select whether the instance should stop or be terminated when issued a shutdown request. In this case, we have opted for the instance to stop when it is issued a shutdown command.

Enable termination protection: Select this option in case you wish to protect your instance against accidental deletions. It adds additional step for instance termination. If, we enable this option, we need to manually Disable to terminate the instance.
Monitoring: By default, AWS will monitor few basic parameters about your instance for free, but if you wish to have an in-depth insight into your instance's performance, then select the **Enable CloudWatch detailed monitoring** option. But you'll get charged for detailed monitoring.

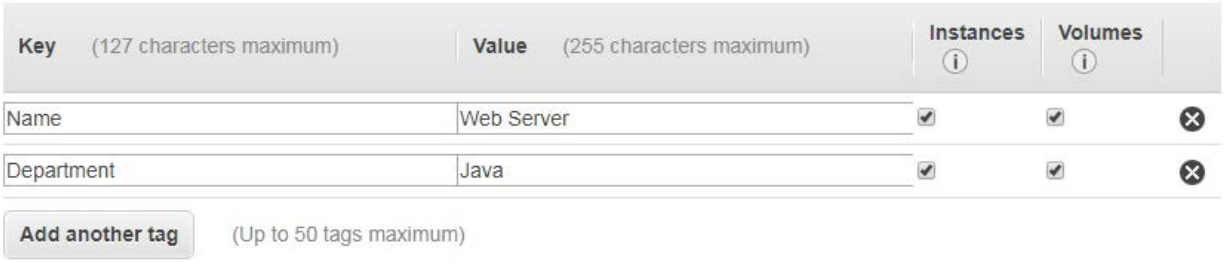
Tenancy: We can choose to run our instances on physical servers fully dedicated for your use. The use of host tenancy will request to launch instances onto dedicated hosts.

Bootstrapping We can configure instances and install applications programmatically when an instance is launched. The process of providing code to be run on an instance at launch is called bootstrapping.
On Linux instances this can be shell script, and on Windows instances this can be a batch style script or a PowerShell script.

Step 4: Add Storage
We can add EBS volumes to your instances. To add new volumes, simply click on the Add New Volume button. This will provide you with options to provide the size of the new volume along with its mount points. There is an 8 GB volume already attached to our instance. This is the t2.micro instance's root volume.

- Try to keep the volume size under 30 GB, It'll comes under free tier eligibility.
- We can create volumes and attach to instance even after instance launch also.

Step 5: Add Tags
Tags are normal key-value pairs. We can manage our AWS resources with Tags options. We can create maximum of 50 tags per Instance.

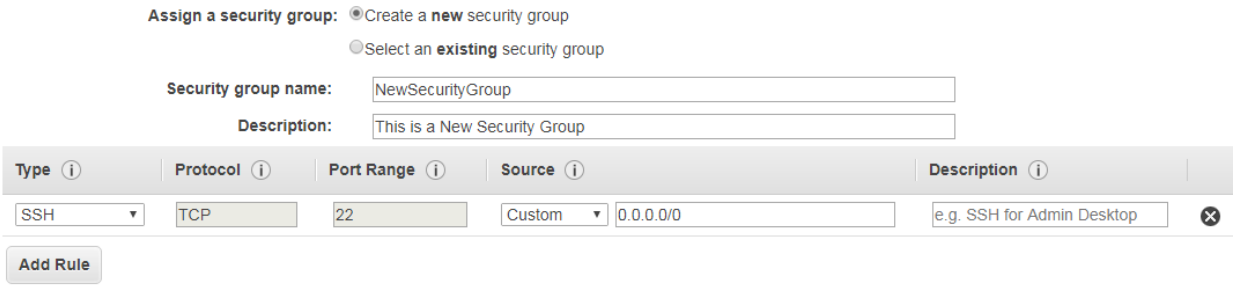


Step 6: Configure Security Group

A security group is a set of firewall rules that control the traffic for our instance. We can add rules to allow specific traffic to reach our instance.

For example, if you want to set up a web server and allow Internet traffic to reach our instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. We can create a new security group or select from an existing one.

Select the **Create a new security group** option and enter the suitable Security group name and Description.



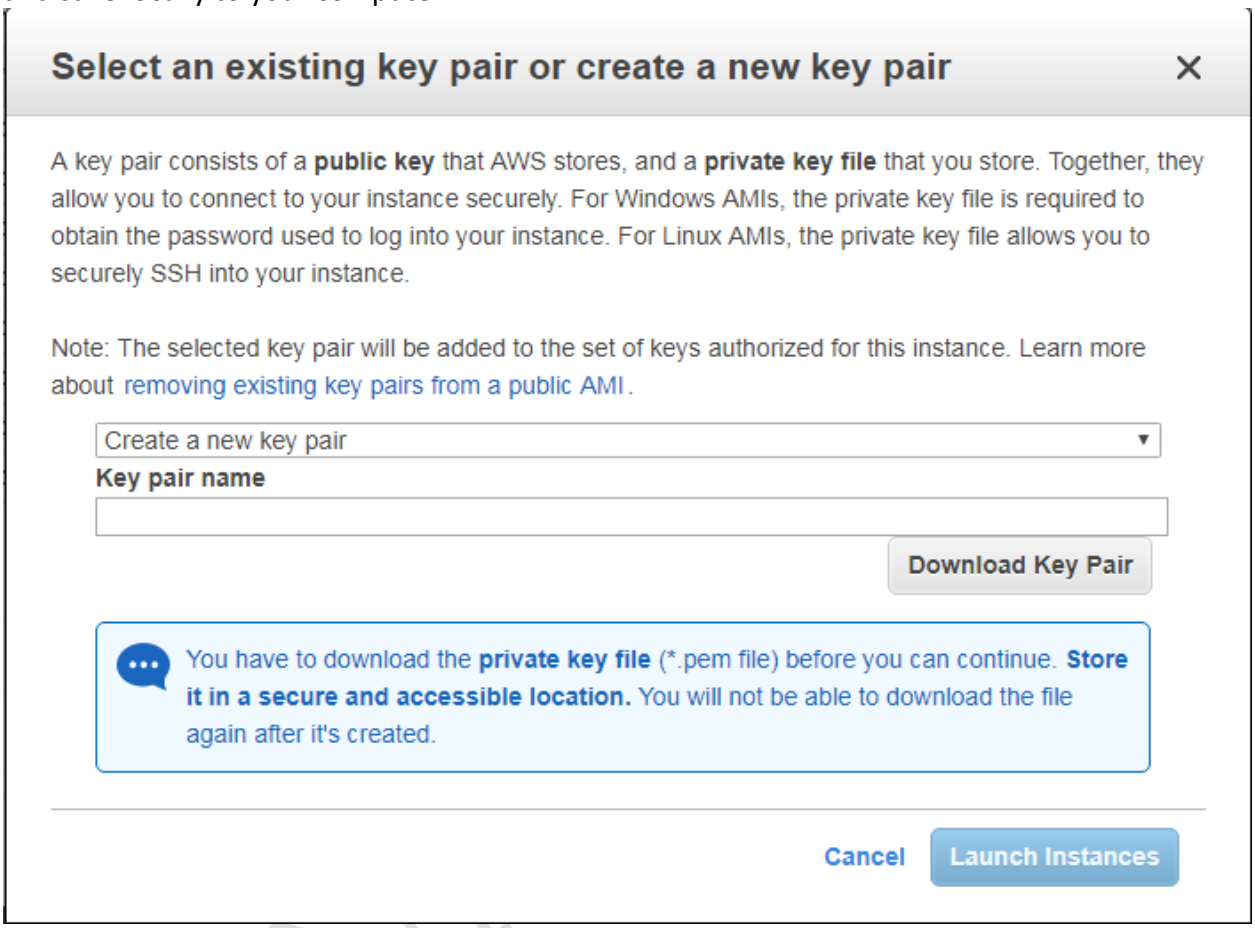
- You need to open SSH to Connect Linux machines, RDP for Windows machines. HTTP and HTTPS if webservers.
- We can give 0.0.0.0/0 to connect this instance from any network and subnet.
- We can select custom option and give the particular Network's public IP, then the service will be available for that particular network only.

Some Important points about Security Groups:

- You can create up to 500 security groups for each Amazon VPC.
- You can add up to 50 inbound and 50 outbound rules to each security group. If you need to apply more than 100 rules to an instance, you can associate up to five security groups with each network interface.
- You can specify allow rules, but not deny rules. This is an important difference between security groups and ACLs.
- By default, no inbound traffic is allowed until you add inbound rules to the security group.
- By default, new security groups have an outbound rule that allows all outbound traffic.
- Security groups are **stateful**. This means that responses to allowed inbound traffic are allowed to flow outbound regardless of outbound rules and vice versa.
- You can change the security groups with which an instance is associated after launch, and the changes will take effect immediately

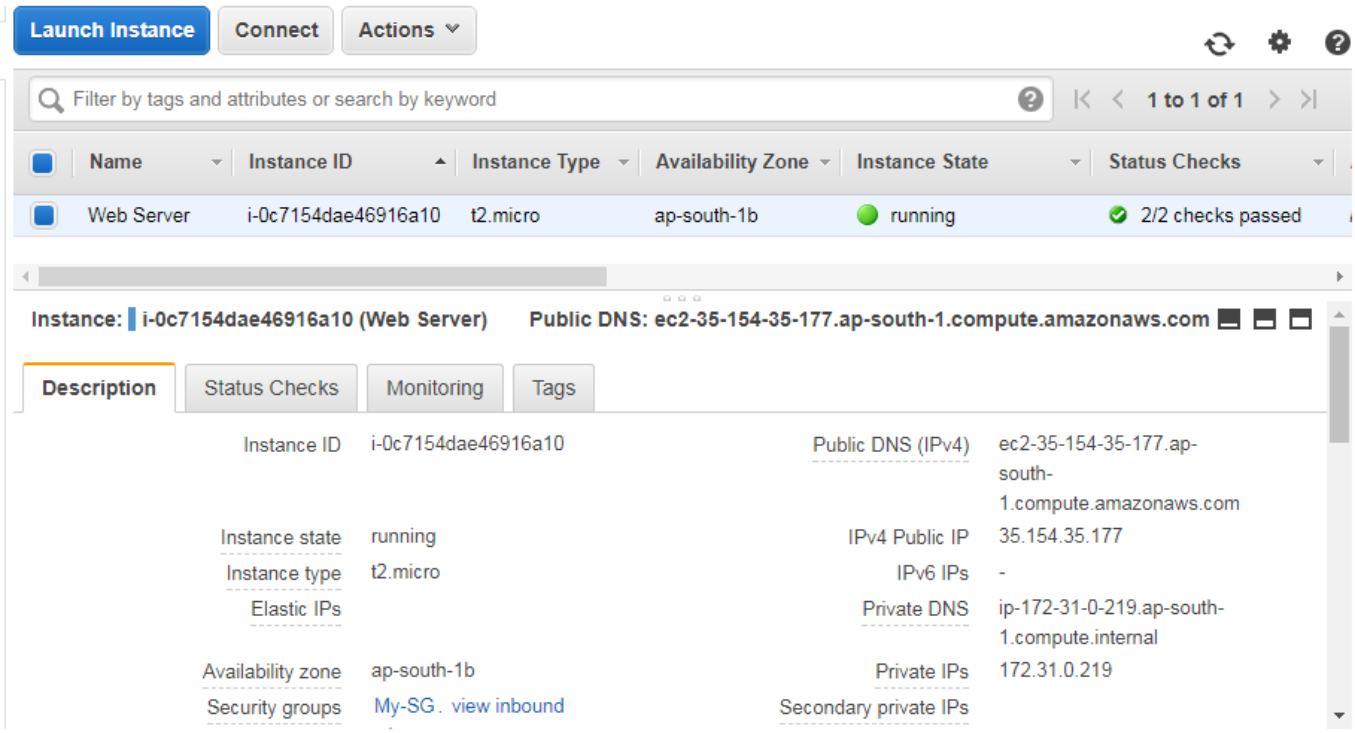
Step 7: Review Instance Launch

Here in step 7, we will get review screen. We will get complete summary of our instance's configuration details, including the AMI details, instance type selected, instance details, and so on. If all the details are correct, then simply go and click on the Launch option.
Then we have to associate a key pair to our instance.
A key pair is basically a combination of a public and a private key, which is used to encrypt and decrypt your instance's login info. AWS generates the key pair for you which you need to download and save locally to your computer.



Once a key pair is created and associated with an instance, we need to use that key pair itself to access the instance. We will not be able to download this key pair again so, save it in a secure location.

Select the **Create a new key pair** option from the dropdown list and provide a suitable name for your key pair as well. Click on the **Download Key Pair** option to download the **.PEM file**. Once completed, select the **Launch Instance** option.

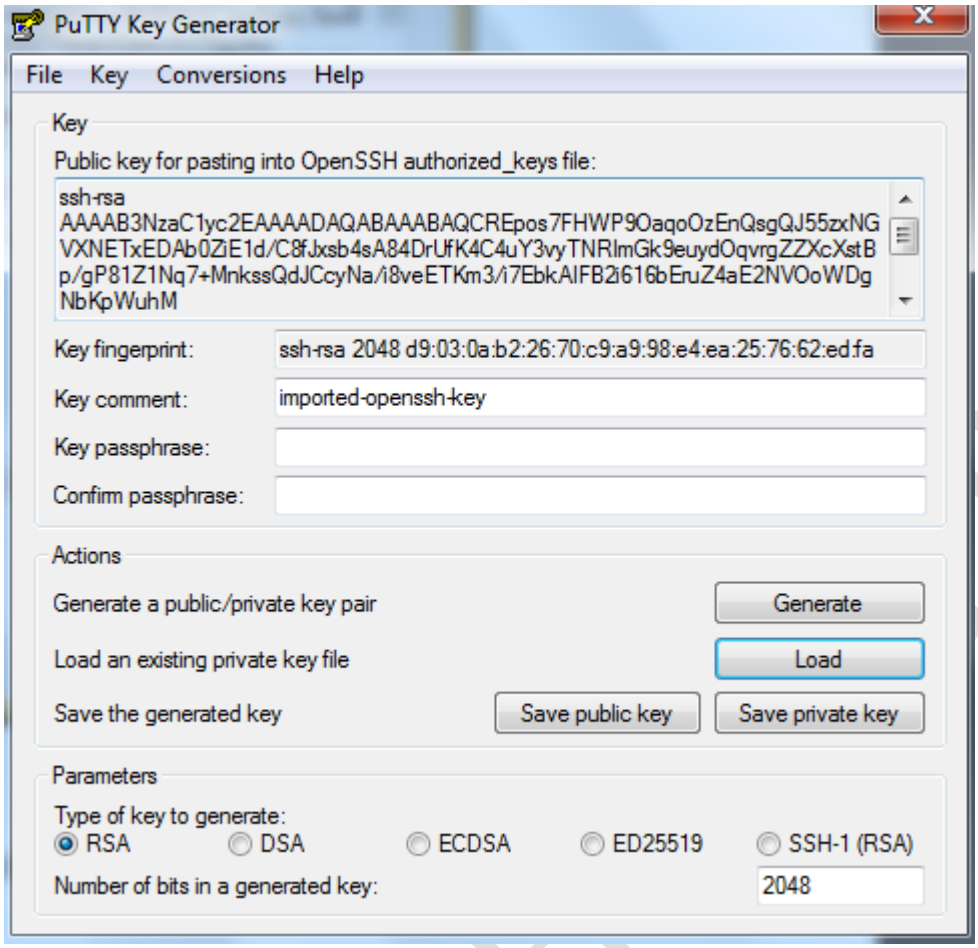


- The dashboard provides all of the information about our instance. We can view instance's ID, instance type, IP information, AZ, Security Group, and a whole lot more info.
- We can also obtain instance's health information using the Status Checks tab and the Monitoring tab.
- We can perform power operations on your instance such as start, stop, reboot, and terminate using the Actions tab located in the preceding instance table.

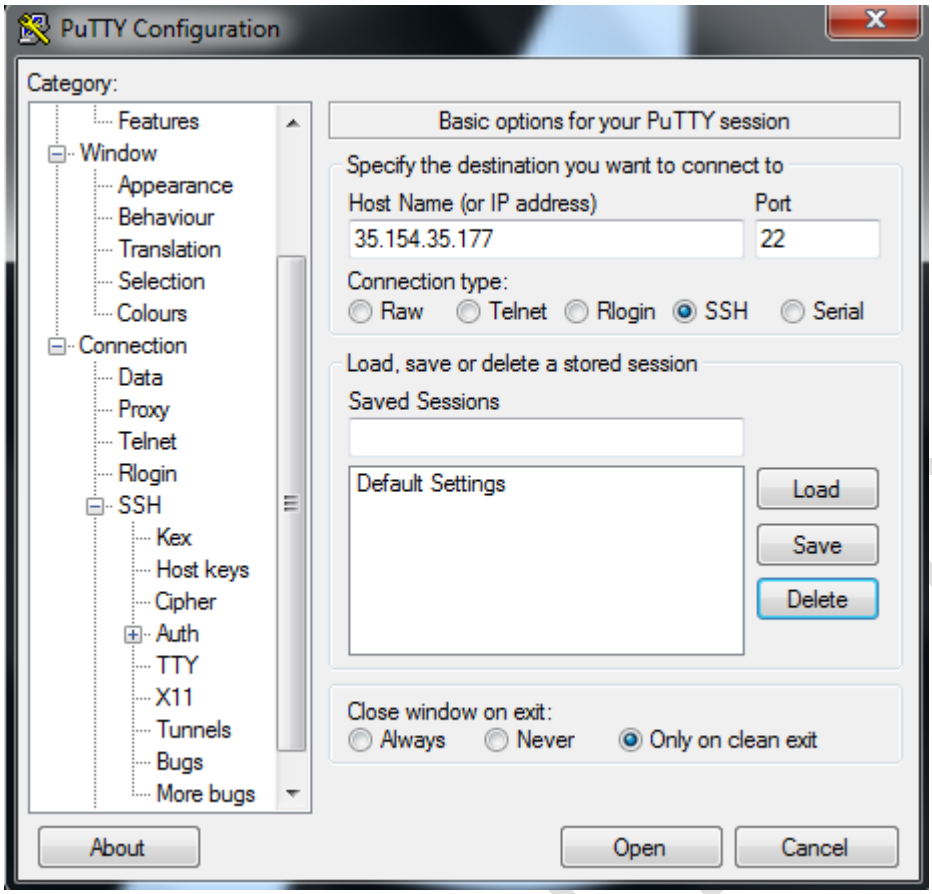
Connecting to Instance:

Once the instance is launched we have multiple options to connect to the instance. Mostly we can use **PuTTY** to connect Linux machines and **Remote Desktop** Feature for Windows Machine.
As we launched Linux machine, here we are going to see PuTTY option now.
PuTTY is basically an SSH and telnet client that can be used to connect to remote Linux instances. But before you get working on Putty, we need a tool called **PuttyGen** to convert the PEM file to PPK (Putty Private Key).
We can download the Putty.exe and PuttyGen.exe from the below URL:
https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html

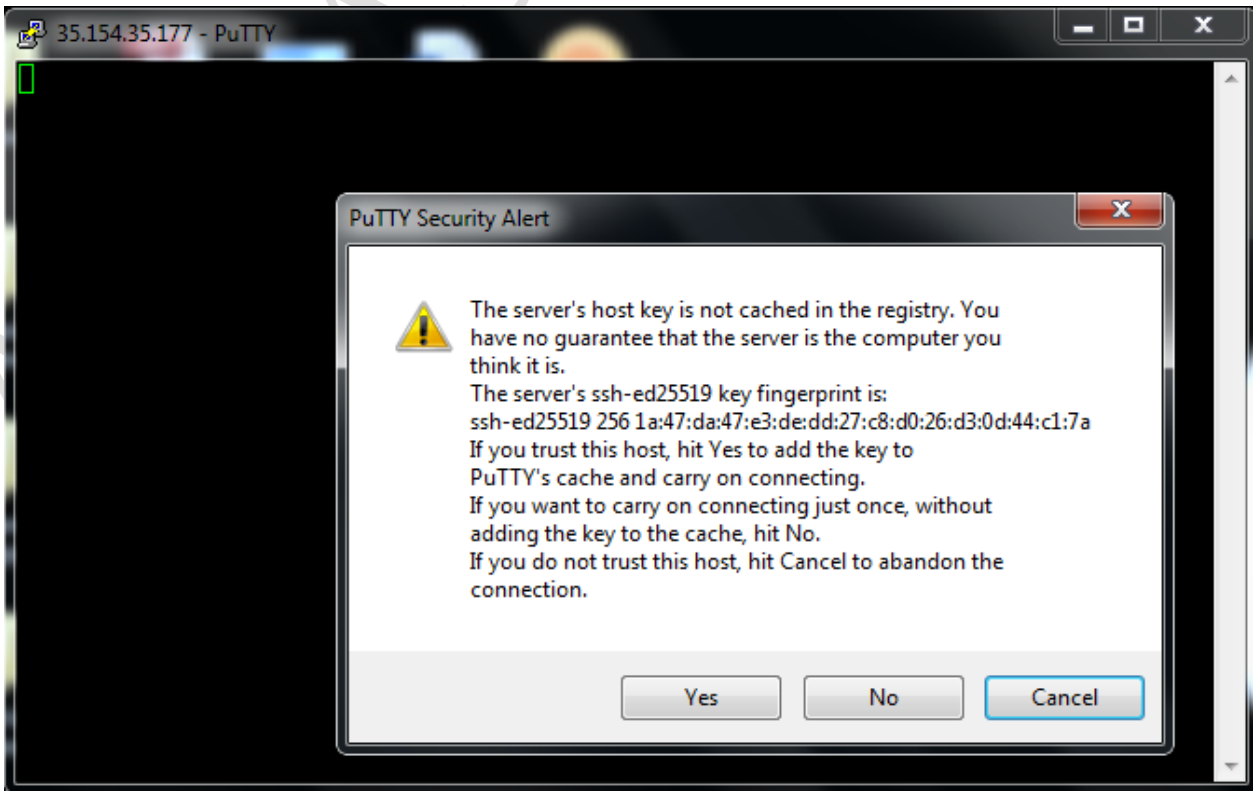
1. Download and install the latest copy of Putty and PuttyGen on local computer.
2. Launch PuttyGen and select the Load button and browse the downloaded Pem file (Which is created at the time of Instance launch).



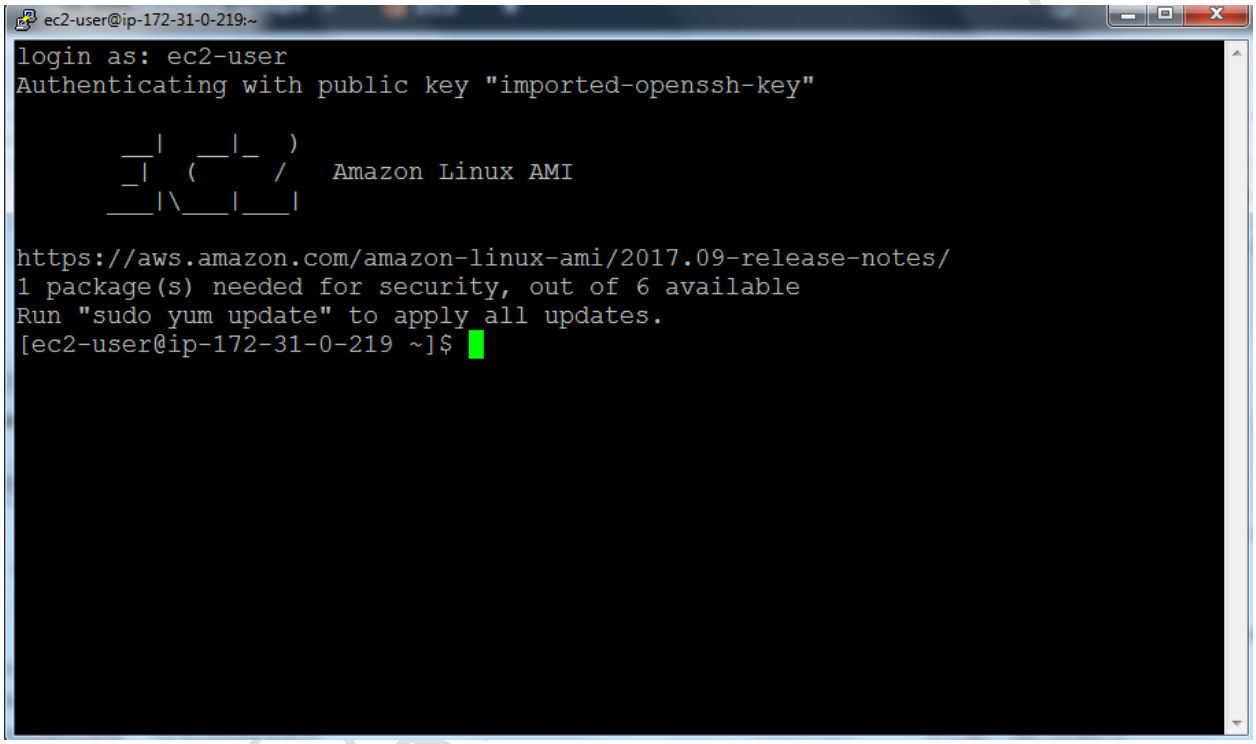
3. Once pem file is loaded, Select **"Save private key"** option.
    a. PuttyGen will prompt you with a warning message that you are saving this key without a passphrase and would you like to continue, Select **YES.**
4. Provide a name and save the new file (*.PPK) at a secure location. You can use this PPK file to connect to your instance using Putty
5. Please note down the **public IP address/ public DNS** of your instance.
6. Now open the **Putty** and enter the public IP in Host Name field and make sure to enter Port **22**



7. In Putty, under **Category pane**, expand the **SSH** option and then select **Auth**, then browse and upload the recently saved PPK file in the **Private key file for authentication** field. Once uploaded, click on Open to establish a connection to instance.
8. Give yes for on the Putty Security Alert.

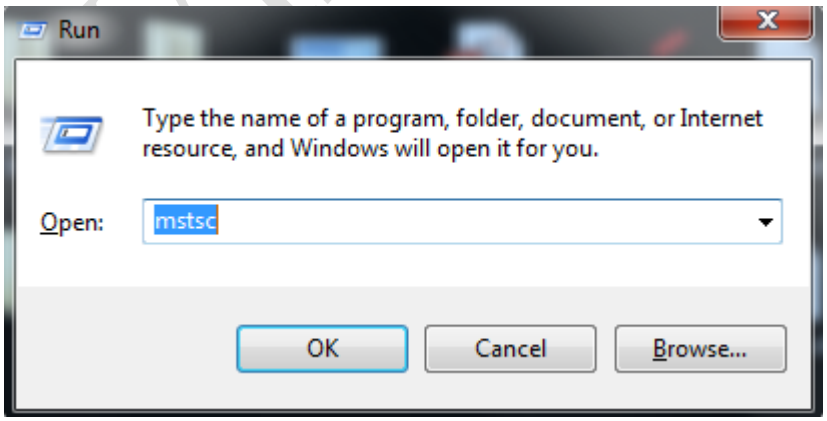


9. In the Putty terminal window, provide the user name for your Amazon Linux instance (ec2-user) and hit the *Enter* key. Now we have connected to our first instance and it is ready for use
10. Each Linux instance type launches with a default Linux system user account. For Amazon Linux, the user name is ec2-user. For RHEL, the user name is **ec2-user** or **root**. For Ubuntu, the user name is **ubuntu** or **root**. For Centos, the user name is **centos**. For Fedora, the user name is **ec2-user**. For SUSE, the user name is **ec2-user** or **root**. Otherwise, if **ec2-user** and **root** don't work, check with your AMI provider.
11

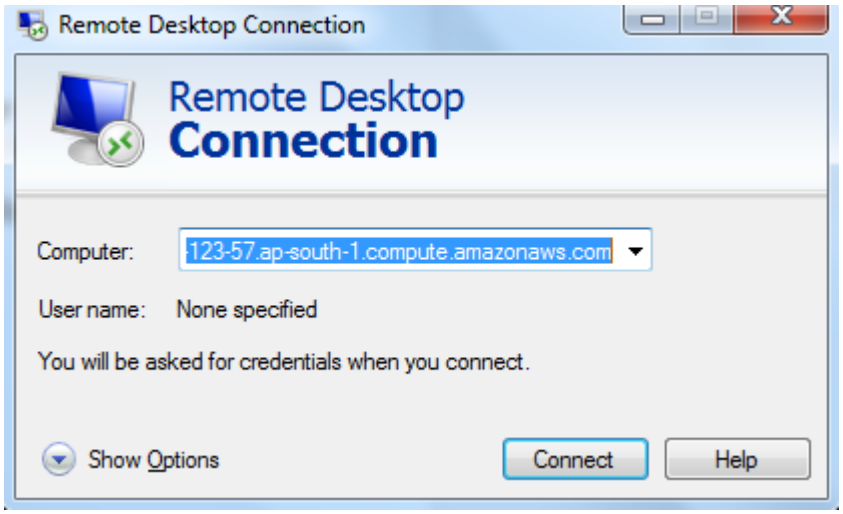


For RHEL-based AMIs (Redhat), the user name is either **root or the ec2-user**, and for Ubuntu-based AMIs, the user name is generally **Ubuntu** itself.

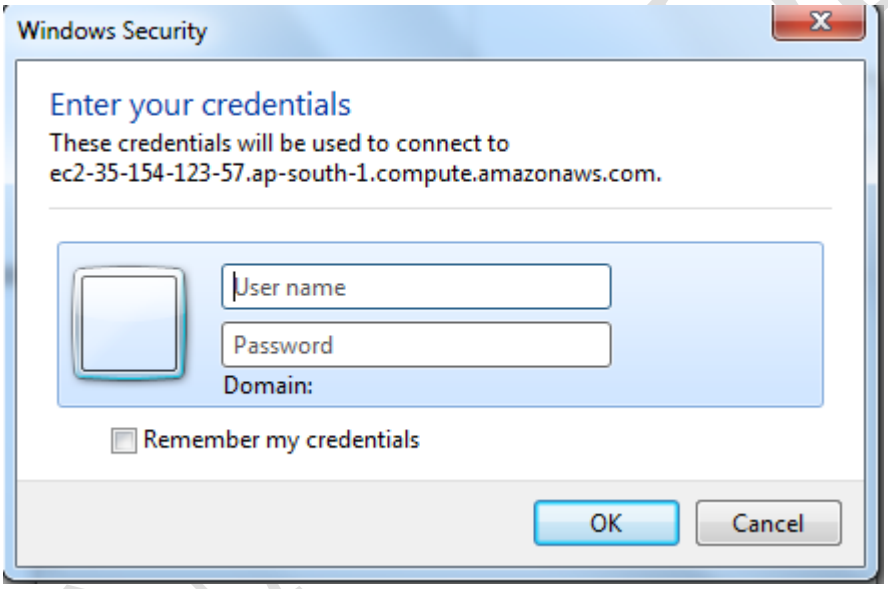
12. **To connect to Windows Instanc**e we have to use Remote Desktop Connection application.
13. Open Run and enter **mstsc**and press enter



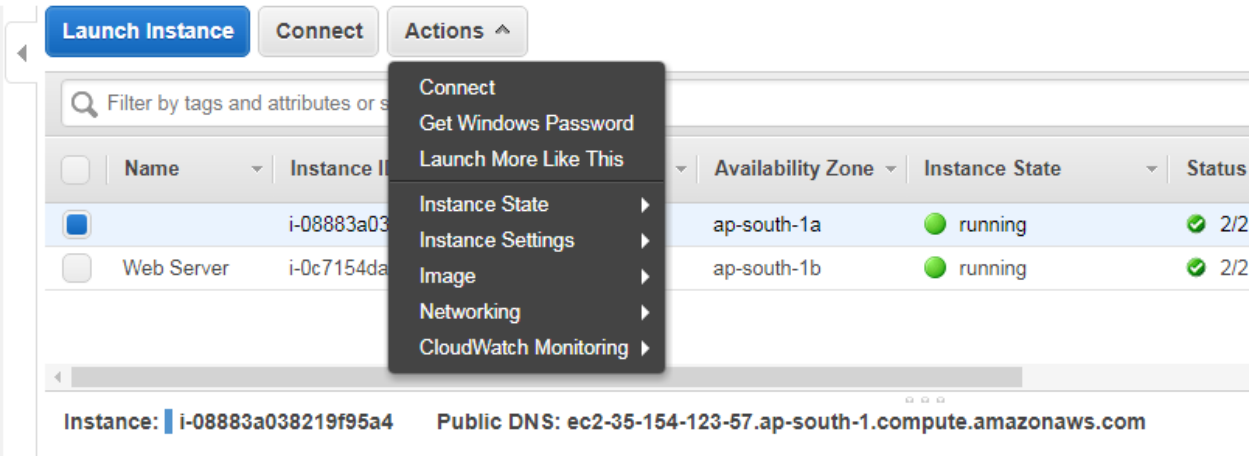
14. Note the public DNS/IP of the windows instance and enter it computer field and click on Connect.



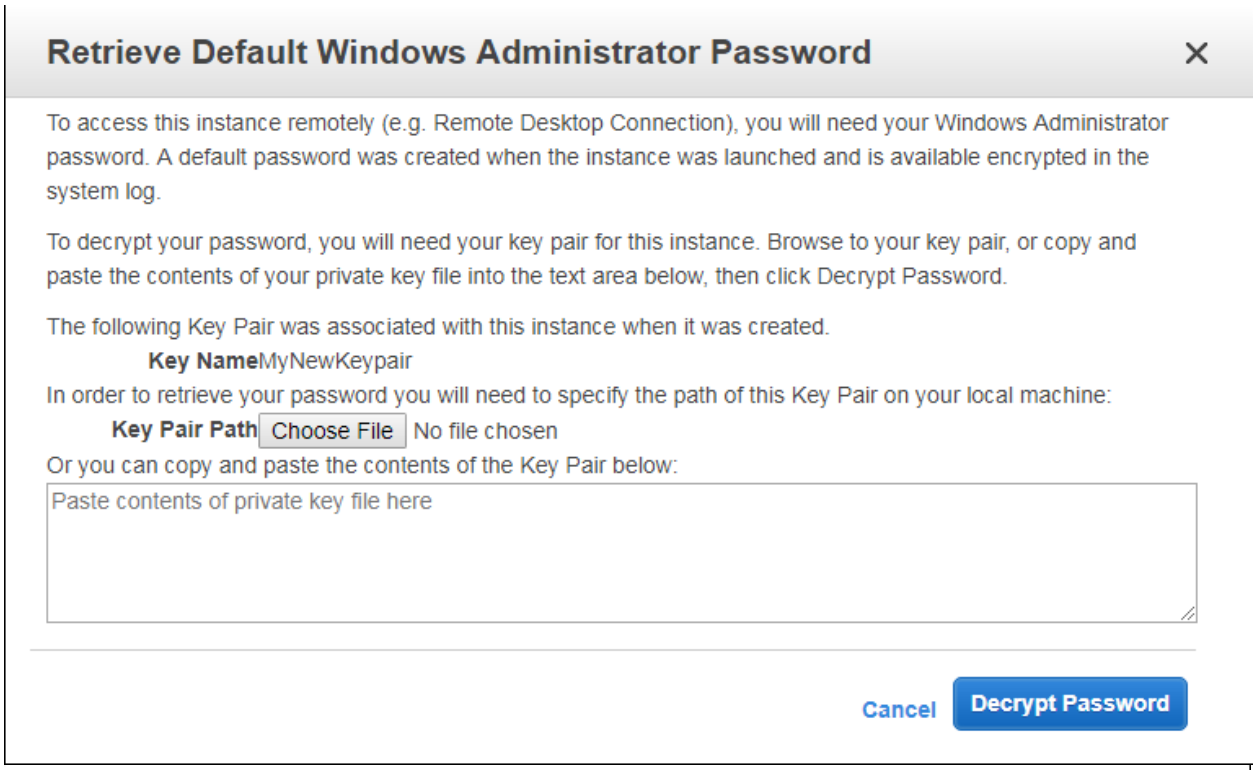
15 Now, It will ask you to enter the username and password to login to the instance.



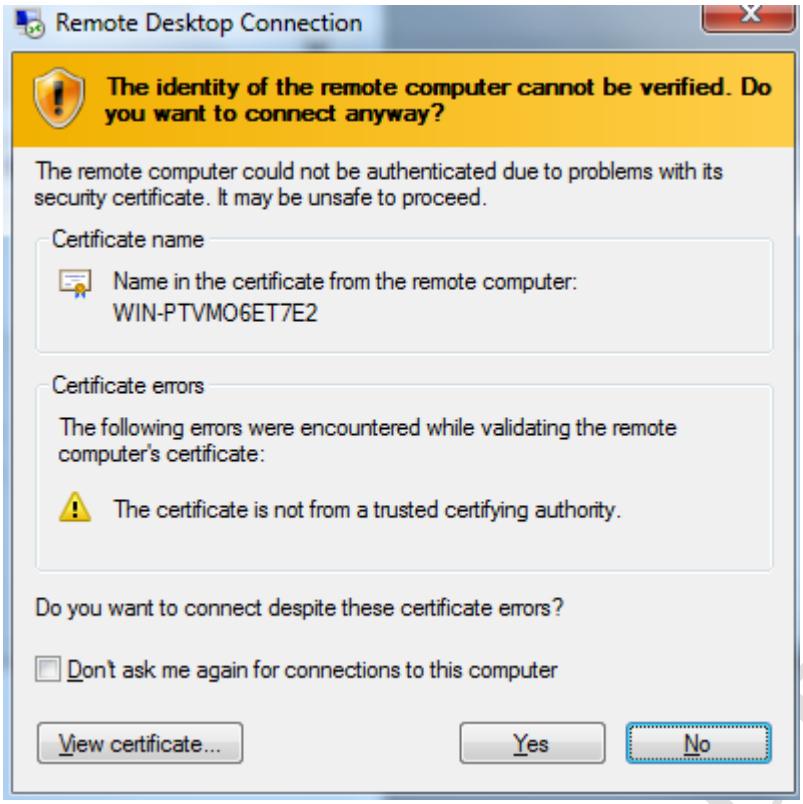
16 To get the Username and password to login to the instance we have get it from EC2 console.



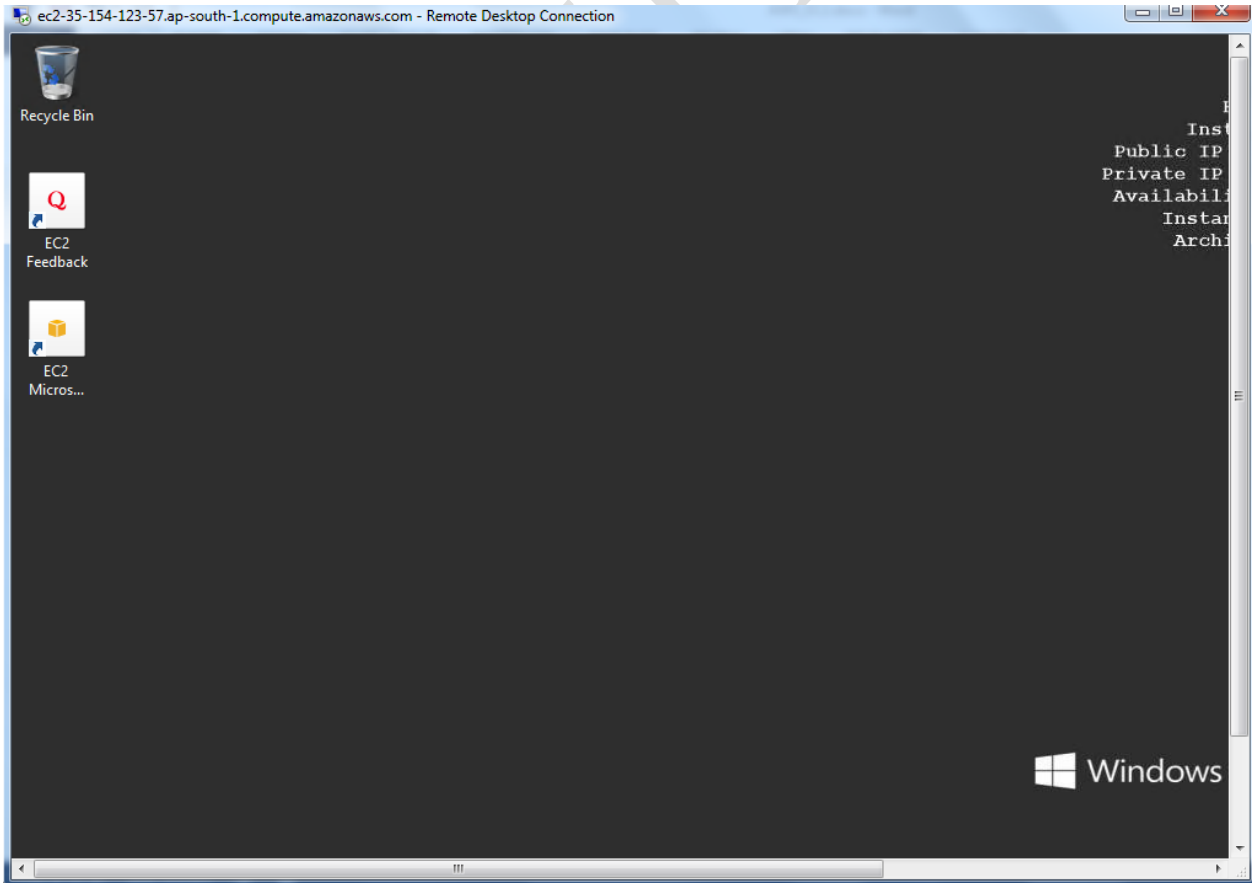
17 Select the instance which you want to get the UN & PWD. Go to Actions and select the “Get Windows Password”, then browse the PEM file and select “Decrypt Password” button.



18 Then you'll get the UN and Password, you can enter this UN &Pwd and click on connect, You'll asked for Certificate error prompt, simply click on **Yes** to connect to this machine.



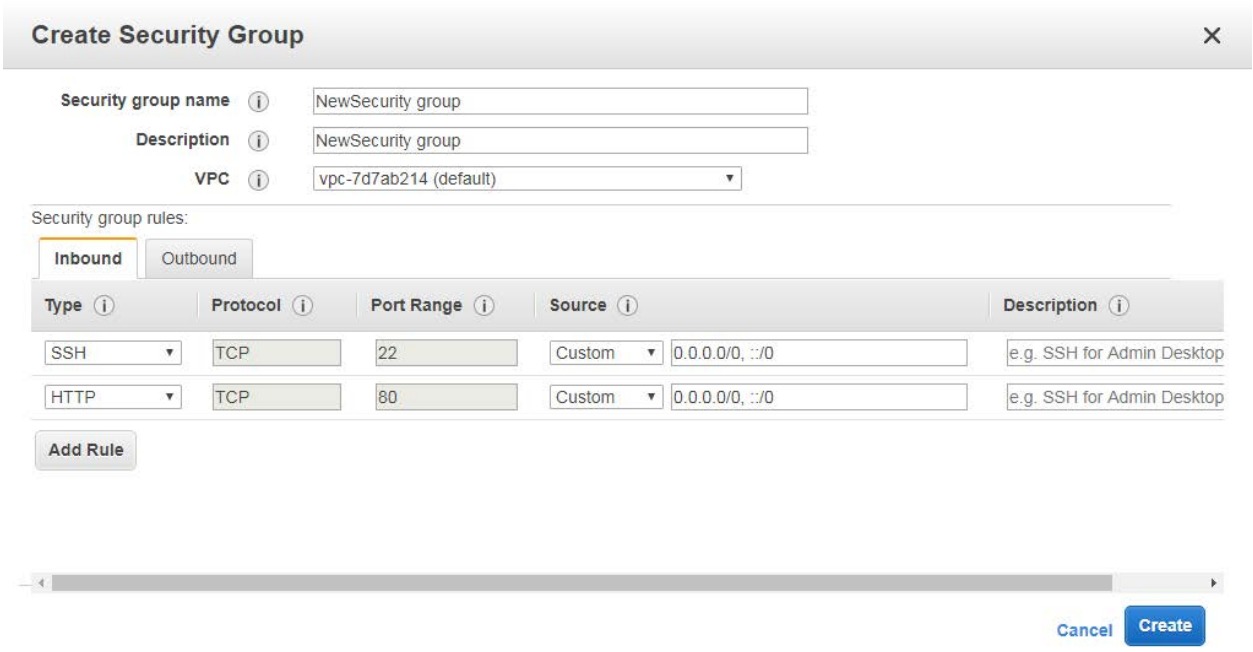
19 Now we have successfully connected to Windows Instance



Security Groups

Security groups allow you to control traffic based on port, protocol, and source/destination.
You can use Security Groups to restrict and filter out both the inbound and outbound traffic of an instance using a set of firewall rules. Each rule can allow traffic based on a particular protocol—TCP or UDP, based on a particular port—such as 22 for SSH, or even based on individual source and destination IP addresses. This provides lot of control and flexibility in terms of designing a secure environment for instances to run from.

- Security groups are associated with instances when they are launched. Every instance must have at least one security group but can have more.
- A security group is **default deny**; that is, it does not allow any traffic that is not explicitly allowed by a security group rule.
- A security group is a **stateful firewall,** If you open some port in inbound, it'll automatically allowed for outbound also.
- Security groups are applied at the instance level.
- Changes to Security Groups take effect immediately
- We cannot block specific IP address using security groups.
- We can specify allow rules, but not deny rules.
- We can modify the firewall rules of Security Groups any time, even when your instance is running.



You can select the Protocol Type inn Type field, automatically it'll show the protocol type and Port Range, and then we have to select the source.
Source field where you can basically specify any of these three options:
Anywhere: Using this option as the source, particular application port will be accessible from any and all networks out there (0.0.0.0/0). This is not a recommended configuration by AWS.
My IP: AWS will autofill the IP address of your local computer/Network here. If you select My IP option then the service works only in that particular network only.

Custom IP: This is the most preferable option, the Custom IP option allows you to specify your own custom source IP address or IP range as per our requirements. Ex: allow the particular application to access only via traffic coming from the network 202.153.31.0/24 CIDR.

VOLUMES AND SNAPSHOTS

An Amazon EBS volume is a durable, block-level storage device that you can attach to a single EC2 instance.

Amazon EBS provides persistent block-level storage volumes for use with Amazon EC2 instances. Each Amazon EBS volume is automatically replicated within its Availability Zone to protect you from component failure, offering high availability and durability.

Multiple Amazon EBS volumes can be attached to a single Amazon EC2 instance, although a volume can only be attached to a single instance at a time.

Types of Amazon EBS Volumes

Amazon EBS provides the following volume types:

- General Purpose SSD (gp2),
- Provisioned IOPS SSD (io1),
- Throughput Optimized HDD (st1),
- Cold HDD (sc1), and
- Magnetic (standard, a previous-generation type).

SSD-backed volumes optimized for transactional workloads involving frequent read/write operations with small I/O size, where the dominant performance attribute is IOPS

HDD-backed volumes optimized for large streaming workloads where throughput (measured in MiB/s) is a better performance measure than IOPS.

General Purpose SSD (gp2):

General Purpose SSD (gp2) volumes offer cost-effective storage that is ideal for a broad range of workloads. These volumes deliver single-digit millisecond latencies and the ability to burst to 3,000 IOPS for extended periods of time. Between a minimum of 100 IOPS (at 33.33 GiB and below) and a maximum of 10,000 IOPS (at 3,334 GiB and above), baseline performance scales linearly at 3 IOPS per GiB of volume size. AWS designs gp2 volumes to deliver the provisioned performance 99% of the time.

A gp2 volume can range in size from 1 GiB to 16 TiB.

Provisioned IOPS SSD (io1):

Provisioned IOPS SSD (io1) volumes are designed to meet the needs of I/O-intensive workloads, particularly database workloads, that are sensitive to storage performance and consistency.

- An io1 volume can range in size from 4 GiB to 16 TiB and you can provision up to 32,000 IOPS per volume.

Throughput Optimized HDD (st1):

Throughput Optimized HDD (st1) volumes provide low-cost magnetic storage that defines performance in terms of throughput rather than IOPS. This volume type is a good fit for large, sequential workloads such as Amazon EMR, ETL, data warehouses, and log processing.

- Not supported to use with root volume (Not Bootable)
- volume sizes ranging from 500 GiB to 16 TiB

- We will get Throughputs and Baseline is 40 MB/s per TiB

Cold HDD (sc1) Volumes

Cold HDD (sc1) volumes provide low-cost magnetic storage that defines performance in terms of throughput rather than IOPS. With a lower throughput limit than st1, sc1 is a good fit ideal for large, sequential cold-data workloads. If you require infrequent access to your data and are looking to save costs, sc1 provides inexpensive block storage.

- Not supported to use with root volume (Not Bootable)
- volume sizes ranging from 500 GiB to 16 TiB
- We will get Throughputs and Baseline is 12 MB/s per TiB

Magnetic volumes:

Magnetic volumes are backed by magnetic drives and are suited for workloads where data is accessed infrequently, and scenarios where low-cost storage for small volume sizes is important. These volumes deliver approximately 100 IOPS on average, with burst capability of up to hundreds of IOPS.

- Volume sizes ranging from 1 GiB to 1 TiB.

Throughput is the maximum rate of production or the maximum rate at which something can be processed.

Network throughput is the rate of successful message delivery over a communication channel.

Instance Store Volume

An instance store provides temporary block-level storage for your instance. This storage is located on disks that are physically attached to the host computer. Instance store is ideal for temporary storage of information that changes frequently, such as buffers, caches, scratch data, and other temporary content

Instance Store Lifetime

- The underlying disk drive fails
- The instance stops
- The instance terminates

Instance Store Volumes are called as Ephemeral Storage.

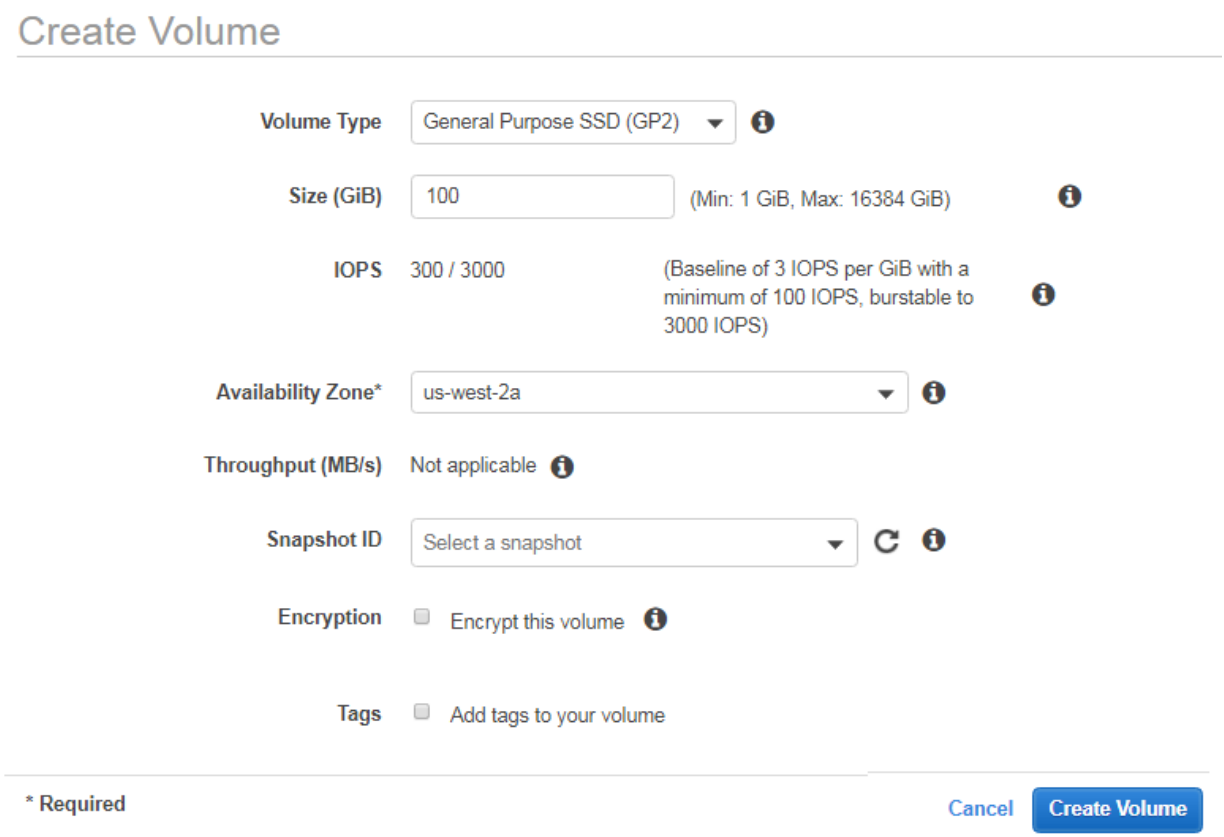
Instance store volumes cannot be stopped. If the underlying host fails, you will lose your data.

EBS backed instances can be stopped. You will not lose the data on this instance if it is stopped.

By default, both ROOT volumes will be deleted on termination, however with EBS volumes, you can keep the root device volume by Unchecking the “Delete on Termination” option.

Create a Volume:

From the Volume Management dashboard, select the Create Volume option.



Type: From the Type drop-down list, select either General Purpose (SSD), Provisioned IOPS (SSD), or Magnetic as per the requirements.

Size (GiB): Provide the size of your volume in GB.

IOPS: This field will only be editable if you have selected Provisioned IOPS (SSD) as the volume's type. Enter the max IOPS value as per your requirements.

Availability Zone: Select the appropriate availability zone in which you wish to create the volume.

Snapshot ID: This is an optional field. We can choose to populate your EBS volume based on a third party's snapshot ID.

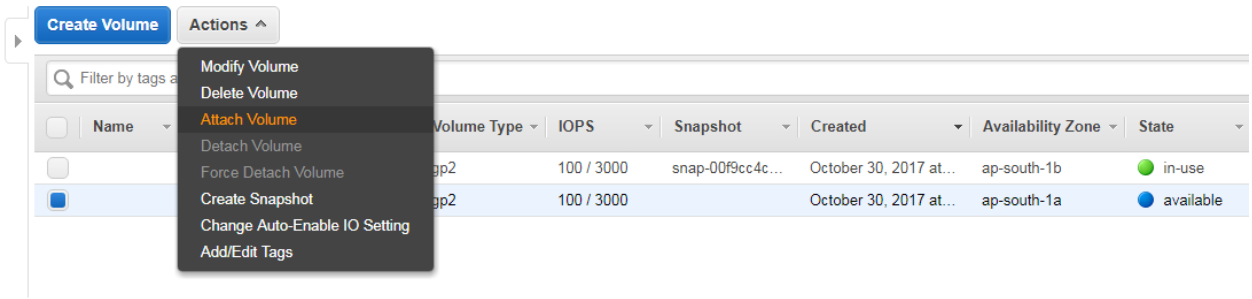
Encryption: We can choose whether or not to encrypt EBS Volume. Select Encrypt this volume checkbox if you wish to do so.

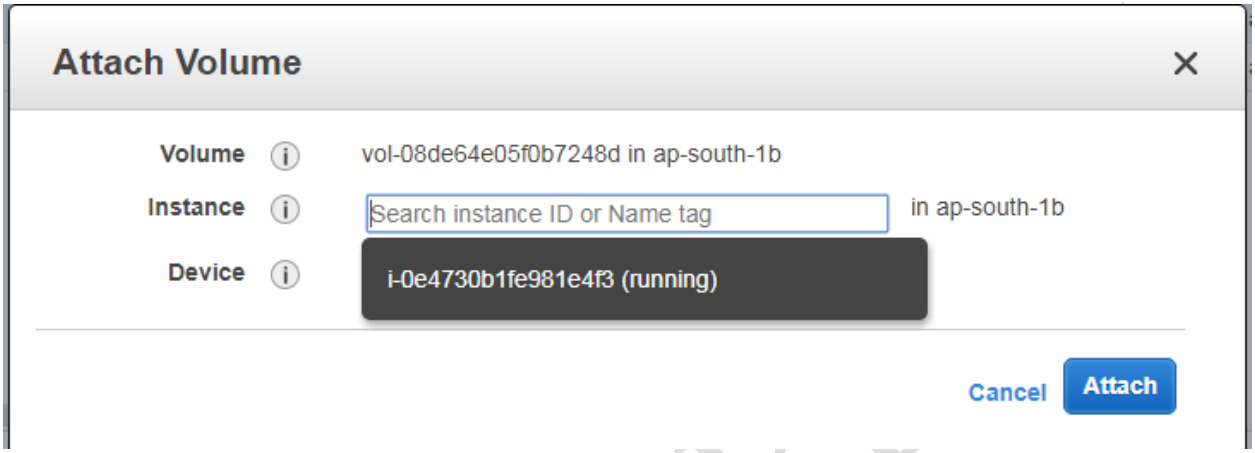
Master Key: On selecting the Encryption option, AWS will automatically create a default key pair for the AWS's KMS.

Once configuration settings are filled in, select Create to complete the volume's creation process. The new volume will take a few minutes to be available for use. Once the volume is created, we can now attach this volume to running instance.

Attaching EBS Volumes: Once the EBS volume is created, make sure it is in the available state before you go ahead and attach it to an instance. You can attach multiple volumes to a single instance at a time.

To attach a volume, select the **volume** from the Volume Management dashboard. Then select the **Actions** tab and click on the **Attach Volume** option.





When you select instance field, automatically you'll get thee running instances list from that particular availability zone. Select the Instance you want to attach this volume. Then click on **Attach**. Now the Volume state will change to **in-use** from Available.

We have to mount this volume from operating system level. For windows, you have to perform it though Disk Management option.

In Linux:

1. Elevate your privileges to root.
2. Type **df –h** command to check the current disk partitioning of instance.
3. Give **fdisk –l** command to verify the newly added disk.

```
[root@ip-172-31-7-51 ~]# fdisk -l
WARNING: fdisk GPT support is currently new, and therefore in an experimental phase.
 Use at your own discretion.

Disk /dev/xvda: 8589 MB, 8589934592 bytes, 16777216 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk label type: gpt


#         Start          End    Size  Type            Name
 1         4096     16777182      8G  Linux filesyste Linux
128         2048         4095      1M  BIOS boot parti BIOS Boot Partition

Disk /dev/xvdf: 1073 MB, 1073741824 bytes, 2097152 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
```

4. We have to choose the file system type. Here am using ext4 file system. Then run the following command.

   Mkfs –t ext4 /dev/xvdf

```
[root@ip-172-31-7-51 ~]# mkfs -t ext4 /dev/xvdf
mke2fs 1.42.12 (29-Aug-2014)
Creating filesystem with 262144 4k blocks and 65536 inodes
Filesystem UUID: 380ed17c-022a-440e-a696-ccd0caa3bd78
Superblock backups stored on blocks:
        32768, 98304, 163840, 229376

Allocating group tables: done
Writing inode tables: done
Creating journal (8192 blocks): done
Writing superblocks and filesystem accounting information: done
```

5. Now volume is formatted, we can create a new directory on Linux instance and mount the volume to it using standard Linux commands:

   mkdir /newvolume

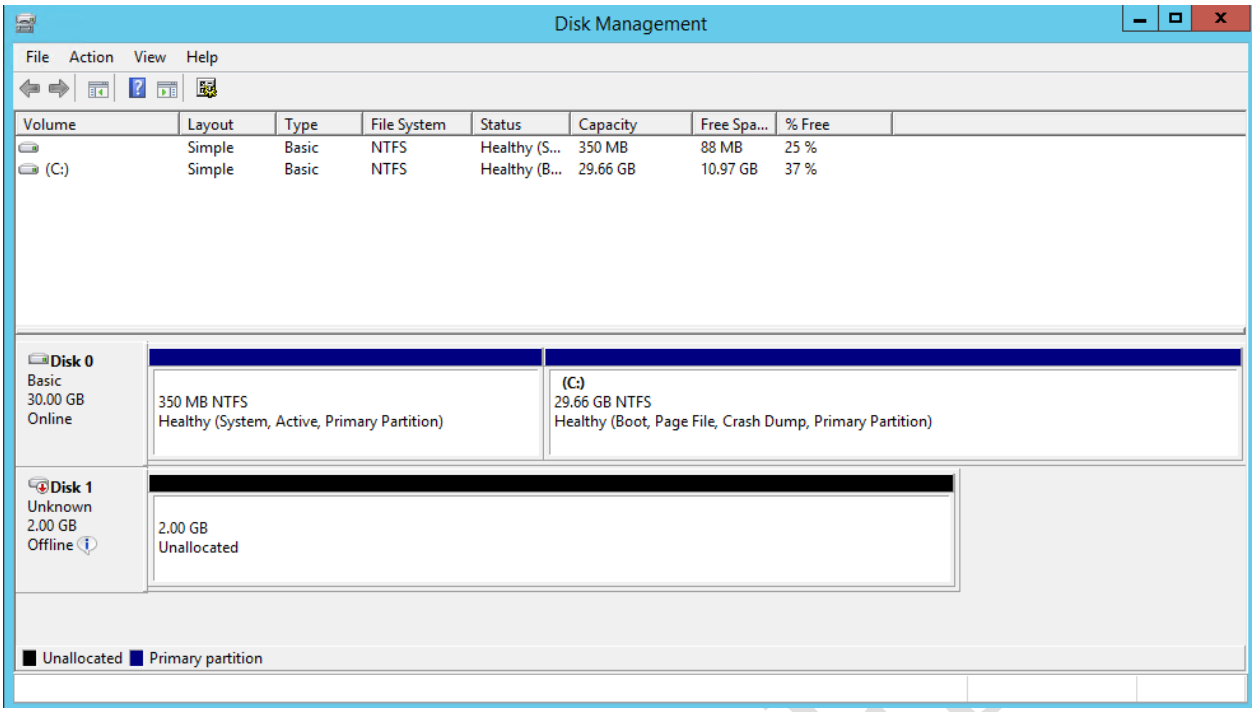
   mount /dev/xvdf /newvolume

```
[root@ip-172-31-7-51 ~]# mkdir /newvolume
[root@ip-172-31-7-51 ~]# mount /dev/xvdf /newvolume
[root@ip-172-31-7-51 ~]#
```

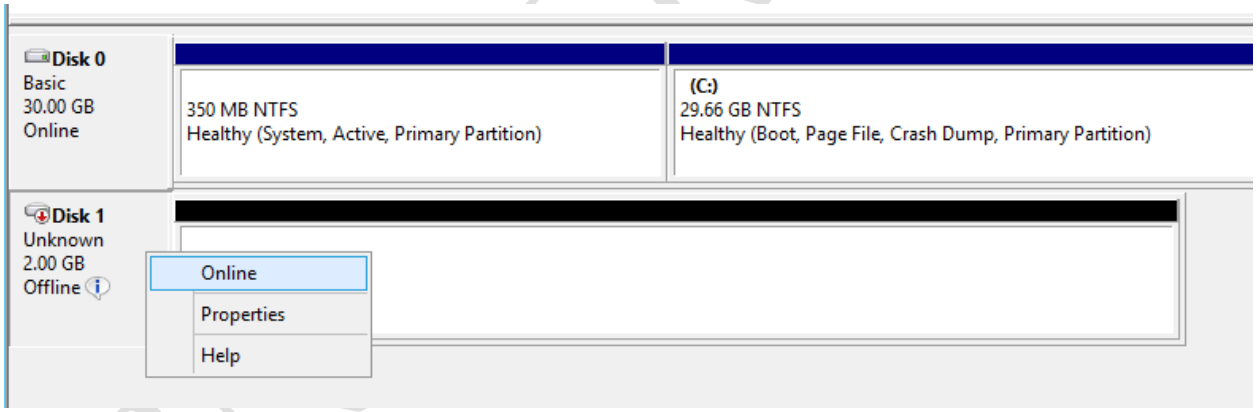
6. Now the volume is available for the use.

For Windows Instances:

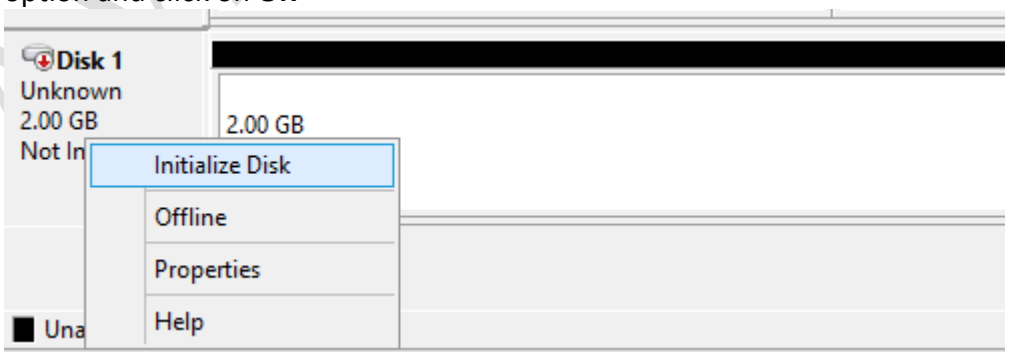
1. Attach the volume to the windows instance same as previous step.
2. Login to the windows instance and open Disk management console.
3. Open Run and give **diskmgmt.msc**command to open the Disk Management.

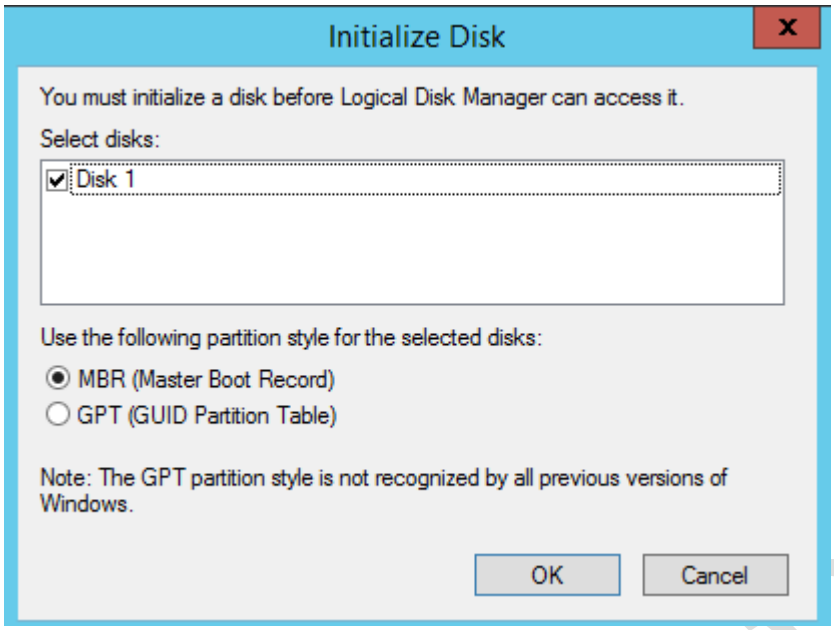


4. The newly created 2GB volume is attached to the Windows instance and by default the status of this drive will set to offline, Select the Disk 1, then choose **Online** option to make the volume online.

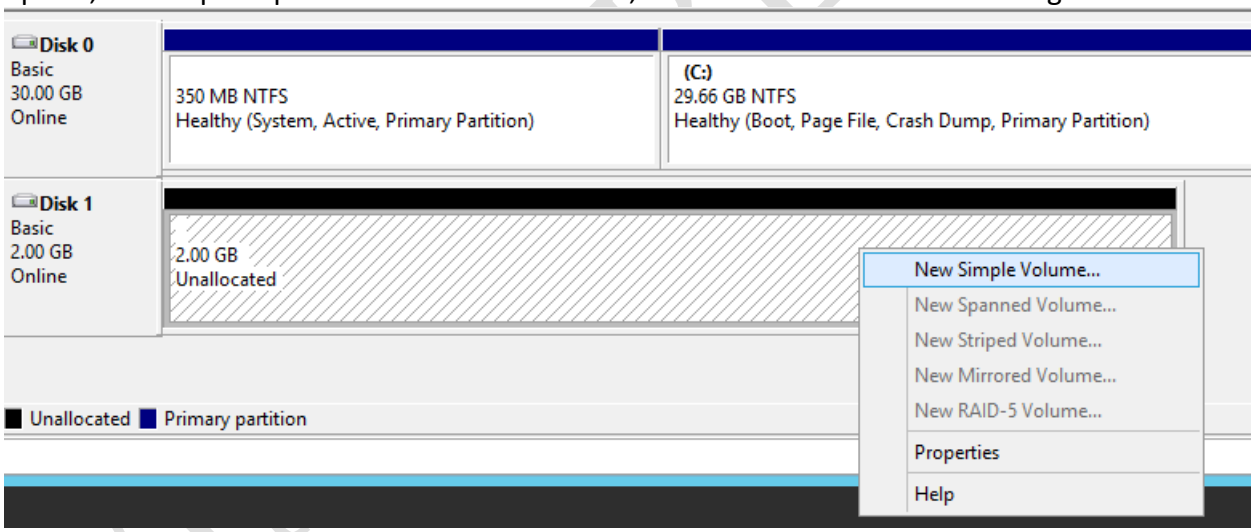


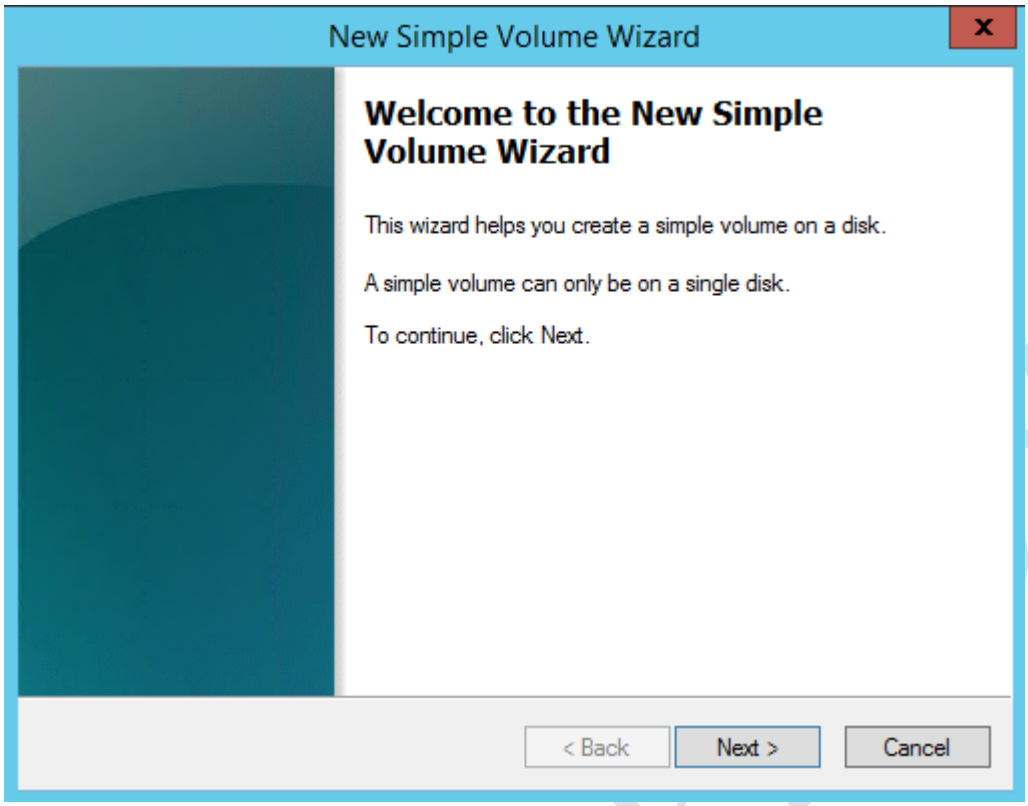
5. Here we have to initialize the Disk, Give right click on Disk then select the **initialize disk** option and click on **OK**

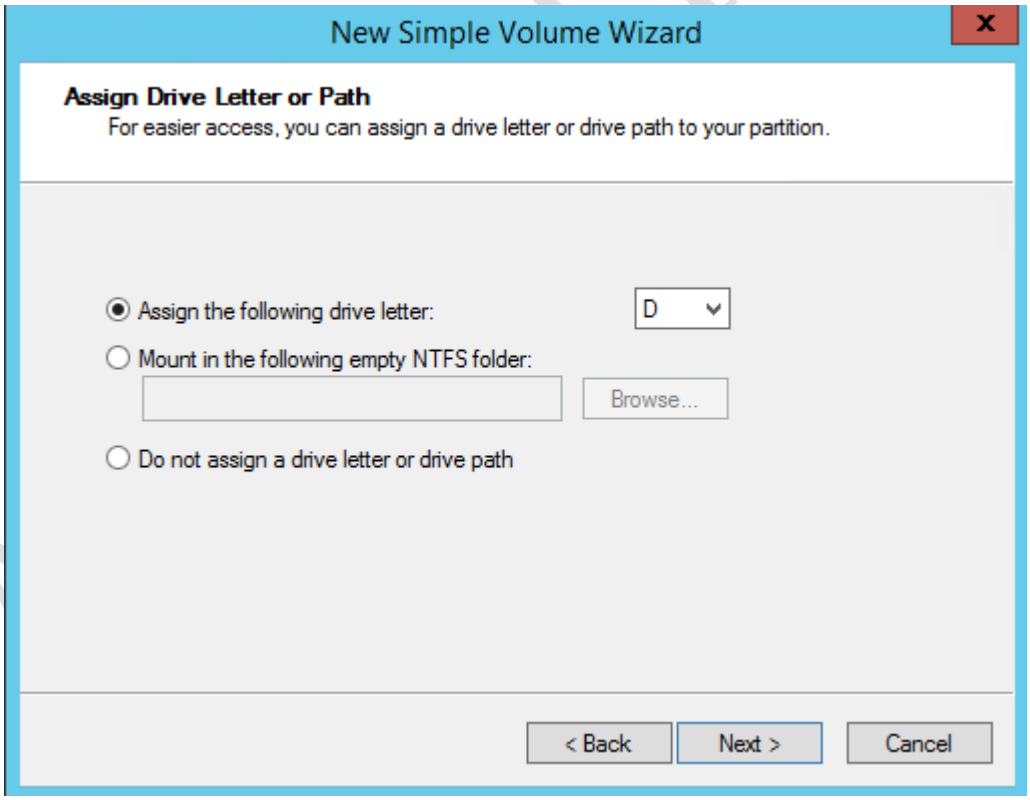


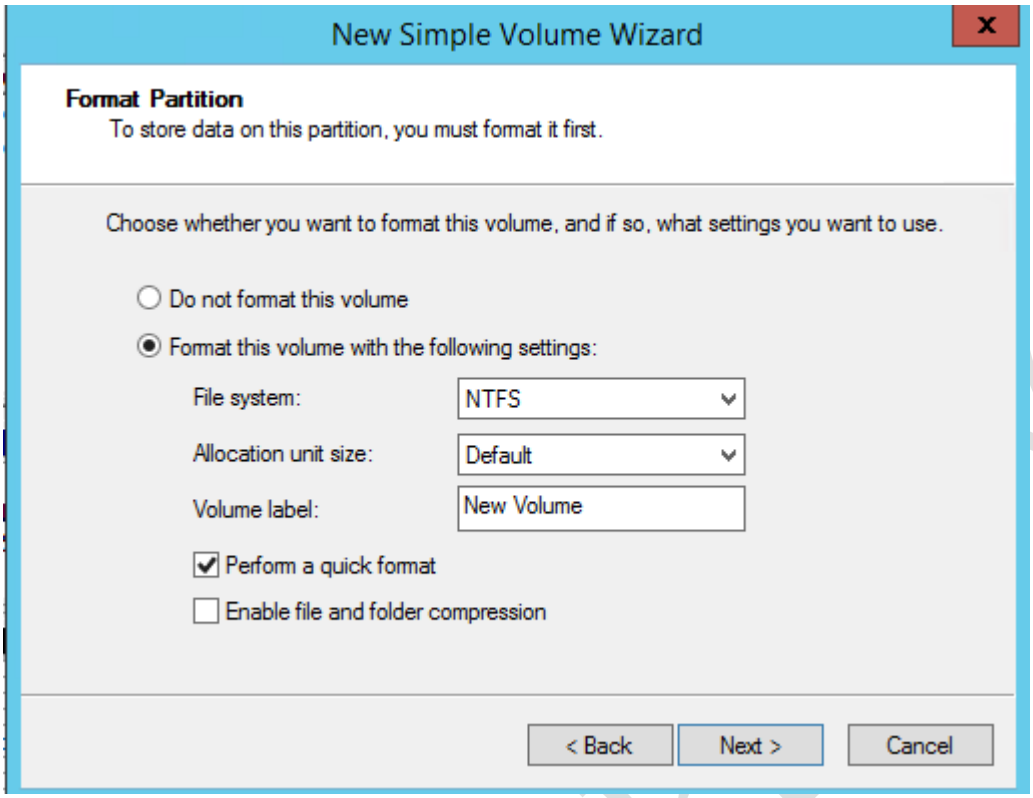


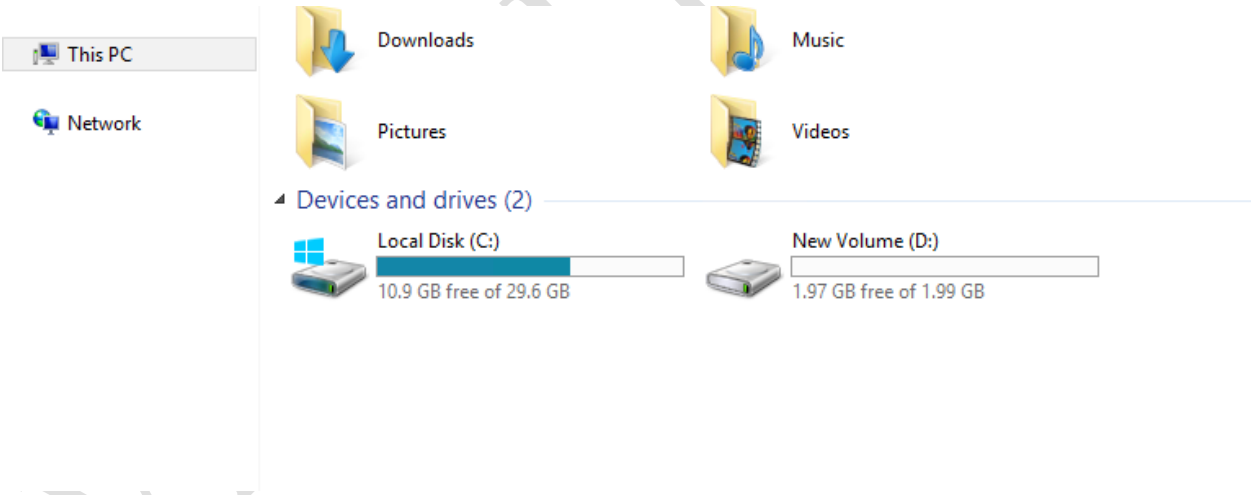
6. Now we have to create a volume, Give right click on dive select the “New Simple Volume” option, It will open up a Volume creation wizard, follow the wizard as below images




New Simple Volume Wizard
Welcome to the New Simple Volume Wizard
This wizard helps you create a simple volume on a disk.
A simple volume can only be on a single disk.
To continue, click Next.
< Back
Next >
Cancel


New Simple Volume Wizard
Assign Drive Letter or Path
For easier access, you can assign a drive letter or drive path to your partition.
Assign the following drive letter:
D
Mount in the following empty NTFS folder:
Browse...
Do not assign a drive letter or drive path
< Back
Next >
Cancel





7. Now we can see the newly created volume along with other volumes. You can use the Disk Management console to Shrink, extend or to delete the volumes.

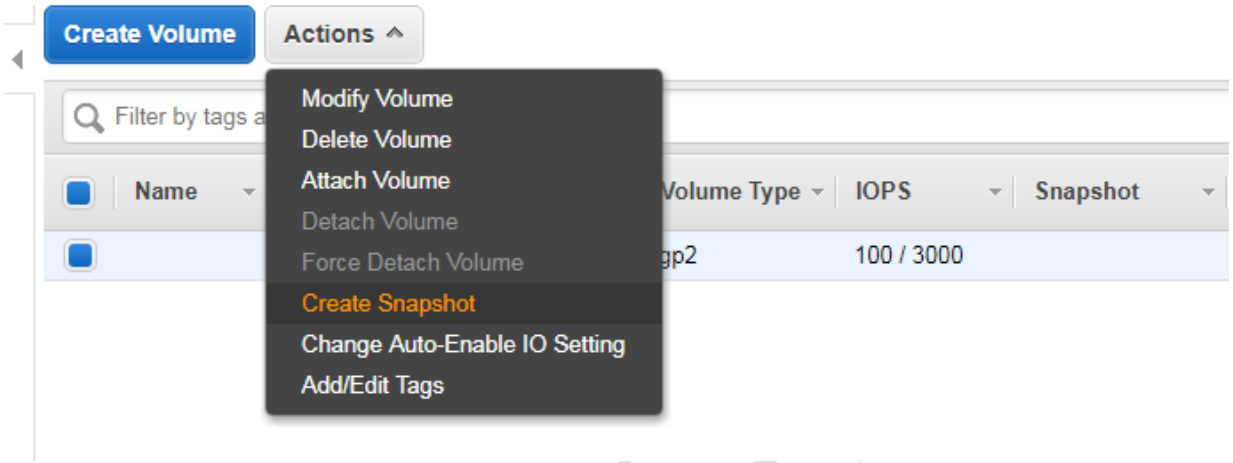
Backup of EBS volumes

We can back up the data on our Amazon EBS volumes, regardless of volume type, by taking point-in-time snapshots.

- Snapshots are incremental backups, which means that only the blocks on the device that have changed since your most recent snapshot are saved.
- Data for the snapshot is stored using Amazon S3 technology.
- While snapshots are stored using Amazon S3 technology, they are stored in AWS-controlled storage and not in your account's Amazon S3 buckets.
- Snapshots are constrained to the region in which they are created, meaning you can use them to create new volumes only in the same region.

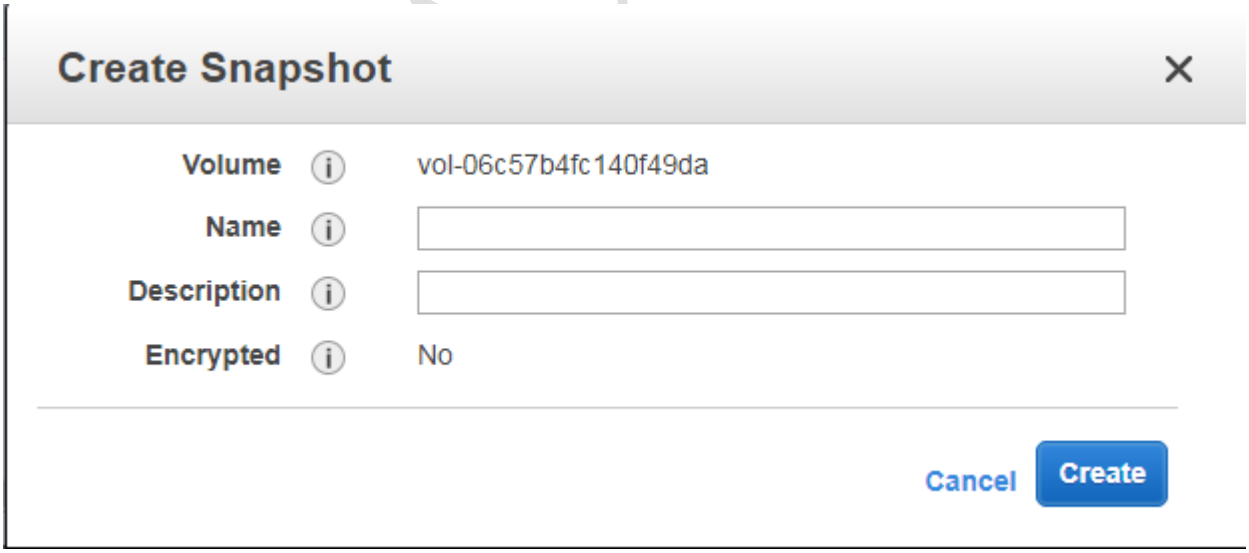
- If you need to restore a snapshot in a different region, you can copy a snapshot to another region.
- Snapshots can also be used to increase the size of an Amazon EBS volume.
    - To increase the size of an Amazon EBS volume, take a snapshot of the volume, then create a new volume of the desired size from the snapshot. Replace the original volume with the new volume.

To create a snapshot of volumes, select the particular volume from the Volume Management dashboard. Click on the **Actions** tab and select the **Create Snapshot** option.

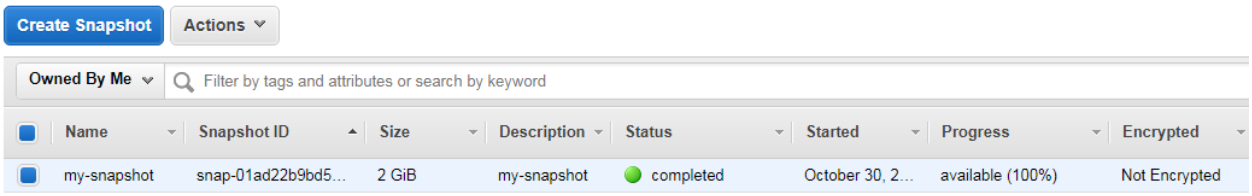


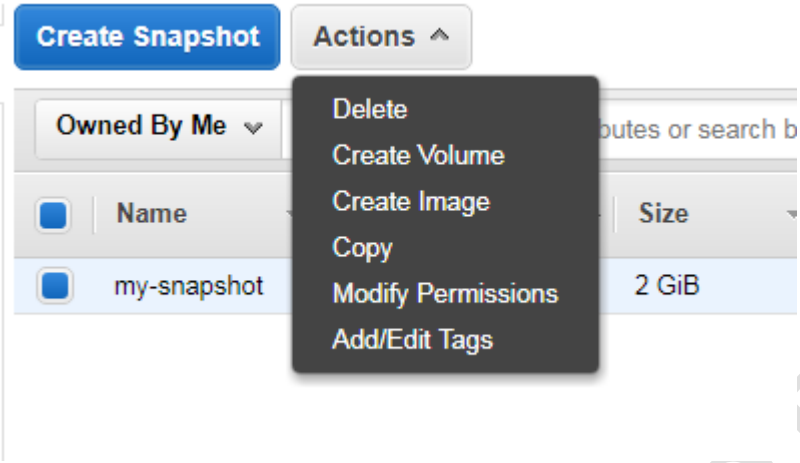
Give a Name and Description for the Snapshot.

- Snapshot of an Encrypted root volume is going to be an encrypted one.
- Volume creating from the encrypted snapshot also going to be an encrypted one.
- We can share the snapshots, but the snapshot must be an **unencrypted.**



We can go to Snapshot dashboard to verify the snapshot creation.





The above are the options available for snapshot.

Delete: we can delete the selected snapshot with this option.

Create Volume: We can create a new volume from this snapshot, while creating the new snapshot, we can change the volume type or increase the size if we want.

Create Image: We can create an AMI from this snapshot.

Copy: We can copy the snapshot from one region to another region.

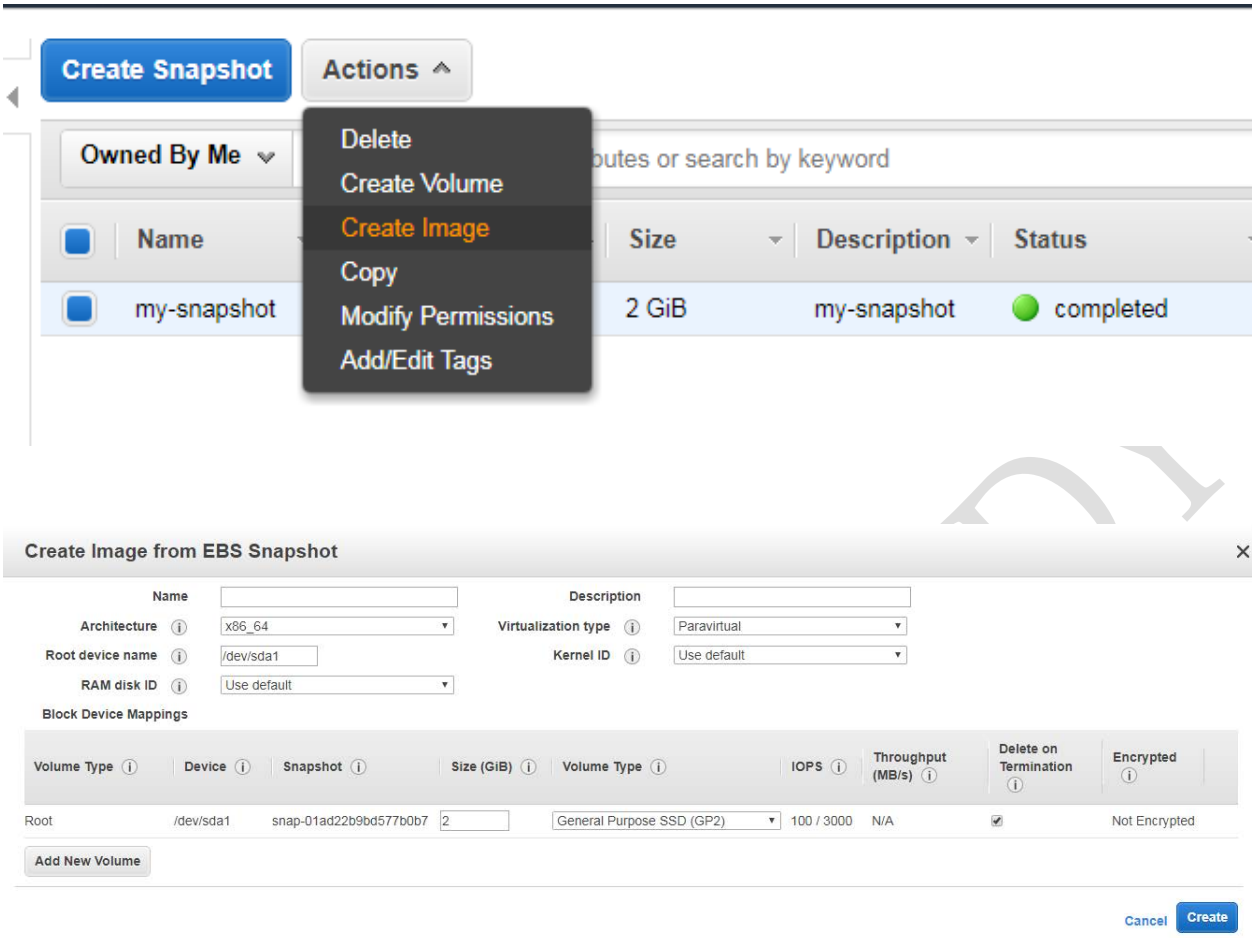
Modify Permissions: We can share the snapshots with specific AWS account user or made available to public, but this option will not enable if our snapshot is an encrypted.

Creating an AMI

An Amazon Machine Image (AMI) provides the information required to launch a virtual server in the cloud. You specify an AMI when you launch an instance, and you can launch as many instances from the AMI as you need. You can also launch instances from as many different AMIs as you need.

- A template for the root volume for the instance
- Launch permissions that control which AWS accounts can use the AMI to launch instances.

To create an AMI, Select the root volume's Snapshot, then select **Create Image** option.



Name: Provide a suitable and meaningful name for your AMI.

Description: Provide a suitable description for your new AMI.

Architecture: We can either choose between i386 (32 bit) or x86_64 (64 bit).

Root device name: Enter a suitable name for your root device volume.

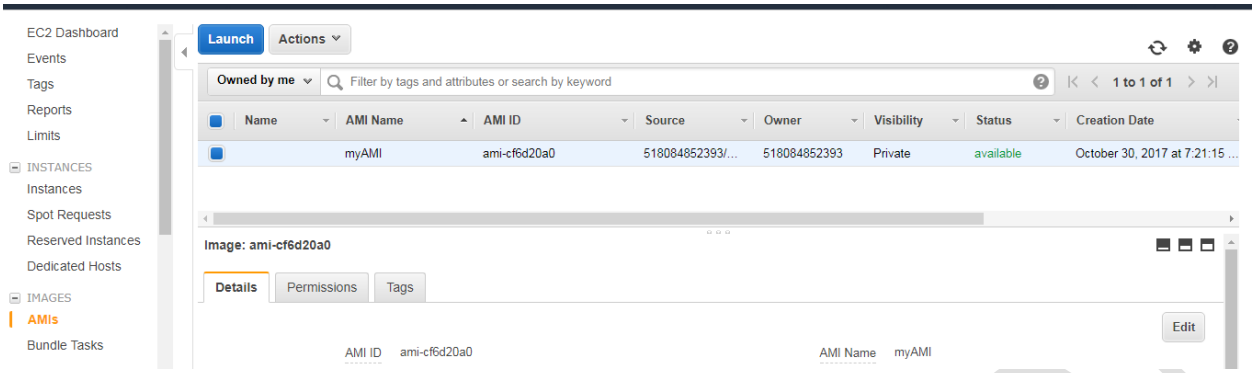
Virtualization type: We can choose whether the instances launched from this particular AMI will support Paravirtualization (PV) or Hardware Virtual Machine (HVM) virtualization.

- **Xen** is an hypervisor that runs on metal (the pc / server) and then hosts virtual machines called domains.
- **PV** domain is a paravirtualizeddomain, that means the operating system has been modified to run under Xen, and there's no need to actually emulate hardware. This should be the most efficient way to go, performance wise.
- **HVM** domain is hardware emulated domain, that means the operating system (could be Linux, Windows, whatever) has not been modified in any way and hardware gets emulated.

RAM disk ID, Kernel ID: We can select and provide your AMI with its own RAM disk ID (ARI) and Kernel ID (AKI); however, in this case I have opted to keep the default ones.

Block Device Mappings: We can use this dialog to either expand root volume's size or add additional volumes to it. We can change the Volume Type from General Purpose (SSD) to Provisioned IOPS (SSD) or Magnetic as per our AMI's requirements.

Click on **Create** to complete the AMI creation process. The new AMI will take a few minutes to spin up.



We can select the AMI and choose **Launch** option to launch a new instance. We will get the instance launch wizard.

- AMI are regional, if required we can copy AMI to another region with Copy option.
- We can share the AMI to any other AWS account users or we can make it public.
- Every AMI will associate with a Snapshot.
- AMI are registered with the AWS accounts, if you no longer required any AMI, you can select Deregister option under **Actions**.
- We cannot delete the Snapshot if it is associated with an AMI.

Elastic Load Balancing

The Elastic Load Balancing service allows you to distribute traffic across a group of Amazon EC2 instancesenabling you to achieve high availability in your applications.

Elastic Load Balancing supports routing and load balancing of Hypertext Transfer Protocol (HTTP), Hypertext Transfer Protocol Secure (HTTPS), Transmission Control Protocol (TCP), and Secure Sockets Layer (SSL) traffic to Amazon EC2 instances.

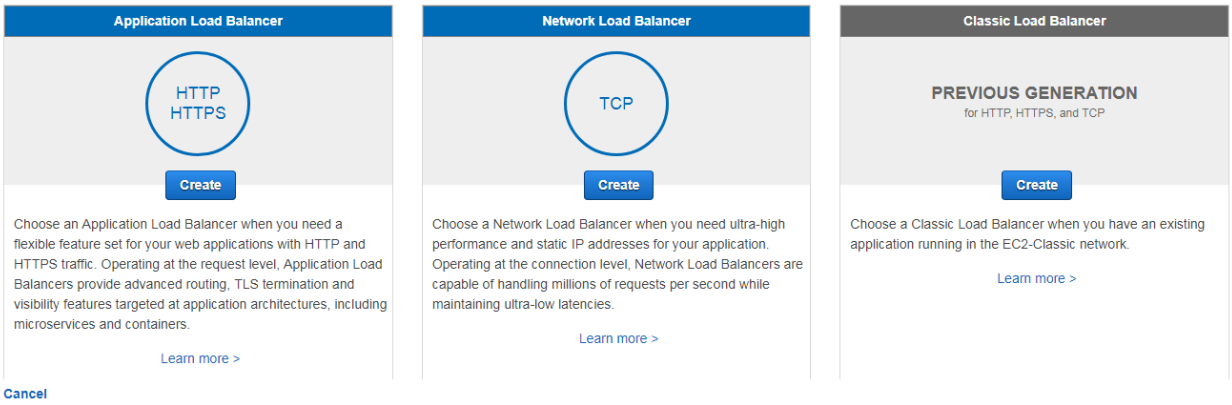
Elastic Load Balancing supports health checks for Amazon EC2 instances to ensure traffic is not routed to unhealthy or failing instances.

We will not get any public IP address for ELBs, We will get a DNS record for every LB.

Advantages of ELB

- Elastic Load Balancing is a managed service, it scales in and out automatically to meet the demands of increased application traffic and is highly available within a region itself as a service.
- ELB helps you achieve high availability for your applications by distributing traffic across healthy instances in multiple Availability Zones.
- ELB seamlessly integrates with the Auto Scaling service to automatically scale the Amazon EC2 instances behind the load balancer.
- ELB is secure, working with Amazon Virtual Private Cloud (Amazon VPC) to route traffic internally between application tiers, allowing you to expose only Internet-facing public IP addresses.
- ELB also supports integrated certificate management and SSL termination.

We have three types of load balancers available with AWS.



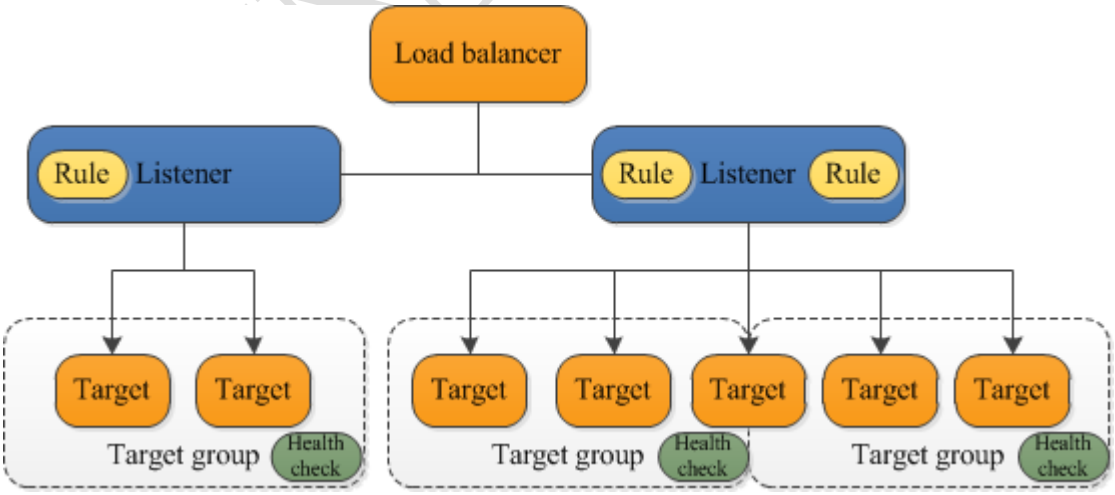
1. Classic Lead balancer
2. Application load Balancer
3. Network Load Balancer

Network Load Balancer:

A Network Load Balancer functions at the fourth layer of the Open Systems Interconnection (OSI) model. It can handle millions of requests per second. After the load balancer receives a connection request, it selects a target from the target group for the default rule. It attempts to open a TCP connection to the selected target on the port specified in the listener configuration.

Application Load Balancer:

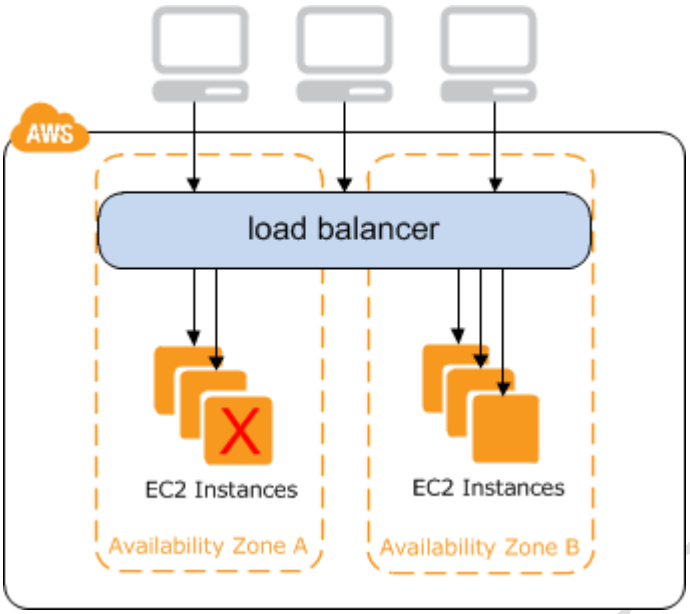
An Application Load Balancer functions at the application layer, the seventh layer of the Open Systems Interconnection (OSI) model. After the load balancer receives a request, it evaluates the listener rules in priority order to determine which rule to apply, and then selects a target from the target group for the rule action using the round robin routing algorithm. Note that you can configure listener rules to route requests to different target groups based on the content of the application traffic. Routing is performed independently for each target group, even when a target is registered with multiple target groups.



We can add and remove targets from load balancer as our needs change, without disrupting the overall flow of requests to our application.

Classic Load Balancer:

A Classic load balancer work with listener checks for connection requests from clients, using the protocol and port that we configure, and forwards requests to one or more registered instances using the protocol and port number that you configure. We can add one or more listeners to our load balancer.



Internet-Facing Load Balancers: An Internet-facing load balancer is a load balancer that takes requests from clients over the Internet and distributes them to Amazon EC2 instances that are registered with the load balancer.

Internal load balancers: Internal Load Balancers that connect and route traffic to private subnets. We can use internal load balancers to route traffic to your Amazon EC2 instances in VPCs with private subnets.

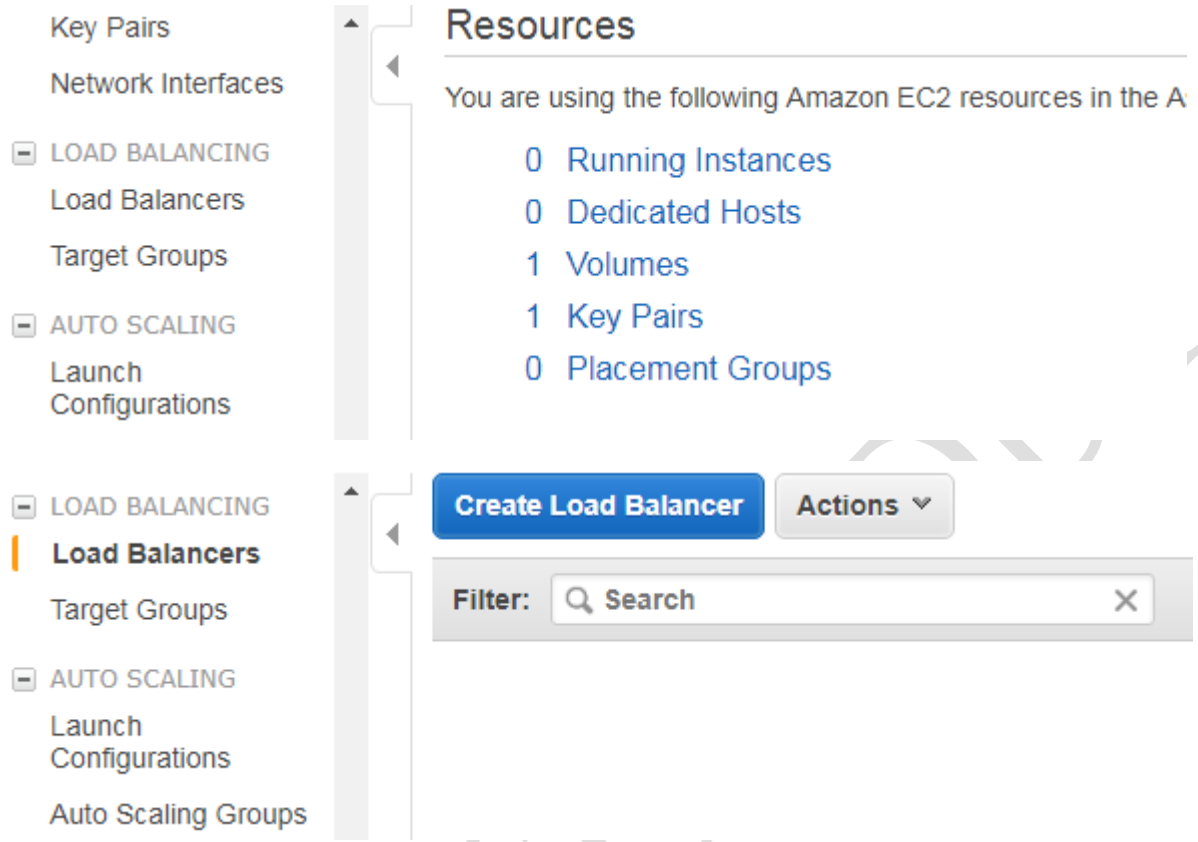
Listeners: Every load balancer must have one or more listeners configured. A listener is a process that checks for connection requests.

Health Checks

Elastic Load Balancing supports health checks to test the status of the Amazon EC2 instances behind an Elastic Load Balancing load balancer.

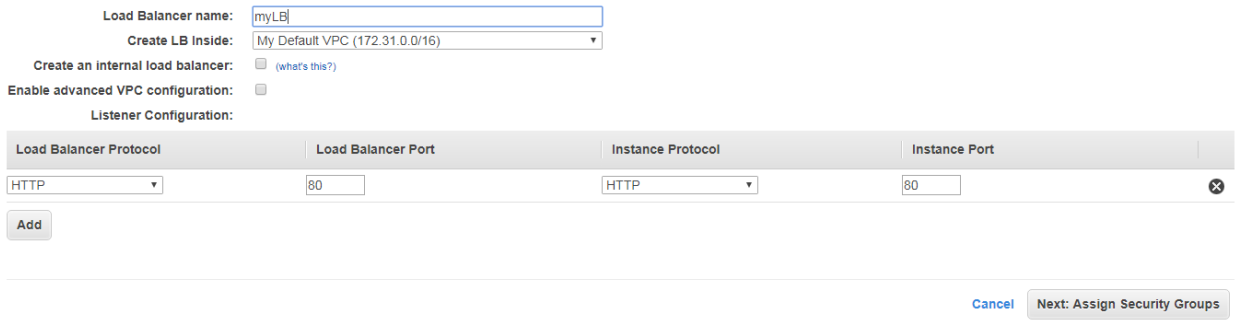
- The status of the instances that are healthy at the time of the health check is **InService**. The status of any instances that are unhealthy at the time of the health check is **OutOfService**.
- The load balancer performs health checks on all registered instances to determine whether the instance is in a healthy state or an unhealthy state.
- A health check is a ping, a connection attempt, or a page that is checked periodically. You can set the time interval between health checks and also the amount of time to wait to respond in case the health check page includes a computational aspect.
- We can set a Threshold for the number of consecutive health check failures before an instance is marked as unhealthy.

To create ELB navigate to EC2ManagementConsole. Next, from the navigation pane, select the Load Balancers option, this will bring up the ELB Dashboard as well, using which you can create and associate ELBs.

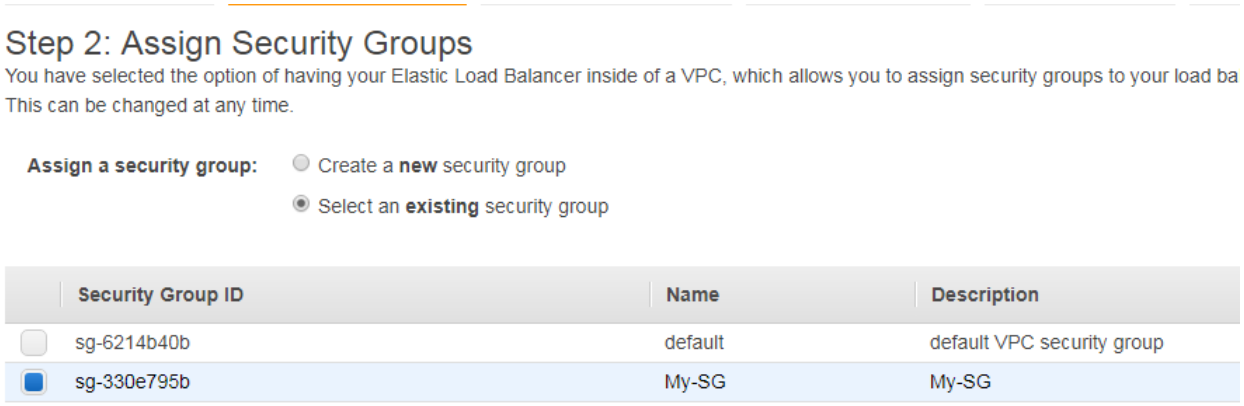


Step 1 – Defining the Load Balancer

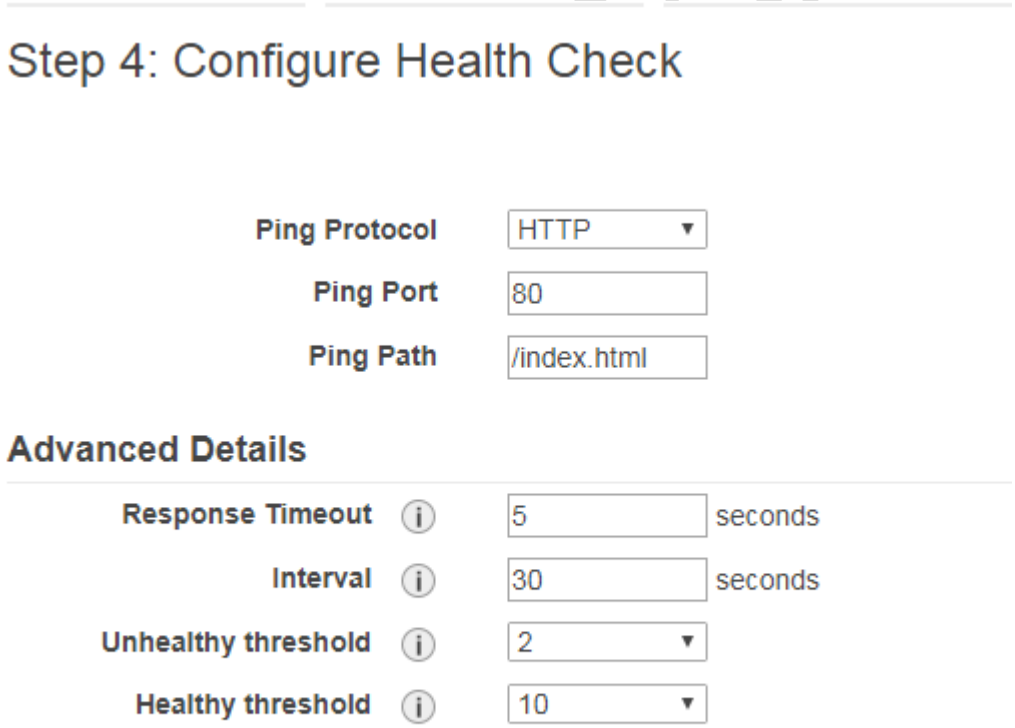
1. Select **Create Load Balancer** option and provide a suitable name for ELB in the Load Balancer name field. Next select the VPC option in which you wish to deploy ELB.
2. Do not check the Create an internal load balancer option as in this scenario, we are creating an Internet-facing ELB for Web Server.
3. In the Listener Configuration section, select HTTP from the Load Balancer Protocol drop-down list and provide the port number 80 in the Load Balancer Port field, as shown in the following screenshot. Provide the same protocol and port number for the Instance Protocol and Instance Port fields.



4. Here, We have to select the Security group for ELB



5. In Step 3 we have to configure security settings, This is an optional page that basically allows you to secure your ELB by using either the HTTPS or the SSL protocol for your frontend connection. But since we have opted for a simple HTTP-based ELB, we can ignore this page. Click on Next: **Configure Health Check** to proceed to the next step.
6. In step 4 we have to configure the health checks.



Ping protocol: This field indicates which protocol the ELB should use to connect to EC2 instances. We can use the TCP, HTTP, HTTPS, or the SSL options.

Ping port: This field is used to indicate the port which the ELB should use to connect to the instance.

Ping path: This value is used for the HTTP and HTTPS protocols. Can also use a /index.html here.

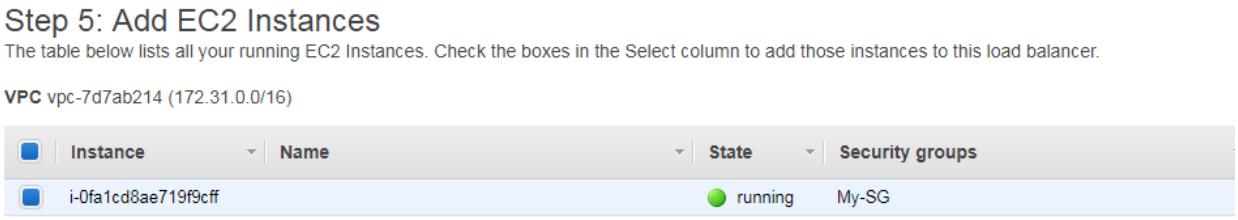
Response time: The Response Time is the time the ELB has to wait in order to receive a response. The default value is 5 seconds with a maximum value up to 60 seconds.

Health Check Interval: This field indicates the amount of time (in seconds) the ELB waits between health checks of an individual EC2 instance. The default value is 30. Maximum value is 300 seconds.

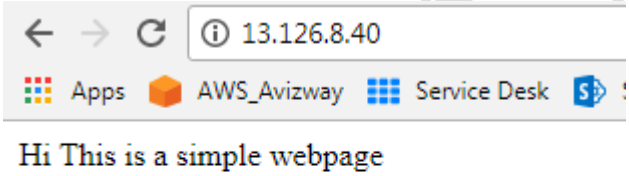
Unhealthy Threshold: This field indicates the number of consecutive failed health checks an ELB must wait before declaring an instance unhealthy. The default value is 2 with a maximum threshold value of 10.

Healthy Threshold: This field indicates the number of consecutive successful health checks an ELB must wait before declaring an instance healthy. The default value is 2 with a maximum threshold value of 10.

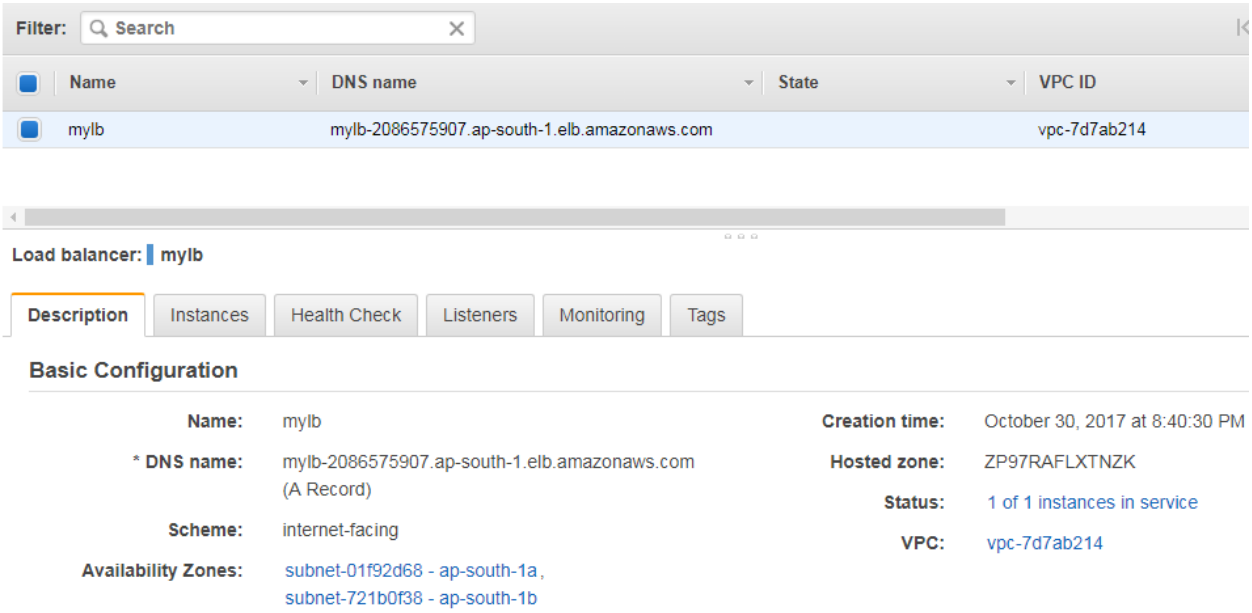
7. Step 5 – Add EC2 instances: We can select any running instance from Subnets to be added and registered with the ELB. Select the EC2 instances you want to launch under this ELS then Click on Next: Add Tags to proceed with the wizard.



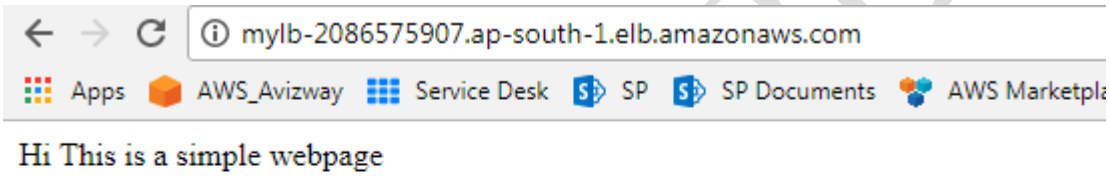
8. In next step, Add any of the tags required and Review the option and click on **Create** option.
9. I have a installed httpd package and created an Index.html file under /var/www/html path in ec2 instance then started the httpd service and am able to get the webpage using the Instance's public IP.



10. And Here is the details for created ELB, As we know we'll get a DNS name for our created ELB, We can access the same webpage by using the ELB's DNS name also.



11. We are able to get the same page by using the DNS name of ELB. This means our ELB configured successfully.



Auto Scaling

Auto Scaling is a service that allows us to scale our Amazon EC2 capacity automatically by scaling out and scaling in according to criteria that we define. With Auto Scaling, we can ensure that the number of running Amazon EC2 instances increases during demand spikes or peak demand periods to maintain application performance and decreases automatically during demand lulls or troughs to minimize costs.

Launch Configuration

A launch configuration is the template that Auto Scaling uses to create new instances, and it is composed of the configuration name, Amazon Machine Image (AMI), Amazon EC2 instance type, security group, and instance key pair. Each Auto Scaling group can have only one launch configuration at a time.

Auto Scaling Group

An Auto Scaling group is a collection of Amazon EC2 instances managed by the Auto Scaling service. Each Auto Scaling group contains configuration options that control when Auto Scaling should launch new instances and terminate existing instances. An Auto Scaling group must contain a name

and a minimum and maximum number of instances that can be in the group. You can optionally specify desired capacity, which is the number of instances that the group must have at all times. If you don't specify a desired capacity, the default desired capacity is the minimum number of instances that you specify.

Scaling plans

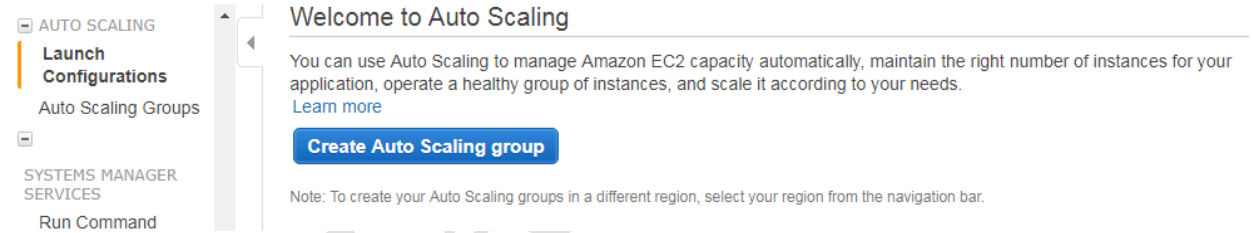
With your Launch Configuration created, the final step left is to create one or more scaling plans. Scaling Plans describe how the Auto Scaling Group should actually scale.

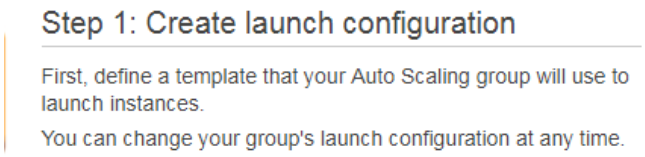
- Manual scaling: here is specify a new desired number of instances value or change the minimum or maximum number of instances in an Auto Scaling Group and the rest is taken care of by the Auto Scaling service itself
- Scheduled scaling: We can scale resources based on a particular time and date
- Dynamic scaling: Dynamic scaling, or scaling on demand is used when the predictability of your application's performance is unknown.

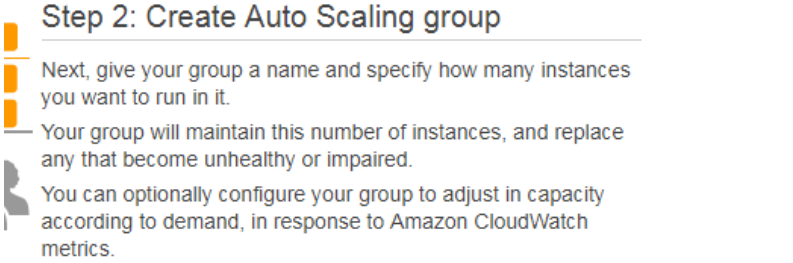
Auto scaling group creation involves with two steps. First one is Creating a Launch Configuration and second is Creating Auto Scaling group.

Creating the Launch Configuration steps

1. Go to **EC2 Management Dashboard** option, select the **AutoScaling Groups** option from the navigation pane. This will bring up the Auto Scaling Groups dashboard. Next, select the **Create Auto Scaling group** option to bring up the Auto Scaling setup wizard.





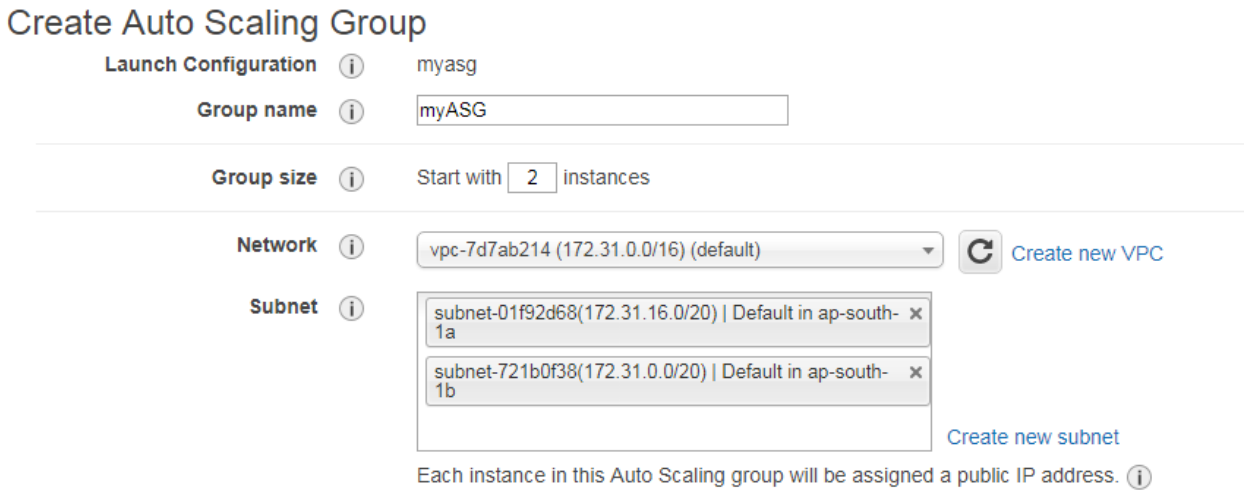


2. Select Create launch configuration is similar to the instance launch wizard. If you have any custom AMIs you can select here.
3. Give a valid name for the Launch configuration. Choose Instance configuration, Storage options, security groups, tags and key pairs and select Create Launch Configuration to complete the process

Step 2: Creating the Auto Scaling Group

An Auto Scaling Group is nothing more than a logical grouping of instances that share some common scaling characteristics between them. Each group has its own set of criteria specified which includes the minimum and maximum number of instances that the group should have along with the desired number of instances which the group must have at all times.

4. When we completes with creating launch configuration, it will take us to Step 2, Here we have to give a name for the Group, We can select the Group size and VPC.



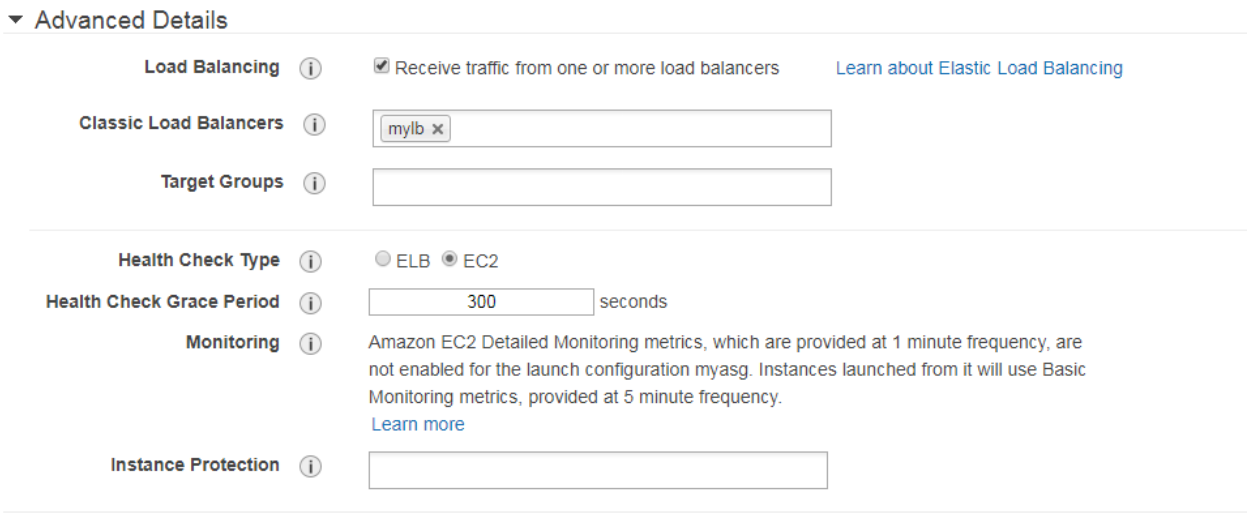
Each instance in this Auto Scaling Group will be provided with a public IP address.

5. We can expand Advanced details option to configure.

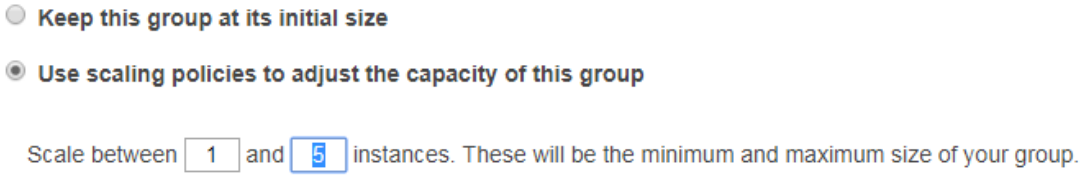
Load Balancing: These are optional settings that you can configure to work with your Auto Scaling Group. Since we have already created and configured our ELB, we will be using that itself to balance out incoming traffic for our instances. Select the Receive traffic from Elastic Load Balancer option.

Health Check Type: You can use either your EC2 instances or even your ELB as a health check mechanism to make sure that your instances are in a healthy state and performing optimally. By default, Auto Scaling will check your EC2 instances periodically for their health status. If an unhealthy instance is found, Auto Scaling will immediately replace that with a healthy one.

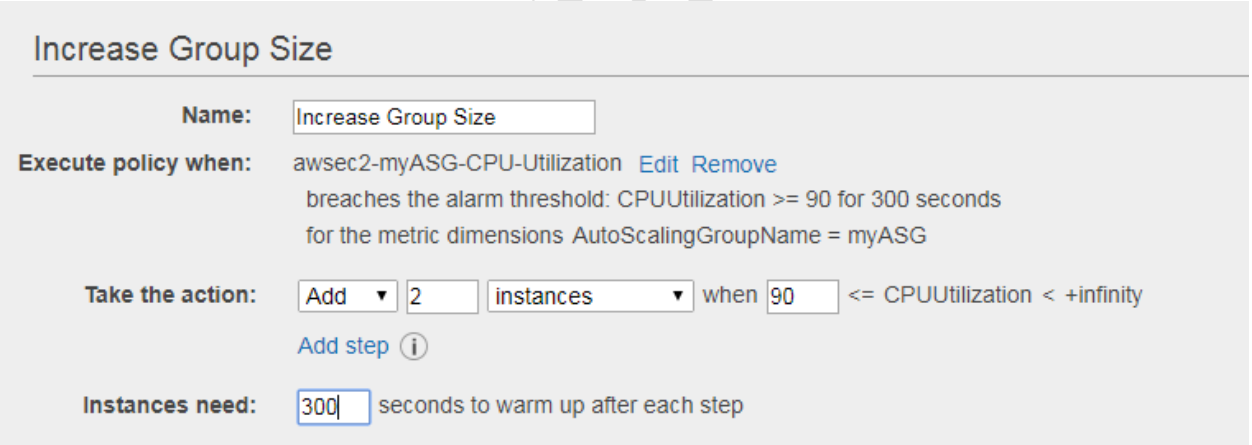
Health Check Grace Period: Enter the health check's grace period in seconds. By default, this value is set to 300 seconds.



6. Step 2 of ASG creation is Configure scaling policies: This is the important part of creating any Auto Scaling Group is defining its scaling policies.



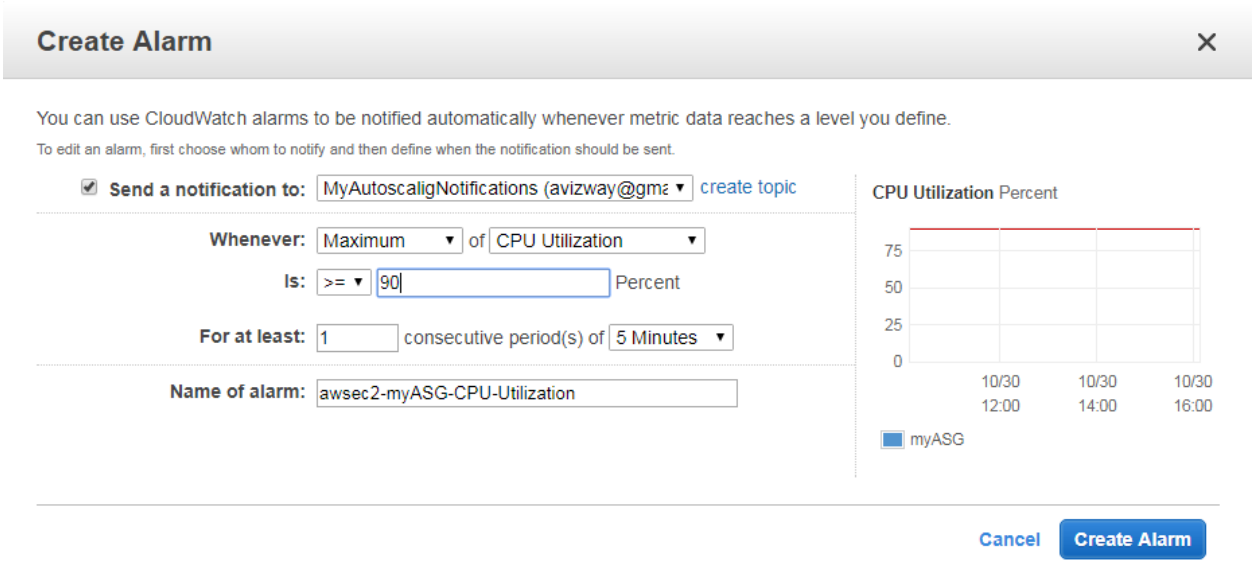
7. Selecting the scaling policies option.



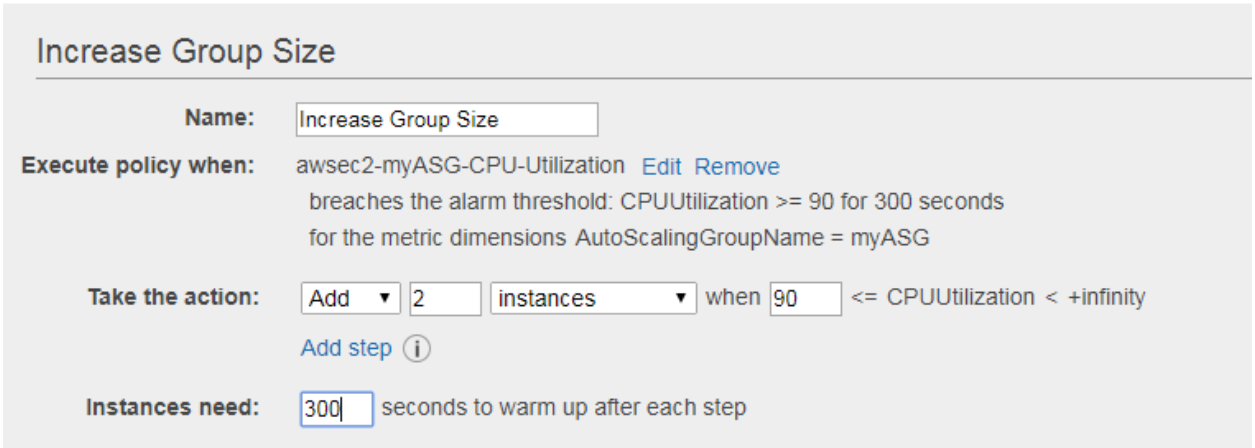
Name: Provide a suitable name for your scale-out policy.

Execute policy when: Here we have to select a pre-configured alarm using which the policy will get triggered. Since this is our first time configuring, select the **Add new alarm** option. This will pop up the Create Alarm dialog,

Creating the alarm is a very simple process; for example, we want our Auto Scaling Group to be monitored based on the CPU Utilization metric for an interval of 5 minutes. If the average CPU Utilization is greater than or equal to 90 percent for at least one consecutive period, then send a notification mail to the specified SNS Topic. click on Create Alarm.

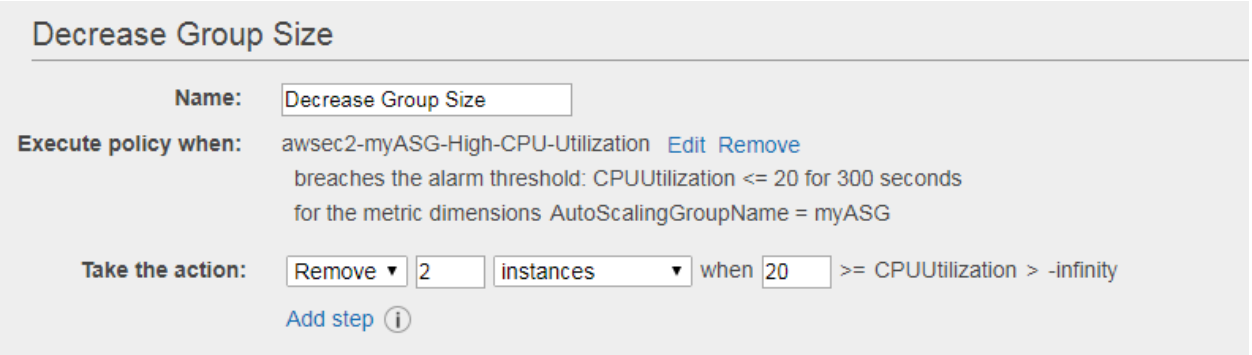


Take the action: Now we can define the policy what action it has to take if the particular threshold is breached. Select Add from the dropdown list and provide a suitable number of instances that you wish to add when a certain condition matches.

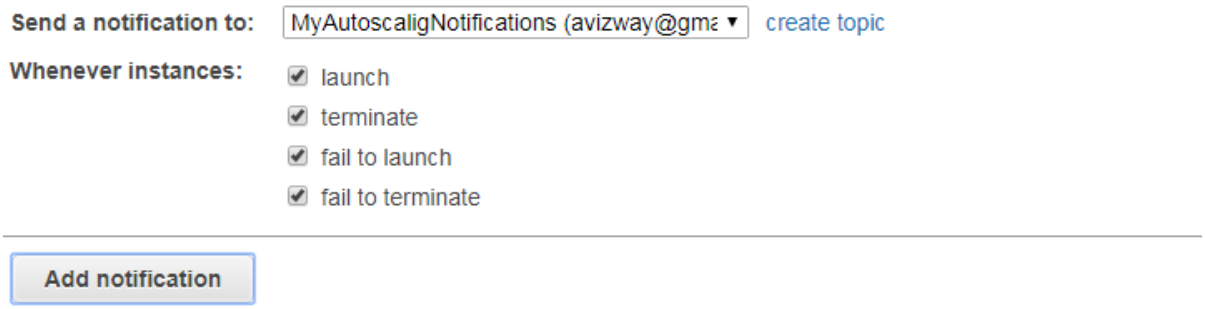


Instances need: The final field is the Cooldown period. By default, this value is set to 300 seconds and can be changed as per your requirements. A Cooldown period is like a grace period that we assign to the Auto Scaling Group to ensure that we don't launch or terminate any more resources before the effects of previous scaling activities are completed.

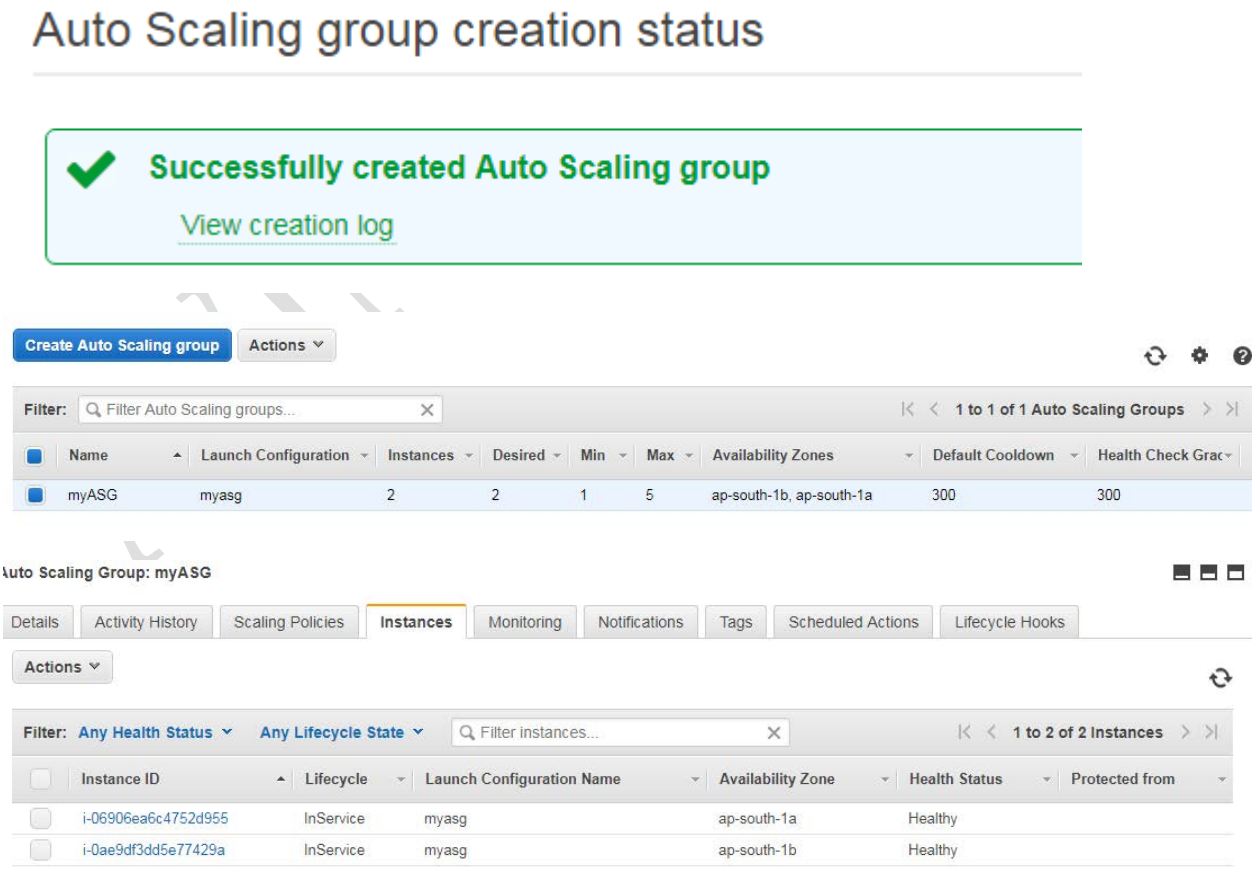
8. By the same way we can configure policies for Decrease Group Size also



9. Select the **Next: Configure Notifications** to proceed with the next steps
10. You can select Add Notification button and select an existing SNS topic or create a new.



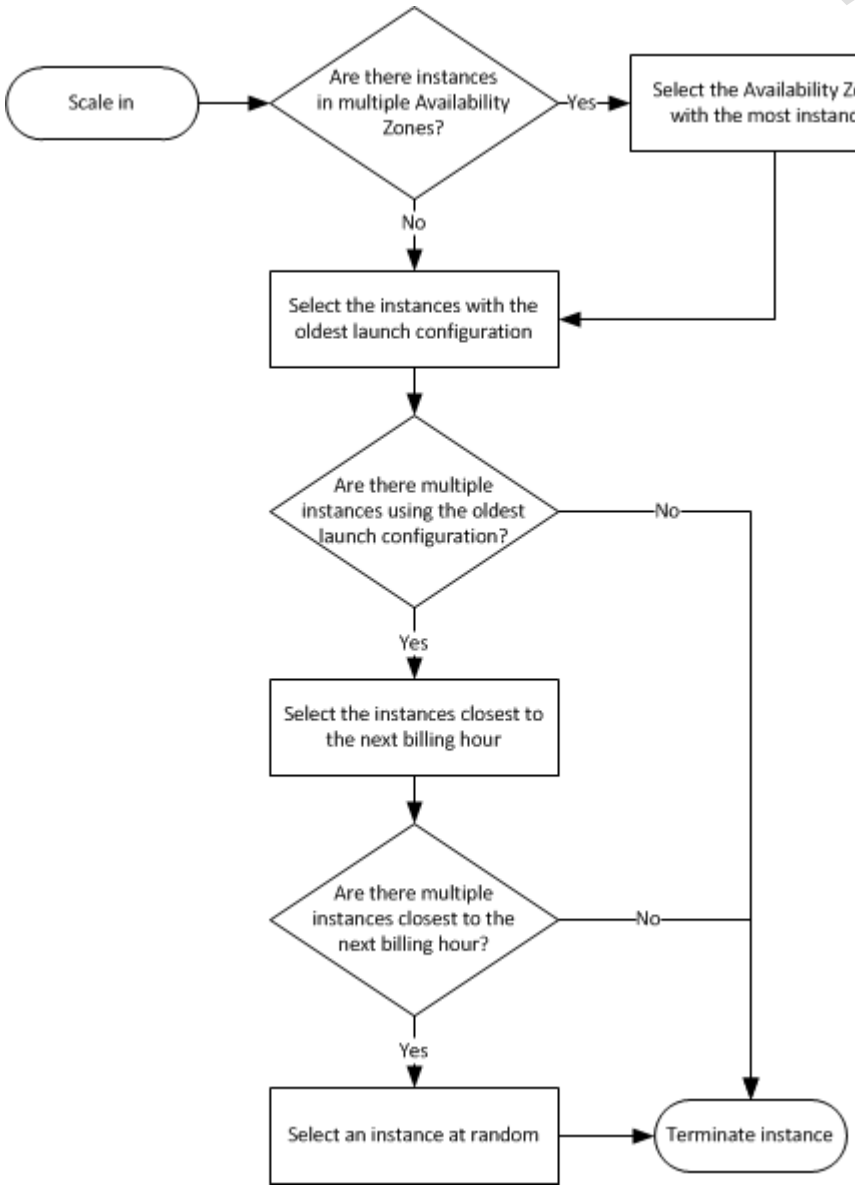
11. Select the review option and Click on Create Auto Scaling option to finish the process.



Default Termination Policy for Auto Scaling Group:

1. If there are instances in multiple Availability Zones, select the Availability Zone with the most instances and at least one instance that is not protected from scale in. If there is more than one Availability Zone with this number of instances, select the Availability Zone with the instances that use the oldest launch configuration.
2. Determine which unprotected instances in the selected Availability Zone use the oldest launch configuration. If there is one such instance, terminate it.
3. If there are multiple instances that use the oldest launch configuration, determine which unprotected instances are closest to the next billing hour. If there is one such instance, terminate it.
4. If there is more than one unprotected instance closest to the next billing hour, select one of these instances at random.

Here is a diagram that shows how the default termination policy works for ASG.



USER DATA:

When you launch an instance in Amazon EC2, you have the option of passing user data to the instance that can be used to perform common automated configuration tasks and even run scripts after the instance starts.

You can pass two types of user data to Amazon EC2: shell scripts and cloud-init directives. You can also pass this data into the launch wizard as plain text, as a file (this is useful for launching instances using the command line tools), or as base64-encoded text.

Here is a simple User Data script to use with **Linux EC2 instances** to make as a simple webserver with a simple index.html page.

```
#!/bin/bash
yum update -y
yum install httpd -y
echo "Hi This is a Bootstrap script generated webpage" > /var/www/html/index.html
servicehttpd start
chkconfighttpd on
```

“yum update” for updating the Operating system with latest security patches.

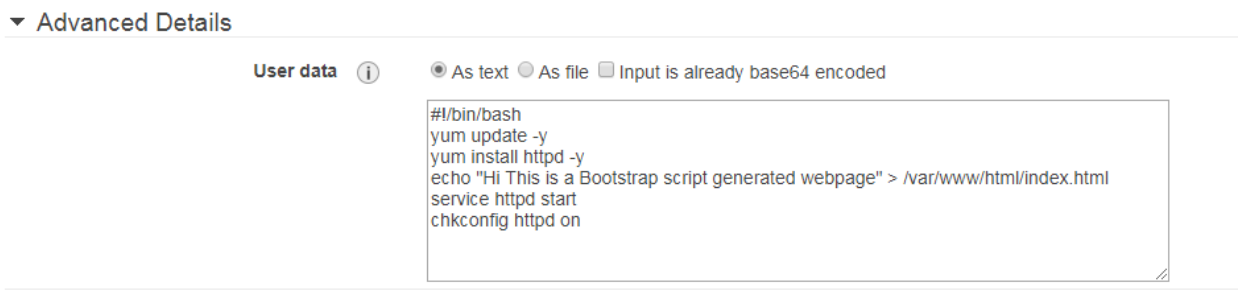
“Yum install httpd” for installing Apache to make this instance as a webserver

By Using echo command generating a string and copying the generated string to a file named “index.html” and saving the file under “/var/www/html” directory.

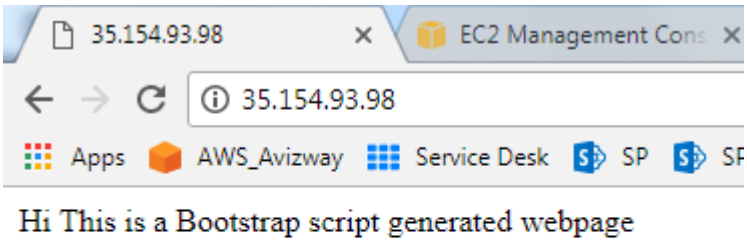
“Service httpd start” to start the apache service

“Chkconfighttpd on” starting and turning the service on / startup service.

1. While launching instance I’ve entered the bootstrap scripting



2. Then launching the instance and entering the public IP in the web browser without connecting to my instance. (Make sure port 80 open in the Security groups)



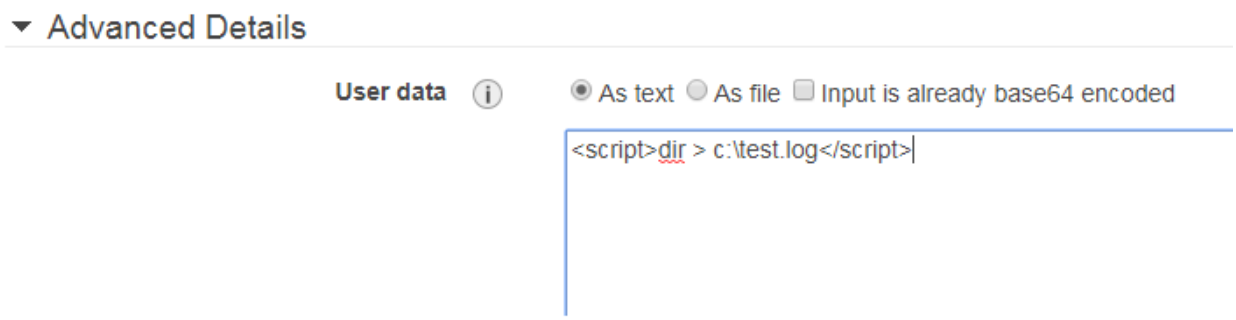
3. We got the output without login to the instance.

For Windows:

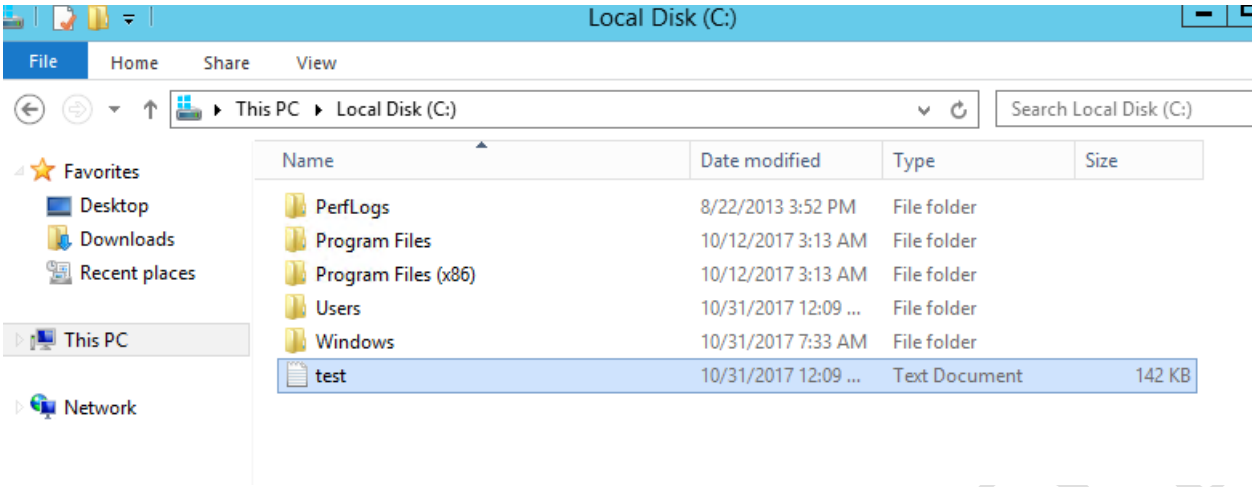
For EC2Config or EC2Launch to execute user data scripts, you must enclose the lines of the specified script within one of the following special tags:

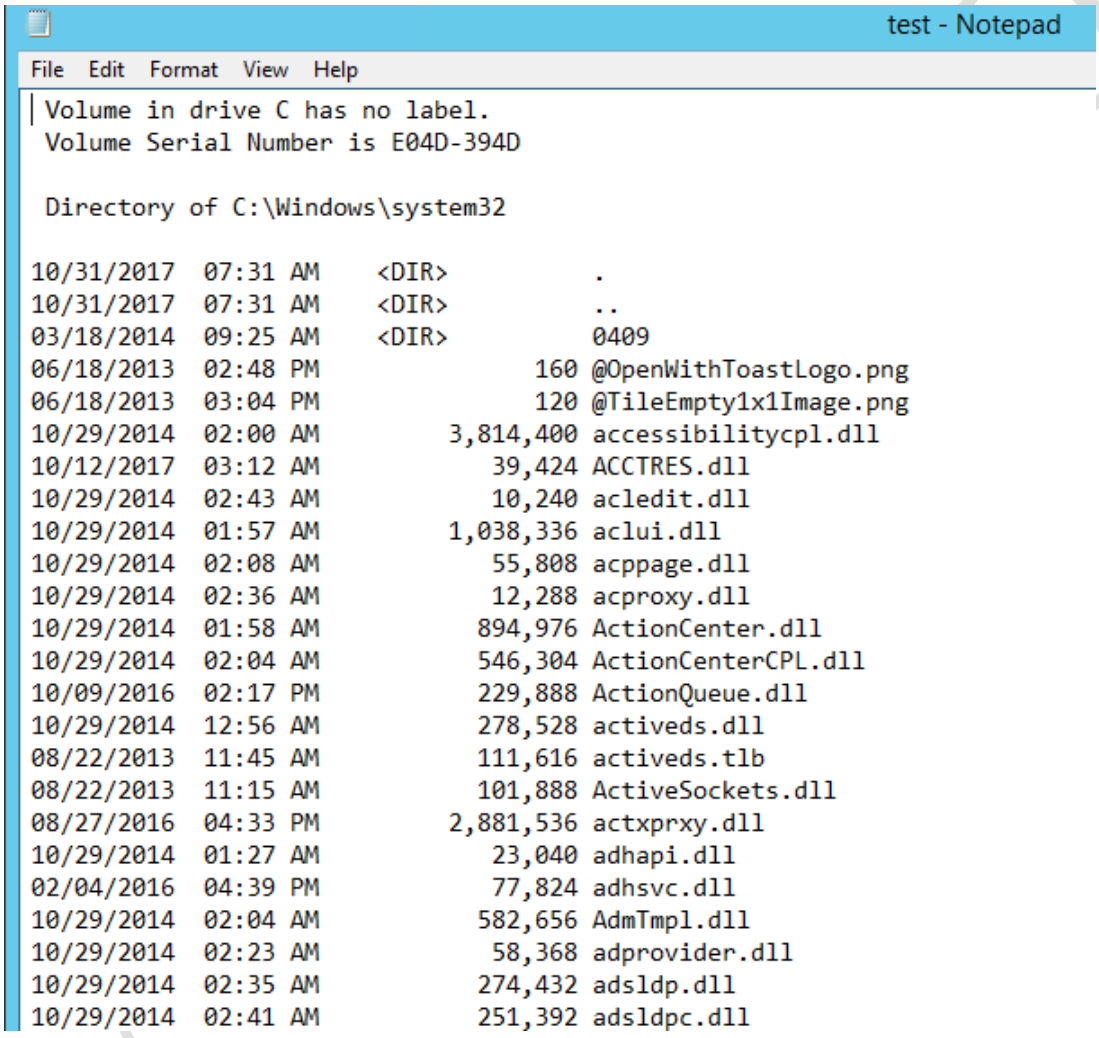
<script></script>

Example: <script>dir> c:\test.log</script>



1. Here we have run very simple script get directory information to a log file. New doc is created with all the information of the given directory.
2.





AWS CLI (Command Line Interface):

The AWS Command Line Interface (CLI) is a unified tool to manage AWS services. With just one tool to download and configure, you can control multiple AWS services from the command line and automate them through scripts.

- We can download the AWS tolls by using this URL: https://aws.amazon.com/cli/
- You can select the setup file based on your system architecture, if you are a windows user.
- Amazon Linux will get the CLI tools pre-installed.

Windows

Download and run the 64-bit or 32-bit Windows installer.

Mac and Linux

Requires Python 2.6.5 or higher.

Install using pip.

```
pip install awscli
```

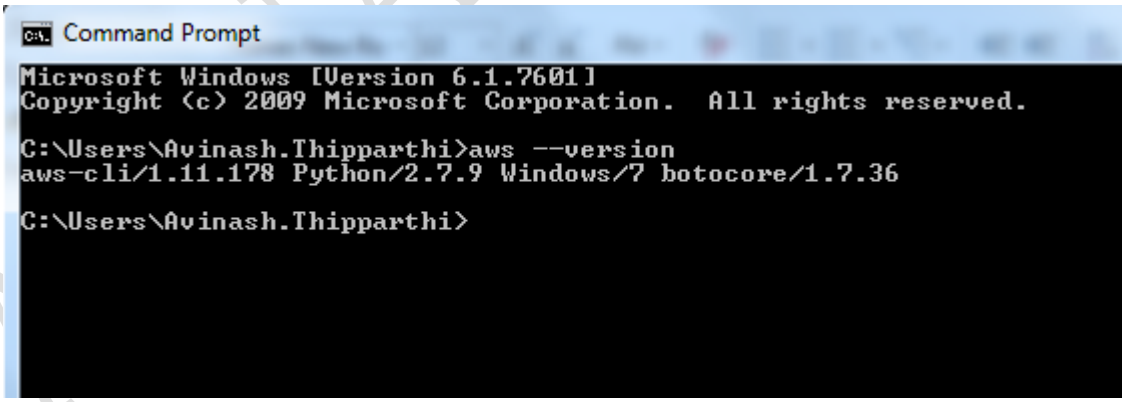
Amazon Linux

The AWS CLI comes pre-installed on Amazon Linux AMI.

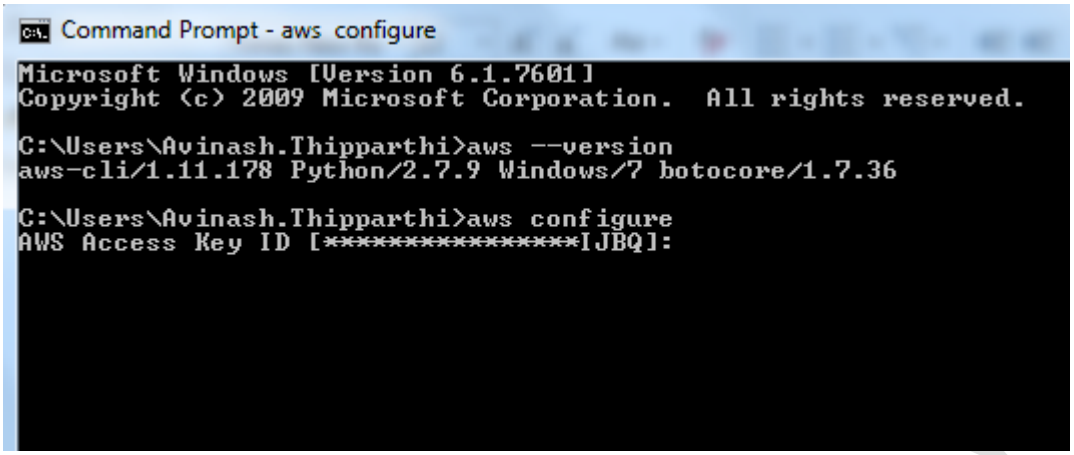
- Here is the url to get all the commands for each and every AWS service: http://docs.aws.amazon.com/cli/latest/reference/

Steps to configure CLI tools on windows Operating systems:

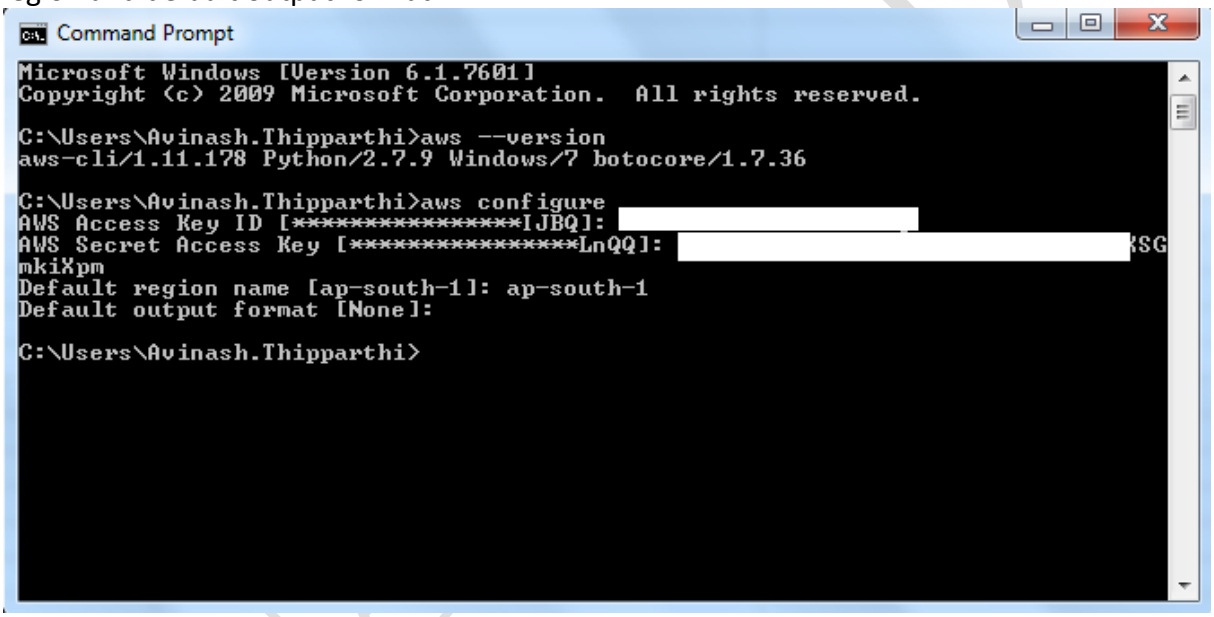
1. First we have to download the setup file from the above mentioned webpage, then follow the simple installation wizard.
2. After installing these tools, we can use the windows command prompt to connect to AWS resources/services.
3. To verify CLI tools installation, open command prompt and enter **"AWS –version",** it should return with installed version information as the below image.



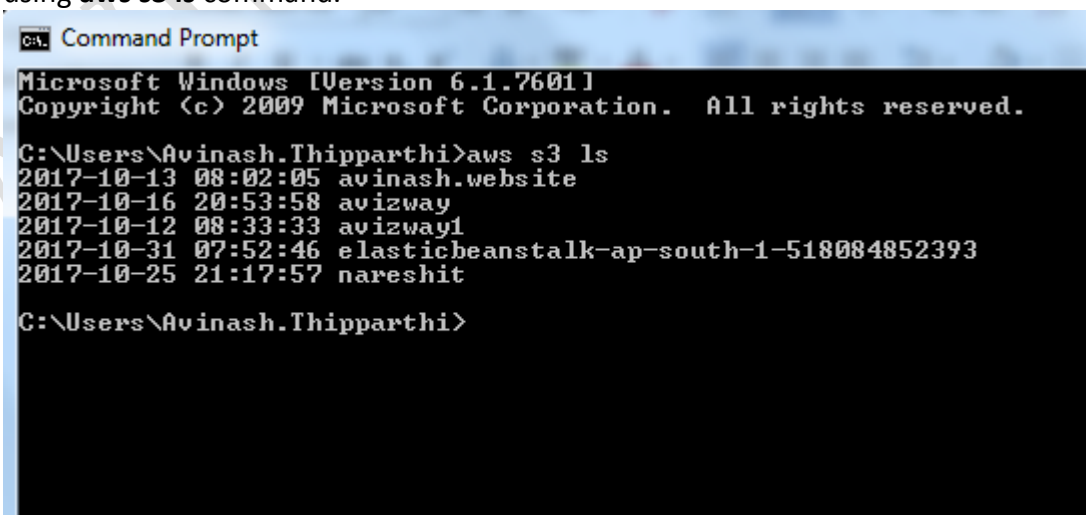
4. But we cannot configure CLI tools using IAM Management console access users, we need to have Programmatic Access IAM user.
5. When we create a Programmatic Access IAM user we will get **Access key ID** and **Secret Access Key**. Please create a user and allocate appropriate permissions.
6. To configure IAM user in local windows machine, we have to "**AWS configure**" command.



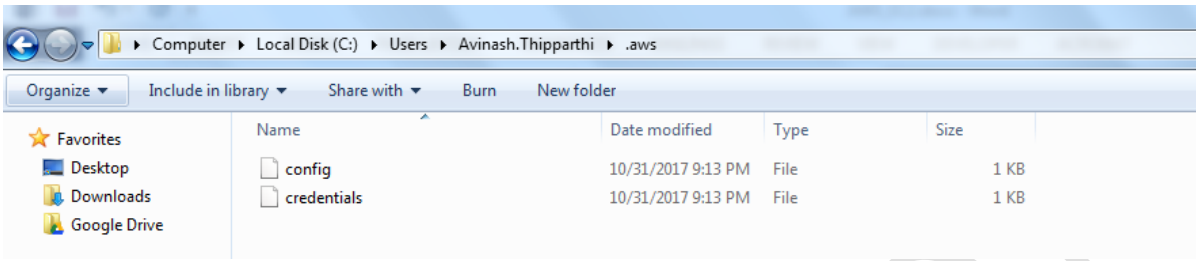
7. Enter the AWS Access Key ID and then enter the Secret Access key, choose the default region and default output format.



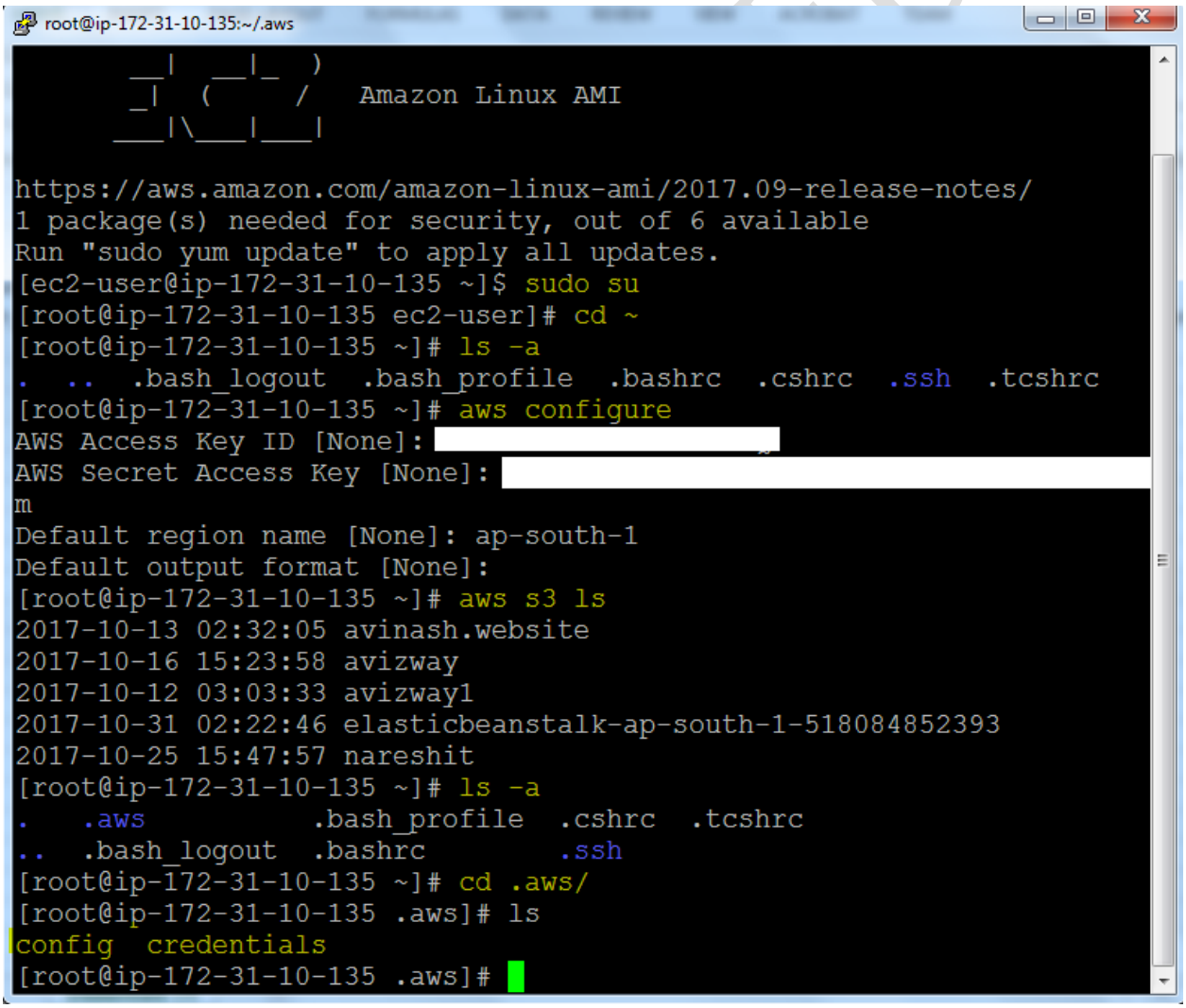
8. We have successfully configured the CLI tools and now try to access any of the AWS resource from the CLI configured device. Here am trying to list my S3 buckets for that am using **aws s3 ls** command.



9. We are able to get the details that means we are connecting to AWS account resources by using the Programmatic access IAM user credentials.
10. But, the IAM user credentials will store in a directory called .aws , In windows the path is **C:\Users\WindowsUserName\.aws** , if you open credentials file, we will get the Configured IAM user's Aceess Key ID and Secret Access Key.



11. In Linux, The .aws directory will store under / (root) and It is a hidden directory, we can give **ls –a** command to get it, and inside the .aws directory we will have config and credentials files.



12. In the above image, I've logged into the linux instance and switched to root, looked for .aws directory, but it is not existed. Then Configured the IAM user with Access Key IA and Secret Access Key and accessed the AWS resources and we get the required resource information.

13. After installing CLI IAM user, we got .aws directory under **/** (give **ls –a** to verify), inside that .aws directory we have config and credentials files, Credential file will contains the Access Key id and secret access key.
14. So this is not a secure method, anybody can view these credentials and configure CLI tools on their own machines and they may access, So amazon will **recommend to use the ROLES** instead of storing the credentials in local machines.

Policy: A policy is a JSON document that fully defines a set of permissions to access and Manipulate AWS resources. Policy documents contain one or more permissions.

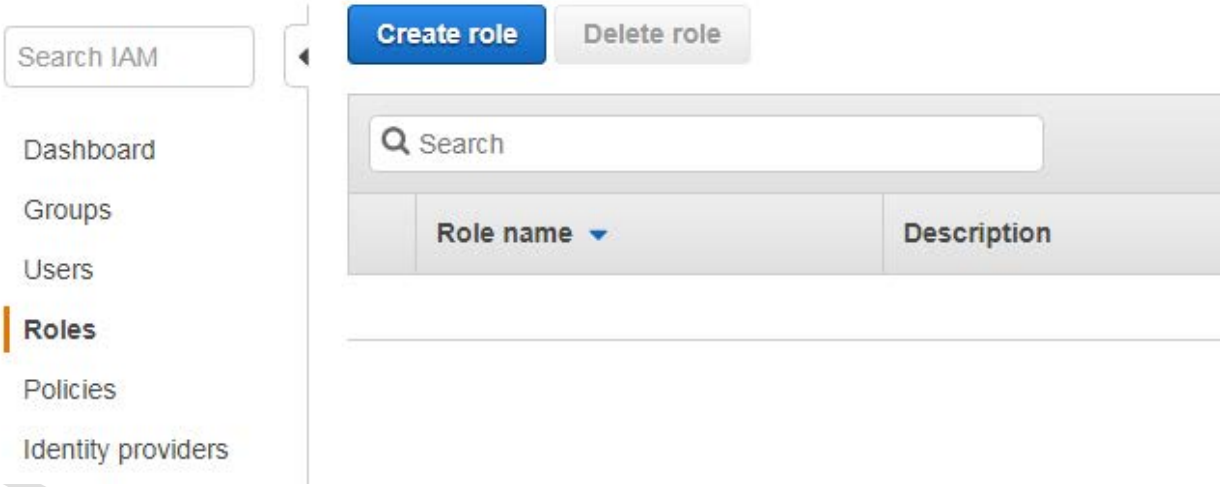
IAM ROLES:

Roles are used to allow AWS services to perform actions on your behalf. Roles are used to grant specific privileges to specific actors.

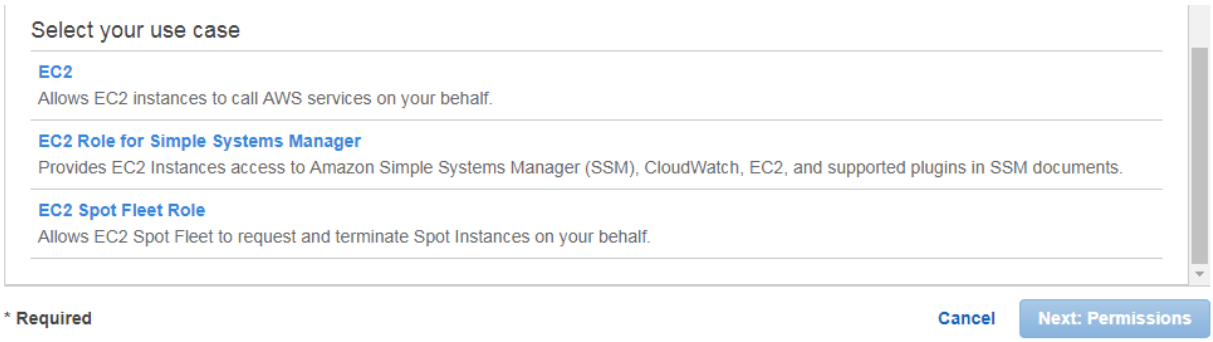
- Roles are more secure than storing your access key and secret access key on individual EC2 instances.
- Roles are easier to manage
- We can attach or Remove role to a running instance now. Previously this option is not available.
- Roles are universal, you can use them in any region.

Steps to create a role and attaching to EC2 instance.

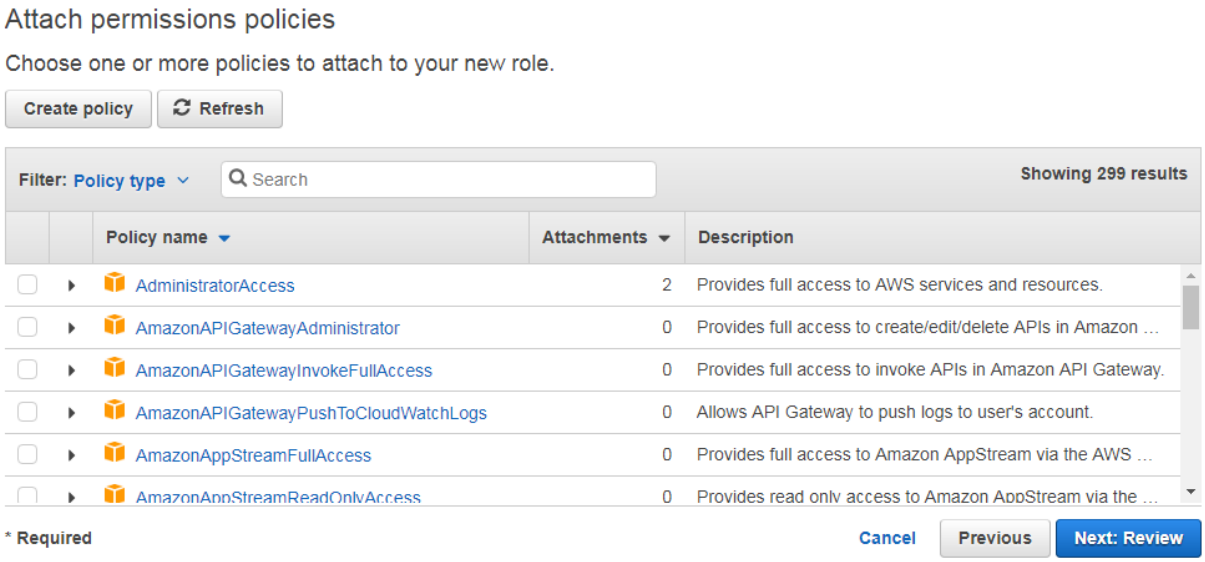
1. Navigate to IAM dashboard to create an IAM role.
2. Select Roles option from dashboard and select **"Create Role"** option.



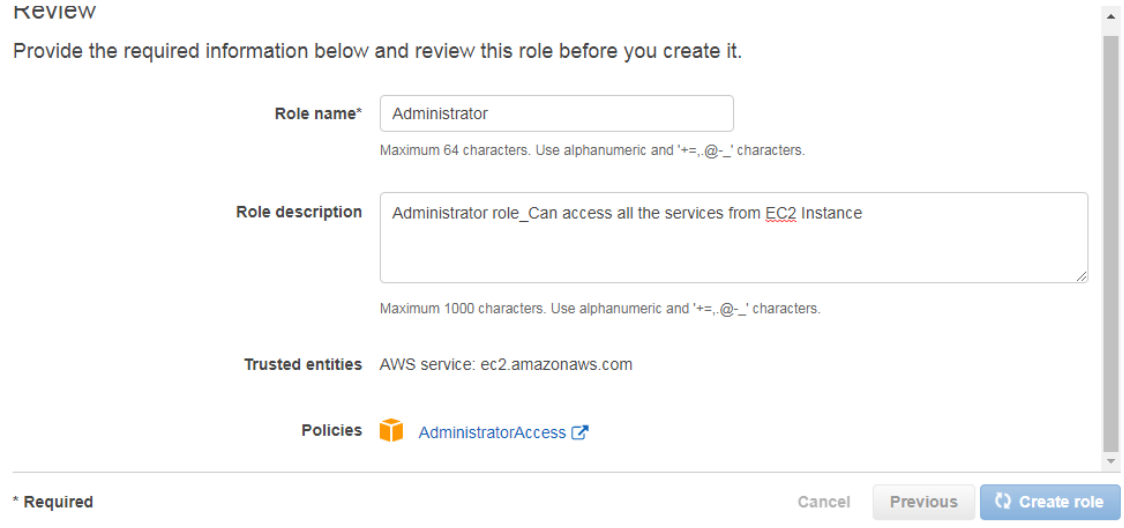
3. We have four option in the roles, We are going to create this role under "AWS Services", and select the **EC2.**
4. After selecting EC2, we have to select the appropriate Use Case. We would like to call some AWS services on our behalf to the EC2 instance. Select EC2 and click on **Next: Permissions** button.



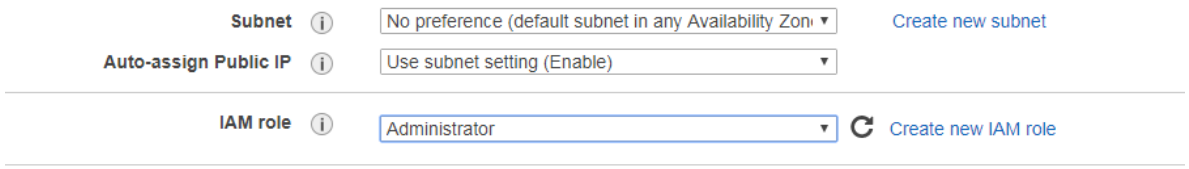
5. In this step, we have to select the policy, you can generate a new policy based on your requirement or choose existing policy.



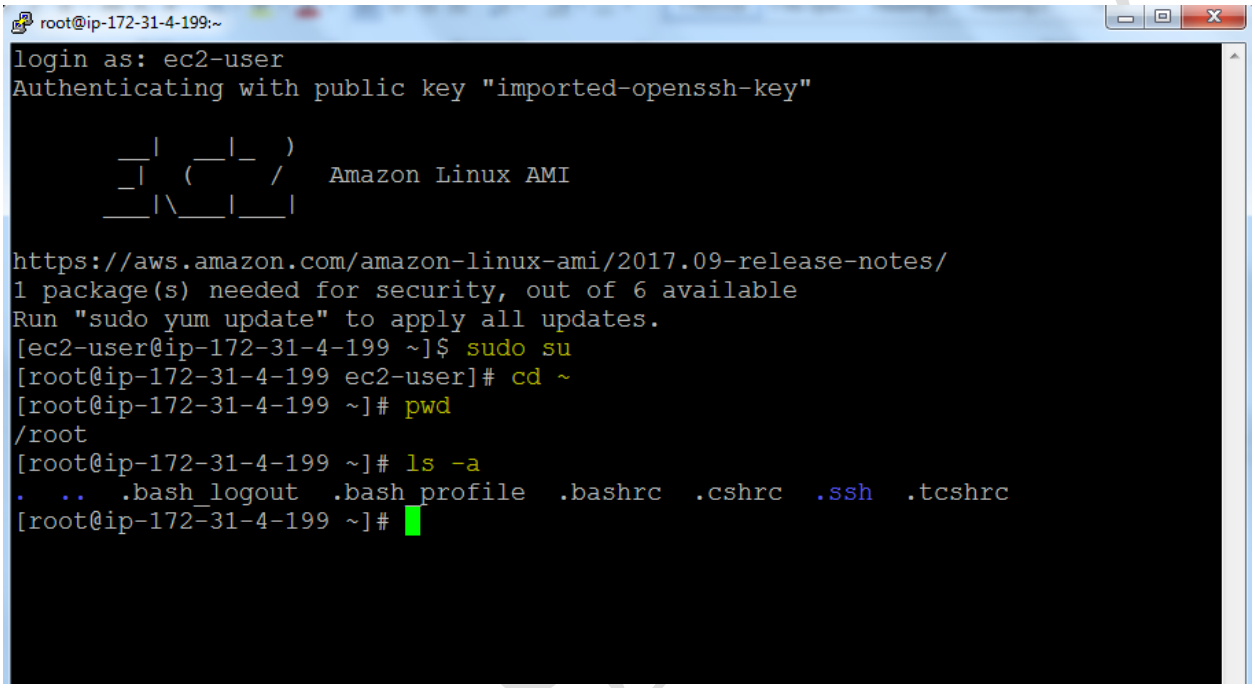
6. Select appropriate role, based on your requirement, am selecting AdministratorAccess role here. Then Select **Review.**
7. In review page, Give a name for the role and a valid description and select **Create Role** option.



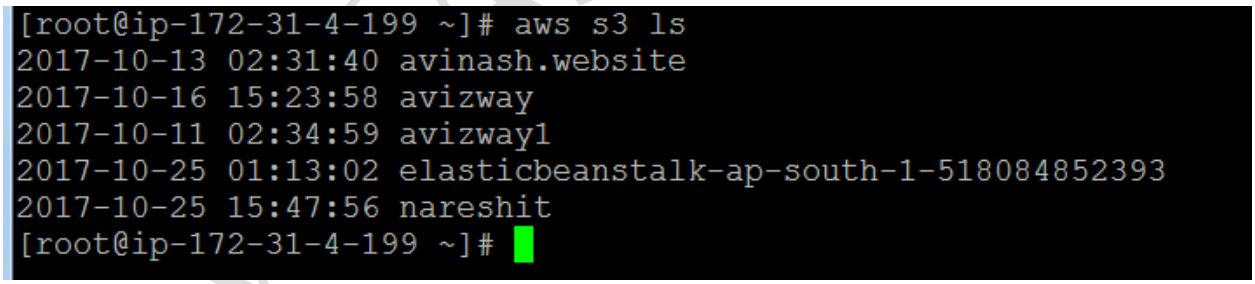
8. Now launch an EC2 instance and try to access/call any AWS service to verify the role.



9. Logged into EC2 instance and elevated privileges to root and trying to find the .aws directory under / , but we cannot find, That means we don't have any credentials on instance.



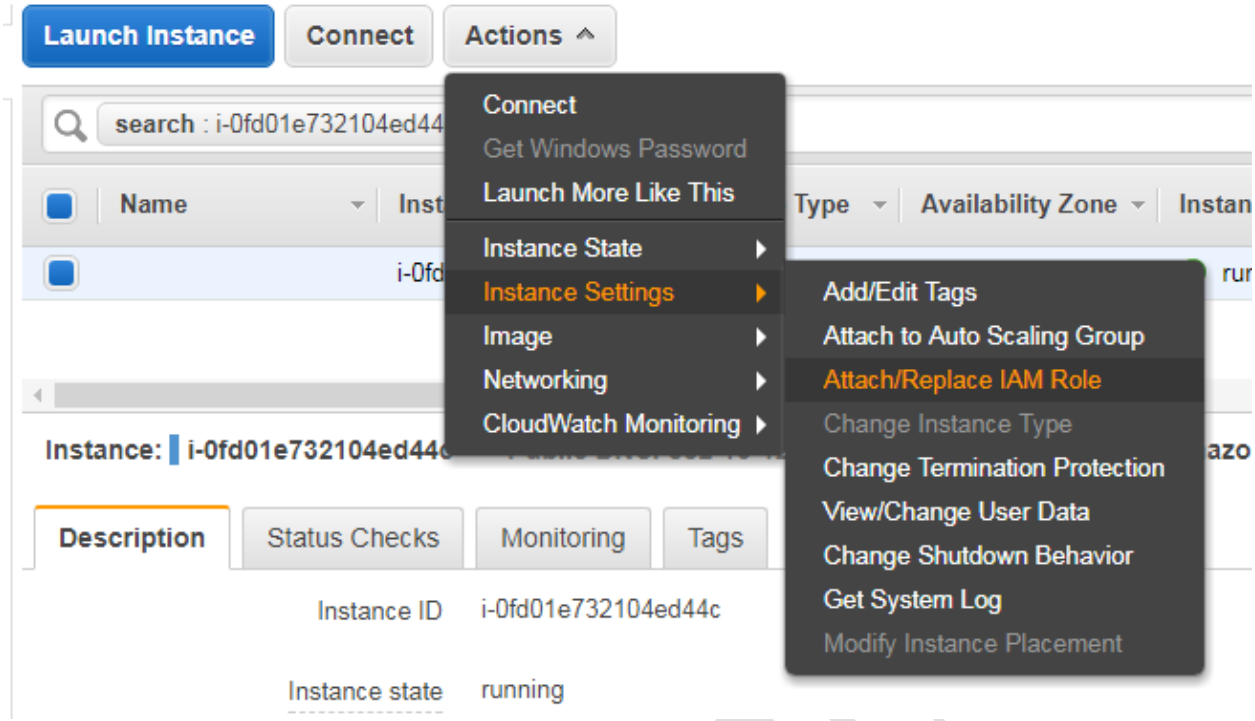
10. Try to access any AWS service, here am trying to list the S3 buckets by **AWS s3 ls** command.



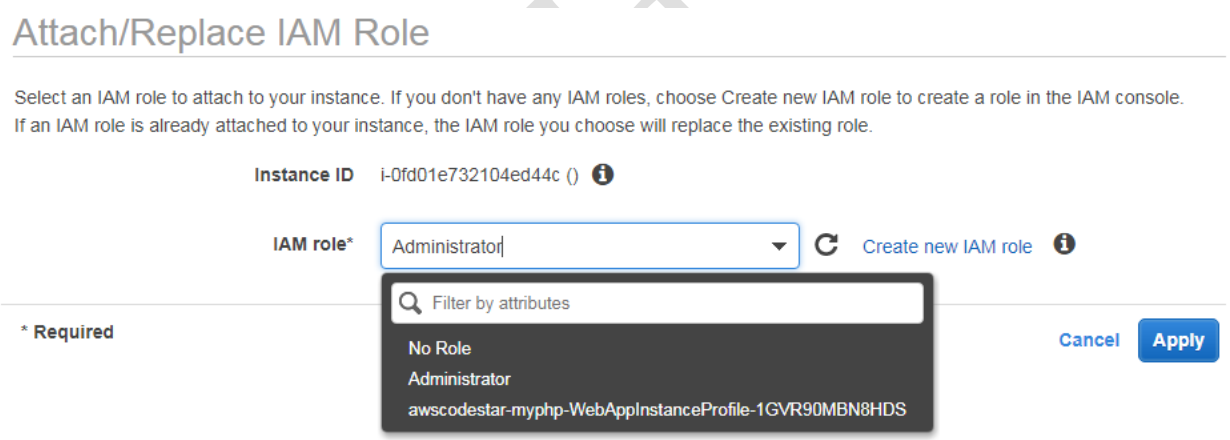
11. We are able to access the resources and nowhere storing the Access key ID and Secret Access key.

Steps to Attach/Replace role from a Running Instance:

1. Select the Instance and go to Actions button and we can find Attach/Replace IAM Role under Instance Settings.



2. Select IAM role filed, automatically it will dropdown the available roles along with No Role option, Select the required option and click on Apply. It will take effect immediately.



Instance Metadata:

Instance metadata is data about your instance that you can use to configure or manage the running instance. This is unique in that it is a mechanism to obtain AWS properties of the instance from within the OS. By using below URL we can query the local instance metadata.

- Curl http://169.254.169.254/latest/meta-data/
- When you enter this URL, it'll return with all the available information to get. We can give the required option after meta-data/ you'll get the information.

Steps to get the instance Metadata:

1. I've logged into my EC2 instance
2. Enter the metadata url

```
[root@ip-172-31-23-113 ec2-user]# curl http://169.254.169.254/latest/meta-data/
ami-id
ami-launch-index
ami-manifest-path
block-device-mapping/
hostname
instance-action
instance-id
instance-type
local-hostname
local-ipv4
mac
metrics/
network/
placement/
profile
public-hostname
public-ipv4
public-keys/
reservation-id
security-groups
services/[root@ip-172-31-23-113 ec2-user]#
```

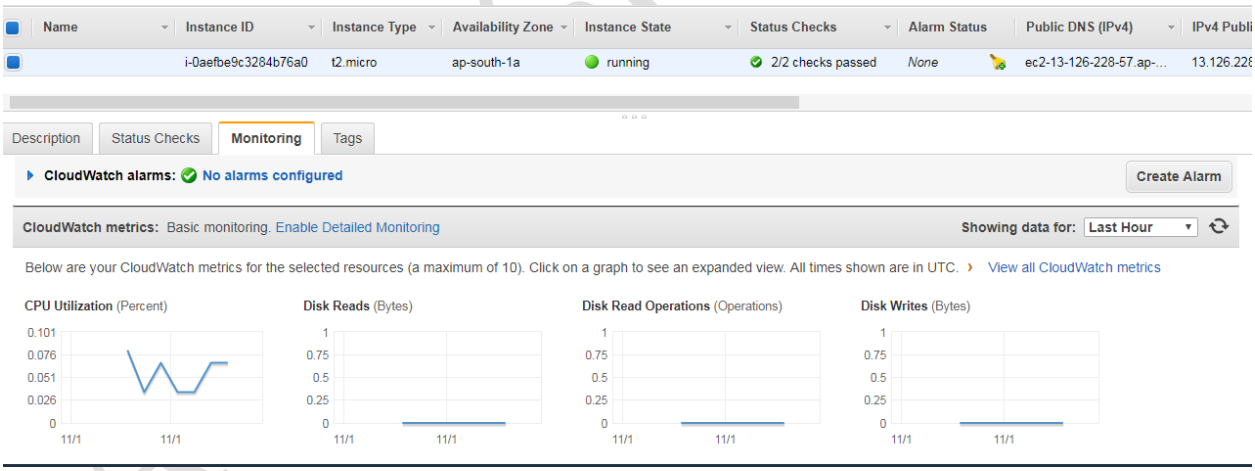
3. It is returned with all the available option, now whatever the information you want to get, give it along with the URL.
   Ex: if you want to know hostname, give as Curl http://169.254.169.254/latest/meta-data/hostname

```
[root@ip-172-31-23-113 ec2-user]# curl http://169.254.169.254/latest/meta-data/hostname
ip-172-31-23-113.ap-south-1.compute.internal[root@ip-172-31-23-113 ec2-user]#
```

AWS CLOUDWATCH

Amazon CloudWatch is a service that you can use to monitor your AWS resources and your applications in real time. With Amazon CloudWatch, you can collect and track metrics, create alarms that send notifications, and make changes to the resources being monitored based on rules you define.

- You can specify parameters for a metric over a time period and configure alarms and automated actions when a threshold is reached.
- Amazon CloudWatch offers either basic or detailed monitoring for supported AWS products.
- Basic monitoring sends data points to Amazon CloudWatch every five minutes for a limited number of preselected metrics at no charge.
- Detailed monitoring sends data points to Amazon CloudWatch every minute and allows data aggregation for an additional charge. If you want to use detailed monitoring, you must enable it—basic is the default.
- AWS provides a rich set of metrics included with each service, but you can also define custom metrics to monitor resources and events.
- Amazon CloudWatch Logs can be used to monitor, store, and access log files from Amazon EC2 instances.
- Amazon CloudWatch Logs can also be used to store your logs in Amazon S3 or Amazon Glacier.
- Each AWS account is limited to 5,000 alarms per AWS account, and metrics data is retained for two weeks by default.



Sample image for EC2 instance cloudwatchmonitorings.

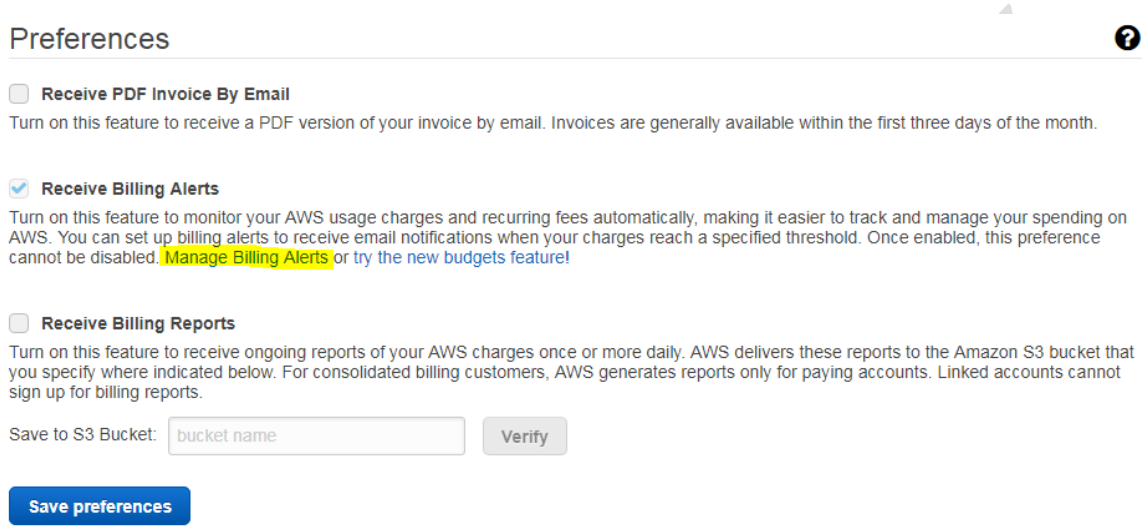
Metrics: Metrics form the core of Amazon CloudWatch's functionality. Essentially, these are nothing more than certain values to be monitored. Each metric has some data points associated with it which tend to change as time progresses.

Alarms: An alarm basically watches over a particular metric for a stipulated period of time and performs some actions based on its trigger. These actions can be anything from sending a notification to the concerned user using the Simple Notification Service (SNS).

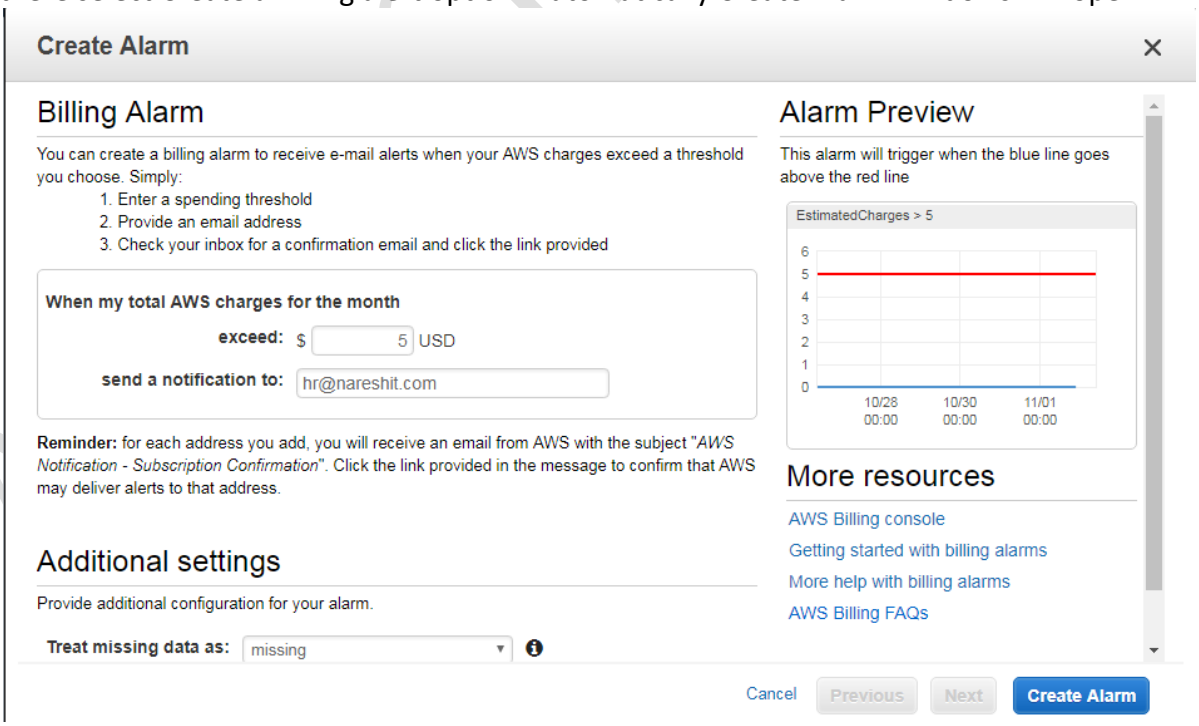
Monitoring your account's estimate charges using CloudWatch

You can configure the alerts on your AWS usage by using the Cloudwatchh alarms. Here is the steps to create an alarm on estimated charges.

1. Login with root account credentials.
2. Select My Account option and navigate to "**Preferences**"
3. Go to Select Receive Billing Alerts checkbox and select **"ManageBilling Alerts**" option. (Cloudwatch alarms will create in North Virginia region).

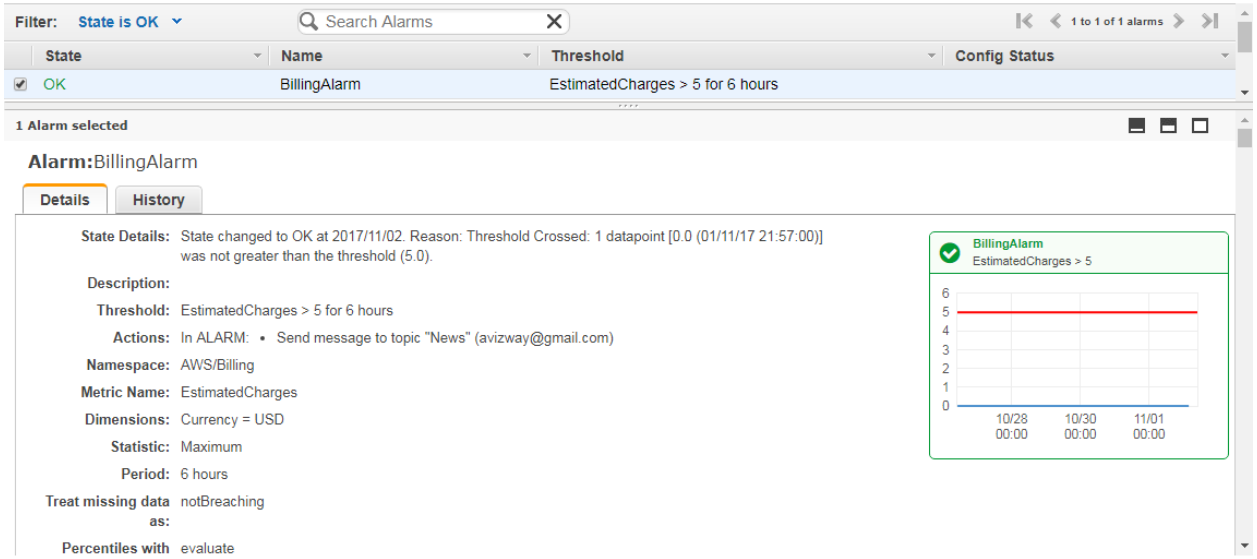


4. When you click on **"ManageBilling Alerts**" option, you'll redirect to Cloudwatch dashboard, there select Create a Billing alert option. Automatically Create Alarm windows will open.



5. In this windows, enter the USD value, when you want to receive the notifications and enter your email id which you want to get the notifications, Click on **"Create Alarm"** When your monthly usage reaches to 5$ you'll get notified by the cloudwatch service through the mentioned email.

6. AWS does not allow the billing alarm's period to be set less than 6 hours. Here is how exactly billing alarm looks like.



ALARM Threshold details:

With the Alarm's threshold set, the final thing that you need to do is define what action the alarm must take when it is triggered. From the Notification section, fill out the required details, as mentioned in the following:

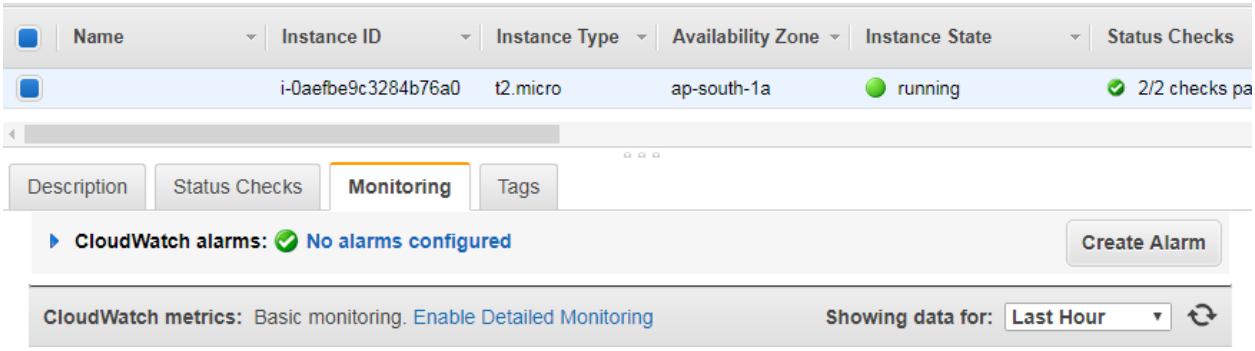
Whenever this alarm: This option will allow you to determine when the alarm will actually perform an action. There are three states of an alarm out of which you can select any one at a single time:

State is ALARM: Triggered when the metric data breaches the threshold value set by you
State is OK: Triggered when the metric data is well within the supplied threshold value
State is INSUFFICIENT: Triggered when the alarm generally doesn't have enough data with itself to accurately determine the alarm's state.

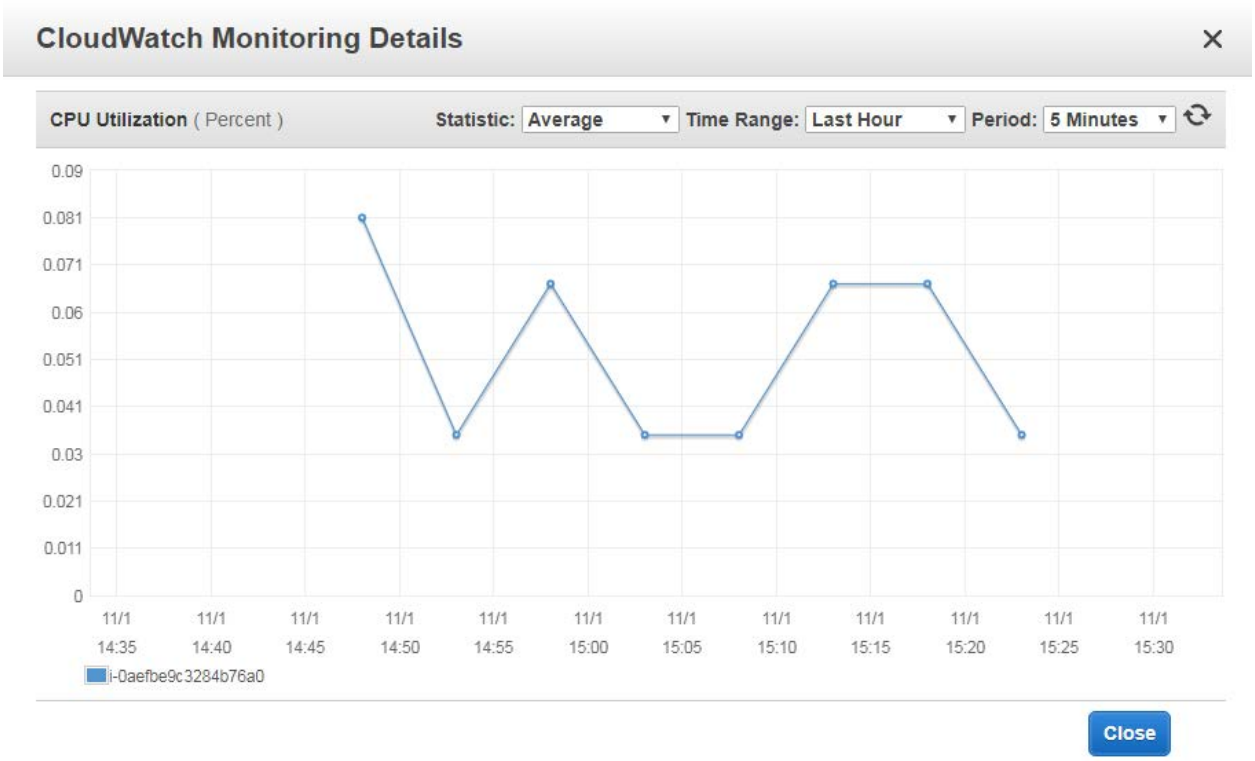
Monitoring your instance's CPU Utilization using CloudWatch

We are going to creating a simple alarm to monitor an instance's CPU utilization. If the CPU utilization breaches a certain threshold, say 75 percent, then the alarm will trigger an email notification as well as perform an additional task such as stop/restart the instance.

AWS makes creating alarms a really simple and straightforward process. The easiest way to do this is by selecting **your individual instances** from the **EC2 Management Dashboard** and selecting the **Monitoring tab**. Each instance is monitored on a five-minute interval by default. We can modify this behavior and set the time interval as low as one minute by selecting the Enable Detailed Monitoring option.

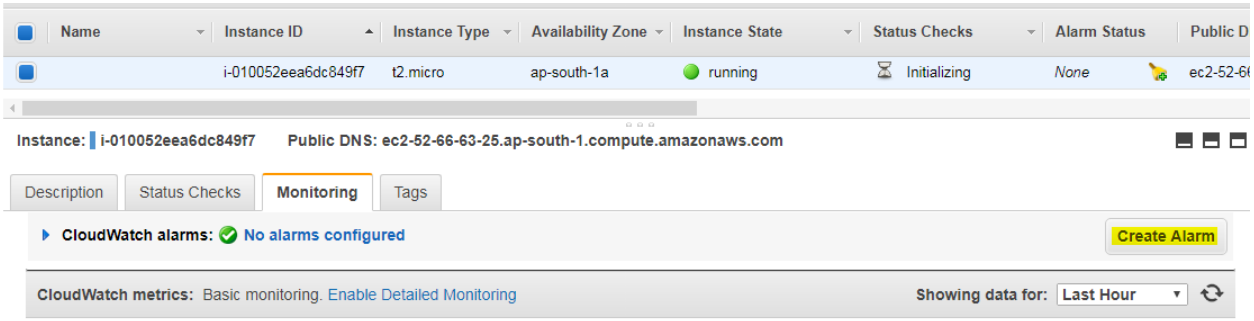


Each instance Monitoring graphs display important metric information such as CPU utilization, disk Read/Writes, bytes transferred in terms of network IO. We can expand on each of the graphs by simply selecting them.

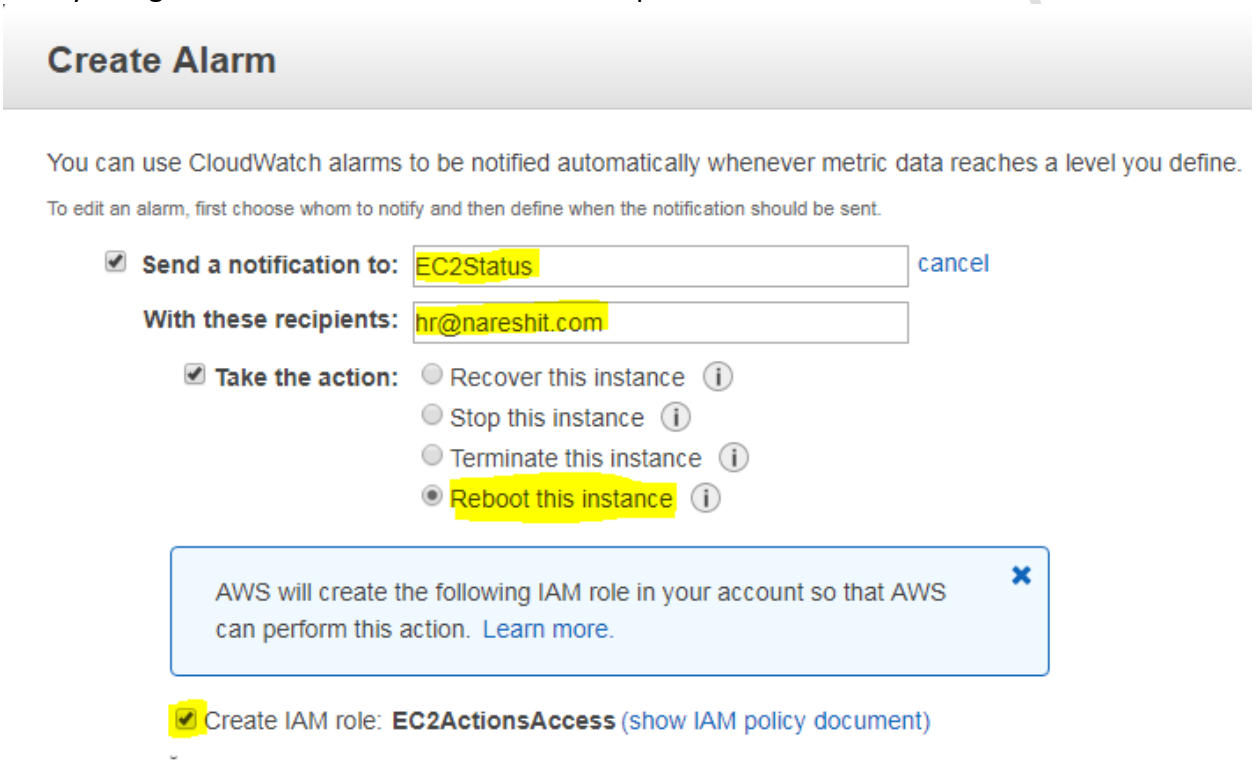


The x axis displays the CPU utilization in percent whereas the y axis display the time as per the current period's settings. We can view the individual data points and their associated values by simply hovering over them on the graph. Alternatively, you can also switch between the Statistics, Time Range, and Period as per our requirements.

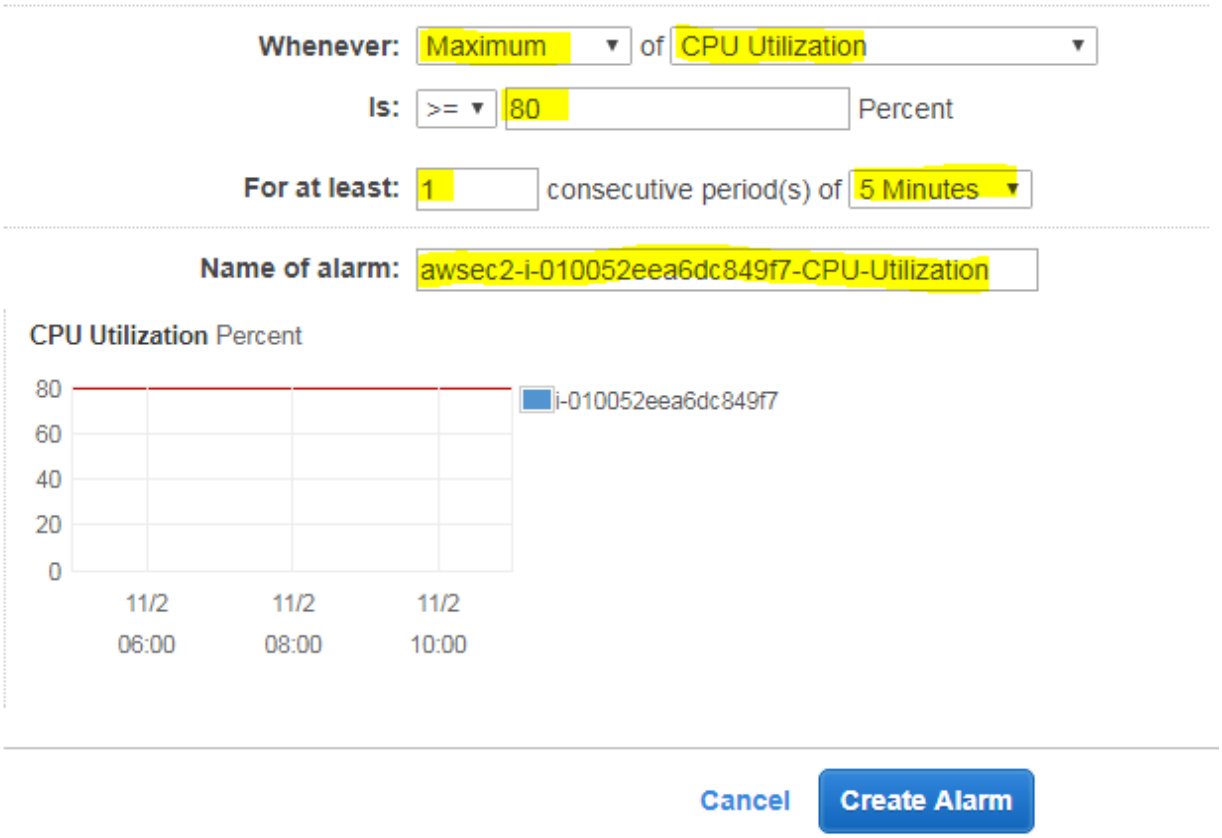
1. Once you have viewed your instance's performances, you can create a simple alarm by selecting the Create Alarm option provided in the Monitoring tab.
2. Click on **Create Alarm** option as shown below image.



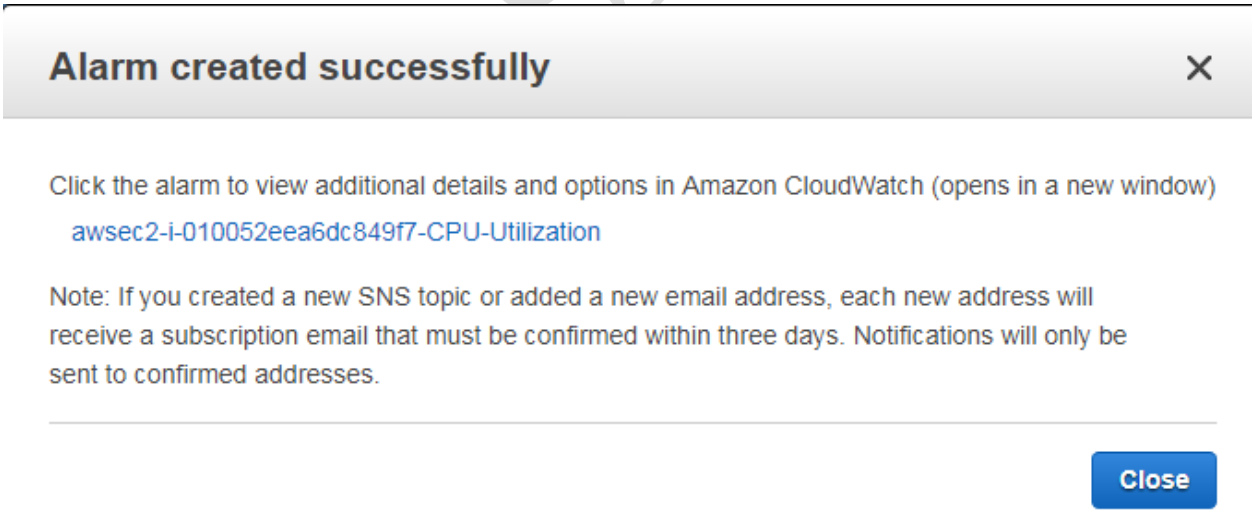
3. Now you'll get a windows with all the available options to create an alarm.



- If you want to get the notifications to an email ID, we need to depend on another service called SNS, click on **"Create topic on Send notifications to"** Then give a name for the topic. Enter a valid email to get the notifications in "**With these recipients field"**.
- Select the Take the action, what action you want to perform on instance, when the alarm matches with the defined threshold. In this case am selecting **Reboot this instance** option. (Criteria am mentioning is when CPU utilization >80 % for consecutive of 5 minutes).
- To perform this action, we have to create a **role**, If we have any existing role, we can attach it, otherwise select the option **"Create IAM role".**



- Here am defining the thresholds about the alarm, Whenever **Maximum of CPU Utilization** is**>= 80** Percent for at least **1** consecutive period of **5 Minutes.**
- Then allocating a name for this Alarm.



- Alarm created successfully, we can verify the same from.
- We have 1,377 Metrics till date. We can use any of the one.

Dashboard: Dashboard is a centralized place to monitor all your resources. Free Tier

- New and existing customers also receive 3 dashboards of up to 50 metrics each per month at no additional charge. ($3.00 per dashboard per month after that)

- Basic Monitoring metrics (at five-minute frequency) for Amazon EC2 instances are free of charge, as are all metrics for Amazon EBS volumes, Elastic Load Balancers, and Amazon RDS DB instances.
- New and existing customers also receive 10 metrics, 10 alarms and 1 million API requests each month at no additional charge.

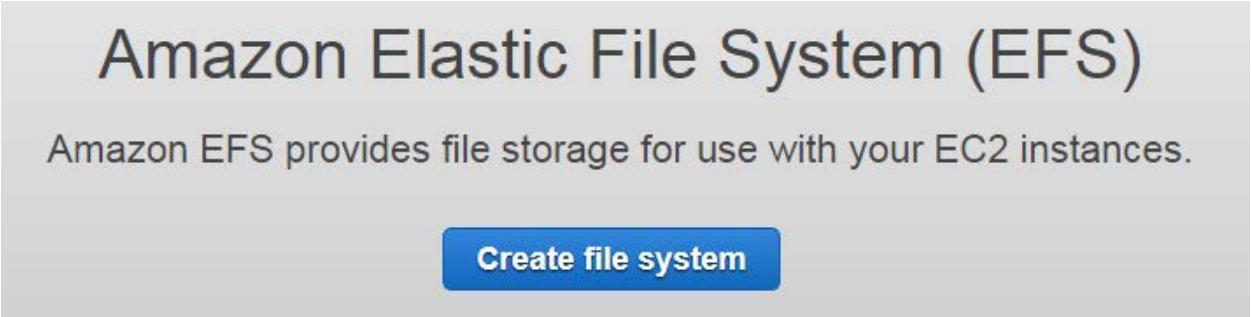
ELASTIC FILE SYSTEM (EFS)

- Amazon EFS is easy to use and offers a simple interface that allows you to create and configure file systems quickly and easily. With Amazon EFS, storage capacity is elastic, growing and shrinking automatically as you add and remove files.
- Supports the Network File System version 4 (NFSv4.1) protocol.
- Multiple Amazon EC2 instances can access an Amazon EFS file system, so applications that scale beyond a single instance can access a file system.
- Amazon EC2 instances running in multiple Availability Zones (AZs) within the same region can access the file system, so that many users can access and share a common data source.
- It is also based on the pay-per-use model, which means that you only have to pay for the storage used by your filesystem
- Using Amazon EFS with Microsoft Windows Amazon EC2 instances is not supported.
- Multiple Amazon EC2 instances can access an Amazon EFS file system at the same time, allowing Amazon EFS to provide a common data source for workloads and applications running on more than one Amazon EC2 instance.
- You can mount your Amazon EFS file systems on your on-premises datacenter servers when connected to your Amazon VPC with AWS Direct Connect.

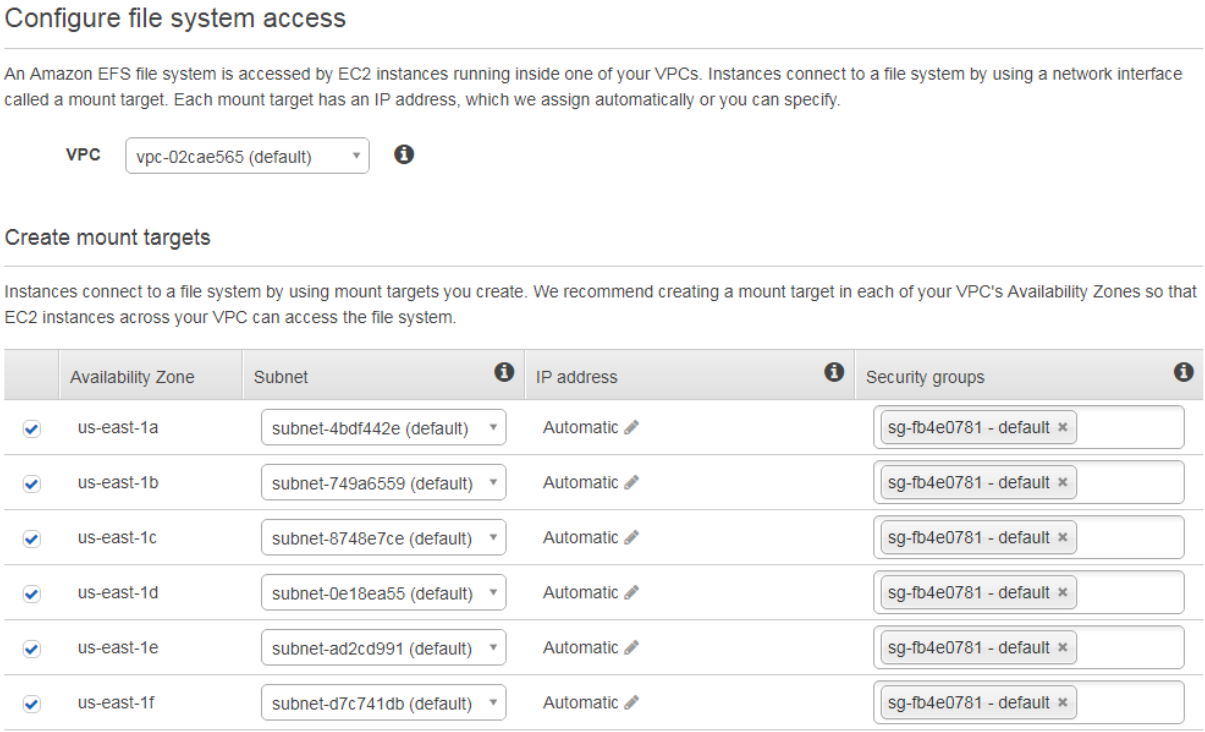
Steps to Create EFS:

1. We can find the EFS under storage category.
2. EFS is not available in all the regions as of now. Here is the supported regions. Switch to the region where you wish to create.

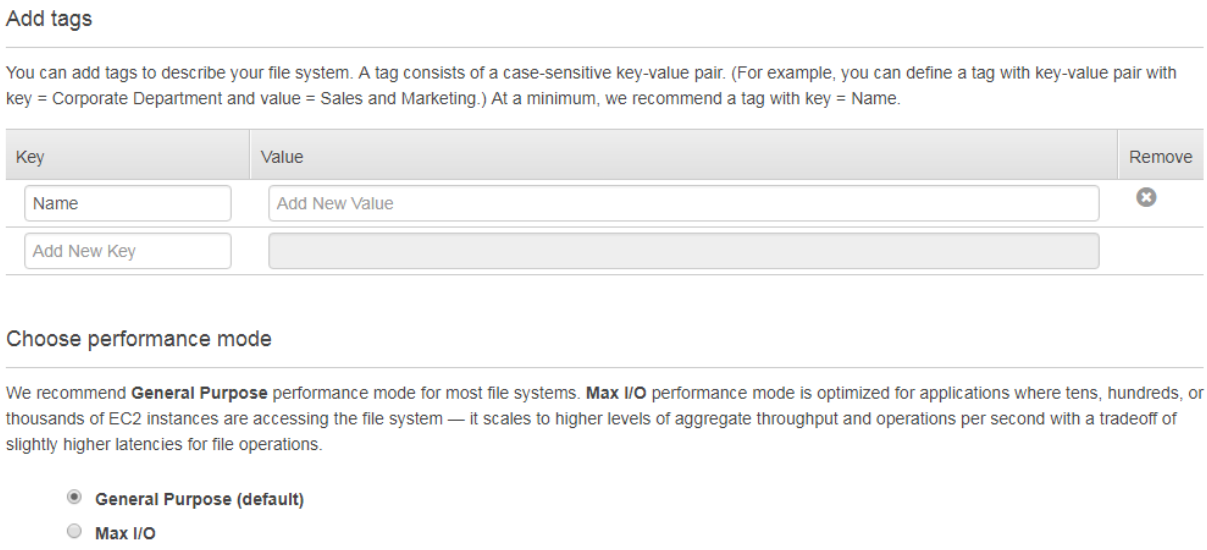
3. So, I switched to N. Virginia to perform the lab and selected EFS and select **Create file system** option.



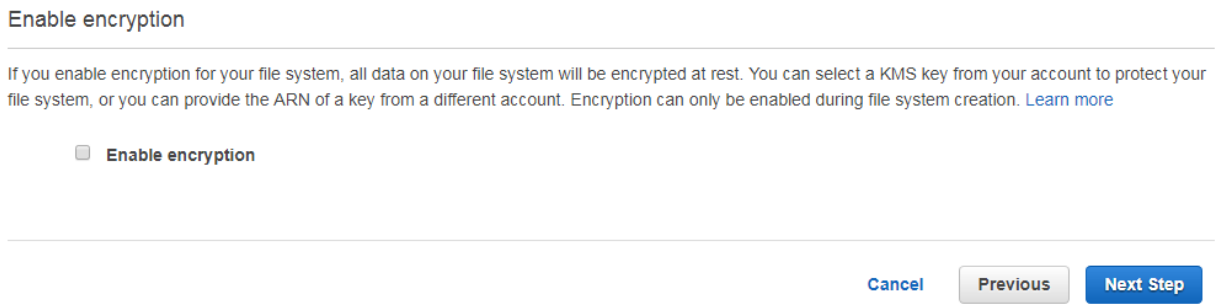
4. Select your VPC and Subnets, if you don't want to make this file system available to any specific subnet, Just untick that here. Then select **Next.**



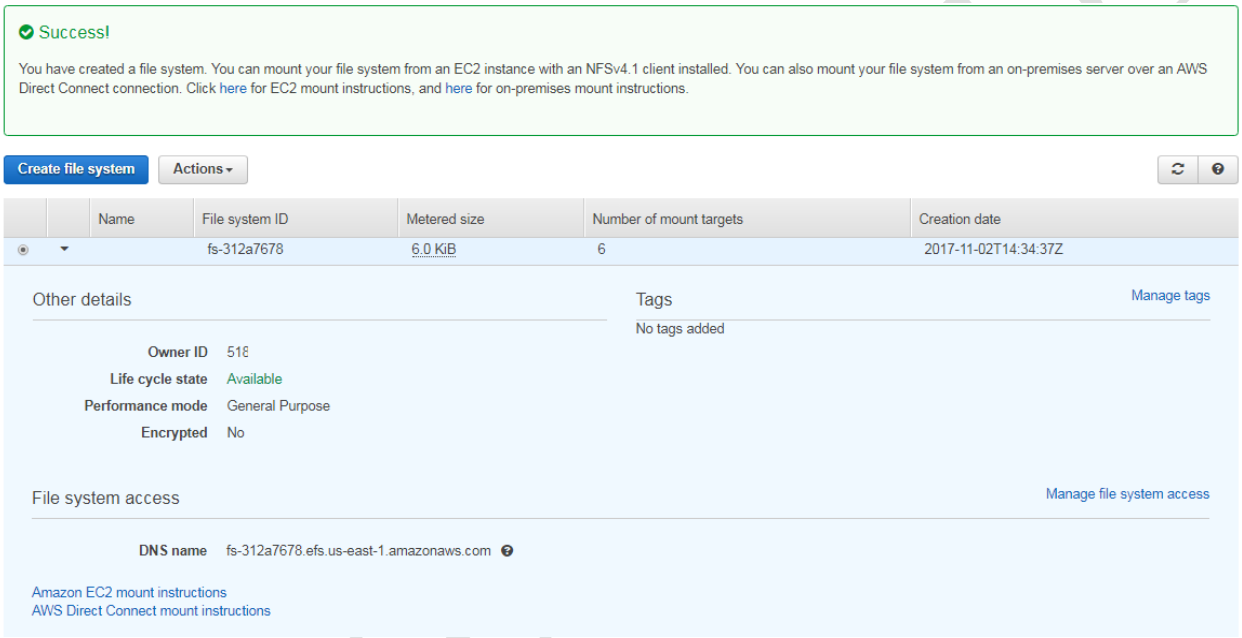
5. If we want to add tags, we can add here and we need to select the Performance Mode. We have to select this based on EC2 instance count.



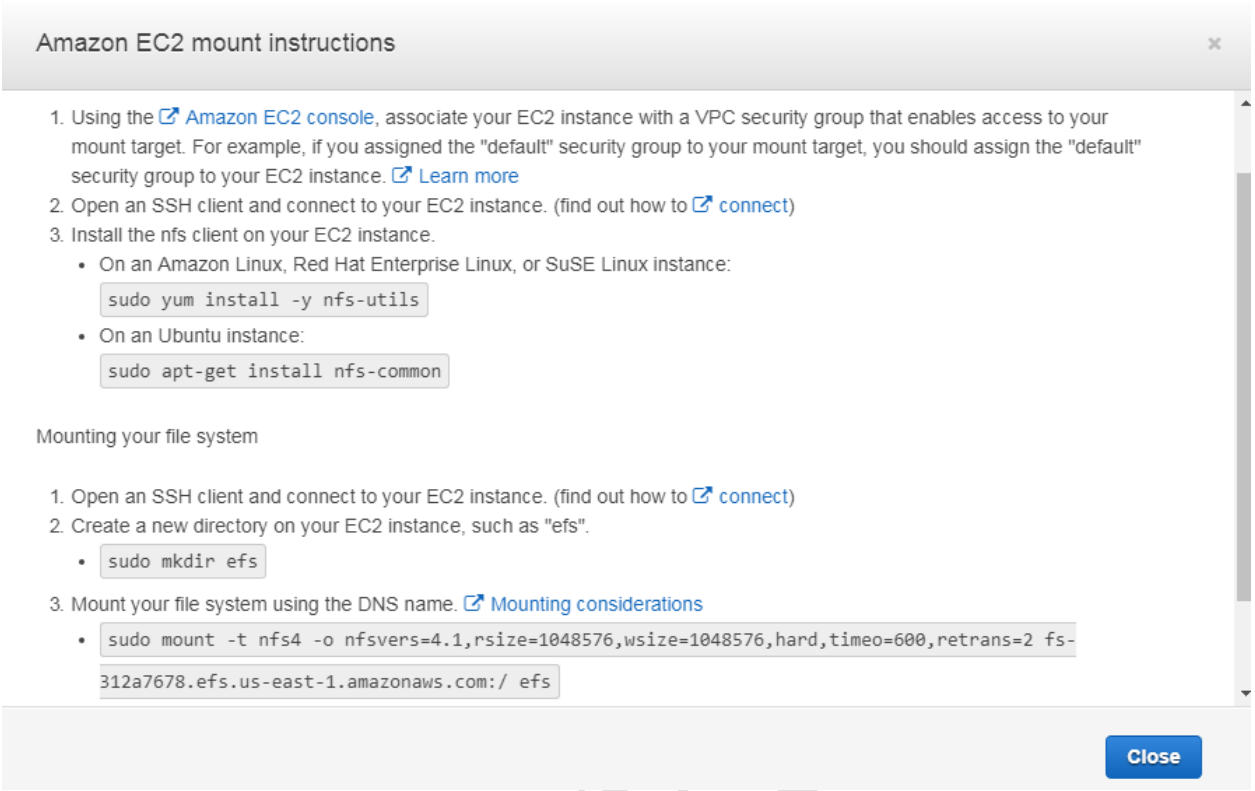
6. If we want to encrypt the data storing under EFS, we can enable the option on same page, then click on **NEXT.**



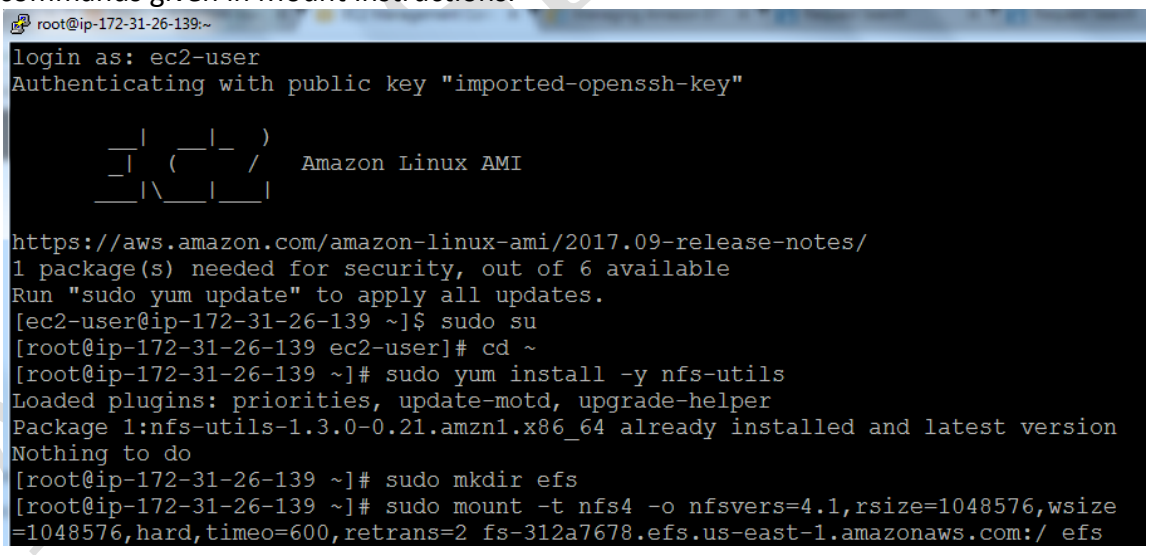
7. Review all the options and select Create File System option, file system will be created now and available for usage.



8. Now we have to mount it to EC2 instances, for mounting we need to login to Instance and need to follow mounting instructions. To get the Instructions select the **Amazon EC2 mount instructions** option.

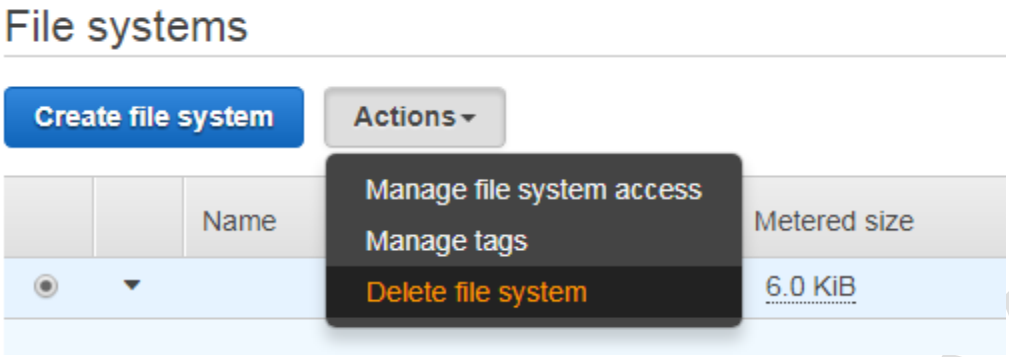


9. You can run the following commands on your EC2 instance.
10. **Your instance must be member of the Default Security group for successful EFS mounting.**
11. Here am launching Linux EC2 instance, as windows not supportable and executing the commands given in Mount Instructions.

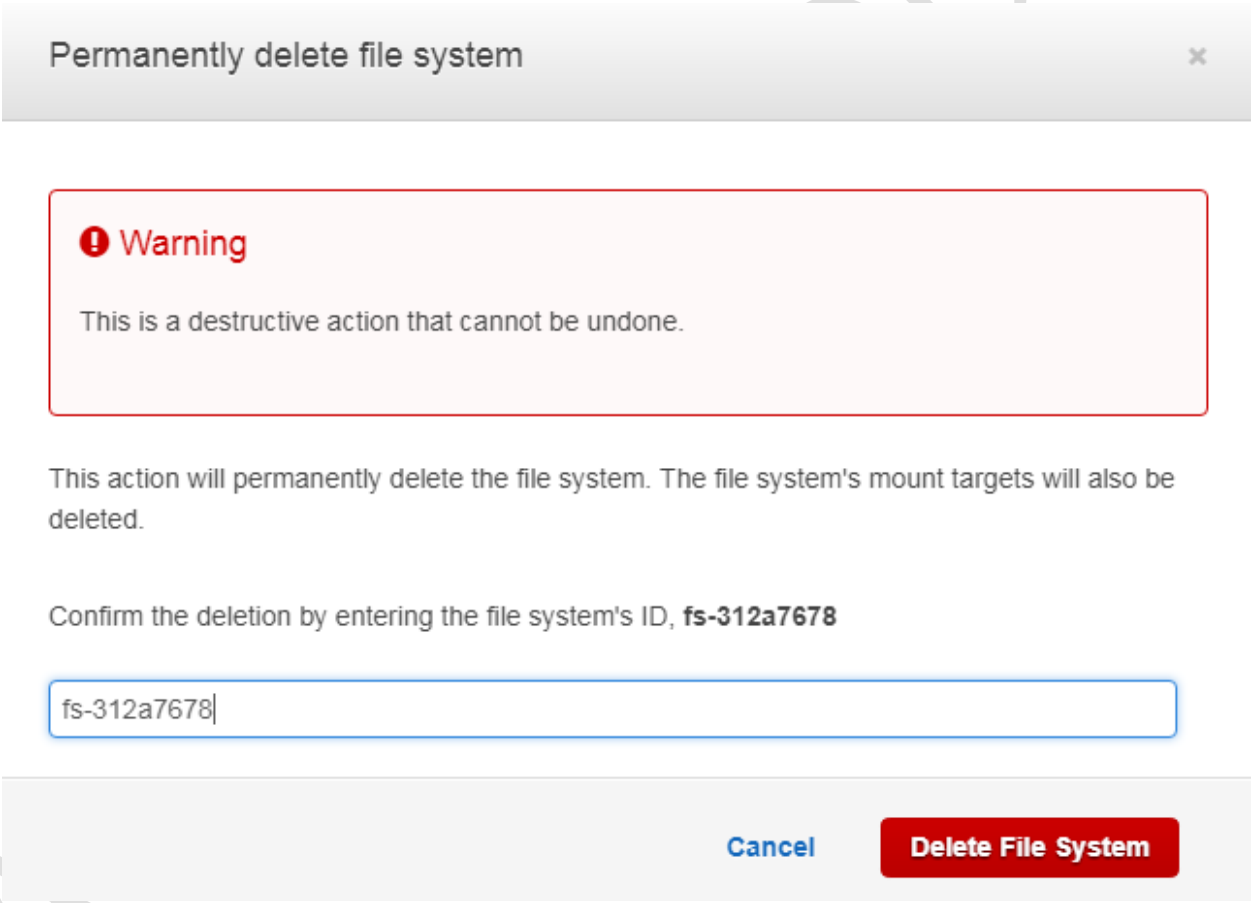


12. In above image, I've elevated my privileges to root and tried to install the required **nfs-utils,** but It'll installed by default in Amazon Linux Instances.
    - Created a directory named efs with "sudomkdirefs" command.
    - And executed the mounting command to the created directory, now whatever the files I created under "efs" is going to available for all EC2 instances.
    - If you want to test this, perform the same steps in another EC2 instance and test it.

13. If you want to delete the EFS, Select the EFS and go to "**Actions"** and **"Delete File System".**



14. Enter the file system's ID in the box and select the "Delete File System" button, File system will delete now.



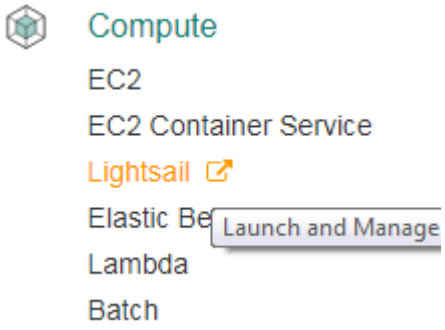
LIGHTSAIL

With Amazon Lightsail with a couple of clicks we can choose a configuration from a menu and launch a virtual machine preconfigured with SSD-based storage, DNS management, and a static IP address.

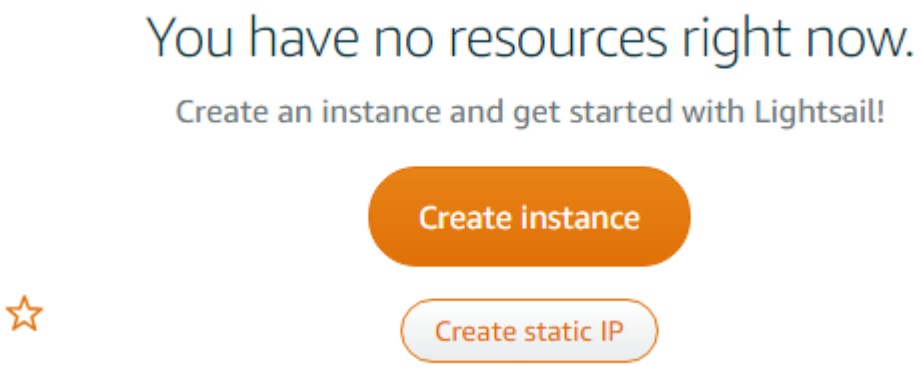
We can launch it on Amazon Linux AMI or Ubuntu operating system, developer stack (LAMP, LEMP, MEAN, or Node.js), or application (Drupal, Joomla, Redmine, GitLab, and many others), with flat-rate pricing plans that start at $5 per month including a generous allowance for data transfer.

Steps to launch Lightsail Instance

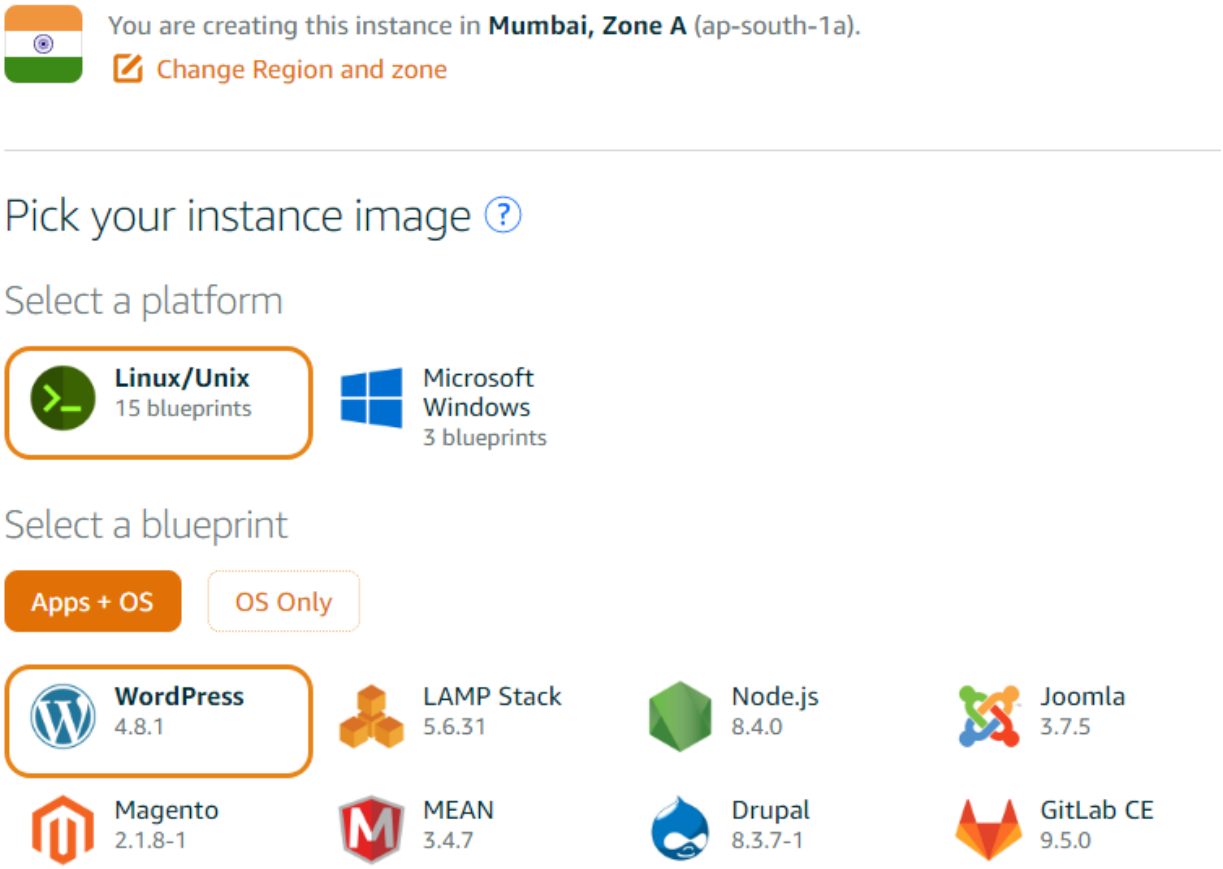
1. Select the **Lightsail** from Compute Service.



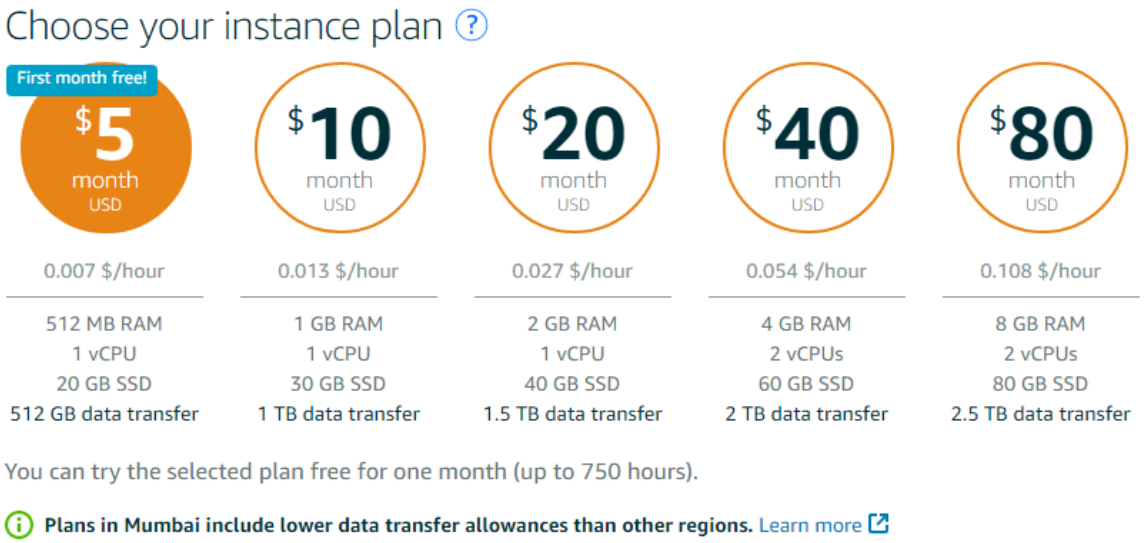
2. Select the **Create instance** option.



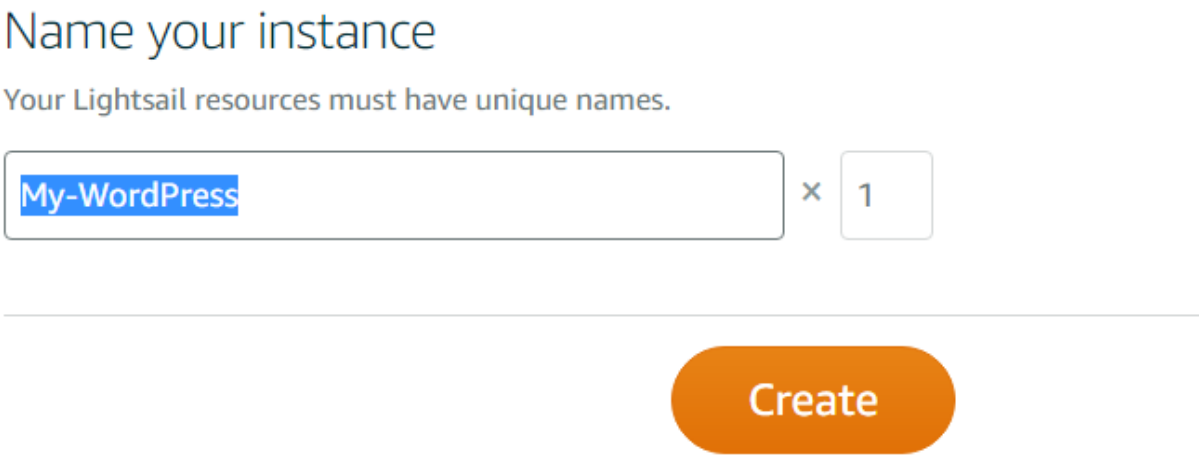
3. Select the Region and Zone, then select the Platform, and a blueprint what instance what application we required. Now am going to launch **Wordpress** website.



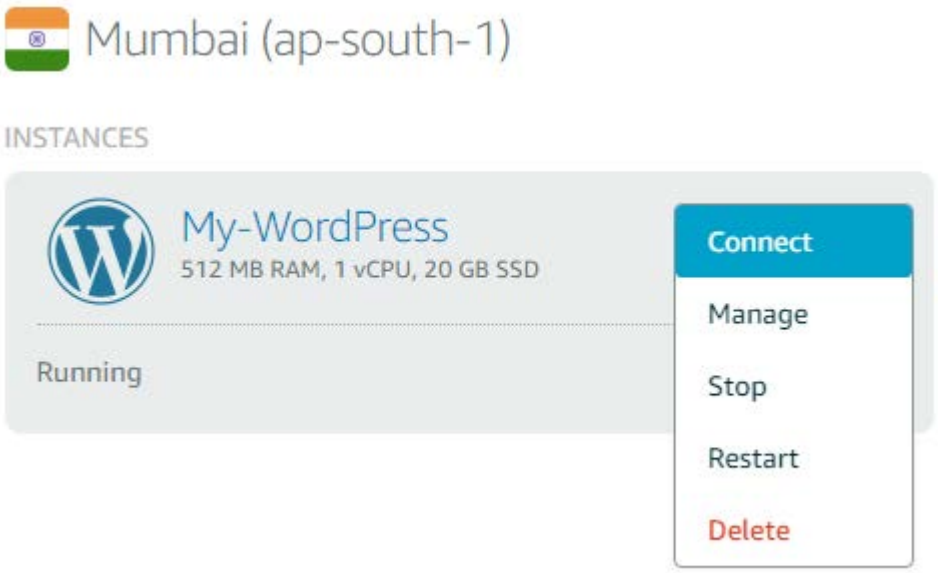
3. Then choose instance plan, am selecting $5/Month.



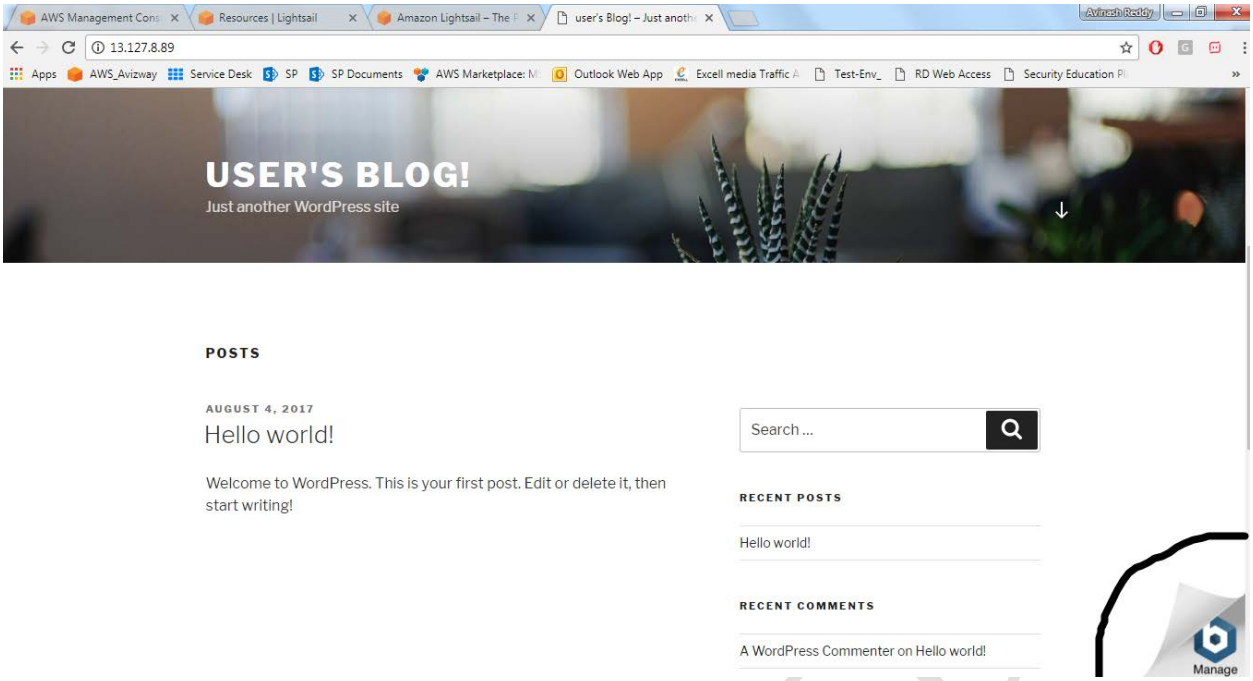
4. And give a name for your instance and select **Create** option.



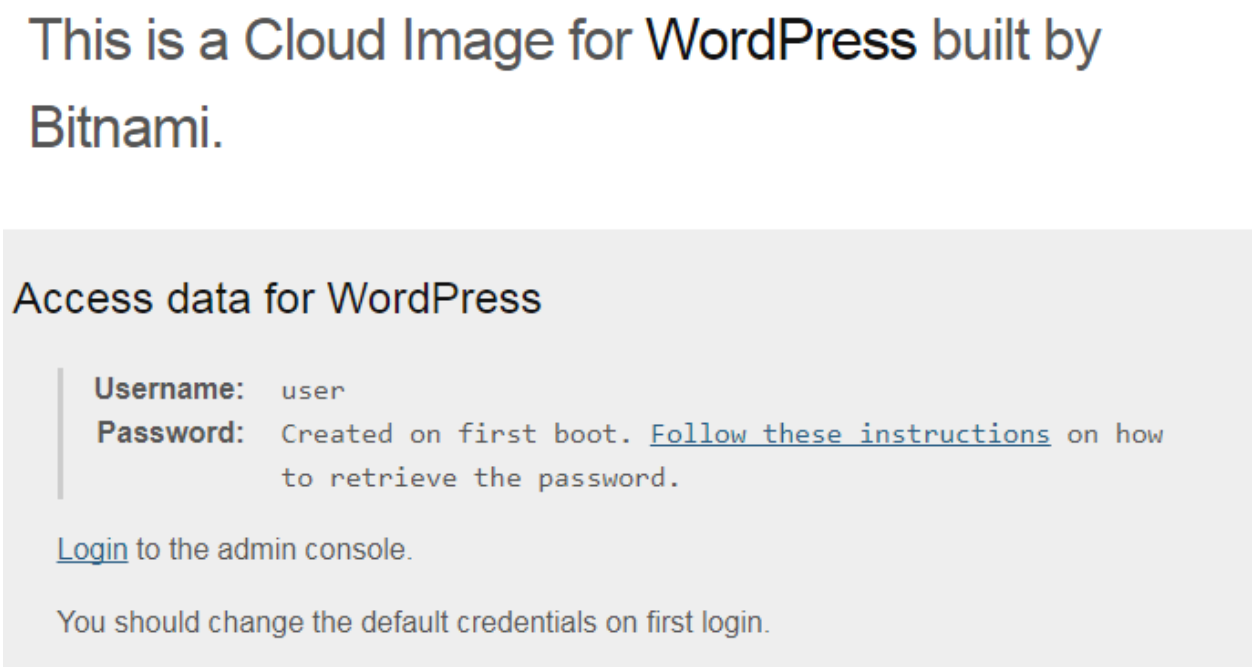
5. When the instance is ready select the connect option and you'll get a console.



6. We'll get a public IP address by using that Public IP, we can access the WP website.
7. We will get a default template, if you want to customize that we have to login to the Admin panel. Here I've entered public IP the browser. In bottom corner, We will get Manage button, select that to login.



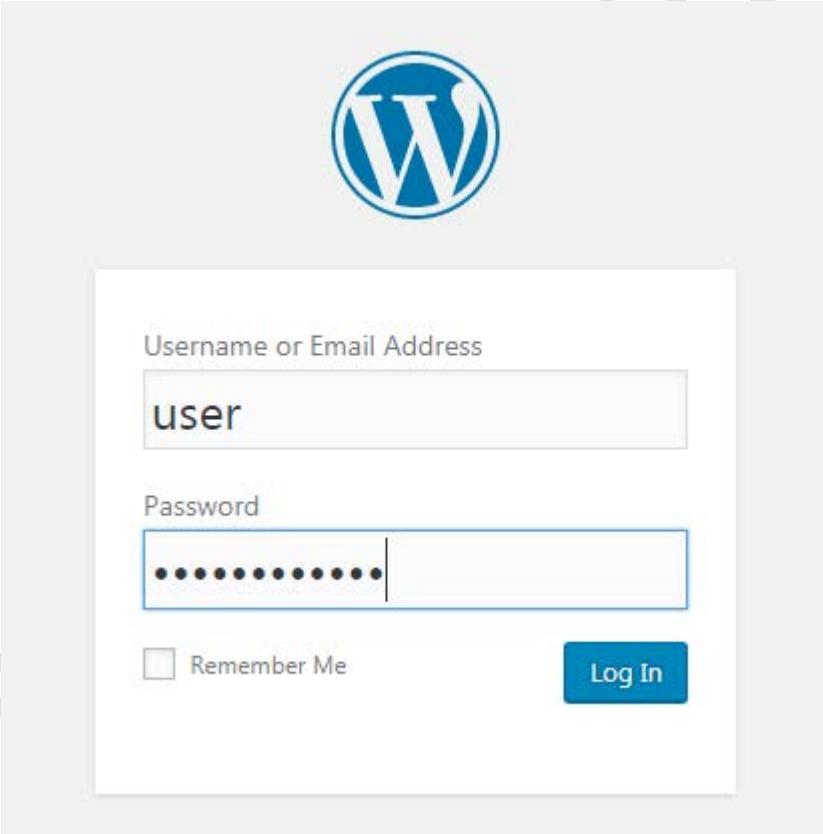
8. Default username is **user** and to get the password am connecting to the instance and entering command as below image. Select on **Login** option.



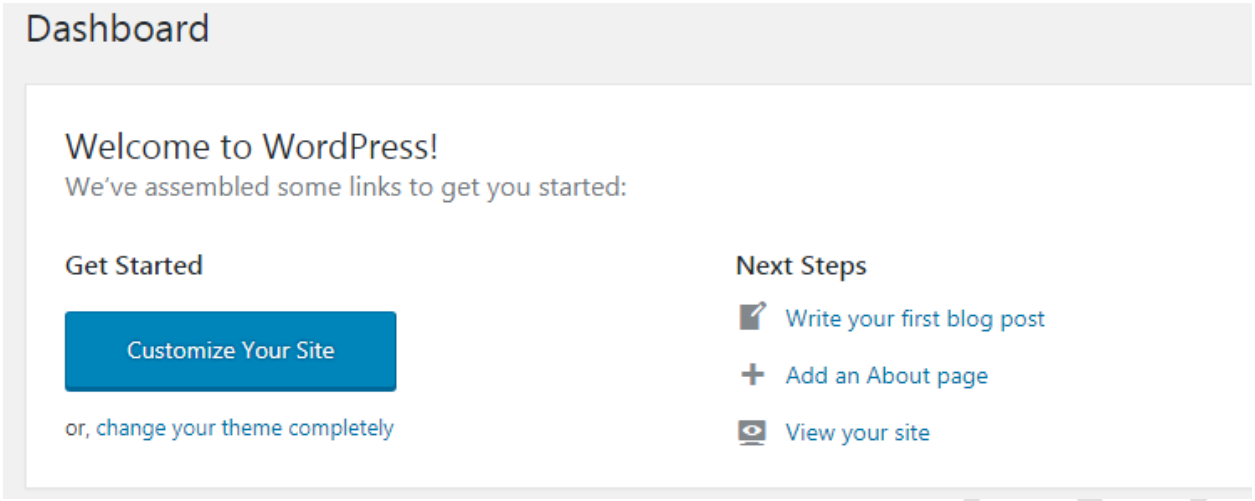
9. After connecting the instance give ls command you'll find bitname_application_password file, open it with cat command you'll get password to login, note it and enter in the login page.

```
Welcome to Ubuntu 14.04.5 LTS (GNU/Linux 3.13.0-125-generic x86_64)
 ___ _ _                   _
| _ |_) |_ _ _  __ _ _ __ (_)
| _ \ |  _| ' \/ _` | '  \| |
|___/_|\__|_|_|\__,_|_|_|_|_|

  *** Welcome to the Bitnami WordPress 4.8.1-0 ***
  *** Documentation:  https://docs.bitnami.com/aws/apps/wordpress/ ***
  ***                 https://docs.bitnami.com/aws/ ***
  *** Bitnami Forums: https://community.bitnami.com/ ***
bitnami@ip-172-26-10-72:~$ ls
apps  bitnami_application_password  htdocs  stack
bitnami@ip-172-26-10-72:~$ cat bitnami_application_password
X9U6ur9pKU5v
bitnami@ip-172-26-10-72:~$
```

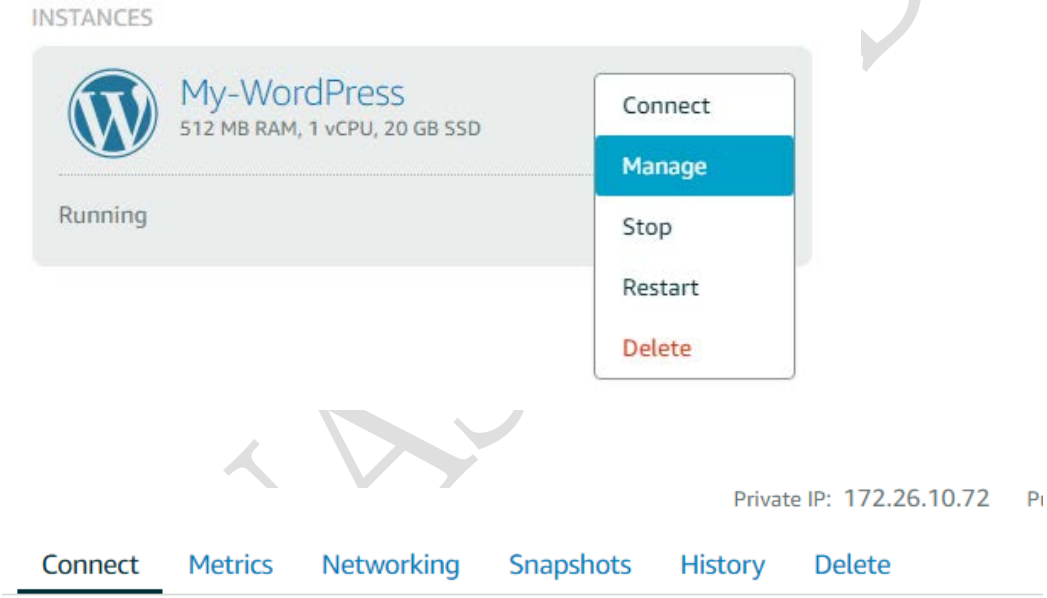
10. Give the username and password in the listed fields.



11. After authenticating, we'll login to the WP website and we can start customizing the website and select the Publish then the changes will update immediately.



12. If you want to manage your instance you can select the Manage option and you'll get the options to view the Metrics, Networking, Snapshots for backup, History and Delete options.



13. You can delete it anytime, by Delete option.

Elastic Beanstalk

With Elastic Beanstalk, we can deploy, monitor, and scale an application quickly and easily.

AWS Elastic Beanstalk is an orchestration service offered from Amazon Web Services for deploying infrastructure which orchestrates various AWS services, including EC2, S3, Simple Notification Service (SNS), CloudWatch, autoscaling, and Elastic Load Balancers.

AWS Elastic Beanstalk supports the following languages and development stacks:

- Apache Tomcat for Java applications
- Apache HTTP Server for PHP applications
- Apache HTTP Server for Python applications
- Nginx or Apache HTTP Server for Node.js applications
- Passenger or Puma for Ruby applications
- Microsoft IIS 7.5, 8.0, and 8.5 for .NET applications
- Java SE
- Docker
- Go

Application Deployment requires a number of components to be defined as follows

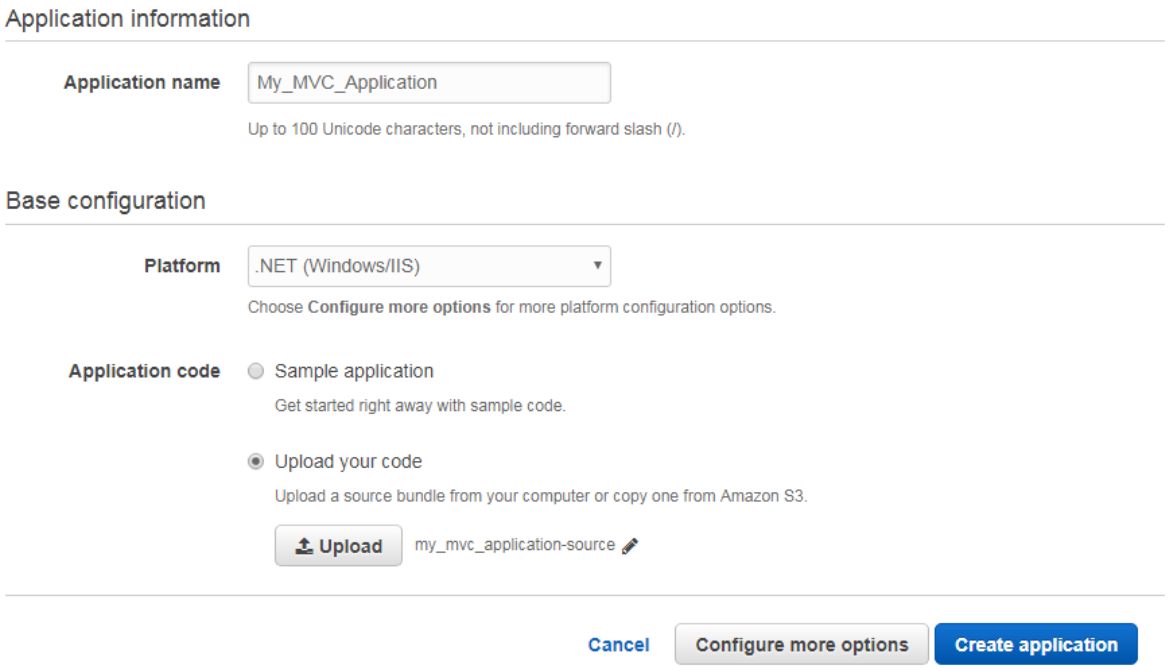
Application: as a logical container for the project.

Version: which is a deployable build of the application executable.

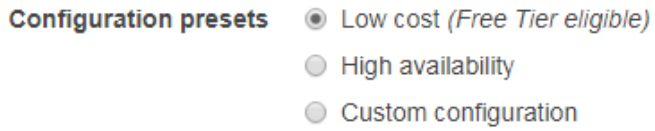
Configuration template: This contains configuration information for both the Beanstalk environment and for the product.

Environment: combines a 'version' with a 'configuration' and deploys them.

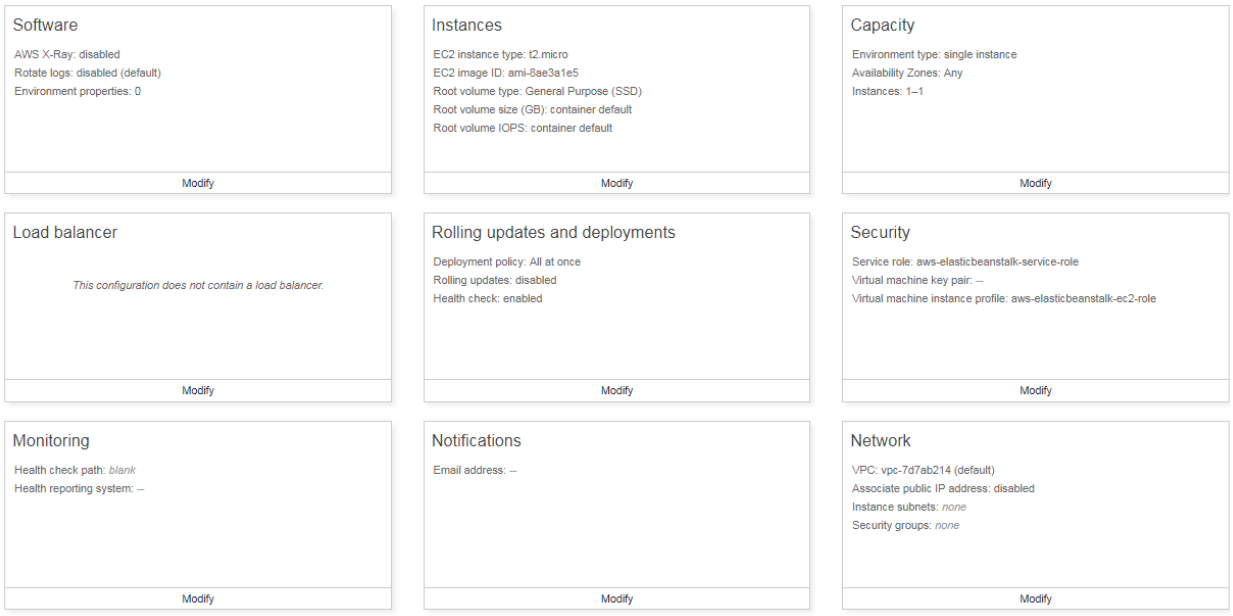
1. Create a Web Application. It involves with multiple options. By creating an environment, we allow AWS Elastic Beanstalk to manage AWS resources and permissions on behalf of us.



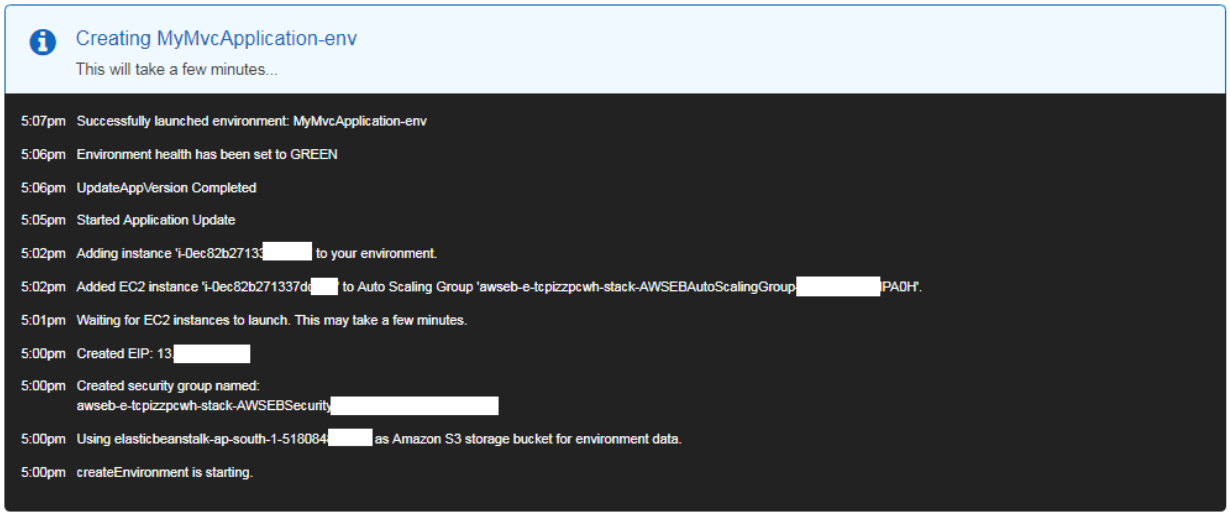
2. You can simply select the Create application option to perform the deployment and selecting the appropriate configuration for our instances.
3. If you want to customize each and every step, as you required, Select **Configure more options** option.
    - Then we'll get three options for **Configuration presets**
        i. **Low Cost (Free Tier eligible)**
        ii. **High Availability**
        iii. **Custom Configuration**



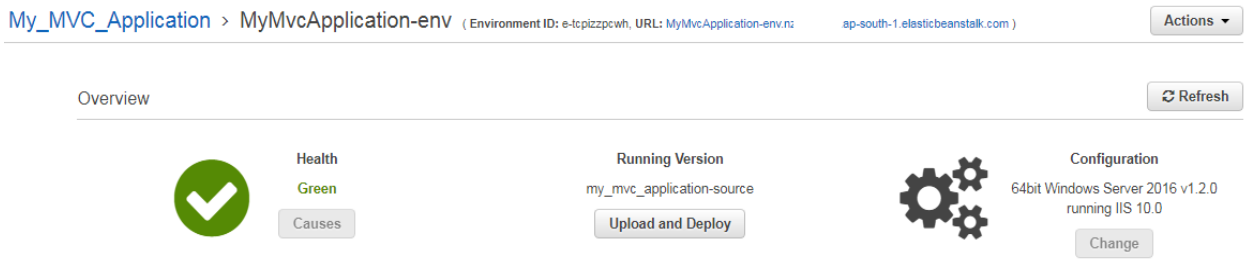
4. If we want to change the Platform of Windows server or IIS, we can select change platform configuration option otherwise go with the default option.
5. Select the appropriate option, here am selecting the Low Cost, Free Tier eligible.
6. Here is the available options to customize



7. Status of Instance creation, and all the required resources are provisioning by Elastic BS i.e; Security group, EIP, EC2, S3, Simple Notification Service (SNS), CloudWatch, autoscaling, and Elastic Load Balancers.

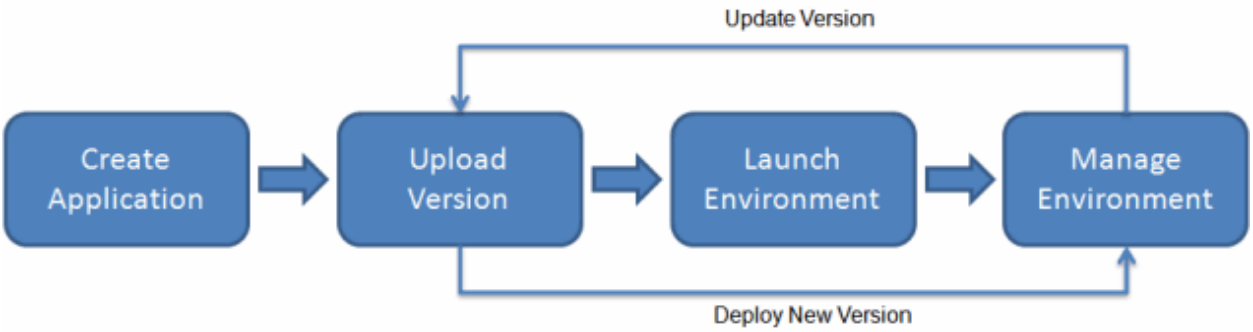


8. Here is the status we'll get when the application is deployed.

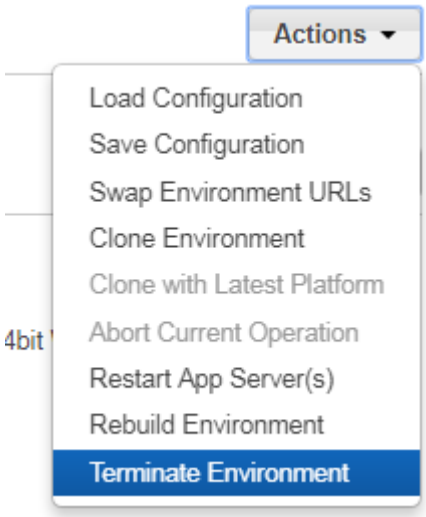


9. We'll get Environment ID to access the application.
10. Here is the output for my uploaded code.

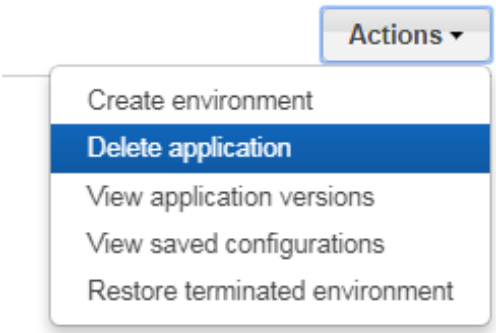
11. If you made any changes to your existing code, you can zip it and upload it.
12. Here is the illustration diagram of workflow



13. If you want to terminate the environment, select the **Actions** option in Top right corner, then choose Terminate Environment.



14. Or go back to the applications page and delete the application.



Route 53

Domain Name System (DNS) and Amazon Route 53

- Domain Name Servers (DNS) are the Internet's equivalent of a phone book. They maintain a directory of domain names and translate them to Internet Protocol (IP) addresses.
- This is necessary because, although domain names are easy for people to remember, computers or machines, access websites based on IP addresses.
- When you type in a web address, e.g., Avinash.website, your Internet Service Provider views the DNS associated with the domain name, translates it into a machine friendly IP address (202.153.xx.xx) and directs your Internet connection to the correct website.
- Amazon Route 53 is an authoritative DNS system. An authoritative DNS system provides an update mechanism that developers use to manage their public DNS names.
- It answers DNS queries, translating domain names into IP addresses so that computers can communicate with each other.

Top-Level Domains (TLDs)

A Top-Level Domain (TLD) is the most general part of the domain. The TLD is the farthestportion to the right (as separated by a dot). Common TLDs are .com, .net, .org, .gov, .edu,and .io.

- The last word in a domain name represents the "top level domain".
- The second word in a domain name is known as a second level domain name.
- These top level domain names are controlled by the Internet Assigned Numbers Authority (IANA) in a root zone database which is essentially a database of all available top level domains.
- You can view this database by visiting http://www.iana.org/domains/root/db
- Each domain name becomes registered in a central database, known as the WhoIS database.

Domain Names

A domain name is the human-friendly name that we are used to associating with an Internet resource.

The URL aws.amazon.com is associated with the servers owned by AWS. The DNS allows usersto reach the AWS servers when they type aws.amazon.com into their browsers.IP Addresses an IP address is a network addressable location. Each IP address must be unique within itsnetwork. For public websites, this network is the entire Internet.

- IPv4 addresses, the most common form of addresses, consist of four sets of numbers separated by a dot, with each set having up to three digits.
  For example, 111.222.111.222 could be a valid IPv4 IP address.

- With DNS, we map a name to that address so that you do not have to remember a complicated set of numbers for each place you want to visit on a network.
- Due to the tremendous growth of the Internet and the number of devices connected to it, the IPv4 address range has quickly been depleted.
- Today, most devices and networks still communicate using IPv4, but migration to IPv6 is proceeding gradually over time.

Domain Name Registrars

All of the names in a given domain must be unique, there needs to be a way toorganize them so that domain names aren't duplicated. This is where domain name registrars come in.

A domain name registrar is an organization or commercial entity thatmanages the reservation of Internet domain names.

- A registrar is an authority that can assign domain names directly under one or more top-level domains.
- These domains are registered with ICANN (The Internet Corporation for Assigned Names and Numbers), which enforces uniqueness of domain names across the Internet.
- Each domain name becomes registered in a central database known as the WHOIS database.
- Domain registrars : GoDaddy.com, BigRock , Amazon etc

Domain Registration

If you want to create a website, you first need to register the domain name.

- If you already registered a domain name with another registrar, you have the option to transfer the domain registration to Amazon Route 53.
- It isn't required to use Amazon Route 53 as your DNS service or to configure health checking for your resources.
- Amazon Route 53 supports domain registration for a wide variety of generic TLDs (for example, .com and .org) and geographic TLDs (for example, .be and .us).

Name Servers

NS stands for Name Server records and are used by Top Level Domain servers to direct traffic to the Content DNS server which contains the authoritative DNS records.

A name server is a computer designated to translate domain names into IP addresses. Theseservers do most of the work in the DNS. Because the total number of domain translations istoo much for any one server, each server may redirect requests to other name servers ordelegate responsibility for the subset of subdomains for which they are responsible.

Name servers can be authoritative, meaning that they give answers to queries about domainsunder their control. Otherwise, they may point to other servers or serve cached copies ofother name servers' data.

Zone Files

A zone file is a simple text file that contains the mappings between domain names and IPaddresses. This is how a DNS server finally identifies which IP address should be contactedwhen a user requests a certain domain name.

Record Types:

Each zone file contains records. In its simplest form, a record is a single mapping between aresource and a name. These can map a domain name to an IP address or define resources forthe domain, such as name servers or mail servers. This section describes each record type indetail.

Start of Authority (SOA) Record

A Start of Authority (SOA) record is mandatory in all zone files, and it identifies the baseDNS information about the domain. Each zone contains a single SOA record.

The SOA record stores information about the following:

- The name of the DNS server for that zone
- The administrator of the zone
- The current version of the data file
- The number of seconds that a secondary name server should wait before checking for updates
- The number of seconds that a secondary name server should wait before retrying a failed zone transfer
- The maximum number of seconds that a secondary name server can use data before it must either be refreshed or expire
- The default TTL value (in seconds) for resource records in the zone

A and AAAA

Both types of address records map a host to an IP address. The A record is used to map a hostto an IPv4 IP address, while AAAA records are used to map a host to an IPv6 address.

Canonical Name (CNAME)

A Canonical Name (CNAME) record is a type of resource record in the DNS that defines analias for the CNAME for your server (the domain name defined in an A or AAAA record).

Mail Exchange (MX)

Mail Exchange (MX) records are used to define the mail servers used for a domain andensure that email messages are routed correctly. The MX record should point to a hostdefined by an A or AAAA record and not one defined by a CNAME.

Name Server (NS)

Name Server (NS) records are used by TLD servers to direct traffic to the DNS server that contains the authoritative DNS records.

Pointer (PTR)

A Pointer (PTR) record is essentially the reverse of an A record. PTR records map an IPaddress to a DNS name, and they are mainly used to check if the server name is associatedwith the IP address from where the connection was initiated.

Text (TXT)

Text (TXT) records are used to hold text information. This record provides the ability toassociate some arbitrary and unformatted text with a host or other name, such as humanreadable information about a server, network, data center, and other accounting information.

Service (SRV)

A Service (SRV) record is a specification of data in the DNS defining the location (the hostname and port number) of servers for specified services. The idea behind SRV is that, given a domain name (for example, example.com) and a service name (for example, web [HTTP],which runs on a protocol [TCP]), a DNS query may be issued to find the host name thatprovides such a service for the domain, which may or may not be within the domain.

Hosted Zones

A hosted zone is a collection of resource record sets hosted by Amazon Route 53. Like atraditional DNS zone file, a hosted zone represents resource record sets that are managedtogether under a single domain name. Each hosted zone has its own metadata andconfiguration information.

There are two types of hosted zones: private and public. A private hosted zone is a containerthat holds information about how you want to route traffic for a domain and its subdomainswithin one or more Amazon Virtual Private Clouds (Amazon VPCs). A public hosted zone is acontainer that holds information about how you want to route traffic on the Internet for adomain (for example, example.com) and its subdomains (for example, apex.example.com andacme.example.com).

- Use an alias record, not a CNAME, for your hosted zone. CNAMEs are not allowedfor hosted zones in Amazon Route 53.

Routing Policies:

SimpleRouting Policy

This is the default routing policy when you create a new resource. Use a simple routing policywhen you have a single resource that performs a given function for your domain (forexample, one web server that serves content for the example.com website). In this case,Amazon Route 53 responds to DNS queries based only on the values in the resource recordset (for example, the IP address in an A record).

WeightedRouting Policy

With weighted DNS, you can associate multiple resources (such as Amazon Elastic ComputeCloud [Amazon EC2] instances or Elastic Load Balancing load balancers) with a single DNSname.

Use the weighted routing policy when you have multiple resources that perform the samefunction (such as web servers that serve the same website), and you want Amazon Route 53to route traffic to those resources in proportions that you specify. For example, you may usethis for load balancing between different AWS regions or to test new versions of your website(you can send 10 percent of traffic to the test environment and 90 percent of traffic to theolder version of your website).

To create a group of weighted resource record sets, you need to create two or more resourcerecord sets that have the same DNS name and type. You then assign each resource record seta unique identifier and a relative weight.

Latency-BasedRouting Policy

Latency-based routing allows you to route your traffic based on the lowest network latencyfor your end user (for example, using the AWS region that will give them the fastest responsetime).

Use the latency routing policy when you have resources that perform the same function inmultiple AWS Availability Zones or regions and you want Amazon Route 53 to respond toDNS queries using the resources that provide the best latency.

FailoverRouting Policy

Use a failover routing policy to configure active-passive failover, in which one resource takesall the traffic when it's available and the other resource takes all the traffic when the firstresource isn't available. Note that you can't create failover resource record sets for privatehosted zones.

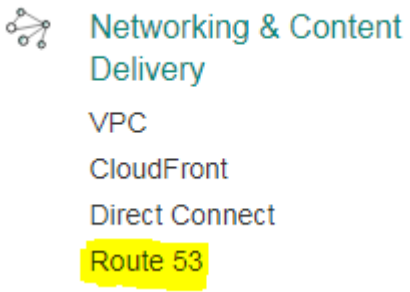
For example, you might want your primary resource record set to be in U.S. West (N.California) and your secondary, Disaster Recovery (DR), resource(s) to be in U.S. East (N.Virginia). Amazon Route 53 will monitor the health of your primary resource endpoints usinga health check.

A health check tells Amazon Route 53 how to send requests to the endpoint whose health youwant to check: which protocol to use (HTTP, HTTPS, or TCP), which IP address and port touse, and, for HTTP/HTTPS health checks, a domain name and path.
After you have configured a health check, Amazon will monitor the health of your selectedDNS endpoint. If your health check fails, then failover routing policies will be applied andyour DNS will fail over to your DR site.

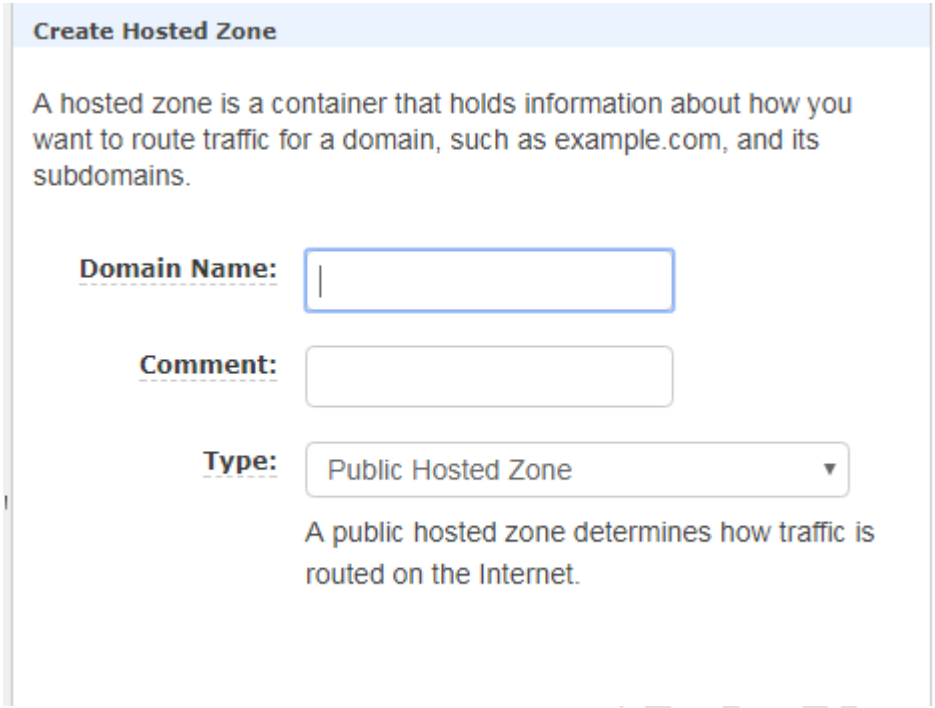
Geolocation
Geolocation routing lets you choose where Amazon Route 53 will send your traffic based onthe geographic location of your users (the location from which DNS queries originate). Forexample, you might want all queries from Europe to be routed to a fleet of Amazon EC2instances that are specifically configured for your European customers, with local languagesand pricing in Euros.
You can also use geolocation routing to restrict distribution of content to only the locations inwhich you have distribution rights. Another possible use is for balancing load acrossendpoints in a predictable, easy-to-manage way so that each user location is consistentlyrouted to the same endpoint.
You can specify geographic locations by continent, by country, or even by state in the UnitedStates. You can also create separate resource record sets for overlapping geographic regions,and priority goes to the smallest geographic region. For example, you might have oneresource record set for Europe and one for the United Kingdom. This allows you to routesome queries for selected countries (in this example, the United Kingdom) to one resourceand to route queries for the rest of the continent (in this example, Europe) to a differentresource.

Steps to Create a Hosted Zone.

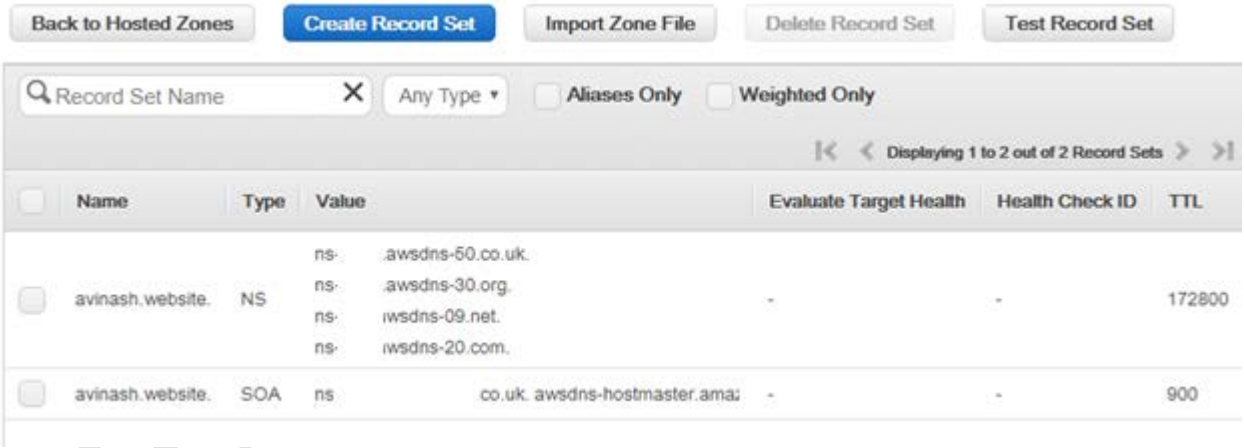
1. Log in to the AWS Management Console, Navigate to Amazon “Route 53” under “Network & Content Delivery”.



2. Create a Hosted Zone by selecting Create Hosted Zone” and Give the Purchased Domain Name, enter thee comments and choose the Type. We have two types of Hosted Zone, Selecting the **Public Hosted Zone** now.
    1. **Public Hosted Zone:**A public hosted zone is acontainer that holds information about how you want to route traffic on the Internet for adomain and its subdomains.
    2. **Private Hosted Zone:**A private hosted zone is a containerthat holds information about how you want to route traffic for a domain and its subdomains within one or more VPC.



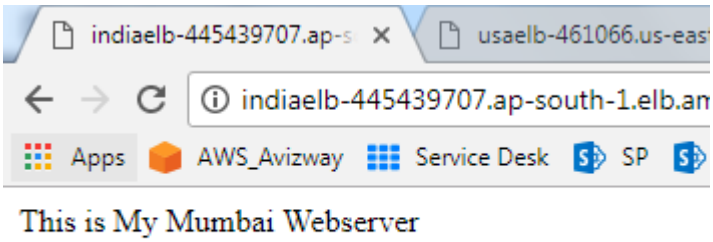
3. When you created a Hosted Zone, you'll get two record sets. Those are NS record and SOA record.



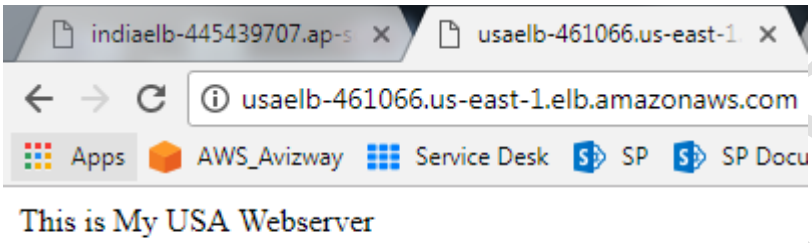
4. Now the Hosted Zone is created. If you purchase the Domain name from any other domain registrar i.e; Godaddy, bigrock we have to configure these NameServers in that account, or we can transfer the domain to AWS.

Now, we are going to create two Web Servers in two different regions and going to configure different routing policies. I've choose Mumbai and N. Virginia.

5. Create an EC2 Instance in Mumbai region and connect to the instance
6. Install **httpd** package and create **index.html under /var/www/html and start the httpd service**and verify the access using public IP address.
7. Create an **Elastic Load Balancer** and add this EC2 instance to ELB and verify the access using the ELB name.



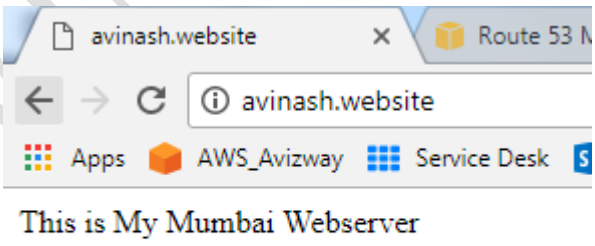
8. Choose another region (N. Virginia) and perform the same in N. Virginia region also. (Instance launch and ELB creation).

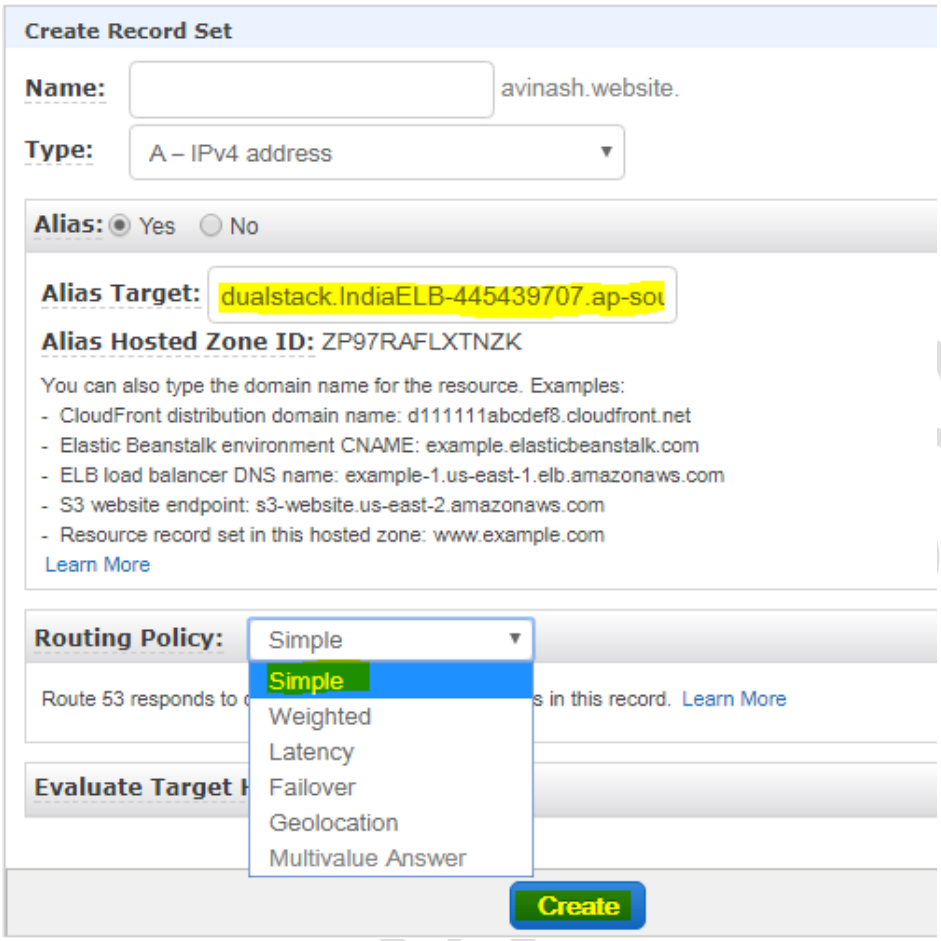


9. Now, we have two web servers in two different regions and we are going to configure routing policies between these two region resources.

Simple Routing Policy: This is the default routing policy when you create a new record set. This is most commonly used when you have a single resource that performs a given function for your domain

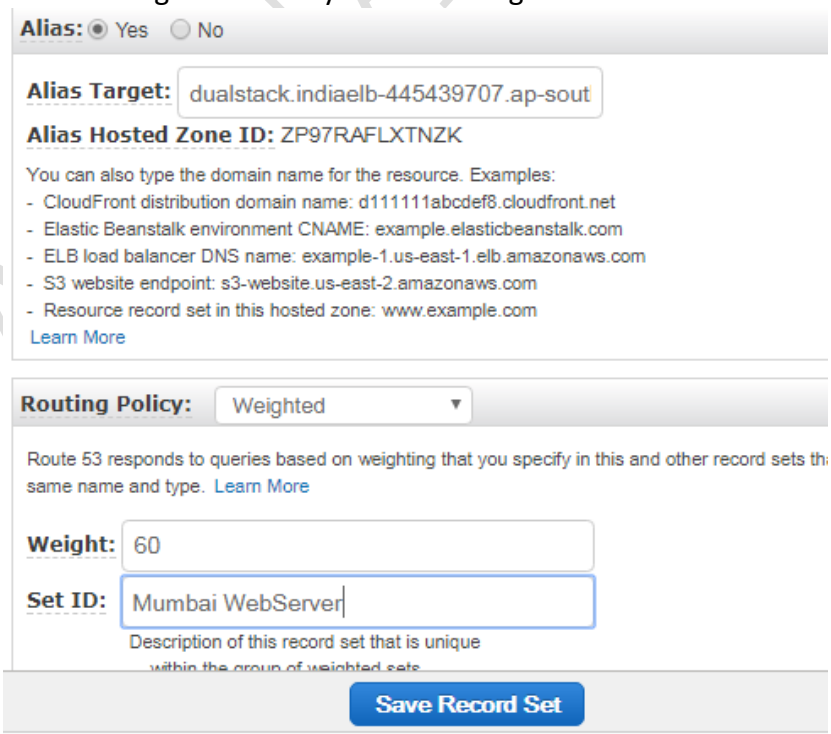
10. Select the Create Record Set option, you'll get an option like below.
    a. Give a name for your record ser
    b. Choose Type as **A – IPV4 address**
    c. Select **Alias**record and click on Alias Target option, you'll get all the available resources under AWS to map your domain with record set. Am selecting **Mumbai ELB**and selected simple **Routing Policy.**

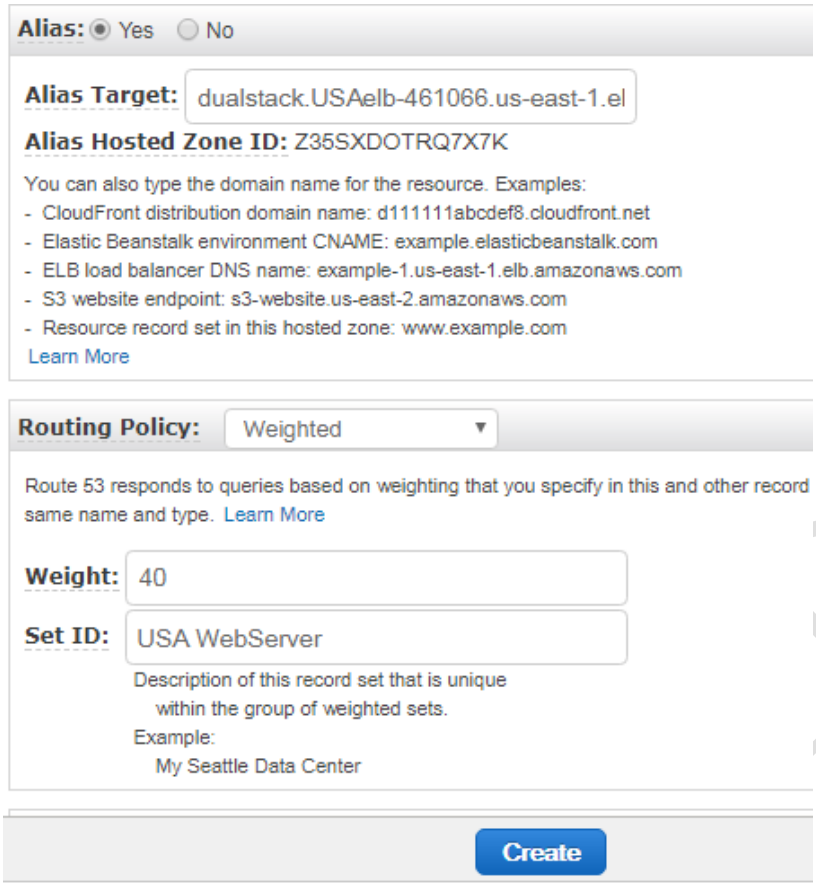




11. Now all my domain requests should route to Mumbai ELB as this is a simple routing policy and we'll have single resource for this routing type.

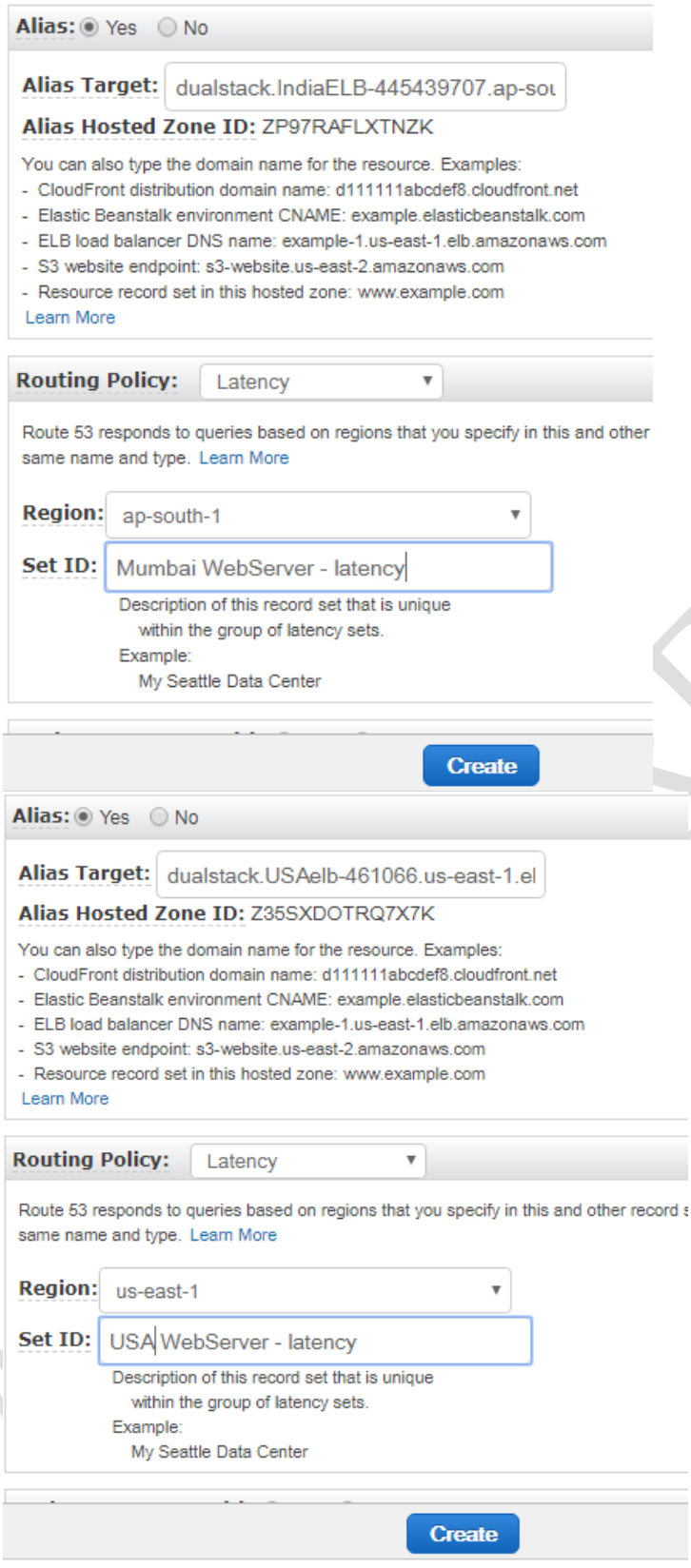
Weighted: Weighted Routing Policies let you split your traffic based on different weights assigned. Below we have assigned 60% of your traffic to go to AP-SOUTH-1 and 40% to go to US-EAST-1.





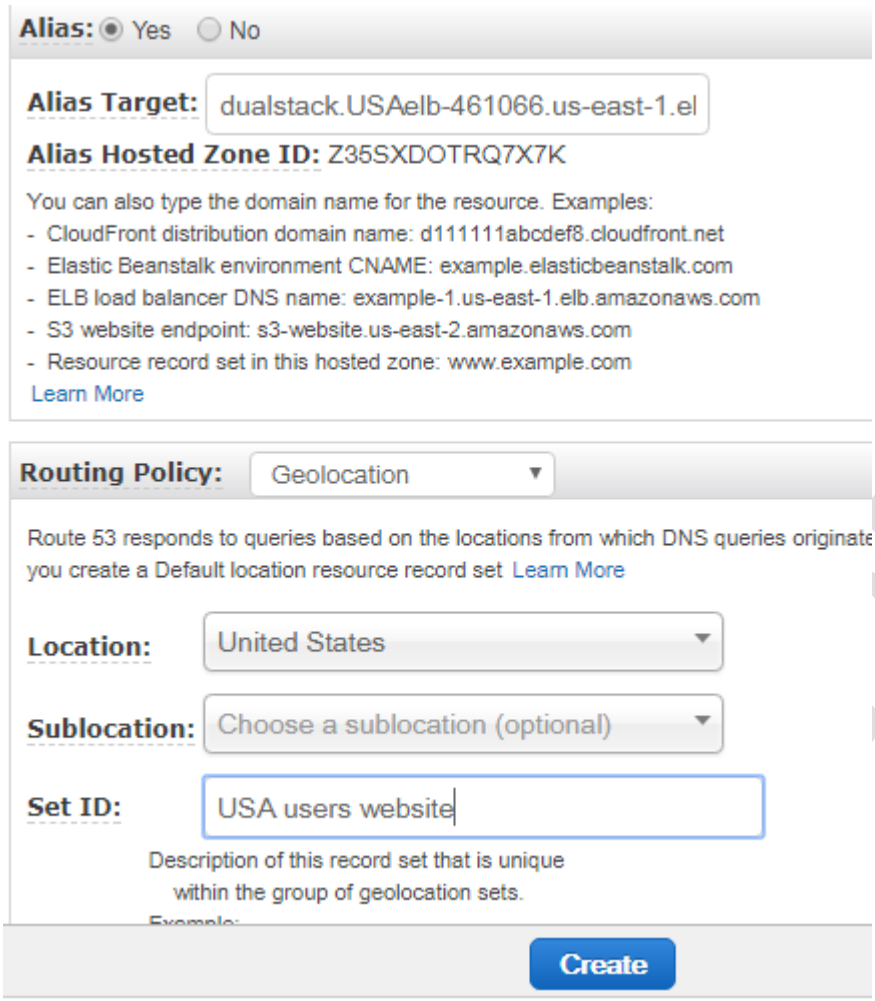
Latency: Latency based routing allows you to route your traffic based on the lowest network latency for your end user (ie which region will give them the fastest response time).

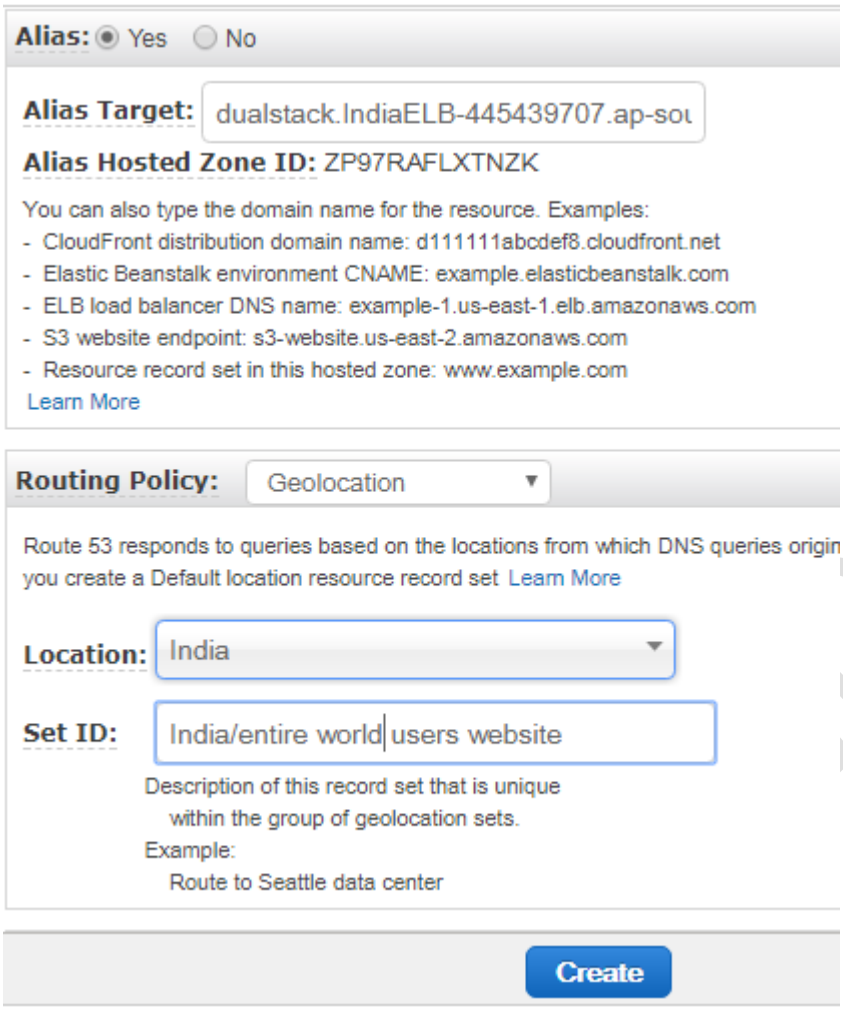
To use latency-based routing you create a latency resource record set for the Amazon EC2 (or ELB) resource in each region that hosts your website. When Amazon Route 53 receives a query for your site, it selects the latency resource record set for the region that gives the user the lowest latency. Route 53 then responds with the value associated with that resource record set.



Geolocation: Geolocation routing lets you choose where your traffic will be sent based on the geographic location of your users (ie the location from which DNS queries originate).

For example, you might want all queries from Europe to be routed to a fleet of EC2 instances that are specifically configured for your European customers. These servers may have the local language of your European customers and all prices are displayed in Euros.





Failover : Failover routing policies are used when you want to create an active/passive set up. For example you may want your primary site to be in US-East-1 and your secondary DR Site in AP-South-1.

Route53 will monitor the health of your primary site using a health check.

A health check monitors the health of your end points.

Configure health check

Route 53 health checks let you track the health status of your resources, such as web servers or mail servers

Name IND WS Healthcheck

What to monitor Endpoint

Monitor an endpoint

Multiple Route 53 health checkers will try to establish a TCP connection with the following resource to determi

Specify endpoint by IP address

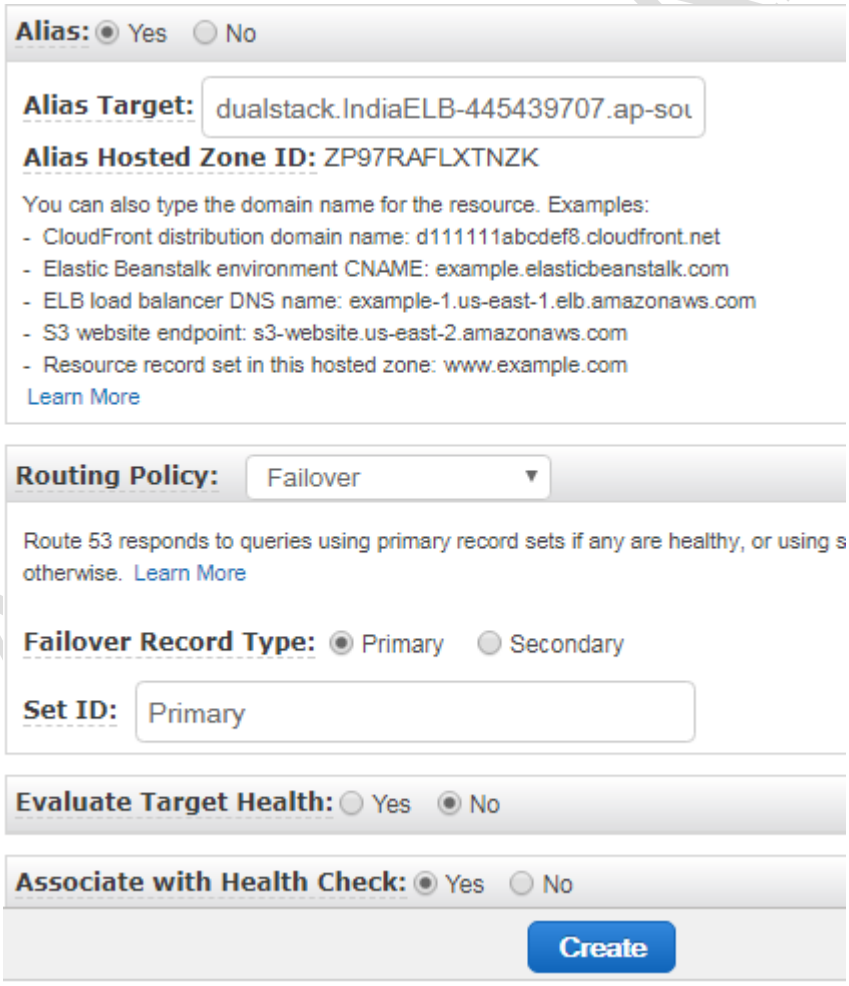
Protocol HTTP

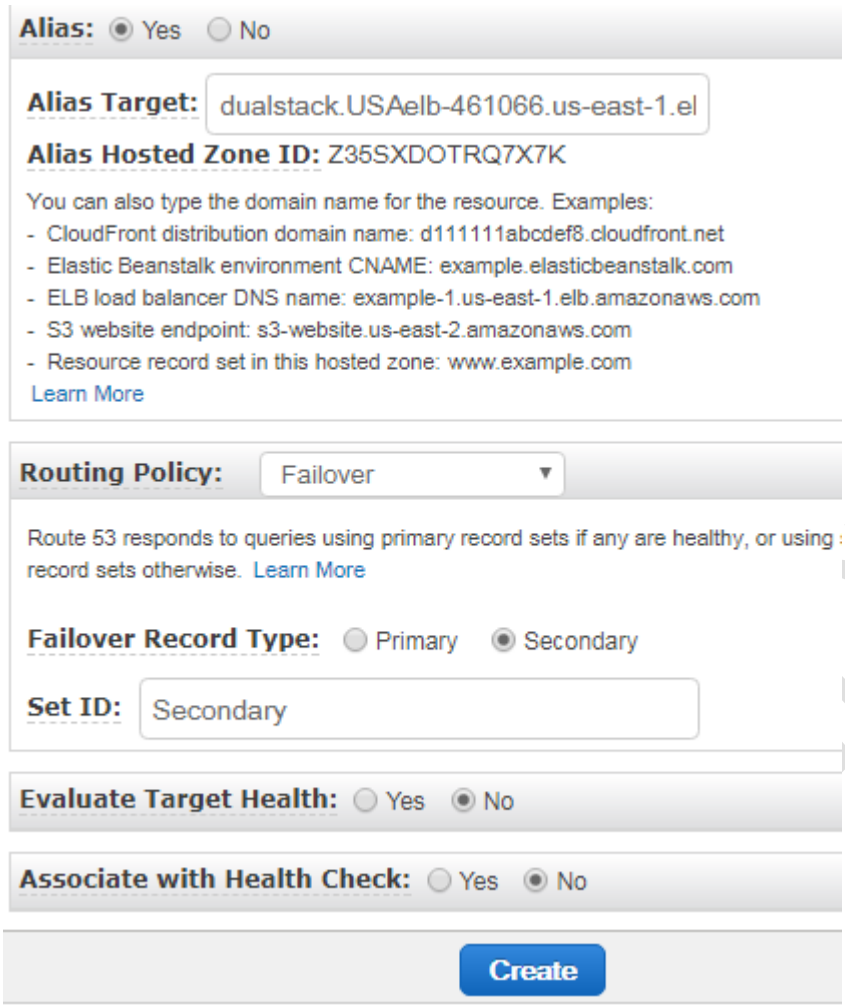
IP address

Host name www.example.com

Port * 80

Path / images





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

Databases:

In AWS we have wide range of database services to fit our application requirements. These database services are fully managed and can be launched in minutes with just a few clicks.
AWS database services include:

- Amazon Relational Database Service (Amazon RDS) support for six commonly used database engines
  - Amazon Aurora,
  - MySQL,
  - PostgreSQL
  - Oracle
  - MS SQL
  - Maria DB
- Amazon DynamoDB, a fast and flexible NoSQL database service,
- Amazon Redshift, a petabyte-scale data warehouse service, and
- Amazon Elasticache, an in-memory cache service with support for Memcached and Redis.
- AWS also provides the AWS Database Migration Service, a service which makes it easy and inexpensive to migrate your databases to AWS cloud.

Amazon Relational Database Service (Amazon RDS)

The most common type of database in use today is the relational database. The relationaldatabase has roots going back to the 1970s when Edgar F. Codd, working for IBM, developedthe concepts of the relational model. Today, relational databases power all types ofapplications from social media apps, e-commerce websites, and blogs to complex enterpriseapplications.
Amazon Relational Database Service (Amazon RDS) makes it easy to set up, Operate, and scale a relational database in the cloud. It provides cost-efficient and resizable capacity while managing time-consuming database administration taskssuch as hardware provisioning, database setup, patching and backups.

- A relational database consists of one or more tables, and a table consists of columns and rows similar to a spreadsheet.
- A database column contains a specific attribute of the record, such as a person's name, address, and telephone number.
- Each attribute is assigned a data type such as text, number, or date, and the database engine will reject invalid inputs.

StudentID	FirstName	LastName	Gender	Age
101	Avinash	Reddy	M	28
102	Anudeep	Thi	M	26
103	Aravind	Reddy	M	24
104	Vikas	Ch	M	24

Here is an example of a basic table that would sit in a relational database. There are five fieldswith different data types:
StudentID = Number or integer
FirstName = String
LastName = String
Gender = String (Character Length = 1)
Age = Integer
This sample table has four records, with each record representing an individual student. Eachstudent has a *StudentID* field, which is usually a unique number per student. A uniquenumber that identifies each student can be called a **primary key.**

A relational database can be categorized as either an Online Transaction Processing (OLTP)or Online Analytical Processing (OLAP) database system, depending on how the tables areorganized and how the application uses the relational database.
OLTP refers to transactionorientedapplications that are frequently writing and changing data (for example, data entryand e-commerce).
OLAP is typically the domain of data warehouses and refers to reporting or analyzing large data sets.

Data Warehouses: A data warehouse is a central repository for data that can come from one or more sources.This data repository is often a specialized type of relational database that can be used forreporting and analysis via OLAP. Organizations typically use data warehouses to compilereports and search the database using highly complex queries.

NoSQL Databases

NoSQL databases have gained significant popularity in recent years because they are oftensimpler to use, more flexible, and can achieve performance levels that are difficult orimpossible with traditional relational databases.
Traditional relational databases are difficultto scale beyond a single server without significant engineering and cost, but a NoSQLarchitecture allows for horizontal scalability on commodity hardware.

- NoSQL databases are non-relational and do not have the same table and column semantics of a relational database.
- NoSQL databases are instead often key/value stores or document stores with flexible schemas.

Advantages if you with RDS over On-Premise or EC2

Responsibility	Database On-Premise	Database on Amazon EC2	Database on Amazon RDS
App Optimization	You	You	You
Scaling	You	You	AWS
High Availability	You	You	AWS
Backups	You	You	AWS
DB Engine Patches	You	You	AWS
Software Installation	You	You	AWS
OS Patches	You	You	AWS
OS Installation	You	AWS	AWS
Server Maintenance	You	AWS	AWS
Rack and Stack	You	AWS	AWS
Power and Cooling	You	AWS	AWS

Database Engines

Amazon RDS supports six database engines: MySQL, PostgreSQL, MariaDB, Oracle, SQLServer, and Amazon Aurora.

MySQL: MySQL is one of the most popular open source databases in the world, and it is used to power a wide range of applications, from small personal blogs to some of the largest websites in the world.Amazon RDS MySQL allows you toconnect using standard MySQL tools such as MySQL Workbench or SQL Workbench/J.

PostgreSQL: PostgreSQL is a widely used open source database engine with a very rich set of features and advanced functionality.Amazon RDS PostgreSQL can be managed usingstandard tools like pgAdmin and supports standard JDBC/ODBC drivers.

MariaDB: MariaDB is apopular open source database engine built by the creators of MySQL and enhanced withenterprise tools and functionality.

Oracle: Oracle is one of the most popular relational databases used in the enterprise and is fullysupported by Amazon RDS. Amazon RDS supports access to schemas on a DB Instance using any standard SQL client application, such as Oracle SQL Plus.

Microsoft SQL Server

Microsoft SQL Server is another very popular relational database used in the enterprise.Amazon RDS allows Database Administrators (DBAs) to connect to their SQL Server DBInstance in the cloud using native tools like SQL Server Management Studio.

Amazon RDS SQL Server also supports four different editions of SQL Server: Express Edition,Web Edition, Standard Edition, and Enterprise Edition.

Licensing: AWS offers two licensing models: **License Included** and **Bring Your Own License (BYOL)**for Amazon RDS Oracle and Microsoft SQL Server as they are commercial software products.

Amazon Aurora: Amazon Aurora is a fully managed service, is MySQLcompatible, and provides for increased reliability and performance overstandard MySQL deployments. Amazon Aurora can deliver up to five times better performance compared to MySQL.We can usethe same code, tools, and applications that we use with existing MySQL databases withAmazon Aurora.

- 2 copies of your data is contained in each availability zone, with minimum of 3 availability zones. 6 copies of your data.
- Aurora is designed to transparently handle the loss of up to two copies of data without affecting database write availability and up to three copies without affecting read availability.
- Aurora storage is also self-healing. Data blocks and disks are continuously scanned for errors and repaired automatically.
- We can create two types of replications for Aurora
    - Aurora Replicas (currently 15)
    - MySQL Read Replicas (currently 5)

Storage Options

Amazon RDS is uses Amazon Elastic Block Store (Amazon EBS).Based on your performance and cost requirements we can select Magnetic, General Purpose (Solid State Drive [SSD]), or Provisioned IOPS (SSD). Depending on the database engine and workload, you can scale up to 4 to 6TB in provisioned storage andup to 30,000 IOPS.

Backup and Recovery

Amazon RDS provides two mechanisms for backing upthe database:

1. Automated backups and
2. Manual snapshots.

Automated Backups: An automated backup is an Amazon RDS feature that continuously tracks changes and backsup your database.

- You can set the backup retention period when you create a DB Instance. Default of 7 days, but you can modify the retention period up to a maximum of 35 days.
- When you delete a DB Instance, all automated backup snapshots are deleted and cannot be recovered.
- Automated backups will occur daily during a configurable 30-minute maintenance windowcalled the backup window.
- Automated backups are kept for a configurable number of days,called the backup retention period.
- You can restore your DB Instance to any specific timeduring this retention period, creating a new DB Instance.

Manual DB Snapshots: This is a manually initiated task. We have to perform this backup manually.

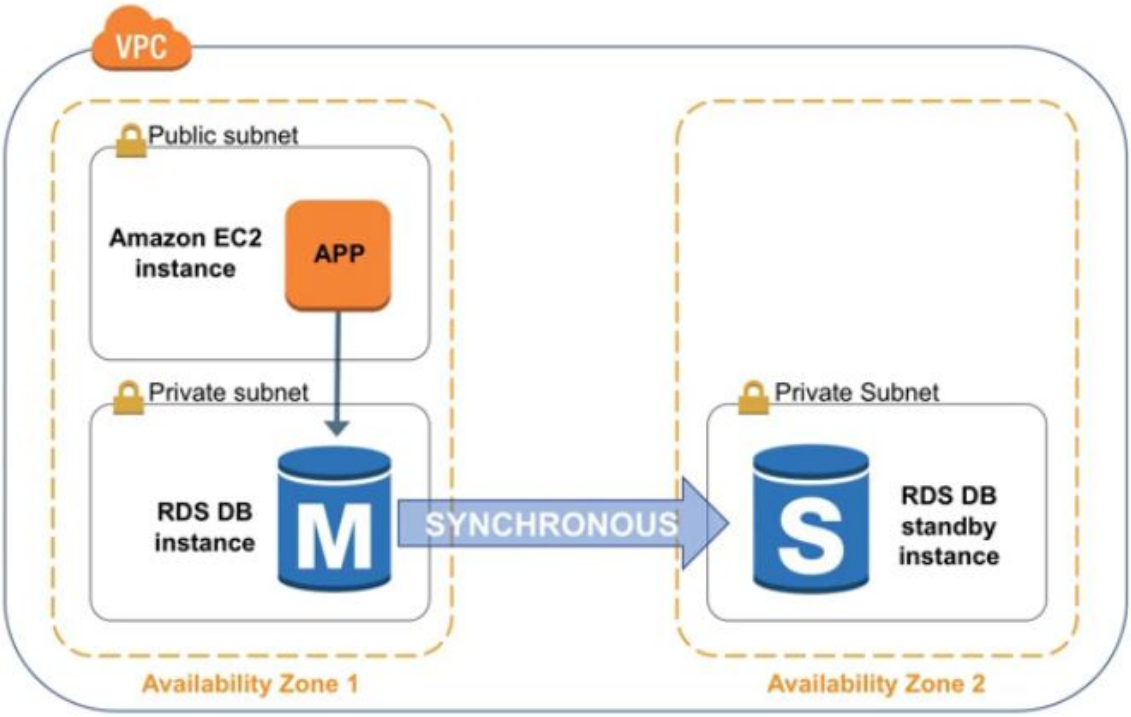
- A DB manual snapshot is initiated by us and can be created as frequently as we want.
- We can then restore the DB Instance to the specific state in the DB snapshot at any time.
- Manual DB snapshots are kept until you explicitly delete them with the Amazon RDS console.

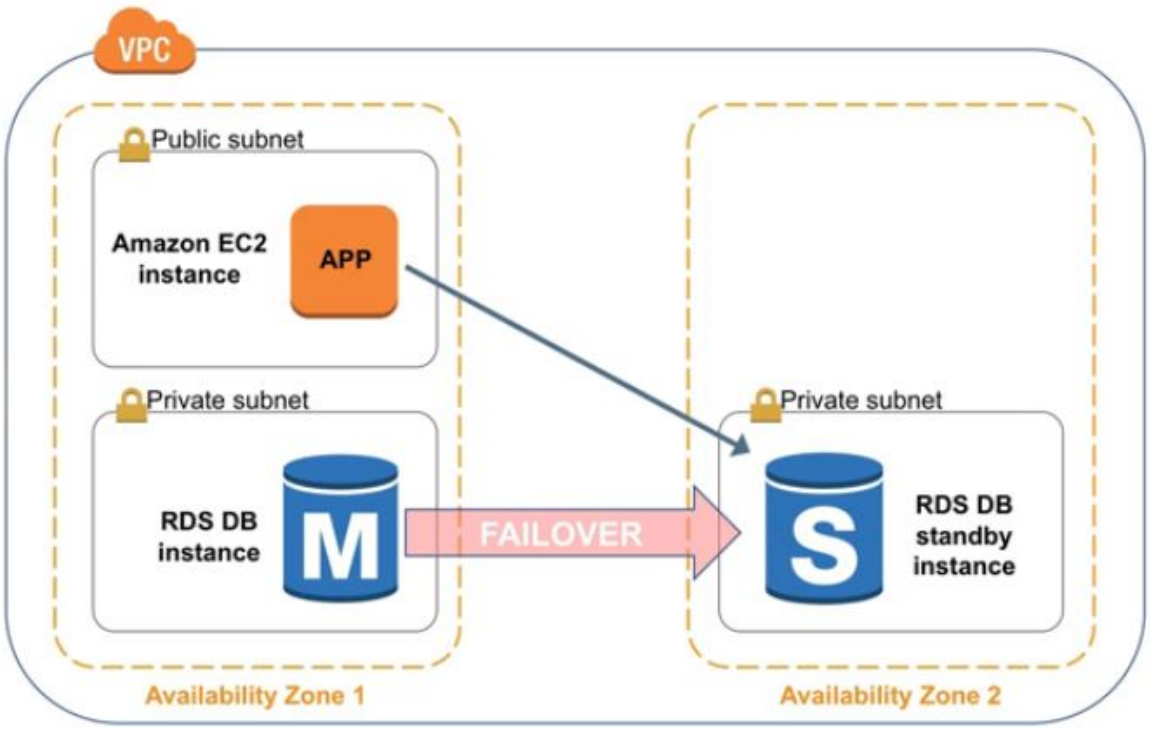
Recovery: We can use automated backup or manual snapshot to recovery the database.

- Amazon RDS allows you to recover your database using automated backups or manual DB snapshots.

- You cannot restore from a DB snapshot to an existing DB Instance; a new DB Instance is created when you restore.
- When using automated backups, Amazon RDS combines the daily backups performed during your predefined maintenance window in conjunction with transaction logs to enable you to restore your DB Instance to any point during your retention period, typically up to the last five minutes.

Multi-AZ: By using Multi-AZ we can increase the availability of the database using replication. We will get a same copy of production database in another availability zone for DR purpose (Disaster Recovery).

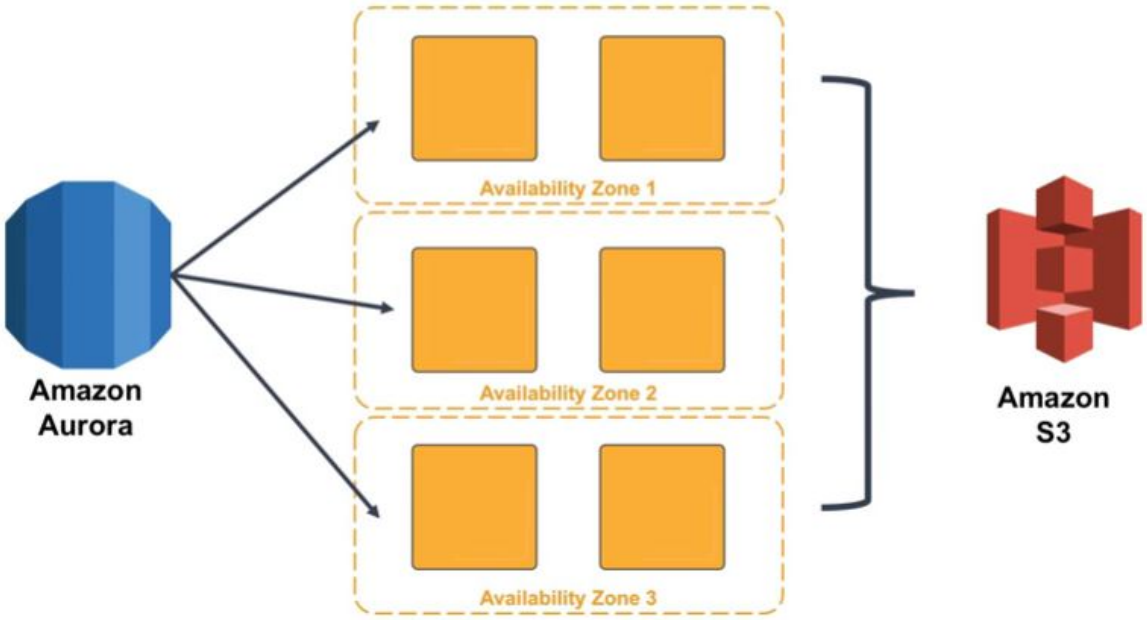




- Multi-AZ allows you to place a secondary copy of your database in another Availability Zone for disaster recovery purposes.
- Multi-AZ deployments are available for all types of Amazon RDS database engines.
- When you create a Multi-AZ DB Instance, a primary instance is created in one Availability Zone and a secondary instance is created in another Availability Zone.
- Amazon will takes care about the replication between primary Database and Secondary database.
- Amazon RDS detects and automatically recovers from most common failures for Multi-AZ deployments so that we will not get any downtimes and recovers without administrative intervention.
- Multi-AZ deployments are for disaster recoveryonly; they are not meant to enhance database performance.
- To improve database performance/Scaling we have to use read replicas or ElastiCache.

Read Replicas:

Read replica's allow you to have a read only copy of your production database. This is achieved by using Asynchronous replication from the primary RDS instance to the read replica. You use read replica's primarily for very read-heavy database workloads.



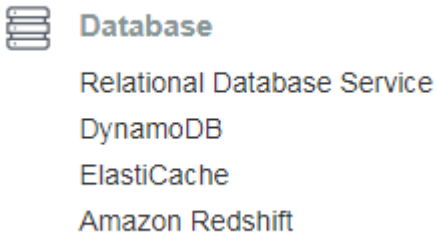
Read replicas are currently supported for:

- MySQL,
- PostgreSQL,
- MariaDB, and
- Amazon Aurora.

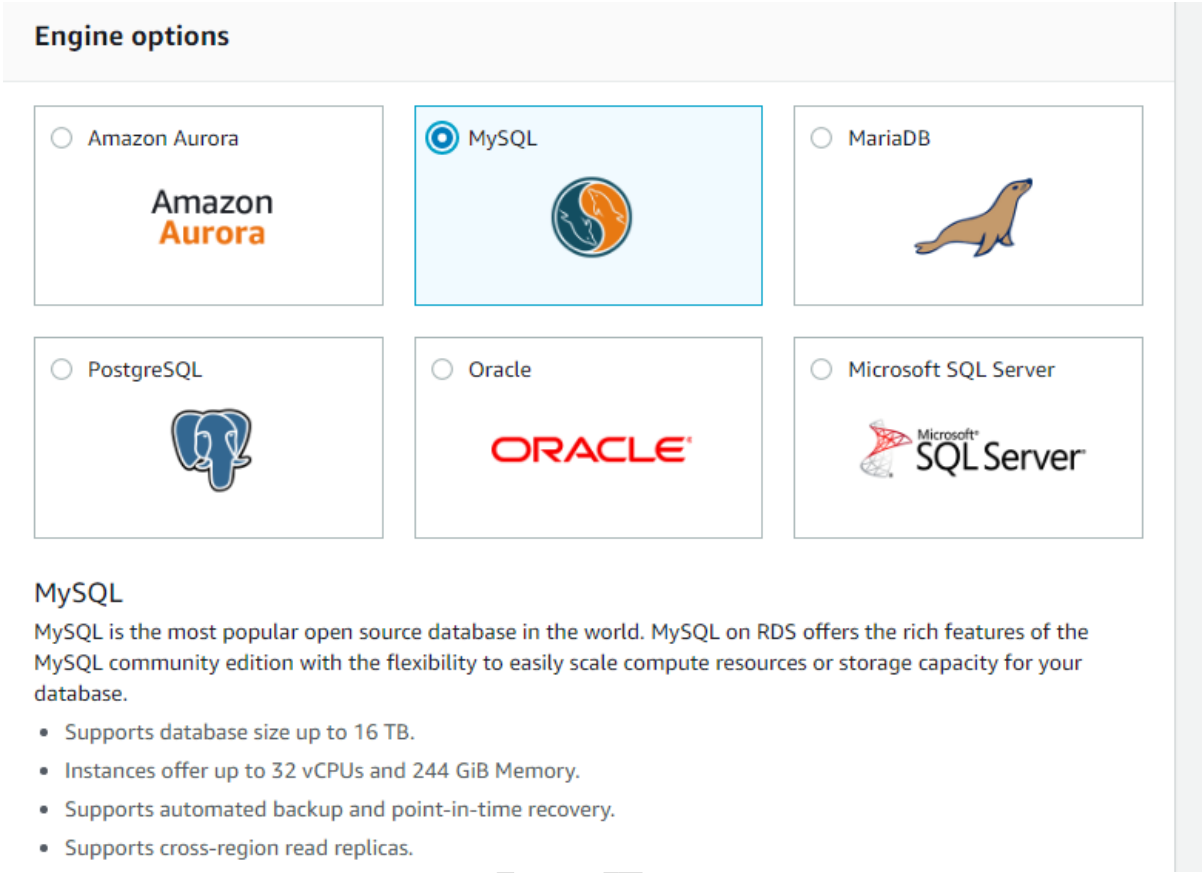
- Updates made to the source DB Instance are asynchronously copied to the read replica.
- You can create one or more replicas of a database within a single AWS Region or across multiple AWS Regions.
- We can use Read replicas for Scaling!!! Not for DR!
- Must have automatic backups turned on in order to deploy a read replica.
- You can have up to 5 read replicas copies of any databases
- You can have read replicas of read replicas and each read replica will have its own DNS end point.
- You cannot have Read Replicas that have Multi-AZ
- You can create Read Replica's of Multi-AZ source databases however.
- Read Replicas can be promoted to be their own databases. This breaks the replication.

Launching RDS instance:

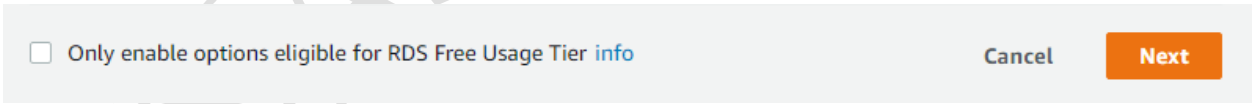
1. Log on to AWS account using the IAM credentials, and from the AWS Management Console, select the Relational Database Service option underDatabase.



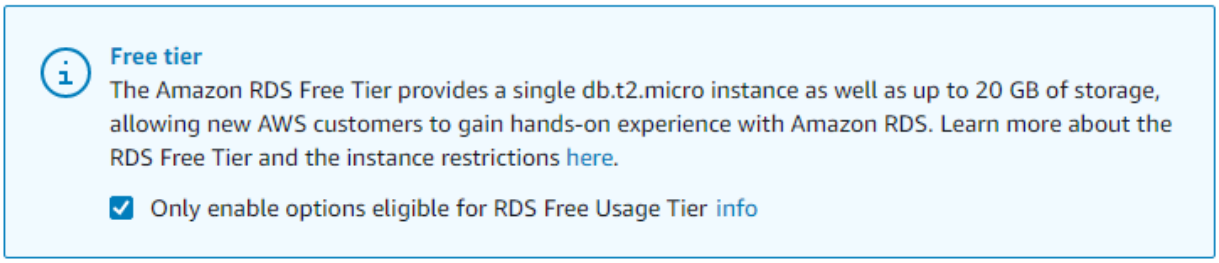
2. Select the **Engine.**



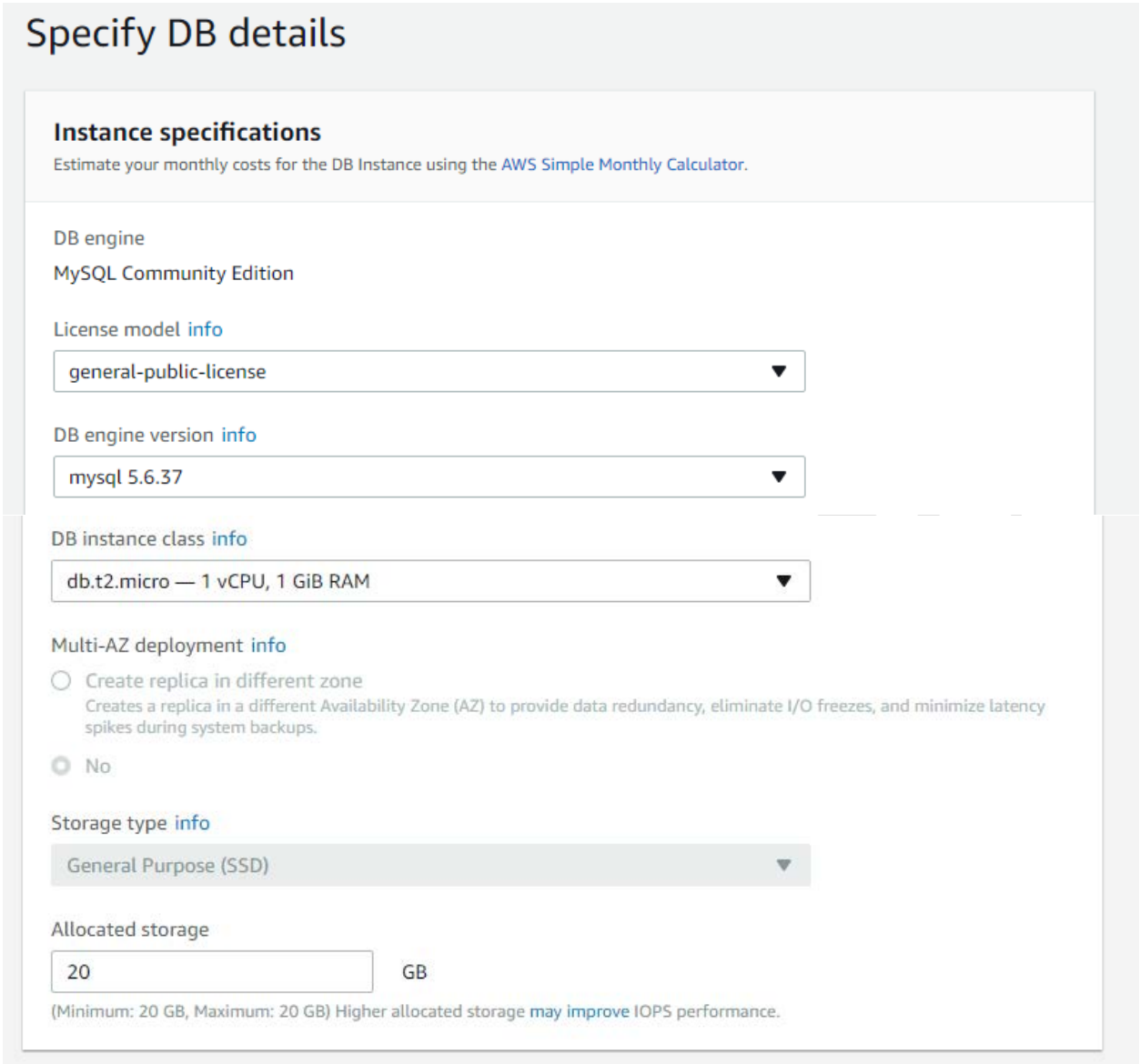
3. As we discussed earlier, we have six relational db engines are available with amazon RDS, Now am going to launch MySQL.
   - If you want to use **Free Tier eligibility**make sure you select the tick mark for the below option and click on **Next**

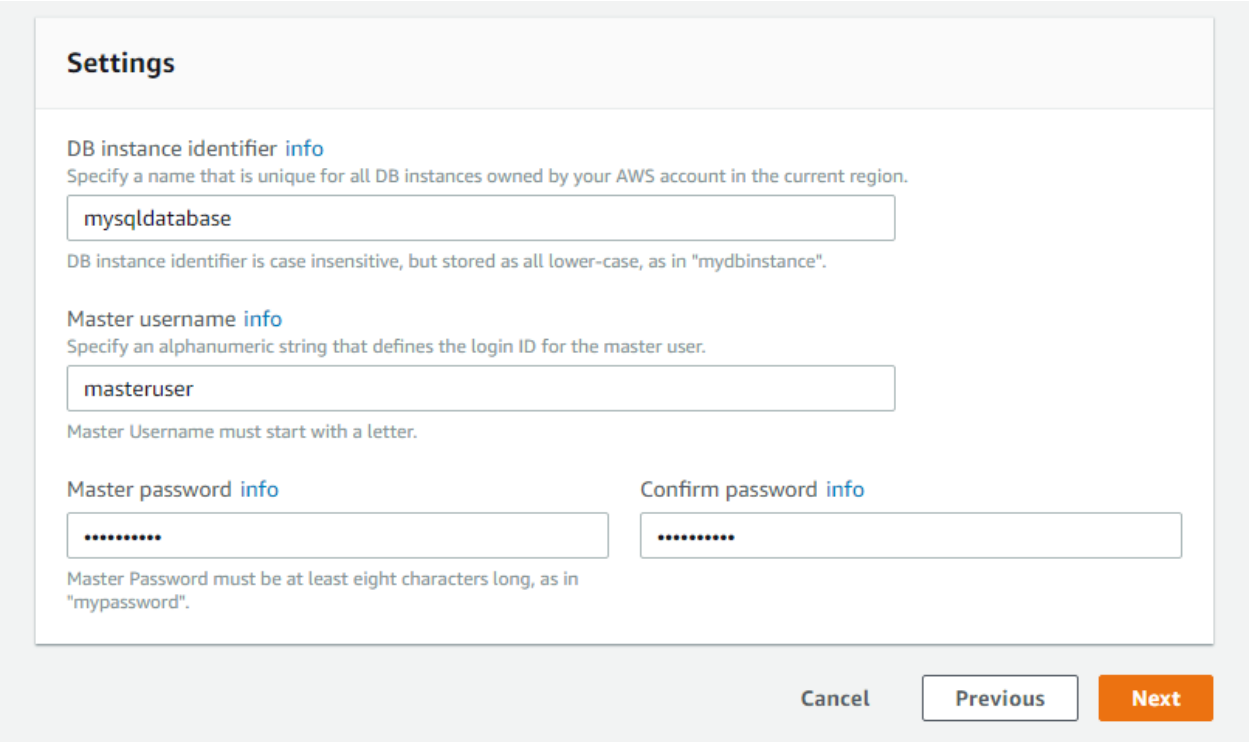


4. If you don't want to get charged or want to use free tier eligibility make sure you select this option **"Only enable options eligible for RDS Free Usage Tier"**



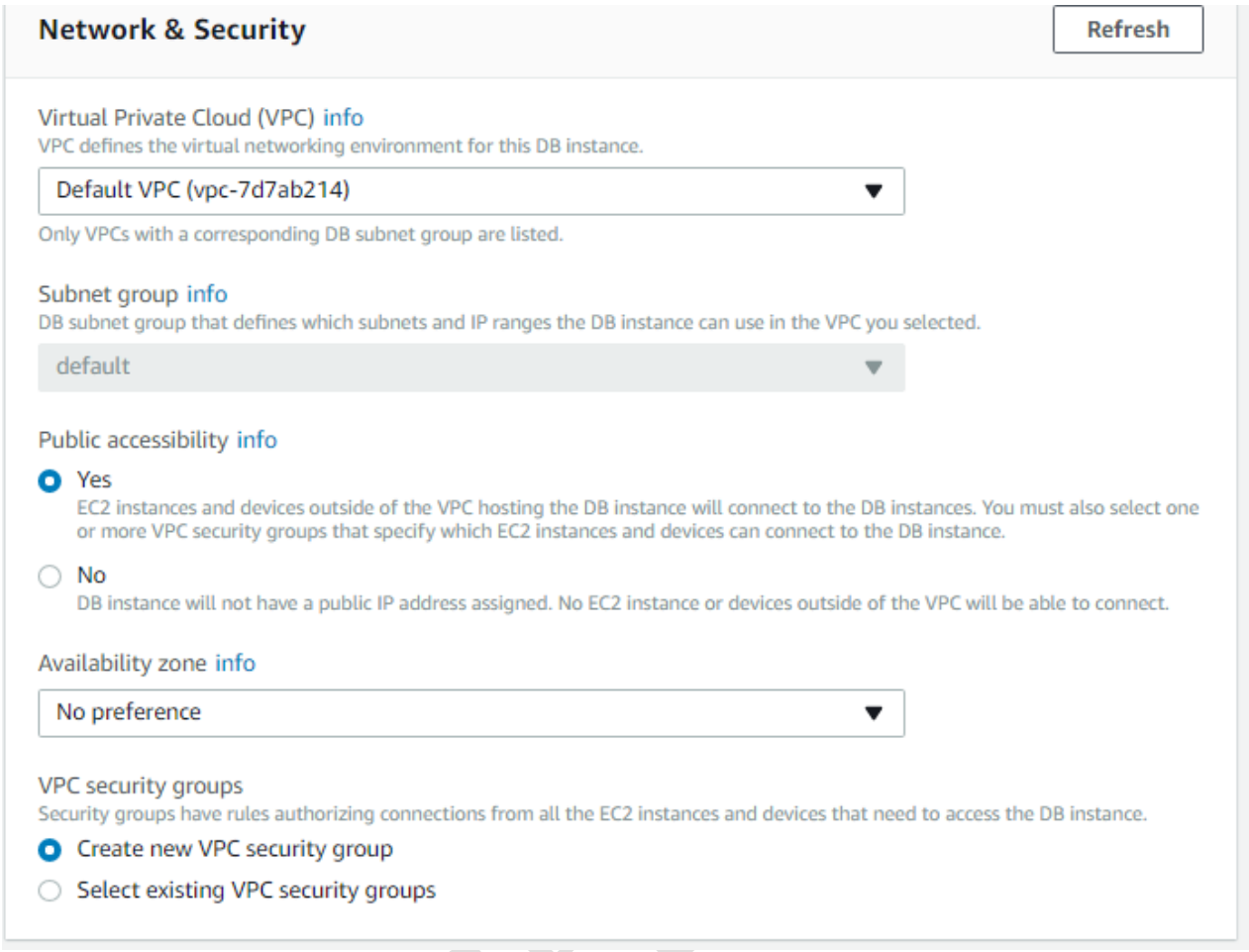
5. I want to use free trier for the DB instance, so selecting MySQL community edition, in next we have to specify the DB Details.


Specify DB details
Instance specifications
Estimate your monthly costs for the DB Instance using the AWS Simple Monthly Calculator.
DB engine
MySQL Community Edition
License model info
general-public-license
DB engine version info
mysql 5.6.37
DB instance class info
db.t2.micro — 1 vCPU, 1 GiB RAM
Multi-AZ deployment info
Create replica in different zone
Creates a replica in a different Availability Zone (AZ) to provide data redundancy, eliminate I/O freezes, and minimize latency spikes during system backups.
No
Storage type info
General Purpose (SSD)
Allocated storage
20
GB
(Minimum: 20 GB, Maximum: 20 GB) Higher allocated storage may improve IOPS performance.

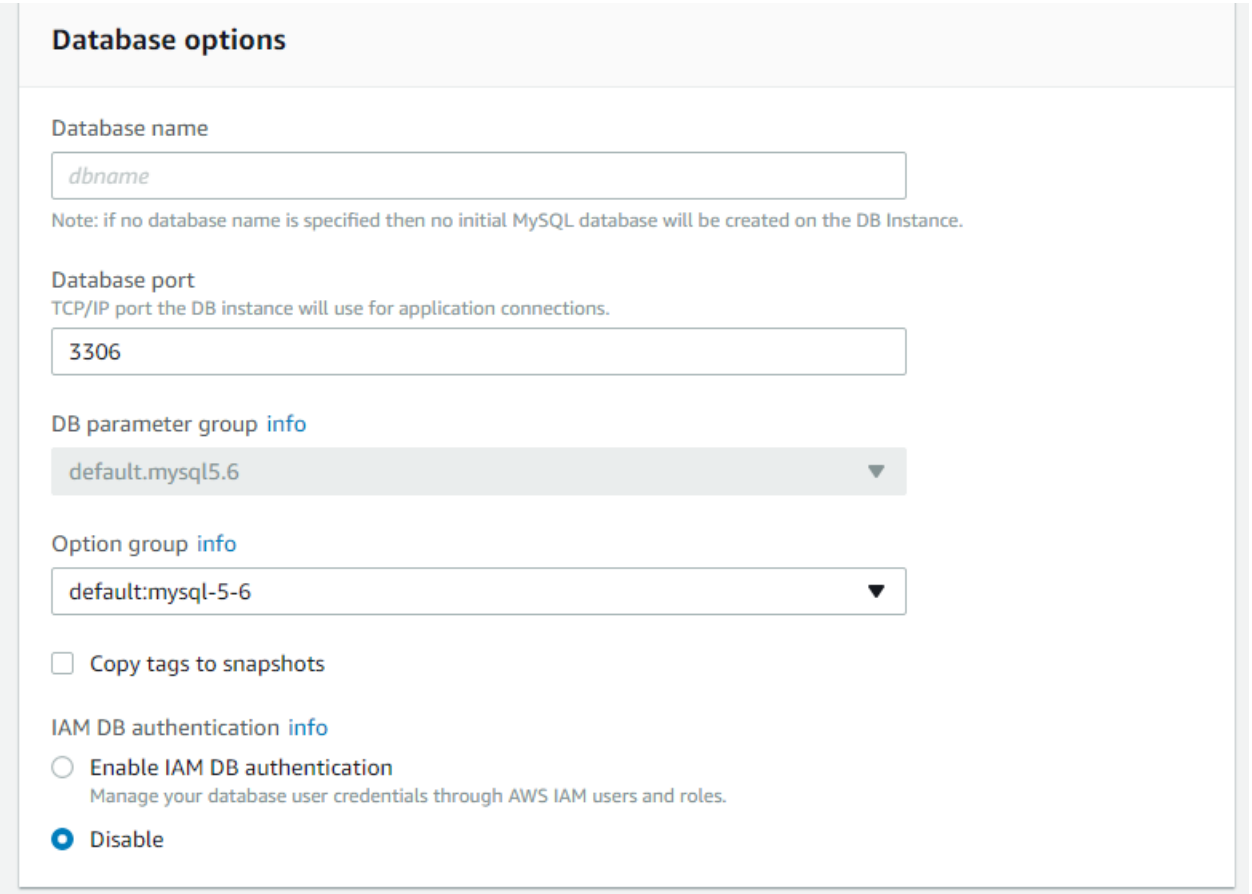


- **DB Engine:** We have selected the Db Engine as MySQL.
- **License Model:** MySQL databases have only one license model; that is, generalpublic-license.AWS provides the required license keys for your databases,so you don't have to separately purchase one.
- **DB Engine Version:**Select the appropriate DB Engine Version as per yourrequirements. RDS provides and supports a variety of database engine versions that you can choose from.
- **DB Instance Class:** We have multiple DB Instance Classes with various configurations (vCPU & RAM), Select the appropriate one as per requirement.
- **Multi-AZ Deployment:** Select "Yes/No" for Multi-AZ based on requirement.
- **Storage Type:**Select the Storage Type between **"General purpose SSD"** and **"Provisioned IOPS".**
- **Allocated Storage:** We can allocate the storage for db instance. We can select from **20 GB to 6TB**.
- **DB Instance Identifier:** Give a valid name for the DB instance and this must be unique in the selected region.
- **Master Username:**Give a valid username to login to the Db instance.
- **Master Password:** Give a valid password for the master username.

6. In Step 3, we need to Configure Advanced Settings.

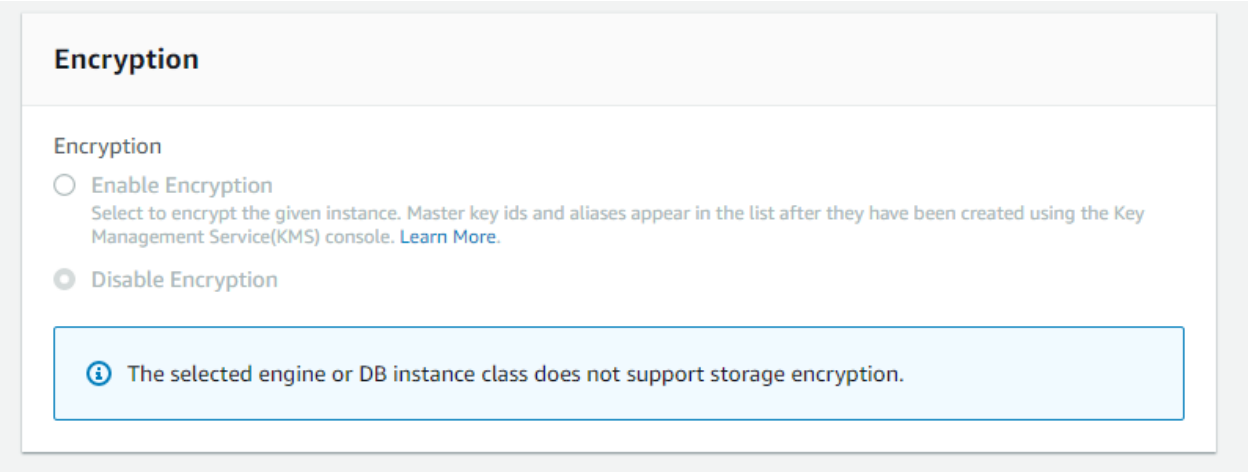


- **VPC:**Select the name of the VPC that will host your MySQL DB instance. Here am selecting Default VPC to host this instance.
- **Subnet Group:** Selecting the default Subnet Group.
- **Public Accessible:**Select "**Yes"** if you want EC2 instances and devices outside of the VPC hosting the DB instance to connect to the DB instance. If you select No, Amazon RDS will not get a public IP address to the DB instance, so we cannot connect over internet.
- **Availability Zone:**We can select the desired AZ based on the region.
- **VPC Security Groups:** We have to attach a security group to the Db instance. It works same as the EC2 instance security group, As we are launch **MySQL port number 3306** must be opened. For **MsSQL port number is 1433**.

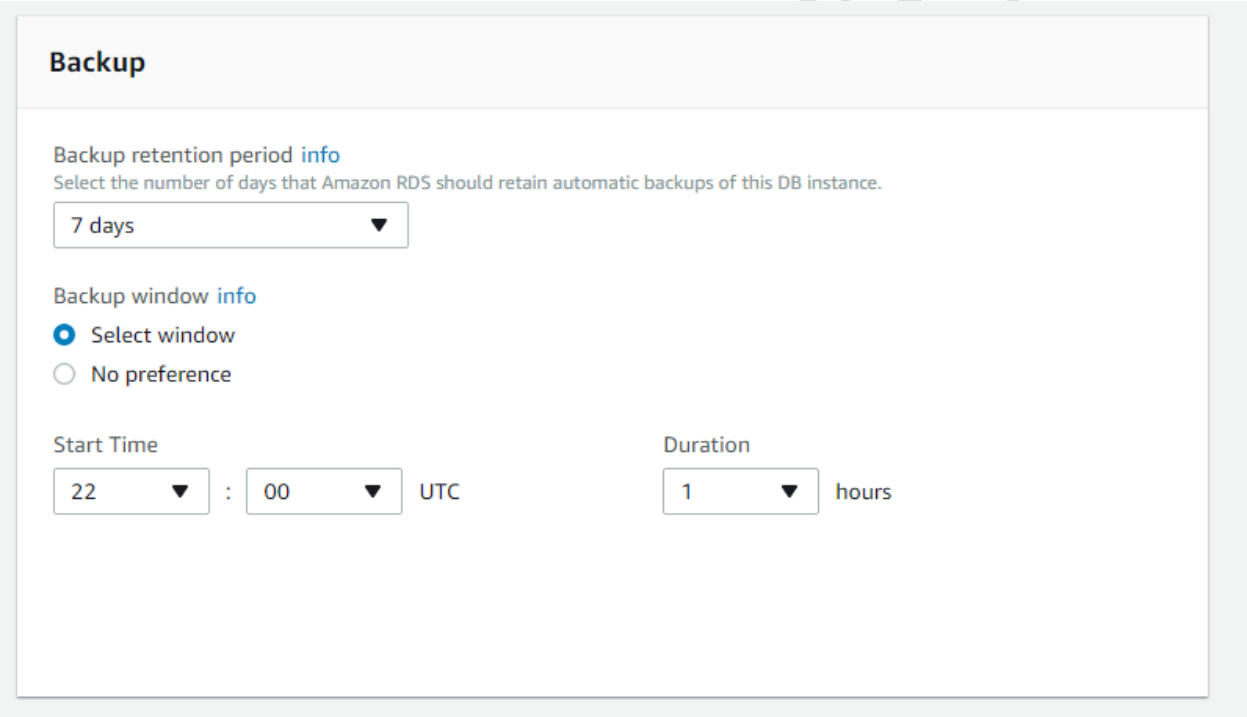


- **Database Name:** Provide a suitable database name here. RDS will not create and initialize any database unless you specify a name here.
- **Database Port:** Provide the port number using which you wish to access the database. MySQL's default port number is 3306. We cannot change the default port number after db instance launch.
- **DB parameter Group:** DB parameter groups are logical groupings of databaseengine configurations that you can apply to one or more DB instances at the same time. Go with the default option here.
- **Option Group:** This option is similar to DB parameter groups in that they tooprovide and support few additional configuration parameters that make it easy to manage databases
- **Copy Tags To Snapshots:** Give a tick on checkbox if you want to copy the tags to created snapshots of the db instance.
- **Enable IAM DB Authentication:** We can use IAM users to use the db, but the IAM user need to have appropriate permissions. Select **"Yes"** to manage your database user credentials through AWS IAM users and roles.

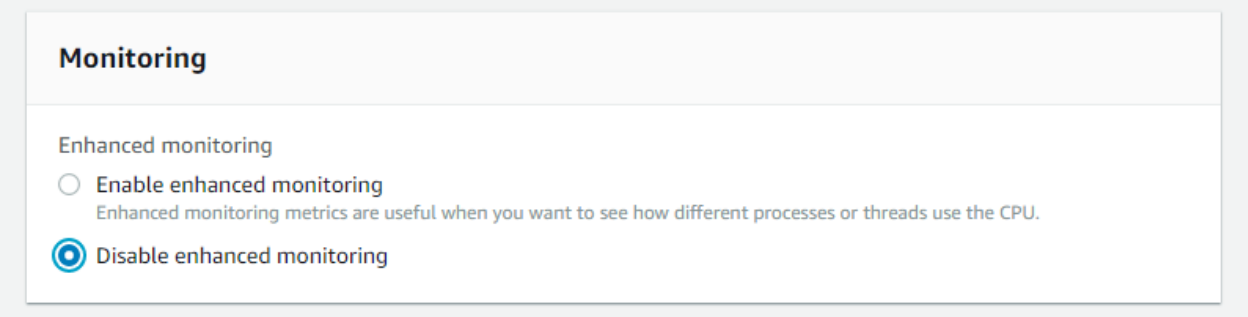
Enable Encryption: RDS provides standard AES-256 encryption algorithms for encrypting data at rest. T2.micro will not support the encryption.



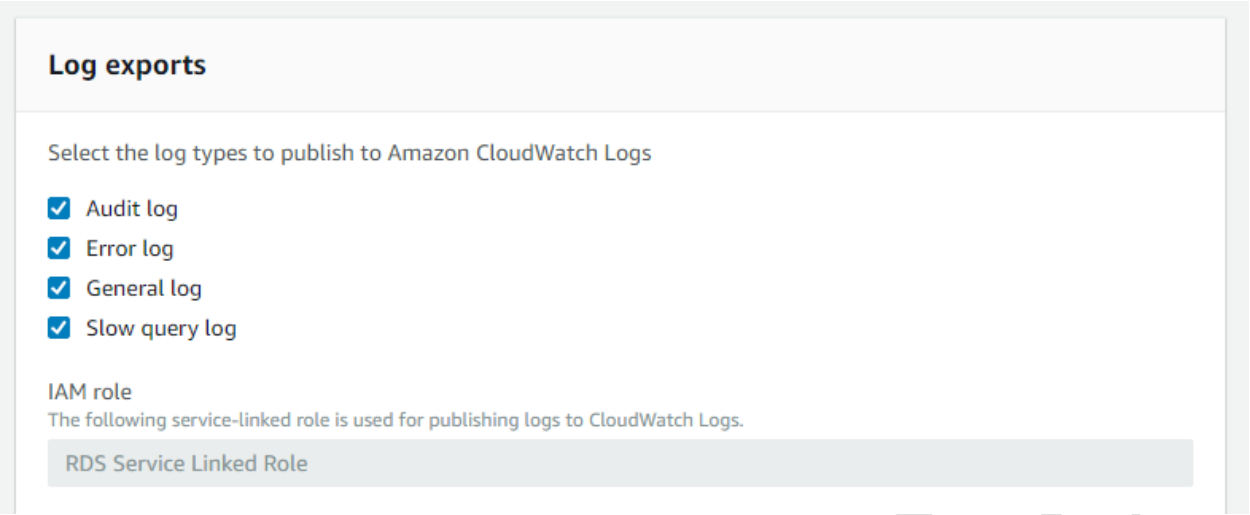
- We can set the BackupRetention Period as well as the Backup window's Start Time and Duration. As discussed above if we enable amazoncreatesautomatedbackups.



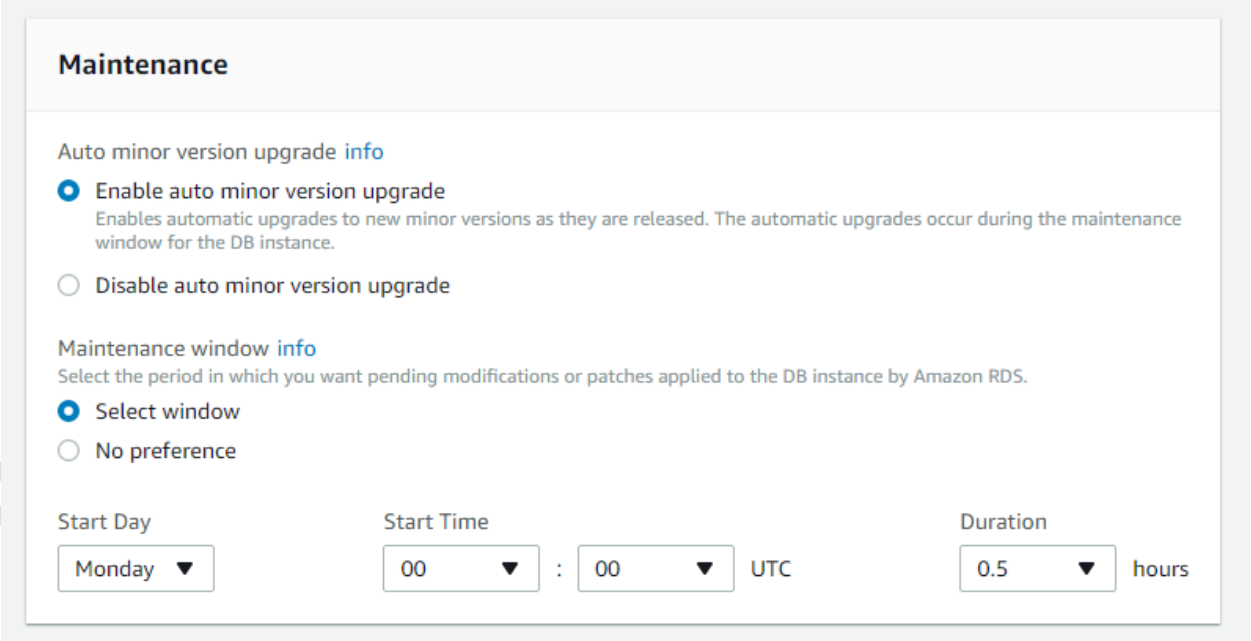
- **Enable Enhanced Monitoring:** We can use Cloudwatch to monitor the db instances, give yes if you want to change the default monitoring period to detailed monitoring.



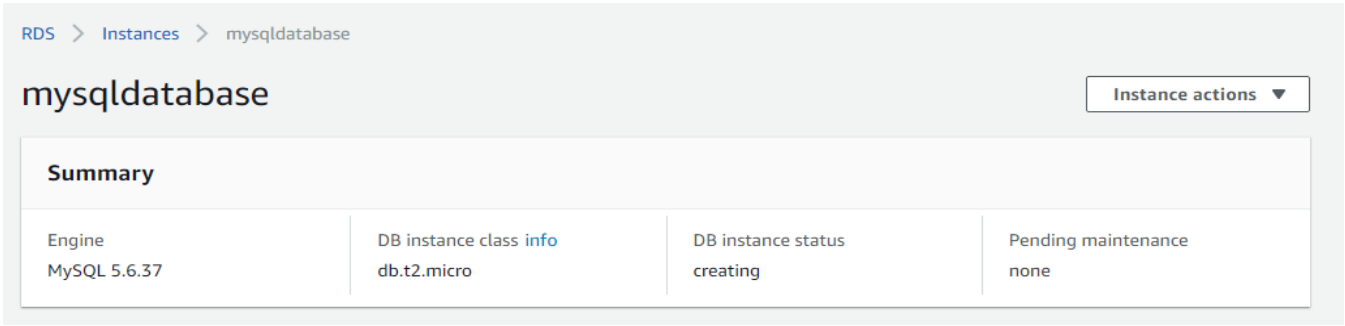
- **Log Exports:**We can get the required logs for the Cloudwatch service.

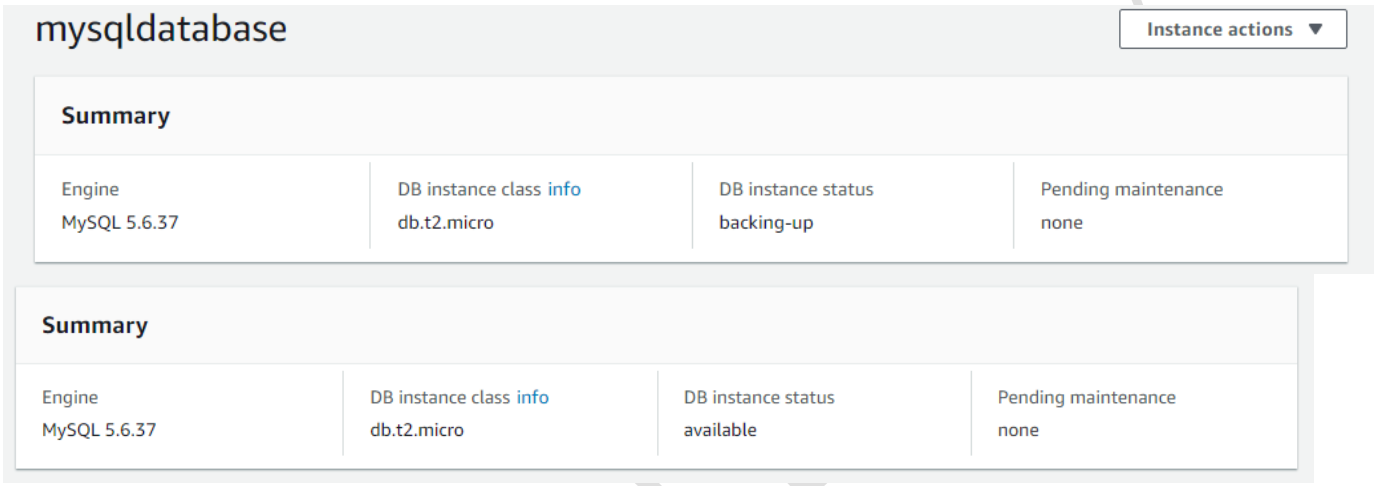


- **Auto Minor Version upgrade:** Specify Yes to enable automatic upgrades to new minor versions as they are released. The automatic upgrades occur during the maintenance window for the DB instance.
- **Maintenance Window:** We can select the period in which you want pending modifications or patches applied to the DB instance by Amazon RDS. Any such maintenance should be started and completed within the selected period. If you do not select a period, Amazon RDS will assign a period randomly.



7. After configuring all the above steps, choose Launch DB instance option. DB instance creation will be initiate now.





We have four steps for instance launch stage: Creating, Modifying, Backing-Up and Available.

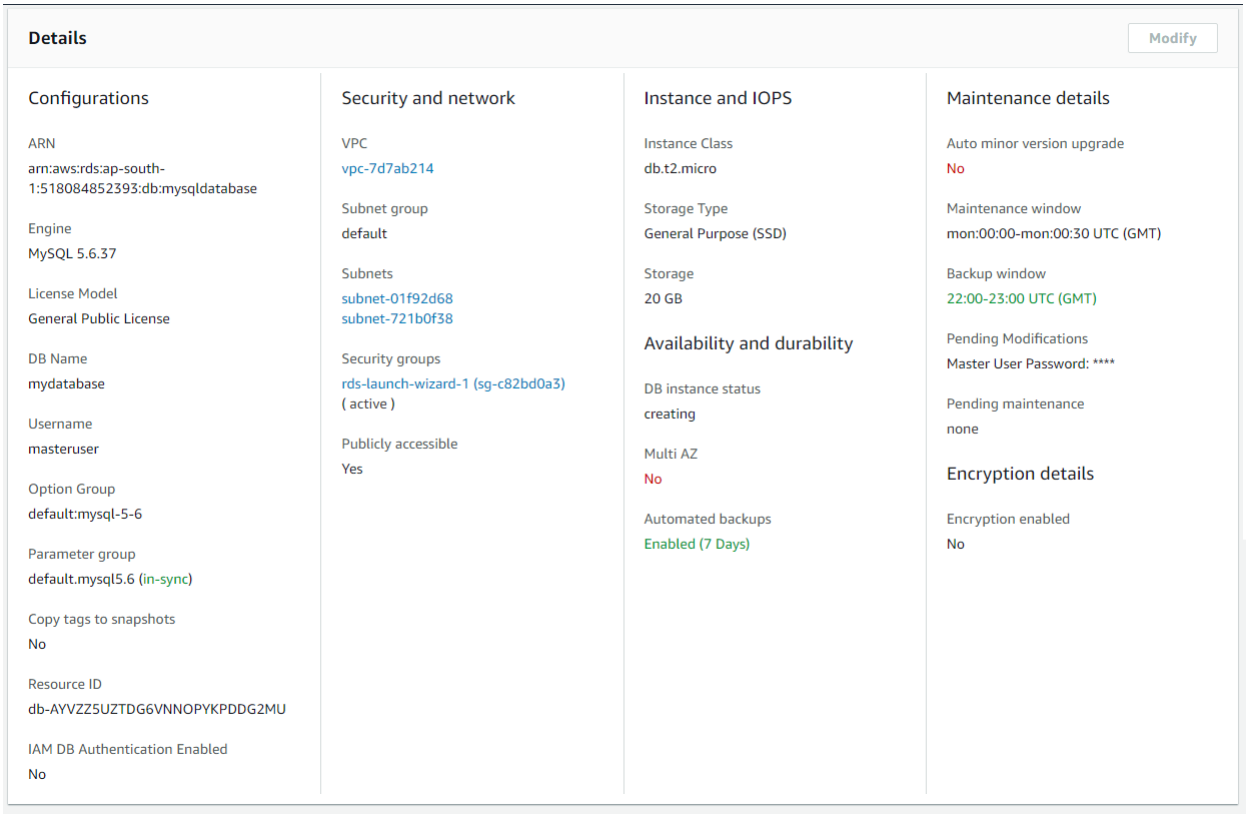
Creating: This is the first stage of any DB instance's lifecycle where the instance isactually created by RDS. During this time, your database will remain inaccessible.

Modifying: This state occurs whenever the DB instance enters any modificationseither set by you or by RDS itself.

Backing-up: RDS will automatically take a backup of your DB instance when it isfirst created. You can view all your DB instance snapshots using the Snapshots optionon the navigation pane.

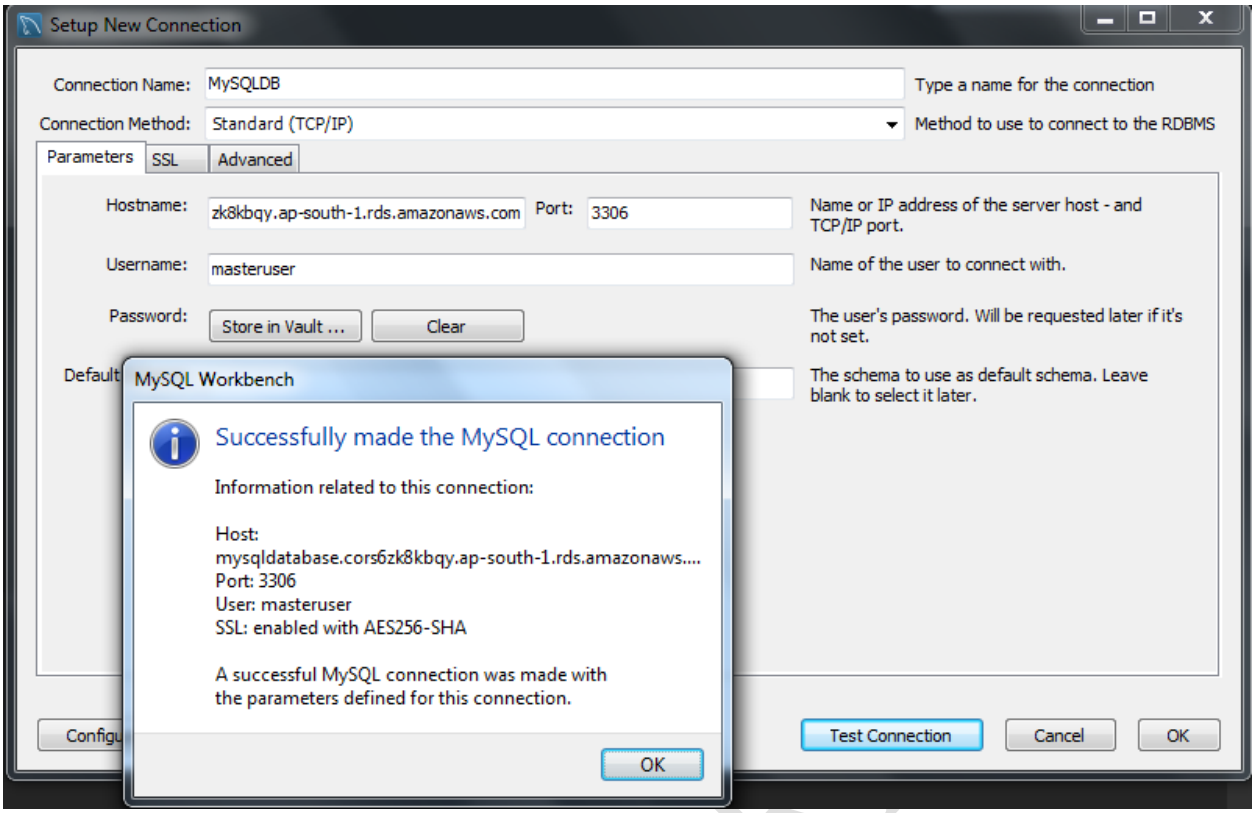
Available: This status indicates that your DB instance is available and ready for use.You can now access your database remotely by copying the database's endpoint.

Here is the details for newly launched RDS instance.

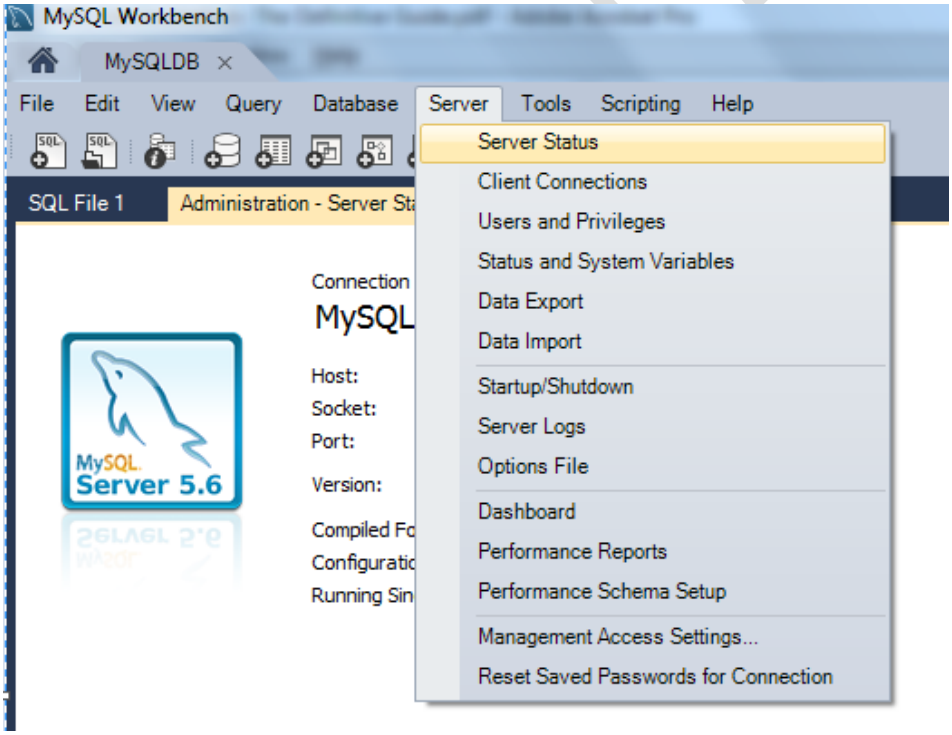


8. To test the connectivity we are going to use MySQL Workbench application, Download and install on any of the local machine or EC2 instance if you want to test it in graphical manner. You can download the MySQL workbench from the following URL: https://dev.mysql.com/downloads/workbench/

9. I've copied the Endpoint URL of my DB instance and opened the installed MySQL workbench application and add a connection and give a name for the connection, Enter the **Endpoint name in Hostname field**, port number is **3306**, Enter **username** and click on **Test Connection** and Give the password, you should get a Successful rest result.



10. We can verify the Server Status by navigating to Server and selecting the Server Status option.



11. By using the workbench, we can create databases, schemas and we can manage the database graphically.

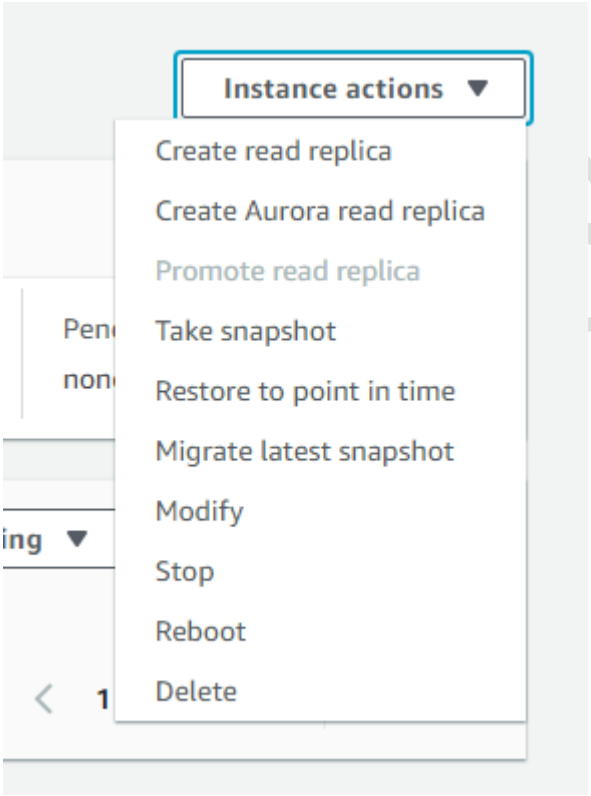
To test the MySql from Linux machine, Launch a Linux instance and install the mysql package by running **yum install mysql**option.
After launching the Linux instance, Installmysql package by running

yum install mysql

Then run **# mysql -u <USERNAME> -h <DATABSE_ENDPOINT> -p**and press enter, It'll ask you to enter the password of connecting user, then you can access the mysql database.

```
[root@ip-172-31-24-253 ec2-user]# mysql -u masteruser -h mysqldatabase.cors6zk8k
bqy.ap-south-1.rds.amazonaws.com -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 36
Server version: 5.6.37-log MySQL Community Server (GPL)

Copyright (c) 2000, 2017, 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> show databases
    ->
```

DB Instance Actions: We can find the below options when you select the **db instance** and choose **Instance Actions** option.



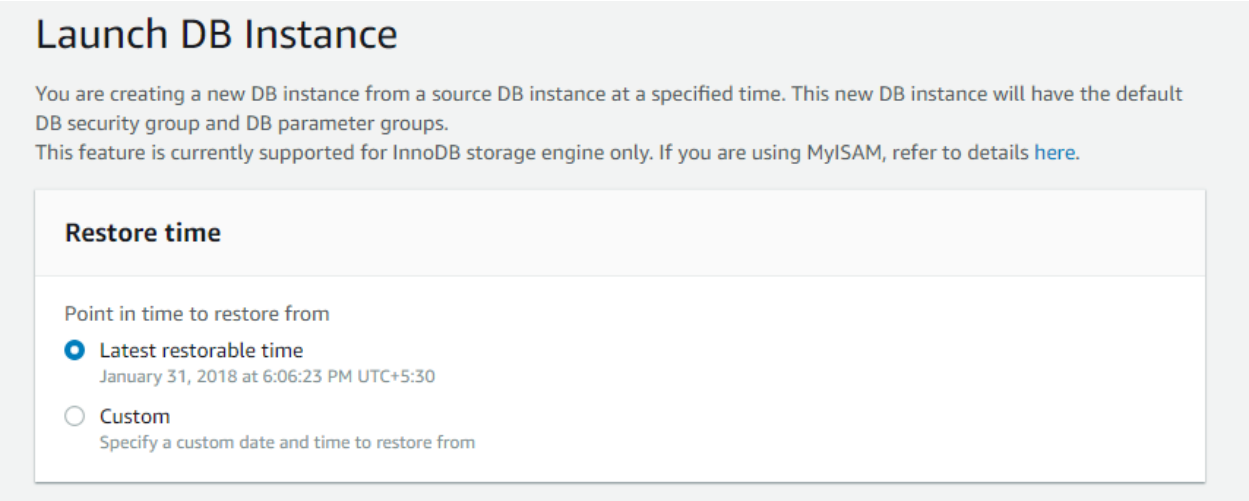
Create Read Replica: As we discussed above, we can create read replicas of the primary db instance for scaling purpose, We'll get a new endpoint for read replicas and the launch wizard is almost same new db instance launch.

Create Aurora Read Replica: If we need a replica with aurora db engine, we can choose this option and follow wizard. Read replica will create with aurora db engine.

Promote Read Replica: If you want to promote read replica to a standalone db instance, we can select this option, But the replication between primary db and read replica will breakdown.

Take Snapshot: For backups of the db instance we can use the snapsots.

Restore to Point in Time: With this option we can create a new DB Instance from a source DB Instance at a specified time. This new DB Instance will have the default DB Security Group and DB Parameter Groups.



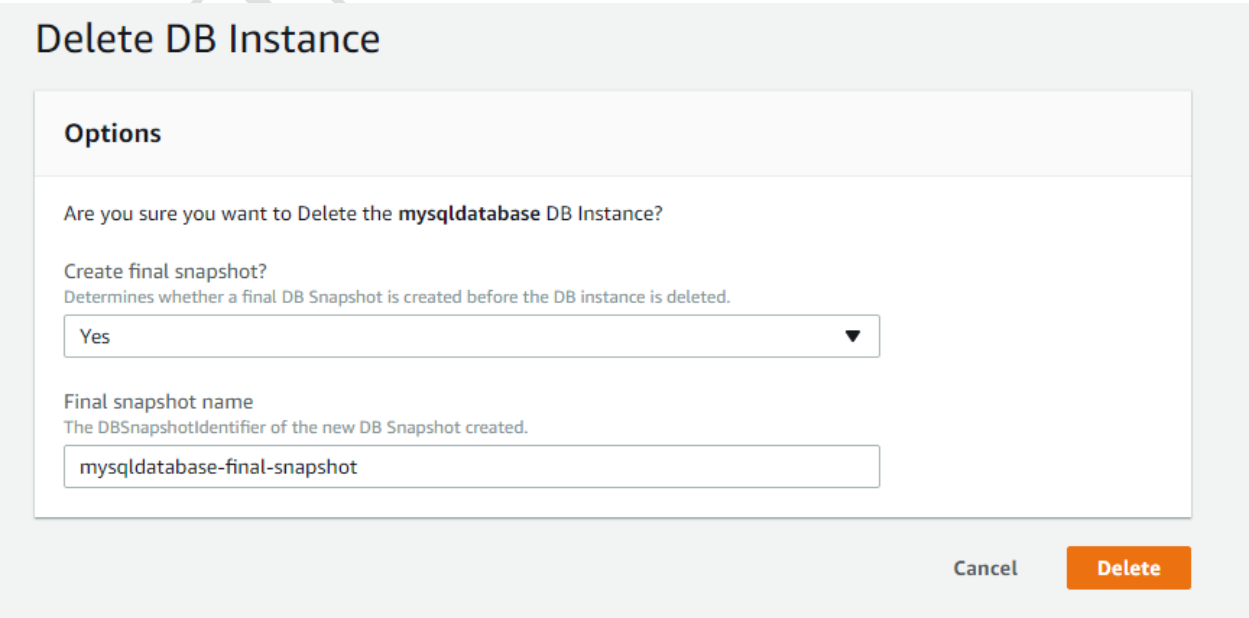
Migrate Latest Snapshot: We can migrate the selected database to a new DB Engine by selecting desired options for the migrated instance. For mysql “Aurora” and “mariadb”.

Modify: By using modify option, we can change the db instance properties i.e; DB engine version, instance class, storage options, master password, backup retention period and maintenance periods.

Stop: Instance will changes it status to Stopped state, we can start at anytime.

Reboot: underlying instances operating system will reboot.

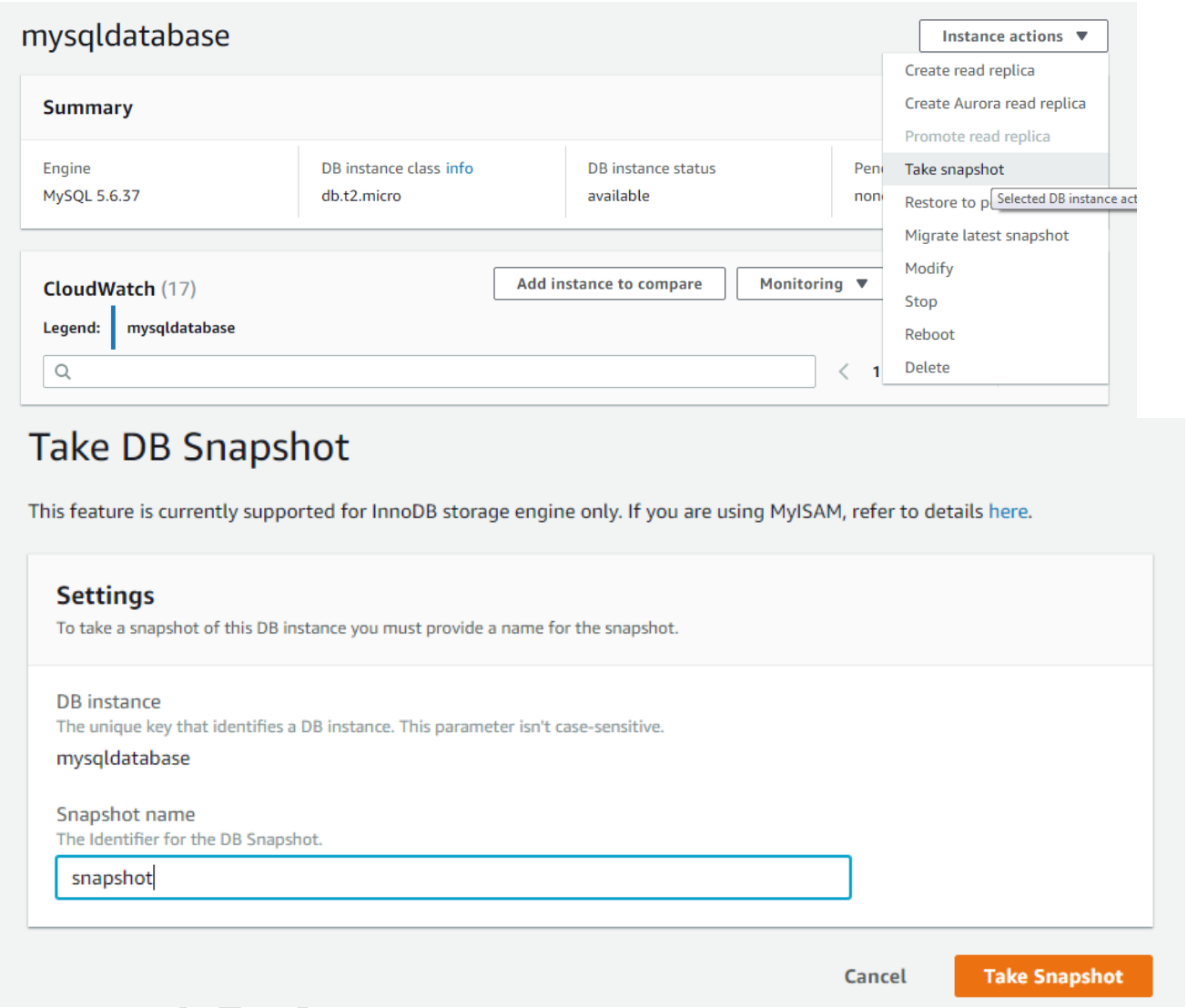
Delete: Db instance will delete. When you perform delete option AWS will ask you to create a final snapshot. If the data in the db is important, we can take a final snapshot to launch it in future, otherwise we can select No and delete the db instance.



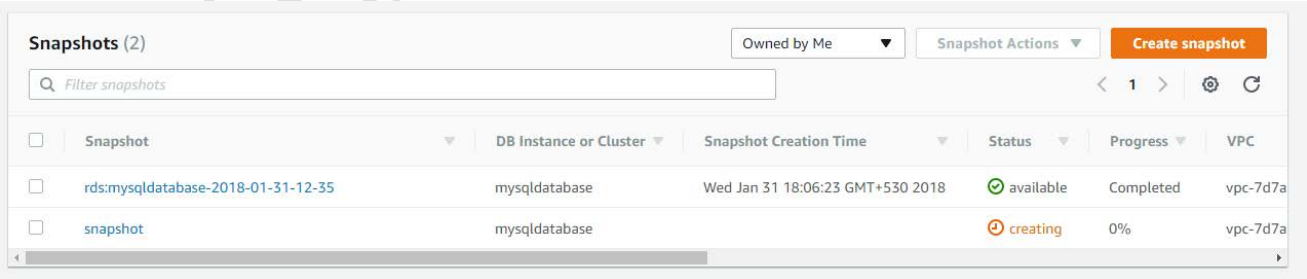
DB INSTANCE BACKUP OPTIONS:

As we discussed above, we have two options for backing up 1. Automated backup and 2. Manual Snapshots.

To create a manual snapshot, select the **"instance Actions"** and choose **"Take Snapshot"** Option.

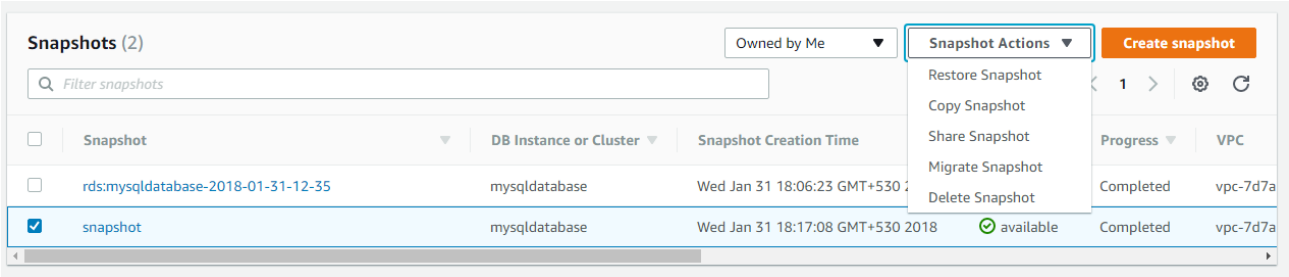


Give a name for the newly creating Snapshot and here is the status of Snapshot creation.

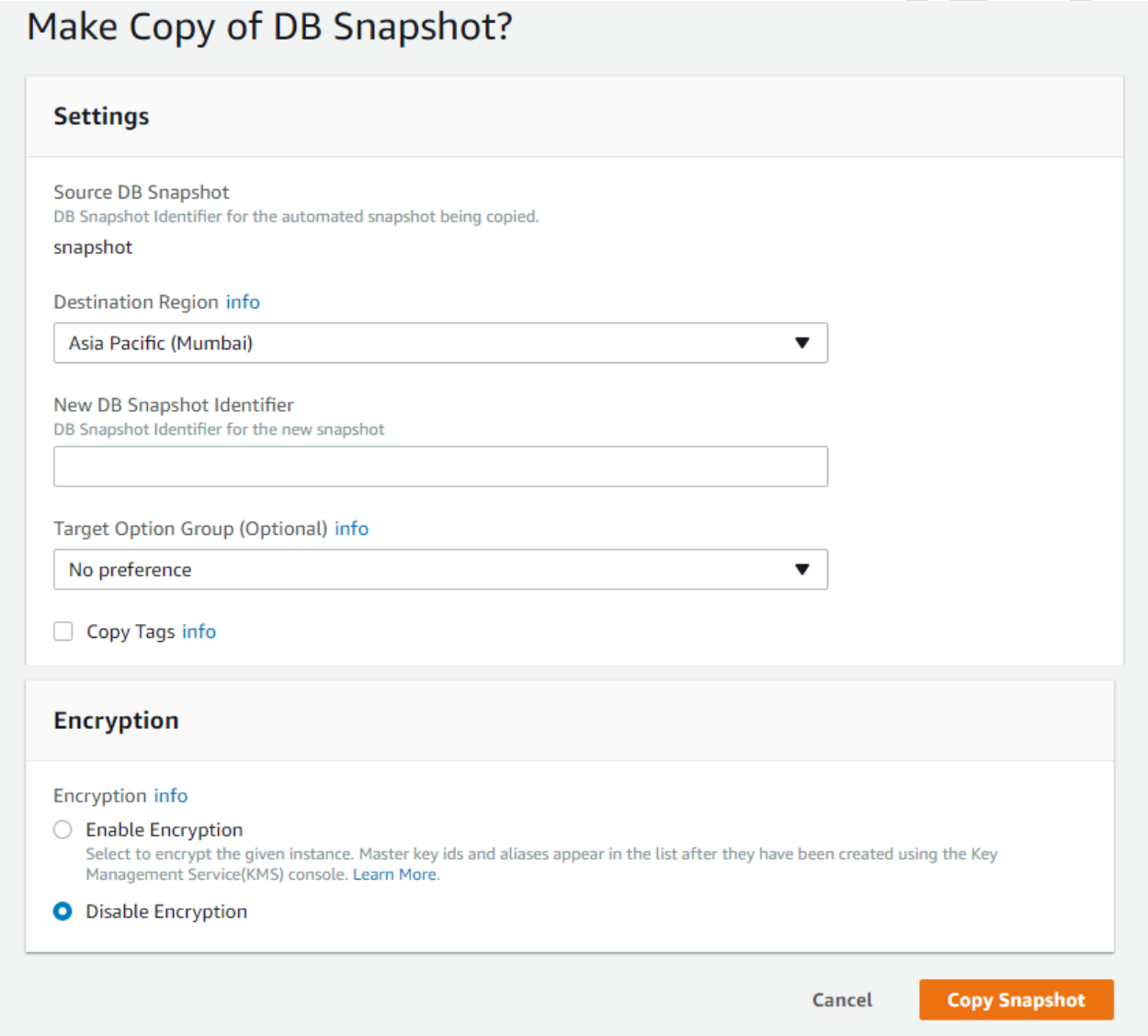


Launching Instance from the Snapshot.

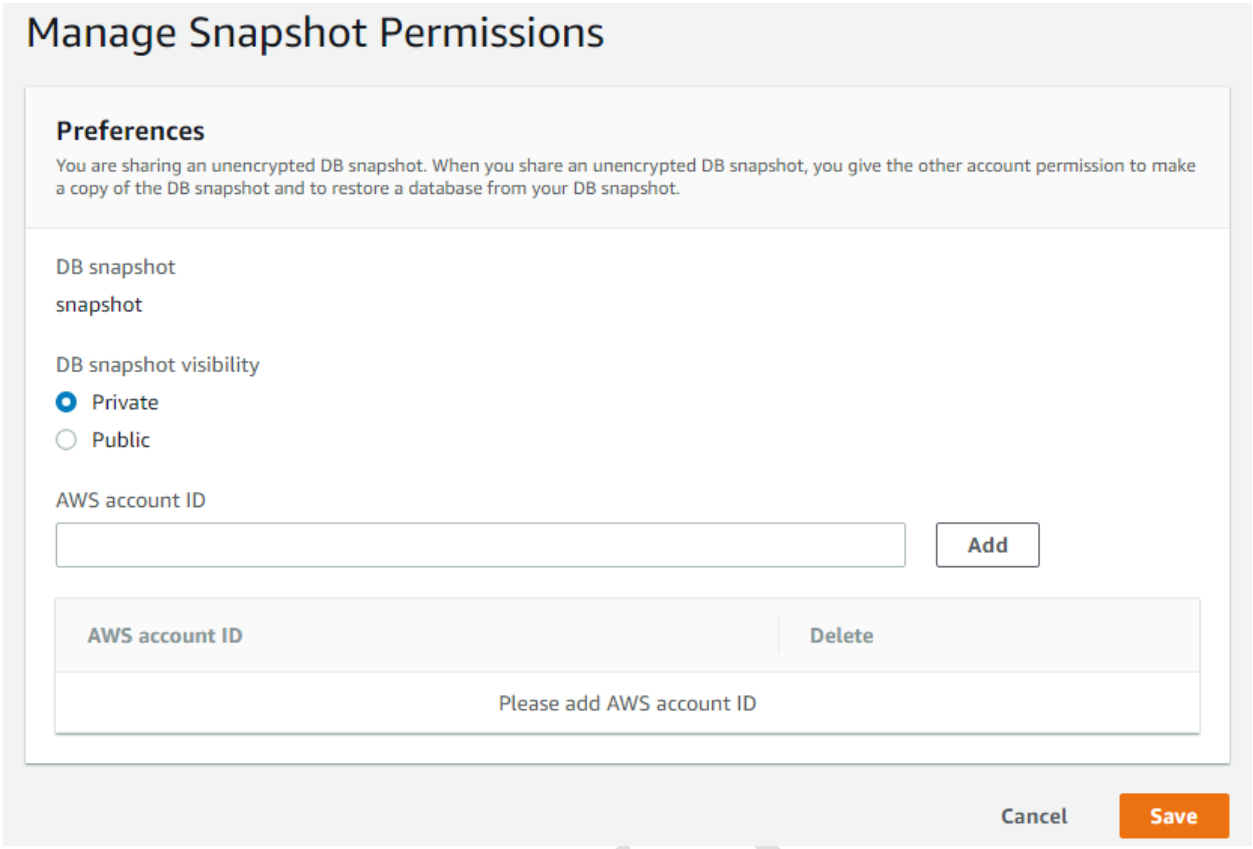
We can use either automated backups or manual snapshots to launch a new instance, but remember we'll get a new endpoint. To create a snapshot, select the snapshot and choose the "**Snapshot Actions"** and choose "**Restore Snapshot"**option, Then automatically an instance launch wizard will launch. You'll find almost all same as the regular instance launch.



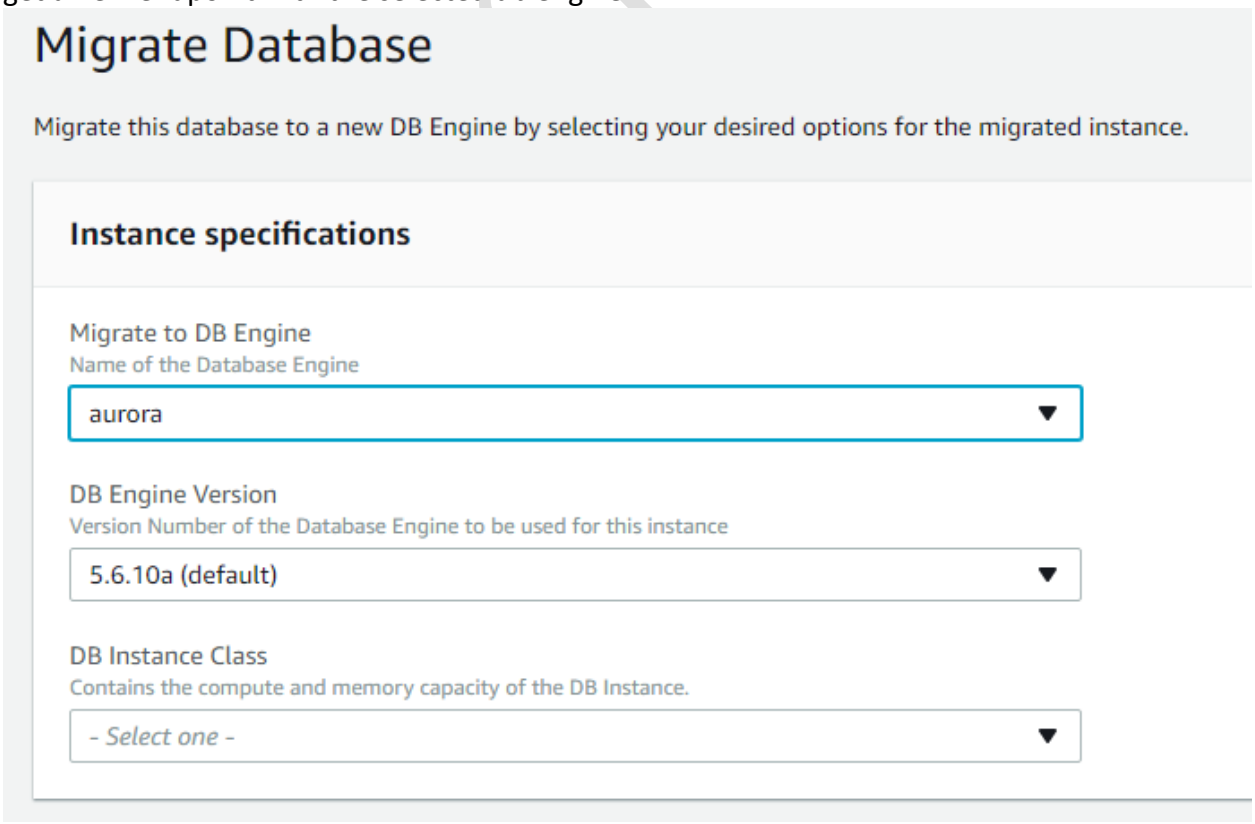
Copy Snapshot: We can make a copy the snapshot in another region. Choose the **"Destination region"**give a name for the new DB snapshot identifier, we can enable the encryption, if required while copying the snapshot.



Share Snapshot: We can share the snapshot with any other AWS account user or make it available for public by selecting the Share Snapshot option.



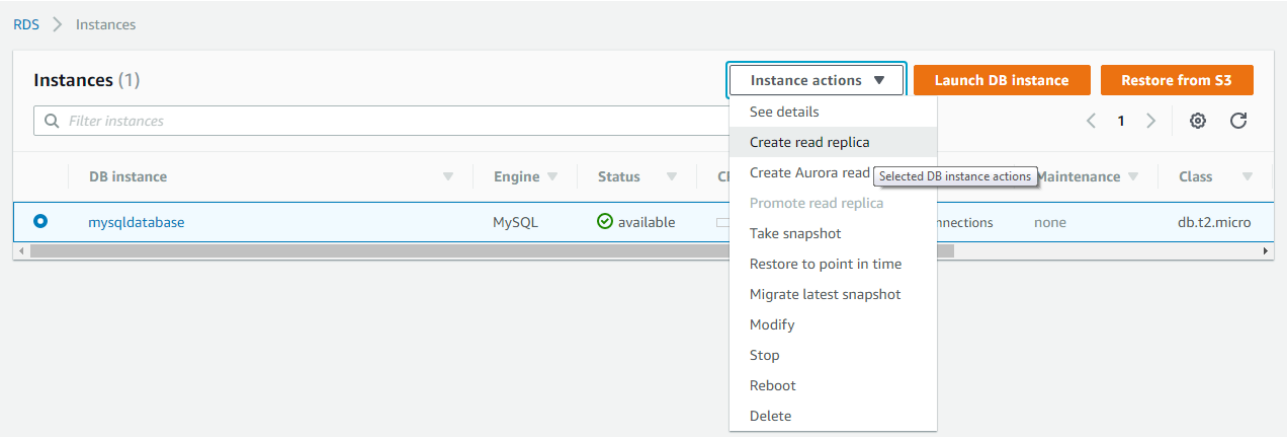
Migrate Snapshot: We can migrate the snapshot to a different db engine by using this option. Choose the Migrate snapshot option and select the **Aurora or Mariadb** and follow the wizard, we'll get a new endpoint with the selected db engine.

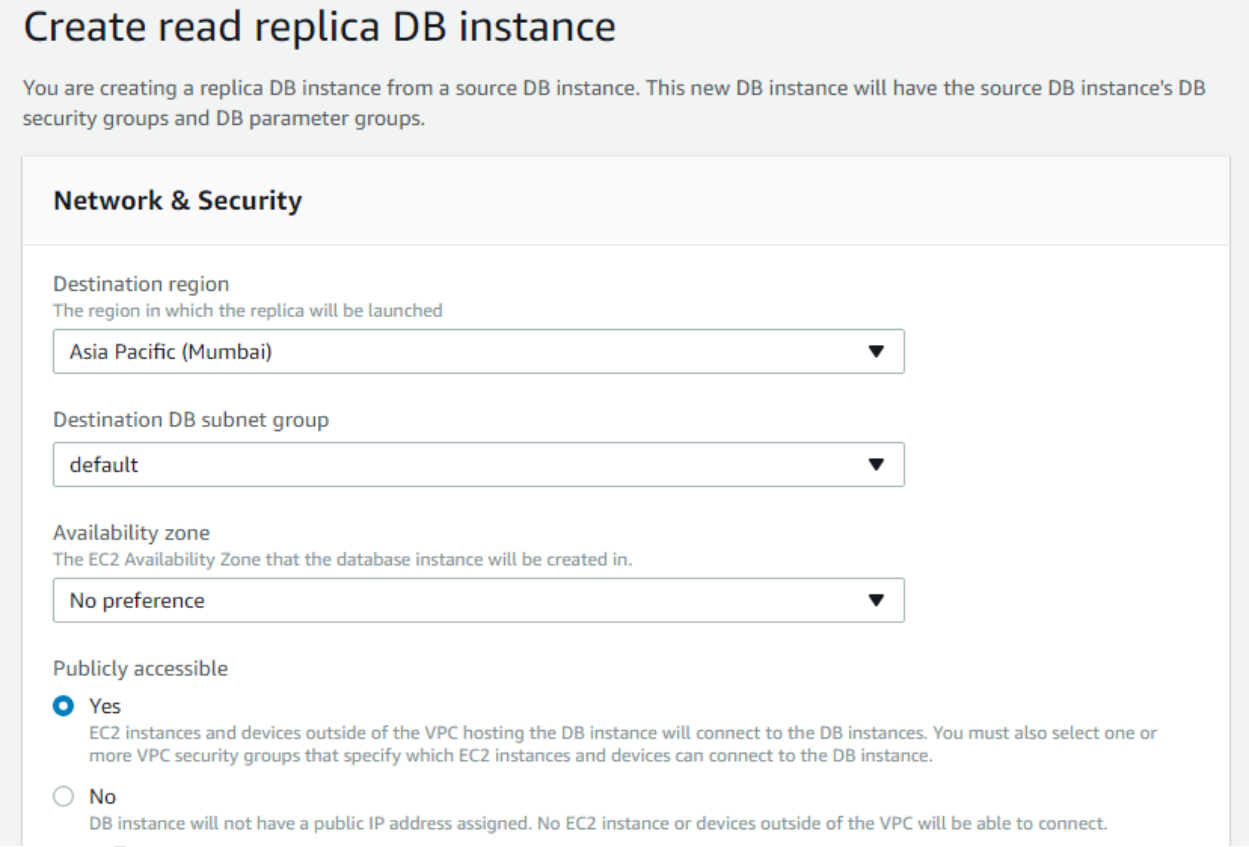


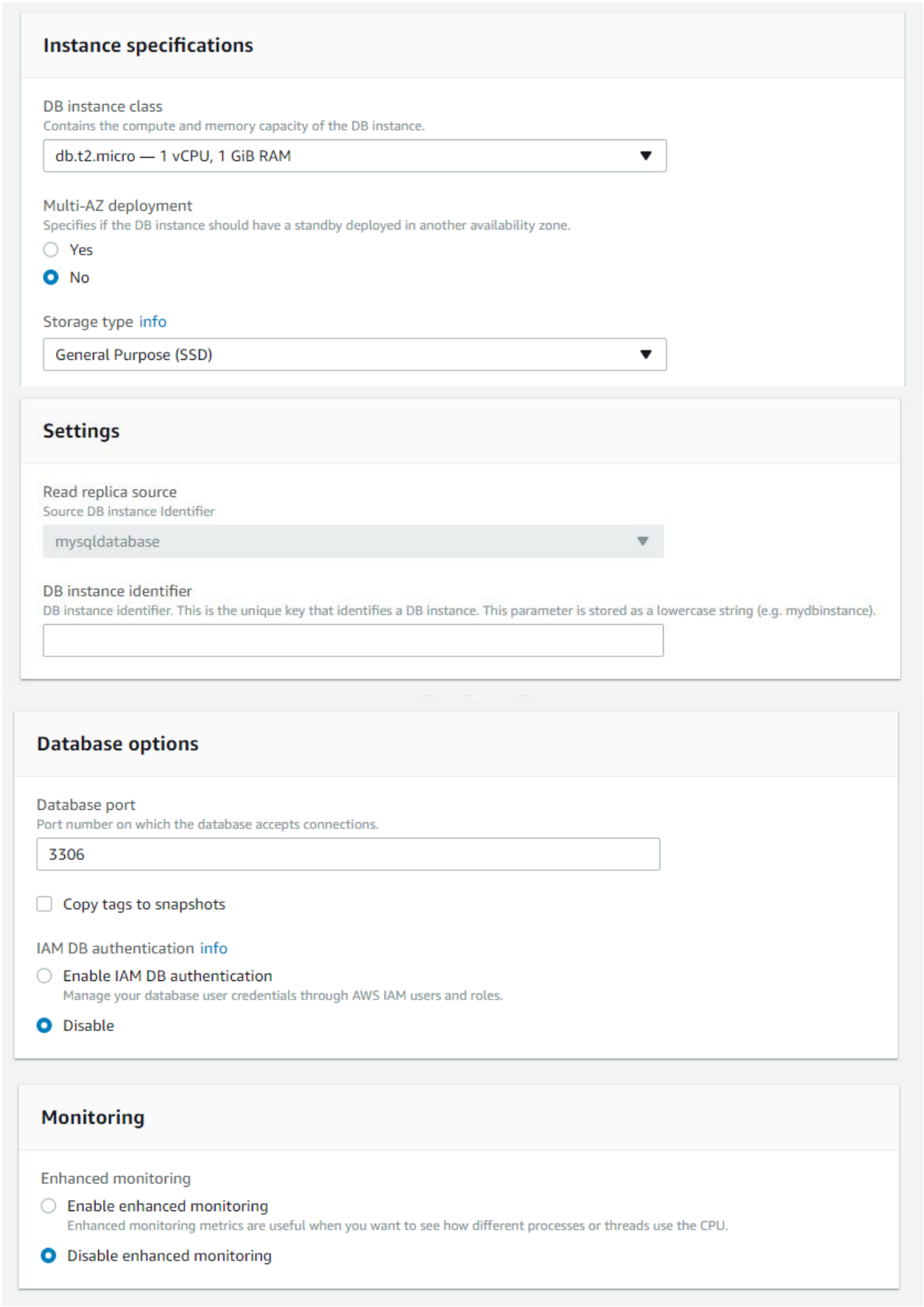
Creating Read Replicas and promoting them

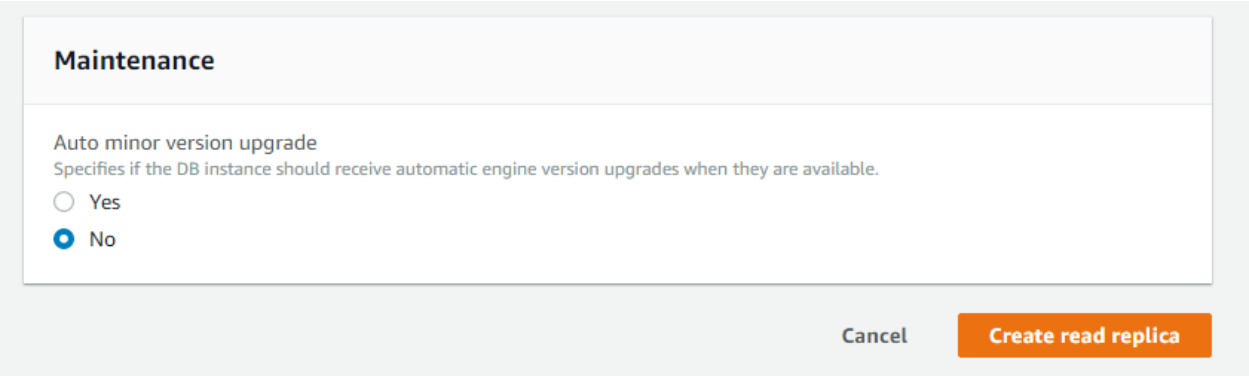
Read replica's allow you to have a read only copy of your production database. This is achieved by using Asynchronous replication from the primary RDS instance to the read replica. You use read replica's primarily for very read-heavy database workloads.

To create a read replica select the Instance Actions tab and select the Create ReadReplica option.



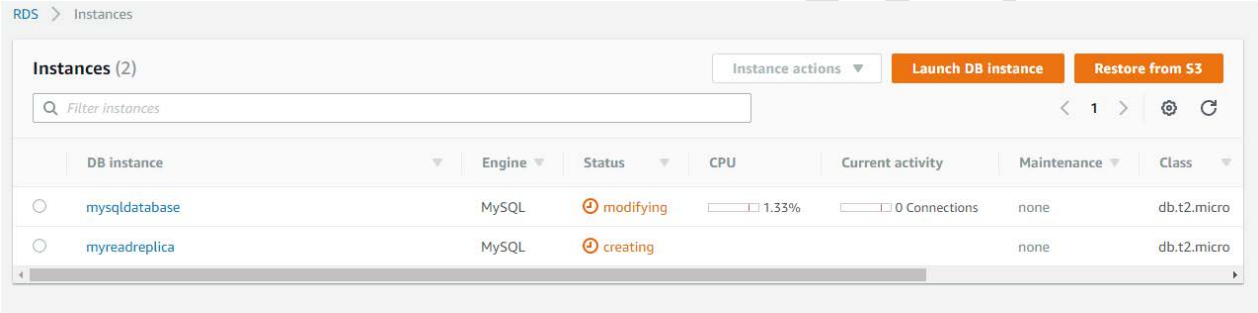



Instance specifications
DB instance class
Contains the compute and memory capacity of the DB instance.
db.t2.micro — 1 vCPU, 1 GiB RAM
Multi-AZ deployment
Specifies if the DB instance should have a standby deployed in another availability zone.
Yes
No
Storage type info
General Purpose (SSD)
Settings
Read replica source
Source DB instance Identifier
mysqldatabase
DB instance identifier
DB instance identifier. This is the unique key that identifies a DB instance. This parameter is stored as a lowercase string (e.g. mydbinstance).
Database options
Database port
Port number on which the database accepts connections.
3306
Copy tags to snapshots
IAM DB authentication info
Enable IAM DB authentication
Manage your database user credentials through AWS IAM users and roles.
Disable
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

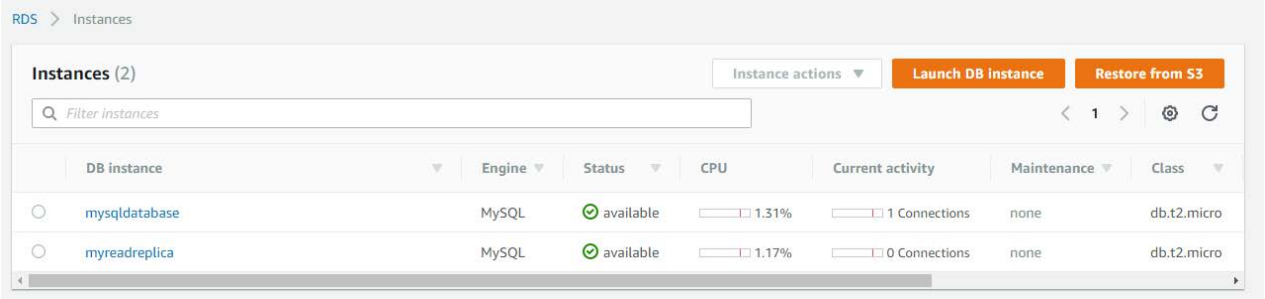


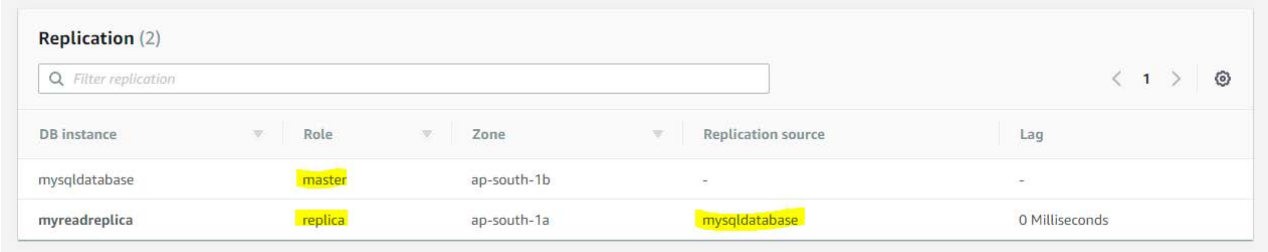
Select the Read replica source and give a name for the replica and choose in what availability zone you want to deploy and even we can select the desired availability zone in the destination region also.

Select the appropriate options and click on Create read replica option. Read replica creation will start and we can see the status in dashboard.



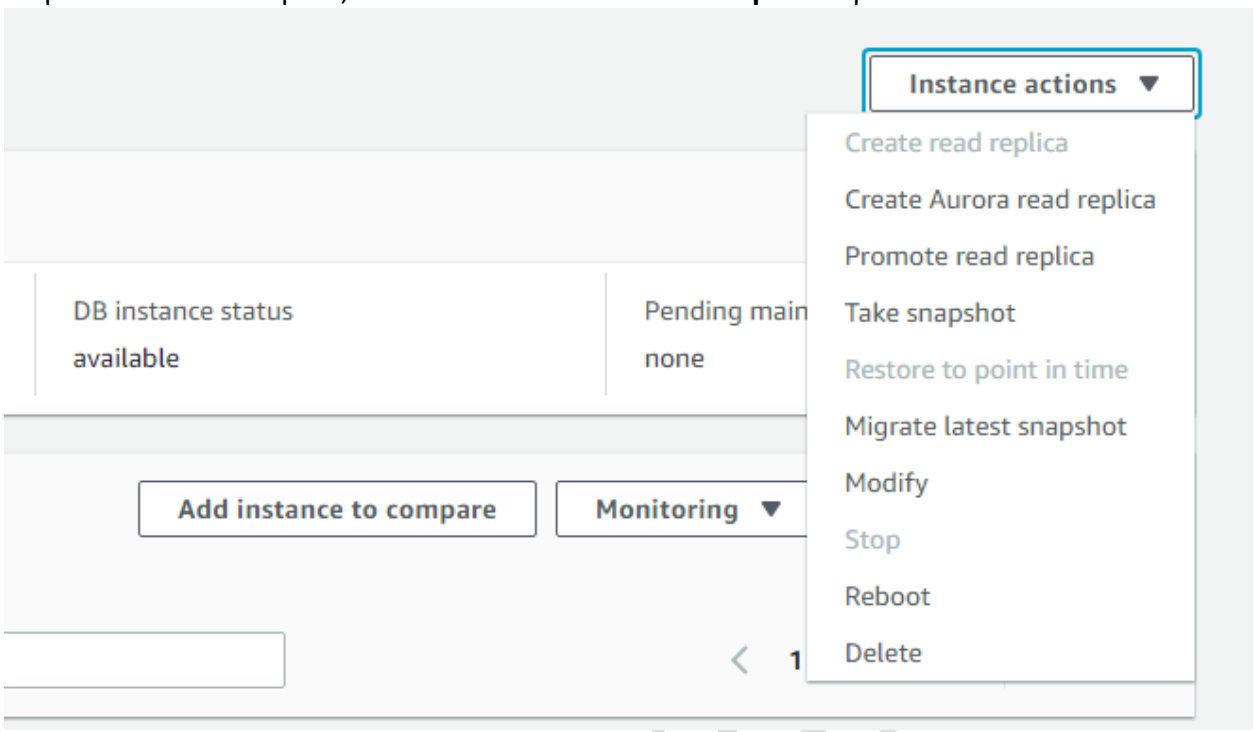
Now read replica is created. To verify the master and slave status we can go to details and verify.

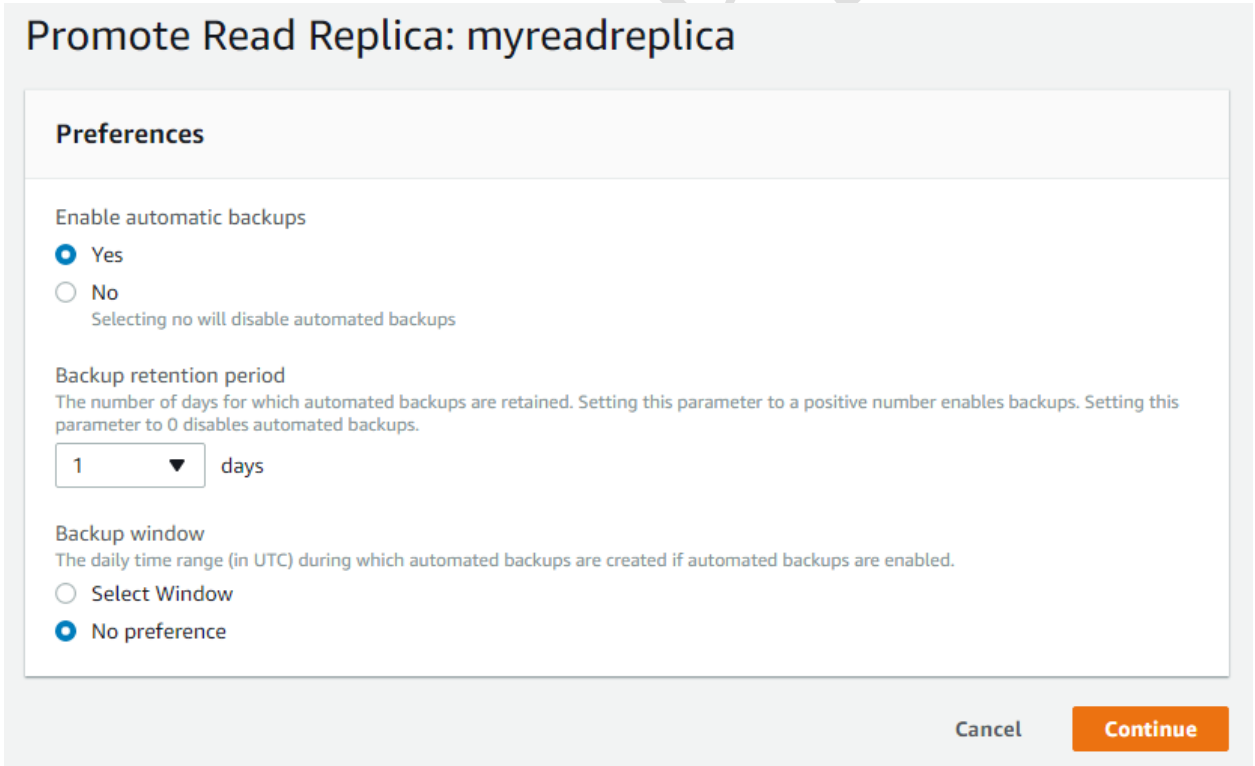




- And we can promote the read replica to a standalone db instance, but this breaks the replication.

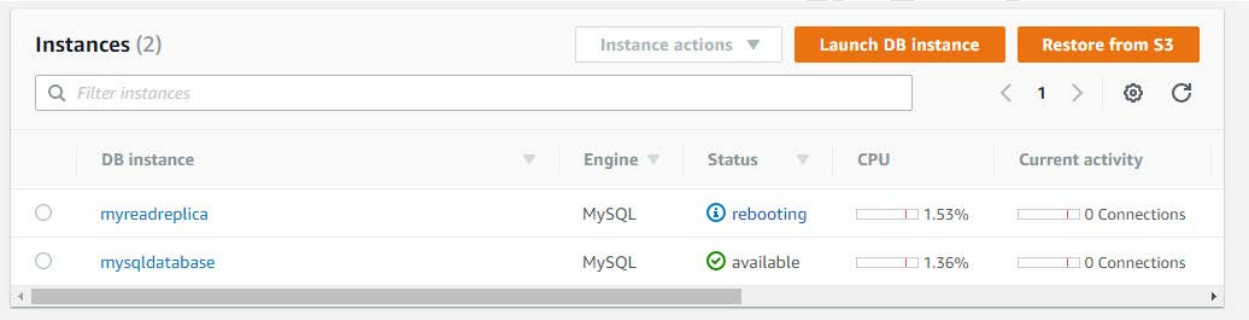
To promote a read replica, choose the **"Promote Read Replica"** option from "Instance Actions"



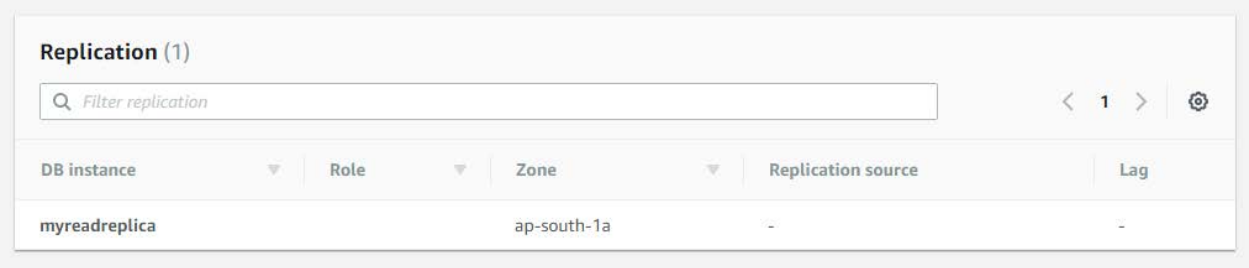


- If we want to promote a read replica, we must enable the automated backups and need to select the backup retention period, and you can select the backup window.

We will get a note with the following information:



Now the read replica is promoted as an individual db instance and no replication is enabled with any other db instances.



Amazon DynamoDB:

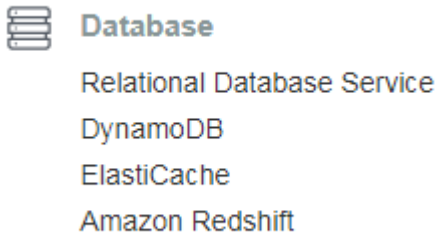
Amazon DynamoDB is a fully managed NoSQL database service that provides fast and lowlatencyperformance that scales with ease.Amazon DynamoDB significantly simplifies the hardware provisioning, setup andconfiguration, replication, software patching, and cluster scaling of NoSQL databases.

Amazon DynamoDB can provide consistent performance levels by automatically distributingthe data and traffic for a table over multiple partitions. After you configure a certain read orwrite capacity, Amazon DynamoDB will automatically add enough infrastructure capacity tosupport the requested throughput levels. As your demand changes over time, you can adjustthe read or write capacity after a table has been created, and Amazon DynamoDB will add orremove infrastructure and adjust the internal partitioning accordingly.

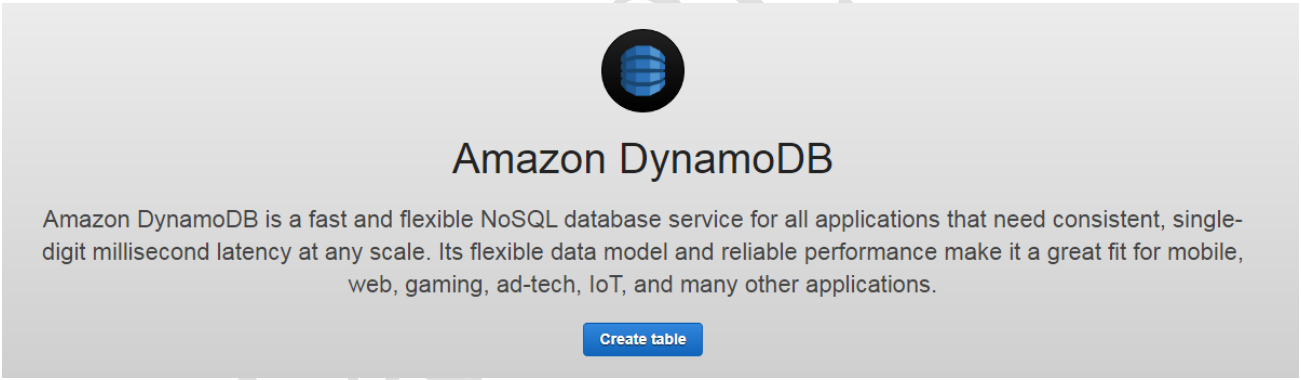
- All table data is stored on high performance SSD disk drives.
- Applications can connect to the Amazon DynamoDB service endpoint and submit requestsover HTTP/S to read and write items to a table or even to create and delete tables.

Provisioned Capacity: When you create an Amazon DynamoDB table, you are required to provision a certainamount of read and write capacity to handle your expected workloads.

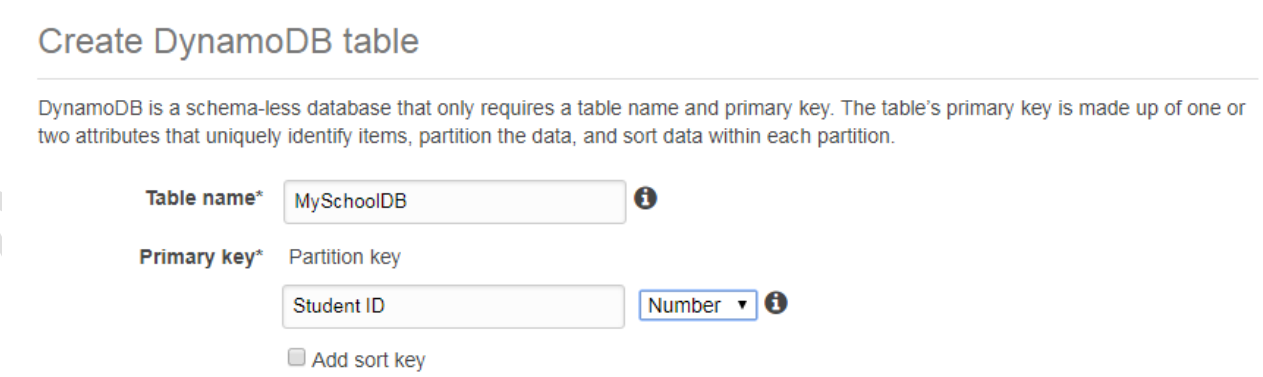
1. We can find Synamo DB under Database module



2. Choose “Create table” option to start creating tables in DynamoDB



3. Choose a Table Name and Primary key for the database table.



4. We choose default settings as mentioned below, or you can customize the setting by unchecking “use default settings” option

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".

Provisioned capacity

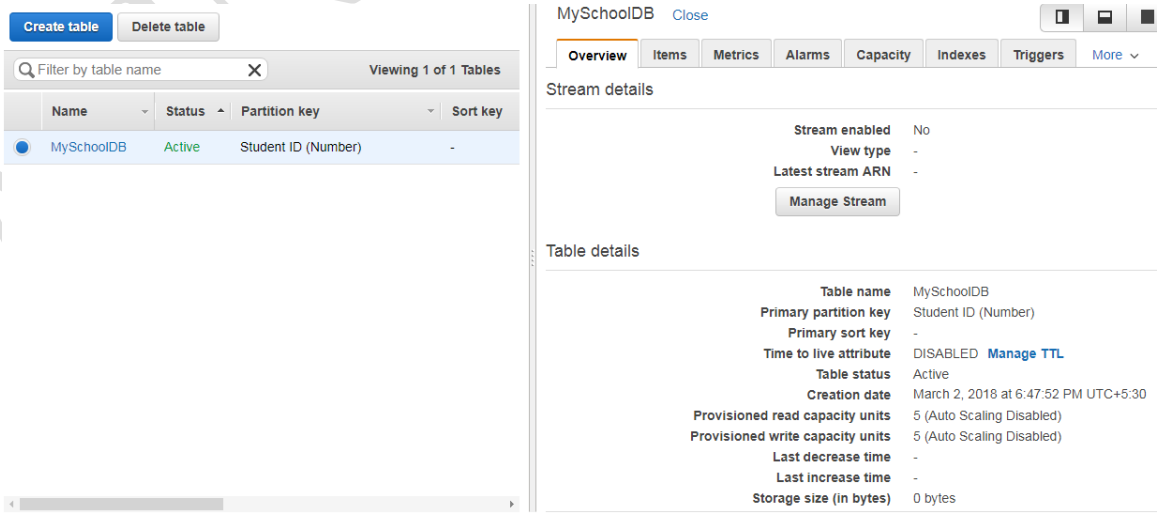
	Read capacity units	Write capacity units
Table	5	5
Estimated cost	$3.31 / month (Capacity calculator)	

Auto Scaling

	Read capacity	Write capacity
		Same settings as read
Target utilization	70 %	70 %
Minimum provisioned capacity	5 units	5 units
Maximum provisioned capacity	10000 units	10000 units
	Apply same settings to global secondary indexes	Apply same settings to global secondary indexes

5. If you don't want to enable auto scaling of DynamoDB, simply uncheck the "Read capacity" & "Write ccapacity" options.

6. As shown below, a table is created and you can navigate to **"Items"**and you can start adding items.



Amazon Redshift

Amazon Redshift is a fast, powerful, fully managed, petabyte-scale data warehouse service in the cloud. Amazon Redshift is a relational database designed for OLAP scenarios andoptimized for high-performance analysis and reporting of very large datasets. Traditional datawarehouses are difficult and expensive to manage, especially for large datasets. AmazonRedshift not only significantly lowers the cost of a data warehouse, but it also makes it easyto analyze large amounts of data very quickly.

Amazon Redshift gives you fast querying capabilities over structured data using standard SQLcommands to support interactive querying over large datasets. With connectivity via ODBC orJDBC, Amazon Redshift integrates well with various data loading, reporting, data mining, andanalytics tools. Amazon Redshift is based on industry-standard PostgreSQL, so most existingSQL client applications will work with only minimal changes.

Amazon Redshift manages the work needed to set up, operate, and scale a data warehouse,from provisioning the infrastructure capacity to automating ongoing administrative taskssuch as backups and patching. Amazon Redshift automatically monitors your nodes anddrives to help you recover from failures.

Clusters and Nodes

The key component of an Amazon Redshift data warehouse is a cluster. A cluster is composed of a leader node and one or more compute nodes. The client application interacts directlyonly with the leader node, and the compute nodes are transparent to external applications.

- Single Node (160Gb)
- Multi-Node
  - Leader Node (manages client connections and receives queries).
  - Compute Node (store data and perform queries and computations). Up to 128 Compute Nodes.

Elasticache

ElastiCache is a web service that makes it easy to deploy, operate, and scale an in-memory cache in the cloud. The service improves the performance of web applications by allowing you to retrieve information from fast, managed, in-memory caches, instead of relying entirely on slower disk-based databases.

Amazon ElastiCache can be used to significantly improve latency and throughput for many read-heavy application workloads or compute-intensive workloads.Caching improves application performance by storing critical pieces of data in memory for low-latency access.Cached information may include the results of I/O-intensive database queries or the results of computationally-intensive calculations.
Elasticache is a good choice if your database is particularly read heavy and not prone to frequent changing.
Memcached: High-performance, distributed memory object caching system, intended for use in speeding up dynamic web applications.

Redis: A popular open-source in-memory key-value store that supports data structures such as sorted sets and lists. ElastiCache supports Master / Slave replication and Multi-AZ which can be used to achieve cross AZ redundancy.

Virtual Private Cloud (Amazon VPC)

Introduction To VPC

The Amazon Virtual Private Cloud (Amazon VPC) is a custom-defined virtual networkwithin the AWS Cloud. You can provision your own logically isolated section of AWS, similarto designing and implementing a separate independent network that would operate in an onpremisesdata center. Amazon VPC is the networking layer for Amazon Elastic Compute Cloud (Amazon EC2), andit allows you to build your own virtual network within AWS.

You will have complete control over your virtual networking environment, including selection of your own IP address range, creation of subnets, and configuration of route tables and network gateways.

You can easily customize the network configuration for your Amazon Virtual Private Cloud.
For example, you can create a public-facing subnet for your webservers that has access to the Internet, and place your backend systems such as databases or application servers in a private-facing subnet with no Internet access.
VPCs also have a few limits set on them by default. For example, **you can have a maximum of five**

VPCs per region. Each VPC can have a max of one Internetgateway as well as one virtual private gateway. Also, **each VPC has a limit of hosting a maximum of up to 200 subnets per VPC**. You can increase these limit by simplyrequesting AWS to do so.

An Amazon VPC consists of the following components:

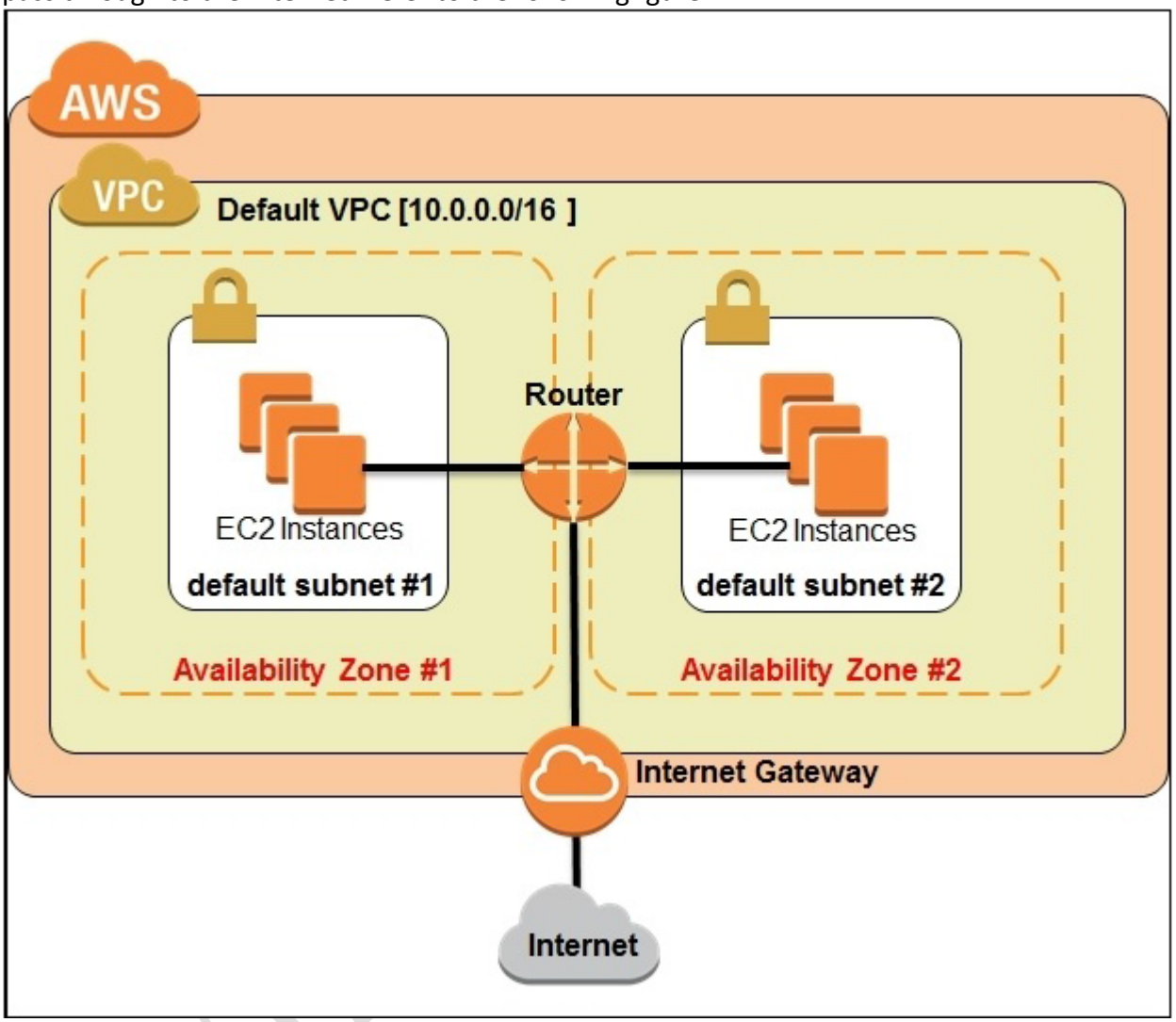
- Subnets
- Route tables
- Dynamic Host Configuration Protocol (DHCP) option sets
- Security groups
- Network Access Control Lists (ACLs)

An Amazon VPC has the following optional components:

- Internet Gateways (IGWs)
- Elastic IP (EIP) addresses
- Elastic Network Interfaces (ENIs)
- Endpoints
- Peering
- Network Address Translation (NATs) instances and NAT gateways
- Virtual Private Gateway (VPG), Customer Gateways (CGWs), and Virtual PrivateNetworks (VPNs)

By default, AWS will create a VPCfor you in your particular region the first time you sign up for the service. This is called asthe default VPC. The default VPC comes preconfigured with the following set ofconfigurations:

The default VPC is always created with a CIDR block of /16, which means itsupports 65,536 IP addresses in it.A default subnet is created in each AZ of your selected region. Instances launched in these default subnets have both a public and a private IP address by default as well.An Internet Gateway is provided to the default VPC for instances to have Internetconnectivity.A few necessary route tables, security groups, and ACLs are also created by defaultthat enable the instance traffic to pass through to the Internet. Refer to the followingfigure:



Classless Inter-Domain Routing (CIDR): When you create an Amazon VPC, you must specify the IPv4 address range by choosing aClassless Inter-Domain Routing (CIDR) block, such as 10.0.0.0/16. The address range of theAmazon VPC cannot be changed after the Amazon VPC is created. An Amazon VPC addressrange may be as large as /16 (65,536 available addresses) or as small as /28 (16 availableaddresses) and should not overlap any other network with which they are to be connected.

Subnets: A subnet is a segment of an Amazon VPC's IP address range where you can launch AmazonEC2 instances, Amazon Relational Database Service (Amazon RDS) databases, and otherAWS resources.

After creating an Amazon VPC, you can add one or more subnets in each Availability Zone.Subnets reside within one Availability Zone and cannot span zones.

- Remember that one subnet equals one Availability Zone. You can, however, have multiple subnets in one Availability Zone.

Subnets can be classified as public, private, or VPN-only

A **public subnet** is one in which the associated route table directs the subnet's traffic to the Amazon VPC's IGW.

A **private subnet** is one in which the associated route table does not direct the subnet's traffic to the Amazon VPC's IGW.

A **VPN-only subnet** is one in which theassociated route table directs the subnet's traffic to the Amazon VPC's VPGand does not have a route to the IGW.

Route Tables:

A route table is a logical construct within an Amazon VPC that contains a set of rules (called routes) that are applied to the subnet and used to determine where network traffic isdirected.

- You can modify route tables and add your own custom routes.
- You can also use route tables to specify which subnets are public (by directing Internet traffic to the IGW) and which subnets are private (by not having a route that directs traffic to the IGW).
- Each route table contains a default route called the local route, which enablescommunication within the Amazon VPC, and this route cannot be modified or removed.
- Additional routes can be added to direct traffic to exit the Amazon VPC via the IGW, the VPG, or the NAT instance.

You should remember the following points about route tables:

- Your VPC has an implicit router.
- Your VPC automatically comes with a main route table that you can modify.
- You can create additional custom route tables for your VPC.
- Each subnet must be associated with a route table, which controls the routing for the subnet. If you don't explicitly associate a subnet with a particular route table, the subnetuses the main route table.
- You can replace the main route table with a custom table that you've created so that eachnew subnet is automatically associated with it.

Internet Gateways:

An Internet Gateway (IGW) is a horizontally scaled, redundant, and highly available AmazonVPC component that allows communication between instances in your Amazon VPC and theInternet.

Amazon EC2 instances within an Amazon VPC are only aware of their private IP addresses.When traffic is sent from the instance to the Internet, the IGW translates the reply address tothe instance's public IP address (or EIP address, covered later) and maintains the one-to-onemap of the instance private IP address and public IP address.

When an instance receivestraffic from the Internet, the IGW translates the destination address (public IP address) tothe instance's private IP address and forwards the traffic to the Amazon VPC.

You must do the following to create a public subnet with Internet access:

- Attach an IGW to your Amazon VPC.
- Create a subnet route table rule to send all non-local traffic (0.0.0.0/0) to the IGW.
- Configure your network ACLs and security group rules to allow relevant traffic to flow to and from your instance.

Elastic IP Addresses (EIP): An Elastic IP Addresses (EIP) is a static,public IP address in the pool for the region that you can allocate to your account (pull fromthe pool) and release (return to the pool).
AWS maintains a pool of public IP addresses in each region and makes them available for youto associate to resources within your Amazon VPCs.

- EIPs are specific to a region (that is, an EIP in one region cannot be assigned to an instance within an Amazon VPC in a different region).
- There is a one-to-one relationship between network interfaces and EIPs.
- You can move EIPs from one instance to another, either in the same Amazon VPC or a different Amazon VPC within the same region.
- EIPs remain associated with your AWS account until you explicitly release them.
- There are charges for EIPs allocated to your account, even when they are not associated with a resource.

Peering:
An Amazon VPC peering connection is a networking connection between two Amazon VPCsthat enables instances in either Amazon VPC to communicate with each other as if they arewithin the same network. You can create an Amazon VPC peering connection between yourown Amazon VPCs or with an Amazon VPC in another AWS account within a single region.
An Amazon VPC may have multiple peering connections, and peering is a one-to-onerelationship between Amazon VPCs, meaning two Amazon VPCs cannot have two peeringagreements between them.

Peering connections are created through a request/accept protocol. The owner of therequesting Amazon VPC sends a request to peer to the owner of the peer Amazon VPC. If thepeer Amazon VPC is within the same account, it is identified by its VPC ID. If the peer VPC iswithin a different account, it is identified by Account ID and VPC ID. The owner of the peerAmazon VPC has one week to accept or reject the request to peer with the requesting AmazonVPC before the peering request expires.

- You cannot create a peering connection between Amazon VPCs that have matching or overlapping CIDR blocks.
- You cannot create a peering connection between Amazon VPCs in different regions.
- Amazon VPC peering connections do not support transitive routing.
- You cannot have more than one peering connection between the same two Amazon VPCs at the same time.

Network Access Control Lists (ACLs):
A network access control list (ACL) is another layer of security that acts as a stateless firewall on a subnet level.
A network ACL is a numbered list of rules that AWS evaluates in order, starting with the lowest numbered rule, to determine whether traffic is allowed in or out ofany subnet associated with the network ACL. Here is a small example of how ACL looks like.

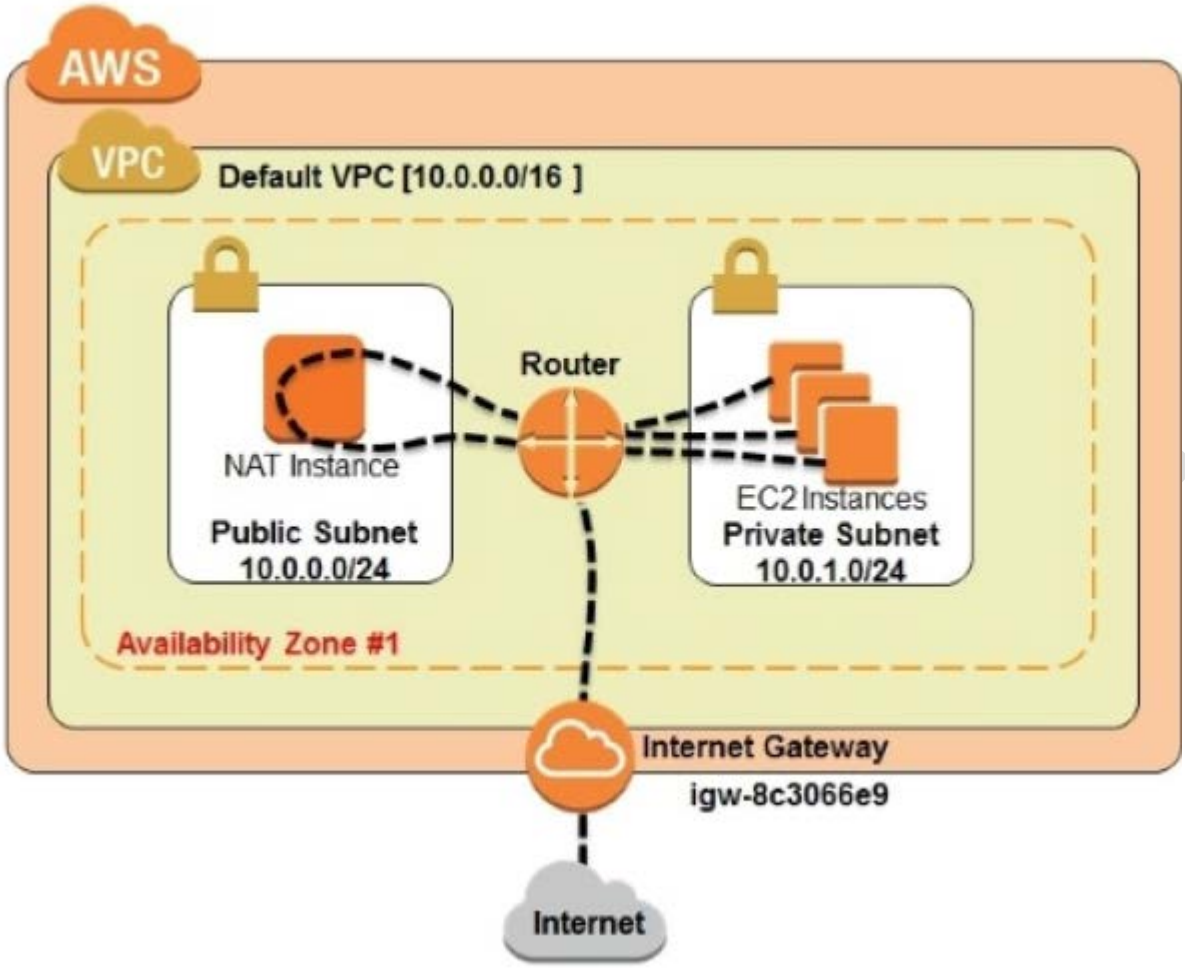
Inbound ACL rules				
Rule No.	**Source IP**	**Protocol**	**Port**	**Allow/Deny**
100	0.0.0.0/0	All	All	ALLOW
*	0.0.0.0/0	All	All	DENY
Outbound ACL rules				
Rule No.	**Dest IP**	**Protocol**	**Port**	**Allow/Deny**
100	0.0.0.0/0	all	all	ALLOW
*	0.0.0.0/0	all	all	DENY

When you create a custom network ACL, its initial configuration will deny all inboundand outbound traffic until you create rules that allow otherwise.

Security Group	**Network ACL**
Operates at the instance level (first layer of defense)	Operates at the subnet level (second layer of defense)
Supports allow rules only	Supports allow rules and deny rules
Stateful: Return traffic is automatically allowed, regardless of any rules	**Stateless:** Return traffic must be explicitly allowed by rules.
AWS evaluates all rules before deciding whether to allow traffic	AWS processes rules in number order when deciding whether toallow traffic.
Applied selectively to individual instances	Automatically applied to all instances in the associated subnets; this is a backup layer of defense, so you don't have to rely onsomeone specifying the security group.

Network Address Translation (NAT) Instances and NAT Gateways

By default, any instance that you launch into a private subnet in an Amazon VPC is not ableto communicate with the Internet through the IGW.AWS provides NATinstances and NAT gateways to allow instances deployed in private subnets to gain Internetaccess.



NAT Instance:A network address translation (NAT) instance is an Amazon Linux Amazon Machine Image(AMI) that is designed to accept traffic from instances within a private subnet, translate thesource IP address to the public IP address of the NAT instance, and forward the traffic to theIGW.

NAT Instances allows in private subnets to send outbound Internetcommunication, but it prevents the instances from receiving inbound traffic initiated bysomeone on the Internet.

- Create a security group for the NAT with outbound rules that specify the needed Internet resources by port, protocol, and IP address.
- Launch an Amazon Linux NAT AMI as an instance in a public subnet and associate it with the NAT security group.
- Disable the Source/Destination Check attribute of the NAT.
- Configure the route table associated with a private subnet to direct Internet-bound traffic to the NAT instance (for example, i-1a2b3c4d).

NAT Gateway: A NAT gateway is an Amazon managed resource that is designed to operate just like a NAT instance, but it is simpler to manage and highly available within an Availability Zone.

- Allocate an EIP and associate it with the NAT gateway.
- Configure the route table associated with the private subnet to direct Internet-boundtraffic to the NAT gateway.

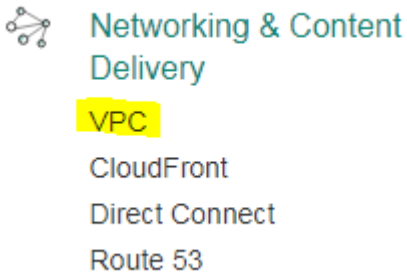
You can connect an existing data center to Amazon VPC using either hardware or softwareVPN connections, which will make Amazon VPC an extension of the data center. Amazon VPCoffers two ways to connect a corporate network to a VPC: VPG and CGW.

A virtual private gateway:VPG is the virtual private network (VPN) concentrator on theAWS side of the VPN connection between the two networks.

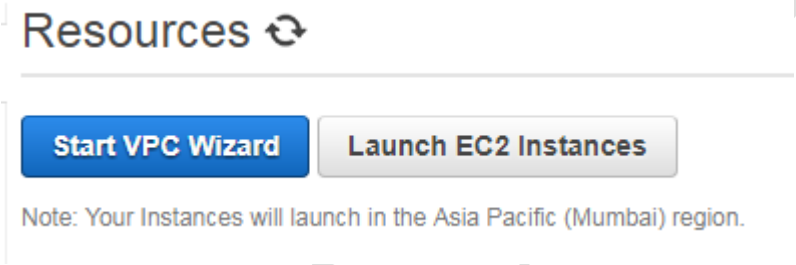
A customer gateway (CGW)represents a physical device or a software application on the customer's side of the VPNconnection.

VPC deployment options:

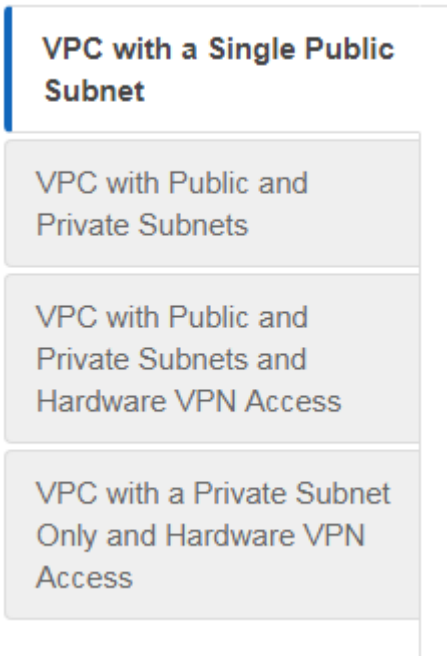
1. You can find VPC under Network & Content Delivery category in AWS console. Select VPC.



2. You can select the **Start VPC Wizard**option to get all the the VPC deployment methods.



3. We have 4 deployment models available currently with AWS VPC. Detailed description given below.



VPC with a single public subnet: This is by far the simplest of the four deploymentscenarios. Using this scenario, we will get a**VPC will provision a single public subnet with a default Internet**

Gateway attached to it. The subnet will also have a few simple andbasic route tables, security groups, and network ACLs created. This type ofdeployment is ideal for small-scaled web applications or simple websites that don'trequire any separate application or subnet tiers.

VPC with public and private subnets (NAT): This is the most commonly useddeployment scenario, this option will provide you with **a public subnet and a private subnet** as well. The public subnet will be connected to an Internet gateway and allowinstances launched within it to have Internet connectivity, whereas the private subnetwill not have any access to the outside world. This scenario will also provision asingle NAT instance inside the public subnet using which your private subnetinstances can connect with the outside world but not vice versa. Besides this, thewizard will also create and assign a route table to both the public and private subnets,each with the necessary routing information prefilled in them. This type ofdeployment is ideal for large-scale web applications and websites that leverage a mixof public facing (web servers) and non-public facing (database servers).

VPC with public and private subnets and hardware VPN access: This deploymentscenario is very much similar to the VPC with public and private subnets, however, withone component added additionally, which is the Virtual Private Gateway. This VirtualPrivate Gateway connects to your on premise network's gateway using a standard VPNconnection. This type of deployment is well suited for organizations that wish to extendtheir on premise datacenters and networks in to the public clouds while allowing theirinstances to communicate with the Internet.

VPC with a private subnet only and hardware VPN access: Unlike the previousdeployment scenario, this scenario only provides you with a private subnet that canconnect to your on premise datacenters using standard VPN connections. There is noInternet Gateway provided and thus your instances remain isolated from the Internet.This deployment scenario is ideal for cases where you wish to extend your onpremise datacenters into the public cloud but do not wish your instances to have anycommunication with the outside world.

Here is a simple use case for creating Custom VPC

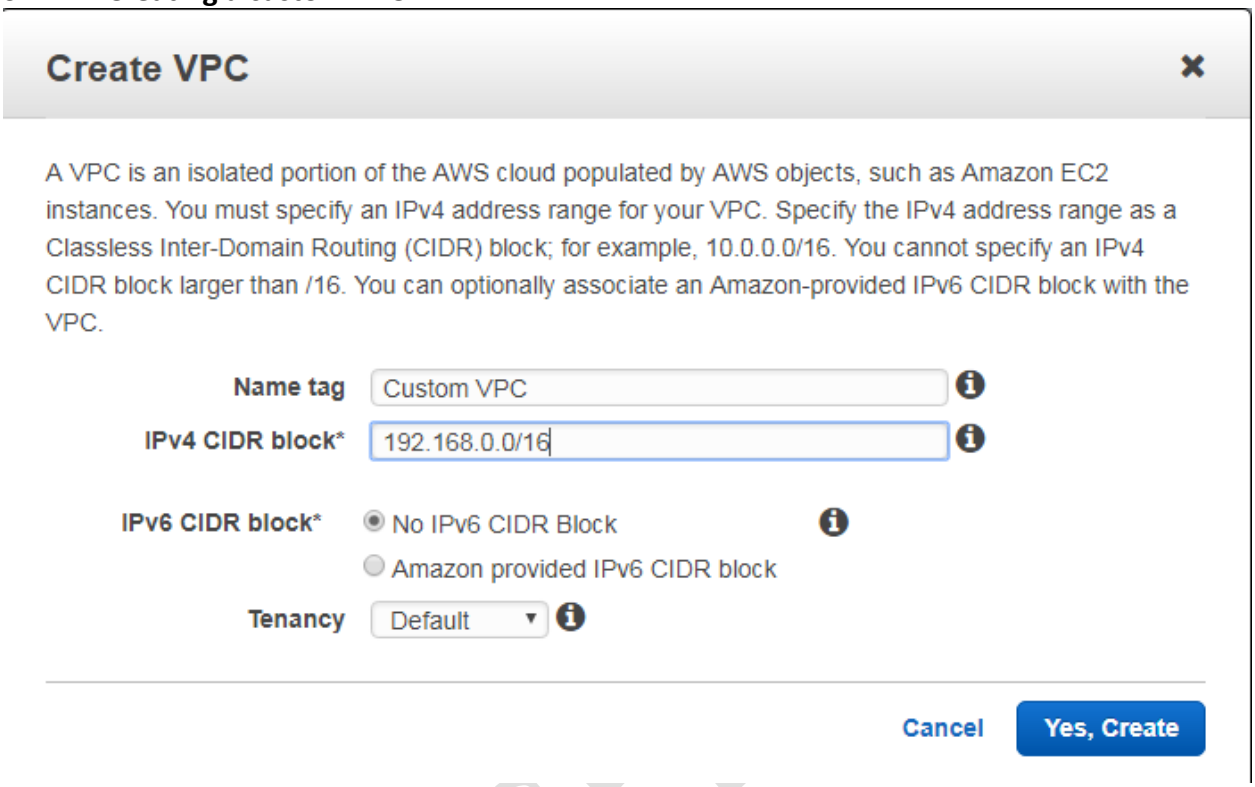
- Create a VPC (AP-SOUTH-PROD-1 - 192.168.0.0/16) with separate secure environments for hosting the web servers and database servers.
- Only the web server environment (AP-SOUTH-PROD-WEB - 192.168.1.0/24) should have direct Internet access.
- The database server environment (AP-SOUTH-PROD-DB - 192.168.2.0/24) should be isolated from any direct access from the outside world.
- The database servers can have restricted Internet access only through a jump server (NAT Instance). The jump server needs to be a part of the web server environment.

You can follow the simple wizard, but to understand the flow clearly am going to create and configure each and every option manually. Here is the steps am going to perform.

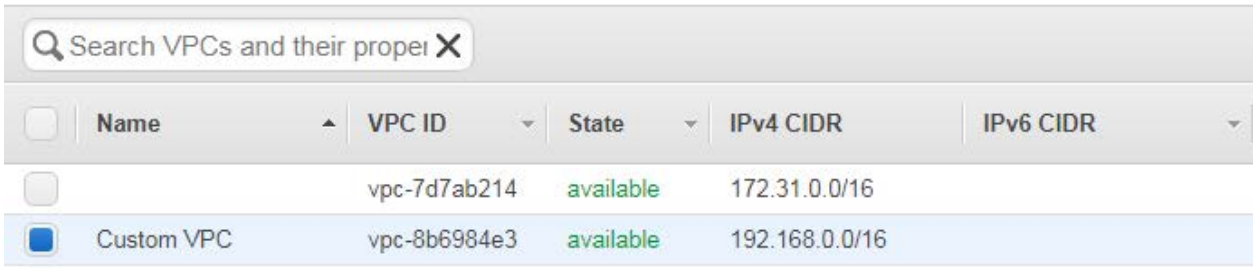
- Creating a Custom VPC
- Creating Subnets under Custom VPC
- Creating IGW and associating with VPC

- Creating a Route table and performing subnet association
- Launching instance in Public subnet and private subnet

STEP 1: Creating a custom VPC

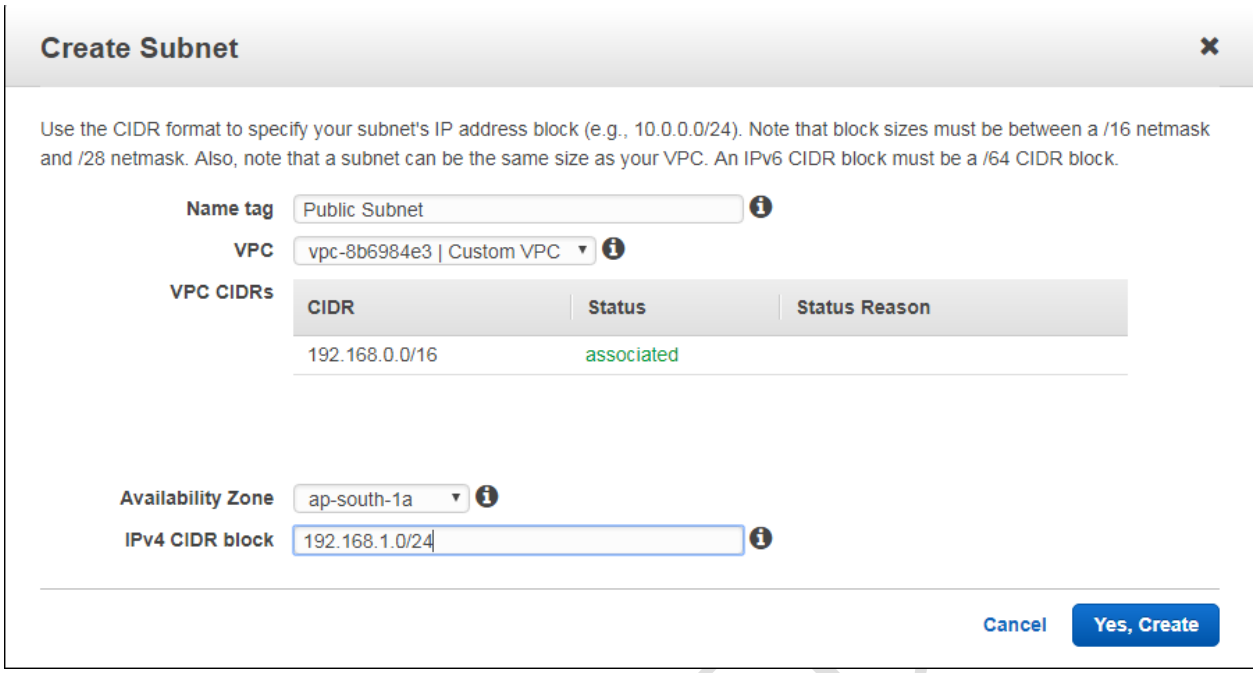


- As mentioned in above image, am creating a VPC with **CustomVPC** name and selecting CIDR block in Class C IP address range **192.168.0.0/16**(provide a /16 subnet will provide us65,531 IP addresses to use) and selecting tenancy as **Default.**

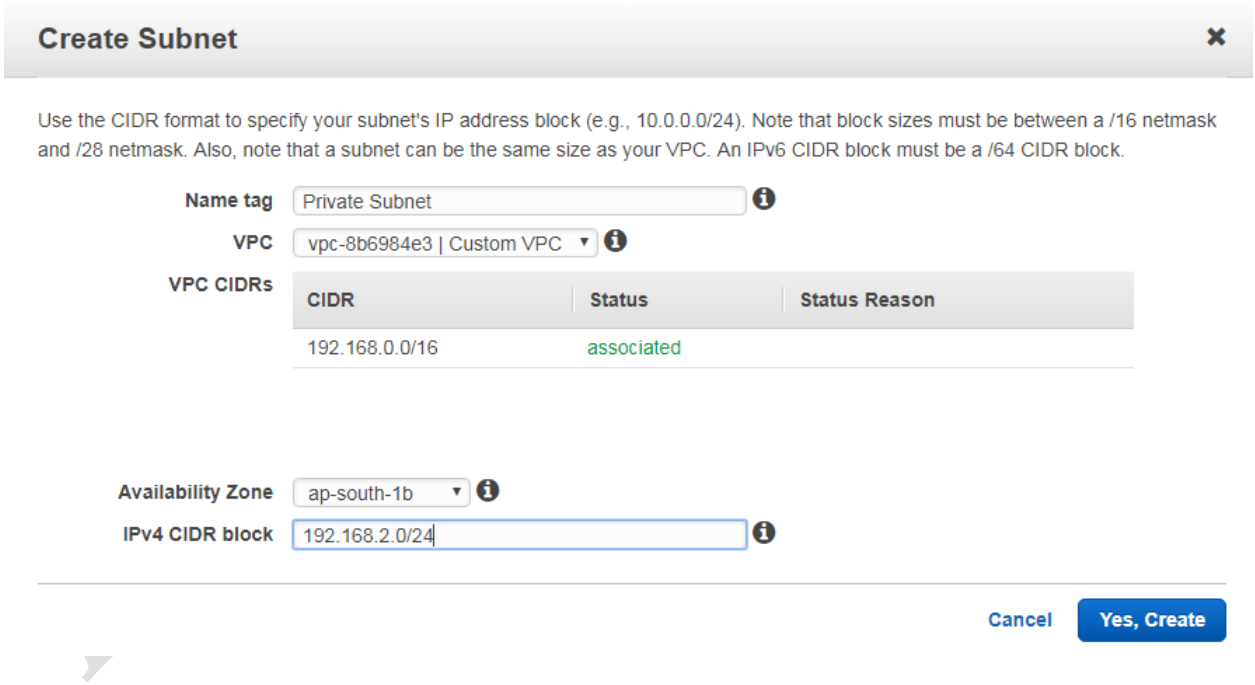


STEP 2: Creating a subnets under custom VPC (One public and one private subnets)

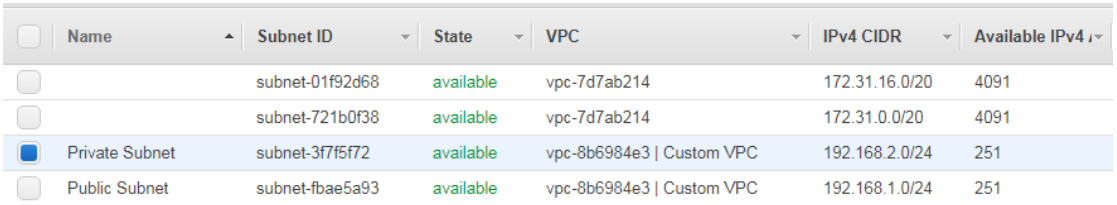
- Navigating to Subnets option and selecting **"Creating Subnet"**and giving name as **"Public Subnet"** where I want to deploy my Internet Facing instances.
- Creating this Subnet under Custom VPC, Select that option and select the **ap-south-1a** Availability Zone , Given a CIDR block as 192.168.1.0/24 (all instances launched under ap-south-1a will get the same range Private IP addresses and we'll get 251 usable IP addresses) and click on Create. Remember again, one subnet is equal to one AZ.



- Now creating another subnet and naming it as **"Private Subnet"** and want to deploy the instance which doesn't required internet faced.
- Creating this subnet under Custom VPC, and named as "Private Subnet" then provided CIDR as 192.168.2.0/24 and selecting Avaiablility Zone as **ap-south-1b**and click on Create option.



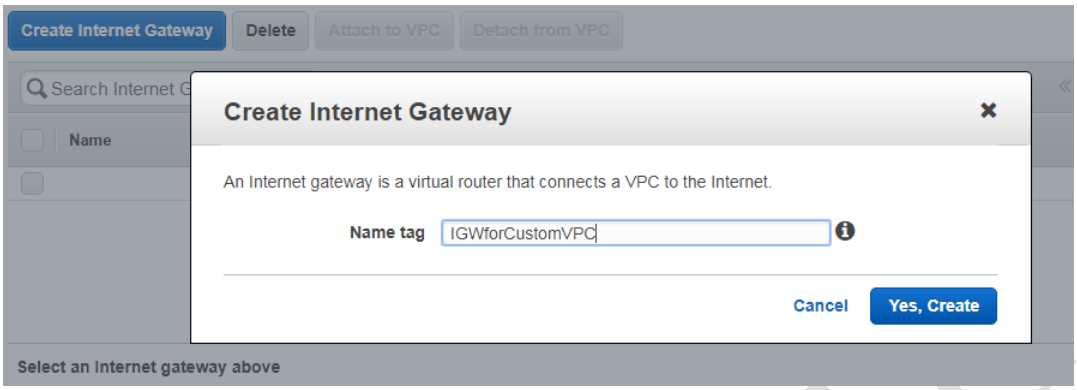
- This is how exactly subnet dashboard looks like now.



Name	Subnet ID	State	VPC	IPv4 CIDR	Available IPv4
	subnet-01f92d68	available	vpc-7d7ab214	172.31.16.0/20	4091
	subnet-721b0f38	available	vpc-7d7ab214	172.31.0.0/20	4091
Private Subnet	subnet-3f7f5f72	available	vpc-8b6984e3 | Custom VPC	192.168.2.0/24	251
Public Subnet	subnet-fbae5a93	available	vpc-8b6984e3 | Custom VPC	192.168.1.0/24	251

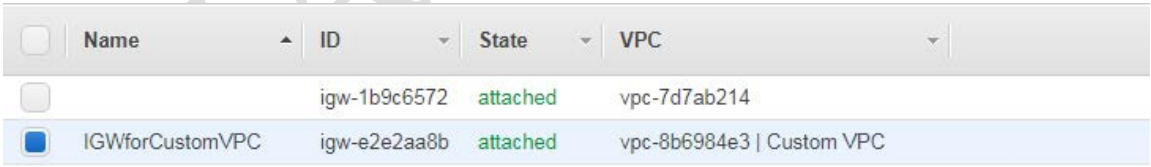
STEP 3: Creating an Internet gateway and Associating with Custom VPC.

- Navigate to internet Gateways from Navigation pane and Select **"Create Internet gateway**" option and provide a name for Internet Gateway.



- And select the "**Attach to VPC**" option and select the Custom VPC and click on **"Yes, Attach"** option.

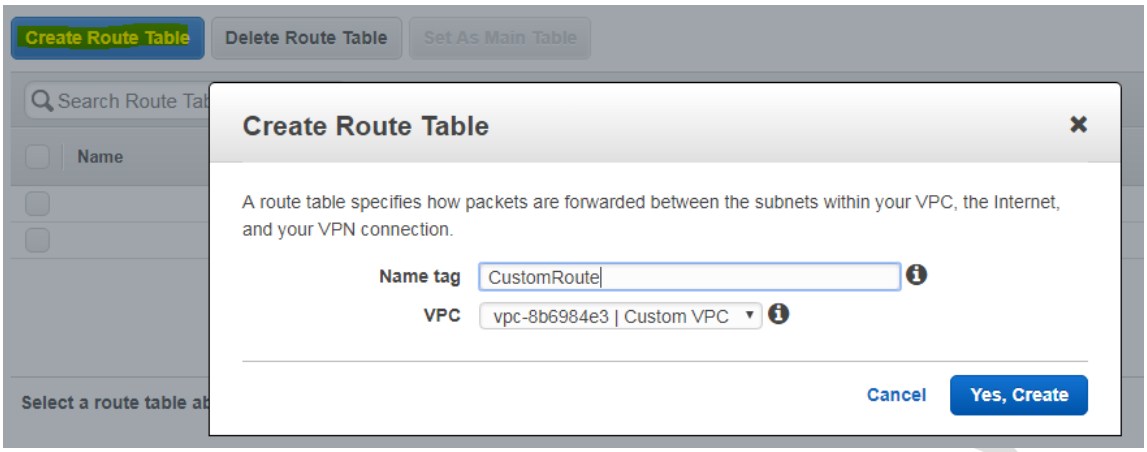
- This is how the IGW dashboard looks like after attaching it to custom VPC. Remember: One Internet gateway can be attached with only one VPC.



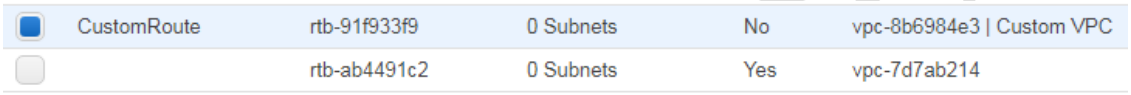
Name	ID	State	VPC
	igw-1b9c6572	attached	vpc-7d7ab214
IGWforCustomVPC	igw-e2e2aa8b	attached	vpc-8b6984e3 | Custom VPC

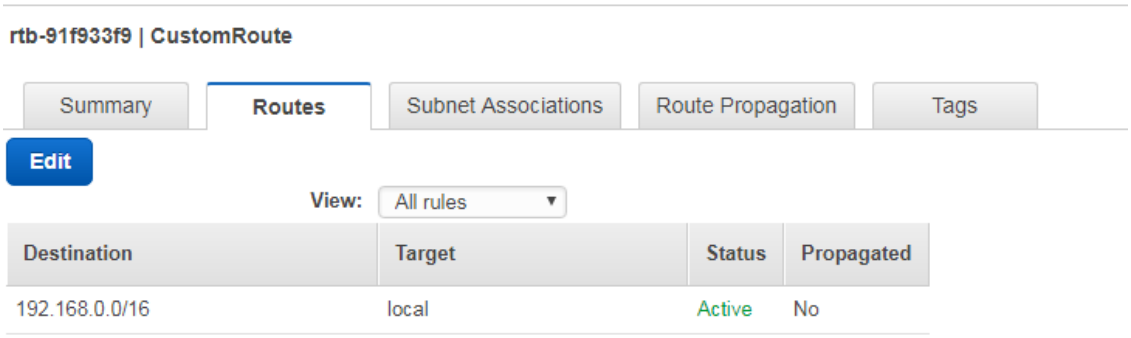
STEP 4: Creating Route Table and Performing Subnet association.

- Till now we have created a Custom VPC, Private and Public subnets, Created internet gateway and associated that to our custom VPC. Now we need to allow the traffic to our newly created subnets through the internet gateway, for that we are going to create a Route Table.
- Select **"Create Route Table"** option and give a name tag and select the Custom VPC and click on **"Yes, Create"** option.

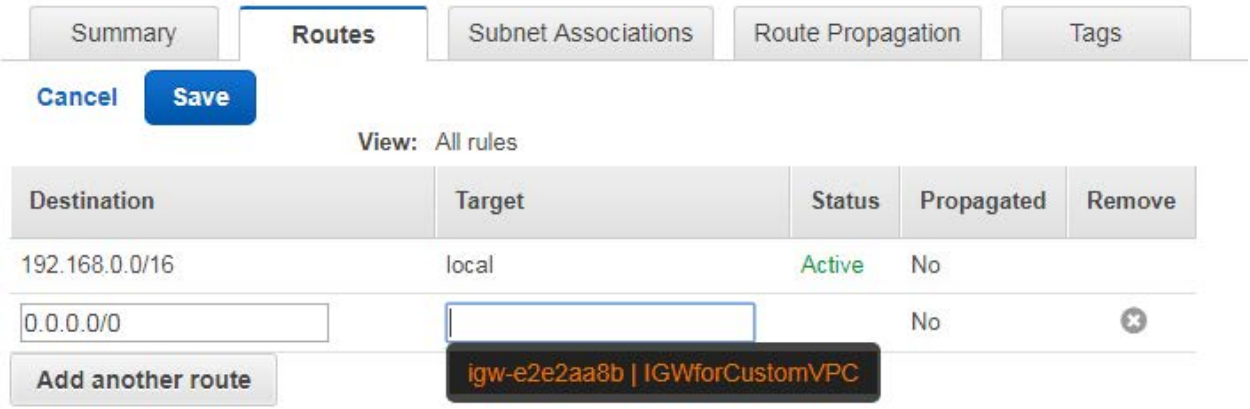


- Newly created route is not enabled with any of the public routes through IGW, Select the newly created route table to choose Route option to verify this.

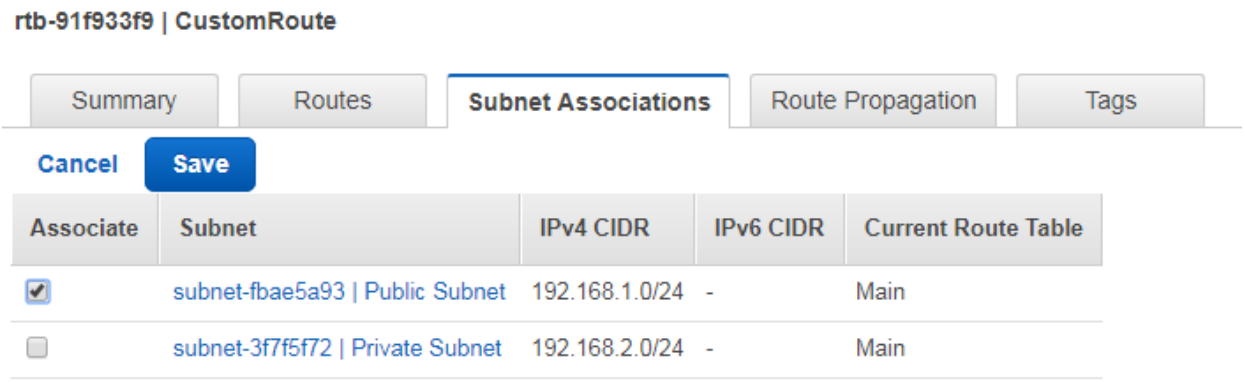




- Now we have to add a route by selecting edit option and select "**Add another Route"** option and enter **0.0.0.0/0** and when you click on Target automatically internet gateway will populate, choose **the populated IGW** and click on **save**.

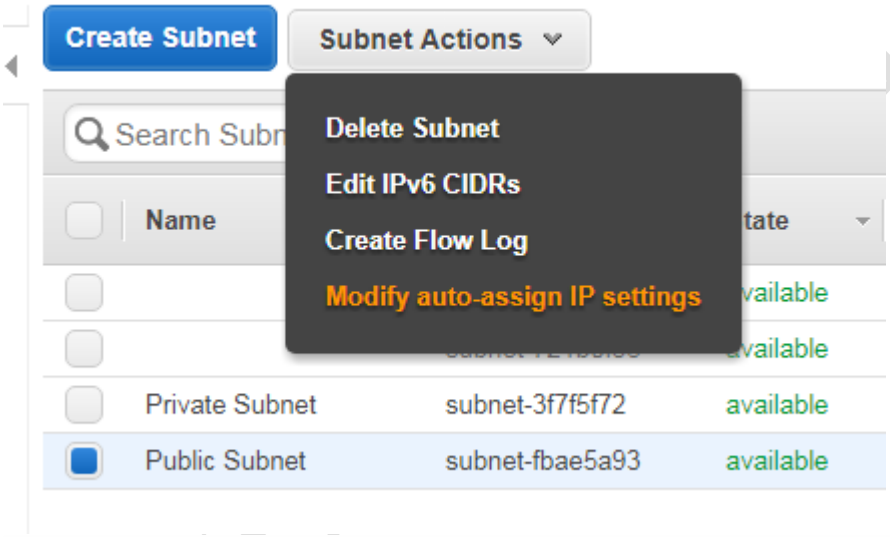


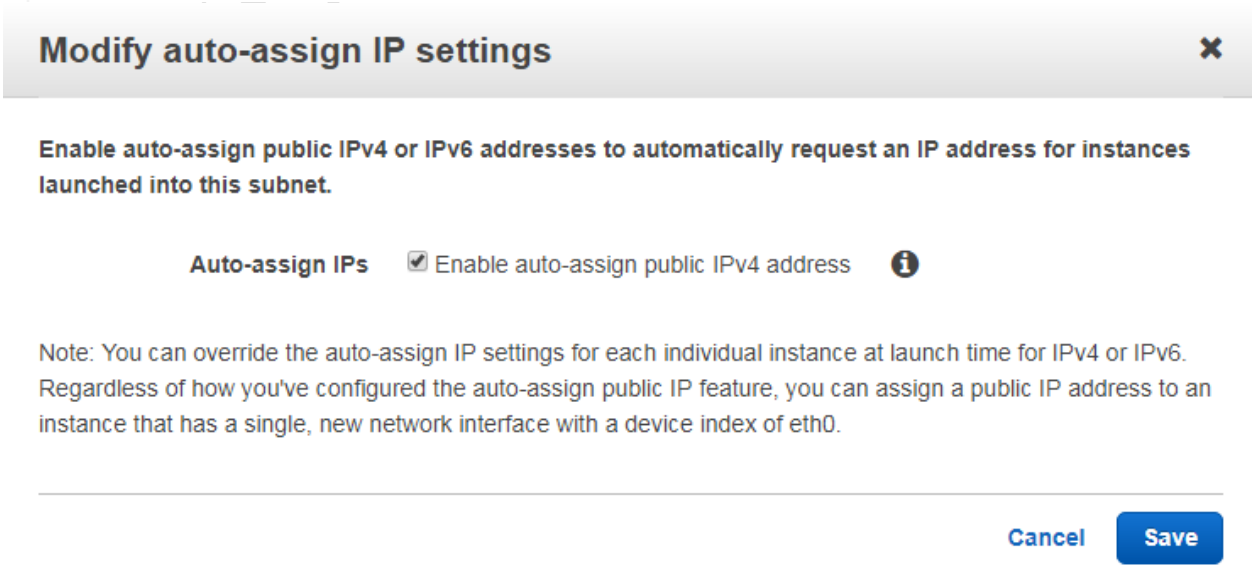
- Then select the **"Subnet Association"**ad click on **"Edit"** option and select the **"Public Subnet"** and click on save.



That's it our custom VPC is ready to deploy the resources. But we have one additional option.

STEP 5: Enabling Auto-assign IP Settings for Public Subnet (Optional Step).
You can enable auto assign public IP address option for Public Subnet instances, by editing the subnet settings. Navigate to Subnets dashboard and select the **"Public Subnet"** and choose the **"Subnet Actions"** and choose "**Modify auto-assign IP settings"**, select the checkbox and click on save.

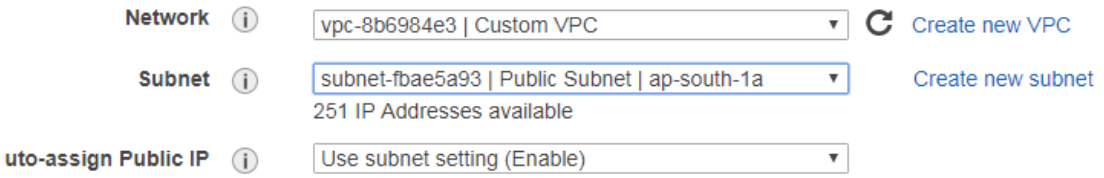




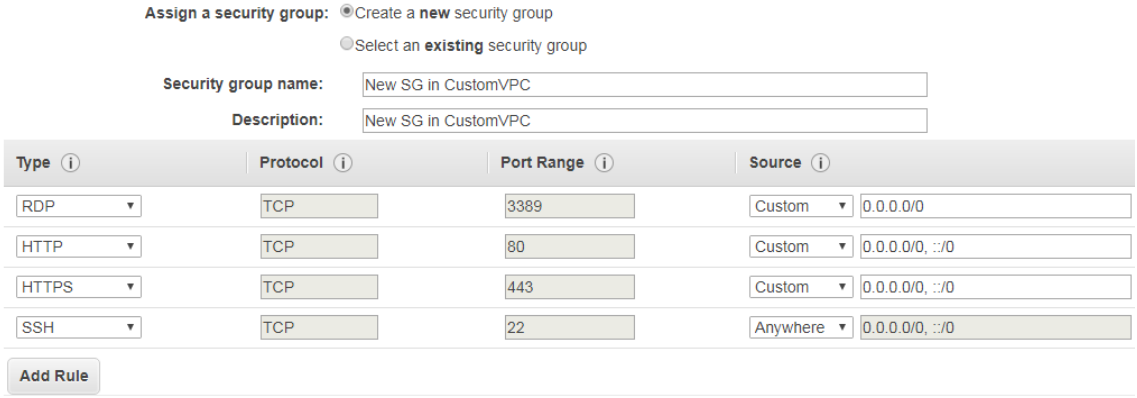
- Now we will get public IP address for every instance when we are launching it under public subnet, we no need to select the option in instance launch wizard.

Now Launch Instances in newly created custom VPC and verify.

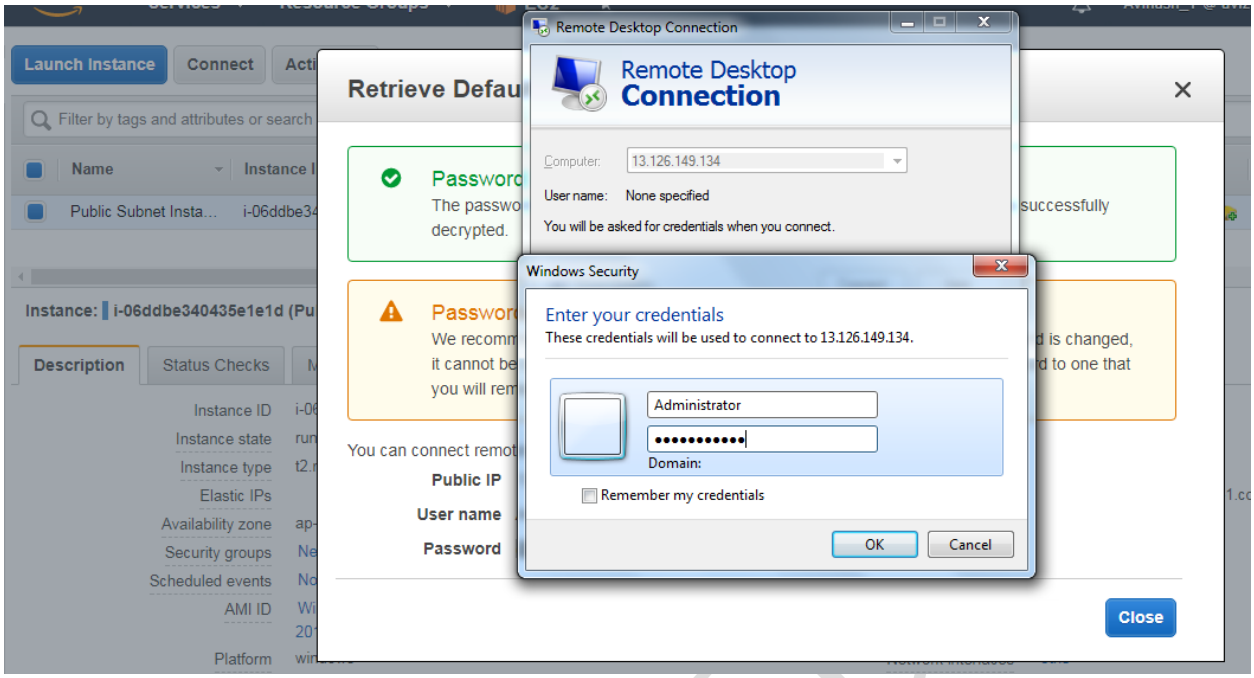
1. Launching an Instance in Custom VPC and selected to launch under "Public Subnet".



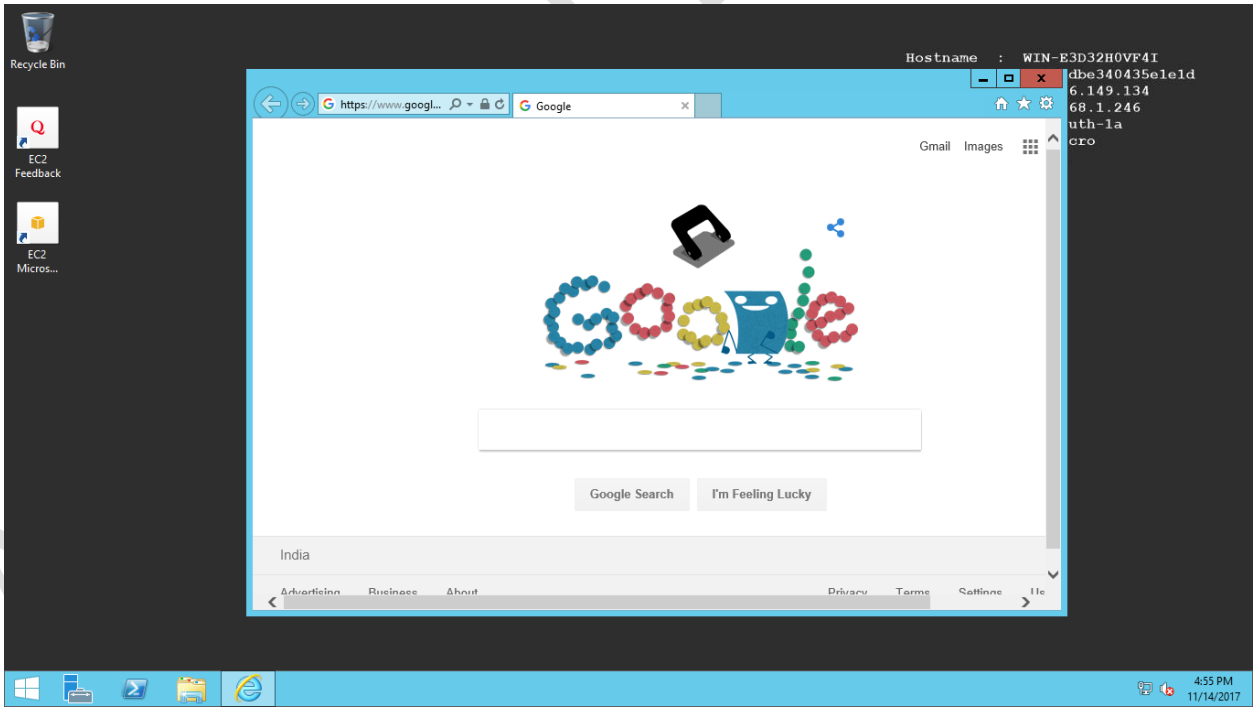
2. As this is a first instance launching under Custom VPC, we have to create new security group and need to open required ports and protocols.



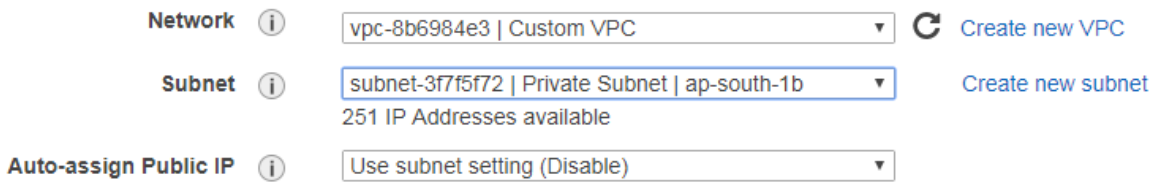
3. Now try to connect to the instance over the internet and verify the status as this is launched in Public Subnet, you can connect without any issues and you can browse the internet also in Instance.



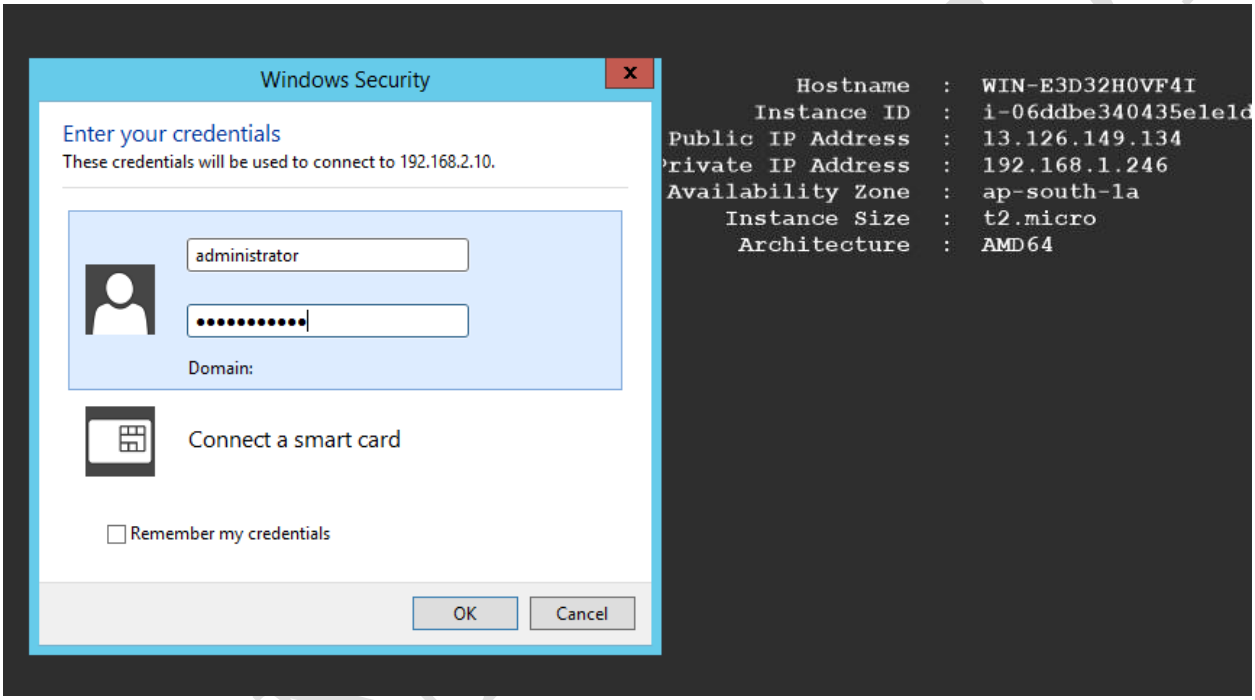
And we have successfully connected to the Instance, That means this instance is internet-faced and we can access anywhere from the world.

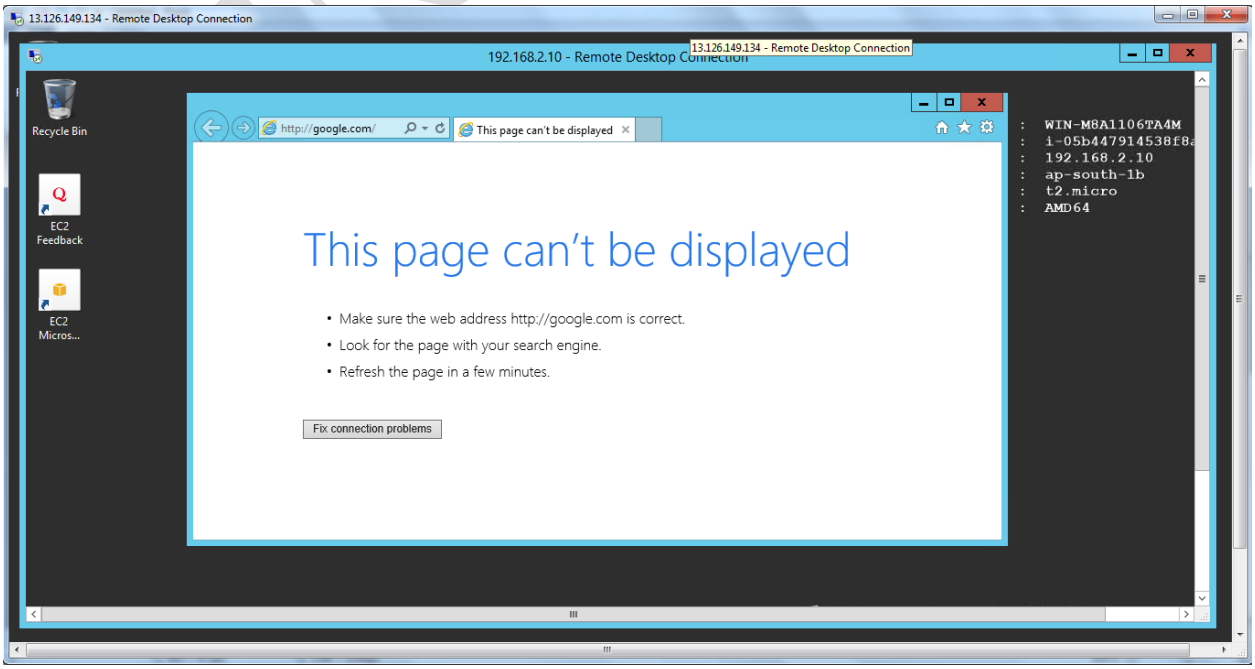


4. Now Launching another Instance in **"Custom VPC"** and selected to launch under **"Private Subnet"**.



5. And try to connect to the Private Subnet launched instance. When you browse for Username and password for instance connectivity, you'll get a Private IP address and we cannot use this to connect to the Launched instance.
    a. But we can connect to the same instance from the Public Subnets launched Instance.
    b. Remember as this is a private subnet instance, we will not get Internet in the Private Subnet instances.

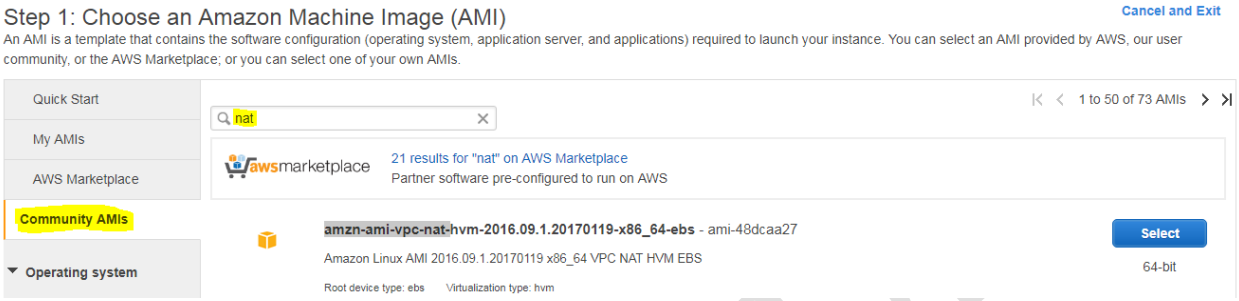




We have successfully connected to the Private Subnet instance from public Subnet instance, But We are not able to get internet connectivity in private subnet instance. TO get Internet in private Hosted instances we need to **launch a NAT Instance or NAT gateway.**

Launching NAT Instance:

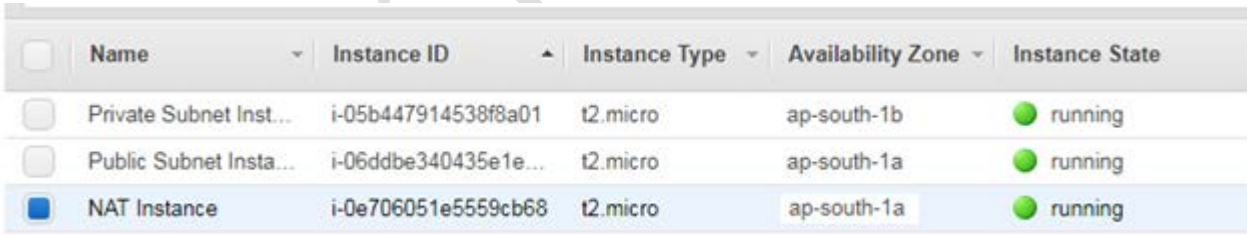
- To launch NAT instance go to EC2 Dashboard and initiate an instance launch and Select **"Community AMI"** and Search for **"NAT"**as shown in below image and choose any of the instance.



- Select one of the instances from the listed instances, and choose NAT instance with t2.micro and follow the instance launch wizard same as a regular instance.
  Note: The amount of traffic that NAT instances supports, depends on the instance size. If you are bottlenecking, increase the instance configuration.

  Note: Make sure your NAT instance security group is opened with Http and Https.

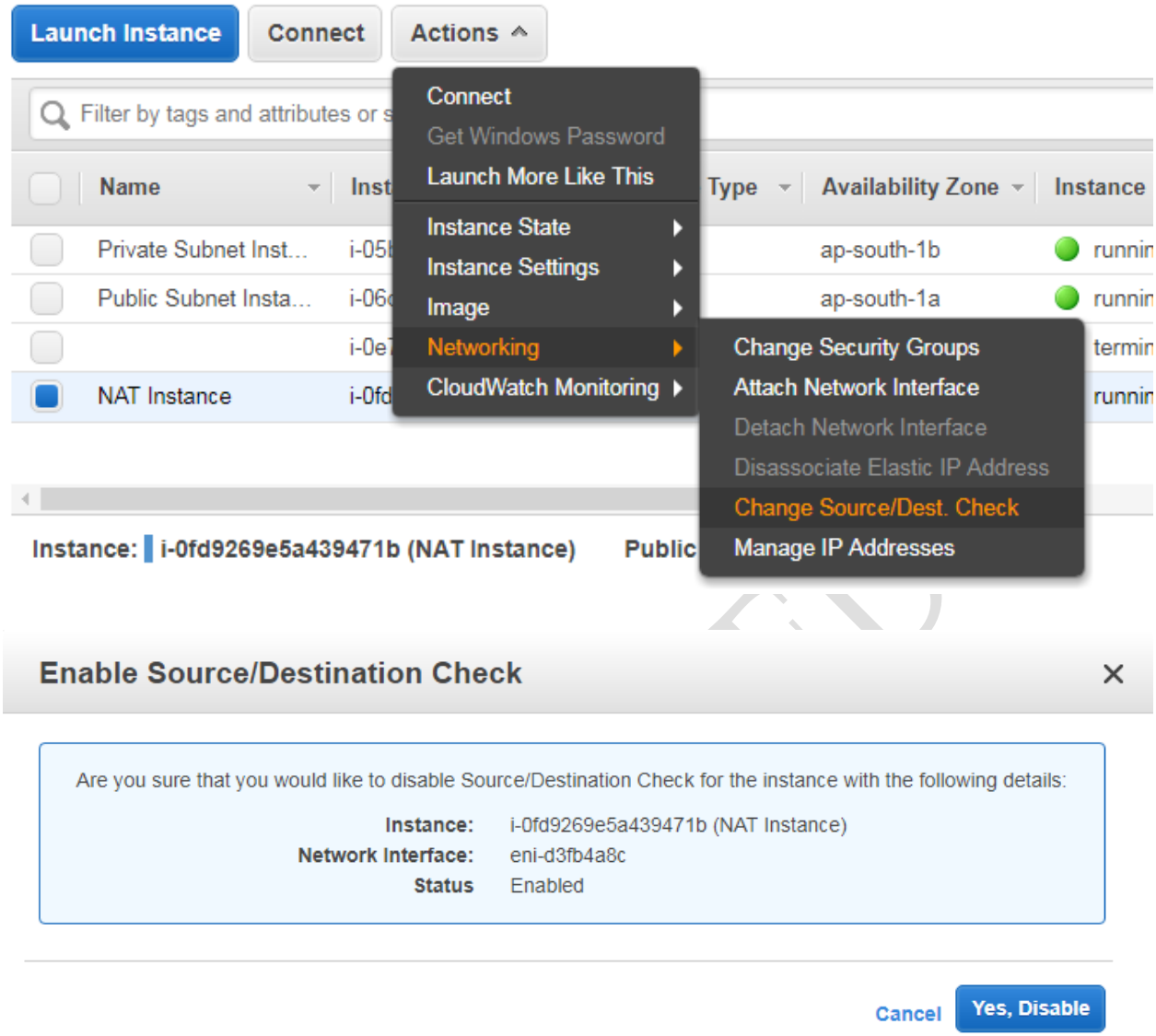
  Note: NAT Instance must be launched in **Custom VPC's Public Subnet.**



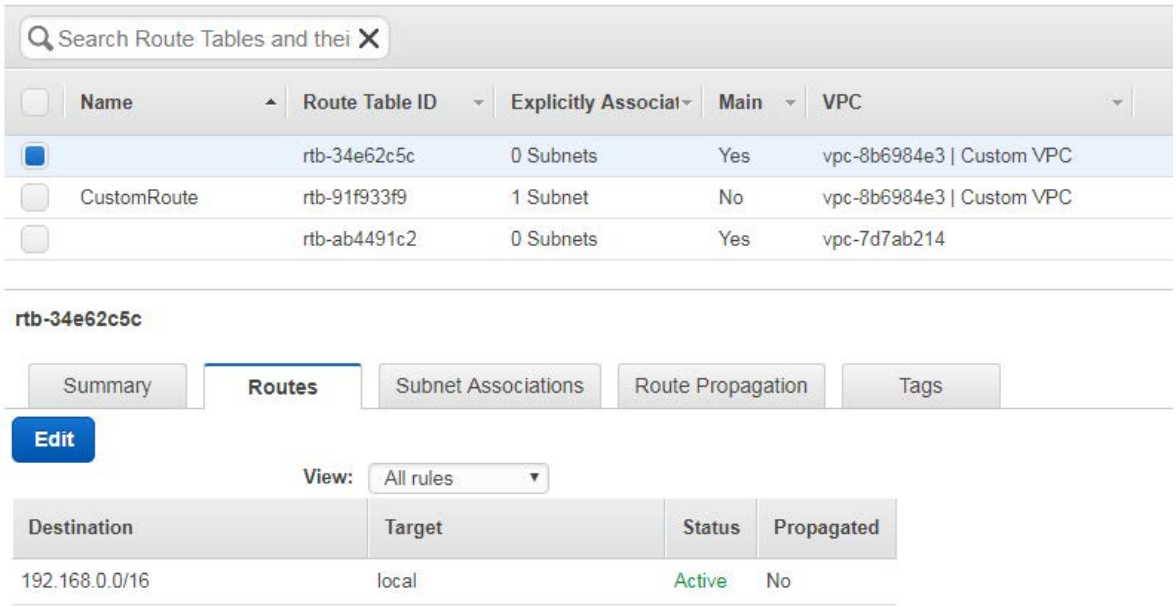
Name	Instance ID	Instance Type	Availability Zone	Instance State
Private Subnet Inst...	i-05b447914538f8a01	t2.micro	ap-south-1b	running
Public Subnet Insta...	i-06ddbe340435e1e...	t2.micro	ap-south-1a	running
NAT Instance	i-0e706051e5559cb68	t2.micro	ap-south-1a	running

- We need to disable Source/Destination check for NAT instance.
  Each EC2 instance performs source/destination checks by default. This means that the instance must be the source or destination of any traffic it sends or receives. However, a NAT instance must be able to send and receive traffic when the source or destination is not itself. Therefore, you must disable source/destination checks on the NAT instance.

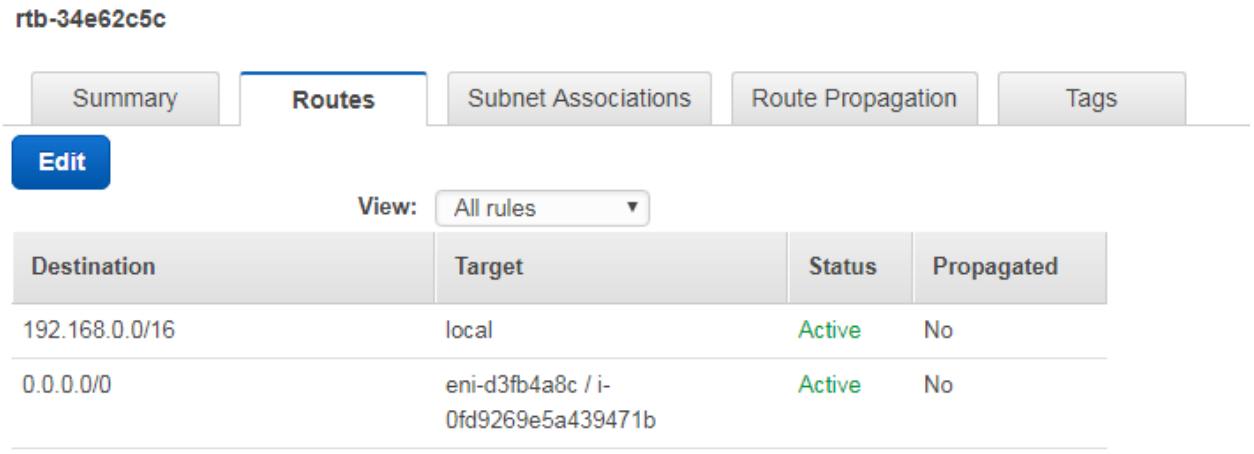
- To disable source/destination check, Select the NAT Instance, Goto Actions, Networking and choose **"Change Source/Destination Check"**and select **"Yes, Disable".**



- Now we have to edit **"Custom VPCs Main Route table"** and need to add a route through the NAT Instance, then the private subnet instances will get the internet connectivity.



- Select the Edit option and enter the Destination as **0.0.0.0/0**and select the target as **NAT Instance**.



- Now we will get the internet for our Private subnet instances through the NAT instances. And here is the output.

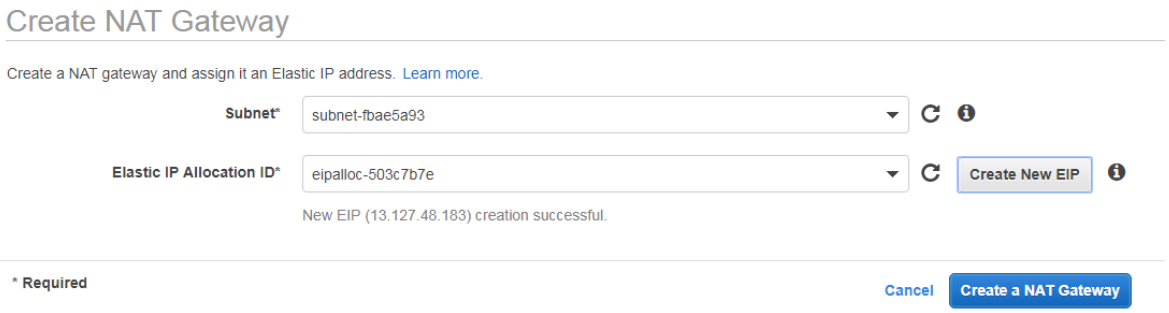
NAT GATEWAYS: Instead of NAT Instances, we can use NAT Gateways. We have lot of advantages with NAT gateways compare to NAT instances. Make sure you terminate the NAT Instance before performing the NAT Gateways, we don't required two resources to provide internet to Private subnet.

Here is some advantages listed:

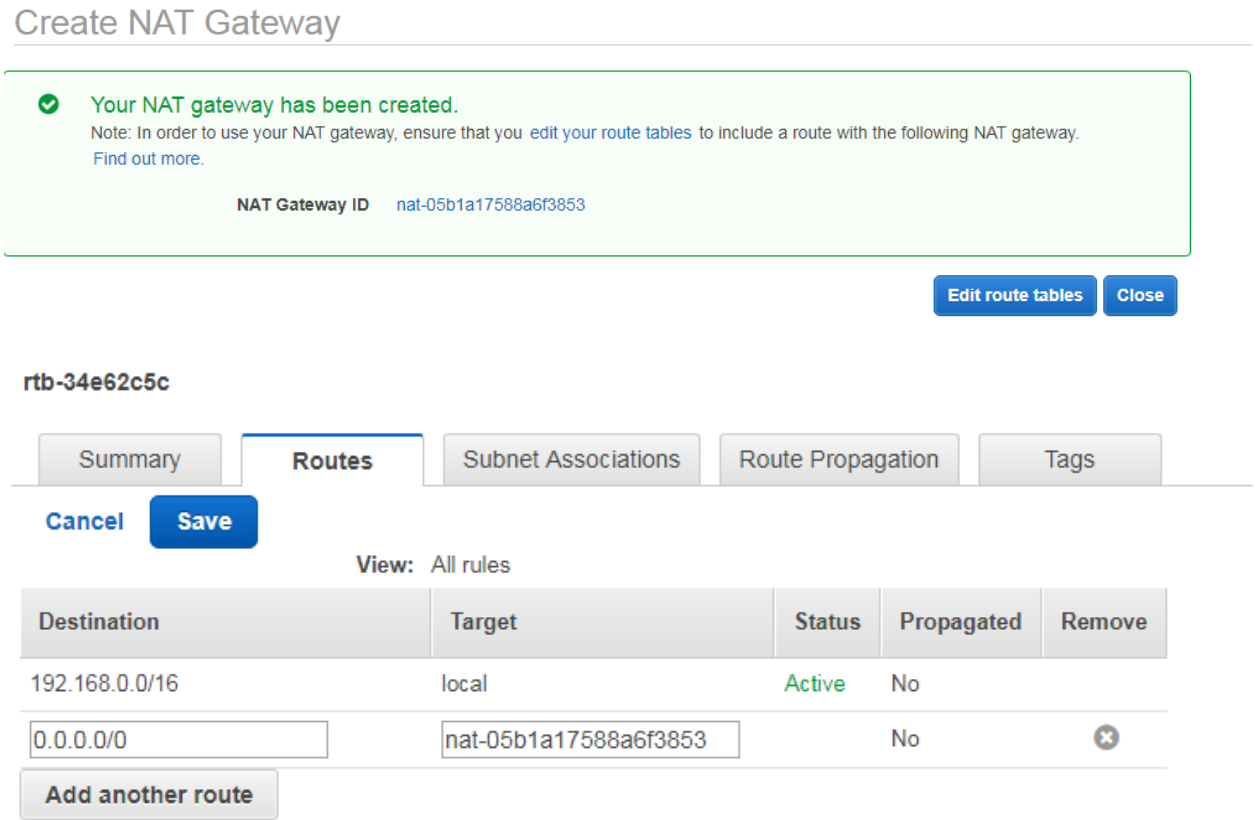
- Preferred for the enterprise/Production level
- Scale automatically up to 10 Gbps
- Not associated with security groups
- Automatically assigned a public ip address (EIP)
- You have to update route tables to take effect.
- No O.S so No need to patch
- No Instance so No need to disable Source/Destination Checks

Steps to create NAT gateways:

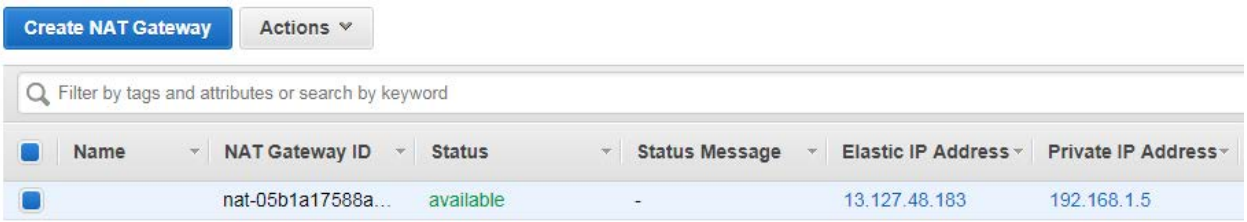
- Select NAT Gateways option from VPC Navigation Pane. And click on **"Create NAT Gateway"** option.
- As same as NAT instance, we have to create the NAT Gateway also in **Public Subnet of CustomVPC**.
- If you have any Elastic IP without associating to any of the resource, we can use the same here, if you don't have select the **Create New EIP** option and click on **Create a NAT Gateway**.



- And we have to edit the Route table as same as NAT instance process. Select the Custom VPCs Main Route table and open the Destination **0.0.0.0/0** and target as **NAT Gateway.**



- Here is the NAT Gateway information after creation.



- Now go to private subnet instance and verify the internet connectivity. You will able to browse the internet and try to look for the public Ip information from the private subnet instance you'll get the NAT gateway's IP Address, That means we are getting internet through NAT Gateway to the Private subnet instance.

Network Access Control Lists (ACLs)

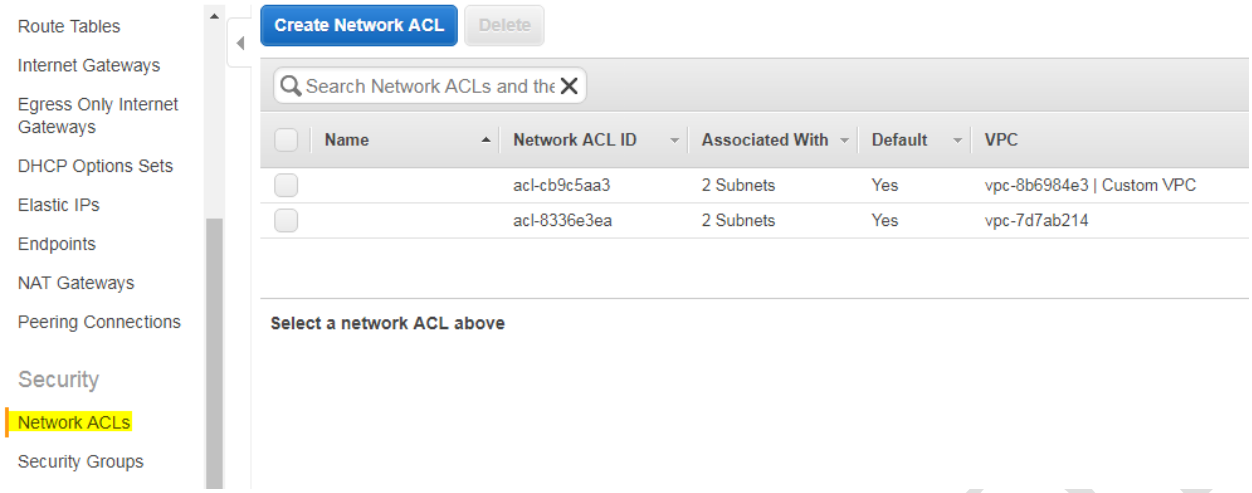
A network access control list (ACL) is another layer of security that acts as a stateless firewallon a subnet level. A network ACL is a numbered list of rules that AWS evaluates in order,starting with the lowest numbered rule, to determine whether traffic is allowed in or out ofany subnet associated with the network ACL.

Every subnet must be associated with a networkACL.

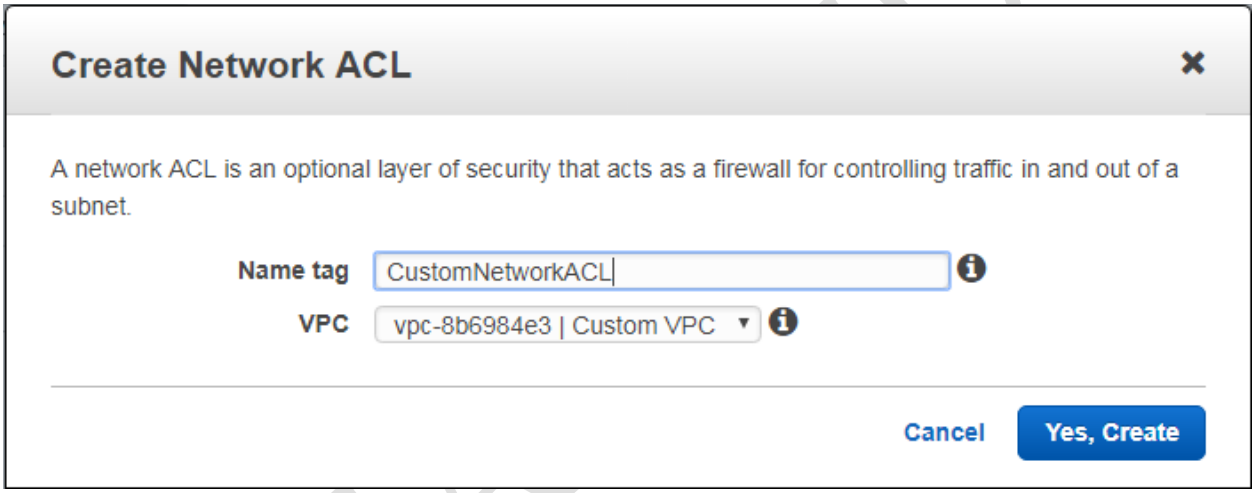
Security Groups Vs Network ACLs

Security Group	Network ACL
Operates at the instance level (first layer of defense)	Operates at the subnet level (second layer of defense)
Supports allow rules only	Supports allow rules and deny rules
Stateful: Return traffic is automatically allowed, regardless of any rules	Stateless: Return traffic must be explicitly allowed by rules.
AWS evaluates all rules before deciding whether to allow traffic	AWS processes rules in number order when deciding whether to allow traffic.
Applied selectively to individual instances	Automatically applied to all instances in the associated subnets; this is a backup layer of defense, so you don't have to rely on someone specifying the security group.

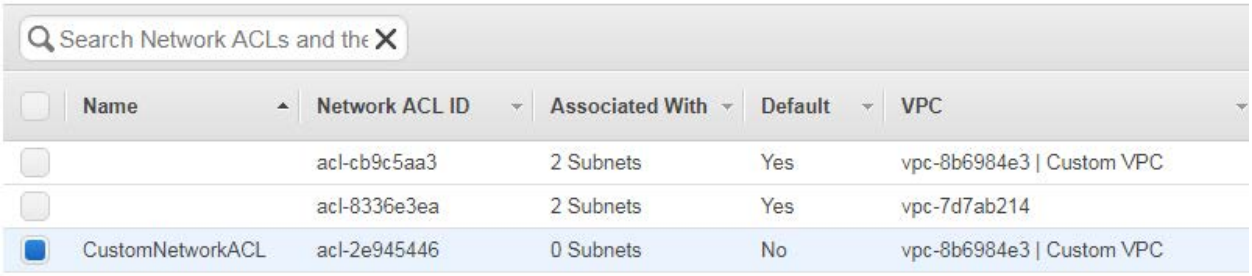
- Navigate to the "Network ACLs" under "Security" option and choose "Create Network ACL" option.



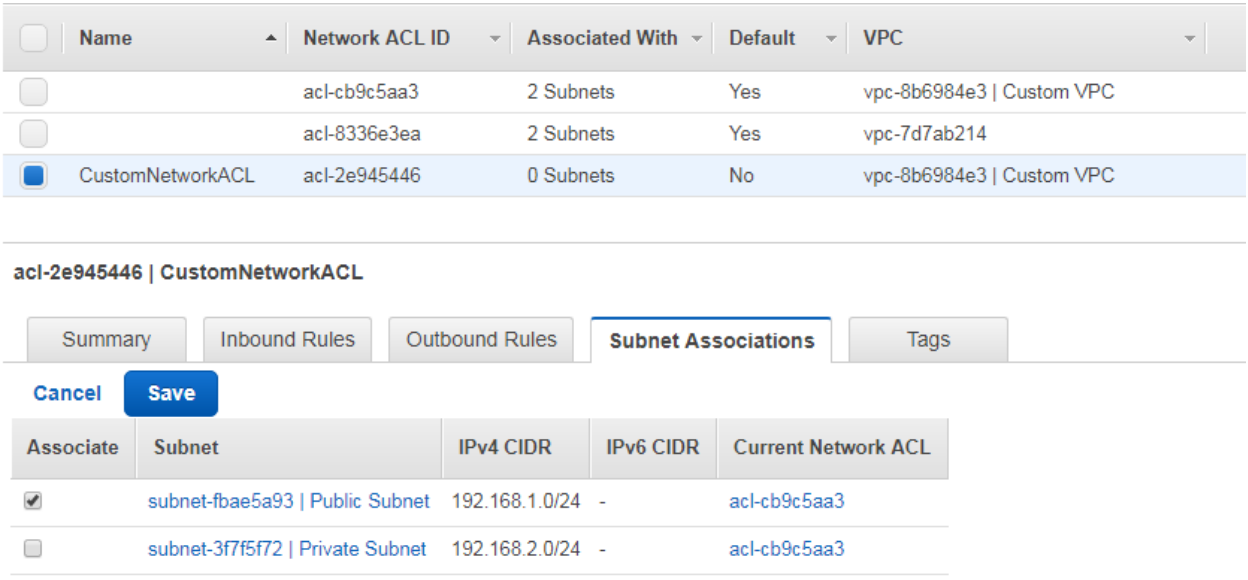
- Give a name for the newly creating Network ACL and Create this under Custom VPC.



- Newly Created NACL will not have any Subnets Associated with it.



- To Associate a subnet Select the “Subnet Association” and choose the subnet you want to associate under the “Custom Network ACL”.



Name	Network ACL ID	Associated With	Default	VPC
	acl-cb9c5aa3	2 Subnets	Yes	vpc-8b6984e3 | Custom VPC
	acl-8336e3ea	2 Subnets	Yes	vpc-7d7ab214
CustomNetworkACL	acl-2e945446	0 Subnets	No	vpc-8b6984e3 | Custom VPC

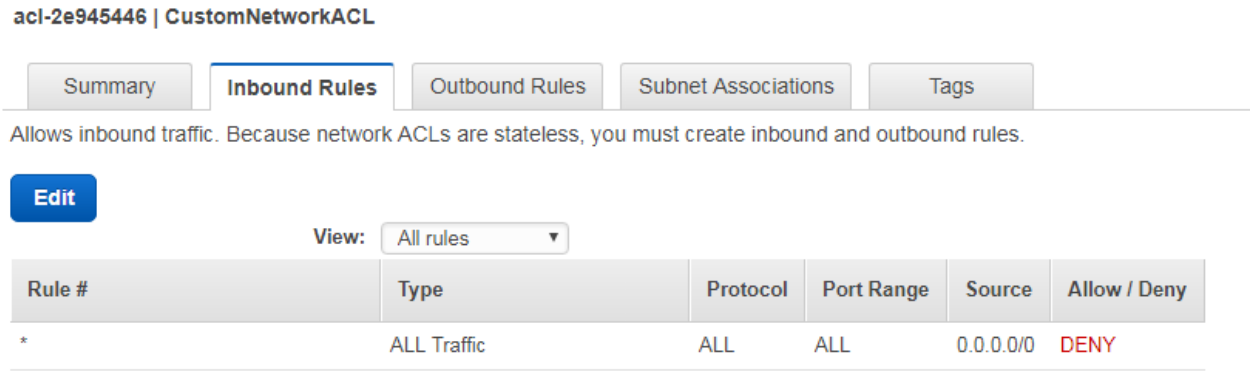
acl-2e945446 | CustomNetworkACL

Summary | Inbound Rules | Outbound Rules | Subnet Associations | Tags

Cancel Save

Associate	Subnet	IPv4 CIDR	IPv6 CIDR	Current Network ACL
☑	subnet-fbae5a93 | Public Subnet	192.168.1.0/24	-	acl-cb9c5aa3
☐	subnet-3f7f5f72 | Private Subnet	192.168.2.0/24	-	acl-cb9c5aa3

- By Default, all the Inbound and outbound traffic will be set to Deny mode.



acl-2e945446 | CustomNetworkACL

Summary | Inbound Rules | Outbound Rules | Subnet Associations | Tags

Allows inbound traffic. Because network ACLs are stateless, you must create inbound and outbound rules.

Edit

View: All rules

Rule #	Type	Protocol	Port Range	Source	Allow / Deny
*	ALL Traffic	ALL	ALL	0.0.0.0/0	DENY

- Here we have to Edit and add the required Protocol and Port Range and Source same as Security groups.

The following are the parts of a network ACL rule:

Rule number: Rules are evaluated starting with the lowest numbered rule. As soon as a rule matches traffic, it's applied regardless of any higher-numbered rule that may contradict it.

Protocol: You can specify any protocol that has a standard protocol number. For more information, see Protocol Numbers. If you specify ICMP as the protocol, you can specify any or all of the ICMP types and codes.

[Inbound rules only] The source of the traffic (CIDR range) and the destination (listening) port or port range.

[Outbound rules only] The destination for the traffic (CIDR range) and the destination port or port range.

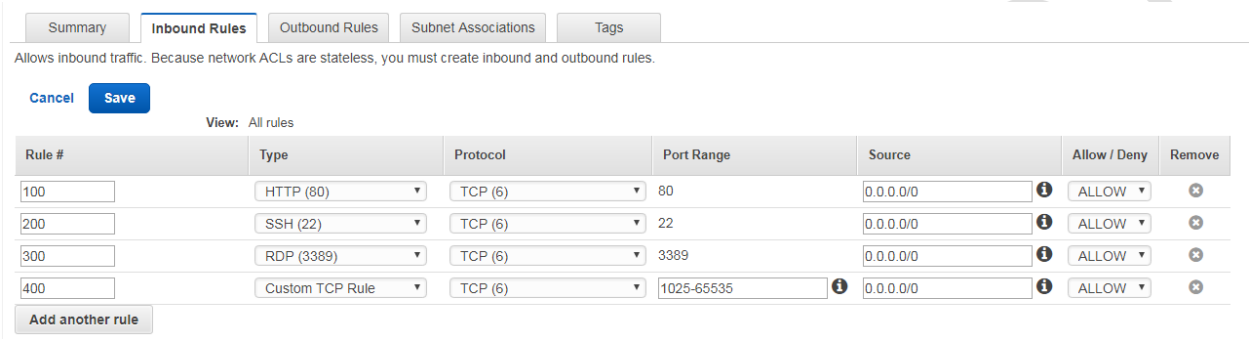
Choice of **ALLOW or DENY** for the specified traffic.

- And AWS will suggest to create the rules increments of 100.
- If you want to use this Network ACL with Elastic Load balancers, open the Ephemeral ports in inbound and outbound.
  Ephemerals port range varies depending on the client's operating system. Many Linux kernels use ports 32768-61000.

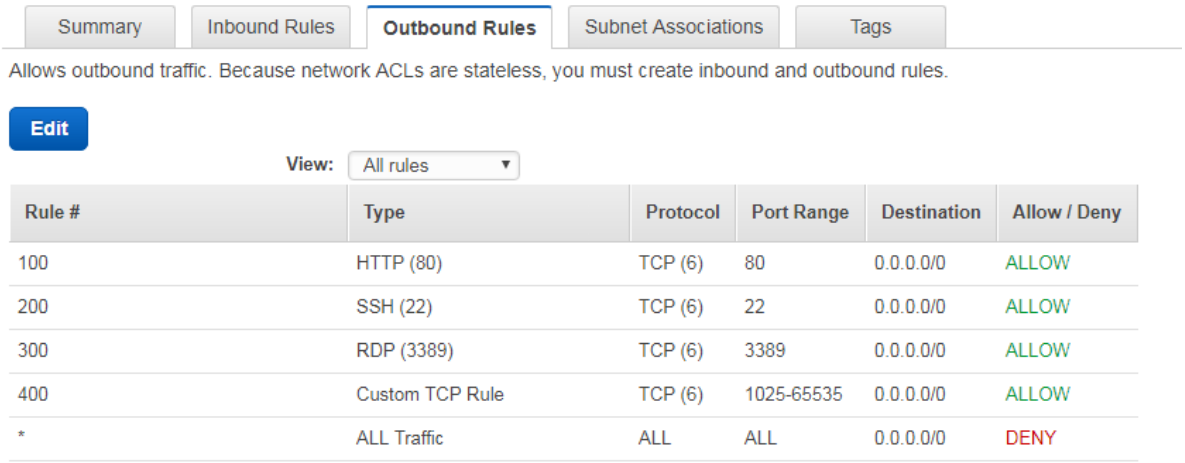
  Elastic Load Balancing use ports 1024-65535.

  Windows Server 2008 and later versions use ports 49152-65535.

  A NAT gateway uses ports 1024-65535.



- Perform the same for Outbound Rules also, as the Network ACLs are Stateless.



- We have Deny option also here with Network ACLs. We can create another rule for same Protocol and we can set it to Allow/Deny based on our requirement. **Lowest Rule will takes the Highest Priority.**

VPC Peering

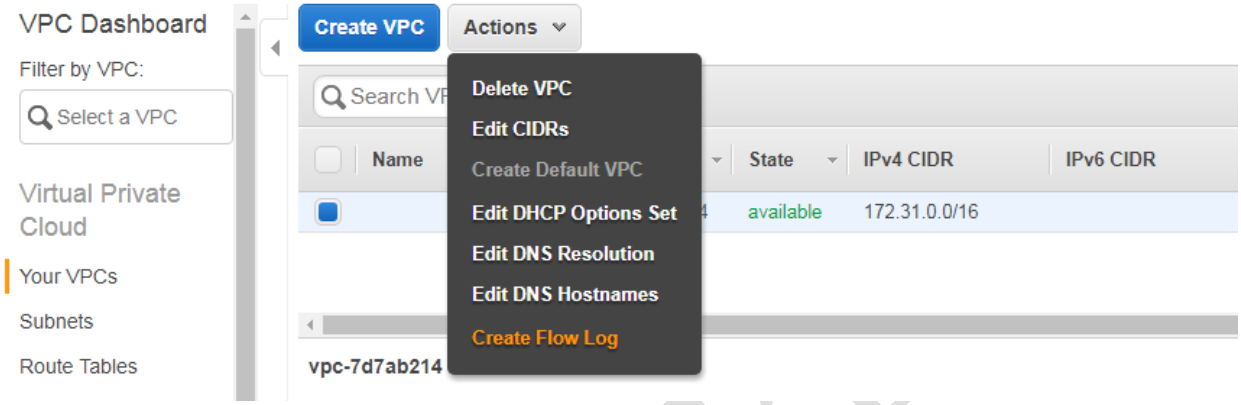
- Allows you to connect one VPC with another via a direct network route using private IP addresses.
- Instances behave as if they were on the same private network
- You can peer VPC's with other AWS accounts as well as with other VPCs in the same account.

- Peering is in a star configuration, ie 1 central VPC peers with 4 others. NO TRANSITIVE PEERING!!!

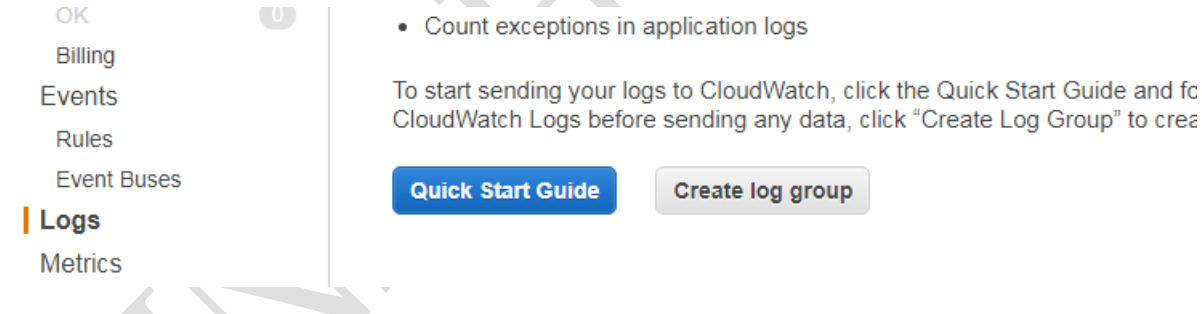
VPC Flow log Creation:

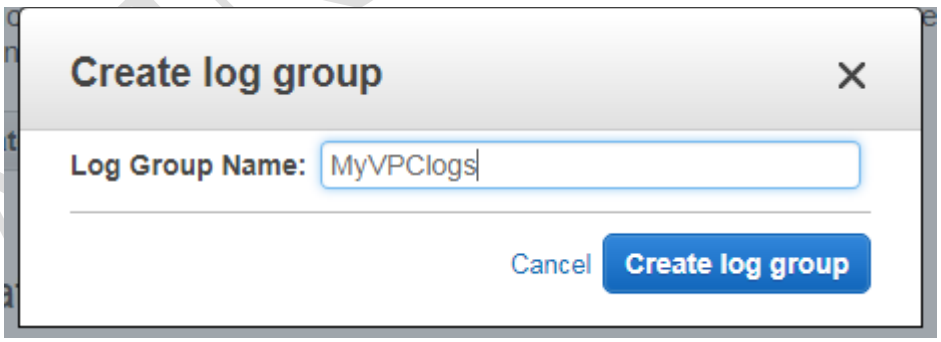
VPC Flow Logs is a feature that enables you to capture information about the IP traffic going to and from network interfaces in your VPC. Flow log data is stored using Amazon CloudWatch Logs. After you've created a flow log, you can view and retrieve its data in Amazon CloudWatch Logs.

1. To enable the VPC Flow Log, Select the VPC and navigate to Create Flow Log under Actions.

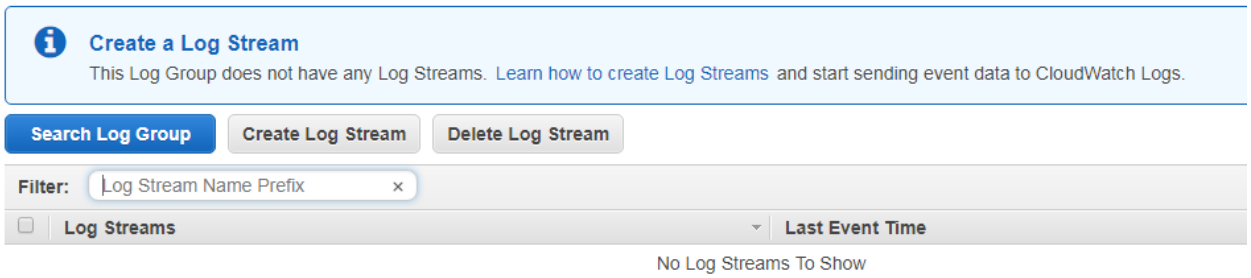


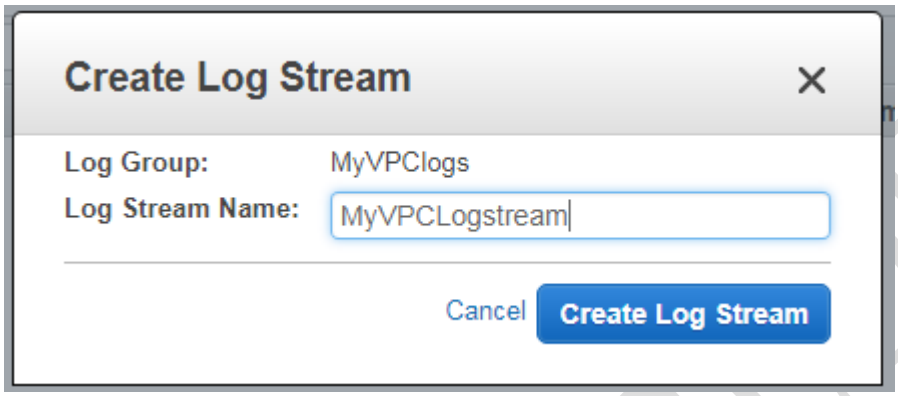
2. Before creating the Flow Log on VPc, We need to Create log Group in cloudwatch. Navigate to cloudwatch and select the Logs option and select the Create log group option.



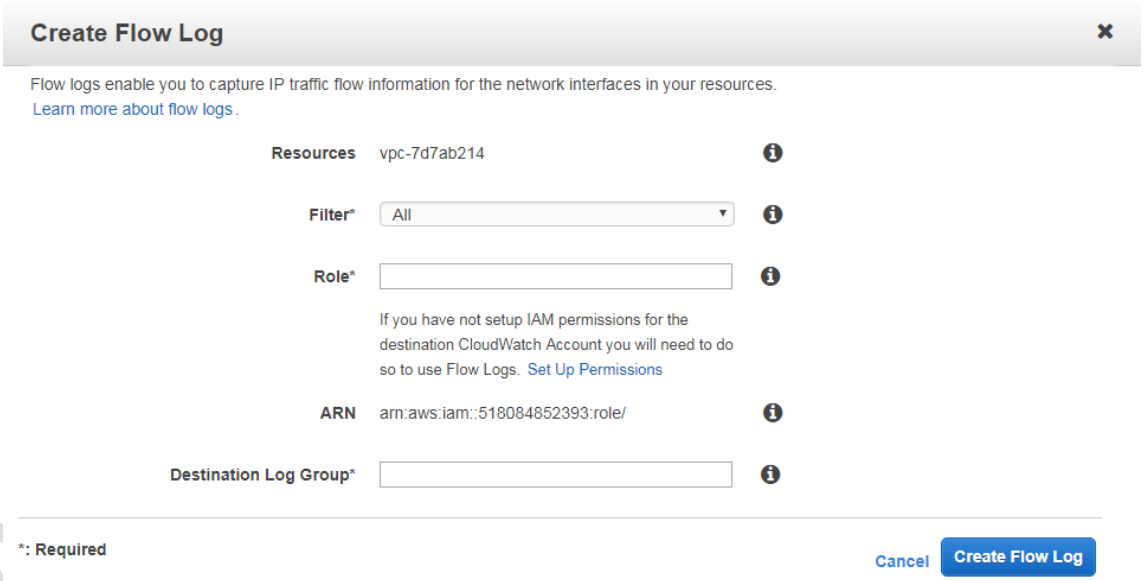


3. Select the Log group and Create a Log Stream as shown in below image.

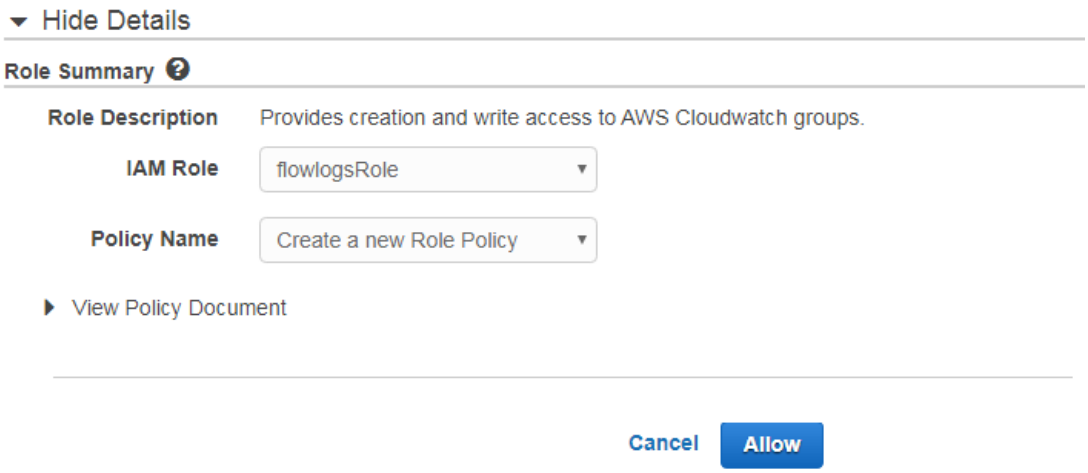




4. Now navigate back to VPC and create a Flow Log.



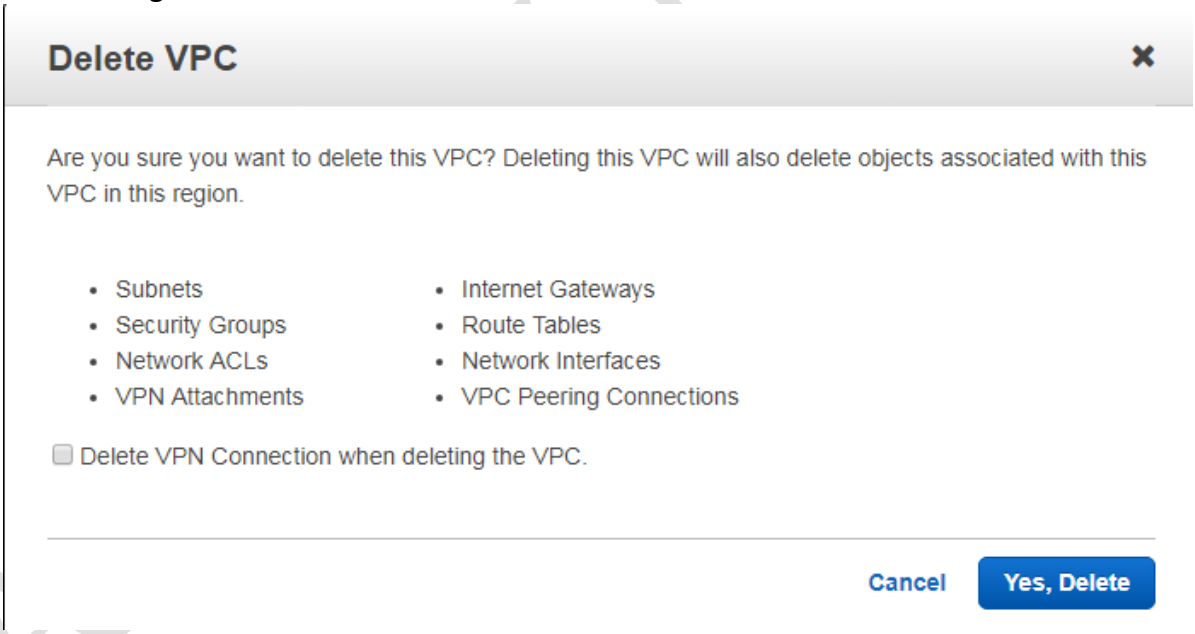
5. Select the Filter and choose what traffic (All/Accept/Reject) you want to gets in Log.
6. Create a new IAM role to perform the task on behalf of us. Click on Setup Permissions option and it'll navigate a new tab and select allow.



7. Select the newly created Log Group in CloudWatch, and all the traffic will be logged into CloudWatch Logs under Logstream.

VPC Cleanup:

When you delete the VPC, Automatically all the resources attached to the VPC also deletes. As mentioned below image, Subnets, Security groups, Network ACLs, interent Gateways, Route tables etc will delete along with VPC.



Bastion host:

Bastion hosts are instances that sit within our public subnet and are typically accessed using SSH or RDP. Once remote connectivity has been established with the bastion host, it then acts as a 'jump' server, allowing you to use SSH or RDP to log in to other instances (within private subnets) deeper within your VPC. When properly configured through the use of security groups and Network ACLs (NACLs), the bastion essentially acts as a bridge to your private instances via the internet.

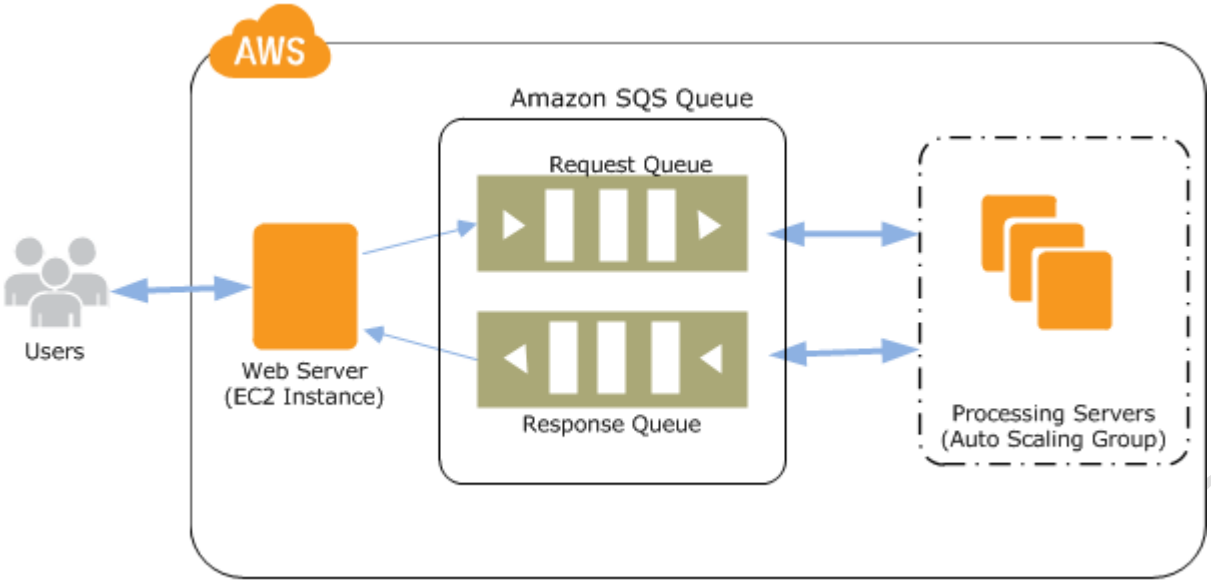
Application Services

Amazon Simple Queue Service (Amazon SQS)

Amazon SQS is a fast, reliable, scalable, and fully managed message queuing service. AmazonSQS makes it simple and cost effective to decouple the components of a cloud application.

- Amazon SQS is a web service that gives you access to a message queue that can be used to store messages while waiting for a computer to process them.
- A queue is a temporary repository for messages that are awaiting processing.
- An Amazon SQS queue is basically a buffer between the application components that receive data and those components that process the data in your system.
- Messages can contain up to 256 KB of text in any format.
- Amazon SQS ensures delivery of each message at least once, and supports multiple readers and writers interacting with the same queue.
- Message Retention period is 14 Days
- Amazon SQS is engineered to provide "**at least once**" delivery of all messages in its queues. Although most of the time each message will be delivered to your application exactly once.
- A single queue can be used simultaneously by many distributed application components, with no need for those components to coordinate with each other to share the queue.
- Maximum message size 256kb now available
- AWS will Billed as Chunks, Each Chunk size is 64kb, That means a 256kb message will be 4 x 64kb "chunks".
- First 1 million Amazon SQS Requests per month are fre
- $0.50 per 1 million Amazon SQS Requests per month thereafter ($0.00000050 per SQS Request)
- A single request can have from 1 to 10 messages, up to a maximum total payload of 256KB.
- Each 64KB 'chunk' of payload is billed as 1 request. For example, a single API call with a 256KB payload will be billed as four requests.

For example, suppose that you have a web app that receives orders from customers. The app runs on EC2 instances in an Auto Scaling group that is configured to handle a typical number of orders. The app places the orders in an Amazon SQS queue until they are picked up for processing, processes the orders, and then sends the processed orders back to the customer. The following diagram illustrates the architecture of this example.



Amazon SQS is a distributed queue system that enables web service applications to quickly and reliably queue messages that one component in the application generates to be consumed by another component.

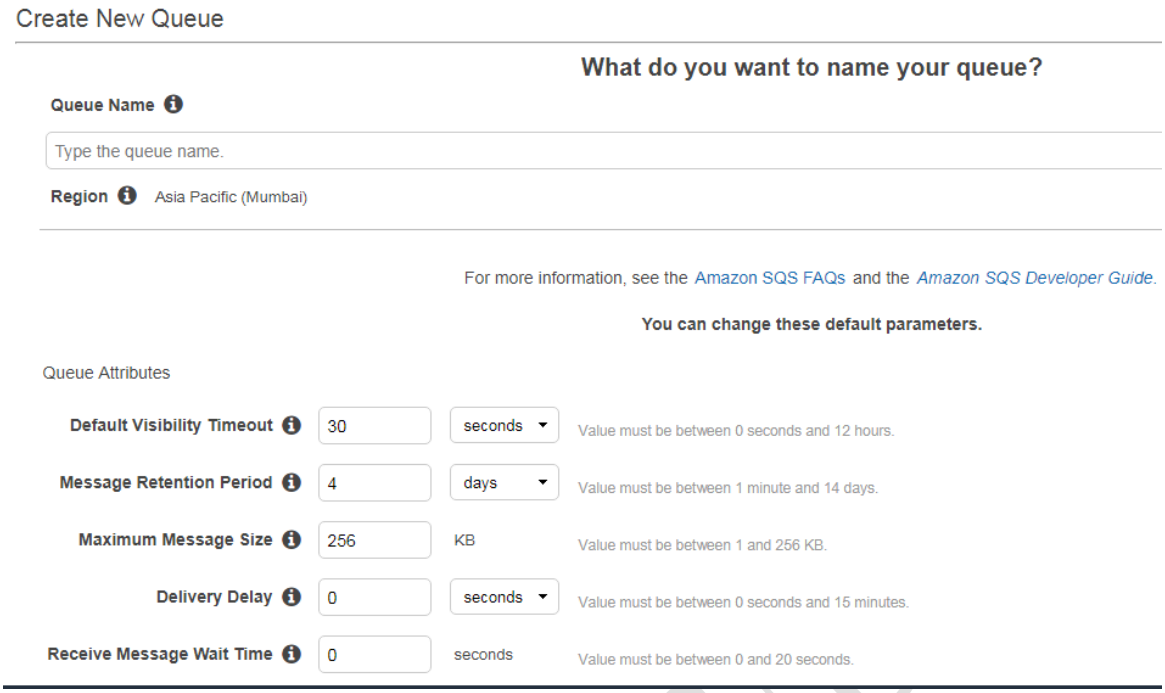
UsingAmazon SQS, you can store application messages on reliable and scalable infrastructure,enabling you to move data between distributed components to perform different tasks asneeded.

Amazon SQS ensures delivery of each message at least once and supports multiple readersand writers interacting with the same queue. A single queue can be used simultaneously bymany distributed application components, with no need for those components to coordinatewith one another to share the queue. Although most of the time each message will be delivered to your application exactly once, you should design your system to be idempotent SQL service does not guarantee First In, First Out (FIFO) delivery ofmessages.

Amazon SQS supports up to 12 hours' maximum visibility timeout.

When creating a new queue, you must provide a queue name that is unique within the scopeof all of your queues. Amazon SQS assigns each queue an identifier called a queue URL,which includes the queue name and other components that Amazon SQS determines.Whenever you want to perform an action on a queue, you must provide its queue URL.

- TO create a Queue, Navigate to "Messaging" section and select the "Simple Queue Service".
- Here is the default values we are getting with the Queue.



Amazon Simple Workflow Service (Amazon SWF)

Amazon Simple Workflow Service (Amazon SWF) is a web service that makes it easy to coordinate work across distributed application components.

Amazon SWF enables applications for a range of use cases, including media processing, web application back-ends, business process workflows, and analytics pipelines, to be designed as a coordination of tasks.

Amazon SWF makes it easy to build applications that coordinate work across distributedcomponents. In Amazon SWF, a task represents a logical unit of work that is performed by acomponent of your application.

Amazon SWF gives you full control over implementing and coordinatingtasks without worrying about underlying complexities such as tracking their progress andmaintaining their state.

We have three SWF Actors:

- Workflow Starters - An application that can initiate (start) a workflow. Could be your e-commerce website when placing an order.
- Deciders - Control the flow of activity tasks in a workflow execution. If something has finished in a workflow (or fails) a Decider decides what to do next.
- Activity Workers - Carry out the activity tasks

Amazon Simple Notification Service (Amazon SNS)

Amazon Simple Notification Service (Amazon SNS) is a web service that makes it easy to set up, operate, and send notifications from the cloud.

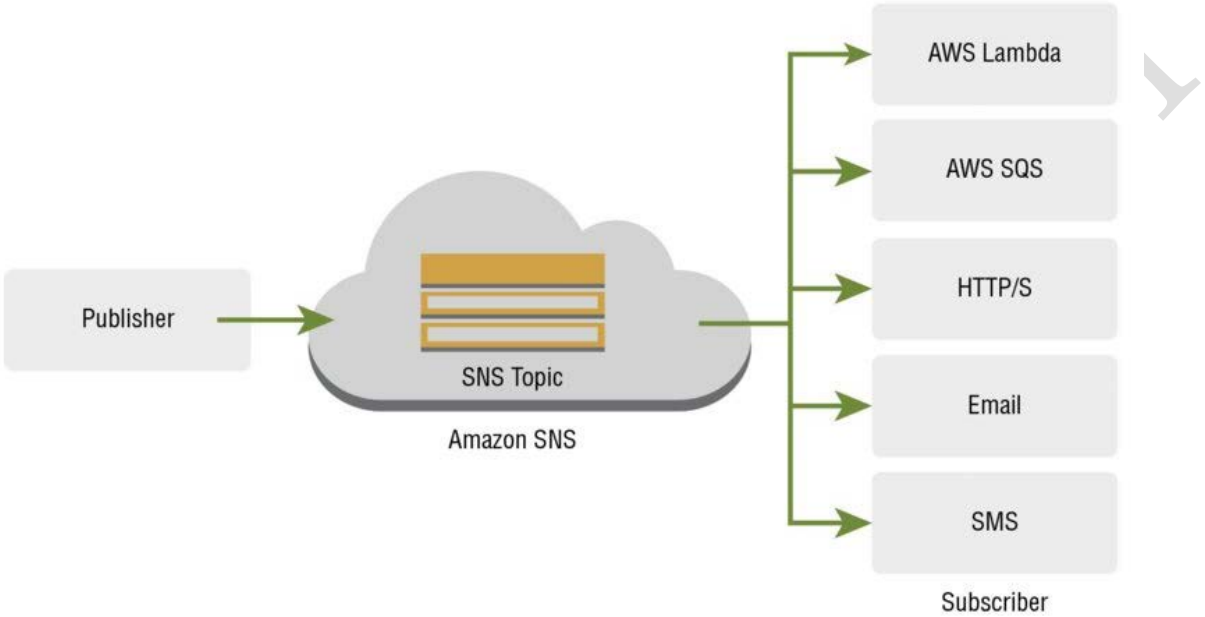
It provides developers with a highly scalable, flexible, and cost-effective capability to publish messages from an application and immediately deliver them to subscribers or other applications.

Push notifications to Apple, Google, Fire OS, and Windows devices, as well as Android devices in China with Baidu Cloud Push.

Amazon SNS consists of two types of clients: publishers and subscribers (sometimes knownas producers and consumers).

- Publishers communicate to subscribers asynchronously by sending a message to a topic.
- A topic is simply a logical access point/communication channel that contains a list of subscribers and the methods used to communicate to them.
- When you send a message to a topic, it is automatically forwarded to each subscriber of that topic using the communication method configured for that subscriber.

Besides pushing cloud notifications directly to mobile devices, Amazon SNS can also deliver notifications by SMS text message or email, to Amazon Simple Queue Service (SQS) queues, or to any HTTP endpoint.



To prevent messages from being lost, all messages published to Amazon SNS are stored redundantly across multiple availability zones.
SNS allows you to group multiple recipients using topics. A topic is an "access point" for allowing recipients to dynamically subscribe for identical copies of the same notification.

Application and System Alerts
Application and system alerts are SMS and/or email notifications that are triggered bypredefined thresholds. For example, we can receive immediate notification when an event occurs, such as a specific change to yourAuto Scaling group in AWS.

Push Email and Text Messaging
Push email and text messaging are two ways to transmit messages to individuals or groups via email and/or SMS. For example, you can use Amazon SNS to push targeted newsheadlines to subscribers by email or SMS. Upon receiving the email or SMS text, interestedreaders can then choose to learn more by visiting a website or launching an application.

Mobile Push Notifications
Mobile push notifications enable you to send messages directly to mobile applications. Forexample, you can use Amazon SNS for sending notifications to an application, indicating thatan update is available. The notification message can include a link to download and install the update.

SNS Benefits

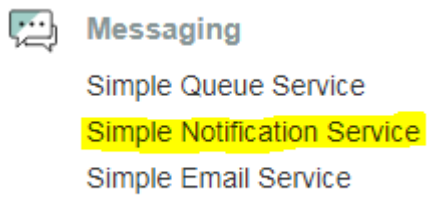
- Instantaneous, push-based delivery (no polling)
- Simple APIs and easy integration with applications
- Flexible message delivery over multiple transport protocols
- Inexpensive, pay-as-you-go model with no up-front costs
- Web-based AWS Management Console offers the simplicity of a
- point-and-click interface

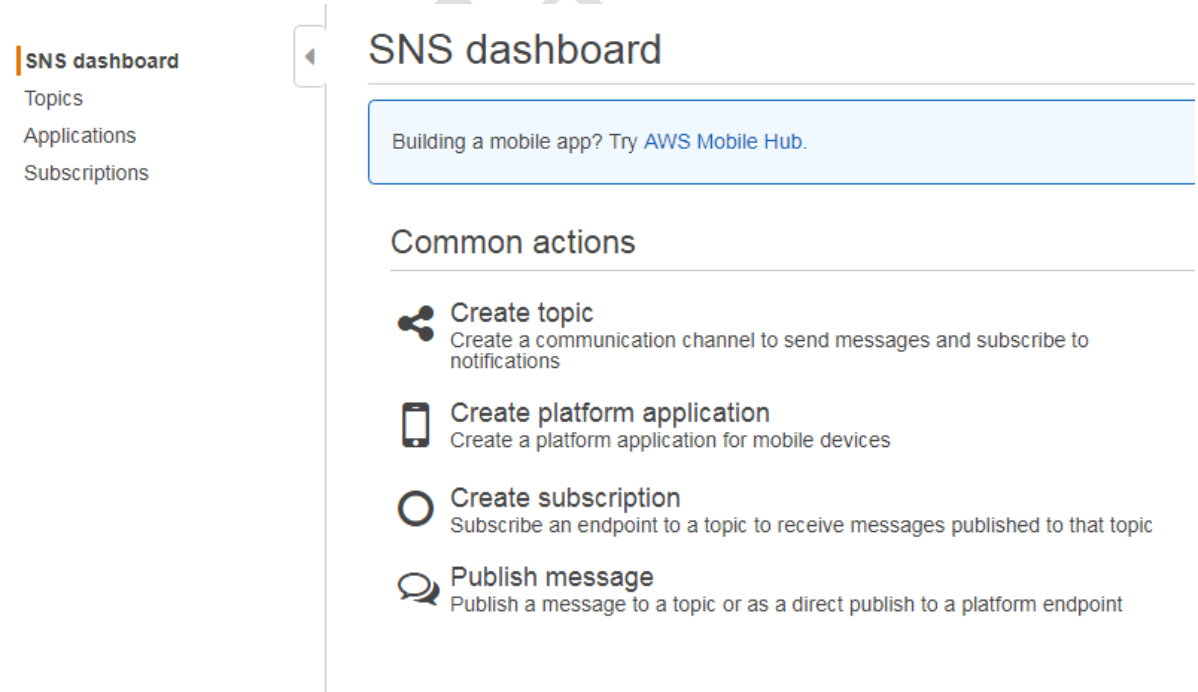
SNS vs SQS

- Both Messaging Services in AWS
- SNS - Push
- SQS - Polls (Pulls)

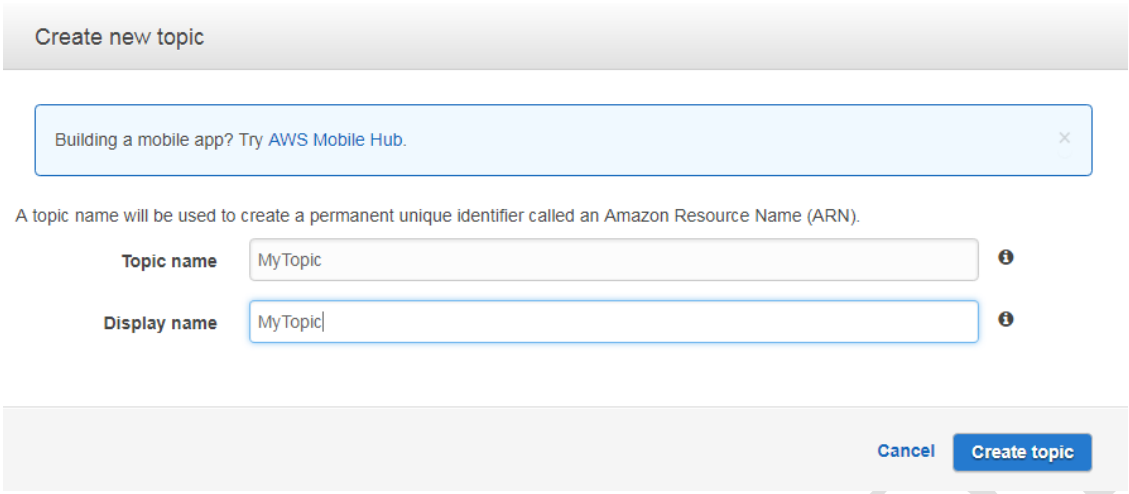
Creating SNS Topic and Publishing:

- Sign in to AWS account and Navigate to Mobile Services and then Amazon SNS to load the Amazon SNS dashboard.

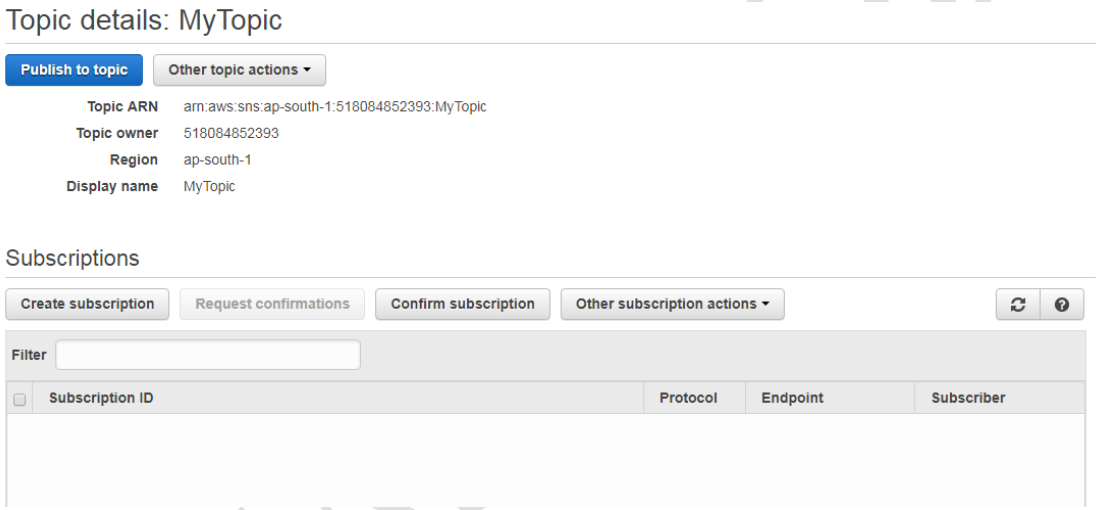




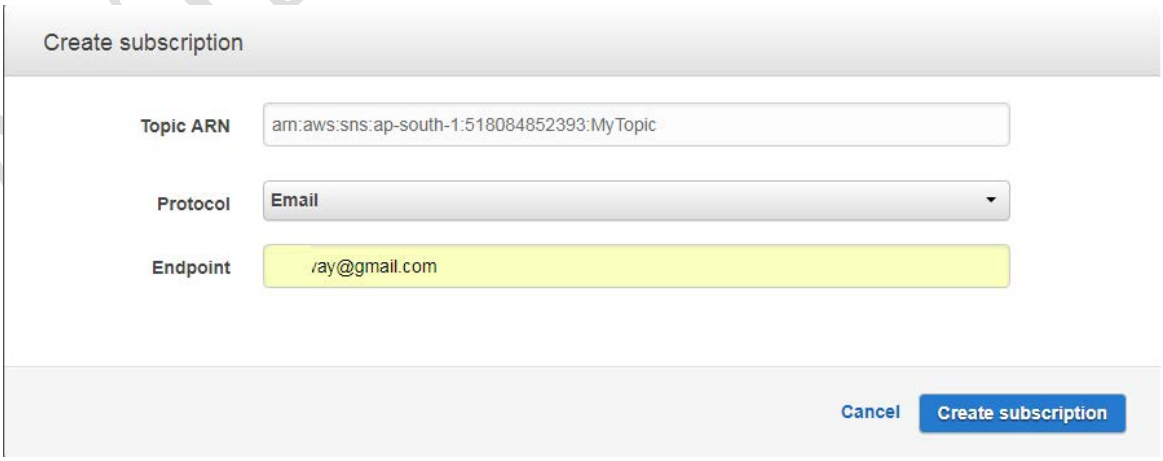
- Create a new topic by selecting "**Create topic"** option, and give a name for Topic and Display Name.



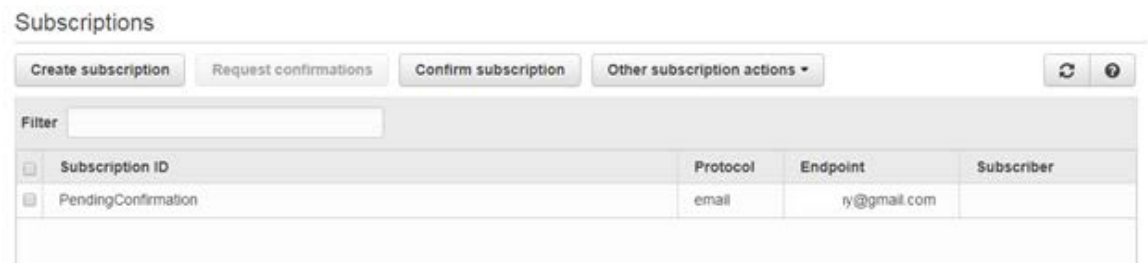
- Here is Topic Details screen



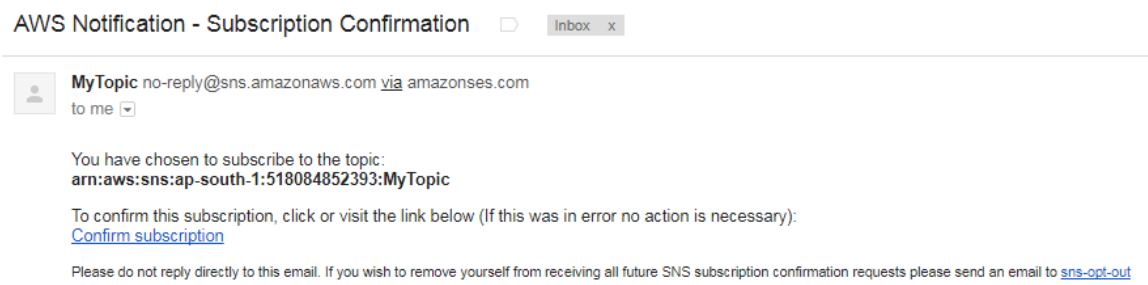
- We can publish to Topic, but we don't have any Subscribers to this topic, we need to add the subscribers then we can publish to all the Subscribers at a time.
  - Click on Create Subscription option and choose the Protocol as Email and Enter the Email ID you want to subscribe to this topic.



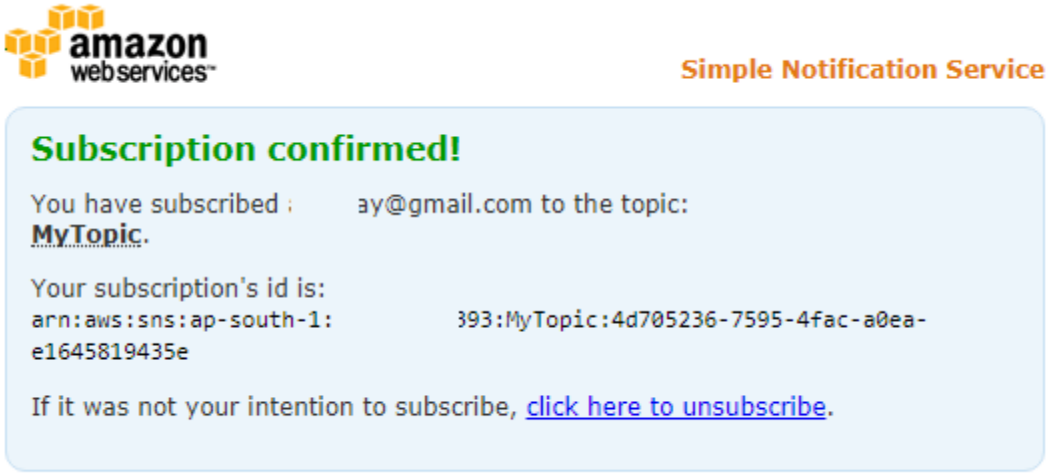
  - Here is the status below user subscribed to the topic.



- Now login to the mentioned Email ID and verify the Email from AWS SNS, and it'll ask you to subscribe to the topic.
- You'll get an Email as mentioned below image.

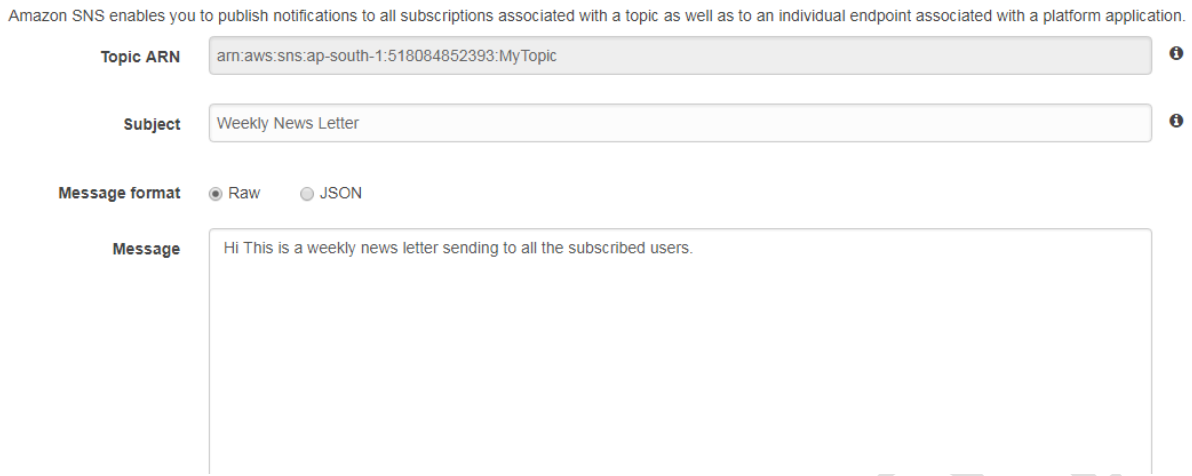


- Click on the Confirm Subscription Link, it'll redirect to another page which shows the subscription status page.

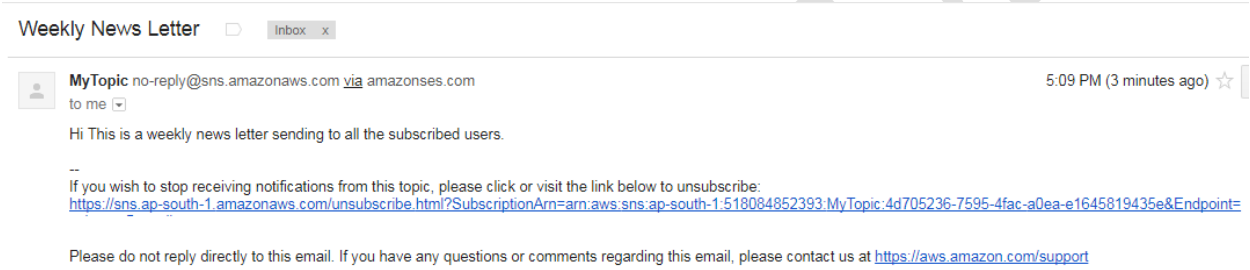


- Now we can publish to the Topic, all the subscribed users will get the email/notification.

➢ Now select the **"Publish to Topic"** Option Then provide the Subject to the email and enter the Message to send to all the subscribers.



- Give TTL value as 300 Seconds and click on Publish message, immediately all the subscribed users will get the email.



- We can unsubscribe to the Topic at any time, and in every email we'll get unsubscribed URL, we can click on that when we want to opt-out from the topic.

Security Options

Amazon CloudFront

Amazon CloudFront is a global Content Delivery Network (CDN) service. It integrates with other AWS products to give developers and businesses an easy way to distribute content to end users with low latency, high data transfer speeds, and no minimum usage commitments.

Amazon CloudFront is AWS CDN. It can be used to deliver your web content using Amazon's global network of edge locations. When a user requests content that you're serving with Amazon CloudFront, the user is routed to the edge location that provides the lowest latency (time delay), so content is delivered with the best possible performance. If the content is already in the edge location with the lowest latency, Amazon CloudFront delivers it immediately. If the content is not currently in that edge location, Amazon CloudFront retrieves it from the origin server, such as an Amazon Simple Storage Service (Amazon S3) bucket or a web server, which stores the original, definitive versions of your files.

Amazon CloudFront is optimized to work with other AWS cloud services as the origin server, including Amazon S3 buckets, Amazon S3 static websites, Amazon Elastic Compute Cloud (Amazon EC2), and Elastic Load Balancing. Amazon CloudFront also works seamlessly with any non-AWS origin server, such as an existing on-premises web server. Amazon CloudFront also integrates with Amazon Route 53.

Amazon CloudFront supports all content that can be served over HTTP or HTTPS. This includes any popular static files that are a part of your web application, such as HTML files, images, JavaScript, and CSS files, and also audio, video, media files, or software downloads. Amazon CloudFront also supports serving dynamic web pages, so it can actually be used to deliver your entire website. Finally, Amazon CloudFront supports media streaming, using both HTTP and RTMP.

Amazon CloudFront Basics

Below are the concepts, we can easily use CloudFront to speed up delivery of static content from your websites.

1. Distributions
2. Origins
3. Cache control.

Distributions: To use Amazon CloudFront, you start by creating a distribution, which is identified by a DNS domain name such as d111111abcdef8.cloudfront.net. To serve files from Amazon CloudFront, you simply use the distribution domain name in place of your website's domain name; the rest of the file paths stay unchanged.

Origins: When you create a distribution, you must specify the DNS domain name of the origin-the Amazon S3 bucket or HTTP server-from which you want Amazon CloudFront to get the definitive version of your objects (web files).

CacheControl: Once requested and served from an edge location, objects stay in the cache until they expire. By default,objects expire from the cache after24hours.

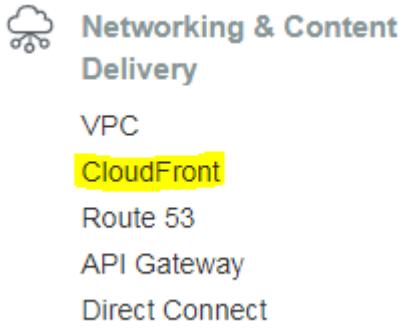
SignedURLs Use URLs that are valid only between certaintimes and optionally from certain IP addresses.

SignedCookies Require authentication via public and private keypairs.

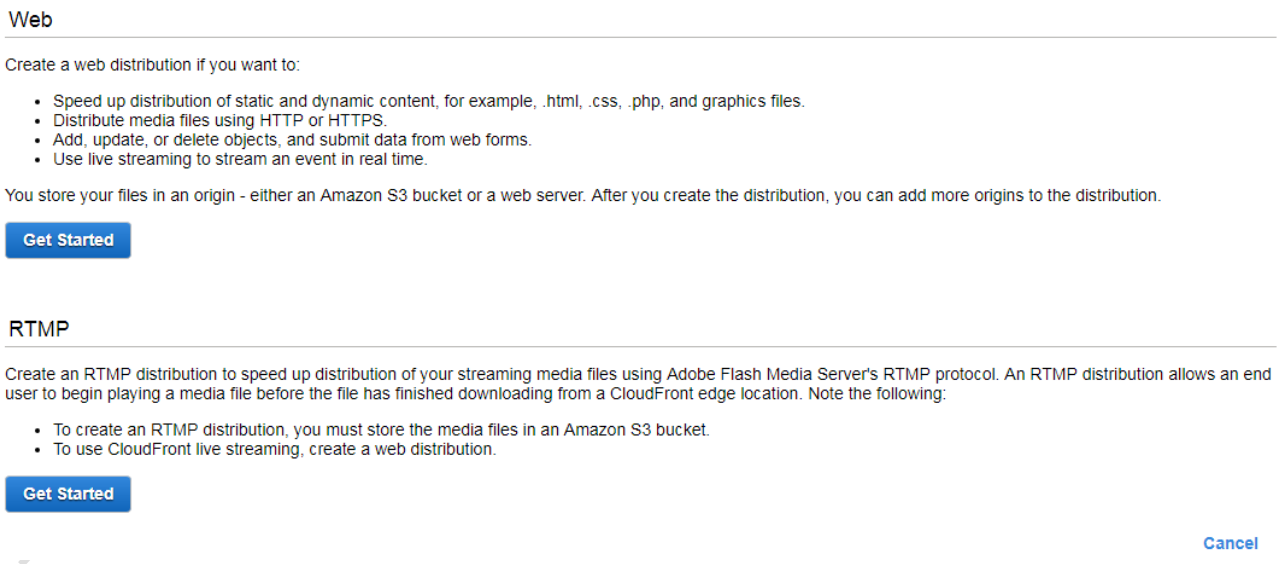
Origin Access Identities(OAI): Restrict access to an Amazon S3 bucket only to a special Amazon Cloud Front user associated with your distribution. This is the easiest way to ensure that content in a bucket is only accessed by Amazon CloudFront.

Creating a Cloudfront Distribution: (Mostly am choosing all the default options)

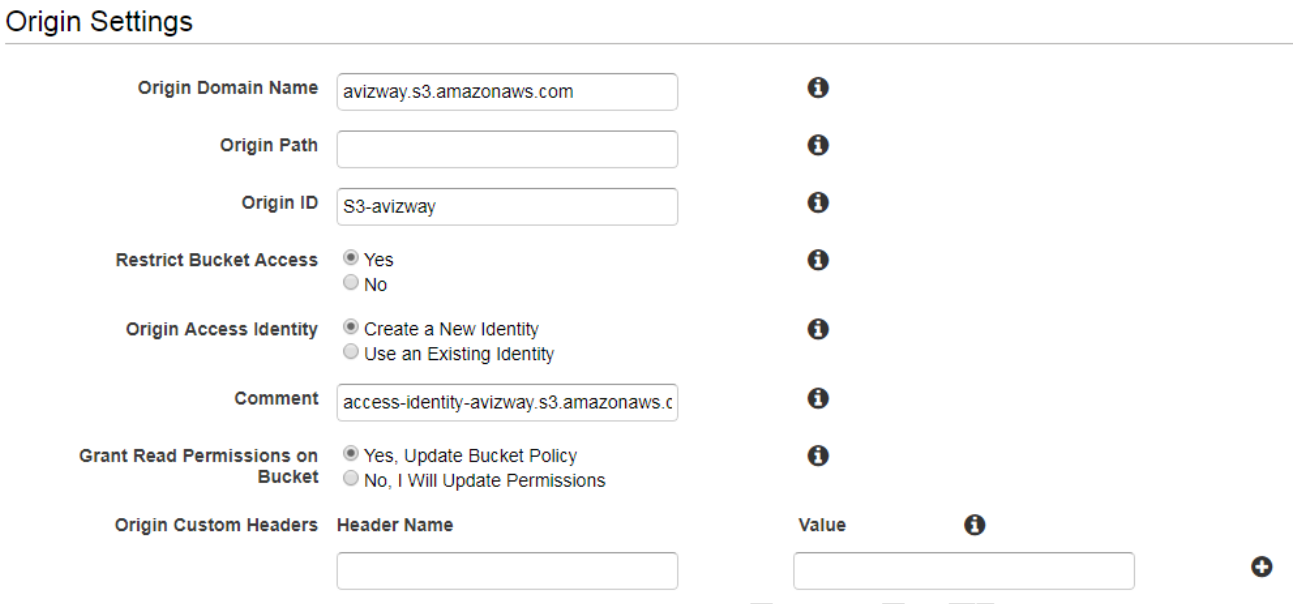
1. We can find the CloudFront distribution under **"Network & Content Delivery"**



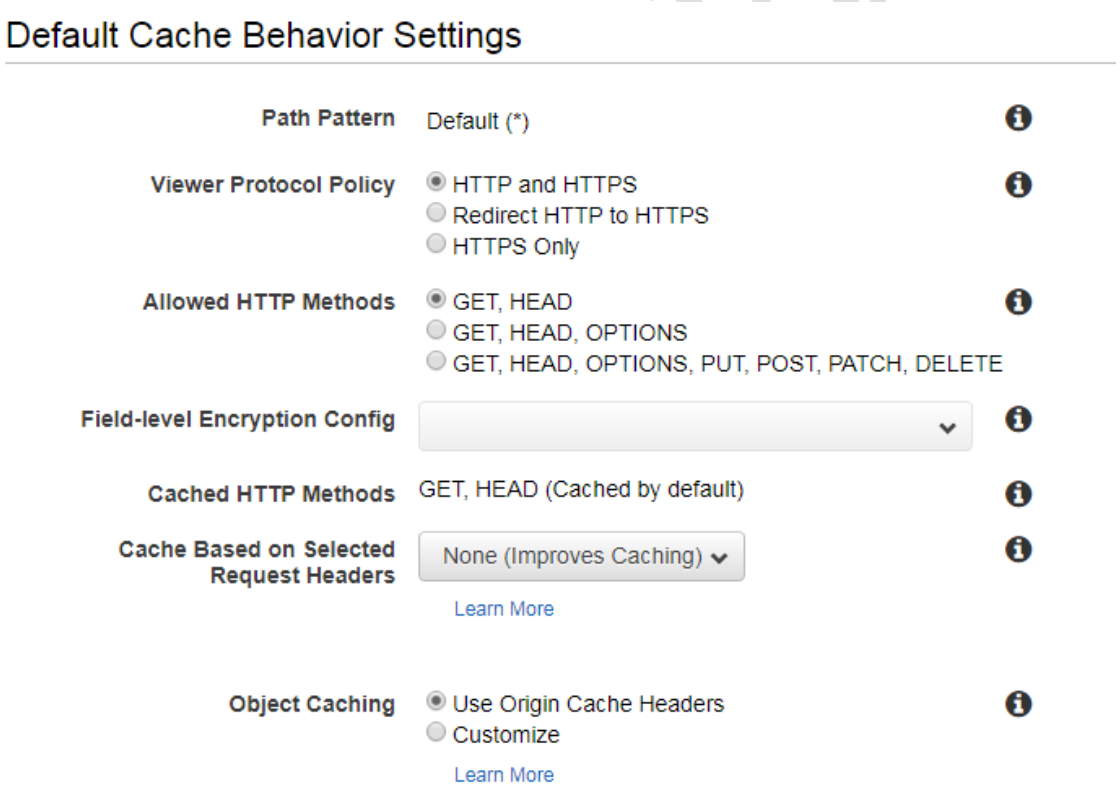
2. Choose a delivery method for content (Web or RTMP).

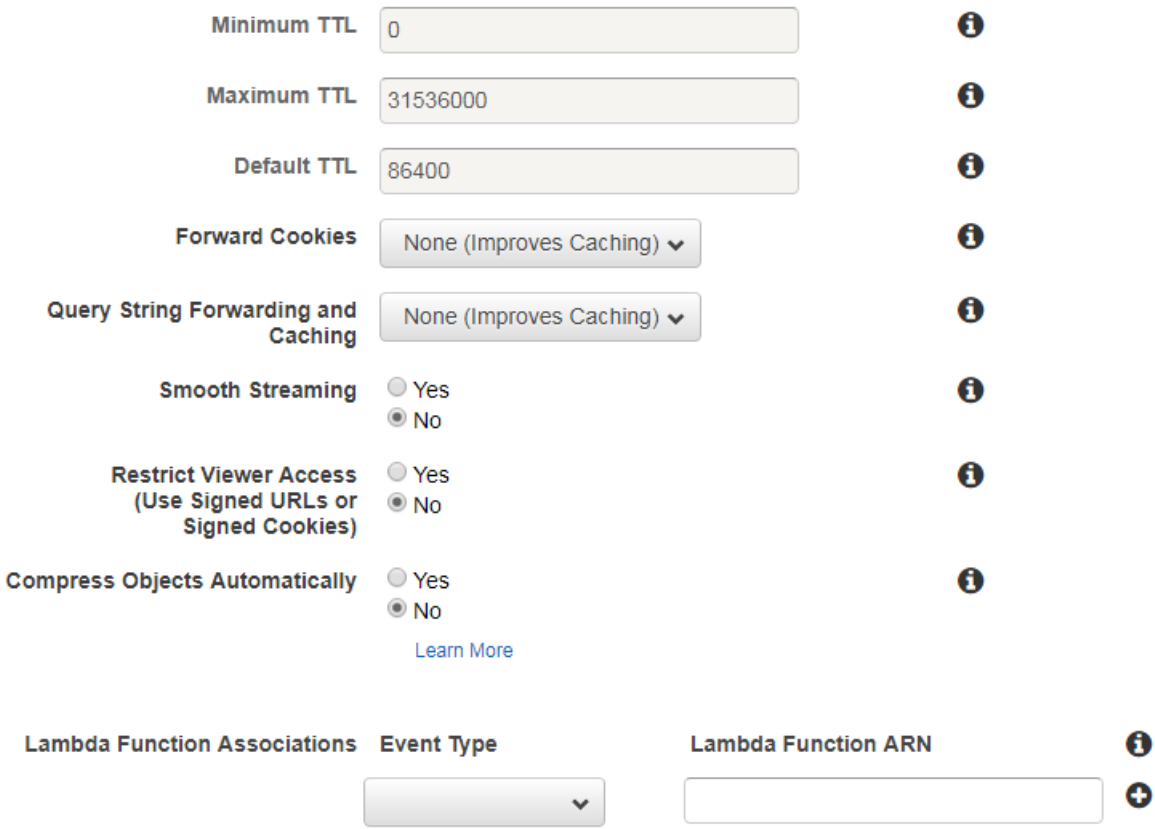


3. Choose the **"Origin Settings"**as below.

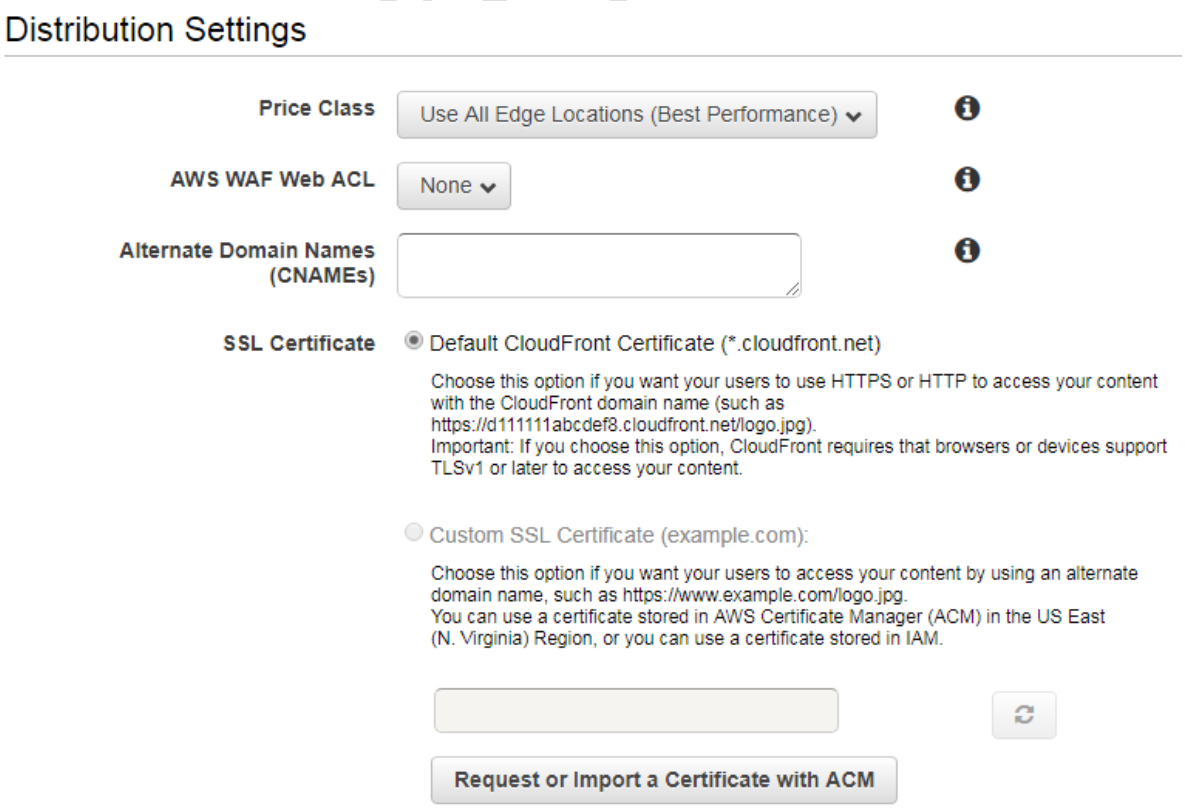


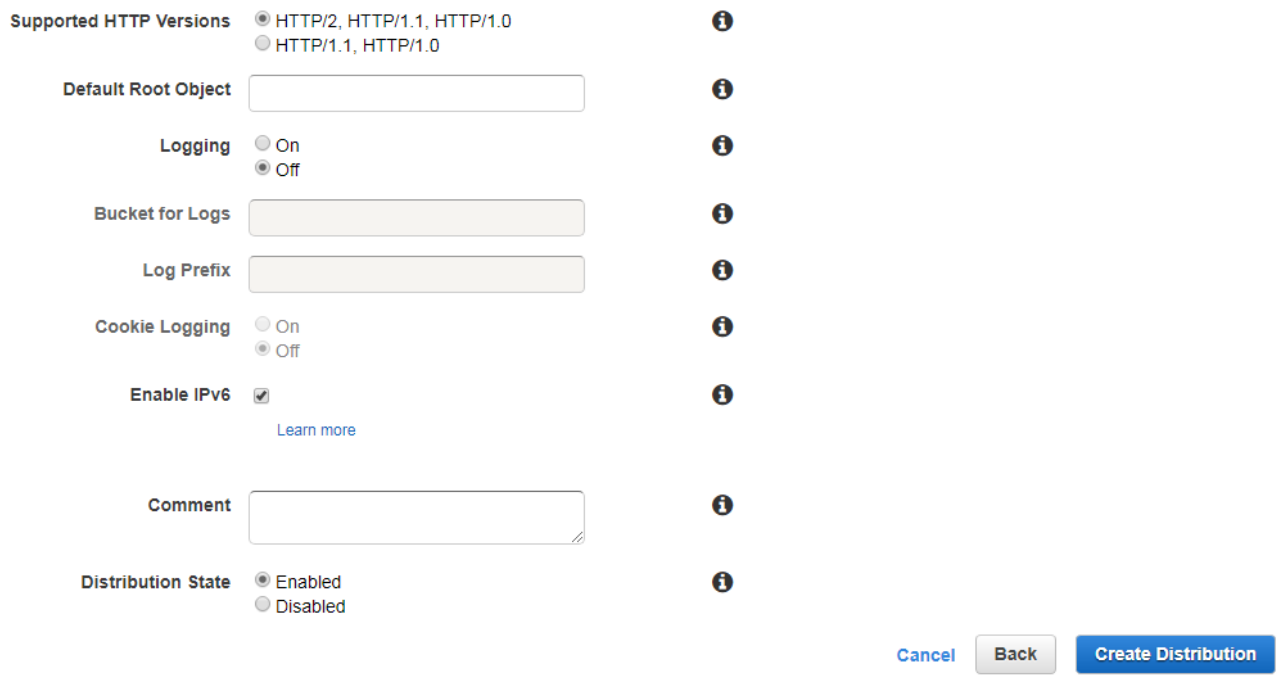
4. Choose the Default Chache Behaviour Settings.





5. Choose the Distribution Settings





6. For Cloudfront we will get a domain name In this format http://d111111abcdef8.cloudfront.net/. We can access the Objects with Cloudfront distribution, the objects are going to deliver from near by edge location.

AWS StorageGateway

AWS Storage Gateway is a service connecting an on-premises software appliance with cloud-based storage to provide seamless and secure integration between an organization's on-premises IT environment and AWS storage infrastructure.

The service enables you to store data securely on the AWS cloud in a scalable and cost-effective manner. AWS Storage Gateway supports industry-standard storage protocols that work with your existing applications. It provides low-latency performance by caching frequently accessed data on-premises while encrypting and storing all of your data in Amazon S3 or Amazon Glacier.

File gateway
Store files as objects in Amazon S3, with a local cache for low-latency access to your most recently used data.

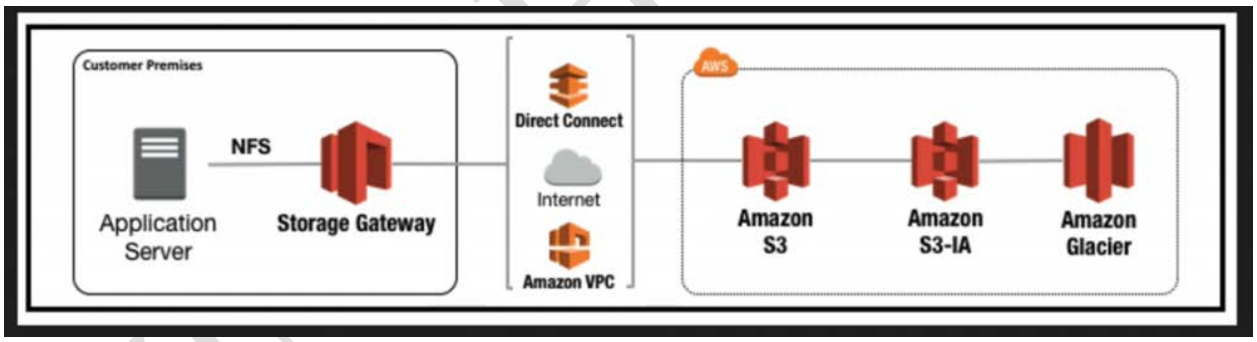
Volume gateway
Block storage in Amazon S3 with point-in-time backups as Amazon EBS snapshots.

- **Cached volumes**
  Low-latency access to your most recently used data.
- **Stored volumes**
  On-premises data with scheduled offsite backups.

Tape gateway
Back up your data to Amazon S3 and archive in Amazon Glacier using your existing tape-based processes.

Mainly we have three types of Gateways:

1. **File gateway:**Store files as objects in Amazon S3, with a local cache for low-latency access to your most recently used data.All the files are stored directly on S3 we can access through NFS mount points.

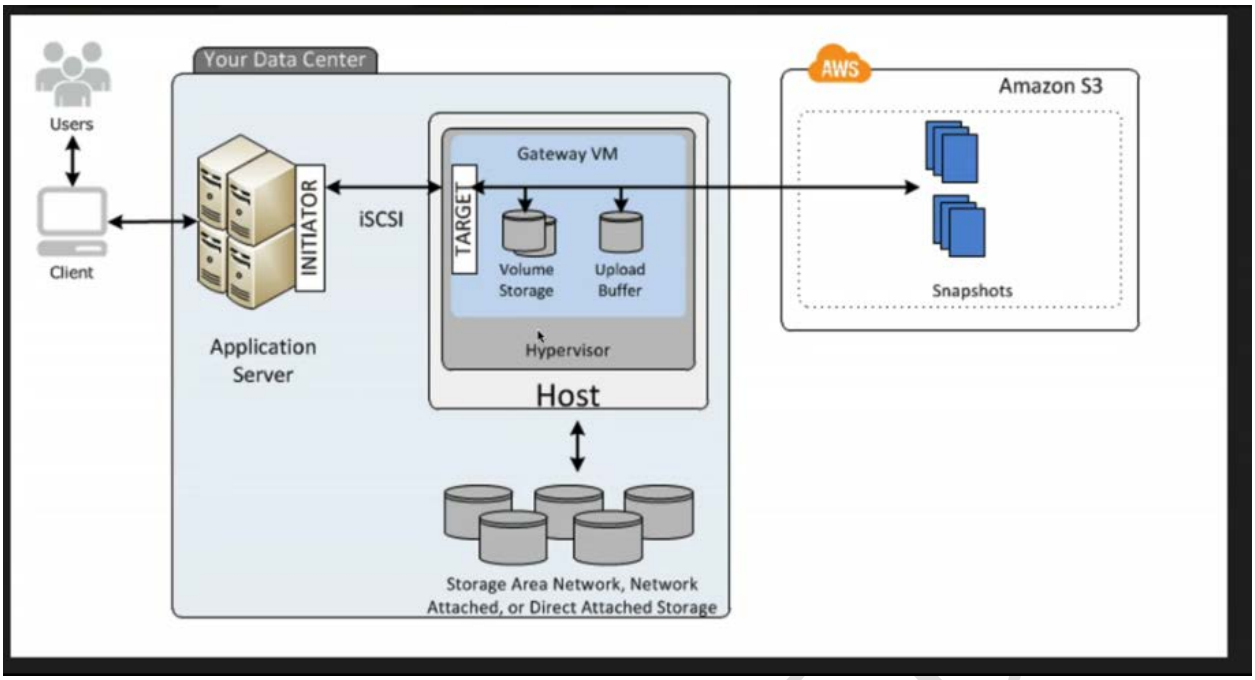


2. **Volume Gateway:** The volume interface presents your applications with disk volumes using the iSCSI block protocol.

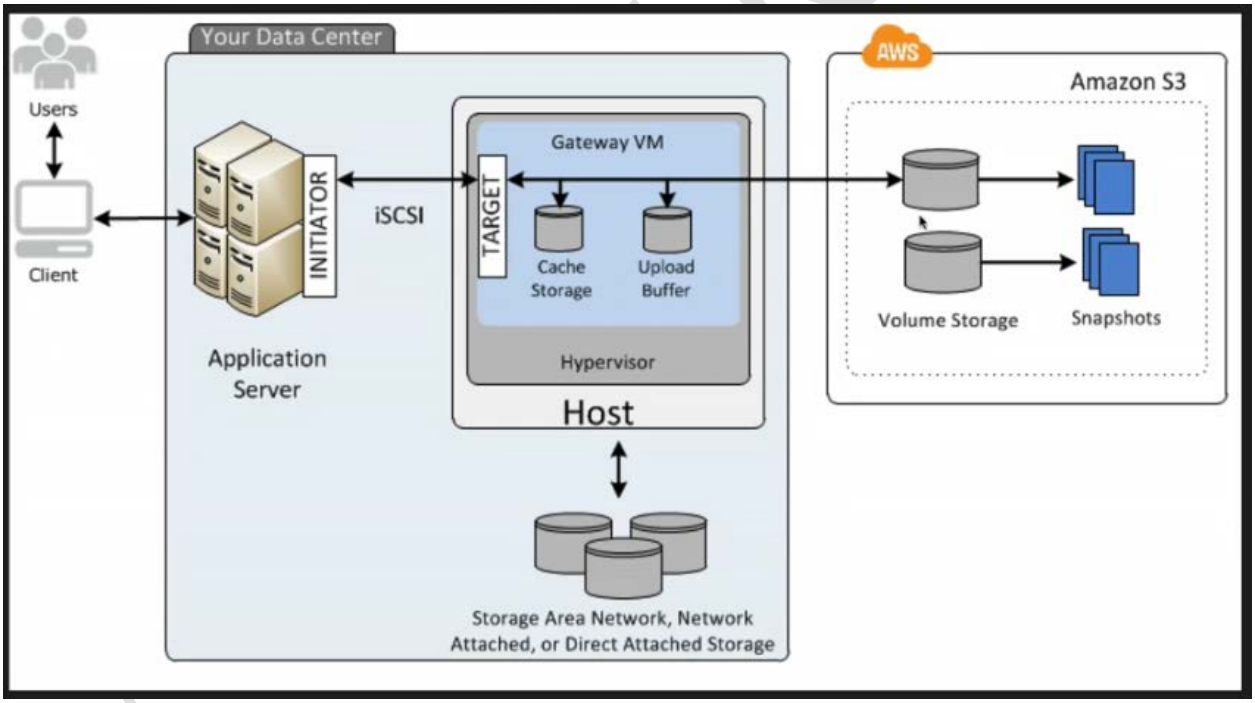
   Data written to these volumes can be asynchronously backed up as point-in-time snapshots of your volumes, and stored in the cloud as Amazon EBS snapshots.

   Snapshots are incremental backups that capture only changed blocks. All snapshot storage is also compressed to minimize your storage charges.

   Volume gateway Stored Volumes – Entire primary Dataset is stored locally and data is asynchronously backed up to S3 in form of Amazon EBS snapshots (1 GB – 16 TB in size for stored volumes).



Volume gateway Cached Volumes - Entire Dataset is stored on S3 and the most frequently accessed data is cached on site. You can create storage volumes up to 32 TB in size. Recent modified data will cache on premise storage gateway's cache.



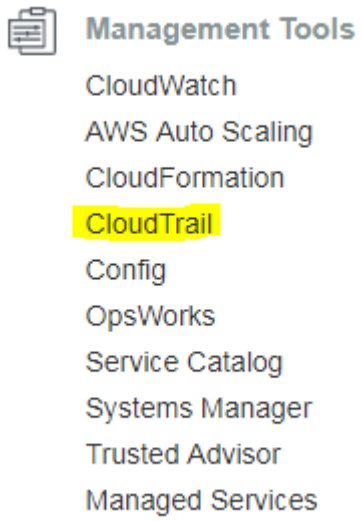
Tape Gateway : Back up your data to Amazon S3 and archive in Amazon Glacier using your existing tape-based processes. Supports popular backup applications like NetBackup, Backup Exec, Veeam etc.

AWS CloudTrail:

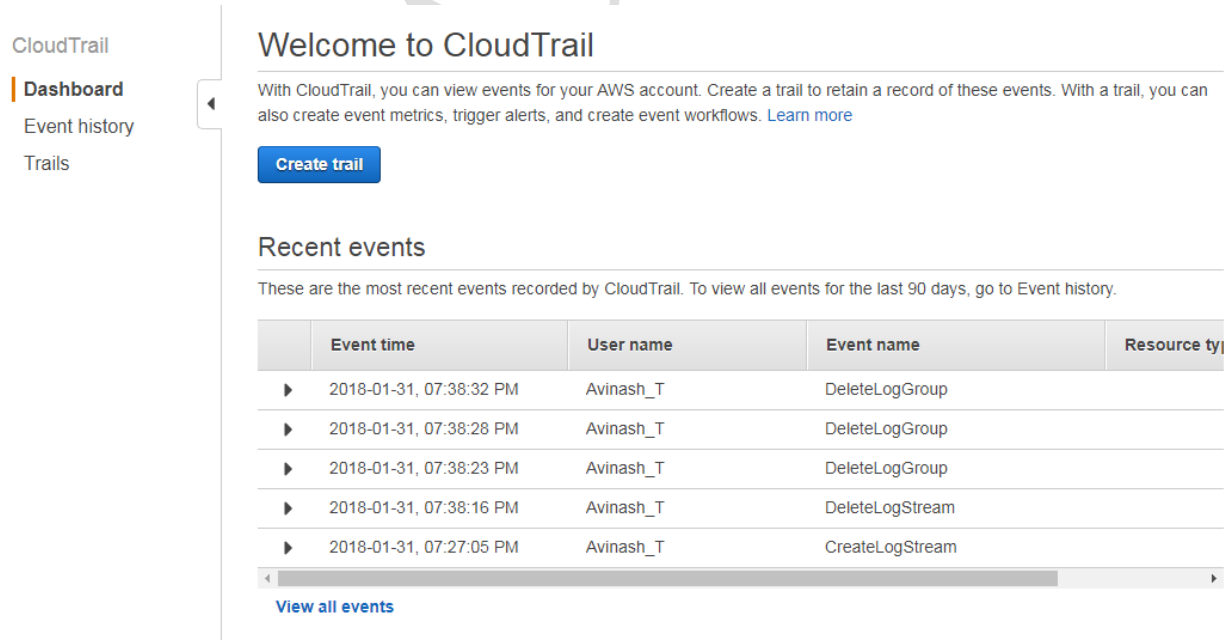
AWS CloudTrail is a service that enables governance, compliance, operational auditing, and risk auditing of your AWS account. With CloudTrail, you can log, continuously monitor, and retain account activity related to actions across your AWS infrastructure.

CloudTrail provides event history of your AWS account activity, including actions taken through the AWS Management Console, AWS SDKs, command line tools, and other AWS services. This event history simplifies security analysis, resource change tracking, and troubleshooting.

We can find the Cloudtrail under Management Tool in AWS dashboard.



Here is the cloudTrail dashboard, By default we can view the last 90 days all events here.

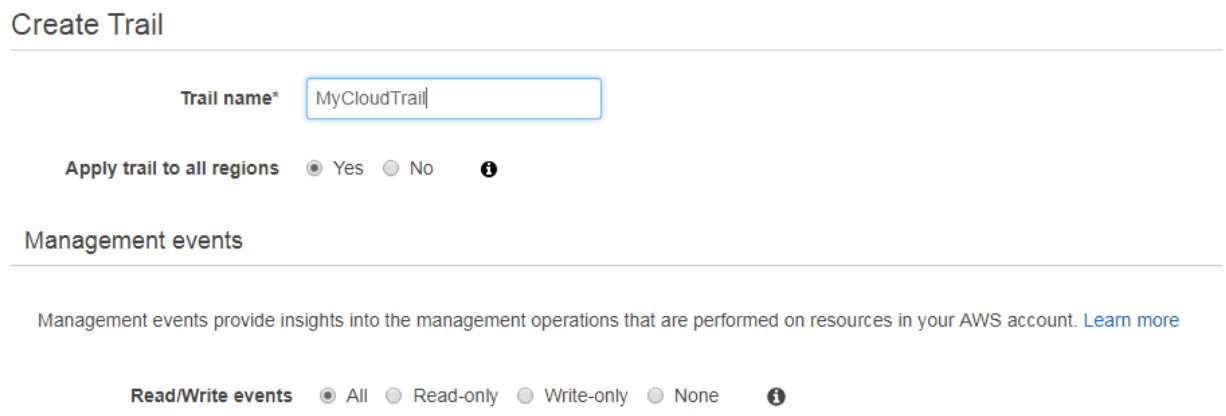


If you want to store the logs more than 90 days, we need to create a Trail and need to copy into S3 bucket.

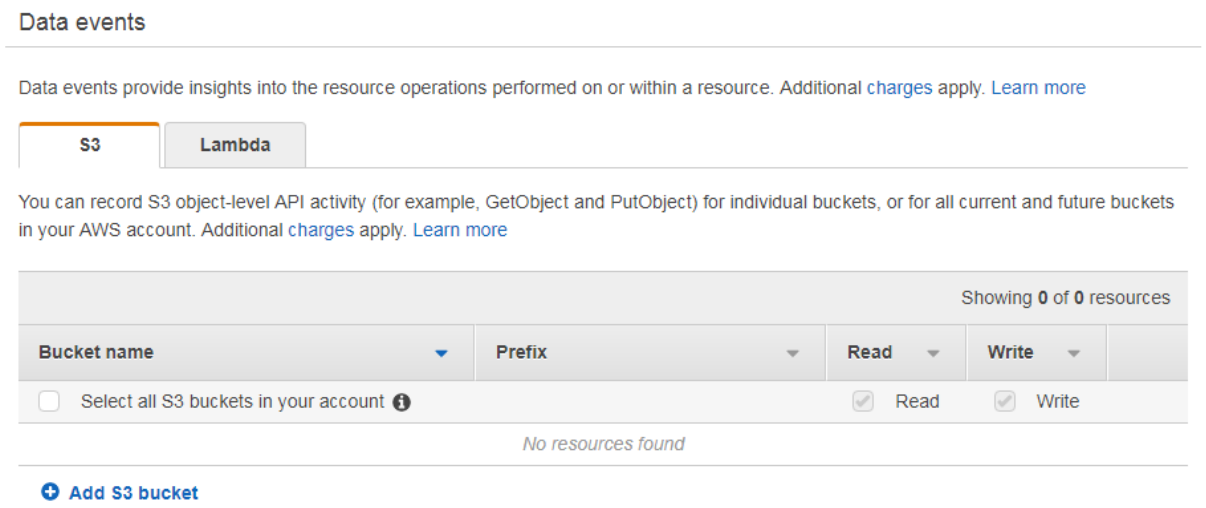
Select the **"Create trail"**option to start. And give a Trail Name

Select Yes option if you want to apply this trail to all the regions.

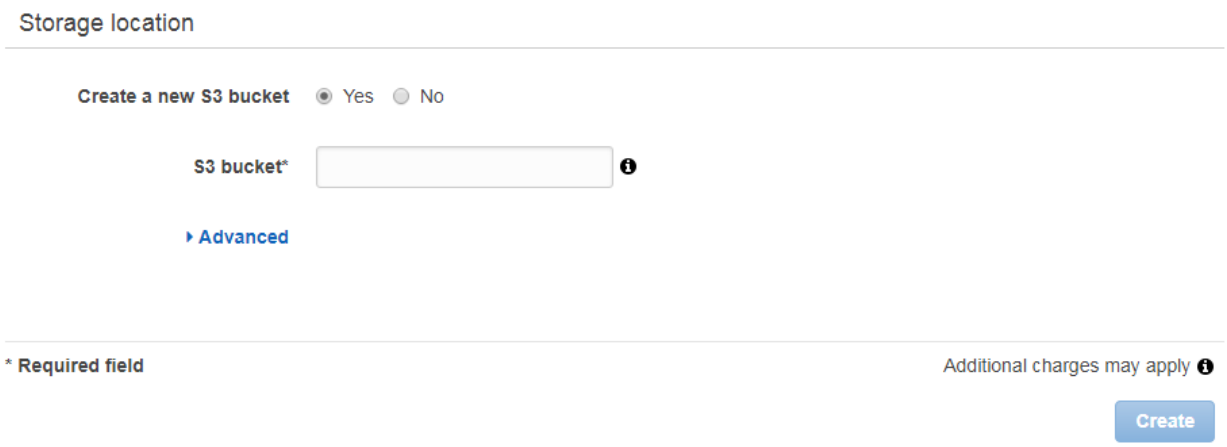
Choose the Management events you want to track (All/Read-Only/Write-only/None)



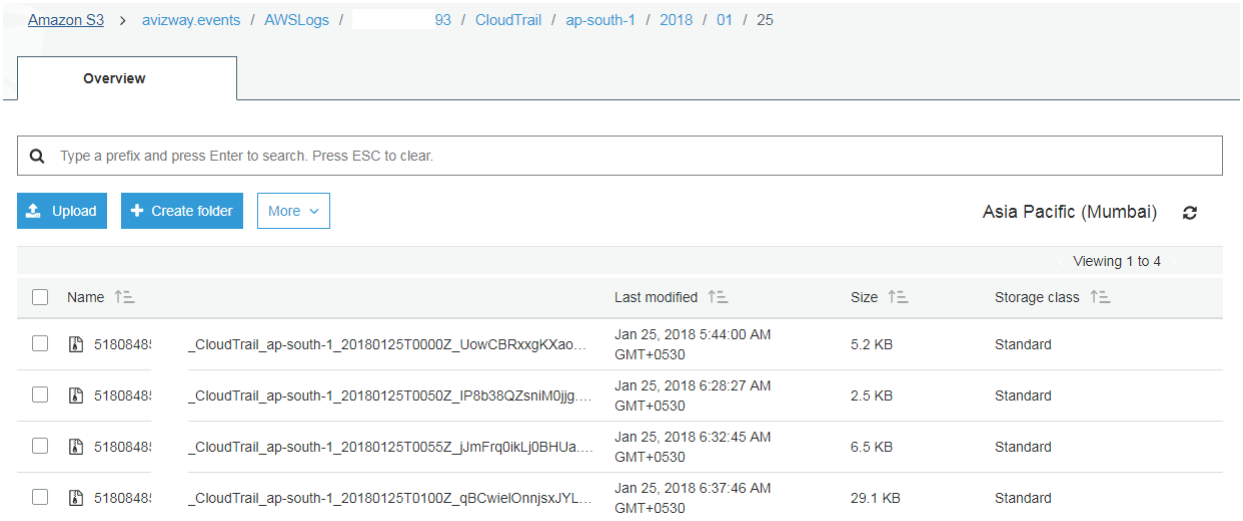
Select the Data events if required (Additional Charges apply for this service)



Then choose the Storage location, CloudTrail logs will store in s3 buckets. We can choose an existing bucket or create a new bucket. Select yes if you want to create a new bucket or No to choose an existing bucket from your AWS account.



Our cloudTrail log is successfully created. Let's navigate to S3 bucket to verify the logs. Logs will store Region wise and after that Year, Month and Date.



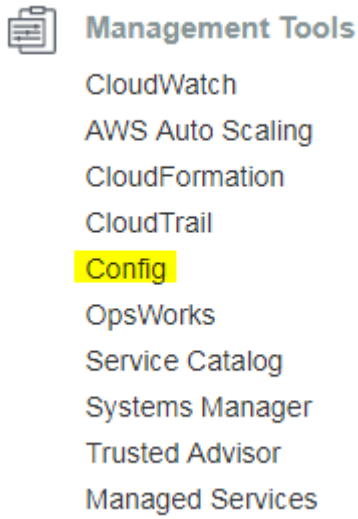
Every Log Contains the below data:

1. Metadata around API calls
2. The identity of the API caller
3. The time of the API call
4. The source IP address of the API caller
5. The request parameters
6. The response elements returned by the service.

AWS Config:

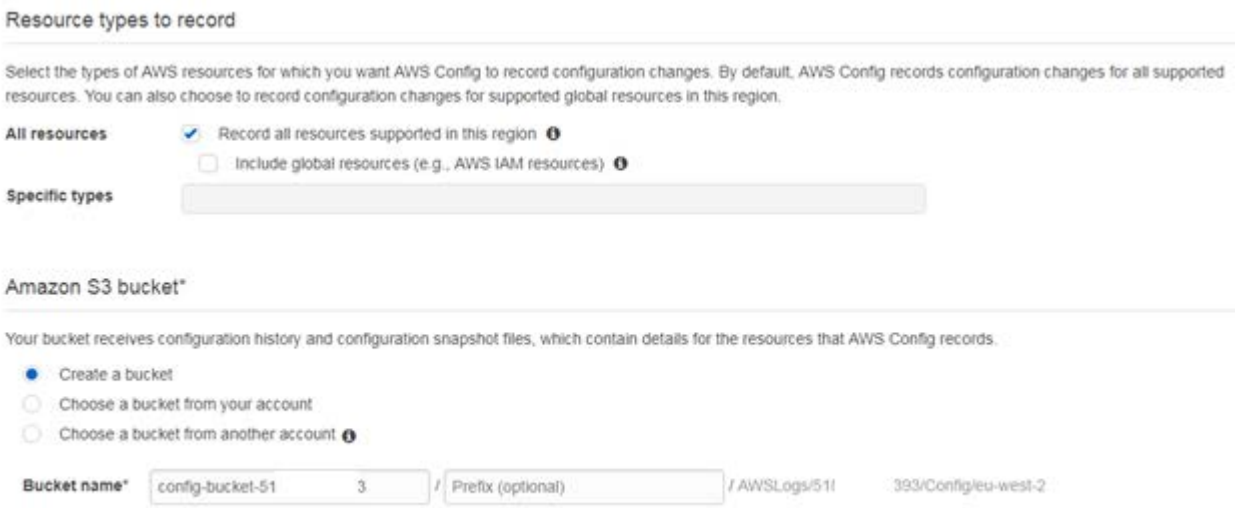
AWS Config is a fully managed service that provides you with an AWS resource inventory, configuration history, and configuration change notifications to enable security and governance. With AWS Config, you can discover existing and deleted AWS resources, determine your overall compliance against rules, and dive into configuration details of are source at any point in time. These capabilities enable compliance auditing, security analysis, resource change tracking, and troubleshooting.

You will find the "Config" service under Management Tools.

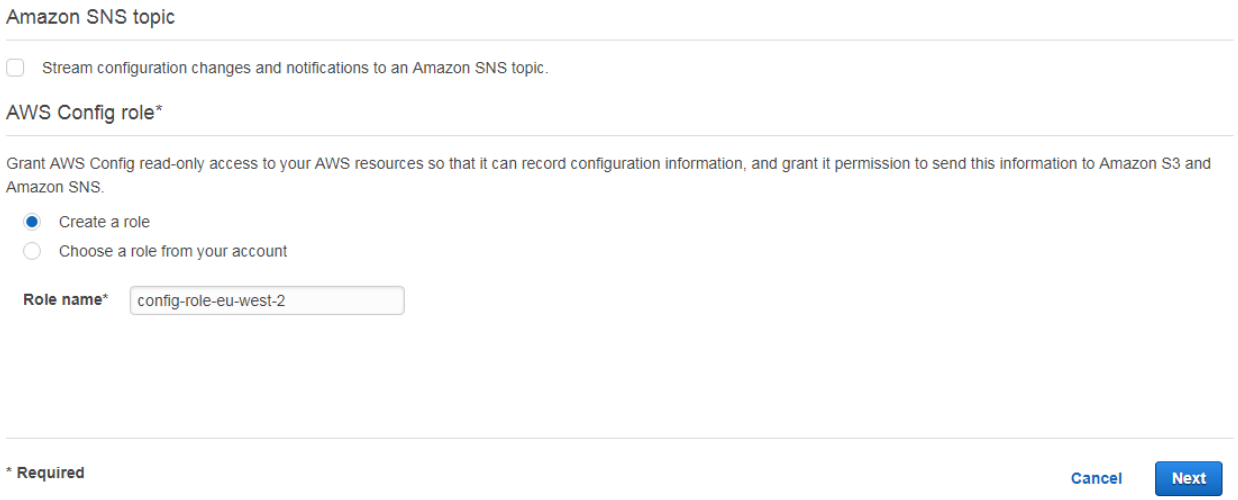


When you navigate to Config for the first time, it'll ask you to setup the AWS config. Here is the steps to configure the AWS config.

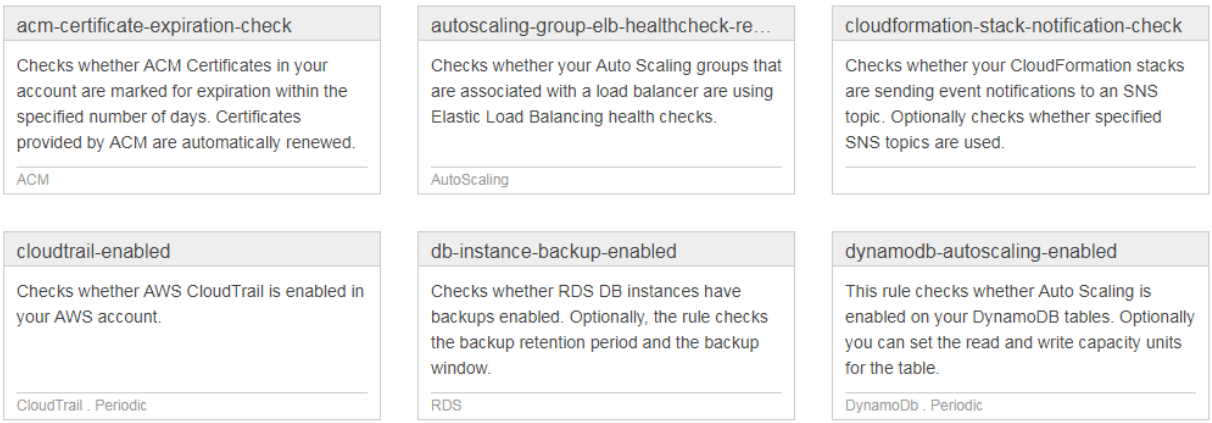
1. Choose what resource types to record with AWS config.
    a. You can choose all the resources in Selected region and even you can choose global resources i.e; S3, IAM
2. Choose the S3 bucket to store all the logs for the AWS Config. You can opt to create a new bucket or choose an existing bucket.



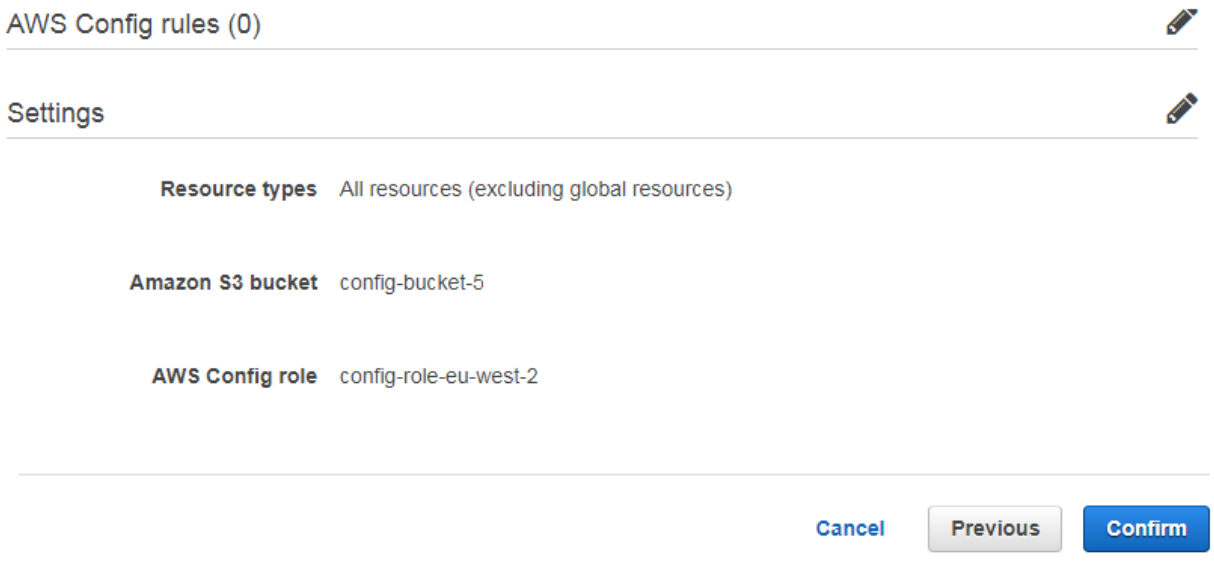
3. Choose an SNS topic to get notification and create an IAM role to perform the tasks on-behalf of us then click on **"Next"**



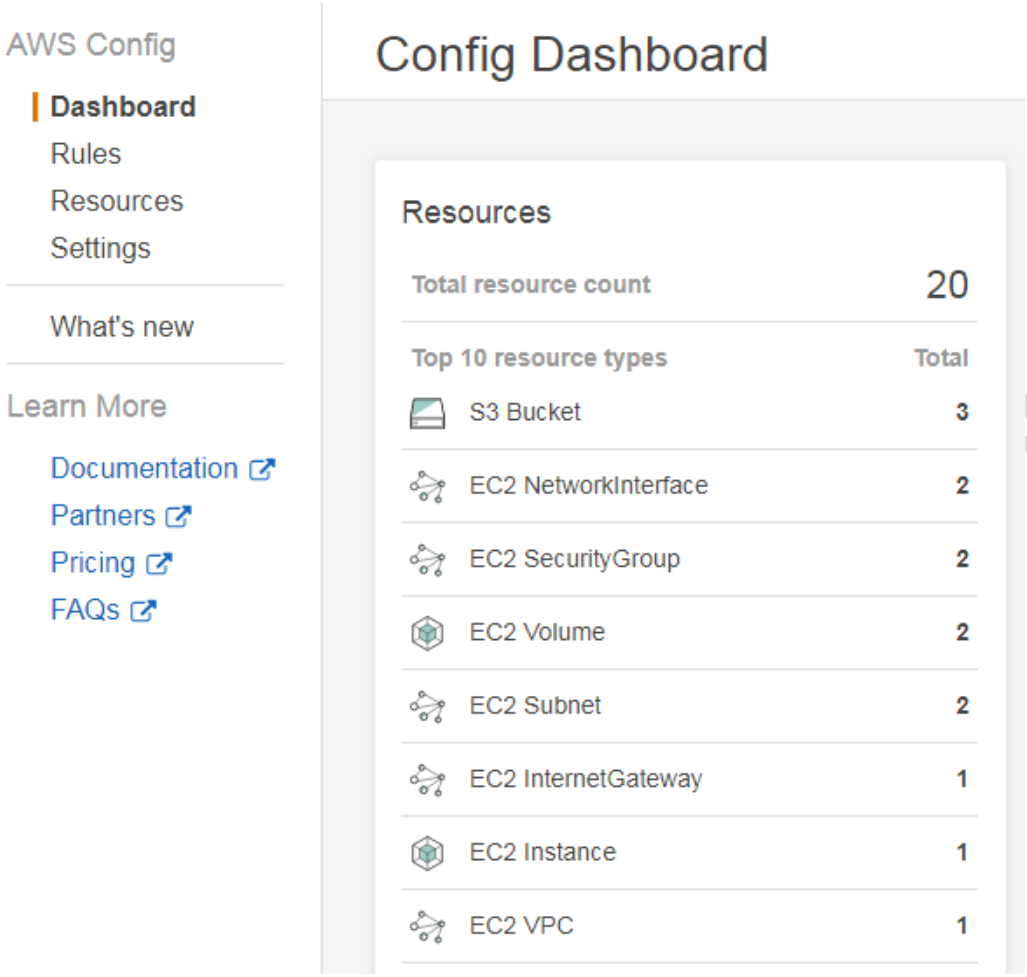
4. If you want to monitor any specific rule, you can select, otherwise you can choose or skip it.



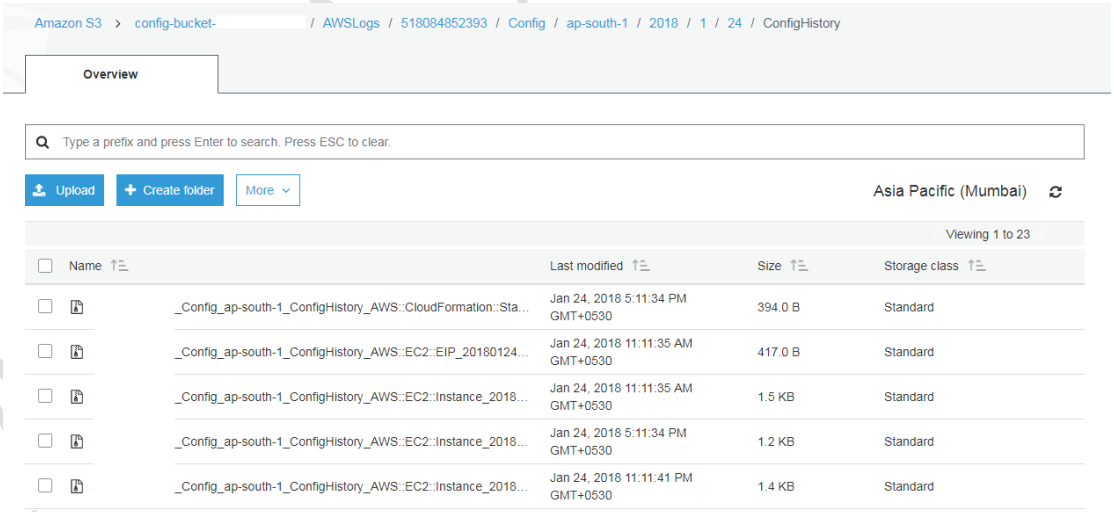
5. Review and click on confirm to complete the AWS config service setup.



6. Here is the Config service dashboard, you can choose the specific service and get the details about the changes, events happened.



7. Let me navigate to S3 bucket to verify the logs, Log path looks similar to CloudTrail path.



We can see the below details with AWS Config service:

1. Resource Type
2. Resource ID
3. Compliance
4. Timeline
    a. Configuration Details
    b. Relationships
    c. Changes
    d. CloudTrail Events

Amazon Kinesis

Amazon Kinesis is a platform for handling massive streaming data on AWS, offering powerful services to make it easy to load and analyze streaming data and also providing the ability for you to build custom streaming data applications for specialized needs.

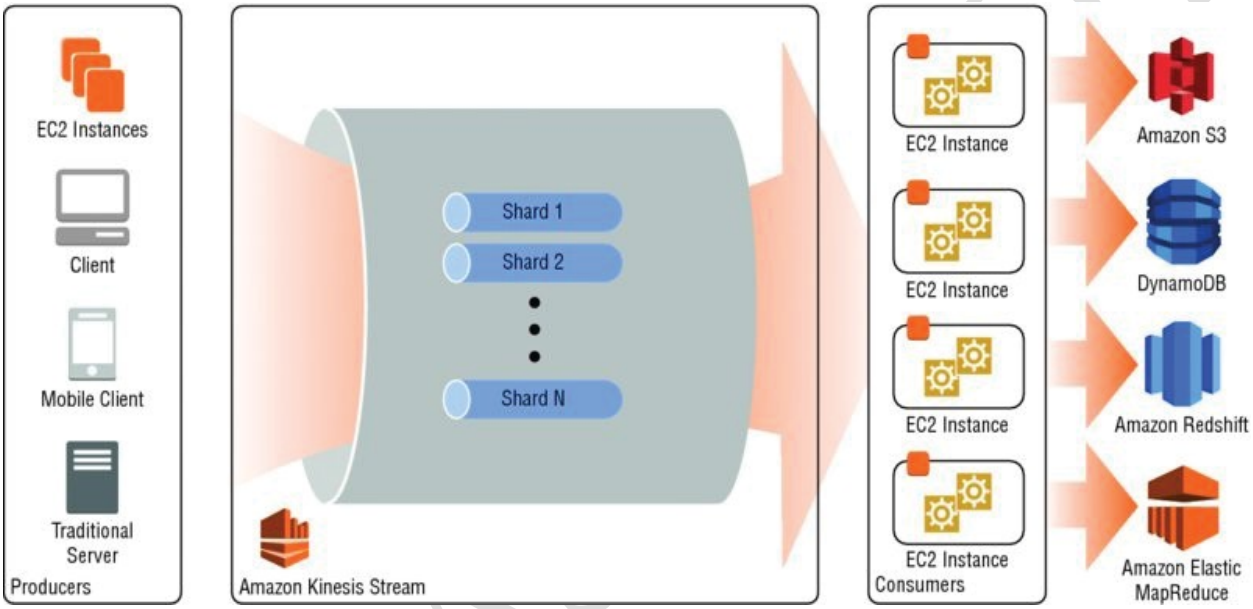
Amazon Kinesis is a streaming data platform consisting of three services addressing different real-time streaming data challenges:

Amazon Kinesis Firehose: This service enabling you to load massive volumes of streaming data into AWS.

Amazon Kinesis Firehouse receives stream data and stores it in AmazonS3, Amazon Redshift, or Amazon Elastic search. You do not need to write any code; just create a delivery stream and configure the destination for your data. Clients write data to the stream using an AWS API call and the data is automatically sent to the proper destination.

Amazon Kinesis Streams: A service enabling you to build custom applications for more complex analysis of streaming data in realtime.

Amazon Kinesis Streams enable you to collect and process large streams of data records in realtime. Using AWS SDKs, you can create an Amazon Kinesis Streams application that processes the data as it moves through the stream. Because responsetime for data intake and processing is in near realtime, the processing is typically light weight. Amazon Kinesis Stream scanscale to support nearly limitless data streams by distributing incoming data across a number of shards. If any shard becomes too busy, it can be further divided into more shards to distribute the load further. The processing is then executed on consumers, which read data from the shards and run the Amazon Kinesis Streams application.



Amazon Kinesis Analytics: A service enabling you to easily analyze streaming data real time with standard SQL.

Amazon Elastic Map Reduce(AmazonEMR)

Amazon Elastic Map Reduce (Amazon EMR) provides you with a fully managed, on-demand Hadoop framework. Amazon EMR reduces the complexity and up-front costs of setting up Hadoopa nd, combined with the scale of AWS gives you the ability to spinup large Hadoop clusters instantly and start processing with in minutes.

UseCases for EMR:

Amazon EMR is well suited for a large number of use cases, including, but not limited to:

Log Processing: Amazon EMR can be used to process logs generated by web and mobile applications. Amazon EMR helps customers turn peta bytes of unstructured or semi-structured data in to useful insights about their applications or users.

Clickstream Analysis: Amazon EMR can be used to analyze *clickstream* data in order to segment users and understand user preferences. Advertisers can also analyze click streams and advertising impression logs to deliver more effective ads.

AWS Data Pipeline:

AWS Data Pipeline is a web service that helps you reliably process and move data between different AWS compute and storage services, and also on-premises data sources, at specified intervals. With AWS Data Pipeline, you can regularly access your data where it's stored, transform and process it at scale, and efficiently transfer the results to AWS services such as Amazon S3, Amazon Relational Database Service (AmazonRDS), Amazon Dynamo DB, and Amazon EMR.

AWS CloudFormation

AWS Cloud Formation is a service that helps you model and setup your AWS resources so that you can spend less time managing those resources and more time focusing on your applications that run in AWS. AWS CloudFormation allows organizations to deploy, modify, and update resources in a controlled and predictable way, in effect applying version control to AWS infrastructure the same way one would do with software.

Overview

AWS CloudFormation gives developers and systems administrators an easy way to create and manage a collection of related AWS resources, provisioning and updating them in an orderly and predictable fashion. When you use AWS CloudFormation, you work with templates and stacks.

Use Case

By allowing you to replicate your entire infrastructure stack easily and quickly, AWS CloudFormation enables a variety of use cases:

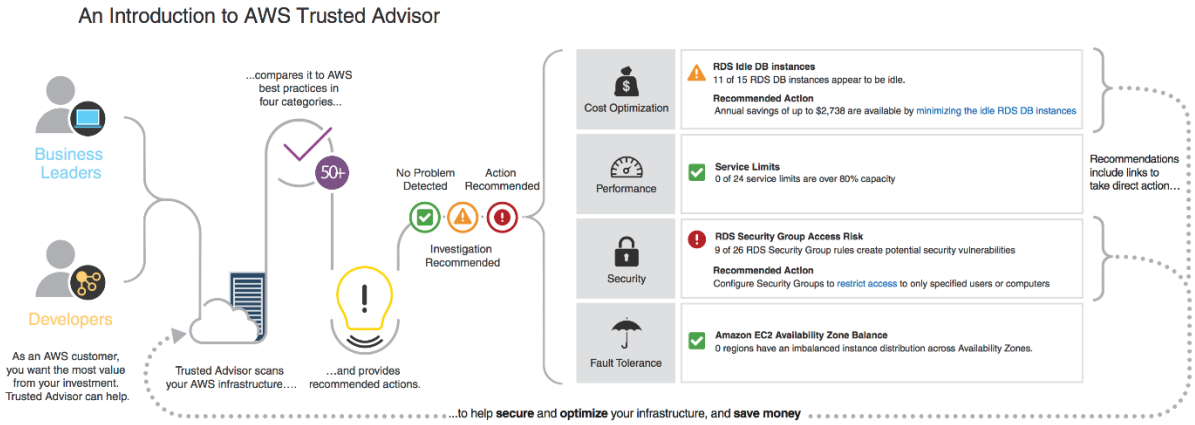
- **Quickly Launch New Test Environments:** AWS CloudFormation let's testing teams quickly create a clean environment to run tests without disturbing on going efforts in other environments.
- **Reliably Replicate Configuration:** between Environments Because AWS CloudFormation scripts the entire environment, human error is eliminated when creating new stacks.
- **Launch Applications in New AWS Regions:** A single script can be used across multiple regions to launch stacks reliably in different markets.

AWS Trusted Advisor:

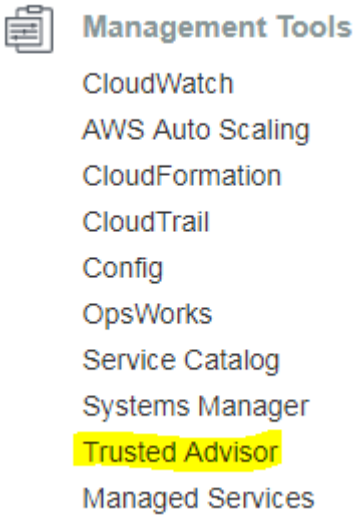
AWS trusted advisor is an online resource to help us to reduce cost, increase performance, and improve security by optimizing AWS environment.

It gives suggestion for

1. Cost Optimization
2. Performance
3. Security
4. Fault Tolerance
5. Service Limit



We can find the Trusted Advisor under Management tools



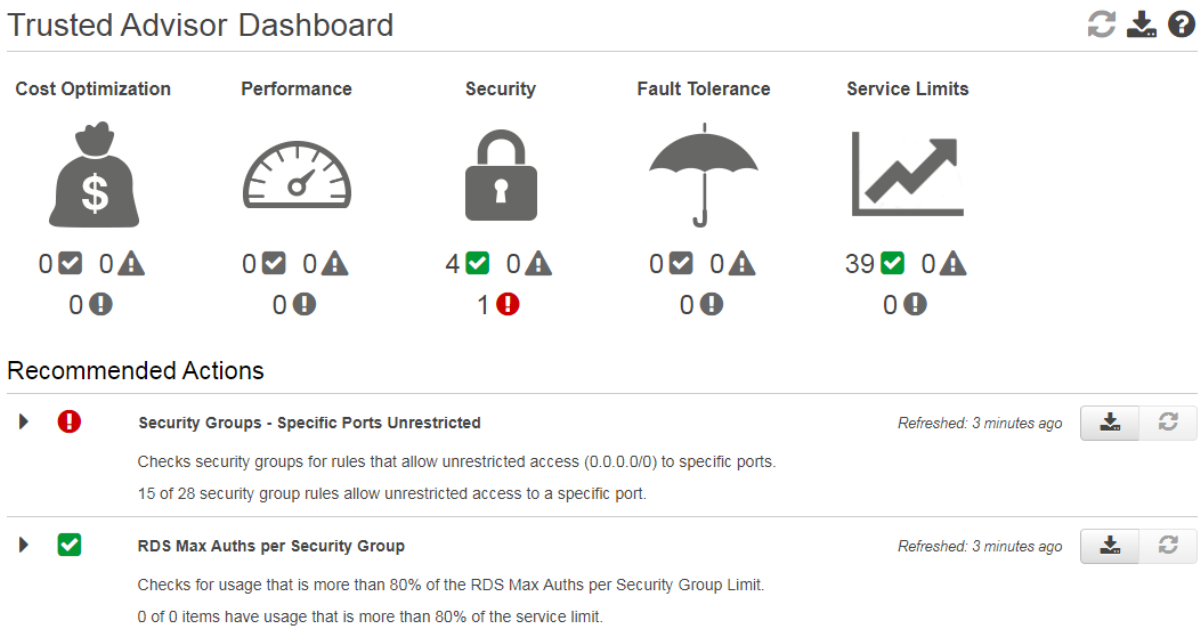
Here is the trusted manager dashboard, it automatically analyzed the AWS environment and given suggestions to improve the listed categories.

The color coding reflects the following information:

***Red*:** Action recommended

Yellow: Investigation recommended

Green: No problem detected



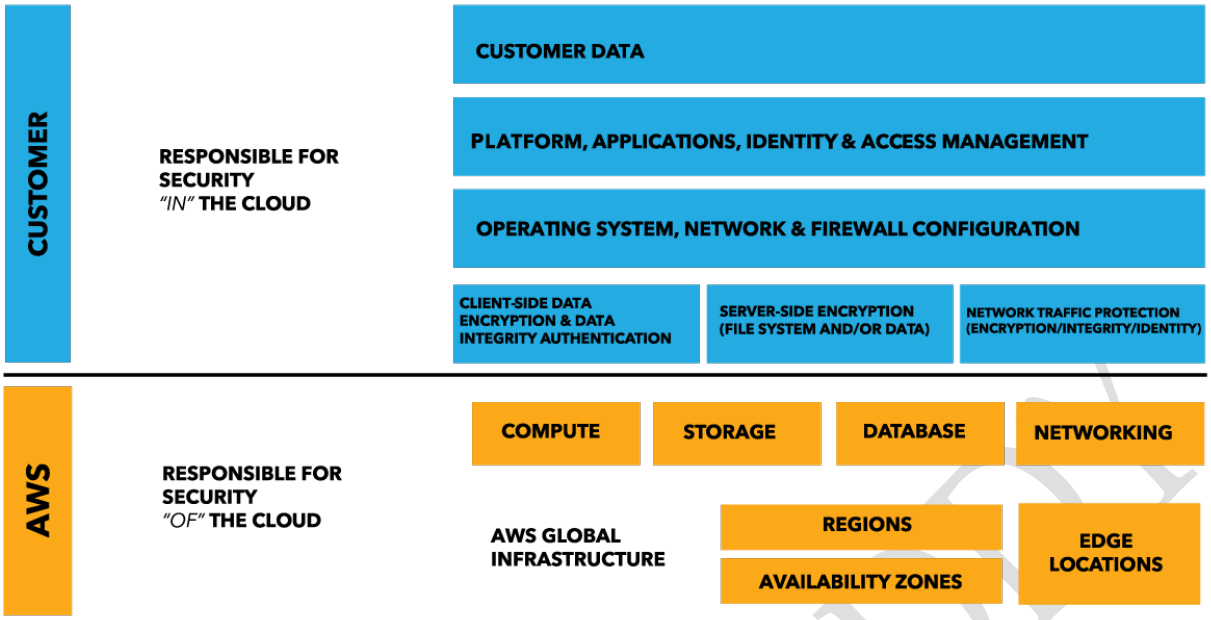
Customers with a Business or Enterprise AWS Support plan can view all AWS Trusted Advisor checks-over 50 checks. We need to upgrade the support plan from Basic to any other to get technical support from Amazon support engineer.

Security:

Security and Compliance is a shared responsibility between AWS and the customer.

AWS responsibility "Security of the Cloud" - AWS is responsible for protecting the infrastructure that runs all of the services offered in the AWS Cloud. This infrastructure is composed of the hardware, software, networking, and facilities that run AWS Cloud services.

Customer responsibility "Security in the Cloud" – Customer responsibility will be determined by the AWS Cloud services that a customer selects. This determines the amount of configuration work the customer must perform as part of their security responsibilities. For example, services such as Amazon Elastic Compute Cloud (Amazon EC2), Amazon Virtual Private Cloud (Amazon VPC), and Amazon S3 are categorized as Infrastructure as a Service (IaaS) and, as such, require the customer to perform all of the necessary security configuration and management tasks. If a customer deploys an Amazon EC2 instance, they are responsible for management of the guest operating system (including updates and security patches), any application software or utilities installed by the customer on the instances, and the configuration of the AWS-provided firewall (called a security group) on each instance.



AWS Well-Architected framework

The AWS Well-Architected framework includes strategies to help you compare your workload against our best practices, and obtain guidance to produce stable and efficient systems so you can focus on functional requirements.

AWS has 5 security pillars for Well Architected framework.

Operational Excellence

The operational excellence pillar focuses on running and monitoring systems to deliver business value, and continually improving processes and procedures. Key topics include managing and automating changes, responding to events, and defining standards to successfully manage daily operations.

Security

The security pillar focuses on protecting information & systems. Key topics include confidentiality and integrity of data, identifying and managing who can do what with privilege management, protecting systems, and establishing controls to detect security events.

Reliability

The reliability pillar focuses on the ability to prevent, and quickly recover from failures to meet business and customer demand. Key topics include foundational elements around setup, cross project requirements, recovery planning, and how we handle change.

Performance Efficiency

The performance efficiency pillar focuses on using IT and computing resources efficiently. Key topics include selecting the right resource types and sizes based on workload requirements, monitoring performance, and making informed decisions to maintain efficiency as business needs evolve.

Cost Optimization

Cost Optimization focuses on avoiding un-needed costs. Key topics include understanding and controlling where money is being spent, selecting the most appropriate and right number of resource types, analyzing spend over time, and scaling to meet business needs without overspending.

QUIZ

1. You currently have an EC2 instance hosting a web application. The number of users is expected to increase in the coming months and hence you need to add more elasticity to your setup. Which of the following methods can help add elasticity to your existing setup. Choose 2 answers from the options given below. Please select:

a) Setup your web app on more EC2 instances and set them behind an Elastic Load balancer
b) Setup an Elastic Cache in front of the EC2 instance.
c) Setup your web app on more EC2 instances and use Route53 to route requests accordingly.
d) Setup DynamoDB behind your EC2 Instances

2. You are creating a Provisioned IOPS volume in AWS. The size of the volume is 8 GiB. Which of the following are the possible values that can put for the IOPS of the volume. Please select:

a) 400
b) 500
c) 600
d) 1000

3. A company is hosting EC2 instances which focuses on work-loads are on non-production and non-priority batch loads. Also these processes can be interrupted at any time. What is the best pricing model which can be used for EC2 instances in this case?

a) Reserved Instances
b) On-Demand Instances
c) Spot Instances
d) Regular Instances

4. You have 2 Ubuntu instances located in different subnets in the same VPC. Now to your understanding these instances should be able to communicate with each other, but when you try to ping from one instance to another, you get a timeout. The Route tables seem to be valid and has the entry for the Target 'local' for your VPC CIDR. Which of the following could be a valid reason for this issue.Please select:

a) The Instances are of the wrong AMI , hence you are not able to ping the instances.
b) The Security Group has not been modified for allow the required traffic.
c) The Instances don't have Public IP, so that the ping commands can be routed
d) The Instances don't have Elastic IP, so that the ping commands can be routed

5. What is the best way to move an EBS volume currently attached to an EC2 instance from one availability zone to another ? Please select:

a) Detach the volume and attach to an EC2 instance in another AZ.
b) Create a new volume in the other AZ and specify the current volume as the source.
c) Create a snapshot of the volume and then create a volume from the snapshot in the other AZ
d) Create a new volume in the AZ and do a disk copy of contents from one volume to another.

6. When it comes to API credentials, what is the best practice recommended by AWS? Please select:

a) Create a role which has the necessary and can be assumed by the EC2 instance.
b) Use the API credentials from an EC2 instance.
c) Use the API credentials from a bastion host.
d) Use the API credentials from a NAT Instance.

7. You wantto retrieve the Public IP addresses assigned to a running instance via the instance metadata. Which of the below urls is valid for retrieving this data.

a) http:// 169.254.169.254/latest/meta-data/public-ipv4
b) http://254.169.254.169/latest/meta-data/public-ipv4
c) http://254.169.254.169/meta-data/latest/public-ipv4
d) http://169.254.169.254/meta-data/latest/public-ipv4

8. You are planning to use the MySQL RDS in AWS. You have a requirement to ensure that you are available to recover from a database crash. Which of the below is not a recommended practise when you want to fulfil this requirement. Please select:

a) Ensure that automated backups are enabled for the RDS
b) Ensure that you use the MyISAM storage engine for MySQL
c) Ensure that the database does not grow too large
d) Ensure that file sizes for the RDS is well under 6 TB.

9. Which of the following is a valid bucket name

a) demo
b) Example
c) .example
d) demo.

10. Which of the following is not a feature provided by Route53?Please select:

a) Registration of Domain Names
b) Routing of internet traffic to domain resources
c) Offloading content to cache locations
d) Health check of resources

11. When working with API gateways in AWS, what is the type of endpoints that exposed

a) HTTP
b) HTTPS
c) JSON
d) XML

12. Which of the following verbs are supported with the API Gateway. Please select :

a) GET
b) POST
c) PUT
d) All of the above

13. Which of the following container technologies are currently supported by the AWS ECS service? Please select:

a) Kubernetes
b) Docker
c) Mesosphere
d) Canonical LXD

14. Which of the following when used alongside with the AWS Secure Token service can be used to provide a single sign-on experience for existing users who are part of an organization using on-premise applications. Please select:

a) OpenID Connect
b) JSON

c) SAML 2.0
d) OAuth

15. While performing status checks on your volume in AWS , you can see that the volume check has a status of "insufficient-data". What can you derive from this status check. Please select:

a) All checks have passed
b) A particular check has failed only
c) All checks have failed
d) The check on the volume is still in progress.

16. Which of the following can constitute the term of a “Golden Image”

a) This is the basic AMI which is available in AWS.
b) This refers to an instance which has been bootstrapped.
c) This refers to an AMI that has been constructed from a customized Image.
d) This refers to a special type of Linux AMI.

17. When designing a health check for your web application which is hosted behind an elastic load balancer, which of the following health checks is ideal to implement
Please select:

a) A TCP health check
b) A UDP health check
c) A HTTP health check
d) A combination of TCP and UDP health checks

18. Which of the following is an example of synchronous replication which occurs in the AWS service. Please select:

a) AWS RDS Read Replica's for MySQL, MariaDB and PostgreSQL
b) AWS Multi-AZ RDS
c) Redis engine for Amazon ElastiCache replication
d) AWS RDS Read Replica's for Oracle

19. You want to get the reason for your EC2 Instance termination from the CLI. Which of the below commands is ideal in getting the reason. Please select:

a) aws ec2 describe-instances
b) aws ec2 describe-images
c) aws ec2 get-console-screenshot
d) aws ec2 describe-volume-status

20. When using the following AWS services, which should be implemented in multiple Availability Zones for high availability solutions? Choose 2 answers from the options below. Please select:

a) Amazon DynamoDB
b) Amazon Elastic Compute Cloud (EC2)
c) Amazon Elastic Load Balancing
d) Amazon Simple Storage Service (S3)

21. An application is currently configured on an EC2 instance to process messages in SQS. The queue has been created with the default settings. The application is configured to just read the messages once a week. It has been noticed that not all the messages are being picked by the application. What could be the issue?

a) The application is configured to long polling, so some messages are not being picked up
b) The application is configured to short polling, so some messages are not being picked up
c) Some of the messages have surpassed the retention period defined for the queue
d) Some of the messages don't have the right permissions to be picked up by the application

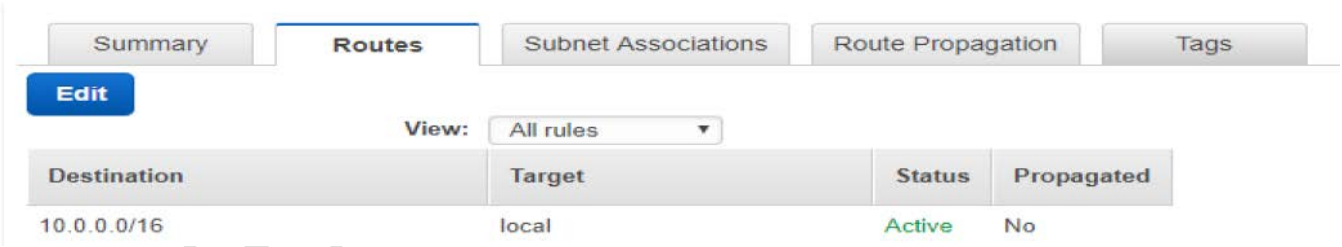
22. Your application is on an EC2 instance in AWS. Users use the application to upload a file to S3. The message first goes to an SQS queue, before it is picked up by a worker process, which fetches the object and uploads it to S3. An email is then sent on successful completion of the upload. You notice though that you are getting numerous emails for each request, when ideally you should be getting only one final email notification for each successful upload. Which of the below could be the possible reasons for this.

a) The application is configured for long polling so the messages are being picked up multiple times.
b) The application is not deleting the messages from SQS.
c) The application is configured to short polling, so some messages are not being picked up
d) The application is not reading the message properly from the SQS queue.

23. You have created your own VPC and subnet in AWS. You have launched an instance in that subnet. You have noticed that the instance is not receiving a DNS name. Which of the below options could be a valid reason for this issue.

a) The CIDR block for the VPC is invalid
b) The CIDR block for the subnet is invalid
c) The VPC configuration needs to be changed.
d) The subnet configuration needs to be changed.

24. You have created your own VPC and subnet in AWS. You have launched an instance in that subnet. You have attached an internet gateway to the VPC and seen that the instance has a public IP. The Route table is shown below



The instance still cannot be reached from the Internet. Which of the below changes need to be made to the route table to ensure that the issue can be resolved.Please select:

a) Add the following entry to the route table – 0.0.0.0/0->Internet Gateway
b) Modify the above route table – 10.0.0.0/16 ->Internet Gateway
c) Add the following entry to the route table – 10.0.0.0/16 ->Internet Gateway
d) Add the following entry to the route table - 0.0.0.0/16->Internet Gateway

25. You wanted to have a VPC created in AWS which will host an application. The application will just consist of web and database servers. The application just requires to be accessed from the internet by internet users. Which of the following VPC configuration wizards options would you use.

a) VPC with a Single Public Subnet Only
b) VPC with Public and Private Subnets
c) VPC with Public and Private Subnets and Hardware VPN Access
d) VPC with a Private Subnet Only and Hardware VPN Access

26. Which of the following statements are true with regards to EBS Volumes. Choose 3 correct answers from the options given below.

a) EBS Volumes are automatically replicated within that zone to prevent data loss due to failure of any single hardware component
b) EBS Volumes can be attached to any EC2 Instance in any AZ.
c) After you attach a volume, it appears as a native block device similar to a hard drive or other physical device.
d) An EBS volume can be attached to only one instance at a time

27. You are a solutions architect working for a large oil and gas company. Your company runs their production environment on AWS and has a custom VPC. The VPC contains 3 subnets, 1 of which is public and the other 2 are private. Inside the public subnet is a fleet of EC2 instances which are the result of an autoscaling group. All EC2 instances are in the same security group. Your company has created a new custom application which connects to mobile devices using a custom port. This application has been rolled out to production and you need to open this port globally to the internet. What steps should you take to do this, and how quickly will the change occur?

a) Open the port on the existing network Access Control List. Your EC2 instances will be able to communicate on this port after a reboot.
b) Open the port on the existing network Access Control List. Your EC2 instances will be able to communicate over this port immediately.
c) Open the port on the existing security group. Your EC2 instances will be able to communicate over this port immediately.
d) Open the port on the existing security group. Your EC2 instances will be able to communicate over this port as soon as the relevant Time To Live (TTL) expires.

28. You are designing various CloudFormation templates, each template to be used for a different purpose. What determines the cost of using the CloudFormation templates?

a) A. CloudFormation does not have a cost itself.
b) B. You are charged based on the size of the template.
c) C. You are charged based on the time it takes to launch the template.
d) D. It has a basic charge of $1.10

29. You are creating a number of EBS Volumes for your EC2 instances. You are concerned on the backups of the EBS Volumes. Which of the below is a way to backup the EBS Volumes

a) Configure Amazon Storage Gateway with EBS volumes as the data source and store the backups on premise through the storage gateway
b) Write a cronjob that uses the AWS CLI to take a snapshot of production EBS volumes.
c) Use a lifecycle policy to back up EBS volumes stored on Amazon S3 for durability
d) Write a cronjob on the server that compresses the data and then copy it to Glacier

30. You are planning on hosting a static website on an EC2 instance. Which of the below aspects can be used to create a highly available environment. Choose three answers from given below.

a) A. An auto scaling group to recover from EC2 instance failures
b) B. Elastic Load balancer
c) C. An SQS queue
d) D. Multiple Availability Zones

31. You have a set of IIS Servers running on EC2 Instances. You want to collect and process the log files generated from the IIS Servers. Which of the below services is ideal to run in this scenario

a) Amazon S3 for storing the log files and Amazon EMR for processing the log files
b) Amazon S3 for storing the log files and EC2 Instances for processing the log files
c) Amazon EC2 for storing and processing the log files
d) Amazon DynamoDB to store the logs and EC2 for running custom log analysis scripts

32. You are trying to configure Cross Region Replication for your S3 bucket. But you are not able to select the option of Cross Region Replication and is disabled.
Which of the below could be the possible reasons for this?

a) The feature is not available in that region
b) You need to enable versioning on the bucket
c) The source region is currently down
d) The destination region is currently down

33. What is the amount of temp space is allocated to you when using Lambda functions per invocation.

a) 256 MB
b) 512 MB
c) 2 GiB
d) 16 GiB

34. You have a requirement to create a subnet in an AWS VPC which will host around 20 hosts. This subnet will be used to host web servers. Which ofthe below could be the possible CIDR block allocated for the subnet Please select:

a) 10.0.1.0/27
b) 10.0.1.0/28
c) 10.0.1.0/29
d) 10.0.1.0/30

35. You run a website which hosts videos and you have two types of members, premium fee paying members and free members. All videos uploaded by both your premium members and free members are processed by a fleet of EC2 instances which will poll SQS as videos are uploaded. However you need to ensure that your premium fee paying members videos have a higher priority than your free members. How do you design SQS?

a) SQS allows you to set priorities on individual items within the queue, so simply set the fee paying members at a higher priority than your free members.
b) Create two SQS queues, one for premium members and one for free members. Program your EC2 fleet to poll the premium queue first and if empty, to then poll your free members SQS queue.
c) SQS would not be suitable for this scenario. It would be much better to use SNS to encode the videos.
d) Use SNS to notify when a premium member has uploaded a video and then process that video accordingly.

36. Which of the following services natively encrypts data at rest within an AWS region? (Choose two.)

a) AWS Storage Gateway
b) Amazon DynamoDB
c) Amazon CloudFront
d) Amazon Glacier
e) Amazon Simple Queue Service

37. Your EC2 instances are configured to run behind an Amazon VPC. You have assigned two web servers instances to an Elastic Load Balancer. However, the instances and the ELB are not reachable via URL to the elastic load balancer serving the web app data from the EC2 instances. What could be done to resolve this issue.

a) Attach an Internet gateway to the VPC and route it to the subnet
b) Add an elastic IP address to the instance
c) Use Amazon Elastic Load Balancer to serve requests to your instances located in the internal subnet
d) Recreate the instances again

38. You want to ensure that you keep a check on the Active Volumes, Active snapshots and Elastic IP addresses you use so that you don't go beyond the service limit. Which of the below services can help in this regard?

a) AWS Cloudwatch
b) AWS EC2
c) AWS Trusted Advisor
d) AWS SNS

39. You are building an automated transcription service in which Amazon EC2 worker instances process an uploaded audio file and generate a text file. You must store both of these files in the same durable storage until the text file is retrieved. You do not know what the storage capacity requirements are. Which storage option is both cost-efficient and scalable?

a) Multiple Amazon EBS volume with snapshots
b) A single Amazon Glacier vault
c) A single Amazon S3 bucket
d) Multiple instance stores

40. You are an AWS Administrator for your company. The company currently has a set of AWS resources hosted in a particular region. You have been requested by your supervisor to create a script which could create duplicate resources in another region incase of a disaster. Which of the below AWS services could help fulfil this requirement.

a) AWS Elastic Beanstalk
b) AWS SQS
c) AWS Cloudformation
d) AWS SNS

41. What are bastion hosts? Please select:

a) They are instances in the public subnet which are used as a jump server to resources within other subnets.
b) They are instances in the private subnet which are used as a jump server to resources within other subnets.
c) They are instances in the public subnet which are used to host web resources that can be accessed by users.
d) They are instances in the private subnet which are used to host web resources that can be accessed by users.

42. You have several AWS reserved instances in your account. They have been running for some time, but now need to be shutdown since they are no longer required. The data is still required for future purposes. Which of the below possible 2 steps can be taken.

a) Convert the instance to on-demand instances
b) Sell the instances on the AWS Reserved Instance Marketplace
c) Take snapshots of the EBS volumes and terminate the instances

d) Convert the instance to spot instances

43. You have an EC2 Instance in a particular region. This EC2 Instance has a preconfigured software running on it. You have been requested to create a disaster recovery solution incase the instance in the region fails. Which of the following is the best solution.

a) Create a duplicate EC2 Instance in another AZ. Keep it in the shutdown state. When required , bring it back up.
b) Backup the EBS data volume. If the instance fails , bring up a new EC2 instance and attach the volume.
c) Store the EC2 data on S3. If the instance fails , bring up a new EC2 instance and restore the data from S3.
d) Create an AMI of the EC2 Instance and copy it to another region

44. You have an EC2 instance located in a subnet in AWS. You have installed a web application on this instance. The security group attached to this instance is shown below

The VPC has the following Route table attached to it

You can SSH into the instance from the internet, but you are not able to access the web server via the web browser. Which of the below steps would resolve the issue?

a) A. Add an HTTP rule to the Security Group
b) B. Remove the SSH rule from the security group
c) C. Add the route 10.0.0.0/16 -> igw-a97272cc to the Route Table
d) D. Add the route 0.0.0.0/0 -> local to the Route Table

45. Amazon's Redshift uses which block size for its columnar storage

a) 2KB
b) 8KB
c) 16KB
d) 32KB
e) 1024KB

46. You working in the media industry and you have created a web application where users will be able to upload photos they create to your website. This web application must be able to call the S3 API in order to be able to function. Where should you store your API credentials whilst maintaining the maximum level of security.

a) Save the API credentials to your php files.
b) Don't save your API credentials. Instead create a role in IAM and assign this role to an EC2 instance when you first create it.
c) Save your API credentials in a public Github repository.
d) Pass API credentials to the instance using instance userdata.

47. You are a systems administrator and you need to monitor the health of your production environment. You decide to do this using Cloud Watch, however you notice that you cannot see the health of every important metric in the default dash board. Which of the following metrics do you need to design a custom cloud watch metric for, when monitoring the health of your EC2 instances?

a) CPU Usage
b) Memory usage
c) Disk read operations
d) Network in

48. In order for an EC2 instance to be accessed from the internet, which of the following are required. Choose 3 answers from the options given below.

a) An Internet gateway attached to the VPC
b) A private IP address attached to the instance
c) A public IP address attached to the instance
d) A route entry to the Internet gateway in the Route table

49. You are IOT sensors to monitor the number of bags that are handled at an airport. The data gets sent back to a Kinesis stream with default settings. Every alternate day, the data from the stream is sent to S3 for processing. But you notice that S3 is not receiving all of the data that is being sent to the Kinesis stream. What could be the reason for this.

a) The sensors probably stopped working on some days hence data is not sent to the stream.
b) S3 can only store data for a day
c) Data records are only accessible for a default of 24 hours from the time they are added to a stream
d) Kinesis streams are not meant to handle IoT related data

50. A customer wants to track access to their Amazon Simple Storage Service (S3) buckets and also use this information for their internal security and access audits. Which of the following will meet the Customer requirement?

a) A. Enable AWS CloudTrail to audit all Amazon S3 bucket access
b) B. Enable server access logging for all required Amazon S3 buckets
c) C. Enable the Requester Pays option to track access via AWS Billing
d) D. Enable Amazon S3 event notifications for Put and Post.

51. You are defined the following Network ACL for your subnet.

Rule #	Type	Protocol	Port Range	Source	Allow / Deny	Remove
100	ALL Traffic	ALL	ALL	0.0.0.0/0	ALLOW	
101	Custom TCP Rule	TCP (6)	1024	20.202.20.252/32	DENY	

What will be the outcome when a workstation of IP 20.202.20.252 tries to access your subnet.
The request will be allowed

a) The request will be denied
b) The request will be allowed initially and then denied
c) The request will be denied initially and then allowed

52. Which procedure for backing up relational database on EC2 that is using a set of RAIDed EBS volumes for storage minimizes the time during which the database cannot be written to and results in a consistent backup?

a) 1. Detach EBS volume, 2. Start EBS snapshot of volumes, 3. Re-attach EBS volumes
b) 1. Stop the EC2 Instance. 2. Snapshot the EBS volumes
c) 1. Suspend disk I/O, 2. Create an image of the EC2 Instance, 3. Resume disk I/O
d) 1. Suspend disk I/O. 2. Start EBS snapshot of volumes, 3. Resume disk I/O
e) 1. Suspend disk I/O, 2. Start EBS snapshot of Volumes, 3. Wait for snapshot to complete, 4. Resume disk

53. You are a solutions architect working for a company. They store their data on S3, however recently an someone accidentally deleted some critical files in S3. You've been asked to prevent this from happening in the future. What options below can prevent this?

a) Make sure you provide signed URL's to all users.
b) Enable S3 versioning and Multifactor Authentication (MFA) on the bucket.
c) Use S3 Infrequently Accessed storage to store the data on.
d) Create an IAM bucket policy that disables deletes.

54. You run an automobile reselling company that has a popular online store on AWS. The application site behind an Auto Scaling group and required new instances of Auto scaling group to identify their puclic and private Ip addresses. How can you achieve this?

a) By using Ipconfig for windows or ifconfig for linux
b) By using a cloud watch metric
c) using a curl or Get command to get the latest meta-data from http://169.254.169.254/latest/meta-data/
d) using a curl or Get command to get the latest meta-data from http://169.254.169.254/latest/user-data/

55. You are the solution architect for a company. The company has a requirement to deploy an application which will need to have session management in place. Which of the following services can be used for session management accordingly?

a) AWS Storage Gateway, Elasticache & ELB
b) ELB, Elasticache & RDS
c) Cloudwatch, RDS & DynamoDb
d) RDS, DynamoDB & Elasticache.

56. You are working for an Enterprise and have been asked to get a support plan in place from AWS.

1) 24x7 access to support
2) Access to the full set of Trusted Advisor checks

Which of the following would meet these requirements ensuring that cost is kept at a minimum

a) Basic
b) Developer
c) Business
d) Enterprise

57. Which of the following is incorrect with regards to Private IP addresses?

a) In Amazon EC2 classic, the private IP addresses are only returned to Amazon EC2 when the instance is stopped or terminated
b) In Amazon VPC, an instance retains its private IP addresses when the instance is stopped.
c) In Amazon VPC, an instance does NOT retain its private IP addresses when the instance is stopped.
d) In Amazon EC2 classic, the private IP address is associated exclusively with the instance for its lifetime

58. Which of the following are best practices for monitoring your EC2 Instances

a) Create and implement a monitoring plan that collects monitoring data from all of the parts in your AWS solution
b) Automate monitoring tasks as much as possible
c) Check the log files on your EC2 instances
d) All of the above

59. For which of the following use cases are Simple Workflow Service (SWF) and Amazon EC2 an appropriate solution? Choose two answers from the options given below

a) using as an endpoint to collect thousands of data points per hour from a distributed fleet of sensors
b) managing a multi-step and multi-decision checkout process of an e-commerce website.
c) Orchestrating the execution of distributed and auditable business process.
d) Using as an SNS endpoint to trigger execution of video transcoding jobs

60. You work for a major news network in Europe. They have just released a new app which allows users to report on events as and when they happen using their mobile phone. Users are able to upload pictures from the app and then other users will be able to view these pics. Your organization expects this app to grow very quickly, essentially doubling it's user base every month. The app uses S3 to store the media and you are expecting sudden and large increases in traffic to S3 when a major news event takes place as people will be uploading content in huge numbers). You need to keep your storage costs to a minimum however and it does not matter if some objects are lost. Which storage media should you use to keep costs as low as possible?

a) S3 – Infrequently Accessed Storage.
b) S3 – Reduced Redundancy Storage (RRS).
c) Glacier.
d) S3 – Provisioned IOPS.

Answers:

1. A and C
2. A
3. C
4. B
5. C
6. A
7. A
8. B
9. A
10. C
11. B
12. D
13. A and B
14. C
15. D
16. C
17. C
18. B
19. A
20. B and C
21. C
22. B
23. C
24. A
25. B
26. A, C and D
27. C
28. A
29. B
30. A, B and D
31. A
32. B
33. B
34. A
35. B
36. C and E
37. A
38. C
39. C
40. C
41. A
42. B and C
43. D
44. A
45. E
46. B
47. B
48. A, C and D
49. C
50. B
51. A

52. E
53. B
54. C
55. D
56. C
57. C
58. D
59. B and C
60. B