

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.

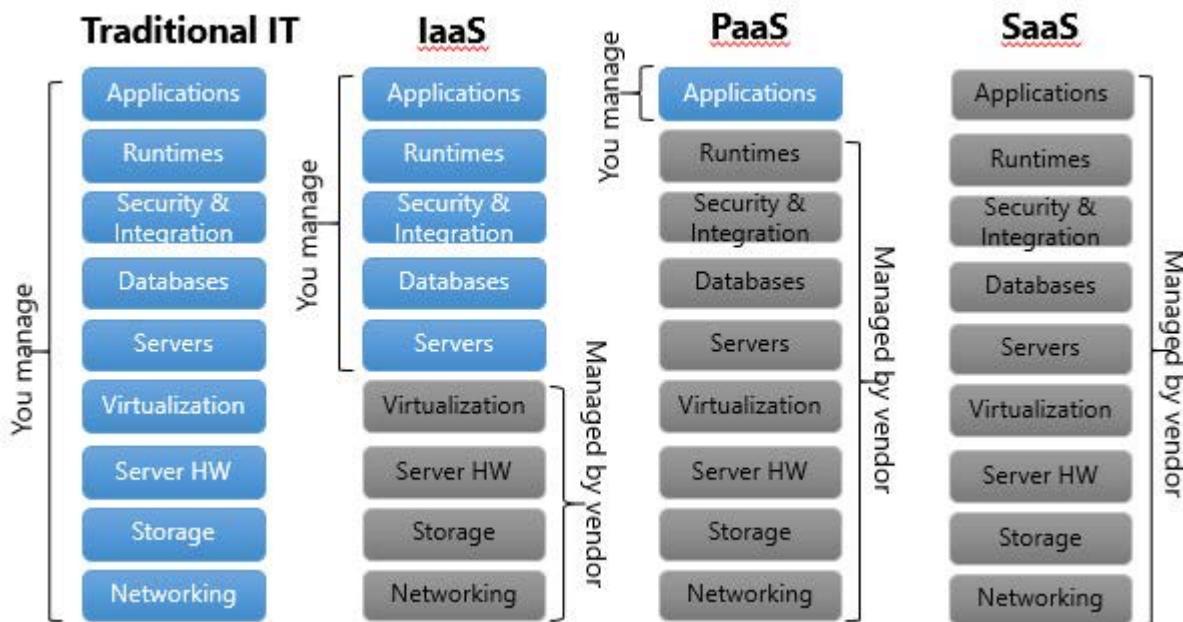
3. Infrastructure as a Service (IaaS)

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.

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 80 Availability Zones within 25 geographic Regions around the world. (Till July, 2021).

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 and

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 contain 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.

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:

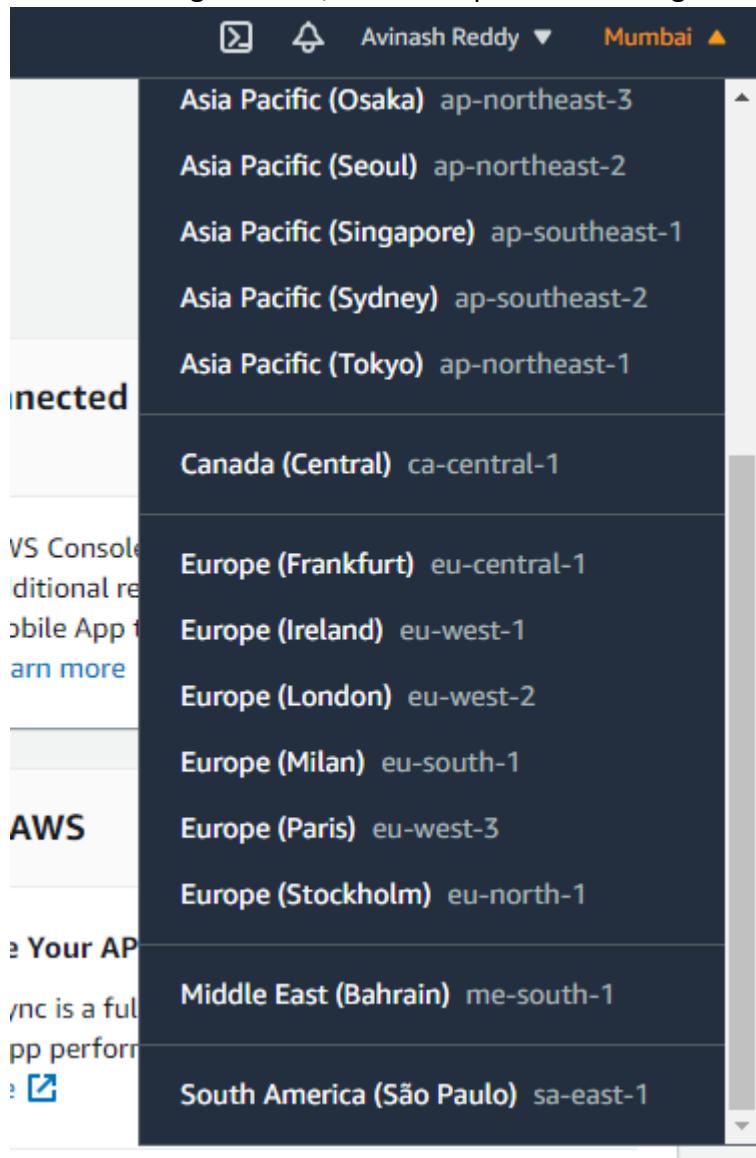
Region Name	Region Code
US East (N. Virginia)	us-east-1
US East (Ohio)	us-east-2
US West (N. California)	us-west-1
US West (Oregon)	us-west-2
Africa (Cape Town)	af-south-1
Asia Pacific (Hong Kong)	ap-east-1
Asia Pacific (Mumbai)	ap-south-1
Asia Pacific (Osaka)	ap-northeast-3
Asia Pacific (Seoul)	ap-northeast-2
Asia Pacific (Singapore)	ap-southeast-1
Asia Pacific (Sydney)	ap-southeast-2
Asia Pacific (Tokyo)	ap-northeast-1
Canada (Central)	ca-central-1
Europe (Frankfurt)	eu-central-1
Europe (Ireland)	eu-west-1
Europe (London)	eu-west-2
Europe (Milan)	eu-south-1
Europe (Paris)	eu-west-3
Europe (Stockholm)	eu-north-1
Middle East (Bahrain)	me-south-1
South America (São Paulo)	sa-east-1
China (Beijing)	cn-north-1
China (Ningxia)	cn-northwest-1
AWS GovCloud (US-West)	us-gov-west-1
AWS GovCloud (US-East)	us-gov-east-1

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



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.

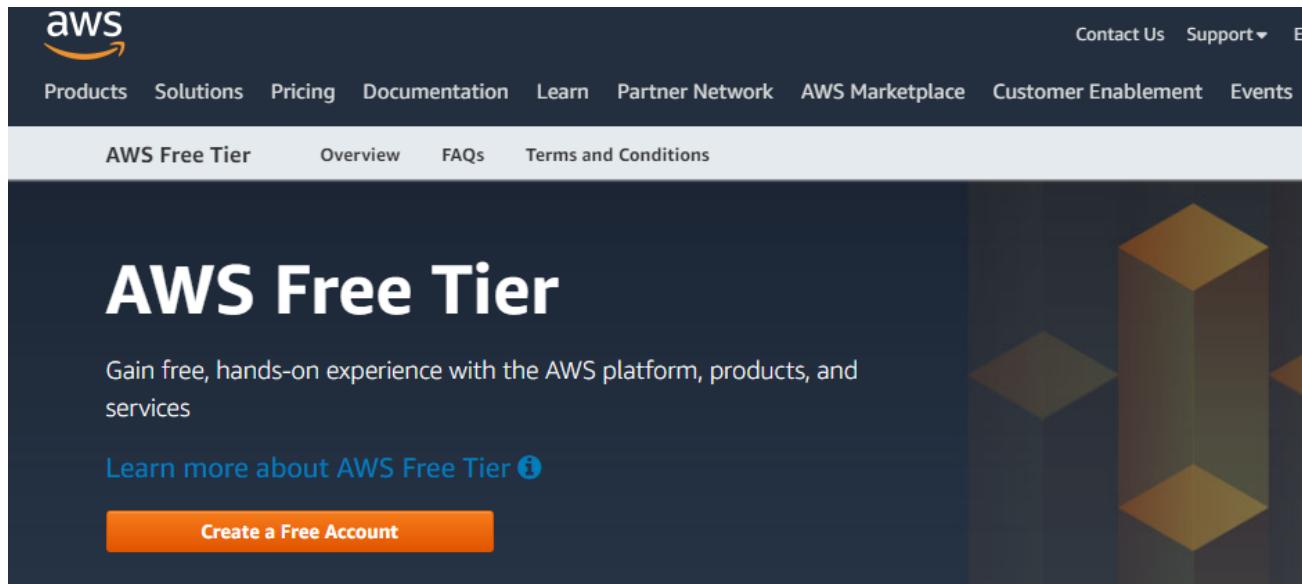


3. You can switch between the regions and some services are region specific and some are global.

AWS ACCOUNT CREATION

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.

Explore Free Tier products with a new AWS account.

To learn more, visit aws.amazon.com/free.



Sign up for AWS

Email address

You will use this email address to sign in to your new AWS account.

Password

Confirm password

AWS account name

Choose a name for your account. You can change this name in your account settings after you sign up.

Continue (step 1 of 5)

[Sign in to an existing 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 “Business 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.

Sign up for AWS

Contact Information

How do you plan to use AWS?

Business - for your work, school, or organization
 Personal - for your own projects

Who should we contact about this account?

Full Name
[Text Input Field]

Organization name
[Text Input Field]

Phone Number
Enter your country code and your phone number.
+1 222-333-4444

Address
Kukatpally
Hyderabad

City
Hyderabad

State, Province, or Region
Telangana

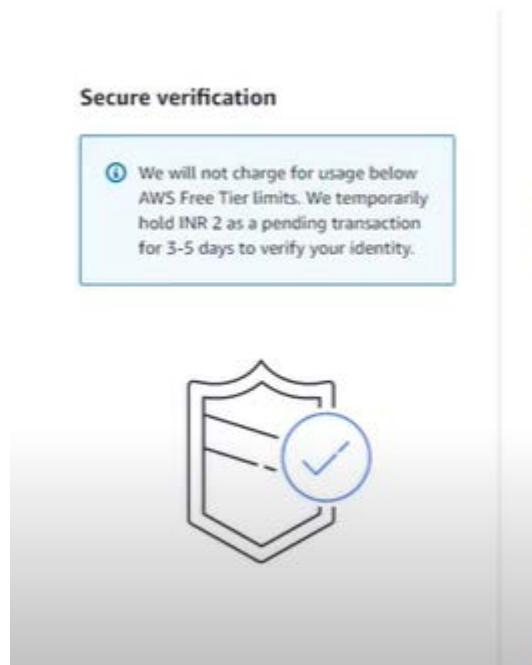
Postal Code
500072

Customers with an Indian contract address are served by Amazon Internet Services Private Ltd. (AISPL). AISPL is the local seller for AWS services in India.

I have read and agree to the terms of the AWS Customer Agreement [\[Link\]](#).

Continue (step 2 of 5)

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



Secure verification

We will not charge for usage below AWS Free Tier limits. We temporarily hold INR 2 as a pending transaction for 3-5 days to verify your identity.

Billing Information

Credit or Debit card number

VISA MASTERCARD AMEX DISCOVER

AWS accepts all major credit and debit cards. To learn more about payment options, review our FAQ

Expiration date

Cardholder's name

CVV

Billing address

Use my contact address

Complete Payment of Rs 2.00 Help

Merchant Details

Merchant: AMAZON INTERNET SERVICES

Amount: Rs 2.00

Date: 08:06:2021

Card Number: XXXX XXXX XXXX 0089

Enter One Time Password (OTP)

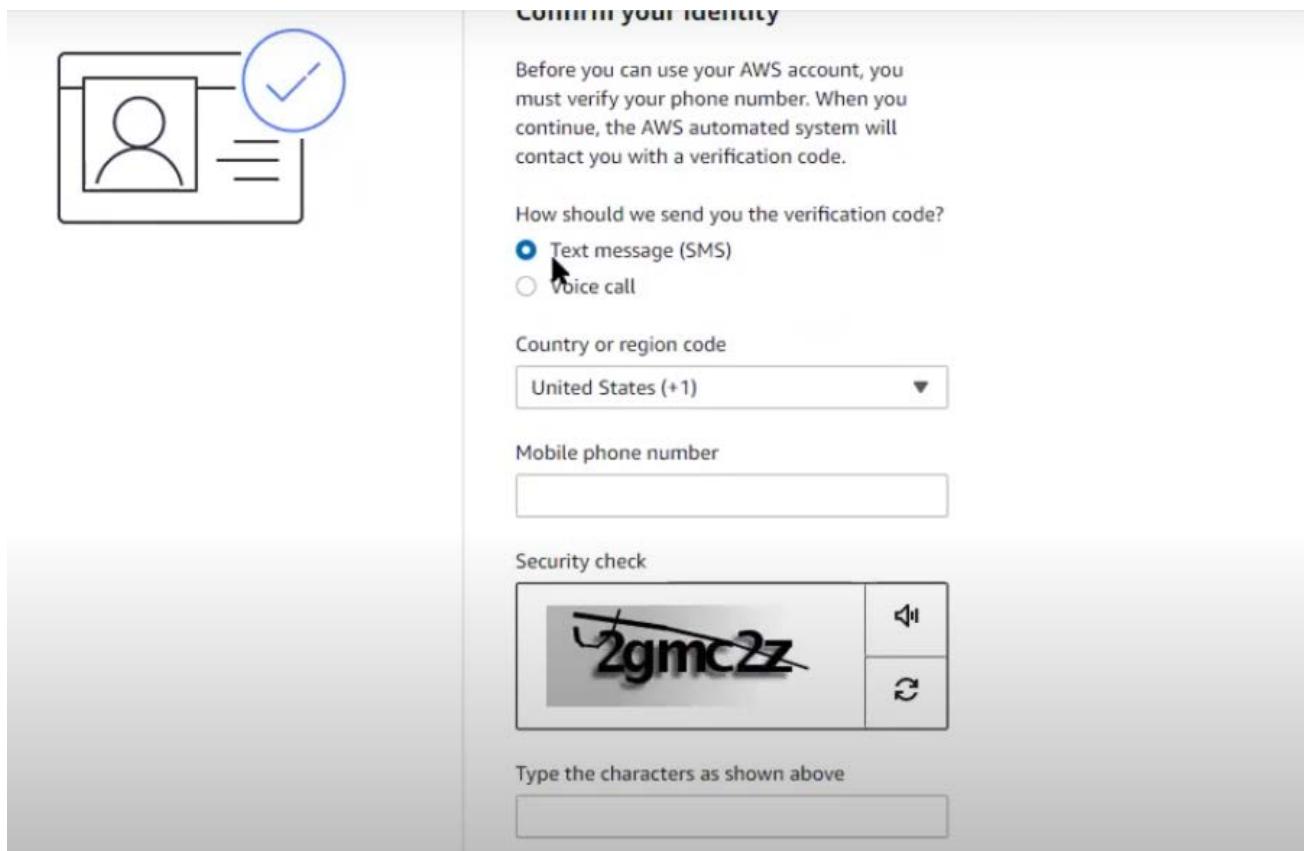
Enter One Time Password (OTP) for secured online transaction.

One Time Password (OTP) has been sent to card holder's registered Mobile Number ending with 3007 and e-mail ID avXXXX@gmail.com.

Submit

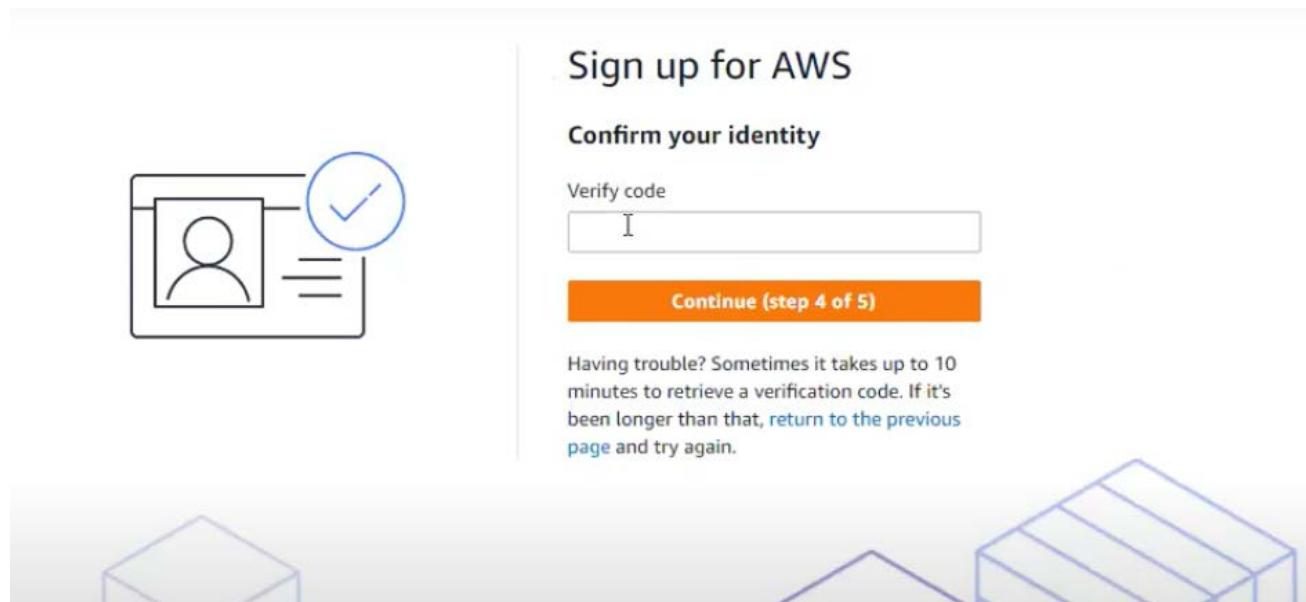
Resend OTP

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. Or we can confirm the identity by Receiving a OTP from AWS, Choose the “Text message (SMS)” option, choose Country code, Enter a Valid phone number and displayed captcha.



The screenshot shows the 'Confirm your identity' step of the AWS account creation process. It includes:

- A user icon with a checkmark.
- A message: "Before you can use your AWS account, you must verify your phone number. When you continue, the AWS automated system will contact you with a verification code."
- A question: "How should we send you the verification code?" with options: Text message (SMS) and Voice call.
- A dropdown for "Country or region code" set to "United States (+1)".
- A field for "Mobile phone number".
- A "Security check" section showing the CAPTCHA text "2gmc2z" with a speaker icon and a refresh icon.
- A text input field for "Type the characters as shown above".



The screenshot shows the "Sign up for AWS" step 4 of 5. It includes:

- A user icon with a checkmark.
- A message: "Sign up for AWS".
- A "Confirm your identity" section with a "Verify code" field containing the letter "I".
- An orange "Continue (step 4 of 5)" button.
- A note: "Having trouble? Sometimes it takes up to 10 minutes to retrieve a verification code. If it's been longer than that, [return to the previous page](#) and try again.".
- Decorative 3D cubes at the bottom.

- After completing the Identity verification, we have to select the "Support Plan" and click on "Continue".

Amazon have 4 support plans, those are

- Basic:** No Monthly Pricing for Basic support plan and no option to get technical support from Amazon if you are facing any.
- 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 three 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.

Select a support plan

Choose a support plan for your business or personal account. [Compare plans and pricing examples](#)
 You can change your plan anytime in the AWS Management Console.

Basic support - Free

- Recommended for new users just getting started with AWS
- 24x7 self-service access to AWS resources
- For account and billing issues only
- Access to Personal Health Dashboard & Trusted Advisor



Developer support - From \$29/month

- Recommended for developers experimenting with AWS
- Email access to AWS Support during business hours
- 12 (business)-hour response times



Business support - From \$100/month

- Recommended for running production workloads on AWS
- 24x7 tech support via email, phone, and chat
- 1-hour response times
- Full set of Trusted Advisor best-practice recommendations



 Need Enterprise level support?

From \$15,000 a month you will receive 15-minute response times and concierge-style experience with an assigned Technical Account Manager. [Learn more](#) 

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



Congratulations

Thank you for signing up for AWS.

We are activating your account, which should only take a few minutes. You will receive an email when this is complete.

[Go to the AWS Management Console](#)

[Sign up for another account or contact sales.](#)

10. 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:

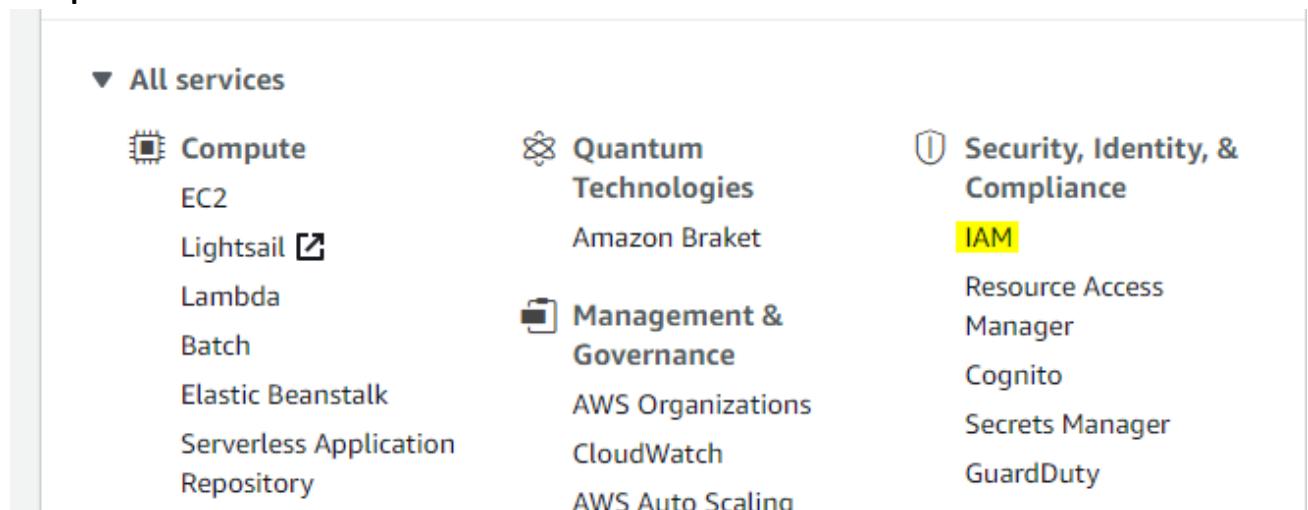
IAM stands for Identity and Access Management (IAM). IAM is a web service that helps you securely control access to AWS resources for your users. We can use IAM to control who can use our AWS resources and how they can use resources.

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.

Welcome to Identity and Access Management

IAM users sign-in link:

<https://518084852000.signin.aws.amazon.com/console>

[Customize](#) | [Copy Link](#)

- 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.

Add user

1 2 3 4

Set user details

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

User name* Avinash_T

[+ Add another user](#)

Select AWS access type

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

Access type* **Programmatic access**

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

AWS Management Console access

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

- 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.

▼ Set permissions



Add user to an existing group or create a new one. Using groups is a best-practice way to manage user's permissions by job functions. [Learn more](#)

Add user to group

The screenshot shows a user interface for managing groups. At the top, there is a search bar labeled 'Search' and a dropdown menu labeled 'Group'. Below this is a table with two rows. The first row shows 'AdminGroup' with the 'Attached policies' column containing 'AdministratorAccess'. The second row shows 's3admins' with the 'Attached policies' column containing 'AmazonS3FullAccess'. There are checkboxes next to each group name.

Group	Attached policies
<input type="checkbox"/> AdminGroup	AdministratorAccess
<input type="checkbox"/> s3admins	AmazonS3FullAccess

To Create group, use the “**Create Group**” option and choose the required policy. Permissions managed at group level, not at user level.

Create group

Create a group and select the policies to be attached to the group. Using groups is a best-practice way to manage users' permissions by job functions, AWS service access, or your custom permissions.

[Learn more](#)

Group name:

[Create policy](#) [Refresh](#)

Filter: Policy type [Policy type](#) Showing 277 results

	Policy name	Type	Attachments	Description
<input type="checkbox"/>	AdministratorAccess	Job function	2	Provides full access to AWS services and resources.
<input type="checkbox"/>	AmazonAPIGatewayAdministrator	AWS managed	0	Provides full access to create/edit/delete APIs in Amazon API Gateway via the AWS Man...
<input type="checkbox"/>	AmazonAPIGatewayInvokeFullAccess	AWS managed	0	Provides full access to invoke APIs in Amazon API Gateway.
<input type="checkbox"/>	AmazonAPIGatewayPushToCloudWatchLogs	AWS managed	0	Allows API Gateway to push logs to user's account

[Cancel](#) [Create group](#)

6. Add required tags. Tag allow us to add metadata to the AWS resources.

Add user

1 2 3 4 5

Add tags (optional)

IAM tags are key-value pairs you can add to your user. Tags can include user information, such as an email address, or can be descriptive, such as a job title. You can use the tags to organize, track, or control access for this user. [Learn more](#)

Key	Value (optional)	Remove
Project	project X	x
Instructor	Avinash T	x
Add new key		

You can add 48 more tags.

7. 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.

User details

User name	Avinash_T
AWS access type	AWS Management Console access - with a password
Console password type	Custom
Require password reset	No
Permissions boundary	Permissions boundary is not set

Permissions summary

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

Type	Name
Managed policy	AdministratorAccess

[Cancel](#) [Previous](#) [Create user](#)

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

Success

You successfully created the users shown below. You can view and download user security credentials. You can also email users instructions for signing in to the AWS Management Console. This is the last time these credentials will be available to download. However, you can create new credentials at any time.

Users with AWS Management Console access can sign-in at: <https://avizway1.signin.aws.amazon.com/console>

[Download .csv](#)

	User	Email login instructions
	Avinash_T	Send email

- 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.

Dashboard

Access management

- User groups
- Users
- Roles
- Policies
- Identity providers
- Account settings

<https://avizway1.signin.aws.amazon.co>

IAM resources

Users: 1
User groups: 2
Customer managed policies: 56

Security alerts

The root user for this account does not have MFA enabled to improve security for this account.

Set password policy

A password policy is a set of rules that define complexity requirements and mandatory rotation periods for your IAM users' passwords. [Learn more](#)

Select your account password policy requirements:

- Enforce minimum password length
- characters
- Require at least one uppercase letter from Latin alphabet (A-Z)
- Require at least one lowercase letter from Latin alphabet (a-z)
- Require at least one number
- Require at least one non-alphanumeric character (! @ # \$ % ^ & * () _ + - = [] { } | ')
- Enable password expiration
- Password expiration requires administrator reset
- Allow users to change their own password
- Prevent password reuse

Create a custom policy for allocating custom permissions for an IAM user.

1. Navigate to Policy option on IAM, and “**Create Policy**” Then choose visual editor option for easy policy creation. We can even create in JSON format.

Create policy

1 2

A policy defines the AWS permissions that you can assign to a user, group, or role. You can create and edit a policy in the visual editor and using JSON. [Learn more](#)

Visual editor **JSON** [Import managed policy](#)

[Expand all](#) | [Collapse all](#)

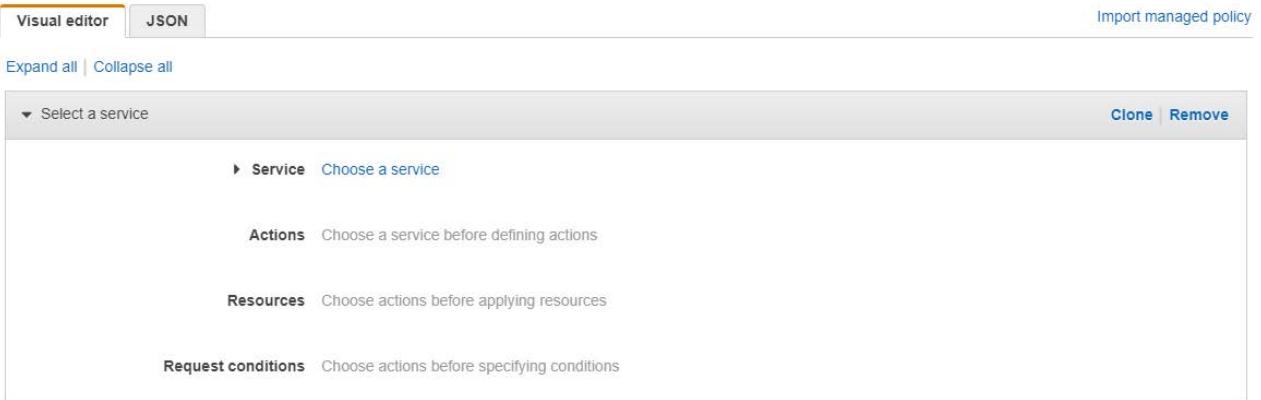
▼ Select a service [Clone](#) | [Remove](#)

► **Service** Choose a service

Actions Choose a service before defining actions

Resources Choose actions before applying resources

Request conditions Choose actions before specifying conditions



2. Now choose the Service, Actions on the required Resources.
3. In this example, am allocating “Describe Instances” and “DescribeInstanceStatus” actions. IAM user who allocated with this policy just can Get information about the Instances, but he cannot perform any operation on ec2 instances.

▼ **EC2 (2 actions)** [Clone](#) | [Remove](#)

► **Service** EC2

► **Actions** List

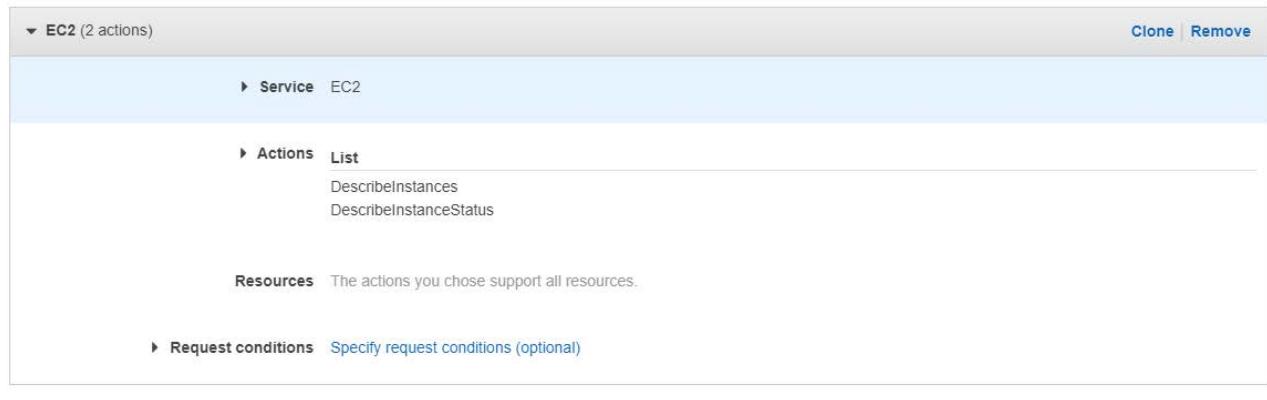
DescribeInstances
DescribeInstanceStatus

Resources The actions you chose support all resources.

► **Request conditions** Specify request conditions (optional)

[Add additional permissions](#)

[Cancel](#) **Review policy**



4. Review the Policy and Create with valid name and Description.

Review policy

Name* Use alphanumeric and '+-, @-_ ' characters. Maximum 128 characters.

Description Maximum 1000 characters. Use alphanumeric and '+-, @-_ ' characters.

Summary

Service	Access level	Resource	Request condition
Allow (1 of 172 services) Show remaining 171	Limited: List	All resources	None
EC2			

* Required [Cancel](#) [Previous](#) [Create policy](#)

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.

EXERCISE 5**Create a custom policy and associate with an IAM user.**

In this exercise, you will create an IAM policy and associate with an IAM user to verify the custom permissions.

1. Create an IAM User.
2. Associate “ec2describe policy” to newly created IAM user.
3. Login as an IAM user and Navigate to ec2 dashboard and check the permissions.

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 0/1 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;
<https://s3-region.amazonaws.com/uniquebucketName/objectname>
- You can access using this URL also, We call it as Virtual Path (It won't work if we have . in bucket name)
<http://mybucket.s3.amazonaws.com/document.doc>
<http://bucket.s3-awsregion.amazonaws.com>
- 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.

Amazon S3 Intelligent-Tiering (S3 Intelligent-Tiering) - The S3 Intelligent-Tiering storage class is designed to optimize costs by automatically moving data to the most cost-effective access tier, without performance impact or operational overhead. It works by storing objects in two access tiers: one tier that is optimized for frequent access and another lower-cost tier that is optimized for infrequent access.

- Designed for durability of 99.999999999% of objects across multiple Availability Zones
- Designed for 99.9% availability over a given year
- Automatically moves objects between two access tiers based on changing access patterns
- Small monthly monitoring and auto-tiering fee

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 across multiple Availability Zones
- 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

S3 One Zone-Infrequent Access - S3 One Zone-Infrequent Access (S3 One Zone-IA; Z-IA) is a new storage class designed for customers who want a lower-cost option for infrequently accessed data, but do not require the multiple Availability Zone data resilience model of the S3 Standard and S3 Standard-Infrequent Access (S3 Standard-IA; S-IA) storage classes. S3 One Zone-IA is intended for use cases with infrequently accessed data that is re-creatable, such as storing secondary backup copies of on-premises data or for storage that is already replicated in another AWS Region for compliance or disaster recovery purposes. With S3 One Zone-IA, customers can now store infrequently accessed data within a single Availability Zone at 20% lower cost than S3 Standard-IA.

- Same low latency and high throughput performance of S3 Standard and S3 Standard-IA.
- Designed for durability of 99.999999999% of objects in a single Availability Zone, but data will be lost in the event of Availability Zone destruction.
- Designed for 99.5% availability over a given year.

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, and you 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.

Glacier Deep Archive : The new Glacier Deep Archive storage class is designed to provide durable and secure long-term storage for large amounts of data at a price that is competitive with off-premises tape archival services. Data is stored across 3 or more AWS Availability Zones and can be retrieved in 12 hours or less.

	Storage class	Designed for	Availability Zones	Min storage duration
<input checked="" type="radio"/>	Standard	Frequently accessed data	≥ 3	-
<input type="radio"/>	Intelligent-Tiering	Long-lived data with changing or unknown access patterns	≥ 3	30 days
<input type="radio"/>	Standard-IA	Long-lived, infrequently accessed data	≥ 3	30 days
<input type="radio"/>	One Zone-IA	Long-lived, infrequently accessed, non-critical data	1	30 days
<input type="radio"/>	Glacier	Long-term data archiving with retrieval times ranging from minutes to hours	≥ 3	90 days
<input type="radio"/>	Glacier Deep Archive	Long-term data archiving with retrieval times within 12 hours	≥ 3	180 days
<input type="radio"/>	Reduced redundancy	Frequently accessed, non-critical data	≥ 3	-

Availability and Durability chart

	S3 Standard	S3 Intelligent-Tiering*	S3 Standard-IA	S3 One Zone-IA†	S3 Glacier	S3 Glacier Deep Archive**
Designed for durability	99.999999999% (11 9's)					
Designed for availability	99.99%	99.9%	99.9%	99.5%	N/A	N/A
Availability SLA	99.9%	99%	99%	99%	N/A	N/A
Availability Zones	≥3	≥3	≥3	1	≥3	≥3
Minimum capacity charge per object	N/A	N/A	128KB	128KB	40KB	40KB
Minimum storage duration charge	N/A	30 days	30 days	30 days	90 days	180 days
Retrieval fee	N/A	N/A	per GB retrieved	per GB retrieved	per GB retrieved	per GB retrieved
First byte latency	milliseconds	milliseconds	milliseconds	milliseconds	select minutes or hours	select hours
Storage type	Object	Object	Object	Object	Object	Object
Lifecycle transitions	Yes	Yes	Yes	Yes	Yes	Yes

S3 Bucket Creation:

Create bucket

Buckets are containers for data stored in S3. [Learn more](#)

General configuration

Bucket name

Bucket name must be unique and must not contain spaces or uppercase letters. [See rules for bucket naming](#)

AWS Region

Copy settings from existing bucket - *optional*

Only the bucket settings in the following configuration are copied.

- Choose a Valid bucket name by following the bucket naming limitations.
- Bucket data physically resides in the selected region.

Block Public Access settings for this bucket

Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public access to this bucket and its objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access points. AWS recommends that you turn on Block all public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you require some level of public access to this bucket or objects within, you can customize the individual settings below to suit your specific storage use cases. [Learn more](#)

Block all public access

Turning this setting on is the same as turning on all four settings below. Each of the following settings are independent of one another.

Block public access to buckets and objects granted through *new* access control lists (ACLs)

S3 will block public access permissions applied to newly added buckets or objects, and prevent the creation of new public access ACLs for existing buckets and objects. This setting doesn't change any existing permissions that allow public access to S3 resources using ACLs.

Block public access to buckets and objects granted through *any* access control lists (ACLs)

S3 will ignore all ACLs that grant public access to buckets and objects.

Block public access to buckets and objects granted through *new* public bucket or access point policies

S3 will block new bucket and access point policies that grant public access to buckets and objects. This setting doesn't change any existing policies that allow public access to S3 resources.

Block public and cross-account access to buckets and objects through *any* public bucket or access point policies

S3 will ignore public and cross-account access for buckets or access points with policies that grant public access to buckets and objects.

- If you want to make any data “publicly accessible” we need to disable the above option. If this option is enabled we cannot make any data public.

Bucket Versioning

Versioning is a means of keeping multiple variants of an object in the same bucket. You can use versioning to preserve, retrieve, and restore every version of every object stored in your Amazon S3 bucket. With versioning, you can easily recover from both unintended user actions and application failures. [Learn more](#)

Bucket Versioning

- Disable
 Enable

Tags (0) - optional

Track storage cost or other criteria by tagging your bucket. [Learn more](#)

No tags associated with this bucket.

[Add tag](#)

- Add appropriate tags (based on project/organization standards)

Default encryption
Automatically encrypt new objects stored in this bucket. [Learn more](#)

Server-side encryption

Disable

Enable

▶ **Advanced settings**

Note: After creating the bucket you can upload files and folders to the bucket, and configure additional bucket settings.

Create bucket

- After selection of files, we can give access to other users who required permissions.
- 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

We can use the S3 Versioning feature to preserve, retrieve, and restore every version of every object stored in our buckets. With versioning we can recover more easily from both unintended user actions and application failures. After versioning is enabled for a bucket, if Amazon S3 receives multiple write requests for the same object simultaneously, it stores all of those objects.

- 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, it will overwrite with a Delete Marker, if you want to get that object back to your S3 bucket, you can delete the delete marker.

If you delete an object, Amazon S3 inserts a delete marker instead of removing the object permanently. The delete marker becomes the current object version. If you overwrite an object, it results in a new object version in the bucket. You can always restore the previous version.

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

The screenshot shows the 'Bucket Versioning' section of the AWS S3 console. It has two options: 'Suspend' (radio button is empty) and 'Enable' (radio button is checked). A note below says: 'This suspends the creation of object versions for all operations but preserves any existing object versions.' Below this is a callout box with the text: 'After enabling Bucket Versioning, you might need to update your lifecycle rules to manage previous versions of objects.' Underneath the note is a section titled 'Multi-factor authentication (MFA) delete' with a description: 'An additional layer of security that requires multi-factor authentication for changing Bucket Versioning settings and permanently deleting object versions. To modify MFA delete settings, use the AWS CLI, AWS SDK, or the Amazon S3 REST API.' A link 'Learn more' with a question mark icon is provided. The status 'Disabled' is shown. At the bottom right are 'Cancel' and 'Save changes' buttons.

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/OneZone-IA/Intelligent tier → Glacier / Glacier Deep Archive → Delete

S3-Standard → Glacier/Glacier Deep archive → Delete

S3-Standard → Delete

Steps to enable lifecycle management rules:

- Select the S3 bucket which we want to add lifecycle rule.
- Go to management option of the bucket and click on “create lifecycle rule”.
- Select Add Lifecycle rule and then give a valid name for the life cycle rule. We can limit the transition to a specific prefix or specific tagged objects. We can even apply to the entire buckets objects.

Lifecycle rule name

Up to 255 characters.

Choose a rule scope

- Limit the scope of this rule using one or more filters
- This rule applies to *all* objects in the bucket

Filter type

You can filter objects by prefix, object tags, or a combination of both.

Prefix

Add filter to limit the scope of this rule to a single prefix.

Don't include the bucket name in the prefix. Using certain characters in key names can cause problems with some applications and protocols.

Object tags

You can limit the scope of this rule to the key/value pairs added below.

- After entering “name and scope” we need to configure the transitions. We can configure transitions for current version and previous versions. Navigate to Lifecycle rule actions and add the required transitions.
- For S3-IA We need to store the object for minimum of 30 days and for Glacier 60 days from object creation date.

Lifecycle rule actions

Choose the actions you want this rule to perform. Per-request fees apply. [Learn more](#)  or see [Amazon S3 pricing](#) 

- Transition *current* versions of objects between storage classes
- Transition *previous* versions of objects between storage classes
- Expire *current* versions of objects
- Permanently delete *previous* versions of objects
- Delete expired delete markers or incomplete multipart uploads
When a lifecycle rule is scoped with tags, these actions are unavailable.

Transition current versions of objects between storage classes

Storage class transitions

Standard-IA

Days after object creation

30

Remove transition

Glacier

60

Remove transition

Add transition**Transition noncurrent versions of objects between storage classes**

Storage class transitions

Glacier

Days after objects become noncurrent

2

Remove transition

Add transition

- 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.

Expire current versions of objects

For version-enabled buckets, Amazon S3 adds a delete marker and the current version of an object is retained as a previous version. For non-versioned buckets, Amazon S3 permanently removes the object. [Learn more](#) 

Number of days after object creation

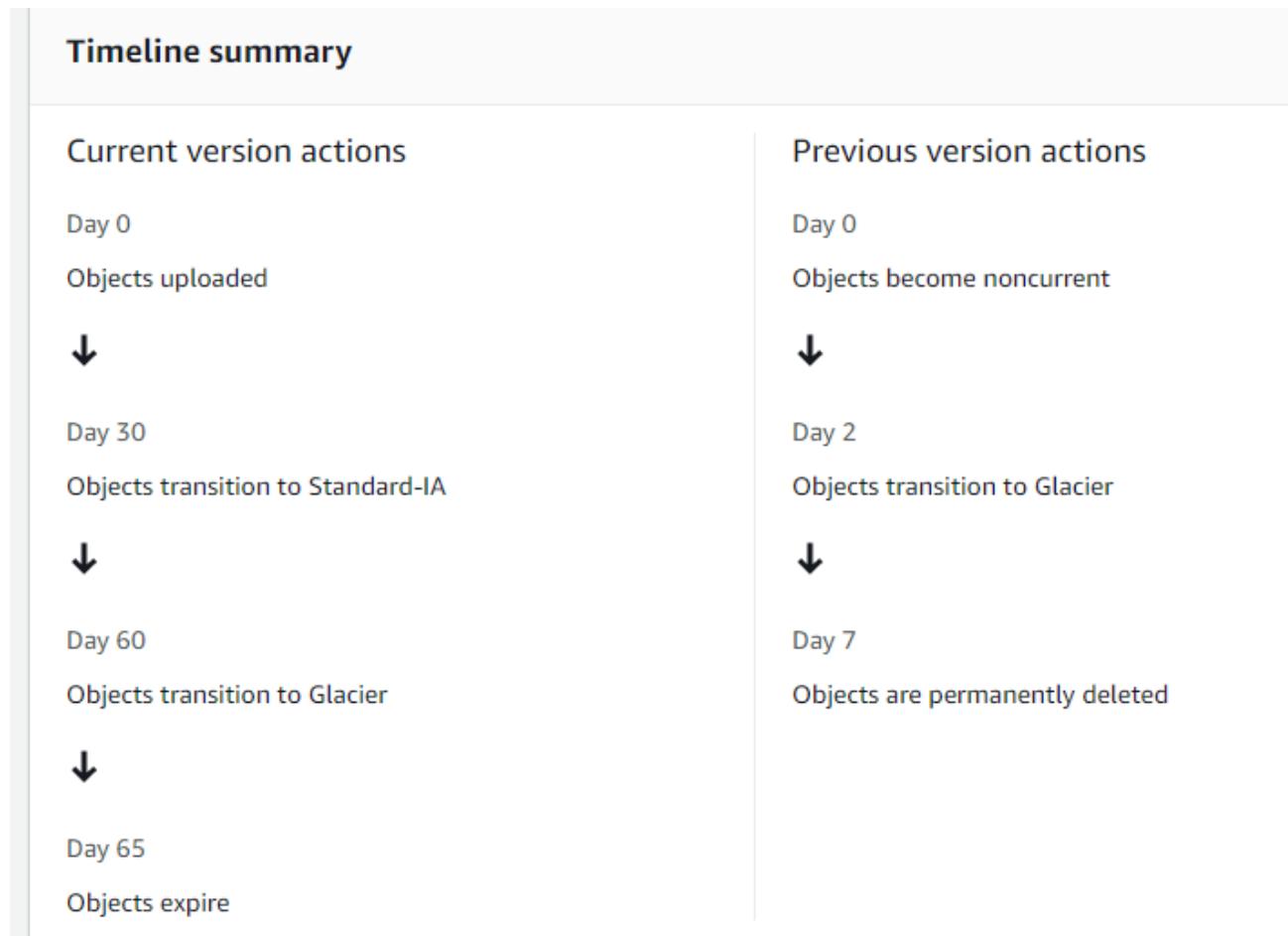
65

Permanently delete previous versions of objects

Number of days after objects become previous versions

7

- 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.



- In the above example, current versioned objects configured to move to “Standard-IA” on Day 30, Glacier on “Day 60” and Expire on “Day 65”.
- For previous versioned objects, Moving to Glacier on “Day 2” and Permanently Deletes on “Day 7”.

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

Server access logging

Log requests for access to your bucket. [Learn more](#) 

Server access logging

Disable

Enable

⚠️ By enabling server access logging, S3 console will automatically update your bucket access control list (ACL) to include access to the S3 log delivery group.

Target bucket

s3://avinashreddy [Browse S3](#)

Format: s3://bucket/prefix

Cross-Region Replication / Same 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.

If Source bucket and Destination bucket is in same region we call it as “Same Region Replication”.

- Versioning must be enabled on both the source and destination buckets.
- Files in an existing bucket are not replicated automatically. All subsequent/future updated files will be replicated automatically.
- **We can have multiple destination buckets from single source bucket.**
- 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.

Create replication rule

Replication rule configuration

Replication rule name

Up to 255 characters.

Status

Choose whether the rule will be enabled or disabled when created.

- Enabled
 Disabled

Priority

The priority value resolves conflicts that occur when an object is eligible for replication under multiple rules to the same destination. The rule is added to the configuration at the highest priority and the priority can be changed on the replication rules table.

0

- 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)

Source bucket

Source bucket name

avinashreddy

Source Region

Asia Pacific (Mumbai) ap-south-1

Choose a rule scope

- Limit the scope of this rule using one or more filters
 This rule applies to *all* objects in the bucket

- 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

Destination

Destination
You can replicate objects across buckets in different AWS Regions (Cross-Region Replication) or you can replicate objects across buckets in the same AWS Region (Same-Region Replication). You can also specify a different bucket for each rule in the configuration. [Learn more](#) or see [Amazon S3 pricing](#)

Choose a bucket in this account
 Specify a bucket in another account

Bucket name
Choose the bucket that will receive replicated objects.
 [Browse S3](#)

Destination Region
Asia Pacific (Singapore) ap-southeast-1

IAM role

IAM role
[Create new role](#)

Encryption

Replicate objects encrypted with AWS KMS
You can use replication for AWS Key Management Service encrypted objects to replicate data encrypted using AWS KMS across AWS Regions.

Destination storage class

Amazon S3 offers a range of storage classes designed for different use cases. [Learn more](#) or see [Amazon S3 pricing](#)

Change the storage class for the replicated objects

- We have to create an IAM role for replication. Click on IAM role and choose “Create new role” option.
- We can enforce the encryption for the destination buckets replicating data.
- We can change the object storage class for the destination bucket, if required.

Replication Time Control / RTC : Replication Time Control replicates 99.99% of new objects within 15 minutes and provides replication metrics and notifications. If we want to replicate the data much faster to the destination bucket this feature will help us. Additional charges applicable.

- Replication Time Control (RTC)**
Replication Time Control replicates 99.99% of new objects within 15 minutes and provides replication metrics and notifications.
Additional fees will apply. [Learn more](#)
- Replication metrics and notifications**
Monitor the progress of your replication rule through Cloudwatch Metrics. Cloudwatch metrics fees apply. [Learn more](#) or see [Amazon Cloudwatch pricing](#)
- Delete marker replication**
Delete markers created by S3 delete operations will be replicated. Delete markers created by lifecycle rules are not replicated. [Learn more](#)
- Replica modification sync**
Replicate metadata changes made to replicas in this bucket to the destination bucket. [Learn more](#)

- 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.

Replication rules (1)
Use replication rules to define options you want Amazon S3 to apply during replication such as server-side encryption, replica ownership, transitioning replicas to another storage class, and more. [Learn more](#)

<input type="checkbox"/>	View details	Edit rule	Delete	Actions ▾	Create replication rule
					◀ 1 ▶ ⌂

Replication rule name	Status	Destination bucket	Destination Region	Priority	Scope	Storage class	Replica owner	Replication Time Control	KMS-encrypted objects	Replica modification sync
avinash_replicatest	Enabled	s3://avinashreddy.replica	Asia Pacific (Singapore) ap-southeast-1	0	Entire bucket	Same as source	Same as source	Disabled	Do not replicate	Disabled

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.

Static website hosting

Use this bucket to host a website or redirect requests. [Learn more](#)

Static website hosting

Disable
 Enable

Hosting type

Host a static website
Use the bucket endpoint as the web address. [Learn more](#)

Redirect requests for an object
Redirect requests to another bucket or domain. [Learn more](#)

Index document

Specify the home or default page of the website.

`index.html`

Error document - *optional*

This is returned when an error occurs.

`error.html`

Redirection rules – *optional*

Redirection rules, written in JSON, automatically redirect webpage requests for specific content. [Learn more](#)

Object Lock:

S3 Object Lock, you can prevent an object from being deleted or overwritten for a fixed amount of time or indefinitely. S3 Object Lock enables you to meet regulatory requirements that require WORM (Write Once and Read Many) storage or simply to add an additional layer of protection against object changes and deletion.

After bucket creation, upload an object and navigate to object properties to lock a specific object. S3 Object Lock provides two ways to manage object retention: retention periods and legal holds. A retention period specifies a fixed period of time during which an object remains locked. During this period, your object will be WORM-protected and can't be overwritten or deleted.

To use Amazon S3 Object Lock, you take the following steps:

1. Create a new bucket with S3 Object Lock enabled.
2. Place the objects that you want to lock in the bucket.
3. Apply a retention period, a legal hold, or both, to the objects that you want to protect.
4. We cannot enable Object lock option to an existing bucket. Need to enable while bucket creation.
5. Versioning must be enabled if we want to use Object lock option.

Expand the “Advanced Options” while creating bucket and enable the Object lock as shown in below image.

▼ Advanced settings

Object Lock

Store objects using a write-once-read-many (WORM) model to help you prevent objects from being deleted or overwritten for a fixed amount of time or indefinitely. [Learn more](#)

Disable

Enable

Permanently allows objects in this bucket to be locked. Additional Object Lock configuration is required in bucket details after bucket creation to protect objects in this bucket from being deleted or overwritten.

 Object Lock works only in versioned buckets. Enabling Object Lock automatically enables Bucket Versioning.



Enabling Object Lock will permanently allow objects in this bucket to be locked

Enable Object Lock only if you need to prevent objects from being deleted to have data integrity and regulatory compliance. After you enable this feature, anyone with the appropriate permissions can put immutable objects in the bucket. You might be blocked from deleting the objects and the bucket.

Additional Object Lock configuration is required in bucket details after bucket creation to protect objects in this bucket from being deleted or overwritten. [Learn more](#)

I acknowledge that enabling Object Lock will permanently allow objects in this bucket to be locked.

A legal hold provides the same protection as a retention period, but has no expiration date. Instead, a legal hold remains in place until you explicitly remove it. Legal holds are independent from retention periods: an object version can have both a retention period and a legal hold, one but not the other, or neither.

Governance mode: Governance mode can be disabled by AWS accounts that have specific IAM permissions.

Compliance mode: Compliance mode cannot be disabled by any user, including the root account.

Object Lock
Enabled

Default retention
Automatically protect new objects put into this bucket from being deleted or overwritten.

Disable
 Enable

Default retention mode

Governance
Users with specific IAM permissions can overwrite or delete protected object versions during the retention period.

Compliance
No users can overwrite or delete protected object versions during the retention period.

Default retention period

2	Days
---	------

Must be a positive whole number.

Cancel Save changes

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.

Edit bucket tagging

Tags
Track storage cost or other criteria by tagging your bucket. [Learn more](#)

Key	Value - optional	
Project	Project X	Remove
Cost Center	AAZAA	Remove
Add tag		

Cancel Save changes

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.

The screenshot shows the 'Transfer acceleration' section of the AWS S3 Bucket Properties. It includes a note about using an accelerated endpoint for faster transfers, a radio button for enabling transfer acceleration (which is selected), and a text input field for the accelerated endpoint URL (set to 'avinashreddy.s3-accelerate.amazonaws.com'). A callout box provides information about additional fees for using the accelerated endpoint. At the bottom are 'Cancel' and 'Save changes' buttons.

Transfer acceleration
Use an accelerated endpoint for faster data transfers. [Learn more](#)

Transfer acceleration

Disable

Enable

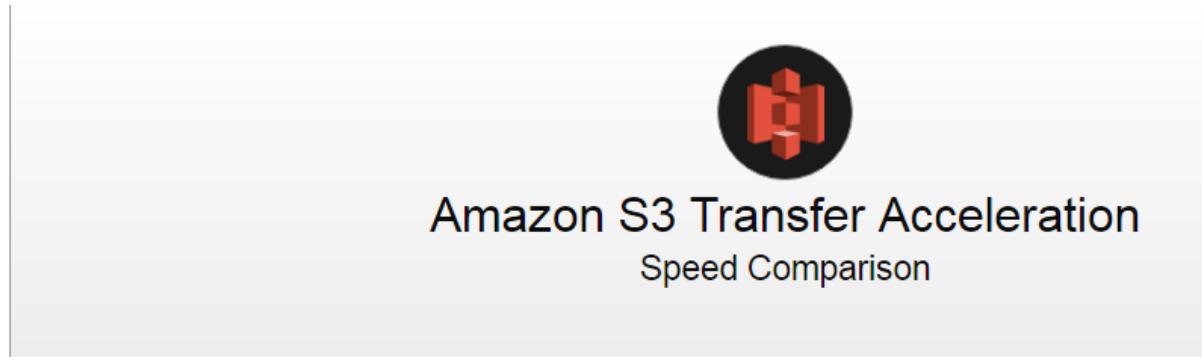
Accelerated endpoint

avinashreddy.s3-accelerate.amazonaws.com

i Use the accelerated endpoint for faster data transfers, which will incur an additional fee. See [Amazon S3 pricing](#)

Cancel Save changes

Here is a sample result for Transfer acceleration result.



Upload speed comparison in the selected region
(Based on the location of bucket: avizway)

Mumbai

(AP-SOUTH-1)

1% slower

S3 Direct Upload Speed



Upload complete

S3 Accelerated Transfer Upload Speed



Upload complete

This speed comparison shows the difference in upload speed when using Amazon S3 Transfer Acceleration compared to using S3 directly. The speed difference can vary depending on the region and the type of file being transferred.

Note: In general, using Amazon S3 Transfer Acceleration can improve upload speeds. You may see similar results in other regions, as the system's performance depends on the specific network conditions.

Upload speed comparison in other regions

San Francisco

(US-WEST-1)

8% faster

S3 Direct Upload Speed



Upload complete

S3 Accelerated Transfer Upload Speed



Upload complete

Oregon

(US-WEST-2)

18% faster

S3 Direct Upload Speed



Upload complete

S3 Accelerated Transfer Upload Speed



Upload complete

Dublin

(EU-WEST-1)

9% faster

S3 Direct Upload Speed



Upload complete

S3 Accelerated Transfer Upload Speed



Upload complete

Frankfurt

(EU-CENTRAL-1)

17% faster

S3 Direct Upload Speed



Upload complete

S3 Accelerated Transfer Upload Speed



Upload complete

Tokyo

(AP-NORTHEAST-1)

39% faster

S3 Direct Upload Speed



Upload complete

S3 Accelerated Transfer Upload Speed



Upload complete

Seoul

(AP-NORTHEAST-2)

7% faster

S3 Direct Upload Speed



Upload complete

S3 Accelerated Transfer Upload Speed



Upload complete

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 and Search for Simple Notifications Service. In SNS dashboard, we have to create topic in SNS service and edit the Topic Policy to publish through S3.
- Click on “Create Topic” and choose “Standard”, Provide a Valid name and Display name.

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

FIFO (first-in, first-out)

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

Standard

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

Name

Maximum 256 characters. Can include alphanumeric characters, hyphens (-) and underscores (_).

Display name - *optional*
To use this topic with SMS subscriptions, enter a display name. Only the first 10 characters are displayed in an SMS message. [Info](#)

Maximum 100 characters, including hyphens (-) and underscores (_).

Make sure you expand the “Access Policy” and Chosse **Everyone** under “Who can publish message” to the topic. If you don’t perform this s3 cannot use the SNS to send notifications.

▼ Access policy - optional

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

Choose method

Basic

Use simple criteria to define a basic access policy

Advanced

Use a JSON object to define an advanced access policy.

Define who can publish messages to the topic

- Only the topic owner
Only the owner of the topic can publish to the topic
- Everyone
Anybody can publish
- Only the specified AWS accounts
Only the specified AWS account IDs can publish to the topic

JSON preview

```
{
  "Version": "2012-10-17",
  "Id": "__defaultPolicy",
  "Statement": [
    {
      "Sid": "__defaultStmt",
      "Effect": "Allow",
      "Principal": "*",
      "Action": "sns:Publish",
      "TopicArn": "arn:aws:sns:ap-south-1:668026757480:S3_Notifications"
    }
  ]
}
```

After creating topic, Add the email id for subscription of notifications. Once we select confirm option from email id then that email got subscribed for event notifications.

Navigate to the “**Subscriptions**” option in SNS topic, and click on “**Create Subscription**”, and enter Choose “**protocol as Email**” and provide Valid Email ID under Endpoint.

Once you create subscription you need to login to the given email ID, and click on the **subscription confirmation email**.

Create subscription

Details

Topic ARN

Protocol

The type of endpoint to subscribe

Endpoint

An email address that can receive notifications from Amazon SNS.

AWS Notification - Subscription Confirmation ➔ Inbox x

**IMP S3 Notifications**

to me ▾

You have chosen to subscribe to the topic:

arn:aws:sns:ap-south-1: :S3_Notifications

To confirm this subscription, click or visit the link below (If this was in error no action is necessary):

[Confirm subscription](#)**Simple Notification Service**

Subscription confirmed!

You have successfully subscribed.

Your subscription's id is:

arn:aws:sns:ap-south-1: :S3_Notifications:5568b6f7-e3f7-466a-

If it was not your intention to subscribe, [click here to unsubscribe](#).

- 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.

Event types

Specify at least one type of event for which you want to receive notifications. [Learn more](#)

- All object create events
s3:ObjectCreated:
 - Put
s3:ObjectCreated:Put
 - Post
s3:ObjectCreated:Post
 - Copy
s3:ObjectCreated:Copy
 - Multipart upload completed
s3:ObjectCreated:CompleteMultipartUpload

Destination

Destination
Choose a destination to publish the event. [Learn more](#) 

Lambda function
Run a Lambda function script based on S3 events.

SNS topic
Send notifications to email, SMS, or an HTTP endpoint.

SQS queue
Send notifications to an SQS queue to be read by a server.

Specify SNS topic

Choose from your SNS topics

Enter SNS topic ARN

SNS topic

S3_Notifications 

[Cancel](#) [Save changes](#)

- 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.

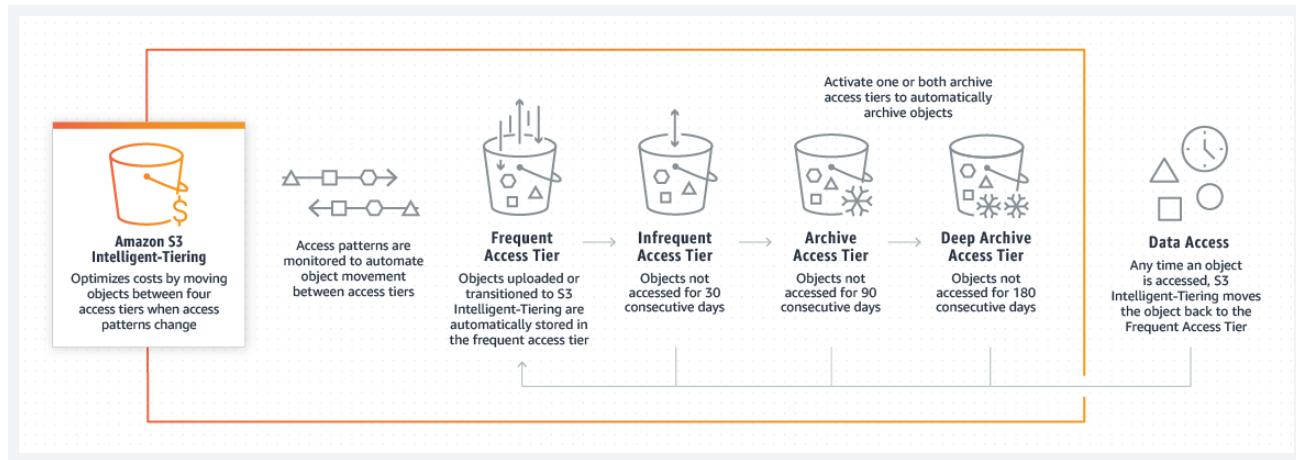
Requester pays

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.

Intelligent-Tiering Archive configurations: S3 Intelligent-Tiering is a storage class that is designed to optimize storage costs by automatically moving data to the most cost-effective access tier without performance impact or operational overhead.

- **Archive Access tier :** When enabled, Intelligent-Tiering will automatically move objects that haven't been accessed for a minimum of 90 days to the Archive Access tier.
- **Deep Archive Access tier :** When enabled, Intelligent-Tiering will automatically move objects that haven't been accessed for a minimum of 180 days to the Deep Archive Access tier.



S3 Performance optimization :

In S3 platform, we can achieve at least 3,500 PUT/COPY/POST/DELETE or 5,500 GET/HEAD requests per second per prefix in a bucket.

For example, if you create 10 prefixes in an Amazon S3 bucket to parallelize reads, you could scale your read performance to 55,000 read requests per second. Similarly, you can scale write operations by writing to multiple prefixes.

S3 Consistency Models :

S3 provides **Read-after-Write consistency for PUTS** of new objects.

S3 provides **Eventual Consistency for overwrite PUTS and Deletes**.

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.

Bucket policies : We can create and configure bucket policies to grant permission to Amazon S3 resources. Bucket policies use JSON-based access policy language.

You can use the AWS Policy Generator to create a bucket policy for your Amazon S3 bucket. While generating policy we need to provide the required resource, effect, action and user information.

Effect : Allow / Deny

Principal : IAM user/Group ARN

Action : What operations you want to Allow / Deny.

ARN : ARN of S3 Bucket you want to apply this policy.

Step 2: Add Statement(s)

A statement is the formal description of a single permission. See [a description of elements](#) that you can use in statements.

Effect Allow Deny

Principal

AWS Service All Services ('*')

Actions PutInventoryConfiguration
 PutJobTagging
 PutLifecycleConfiguration
 PutMetricsConfiguration
 PutObject
 PutObjectAcl
 PutObjectLegalHold
 PutObjectRetention

Amazon Resource Name (ARN)

bucket_name>/<key_name>,

All Actions ('*')

Choose “Add Statement” and click on “generate Policy”.

Below sample policy can deny “PUT” operation for an IAM user “avinash” on “avinashreddy” bucket.

```
{
  "Id": "Policy1622115926433",
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Stmt1622115924486",
      "Action": [
        "s3:PutObject"
      ],
      "Effect": "Deny",
      "Resource": "arn:aws:s3:::avinashreddy/*",
      "Principal": {
        "AWS": [
          "arn:aws:iam::000000000000:user/avinash"
        ]
      }
    }
  ]
}
```

Close

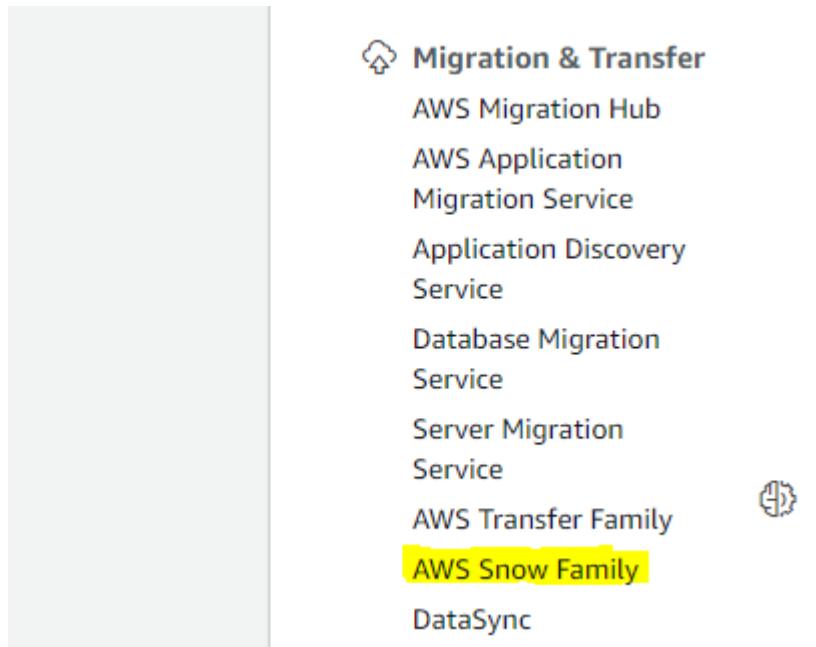
AWS SNOWBALL

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 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)

The screenshot shows the 'Choose a Snow job type' step in the AWS Snowball wizard. It displays three options:

- Import into Amazon S3** (selected): AWS will ship an empty device to you for storage and compute workloads. You'll transfer your data onto it, and ship it back. After AWS gets it, your data will be moved.
- Export from Amazon S3**: Choose what data you want to export from your S3 buckets for storage and compute workloads. AWS will load that data onto a device and ship it to you. When you're done ship the device back for erasing.
- Local compute and storage only**: Perform local compute and storage workloads, without transferring data. You can order multiple devices in a cluster for increased durability and storage capacity.

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.

Choose your shipping preferences Info

Shipping address Info

- Use recent address
- Add a new address

Name

Ex. John Doe

City

Ex. Seattle

Shipping speed

Your selection here will incur the respective shipping charges

- Express Shipping
- Standard Shipping

Tax Information Info

This information is required for taxation purposes by the country to which your device is being shipped.

GSTIN

123456789

Name your job [Info](#)

Your job will be created in the Asia Pacific (Mumbai) region.

Job name

Choose your Snow device [Info](#)**Snowball Edge Storage Optimized**

Storage (HDD) Memory
80 TB 32 GB

Storage (SSD) Compute
- 24 vCPUs

Snowball Edge Compute Optimized

Storage (HDD) Memory
39.5 TB 208 GB

Storage (SSD) Compute
7.68 TB 52 vCPUs

Snowball Edge Compute Optimized with GPU

Storage (HDD) Memory
39.5 TB 208 GB

Storage (SSD) Compute
7.68 TB 52 vCPUs, GPU

Choose your S3 storage [Info](#)

The S3 buckets you choose will appear as directories on your device. The data in these directories will be transferred back to S3.

 [Search for an item](#)

S3 bucket name

Date created



avinashreddy

5/27/2021, 4:00:06 PM GMT+5:30

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.

Encryption [Info](#)

Select the AWS KMS key to encrypt your data.

KMS key

aws/importexport (default)

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 (ELASTIC 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
- FPGA Instances
- GPU Graphics
- GPU Instances

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

General Purpose

General purpose instances provide a balance of compute, memory and networking resources, and can be used for a variety of diverse workloads. These instances are ideal for applications that use these resources in equal proportions such as web servers and code repositories.

Mac	T4g	T3	T3a	T2	M6g	M5	M5a	M5n	M5zn	M4	A1
-----	-----	----	-----	----	-----	----	-----	-----	------	----	----

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

Compute Optimized

Compute Optimized instances are ideal for compute bound applications that benefit from high performance processors. Instances belonging to this family are well suited for batch processing workloads, media transcoding, high performance web servers, high performance computing (HPC), scientific modeling, dedicated gaming servers and ad server engines, machine learning inference and other compute intensive applications.

C6g	C6gn	C5	C5a	C5n	C4
-----	------	----	-----	-----	----

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.

Accelerated Computing

Accelerated computing instances use hardware accelerators, or co-processors, to perform functions, such as floating point number calculations, graphics processing, or data pattern matching, more efficiently than is possible in software running on CPUs.

P4 P3 P2 Inf1 G4dn G4ad G3 F1

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.

Memory Optimized

Memory optimized instances are designed to deliver fast performance for workloads that process large data sets in memory.

R6g R5 R5a R5b R5n R4 X2gd X1e X1 High Memory z1d

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.

Storage Optimized

Storage optimized instances are designed for workloads that require high, sequential read and write access to very large data sets on local storage. They are optimized to deliver tens of thousands of low-latency, random I/O operations per second (IOPS) to applications.

I3 I3en D2 D3 D3en H1

Instance launch pricing Options:

- On-Demand Instances
- Reserved Instances
- Spot Instances

On-Demand Instances:

The price **per hour/ per Second (with min of 60 sec)** 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 require up-front commitment.
- We will have control over when the instance is launched and when it is terminated.
- Suitable for unpredictable workloads.

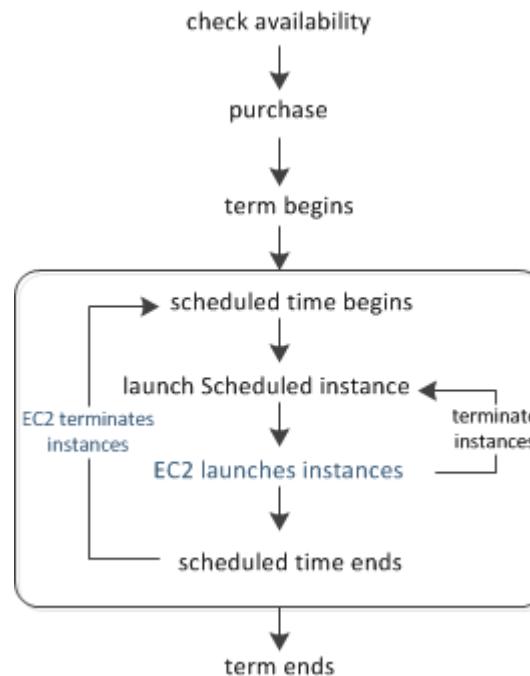
Reserved Instances:

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 three offering classes in RI: Convertible, Standard and Scheduled.

Standard reserved Instances: These provide the most significant discount (up to 75% off On-Demand) and are best suited for steady-state usage.

Convertible reserved Instances: These provide a discount (up to 54% off On-Demand) and the capability to change the attributes of the RI as long as the exchange results in the creation of Reserved Instances of equal or greater value. Like Standard RIs, Convertible RIs are best suited for steady-state usage.

Scheduled Reserved Instances: These are available to launch within the time windows you reserve. This option allows you to match your capacity reservation to a predictable recurring schedule that only requires a fraction of a day, a week, or a month.



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.

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.

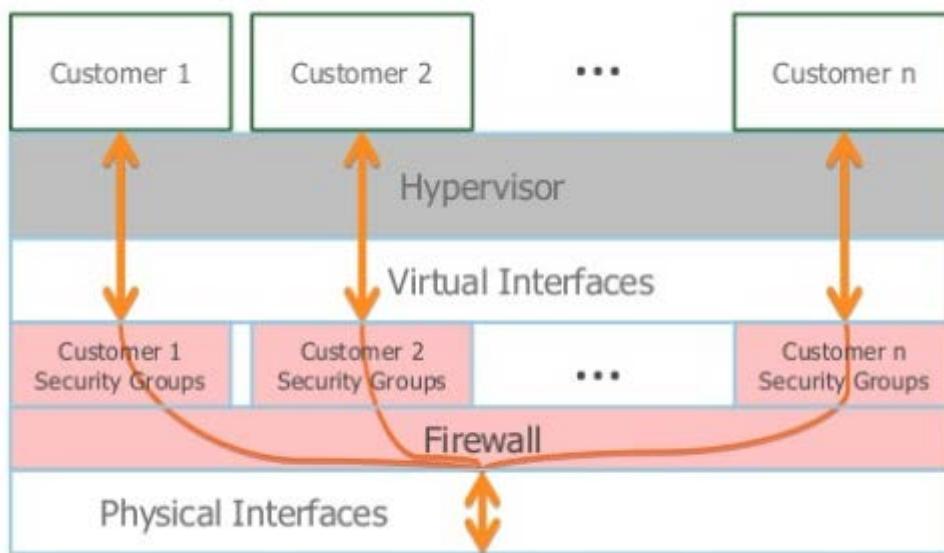
Tenancy Options:

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.

EC2 Instance Isolation Diagram:



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 find 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:

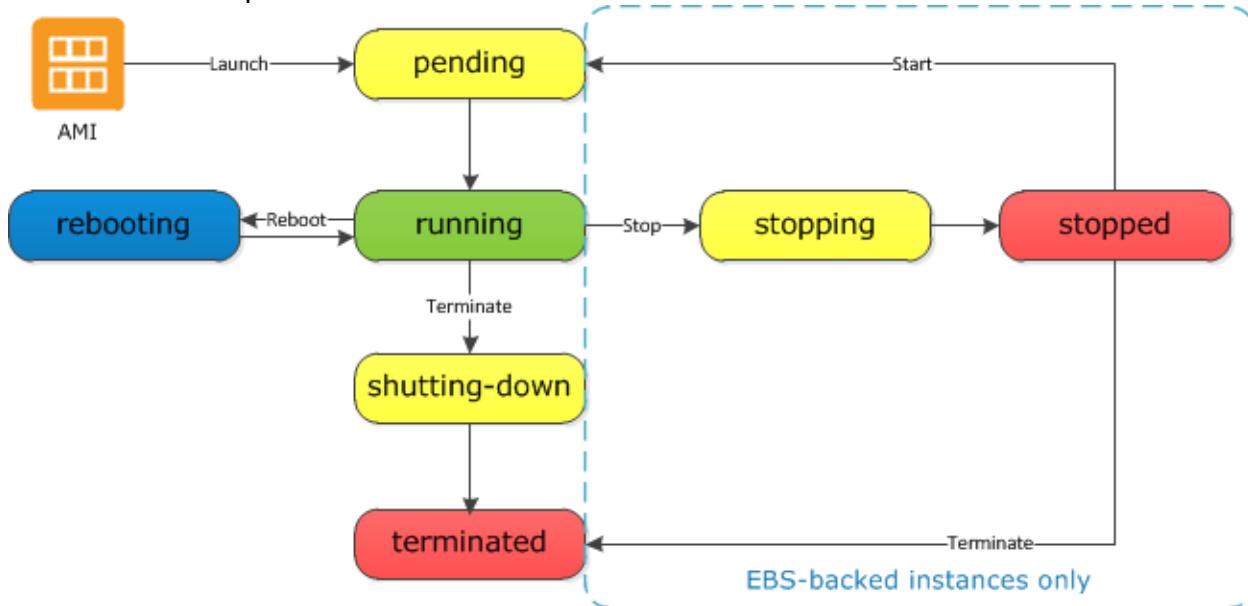
1. Login to AWS account and navigate to Amazon EC2 console.
2. In the navigation pane, choose **Elastic IPs**.
3. Choose Allocate new address.
4. Select Allocate. Close the confirmation screen.

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.

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.

The screenshot shows the AWS services dashboard with the following details:

- AWS services** header.
- Recently visited services**: EC2 (highlighted).
- All services** section:
 - Compute**: EC2 (highlighted), Lightsail, Lambda, Batch.
 - Quantum Technologies**: Amazon Braket.
 - Security, Identity, & Compliance**: IAM, Resource Access Manager, Cognito.

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

Resources

You are using the following Amazon EC2 resources in the Asia Pacific (Mumbai) Region:

Instances (running)	0	Dedicated Hosts	0
Elastic IPs	0	Instances	0
Key pairs	0	Load balancers	0
Placement groups	0	Security groups	1
Snapshots	0	Volumes	0

Launch instance

Launch instance Launch instance from template

Launch instance, which is a virtual server in the cloud.

Launch instance ▲

Note: Your instances will launch in the Asia Pacific (Mumbai) Region

Step 1 : Choose an Amazon Machine Image (AMI)

I want to launch an Amazon Linux 2 AMI, so selecting Amazon Linux AMI from the Quick Start menu.

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

[Cancel and Exit](#)

Step 1: Choose an Amazon Machine Image (AMI)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. You can select an AMI provided by AWS, our user community, or the AWS Marketplace; or you can select one of your own AMIs.

Search for an AMI by entering a search term e.g. "Windows" X

Search by Systems Manager parameter < < 1 to 17 of 17 AMIs > >

Quick Start

- My AMIs
- Amazon Linux Free tier eligible
- AWS Marketplace
- Community AMIs

Free tier only (1)

 Amazon Linux 2 AMI (HVM), SSD Volume Type - ami-0ad704c126371a549 (64-bit x86) / ami-0a05821577033f94d (64-bit Arm)	<input checked="" type="radio"/> 64-bit (x86) <input type="radio"/> 64-bit (Arm)
<small>Amazon Linux 2 comes with five years support. It provides Linux kernel 4.14 tuned for optimal performance on Amazon EC2, systemd 219, GCC 7.3, Glibc 2.26, Binutils 2.29.1, and the latest software packages through extras. This AMI is the successor of the Amazon Linux AMI that is approaching end of life on December 31, 2020 and has been removed from this wizard.</small>	
<small>Root device type: ebs Virtualization type: hvm ENA Enabled: Yes</small>	
 Red Hat Enterprise Linux 8 (HVM), SSD Volume Type - ami-06a0b4e3b7eb7a300 (64-bit x86) / ami-0cbe04a3ce796c98e (64-bit Arm)	Select (1)
<small>Red Hat Enterprise Linux 8 is a stable, enterprise-grade Linux distribution designed for mission-critical workloads. It includes the latest security patches and updates, and is supported for five years.</small>	

- 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.

Step 2 : 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 comes under the free tier eligibility and configuration is 1 vCPU and 1 GB of RAM.

Step 2: Choose an Instance Type

Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instances are virtual servers that can run applications. They have varying combinations of CPU, memory, storage, and networking capacity, and give you the flexibility to choose the appropriate mix of resources for your applications. [Learn more](#) about instance types and how they can meet your computing needs.

Filter by: All instance families Current generation Show/Hide Columns								
Currently selected: t2.micro (- ECUs, 1 vCPUs, 2.5 GHz, -, 1 GiB memory, EBS only)								
	Family	Type	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance	IPv6 Support
<input type="checkbox"/>	t2	t2.nano	1	0.5	EBS only	-	Low to Moderate	Yes
<input checked="" type="checkbox"/>	t2	t2.micro Free tier eligible	1	1	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	t2	t2.small	1	2	EBS only	-	Low to Moderate	Yes

Step 3 : Configure instance details

Step 3: Configure Instance Details

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage of the instance, and more.

Number of instances i	<input type="text" value="1"/> Launch into Auto Scaling Group i
Purchasing option i	<input type="checkbox"/> Request Spot instances
Network i	<input type="text" value="vpc-07232c8c132e8512f (default)"/> C Create new VPC
Subnet i	<input type="text" value="No preference (default subnet in any Availability Zone)"/> C Create new subnet
Auto-assign Public IP i	<input type="text" value="Use subnet setting (Enable)"/>
Placement group i	<input type="checkbox"/> Add instance to placement group
Capacity Reservation i	<input type="text" value="Open"/>
Domain join directory i	<input type="text" value="No directory"/> C Create new directory
IAM role i	<input type="text" value="None"/> C Create new IAM role
Shutdown behavior i	<input type="text" value="Stop"/>
Stop - Hibernate behavior i	<input type="checkbox"/> Enable hibernation as an additional stop behavior
Enable termination protection i	<input type="checkbox"/> Protect against accidental termination
Monitoring i	<input type="checkbox"/> Enable CloudWatch detailed monitoring <small>Additional charges apply.</small>
Tenancy i	<input type="text" value="Shared - Run a shared hardware instance"/> C Additional charges will apply for dedicated tenancy.
Credit specification i	<input type="checkbox"/> Unlimited <small>Additional charges may apply</small>
File systems i	Add file system C Create new file system

▼ Advanced Details

Enclave	<input type="checkbox"/> Enable
Metadata accessible	<input type="button" value="Enabled"/>
Metadata version	<input type="button" value="V1 and V2 (token optional)"/>
Metadata token response hop limit	<input type="button" value="1"/>
User data	<input checked="" type="radio"/> As text <input type="radio"/> As file <input type="checkbox"/> Input is already base64 encoded <input type="text" value="Optional"/>

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. We can launch this under Spot category.

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.

Placement Groups: A placement group is a logical grouping of instances within a single Availability Zone. We have three types of Placement groups.

1. Cluster Placement groups
 2. Partition Placemnt groups
 3. Spread Placement groups
- Recommended for applications that benefit from low network latency, high network throughput, or both.
 - 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.

Capacity Reservations enable you to reserve compute capacity for your Amazon EC2 instances in a specific Availability Zone for any duration. By creating Capacity Reservations, we can ensure that we always have access to EC2 capacity when we need it, for as long as we need it.

Domain join directory : We can join launching ec2 instance into a Directory (if we have any).

IAM role: You can additionally select a particular IAM role to be associated with your instance. An IAM role automatically deploys AWS credentials to resources that assume it. We can access aws services from instance without configuring IAM Credentials.

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.

Stop - Hibernate behavior: Hibernation stops our instance and saves the contents of the instance's RAM to the root volume. We cannot enable hibernation after launch.

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.

File systems: While launching ec2 instance we can mount an EFS file system, ignore now.

Metadata accessible : We can turn on/off access to our instance metadata. Metadata is nothing but data about the data.

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.

Step 4: Add Storage

Your instance will be launched with the following storage device settings. You can attach additional EBS volumes and instance store volumes to your instance, or edit the settings of the root volume. You can also attach additional EBS volumes after launching an instance, but not instance store volumes. [Learn more](#) about storage options in Amazon EC2.

Volume Type	Device	Snapshot	Size (GiB)	Volume Type	IOPS	Throughput (MB/s)	Delete on Termination	Encryption
Root	/dev/xvda	snap-041de31327a858319	8	General Purpose SSD (gp2)	100 / 3000	N/A	<input checked="" type="checkbox"/>	Not Encrypted

[Add New 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 5: Add Tags

A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = Webserver.

A copy of a tag can be applied to volumes, instances or both.

Tags will be applied to all instances and volumes. [Learn more](#) about tagging your Amazon EC2 resources.

Key	(128 characters maximum)	Value	(256 characters maximum)	Instances	Volumes	Network Interfaces
Name		First Server		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Project		Project X		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Platform		Linux		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Cost Center		AAZAA		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

[Add another tag](#) (Up to 50 tags maximum)

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. A security group acts as a virtual firewall for your EC2 instances to control incoming and outgoing traffic. Inbound rules control the incoming traffic to your instance, and outbound rules control the outgoing traffic from your 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.

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

Security group name:
Description:

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

[Add Rule](#)

⚠ Warning

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

- 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:

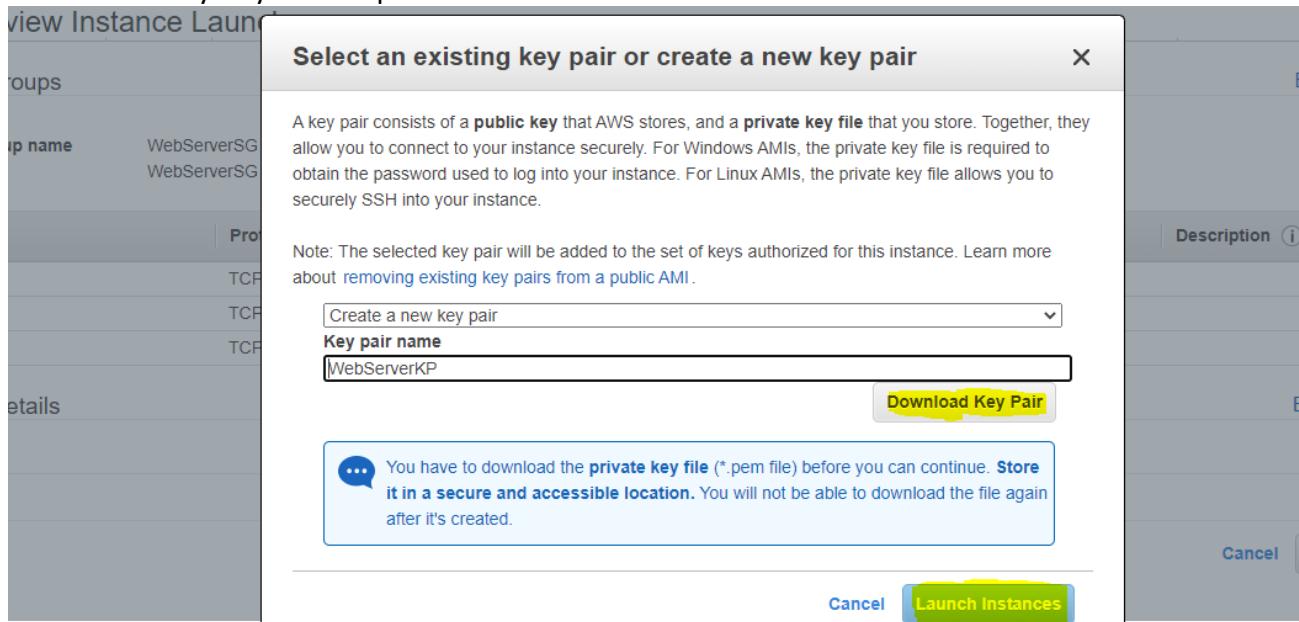
- Security group rules are always permissive; you can't create rules that deny access.
- You can add up to 50 inbound and 50 outbound rules to each security group. If you need to apply more than 100 rules to an instance, you can associate up to five security groups with each network interface.
- You 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 screenshot shows the AWS EC2 Instances dashboard. At the top, there's a navigation bar with tabs for 'Instances (1/1)', 'Info', and several action buttons like 'Connect', 'Actions', and 'Launch instances'. Below the navigation is a search bar labeled 'Filter instances'. The main table lists one instance: 'First Server' with ID 'i-0881f8a6d9c1c26da'. The instance is shown as 'Running' with a green checkmark, type 't2.micro', and a status check of 'Initializing'. It has no alarms and is in the 'ap-south-1a' availability zone. Below the table, a modal window is open for the instance 'i-0881f8a6d9c1c26da (First Server)'. The 'Details' tab is selected, showing the 'Instance summary' section. This section contains details like Instance ID (i-0881f8a6d9c1c26da), Instance state (Running), Public IPv4 address (13.233.172.45), Public IPv4 DNS (ec2-13-233-172-45.ap-south-1.compute.amazonaws.com), and Private IPv4 addresses (172.31.41.106). Other tabs in the modal include 'Security', 'Networking', 'Storage', 'Status checks', 'Monitoring', and 'Tags'.

- 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.

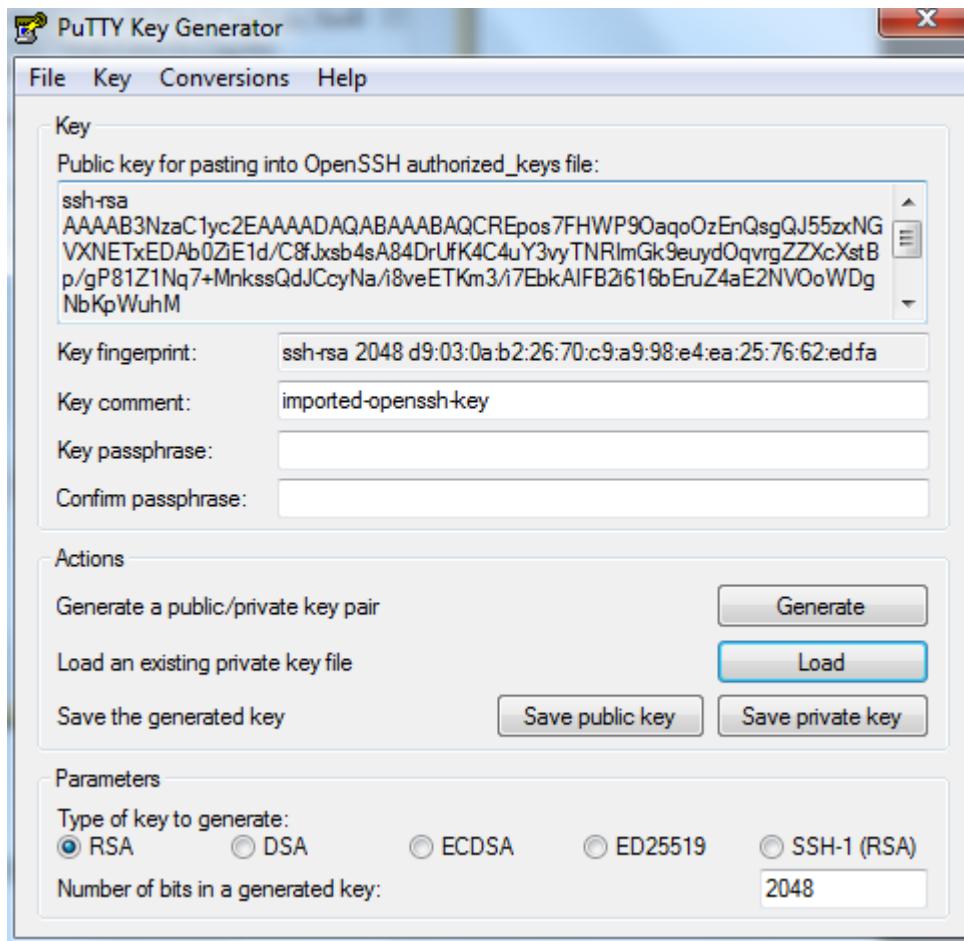
As we launched Linux machine, here we are going to see PuTTY option now.

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).

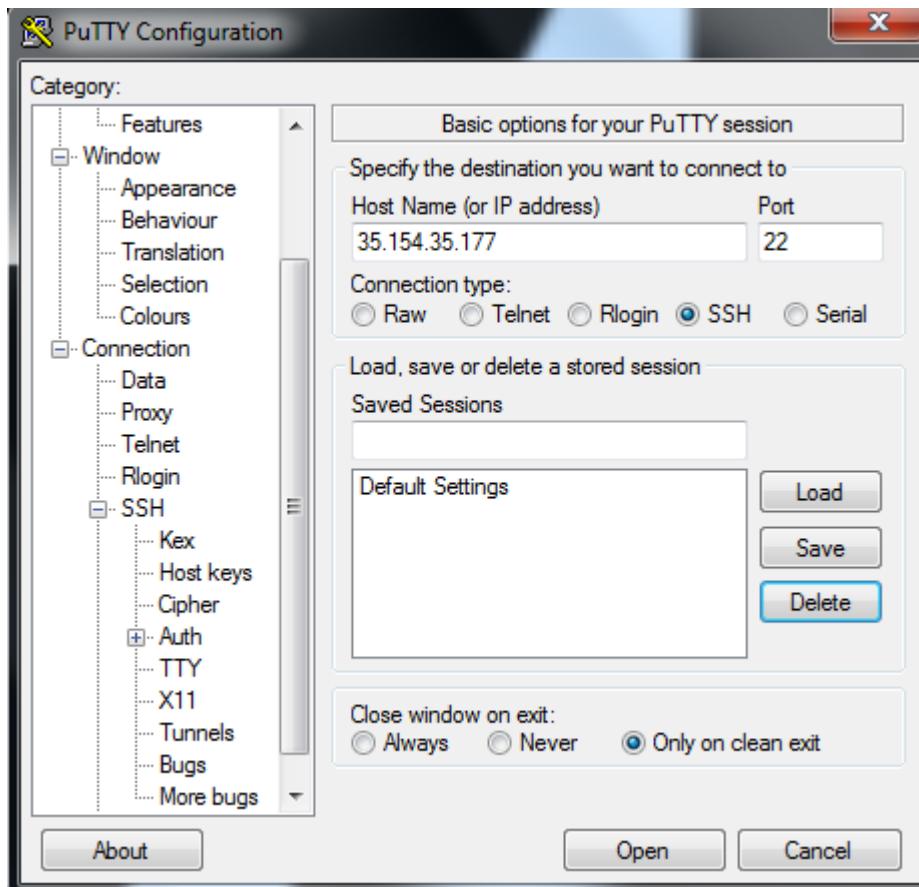
We can download the Putty.exe and PuttyGen.exe from the below URL:

<https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html>

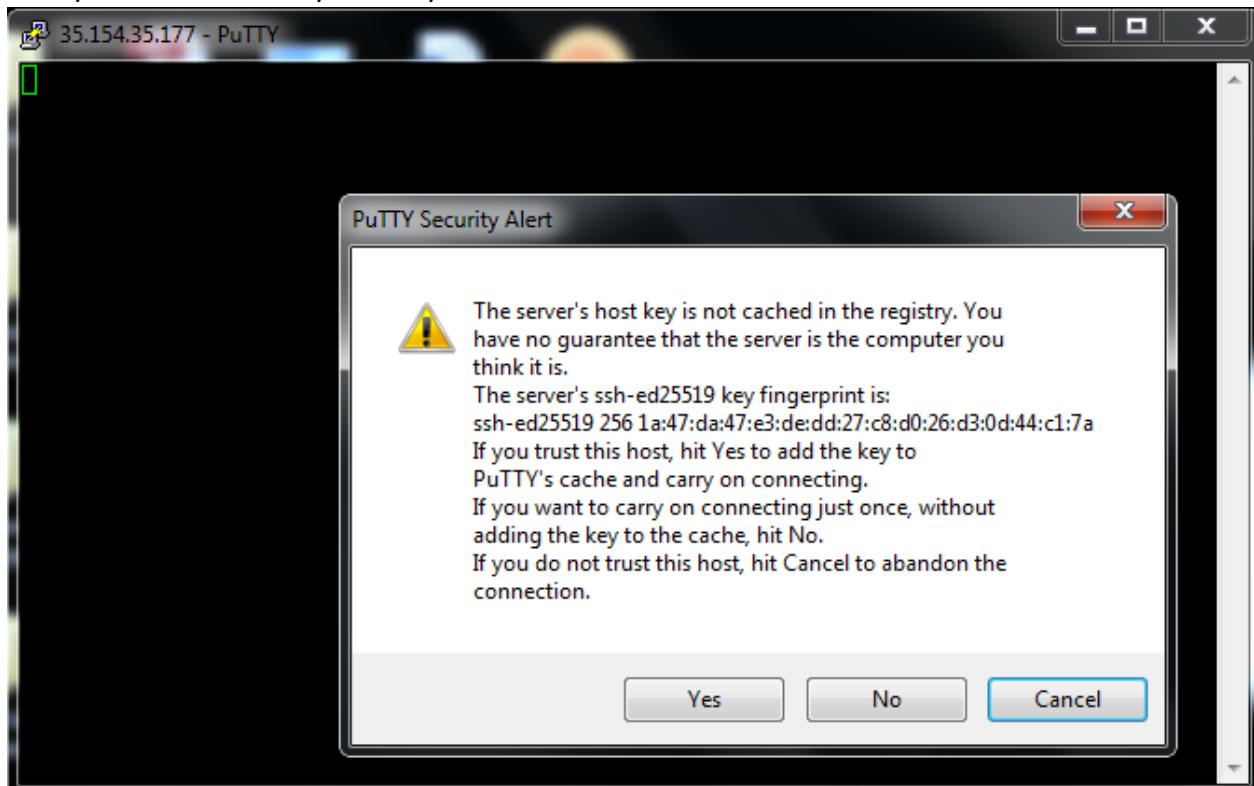
1. Download and install the latest copy of Putty and PuttyGen on local computer.
2. Launch PuttyGen and select the Load button and browse the downloaded Pem file (Which is created at the time of Instance launch).



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.



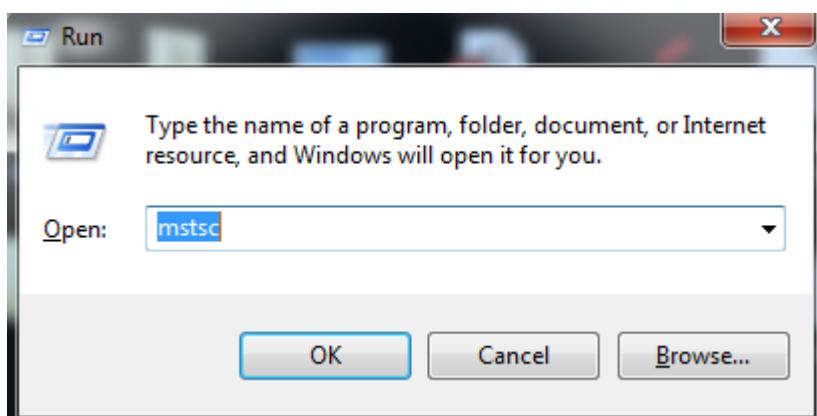
9. In the Putty terminal window, provide the user name for your Amazon Linux instance (ec2-user) and hit the *Enter* key. Now we have connected to our first instance and it is ready for use
10. Each Linux instance type launches with a default Linux system user account. For Amazon Linux, the user name 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

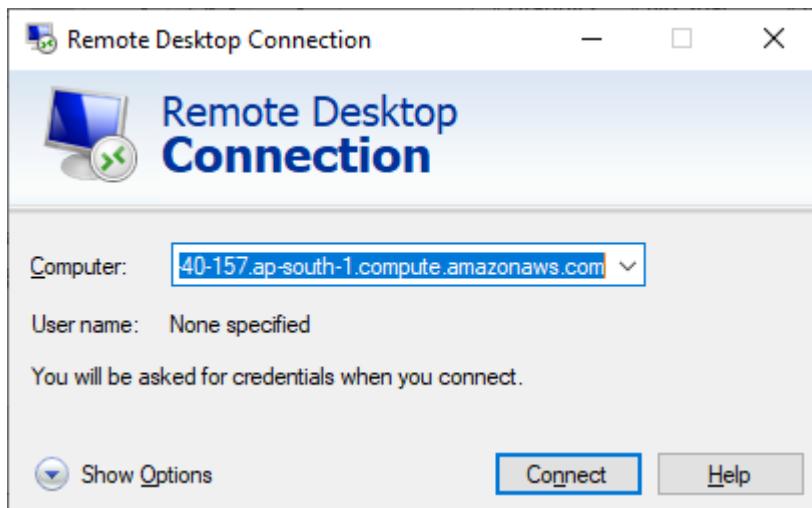
```
ec2-user@ip-172-31-0-219:~  
login as: ec2-user  
Authenticating with public key "imported-openssh-key"  
  
_ _ | _ )  
_ | ( _ / Amazon Linux AMI  
_ \_ | _ |  
  
https://aws.amazon.com/amazon-linux-ami/2017.09-release-notes/  
1 package(s) needed for security, out of 6 available  
Run "sudo yum update" to apply all updates.  
[ec2-user@ip-172-31-0-219 ~]$
```

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 Instance** 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.

Instances (1/2)		Info	<input type="button" value="C"/>	<input style="background-color: yellow; border: 1px solid black; color: black; font-weight: bold; font-size: 1em; padding: 2px 10px;" type="button" value="Connect"/>	<input type="button" value="Instance state ▾"/>
<input type="text"/> Filter instances					
	Name	Instance ID	Instance state	Instance type	Status
<input checked="" type="checkbox"/>	Windows Server	i-09c8420d382231445	Running	t2.micro	-
<input type="checkbox"/>	First Server	i-0881f8a6d9c1c26da	Terminated	t2.micro	-

17 Select the instance which you want to get the UN & PWD. Click on “Connect”, then choose “RDP Client” click on “Get Password” browse the PEM file and select “Decrypt Password” button.

Connect to instance Info

Connect to your instance i-09c8420d382231445 (Windows Server) using any of these options

Session Manager

RDP client

You can connect to your Windows instance using a remote desktop client of your choice, and by downloading and running the RDP shortcut file below:

[Download remote desktop file](#)

When prompted, connect to your instance using the following details:

Public DNS

ec2-13-127-40-157.ap-south-
1.compute.amazonaws.com

User name

Administrator

Password

[Get password](#)**Key pair associated with this instance**

WebServerKP

Browse to your key pair:

[Browse](#) WebServerKP.pem

1.704KB

Or copy and paste the contents of the key pair below:

```
-----BEGIN RSA PRIVATE KEY-----  
MIIEpAIBAAKCAQEjQa5CTLO9aN0zL8DA1NML+9ISNNX7E5rBOYjVSK9h8tSg/iO  
jU8bU8Dr34sT2hrXpOiMvzCzcwTFfmBUAoCjLkapsoD8uobiHX10Sc+4JyBaUqzc  
n231Hfod1WJzXOTGnmACT98yYZKhKbTGhIdJTRtEOq+GmS04TVjbTxrHQ+zpXXFs  
qZbYJcDqZCKC1/FnzIC/JPx1hKC4WPYFuUtyRGclqatz+MqPxQWFZdxwBhVztRve  
XM2miGCq4voua1dUsfrXaa/31rVDRCC0rJJlfx9pZCc53R95UICLB9uQ5omOsuJ  
iLauxgngeCejFYOH2vSEyoGNjg/+TkmlIA2mmQIDAQABAoIBAQCBjxuPB856fKBh  
OVpbQ7tKjnROY1rzDVfkOEldfX0BtBkqhL7xsys31pTb2blzNhXGup4TR3qUwF1v
```

[Cancel](#)[Decrypt Password](#)

Session Manager | **RDP client**

You can connect to your Windows instance using a remote desktop client of your choice, and by downloading and running the RDP shortcut file below:

[Download remote desktop file](#)

When prompted, connect to your instance using the following details:

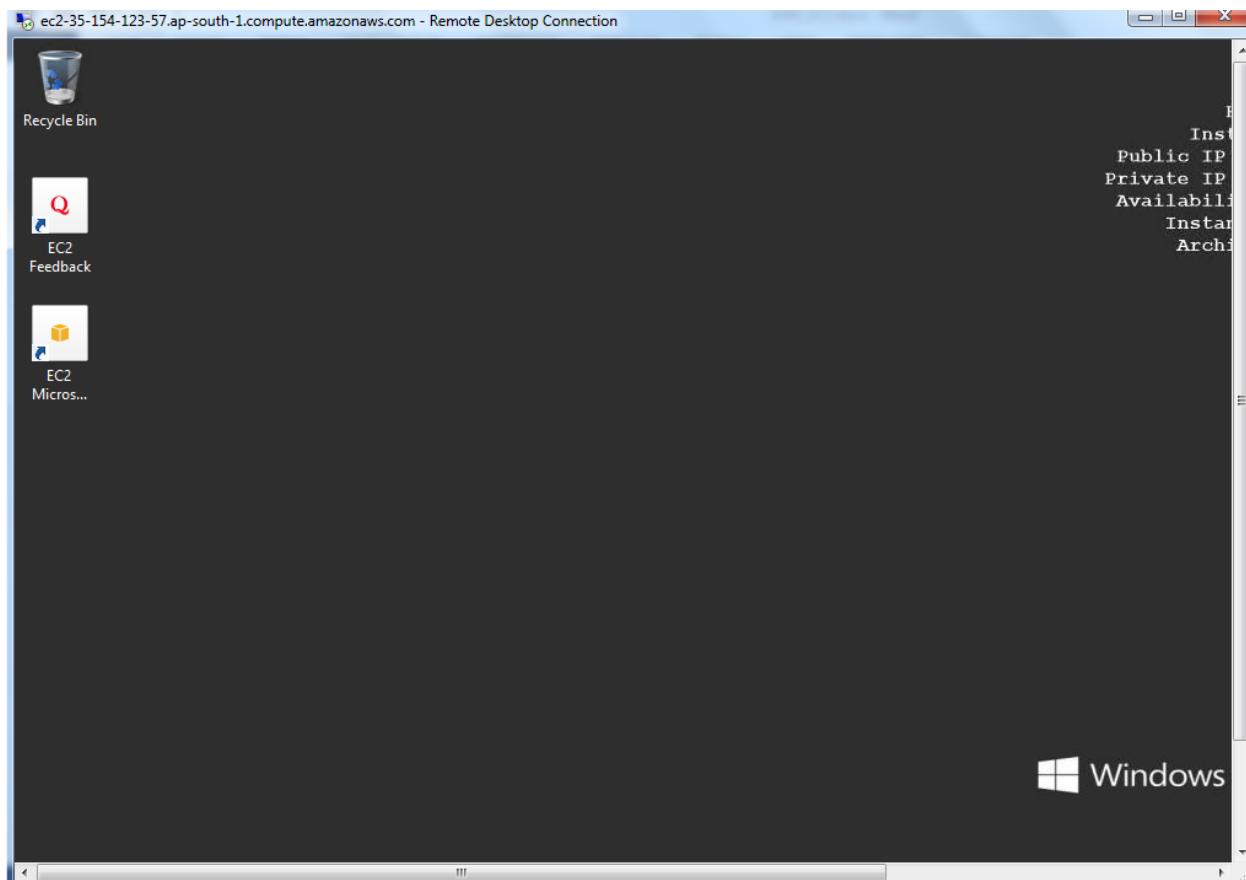
Public DNS	User name
<input type="checkbox"/> ec2-13-127-40-157.ap-south-1.compute.amazonaws.com	<input type="checkbox"/> Administrator
Password	
<input type="checkbox"/> 8jP9RWaou(w4HDVymF3Y7kx@yjUiylVz	

If you've joined your instance to a directory, you can use your directory credentials to connect to your instance.

- 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.

Create security group Info

A security group acts as a virtual firewall for your instance to control inbound and outbound traffic. To create a new security group, complete the fields below.

Basic details

Security group name Info
 Name cannot be edited after creation.

Description Info

VPC Info

Inbound rules Info

Type <small>Info</small>	Protocol <small>Info</small>	Port range <small>Info</small>	Source <small>Info</small>	Description - optional <small>Info</small>
SSH	TCP	22	Custom ▾	<input type="text" value="0.0.0.0/0"/> X Delete
HTTP	TCP	80	Custom ▾	<input type="text" value="0.0.0.0/0"/> X Delete
HTTPS	TCP	443	Custom ▾	<input type="text" value="0.0.0.0/0"/> X Delete

Add rule

Outbound rules [Info](#)

Type Info	Protocol Info	Port range Info	Destination Info	Description - optional Info
All traffic ▼	All	All	Custom ▼	<input type="text"/> 0.0.0.0/0 X
Delete				

[Add rule](#)

Tags - optional
A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

No tags associated with the resource.

[Add new tag](#)
You can add up to 50 more tag

[Cancel](#) [Create security group](#)

You can select the Protocol Type in 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 an 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. AWS have SSD-based volumes and HDD-based volumes.

Types of Amazon EBS Volumes

Amazon EBS provides the following volume types:

- General Purpose SSD (gp2)
- General Purpose SSD (gp3)
- Provisioned IOPS SSD (io1)
- Provisioned IOPS SSD (io2)
- Provisioned IOPS SSD (io2 Block Express) (in preview- July'21)
- 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.

General Purpose SSD (gp3): Amazon gp3 volumes are the latest generation of gp2 that enable customers to provision performance independent of storage capacity, while providing up to 20% lower pricing per GB than existing gp2 volumes.

A gp3 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.

Provisioned IOPS SSD (io2): io2 is the latest generation of the Provisioned IOPS SSD volumes that is designed to provide 100X durability of 99.999% as well as a 10X higher IOPS to storage ratio of 500 IOPS for every provisioned GB –at the same price as the previous generation (io1). io2 is a high performance EBS storage option designed for business-critical, I/O intensive database applications, such as SAP HANA, Oracle, Microsoft SQL Server, and IBM DB2 that have high durability requirements.

To achieve the limit of 64,000 IOPS and 1,000 MB/s throughput, the volume must be attached to an EC2 instance built on the AWS Nitro System.

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 125 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 125 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.

AWS Recently introduced Multi Attach volume feature.

Amazon EBS Multi-Attach enables you to attach a single Provisioned IOPS SSD (io1 or io2) volume to multiple instances that are in the same Availability Zone.

- Multi-Attach enabled volumes can be attached to up to 16 Linux instances built on the Nitro System that are in the same Availability Zone.
- Multi-Attach is supported exclusively on Provisioned IOPS SSD volumes.
- Multi-Attach for io2 volumes is available in all Regions that support io2 volumes. Multi-Attach for io1 volumes is available in the following Regions only: us-east-1, us-west-2, eu-west-1, and ap-northeast-2.
- Multi-Attach enabled volumes can't be created as boot volumes.
- Multi-Attach enabled volumes can be attached to one block device mapping per instance.
- Multi-Attach can't be enabled during instance launch using either the Amazon EC2 console or RunInstances API.
- The following table shows volume modification support for Multi-Attach enabled io1 and io2 volumes after creation. (* We can't enable or disable Multi-Attach while the volume is attached to an instance.)

	io2 volumes	io1 volumes
Modify volume type	X	X
Modify volume size	✓	X
Modify provisioned IOPS	✓	X
Enable Multi-Attach	✓ *	X
Disable Multi-Attach	✓ *	X

Solid State Drives (SSD)					
Volume Type	EBS Provisioned IOPS SSD (io2 Block Express)	EBS Provisioned IOPS SSD (io2)	EBS Provisioned IOPS SSD (io1)	EBS General Purpose SSD (gp3)	EBS General Purpose SSD (gp2)*
Short Description	Highest performance SSD volume designed for business-critical latency-sensitive transactional workloads	Highest performance and highest durability SSD volume designed for latency-sensitive transactional workloads	Highest performance SSD volume designed for latency-sensitive transactional workloads	Lowest cost SSD volume that balances price performance for a wide variety of transactional workloads	General Purpose SSD volume that balances price performance for a wide variety of transactional workloads
Durability	99.999%	99.999%	99.8% - 99.9% durability	99.8% - 99.9% durability	99.8% - 99.9% durability
Use Cases	Largest, most I/O intensive, mission critical deployments of NoSQL and relational databases such as Oracle, SAP HANA, Microsoft SQL Server, and SAS Analytics	I/O-intensive NoSQL and relational databases	I/O-intensive NoSQL and relational databases	Virtual desktops, medium sized single instance databases such as Microsoft SQL Server and Oracle, latency sensitive interactive applications, boot volumes, and dev/test environments	Virtual desktops, medium sized single instance databases such as Microsoft SQL Server and Oracle, latency sensitive interactive applications, boot volumes, and dev/test environments
API Name	io2	io2	io1	gp3	gp2
Volume Size	4 GB – 64 TB	4 GB – 16 TB	4 GB - 16 TB	1 GB - 16 TB	1 GB - 16 TB
Max IOPS**/Volume	256,000	64,000	64,000	16,000	16,000

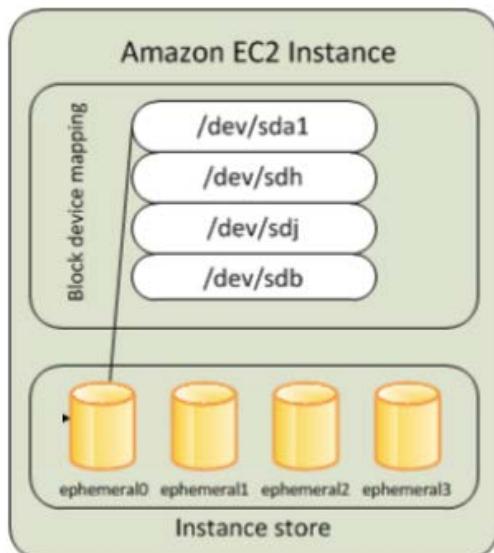
Hard Disk Drives (HDD)		
	Throughput Optimized HDD (st1)	Cold HDD (sc1)
Short Description	Low cost HDD volume designed for frequently accessed, throughput intensive workloads	Lowest cost HDD volume designed for less frequently accessed workloads
Durability	99.8% - 99.9% durability	99.8% - 99.9% durability
Use Cases	Big data, data warehouses, log processing	Colder data requiring fewer scans per day
API Name	st1	sc1
Volume Size	125 GB - 16 TB	125 GB - 16 TB
Max IOPS**/Volume	500	250
Max Throughput***/Volume	500 MB/s	250 MB/s

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 also 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.

Create Volume

The screenshot shows the 'Create Volume' wizard interface. The configuration fields are as follows:

- Volume Type:** General Purpose SSD (gp2) (with info icon)
- Size (GiB):** 100 (Min: 1 GiB, Max: 16384 GiB) (with info icon)
- IOPS:** 300 / 3000 (Baseline of 3 IOPS per GiB with a minimum of 100 IOPS, burstable to 3000 IOPS) (with info icon)
- Throughput (MB/s):** Not applicable (with info icon)
- Availability Zone*:** ap-south-1a (with info icon)
- Snapshot ID:** Select a snapshot (with cancel and info icons)
- Encryption:** Encrypt this volume

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.

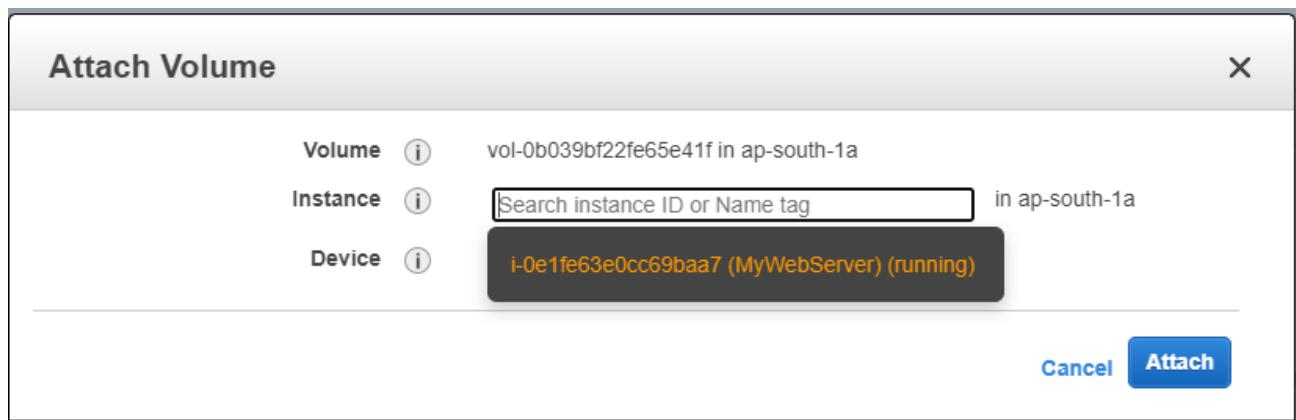
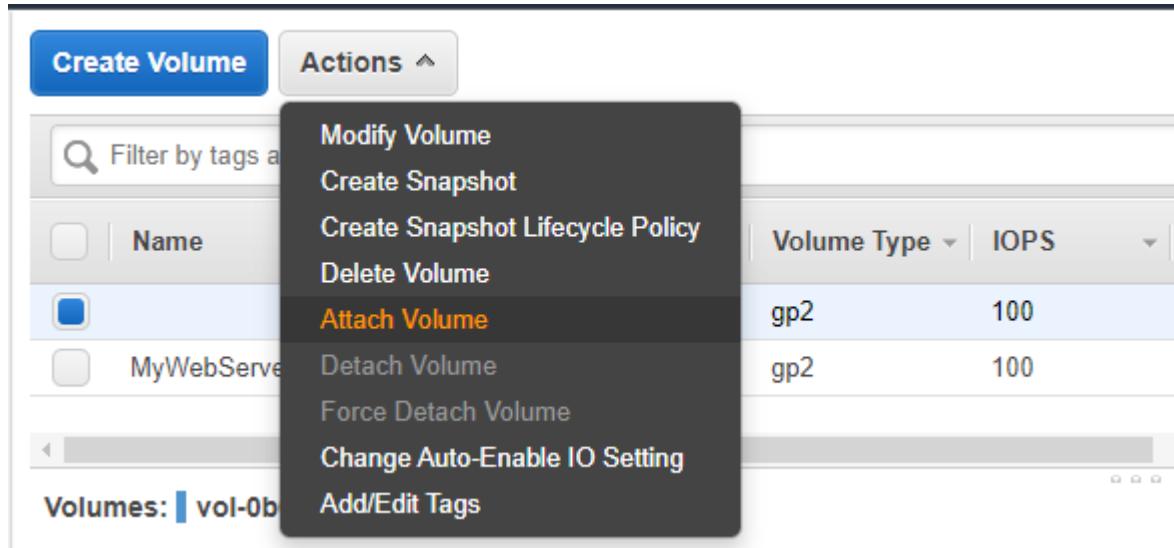
Encryption: We can choose whether or not to encrypt EBS Volume. Select Encrypt this volume checkbox if you wish to do so.

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 the 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 through 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 filesystem  Linux
  128      2048        4095     1M  BIOS boot partition  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.

If you wish to use ext4 file system : mkfs -t ext4 /dev/xvdf

If you wish to use xfs file system : mkfs -t xfs /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

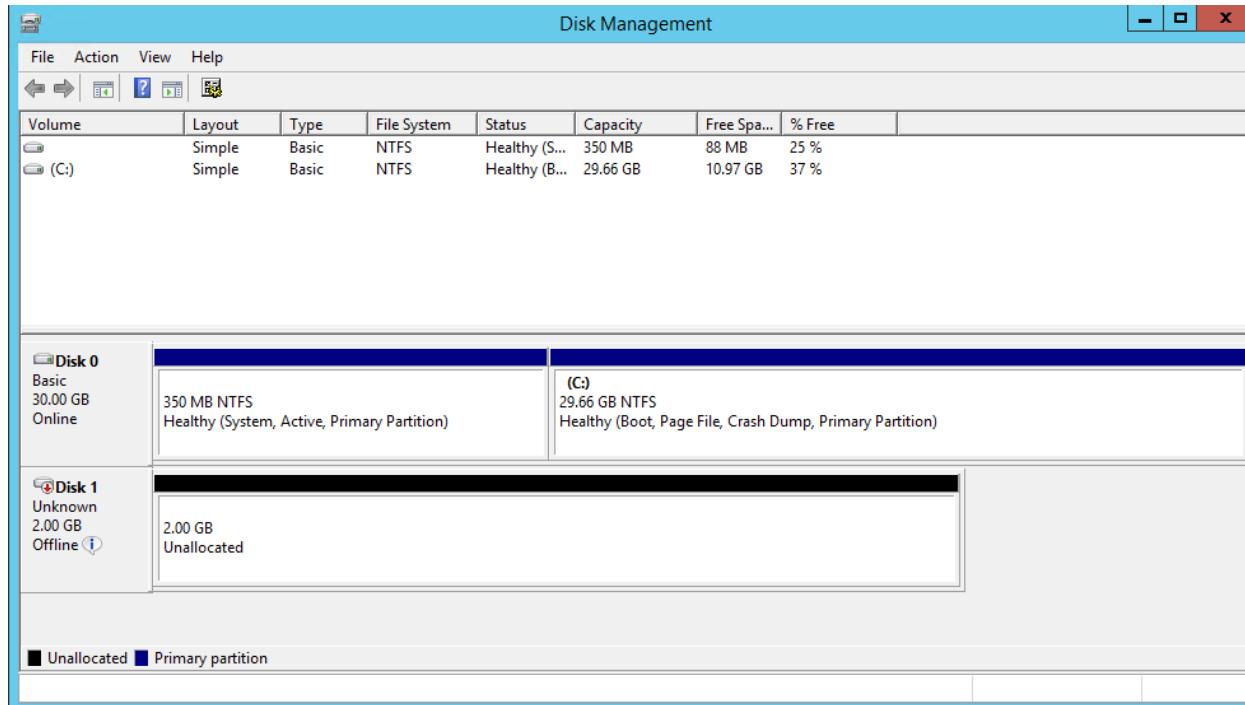
```
[root@ip-172-31-7-51 ~]# mkdir /newvolume
[root@ip-172-31-7-51 ~]# mount /dev/xvdf /newvolume
[root@ip-172-31-7-51 ~]# 
```

6. Now the volume is available for the use.

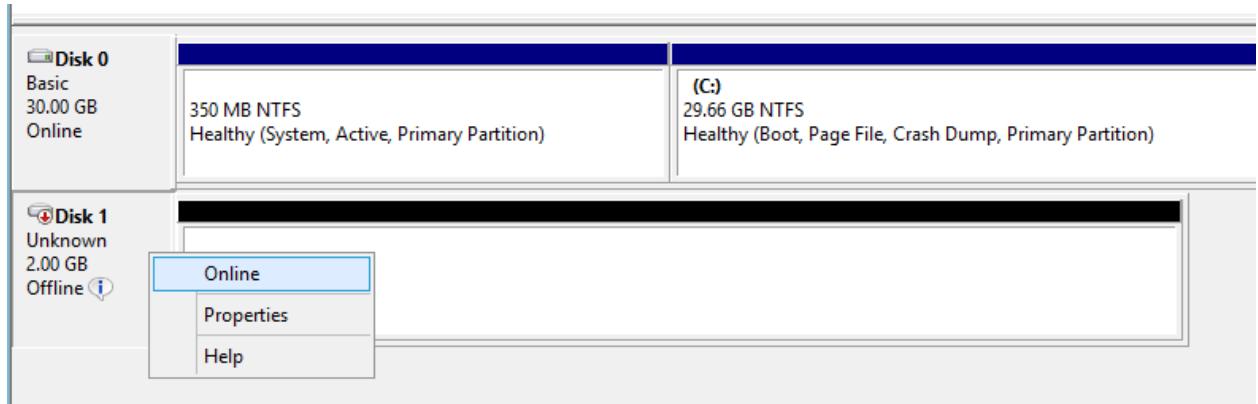
-
7. This is a temporary mount, for permanent mount, make sure you enter the volume information in “/etc/fstab” document.

For Windows Instances:

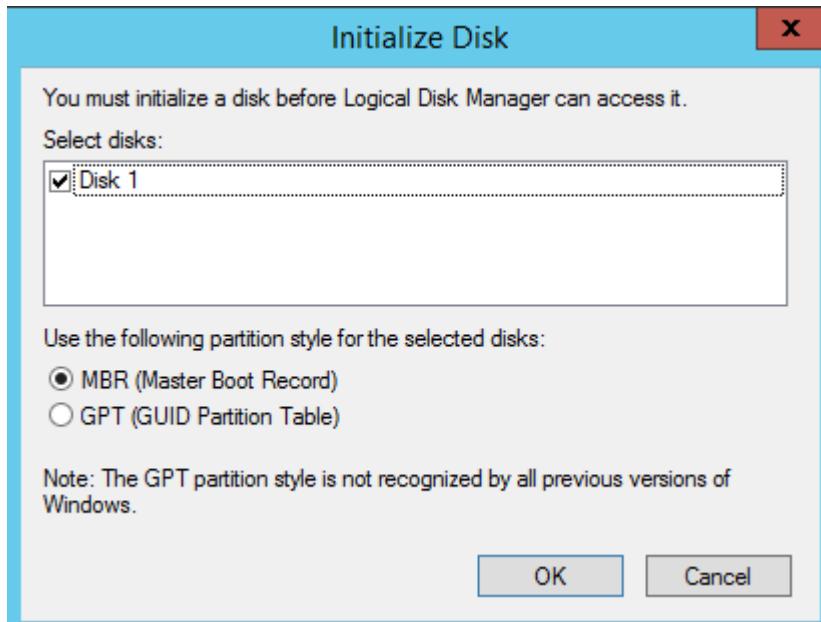
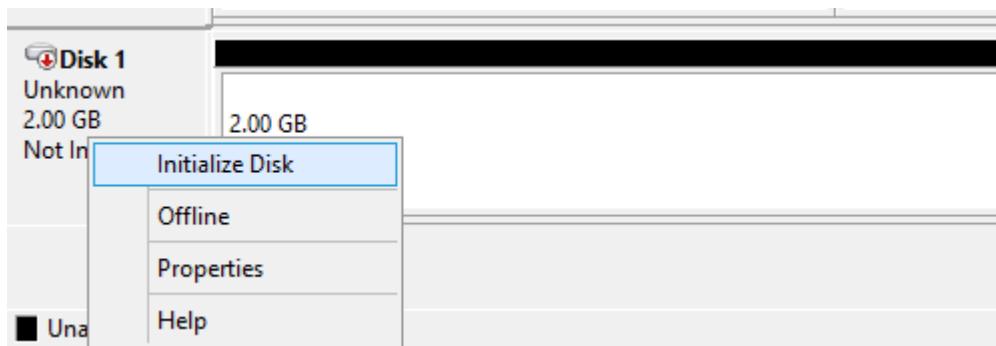
1. Attach the volume to the windows instance same as previous step.
2. Login to the windows instance and open Disk management console.
3. Open Run and give **diskmgmt.msc** command to open the Disk Management.



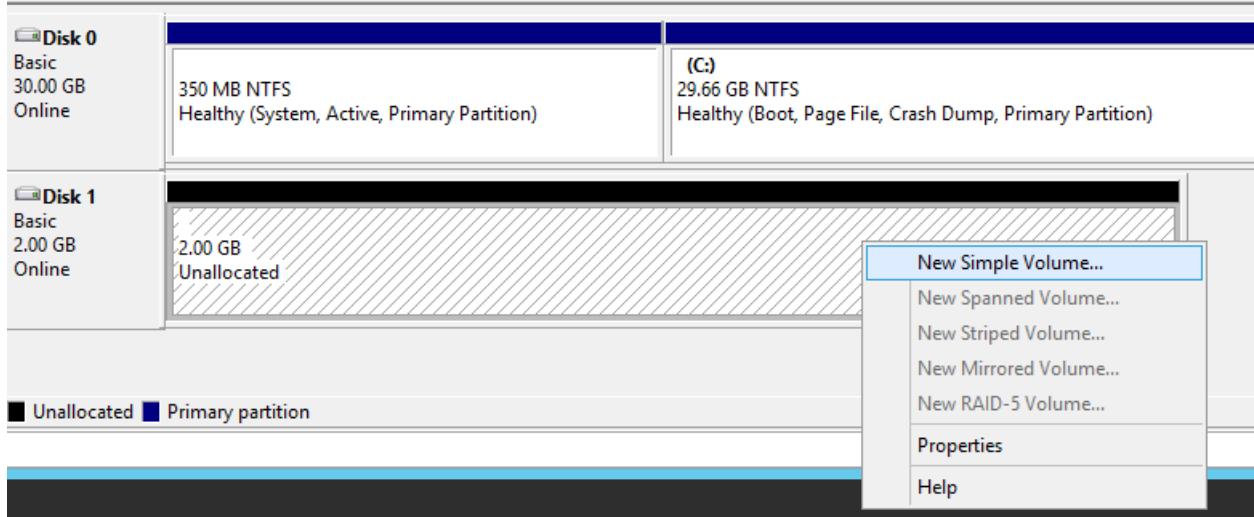
4. The newly created 2GB volume is attached to the Windows instance and by default the status of this drive will set to offline, Select the Disk 1, then choose **Online** option to make the volume online.

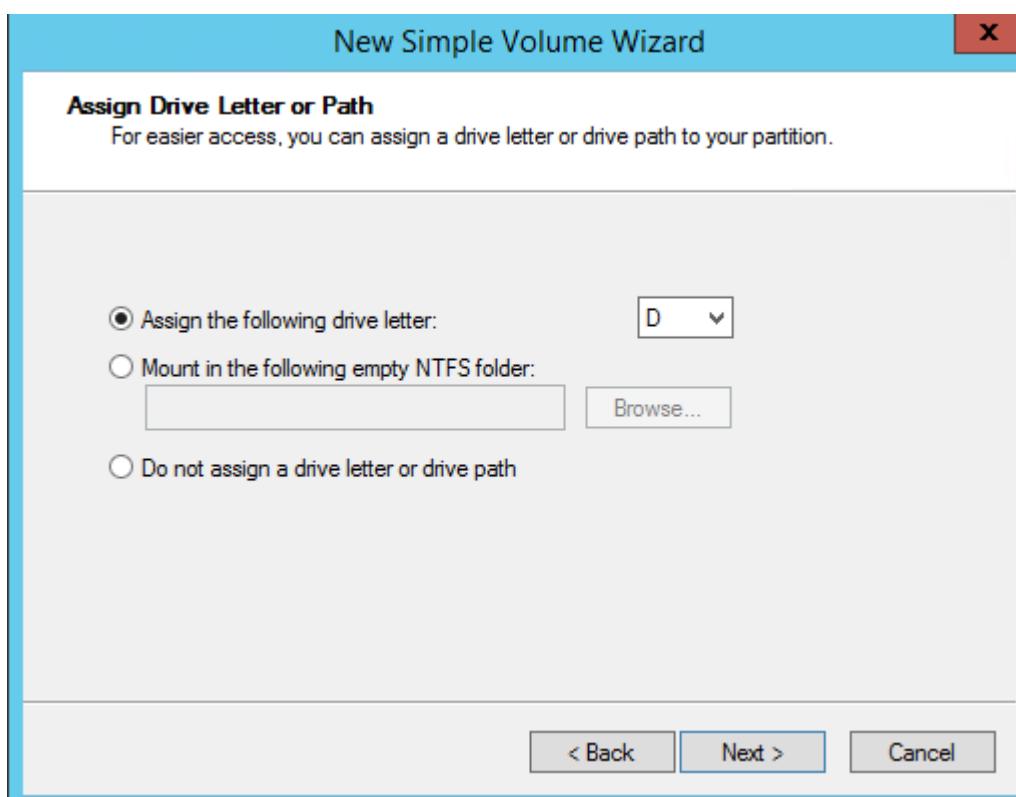
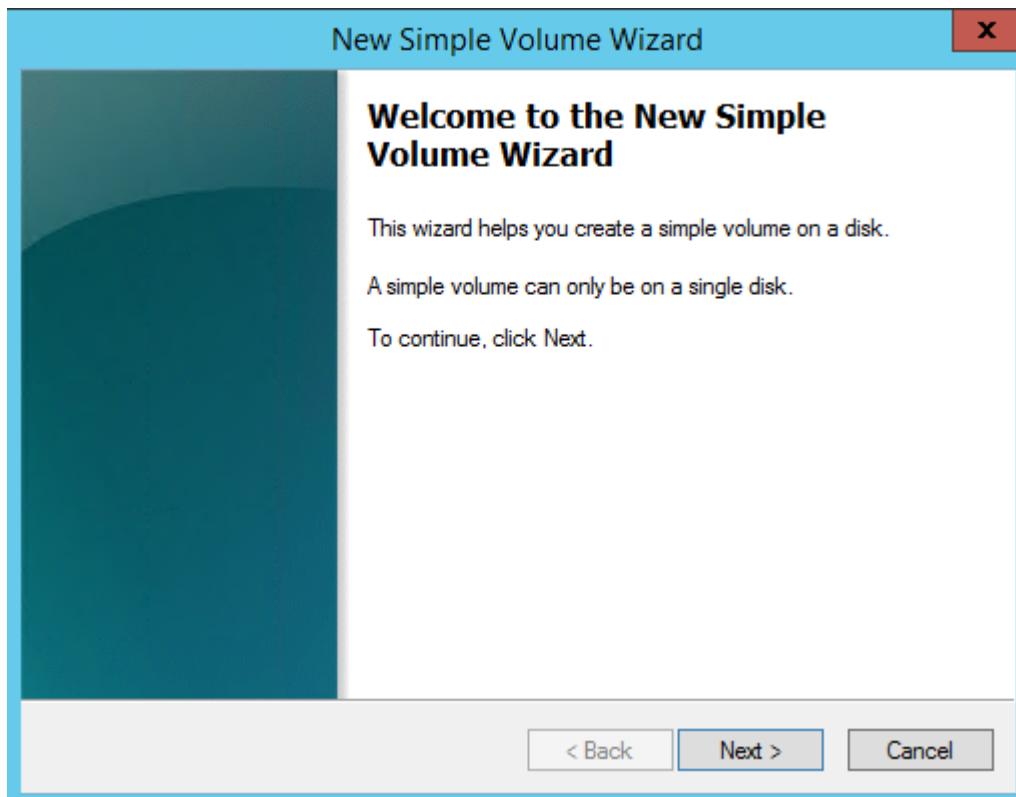


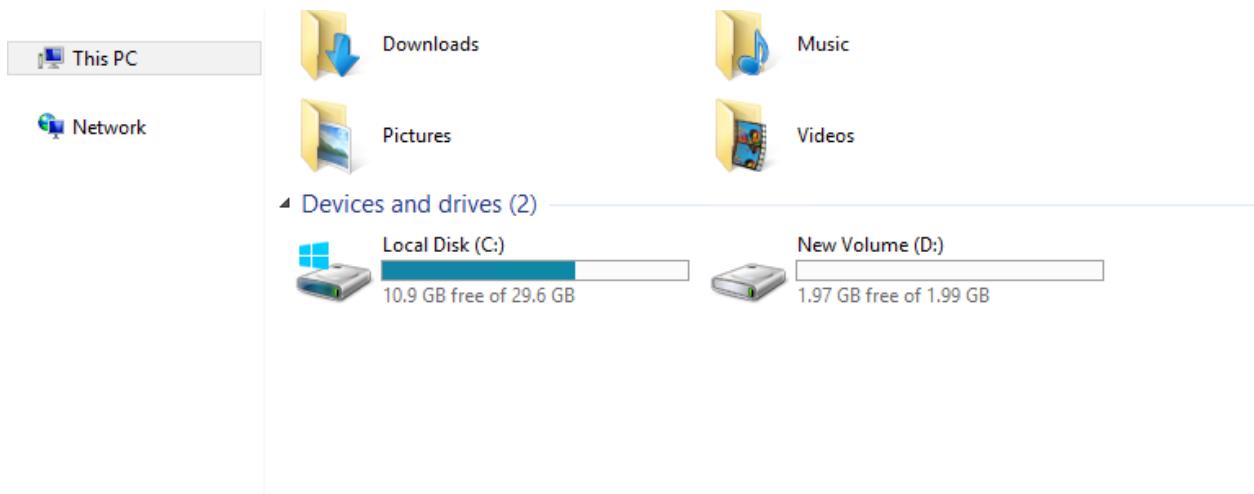
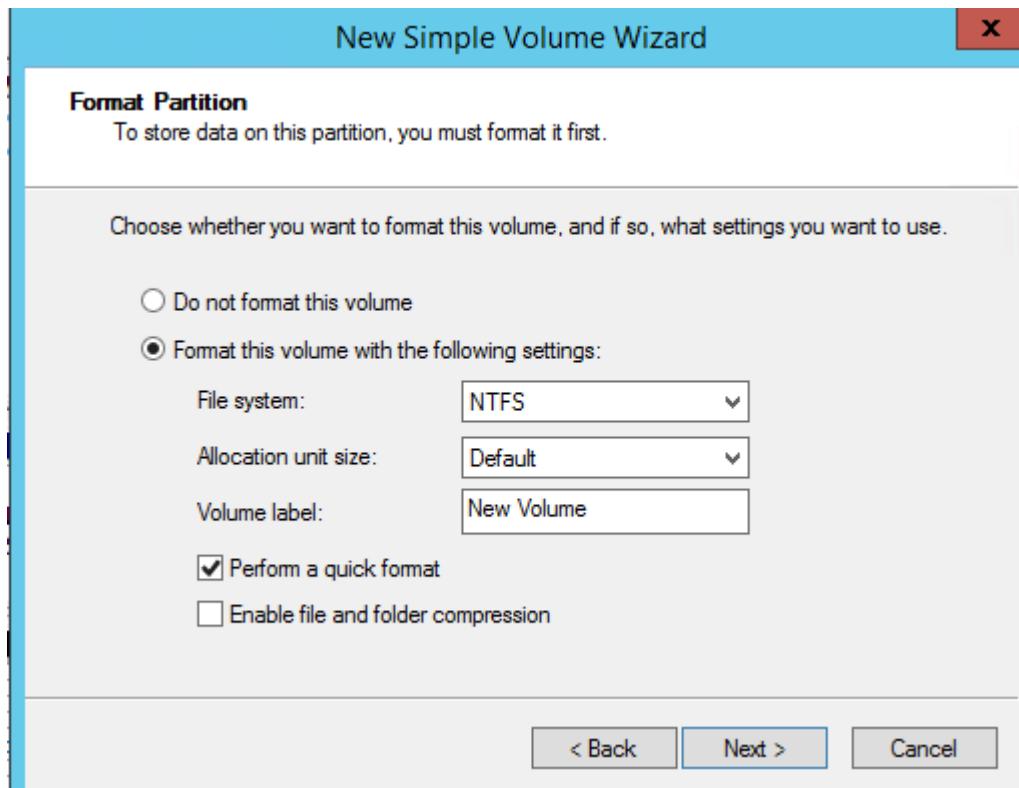
5. Here we have to initialize the Disk, Give right click on Disk then select the **initialize disk** option and click on **OK**



- 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







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.

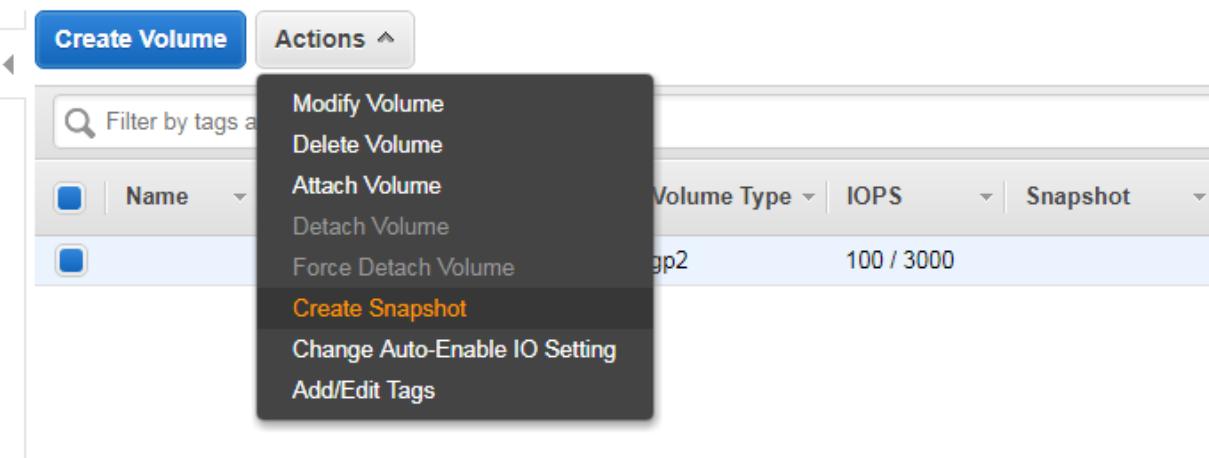
Backup of EBS volumes

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**.

A screenshot of the 'Create Snapshot' dialog box. It has fields for 'Volume' (set to vol-06c57b4fc140f49da), 'Name' (an empty input field), 'Description' (an empty input field), and 'Encrypted' (set to 'No'). At the bottom right are 'Cancel' and 'Create' buttons.

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

The screenshot shows the AWS EBS Snapshot Management interface. At the top, there are buttons for "Create Snapshot" and "Actions". A dropdown menu is open under "Actions" with the following options: Delete, Create Volume, Create Image, Copy, Modify Permissions, and Add/Edit Tags. The main area displays a table of snapshots. One row is selected, showing details: Name: my-snapshot, Snapshot ID: snap-01ad22b9bd5..., Size: 2 GiB, Description: my-snapshot, Status: completed (green dot), Started: October 30, 2018, Progress: available (100%), and Encrypted: Not Encrypted.

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.

Amazon Data Lifecycle Manager : Amazon Data Lifecycle Manager provides a simple, automated way to back up data stored on Amazon EBS volumes. We can use Amazon Data Life Cycle Manager (Amazon DLM) lifecycle policies to automate the creation, retention, and deletion of Amazon EBS snapshots.

Choose policy type as “**EBS Snapshot Policy**” as shown image and click “**Next Step**”

We can Create Snapshot Lifecycle Policy that applies to the resources within the selected account for the selected AWS Region.

DLM Uses the Tags to choose/find the resources. (Tag is combination of Key and value), so adding valid tags to the resources is very important in real environments.



In Below image I have selected Target resource type as “**Volume**” and all the volumes having tags (**Key = Backup, value = Schedule 1**) will comes under this backup job.

Specify settings

Target resources Info

Specify the resources that are to be targeted by this policy.

Target resource types
Select the type of resources that are to be targeted.

Volume
 Instance

Target resource tags
Only resources of the selected type that have these tags will be targeted.

Enter a key Enter a value

Backup
Schedule1

44 tags remaining of 45.

Provide a meaningful description for this DLM policy and associate a valid role (Choose Default role)

Description

Policy description

IAM role Info

This policy must be associated with an IAM role that has the appropriate permissions. If you choose to create a new role, you must grant relevant role permissions and set up trust relationships correctly. If you are unsure of what role to use, choose Default role.

Default role

(i) If the default role does not exist in your account, it will be automatically created with all of the required permissions. [View default role permissions](#)

Choose another role

Set policy status as “Enabled” to take this policy effect immediate after the creation.

Policy status

Specify whether to enable the policy immediately after creation or modification. If you do not enable the policy now, then it will not begin creating snapshots or AMIs until you manually set its activation status to enabled.

Enabled

Not enabled

[Cancel](#) [Next](#)

In next step, we need to Configure schedule. Schedules define how often the policy runs and the specific actions that are to be performed. Every policy must have at least one schedule, and we can add maximum of 3 schedules.

As per the below image, Starting from 09:00 UTC DLM will create a backup copy for every 12 hrs and it will retain the most recently created 7 snapshots. We can define snapshots in age wise also.

Schedule details [Info](#)

Schedule name
Schedule 1

Frequency
Daily ▾

Every
12 hours ▾

Starting at
09:00 UTC

Retention type
Count ▾

Keep
7 Snapshots

Fast snapshot restore : Fast snapshot restore enables you to restore a volume from a snapshot that is fully initialized at creation and that instantly delivers all of its provisioned performance.

Cross-region copy : Cross region copy feature allow us to copy the snapshots to desired/selected region. We can choose retention period for the copies snapshot also.

Cross-account sharing : Cross-account sharing allows us to share snapshots created by this schedule with up to 50 AWS accounts. To share snapshots, specify the AWS account IDs of the accounts.

Advanced settings - *optional*

► Tagging [Info](#)

▼ Fast snapshot restore [Info](#)
Enable fast snapshot restore to ensure that volumes created from snapshots created by this schedule instantly deliver all of their provisioned performance.

Enable fast snapshot restore for snapshots created by this schedule

▼ Cross-Region copy [Info](#)
Enable cross-Region copy to copy snapshots created by this schedule to up to three additional Regions.

Enable cross-Region copy for this schedule

▼ Cross-account sharing [Info](#)
Enable cross-account sharing to share the snapshots created by this schedule with other AWS accounts.

Enable cross-account sharing for this schedule

Review all the settings and Create Policy. Once policy is choose the Schedules option and it will show the Schedule details as below image.

Policy: policy-0f421176efa00e983

Details | **Schedules** | Tags

▼ Schedule 1

Schedule details

Frequency	Retain rule
Every 12 hours starting at 09:00 UTC.	A maximum of 7 will be retained. The oldest snapshot retained will be <= 3.5 days old.

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 EC2 instance, Actions, “**Images and templates**” then select **Create Image** option.

The screenshot shows the AWS EC2 Instances page. A single instance, **i-0e1fe63e0cc69baa7 (MyWebServer)**, is listed as **Running** with an **t2.micro** instance type. The **Actions** menu is open, with the **Create image** option highlighted. Other options in the menu include **Create template from instance** and **Launch more like this**.

Create image Info

An image (also referred to as an AMI) defines the programs and settings that are applied when you launch an EC2 instance. You can create an image from an existing instance or launch more instances like this one.

Instance ID
 i-0e1fe63e0cc69baa7 (MyWebServer)

Image name
 WebServer-GoldenAMI
Maximum 127 characters. Can't be modified after creation.

Image description - optional
 Web Server GAM
Maximum 255 characters

No reboot
 Enable

Instance volumes

Volume type	Device	Snapshot	Size	Volume type	IOPS	Throughput	Delete on termination	Encrypted
EBS	/dev/x...	Create new snapshot fr...	8	EBS General Purpose SS...	100		<input checked="" type="checkbox"/> Enable	<input type="checkbox"/> Enable

Add volume

During the image creation process, Amazon EC2 creates a snapshot of each of the above volumes.

Tags - optional
A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Tag image and snapshots together
Tag the image and the snapshots with the same tag.

Tag image and snapshots separately
Tag the image and the snapshots with different tags.

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

Description: Provide a suitable description for your new AMI.

No reboot: Amazon EC2 shuts down the instance, takes snapshots of any attached volumes, creates and registers the AMI, and then reboots the instance. Select No reboot to avoid having your instance shut down.

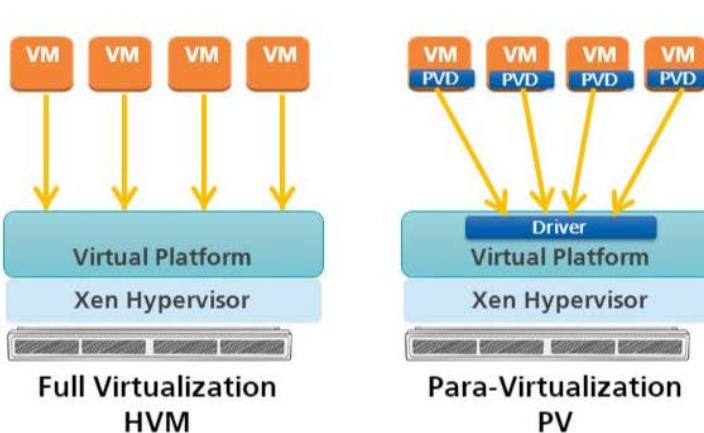
Tags: We can tag the AMI and the snapshots with the same tags, or we can tag them with different tags.

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 paravirtualized domain, 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.

Compute Isolation:

Paravirtual (PV) and Hardware Virtual Machine (HVM)



- Why HVM?
 - Ability to use hardware extensions
 - Faster access to hardware
 - Why PV?
 - OS requires modification
 - Performance penalty since there is no direct access to hardware

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

The screenshot shows the AWS EC2 AMI Management console. At the top, there are buttons for 'Launch', 'EC2 Image Builder', and 'Actions'. A dropdown menu 'Owned by me' is open. A search bar says 'Filter by tags and attributes or search by keyword'. Below the search bar is a table header with columns: Name, AMI Name, AMI ID, Source, Owner, Visibility, Status, and Creation Date. A single row is visible in the table, representing an AMI named 'WebServer-GoldenAMI' with the ID 'ami-0e054b11834251a74'. The status is 'available'. Below the table, it says 'Image: ami-0e054b11834251a74'. At the bottom, there are tabs for 'Details', 'Permissions', and 'Tags', with 'Details' being the active tab. The 'Details' section contains two tables of information.

AMI ID	ami-0e054b11834251a74	AMI Name	WebServer-GoldenAMI
Owner	501170964283	Source	501170964283/WebServer-GoldenAMI
Status	available	State Reason	-
Creation date	June 16, 2021 at 6:52:18 PM UTC+5:30	Platform details	Linux/UNIX
Architecture	x86_64	Usage operation	RunInstances
Image Type	machine	Virtualization type	hvm
Description	Web Server GAMI	Root Device Name	/dev/xvda
Root Device Type	ebs	RAM disk ID	-

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**.
- Delete the backend snapshots used for this AMI.

Elastic Load Balancing

The Elastic Load Balancing service allows you to distribute traffic across a group of Amazon EC2 instances enabling 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.

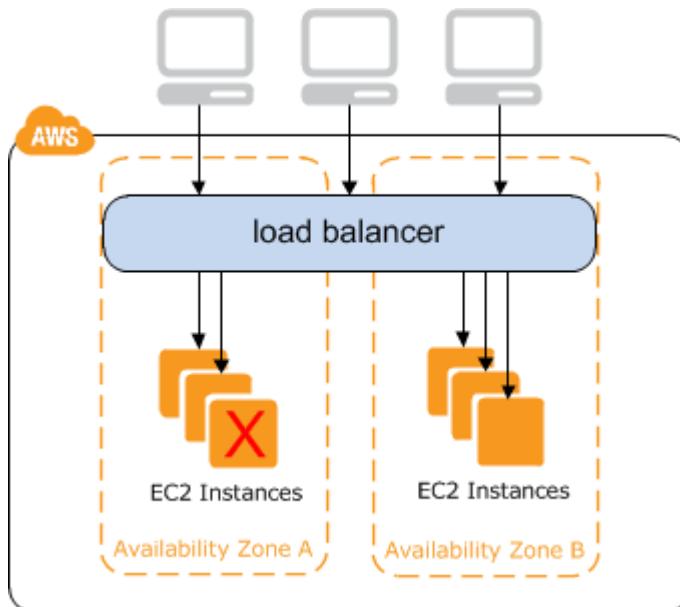
We have four types of load balancers available with AWS.

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

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

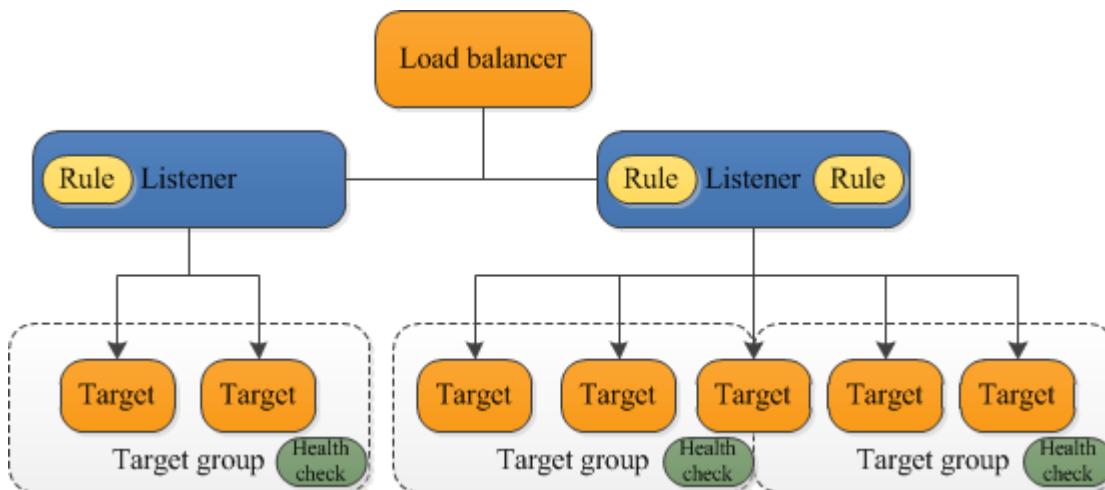
Classic Load Balancer:

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.



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.

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.

Gateway Load Balancer : Gateway Load Balancer makes it easy to deploy, scale, and manage third-party virtual appliances. It gives one gateway for distributing traffic across multiple virtual appliances, while scaling them up, or down, based on demand.

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.

Load Balancer Comparison chart

Feature	Application Load Balancer	Network Load Balancer	Gateway Load Balancer	Classic Load Balancer
Load Balancer type	Layer 7	Layer 4	Layer 3 Gateway + Layer 4 Load Balancing	Layer 4/7
Target type	IP, Instance, Lambda	IP, Instance	IP, Instance	
Terminates flow/proxy behavior	Yes	Yes	No	Yes
Protocol listeners	HTTP, HTTPS, gRPC	TCP, UDP, TLS	IP	TCP, SSL/TLS, HTTP, HTTPS

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.

The screenshot shows the AWS CloudFormation console. On the left, there's a navigation sidebar with sections like 'Snapshots', 'Lifecycle Manager', and expanded sections for 'Network & Security' and 'Load Balancing'. Under 'Load Balancing', 'Load Balancers' is highlighted. The main area has a 'Create Load Balancer' button at the top. Below it is a search bar and a table listing existing resources. One row in the table is selected, showing 'Name: learnaws' and 'DNS name: learnaws-1760444141.us-east-1.elb.amazonaws.com'.

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. Make sure you have selected 2 subnets for high availability.
3. Choose Internet-facing as we want to get/test the output from outside worlds.
4. 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.

Step 1: Configure Load Balancer

Basic Configuration

To configure your load balancer, provide a name, select a scheme, specify one or more listeners, and select a network listener that receives HTTP traffic on port 80.

Name i	<input type="text" value="WebserverELB"/>
Scheme i	<input checked="" type="radio"/> internet-facing <input type="radio"/> internal
IP address type i	<input type="text" value="ipv4"/>

Listeners

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

Load Balancer Protocol	Load Balancer Port
HTTP	80
<input type="button" value="Add listener"/>	

Availability Zones

Specify the Availability Zones to enable for your load balancer. The load balancer routes traffic to the targets in these Availability Zones only. Subnets from at least two Availability Zones to increase the availability of your load balancer.

VPC	vpc-a15391ca (172.31.0.0/16) (default)
Availability Zones	<input checked="" type="checkbox"/> ap-south-1a subnet-7b3ed510 <input checked="" type="checkbox"/> ap-south-1b subnet-5e451d12 <input type="checkbox"/> ap-south-1c subnet-caddbeb1
	IPv4 address Assigned by AWS

- In Step 2, 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.

1. Configure Load Balancer 2. Configure Security Settings 3. Configure Security Groups 4. Configure Routing 5. Register Targets 6. Review

Step 2: Configure Security Settings

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

- We have to select the Security group for ELB, Make sure your ELB is opened with port 80 for everyone.

Step 3: Configure Security Groups

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

Assign a security group	<input checked="" type="radio"/> Create a new security group <input type="radio"/> Select an existing security group		
Security group name	LoadBalancerSG		
Description	LoadBalancerSG		
Type	Protocol	Port Range	Source
HTTP	TCP	80	Custom 0.0.0.0, ::/0
HTTPS	TCP	443	Custom 0.0.0.0, ::/0
<input type="button" value="Add Rule"/>			

- In Step 4 we have to configurerouting, Give a Target Group name and choose target type as "Instance / IP" and go with http and port 80.

Target group

Target group (i)	<input type="text" value="New target group"/> ▼
Name (i)	<input type="text" value="AVltg"/>
Target type	<input checked="" type="radio"/> Instance <input type="radio"/> IP <input type="radio"/> Lambda function
Protocol (i)	<input type="text" value="HTTP"/> ▼
Port (i)	<input type="text" value="80"/>
Protocol version (i)	<input checked="" type="radio"/> HTTP1 <small>Send requests to targets using HTTP/1.1. Supported when the request protocol is HTTP/1.1 or HTTP/2.</small>

Health checks

Protocol (i)	<input type="text" value="HTTP"/> ▼
Path (i)	<input type="text" value="/index.html"/>

▼ Advanced health check settings

Port (i)	<input checked="" type="radio"/> traffic port <input type="radio"/> override
Healthy threshold (i)	<input type="text" value="5"/>
Unhealthy threshold (i)	<input type="text" value="2"/>
Timeout (i)	<input type="text" value="5"/> seconds
Interval (i)	<input type="text" value="30"/> seconds
Success codes (i)	<input type="text" value="200"/>

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

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.

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.

Timeout: 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.

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.

Success Codes : We can define http success codes from 200-299

8. Step 5 – Add EC2 instances to the target group we created in previous step. Choose all the instance we want to run top of the Load Balancer and make sure you select “**Add to Registered**” then choose review option and create load balancer.

Step 5: Register Targets

The screenshot shows the AWS Lambda 'Register Targets' step. At the top, there's a 'Remove' button and a table with columns: Instance, Name, Port, State, Security groups, and Zone. One row is listed: i-0e1fe63e0cc69baa7, MyWebServer, 80, running, WebServer, ap-south-1a. Below the table, there's a section titled 'Instances' with a note about registering additional instances. A search bar says 'Add to registered on port 80'. Another table shows the instance details again: i-0e1fe63e0cc69baa7, MyWebServer, 80, running, WebServer, ap-south-1a, subnet-7b3ed510, 172.31.32.0/20. At the bottom right are 'Cancel', 'Previous', and 'Next: Review' buttons.

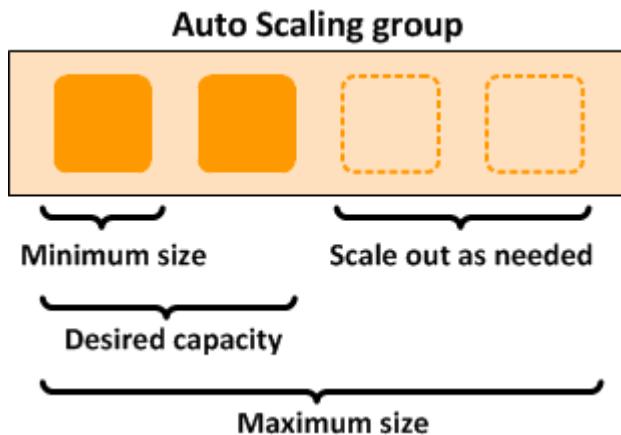
9. I have installed sample wordpress website in ec2 instance then started the httpd service and am able to get the webpage using the Instance’s public IP.
10. And Here is the details for created ELB, As we know we’ll get a DNS name for our created ELB, We can access the same webpage by using the ELB’s DNS name also.

The screenshot shows the AWS Lambda 'Actions' tab. It has a 'Create Load Balancer' button and a 'Actions' dropdown. Below is a search bar and a table with columns: Name, DNS name, State, VPC ID, Availability Zones, and Type. Two rows are listed: 'learnaws' (Active, vpc-a15391ca, ap-south-1b, ap-south-1a, application) and 'WebserverELB' (Active, vpc-a15391ca, ap-south-1b, ap-south-1a, application). The 'WebserverELB' row is highlighted with a yellow background.

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

Auto Scaling Group (ASG)

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.

Launch Template

A launch template is similar to a launch configuration, in that it specifies instance configuration information. Included are the ID of the Amazon Machine Image (AMI), the instance type, a key pair, security groups, and the other parameters that you use to launch EC2 instances.

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 Options :

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.

- **Maintain current instance levels at all times :** You can configure your Auto Scaling group to maintain a specified number of running instances at all times. To maintain the current instance levels, Amazon EC2 Auto Scaling performs a periodic health check on running instances within an Auto Scaling group. When Amazon EC2 Auto Scaling finds an unhealthy instance, it terminates that instance and launches a new one.
- **Scheduled scaling / Scale based on a schedule:** We can scale resources based on a particular time and date. This is useful when you know exactly when to increase or decrease the number of instances in your group, simply because the need arises on a predictable schedule.

- **Scale based on demand / Dynamic scaling:** A more advanced way to scale your resources, using dynamic scaling, lets you define a scaling policy that dynamically resizes your Auto Scaling group to meet changes in demand. For example, let's say that you have a web application that currently runs on two instances and you want the CPU utilization of the Auto Scaling group to stay at around 50 percent when the load on the application changes. This method is useful for scaling in response to changing conditions, when you don't know when those conditions will change. We have 3 types of scaling policies.
 - **Target tracking scaling :** Increase or decrease the current capacity of the group based on a target value for a specific metric. This is similar to the way that your thermostat maintains the temperature of your home, you select a temperature and the thermostat does the rest. Example, We can Configure a target tracking scaling policy to keep the average aggregate CPU utilization of your Auto Scaling group at 40 percent.
 - **Step scaling :** Increase or decrease the current capacity of the group based on a set of scaling adjustments, known as step adjustments, that vary based on the size of the alarm breach.
 - **Simple scaling :** Increase or decrease the current capacity of the group based on a single scaling adjustment.

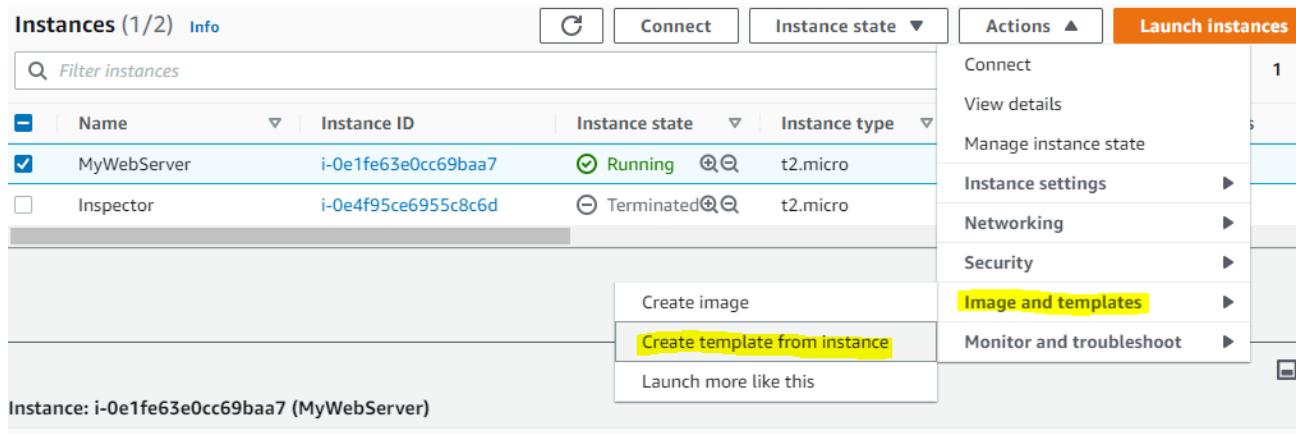
Auto scaling group creation involves with two steps. First one is Creating a Launch Configuration and second is Creating Auto Scaling group.

Amazon recently introduced “launch template” feature.

**** Create an CustomAMI/GoldenAMI to use Launch template / Launch configuration feature.**

Create Launch template,

Navigate to ec2 console and Choose the Instance you want to use for creating a template.



Then Launch template will pick all settings from the instance we selected.

Launch template name and description

Source instance
i-0e1fe63e0cc69baa7

Launch template name - *required*

Must be unique to this account. Max 128 chars. No spaces or special characters like '&', '*', '@'.

Template version description

Max 255 chars

Auto Scaling guidance [Info](#)
Select this if you intend to use this template with EC2 Auto Scaling
 Provide guidance to help me set up a template that I can use with EC2 Auto Scaling

▼ Amazon machine image (AMI) - required [Info](#)

AMI - *required*

ami-010aff33ed5991201
architecture: 64-bit (x86) virtualization: hvm

▼ Instance type [Info](#)

Instance type

Family: t2 1 vCPU 1 GiB Memory
On-Demand Linux pricing: 0.0124 USD per Hour
On-Demand Windows pricing: 0.017 USD per Hour

Free tier eligible

[Compare instance types](#)

▼ Key pair (login) [Info](#)

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name

learnawskp

Template value ▾

 Create new key pair**▼ Network settings**Networking platform [Info](#) Virtual Private Cloud (VPC)

Launch into a virtual network in your own logically isolated area within the AWS cloud

 EC2-Classic

Launch into a single flat network that you share with other customers.

Security groups

Select security groups

**▼ Storage (volumes)** [Info](#)

- ▶ Volume 1 (AMI Root) (Custom) (8 GiB, EBS, General purpose SSD (gp2))

[Add new volume](#)**▼ Network interfaces** [Info](#)

Network interface 1

[Remove](#)Device index [Info](#)

0

Network interface [Info](#)

New interface ▾

Description [Info](#)

My Primary ENI

Subnet [Info](#)

subnet-7b3ed510 ▾

 Remove subnet for EC2 Auto Scaling templates

Security groups

Select security groups ▾

Auto-assign public IP [Info](#)

Enable ▾

Not applicable for EC2 Auto Scaling

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

Make sure you remove the subnet selection for ASG and Click on “Create launch template”

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.

Create launch configuration [Info](#)

Launch configuration name

Name
Launchconfigv1.0

Amazon machine image (AMI) [Info](#)

AMI
WebServerAMI

Instance type [Info](#)

Instance type
t2.micro (1 vCPUs, 1 GiB, EBS Only) [Choose instance type](#)

Storage (volumes) [Info](#)

EBS volumes

<input type="checkbox"/>	Volume type	Devices	Snapshot	Size (GiB)	Volume type
<input checked="" type="checkbox"/>	Root	/dev/xvda	snap-04504e1afcb025a38	8	General purpose SSD (g)

+ Add new volume [Remove](#)

The screenshot shows the AWS Security Groups configuration interface. At the top, there are fields for 'Security group name' (AutoScaling-Security-Group-1) and 'Description' (AutoScaling-Security-Group-1 (2021-06-25T11:04:49.825Z)). Below these are 'Rules' settings, which include a table with columns: Type, Protocol, Port range, Source type, and Source. A single rule is listed: Type SSH, Protocol TCP, Port range 22, Source type Anywhere, and Source 0.0.0.0/0. There is also a 'Remove' button. Underneath the rules section is a 'Key pair (login)' field, which currently says 'Choose an existing key pair'. Below this is a dropdown for 'Existing key pair' containing 'learnawskp'. A checkbox at the bottom states: 'I acknowledge that I have access to the selected private key file (learnawskp.pem), and that without this file, I won't be able to log into my instance.' The bottom right of the screen has 'Cancel' and 'Create launch configuration' buttons.

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.

Auto Scaling group name
Enter a name to identify the group.

Must be unique to this account in the current Region and no more than 255 characters.

Launch configuration [Info](#) [Switch to launch template](#)

Launch configuration
Choose a launch configuration that contains the instance-level settings, such as the Amazon Machine Image (AMI), instance type, key pair, and security groups.

[C](#)

[Create a launch configuration](#)

Launch configuration	AMI ID	Date created
Launchconfigv1.0	ami-05a52ff0e5e6e80a3	Fri Jun 25 2021 17:28:44 GMT+0530 (India Standard Time)
Security groups	Instance type	Key pair name
sg-00b8ff10b49315c74	t2.micro	-

On the Choose launch template or configuration page, for Auto Scaling group name, **enter a name for your Auto Scaling group**.

Network [Info](#)

For most applications, you can use multiple Availability Zones and let EC2 Auto Scaling balance your instances across the zones. The default VPC and default subnets are suitable for getting started quickly.

VPC
 [C](#)
172.31.0.0/16 Default

[Create a VPC](#)

Subnets
 [C](#)

[ap-south-1a | subnet-7b3ed510](#) [X](#)
172.31.32.0/20 Default

[ap-south-1b | subnet-5e451d12](#) [X](#)
172.31.0.0/20 Default

[Create a subnet](#)

[Cancel](#) [Previous](#) [Skip to review](#) [Next](#)

For **Network**, choose the VPC that you used for your load balancer. I have selected Default vpc with 2 subnets, Always choose 2 subnets for High availability purpose.

Use the options below to attach your Auto Scaling group to an existing load balancer, or to a new load balancer that you define.

 No load balancer

Traffic to your Auto Scaling group will not be fronted by a load balancer.

 Attach to an existing load balancer

Choose from your existing load balancers.

 Attach to a new load balancer

Quickly create a basic load balancer to attach to your Auto Scaling group.

Attach to a new load balancer

Define a new load balancer to create for attachment to this Auto Scaling group.

Load balancer type

Choose from the load balancer types offered below. Type selection cannot be changed after the load balancer is created. If you need a different type of load balancer than those offered here, [visit the Load Balancing console.](#)

 Application Load Balancer

HTTP, HTTPS

 Network Load Balancer

TCP, UDP, TLS

Choose an Existing ELB (Create an ELB and keep it ready) or create a new ELB by following below images.

Load balancer scheme

Scheme cannot be changed after the load balancer is created.

 Internal Internet-facing

Network mapping

Your new load balancer will be created using the same VPC and Availability Zone selections as your Auto Scaling group. You can select different subnets and add subnets from additional Availability Zones.

VPC

vpc-a15391ca

Availability Zones and subnets

You must select a single subnet for each Availability Zone enabled. Only public subnets are available for selection to support DNS resolution.

ap-south-1a

subnet-7b3ed510

ap-south-1b

subnet-5e451d12

ap-south-1c

Select a subnet

Listeners and routing
If you require secure listeners, or multiple listeners, you can configure them from the [Load Balancing console](#) after your load balancer is created.

Protocol	Port	Default routing (forward to)
HTTP	80	Create a target group ▾

New target group name
An instance target group with default settings will be created.
MyASG-1

Tags - *optional*
Consider adding tags to your load balancer. Tags enable you to categorize your AWS resources so you can more easily manage them.

Add tag

50 remaining

Health checks - *optional*

Health check type [Info](#)
EC2 Auto Scaling automatically replaces instances that fail health checks. If you enabled load balancing, you can enable ELB health checks in addition to the EC2 health checks that are always enabled.

EC2 ELB

Health check grace period
The amount of time until EC2 Auto Scaling performs the first health check on new instances after they are put into service.
300 seconds

Additional settings - *optional*

Monitoring [Info](#)
 Enable group metrics collection within CloudWatch

Cancel Previous Skip to review Next

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.

Enable group metrics if we are looking to get group metrics for all the instances running in ASG.

Group size - optional Info

Specify the size of the Auto Scaling group by changing the desired capacity. You can also specify minimum and maximum capacity limits. Your desired capacity must be within the limit range.

Desired capacity

Minimum capacity

Maximum capacity

In above example, I have setup “Desired Capacity” as 2, so application runs with 2 instances always, if any one of the instance is down ASG creates a new instance automatically.

Scaling policies - optional

Choose whether to use a scaling policy to dynamically resize your Auto Scaling group to meet changes in demand. Info

Target tracking scaling policy
Choose a desired outcome and leave it to the scaling policy to add and remove capacity as needed to achieve that outcome.

None

Instance scale-in protection - optional

Instance scale-in protection
If protect from scale in is enabled, newly launched instances will be protected from scale in by default.

Enable instance scale-in protection

Cancel **Previous** **Skip to review** **Next**

Add notifications Info

Send notifications to SNS topics whenever Amazon EC2 Auto Scaling launches or terminates the EC2 instances in your Auto Scaling group.

Add notification

Cancel **Previous** **Skip to review** **Next**

Tags (1)

Key	Value - optional	Tag new instances
Name	ASG Instance	<input checked="" type="checkbox"/>
Add tag		Remove
49 remaining		

[Cancel](#) [Previous](#) [Next](#)

EC2 > Auto Scaling groups

Auto Scaling groups (1)

<input type="checkbox"/>	Name	Launch template/configuration	Instances	Status	Desired capacity	Min v
<input type="checkbox"/>	MyASG	Launchconfigv1.0	2	-	2	1

Instances (3) [Info](#)

<input type="checkbox"/>	Name	Instance ID	Instance state	Instance type
<input type="checkbox"/>	ASG Instance	i-0f1cdae6a40aa1c47	Running	t2.micro
<input type="checkbox"/>	ASG Instance	i-08fe8dfc874f9c335	Running	t2.micro

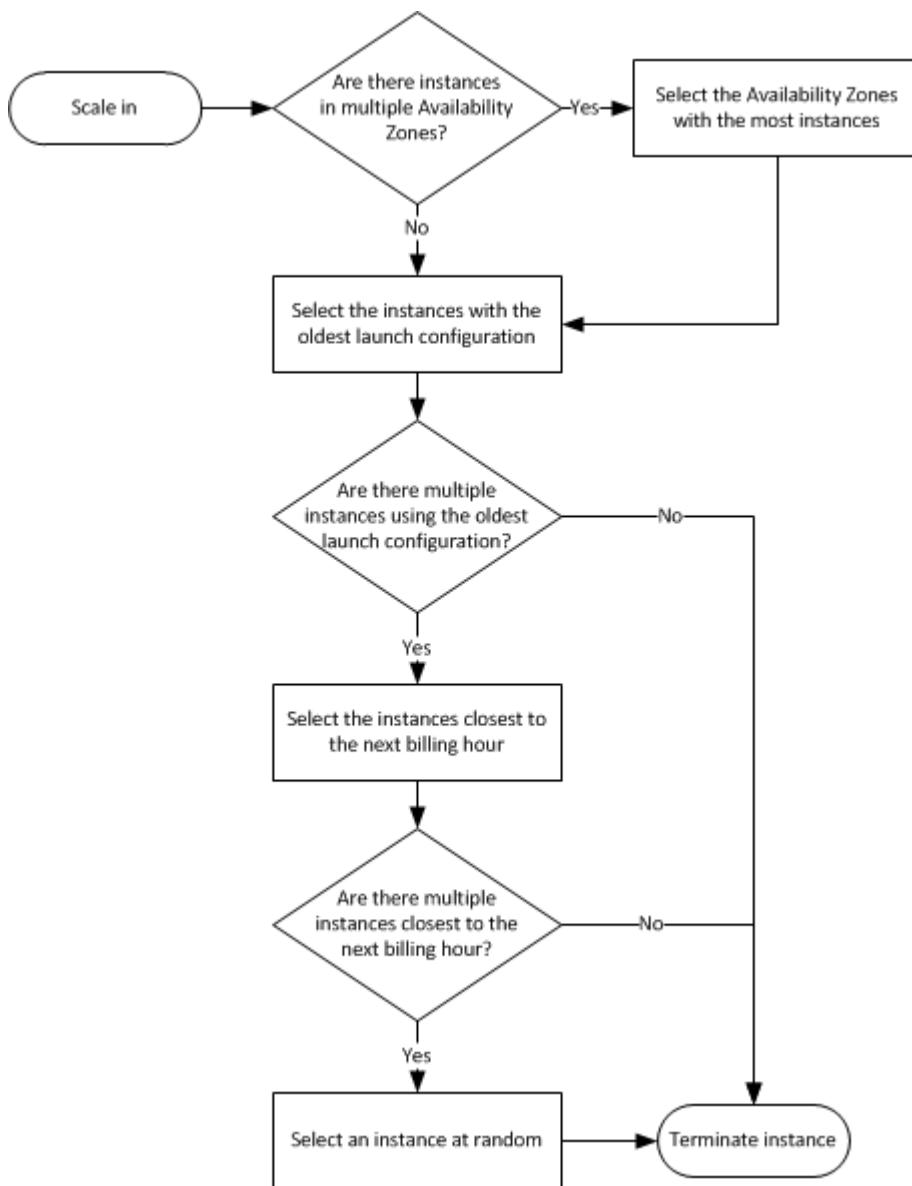
To configure Dynamic Scaling, “Create an Alarm” and define the required count.

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.

If you want to know, what userdata given to an ec2 instance, run the below URL in browser or terminal, it displays the output with given userdata.

Curl <http://169.254.169.254/latest/user-data/>

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 install httpd -y  
echo "Hi This is a Bootstrap script generated webpage" > /var/www/html/index.html  
service httpd start  
chkconfig httpd 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

User data (i) As text As file Input is already base64 encoded

```
#!/bin/bash  
yum install httpd -y  
service httpd start  
chkconfig httpd on  
echo "<h1>This is BSS Webserver</h1>" > /var/www/html/index.html
```

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)



This is BSS Webserver

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

For Windows:

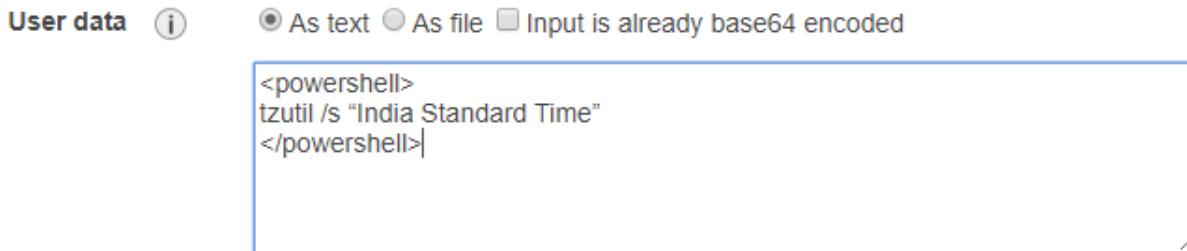
For EC2Config or EC2Launch to execute user data scripts, you must enclose the lines of the specified script within one of the following special tags:

```
<script></script>
```

```
<powershell></powershell>
```

Example: <powershell> tzutil /s "India Standard Time" </powershell>

Tzutil is a tool to set the time zone in windows operating systems. Without loggin into the ec2 instance, we can setup the timezone.



1. Here we have run very simple script to change the timezone to "India standard time" inside an ec2 instance.
2. Login to instance and verify the opoutput.

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 Windows](#) installer.

MacOS

Download and run the [MacOS PKG](#) installer.

Linux

Download, unzip, and then run the [Linux](#) installer

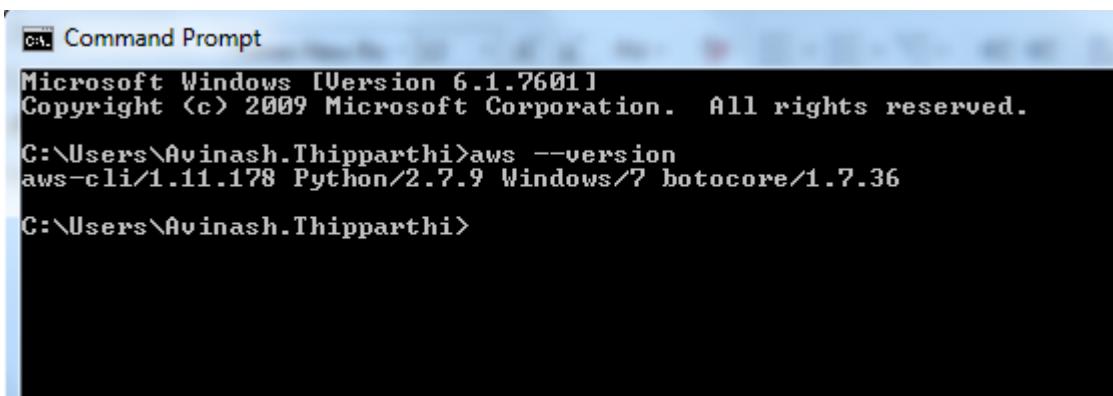
Amazon Linux

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:

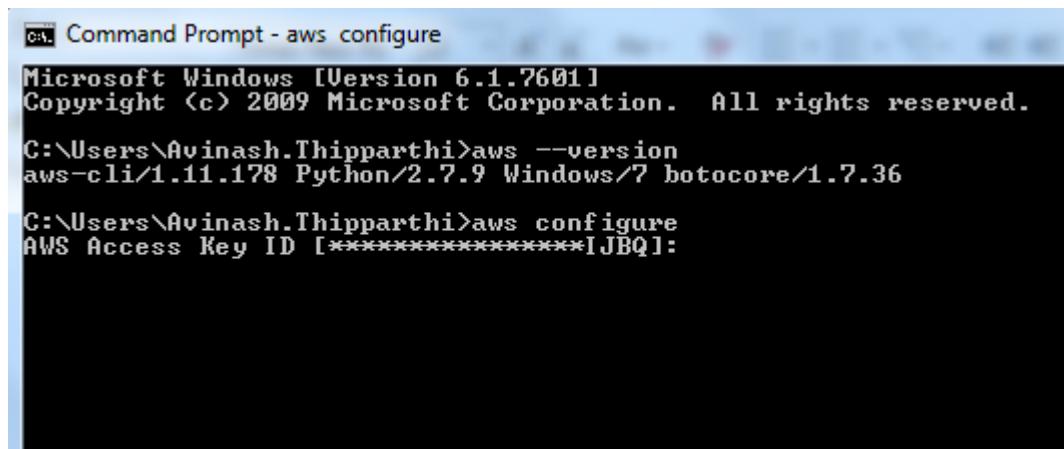
1. First we have to download the setup file from the above mentioned webpage, then follow the simple installation wizard.
2. After installing these tools, we can use the windows command prompt to connect to AWS resources/services.
3. To verify CLI tools installation, open command prompt and enter “**AWS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



```
cmd Command Prompt
Microsoft Windows [Version 6.1.7601]
Copyright <c> 2009 Microsoft Corporation. All rights reserved.

C:\Users\Avinash.Thipparthi>aws --version
aws-cli/1.11.178 Python/2.7.9 Windows/7 botocore/1.7.36
C:\Users\Avinash.Thipparthi>
```

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.

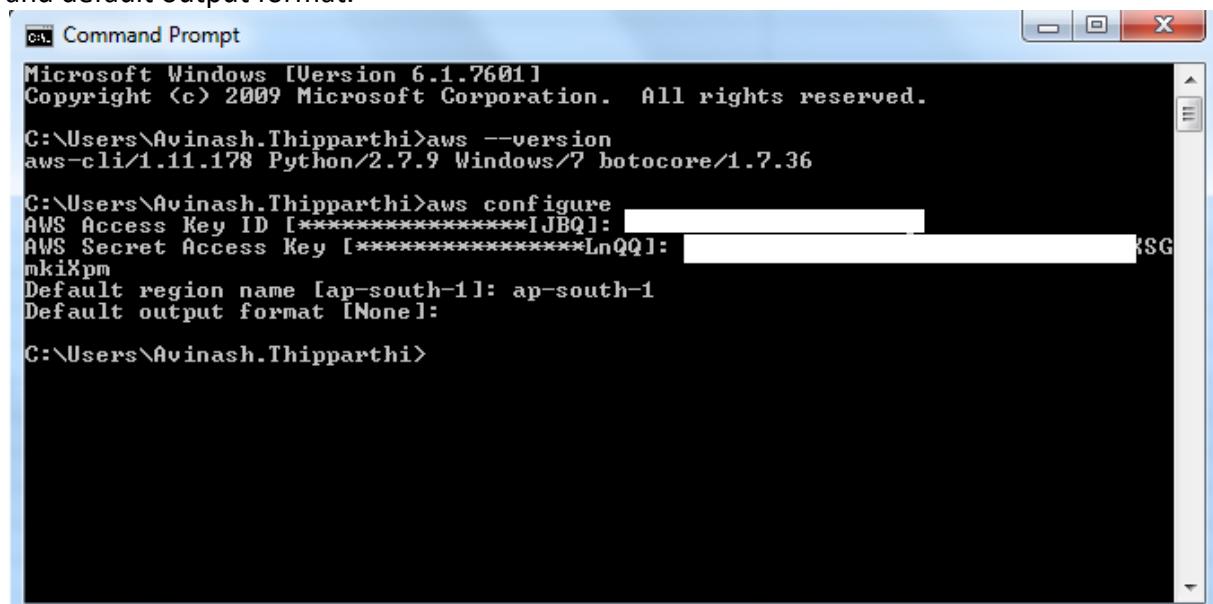


```
cmd Command Prompt - aws configure
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\Avinash.Thipparthi>aws --version
aws-cli/1.11.178 Python/2.7.9 Windows/7 botocore/1.7.36

C:\Users\Avinash.Thipparthi>aws configure
AWS Access Key ID [*****JBQ]:
```

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



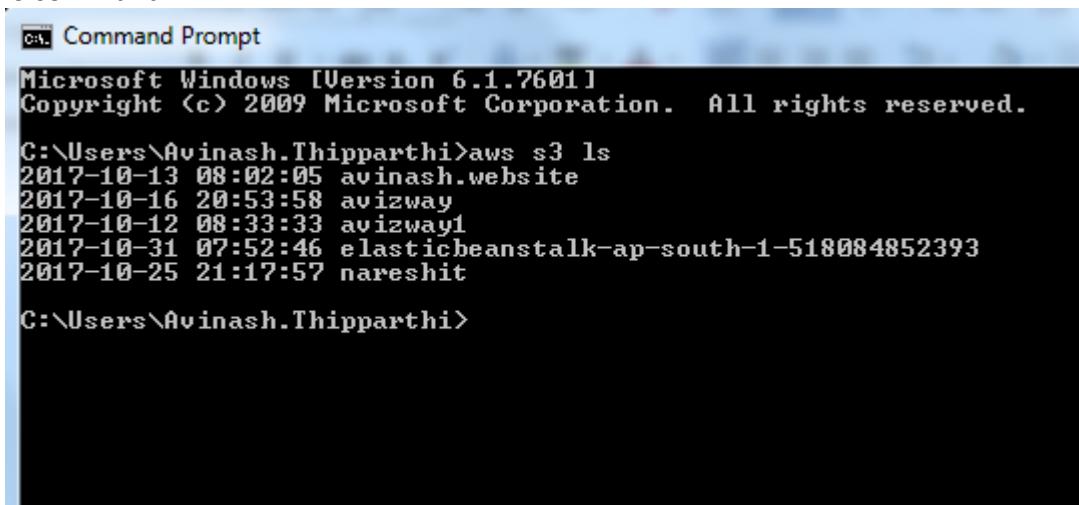
```
cmd Command Prompt
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\Avinash.Thipparthi>aws --version
aws-cli/1.11.178 Python/2.7.9 Windows/7 botocore/1.7.36

C:\Users\Avinash.Thipparthi>aws configure
AWS Access Key ID [*****JBQ]: [REDACTED]
AWS Secret Access Key [*****LnQQ]: [REDACTED] {SG
mkiXpm
Default region name [ap-south-1]: ap-south-1
Default output format [None]:}

C:\Users\Avinash.Thipparthi>
```

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.

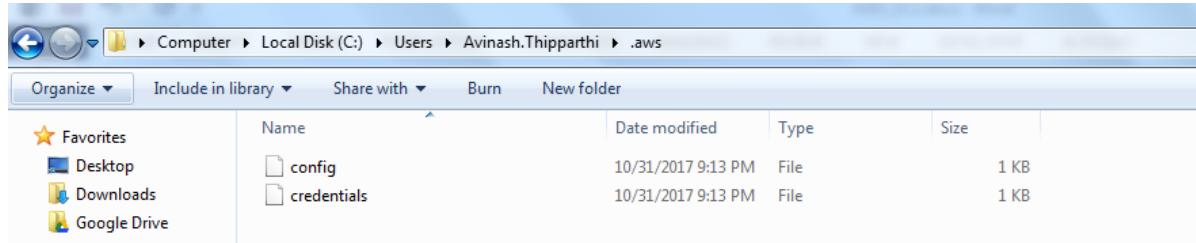


```
cmd Command Prompt
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\Avinash.Thipparthi>aws s3 ls
2017-10-13 08:02:05 avinash.website
2017-10-16 20:53:58 avizway
2017-10-12 08:33:33 avizway1
2017-10-31 07:52:46 elasticbeanstalk-ap-south-1-518084852393
2017-10-25 21:17:57 nareshit

C:\Users\Avinash.Thipparthi>
```

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.

```

root@ip-172-31-10-135:~/aws
[ec2-user@ip-172-31-10-135 ~]$ sudo su
[root@ip-172-31-10-135 ec2-user]# cd ~
[root@ip-172-31-10-135 ~]# ls -a
. .. .bash_logout .bash_profile .bashrc .cshrc .ssh .tcshrc
[root@ip-172-31-10-135 ~]# aws configure
AWS Access Key ID [None]: [REDACTED]
AWS Secret Access Key [None]: [REDACTED]
Default region name [None]: ap-south-1
Default output format [None]:
[root@ip-172-31-10-135 ~]# aws s3 ls
2017-10-13 02:32:05 avinash.website
2017-10-16 15:23:58 avizway
2017-10-12 03:03:33 avizway1
2017-10-31 02:22:46 elasticbeanstalk-ap-south-1-518084852393
2017-10-25 15:47:57 nareshit
[root@ip-172-31-10-135 ~]# ls -a
. .aws .bash_profile .cshrc .tcshrc
.. .bash_logout .bashrc .ssh
[root@ip-172-31-10-135 ~]# cd .aws/
[root@ip-172-31-10-135 .aws]# ls
config credentials
[root@ip-172-31-10-135 .aws]#

```

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 contain 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.

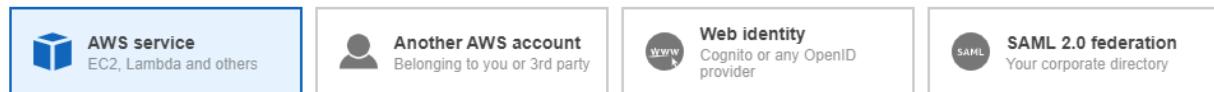
Role name	Trust entities	Last act...
adminrole	AWS Service: ec2	13 days ago
aws	AWS Service: eks	None

3. We have four options 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.

Create role

1 2 3 4

Select type of trusted entity

Allows AWS services to perform actions on your behalf. [Learn more](#)

Choose a use case

Common use cases

EC2

Allows EC2 instances to call AWS services on your behalf.

Lambda

Allows Lambda functions to call AWS services on your behalf.

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

The screenshot shows a list of policies under the "Filter policies" dropdown. The "Showing 840 results" label is visible at the top right. The "AdministratorAccess" policy is selected (indicated by a checked checkbox). Other policies listed include "AccessAnalyzerServiceRolePolicy", "AdministratorAccess-Amplify", and "AdministratorAccess-AWSElasticBeanstalk".

Policy name	Used as
AccessAnalyzerServiceRolePolicy	None
AdministratorAccess	Permissions policy (2)
AdministratorAccess-Amplify	None
AdministratorAccess-AWSElasticBeanstalk	None

6. Select appropriate role, based on your requirement, am selecting AdministratorAccess role here. Add tags if required, Then Select **Review**.

7. In review page, Give a name for the role and a valid description and select **Create Role** option.

Create role

1 2 3 4

Review

Provide the required information below and review this role before you create it.

Role name* AdminRole_Temp

Use alphanumeric and '+=-_,@-_-' characters. Maximum 64 characters.

Role description

Allows EC2 instances to call AWS services on your behalf.

Maximum 1000 characters. Use alphanumeric and '+=-_,@-_-' characters.

Trusted entities AWS service: ec2.amazonaws.com

Policies AdministratorAccess

Permissions boundary Permissions boundary is not set

* Required

Cancel

Previous

Create role

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

The screenshot shows the 'IAM role' configuration section of a Lambda function. A dropdown menu is open, listing several IAM roles: 'None', 'AdminRole_Temp' (which is highlighted in blue), 'aws-elasticbeanstalk-ec2-role', and 'ssmfullrole'. Below the dropdown, there is a checkbox for 'Enable CloudWatch detailed monitoring' and a note stating 'Additional charges apply.'

9. We can even associate a role after instance launch. To associate a role to existing instance, "Actions", "Security" and "**modify IAM Role**"

The screenshot shows the EC2 instance details page for an instance named 'ef'. The instance is 'Running' and of type 't2.micro'. A context menu is open under the 'Actions' button, with the 'Modify IAM role' option highlighted in yellow.

10. 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.

```
root@ip-172-31-4-199:~#
login as: ec2-user
Authenticating with public key "imported-openssh-key"
_____|_____|_
 ____| |_____|_ )   Amazon Linux AMI
____| \_____|__|_

https://aws.amazon.com/amazon-linux-ami/2017.09-release-notes/
1 package(s) needed for security, out of 6 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-4-199 ~]$ sudo su
[root@ip-172-31-4-199 ec2-user]# cd ~
[root@ip-172-31-4-199 ~]# pwd
/root
[root@ip-172-31-4-199 ~]# ls -a
. . . .bash_logout .bash_profile .bashrc .cshrc .ssh .tcshrc
[root@ip-172-31-4-199 ~]#
```

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

```
[root@ip-172-31-4-199 ~]# aws s3 ls
2017-10-13 02:31:40 avinash.website
2017-10-16 15:23:58 avizway
2017-10-11 02:34:59 avizway1
2017-10-25 01:13:02 elasticbeanstalk-ap-south-1-518084852393
2017-10-25 15:47:56 nareshit
[root@ip-172-31-4-199 ~]#
```

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

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]# █
```

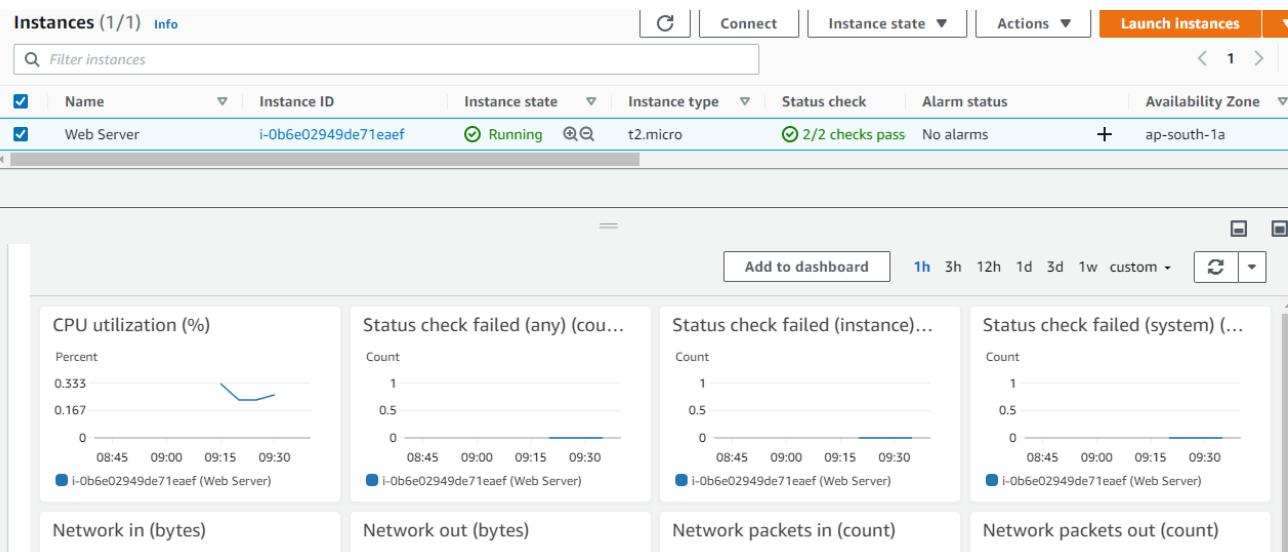
3. It is returned with all the available option, now whatever the information you want to get, give it along with the URL.
Ex: if you want to know hostname, give as Curl <http://169.254.169.254/latest/meta-data/hostname>

```
[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 cloudwatch monitorings.

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).

▼ Cost Management Preferences

Receive Free Tier Usage Alerts

Turn on this feature to receive email alerts when your AWS service usage is approaching, or has exceeded, the AWS Free Tier limit. If you wish to receive these alerts at an email address that is not the primary email address associated with this account, please enter the email address below.

Email Address:

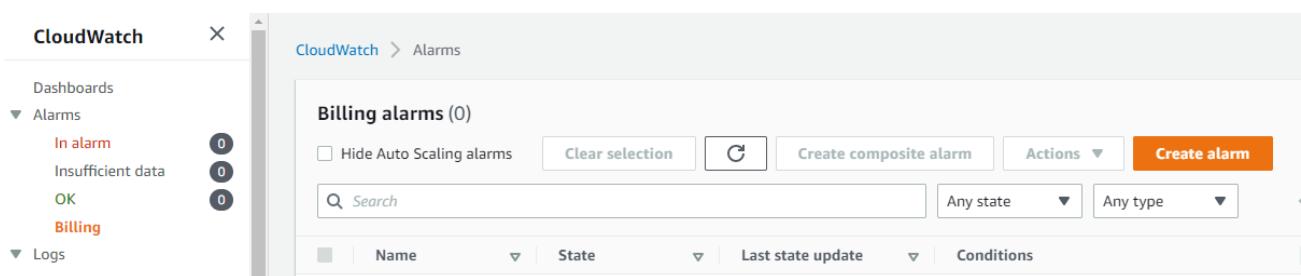
Receive Billing Alerts

Turn on this feature to monitor your AWS usage charges and recurring fees automatically, making it easier to track and manage your AWS bill. You can set up billing alerts to receive email notifications when your charges reach a specified threshold. Once enabled, this feature cannot be disabled. [Manage Billing Alerts](#) or [try the new budgets feature!](#)

► Detailed Billing Reports [Legacy]

Save preferences

4. When you click on "**ManageBilling Alerts**" option, you'll redirect to Cloudwatch dashboard, there select "**Billing**" option from Left Pane, then click on "**Create Alarm**".



Specify metric and conditions

Metric

Graph

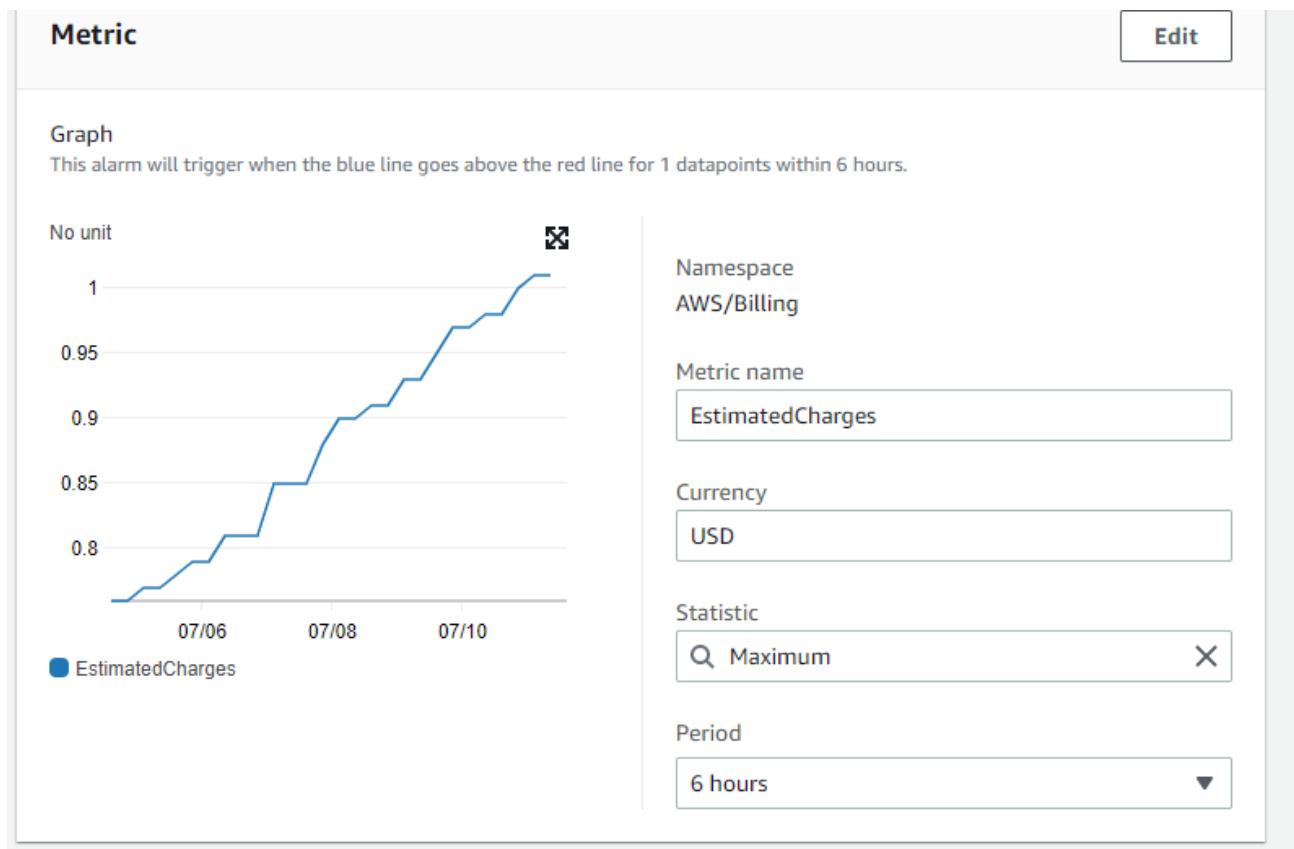
Preview of the metric or metric expression and the alarm threshold.

Select metric

Click on “Select metric”, then Choose “**billing**” then “**Total estimated charges**”.

The screenshot shows the AWS CloudWatch Metrics search interface. At the top, there is a legend with a blue circle labeled "EstimatedCharges". Below it, a navigation bar shows "Metrics (1)" and "View graphed metrics (1)". The main search area has a "Search for any metric, dimension or resource id" input field. Under the search results, there is a table with two rows. The first row has a checked checkbox and the text "Currency (1)". The second row has a checked checkbox and the text "USD ▾". To the right of the table, there are "Metric Name" and "EstimatedCharges ▾" labels. At the bottom right of the interface are "Cancel" and "Select metric" buttons.

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.



Conditions

Threshold type

Static
Use a value as a threshold

Anomaly detection
Use a band as a threshold

Whenever EstimatedCharges is...
Define the alarm condition.

Greater
> threshold

Greater/Equal
>= threshold

Lower/Equal
<= threshold

Lower
< threshold

than...
Define the threshold value.

0.5 USD

Must be a number

Alarm state trigger
Define the alarm state that will trigger this action.

In alarm
The metric or expression is outside of the defined threshold.

OK
The metric or expression is within the defined threshold.

Insufficient data
The alarm has just started or not enough data is available.

Remove

Select an SNS topic
Define the SNS (Simple Notification Service) topic that will receive the notification.

Select an existing SNS topic

Create new topic

Use topic ARN

Send a notification to...

Billing_Alarm

Only email lists for this account are available.

Email (endpoints)
[avizway@gmail.com](#) - View in SNS Console

Add notification

Add name and description

Name and description

Alarm name

Alarm description - optional

Up to 1024 characters (0/1024)

Cancel

The screenshot shows the AWS CloudWatch Alarms interface. At the top, there's a breadcrumb navigation: CloudWatch > Alarms > Demo_BillingAlarm. Below that, a navigation bar has 'Alarms' selected. The main area is titled 'Details' and contains a table with the following data:

Name	Demo_BillingAlarm	State	Insufficient data	Namespace	AWS/Billing	Datapoints to alarm	1 out of 1
Type	Metric alarm	Threshold	EstimatedCharges >= 0.5 for 1 datapoints within 6 hours	Metric name	EstimatedCharges	Missing data treatment	Treat missing data as missing
Description	No description	Last change	2021-07-12 10:02:47	Currency	USD	Percentiles with low samples evaluate	
		Actions	1 action(s) enabled	Statistic	Maximum	ARN	arn:aws:cloudwatch:us-east-1:501170964283:alarm:Demo_BillingAlarm
				Period	6 hours		

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.

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.

To create an Alarm, Choose the instance and Navigate to "Monitoring and troubleshooting" and click on "**Manage Cloudwatch Alarm**".

Instances (1/1) [Info](#)

[Filter instances](#)

Instance state	Instance type	Status check	Alarm status
Running	t2.micro	2/2 checks pass	No alarms

Instance: i-0b6e02949de71eaef (Web Server)

[Details](#) [Security](#) [Networking](#) [Storage](#) [Status check](#)

- [Get system log](#)
- [Get instance screenshot](#)
- [Manage detailed monitoring](#)
- [Manage CloudWatch alarms](#)

Instance ID: i-0b6e02949de71eaef (Web Server) | Public IPv4 address: 3.108.196.242 | Replace root volume | Private IPv4 addresses: 172.31.32.84

Choose “Create an Alarm” and choose the “Alarm Notification” and thresholds when to trigger alarm and what happens when an alarm triggered.

Add or edit alarm [Info](#)

You can create a new alarm or edit an existing alarm.

Create an alarm
Create an alarm for i-0b6e02949de71eaef

Edit an alarm
Edit an existing alarm for i-0b6e02949de71eaef

Search for alarm
Find an alarm to modify
 Select an existing alarm to edit

Alarm notification [Info](#)

Configure the alarm to send notifications to an Amazon SNS topic when it is triggered.

ImpNotifications X

Alarm action [Info](#)

Specify the action to take when the alarm is triggered.

Reboot ▼

Alarm thresholds

Specify the metric thresholds for the alarm.

Group samples by	Type of data to sample
Maximum	CPU utilization
Alarm when	Percent
>=	90
Consecutive period	Period
1	15 Minutes
Alarm name	HighUsageAlarm

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.

Cloudwatch Events: Amazon CloudWatch Events delivers a near real-time stream of system events that describe changes in Amazon Web Services (AWS) resources. You can configure the following AWS services as targets for CloudWatch Events:

- Amazon EC2 instances
- AWS Lambda functions
- Streams in Amazon Kinesis Data Streams
- Delivery streams in Amazon Kinesis Data Firehose
- Amazon ECS tasks
- Systems Manager Run Command
- Systems Manager Automation
- AWS Batch jobs
- Step Functions state machines
- Pipelines in AWS CodePipeline
- AWS CodeBuild projects
- Amazon Inspector assessment templates
- Amazon SNS topics
- Amazon SQS queues

- Built-in targets—EC2 CreateSnapshot API call, EC2 RebootInstances API call, EC2 StopInstances API call, and EC2 TerminateInstances API call.
- The default event bus of another AWS account.

Cloudwatch Logs: You can use Amazon CloudWatch Logs to monitor, store, and access your log files from Amazon Elastic Compute Cloud (Amazon EC2) instances, AWS CloudTrail, Route 53, and other sources.

- Monitor Logs from Amazon EC2 Instances in Real-time
- Monitor AWS CloudTrail Logged Events
- Log Retention
- Archive Log Data
- Log Route 53 DNS Queries

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. Click on “**Create file system**” and then choose “**customize**”.

Amazon Elastic File System

Scalable, elastic, cloud-native NFS file system

Amazon Elastic File System (Amazon EFS) provides a simple, scalable, elastic file system for general purpose workloads for use with AWS Cloud services and on-premises resources.

Create file system

Create an EFS file system with service recommended settings.

[Create file system](#)

2. 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**.

General

Name - *optional*
Name your file system.

Name must not be longer than 256 characters, and must only contain letters, numbers, and these characters: + - = . _ : /

Availability and Durability
Choose Regional (recommended) to create a file system using regional storage classes. Choose One Zone to create a file system using One Zone storage classes. [Learn more](#)

Regional
Stores data redundantly across multiple AZs

One Zone
Stores data redundantly within a single AZ

Automatic backups
Automatically backup your file system data with AWS Backup using recommended settings. Additional pricing applies. [Learn more](#)

Enable automatic backups

Lifecycle management
Automatically save money as access patterns change by moving files into the Standard - Infrequent Access storage class. [Learn more](#)

- **Regional** to create a file system that uses Standard storage classes. Standard storage classes store file system data and metadata redundantly across all Availability Zones within an AWS Region. Regional offers the highest levels of availability and durability.
- **One Zone** to create a file system that uses One Zone storage classes. One Zone storage classes store file system data and metadata redundantly within a single Availability Zone which makes it less expensive than Standard storage classes.
- **Automatic backups** are turned on by default. You can turn off automatic backups by clearing the check box.
- Choose a Lifecycle management policy. The default policy is 30 days after last access. If you don't want to use lifecycle management, choose None. We can use this feature to reduce the cost.

- **General Purpose** is ideal for latency-sensitive use cases, like web serving environments, content management systems, home directories, and general file serving.
- **Max I/O mode** can scale to higher levels of aggregate throughput and operations per second.
- **Bursting Throughput mode:** throughput on Amazon EFS scales as the size of your file system in the EFS Standard or One Zone storage class grows.
- **Provisioned Throughput mode:** you can instantly provision the throughput of your file system (in MiB/s) independent of the amount of data stored.

3. Configure the Network options in Next screen. Am making EFS available in all the 3 AZs. I have a Security with “NFS Protocol” opened.

Network

Virtual Private Cloud (VPC)
Choose the VPC where you want EC2 instances to connect to your file system. [Learn more](#)

vpc-01265273360128840 default	▼
----------------------------------	---

Mount targets
A mount target provides an NFSv4 endpoint at which you can mount an Amazon EFS file system. We recommend creating one mount target per Availability Zone. [Learn more](#)

Availability zone	Subnet ID	IP address	Security groups
ap-south-1a	subnet-0f679df974f09ab7b	Automatic	Choose security groups ▼ sg-0f85c8b81dd95a380 X EFSSG
ap-south-1b	subnet-0051b17287be7ff5f	Automatic	Choose security groups ▼ sg-0f85c8b81dd95a380 X EFSSG
ap-south-1c	subnet-06be8837d1e387...	Automatic	Choose security groups ▼ sg-0f85c8b81dd95a380 X EFSSG

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

Amazon EFS > File systems > fs-ad003a7c

EFS (fs-ad003a7c)

[Delete](#) [Attach](#)

General		Edit
Performance mode	Automatic backups	
General Purpose	(●) Enabling	
Throughput mode	Encrypted	
Bursting	a45fdacb-7621-495f-9e34-5b652812f5f1 (aws/elasticfilesystem)	
Lifecycle policy	File system state	
30 days since last access	(●) Available	
Availability zone		
Regional		

- Metered size
- Monitoring
- Tags
- File system policy
- Access points
- Network

Metered size

Total size 6.00 KiB Size in Standard / One Zone 6.00 KiB (100%) Size in Standard-IA / One Zone-IA 0 Bytes (0%)	 <div style="margin-top: 10px;"> ■ Size in Standard / One Zone ■ Size in Standard-IA / One Zone-IA </div>
---	---

5. 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.

Attach X

Mount your Amazon EFS file system on a Linux instance. [Learn more](#)

Mount via DNS Mount via IP

Using the EFS mount helper:

```
sudo mount -t efs -o tls fs-ad003a7c:/ efs
```

Using the NFS client:

```
sudo mount -t nfs4 -o nfsvers=4.1,rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2,noresvport fs-ad003a7c.efs.ap-south-1.amazonaws.com:/ efs
```

See our user guide for more information. [User guide](#)

[Close](#)

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

```

root@ip-172-31-26-139:~#
login as: ec2-user
Authenticating with public key "imported-openssh-key"

      _|_(_|_) /   Amazon Linux AMI
      __\_\_|__|_|

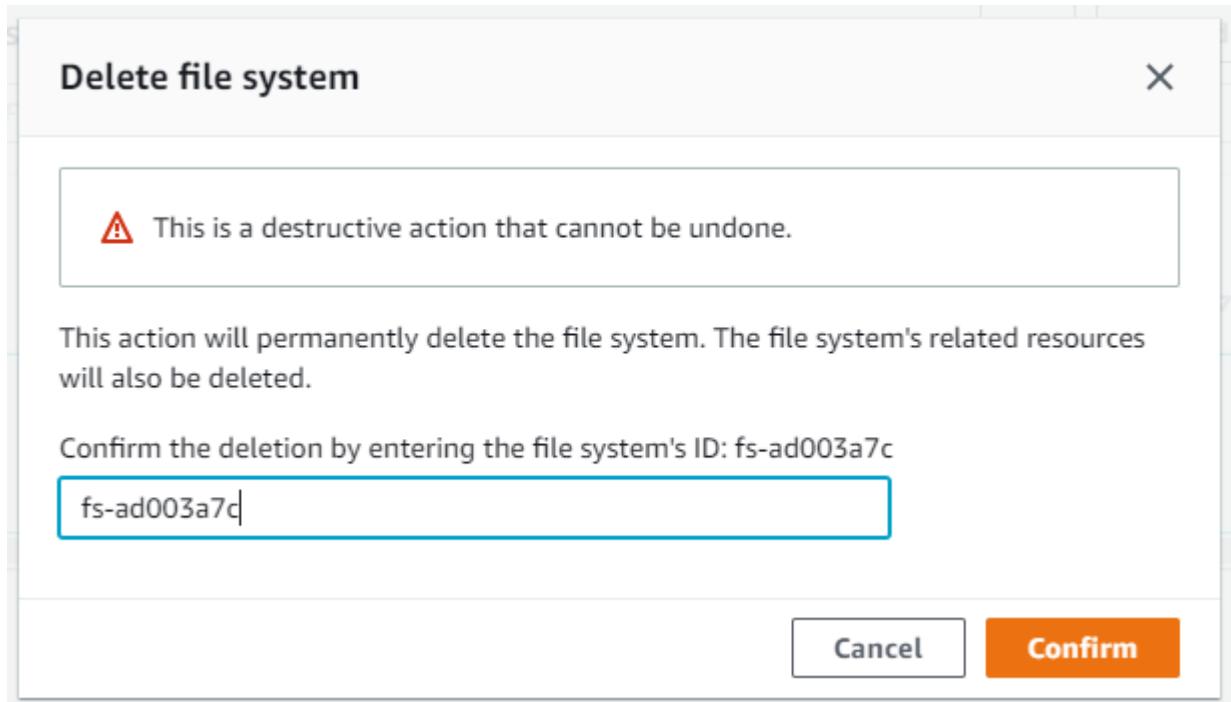
https://aws.amazon.com/amazon-linux-ami/2017.09-release-notes/
1 package(s) needed for security, out of 6 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-26-139 ~]$ sudo su
[root@ip-172-31-26-139 ec2-user]# cd ~
[root@ip-172-31-26-139 ~]# sudo yum install -y nfs-utils
Loaded plugins: priorities, update-motd, upgrade-helper
Package 1:nfs-utils-1.3.0-0.21.amzn1.x86_64 already installed and latest version
Nothing to do
[root@ip-172-31-26-139 ~]# sudo mkdir efs
[root@ip-172-31-26-139 ~]# sudo mount -t nfs4 -o nfsvers=4.1,rsiz
e=1048576,wsiz
e=1048576,hard,timeo=600,retrans=2 fs-312a7678.efs.us-east-1.amazonaws.com:/ efs

```

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

Name	File system ID	Encrypted	Total size	Size in Standard / One Zone	Size in Standard-IA / One Zone-IA
EFS	fs-ad003a7c	Encrypted	6.00 KiB	6.00 KiB	0 Bytes

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



Amzon FSx : Amazon FSx provides you with the native compatibility of third-party file systems with feature sets for workloads such as Windows-based storage, high-performance computing (HPC), machine learning, and electronic design automation (EDA).

We need to have Microsoft Active Directory ID to use Amazon FSx Service, Please refer to this video to create a Simple AD in AWS and continue creating the Amazon FSx.

AWS Directory Service Creation : <https://www.youtube.com/watch?v=RDIBoAHVmZs>

FSx: <https://www.youtube.com/watch?v=s15ljWSTWY4>

AWS Server Migration Service : <https://www.youtube.com/watch?v=Vx9lRRY-shg>

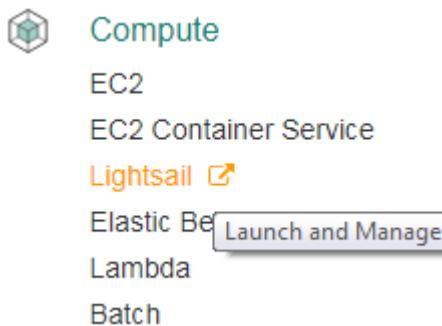
LIGHTSAIL :

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.

You have no resources right now.

Create an instance and get started with Lightsail!

[Create instance](#)



[Create static IP](#)

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.



You are creating this instance in **Mumbai, Zone A** (ap-south-1a).

[Change Region and zone](#)

Pick your instance image [?](#)

Select a platform



Linux/Unix
15 blueprints



Microsoft Windows
3 blueprints

Select a blueprint

[Apps + OS](#)

[OS Only](#)



WordPress
4.8.1



LAMP Stack
5.6.31



Node.js
8.4.0



Joomla
3.7.5



Magento
2.1.8-1



MEAN
3.4.7



Drupal
8.3.7-1



GitLab CE
9.5.0

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

Choose your instance plan ?

\$5 month USD <small>First month free!</small>	\$10 month USD	\$20 month USD	\$40 month USD	\$80 month USD
0.007 \$/hour	0.013 \$/hour	0.027 \$/hour	0.054 \$/hour	0.108 \$/hour
512 MB RAM 1 vCPU 20 GB SSD 512 GB data transfer	1 GB RAM 1 vCPU 30 GB SSD 1 TB data transfer	2 GB RAM 1 vCPU 40 GB SSD 1.5 TB data transfer	4 GB RAM 2 vCPUs 60 GB SSD 2 TB data transfer	8 GB RAM 2 vCPUs 80 GB SSD 2.5 TB data transfer

You can try the selected plan free for one month (up to 750 hours).

i Plans in Mumbai include lower data transfer allowances than other regions. [Learn more ↗](#)

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

Name your instance

Your Lightsail resources must have unique names.

x 1
Create

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

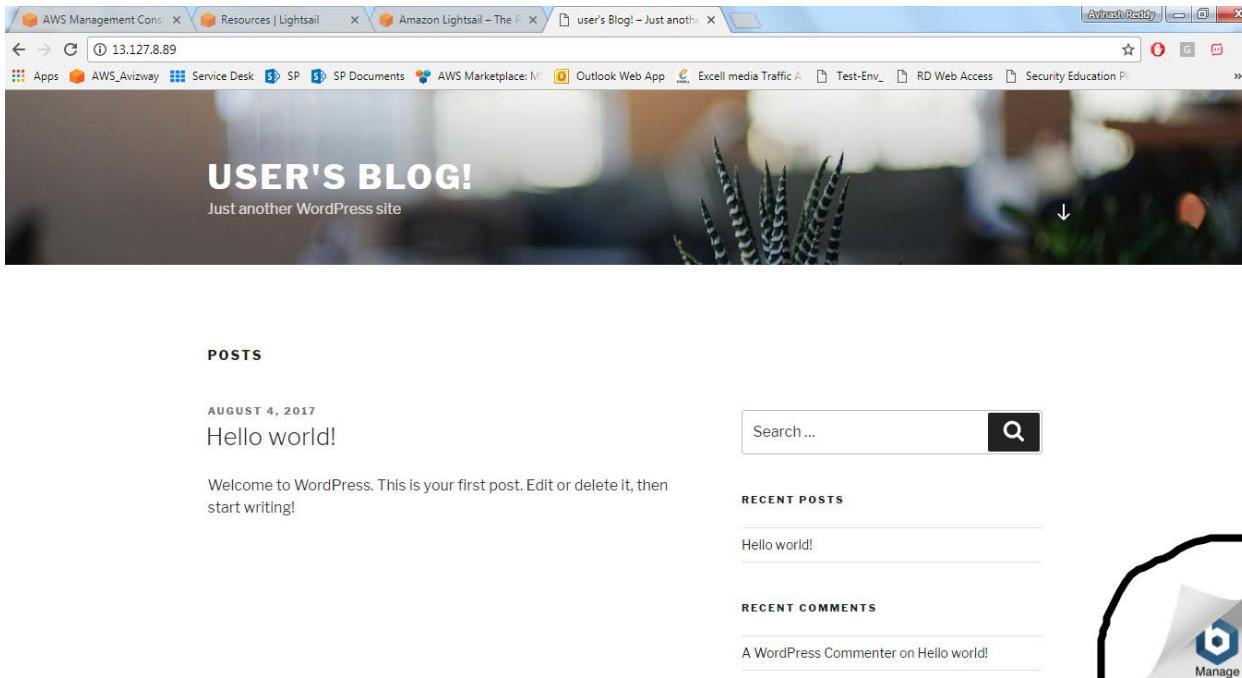


Mumbai (ap-south-1)

INSTANCES

 My-WordPress 512 MB RAM, 1 vCPU, 20 GB SSD Running	<div style="border: 1px solid #ccc; padding: 5px; background-color: #f0f0f0; margin-bottom: 5px;"> Connect </div> <div style="border: 1px solid #ccc; padding: 5px; background-color: #fff; display: inline-block; width: 150px; height: 150px; vertical-align: middle;"></div> <div style="list-style-type: none; padding-left: 0;"> Manage Stop Restart Delete </div>
---	--

4. We'll get a public IP address by using that Public IP, we can access the WP website.
5. 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.



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

This is a Cloud Image for WordPress built by Bitnami.

Access data for WordPress

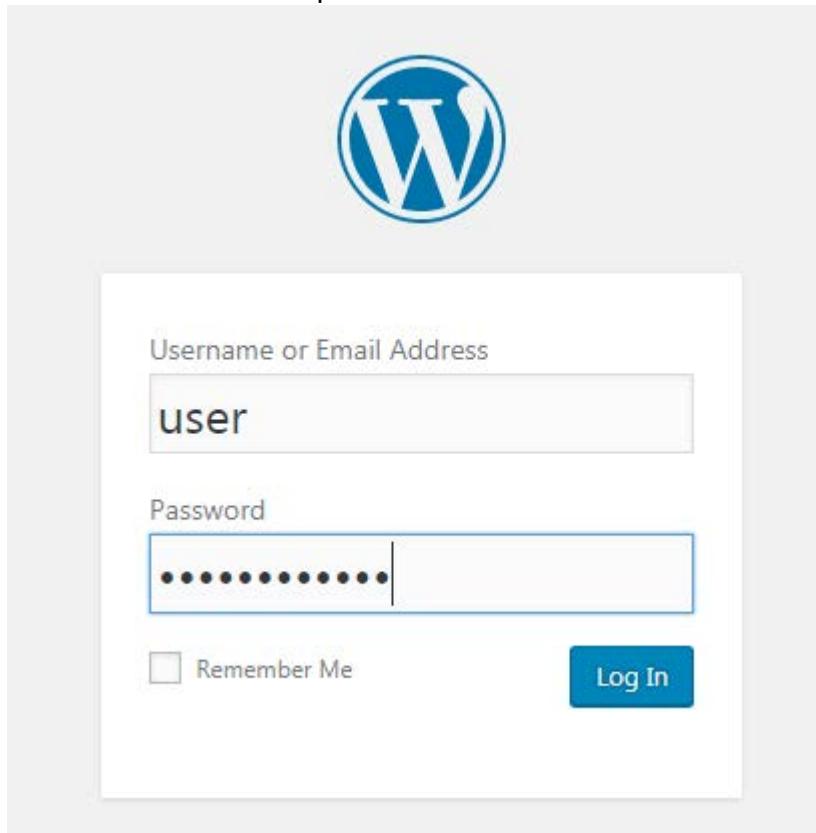
Username:	user
Password:	Created on first boot. Follow these instructions on how to retrieve the password.

[Login](#) to the admin console.

You should change the default credentials on first login.

7. 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.

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



9. 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.

Dashboard

Welcome to WordPress!

We've assembled some links to get you started:

Get Started

Customize Your Site

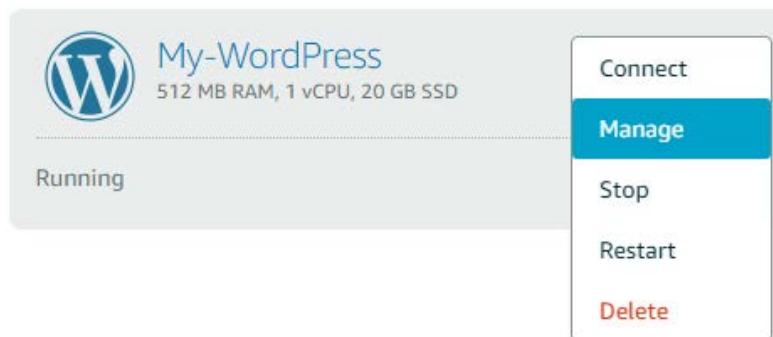
or, change your theme completely

Next Steps

-  Write your first blog post
-  Add an About page
-  View your site

10. 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.

INSTANCES



Private IP: 172.26.10.72 Pi

[Connect](#) [Metrics](#) [Networking](#) [Snapshots](#) [History](#) [Delete](#)

11. 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.

Application information

Application name Up to 100 Unicode characters, not including forward slash (/).

Base configuration

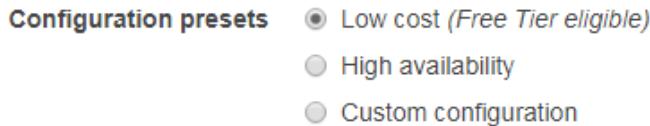
Platform Choose Configure more options for more platform configuration options.

Application code Sample application
Get started right away with sample code.

Upload your code
Upload a source bundle from your computer or copy one from Amazon S3.

my_mvc_application-source

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**



Platform 64bit Windows Server 2016 v1.2.0 running IIS 10.0 [Change platform 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

<p>Software</p> <p>AWS X-Ray: disabled Rotate logs: disabled (default) Environment properties: 0</p> <p>Modify</p>	<p>Instances</p> <p>EC2 instance type: t2.micro EC2 image ID: ami-8ae3a1e5 Root volume type: General Purpose (SSD) Root volume size (GB): container default Root volume IOPS: container default</p> <p>Modify</p>	<p>Capacity</p> <p>Environment type: single instance Availability Zones: Any Instances: 1–1</p> <p>Modify</p>
<p>Load balancer</p> <p><i>This configuration does not contain a load balancer.</i></p> <p>Modify</p>	<p>Rolling updates and deployments</p> <p>Deployment policy: All at once Rolling updates: disabled Health check: enabled</p> <p>Modify</p>	<p>Security</p> <p>Service role: aws-elasticbeanstalk-service-role Virtual machine key pair: -- Virtual machine instance profile: aws-elasticbeanstalk-ec2-role</p> <p>Modify</p>
<p>Monitoring</p> <p>Health check path: <i>blank</i> Health reporting system: --</p> <p>Modify</p>	<p>Notifications</p> <p>Email address: --</p> <p>Modify</p>	<p>Network</p> <p>VPC: vpc-7d7ab214 (default) Associate public IP address: disabled Instance subnets: <i>none</i> Security groups: <i>none</i></p> <p>Modify</p>

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.

i Creating MyMvcApplication-env
This will take a few minutes...

```

5:07pm Successfully launched environment: MyMvcApplication-env
5:06pm Environment health has been set to GREEN
5:06pm UpdateAppVersion Completed
5:05pm Started Application Update
5:02pm Adding instance i-Dec82b27133d[REDACTED] to your environment.
5:02pm Added EC2 instance i-Dec82b27133d[REDACTED] to Auto Scaling Group 'awseb-e-tcpizzpcwh-stack-AWSEBAutoScalingGroup[REDACTED]' PAOH.
5:01pm Waiting for EC2 instances to launch. This may take a few minutes.
5:00pm Created EIP: 13[REDACTED]
5:00pm Created security group named: awseb-e-tcpizzpcwh-stack-AWSEBSecurity[REDACTED]
5:00pm Using elasticbeanstalk-ap-south-1-518084[REDACTED] as Amazon S3 storage bucket for environment data.
5:00pm createEnvironment is starting.

```

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

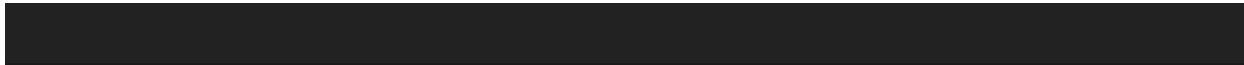
My_MVC_Application > MyMvcApplication-env (Environment ID: e-tcpizzpcwh, URL: MyMvcApplication-env.n[REDACTED] ap-south-1.elasticbeanstalk.com) Actions ▾

Overview Refresh

 Health Green Causes	Running Version my_mvc_application-source Upload and Deploy	 Configuration 64bit Windows Server 2016 v1.2.0 running IIS 10.0 Change
---	---	--

9. We'll get Environment ID to access the application.

10. Here is the output for my uploaded code.



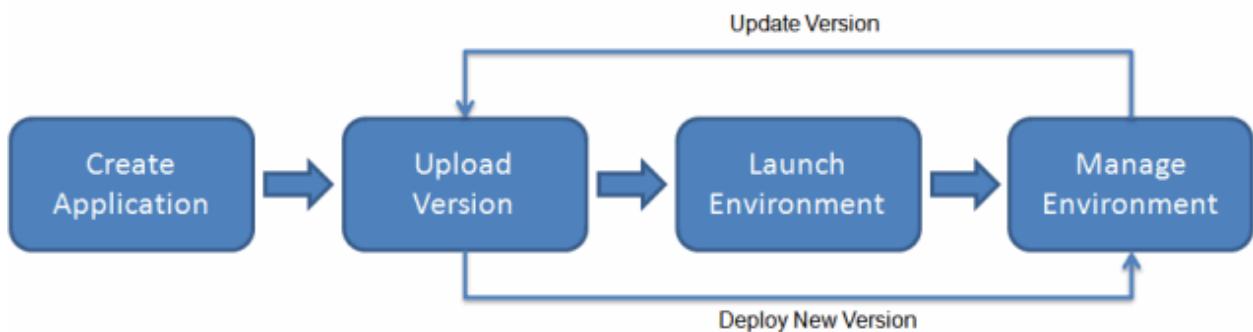
Hello Cloud World..!!

Here is My First .Net Project deployed in Minutes

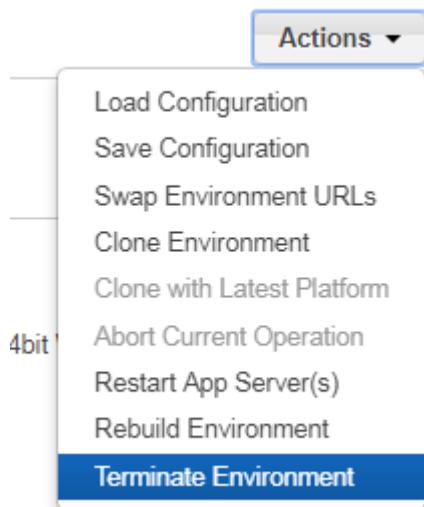


11. If you made any changes to your existing code, you can zip it and upload it.

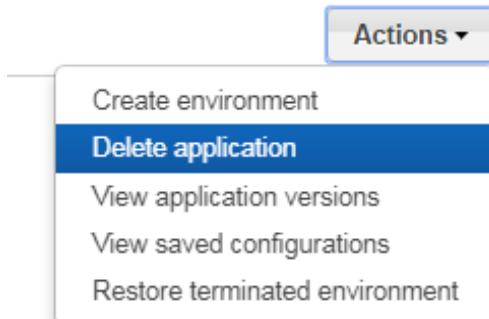
12. Here is the illustration diagram of workflow



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.



Amazon Route 53 (DNS Service)

- 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 farthest portion 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 users to reach the AWS servers when they type aws.amazon.com into their browsers. IP Address is a network addressable location. Each IP address must be unique within its network. 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 to organize 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 that manages 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. These servers do most of the work in the DNS. Because the total number of domain translations is too much for any one server, each server may redirect requests to other name servers or delegate responsibility for the subset of subdomains for which they are responsible.

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

Zone Files

A zone file is a simple text file that contains the mappings between domain names and IP addresses. This is how a DNS server finally identifies which IP address should be contacted when 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 a resource and a name. These can map a domain name to an IP address or define resources for the domain, such as name servers or mail servers. This section describes each record type in detail.

Start of Authority (SOA) Record

A Start of Authority (SOA) record is mandatory in all zone files, and it identifies the base DNS 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 host to 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 an alias 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 and ensure that email messages are routed correctly. The MX record should point to a host defined 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 IP address to a DNS name, and they are mainly used to check if the server name is associated with 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 to associate some arbitrary and unformatted text with a host or other name, such as human-readable 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 that provides 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 a traditional DNS zone file, a hosted zone represents resource record sets that are managed together under a single domain name. Each hosted zone has its own metadata and configuration information.

There are two types of hosted zones: private and public. A private hosted zone is a container that holds information about how you want to route traffic for a domain and its subdomains within one or more Amazon Virtual Private Clouds (Amazon VPCs). A public hosted zone is a container that holds

information about how you want to route traffic on the Internet for a domain (for example, example.com) and its subdomains (for example, apex.example.com and acme.example.com).

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

Routing Policies:

Simple Routing Policy

This is the default routing policy when you create a new resource. Use a simple routing policy when you have a single resource that performs a given function for your domain (for example, 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).

Weighted Routing Policy

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

Use the weighted routing policy when you have multiple resources that perform the same function (such as web servers that serve the same website), and you want Amazon Route 53 to route traffic to those resources in proportions that you specify. For example, you may use this 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 the older version of your website).

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

Latency-Based Routing Policy

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

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

Failover Routing Policy

Use a failover routing policy to configure active-passive failover, in which one resource takes all the traffic when it's available and the other resource takes all the traffic when the first resource isn't available. Note that you can't create failover resource record sets for private hosted 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 using a health check.

A health check tells Amazon Route 53 how to send requests to the endpoint whose health you want to check: which protocol to use (HTTP, HTTPS, or TCP), which IP address and port to use, 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 selected DNS endpoint. If your health check fails, then failover routing policies will be applied and your DNS will fail over to your DR site.

Geolocation

Geolocation routing lets you choose where Amazon Route 53 will send your traffic based on the geographic location of your users (the location from which DNS queries originate). For example, you might want all queries from Europe to be routed to a fleet of Amazon EC2 instances that are specifically configured for your European customers, with local languages and pricing in Euros.

You can also use geolocation routing to restrict distribution of content to only the locations in which you have distribution rights. Another possible use is for balancing load across endpoints in a predictable, easy-to-manage way so that each user location is consistently routed to the same endpoint.

You can specify geographic locations by continent, by country, or even by state in the United States. 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 one resource record set for Europe and one for the United Kingdom. This allows you to route some queries for selected countries (in this example, the United Kingdom) to one resource and to route queries for the rest of the continent (in this example, Europe) to a different resource.

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 the 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 a container that holds information about how you want to route traffic on the Internet for a domain and its subdomains.
 2. **Private Hosted Zone:** A private hosted zone is a container that holds information about how you want to route traffic for a domain and its subdomains within one or more VPC.

Create Hosted Zone

A hosted zone is a container that holds information about how you want to route traffic for a domain, such as example.com, and its subdomains.

Domain Name:

Comment:

Type: A public hosted zone determines how traffic is routed on the Internet.

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

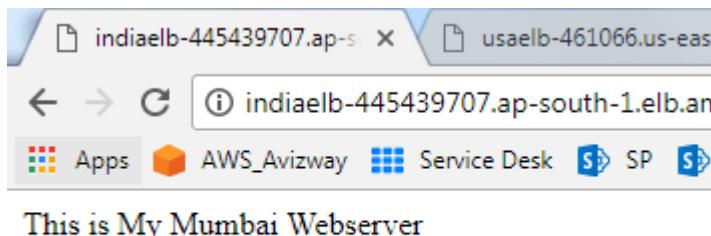
The screenshot shows the AWS Route 53 Record Set Manager interface. At the top, there are buttons for 'Back to Hosted Zones', 'Create Record Set' (which is highlighted in blue), 'Import Zone File', 'Delete Record Set', and 'Test Record Set'. Below this is a search bar with placeholder 'Record Set Name' and a dropdown 'Any Type'. There are also checkboxes for 'Aliases Only' and 'Weighted Only'. The main table displays two record sets:

	Name	Type	Value	Evaluate Target Health	Health Check ID	TTL
<input type="checkbox"/>	avinash.website.	NS	ns-awsdns-50.co.uk. ns-awsdns-30.org. ns-awsdns-09.net. ns-awsdns-20.com.	-	-	172800
<input type="checkbox"/>	avinash.website.	SOA	ns-co.uk. awsdns-hostmaster.amazon.com.	-	-	900

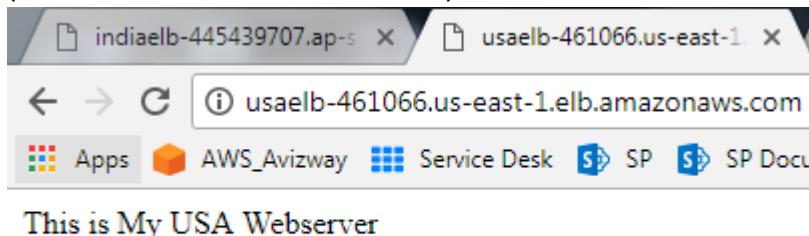
- Now the Hosted Zone is created. If you purchase the Domain name from any other domain registrar i.e; Godaddy, bigrock we have to configure these NameServers in that account, or we can transfer the domain to AWS.

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.

- Create an EC2 Instance in Mumbai region and connect to the instance
- Install **httpd** package and create **index.html** under **/var/www/html** and start the **httpd** service and verify the access using public IP address.
- 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).

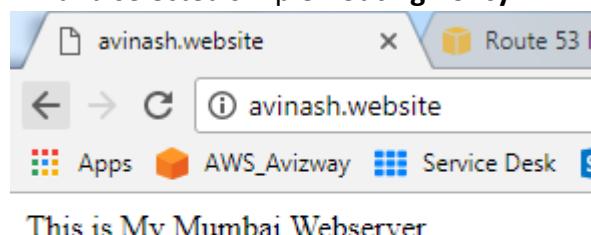


9. Now, we have two web servers in two different regions and we are going to configure routing policies between these two region resources.

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 set
- b. Choose Type as **A – IPV4 address**
- c. Select **Aliasrecord** 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**.



Create Record Set

Name: avinash.website.

Type: A – IPv4 address

Alias: Yes No

Alias Target: dualstack.IndiaELB-445439707.ap-sou

Alias Hosted Zone ID: ZP97RAFLXTNZK

You can also type the domain name for the resource. Examples:
 - CloudFront distribution domain name: d111111abcdef8.cloudfront.net
 - Elastic Beanstalk environment CNAME: example.elasticbeanstalk.com
 - ELB load balancer DNS name: example-1.us-east-1.elb.amazonaws.com
 - S3 website endpoint: s3-website.us-east-2.amazonaws.com
 - Resource record set in this hosted zone: www.example.com

[Learn More](#)

Routing Policy: Simple

Route 53 responds to queries based on the routing policy you specify in this record. [Learn More](#)

Evaluate Target

Simple
Weighted
Latency
Failover
Geolocation
Multivalue Answer

Create

- 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.

Alias: Yes No

Alias Target: dualstack.indiaelb-445439707.ap-sout

Alias Hosted Zone ID: ZP97RAFLXTNZK

You can also type the domain name for the resource. Examples:
 - CloudFront distribution domain name: d111111abcdef8.cloudfront.net
 - Elastic Beanstalk environment CNAME: example.elasticbeanstalk.com
 - ELB load balancer DNS name: example-1.us-east-1.elb.amazonaws.com
 - S3 website endpoint: s3-website.us-east-2.amazonaws.com
 - Resource record set in this hosted zone: www.example.com

[Learn More](#)

Routing Policy: Weighted

Route 53 responds to queries based on weighting that you specify in this and other record sets that have the same name and type. [Learn More](#)

Weight: 60

Set ID: Mumbai WebServer

Description of this record set that is unique within the group of weighted sets

Save Record Set

Alias: Yes No

Alias Target: `dualstack.USAAelb-461066.us-east-1.elb.amazonaws.com`

Alias Hosted Zone ID: Z35SXDOTRQ7X7K

You can also type the domain name for the resource. Examples:

- CloudFront distribution domain name: d111111abcdef8.cloudfront.net
- Elastic Beanstalk environment CNAME: example.elasticbeanstalk.com
- ELB load balancer DNS name: example-1.us-east-1.elb.amazonaws.com
- S3 website endpoint: s3-website.us-east-2.amazonaws.com
- Resource record set in this hosted zone: www.example.com

[Learn More](#)

Routing Policy: Weighted

Route 53 responds to queries based on weighting that you specify in this and other record sets with the same name and type. [Learn More](#)

Weight: 40

Set ID: USA WebServer

Description of this record set that is unique within the group of weighted sets.

Example:
My Seattle Data Center

Create

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.

Alias: Yes No

Alias Target: dualstack.IndiaELB-445439707.ap-southeast-1.elb.amazonaws.com

Alias Hosted Zone ID: ZP97RAFLXTNZK

You can also type the domain name for the resource. Examples:

- CloudFront distribution domain name: d111111abcdef8.cloudfront.net
- Elastic Beanstalk environment CNAME: example.elasticbeanstalk.com
- ELB load balancer DNS name: example-1.us-east-1.elb.amazonaws.com
- S3 website endpoint: s3-website.us-east-2.amazonaws.com
- Resource record set in this hosted zone: www.example.com

[Learn More](#)

Routing Policy: Latency

Route 53 responds to queries based on regions that you specify in this and other same name and type. [Learn More](#)

Region: ap-south-1

Set ID: Mumbai WebServer - latency

Description of this record set that is unique within the group of latency sets.

Example:
My Seattle Data Center

Create

Alias: Yes No

Alias Target: dualstack.USAAelb-461066.us-east-1.elb.amazonaws.com

Alias Hosted Zone ID: Z35SXDOTRQ7X7K

You can also type the domain name for the resource. Examples:

- CloudFront distribution domain name: d111111abcdef8.cloudfront.net
- Elastic Beanstalk environment CNAME: example.elasticbeanstalk.com
- ELB load balancer DNS name: example-1.us-east-1.elb.amazonaws.com
- S3 website endpoint: s3-website.us-east-2.amazonaws.com
- Resource record set in this hosted zone: www.example.com

[Learn More](#)

Routing Policy: Latency

Route 53 responds to queries based on regions that you specify in this and other record sets with the same name and type. [Learn More](#)

Region: us-east-1

Set ID: USA|WebServer - latency

Description of this record set that is unique within the group of latency sets.

Example:
My Seattle Data Center

Create

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.

Alias: Yes No

Alias Target:

Alias Hosted Zone ID: Z35SXDOTRQ7X7K

You can also type the domain name for the resource. Examples:

- CloudFront distribution domain name: d111111abcdef8.cloudfront.net
- Elastic Beanstalk environment CNAME: example.elasticbeanstalk.com
- ELB load balancer DNS name: example-1.us-east-1.elb.amazonaws.com
- S3 website endpoint: s3-website.us-east-2.amazonaws.com
- Resource record set in this hosted zone: www.example.com

[Learn More](#)

Routing Policy:

Route 53 responds to queries based on the locations from which DNS queries originate. You can create a Default location resource record set [Learn More](#)

Location:

Sublocation:

Set ID:

Description of this record set that is unique within the group of geolocation sets.

Example:

Create

Alias: Yes No

Alias Target: dualstack.IndiaELB-445439707.ap-southeast-1.elb.amazonaws.com

Alias Hosted Zone ID: ZP97RAFLXTNZK

You can also type the domain name for the resource. Examples:
- CloudFront distribution domain name: d111111abcdef8.cloudfront.net
- Elastic Beanstalk environment CNAME: example.elasticbeanstalk.com
- ELB load balancer DNS name: example-1.us-east-1.elb.amazonaws.com
- S3 website endpoint: s3-website.us-east-2.amazonaws.com
- Resource record set in this hosted zone: www.example.com

[Learn More](#)

Routing Policy: Geolocation ▾

Route 53 responds to queries based on the locations from which DNS queries originate. You can create a Default location resource record set. [Learn More](#)

Location: India

Set ID: India/entire world/users website

Description of this record set that is unique within the group of geolocation sets.
Example:
Route to Seattle data center

Create

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 determine its health status.

Specify endpoint by IP address

Protocol	HTTP
IP address	<input type="text"/>
Host name	<input type="text"/> www.example.com
Port *	<input type="text"/> 80
Path	<input type="text"/> /images

Alias: Yes No

Alias Target: dualstack.IndiaELB-445439707.ap-southeast-1.elb.amazonaws.com

Alias Hosted Zone ID: ZP97RAFLXTNZK

You can also type the domain name for the resource. Examples:

- CloudFront distribution domain name: d111111abcdef8.cloudfront.net
- Elastic Beanstalk environment CNAME: example.elasticbeanstalk.com
- ELB load balancer DNS name: example-1.us-east-1.elb.amazonaws.com
- S3 website endpoint: s3-website.us-east-2.amazonaws.com
- Resource record set in this hosted zone: www.example.com

[Learn More](#)

Routing Policy:

Route 53 responds to queries using primary record sets if any are healthy, or using secondary record sets otherwise. [Learn More](#)

Failover Record Type: Primary Secondary

Set ID: Primary

Evaluate Target Health: Yes No

Associate with Health Check: Yes No

Create

Alias: Yes No

Alias Target:

Alias Hosted Zone ID: Z35SXDOTRQ7X7K

You can also type the domain name for the resource. Examples:

- CloudFront distribution domain name: d111111abcdef8.cloudfront.net
- Elastic Beanstalk environment CNAME: example.elasticbeanstalk.com
- ELB load balancer DNS name: example-1.us-east-1.elb.amazonaws.com
- S3 website endpoint: s3-website.us-east-2.amazonaws.com
- Resource record set in this hosted zone: www.example.com

[Learn More](#)

Routing Policy:

Route 53 responds to queries using primary record sets if any are healthy, or using secondary record sets otherwise. [Learn More](#)

Failover Record Type: Primary Secondary

Set ID:

Evaluate Target Health: Yes No

Associate with Health Check: Yes No

Create

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.

Virtual Private Cloud (Amazon VPC)

The Amazon Virtual Private Cloud (Amazon VPC) is a custom-defined virtual network within the AWS Cloud. You can provision your own logically isolated section of AWS, similar to designing and implementing a separate independent network that would operate in an on-premises data center. Amazon VPC is the networking layer for Amazon Elastic Compute Cloud (Amazon EC2), and it 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 web servers 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 Internet gateway 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 limits by simply requesting 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)

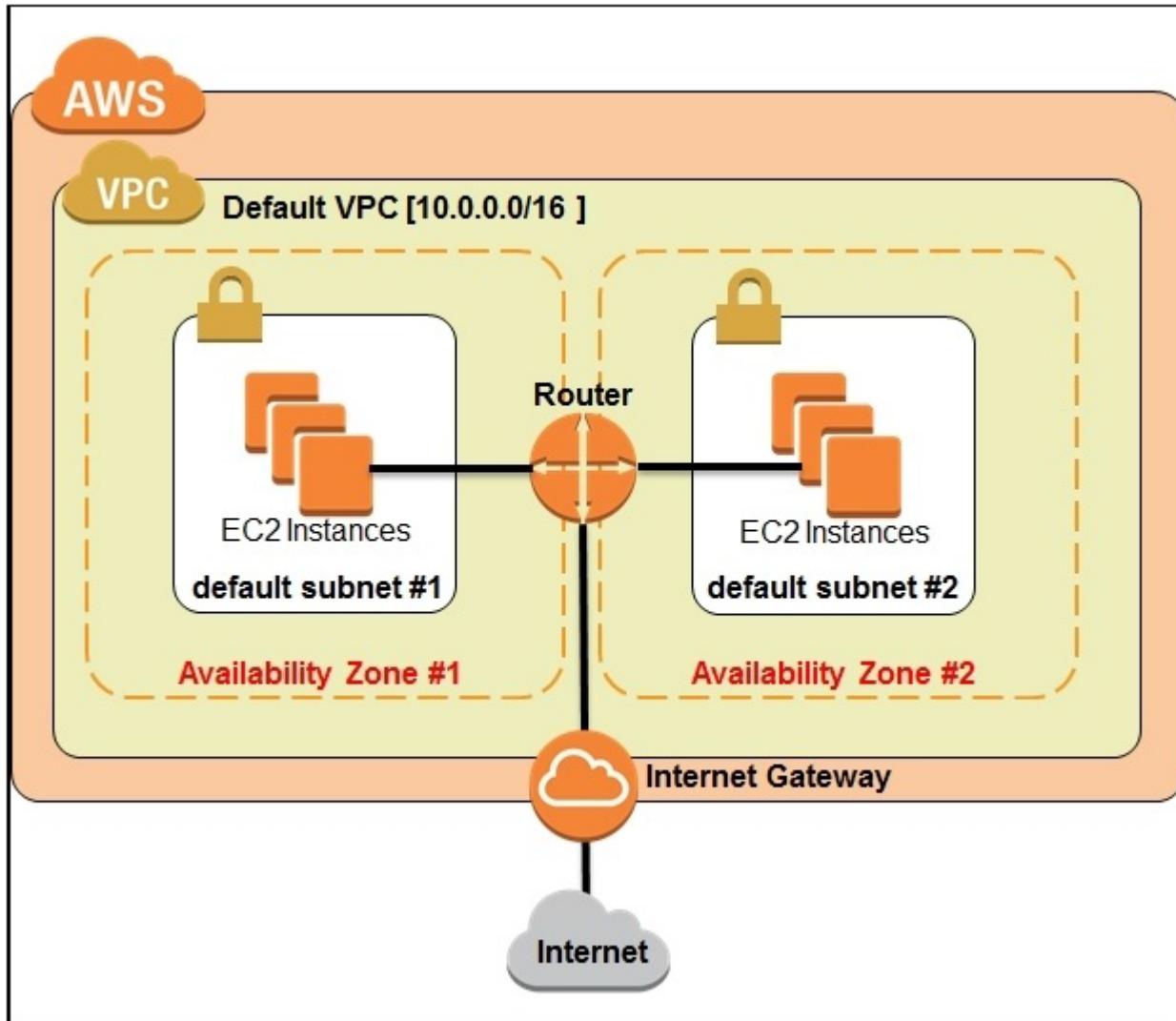
An Amazon VPC has the following optional components:

- 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 Private Networks (VPNs)

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

The default VPC is always created with a CIDR block of /16, which means it supports 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 Internet connectivity. A few necessary

route tables, security groups, and ACLs are also created by default. That enable the instance traffic to pass through to the Internet. Refer to the following figure:



Classless Inter-Domain Routing (CIDR): When you create an Amazon VPC, you must specify the IPv4 address range by choosing a Classless Inter-Domain Routing (CIDR) block, such as 10.0.0.0/16. The address range of the Amazon VPC cannot be changed after the Amazon VPC is created. An Amazon VPC address range may be as large as /16 (65,536 available addresses) or as small as /28 (16 available addresses) 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 Amazon EC2 instances, Amazon Relational Database Service (Amazon RDS) databases, and other AWS 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 the associated route table directs the subnet's traffic to the Amazon VPC's VPG and 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 is directed.

- 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 enables communication 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 subnet uses the main route table.
- You can replace the main route table with a custom table that you've created so that each new subnet is automatically associated with it.

Internet Gateways:

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

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 to the instance's public IP address (or EIP address, covered later) and maintains the one-to-one map of the instance private IP address and public IP address.

When an instance receives traffic from the Internet, the IGW translates the destination address (public IP address) to the 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 Address (EIP) is a static, public IP address in the pool for the region that you can allocate to your account (pull from the pool) and release (return to the pool).

AWS maintains a pool of public IP addresses in each region and makes them available for you to 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 VPCs that enables instances in either Amazon VPC to communicate with each other as if they are within the same network. You can create an Amazon VPC peering connection between your own 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-one relationship between Amazon VPCs, meaning two Amazon VPCs cannot have two peering agreements between them.

Peering connections are created through a request/accept protocol. The owner of the requesting Amazon VPC sends a request to peer to the owner of the peer Amazon VPC. If the peer Amazon VPC is within the same account, it is identified by its VPC ID. If the peer VPC is within a different account, it is identified by Account ID and VPC ID. The owner of the peer Amazon VPC has one week to accept or reject the request to peer with the requesting Amazon VPC 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 of any subnet associated with the network ACL. Here is a small example of how ACL looks like.

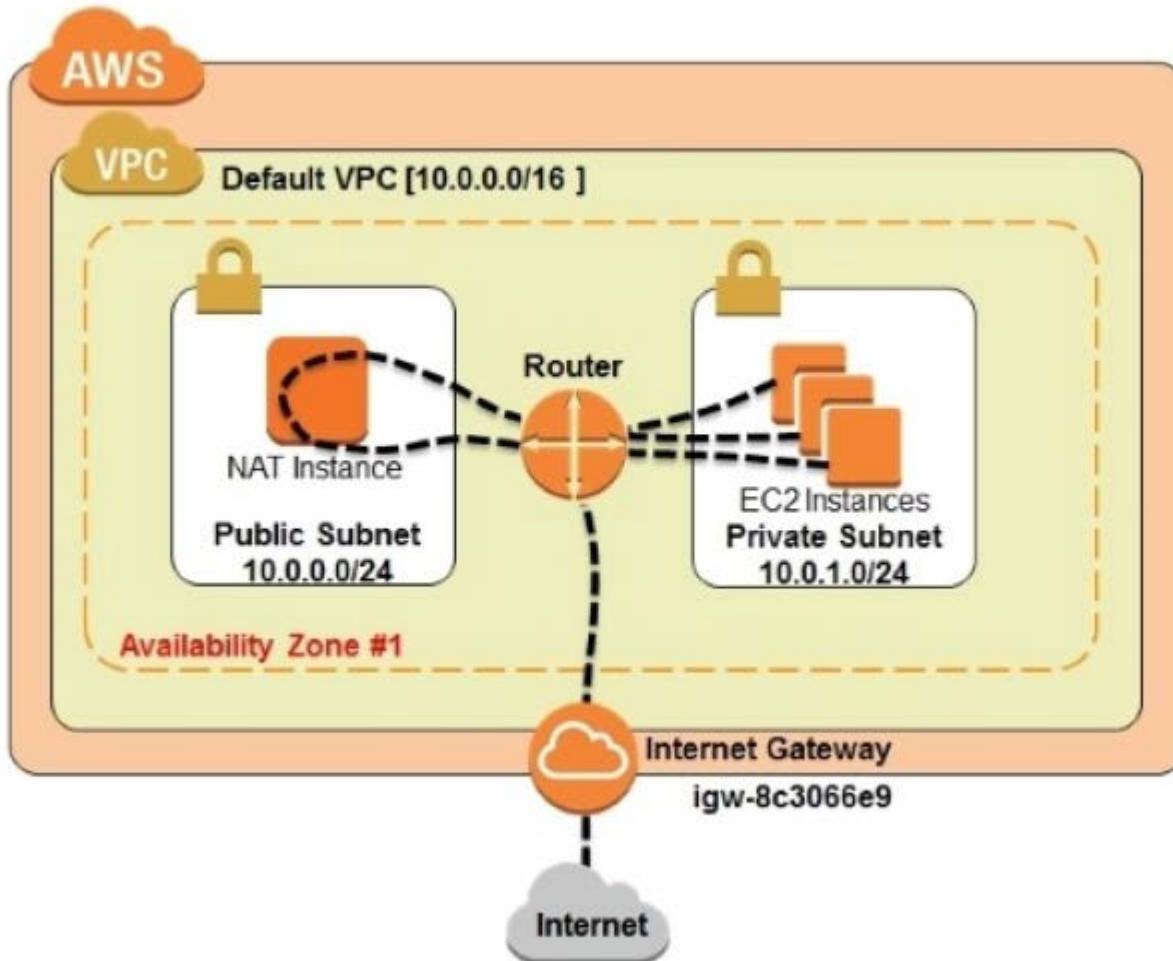
Inbound ACL rules				
Rule No.	Source IP	Protocol	Port	Allow/Deny
100	0.0.0.0/0	All	All	ALLOW
*	0.0.0.0/0	All	All	DENY
Outbound ACL rules				
Rule No.	Dest IP	Protocol	Port	Allow/Deny
100	0.0.0.0/0	all	all	ALLOW
*	0.0.0.0/0	all	all	DENY

When you create a custom network ACL, its initial configuration will deny all inbound and 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 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.

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 able to communicate with the Internet through the IGW. AWS provides NAT instances and NAT gateways to allow instances deployed in private subnets to gain Internet access.



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 the source IP address to the public IP address of the NAT instance, and forward the traffic to the IGW. NAT Instances allows in private subnets to send outbound Internet communication, but it prevents the instances from receiving inbound traffic initiated by someone 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.

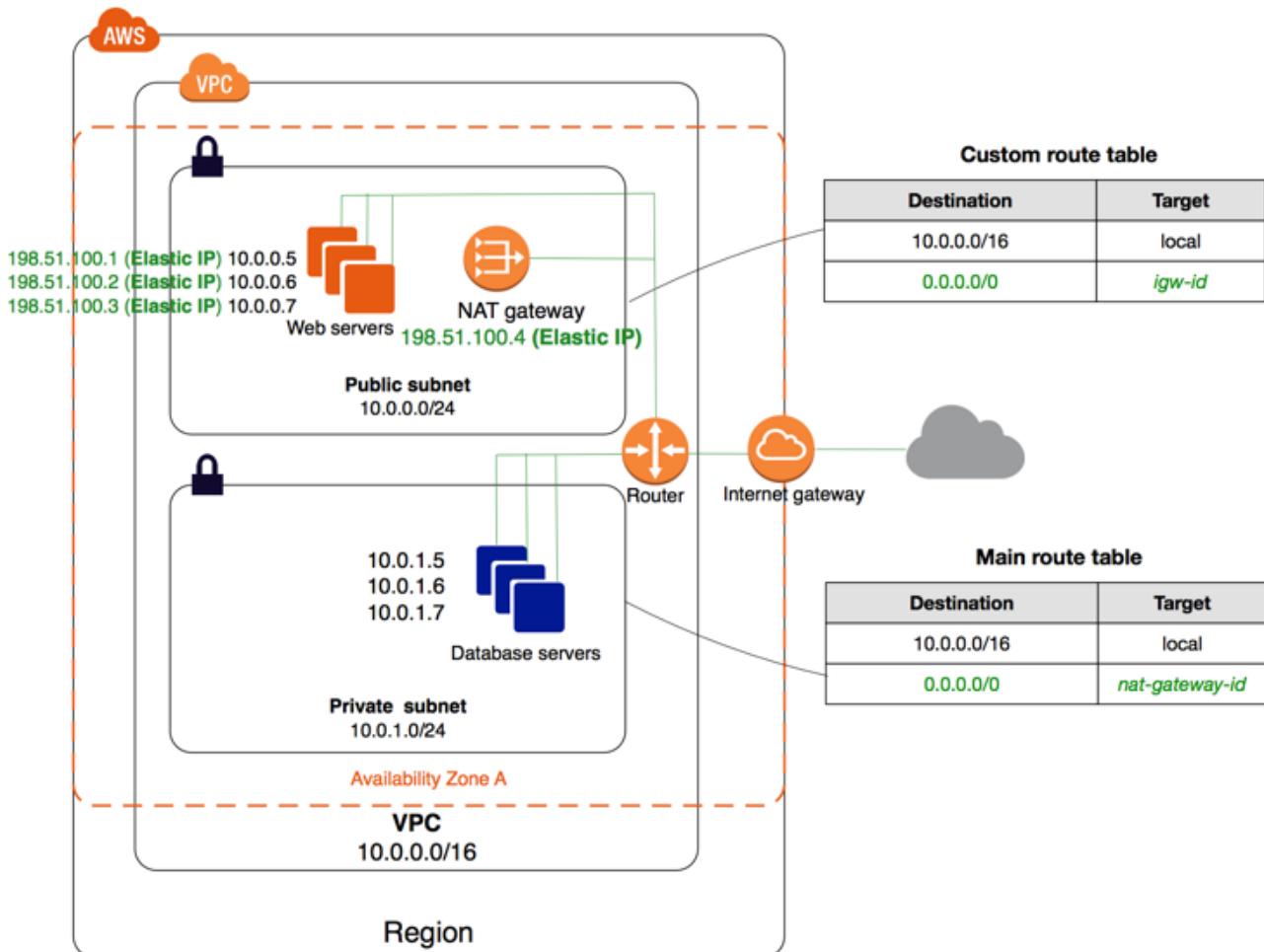
- Allocate an EIP and associate it with the NAT gateway.
- Configure the route table associated with the private subnet to direct Internet-bound traffic to the NAT gateway.

You can connect an existing data center to Amazon VPC using either hardware or software VPN connections, which will make Amazon VPC an extension of the data center. Amazon VPC offers 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 the AWS side of the VPN connection between the two networks.

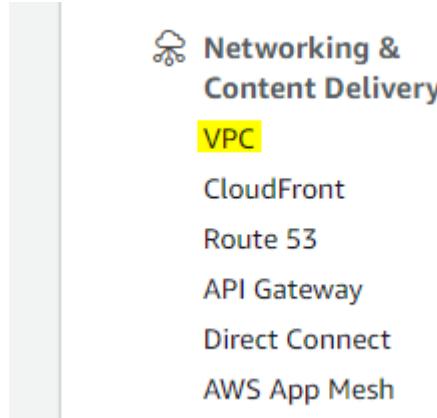
A customer gateway (CGW) represents a physical device or a software application on the customer's side of the VPN connection.

Here is the VPC diagram we are about to deploy with Public and private Subnets including NAT:



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.

Resources

Start VPC Wizard

Launch EC2 Instances

Note: Your Instances will launch in the Asia Pacific (Mumbai) region.

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

VPC with a Single Public Subnet

VPC with Public and Private Subnets

VPC with Public and Private Subnets and Hardware VPN Access

VPC with a Private Subnet Only and Hardware VPN Access

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 deployment scenario is very much similar to the VPC with public and private subnets, however, with one component added additionally, which is the Virtual Private Gateway. This Virtual Private Gateway connects to your on-premise network's gateway using a standard VPN connection. This type of deployment is well suited for organizations that wish to extend their on-premise datacenters and networks into the public clouds while allowing their instances to communicate with the Internet.

VPC with a private subnet only and hardware VPN access: Unlike the previous deployment scenario, this scenario only provides you with a private subnet that can connect to your on-premise datacenters using standard VPN connections. There is no Internet Gateway provided and thus your instances remain isolated from the Internet. This deployment scenario is ideal for cases where you wish to extend your on-premise datacenters into the public cloud but do not wish your instances to have any communication with the outside world.

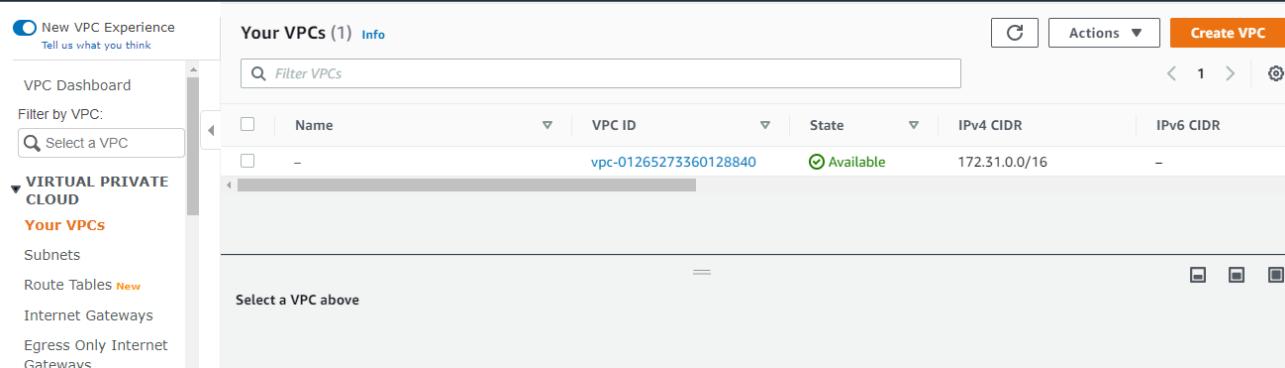
Here is a simple use case for creating Custom VPC

- 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: To create a custom VPC Navigate to “Your VPCs” from left pane, and Click on “Create VPC”.



VPC settings

Name tag - *optional*
Creates a tag with a key of 'Name' and a value that you specify.

CustomVPC

IPv4 CIDR block [Info](#)
192.168.0.0/16

IPv6 CIDR block [Info](#)
 No IPv6 CIDR block
 Amazon-provided IPv6 CIDR block
 IPv6 CIDR owned by me

Tenancy [Info](#)
Default

Tags

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Key	Value - <i>optional</i>	
<input type="text"/> Name X	<input type="text"/> CustomVPC X	Remove

[Add new tag](#)
You can add 49 more tags.

[Cancel](#) Create 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 us 65,531 IP addresses to use) and selecting tenancy as **Default**.

Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR
<input checked="" type="checkbox"/> CustomVPC	vpc-0a242371c74173265	Available	192.168.0.0/16	-
<input type="checkbox"/> -	vpc-01265273360128840	Available	172.31.0.0/16	-

Details				
VPC ID <input checked="" type="checkbox"/> vpc-0a242371c74173265	State Available	DNS hostnames Disabled	DNS resolution Enabled	
Tenancy Default	DHCP options set dopt-18369f73	Main route table rtb-004489847c61c5fc8	Main network ACL acl-058b85f91f40de034	
Default VPC No	IPv4 CIDR 192.168.0.0/16	IPv6 pool -	IPv6 CIDR -	

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

- 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.

Create subnet Info

VPC

VPC ID
Create subnets in this VPC.
vpc-0a242371c74173265 (CustomVPC)

Associated VPC CIDRs
IPv4 CIDRs
192.168.0.0/16

Subnet settings

Specify the CIDR blocks and Availability Zone for the subnet.

Subnet 1 of 1

Subnet name
Create a tag with a key of 'Name' and a value that you specify.

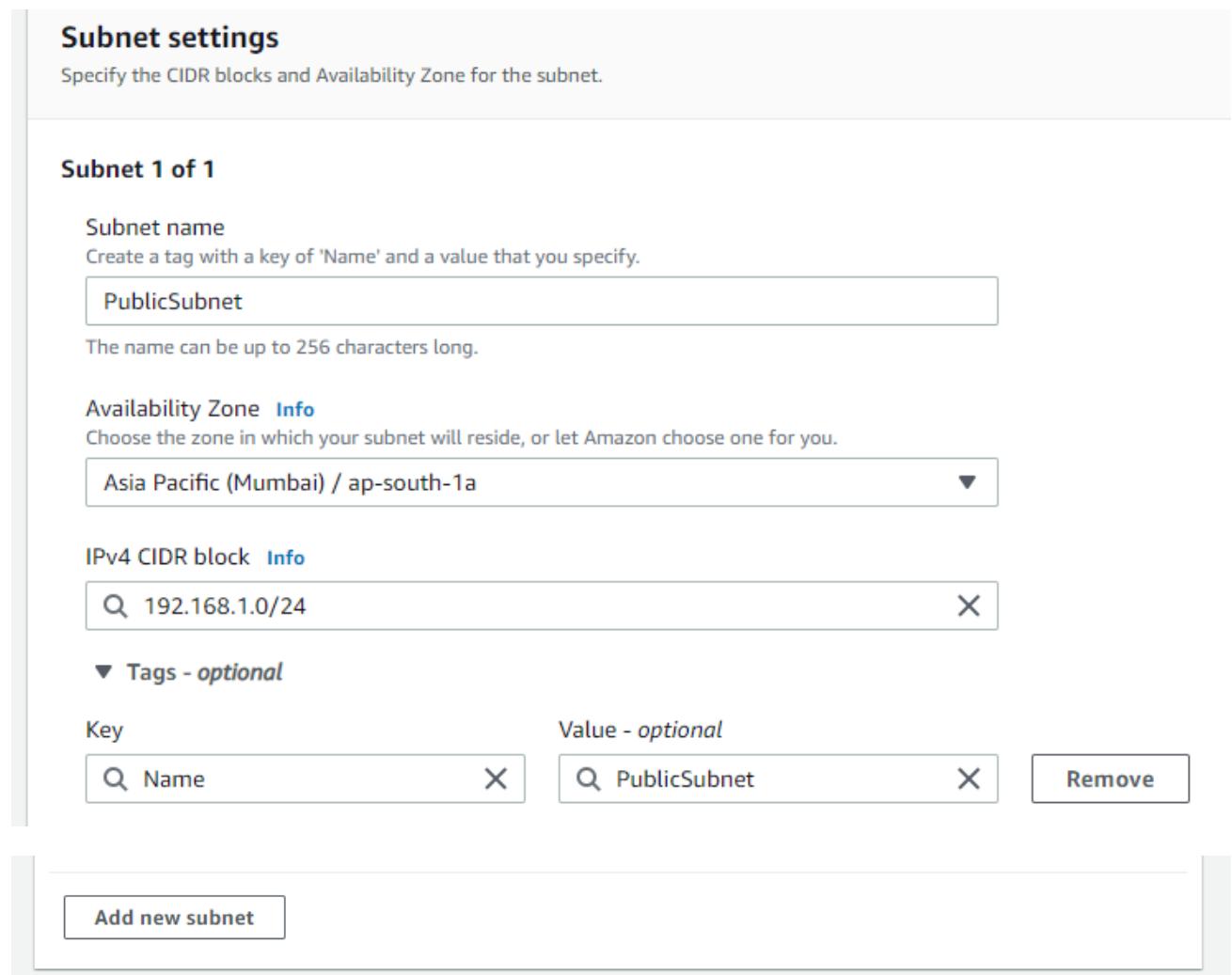
The name can be up to 256 characters long.

Availability Zone Info
Choose the zone in which your subnet will reside, or let Amazon choose one for you.

IPv4 CIDR block Info

▼ Tags - optional

Key	Value - optional	Remove
<input type="text" value="Name"/>	<input type="text" value="PublicSubnet"/>	<input type="button" value="Remove"/>



- Now creating another subnet by clicking “**Add new 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 Availability Zone as **ap-south-1b** and click on Create option.

Subnet 2 of 2

Create a tag with a key of 'Name' and a value that you specify.

PrivateSubnet

The name can be up to 256 characters long.

Availability Zone [Info](#)

Choose the zone in which your subnet will reside, or let Amazon choose one for you.

Asia Pacific (Mumbai) / ap-south-1b

IPv4 CIDR block [Info](#)

192.168.2.0/24

▼ Tags - optional

Key

Value - optional

Name

PrivateSubnet

Remove

Add new tag

You can add 49 more tags.

- This is how exactly subnet dashboard looks like now.

Subnets (5) Info						Actions	Create subnet
	Name	Subnet ID	State	VPC	IPv4 CIDR		
<input type="checkbox"/>	PublicSubnet	subnet-0afcd8235e7bbba01	Available	vpc-0a242371c74173265 Cu...	192.168.1.0/24		
<input type="checkbox"/>	PrivateSubnet	subnet-03d4485e523444252	Available	vpc-0a242371c74173265 Cu...	192.168.2.0/24		
<input type="checkbox"/>	-	subnet-06be8837d1e387cd8	Available	vpc-01265273360128840	172.31.16.0/20		

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.

Internet gateway settings

Name tag
Creates a tag with a key of 'Name' and a value that you specify.

IGWforCustomVPC

Tags - optional
A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Key	Value - optional
<input type="text" value="Name"/>	<input type="text" value="IGWforCustomVPC"/> X

Add new tag

You can add 49 more tags.

Cancel Create internet gateway

- Internet Gateway is now in “**Detached**” mode, Select the VPC and navigate to “Actions” and click on “**Attach to VPC**” option and select the Custom VPC and click on “**Yes, Attach**” option.

Internet gateways (1/2) Info

C Actions ▾ Create internet gateway

View details Attach to VPC Detach from VPC Manage tags Delete internet gateway

-	Name	Internet gateway ID	State	VPC ID
<input checked="" type="checkbox"/>	IGWforCustomVPC	igw-018e43bf7761459cb	Detached	-
<input type="checkbox"/>	-	igw-0b07278d1bb3aae65	Attached	vpc-01

- 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.

Attach to VPC (igw-018e43bf7761459cb) Info

VPC

Attach an internet gateway to a VPC to enable the VPC to communicate with the internet. Specify the VPC to attach below.

Available VPCs

Attach the internet gateway to this VPC.

Select a VPC

vpc-0a242371c74173265 - CustomVPC

▶ AWS Command Line Interface command

vpc-0a242371c74173265 - CustomVPC

Cancel

Attach internet gateway

Internet gateways (1/2) Info

C

Actions ▾

Create internet gateway

< 1 >

Name	Internet gateway ID	State	VPC ID	Owner
<input checked="" type="checkbox"/> IGWforCustomVPC	igw-018e43bf7761459cb	Attached	vpc-0a242371c74173265 CustomVPC	501170964283
<input type="checkbox"/> -	igw-0b07278d1bb3aae65	Attached	vpc-01265273360128840	501170964283

igw-018e43bf7761459cb / IGWforCustomVPC

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.

Route tables (2) Info

C

Actions ▾

Create route table

< 1 >

Name	Route table ID	Explicit subnet associat...	Edge associations	Main	VPC
<input type="checkbox"/> -	rtb-004489847c61c5fc8	-	-	Yes	vpc-0a242371c74173265
<input type="checkbox"/> -	rtb-0e24081c554219dd0	-	-	Yes	vpc-01265273360128840

Create route table Info

A route table specifies how packets are forwarded between the subnets within your VPC, the internet, and your VPN connection.

Route table settings

Name - optional

Create a tag with a key of 'Name' and a value that you specify.

VPC

The VPC to use for this route table.

- 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.

Route tables (1/3) Info

<input type="checkbox"/>	-	rtb-004489847c61c5fc8	-	-	Yes	vpc-0a242371c741
<input type="checkbox"/>	-	rtb-0e24081c554219dd0	-	-	Yes	vpc-012652733601
<input checked="" type="checkbox"/>	CustomRoute	rtb-0e74b13d1e77109f4	-	-	No	vpc-0a242371c741

Details Routes Subnet associations Edge associations Route propagation Tags

Routes (1)

<small>Filter routes</small>					<small>Both</small>	<small>< 1 ></small>	<small>⚙️</small>
Destination	Target	Status	Propagated				
192.168.0.0/16	local	Active	No				

- Now we have to add a route by selecting edit option and select “**Add 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 changes**.

Edit routes

Destination	Target	Status	Propagated
192.168.0.0/16	local	Active	No
<input type="text" value="0.0.0.0/0"/>	<input type="text" value="igw-018e43bf7761459cb (IGWforCustomVPC)"/>	-	No
<input type="button" value="Add route"/>	<input type="button" value="igw-018e43bf7761459cb (IGWforCustomVPC)"/>		

Cancel Preview Save changes

- Then select the “**Subnet Association**” and click on “Edit” option and select the “**Public Subnet**” and click on save.

The screenshot shows the AWS Route Tables interface with the "Subnet associations" tab selected. The main area displays "Explicit subnet associations (0)". A search bar at the top says "Find subnet association". Below it are dropdown menus for "Subnet ID", "IPv4 CIDR", and "IPv6 CIDR". A message in the center says "No subnet associations" and "You do not have any subnet associations." There is a "Edit subnet associations" button in the top right corner.

Edit subnet associations

Change which subnets are associated with this route table.

The screenshot shows the "Edit subnet associations" dialog. Under "Available subnets (1/2)", there is a table with columns: Name, Subnet ID, IPv4 CIDR, IPv6 CIDR, and Route table ID. It lists "PrivateSubnet" and "PublicSubnet". The "PublicSubnet" row has a checked checkbox. Under "Selected subnets", the "PublicSubnet" row is listed with the ID "subnet-0afcd8235e7bbaa01 / PublicSubnet". At the bottom are "Cancel" and "Save associations" buttons.

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 “**Actions**” and choose “**Modify auto-assign IP settings**”, select the checkbox and click on save.

The screenshot shows the AWS Subnets dashboard. A public subnet named "PublicSubnet" is selected. A context menu is open over the subnet row, with "Modify auto-assign IP settings" highlighted. Other options in the menu include "View details", "Create flow log", "Edit IPv6 CIDRs", "Edit network ACL association", "Edit route table association", "Share subnet", "Manage tags", and "Delete subnet".

Modify auto-assign IP settings Info

Enable the auto-assign IP address setting to automatically request a public IPv4 or IPv6 address for a new network interface in this subnet.

Settings

Subnet ID

subnet-0afcd8235e7bbaa01

Auto-assign IPv4 Info

Enable auto-assign public IPv4 address

Auto-assign customer-owned IPv4 address Info

Enable auto-assign customer-owned IPv4 address
Option disabled because no customer owned pools found.

[Cancel](#)

[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”.

Network <small>i</small>	<input type="text" value="vpc-8b6984e3 Custom VPC"/>	<input type="button" value="C"/> Create new VPC
Subnet <small>i</small>	<input type="text" value="subnet-fbae5a93 Public Subnet ap-south-1a"/>	<input type="button" value="Create new subnet"/>
Auto-assign Public IP <small>i</small>	<input type="text" value="Use subnet setting (Enable)"/>	

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.

Assign a security group: Create a new security group

Select an existing security group

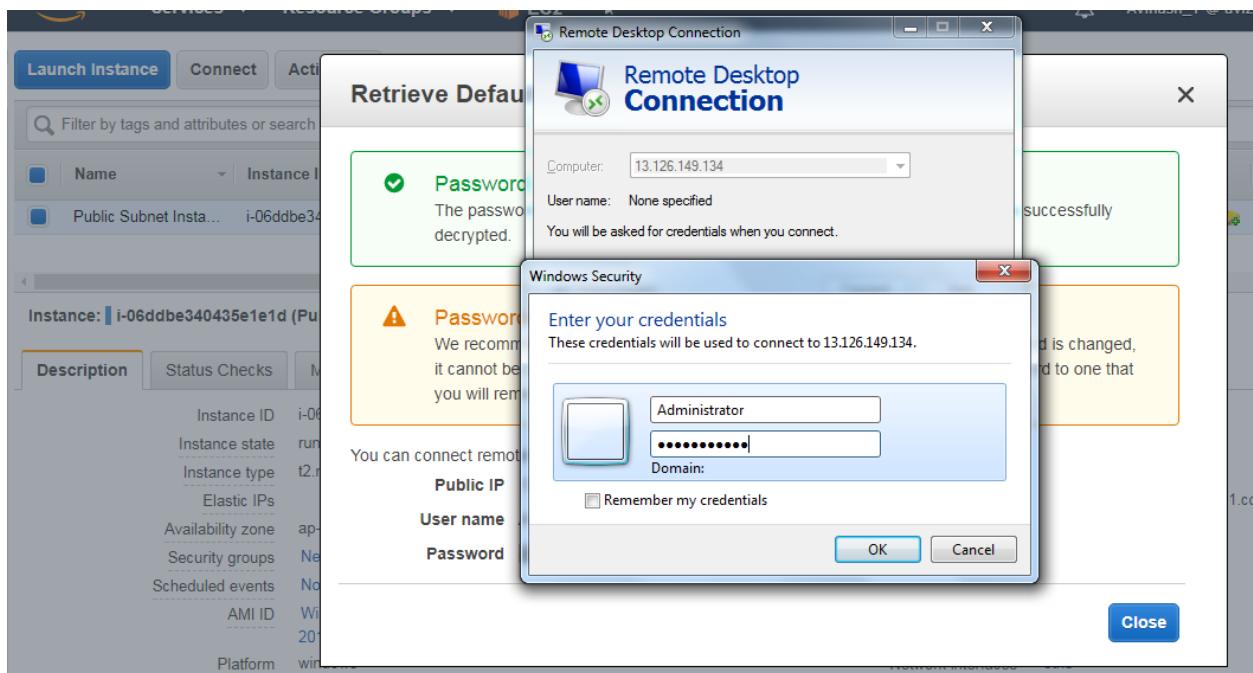
Security group name:

Description:

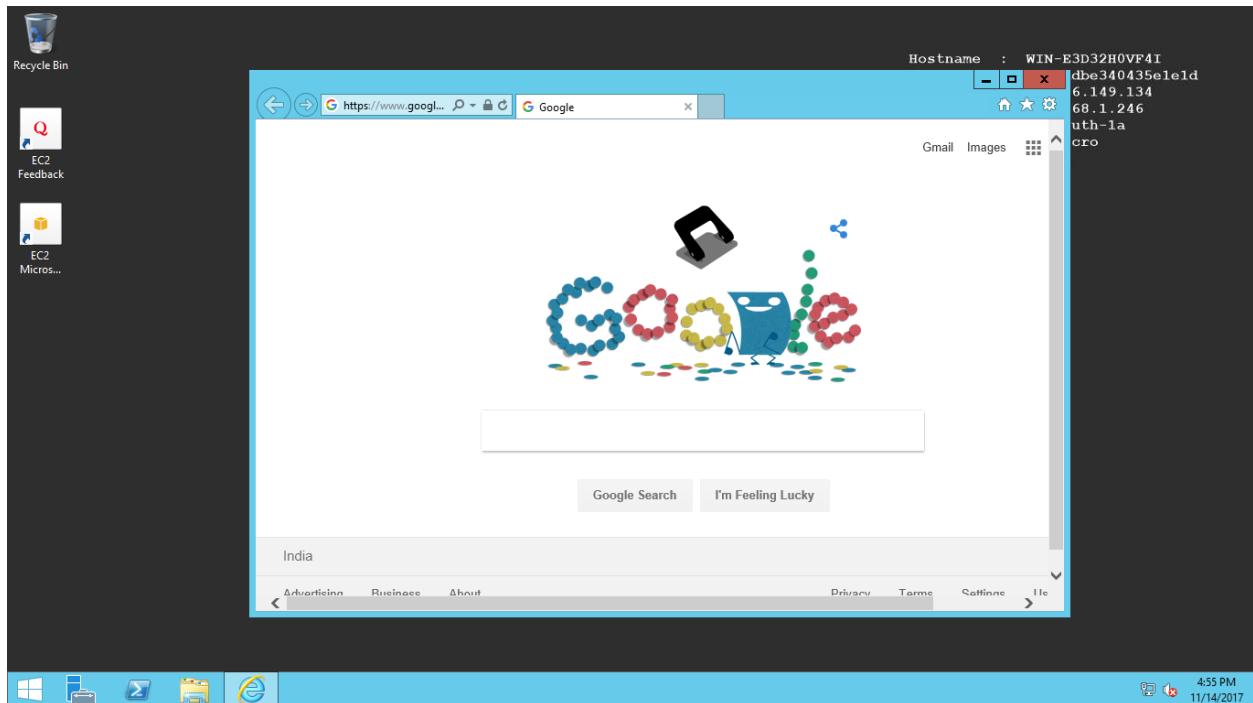
Type <small>i</small>	Protocol <small>i</small>	Port Range <small>i</small>	Source <small>i</small>
RDP	TCP	3389	Custom <input type="button"/> 0.0.0.0/0
HTTP	TCP	80	Custom <input type="button"/> 0.0.0.0/0, ::/0
HTTPS	TCP	443	Custom <input type="button"/> 0.0.0.0/0, ::/0
SSH	TCP	22	Anywhere <input type="button"/> 0.0.0.0/0, ::/0

[Add Rule](#)

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.



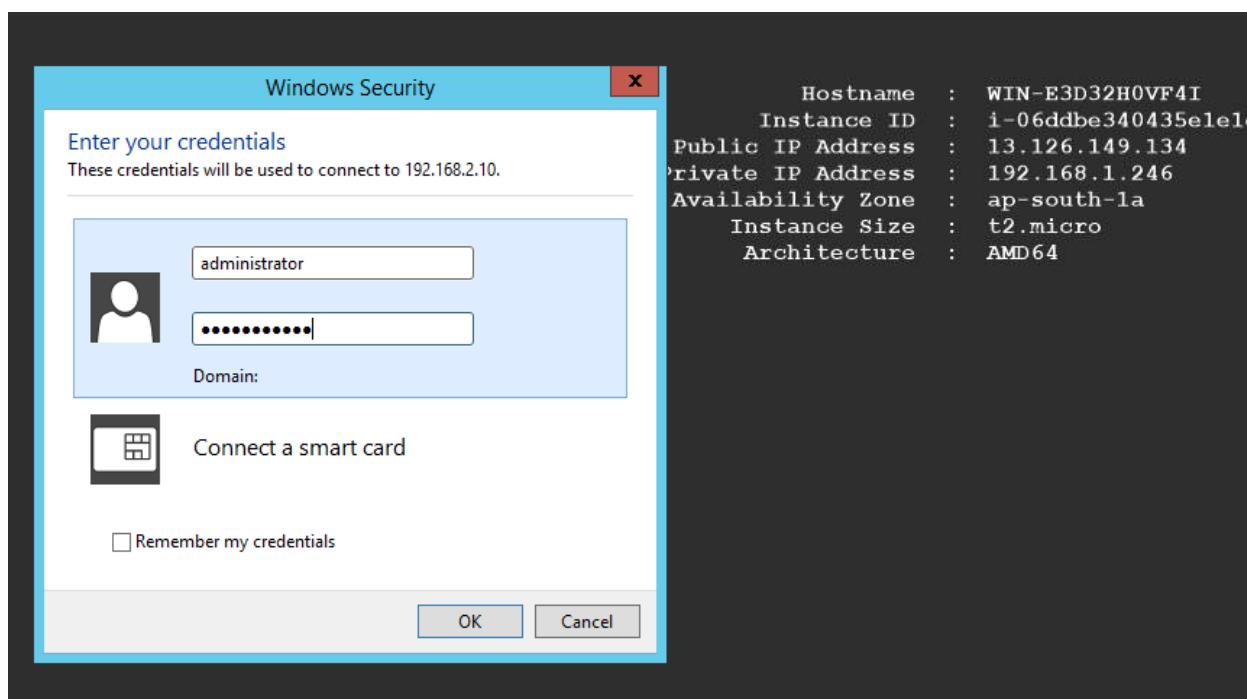
And we have successfully connected to the Instance, That means this instance is internet-faced and we can access anywhere from the world.

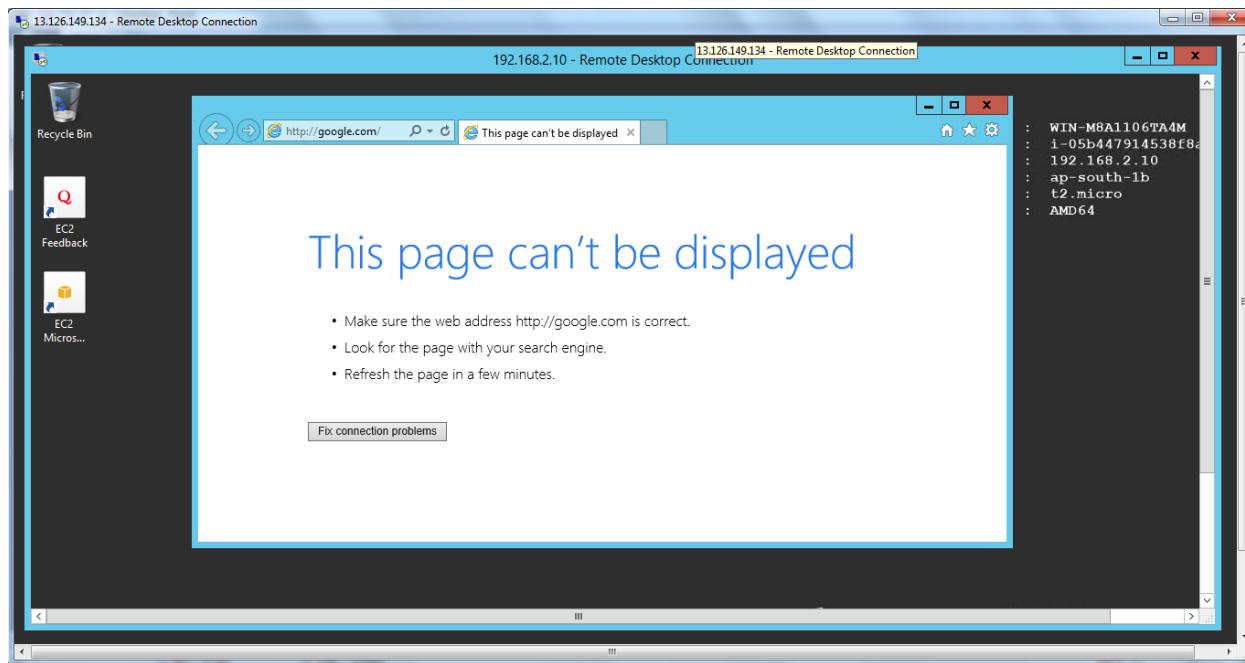


4. Now Launching another Instance in “Custom VPC” and selected to launch under “Private Subnet”.



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.
- But we can connect to the same instance from the Public Subnets launched Instance.
 - 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.

Step 1: Choose an Amazon Machine Image (AMI)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. You can select an AMI provided by AWS, our user community, or the AWS Marketplace; or you can select one of your own AMIs.

- 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 andHttps.

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

	Name	Instance ID	Instance Type	Availability Zone	Instance State
<input type="checkbox"/>	Private Subnet Inst...	i-05b447914538f8a01	t2.micro	ap-south-1b	running
<input type="checkbox"/>	Public Subnet Insta...	i-06ddbe340435e1e...	t2.micro	ap-south-1a	running
<input checked="" type="checkbox"/>	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”.

The screenshot shows the AWS EC2 Instances page. A NAT instance is selected. The 'Actions' dropdown menu is open, and the 'Networking' option is highlighted. A sub-menu under 'Networking' includes 'Change Source/Dest. Check'. Other options in the sub-menu include 'Change Security Groups', 'Attach Network Interface', 'Detach Network Interface', 'Disassociate Elastic IP Address', and 'Manage IP Addresses'.

Enable Source/Destination Check

×

Are you sure that you would like to disable Source/Destination Check for the instance with the following details:

Instance:	i-0fd9269e5a439471b (NAT Instance)
Network Interface:	eni-d3fb4a8c
Status	Enabled

[Cancel](#) [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.

Search Route Tables and their Details				
	Name	Route Table ID	Explicitly Associated	Main
<input checked="" type="checkbox"/>	rtb-34e62c5c		0 Subnets	Yes
<input type="checkbox"/>	CustomRoute	rtb-91f933f9	1 Subnet	No
<input type="checkbox"/>		rtb-ab4491c2	0 Subnets	Yes

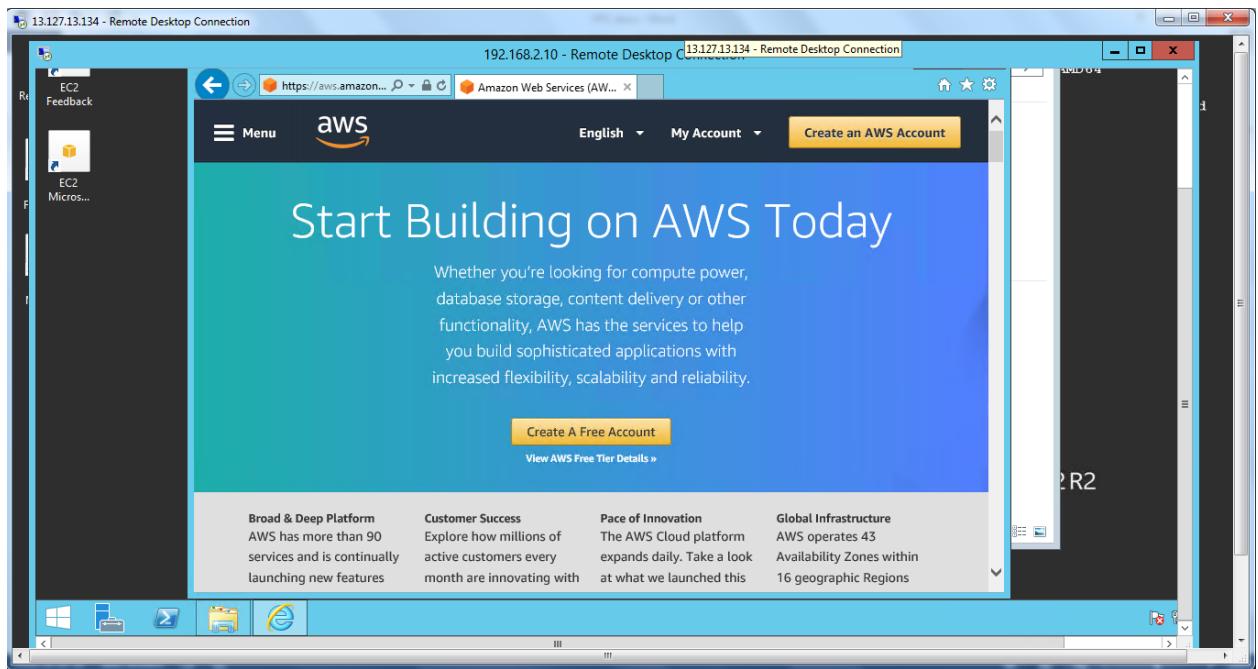
rtb-34e62c5c

Summary	Routes	Subnet Associations	Route Propagation	Tags
	Edit			
	View: All rules			
Destination	Target	Status	Propagated	
192.168.0.0/16	local	Active	No	

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

rtb-34e62c5c				
	Summary	Routes	Subnet Associations	Route Propagation
		Edit		
		View: All rules		
Destination	Target	Status	Propagated	
192.168.0.0/16	local	Active	No	
0.0.0.0/0	eni-d3fb4a8c / i-0fd9269e5a439471b	Active	No	

- 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**.

NAT gateway settings

Name - optional
Create a tag with a key of 'Name' and a value that you specify.

The name can be up to 256 characters long.

Subnet
Select a subnet in which to create the NAT gateway.

Connectivity type
Select a connectivity type for the NAT gateway.

Public
 Private

Elastic IP allocation ID [Info](#)
Assign an Elastic IP address to the NAT gateway.

[Allocate Elastic IP](#)

- Here is the NAT Gateway information after creation.

⌚ NAT gateway nat-08a3ced88d4a773b4 | NATGateway was created successfully. X

VPC > NAT gateways > nat-08a3ced88d4a773b4

nat-08a3ced88d4a773b4 / NATGateway

[Delete](#)

Details Info			
NAT gateway ID nat-08a3ced88d4a773b4	Connectivity type Public	State Pending	State message Info -
Elastic IP address -	Private IP address -	Network interface ID -	VPC vpc-0a242371c74173265 / CustomVPC
Subnet subnet-0afcd8235e7bbaa01 / PublicSubnet	Created 2021/07/12 19:25 GMT+5:30	Deleted -	

- 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**.

Route tables (1/3) [Info](#)

[rtb-004489847c61c5fc8](#) [Edit](#) [Actions](#) [Create route table](#)

Filter route tables

Name	Route table ID	Explicit subnet associations	Edge associations	Main	VPC
-	rtb-004489847c61c5fc8	-	-	Yes	vpc-0a242371c741'

Details [Routes](#) Subnet associations Edge associations Route propagation Tags

Routes (1) [Edit routes](#)

Filter routes Both

Destination	Target	Status	Propagated
192.168.0.0/16	local	Active	No

VPC > Route tables > rtb-004489847c61c5fc8 > Edit routes

Edit routes

Destination	Target	Status	Propagated
192.168.0.0/16	local	Active	No
0.0.0.0/0	nat- nat-08a3ced88d4a773b4 (NATGateway)	-	No

Add route

Route table ID: rtb-004489847c61c5fc8 Main: Yes Explicit subnet associations: - Edge associations: -

VPC: vpc-0a242371c74173265 | CustomVPC Owner ID: 501170964283

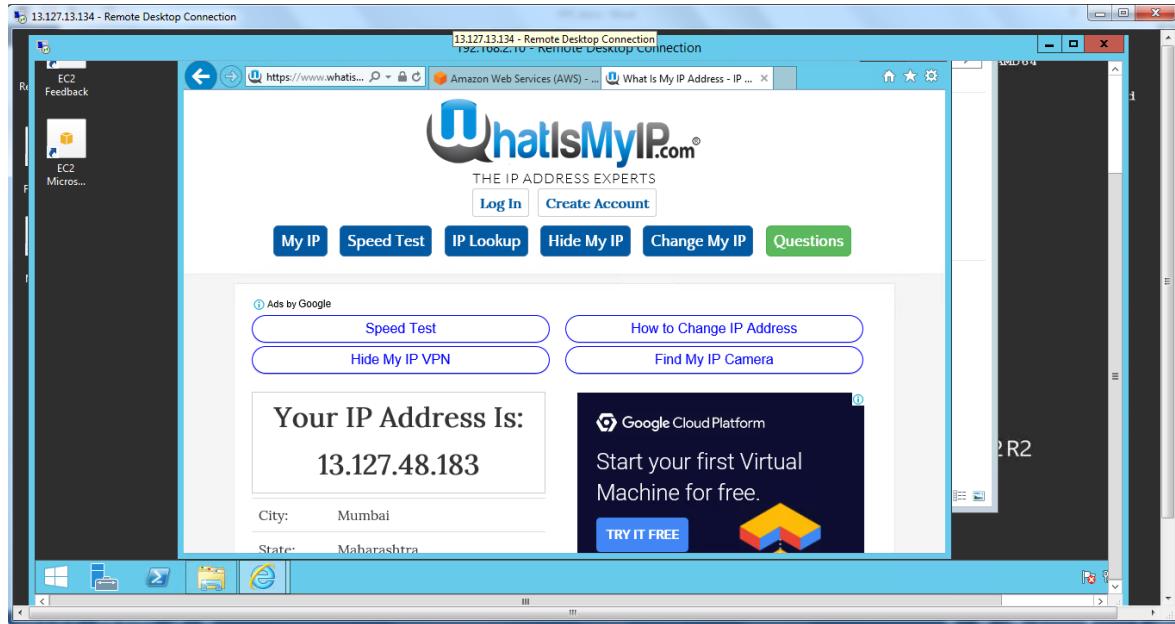
Routes (2) [Edit routes](#)

Filter routes Both

Destination	Target	Status	Propagated
192.168.0.0/16	local	Active	No
0.0.0.0/0	nat-08a3ced88d4a773b4	Active	No

- Now go to private subnet instance and verify the internet connectivity. You will be 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 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 of any subnet associated with the network ACL.

Every subnet must be associated with a network ACL.

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.

Create network ACL [Info](#)

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

Network ACL settings

Name - optional

Creates a tag with a key of 'Name' and a value that you specify.

VPC

VPC to use for this network ACL.

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

Network ACLs (1/3) Info						Actions	Create network ACL	
<input type="text"/> Filter network ACLs						Actions	Create network ACL	
-	Name	Network ACL ID	Associated with	Default	VPC ID			
<input checked="" type="checkbox"/>	CUstomNACL	acl-028930a6f92ac1c82	-	No	vpc-0a242371c74173265	Edit	Delete	Details
<input type="checkbox"/>	-	acl-028930a6f92ac1c82	2 Subnets	Yes	vpc-0a242371c74173265	Edit	Delete	Details

acl-028930a6f92ac1c82 / CUstomNACL

[Details](#) [Inbound rules](#) [Outbound rules](#) [Subnet associations](#) [Tags](#)

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

Edit subnet associations [Info](#)

Change which subnets are associated with this network ACL.

Available subnets (1/2)

Available subnets (1/2)						Actions	Create subnet association
-	Name	Subnet ID	Associated with	Availability Zone	IPv4 CIDR	IPv6 CIDR	
<input type="checkbox"/>	PrivateSubnet	subnet-03d4485e523444252	acl-058b85f91f40de034	ap-south-1b	192.168.2.0/24	-	Edit
<input checked="" type="checkbox"/>	PublicSubnet	subnet-0afcd8235e7bbbaa01	acl-058b85f91f40de034	ap-south-1a	192.168.1.0/24	-	Edit

Selected subnets

subnet-0afcd8235e7bbbaa01 / PublicSubnet [X](#)

[Cancel](#)
[Save changes](#)

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

Name	Network ACL ID	Associated with	Default	VPC ID
CUstomNACL	acl-028930a6f92ac1c82	subnet-0afcd8235e7bbaa01 / PublicSubnet	No	vpc-0a242371c7417321
-	acl-058h85fq1f1n1o1z1	subnet-02d11185a521111252 / PrivateSubnet	Yes	vpc-0a242371c7417321

acl-028930a6f92ac1c82 / CUstomNACL

Details **Inbound rules** Outbound rules Subnet associations Tags

Inbound rules (1)

Filter inbound rules

Rule number	Type	Protocol	Port range	Source	Allow/Deny
*	All traffic	All	All	0.0.0.0	Deny

Edit inbound rules

- 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.

Rule #	Type	Protocol	Port Range	Source	Allow / Deny	Remove
100	HTTP (80)	TCP (6)	80	0.0.0.0/0	ALLOW	
200	SSH (22)	TCP (6)	22	0.0.0.0/0	ALLOW	
300	RDP (3389)	TCP (6)	3389	0.0.0.0/0	ALLOW	
400	Custom TCP Rule	TCP (6)	1025-65535	0.0.0.0/0	ALLOW	

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

Rule #	Type	Protocol	Port Range	Destination	Allow / Deny
100	HTTP (80)	TCP (6)	80	0.0.0.0/0	ALLOW
200	SSH (22)	TCP (6)	22	0.0.0.0/0	ALLOW
300	RDP (3389)	TCP (6)	3389	0.0.0.0/0	ALLOW
400	Custom TCP Rule	TCP (6)	1025-65535	0.0.0.0/0	ALLOW
*	All Traffic	ALL	ALL	0.0.0.0/0	DENY

- 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.

Your VPCs (1/2) [Info](#)

Filter VPCs

Name	VPC ID	State	IPv4 CIDR
<input checked="" type="checkbox"/> CustomVPC	vpc-0a242371c74173265	Available	192.168.0.0/16
<input type="checkbox"/>	vpc-01265273360128840	Available	172.31.0.0/16

vpc-0a242371c74173265 / CustomVPC

[Details](#) | [CIDRs](#) | [Flow logs](#) | [Tags](#)

Actions ▲ [Create VPC](#)

- [Create default VPC](#)
- [Create flow log](#)
- [Edit CIDRs](#)
- [Edit DHCP options set](#)
- [Edit DNS hostnames](#)
- [Edit DNS resolution](#)
- [Manage tags](#)
- [Delete VPC](#)

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.

Create flow log [Info](#)

Flow logs can capture IP traffic flow information for the network interfaces associated with your resources. You can create multiple flow logs to send traffic to different destinations.

Selected resources [Info](#)

Name	Resource ID	State
CustomVPC	vpc-0a242371c74173265	Available

Flow log settings

Name - optional

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

Accept
 Reject
 All

Maximum aggregation interval [Info](#)
The maximum interval of time during which a flow of packets is captured and aggregated into a flow log record.

10 minutes
 1 minute

Destination
The destination to which to publish the flow log data.

Send to CloudWatch Logs
 Send to an Amazon S3 bucket

S3 bucket ARN
The ARN of the Amazon S3 bucket to which the flow log is published. You can specify a specific folder in the bucket using the bucket_ARN/folder_name/ format.

arn:aws:s3:::avinash.bucket

Info Please note, a resource-based policy will be created for you and attached to the target bucket.

Log record format
Specify the fields to include in the flow log record.

AWS default format
 Custom format

Format preview

```
 ${version} ${account-id} ${interface-id} ${srcaddr} ${dstaddr} ${srcport} ${dstport}
 ${protocol} ${packets} ${bytes} ${start} ${end} ${action} ${log-status}
```

Copy

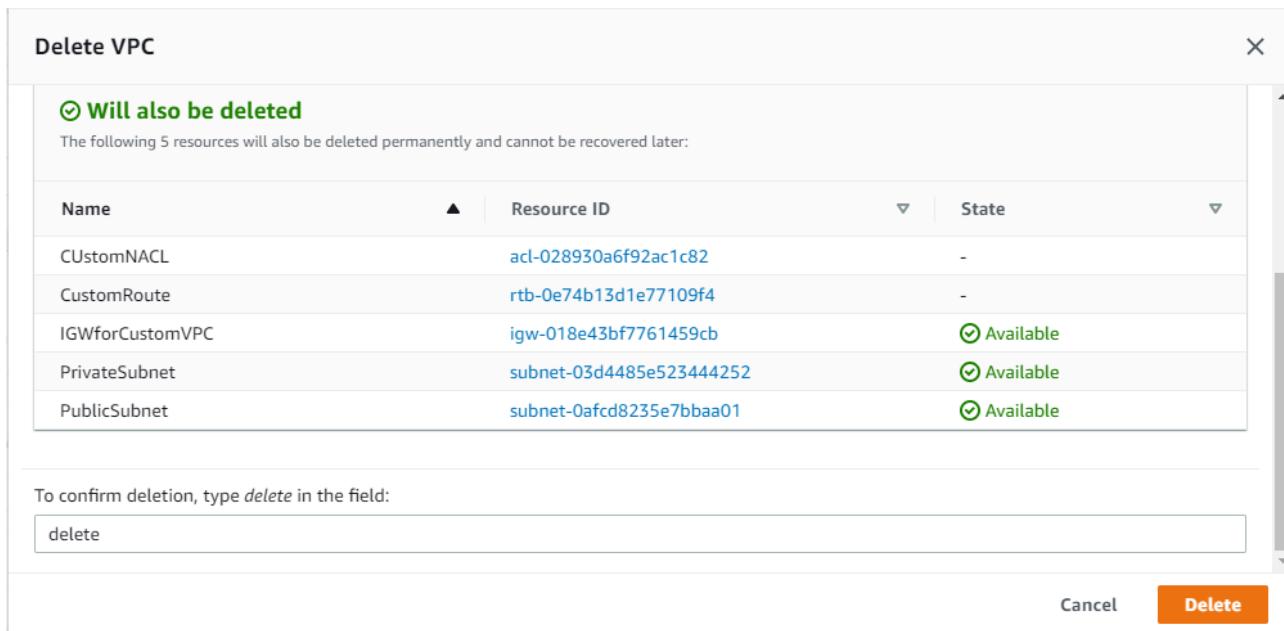
3. Select the Filter and choose what traffic (All/Accept/Reject) you want to get in Log.
4. We are storing all the vpc logs to an s3 bucket.

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, internet Gateways, Route tables etc will delete along with VPC.

Your VPCs (1/2)					Actions	Create
					C	
<input type="text"/> Filter VPCs						
Name	VPC ID	State	IPv4 CIDR			
CustomVPC	vpc-0a242371c74173265	Available	192.168.0.0/1			
-	vpc-01265273360128840	Available	172.31.0.0/16			

Details CIDRs **Flow logs** Tags



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.

If we want to enable communication between On-Premise and AWS environment, we need to implement VPN solutions. Please refer to the below video for VPC creation and OpenVPN deployment.

VPC Creation Step by Step Process:

https://www.youtube.com/watch?v=Ze_X53wdogA

OpenVPN Deployment Creation Step by Step Process:

<https://www.youtube.com/watch?v=JSTyjISscNg>

Databases on AWS

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 relational database has roots going back to the 1970s when Edgar F. Codd, working for IBM, developed the concepts of the relational model. Today, relational databases power all types of applications from social media apps, e-commerce websites, and blogs to complex enterprise applications.

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 tasks such 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	29
102	Anudeep	T	M	27
103	Aravind	Reddy	M	25
104	Vikas	Ch	M	23

Here is an example of a basic table that would sit in a relational database. There are five fields with 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. Each student has a *StudentID* field, which is usually a unique number per student. A unique number 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 are organized and how the application uses the relational database.

OLTP refers to transaction-oriented applications that are frequently writing and changing data (for example, data entry and 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 for reporting and analysis via OLAP. Organizations typically use data warehouses to compile reports and search the database using highly complex queries.

NoSQL Databases

NoSQL databases have gained significant popularity in recent years because they are often simpler to use, more flexible, and can achieve performance levels that are difficult or impossible with traditional relational databases.

Traditional relational databases are difficult to scale beyond a single server without significant engineering and cost, but a NoSQL architecture 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 to connect 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 using standard tools like pgAdmin and supports standard JDBC/ODBC drivers.

MariaDB: MariaDB is a popular open source database engine built by the creators of MySQL and enhanced with enterprise tools and functionality.

Oracle: Oracle is one of the most popular relational databases used in the enterprise and is fully supported 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 DB Instance 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 MySQL compatible, and provides for increased reliability and performance over standard MySQL deployments. Amazon Aurora can deliver up to five times better performance compared to MySQL. We can use the same code, tools, and applications that we use with existing MySQL databases with Amazon 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 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 16TB in provisioned storage and up to 30,000 IOPS.

Backup and Recovery

Amazon RDS provides two mechanisms for backing up the database:

1. Automated backups and
2. Manual snapshots.

Automated Backups: An automated backup is an Amazon RDS feature that continuously tracks changes and backs up 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 window called 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 time during this retention period, creating a new DB Instance.

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

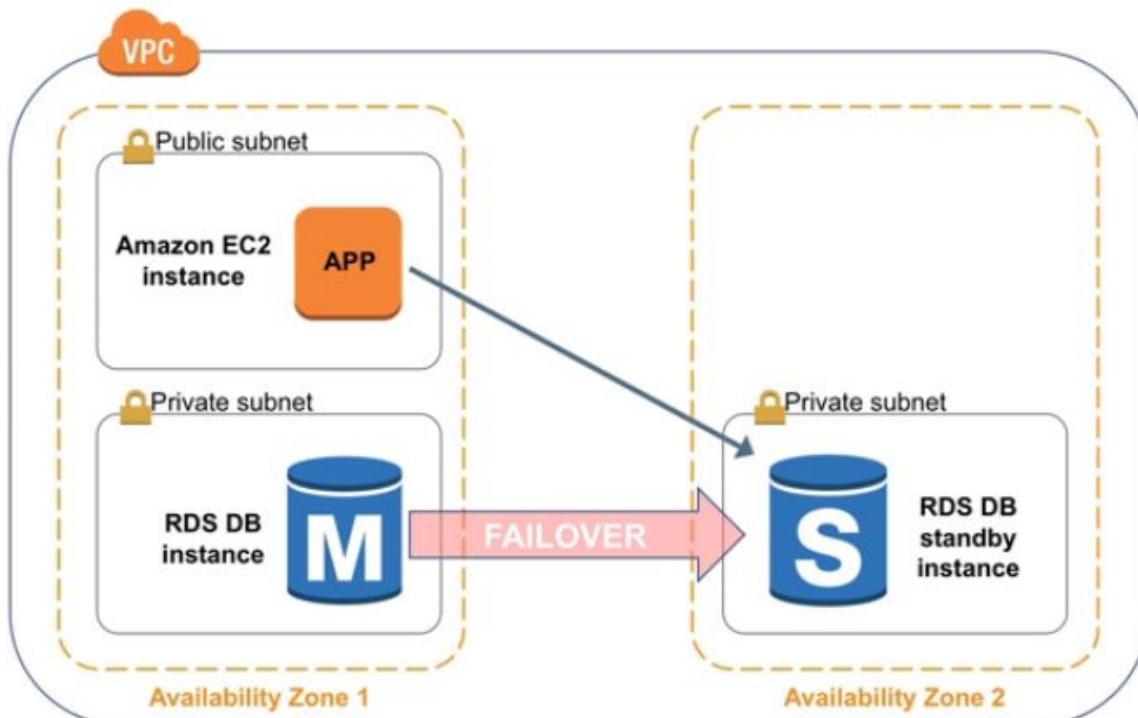
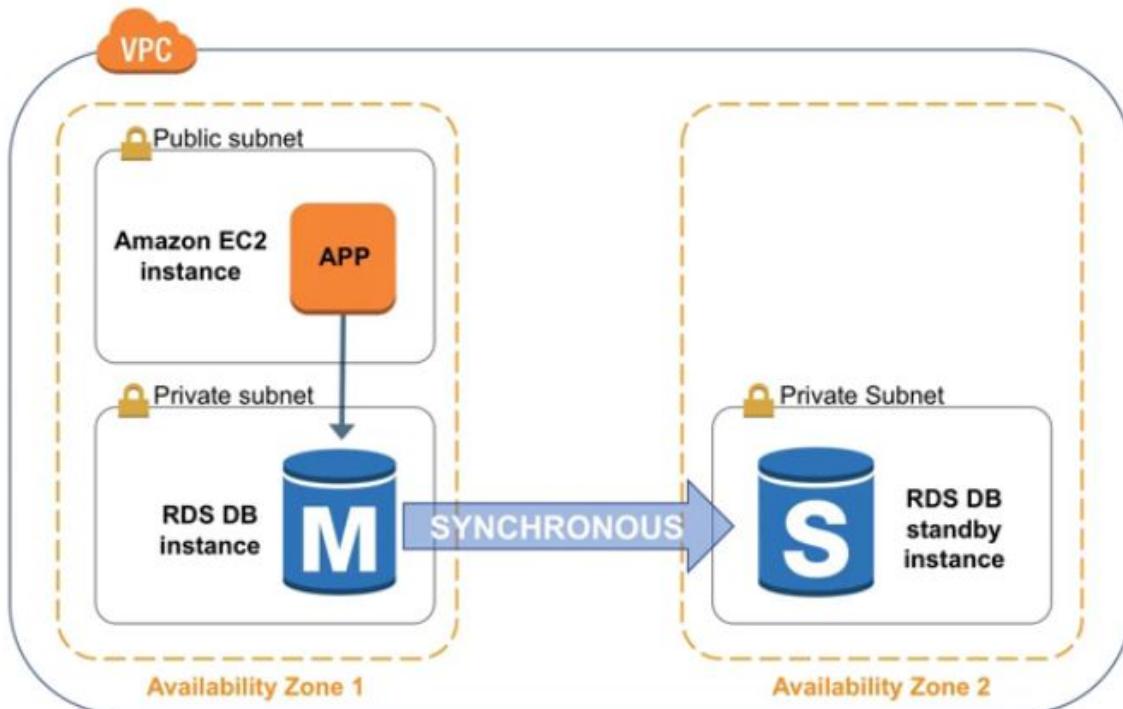
- A DB manual snapshot is initiated by us and can be created as frequently as we want.
- We can then restore the DB Instance to the specific state in the DB snapshot at any time.
- Manual DB snapshots are kept until you explicitly delete them with the Amazon RDS console.

Recovery: We can use automated backup or manual snapshot to recover 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.

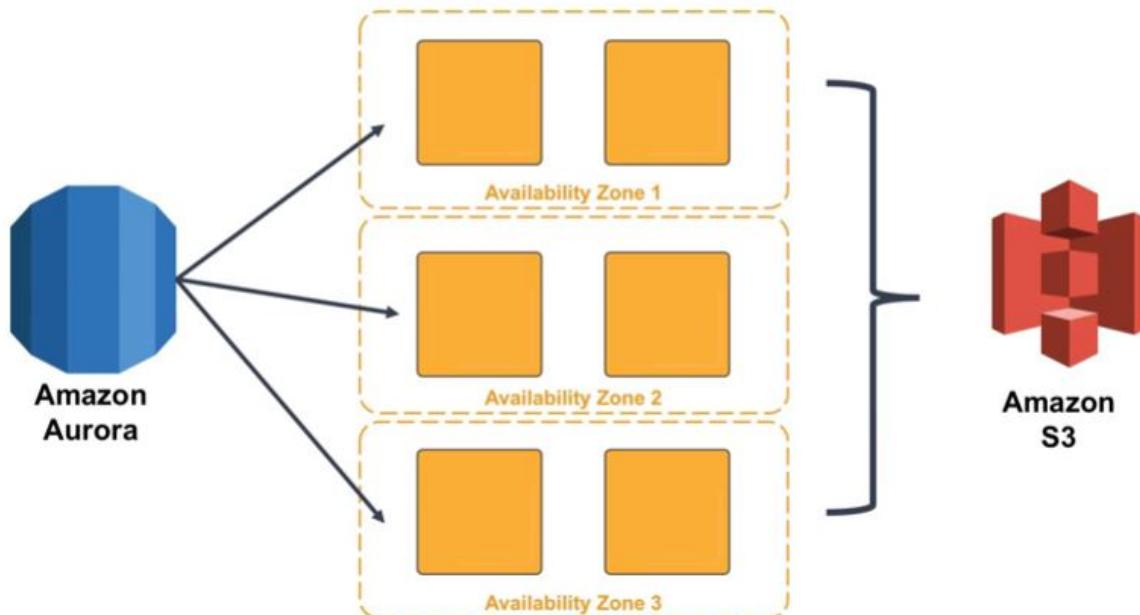
Multi-AZ: By using Multi-AZ we can increase the availability of the database using replication. We will get a same copy of production database in another availability zone for DR purpose (Disaster Recovery).



- 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 take 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 recovery only; 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.



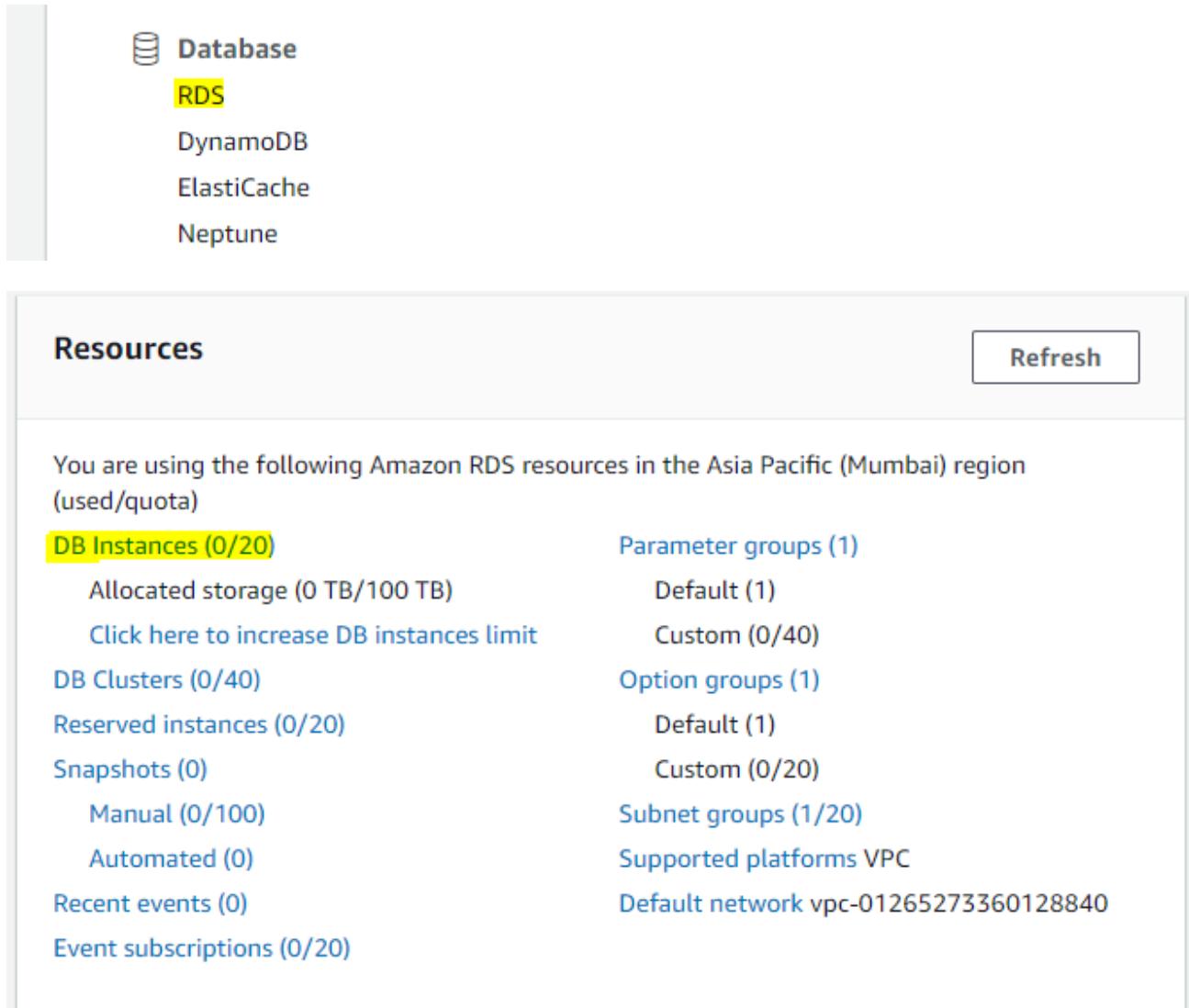
Read replicas are currently supported for:

- 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 endpoint.

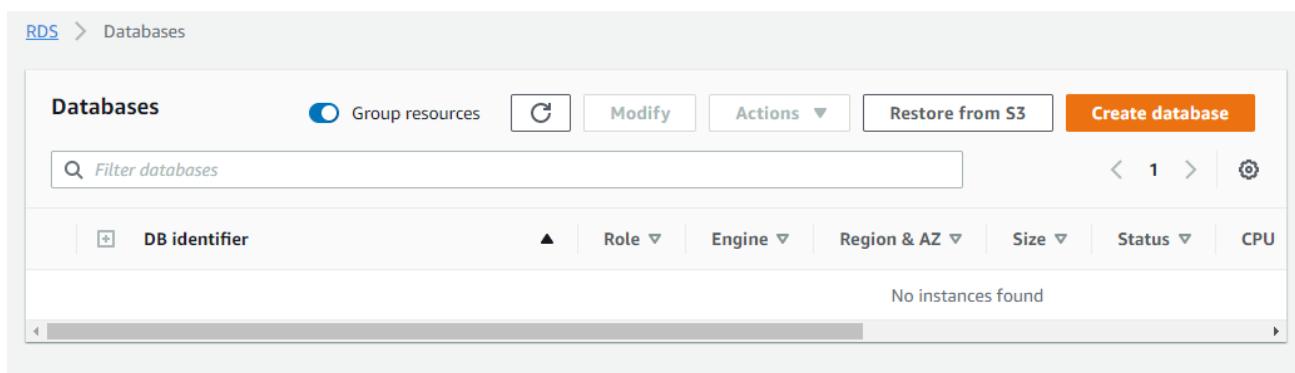
- You cannot have Read Replicas that have Multi-AZ
- You can create Read Replica's of Multi-AZ source databases however.
- Read Replicas can be promoted to be their own databases. This breaks the replication.

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 under Database.



The screenshot shows the AWS Management Console navigation bar on the left with 'Database' selected, and 'RDS' highlighted in yellow. Below the navigation are links for 'DynamoDB', 'ElastiCache', and 'Neptune'. The main content area is titled 'Resources' and includes a 'Refresh' button. A message states: 'You are using the following Amazon RDS resources in the Asia Pacific (Mumbai) region (used/quota)'. It lists various resource counts: DB Instances (0/20), DB Clusters (0/40), Reserved instances (0/20), Snapshots (0), Recent events (0), Event subscriptions (0/20), Parameter groups (1), Option groups (1), Subnet groups (1/20), and Supported platforms VPC. A Default network is listed as vpc-01265273360128840.



The screenshot shows the 'Databases' section of the RDS service. The top navigation bar has 'RDS' and 'Databases' selected. The main interface includes a 'Create database' button in orange. Below it is a search bar with 'Filter databases' placeholder text. A table header with columns: 'DB identifier', 'Role', 'Engine', 'Region & AZ', 'Size', 'Status', and 'CPU'. A note at the bottom says 'No instances found'.

2. Click on the “Create database” option.

Engine options

Engine type [Info](#)

- Amazon Aurora 
- MySQL 
- MariaDB 
- PostgreSQL 
- Oracle 
- Microsoft SQL Server 

Edition

- MySQL Community

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 “**Free tier**”.

Version

MySQL 8.0.23

Templates

Choose a sample template to meet your use case.

- Production
Use defaults for high availability and fast, consistent performance.
- Dev/Test
This instance is intended for development use outside of a production environment.
- Free tier
Use RDS Free Tier to develop new applications, test existing applications, or gain hands-on experience with Amazon RDS.
[Info](#)

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

Settings

DB instance identifier [Info](#)

Type a name for your DB instance. The name must be unique across all DB instances owned by your AWS account in the current AWS Region.

The DB instance identifier is case-insensitive, but is stored as all lowercase (as in "mydbinstance"). Constraints: 1 to 60 alphanumeric characters or hyphens. First character must be a letter. Can't contain two consecutive hyphens. Can't end with a hyphen.

▼ Credentials Settings

Master username [Info](#)

Type a login ID for the master user of your DB instance.

1 to 16 alphanumeric characters. First character must be a letter

Auto generate a password

Amazon RDS can generate a password for you, or you can specify your own password

DB instance class

DB instance class [Info](#)

Choose a DB instance class that meets your processing power and memory requirements. The DB instance class options below are limited to those supported by the engine you selected above.

- Standard classes (includes m classes)
- Memory optimized classes (includes r and x classes)
- Burstable classes (includes t classes)

db.t2.micro

1 vCPUs 1 GiB RAM Not EBS Optimized

Include previous generation classes

Storage

Storage type [Info](#)

General Purpose (SSD)

Allocated storage
20 GiB
(Minimum: 20 GiB, Maximum: 16,384 GiB) Higher allocated storage [may improve](#) IOPS performance.

Storage autoscaling [Info](#)
Provides dynamic scaling support for your database's storage based on your application's needs.

Enable storage autoscaling
Enabling this feature will allow the storage to increase once the specified threshold is exceeded.

Availability & durability

Multi-AZ deployment [Info](#)

Create a standby instance (recommended for production usage)
Creates a standby in a different Availability Zone (AZ) to provide data redundancy, eliminate I/O freezes, and minimize latency spikes during system backups.

Do not create a standby instance

- **DB Engine:** We have selected the Db Engine as MySQL.
- **DB Engine Version:** Select the appropriate DB Engine Version as per your requirements. 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 or choose to “auto Generate” password.

Virtual private cloud (VPC) [Info](#)
VPC that defines the virtual networking environment for this DB instance.

Default VPC (vpc-01265273360128840) ▾

Only VPCs with a corresponding DB subnet group are listed.

ⓘ After a database is created, you can't change the VPC selection.

Subnet group [Info](#)
DB subnet group that defines which subnets and IP ranges the DB instance can use in the VPC you selected.

default ▾

Public access [Info](#)

Yes
Amazon EC2 instances and devices outside the VPC can connect to your database. Choose one or more VPC security groups that specify which EC2 instances and devices inside the VPC can connect to the database.

No
RDS will not assign a public IP address to the database. Only Amazon EC2 instances and devices inside the VPC can connect to your database.

VPC security group
Choose a VPC security group to allow access to your database. Ensure that the security group rules allow the appropriate incoming traffic.

Choose existing
Choose existing VPC security groups

Create new
Create new VPC security group

Existing VPC security groups

Choose VPC security groups ▾

mysqlsg X

Availability Zone [Info](#)

ap-south-1a ▾

▼ Additional configuration

Database port [Info](#)
TCP/IP port that the database will use for application connections.

3306

- **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 authentication

Database authentication options [Info](#)

Password authentication
Authenticates using database passwords.

Password and IAM database authentication
Authenticates using the database password and user credentials through AWS IAM users and roles.

Password and Kerberos authentication
Choose a directory in which you want to allow authorized users to authenticate with this DB instance using Kerberos Authentication.

▼ Additional configuration

Database options, backup enabled, backtrack disabled, Enhanced Monitoring disabled, maintenance, CloudWatch Logs, delete protection disabled

Database options

Initial database name [Info](#)

If you do not specify a database name, Amazon RDS does not create a database.

DB parameter group [Info](#)

Option group [Info](#)

- **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 database engine 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 provide 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.

Backup

Enable automated backups
Creates a point-in-time snapshot of your database

⚠ Please note that automated backups are currently supported for InnoDB storage engine only. If you are using MyISAM, refer to details [here](#).

Backup retention period [Info](#)

Choose the number of days that RDS should retain automatic backups for this instance.

7 days ▾

Backup window [Info](#)

Select the period for which you want automated backups of the database to be created by Amazon RDS.

- Select window
 No preference

Copy tags to snapshots

We can set the BackupRetention Period as well as the Backup window's Start Time and Duration.

Monitoring

Enable Enhanced monitoring

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

Log exports

Select the log types to publish to Amazon CloudWatch Logs

Audit log

Error log

General log

Slow query log

IAM role

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

RDS service-linked role

Maintenance

Auto minor version upgrade [Info](#)

Enable auto minor version upgrade

Enabling auto minor version upgrade will automatically upgrade to new minor versions as they are released. The automatic upgrades occur during the maintenance window for the database.

Maintenance window [Info](#)

Select the period you want pending modifications or maintenance applied to the database by Amazon RDS.

Select window

No preference

Deletion protection

Enable deletion protection

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

- **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.

5. 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 is actually created by RDS. During this time, your database will remain inaccessible.

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

Backing-up: RDS will automatically take a backup of your DB instance when it is first created. You can view all your DB instance snapshots using the Snapshots option on 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.

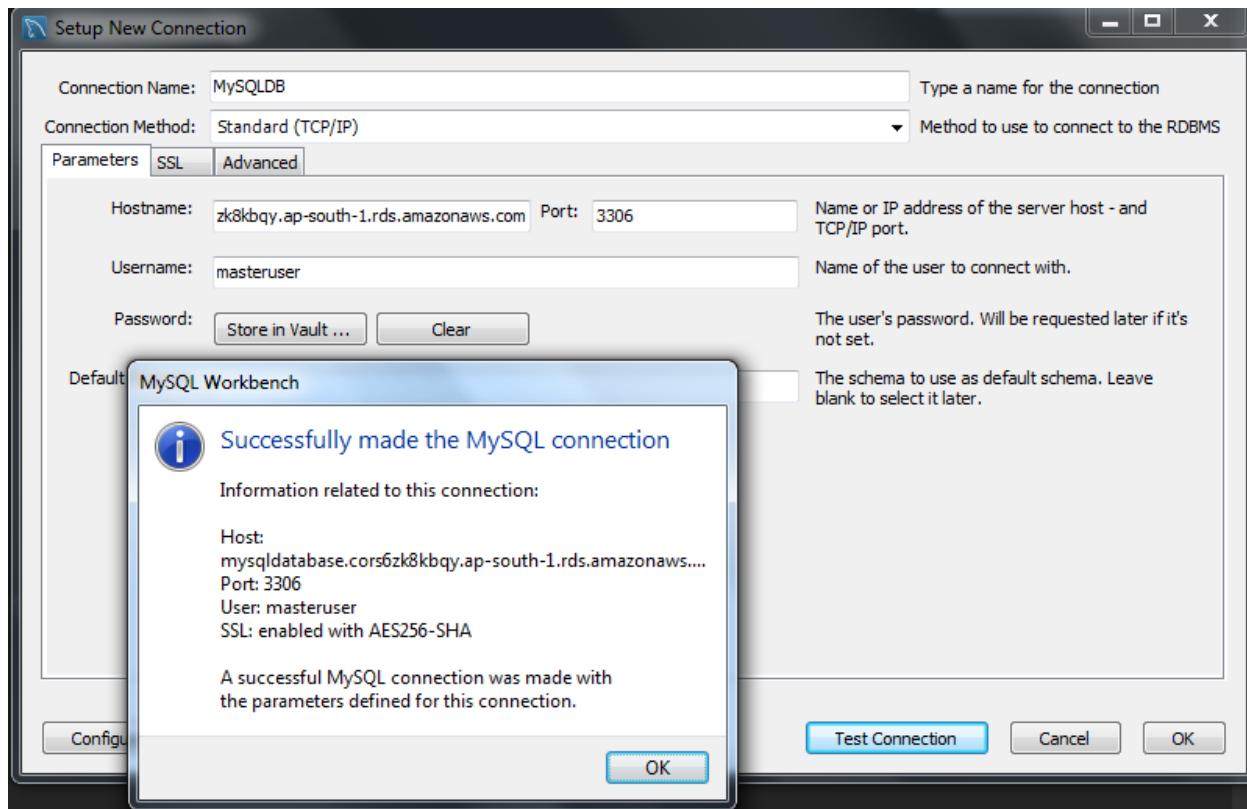
Here is the details for newly launched RDS instance.

Details			
Configurations	Security and network	Instance and IOPS	Maintenance details
ARN arn:aws:rds:ap-south-1:1518084852393:db:mysqldatabase	VPC vpc-7d7ab214 Subnet group default Subnets subnet-01f92d68 subnet-721b0f38 Security groups rds-launch-wizard-1 (sg-c82bd0a3) (active) Publicly accessible Yes	Instance Class db.t2.micro Storage Type General Purpose (SSD) Storage 20 GB Availability and durability DB instance status creating Multi AZ No Automated backups Enabled (7 Days)	Auto minor version upgrade No Maintenance window mon:00:00-mon:00:30 UTC (GMT) Backup window 22:00-23:00 UTC (GMT) Pending Modifications Master User Password: **** Pending maintenance none Encryption details Encryption enabled No
Engine MySQL 5.6.37			
License Model General Public License			
DB Name mydatabase			
Username masteruser			
Option Group default:mysql-5-6			
Parameter group default.mysql5.6 (in-sync)			
Copy tags to snapshots No			
Resource ID db-AYVZZ5UZTDG6VNNOYPKPDDG2MU			
IAM DB Authentication Enabled No			

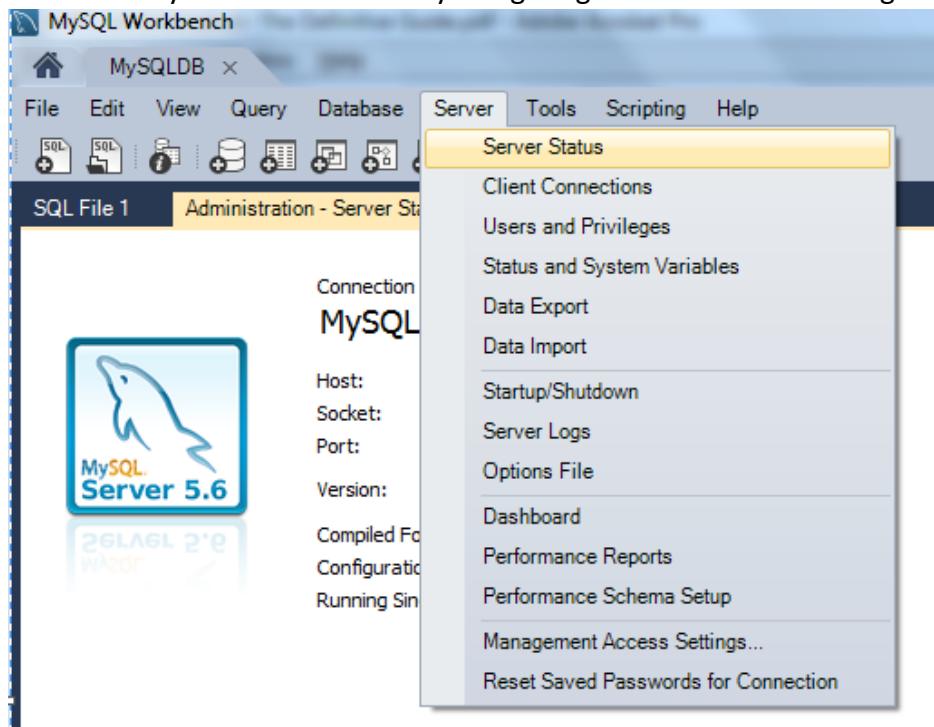
6. To test the connectivity we are going to use MySQL Workbench application, Download and install on any of the local machine or EC2 instance if you want to test it in graphical manner. You can download the MySQL workbench from the following URL:

<https://dev.mysql.com/downloads/workbench/>

7. 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 result.



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



9. 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**.

After launching the Linux instance, Installmysql package by running
yum install mysql

Then run `# mysql -u <USERNAME> -h <DATABASE_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)

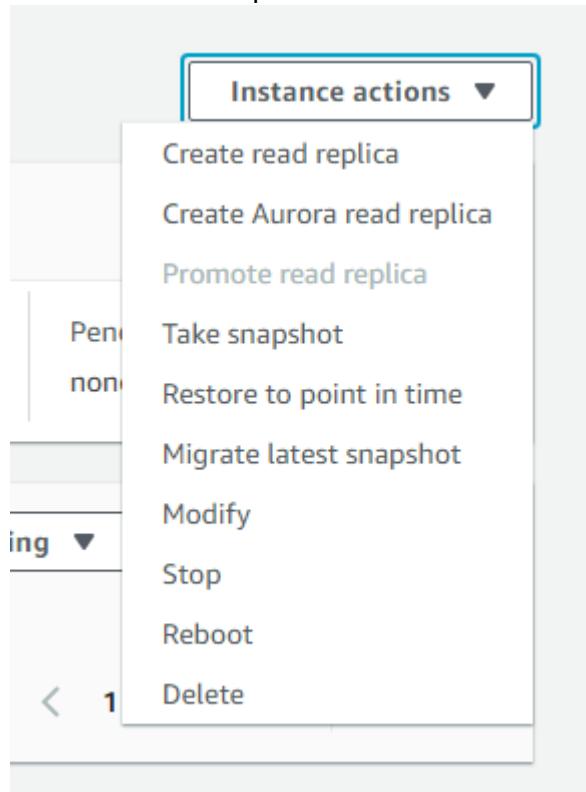
Copyright (c) 2000, 2017, Oracle and/or its affiliates. All rights reserved.

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 snapshots.

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.

Launch DB Instance

You are creating 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.
This feature is currently supported for InnoDB storage engine only. If you are using MyISAM, refer to details [here](#).

Restore time

Point in time to restore from

Latest restorable time
January 31, 2018 at 6:06:23 PM UTC+5:30

Custom
Specify a custom date and time to restore from

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 its status to Stopped state, we can start it 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.

Delete DB Instance

Options

Are you sure you want to Delete the mysqldatabase DB Instance?

Create final snapshot?
Determines whether a final DB Snapshot is created before the DB instance is deleted.

Yes ▾

Final snapshot name
The DBSnapshotIdentifier of the new DB Snapshot created.

mysqldatabase-final-snapshot

Cancel

Delete

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.

Take DB Snapshot

This feature is currently supported for InnoDB storage engine only. If you are using MyISAM, refer to details [here](#).

Settings

To take a snapshot of this DB instance you must provide a name for the snapshot.

DB instance
The unique key that identifies a DB instance. This parameter isn't case-sensitive.
mysqldatabase

Snapshot name
The Identifier for the DB Snapshot.

Cancel
Take Snapshot

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

Snapshots (2)		Owned by Me	Snapshot Actions	Create snapshot
<input type="checkbox"/>	Snapshot	DB Instance or Cluster	Snapshot Creation Time	Status Progress VPC
<input type="checkbox"/>	rds:mysqldatabase-2018-01-31-12-35	mysqldatabase	Wed Jan 31 18:06:23 GMT+530 2018	available Completed vpc-7d7a
<input type="checkbox"/>	snapshot	mysqldatabase		creating 0% vpc-7d7a

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.

Snapshots (2)			Owned by Me	Snapshot Actions	Create snapshot
	Snapshot	DB Instance or Cluster	Snapshot Creation Time		
<input type="checkbox"/>	rds:mysqldatabase-2018-01-31-12-35	mysqldatabase	Wed Jan 31 18:06:23 GMT+530 2018	Restore Snapshot	Create snapshot
<input checked="" type="checkbox"/>	snapshot	mysqldatabase	Wed Jan 31 18:17:08 GMT+530 2018	Copy Snapshot	Share Snapshot

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.

Make Copy of DB Snapshot?

Settings

Source DB Snapshot
DB Snapshot Identifier for the automated snapshot being copied.
snapshot

Destination Region [info](#)
Asia Pacific (Mumbai)

New DB Snapshot Identifier
DB Snapshot Identifier for the new snapshot

Target Option Group (Optional) [info](#)
No preference

Copy Tags [info](#)

Encryption

Encryption [info](#)

Enable Encryption
Select to encrypt the given instance. Master key ids and aliases appear in the list after they have been created using the Key Management Service(KMS) console. [Learn More](#).

Disable Encryption

[Cancel](#)
Copy 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.

Manage Snapshot Permissions

Preferences

You are sharing an unencrypted DB snapshot. When you share an unencrypted DB snapshot, you give the other account permission to make a copy of the DB snapshot and to restore a database from your DB snapshot.

DB snapshot

snapshot

DB snapshot visibility

- Private
- Public

AWS account ID

Add

AWS account ID

Delete

Please add AWS account ID

Cancel

Save

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.

Migrate Database

Migrate this database to a new DB Engine by selecting your desired options for the migrated instance.

Instance specifications

Migrate to DB Engine

Name of the Database Engine

aurora

DB Engine Version

Version Number of the Database Engine to be used for this instance

5.6.10a (default)

DB Instance Class

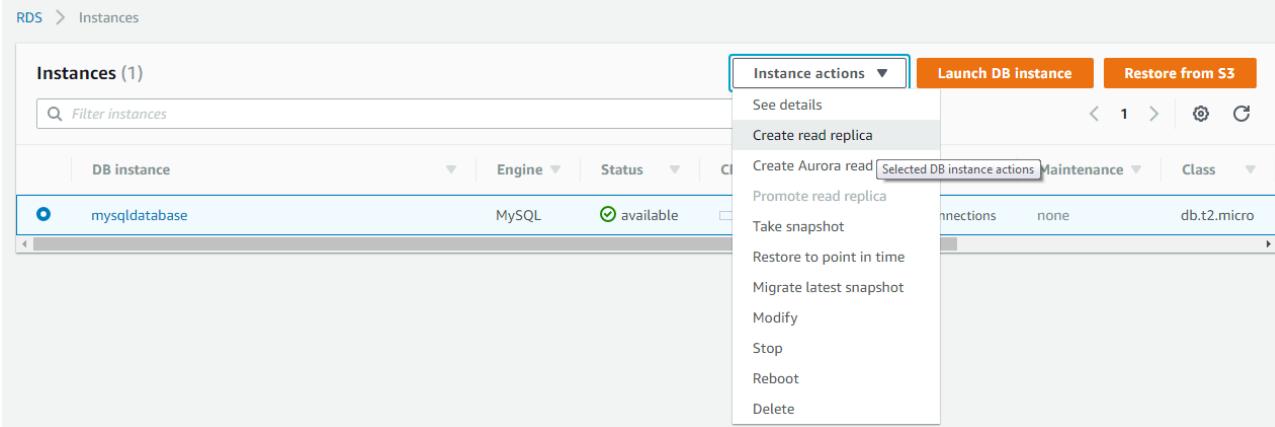
Contains the compute and memory capacity of the DB Instance.

- Select one -

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.



Create read replica DB instance

You are creating a replica DB instance from a source DB instance. This new DB instance will have the source DB instance's DB security groups and DB parameter groups.

Network & Security

Destination region
The region in which the replica will be launched

Asia Pacific (Mumbai)

Destination DB subnet group

default

Availability zone
The EC2 Availability Zone that the database instance will be created in.

No preference

Publicly accessible

Yes
EC2 instances and devices outside of the VPC hosting the DB instance will connect to the DB instances. You must also select one or more VPC security groups that specify which EC2 instances and devices can connect to the DB instance.

No
DB instance will not have a public IP address assigned. No EC2 instance or devices outside of the VPC will be able to connect.

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

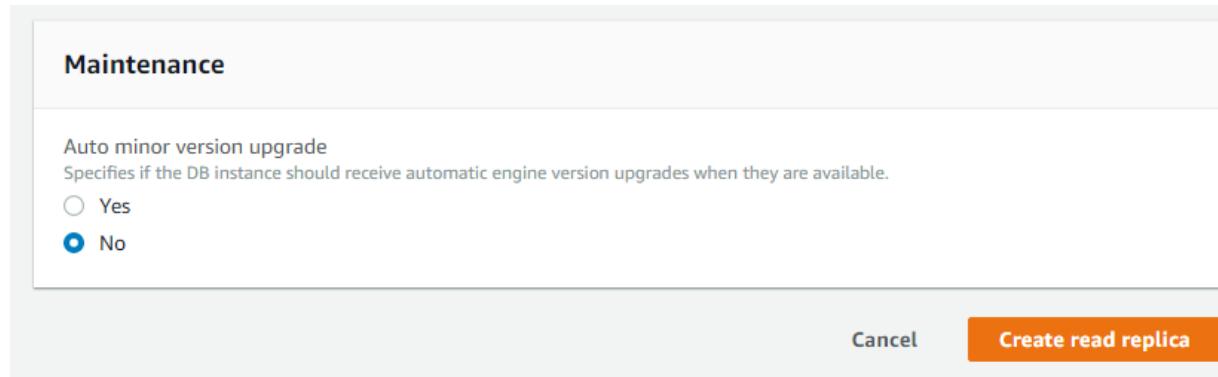
IAM DB authentication [info](#)

- 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.

RDS > Instances

Instances (2)							Instance actions	Launch DB instance	Restore from S3
DB instance		Engine	Status	CPU	Current activity	Maintenance	Class		
<input type="radio"/>	mysqldatabase	MySQL	modifying	<div style="width: 1.33%;">1.33%</div>	<div style="width: 0%;">0 Connections</div>	none	db.t2.micro		
<input type="radio"/>	myreadreplica	MySQL	creating	<div style="width: 0%;">0%</div>	<div style="width: 0%;">0 Connections</div>	none	db.t2.micro		

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

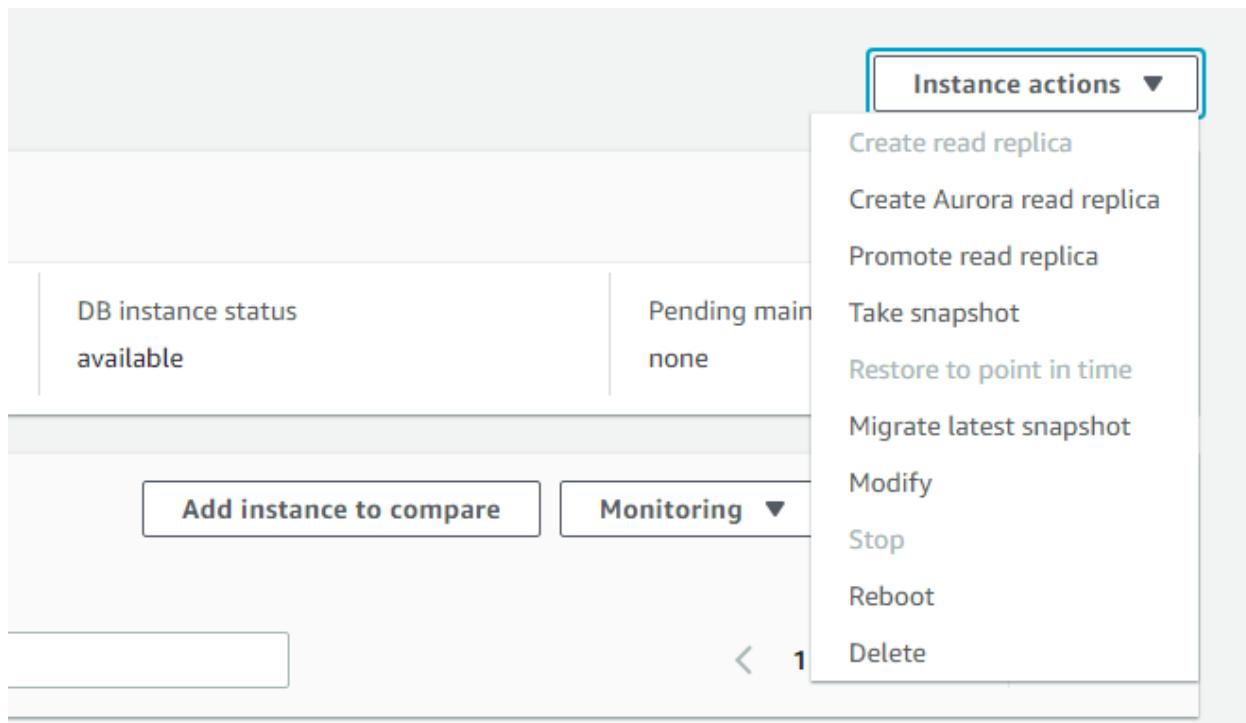
RDS > Instances

Instances (2)							Instance actions	Launch DB instance	Restore from S3
DB instance		Engine	Status	CPU	Current activity	Maintenance	Class		
<input type="radio"/>	mysqldatabase	MySQL	available	<div style="width: 1.31%;">1.31%</div>	<div style="width: 1%;">1 Connections</div>	none	db.t2.micro		
<input type="radio"/>	myreadreplica	MySQL	available	<div style="width: 1.17%;">1.17%</div>	<div style="width: 0%;">0 Connections</div>	none	db.t2.micro		

Replication (2)				
DB instance Role Zone Replication source Lag				
mysqldatabase	master	ap-south-1b	-	-
myreadreplica	replica	ap-south-1a	mysqldatabase	0 Milliseconds

- 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”



Promote Read Replica: myreadreplica

The screenshot shows the 'Promote Read Replica' configuration dialog. It has a 'Preferences' section with the following settings:

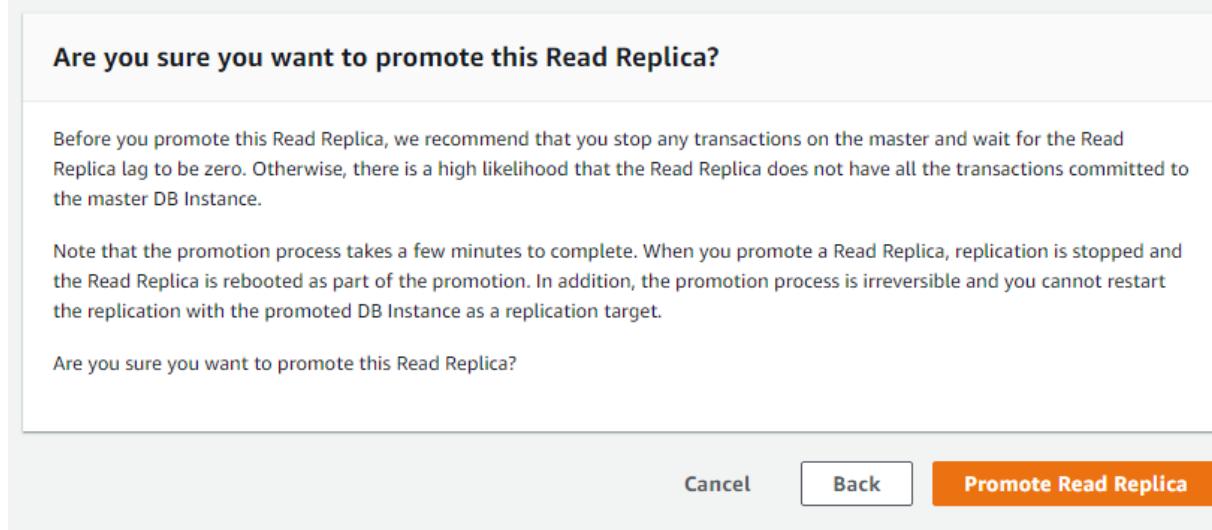
- Enable automatic backups: Yes
- Selecting no will disable automated backups
- Backup retention period: 1 days
- Backup window: No preference
- The daily time range (in UTC) during which automated backups are created if automated backups are enabled.

At the bottom right are 'Cancel' and 'Continue' buttons.

- 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:

Promote Read Replica: myreadreplica



Instances (2)		Instance actions	Launch DB Instance	Restore from S3		
		DB instance	Engine	Status	CPU	Current activity
<input type="radio"/>	myreadreplica	MySQL	(i) rebooting	<div style="width: 1.53%;">1.53%</div>	<div style="width: 0%;">0 Connections</div>	
<input type="radio"/>	mysqldatabase	MySQL	(✓) available	<div style="width: 1.36%;">1.36%</div>	<div style="width: 0%;">0 Connections</div>	

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

Replication (1)		Filter replication	< 1 >	①
DB instance	Role	Zone	Replication source	Lag
myreadreplica		ap-south-1a	-	-

Amazon DynamoDB:

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

Amazon DynamoDB can provide consistent performance levels by automatically distributing the data and traffic for a table over multiple partitions. After you configure a certain read or write capacity, Amazon DynamoDB will automatically add enough infrastructure capacity to support the requested throughput levels. As your demand changes over time, you can adjust the read or write capacity after a table has been created, and Amazon DynamoDB will add or remove 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 requests over 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 certain amount of read and write capacity to handle your expected workloads.

1. We can find Dynamo DB under Database module

Database

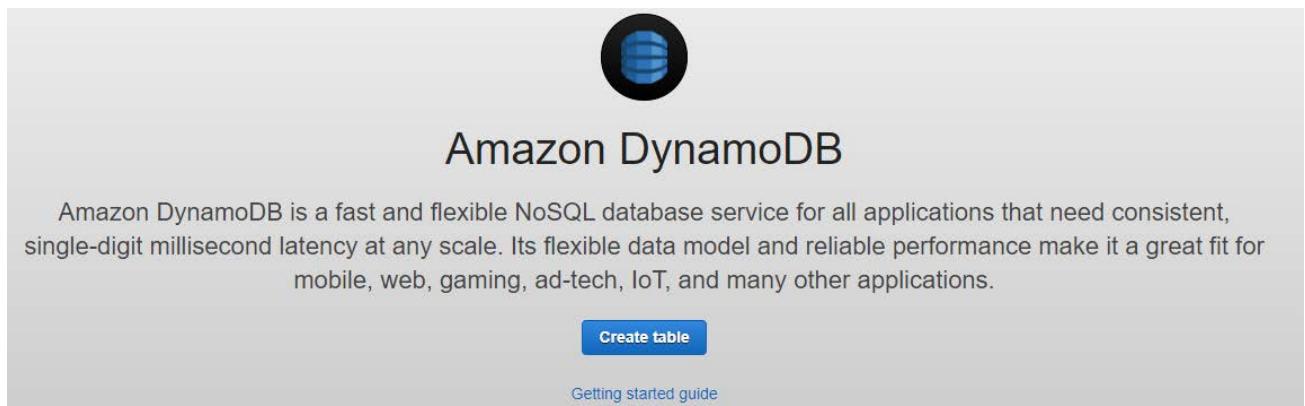
RDS

DynamoDB

ElastiCache

Neptune

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



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

Create DynamoDB table

Tutorial



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

Table name*	<input type="text" value="schooldb"/>	
Primary key*	Partition key	
	<input type="text" value="studentid"/>	<input type="button" value="Number"/> 
<input type="checkbox"/> Add sort key		

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

<input checked="" type="checkbox"/> Read capacity	<input checked="" type="checkbox"/> Write capacity
<input type="checkbox"/> Same settings as read	
Target utilization <input type="text" value="70"/> %	<input type="text" value="70"/> %
Minimum provisioned capacity <input type="text" value="5"/> units	<input type="text" value="5"/> units
Maximum provisioned capacity <input type="text" value="10000"/> units	<input type="text" value="10000"/> units
<input checked="" type="checkbox"/> Apply same settings to global secondary indexes	
<input checked="" type="checkbox"/> Apply same settings to global secondary indexes	

- If you don't want to enable auto scaling of DynamoDB, simply uncheck the "Read capacity" & "Write capacity" options.
- As shown below, a table is created and you can navigate to "Items" and you can start adding items.

The screenshot shows the AWS DynamoDB console interface. On the left, there's a table list with a single entry: 'MySchoolDB' (Active, Student ID (Number) as partition key). On the right, the details for 'MySchoolDB' are displayed under the 'Overview' tab. The table details include:

- Table name: MySchoolDB
- Primary partition key: Student ID (Number)
- Primary sort key: -
- Time to live attribute: DISABLED [Manage TTL](#)
- Table status: Active
- Creation date: March 2, 2018 at 6:47:52 PM UTC+5:30
- Provisioned read capacity units: 5 (Auto Scaling Disabled)
- Provisioned write capacity units: 5 (Auto Scaling Disabled)
- Last decrease time: -
- Last increase time: -
- Storage size (in bytes): 0 bytes

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 and optimized for high-performance analysis and reporting of very large datasets. Traditional data warehouses are difficult and expensive to manage, especially for large datasets. Amazon Redshift not only significantly lowers the cost of a data warehouse, but it also makes it easy to analyze large amounts of data very quickly.

Amazon Redshift gives you fast querying capabilities over structured data using standard SQL commands to support interactive querying over large datasets. With connectivity via ODBC or JDBC, Amazon Redshift integrates well with various data loading, reporting, data mining, and analytics tools. Amazon Redshift is based on industry-standard PostgreSQL, so most existing SQL 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 tasks such as backups and patching. Amazon Redshift automatically monitors your nodes and drives 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 directly only 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.

Read the FAQs on Redshift at given URL: <https://aws.amazon.com/redshift/faqs/>

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.

Read the FAQs on ElastiCache at given URL:

<https://aws.amazon.com/elasticsearch/faqs/>

Please refer below video for Database migration service.

<https://www.youtube.com/watch?v=iRhJDGp-ew8>

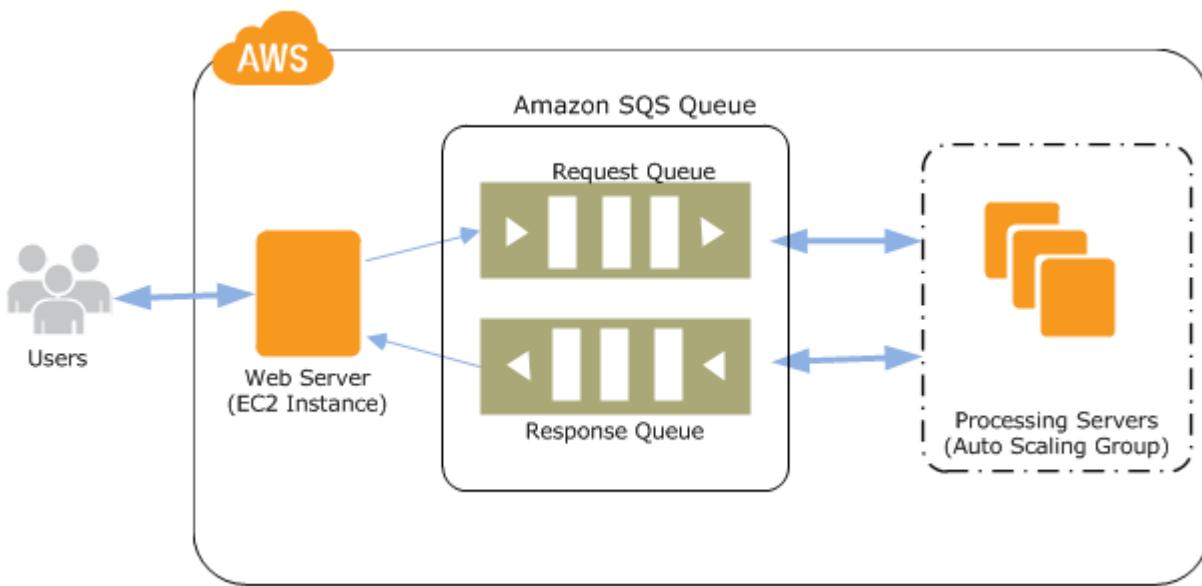
APPLICATION SERVICES:

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 Bill 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 free
- \$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.

Using Amazon SQS, you can store application messages on reliable and scalable infrastructure, enabling you to move data between distributed components to perform different tasks as needed.

Amazon SQS ensures delivery of each message at least once and supports multiple readers and writers interacting with the same queue. A single queue can be used simultaneously by many distributed application components, with no need for those components to coordinate with 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 of messages.

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 scope of 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.

Standard [Info](#)

At-least-once delivery, message ordering isn't preserved

- At-least once delivery
- Best-effort ordering

FIFO [Info](#)

First-in-first-out delivery, message ordering is preserved

- First-in-first-out delivery
- Exactly-once processing

Name

A queue name is case-sensitive and can have up to 80 characters. You can use alphanumeric characters, hyphens (-), and underscores (_).

Configuration

Set the maximum message size, visibility to other consumers, and message retention. [Info](#)

Visibility timeout Info	Message retention period Info
<input type="text" value="30"/> Seconds ▾	<input type="text" value="4"/> Days ▾
Should be between 0 seconds and 12 hours.	
Delivery delay Info	Maximum message size Info
<input type="text" value="0"/> Seconds ▾	<input type="text" value="256"/> KB
Should be between 0 seconds and 15 minutes.	
Receive message wait time Info	
<input type="text" value="0"/> Seconds	
Should be between 0 and 20 seconds.	

Amazon Simple Notification Service (Amazon SNS)

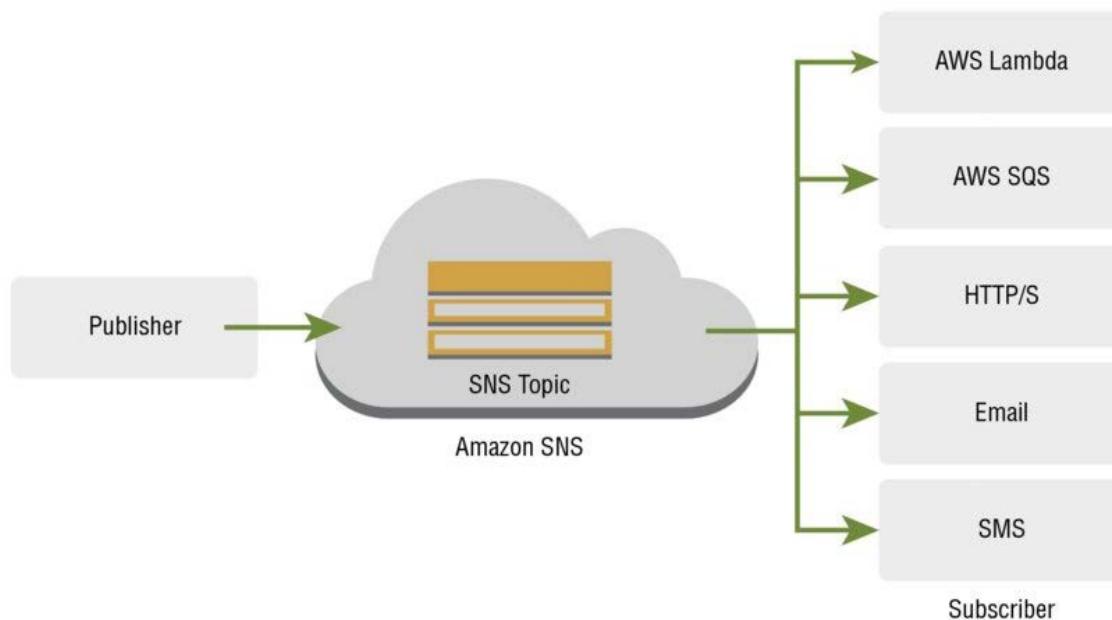
Amazon Simple Notification Service (Amazon SNS) is a web service that makes it easy to set up, operate, and send notifications from the cloud.

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 known as 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 by predefined thresholds. For example, we can receive immediate notification when an event occurs, such as a specific change to your Auto 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 news headlines to subscribers by email or SMS. Upon receiving the email or SMS text, interested readers 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. For example, you can use Amazon SNS for sending notifications to an application, indicating that an 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

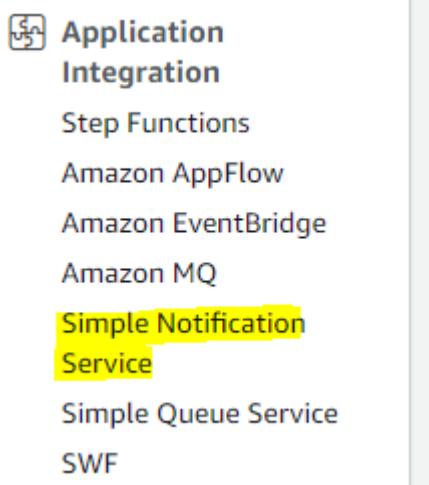
SNS vs SQS

- 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.



The screenshot shows the 'Topics' section of the Amazon SNS console. The sidebar on the left has 'Topics' selected. The main area displays a table with one row. The columns are 'Name', 'Type', and 'ARN'. The data row shows 'S3Notifications' under 'Name', 'Standard' under 'Type', and 'arn:aws:sns:ap-south-1:4283:S3Notifications' under 'ARN'. There are buttons for 'Edit', 'Delete', 'Publish message', and 'Create topic' at the top right of the table.

Name	Type	ARN
S3Notifications	Standard	arn:aws:sns:ap-south-1:4283:S3Notifications

- Create a new topic by selecting “**Create topic**” option, and give a name for Topic and Display Name.

Details

Type [Info](#)

Topic type cannot be modified after topic is created

FIFO (first-in, first-out)

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

Standard

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

Name

avinashsns

Maximum 256 characters. Can include alphanumeric characters, hyphens (-) and underscores (_).

Display name - *optional*

To use this topic with SMS subscriptions, enter a display name. Only the first 10 characters are displayed in an SMS message. [Info](#)

[Notifications](#)

Maximum 100 characters, including hyphens (-) and underscores (_).

► Encryption - *optional*

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

► Access policy - *optional*

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

► Delivery retry policy (HTTP/S) - *optional*

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

► Delivery status logging - *optional*

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

► Tags - *optional*

A tag is a metadata label that you can assign to an Amazon SNS topic. Each tag consists of a key and an optional value. You can use tags to search and filter your topics and track your costs. [Learn more](#)

- Once topic is created, 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.

Subscriptions | Access policy | Delivery retry policy (HTTP/S) | Delivery status logging | Encryption | Tags

Subscriptions (0) Edit Delete Request confirmation Confirm subscription Create subscription

Search < 1 > ⚙️

ID	Endpoint	Status	Protocol
No subscriptions found			

You don't have any subscriptions to this topic.

Create subscription

Create a subscription and choose “Protocol” as “Email” and enter a valid email address. **Make sure you login to email to confirm the subscription.**

Details

Topic ARN
 arn:aws:sns:ap-south-1:501170964283:avinashsns X

Protocol
The type of endpoint to subscribe

Select protocol

- Amazon Kinesis Data Firehose
- Amazon SQS
- AWS Lambda
- Email
- Email-JSON
- HTTP
- HTTPS
- Platform application endpoint
- SMS

Protocol
The type of endpoint to subscribe

Email

Endpoint
An email address that can receive notifications from Amazon SNS.

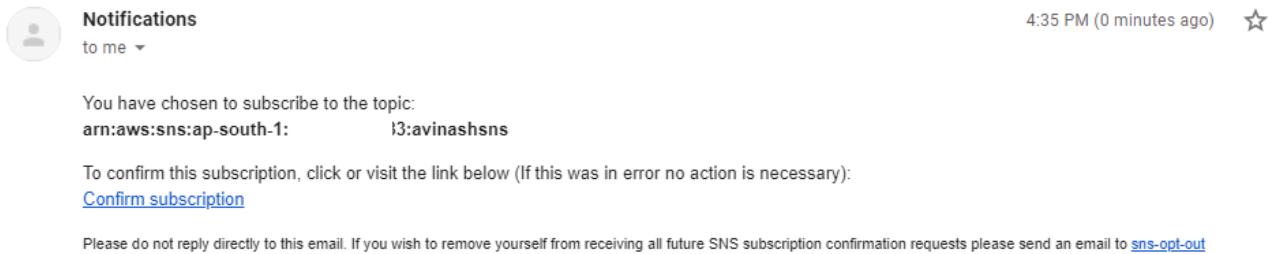
avizway@gmail.com

ⓘ After your subscription is created, you must confirm it. [Info](#)

- Now login to the mentioned Email ID and verify the Email from AWS SNS, and it'll ask you to subscribe to the topic.

- o You'll get an Email as mentioned below image.

AWS Notification - Subscription Confirmation ➔ Inbox x



- o Click on the Confirm Subscription Link, it'll redirect to another page which shows the subscription status page.
 - o 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.

The screenshot shows the AWS SNS Topics page. A topic named "avinashsns" is selected. The "Details" section shows the following information:

Name	Display name
avinashsns	Notifications

Buttons for "Edit", "Delete", and "Publish message" are visible.

- 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.

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 after 24 hours.

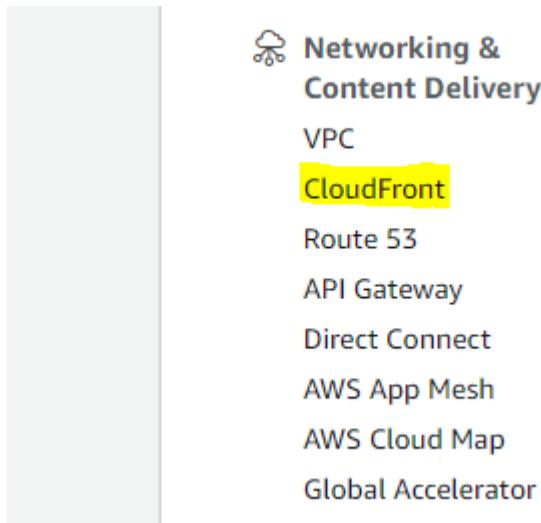
SignedURLs Use URLs that are valid only between certain times and optionally from certain IP addresses.

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 “Networking & Content Delivery”



2. Click on “Create Distribution” then we need to choose the “Origin Settings”. In below example am using S3 bucket as Origin.

The screenshot shows the "Origin" configuration page. It includes fields for "Origin domain" (set to "avinash.bucket.s3.ap-south-1.amazonaws.com"), "Origin path - optional" (set to "Enter the origin path"), and "Name" (set to "avinash.bucket.s3.ap-south-1.amazonaws.com").

3. It's always recommended to use OAI (Origin Access Identity) is used for sharing private content via CloudFront. The OAI is a virtual user identity that will be used to give your CF distribution permission to fetch a private object from your origin server (e.g. S3 bucket)).
4. Choose “Create new OAI”

S3 bucket access [Info](#)

Use a CloudFront origin access identity (OAI) to access the S3 bucket.

- Don't use OAI (bucket must allow public access)
 Yes use OAI (bucket can restrict access to only CloudFront)

Origin access identity

Select an existing origin access identity (recommended) or create a new identity.

[Create new OAI](#)**Create new OAI****Origin access identity**

Name your new origin access identity.

[Cancel](#)[Create](#)**S3 bucket access** [Info](#)

Use a CloudFront origin access identity (OAI) to access the S3 bucket.

- Don't use OAI (bucket must allow public access)
 Yes use OAI (bucket can restrict access to only CloudFront)

Origin access identity

Select an existing origin access identity (recommended) or create a new identity.

[Create new OAI](#)**Bucket policy**

Update the S3 bucket policy to allow read access to the OAI.

- No, I will update the bucket policy
 Yes, update the bucket policy

5. In this example am Choose "Legacy Cache settings". Default TTL is 86400 Sec (1 Day), Max TTL is 365 Days.

Cache policy and origin request policy (recommended)

Legacy cache settings

Headers
Choose which headers to include in the cache key.

Query strings
Choose which query strings to include in the cache key.

Cookies
Choose which cookies to include in the cache key.

Object caching

Use origin cache headers

Customize

Minimum TTL
Minimum time to live in seconds.

Maximum TTL
Maximum time to live in seconds.

Default TTL
Default time to live in seconds.

6. Choose the Distribution Settings

Settings

Price class [Info](#)
Choose the price class associated with the maximum price that you want to pay.

Use all edge locations (best performance)

Use only North America and Europe

Use North America, Europe, Asia, Middle East, and Africa

AWS WAF web ACL - optional
Choose the web ACL in AWS WAF to associate with this distribution.

Alternate domain name (CNAME) - optional
Add the custom domain names that you use in URLs for the files served by this distribution.

i To add a list of alternative domain names, use the [bulk editor](#).

Custom SSL certificate - optional
Associate a certificate from AWS Certificate Manager. The certificate must be in the US East (N. Virginia) Region (us-east-1).

7. 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 Global Accelerator

AWS Global Accelerator is a networking service that improves the availability and performance of the applications that you offer to your global users. AWS Global Accelerator uses the AWS global network to direct internet traffic from your users to your applications on AWS, making your users' experience more consistent.

Example diagram without Global Accelerator:



With AWS Global Accelerator:



We can perform the Speed comparison using below link before deploying our application via Global accelerator.

<https://speedtest.globalaccelerator.aws/#/>

AWS Storage Gateway

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.



Gateway type

Choose gateway type

Amazon S3 File Gateway

Store files as objects in Amazon S3, with a local cache for low-latency access to your most recently used data.



Amazon FSx File Gateway

Low-latency on-premises access to fully managed, highly reliable, and virtually unlimited Windows file shares provided by Amazon FSx for Windows File Server.



Volume gateway

Block storage in Amazon S3 with point-in-time backups as Amazon EBS snapshots.



Tape gateway

Back up your data to Amazon S3 and archive in Amazon S3 Glacier using your existing tape-based processes.



- 1. Amazon S3 File Gateway :** presents a file interface that enables you to store files as objects in Amazon S3 using the industry-standard NFS and SMB file protocols, and access those files via NFS and SMB from your data center or Amazon EC2, or access those files as objects directly in

Amazon S3.

2. **Amazon FSx File Gateway** provides fast, low-latency on-premises access to fully managed, highly reliable, and scalable file shares in the cloud using the industry-standard SMB protocol. Customers can store and access file data in Amazon FSx with Windows-native compatibility including full NTFS support, shadow copies, and Access Control Lists (ACLs). Use Amazon FSx File Gateway for your on-premises file-based business applications and workloads such as user or group file shares, web content management, and media workflows.
3. **Tape Gateway** : presents a virtual tape library (VTL) consisting of virtual tape drives and a virtual media changer to your backup application using storage industry standard iSCSI protocol. You can continue to use your existing backup applications and workflows while writing to a nearly limitless collection of virtual tapes. Each virtual tape is stored in Amazon S3.
4. **Volume Gateway** : presents your applications block storage volumes using the iSCSI 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. You can back up your on-premises Volume Gateway volumes using the service's native snapshot scheduler or by using the AWS Backup service. In both cases, volume backups are stored as Amazon EBS snapshots in AWS. These snapshots are incremental backups that capture only changed blocks. All snapshot storage is also compressed to minimize your storage charges.

Refer to this video to configure AWS storagegateway :

<https://www.youtube.com/watch?v=wmcBSHpoHhs>

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.

Cloudtrail supports 3 types of events.

- **Management events** provide information about management operations that are performed on resources in your AWS account. These are also known as control plane operations.
- **Data events** provide information about the resource operations performed on or in a resource. These are also known as data plane operations. Data events are often high-volume activities. Data events can log s3, Dynamodb and Lambda services.
- **Insight Events** : CloudTrail Insights events capture unusual activity in your AWS account. If you have Insights events enabled, and CloudTrail detects unusual activities.

We can find the Cloudtrail under Management Tool in AWS dashboard.

- █ Management & Governance
- AWS Organizations
- CloudWatch
- AWS Auto Scaling
- CloudFormation
- █ CloudTrail
- Config

Here is the cloudTrail dashboard, By default we can view the last 90 days all events here.

The screenshot shows the AWS CloudTrail Dashboard. On the left, there's a sidebar with links like Dashboard, Event history, Insights, Trails, Pricing, Documentation, Forums, and FAQs. The main area has a breadcrumb navigation path: CloudTrail > Dashboard. The dashboard itself has two main sections: "Trails" and "Event history".

Trails: This section shows a table with one row for "codepipeline-source-trail". The "Status" column indicates it is "Logging". A "Create trail" button is also present.

Event history: This section shows a table with two rows of event data. The columns are "Event name", "Event time", and "Event source".

Event name	Event time	Event source
CreateCloudFrontOri...	July 14, 2021, 16:53:27 (UTC+0...)	cloudfront.amazonaws.com
Subscribe	July 14, 2021, 16:35:24 (UTC+0...)	sns.amazonaws.com

This screenshot shows the "Event history" page under the CloudTrail service. It displays a table of events from the last 90 days. The table includes columns for "Event name", "Event time", "User name", "Event source", and "Resource type".

Event name	Event time	User name	Event source	Resource type
CreateCloudFrontOri...	July 14, 2021, 16:53:27 (UTC+0...)	Avinash_T	cloudfront.amazonaws.com	-
Subscribe	July 14, 2021, 16:35:24 (UTC+0...)	Avinash_T	sns.amazonaws.com	AWS::SNS::Subscri...
CreateTopic	July 14, 2021, 16:31:19 (UTC+0...)	Avinash_T	sns.amazonaws.com	AWS::SNS::Topic
DeleteTopic	July 14, 2021, 16:03:20 (UTC+0...)	Avinash_T	sns.amazonaws.com	AWS::SNS::Topic
DeleteAlarms	July 14, 2021, 10:25:34 (UTC+0...)	Avinash_T	monitoring.amazonaws.com	AWS::CloudWatch...

If you want to store the logs more than 90 days, we need to create a Trail and need to copy into S3 bucket. Click on “Trail” option from left pane.

Select the “Create trail” option to start. And give a Trail Name

Choose the Management events you want to track (All/Read-Only/Write-only/None)

The screenshot shows the AWS CloudTrail Trails management interface. On the left, a sidebar menu includes options like Dashboard, Event history, Insights, and Trails (which is selected and highlighted in yellow). The main content area is titled 'Trails' and displays a table with columns: Name, Home region, Multi-region trail, Insights, Organization trail, S3 bucket, Log file prefix, CloudWatch Logs log group, and Status. A large message in the center states 'No trails' and 'No trails to display.' At the bottom right of the table area is a prominent orange 'Create trail' button.

Provide a Valid name for Trail and if you want to enable trail for AWS Organization, choose “Enable for all accounts in my Org”.

Choose trail attributes

This screenshot shows the 'General details' section of the CloudTrail Trail creation wizard. It includes a note that a trail created in the console is a multi-region trail, a 'Trail name' input field containing 'avinashdemoevents', and a checkbox for 'Enable for all accounts in my organization' which is unchecked. Below the checkbox is a link to 'See all accounts'.

We can store the Cloudtrail logs to S3 Bucket, Either an existing bucket or we can initiate to create a new bucket in account.

Storage location [Info](#)

Create new S3 bucket
Create a bucket to store logs for the trail.

Use existing S3 bucket
Choose an existing bucket to store logs for this trail.

Trail log bucket and folder
Enter a new S3 bucket name and folder (prefix) to store your logs. Bucket names must be globally unique.
aws-cloudtrail-logs-501170964283-f0bde81e
Logs will be stored in aws-cloudtrail-logs-501170964283-f0bde81e/AWSLogs/501170964283

Log file SSE-KMS encryption [Info](#)
 Enabled

▼ Additional settings

Log file validation [Info](#)
 Enabled

SNS notification delivery [Info](#)
 Enabled

Log file validation : To determine whether a log file was modified, deleted, or unchanged after CloudTrail delivered it, you can use CloudTrail log file integrity validation.

Choose the desired event.

Choose log events

Events [Info](#)
Record API activity for individual resources, or for all current and future resources in AWS account. [Additional charges apply](#)

Event type
Choose the type of events that you want to log.

Management events
Capture management operations performed on your AWS resources.

Data events
Log the resource operations performed on or within a resource.

Insights events
Identify unusual activity, errors, or user behavior in your account.

The screenshot shows the 'Management events' step of the AWS CloudTrail configuration wizard. It includes a note that no additional charges apply for logging management events because it's the first copy. Under 'API activity', 'Read' and 'Write' are selected, while 'Exclude AWS KMS events' and 'Exclude Amazon RDS Data API events' are not. At the bottom are 'Cancel', 'Previous', and 'Next' buttons.

Once you configure Review all the option and Click on “create trail”. We can see the logs in selected/created s3 bucket.

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 any resource at any point in time. These capabilities enable compliance auditing, security analysis, resource change tracking, and troubleshooting.

AWS Config is a service that enables us to assess, audit, and evaluate the configurations of our AWS resources. Config continuously monitors and records our AWS resource configurations and allows us to automate the evaluation of recorded configurations against desired configurations.

You will find the “Config” service under Management Tools.



Management & Governance

- AWS Organizations
- CloudWatch
- AWS Auto Scaling
- CloudFormation
- CloudTrail
- Config**
- OpsWorks

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.

Settings

General settings

Resource types to record

Record all resources supported in this region

Record specific resource types

To learn more, see [Supported Resource Types](#).

Include global resources (e.g., AWS IAM resources)
Supported global resource types are IAM users, groups, roles, and customer managed policies.

AWS Config role

Create AWS Config service-linked role

Choose a role from your account

Delivery method

Amazon S3 bucket

Create a bucket Choose a bucket from your account Choose a bucket from another account

Ensure appropriate permissions are available in this S3 bucket's policy. [Learn more](#).

S3 bucket name: config-bucket-501170964283 Prefix (optional): /AWSLogs/501170964283/Config/ap-south-1

Amazon SNS topic

Stream configuration changes and notifications to an Amazon SNS topic.
If you choose email as the notification endpoint for your SNS topic, this can cause a high volume of email. [Learn more](#).

Cancel **Next**

3. If you want to monitor any specific rule, you can select, otherwise you can choose or skip it.

Rules

AWS Managed Rules (143)			
	Name	Labels	Description
<input type="checkbox"/>	account-part-of-organizations	Organizations, Account	Rule checks whether AWS account is part of AWS Organizations. The rule is NON_COMPLIANT if the AWS account is not part of AWS Organizations or AWS Organizations master account ID does not match rule parameter MasterAccountId.
<input type="checkbox"/>	acm-certificate-expiration-check	ACM	Checks whether ACM Certificates in your account are marked for expiration within the specified number of days. Certificates provided by ACM are automatically renewed. ACM does not automatically renew certificates that you import.
<input type="checkbox"/>	alb-http-drop-invalid-header-enabled	ELBv2, http headers	Checks if rule evaluates AWS Application Load Balancers (ALB) to ensure they are configured to drop http headers. The rule is NON_COMPLIANT if the value of routing.http.drop_invalid_header_fields.enabled is set to false.

4. Review and click on confirm to complete the AWS config service setup.
5. Here is the Config service dashboard, you can choose the specific service and get the details about the changes, events happened.

The screenshot shows the AWS Config Dashboard. On the left, under 'Resource inventory', it says 'View the inventory of your AWS and non-AWS resources' and 'Learn more'. A dropdown menu shows 'All resources'. Below it, 'Total resources' is listed as 0. A note says 'Your resources are being discovered'. Another note says 'Run advanced queries against your resource configuration data.' On the right, under 'Compliance status', it shows 'Rules' with 0 Noncompliant rule(s) and 0 Compliant rule(s), and 'Resources' with 0 Noncompliant resource(s) and 0 Compliant resource(s). Below this, a section titled 'Noncompliant rules by noncompliant resource count' shows a table with no data, stating 'No noncompliant rules.'

- Let me navigate to S3 bucket to verify the logs, Log path looks similar to CloudTrail path.

The screenshot shows the AWS S3 console. The URL is 'Amazon S3 > config-bucket- / AWSLogs / 518084852393 / Config / ap-south-1 / 2018 / 1 / 24 / ConfigHistory'. There are tabs for 'Overview' and 'Details'. A search bar says 'Type a prefix and press Enter to search. Press ESC to clear.' Below are buttons for 'Upload', 'Create folder', and 'More'. The region is 'Asia Pacific (Mumbai)' and the view is 'Viewing 1 to 23'. A table lists log files with columns: Name, Last modified, Size, and Storage class. The log files are:

Name	Last modified	Size	Storage class
_Config_ap-south-1_ConfigHistory_AWS::CloudFormation::Sta...	Jan 24, 2018 5:11:34 PM GMT+0530	394.0 B	Standard
_Config_ap-south-1_ConfigHistory_AWS::EC2::EIP_20180124...	Jan 24, 2018 11:11:35 AM GMT+0530	417.0 B	Standard
_Config_ap-south-1_ConfigHistory_AWS::EC2::Instance_2018...	Jan 24, 2018 11:11:35 AM GMT+0530	1.5 KB	Standard
_Config_ap-south-1_ConfigHistory_AWS::EC2::Instance_2018...	Jan 24, 2018 5:11:34 PM GMT+0530	1.2 KB	Standard
_Config_ap-south-1_ConfigHistory_AWS::EC2::Instance_2018...	Jan 24, 2018 11:11:41 PM GMT+0530	1.4 KB	Standard

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

AWS Cloud Formation

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 lets testing teams quickly create a clean environment to run tests without disturbing ongoing 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:

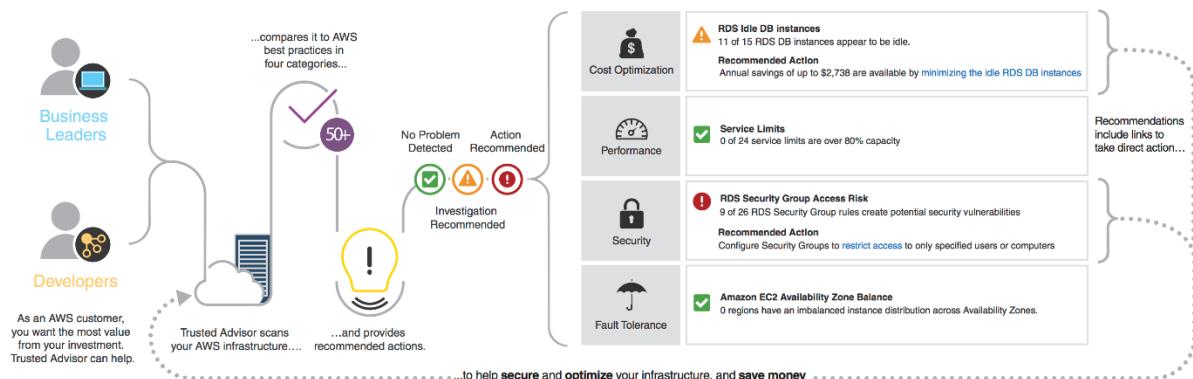
AWS trusted advisor is an online resource to help us to reduce cost, increase performance, and improve security by optimizing AWS environment.

It gives suggestion for

1. Cost Optimization

2. Performance
3. Security
4. Fault Tolerance
5. Service Limit

An Introduction to AWS Trusted Advisor



We can find the Trusted Advisor under Management tools

Management Tools

- CloudWatch
- AWS Auto Scaling
- CloudFormation
- CloudTrail
- Config
- OpsWorks
- Service Catalog
- Systems Manager
- Trusted Advisor**
- Managed Services

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

Trusted Advisor Dashboard



Cost Optimization



0 ✓ 0 ⚠
0 ⓘ

Performance



0 ✓ 0 ⚠
0 ⓘ

Security



4 ✓ 0 ⚠
1 ⓘ

Fault Tolerance



0 ✓ 0 ⚠
0 ⓘ

Service Limits



39 ✓ 0 ⚠
0 ⓘ

Recommended Actions

▶ ! Security Groups - Specific Ports Unrestricted	<small>Refreshed: 3 minutes ago</small>
Checks security groups for rules that allow unrestricted access (0.0.0.0/0) to specific ports. 15 of 28 security group rules allow unrestricted access to a specific port.	
▶ ✓ RDS Max Auths per Security Group	<small>Refreshed: 3 minutes ago</small>
Checks for usage that is more than 80% of the RDS Max Auths per Security Group Limit. 0 of 0 items have usage that is more than 80% of the service limit.	

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.

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 Hadoop and, combined with the scale of AWS gives you the ability to spinup large Hadoop clusters instantly and start processing with in minutes.

Use Cases 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:

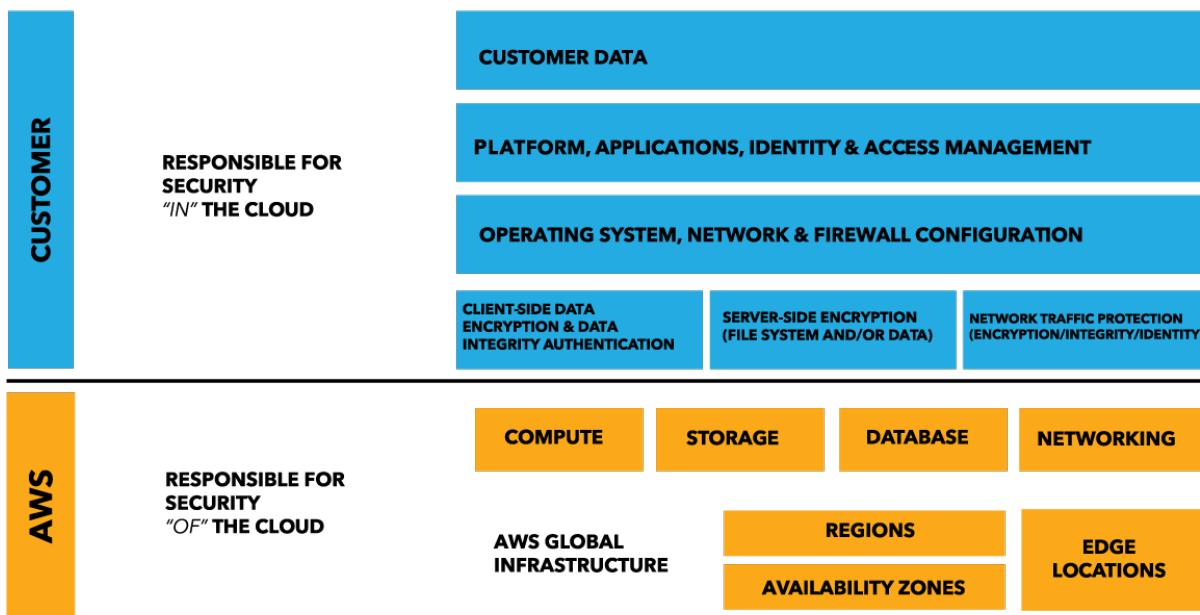
AWS Data Pipeline is a web service that helps you reliably process and move data between different AWS compute and storage services, and also on-premises data sources, at specified intervals. With AWS Data Pipeline, you can regularly access your data where it's stored, transform and process it at scale, and efficiently transfer the results to AWS services such as Amazon S3, Amazon Relational Database Service (AmazonRDS), Amazon Dynamo DB, and Amazon EMR.

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.