AWS DOCUMENT - SYSOPS

Sreejesh KV

Table of Content

VIRTUALIZATION AND CLOUD COMPUTING

VIRTUALIZATION

Virtualization is the fundamental technology that powers cloud computing. This software separates compute environments from physical infrastructures, so you can run multiple operating systems and applications simultaneously on the same machine. For example, if you do most of your work on a Mac but use select applications that are exclusive to PCs, you can run Windows on a virtual machine to get access to those applications without having to switch computers.

CLOUD COMPUTING

Virtualization is software that manipulates hardware, while cloud computing refers to a service that results from that manipulation. You can't have cloud computing without virtualization.

Cloud computing gives users access to data wherever they have an internet connection. In today's ever-changing business climate, it's critical that small business owners get what they need right when they need it, whether they're on their computers, tablets or mobile phones – or in the office, out in the field or on the road. This is exactly the convenience that cloud computing provides.

HYPERVISOR

A hypervisor is a function which abstracts -- isolates -- operating systems and applications from the underlying computer hardware. This abstraction allows the underlying host machine hardware to independently operate one or more virtual machines as guests, allowing multiple guest VMs to effectively share the system's physical compute resources, such as processor cycles, memory space, network bandwidth and so on. A hypervisor is sometimes also called a virtual machine monitor.

BENEFITS OF CLOUD COMPUTING

Flexibility: Cloud-based services are ideal for businesses with growing or fluctuating bandwidth demands. If your needs increase it’s easy to scale up your cloud capacity, drawing on the service’s remote servers.

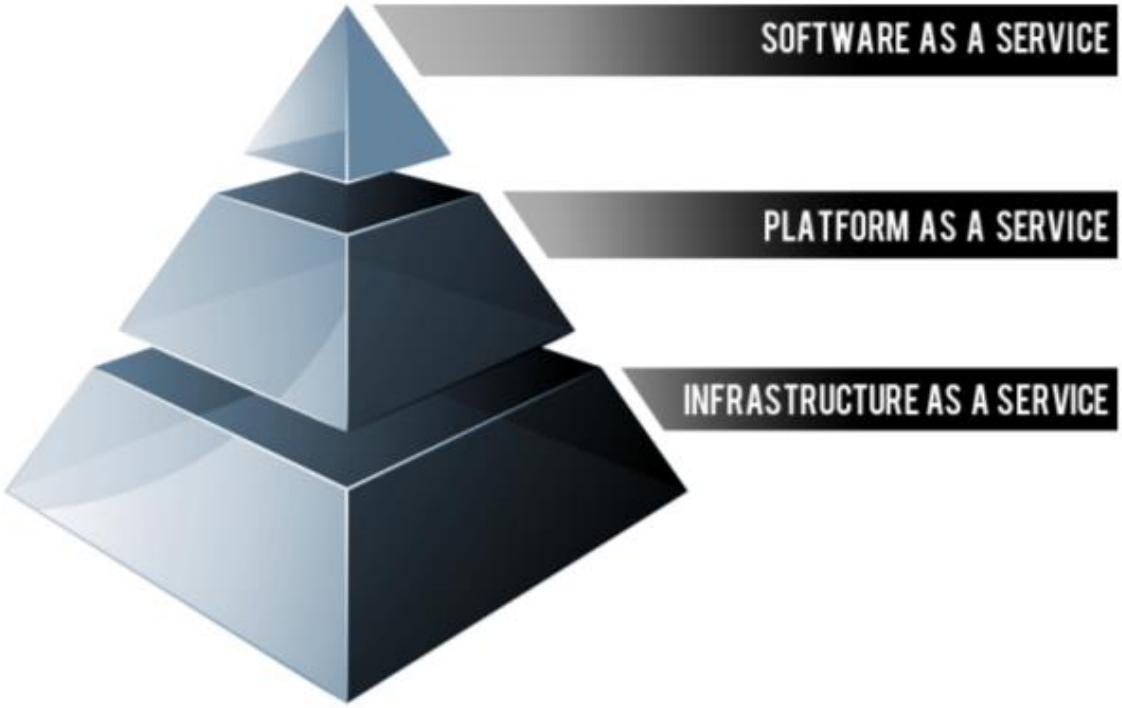
Capital-expenditure Free: Cloud computing cuts out the high cost of hardware. You simply pay as you go and enjoy a subscription-based model that’s kind to your cash flow.

Work from anywhere: With cloud computing, if you’ve got an internet connection you can be at work. And with most serious cloud services offering mobile apps.

Security: Cloud computing gives you greater security. Your data is stored in the cloud; you can access it no matter what happens to your machine.

CLOUD COMPUTING MODELS

Companies can use cloud computing to increase their IT functionality or capacity without having to add software, personnel, invest in additional training or set up new infrastructure. Below are the major types of cloud computing.



IAAS (INFRASTRUCTURE AS A SERVICE)

Provides you the computing infrastructure, physical or (quite often) virtual machines and other resources like virtual-machine disk image library, block and file-based storage, firewalls, load balancers, IP addresses, virtual local area networks etc.

Examples: Amazon EC2, Windows Azure, Rackspace, Google Compute Engine.

PAAS (PLATFORM AS A SERVICE)

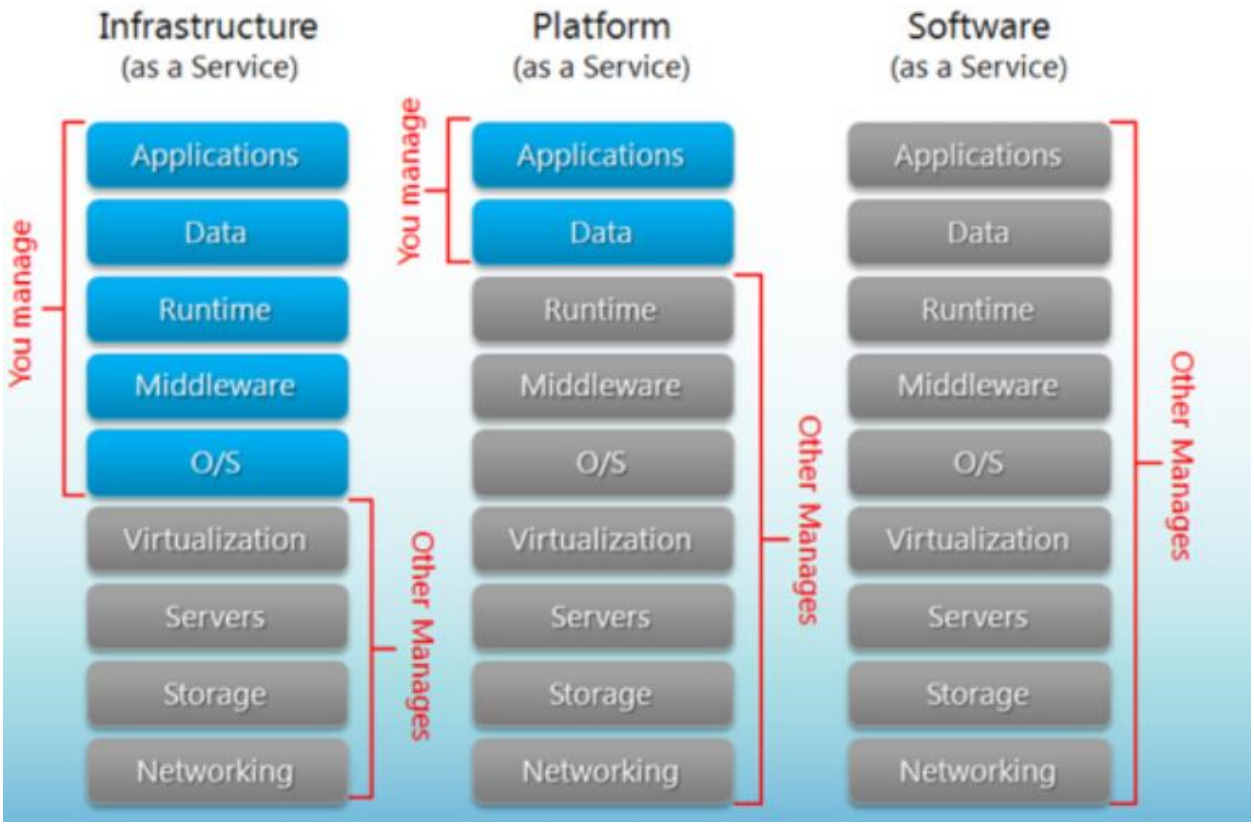
Provides you computing platforms which typically includes operating system, programming language execution environment, database, web server etc.

Examples: AWS Elastic Beanstalk, Windows Azure, Heroku, Force.com, Google App Engine, Apache Stratos.

SAAS (SOFTWARE AS A SERVICE)

In this model you are provided with access to application software often referred to as "on-demand software". You don't have to worry about the installation, setup and running of the application. Service provider will do that for you. You just have to pay and use it through some client.

Examples: Google Apps, Microsoft Office 365



CLOUD COMPUTING DEPLOYMENT MODELS

PUBLIC CLOUD

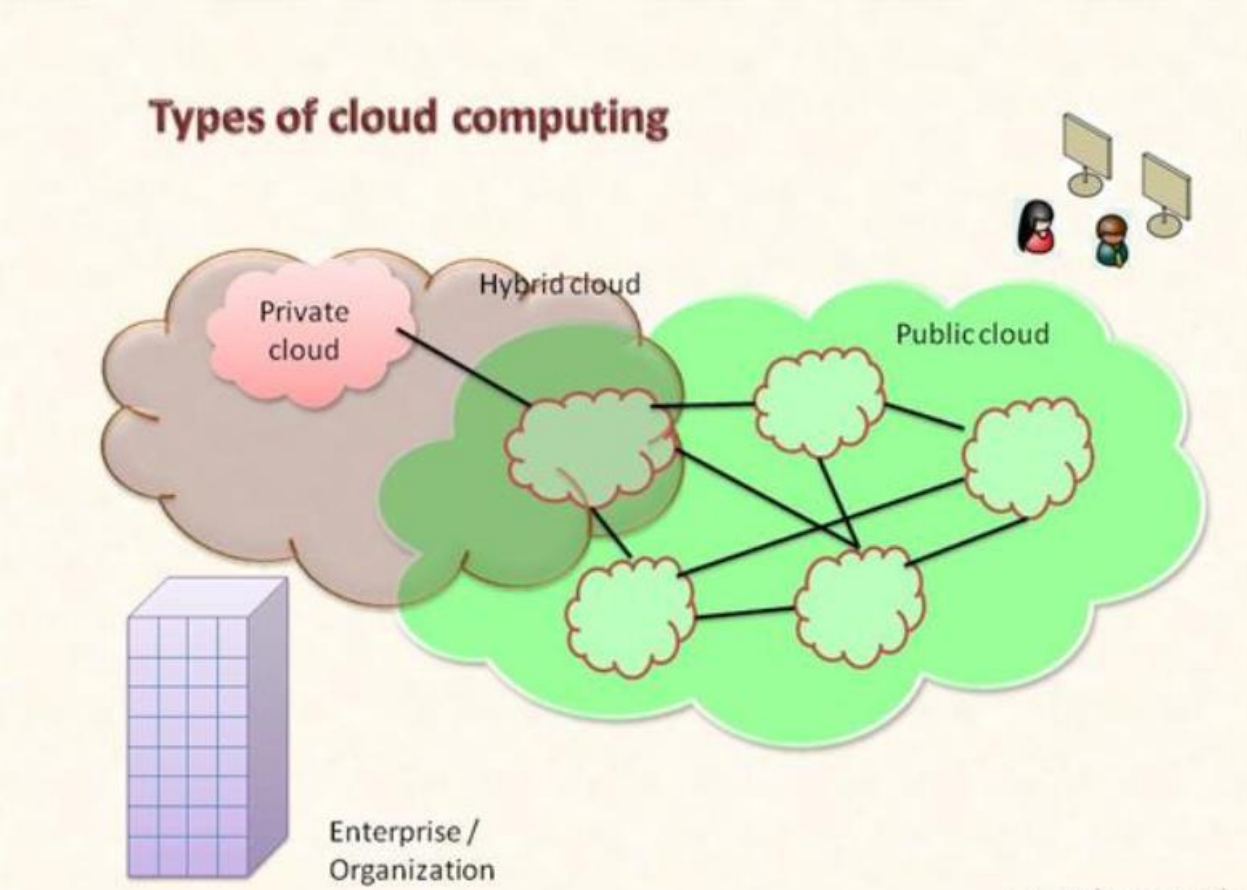
A public cloud is basically the internet. Service providers use the internet to make resources, such as applications (also known as Software-as-a-service) and storage, available to the general public, or on a 'public cloud. Examples of public clouds include Amazon Elastic Compute Cloud (EC2), IBM's Blue Cloud, Sun Cloud, Google AppEngine and Windows Azure Services Platform

PRIVATE CLOUD

Private clouds are data center architectures owned by a single company that provides flexibility, scalability, provisioning, automation and monitoring. The goal of a private cloud is not sell "as-a-service" offerings to external customers but instead to gain the benefits of cloud architecture without giving up the control of maintaining your own data center.

HYBRID CLOUD

By using a Hybrid approach, companies can maintain control of an internally managed private cloud while relying on the public cloud as needed. For instance, during peak periods individual applications, or portions of applications can be migrated to the Public Cloud. This will also be beneficial during predictable outages: hurricane warnings, scheduled maintenance windows, rolling brown/blackouts.



HORIZONTAL & VERTICAL SCALING

VERTICAL SCALING

To increase the capacity if we increase resources in same logical unit/server then it is vertical scaling. E.g. Add more CPUs in existing sever. If system is not handled by one CPU then increase the CPU to 3 or 4. Another example is server is having 8 GB RAM then scale it to 16 GB. Same applicable to storages also.

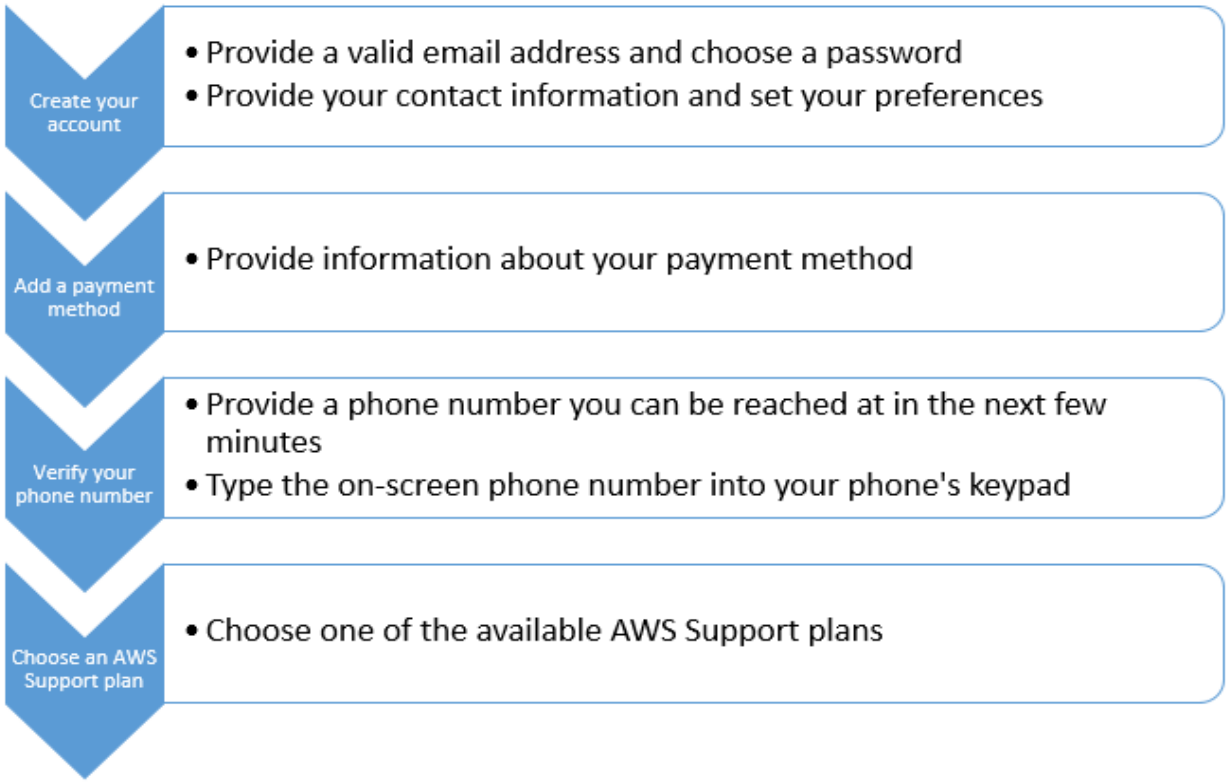
HORIZONTAL SCALING

Horizontal scaling means enhancing the performance of server /node by adding more instances of server to your pool of servers so that load can be spread.

AWS CLOUD PLATFORM

CREATING NEW AWS ACCOUNT

Below are the primary steps to sign up new AWS account.

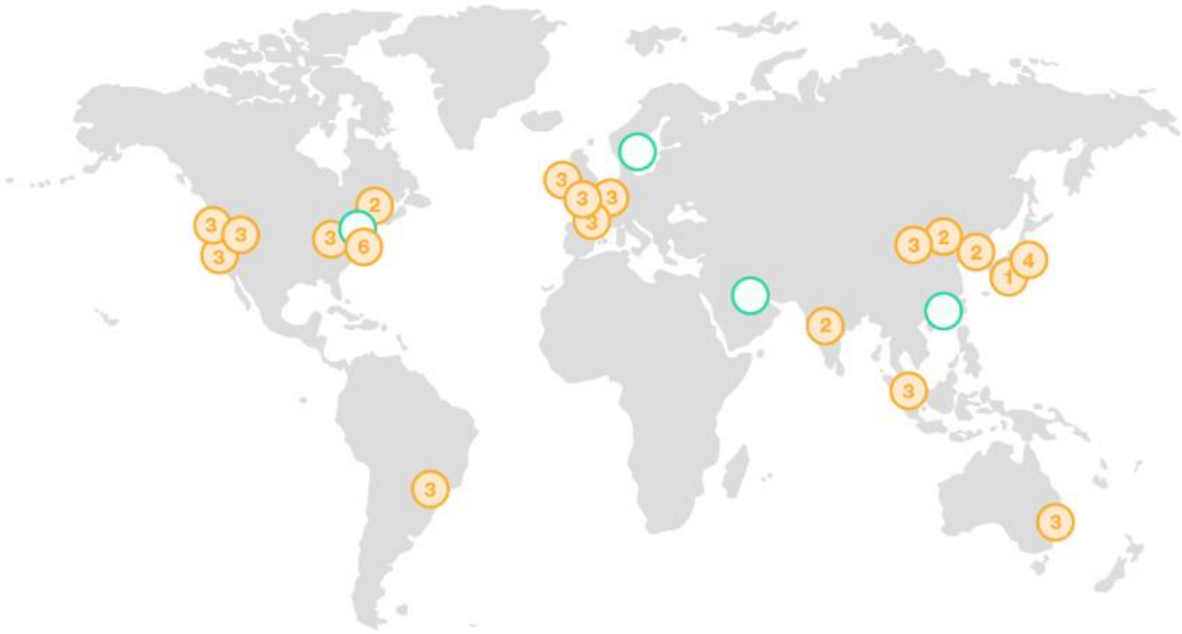


AWS GLOBAL INFRASTRUCTURE

AWS REGIONS AND AVAILABILITY ZONE

The AWS Cloud infrastructure is built around Regions and Availability Zones (AZs). AWS Regions provide multiple, physically separated and isolated Availability Zones which are connected with low latency, high throughput, and highly redundant networking. These Availability Zones offer AWS customers an easier and more effective way to design and operate applications and databases, making them more highly available, fault tolerant, and scalable than traditional single datacenter infrastructures or multi-datacenter infrastructures. For customers who specifically need to replicate their data or applications over greater geographic distances, there are AWS Local Regions. An AWS Local Region is a single datacenter designed to complement an existing AWS Region. Like all AWS Regions, AWS Local Regions are completely isolated from other AWS Regions. The AWS Cloud spans 55 Availability Zones within 18 geographic Regions and one Local Region around the world.

Below diagram represent the AWS global infrastructure where the regions are available.



Region & Number of Availability Zones

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

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

Asia Pacific
Mumbai (2),
Seoul (2),
Singapore (3),
Sydney (3),
Tokyo (4),
Osaka-Local (1)[1]

Canada
Central (2)

China
Beijing (2),
Ningxia (3)

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

South America
São Paulo (3)

AWS GovCloud (US-West) (3)

New Region (coming soon)

Bahrain

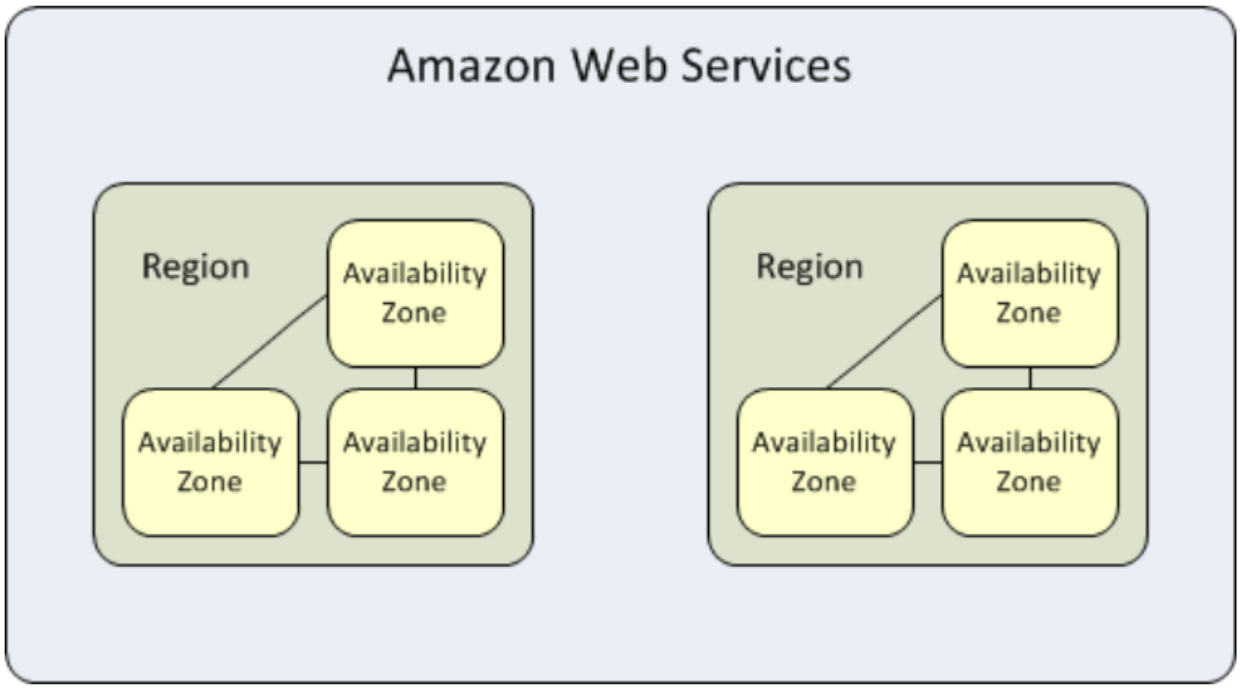
Hong Kong SAR, China

Sweden

AWS GovCloud (US-East)

- Each AWS Region is a separate geographic area.
- Each AWS Region has multiple, isolated locations known as Availability Zones
- AWS regions are fully isolated from other AWS Regions.
- Each region will have minimum 2 availability zone for high availability
- AWS will not replicate resources between regions automatically.

The following diagram illustrates the relationship between regions and Availability Zones.



AVAILABLE REGIONS

Your account determines the regions that are available to you. For example:

An AWS account provides multiple regions so that you can launch Amazon EC2 instances in locations that meet your requirements. For example, you might want to launch instances in Europe to be closer to your European customers or to meet legal requirements.

An AWS GovCloud (US) account provides access to the AWS GovCloud (US) region only.

An Amazon AWS (China) account provides access to the Beijing and Ningxia Regions only.

The following table lists the regions provided by an AWS account. You can't describe or access additional regions from an AWS account, such as AWS GovCloud (US) or the China Regions.

Code	Name
us-east-1	US East (N. Virginia)
us-east-2	US East (Ohio)
us-west-1	US West (N. California)
us-west-2	US West (Oregon)
ca-central-1	Canada (Central)
eu-central-1	EU (Frankfurt)
eu-west-1	EU (Ireland)
eu-west-2	EU (London)
eu-west-3	EU (Paris)
ap-northeast-1	Asia Pacific (Tokyo)
ap-northeast-2	Asia Pacific (Seoul)
ap-northeast-3	Asia Pacific (Osaka-Local)
ap-southeast-1	Asia Pacific (Singapore)
ap-southeast-2	Asia Pacific (Sydney)
ap-south-1	Asia Pacific (Mumbai)
sa-east-1	South America (São Paulo)

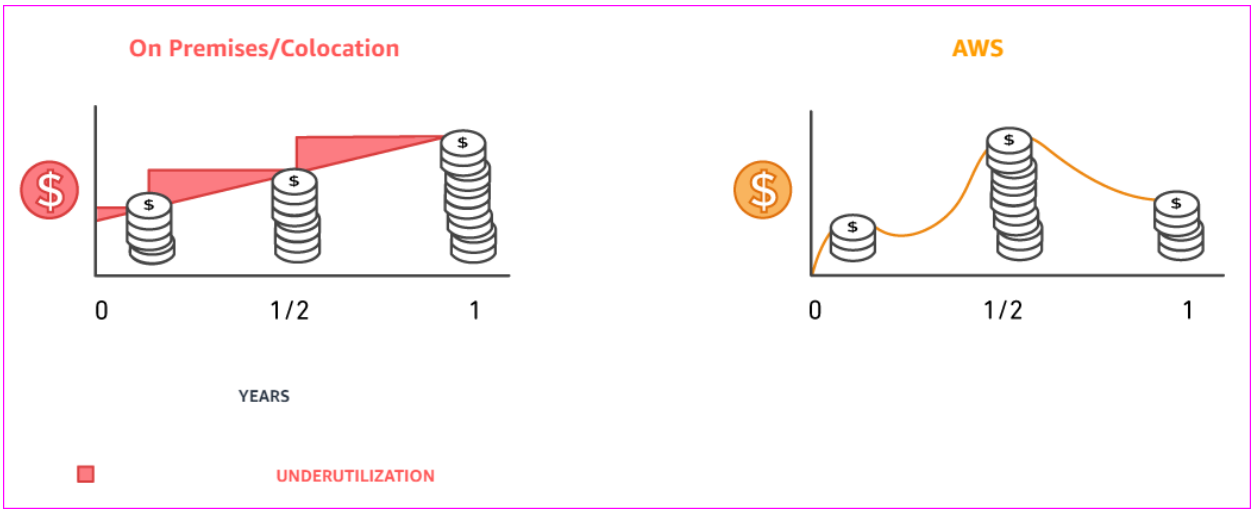
AWS COSTING MODEL

PAY-AS-YOU-GO

With AWS you only pay for what use, helping your organization remain agile, responsive and always able to meet scale demands.

Pay-as-you-go pricing allows you to easily adapt to changing business needs without overcommitting budgets and improving your responsiveness to changes. With a pay as you go model, you can adapt your business depending on need and not on forecasts, reducing the risk or overprovisioning or missing capacity.

By paying for services on an as needed basis, you can redirect your focus to innovation and invention, reducing procurement complexity and enabling your business to be fully elastic.

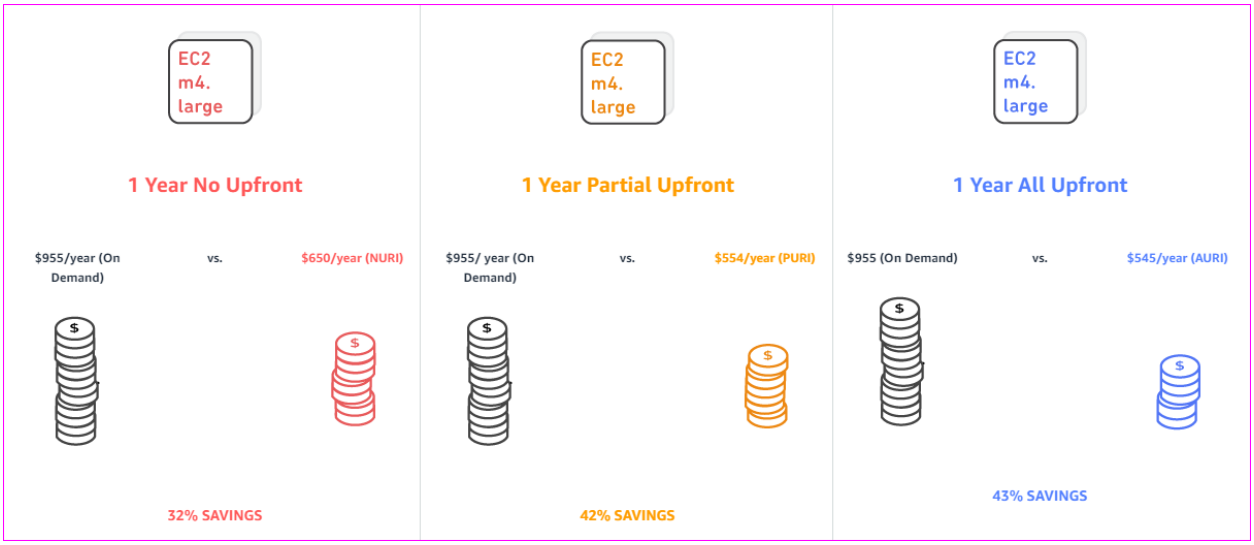


SAVE WHEN YOU RESERVE

For certain services like Amazon EC2 and Amazon RDS, you can invest in reserved capacity. With Reserved Instances, you can save up to 75% over equivalent on-demand capacity. Reserved Instances are available in 3 options – All up-front (AURI), partial up-front (PURI) or no upfront payments (NURI).

When you buy Reserved Instances, the larger the upfront payment, the greater the discount. To maximize your savings, you can pay all up-front and receive the largest discount. Partial up-front RI's offer lower discounts but give you the option to spend less up front. Lastly, you can choose to spend nothing up front and receive a smaller discount, but allowing you to free up capital to spend in other projects.

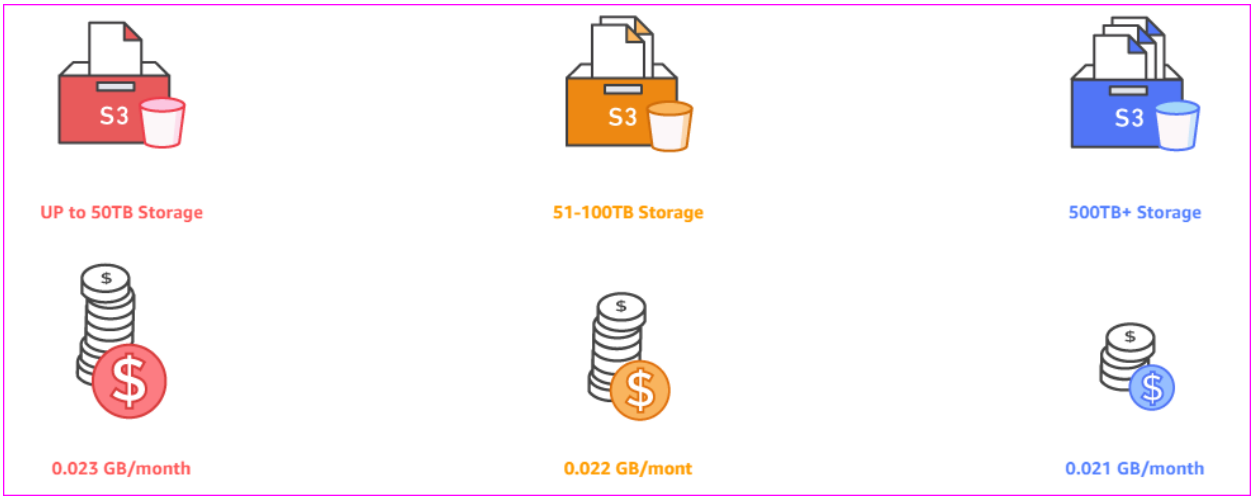
By using reserved capacity, your organization can minimize risks, more predictably manage budgets, and comply with policies that require longer-term commitments.



PAY LESS BY USING MORE

With AWS, you can get volume based discounts and realize important savings as your usage increases. For services such as S3 and data transfer OUT from EC2, pricing is tiered, meaning the more you use, the less you pay per GB. In addition, data transfer IN is always free of charge. As a result, as your AWS usage needs increase, you benefit from the economies of scale that allow you to increase adoption and keep costs under control.

As your organization evolves, AWS also gives you options to acquire services that help you address your business needs. For example, AWS' storage services portfolio, offers options to help you lower pricing based on how frequently you access data, and the performance you need to retrieve it. To optimize your savings, choose the right combinations of storage solutions that help you reduce costs while preserving performance, security and durability.



AWS SUPPORT MODEL

All customers receive Basic Support included with your AWS account. All plans, including Basic Support, provide 24x7 access to customer service, AWS documentation, whitepapers, and support forums.

For access to technical support and additional Support resources, we offer plans to fit your unique needs.

- Basic
- Developer
- Business
- Enterprise

	Basic	Developer	Business	Enterprise
Customer Service and Communities	24x7 access to customer service, documentation, whitepapers, and support forums	24x7 access to customer service, documentation, whitepapers, and support forums	24x7 access to customer service, documentation, whitepapers, and support forums	24x7 access to customer service, documentation, whitepapers, and support forums
Health status and Notifications	Access to Personal Health Dashboard	Access to Personal Health Dashboard	Access to Personal Health Dashboard & Health API	Access to Personal Health Dashboard & Health API
Technical Support		Business hours** access to Cloud Support Associates via email	24x7 access to Cloud Support Engineers via email, chat & phone	24x7 access to Sr. Cloud Support Engineers via email, chat & phone
Who Can Open Cases		One primary contact/ Unlimited cases	Unlimited contacts/ Unlimited cases (IAM supported)	Unlimited contacts/ Unlimited cases (IAM supported)
Case Severity/ Response Times*		General guidance: < 24 business hours System impaired: < 12 business hours	General guidance: < 24 hours System impaired: < 12 hours Production system impaired: < 4 hours Production system down: < 1 hour	General guidance: < 24 hours System impaired: < 12 hours Production system impaired: < 4 hours Production system down: < 1 hour Business-critical system down: < 15 minutes
Proactive Guidance				Designated Technical Account Manager
Pricing	Included	Starts at $29 per month	Starts at $100 per month	Starts at $15k per month

AWS INSTANCE PURCHASING OPTIONS

Amazon EC2 provides the following purchasing options to enable you to optimize your costs based on your needs:

- **On-Demand Instances** – Pay, by the second or per hour, for the instances that you launch.
- **Reserved Instances** – Purchase, at a significant discount, instances that are always available, for a term from one to three years.
- **Scheduled Instances** – Purchase instances that are always available on the specified recurring schedule, for a one-year term.
- **Spot Instances** – Request unused EC2 instances, which can lower your Amazon EC2 costs significantly.
- **Dedicated Hosts** – Pay for a physical host that is fully dedicated to running your instances, and bring your existing per-socket, per-core, or per-VM software licenses to reduce costs.
- **Dedicated Instances** – Pay, by the hour, for instances that run on single-tenant hardware.

EC2 INSTANCES

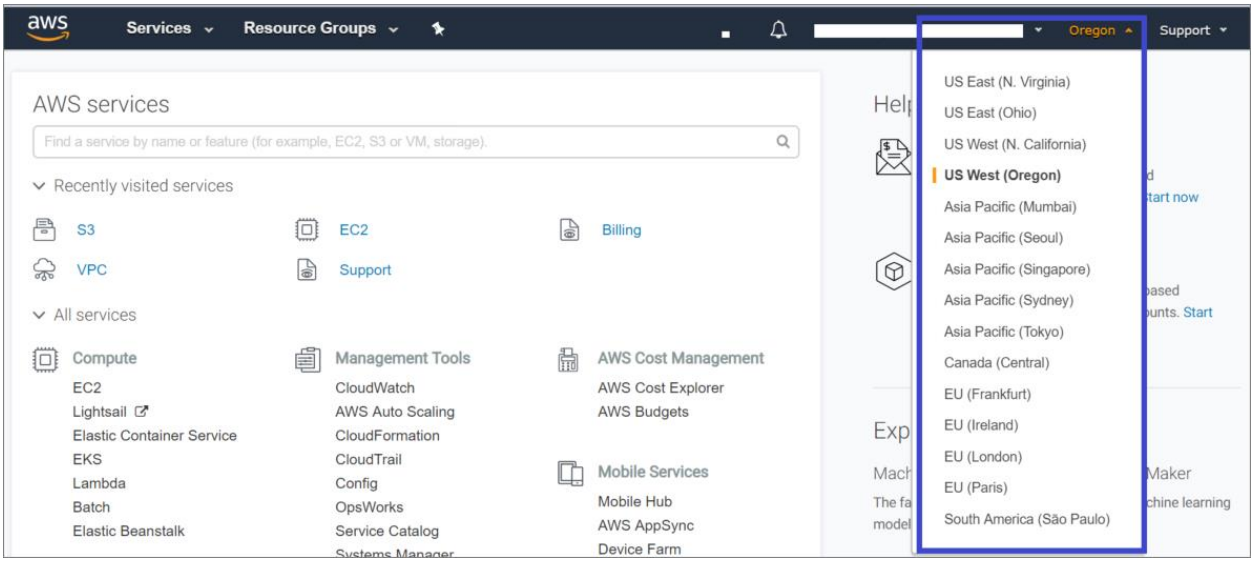
Amazon Elastic Compute Cloud (Amazon EC2) is a web service that provides secure, resizable compute capacity in the cloud. It is designed to make web-scale cloud computing easier for developers.

Amazon EC2's simple web service interface allows you to obtain and configure capacity with minimal friction. It provides you with complete control of your computing resources and lets you run on Amazon's proven computing environment. Amazon EC2 reduces the time required to obtain and boot new server instances to minutes, allowing you to quickly scale capacity, both up and down, as your computing requirements change. Amazon EC2 changes

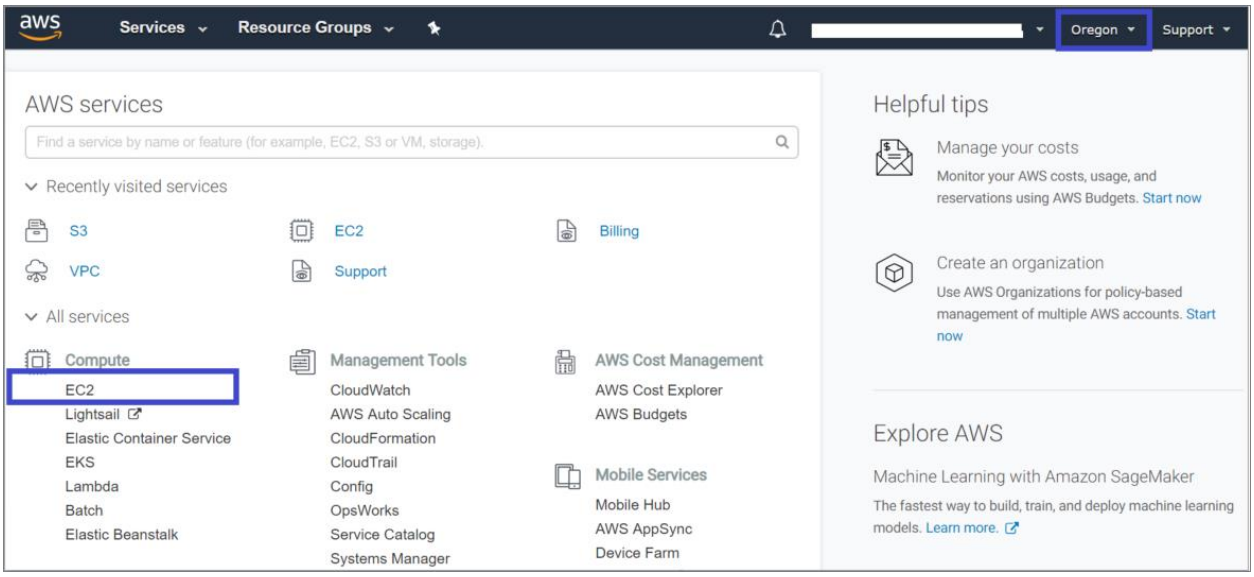
the economics of computing by allowing you to pay only for capacity that you actually use. Amazon EC2 provides developers the tools to build failure resilient applications and isolate them from common failure scenarios.

LAUNCHING AN EC2 INSTANCE

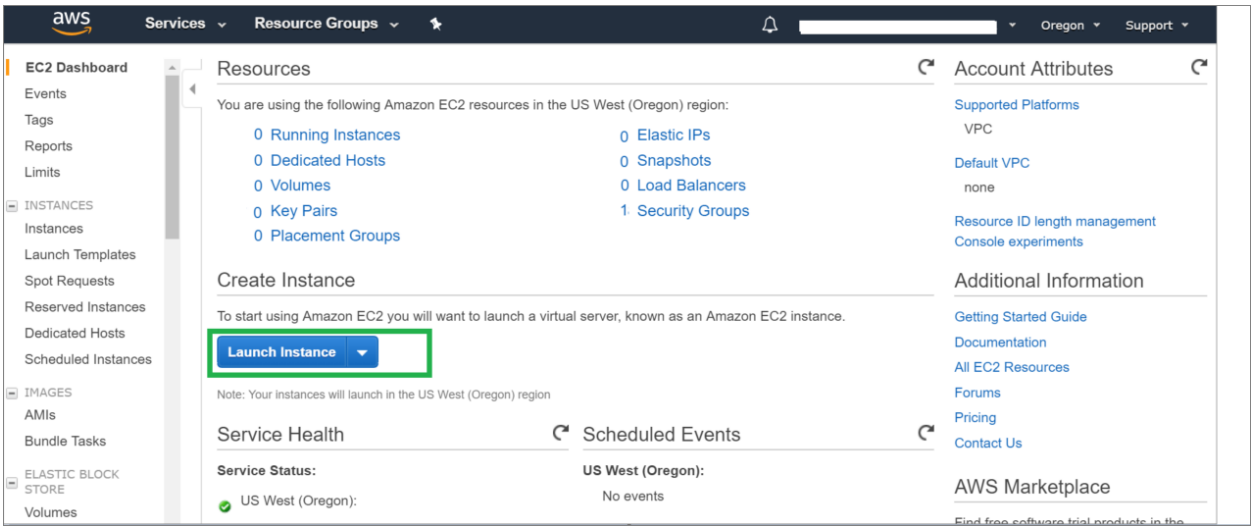
1. Once you logged in to AWS account, select the region from dropdown to launch the server in respective region.



2. Once the region is selected, select EC2 which is under Compute option. In this example, we are going to launch the server in Oregon region.



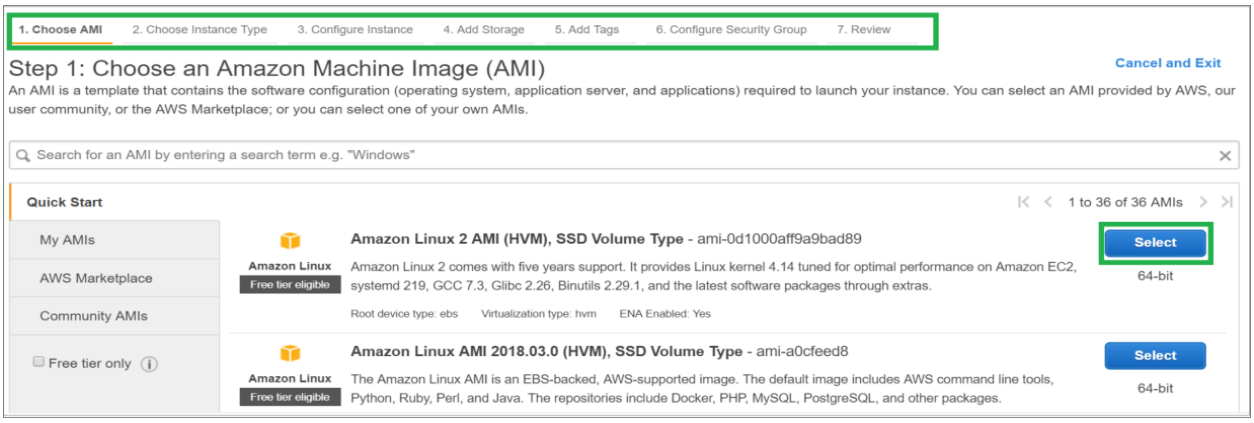
3. From the EC2 instance dashboard, click on Launch instance to create an instance.



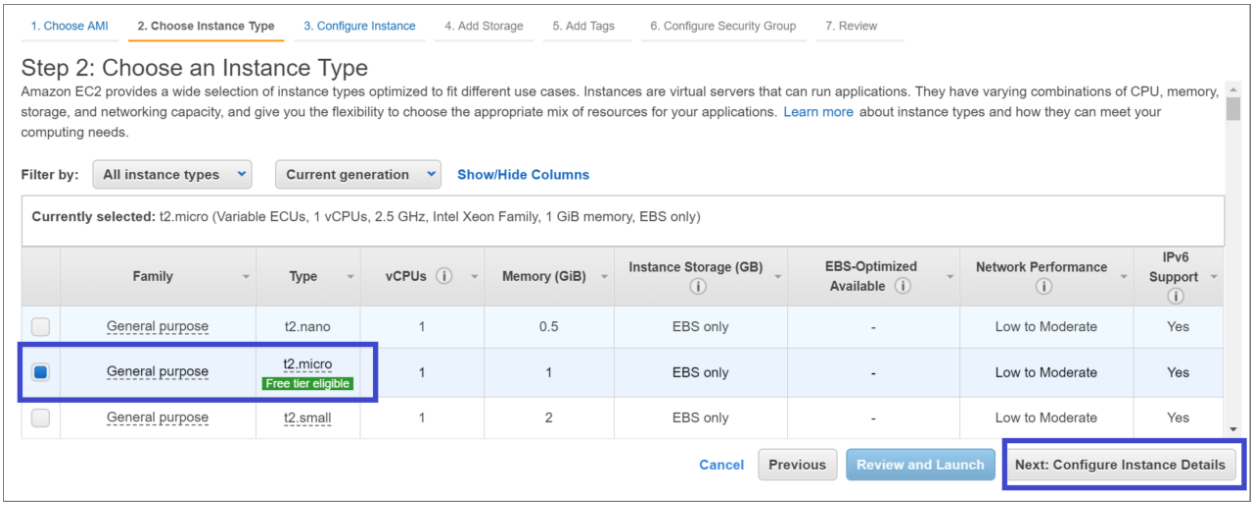
4. Need to follow the below 7 steps to complete the instance launching. In this first step is – Choose AMI.

AMI (Amazon machine image): This is the image copy of an Operating system.

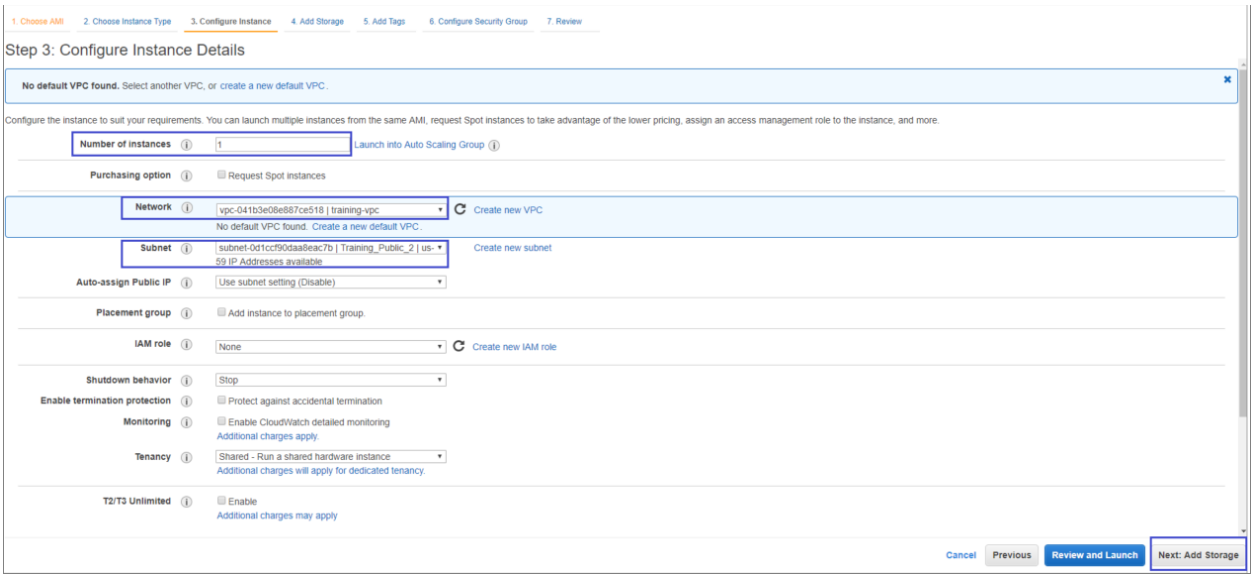
In this example, we are going to create a Linux machine.



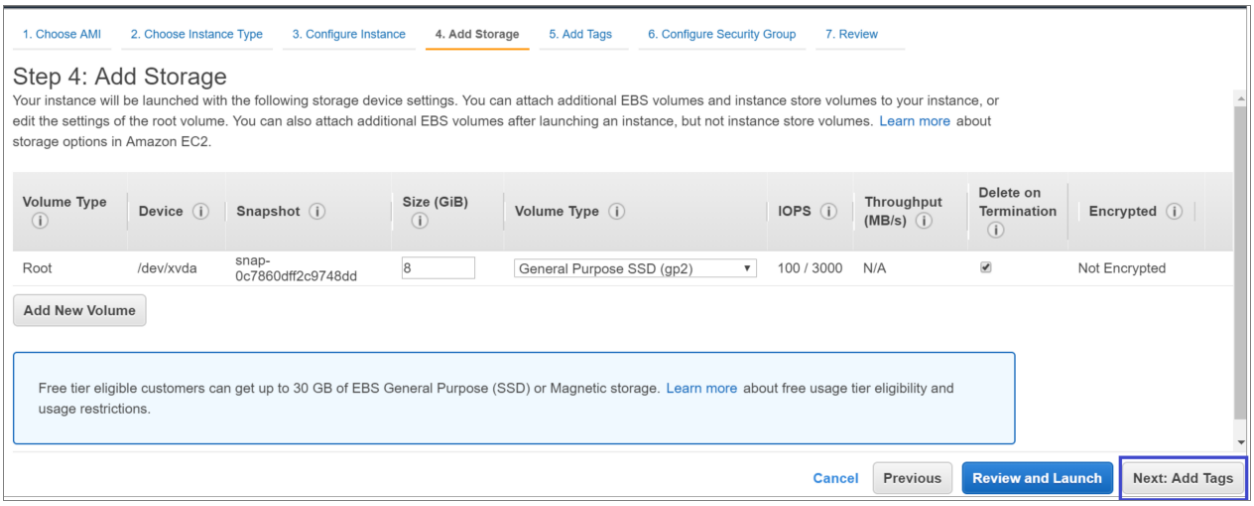
5. As we are in Free tier account, so selecting “t2.micro” instance type(Free tier eligible).



6. Select the details for below fields.

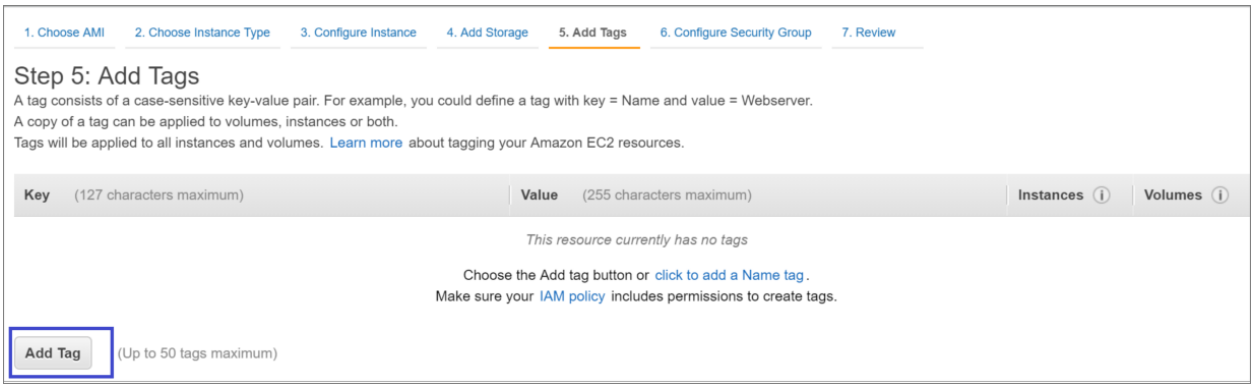


7. Add the storage – in this it is came with default root volume size. We can increase the size and we can add additional volumes.

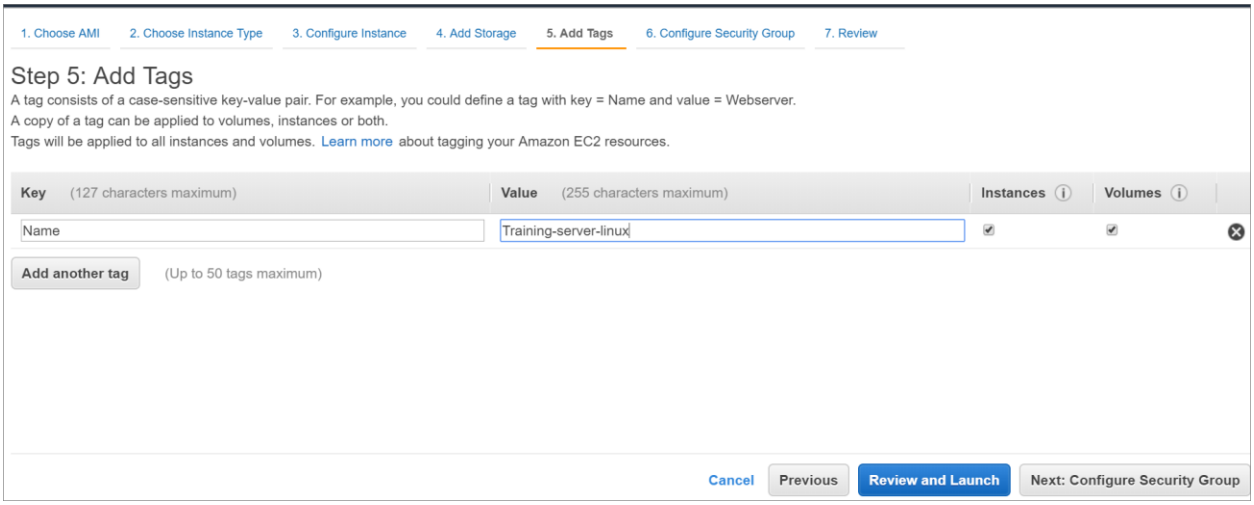


8. Click on Add Tag in below window to add the identification name.

We have option to add multiple tags for billing purpose or some other identification purpose.



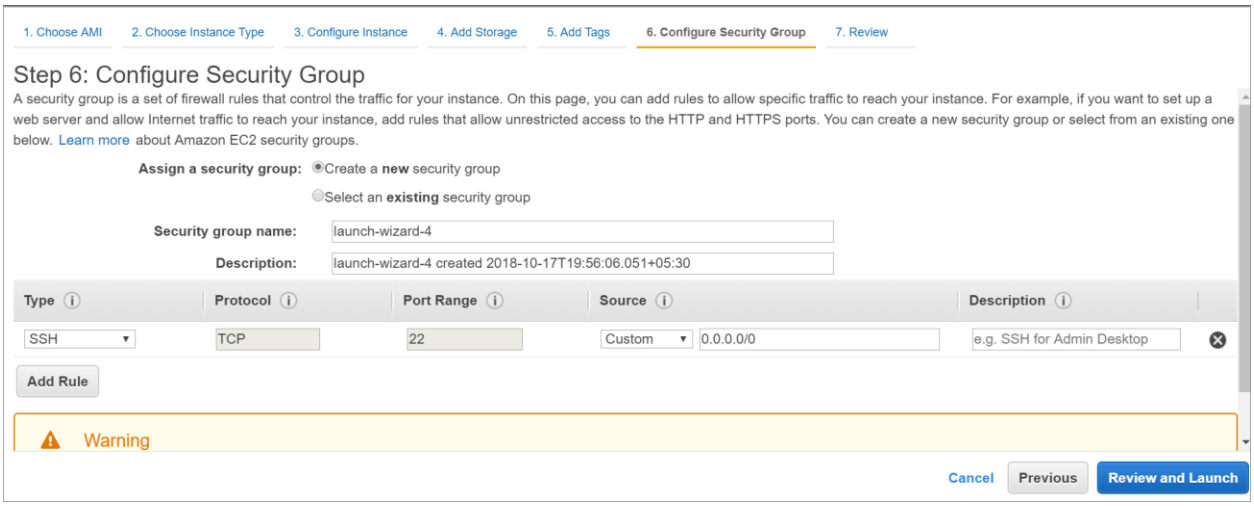
9. Then click on Configure security group.



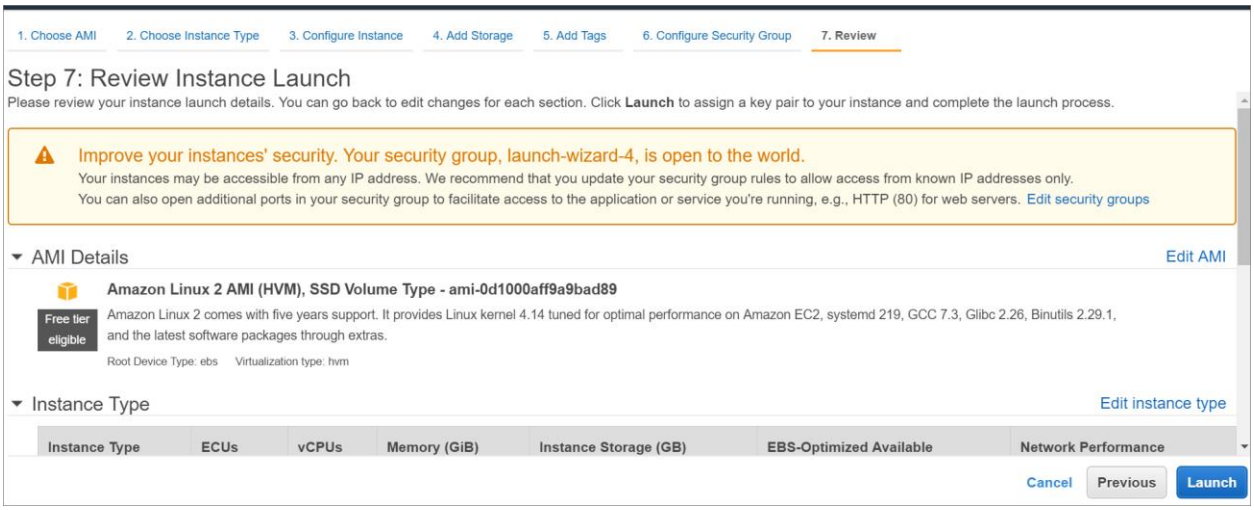
10. In this option, either we can create a new security group or we can choose the existing security group.

Below it came default with SSH with port 22 as it is a Linux server. So for connecting a Linux server remotely, we are using SSH protocol type and port number 22.

Then click Review and Launch



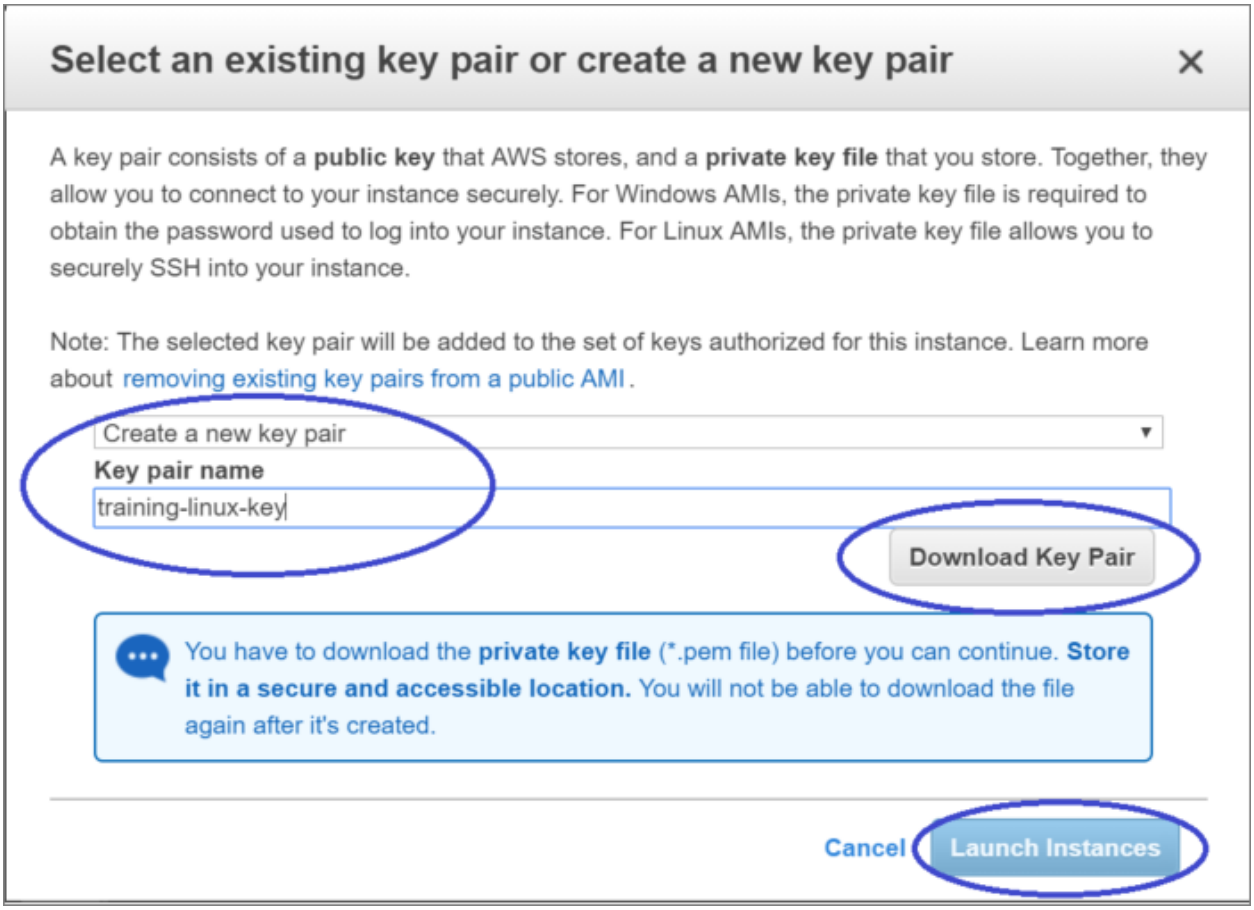
11. It will ask to review the selected configurations before launching. Review once and click Launch.



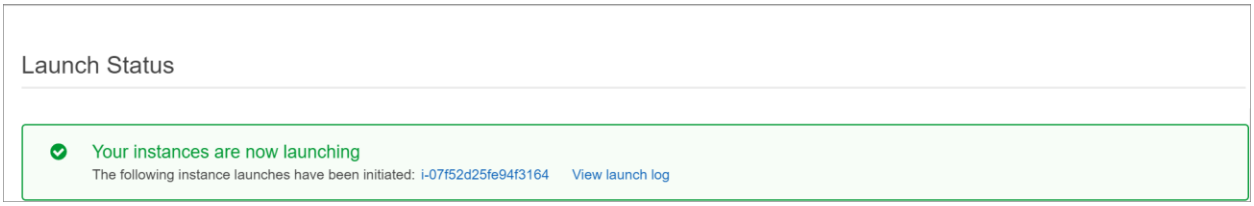
12. A popup will come to Select an existing key pair or to create a new key pair.

Key pair: This is the key which we will use to connect/authenticate the Linux server while connecting through SSH. Insert the key name and then download the keypair. This will be in ".pem" format.

Note :If you are creating a windows server also, this key pair need to be downloaded. If you have existing key pair with you, you can select existing key pair.

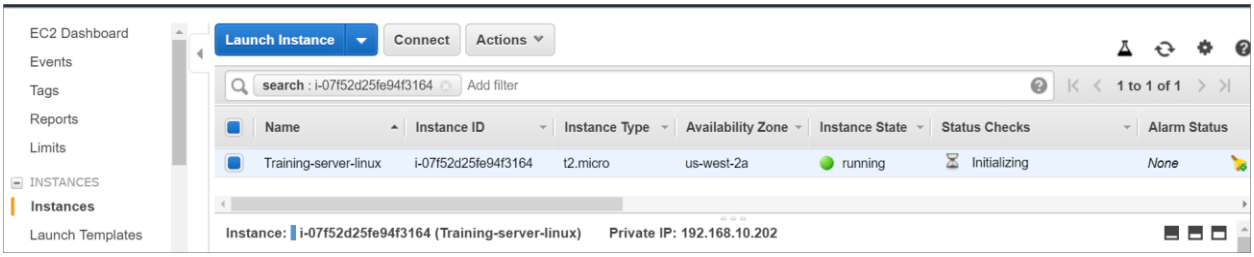


13. Once click on Launch Instances, it will navigate to next window with instance ID.

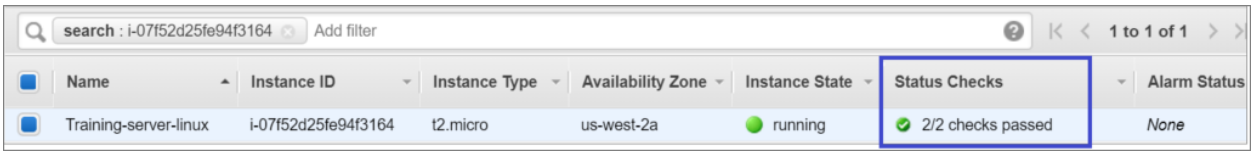


14. Once click on the instance ID, it will take to a new window which will have the instance which is in launching progress.

Keep an eye on Status Checks until it come to 2/2 Checks Passed.



Now the server Status Check is 2/2. Means it is up and running. We are ready to connect it.

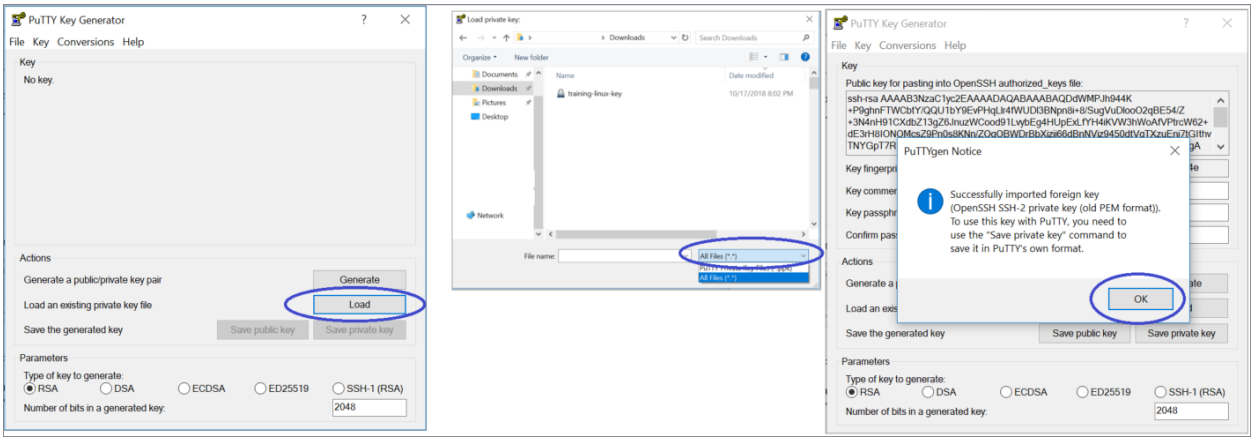


15. For connecting the Linux machine from a windows machine,We need to convert the .pem keypair to .ppk format. Follow the below step for converting it.

a. Download “Puttygen” software and open it.

b. Click “Load” and browse for the .pem key [Select All files(*.*) to show up the pem file] pair which we downloaded while launching the server.

c. Click “**Save private key**” then provide a name. It will save the key in .ppk format. Then close puttygeen.

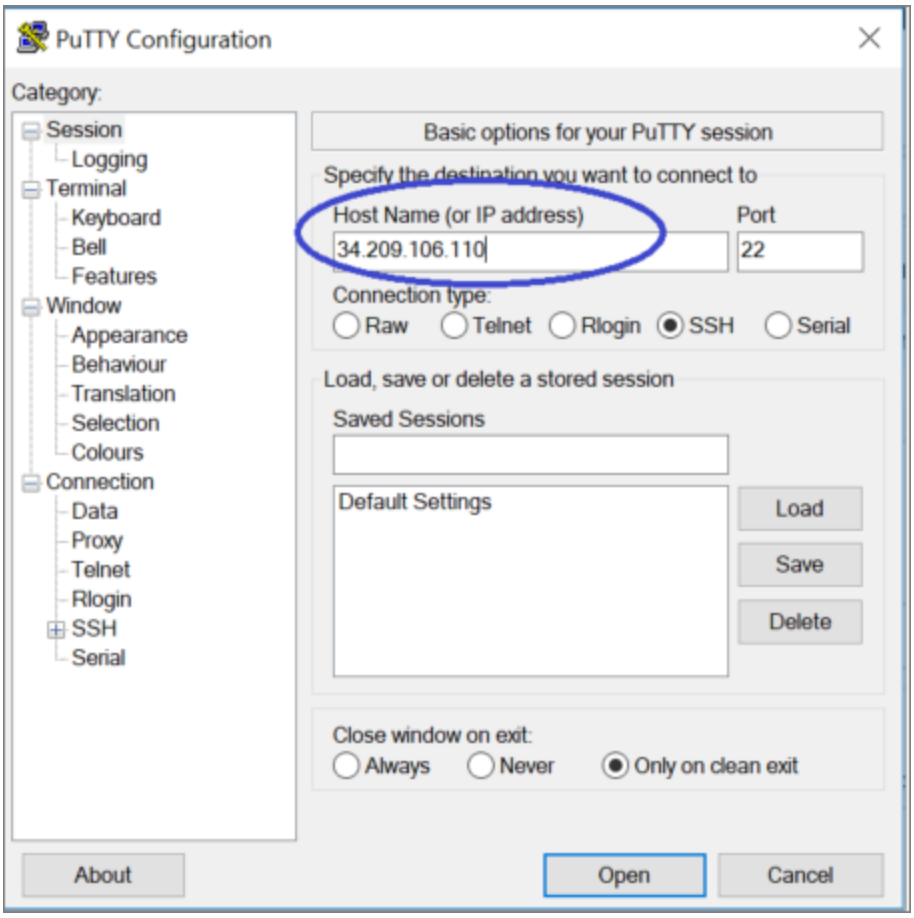


16. Then download and open putty software.

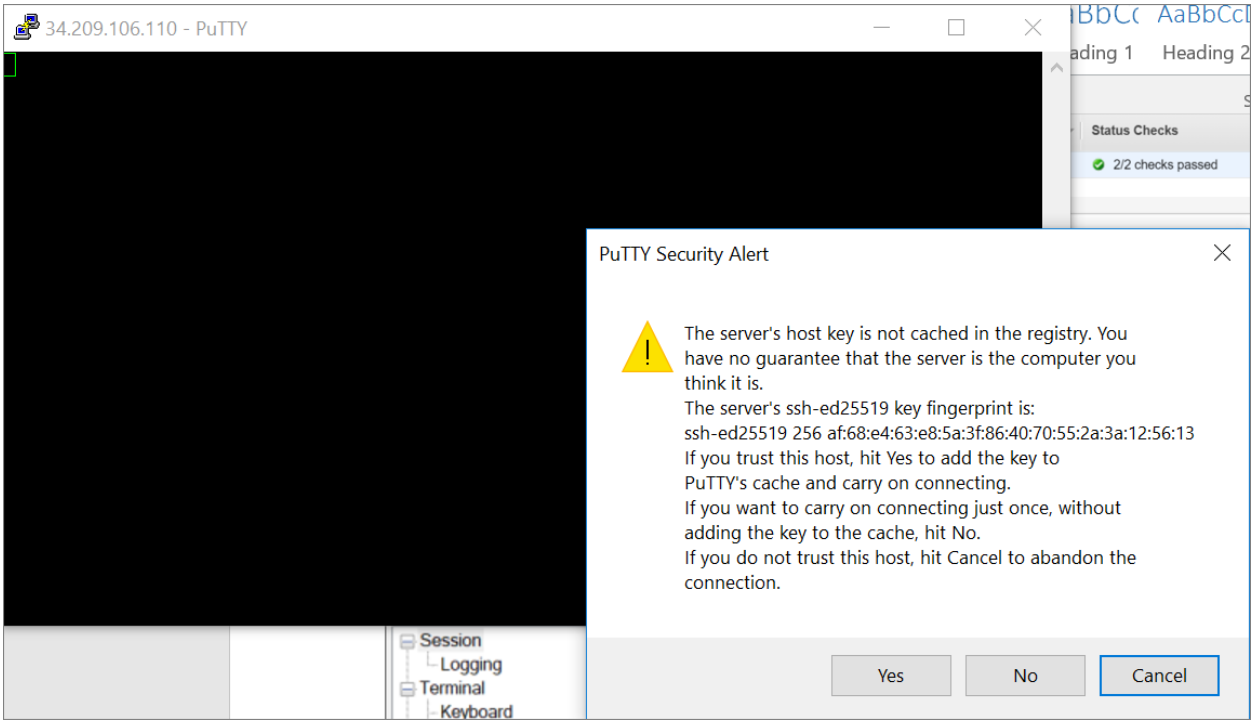
Provide the public IP of the server which is shown in the below snap.

Port will be 22 default.

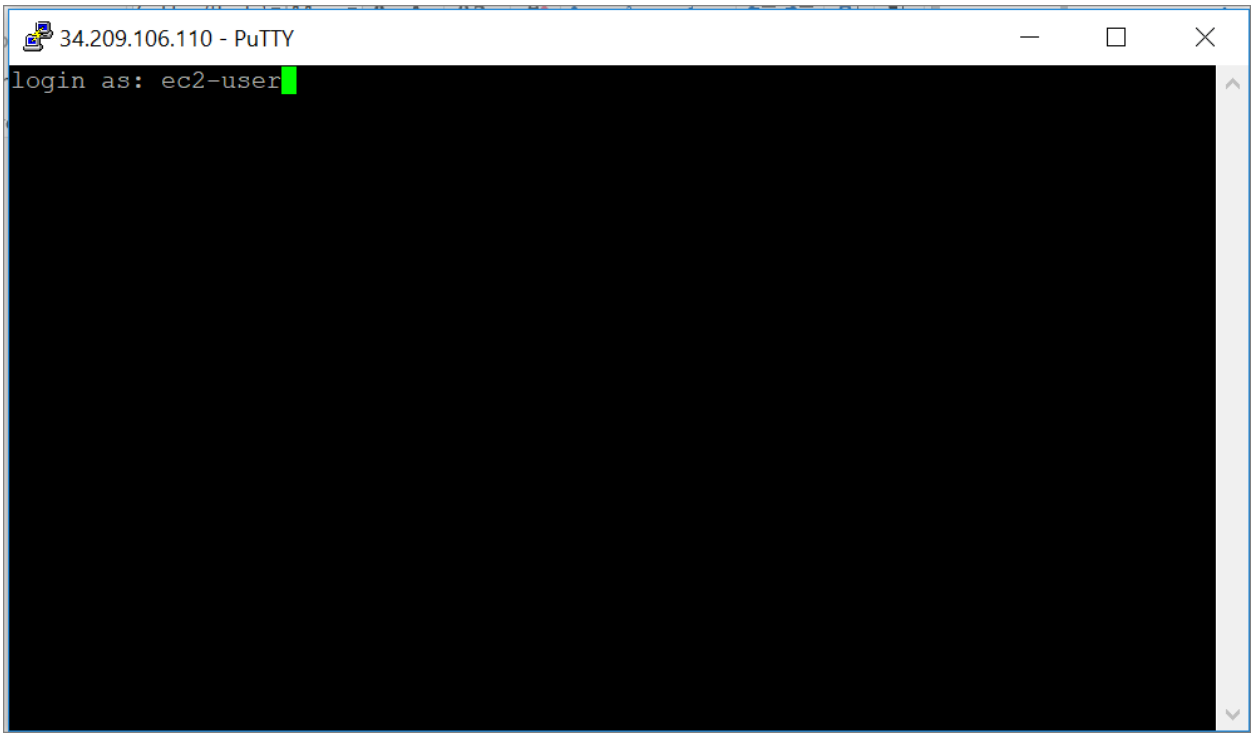
17. Expand SSH and click on Auth.
18. Click browse and browse the ppk file and click open->open



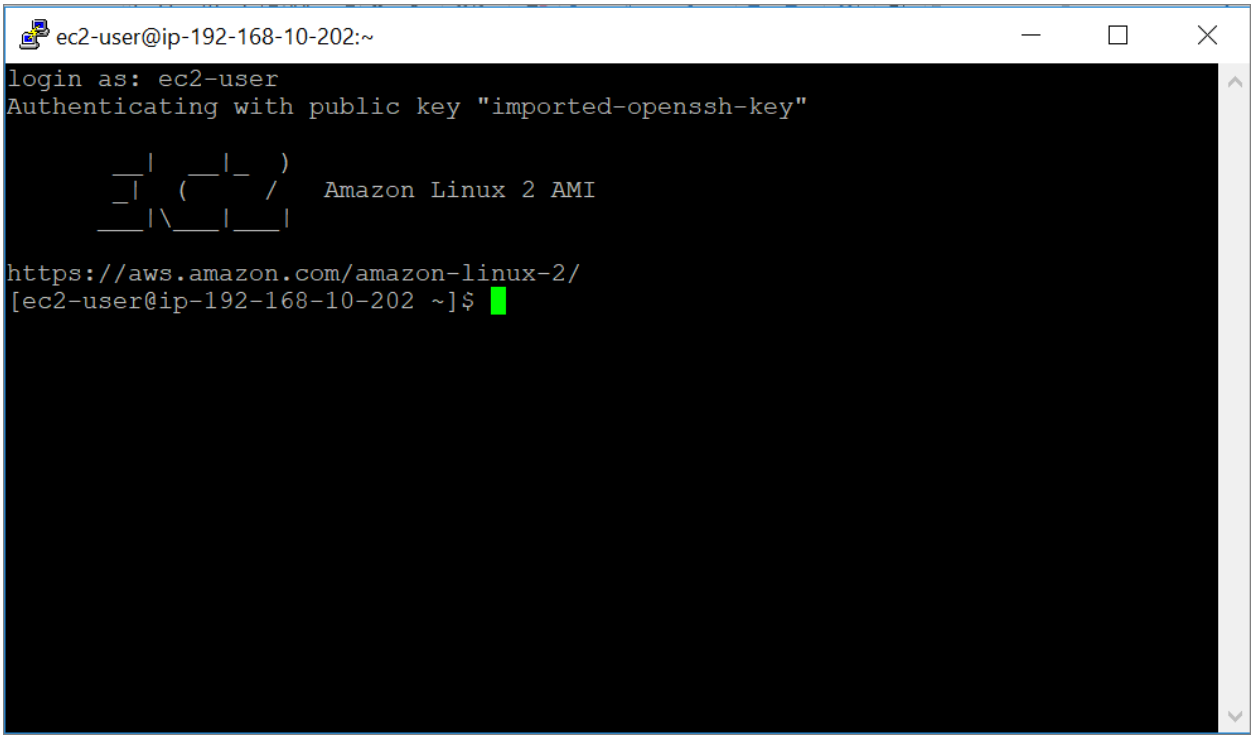
19. Click Yes on the popup window.



20. Give user name “ec2-user”



It will get connect to server and the server is ready to use. We can install the software, configure the website etc.



For creating a Windows server in AWS, follow the step from 1 to 14 by selecting the Windows AMI in step 4.

We can generate the Windows password by using pem file.

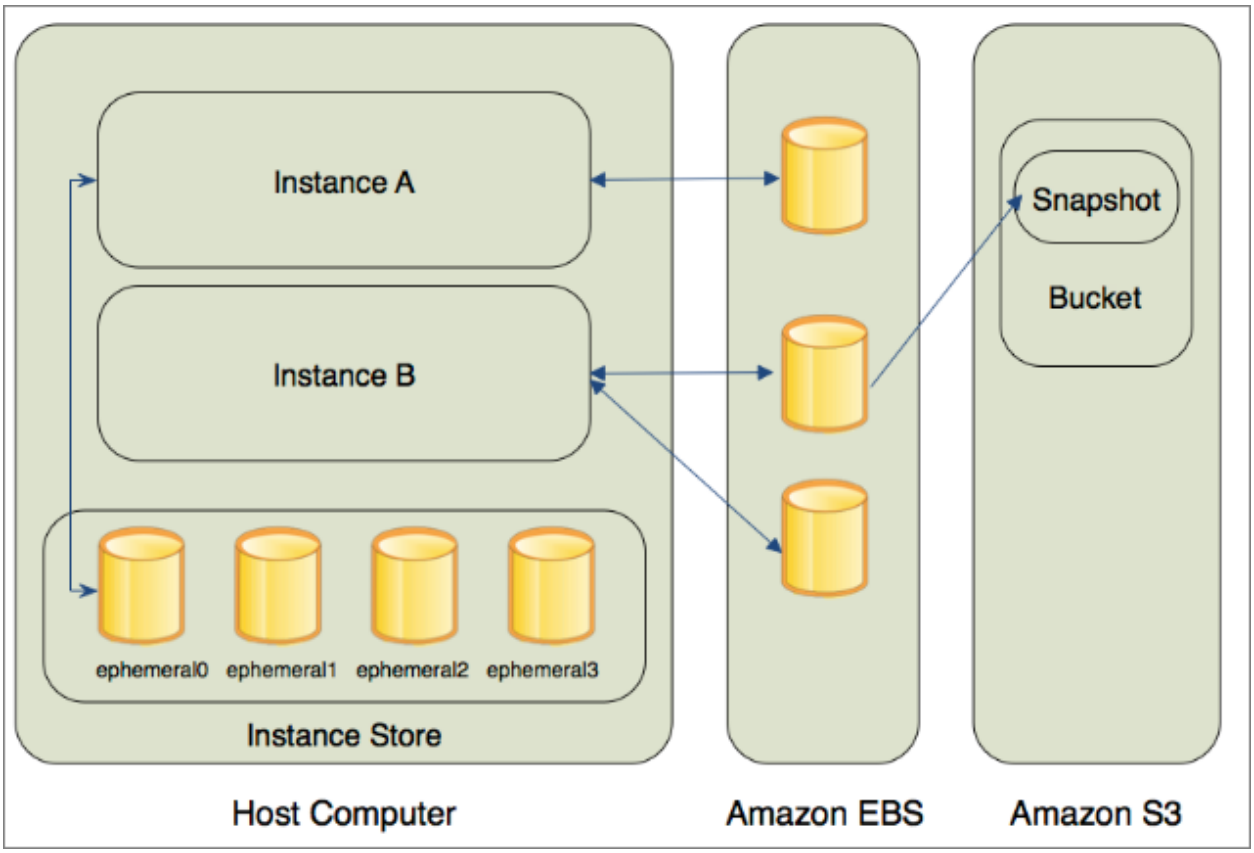
1. Choose Instances.
2. Select the instance, and from the Actions menu, choose Get Windows Password.
3. Note: It may take a few minutes for this option to be available after first launching a new instance.
4. Choose Browse, select your key pair file (. pem file), and choose Open.
   -or-
5. Paste the contents of your key pair into the provided text box.
6. Choose Decrypt Password. This will give the user name and password for windows login.

STORAGE

When using Amazon EC2, you may have data that you need to store. Amazon EC2 offers the following storage options:

- Amazon Elastic Block Store (Amazon EBS)
- Amazon EC2 Instance Store (Ephermal Storage)
- Amazon Simple Storage Service (Amazon S3)

The following figure shows the relationship between these types of storage.



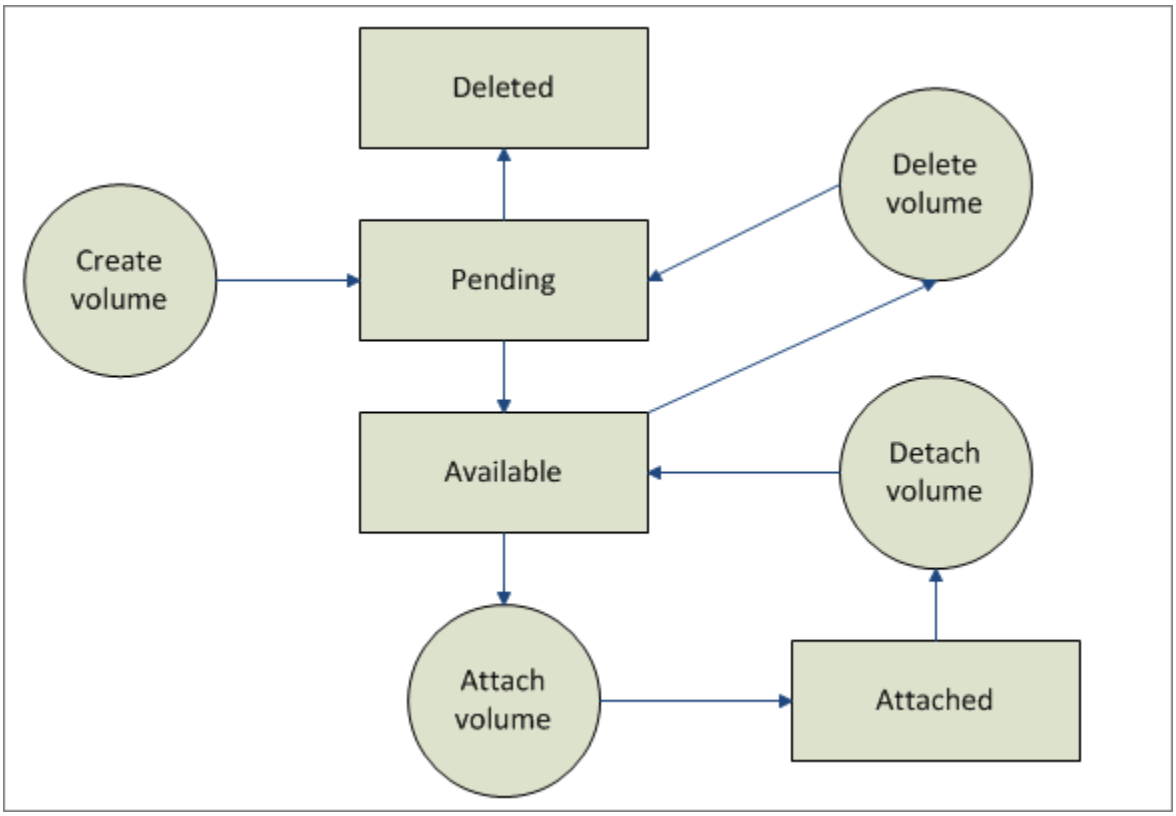
AMAZON EBS VOLUMES

Amazon EBS volumes are the recommended storage option for the majority of use cases. Amazon EBS provides your instances with persistent, block-level storage. Amazon EBS volumes are essentially hard disks that you can attach to a running instance.

Amazon EBS is especially suited for applications that require a database, a file system, or access to raw block-level storage.

you can attach multiple volumes to an instance. Also, to keep a backup copy of your data, you can create a snapshot of an EBS volume, which is stored in Amazon S3. You can create a new Amazon EBS volume from a snapshot, and attach it to another instance. You can also detach a volume from an instance and attach it to a different instance.

The following figure illustrates the life cycle of an EBS volume.



INSTANCE STORE

All instance types, with the exception of Micro instances, offer instance store, which provides your instances with temporary, block-level storage. This is storage that is physically attached to the host computer. The data on an instance store volume doesn't persist when the associated instance is stopped or terminated.

AMAZON S3

Amazon S3 is storage for the Internet. It provides a simple web service interface that enables you to store and retrieve any amount of data from anywhere on the web.

Amazon S3 is intentionally built with a minimal feature set that focuses on simplicity and robustness. Following are some of advantages of the Amazon S3 service:

- Create Buckets – Create and name a bucket that stores data. Buckets are the fundamental container in Amazon S3 for data storage.

- Store data in Buckets – Store an infinite amount of data in a bucket. Upload as many objects as you like into an Amazon S3 bucket. Each object can contain up to 5 TB of data. Each object is stored and retrieved using a unique developer-assigned key.
- Download data – Download your data or enable others to do so. Download your data any time you like or allow others to do the same.
- Permissions – Grant or deny access to others who want to upload or download data into your Amazon S3 bucket. Grant upload and download permissions to three types of users. Authentication mechanisms can help keep data secure from unauthorized access.
- Standard interfaces – Use standards-based REST and SOAP interfaces designed to work with any Internet-development toolkit.

This section describes key concepts and terminology you need to understand to use Amazon S3 effectively. They are presented in the order you will most likely encounter them.

BUCKETS

A bucket is a container for objects stored in Amazon S3. Every object is contained in a bucket. AWS S3 bucket name should be globally unique across all AWS accounts.

Buckets serve several purposes: they organize the Amazon S3 namespace at the highest level, they identify the account responsible for storage and data transfer charges, they play a role in access control, and they serve as the unit of aggregation for usage reporting. You can configure buckets so that they are created in a specific region.

OBJECTS

Objects are the fundamental entities stored in Amazon S3. Objects consist of object data and metadata. The data portion is opaque to Amazon S3. The metadata is a set of name-value pairs that describe the object. These include some default metadata, such as the date last modified, and standard HTTP metadata, such as Content-Type. You can also specify custom metadata at the time the object is stored.

S3 STORAGE CLASSES

AMAZON S3 STANDARD

Amazon S3 Standard offers high durability, availability, and performance object storage for frequently accessed data. Because it delivers low latency and high throughput, S3 Standard is perfect for a wide variety of use cases including cloud applications, dynamic websites, content distribution, mobile and gaming applications, and Big Data

analytics. S3 Lifecycle management offers configurable policies to automatically migrate objects to the most appropriate storage class.

Key Features:

- Low latency and high throughput performance
- Designed for durability of 99.999999999% of objects across multiple Availability Zones
- Data is resilient in the event of one entire Availability Zone destruction
- Designed for 99.99% availability over a given year

AMAZON S3 STANDARD-INFREQUENT ACCESS

Amazon S3 Standard-Infrequent Access (S3 Standard-IA) is an Amazon S3 storage class for data that is accessed less frequently, but requires rapid access when needed. S3 Standard-IA offers the high durability, high throughput, and low latency of S3 Standard, with a low per GB storage price and per GB retrieval fee. This combination of low cost and high performance make S3 Standard-IA ideal for long-term storage, backups, and as a data store for disaster recovery. The S3 Standard-IA storage class is set at the object level and can exist in the same bucket as S3 Standard, allowing you to use S3 Lifecycle Policies to automatically transition objects between storage classes without any application changes.

Key Features:

- Same low latency and high throughput performance of S3 Standard
- Designed for durability of 99.999999999% of objects across multiple Availability Zones
- Data is resilient in the event of one entire Availability Zone destruction
- Designed for 99.9% availability over a given year

AMAZON S3 ONE ZONE-INFREQUENT ACCESS

Amazon S3 One Zone-Infrequent Access (S3 One Zone-IA) is an Amazon S3 storage class for data that is accessed less frequently, but requires rapid access when needed. Unlike other Amazon object storage classes, which store data in a minimum of three Availability Zones (AZs), S3 One Zone-IA stores data in a single AZ. Because of this, storing data in S3 One Zone-IA costs 20% less than storing it in S3 Standard-IA. S3 One Zone-IA is ideal for customers who want a lower cost option for infrequently accessed data but do not require the availability and resilience of S3 Standard or S3 Standard-IA storage. It's a good choice, for example, for storing secondary backup copies of on-premises data or easily re-creatable data, or for storage used as an S3 Cross-Region Replication target from another AWS Region.

S3 One Zone-IA offers the same high durability†, high throughput, and low latency of Amazon S3 Standard and S3 Standard-IA, with a low per GB storage price and per GB retrieval fee. The S3 One Zone-IA storage class is set at the object level and can exist in the same bucket as S3 Standard and S3 Standard-IA, allowing you to use S3 Lifecycle Policies to automatically transition objects between storage classes without any application changes.

Key Features:

- 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

AMAZON GLACIER

Amazon Glacier is a secure, durable, and extremely low-cost storage service for data archiving. You can reliably store any amount of data at costs that are competitive with or cheaper than on-premises solutions. To keep costs low yet suitable for varying retrieval needs, Amazon Glacier provides three options for access to archives, from a few minutes to several hours. Amazon Glacier supports S3 Lifecycle Policies for automatic migration between S3 & Amazon Glacier storage classes. Please see the Amazon Glacier page for more details.

Key Features:

- Designed for durability of 99.999999999% of objects across multiple Availability Zones
- Data is resilient in the event of one entire Availability Zone destruction
- Supports SSL for data in transit and encryption of data at rest
- Extremely low cost design is ideal for long-term archive

	S3 Standard	S3 Standard-IA	S3 One Zone-IA	Amazon Glacier
Designed for Durability	99.999999999%	99.999999999%	99.999999999%†	99.999999999%
Designed for Availability	99.99%	99.9%	99.5%	N/A
Availability SLA	99.9%	99%	99%	N/A
Availability Zones	≥3	≥3	1	≥3
Minimum Capacity Charge per Object	N/A	128KB*	128KB*	N/A
Minimum Storage Duration Charge	N/A	30 days	30 days	90 days
Retrieval Fee	N/A	per GB retrieved	per GB retrieved	per GB retrieved**
First Byte Latency	milliseconds	milliseconds	milliseconds	select minutes or hours***
Storage Type	Object	Object	Object	Object
Lifecycle Transitions	Yes	Yes	Yes	Yes

S3 PROPERTIES

OBJECT 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.

In one bucket, for example, you can have two objects with the same key, but different version IDs, such as photo.gif (version 111111) and photo.gif (version 121212).

Versioning-enabled buckets enable you to recover objects from accidental deletion or overwrite

OBJECT LIFECYCLE MANAGEMENT

To manage your objects so that they are stored cost effectively throughout their lifecycle, configure their lifecycle. A lifecycle configuration is a set of rules that define actions that Amazon S3 applies to a group of objects. There are two types of actions.

Transition actions - Define when objects transition to another storage class. For example, you might choose to transition objects to the STANDARD_IA storage class 30 days after you created them, or archive objects to the GLACIER storage class one year after creating them.

Expiration actions - Define when objects expire. Amazon S3 deletes expired objects on your behalf.

AWS IAM (IDENTITY AND ACCESS MANAGEMENT)

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

When you first create an AWS account, you begin with a single sign-in identity that has complete access to all AWS services and resources in the account. This identity is called the AWS account root user and is accessed by signing in with the email address and password that you used to create the account.

We strongly recommend that you do not use the root user for your everyday tasks, even the administrative ones. Instead, adhere to the best practice of using the root user only to create your first IAM user. Then securely lock away the root user credentials and use them to perform only a few account and service management tasks.

IAM gives you the following features:

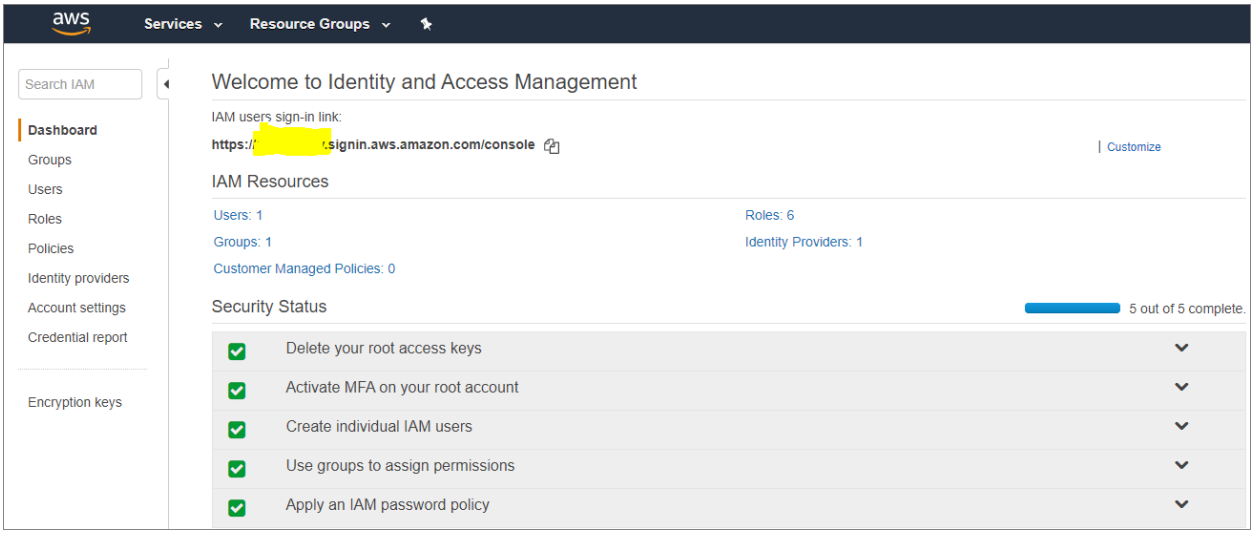
Shared access to your AWS account: You can grant other people permission to administer and use resources in your AWS account without having to share your password or access key.

Free of Cost: AWS Identity and Access Management (IAM) and AWS Security Token Service (AWS STS) are features of your AWS account offered at no additional charge. You are charged only when you access other AWS services using your IAM users or AWS STS temporary security credentials

Granular permissions: You can grant different permissions to different people for different resources. For example, you might allow some users complete access to Amazon Elastic Compute Cloud (Amazon EC2), Amazon Simple Storage Service (Amazon S3), Amazon DynamoDB, Amazon Redshift, and other AWS services. For other users, you can allow read-only access to just some S3 buckets, or permission to administer just some EC2 instances, or to access your billing information but nothing else.

Multi-factor authentication (MFA): You can add two-factor authentication to your account and to individual users for extra security. With MFA you or your users must provide not only a password or access key to work with your account, but also a code from a specially configured device.

IAM CONSOLE OVERVIEW



GROUPS

An IAM group is a collection of IAM users. Groups let you specify permissions for multiple users, which can make it easier to manage the permissions for those users. For example, you could have a group called Admins and give that group the types of permissions that administrators typically need. Any user in that group automatically has the permissions that are assigned to the group. If a new user joins your organization and needs administrator privileges, you can assign the appropriate permissions by adding the user to that group. Similarly, if a person changes jobs in your organization, instead of editing that user's permissions, you can remove him or her from the old groups and add him or her to the appropriate new groups.

Note that a group is not truly an "identity" in IAM because it cannot be identified as a Principal in a permission policy. It is simply a way to attach policies to multiple users at one time.

USERS

An IAM user is an entity that you create in AWS to represent the person or service that uses it to interact with AWS. A user in AWS consists of a name and credentials.

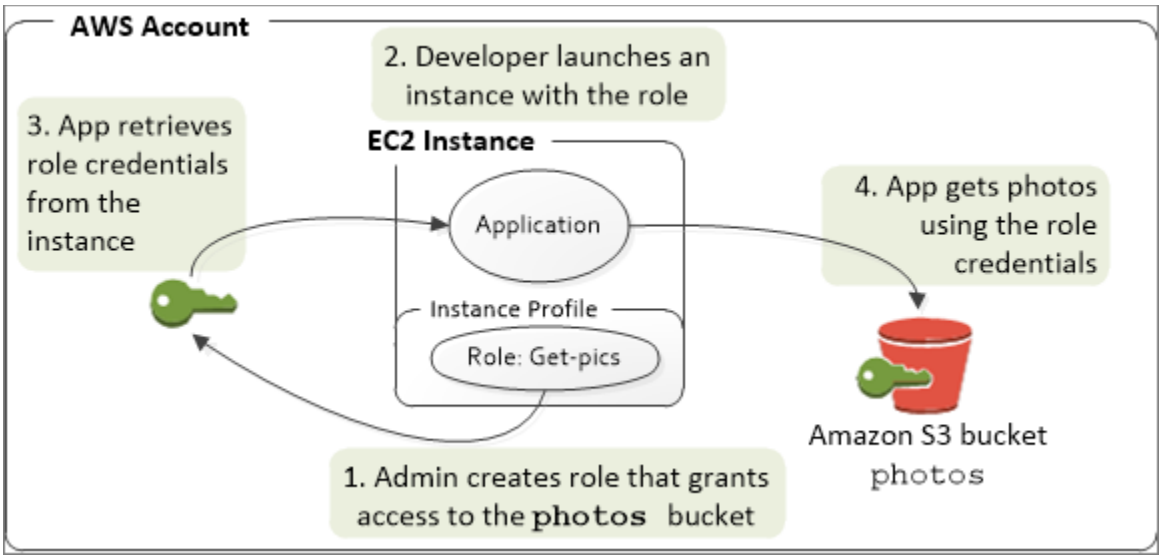
An IAM user with administrator permissions is not the same thing as the AWS account root user

ROLES

An IAM role is similar to a user, in that it is an AWS identity with permission policies that determine what the identity can and cannot do in AWS. However, instead of being uniquely associated with one person, a role is intended to be assumable by anyone who needs it. Also, a role does not have standard long-term credentials (password or access keys) associated with it. Instead, if a user assumes a role, temporary security credentials are created dynamically and provided to the user.

You can use roles to delegate access to users, applications, or services that don't normally have access to your AWS resources. For example, you might want to grant users in your AWS account access to resources they don't usually have, or grant users in one AWS account access to resources in another account. Or you might want to allow a mobile app to use AWS resources, but not want to embed AWS keys within the app (where they can be difficult to rotate and where users can potentially extract them). Sometimes you want to give AWS access to users who already have identities defined outside of AWS, such as in your corporate directory. Or, you might want to grant access to your account to third parties so that they can perform an audit on your resources.

For these scenarios, you can delegate access to AWS resources using an IAM role.



POLICIES

A policy is an object in AWS that, when associated with an entity or resource, defines their permissions. AWS evaluates these policies when a principal, such as a user, makes a request. Permissions in the policies determine whether the request is allowed or denied. Most policies are stored in AWS as JSON documents.

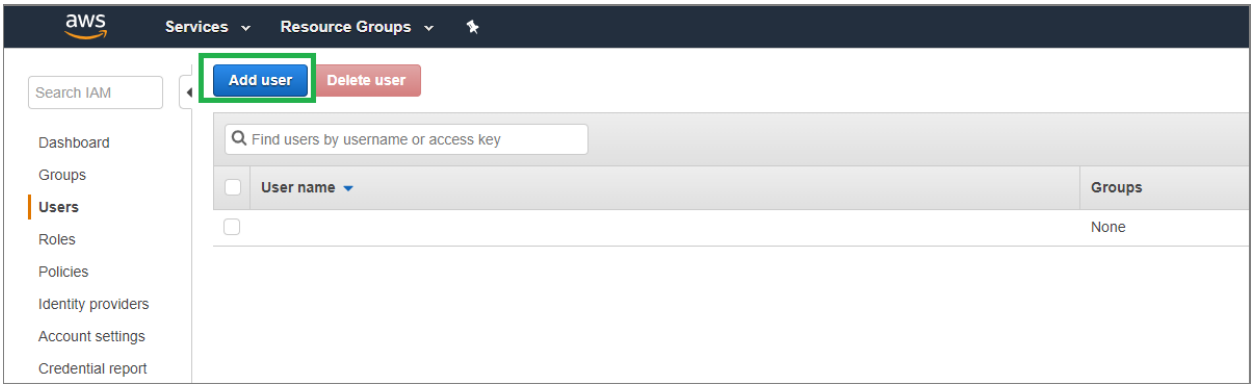
IAM policies define permissions for an action regardless of the method that you use to perform the operation. For example, if a policy allows the GetUser action, then a user with that policy can get user information from the AWS Management Console, the AWS CLI, or the AWS API. When you create an IAM user, you can set up the user to allow console or programmatic access. The IAM user can sign in to the console using a user name and password. Or they can use access keys to work with the CLI or API.

Creating New User

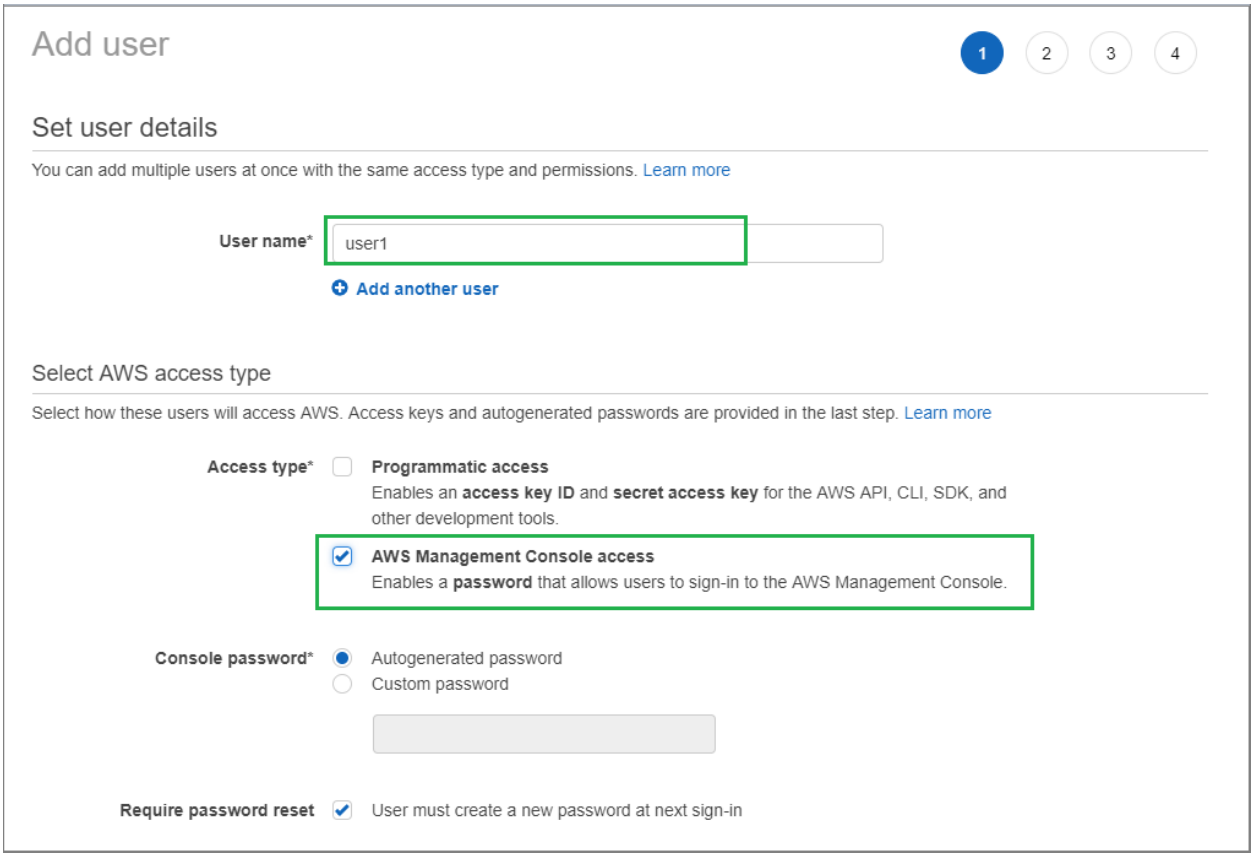
IAM is a global service. So it is not respective to any region.

To create an IAM user, follow the below step.

1. Go to AWS console->search for "IAM"
2. Click "Users"
3. Click Add user.



4. Provide user name.
5. Under access type, if you select programmatic access you will be getting an Access key and secret key to connect AWS through command line method. With this key and password, you will not be able to login to console.
6. If you choose, console access, you will be getting a password generated. With this you can login to AWS console.



Click next:Permission

There are three options available:

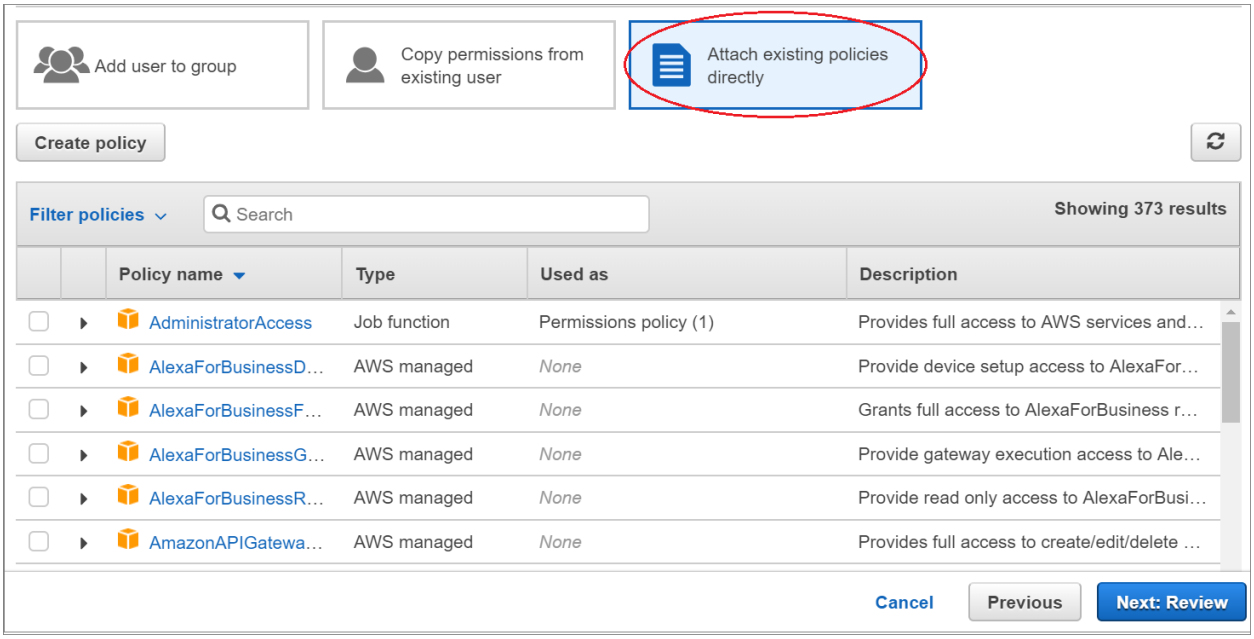
Add to Group – This option to add your user account to your existing group or you can create a group and add the user to it.

Copy permission from existing user – You can copy the permission of an existing user to the new user

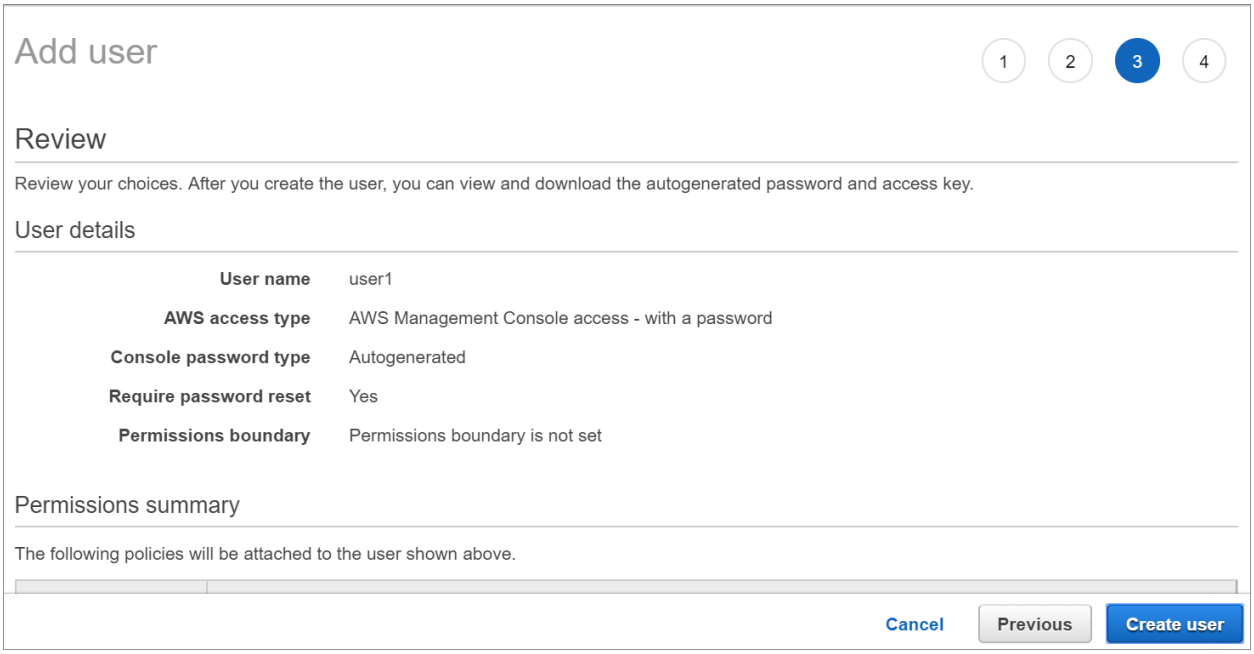
Attach existing policy directly – You can attach a policy (either AWS created policy or your custom policy) to the new user.

Here in our example, we are attaching a policy directly to the user.

In the below, select any policy which you want to assign to the user and click review.

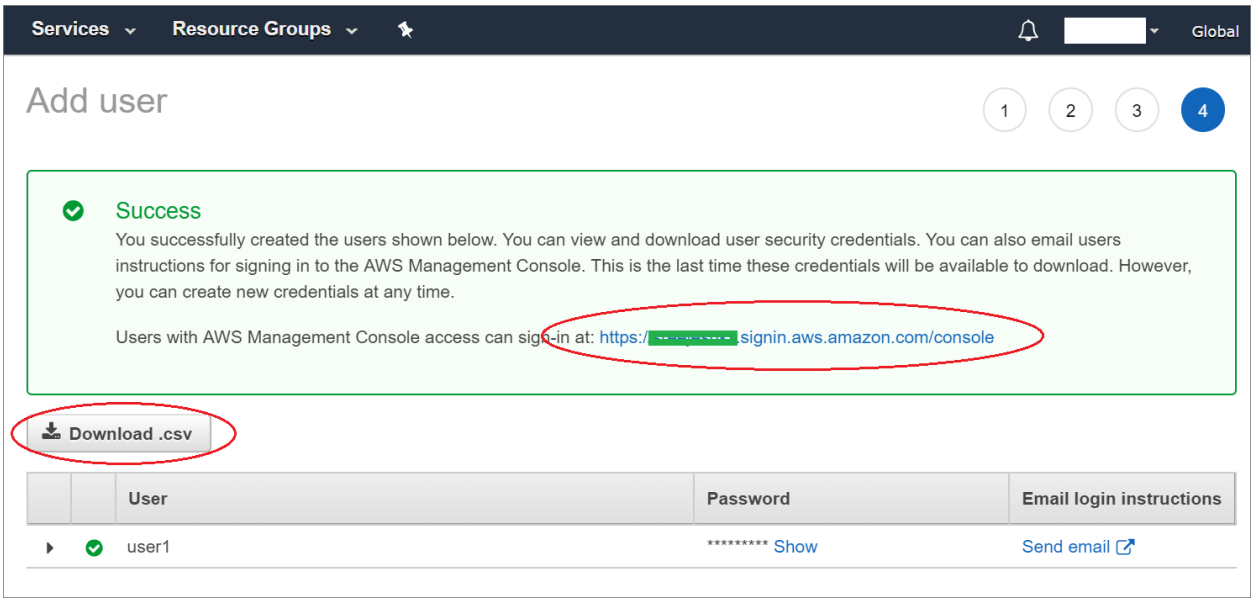


Click on Create user.

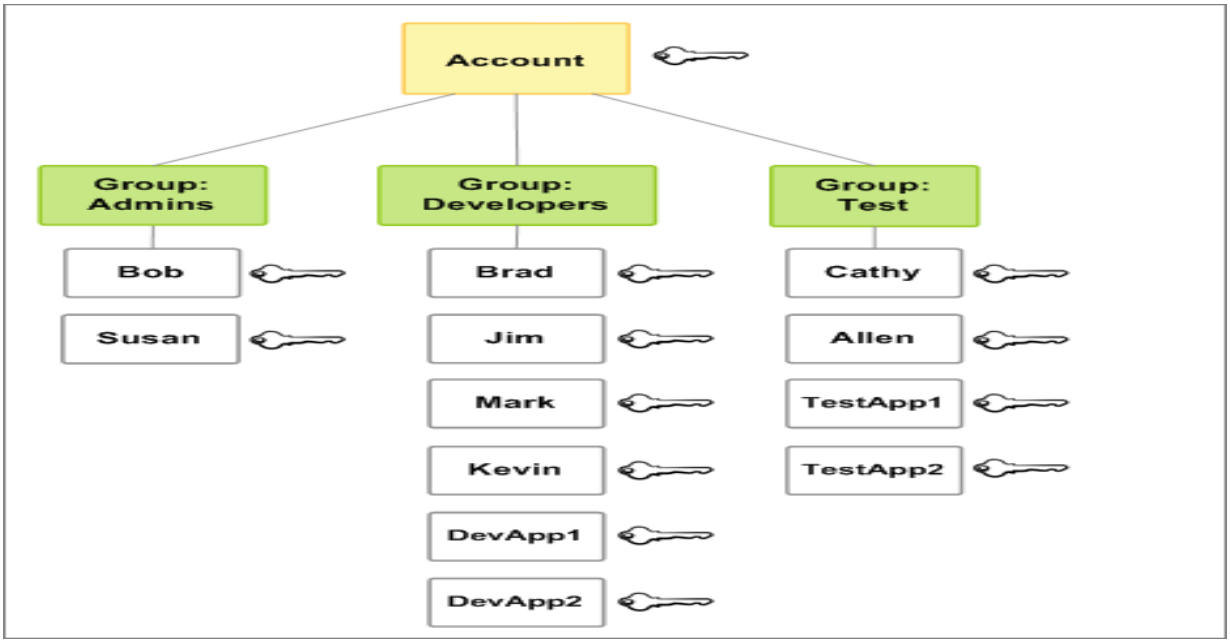


Once click on Create, you will be getting the link for downloading to the AWS account and getting an option to download the CSV file which will have user name, password and sign-in url.

With this user and password, you will be able to login to AWS console.



Below diagram shows the mapping between users and group.



IAM USER PASSWORD POLICY

You can configure the password policy for IAM user account. So while creating a user account, you need to follow certain conditions for password like the number of character in password, Upper case, lower case etc. Refer the below pic for reference.

Go to IAM->Account settings.