-: A W S :-

Amazon Web Services

AWS or Amazon Web Services is a cloud service provider that offers various computing services that are accessible over the public internet.

AWS manages and maintains hardware and infrastructure, saving organizations and individuals the cost and complexity of purchasing and running resources on site. These resources may be accessed for free or on a pay-per-use basis.

Some of the services available in AWS are:

```
EC2          ( Elastic Compute Cloud)
Lambda       ( Serverless Compute Service)
S3           ( Simple Storage Service)
EBS          ( Elastic Block Storage)
RDS          ( Relational Database Service)
IAM          ( Identity and Access Management)
Cloud Watch ( Monitoring and Observability Service )
Cloud Front ( Fast content delivery network(CDN) service)
Terraform    ( Infrastructure as Code)
```

Compute

These services help developers build, deploy, and scale an application in the cloud platform.

AWS EC2

- It is a web service that allows developers to rent virtual machines and automatically scales the compute capacity when required.
- It offers various instance types to developers so that they can choose required resources such as CPU, memory, storage, and networking capacity based on their application requirements.

AWS Lambda

- It is a serverless compute service. It is also responsible for executing code for applications.
- It helps you execute a program without the hassle of managing servers.

Storage

AWS provides web data storage service for archiving data. Also, its primary advantage is disaster data recovery with high durability.

Amazon S3

- It is an open cloud-based storage service that is utilized for online data backup.
- Amazon S3 provides storage through a web services interface and is designed for developers where web-scale computing can be easier for them.

Amazon EBS

- It provides a high availability storage volume for persistent data. It is mainly used by Amazon EC2 instances.
- EBS volumes are used explicitly for primary storage such as file storage, databases storage, and block-level storage.

Database

AWS database domain service offers cost-efficient, highly secure, and scalable database instances in the cloud.

Amazon DynamoDB

- It is a flexible NoSQL database service that offers fast and reliable performance with no scalability issues.

- It is a multi-region and durable database with instant built-in security, backup and restores features.

Amazon RDS
- It is a managed distributed relational database cloud service that helps developers to operate and scale a database in a simple manner.
- We launched it to simplify the setup, operation, and scaling process for developers while accessing a relational database.

Networking and Delivery of Content

It offers a highly secure cloud platform and connects your physical network to your private VN with a high transfer speed.

VPC – Virtual Private Cloud
- It helps a developer to deploy AWS resources, such as Amazon EC2 instances into a private virtual cloud.
- It gives you control over the complete cloud network environment, including the section of your IP address range, subnets, route table configuration, and network gateways.
- With this, developers can both IPv4 and IPv6 at a time for your resources in a highly secure environment.

Route 53
- It is a web service with a highly available Domain Name System (DNS) that helps users to route software by translating the text into an IP address.
- We launched it for developers to provide them a cost-effective method of routing end users to cloud applications.

Developer Tools

It helps a user build, deploy, and run an application source code automatically. It also updates the server and instance on the workload.

CodeStar
- It is a service designed to manage application development in a single place. Here, developers can quickly develop, build and deploy applications on AWS

Code Build
- This removes the hassle of managing physical servers and helps developers build and test code with continuous scaling.

- In simple words, it compiles your code, executes unit tests, and gives output artifacts that are ready to deploy.

Security, Identity & Compliance

It helps in monitoring a safe environment for your AWS resources by providing limited access to specific users.

IAM
- Identity Access Management is a framework that helps in maintaining access to AWS services in a secure way.
- The service gives you Shared access to your AWS account and Secure access to AWS services that run on the AWS EC2 application.

KMS – Key Management Service
- It enables users to create and manage the encryption keys that are used for encrypting data.
- The service includes a key generation method where digital sign within your applications becomes easier.

Management Tools

Using this service, an individual can optimize costs, minimize risks, and automate all the resources running efficiently on the AWS infrastructure.

Cloud Watch
- It is a monitoring tool for AWS resources and customer applications running on the AWS platform.
- The service helps you gather and access all your operational data in the form of logs from a single interface.
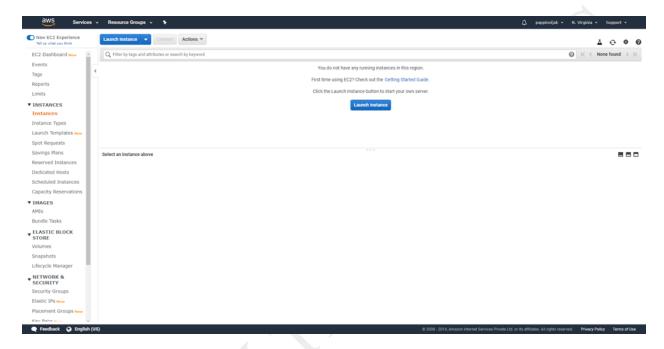
Cloud Formation
- This service helps you in monitoring all your AWS resources in one place so that you can spend minimum time managing those resources and maximum time developing applications.
- It allows developers to manage their cloud infrastructure either in a text file or a template.
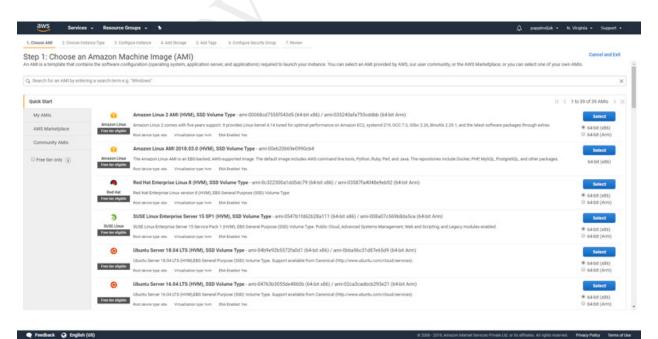
Steps to Launch a Virtual Machine(VM) using EC2

Step 1: Configure Amazon Machine Image

Let's start creating an AWS EC2 Instance. Go to AWS console and navigate to Services >> EC2.
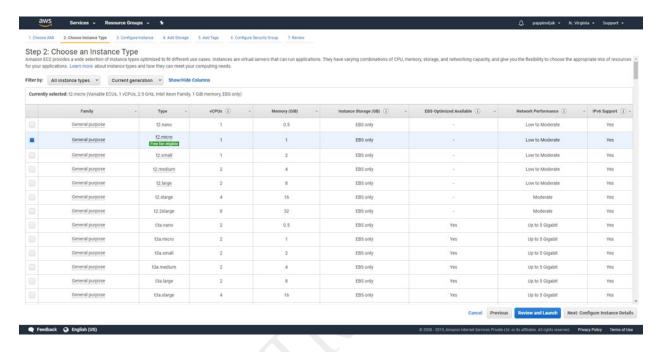
Click on Instances on the left sidebar.

Click **Launch instance**.

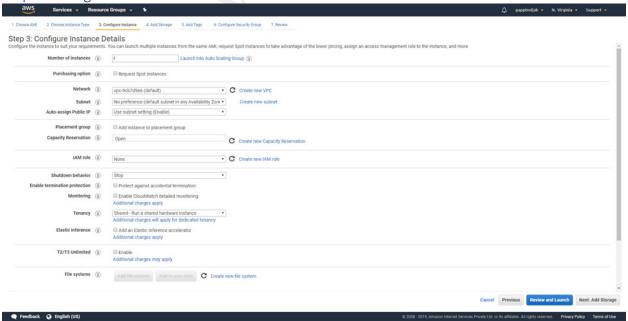Now you can select the AMI here. Choose Ubuntu server 18.04 image here.

Step 2: Choose Instance type

Here you can choose your desired machine type. If you wish to get benefited using the free tier you can choose **t2.micro** type instance.

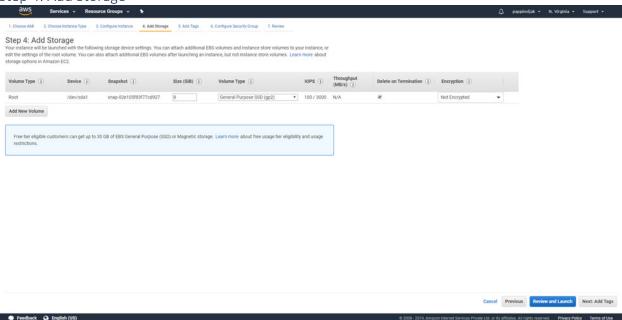Click **Configure Instance Details**.

Step 3: Configure Instance

Here you can choose how your instance should run as dedicated or as shared. But for now you can leave all settings to be as default.
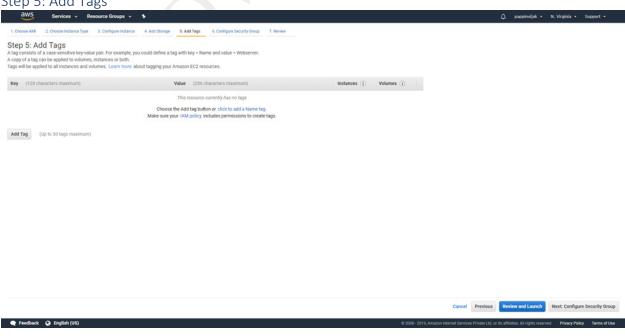
Click **Add Storage**.

Step 4: Add Storage

You can also leave this step to be with the default settings.

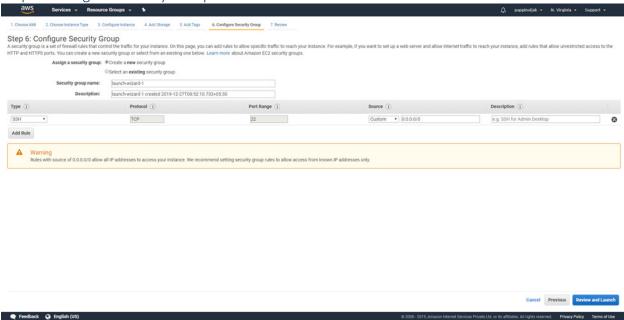Click **Add Tags**.

Step 5: Add Tags

Tags are useful to identify the Instance, here we will assign a name to our instance.

Click 'Add Tag' **to add a name tag** and enter a name in the value field.

Click **Configure Security Group**.

Step 6: Configure Security Group

This is an important setup where you can allow connections to your instance by opening certain ports.

By default SSH port is enabled. (port 22)

If you need your website to be viewed in the internet then you need add rules to allow connections to HTTP and HTTPS.

Click **Add rule**.

From the type select **HTTP**. (port 80)

Again click **Add rule**.
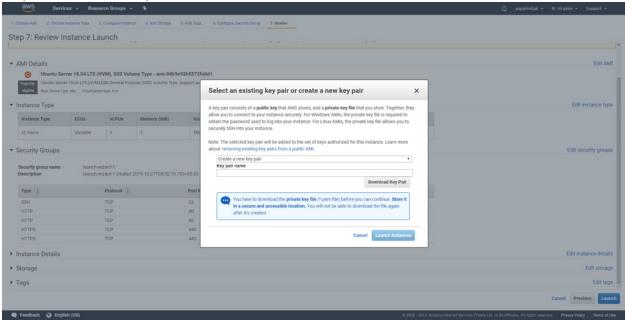
From the type select **HTTPS**. (port 443)

If you need any additional ports opened you can add rules for those ports also.

Click **Review and Launch**.

In this section you can review all the options you have chosen in the steps before. Once you have reviewed them you can click **launch**.

Step 8: Create PEM key and launch Instance

Now you will be prompted to assign a key pair to access your instance.

Select **create a new key pair.**

Enter a name to your key. <yourname_aws_key>

Click **Download Key Pair**.

Important: Save the key pair, if you miss this key you won't be able to access your instance using SSH.

Click **Launch instances**.

That's all. Now your instances will be launched with Ubuntu 18.04 and will all the configurations you made.

Once the VM comes to running state, Make a copy of Public IP assigned to the VM.
Then from your local CLI, connect to that VM using ssh command
```
ssh -i <pem_key> ubuntu@<public_ip>
```

P.S. Restart of VM will get a new ip each time. Use Elastic ip to retain same public ip for the VM.

Steps to Install Jenkins application on above VM

1) Step1: Connect to VM using SSH, PEM key and Public IP
   a. ssh -i aws_key.pem ubuntu@13.127.8.20

2) Step2: Install JAVA
   a. sudo agt update
   b. sudo apt install openjdk-8-jdk

3) Step3: Add the jenkins Repository
   a. Import GPG Keys
       wget -q -O - https://pkg.jenkins.io/debian/jenkins.io.key | sudo apt-key add –
   b. Add the Jenkins repository to the system
   c. sudo sh -c 'echo deb http://pkg.jenkins.io/debian-stable binary/ > /etc/apt/sources.list.d/jenkins.list'

4) Step4: Install jenkins
   a. sudo apt update
   b. sudo apt install jenkins
   c. Press 'Y' to confirm the download and installation
   d. To verify if jenkins is up and running
   e. sudo systemctl jenkins status

5) Step5: Modify the Firewall to allow Jenkins
   a. sudo ufw allow 8080
   b. sudo ufw status
   c. sudo ufw enable

P.S. Ensure in AWS console, for the VM, this port is opened for Inbount/Outbound traffic.

6) Setup Jenkins
   a. To launch and set up Jenkins, open a web browser, and navigate to the IP address of your server:
       http://<ip_address_or_domain>:8080
   b. Use the actual IP address or domain name for the server you're using Jenkins on.
   c. You should see a page that prompts you to Unlock Jenkins. You'll need the default password. You can get the default password by switching to a command line and entering the following:
       sudo cat /var/lib/jenkins/secrets/initialAdminPassword
   d. The system returns an alphanumeric code. Enter that code, then click Continue.

e. Next, you are prompted to either Install suggested plugins or Select plugins to install. It's fine to simply install the suggested plugins. You can always install more plugins later. The system continues the initial Jenkins setup.

f. Then you will be prompted to Create First Admin User. Enter the credentials you want to use for your Jenkins administrator, then Save and Continue.

g. After this, you should set up the Instance Configuration. This is the preferred network address for this Jenkins installation. Confirm the address you want to use for your server. This is most likely the same address you used to get to this configuration page. When you're satisfied, click Save and Finish.

h. You should see a page that says Jenkins is ready! You can click Start using Jenkins to open the Jenkins dashboard.

-: Amazon S3 Service :-

Amazon Simple Storage Service (Amazon S3) is an object storage service that offers industry-leading scalability, data availability, security, and performance.

Amazon S3 provides easy-to-use management features so you can organize your data and configure finely-tuned access controls to meet your specific business, organizational, and compliance requirements.

Amazon S3 is designed for 99.999999999% (11 9's) of durability, and stores data for millions of applications for companies all around the world.

Uses
Backup and Restore
Disaster Recovery (DR)
Archive
Data lakes and Big data analytics
Hybrid cloud storage
Cloud-native applications

Steps to use S3
1) Get AWS Login
2) Go to S3 service home page
3) Create a Bucket
4) Create folders
5) Upload some files
6) Download a file