

Practical-1

Working of Google Drive to make spreadsheets and notes.

- Google Docs is a free cloud-based suite of tools for creating documents, spreadsheets, presentations, and more. This tutorial will cover the Spreadsheets application in Google Docs, in addition to showing you how to access and store your Docs from Google Drive.
- Google Docs, Sheets, and Slides are productivity apps that let you create different kinds of online documents, work on them in real time with other people, and store them in your Google Drive online — all for free. You can access the documents, spreadsheets, and presentations you create from any computer, anywhere in the world. (There's even some work you can do without an Internet connection!) This guide will give you a quick overview of the many things that you can do with Google Docs, Sheets, and Slides.

Google Docs

- Google Docs is an online word processor that lets you create and format text documents and collaborate with other people in real time. Here's what you can do with Google Docs:
 - Upload a Word document and convert it to a Google document
 - Add flair and formatting to your documents by adjusting margins, spacing, fonts, and colors — all that fun stuff
 - Invite other people to collaborate on a document with you, giving them edit, comment or view access
 - Collaborate online in real time and chat with other collaborators — right from inside the document
 - View your document's revision history and roll back to any previous version
 - Download a Google document to your desktop as a Word, OpenOffice, RTF, PDF, HTML or zip file
 - Translate a document to a different language
 - Email your documents to other people as attachments

Google Sheets

- Google Sheets is an online spreadsheet app that lets you create and format spreadsheets and simultaneously work with other people. Here's what you can do with Google Sheets:
 - Import and convert Excel, .csv, .txt and .ods formatted data to a Google spreadsheet
 - Export Excel, .csv, .txt and .ods formatted data, as well as PDF and HTML files
 - Use formula editing to perform calculations on your data, and use formatting to make it look the way you'd like
 - Chat in real time with others who are editing your spreadsheet
 - Create charts with your data
 - Embed a spreadsheet — or individual sheets of your spreadsheet — on your blog or website

Practical-2

Install Virtualbox/VMware Workstation with different flavours of linux or windows OS

Downloading and installing VMware



Download the Ubuntu OS

Download Ubuntu Desktop

Ubuntu 20.04.1 LTS

Download the latest LTS version of Ubuntu for desktop PCs and laptops. LTS stands for long-term support — which means five years, until April 2025, of free security and maintenance updates, guaranteed.

[Ubuntu 20.04 LTS release notes](#)

Recommended system requirements:

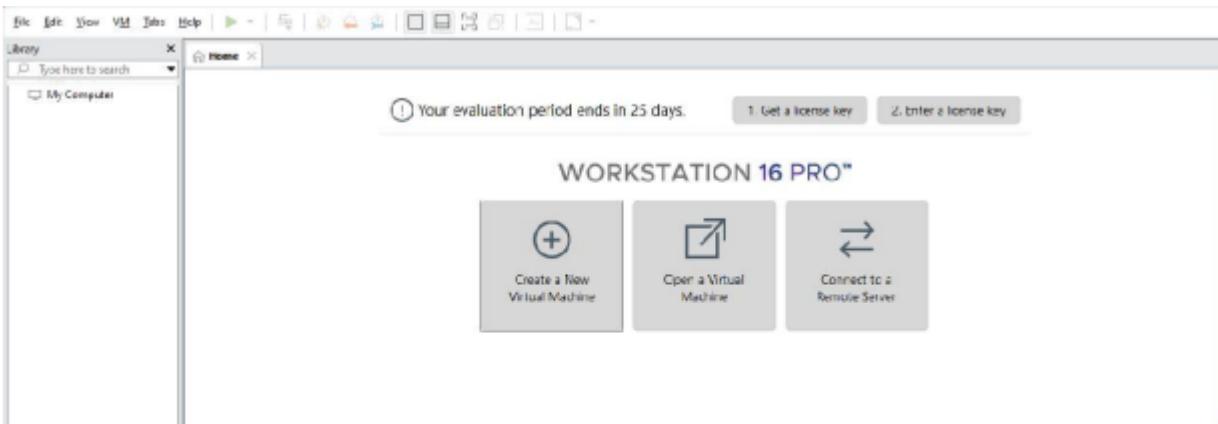
- 2 GHz dual-core processor or better
- 4 GB system memory
- 25 GB of free hard drive space

- Internet access is helpful
- Either a DVD drive or a USB port for the installer media

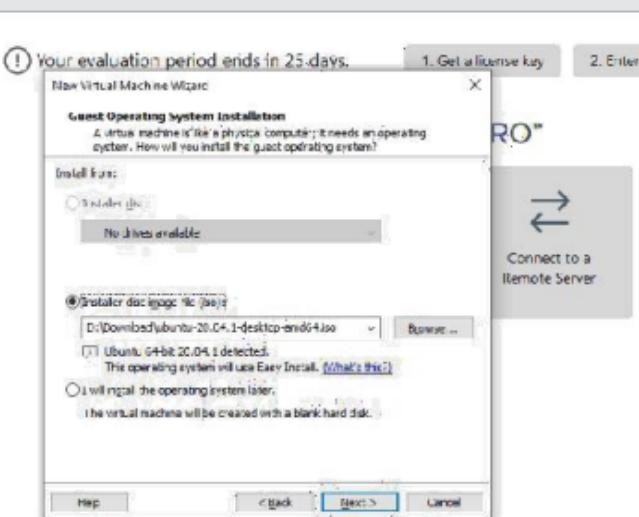
[Download](#)

For other versions of Ubuntu Desktop including torrents, the network installer, a list of local mirrors, and beta releases see our [alternative downloads](#).

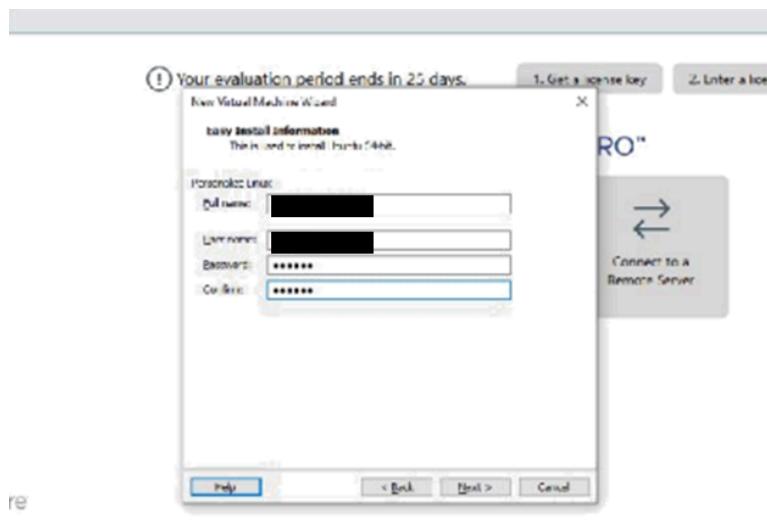
Create new virtual machine



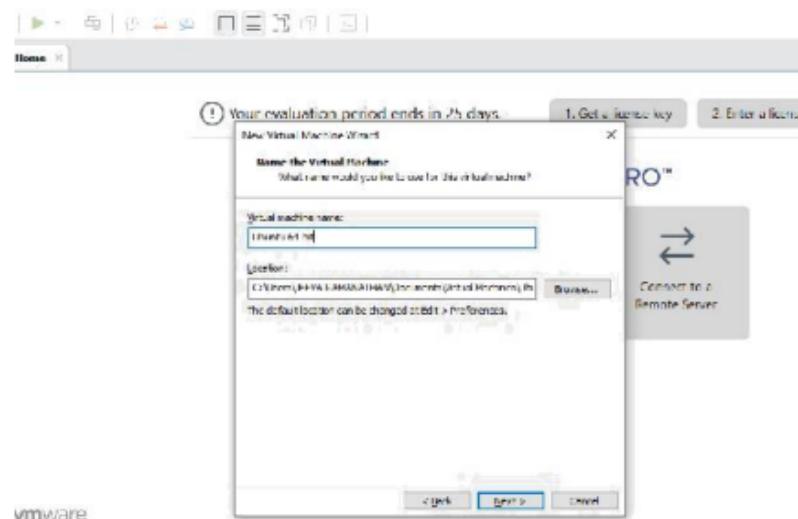
Browse the downloaded Ubuntu file and click next



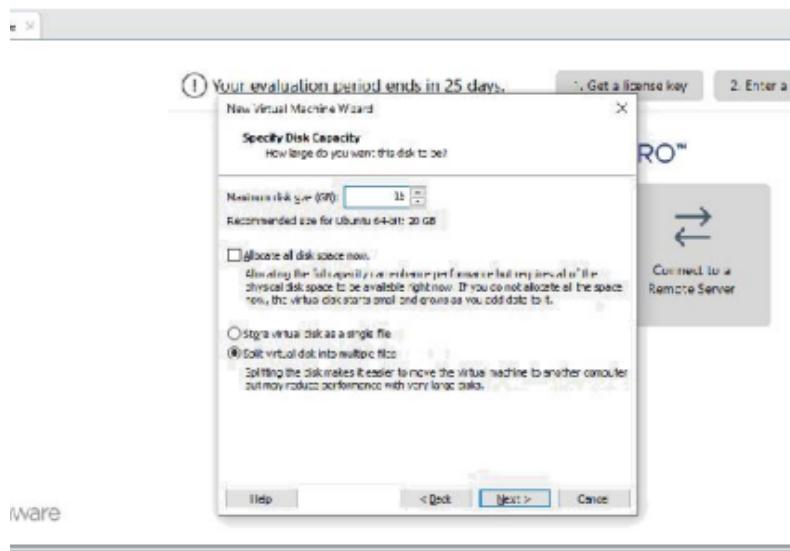
Create an username and password and click next



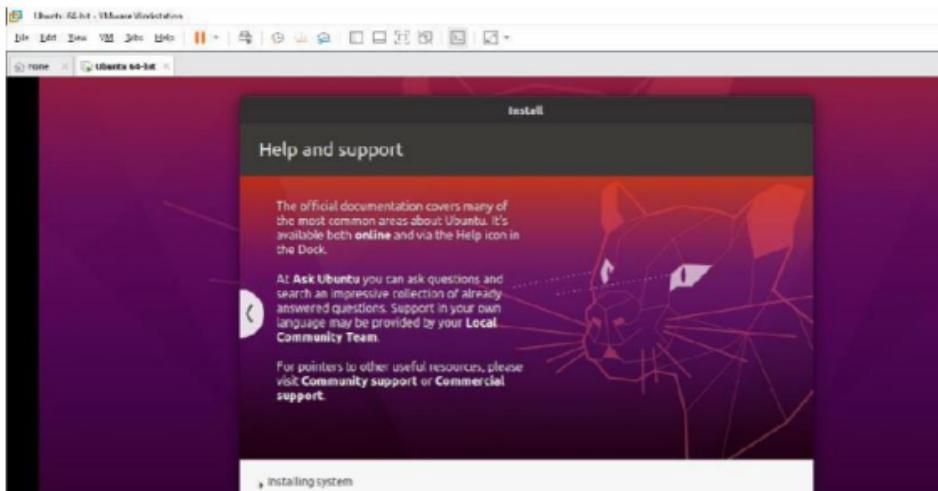
Choose the location to use your virtual machine and click next



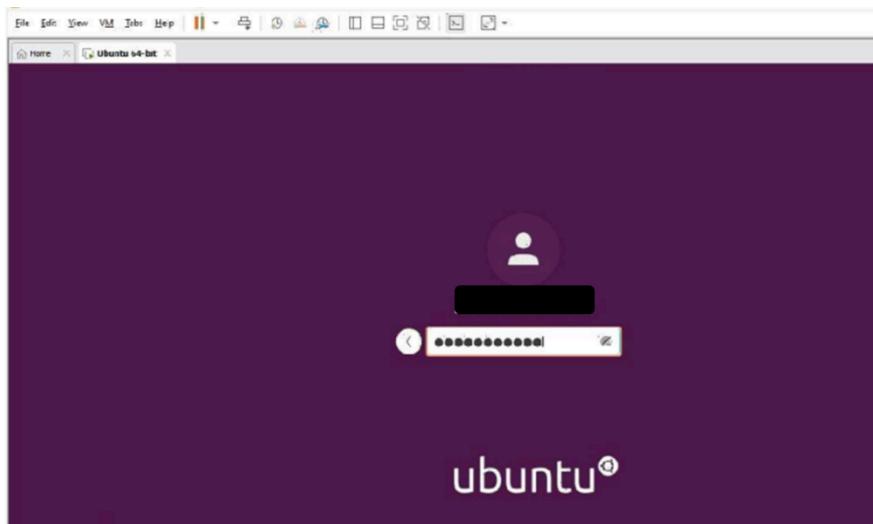
Specify the disk size and click next



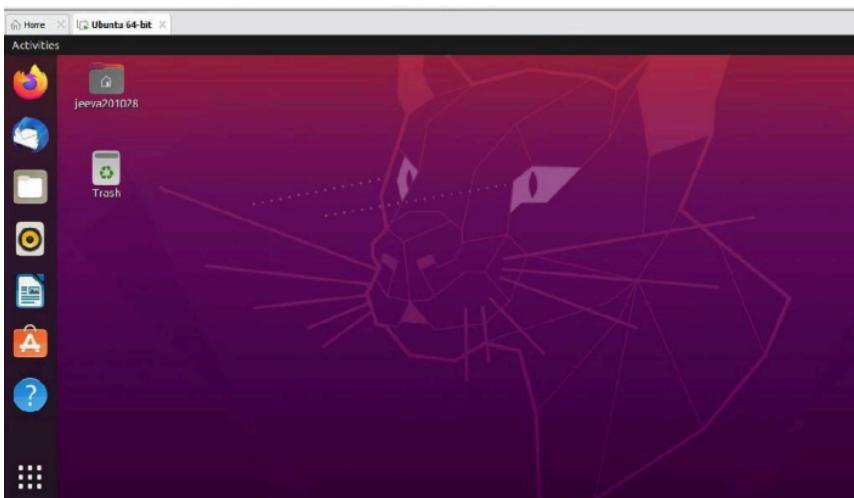
Installing Ubuntu on VMware and unzipping files



Login to Ubuntu



Thus, we have installed VMware Workstation with different flavours of linux on top of windows



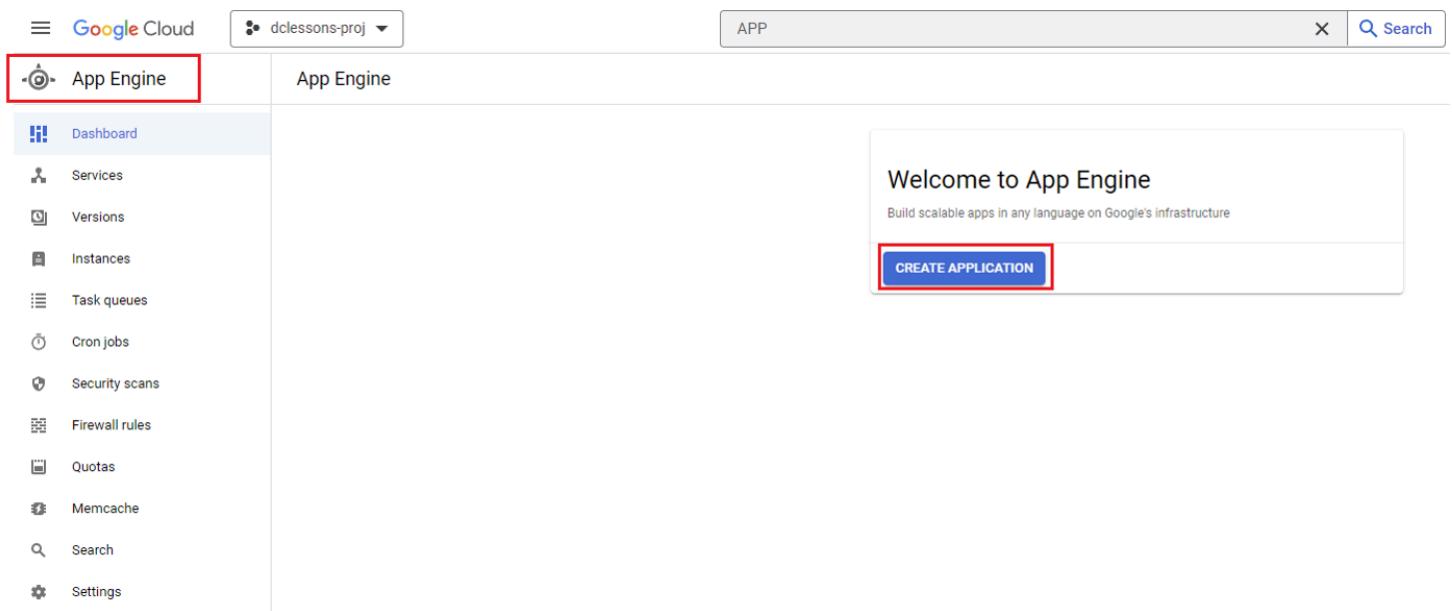
Practical-3

Installation and configuration of Google App Engine

Google App Engine (often referred to as GAE or simply App Engine) is a web framework and cloud computing platform for developing and hosting web applications in Google-managed data centers. Applications are sandboxed and run across multiple servers. App Engine offers automatic scaling for web applications—as the number of requests increases for an application, App Engine automatically allocates more resources for the web application to handle the additional demand.

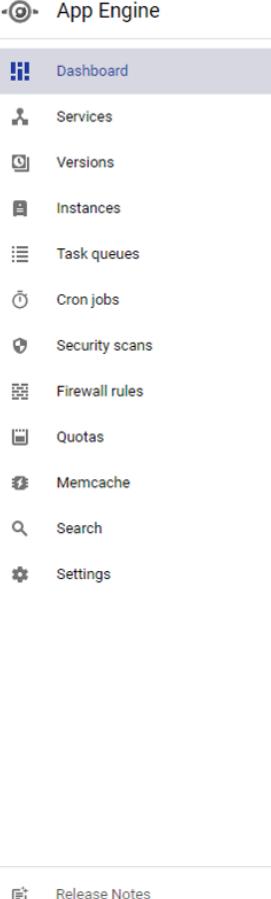
The Google App Engine is serverless so you don't have to think about any of the hardware behind your web application. You don't have to know what OS your application is running under, it will scale accordingly, Google will take care of security patches, and you only pay for what you consume. If you are looking for a hands-off web hosting option, this may be it. I will write a post on Amazon's AWS Lambda shortly to compare services. There are two types of App Engines you can opt for - one is very simple but less customizable, while the other isn't. We'll stick to the simple one - Standard Environment.

Go to Google Cloud Console | Select App Engine | Select Create APPLICATION



The screenshot shows the Google Cloud Console interface for the 'dclessons-proj' project. The top navigation bar includes 'Google Cloud', a dropdown for the project ('dclessons-proj'), a search bar, and a 'Search' button. Below the navigation is a header with 'APP' and a close button ('X'). On the left, there's a sidebar menu for 'App Engine' with options like Dashboard, Services, Versions, Instances, Task queues, Cron jobs, Security scans, Firewall rules, Quotas, Memcache, Search, and Settings. The main content area displays a 'Welcome to App Engine' message: 'Build scalable apps in any language on Google's infrastructure' with a prominent blue 'CREATE APPLICATION' button, which is highlighted with a red box.

Select Region | us-central | Next



- Dashboard**
- Services
- Versions
- Instances
- Task queues
- Cron jobs
- Security scans
- Firewall rules
- Quotas
- Memcache
- Search
- Settings

[Release Notes](#)

Create app

1 Configure application — 2 Get started

Region

Select a region for your App Engine application. Please remember, once selected the region is permanently tied to the project.



Select a region * **us-central**

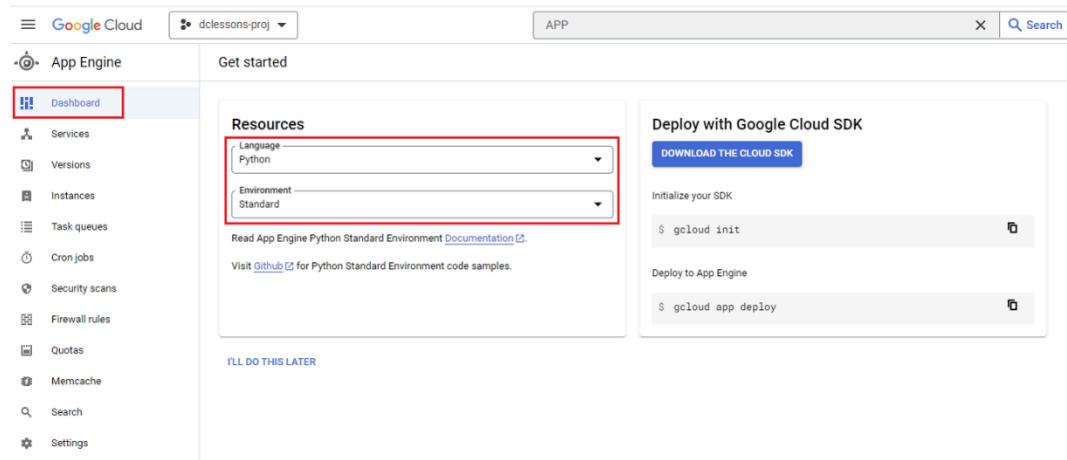
Identity and API access

Select a service account

If no service account is selected the default App Engine service account will be used.

NEXT

In Dashboard | Select Language: Python | Environment: Standard



The screenshot shows the Google Cloud Platform App Engine dashboard for a project named 'dclessons-proj'. The 'Dashboard' tab is selected. In the 'Resources' section, 'Language' is set to 'Python' and 'Environment' is set to 'Standard'. The 'Deploy with Google Cloud SDK' section contains commands for initializing the SDK and deploying to App Engine.

git clone <https://github.com/GoogleCloudPlatform/python-docs-samples>

```
dclessons1@cloudshell:~ (dclessons-proj)$ git clone https://github.com/GoogleCloudPlatform/python-docs-samples
Cloning into 'python-docs-samples'...
remote: Enumerating objects: 93083, done.
remote: Counting objects: 100% (244/244), done.
remote: Compressing objects: 100% (179/179), done.
remote: Total 93083 (delta 102), reused 177 (delta 61), pack-reused 92839
Receiving objects: 100% (93083/93083), 196.39 MiB | 17.78 MiB/s, done.
Resolving deltas: 100% (56083/56083), done.
dclessons1@cloudshell:~ (dclessons-proj)$
dclessons1@cloudshell:~ (dclessons-proj)$
dclessons1@cloudshell:~ (dclessons-proj)$
```

Now Change the directory to browse to application

cd python-docs-samples/appengine/standard_python3/hello_World

```
dclessons1@cloudshell:~ (dclessons-proj)$ cd python-docs-samples/appengine/standard_python3/hello_world
dclessons1@cloudshell:~/python-docs-samples/appengine/standard_python3/hello_world (dclessons-proj)$
dclessons1@cloudshell:~/python-docs-samples/appengine/standard_python3/hello_world (dclessons-proj)$
dclessons1@cloudshell:~/python-docs-samples/appengine/standard_python3/hello_world (dclessons-proj)$
dclessons1@cloudshell:~/python-docs-samples/appengine/standard_python3/hello_world (dclessons-proj)$
dclessons1@cloudshell:~/python-docs-samples/appengine/standard_python3/hello_world (dclessons-proj)$
```

Practical-4

Working and installation of Microsoft Azure

Microsoft Azure is a cloud computing platform and infrastructure, created by Microsoft, for building, deploying and managing applications and services through a global network of Microsoft-managed data centers.

It provides software as a service (SaaS), platform as a service (PaaS) and infrastructure as a service (IaaS) and supports many different programming languages, tools and frameworks, including both Microsoft-specific and third-party software and systems.

If you're looking to take advantage of the cloud-based services offered by Microsoft Azure, you've come to the right place.

Step 1: Register for a Microsoft Azure Account

The first step in installing Microsoft Azure is to register for an account. This is a simple process and requires a valid email address and credit card information. Once registered, you will be able to access the Microsoft Azure Portal, which is the main interface for managing your Azure resources.

Step 2: Configure Your Azure Subscription

Once you have registered for an account, you will need to configure your Azure subscription. This includes setting up billing information, selecting the type of subscription you want to use, and setting up any other features you may need.

Step 3: Create an Azure Resource Group

The next step in installing Microsoft Azure is to create an Azure resource group. This is a logical grouping of resources in your Azure subscription that you can use to manage and deploy Azure services. You can create multiple resource groups and assign different services to them.

Step 4: Create an Azure Storage Account

An Azure storage account is a secure place to store your data and files. You will need to create an Azure storage account if you plan to use Azure services. This can be done easily through the Azure Portal.

Step 5: Deploy Your Resources

Once you have set up your subscription, resource group, and storage account, you can start deploying resources. This includes deploying virtual machines, web apps, databases, and more. You can easily do this through the Azure Portal.

Step 6: Monitor and Manage Your Resources

The final step in installing Microsoft Azure is to monitor and manage your resources. This includes setting up alerts, scaling up and down resources, and keeping track of resource usage and costs. You can use the Azure Portal to monitor and manage your resources.

Practical-5

Installation of docker on different operating systems

- Docker is a popular containerization platform that allows you to package and deploy applications in a consistent and reproducible way, regardless of the underlying operating system.
- Docker can be installed on a variety of operating systems, including Windows, macOS, and various Linux distributions. In this answer, we'll explore the steps to install Docker on different platforms.

Docker Desktop

- Docker Desktop is a graphical user interface for Docker.
- Below is a walk-through of its features, illustrated on the Mac. Docker Desktop is installed by default on Mac and Windows.
- Although Docker Desktop is available for some Unix systems, it is not part of the default installation.

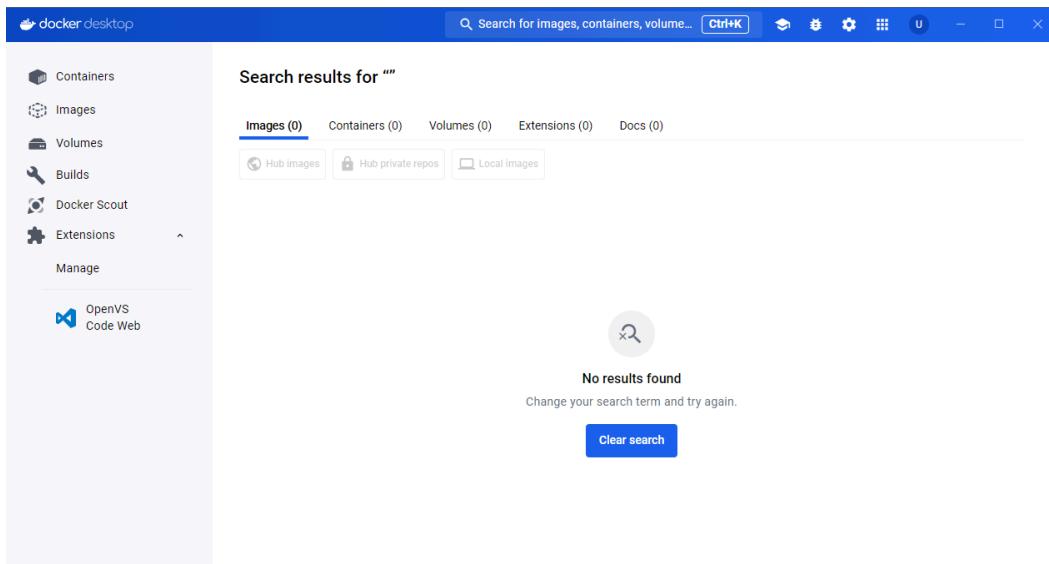
Installing Docker on Windows: Steps to Install Docker Desktop on Windows:

- Docker Desktop for Windows requires either WSL2 or Hyper-V
- Windows Subsystem for Linux (WSL) 2
- WSL allows you to run full Linux environments directly on Windows 10 or Windows 11.
- Download Docker Desktop: Visit the Docker Hub website and download the Docker Desktop installer for Windows.
- Run Installer: Double-click the downloaded installer file and follow the installation wizard's instructions. Ensure that Hyper-V is enabled in the BIOS settings if prompted.
- Launch Docker Desktop: Once installed, Docker Desktop will appear in your system tray. Click on it to start Docker.
- Verification: Open a command prompt or PowerShell window and enter docker version to verify that Docker is installed correctly.

Start Docker Desktop Tool

- After the installation process is complete, the tool does not start automatically. To start the Docker tool, search for the tool, and select Docker Desktop in your desktop search results.
- Before starting the application, Docker offers an onboarding tutorial. The tutorial explains how to build a Docker image and run a container.

- You are now successfully running Docker Desktop on Windows.



```
PS C:\Users\HareKrishna> docker --version
Docker version 26.1.4, build 5650f9b
PS C:\Users\HareKrishna> |
```

Practical-6

Debugging docker containers with docker exec and docker logs

```
[DBG]: PS C:\Users\HareKrishna>> docker images
REPOSITORY          TAG      IMAGE ID      CREATED       SIZE
myubuntudemoimage  latest   3aa37af7eb7a  2 weeks ago  78.1MB
demo1              latest   c2892d8c56b8  2 weeks ago  1.02GB
demo               latest   2060dc602362  2 weeks ago  1.02GB
```

docker pull

- Usage: docker pull <image name>: This command is used to get the currently installed version of docker
- This command is used to pull images from the docker repository(hub.docker.com)
- Example:

```
PS C:\Users\HareKrishna> docker pull python
Using default tag: latest
latest: Pulling from library/python
903681d87777: Downloading 9.628MB/49.55MB
3cbbe86a28c2: Downloading 7.146MB/24.05MB
6ed93aa58a52: Downloading 4.303MB/64.14MB
787c78da4383: Waiting
239e5f5bf9b4: Waiting
ed3e311c15d3: Waiting
f445a5851dab: Waiting
815b2c9ae66d: Waiting
```

docker run

- Usage: docker run -it -d <image name>
- This command is used to create a container from an image

docker ps : This command is used to list the running containers

```
PS C:\Users\HareKrishna> docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              PORTS               NAMES
458023eff69f        d1d39f5c5b14    "python3"          About a minute ago   Up About a minute   pythonCont
```

docker exec

- Usage: docker exec -it <container id> bash
- This command is used to access the running container

docker commit

- Usage: docker commit <conatainer id> <username/imagename>

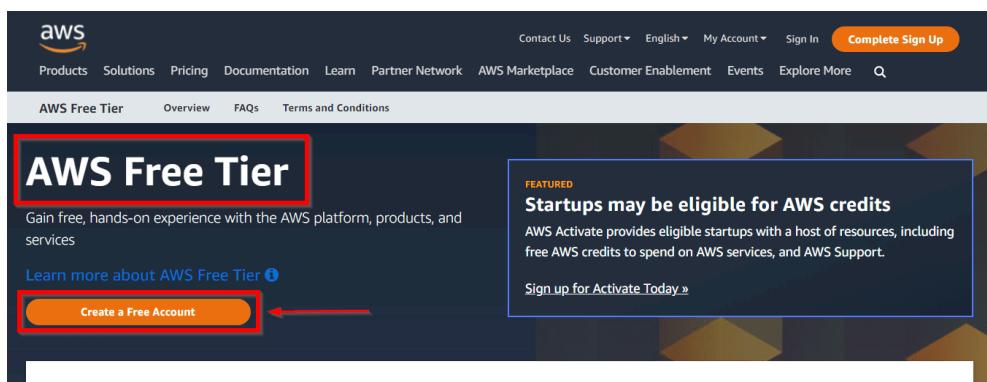
- This command creates a new image of an edited container on the local system

Practical-7

Basics about AWS Login and Step by step about How to create AWS Free Tier Account

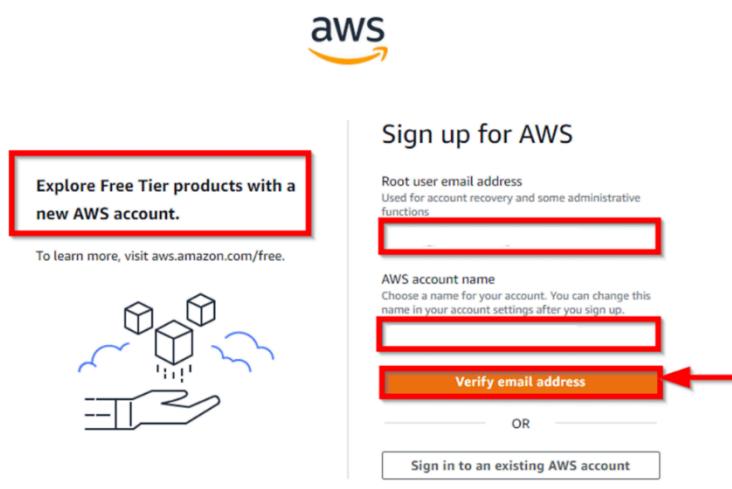
Step 1: First Open your web browser and navigate to AWS Free Tier Page

Step 2: On the middle click of Create a Free Account



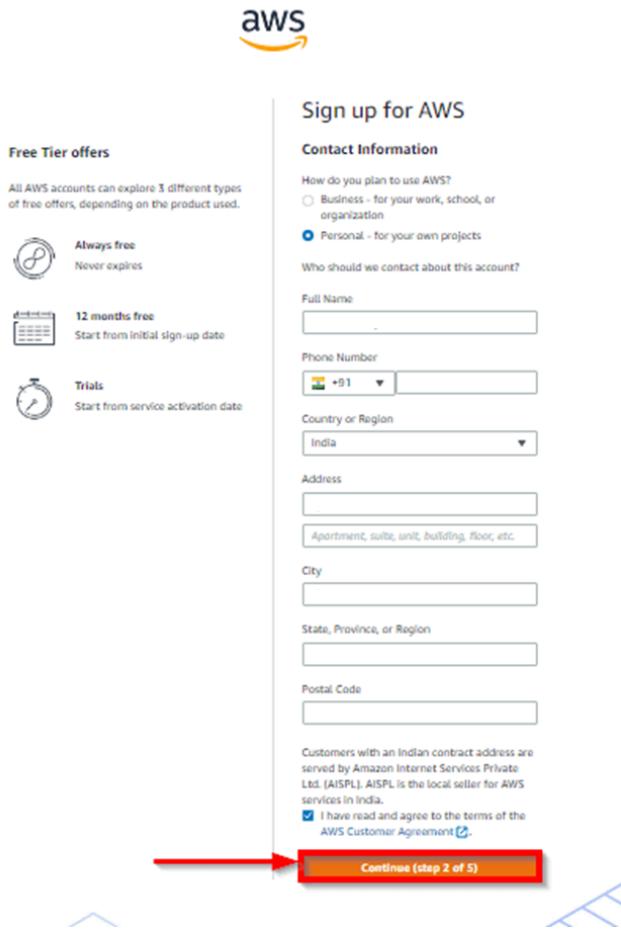
Step 3: Issue the details that you want to use to log in to your AWS account and click on Continue

- Email address: Provide the email ID that hasn't been registered yet with Amazon AWS.
- Password: Type your password.
- Confirm password: Authenticate the password.
- AWS Account name: Choose a name for your account. You can change this name in your account settings after you sign up.



Step 4: Contact Information

Select your AWS type (Professional/ Personal) Fill in the correct information to validate your account if you're going to create personal use then click on "Personal Account" else use "Company Account", Accept the Terms and condition and then click on Create Account and Continue



The screenshot shows the 'Sign up for AWS' step 2 of 5 form. It includes sections for 'Free Tier offers' (Always free, 12 months free, Trials), 'Contact Information' (Full Name, Phone Number, Country or Region set to India, Address, City, State, Province, or Region, Postal Code), and a note about Indian contract addresses. A checkbox for accepting the AWS Customer Agreement is checked. A red arrow points to the 'Continue (step 2 of 5)' button at the bottom right.

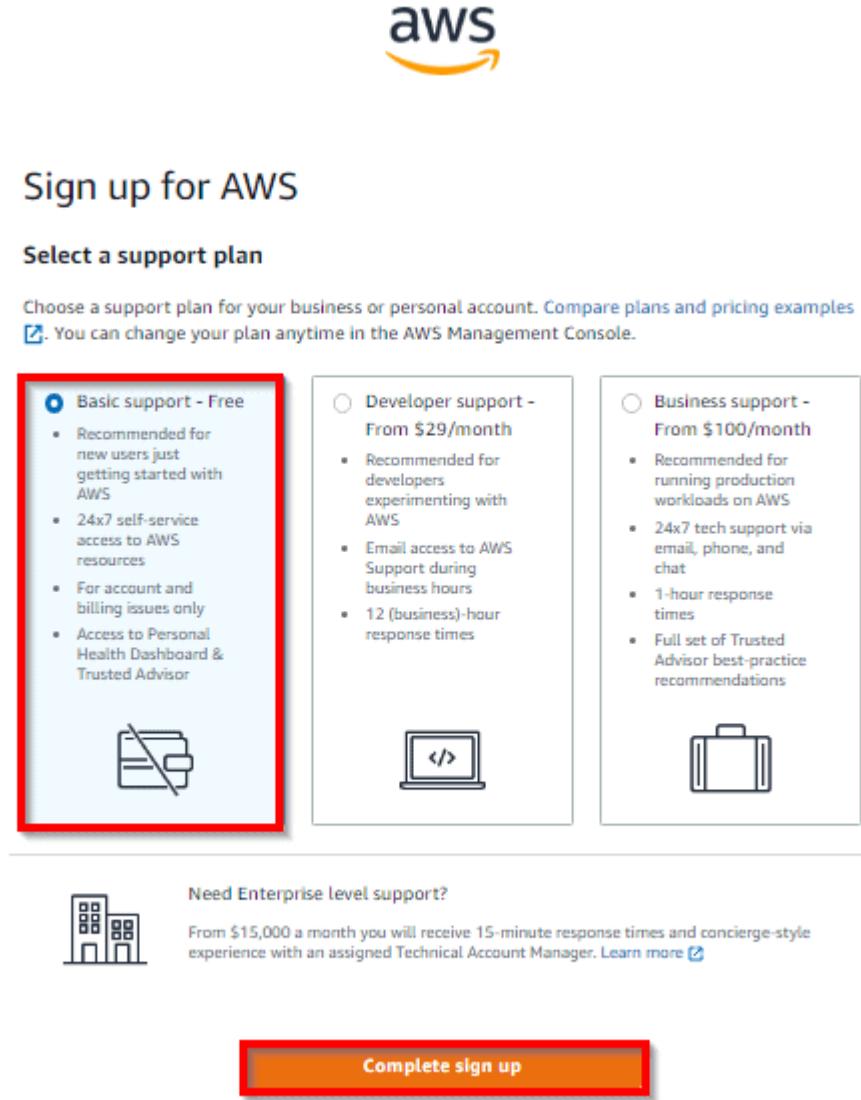
Step 5: Payment and PAN information: In this step, you must fill in your credit card /Debit Card info and billing address and click on Secure Submit.

Step 6: In this step, it will take you to the payment gateway to validate your payment information and for your credit card verification, Amazon will charge the minimum price based on Country. Here I have provided India, so Amazon charged 2 INR.

Step 7: Phone verification: Here you will be taken to an identity verification page that will already have your phone number, so you just have to select either "Text message or Voice call" Provide a valid phone number, Solve the captcha, and then click on Send SMS or Call Me Now(depending upon your selection).

Step 8: After clicking on Send SMS or Call me Now, you will immediately receive a call or SMS from Amazon, for verification code, Enter your code then click on Verify Code.

Step 9: Support plan: AWS support offers a selection of plans to meet your business needs. Select your suitable plan then click continue.



The screenshot shows the 'Sign up for AWS' page with the 'Select a support plan' section highlighted. Three support plan options are listed:

- Basic support - Free** (radio button selected, highlighted with a red box):
 - 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](#)

Complete sign up

Step 10: Registration Confirmation page.

Once you completed all the above steps and processes. You'll get the confirmation page below. Now your account will be processed for activation. It may take somewhere between 30 minutes to

1 hour for you to receive an email confirmation that your Amazon Cloud Services account has been activated.



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.](#)

Free Tier Limits

Monthly 750 hours of Amazon EC2 Cloud computing capability that is scalable.

5 GB of basic storage on Amazon S3, Infrastructure for scalable, robust, and secure object storage.

Monthly database usage allotted to Amazon RDS 750 hours (for relevant database engines) SQL Server, MariaDB, PostgreSQL, and MySQL managed relational database services.

5 GB of Amazon EFS storage. A shared file storage solution that is easy to use and scales for Amazon EC2 instances.

30 GB of General Purpose (SSD) or Magnetic Elastic Block Storage from Amazon Elastic Store are long-lasting, dependable, low-latency block-level storage volumes for EC2 instances.

Practical-8

Basics about EC2 and Steps to create Virtual machine on AWS

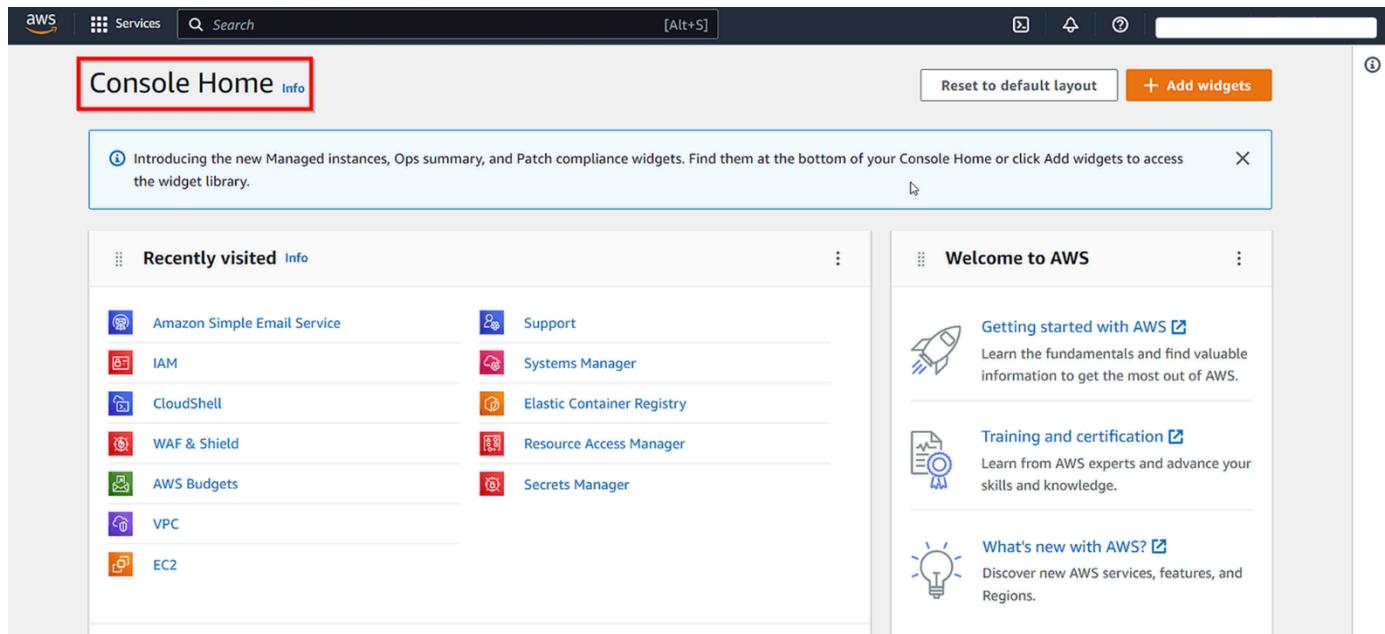
With over 32 percent of the entire world's public cloud share, it's no surprise that AWS serves more than 190 countries with scalable, reliable, and low-cost infrastructure. One of its most powerful and commonly used services are Amazon EC2 (Elastic Cloud Compute).

Amazon EC2 provides scalable computing capacity in the AWS cloud. Leveraging it enables organizations to develop and deploy applications faster, without needing to invest in hardware upfront. Users can launch virtual servers, configure security and networking, and manage cookies from an intuitive dashboard.

Among the vast array of services that Amazon offers, EC2 is the core compute component of the technology stack. In practice, EC2 makes life easier for developers by providing secure, and resizable compute capacity in the cloud. It greatly eases the process of scaling up or down, can be integrated into several other services, and comes with a plan where you only pay for how much you use it.

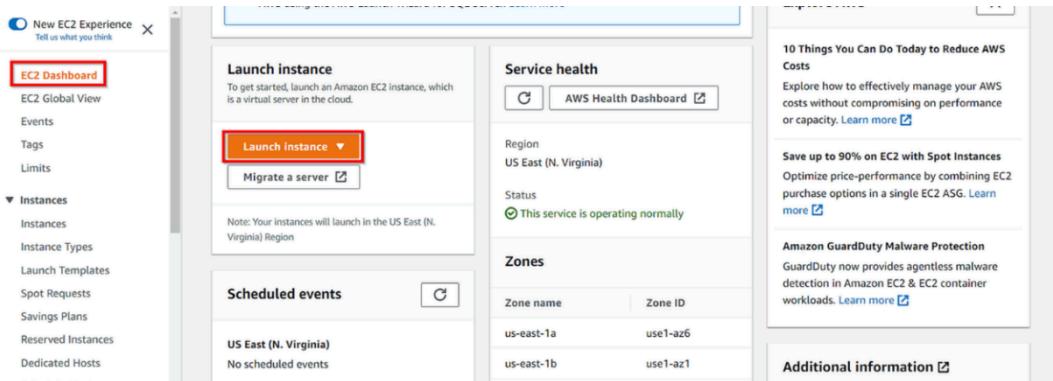
Steps to Create AWS Ubuntu Virtual Machine [EC2 Instance]

Step 1: Open your AWS console and log in.

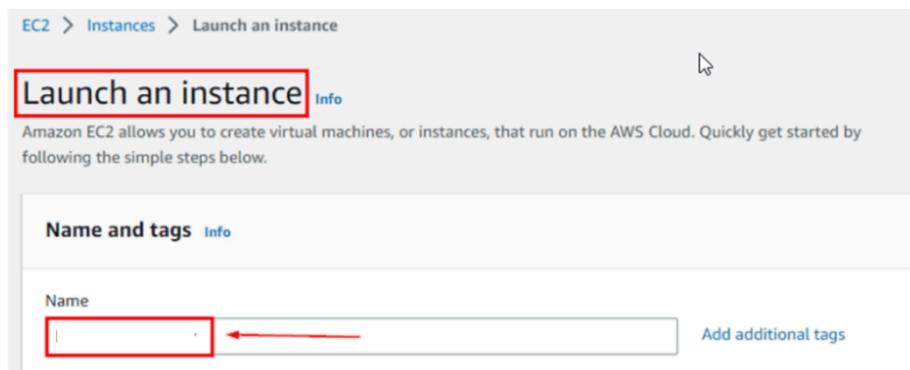


Step 2: Search for EC2 in the search bar and click on it.

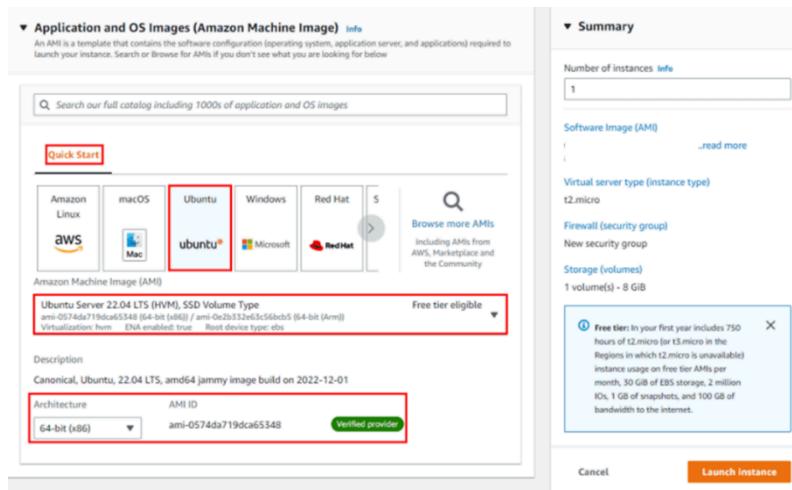
Step 3: Once you are at the EC2 Dashboard, click on Launch to launch an instance



Step 4: In the Name and Tags step you can add tags to an instance, here tags help you to enable categorizing AWS resources in different ways, for example, by owner, environment, or purpose.

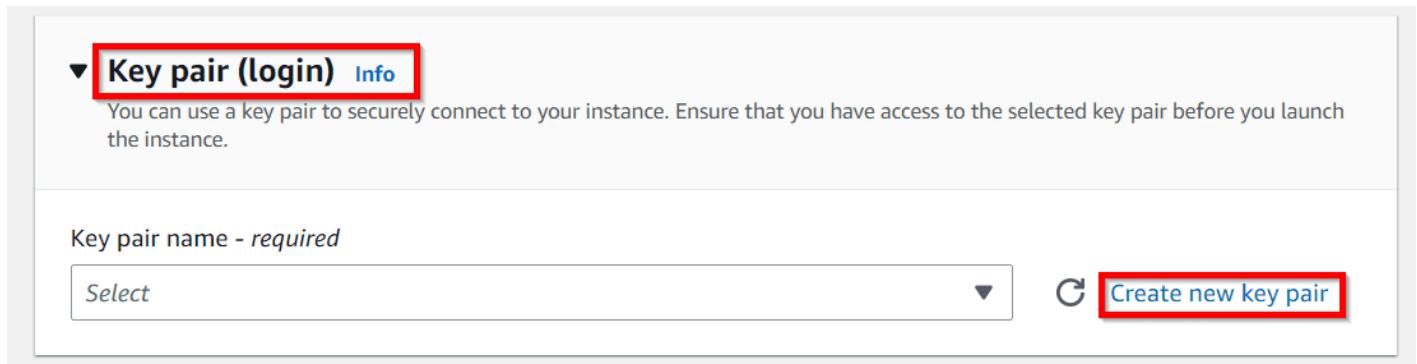


Step 5: Choose Ubuntu, select Ubuntu Server (HVM) architecture, and Click on Select.



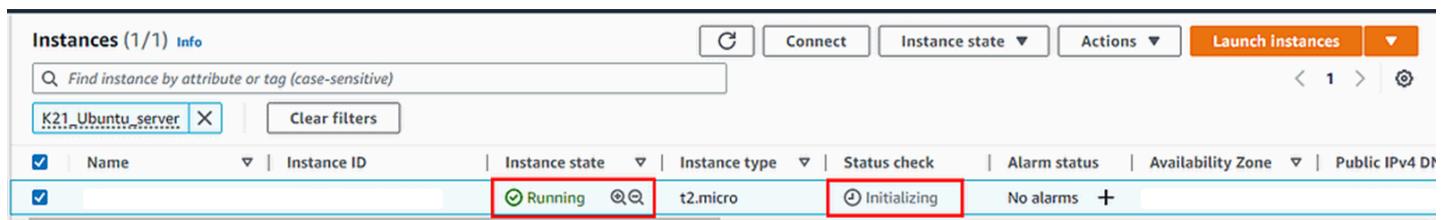
Step 6: For Select the t2.micro instance type, if you want you may select another instance type but they are chargeable so we choose the t2.micro instance type which is eligible for the free tier and limited resources.

Step 7: Select an existing key pair or create a new one, we will Create a new one, enter the name of the Key-pair as ubuntu-Key, select .ppk, and Create the Key Pair.

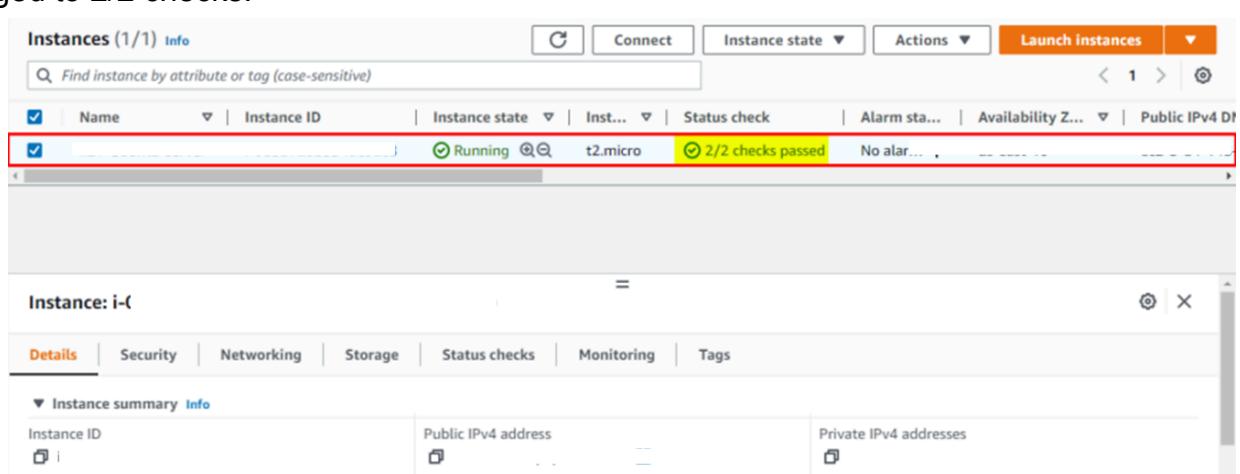


Step 8: Now review all the things you have Configured and Click on Launch Instance

Step 9: Now Click on View all Instances. Here, you shall see your instance is launching and the Status check is Initializing, wait for some time



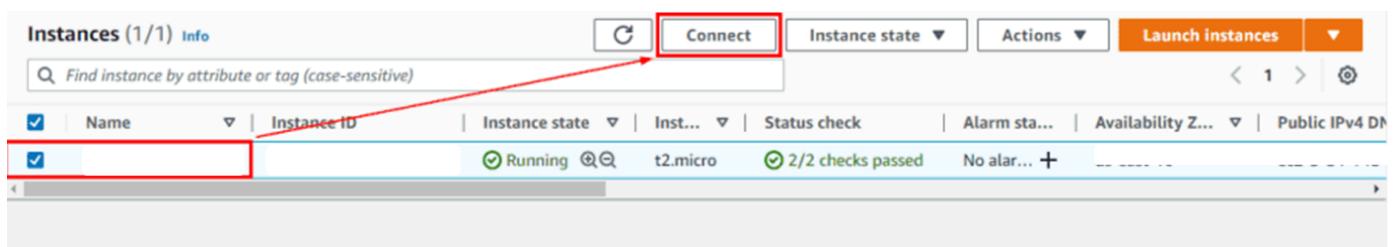
Step 10: Refresh and you shall see your instance are Up and Running, and the Status check has changed to 2/2 checks.



Now, we have successfully created our Ubuntu EC2 Instance (VM) on AWS.

Connect To Ubuntu Virtual Machine

Step 11: Select your Instance and Click on Connect.



Step 12: Select EC2 Instance Connect and Click on Connect.

Step 13: Now you shall be redirected to a new window and you shall see the Ubuntu CLI.

Practical-9

Implementation of Auto Scaling in AWS

AWS Auto-Scaling is a mechanism that automatically permits you to increase or decrease your resources to meet demand based on custom-defined metrics and thresholds. Through Auto-scaling, it's simple to set up application scaling for multiple resources across multiple services in minutes.

AWS Auto-Scaling monitors your applications continuously and adjusts the capacity automatically to take care of steady, predictable performance at the lowest possible cost. It gives the ability to ensure a correct number of EC2 instances are running every time to handle the load of the application. It helps to accomplish better availability and cost management.

Features of Auto-Scaling

- Unified Scaling: Through Auto-Scaling, we can configure automatic scaling for all of the scalable resources powering your application from one unified interface, including:
Amazon EC2, Amazon EC2 Spot Fleets, Amazon ECS, Amazon DynamoDB and Amazon Aurora
- Automatic Resource Discovery: AWS Auto-Scaling scans your environment and finds out the scalable cloud resources automatically underlying your application, so we don't need to manually identify these resources.
- Built-in scaling strategies: Through AWS Auto-Scaling, we can select one of three predefined optimization strategies designed to optimize performance, optimize costs, or balance the two. We can also set our own target resource utilization.
- Predictive Scaling: It predicts future traffic, including continuously occurring spikes, and provisions the right number of EC2 instances beforehand of predicted changes.
- Fully managed: It automatically creates target-tracking scaling policies for all of the resources in our scaling plan, using our selected scaling strategy to line the target values for every metric.
- Smart scaling policies: It continually calculates the acceptable scaling adjustments and immediately adds and removes capacity as required to stay your metrics on target.

EC2 Auto-Scaling

Amazon EC2 Auto-Scaling helps us to take care of our application availability and allows us to add or remove EC2 instances automatically according to the conditions defined. With the help of fleet management features of EC2 Auto-Scaling to maintain the health and availability of our fleet. We can also use the dynamic and predictive scaling features of EC2 Auto-Scaling to feature or remove EC2 instances.

Auto-Scaling Policy Types

- Target tracking scaling: Increase or decrease the present capacity of the group based on a target value for a selected metric.
- Step scaling: Increase or decrease the present capacity of the group based on a set of scaling adjustments, known as step adjustments, that change based on the size of the alarm breach.
- Simple scaling: Increase or decrease the present capacity of the group based on a single scaling adjustment

Practical-10

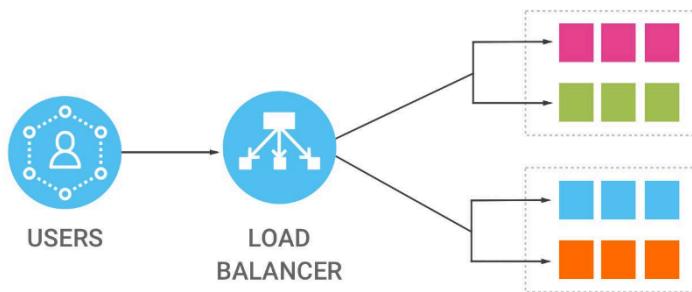
Learn the working of Load Balancer and types of Load Balancer in AWS

A load-balancing service for AWS deployments is called Elastic Load Balancing (ELB). It scales resources and automatically distributes incoming application traffic to handle heavy traffic demands.

Elastic Load Balancing (ELB) is a service offered by Amazon Web Services (AWS) that automatically distributes incoming network traffic across multiple targets, such as Amazon Elastic Compute Cloud (EC2) instances, containers, and IP addresses, in one or more Availability Zones.

ELB is designed to improve the availability, fault tolerance, and scalability of your applications by efficiently distributing traffic and ensuring that it is routed only to healthy targets.

In order to avoid sending traffic to servers that are unable to process requests, load balancers monitor the health of the back-end resources. The process of redistributing workload in a distributed system to ensure that no computer is overloaded, underloaded, or idle is known as load balancing. It makes an effort to accelerate several aspects of the cloud environment, such as reaction time, execution time, system stability, etc. The load-balancing survey papers were unable to offer an accurate, systematic classification of methods and procedures.



Elastic Load Balancing in AWS

- Distributes incoming application traffic across multiple targets, such as EC2 instances, containers (ECS), Lambda functions, and IP addresses in multiple Availability Zones(AZs).
- Distributes Client traffic across servers.
- Improves the performance of applications.
- It Accepts incoming traffic from clients and routes requests to its registered targets.

- Monitors the condition of its registered targets and routes traffic only to healthy targets.
- Enable deletion protection to stop your load balancer from being deleted accidentally. Disabled by default.
- Deleting ELB won't delete the instances registered to it.
- Cross Zone Load Balancing – If enabled, each load balancer node automatically distributes traffic across the registered targets in all enabled AZs.
- Supports SSL Offloading which is a feature that allows the AWS Elastic Load Balancer to bypass the SSL termination by removing the SSL-based encryption from the incoming traffic.

AWS Load Balancer Types

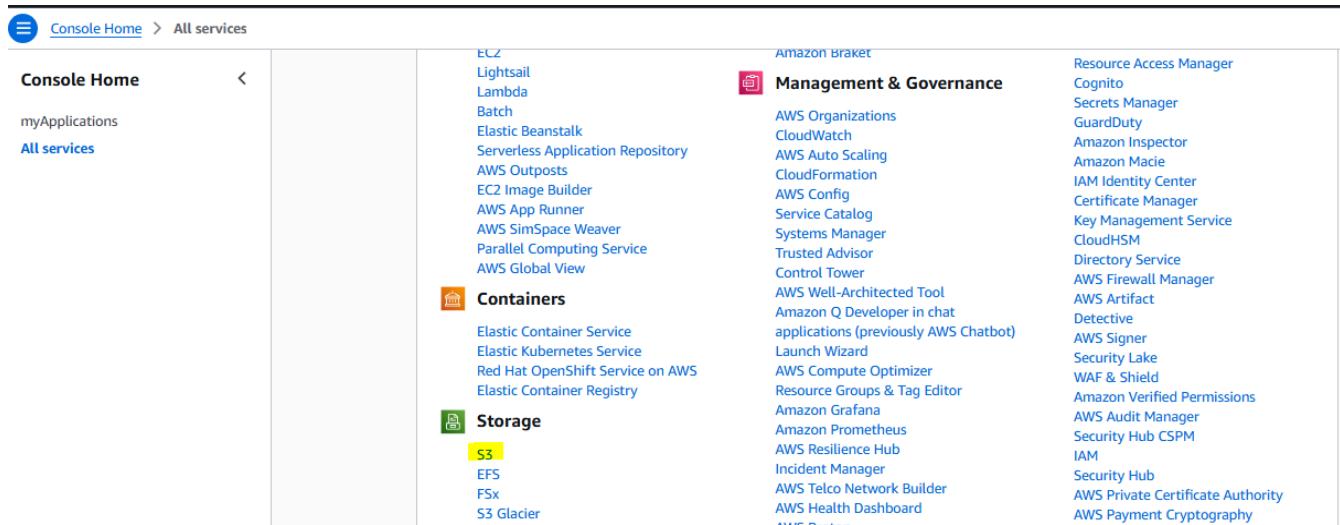
- Application Load Balancers: Ideal for routing HTTP/HTTPS traffic and performing advanced traffic routing and content-based routing.
- Network Load Balancers: Designed for handling TCP/UDP traffic with high performance and low latency.
- Gateway Load Balancers: Used for deploying third-party virtual appliances, such as firewalls, intrusion detection systems, and other network appliances.
- Classic Load Balancers: An older type of load balancer that is still available for use, primarily for applications not yet migrated to the newer load balancer types.

Practical-11

Create cloud storage Bucket using Amazon Simple Storage Service (S3)

Step 1: Login to AWS Management Console

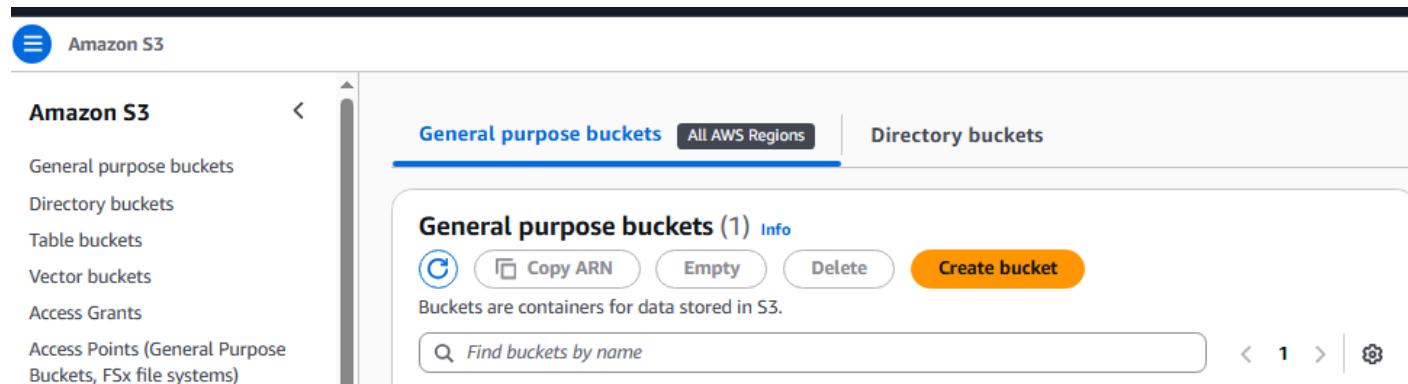
Step 2: In the Search bar, type S3 and open the S3 service or Goto “All services” “Storage” S3



Console Home > All services

Category	Service
EC2	Lightsail
Amazon Braket	AWS Organizations
Management & Governance	CloudWatch
Containers	AWS Auto Scaling
Storage	CloudFormation
S3	AWS Config
EFS	Service Catalog
FSx	Systems Manager
S3 Glacier	Trusted Advisor
	Control Tower
	AWS Well-Architected Tool
	Amazon Q Developer in chat applications (previously AWS Chatbot)
	Launch Wizard
	AWS Compute Optimizer
	Resource Groups & Tag Editor
	Amazon Grafana
	Amazon Prometheus
	AWS Resilience Hub
	Incident Manager
	AWS Telco Network Builder
	AWS Health Dashboard
	AWS CloudTrail
	AWS Firewall Manager
	AWS Artifact
	Detective
	AWS Signer
	Security Lake
	WAF & Shield
	Amazon Verified Permissions
	AWS Audit Manager
	Security Hub CSPM
	IAM
	Security Hub
	AWS Private Certificate Authority
	AWS Payment Cryptography

Step 3: Click Create Bucket



Amazon S3

Amazon S3

General purpose buckets All AWS Regions Directory buckets

General purpose buckets (1) Info

(1) Copy ARN Empty Delete **Create bucket**

Buckets are containers for data stored in S3.

Find buckets by name

Step 4: Enter a Bucket name (must be globally unique, e.g., my-student-s3-bucket).

Create bucket Info

Buckets are containers for data stored in S3.

General configuration

AWS Region
Europe (Stockholm) eu-north-1

Bucket type Info

General purpose
Recommended for most use cases and access patterns. General purpose buckets are the original S3 bucket type. They allow a mix of storage classes that redundantly store objects across multiple Availability Zones.

Directory
Recommended for low-latency use cases. These buckets use only the S3 Express One Zone, which provides faster processing of data within a single Availability Zone.

Bucket name Info
myfirstbucketforlearn

Bucket names must be 3 to 63 characters and unique within the global namespace. Bucket names must also begin and end with a letter or number. Valid characters are a-z, 0-9, periods (.), and hyphens (-). [Learn More](#)

Copy settings from existing bucket - optional
Only the bucket settings in the following configuration are copied.

Step 5: Configure options:

- Block Public Access (recommended: keep enabled).
- Enable/disable versioning (optional).

Object Ownership Info

Control ownership of objects written to this bucket from other AWS accounts and the use of access control lists (ACLs). Object ownership determines who can specify access to objects.

ACLs disabled (recommended)
All objects in this bucket are owned by this account. Access to this bucket and its objects is specified using only policies.

ACLs enabled
Objects in this bucket can be owned by other AWS accounts. Access to this bucket and its objects can be specified using ACLs.

Object Ownership
Bucket owner enforced

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.

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 - optional (0)

You can use bucket tags to track storage costs and organize buckets. [Learn more](#)

No tags associated with this bucket.

[Add new tag](#)

You can add up to 50 tags.

Step 6: Click Create Bucket.

Default encryption [Info](#)

Server-side encryption is automatically applied to new objects stored in this bucket.

Encryption type [Info](#)

Secure your objects with two separate layers of encryption. For details on pricing, see DSSE-KMS pricing on the Storage tab of the [Amazon S3 pricing page](#).

- Server-side encryption with Amazon S3 managed keys (SSE-S3)
- Server-side encryption with AWS Key Management Service keys (SSE-KMS)
- Dual-layer server-side encryption with AWS Key Management Service keys (DSSE-KMS)

Bucket Key

Using an S3 Bucket Key for SSE-KMS reduces encryption costs by lowering calls to AWS KMS. S3 Bucket Keys aren't supported for DSSE-KMS. [Learn more](#)

- Disable
- Enable

► Advanced settings

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

[Cancel](#)

[Create bucket](#)

Step 7: Once created, click the bucket name to open it.

General purpose buckets (2) [Info](#)

Buckets are containers for data stored in S3.

<input type="text"/> Find buckets by name			Copy ARN	Empty	Delete	Create bucket
Name	AWS Region	Creation date				
do-not-delete-ssm-diagnosis-571594412096-eu-north-1-b8z66	Europe (Stockholm) eu-north-1	August 18, 2025, 16:46:01 (UTC+05:30)				
myfirstbucketforlearnsamirkariya	Europe (Stockholm) eu-north-1	September 10, 2025, 19:14:32 (UTC+05:30)				

Step 8: Upload a file by choosing Upload → Add files → Upload.

myfirstbucketforlearnsamirkariya Info

Objects Properties Permissions Metrics Management Access Points

Objects (0)

Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. [Learn more](#)

Find objects by prefix ✖

Name	Type	Last modified	Size	Storage class
No objects You don't have any objects in this bucket.				

[Upload](#)

Step 9: Verify the object is stored successfully.

Practical-12

Build, Deploy and Manage web application on docker.

Step 1: Create Project Structure

```
webapp/
└── index.html
└── style.css
└── Dockerfile
```

Step 2: Create HTML File (**index.html**)

```
<!DOCTYPE html>
<html>
<head>
    <title>My Docker WebApp</title>
    <link rel="stylesheet" type="text/css" href="style.css">
</head>
<body>
    <h1>Hello, Docker World!</h1>
    <p>This is a simple web app using HTML & CSS inside Docker.</p>
</body>
</html>
```

Step 3: Create CSS File (**style.css**)

```
body {
    font-family: Arial, sans-serif;
    background-color: #f4f4f9;
    text-align: center;
    margin-top: 100px;
}
h1 {
    color: #2c3e50;
}
p {
    color: #16a085;
    font-size: 18px;
}
```

Step 4: Create **Dockerfile**

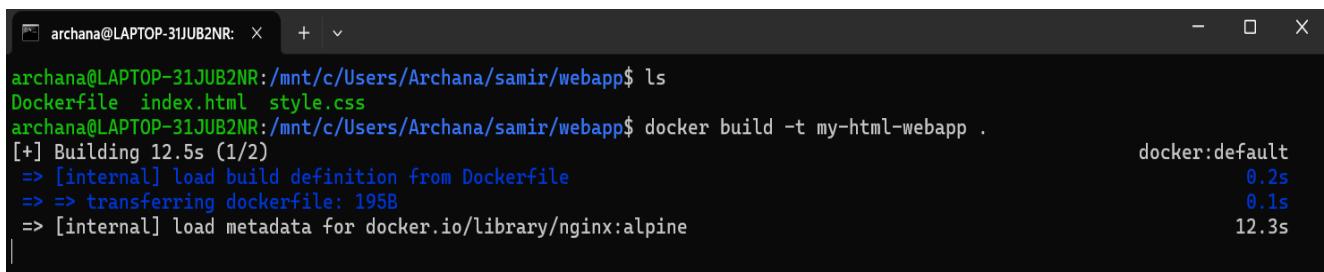
```
# Use Nginx as base image
```

```
FROM nginx:alpine
```

```
# Copy website files to Nginx default directory
COPY ./usr/share/nginx/html
# Expose port 80
EXPOSE 80
```

Step 5: Build Docker Image

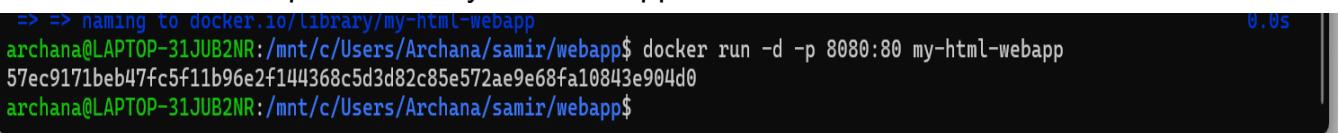
```
$ docker build -t my-html-webapp .
```



```
archana@LAPTOP-31JUB2NR:~/mnt/c/Users/Archana/samir/webapp$ ls
Dockerfile index.html style.css
archana@LAPTOP-31JUB2NR:~/mnt/c/Users/Archana/samir/webapp$ docker build -t my-html-webapp .
[+] Building 12.5s (1/2)
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 195B
=> [internal] load metadata for docker.io/library/nginx:alpine
|
```

Step 6: Run Container

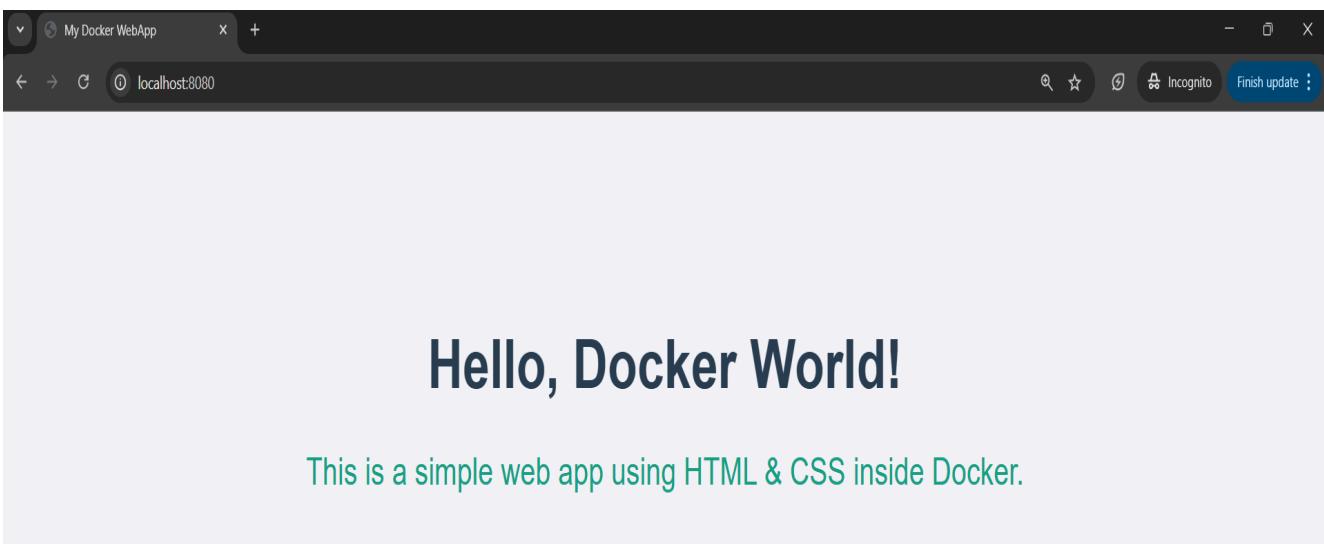
```
$ docker run -d -p 8080:80 my-html-webapp
```



```
=> => naming to docker.io/library/my-html-webapp
archana@LAPTOP-31JUB2NR:~/mnt/c/Users/Archana/samir/webapp$ docker run -d -p 8080:80 my-html-webapp
57ec9171beb47fc5f11b96e2f144368c5d3d82c85e572ae9e68fa10843e904d0
archana@LAPTOP-31JUB2NR:~/mnt/c/Users/Archana/samir/webapp$
```

Step 7: Access Application

Open browser → <http://localhost:8080>



Step 8: Manage Container

```
archana@LAPTOP-31JUB2NR:/mnt/c/Users/Archana/samir/webapp$ docker run -d -p 8080:80 my-html-webapp
57ec9171beb47fc5f11b96e2f144368c5d3d82c85e572ae9e68fa10843e904d0
archana@LAPTOP-31JUB2NR:/mnt/c/Users/Archana/samir/webapp$ docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS
57ec9171beb4 my-html-webapp "/docker-entrypoint..." About a minute ago Up About a minute 0.0.0.
0:8080->80/tcp, [::]:8080->80/tcp cranky_torvalds
5f23616f1b27 hyperledger/fabric-peer:2.5.12 "peer node start"
0:9051->7051/tcp, [::]:9051->7051/tcp peer0.org2.example.com
fec786044262 hyperledger/fabric-peer:2.5.12 "peer node start"
0:7051->7051/tcp, [::]:7051->7051/tcp peer0.org1.example.com
archana@LAPTOP-31JUB2NR:/mnt/c/Users/Archana/samir/webapp$ docker stop 57ec9171beb4
57ec9171beb4
archana@LAPTOP-31JUB2NR:/mnt/c/Users/Archana/samir/webapp$ docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS
5f23616f1b27 hyperledger/fabric-peer:2.5.12 "peer node start" About an hour ago Up About an hour 0.0.0.0:9051->7051/tcp, [::]:9051->7051/tcp peer0.org2.example.com
fec786044262 hyperledger/fabric-peer:2.5.12 "peer node start" 2 hours ago Up 2 hours 0.0.0.0:7051->7051/tcp, [::]:7051->7051/tcp peer0.org1.example.com
archana@LAPTOP-31JUB2NR:/mnt/c/Users/Archana/samir/webapp$ |
```

Practical-13

Introduction to Jenkins pipelines and basic pipeline creation

What is a Jenkins pipeline?

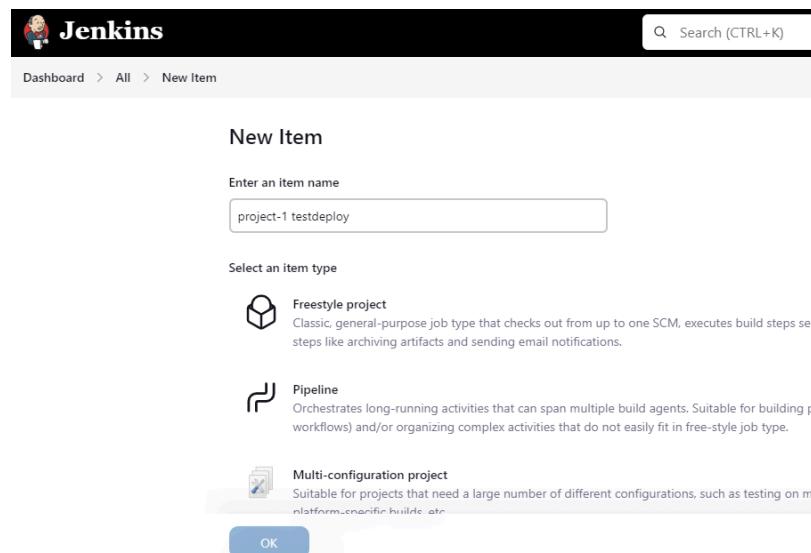
- A Jenkins pipeline is a set of plug-ins to create automated, recurring workflows that constitute CI/CD pipelines.
- A Jenkins pipeline includes all the tools you need to orchestrate testing, merging, packaging, shipping, and code deployment.
- A pipeline is typically divided into multiple stages and steps, with each step representing a single task and each stage grouping together similar steps.
- For example, you may have “Build”, “Test”, and “Deploy” stages in your pipeline. You can also run existing jobs within a pipeline.

Pipelines offer several benefits. You can:

- Fast-track the delivery of code to production
- Automate build generation for pull requests, ensuring no syntax errors are merged to the main branch/repository
- Perform automated unit, sanity, and regression testing
- Create customized automation workflows for different clients, environments, or products.
- Ensure security best practices are followed by performing static code analysis, vulnerability scanning, and penetration testing on every commit
- Reduce the need for manual maintenance, testing, and deployment, allowing your developers and DevOps engineers to focus on more productive tasks

Steps

- Add new item



New Item

Enter an item name

project-1 testdeploy

Select an item type

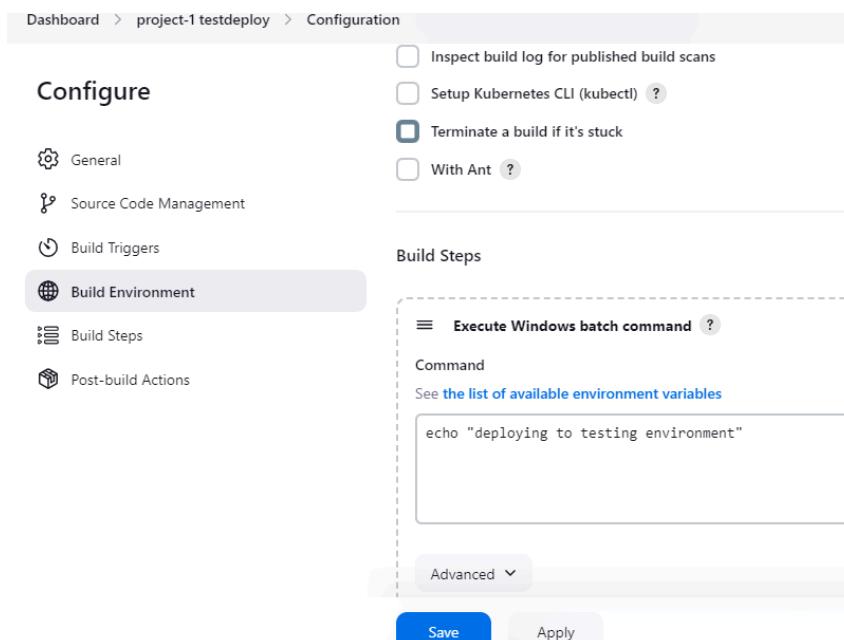
Freestyle project
Classic, general-purpose job type that checks out from up to one SCM, executes build steps series like archiving artifacts and sending email notifications.

Pipeline
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines and/or organizing complex activities that do not easily fit in free-style job type.

Multi-configuration project
Suitable for projects that need a large number of different configurations, such as testing on multiple platforms, environments, builds etc.

OK

- Add Build Steps



Dashboard > project-1 testdeploy > Configuration

Configure

- General
- Source Code Management
- Build Triggers
- Build Environment**
- Build Steps
- Post-build Actions

Build Steps

Execute Windows batch command

Command

See [the list of available environment variables](#)

```
echo "deploying to testing environment"
```

Advanced ▾

Save Apply

- Add plugins

Q build pipe /

Name ↓

Enabled

Build Pipeline Plugin 2.0.2

This plugin renders upstream and downstream connected jobs that typically form a build pipeline. In addition, it offers the ability to define manual triggers for jobs that require intervention prior to execution, e.g. an approval process outside of Jenkins.

[Report an issue with this plugin](#)

- Add post build actions and trigger option

Dashboard > project-1 build > Configuration

Configure

Post-build Actions

General

Source Code Management

Build Triggers

Build Environment

Build Steps

Post-build Actions

Build other projects ?

Projects to build

project-1 testdeploy

Trigger only if build is stable

Trigger even if the build is unstable

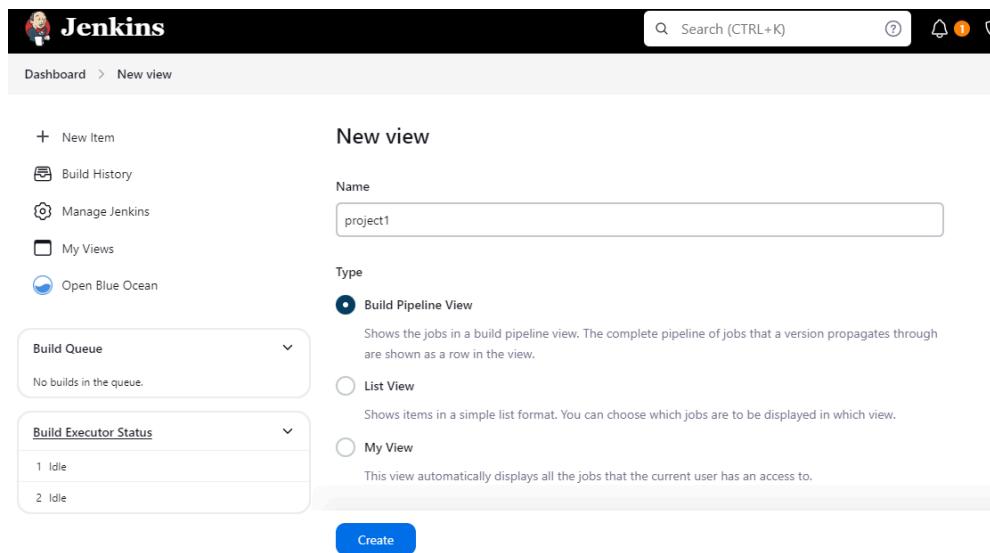
Trigger even if the build fails

Add post-build action ▾

Save

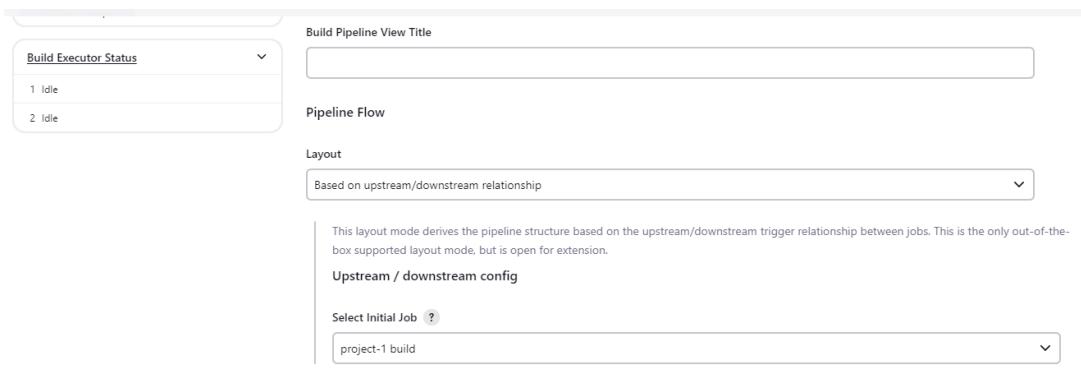
Apply

- Add new view



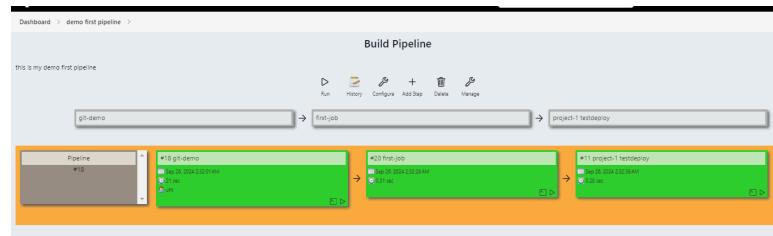
The screenshot shows the Jenkins dashboard with the URL [http://localhost:8080](#). The top navigation bar includes links for 'Dashboard', 'New view', 'Build History', 'Manage Jenkins', 'My Views', and 'Open Blue Ocean'. A search bar at the top right contains the placeholder 'Search (CTRL+K)'. On the left, there are two collapsed sections: 'Build Queue' (No builds in the queue) and 'Build Executor Status' (1 Idle, 2 Idle). The main area is titled 'New view' and contains fields for 'Name' (set to 'project1') and 'Type' (selected as 'Build Pipeline View'). A detailed description of 'Build Pipeline View' is provided: 'Shows the jobs in a build pipeline view. The complete pipeline of jobs that a version propagates through are shown as a row in the view.' Below this are three other options: 'List View' (Shows items in a simple list format. You can choose which jobs are to be displayed in which view.) and 'My View' (This view automatically displays all the jobs that the current user has an access to.). A blue 'Create' button is located at the bottom of the form.

- Add pipe line details



This screenshot shows the Jenkins Pipeline configuration page. It includes a sidebar with 'Build Executor Status' (1 Idle, 2 Idle) and a main panel for 'Build Pipeline View Title' (empty). The 'Pipeline Flow' section contains a 'Layout' dropdown set to 'Based on upstream/downstream relationship'. A note explains that this layout mode derives the pipeline structure based on upstream/downstream trigger relationships. The 'Upstream / downstream config' section includes a 'Select Initial Job' dropdown containing 'project-1 build'.

View of Jenkins pipeline

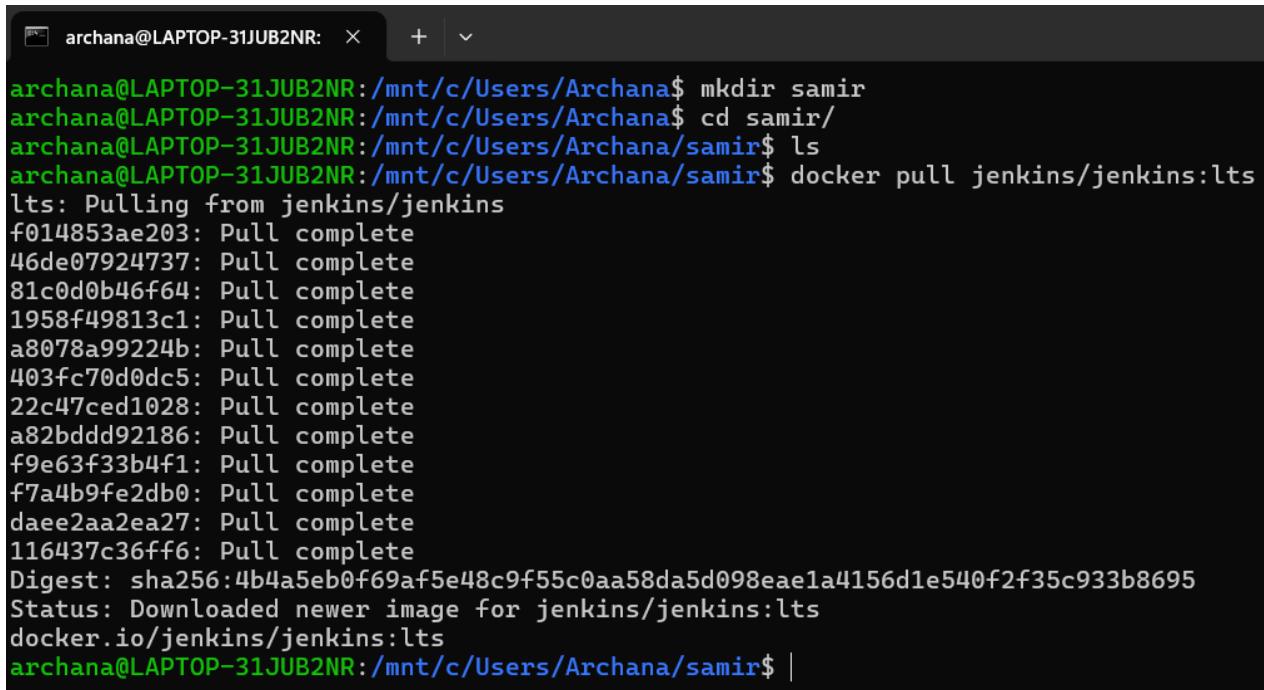


Practical-14

Run Jenkins in docker container

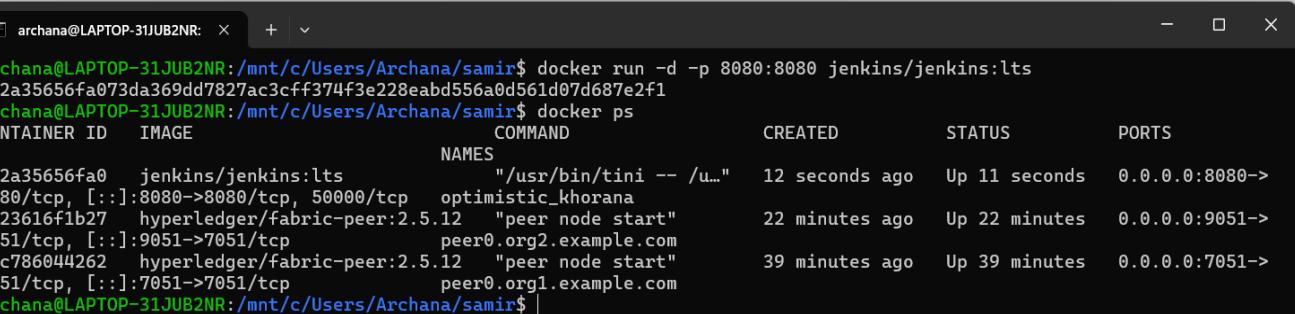
Step 1: Pull Jenkins Docker Image

```
$ docker pull jenkins/jenkins:lts
```



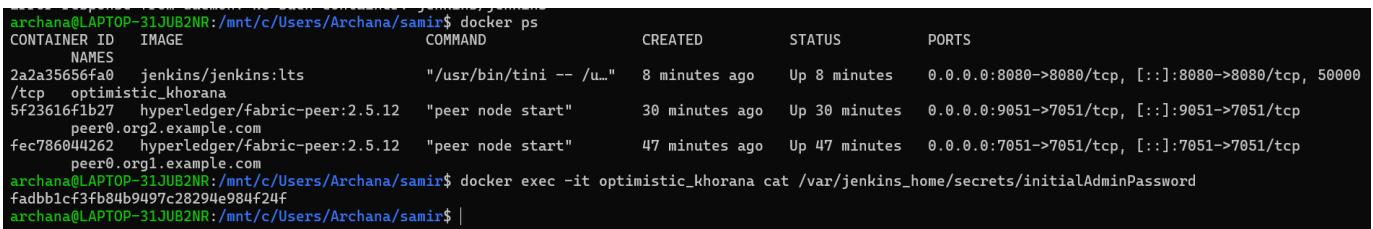
```
archana@LAPTOP-31JUB2NR:~$ docker pull jenkins/jenkins:lts
lts: Pulling from jenkins/jenkins
f014853ae203: Pull complete
46de07924737: Pull complete
81c0d0b46f64: Pull complete
1958f49813c1: Pull complete
a8078a99224b: Pull complete
403fc70d0dc5: Pull complete
22c47ced1028: Pull complete
a82bddd92186: Pull complete
f9e63f33b4f1: Pull complete
f7a4b9fe2db0: Pull complete
daee2aa2ea27: Pull complete
116437c36ff6: Pull complete
Digest: sha256:4b4a5eb0f69af5e48c9f55c0aa58da5d098eae1a4156d1e540f2f35c933b8695
Status: Downloaded newer image for jenkins/jenkins:lts
docker.io/jenkins/jenkins:lts
archana@LAPTOP-31JUB2NR:~$
```

Step 2: Run Jenkins Container



```
archana@LAPTOP-31JUB2NR:~$ docker run -d -p 8080:8080 jenkins/jenkins:lts
2a2a35656fa073da369dd7827ac3cff374f3e228eabd556a0d561d07d687e2f1
archana@LAPTOP-31JUB2NR:~$ docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS
NAMES
2a2a35656fa0 jenkins/jenkins:lts "/usr/bin/tini -- /u..." 12 seconds ago Up 11 seconds 0.0.0.0:8080->
8080/tcp, [::]:8080->8080/tcp, 50000/tcp optimisitic_khorana
5f23616fb27 hyperledger/fabric-peer:2.5.12 "peer node start" 22 minutes ago Up 22 minutes 0.0.0.0:9051->
7051/tcp, [::]:9051->7051/tcp peer0.org2.example.com
fec786044262 hyperledger/fabric-peer:2.5.12 "peer node start" 39 minutes ago Up 39 minutes 0.0.0.0:7051->
7051/tcp, [::]:7051->7051/tcp peer0.org1.example.com
archana@LAPTOP-31JUB2NR:~$
```

Step 3: Get Jenkins Admin Password

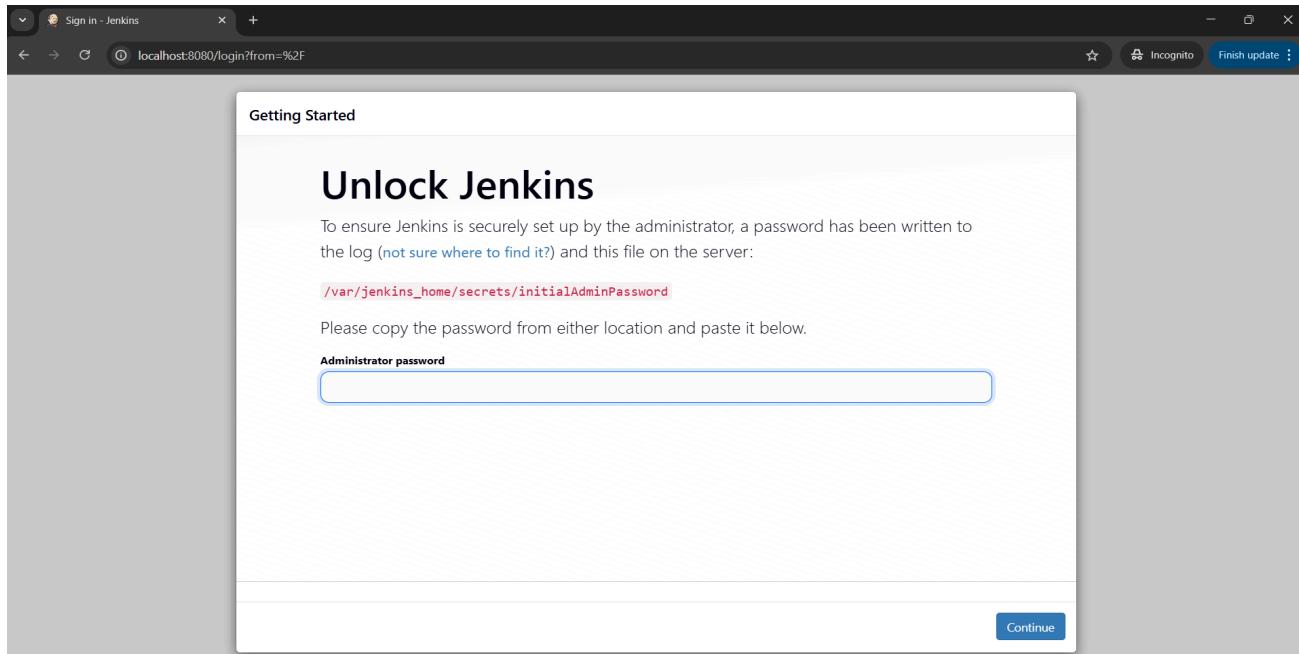


```
archana@LAPTOP-31JUB2NR:~$ docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS
NAMES
2a2a35656fa0 jenkins/jenkins:lts "/usr/bin/tini -- /u..." 8 minutes ago Up 8 minutes 0.0.0.0:8080->8080/tcp, [::]:8080->8080/tcp, 50000/tcp
/tcp optimisitic_khorana
5f23616fb27 hyperledger/fabric-peer:2.5.12 "peer node start" 30 minutes ago Up 30 minutes 0.0.0.0:9051->7051/tcp, [::]:9051->7051/tcp
peer0.org2.example.com
fec786044262 hyperledger/fabric-peer:2.5.12 "peer node start" 47 minutes ago Up 47 minutes 0.0.0.0:7051->7051/tcp, [::]:7051->7051/tcp
peer0.org1.example.com
archana@LAPTOP-31JUB2NR:~$ docker exec -it optimisitic_khorana cat /var/jenkins_home/secrets/initialAdminPassword
fadbb1cf3fb84b9497c28294e984f24f
archana@LAPTOP-31JUB2NR:~$
```

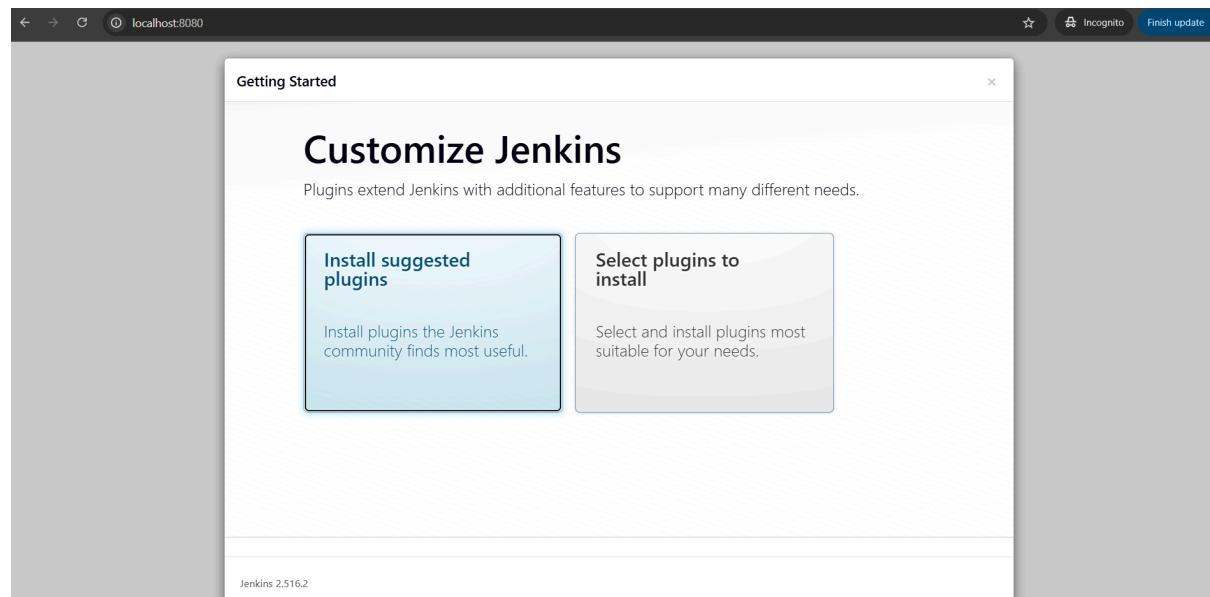
Step 4: Step 5: Access Jenkins UI

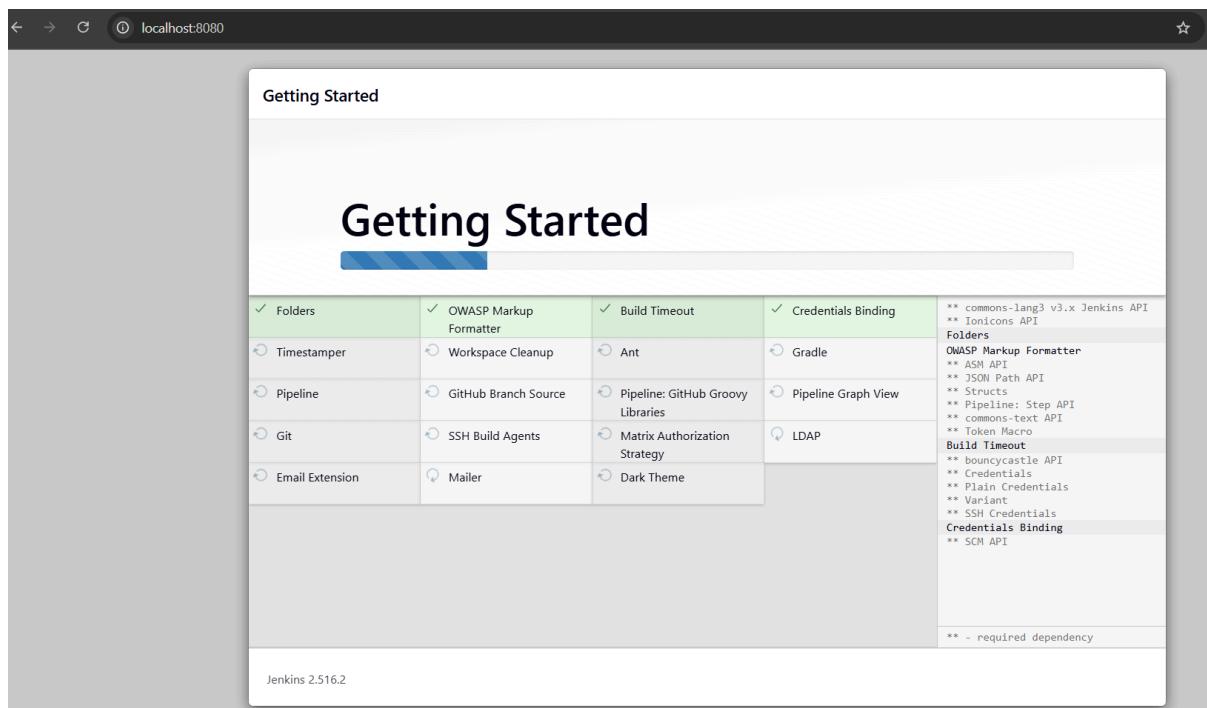
Open browser → <http://localhost:8080>

Paste the initial admin password.

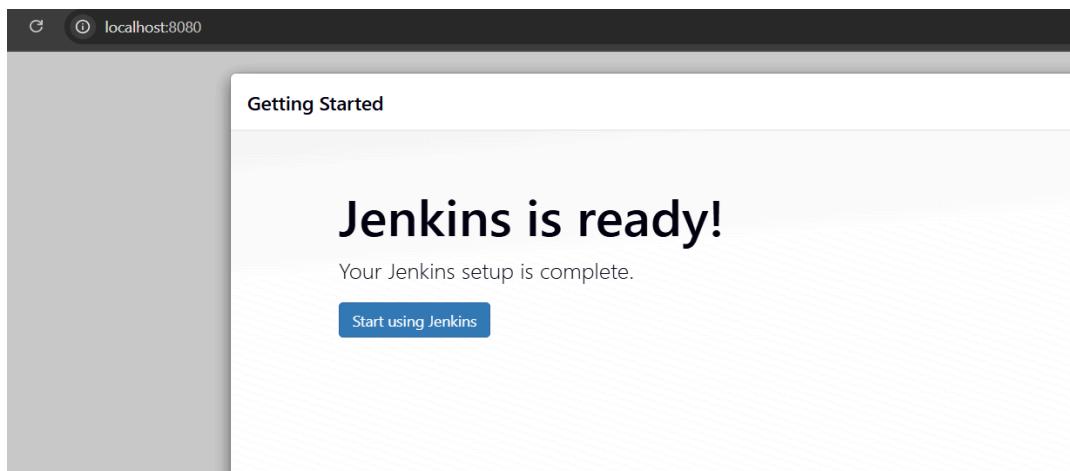


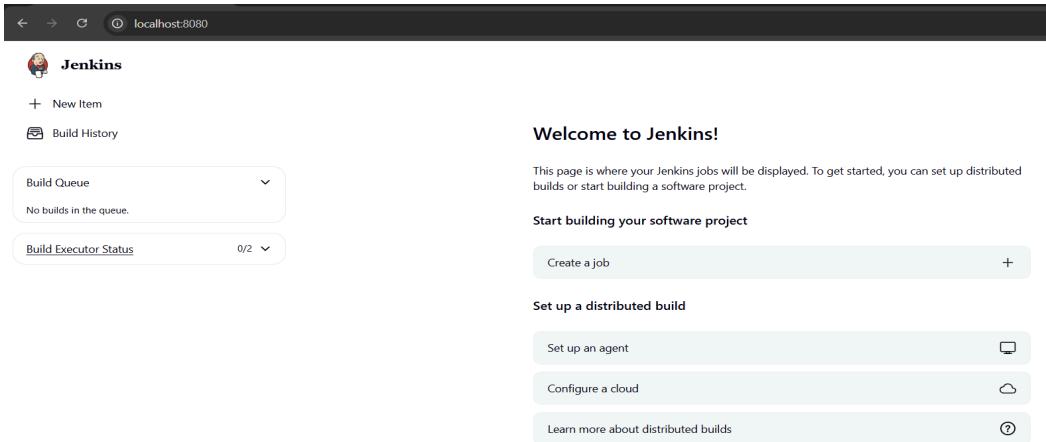
Step 5: Install recommended plugins.





Step 6: Enter User Detail and Jenkins is ready!





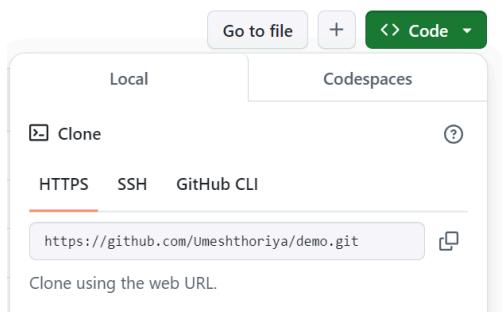
The screenshot shows the Jenkins welcome page. At the top, there are links for 'New Item' and 'Build History'. Below that, a 'Build Queue' section indicates 'No builds in the queue.' To the right, a large 'Welcome to Jenkins!' heading is displayed, followed by a message: 'This page is where your Jenkins jobs will be displayed. To get started, you can set up distributed builds or start building a software project.' A 'Start building your software project' button is present. Further down, under 'Set up a distributed build', there are three buttons: 'Create a job' (with a '+' icon), 'Set up an agent' (with a monitor icon), and 'Configure a cloud' (with a cloud icon). A link 'Learn more about distributed builds' with a help icon is also shown.

Practical-15

Integrate GitHub with Jenkins with public and private repositories.

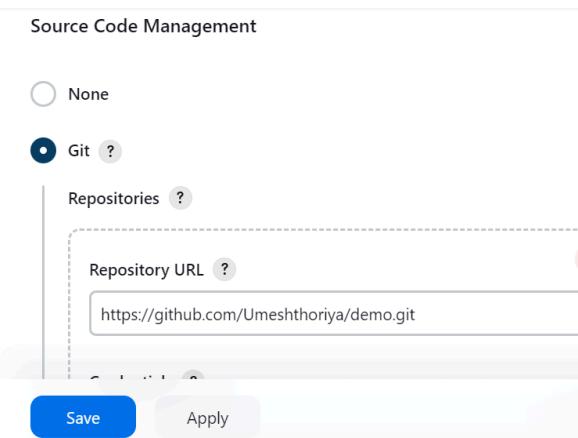
Github steps

- Add new Repositories
- Add some codes in repositories
- Copy clone : HTTPS



Jenkins steps

- Add new item
- In source code management add Github link



- Set build trigger option

Build Triggers

Trigger builds remotely (e.g., from scripts) ?

Build after other projects are built ?

Build periodically ?

Build when a change is pushed to GitLab. GitLab webhook URL: http://localhost:8080/project/git-

GitHub hook trigger for GITScm polling ?

Poll SCM ?

Schedule ?

```
* * * * *
```

Save **Apply**

- Add build steps

Build Steps

Execute Windows batch command ?

Command

See the list of available environment variables

```
python hello.py
```

Advanced ▾

- Click on build now
- See the console output'

Status **Console Output** Download

</> Changes

Console Output

Edit Build Information

Timings

Git Build Data

Open Blue Ocean

← Previous Build

```

Started by user uht
Running on SYSTEM
Building in workspace C:\ProgramData\Jenkins\jenkins\workspace\git-demo
The recommended git tool is: NONE
No credentials specified
> git.exe rev-parse --resolve-git-dir C:\ProgramData\Jenkins\jenkins\workspace\git-demo.git # timeout=10
Fetching changes from the remote git repository
> git.exe config remote.origin.url https://github.com/meshtioriya/demo.git # timeout=10
Fetching upstream changes from https://github.com/meshtioriya/demo.git
> git.exe -version # timeout=10
> git --version # timeout=10
> git version 2.43.0.windows.1
> git.exe fetch --tags --force --progress -- https://github.com/meshtioriya/demo.git +refs/heads/*:refs/remotes/origin/*
Seen branch in repository origin/main
Seen 1 remote branch
> git.exe show-ref --tags -d # timeout=10
Checking out Revision 190bf13be75a54bd3c4d5e073f97804e15985326 (origin/main)
> git.exe config core.sparsecheckout # timeout=10
> git.exe checkout -f 190bf13be75a54bd3c4d5e073f97804e15985326 # timeout=10
Commit message: "update hello.py"
> git.exe rev-list --no-walk 190bf13be75a54bd3c4d5e073f97804e15985326 # timeout=10
[git-demo] $ cmd /c call C:\WINDOWS\TEMP\jenkins1739720117522489098.bat

C:\ProgramData\Jenkins\jenkins\workspace\git-demo>python hello.py
Hello github update ......

C:\ProgramData\Jenkins\jenkins\workspace\git-demo>exit 0
Triggering a new build of First-job
Finished: SUCCESS

```