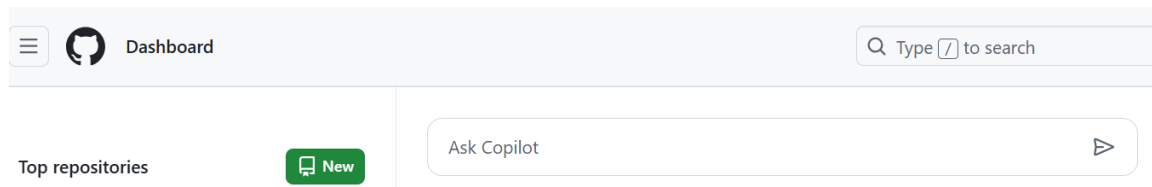


## TO RUN OUR APPLICATION ON SERVER LEVEL

**Step-1:** Go to GitHub account and login with email and password after entering it will open dashboard page.

### To Create a Repository

1. Click on > **New**



2. Give Repository Name >

3. Write Description >


4. Select > public

5. Select > Add a README file

#### Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)


Required fields are marked with an asterisk (\*).


Owner \* / Repository name \*  
 Mahithamoyella / employee-leave-code-new  
✔ employee-leave-code-new is available.

Great repository names are short and memorable. Need inspiration? How about **musical-succotash** ?

Description (optional)

employee-leave-code

☒  Public  
Anyone on the internet can see this repository. You choose who can commit.


☐  Private  
You choose who can see and commit to this repository.

Initialize this repository with:

☒ Add a README file  
This is where you can write a long description for your project. [Learn more about READMEs.](#)

6. Click on > create Repository

This will set `main` as the default branch. Change the default name in your [settings](#).

 You are creating a public repository in your personal account.

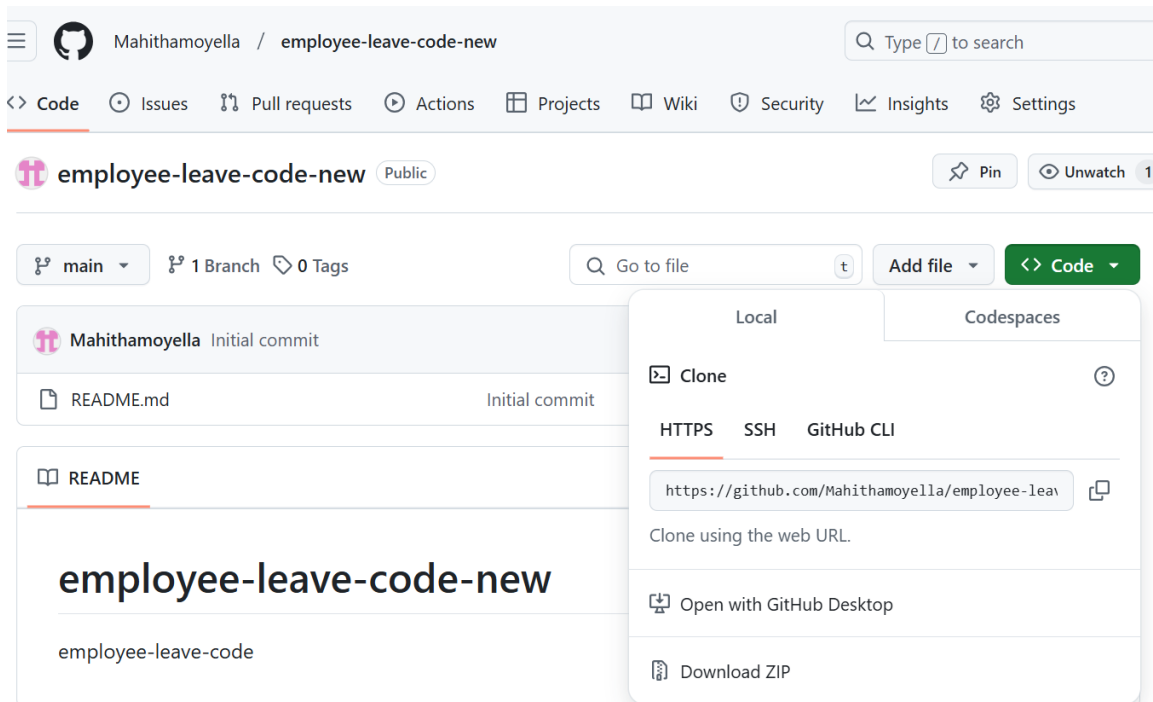
Create repository

Click on GitHub logo

On Left side You can see the > Find a Repository in that type your created repository name and Open that Repository

## Step -2:

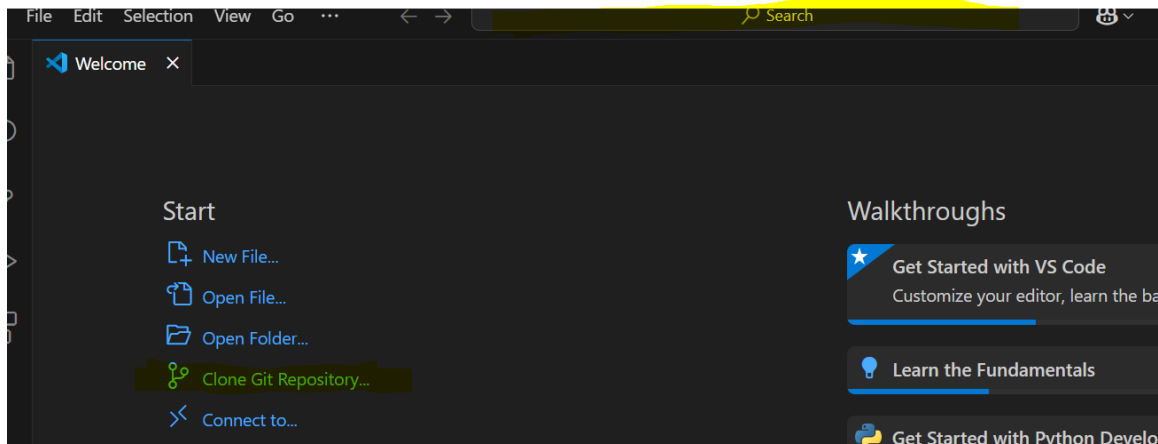
Click on the CODE and Select HTTPS copy the URL



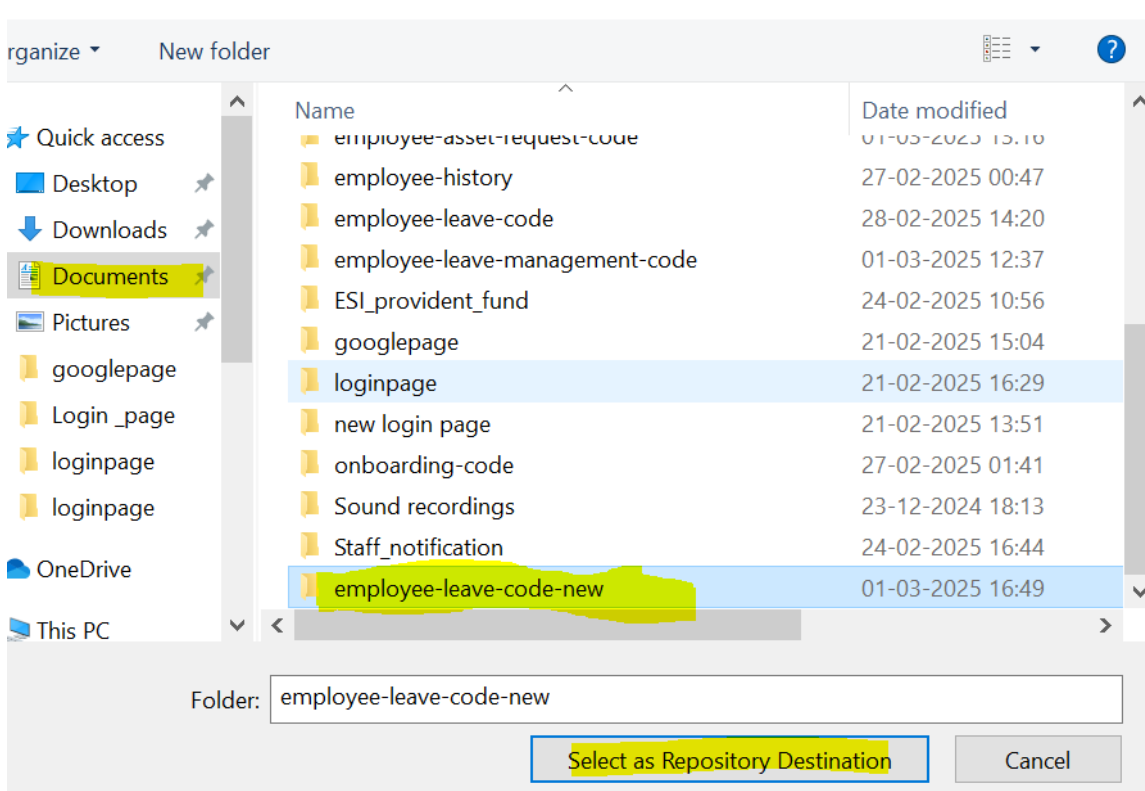
## Step-3:

Open Visual Studio Code and take new window from the file.

Click on > clone on Repository and on Top Search bar inside paste the URL



Create a folder and select same the folder

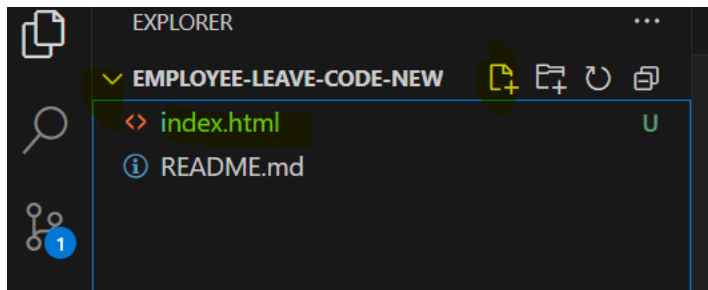


Now click on open

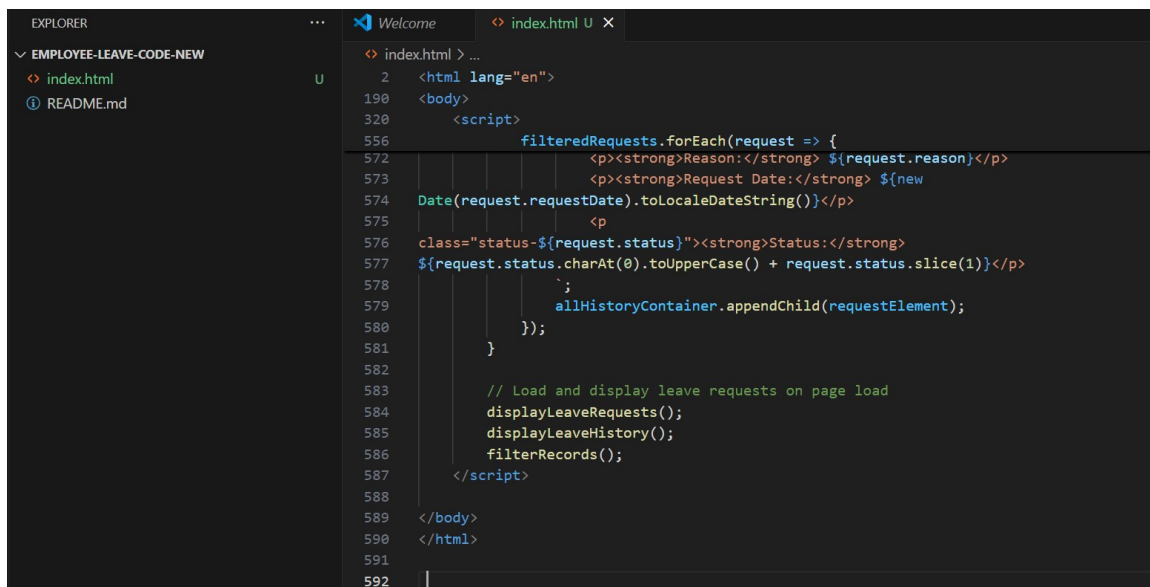
Note: if it display TRUST means select Trust

By clicking on First Plus creates a newfile and also it displays new file

For Example create file name as (index.html)click on Enter



Copy the CODE and paste it in index.html and cntrl+s to save



Go back to Folder and check the web is ready to view or not

## Employee Leave Request Portal

[New Request](#)[Leave History](#)

Employee ID:

Leave Type:

Annual Leave

Start Date:

dd-mm-yyyy

End Date:

dd-mm-yyyy

Reason:

Submit Request

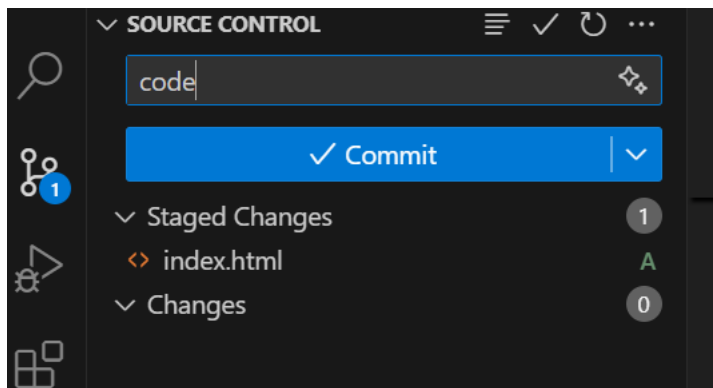
Once its available webpage - Test that by entering details wrong and right details

**Note:** Check the webpage did you find any errors alignment is missing how the page looks like and while entering details how many times its taking also share your ideas like we need change in that to page looks more better everything from A-z

Then from Visual studio code push code GitHub account

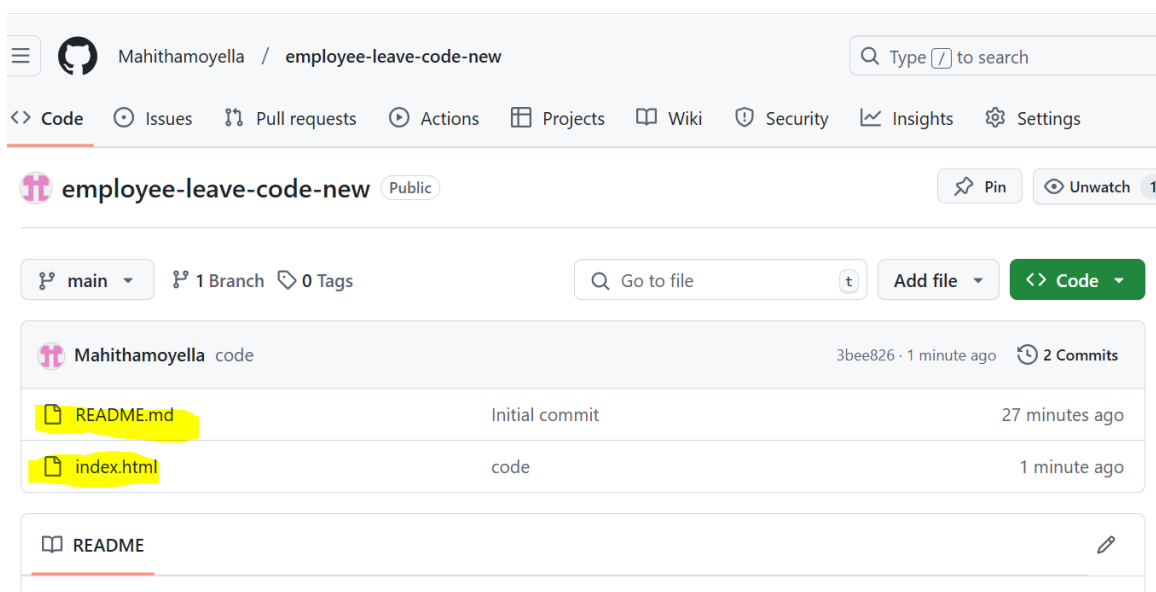
By clicking on **+** (**stage**)

Give a message and click on **> commit**

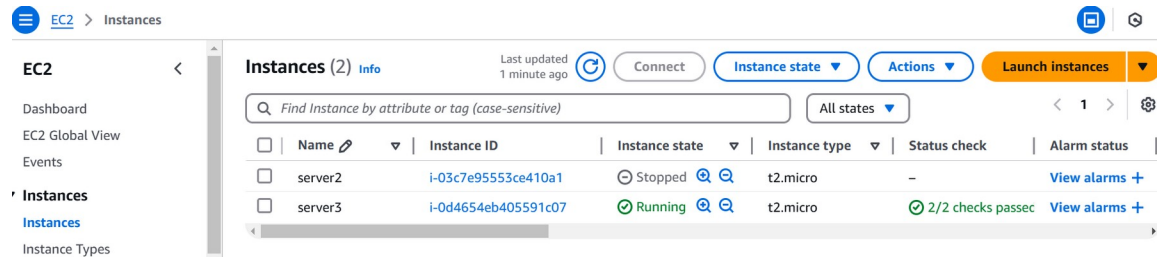


Click on **sync changes** and click on **open**

Now Go back to GITHUB Account check the files is available or not



Go to AWS account and click **sign in to the console**. After sign in, click on **EC2** and then click on **instances** . Now click on **launch instances**.



## Give the server name

### Launch an instance Info

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

**Name and tags** Info

**Name**

server 4

[Add additional tags](#)

## Now create a keypair and select ubuntu

**Create key pair**

**Key pair name**  
Key pairs allow you to connect to your instance securely.

keypair1

The name can include up to 255 ASCII characters. It can't include leading or trailing spaces.

**Key pair type**

☒ **RSA**  
RSA encrypted private and public key pair

☐ **ED25519**  
ED25519 encrypted private and public key pair

**Private key file format**

☒ **.pem**  
For use with OpenSSH

☐ **.ppk**  
For use with PuTTY

[Cancel](#) [Create key pair](#)

## Setup Docker

- Setup t2.large instance with 20 GB Volume
- Add Following Rules in Security Groups 32768-61000, and 8080-9090, 80, 22
- We can install docker on any operating system whether it is Linux, Windows or MAC.

Go To Network settings click on **Edit**

▼ **Network settings** [Info](#)

Network | [Info](#)

vpc-03ca897f68e9aba28

Subnet | [Info](#)

Edit

Now click **add security groups rule**

Inbound Security Group Rules

▼ Security group rule 1 (TCP, 22, 0.0.0.0/0)

Type | [Info](#)

ssh ▼

Protocol | [Info](#)

TCP

Port range | [Info](#)

22

Source type | [Info](#)

Anywhere ▼

Source | [Info](#)

Q Add CIDR, prefix list or security

0.0.0.0/0 ✕

Description - *optional* | [Info](#)

e.g. SSH for admin desktop

Remove

⚠ Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

✕

Add security group rule

Now add this ports to run

In port range - 8080-9090

Source - 0.0.0.0/0

Port range - 32768-61000

Source - 0.0.0.0/0

Port range- 80

Source - 0.0.0.0/0

Type

Info

Custom TCP

Protocol

Info

TCP

Port range

Info

8080-9090

Source type

Info

Custom

Source

Info

Q Add CIDR, prefix list or security

0.0.0.0/0 X

Description - optional

Info

e.g. SSH for admin desktop

▼ Security group rule 3 (TCP, 32768-61000, 0.0.0.0/0)

Remove

Type

Info

Custom TCP

Protocol

Info

TCP

Port range

Info

32768-61000

Source type

Info

Custom

Source

Info

Q Add CIDR, prefix list or security

0.0.0.0/0 X

Description - optional

Info

e.g. SSH for admin desktop

▼ Security group rule 4 (TCP, 80, 0.0.0.0/0)

Remove

Type

Info

Custom TCP

Protocol

Info

TCP

Port range

Info

80

Source type

Info

Custom

Source

Info

Q Add CIDR, prefix list or security

0.0.0.0/0 X

Description - optional

Info

e.g. SSH for admin desktop

▼ Security group rule 5 (All, All, 0.0.0.0/0)

Remove

Type

Info

All traffic

Protocol

Info

All

Port range

Info

All

Source type

Info

Custom

Source

Info

Q Add CIDR, prefix list or security

0.0.0.0/0 X

Description - optional

Info

e.g. SSH for admin desktop

Now give **15 GB** or **20 GB**

Click in **launch instance**

▼ Configure storage

Info

Advanced

1x

15

GIB

gp3

Root volume, 3000 IOPS, Not encrypted

Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage

X

Add new volume

Virtual server type (instance type)

t2.micro

Firewall (security group)

New security group

Storage (volumes)

Cancel

Launch instance

Preview code



Now select the server for public IP address

**Instances (1/3)** [Info](#) Last updated 2 minutes ago [Connect](#) [Instance state](#) [Actions](#) [Launch instances](#)

Find Instance by attribute or tag (case-sensitive) All states

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status
<input type="checkbox"/>	server2	i-03c7e95553ce410a1	Stopped	t2.micro	-	<a href="#">View alarms +</a>
<input type="checkbox"/>	server3	i-0d4654eb405591c07	Stopped	t2.micro	-	<a href="#">View alarms +</a>
<input checked="" type="checkbox"/>	server-4	i-0689ce549e494b322	Running	t2.micro	Initializing	<a href="#">View alarms +</a>

---

**i-0689ce549e494b322 (server 4)**

▼ Instance summary [Info](#)

Instance ID i-0689ce549e494b322	Public IPv4 address 54.86.77.122   <a href="#">open address</a>	Private IPv4 addresses 172.31.87.226
------------------------------------	--	---

Now go to **GitBash** , enter the following commands to login into the server  
**commands:**

**cd downloads/** <in which folder the key pair stored>

**ls**

**ssh -i keypair.pem ubuntu@<ip-address>**

then enter **yes**

```

LENOVO L480@DESKTOP-E238M55 MINGW64 ~
$ cd downloads/

LENOVO L480@DESKTOP-E238M55 MINGW64 ~/downloads
$ ls
'AWS DevOps Session.pdf'          UltraViewer_setup_6.6.113_en.exe*
'AWS- DevOps (1).pdf'            VSCodeUserSetup-x64-1.97.2.exe*
'AWS- DevOps.pdf'                _Employee-leave-code.pdf
'ChromeSetup.exe*'               'attendance monthly report (1).pdf'
'DOCKER (1).pdf'                 'attendance monthly report.pdf'
'DOCKER.pdf'                     desktop.ini
'Devops.pdf'                     'employee History code.pdf'
'Docker with commands (1).pdf'    employee-Leave-management-code.pdf
'Docker with commands.pdf'        employee-asset-request-code.pdf
'EMPLOYEE ASSET code.pdf'         get-docker.sh*
'ESI_Provident Fund code.pdf'     invite.ics
'Git-2.48.1-64-bit.exe*'          keypair.pem
'GitHub & VS code.txt'            'login- code (1).pdf'
'IMG-20241114-WA0012.jpg'         'login- code.pdf'
'Image_20250216_201338_813.jpeg'  mykey1.pem
'Image_20250216_201338_865.jpeg'  npp.8.7.7.Installer.x64.exe*
'Image_20250216_201338_924.jpeg'  'onboarding code.pdf'
'LSBSetup.exe*'                  putty-64bit-0.83-installer.msi
'Staff Notification code.pdf'      usbcdkfw3704_2.exe*
'TeamViewer_Setup_x64.exe*'

LENOVO L480@DESKTOP-E238M55 MINGW64 ~/downloads
$ ssh -i keypair.pem ubuntu@18.212.173.49
The authenticity of host '18.212.173.49 (18.212.173.49)' can't be established.
ED25519 key fingerprint is SHA256:c7mqtpZlFb1bN5GWxJawmri98IHwrghaBDLxTVRu6hU.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes

```

Check the Docker First

**Command: docker**

Update your system

**Command: sudo apt update -y**

```

ubuntu@ip-172-31-95-166:~$ docker
Command 'docker' not found, but can be installed with:
sudo snap install docker          # version 27.5.1, or
sudo snap install docker          # version 27.2.0
sudo apt install docker.io         # version 24.0.7-0ubuntu4.1
sudo apt install podman-docker     # version 4.9.3+ds1-1ubuntu0.2
See 'snap info <snapname>' for additional versions.
ubuntu@ip-172-31-95-166:~$ sudo apt update -y
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease

```

Now install Docker

**Command: curl -fsSL https://get.docker.com -o get-docker.sh && sh get-docker.sh**

```
134 packages can be upgraded. Run 'apt list --upgradable' to see them.
ubuntu@ip-172-31-95-166:~$ curl -fsSL https://get.docker.com -o get-docker.sh && sh get-docker.sh
# Executing docker install script, commit: 4c94a56999e10efcf48c5b8e3f6afea464f9108e
```

Now check docker

**Command: docker**

When installing Docker, you can use the command "**curl fsSL https://get.docker.com -o get-docker.sh**" to download the installation script, followed by executing the script using the "**sh**" command. This process is quick and easy, and once Docker is installed, you can enjoy its many benefits, such as the ability to easily create and run containers for your applications.

**Command: docker info**

**Command: docker image ls**

**Command: docker container ls**

```
server:
ERROR: permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Get "http://%2Fvar%2Frun%2Fdocker.sock/v1.48/info": dial unix
/var/run/docker.sock: connect: permission denied
errors pretty printing info
ubuntu@ip-172-31-95-166:~$ docker image ls
permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Head "http://%2Fvar%2Frun%2Fdocker.sock/_ping": dial unix /var/run/do
cker.sock: connect: permission denied
ubuntu@ip-172-31-95-166:~$ docker container ls
permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Get "http://%2Fvar%2Frun%2Fdocker.sock/v1.48/containers/json": dial u
```

To run the docker related work, without sudo access, add the current user to the docker group.

**Command: sudo usermod -aG <group> <user-name>**

Note: group = **docker**

User-name = **ubuntu**

**Command: sudo usermod -aG docker ubuntu**

**Command: logout**

The command above allows you to add a user to a specific group. To be more precise, the "**sudo usermod**" command is used to modify a user's account details, with "**-aG**" specifying the group to which the user will be added. In the example provided, the group is "**docker**" and the user is "**ubuntu**". Once again ssh back into docker host and start executing docker commands.

**Command: docker info**

**Command: docker image ls**

## Command: docker container ls

● **docker info:** This will display system-wide information about the Docker installation on your machine, including the version number, the number of containers and images, and the status of your containers.

```
ubuntu@ip-172-31-95-166:~$ docker info
Client: Docker Engine - Community
Version: 28.0.1
Context: default
Debug Mode: false
Plugins:
  buildx: Docker Buildx (Docker Inc.)
    Version: v0.21.1
    Path: /usr/libexec/docker/cli-plugins/docker-buildx
  compose: Docker Compose (Docker Inc.)
    Version: v2.33.1
    Path: /usr/libexec/docker/cli-plugins/docker-compose
Server:
Containers: 0
  Running: 0
  Paused: 0
  Stopped: 0
Images: 0
Server Version: 28.0.1
Storage Driver: overlay2
  Backing Filesystem: extfs
  Supports d_type: true
  Using metacopy: false
  Native Overlay Diff: true
  userxattr: false
Logging Driver: json-file
Cgroup Driver: systemd
Cgroup Version: 2
Plugins:
  Volume: local
  Network: bridge host ipvlan macvlan null overlay
  Log: awslogs fluentd gcplogs gelf journald json-file local splunk syslog
Swarm: inactive
Runtimes: io.containerd.runc.v2 runc
Default Runtime: runc
Init Binary: docker-init
containerd version: bcc810d6b9066471b0b6fa75f557a15a1cbf31bb
runc version: v1.2.4-0-g6c52b3f
init version: de40ad0
Security Options:
  apparmor
  seccomp
    Profile: builtin
  cgroupns
Kernel Version: 6.8.0-1021-aws
```

● **docker image ls:** This will show a list of all the images that are currently stored on your machine. You can use this command to find the image you need for your application.

● **docker container ls:** This will display a list of all the containers that are currently running on your machine. You can use this command to check the status of your containers and to manage them as needed.

# Setup Containers

## Setup Nginx Container

To get started with using Nginx in a Docker container, you can begin by pulling the Nginx image from the Docker Hub registry using the command:

Check the image is available or not

**Command: docker image ls**

If image is not available pull the image now

**Command: docker pull nginx**

Now check the image

**Command: docker image ls**

```
ubuntu@ip-172-31-95-166:~$ docker image ls
REPOSITORY TAG IMAGE ID CREATED SIZE
ubuntu@ip-172-31-95-166:~$ docker container ls
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
ubuntu@ip-172-31-95-166:~$ docker pull nginx
Using default tag: latest
latest: Pulling from library/nginx
7cf63256a31a: Pull complete
bf9acace214a: Pull complete
513c3649bb14: Pull complete
d014f92d532d: Pull complete
9dd21ad5a4a6: Pull complete
943ea0f0c2e4: Pull complete
103f50cb3e9f: Pull complete
Digest: sha256:9d6b58feebd2dbd3c56ab5853333d627cc6e281011cfd6050fa4bcf2072c9496
Status: Downloaded newer image for nginx:latest
docker.io/library/nginx:latest
ubuntu@ip-172-31-95-166:~$ docker image ls
REPOSITORY TAG IMAGE ID CREATED SIZE
nginx latest b52e0b094bc0 3 weeks ago 192MB
ubuntu@ip-172-31-95-166:~$ |
```

## Creating a volume

Creating volume with name

**Command: docker volume create employee-asset-volume**

```
ubuntu@ip-172-31-95-166:~$ docker volume create employee-asset-volume
employee-asset-volume
ubuntu@ip-172-31-95-166:~$ docker volume ls
DRIVER VOLUME NAME
local employee-asset-volume
ubuntu@ip-172-31-95-166:~$ |
```

Inspect the employee-asset-volume to check the path of the volume

**Command: docker volume inspect employee-asset-volume**

```
ubuntu@ip-172-31-95-166:~$ docker volume inspect employee-asset-volume
[
  {
    "CreatedAt": "2025-03-02T10:56:46Z",
    "Driver": "local",
    "Labels": null,
    "Mountpoint": "/var/lib/docker/volumes/employee-asset-volume/_data",
    "Name": "employee-asset-volume",
    "Options": null,
    "Scope": "local"
  }
]
```

Check the path, inside the volume do we have any files inside path

**Command: sudo ls /var/lib/docker/volumes/employee-asset-volume/\_data**

```
ubuntu@ip-172-31-95-166:~$ sudo ls /var/lib/docker/volumes/employee-asset-volume/_data
ubuntu@ip-172-31-95-166:~$ |
```

Now create a container using volume = **employee-asset-volume**

**Command: docker container run -dt --name employee22 -p 8055:80 -v employee-asset-volume:/usr/share/nginx/html nginx**

Now check the volume path is there any files by using below command

**Command: sudo cat /var/lib/docker/volumes/employee-asset-volume/\_data/index.html**

```
ubuntu@ip-172-31-95-166:~$ docker container run -dt --name employee22 -p 8055:80 -v employee-asset-volume:/usr/share/nginx/html nginx
66fb1d38be810818067c4c74115dc00316f4af8454f8c2fe36308524566899b8
ubuntu@ip-172-31-95-166:~$ sudo ls /var/lib/docker/volumes/employee-asset-volume/_data
50x.html index.html
ubuntu@ip-172-31-95-166:~$ sudo cat /var/lib/docker/volumes/employee-asset-volume/_data/html
cat: /var/lib/docker/volumes/employee-asset-volume/_data/html: No such file or directory
ubuntu@ip-172-31-95-166:~$ sudo cat /var/lib/docker/volumes/employee-asset-volume/_data/index.html
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
html { color-scheme: light dark; }
body { width: 35em; margin: 0 auto;
font-family: Tahoma, Verdana, Arial, sans-serif; }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>

<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>

<p><em>Thank you for using nginx.</em></p>
</body>
</html>
```

Now go inside the container by using below command:

**Command: docker container exec -it employee22 bash**

**Command: ls /usr/share/nginx/html**

```
ubuntu@ip-172-31-95-166:~$ docker container exec -it employee22 bash
root@66fb1d38be81:/# ls /usr/share/nginx/html
50x.html  index.html
root@66fb1d38be81:/# rm -rf /usr/share/nginx/html/*
root@66fb1d38be81:/# ls /usr/share/nginx/html
root@66fb1d38be81:/# |
```

Update the system with below command:

**Command: apt update -y**

```
root@66fb1d38be81:/# apt update -y
Get:1 http://deb.debian.org/debian bookworm InRelease [151 kB]
Get:2 http://deb.debian.org/debian bookworm-updates InRelease [55.4 kB]
Get:3 http://deb.debian.org/debian-security bookworm-security InRelease [48.0 kB]
Get:4 http://deb.debian.org/debian bookworm/main amd64 Packages [8792 kB]
Get:5 http://deb.debian.org/debian bookworm-updates/main amd64 Packages [13.5 kB]
Get:6 http://deb.debian.org/debian-security bookworm-security/main amd64 Packages [246 kB]
Fetched 9306 kB in 1s (6281 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
All packages are up to date.
```

To Install git use below command:

**Command: apt install -y git**

```
root@66fb1d38be81:/# apt install -y git
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  git-man less libcbor0.8 libcurl3-gnutls liberror-perl libfido2-1 libgdbm-compat4 libgdbm6 libperl5.36 libperl5.38
  perl-modules-5.36 xauth
Suggested packages:
  git-daemon-run | git-daemon-sysvinit git-doc git-email git-gui gitk gitweb git-cvs git-mediawiki git-svn
  ssh-askpass ed diffutils-doc perl-doc libterm-readline-gnu-perl | libterm-readline-perl-perl make libtag
The following NEW packages will be installed:
  git git-man less libcbor0.8 libcurl3-gnutls liberror-perl libfido2-1 libgdbm-compat4 libgdbm6 libperl5.36 libperl5.38
  perl-modules-5.36 xauth
```

Now list the path with below command:

**Command: ls /usr/share/nginx/html**

Remove the files which are inside nginx path with below command:

**Command: rm -rf /usr/share/nginx/html/\***

Now list the path and check do you find any files after deleting ?

**Command: ls /usr/share/nginx/html**

Now clone the Application URL code files from github with below command:

**Command: git clone https://github.com/Mahithamoyella/employee-leave-code-new.git /usr/share/nginx/html/**

```
root@66fb1d38be81:/# git clone https://github.com/Mahithamoyella/employee-leave-code-new.git /usr/share/nginx/html/
Cloning into '/usr/share/nginx/html'...
remote: Enumerating objects: 6, done.
remote: Counting objects: 100% (6/6), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 6 (delta 0), reused 3 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (6/6), 4.89 KiB | 4.89 MiB/s, done.
```

Now exit from container

**Command: exit**

```
root@66fb1d38be81:/# exit
exit
ubuntu@ip-172-31-95-166:~$ |
```

Now browse the IP address with port number 8055

## Employee Leave Request Portal

**New Request**

Leave History

Employee ID:

Leave Type:

Annual Leave

Start Date:

dd-mm-yyyy

End Date:

dd-mm-yyyy

Reason:

Now you are back to host, and lets check the path of the volume with below command

**Command: sudo ls /var/lib/docker/volumes/employee-asset-volume/\_data**

Note: Here you can see the URL code files

```
ubuntu@ip-172-31-95-166:~$ sudo ls /var/lib/docker/volumes/employee-asset-volume/_data
README.md  index.html
ubuntu@ip-172-31-95-166:~$ |
```



Now check the docker containers with ls command

### Command: docker container ls

```
ubuntu@ip-172-31-95-166:~$ docker container ls
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS        PORTS                               NAMES
d6af7daa86e5   shreyas978/ats-leave-manage-code   "/docker-entrypoint..." 15 minutes ago Up 15 minutes 0.0.0.0:8094->80/tcp, [::]:8094->80/tcp employee14
39ec3459eadd   shreyas978/ats-emp-asset           "/docker-entrypoint..." 19 minutes ago Up 19 minutes 0.0.0.0:8093->80/tcp, [::]:8093->80/tcp employee13
5198beb91idd   daddlasahith/app-leave             "/docker-entrypoint..." 32 minutes ago Up 32 minutes 0.0.0.0:8092->80/tcp, [::]:8092->80/tcp employee12
8983d367cdf5   daddlasahith/employee-asset        "/docker-entrypoint..." 38 minutes ago Up 38 minutes 0.0.0.0:8091->80/tcp, [::]:8091->80/tcp employee11
56fb1d38be81   nginx                               "/docker-entrypoint..." 19 hours ago   Up 19 hours   0.0.0.0:8055->80/tcp, [::]:8055->80/tcp employee22
ubuntu@ip-172-31-95-166:~$
```

Now delete the container which you have created **employee22** container

```
ubuntu@ip-172-31-95-166:~$ docker container rm -f employee22
employee22
ubuntu@ip-172-31-95-166:~$ docker container ls
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS        PORTS                               NAMES
d6af7daa86e5   shreyas978/ats-leave-manage-code   "/docker-entrypoint..." 23 minutes ago Up 23 minutes 0.0.0.0:8094->80/tcp, [::]:8094->80/tcp employee14
39ec3459eadd   shreyas978/ats-emp-asset           "/docker-entrypoint..." 27 minutes ago Up 27 minutes 0.0.0.0:8093->80/tcp, [::]:8093->80/tcp employee13
5198beb91idd   daddlasahith/app-leave             "/docker-entrypoint..." 40 minutes ago Up 40 minutes 0.0.0.0:8092->80/tcp, [::]:8092->80/tcp employee12
8983d367cdf5   daddlasahith/employee-asset        "/docker-entrypoint..." 46 minutes ago Up 46 minutes 0.0.0.0:8091->80/tcp, [::]:8091->80/tcp employee11
ubuntu@ip-172-31-95-166:~$
```

After deleting the containers also still you created volume are available to check use below command

### Command: docker volumes ls

Now inspect the volume to take volume path

### Command: docker volume inspect employee-asset-volume

Now list the path of the volume and check Application code files are available or not.

### Command: sudo ls /var/lib/docker/volumes/employee-asset-volume/\_data

### Output: README.md index.html

```
ubuntu@ip-172-31-95-166:~$ docker volume ls
DRIVER      VOLUME NAME
local      employee-asset-volume
ubuntu@ip-172-31-95-166:~$ docker volume inspect employee-asset-volume
[
  {
    "CreatedAt": "2025-03-02T10:56:46Z",
    "Driver": "local",
    "Labels": null,
    "Mountpoint": "/var/lib/docker/volumes/employee-asset-volume/_data",
    "Name": "employee-asset-volume",
    "Options": null,
    "Scope": "local"
  }
]
ubuntu@ip-172-31-95-166:~$ sudo ls /var/lib/docker/volumes/employee-asset-volume/_data
README.md  index.html
ubuntu@ip-172-31-95-166:~$
```

**Note** : By using volume **employee-asset-volume** you can create multiple containers to load the same application.

**For Example** : create a one more container with same volume ( employee-asset-

volume) with port no and go to browser and check the application is loading or not.

**Command: docker container run -dt --name employee20 -p 8083:80 -v employee-asset-volume:/usr/share/nginx/html nginx**

```
ubuntu@ip-172-31-95-166:~$ docker container run -dt --name employee20 -p 8083:80 -v employee-asset-volume:/usr/share/nginx/html nginx
7247dbbb49ccf3c90e9e7b6a02702b6405d1b16141c4ccb35ea74cd1c43f2127
ubuntu@ip-172-31-95-166:~$
```

Now browse the IP address with port no 8083

→ ↻ ⚠ Not secure 18.212.173.49:8083

---

## Employee Leave Request Portal

**New Request**   Leave History

---

Employee ID:

Leave Type:

Start Date:

End Date:

Reason:

## HOST VOLUMES

On Host clone the Application URL From github

**Command: git clone https://github.com/Mahithamoyella/employee-leave-code-new.git**

```

ubuntu@ip-172-31-95-166:~$ git clone https://github.com/Mahithamoyella/employee-leave-code-new.git
Cloning into 'employee-leave-code-new'...
remote: Enumerating objects: 6, done.
remote: Counting objects: 100% (6/6), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 6 (delta 0), reused 3 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (6/6), 4.89 KiB | 1.63 MiB/s, done.
ubuntu@ip-172-31-95-166:~$ ls
employee-leave-code-new  get-docker.sh
ubuntu@ip-172-31-95-166:~$ cd employee-leave-code-new
ubuntu@ip-172-31-95-166:~/employee-leave-code-new$ ls
README.md  index.html
ubuntu@ip-172-31-95-166:~/employee-leave-code-new$ |

```

Now create a container on host level with nginx path

**Command:** `docker container run -dt --name employee25 -p 8067:80 -v ~/employee-leave-code-new:/usr/share/nginx/html nginx`

```

ubuntu@ip-172-31-95-166:~/employee-leave-code-new$ docker container run -dt --name employee25 -p 8067:80 -v ~/employee-leave-code-new:/usr/share/nginx/html nginx
79d2ba47f3afc17736fedccc12cf49a2c5123d33608eb099cf27e9baeb89004a
ubuntu@ip-172-31-95-166:~/employee-leave-code-new$ |

```

On port 8067 page is available

🔄 ⚠ Not secure 18.212.173.49:8067

# Employee Leave Request Portal

New Request

Leave History

Employee ID:

Leave Type:

Annual Leave

Start Date:

dd-mm-yyyy

End Date:

dd-mm-yyyy

Reason:

This is how host volumes will work.

## CUSTOMIZED IMAGE

**Command: ls**

Output: **employee-leave-code-new get-docker.sh**

**Command: cd employee-leave-code-new**

**Command: ls**

Output: **README.md index.html**

```
ubuntu@ip-172-31-95-166:~$ ls
employee-leave-code-new  get-docker.sh
ubuntu@ip-172-31-95-166:~$ cd employee-leave-code-new
ubuntu@ip-172-31-95-166:~/employee-leave-code-new$ ls
README.md  index.html
```

Now create docker file with vi mode using below command:

**Command: vi dockerfile**

```
ubuntu@ip-172-31-95-166:~/employee-leave-code-new$ vi dockerfile
ubuntu@ip-172-31-95-166:~/employee-leave-code-new$
```

From inside the dockerfile , we press i for insert mode and then enter following commands

**command: FROM nginx**

**command: COPY . /usr/share/nginx/html**

Now press **Esc** button and give **:wq** to save the file.

```
FROM nginx
COPY . /usr/share/nginx/html

:wq
```

**Command:** ls

**Output:** README.md dockerfile index.html

After creating the dockerfile run this below command to build your own image

**Command:** docker build -t mahithamoyella/employee-leave-code-new .

**Note:** create a docker hub account

- Take username of docker hub (example- mahithamoyella)

Now create a own image name you want, with giving any name

- (example - employee-leave-code-new)

```
ubuntu@ip-172-31-95-166:~/employee-leave-code-new$ ls
README.md  dockerfile  index.html
ubuntu@ip-172-31-95-166:~/employee-leave-code-new$ docker build -t mahithamoyella/employee-leave-code-new .
[+] Building 0.5s (7/7) FINISHED
=> [internal] load build definition from dockerfile
=> => transferring dockerfile: 79B
=> [internal] load metadata for docker.io/library/nginx:latest
=> [internal] load .dockerignore
=> => transferring context: 28
=> [internal] load build context
=> => transferring context: 75.69kB
=> [1/2] FROM docker.io/library/nginx:latest
=> [2/2] COPY . /usr/share/nginx/html
=> exporting to image
=> => exporting layers
=> => writing image sha256:da70094f92219826773cc65b3b95dc2c4d22713dde07257413acbe90d1caa783
=> => naming to docker.io/mahithamoyella/employee-leave-code-new
ubuntu@ip-172-31-95-166:~/employee-leave-code-new$ |
```

Now create a new container with port no

**Command:** docker container run -dt --name employee50 -p 8097:80 mahithamoyella/employee-leave-code-new

```
ubuntu@ip-172-31-95-166:~/employee-leave-code-new$ docker container run -dt --name employee50 -p 8097:80 mahithamoyella/employee-leave-code-new
61690553380a7d72262568940a24bc91ae790252140458d845eb604c7d27ae1e
```

Browse and Check the URL:ipaddress:8097

# Employee Leave Request Portal

New Request

Leave History

Employee ID:

Leave Type:

Annual Leave

Start Date:

dd-mm-yyyy

End Date:

dd-mm-yyyy


Reason:

Now login into docker by using

**command: docker login**

```
ubuntu@ip-172-31-95-166:~/employee-leave-code-new$ docker login
```

USING WEB-BASED LOGIN

 Info → To sign in with credentials on the command line, use 'docker login -u <username>'

Your one-time device confirmation code is: **JKMV-NZDJ**

Press ENTER to open your browser or submit your device code here: <https://login.docker.com/activate>

Now browse that url and paste that code and click on continue then click confirm.



## Device Activation

Enter the code displayed on your device

Enter your one-time code\*

JKMV-NZDJ

Continue

Now push image to docker hub

**Command: `docker push mahithamoyella/employee-leave-code-new`**

```
ubuntu@ip-172-31-95-166:~/employee-leave-code-new$ docker push mahithamoyella/employee-leave-code-new
Using default tag: latest
The push refers to repository [docker.io/mahithamoyella/employee-leave-code-new]
f2649f3c83f0: Pushed
55e9644f21c3: Mounted from daddlasahith/app-leave
7d22e2347c12: Mounted from daddlasahith/app-leave
f6d5815f290e: Mounted from daddlasahith/app-leave
791f0a07985c: Mounted from daddlasahith/app-leave
cabea05c000e: Mounted from daddlasahith/app-leave
c68632c455ae: Mounted from daddlasahith/app-leave
5f1ee22ffb5e: Mounted from daddlasahith/app-leave
latest: digest: sha256:2d1e80fff6f9d21ba83d8973024e1cbe5faf38046062034f53c716514e23f5e6 size: 1987
ubuntu@ip-172-31-95-166:~/employee-leave-code-new$
```

Go back you Docker hub now check the image is available or not

dockerhub

Explore

Repositories

Organizations

Usage

Search Docker Hub

ctrl+K

mahithamoyella / Repositories / employee-leave-code-new / General

Using 0 of 1 private repositories.

mahithamoyella/employee-leave-code-new

Last pushed 3 minutes ago • Repository size: 68.9 MB

Add a description

Add a category

General

Tags

Image Management

BETA

Builds

Collaborators

Webhooks

Settings

Docker commands

To push a new tag to this repository:

```
docker push mahithamoyella/employee-leave-code-new:tagname
```

Public view

Tags

This repository contains 1 tag(s).

Tag	OS	Type	Pulled	Pushed
latest		Image	less than 1 day	3 minutes

Automated builds

Manually pushing images to Docker Hub? Connect your account to GitHub or Bitbucket to automatically build and tag new images whenever your code is updated, so you can focus your time on creating.

Available with Pro, Team and Business subscriptions. [Read more](#)

If we remove all containers and volumes also we get the files from this customized images in Docker Hub. This is how customized images will work.



