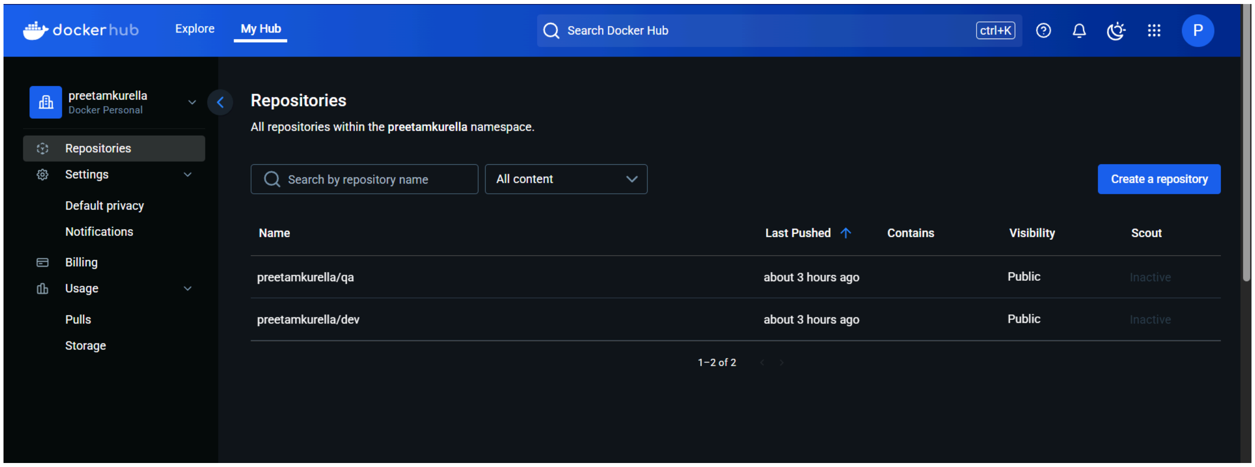
# REPORT

Mar 18, 2025 P. Kurella

# DOCKER 101.

## Docker Installation & usage:

- Go to docker hub.com and create an account for yourself.
- Now open the dashboard and leave it there.

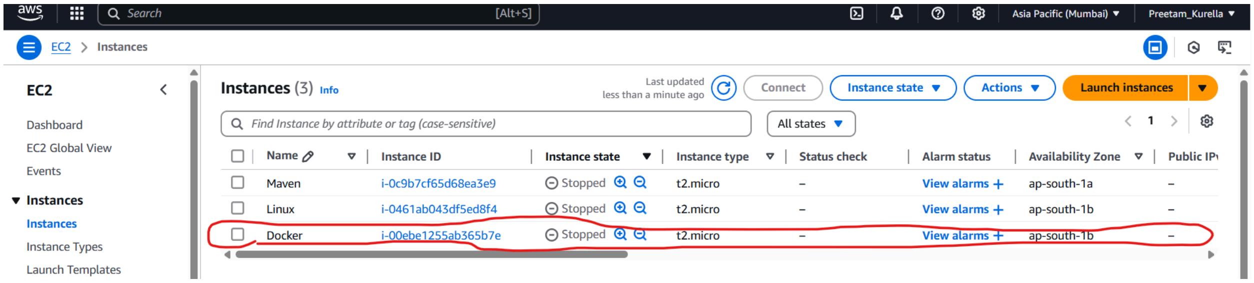


Repositories

All repositories within the **preetamkurella** namespace.

| Name | Last Pushed | Contains | Visibility | Scout |
|---|---|---|---|---|
| preetamkurella/qa | about 3 hours ago | | Public | Inactive |
| preetamkurella/dev | about 3 hours ago | | Public | Inactive |

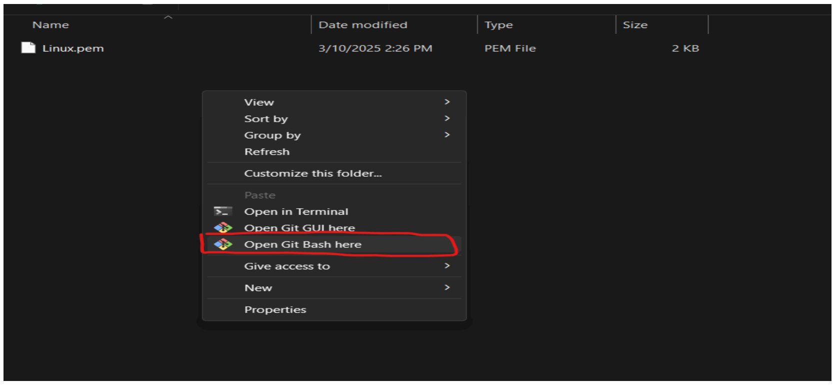
- Open AWS and create an AWS Linux docker instance and using the git bash we can now use the command line interface for docker.



Instances (3)

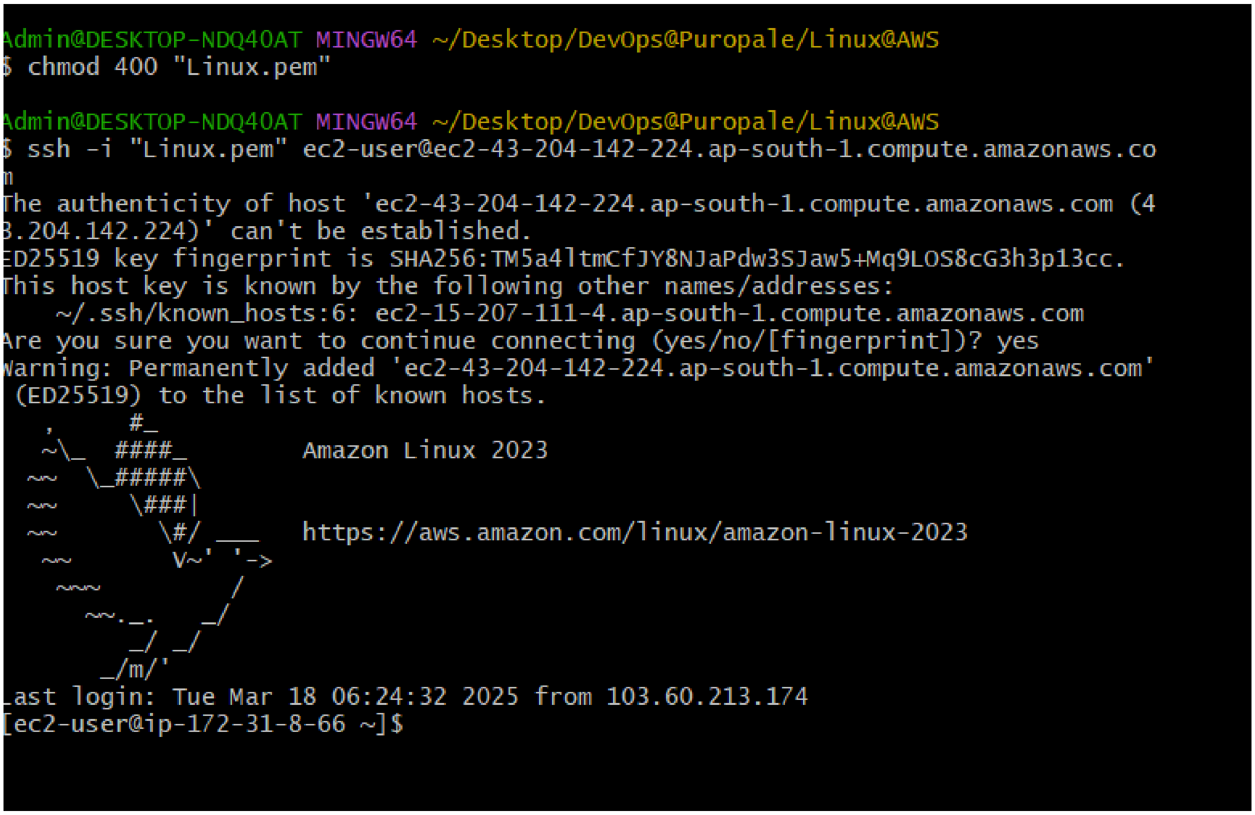
| Name | Instance ID | Instance state | Instance type | Status check | Alarm status | Availability Zone | Public IP |
|---|---|---|---|---|---|---|---|
| Maven | i-0c9b7cf65d68ea3e9 | Stopped | t2.micro | – | View alarms + | ap-south-1a | – |
| Linux | i-0461ab043df5ed8f4 | Stopped | t2.micro | – | View alarms + | ap-south-1b | – |
| Docker | i-00ebe1255ab365b7e | Stopped | t2.micro | – | View alarms + | ap-south-1b | – |

- Select the instance and run it in the pem file location in git-bash



| Name | Date modified | Type | Size |
|---|---|---|---|
| Linux.pem | 3/10/2025 2:26 PM | PEM File | 2 KB |

- Select the git bash option and the git bash command line will open.
- Copy the commands from the AWS instance to connect and run the two commands one after another.
    - `chmod 400 "Linux.pem"`
    - `ssh -i "Linux.pem" ec2-user@ec2-15-207-16-193.ap-south-1.compute.amazonaws.com`
- After entering the commands your interface will look like this.

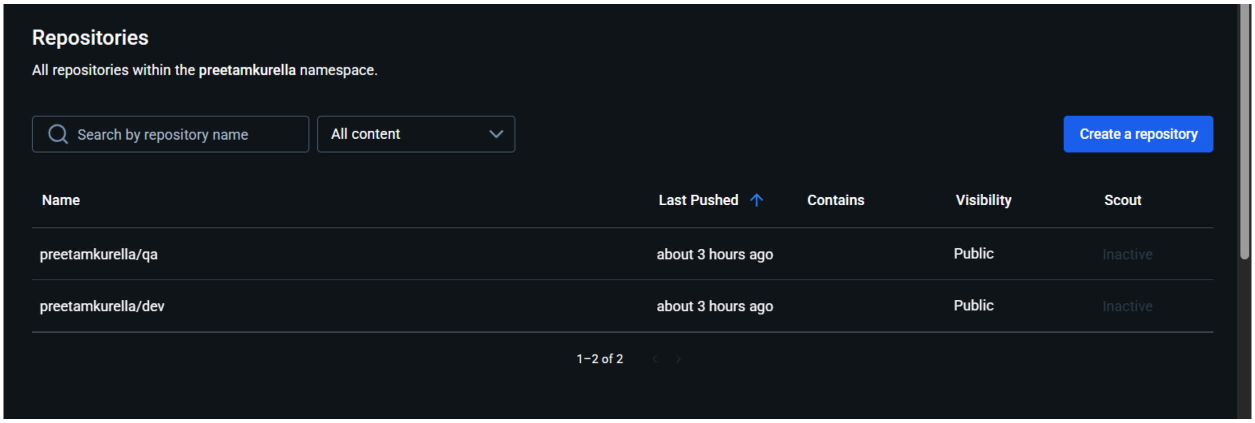


```
Admin@DESKTOP-NDQ40AT MINGW64 ~/Desktop/DevOps@Puropale/Linux@AWS
$ chmod 400 "Linux.pem"

Admin@DESKTOP-NDQ40AT MINGW64 ~/Desktop/DevOps@Puropale/Linux@AWS
$ ssh -i "Linux.pem" ec2-user@ec2-43-204-142-224.ap-south-1.compute.amazonaws.co
m
The authenticity of host 'ec2-43-204-142-224.ap-south-1.compute.amazonaws.com (4
3.204.142.224)' can't be established.
ED25519 key fingerprint is SHA256:TM5a4ltmCfJY8NJaPdw3SJaw5+Mq9LOS8cG3h3p13cc.
This host key is known by the following other names/addresses:
    ~/.ssh/known_hosts:6: ec2-15-207-111-4.ap-south-1.compute.amazonaws.com
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-43-204-142-224.ap-south-1.compute.amazonaws.com'
 (ED25519) to the list of known hosts.
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
Last login: Tue Mar 18 06:24:32 2025 from 103.60.213.174
[ec2-user@ip-172-31-8-66 ~]$
```

- Now check if docker is installed in your environment or not.
- To check this type the command `$ docker --version`.
- If the interface shows a docker version, then it indicates that a docker version is already installed on your device.
- If not then type this command `$ yum install -y docker`.
- Now docker is installed.
- Now before checking the installed docker configuration, first start the docker service using the command `$ service docker start`.
- Now use the command `$ ifconfig -a` to see the configuration of the installed docker application.
- Use the `$ docker info` command to know the info of the running instance, the type of os, home directory location, etc.

- The default home directory location for the docker application will be `/var/lib/docker.`
- To enter into the docker directory use the `$ cd /var/lib/docker` command.
- Now go to the docker hub account in your browser, create two repositories, and name them dev and qa.
- They should look like this once they have been created.



Repositories

All repositories within the **preetamkurella** namespace.

| Name | Last Pushed ↑ | Contains | Visibility | Scout |
|---|---|---|---|---|
| preetamkurella/qa | about 3 hours ago | | Public | Inactive |
| preetamkurella/dev | about 3 hours ago | | Public | Inactive |

1–2 of 2

- Now go to the bash and use the `$ ls` command to see the list of files in the docker file.
- It will show all the files in the current working directory.
- Use the `$ docker images` or the `$ docker image` ls command to check for images in the files.
- The parameters of the docker image will only show the headings of the structure as there are no images present in the current directory.
- The parameters will be:

```
REPO        TAG         IMAGE ID        CREATED         SIZE
```

- Use the `$ docker ps` command to check for containers in the current directory.
- The parameters will be:

```
CONTAINER ID    IMG   COMMAND    CREATED    STATUS    PORTS     NAMES
```

- Use the `$ docker ps -a` command to check all the running containers.
- To pull an image to the current directory we will pull an image from an online repository which is an official website.
- Let's pull the Nginx image from online.
- To do this use the `$ docker pull nginx` command.
- Use the `$ docker images` command to see the image ID and other attributes of the docker image we pulled.
- Use the `$ docker inspect <imageid>` command to inspect the details of the docker image.
- Now we will clone the image to our repository. But before that, we need to login to our docker account in the command line.
- To do that, use the `$ docker login` command to log into the account by entering the username and password.

- Now use the `$ docker run -itd --name <your name> -p cid:apn <imageID>` to create a docker container in your repository.
- Now use the `$ docker ps` command to check if the container is created.
- Now use the `$ docker tag nginx:latest <username/repo>:<filename>` to add the container to the staging area to your repository.
- Now use the `$ docker push <username/repo>:<filename>` to push the container to the docker repository.