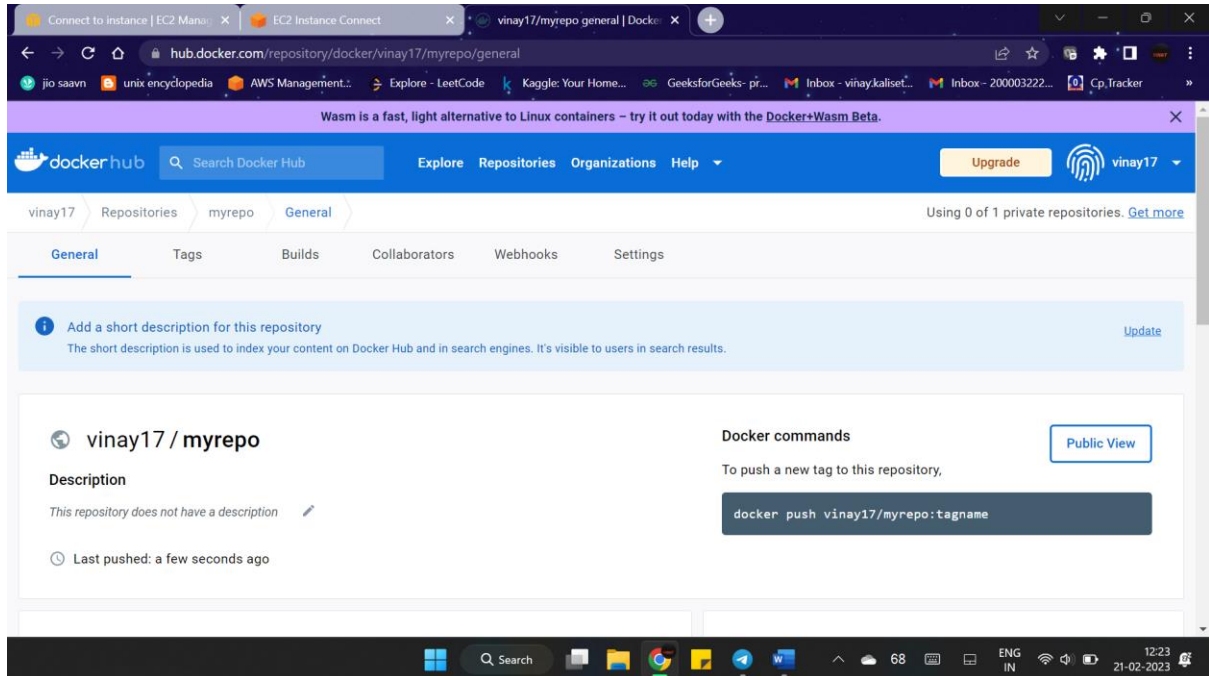
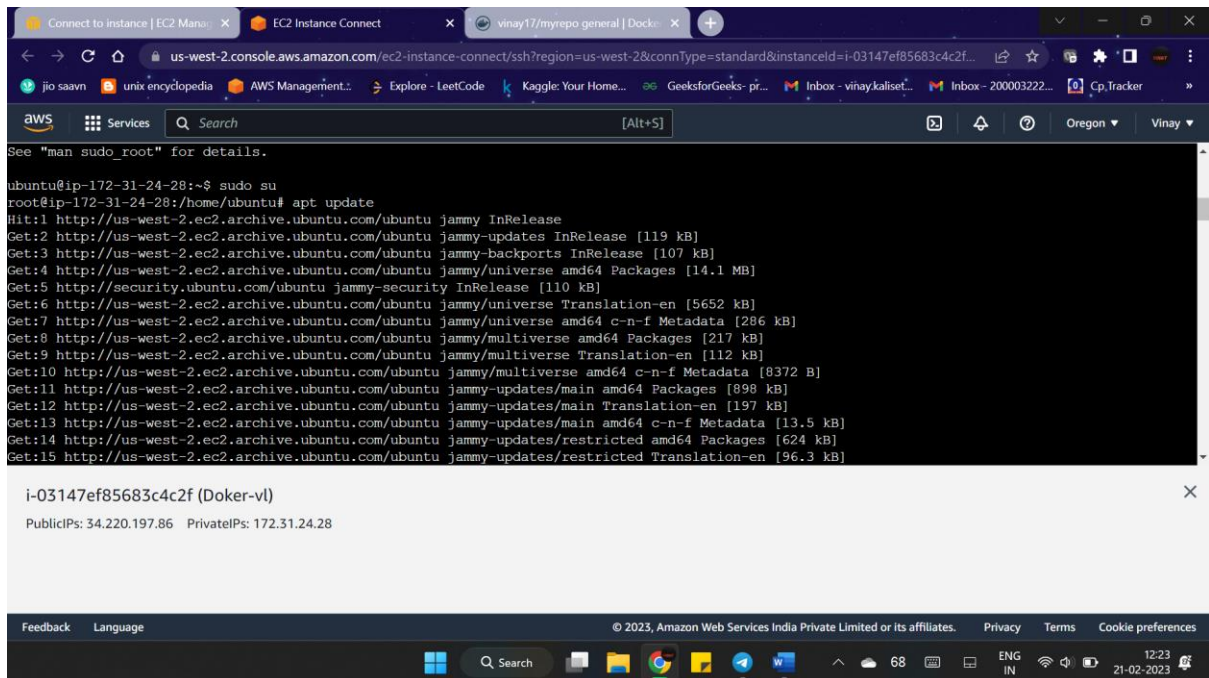


Pushing and Pulling Docker image

1. Create a repository in Docker Hub.



2. Connect your ec2 instance and update sudo and install Docker

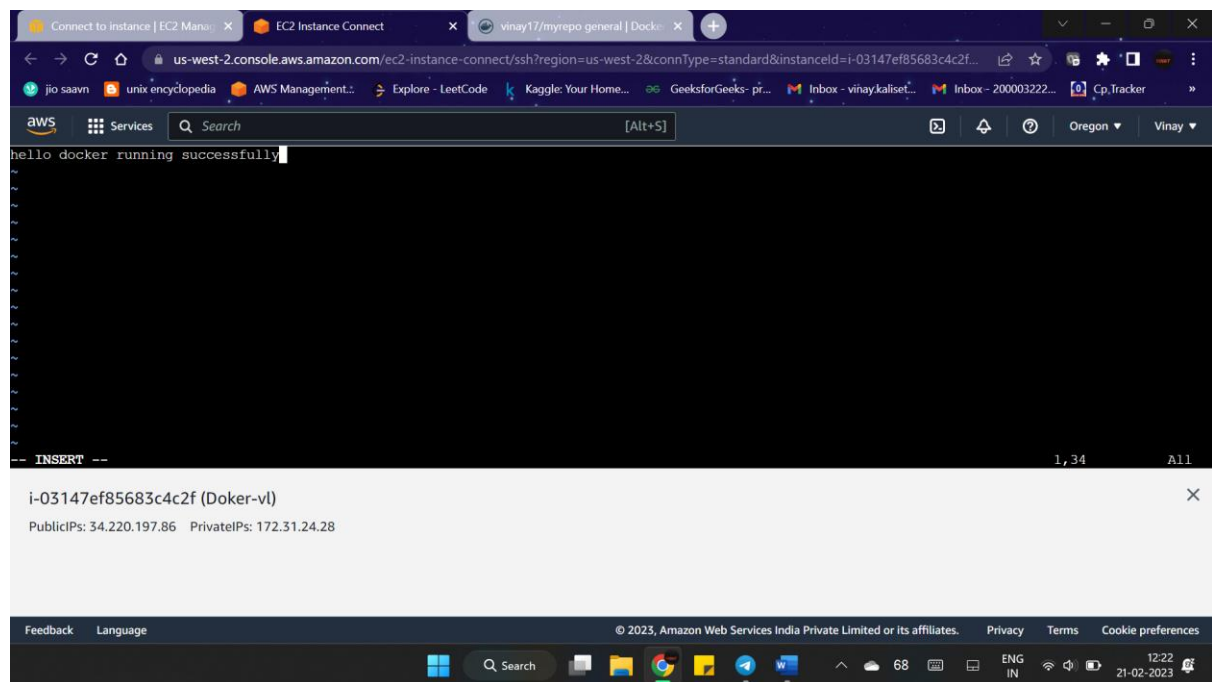


3. Create a directory using **mkdir myrepo**

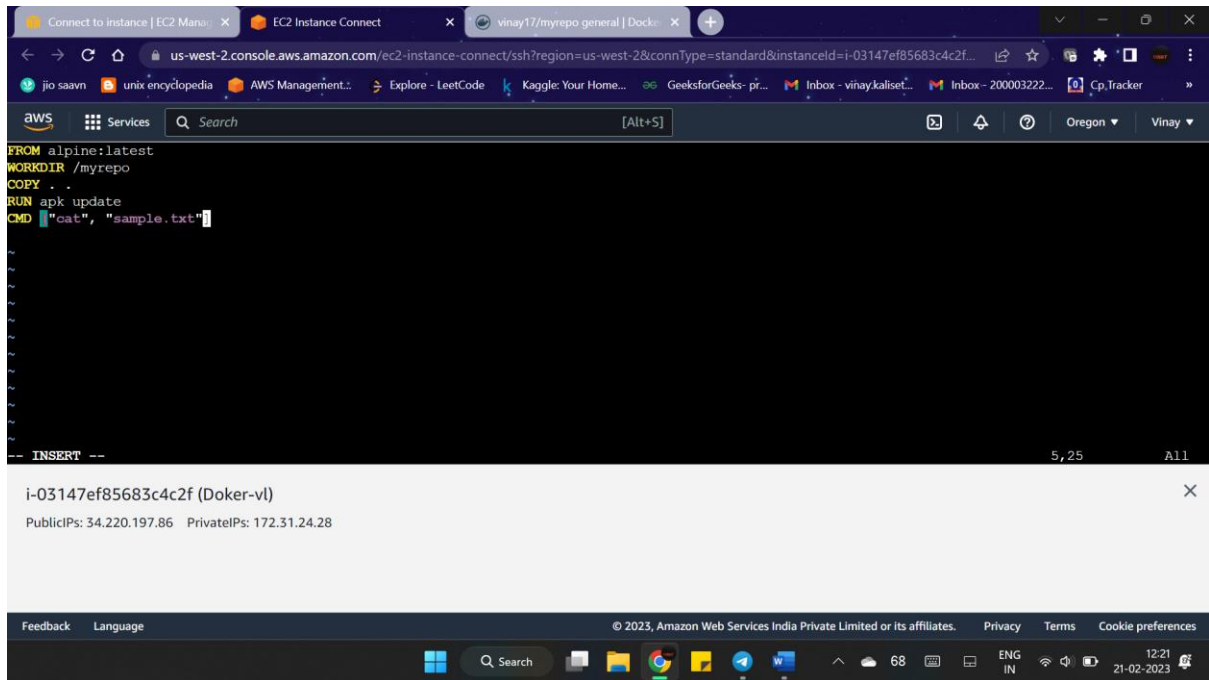
```
root@ip-172-31-24-28:/home/ubuntu# ls
myrepo
root@ip-172-31-24-28:/home/ubuntu# cd myrepo
root@ip-172-31-24-28:/home/ubuntu/myrepo# ls
Dockerfile  sample.txt
root@ip-172-31-24-28:/home/ubuntu/myrepo# vi Dockerfile
root@ip-172-31-24-28:/home/ubuntu/myrepo# vi sample.txt
root@ip-172-31-24-28:/home/ubuntu/myrepo#
```

4. Then create two files named Dockerfile and any sample .txt file

Sample.txt:



Dockerfile:

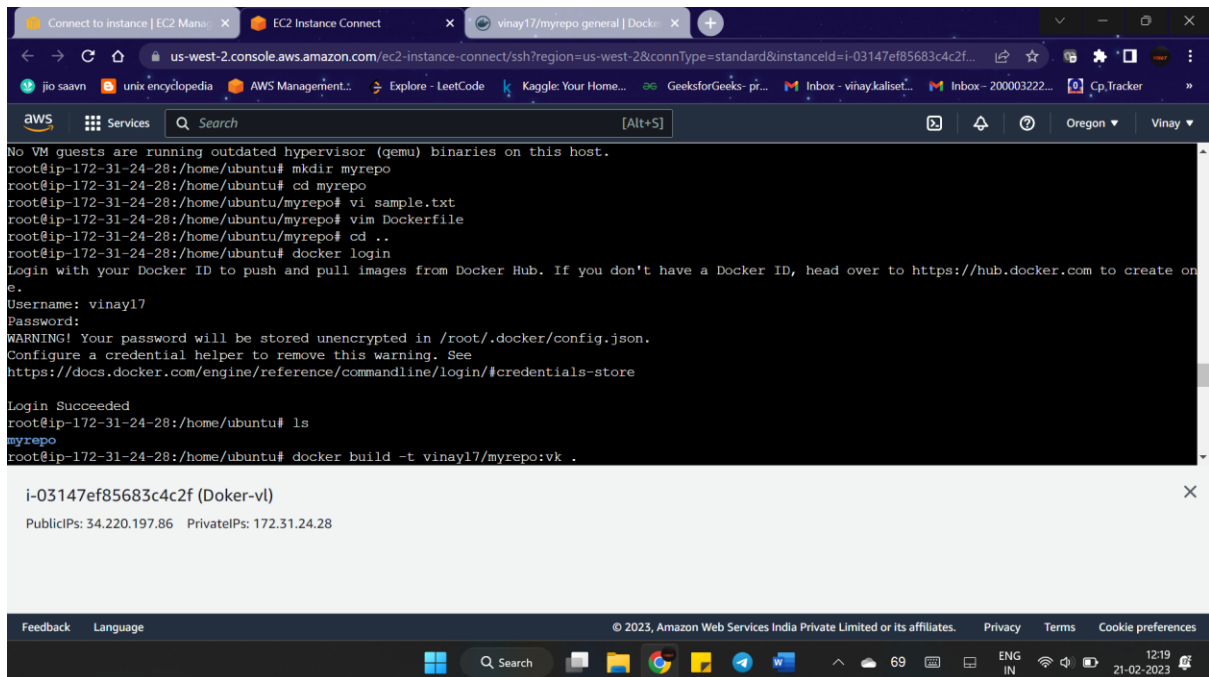


The screenshot shows a web browser window with the AWS Management Console. The active tab is 'EC2 Instance Connect', showing a terminal session for an EC2 instance. The terminal output is as follows:

```
FROM alpine:latest
WORKDIR /myrepo
COPY . .
RUN apk update
CMD ["cat", "sample.txt"]
```

Below the terminal output, a notification box displays the instance ID: `i-03147ef85683c4c2f (Doker-vl)` and its IP addresses: `PublicIPs: 34.220.197.86 PrivateIPs: 172.31.24.28`.

5. Login into your Docker Hub account using “docker login” command



The screenshot shows the same AWS EC2 Instance Connect terminal session. The terminal output continues with the following commands and their results:

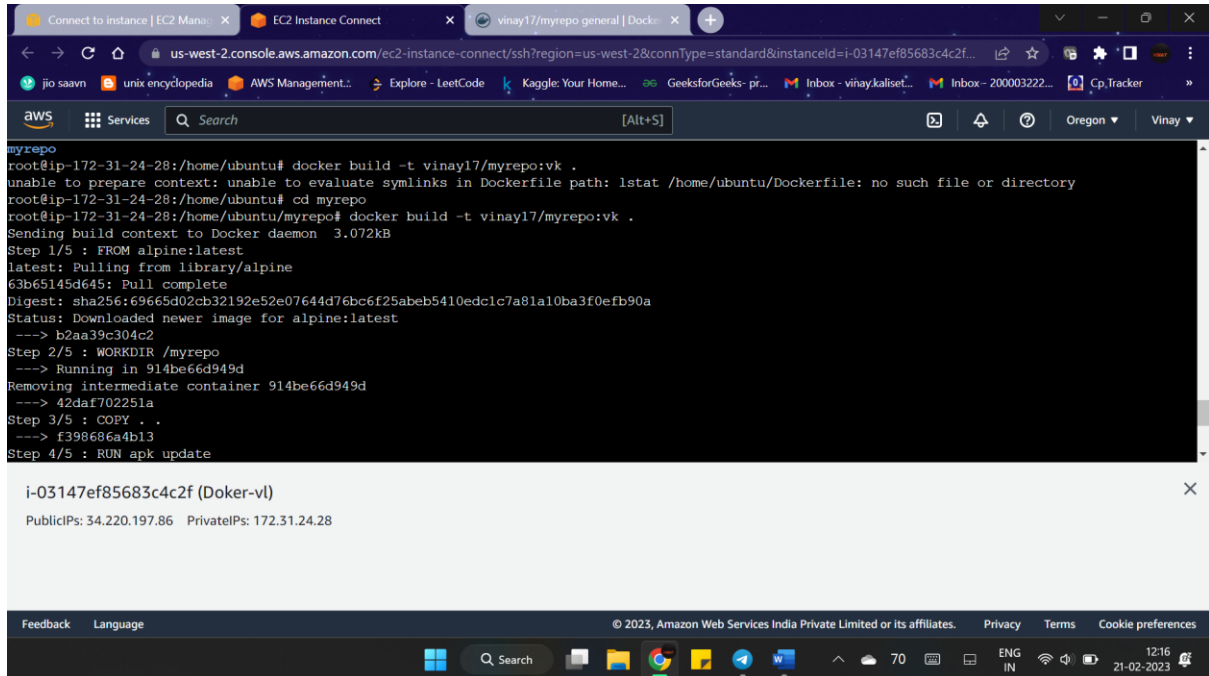
```
No VM guests are running outdated hypervisor (qemu) binaries on this host.
root@ip-172-31-24-28:/home/ubuntu# mkdir myrepo
root@ip-172-31-24-28:/home/ubuntu# cd myrepo
root@ip-172-31-24-28:/home/ubuntu/myrepo# vi sample.txt
root@ip-172-31-24-28:/home/ubuntu/myrepo# vim Dockerfile
root@ip-172-31-24-28:/home/ubuntu/myrepo# cd ..
root@ip-172-31-24-28:/home/ubuntu# docker login
Login with your Docker ID to push and pull images from Docker Hub. If you don't have a Docker ID, head over to https://hub.docker.com to create one.
Username: vinay17
Password:
WARNING! Your password will be stored unencrypted in /root/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
root@ip-172-31-24-28:/home/ubuntu# ls
myrepo
root@ip-172-31-24-28:/home/ubuntu# docker build -t vinay17/myrepo:vk .
```

Below the terminal output, the same notification box displays the instance ID: `i-03147ef85683c4c2f (Doker-vl)` and its IP addresses: `PublicIPs: 34.220.197.86 PrivateIPs: 172.31.24.28`.

6. Now build the repo using the following command:

docker build -t [username]/[reponame]:[tag name] .

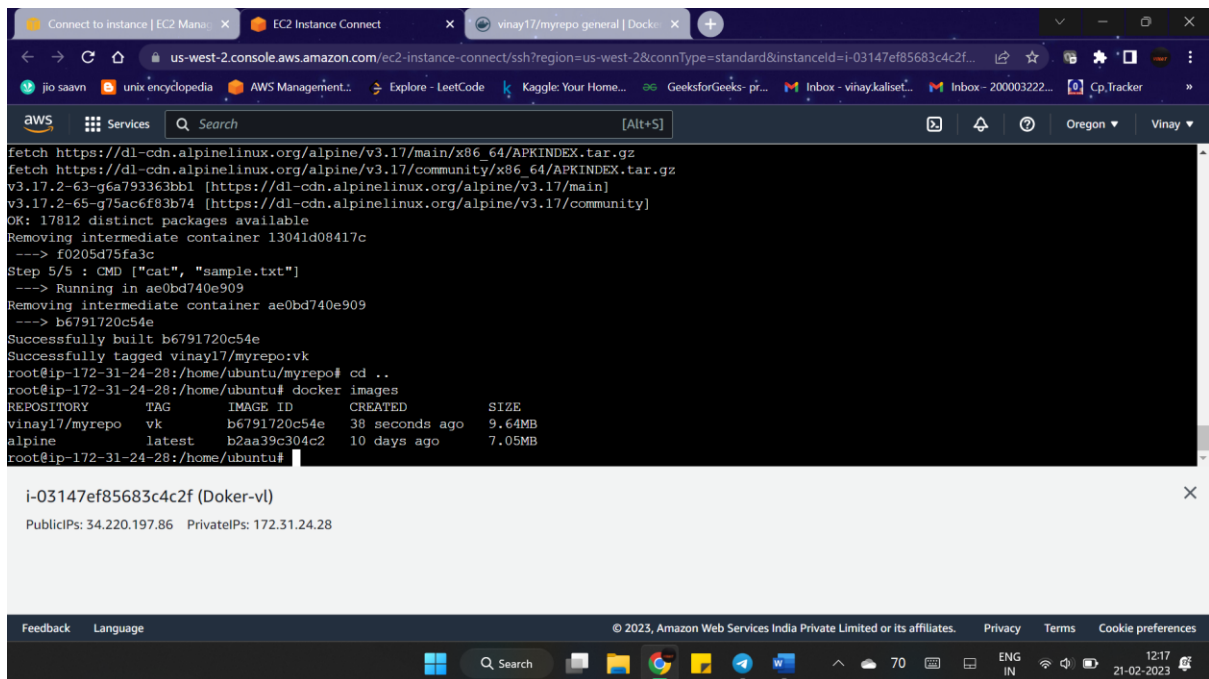


The screenshot shows the AWS Management Console interface for an EC2 instance. The terminal window displays the output of the `docker build` command. The build process starts by pulling the `alpine:latest` image, which is successful. It then proceeds to the `WORKDIR` and `COPY` steps, which are also successful. The final step is `RUN apk update`, which is still in progress. The console also shows the instance's public and private IP addresses.

```
root@ip-172-31-24-28:/home/ubuntu# docker build -t vinay17/myrepo:vk .
unable to prepare context: unable to evaluate symlinks in Dockerfile path: lstat /home/ubuntu/Dockerfile: no such file or directory
root@ip-172-31-24-28:/home/ubuntu# cd myrepo
root@ip-172-31-24-28:/home/ubuntu/myrepo# docker build -t vinay17/myrepo:vk .
Sending build context to Docker daemon 3.072kB
Step 1/5 : FROM alpine:latest
latest: Pulling from library/alpine
63b65145d645: Pull complete
Digest: sha256:69665d02cb32192e52e07644d76bc6f25abeb5410edc1c7a81a10ba3f0efb90a
Status: Downloaded newer image for alpine:latest
----> b2aa39c304c2
Step 2/5 : WORKDIR /myrepo
----> Running in 914be66d949d
Removing intermediate container 914be66d949d
----> 42daf702251a
Step 3/5 : COPY . .
----> f398686a4b13
Step 4/5 : RUN apk update
```

i-03147ef85683c4c2f (Docker-vl)
PublicIPs: 34.220.197.86 PrivateIPs: 172.31.24.28

7. Now check your image using “docker images” command



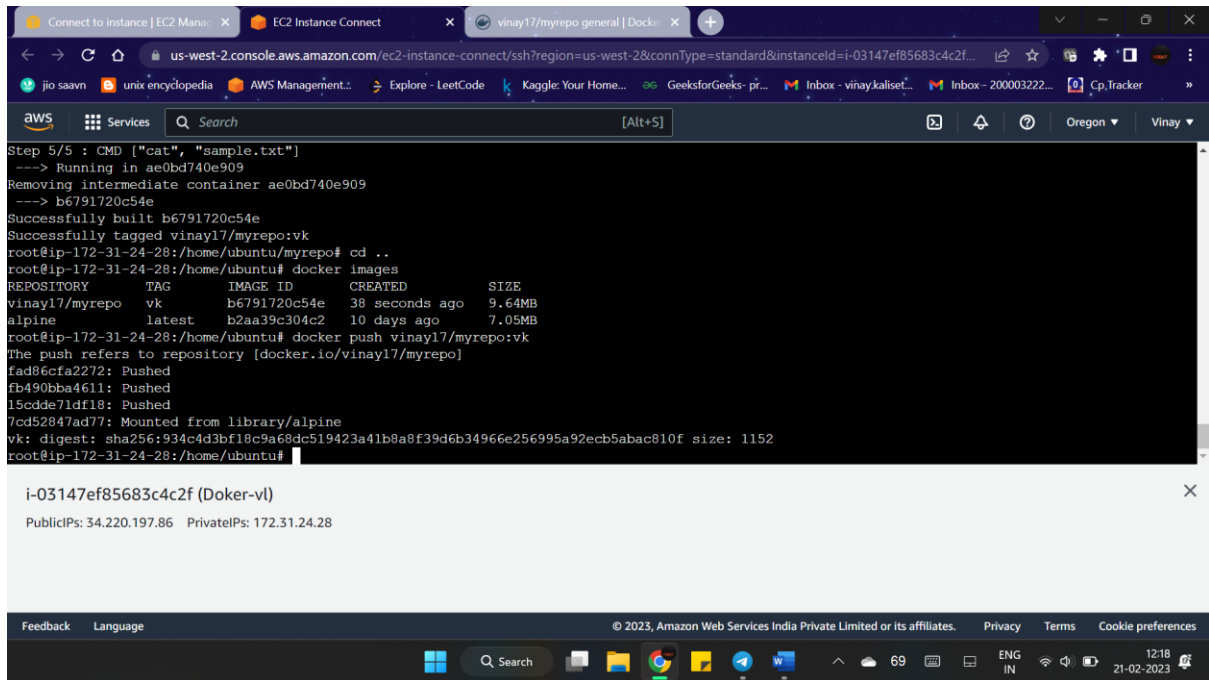
The screenshot shows the AWS Management Console interface for an EC2 instance. The terminal window displays the output of the `docker images` command. The output shows two images: `vinay17/myrepo:vk` and `alpine:latest`. The `vinay17/myrepo:vk` image is 9.64MB and was created 38 seconds ago. The `alpine:latest` image is 7.05MB and was created 10 days ago.

```
root@ip-172-31-24-28:/home/ubuntu/myrepo# cd ..
root@ip-172-31-24-28:/home/ubuntu# docker images
REPOSITORY          TAG             IMAGE ID        CREATED         SIZE
vinay17/myrepo      vk              b6791720c54e   38 seconds ago  9.64MB
alpine               latest          b2aa39c304c2   10 days ago    7.05MB
```

i-03147ef85683c4c2f (Docker-vl)
PublicIPs: 34.220.197.86 PrivateIPs: 172.31.24.28

8. Now push the directory into your repository:

docker push [username]/[reponame]:[tag name]

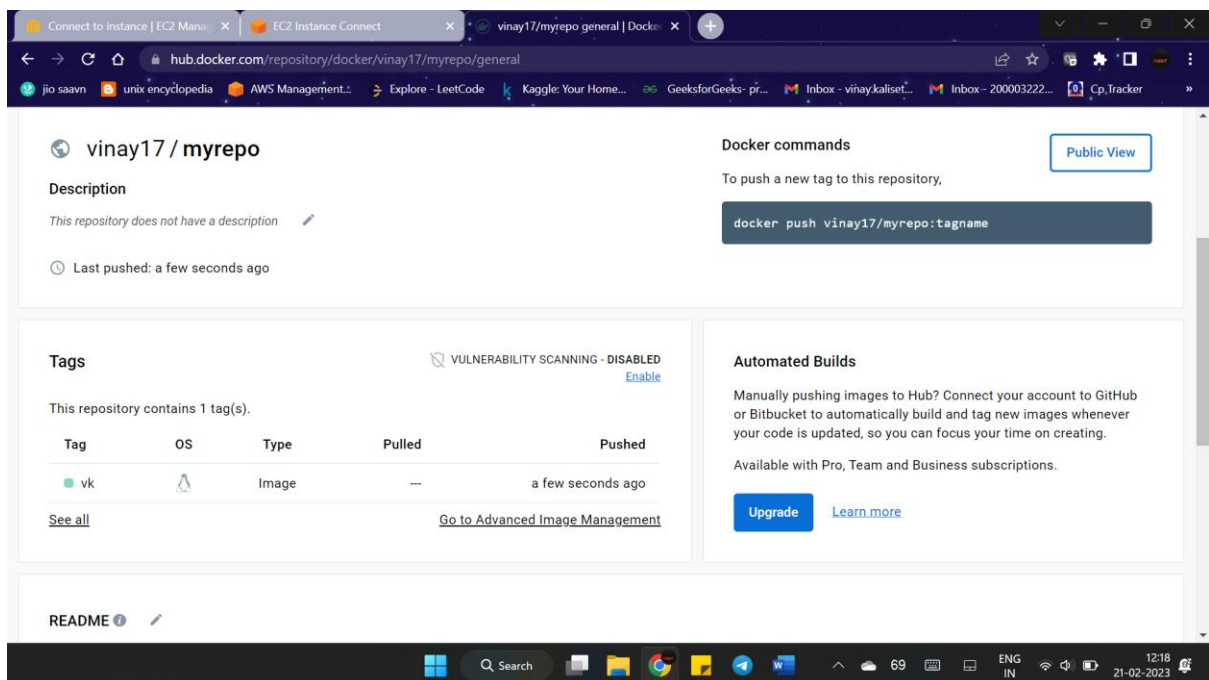


```
Step 5/5 : CMD ["cat", "sample.txt"]
----> Running in ae0bd740e909
Removing intermediate container ae0bd740e909
----> b6791720c54e
Successfully built b6791720c54e
Successfully tagged vinay17/myrepo:vk
root@ip-172-31-24-28:/home/ubuntu/myrepo# cd ..
root@ip-172-31-24-28:/home/ubuntu# docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
vinay17/myrepo       vk                  b6791720c54e       38 seconds ago     9.64MB
alpine               latest             b2aa39c304c2       10 days ago        7.05MB
root@ip-172-31-24-28:/home/ubuntu# docker push vinay17/myrepo:vk
The push refers to repository [docker.io/vinay17/myrepo]
fad86cfa2272: Pushed
fb490bba4611: Pushed
15cdde71df18: Pushed
7cd52847ad77: Mounted from library/alpine
vk: digest: sha256:934c4d3bf18c9a68dc519423a41b8a8f39d6b34966e256995a92ecb5abac810f size: 1152
root@ip-172-31-24-28:/home/ubuntu#
```

i-03147ef85683c4c2f (Docker-vl)

PublicIPs: 34.220.197.86 PrivateIPs: 172.31.24.28

9. The image will be pushed and updated in your Docker Hub repository.



vinay17 / myrepo

Description

This repository does not have a description

Last pushed: a few seconds ago

Docker commands

To push a new tag to this repository,

```
docker push vinay17/myrepo:tagname
```

Tags

This repository contains 1 tag(s).

Tag	OS	Type	Pulled	Pushed
vk	Linux	Image	---	a few seconds ago

See all

Go to Advanced Image Management

Automated Builds

Manually pushing images to 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.

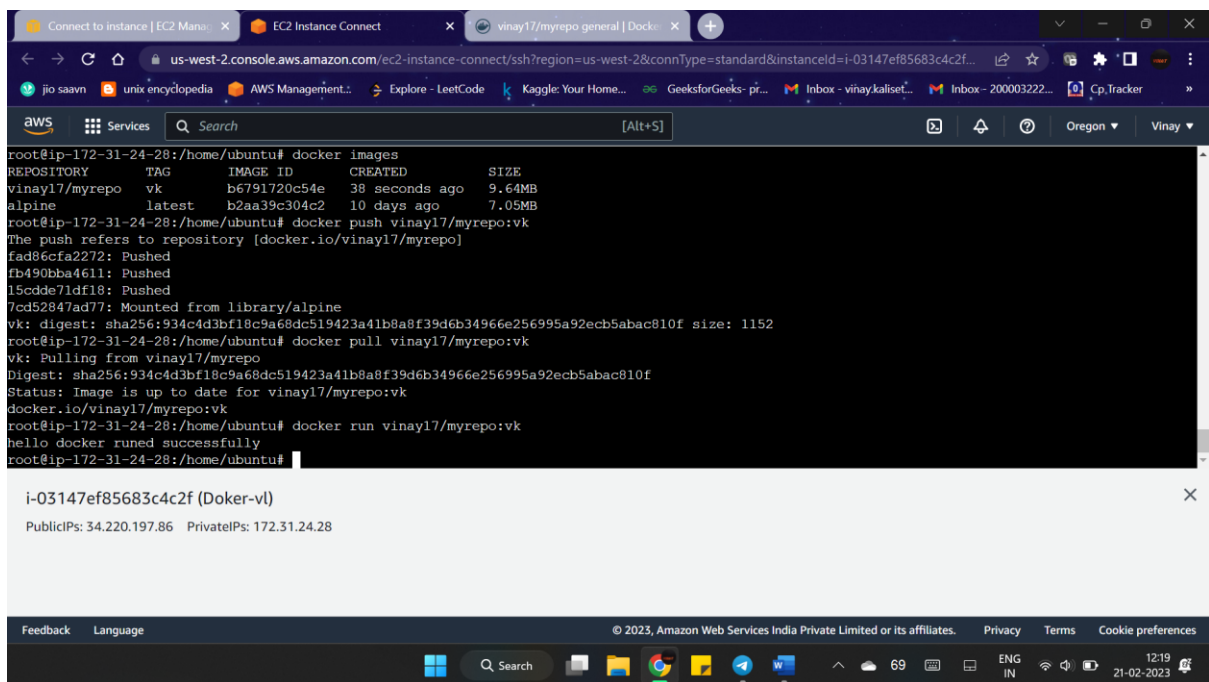
Upgrade Learn more

10. You can also pull the repo using the command:

docker pull [username]/[reponame]:[tagname]

11. Now run the file using the command:

docker run [username]/[reponame]:[tagname]



The screenshot shows a terminal window within the AWS Management Console, connected to an EC2 instance. The terminal displays the following commands and output:

```
root@ip-172-31-24-28:/home/ubuntu# docker images
REPOSITORY          TAG         IMAGE ID      CREATED        SIZE
vinay17/myrepo      vk          b6791720c54e  38 seconds ago 9.64MB
alpine              latest      b2aa39c304c2  10 days ago   7.05MB
root@ip-172-31-24-28:/home/ubuntu# docker push vinay17/myrepo:vk
The push refers to repository [docker.io/vinay17/myrepo]
fad86cfa2272: Pushed
fb490bba4611: Pushed
15cdde71df18: Pushed
7cd52847ad77: Mounted from library/alpine
vk: digest: sha256:934c4d3bf18c9a68dc519423a41b8a8f39d6b34966e256995a92ecb5abac810f size: 1152
root@ip-172-31-24-28:/home/ubuntu# docker pull vinay17/myrepo:vk
vk: Pulling from vinay17/myrepo
Digest: sha256:934c4d3bf18c9a68dc519423a41b8a8f39d6b34966e256995a92ecb5abac810f
Status: Image is up to date for vinay17/myrepo:vk
docker.io/vinay17/myrepo:vk
root@ip-172-31-24-28:/home/ubuntu# docker run vinay17/myrepo:vk
hello docker runed successfully
root@ip-172-31-24-28:/home/ubuntu#
```

Below the terminal output, a summary box for the container instance is visible:

```
i-03147ef85683c4c2f (Doker-vl)
PublicIPs: 34.220.197.86 PrivateIPs: 172.31.24.28
```

The bottom of the screenshot shows the AWS console footer with the text "© 2023, Amazon Web Services India Private Limited or its affiliates." and various utility links like Feedback, Language, Privacy, Terms, and Cookie preferences.