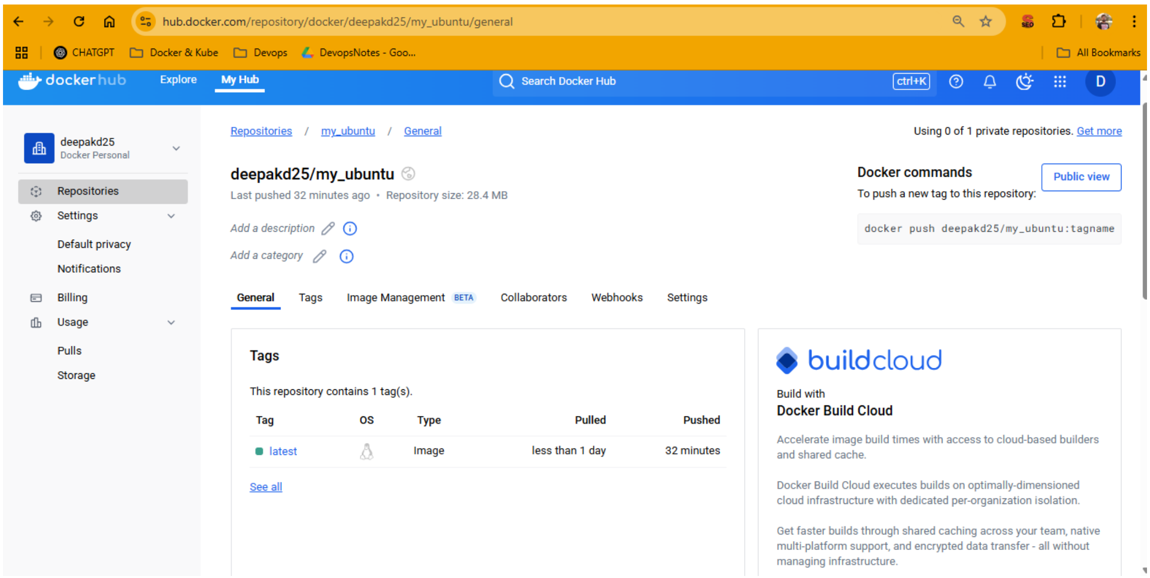
Storing a Docker Image in Docker Registry:

- Create a repo inside Docker Hub
- Rename desired Image with the Repo Name using "Docker Tag"
  - ➢ *docker tag ubuntu:latest deepakd25/my_ubuntu:latest*
- Do "Docker Login" for pushing the image to registry
  - ➢ *docker login -u deepakd25*
- Push the Image to Registry
  - ➢ *docker push deepakd25/my_ubuntu:latest*



hub.docker.com/repository/docker/deepakd25/my_ubuntu/general

CHATGPT | Docker & Kube | Devops | DevopsNotes - Goo... | All Bookmarks

docker hub | Explore | My Hub | Search Docker Hub | ctrl+K | D

deepakd25
Docker Personal

Repositories
Settings
Default privacy
Notifications
Billing
Usage
Pulls
Storage

Repositories / my_ubuntu / General

Using 0 of 1 private repositories. Get more

deepakd25/my_ubuntu
Last pushed 32 minutes ago • Repository size: 28.4 MB

Add a description
Add a category

Docker commands | Public view
To push a new tag to this repository:

```
docker push deepakd25/my_ubuntu:tagname
```

General | Tags | Image Management BETA | Collaborators | Webhooks | Settings

Tags

This repository contains 1 tag(s).

Tag	OS	Type	Pulled	Pushed
latest		Image	less than 1 day	32 minutes

See all

buildcloud

Build with
Docker Build Cloud

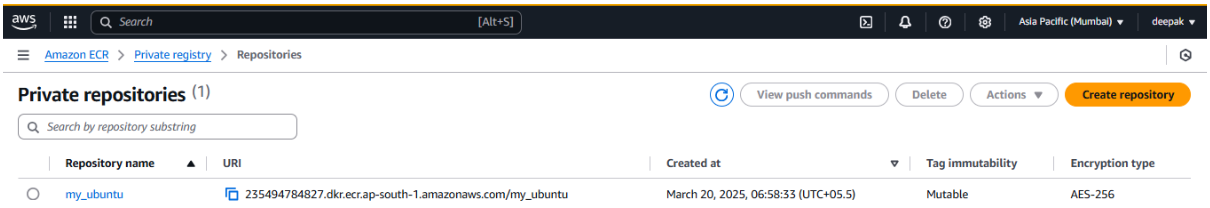
Accelerate image build times with access to cloud-based builders and shared cache.

Docker Build Cloud executes builds on optimally-dimensioned cloud infrastructure with dedicated per-organization isolation.

Get faster builds through shared caching across your team, native multi-platform support, and encrypted data transfer - all without managing infrastructure.

Storing a Docker Image in ECR (AWS Resource):

- Create a private Repo under ECR



aws | Search [Alt+S] | Asia Pacific (Mumbai) | deepak

Amazon ECR > Private registry > Repositories

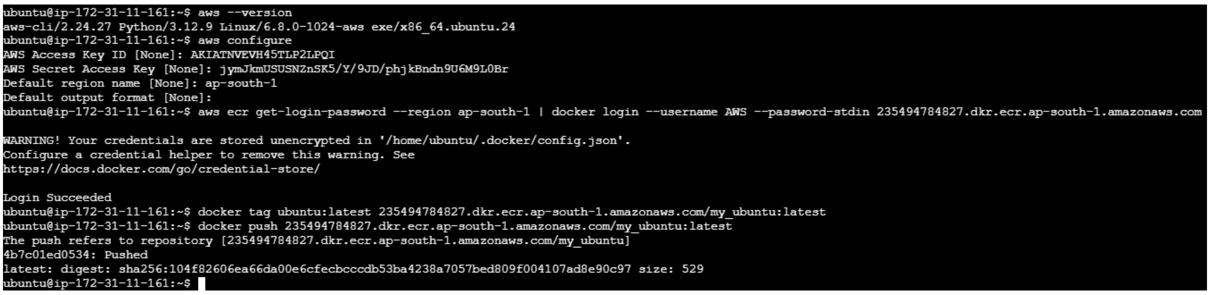
Private repositories (1)

View push commands | Delete | Actions | Create repository

Search by repository substring

	Repository name	URI	Created at	Tag immutability	Encryption type
○	my_ubuntu	235494784827.dkr.ecr.ap-south-1.amazonaws.com/my_ubuntu	March 20, 2025, 06:58:33 (UTC+05.5)	Mutable	AES-256

- Login using:
  - ➢ *aws ecr get-login-password --region ap-south-1 | docker login --username AWS --password-stdin 235494784827.dkr.ecr.ap-south-1.amazonaws.com*
- Create a new tag for the Image
  - ➢ *docker tag ubuntu:latest 235494784827.dkr.ecr.ap-south-1.amazonaws.com/my_ubuntu:latest*
- Push the Image onto the ECR Repo
  - ➢ *docker push 235494784827.dkr.ecr.ap-south-1.amazonaws.com/my_ubuntu:latest*



```
ubuntu@ip-172-31-11-161:~$ aws --version
aws-cli/2.24.27 Python/3.12.9 Linux/6.8.0-1024-aws exe/x86_64.ubuntu.24
ubuntu@ip-172-31-11-161:~$ aws configure
AWS Access Key ID [None]: AKIATNVEVH45TLP2LPQI
AWS Secret Access Key [None]: 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
Default region name [None]: ap-south-1
Default output format [None]:
ubuntu@ip-172-31-11-161:~$ aws ecr get-login-password --region ap-south-1 | docker login --username AWS --password-stdin 235494784827.dkr.ecr.ap-south-1.amazonaws.com

WARNING! Your credentials are stored unencrypted in '/home/ubuntu/.docker/config.json'.
Configure a credential helper to remove this warning. See
https://docs.docker.com/go/credential-store/

Login Succeeded
ubuntu@ip-172-31-11-161:~$ docker tag ubuntu:latest 235494784827.dkr.ecr.ap-south-1.amazonaws.com/my_ubuntu:latest
ubuntu@ip-172-31-11-161:~$ docker push 235494784827.dkr.ecr.ap-south-1.amazonaws.com/my_ubuntu:latest
The push refers to repository [235494784827.dkr.ecr.ap-south-1.amazonaws.com/my_ubuntu]
4b7c01ed0534: Pushed
latest: digest: sha256:104f82606ea66da00e6cfecbcccdb53ba4238a7057bed809f004107ad8e90c97 size: 529
ubuntu@ip-172-31-11-161:~$
```

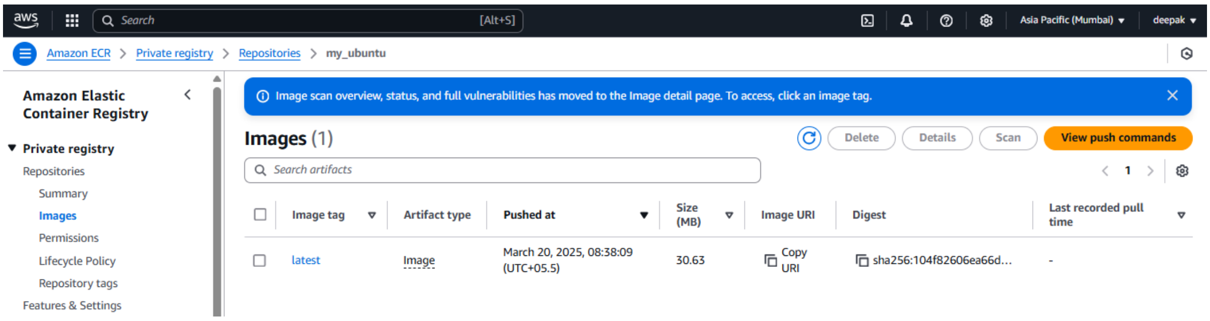


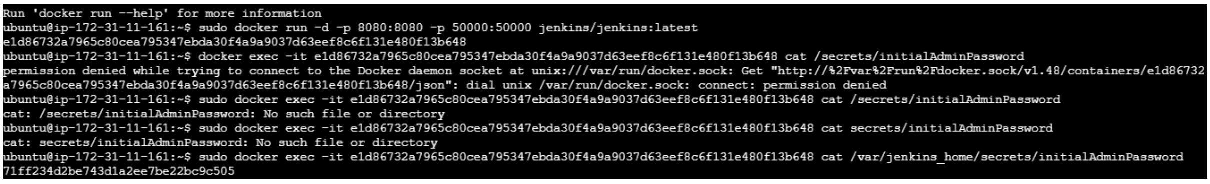
Image scan overview, status, and full vulnerabilities has moved to the Image detail page. To access, click an image tag.

Images (1)

	Image tag	Artifact type	Pushed at	Size (MB)	Image URI	Digest	Last recorded pull time
□	latest	Image	March 20, 2025, 08:38:09 (UTC+05.5)	30.63	Copy URI	sha256:104f82606ea66d...	-

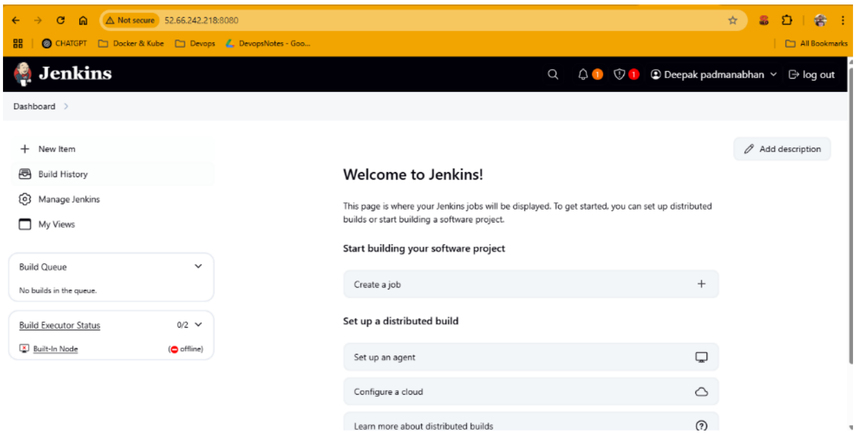
Preserve Persistent data without Binding → By creating an Image from a Running Container (using Docker Commit)

- Create a Container for running Jenkins without bind mode
  - ➢ ***sudo docker run -d -p 8080:8080 -p 50000:50000 jenkins/jenkins:latest***



```
Run 'docker run --help' for more information
ubuntu@ip-172-31-11-161:~$ sudo docker run -d -p 8080:8080 -p 50000:50000 jenkins/jenkins:latest
e1d86732a7965c80cea795347ebda30f4a9a9037d63eef8c6f131e480f13b648
ubuntu@ip-172-31-11-161:~$ docker exec -it e1d86732a7965c80cea795347ebda30f4a9a9037d63eef8c6f131e480f13b648 cat /secrets/initialAdminPassword
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/e1d86732
a7965c80cea795347ebda30f4a9a9037d63eef8c6f131e480f13b648/json": dial unix /var/run/docker.sock: connect: permission denied
ubuntu@ip-172-31-11-161:~$ sudo docker exec -it e1d86732a7965c80cea795347ebda30f4a9a9037d63eef8c6f131e480f13b648 cat /secrets/initialAdminPassword
cat: /secrets/initialAdminPassword: No such file or directory
ubuntu@ip-172-31-11-161:~$ sudo docker exec -it e1d86732a7965c80cea795347ebda30f4a9a9037d63eef8c6f131e480f13b648 cat secrets/initialAdminPassword
cat: secrets/initialAdminPassword: No such file or directory
ubuntu@ip-172-31-11-161:~$ sudo docker exec -it e1d86732a7965c80cea795347ebda30f4a9a9037d63eef8c6f131e480f13b648 cat /var/jenkins_home/secrets/initialAdminPassword
71ff234d2be743d1a2ee7be22bc9c505
```

- Setup the Jenkins



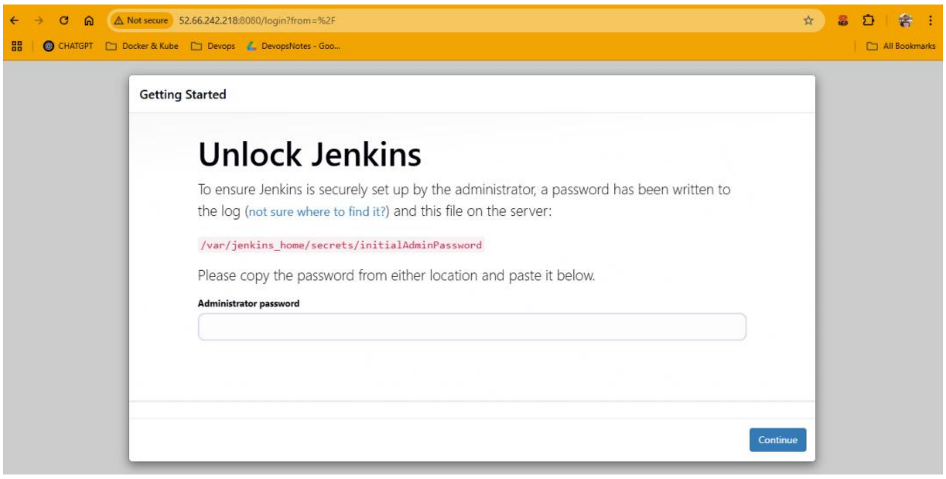
Welcome to Jenkins!

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.

Start building your software project

Set up a distributed build

- Creating an Image from a Running Container (using Docker Commit)
  - ➢ ***sudo docker commit e1d86732a796 my_jenkins_setup:latest***
- Remove the Container and do the Docker run command using:
  - ➢ ***sudo docker run -d -p 8080:8080 -p 50000:50000 my_jenkins_setup:latest***
- Try accessing Jenkins again



Getting Started

Unlock Jenkins

To ensure Jenkins is securely set up by the administrator, a password has been written to the log (not sure where to find it?) and this file on the server:

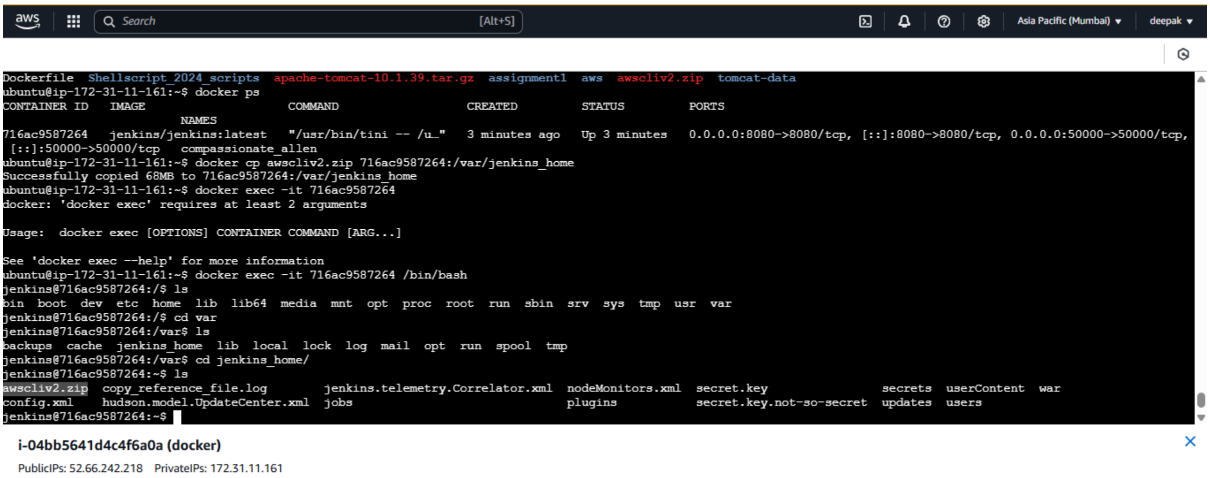
/var/jenkins_home/secrets/initialAdminPassword

Please copy the password from either location and paste it below.

Administrator password

Copy file from local to Running Container

- ➢ ***docker cp source container_id:path***



```
Dockerfile  Shellscript_2024_scripts  apache-tomcat-10.1.39.tar.gz  assignment1  aws  awscliv2.zip  tomcat-data
ubuntu@ip-172-31-11-161:~$ docker ps
CONTAINER ID   IMAGE                     COMMAND                  CREATED         STATUS         PORTS
                         NAMES
716ac9587264   jenkins/jenkins:latest   "/usr/bin/tini -- /u…"   3 minutes ago   Up 3 minutes   0.0.0.0:8080->8080/tcp, [::]:8080->8080/tcp, 0.0.0.0:50000->50000/tcp,
 [::]:50000->50000/tcp   compassionate_allen
ubuntu@ip-172-31-11-161:~$ docker cp awscliv2.zip 716ac9587264:/var/jenkins_home
Successfully copied 68MB to 716ac9587264:/var/jenkins_home
ubuntu@ip-172-31-11-161:~$ docker exec -it 716ac9587264
docker: 'docker exec' requires at least 2 arguments

Usage:  docker exec [OPTIONS] CONTAINER COMMAND [ARG...]

See 'docker exec --help' for more information
ubuntu@ip-172-31-11-161:~$ docker exec -it 716ac9587264 /bin/bash
jenkins@716ac9587264:/$ ls
bin  boot  dev  etc  home  lib  lib64  media  mnt  opt  proc  root  run  sbin  srv  sys  tmp  usr  var
jenkins@716ac9587264:/$ cd var
jenkins@716ac9587264:/var$ ls
backups  cache  jenkins_home  lib  local  lock  log  mail  opt  run  spool  tmp
jenkins@716ac9587264:/var$ cd jenkins_home/
jenkins@716ac9587264:~$ ls
awscliv2.zip  copy_reference_file.log       jenkins.telemetry.Correlator.xml  nodeMonitors.xml  secret.key                secrets  userContent  war
config.xml    hudson.model.UpdateCenter.xml  jobs                              plugins           secret.key.not-so-secret  updates  users
jenkins@716ac9587264:~$
```

i-04bb5641d4c4f6a0a (docker)

PublicIPs: 52.66.242.218   PrivateIPs: 172.31.11.161