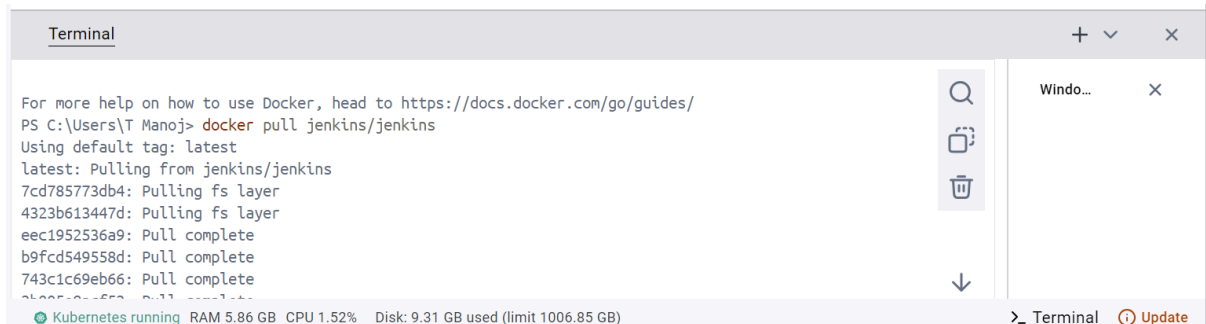


NAME : T MANOJ

EMPID : LYAKE2KHS

Setting up Jenkins with Docker

1. Go to hub.docker.com and open the Jenkins repository
2. Copy the pull command



```
Terminal
For more help on how to use Docker, head to https://docs.docker.com/go/guides/
PS C:\Users\T Manoj> docker pull jenkins/jenkins
Using default tag: latest
latest: Pulling from jenkins/jenkins
7cd785773db4: Pulling fs layer
4323b613447d: Pulling fs layer
eec1952536a9: Pull complete
b9fcd549558d: Pull complete
743c1c69eb66: Pull complete
Digest: sha256:be95e0848c42be95e0848c42be95e0848c42be95e0848c42be95e0848c42be95e0848c42
Status: Pull complete
Kubernetes running RAM 5.86 GB CPU 1.52% Disk: 9.31 GB used (limit 1006.85 GB) Terminal Update
```

Images [Give feedback](#)

View and manage your local and Docker Hub images. [Learn more](#)

<input type="checkbox"/>	<input type="radio"/>	docker/desktop-vpnkit-cr	dc331cb22850be0cdd97	556098075b3d	2 years ago	36.22 MB	▶	:	🗑
<input type="checkbox"/>	<input type="radio"/>	docker/desktop-storage-	v2.0	99f89471f470	4 years ago	41.85 MB	▶	:	🗑
<input type="checkbox"/>	<input type="radio"/>	jenkins/jenkins	latest	be95e0848c42	8 days ago	465.56 MB	▶	:	🗑

Showing 22 items

3. Run the command in your Docker desktop terminal

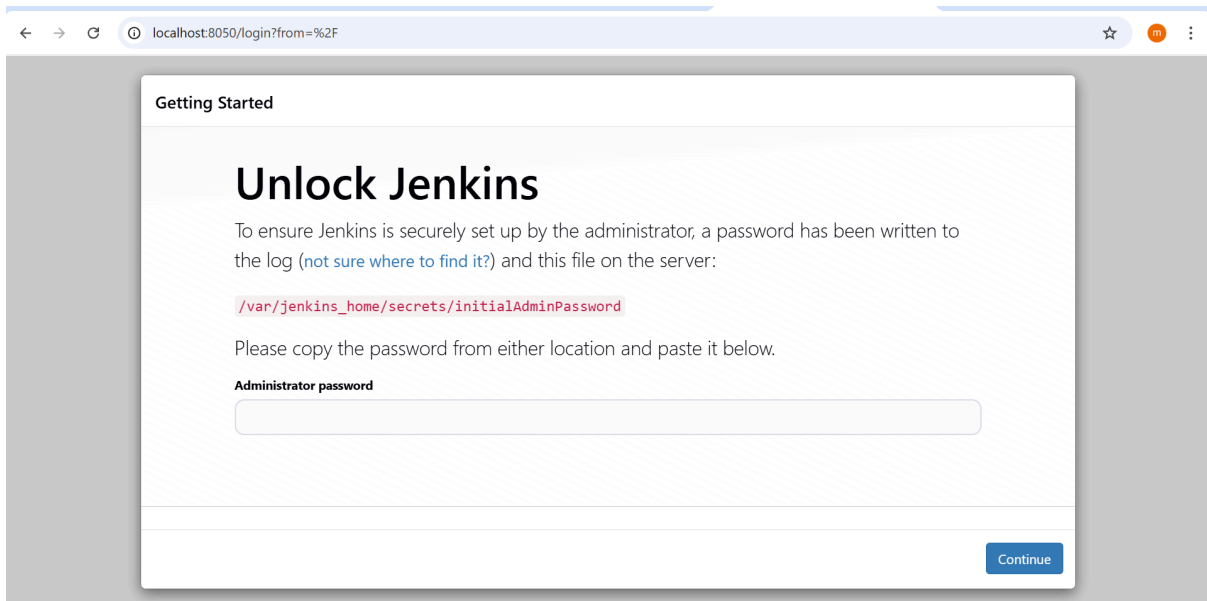
To create and run the Jenkins container:

```
docker run -p 8050:8080 -p 50000:50000 -v
jenkins_home:/var/jenkins_home jenkins/jenkins
```



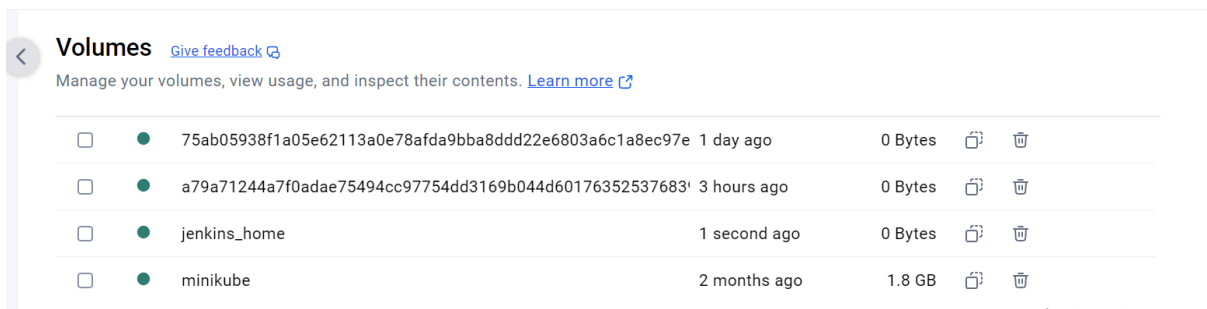
```
Terminal
*****
2025-04-02 07:40:09.376+0000 [id=35] INFO jenkins.InitReactorRunner$1#onAttained: Completed initiali
zation
2025-04-02 07:40:09.402+0000 [id=25] INFO hudson.lifecycle.Lifecycle#onReady: Jenkins is fully up an
d running
2025-04-02 07:40:13.658+0000 [id=68] INFO h.m.DownloadService$Downloadable#load: Obtained the update
d data file for hudson.tasks.Maven.MavenInstaller
2025-04-02 07:40:13.659+0000 [id=68] INFO hudson.util.Retrier#start: Performed the action check upda
tes server successfully at the attempt #1
Windo... X
```

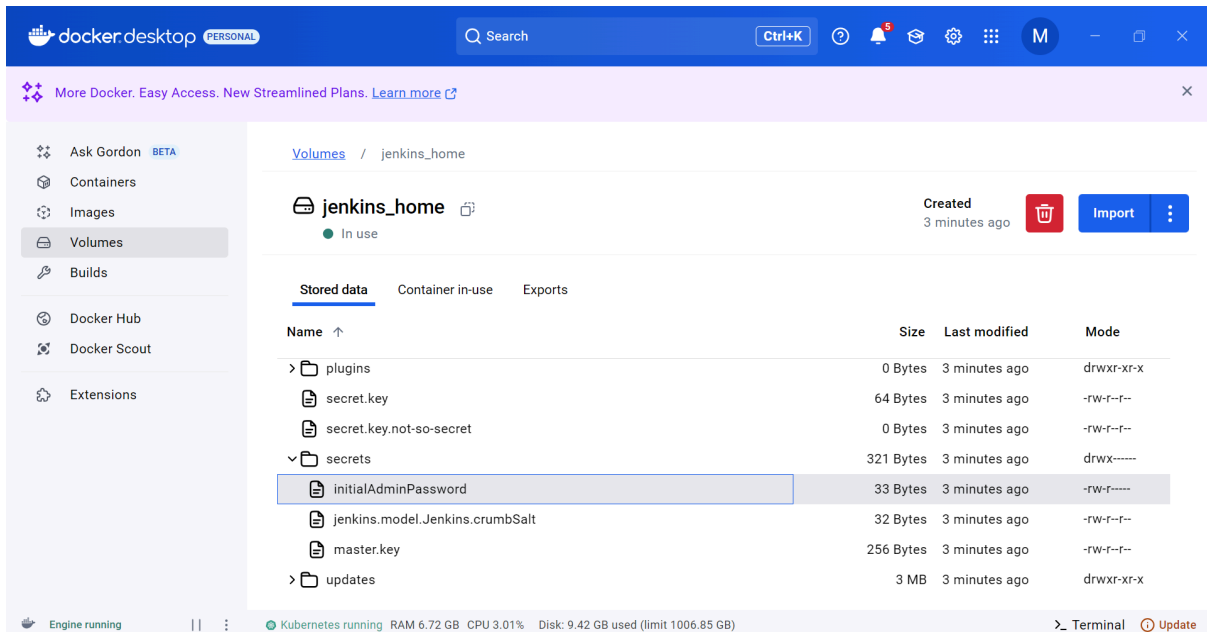
4. Open Chrome and navigate to **localhost:8050**



5. Get the initial admin password:

- Navigate to Volume > secrets > initial admin password
- Save the password to your local system
- View and copy the password
- Paste it into Jenkins setup



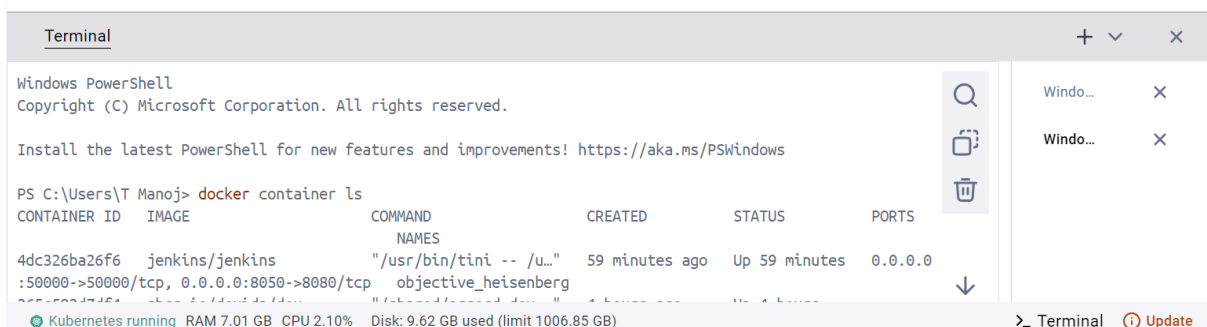


Setting up VS Code and Jenkins Pipeline

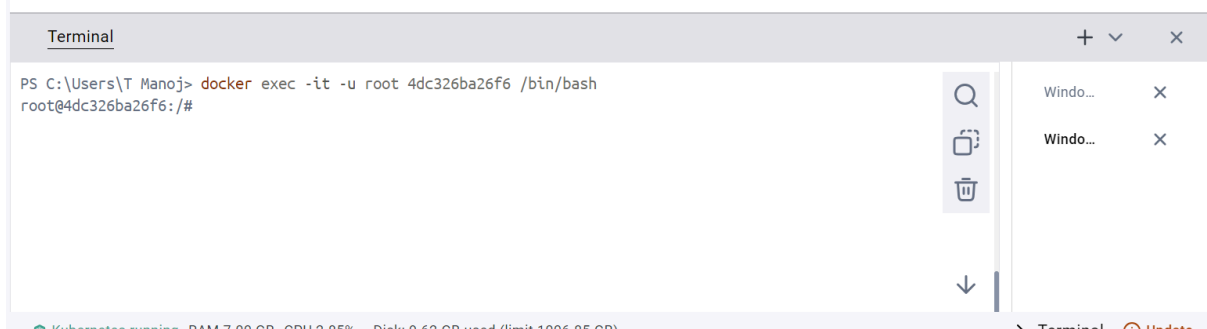
1. Open VS Code and install the Jenkins extension
2. Create a Jenkinsfile

Creating a Pipeline in Jenkins

1. Write the pipeline script
2. Open a new terminal without closing the previous one
3. Run `docker container ls` to get the container ID

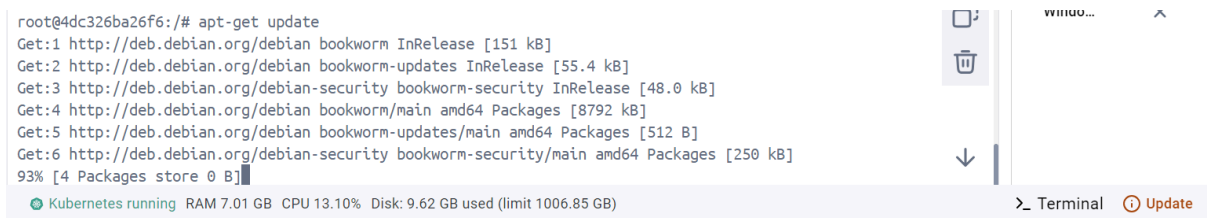


4. Access the container: `docker exec -it -u root [containerID] /bin/bash`

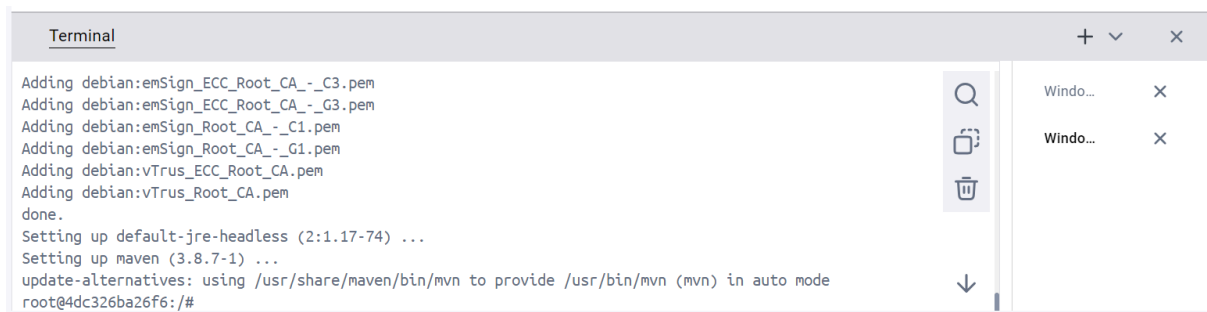


```
Terminal
PS C:\Users\T Manoj> docker exec -it -u root 4dc326ba26f6 /bin/bash
root@4dc326ba26f6:/#
```

5. Update and install Maven (build tool)



```
root@4dc326ba26f6:/# apt-get update
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 [512 B]
Get:6 http://deb.debian.org/debian-security bookworm-security/main amd64 Packages [250 kB]
93% [4 Packages store 0 B]
```

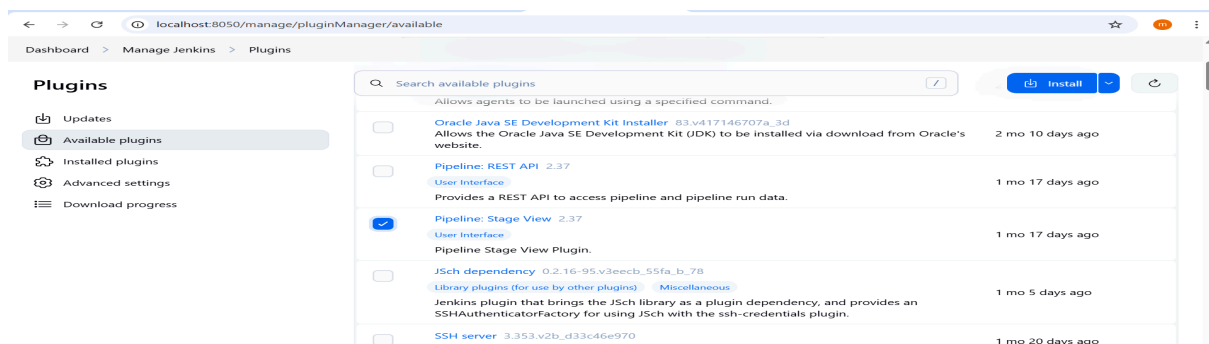


```
Adding debian:emSign_ECC_Root_CA_-_C3.pem
Adding debian:emSign_ECC_Root_CA_-_G3.pem
Adding debian:emSign_Root_CA_-_C1.pem
Adding debian:emSign_Root_CA_-_G1.pem
Adding debian:vTrus_ECC_Root_CA.pem
Adding debian:vTrus_Root_CA.pem
done.
Setting up default-jre-headless (2:1.17-74) ...
Setting up maven (3.8.7-1) ...
update-alternatives: using /usr/share/maven/bin/mvn to provide /usr/bin/mvn (mvn) in auto mode
root@4dc326ba26f6:/#
```

6. Exit the container

Setting up the Jenkins Job

1. Install the "Pipeline: Stage View" plugin



←

→

↺

localhost:8050/manage/pluginManager/updates/

☆

m

⋮

Dashboard > Manage Jenkins > Plugins

Plugins

Updates

Available plugins

Installed plugins

Advanced settings

Download progress

Matrix Authorization Strategy

✓ Success

PAM Authentication

✓ Success

LDAP

✓ Success

Email Extension

✓ Success

Mailer

✓ Success

Theme Manager

✓ Success

Dark Theme

✓ Success

Loading plugin extensions

✓ Success

Pipeline: REST API

✓ Success

Pipeline: Stage View

✓ Success

Loading plugin extensions

✓ Success

→ [Go back to the top page](#)

(you can start using the installed plugins right away)

→ ☐ Restart Jenkins when installation is complete and no jobs are running

REST APIJenkins 2.503

←

→

↺

localhost:8050/view/all/newJob

☆

m

⋮

Dashboard > All > New Item

Jenkins

🔍🔔1🛡️1manojlog out

New Item

Enter an item name

PipeLine

Select an item type

Freestyle project

Classic, general-purpose job type that checks out from up to one SCM, executes build steps serially, followed by post-build steps like archiving artifacts and sending email notifications.

Pipeline

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

Multi-configuration project

OK

localhost:8050/job/Pipeline/configure

Jenkins

Dashboard > Pipeline > Configuration

Configure

General

Enabled

Description

PIPELINE FOR GIT AND JENKINS USING DOCKER

Plain text [Preview](#)

☐ Discard old builds ?

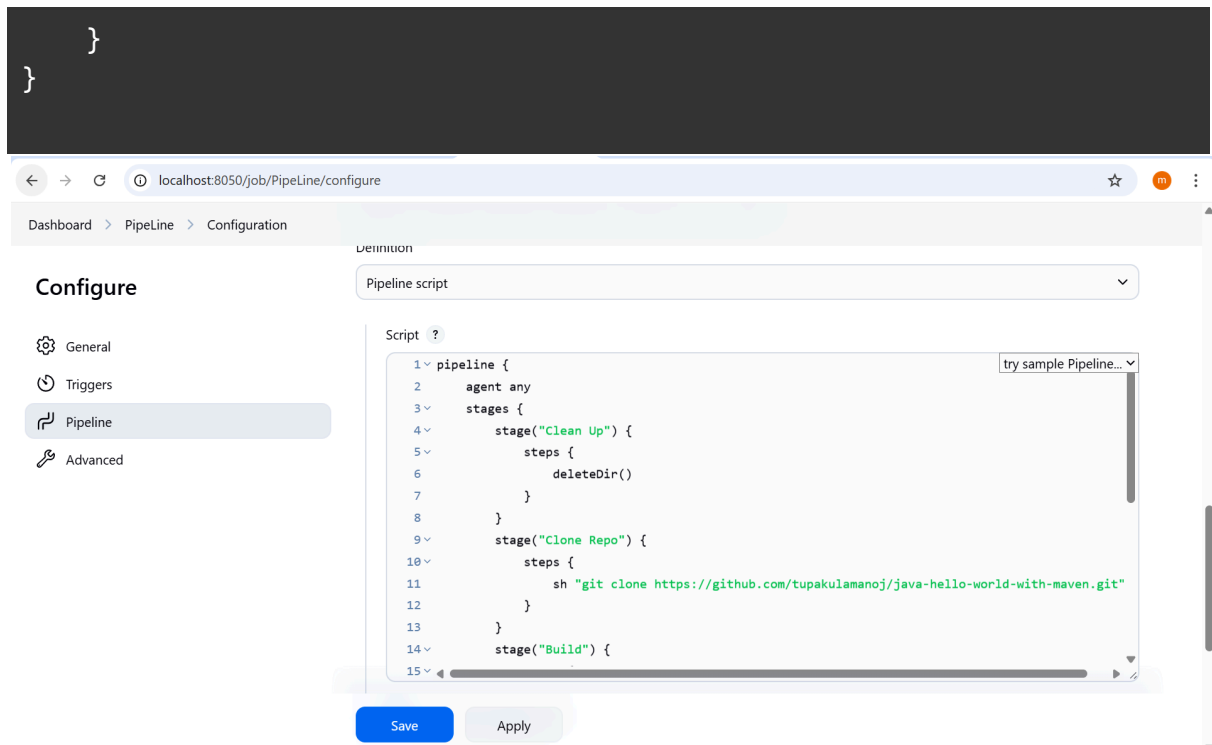
☐ Do not allow concurrent builds

☐ Do not allow the pipeline to resume if the controller restarts

Save Apply

2. Create a job with this pipeline script:

```
pipeline {
    agent any
    stages {
        stage("Clean Up") {
            steps {
                deleteDir()
            }
        }
        stage("Clone Repo") {
            steps {
                sh "git clone
https://github.com/tupakulamanoj/java-hello-world-with-maven.git"
            }
        }
        stage("Build") {
            steps {
                dir("java-hello-world-with-maven") {
                    sh "mvn clean install"
                }
            }
        }
        stage("Test") {
            steps {
                dir("java-hello-world-with-maven") {
                    sh "mvn test"
                }
            }
        }
    }
}
```



3. Run and view the pipeline execution

