Jenkin Installation set up:

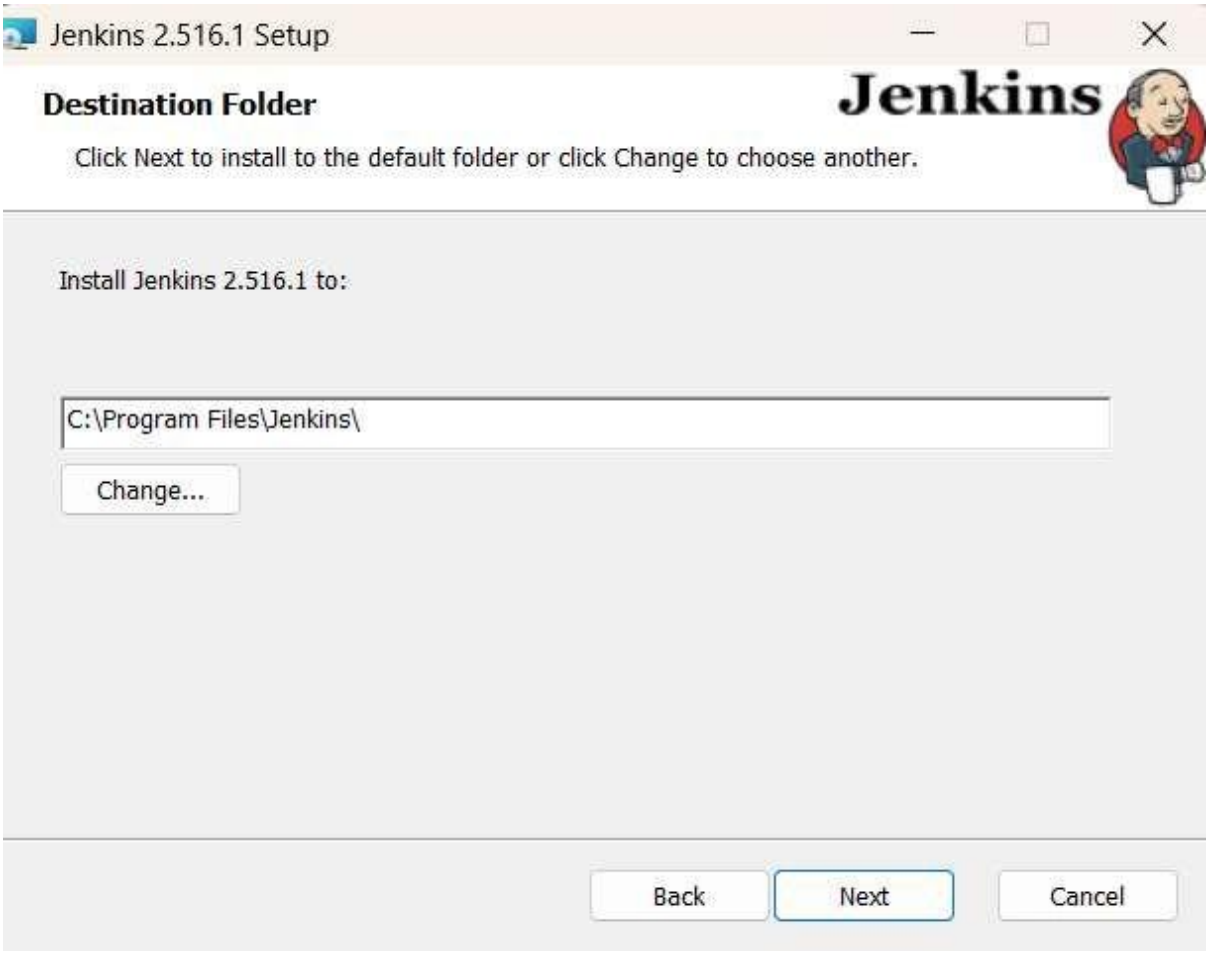
Downloading:

1. Type Jenkins in browser
2. Click on Jenkins.io website and click on download
3. Scrolldown and select windows from “Download Jenkins 2.504.3LTS for:”
4. It download Jenkins.msi

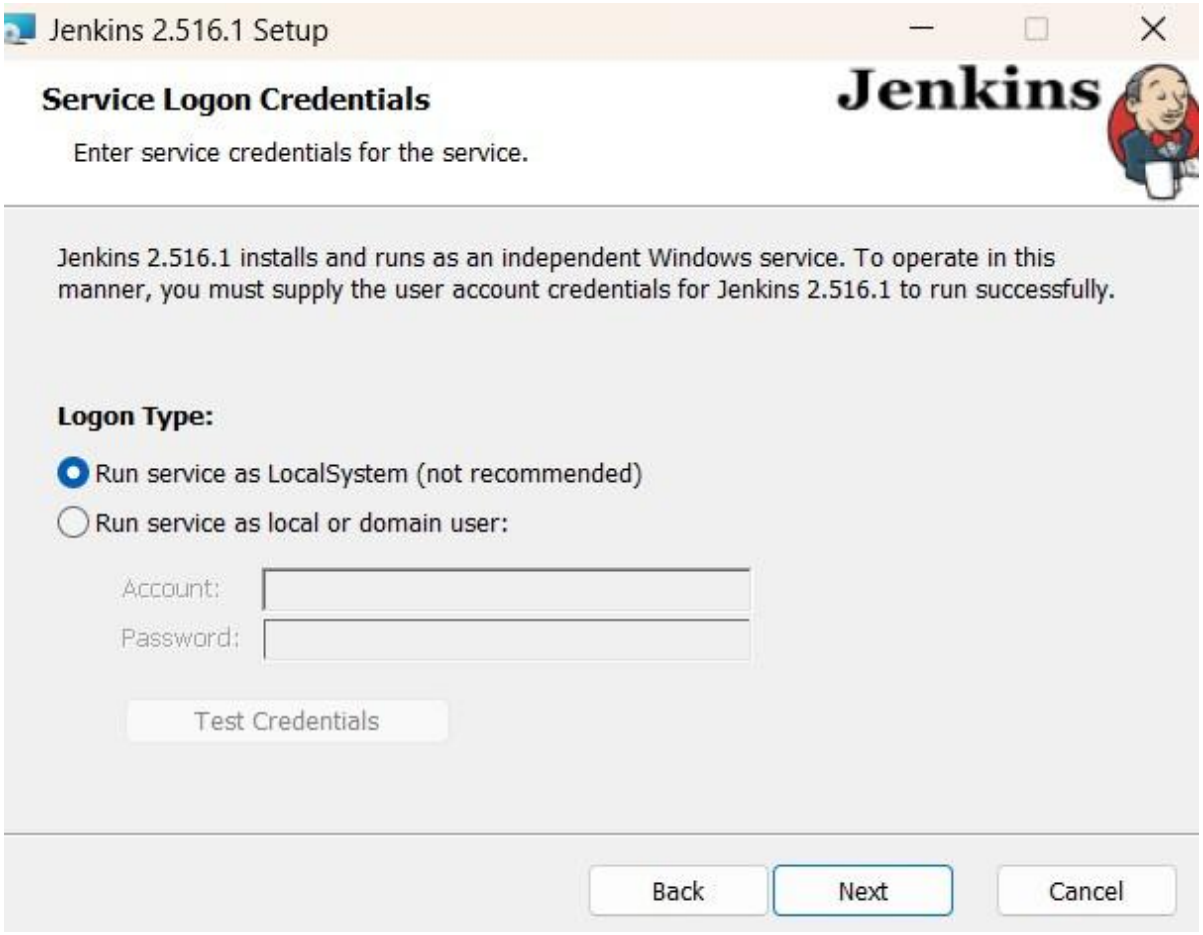
Installation:

5. After Downloading double click on .msi file.

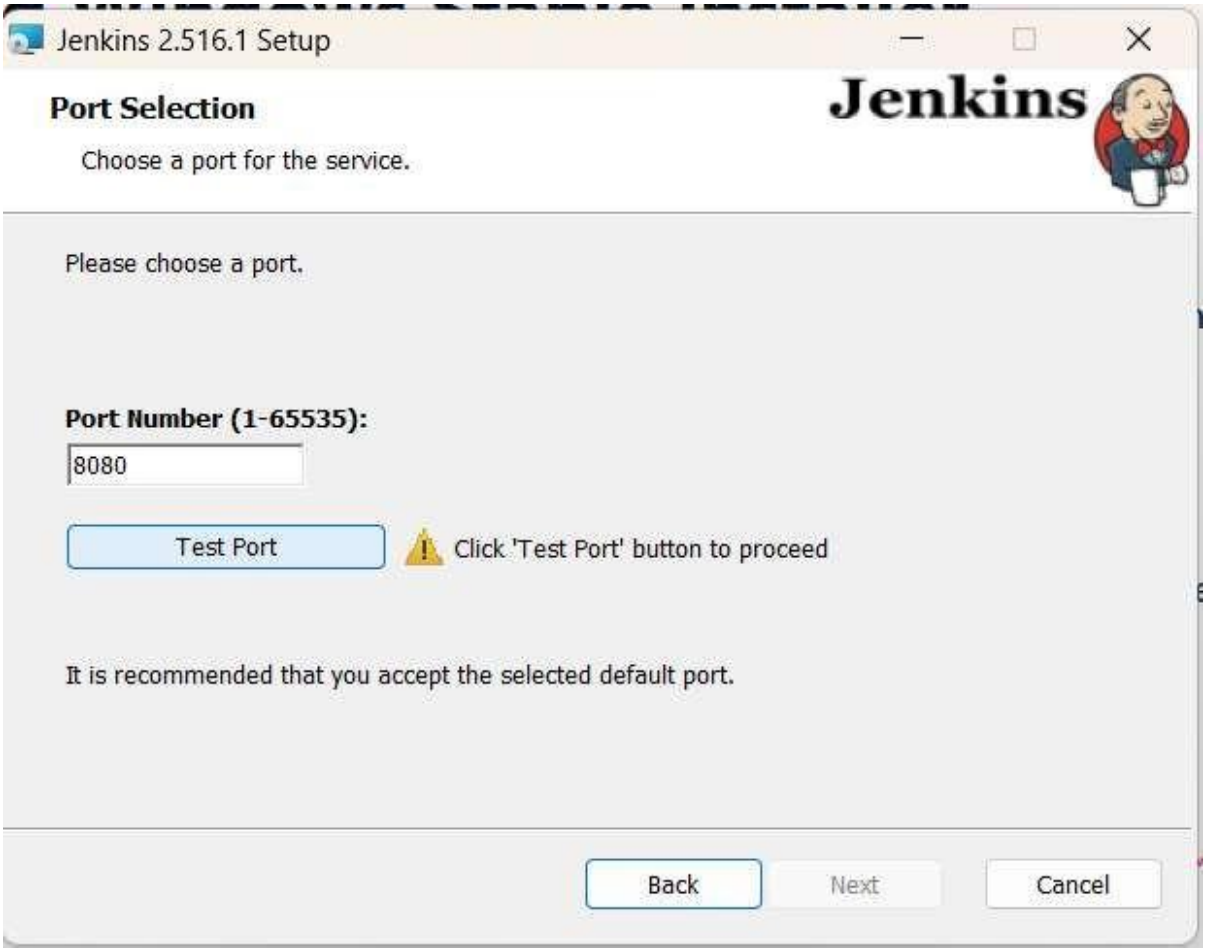
6. It gives the default folder to install select next



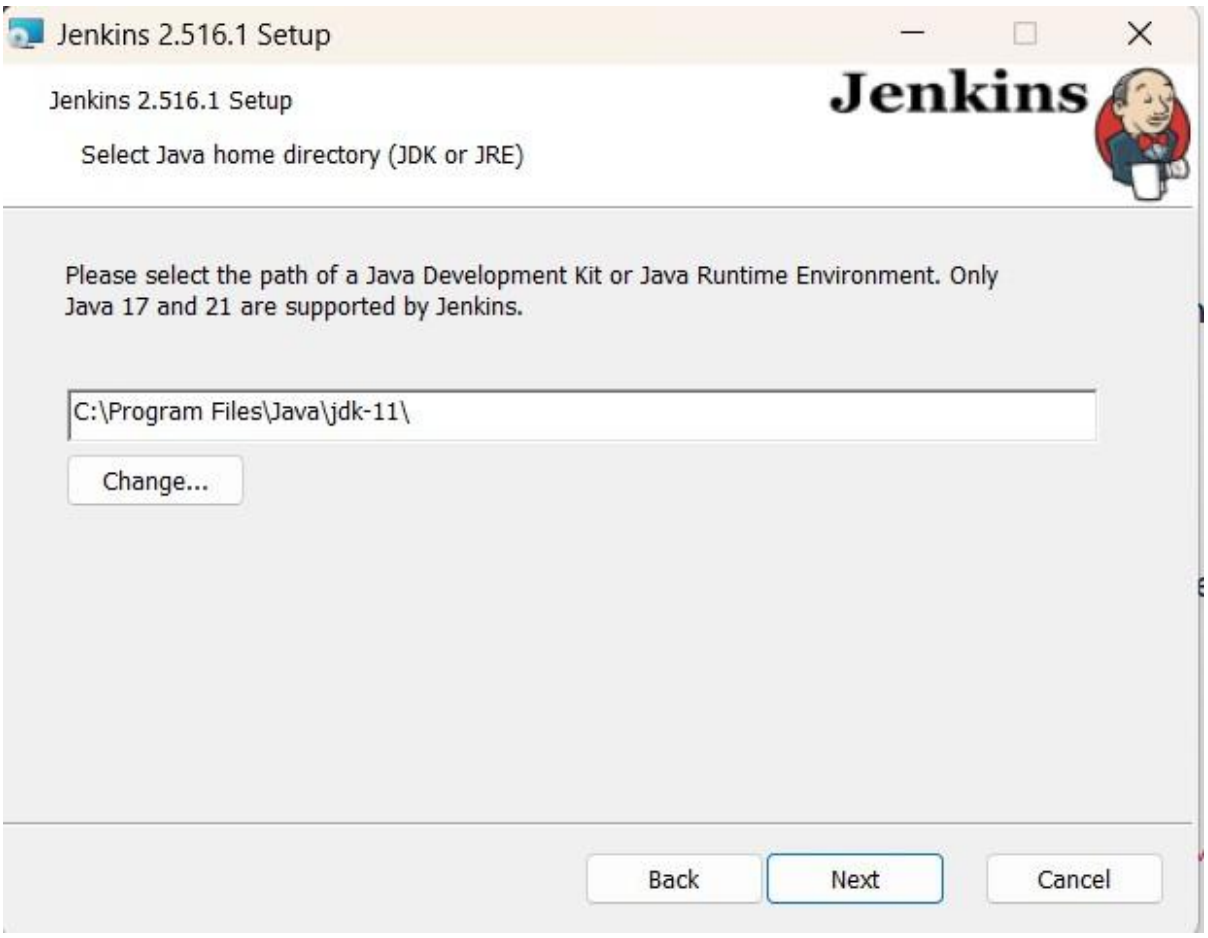
7. Select “Run service as a local system“ click next.



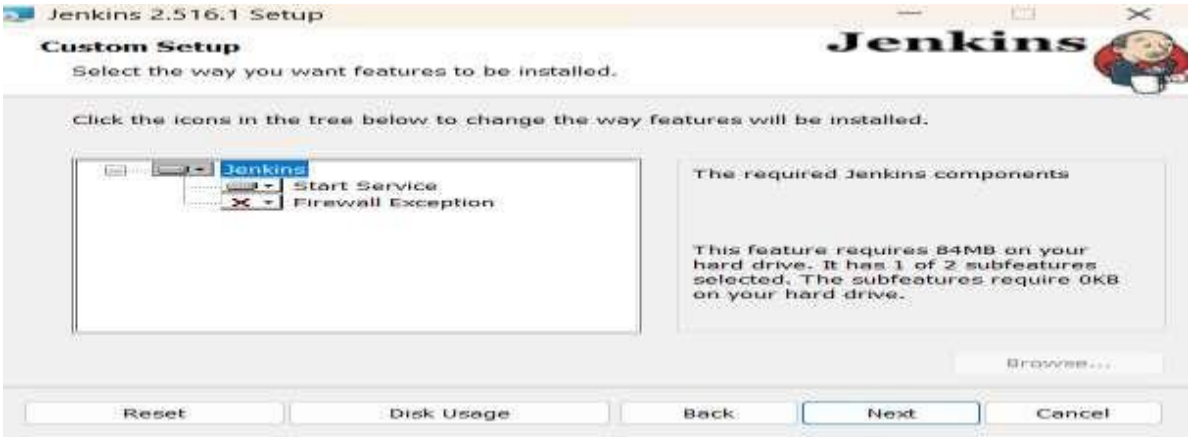
8. It gives default port number 8080.Click on test port if available it gives tick mark.



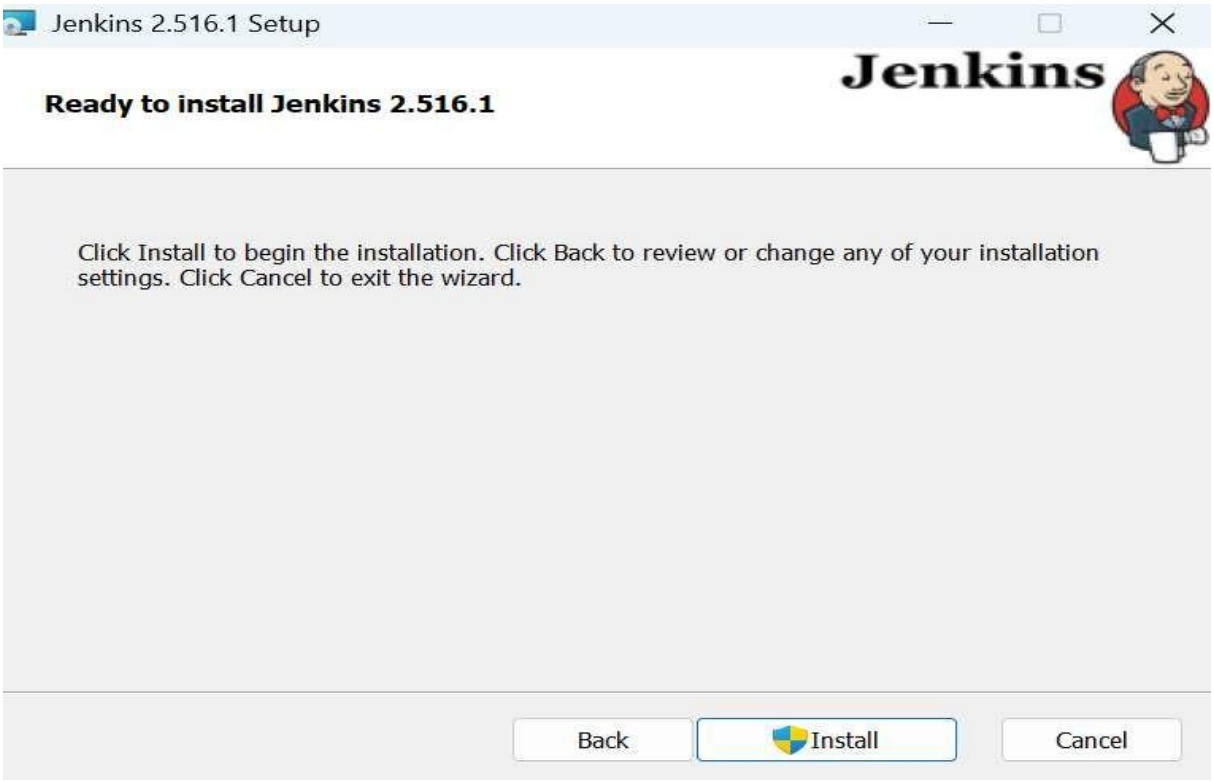
9. It gives the path automatically ,otherwise select the proper java path by clicking change button ("C.\Program Files\Java\jdk-17) Click next.



10. click next again



11. Click Install



12. click finish

13. type localhost:8080 in browser

14. Goto the location “C:\programData\Jenkins\.jenkins\secrets” for initial password

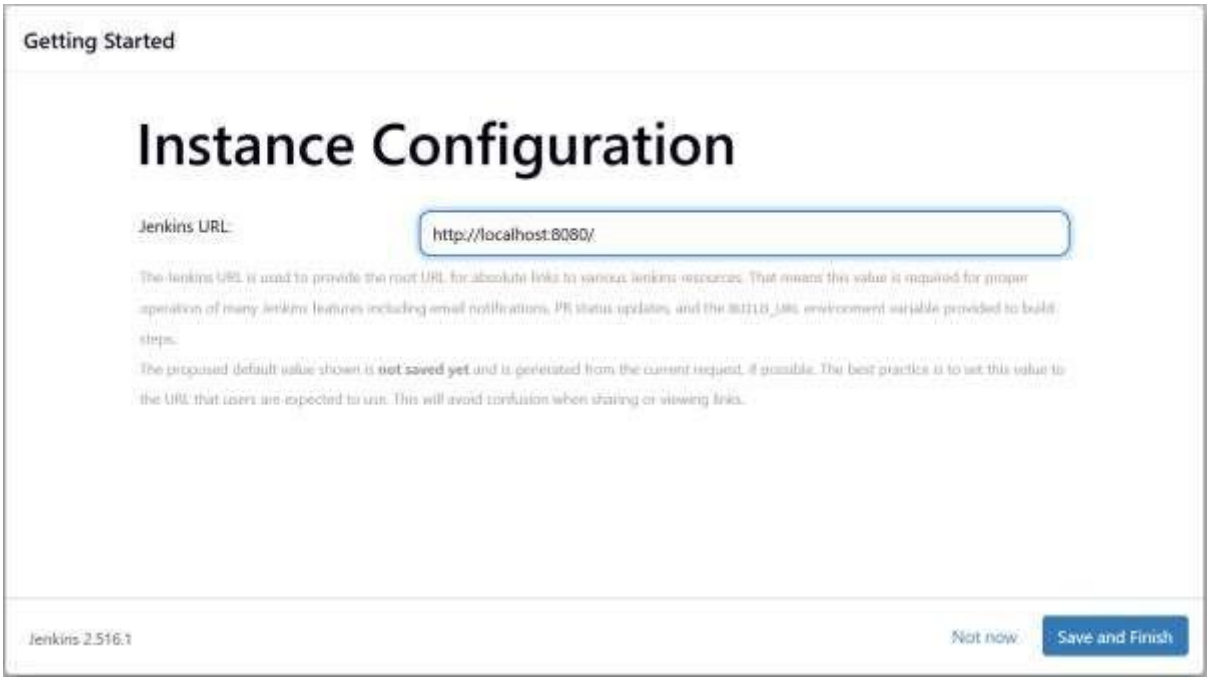
15. Open Initial Adimin Password using notepad copy the password and paste in the above Adminstarte password. And click on continue.

16. Click on install suggested plugins.

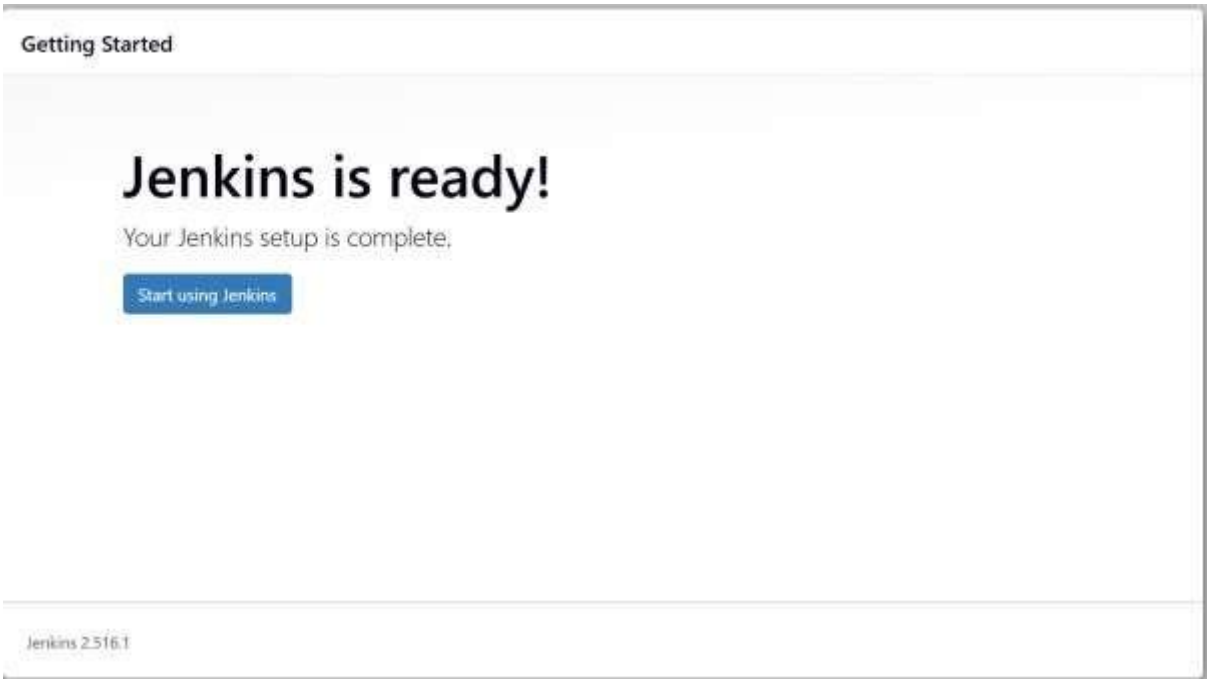
17. Then create first admin user by giving username and password

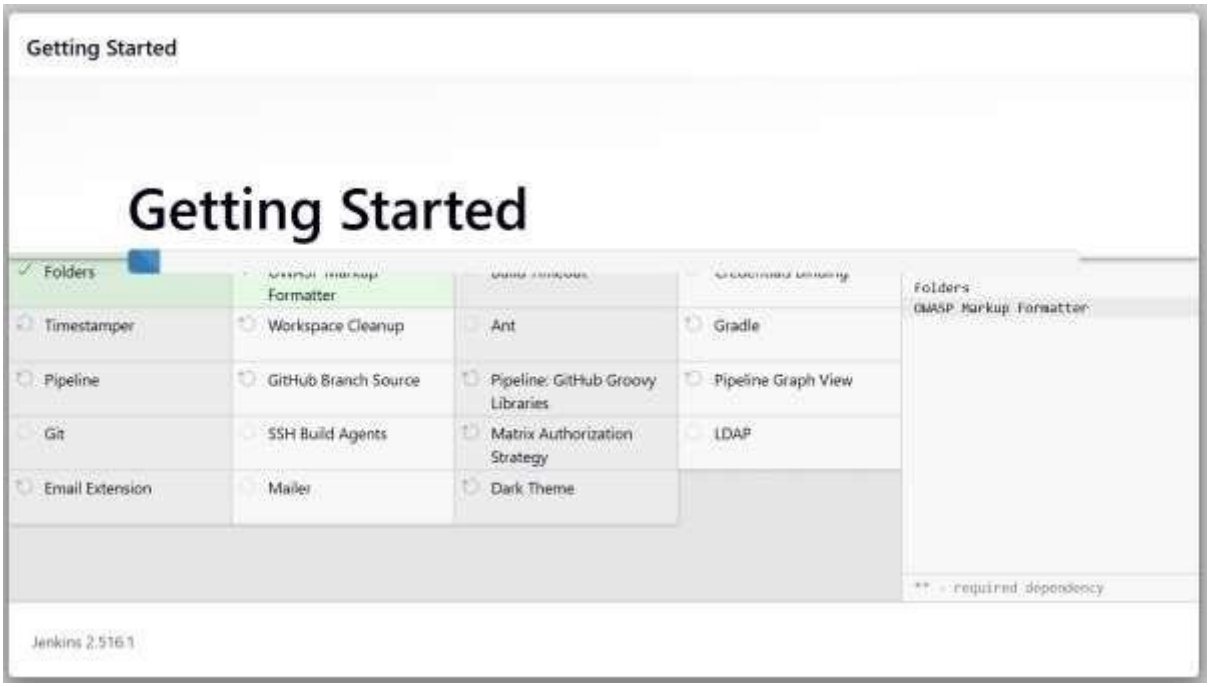
18. click on save and continue.

19. Itgives Url, jusrt click on save and finish



20. Click on start using Jenkins. Redirects to the Jenkins page.

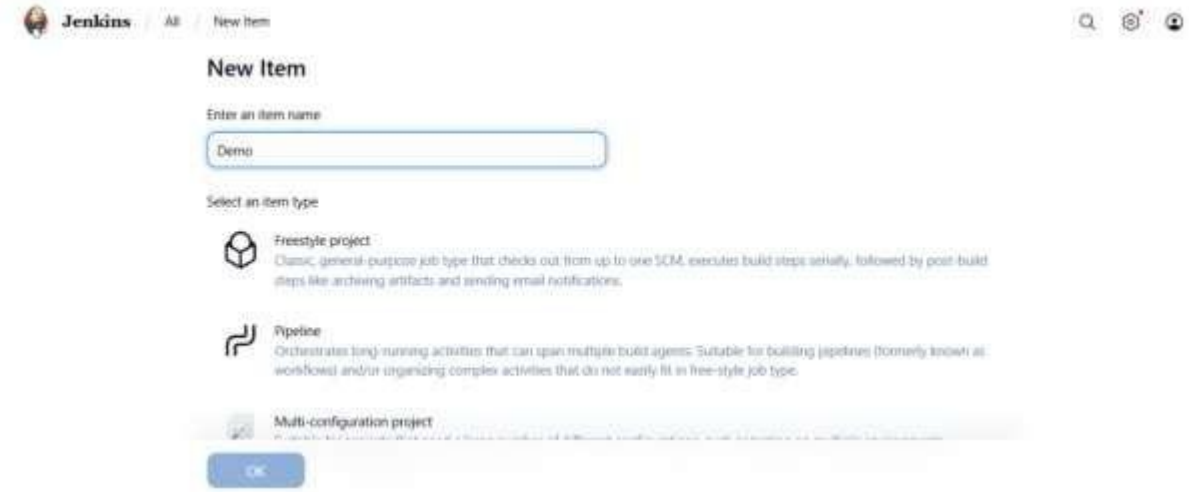




Steps to Create a Job:

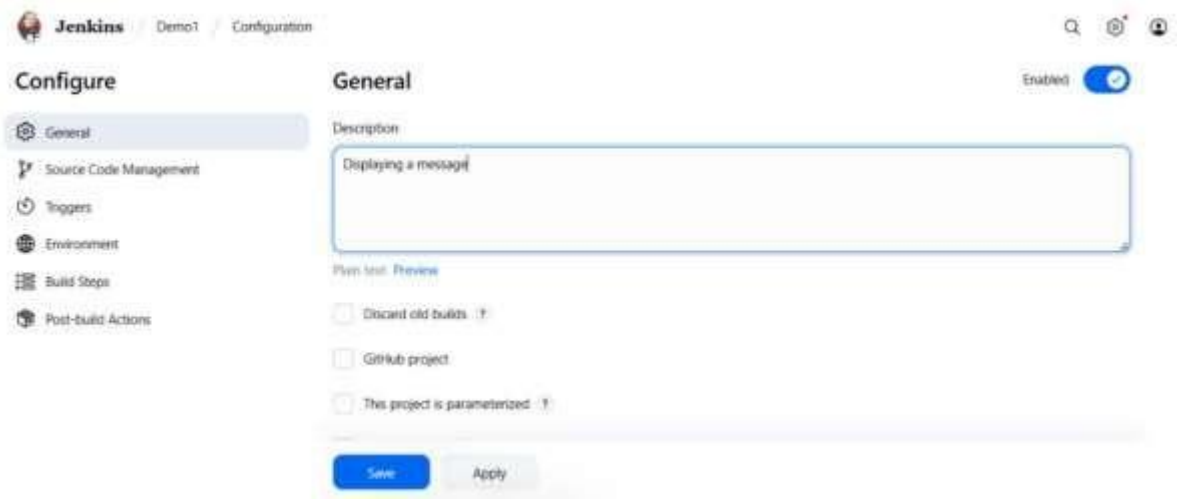
1. Click on create new item

2. Give the name

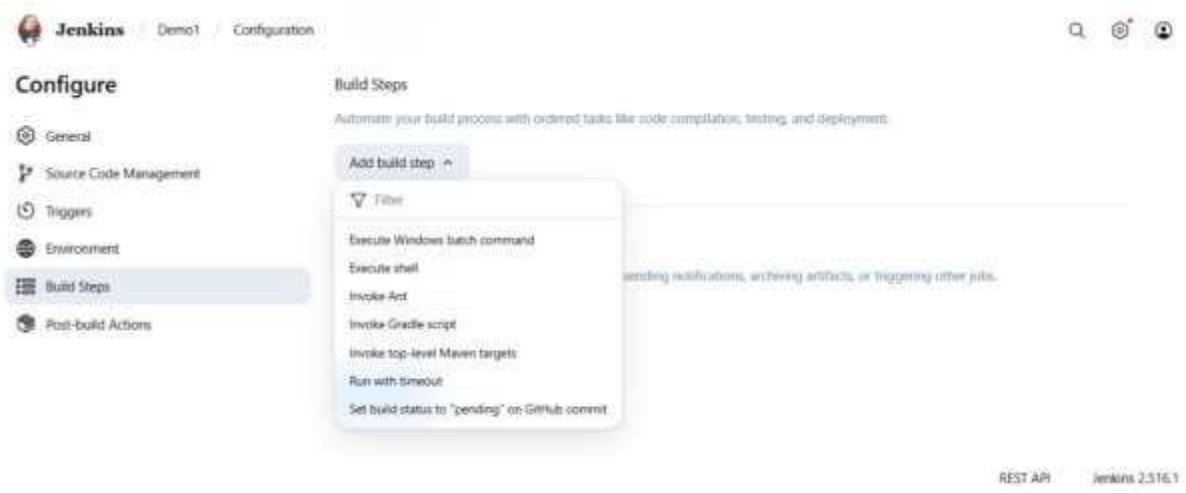


3. Select the free style project and click on ok.

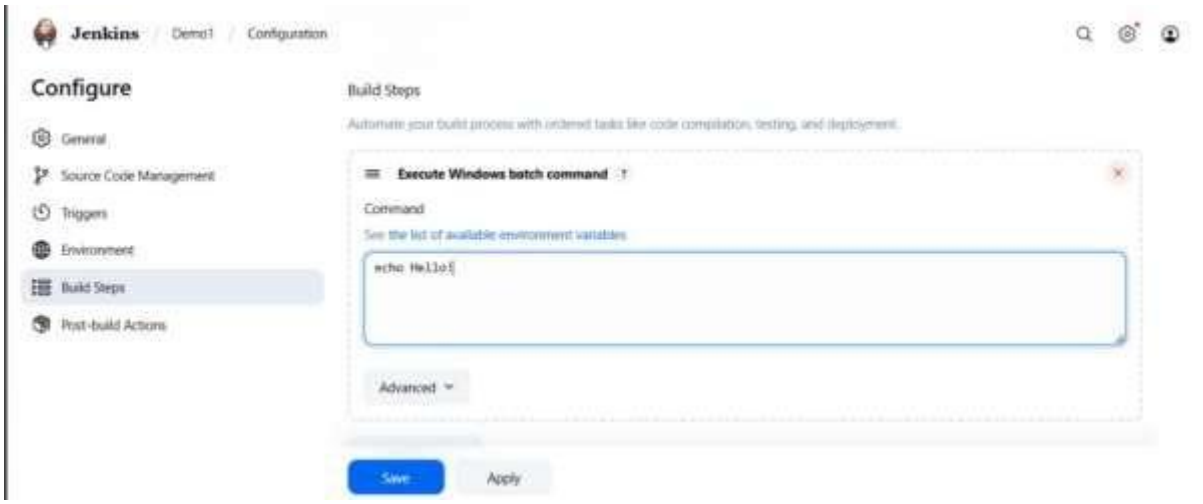
4. In the configuration give general description



5. Scroll Down go to build steps : execute windows batch commands



6. click save

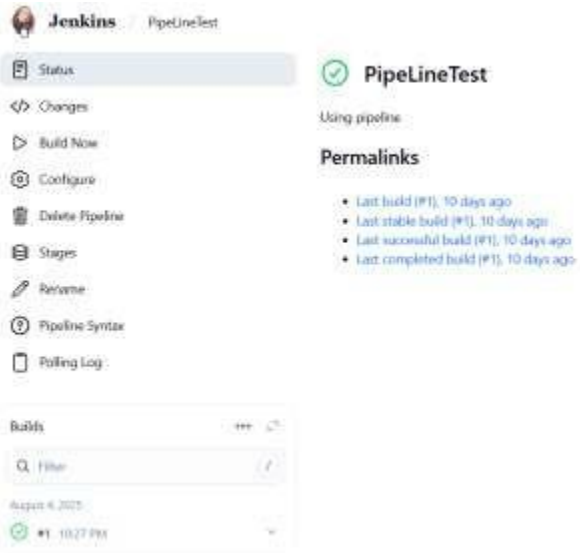


7. Then build now


Jenkins
pipe
#4
Console Output
Status
Changes
Console Output
Edit Build Information
Delete build '#4'
Timings
Git Build Data
Previous Build
Console Output
Download
Copy
View as plain text
Started by user Kaveri Mekala
Running as SYSTEM
Building in workspace C:\Users\HP\.jenkins\workspace\pipe
The recommended git tool is: NONE
No credentials specified
> git.exe rev-parse --resolve-git-dir C:\Users\HP\.jenkins\workspace\pipe\.git # timeout=10
Fetching changes from the remote Git repository
> git.exe config remote.origin.url https://github.com/MekalaKaveri18/TestPipe.git # timeout=10
Fetching upstream changes from https://github.com/MekalaKaveri18/TestPipe.git
> git.exe --version # timeout=10
> git --version # 'git version 2.49.0.windows.1'
> git.exe fetch --tags --force --progress -- https://github.com/MekalaKaveri18/TestPipe.git +refs/heads/*:refs/remotes/origin/* # timeout=10
> git.exe rev-parse "refs/remotes/origin/main^{commit}" # timeout=10
Checking out Revision 2bfa9931a5918897E120ccbea4ac956fdfa05370 (refs/remotes/origin/main)
> git.exe config core.sparsecheckout # timeout=10
> git.exe checkout -f 2bfa9931a5918897E120ccbea4ac956fdfa05370 # timeout=10
Commit message: "Update Jenkinsfile 1"
> git.exe rev-list --no-walk 704ab8664a047a6612c75c1842d1c6878c324887 # timeout=10
[pipe] $ cmd /c call C:\Users\HP\AppData\Local\Temp\jenkins15884836680018182150.bat
C:\Users\HP\.jenkins\workspace\pipe>javac Hello.java
C:\Users\HP\.jenkins\workspace\pipe>java Hello
Hello Devops World!


Jenkins
Console Output
Download
Copy
View as plain text
Started by an SCM change
Resume disabled by user, switching to high-performance, low-durability mode.
Hello World
Finished: SUCCESS
REST API


Stages
Start
Hello
End
#1


Jenkins
PipeLineTest
Status
Changes
Build Now
Configure
Delete Pipeline
Stages
Rename
Pipeline Syntax
Polling Log
Builds
PipeLineTest
Using pipeline
Permalinks
Last build (#1), 10 days ago
Last stable build (#1), 10 days ago
Last successful build (#1), 10 days ago
Last completed build (#1), 10 days ago

Started by an SCM change

Obtained Jenkinsfile from git https://github.com/MekalaKaveri18/TestingFact.git

[Pipeline] Start of Pipeline

[Pipeline] node

Running on Jenkins

in C:\Users\HP\.jenkins\workspace\TestFact

[Pipeline] {

[Pipeline] stage

[Pipeline] { (Declarative: Checkout SCM)

[Pipeline] checkout

Selected Git installation does not exist. Using Default

The recommended git tool is: NONE

No credentials specified

> git.exe rev-parse --resolve-git-dir C:\Users\HP\.jenkins\workspace\TestFact\.git # timeout=10

Fetching changes from the remote Git repository

> git.exe config remote.origin.url https://github.com/MekalaKaveri18/TestingFact.git # timeout=10

Fetching upstream changes from https://github.com/MekalaKaveri18/TestingFact.git

> git.exe --version # timeout=10

> git --version # 'git version 2.49.0.windows.1'

> git.exe fetch --tags --force --progress -- https://github.com/MekalaKaveri18/TestingFact.git +refs/heads/*:refs/remotes/origin/* # timeout=10

> git.exe rev-parse "refs/remotes/origin/main^{commit}" # timeout=10

Checking out Revision 406cd55932d788ab030098f1b29d73ac1b88a215 (refs/remotes/origin/main)

> git.exe config core.sparsecheckout # timeout=10

> git.exe checkout -f 406cd55932d788ab030098f1b29d73ac1b88a215 # timeout=10

Commit message: "correct manifest.txt"

> git.exe rev-list --no-walk 0b16e1914be29e74b483e18bce395bd1c3f207e8 # timeout=10

[Pipeline] }

[Pipeline] // stage

[Pipeline] withEnv

[Pipeline] {

[Pipeline] stage

[Pipeline] { (compile)

[Pipeline] bat

C:\Users\HP\.jenkins\workspace\TestFact>javac Factorial.java TestFact.java

[Pipeline] }

[Pipeline] // stage

[Pipeline] stage

[Pipeline] { (Test)

[Pipeline] bat

C:\Users\HP\.jenkins\workspace\TestFact>java TestFact.java

All tests passed

[Pipeline] }

[Pipeline] // stage

[Pipeline] stage

[Pipeline] { (Run)

[Pipeline] bat

C:\Users\HP\.jenkins\workspace\TestFact>java Factorial.java

Factorial of5is120

[Pipeline] }

[Pipeline] // stage

[Pipeline] stage

[Pipeline] { (Package JAR)

[Pipeline] bat

C:\Users\HP\.jenkins\workspace\TestFact>jar cfm factorial.jar manifest.txt Factorial.class

[Pipeline] }

[Pipeline] // stage

[Pipeline] stage

[Pipeline] { (Archive JAR)

[Pipeline] archiveArtifacts

Archiving artifacts

[Pipeline] }

[Pipeline] // stage

[Pipeline] stage

[Pipeline] { (Declarative: Post Actions)

[Pipeline] echo

Build,test,run and jar creation successful and artifact is ready!

[Pipeline] }

[Pipeline] // stage

[Pipeline] }

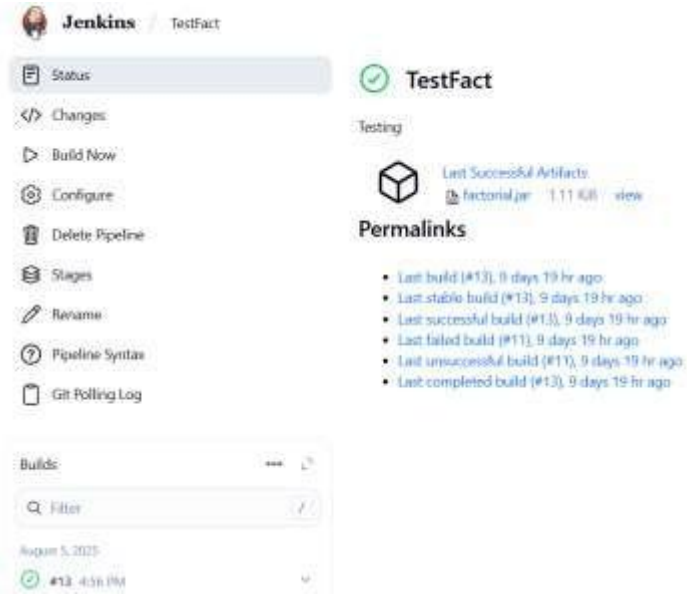
[Pipeline] // withEnv

[Pipeline] }

[Pipeline] // node

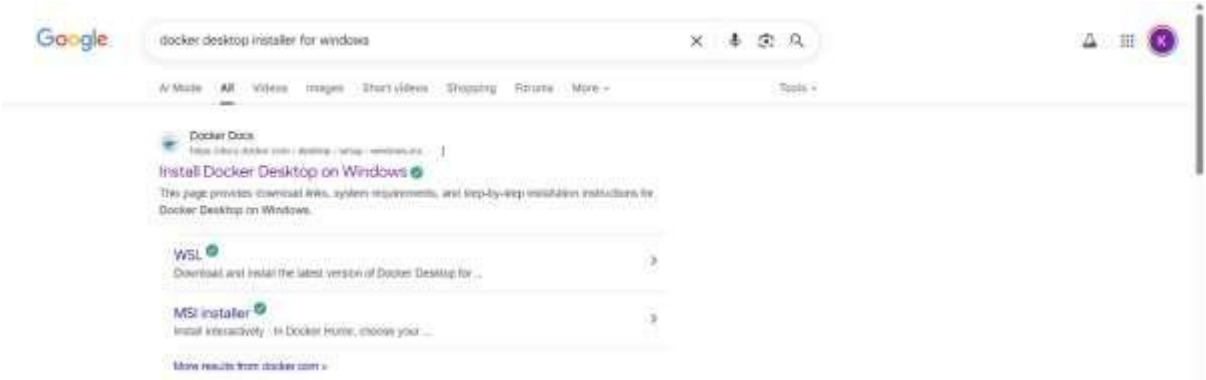
[Pipeline] End of Pipeline

Finished: SUCCESS



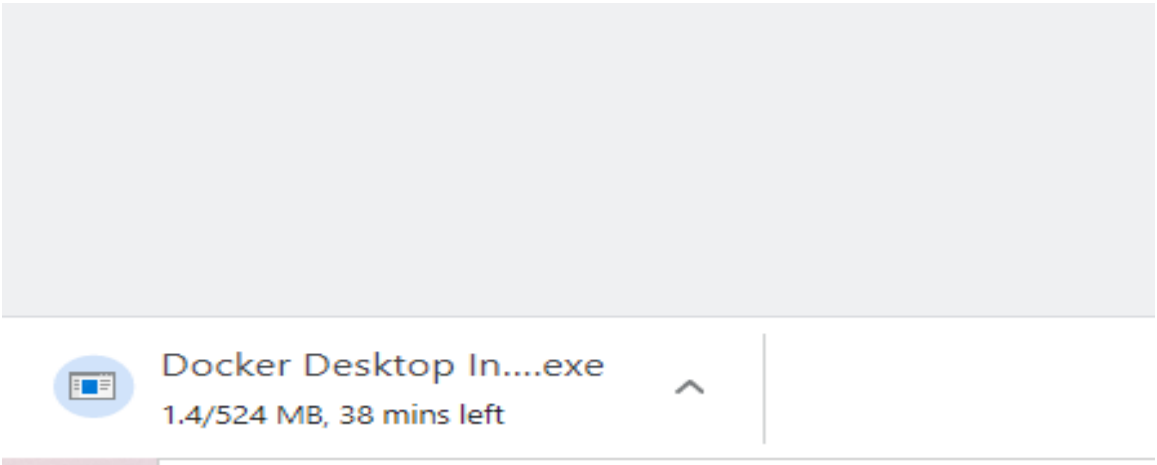
Steps to Install Docker Desktop:

1) Open the web browser & type dockerdocs.com

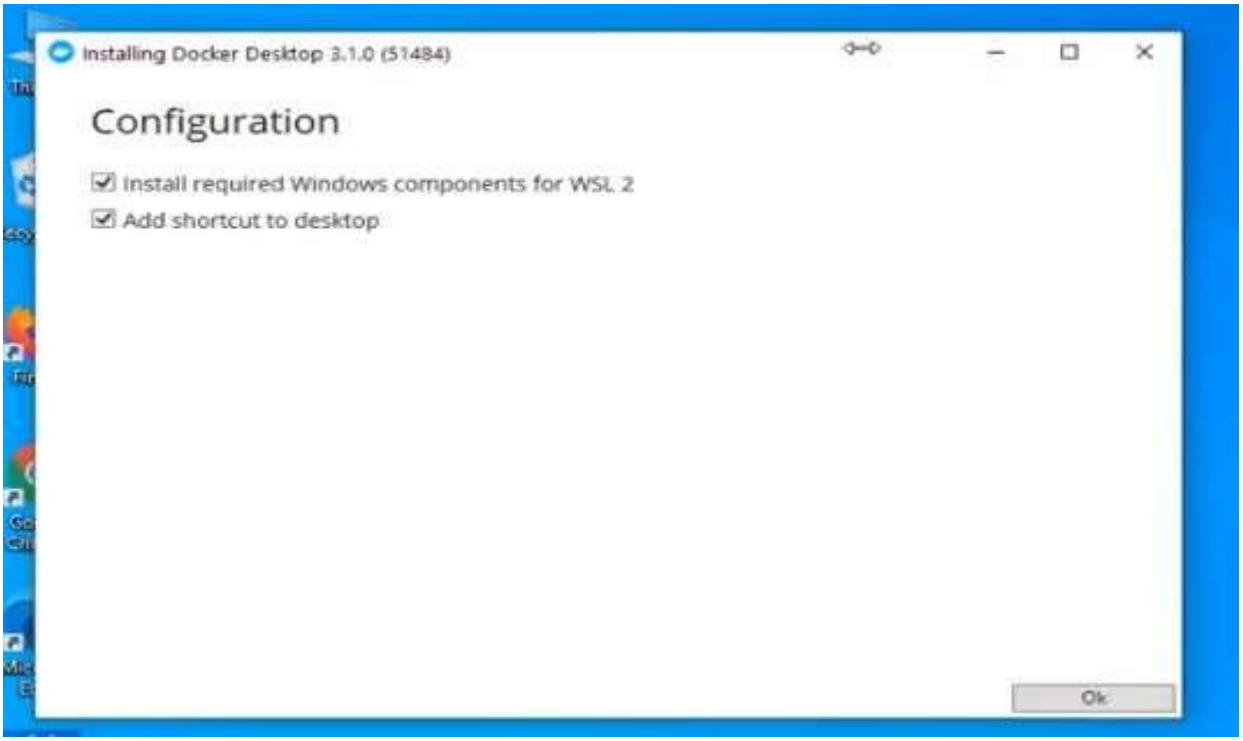


2) Click on getdocker, download docker for windows with latest version.

3)The download will begin automatically .The duration will depend on your internet speed.

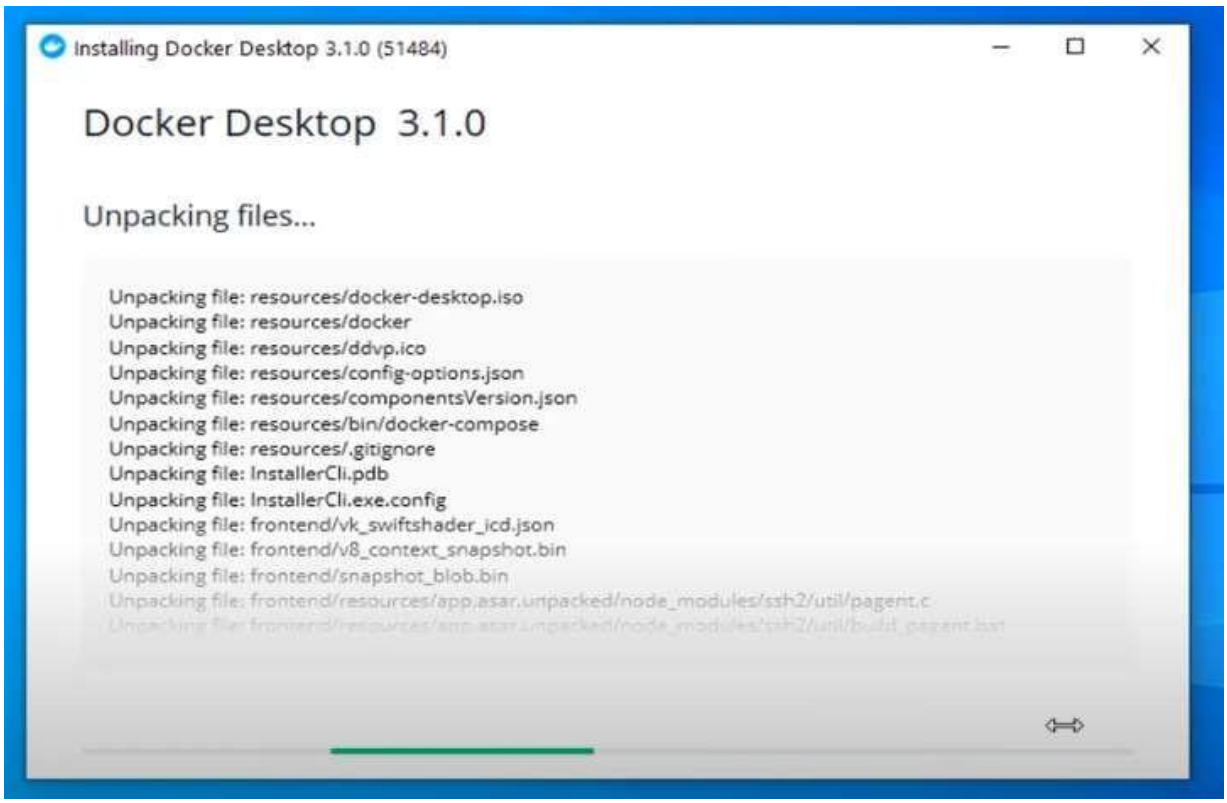


4)After installation open Docker Desktop click continue.

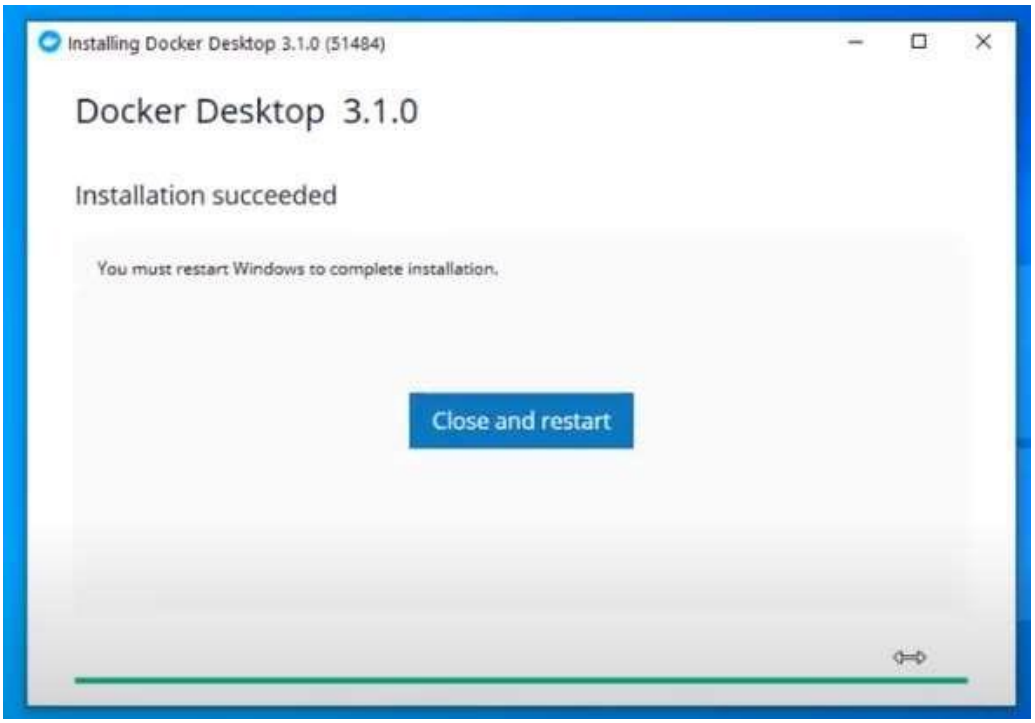


5)Docker Desktop will start after accepting terms and Conditions.

6) After clicking ok the installation will start.



7) Then click close and Restart the system.



8)Restart pc to install wsl2 or update using wsl—update

9) Now the docker will open where we can see images, containers, volumes, dockerhub etc…

Docker Commands:

docker -- version: Shows the Docker client and server versions.

docker ps: Lists all running containers.

docker ps -a: Usedto list all containers (running and stopped).

docker run <image_name>: Creates and runs a new container from a specified image.

docker pull <image_name>: Pulls an image from a registry (e.g., Docker Hub).

docker images: Lists all local Docker images.

docker rmi <image_id_or_name>: Removes a local Docker image.

docker rm <container_id_or_name>: Removes a stopped container.

docker rm -f <container_id_or_name> to force removal of a running container.

```
PS C:\Users\HP> docker --version
Docker version 28.3.2, build 578ccf6

PS C:\Users\HP> docker ps
CONTAINER ID   IMAGE                  COMMAND
CREATED       STATUS        PORTS
                NAMES
b6a70903da0b   nginx:1.27.0-alpine    "/docker-entrypoint.…"   3
months ago   Up 41 seconds   0.0.0.0:4001-4002->4001-4002/tcp,
80/tcp, 0.0.0.0:8980->8980/tcp   algokit_sandbox_proxy

PS C:\Users\HP> docker ps -a
CONTAINER ID   IMAGE                  COMMAND
CREATED       STATUS              PORTS                NAMES
unruffled_lamport   09b9437b7301   postgres:16-alpine   "docker-
entrypoint.s…"   3 days ago   Exited (1) 3 days ago
reverent_blackwell

PS C:\Users\HP> docker pull hello-world
Using default tag: latest
latest: Pulling from library/hello-world
e6590344b1a5: Pull complete
Digest:
sha256:ec153840d1e635ac434fab5e377081f17e0e15afab27beb3f726
c3265039cfff
Status: Downloaded newer image for hello-world:latest
```

docker.io/library/hello-world:latest

PS C:\Users\HP> docker run hello-world

Hello from Docker!

This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:

1. The Docker client contacted the Docker daemon.

2. The Docker daemon pulled the "hello-world" image from the Docker Hub.

(amd64)

3. The Docker daemon created a new container from that image which runs the

executable that produces the output you are currently reading.

4. The Docker daemon streamed that output to the Docker client, which sent it

to your terminal.

To try something more ambitious, you can run an Ubuntu container with:

$ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:

https://hub.docker.com/

For more examples and ideas, visit:

https://docs.docker.com/get-started/

PS C:\Users\HP> docker ps-a

docker: unknown command: docker ps-a

Run 'docker --help' for more information

PS C:\Users\HP> docker ps -all

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

6eb8e54b89ab hello-world "/hello" 4 minutes ago Exited (0) 4 minutes ago unruffled_lamport

PS C:\Users\HP> docker ps -a

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

6eb8e54b89ab hello-world "/hello" 5 minutes ago Exited (0) 5 minutes ago unruffled_lamport

09b9437b7301 postgres:16-alpine "docker-entrypoint.s…" 34 minutes ago Exited (1) 34 minutes ago reverent_blackwell

b6a70903da0b nginx:1.27.0-alpine "/docker-entrypoint.…" 3 months ago Up 24 minutes 0.0.0.0:4001-4002->4001-4002/tcp, 80/tcp, 0.0.0.0:8980->8980/tcp algokit_sandbox_proxy

fc271fbb4869 algorand/indexer:latest "docker-entrypoint.s…" 3 months ago Up 25 seconds algokit_sandbox_indexer

8f9e7ac2c3d3 algorand/conduit:latest "docker-entrypoint.sh" 3 months ago Up About a minute algokit_sandbox_conduit

550651d39aa6 postgres:16-alpine "docker-entrypoint.s…" 3 months ago Exited (255) 8 days ago 0.0.0.0:5443->5432/tcp algokit_sandbox_postgres

ae65eeb4e3e8 algorand/algod:latest "/node/run/run.sh" 3 months ago Exited (255) 8 days ago 4160/tcp, 8080/tcp, 9100/tcp, 0.0.0.0:9392->9392/tcp, 127.0.0.1:32768->7833/tcp algokit_sandbox_algod

PS C:\Users\HP> docker images

REPOSITORY TAG IMAGE ID CREATED SIZE

postgres 16-alpine 7c8c4bf31976 3 months ago 394MB

algorand/algod latest 8f6395c74402 3 months ago 893MB

<none> <none> 3b057e1c2c6d 5 months ago 394MB

<none> <none> 75680de7201f 5 months ago 893MB

hello-world latest ec153840d1e6 6 months ago 20.4kB

algorand/indexer latest b913032bcfb7 6 months ago 159MB

algorand/conduit latest 1ab136352864 7 months ago 160MB

nginx 1.27.0-alpine 208b70eefac1 13 months ago 66.9MB

PS C:\Users\HP> docker rm 3aa84471950d8793a333a888fef92c7a7841c4a6d286e9c7bec27822461875a2

3aa84471950d8793a333a888fef92c7a7841c4a6d286e9c7bec27822461875a2

PS C:\Users\HP> docker rmi hello-world

Error response from daemon: conflict: unable to delete hello-world:latest (must be forced) - container 6eb8e54b89ab is using its referenced image ec153840d1e6

PS C:\Users\HP> docker rm hello-world:latest

Error response from daemon: No such container: hello-world:latest

PS C:\Users\HP> docker rmi hello-world:latest

Error response from daemon: conflict: unable to delete hello-world:latest (must be forced) - container 6eb8e54b89ab is using its referenced image ec153840d1e6

PS C:\Users\HP> docker rm b9c2cea9e73765ee5bab915427ead91cc9541d153cf281b8d1e395f1b965227a

b9c2cea9e73765ee5bab915427ead91cc9541d153cf281b8d1e395f1b965227a

Week-7

Develop a simple containerized application using Docker (through Jenkins CI/CD pipeline)

Pre-requisites:

- Install JDK and set jdk path in environment variables
- Install Git Client and set git path in environment variables
- Install Docker Desktop
- Install Docker plugin in Jenkins
- Open Jenkins Dashboard**,** then go to Manage Jenkins → Plugins**.**
- Go to Available plugins and search for Docker.
- Select Docker and click Install**.**
- Configure Docker in Jenkins:
- Open Jenkins Dashboard, then go to Manage Jenkins → Tools.
- Scroll down to the Docker Installations section.
- If Docker Desktop is already installed, no additional tool configuration is needed. Click Save.
- Otherwise, click Add Docker, give it a name (for example, Docker), check Install automatically, then click Add Installer and select one of the available installers. Finally, click Save.

Steps:

1. Write a Python web application code for a simple user registration form.
2. Write/Create Dockerfile

```
FROM python:3.9-slim-buster
WORKDIR /app
COPY . /app
RUN pip install --no-cache-dir -r requirements.txt
EXPOSE 5000
CMD ["python","app.py"]
```

3. Write/Create **Jenkinsfile**

```
pipeline {
   agent any
   stages {
      stage('Build') {
         steps {
            echo "Build Docker Image"
            bat "docker build -t mypythonflaskapp ."
```

```
            }
        }
        stage('Run') {
            steps {
                echo "Run application in Docker Container"
                bat "docker rm -f mycontainer || exit 0"
                //forcibly removes the Docker container named mycontainer
                //If the container does not exist, this command will fail and return anerror
                //To avoid this error, exit 0, tells the shell to exit with a success status

                bat "docker run -d -p 5000:5000 --name mycontainer mypythonflaskapp"
                //with -d runs the container in detached mode,
                //meaning it runs in the background, and you get your terminal back
                // immediately.
                //Without –d, app runs in the foreground, terminal shows container logs
                        and is “blocked” by the container process.
            }
        }
    }
    post {
        success {
            echo 'Pipeline completed successfully!'
        }
        failure {
            echo 'Pipeline failed. Please check the logs.'
        }
    }
}
```

4. Write **requirements.txt**

   flask

5. Commit and Check in your project to the SCM repository using git commands.
6. git init
7. git add .
8. git commit -m "Initial commit - Flask app with Docker and Jenkins"
9. git remote add origin < https://github.com/MekalaKaveri18/application.git >
10. git push -u origin main
11. Project folder structure should be as follows:

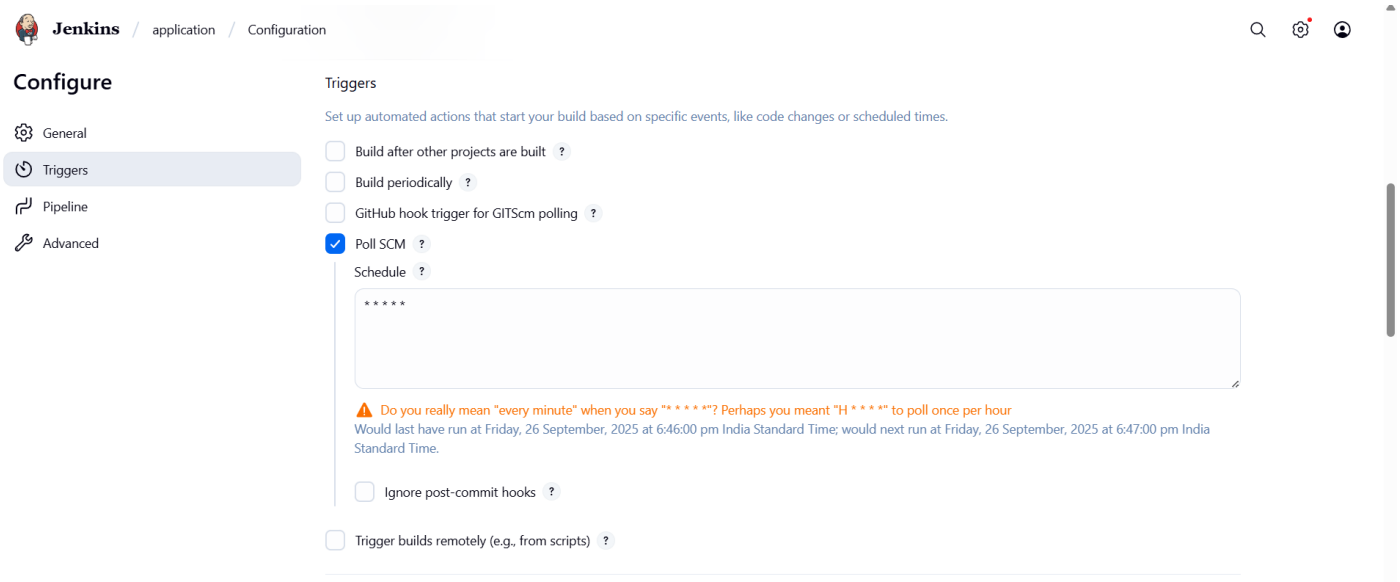
SamplePythonFlaskApp

- app.py

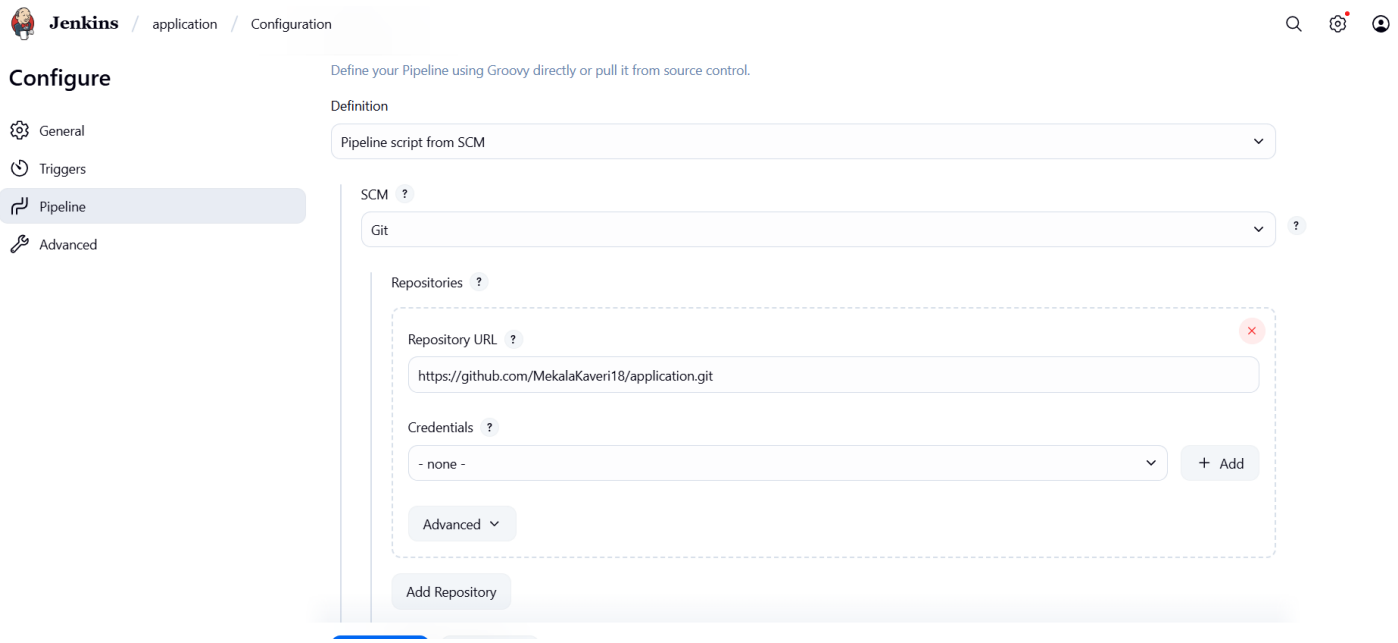
- templates
  - registration.html
  - success.html
- requirements.txt
- Dockerfile
- Jenkinsfile

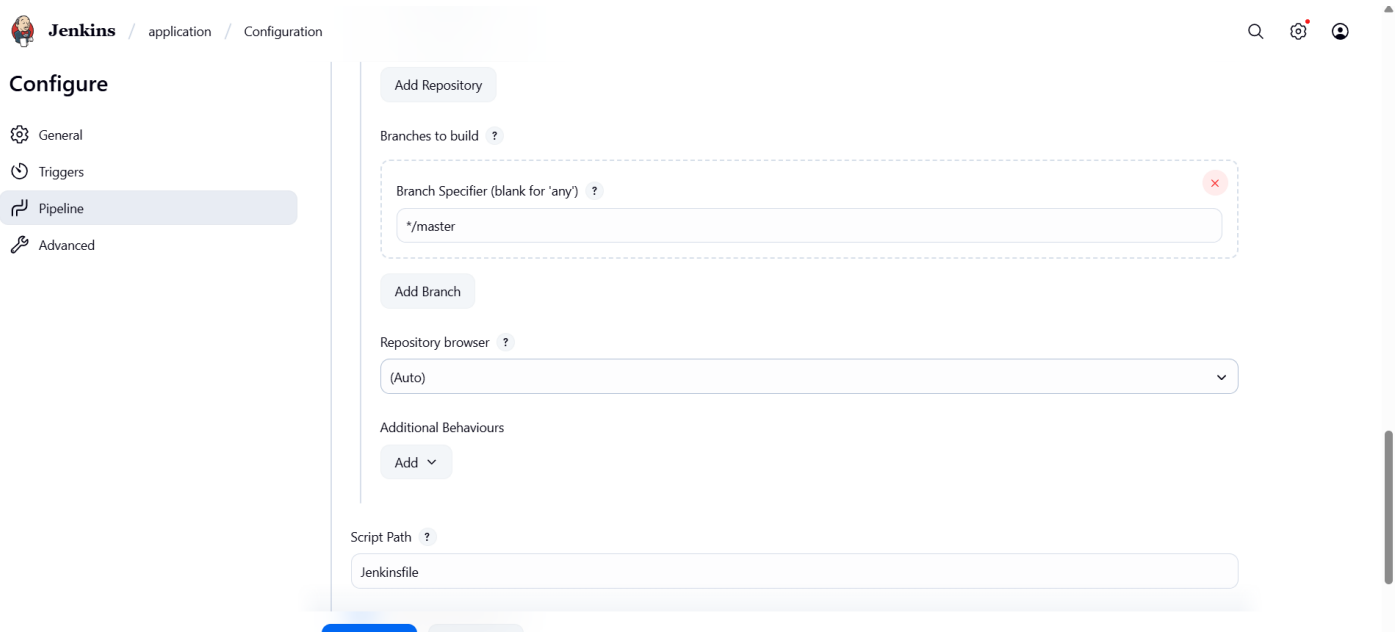
12.Create CI/CD pipeline in Jenkins

Open **Jenkins Dashboard**, then click **New Item**, select **Pipeline**, and click **Ok**.
Configure the pipeline by adding a **description**.
Configure the pipeline
Give description then Check Poll SCM and schedule polling for 5minutes – to automate the process of source code change management and build the application



Configure Pipeline section
Select Define Pipeline Script from SCM, as pipeline script is written in Jenkinsfile which is in SCM.
Select SCM as Git
Specify the git repo url https://github.com/MekalaKaveri18/application.git
Specify the branch as master
Specify the Script Path as Jenkinsfile
Click on Apply/Save – saves the pipeline job after configuration



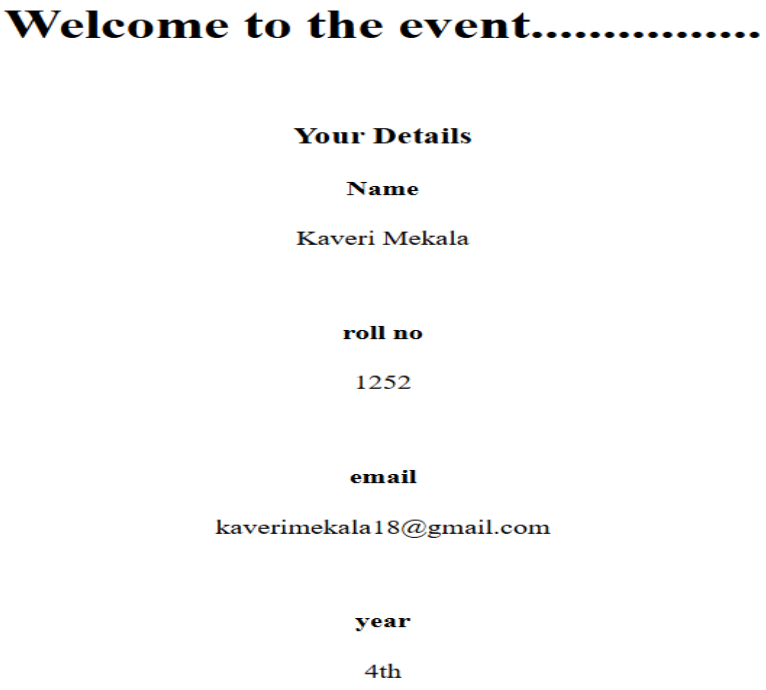


13.Run and Monitor Pipeline Job

- Go to the **Pipeline dashboard** and click **Build Now** to trigger the pipeline manually for the first run.
- Jenkins will then execute the stages defined inside the Jenkinsfile:
- The **Build** stage builds the Docker image.
- The **Run** stage starts the container on port 5000.
- After that, click the build number (for example, #1).
- Then click **Console Output** to view pipeline logs in real time.
- Finally, click **Pipeline Overview** to see the graphical representation of the

stages executed in the pipeline.

- **Output:**



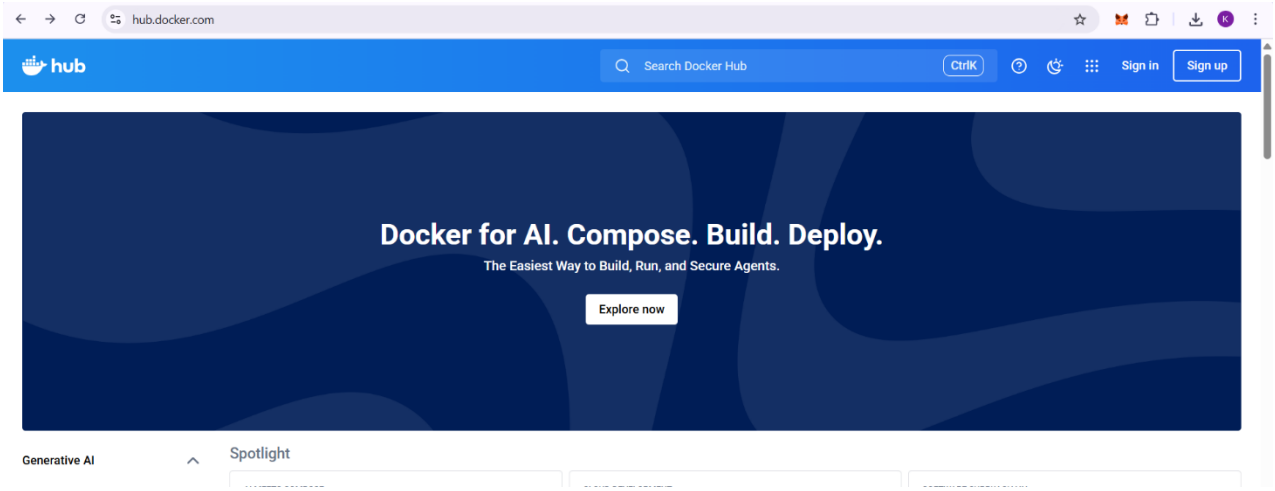
Integrate Kubernetes and Docker

Create a application with app.py, template/form.html, result.html,docker file, requirements.txt

Then push into github and build the docker image using docker build command.

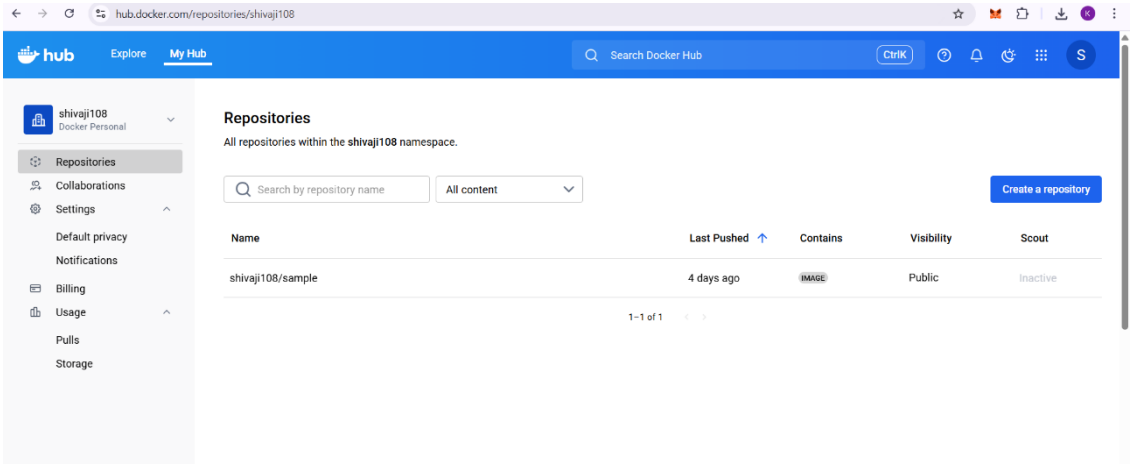
To push the image into the docker hub

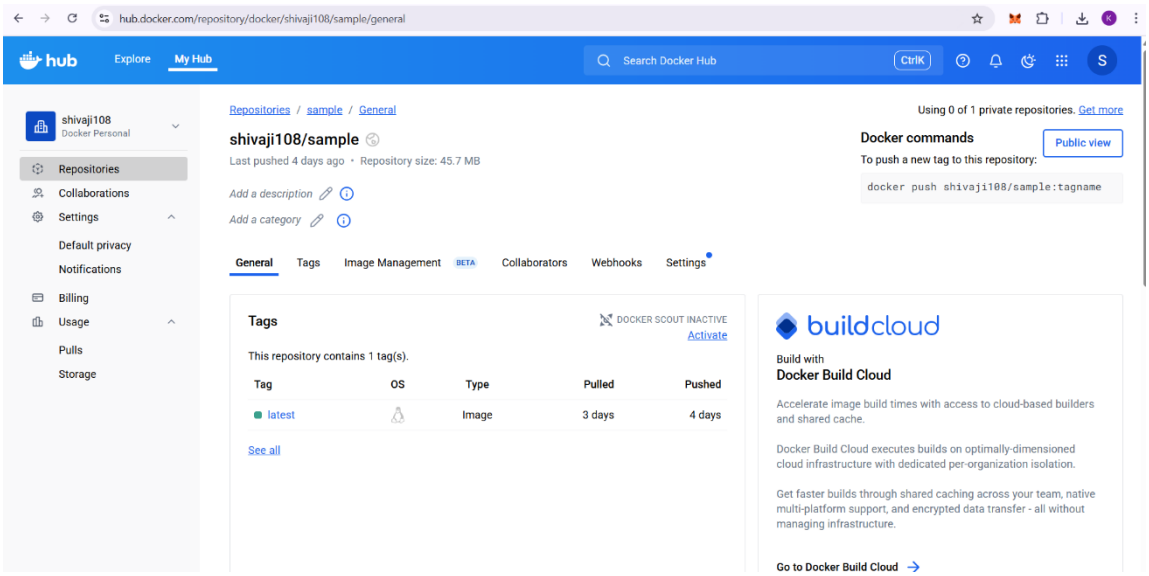
1.Create a login in docker hub(hub.docker.com)



2.After the email verification. Login into the docker hub

3.Click on Repositories then click on create repo then give repo name and click on create.





4.Go to Docker desktop and give the following commands in terminals.
PS C:\Users\HP> docker login
Authenticating with existing credentials... [Username: shivaji108]
i Info → To login with a different account, run 'docker logout' followed by 'docker login'
Login Succeeded

The docker tag command creates a new tag, or alias, that refers to an existing image. To push an image to a registry like Docker Hub, it needs to be tagged with the format <username>/<repository>:<tag>

PS C:\Users\HP> docker tag mypythonflaskapp:latest shivaji108/sample:latest

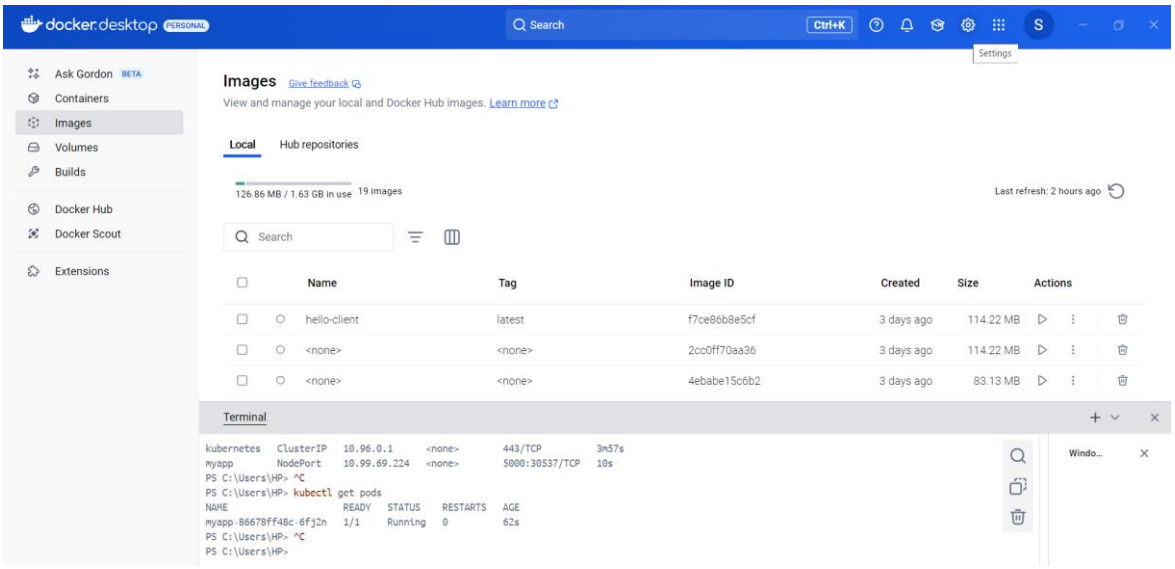
The docker push command uploads a local Docker image to a remote registry, making it accessible to others or to a Kubernetes cluster.

PS C:\Users\HP> docker push shivaji108/sample:latest
The push refers to repository [docker.io/shivaji108/sample]
5bc1815354a5: Pushed
cbf3c05eca57: Pushed
fe33a435e1c2: Pushed
067ea27560c1: Mounted from library/python
7fb1037e08b3: Mounted from library/python
14cbeede8d6e: Mounted from library/python
ae2d55769c5e: Mounted from library/python
e2ef8a51359d: Mounted from library/python
latest: digest:
sha256:8c1713aa5cfd8852feb6ebb76d6b586088bd2a13e261eb0bc1972eb56b027201
size: 1996

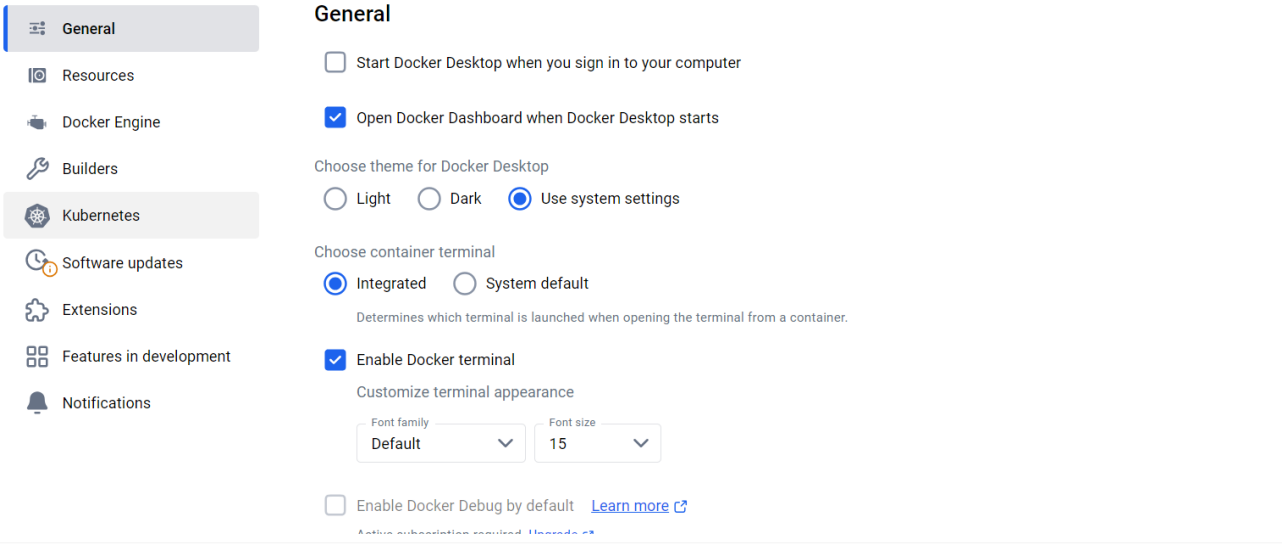
Steps:

Open the Docker Desktop

Click on settings

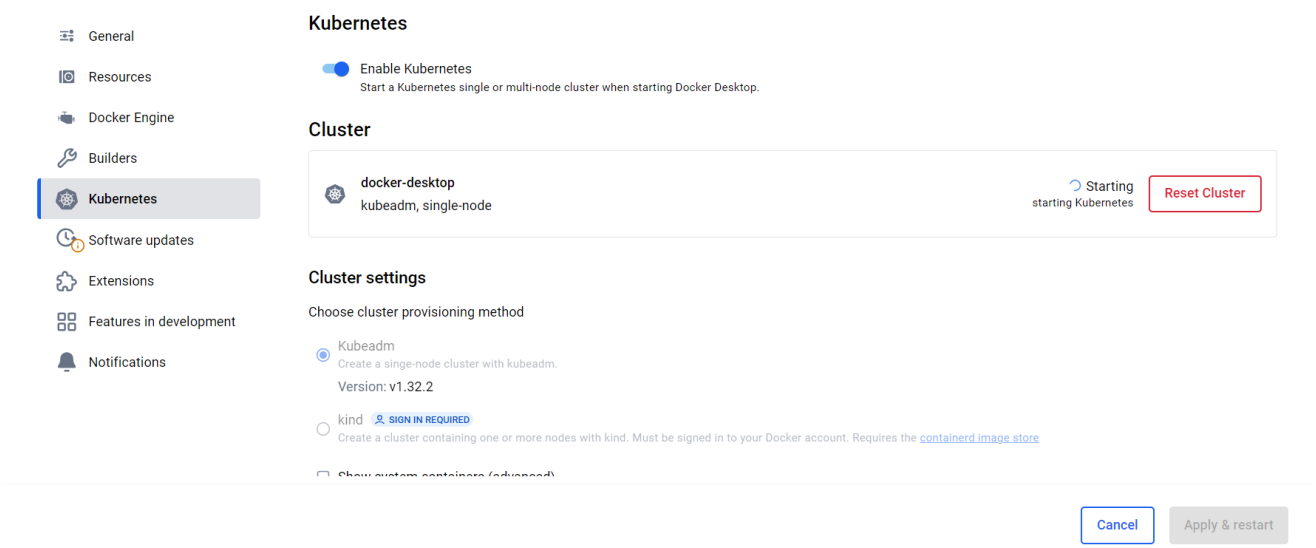


Select the Kubernetes tab



Enable the Kubernetes

Click Apply and Restart



Click install to start the Kubernetes

Docker desktop will set up a single-node Kubernetes cluster
PS C:\Users\HP> **kubectl version**
Client Version: v1.32.2
Kustomize Version: v5.5.0
Server Version: v1.32.2

PS C:\Users\HP> **kubectl get nodes**
NAME STATUS ROLES AGE VERSION
docker-desktop Ready control-plane 6m16s v1.32.2
This command lists all the nodes (worker machines) in your Kubernetes cluster and shows their status

PS C:\Users\HP> **kubectl cluster-info**
Kubernetes control plane is running at https://kubernetes.docker.internal:6443
CoreDNS is running at
https://kubernetes.docker.internal:6443/api/v1/namespaces/kube-system/services/kube-dns:dns/proxy
To further debug and diagnose cluster problems, use 'kubectl cluster-info dump'.
This command displays the address of the Kubernetes control plane and core services like CoreDNS, which is used for service discovery within the cluster.

PS C:\Users\HP> kubectl create deployment myapp --image=shivaji108/sample:latest
deployment.apps/myapp created
This command creates a new Deployment object. A Deployment manages a set of replica Pods and handles updates and self-healing.

PS C:\Users\HP> **kubectl get deployments**

NAME	READY	UP-TO-DATE	AVAILABLE	AGE
myapp	1/1	1	1	53s

This command lists the deployments in the current namespace and shows their status, including how many replicas are ready and available.

PS C:\Users\HP> kubectl get pods

NAME	READY	STATUS	RESTARTS	AGE
myapp-86678ff48c-zwq7r	1/1	Running	0	82s

It list the pods that are running.

PS C:\Users\HP> **kubectl expose deployment myapp --type=NodePort --port=5000**
service/myapp exposed
This command creates a new Service to expose a deployment to network traffic.
--type=NodePort makes the service accessible on a static port on the node's IP address.
--port=5000 specifies the port the service will listen on internally within the cluster.

PS C:\Users\HP> **kubectl get svc**

NAME	TY0PE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
kubernetes	ClusterIP	10.96.0.1	<none>	443/TCP	24m
myapp	NodePort	10.109.196.24	<none>	5000:32094/TCP	12s

An alias for kubectl get services, this command lists all the Services in the cluster.

Output:

Registration Form

Name: Kaveri Mekala

Roll No: 1252

Email: kaverimekala18@gmail.con

Year: 4th Submit

Welcome to the event.....

Your Details

Name

Kaveri Mekala

roll no

1252

email

kaverimekala18@gmail.com

year

4th

PS C:\Users\HP> **kubectl scale deployment myapp --replicas=6**
deployment.apps/myapp scaled
This command changes the desired number of replicas for a deployment, scaling the application up or down

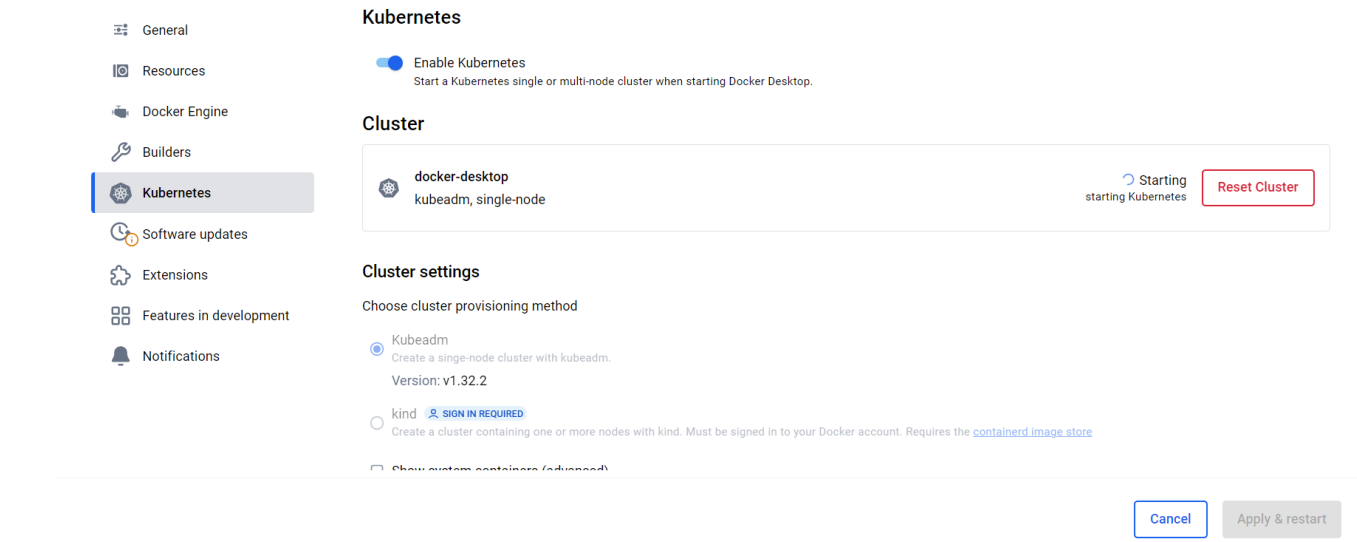
PS C:\Users\HP> **kubectl get pods**

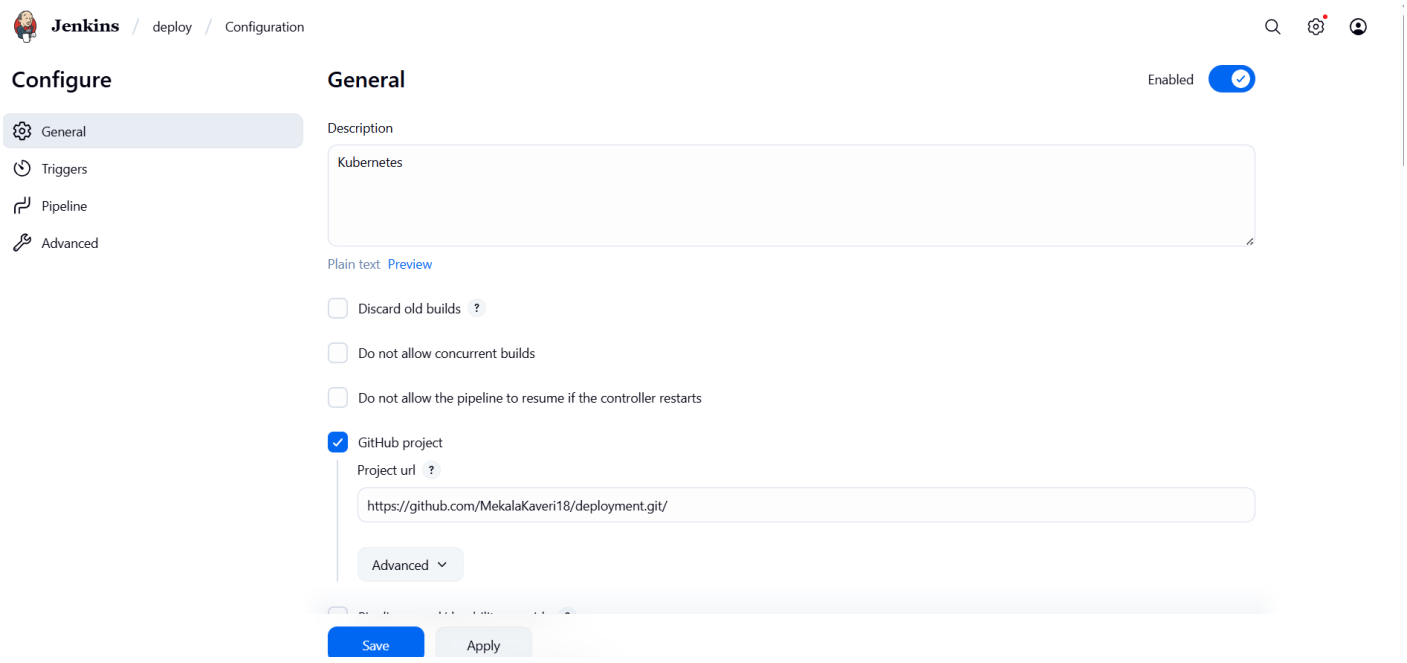
NAME	READY	STATUS	RESTARTS	AGE
myapp-86678ff48c-5lnch	1/1	Running	0	31s
myapp-86678ff48c-b8wd5	1/1	Running	0	31s
myapp-86678ff48c-dw8s9	1/1	Running	0	31s
myapp-86678ff48c-qs4bl	1/1	Running	0	31s
myapp-86678ff48c-vj7kt	1/1	Running	0	31s
myapp-86678ff48c-zwq7r	1/1	Running	0	17m

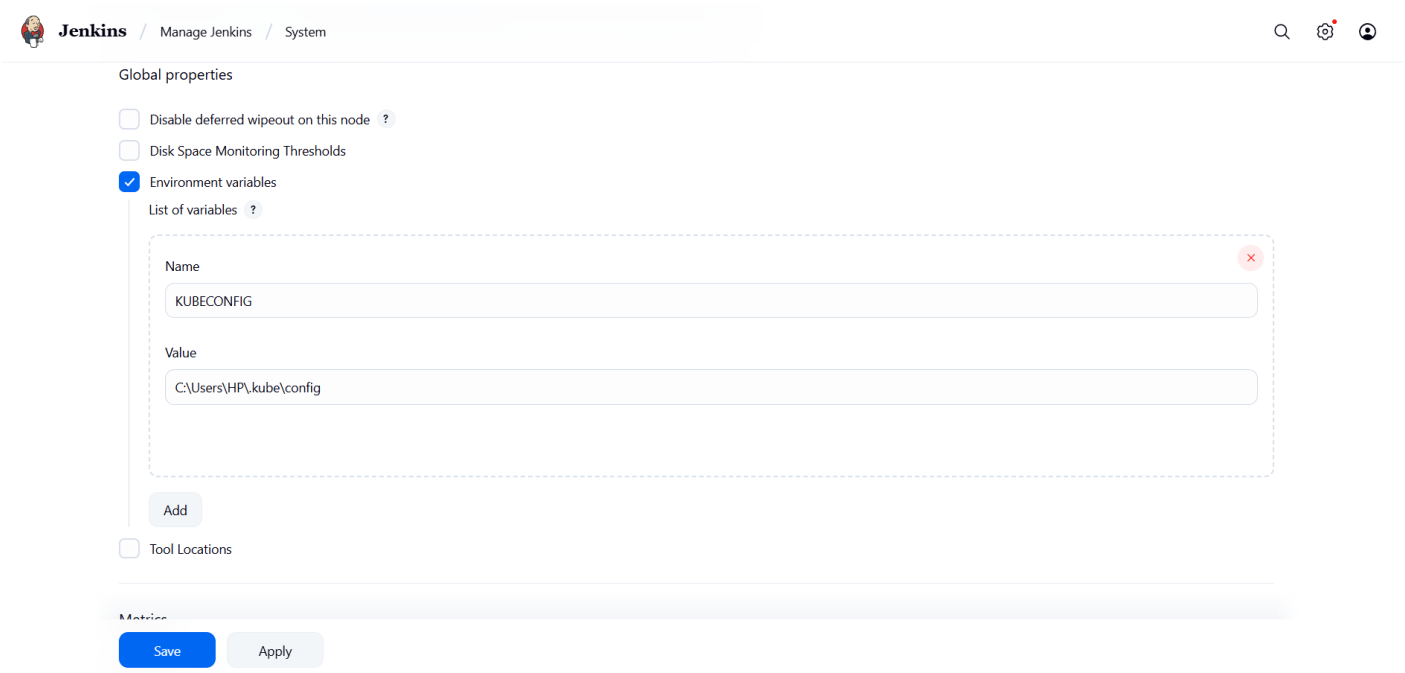
G. Narayanamma Institute Technology and Science (For Women)

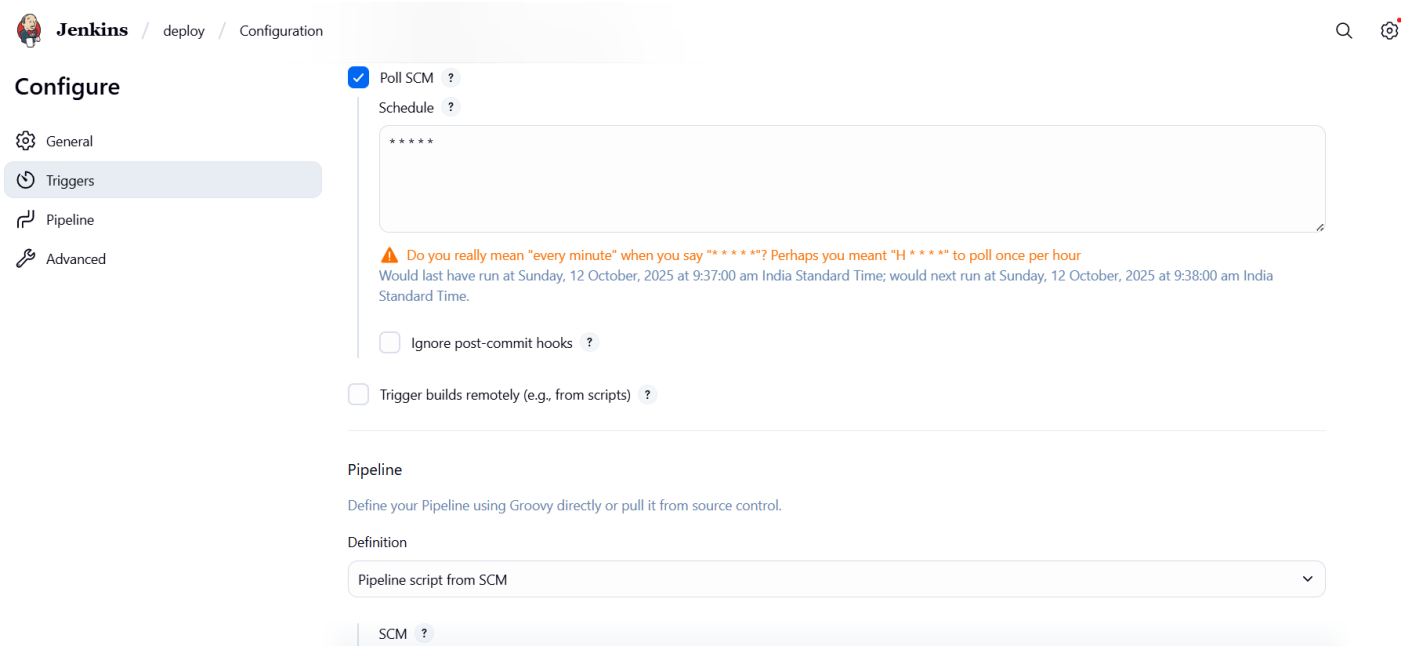
Devops Lab

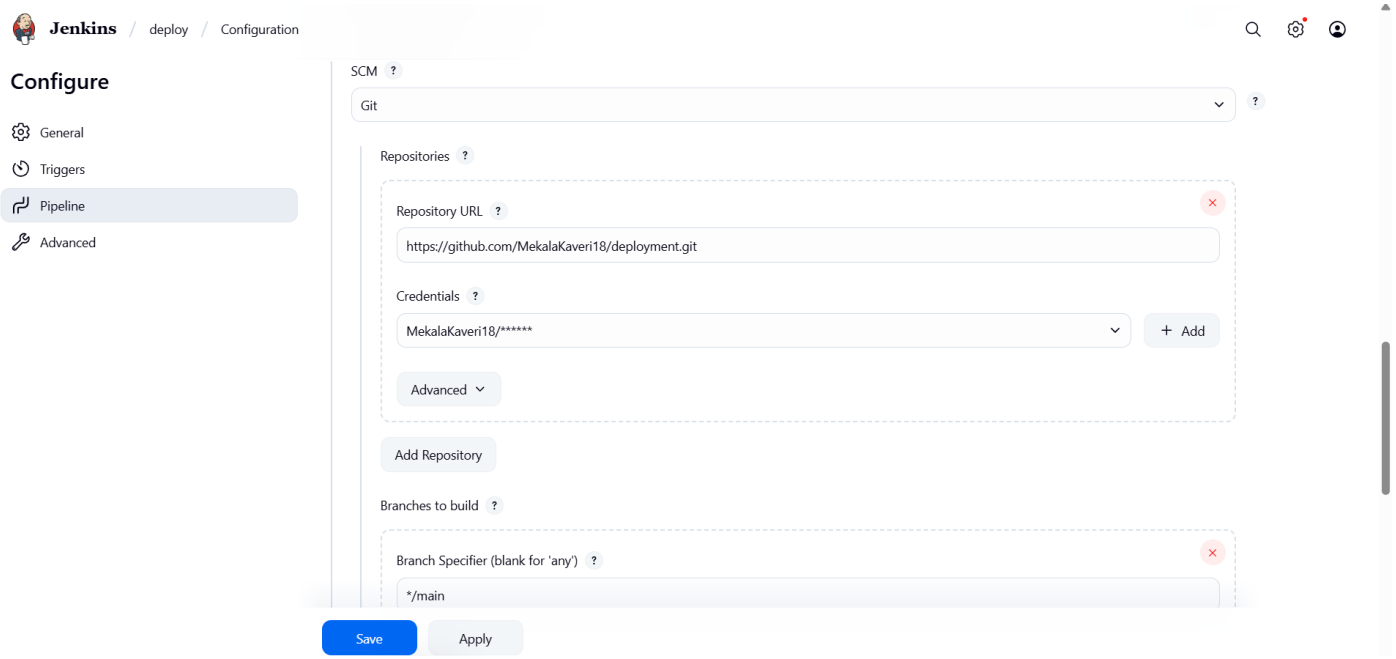
IV-IT-A 2025-2026

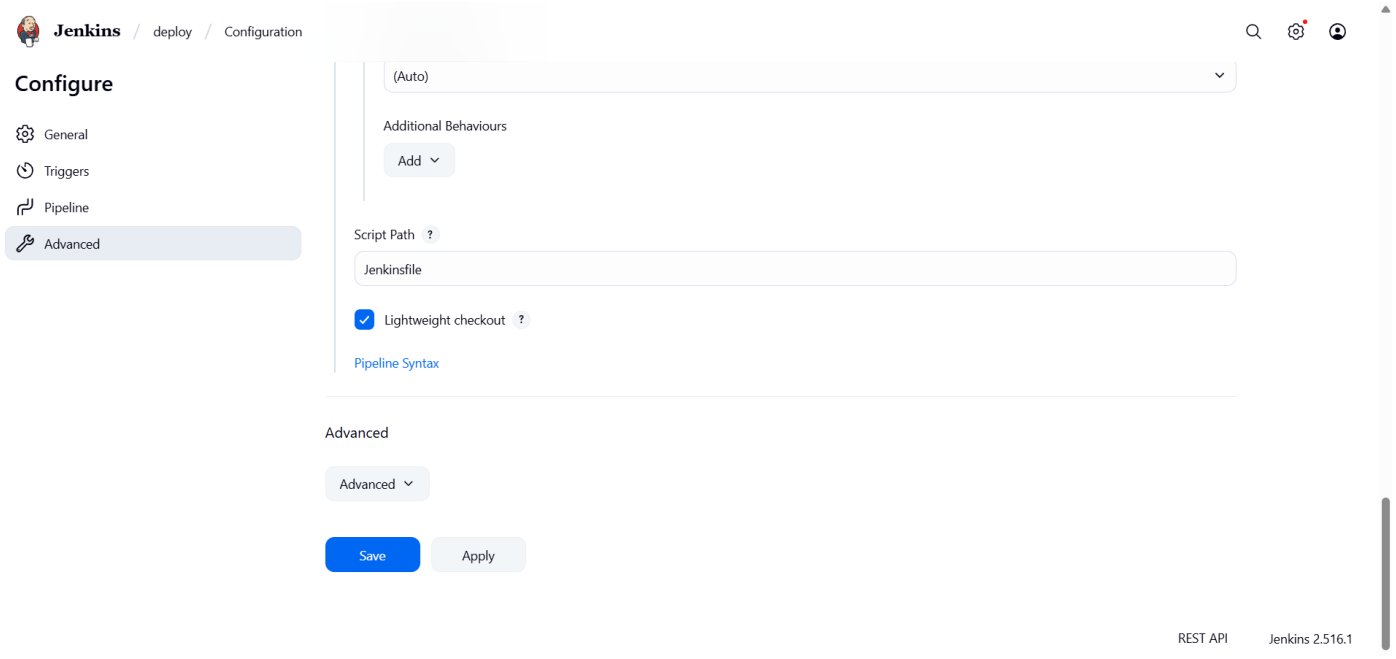





Jenkins / Manage Jenkins / System
Global properties
Disable deferred wipeout on this node
Disk Space Monitoring Thresholds
Environment variables
List of variables
Name
KUBECONFIG
Value
C:\Users\HP\.kube\config
Add
Tool Locations
Save
Apply


Jenkins / deploy / Configuration
Configure
General
Triggers
Pipeline
Advanced
Poll SCM
Schedule
* * * * *
Do you really mean "every minute" when you say "* * * * *"? Perhaps you meant "H * * * *" to poll once per hour
Would last have run at Sunday, 12 October, 2025 at 9:37:00 am India Standard Time; would next run at Sunday, 12 October, 2025 at 9:38:00 am India Standard Time.
Ignore post-commit hooks
Trigger builds remotely (e.g., from scripts)
Pipeline
Define your Pipeline using Groovy directly or pull it from source control.
Definition
Pipeline script from SCM
SCM


Jenkins / deploy / Configuration
Configure
General
Triggers
Pipeline
Advanced
SCM
Git
Repositories
Repository URL
https://github.com/MekalaKaveri18/deployment.git
Credentials
MekalaKaveri18/******
+ Add
Advanced
Add Repository
Branches to build
Branch Specifier (blank for 'any')
*/main
Save
Apply


Jenkins / deploy / Configuration
Configure
General
Triggers
Pipeline
Advanced
(Auto)
Additional Behaviours
Add
Script Path
Jenkinsfile
Lightweight checkout
Pipeline Syntax
Advanced
Advanced
Save
Apply
REST API
Jenkins 2.516.1

Started by an SCM change

Obtained Jenkinsfile from git
https://github.com/MekalaKaveri18/deployment.git

[Pipeline] Start of Pipeline

[Pipeline] node

Running on Jenkins

in C:\ProgramData\Jenkins\.jenkins\workspace\deploy

[Pipeline] {

[Pipeline] stage

[Pipeline] { (Declarative: Checkout SCM)

[Pipeline] checkout

Selected Git installation does not exist. Using Default

The recommended git tool is: NONE

using credential 096e4648-f842-4a49-8853-af67e453d532

> C:\Program Files\Git\bin\git.exe rev-parse --resolve-git-dir
C:\ProgramData\Jenkins\.jenkins\workspace\deploy\.git # timeout=10

Fetching changes from the remote Git repository

> C:\Program Files\Git\bin\git.exe config remote.origin.url
https://github.com/MekalaKaveri18/deployment.git # timeout=10

Fetching upstream changes from
https://github.com/MekalaKaveri18/deployment.git

> C:\Program Files\Git\bin\git.exe --version # timeout=10

> git --version # 'git version 2.51.0.windows.1'

using GIT_ASKPASS to set credentials

> C:\Program Files\Git\bin\git.exe fetch --tags --force --progress --
https://github.com/MekalaKaveri18/deployment.git
+refs/heads/*:refs/remotes/origin/* # timeout=10

> C:\Program Files\Git\bin\git.exe rev-parse
"refs/remotes/origin/main^{commit}" # timeout=10

Checking out Revision c0b4d14c1edb3dcc7328e5132d2a7c21baa9d816 (refs/remotes/origin/main)

> C:\Program Files\Git\bin\git.exe config core.sparsecheckout # timeout=10

> C:\Program Files\Git\bin\git.exe checkout -f c0b4d14c1edb3dcc7328e5132d2a7c21baa9d816 # timeout=10

Commit message: "Update service.yaml"

> C:\Program Files\Git\bin\git.exe rev-list --no-walk 8b90ee8c9b69ebb9b04b006417eeef1cf1954d3d # timeout=10

[Pipeline] }

[Pipeline] // stage

[Pipeline] withEnv

[Pipeline] {

[Pipeline] stage

[Pipeline] { (Build Docker Image)

[Pipeline] echo

Build Docker Image

[Pipeline] bat

C:\ProgramData\Jenkins\.jenkins\workspace\deploy>docker build -t kubedemoapp:v1 .

#0 building with "default" instance using docker driver

#1 [internal] load build definition from Dockerfile

#1 transferring dockerfile: 186B 0.0s done

#1 DONE 0.0s

#2 [internal] load metadata for docker.io/library/python:3.9-slim-buster

#2 DONE 0.4s

#3 [1/4] FROM docker.io/library/python:3.9-slim-buster@sha256:320a7a4250aba4249f458872adecf92eea88dc6abd2d76dc5c0f01cac9b53990

#3 resolve docker.io/library/python:3.9-slim-buster@sha256:320a7a4250aba4249f458872adecf92eea88dc6abd2d76dc5c0f01cac9b53990 0.0s done

#3 DONE 0.1s

#4 [2/4] WORKDIR /app

#4 CACHED

#5 [internal] load .dockerignore

#5 transferring context: 2B done

#5 DONE 0.0s

#6 [internal] load build context

#6 transferring context: 22.62kB 0.1s done

#6 DONE 0.1s

#7 [3/4] COPY . /app

#7 DONE 0.2s

#8 [4/4] RUN pip install --no-cache-dir -r requirements.txt

#8 5.828 Collecting flask

#8 5.925   Downloading flask-3.1.2-py3-none-any.whl (103 kB)

#8 5.967
â”□â”□â”□â”□â”□â”□â”□â”□â”□â”□â”□â”□â”□â”□â”□â”□â”□â”□â”
□â”□â”□â”□â”□â”□â”□â”□â”□â”□â”□â”□â”□â”□â”□â”□â”□â”□â”□â

”□â”□ 103.3/103.3 kB 3.0 MB/s eta 0:00:00

#8 6.322 Collecting markupsafe>=2.1.1

#8 6.352 Downloading markupsafe-3.0.3-cp39-cp39-manylinux2014_x86_64.manylinux_2_17_x86_64.manylinux_2_28_x86_64.whl (20 kB)

#8 6.399 Collecting itsdangerous>=2.2.0

#8 6.426 Downloading itsdangerous-2.2.0-py3-none-any.whl (16 kB)

#8 6.472 Collecting blinker>=1.9.0

#8 6.496 Downloading blinker-1.9.0-py3-none-any.whl (8.5 kB)

#8 6.653 Collecting importlib-metadata>=3.6.0

#8 6.679 Downloading importlib_metadata-8.7.0-py3-none-any.whl (27 kB)

#8 6.747 Collecting jinja2>=3.1.2

#8 6.772 Downloading jinja2-3.1.6-py3-none-any.whl (134 kB)

#8 6.798
â”□â”□â”□â”□â”□â”□â”□â”□â”□â”□â”□â”□â”□â”□â”□â”□â”□â”□â”
□â”□â”□â”□â”□â”□â”□â”□â”□â”□â”□â”□â”□â”□â”□â”□â”□â”□â”□â
”□â”□ 134.9/134.9 kB 7.2 MB/s eta 0:00:00

#8 6.885 Collecting click>=8.1.3

#8 6.912 Downloading click-8.1.8-py3-none-any.whl (98 kB)

#8 6.927
â”□â”□â”□â”□â”□â”□â”□â”□â”□â”□â”□â”□â”□â”□â”□â”□â”□â”□â”
□â”□â”□â”□â”□â”□â”□â”□â”□â”□â”□â”□â”□â”□â”□â”□â”□â”□â”□â
”□â”□â”□ 98.2/98.2 kB 10.7 MB/s eta 0:00:00

#8 7.067 Collecting werkzeug>=3.1.0

#8 7.097 Downloading werkzeug-3.1.3-py3-none-any.whl (224 kB)

#8 7.130
â”□â”□â”□â”□â”□â”□â”□â”□â”□â”□â”□â”□â”□â”□â”□â”□â”□â”□â”
□â”□â”□â”□â”□â”□â”□â”□â”□â”□â”□â”□â”□â”□â”□â”□â”□â”□â”□â
”□â”□ 224.5/224.5 kB 8.5 MB/s eta 0:00:00

#8 7.328 Collecting zipp>=3.20

#8 7.358 Downloading zipp-3.23.0-py3-none-any.whl (10 kB)

#8 7.682 Installing collected packages: zipp, markupsafe, itsdangerous, click, blinker, werkzeug, jinja2, importlib-metadata, flask

#8 9.203 Successfully installed blinker-1.9.0 click-8.1.8 flask-3.1.2 importlib-metadata-8.7.0 itsdangerous-2.2.0 jinja2-3.1.6 markupsafe-3.0.3 werkzeug-3.1.3 zipp-3.23.0

#8 9.205 WARNING: Running pip as the 'root' user can result in broken permissions and conflicting behaviour with the system package manager. It is recommended to use a virtual environment instead: https://pip.pypa.io/warnings/venv

#8 9.455

#8 9.455 [notice] A new release of pip is available: 23.0.1 -> 25.2

#8 9.455 [notice] To update, run: pip install --upgrade pip

#8 DONE 9.7s

#9 exporting to image

#9 exporting layers

#9 exporting layers 1.6s done

#9 exporting manifest sha256:dff46c7c04275d84927a4c5f1de7360130d8721c5480c2953f71650320d2227e 0.0s done

#9 exporting config sha256:c0365ba60813bad9f18f4606941dee044cae338c0c524ea5753b0ab61766707f 0.0s done

#9 exporting attestation manifest sha256:a54542dee728ea59851f1d9ec988bd16cd94d9c52fabc32504cd495d2ad5460b

#9 exporting attestation manifest sha256:a54542dee728ea59851f1d9ec988bd16cd94d9c52fabc32504cd495d2ad5460b 0.3s done

#9 exporting manifest list sha256:5937823b9e70734c266729533d16d94c9119c948da740fe552c0ebadc5

e782cb

#9 exporting manifest list sha256:5937823b9e70734c266729533d16d94c9119c948da740fe552c0ebadc5e782cb 0.1s done

#9 naming to docker.io/library/kubedemoapp:v1 done

#9 unpacking to docker.io/library/kubedemoapp:v1

#9 unpacking to docker.io/library/kubedemoapp:v1 1.3s done

#9 DONE 3.6s

[Pipeline] }

[Pipeline] // stage

[Pipeline] stage

[Pipeline] { (Docker Login)

[Pipeline] bat

C:\ProgramData\Jenkins\.jenkins\workspace\deploy>docker login -u shivaji108 -p Kaveri@1729

WARNING! Using --password via the CLI is insecure. Use --password-stdin.

Login Succeeded

[Pipeline] }

[Pipeline] // stage

[Pipeline] stage

[Pipeline] { (push Docker image to docker hub)

[Pipeline] echo

push Docker image to docker hub

[Pipeline] bat

C:\ProgramData\Jenkins\.jenkins\workspace\deploy>docker tag kubedemoapp:v1 shivaji108/sample:kuberimg1

[Pipeline] bat

C:\ProgramData\Jenkins\.jenkins\workspace\deploy>docker push shivaji108/sample:kuberimg1

The push refers to repository [docker.io/shivaji108/sample]

824416e23423: Waiting

84c8c79126f6: Waiting

8b91b88d5577: Waiting

27e93becc0ce: Waiting

2e1c130fa3ec: Waiting

648e56f234c2: Waiting

93c337da394f: Waiting

3d6c2229f924: Waiting

8d53da260408: Waiting

8b91b88d5577: Waiting

27e93becc0ce: Waiting

2e1c130fa3ec: Waiting

648e56f234c2: Waiting

93c337da394f: Waiting

3d6c2229f924: Waiting

8d53da260408: Waiting

824416e23423: Waiting

84c8c79126f6: Waiting

2e1c130fa3ec: Waiting

648e56f234c2: Waiting

93c337da394f: Waiting

3d6c2229f924: Waiting

8d53da260408: Layer already exists

93c337da394f: Waiting

93c337da394f: Waiting

93c337da394f: Waiting

93c337da394f: Already exists

27e93becc0ce: Pushed

3d6c2229f924: Pushed

kuberimg1: digest: sha256:5937823b9e70734c266729533d16d94c9119c948da740fe552c0ebadc5e782cb size: 856

[Pipeline] }

[Pipeline] // stage

[Pipeline] stage

[Pipeline] { (Deploy to Kubernetes)

[Pipeline] bat

C:\ProgramData\Jenkins\.jenkins\workspace\deploy>kubectl apply -f deployment.yaml --validate=false

deployment.apps/kuberdemoapp-deployment unchanged

[Pipeline] bat

C:\ProgramData\Jenkins\.jenkins\workspace\deploy>kubectl apply -f service.yaml

service/kubedemoapp-service unchanged

[Pipeline] }

[Pipeline] // stage

[Pipeline] }

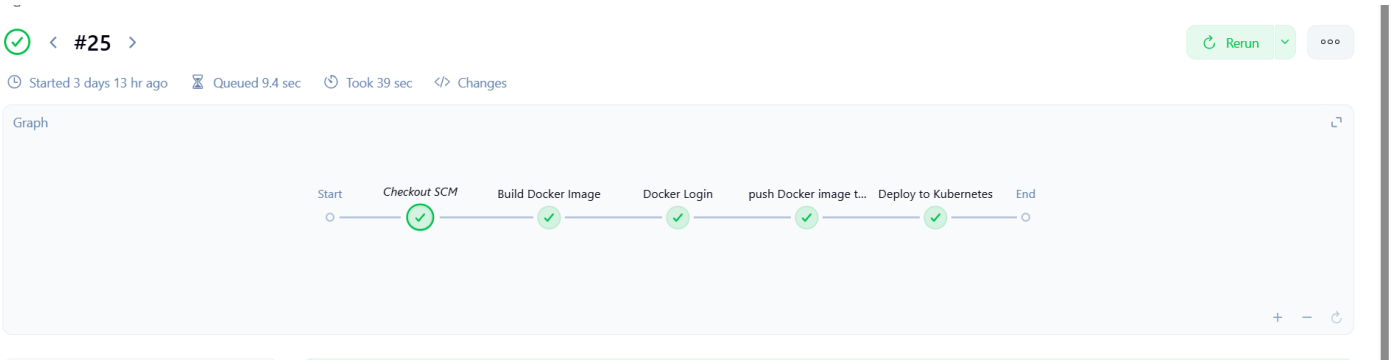
[Pipeline] // withEnv

[Pipeline] }

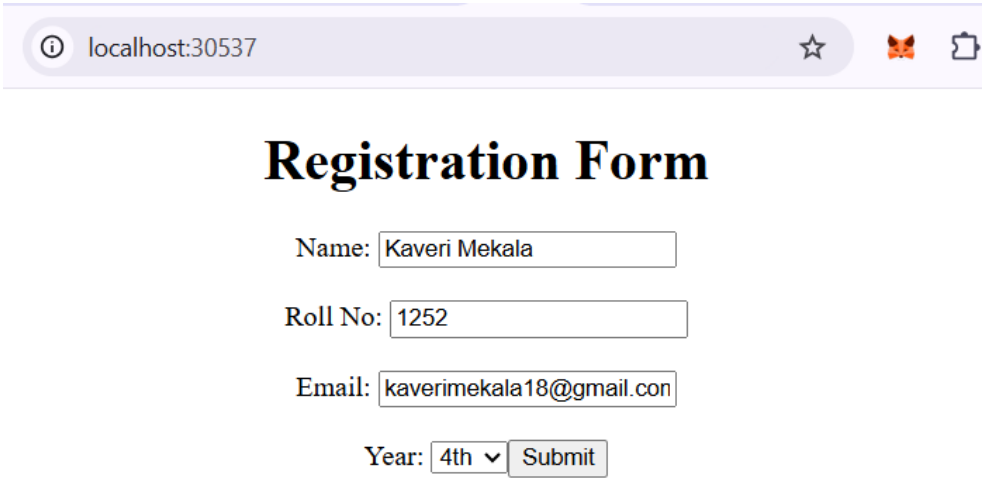
[Pipeline] // node

[Pipeline] End of Pipeline

Finished: SUCCESS



Output:



localhost:30537/submit

Welcome to the event.....

Your Details

Name

Kaveri Mekala

roll no

1252

email

kaverimekala18@gmail.com

year

4th