TASK 2

Step 1: Open Jenkins and click **"New Item."**

Step 2: Enter a project name, select **"Pipeline,"** then click **OK.**

Step 3: In the **General** tab, scroll down to the **"Pipeline"** section.

Step 4: Under **"Definition,"** select **"Pipeline script from SCM."**

Step 5: In the **SCM** field, choose **"Git,"** then enter your GitHub repository URL.

Step 6: Ensure your GitHub repository contains the following files:

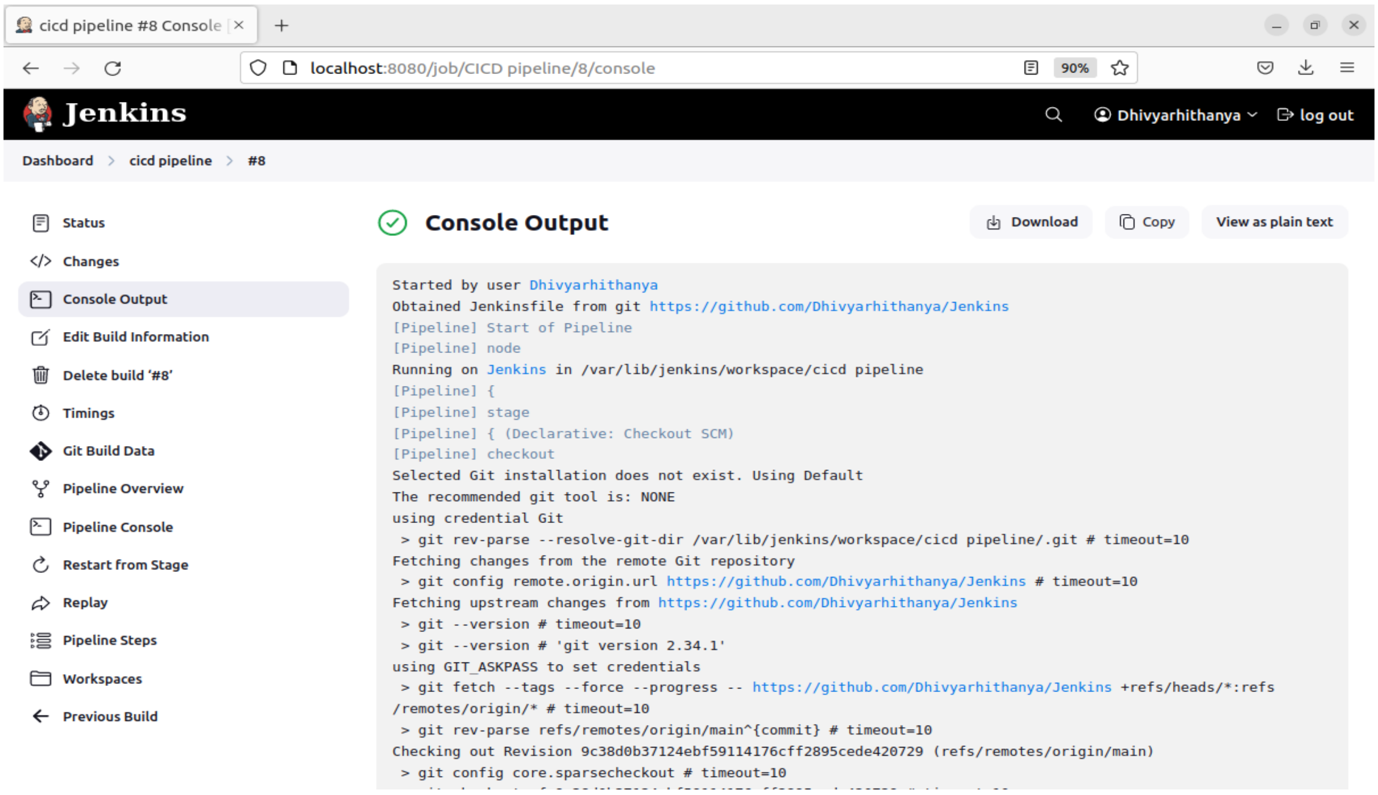
- app.py (Application code)
- Dockerfile (Instructions to build the Docker image)
- docker-compose.yml (For container orchestration)
- Jenkinsfile (Defines the pipeline stages)
- requirements.txt (Dependencies for the application)

Step 7: Click **"Save"** to apply the configuration.

Step 8: On the project page, click **"Build Now"** to start the job.

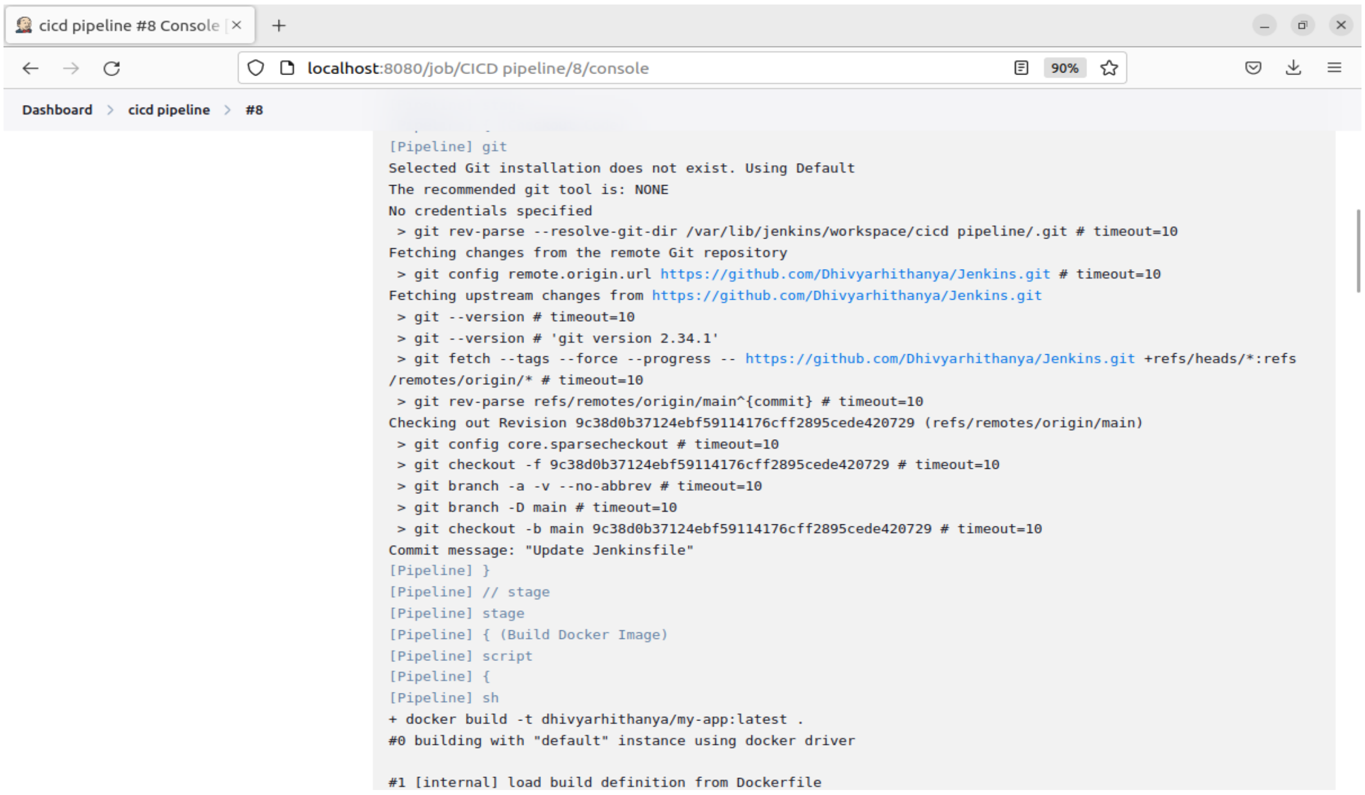
Step 9: The build status will appear in the **Build History** panel (left-side panel).

Step 10: Click on the latest build (#1, #2, etc.) and select **"Console Output"** to view logs and results.

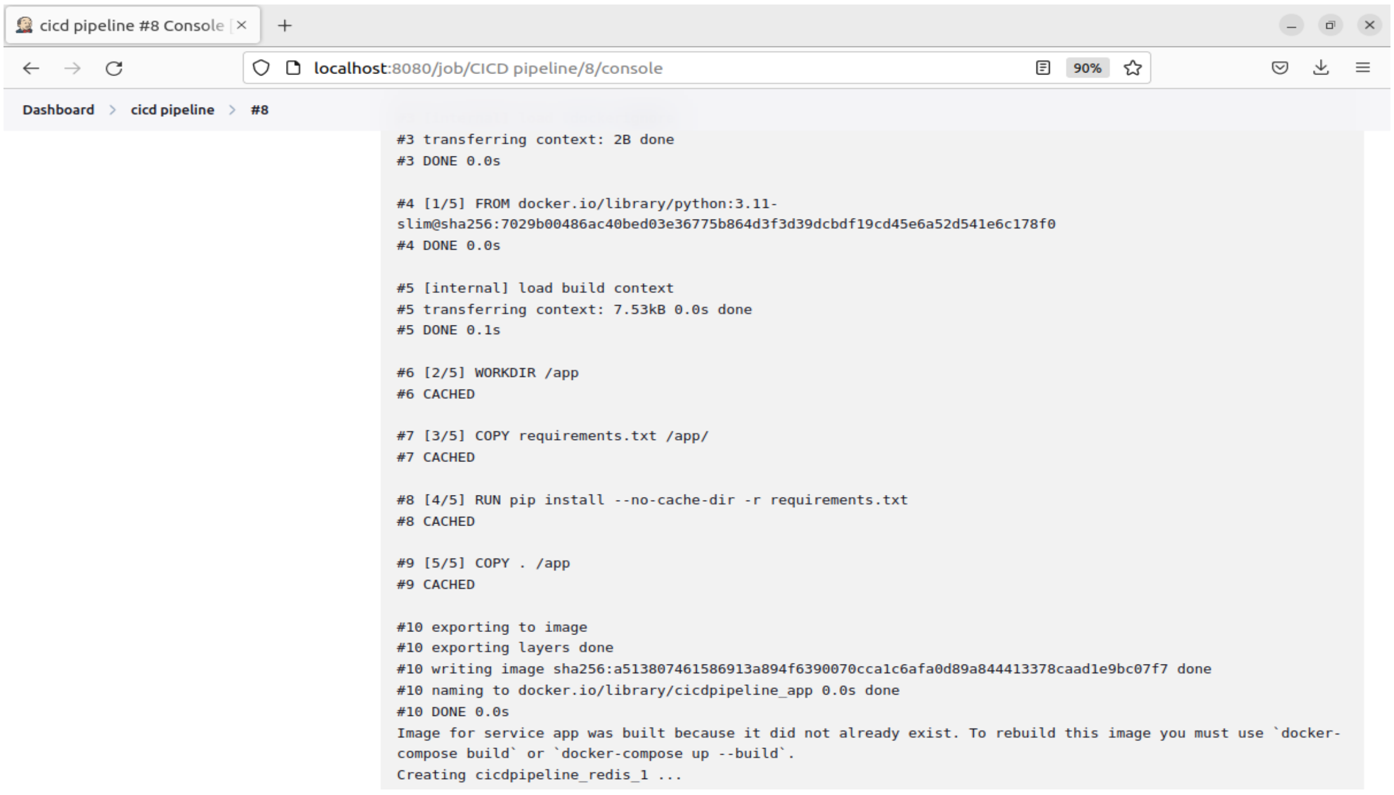


Console Output

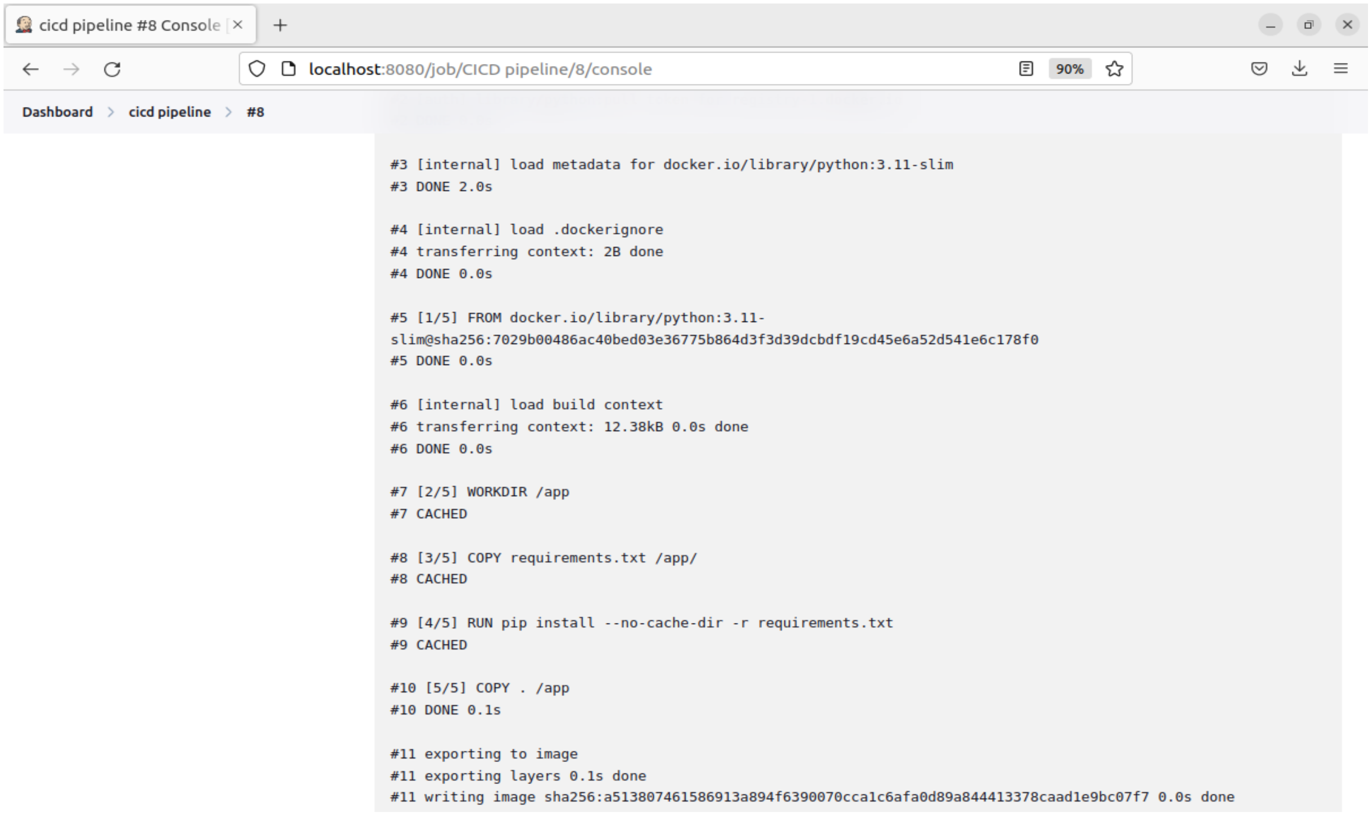
```
Started by user Dhivyarhithanya
Obtained Jenkinsfile from git https://github.com/Dhivyarhithanya/Jenkins
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in /var/lib/jenkins/workspace/cicd pipeline
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Declarative: Checkout SCM)
[Pipeline] checkout
Selected Git installation does not exist. Using Default
The recommended git tool is: NONE
using credential Git
 > git rev-parse --resolve-git-dir /var/lib/jenkins/workspace/cicd pipeline/.git # timeout=10
Fetching changes from the remote Git repository
 > git config remote.origin.url https://github.com/Dhivyarhithanya/Jenkins # timeout=10
Fetching upstream changes from https://github.com/Dhivyarhithanya/Jenkins
 > git --version # timeout=10
 > git --version # 'git version 2.34.1'
using GIT_ASKPASS to set credentials
 > git fetch --tags --force --progress -- https://github.com/Dhivyarhithanya/Jenkins +refs/heads/*:refs
/remotes/origin/* # timeout=10
 > git rev-parse refs/remotes/origin/main^{commit} # timeout=10
Checking out Revision 9c38d0b37124ebf59114176cff2895cede420729 (refs/remotes/origin/main)
 > git config core.sparsecheckout # timeout=10
```


cicd pipeline #8 Console
localhost:8080/job/CICD pipeline/8/console
90%
Dashboard > cicd pipeline > #8

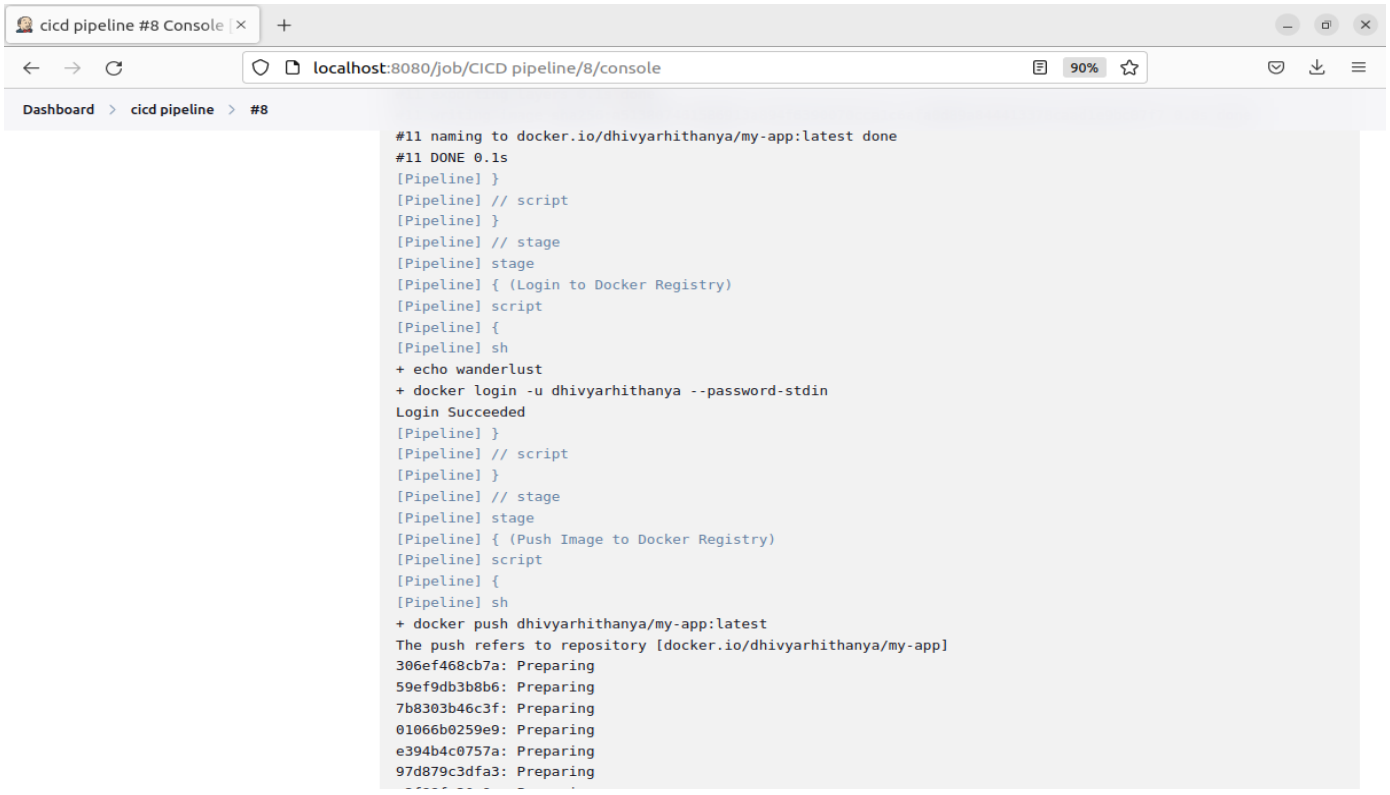
```
[Pipeline] git
Selected Git installation does not exist. Using Default
The recommended git tool is: NONE
No credentials specified
 > git rev-parse --resolve-git-dir /var/lib/jenkins/workspace/cicd pipeline/.git # timeout=10
Fetching changes from the remote Git repository
 > git config remote.origin.url https://github.com/Dhivyarhithanya/Jenkins.git # timeout=10
Fetching upstream changes from https://github.com/Dhivyarhithanya/Jenkins.git
 > git --version # timeout=10
 > git --version # 'git version 2.34.1'
 > git fetch --tags --force --progress -- https://github.com/Dhivyarhithanya/Jenkins.git +refs/heads/*:refs
/remotes/origin/* # timeout=10
 > git rev-parse refs/remotes/origin/main^{commit} # timeout=10
Checking out Revision 9c38d0b37124ebf59114176cff2895cede420729 (refs/remotes/origin/main)
 > git config core.sparsecheckout # timeout=10
 > git checkout -f 9c38d0b37124ebf59114176cff2895cede420729 # timeout=10
 > git branch -a -v --no-abbrev # timeout=10
 > git branch -D main # timeout=10
 > git checkout -b main 9c38d0b37124ebf59114176cff2895cede420729 # timeout=10
Commit message: "Update Jenkinsfile"
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Build Docker Image)
[Pipeline] script
[Pipeline] {
[Pipeline] sh
+ docker build -t dhivyarhithanya/my-app:latest .
#0 building with "default" instance using docker driver

#1 [internal] load build definition from Dockerfile
```


cicd pipeline #8 Console
localhost:8080/job/CICD pipeline/8/console
90%
Dashboard > cicd pipeline > #8

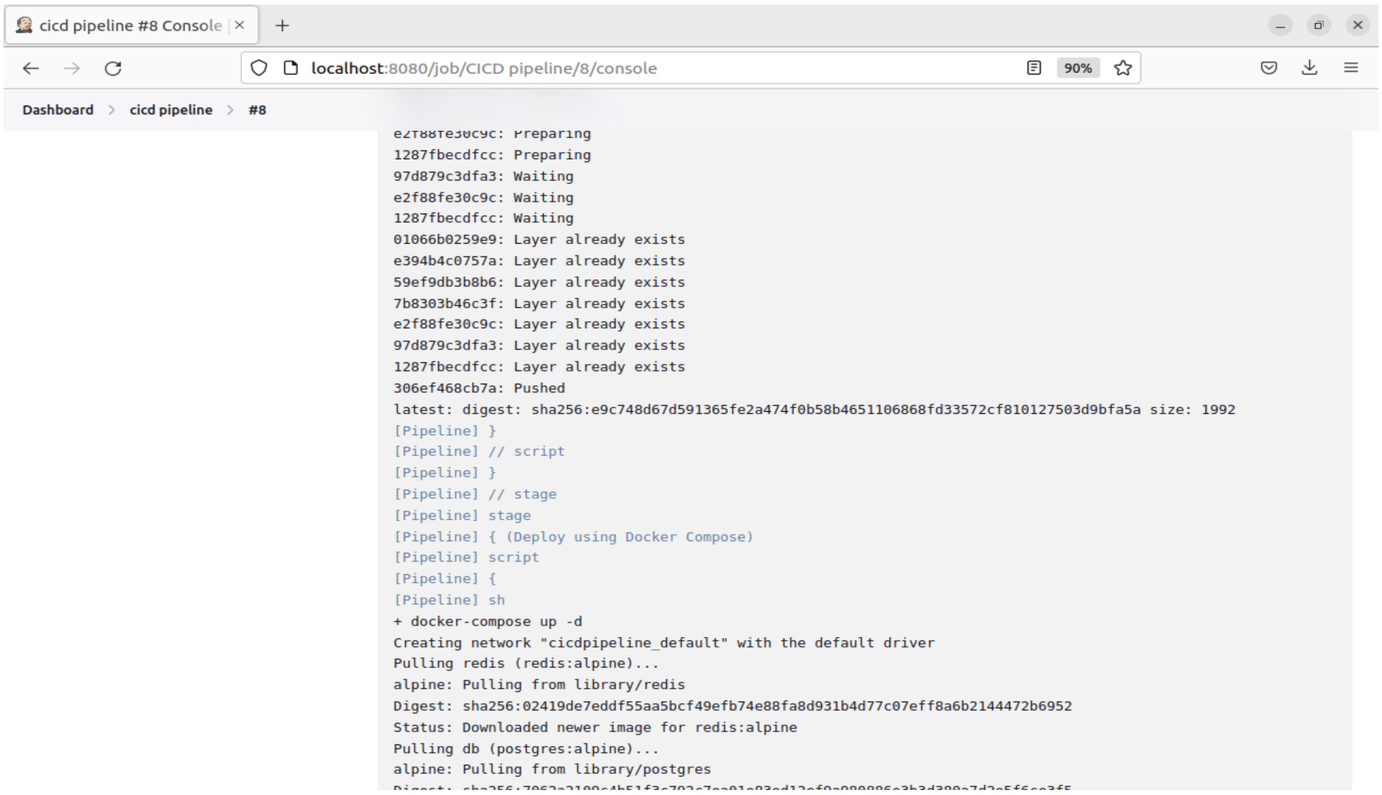
```
#3 transferring context: 2B done
#3 DONE 0.0s

#4 [1/5] FROM docker.io/library/python:3.11-
slim@sha256:7029b00486ac40bed03e36775b864d3f3d39dcbdf19cd45e6a52d541e6c178f0
#4 DONE 0.0s

#5 [internal] load build context
#5 transferring context: 7.53kB 0.0s done
#5 DONE 0.1s

#6 [2/5] WORKDIR /app
#6 CACHED

#7 [3/5] COPY requirements.txt /app/
#7 CACHED

#8 [4/5] RUN pip install --no-cache-dir -r requirements.txt
#8 CACHED

#9 [5/5] COPY . /app
#9 CACHED

#10 exporting to image
#10 exporting layers done
#10 writing image sha256:a513807461586913a894f6390070cca1c6afa0d89a844413378caad1e9bc07f7 done
#10 naming to docker.io/library/cicdpipeline_app 0.0s done
#10 DONE 0.0s
Image for service app was built because it did not already exist. To rebuild this image you must use `docker-
compose build` or `docker-compose up --build`.
Creating cicdpipeline_redis_1 ...
```


cicd pipeline #8 Console
localhost:8080/job/CICD pipeline/8/console
90%
Dashboard > cicd pipeline > #8

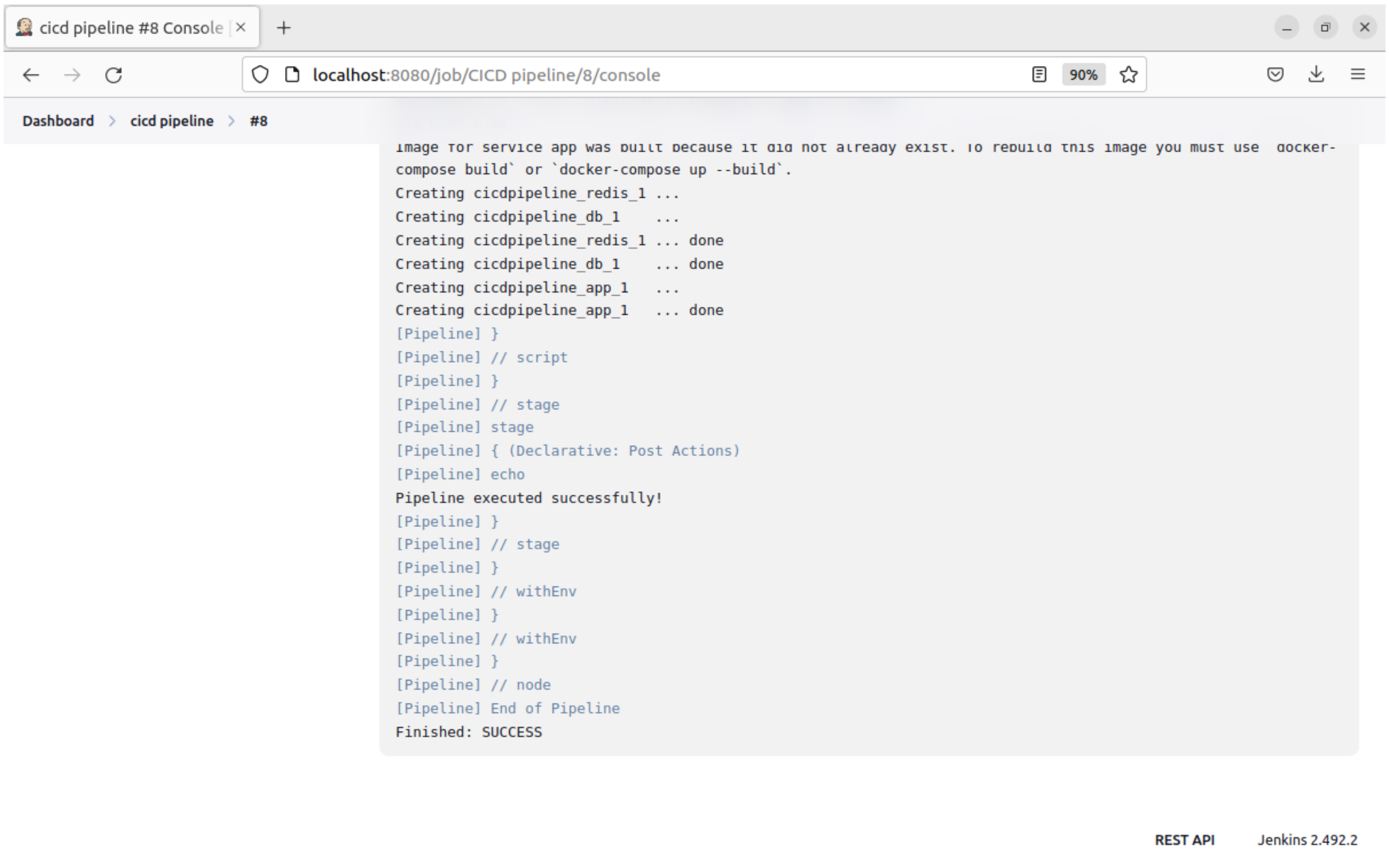
```
#3 [internal] load metadata for docker.io/library/python:3.11-slim
#3 DONE 2.0s

#4 [internal] load .dockerignore
#4 transferring context: 2B done
#4 DONE 0.0s

#5 [1/5] FROM docker.io/library/python:3.11-
slim@sha256:7029b00486ac40bed03e36775b864d3f3d39dcbdf19cd45e6a52d541e6c178f0
#5 DONE 0.0s

#6 [internal] load build context
#6 transferring context: 12.38kB 0.0s done
#6 DONE 0.0s

#7 [2/5] WORKDIR /app
#7 CACHED

#8 [3/5] COPY requirements.txt /app/
#8 CACHED

#9 [4/5] RUN pip install --no-cache-dir -r requirements.txt
#9 CACHED

#10 [5/5] COPY . /app
#10 DONE 0.1s

#11 exporting to image
#11 exporting layers 0.1s done
#11 writing image sha256:a513807461586913a894f6390070cca1c6afa0d89a844413378caad1e9bc07f7 0.0s done
```


cicd pipeline #8 Console
localhost:8080/job/CICD pipeline/8/console
90%
Dashboard > cicd pipeline > #8

```
#11 naming to docker.io/dhivyarhithanya/my-app:latest done
#11 DONE 0.1s
[Pipeline] }
[Pipeline] // script
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Login to Docker Registry)
[Pipeline] script
[Pipeline] {
[Pipeline] sh
+ echo wanderlust
+ docker login -u dhivyarhithanya --password-stdin
Login Succeeded
[Pipeline] }
[Pipeline] // script
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Push Image to Docker Registry)
[Pipeline] script
[Pipeline] {
[Pipeline] sh
+ docker push dhivyarhithanya/my-app:latest
The push refers to repository [docker.io/dhivyarhithanya/my-app]
306ef468cb7a: Preparing
59ef9db3b8b6: Preparing
7b8303b46c3f: Preparing
01066b0259e9: Preparing
e394b4c0757a: Preparing
97d879c3dfa3: Preparing
e2f88fe30c9c: Preparing
```


cicd pipeline #8 Console
localhost:8080/job/CICD pipeline/8/console
90%
Dashboard > cicd pipeline > #8

```
e2f88fe30c9c: Preparing
1287fbecdfcc: Preparing
97d879c3dfa3: Waiting
e2f88fe30c9c: Waiting
1287fbecdfcc: Waiting
01066b0259e9: Layer already exists
e394b4c0757a: Layer already exists
59ef9db3b8b6: Layer already exists
7b8303b46c3f: Layer already exists
e2f88fe30c9c: Layer already exists
97d879c3dfa3: Layer already exists
1287fbecdfcc: Layer already exists
306ef468cb7a: Pushed
latest: digest: sha256:e9c748d67d591365fe2a474f0b58b4651106868fd33572cf810127503d9bfa5a size: 1992
[Pipeline] }
[Pipeline] // script
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Deploy using Docker Compose)
[Pipeline] script
[Pipeline] {
[Pipeline] sh
+ docker-compose up -d
Creating network "cicdpipeline_default" with the default driver
Pulling redis (redis:alpine)...
alpine: Pulling from library/redis
Digest: sha256:02419de7eddf55aa5bcf49efb74e88fa8d931b4d77c07eff8a6b2144472b6952
Status: Downloaded newer image for redis:alpine
Pulling db (postgres:alpine)...
alpine: Pulling from library/postgres
Digest: sha256:7062a2100c4b51f3c792c7ea01e83ed12af0a080886a3b3d380a7d2e5f6ce3f5
```



```
Image for service app was built because it did not already exist. To rebuild this image you must use `docker-
compose build` or `docker-compose up --build`.
Creating cicdpipeline_redis_1 ...
Creating cicdpipeline_db_1    ...
Creating cicdpipeline_redis_1 ... done
Creating cicdpipeline_db_1    ... done
Creating cicdpipeline_app_1   ...
Creating cicdpipeline_app_1   ... done
[Pipeline] }
[Pipeline] // script
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Declarative: Post Actions)
[Pipeline] echo
Pipeline executed successfully!
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```