

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:

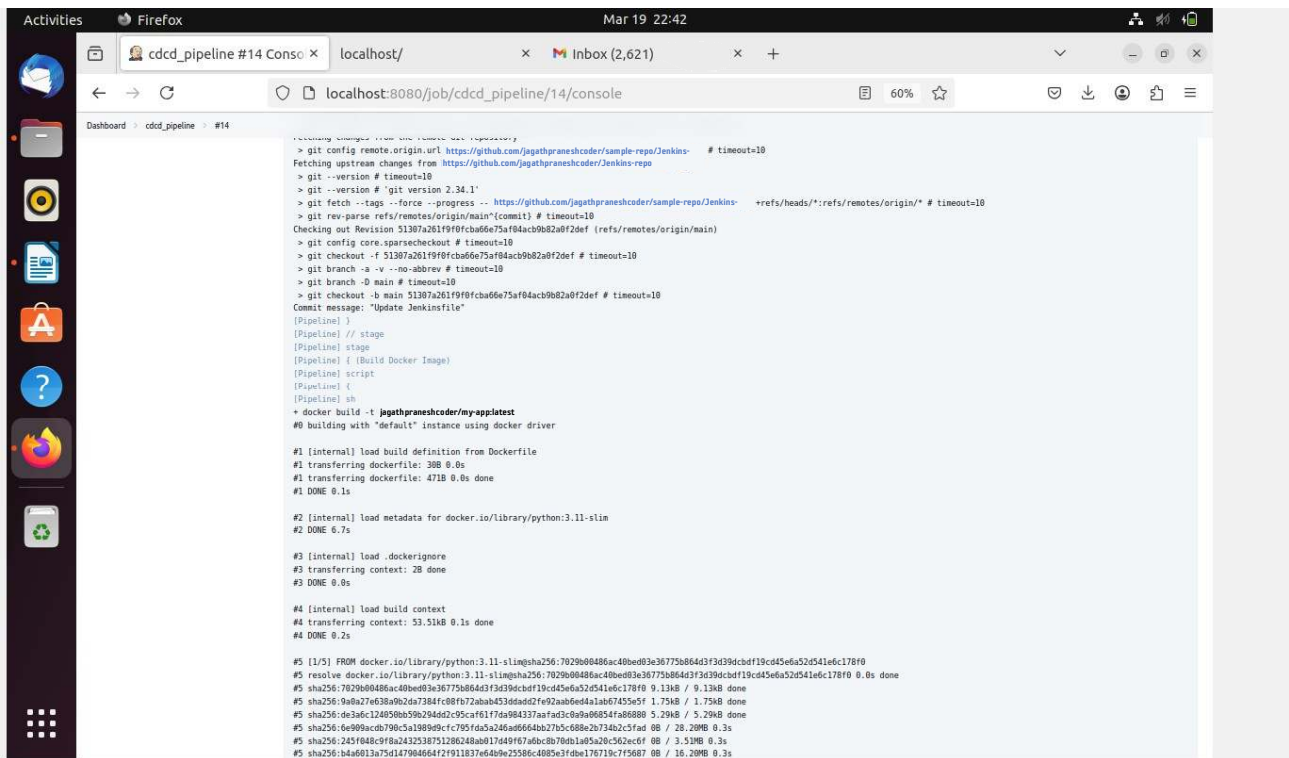
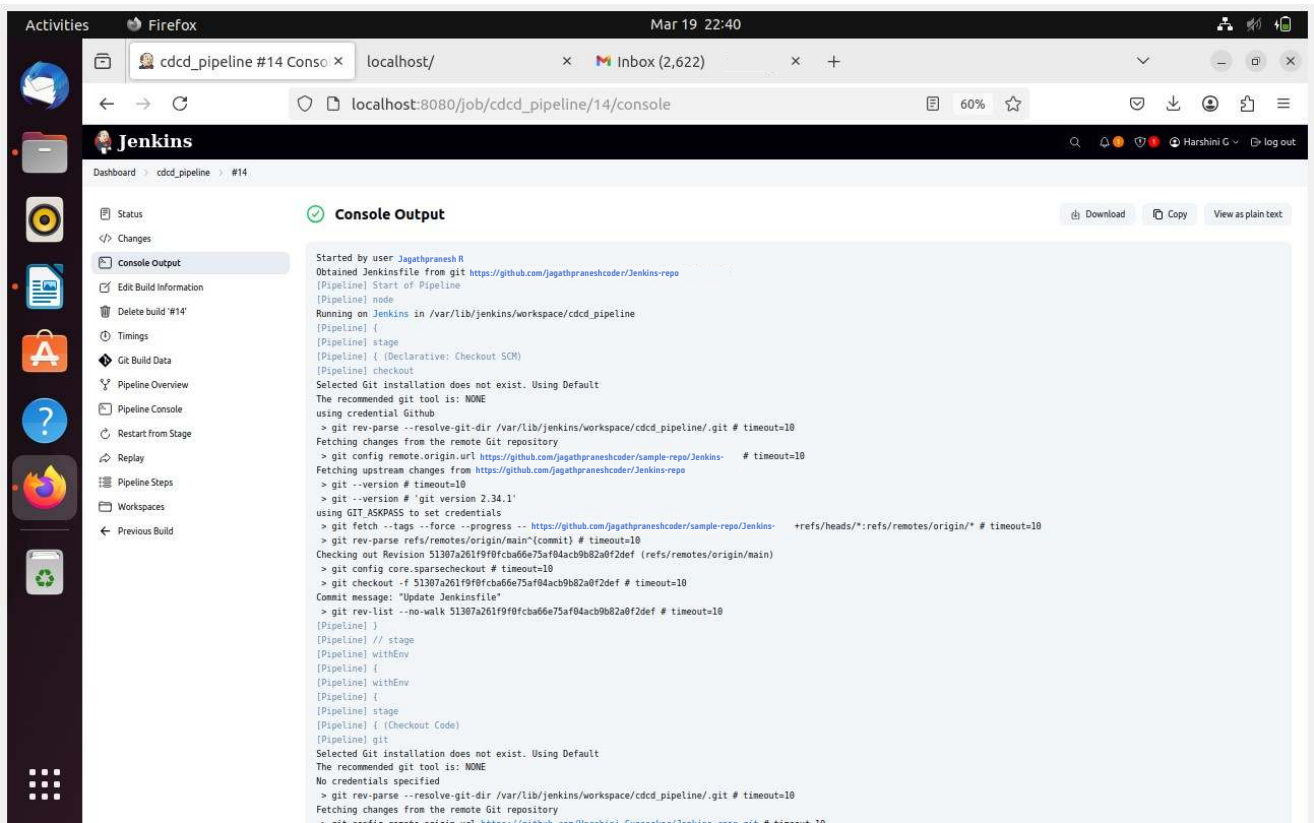
- app.py (Application code)
- Dockerfile (Instructions to build the Docker image)
- docker-compose.yml (For container orchestration)
- Jenkinsfile (Defines the pipeline stages)
- requirements.txt (Dependencies for the application)

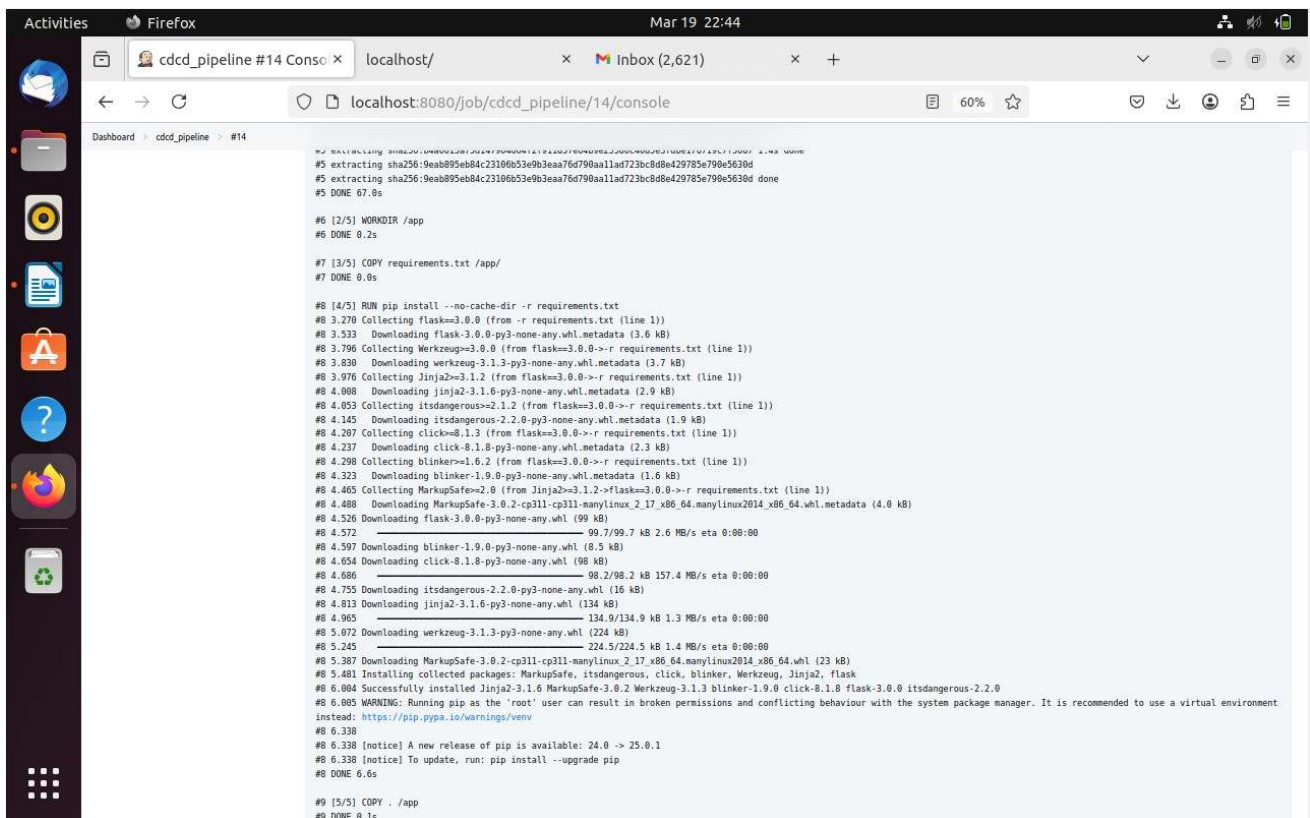
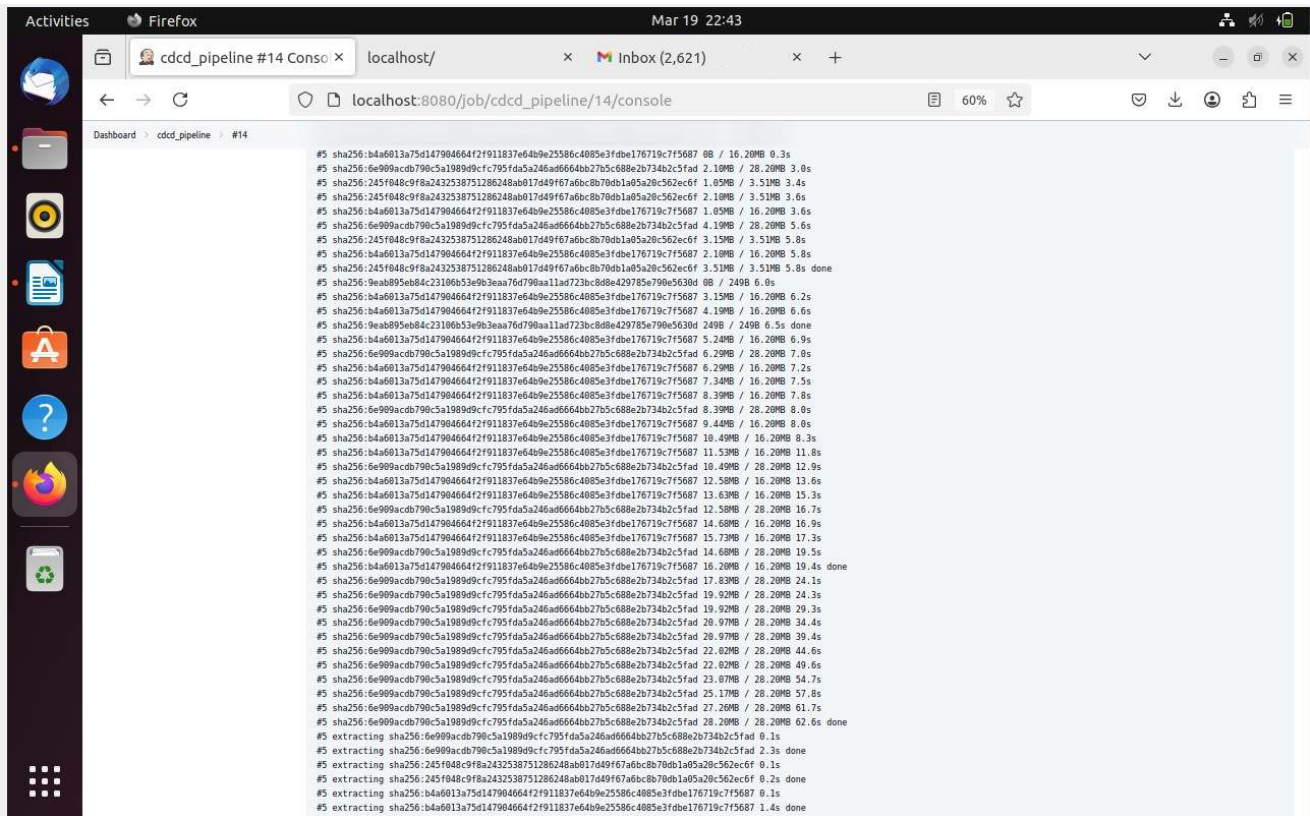
Step 7: Click "**Save**" to apply the configuration.

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.





Activities Firefox Mar 19 22:45

cdcd_pipeline #14 Conso x localhost/ x Inbox (2,621) x +

localhost:8080/job/cdcd_pipeline/14/console 60%

Dashboard > cdcd_pipeline > #14

```
#9 DONE 0.1s

#10 exporting to image
#10 exporting layers
#10 exporting layers 0.3s done
#10 writing image sha256:4379f1ed67e821f45a77c1f99f56e6ed6c748d9ab229db5b5e540ff67443944a
#10 writing image sha256:4379f1ed67e821f45a77c1f99f56e6ed6c748d9ab229db5b5e540ff67443944a done
#10 naming to docker.io/jagathraneshcoder/my-apptest
#10 DONE 0.3s

[Pipeline] }
[Pipeline] // script
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Login to Docker Registry)
[Pipeline] script
[Pipeline] {
[Pipeline] sh
+ docker login -u jagathraneshc --password-stdin
+ echo jagathraneshcoder

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

Login Succeeded
[Pipeline] }
[Pipeline] // script
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Push Image to Docker Registry)
[Pipeline] script
[Pipeline] {
[Pipeline] sh
+ docker push jagathraneshcoder/my-apptest
The push refers to repository [docker.io/jagathraneshcoder/my-apptest/]
7ba075448493: Preparing
c514f6da6578: Preparing
eedf7ad881ca: Preparing
44bd3b346f68: Preparing
e394b4c0757a: Preparing
97d879c3d3fa3: Preparing
e2f88fe38c9c: Preparing
1287fbecdfcc: Preparing
97d879c3d3fa2: Waiting
e2f88fe38c9c: Waiting
1287fbecdfcc: Waiting
```

Activities Firefox Mar 19 22:46

cdcd_pipeline #14 Conso x localhost/ x Inbox (2,621) x +

localhost:8080/job/cdcd_pipeline/14/console 60%

Dashboard > cdcd_pipeline > #14

```
Waiting
1287fbecdfcc: Waiting
e394b4c0757a: Mounted from library/python
eedf7ad881ca: Pushed
44bd3b346f68: Pushed
7ba075448493: Pushed
97d879c3d3fa3: Mounted from library/python
1287fbecdfcc: Mounted from library/python
e2f88fe38c9c: Mounted from library/python
c514f6da6578: Pushed
latest: digest: sha256:de7ed47642a853f103e4f9f0103bc6c5f18a04ff8bd3ef49f725c92980bc949b7 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 "cdcd_pipeline_default" with the default driver
Pulling redis (redis:alpine)...
alpine: Pulling from library/redis
Digest: sha256:82419de7edd55a5b6cf49efb74e80fab06931b4d77c07eff8a6b214447266952
Status: Downloaded newer image for redis:alpine
Pulling db (postgres:alpine)...
alpine: Pulling from library/postgres
Digest: sha256:7062a2109c4b51f3c792c7ea01e83ed12ef9a980886e3b3d380a7d2e5f6ce3f5
Status: Downloaded newer image for postgres:alpine
Building app
#0 building with "default" instance using docker driver

#1 [internal] load build definition from Dockerfile
#1 transferring dockerfile: 471B done
#1 DONE 0.0s

#2 [internal] load metadata for docker.io/library/python:3.11-slim
#2 ...

#3 [auth] library/python:pull token for registry-1.docker.io
#3 DONE 0.0s

#2 [internal] load metadata for docker.io/library/python:3.11-slim
#2 DONE 3.4s

#4 [internal] load .dockerignore
#4 transferring context: 2B done
```


Activities Firefox Mar 19 22:46

cdcd_pipeline #14 Conso x localhost/ x Inbox (2,621) x +

localhost:8080/job/cdcd_pipeline/14/console 60%

Dashboard > cdcd_pipeline > #14

```
#4 transferring context: 28 done
#4 DONE 0.0s

#5 [1/5] FROM docker.io/library/python:3.11-slimsha256:7029b60486ac40bed03e36775b864d3f3d39dcdf19cd45e6a52d541ef6c178f0
#5 DONE 0.0s

#6 [internal] load build context
#6 transferring context: 4.81kB 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 CACHED

#11 exporting to image
#11 exporting layers done
#11 writing image sha256:4379f1ed67e821f45a77c1f99f56e6ed6c748dbab229db5b5e540ff67443944a done
#11 naming to docker.io/library/cdcd_pipeline_app done
#11 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 cdcd_pipeline_db_1 ...
Creating cdcd_pipeline_redis_1 ...
Creating cdcd_pipeline_db_1 ... done
Creating cdcd_pipeline_app_1 ...
Creating cdcd_pipeline_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] }
```

Activities Firefox Mar 19 22:47

cdcd_pipeline #14 Conso x localhost/ x Inbox (2,621) x +

localhost:8080/job/cdcd_pipeline/14/console 60%

Dashboard > cdcd_pipeline > #14

```
#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 CACHED

#11 exporting to image
#11 exporting layers done
#11 writing image sha256:4379f1ed67e821f45a77c1f99f56e6ed6c748dbab229db5b5e540ff67443944a done
#11 naming to docker.io/library/cdcd_pipeline_app done
#11 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 cdcd_pipeline_db_1 ...
Creating cdcd_pipeline_redis_1 ...
Creating cdcd_pipeline_db_1 ... done
Creating cdcd_pipeline_app_1 ...
Creating cdcd_pipeline_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
```

REST API Jenkins 2.501