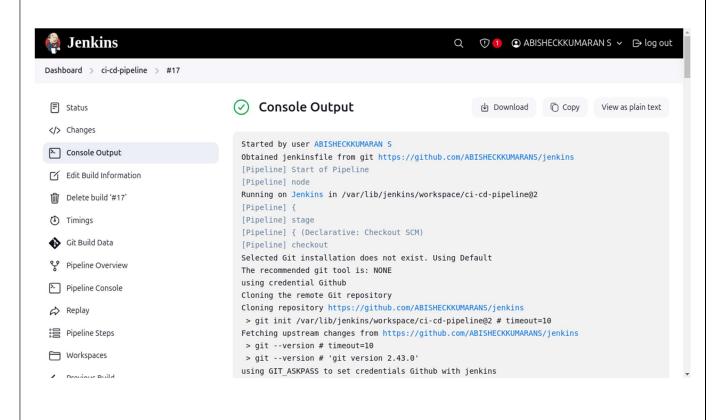
## TASK 2

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.



```
Dashboard > ci-cd-pipeline > #17
                                > git config core.sparsecheckout # timeout=10
                                > git checkout -f 2954a01d8c2db5cd5abfee072120bb6532cb5a2b # timeout=10
                                > git branch -a -v --no-abbrev # timeout=10
                                > git checkout -b main 2954a01d8c2db5cd5abfee072120bb6532cb5a2b # timeout=10
                               Commit message: "Update jenkinsfile"
                               [Pipeline] }
                               [Pipeline] // stage
                               [Pipeline] stage
                               [Pipeline] { (Build Docker Image)
                               [Pipeline] script
                               [Pipeline] {
                               [Pipeline] sh
                               + docker build -t abisheck2288/my-app:latest .
                               #0 building with "default" instance using docker driver
                               #1 [internal] load build definition from Dockerfile
                               #1 transferring dockerfile:
                               #1 transferring dockerfile: 471B 0.0s done
                               #1 DONE 0.3s
                               #2 [internal] load metadata for docker.io/library/python:3.11-slim
                               #2 DONE 1.0s
                               #3 [internal] load .dockerignore
                               #3 transferring context:
                               #3 transferring context: 2B 0.0s done
```

```
Dashboard > ci-cd-pipeline > #17
                               [Pipeline] script
                               [Pipeline] {
                               [Pipeline] sh
                               + docker push abisheck2288/my-app:latest
                               The push refers to repository [docker.io/abisheck2288/my-app]
                               b96e4770dd5b: Preparing
                               c690c6b3e3e3: Preparing
                               f08c3ca03772: Preparing
                              6f6a78890d2b: Preparing
                               e394b4c0757a: Preparing
                              97d879c3dfa3: Preparing
                               e2f88fe30c9c: Preparing
                              1287fbecdfcc: Preparing
                              97d879c3dfa3: Waiting
                              e2f88fe30c9c: Waiting
                              1287fbecdfcc: Waiting
                               6f6a78890d2b: Layer already exists
                              e394b4c0757a: Layer already exists
                               c690c6b3e3e3: Layer already exists
                               f08c3ca03772: Layer already exists
                              97d879c3dfa3: Layer already exists
                               1287fbecdfcc: Layer already exists
                              e2f88fe30c9c: Layer already exists
                               b96e4770dd5b: Pushed
                              latest: digest: sha256:fed3c30abe5e845d41a5ba446da82d439b4cf3c3f807bf4eb4c8b122b07203b3
```

```
Dashboard > ci-cd-pipeline > #17
                                                                                                                                           CONTRACTOR
                                                                                                                                           \label{time=2025-03-19T12:08:55+05:30"} time="2025-03-19T12:08:55+05:30" level=warning msg="Found orphan containers" ([jenkins-app-1]) and the second orphan containers of the second orphan
                                                                                                                                           jenkins-redis-1 jenkins-db-1]) for this project. If you removed or renamed this service in
                                                                                                                                           your compose file, you can run this command with the --remove-orphans flag to clean it up."
                                                                                                                                            Container my-app Creating
                                                                                                                                              Container my-app Created
                                                                                                                                            Container my-app Starting
                                                                                                                                             Container my-app Started
                                                                                                                                           [Pipeline] }
                                                                                                                                           [Pipeline] // script
                                                                                                                                           [Pipeline] }
                                                                                                                                           [Pipeline] // stage
                                                                                                                                           [Pipeline] stage
                                                                                                                                           [Pipeline] { (Declarative: Post Actions)
                                                                                                                                           [Pipeline] echo
                                                                                                                                           Pipeline executed successfully!
                                                                                                                                           [Pipeline] }
                                                                                                                                           [Pipeline] // stage
                                                                                                                                           [Pipelinel }
                                                                                                                                           [Pipeline] // withEnv
                                                                                                                                           [Pipeline] }
                                                                                                                                           [Pipeline] // withEnv
                                                                                                                                           [Pipeline] }
                                                                                                                                           [Pipeline] // node
                                                                                                                                           [Pipeline] End of Pipeline
                                                                                                                                           Finished: SUCCESS
```