

Write down the steps and the commands to upload the project files to a GitHub repository using Git bash.

Solution:

Step 1: Initialise the git repository

- `git init`

Step 2: Add the created files.

- `git add .`

Step 3: Check the status

- `git status`

Step 4: Commit the staged changes

- `git commit -m "commit message"`

Step 5: Add a remote repository <https://github.com/abc/myrepo.git>

- `git remote add origin master https://github.com/abc/myrepo.git`

Step 6: Push the changes to GitHub.

- `git push origin master`

---

Explain the steps involved in defining a Delivery Pipeline using Jenkins.

Solution:

To create a basic Pipeline through the Jenkins classic UI:

Step 1: If required, ensure you are logged in to Jenkins and install pipeline plugin.

Step 2: From the Jenkins home page (i.e. the Dashboard of the Jenkins classic UI), click New Item at the top left.

Step 3: In the Enter an item name field, specify the name for your new Pipeline project.

Step 4: Scroll down and click Pipeline, then click OK at the end of the page to open the Pipeline configuration page.

Step 5: Click the Pipeline tab at the top of the page to scroll down to the Pipeline section.

Step 6: In the Pipeline section, ensure that the Definition field indicates the Pipeline script option. Enter your Pipeline code into the Script text area.

Step 7: Click Save to open the Pipeline project/item view page.

Step 8: On this page, click Build Now on the left to run the Pipeline.

Step 9: Under Build History on the left, click #1 to access the details for this particular Pipeline run.

Step 10: Click Console Output to see the full output from the Pipeline run.

---

Illustrate the process of creating a Jenkins job <project\_name>\_source to poll the SCM for every 15 minutes and compile the project and archive the artefacts.

Solution:

Step 1: Create a new Jenkins job named 'payslip\_source.'

Step 2: Under general section, select 'Discard old builds' option and select strategy as Log rotation and days to keep build as 2 and maximum builds to keep as 5.

Step 3: Select Git, as source code management.

a. Under repositories give the repository url as 'https://github.com/devopskce/Payslip-app-tests.git'

b. Branches to build as \*/master

Step 4: Under build triggers section, select Poll SCM option and give the value as

H/15 \* \* \* \* so that the build is triggered every 15 minutes.

Step 5: Under Build Section, select Invoke top level maven targets,

- a. Select the maven version
- b. Set the goals as 'compile'

Step 6: Under post-build actions, click on Archive the artifacts and give the files to archive as '\*\*'

Step 7: Apply and save the created job.

---