# In the jenkins server, we need to install the list of plugins to be installed and configured before we create the pipelines as per the requirement.

•   Git

•   Bitbucket

•   Sonarqube and sonar-scannner(code quality)

•   Artifactory(**for** storing the artifacts)

•   Credentials (to store the credentials **for** target servers)

•   Test evnironment related plugins(UAT, SMOKE and Release)

•   SSH Steps

•   AWS Plugin etc.,

# Configure the credentials **in** the credentials manager to access the target servers.

# Create a multibranch pipeline and provide the source code repository details **in** the job.

1. Create a Jenkinsfile(decalrative) **in** the source code repository which has the steps/stages configured to create the pipeline **in** Jenkins.
2. Jenkinsfile has various **default** stages like checkout, Build, Unittest, Codequality.
3. Assuming a python project, created a sample python file **in** the python folder **of** repository.
4. Now creating the zip file **of** the python folder **in** the **package** stage and storing **this** **package** **in** JFrog artifactory.
5. If the artifact **is** created **from** feature/hotfix branch then storing them **in** snapshot version **of** artifact repository.
6. If the artifact created **from** release/master branch then storing them **in** release version **of** artifact repository inside a release vrsion folder.
7. Assuming the deployment **for** an Ec2/linux machine, copying th artifact to the target environment via the SSH based on the approval strategy mentioned **in** the use **case** and sending the notifications to the respective teams.
8. Similarly, Performing the post deployment test executions **in** the test stage and sending the notifications to the qa and respective committer.
9. After all the above stage sending the final build notifications via email to the commiter, DL team **with** the commit id and branchname **with** the build notifications.