1. Create a simple freestyle project that triggers a job after every 1 minute.

**Solution**

Log into your Jenkins server-

1. Go to new item, give a name to your project, select freestyle project and click OK
2. Give a description to your project(optional)
3. Under ‘Build Triggers’, select ‘build periodically’ and type \* \* \* \* \* in the shell
4. Under ‘Build Step’, select ‘Execute windows batch command’ and type the following commands-

echo “my first Jenkins project: %date% : %time%”

1. Save and apply
2. Build.
3. Disable the project to stop the build triggers.
4. Create a Jenkins freestyle project that integrates Jenkins with Git to execute a simple java program

Solution

1. Write a simple java program and save it as ‘myprog.java’(say) and save it in any directory of your choice.
2. **Write the java code in step 2.**

public class Switch {

public static void main(String args[]){

int day = 10;

switch(day){

case 1:{

System.out.println("Monday");

break;

}

case 2:{

System.out.println("Tuesday");

break;

}

case 3:{

System.out.println("Wednesday");

break;

}

case 4:{

System.out.println("Thursday");

break;

}

case 5:{

System.out.println("Friday");

break;

}

case 6:{

System.out.println("Saturday");

break;

}

case 7:{

System.out.println("Sunday");

break;

}

default:{

System.out.println("Please enter numbers in the range 1:7");

break;

}

}

}

}

1. Create a github repository for your project.
2. **Open gitbash from the local directory consisting of your ‘myprog.java’ file.** [Say the directory in your local system is ‘C:\Users\user\Desktop\Namrata\_Das\_PU\Fall\_AY\_2023-24\Subject\_Handled\DevOps\java\myprog.java’].
3. Push the project to your repository [**mention all steps executed in git bash to push the project to the github repository**]
4. In your Jenkins server-
5. Go to new item, give a name to your project, select freestyle project and click OK.
6. Give a description to your project(optional)
7. Choose the SCM of your choice, say Git.
8. Provide the repository URL , credentials(optional, if public) and Branches to build ‘\*/master’ (if git branch is master) or ‘\*/main’ (if git branch is main)
9. Under ‘Build Step’, select ‘Execute windows batch command’ and type the following commands-

javac myprog.java

java myprog

1. Save and apply
2. Build.
3. Create a simple Jenkins scripted pipeline.

**Solution**

In your Jenkins server-

1. Go to new item, give a name to your project, select pipeline and click OK
2. Give a description to your project(optional)
3. Go to pipeline, select the ‘Pipeline Script’ option and copy paste the sample pipeline in the command-line shell as given-

node{

stage('Compile'){

echo "Compiled Successfully!!";

}

stage('JUnit') {

echo "JUnit Passed Succesfully!";

}

stage('Quality-Gate') {

echo "SonarQube Quality Gate passed succesfully!!";

/\*sh exit ("1");\*/

}

stage('Deploy') {

echo "Pass!";

}

}

1. Save and apply
2. Build.
3. Create a simple Jenkins declarative pipeline.

**Solution**

In your Jenkins server-

1. Go to new item, give a name to your project, select pipeline and click OK
2. Give a description to your project(optional)
3. Go to pipeline, select the ‘Pipeline Script’ option and copy paste the sample pipeline in the command-line shell as given-

pipeline {

agent any

stages {

stage('Git-Checkout'){

steps {

echo "Checking out from Git Repo";

}

}

stage('Build') {

steps {

echo "Building the checkedout project!";

}

}

stage('Unit-Test') {

steps {

echo "Running JUnit tests";

}

}

stage('Quality-Gate') {

steps {

echo "Verifying Quality Gates";

}

}

stage('Deploy') {

steps {

echo "Deploying to stage environments for more tests";

}

}

}

post {

always {

echo 'This will always run'

}

success {

echo 'This will run only if successful'

}

failure {

echo 'This will run only if failed'

}

unstable {

echo 'This will run only if the run was marked as unstable'

}

changed {

echo 'This will run only if the state of the pipeline has changed'

echo 'For example, if the pipeline was previously failing but is now successful'

}

}

}

1. Save and apply
2. Build.