Lab 10

Laboratory Exercise

Part 1:

LAB EXERCISE

This lab will cover common Jenkins fundamentals, as well as to setup a Jenkins Pipeline (used the pipeline script directly on Jenkins)

Time to Complete
Approximately 40 Minutes

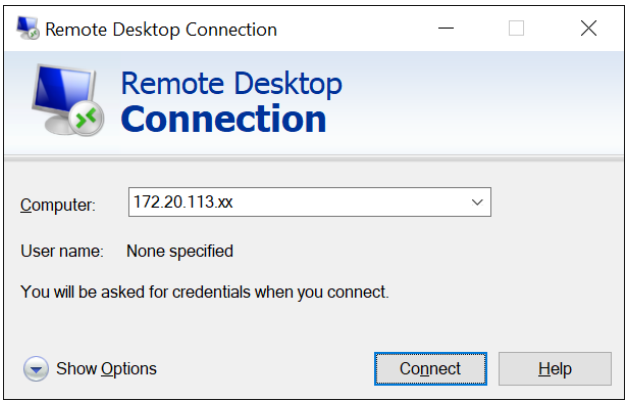
What You Need

1. Jenkins master (Check Jenkins installation steps)
2. Jenkins slave node connected to the master (Check Jenkins slave setup guide). For this module, we will work on master node (i.e. no slave node).
3. Access to Github.com from your Jenkins server.

From your machine logged-in to RP VPN, run Remote Desktop Connection to connect to the ubuntu Linux Virtual Machine (VM). Please login based on your assigned VM as shown below:

S/N	Name	VM	IP Address	User Name	Password
1	ABDUL SALIM BIN ABDUL RASHITH	LABC03 - 172.20.115.50	172.20.115.50	dockeradm	docker!2
2	CASPER LEOW YU HAN (LIAO YU HANG)	LABC03 - 172.20.115.51	172.20.115.51	dockeradm	docker!2
3	CHAN JUN ZHI, GLENN	LABC03 - 172.20.115.52	172.20.115.52	dockeradm	docker!2
4	CHIA WAI TAT	LABC03 - 172.20.115.53	172.20.115.53	dockeradm	docker!2
5	HOI WAI TECK	LABC03 - 172.20.115.54	172.20.115.54	dockeradm	docker!2
6	KOH JIN CAI DAEMIAN	LABC03 - 172.20.115.55	172.20.115.55	dockeradm	docker!2
7	KYAW KYAW OO	LABC03 - 172.20.115.56	172.20.115.56	dockeradm	docker!2
8	LUM YOKE FAI	LABC03 - 172.20.115.57	172.20.115.57	dockeradm	docker!2
9	MUHAMMAD FADHLI BIN MOHAMED NOOR	LABC03 - 172.20.115.58	172.20.115.58	dockeradm	docker!2
10	MUHAMMAD HILMEE BIN MD ALI	LABC03 - 172.20.115.59	172.20.115.59	dockeradm	docker!2

11	NG SAY WEE	LABC03 - 172.20.115.60	172.20.115.60	dockeradm	docker!2
12	NGUI WEILY	LABC03 - 172.20.115.61	172.20.115.61	dockeradm	docker!2
13	NU'MAN HARITH BIN NORRAIMI	LABC03 - 172.20.115.62	172.20.115.62	dockeradm	docker!2
14	RULY JANUAR FACHMI	LABC03 - 172.20.115.76	172.20.115.76	dockeradm	docker!2
15	SEAH SHIH WEI GEROME	LABC03 - 172.20.115.64	172.20.115.64	dockeradm	docker!2
16	SEAN CHENG ZHI WEI	LABC03 - 172.20.115.65	172.20.115.65	dockeradm	docker!2
17	SEY KOK SIONG	LABC03 - 172.20.115.66	172.20.115.66	dockeradm	docker!2
18	TAN JOON YEE DOUGLAS	LABC03 - 172.20.115.67	172.20.115.67	dockeradm	docker!2
19	WU WAI TENG VANESSA	LABC03 - 172.20.115.68	172.20.115.68	dockeradm	docker!2
20	YAP KOON SING	LABC03 - 172.20.115.69	172.20.115.69	dockeradm	docker!2
21	YE CHENG LIM	LABC03 - 172.20.115.70	172.20.115.70	dockeradm	docker!2
22	SHAIFUL BIN ABDUL KARIM	LABC03 - 172.20.115.71	172.20.115.71	dockeradm	docker!2
23	CHAI RU YI	LABC03 - 172.20.115.72	172.20.115.72	dockeradm	docker!2
24	JWAY HWEE LING JULIE	LABC03 - 172.20.115.73	172.20.115.73	dockeradm	docker!2
25	SAMANTHA TEO XING YEE	LABC03 - 172.20.115.74	172.20.115.74	dockeradm	docker!2
26	ZIL AZZA HILMIAH BINTE RADUAN	LABC03 - 172.20.115.75	172.20.115.75	dockeradm	docker!2



Replace xx with the IP address of the VM that you have been assigned.

Upgrade Jenkins

As the jenkins version is old, we need to upgrade the jenkins software by following the following steps.

a) Login to jenkins:

Access http://localhost:9090

Username: jadmin

Password: Jadmin!2

b) Click on Manage Jenkins -> Look for the following and click on download.

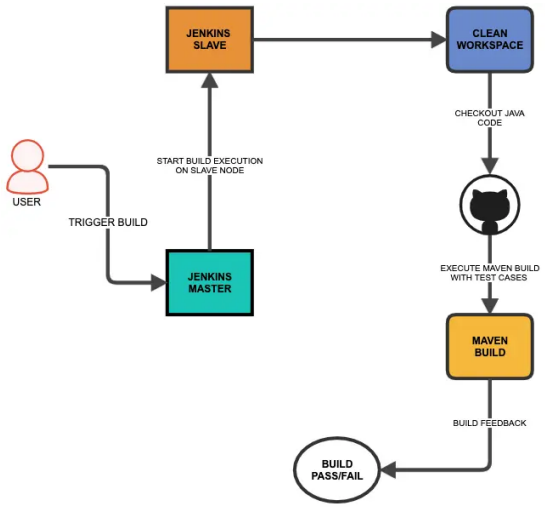
c)

```
dockeradm@sddo-vm$ /etc/init.d/jenkins  stop
dockeradm@sddo-vm$ cd /home/dockeradm/Downloads
dockeradm@sddo-vm$ mv /usr/share/jenkins/jenkins.war /opt
dockeradm@sddo-vm$ mv jenkins.war /usr/share/jenkins
dockeradm@sddo-vm$ /etc/init.d/jenkins  start
```

Upgrade is complete!!

Java Spring Boot application build using the declarative pipeline as code

1. Here is the pictorial representation of the simple build pipeline we are going to build.



2. Configure Pipeline as Code Job in Jenkins

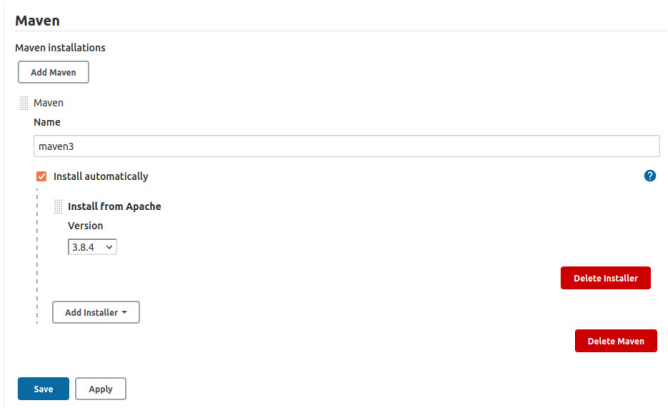
   Access http://localhost:9090

   Username: jadmin

   Password: Jadmin!2

2.1 Go to Manage Jenkins -> Global Tool Configuration -> Maven -> Maven Installation.

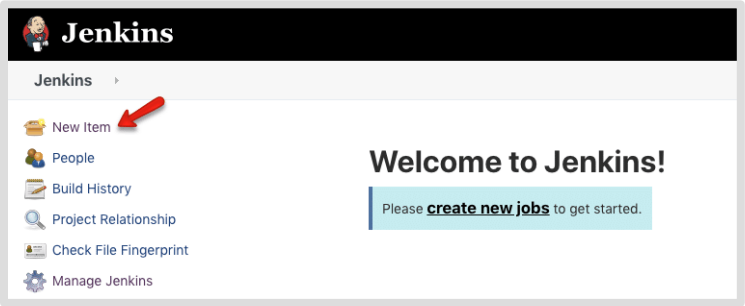
2.2 Add a maven configuration as shown below. We are using the tool named maven3 in the pipeline. Maven tool is under “Global Tool Configuration”.



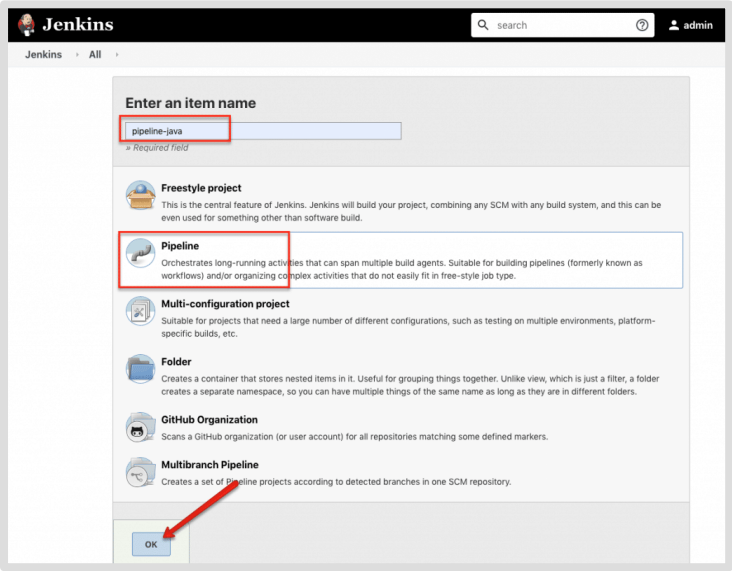
Click **Save**.

3. Creating and Building a Jenkins Pipeline Job

3.1. Go to Jenkins home and select “New Item”



3.2. Give a name, select “Pipeline” and click ok.



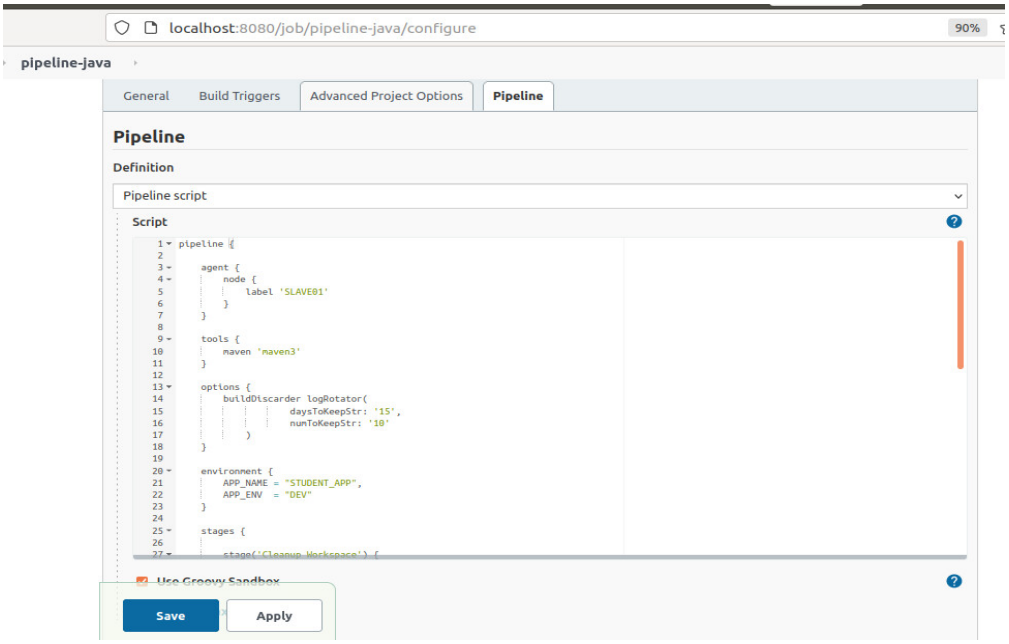
3.3. Scroll down to the Pipeline Script section, copy the whole pipeline code in the script section.

Alternatively, create and edit a file named **Jenkinsfile** directly on GitHub repo. After which, start a browser from your VM to access the **Jenkinsfile** on the GitHub repo. Copy and paste **Jenkinsfile** content directly into the Pipeline Script section.

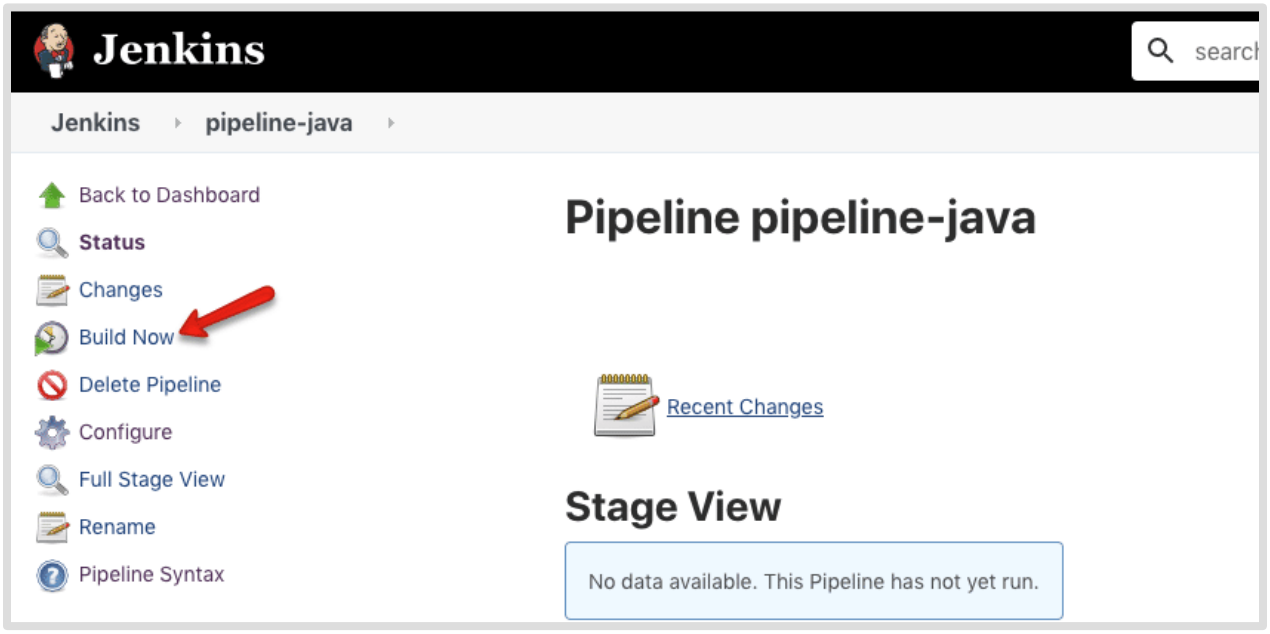
```
pipeline {

    agent {
        node {
            label 'master'
        }
    }

    tools {
        maven 'maven3'
    }
```

```
    options {
        buildDiscarder logRotator(
                    daysToKeepStr: '15',
                    numToKeepStr: '10'
            )
    }

    environment {
        APP_NAME = "STUDENT_APP"
        APP_ENV  = "DEV"
    }

    stages {

        stage('Cleanup Workspace') {
            steps {
                cleanWs()
                sh """
                echo "Cleaned Up Workspace for ${APP_NAME}"
                """
            }
        }

        stage('Code Checkout') {
            steps {
                checkout([
                    $class: 'GitSCM',
                    branches: [[name: '*/main']],
                    userRemoteConfigs: [[url: 'https://github.com/spring-projects/spring-petclinic.git']]
                ])
            }
        }

        stage('Code Build') {
            steps {
                 sh 'mvn install -Dmaven.test.skip=true'
            }
        }

        stage('Printing All Global Variables') {
            steps {
                sh """
                env
                """
            }
        }

    }
}
```

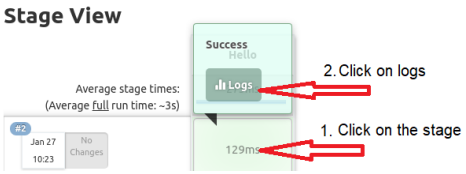
3.3.1 Click on the apply and save button.

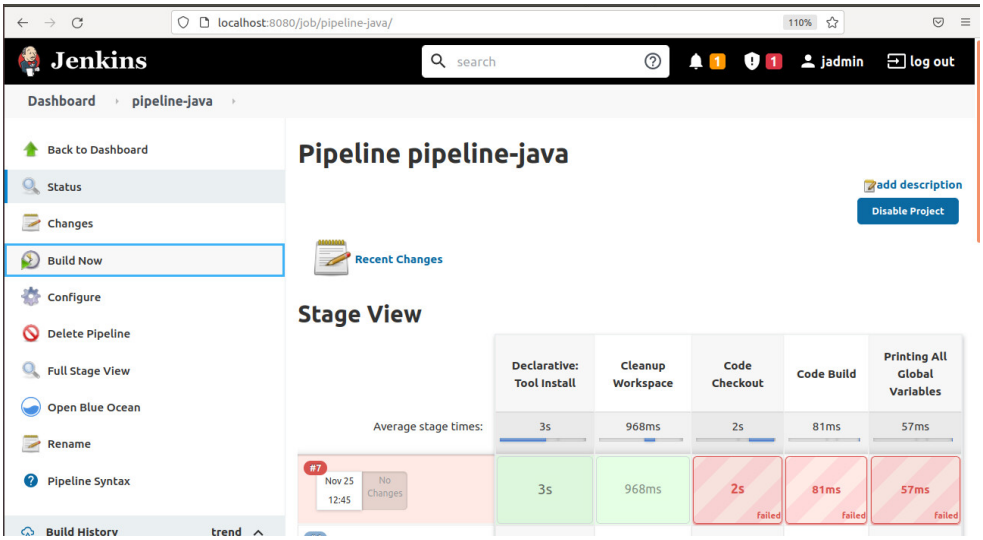


3.4. Now, click “Build Now” and wait for the build to start.



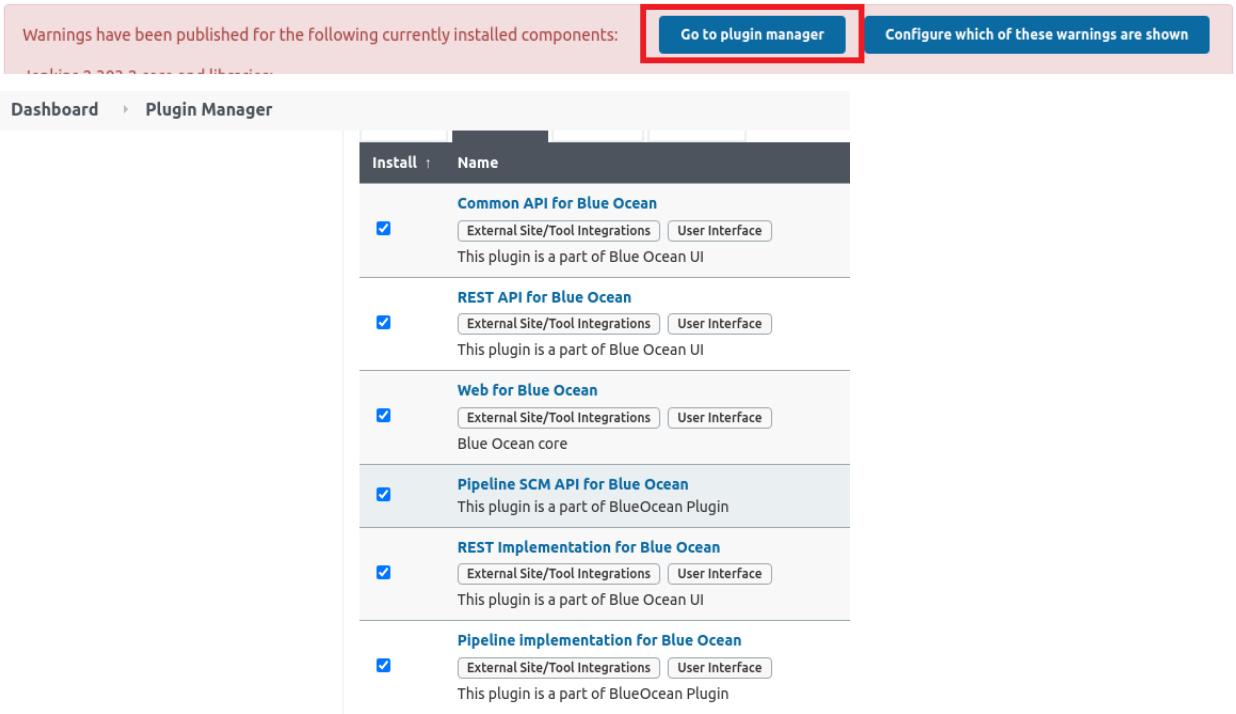
While the job starts you can view each stage executing in stage view. Here is the screenshot of an unsuccessfully executed job. Also, you can see the job logs by clicking the blue icon.

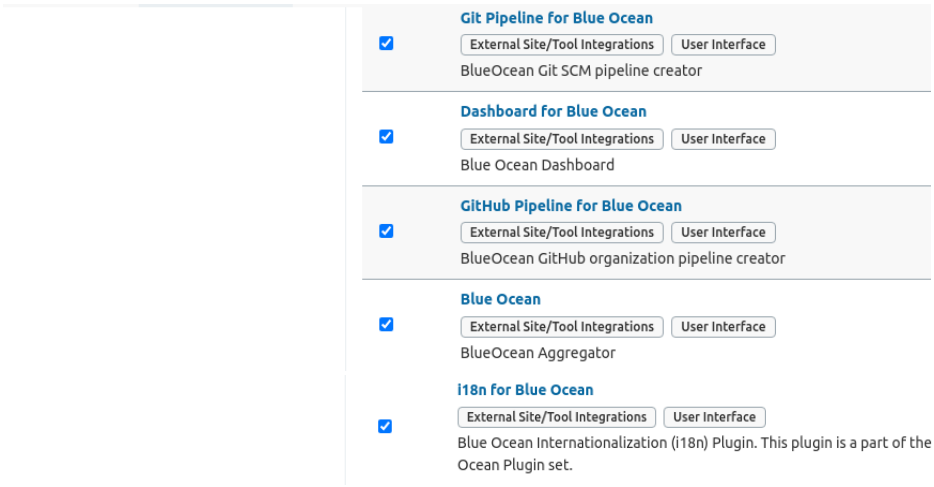




3.4. Install Blue Ocean plugin on Jenkins

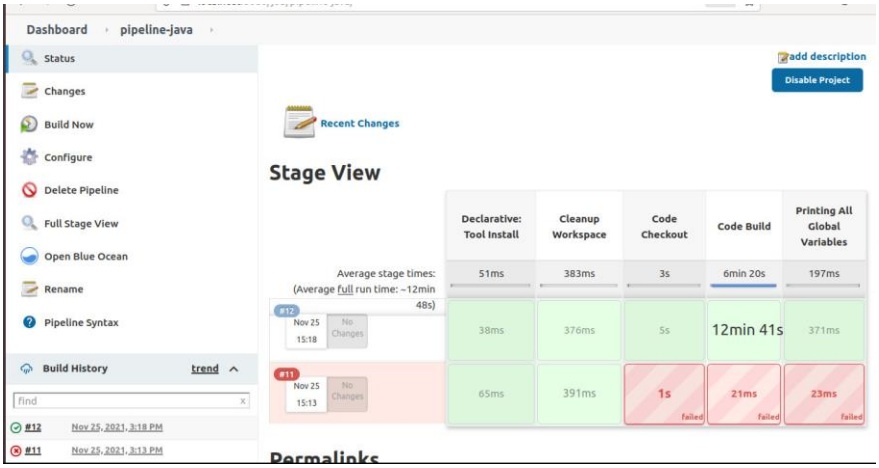
- Select **Manage Jenkins** => **Manage Plugins** => **Select Available Tab** => **Find Blue Ocean Plugin**. Select all blue Ocean related. Then click **install** (Install Without Restart)





- Check if Blue Ocean has been installed successfully by going to any Pipeline and selecting **Open Blue Ocean**. The interface of displaying the history of Build times (Build History)

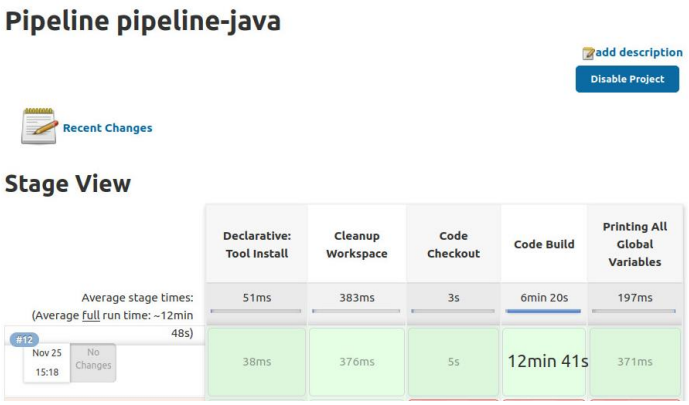
3.5. In the **Build History interface**, select any build and the detailed Pipeline will be displayed.



3.5. Pipeline display on Blue Ocean makes it easy to spot errors in a complex Pipeline.

You can have a very good UI to view your job status and logs as shown above. Use the "Open in Blue Ocean" from the left to open a job in the blue ocean view and view the individual stages of the pipeline. You may encounter errors in **code build** stage as there are version changes for the code to be build.

3.5. Here is the screenshot of a successfully executed job (takes roughly 13 mins to run through the pipeline). Also, you can see the job logs by clicking the blue icon.



Summary:

- Pipeline as code basics
- Building a basic CI pipeline as code for a java app.
- Building a job from pipeline code present in source code repo.
- Executing parallel stages in a pipeline
- Generating pipeline script & directives using Jenkins inbuilt generators.

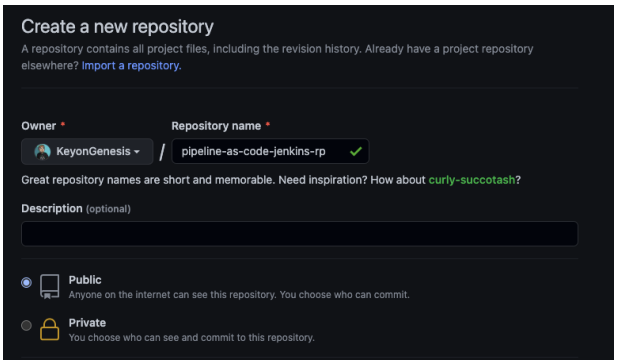
Part 2:

LAB EXERCISE

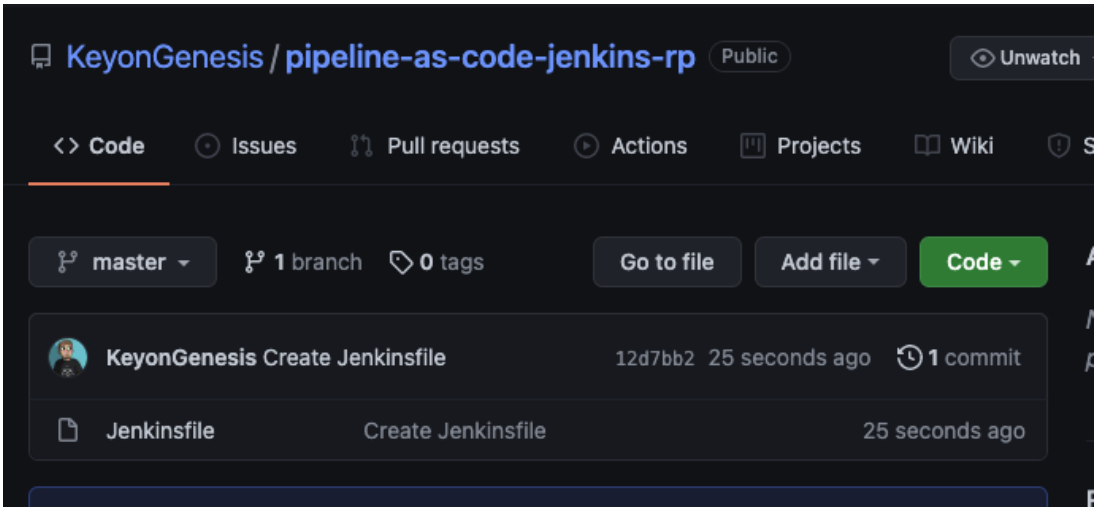
This lab will be executing web applications using Jenkins Pipeline from Github (Jenkinsfile). In Part 1, we used the pipeline script directly on Jenkins. In this section, we will look at how to execute a pipeline script available in an SCM system like Github.

Time to Complete
Approximately 40 Minutes

Executing Jenkins Pipeline From Github (Jenkinsfile)
1. Create a Github repo with our pipeline code in a file named **Jenkinsfile**. The repo can be called : **pipeline-as-code-jenkins-rp** and set it to public.



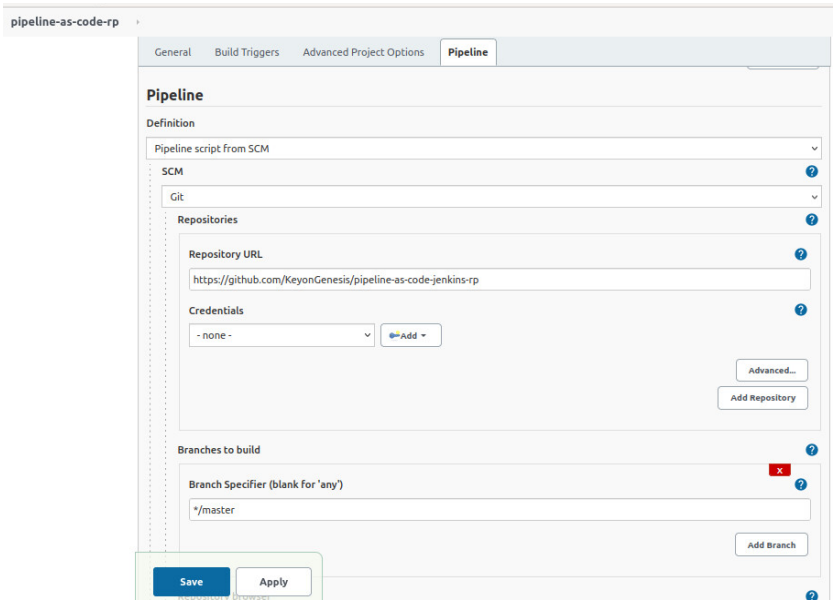
1.1 Do verify that the Jenkins file is created and if needed change the default branch to master.



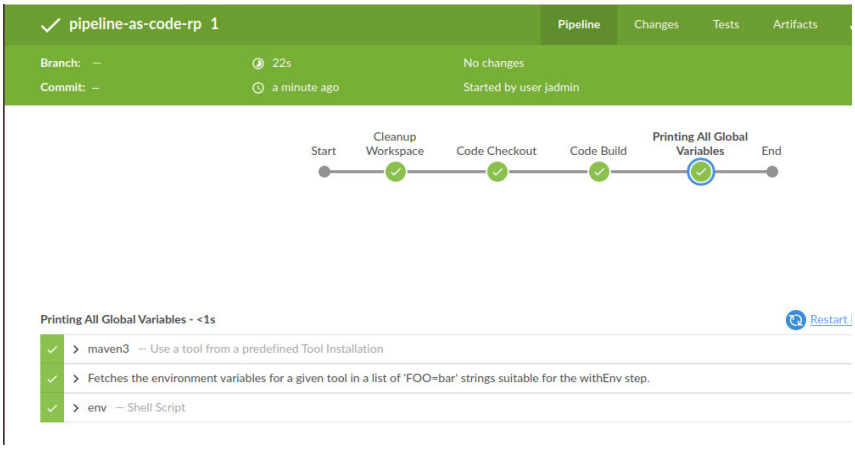
1.2 Edit the Jenkinsfile, copy the code in **Part 1 - 3.3**. Commit the changes and Paste directly to GitHub.

2. Follow the same steps mentioned in Part 1, that was used for creating a pipeline job. But instead of entering the code directly into the script block, select the "**Pipeline script from SCM**" option and fill in the details as shown below.

Definition: Pipeline script from SCM
Repository URL: <your-repo-url>
Script Path: Jenkinsfile



3. Apply and save the configuration and run the build (alternatively you can select Open Blue Ocean). You should see a successful build.



Executing Jenkins Pipeline Stages In Parallel

There are use cases where you have to execute different stages in parallel because each stage will be independent and does not depend on other steps. Also, running separate stages in parallel will reduce the build times as well.

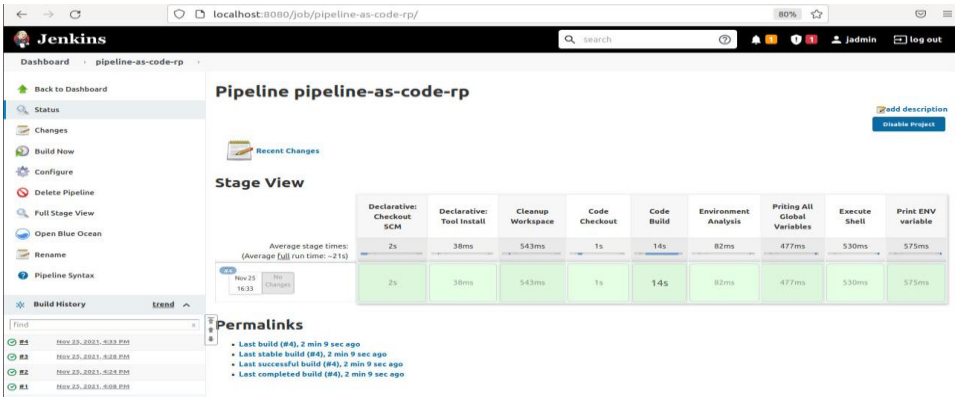
You can achieve parallelism in Jenkins pipelines as code using the parallel block.

4. Modify the Jenkinsfile in your local computer to contain three parallel stages. The code block should be placed after the **stage ('Code Build')**. Don't forget to Commit the changes and Push to GitHub.

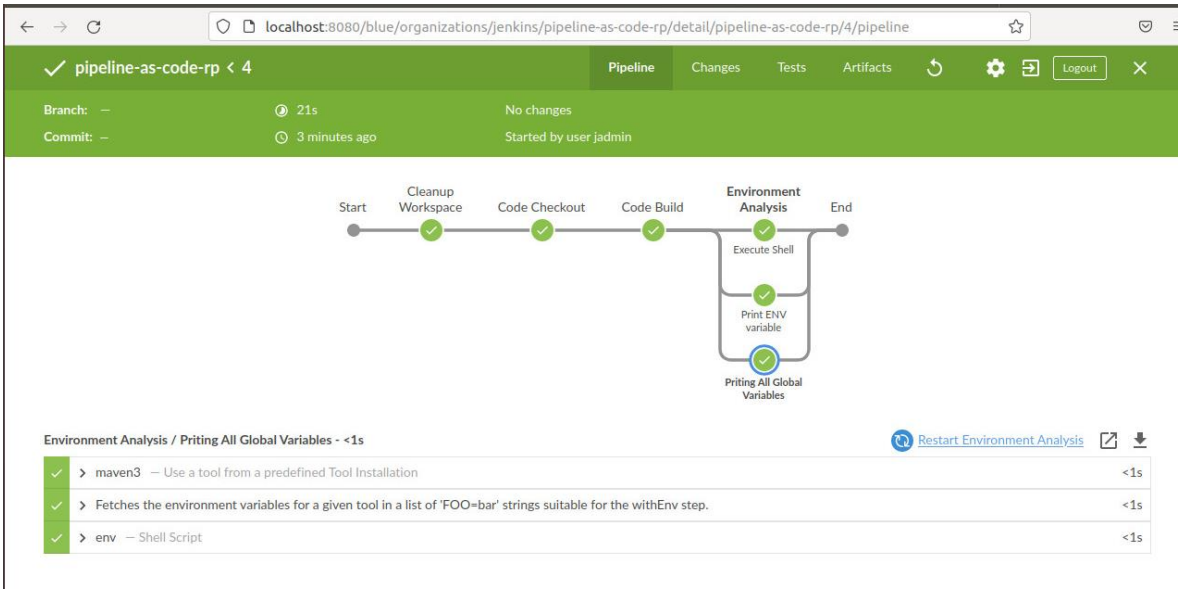
Alternatively, edit the Jenkinsfile to add in the following directly on GitHub.

```
53          stage('Environment Analysis') {
54
55              parallel {
56
57                  stage('Priting All Global Variables') {
58                      steps {
59                          sh """
60                          env
61                          """
62                      }
63                  }
64
65                  stage('Execute Shell') {
66                      steps {
67                          sh 'echo "Hello Student. Thanks for keeping up!"'
68                      }
69                  }
70
71                  stage('Print ENV variable') {
72                      steps {
73                          sh "echo ${APP_ENV}"
74                      }
75                  }
76
77              }
78          }
```

5. On the Jenkins Dashboard, Click on "Build Now".



5.1 Return back to the Blue Ocean Dashboard. You can clearly see the parallel execution on the blue ocean view (Environment Analysis).



--End of Lab Exercise --