

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

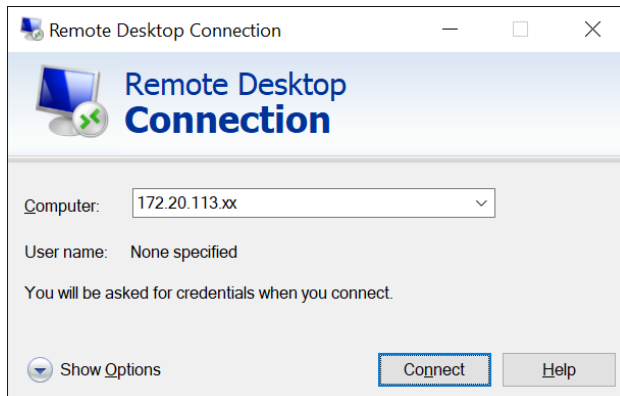
1. Jenkins master (Check Jenkins installation steps)
2. Jenkins slave node connected to the master (Check Jenkins slave setup guide). For this module, we will work on master node (i.e. no slave node).
3. Access to Github.com from your Jenkins server.

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

Republic Polytechnic - School of Infocomm

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.

New version of Jenkins (2.375.2) is available for [download](#) ([changelog](#)).

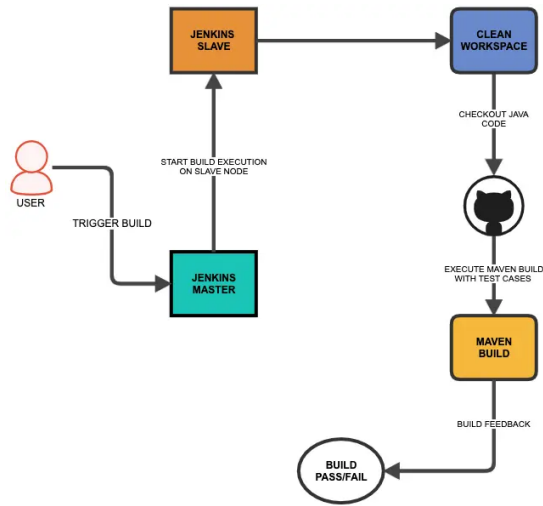
- c)

```
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”.

Maven

Maven installations

Add Maven

Maven

Name

maven3

☒ Install automatically

Install from Apache

Version

3.8.4

Add Installer

Delete Installer

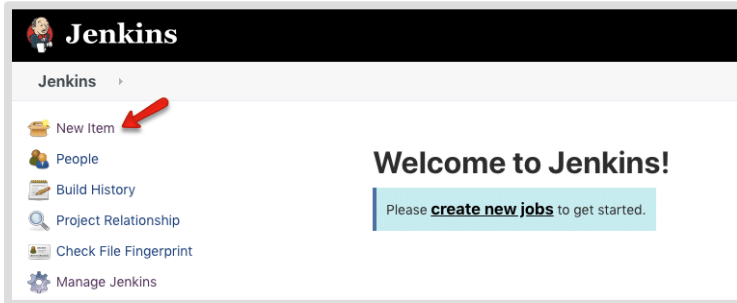
Delete Maven

Save Apply

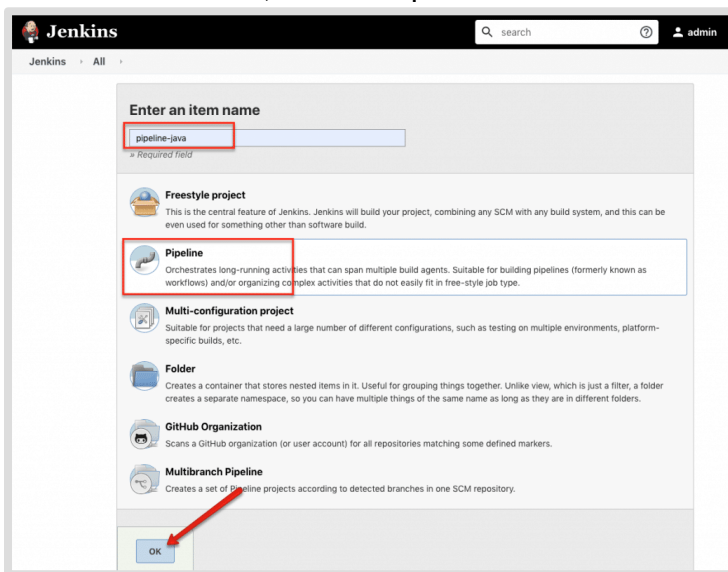
Click **Save**.

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 {

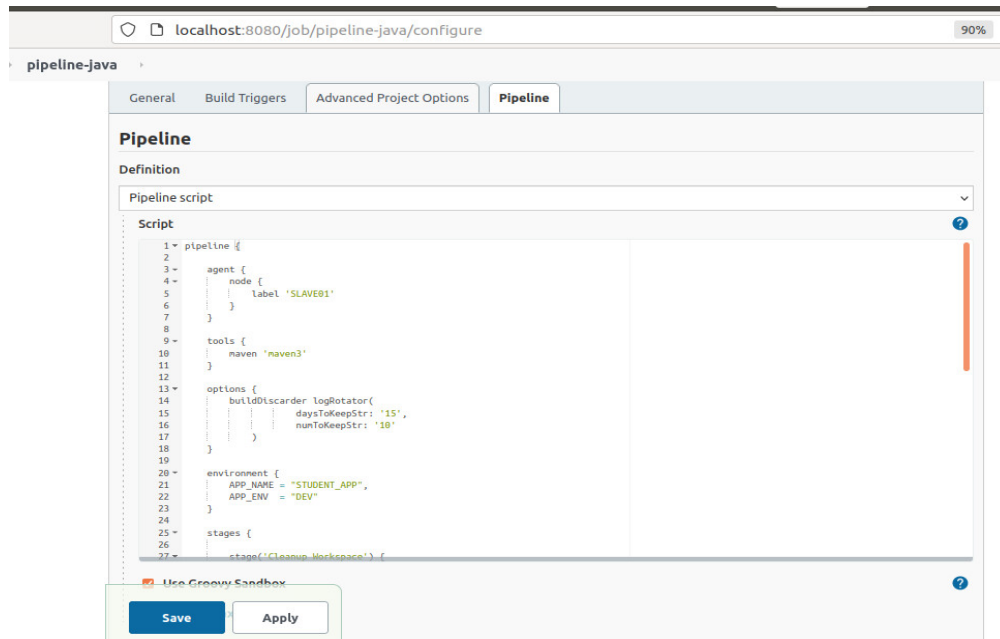
  stage('Cleanup Workspace') {
    steps {
      cleanWs()
      sh """
      echo "Cleaned Up Workspace for ${APP_NAME}"
      """
    }
  }

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

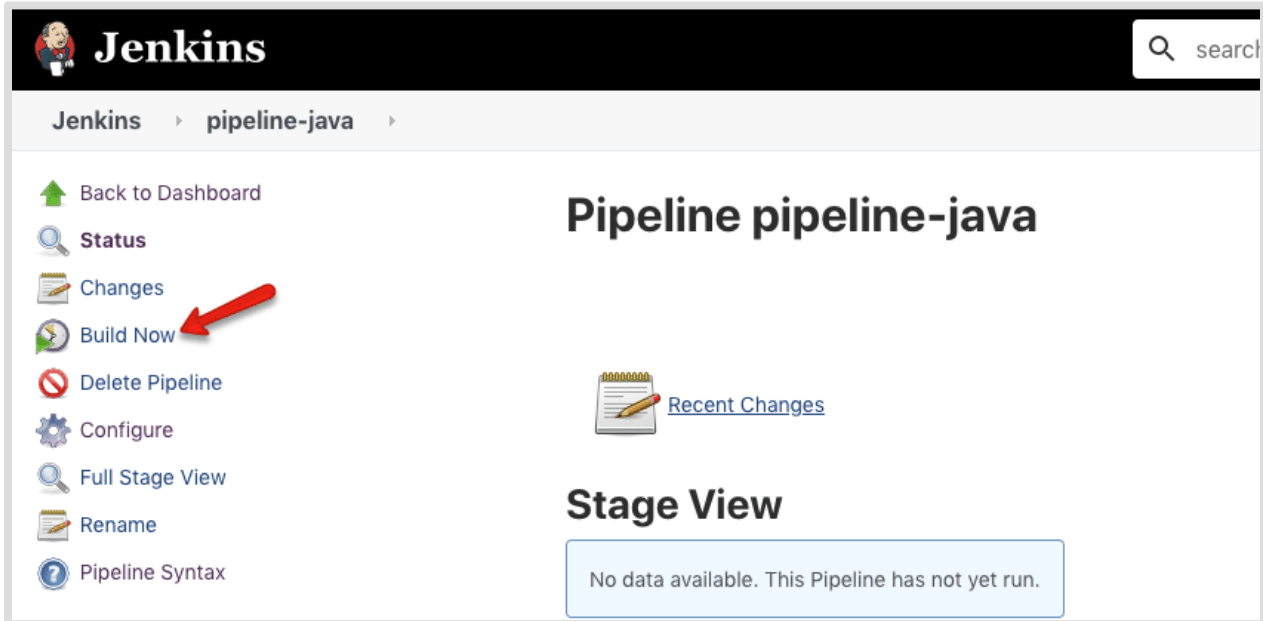
  stage('Code Build') {
    steps {
      sh 'mvn install -Dmaven.test.skip=true'
    }
  }

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

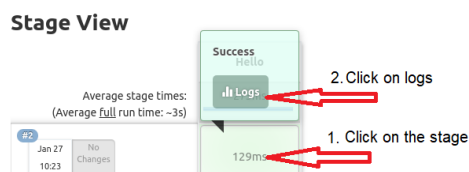
3.3.1 Click on the apply and save button.



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.



Republic Polytechnic - School of Infocomm

The screenshot shows the Jenkins web interface for a pipeline named 'pipeline-java'. The left sidebar contains navigation links: Back to Dashboard, Status, Changes, Build Now (highlighted), Configure, Delete Pipeline, Full Stage View, Open Blue Ocean, Rename, Pipeline Syntax, and Build History. The main area displays the 'Pipeline pipeline-java' with a 'Recent Changes' section and a 'Stage View' table. The 'Stage View' table shows the following stages and their durations:

Stage	Declarative: Tool Install	Cleanup Workspace	Code Checkout	Code Build	Printing All Global Variables
Average stage times:	3s	968ms	2s	81ms	57ms
#7 Nov 25 12:45 No Changes	3s	968ms	2s failed	81ms failed	57ms failed

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)

The screenshot shows the Jenkins Plugin Manager interface. At the top, there is a warning banner: "Warnings have been published for the following currently installed components:" with a "Go to plugin manager" button. Below the banner, the "Plugin Manager" section is visible. The "Install" tab is selected, showing a list of plugins with checkboxes for installation. The following table lists the plugins shown:

Install	Name
<input checked="" type="checkbox"/>	Common API for Blue Ocean External Site/Tool Integrations User Interface This plugin is a part of Blue Ocean UI
<input checked="" type="checkbox"/>	REST API for Blue Ocean External Site/Tool Integrations User Interface This plugin is a part of Blue Ocean UI
<input checked="" type="checkbox"/>	Web for Blue Ocean External Site/Tool Integrations User Interface Blue Ocean core
<input checked="" type="checkbox"/>	Pipeline SCM API for Blue Ocean This plugin is a part of BlueOcean Plugin
<input checked="" type="checkbox"/>	REST Implementation for Blue Ocean External Site/Tool Integrations User Interface This plugin is a part of Blue Ocean UI
<input checked="" type="checkbox"/>	Pipeline implementation for Blue Ocean External Site/Tool Integrations User Interface This plugin is a part of BlueOcean Plugin

☒ **Git Pipeline for Blue Ocean**

External Site/Tool Integrations User Interface

BlueOcean Git SCM pipeline creator

☒ **Dashboard for Blue Ocean**

External Site/Tool Integrations User Interface

Blue Ocean Dashboard

☒ **GitHub Pipeline for Blue Ocean**

External Site/Tool Integrations User Interface

BlueOcean GitHub organization pipeline creator

☒ **Blue Ocean**

External Site/Tool Integrations User Interface

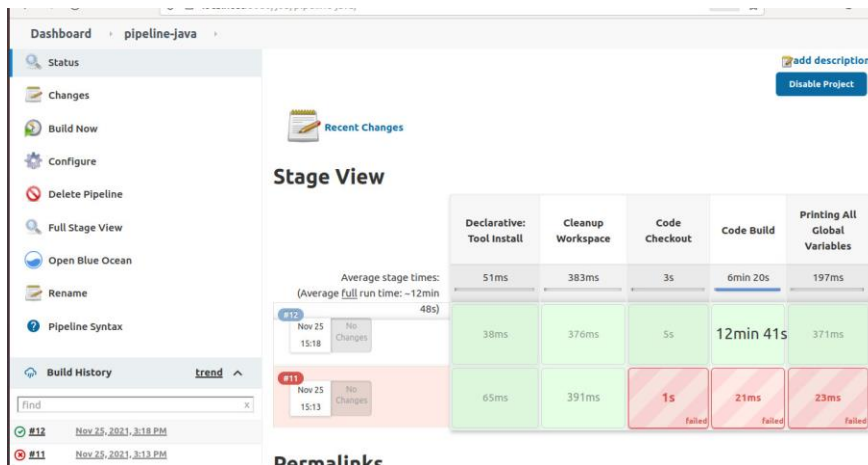
BlueOcean Aggregator

☒ **i18n for Blue Ocean**

External Site/Tool Integrations User Interface

Blue Ocean Internationalization (i18n) Plugin. This plugin is a part of the Blue Ocean Plugin set.

- 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.

Republic Polytechnic - School of Infocomm

The screenshot shows the Jenkins Pipeline view for a job named 'pipeline-java' (run #12). The top bar indicates the pipeline is successful (green checkmark) and shows the branch, duration (12m 48s), and commit information. Below this, a pipeline graph shows five stages: Start, Cleanup Workspace, Code Checkout, Code Build, and Printing All Global Variables, all marked as successful. The 'Printing All Global Variables' stage is expanded, showing its logs: 'maven3' (using a predefined tool installation), 'Fetches the environment variables for a given tool in a list of 'FOO=bar' strings suitable for the withEnv step.', and 'env' (Shell Script).

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.

Pipeline pipeline-java

The screenshot shows the Jenkins Pipeline view for 'pipeline-java'. It includes a 'Recent Changes' section with a 'Recent Changes' button. Below this is the 'Stage View' table, which displays the average stage times and the actual run times for the current job.

	Declarative: Tool Install	Cleanup Workspace	Code Checkout	Code Build	Printing All Global Variables
Average stage times:	51ms	383ms	3s	6min 20s	197ms
(Average full run time: ~12min 48s)					
#12 Nov 25 15:18 No Changes	38ms	376ms	5s	12min 41s	371ms

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

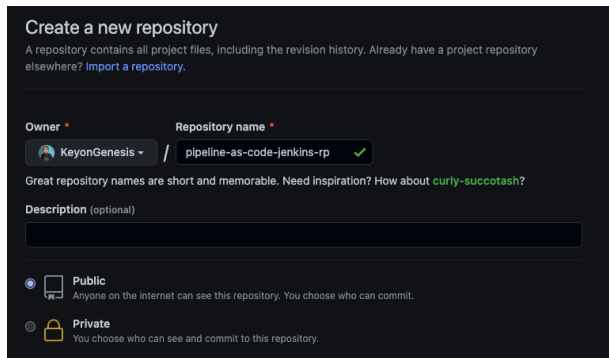
This lab will be executing web applications using Jenkins Pipeline from Github (Jenkinsfile). In Part 1, we used the pipeline script directly on Jenkins. In this section, we will look at how to execute a pipeline script available in an SCM system like Github.

Time to Complete

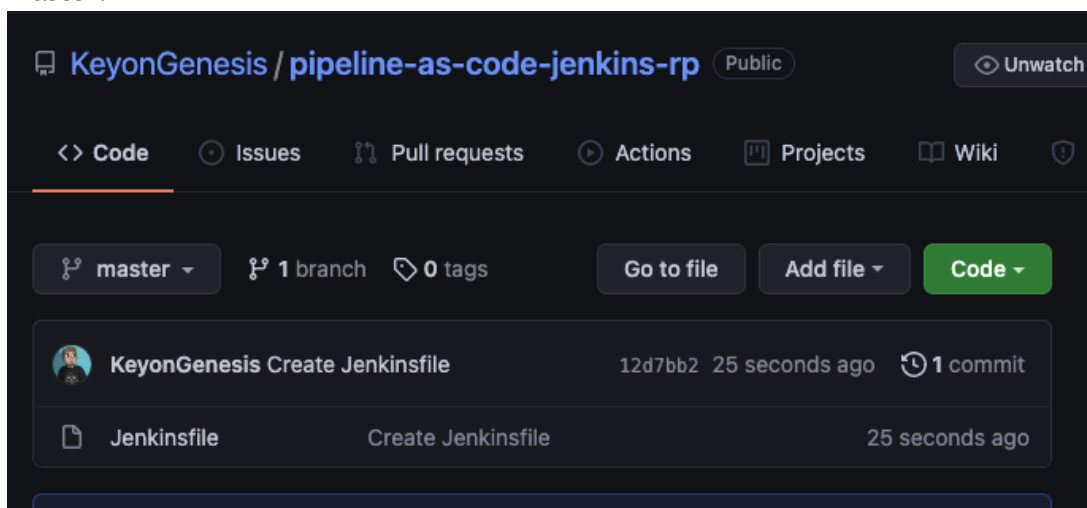
Approximately 40 Minutes

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

The screenshot shows the Jenkins configuration page for a pipeline named 'pipeline-as-code-rp'. The 'Pipeline' tab is selected. Under the 'Definition' section, 'Pipeline script from SCM' is chosen. The 'SCM' is set to 'Git'. The 'Repository URL' is 'https://github.com/KeyonGenesis/pipeline-as-code-jenkins-rp'. The 'Credentials' dropdown is set to 'none'. The 'Branches to build' section has a 'Branch Specifier (blank for 'any')' set to '*/master'. There are 'Save' and 'Apply' buttons at the bottom left, and 'Advanced...', 'Add Repository', and 'Add Branch' buttons on the right.

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

The screenshot shows the Jenkins build page for 'pipeline-as-code-rp 1'. The 'Pipeline' tab is selected. The build status is 'Success' (green checkmark). The build duration is '22s'. The commit message is 'No changes'. The build was started by 'user jadmin' 'a minute ago'. The build steps are: Start, Cleanup Workspace, Code Checkout, Code Build, Printing All Global Variables, and End. The 'Printing All Global Variables' step is highlighted with a blue circle and a green checkmark. Below the build steps, there is a log showing the output of the 'Printing All Global Variables' step, which includes the command 'maven3' and the output 'Fetches the environment variables for a given tool in a list of 'FOO=bar' strings suitable for the withEnv step.' and 'env - Shell Script'.

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('Printing 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”.

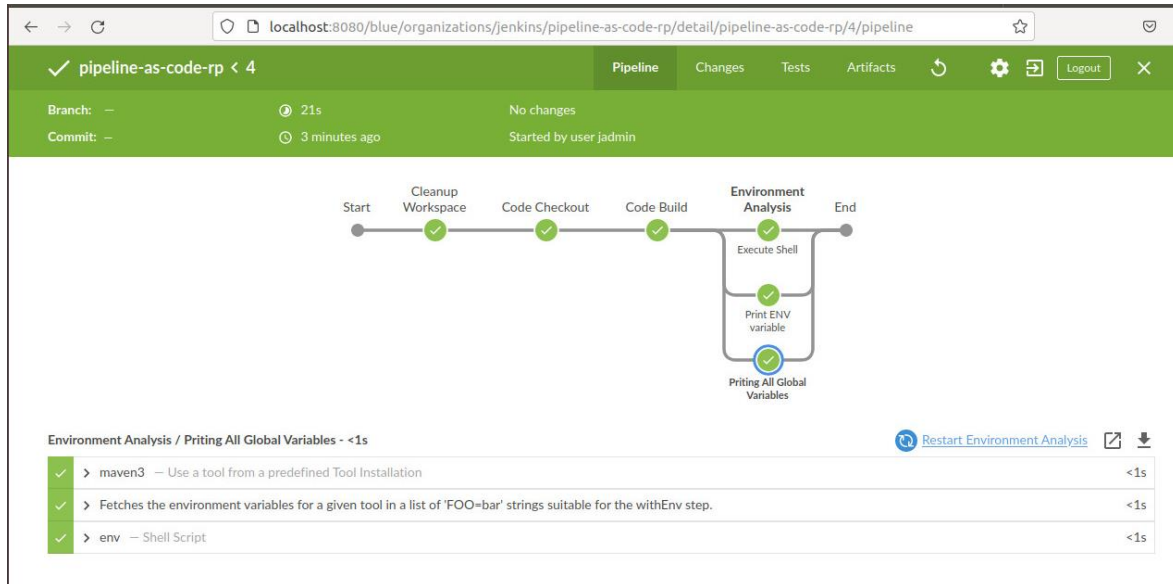
The screenshot shows the Jenkins Dashboard for a pipeline named 'pipeline-as-code-rp'. The 'Build Now' button is visible in the sidebar. The main view displays the 'Stage View' for the pipeline, showing a table of stages and their durations. The stages are: Declarative Checkout SCM (2s), Declarative Tool Install (38ms), Cleanup Workspace (543ms), Code Checkout (1s), Code Build (14s), Environment Analysis (82ms), Printing All Global Variables (477ms), Execute Shell (530ms), and Print ENV variable (575ms). Below the table, there are 'Permalinks' for the last build, last stable build, last successful build, and last completed build, all of which are 2 minutes and 9 seconds ago.

Stage	Duration
Declarative Checkout SCM	2s
Declarative Tool Install	38ms
Cleanup Workspace	543ms
Code Checkout	1s
Code Build	14s
Environment Analysis	82ms
Printing All Global Variables	477ms
Execute Shell	530ms
Print ENV variable	575ms

Permalinks

- Last build (#4), 2 min 9 sec ago
- Last stable build (#4), 2 min 9 sec ago
- Last successful build (#4), 2 min 9 sec ago
- Last completed build (#4), 2 min 9 sec ago

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 --