

**Name - Parmeshwar\***

## Assignment ( Gradle & JenKINS)

### Problem Statement: Automated Deployment of 2-Tier Web application

**Q2) Design and develop tasks required to build CI/CD pipeline using learned technologies to deploy 2-Tier application on cloud platform as a set of microservices on containerized platform such as docker. Deploy a web application on Docker container using GRADLE as build tool to install, test, package application which is fetched from GitHub Repository & deploy through Web Application . Provision Infrastructure on AWS.**

### Step 1) Created Two Instances Of Linux

#### i) One is Master Node

The screenshot displays the AWS Management Console interface. The left sidebar shows the navigation menu with 'Instances' selected. The main content area shows a list of EC2 instances. The 'Jenkins\_Master' instance is highlighted, and its details are shown in the right pane. The instance is running, of type t2.micro, in the us-east-2a availability zone. It has a public IP address of 3.139.7.193 and an Elastic IP of 3.139.7.193. The instance is configured with a Linux/UNIX operating system and a VPC ID of vpc-0579ee0749ed22640.

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)	IPv4 Public IP	IPv6
Jstaging A...	i-02dddf6c2d02c1c2	t2.micro	us-east-2a	running	2/2 checks ...	None	ec2-3-145-157-173.us-...	3.145.157.173	-
Jenkins_Mas...	i-05dd5a6529f8e4056	t2.micro	us-east-2a	running	2/2 checks ...	None	ec2-3-139-7-193.us-ea...	3.139.7.193	-
gradle	i-0b2c9e72ecf334153	t2.micro	us-east-2a	running	2/2 checks ...	None	ec2-3-14-145-28.us-ea...	3.14.145.28	-
bothmasStat	i-099e1ef885b193863	t2.micro	us-east-2a	running	2/2 checks ...	None	ec2-18-220-226-182.us...	18.220.226.182	-
1s	i-038abe911536aee...	t2.micro	us-east-2a	running	2/2 checks ...	None	ec2-18-217-56-36.us-e...	18.217.56.36	-
1M	i-0ce1e316c3741e2db	t2.micro	us-east-2a	running	2/2 checks ...	None	ec2-18-217-85-80.us-e...	18.217.85.80	-

**Instance: i-05dd5a6529f8e4056 (Jenkins\_Master) Elastic IP: 3.139.7.193**

Description	Status Checks	Monitoring	Tags
Instance ID	i-05dd5a6529f8e4056		
Instance state	running		
Instance type	t2.micro		
Finding	Opt-in to AWS Compute Optimizer for recommendations.		
Private DNS	ip-172-31-5-47.us-east-2.compute.internal		
Private IPs	172.31.5.47		
Secondary private IPs			
VPC ID	vpc-0579ee0749ed22640		
Platform	Amazon Linux		
Platform details	Linux/UNIX		
Usage operation	RunInstances		
Source/dest. check	True		
Public DNS (IPv4)	ec2-3-139-7-193.us-east-2.compute.amazonaws.com		
IPv4 Public IP	3.139.7.193		
IPv6 IPs	-		
Elastic IPs	3.139.7.193*		
Availability zone	us-east-2a		
Security groups	all tcp traffic, view inbound rules, view outbound rules		
Scheduled events	No scheduled events		
AMI ID	amzn2-ami-kernel-5.10-hvm-2.0.20220805.0-x86_64-gp2 (ami-0568773882d492fc8)		
Subnet ID	subnet-007c87b20783764a3		
Network interfaces	eth0		
IAM role	-		
Key pair name	realme1		

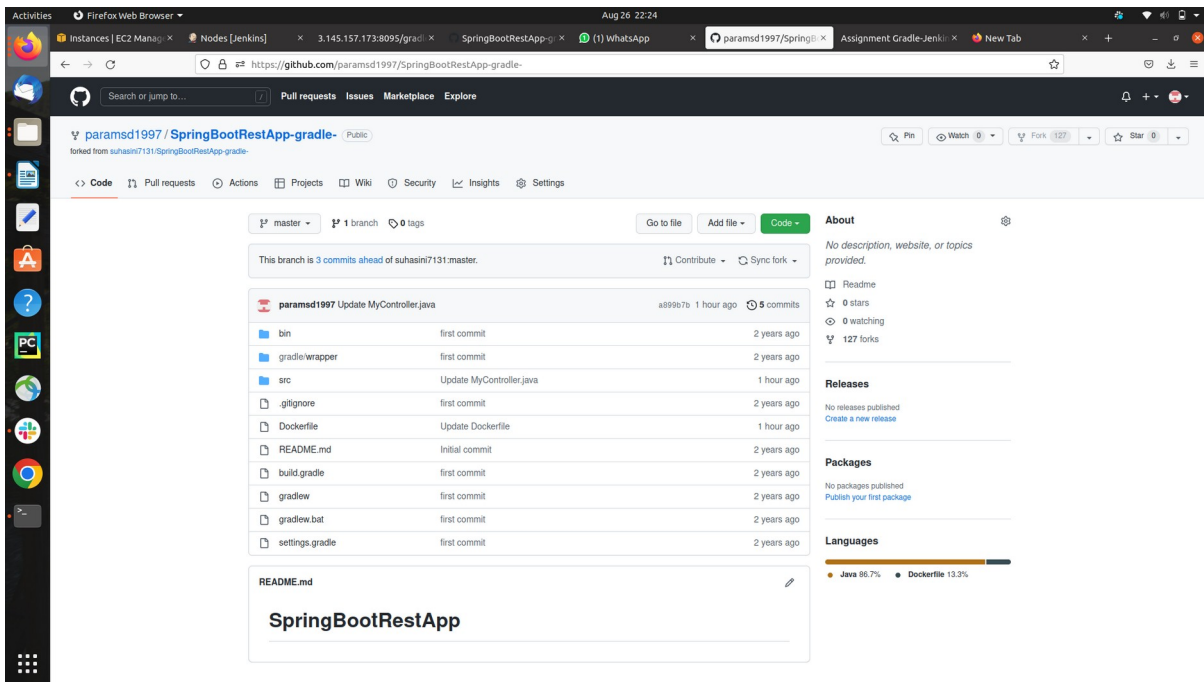
## i) One is tagging (s1) Node

The screenshot shows the AWS Management Console interface. The left sidebar contains navigation options like EC2 Dashboard, EC2 Global View, Events, Tags, Limits, Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, Images, AMIs, AMI Catalog, Elastic Block Store, Volumes, Snapshots, and Lifecycle Manager. The main content area displays the details of an EC2 instance named 'Jstaging A...' with Instance ID 'i-02ddd6c2d02fc1c2'. The instance is in the 'running' state, located in the 'us-east-2a' Availability Zone. It is a 't2.micro' instance type. The console shows various details including Status Checks (2/2 checks passed), Alarm Status (None), Public DNS (IPv4) (ec2-3-145-157-173.us-east-2.compute.amazonaws.com), IPv4 Public IP (3.145.157.173), and IPv6 Public IP (3.139.7.193). The instance is tagged with 'Jstaging A...'. The console also shows the instance's configuration, including the AMI ID (ami-0568773882d4921c8), Subnet ID (subnet-007c87b20783764a3), Network interfaces (eth0), IAM role (realme1), and Owner (556676077223).

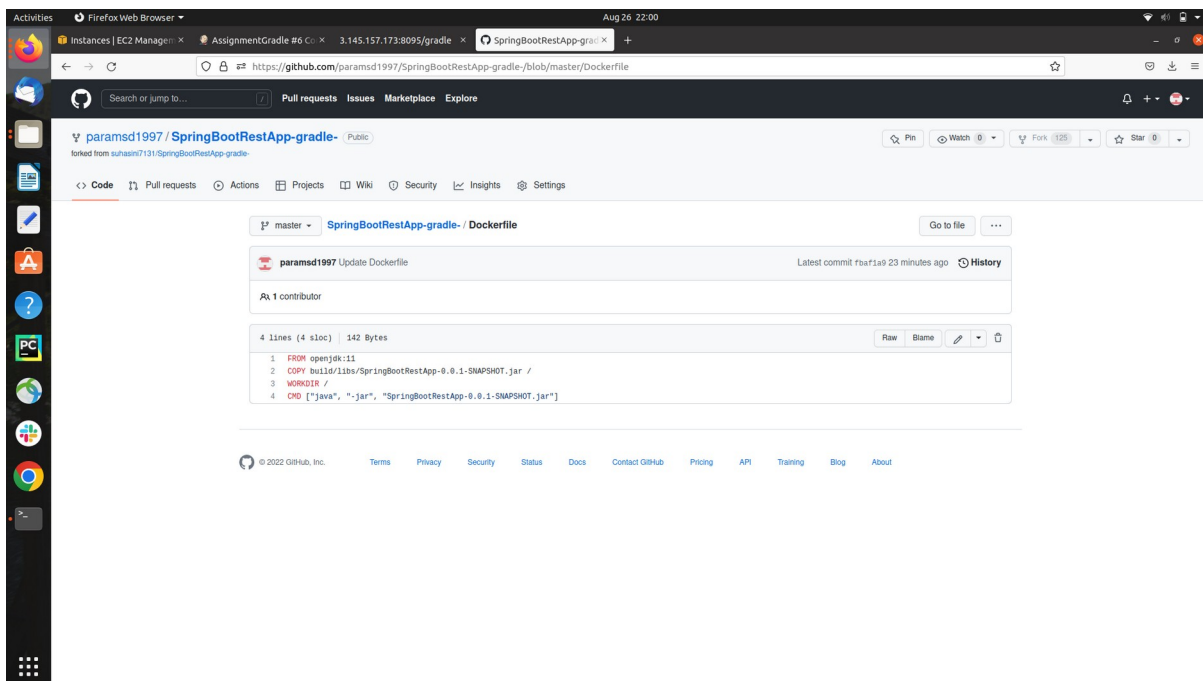
## step 2) Staggle Connected with Master Nodes

The screenshot shows the Jenkins web interface. The top navigation bar includes 'Dashboard', 'Manage Jenkins', and 'Nodes'. The 'Nodes' page is active, displaying a table of managed nodes. The table has columns for 'S', 'Name', 'Architecture', 'Clock Difference', 'Free Disk Space', 'Free Swap Space', 'Free Temp Space', and 'Response Time'. There are two nodes listed: 'Built-in Node' and 's1'. The 's1' node is a Linux (amd64) architecture, in sync, with 21.24 GB of free disk space, 0 B of free swap space, 21.24 GB of free temp space, and a response time of 57ms. The 'Built-in Node' is also a Linux (amd64) architecture, in sync, with 21.27 GB of free disk space, 0 B of free swap space, 21.27 GB of free temp space, and a response time of 0ms. The 'Data obtained' row shows the time taken to obtain data for each node: 7 min 27 sec for 's1' and 7 min 27 sec for 'Built-in Node'.

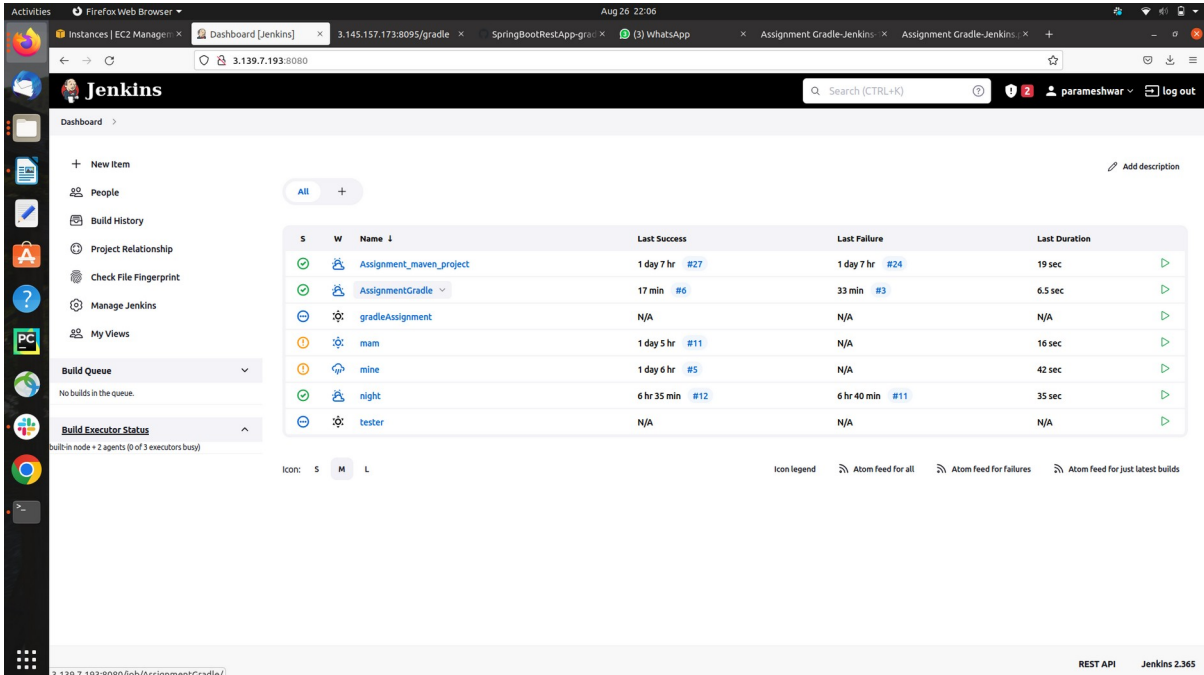
## step 4) Pushed the code to the Github repository



## step 4) Created the Docker file to copy the .jar file and the create Image



## step 5) Build a new job (AssignmentGradle) in the jenkins

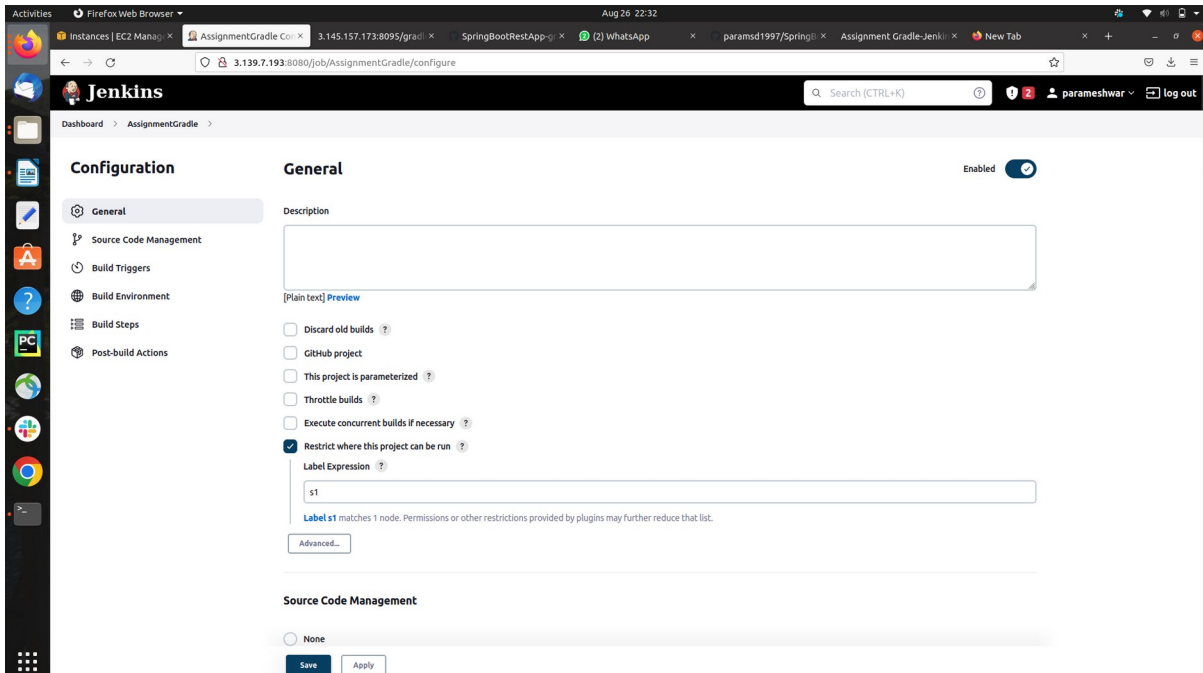


The screenshot shows the Jenkins Dashboard in a Firefox browser window. The dashboard lists several jobs with their status, last success, last failure, and last duration. The jobs are:

S	W	Name	Last Success	Last Failure	Last Duration
✓	🔧	Assignment_maven_project	1 day 7 hr #27	1 day 7 hr #24	19 sec
✓	🔧	AssignmentGradle	17 min #6	33 min #3	6.5 sec
⊖	⚙️	gradleAssignment	N/A	N/A	N/A
⚠️	⚙️	mam	1 day 5 hr #11	N/A	16 sec
⚠️	🔧	mine	1 day 6 hr #5	N/A	42 sec
✓	🔧	night	6 hr 35 min #12	6 hr 40 min #11	35 sec
⊖	⚙️	tester	N/A	N/A	N/A

The left sidebar shows the 'Build Queue' and 'Build Executor Status' sections. The bottom right corner indicates the REST API and Jenkins version 2.365.

## step 6) Configurig the Project New Job



The screenshot shows the Jenkins Configuration page for the 'AssignmentGradle' job. The 'General' tab is selected, and the job is enabled. The configuration includes a description field, a 'Discard old builds' checkbox, a 'GitHub project' checkbox, a 'This project is parameterized' checkbox, a 'Throttle builds' checkbox, an 'Execute concurrent builds if necessary' checkbox, and a 'Restrict where this project can be run' checkbox. The 'Label Expression' field is set to 's1'. The 'Source Code Management' section is currently set to 'None'.

**Configuration**

- General
- Source Code Management
- Build Triggers
- Build Environment
- Build Steps
- Post-build Actions

**General** Enabled

Description

[Plain text] [Preview](#)

☐ Discard old builds ?

☐ GitHub project

☐ This project is parameterized ?

☐ Throttle builds ?

☐ Execute concurrent builds if necessary ?

☒ Restrict where this project can be run ?

Label Expression ?

s1

Label s1 matches 1 node. Permissions or other restrictions provided by plugins may further reduce that list.

[Advanced...](#)

**Source Code Management**

☐ None

[Save](#) [Apply](#)

## Adding the Gradle set up

The screenshot shows the Jenkins Configuration page for a job named 'AssignmentGradle'. The left sidebar contains a 'Configuration' menu with options: General, Source Code Management, Build Triggers, Build Environment, Build Steps, and Post-build Actions. The 'Build Steps' option is selected. The main configuration area is titled 'Invoke Gradle script' and contains the following fields:

- Invoke Gradle**: A radio button that is selected.
- Gradle Version**: A dropdown menu with 'gradle' selected.
- Use Gradle Wrapper**: An unselected radio button.
- Tasks**: A text input field containing 'build'.
- Advanced...**: A button to expand the configuration.

Below the 'Invoke Gradle script' section is the 'Send files or execute commands over SSH' section, which includes:

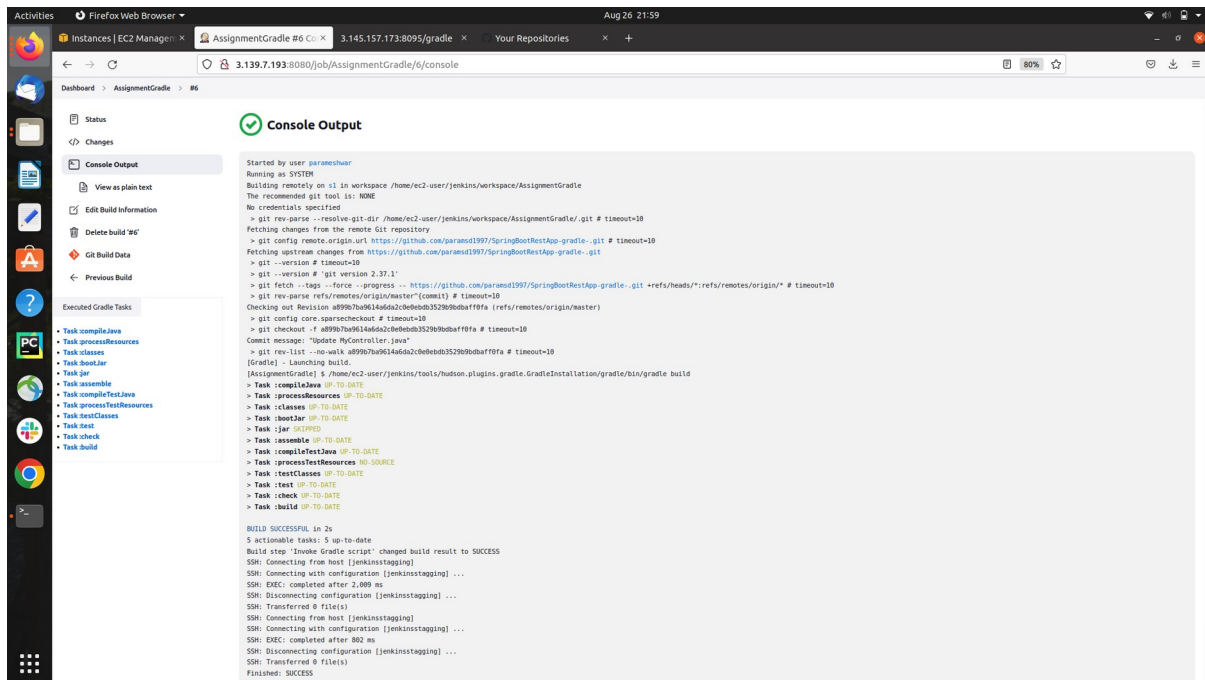
- SSH Publishers**: A section with a 'Name' dropdown set to 'jenkinsstaging' and an 'Advanced...' button.
- Transfers**: A section with a 'Transfer Set' dropdown set to 'Source Files' and a 'Remove prefix' checkbox.
- Save** and **Apply** buttons at the bottom.

## executing the Command Using SSH Server

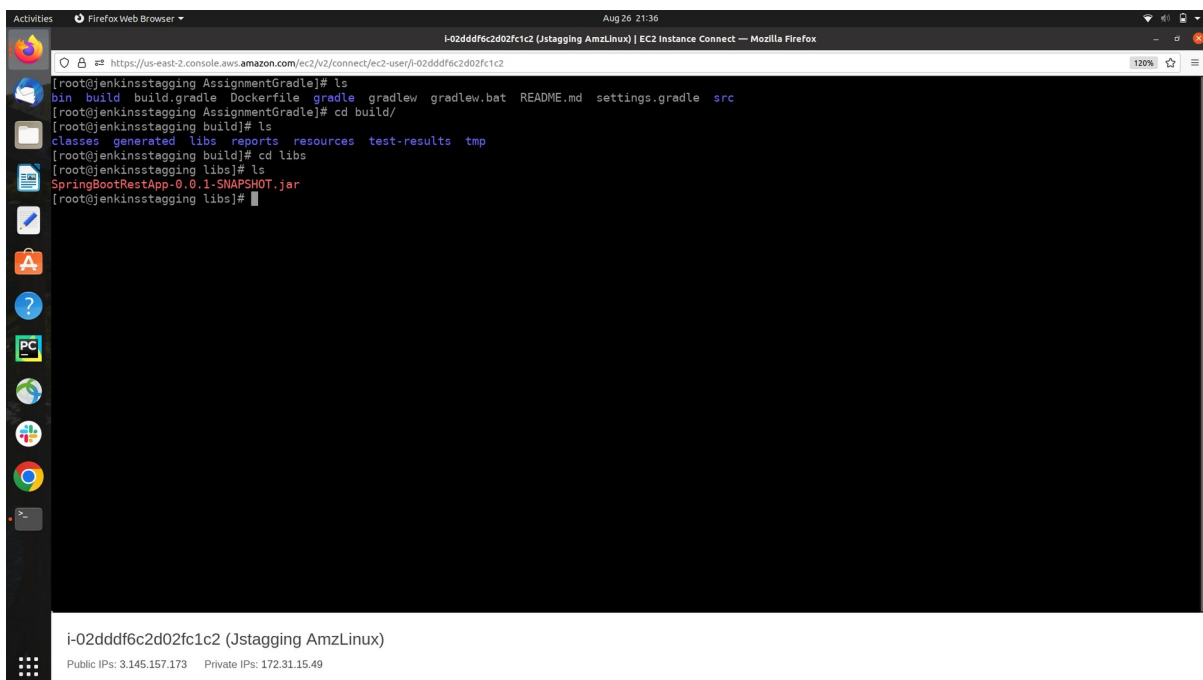
The screenshot shows the Jenkins Configuration page for a job named 'AssignmentGradle'. The left sidebar contains a 'Configuration' menu with options: General, Source Code Management, Build Triggers, Build Environment, Build Steps, and Post-build Actions. The 'Source Code Management' option is selected. The main configuration area is titled 'Source Code Management' and contains the following fields:

- None**: A radio button that is unselected.
- Git**: A radio button that is selected.
- Repositories**: A section with a 'Repository URL' text input field containing 'https://github.com/paramsd1997/SpringBootRestApp-gradle-git', a 'Credentials' dropdown menu set to 'none', an 'Add' button, and an 'Advanced...' button.
- Branches to build**: A section with a 'Branch Specifier (blank for 'any')' text input field containing '\*/master' and an 'Add Branch' button.
- Repository browser**: A section with a 'Repository browser' dropdown menu.
- Save** and **Apply** buttons at the bottom.

## step 7) after Built the project Ouput got successful

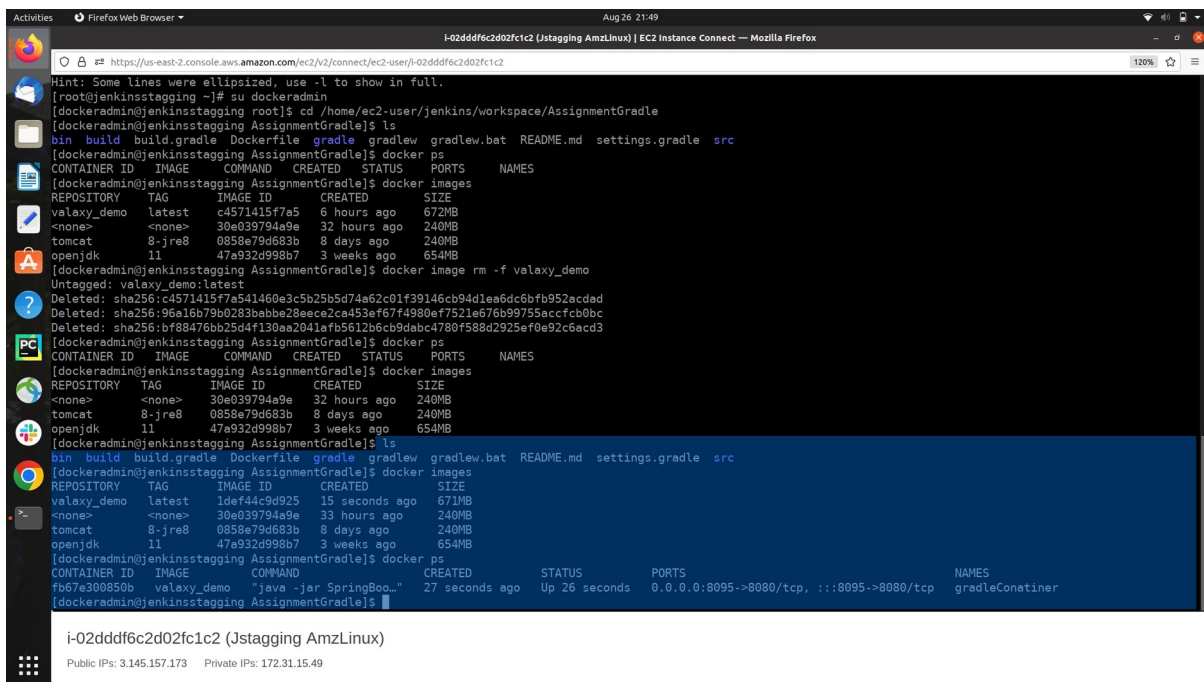


## step 8) After Building we .jar file got created into build/libs





## step 9) After getting .jar file Launching the Conatiner and creating the image using Dockerfile

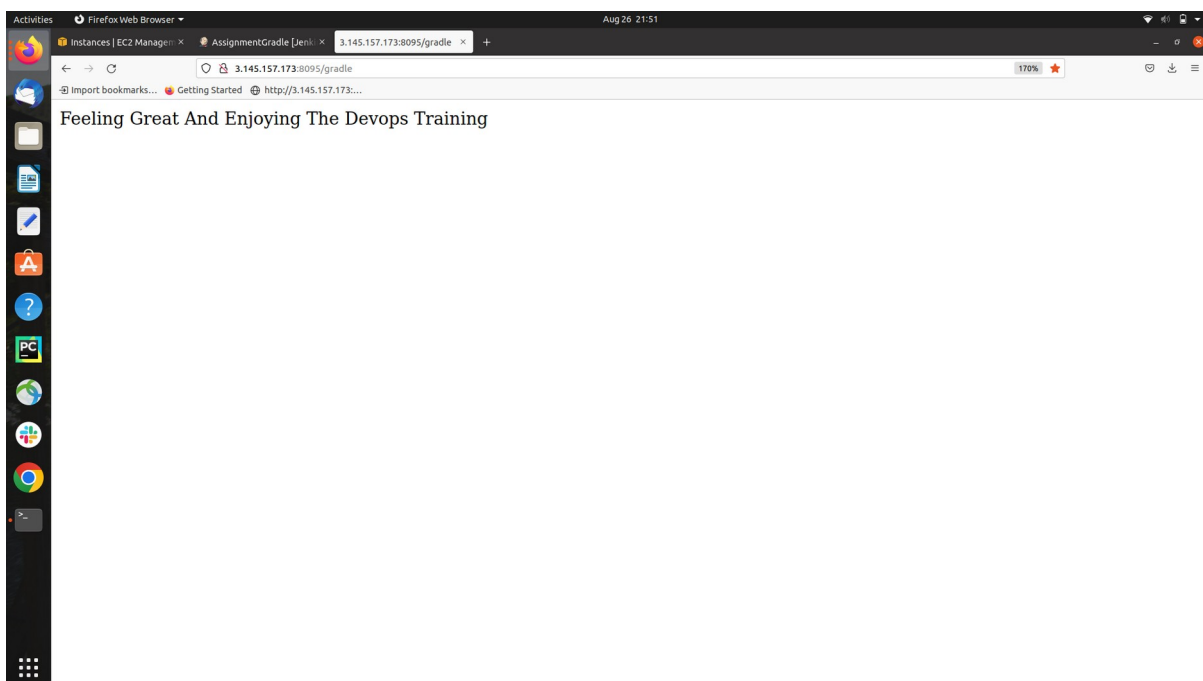


```
Hint: Some lines were ellipsized, use -l to show in full.
[root@jenkinsstaging ~]# su dockeradmin
[dockeradmin@jenkinsstaging root]$ cd /home/ec2-user/jenkins/workspace/AssignmentGradle
[dockeradmin@jenkinsstaging AssignmentGradle]$ ls
bin build build.gradle Dockerfile gradle gradlew gradlew.bat README.md settings.gradle src
[dockeradmin@jenkinsstaging AssignmentGradle]$ docker ps
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS        PORTS        NAMES
[dockeradmin@jenkinsstaging AssignmentGradle]$ docker images
REPOSITORY    TAG        IMAGE ID      CREATED        SIZE
valaxy_demo   latest    c4571415f7a5  6 hours ago    672MB
<none>        <none>     30e039794a9e  32 hours ago    240MB
tomcat        8-jre8    0858e79d683b  8 days ago     240MB
openjdk       11        47a932d998b7  3 weeks ago    654MB
[dockeradmin@jenkinsstaging AssignmentGradle]$ docker image rm -f valaxy_demo
Untagged: valaxy_demo:latest
Deleted: sha256:c4571415f7a541460a3c5b25b5d74a62c01f39146cb94d1ea6dc6bfb952acdada
Deleted: sha256:96a16b79b0283babbe28eece2ca453ef67f4980ef7521e676b99755accfcb0bc
Deleted: sha256:b788476bb25d4f130aa2041afb5612b6cb9dabc4780f588d2925ef0e92c6acd3
[dockeradmin@jenkinsstaging AssignmentGradle]$ docker ps
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS        PORTS        NAMES
[dockeradmin@jenkinsstaging AssignmentGradle]$ docker images
REPOSITORY    TAG        IMAGE ID      CREATED        SIZE
valaxy_demo   latest    1def44c9d925  15 seconds ago  671MB
<none>        <none>     30e039794a9e  33 hours ago    240MB
tomcat        8-jre8    0858e79d683b  8 days ago     240MB
openjdk       11        47a932d998b7  3 weeks ago    654MB
[dockeradmin@jenkinsstaging AssignmentGradle]$ ls
bin build build.gradle Dockerfile gradle gradlew gradlew.bat README.md settings.gradle src
[dockeradmin@jenkinsstaging AssignmentGradle]$ docker images
REPOSITORY    TAG        IMAGE ID      CREATED        SIZE
valaxy_demo   latest    1def44c9d925  15 seconds ago  671MB
<none>        <none>     30e039794a9e  33 hours ago    240MB
tomcat        8-jre8    0858e79d683b  8 days ago     240MB
openjdk       11        47a932d998b7  3 weeks ago    654MB
[dockeradmin@jenkinsstaging AssignmentGradle]$ docker ps
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS        PORTS        NAMES
fb67e308850b   valaxy_demo  "java -jar SpringBoo..."  27 seconds ago  Up 26 seconds  0.0.0.0:8095->8080/tcp, :::8095->8080/tcp  gradleConatiner
[dockeradmin@jenkinsstaging AssignmentGradle]$
```

i-02dddf6c2d02fc1c2 (Jstaggging AmzLinux)

Public IPs: 3.145.157.173 Private IPs: 172.31.15.49

## step 10 ) Accessing the Server using docker container



THANK YOU !!