

Finance Me

Banking and Finance Domain

Finance me is a project which is used for banking services. It uses java code for developing application. The application is automated using CI/CD pipeline. The flow of the project is explained in detail.

To implement Continuous Integration & Continuous Deployment using following tools:

- ❖ Git - For version control for tracking changes in the code files.
- ❖ Maven – For Continuous Build.
- ❖ Jenkins - For continuous integration and continuous deployment.
- ❖ Docker - For deploying containerized applications.
- ❖ Ansible - Configuration management tools.
- ❖ Selenium - For automating tests on the deployed web application.
- ❖ Terraform - For creation of infrastructure.
- ❖ Prometheus and Grafana – For Automated Monitoring and Report Visualization.

Created an EC2 instance with name Master-server and configured java, maven, docker, Jenkins, Ansible and Terraform on the instance, this server acts as master server, which is used to implement CI/CD pipeline script.

These are details of master-server

Public IP: 13.233.102.84

IAM Role: srija with EC2 Full Access.

Java Version: jdk-17

Maven version: maven 3.6.3.

Ansible Version: 2.14.5

Terraform Version: v1.4.6

Docker Version: 20.10.21

Jenkins configured on URL: 13.233.102.84:8080

The screenshot shows the AWS CloudWatch Metrics interface. On the left, there's a navigation pane with links like 'New EC2 Experience', 'EC2 Dashboard', 'EC2 Global View', 'Events', 'Tags', 'Limits', 'Instances' (which is expanded to show 'Instances', 'Instance Types', 'Launch Templates', 'Spot Requests', 'Savings Plans', 'Reserved Instances', 'Dedicated Hosts', 'Capacity Reservations', and 'Images'), and 'Images'. The main content area displays a table for an EC2 instance with the ID i-05578e7537bce2225. The table columns include: Instance ID, Public IPv4 address, Private IPv4 addresses; IPv6 address, Instance state, Public IPv4 DNS; Hostname type, Private IP DNS name (IPv4 only), Elastic IP addresses; IP name, Instance type, AWS Compute Optimizer finding; Answer private resource DNS name, IPv4 (A), Auto Scaling Group name; Auto-assigned IP address, VPC ID, IAM Role; and IAM Role, Subnet ID. At the bottom, there are links for 'CloudShell', 'Feedback', 'Language', and copyright information from 2023, along with system status icons and a search bar.

```

root@ip-172-31-0-4:/home/ubuntu# docker --version
Docker version 20.10.21, build 20.10.21-0ubuntu1-22.04.3
root@ip-172-31-0-4:/home/ubuntu# ansible --version
ansible [core 2.14.5]
  config file = /etc/ansible/ansible.cfg
  configured module search path = ['/root/.ansible/plugins/modules', '/usr/share/ansible/plugins/modules']
  ansible python module location = /usr/lib/python3/dist-packages/ansible
  ansible collection location = /root/.ansible/collections:/usr/share/ansible/collections
  executable location = /usr/bin/ansible
  python version = 3.10.6 (main, Mar 10 2023, 10:55:28) [GCC 11.3.0] (/usr/bin/python3)
  jinja version = 3.0.3
  libyaml = True
root@ip-172-31-0-4:/home/ubuntu# terraform --version
Terraform v1.4.6
on linux_amd64
root@ip-172-31-0-4:/home/ubuntu# java --version
openjdk 17.0.6 2023-01-17
OpenJDK Runtime Environment (build 17.0.6+10-Ubuntu-0ubuntu122.04)
OpenJDK 64-Bit Server VM (build 17.0.6+10-Ubuntu-0ubuntu122.04, mixed mode, sharing)
root@ip-172-31-0-4:/home/ubuntu# mvn --version
Apache Maven 3.6.3
Maven home: /usr/share/maven
Java version: 17.0.6, vendor: Private Build, runtime: /usr/lib/jvm/java-17-openjdk-amd64
Default locale: en, platform encoding: UTF-8
OS name: "linux", version: "5.15.0-1031-aws", arch: "amd64", family: "unix"
root@ip-172-31-0-4:/home/ubuntu#

```

CloudShell Feedback Language © 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences ENG IN 16:51 28-04-2023

Here are the screenshots of master-server connected on browser displaying all the versions.

```

root@ip-172-31-0-4:/home/ubuntu# systemctl status jenkins
● jenkins.service - Jenkins Continuous Integration Server
  Loaded: loaded (/lib/systemd/system/jenkins.service; enabled; vendor preset: enabled)
  Active: active (running) since Fri 2023-04-28 10:52:57 UTC; 29min ago
    Main PID: 17946 (java)
       Tasks: 48 (limit: 4686)
      Memory: 1.2G
        CPU: 1min 9.060s
       CGroup: /system.slice/jenkins.service
               └─17946 /usr/bin/java -Djava.awt.headless=true -jar /usr/share/java/jenkins.war --webroot=/var/cache/jenkins/war --httpPort=8080

Apr 28 11:02:11 ip-172-31-0-4 jenkins[17946]: 2023-04-28 11:02:11.691+0000 [id=103]           WARNING   o.j.p.p.DiskUsageCollector#collect: Cannot >
Apr 28 11:04:11 ip-172-31-0-4 jenkins[17946]: 2023-04-28 11:04:11.673+0000 [id=112]           WARNING   o.j.p.p.DiskUsageCollector#collect: Cannot >
Apr 28 11:06:11 ip-172-31-0-4 jenkins[17946]: 2023-04-28 11:06:11.687+0000 [id=121]           WARNING   o.j.p.p.DiskUsageCollector#collect: Cannot >
Apr 28 11:08:11 ip-172-31-0-4 jenkins[17946]: 2023-04-28 11:08:11.664+0000 [id=140]           WARNING   o.j.p.p.DiskUsageCollector#collect: Cannot >
Apr 28 11:10:11 ip-172-31-0-4 jenkins[17946]: 2023-04-28 11:10:11.671+0000 [id=149]           WARNING   o.j.p.p.DiskUsageCollector#collect: Cannot >
Apr 28 11:12:11 ip-172-31-0-4 jenkins[17946]: 2023-04-28 11:12:11.671+0000 [id=160]           WARNING   o.j.p.p.DiskUsageCollector#collect: Cannot >
Apr 28 11:14:11 ip-172-31-0-4 jenkins[17946]: 2023-04-28 11:14:11.665+0000 [id=170]           WARNING   o.j.p.p.DiskUsageCollector#collect: Cannot >
Apr 28 11:16:11 ip-172-31-0-4 jenkins[17946]: 2023-04-28 11:16:11.668+0000 [id=179]           WARNING   o.j.p.p.DiskUsageCollector#collect: Cannot >
Apr 28 11:18:11 ip-172-31-0-4 jenkins[17946]: 2023-04-28 11:18:11.665+0000 [id=206]           WARNING   o.j.p.p.DiskUsageCollector#collect: Cannot >
Apr 28 11:20:11 ip-172-31-0-4 jenkins[17946]: 2023-04-28 11:20:11.666+0000 [id=215]           WARNING   o.j.p.p.DiskUsageCollector#collect: Cannot >

lines 1-20/20 (END)

```

CloudShell Feedback Language © 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences ENG IN 16:51 28-04-2023

The code is pushed onto Github and the repository URL is:

<https://github.com/Srija1991/Finance-me.git>

Finance-me at master · Srija1991 · [+ New Pull Request](#)

[Simple-DevOps-Pro... Dec-3rd Batch - Go... Docker Hub Architecture Comp... Application 3 tier A... Jenkins Tomcat Dep... Devops GI \(03 Dec-... Book Your Appoint... Weekly Feedback F...](#)

master · Finance-me

[mvnw.cmd](#) finance-me committed 6 months ago

[pom.xml](#) finance-me committed 6 months ago

[selejarfinance.jar](#) jar 2 hours ago

[README.md](#)

Finance-me

Finance application Application is a micro service based application developed using CI/CD pipeline using following tools:

1. Git - For version control for tracking changes in the code files
2. Maven – For Continuous Build
3. Jenkins - For continuous integration and continuous deployment
4. Docker - For deploying containerized applications
5. Ansible - Configuration management tools
6. Selenium - For automating tests on the deployed web application
7. Terraform - For creation of infrastructure. 8.Prometheus and Grafana – For Automated Monitoring and Report Visualization.

The new job named Finance-me is created and added pipelineSCM for pipeline definition.

The SCM used here is Git and URL is given as shown.

Not secure | ec2-13-233-102-84.ap-south-1.compute.amazonaws.com:8080/job/Finance-me/configure

[Simple-DevOps-Pro... Dec-3rd Batch - Go... Docker Hub Architecture Comp... Application 3 tier A... Jenkins Tomcat Dep... Devops GI \(03 Dec-... Book Your Appoint... Weekly Feedback F...](#)

Jenkins

Search (CTRL+K)

Dashboard > Finance-me > Configuration

Configure

General

Enabled

Description

Banking and Finance Project

[Plain text] [Preview](#)

Discard old builds ?

Do not allow concurrent builds

Do not allow the pipeline to resume if the controller restarts

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

The screenshot shows the Jenkins Pipeline configuration page for a project named "Finance-me". The "Pipeline" tab is selected. Under the "Definition" section, it is set to "Pipeline script from SCM". The "SCM" dropdown is set to "Git". The "Repository URL" field contains "https://github.com/Srija1991/Finance-me.git". The "Credentials" dropdown is set to "- none -". At the bottom, there are "Save" and "Apply" buttons.

The pipeline script is provided through file named *Jenkinsfile*.

The screenshot shows the Jenkins Pipeline configuration page for the same project. The "Pipeline" tab is selected. In the "Script Path" field, the value "Jenkinsfile" is entered and circled with a red marker. Below this field is a checked checkbox for "Lightweight checkout". At the bottom, there are "Save" and "Apply" buttons.

The repository is shown here with Dockerfile, Jenkinsfile, Jenkinsfile2 , folders test-server and prod-server for creating and deploying application onto test-server and production server respectively.

The screenshot shows a GitHub repository page for 'Finance-me'. The commit history lists several commits from 'Srija1991' and 'Shubhamkushwah123'. Key commits include 'Update Jenkinsfile2', 'Update Jenkinsfile', and 'Update Jenkinsfile2' again. The repository details sidebar shows 0 stars, 1 watching, and 0 forks. The releases section indicates no releases have been published. The packages section also shows no packages have been published. The contributors section lists Srija1991 and Shubhamkushwah123.

Here is the pipeline script written in file named Jenkinsfile. The flow of the script is as follows:

1. Checkout the code from git.

The screenshot shows the Jenkinsfile content on GitHub. The file starts with a pipeline block, specifying an agent and tools (maven and terraform). It then defines stages, with one stage named 'checkout' containing steps to echo a message and run a git command to clone the repository from 'https://github.com/Srija1991/Finance-me.git'.

```

1 pipeline {
2   agent any
3
4   tools {
5     maven 'maven'
6     terraform 'terraform'
7   }
8   stages {
9     stage('checkout'){
10       steps {
11         echo 'checkout the code from GitRepo'
12         git 'https://github.com/Srija1991/Finance-me.git'
13       }
14     }
15   }
16 }
```

2. Build the application using maven build tool using command “mvn clean package”.
3. Publish the HTML reports.
4. Create the docker using jar file and Dockerfile.
5. Push image to docker hub using appropriate credentials.

```

10 steps {
11     echo 'Checkout the code from GitRepo'
12     git 'https://github.com/Srija1991/Finance-me.git'
13 }
14
15 stage('Build the Application'){
16     steps {
17         echo "Cleaning.... Compiling.....Testing.....Packaging"
18         sh 'mvn clean package'
19     }
20 }
21 stage('publish Reports'){
22     steps {
23         publishHTML([allowMissing: false, alwaysLinkToLastBuild: false, keepAll: false, reportDir: '/var/lib/jenkins/workspace/Finance-me/target/surefire-reports', reportFiles: 'index.html'])
24         echo "Publishing HTML reports"
25     }
26 }
27 stage('Docker Image Creation'){
28     steps {
29         sh 'docker build -t srija1991/financeme .'
30     }
31 }
32
33 stage('Push Image to DockerHub'){
34     steps {
35         withCredentials([usernamePassword(credentialsId: 'logindocker', passwordVariable: 'docker_pswd', usernameVariable: 'docker_usr')])
36     }
}

```

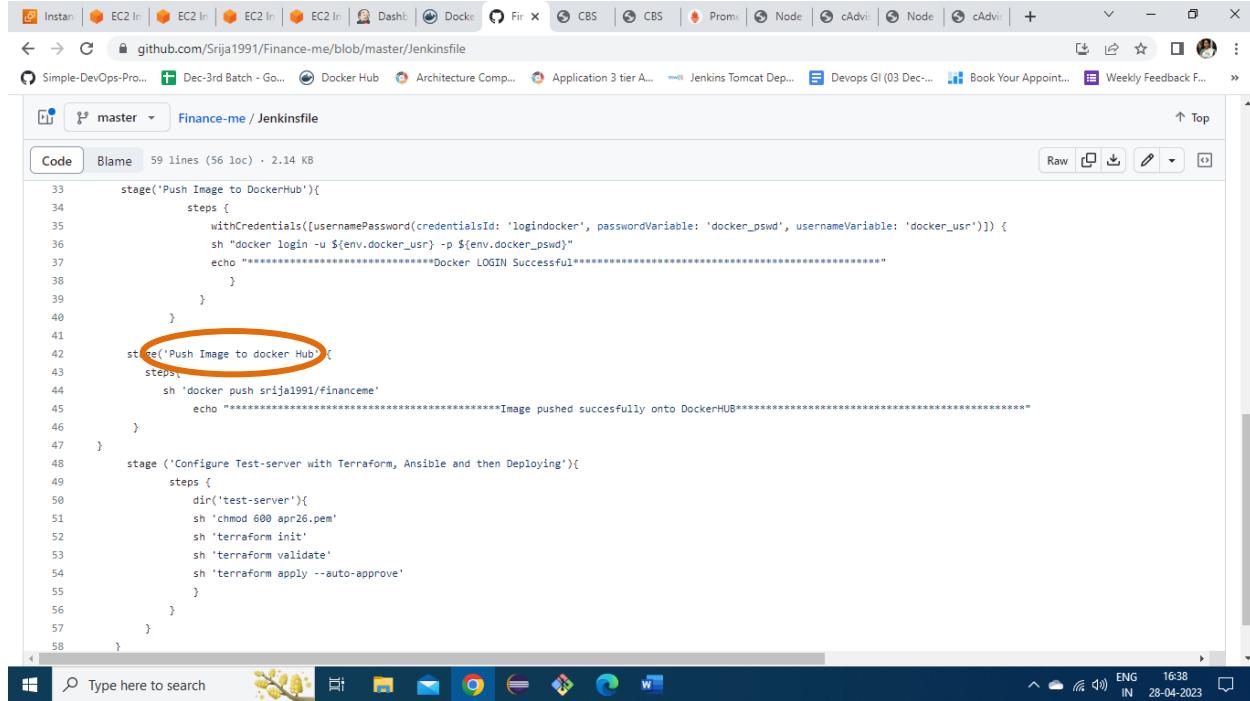
```

1 FROM openjdk:17
2 ARG JAR_FILE=target/*.jar
3 COPY ${JAR_FILE} app.jar
4 ENTRYPOINT ["java","-jar","/app.jar"]

```



6. Terraform is configured to create a test-server and deploy application using Ansible playbook.

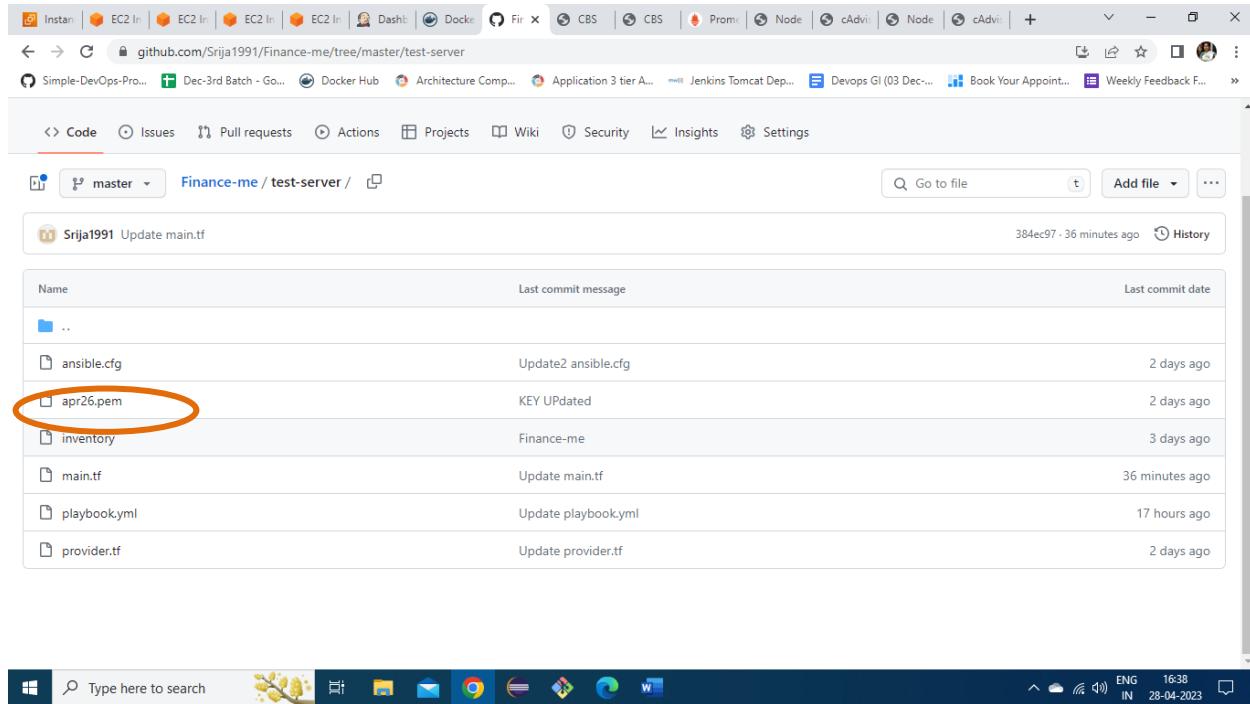


```

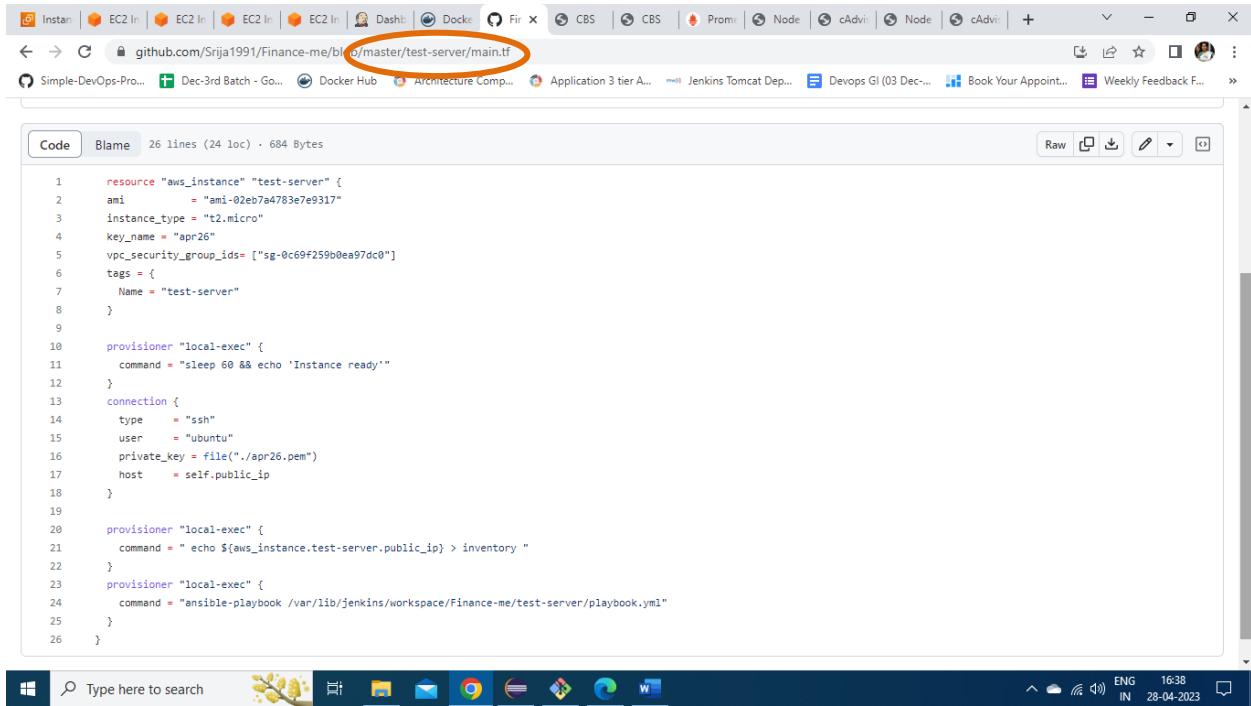
33     stage('Push Image to DockerHub'){
34         steps {
35             withCredentials([usernamePassword(credentialsId: 'logindocker', passwordVariable: 'docker_pswd', usernameVariable: 'docker_usr')])
36             sh "docker login -u ${env.docker_usr} -p ${env.docker_pswd}"
37             echo "*****Docker LOGIN Successful*****"
38         }
39     }
40 }
41
42 stage('Push Image to docker Hub'){
43     steps{
44         sh 'docker push srija1991/financeme'
45         echo "*****Image pushed sucessfully onto DockerHUB*****"
46     }
47 }
48 stage ('Configure Test-server with Terraform, Ansible and then Deploying'){
49     steps {
50         dir("test-server"){
51             sh "chmod 600 apr26.pem"
52             sh 'terraform init'
53             sh 'terraform validate'
54             sh 'terraform apply --auto-approve'
55         }
56     }
57 }
58

```

Here is the test-server folder which has *main.tf* file to create an EC2 instance in the region mentioned and attach the keypair “*apr26*”.



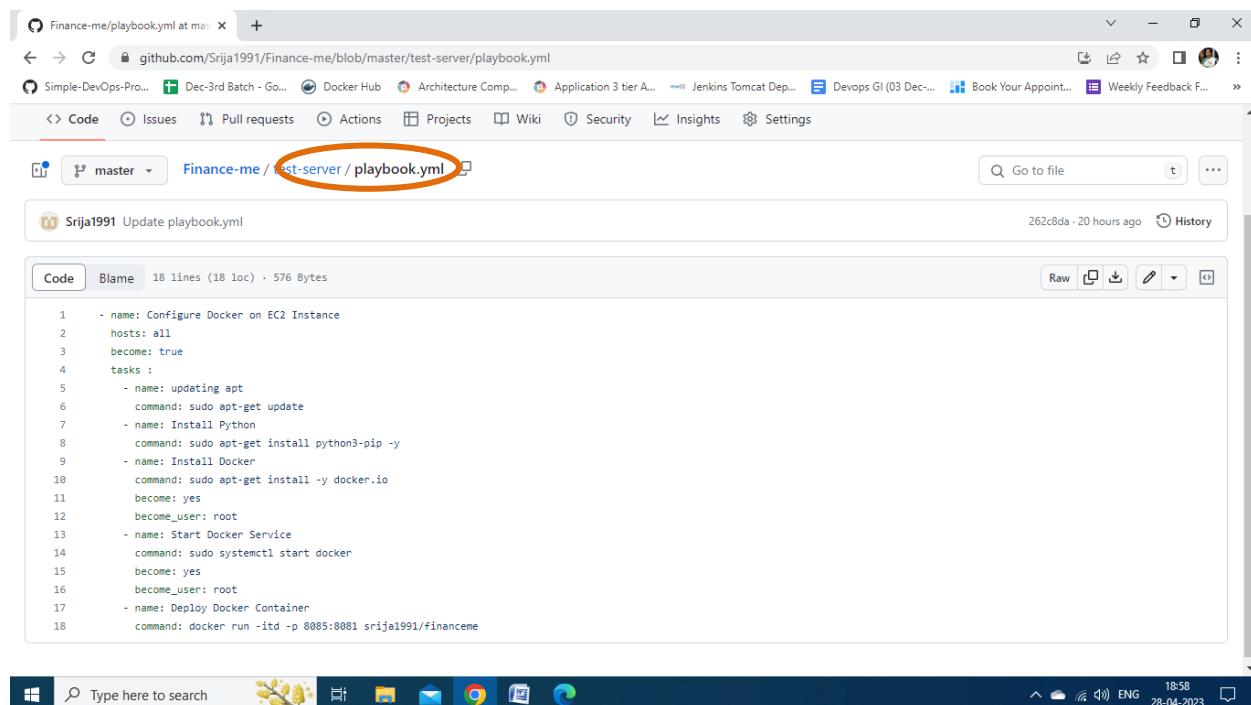
Name	Last commit message	Last commit date
..		
ansible.cfg	Update2 ansible.cfg	2 days ago
apr26.pem	KEY UPdated	2 days ago
inventory	Finance-me	3 days ago
main.tf	Update main.tf	36 minutes ago
playbook.yml	Update playbook.yml	17 hours ago
provider.tf	Update provider.tf	2 days ago



The screenshot shows a browser window with the URL github.com/Srija1991/Finance-me/blob/master/test-server/main.tf. The page displays a code editor with the following Terraform code:

```
1 resource "aws_instance" "test-server" {
2   ami           = "ami-02eb7a4783e7e9317"
3   instance_type = "t2.micro"
4   key_name      = "apr26"
5   vpc_security_group_ids = ["sg-0c69f259b0ea97dc0"]
6   tags = [
7     { Name = "test-server" }
8   ]
9
10  provisioner "local-exec" {
11    command = "sleep 60 && echo 'Instance ready.'"
12  }
13  connection {
14    type     = "ssh"
15    user     = "ubuntu"
16    private_key = file("./apr26.pem")
17    host     = self.public_ip
18  }
19
20  provisioner "local-exec" {
21    command = "echo ${aws_instance.test-server.public_ip} > inventory"
22  }
23  provisioner "local-exec" {
24    command = "ansible-playbook /var/lib/jenkins/workspace/Finance-me/test-server/playbook.yml"
25  }
26 }
```

Here is the playbook to configure the *test-server* with python and docker.



The screenshot shows a browser window with the URL github.com/Srija1991/Finance-me/blob/master/test-server/playbook.yml. The page displays a code editor with the following Ansible playbook code:

```
1 - name: Configure Docker on EC2 Instance
2   hosts: all
3   become: true
4   tasks :
5     - name: updating apt
6       command: sudo apt-get update
7     - name: Install Python
8       command: sudo apt-get install python3-pip -
9     - name: Install Docker
10    command: sudo apt-get install -y docker.io
11    become: yes
12    become_user: root
13    - name: Start Docker Service
14      command: sudo systemctl start docker
15      become: yes
16      become_user: root
17    - name: Deploy Docker Container
18      command: docker run -itd -p 8085:8081 srija1991/financeme
```

A job(Finance-me) is created which then triggers selenium test(Selenium Test) job and then deploy the application on to production server using (Prod-server) job.

Here is the screenshot of Finance-me job executed successfully at build number 10.

The screenshot shows the Jenkins interface for the 'Finance-me' job at build #10. The main window title is 'Console Output'. The left sidebar contains links like Status, Changes, Console Output (which is selected), View as plain text, Edit Build Information, Delete build '#10', Git Build Data, Restart from Stage, Replay, Pipeline Steps, and Workspaces. The right pane displays the Jenkinsfile's git URL: <https://github.com/Srija1991/Finance-me.git>. The log output shows the pipeline starting, running on Jenkins, and fetching code from the specified GitHub repository. A red oval highlights the URL in the log.

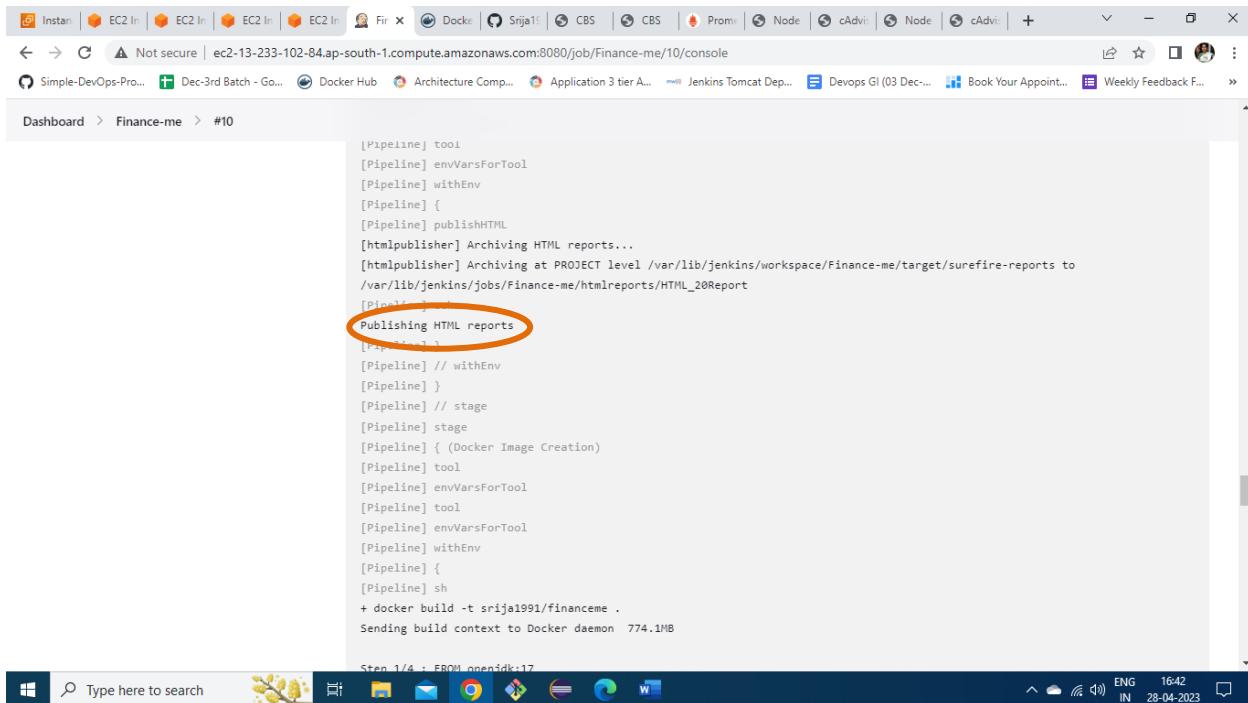
```
Started by user Srija Bichala
Obtained Jenkinsfile from git https://github.com/Srija1991/Finance-me.git
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in /var/lib/jenkins/workspace/Finance-me
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Declarative: Checkout SCM)
[Pipeline] checkout
The recommended git tool is: git
No credentials specified
> git rev-parse --resolve-git-dir /var/lib/jenkins/workspace/Finance-me/.git # timeout=10
Fetching changes from the remote Git repository
> git config remote.origin.url https://github.com/Srija1991/Finance-me.git # timeout=10
Fetching upstream changes from https://github.com/Srija1991/Finance-me.git
> git --version # timeout=10
> git --version # 'git version 2.34.1'
> git fetch --tags --force --progress -- https://github.com/Srija1991/Finance-me.git +refs/heads/*:refs/remotes/origin/* #
```

Created jar file displaying “build success”.

The screenshot shows the Jenkins interface for the 'Finance-me' job at build #10. The main window title is 'Console Output'. The left sidebar contains links like Status, Changes, Console Output (selected), View as plain text, Edit Build Information, Delete build '#10', Git Build Data, Restart from Stage, Replay, Pipeline Steps, and Workspaces. The right pane displays the Maven build logs. The log output shows the cleaning, compiling, testing, and packaging process. A red oval highlights the 'Scanning for projects...' message in the log.

```
Cleaning... Compiling.....Testing.....Packaging
[Pipeline] sh
+ mvn clean package
[1m[1;34mINFO[m] Scanning for projects...
[1m[1;34mINFO[m]
[1m[1;34mINFO[m] [1m-----< [0;36mcom.project.staragile:banking@[0;1m >-----[m
[1m[1;34mINFO[m] [1mBuilding banking 0.0.1-SNAPSHOT@[m
[1m[1;34mINFO[m] [1m-----[ jar ]-----[m
[1m[1;34mINFO[m]
[1m[1;34mINFO[m] [1m--- [0;32mmaven-clean-plugin:3.2.0:clean@[1m(default-clean)@[m @ [36mbanking@[0;1m ---@[m
[1m[1;34mINFO[m] Deleting /var/lib/jenkins/workspace/Finance-me/target
[1m[1;34mINFO[m]
[1m[1;34mINFO[m] [1m--- [0;32mmaven-resources-plugin:3.2.0:resources@[m @ [1m(default-resources)@[m @ [36mbanking@[0;1m --
[1m[1;34mINFO[m] Using 'UTF-8' encoding to copy filtered resources.
[1m[1;34mINFO[m] Using 'UTF-8' encoding to copy filtered properties files.
[1m[1;34mINFO[m] Copying 1 resource
[1m[1;34mINFO[m] Copying 81 resources
[1m[1;34mINFO[m]
[1m[1;34mINFO[m] [1m--- [0;32mmaven-compiler-plugin:3.10.1:compile@[m @ [1m(default-compile)@[m @ [36mbanking@[0;1m ---@[m
[1m[1;34mINFO[m] Changes detected - recompiling the module!
[1m[1;34mINFO[m] Compiling 5 source files to /var/lib/jenkins/workspace/Finance-me/target/classes
[1m[1;34mINFO[m]
[1m[1;34mINFO[m] [1m--- [0;32mmaven-resources-plugin:3.2.0:testResources@[m @ [1m(default-testResources)@[m @ [36mbanking@[0;1m ---@[m
[1m[1;34mINFO[m] Using 'UTF-8' encoding to copy filtered resources.
```

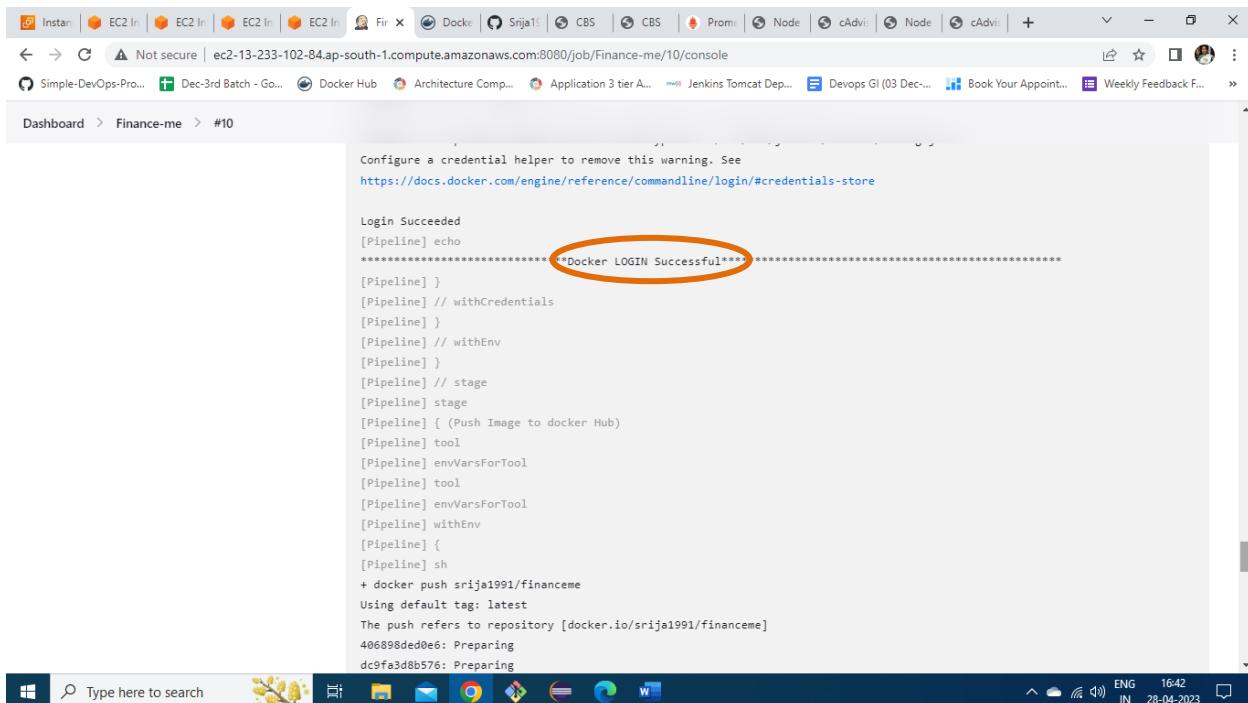
Generated HTML reports.



The screenshot shows a Jenkins pipeline log for job #10. The log output includes several pipeline steps and a section where it is publishing HTML reports. The line "[Pipeline] Publishing HTML reports" is highlighted with a red oval.

```
[Pipeline] tool
[Pipeline] envVarsForTool
[Pipeline] withEnv
[Pipeline] {
[Pipeline] publishHTML
[htmlpublisher] Archiving HTML reports...
[htmlpublisher] Archiving at PROJECT level /var/lib/jenkins/workspace/Finance-me/target/surefire-reports to
/var/lib/jenkins/jobs/Finance-me/htmlreports/HTML_20Report
[Pipeline] }
[Pipeline] Publishing HTML reports
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Docker Image Creation)
[Pipeline] tool
[Pipeline] envVarsForTool
[Pipeline] tool
[Pipeline] envVarsForTool
[Pipeline] withEnv
[Pipeline] {
[Pipeline] sh
+ docker build -t srija1991/financeme .
Sending build context to Docker daemon 774.1MB
```

Created a docker image and pushed onto Docker hub successfully.



The screenshot shows a Jenkins pipeline log for job #10. It includes a warning about configuring a credential helper and a link to the Docker documentation. The log then shows a successful Docker login step, which is highlighted with a red oval.

```
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
[Pipeline] echo
***** *Docker LOGIN Successful*** ****
[Pipeline] }
```

```

406898ded0e6: Pushed
latest: digest: sha256:0cba2976e58f6be20dc43a9336096120e8b3b52d5bd2a9f017e3b27952515a09 size: 1166
[Pipeline] echo
*****Image pushed successfully onto
DockerHUB*****
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Configure Test-server with Terraform, Ansible and then Deploying)
[Pipeline] tool
[Pipeline] envVarsForTool
[Pipeline] tool
[Pipeline] envVarsForTool
[Pipeline] withEnv
[Pipeline] {
[Pipeline] dir
Running in /var/lib/jenkins/workspace/Finance-me/test-server
[Pipeline] {
[Pipeline] sh
+ chmod 600 apr26.pem
[Pipeline] sh
+ terraform init

```

Image is pushed onto dockerhub with name *srija1991/financeme* which is used for further deployments.

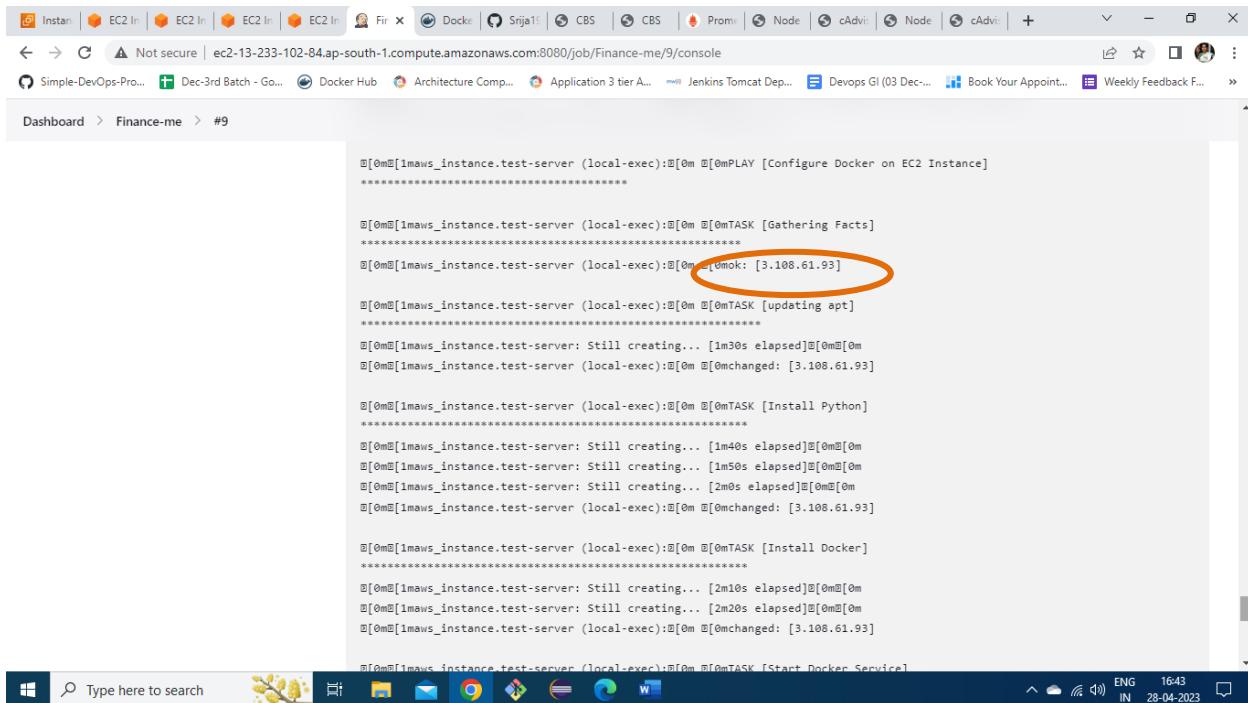
srija1991 / financeme
Contains: Image | Last pushed: 39 minutes ago

srija1991 / insureme
Contains: Image | Last pushed: 6 days ago

srija1991 / springboot-demo
Contains: Image | Last pushed: 2 months ago

srija1991 / springapp
Contains: Image | Last pushed: 2 months ago

The Ansible-playbook is executed successfully and application is deployed onto test-server.



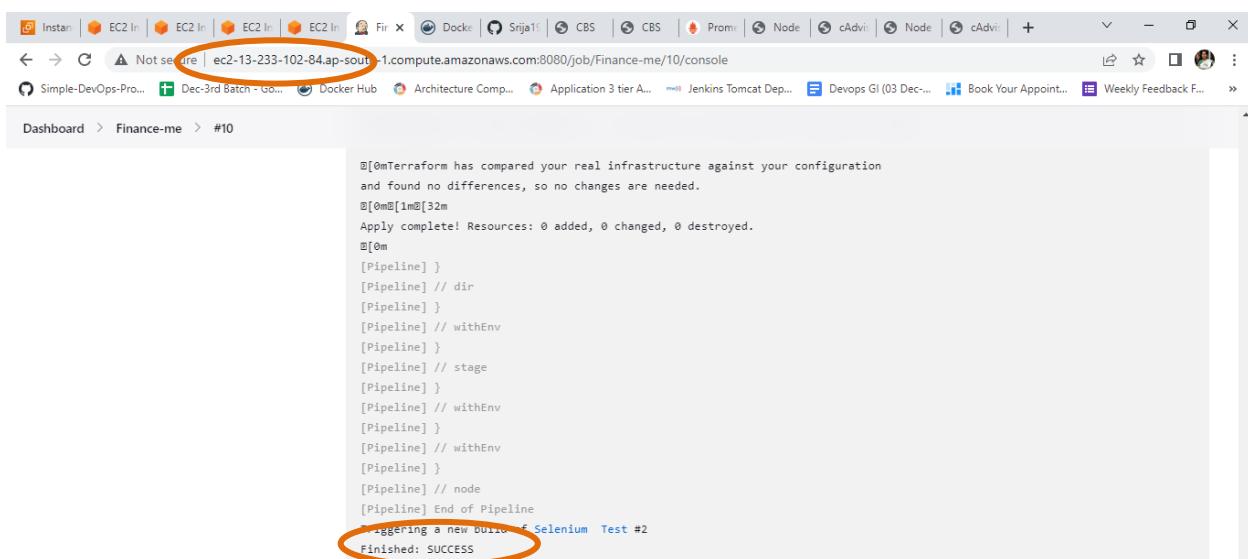
```
[0m@1maws_instance.test-server (local-exec):[0m @[0mPLAY [Configure Docker on EC2 Instance]
*****
[0m@1maws_instance.test-server (local-exec):[0m @[0mTASK [Gathering Facts]
*****
[0m@1maws_instance.test-server (local-exec):[0m @[0mok: [3.108.61.93]

[0m@1maws_instance.test-server (local-exec):[0m @[0mTASK [updating apt]
*****
[0m@1maws_instance.test-server: Still creating... [1m30s elapsed][0m@0m
[0m@1maws_instance.test-server (local-exec):[0m @[0mchanged: [3.108.61.93]

[0m@1maws_instance.test-server (local-exec):[0m @[0mTASK [Install Python]
*****
[0m@1maws_instance.test-server: Still creating... [1m40s elapsed][0m@0m
[0m@1maws_instance.test-server: Still creating... [1m50s elapsed][0m@0m
[0m@1maws_instance.test-server: Still creating... [2m0s elapsed][0m@0m
[0m@1maws_instance.test-server (local-exec):[0m @[0mchanged: [3.108.61.93]

[0m@1maws_instance.test-server (local-exec):[0m @[0mTASK [Install Docker]
*****
[0m@1maws_instance.test-server: Still creating... [2m10s elapsed][0m@0m
[0m@1maws_instance.test-server: Still creating... [2m20s elapsed][0m@0m
[0m@1maws_instance.test-server (local-exec):[0m @[0mchanged: [3.108.61.93]

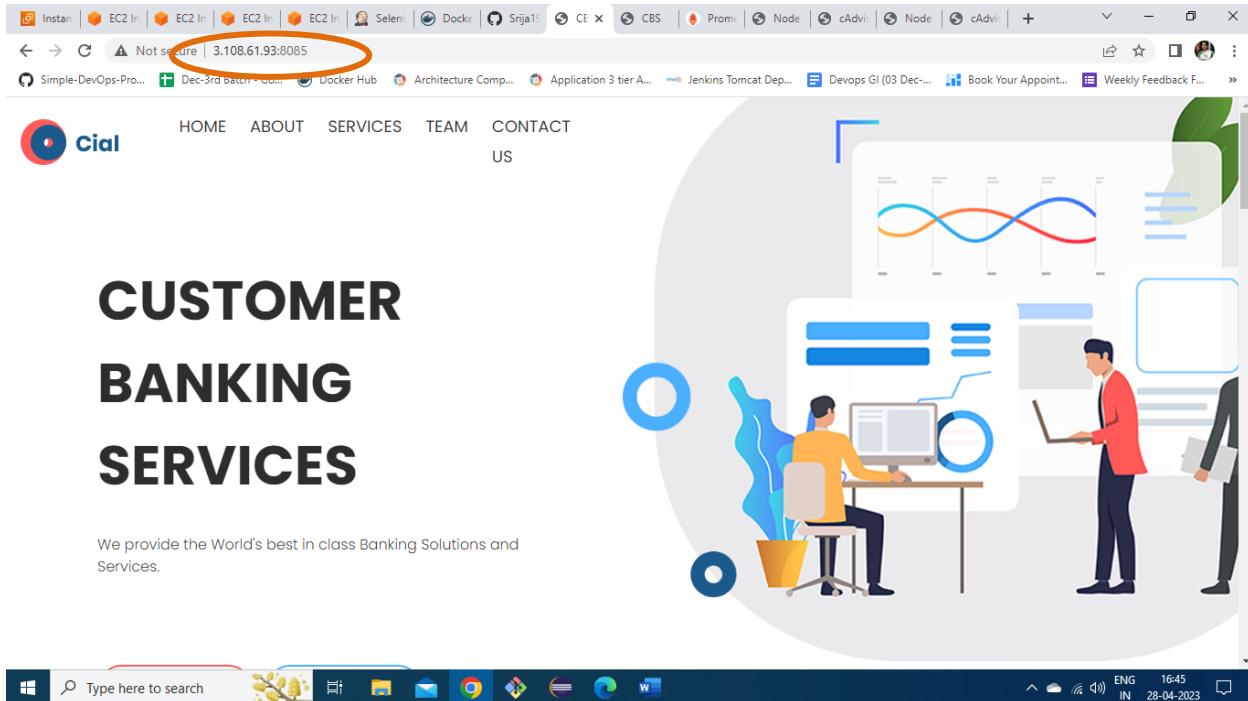
[0m@1maws_instance.test-server (local-exec):[0m @[0mTASK [Start Docker Service]
```



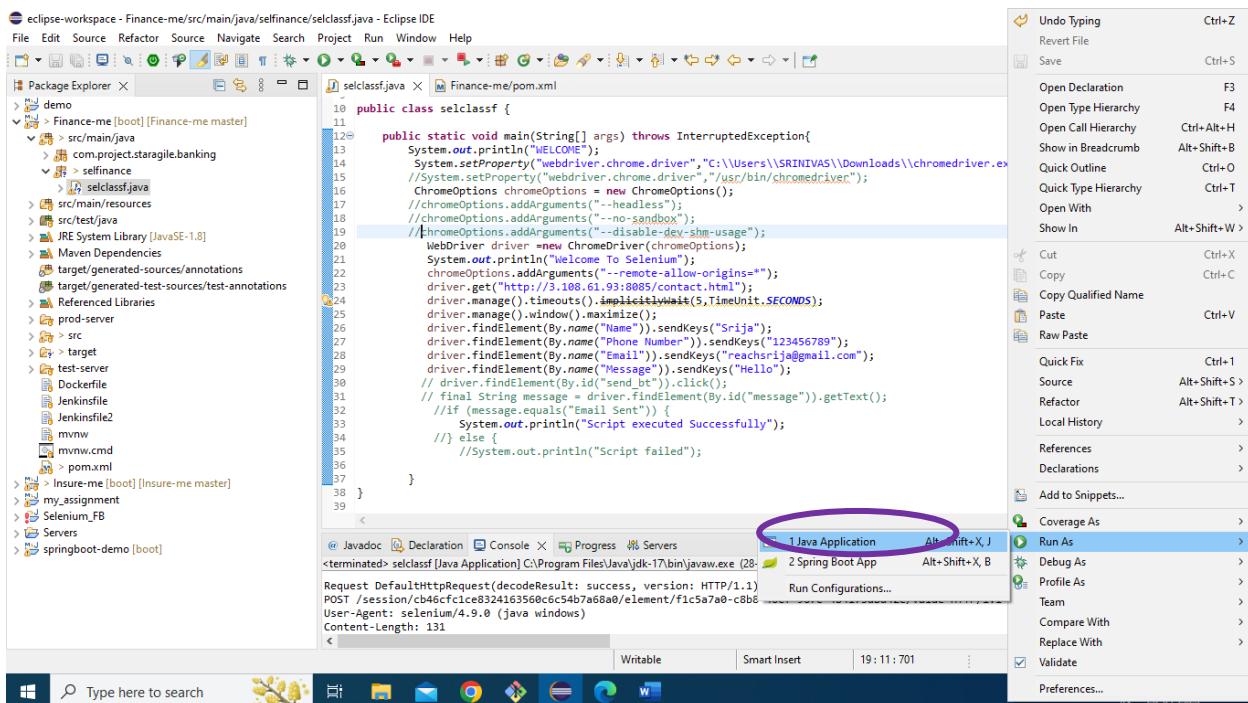
```
[0mTerraform has compared your real infrastructure against your configuration
and found no differences, so no changes are needed.
[0m@1maws[32m
Apply complete! Resources: 0 added, 0 changed, 0 destroyed.
[0m
[Pipeline]
[Pipeline] // dir
[Pipeline]
[Pipeline] // withEnv
[Pipeline]
[Pipeline] // stage
[Pipeline]
[Pipeline] // withEnv
[Pipeline]
[Pipeline] // withEnv
[Pipeline]
[Pipeline] // node
[Pipeline] End of Pipeline
    -> Triggering a new build < Selenium Test #2
Finished: SUCCESS
```



Test-server IP:3.108.61.93, and application is deployed on 8085 port.



Then the selenium test code is written in eclipse and cross checked the automation test on windows platform.



It is run as Java application to check the automation.

The screenshot shows a browser window with a contact form on the left and a map on the right. The contact form fields are circled in purple: the recipient's name 'Srija', their phone number '123456789', and their email address 'reachsrija@gmail.com'. The 'SEND' button is red. To the right is a map of the Eiffel Tower area in Paris, France, with various landmarks labeled. The browser status bar at the bottom indicates it's running on port 8085.

The jar file is exported as runnable jar and pushed onto github with name *selejarfinance.jar*.

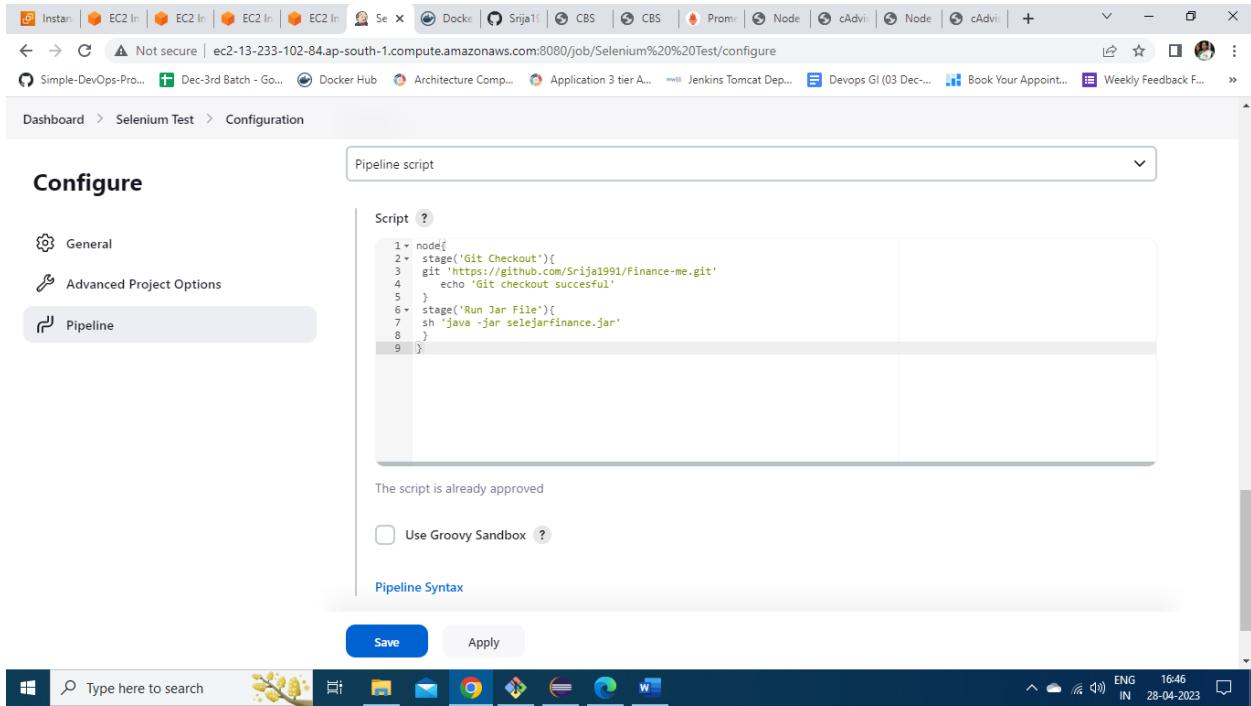
The screenshot shows a GitHub repository page for 'Finance-me'. A commit by 'Srija1991' titled 'Update Jenkinsfile2' is highlighted with a purple circle. The commit details show it was pushed 28 minutes ago with 134 commits. The repository has 0 stars, 1 watching, and 0 forks. It also lists releases, packages, and contributors. The GitHub interface includes a search bar and a taskbar at the bottom.

The Selenium Test is configured and triggered after the Finance-me job is successful and is shown here.

This screenshot shows the Jenkins General configuration page for the 'Selenium Test' job. The 'Enabled' switch is turned on. The 'Description' field contains the text 'Selenium test for finance -me application'. Under the 'Advanced Project Options' section, there are three checkboxes: 'Discard old builds', 'Do not allow concurrent builds', and 'Do not allow the pipeline to resume if the controller restarts'. At the bottom are 'Save' and 'Apply' buttons. The browser's address bar shows the URL as 'ec2-13-233-102-84.ap-south-1.compute.amazonaws.com:8080/job/Selenium%20%20Test/configure'. The operating system taskbar at the bottom indicates it's 16:46 on 28-04-2023.

This screenshot shows the Jenkins Configure page for the 'Selenium Test' job. The 'Build after other projects are built' checkbox is checked, and the 'Finance-me' project is listed in the 'Projects to watch' field. The 'Trigger only if build is stable' radio button is selected. There are also options for 'Trigger even if the build is unstable', 'Trigger even if the build fails', 'Always trigger, even if the build is aborted', 'Build periodically', 'Build whenever a SNAPSHOT dependency is built', 'GitHub hook trigger for GITScm polling', 'Poll SCM', and 'Quiet period'. At the bottom are 'Save' and 'Apply' buttons. The browser's address bar shows the URL as 'ec2-13-233-102-84.ap-south-1.compute.amazonaws.com:8080/job/Selenium%20%20Test/configure'. The operating system taskbar at the bottom indicates it's 16:46 on 28-04-2023.

The pipeline script for selenium test case is shown here,

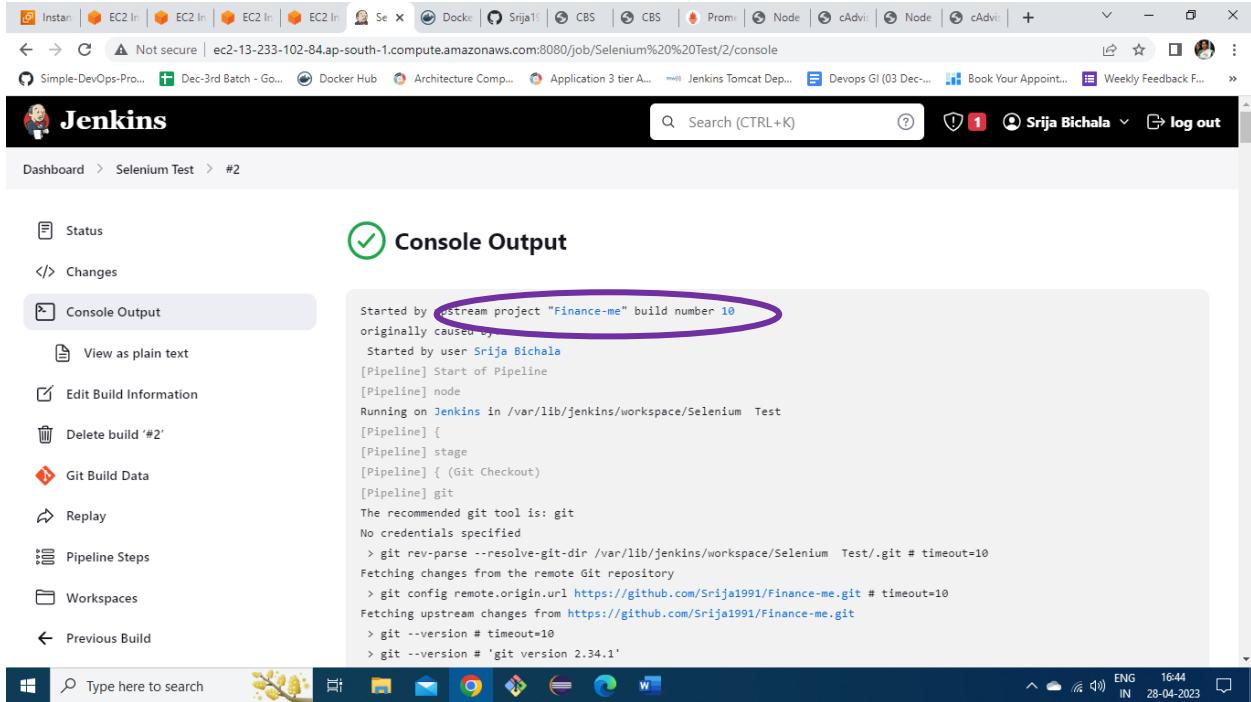


The screenshot shows the Jenkins Pipeline configuration page for a job named "Selenium Test". The "Pipeline" tab is selected. The pipeline script is defined as follows:

```
node{
    stage('Git Checkout'){
        git 'https://github.com/Srija1991/Finance-me.git'
        echo 'Git checkout successful'
    }
    stage('Run Jar File'){
        sh 'java -jar seleniumfinance.jar'
    }
}
```

The script is already approved. There is an option to "Use Groovy Sandbox". At the bottom, there are "Save" and "Apply" buttons.

This is console output for selenium test which is triggered from Finance-me build 10. The build is success displaying the *google-chrome is started successfully* and *script is executed successfully*.



The screenshot shows the Jenkins Console Output for build #2. The output starts with:

```
Started by upstream project "Finance-me" build number 10
originally caused by:
Started by user Srija Bichala
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in /var/lib/jenkins/workspace/Selenium Test
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Git Checkout)
[Pipeline] git
The recommended git tool is: git
No credentials specified
> git rev-parse --resolve-git-dir /var/lib/jenkins/workspace/Selenium Test/.git # timeout=10
Fetching changes from the remote Git repository
> git config remote.origin.url https://github.com/Srija1991/Finance-me.git # timeout=10
Fetching upstream changes from https://github.com/Srija1991/Finance-me.git
> git --version # timeout=10
> git --version # 'git version 2.34.1'
```

```
10:43:28.960 [main] DEBUG io.netty.buffer.PooledByteBufAllocator - -Dio.netty.allocator.cacheTrimInterval: 8192
10:43:28.960 [main] DEBUG io.netty.buffer.PooledByteBufAllocator - -Dio.netty.allocator.cacheTrimIntervalMillis: 0
10:43:28.960 [main] DEBUG io.netty.buffer.PooledByteBufAllocator - -Dio.netty.allocator.useCacheForAllThreads: false
10:43:28.960 [main] DEBUG io.netty.buffer.PooledByteBufAllocator - -Dio.netty.allocator.maxCachedByteBuffersPerChunk: 1023
10:43:28.969 [main] DEBUG io.netty.buffer.ByteBufUtil - -Dio.netty.allocator.type: pooled
10:43:28.969 [main] DEBUG io.netty.buffer.ByteBufUtil - -Dio.netty.threadLocalDirectBufferSize: 0
10:43:28.970 [main] DEBUG io.netty.buffer.ByteBufUtil - -Dio.netty.maxThreadLocalCharBufferSize: 16384
Starting ChromeDriver 112.0.5615.49 (bd2a7bcbb881c11e8cf3e3916914-refs/branch-heads/5615@#936) on port 17042
Only local connections are allowed.
Please see https://chromedriver.chromium.org/security-considerations for suggestions on keeping ChromeDriver safe.
ChromeDriver was started successfully.
10:43:29.256 [Forwarding newSession on session null to remote] DEBUG io.netty.channel.DefaultChannelId - -
Dio.netty.processId: 17353 (auto-detected)
10:43:29.257 [Forwarding newSession on session null to remote] DEBUG io.netty.util.NetUtil - -Djava.net.preferIPv4Stack: false
10:43:29.258 [Forwarding newSession on session null to remote] DEBUG io.netty.util.NetUtil - -Djava.net.preferIPv6Addresses: false
10:43:29.260 [Forwarding newSession on session null to remote] DEBUG io.netty.util.NetUtilInitializations - Loopback interface: lo (lo, 0:0:0:0:0:0:1%lo)
10:43:29.261 [Forwarding newSession on session null to remote] DEBUG io.netty.util.NetUtil - /proc/sys/net/core/somaxconn: 4096
10:43:29.262 [Forwarding newSession on session null to remote] DEBUG io.netty.channel.DefaultChannelId - -
Dio.netty.machineId: 02:42:49:ff:fe:5d:0:2b (auto-detected)
10:43:29.314 [AsyncHttpClient-1-2] DEBUG io.netty.buffer.AbstractByteBuf - -Dio.netty.buffer.checkAccessible: true
10:43:29.314 [AsyncHttpClient-1-2] DEBUG io.netty.buffer.AbstractByteBuf - -Dio.netty.buffer.checkBounds: true
10:43:29.314 [AsyncHttpClient-1-2] DEBUG io.netty.util.ResourceLeakDetectorFactory - Loaded default ResourceLeakDetector:
```

```
Content-Type: application/json; charset=utf-8
host: localhost:17042
accept: /*

Response DefaultHttpResponse(decodeResult: success, version: HTTP/1.1)
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
cache-control: no-cache
content-length: 14

10:43:33.215 [AsyncHttpClient-1-2] DEBUG org.asynchttpclient.netty.channel.ChannelManager - Adding key:
http://127.0.0.1:17042/ // stage
[Pipeline] // stage
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Triggering a new build of prod_server #
Finished: SUCCESS

Script executed Successfully
```



Once the test is successful, the application is deployed onto production server by triggering the job *prod_server*. The configuration for *prod_server* is shown here.

The screenshot shows two consecutive screenshots of the Jenkins configuration interface for the 'prod_server' job.

Top Screenshot (General Tab):

- General Tab:** Enabled (checkbox checked).
- Description:** Production server.
- Options:**
 - Discard old builds
 - Do not allow concurrent builds
 - Do not allow the pipeline to resume if the controller restarts
- Buttons:** Save, Apply.

Bottom Screenshot (Build Triggers Tab):

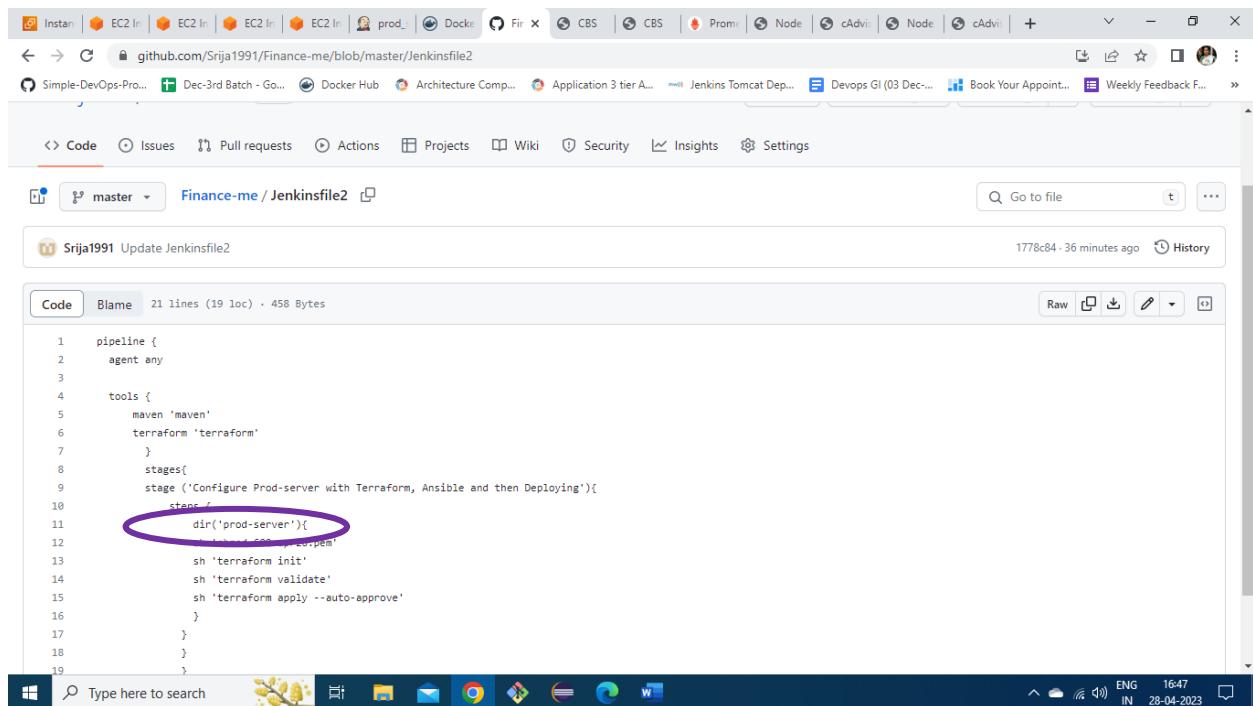
- Build Triggers Tab:** Build after other projects are built (checkbox checked).
- Projects to watch:** Selenium Test.
- Trigger Options:**
 - Trigger only if build is stable
 - Trigger even if the build is unstable
 - Trigger even if the build fails
 - Always trigger, even if the build is aborted
- Other Options:**
 - Build periodically
 - Build whenever a SNAPSHOT dependency is built
 - GitHub hook trigger for GITScm polling
- Buttons:** Save, Apply.

The screenshot shows the Jenkins Pipeline configuration page for a job named 'prod_server'. The 'Pipeline' tab is selected. Under the 'SCM' section, 'Git' is chosen as the provider. The 'Repository URL' is set to `https://github.com/Srija1991/Finance-me.git`. The 'Credentials' dropdown is set to 'none'. At the bottom, there are 'Save' and 'Apply' buttons.

This job is triggered after successful build of Selenium Test job.

The screenshot shows the Jenkins Pipeline configuration page for a job named 'prod_server'. The 'Pipeline' tab is selected. In the 'Script Path' section, the value `Jenkinsfile2` is entered. This line is circled in purple. Below it, the 'Lightweight checkout' checkbox is checked. At the bottom, there are 'Save' and 'Apply' buttons.

Here is the script for deployment onto production server.



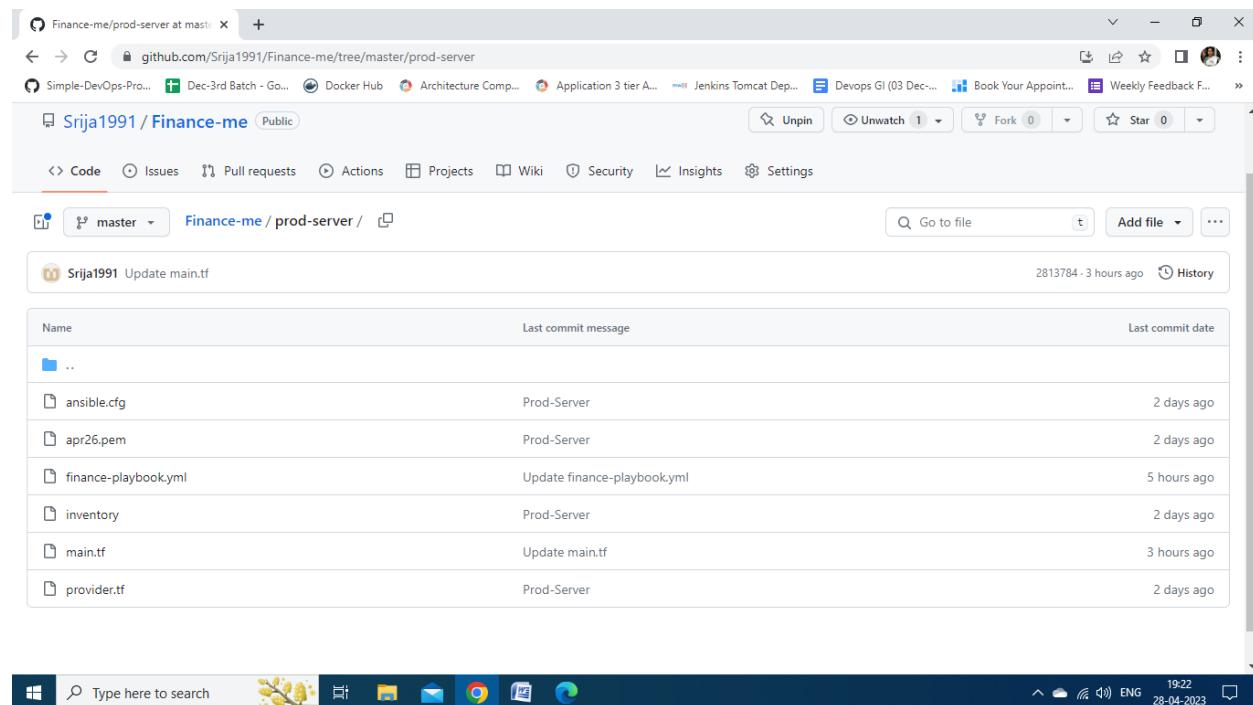
The screenshot shows a GitHub repository page for 'Finance-me'. The file 'Jenkinsfile2' is open. A specific section of the code is highlighted with a blue oval:

```

1 pipeline {
2     agent any
3
4     tools {
5         maven 'maven'
6         terraform 'terraform'
7     }
8     stages{
9         stage ('Configure Prod-server with Terraform, Ansible and then Deploying'){
10            steps {
11                dir('prod-server'){
12                    sh 'cd prod-server & cat ap26.pem'
13                    sh 'terraform init'
14                    sh 'terraform validate'
15                    sh 'terraform apply --auto-approve'
16                }
17            }
18        }
19    }

```

The prod-server folder on the repository is shown below. It has *main.tf* file used to create EC2 infrastructure, *finance-playbook.yml* which configures docker onto prod-server.



The screenshot shows a GitHub repository page for 'Finance-me'. The 'prod-server' directory is selected. The contents of the directory are listed in a table:

Name	Last commit message	Last commit date
...		
ansible.cfg	Prod-Server	2 days ago
ap26.pem	Prod-Server	2 days ago
finance-playbook.yml	Update finance-playbook.yml	5 hours ago
inventory	Prod-Server	2 days ago
main.tf	Update main.tf	3 hours ago
provider.tf	Prod-Server	2 days ago

Finance-me/main.tf at master · Srija1991/Finance-me · GitHub

Simple-DevOps-Pro... Dec-3rd Batch - Go... Docker Hub Architecture Comp... Application 3 tier A... Jenkins Tomcat Dep... Devops GI (03 Dec... Book Your Appoint... Weekly Feedback F... master Finance-me / prod-server / main.tf

Srija1991 Update main.tf

Code Blame 25 lines (24 loc) · 710 Bytes

```
resource "aws_instance" "prod-server" {
  ami           = "ami-02eb7a4783e7e9317"
  instance_type = "t2.micro"
  key_name      = "apr26"
  vpc_security_group_ids = ["sg-0c69f259b0ea97dc0"]
  tags = {
    Name = "prod-server"
  }
  provisioner "local-exec" {
    command = "sleep 60 & echo 'Instance ready'"
  }
  connection {
    type     = "ssh"
    user     = "ubuntu"
    private_key = file("./apr26.pem")
    host     = self.public_ip
  }
  provisioner "local-exec" {
    command = "echo ${aws_instance.prod-server.public_ip} > inventory"
  }
  provisioner "local-exec" {
    command = "ansible-playbook /var/lib/jenkins/workspace/Finance-me/prod-server/finance-playbook.yml"
  }
}
```

Type here to search 19:22 28-04-2023

Finance-me/finance-playbook.yml at master · Srija1991/Finance-me · GitHub

Simple-DevOps-Pro... Dec-3rd Batch - Go... Docker Hub Architecture Comp... Application 3 tier A... Jenkins Tomcat Dep... Devops GI (03 Dec... Book Your Appoint... Weekly Feedback F... master Finance-me / prod-server / finance-playbook.yml

Srija1991 Update finance-playbook.yml

Code Blame 18 lines (18 loc) · 576 Bytes

```
- name: Configure Docker on EC2 Instance
hosts: all
become: true
tasks :
- name: updating apt
  command: sudo apt-get update
- name: Install Python
  command: sudo apt-get install python3-pip -y
- name: Install Docker
  command: sudo apt-get install -y docker.io
  become: yes
  become_user: root
- name: Start Docker Service
  command: sudo systemctl start docker
  become: yes
  become_user: root
- name: Deploy Docker Container
  command: docker run -itd -p 8085:8081 srija1991/financeme
```

Type here to search 19:23 28-04-2023

Here is the console output for prod-server and application successfully deployed.

The screenshot shows the Jenkins interface for a build named 'prod_server' (Build #2). The 'Console Output' tab is selected. The output log is displayed, starting with the message: "Started by upstream project "Selenium Test" build number 2 originally caused by: Started by upstream project "Finance-me" build number 10 originally caused by: Started by user Srija Bichala". A purple oval highlights this initial deployment information. The log continues with the Jenkinsfile2 being obtained from GitHub, the pipeline starting, and the build running on Jenkins. It then details the git checkout process, fetching changes from the remote repository, and finally executing the Terraform init command. The output concludes with the message: "Terraform has created a lock file .terraform.lock.hcl to record the provider selections it made above. Include this file in your version control repository so that Terraform can guarantee to make the same selections by default when you run "terraform init" in the future." A note at the bottom of the log states: "If you ever set or change modules or backend configuration for Terraform," followed by a long ellipsis.

```
Started by upstream project "Selenium Test" build number 2
originally caused by:
Started by upstream project "Finance-me" build number 10
originally caused by:
Started by user Srija Bichala
Obtained Jenkinsfile2 from git https://github.com/Srija1991/Finance-me.git
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in /var/lib/jenkins/workspace/prod_server
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Declarative: Checkout SCM)
[Pipeline] checkout
The recommended git tool is: git
No credentials specified
> git rev-parse --resolve-git-dir /var/lib/jenkins/workspace/prod_server/.git # timeout=10
Fetching changes from the remote Git repository
> git config remote.origin.url https://github.com/Srija1991/Finance-me.git # timeout=10
[Pipeline] }
[Pipeline] sh
+ chmod 600 apr26.pem
[Pipeline] sh
+ terraform init
@0m@[im]initializing the backend...@0m
@0m@[im]initializing provider plugins...@0m
- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v4.65.0...
- Installed hashicorp/aws v4.65.0 (signed by HashiCorp)

Terraform has created a lock file .terraform.lock.hcl to record the provider
selections it made above. Include this file in your version control repository
so that Terraform can guarantee to make the same selections by default when
you run "terraform init" in the future.@0m

@0m@[1m@[32mTerraform has been successfully initialized!@0m@[32m@0m
@0m@[32m
You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.

If you ever set or change modules or backend configuration for Terraform,
```

This screenshot shows the continuation of the Jenkins build log for 'prod_server' (Build #2). The log begins with the Terraform init command being executed, followed by the creation of a lock file. It then proceeds through the Terraform apply command, showing the execution of various resources like AWS Lambda functions and API Gateway endpoints. The log concludes with the message: "Terraform has successfully applied all changes to the desired state: 12 resources in 12s." A note at the bottom of the log states: "If you ever set or change modules or backend configuration for Terraform," followed by a long ellipsis.

```
Running in /var/lib/jenkins/workspace/prod_server/prod-server
[Pipeline] {
[Pipeline] sh
+ chmod 600 apr26.pem
[Pipeline] sh
+ terraform init
@0m@[im]initializing the backend...@0m
@0m@[im]initializing provider plugins...@0m
- Finding latest version of hashicorp/aws...
- Installing hashicorp/aws v4.65.0...
- Installed hashicorp/aws v4.65.0 (signed by HashiCorp)

Terraform has created a lock file .terraform.lock.hcl to record the provider
selections it made above. Include this file in your version control repository
so that Terraform can guarantee to make the same selections by default when
you run "terraform init" in the future.@0m

@0m@[1m@[32mTerraform has been successfully initialized!@0m@[32m@0m
@0m@[32m
You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.

If you ever set or change modules or backend configuration for Terraform,
```

```
Insta EC2 In EC2 In EC2 In pr Docker Financ CBS CBS Prom Node cAdvi Node cAdvi + - X
Not secure | ec2-13-233-102-84.ap-south-1.compute.amazonaws.com:8080/job/prod_server/2/console
Simple-DevOps-Pro... Dec-3rd Batch - Go... Docker Hub Architecture Comp... Application 3 tier A... Jenkins Tomcat Dep... Devops GI (03 Dec-... Book Your Appoint... Weekly Feedback F...
Dashboard > prod_server > #2
-----
  ]}
}

@imPlan:@[0m 1 to add, 0 to change, 0 to destroy.
@[0m@[1aws_instance.prod-server: Creating...@[0m@[0m
@[0m@[1aws_instance.prod-server: Still creating... [10s elapsed]@[0m@[0m
@[0m@[1aws_instance.prod-server: Still creating... [20s elapsed]@[0m@[0m
@[0m@[1aws_instance.prod-server: Still creating... [30s elapsed]@[0m@[0m
@[0m@[1aws_instance.prod-server Provisioning with 'local-exec'...@[0m@[0m
@[0m@[1aws_instance.prod-server (local-exec):@[0m@[0mExecuting: ["/bin/sh" "-c" "sleep 60 && echo 'Instance ready'"]
@[0m@[1aws_instance.prod-server: Still creating... [40s elapsed]@[0m@[0m
@[0m@[1aws_instance.prod-server: Still creating... [50s elapsed]@[0m@[0m
@[0m@[1aws_instance.prod-server: Still creating... [1m0s elapsed]@[0m@[0m
@[0m@[1aws_instance.prod-server: Still creating... [1m10s elapsed]@[0m@[0m
@[0m@[1aws_instance.prod-server: Still creating... [1m20s elapsed]@[0m@[0m
@[0m@[1aws_instance.prod-server: Still creating... [1m30s elapsed]@[0m@[0m
@[0m@[1aws_instance.prod-server (local-exec):@[0m@[0mInstance ready
@[0m@[1aws_instance.prod-server: Provisioning with 'local-exec'...@[0m@[0m
@[0m@[1aws_instance.prod-server (local-exec):@[0m@[0mExecuting: ["/bin/sh" "-c" "echo 15.207.107.115 > inventory"]
@[0m@[1aws_instance.prod-server: Provisioning with 'local-exec'...@[0m@[0m
@[0m@[1aws_instance.prod-server (local-exec):@[0m@[0mExecuting: ["/bin/sh" "-c" "ansible-playbook
/var/lib/jenkins/workspace/Finance-me/prod-server/finance-playbook.yml"]

@[0m@[1aws_instance.prod-server (local-exec):@[0m@[0mPLAY [Configure Docker on EC2 Instance]
*****
```

```
Insta EC2 In EC2 In EC2 In pr Docker Financ CBS CBS Prom Node cAdvi Node cAdvi + - X
Not secure | ec2-13-233-102-84.ap-south-1.compute.amazonaws.com:8080/job/prod_server/2/console
Simple-DevOps-Pro... Dec-3rd Batch - Go... Docker Hub Architecture Comp... Application 3 tier A... Jenkins Tomcat Dep... Devops GI (03 Dec-... Book Your Appoint... Weekly Feedback F...
Dashboard > prod_server > #2
*****
@[0m@[1aws_instance.prod-server: Still creating... [1m40s elapsed]@[0m@[0m
@[0m@[1aws_instance.prod-server (local-exec):@[0m@[0mchanged: [15.207.107.115] [15.207.107.115]

@[0m@[1aws_instance.prod-server (local-exec):@[0m@[0mTASK [Install Python]
*****
@[0m@[1aws_instance.prod-server: Still creating... [1m50s elapsed]@[0m@[0m
@[0m@[1aws_instance.prod-server: Still creating... [2m0s elapsed]@[0m@[0m
@[0m@[1aws_instance.prod-server (local-exec):@[0m@[0mchanged: [15.207.107.115]

@[0m@[1aws_instance.prod-server (local-exec):@[0m@[0mTASK [Install Docker]
*****
@[0m@[1aws_instance.prod-server: Still creating... [2m10s elapsed]@[0m@[0m
@[0m@[1aws_instance.prod-server: Still creating... [2m20s elapsed]@[0m@[0m
@[0m@[1aws_instance.prod-server (local-exec):@[0m@[0mchanged: [15.207.107.115]

@[0m@[1aws_instance.prod-server (local-exec):@[0m@[0mTASK [Start Docker Service]
*****
@[0m@[1aws_instance.prod-server (local-exec):@[0m@[0mchanged: [15.207.107.115]

@[0m@[1aws_instance.prod-server (local-exec):@[0m@[0mTASK [Deploy Docker Container]
*****
@[0m@[1aws_instance.prod-server: Still creating... [2m30s elapsed]@[0m@[0m
@[0m@[1aws_instance.prod-server: Still creating... [2m40s elapsed]@[0m@[0m
@[0m@[1aws_instance.prod-server (local-exec):@[0m@[0mchanged: [15.207.107.115]

@[0m@[1aws_instance.prod-server (local-exec):@[0m@[0mPLAY [Deploy Docker Container]
*****
```

Dashboard > prod_server > #2

```

@[0m@[1aws_instance.prod-server (local-exec):@[0m @[0mchanged: [15.207.107.115]
@[0m@[1aws_instance.prod-server (local-exec):@[0m @[0mPLAY RECAP
*****
@[0m@[1aws_instance.prod-server (local-exec):@[0m @[0m15.207.107.115      : ok=6    changed=5    unreachable=0
failed=0    skipped=0   rescued=0   ignored=0

@[0m@[1aws_instance.prod-server: Creation complete after 2m41s [id=i-05344b912772d65d9]@[0m
@*[32m
Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
@*[0m
[Pipeline]
[Pipeline] // dir
[Pipeline]
[Pipeline] // withEnv
[Pipeline]
[Pipeline] // stage
[Pipeline]
[Pipeline] // withEnv
[Pipeline]
[Pipeline] // withEnv
[Pipeline]
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS

```

Type here to search ENG 16:48 IN 28-04-2023

Production server IP: 15.207.107.115:8085.

Not secure | 15.207.107.115:8085

Cial

HOME ABOUT SERVICES TEAM CONTACT US

CUSTOMER BANKING SERVICES

We provide the World's best in class Banking Solutions and Services.

15.207.107.115:8085/contact.html

Type here to search ENG 16:49 IN 28-04-2023

The three jobs are successfully build and application is first deployed on test-server after selenium test it is deployed onto production server.

The screenshot shows the Jenkins dashboard with the following details:

- Build History:** Shows three successful builds:
 - Finance-me:** Last success 1 hr 5 min ago, build #10, duration 1 min 11 sec.
 - prod_server:** Last success 1 hr 4 min ago, build #2, duration 2 min 59 sec.
 - Selenium Test:** Last success 1 hr 4 min ago, build #2, duration 9.7 sec.
- Build Queue:** No builds in the queue.
- Build Executor Status:** One executor is idle.

Now cAdvisor and Node-exporter are run on the test-server and prod-server to monitor using Prometheus and Grafana tools.

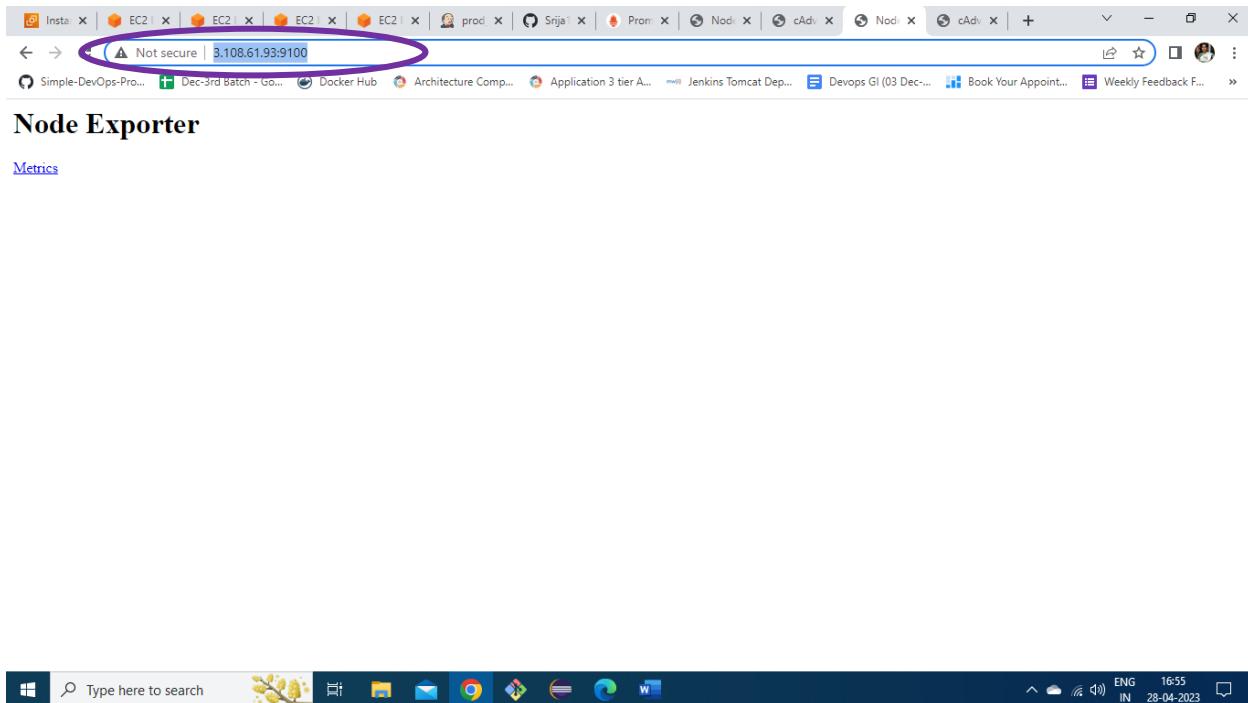
Test-server

The screenshot shows the AWS CloudShell interface with the following details:

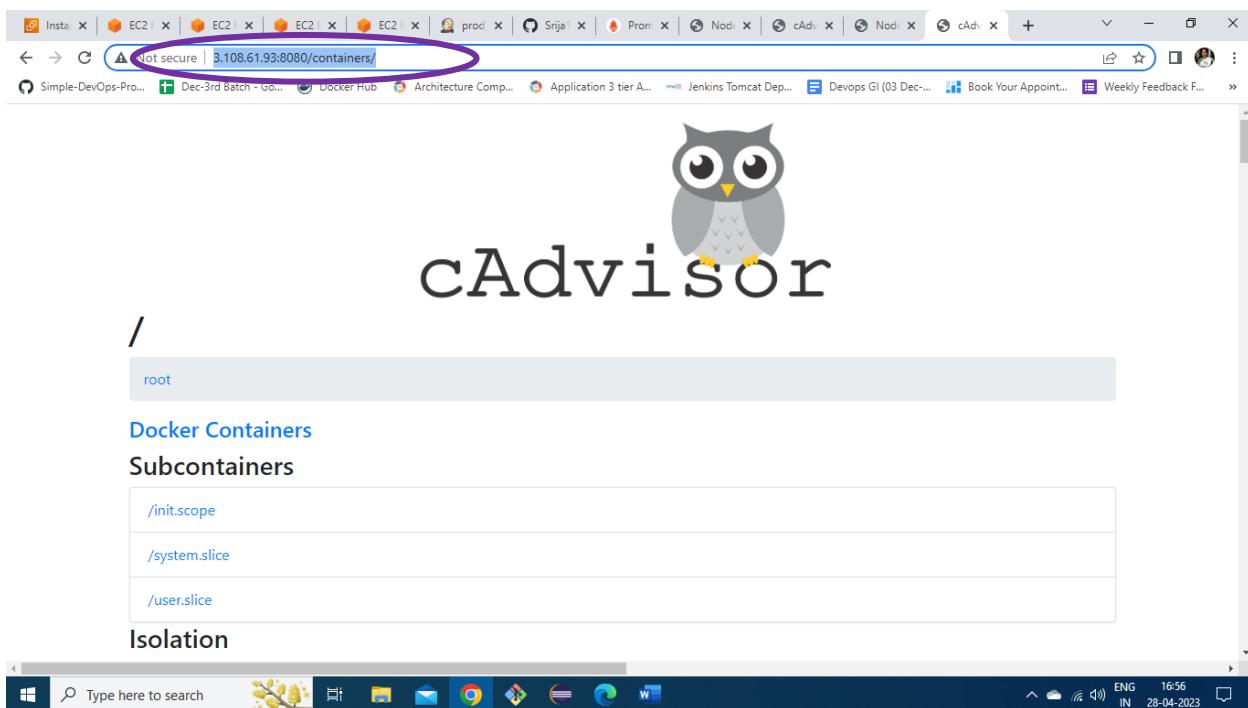
- Terminal Output:**

```
ubuntu@ip-172-31-38-135:~$ sudo docker ps -a
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              PORTS
 NAMES
e2d9fdfea5ce      prom/node-exporter:latest   "/bin/node_exporter ..."   25 minutes ago    Up 25 minutes
cool_jepsen
338c839d7f02      gcr.io/cadvisor/cadvisor:v0.39.3  "/usr/bin/cadvisor ..."   25 minutes ago    Up 25 minutes (healthy)  0.0.0.0:8080->8080/tcp, :::8080-
>8080/tcp  cadvisor
421ca5275540      srija1991/financeme     "java -jar /app.jar"       45 minutes ago   Up 45 minutes      0.0.0.0:8085->8081/tcp, :::8085-
>8081/tcp  nostalgic_ellis
ubuntu@ip-172-31-38-135:~$
```
- Message Bar:** i-0b1a1a1fbc67e7526 (test-server)
- Message Bar (circled):** Control IPs: 3.108.61.95 Private IPs: 172.31.38.155
- CloudShell Footer:** CloudShell Feedback Language © 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences

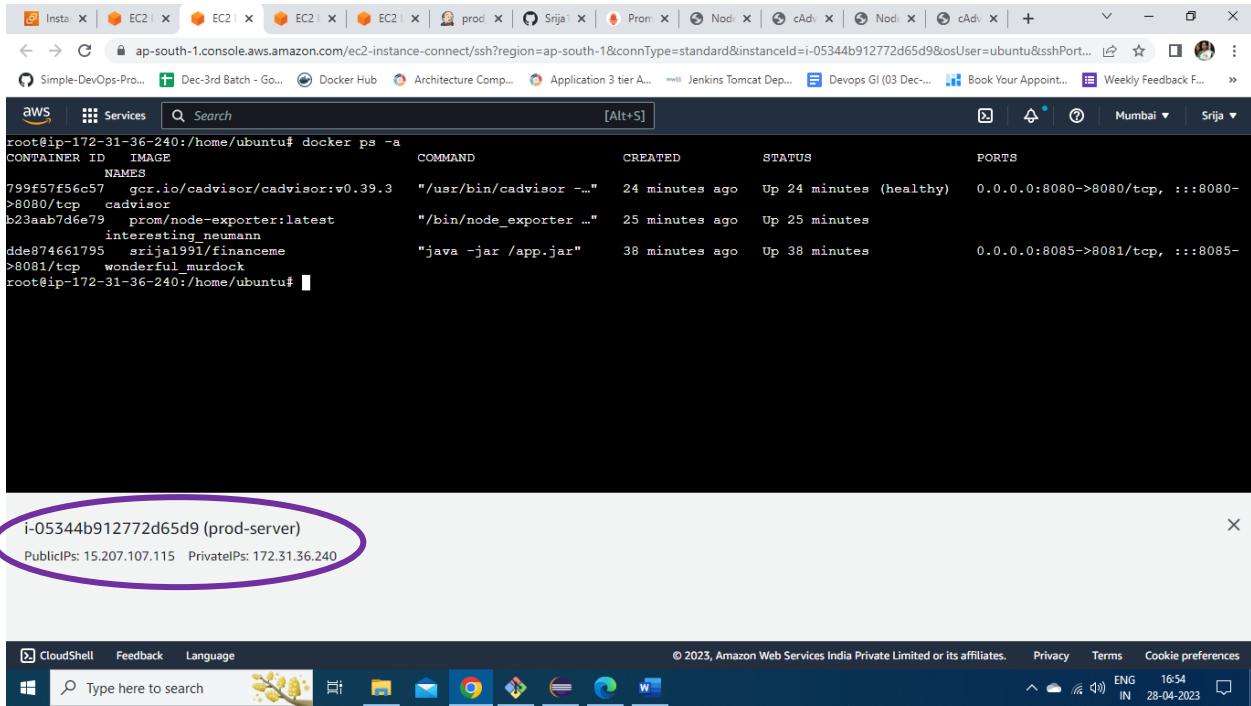
Node-exporter on test-server: port number : 3.108.61.93:9100



cAdvisor on test-server : port number: 3.108.61.93:8080



Cadvisor and Node-exporter on production server.

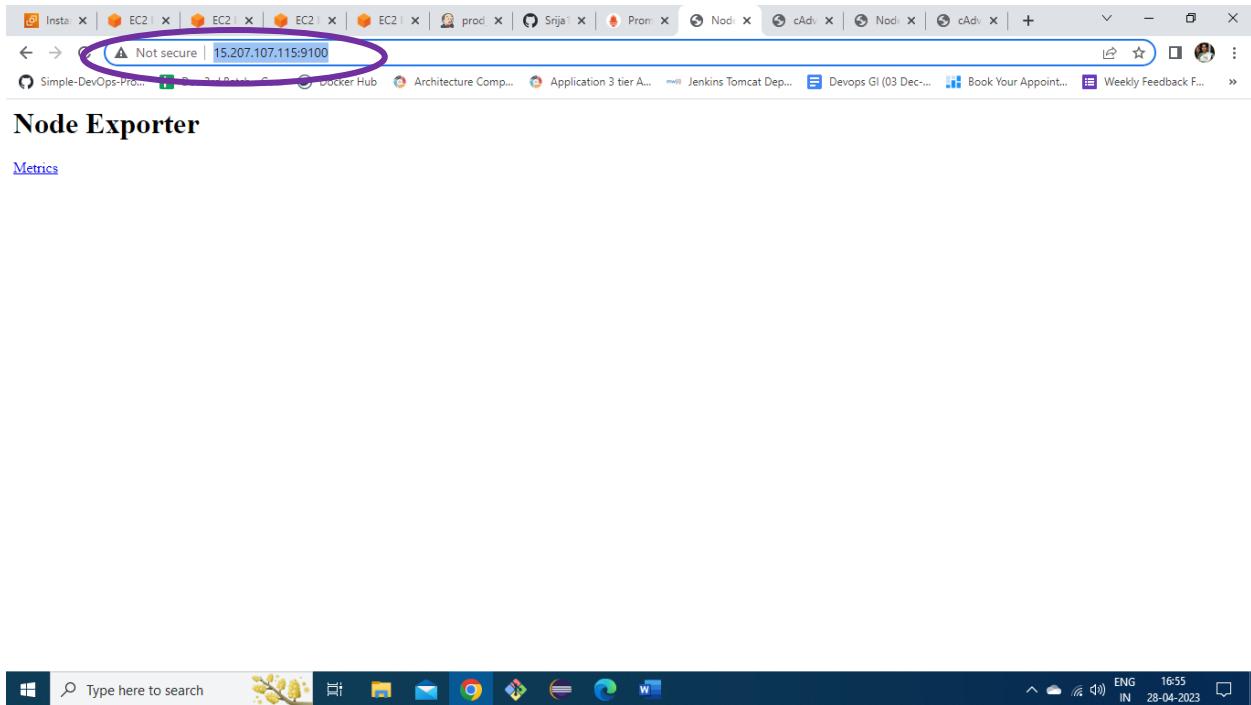


```
root@ip-172-31-36-240:/home/ubuntu# docker ps -a
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              PORTS
 NAMES
799f57ff6c57        gcr.io/cadvisor/cadvisor:v0.39.3   "/usr/bin/cadvisor ..."   24 minutes ago    Up 24 minutes (healthy)   0.0.0.0:8080->8080/tcp, :::8080-
>8080/tcp          cadvisor
b23aab7d6e79        prom/node-exporter:latest           "/bin/node_exporter ..."   25 minutes ago    Up 25 minutes
interesting_neumann dde84661795        srija1991/financeme      "java -jar /app.jar"     38 minutes ago    Up 38 minutes
>8081/tcp          wonderful_murdock

root@ip-172-31-36-240:/home/ubuntu#
```

i-05344b912772d65d9 (prod-server)
PublicIPs: 15.207.107.115 PrivateIPs: 172.31.36.240

The Node-exporter on the production-server configured on URL: 15.207.107.115:9100.



The cAdvisor on production server at URL: 15.207.107.115:8080.

The screenshot shows a web browser window with multiple tabs open. The active tab is titled "Not secure | 15.207.107.115:8080/containers/". The content of the page features a large owl logo above the word "cAdvisor". Below this, there are sections for "Docker Containers" and "Subcontainers". Under "Subcontainers", there are three entries: "/init.scope", "/system.slice", and "/user.slice". A sidebar on the left shows a file structure starting with "root". At the bottom of the page, there is an "Isolation" section. The browser's address bar shows the URL "15.207.107.115:8080/containers/". The taskbar at the bottom of the screen includes icons for File Explorer, Task View, Start, Taskbar settings, and system status.

The snapshot of the servers .

The screenshot shows the AWS Management Console with the EC2 service selected. The main pane displays the "Instances (4) info" table. The table has columns for Name, Instance ID, Instance state, Instance type, Status check, Alarm status, and Availability. All four instances listed are in a "Running" state. The table includes a search bar and filter options. The left sidebar shows the "Instances" section with sub-options like Instance Types, Launch Templates, and Spot Requests. The bottom navigation bar includes CloudShell, Feedback, Language, and links to Privacy, Terms, and Cookie preferences. The taskbar at the bottom of the screen is visible.

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability
Master-server	i-05578e7537bce2225	Running	t2.medium	2/2 checks passed...	No alarms	ap-sout
Prometheus_Grafana	i-03b139f2b343dd9b1	Running	t2.micro	2/2 checks passed...	No alarms	ap-sout
prod-server	i-05344b912772d65d9	Running	t2.micro	2/2 checks passed...	No alarms	ap-sout
test-server	i-0b1a1a1fb67e7526	Running	t2.micro	2/2 checks passed...	No alarms	ap-sout

Here is the Monitoring server, public IP:15.206.168.177

The screenshot shows the AWS EC2 Instances page. The instance summary for i-03b139f2b343dd9b1 (Prometheus_Grafana) is displayed. The Public IPv4 address is listed as 15.206.168.177, which is circled in purple. Other details shown include Instance ID (i-03b139f2b343dd9b1), Instance state (Running), Private IP address (172.31.36.123), Public IPv4 DNS (ec2-15-206-168-177.ap-south-1.compute.amazonaws.com), and VPC ID (vpc-0243302b4b1acc768).

Prometheus is configured to add the metrics on to the monitoring server.

The screenshot shows the AWS CloudShell terminal displaying the contents of the prometheus.yml file. The configuration includes static_configs for monitoring Node exporter and Cadvisor, and a job for Jenkins. The sections for monitoring Node exporter and Cadvisor are highlighted with blue ovals.

```
static_configs:
  - targets: ["localhost:9090"]

- job_name: "Monitoring Pipeline through Nodeexporter"
  static_configs:
  - targets: ["15.207.107.115:9100", "3.108.61.93:9100"]

- job_name: "Monitoring thorough Cadvisor"
  static_configs:
  - targets: ["15.207.107.115:8080", "3.108.61.93:8080"]

  # - job_name: "Node_exporter"
  # static_configs:
  # - targets: ['172.31.7.129:9100']

  # - job_name: "CAvisor"
  # static_configs:
  # - targets: ['172.31.7.129:8080']

  # - job_name: "Jenkins"
#prometheus.yml" 53L, 1594B
```

Prometheus GUI is accessed from URL: 15.206.168.177:9090 and targets are added.

The screenshot shows the Prometheus web interface with three sections:

- Monitoring Pipeline through Nodeexporter (2/2 up)**: Shows two targets:

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
http://15.207.107.115:9100/metrics	UP	instance="15.207.107.115:9100" job="Monitoring Pipeline through Nodeexporter"	12.477s ago	20.394ms	
http://3.108.61.93:9100/metrics	UP	instance="3.108.61.93:9100" job="Monitoring Pipeline through Nodeexporter"	5.161s ago	14.580ms	
- Monitoring thorough Cadvisor (2/2 up)**: Shows two targets:

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
http://15.207.107.115:8080/metrics	UP	instance="15.207.107.115:8080" job="Monitoring thorough Cadvisor"	7.421s ago	68.409ms	
http://3.108.61.93:8080/metrics	UP	instance="3.108.61.93:8080" job="Monitoring thorough Cadvisor"	8.927s ago	56.189ms	
- prometheus (1/1 up)**: Shows one target:

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
http://15.206.168.177:9090/metrics	UP	instance="15.206.168.177:9090" job="prometheus"	16.477s ago	20.394ms	

The browser taskbar at the bottom shows various open tabs including Insta, EC2, Docker Hub, Jenkins Tomcat Dep., Devops GI, Book Your Appoint..., and Weekly Feedback F...

The Grafana is accessed on URL: 15.206.168.177:3000

The screenshot shows the Grafana dashboard with the following components:

- General / Home**: The main dashboard view.
- Welcome to Grafana**: The welcome message.
- Need help?**: Links to Documentation, Tutorials, Community, and Public Slack.
- Dashboards**: A sidebar menu with options like Starred dashboards, GRAFANA_ASSIGNMENT, and Recently viewed dashboards.
- Latest from the blog**: A section displaying a recent blog post:

Apr 27
Grafana k6 v0.44.0 release: web crypto API, Web Vitals metrics, and more!
Grafana k6 v0.44.0 has been released, featuring new experimental modules, an upgraded browser module, and tons of improvements. Get Grafana k6 0.44.0 Here's a quick overview of the latest k6 news from the team and the community: Grafana k6 v0.44.0 release This release includes a new experimental web crypto module to perform basic cryptographic operations such as signing, hashing, verification, encryption, and decryption from your k6 tests. This module implements the Web Crypto API specification, and this first release has covered a subset of the most common APIs.

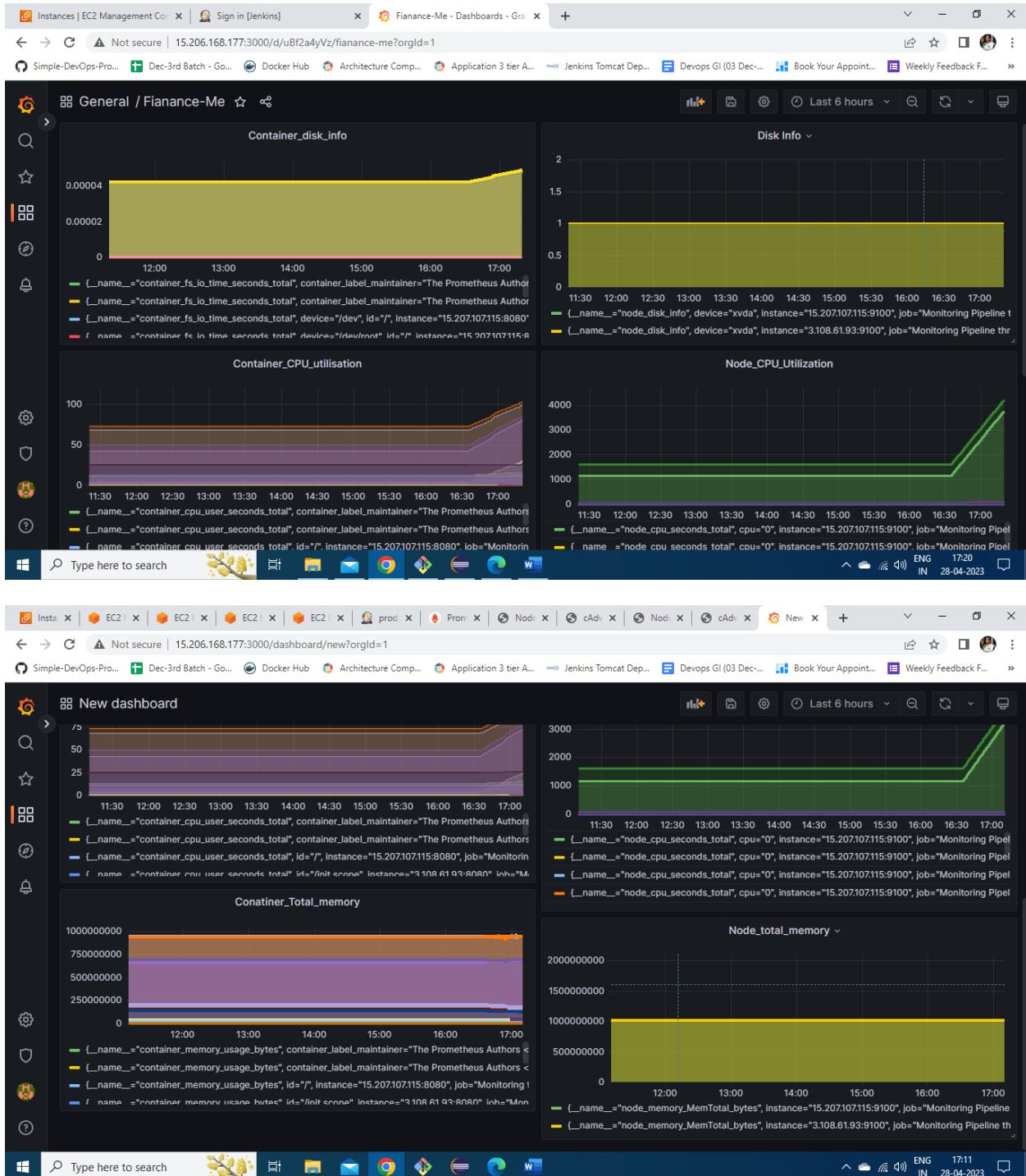
The browser taskbar at the bottom shows various open tabs including Insta, EC2, Docker Hub, Jenkins Tomcat Dep., Devops GI, Book Your Appoint..., and Weekly Feedback F...

The new data source is added as Prometheus 5.

This screenshot shows the Grafana interface for managing data sources. The title bar indicates the URL is 15.206.168.177:3000/datasources/edit/Kg1Nb4s4k. The main panel is titled "Data Sources / Prometheus-5" and specifies the type as "Prometheus". A prominent message box says: "Configure your Prometheus data source below" and "Or skip the effort and get Prometheus (and Loki) as fully-managed, scalable, and hosted data sources from Grafana Labs with the [free-forever Grafana Cloud plan](#)". Below this, there's a section for "HTTP" with fields for "URL" (set to http://15.206.168.177:9090/), "Allowed cookies" (with a "New tag" input field and an "Add" button), and "Timeout" (set to Timeout in seconds). A "Settings" tab is selected, and a "Dashboards" tab is also present. At the bottom right of the main panel, a green checkmark icon with the text "Datasource updated" is visible. The system tray at the bottom shows the date and time as 28-04-2023.

This screenshot shows the continuation of the data source configuration process. The URL in the title bar remains 15.206.168.177:3000/datasources/edit/Kg1Nb4s4k. The main panel now displays configuration options under "Type and version" and "Misc". Under "Type and version", the "Prometheus type" dropdown is set to "Choose". Under "Misc", there are settings for "Disable metrics lookup" (disabled), "Default Editor" (set to "Choose"), and "Custom query parameters" (example: max_source_resolution=5m&timeout=10). A "Scrape interval" section is also visible with values 15s and 60s for "Query timeout" and "HTTP method" respectively. A green checkmark icon with the text "Datasource updated" is shown in the top right. The status message "Data source is working" is displayed in a green box at the bottom. The system tray at the bottom shows the date and time as 28-04-2023.

The metrics Disk utilization of node and container, CPU utilization of Node and Container and Container and node memory of both test server and production server are observed.



THANK YOU