DevOps Certification

Training Certification Project –02

Finance Me - Banking and Finance Domain

The company's goal is to deliver the product updates frequently to production with High quality & Reliability. They also want to accelerate software delivery speed, quality and reduce feedback time between developers and testers.

Following are the problems the company is facing at the moment

✓ Building Complex builds is difficult
✓ Manual efforts to test various components/modules of the project
✓ Incremental builds are difficult to manage, test and deploy
✓ Creation of infrastructure and configure it manually is very time consuming
✓ Continuous manual monitoring of the application is quite challenging.

In order to implement a POC, you are requested to develop a Mavenaised microservice using spring boot and in memory h2 database.

1. A microservice which exposes below mentioned endpoints as APIs and uses in memory h2 database to store the data.

a. /createAccount (HTTP Method : POST) (Request Body : JSON)
b. /updateAccount/{account no.} (HTTP Method : PUT) (Request Body : JSON)
c. /viewPolicy/{account no.} (HTTP Method : GET) (No Request Body)
d. /deletePolicy/{account no.} (HTTP Method : DELETE) (No Request Body)

- Here I wrote the code with Junit test cases & added maven dependencies for the application using eclipse IDE and pushed it to the github repository.

https://github.com/Dilipkumar-M/FinanceMe

1. A microservice which exposes below mentioned endpoints as APIs and uses pre configured AWS RDS – mysql database to store the data.

a. /createAccount (HTTP Method : POST) (Request Body : JSON)
b. /updateAccount/{account no.} (HTTP Method : PUT) (Request Body : JSON)
c. /viewPolicy/{account no.} (HTTP Method : GET) (No Request Body)
d. /deletePolicy/{account no.} (HTTP Method : DELETE) (No Request Body)

eclipse-workspace - Finance/src/main/java/com/staragile/Banking/Finance/Account.java - Eclipse IDE

```java
package com.staragile.Banking.Finance;

import javax.persistence.Entity;

@Entity
public class Account {

    @Id
    private String accountId;
    private String customerName;
    private String customerAddress;
    private String contactNumber;

    public Account() {
        super();
    }

    public Account(String accountId, String customerName, String customerAddress, String contactNum
        super();
        this.accountId = accountId;
        this.customerName = customerName;
        this.customerAddress = customerAddress;
        this.contactNumber = contactNumber;
    }
```

eclipse-workspace - Finance/src/main/java/com/staragile/Banking/Finance/AccountApplication.java - Eclipse IDE

```java
package com.staragile.Banking.Finance;

import org.springframework.boot.SpringApplication;

@SpringBootApplication
public class AccountApplication {

    public static void main(String[] args) {
        SpringApplication.run(AccountApplication.class, args);
    }
}
```

eclipse-workspace - Finance/src/main/java/com/staragile/Banking/Finance/AccountController.java - Eclipse IDE

File Edit Source Refactor Source Navigate Search Project Run Window Help

Account.java | AccountApplication.java | *AccountController.java | AccountDAORepository.java | AccountService.java | DatabaseConnection.java | MvcFinanceController.java

```java
package com.staragile.Banking.Finance;

import java.util.List;

@RestController
@RequestMapping("/account")
public class AccountController {

    @Autowired
    private AccountService accountSvc;

    @RequestMapping("/hello")
    public String doSomething() {
        return "Hello World";
    }

    @RequestMapping("/seeddata")
    public String seedData() {
        System.out.println("Inside seeddata");
        Account account1 = new Account("1", "DiilipKmar", "DilipKumar Address", "923456789");
        accountSvc.addAccount(account1);

        Account account2 = new Account("2", "Sharanya", "Sharanya Address", "987654321");
        accountSvc.addAccount(account2);
```

eclipse-workspace - Finance/src/main/java/com/staragile/Banking/Finance/AccountDAORepository.java - Eclipse IDE

File Edit Source Refactor Source Navigate Search Project Run Window Help

Account.java | AccountApplication.java | *AccountController.java | AccountDAORepository.java | AccountService.java | DatabaseConnection.java | MvcFinanceController.java

```java
package com.staragile.Banking.Finance;

import org.springframework.data.repository.CrudRepository;

public interface AccountDAORepository extends CrudRepository<Account, String> {

}
```

eclipse-workspace - Finance/src/main/java/com/staragile/Banking/Finance/AccountService.java - Eclipse IDE

File Edit Source Refactor Source Navigate Search Project Run Window Help

Account.java | AccountApplication.java | *AccountController.java | AccountDAORepository.java | AccountService.java | DatabaseConnection.java | MvcFinanceController.java

```java
package com.staragile.Banking.Finance;

import org.springframework.beans.factory.annotation.Autowired;

@Service
public class AccountService {

    @Autowired
    private AccountDAORepository accountDAORepository;

    public List<Account> getAccounts() {
        List<Account> accountList = new ArrayList<>();
        accountDAORepository.findAll().forEach(accountList::add);
        return accountList;
    }

    public Optional<Account> getAccount(String id) {
        return accountDAORepository.findById(id);
    }

    public void addAccount(Account account) {
        accountDAORepository.save(account);
    }

    public void updateAccount(String id, Account account) {
```

eclipse-workspace - Finance/src/main/java/com/staragile/Banking/Finance/MvcFinanceController.java - Eclipse IDE

File Edit Source Refactor Source Navigate Search Project Run Window Help

Account.java | AccountApplication.java | *AccountController.java | AccountDAORepository.java | AccountService.java | DatabaseConnection.java | MvcFinanceController.java

```java
package com.staragile.Banking.Finance;

import org.springframework.beans.factory.annotation.Autowired;

@Controller
public class MvcFinanceController {

    @Autowired
    private AccountService accountSvc;

    @RequestMapping("mvchello")
    public String doSomething() {
        return "Hello World!!!";
    }

    @RequestMapping("/getallaccount")
    public String getAllAccounts(HttpServletRequest req, HttpServletResponse res) {
        List<Account> accountList = accountSvc.getAccounts();
        req.getSession().setAttribute("accountList", accountList);
        return "account.jsp";

        /*model.addAttribute("accountList", accountList);
        return "account"; // Return the name of the JSP file (without the .jsp extension)*/
    }
```

```java
package com.staragile.Banking.Finance;

import java.sql.Connection;

public class DatabaseConnection {
    public static void main(String[] args) {
        Connection conn = null;
        try {
            // Establish a connection to the H2 database
            conn = DriverManager.getConnection("jdbc:h2:file:C:/Users/user/test", "dilip", "dilip")

            // Perform database operations here
            // You can execute SQL queries using this connection.

            // Close the connection when done
            conn.close();
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
}
```

Here I wrote the code for following endpoints mentioned in the question 1).

2. Write necessary Junit test cases.

In eclipse choose file→java project→new project→policytest→new policytest.java

```java
package SeleniumProject.SeleniumProject;

import java.util.concurrent.TimeUnit;

Run All
public class BankingTest extends TestCase{

    //download chrome driver from https://googlechromelabs.github.io/chrome-for-testing/
    public String baseUrl = "http://localhost:8080/hello";
    String driverPath = "E:\\chromedriver-win64\\chromedriver-win64\\chromedriver.exe";

    public WebDriver driver;

    @Test
    Run | Debug
    public void testGoogle() {
        // set the system property for Chrome driver
        System.out.println("inside testGoogle");
        System.setProperty("webdriver.chrome.driver", driverPath);
        System.out.println("inside testGoogle "+driverPath);
        // Create driver object for CHROME browser
        driver = new ChromeDriver();
        driver.manage().timeouts().implicitlyWait(20, TimeUnit.SECONDS);
        driver.manage().window().maximize();
        driver.get(baseUrl);
```

3. Generate HTML reports using TestNG.

After file→java project→new project→policytest→new policytest.java run the application as TestNG application before that make sure you're installing the chromedriver.exe in the local machine and set the path in policytest.java.Download it from given link below:

https://googlechromelabs.github.io/chrome-for-testing/

/ersion: 118.0.5993.54 (r1192594)

Binary	Platform	URL	HTTP status
chrome	linux64	https://edgedl.me.gvt1.com/edgedl/chrome/chrome-for-testing/118.0.5993.54/linux64/chrome-linux64.zip	200
chrome	mac-arm64	https://edgedl.me.gvt1.com/edgedl/chrome/chrome-for-testing/118.0.5993.54/mac-arm64/chrome-mac-arm64.zip	200
chrome	mac-x64	https://edgedl.me.gvt1.com/edgedl/chrome/chrome-for-testing/118.0.5993.54/mac-x64/chrome-mac-x64.zip	200
chrome	win32	https://edgedl.me.gvt1.com/edgedl/chrome/chrome-for-testing/118.0.5993.54/win32/chrome-win32.zip	200
chrome	win64	https://edgedl.me.gvt1.com/edgedl/chrome/chrome-for-testing/118.0.5993.54/win64/chrome-win64.zip	200
chromedriver	linux64	https://edgedl.me.gvt1.com/edgedl/chrome/chrome-for-testing/118.0.5993.54/linux64/chromedriver-linux64.zip	200
chromedriver	mac-arm64	https://edgedl.me.gvt1.com/edgedl/chrome/chrome-for-testing/118.0.5993.54/mac-arm64/chromedriver-mac-arm64.zip	200
chromedriver	mac-x64	https://edgedl.me.gvt1.com/edgedl/chrome/chrome-for-testing/118.0.5993.54/mac-x64/chromedriver-mac-x64.zip	200
chromedriver	win32	https://edgedl.me.gvt1.com/edgedl/chrome/chrome-for-testing/118.0.5993.54/win32/chromedriver-win32.zip	200
chromedriver	win64	https://edgedl.me.gvt1.com/edgedl/chrome/chrome-for-testing/118.0.5993.54/win64/chromedriver-win64.zip	200
chrome-headless-shell	linux64	https://edgedl.me.gvt1.com/edgedl/chrome/chrome-for-testing/118.0.5993.54/linux64/chrome-headless-shell-linux64.zip	200
chrome-headless-shell	mac-arm64	https://edgedl.me.gvt1.com/edgedl/chrome/chrome-for-testing/118.0.5993.54/mac-arm64/chrome-headless-shell-mac-arm64.zip	200
chrome-headless-shell	mac-x64	https://edgedl.me.gvt1.com/edgedl/chrome/chrome-for-testing/118.0.5993.54/mac-x64/chrome-headless-shell-mac-x64.zip	200
chrome-headless-shell	win32	https://edgedl.me.gvt1.com/edgedl/chrome/chrome-for-testing/118.0.5993.54/win32/chrome-headless-shell-win32.zip	200
chrome-headless-shell	win64	https://edgedl.me.gvt1.com/edgedl/chrome/chrome-for-testing/118.0.5993.54/win64/chrome-headless-shell-win64.zip	200

Get the html reports after running the application

file:///E:/banking-project-staragile/SeleniumProject/test-output/emailable-report.html

Test	# Passed	# Skipped	# Retried	# Failed	Time (ms)	Included Groups	Excluded Groups
Default suite							
Default test	1	0	0	0	6,570		

Class	Method	Start	Time (ms)
Default suite			
Default test — passed			
SeleniumProject.SeleniumProject.BankingTest	testGoogle	1696410241276	6479

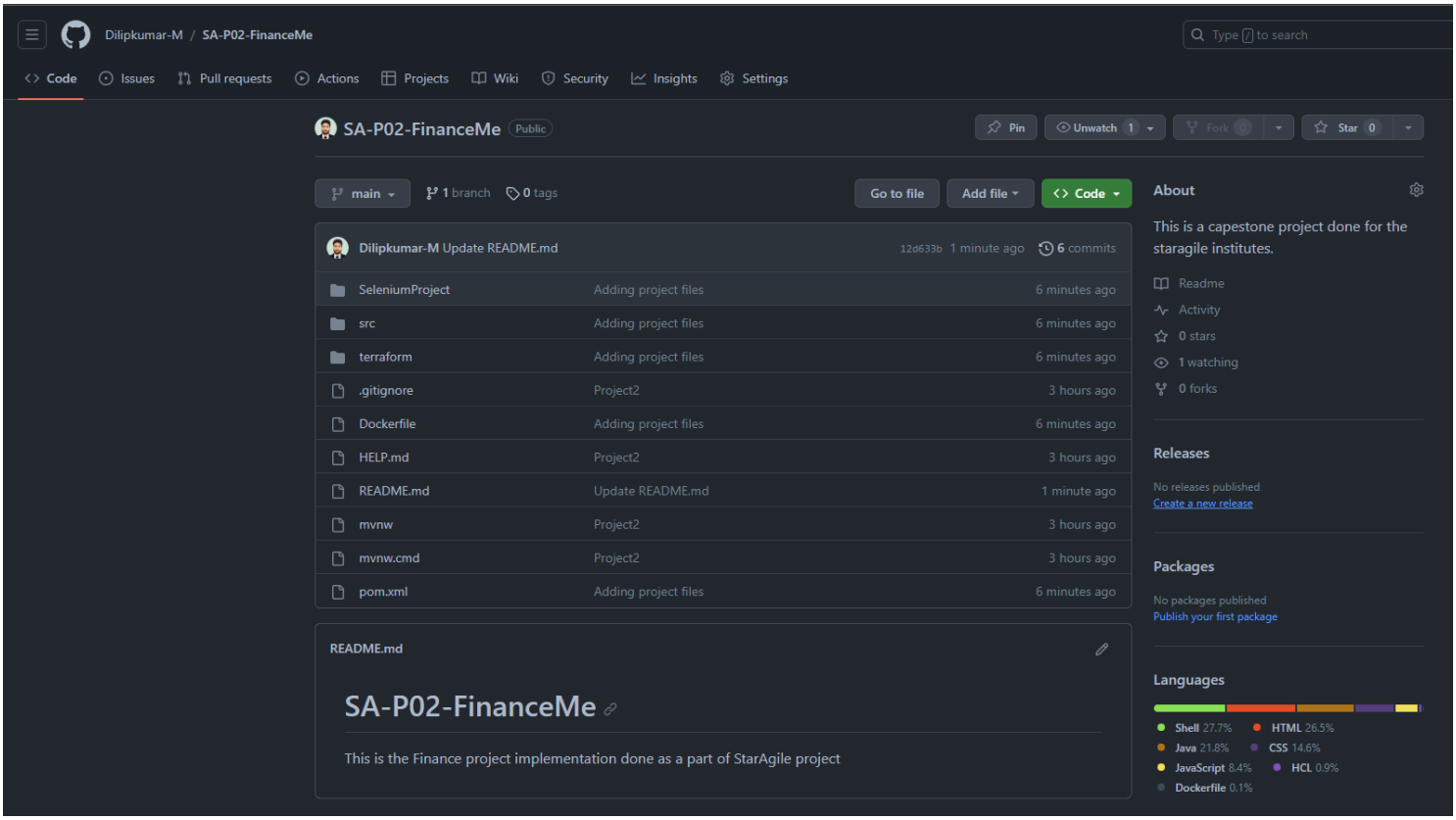
Default test

SeleniumProject.SeleniumProject.BankingTest#testGoogle

back to summary

4. Push your code into your GitHub Repository

```
 create mode 100644 SeleniumProject/test-output/testng-failed.xml
 create mode 100644 SeleniumProject/test-output/testng-reports.css
 create mode 100644 SeleniumProject/test-output/testng-reports.js
 create mode 100644 SeleniumProject/test-output/testng-reports1.css
 create mode 100644 SeleniumProject/test-output/testng-reports2.js
 create mode 100644 SeleniumProject/test-output/testng-results.xml
 create mode 100644 SeleniumProject/test-output/testng.css
 delete mode 100644 jenkins pipeline
 delete mode 100644 src/main/java/com/staragile/Banking/Finance/Account.java
 delete mode 100644 src/main/java/com/staragile/Banking/Finance/AccountControlle
r.java
 delete mode 100644 src/main/java/com/staragile/Banking/Finance/AccountDAORepos
itory.java
 delete mode 100644 src/main/java/com/staragile/Banking/Finance/AccountService.j
ava
 delete mode 100644 src/main/java/com/staragile/Banking/Finance/DatabaseConnecti
on.java
 delete mode 100644 src/main/java/com/staragile/Banking/Finance/MvcFinanceContro
ller.java
 create mode 100644 src/main/java/com/staragile/banking/banking/Account.java
 rename src/main/java/com/staragile/{Banking/Finance => banking/banking}/Account
Application.java (54%)
 create mode 100644 src/main/java/com/staragile/banking/banking/AccountControlle
r.java
 create mode 100644 src/main/java/com/staragile/banking/banking/AccountDAORepos
itory.java
 create mode 100644 src/main/java/com/staragile/banking/banking/AccountService.j
ava
 create mode 100644 src/main/java/com/staragile/banking/banking/MvcAccountContro
ller.java
 create mode 100644 src/main/webapp/Account.jsp
 delete mode 100644 src/test/java/com/staragile/Banking/Test/AccountControllerTe
sts.java
 create mode 100644 src/test/java/com/staragile/banking/banking/AccountApplicati
onTests.java
 create mode 100644 src/test/java/com/staragile/banking/banking/AccountTest.java
 create mode 100644 terraform/abc.sh
 create mode 100644 terraform/compute.tf
 create mode 100644 terraform/get-docker.sh
 create mode 100644 terraform/install_jenkins.sh
 create mode 100644 terraform/main.tf
 create mode 100644 terraform/terraform.tfvars
 create mode 100644 terraform/variables.tf

Dilip@Dilipkumar MINGW64 /e/SA-P02-FinanceMe (main)
$ git push
Enumerating objects: 106, done.
Counting objects: 100% (106/106), done.
Delta compression using up to 4 threads
Compressing objects: 100% (72/72), done.
Writing objects: 100% (92/92), 62.64 KiB | 2.98 MiB/s, done.
Total 92 (delta 10), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (10/10), completed with 3 local objects.
To https://github.com/Dilipkumar-M/SA-P02-FinanceMe.git
   af54a98..fb7fcfb  main -> main

Dilip@Dilipkumar MINGW64 /e/SA-P02-FinanceMe (main)
$
```



Here I pushed the application source code into my Github repository repo link is mentioned below:

https://github.com/Dilipkumar-M/FinanceMe

5. Implementing CI/CD in the aws instance.

Here I implemented Continuous Integration & Continuous Deployment of the application using DevOps tools like,Jenkins,Git,Docker,aws,Maven,Java.all these tools are configured with terraform and launched the Jenkins-instance from the local machine through aws configure using the accesskey and the secret key from the IAM security policies,and configured jenkins permissions to access the docker and the push image to dockerhub.

Files which are needed to start terraform:-

- **Compute.tf**

```
resource "aws_instance" "jenkins-instance" {
  ami             = var.instance_ami   #ubuntu ami for N.virginia region.
  instance_type   = var.instance_type
  key_name        = var.keyname
  #user_data = file("install_jenkins.sh")
  associate_public_ip_address = true
  tags = {
    Name = "Jenkins-Instance"
```

- **main.tf**

```
provider "aws" {
  region                  = "${var.region}"
  shared_credentials_files = ["~/.aws/credentials"]
}
```

- **terraform.tfvars**

```
region = "us-east-1"
instance_type = "t2.medium"
instance_ami = "ami-032df771421fcffbd"#"ami-053b0d53c279acc90"
keyname = "awsLinux"
```

- **variables.tf**

```
variable "region" {
  default = "us-east-1"
```

```
}
variable "instance_type" {

}
variable "instance_ami" {

}
variable "keyname" {
  default = "awsLinux"
}
```

Follow the commands to start the aws instance from the local machine:

In, terraform folder→terraform init→terraform plan→terraform apply→terraform destroy.

Terraform init

```
Dilip@Dilipkumar MINGW64 /e/Banking/terraform (main)
$ terraform init

Initializing the backend...

Initializing provider plugins...
- Reusing previous version of hashicorp/aws from the dependency lock file
- Using previously-installed hashicorp/aws v5.21.0

Terraform has been successfully initialized!

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,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
```

Terraform plan

```
Dilip@Dilipkumar MINGW64 /e/Banking/terraform (main)
$ terraform plan

Terraform used the selected providers to generate the following execution
plan. Resource actions are indicated with the following symbols:
  + create

Terraform will perform the following actions:

  # aws_instance.Terraform_handson will be created
  + resource "aws_instance" "Terraform_handson" {
      + ami                                  = "ami-032df771421fcffbd"
      + arn                                  = (known after apply)
      + associate_public_ip_address          = true
      + availability_zone                    = (known after apply)
      + cpu_core_count                       = (known after apply)
      + cpu_threads_per_core                 = (known after apply)
      + disable_api_stop                     = (known after apply)
      + disable_api_termination              = (known after apply)
      + ebs_optimized                        = (known after apply)
```

```
          + volume_type           = (known after apply)
        }
    }

Plan: 1 to add, 0 to change, 0 to destroy.

Do you want to perform these actions?
  Terraform will perform the actions described above.
  Only 'yes' will be accepted to approve.

  Enter a value: yes

aws_instance.Terraform_handson: Creating...
aws_instance.Terraform_handson: Still creating... [10s elapsed]
aws_instance.Terraform_handson: Still creating... [20s elapsed]
aws_instance.Terraform_handson: Still creating... [30s elapsed]
aws_instance.Terraform_handson: Still creating... [40s elapsed]
aws_instance.Terraform_handson: Creation complete after 48s [id=i-05bca61db79b0a
69b]

Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
```

Terraform apply

```
Dilip@Dilipkumar MINGW64 /e/Banking/terraform (main)
$ terraform apply

Terraform used the selected providers to generate the following execution
plan. Resource actions are indicated with the following symbols:
  + create

Terraform will perform the following actions:

  # aws_instance.Terraform_handson will be created
  + resource "aws_instance" "Terraform_handson" {
      + ami                                  = "ami-032df771421fcffbd"
      + arn                                  = (known after apply)
      + associate_public_ip_address          = true
      + availability_zone                    = (known after apply)
      + cpu_core_count                       = (known after apply)
      + cpu_threads_per_core                 = (known after apply)
      + disable_api_stop                     = (known after apply)
      + disable_api_termination              = (known after apply)
      + ebs_optimized                        = (known after apply)
      + get_password_data                    = false
      + host_id                              = (known after apply)
      + host_resource_group_arn              = (known after apply)
      + iam_instance_profile                 = (known after apply)
      + id                                   = (known after apply)
      + instance_initiated_shutdown_behavior = (known after apply)
      + instance_lifecycle                   = (known after apply)
      + instance_state                       = (known after apply)
      + instance_type                        = "t2.midium"
      + ipv6_address_count                   = (known after apply)
      + ipv6_addresses                       = (known after apply)
      + key_name                             = "awsLinux"
      + monitoring                           = (known after apply)
      + outpost_arn                          = (known after apply)
      + password_data                        = (known after apply)
      + placement_group                      = (known after apply)
      + placement_partition_number           = (known after apply)
      + primary_network_interface_id         = (known after apply)
      + private_dns                          = (known after apply)
      + private_ip                           = (known after apply)
      + public_dns                           = (known after apply)
      + public_ip                            = (known after apply)
      + secondary_private_ips                = (known after apply)
      + security_groups                      = (known after apply)
      + source_dest_check                    = true
      + spot_instance_request_id             = (known after apply)
      + subnet_id                            = (known after apply)
```

```
      + tags                                 = {
          + "Name" = "Terraform_handson"
        }
      + tags_all                             = {
          + "Name" = "Terraform_handson"
        }
      + tenancy                              = (known after apply)
      + user_data                            = (known after apply)
      + user_data_base64                     = (known after apply)
      + user_data_replace_on_change          = false
      + vpc_security_group_ids               = (known after apply)

      + ebs_block_device {
          + delete_on_termination = true
          + device_name           = "/dev/sda1"
          + encrypted             = (known after apply)
          + iops                  = (known after apply)
          + kms_key_id            = (known after apply)
          + snapshot_id           = (known after apply)
          + throughput            = (known after apply)
          + volume_id             = (known after apply)
          + volume_size           = 20
          + volume_type           = (known after apply)
        }
    }

Plan: 1 to add, 0 to change, 0 to destroy.

------------------------------------------------------------------------

Note: You didn't use the -out option to save this plan, so Terraform can't
guarantee to take exactly these actions if you run "terraform apply" now.
```

Aws instance created using terraform

☑	Terraform_handson	i-05bca61db79...	Running	t2.medium	Initializing...	No alarms	us-east-1d	ec2-54-80-225-1...	54.80.225.120
☐	Master	i-0a63c26ffa07...	Stopped	t2.micro	–	No alarms	us-east-1b	–	–
☐	Slave	i-0356355398c...	Stopped	t2.micro	–	No alarms	us-east-1b	–	–

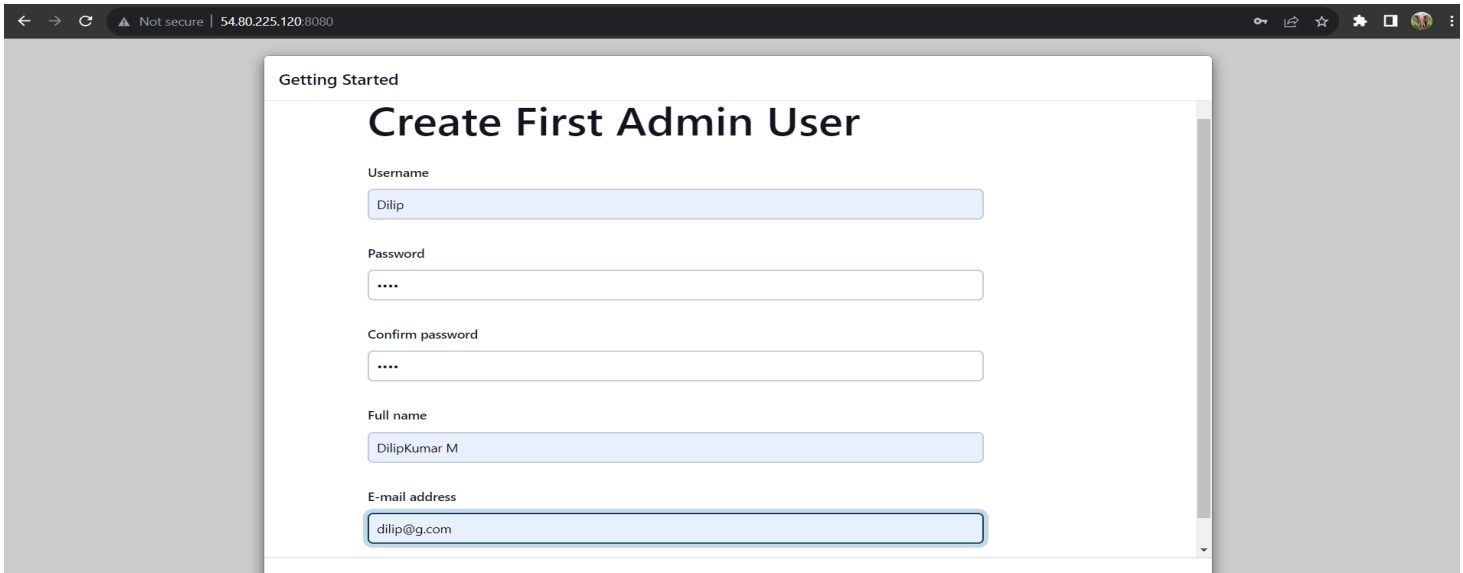
Above image shows how terraform created the Terraform-hands on instance which includes all the tools mentioned.
After initialization of the Jenkins-instance with terraform from the local machine, connect it to run the job of CICD with jenkins and docker to produce images from the container.and push it to the docker hub.

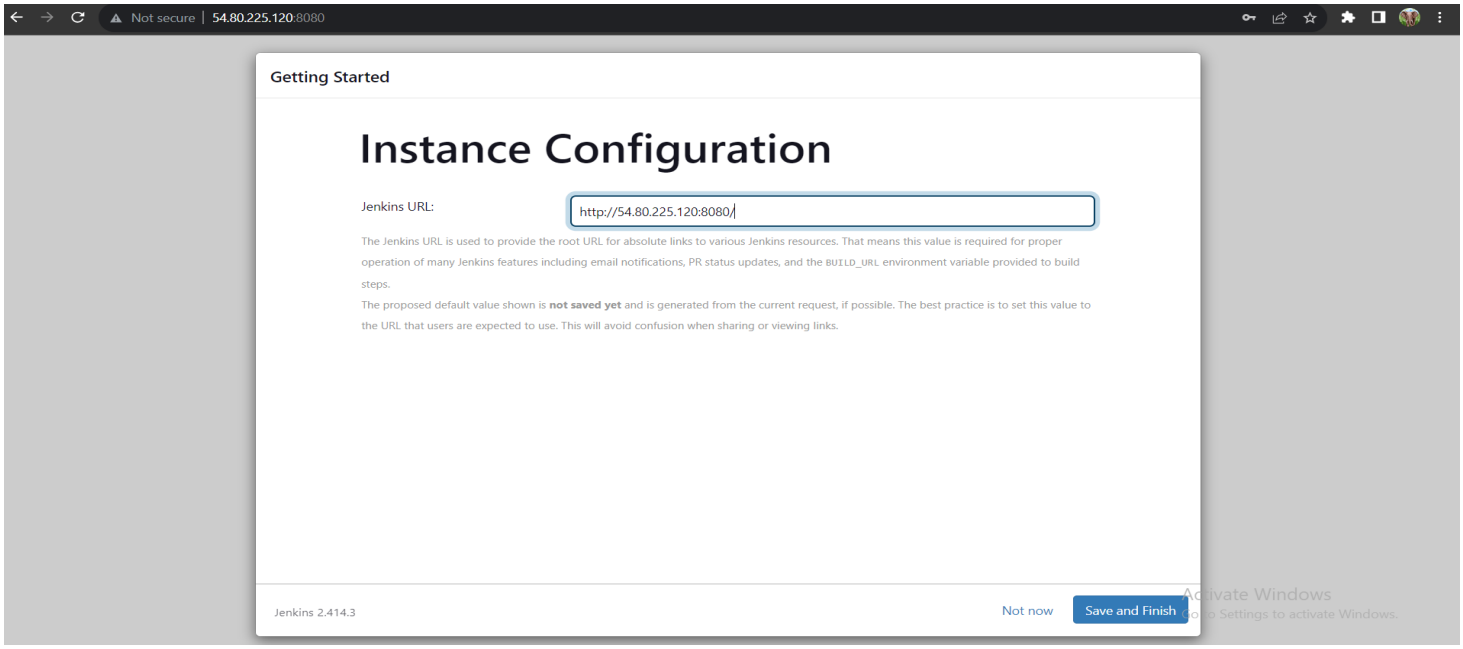
Here are the commands given in the form of shell script to the terraform folder to execute within the aws Virtual machine to install all the required Devops tools for the initial process.

#!/bin/bash

```
sudo apt update -y && apt upgrade -y

echo "Install Java JDK 8"
sudo apt remove -y java
sudo apt install default-jdk -y

echo "Install Maven"
sudo apt install maven -y

echo "Install git"
sudo apt install -y git

echo "Install Docker engine"
curl -fsSL https://get.docker.com -o get-docker.sh
sudo sh get-docker.sh

#sudo usermod -a -G docker jenkins
#sudo service docker start
#sudo chkconfig docker on

echo "Install Jenkins"
curl -fsSL https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key | sudo tee
/usr/share/keyrings/jenkins-keyring.asc > /dev/null
echo deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc]
https://pkg.jenkins.io/debian-stable binary/ | sudo tee  /etc/apt/sources.list.d/jenkins.list >
/dev/null
sudo apt-get update -y
sudo apt-get install fontconfig -y
sudo apt-get install jenkins -y
```

After running this script the tools were automatically installed in the respective virtual machine that have we chosen





After accessing the jenkins server choose the Freestyle job and provide the github project configure git access to the jenkins to run continuous integration,and change give permissions to the jenkins to configure docker user to it and run the below commands,and also add git SCM webhooks to the job.

After adding the webhooks to the jenkins job this webhooks alerts the developer with notifications whenever the source code of the project is deployed by users or the developers,then builds the CI/CD.

Commands as script in ShellExecution to do a job:

```
mvn clean install
docker build -t java-web-app-cicd:latest .
if (docker ps -a | grep 'java-web-app-cicd')
then
  docker stop java-web-app-cicd
  docker rm -f java-web-app-cicd
fi
docker run -d -p 8888:8080 --name java-web-app-cicd java-web-app-cicd
mvn test
docker login -u dilipkumar5656 -pdckr_pat_1foZ87ZAbTpjBKZgR9t242fAwlc
docker commit java-web-app-cicd dilipkumar5656/java-web-app-cicd:latest
docker push dilipkumar5656/java-web-app-cicd:latest
```

Add slave node centos

Nodes > CentosSlave1 > Log

```
XDG_SESSION_TYPE=tty
_='']'
Checking Java version in the PATH
openjdk version "11.0.20.1" 2023-08-24
OpenJDK Runtime Environment (build 11.0.20.1+1-post-Ubuntu-0ubuntu120.04)
OpenJDK 64-Bit Server VM (build 11.0.20.1+1-post-Ubuntu-0ubuntu120.04, mixed mode, sharing)
[10/01/23 11:18:17] [SSH] Checking java version of /home/jenkins-slave-01/jdk/bin/java
Couldn't figure out the Java version of /home/jenkins-slave-01/jdk/bin/java
bash: /home/jenkins-slave-01/jdk/bin/java: No such file or directory

[10/01/23 11:18:17] [SSH] Checking java version of java
[10/01/23 11:18:17] [SSH] java -version returned 11.0.20.1.
[10/01/23 11:18:17] [SSH] Starting sftp client.
[10/01/23 11:18:17] [SSH] Copying latest remoting.jar...
[10/01/23 11:18:18] [SSH] Copied 1,371,113 bytes.
Expanded the channel window size to 4MB
[10/01/23 11:18:18] [SSH] Starting agent process: cd "/home/jenkins-slave-01" && java  -jar remoting.jar -workDir /home/jenkins-slave-01 -jar-cache /home/jenkins-slave-01/remoting/jarCache
Oct 01, 2023 11:18:18 AM org.jenkinsci.remoting.engine.WorkDirManager initializeWorkDir
INFO: Using /home/jenkins-slave-01/remoting as a remoting work directory
Oct 01, 2023 11:18:18 AM org.jenkinsci.remoting.engine.WorkDirManager setupLogging
INFO: Both error and output logs will be printed to /home/jenkins-slave-01/remoting
<===[JENKINS REMOTING CAPACITY]===>channel started
Remoting version: 3131.vf2b_b_798b_ce99
Launcher: SSHLauncher
Communication Protocol: Standard in/out
This is a Unix agent
WARNING: An illegal reflective access operation has occurred
WARNING: Illegal reflective access by jenkins.slaves.StandardOutputSwapper$ChannelSwapper to constructor java.io.FileDescriptor(int)
WARNING: Please consider reporting this to the maintainers of jenkins.slaves.StandardOutputSwapper$ChannelSwapper
WARNING: Use --illegal-access=warn to enable warnings of further illegal reflective access operations
WARNING: All illegal access operations will be denied in a future release
Evacuated stdout
Agent successfully connected and online
```

6. Ansible Configuration

Installing ansible and configuring both machines and adding inventory files.

```
dilip@ip-172-31-41-223: /etc/ansible
[frontend]
172.31.43.56
~
~
~
```

```
dilip@ip-172-31-41-223:/etc/ansible$ ansible all -m ping
[WARNING]: Ansible is being run in a world writable directory (/etc/ansible),
ignoring it as an ansible.cfg source. For more information see
https://docs.ansible.com/ansible/devel/reference_appendices/config.html#cfg-in-
world-writable-dir
172.31.43.56 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
```

Write a playbook to install some tools in the Frontend server.

```
dilip@ip-172-31-41-223: /home/ubuntu/SA-P02-FinanceMe
- name : Configure Docker on EC2 Instances
  hosts : all
  become: true
  connection : ssh
  tasks :
  - name: updating apt
    command : sudo apt-get update

  - name : Install Docker
    command : sudo apt-get install -y docker.io

  - name : Start Docker Service
    command : sudo systemctl start docker

  - name: Deploy Docker Container
    command: docker run -itd -p 8084:8081 dilipkumar5656/java-web-app-cicd:1.0
~
~
~
~
~
~
~
~
~
~
```

Docker installation in frontend server using scripts

```
dilip@ip-172-31-41-223:/home/ubuntu$ sudo snap install docker
docker 20.10.24 from Canonical✓ installed
dilip@ip-172-31-41-223:/home/ubuntu$ docker --version
Docker version 20.10.24, build 297e128
```

```
Docker version 20.10.24, build 297e128
dilip@ip-172-31-41-223:/home/ubuntu$ docker run hello-world
docker: Got permission denied while trying to connect to the Docker daemon soc
: connect: permission denied.
See 'docker run --help'.
dilip@ip-172-31-41-223:/home/ubuntu$ sudo docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
719385e32844: Pull complete
Digest: sha256:4f53e2564790c8e7856ec08e384732aa38dc43c52f02952483e3f003afbf23c
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
 $ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
 https://hub.docker.com/

For more examples and ideas, visit:
 https://docs.docker.com/get-started/
```

Jenkins

Search (CTRL+K) DilipKumar M log out

Dashboard > CICD > #1 > Console Output

- Status
- Changes
- Console Output
- View as plain text
- Edit Build Information
- Git Build Data

Console Output

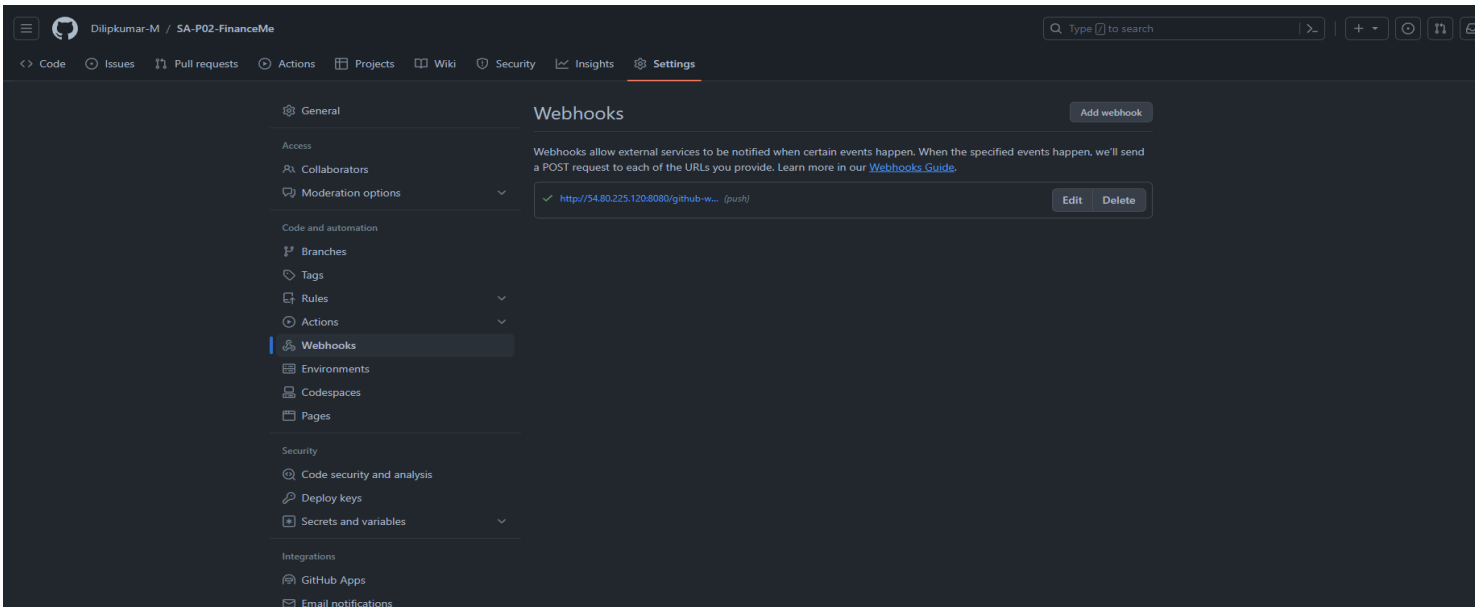
```
Started by user DilipKumar M
Running as SYSTEM
Building in workspace /var/lib/jenkins/workspace/CICD
The recommended git tool is: NONE
No credentials specified
Cloning the remote Git repository
Cloning repository https://github.com/Dilipkumar-M/SA-P02-FinanceMe
 > git init /var/lib/jenkins/workspace/CICD # timeout=10
Fetching upstream changes from https://github.com/Dilipkumar-M/SA-P02-FinanceMe
 > git --version # timeout=10
 > git --version # 'git version 2.34.1'
 > git fetch --tags --force --progress -- https://github.com/Dilipkumar-M/SA-P02-FinanceMe +refs/heads/*:refs/remotes/origin/* # timeout=10
 > git config remote.origin.url https://github.com/Dilipkumar-M/SA-P02-FinanceMe # timeout=10
 > git config --add remote.origin.fetch +refs/heads/*:refs/remotes/origin/* # timeout=10
Avoid second fetch
 > git rev-parse refs/remotes/origin/main^{commit} # timeout=10
Checking out Revision 12d633be40c856596970c9b90279e47cbde8559d (refs/remotes/origin/main)
 > git config core.sparsecheckout # timeout=10
 > git checkout -f 12d633be40c856596970c9b90279e47cbde8559d # timeout=10
Commit message: "Update README.md"
First time build. Skipping changelog.
[CICD] $ /bin/sh -xe /tmp/jenkins306674314443095792 2.sh
+ mvn clean install
[INFO] Scanning for projects...
Downloading from spring-snapshots: https://repo.spring.io/snapshot/org/springframework/boot/spring-boot-starter-parent/2.7.16-SNAPSHOT/maven-metadata.xml
Progress (1): 810 B

Downloaded from spring-snapshots: https://repo.spring.io/snapshot/org/springframework/boot/spring-boot-starter-parent/2.7.16-SNAPSHOT/maven-metadata.xml (810 B at 2.4 kB/s)
Downloading from spring-snapshots: https://repo.spring.io/snapshot/org/springframework/boot/spring-boot-starter-parent/2.7.16-SNAPSHOT/spring-boot-starter-parent-2.7.16-20230921.003107-39.pom
Progress (1): 3.2/9.2 kB
```

```
9c742cd6c7a5: Preparing
b626401ef603: Waiting
293d5db30c9f: Waiting
9b55156abf26: Waiting
03127cdb479b: Waiting
9c742cd6c7a5: Waiting
826c3ddbb29c: Layer already exists
9386e5a7a930: Layer already exists
7b7f3078e1db: Layer already exists
9b55156abf26: Layer already exists
b626401ef603: Layer already exists
293d5db30c9f: Layer already exists
03127cdb479b: Layer already exists
9c742cd6c7a5: Layer already exists
67db6c29cedd: Pushed
89e769b92cb5: Pushed
latest: digest: sha256:7bf0385fa760f613327cb586e6cda02c09a4542ec07ab221616a80172f9b3da2 size: 2423
Finished: SUCCESS
```

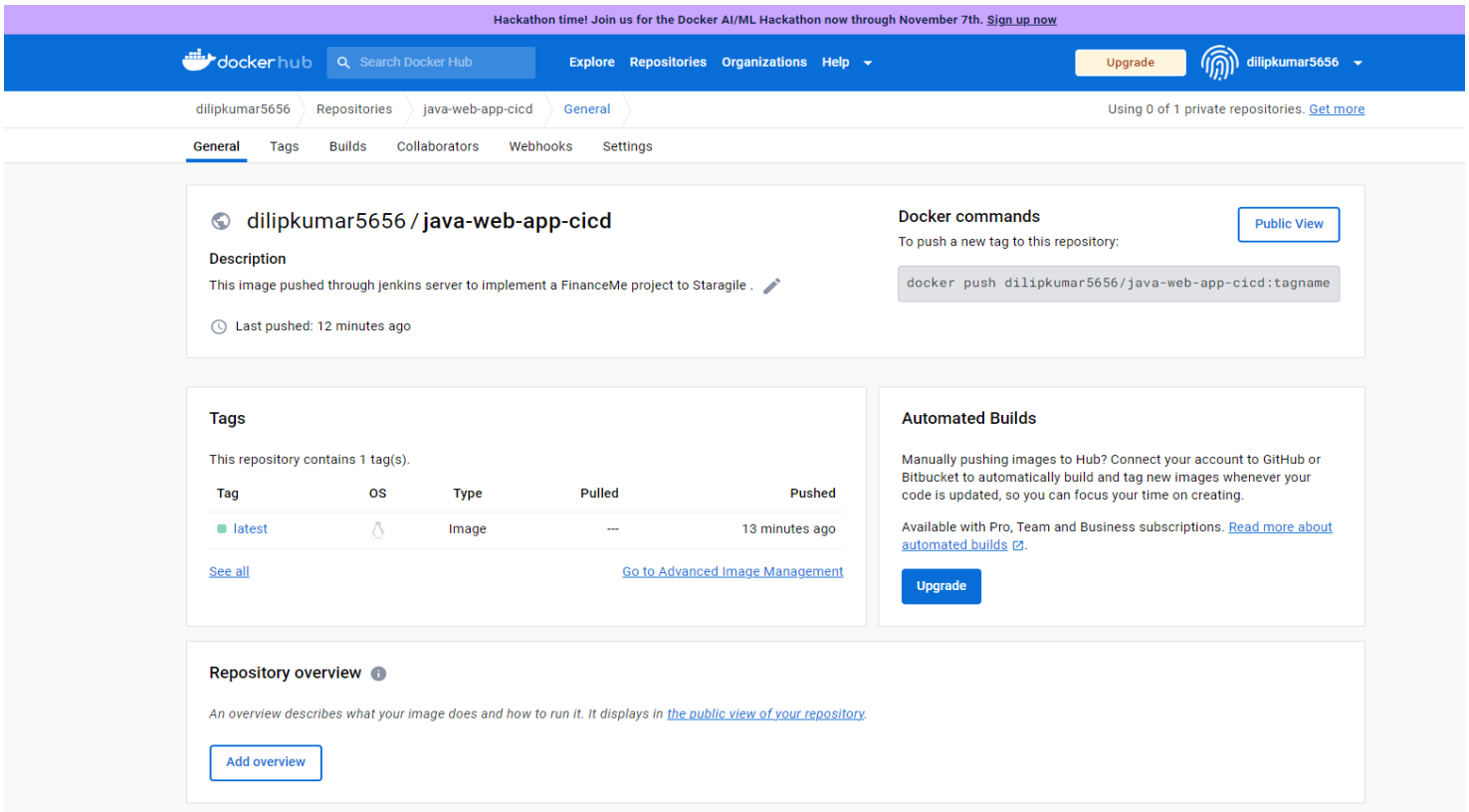
Give permissions to the jenkins slave to configure with docker to push the images and restart both jenkins and docker.

sudo usermod -aG docker jenkins-centos
sudo systemctl restart docker
sudo systemctl restart jenkins

Add Webhook

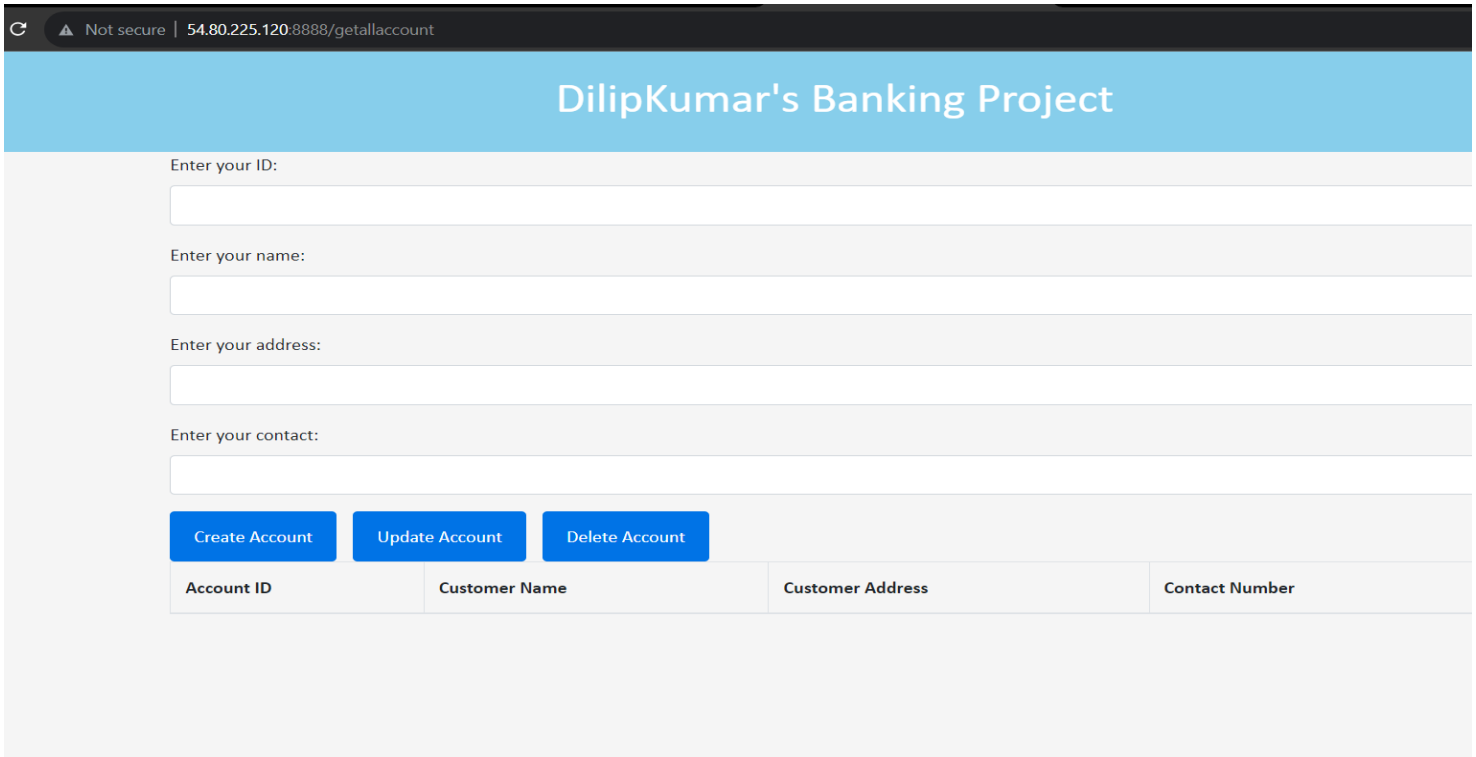


After all these processes the jenkins pushes automatically the image into the docker hub account repository.



OUTPUT:

http://54.80.225.120:8888/getallaccount



Here I added the users to conduct the operations CREATE,UPDATE and DELETE node points, form bank website.

a.CREATE

Not secure | 54.80.225.120:8888/createaccount

DilipKumar's Banking Project

Enter your ID:

Enter your name:

Enter your address:

Enter your contact:

Create Account | Update Account | Delete Account

Account ID	Customer Name	Customer Address	Contact Number
1	DILIPKUMAR M	Srinivaspura Taluk,Kolar District,Karnataka.	9090923454
2	Sharanya K	Shivajinagar,Bangalore,Karnataka.	9686876151
3	Karthik	Mumbai,India	9451234555

b.UPDATE

Not secure | 54.80.225.120:8888/updateaccount

DilipKumar's Banking Project

Enter your ID:

Enter your name:

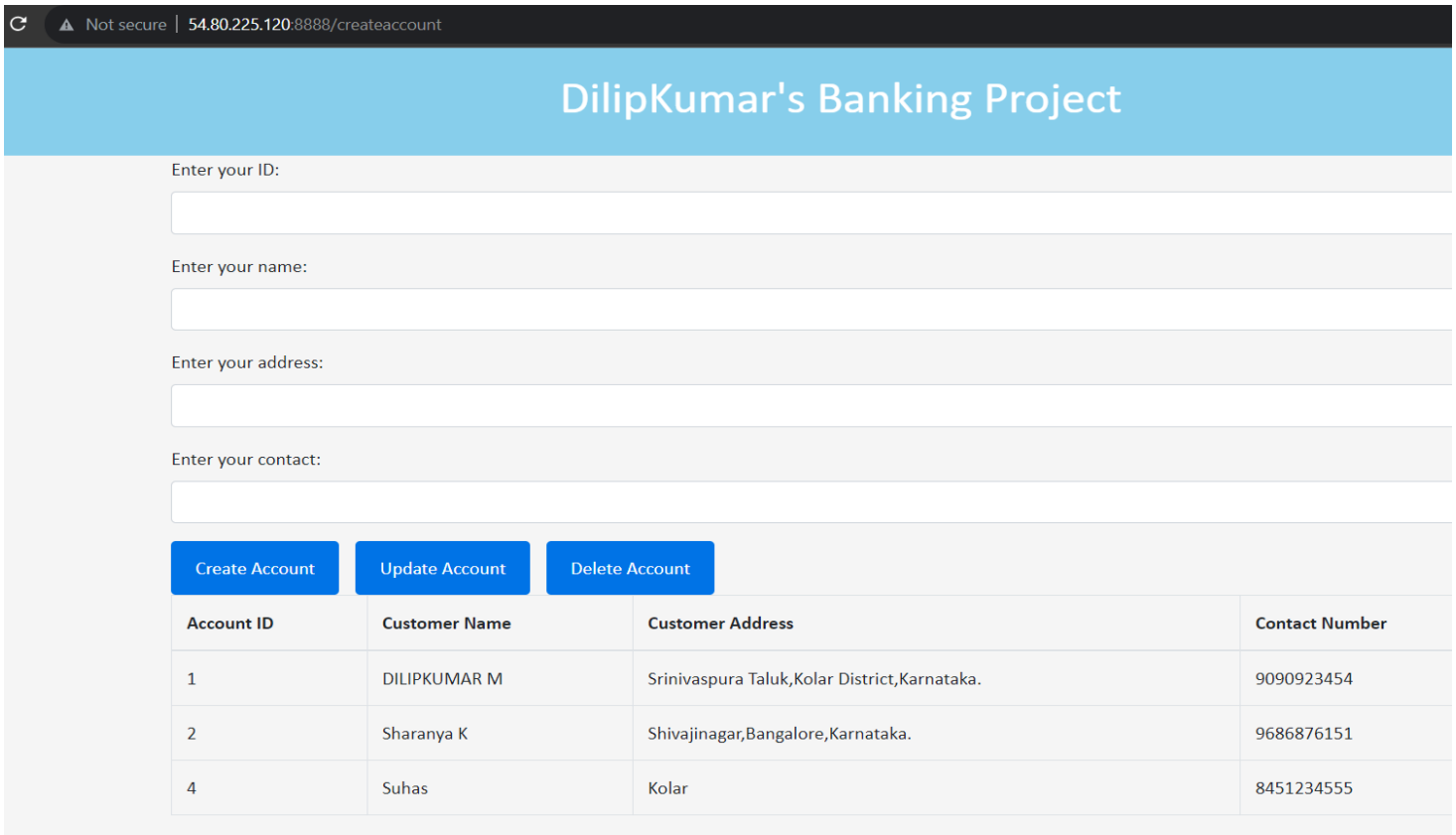
Enter your address:

Enter your contact:

Create Account | Update Account | Delete Account

Account ID	Customer Name	Customer Address	Contact Number
1	Sharanya S	Ramnagar,Karnataka	9451234553

c.DELETE



use the outcome of this Ansible playbook to determine whether updates occurred in the Git repository on the target machine. If updates were made, i can perform additional tasks or actions as needed based on the result.

https://github.com/Dilipkumar-M/SA-P02-FinanceMe

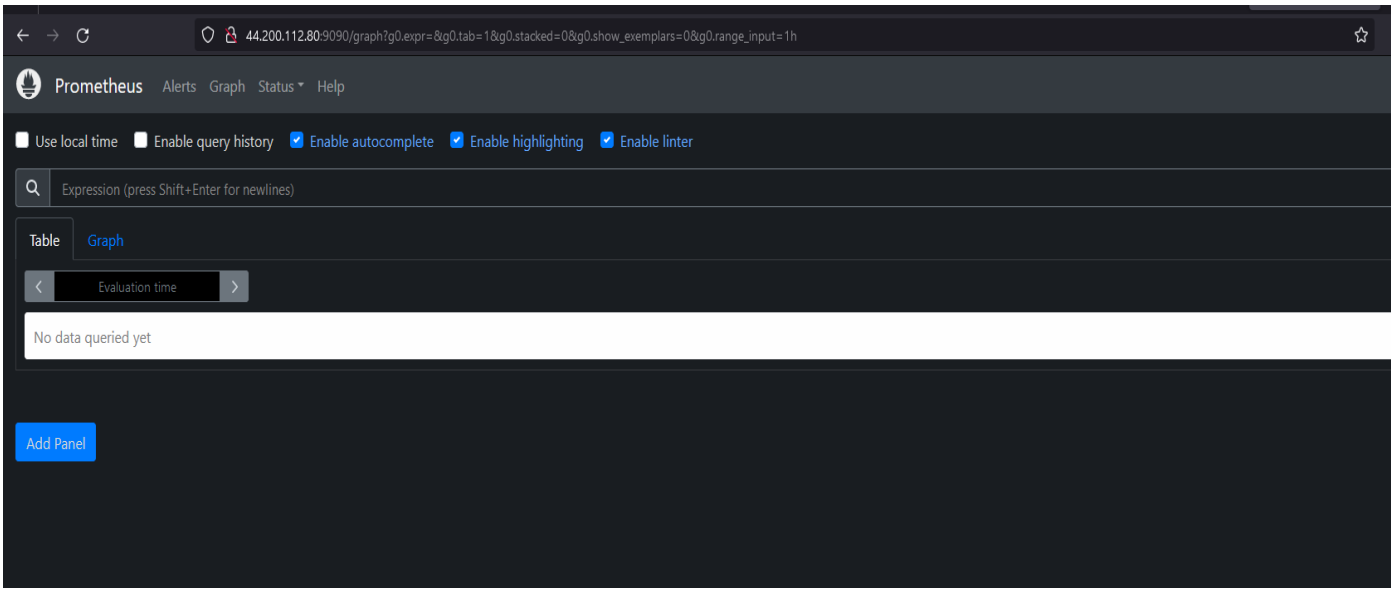
Continuous monitoring setup is established ,by installing prometheus and grafana in jenkins server and node- exporter.

Install Docker-compose:

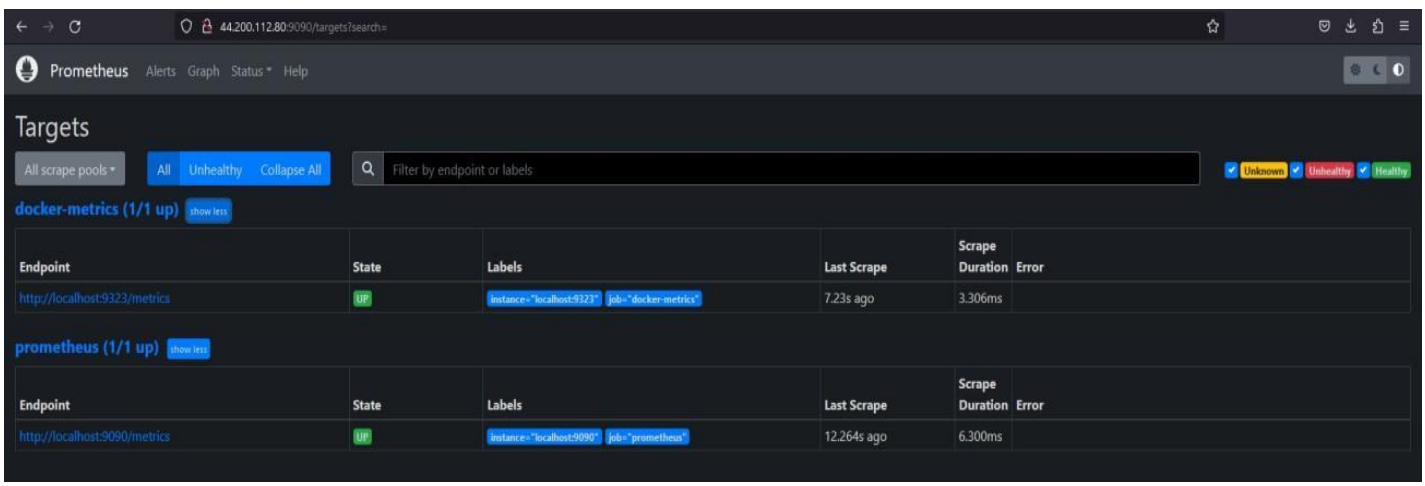
```
[root@ip-172-31-75-217 bin]# docker-compose
Define and run multi-container applications with Docker.

Usage:
  docker-compose [-f <arg>...] [options] [COMMAND] [ARGS...]
  docker-compose -h|--help

Options:
  -f, --file FILE             Specify an alternate compose file
```

```
[root@ip-172-31-75-217 bin]# vi /etc/docker/daemon.json
[root@ip-172-31-75-217 bin]# vi /etc/docker/daemon.json
[root@ip-172-31-75-217 bin]# systemctl restart docker
[root@ip-172-31-75-217 bin]# systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/usr/lib/systemd/system/docker.service; enab
t: disabled)
```

```
{
  "exec-opts" : ["native.cgroupdriver=systemd"] ,
  "log-driver" :"json-file" ,
  "log-opts" :{
    "max-size" : "100m"
  },
  "storage-driver" : "overlay2",
  "metrics-addr" : "127.0.0.1:9323" ,
  "experimental" : true
}
~
```



```yaml
# my global config
global:
  scrape_interval: 15s
  evaluation_interval: 15s

# Alertmanager configuration
alerting:
  alertmanagers:
    - static_configs:
        - targets:
          # - alertmanager:9093

# Load rules once and periodically evaluate them according to the global 'evaluation_interval'.
rule_files:
  #- "first_rules.yml"
  #- "second_rules.yml"

# A scrape configuration containing exactly one endpoint to scrape:
# Here it's Prometheus itself.
scrape_configs:
  - job_name: "prometheus"
    static_configs:
      - targets: ["localhost:9090"]
  - job_name: 'docker-metrics'
    scrape_interval: 5s
    static_configs:
      - targets: ['localhost:9323']
```



Add the scrape configuration to the jenkins

```yaml
# Add a new scrape configuration for Jenkins
  - job_name: 'jenkins'
    metrics_path: '/prometheus'
    static_configs:
      - targets: ['54.89.102.168:8080']
~
~
```

Not secure | 44.203.178.158:8080/manage/configure

Dashboard > Manage Jenkins > System >

Prometheus

Path ?

prometheus

Default Namespace ?

default

☐ Enable authentication for prometheus end-point ?

Collecting metrics period in seconds ?

120

☑ Count duration of successful builds ?

☑ Count duration of unstable builds ?

☑ Count duration of failed builds ?

☑ Count duration of not-built builds ?

☑ Count duration of aborted builds ?

☑ Fetch the test results of builds ?

☐ Add build parameter label to metrics ?

☐ Add build status label to metrics ?

☐ Process disabled jobs ?

Job attribute name ?

jenkins_job

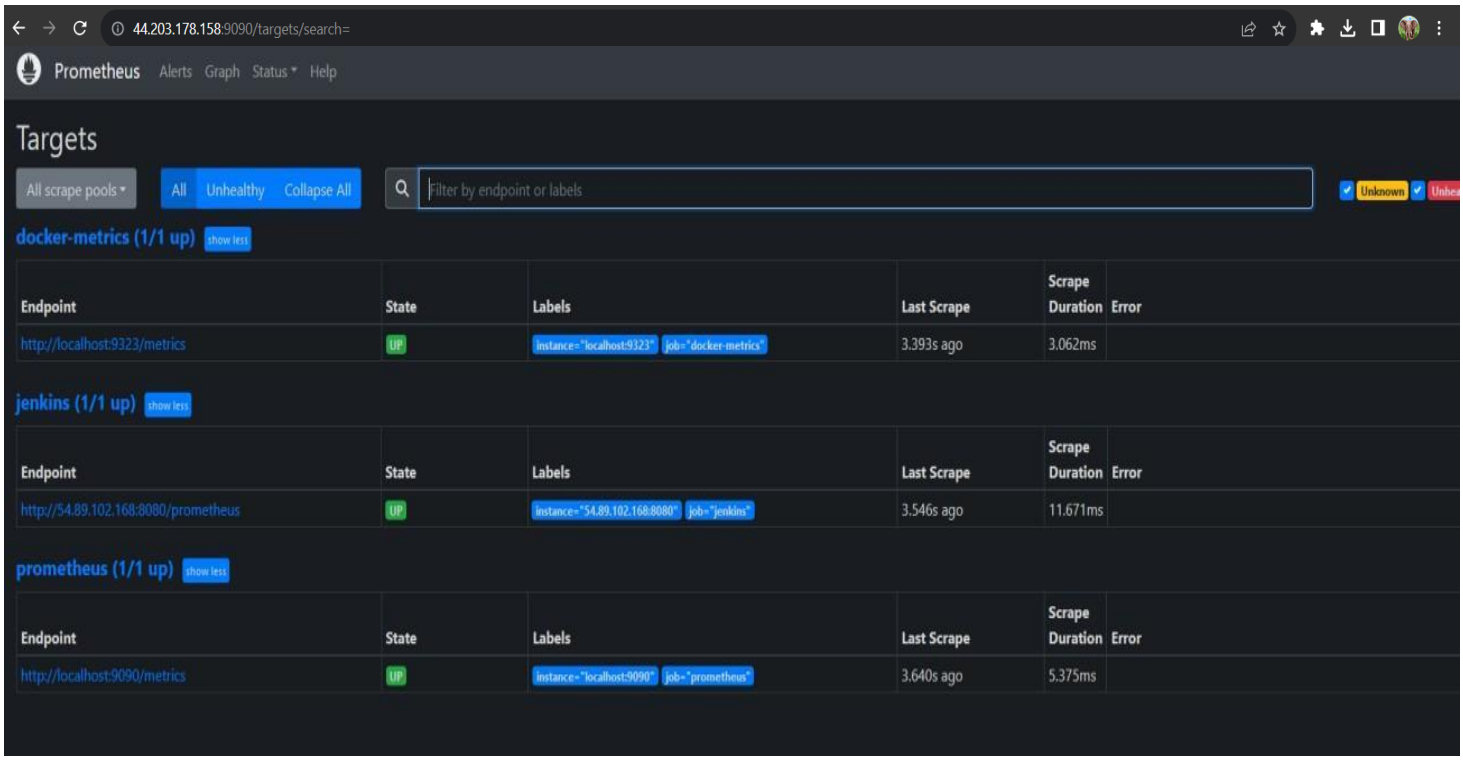
Activate Windows
Go to Settings to activate Windows.

Not secure | 44.203.178.158:8080/prometheus/

```
# HELP jvm_threads_current Current thread count of a JVM
# TYPE jvm_threads_current gauge
jvm_threads_current 33.0
# HELP jvm_threads_daemon Daemon thread count of a JVM
# TYPE jvm_threads_daemon gauge
jvm_threads_daemon 18.0
# HELP jvm_threads_peak Peak thread count of a JVM
# TYPE jvm_threads_peak gauge
jvm_threads_peak 41.0
# HELP jvm_threads_started_total Started thread count of a JVM
# TYPE jvm_threads_started_total counter
jvm_threads_started_total 99.0
# HELP jvm_threads_deadlocked Cycles of JVM-threads that are in deadlock waiting to acquire object monitors or ownable synchronizers
# TYPE jvm_threads_deadlocked gauge
jvm_threads_deadlocked 0.0
# HELP jvm_threads_deadlocked_monitor Cycles of JVM-threads that are in deadlock waiting to acquire object monitors
# TYPE jvm_threads_deadlocked_monitor gauge
jvm_threads_deadlocked_monitor 0.0
# HELP jvm_threads_state Current count of threads by state
# TYPE jvm_threads_state gauge
jvm_threads_state{state="NEW",} 0.0
jvm_threads_state{state="TERMINATED",} 0.0
jvm_threads_state{state="RUNNABLE",} 6.0
jvm_threads_state{state="BLOCKED",} 0.0
jvm_threads_state{state="WAITING",} 14.0
jvm_threads_state{state="TIMED_WAITING",} 13.0
jvm_threads_state{state="UNKNOWN",} 0.0
# HELP jvm_classes_currently_loaded The number of classes that are currently loaded in the JVM
# TYPE jvm_classes_currently_loaded gauge
jvm_classes_currently_loaded 14628.0
# HELP jvm_classes_loaded_total The total number of classes that have been loaded since the JVM has started execution
# TYPE jvm_classes_loaded_total counter
jvm_classes_loaded_total 14629.0
# HELP jvm_classes_unloaded_total The total number of classes that have been unloaded since the JVM has started execution
# TYPE jvm_classes_unloaded_total counter
jvm_classes_unloaded_total 1.0
# HELP process_cpu_seconds_total Total user and system CPU time spent in seconds.
# TYPE process_cpu_seconds_total counter
process_cpu_seconds_total 72.26
# HELP process_start_time_seconds Start time of the process since unix epoch in seconds.
# TYPE process_start_time_seconds gauge
process_start_time_seconds 1.696496206353E9
# HELP process_open_fds Number of open file descriptors.
# TYPE process_open_fds gauge
process_open_fds 306.0
# HELP process_max_fds Maximum number of open file descriptors.
# TYPE process_max_fds gauge
process_max_fds 524288.0
# HELP process_virtual_memory_bytes Virtual memory size in bytes.
# TYPE process_virtual_memory_bytes gauge
process_virtual_memory_bytes 3.74552576E9
# HELP process_resident_memory_bytes Resident memory size in bytes.
# TYPE process_resident_memory_bytes gauge
```

Restart Jenkins and login into it to establish Grafana and prometheus

Continuous monitoring setup is established ,by installing prometheus and grafana in server and node- exporter and I also included monitoring of jenkins –server and also docker-metrics.



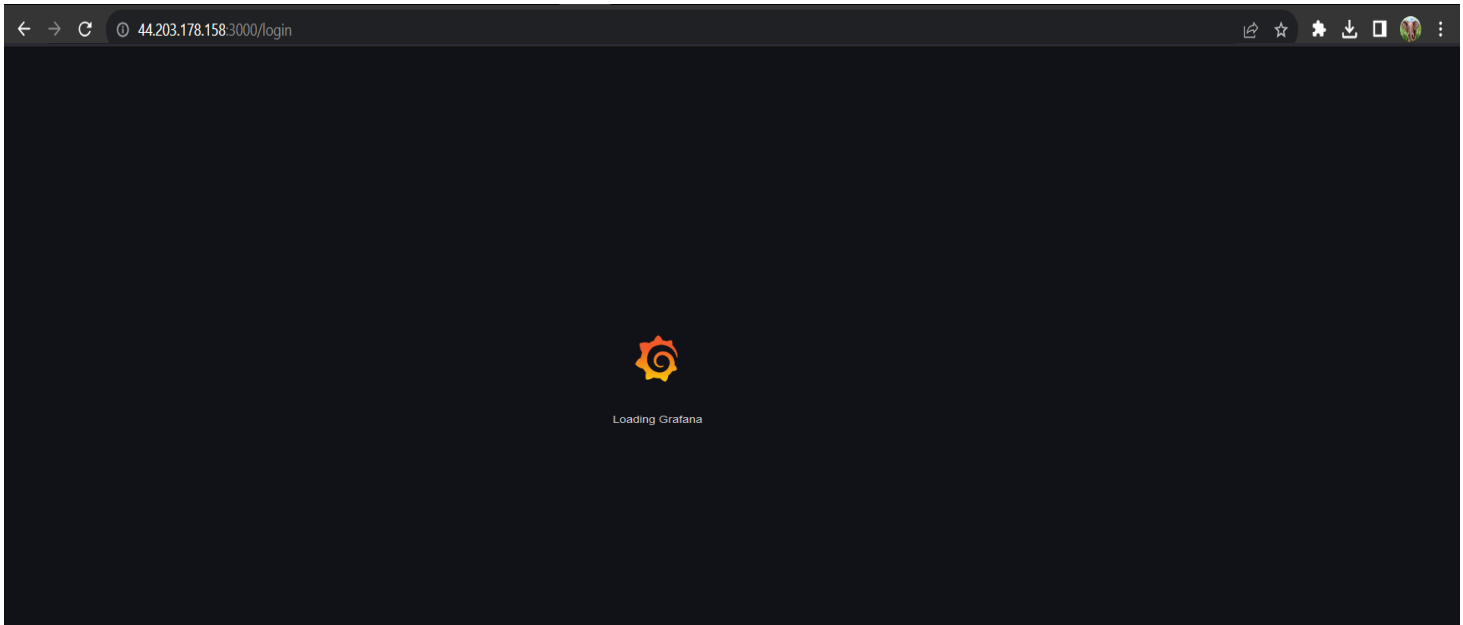

```
# Corrected indentation for the node_exporter job
  - job_name: 'node_exporter'
    metrics_path: '/metrics' # Use '/metrics' as it's the default path for Node Exporter
    static_configs:
      - targets: ['localhost:9100'] # Assuming Node Exporter is running on the same machine as Prometheus
```

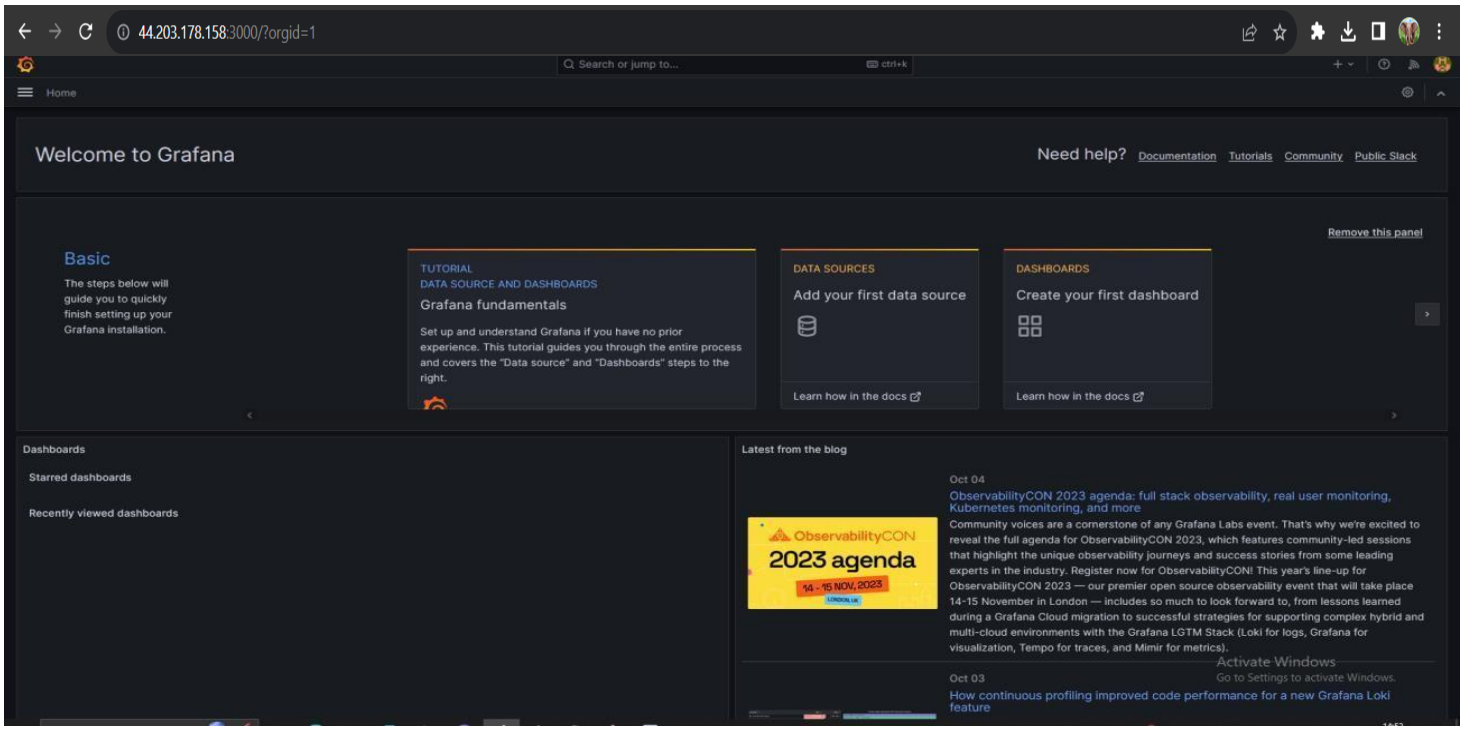
Commands to Installation of grafana:

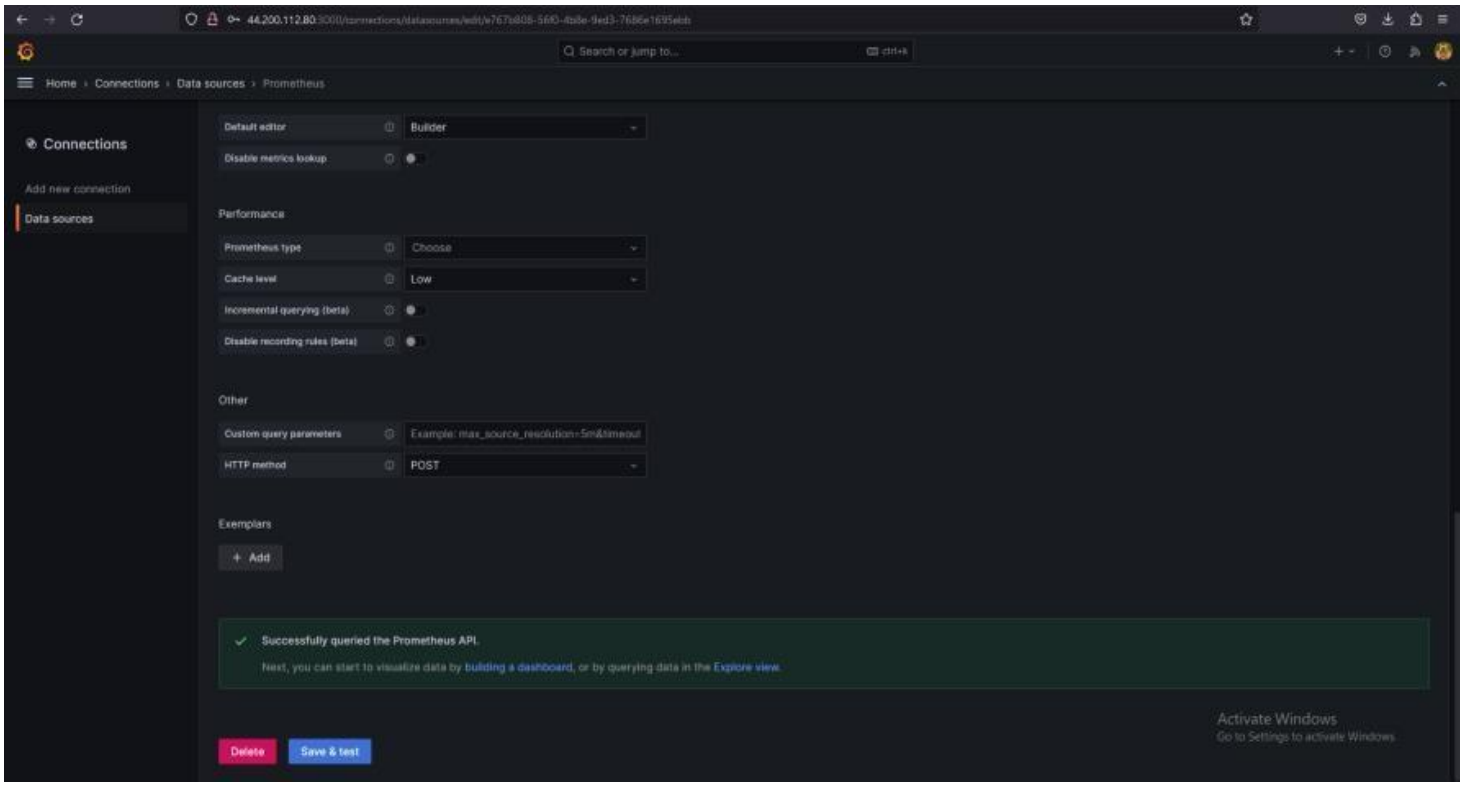
- sudo yum update -y
- sudo vi /etc/yum.repos.d/grafana.repo
- [grafana]
- name-grafana
- baseurl=https://packages.grafana.com/oss/rpm
- repo_gpgcheck-1
- enabled-1
- gpgcheck-1
- gpgkey-https://packages.grafana.com/gpg.key sslverify-1
- sslcacert-/etc/pki/tls/certs/ca-bundle.crt

- sudo yum install grafana -y

```
[ec2-user@ip-172-31-75-217 prometheus]$ sudo systemctl start grafana-server
[ec2-user@ip-172-31-75-217 prometheus]$ sudo systemctl status grafana-server
● grafana-server.service - Grafana instance
   Loaded: loaded (/usr/lib/systemd/system/grafana-server.service; disabled; vendor preset: disabled)
   Active: active (running) since Thu 2023-10-05 09:17:51 UTC; 11s ago
     Docs: http://docs.grafana.org
 Main PID: 2840 (grafana)
    Tasks: 7
   Memory: 84.6M
   CGroup: /system.slice/grafana-server.service
           └─2840 /usr/share/grafana/bin/grafana server --config=/etc/grafana/grafana.ini --pidfile=/var/run/

Oct 05 09:17:51 ip-172-31-75-217.ec2.internal grafana[2840]: logger=modules t=2023-10-05T09:17:51.607866243Z
Oct 05 09:17:51 ip-172-31-75-217.ec2.internal grafana[2840]: logger=ngalert.state.manager t=2023-10-05T09:17:
Oct 05 09:17:51 ip-172-31-75-217.ec2.internal grafana[2840]: logger=ngalert.state.manager t=2023-10-05T09:17:
Oct 05 09:17:51 ip-172-31-75-217.ec2.internal grafana[2840]: logger=ngalert.scheduler t=2023-10-05T09:17:51.6
Oct 05 09:17:51 ip-172-31-75-217.ec2.internal grafana[2840]: logger=ticker t=2023-10-05T09:17:51.608605248Z l
Oct 05 09:17:51 ip-172-31-75-217.ec2.internal grafana[2840]: logger=grafanaStorageLogger t=2023-10-05T09:17:5
Oct 05 09:17:51 ip-172-31-75-217.ec2.internal systemd[1]: Started Grafana instance.
```

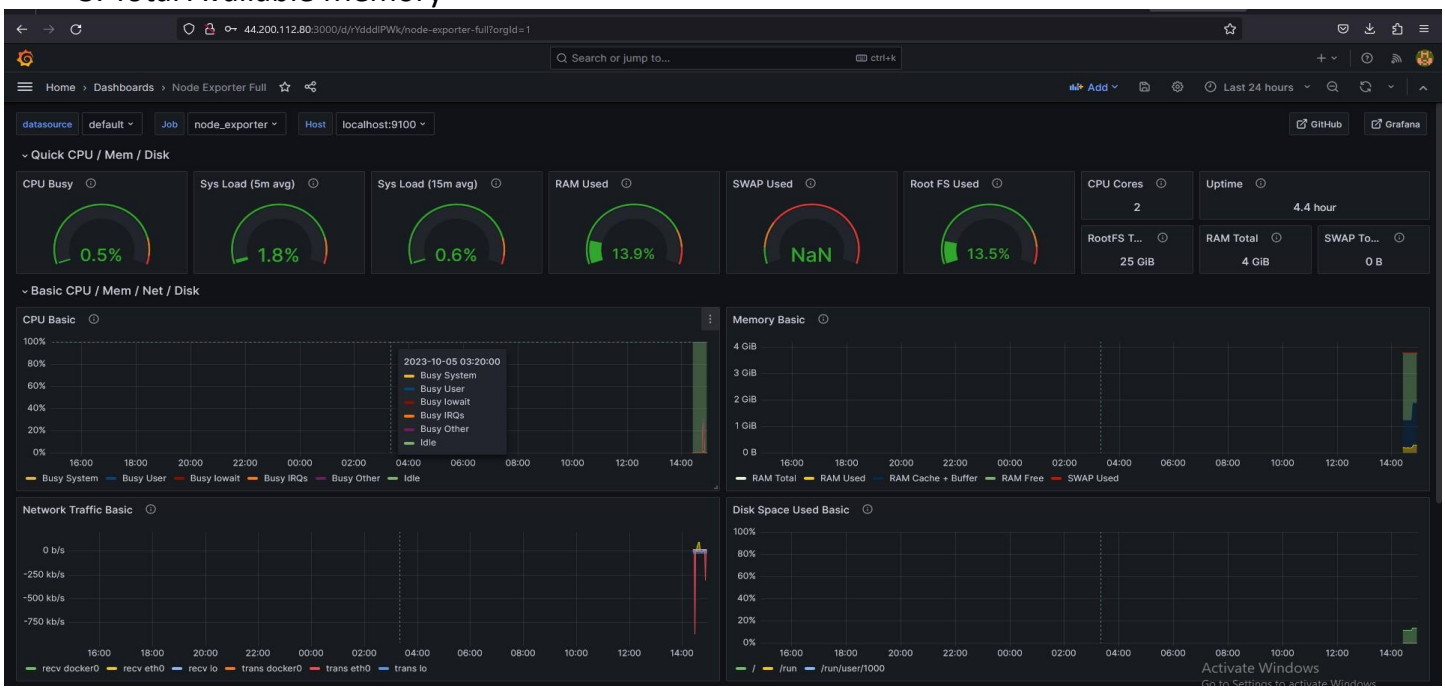



44.203.178.158:3000/?orgid=1
Search or jump to...
Home
Welcome to Grafana
Need help? Documentation Tutorials Community Public Slack
Remove this panel
Basic
The steps below will guide you to quickly finish setting up your Grafana installation.
TUTORIAL
DATA SOURCE AND DASHBOARDS
Grafana fundamentals
Set up and understand Grafana if you have no prior experience. This tutorial guides you through the entire process and covers the "Data source" and "Dashboards" steps to the right.
DATA SOURCES
Add your first data source
Learn how in the docs
DASHBOARDS
Create your first dashboard
Learn how in the docs
Dashboards
Starred dashboards
Recently viewed dashboards
Latest from the blog
Oct 04
ObservabilityCON 2023 agenda: full stack observability, real user monitoring, Kubernetes monitoring, and more
Community voices are a cornerstone of any Grafana Labs event. That's why we're excited to reveal the full agenda for ObservabilityCON 2023, which features community-led sessions that highlight the unique observability journeys and success stories from some leading experts in the industry. Register now for ObservabilityCON! This year's line-up for ObservabilityCON 2023 — our premier open source observability event that will take place 14-15 November in London — includes so much to look forward to, from lessons learned during a Grafana Cloud migration to successful strategies for supporting complex hybrid and multi-cloud environments with the Grafana LGTM Stack (Loki for logs, Grafana for visualization, Tempo for traces, and Mimir for metrics).
ObservabilityCON
2023 agenda
14 - 15 NOV, 2023
Activate Windows
Go to Settings to activate Windows.
Oct 03
How continuous profiling improved code performance for a new Grafana Loki feature


Search or jump to...
Home › Connections › Data sources › Prometheus
Connections
Add new connection
Data sources
Default editor
Builder
Disable metrics lookup
Performance
Prometheus type
Choose
Cache level
Low
Incremental querying (beta)
Disable recording rules (beta)
Other
Custom query parameters
Example: max_source_resolution=5m&timeout
HTTP method
POST
Exemplars
+ Add
Successfully queried the Prometheus API.
Next, you can start to visualize data by building a dashboard, or by querying data in the Explore view.
Activate Windows
Go to Settings to activate Windows.
Delete
Save & test

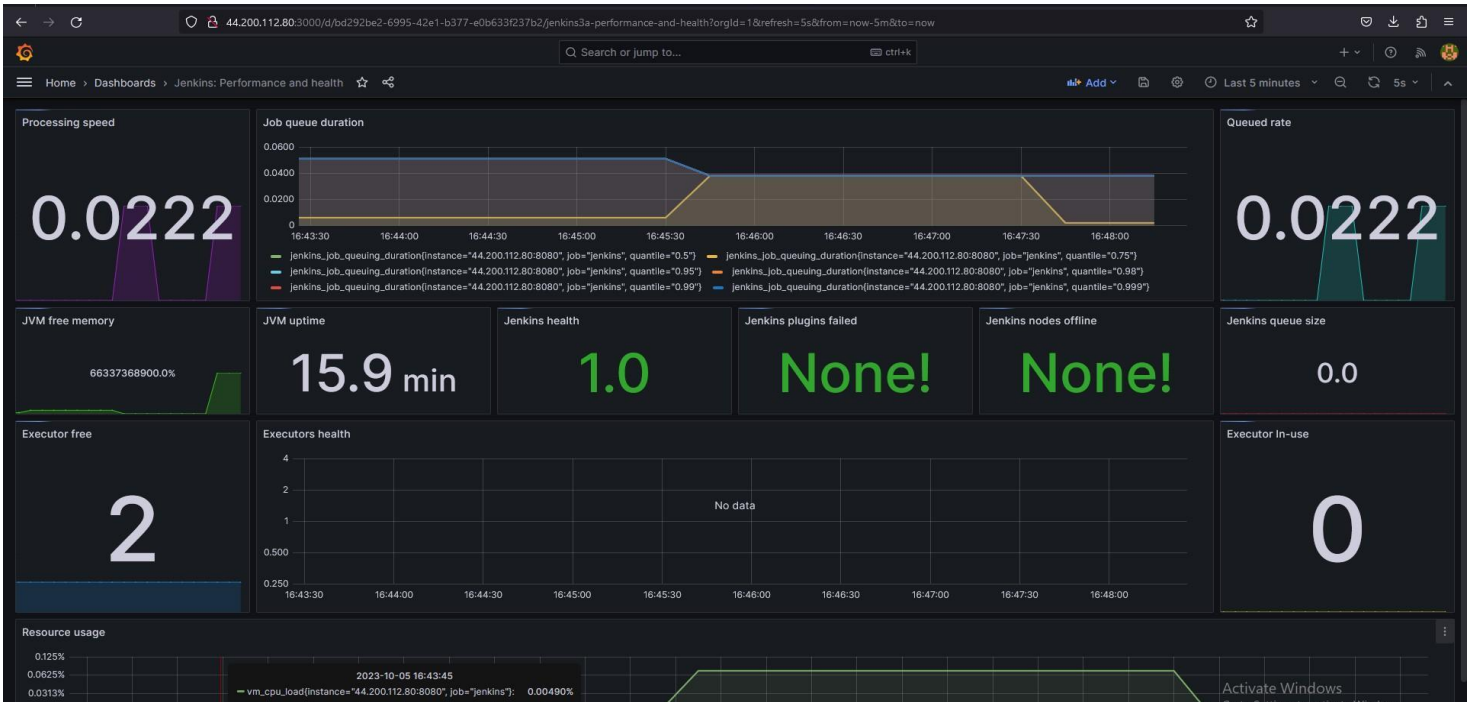
The below monitoring is done on the Node-Exporter (Id=1860)

1. CPU utilization
2. Disk Space Utilization
3. Total Available Memory



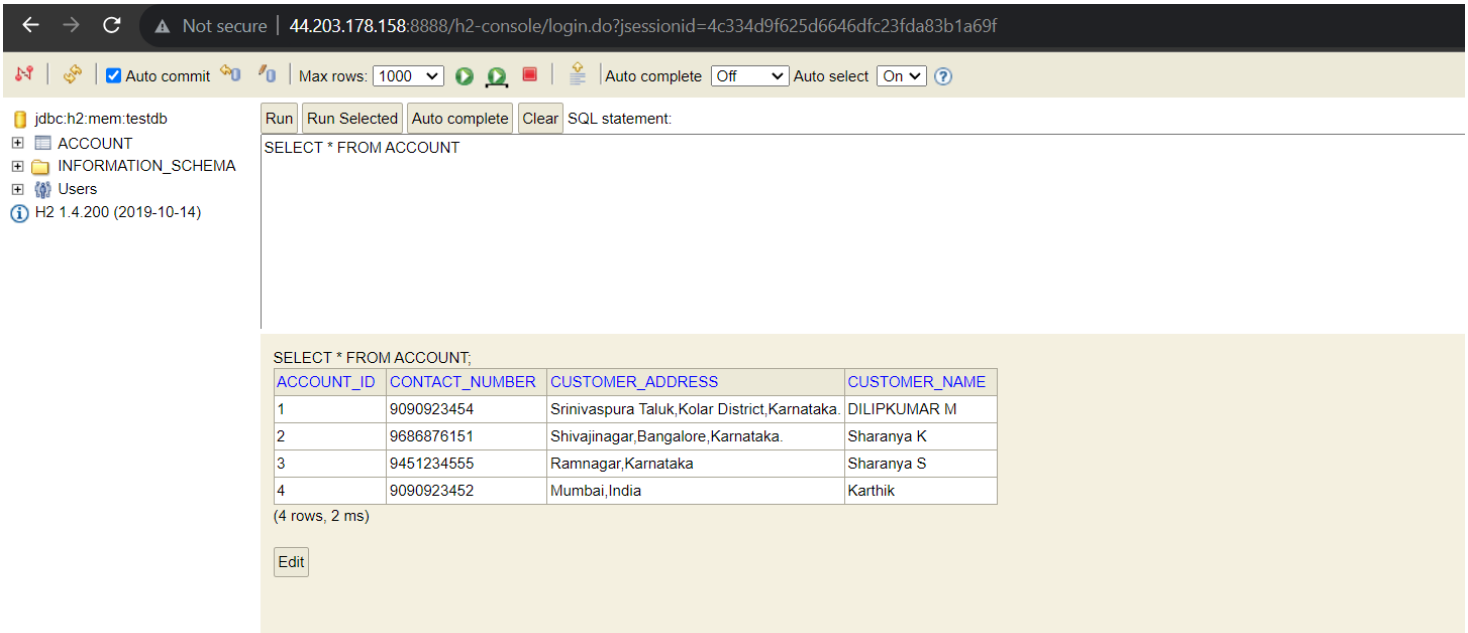
The below monitoring is done for the Jenkins-server (Id=306) which is having,

1. CPU utilization 2.Disk space Utilization 3.Total available Memory



At last the all database of customers that operated from the bank will be stored in the h2-console.

http://44.203.178.158:8888/h2-console/login.do?jsessionid=4c334d9f625d6646dfc23fda83b1a69f



Submitted by

DILIP KUMAR M
DevOps certification batch

To

StarAgile Institutes
Bangalore.

Thank You