Project - Finance Me
Banking and Finance Domain

FinanceMe Project Overview

Project Information

- FinanceMe is a German banking and financial services provider.
- It offers services like banking, fund management, loans, debit/credit cards, and investment banking.
- Moving from a monolithic architecture to a microservices-based system using DevOps and AWS.

Problem Statement

- Struggles with scalability and management of a monolithic application.
- Wants automated, frequent, and reliable software updates.

Problems Faced

- ❌ Complex Monolithic Application – Hard to maintain and update 🏗️
- 🛠️ Manual Testing of Different Components – Slows down the release process 🐌
- 🔄 Difficulties in Managing Incremental Builds – Inefficient and error-prone ⚠️
- 📏 Unable to Scale Individual Modules Independently – Wastes resources 🚧
- ⏳ Time-Consuming Manual Infrastructure Setup – Slows down deployments 🏗️
- 👀 Continuous Monitoring is Challenging – Hard to detect and resolve issues ⚡

Solution

- Microservices using Spring Boot and AWS RDS (MySQL).
- DevOps automation with CI/CD pipelines.
- Infrastructure automation for deployment.
- Automated testing and monitoring.

Tools Used

- 🛠️ Git – Version control 📂
- 🚀 Maven – Continuous build ⚙️
- 🤖 Jenkins – CI/CD automation 🔄
- 🐳 Docker – Containerization 📦
- 🔧 Ansible – Configuration management ⚡
- ☸️ Kubernetes – Deployment of Pods 🚢
- 🏗️ Terraform – Infrastructure automation 🌍
- 📊 Prometheus & Grafana – Monitoring and visualization 📈

Step-by-Step Process:

Version Control with GitHub 🗂️

- All source code is stored in a GitHub repository.
- Any changes to the code trigger the CI/CD pipeline.

Building and Testing with Jenkins 🔧

- Jenkins fetches the latest code from GitHub.
- The code is built using Maven.
- Unit tests and integration tests are executed automatically. ✅

Containerization with Docker 🐳

- A Docker image of the application is created.
- The image is stored in Docker Hub for easy deployment.

Infrastructure Provisioning with Terraform 🏗️

- Terraform scripts are used to create EC2 instances on AWS.
- The necessary computing environment is set up automatically.

Configuration Management with Ansible ⚙️

- Ansible configures the EC2 instances.
- Required software and dependencies are installed automatically.

Deployment with Kubernetes ☸️

- Kubernetes pulls the Docker image from Docker Hub.
- The application is deployed as multiple pods.
- If a pod fails, Kubernetes ensures high availability by running other instances. 🔄

Monitoring with Prometheus and Grafana 📊

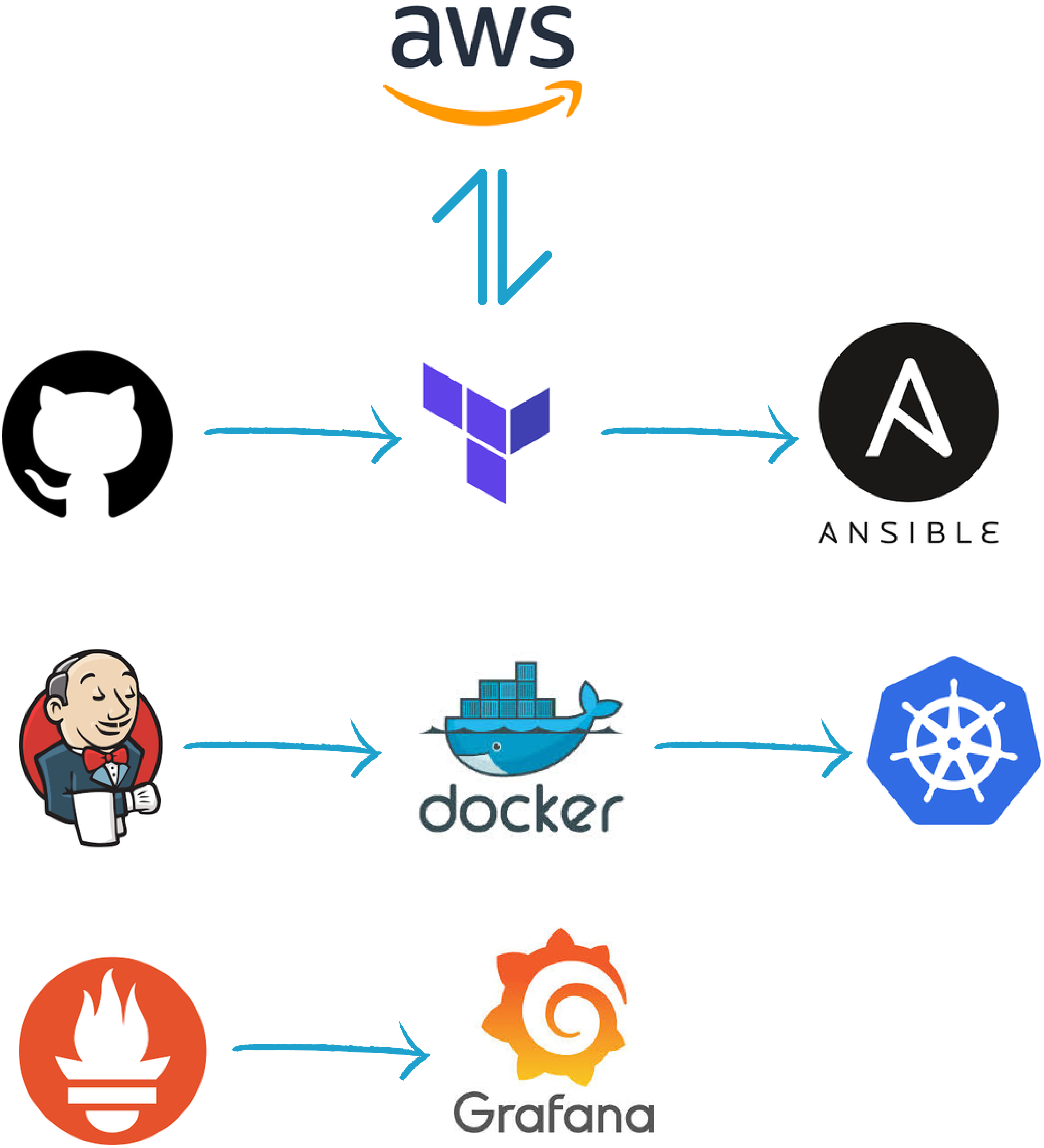
- Prometheus collects system performance metrics.
- Grafana visualizes key metrics like CPU usage, memory usage, and disk space.
- Ensures real-time monitoring of application and infrastructure health. 🔍

Outcome:

- Automated CI/CD pipeline reduces manual effort and accelerates deployment. 🚀
- Scalable and resilient application infrastructure. 🔄
- Continuous monitoring ensures system reliability and performance. ✅
- This streamlined DevOps workflow enhances efficiency and ensures high-quality software delivery. 🎯

Visual Representation of WorkFlow

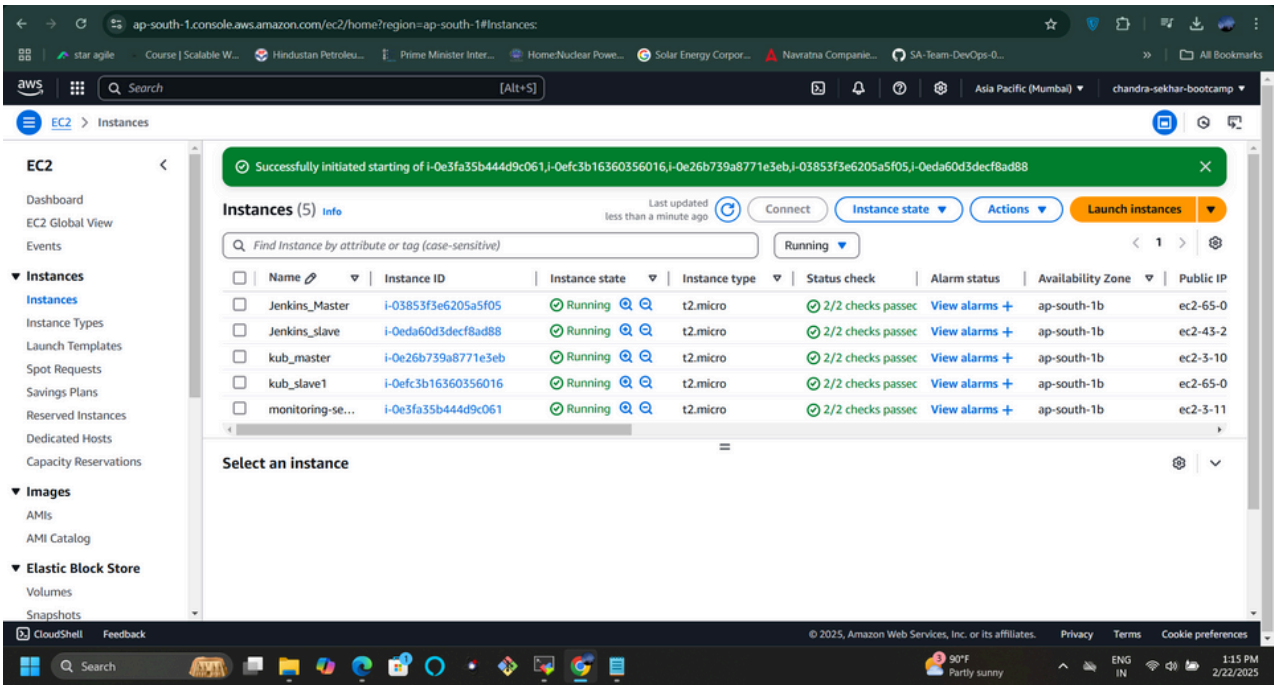
Below is the image that shows on how each tool is connected and works together



Launching the instances

Since the source code repository is already provided, my first step is to launch EC2 instances in AWS.

- I will create two EC2 instances for Jenkins—one for the master and one for the slave.
- I will set up two EC2 instances for Kubernetes—one as the master and one as the slave. I will also install Docker on the same instance as the Kubernetes setup.
- I will launch one EC2 instance to install Prometheus and Grafana for monitoring.



Successfully initiated starting of i-0e3fa35b444d9c061,i-0efc3b16360356016,i-0e26b739a8771e3eb,i-03853f3e6205a5f05,i-0eda60d3decf8ad88

Instances (5)

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IP
Jenkins_Master	i-03853f3e6205a5f05	Running	t2.micro	2/2 checks passec	View alarms +	ap-south-1b	ec2-65-0
Jenkins_slave	i-0eda60d3decf8ad88	Running	t2.micro	2/2 checks passec	View alarms +	ap-south-1b	ec2-43-2
kub_master	i-0e26b739a8771e3eb	Running	t2.micro	2/2 checks passec	View alarms +	ap-south-1b	ec2-3-10
kub_slave1	i-0efc3b16360356016	Running	t2.micro	2/2 checks passec	View alarms +	ap-south-1b	ec2-65-0
monitoring-se...	i-0e3fa35b444d9c061	Running	t2.micro	2/2 checks passec	View alarms +	ap-south-1b	ec2-3-11

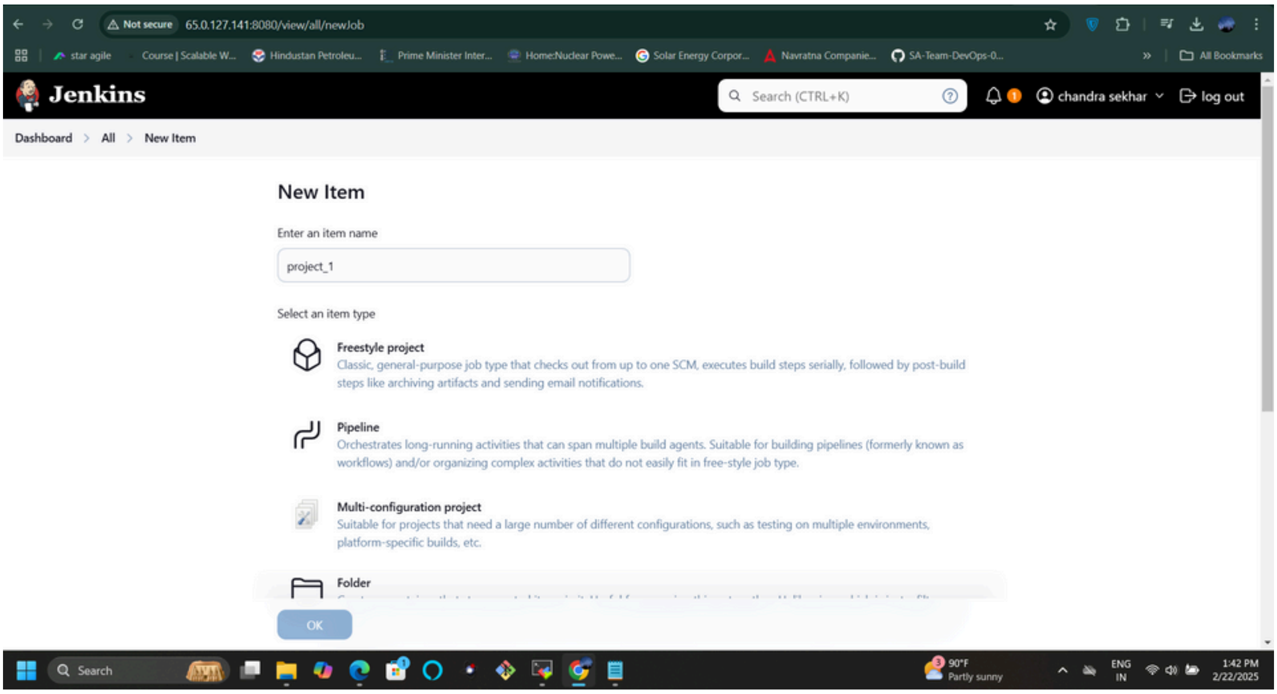
Select an instance

Since my EC2 instances are already set up with the required services, my task is simple.

- I will pull the source code from the given repository.
- I will write a pipeline script in the Jenkins module to define the automation steps.
- I will automate the deployment process so that every code change triggers a smooth and efficient build, test, and deployment workflow.
- With everything already configured, my focus is on setting up and running the automation in Jenkins.

- **Connecting to the Jenkins Server and Creating a Project Pipeline project (Project_01)**

Source code URL - https://github.com/StarAgileDevOpsTraining/star-agile-banking-finance.git



New Item

Enter an item name

project_1

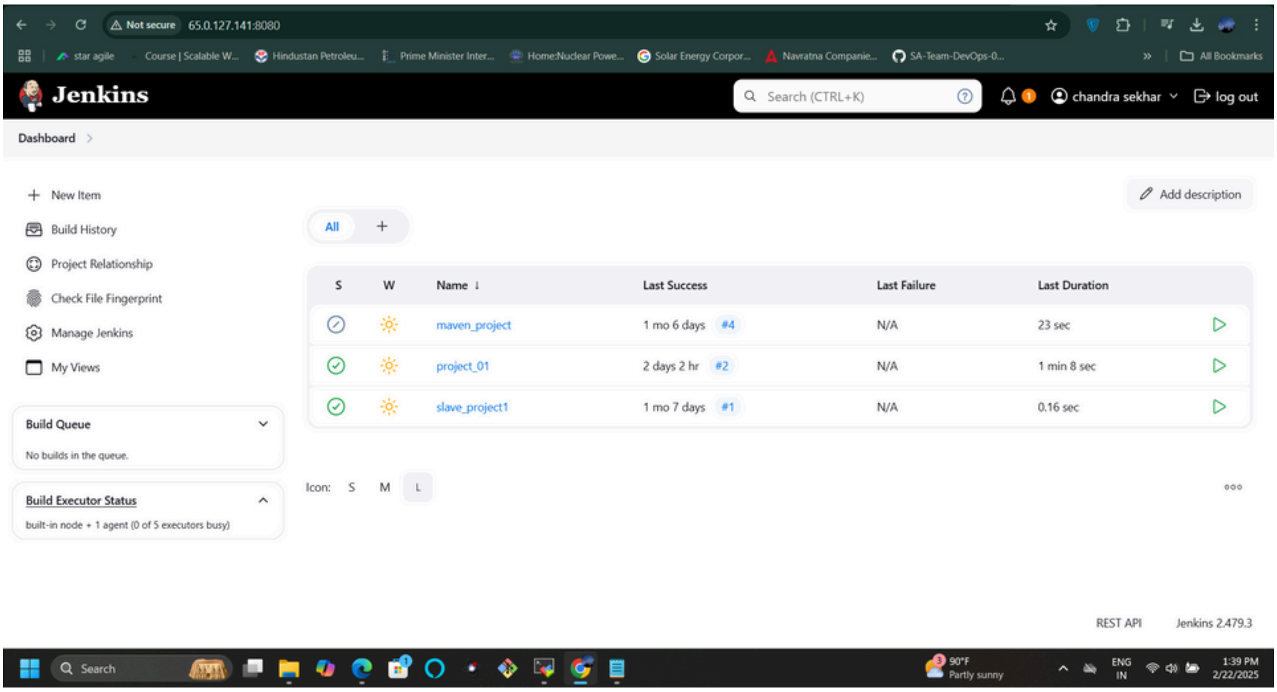
Select an item type

Freestyle project
Classic, general-purpose job type that checks out from up to one SCM, executes build steps serially, followed by post-build steps like archiving artifacts and sending email notifications.

Pipeline
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.

Multi-configuration project
Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.

Folder



S	W	Name ↓	Last Success	Last Failure	Last Duration
⊘	☀	maven_project	1 mo 6 days #4	N/A	23 sec
✓	☀	project_01	2 days 2 hr #2	N/A	1 min 8 sec
✓	☀	slave_project1	1 mo 7 days #1	N/A	0.16 sec

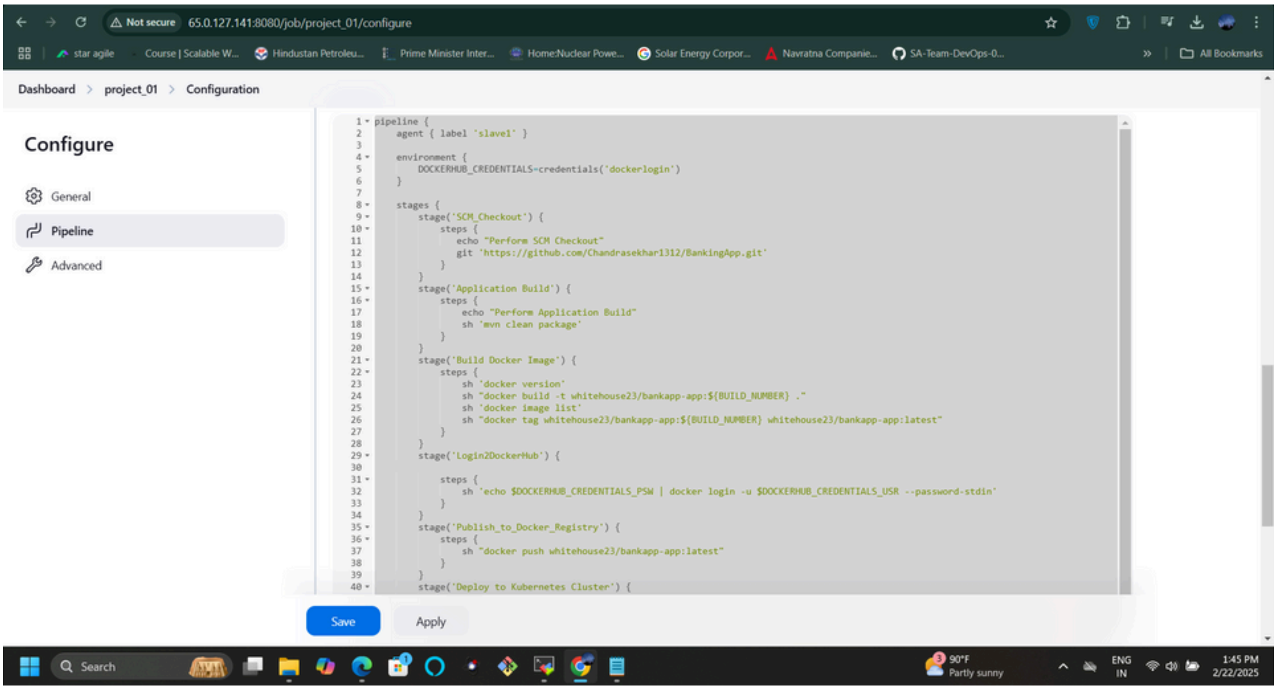
Build Queue

No builds in the queue.

Build Executor Status

built-in node + 1 agent (0 of 5 executors busy)

- **Now Configuring the Pipeline and Writing the Pipeline Script.**

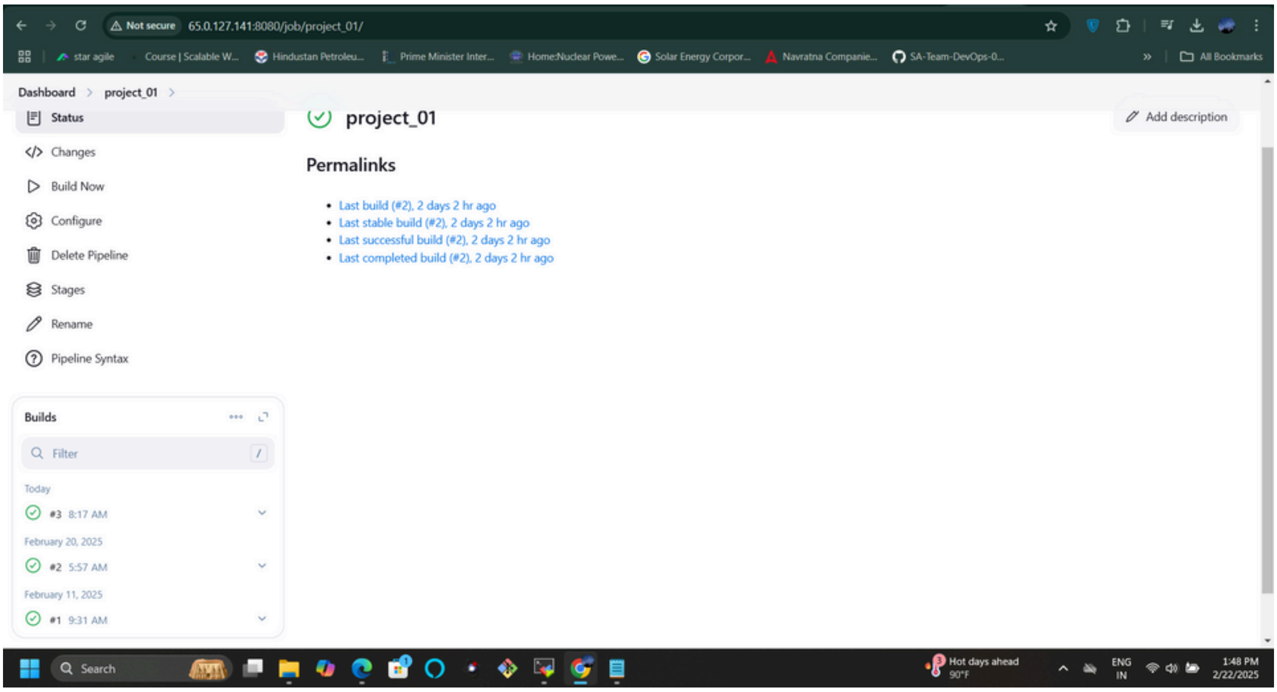


Dashboard > project_01 > Configuration

Configure

```
pipeline {
    agent { label 'slave1' }

    environment {
        DOCKERHUB_CREDENTIALS=credentials('dockerlogin')
    }

    stages {
        stage('SCM_Checkout') {
            steps {
               echo "Perform SCM Checkout"
               git 'https://github.com/Chandrasekhar1312/BankingApp.git'
            }
        }
        stage('Application Build') {
            steps {
                echo "Perform Application Build"
                sh 'mvn clean package'
            }
        }
        stage('Build Docker Image') {
            steps {
                sh 'docker version'
                sh "docker build -t whitehouse23/bankapp-app:${BUILD_NUMBER} ."
                sh 'docker image list'
                sh "docker tag whitehouse23/bankapp-app:${BUILD_NUMBER} whitehouse23/bankapp-app:latest"
            }
        }
        stage('Login2DockerHub') {

            steps {
                sh 'echo $DOCKERHUB_CREDENTIALS_PSW | docker login -u $DOCKERHUB_CREDENTIALS_USR --password-stdin'
            }
        }
        stage('Publish_to_Docker_Registry') {
            steps {
                sh "docker push whitehouse23/bankapp-app:latest"
            }
        }
        stage('Deploy to Kubernetes Cluster') {
```

- **After writing the pipeline script now running the pipeline**



Dashboard > project_01 >

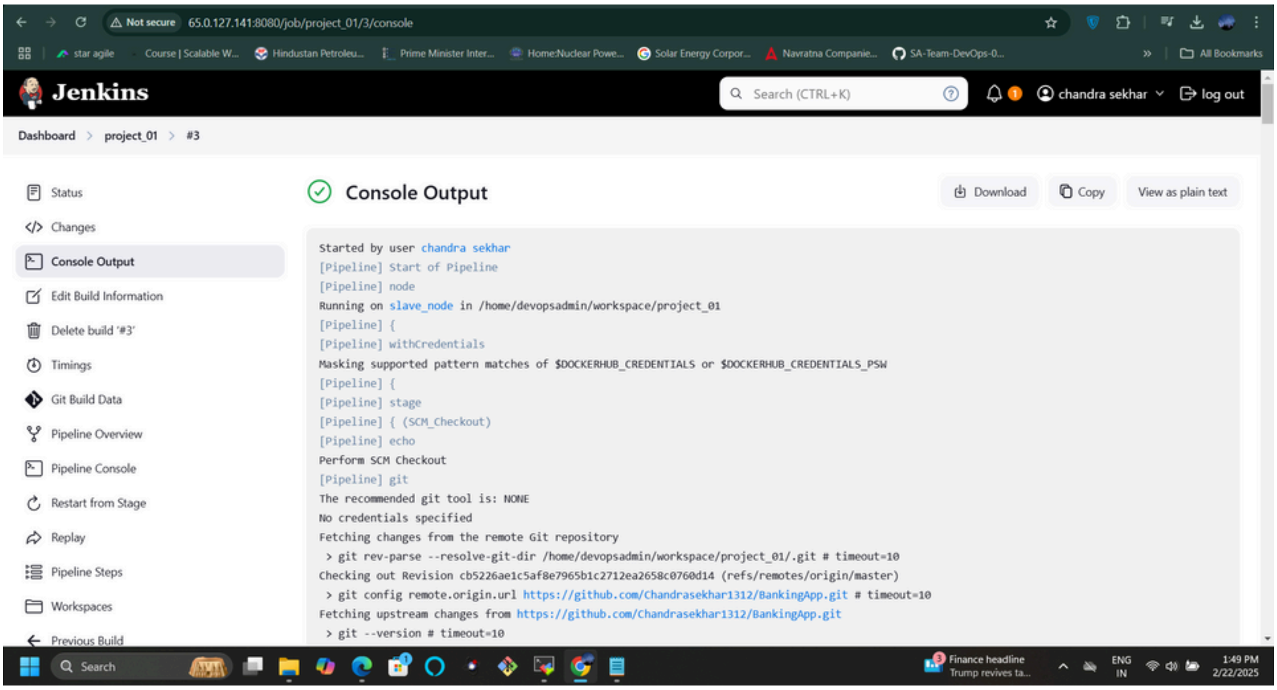
project_01

Permalinks

- Last build (#2), 2 days 2 hr ago
- Last stable build (#2), 2 days 2 hr ago
- Last successful build (#2), 2 days 2 hr ago
- Last completed build (#2), 2 days 2 hr ago

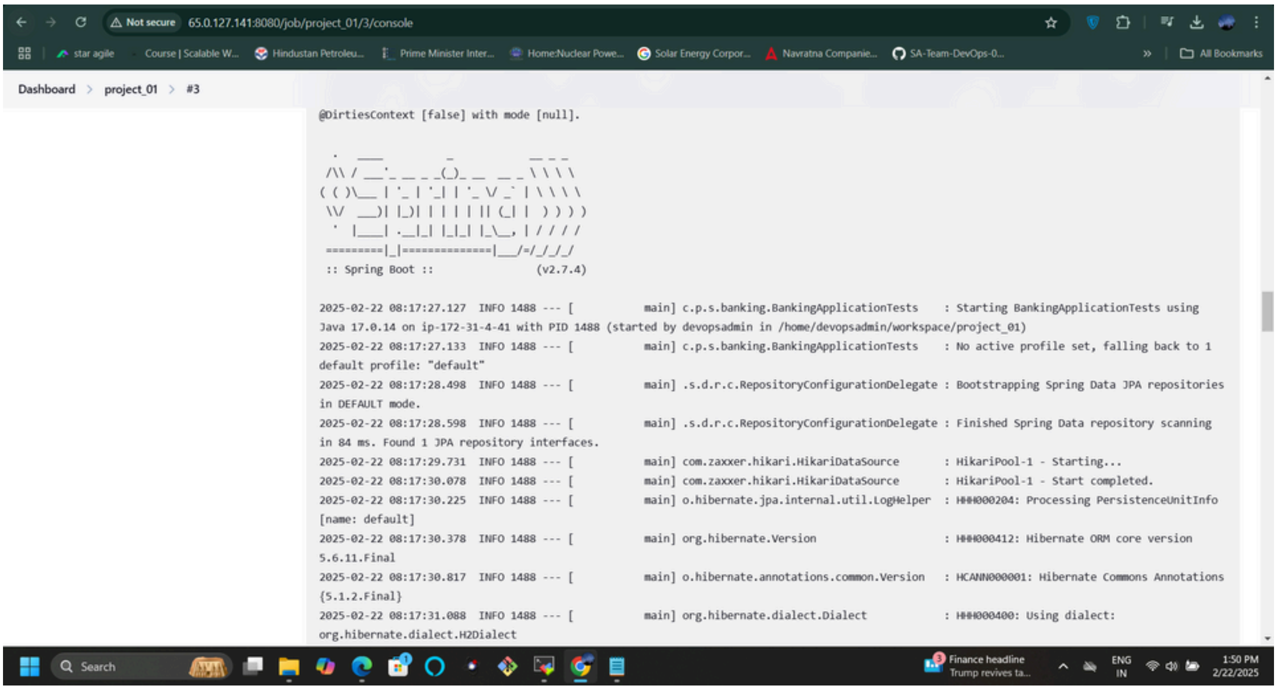
Builds

Today
✓ #3 8:17 AM
February 20, 2025
✓ #2 5:57 AM
February 11, 2025
✓ #1 9:31 AM

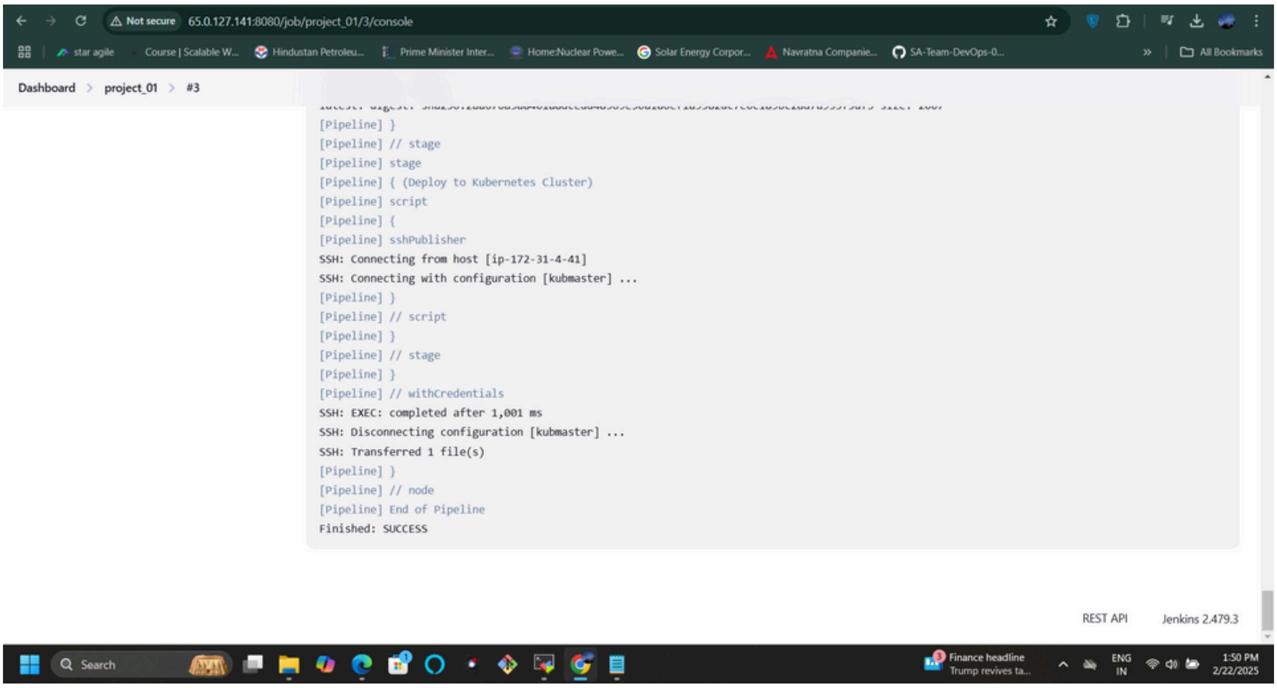


Console Output

```
Started by user chandra sekhar
[Pipeline] Start of Pipeline
[Pipeline] node
Running on slave_node in /home/devopsadmin/workspace/project_01
[Pipeline] {
[Pipeline] withCredentials
Masking supported pattern matches of $DOCKERHUB_CREDENTIALS or $DOCKERHUB_CREDENTIALS_PSW
[Pipeline] {
[Pipeline] stage
[Pipeline] { (SCM_Checkout)
[Pipeline] echo
Perform SCM Checkout
[Pipeline] git
The recommended git tool is: NONE
No credentials specified
Fetching changes from the remote Git repository
 > git rev-parse --resolve-git-dir /home/devopsadmin/workspace/project_01/.git # timeout=10
Checking out Revision cb5226ae1c5af8e7965b1c2712ea2658c0760d14 (refs/remotes/origin/master)
 > git config remote.origin.url https://github.com/Chandrasekhar1312/BankingApp.git # timeout=10
Fetching upstream changes from https://github.com/Chandrasekhar1312/BankingApp.git
 > git --version # timeout=10
```

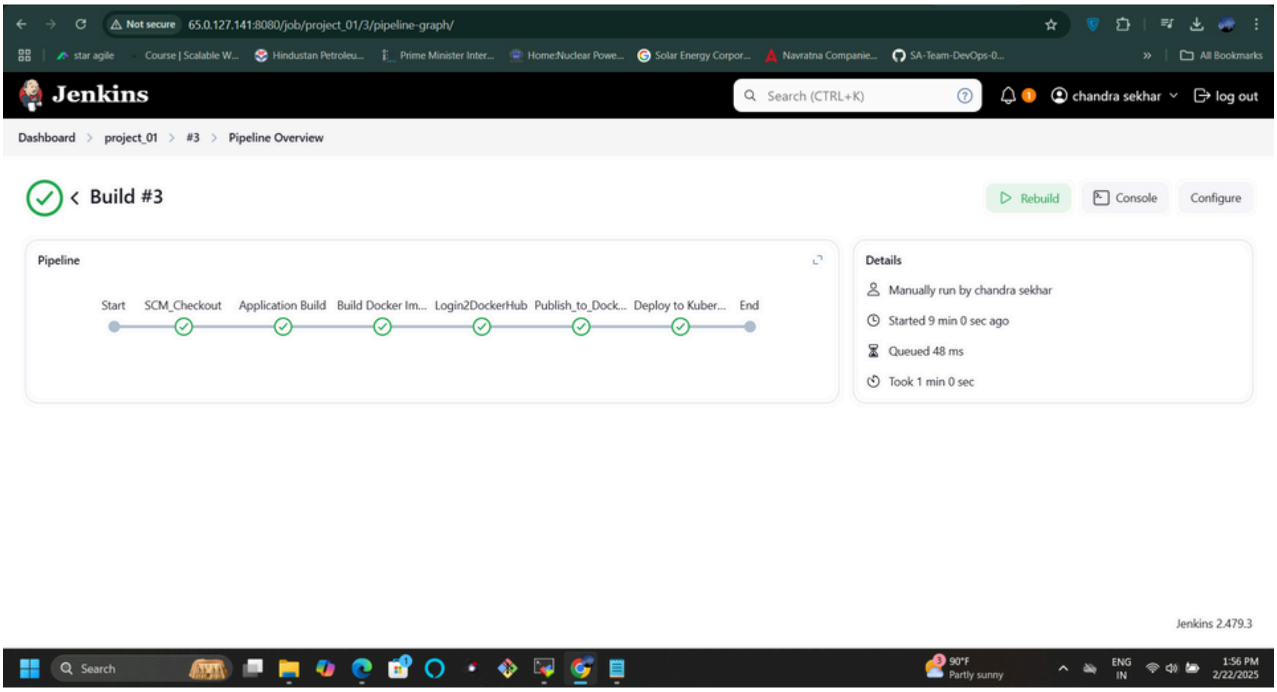


```
@DirtiesContext [false] with mode [null].

  .   ____          _            __ _ _
 /\\ / ___'_ __ _ _(_)_ __  __ _ \ \ \ \
( ( )\___ | '_ | '_| | '_ \/ _` | \ \ \ \
 \\/  ___)| |_)| | | | | || (_| |  ) ) ) )
  '  |____| .__|_| |_|_| |_\__, | / / / /
 =========|_|==============|___/=/_/_/_/
 :: Spring Boot ::                (v2.7.4)

2025-02-22 08:17:27.127  INFO 1488 --- [           main] c.p.s.banking.BankingApplicationTests    : Starting BankingApplicationTests using
Java 17.0.14 on ip-172-31-4-41 with PID 1488 (started by devopsadmin in /home/devopsadmin/workspace/project_01)
2025-02-22 08:17:27.133  INFO 1488 --- [           main] c.p.s.banking.BankingApplicationTests    : No active profile set, falling back to 1
default profile: "default"
2025-02-22 08:17:28.498  INFO 1488 --- [           main] .s.d.r.c.RepositoryConfigurationDelegate : Bootstrapping Spring Data JPA repositories
in DEFAULT mode.
2025-02-22 08:17:28.598  INFO 1488 --- [           main] .s.d.r.c.RepositoryConfigurationDelegate : Finished Spring Data repository scanning
in 84 ms. Found 1 JPA repository interfaces.
2025-02-22 08:17:29.731  INFO 1488 --- [           main] com.zaxxer.hikari.HikariDataSource       : HikariPool-1 - Starting...
2025-02-22 08:17:30.078  INFO 1488 --- [           main] com.zaxxer.hikari.HikariDataSource       : HikariPool-1 - Start completed.
2025-02-22 08:17:30.225  INFO 1488 --- [           main] o.hibernate.jpa.internal.util.LogHelper  : HHH000204: Processing PersistenceUnitInfo
[name: default]
2025-02-22 08:17:30.378  INFO 1488 --- [           main] org.hibernate.Version                    : HHH000412: Hibernate ORM core version
5.6.11.Final
2025-02-22 08:17:30.817  INFO 1488 --- [           main] o.hibernate.annotations.common.Version   : HCANN000001: Hibernate Commons Annotations
{5.1.2.Final}
2025-02-22 08:17:31.088  INFO 1488 --- [           main] org.hibernate.dialect.Dialect            : HHH000400: Using dialect:
org.hibernate.dialect.H2Dialect
```



```
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Deploy to Kubernetes Cluster)
[Pipeline] script
[Pipeline] {
[Pipeline] sshPublisher
SSH: Connecting from host [ip-172-31-4-41]
SSH: Connecting with configuration [kubmaster] ...
[Pipeline] }
[Pipeline] // script
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // withCredentials
SSH: EXEC: completed after 1,001 ms
SSH: Disconnecting configuration [kubmaster] ...
SSH: Transferred 1 file(s)
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

Pipeline Overview



Dashboard > project_01 > #3 > Pipeline Overview

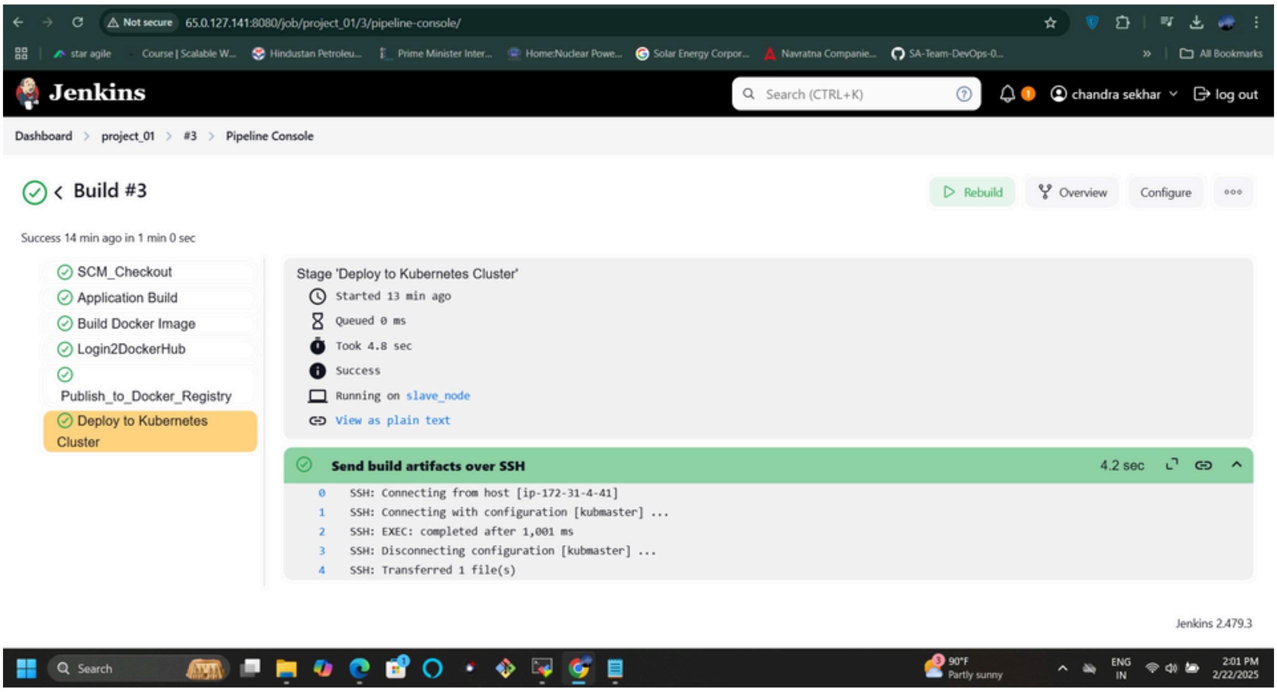
Build #3

Pipeline

Start → SCM_Checkout → Application Build → Build Docker Im... → Login2DockerHub → Publish_to_Dock... → Deploy to Kuber... → End

Details

- Manually run by chandra sekhar
- Started 9 min 0 sec ago
- Queued 48 ms
- Took 1 min 0 sec



Dashboard > project_01 > #3 > Pipeline Console

Build #3

Success 14 min ago in 1 min 0 sec

- ✓ SCM_Checkout
- ✓ Application Build
- ✓ Build Docker Image
- ✓ Login2DockerHub
- ✓ Publish_to_Docker_Registry
- ✓ Deploy to Kubernetes Cluster

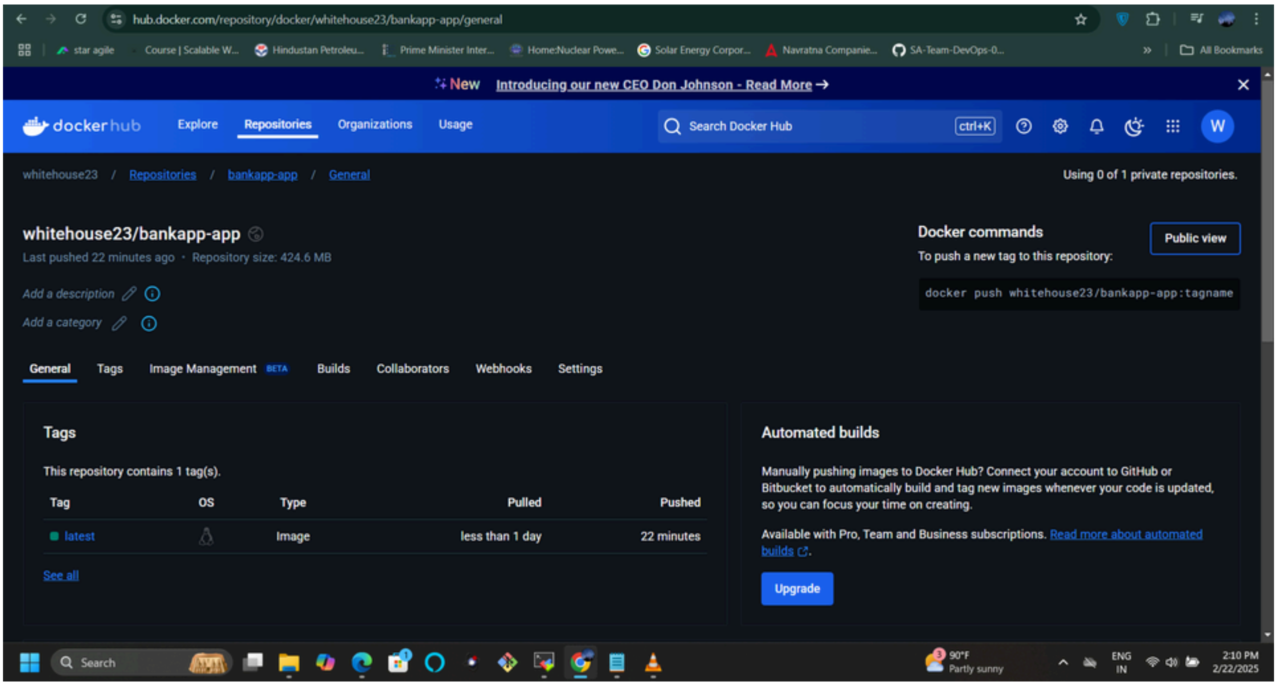
Stage 'Deploy to Kubernetes Cluster'

```
Started 13 min ago
Queued 0 ms
Took 4.8 sec
Success
Running on slave_node
View as plain text
```

✓ Send build artifacts over SSH 4.2 sec

```
SSH: Connecting from host [ip-172-31-4-41]
SSH: Connecting with configuration [kubmaster] ...
SSH: EXEC: completed after 1,001 ms
SSH: Disconnecting configuration [kubmaster] ...
SSH: Transferred 1 file(s)
```

- **we can see the docker image that has been deployed in my docker hub**



hub.docker.com/repository/docker/whitehouse23/bankapp-app/general

New Introducing our new CEO Don Johnson - Read More →

whitehouse23 / Repositories / bankapp-app / General

Using 0 of 1 private repositories.

whitehouse23/bankapp-app

Last pushed 22 minutes ago · Repository size: 424.6 MB

Add a description

Add a category

Docker commands

To push a new tag to this repository:

```
docker push whitehouse23/bankapp-app:tagname
```

Tags

This repository contains 1 tag(s).

Tag	OS	Type	Pulled	Pushed
latest		Image	less than 1 day	22 minutes

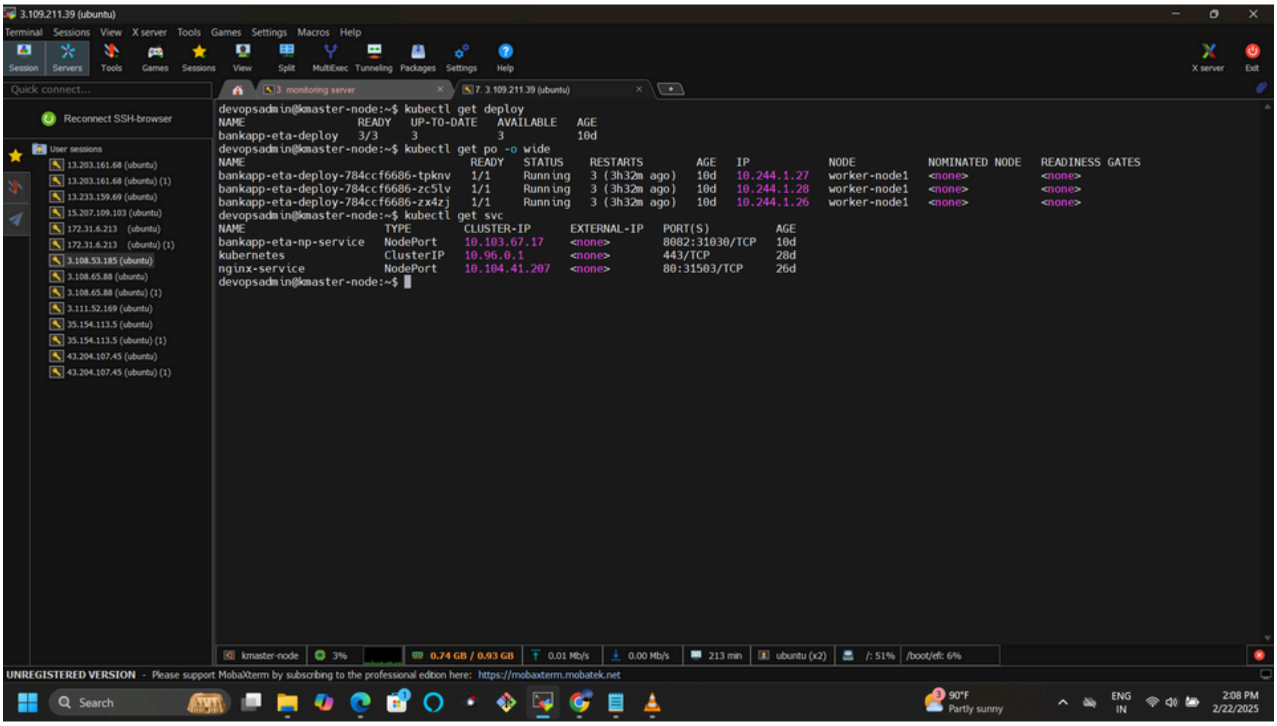
See all

Automated builds

Manually pushing images to Docker Hub? Connect your account to GitHub or Bitbucket to automatically build and tag new images whenever your code is updated, so you can focus your time on creating.

Available with Pro, Team and Business subscriptions. Read more about automated builds.

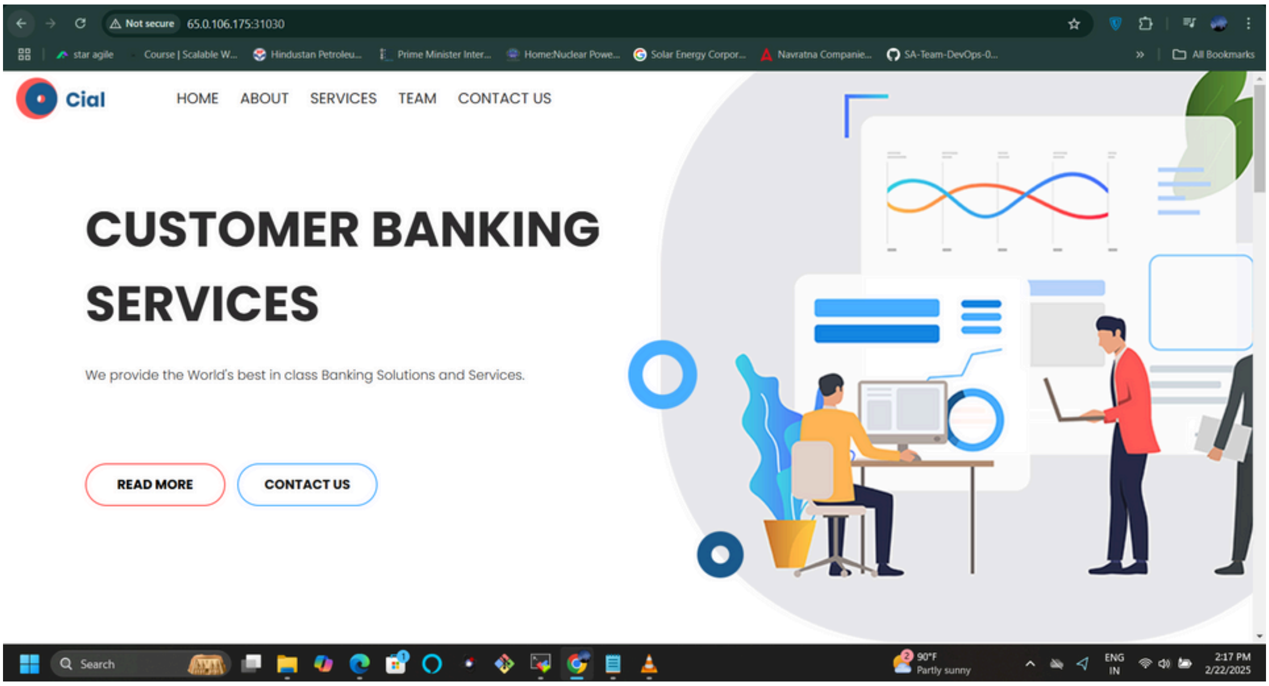
- **now log into the kube master node to check the launched pods since i only had one slace node all the pods will be launched in slave node-1**



```
devopsadmin@kmaster-node:~$ kubectl get deploy
NAME                 READY   UP-TO-DATE   AVAILABLE   AGE
bankapp-eta-deploy   3/3     3            3           10d
devopsadmin@kmaster-node:~$ kubectl get po -o wide
NAME                                  READY   STATUS    RESTARTS       AGE   IP            NODE           NOMINATED NODE   READINESS GATES
bankapp-eta-deploy-784ccf6686-tpknv   1/1     Running   3 (3h32m ago)  10d   10.244.1.27   worker-node1   <none>           <none>
bankapp-eta-deploy-784ccf6686-zc5lv   1/1     Running   3 (3h32m ago)  10d   10.244.1.28   worker-node1   <none>           <none>
bankapp-eta-deploy-784ccf6686-zx4zj   1/1     Running   3 (3h32m ago)  10d   10.244.1.26   worker-node1   <none>           <none>
devopsadmin@kmaster-node:~$ kubectl get svc
NAME                    TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)          AGE
bankapp-eta-np-service  NodePort    10.103.67.17    <none>        8082:31030/TCP   10d
kubernetes              ClusterIP   10.96.0.1       <none>        443/TCP          28d
nginx-service           NodePort    10.104.41.207   <none>        80:31503/TCP     26d
devopsadmin@kmaster-node:~$
```

3.109.211.39 (ubuntu)
Terminal Sessions View X server Tools Games Settings Macros Help
3. monitoring server
7. 3.109.211.39 (ubuntu)
Reconnect SSH-browser
User sessions
kmaster-node
UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: https://mobaxterm.mobatek.net
2:08 PM
2/22/2025

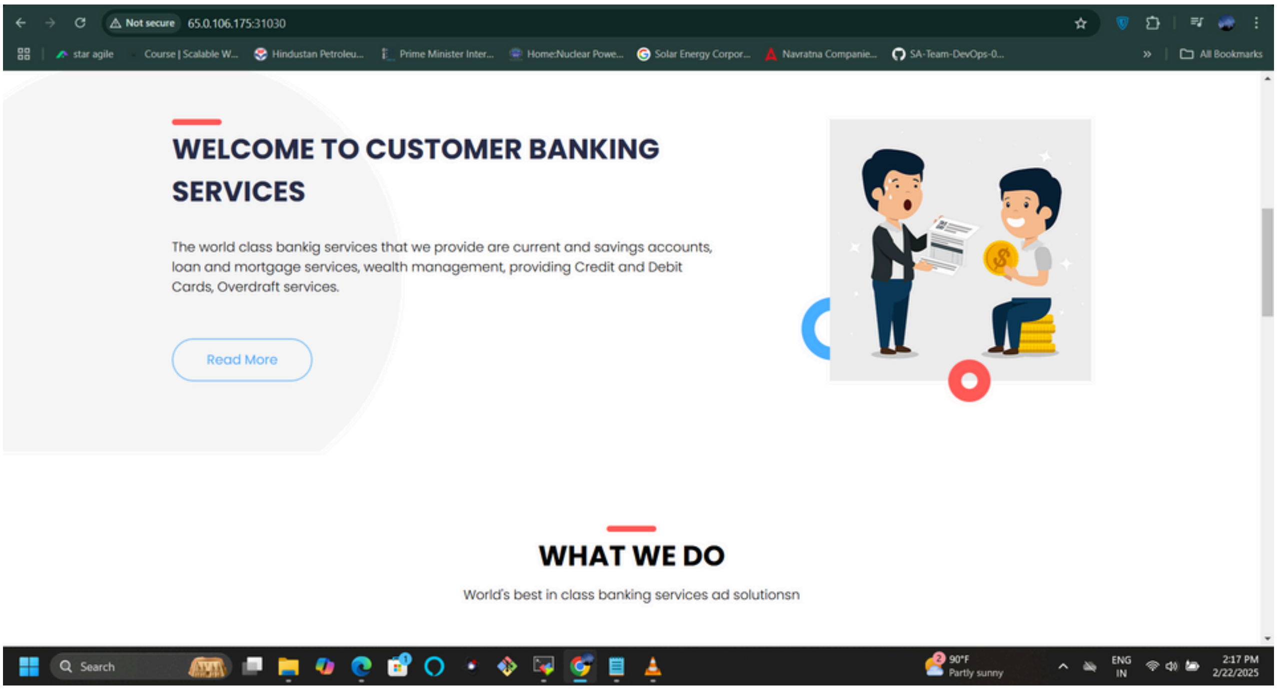
- **In the above we can see 3 pods have been deployed with node port services for accessing through internet. we can connet to them using the public IP address:node port**



Not secure 65.0.106.175:31030

CUSTOMER BANKING SERVICES

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



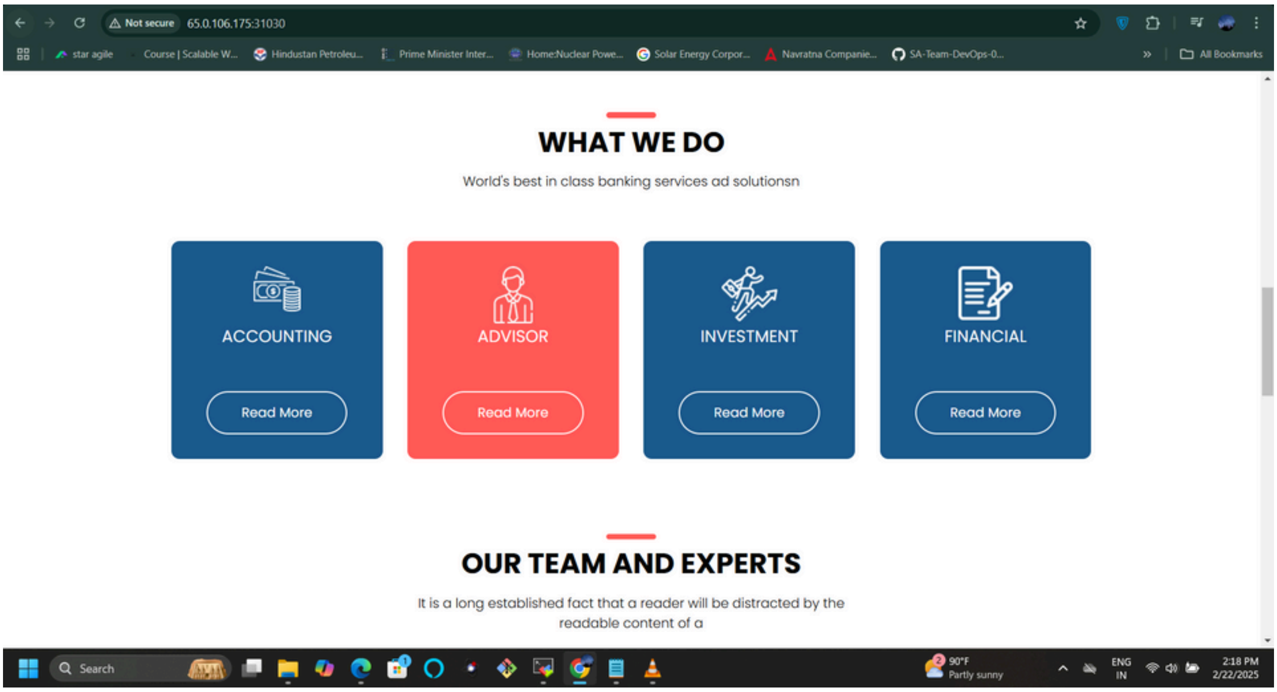
Not secure 65.0.106.175:31030

WELCOME TO CUSTOMER BANKING SERVICES

The world class bankig services that we provide are current and savings accounts, loan and mortgage services, wealth management, providing Credit and Debit Cards, Overdraft services.

WHAT WE DO

World's best in class banking services ad solutionsn



WHAT WE DO

World's best in class banking services ad solutionsn

ACCOUNTING
Read More

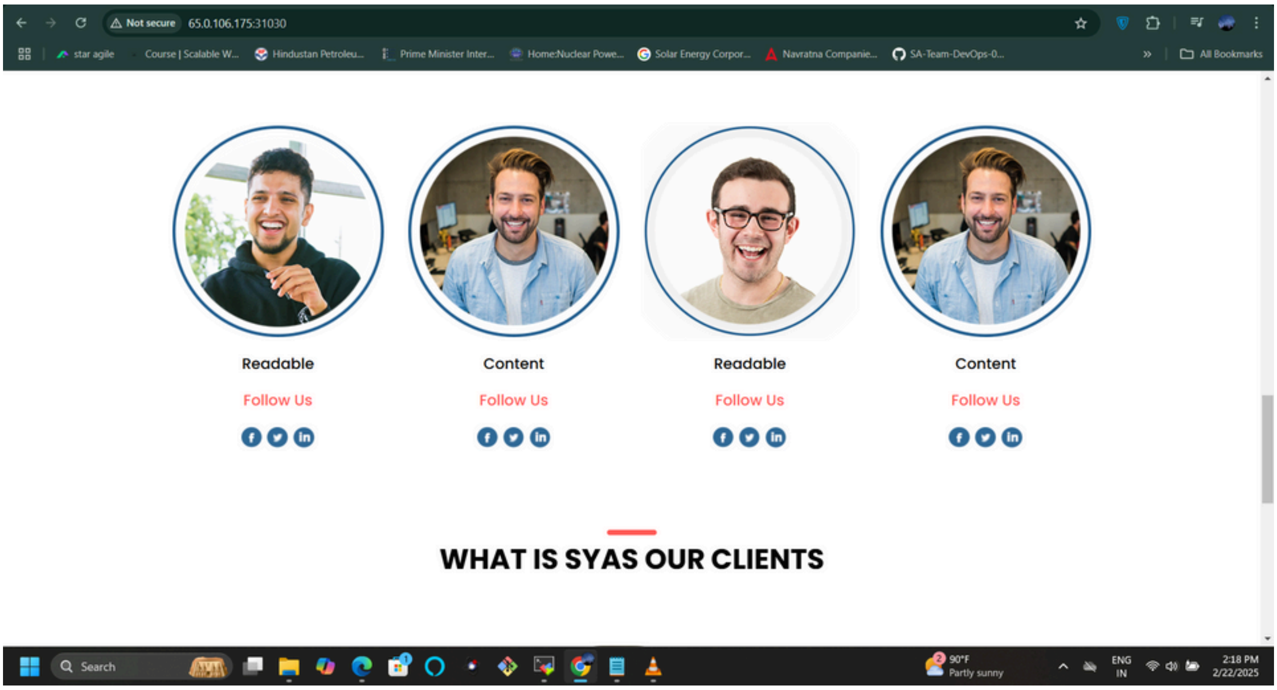
ADVISOR
Read More

INVESTMENT
Read More

FINANCIAL
Read More

OUR TEAM AND EXPERTS

It is a long established fact that a reader will be distracted by the readable content of a



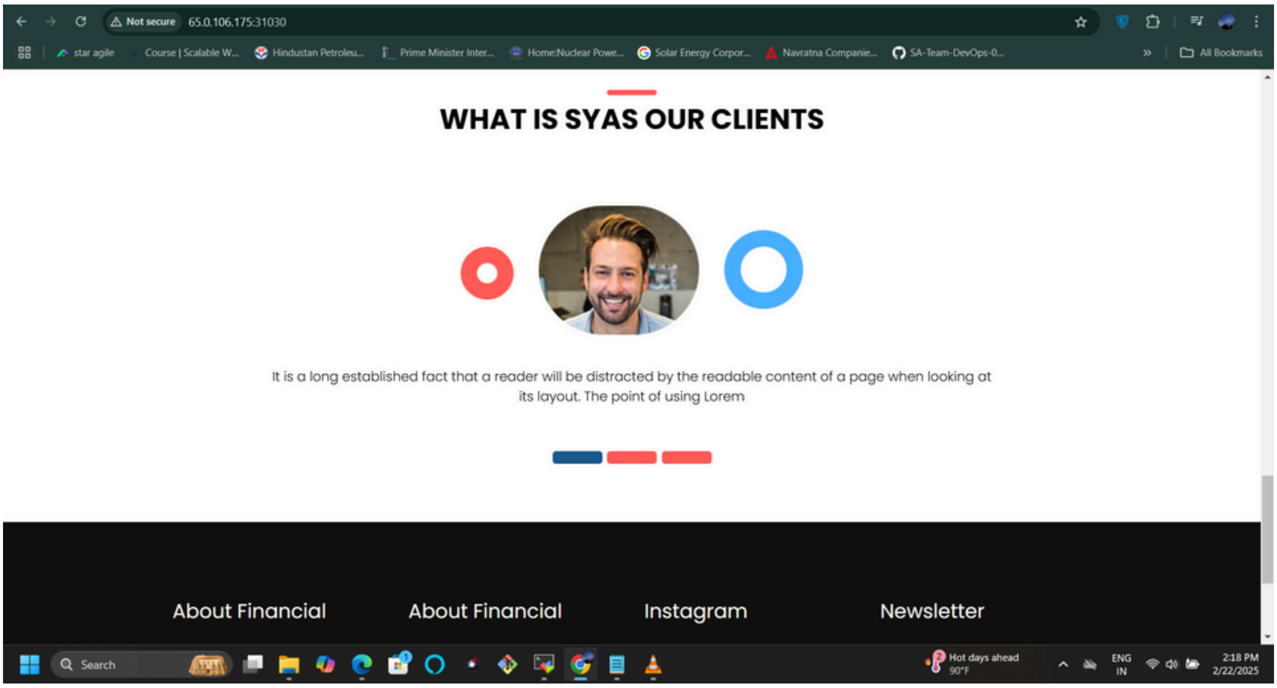
Readable
Follow Us

Content
Follow Us

Readable
Follow Us

Content
Follow Us

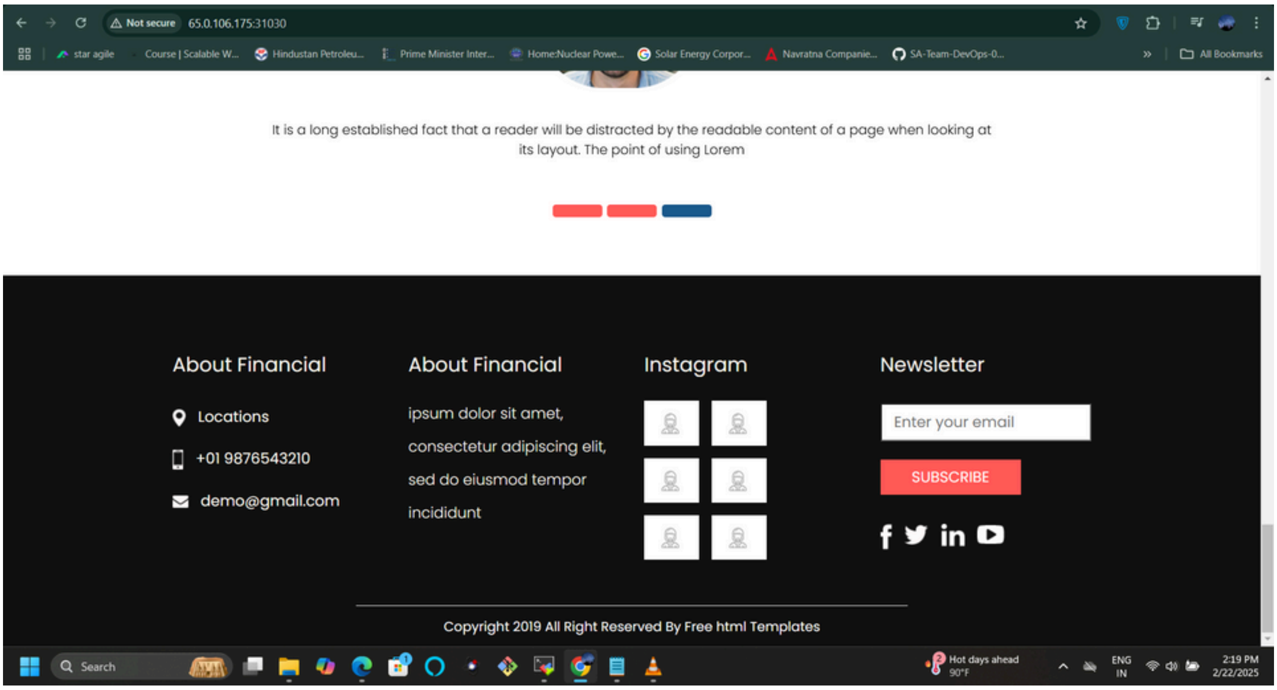
WHAT IS SYAS OUR CLIENTS



WHAT IS SYAS OUR CLIENTS

It is a long established fact that a reader will be distracted by the readable content of a page when looking at its layout. The point of using Lorem

About Financial
About Financial
Instagram
Newsletter



It is a long established fact that a reader will be distracted by the readable content of a page when looking at its layout. The point of using Lorem

About Financial

Locations

+01 9876543210

demo@gmail.com

About Financial

ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt

Instagram

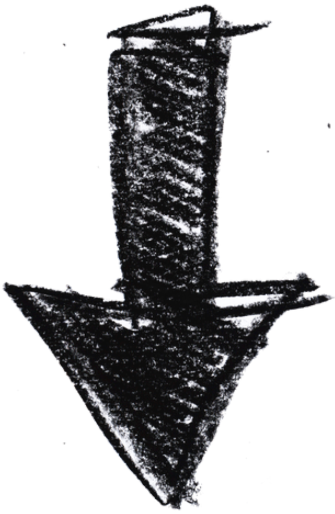
Newsletter

Enter your email

SUBSCRIBE

Copyright 2019 All Right Reserved By Free html Templates

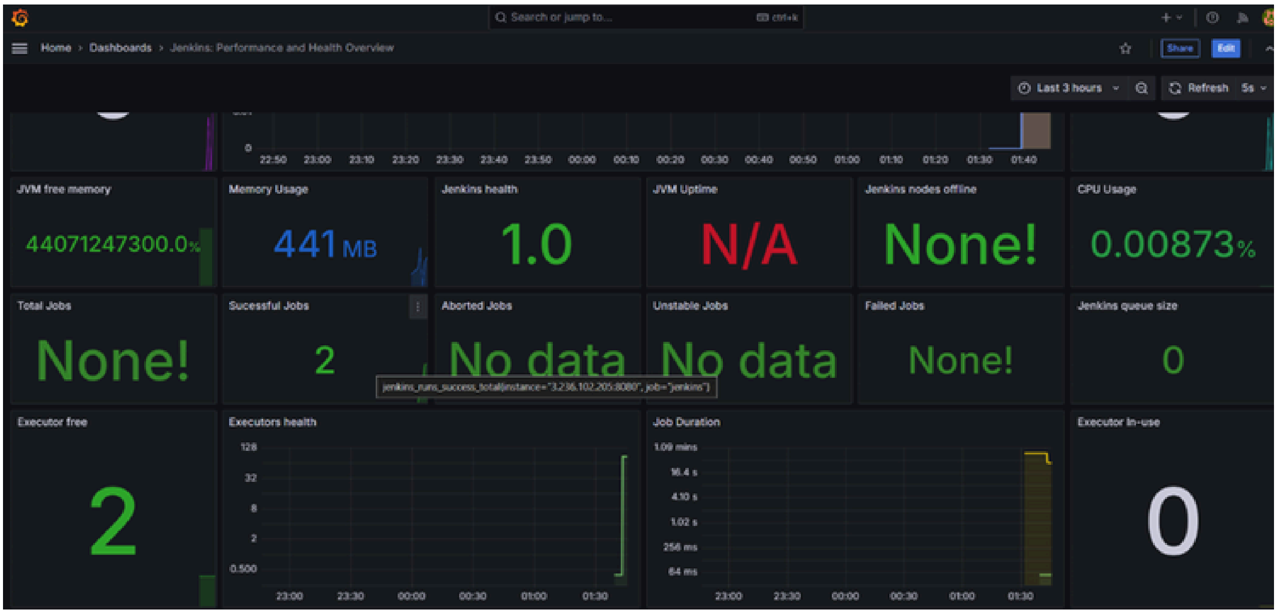
- The bank's application runs smoothly ✅ without any delays or issues, providing a reliable experience 🚀. This is achieved through DevOps automation 🤖.
- This pipeline follows these steps:
- 1️⃣ Get the Source Code – Download the latest code from the repository 📥.
- 2️⃣ Build the Application – Use Maven to compile and package the app 🏗️.
- 3️⃣ Create a Docker Image – Convert the app into a Docker container 🐳.
- 4️⃣ Upload to DockerHub – Store the container in a registry 📦.
- 5️⃣ Deploy to Kubernetes – Use kubectl to launch the app in a cluster ☸️.
- With CI/CD automation 🔄, every code update is built, tested, and deployed automatically ⚡, keeping the bank's application running smoothly with zero downtime 🏦🔥.

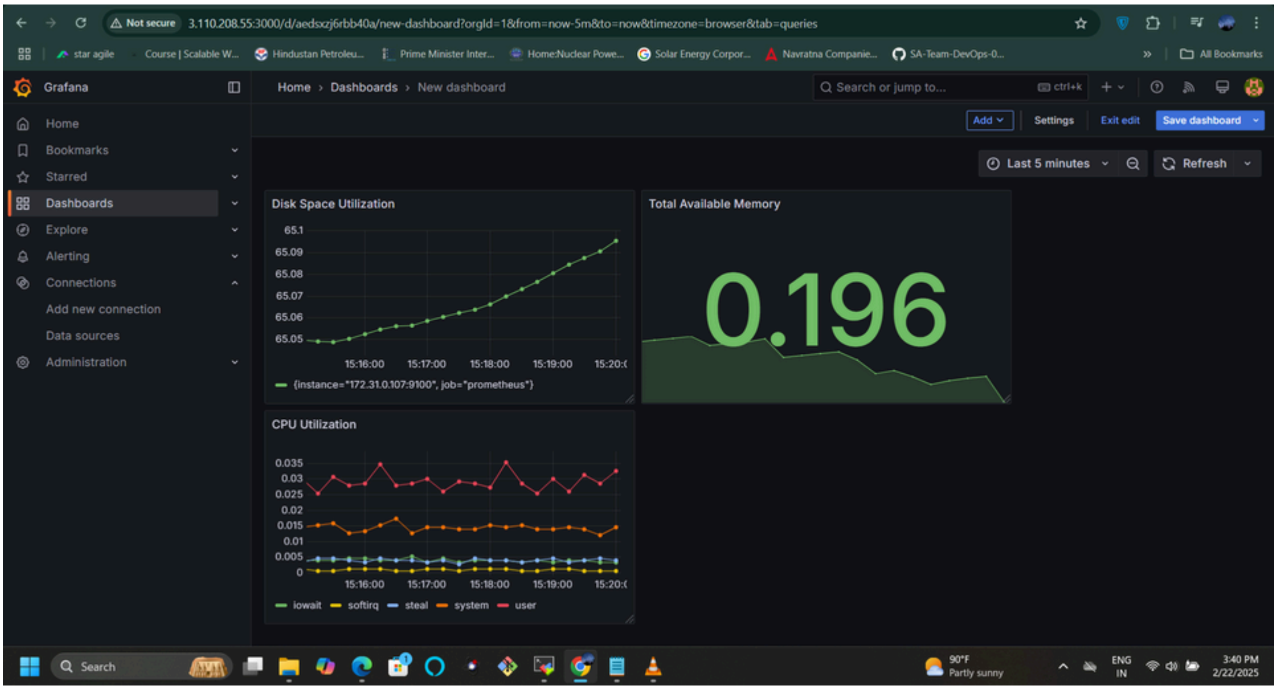


- **This Jenkins pipeline script automates the process of building, containerizing, pushing, and deploying a BankingApp to a Kubernetes cluster using Docker and Maven.**

```
pipeline {
   agent { label 'slave1' }

 environment {
  DOCKERHUB_CREDENTIALS=credentials('dockerlogin')
 }

   stages {
      stage('SCM_Checkout') {
         steps {
            echo "Perform SCM Checkout"
    git 'https://github.com/Chandrasekhar1312/BankingApp.git'
         }
      }
      stage('Application Build') {
         steps {
            echo "Perform Application Build"
   sh 'mvn clean package'
         }
      }
      stage('Build Docker Image') {
         steps {
   sh 'docker version'
   sh "docker build -t whitehouse23/bankapp-app:${BUILD_NUMBER} ."
   sh 'docker image list'
   sh "docker tag whitehouse23/bankapp-app:${BUILD_NUMBER} whitehouse23/bankapp-app:latest"
         }
      }
  stage('Login2DockerHub') {

   steps {
       sh 'echo $DOCKERHUB_CREDENTIALS_PSW | docker login -u $DOCKERHUB_CREDENTIALS_USR --
password-stdin'
   }
  }
  stage('Publish_to_Docker_Registry') {
   steps {
    sh "docker push whitehouse23/bankapp-app:latest"
   }
  }
      stage('Deploy to Kubernetes Cluster') {
         steps {
            script{
                                   sshPublisher(publishers: [sshPublisherDesc(configName: 'kubmaster', transfers:
[sshTransfer(cleanRemote: false, excludes: '', execCommand: 'kubectl apply -f kubernetesdeploy.yaml',
execTimeout: 120000, flatten: false, makeEmptyDirs: false, noDefaultExcludes: false, patternSeparator: '[,
]+', remoteDirectory: '.', remoteDirectorySDF: false, removePrefix: '', sourceFiles: '*.yaml')],
usePromotionTimestamp: false, useWorkspaceInPromotion: false, verbose: false)])
            }
         }
      }
   }
}
```

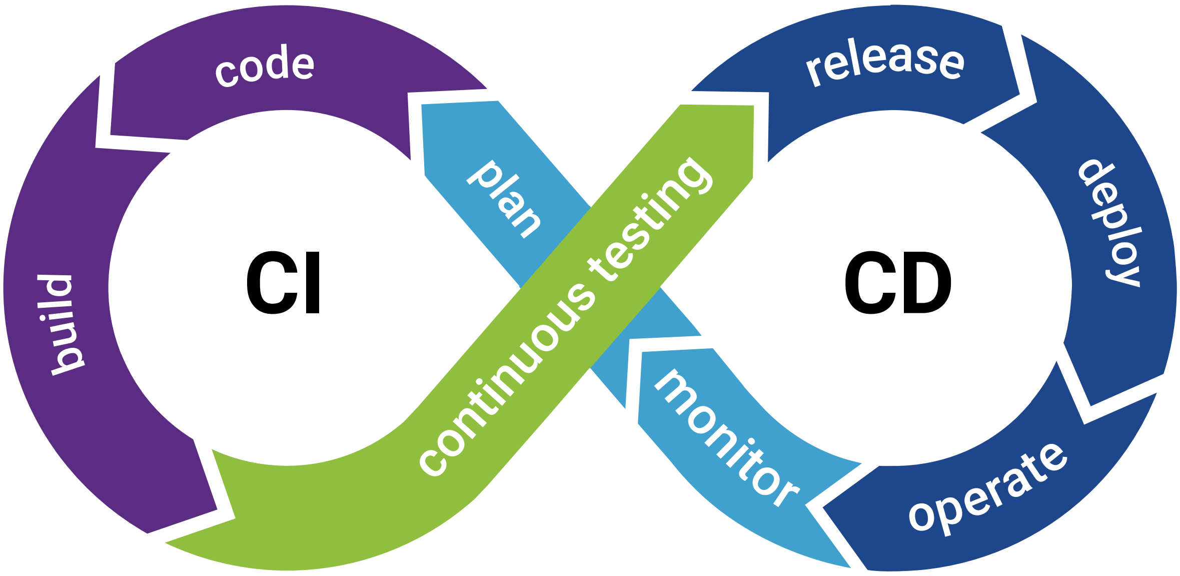
- **Now we can use prometheus and grafana for continuous monitoring of jenkins server**





Conclusion

- 🚀 The DevOps implementation automates and streamlines the software development lifecycle.
- 🔧 Tools like Jenkins, Docker, Kubernetes, Terraform, and Ansible enable efficient deployment and management.
- 📊 Prometheus and Grafana enhance system monitoring and observability.
- ⏩ The approach accelerates software delivery speed and minimizes errors.
- ✅ Ensures a scalable, reliable, and high-performance application infrastructure.
- 🎯 Optimizes operational efficiency, making it a robust solution for modern software development.


code
build
CI
plan
continuous testing
release
deploy
CD
monitor
operate