**Healthcare Domain Project – Medicure**

**Project Name:** Medicure – Healthcare Domain Project  
**Objective:** Migrate from monolithic to microservices architecture using DevOps.  
**Cloud Provider:** AWS  
**GitHub Repository:** [Star-Agile Healthcare Project](https://github.com/StarAgileDevOpsTraining/star-agile-health-care.git)

**📌 1. Problem Statement**

**Medicure is a super specialty hospital** based in New York, USA and provide world class treatment and surgery including Heart, Liver, Kidney transplants and first robotic surgery center. The chain is owned and managed by Global Health Limited.

**Challenges Faced by Medicure:**

✅ Complex monolithic architecture was difficult to scale.  
  
✅ Manual testing and deployments caused delays.  
  
✅ Scaling issues during high patient inflow.  
  
✅ Manual infrastructure provisioning took too much time.  
  
✅ Application monitoring was inefficient.

**Solution:**

👉 Implement **Microservices** using **Spring Boot & H2 Database**.  
  
👉 Use **DevOps tools** to automate Continuous Integration & Deployment (CI/CD).  
  
👉 Deploy the application on **AWS** using **Docker, Kubernetes, and Terraform**.  
  
👉 Monitor application performance with **Prometheus & Grafana**.

**📌 2. Technologies Used**

| **Component** | **Technology** |
| --- | --- |
| **Version Control** | Git, GitHub |
| **CI/CD Pipeline** | Jenkins |
| **Containerization** | Docker |
| **Orchestration** | Kubernetes |
| **Infrastructure as Code** | Terraform |
| **Configuration Management** | Ansible |
| **Application Framework** | Java, Spring Boot, Maven |
| **Testing** | Junit, Selenium, TestNG |

**📌 Step-by-Step CI/CD Implementation**

**🛠️ Step 1: Automate Configuration Using Ansible**

1. Install Ansible on the **Jenkins Master Node**:

sudo apt update && sudo apt install ansible -y

1. Create Ansible playbooks to:
   * Install required dependencies.
   * Configure **Kubernetes Master & Worker Nodes**.
   * Set up **Nginx as a Load Balancer**.

**🛠️ Step 2: Push Code to GitHub**

1. Ensure your GitHub repository is up-to-date:

git add .

git commit -m "Final CI/CD implementation"

git push origin master

1. **Jenkins** will automatically trigger the deployment.

**🛠️ Step 3: Set Up AWS EC2 Instances**

Launch **5 EC2 instances** on AWS:

1. Jenkins Master
2. Jenkins Slave
3. Kubernetes Master
4. Kubernetes Worker Node 1
5. Kubernetes Worker Node 2c

**Install Required Software on Each Instance:**

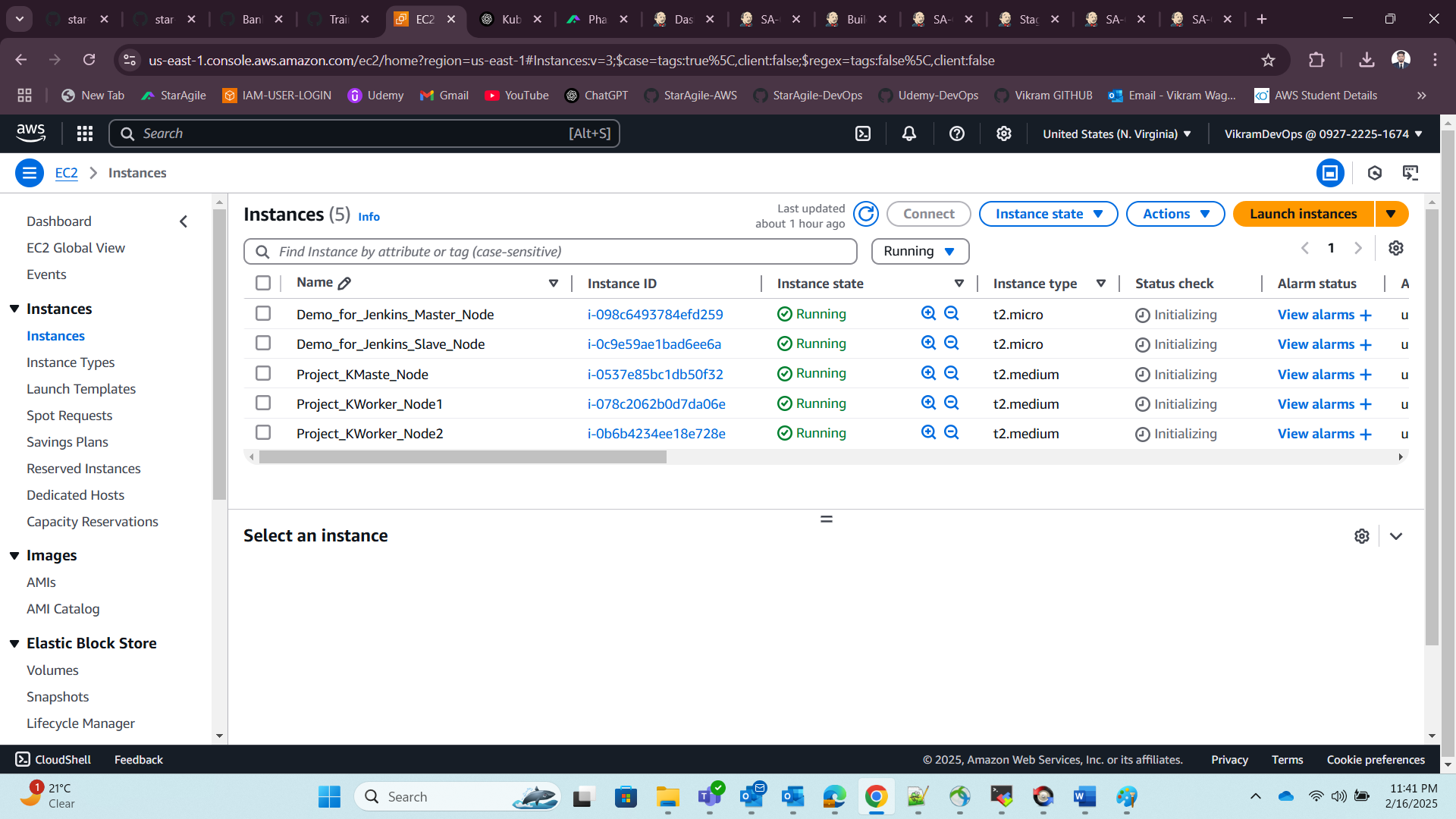
sudo apt update

sudo apt install docker.io -y

sudo apt install kubectl -y

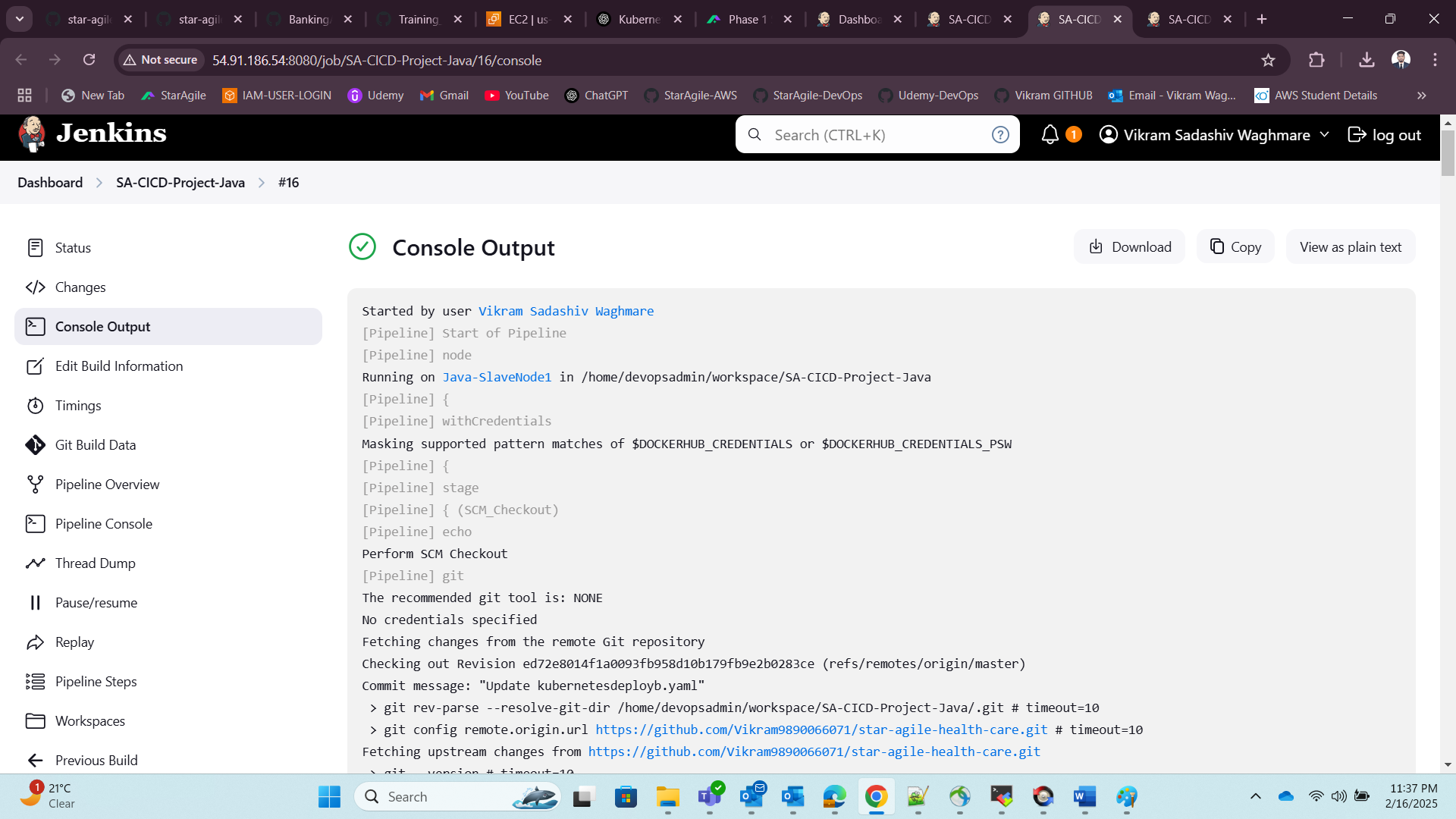
**Configure Security Groups:**

✅ **Port 22** - SSH access  
✅ **Port 8080** - Jenkins  
✅ **Port 31030** - Kubernetes Service

📷 **EC2 Instances Running on AWS**

**🛠️ Step 4: Set Up Jenkins for CI/CD**

1. **Install Jenkins** on the Master EC2 instance.
2. Install **Jenkins Plugins**:
   * **Pipeline, Docker, Kubernetes, GitHub, and Ansible** plugins.
3. **Configure GitHub Webhook**:
   * Auto-trigger Jenkins job on **Git push**.

📷 **Jenkins Console Output - Build Execution**  
  
📷 **Jenkins Build Success Screenshot**A screenshot of a computer

Description automatically generated

📷 **Jenkins Pipeline Steps**A screenshot of a computer

Description automatically generated

📷 **Jenkins Multi-Pipeline View**A screenshot of a computer

Description automatically generated

**🛠️ Step 5: Build & Deploy the Application**

1. **SCM Checkout**: Clone the GitHub repository.

git clone https://github.com/StarAgileDevOpsTraining/star-agile-health-care.git

1. **Build the application using Maven**:

mvn clean package

1. **Run unit tests and generate reports**:

mvn test

1. **Create a Docker Image**:

docker build -t vsw210/healthcare-eta-app:latest .

1. **Push the Docker image to Docker Hub**:

docker push vsw210/healthcare-eta-app:latest

1. **Deploy to Kubernetes Cluster**:

kubectl apply -f kubernetesdeployb.yaml

📷 **Kubernetes Deployment Status**A screenshot of a computer

Description automatically generated

📷 **Docker Hub Repository with Uploaded Images**A screenshot of a computer

Description automatically generated

**🛠️ Step 6: Application Deployment**

📷 **Final Running Web Application**

A screenshot of a computer

Description automatically generated

**Monitoring Setup**

**Installing & Configuring Prometheus & Grafana**

Installed Prometheus to collect metrics from Kubernetes clusters.

Configured Grafana dashboards to visualize application and infrastructure metrics.

**Monitored key metrics such as:**

* CPU Utilization
* Memory Usage
* Disk Space Utilization
* Network Traffic

**Monitoring Setup**

**Step 6: Installing & Configuring Prometheus & Grafana**

In this step, we will install **Prometheus** and **Grafana** on our Kubernetes cluster to monitor the infrastructure and application metrics.

**1. Install Prometheus on Kubernetes**

**Step 1: Create a Prometheus Namespace**

kubectl create namespace monitoring

**Step 2: Deploy Prometheus using Helm**

1. Add the Helm repository for Prometheus:

helm repo add prometheus-community <https://prometheus-community.github.io/helm->charts

helm repo update

1. Install Prometheus:

helm install prometheus prometheus-community/kube-prometheus-stack --namespace monitoring

1. Verify the installation:

kubectl get pods -n monitoring

**Output should show running Prometheus components:**

NAME READY STATUS RESTARTS AGE

prometheus-operator-xyz123 1/1 Running 0 2m

prometheus-prometheus-node-exporter-0 1/1 Running 0 2m

prometheus-kube-state-metrics-0 1/1 Running 0 2m

prometheus-prometheus-server-0 2/2 Running 0 2m

**2. Install Grafana on Kubernetes**

**Step 1: Deploy Grafana using Helm**

helm install grafana prometheus-community/grafana --namespace monitoring **Step 2: Get Grafana Admin Password**

kubectl get secret --namespace monitoring grafana -o jsonpath="{.data.admin-password}" | base64 --decode ; echo

**Step 3: Port-forward Grafana to Access the UI**

kubectl port-forward svc/grafana 3000:80 -n monitoring

* Open **http://localhost:3000** in your browser.
* Login with:
  + **Username**: admin
  + **Password**: (output from Step 2)

**3. Configure Prometheus as a Data Source in Grafana**

1. **Login to Grafana UI** at http://localhost:3000
2. Navigate to **Configuration → Data Sources**
3. Click **Add Data Source**
4. Select **Prometheus**
5. Enter Prometheus URL:

http://prometheus-server.monitoring.svc.cluster.local

1. Click **Save & Test**

**4. Create Grafana Dashboards for Monitoring**

**Step 1: Import a Prebuilt Dashboard**

1. Go to **Grafana → Dashboards → Import**
2. Enter dashboard ID **3119** (Kubernetes Cluster Monitoring)
3. Select **Prometheus** as the data source
4. Click **Import**

**Step 2: View Metrics**

You will see:

* **CPU Utilization**
* **Memory Usage**
* **Disk Space Utilization**
* **Network Traffic**
* **Pod Status**

**5. Validate Monitoring Setup**

* Run the following command to check if Prometheus is scraping metrics:

kubectl logs -l app=prometheus -n monitoring

* Open **Grafana Dashboards** and verify that data is being visualized.

**Verifying Prometheus Metrics**

After setting up **Prometheus**, you can check the **real-time system metrics**.

**Steps:**

1. **Run Port Forwarding**

kubectl port-forward svc/prometheus-server 9090:80 -n monitoring

1. **Access Prometheus UI**
   * Open **http://localhost:9090**
   * Run **PromQL Queries**:

container\_cpu\_usage\_seconds\_total

container\_memory\_usage\_bytes

node\_filesystem\_avail\_bytes

**Sample Prometheus Query Output**

Time | CPU Usage | Memory Usage | Disk Usage | Network Traffic

---- | -------- | ------------ | ---------- | ---------------

12:00 | 0.2 cores | 450 MB | 50 GB | 1.5 MB/s

12:05 | 0.3 cores | 500 MB | 49.5 GB | 2.0 MB/s

12:10 | 0.4 cores | 600 MB | 49 GB | 2.8 MB/s

**2. Grafana Dashboard - Sample Metrics**

Once **Prometheus is configured** as a **data source**, create **visual dashboards in Grafana**.

**3. Kubernetes Monitoring - Sample Outputs**

Run:

kubectl top nodes

**Sample Output**

NAME CPU(cores) CPU% MEMORY(bytes) MEMORY%

healthcare-node1 130m 20% 500Mi 35%

healthcare-node2 250m 30% 750Mi 45%

Run:

kubectl top pods -n monitoring

**Sample Output**

NAME CPU(cores) MEMORY(bytes)

prometheus-server-0 90m 200Mi

grafana-xyz123 75m 180Mi

healthcare-app-backend 110m 400Mi

**4. Alerts & Anomaly Detection**

Set **Alerts for High CPU Usage or API Downtime** using **Prometheus AlertManager**.

**Example Alert Rule for API Downtime**

groups:

- name: HealthcareAPIDowntime

rules:

- alert: APIDown

expr: up{job="healthcare-api"} == 0

for: 2m

labels:

severity: critical

annotations:

description: "The Healthcare API has been down for more than 2 minutes."

**📌 Jenkins Pipeline Code (Jenkinsfile)**

This **Jenkinsfile** automates the build, test, containerization, and deployment process.

pipeline {

agent any

environment {

DOCKER\_HUB\_CREDENTIALS = credentials('docker-hub-credentials')

REPO\_URL = 'https://github.com/StarAgileDevOpsTraining/star-agile-health-care.git'

}

stages {

stage('SCM Checkout') {

steps {

git branch: 'master', url: "${REPO\_URL}"

}

}

stage('Build Application') {

steps {

sh 'mvn clean package'

}

}

stage('Run Unit Tests') {

steps {

sh 'mvn test'

}

}

stage('Build Docker Image') {

steps {

sh 'docker build -t vsw210/healthcare-eta-app:latest .'

}

}

stage('Push to Docker Hub') {

steps {

withDockerRegistry([credentialsId: 'docker-hub-credentials']) {

sh 'docker push vsw210/healthcare-eta-app:latest'

}

}

}

stage('Deploy to Kubernetes') {

steps {

sh 'kubectl apply -f kubernetesdeployb.yaml'

}

}

stage('Run Selenium Tests') {

steps {

sh 'pytest selenium\_tests.py'

}

}

}

}

**📌 Kubernetes Deployment YAML**

This file **defines the deployment and service configuration**.

apiVersion: apps/v1

kind: Deployment

metadata:

name: healthcare-eta-deploy

labels:

app: healthcare-eta-app

spec:

replicas: 3

selector:

matchLabels:

app: healthcare-eta-app

template:

metadata:

labels:

app: healthcare-eta-app

spec:

containers:

- name: healthcare-eta-container

image: vsw210/healthcare-eta-app:latest

ports:

- containerPort: 8081

---

apiVersion: v1

kind: Service

metadata:

name: healthcare-eta-np-service

labels:

app: healthcare-eta-app

spec:

selector:

app: healthcare-eta-app

type: NodePort

ports:

- nodePort: 31030

port: 8081

targetPort: 8081

**DETAIL LOG FIL: -**Started by user Vikram Sadashiv Waghmare

[Pipeline] Start of Pipeline

[Pipeline] node

Running on Java-SlaveNode1 in /home/devopsadmin/workspace/SA-CICD-Project-Java

[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

Checking out Revision ed72e8014f1a0093fb958d10b179fb9e2b0283ce (refs/remotes/origin/master)

Commit message: "Update kubernetesdeployb.yaml"

> git rev-parse --resolve-git-dir /home/devopsadmin/workspace/SA-CICD-Project-Java/.git # timeout=10

> git config remote.origin.url https://github.com/Vikram9890066071/star-agile-health-care.git # timeout=10

Fetching upstream changes from https://github.com/Vikram9890066071/star-agile-health-care.git

> git --version # timeout=10

> git --version # 'git version 2.34.1'

> git fetch --tags --force --progress -- https://github.com/Vikram9890066071/star-agile-health-care.git +refs/heads/\*:refs/remotes/origin/\* # timeout=10

> git rev-parse refs/remotes/origin/master^{commit} # timeout=10

> git config core.sparsecheckout # timeout=10

> git checkout -f ed72e8014f1a0093fb958d10b179fb9e2b0283ce # timeout=10

> git branch -a -v --no-abbrev # timeout=10

> git branch -D master # timeout=10

> git checkout -b master ed72e8014f1a0093fb958d10b179fb9e2b0283ce # timeout=10

> git rev-list --no-walk ecd9050b1c28b28940e5d74aa08fc73ccc405760 # timeout=10

[Pipeline] }

[Pipeline] // stage

[Pipeline] stage

[Pipeline] { (Application Build)

[Pipeline] echo

Perform Application Build

[Pipeline] sh

+ mvn clean package

[[1;34mINFO[m] Scanning for projects...

[[1;34mINFO[m]

[[1;34mINFO[m] [1m-------------------< [0;36mcom.project.staragile:medicure[0;1m >-------------------[m

[[1;34mINFO[m] [1mBuilding medicure 0.0.1-SNAPSHOT[m

[[1;34mINFO[m] [1m--------------------------------[ jar ]---------------------------------[m

[[1;34mINFO[m]

[[1;34mINFO[m] [1m--- [0;32mmaven-clean-plugin:3.2.0:clean[m [1m(default-clean)[m @ [36mmedicure[0;1m ---[m

[[1;34mINFO[m] Deleting /home/devopsadmin/workspace/SA-CICD-Project-Java/target

[[1;34mINFO[m]

[[1;34mINFO[m] [1m--- [0;32mmaven-resources-plugin:3.2.0:resources[m [1m(default-resources)[m @ [36mmedicure[0;1m ---[m

[[1;34mINFO[m] Using 'UTF-8' encoding to copy filtered resources.

[[1;34mINFO[m] Using 'UTF-8' encoding to copy filtered properties files.

[[1;34mINFO[m] Copying 1 resource

[[1;34mINFO[m] Copying 32 resources

[[1;34mINFO[m]

[[1;34mINFO[m] [1m--- [0;32mmaven-compiler-plugin:3.10.1:compile[m [1m(default-compile)[m @ [36mmedicure[0;1m ---[m

[[1;34mINFO[m] Changes detected - recompiling the module!

[[1;34mINFO[m] Compiling 5 source files to /home/devopsadmin/workspace/SA-CICD-Project-Java/target/classes

[[1;34mINFO[m]

[[1;34mINFO[m] [1m--- [0;32mmaven-resources-plugin:3.2.0:testResources[m [1m(default-testResources)[m @ [36mmedicure[0;1m ---[m

[[1;34mINFO[m] Using 'UTF-8' encoding to copy filtered resources.

[[1;34mINFO[m] Using 'UTF-8' encoding to copy filtered properties files.

[[1;34mINFO[m] skip non existing resourceDirectory /home/devopsadmin/workspace/SA-CICD-Project-Java/src/test/resources

[[1;34mINFO[m]

[[1;34mINFO[m] [1m--- [0;32mmaven-compiler-plugin:3.10.1:testCompile[m [1m(default-testCompile)[m @ [36mmedicure[0;1m ---[m

[[1;34mINFO[m] Changes detected - recompiling the module!

[[1;34mINFO[m] Compiling 2 source files to /home/devopsadmin/workspace/SA-CICD-Project-Java/target/test-classes

[[1;34mINFO[m]

[[1;34mINFO[m] [1m--- [0;32mmaven-surefire-plugin:2.22.2:test[m [1m(default-test)[m @ [36mmedicure[0;1m ---[m

[[1;34mINFO[m]

[[1;34mINFO[m] -------------------------------------------------------

[[1;34mINFO[m] T E S T S

[[1;34mINFO[m] -------------------------------------------------------

[[1;34mINFO[m] Running com.project.staragile.[1mTestMedicureService[m

18:03:05.996 [main] DEBUG org.springframework.test.context.BootstrapUtils - Instantiating CacheAwareContextLoaderDelegate from class [org.springframework.test.context.cache.DefaultCacheAwareContextLoaderDelegate]

18:03:06.044 [main] DEBUG org.springframework.test.context.BootstrapUtils - Instantiating BootstrapContext using constructor [public org.springframework.test.context.support.DefaultBootstrapContext(java.lang.Class,org.springframework.test.context.CacheAwareContextLoaderDelegate)]

18:03:06.165 [main] DEBUG org.springframework.test.context.BootstrapUtils - Instantiating TestContextBootstrapper for test class [com.project.staragile.TestMedicureService] from class [org.springframework.boot.test.context.SpringBootTestContextBootstrapper]

18:03:06.192 [main] INFO org.springframework.boot.test.context.SpringBootTestContextBootstrapper - Neither @ContextConfiguration nor @ContextHierarchy found for test class [com.project.staragile.TestMedicureService], using SpringBootContextLoader

18:03:06.203 [main] DEBUG org.springframework.test.context.support.AbstractContextLoader - Did not detect default resource location for test class [com.project.staragile.TestMedicureService]: class path resource [com/project/staragile/TestMedicureService-context.xml] does not exist

18:03:06.204 [main] DEBUG org.springframework.test.context.support.AbstractContextLoader - Did not detect default resource location for test class [com.project.staragile.TestMedicureService]: class path resource [com/project/staragile/TestMedicureServiceContext.groovy] does not exist

18:03:06.206 [main] INFO org.springframework.test.context.support.AbstractContextLoader - Could not detect default resource locations for test class [com.project.staragile.TestMedicureService]: no resource found for suffixes {-context.xml, Context.groovy}.

18:03:06.208 [main] INFO org.springframework.test.context.support.AnnotationConfigContextLoaderUtils - Could not detect default configuration classes for test class [com.project.staragile.TestMedicureService]: TestMedicureService does not declare any static, non-private, non-final, nested classes annotated with @Configuration.

18:03:06.343 [main] DEBUG org.springframework.test.context.support.ActiveProfilesUtils - Could not find an 'annotation declaring class' for annotation type [org.springframework.test.context.ActiveProfiles] and class [com.project.staragile.TestMedicureService]

18:03:06.474 [main] DEBUG org.springframework.context.annotation.ClassPathScanningCandidateComponentProvider - Identified candidate component class: file [/home/devopsadmin/workspace/SA-CICD-Project-Java/target/classes/com/project/staragile/MedicureApplication.class]

18:03:06.502 [main] INFO org.springframework.boot.test.context.SpringBootTestContextBootstrapper - Found @SpringBootConfiguration com.project.staragile.MedicureApplication for test class com.project.staragile.TestMedicureService

18:03:06.693 [main] DEBUG org.springframework.boot.test.context.SpringBootTestContextBootstrapper - @TestExecutionListeners is not present for class [com.project.staragile.TestMedicureService]: using defaults.

18:03:06.697 [main] INFO org.springframework.boot.test.context.SpringBootTestContextBootstrapper - Loaded default TestExecutionListener class names from location [META-INF/spring.factories]: [org.springframework.boot.test.mock.mockito.MockitoTestExecutionListener, org.springframework.boot.test.mock.mockito.ResetMocksTestExecutionListener, org.springframework.boot.test.autoconfigure.restdocs.RestDocsTestExecutionListener, org.springframework.boot.test.autoconfigure.web.client.MockRestServiceServerResetTestExecutionListener, org.springframework.boot.test.autoconfigure.web.servlet.MockMvcPrintOnlyOnFailureTestExecutionListener, org.springframework.boot.test.autoconfigure.web.servlet.WebDriverTestExecutionListener, org.springframework.boot.test.autoconfigure.webservices.client.MockWebServiceServerTestExecutionListener, org.springframework.test.context.web.ServletTestExecutionListener, org.springframework.test.context.support.DirtiesContextBeforeModesTestExecutionListener, org.springframework.test.context.event.ApplicationEventsTestExecutionListener, org.springframework.test.context.support.DependencyInjectionTestExecutionListener, org.springframework.test.context.support.DirtiesContextTestExecutionListener, org.springframework.test.context.transaction.TransactionalTestExecutionListener, org.springframework.test.context.jdbc.SqlScriptsTestExecutionListener, org.springframework.test.context.event.EventPublishingTestExecutionListener]

18:03:06.754 [main] INFO org.springframework.boot.test.context.SpringBootTestContextBootstrapper - Using TestExecutionListeners: [org.springframework.test.context.web.ServletTestExecutionListener@12359a82, org.springframework.test.context.support.DirtiesContextBeforeModesTestExecutionListener@68df9280, org.springframework.test.context.event.ApplicationEventsTestExecutionListener@479460a6, org.springframework.boot.test.mock.mockito.MockitoTestExecutionListener@7164ca4c, org.springframework.boot.test.autoconfigure.SpringBootDependencyInjectionTestExecutionListener@4f3bbf68, org.springframework.test.context.support.DirtiesContextTestExecutionListener@5be46f9d, org.springframework.test.context.transaction.TransactionalTestExecutionListener@3f91b517, org.springframework.test.context.jdbc.SqlScriptsTestExecutionListener@68702e03, org.springframework.test.context.event.EventPublishingTestExecutionListener@7a220c9a, org.springframework.boot.test.mock.mockito.ResetMocksTestExecutionListener@2421cc4, org.springframework.boot.test.autoconfigure.restdocs.RestDocsTestExecutionListener@30c93896, org.springframework.boot.test.autoconfigure.web.client.MockRestServiceServerResetTestExecutionListener@59a008ba, org.springframework.boot.test.autoconfigure.web.servlet.MockMvcPrintOnlyOnFailureTestExecutionListener@338c99c8, org.springframework.boot.test.autoconfigure.web.servlet.WebDriverTestExecutionListener@21ba0741, org.springframework.boot.test.autoconfigure.webservices.client.MockWebServiceServerTestExecutionListener@58ce9668]

18:03:06.765 [main] DEBUG org.springframework.test.context.support.AbstractDirtiesContextTestExecutionListener - Before test class: context [DefaultTestContext@5c2375a9 testClass = TestMedicureService, testInstance = [null], testMethod = [null], testException = [null], mergedContextConfiguration = [WebMergedContextConfiguration@60129b9a testClass = TestMedicureService, locations = '{}', classes = '{class com.project.staragile.MedicureApplication}', contextInitializerClasses = '[]', activeProfiles = '{}', propertySourceLocations = '{}', propertySourceProperties = '{org.springframework.boot.test.context.SpringBootTestContextBootstrapper=true}', contextCustomizers = set[org.springframework.boot.test.context.filter.ExcludeFilterContextCustomizer@2accdbb5, org.springframework.boot.test.json.DuplicateJsonObjectContextCustomizerFactory$DuplicateJsonObjectContextCustomizer@27ff5d15, org.springframework.boot.test.mock.mockito.MockitoContextCustomizer@0, org.springframework.boot.test.web.client.TestRestTemplateContextCustomizer@6bedbc4d, org.springframework.boot.test.autoconfigure.actuate.metrics.MetricsExportContextCustomizerFactory$DisableMetricExportContextCustomizer@37654521, org.springframework.boot.test.autoconfigure.properties.PropertyMappingContextCustomizer@0, org.springframework.boot.test.autoconfigure.web.servlet.WebDriverContextCustomizerFactory$Customizer@76c3e77a, org.springframework.boot.test.context.SpringBootTestArgs@1, org.springframework.boot.test.context.SpringBootTestWebEnvironment@394e1a0f], resourceBasePath = 'src/main/webapp', contextLoader = 'org.springframework.boot.test.context.SpringBootContextLoader', parent = [null]], attributes = map['org.springframework.test.context.web.ServletTestExecutionListener.activateListener' -> true]], class annotated with @DirtiesContext [false] with mode [null].

. \_\_\_\_ \_ \_\_ \_ \_

/\\ / \_\_\_'\_ \_\_ \_ \_(\_)\_ \_\_ \_\_ \_ \ \ \ \

( ( )\\_\_\_ | '\_ | '\_| | '\_ \/ \_` | \ \ \ \

\\/ \_\_\_)| |\_)| | | | | || (\_| | ) ) ) )

' |\_\_\_\_| .\_\_|\_| |\_|\_| |\_\\_\_, | / / / /

=========|\_|==============|\_\_\_/=/\_/\_/\_/

:: Spring Boot :: (v2.7.4)

2025-02-16 18:03:07.375 INFO 2718 --- [ main] c.project.staragile.TestMedicureService : Starting TestMedicureService using Java 17.0.13 on ip-172-31-84-112 with PID 2718 (started by devopsadmin in /home/devopsadmin/workspace/SA-CICD-Project-Java)

2025-02-16 18:03:07.378 INFO 2718 --- [ main] c.project.staragile.TestMedicureService : No active profile set, falling back to 1 default profile: "default"

2025-02-16 18:03:08.632 INFO 2718 --- [ main] .s.d.r.c.RepositoryConfigurationDelegate : Bootstrapping Spring Data JPA repositories in DEFAULT mode.

2025-02-16 18:03:08.719 INFO 2718 --- [ main] .s.d.r.c.RepositoryConfigurationDelegate : Finished Spring Data repository scanning in 75 ms. Found 1 JPA repository interfaces.

2025-02-16 18:03:09.794 INFO 2718 --- [ main] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Starting...

2025-02-16 18:03:10.213 INFO 2718 --- [ main] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Start completed.

2025-02-16 18:03:10.359 INFO 2718 --- [ main] o.hibernate.jpa.internal.util.LogHelper : HHH000204: Processing PersistenceUnitInfo [name: default]

2025-02-16 18:03:10.499 INFO 2718 --- [ main] org.hibernate.Version : HHH000412: Hibernate ORM core version 5.6.11.Final

2025-02-16 18:03:10.873 INFO 2718 --- [ main] o.hibernate.annotations.common.Version : HCANN000001: Hibernate Commons Annotations {5.1.2.Final}

2025-02-16 18:03:11.172 INFO 2718 --- [ main] org.hibernate.dialect.Dialect : HHH000400: Using dialect: org.hibernate.dialect.H2Dialect

2025-02-16 18:03:12.172 INFO 2718 --- [ main] o.h.e.t.j.p.i.JtaPlatformInitiator : HHH000490: Using JtaPlatform implementation: [org.hibernate.engine.transaction.jta.platform.internal.NoJtaPlatform]

2025-02-16 18:03:12.197 INFO 2718 --- [ main] j.LocalContainerEntityManagerFactoryBean : Initialized JPA EntityManagerFactory for persistence unit 'default'

2025-02-16 18:03:13.192 WARN 2718 --- [ main] JpaBaseConfiguration$JpaWebConfiguration : spring.jpa.open-in-view is enabled by default. Therefore, database queries may be performed during view rendering. Explicitly configure spring.jpa.open-in-view to disable this warning

2025-02-16 18:03:14.072 INFO 2718 --- [ main] o.s.b.a.w.s.WelcomePageHandlerMapping : Adding welcome page: class path resource [static/index.html]

2025-02-16 18:03:14.633 INFO 2718 --- [ main] c.project.staragile.TestMedicureService : Started TestMedicureService in 7.758 seconds (JVM running for 10.001)

[[1;34mINFO[m] [1;32mTests run: [0;1;32m1[m, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 9.279 s - in com.project.staragile.[1mTestMedicureService[m

[[1;34mINFO[m] Running com.project.staragile.[1mMedicureApplicationTests[m

2025-02-16 18:03:15.041 INFO 2718 --- [ main] .b.t.c.SpringBootTestContextBootstrapper : Neither @ContextConfiguration nor @ContextHierarchy found for test class [com.project.staragile.MedicureApplicationTests], using SpringBootContextLoader

2025-02-16 18:03:15.046 INFO 2718 --- [ main] o.s.t.c.support.AbstractContextLoader : Could not detect default resource locations for test class [com.project.staragile.MedicureApplicationTests]: no resource found for suffixes {-context.xml, Context.groovy}.

2025-02-16 18:03:15.046 INFO 2718 --- [ main] t.c.s.AnnotationConfigContextLoaderUtils : Could not detect default configuration classes for test class [com.project.staragile.MedicureApplicationTests]: MedicureApplicationTests does not declare any static, non-private, non-final, nested classes annotated with @Configuration.

2025-02-16 18:03:15.047 INFO 2718 --- [ main] .b.t.c.SpringBootTestContextBootstrapper : Found @SpringBootConfiguration com.project.staragile.MedicureApplication for test class com.project.staragile.MedicureApplicationTests

2025-02-16 18:03:15.052 INFO 2718 --- [ main] .b.t.c.SpringBootTestContextBootstrapper : Loaded default TestExecutionListener class names from location [META-INF/spring.factories]: [org.springframework.boot.test.mock.mockito.MockitoTestExecutionListener, org.springframework.boot.test.mock.mockito.ResetMocksTestExecutionListener, org.springframework.boot.test.autoconfigure.restdocs.RestDocsTestExecutionListener, org.springframework.boot.test.autoconfigure.web.client.MockRestServiceServerResetTestExecutionListener, org.springframework.boot.test.autoconfigure.web.servlet.MockMvcPrintOnlyOnFailureTestExecutionListener, org.springframework.boot.test.autoconfigure.web.servlet.WebDriverTestExecutionListener, org.springframework.boot.test.autoconfigure.webservices.client.MockWebServiceServerTestExecutionListener, org.springframework.test.context.web.ServletTestExecutionListener, org.springframework.test.context.support.DirtiesContextBeforeModesTestExecutionListener, org.springframework.test.context.event.ApplicationEventsTestExecutionListener, org.springframework.test.context.support.DependencyInjectionTestExecutionListener, org.springframework.test.context.support.DirtiesContextTestExecutionListener, org.springframework.test.context.transaction.TransactionalTestExecutionListener, org.springframework.test.context.jdbc.SqlScriptsTestExecutionListener, org.springframework.test.context.event.EventPublishingTestExecutionListener]

2025-02-16 18:03:15.057 INFO 2718 --- [ main] .b.t.c.SpringBootTestContextBootstrapper : Using TestExecutionListeners: [org.springframework.test.context.web.ServletTestExecutionListener@3810806c, org.springframework.test.context.support.DirtiesContextBeforeModesTestExecutionListener@2b76ecd5, org.springframework.test.context.event.ApplicationEventsTestExecutionListener@7bd7c4cf, org.springframework.boot.test.mock.mockito.MockitoTestExecutionListener@5b9df3b3, org.springframework.boot.test.autoconfigure.SpringBootDependencyInjectionTestExecutionListener@67b20b4c, org.springframework.test.context.support.DirtiesContextTestExecutionListener@7bbe532b, org.springframework.test.context.transaction.TransactionalTestExecutionListener@3e0a112f, org.springframework.test.context.jdbc.SqlScriptsTestExecutionListener@abe7d36, org.springframework.test.context.event.EventPublishingTestExecutionListener@7da1ef46, org.springframework.boot.test.mock.mockito.ResetMocksTestExecutionListener@740fccd2, org.springframework.boot.test.autoconfigure.restdocs.RestDocsTestExecutionListener@1ae23815, org.springframework.boot.test.autoconfigure.web.client.MockRestServiceServerResetTestExecutionListener@63ccb1b2, org.springframework.boot.test.autoconfigure.web.servlet.MockMvcPrintOnlyOnFailureTestExecutionListener@1165a952, org.springframework.boot.test.autoconfigure.web.servlet.WebDriverTestExecutionListener@1b868ef0, org.springframework.boot.test.autoconfigure.webservices.client.MockWebServiceServerTestExecutionListener@102f3f05]

[[1;34mINFO[m] [1;32mTests run: [0;1;32m2[m, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.022 s - in com.project.staragile.[1mMedicureApplicationTests[m

2025-02-16 18:03:15.171 INFO 2718 --- [ionShutdownHook] j.LocalContainerEntityManagerFactoryBean : Closing JPA EntityManagerFactory for persistence unit 'default'

2025-02-16 18:03:15.174 INFO 2718 --- [ionShutdownHook] .SchemaDropperImpl$DelayedDropActionImpl : HHH000477: Starting delayed evictData of schema as part of SessionFactory shut-down'

2025-02-16 18:03:15.182 INFO 2718 --- [ionShutdownHook] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Shutdown initiated...

2025-02-16 18:03:15.196 INFO 2718 --- [ionShutdownHook] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Shutdown completed.

[[1;34mINFO[m]

[[1;34mINFO[m] Results:

[[1;34mINFO[m]

[[1;34mINFO[m] [1;32mTests run: 3, Failures: 0, Errors: 0, Skipped: 0[m

[[1;34mINFO[m]

[[1;34mINFO[m]

[[1;34mINFO[m] [1m--- [0;32mmaven-jar-plugin:3.2.2:jar[m [1m(default-jar)[m @ [36mmedicure[0;1m ---[m

[[1;34mINFO[m] Building jar: /home/devopsadmin/workspace/SA-CICD-Project-Java/target/medicure-0.0.1-SNAPSHOT.jar

[[1;34mINFO[m]

[[1;34mINFO[m] [1m--- [0;32mspring-boot-maven-plugin:2.7.4:repackage[m [1m(repackage)[m @ [36mmedicure[0;1m ---[m

[[1;34mINFO[m] Replacing main artifact with repackaged archive

[[1;34mINFO[m] [1m------------------------------------------------------------------------[m

[[1;34mINFO[m] [1;32mBUILD SUCCESS[m

[[1;34mINFO[m] [1m------------------------------------------------------------------------[m

[[1;34mINFO[m] Total time: 18.667 s

[[1;34mINFO[m] Finished at: 2025-02-16T18:03:17Z

[[1;34mINFO[m] [1m------------------------------------------------------------------------[m

[Pipeline] }

[Pipeline] // stage

[Pipeline] stage

[Pipeline] { (Build Docker Image)

[Pipeline] sh

+ docker version

Client:

Version: 24.0.7

API version: 1.43

Go version: go1.21.1

Git commit: 24.0.7-0ubuntu2~22.04.1

Built: Wed Mar 13 20:23:54 2024

OS/Arch: linux/amd64

Context: default

Server:

Engine:

Version: 24.0.7

API version: 1.43 (minimum version 1.12)

Go version: go1.21.1

Git commit: 24.0.7-0ubuntu2~22.04.1

Built: Wed Mar 13 20:23:54 2024

OS/Arch: linux/amd64

Experimental: false

containerd:

Version: 1.7.12

GitCommit:

runc:

Version: 1.1.12-0ubuntu2~22.04.1

GitCommit:

docker-init:

Version: 0.19.0

GitCommit:

[Pipeline] sh

+ docker build -t vsw210/healthcare-eta-app:16 .

DEPRECATED: The legacy builder is deprecated and will be removed in a future release.

Install the buildx component to build images with BuildKit:

https://docs.docker.com/go/buildx/

Sending build context to Docker daemon 72.79MB

Step 1/4 : FROM openjdk:11

---> 47a932d998b7

Step 2/4 : ARG JAR\_FILE=target/\*.jar

---> Using cache

---> ad5404ccb471

Step 3/4 : COPY ${JAR\_FILE} app.jar

---> 118a1b63faf2

Step 4/4 : ENTRYPOINT ["java","-jar","/app.jar"]

---> Running in 58e70a8b06ab

Removing intermediate container 58e70a8b06ab

---> 0247b51381aa

Successfully built 0247b51381aa

Successfully tagged vsw210/healthcare-eta-app:16

[Pipeline] sh

+ docker image list

REPOSITORY TAG IMAGE ID CREATED SIZE

vsw210/healthcare-eta-app 16 0247b51381aa 1 second ago 695MB

vsw210/healthcare-eta-app 15 5f02d939e984 15 minutes ago 695MB

vsw210/healthcare-eta-app latest 5f02d939e984 15 minutes ago 695MB

vsw210/healthcare-eta-app 14 f71eb83b433c 25 minutes ago 695MB

vsw210/healthcare-eta-app 13 38c8015edb3e 27 minutes ago 695MB

vsw210/healthcare-eta-app 12 e942bad7f9d2 31 minutes ago 695MB

vsw210/healthcare-eta-app 11 57bc0d406c88 33 minutes ago 695MB

vsw210/insurance-eta-app 10 5fd113536b08 2 hours ago 695MB

vsw210/insurance-eta-app latest 5fd113536b08 2 hours ago 695MB

vsw210/insurance-eta-app 9 607f6e16448a 2 hours ago 695MB

vsw210/insurance-eta-app 7 8f9ae2912f39 2 hours ago 695MB

vsw210/instance-eta-app 6 a4919861f7e9 3 hours ago 695MB

vsw210/instance-eta-app latest a4919861f7e9 3 hours ago 695MB

vsw210/instance-eta-app 5 8305665c2e6f 3 hours ago 695MB

vsw210/instance-eta-app 4 602770139f29 3 hours ago 695MB

vsw210/bankapp-eta-app 3 fc2310fd3476 4 hours ago 696MB

vsw210/bankapp-eta-app latest fc2310fd3476 4 hours ago 696MB

vsw210/bankapp-eta-app 2 931e60da6f65 5 hours ago 696MB

vsw210/sa-mywebapp 1 a508e0c5b1d3 6 hours ago 402MB

vsw210/sa-mywebapp latest a508e0c5b1d3 6 hours ago 402MB

vsw210/bankapp-eta-app 41 97fb7b492400 9 hours ago 696MB

vsw210/bankapp-eta-app 40 eebea5fd6cb8 10 hours ago 696MB

openjdk 11 47a932d998b7 2 years ago 654MB

tomcat 8.0 ef6a7c98d192 6 years ago 356MB

[Pipeline] sh

+ docker tag vsw210/healthcare-eta-app:16 vsw210/healthcare-eta-app:latest

[Pipeline] }

[Pipeline] // stage

[Pipeline] stage

[Pipeline] { (Login2DockerHub)

[Pipeline] sh

+ echo \*\*\*\*

+ docker login -u vsw210 --password-stdin

WARNING! Your password will be stored unencrypted in /home/devopsadmin/.docker/config.json.

Configure a credential helper to remove this warning. See

https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded

[Pipeline] }

[Pipeline] // stage

[Pipeline] stage

[Pipeline] { (Publish\_to\_Docker\_Registry)

[Pipeline] sh

+ docker push vsw210/healthcare-eta-app:latest

The push refers to repository [docker.io/vsw210/healthcare-eta-app]

e200c19c99fe: Preparing

7b7f3078e1db: Preparing

826c3ddbb29c: Preparing

b626401ef603: Preparing

9b55156abf26: Preparing

293d5db30c9f: Preparing

03127cdb479b: Preparing

9c742cd6c7a5: Preparing

293d5db30c9f: Waiting

03127cdb479b: Waiting

9c742cd6c7a5: Waiting

b626401ef603: Layer already exists

9b55156abf26: Layer already exists

826c3ddbb29c: Layer already exists

7b7f3078e1db: Layer already exists

03127cdb479b: Layer already exists

9c742cd6c7a5: Layer already exists

293d5db30c9f: Layer already exists

e200c19c99fe: Pushed

latest: digest: sha256:4d06c076cca02414a4d3d48404fb278fbcfcdecba59661e739d564c17d048f7e size: 2007

[Pipeline] }

[Pipeline] // stage

[Pipeline] stage

[Pipeline] { (Deploy to Kubernetes Cluster)

[Pipeline] script

[Pipeline] {

[Pipeline] sshPublisher

[Pipeline] }

[Pipeline] // script

SSH: Connecting from host [ip-172-31-84-112]

SSH: Connecting with configuration [KMaster\_Project] ...

SSH: EXEC: completed after 200 ms

SSH: Disconnecting configuration [KMaster\_Project] ...

SSH: Transferred 1 file(s)

[Pipeline] }

[Pipeline] // stage

[Pipeline] }

[Pipeline] // withCredentials

[Pipeline] }

[Pipeline] // node

[Pipeline] End of Pipeline

Finished: SUCCESS