Project Report: Deployment of a Java Web Application Using Jenkins, Docker, and Kubernetes

---

Overview:

This project involved automating the deployment of a Java-based web application using a complete DevOps pipeline. The goal was to containerize the application, build it through Jenkins, and deploy it seamlessly to a Kubernetes cluster managed via Minikube. This setup reflects a real-world CI/CD workflow that integrates development, testing, and production deployment in a continuous and efficient manner.

Technology Used:

During the project, the following tools and technologies were used:

- **Programming Language:** Java (Servlet/JSP-based application)
- **Build Tool:** Maven
- **Web Server:** Apache Tomcat
- **Containerization:** Docker
- **Orchestration:** Kubernetes (Minikube)
- **CI/CD Automation:** Jenkins
- **Operating System:** Ubuntu (EC2 instance)
- **Others:** YAML for Kubernetes manifests, Git for version control

---

Implementation Overview:

Here's a breakdown of the step-by-step process followed:

Step 1: Develop and Package the Web Application

- A Java-based application was created and built using Maven.
- The final .war file was generated in the target/ directory.

Step 2: Dockerize the Application

- A Dockerfile was written using the official Tomcat base image.
- The .war file was copied into the Tomcat webapps/ directory.

Dockerfile

```
FROM tomcat:latest

RUN cp -R /usr/local/tomcat/webapps.dist/* /usr/local/tomcat/webapps

COPY webapp/webapp/target/webapp.war /usr/local/tomcat/webapps/
```

- The image was built and tagged as amand0125/hello-world.

Step 3: Set Up Jenkins Pipeline

- Jenkins was configured on an EC2 Ubuntu instance.
- A pipeline was created to:
  - Pull code from GitHub
  - Build the Docker image
  - Push it to Docker Hub
  - Deploy it to Kubernetes using kubectl apply
- The Jenkins pipeline ran successfully, indicating all steps completed without errors.

Step 4: Create Kubernetes Manifests

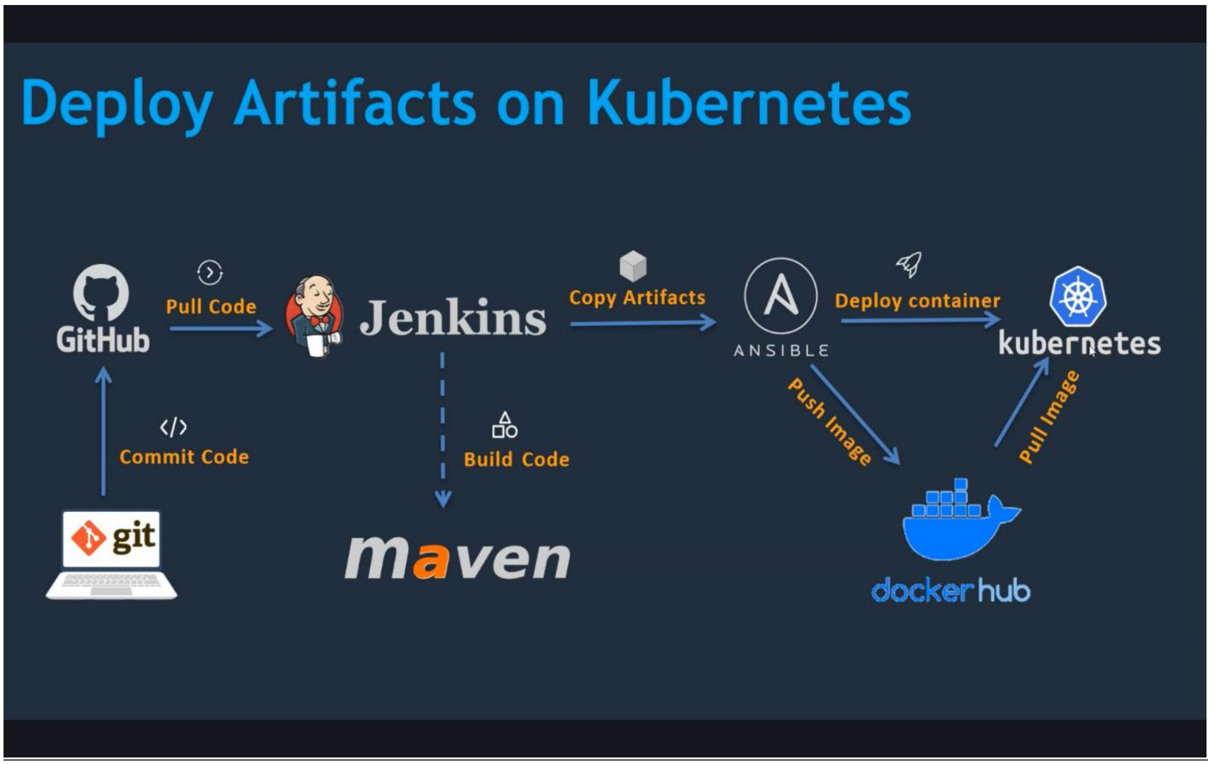
- A deploy.yml file was used to define the Deployment for the application.
- A regapp-service.yml file exposed the app using a NodePort service:

```
apiVersion: v1
kind: Service
metadata:
 name: regapp-service
spec:
 type: NodePort
 selector:
  app: valaxy-regapp
 ports:
  - protocol: TCP
   port: 8080
   targetPort: 8080
   nodePort: 30080
```

Step 5: Deploy and Debug via Minikube

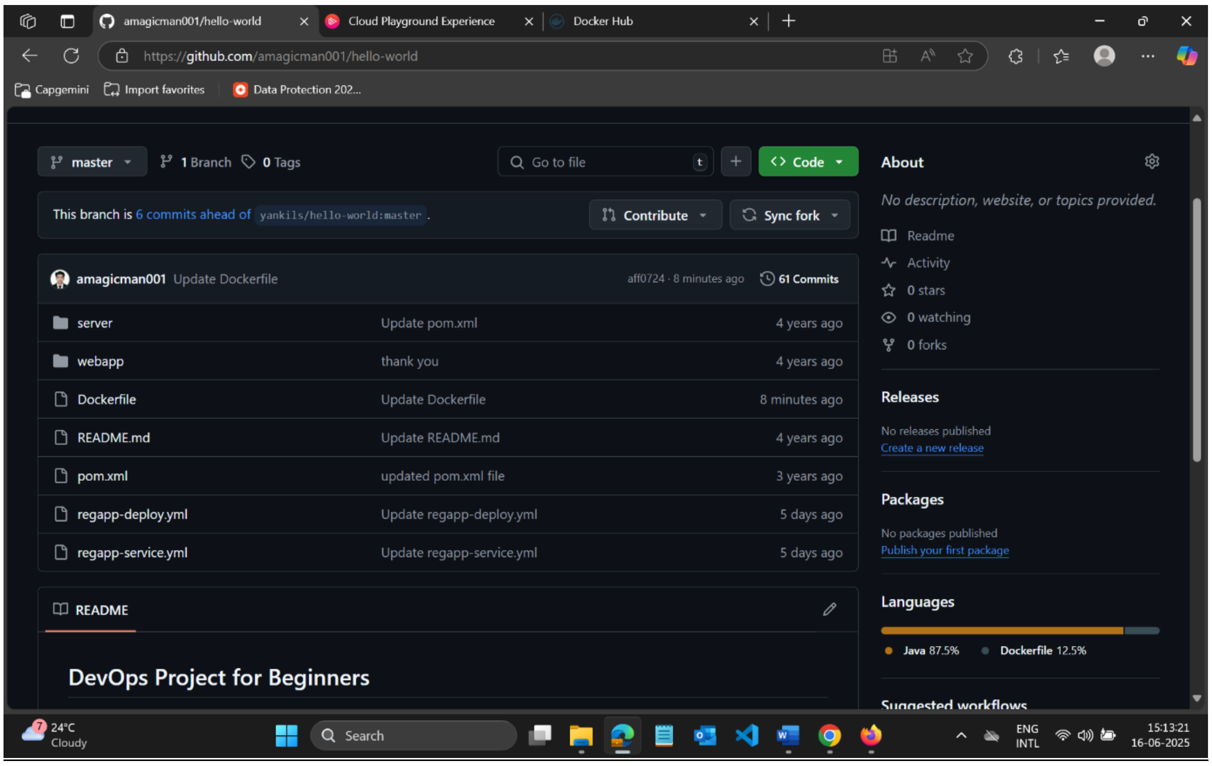
- The image was deployed to Minikube using kubectl.
- Issues were faced accessing the app via minikube service, which were debugged using:
    - kubectl get all
    - Checking logs
    - Entering the container to validate the deployed WAR
- Eventually, the app was successfully accessible at http://kubernetes-public-Ip:30080.

Stepwise Screenshots Demonstration of the Whole Project



- Infrastructure of the whole Project

The GitHub repository which contains the Project.



amagicman001/hello-world

https://github.com/amagicman001/hello-world

master · 1 Branch · 0 Tags

This branch is 6 commits ahead of `yankils/hello-world:master`.

amagicman001 Update Dockerfile — aff0724 · 8 minutes ago — 61 Commits

server	Update pom.xml	4 years ago
webapp	thank you	4 years ago
Dockerfile	Update Dockerfile	8 minutes ago
README.md	Update README.md	4 years ago
pom.xml	updated pom.xml file	3 years ago
regapp-deploy.yml	Update regapp-deploy.yml	5 days ago
regapp-service.yml	Update regapp-service.yml	5 days ago

README

DevOps Project for Beginners

About

No description, website, or topics provided.

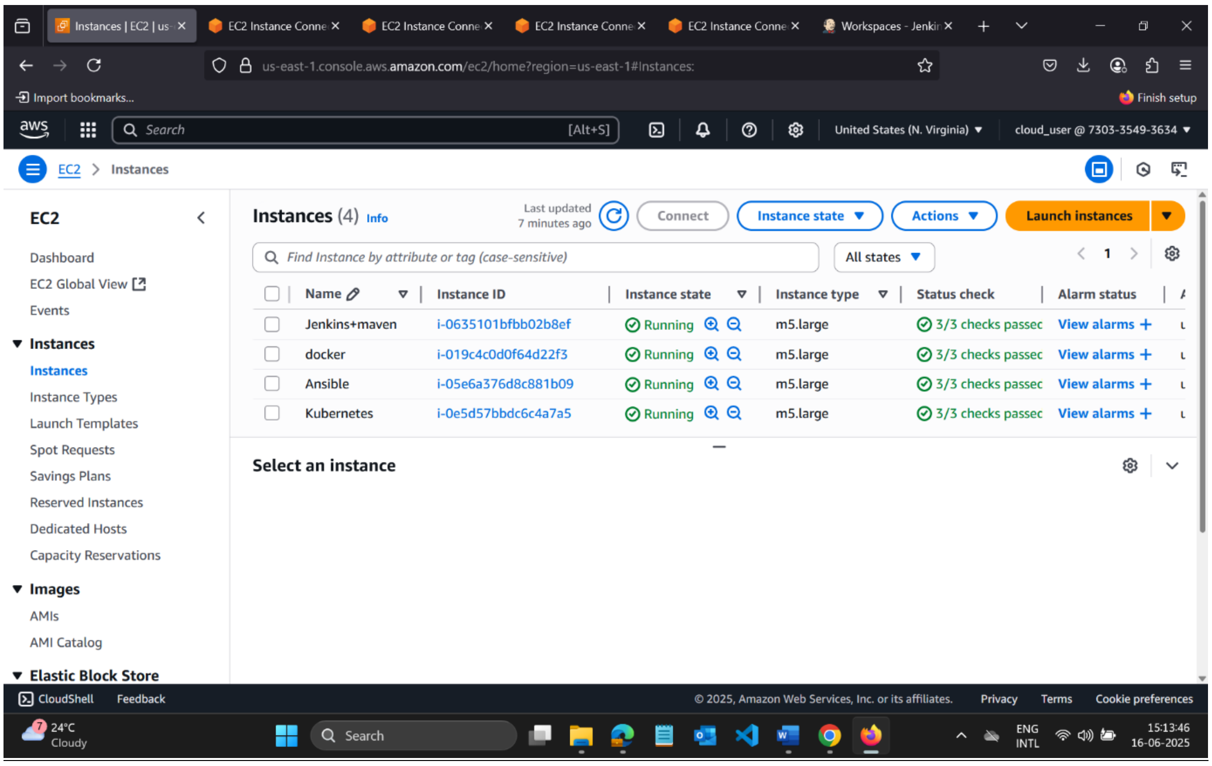
Readme · Activity · 0 stars · 0 watching · 0 forks

Releases: No releases published

Packages: No packages published

Languages: Java 87.5% · Dockerfile 12.5%

Instances created on Aws for Jenkins+maven,docker,Ansible



us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#Instances:

United States (N. Virginia) · cloud_user @ 7303-3549-3634

EC2 > Instances

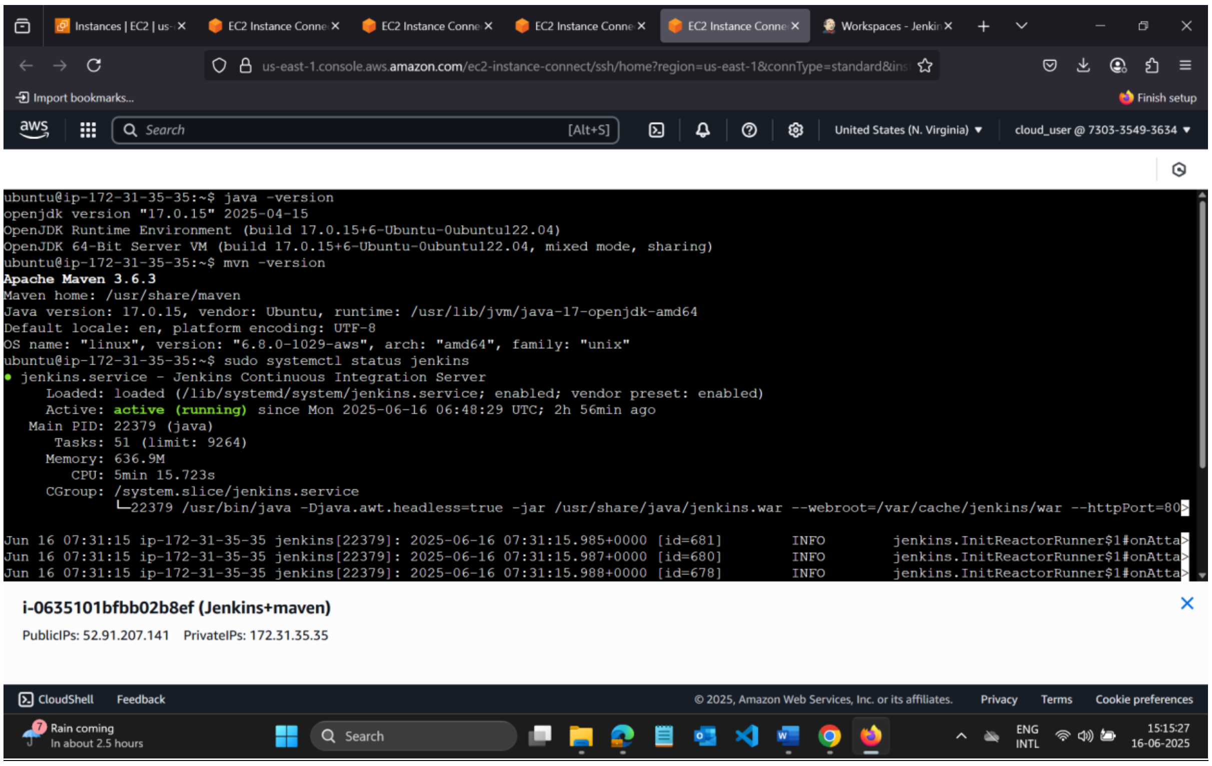
Instances (4) — Last updated 7 minutes ago

Name	Instance ID	Instance state	Instance type	Status check	Alarm status
Jenkins+maven	i-0635101bfbb02b8ef	Running	m5.large	3/3 checks passec	View alarms +
docker	i-019c4c0d0f64d22f3	Running	m5.large	3/3 checks passec	View alarms +
Ansible	i-05e6a376d8c881b09	Running	m5.large	3/3 checks passec	View alarms +
Kubernetes	i-0e5d57bbdc6c4a7a5	Running	m5.large	3/3 checks passec	View alarms +

Select an instance

© 2025, Amazon Web Services, Inc. or its affiliates.

On Jenkins and maven instance install javaopenjdk-17, Maven, Jenkins and Docker. Set the inbound rule to custom TCP 8080 along with that store .pem file which was earlier created in /home/ubuntu/prj.pem

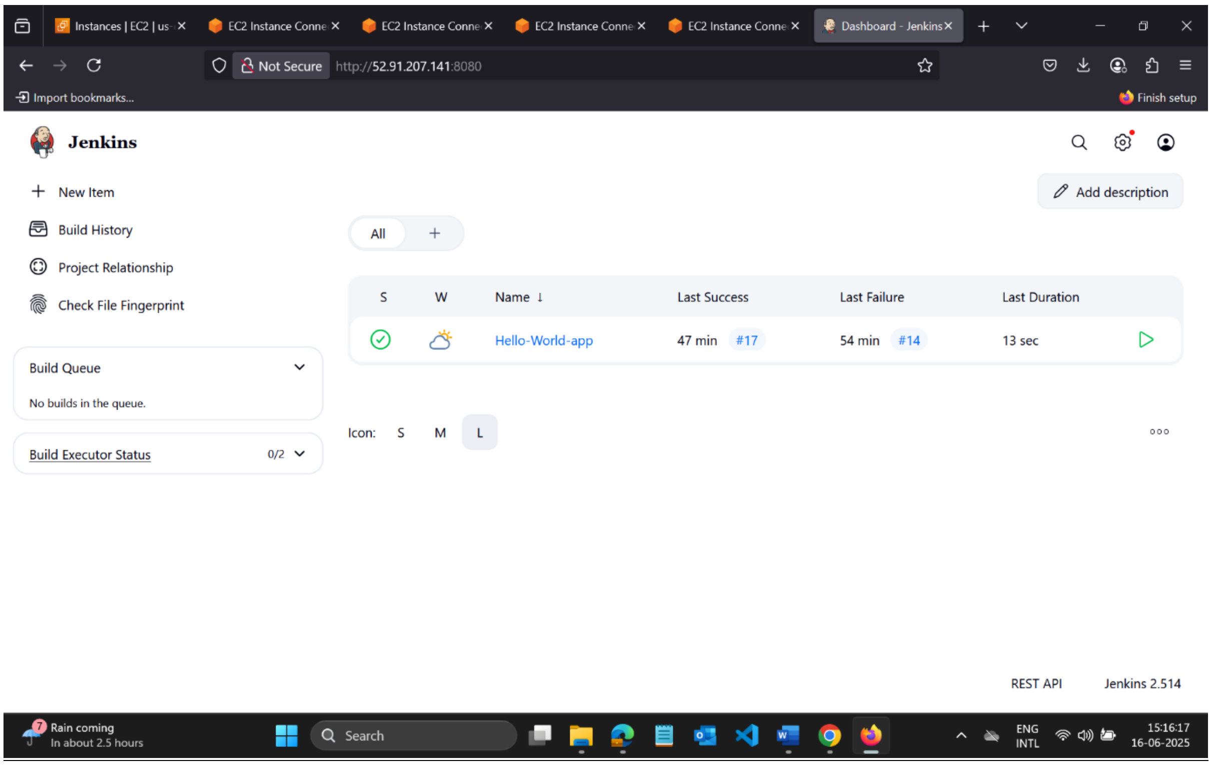


```
ubuntu@ip-172-31-35-35:~$ java -version
openjdk version "17.0.15" 2025-04-15
OpenJDK Runtime Environment (build 17.0.15+6-Ubuntu-0ubuntu122.04)
OpenJDK 64-Bit Server VM (build 17.0.15+6-Ubuntu-0ubuntu122.04, mixed mode, sharing)
ubuntu@ip-172-31-35-35:~$ mvn -version
Apache Maven 3.6.3
Maven home: /usr/share/maven
Java version: 17.0.15, vendor: Ubuntu, runtime: /usr/lib/jvm/java-17-openjdk-amd64
Default locale: en, platform encoding: UTF-8
OS name: "linux", version: "6.8.0-1029-aws", arch: "amd64", family: "unix"
ubuntu@ip-172-31-35-35:~$ sudo systemctl status jenkins
● jenkins.service - Jenkins Continuous Integration Server
     Loaded: loaded (/lib/systemd/system/jenkins.service; enabled; vendor preset: enabled)
     Active: active (running) since Mon 2025-06-16 06:48:29 UTC; 2h 56min ago
   Main PID: 22379 (java)
      Tasks: 51 (limit: 9264)
     Memory: 636.9M
        CPU: 5min 15.723s
     CGroup: /system.slice/jenkins.service
             └─22379 /usr/bin/java -Djava.awt.headless=true -jar /usr/share/java/jenkins.war --webroot=/var/cache/jenkins/war --httpPort=80>

Jun 16 07:31:15 ip-172-31-35-35 jenkins[22379]: 2025-06-16 07:31:15.985+0000 [id=681]        INFO       jenkins.InitReactorRunner$1#onAtta>
Jun 16 07:31:15 ip-172-31-35-35 jenkins[22379]: 2025-06-16 07:31:15.987+0000 [id=680]        INFO       jenkins.InitReactorRunner$1#onAtta>
Jun 16 07:31:15 ip-172-31-35-35 jenkins[22379]: 2025-06-16 07:31:15.988+0000 [id=678]        INFO       jenkins.InitReactorRunner$1#onAtta>
```

i-0635101bfbb02b8ef (Jenkins+maven)

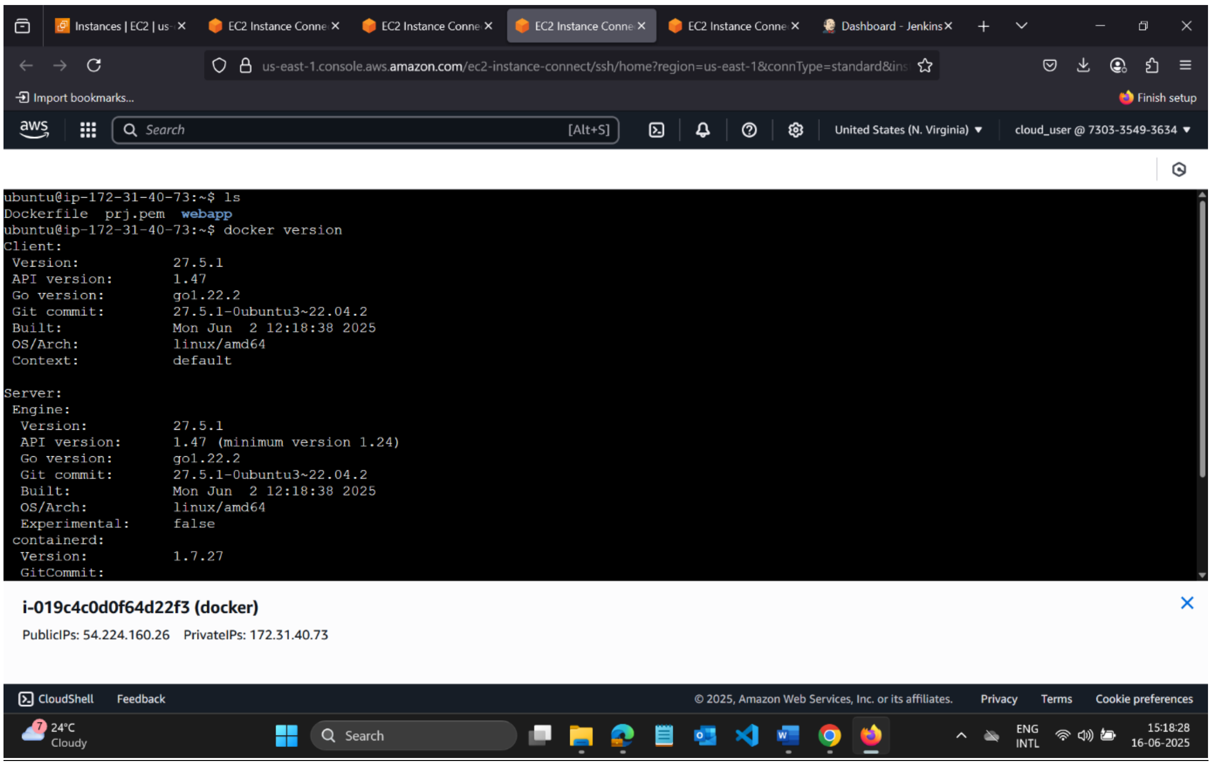
PublicIPs: 52.91.207.141   PrivateIPs: 172.31.35.35

Get the Jenkins dashboard at port 8080 using Jenkins Configure Jenkins by Installing GitHub plugin, ssh Agent plugin, Docker API,Commons,Pipeline.



S	W	Name ↓	Last Success	Last Failure	Last Duration
✓		Hello-World-app	47 min #17	54 min #14	13 sec

On the docker instance install docker and login to you remote docker hub along with that store the .pem file which was earlier created in /home/ubuntu/prj.pem



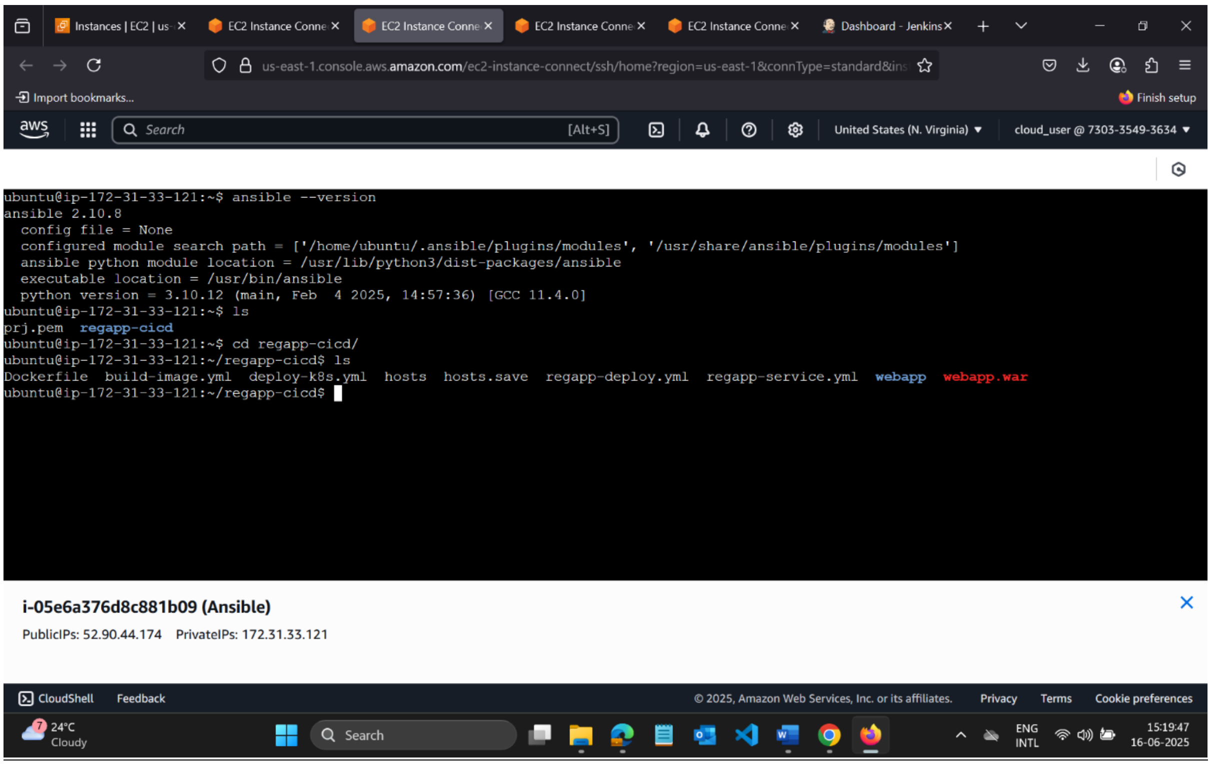
us-east-1.console.aws.amazon.com/ec2-instance-connect/ssh/home?region=us-east-1&connType=standard&in
United States (N. Virginia)
cloud_user @ 7303-3549-3634

```
ubuntu@ip-172-31-40-73:~$ ls
Dockerfile  prj.pem  webapp
ubuntu@ip-172-31-40-73:~$ docker version
Client:
 Version:           27.5.1
 API version:       1.47
 Go version:        go1.22.2
 Git commit:        27.5.1-0ubuntu3~22.04.2
 Built:             Mon Jun  2 12:18:38 2025
 OS/Arch:           linux/amd64
 Context:           default

Server:
 Engine:
  Version:          27.5.1
  API version:      1.47 (minimum version 1.24)
  Go version:       go1.22.2
  Git commit:       27.5.1-0ubuntu3~22.04.2
  Built:            Mon Jun  2 12:18:38 2025
  OS/Arch:          linux/amd64
  Experimental:     false
 containerd:
  Version:          1.7.27
  GitCommit:
```

i-019c4c0d0f64d22f3 (docker)

PublicIPs: 54.224.160.26 PrivateIPs: 172.31.40.73

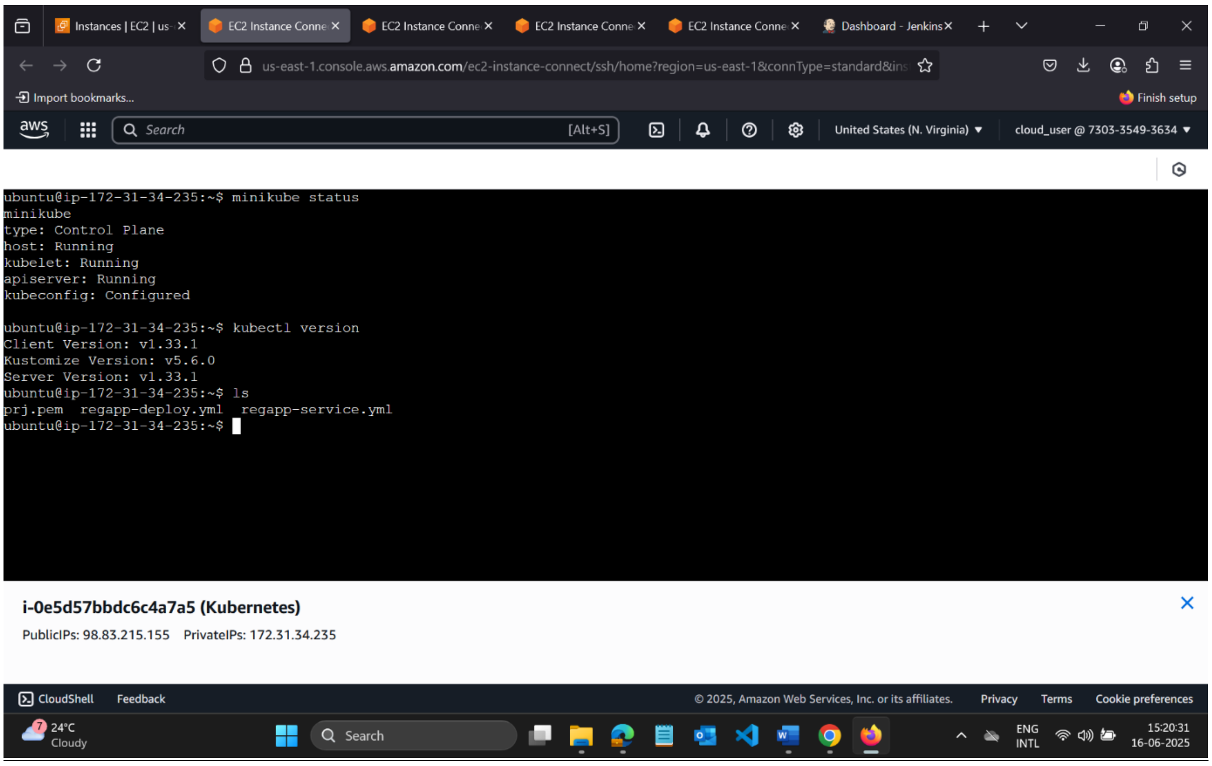
Install Ansible at your ansible instance along with that store .pem file which was earlier created in /home/ubuntu/prj.pem



```
ubuntu@ip-172-31-33-121:~$ ansible --version
ansible 2.10.8
  config file = None
  configured module search path = ['/home/ubuntu/.ansible/plugins/modules', '/usr/share/ansible/plugins/modules']
  ansible python module location = /usr/lib/python3/dist-packages/ansible
  executable location = /usr/bin/ansible
  python version = 3.10.12 (main, Feb  4 2025, 14:57:36) [GCC 11.4.0]
ubuntu@ip-172-31-33-121:~$ ls
prj.pem  regapp-cicd
ubuntu@ip-172-31-33-121:~$ cd regapp-cicd/
ubuntu@ip-172-31-33-121:~/regapp-cicd$ ls
Dockerfile  build-image.yml  deploy-k8s.yml  hosts  hosts.save  regapp-deploy.yml  regapp-service.yml  webapp  webapp.war
ubuntu@ip-172-31-33-121:~/regapp-cicd$
```

i-05e6a376d8c881b09 (Ansible)
PublicIPs: 52.90.44.174   PrivateIPs: 172.31.33.121

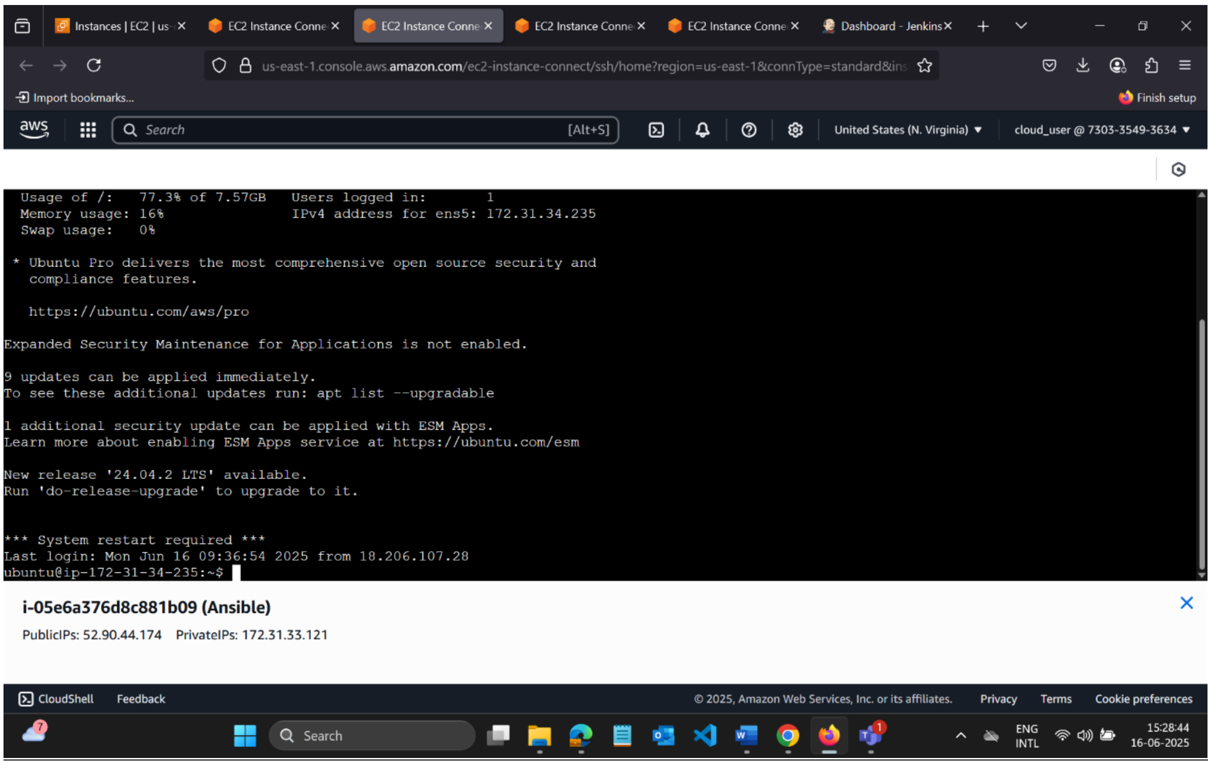
Install Minikube & Kubectl at the Kubernetes instance with storing the .pem file at /home/ubuntu/prj.pem



```
ubuntu@ip-172-31-34-235:~$ minikube status
minikube
type: Control Plane
host: Running
kubelet: Running
apiserver: Running
kubeconfig: Configured

ubuntu@ip-172-31-34-235:~$ kubectl version
Client Version: v1.33.1
Kustomize Version: v5.6.0
Server Version: v1.33.1
ubuntu@ip-172-31-34-235:~$ ls
prj.pem  regapp-deploy.yml  regapp-service.yml
ubuntu@ip-172-31-34-235:~$
```

i-0e5d57bbdc6c4a7a5 (Kubernetes)
PublicIPs: 98.83.215.155   PrivateIPs: 172.31.34.235

Perform SSH from Ansible to Docker and Kubernetes Instance.

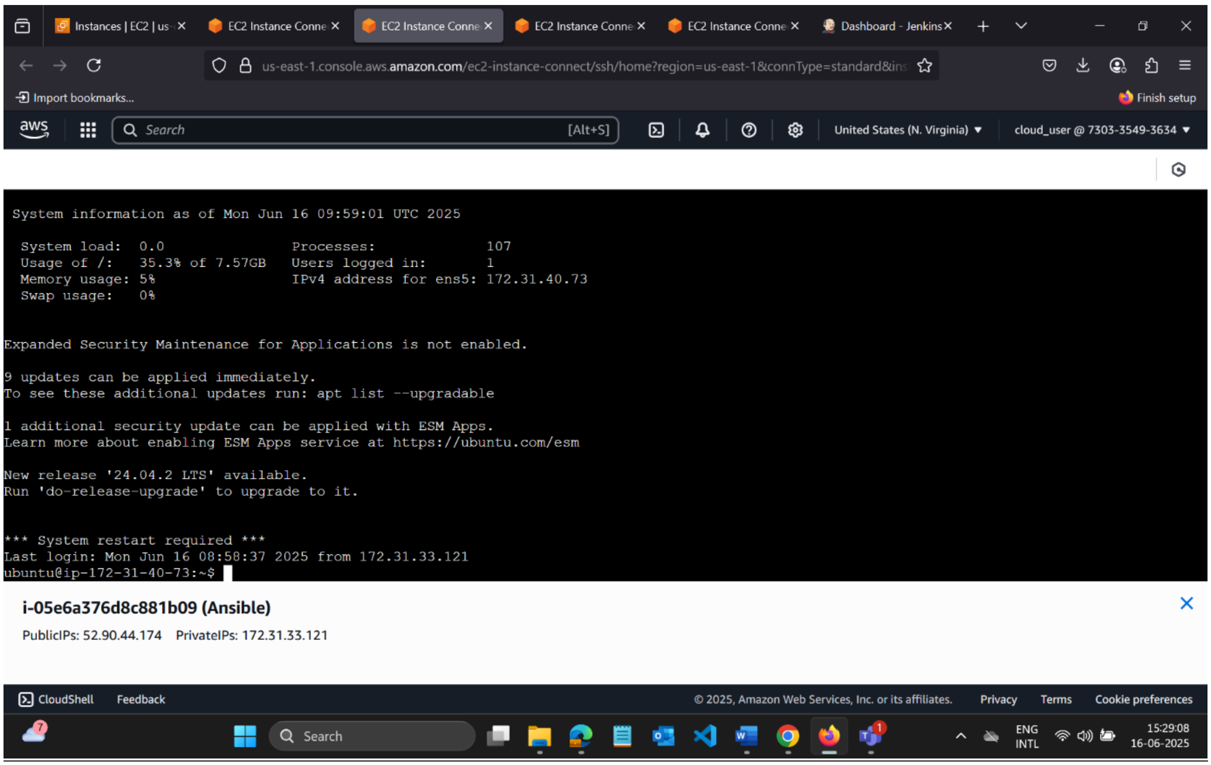


```
 Usage of /:   77.3% of 7.57GB   Users logged in:       1
 Memory usage: 16%               IPv4 address for ens5: 172.31.34.235
 Swap usage:   0%

 * Ubuntu Pro delivers the most comprehensive open source security and
   compliance features.

   https://ubuntu.com/aws/pro

Expanded Security Maintenance for Applications is not enabled.

9 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

1 additional security update can be applied with ESM Apps.
Learn more about enabling ESM Apps service at https://ubuntu.com/esm

New release '24.04.2 LTS' available.
Run 'do-release-upgrade' to upgrade to it.


*** System restart required ***
Last login: Mon Jun 16 09:36:54 2025 from 18.206.107.28
ubuntu@ip-172-31-34-235:~$
```

i-05e6a376d8c881b09 (Ansible)

PublicIPs: 52.90.44.174   PrivateIPs: 172.31.33.121

Instances | EC2 | us
EC2 Instance Conne
Dashboard - Jenkins
us-east-1.console.aws.amazon.com/ec2-instance-connect/ssh/home?region=us-east-1&connType=standard&ins
Import bookmarks...
Finish setup
Search
[Alt+S]
United States (N. Virginia)
cloud_user @ 7303-3549-3634
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
ENG INTL
15:28:44
16-06-2025



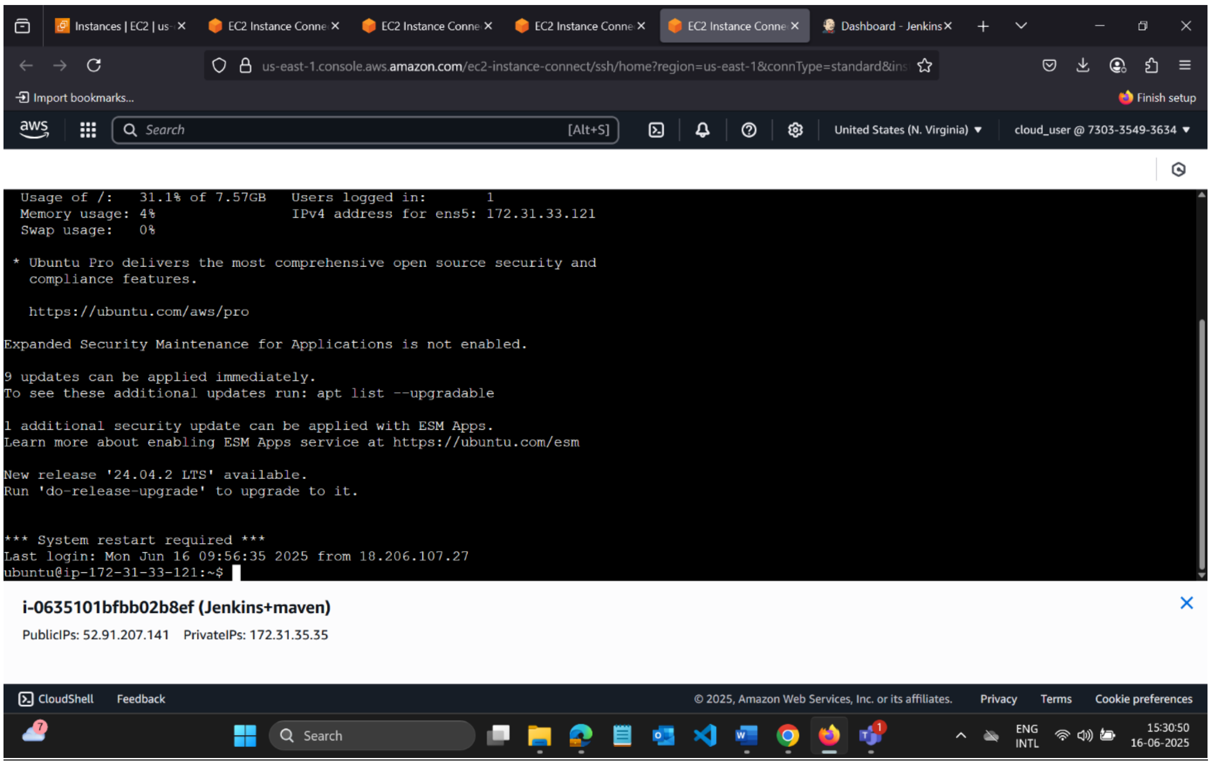
```
 System information as of Mon Jun 16 09:59:01 UTC 2025

  System load:  0.0               Processes:             107
  Usage of /:   35.3% of 7.57GB   Users logged in:       1
  Memory usage: 5%                IPv4 address for ens5: 172.31.40.73
  Swap usage:   0%


Expanded Security Maintenance for Applications is not enabled.

9 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

1 additional security update can be applied with ESM Apps.
Learn more about enabling ESM Apps service at https://ubuntu.com/esm

New release '24.04.2 LTS' available.
Run 'do-release-upgrade' to upgrade to it.


*** System restart required ***
Last login: Mon Jun 16 08:58:37 2025 from 172.31.33.121
ubuntu@ip-172-31-40-73:~$
```

i-05e6a376d8c881b09 (Ansible)

PublicIPs: 52.90.44.174   PrivateIPs: 172.31.33.121

Instances | EC2 | us
EC2 Instance Conne
Dashboard - Jenkins
us-east-1.console.aws.amazon.com/ec2-instance-connect/ssh/home?region=us-east-1&connType=standard&ins
Import bookmarks...
Finish setup
Search
[Alt+S]
United States (N. Virginia)
cloud_user @ 7303-3549-3634
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
ENG INTL
15:29:08
16-06-2025

Maintain the SSH Connection between the Jenkins and Ansible



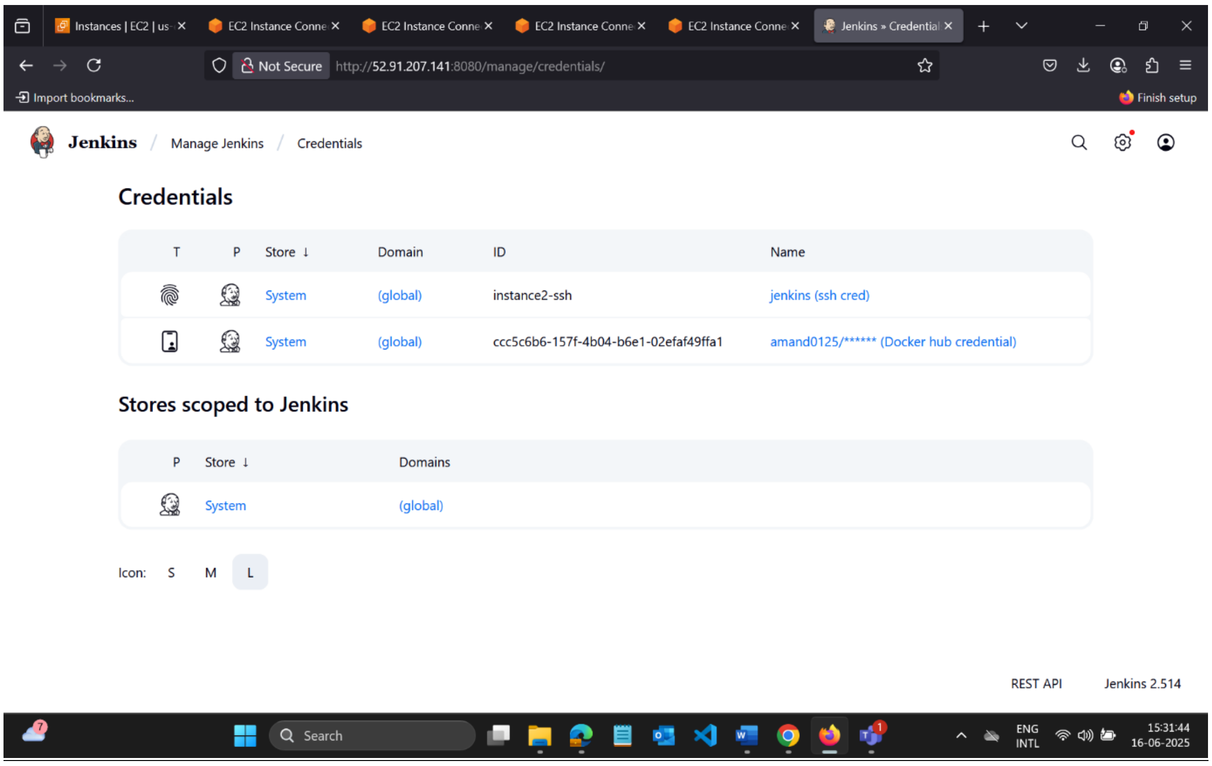
```
  Usage of /:   31.1% of 7.57GB   Users logged in:        1
  Memory usage: 4%                IPv4 address for ens5: 172.31.33.121
  Swap usage:   0%

 * Ubuntu Pro delivers the most comprehensive open source security and
   compliance features.

   https://ubuntu.com/aws/pro

Expanded Security Maintenance for Applications is not enabled.

9 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

1 additional security update can be applied with ESM Apps.
Learn more about enabling ESM Apps service at https://ubuntu.com/esm

New release '24.04.2 LTS' available.
Run 'do-release-upgrade' to upgrade to it.


*** System restart required ***
Last login: Mon Jun 16 09:56:35 2025 from 18.206.107.27
ubuntu@ip-172-31-33-121:~$
```

i-0635101bfbb02b8ef (Jenkins+maven)

PublicIPs: 52.91.207.141   PrivateIPs: 172.31.35.35

us-east-1.console.aws.amazon.com/ec2-instance-connect/ssh/home?region=us-east-1&connType=standard&ins
United States (N. Virginia)
cloud_user @ 7303-3549-3634
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
15:30:50
16-06-2025

Create credentials for ssh and docker in order to be used in the pipeline as environment variables.



Jenkins / Manage Jenkins / Credentials

Credentials

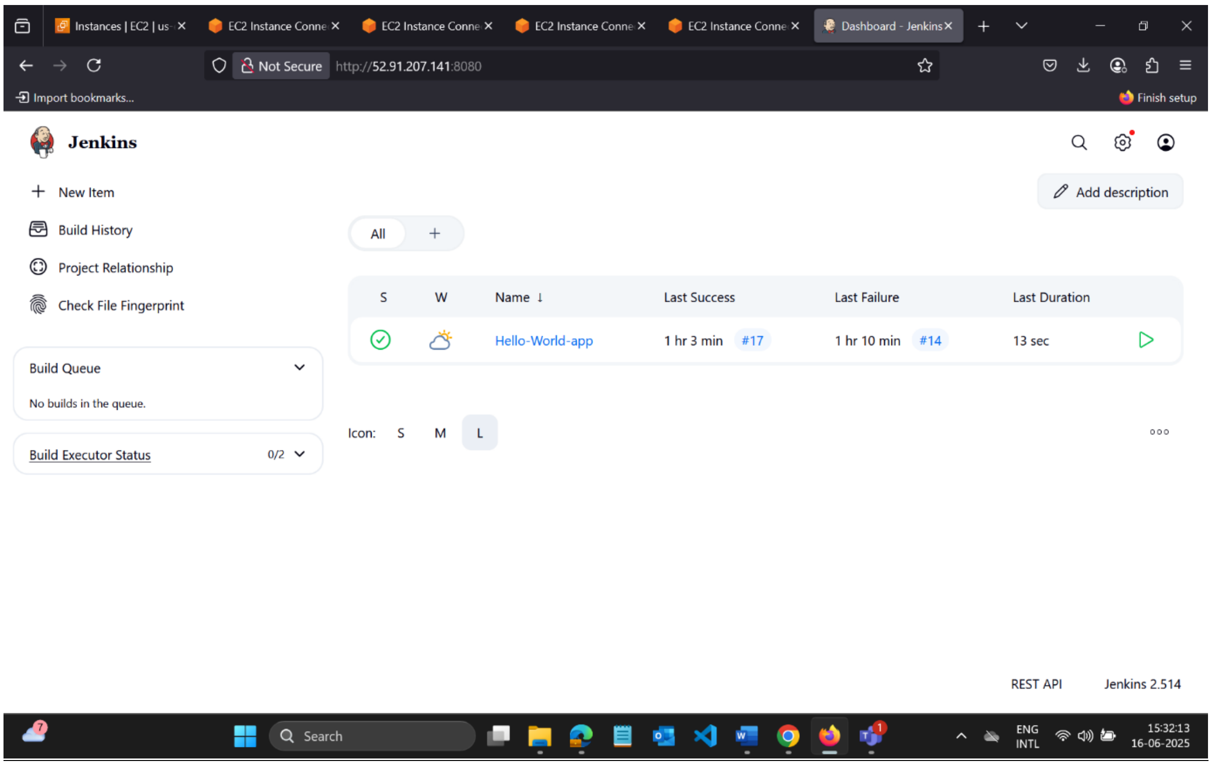
T	P	Store ↓	Domain	ID	Name
		System	(global)	instance2-ssh	jenkins (ssh cred)
		System	(global)	ccc5c6b6-157f-4b04-b6e1-02efaf49ffa1	amand0125/****** (Docker hub credential)

Stores scoped to Jenkins

P	Store ↓	Domains
	System	(global)

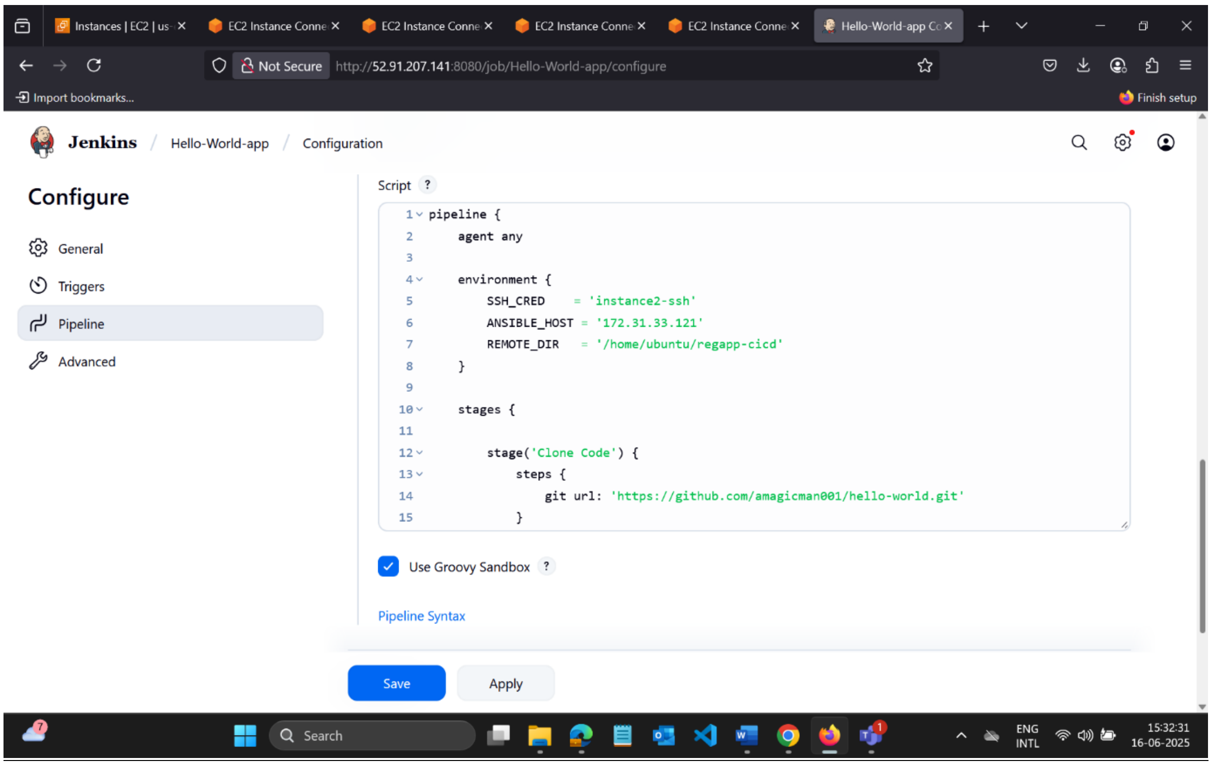
Icon: S M L

In the Jenkins Create a Declarative Pipeline Job named Hello-World-app which will orchestrate the whole project deployment



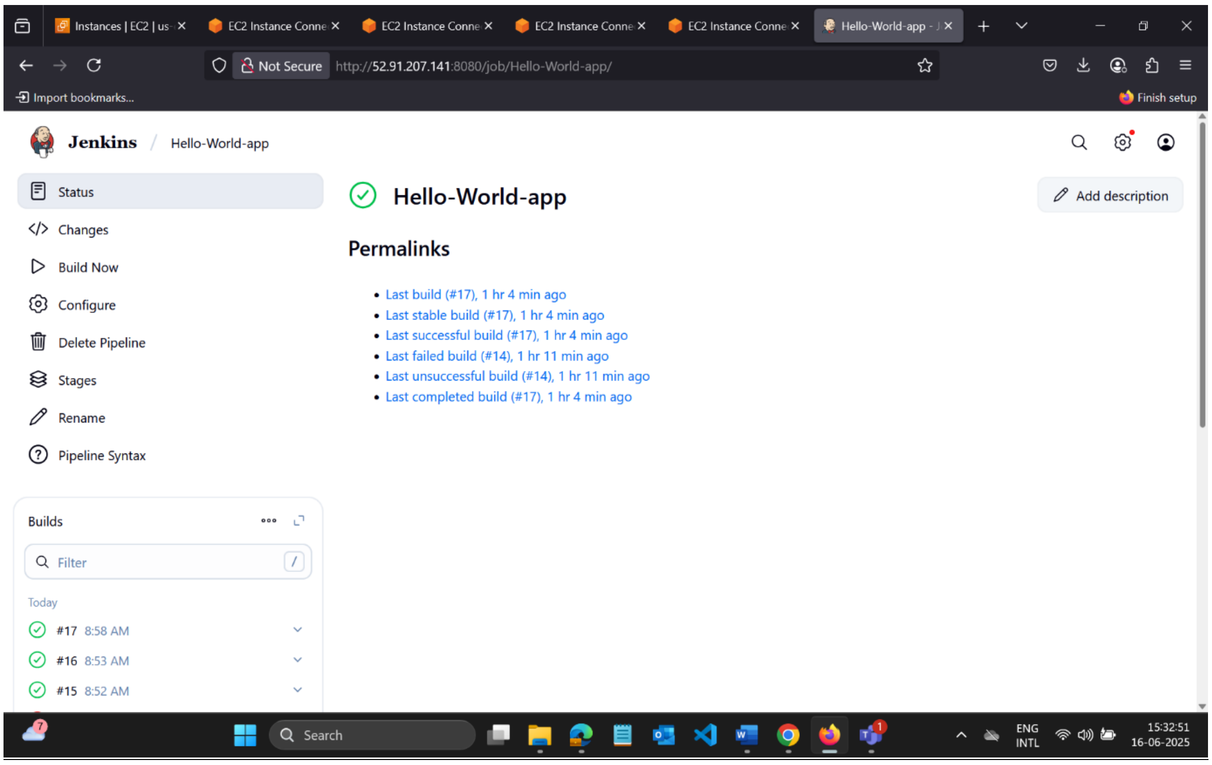
S	W	Name ↓	Last Success	Last Failure	Last Duration
✓		Hello-World-app	1 hr 3 min #17	1 hr 10 min #14	13 sec

This the Pipeline for the job. *Check end of ss for detailed pipeline



```
pipeline {
    agent any

    environment {
        SSH_CRED    = 'instance2-ssh'
        ANSIBLE_HOST = '172.31.33.121'
        REMOTE_DIR   = '/home/ubuntu/regapp-cicd'
    }

    stages {

        stage('Clone Code') {
            steps {
                git url: 'https://github.com/amagicman001/hello-world.git'
            }
```

The Pipeline job got build Sucessfully



http://52.91.207.141:8080/job/Hello-World-app/

Jenkins / Hello-World-app

Status
Changes
Build Now
Configure
Delete Pipeline
Stages
Rename
Pipeline Syntax

Hello-World-app

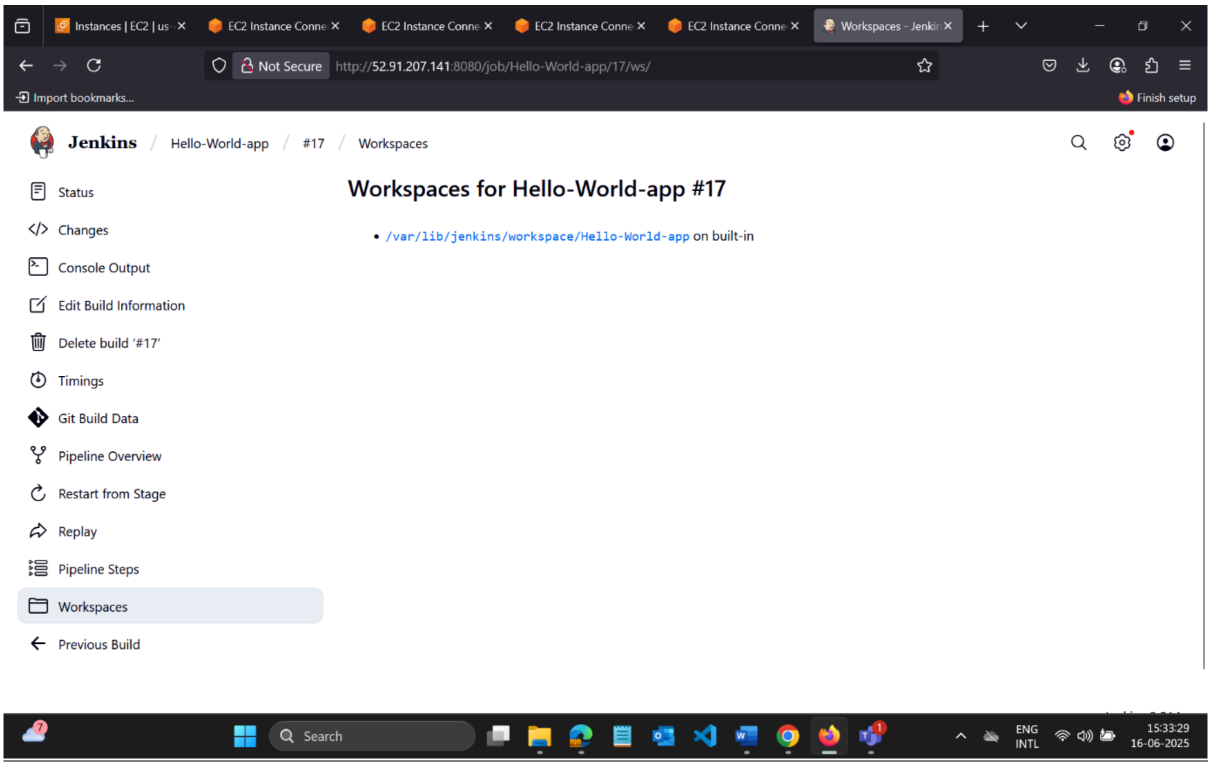
Permalinks

- Last build (#17), 1 hr 4 min ago
- Last stable build (#17), 1 hr 4 min ago
- Last successful build (#17), 1 hr 4 min ago
- Last failed build (#14), 1 hr 11 min ago
- Last unsuccessful build (#14), 1 hr 11 min ago
- Last completed build (#17), 1 hr 4 min ago

Builds

Today
#17 8:58 AM
#16 8:53 AM
#15 8:52 AM

This is the Jenkins workspace where the .war build file is stored



http://52.91.207.141:8080/job/Hello-World-app/17/ws/

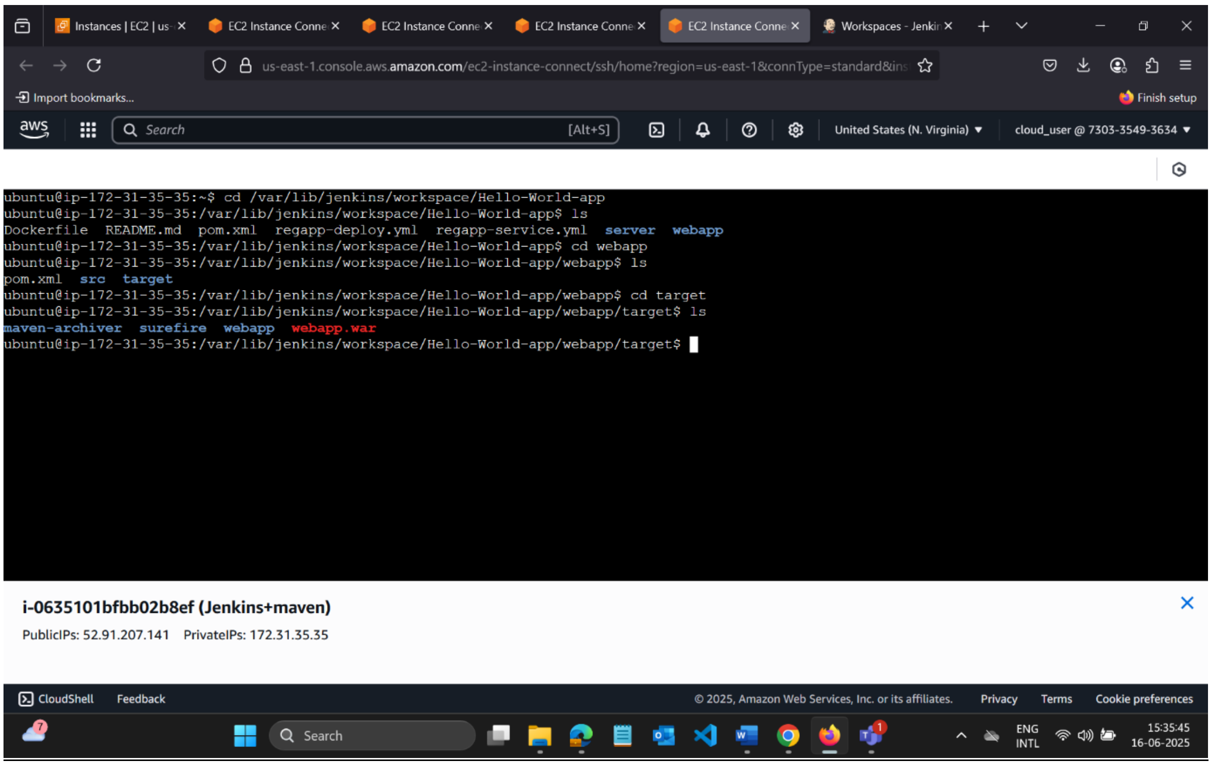
Jenkins / Hello-World-app / #17 / Workspaces

Status
Changes
Console Output
Edit Build Information
Delete build '#17'
Timings
Git Build Data
Pipeline Overview
Restart from Stage
Replay
Pipeline Steps
Workspaces
Previous Build

Workspaces for Hello-World-app #17

- /var/lib/jenkins/workspace/Hello-World-app on built-in

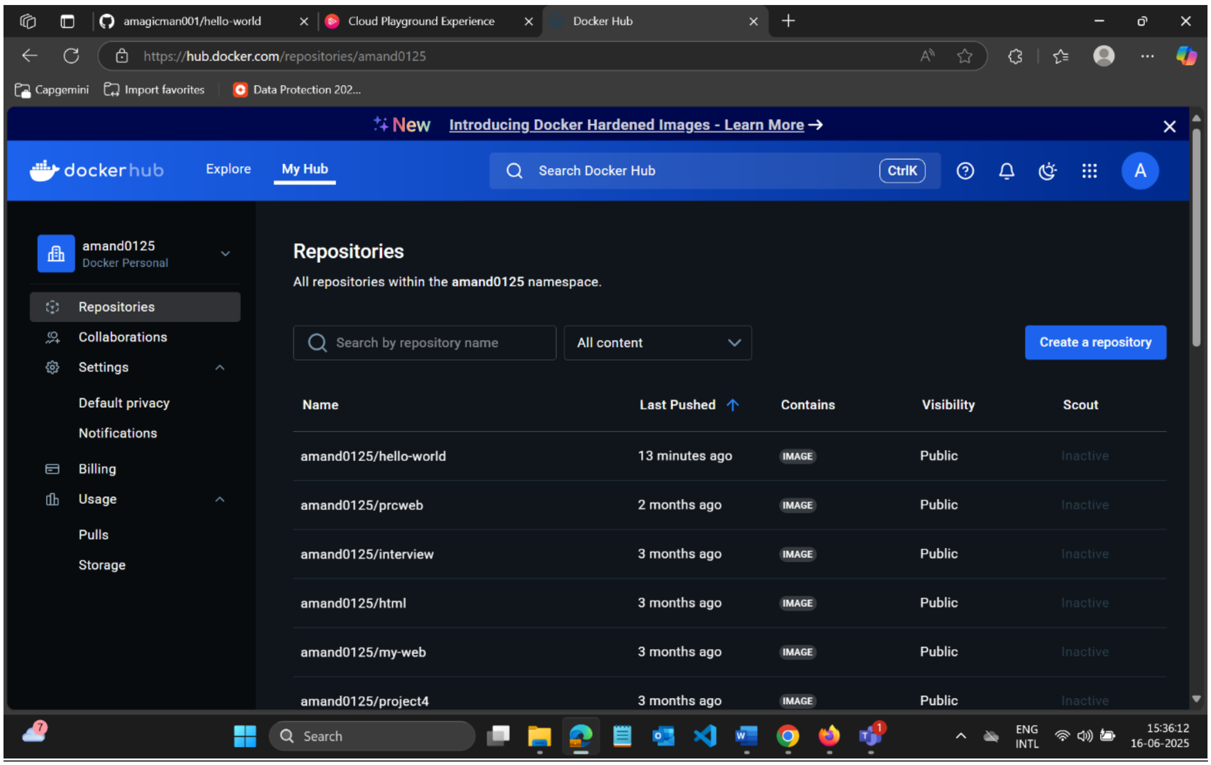
The webapp directory at Jenkins home contains the .war file.



```
ubuntu@ip-172-31-35-35:~$ cd /var/lib/jenkins/workspace/Hello-World-app
ubuntu@ip-172-31-35-35:/var/lib/jenkins/workspace/Hello-World-app$ ls
Dockerfile  README.md  pom.xml  regapp-deploy.yml  regapp-service.yml  server  webapp
ubuntu@ip-172-31-35-35:/var/lib/jenkins/workspace/Hello-World-app$ cd webapp
ubuntu@ip-172-31-35-35:/var/lib/jenkins/workspace/Hello-World-app/webapp$ ls
pom.xml  src  target
ubuntu@ip-172-31-35-35:/var/lib/jenkins/workspace/Hello-World-app/webapp$ cd target
ubuntu@ip-172-31-35-35:/var/lib/jenkins/workspace/Hello-World-app/webapp/target$ ls
maven-archiver  surefire  webapp  webapp.war
ubuntu@ip-172-31-35-35:/var/lib/jenkins/workspace/Hello-World-app/webapp/target$
```

i-0635101bfbb02b8ef (Jenkins+maven)
PublicIPs: 52.91.207.141 PrivateIPs: 172.31.35.35

After the successful Build of the pipeline job the docker is getting triggered via ansible which create the docker image containerizes the application and pushes it to docker hub

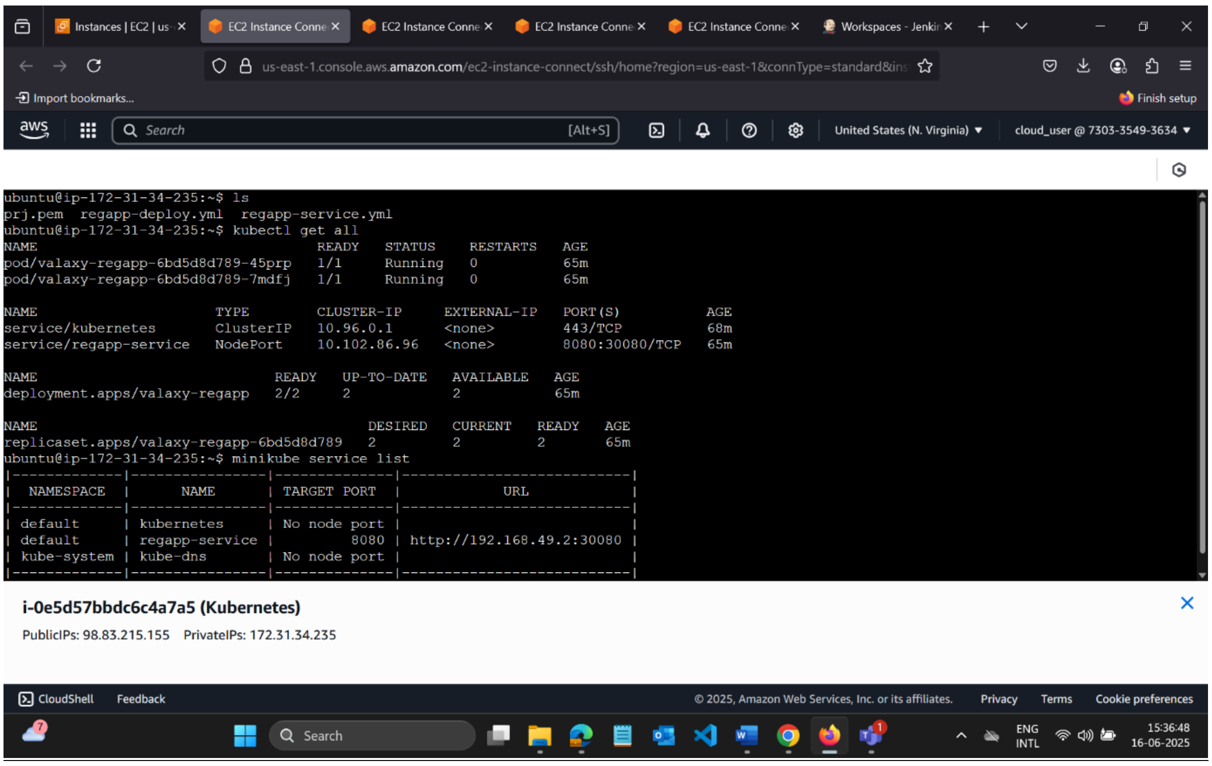


Repositories

All repositories within the amand0125 namespace.

Name	Last Pushed	Contains	Visibility	Scout
amand0125/hello-world	13 minutes ago	IMAGE	Public	Inactive
amand0125/prcweb	2 months ago	IMAGE	Public	Inactive
amand0125/interview	3 months ago	IMAGE	Public	Inactive
amand0125/html	3 months ago	IMAGE	Public	Inactive
amand0125/my-web	3 months ago	IMAGE	Public	Inactive
amand0125/project4	3 months ago	IMAGE	Public	Inactive

After the successful build of pipeline the ansible first triggers the docker instance and docker containerizes and pushes the application to docker hub later on the the Kubernetes instance is triggered and deployment takes place.

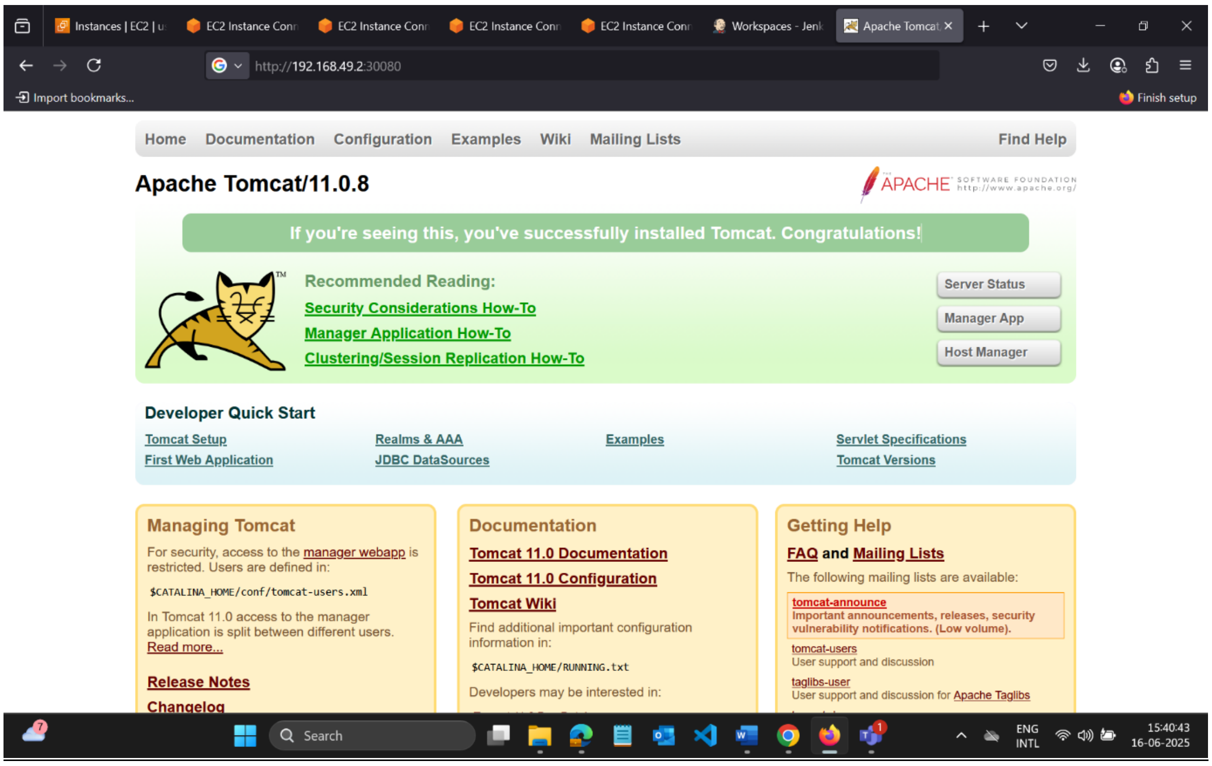


```
ubuntu@ip-172-31-34-235:~$ ls
prj.pem  regapp-deploy.yml  regapp-service.yml
ubuntu@ip-172-31-34-235:~$ kubectl get all
NAME                                  READY   STATUS    RESTARTS   AGE
pod/valaxy-regapp-6bd5d8d789-45prp   1/1     Running   0          65m
pod/valaxy-regapp-6bd5d8d789-7mdfj   1/1     Running   0          65m

NAME                      TYPE        CLUSTER-IP     EXTERNAL-IP   PORT(S)          AGE
service/kubernetes        ClusterIP   10.96.0.1      <none>        443/TCP          68m
service/regapp-service    NodePort    10.102.86.96   <none>        8080:30080/TCP   65m

NAME                            READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/valaxy-regapp   2/2     2            2           65m

NAME                                       DESIRED   CURRENT   READY   AGE
replicaset.apps/valaxy-regapp-6bd5d8d789   2         2         2       65m
ubuntu@ip-172-31-34-235:~$ minikube service list
|-------------|----------------|-------------|---------------------------|
|  NAMESPACE  |      NAME      | TARGET PORT |            URL            |
|-------------|----------------|-------------|---------------------------|
| default     | kubernetes     | No node port |                          |
| default     | regapp-service |        8080 | http://192.168.49.2:30080 |
| kube-system | kube-dns       | No node port |                          |
|-------------|----------------|-------------|---------------------------|
```

i-0e5d57bbdc6c4a7a5 (Kubernetes)

PublicIPs: 98.83.215.155   PrivateIPs: 172.31.34.235

Deployment of Tomcat Apache Server:


http://192.168.49.2:30080

Home Documentation Configuration Examples Wiki Mailing Lists Find Help

Apache Tomcat/11.0.8

If you're seeing this, you've successfully installed Tomcat. Congratulations!

Recommended Reading:
Security Considerations How-To
Manager Application How-To
Clustering/Session Replication How-To

Server Status
Manager App
Host Manager

Developer Quick Start
Tomcat Setup
First Web Application
Realms & AAA
JDBC DataSources
Examples
Servlet Specifications
Tomcat Versions

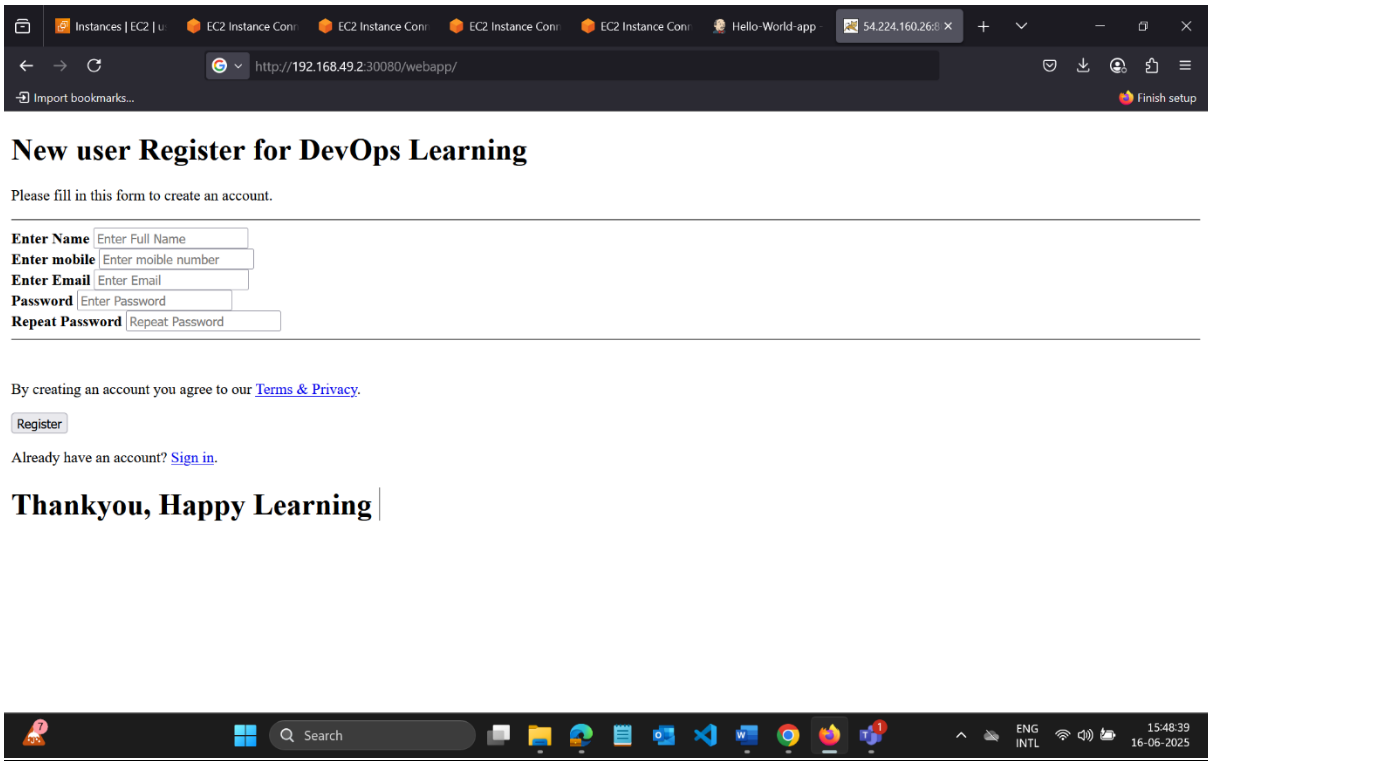
Managing Tomcat
For security, access to the manager webapp is restricted. Users are defined in:
```
$CATALINA_HOME/conf/tomcat-users.xml
```
In Tomcat 11.0 access to the manager application is split between different users. Read more...

Release Notes
Changelog

Documentation
Tomcat 11.0 Documentation
Tomcat 11.0 Configuration
Tomcat Wiki
Find additional important configuration information in:
```
$CATALINA_HOME/RUNNING.txt
```
Developers may be interested in:

Getting Help
FAQ and Mailing Lists
The following mailing lists are available:
tomcat-announce
Important announcements, releases, security vulnerability notifications. (Low volume).
tomcat-users
User support and discussion
taglibs-user
User support and discussion for Apache Taglibs

Accessing the deployed application at /webapp


http://192.168.49.2:30080/webapp/

New user Register for DevOps Learning

Please fill in this form to create an account.

Enter Name [Enter Full Name]
Enter mobile [Enter moible number]
Enter Email [Enter Email]
Password [Enter Password]
Repeat Password [Repeat Password]

By creating an account you agree to our Terms & Privacy.

Register

Already have an account? Sign in.

Thankyou, Happy Learning

Listing all the Ansible playbook files along with the Declarative Pipeline of the jenkins job used for project deployment

Pipeline:

```
pipeline {
  agent any

  environment {
    SSH_CRED    = 'instance2-ssh'
    ANSIBLE_HOST = '172.31.33.121'
    REMOTE_DIR   = '/home/ubuntu/regapp-cicd'
  }

  stages {

    stage('Clone Code') {
      steps {
        git url: 'https://github.com/amagicman001/hello-world.git'
      }
    }

    stage('Build WAR with Maven') {
      steps {
        sh 'mvn clean install'
      }
    }

    stage('Copy WAR and Dockerfile to Ansible') {
      steps {
        sshagent(credentials: [env.SSH_CRED]) {
          sh '''
            scp -o StrictHostKeyChecking=no \
                webapp/target/webapp.war \
                Dockerfile \
                ubuntu@${ANSIBLE_HOST}:${REMOTE_DIR}/
            '''
        }
      }
    }

    stage('Trigger Docker Build and Push via Ansible') {
      steps {
        sshagent(credentials: [env.SSH_CRED]) {
          sh """
            ssh ubuntu@${ANSIBLE_HOST} \
            'ansible-playbook ${REMOTE_DIR}/build-image.yml -i ${REMOTE_DIR}/hosts'
          """
        }
      }
    }

    stage('Trigger Kubernetes Deployment via Ansible') {
      steps {
        sshagent(credentials: [env.SSH_CRED]) {
          sh """
            ssh ubuntu@${ANSIBLE_HOST} \
            'ansible-playbook ${REMOTE_DIR}/deploy-k8s.yml -i ${REMOTE_DIR}/hosts'
          """
        }
      }
    }
  }
  post {
    success {
      echo 'Deployment Successful. Access your app via Kubernetes LoadBalancer!'
    }
    failure {
      echo ' Deployment Failed. Please check the logs.'
    }
  }
}
```

BUILD-IMAGE.YML:

```
---
- name: Build and Push Docker Image
  hosts: docker
  become: yes
  tasks:
    - name: Copy app files to Docker instance
      copy:
        src: /home/ubuntu/regapp-cicd/webapp
        dest: /home/ubuntu/webapp

    - name: Copy Dockerfile
      copy:
        src: /home/ubuntu/regapp-cicd/Dockerfile
        dest: /home/ubuntu/Dockerfile

    - name: Build Docker image
      shell: docker build -t amand0125/hello-world /home/ubuntu

    - name: Login to Docker Hub
      shell: echo "Aman@63863" | docker login -u "amand0125" --password-stdin
      args:
        executable: /bin/bash

    - name: Push Docker image to Docker Hub
      shell: docker push amand0125/hello-world
```

DEPLOY-K8S.YML

```
---
- name: Deploy App to Kubernetes
  hosts: k8s
  become: yes
  tasks:
   - name: Copy deployment YAML to Kubernetes server
     copy:
       src: /home/ubuntu/regapp-cicd/regapp-deploy.yml
       dest: /home/ubuntu/regapp-deploy.yml

   - name: Copy service YAML to Kubernetes server
     copy:
       src: /home/ubuntu/regapp-cicd/regapp-service.yml
       dest: /home/ubuntu/regapp-service.yml

   - name: Apply Deployment
     shell: kubectl apply -f /home/ubuntu/regapp-deploy.yml --kubeconfig /home/ubuntu/.kube/config

   - name: Apply Service
     shell: kubectl apply -f /home/ubuntu/regapp-service.yml --kubeconfig /home/ubuntu/.kube/config
```

```
HOSTS:

[docker]

172.31.40.73 ansible_user=ubuntu ansible_ssh_private_key_file=/home/ubuntu/prj.pem



[kubernetes]

172.31.34.235 ansible_user=ubuntu ansible_ssh_private_key_file=/home/ubuntu/prj.pem
```

Conclusion:

This project was a great hands-on experience in implementing a real-world CI/CD workflow using DevOps tools.

Key Learnings:

- Practical understanding of how Docker and Kubernetes integrate with Jenkins
- Experience building and deploying a Java application through automation
- Deep dive into how services work in Kubernetes, including NodePorts and access issues

Challenges Faced:

- Accessing the app via NodePort initially failed due to service misconfiguration and pod-image mismatches.
- kubectl get all showed no updates due to deployment YAML referencing old image or caching issues.
- Debugging involved checking logs inside containers and understanding how Tomcat loads WAR files.

How They Were Solved:

- Carefully updated the deploy.yml to use the correct image.
- Verified WAR deployment inside the container using kubectl exec.

- Cleaned up older services and re-applied manifests to ensure fresh deployments.