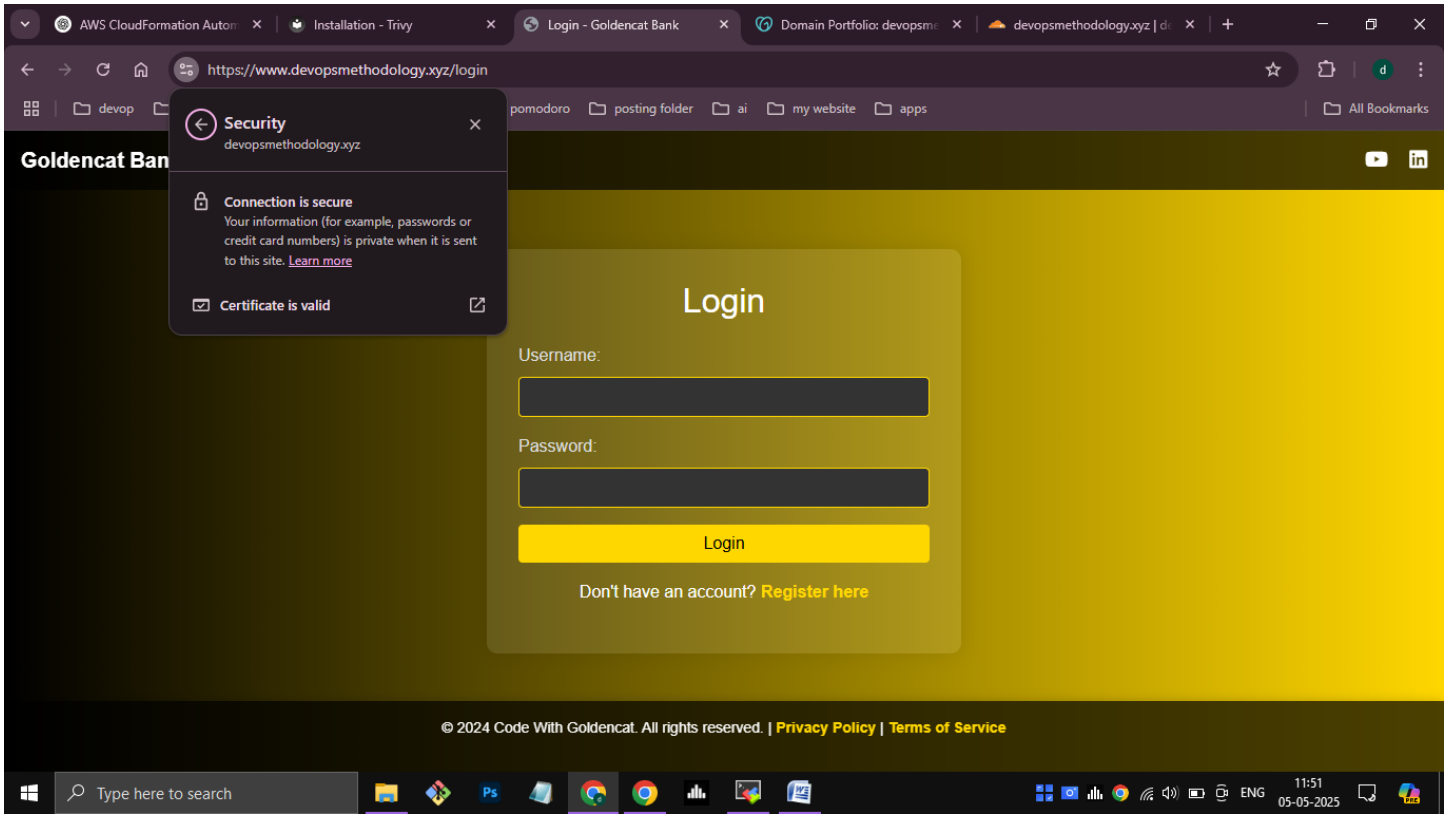
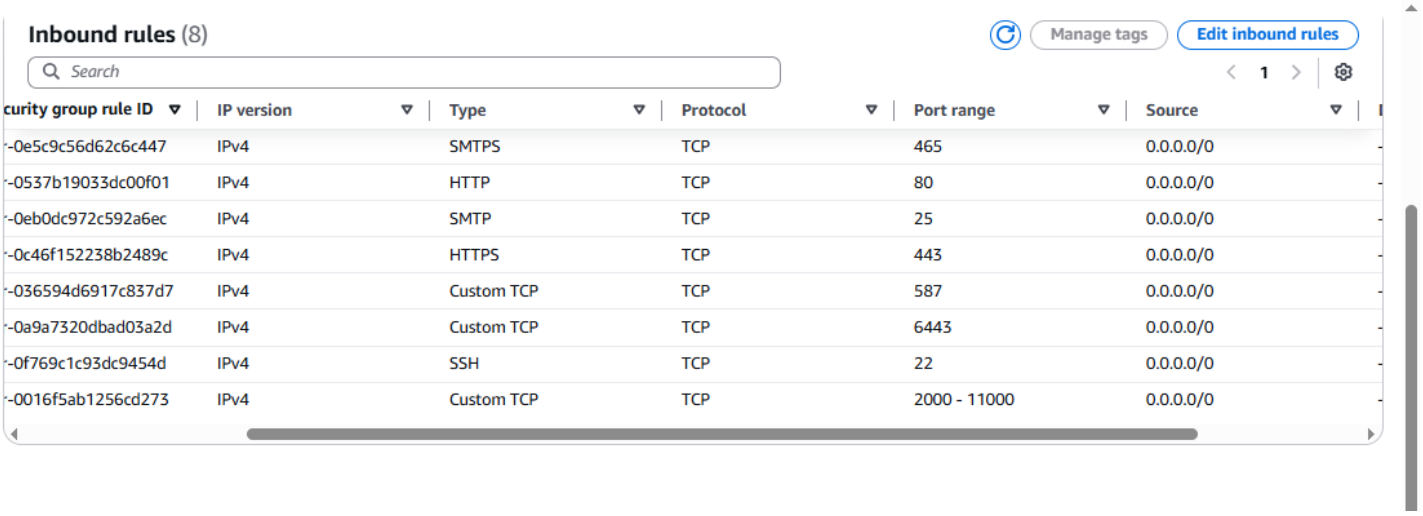
Multi-Tier CICD Project With SSL Certificate CICD DevOps Project



Port to be opened



Inbound rules (8)

Search

Manage tags | Edit inbound rules

1

curity group rule ID	IP version	Type	Protocol	Port range	Source
-0e5c9c56d62c6c447	IPv4	SMTPS	TCP	465	0.0.0.0/0
-0537b19033dc00f01	IPv4	HTTP	TCP	80	0.0.0.0/0
-0eb0dc972c592a6ec	IPv4	SMTP	TCP	25	0.0.0.0/0
-0c46f152238b2489c	IPv4	HTTPS	TCP	443	0.0.0.0/0
-036594d6917c837d7	IPv4	Custom TCP	TCP	587	0.0.0.0/0
-0a9a7320dbad03a2d	IPv4	Custom TCP	TCP	6443	0.0.0.0/0
-0f769c1c93dc9454d	IPv4	SSH	TCP	22	0.0.0.0/0
-0016f5ab1256cd273	IPv4	Custom TCP	TCP	2000 - 11000	0.0.0.0/0

For Central server/VM

where we install terraform and install aws cli to authenticate and perform task through our aws account and to create eks cluster

Ec2 instance launched

Ubuntu 24.04 lts

T2 medium

Security group port to be opened

22 –ssh

443-https

80-http

465-smtps

25- smtp

1000-11000 custom tcp

6443 –custom tcp

Storage -20 gb

In server

aws cli install

curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip"

sudo apt install unzip

unzip awscliv2.zip

sudo ./aws/install

aws configure

install terraform

sudo snap install terraform --classic

go to github account

https://github.com/devops-methodology/Multi-Tier-With-Database.git

git clone https://github.com/devops-methodology/Multi-Tier-With-Database.git

cd EKS_terraform > terraform init

terraform plan

terraform apply --auto-approve

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

Terraform will perform the following actions:

  # aws_eks_cluster.devopsshack will be created
  + resource "aws_eks_cluster" "devopsshack" {
      + arn                       = (known after apply)
      + bootstrap_self_managed_addons = true
      + certificate_authority     = (known after apply)
      + cluster_id                = (known after apply)
      + created_at                = (known after apply)
      + endpoint                  = (known after apply)
      + id                        = (known after apply)
      + identity                  = (known after apply)
      + name                      = "devopsshack-cluster"
      + platform_version          = (known after apply)
      + role_arn                  = (known after apply)
      + status                    = (known after apply)
      + tags_all                  = (known after apply)
      + version                   = (known after apply)

      + access_config (known after apply)

      + kubernetes_network_config (known after apply)
```

Then create sonarqube and nexus

T2 medium storage -20 gb(2 instance)

After that

In Nexus server

sudo apt install docker.io –y

docker usermod –aG docker ubuntu

newgrp docker

docker run –d --name nexus3 –p 8081:8081 sonatype/nexus3

after that access instance ip:8081

user id: admin

password: go to inside the docker container

before going to container we have to write <container id> for that check for docker ps

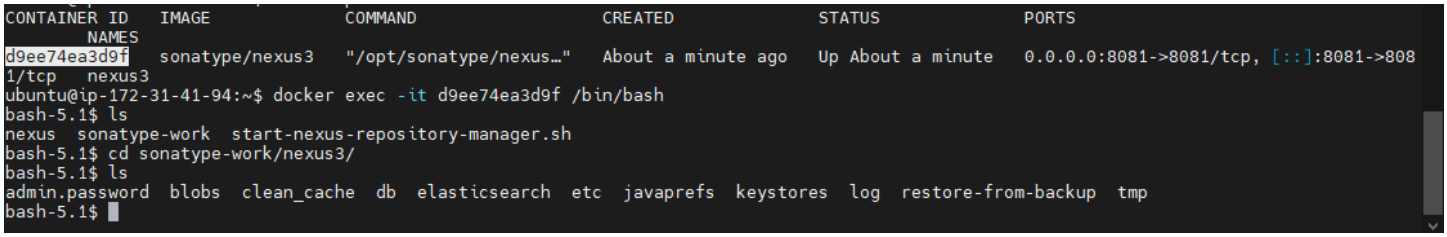
copy <container id>

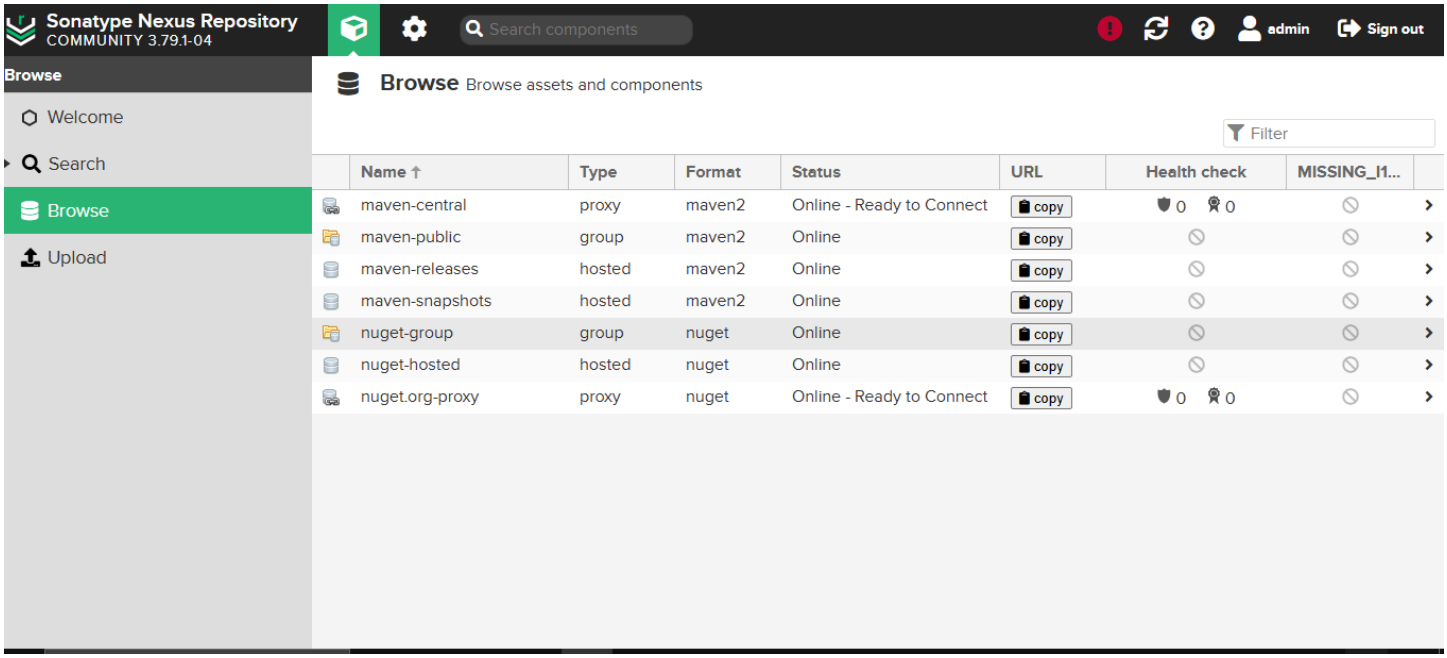
then docker exec –it <containerid> /bin/bash

ls

cd sonatype-work/nexus3/

cat admin.password… ▶





after that go for SonarQube server

sudo apt update

sudo apt install docker.io –y

sudo usermod –aG docker ubuntu

newgrp docker

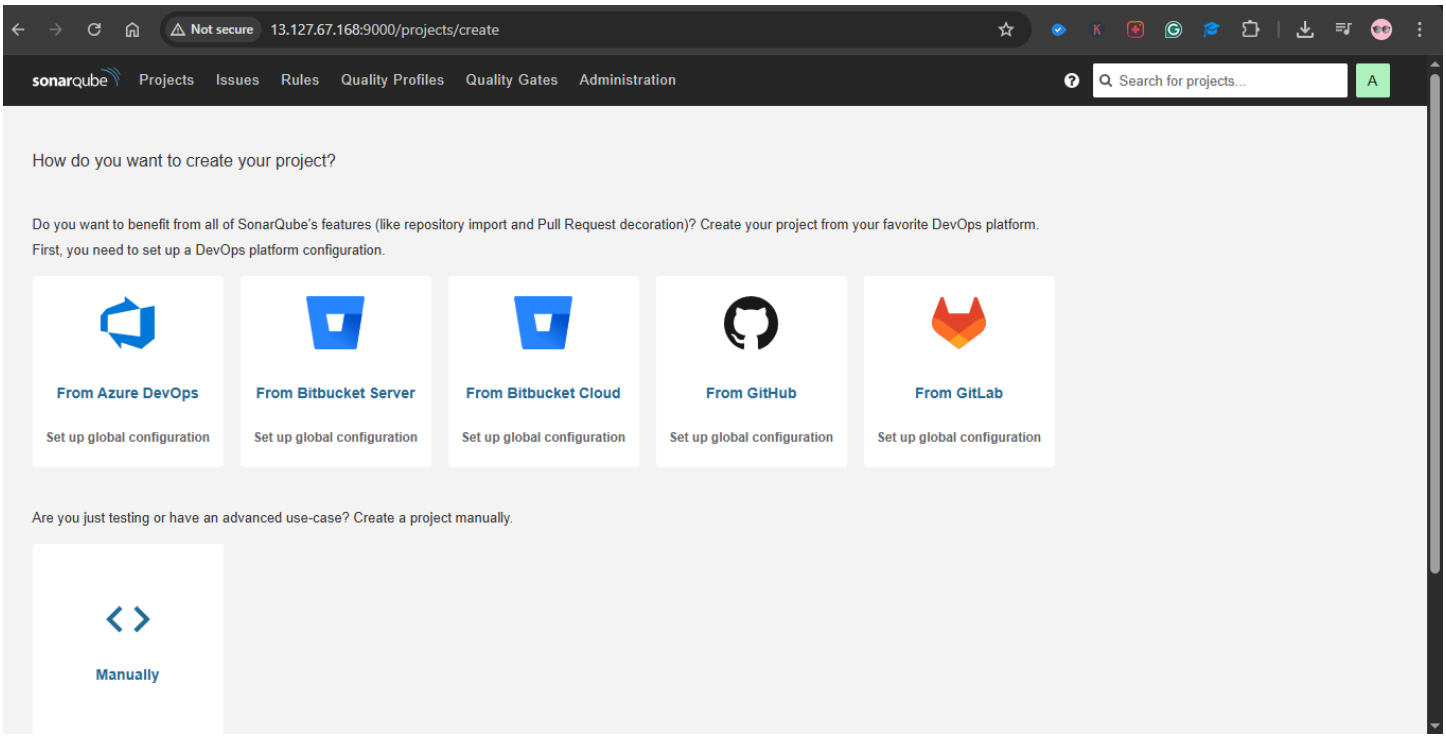
docker run –d --name sonar –p 9000:9000 sonarqube:lts-community

then access sonarqube server

instanceip:9000

user id:admin

password: admin



Then to setup Jenkins server as a master node so bigger machine

Create an instance

T2 large,storage -25 gb

After accessing terminal

sudo apt update

sudo apt install openjdk-17-jre-headless –y

sudo wget -O /etc/apt/keyrings/jenkins-keyring.asc \

https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key

echo "deb [signed-by=/etc/apt/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

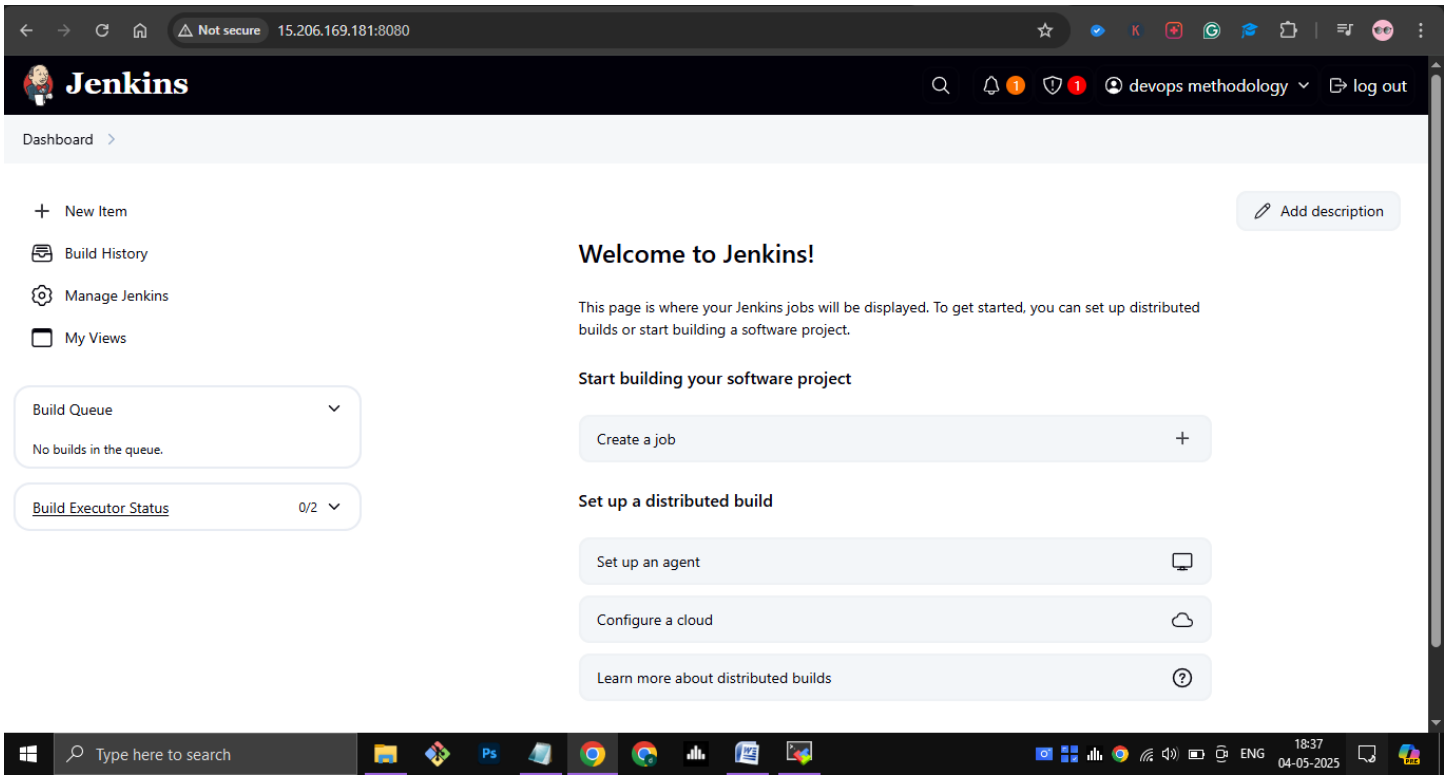
sudo apt-get install Jenkins –y

after that

sudo systemctl enable Jenkins

sudo systemctl start Jenkins

then access Jenkins instance ip:8080



for password-: service Jenkins status

there will be the password and copy and paste it

then go to central server

to interact with eks cluster we have to install kubectl

sudo snap install kubectl --classic

we have created the cluster but to interact with we have to update kubernetes configuration file

aws eks --region ap-south-1 update-kubeconfig --name devopsshack-cluster

you can check kubectl get nodes

```
ubuntu@ip-172-31-39-177:~/Multi-Tier-With-Database/EKS_Terraform$ aws eks --region ap-south-1 update-kubeconfig --name devopsshack-cluste
r
Added new context arn:aws:eks:ap-south-1:608729706295:cluster/devopsshack-cluster to /home/ubuntu/.kube/config
ubuntu@ip-172-31-39-177:~/Multi-Tier-With-Database/EKS_Terraform$ kubectl get nodes
NAME                                          STATUS   ROLES    AGE   VERSION
ip-10-0-0-14.ap-south-1.compute.internal      Ready    <none>   24m   v1.32.3-eks-473151a
ip-10-0-0-225.ap-south-1.compute.internal     Ready    <none>   24m   v1.32.3-eks-473151a
ip-10-0-1-231.ap-south-1.compute.internal     Ready    <none>   24m   v1.32.3-eks-473151a
ubuntu@ip-172-31-39-177:~/Multi-Tier-With-Database/EKS_Terraform$
```

then go to Jenkins server

we have to install plugin

manage Jenkins

>plugin

1/Sonarqube-scanner

2/maven integration

3/pipeline maven integration

4/kubernetes

5/kubernetes cli

6/kubernetes credentials

7/kubernetes client api

8/pipeline stage view

9/docker pipeline

10/generic webhook trigger

11/config file provider

12/docker

13/kubectl (in Jenkins server)

14/trivy

(sudo apt-get install wget apt-transport-https gnupg lsb-release

wget -qO - https://aquasecurity.github.io/trivy-repo/deb/public.key | sudo apt-key add -

echo deb https://aquasecurity.github.io/trivy-repo/deb $(lsb_release -sc) main | sudo tee -a /etc/apt/sources.list.d/trivy.list

sudo apt-get update

sudo apt-get install trivy)

Install

In Jenkins server install docker from official website

Add Docker's official GPG key:

sudo apt-get update

sudo apt-get install ca-certificates curl

sudo install -m 0755 -d /etc/apt/keyrings

sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg -o /etc/apt/keyrings/docker.asc

sudo chmod a+r /etc/apt/keyrings/docker.asc

Add the repository to Apt sources:

echo \

"deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.asc] https://download.docker.com/linux/ubuntu \

$(. /etc/os-release && echo "${UBUNTU_CODENAME:-$VERSION_CODENAME}") stable" | \

sudo tee /etc/apt/sources.list.d/docker.list > /dev/null

sudo apt-get update

then install the latest version of the docker

sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin

after that

sudo docker usermod –aG docker Jenkins

then instance ip:8080/restart

then go to manage Jenkins >tool>

maven>maven3 (with latest version)

sonarqube scanner

sonar-scanner(with latest version)

to configure sonarqube server we need credential

so go to sonarqube server

Administration>security>users>A(token name)>generate token

Then copy it

then go to manage Jenkins>system> we have to configure sonarqube server

under sonarqube

name –sonar

sonarqube url:9000

paste the token and save it

Lets start for pipeline

```
Pipeline {
    agent any
      tools {
          maven 'maven3'
      environment{
           SCANNER_HOME= tool 'sonar-scanner'
}
}
}
```

1/git checkout

2/sh "mvn compile" (to know any syntax based error is there or not"

3/sh "mvn test –DskipTests=true"

We have to scan for that we will use trivy and we have to install as a third party tool as there is no plugin injenkins

sudo apt-get install wget apt-transport-https gnupg lsb-release

wget -qO - https://aquasecurity.github.io/trivy-repo/deb/public.key | sudo apt-key add -

echo deb https://aquasecurity.github.io/trivy-repo/deb $(lsb_release -sc) main | sudo tee -a /etc/apt/sources.list.d/trivy.list

sudo apt-get update

sudo apt-get install trivy

(after installation it will be available directly in the pipeline)

4/sh "trivy fs --format table –o fs-report.html ."

```
stage('Trivy Fs Scan') {
    steps {
        sh "trivy fs --format table -o fs-report.html ."
    }
}
```

5/sonarqube analysis

```
stage('SonarQube Analysis') {
    steps {
        withSonarQubeEnv('sonar') {
        sh "$SCANNER_HOME/bin/sonar-scanner -Dsonar.projectName=Multitier -Dsonar.projectKey=Multitier -Dsc

        }
    }
}
```

sh “$SCANNER_HOME/bin/sonar-scanner -Dsonar.projectName=Multitier –Dsonar.projectKey=Multitier –Dsonar.java.binaries=target”

6/buld my project artifact in my local folder

sh “mvn package –DskipTests=true”

7/publish to nexus

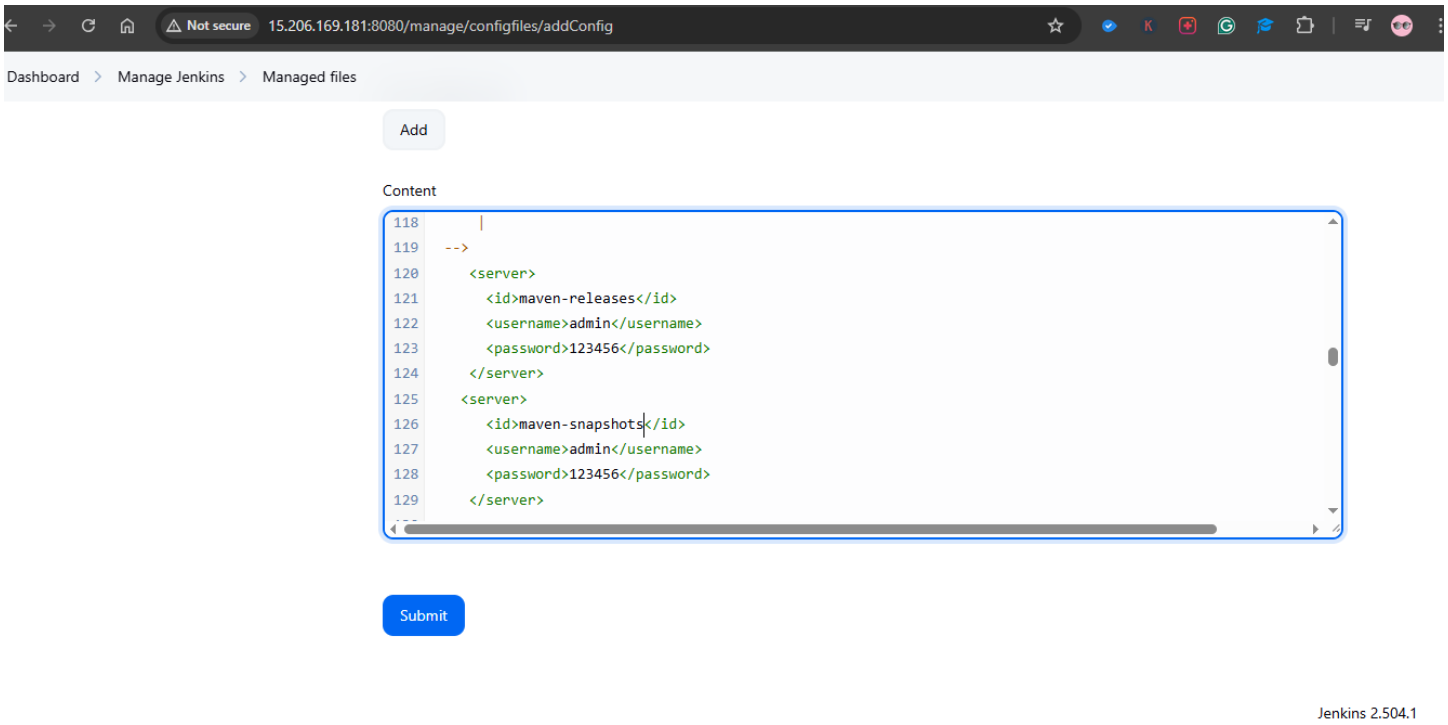
For that we have to go to nexus server copy the url of maven-releases and maven-snapshots

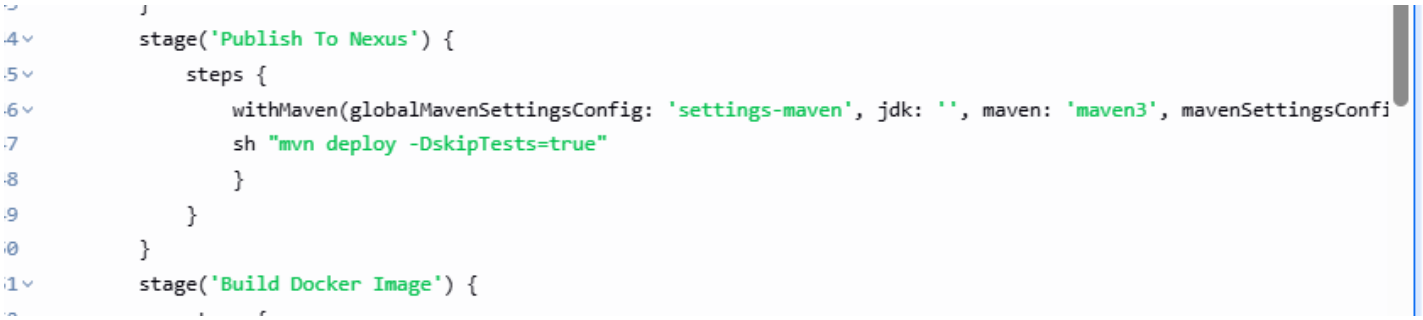
And then go to pom.xml in github in <distribution-management>

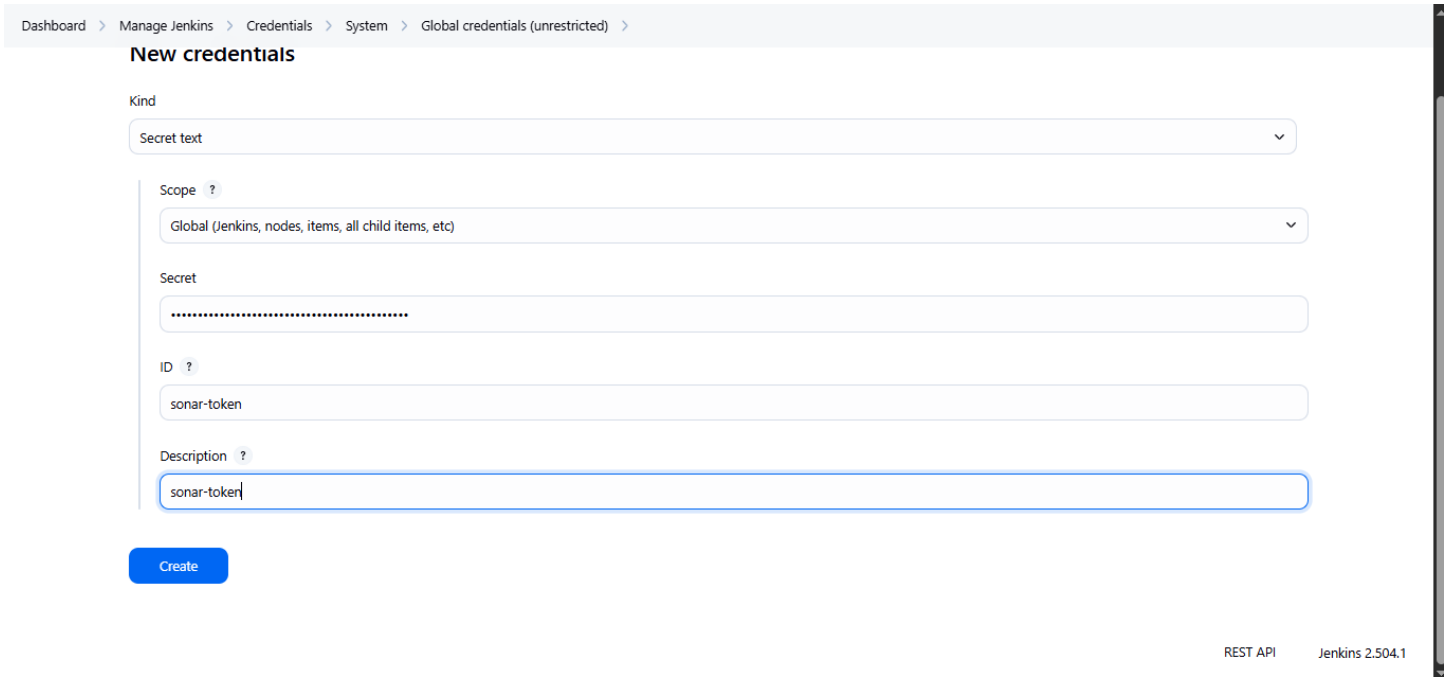
Edit

In maven release-paste

Maven-snapshot-paste and then go to Jenkins>config-file-provider>global-maven-settings>settings-maven>(for credential)







Sh “mvn deploy –DskipTests=true”

8/docker build image

```
stage('Build Docker Image') {
    steps {
        script{
            withDockerRegistry(credentialsId: 'docker-cred') {
                sh "docker build -t premd91/bankapp:latest ."
            }
        }
    }
}
```

Add docker-cred

Withdockerregistry(pipeline syntax)

Script{

“docker build –t premd91/bankapp:latest .”

}

9/trivy image scan

```
stage('Trivy Image Scan') {
    steps {
        sh "trivy image --format table -o fs-report.html premd91/bankapp:latest"
    }
}
```

sh “trivy image –format table –o fs-report.html premd91/bankapp”

10/docker push image

```
stage('Docker Push Image') {
    steps {
        script{
            withDockerRegistry(credentialsId: 'docker-cred') {
                sh "docker push premd91/bankapp:latest"
            }
        }
    }
}
```

sh “docker push premd91/bankapp:latest”

lets say when we go for deployment in kubernetes we have the security for that we go for rbac where everybody will not perform deployment,only the specific user account is accessible for that

so go to server

mkdir RBAC/

inside we will create a service account in namespace webapps

so we have to 1st create webapps namespace

kubectl create namespace webapps

a/create service account named as Jenkins

```
ubuntu@ip-172-31-39-177:~/Multi-Tier-With-Database/EKS_Terraform$ mkdir RBAC
ubuntu@ip-172-31-39-177:~/Multi-Tier-With-Database/EKS_Terraform$ ls
RBAC  main.tf  output.tf  terraform.tfstate  variables.tf
ubuntu@ip-172-31-39-177:~/Multi-Tier-With-Database/EKS_Terraform$ cd RBAC/
ubuntu@ip-172-31-39-177:~/Multi-Tier-With-Database/EKS_Terraform/RBAC$ ls
ubuntu@ip-172-31-39-177:~/Multi-Tier-With-Database/EKS_Terraform/RBAC$ kubetcl create ns webapps
Command 'kubetcl' not found, did you mean:
  command 'kubectl' from snap kubectl (1.32.4)
See 'snap info <snapname>' for additional versions.
ubuntu@ip-172-31-39-177:~/Multi-Tier-With-Database/EKS_Terraform/RBAC$ kubectl create ns webapps
namespace/webapps created
ubuntu@ip-172-31-39-177:~/Multi-Tier-With-Database/EKS_Terraform/RBAC$ vi svc.yml
ubuntu@ip-172-31-39-177:~/Multi-Tier-With-Database/EKS_Terraform/RBAC$ kuebctl apply -f svc.yml
Command 'kuebctl' not found, did you mean:
  command 'kubectl' from snap kubectl (1.32.4)
See 'snap info <snapname>' for additional versions.
ubuntu@ip-172-31-39-177:~/Multi-Tier-With-Database/EKS_Terraform/RBAC$ kubectl apply -f svc.yml
serviceaccount/jenkins created
ubuntu@ip-172-31-39-177:~/Multi-Tier-With-Database/EKS_Terraform/RBAC$
```

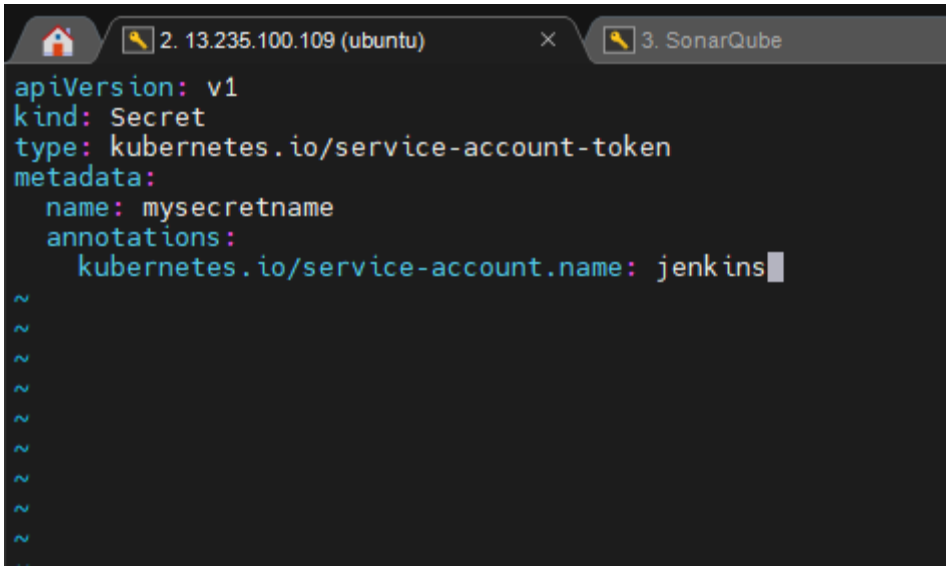
b/role-what action to be performed on the resources

c/rolebinding

now Jenkins have the permission to perform these actions

create a token

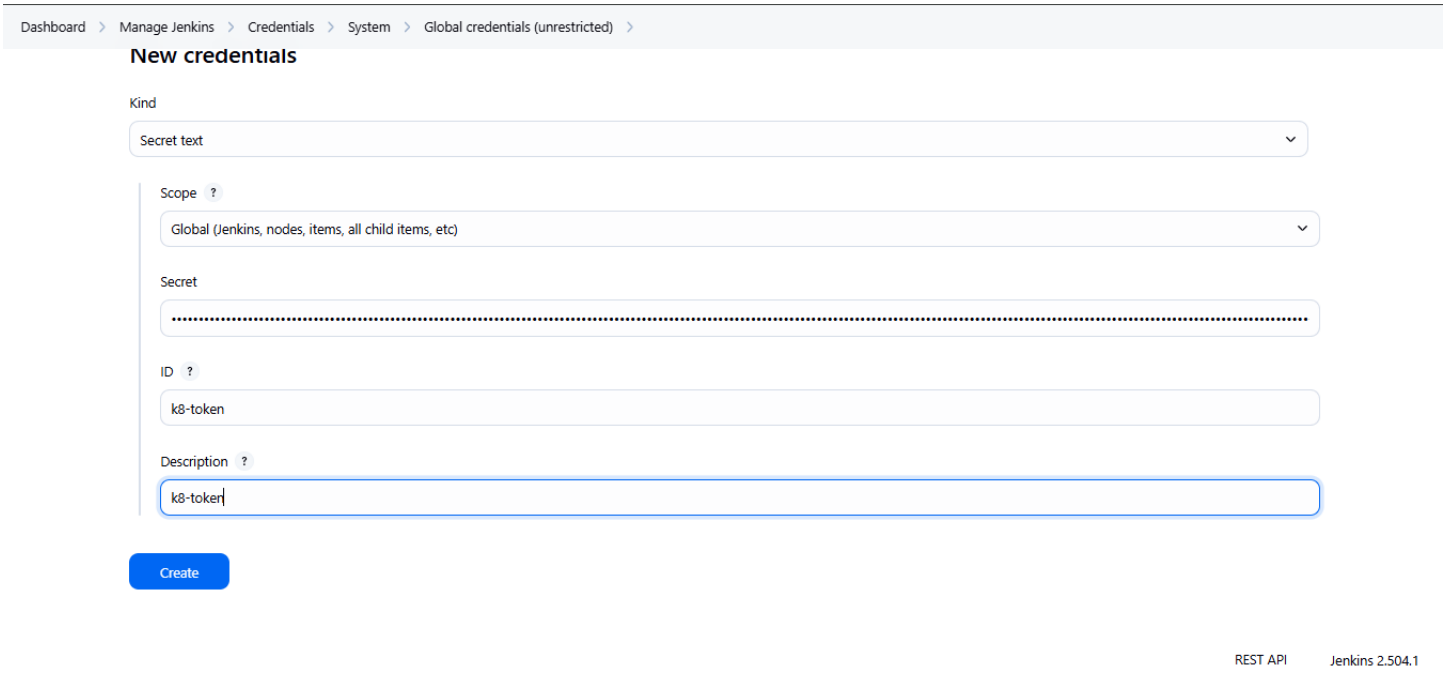
kubectl apply –f sec.yml –n webapps



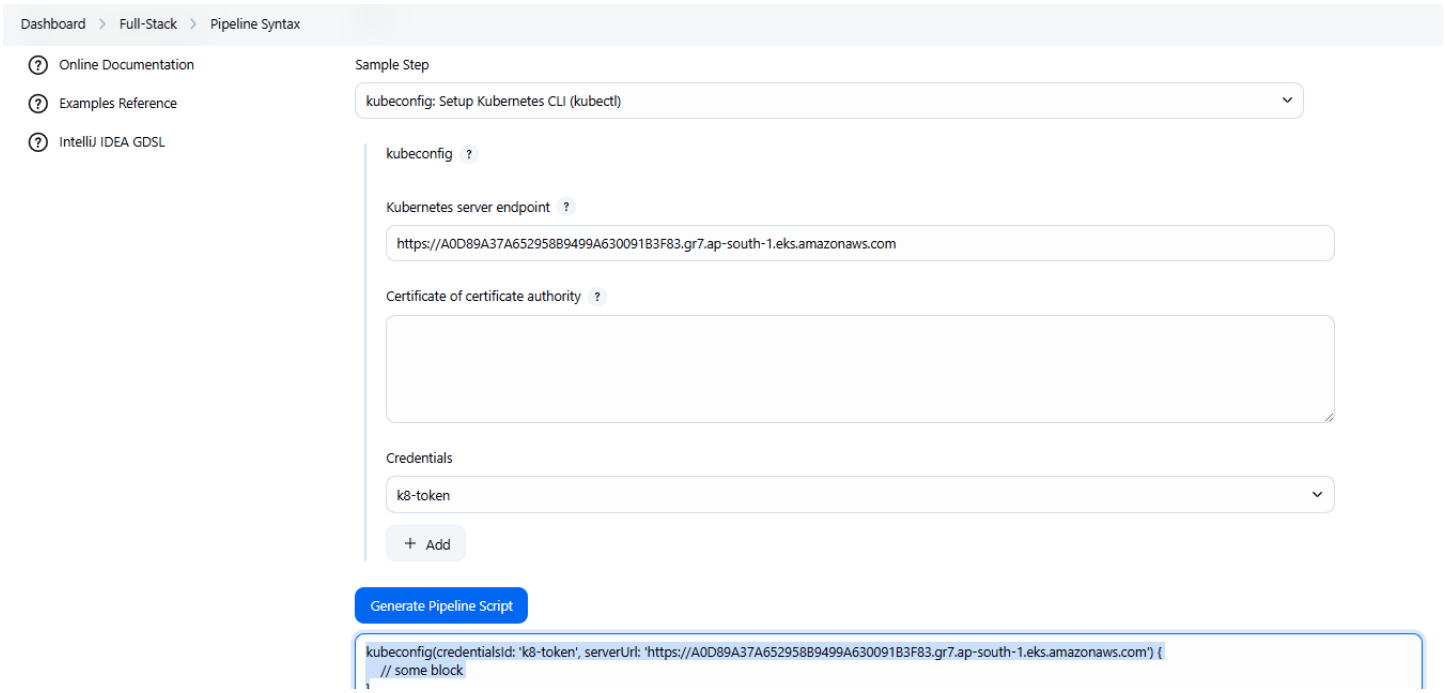
kubectl describe secret mysecretname –n webapps

copy the token

go to Jenkins>manage Jenkins>credentials>global>add credentials>secret txt>paste token>k8-token



then go to pipeline syntax withkubeconfigenv>add k8-token>eks api server end point paste>devopsshack-cluster>webapps.



10/deploy to kubernetes

```
66          steps {
67              script{
68                  withDockerRegistry(credentialsId: 'docker-cred') {
69                      sh "docker push premd91/bankapp:latest"
70                  }
71              }
72          }
73      }
74      stage('Deploy To Kubernetes') {
75          steps {
76              kubeconfig(credentialsId: 'k8-token', serverUrl: 'https://A0D89A37A652958B9499A630091B3F83.gr7.ap-s
77                  sh "kubectl apply -f ds.yml -n webapps"
78                  sleep 30
79              }
80          }
81      }
```

sh “kubectl apply –f ds.yml –n webapps”

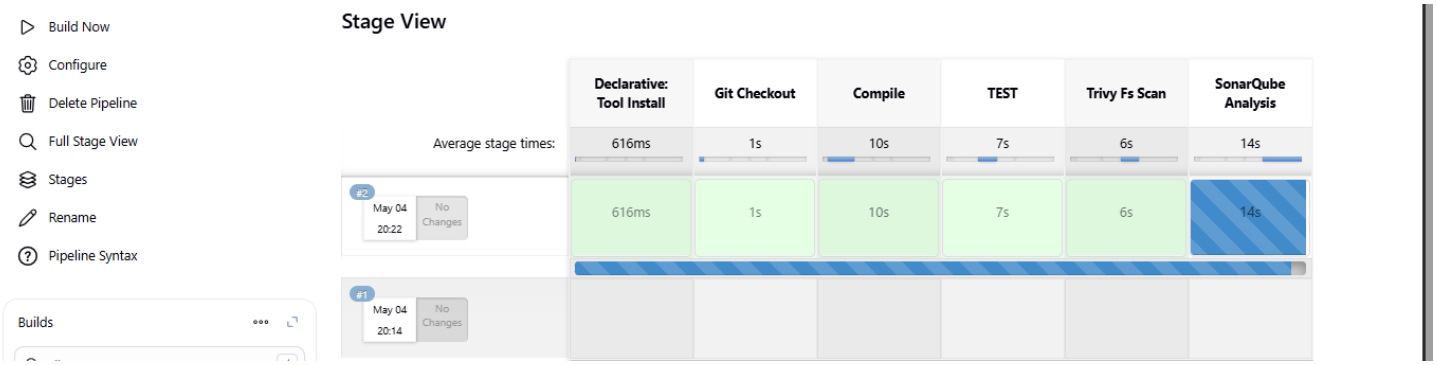
11/verify deployment

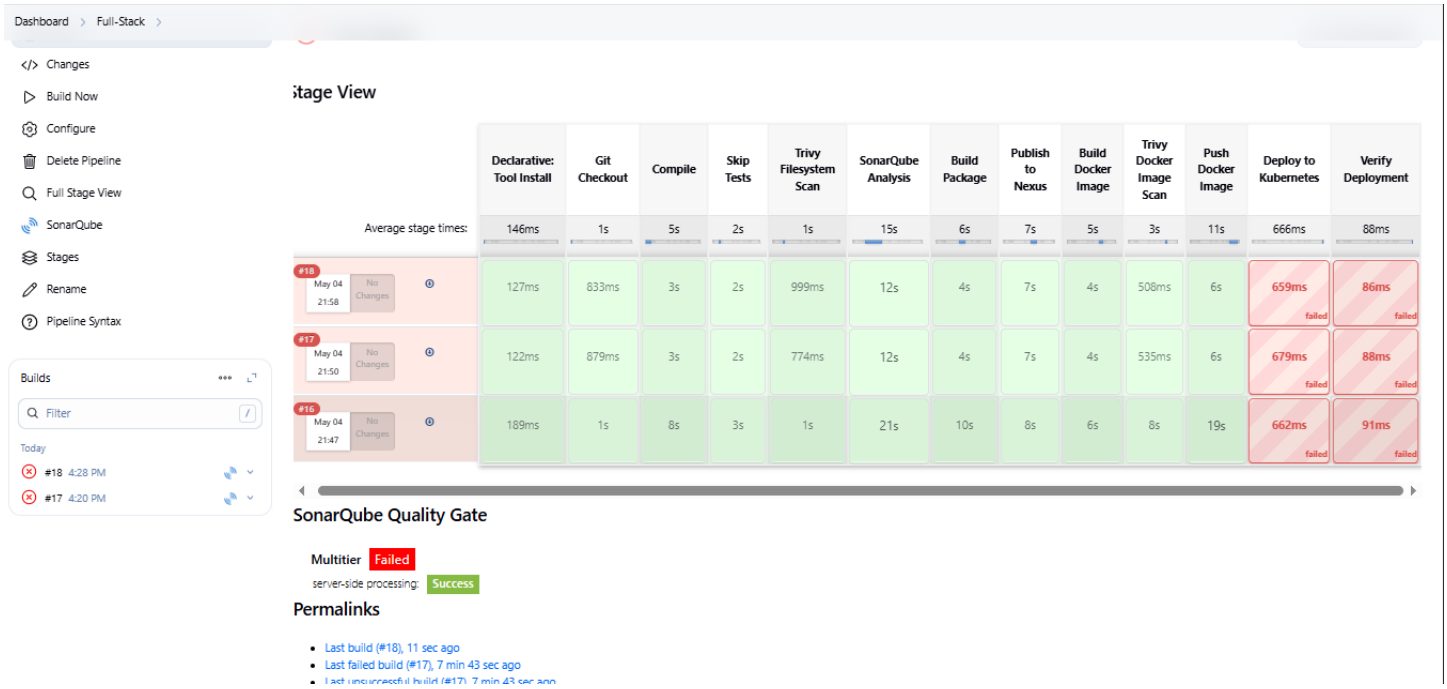
```
81      }
82      stage('Verify Deployment') {
83          steps {
84              kubeconfig(credentialsId: 'k8-token', serverUrl: 'https://A0D89A37A652958B9499A630091B3F83.gr7.ap-s
85                  sh "kubectl get pods -n webapps"
86                  sh "kubectl get svc -n webapps"
87              }
88          }
89      }
90  }
91  }
92
```

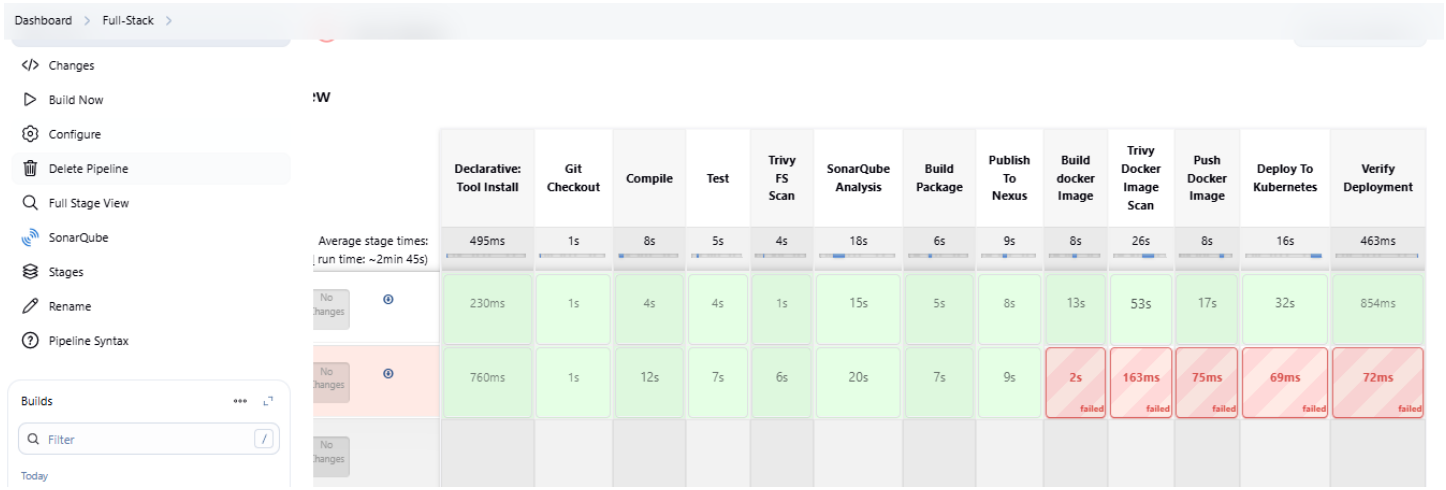
Use Groovy Sandbox ?

sh “ kubectl get pods –n webapps”

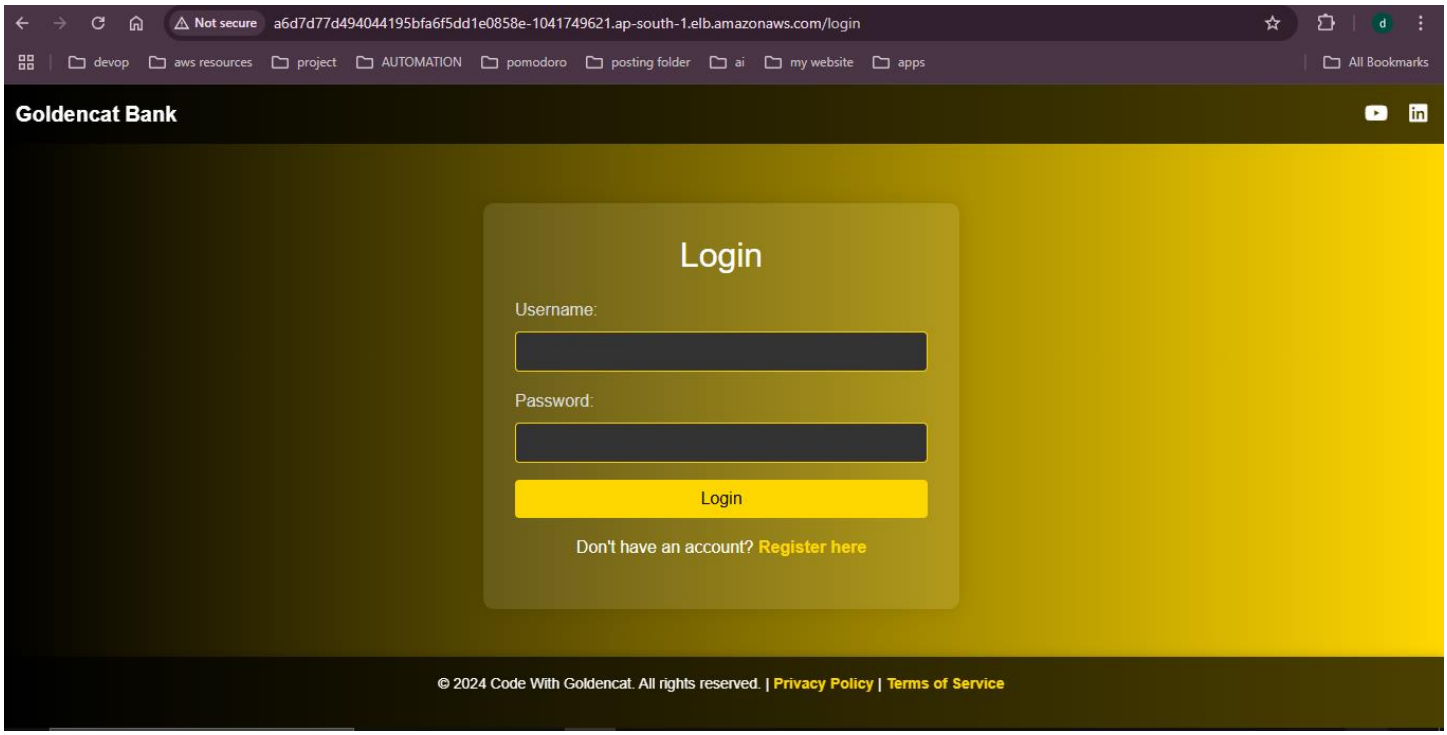
“kubectl get svc –n webapps”

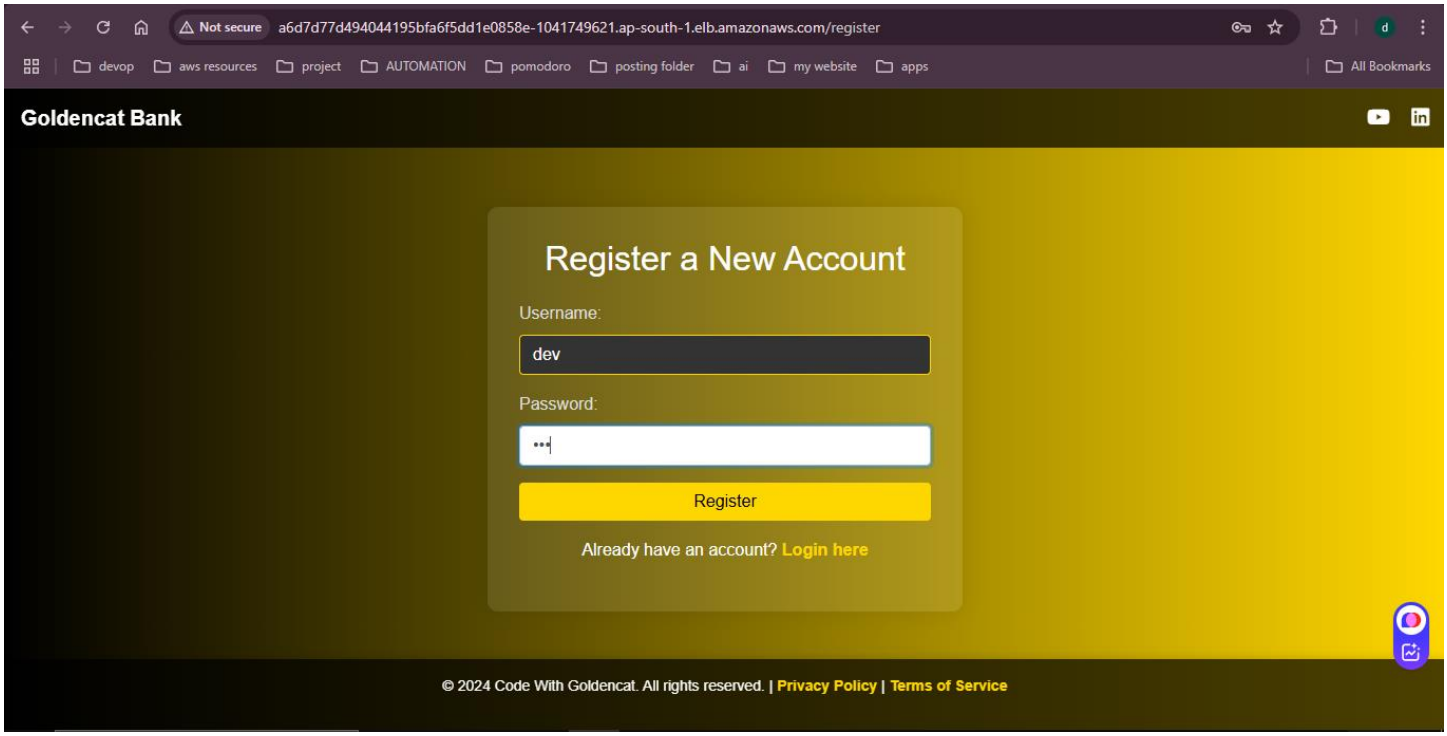


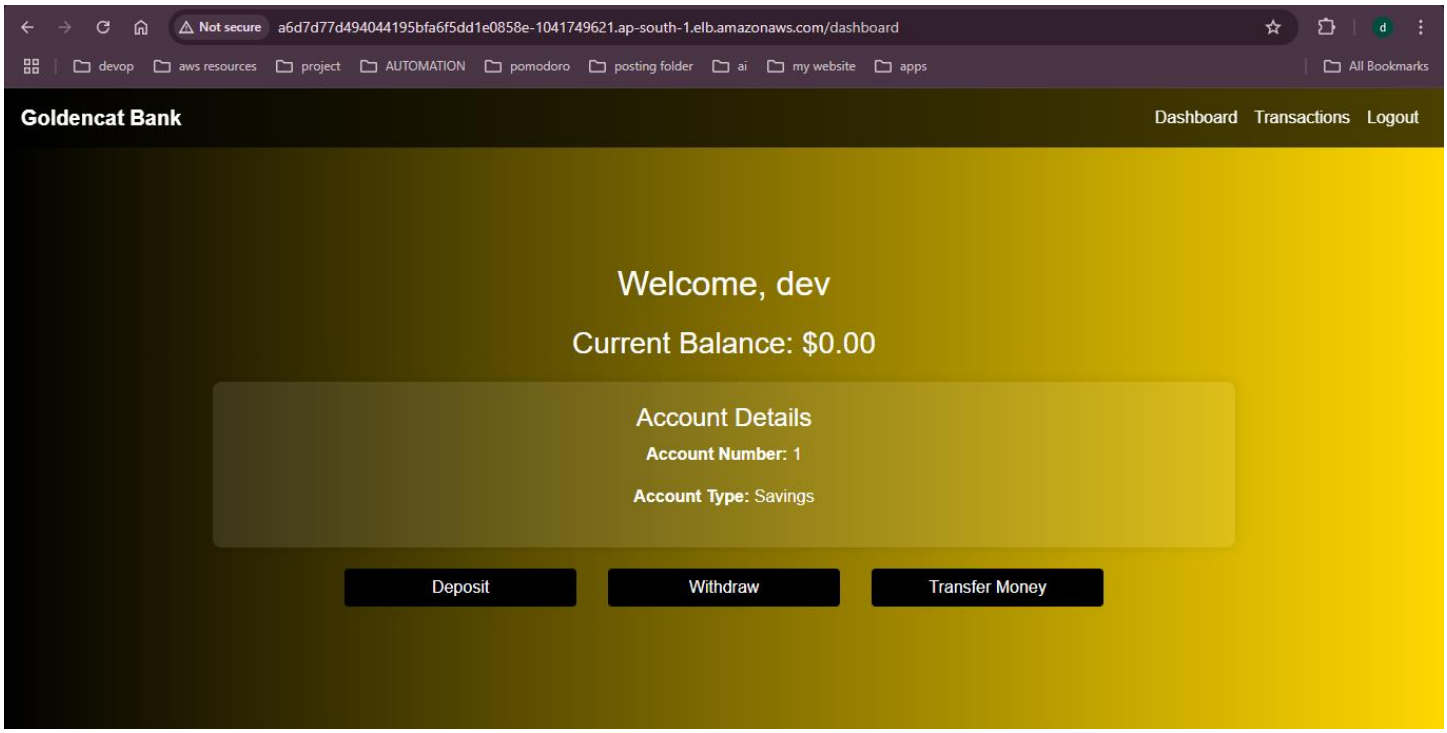


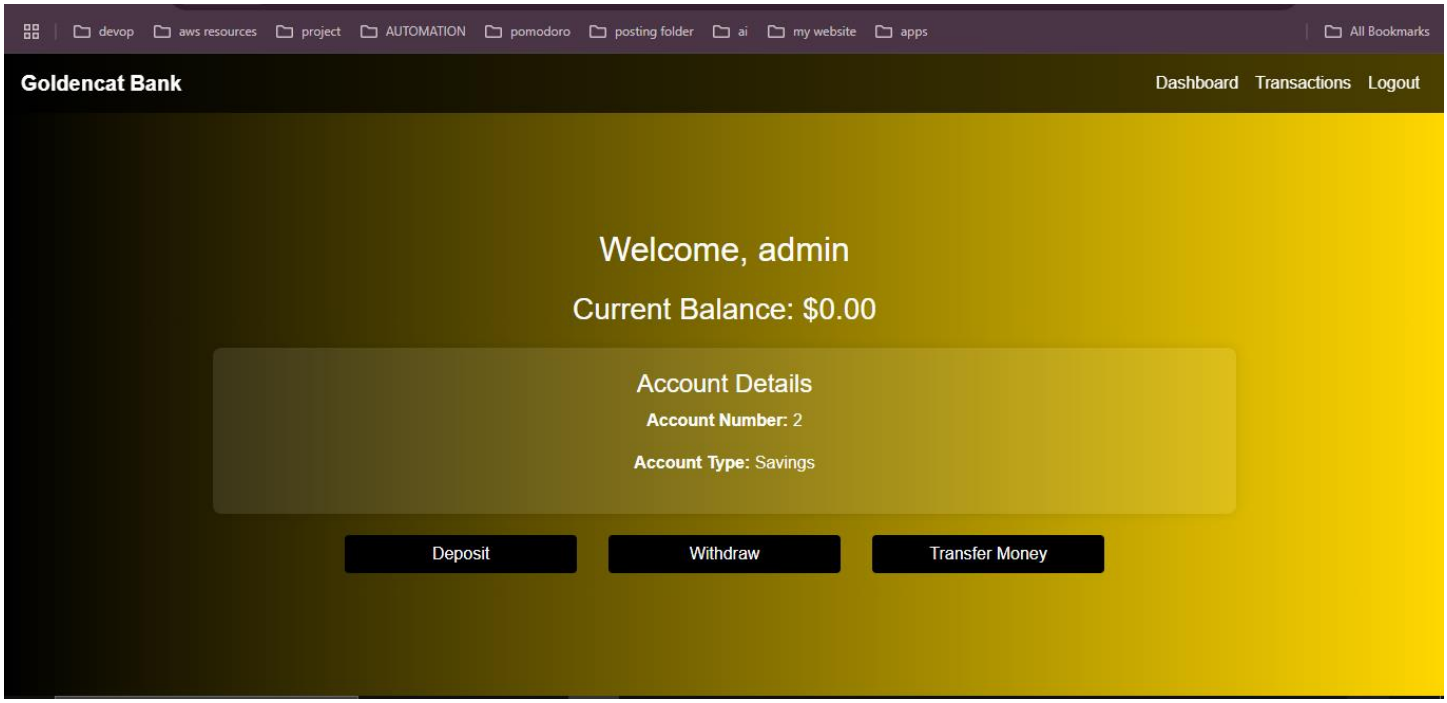


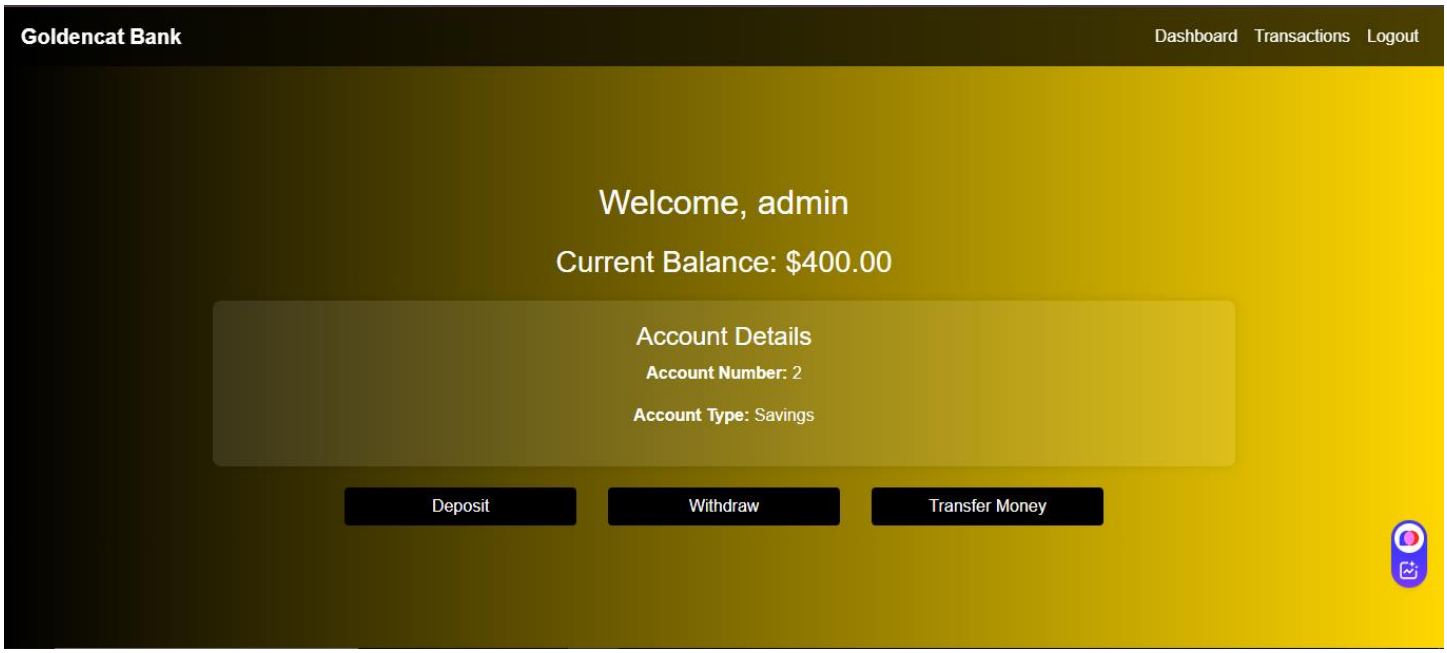

```
ubuntu@ip-172-31-34-147:~/Multi-Tier-With-Database/EKS_Terraform/RBAC$ kubectl get all -n webapps
NAME                          READY   STATUS    RESTARTS      AGE
pod/bankapp-877f8c49c-5z82d   1/1     Running   2 (71s ago)   104s
pod/mysql-7cbcb976bf-7dxkx    1/1     Running   0             104s

NAME                       TYPE           CLUSTER-IP      EXTERNAL-IP                                                              PORT(
S)        AGE
service/bankapp-service    LoadBalancer   172.20.203.15   a6d7d77d494044195bfa6f5dd1e0858e-1041749621.ap-south-1.elb.amazonaws.com   80:30
607/TCP   104s
service/mysql-service      ClusterIP      172.20.98.54    <none>                                                                   3306/
TCP       104s

NAME                     READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/bankapp  1/1     1            1           104s
deployment.apps/mysql    1/1     1            1           104s

NAME                                DESIRED   CURRENT   READY   AGE
replicaset.apps/bankapp-877f8c49c   1         1         1       104s
replicaset.apps/mysql-7cbcb976bf    1         1         1       104s
ubuntu@ip-172-31-34-147:~/Multi-Tier-With-Database/EKS_Terraform/RBAC$
```

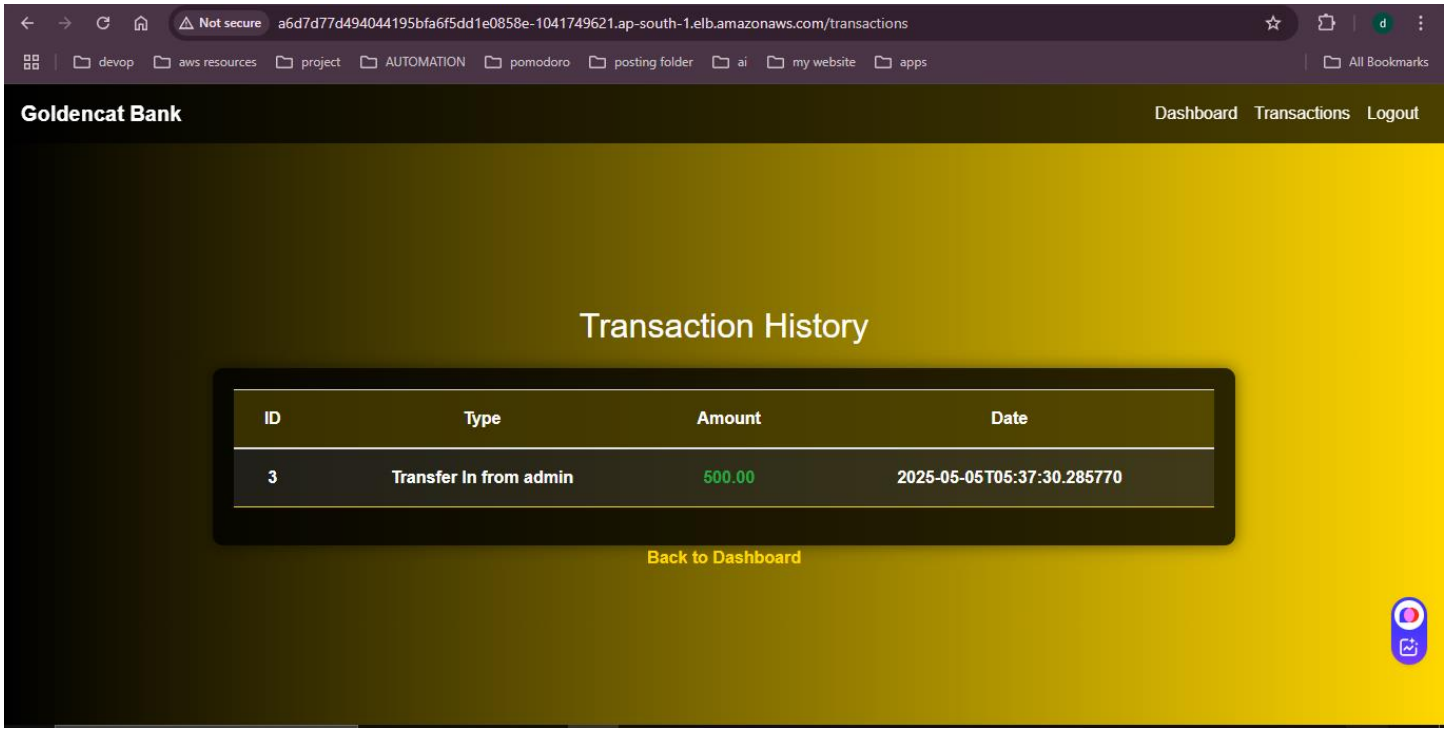

Not secure
a6d7d77d494044195bfa6f5dd1e0858e-1041749621.ap-south-1.elb.amazonaws.com/login
devop
aws resources
project
AUTOMATION
pomodoro
posting folder
ai
my website
apps
All Bookmarks
Goldencat Bank
Login
Username:
Password:
Login
Don't have an account? Register here
© 2024 Code With Goldencat. All rights reserved. | Privacy Policy | Terms of Service


Not secure
a6d7d77d494044195bfa6f5dd1e0858e-1041749621.ap-south-1.elb.amazonaws.com/register
devop
aws resources
project
AUTOMATION
pomodoro
posting folder
ai
my website
apps
All Bookmarks
Goldencat Bank
Register a New Account
Username:
dev
Password:
Register
Already have an account? Login here
© 2024 Code With Goldencat. All rights reserved. | Privacy Policy | Terms of Service

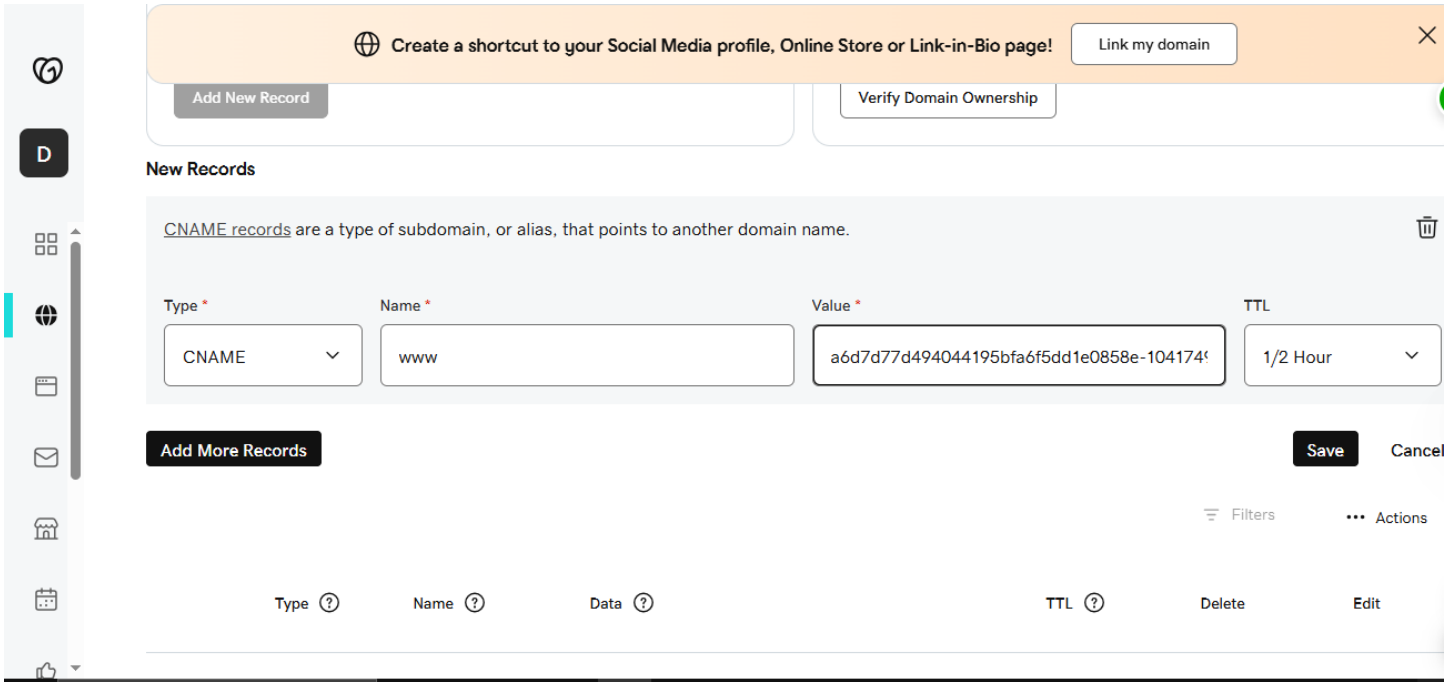

Not secure
a6d7d77d494044195bfa6f5dd1e0858e-1041749621.ap-south-1.elb.amazonaws.com/dashboard
devop
aws resources
project
AUTOMATION
pomodoro
posting folder
ai
my website
apps
All Bookmarks
Goldencat Bank
Dashboard
Transactions
Logout
Welcome, dev
Current Balance: $0.00
Account Details
Account Number: 1
Account Type: Savings
Deposit
Withdraw
Transfer Money


devop
aws resources
project
AUTOMATION
pomodoro
posting folder
ai
my website
apps
All Bookmarks
Goldencat Bank
Dashboard
Transactions
Logout
Welcome, admin
Current Balance: $0.00
Account Details
Account Number: 2
Account Type: Savings
Deposit
Withdraw
Transfer Money

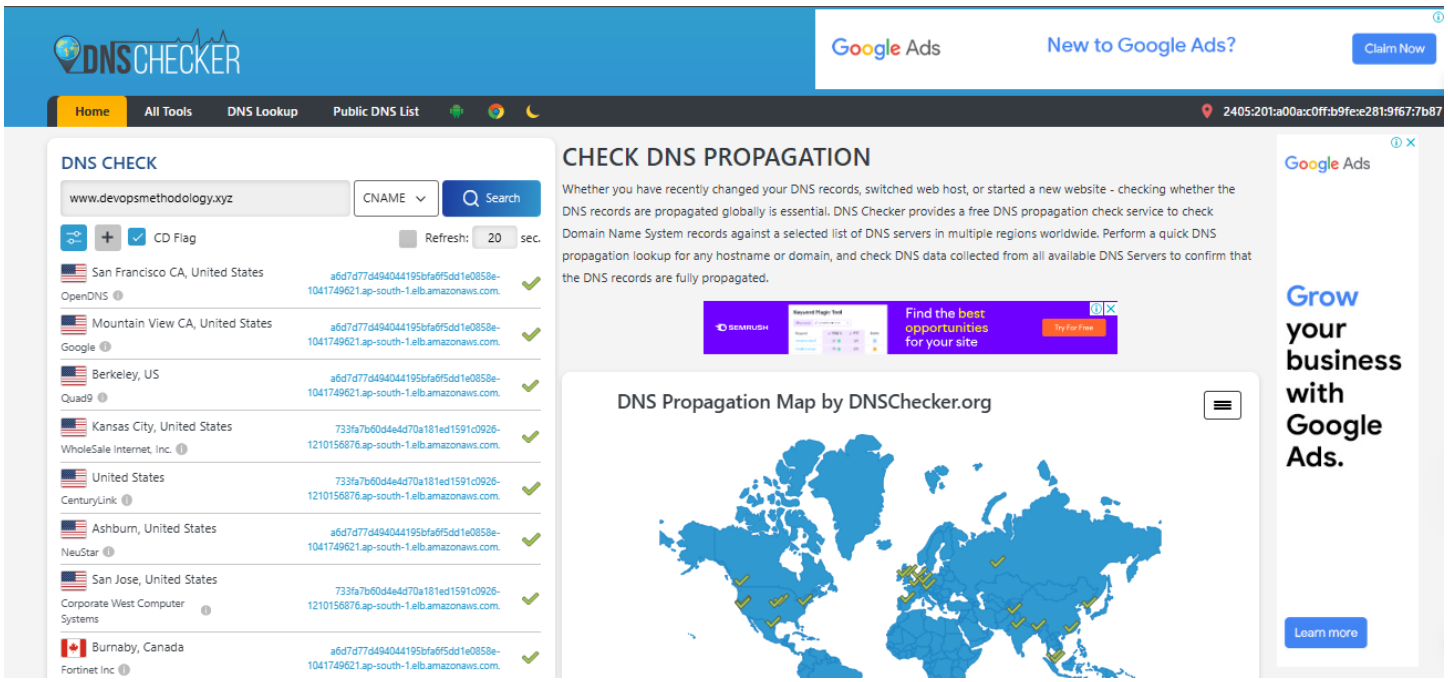

Goldencat Bank
Dashboard Transactions Logout
Welcome, admin
Current Balance: $400.00
Account Details
Account Number: 2
Account Type: Savings
Deposit
Withdraw
Transfer Money

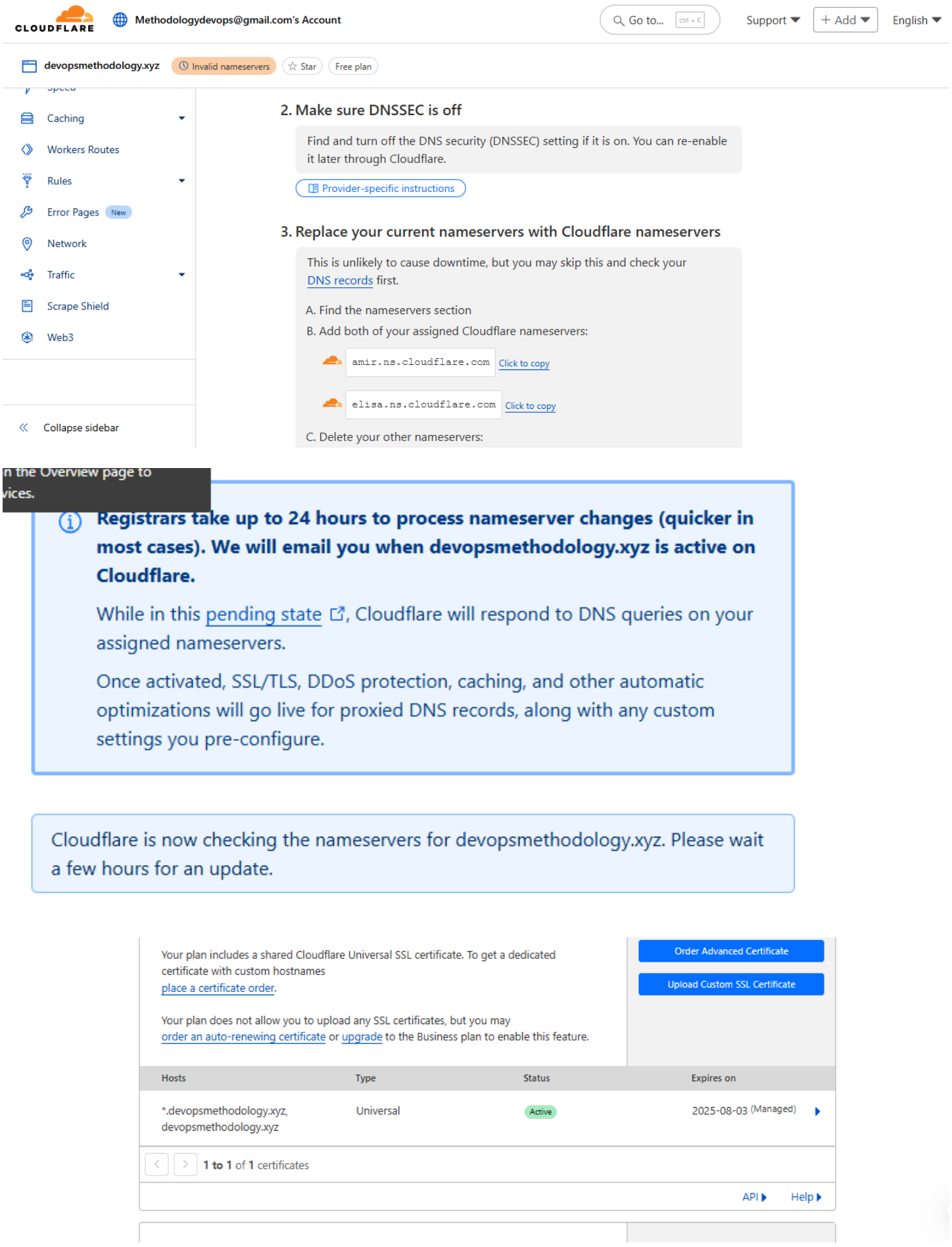

Not secure
a6d7d77d494044195bfa6f5dd1e0858e-1041749621.ap-south-1.elb.amazonaws.com/transactions
devop
aws resources
project
AUTOMATION
pomodoro
posting folder
ai
my website
apps
All Bookmarks
Goldencat Bank
Dashboard Transactions Logout
Transaction History
ID
Type
Amount
Date
3
Transfer In from admin
500.00
2025-05-05T05:37:30.285770
Back to Dashboard

For SECURE SSL CERTIFICATE



CHECK YOUR WEBSITE




CLOUDFLARE
Methodologydevops@gmail.com's Account
Go to... ctrl + K
Support
+ Add
English
devopsmethodology.xyz
Invalid nameservers
Star
Free plan
Caching
Workers Routes
Rules
Error Pages New
Network
Traffic
Scrape Shield
Web3
Collapse sidebar
2. Make sure DNSSEC is off
Find and turn off the DNS security (DNSSEC) setting if it is on. You can re-enable it later through Cloudflare.
Provider-specific instructions
3. Replace your current nameservers with Cloudflare nameservers
This is unlikely to cause downtime, but you may skip this and check your DNS records first.
A. Find the nameservers section
B. Add both of your assigned Cloudflare nameservers:
amir.ns.cloudflare.com Click to copy
elisa.ns.cloudflare.com Click to copy
C. Delete your other nameservers:
n the Overview page to
vices.
Registrars take up to 24 hours to process nameserver changes (quicker in most cases). We will email you when devopsmethodology.xyz is active on Cloudflare.
While in this pending state, Cloudflare will respond to DNS queries on your assigned nameservers.
Once activated, SSL/TLS, DDoS protection, caching, and other automatic optimizations will go live for proxied DNS records, along with any custom settings you pre-configure.
Cloudflare is now checking the nameservers for devopsmethodology.xyz. Please wait a few hours for an update.
Your plan includes a shared Cloudflare Universal SSL certificate. To get a dedicated certificate with custom hostnames place a certificate order.
Your plan does not allow you to upload any SSL certificates, but you may order an auto-renewing certificate or upgrade to the Business plan to enable this feature.
Order Advanced Certificate
Upload Custom SSL Certificate
Hosts
Type
Status
Expires on
*.devopsmethodology.xyz, devopsmethodology.xyz
Universal
Active
2025-08-03 (Managed)
1 to 1 of 1 certificates
API
Help

Overview

devopsmethodology.xyz

Monitor security and performance for devopsmethodology.xyz. Configure products and services from the menu.

Review Cloudflare fundamentals

Great news! Cloudflare is now protecting your site

Data about your site's usage will be here once available.

Recommendations

Hide

Speed up your website or application

Turn on recommended image, content, and protocol settings.

Go to speed optimization

DNS

DNS Setup: Full

DNS Records

Quick Actions

Under Attack Mode

Show visitors a JavaScript challenge when they visit your site.

About Under Attack Mode

Development Mode

Temporarily bypass our cache. See changes to your origin server in realtime.

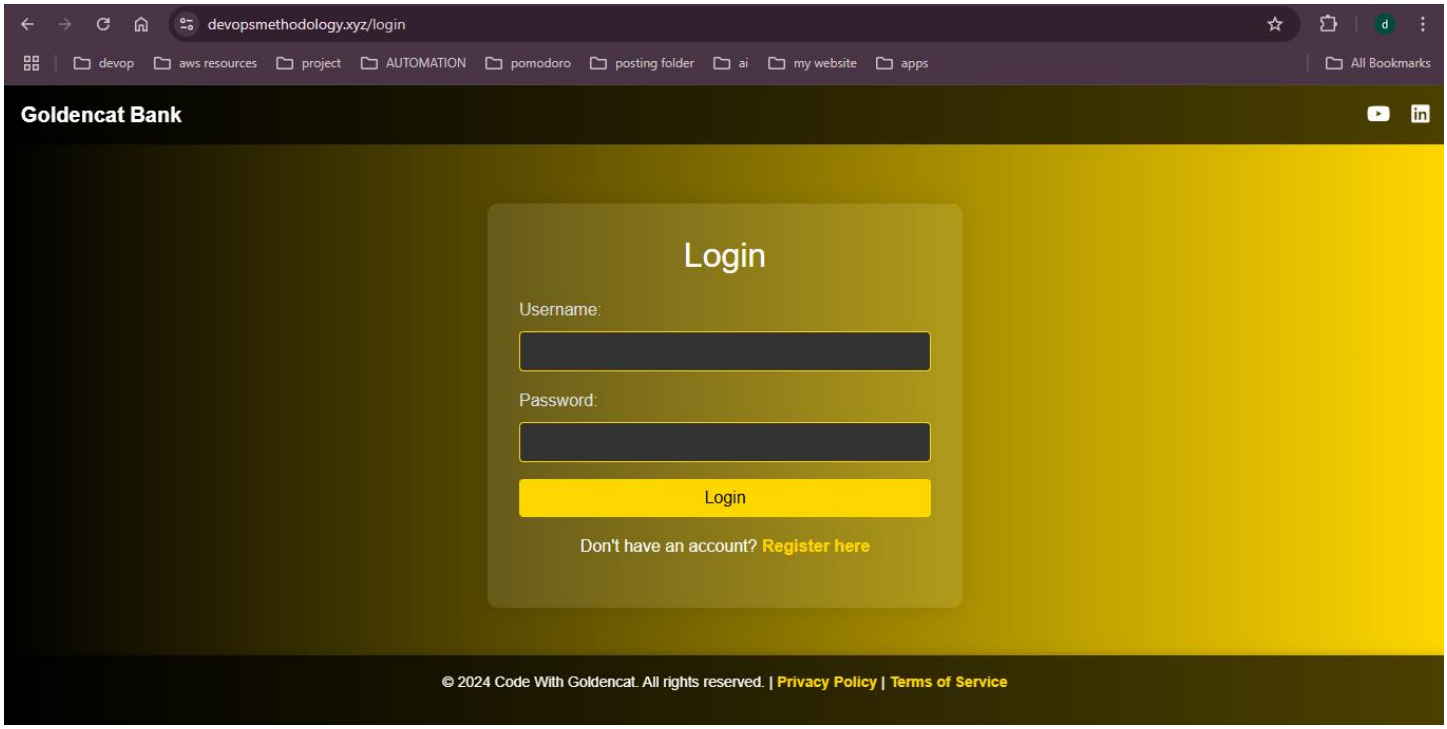
About Development Mode

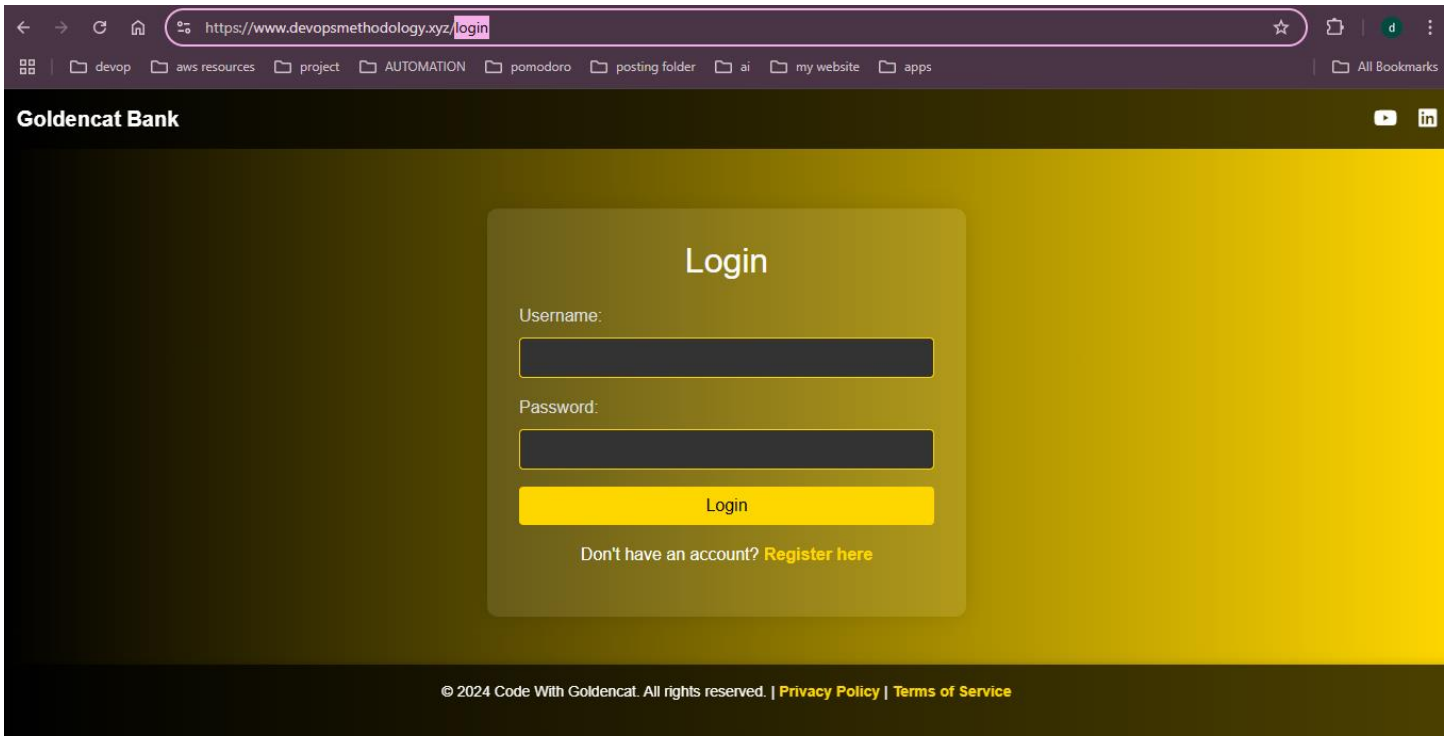
Run speed test

Configure caching

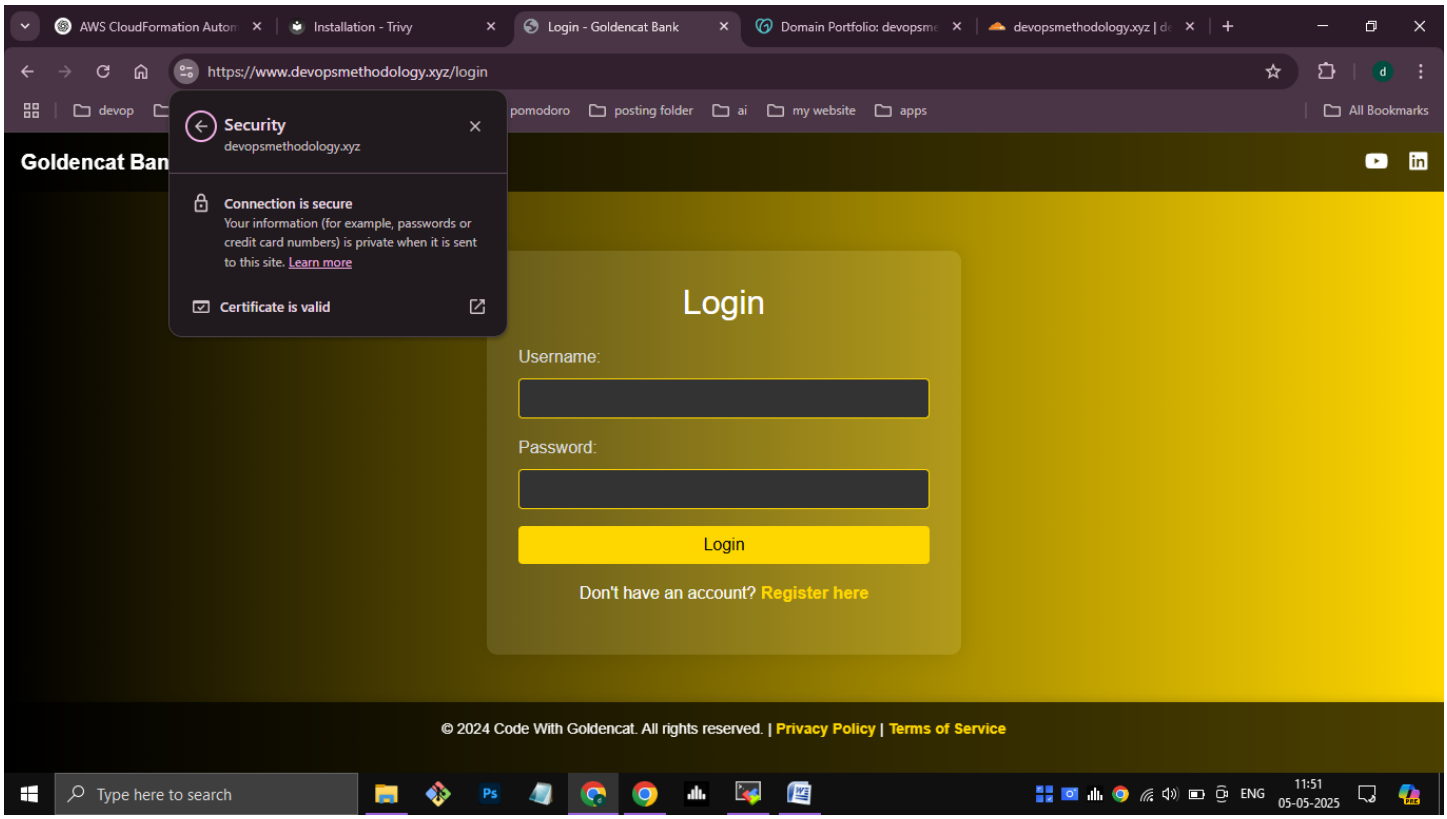
Domain Registration

Registrar: Unknown (Centralnic-H3105376)





CHECK IT IS SECURE



pipeline

```
pipeline {
    agent any
    tools {
        maven 'maven3'
    }
    environment {
        SCANNER_HOME= tool 'sonar-scanner'
    }

    stages {
        stage('Git Checkout') {
            steps {
                git branch: 'main', url: 'https://github.com/devops-methodology/Multi-Tier-With-Database.git'
            }
        }
        stage('Compile') {
            steps {
                sh "mvn compile"
            }
        }
        stage('Test') {
            steps {
                sh "mvn test -DskipTests=true"
            }
        }
        stage('Trivy FS Scan') {
            steps {
```

```
            sh "trivy fs --format table -o fs-report.html ."
        }
    }
    stage('SonarQube Analysis') {
        steps {
            withSonarQubeEnv('sonar') {
                sh '''$SCANNER_HOME/bin/sonar-scanner -Dsonar.projectName=Multitier -Dsonar.projectKey=Multitier -Dsonar.java.binaries=target'''
            }
        }
    }
    stage('Build Package') {
        steps {
            sh "mvn package -DskipTests=true"
        }
    }
    stage('Publish To Nexus') {
        steps {
            withMaven(globalMavenSettingsConfig: 'settings-maven', jdk: '', maven: 'maven3', mavenSettingsConfig: '', traceability: true) {
                sh "mvn deploy -DskipTests=true"
            }
        }
    }
    stage('Build docker Image') {
        steps {
            script {
                withDockerRegistry(credentialsId: 'docker-cred') {
                    sh "docker build -t premd91/bankapp:latest ."
```

```
            }
        }
    }
}
stage('Trivy Docker Image Scan') {
    steps {
        sh "trivy image --format table -o fs-report.html premd91/bankapp:latest"
    }
}
stage('Push Docker Image') {
    steps {
        script {
            withDockerRegistry(credentialsId: 'docker-cred') {
                sh "docker push premd91/bankapp:latest"
            }
        }
    }
}
stage('Deploy To Kubernetes') {
    steps {
        withKubeConfig(caCertificate: '', clusterName: 'devopsshack-cluster', contextName: '', credentialsId: 'k8-token',
namespace: 'webapps', restrictKubeConfigAccess: false, serverUrl:
'https://2DC2D51FC98F936FB04230C6AF678A3D.yl4.ap-south-1.eks.amazonaws.com') {
            sh "kubectl apply -f ds.yml -n webapps"
            sleep 30
        }
    }
}
stage('Verify Deployment') {
```

```
        steps {

            withKubeConfig(caCertificate: '', clusterName: 'devopsshack-cluster', contextName: '', credentialsId: 'k8-token', namespace: 'webapps', restrictKubeConfigAccess: false, serverUrl: 'https://2DC2D51FC98F936FB04230C6AF678A3D.yl4.ap-south-1.eks.amazonaws.com') {

                sh "kubectl get pods -n webapps"

                sh "kubectl get svc -n webapps"

            }

        }

    }

  }

}
```

To be deleted-(Very Important)

Eks cluster (1st node group from compute tab)

Roles

Check for elb and autoscaling group,

Terminate instance

Vpc

Volumes

Double check ec2 dashboard