ASecurityGuru

**DevSecOps & DevOps with Jenkins, Kubernetes, Terraform & AWS**

**Topics:** Implement SAST, SCA & DAST in Jenkins DevSecOps Pipeline from scratch and setup infra using Terraform, Kubernetes in AWS

**1. Install Terraform in centos 8**

**A. Install yum-utils package:**

**#sudo yum install -y yum-utils**

**B. Add the HashiCorp repository:**

**#sudo yum-config-manager --add-repo** <https://rpm.releases.hashicorp.com/RHEL/hashicorp.repo>

**C. Install terraform**

**#sudo yum -y install terraform**

**d. Verify Installation**

**#terraform -v**

**A computer screen with white text

Description automatically generated**

**============== Terraform Installation completed =================**

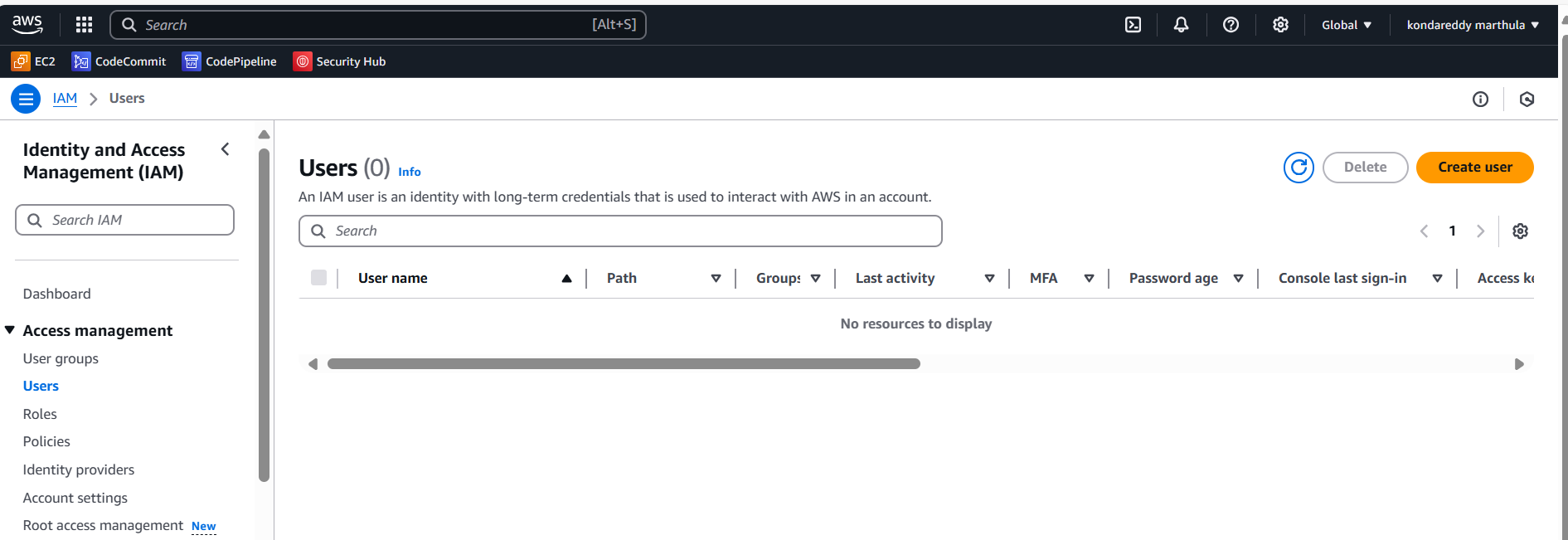
**2. Create AWS admin user**

**A. Sign in to the AWS Management Console:**

**Go to the IAM Management Console.**

**B. Create a new IAM user:**

**A. In the navigation pane, choose Users and then select Add user**

****

**B. Enter a username for the new user (e.g., AdminUser). Click Next**

**A screenshot of a computer

Description automatically generated**

**C. On the Permissions page, choose Attach existing policies directly.**

**Search for and select the appropriate policy(e.g., AdministratorAccess or a custom policy with the necessary permissions)**

**Click Next: Tags to add optional tags, then Next: Review**

**A screenshot of a computer

Description automatically generated**

**D. Click on create user**

**Review the user details and permissions.**

Click **Create user**.

**A screenshot of a computer

Description automatically generated**

**e. Click on the “akondareddyguru” user name**

**f. click on the “Security credential’ tab and Click on the “Create access key**

**A screenshot of a computer

Description automatically generated**

**g. Select the “Command line interface” in use case and mark on confirmation box and click on Next**

**H. Click on the create access key**

**A screenshot of a computer

Description automatically generated**

* **After creating the user, you will see a confirmation page with the user’s Access key ID and Secret access key. Download the .csv file or copy these credentials securely.**

**A screenshot of a computer

Description automatically generated**

**================ Creating IAM admin user completed =============**

**3. Install AWS CLI and Authenticate with AWS**

**A. Update your package repository:**

**# sudo yum update -y**

**B. Install the AWS CLI using the package manager:**

**# sudo yum install -y awscli**

**C. Verify the installation**

**# aws –version**

**4: Configure AWS CLI**

**A. Run the AWS configure command**

**# aws configure**

**B. Enter your AWS credentials:**

* **AWS Access Key ID: [Your Access Key ID]**
* **AWS Secret Access Key: [Your Secret Access Key]**
* **Default region name: [Your Preferred Region] (e.g., us-west-2)**
* **Default output format: [Your Preferred Format] (e.g., json)**

**A black screen with a black border

Description automatically generated**

**This will create a configuration file in your home directory at ~/.aws/credentials and ~/.aws/config.**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**==========Completed Install AWS CLI and Authenticate with AWS ====**

**GitHub Repo URL:** <https://github.com/asecurityguru/terraform-ec2-jenkins-aws-k8s-infra-creation.git>

**My GitHub URL**: <https://github.com/prasannaannavarapu/akondareddyguru.git>

Now infrastructure is setup ready awscli and terraform.

Clone the url into server : <https://github.com/prasannaannavarapu/akondareddyguru.git>

The below files are available in Repo. Please do some changes

* Vars/dev-west-2.tfvars
* README.md
* install\_jenkins.sh
* main.tf
* outputs.tf

1. under Vars/dev-west-2.tfvars folder change the **aws\_region** and **Key\_name**

2. Use below command to create a key

**# ssh-keygen -f akondakey**

**Now publicand private key is created and paste the key where terraform main.tf folder.**

**A screenshot of a computer program

Description automatically generated**

**Create the key in aws also**

**AWS console >> keypairs>> import key or create key>>**

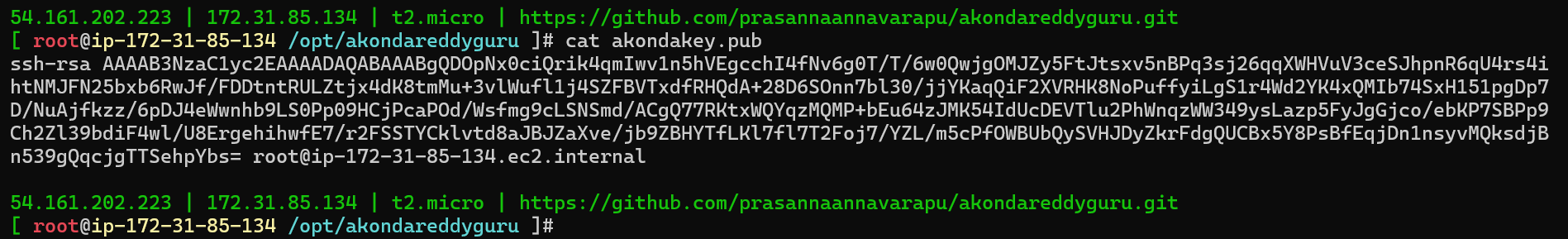
**A screenshot of a computer

Description automatically generated**

Go to the server where pubic and private keys are available

Copy the public key data paste in the aws import key section and click import key

# cat akondskey.pub



A screenshot of a computer

Description automatically generated

**A screenshot of a computer

Description automatically generated**

under Vars/dev-west-2.tfvars folder change the **aws\_region** and **Key\_name**

A screenshot of a computer program

Description automatically generated

Now run the terraform file with below commands

Step1: **Initialize Terraform**

# terraform init

Step 2: **Plan Resources**

# terraform plan -var-file="vars/dev-west-2.tfvars"

Step 3: **Apply Resources**

#terraform apply -var-file="vars/dev-west-2.tfvars"

Now got to aws console check ec2 instance will created.

Now copy the ip address login in web browser

A screenshot of a computer

Description automatically generated

**jenkins**

**Jenkins login**

**http://<IP address>:8080**

[**http://54.197.4.117:8081**](http://54.197.4.117:8081)

**Setup maven in Jenkins and copy the maven version and maven path from server**

**A computer screen with text

Description automatically generated with medium confidence**

**A screenshot of a computer

Description automatically generated**

**Download the below plugins**

**1. Docker pipeline**

**2. aws credentials**

**3.amazon ecr**

**4. Kubernetes cli**

**Sonar Cloud**

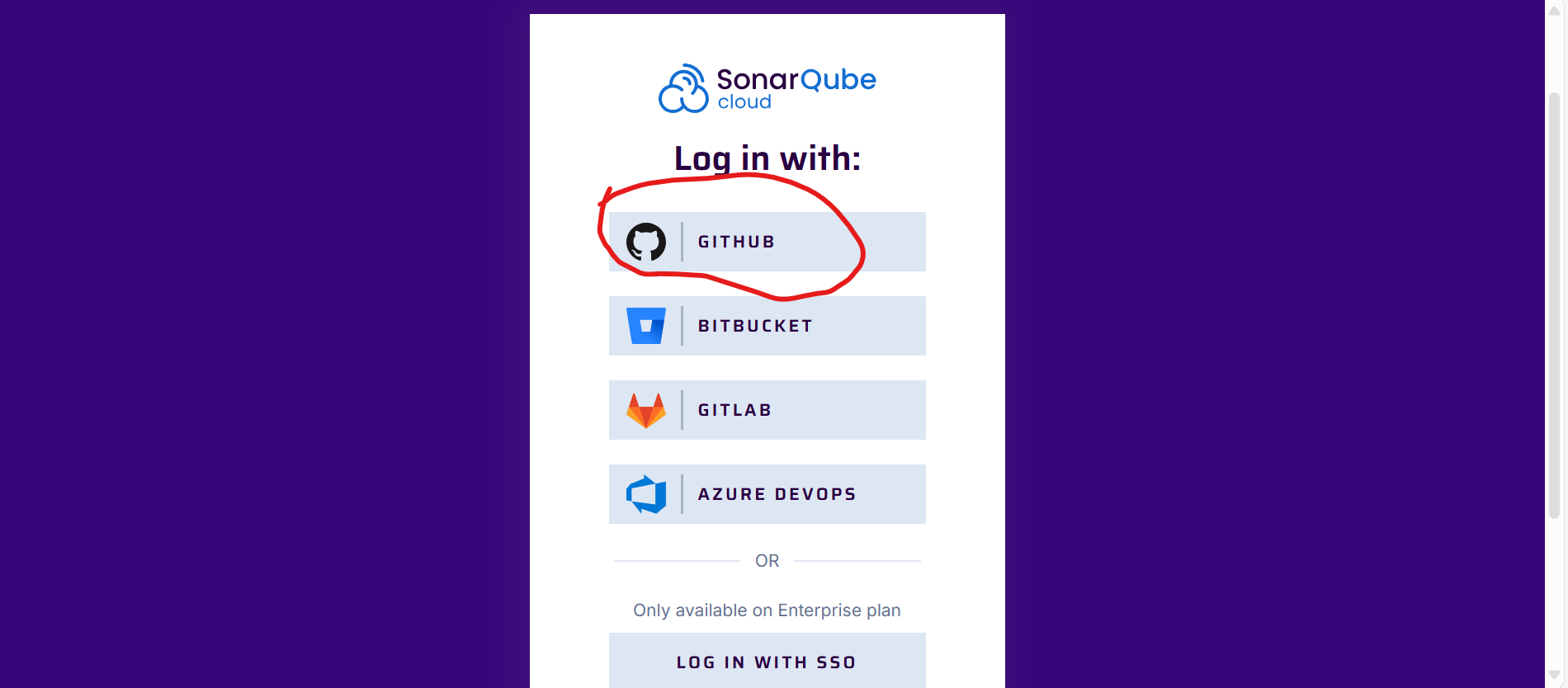
**Github URL: https://github.com/asecurityguru/devsecops-jenkins-k8s-tf-sast-sonarcloud-repo.git**

**Install Sonar cloud installation**

**A screenshot of a cell phone

Description automatically generated**

**Login with Github account credential**

****

**Click on authorize sonarqube cloud.**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**Integrate Sonarcloud with Jenkins**

**Craete New organisation in sonor cloud follow the below steps**

**Go to sonor cloud dashboard>> click + symbol>> create manual one>> give name and select plan>>**

**A screenshot of a computer

Description automatically generated**

**Select the free plan and click on the Create organisation**

**A screenshot of a computer

Description automatically generated**

**To Analize the new project key follow below steps**

**A screenshot of a chat

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**Snyk**

**Snyk Github URL:** [**https://github.com/asecurityguru/devsecops-jenkins-k8s-tf-sast-sca-sonarcloud-snyk-repo.git**](https://github.com/asecurityguru/devsecops-jenkins-k8s-tf-sast-sca-sonarcloud-snyk-repo.git)

**Two changes**

**1. pom.xml**

**2. Jenkinsfile**

**DePLOY ON K8**

**#eksctl create cluster --name kubernetes-cluster --version 1.32 --region us-east-1 --nodegroup-name linux-nodes --node-type t2.medium --nodes 2**

**#kubectl get nodes**

**#eksctl delete cluster --region=us-east-1 --name=kubernetes-cluster**

**1. Copy the .kube config file and save in locally as text file**

**Path : cat .kube/config**

**Open the Jenkins crdetials select the secret as file and load the local text file**

**Trigger the deployment**

**#Kubectl create namespace devsecops**

**#kubectl get ns**

**#** **kubectl get deploymnet -n devsecops**

**#** **kubectl get pods -n devsecops**

**http://** **ac7f210b654744d7eb517ca6d07168da-1937262070.us-east-1.elb.amazonaws.com**

**this page loaded correctly**

**#** **kubectl get services/asgbuggy --namespace=devsecops -o json| jq -r ".status.loadBalancer.ingress[] | .hostname"**

**Displayedafter load balanceurl**

**ac7f210b654744d7eb517ca6d07168da-1937262070.us-east-1.elb.amazonaws.com**