DevSecOps CI/CD : Deploying a Secure Hotstar Clone

GITHUB : https://github.com/Bijan1235/Hotstar-Clone.git

Prerequisites

- AWS account setup
- Basic knowledge of AWS services
- Understanding of DevSecOps principles
- Familiarity with Docker, Jenkins, Java, SonarQube, AWS CLI, Kubectl, and Terraform,Docker Scout

Step-by-Step Deployment Process

Step 1: Setting up AWS EC2 Instance

- Creating an EC2 instance with Ubuntu AMI, t2.large, and 30 GB storage
- Assigning an IAM role with Admin access for learning purposes

Step 2: Installation of Required Tools on the Instance

- Writing a script to automate the installation of:
  - Docker
  - Jenkins
  - Java
  - SonarQube container
  - AWS CLI
  - Kubectl
  - Terraform

Step 3: Jenkins Job Configuration

- Creating Jenkins jobs for:
  - Creating an EKS cluster
  - Deploying the Hotstar clone application
- Configuring the Jenkins job stages:
  - Sending files to SonarQube for static code analysis
  - Running npm install
  - Implementing OWASP for security checks
  - Installing and running Docker Scout for container security
  - Scanning files and Docker images with Docker Scout
  - Building and pushing Docker images
  - Deploying the application to the EKS cluster

Step 4: Clean-Up Process

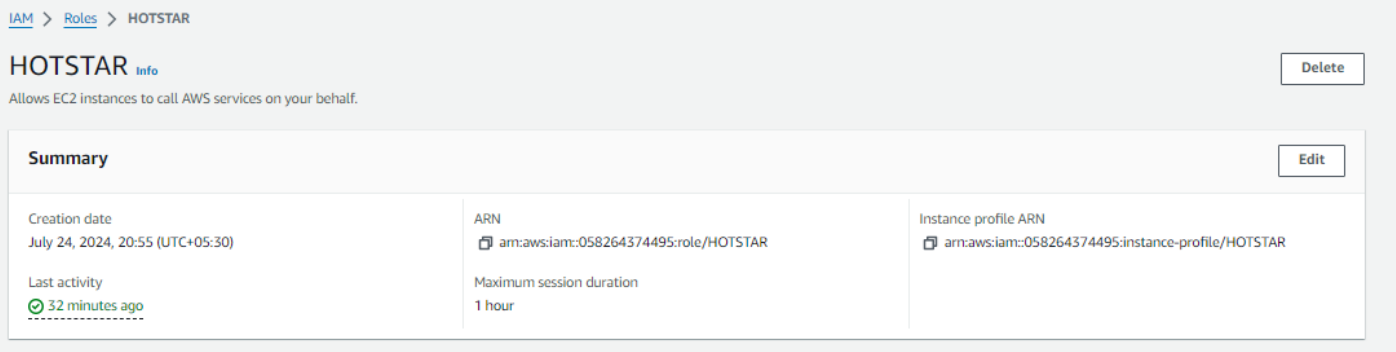
- Removing the EKS cluster
- Deleting the IAM role
- Terminating the Ubuntu instance

STEP 1A: Setting up AWS EC2 Instance and IAM Role

1. **Sign in to the AWS Management Console:** Access the AWS Management Console using your credentials
2. **Navigate to the EC2 Dashboard:** Click on the "Services" menu at the top of the page and select "EC2" under the "Compute" section. This will take you to the EC2 Dashboard.
3. **Launch Instance:** Click on the "Instances" link on the left sidebar and then click the "Launch Instance" button.
4. **Choose an Amazon Machine Image (AMI):** In the "Step 1: Choose an Amazon Machine Image (AMI)" section:
    - Select "AWS Marketplace" from the left-hand sidebar.
    - Search for "Ubuntu" in the search bar and choose the desired Ubuntu AMI (e.g., Ubuntu Server 24.04 LTS).
    - Click on "Select" to proceed.
5. **Choose an Instance Type:** In the "Step 2: Choose an Instance Type" section:
    - Scroll through the instance types and select "t2.large" from the list.
    - Click on "Next: Configure Instance Details" at the bottom.
6. **Configure Instance Details:** In the "Step 3: Configure Instance Details" section, you can leave most settings as default for now. However, you can configure settings like the network, subnet, IAM role, etc., according to your requirements.
    - Once done, click on "Next: Add Storage."
7. **Add Storage:** In the "Step 4: Add Storage" section:
    - You can set the size of the root volume (usually /dev/sda1) to 30 GB by specifying the desired size in the "Size (GiB)" field.
    - Customize other storage settings if needed.
    - Click on "Next: Add Tags" when finished.
8. **Add Tags (Optional):** In the "Step 5: Add Tags" section, you can add tags to your instance for better identification and management. This step is optional but recommended for organizational purposes.
    - Click on "Next: Configure Security Group" when done.
9. **Configure Security Group:** In the "Step 6: Configure Security Group" section:
    - Create a new security group or select an existing one.
    - Ensure that at least SSH (port 22) is open for inbound traffic to allow remote access.
    - You might also want to open other ports as needed for your application's requirements.
    - Click on "Review and Launch" when finished.
10. **Review and Launch:** Review the configuration details of your instance. If everything looks good:
    - Click on "Launch" to proceed.
    - A pop-up will prompt you to select or create a key pair. Choose an existing key pair or create a new one.
    - Finally, click on "Launch Instances."
11. **Accessing the Instance:** Once the instance is launched, you can connect to it using SSH. Use the private key associated with the selected key pair to connect to the instance's public IP or DNS address.

STEP 1B: IAM ROLE

1. Search for IAM in the search bar of AWS and Click on Create Role
2. Select entity type as AWS service
3. Use case as EC2 and click on Next.
4. For permission policy select Administrator Access (Just for learning purpose), click Next.
5. Provide a Name for Role and click on Create role.



IAM > Roles > HOTSTAR

HOTSTAR Info

Allows EC2 instances to call AWS services on your behalf.

Summary

Creation date
July 24, 2024, 20:55 (UTC+05:30)

ARN
arn:aws:iam::058264374495:role/HOSTAR

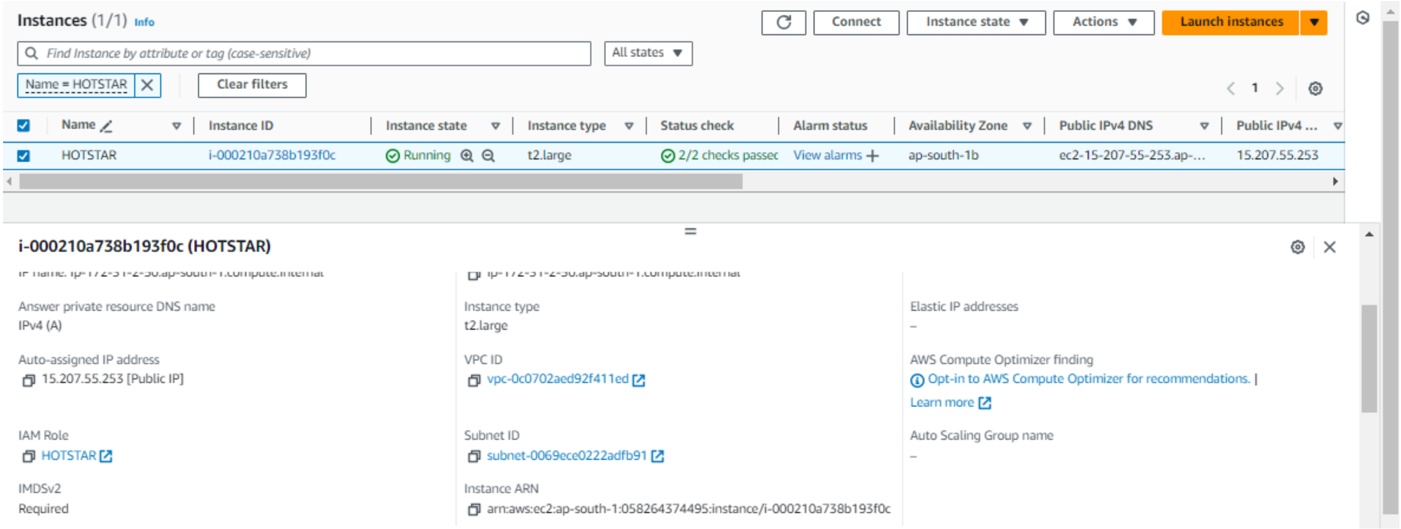
Instance profile ARN
arn:aws:iam::058264374495:instance-profile/HOTSTAR

Last activity
32 minutes ago

Maximum session duration
1 hour

Now Attach this role to EC2 Instance that we created earlier, so we can provision cluster from that instance.

Click on Actions –> Security –> Modify IAM role.



☑	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...
☑	HOTSTAR	i-000210a738b193f0c	Running	t2.large	2/2 checks passec	View alarms +	ap-south-1b	ec2-15-207-55-253.ap-...	15.207.55.253

i-000210a738b193f0c (HOTSTAR)

IP name: ip-172-31-2-50.ap-south-1.compute.internal

ip-172-31-2-50.ap-south-1.compute.internal

Answer private resource DNS name
IPv4 (A)

Instance type
t2.large

Elastic IP addresses
–

Auto-assigned IP address
15.207.55.253 [Public IP]

VPC ID
vpc-0c0702aed92f411ed

AWS Compute Optimizer finding
Opt-in to AWS Compute Optimizer for recommendations. | Learn more

IAM Role
HOTSTAR

Subnet ID
subnet-0069ece0222adfb91

Auto Scaling Group name
–

IMDSv2
Required

Instance ARN
arn:aws:ec2:ap-south-1:058264374495:instance/i-000210a738b193f0c

Step 2: Installation of Required Tools on the Instance

vi script1.sh

```
#!/bin/bash
sudo apt update -y
sudo apt install openjdk-17-jre -y
sudo wget -O /usr/share/keyrings/jenkins-keyring.asc \
  https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key
echo "deb [signed-by=/usr/share/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
```

Now make the script1.sh executable;

```
sudo chmod +x script1.sh
```

Now apply by using below command;

```
./script1.sh
```

vi script2.sh

```
#!/bin/bash
sudo apt update -y
sudo apt-get update
sudo apt install docker.io -y
sudo chmod 666 /var/run/docker.sock
sudo apt-get install -y apt-transport-https ca-certificates curl gpg
sudo mkdir -p -m 755 /etc/apt/keyrings
curl -fsSL https://pkgs.k8s.io/core:/stable:/v1.28/deb/Release.key | sudo gpg --dearmor -o
/etc/apt/keyrings/kubernetes-apt-keyring.gpg
echo 'deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg]
https://pkgs.k8s.io/core:/stable:/v1.28/deb/ /' | sudo tee /etc/apt/sources.list.d/kubernetes.list
sudo apt-get update
sudo apt-get install -y kubelet kubeadm kubectl
sudo systemctl enable --now kubelet


#install terraform

sudo apt-get install -y gnupg software-properties-common
wget -O- https://apt.releases.hashicorp.com/gpg | \
gpg --dearmor | \
sudo tee /usr/share/keyrings/hashicorp-archive-keyring.gpg > /dev/null
gpg --no-default-keyring \
--keyring /usr/share/keyrings/hashicorp-archive-keyring.gpg \
--fingerprint
echo "deb [signed-by=/usr/share/keyrings/hashicorp-archive-keyring.gpg] \
https://apt.releases.hashicorp.com $(lsb_release -cs) main" | \
sudo tee /etc/apt/sources.list.d/hashicorp.list
sudo apt update && sudo apt-get install terraform


#install Aws cli

curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip"
sudo apt install unzip
unzip awscliv2.zip
sudo ./aws/install
```

Now make the script2.sh executable;

```
sudo chmod +x script2.sh
```

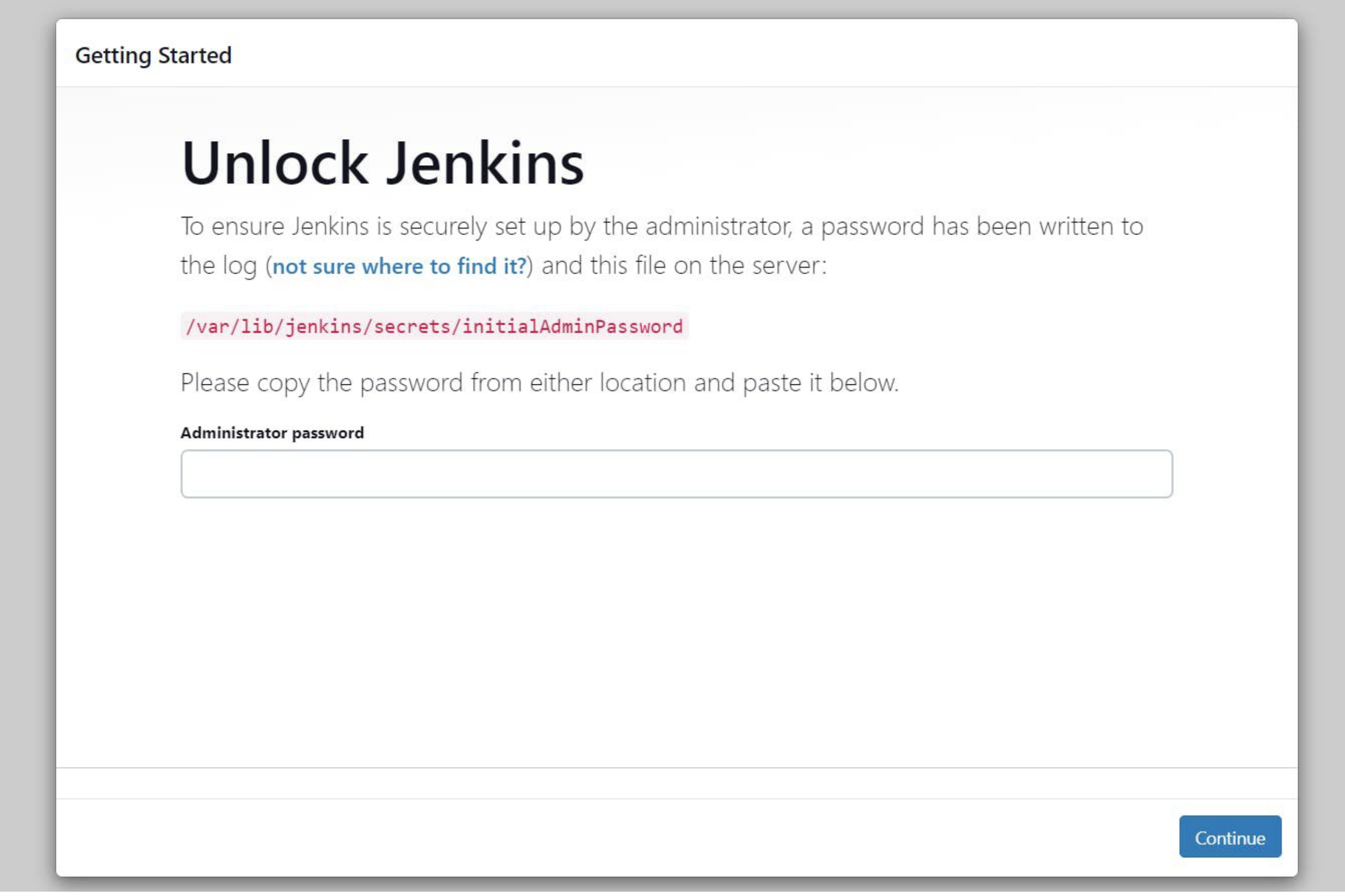
Now apply by using below command;

```
./script2.sh
```

Now time to SonarQube installation;

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

Now copy the public IP address of ec2 and paste it into the browser with :8080 for jenkins



Getting Started

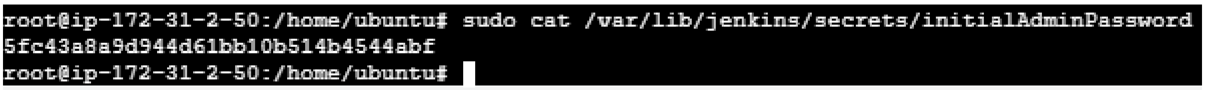
Unlock Jenkins

To ensure Jenkins is securely set up by the administrator, a password has been written to the log (not sure where to find it?) and this file on the server:

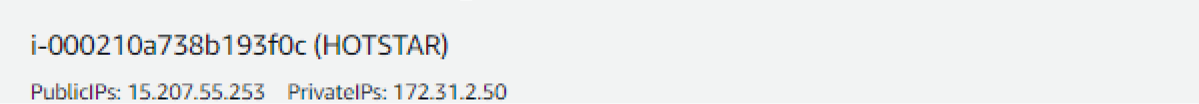
`/var/lib/jenkins/secrets/initialAdminPassword`

Please copy the password from either location and paste it below.

Administrator password



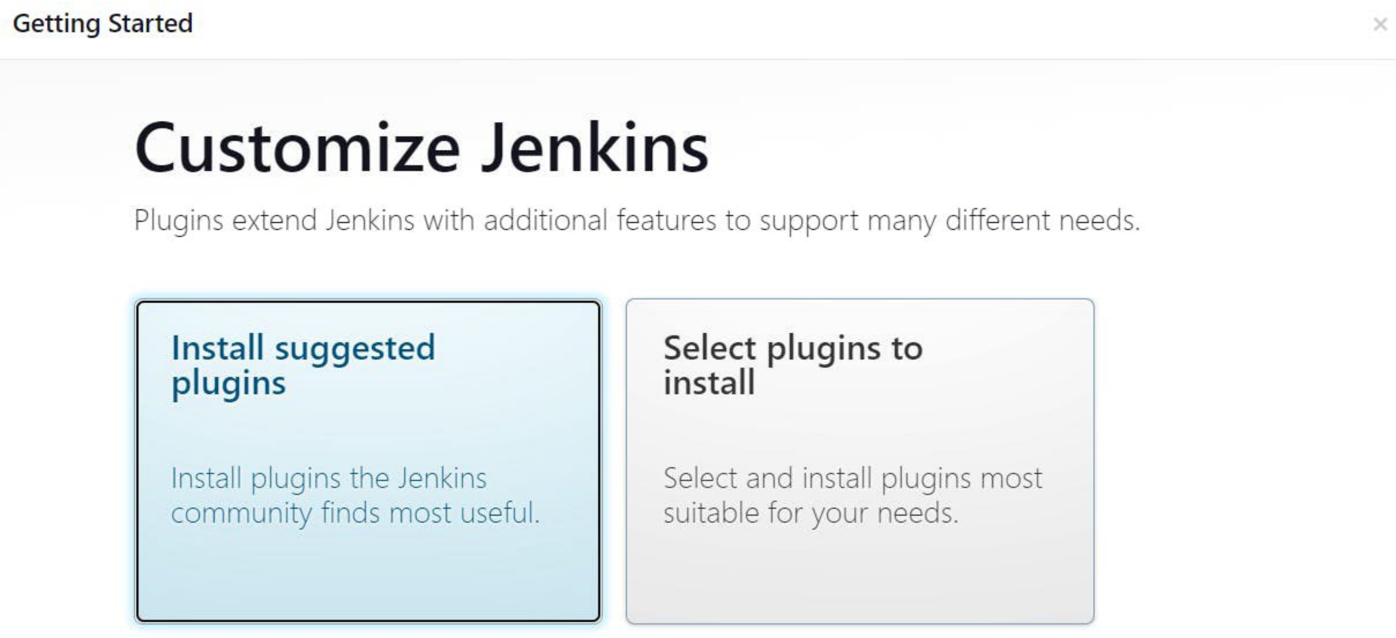
```
root@ip-172-31-2-50:/home/ubuntu# sudo cat /var/lib/jenkins/secrets/initialAdminPassword
5fc43a8a9d944d61bb10b514b4544abf
root@ip-172-31-2-50:/home/ubuntu# 
```



i-000210a738b193f0c (HOTSTAR)

PublicIPs: 15.207.55.253   PrivateIPs: 172.31.2.50

Now, install the suggested plugins.

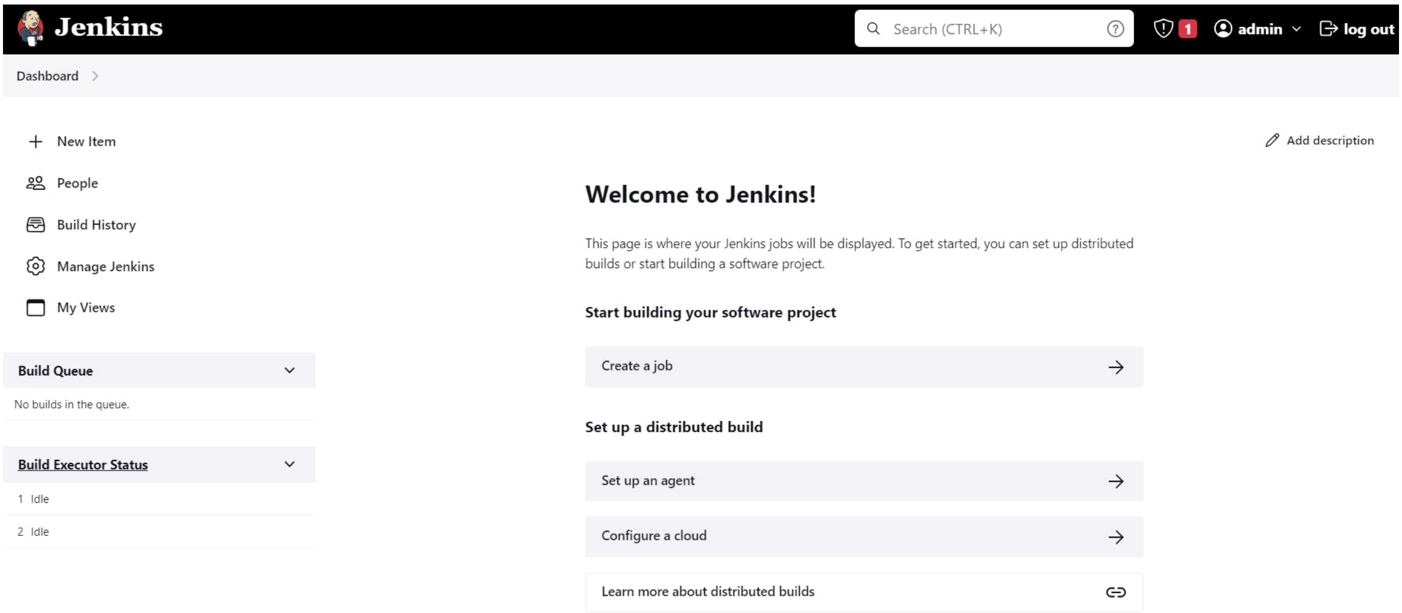


Getting Started

Customize Jenkins

Plugins extend Jenkins with additional features to support many different needs.

×
Install suggested plugins
Install plugins the Jenkins community finds most useful.
Select plugins to install
Select and install plugins most suitable for your needs.



Build Queue

No builds in the queue.

Build Executor Status

1 Idle

2 Idle

Welcome to Jenkins!

This page is where your Jenkins jobs will be displayed. To get started, you can set up distributed builds or start building a software project.

Start building your software project

Create a job

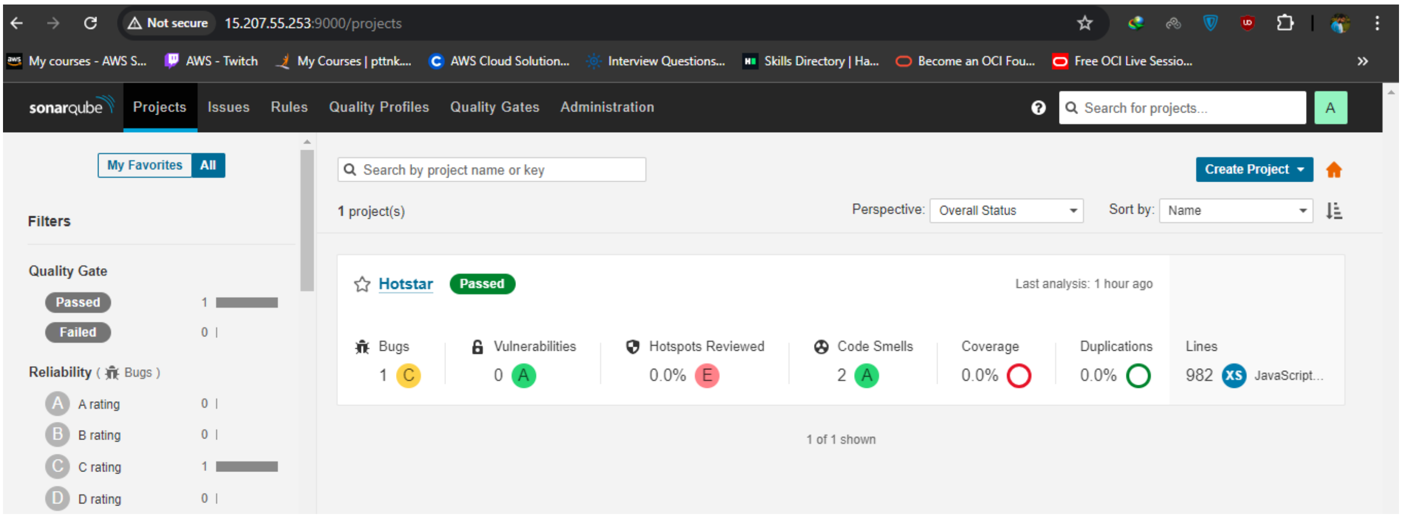
Set up a distributed build

Set up an agent

Configure a cloud

Learn more about distributed builds

Now Copy the public IP again and paste it into a new tab in the browser with 9000



Filters

Quality Gate

Passed	1
Failed	0

Reliability (Bugs)

A rating	0
B rating	0
C rating	1
D rating	0

1 project(s)

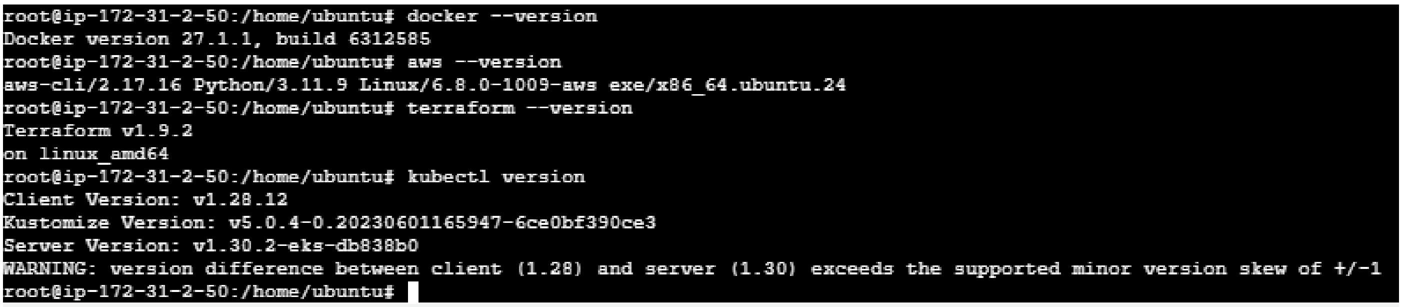
Perspective: Overall Status Sort by: Name

Hotstar Passed Last analysis: 1 hour ago

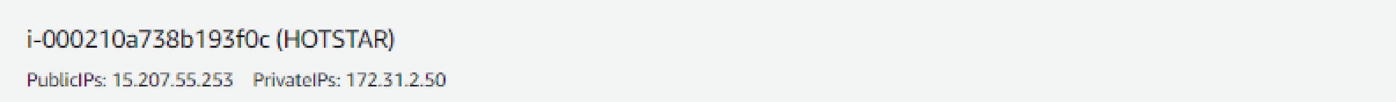
Bugs	Vulnerabilities	Hotspots Reviewed	Code Smells	Coverage	Duplications	Lines
1 C	0 A	0.0% E	2 A	0.0%	0.0%	982 XS JavaScript...

1 of 1 shown

Note:- Initial username & password both are admin, you will have to change as per your choice



```
root@ip-172-31-2-50:/home/ubuntu# docker --version
Docker version 27.1.1, build 6312585
root@ip-172-31-2-50:/home/ubuntu# aws --version
aws-cli/2.17.16 Python/3.11.9 Linux/6.8.0-1009-aws exe/x86_64.ubuntu.24
root@ip-172-31-2-50:/home/ubuntu# terraform --version
Terraform v1.9.2
on linux_amd64
root@ip-172-31-2-50:/home/ubuntu# kubectl version
Client Version: v1.28.12
Kustomize Version: v5.0.4-0.20230601165947-6ce0bf390ce3
Server Version: v1.30.2-eks-db838b0
WARNING: version difference between client (1.28) and server (1.30) exceeds the supported minor version skew of +/-1
root@ip-172-31-2-50:/home/ubuntu#
```



i-000210a738b193f0c (HOTSTAR)

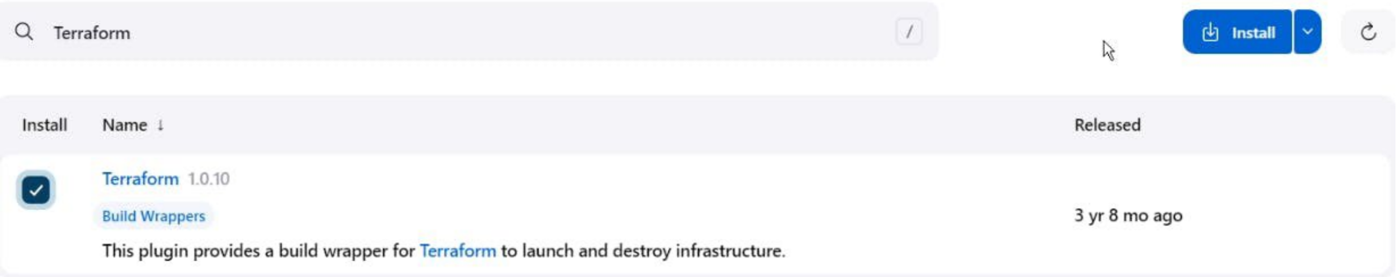
PublicIPs: 15.207.55.253 PrivateIPs: 172.31.2.50

Step 3A: Jenkins Job Configuration

That is done now go to Jenkins and add a terraform plugin to provision the AWS EKS using the Pipeline Job.

Go to Jenkins dashboard –> Manage Jenkins –> Plugins

Available Plugins, Search for Terraform and install it.

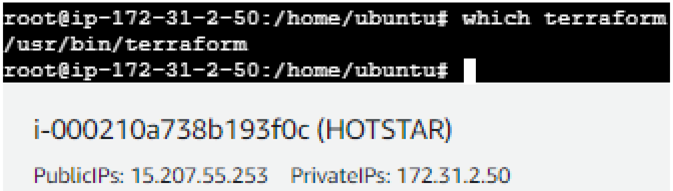


☑ Terraform 1.0.10 — Build Wrappers — 3 yr 8 mo ago

This plugin provides a build wrapper for Terraform to launch and destroy infrastructure.

let’s find the path to our Terraform (we will use it in the tools section of Terraform)

which terraform



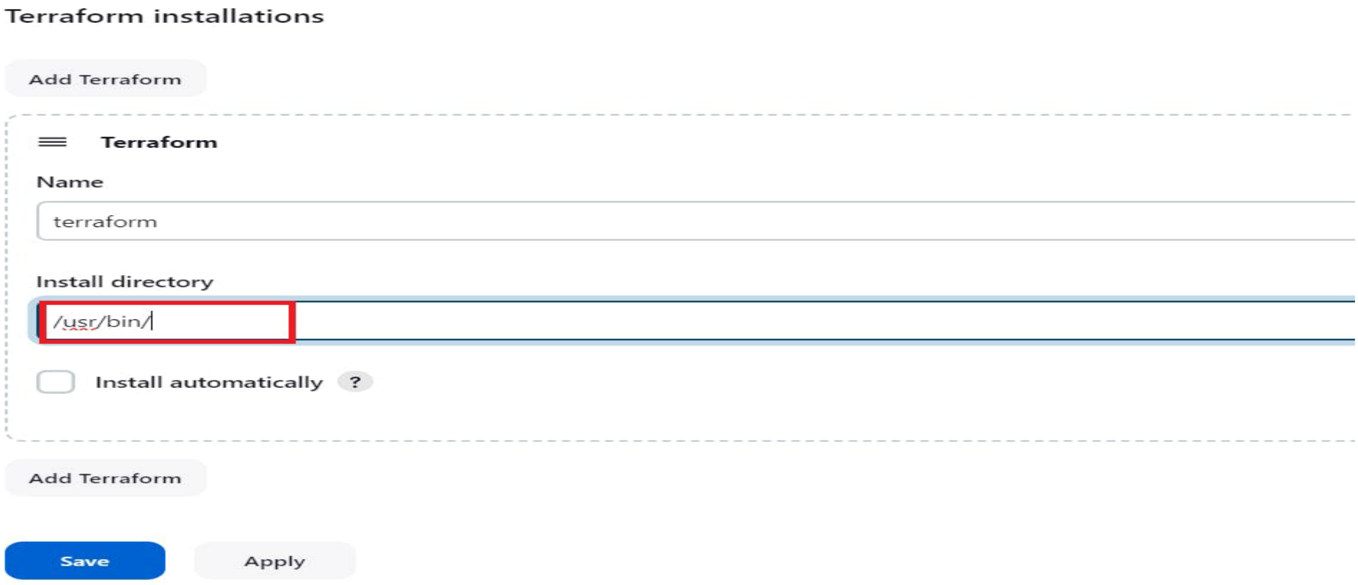
```
root@ip-172-31-2-50:/home/ubuntu# which terraform
/usr/bin/terraform
root@ip-172-31-2-50:/home/ubuntu#
```

i-000210a738b193f0c (HOTSTAR)

PublicIPs: 15.207.55.253 PrivateIPs: 172.31.2.50

Now come back to Manage Jenkins –> Tools

Add the terraform in Tools



Terraform installations

Terraform

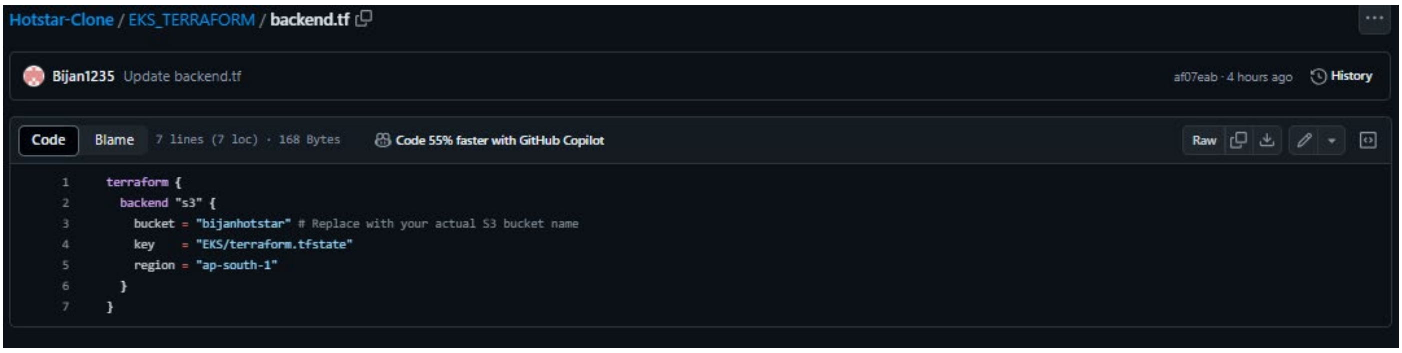
Name: terraform

Install directory: /usr/bin/

□ Install automatically ?

Apply and save.

CHANGE YOUR S3 BUCKET NAME IN THE BACKEND.TF



Hotstar-Clone / EKS_TERRAFORM / backend.tf

Bijan1235 Update backend.tf af07eab · 4 hours ago

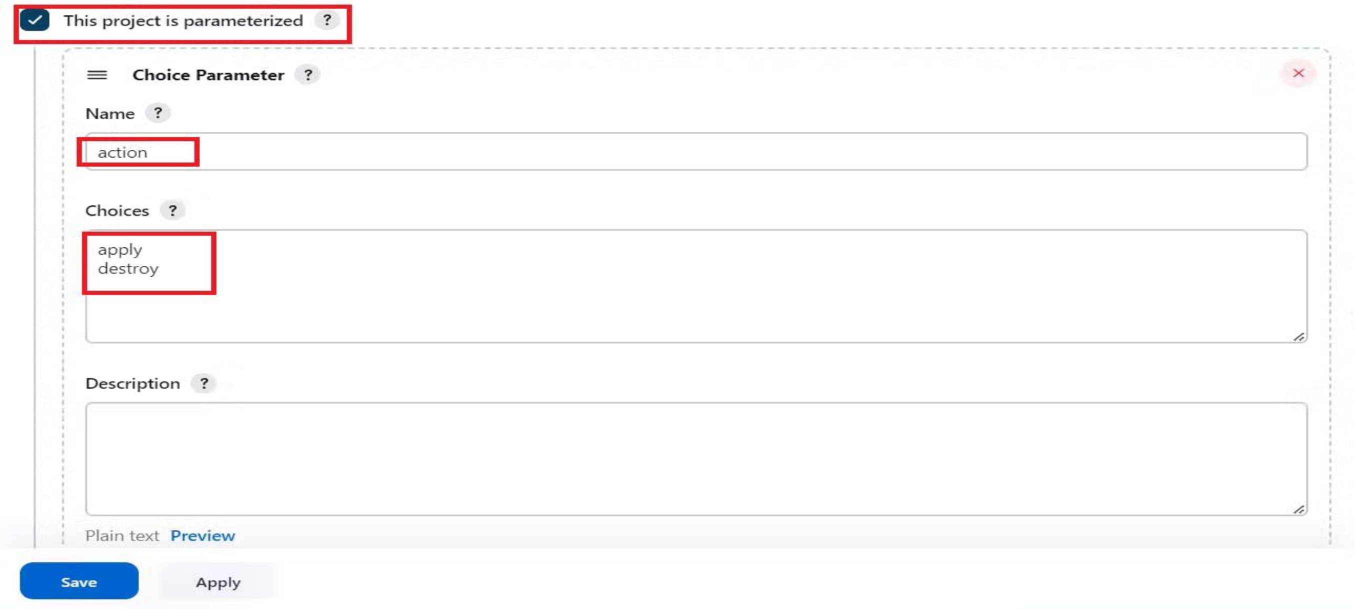
7 lines (7 loc) · 168 Bytes Code 55% faster with GitHub Copilot

```
terraform {
  backend "s3" {
    bucket = "bijanhotstar" # Replace with your actual S3 bucket name
    key    = "EKS/terraform.tfstate"
    region = "ap-south-1"
  }
}
```

Now create a new job for the Eks provision

I want to do this with build parameters to apply and destroy while building only.

you have to add this inside job like the below image



☑ This project is parameterized

Choice Parameter

Name: action

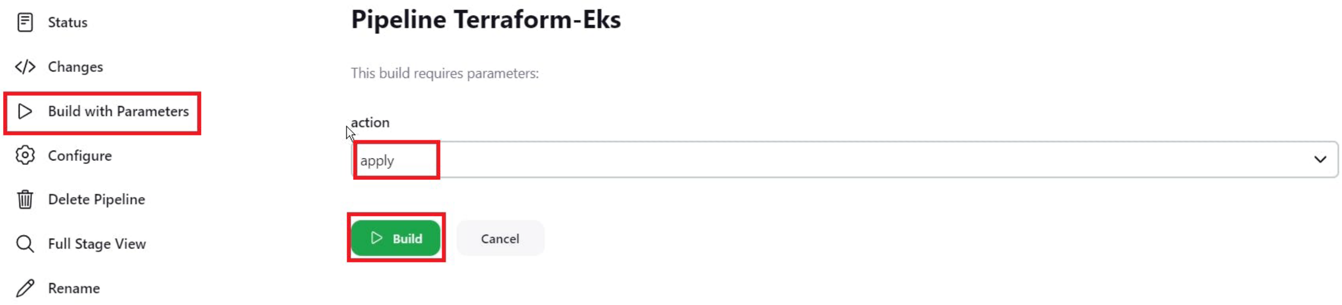
Choices:
apply
destroy

Description:

Let’s add a pipeline

```
pipeline{
    agent any
    stages {
        stage('Checkout from Git'){
            steps{
                git branch: 'main', url: ' https://github.com/Bijan1235/Hotstar-Clone.git'
            }
        }
        stage('Terraform version'){
             steps{
                 sh 'terraform --version'
             }
        }
        stage('Terraform init'){
             steps{
                 dir('EKS_TERRAFORM') {
                      sh 'terraform init'
                   }
             }
        }
        stage('Terraform validate'){
            steps{
                dir('EKS_TERRAFORM') {
                    sh 'terraform validate'
                  }
            }
        }
        stage('Terraform plan'){
            steps{
                dir('EKS_TERRAFORM') {
                    sh 'terraform plan'
                  }
            }
        }
        stage('Terraform apply/destroy'){
            steps{
                dir('EKS_TERRAFORM') {
                    sh 'terraform ${action} --auto-approve'
                  }
            }
        }
    }
}
```

let’s apply and save and Build with parameters and select action as apply

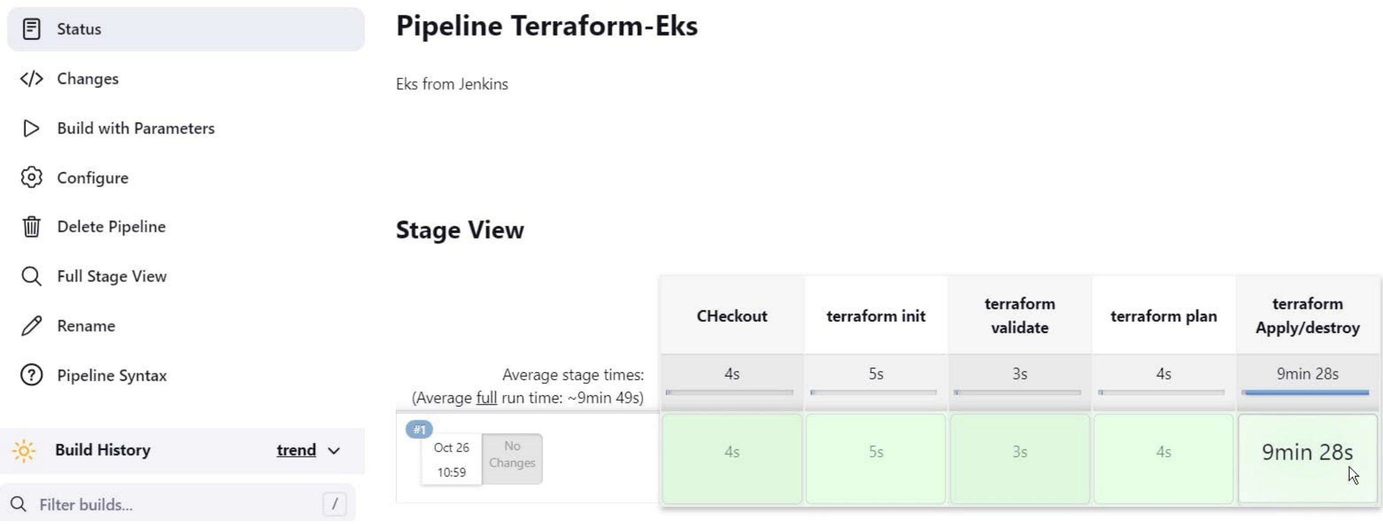


Pipeline Terraform-Eks

This build requires parameters:

action

apply



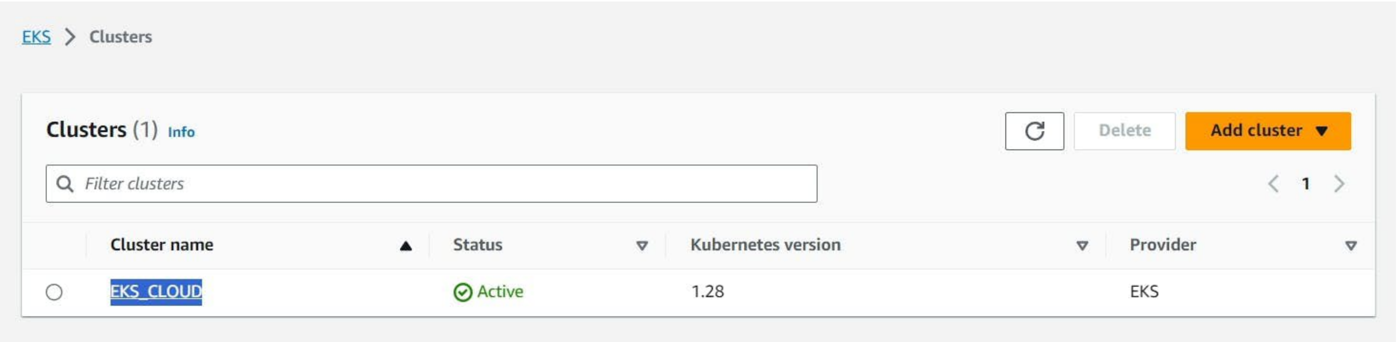
Pipeline Terraform-Eks

Eks from Jenkins

Stage View

	CHeckout	terraform init	terraform validate	terraform plan	terraform Apply/destroy
Average stage times: (Average full run time: ~9min 49s)	4s	5s	3s	4s	9min 28s
#1 Oct 26 10:59 No Changes	4s	5s	3s	4s	9min 28s

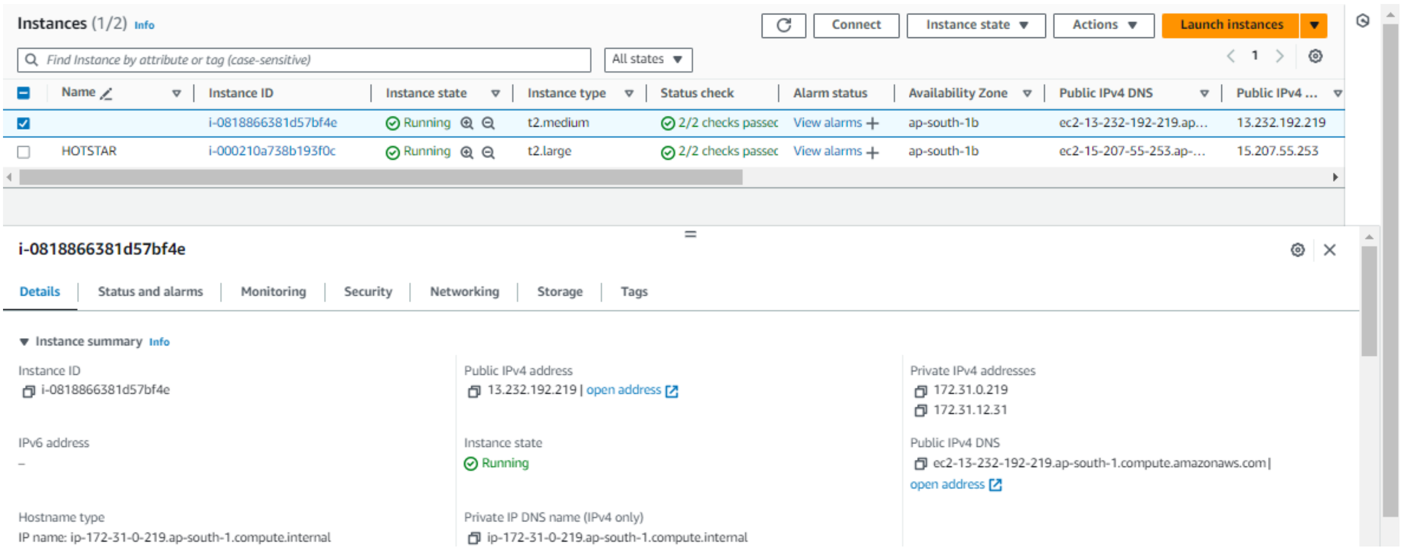
Check in Your Aws console whether it created EKS or not.



Clusters (1) Info

	Cluster name	Status	Kubernetes version	Provider
○	EKS_CLOUD	Active	1.28	EKS

Ec2 instance is created for the Node group



Instances (1/2) Info

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...
☑		i-0818866381d57bf4e	Running	t2.medium	2/2 checks passec	View alarms +	ap-south-1b	ec2-13-232-192-219.ap...	13.232.192.219
□	HOTSTAR	i-000210a738b193f0c	Running	t2.large	2/2 checks passec	View alarms +	ap-south-1b	ec2-15-207-55-253.ap-...	15.207.55.253

i-0818866381d57bf4e

Instance summary Info

Instance ID: i-0818866381d57bf4e

Public IPv4 address: 13.232.192.219 | open address

Private IPv4 addresses: 172.31.0.219, 172.31.12.31

IPv6 address: –

Instance state: Running

Public IPv4 DNS: ec2-13-232-192-219.ap-south-1.compute.amazonaws.com | open address

Hostname type: IP name: ip-172-31-0-219.ap-south-1.compute.internal

Private IP DNS name (IPv4 only): ip-172-31-0-219.ap-south-1.compute.internal

Step 3B: Hotstar job

Go to Jenkins dashboard

Manage Jenkins –> Plugins –> Available Plugins

Search for the Below Plugins

Eclipse Temurin installer

Sonarqube Scanner

NodeJs

Owasp Dependency-Check

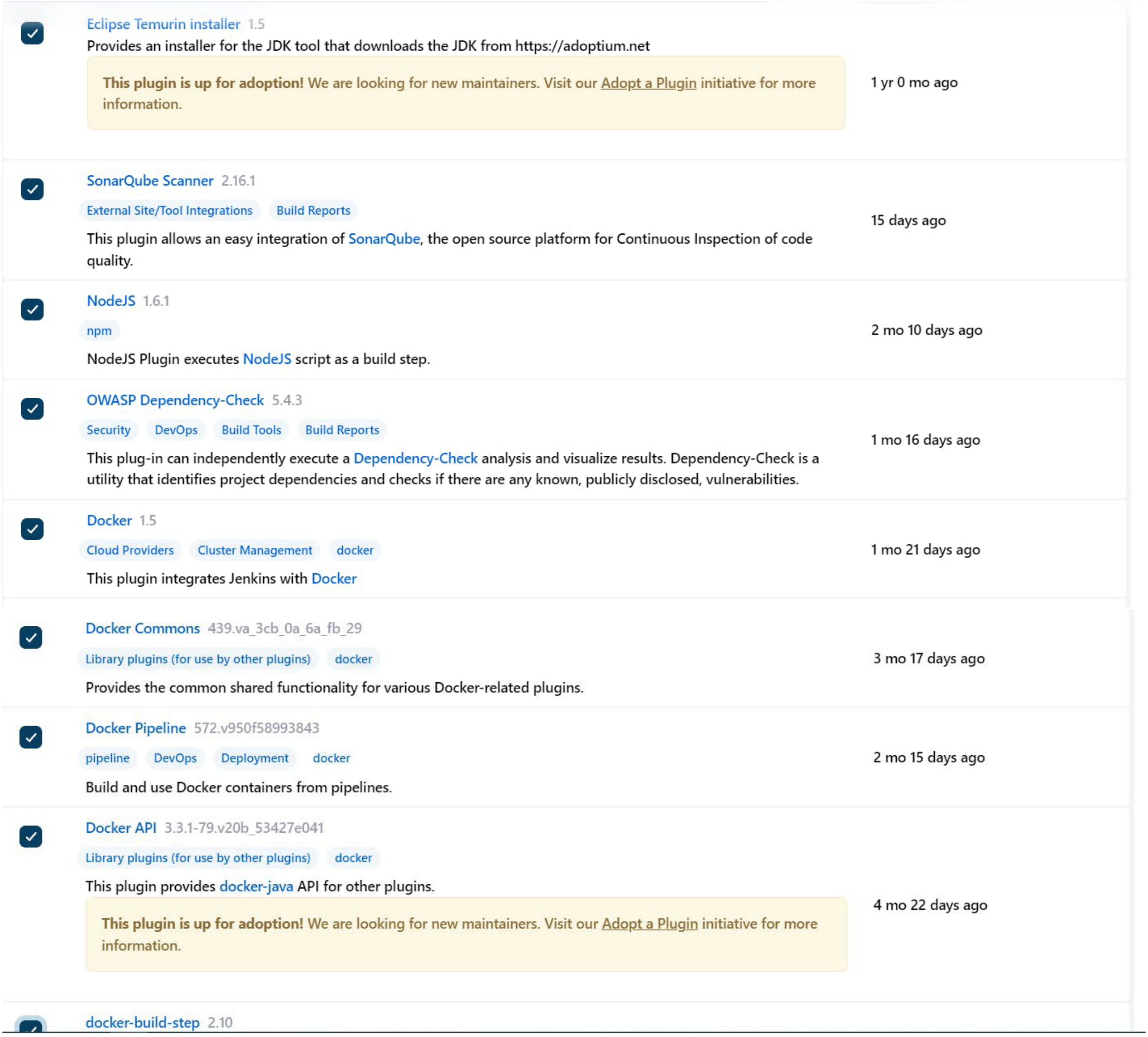
Docker

Docker Commons

Docker Pipeline

Docker API

Docker-build-step



☑ Eclipse Temurin installer 1.5
Provides an installer for the JDK tool that downloads the JDK from https://adoptium.net
This plugin is up for adoption! We are looking for new maintainers. Visit our Adopt a Plugin initiative for more information.
1 yr 0 mo ago

☑ SonarQube Scanner 2.16.1
External Site/Tool Integrations · Build Reports
This plugin allows an easy integration of SonarQube, the open source platform for Continuous Inspection of code quality.
15 days ago

☑ NodeJS 1.6.1
npm
NodeJS Plugin executes NodeJS script as a build step.
2 mo 10 days ago

☑ OWASP Dependency-Check 5.4.3
Security · DevOps · Build Tools · Build Reports
This plug-in can independently execute a Dependency-Check analysis and visualize results. Dependency-Check is a utility that identifies project dependencies and checks if there are any known, publicly disclosed, vulnerabilities.
1 mo 16 days ago

☑ Docker 1.5
Cloud Providers · Cluster Management · docker
This plugin integrates Jenkins with Docker
1 mo 21 days ago

☑ Docker Commons 439.va_3cb_0a_6a_fb_29
Library plugins (for use by other plugins) · docker
Provides the common shared functionality for various Docker-related plugins.
3 mo 17 days ago

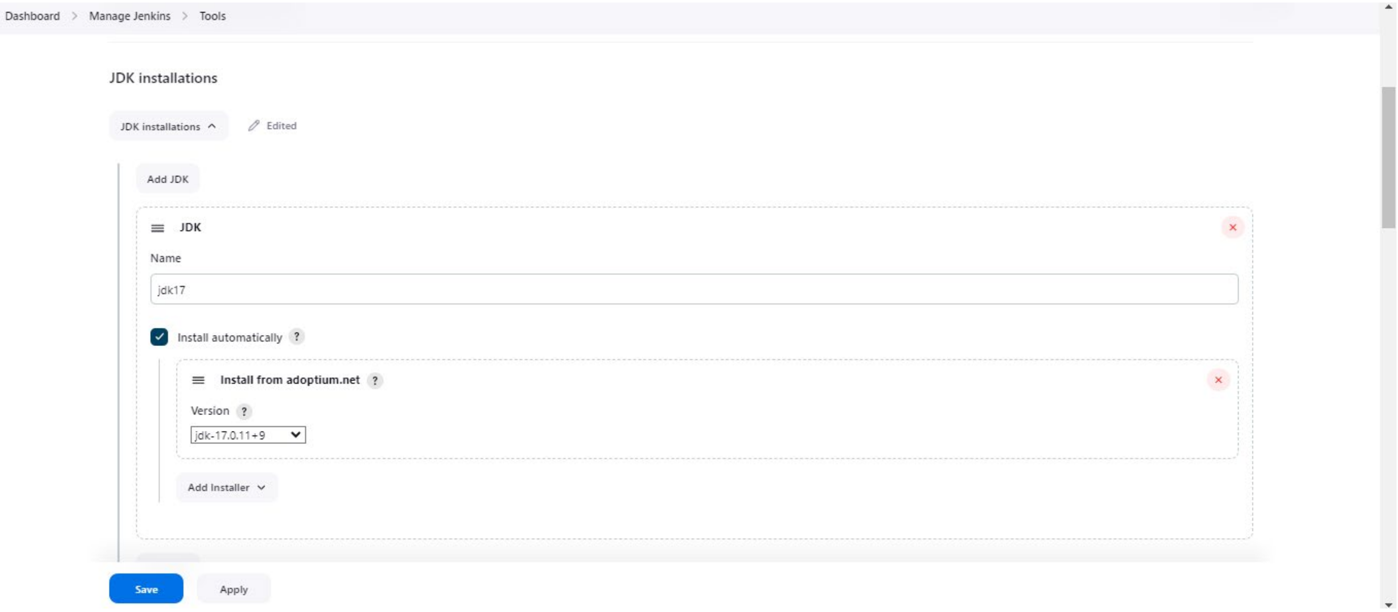
☑ Docker Pipeline 572.v950f58993843
pipeline · DevOps · Deployment · docker
Build and use Docker containers from pipelines.
2 mo 15 days ago

☑ Docker API 3.3.1-79.v20b_53427e041
Library plugins (for use by other plugins) · docker
This plugin provides docker-java API for other plugins.
This plugin is up for adoption! We are looking for new maintainers. Visit our Adopt a Plugin initiative for more information.
4 mo 22 days ago

☑ docker-build-step 2.10

Configure in Global Tool Configuration

Go to Manage Jenkins → Tools → Install JDK(17) and NodeJs(22)→ Click on Apply and Save



JDK installations

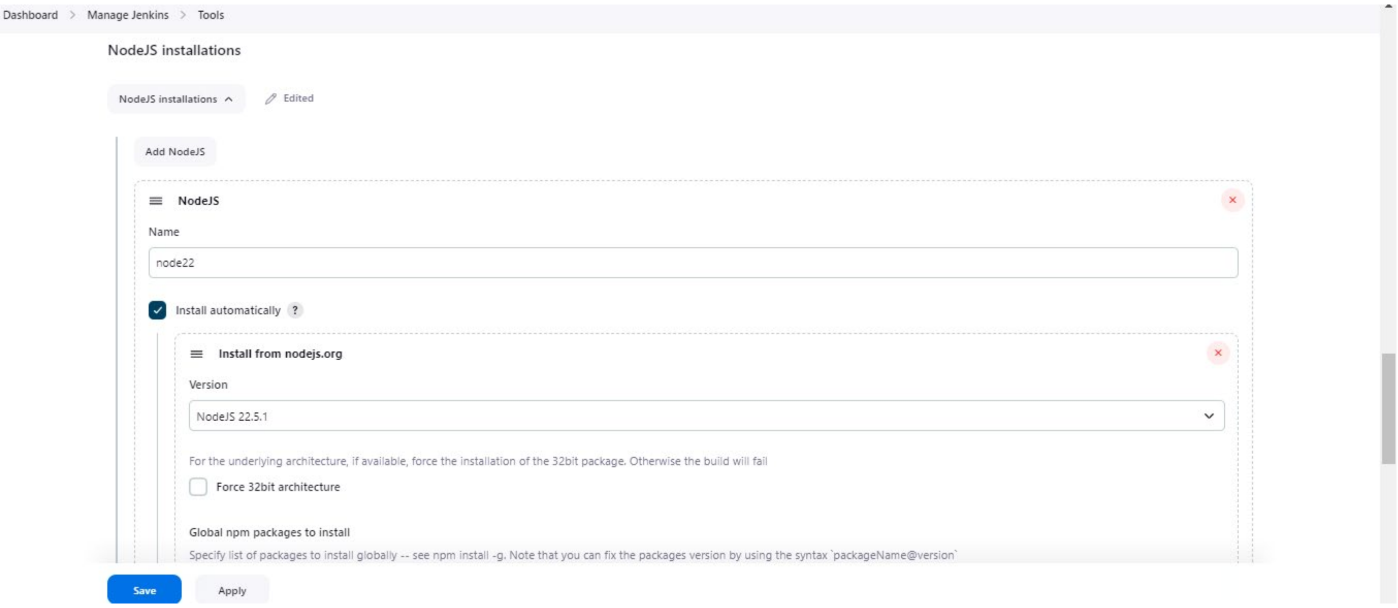
JDK

Name: jdk17

☑ Install automatically ?

Install from adoptium.net ?

Version ?: jdk-17.0.11+9



NodeJS installations

NodeJS

Name: node22

☑ Install automatically ?

Install from nodejs.org

Version: NodeJS 22.5.1

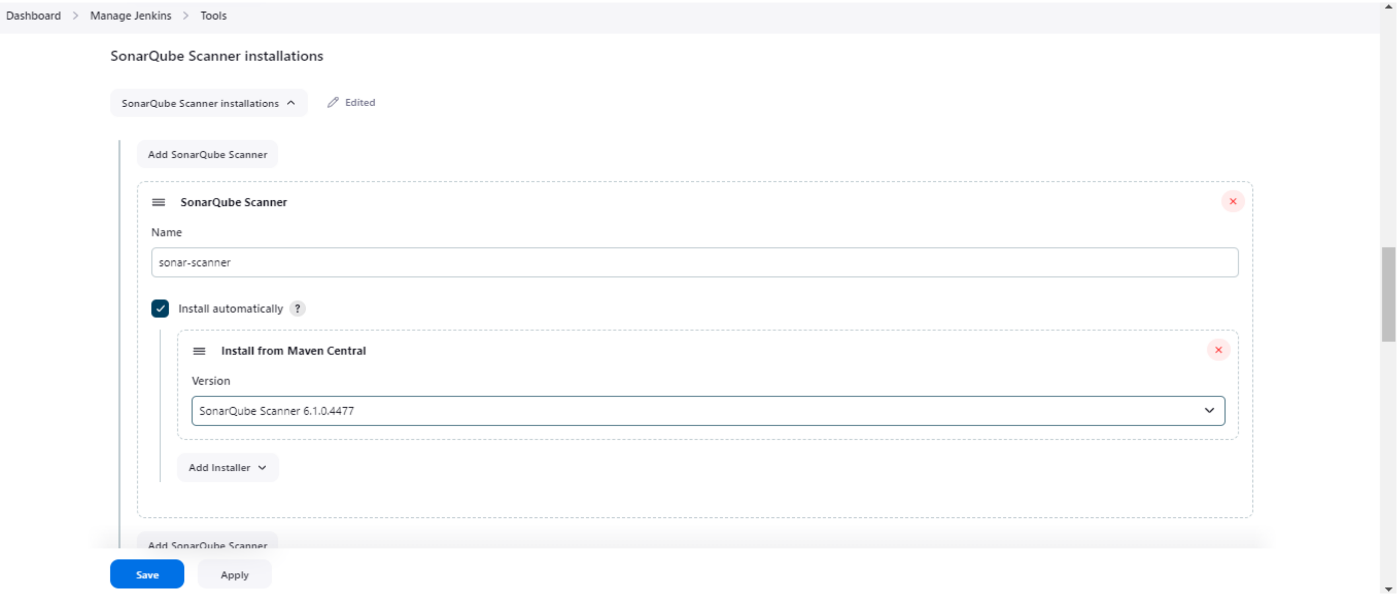
For the underlying architecture, if available, force the installation of the 32bit package. Otherwise the build will fail

□ Force 32bit architecture

Global npm packages to install

Specify list of packages to install globally -- see npm install -g. Note that you can fix the packages version by using the syntax `packageName@version`

For Sonarqube use the latest version



SonarQube Scanner installations

SonarQube Scanner

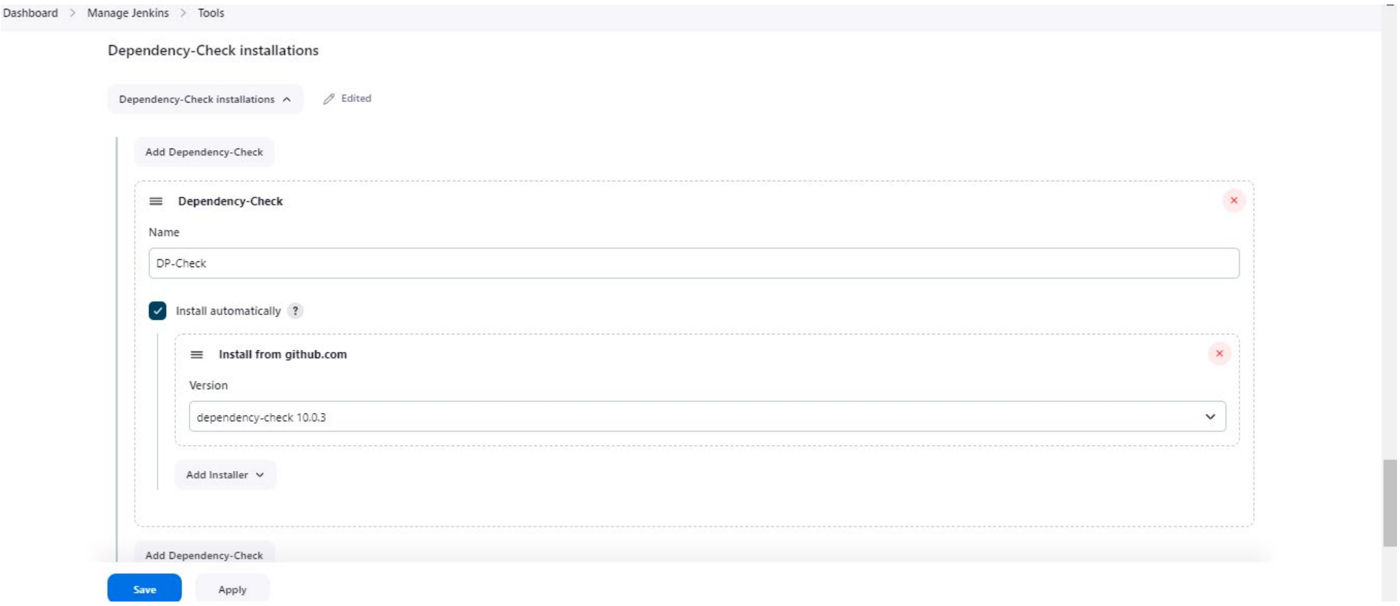
Name: sonar-scanner

☑ Install automatically ?

Install from Maven Central

Version: SonarQube Scanner 6.1.0.4477

For Owasp use the 10.0.3 version



Dependency-Check installations

Dependency-Check

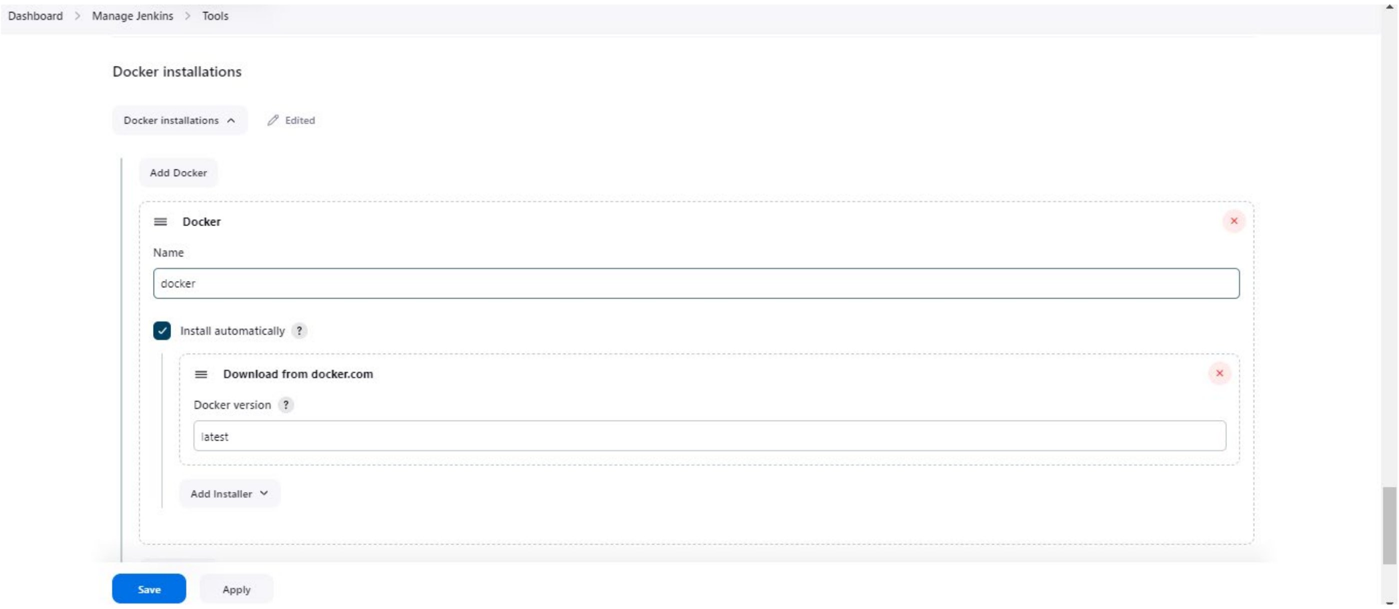
Name: DP-Check

☑ Install automatically

Install from github.com

Version: dependency-check 10.0.3

Use the latest version of Docker



Docker installations

Docker

Name: docker

☑ Install automatically

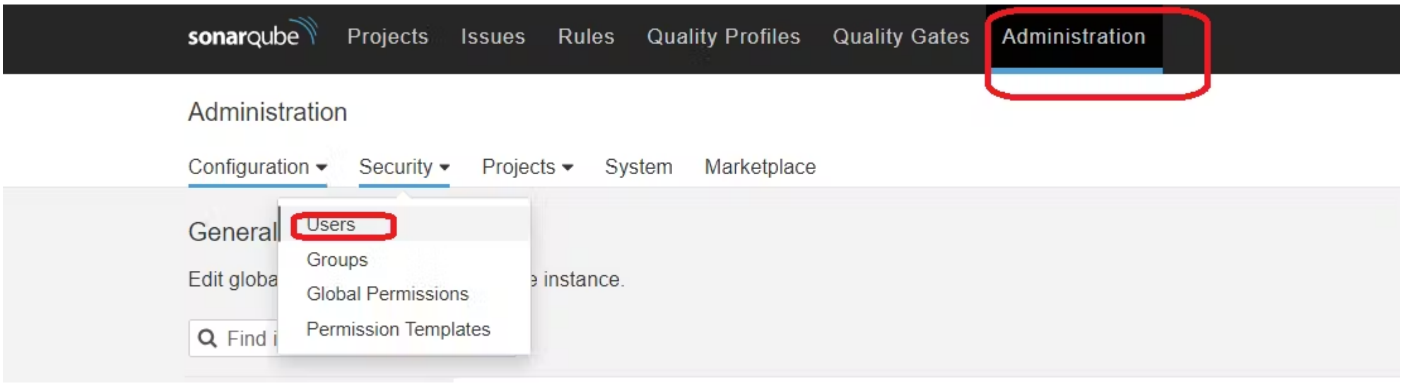
Download from docker.com

Docker version: latest

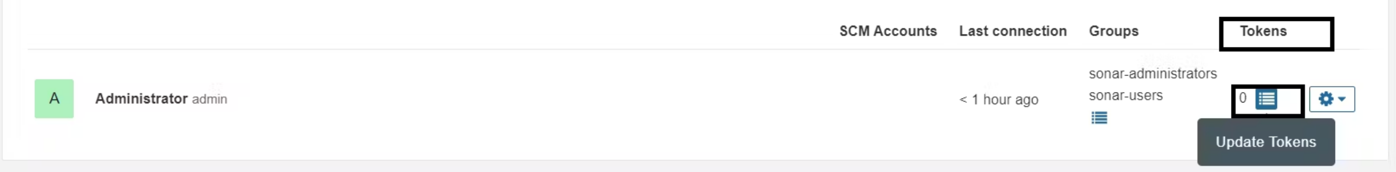
Click apply and save.

Configure Sonar Server in Manage Jenkins

Grab the Public IP Address of your EC2 Instance, Sonarqube works on Port 9000, so <Public IP>:9000. Goto your Sonarqube Server. Click on Administration → Security → Users → Click on Tokens and Update Token → Give it a name → and click on Generate Token

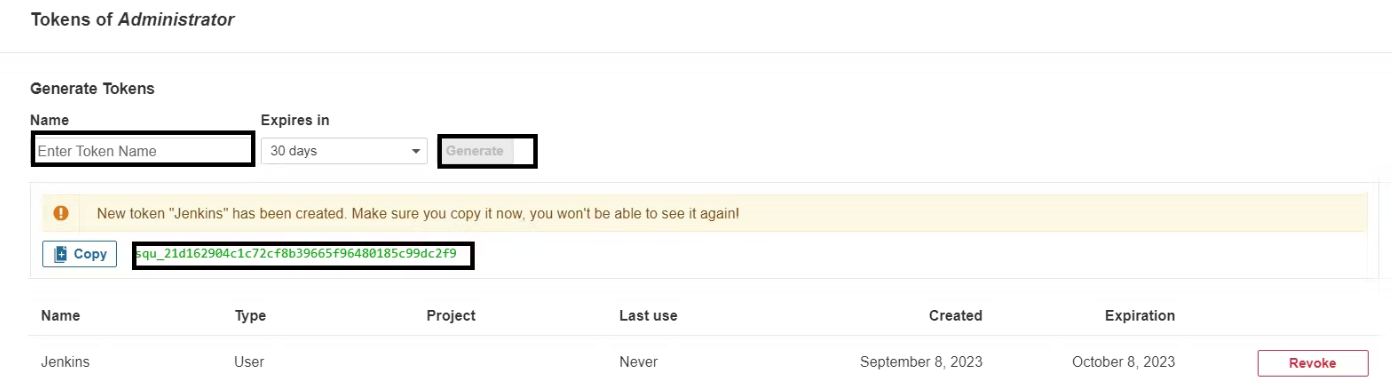


click on update Token



		SCM Accounts	Last connection	Groups	Tokens
A	Administrator admin		< 1 hour ago	sonar-administrators sonar-users	0

Create a token with a name and generate



Tokens of *Administrator*

Generate Tokens

Name: Enter Token Name

Expires in: 30 days

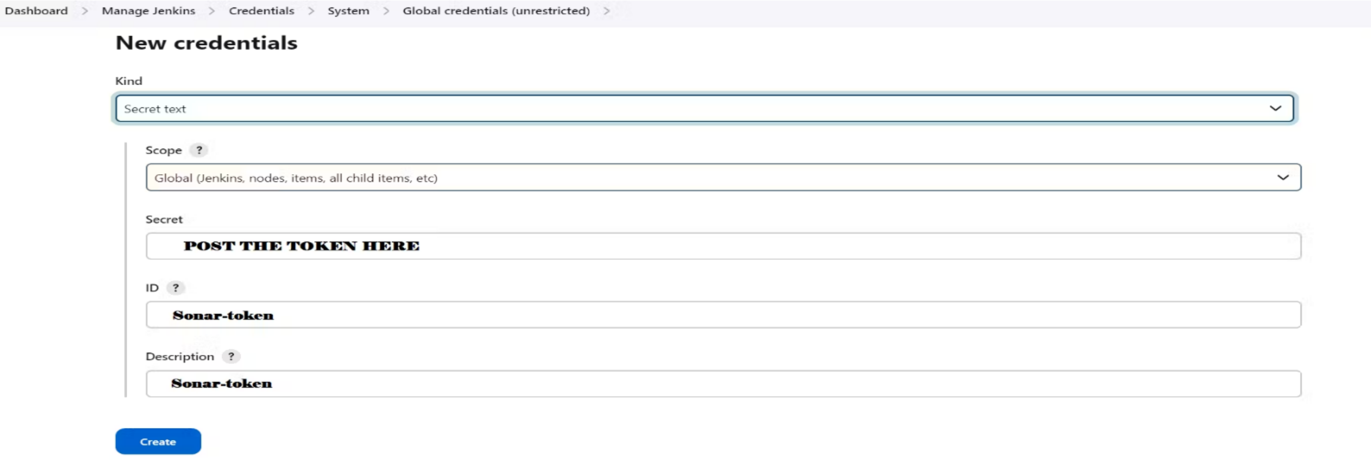
New token "Jenkins" has been created. Make sure you copy it now, you won't be able to see it again!

```
squ_21d162904c1c72cf8b39665f96480185c99dc2f9
```

Name	Type	Project	Last use	Created	Expiration	
Jenkins	User		Never	September 8, 2023	October 8, 2023	Revoke

copy Token

Go to Jenkins Dashboard → Manage Jenkins → Credentials → Add Secret Text. It should look like this



Dashboard > Manage Jenkins > Credentials > System > Global credentials (unrestricted) >

New credentials

Kind: Secret text

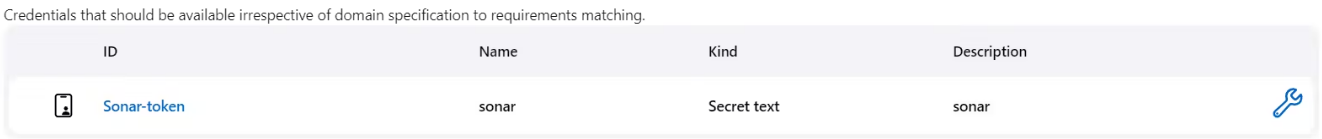
Scope: Global (Jenkins, nodes, items, all child items, etc)

Secret: POST THE TOKEN HERE

ID: Sonar-token

Description: Sonar-token

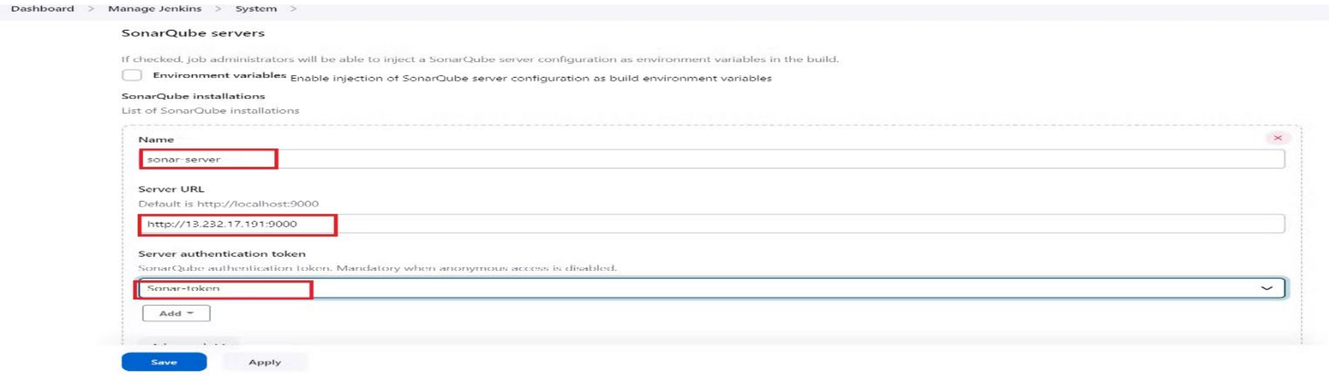
You will this page once you click on create



Credentials that should be available irrespective of domain specification to requirements matching.

	ID	Name	Kind	Description	
	Sonar-token	sonar	Secret text	sonar	

Now, go to Dashboard → Manage Jenkins → System and Add like the below image.



Dashboard > Manage Jenkins > System >

SonarQube servers

If checked, job administrators will be able to inject a SonarQube server configuration as environment variables in the build.

□ Environment variables Enable injection of SonarQube server configuration as build environment variables

SonarQube installations

List of SonarQube installations

Name: sonar-server

Server URL

Default is http://localhost:9000

http://13.232.17.191:9000

Server authentication token

SonarQube authentication token. Mandatory when anonymous access is disabled.

Sonar-token

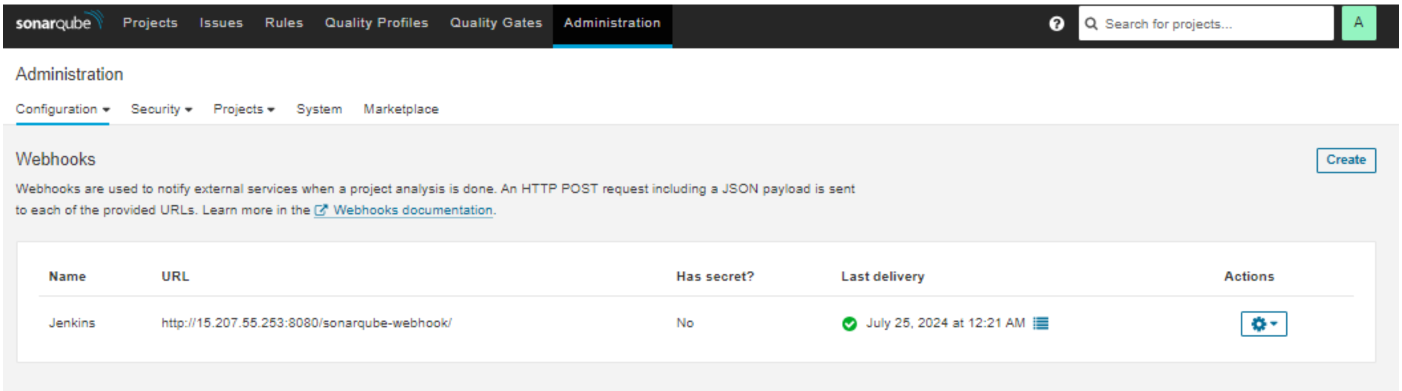
Click on Apply and Save

In the Sonarqube Dashboard add a quality gate also

Administration–> Configuration–>Webhooks

Click on Create

Add details



Webhooks

Webhooks are used to notify external services when a project analysis is done. An HTTP POST request including a JSON payload is sent to each of the provided URLs. Learn more in the Webhooks documentation.

Name	URL	Has secret?	Last delivery	Actions
Jenkins	http://15.207.55.253:8080/sonarqube-webhook/	No	✓ July 25, 2024 at 12:21 AM	⚙

Now add Docker credentials to the Jenkins to log in and push the image

Manage Jenkins –> Credentials –> global –> add credential

Add DockerHub Username and Password under Global Credentials

Now install Docker Scout on instance CLI;

```
docker login    #use credentials to login
```

```
curl -sSfL https://raw.githubusercontent.com/docker/scout-cli/main/install.sh | sh -s -- -b /usr/local/bin
```

Pipeline upto Docker

```
pipeline{
    agent any
    tools{
        jdk 'jdk17'
        nodejs 'node22'
    }
    environment {
        SCANNER_HOME=tool 'sonar-scanner'
    }
    stages {
        stage('clean workspace'){
            steps{
                cleanWs()
            }
    }
    stage('Checkout from Git'){
      steps{
        git branch: 'main', url: ' https://github.com/Bijan1235/Hotstar-Clone.git'
      }
    }
    stage("Sonarqube Analysis "){
      steps{
        withSonarQubeEnv('sonar-server') {
          sh ''' $SCANNER_HOME/bin/sonar-scanner -Dsonar.projectName=Hotstar \
          -Dsonar.projectKey=Hotstar'''
        }
      }
    }
    stage("quality gate"){
     steps {
        script {
          waitForQualityGate abortPipeline: false, credentialsId: 'Sonar-token'
        }
      }
    }
    stage('Install Dependencies') {
      steps {
        sh "npm install"
      }
    }
    stage('OWASP FS SCAN') {
      steps {
        dependencyCheck additionalArguments: '--scan ./ --disableYarnAudit --disableNodeAudit', odcInstallation: 'DP-Check'
        dependencyCheckPublisher pattern: '**/dependency-check-report.xml'
      }
    }
```

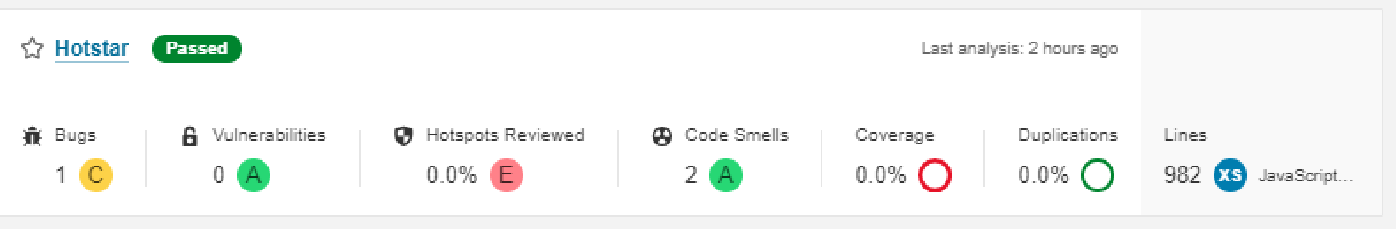
```
stage('Docker Scout FS') {
    steps {
        script{
           withDockerRegistry(credentialsId: 'docker', toolName: 'docker'){
               sh 'docker-scout quickview fs://.'
               sh 'docker-scout cves fs://.'
           }
        }
    }
}
stage("Docker Build & Push"){
    steps{
        script{
           withDockerRegistry(credentialsId: 'docker', toolName: 'docker'){
               sh "docker build -t hotstar ."
               sh "docker tag hotstar bijan9438/hotstar:latest "
               sh "docker push bijan9438/hotstar:latest"
            }
        }
    }
}
stage('Docker Scout Image') {
    steps {
        script{
           withDockerRegistry(credentialsId: 'docker', toolName: 'docker'){
               sh 'docker-scout quickview bijan9438/hotstar:latest'
               sh 'docker-scout cves bijan9438/hotstar:latest'
               sh 'docker-scout recommendations bijan9438/hotstar:latest'
           }
        }
    }
}
```

```
    stage("deploy_docker"){
      steps{
        sh "docker run -d --name hotstar -p 3000:3000 bijan9438/hotstar:latest"
      }
    }
  }
}
```

Click on Apply and save.

Build now

To see the report, you can go to Sonarqube Server and go to Projects.

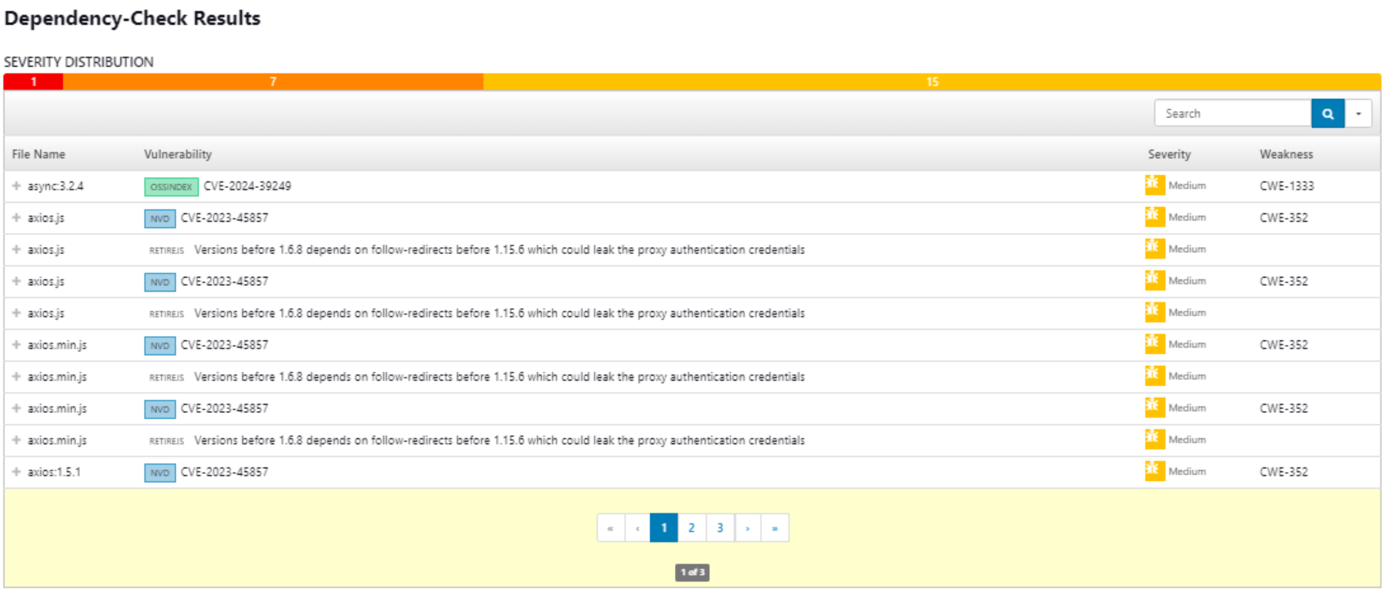


☆ Hotstar Passed Last analysis: 2 hours ago

Bugs	Vulnerabilities	Hotspots Reviewed	Code Smells	Coverage	Duplications	Lines
1 C	0 A	0.0% E	2 A	0.0%	0.0%	982 XS JavaScript...

You can see the report has been generated and the status shows as passed.

OWASP, You will see that in status, a graph will also be generated and Vulnerabilities.

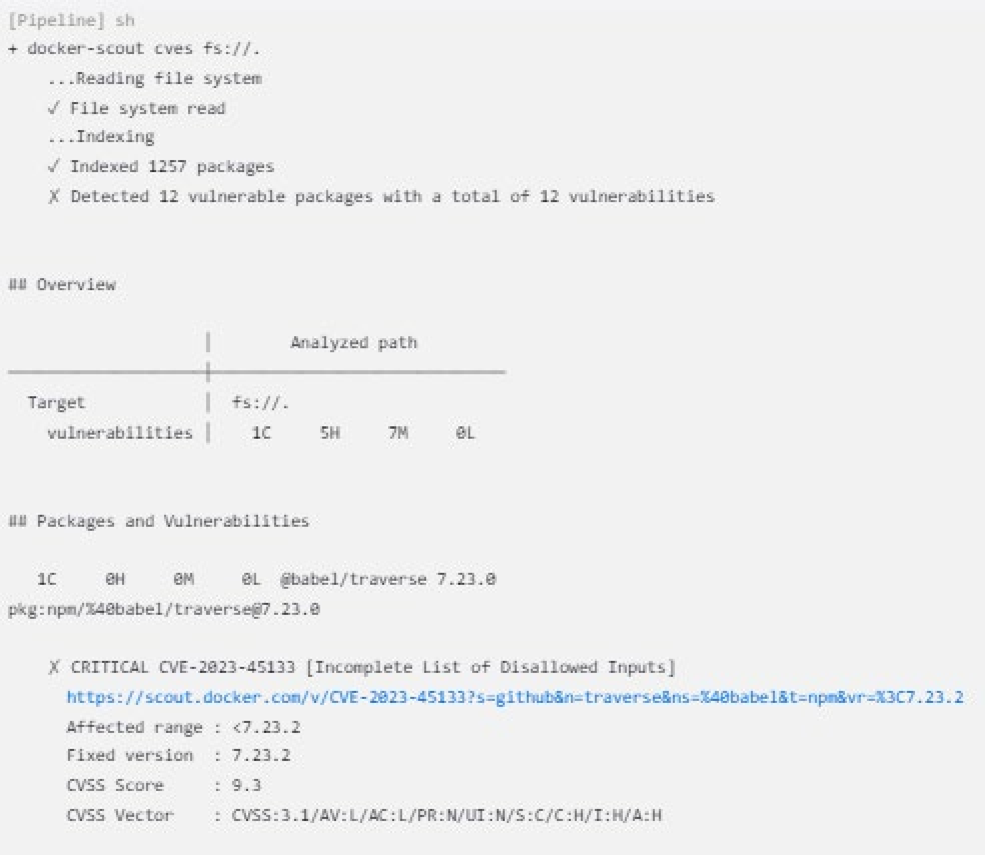


Dependency-Check Results

SEVERITY DISTRIBUTION

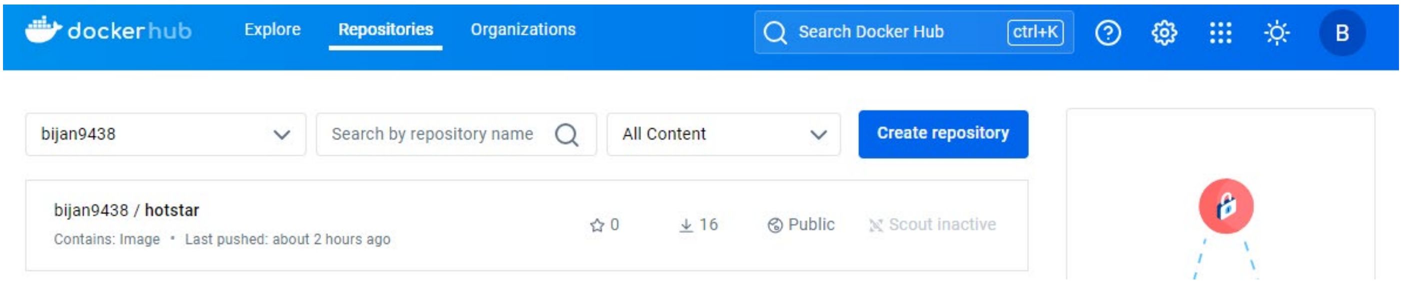
File Name	Vulnerability	Severity	Weakness
+ async:3.2.4	OSSINDEX CVE-2024-39249	Medium	CWE-1333
+ axios.js	NVD CVE-2023-45857	Medium	CWE-352
+ axios.js	RETIREJS Versions before 1.6.8 depends on follow-redirects before 1.15.6 which could leak the proxy authentication credentials	Medium	
+ axios.js	NVD CVE-2023-45857	Medium	CWE-352
+ axios.js	RETIREJS Versions before 1.6.8 depends on follow-redirects before 1.15.6 which could leak the proxy authentication credentials	Medium	
+ axios.min.js	NVD CVE-2023-45857	Medium	CWE-352
+ axios.min.js	RETIREJS Versions before 1.6.8 depends on follow-redirects before 1.15.6 which could leak the proxy authentication credentials	Medium	
+ axios.min.js	NVD CVE-2023-45857	Medium	CWE-352
+ axios.min.js	RETIREJS Versions before 1.6.8 depends on follow-redirects before 1.15.6 which could leak the proxy authentication credentials	Medium	
+ axios:1.5.1	NVD CVE-2023-45857	Medium	CWE-352

Let’s See Docker Scout File scan report



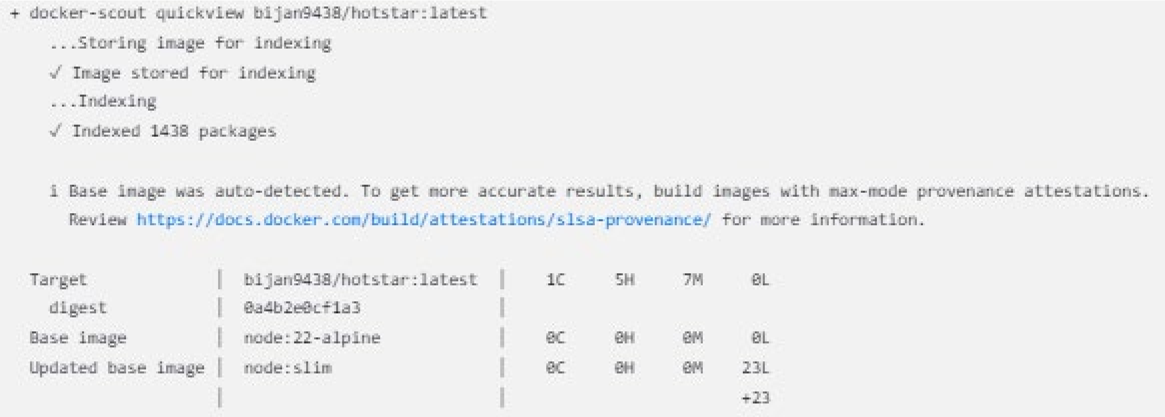
```
[Pipeline] sh
+ docker-scout cves fs://.
    ...Reading file system
    ✓ File system read
    ...Indexing
    ✓ Indexed 1257 packages
    ✗ Detected 12 vulnerable packages with a total of 12 vulnerabilities


## Overview

                    │       Analyzed path
────────────────────┼──────────────────────────────
  Target            │  fs://.
    vulnerabilities │    1C     5H     7M     0L


## Packages and Vulnerabilities

   1C     0H     0M     0L  @babel/traverse 7.23.0
pkg:npm/%40babel/traverse@7.23.0

    ✗ CRITICAL CVE-2023-45133 [Incomplete List of Disallowed Inputs]
      https://scout.docker.com/v/CVE-2023-45133?s=github&n=traverse&ns=%40babel&t=npm&vr=%3C7.23.2
      Affected range : <7.23.2
      Fixed version  : 7.23.2
      CVSS Score     : 9.3
      CVSS Vector    : CVSS:3.1/AV:L/AC:L/PR:N/UI:N/S:C/C:H/I:H/A:H
```

When you log in to Dockerhub, you will see a new image is created



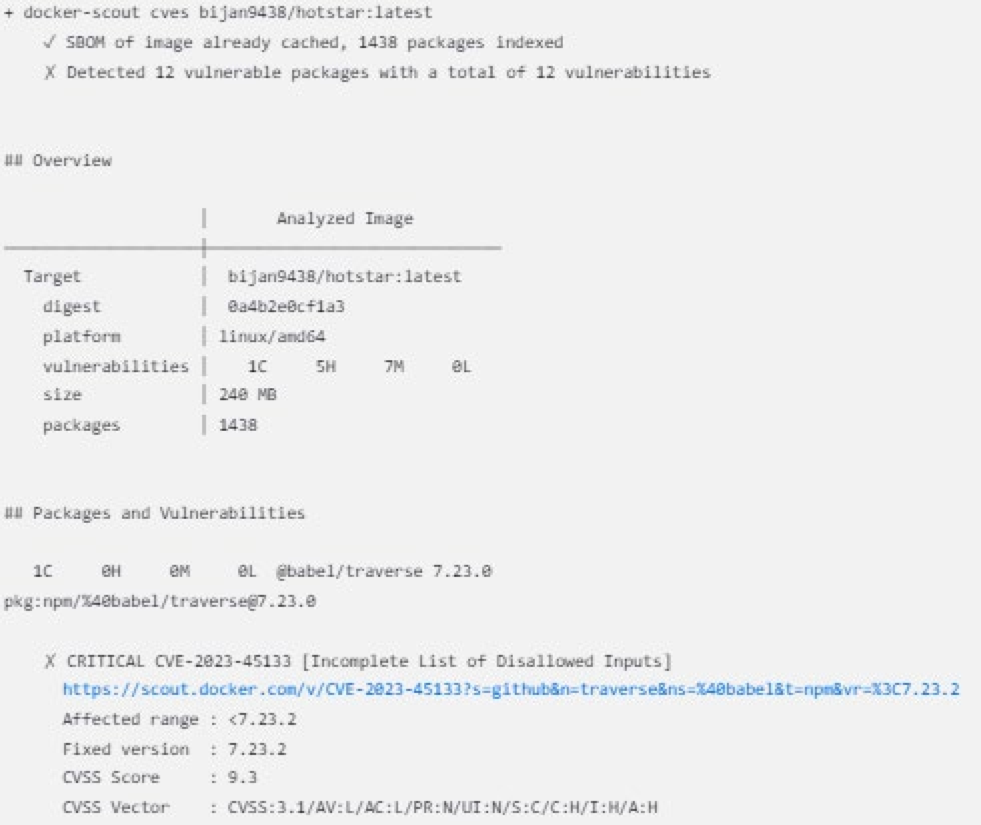
bijan9438 / hotstar
Contains: Image • Last pushed: about 2 hours ago

Let’s See Docker Scout Image analysis



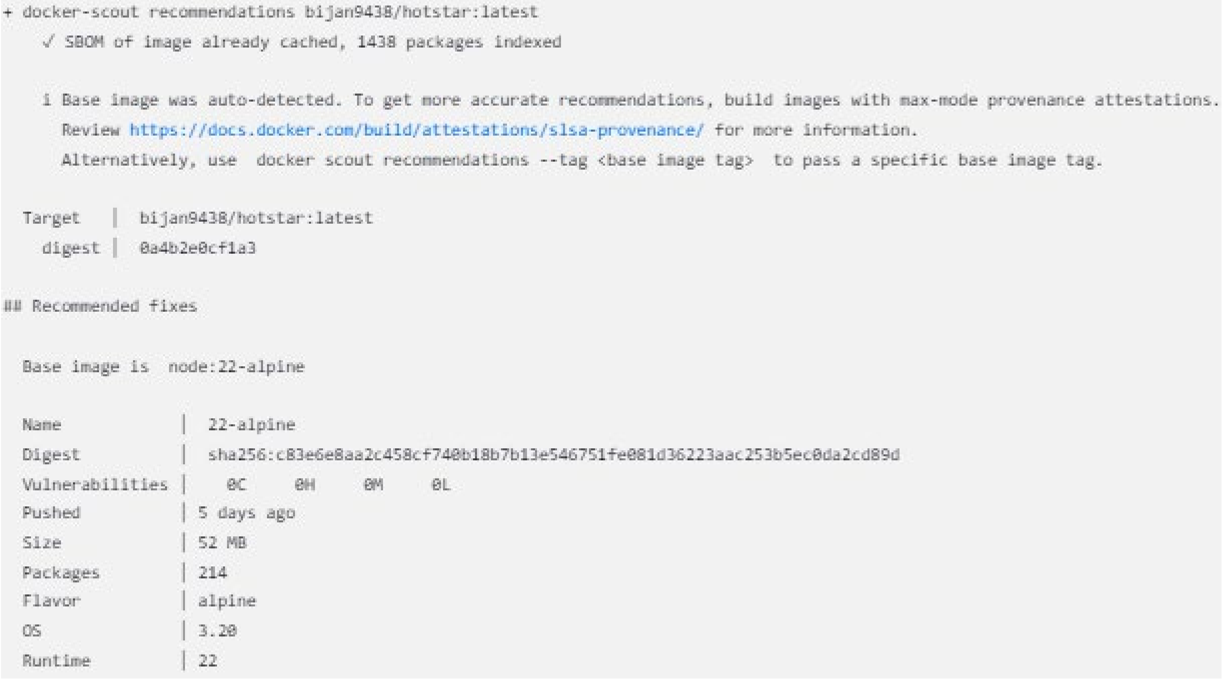
```
+ docker-scout quickview bijan9438/hotstar:latest
    ...Storing image for indexing
    ✓ Image stored for indexing
    ...Indexing
    ✓ Indexed 1438 packages

    i Base image was auto-detected. To get more accurate results, build images with max-mode provenance attestations.
      Review https://docs.docker.com/build/attestations/slsa-provenance/ for more information.

  Target             │  bijan9438/hotstar:latest  │    1C     5H     7M     0L
    digest           │  0a4b2e0cf1a3              │
  Base image         │  node:22-alpine            │    0C     0H     0M     0L
  Updated base image │  node:slim                 │    0C     0H     0M    23L
                     │                            │                      +23
```

Cves



```
+ docker-scout cves bijan9438/hotstar:latest
    ✓ SBOM of image already cached, 1438 packages indexed
    ✗ Detected 12 vulnerable packages with a total of 12 vulnerabilities


## Overview

                    │       Analyzed Image
────────────────────┼──────────────────────────────
  Target            │  bijan9438/hotstar:latest
    digest          │  0a4b2e0cf1a3
    platform        │ linux/amd64
    vulnerabilities │    1C     5H     7M     0L
    size            │ 240 MB
    packages        │ 1438


## Packages and Vulnerabilities

   1C     0H     0M     0L  @babel/traverse 7.23.0
pkg:npm/%40babel/traverse@7.23.0

    ✗ CRITICAL CVE-2023-45133 [Incomplete List of Disallowed Inputs]
      https://scout.docker.com/v/CVE-2023-45133?s=github&n=traverse&ns=%40babel&t=npm&vr=%3C7.23.2
      Affected range : <7.23.2
      Fixed version  : 7.23.2
      CVSS Score     : 9.3
      CVSS Vector    : CVSS:3.1/AV:L/AC:L/PR:N/UI:N/S:C/C:H/I:H/A:H
```

Recommendations

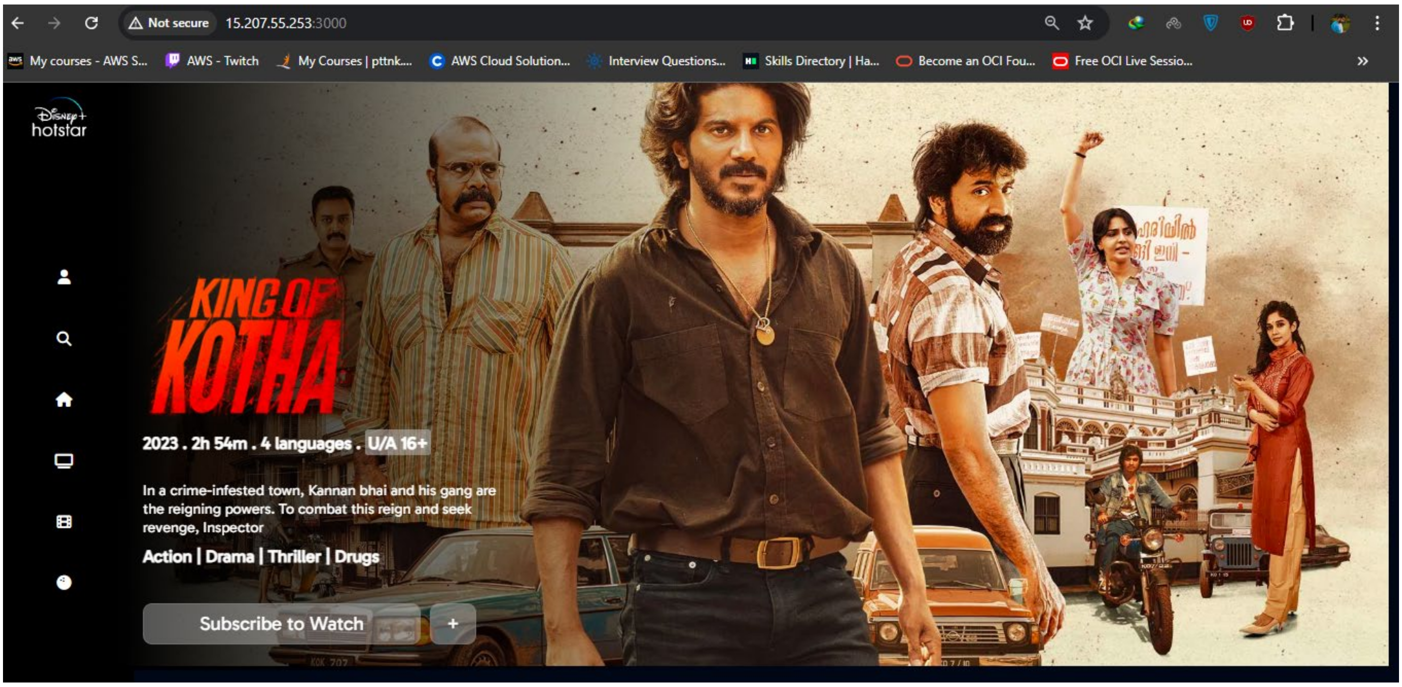


```
+ docker-scout recommendations bijan9438/hotstar:latest
    ✓ SBOM of image already cached, 1438 packages indexed

    i Base image was auto-detected. To get more accurate recommendations, build images with max-mode provenance attestations.
      Review https://docs.docker.com/build/attestations/slsa-provenance/ for more information.
      Alternatively, use  docker scout recommendations --tag <base image tag>  to pass a specific base image tag.

  Target   │  bijan9438/hotstar:latest
    digest │  0a4b2e0cf1a3

## Recommended fixes

  Base image is  node:22-alpine

  Name            │  22-alpine
  Digest          │  sha256:c83e6e8aa2c458cf740b18b7b13e546751fe081d36223aac253b5ec0da2cd89d
  Vulnerabilities │    0C     0H     0M     0L
  Pushed          │ 5 days ago
  Size            │ 52 MB
  Packages        │ 214
  Flavor          │ alpine
  OS              │ 3.20
  Runtime         │ 22
```

Deploy to Container

<ec2publicip:3000>

Output



2023 . 2h 54m . 4 languages . U/A 16+

In a crime-infested town, Kannan bhai and his gang are the reigning powers. To combat this reign and seek revenge, Inspector

Action | Drama | Thriller | Drugs

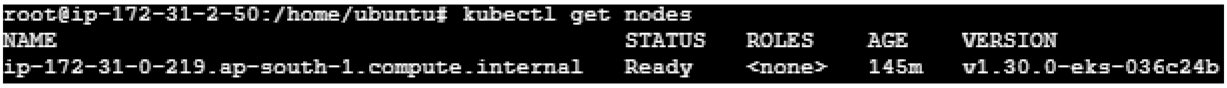
Subscribe to Watch +

Go to instance CLI and write;

```
aws eks update-kubeconfig --name EKS_CLOUD --region ap-south-1
```

Let’s see the nodes

```
kubectl get nodes
```



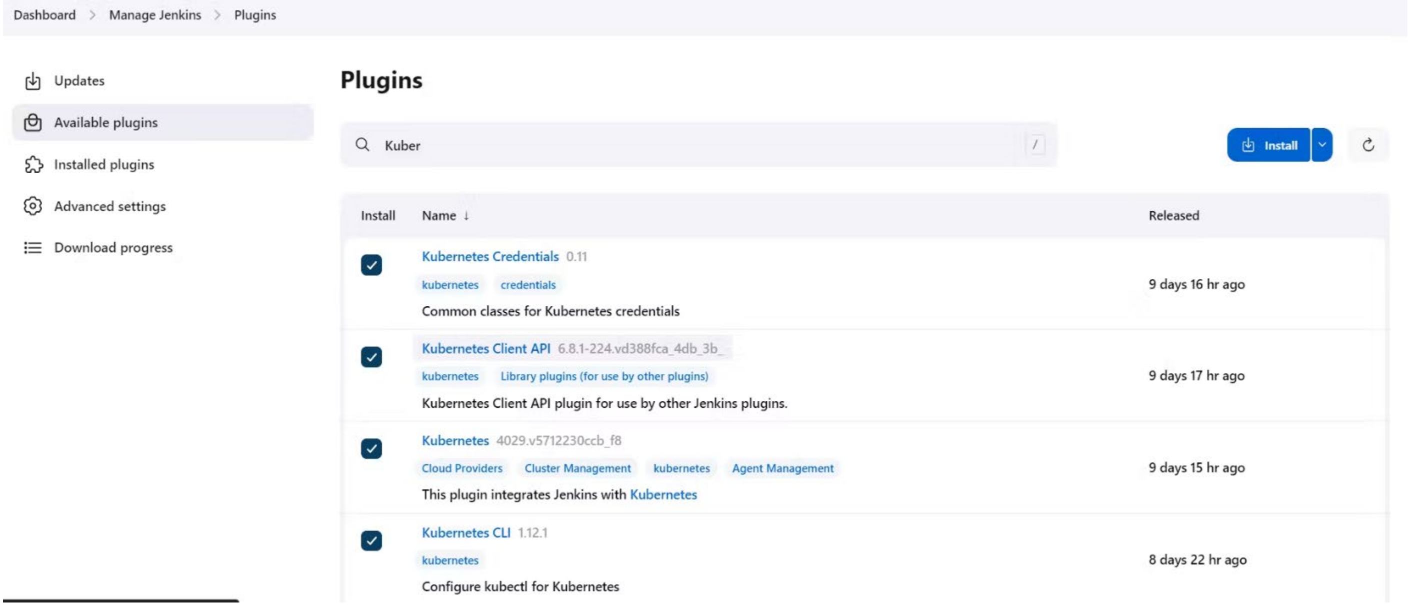
```
root@ip-172-31-2-50:/home/ubuntu# kubectl get nodes
NAME                                        STATUS   ROLES    AGE    VERSION
ip-172-31-0-219.ap-south-1.compute.internal   Ready    <none>   145m   v1.30.0-eks-036c24b
```

Now Give this command in CLI

```
cat /root/.kube/config
```

Copy the config file to Jenkins master or the local file manager and save it

Install Kubernetes Plugin, Once it’s installed successfully

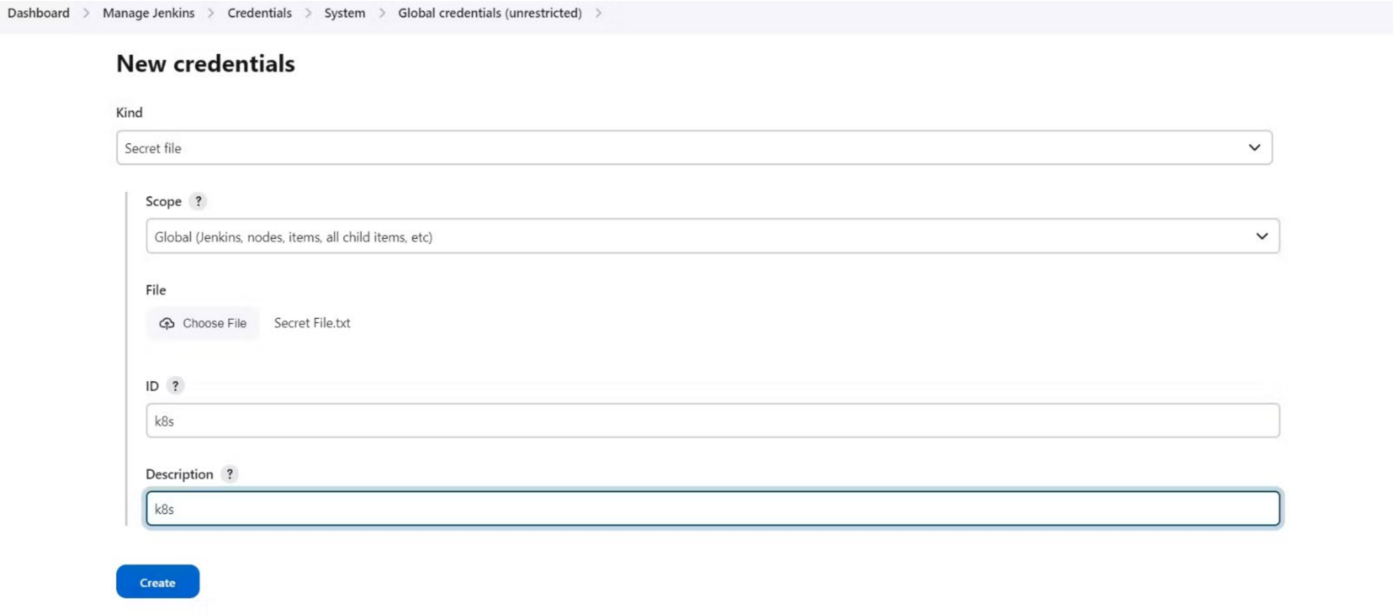


Plugins

Kuber

Install	Name ↓	Released
☑	Kubernetes Credentials 0.11 kubernetes credentials Common classes for Kubernetes credentials	9 days 16 hr ago
☑	Kubernetes Client API 6.8.1-224.vd388fca_4db_3b_ kubernetes Library plugins (for use by other plugins) Kubernetes Client API plugin for use by other Jenkins plugins.	9 days 17 hr ago
☑	Kubernetes 4029.v5712230ccb_f8 Cloud Providers Cluster Management kubernetes Agent Management This plugin integrates Jenkins with Kubernetes	9 days 15 hr ago
☑	Kubernetes CLI 1.12.1 kubernetes Configure kubectl for Kubernetes	8 days 22 hr ago

goto manage Jenkins –> manage credentials –> Click on Jenkins global –> add credentials



Dashboard > Manage Jenkins > Credentials > System > Global credentials (unrestricted) >

New credentials

Kind: Secret file

Scope: Global (Jenkins, nodes, items, all child items, etc)

File: Choose File Secret File.txt

ID: k8s

Description: k8s

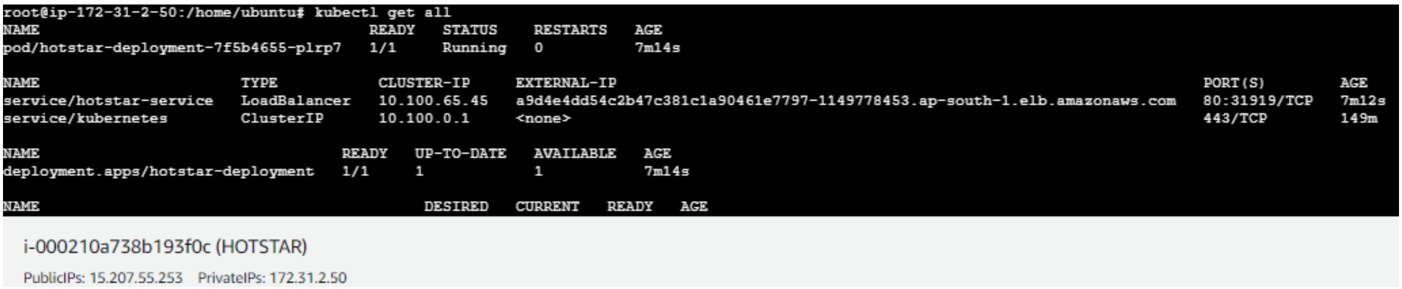
Create

final step to deploy on the Kubernetes cluster

```
stage('Deploy to kubernets'){
      steps{
        script{
          dir('K8S') {
            withKubeConfig(caCertificate: '', clusterName: '', contextName: '', credentialsId: 'k8s', namespace: '', restrictKubeConfigAccess: false, serverUrl: '') {
                sh 'kubectl apply -f deployment.yml'
                sh 'kubectl apply -f service.yml'
            }
          }
        }
      }
    }
```

Give the command after pipeline success

kubectl get all

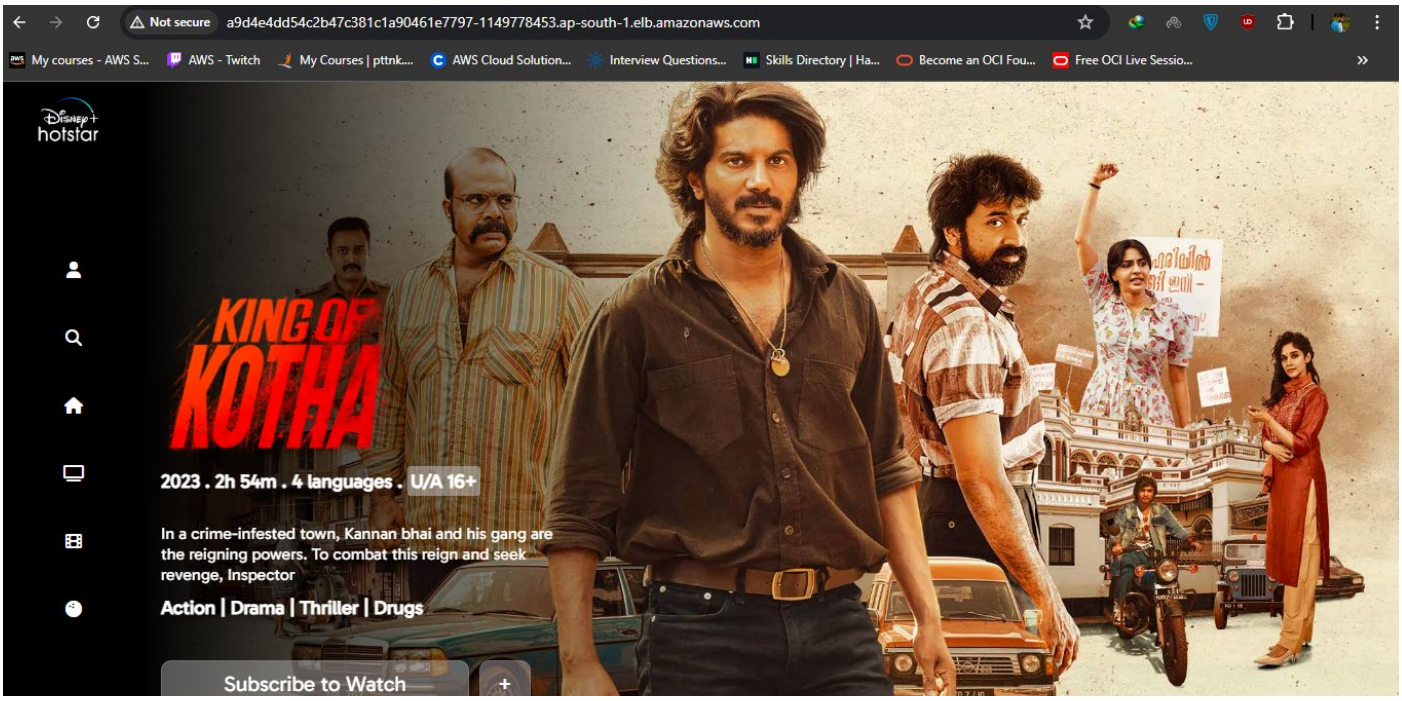


```
root@ip-172-31-2-50:/home/ubuntu# kubectl get all
NAME                                  READY   STATUS    RESTARTS   AGE
pod/hotstar-deployment-7f5b4655-plrp7   1/1     Running   0          7m14s

NAME                        TYPE           CLUSTER-IP     EXTERNAL-IP                                                                  PORT(S)        AGE
service/hotstar-service     LoadBalancer   10.100.65.45   a9d4e4dd54c2b47c381c1a90461e7797-1149778453.ap-south-1.elb.amazonaws.com   80:31919/TCP   7m12s
service/kubernetes          ClusterIP      10.100.0.1     <none>                                                                       443/TCP        149m

NAME                                 READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/hotstar-deployment   1/1     1            1           7m14s

NAME                                            DESIRED   CURRENT   READY   AGE
```

i-000210a738b193f0c (HOTSTAR)

PublicIPs: 15.207.55.253   PrivateIPs: 172.31.2.50

Copy the External IP and paste it in your browser, You will see output like this.



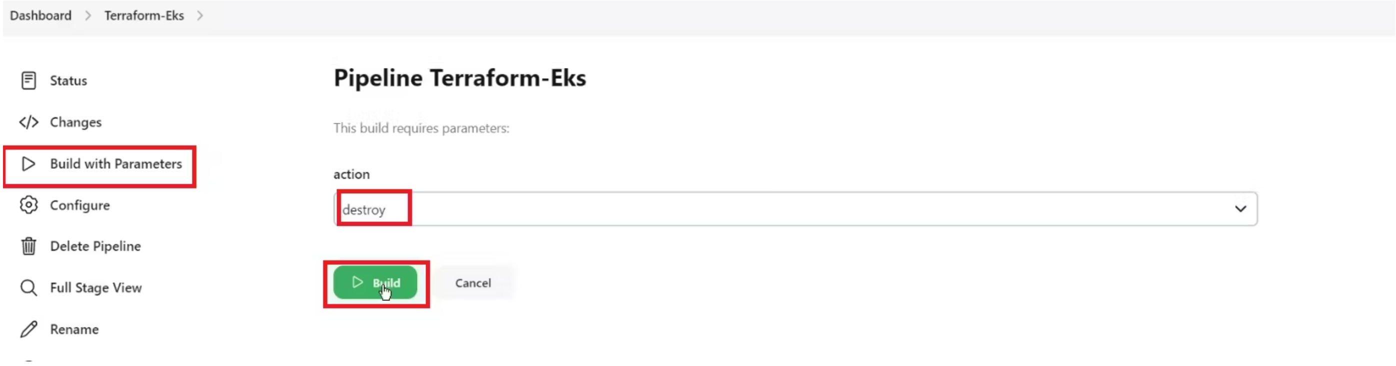
In a crime-infested town, Kannan bhai and his gang are the reigning powers. To combat this reign and seek revenge, Inspector

Step 4: Destruction

Now Go to Jenkins Dashboard and click on Terraform-Eks job

And build with parameters and destroy action

It will delete the EKS cluster that provisioned



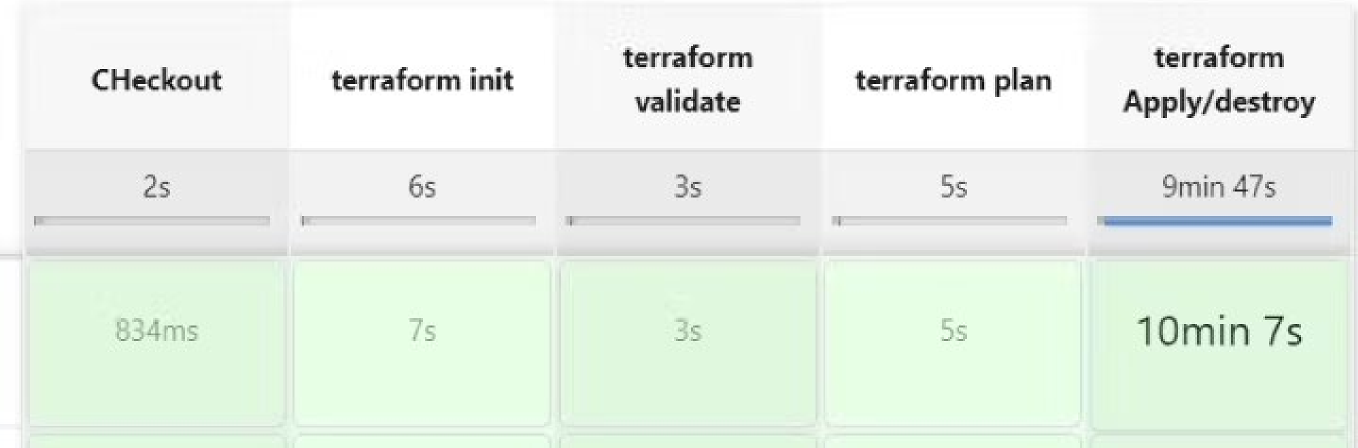
Pipeline Terraform-Eks

This build requires parameters:

action

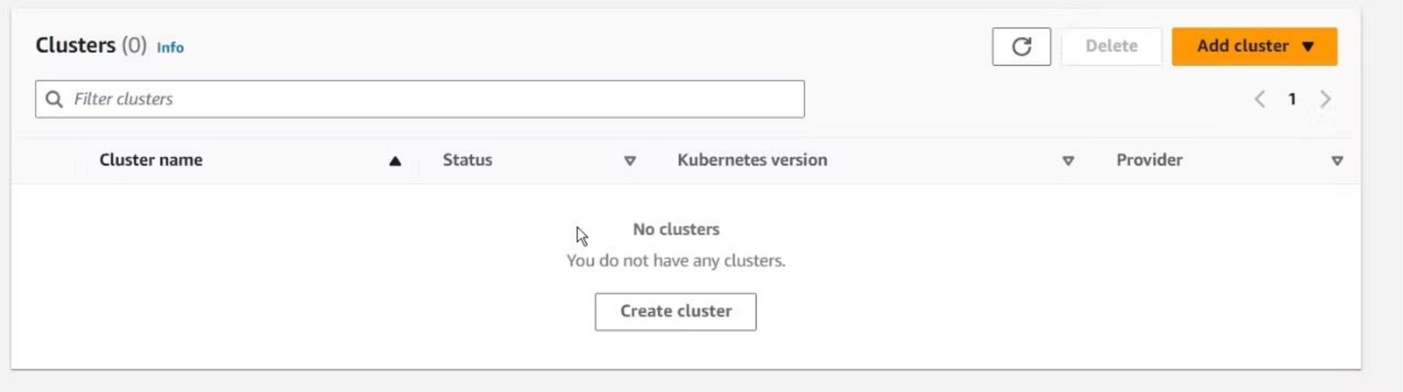
destroy

After 10 minutes cluster will delete and wait for it. Don't remove ec2 instance till that time.



CHeckout	terraform init	terraform validate	terraform plan	terraform Apply/destroy
2s	6s	3s	5s	9min 47s
834ms	7s	3s	5s	10min 7s

Cluster deleted



Cluster name ▲	Status ▽	Kubernetes version ▽	Provider ▽

No clusters

You do not have any clusters.

Delete the Ec2 instance & IAM role.

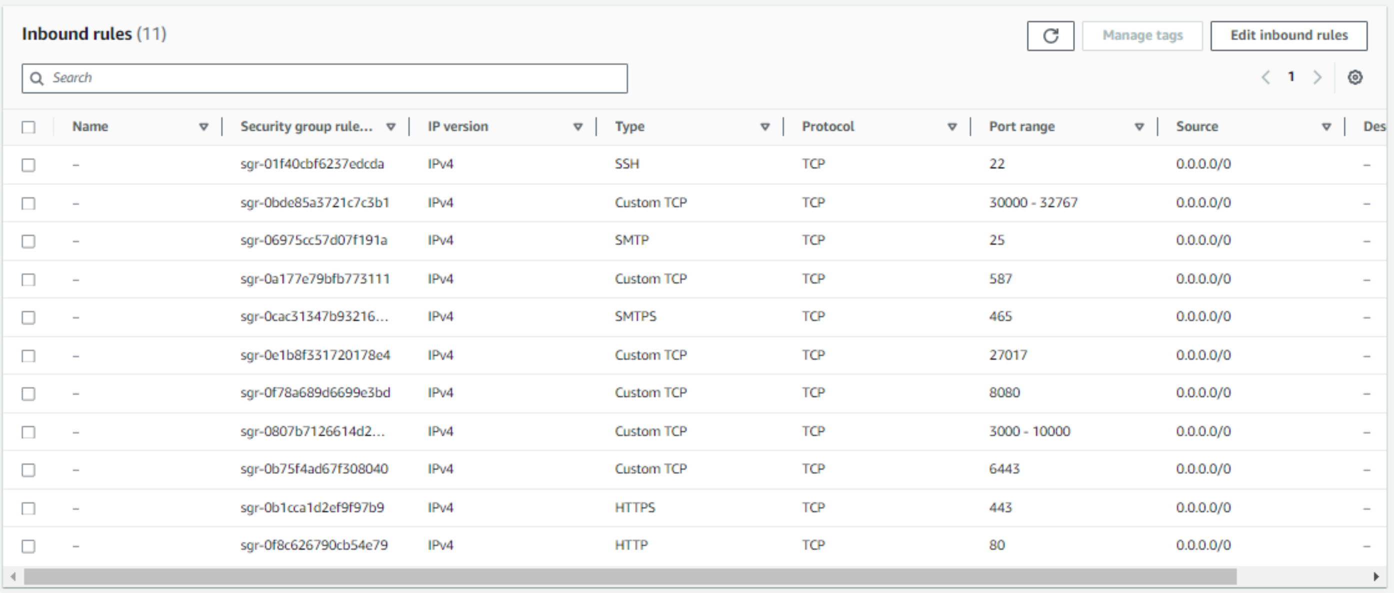
Check the load balancer also if it is deleted or not.

Finally completing the journey of deploying Hotstar clone using DevSecOps practices on AWS!
This process has highlighted the power of integrating security measures seamlessly into the deployment pipeline, ensuring not only efficiency but also a robust shield against potential threats.

Key Highlights:

- Leveraging AWS services, Docker, Jenkins, and security tools, we orchestrated a secure and automated deployment pipeline.
- Implementing DevSecOps principles helped fortify the application against vulnerabilities through continuous security checks.
- The seamless integration of static code analysis, container security, and automated deployment showcases the strength of DevSecOps methodologies.

PORTS(UNLOCKED FOR THIS PROJECT):



□	Name	Security group rule...	IP version	Type	Protocol	Port range	Source	Des
□	–	sgr-01f40cbf6237edcda	IPv4	SSH	TCP	22	0.0.0.0/0	–
□	–	sgr-0bde85a3721c7c3b1	IPv4	Custom TCP	TCP	30000 - 32767	0.0.0.0/0	–
□	–	sgr-06975cc57d07f191a	IPv4	SMTP	TCP	25	0.0.0.0/0	–
□	–	sgr-0a177e79bfb773111	IPv4	Custom TCP	TCP	587	0.0.0.0/0	–
□	–	sgr-0cac31347b93216...	IPv4	SMTPS	TCP	465	0.0.0.0/0	–
□	–	sgr-0e1b8f331720178e4	IPv4	Custom TCP	TCP	27017	0.0.0.0/0	–
□	–	sgr-0f78a689d6699e3bd	IPv4	Custom TCP	TCP	8080	0.0.0.0/0	–
□	–	sgr-0807b7126614d2...	IPv4	Custom TCP	TCP	3000 - 10000	0.0.0.0/0	–
□	–	sgr-0b75f4ad67f308040	IPv4	Custom TCP	TCP	6443	0.0.0.0/0	–
□	–	sgr-0b1cca1d2ef9f97b9	IPv4	HTTPS	TCP	443	0.0.0.0/0	–
□	–	sgr-0f8c626790cb54e79	IPv4	HTTP	TCP	80	0.0.0.0/0	–