

Mr Cloud Book

Home Blog DevSecOps Contact About Me Testimonials

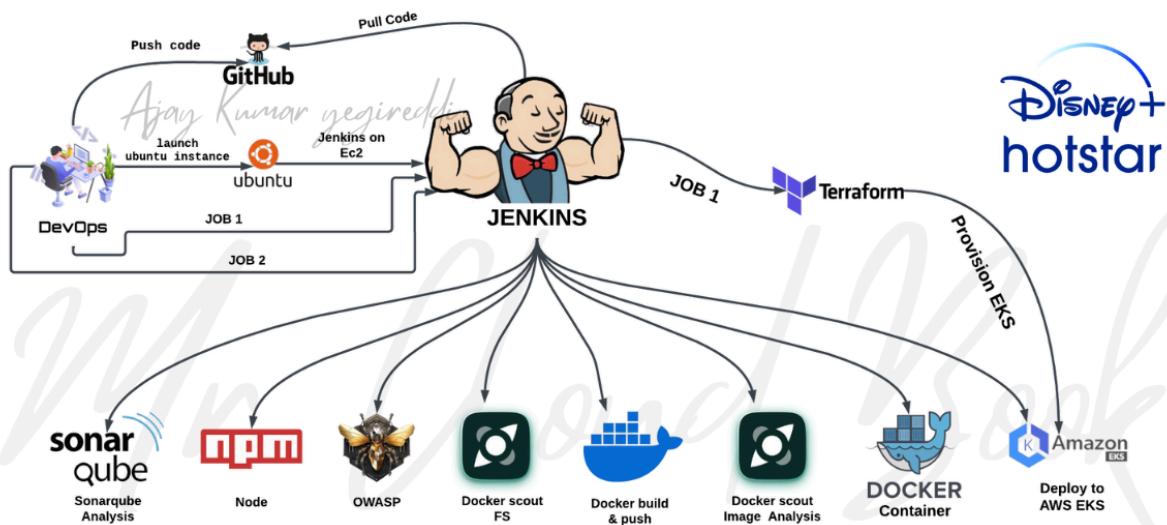
Search Blogs

Uncategorized

DevSecOps CI/CD : Deploying a Secure Hotstar Clone (Even if You're Not a Pro)



mrcloobook.com · 6 January 2024



DEVSECOPS CI/CD

In the ever-evolving landscape of software development and deployment, the integration of robust security practices into the development pipeline has become imperative. DevSecOps, an amalgamation of Development, Security, and Operations, emphasizes the integration of security measures throughout the software development lifecycle, promoting a proactive approach to mitigate potential vulnerabilities and threats.

This blog serves as a comprehensive guide to deploying a Hotstar clone—a popular streaming platform—using the principles of DevSecOps on Amazon Web Services (AWS). The deployment process encompasses the utilization of various tools and services, including Docker, Jenkins, Java, SonarQube, AWS CLI, Kubectl, and Terraform, to automate, secure, and streamline the deployment pipeline.

The journey begins with the setup of an AWS EC2 instance configured with Ubuntu, granting it a specific IAM role to facilitate the learning process. Subsequently, an automated script is crafted to install crucial tools and dependencies required for the deployment pipeline, ensuring efficiency and consistency in the setup process.

Central to this DevSecOps approach is the orchestration through Jenkins jobs, where stages are defined to execute tasks such as creating an Amazon EKS cluster, deploying the Hotstar clone application, and implementing security measures at various stages of the deployment.

The integration of security practices is a pivotal aspect of this process. Utilizing SonarQube, OWASP, and Docker Scout, the blog will elucidate how static code analysis, security checks, and container security scans are seamlessly embedded within the pipeline. These measures fortify the application against potential vulnerabilities, ensuring a robust and secure deployment.

GITHUB : <https://github.com/Aj7Ay/Hotstar-Clone.git>

Contents [hide]

[Prerequisites](#)

[Step-by-Step Deployment Process](#)

[STEP 1A: Setting up AWS EC2 Instance and IAM Role](#)

[STEP 1B: IAM ROLE](#)

[Step 2: Installation of Required Tools on the Instance](#)

[Step 3: Jenkins Job Configuration](#)

[Step 3A: EKS Provision job](#)

[Step 3B: Hotstar job](#)

[Configure in Global Tool Configuration](#)

[Configure Sonar Server in Manage Jenkins](#)

[Pipeline upto Docker](#)

[Step 4: Destruction](#)

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 22.04 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:

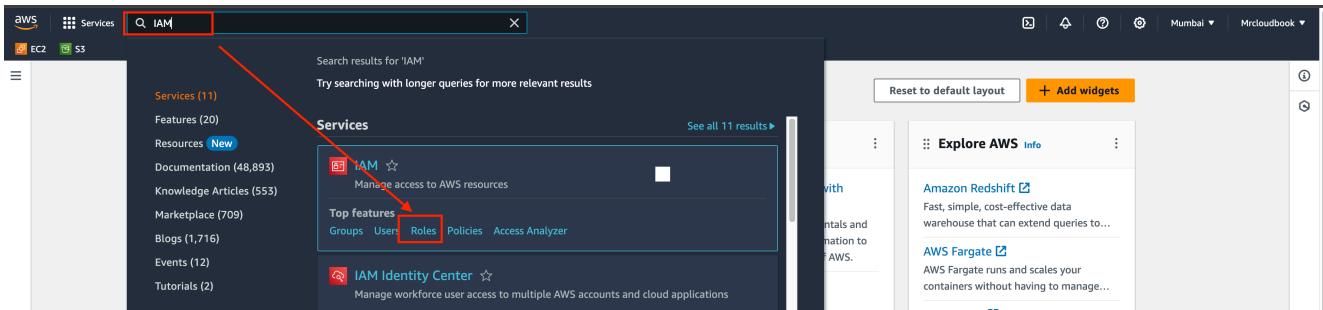
- Click on “Launch” to proceed.
- A pop-up will prompt you to select or create a key pair. Choose an existing key pair or create a new one.

- o Finally, click on “Launch Instances.”

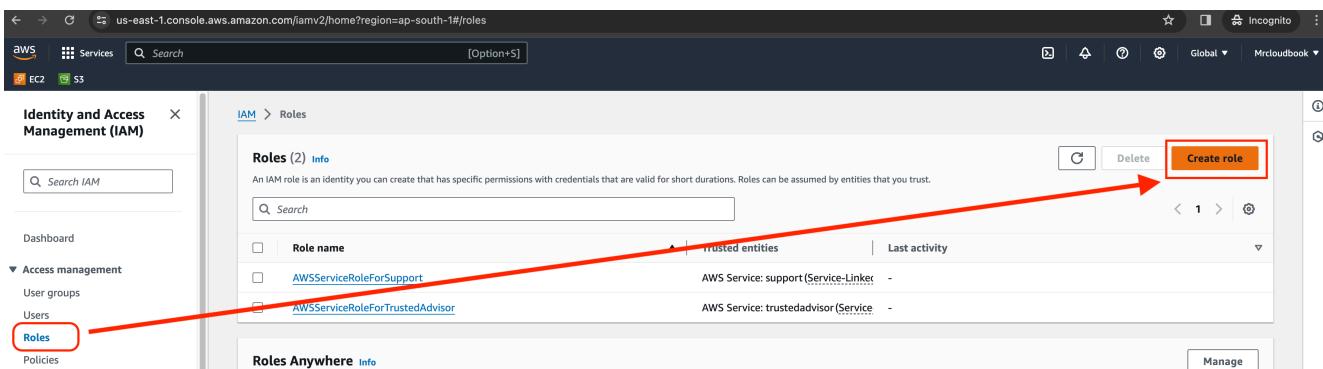
11. Accessing the Instance: Once the instance is launched, you can connect to it using SSH. Use the private key associated with the selected key pair to connect to the instance’s public IP or DNS address.

STEP 1B: IAM ROLE

Search for IAM in the search bar of AWS and click on roles.



Click on Create Role



Select entity type as AWS service

Use case as EC2 and click on Next.

Trusted entity type

- AWS service**
Allow AWS services like EC2, Lambda, or others to perform actions in this account.
- AWS account**
Allow entities in other AWS accounts belonging to you or a 3rd party to perform actions in this account.
- SAML 2.0 federation**
Allow users federated with SAML 2.0 from a corporate directory to perform actions in this account.
- Custom trust policy**
Create a custom trust policy to enable others to perform actions in this account.

Use case
Allow an AWS service like EC2, Lambda, or others to perform actions in this account.

Service or use case
EC2

Choose a use case for the specified service.
Use case

- EC2**
Allows EC2 instances to call AWS services on your behalf.
- EC2 Role for AWS Systems Manager**
Allows EC2 instances to call AWS services like CloudWatch and Systems Manager on your behalf.

For permission policy select Administrator Access (Just for learning purpose), click Next.

Add permissions

Permissions policies (1/897)

Choose one or more policies to attach to your new role.

Policy name	Type	Description
<input checked="" type="checkbox"/> AdministratorAccess	AWS managed - job function	Provides full access to AWS services an...
<input type="checkbox"/> AdministratorAccess-Amplify	AWS managed	Grants account administrative permis...

Provide a Name for Role and click on Create role.

Role details

Role name
Enter a meaningful name to identify this role.
MARIO

Description
Add a short explanation for this role.
Allows EC2 instances to call AWS services on your behalf.

Step 1: Select trusted entities

Trust policy

```

1  {
2    "Version": "2012-10-17",
3    "Statement": [
4      {
5        "Effect": "Allow",
6        "Action": [
7          "sts:AssumeRole"
8        ],
9        "Principal": {
10          "Service": [
11            "ec2.amazonaws.com"
12          ]
13        }
14      }
15    ]
16  }

```

Role is created.

The screenshot shows the AWS IAM Roles page. At the top, a green banner says "Role MARIO created." The main table lists four roles: "AWSServiceRoleForSupport", "AWSServiceRoleForTrustedAdvisor", and "MARIO" (which is highlighted with a red box). The "Create role" button is visible at the top right.

Now Attach this role to Ec2 instance that we created earlier, so we can provision cluster from that instance.

Go to EC2 Dashboard and select the instance.

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

The screenshot shows the AWS EC2 Instances page. A single instance named "Test" is listed. In the Actions dropdown menu, the "Modify IAM role" option is highlighted with a red box. A red arrow points from the "Modify IAM role" option to the "Actions" menu.

Select the Role that created earlier and click on Update IAM role.

The screenshot shows the "Modify IAM role" dialog box for the instance "i-0c92d797a3813a128 (Test)". The "MARIO" role is selected in the dropdown menu. The "Update IAM role" button is highlighted with a blue box.

Connect the instance to Mobaxtreme or Putty.

Step 2: Installation of Required Tools on the Instance

Scripts to install Required tools



```
sudo su      #Into root  
vi script1.sh
```



Script1 for Java,Jenkins,Docker



```
#!/bin/bash  
sudo apt update -y  
wget -O - https://packages.adoptium.net/artifactory/api/gpg/key/public  
echo "deb [signed-by=/etc/apt/keyrings/adoptium.asc] https://packages.ac  
sudo apt update -y  
sudo apt install temurin-17-jdk -y  
/usr/bin/java --version  
curl -fsSL https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key | su  
echo deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc] https://pkj  
sudo apt-get update -y  
sudo apt-get install jenkins -y  
sudo systemctl start jenkins  
#install docker  
# Add Docker's official GPG key:  
sudo apt-get update  
sudo apt-get install ca-certificates curl gnupg -y  
sudo install -m 0755 -d /etc/apt/keyrings  
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --de  
sudo chmod a+r /etc/apt/keyrings/docker.gpg  
# Add the repository to Apt sources:  
echo \  
"deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/d  
$(. /etc/os-release && echo "$VERSION_CODENAME") stable" | \  
sudo tee /etc/apt/sources.list.d/docker.list > /dev/null  
sudo apt-get update  
sudo apt-get install docker-ce docker-ce-cli containerd.io docker-build  
sudo usermod -aG docker ubuntu  
newgrp docker
```



Run script by providing permissions



```
sudo chmod 777 script1.sh  
sh script1.sh
```



```
root@ip-172-31-41-147:/home/ubuntu# ls  
k8s-mario  
root@ip-172-31-41-147:/home/ubuntu# cd k8s-mario/  
root@ip-172-31-41-147:/home/ubuntu/k8s-mario#  
root@ip-172-31-41-147:/home/ubuntu/k8s-mario# ls  
EKS-TF deployment.yaml script.sh service.yaml  
root@ip-172-31-41-147:/home/ubuntu/k8s-mario#  
root@ip-172-31-41-147:/home/ubuntu/k8s-mario# chmod +x script.sh  
root@ip-172-31-41-147:/home/ubuntu/k8s-mario#  
root@ip-172-31-41-147:/home/ubuntu/k8s-mario# ls  
EKS-TF deployment.yaml script.sh service.yaml  
root@ip-172-31-41-147:/home/ubuntu/k8s-mario#  
root@ip-172-31-41-147:/home/ubuntu/k8s-mario# ./script.sh
```

Script 2 for Terraform,kubectl,Aws cli



```
vi script2.sh
```



Script 2



```
#!/bin/bash  
#install terraform  
sudo apt install wget -y  
wget -O- https://apt.releases.hashicorp.com/gpg | sudo gpg --dearmor -o  
echo "deb [signed-by=/usr/share/keyrings/hashicorp-archive-keyring.gpg]  
sudo apt update && sudo apt install terraform  
  
#install Kubectl on Jenkins  
sudo apt update
```

```
sudo apt install curl -y
curl -LO https://dl.k8s.io/release/$(curl -L -s https://dl.k8s.io/release/stable.txt)
sudo install -o root -g root -m 0755 kubectl /usr/local/bin/kubectl
kubectl version --client
```

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

Give permissions

```
sudo chmod 777 script2.sh
sh script2.sh
```



Now Run sonarqube container

```
sudo chmod 777 /var/run/docker.sock
docker run -d --name sonar -p 9000:9000 sonarqube:lts-community
```



Ec2 is created in the Aws console

Instances (1) Info								C	Connect	Instance state ▾	Actions ▾	Launch instances ▾	
								Find Instance by attribute or tag (case-sensitive)	Clear filters	<	1	>	@
<input type="checkbox"/>	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Pub					
<input type="checkbox"/>	Jenkins-ARGO	i-0323f37f837248e53	Running Q Q	t2.large	Initializing	No alarms +	ap-south-1b	ec2					

Now copy the public IP address of ec2 and paste it into the browser



Ec2-ip:8080 #you will Jenkins login page



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

Continue

Connect your Instance to Putty or Mobaxtreme and provide the below command for the Administrator password



`sudo cat /var/lib/jenkins/secrets/initialAdminPassword`



```
ubuntu@ip-172-31-33-57:~$ sudo cat /var/lib/jenkins/secrets/initialAdminPassword
0ed1cb07ea7447c5a47d723022e74968
ubuntu@ip-172-31-33-57:~$ █
```

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.

Jenkins will now get installed and install all the libraries.

Create an admin user

Getting Started

Create First Admin User

Username

Password

Confirm password

Full name

E-mail address

Jenkins 2.414.1

Skip and continue as admin

Save and Continue

Click on save and continue.

Jenkins Dashboard

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



ec2-ip:9000 #runs sonar container



Instances | EC2 | ap-south-1 | petstore Config [Jenkins] | SonarQube

52.66.140.95:9000/sessions/new?return_to=%2F

Login

Password

Log in Cancel

Enter username and password, click on login and change password



```
username admin
password admin
```

Instances | EC2 | ap-south-1 | petstore Config [Jenkins] | SonarQube | +

6.140.95.9000/account/reset_password

Web Servic... Coaching on DevO... LinkedIn Docker — A Beginn... Cloud Quest 100+ Essential Com... Advanced End-to-E...

Update your password

This account should not use the default password.

Enter a new password

All fields marked with * are required

Old Password *

New Password *

Confirm Password *

Update

Update New password, This is Sonar Dashboard.

Not secure | 52.66.140.95:9000/projects/create

Gmail YouTube Amazon Web Service... Coaching on DevO... LinkedIn Docker — A Beginn... Cloud Quest 100+ Essential Com... Advanced End-to-E... LINUX - YouTube T... How to Install Jenk...

sonarqube Projects Issues Rules Quality Profiles Quality Gates Administration

Search for projects... A

How do you want to create your project?

Do you want to benefit from all of SonarQube's features (like repository import and Pull Request decoration)? Create your project from your favorite DevOps platform. First, you need to set up a DevOps platform configuration.

From Azure DevOps **From Bitbucket Server** **From Bitbucket Cloud** **From GitHub** **From GitLab**

Set up global configuration Set up global configuration Set up global configuration Set up global configuration Set up global configuration

Now go to Putty and see whether it's installed docker, Terraform, Aws cli, Kubectl or not.



```
docker --version  
aws --version  
terraform --version  
kubectl version
```



```
ubuntu@ip-172-31-11-71:~$  
ubuntu@ip-172-31-11-71:~$ trivy --version  
Version: 0.46.0  
ubuntu@ip-172-31-11-71:~$  
ubuntu@ip-172-31-11-71:~$ aws --version  
aws-cli/2.13.29 Python/3.11.6 Linux/5.19.0-1025-aws exe/x86_64.ubuntu.22 prompt/off  
ubuntu@ip-172-31-11-71:~$  
ubuntu@ip-172-31-11-71:~$ terraform --version  
Terraform v1.6.2  
on linux_amd64  
ubuntu@ip-172-31-11-71:~$  
ubuntu@ip-172-31-11-71:~$ kubectl --version  
error: unknown flag: --version  
See 'kubectl --help' for usage.  
ubuntu@ip-172-31-11-71:~$ kubectl version  
Client Version: v1.28.3  
Kustomize Version: v5.0.4-0.20230601165947-6ce0bf390ce3  
Error from server (Forbidden): <html><head><meta http-equiv='refresh' content='1;url=/login?from=%2Fvers<br>timeout%3D32s' );</script></head><body style='background-color:white; color:white;'>  
  
Authentication required  
<!--  
-->  
  
</body></html>  
ubuntu@ip-172-31-11-71:~$ █
```

Step 3: Jenkins Job Configuration

Step 3A: EKS Provision job

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.

The screenshot shows the Jenkins plugin store interface. A search bar at the top contains the text "Terraform". Below it, a table lists a single plugin: "Terraform 1.0.10". The "Install" button next to it is greyed out, indicating it's already installed. The "Name" column shows "Terraform" and the "Released" column shows "3 yr 8 mo ago". A note below the table states: "This plugin provides a build wrapper for [Terraform](#) to launch and destroy infrastructure."

Go to Putty and use the below command

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



```
which terraform
```



The screenshot shows a terminal window in Putty. The title bar says "Quick connect...". The window displays the command "which terraform" followed by its output: "/usr/bin/terraform". The prompt "ubuntu@ip-172-31-42-229:~\$" appears twice.

Now come back to Manage Jenkins -> Tools

Add the terraform in Tools

Terraform installations

Add Terraform

Terraform

Name

Install directory

Install automatically ?

Add Terraform

Save
Apply

Apply and save.

CHANGE YOUR S3 BUCKET NAME IN THE BACKEND.TF

Now create a new job for the Eks provision

Enter an item name

» Required field

Freestyle project



This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.

Pipeline



Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.

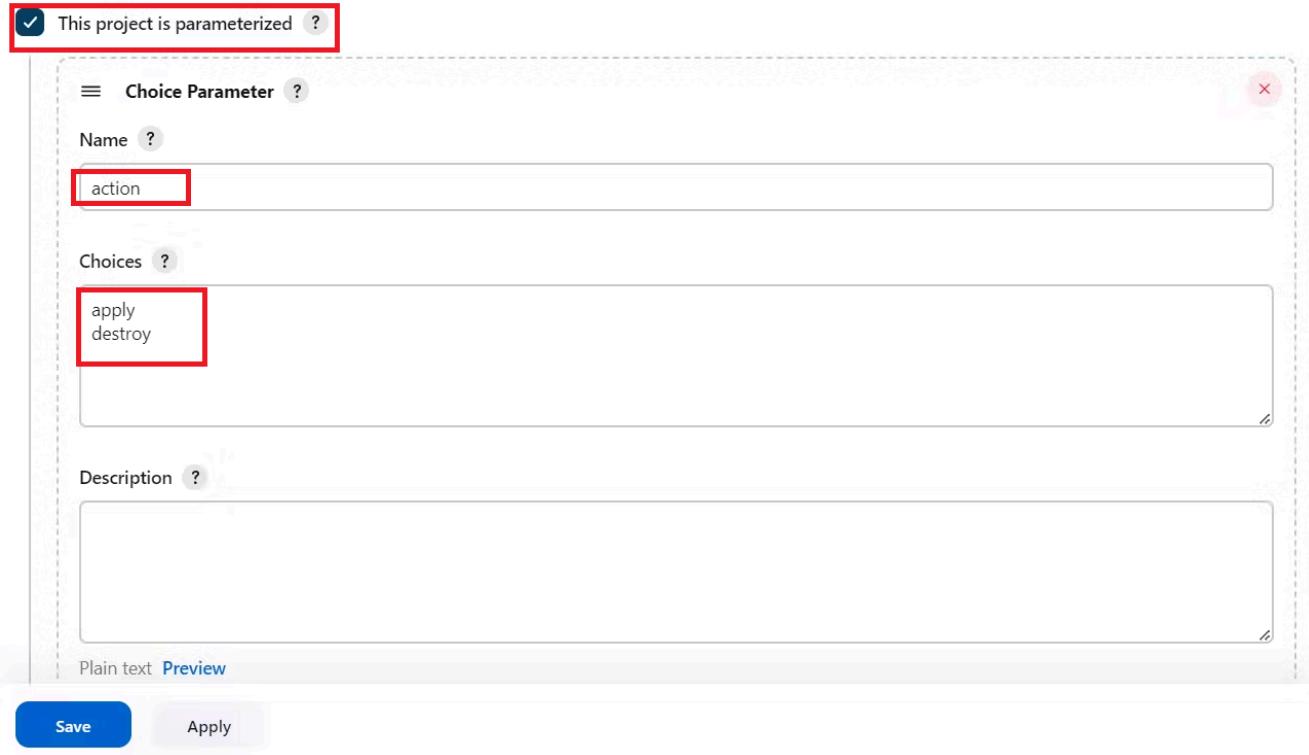
Multi-configuration project



Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds etc.

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



Let's add a pipeline



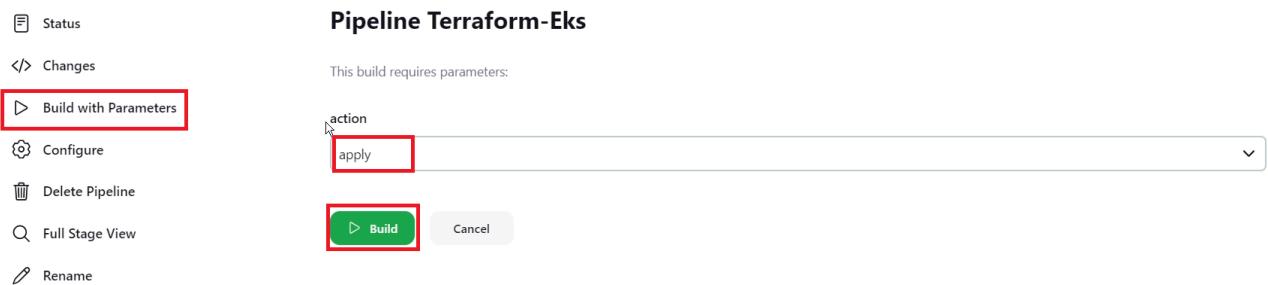
```
pipeline{
    agent any
    stages {
        stage('Checkout from Git'){
            steps{
                git branch: 'main', url: 'https://github.com/Aj7Ay/Hotstar-Clone'
            }
        }
        stage('Terraform version'){
            steps{
                sh 'terraform --version'
            }
        }
        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



Stage view it will take max 10mins to provision

Pipeline Terraform-Eks

- Status
- </> Changes
- Eks from Jenkins
- Build with Parameters
- Configure
- Delete Pipeline
- Full Stage View
- Rename
- Pipeline Syntax
- Build History
- trend
- Filter builds...

Average stage times:
(Average full run time: ~9min 49s)

Checkout	terraform init	terraform validate	terraform plan	terraform Apply/destroy
4s	5s	3s	4s	9min 28s
4s	5s	3s	4s	9min 28s

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

EKS > Clusters

Clusters (1) Info

Filter clusters

Cluster name	Status	Kubernetes version	Provider
EKS_CLOUD	Active	1.28	EKS

Ec2 instance is created for the Node group

Instances (1/2) Info

Find Instance by attribute or tag (case-sensitive)

Instance state = running

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Put
Jenkins-ARGO	i-0323f37f837248e53	Running	t2.large	2/2 checks passed	No alarms	ap-south-1b	ec2
	i-049634a401c64808b	Running	t2.medium	2/2 checks passed	No alarms	ap-south-1b	ec2

Step 3B: Hotstar job

Plugins installation & setup (Java, Sonar, Nodejs, owasp, Docker)

Go to Jenkins dashboard

Manage Jenkins -> Plugins -> Available Plugins

Search for the Below Plugins

Eclipse Temurin installer

Sonarqube Scanner

NodeJs

Owasp Dependency-Check

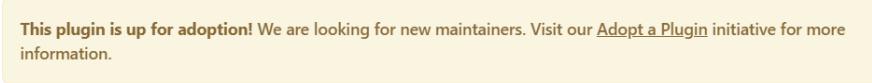
Docker

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	 <p>This plugin is up for adoption! We are looking for new maintainers. Visit our Adopt a Plugin initiative for more information.</p>	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

<input checked="" type="checkbox"/>	Docker Commons 439.va_3cb_0a_6a_fb_29	3 mo 17 days ago
Provides the common shared functionality for various Docker-related plugins.		
<input checked="" type="checkbox"/>	Docker Pipeline 572.v950f58993843	2 mo 15 days ago
Build and use Docker containers from pipelines.		
<input checked="" type="checkbox"/>	Docker API 3.3.1-79.v20b_53427e041	4 mo 22 days ago
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.		
<input checked="" type="checkbox"/>	docker-build-step 2.10	

Configure in Global Tool Configuration

Goto Manage Jenkins → Tools → Install JDK(17) and NodeJs(16) → Click on Apply and Save

NOTE: USE ONLY NODE JS 16

Dashboard > Manage Jenkins > Tools
JDK Installations

Add JDK

JDK

Name: jdk17

Install automatically:

Install from adoptium.net:

Version: jdk-17.0.8.1+1

Add Installer

Dashboard > Manage Jenkins > Tools

NodeJS

Name: node16

Install automatically:

Install from nodejs.org:

Version: NodeJS 16.2.0

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

The screenshot shows the Jenkins Manage Jenkins > Tools page. Under the SonarQube Scanner installations section, a new configuration is being added. The 'Name' field is set to 'sonar-scanner'. The 'Install automatically' checkbox is checked. Under the 'Install from Maven Central' section, the 'Version' dropdown is set to 'SonarQube Scanner 5.0.1.3006'. There is also an 'Add Installer' button. At the bottom, there are 'Save' and 'Apply' buttons.

For Owasp use the 9.0.7 version

The screenshot shows the Jenkins Manage Jenkins > Tools page. Under the Dependency-Check installations section, a new configuration is being added. The 'Name' field is set to 'DP-Check'. The 'Install automatically' checkbox is checked. Under the 'Install from github.com' section, the 'Version' dropdown is set to 'dependency-check 9.0.7'. There is also an 'Add Installer' button. At the bottom, there are 'Save' and 'Apply' buttons.

Use the latest version of Docker

Add Docker

Docker

Name

docker

Install automatically ?

Download from docker.com

Docker version ?

latest

Add Installer ▾

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

sonarqube

Projects Issues Rules Quality Profiles Quality Gates Administration

Administration

Configuration ▾ Security ▾ Projects ▾ System Marketplace

General

Users (highlighted)

Groups

Global Permissions

Permission Templates

Find i...

click on update Token

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

Update Tokens

Create a token with a name and generate

Tokens of Administrator

Generate Tokens

Name	Expires in
Enter Token Name	30 days
<input type="button" value="Generate"/>	

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

Copy `squ_21d162904c1c72cf8b39665f96480185c99dc2f9`

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

copy Token

Goto 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

Create

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.

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

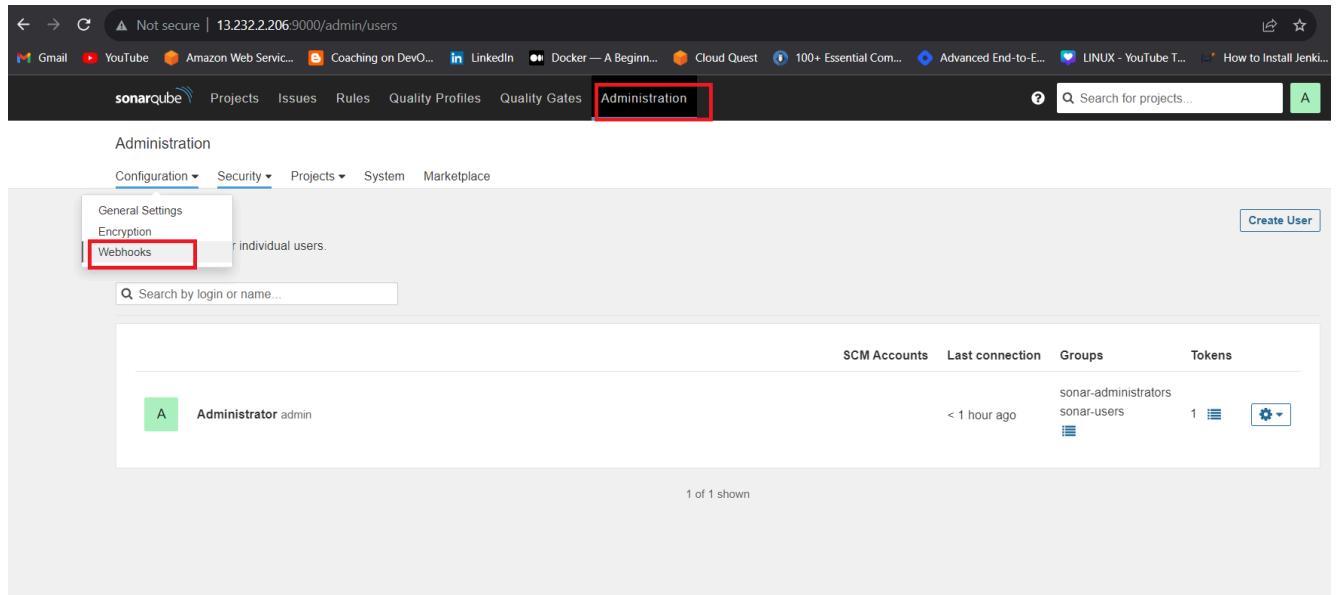
List of SonarQube installations

Name	<input type="text" value="sonar-server"/>	X
Server URL	Default is http://localhost:9000 <input type="text" value="http://13.232.17.191:9000"/>	
Server authentication token	SonarQube authentication token. Mandatory when anonymous access is disabled. <input type="text" value="Sonar-token"/>	
<input style="margin-right: 10px;" type="button" value="Add"/> <input type="button" value="Save"/> <input type="button" value="Apply"/>		

Click on Apply and Save

In the Sonarqube Dashboard add a quality gate also

Administration-> Configuration->Webhooks



The screenshot shows the SonarQube Administration interface. The top navigation bar has a red box around the 'Administration' tab. Below it, the main menu includes 'Configuration', 'Security', 'Projects', 'Issues', 'Rules', 'Quality Profiles', 'Quality Gates', and 'Administration'. Under 'Administration', there are tabs for 'General Settings', 'Encryption', and 'Webhooks', with a red box around the 'Webhooks' tab. A search bar is present. The main content area displays user information for 'Administrator admin'. The table columns are 'SCM Accounts', 'Last connection', 'Groups', and 'Tokens'. The user has 'sonar-administrators' and 'sonar-users' groups, and 1 token. At the bottom, it says '1 of 1 shown'.

Click on Create

No webhook defined.

Create

Add details



```
#in url section of quality gate
http://jenkins-public-ip:8080;/sonarqube-webhook/
```



Create Webhook

All fields marked with * are required

Name *

jenkins

URL *

http://43.204.36.242:8090/sonarqube-webhook/

Secret

If provided, secret will be used as the key to generate the HMAC hex (lowercase) digest value in the 'X-Sonar-Webhook-HMAC-SHA256' header.

Create Cancel

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

Kind

Username with password

Scope ?

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

Username ?

sevenajay

 Treat username as secret ?

Password ?

.....

ID ?

docker

Description ?

docker

Create

Create.

Pipeline upto Docker

Now let's create a new job for our pipeline

Enter an item name

HOTSTAR

» Required field

MR CLOUD BOOK

Freestyle project

This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.

Maven project

Build a maven project. Jenkins takes advantage of your POM files and drastically reduces the configuration.

Pipeline

Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.

Before Adding pipeline install Docker Scout

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

```
curl -sSfL https://raw.githubusercontent.com/docker/scout-cli/main/install.sh | sh
```

Adc...to .yaml



```
pipeline{
    agent any
    tools{
        jdk 'jdk17'
        nodejs 'node16'
    }
    environment {
        SCANNER_HOME=tool 'sonar-scanner'
    }
    stages {
        stage('clean workspace'){
            steps{
                cleanWs()
            }
        }
        stage('Checkout from Git'){
            steps{
                git branch: 'main', url: 'https://github.com/Aj7Ay/Hotstar-Clone'
            }
        }
        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, credentials: []
                }
            }
        }
    }
}
```

```
stage('Install Dependencies') {
    steps {
        sh "npm install"
    }
}

stage('OWASP FS SCAN') {
    steps {
        dependencyCheck additionalArguments: '--scan ./ --disableCheckForVulnerableDependencies'
        dependencyCheckPublisher pattern: '**/dependency-check-report.html'
    }
}

stage('Docker Scout FS') {
    steps {
        script{
            withDockerRegistry(credentialsId: 'docker', toolName: 'Docker Scout')
            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 sevenajay/hotstar:latest"
            sh "docker push sevenajay/hotstar:latest"
        }
    }
}

stage('Docker Scout Image') {
    steps {
        script{
            withDockerRegistry(credentialsId: 'docker', toolName: 'Docker Scout')
            sh 'docker-scout quickview sevenajay/hotstar:latest'
            sh 'docker-scout cves sevenajay/hotstar:latest'
            sh 'docker-scout recommendations sevenajay/hotstar:latest'
        }
    }
}

stage("deploy_docker"){
    steps{

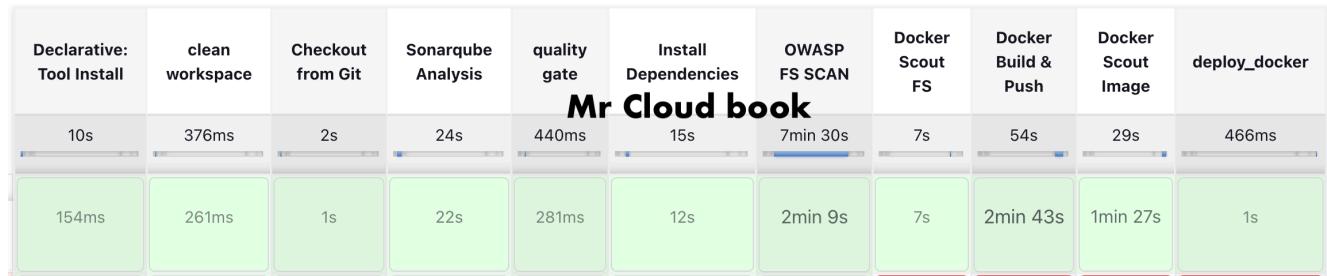
```

```
        sh "docker run -d --name hotstar -p 3000:3000 sevenajay,
    }
}
}
}
```

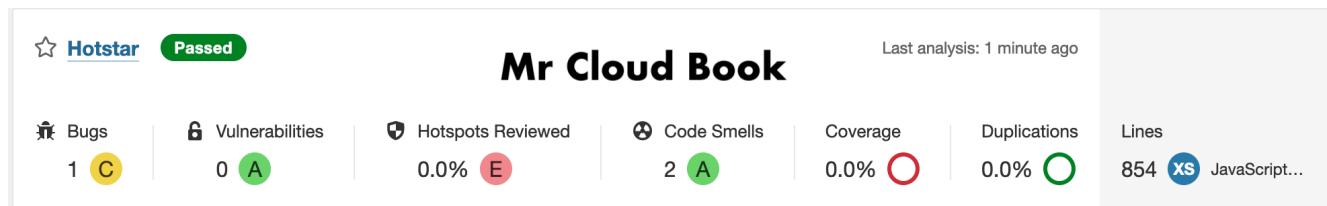
Click on Apply and save.

Build now

Stage view



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



You can see the report has been generated and the status shows as passed. You can see that there are 854 lines it scanned. To see a detailed report, you can go to issues.

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

SEVERITY DISTRIBUTION		2	9	1
File Name	Vulnerability	Severity	Weakness	
+ json5:1.0.1	[NVD] CVE-2022-46175	⚡ High	CWE-1321	
+ jszip.js	[NVD] CVE-2022-48285	⚡ High	CWE-22	
+ jszip.min.js	[NVD] CVE-2022-48285	⚡ High	CWE-22	
+ jszip:3.7.1	[NVD] CVE-2022-48285	⚡ High	CWE-22	
+ minimist:1.2.5	[NVD] CVE-2021-44906	⚡ Critical	CWE-1321	
+ next:12.0.10	[OSSINDEX] CVE-2022-23646	⚡ High	CWE-451	
+ postcss:8.4.6	[NVD] CVE-2023-44270	⚡ Medium	CWE-74	
+ protobufjs:6.11.2	[NVD] CVE-2023-36665	⚡ Critical	CWE-1321	
+ protobufjs:6.11.2	[NVD] CVE-2022-25878	⚡ High	CWE-1321	
+ semver:6.3.0	[OSSINDEX] CVE-2022-25883	⚡ High	CWE-1333	

Mr cloud book

Let's See Docker Scout File scan report

```
[Pipeline] sh
+ docker-scout quickview fs://.
...Reading file system
✓ File system read
...Indexing
✓ Indexed 1257 packages
```

Target	fs://.	1C	1H	3M	0L
--------	--------	----	----	----	----

What's Next?

View vulnerabilities → docker scout cves fs://.

```
[Pipeline] sh
+ docker-scout cves fs://.
...Reading file system
✓ File system read
...Indexing
✓ Indexed 1257 packages
✗ Detected 5 vulnerable packages with a total of 5 vulnerabilities
```

Mr Cloud book

Overview

		Analyzed path			
Target	fs://.	1C	1H	3M	0L
vulnerabilities		1C	1H	3M	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
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

 sevenajay / hotstar

Description

This simple Hotstar clone added with DevSecOps  Last pushed: a day ago

Docker commands

To push a new tag to this repository:

[Public View](#)

docker push sevenajay/hotstar:tagname

Let's See Docker Scout Image analysis

Quickview

```
+ docker-scout quickview sevenajay/hotstar:latest
...Storing image for indexing
✓ Image stored for indexing
...Indexing
✓ Indexed 1456 packages

Target          | sevenajay/hotstar:latest | 1C   1H   3M   0L
digest         | 90df4be4e344           | 
Base image     | node:21-alpine        | 0C   0H   0M   0L
Updated base image | node:20-alpine        | 0C   0H   0M   0L
```

Cves

```
+ docker-scout cves sevenajay/hotstar:latest
  ✓ SBOM of image already cached, 1456 packages indexed
  ✗ Detected 5 vulnerable packages with a total of 5 vulnerabilities
```

Overview

Analyzed Image				
Target	sevenajay/hotstar:latest			
digest	90df4be4e344			
platform	linux/amd64			
vulnerabilities	1C	1H	3M	0L
size	236 MB			
packages	1456			

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>

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 sevenajay/hotstar:latest
```

✓ SBOM of image already cached, 1456 packages indexed

```
Target | sevenajay/hotstar:latest
digest | 90df4be4e344
```

```
## Recommended fixes
```

```
Base image is node:21-alpine
```

Name	21-alpine
Digest	sha256:9b54d010b382f0ef176dc93cd829bd4f2a905092b260746b3999aa824c9b7121
Vulnerabilities	0C 0H 0M 0L
Pushed	1 week ago
Size	49 MB
Packages	240
Flavor	alpine
OS	3.19
Runtime	19

Deploy to Container



ec2-ip:3000



Output



Go to Putty of your Jenkins instance SSH and enter the below command



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



```
ubuntu@ip-172-31-11-71:~$  
ubuntu@ip-172-31-11-71:~$ aws eks update-kubeconfig --name EKS_CLOUD --region ap-south-1  
Added new context arn:aws:eks:ap-south-1:07201807785:cluster/EKS_CLOUD to /home/ubuntu/.kube/config  
ubuntu@ip-172-31-11-71:~$  
ubuntu@ip-172-31-11-71:~$  
ubuntu@ip-172-31-11-71:~$ █
```

Let's see the nodes



```
kubectl get nodes
```



```
ubuntu@ip-172-31-11-71:~$  
ubuntu@ip-172-31-11-71:~$ kubectl get nodes  
NAME           STATUS   ROLES      AGE    VERSION  
ip-172-31-13-85.ap-south-1.compute.internal   Ready     <none>    111m   v1.28.1-eks-43840fb  
ubuntu@ip-172-31-11-71:~$ █
```

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

```

/drvess/c/Users/Admin/Downloads Terminal Sessions View X server Tools Games Settings Macros Help Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help Quick connect... / 1. /drvess/c/Users/Admin/Downloads / 2. /drvess/c/Users/Admin/Downloads / 3. /drvess/c/Users/Admin/Downloads
cat: c: No such file or directory
ubuntu@ip-172-31-40-131:~/kubek$ cat config
apiVersion: v1
clusters:
- cluster:
  certificate-authority-data: LS0tLS1CRUJDTlBDRVJSUZJ00FURS0tLS0tCk1JSURJVEND0Wdzt0F3SUJBz0Lj0i9gYl12bwU0Vgd3RFZSktvklodmN00VFTEJR0xGVEVUTUJFR0ExVUJK0XhNS2EzVm1awEp1wlhsbGn60WVG
dzB5TxBpNU1Ez3doakzTVRSYUZMph0ekETURVd0qTXlNWFjhTU1VeAp+kF5m0d0VkbTlVRDbsXoxw1hNeekfTv1jBaWEl132dFa1U1BMeDU3FHb01LM0RRR0JU8VVR0TR0K30XhZ0VlClkVs1JBUIN3THhcbG02T3hk1T3RjVUz7Vdcyrt1
ZDj6R1Vjbp01YiN00982Mm0V0TE1Jb2ykpdyt0vYso2X203Lw082Yj1wHfFWkcd012G0mfCMh05b1l05zfscF1Bd2pLcJRCtU13dGoycW0b1dNtTh35fPMsD1TkwpsR0xGr1TwpXHeHb02c8zUFNNT3NFtKwdhNkY2Lzd2wQmUSehx
K0R2zuJb010YiN00982Mm0V0TE1Jb2ykpdyt0vYso2X203Lw082Yj1wHfFWkcd012G0mfCMh05b1l05zfscF1Bd2pLcJRCtU13dGoycW0b1dNtTh35fPMsD1TkwpsR0xGr1TwpXHeHb02c8zUFNNT3NFtKwdhNkY2Lzd2wQmUSehx
bmxtuEs93a1B1VXB0menczj1BbpoY24d4TNC0nLxEhloXojoYj0iRmTSVHTYsUz0qMh0N0kfBRp2XvEYTUE0R0exWMrEd0VCL3dRRUf3SUmUREFOCkjnT1ZUk1C0WY4R0JU0URBuIjgjTU1iR0ExWREZfX0k1TjkahWRCvnt2ZrLcJvceF9avdQ
cWdd1JNkR1Nq0pVYK0md0VksnRUeakT2ndwmCmxjg1bVtVsLd01Vb0TR1j0kFkR0q0wvmsnfUEN6Wvpz03c131UU0xRvhsGxZtEJrS+TSGFgYmR0yGd0UTR1k55a9q1eYhRfCvVEnLvk84a1Vnfed
ZKzYm9yUzFVUML2RJDD0kVc09oSDNC5HRYtH1LSC9J3zhWfT3ijFXXVhmlDdytHkw1pCmhd0Yn1k2rJHSUs2R0wKL0vavBraojPUtErYk1J5mY5L3W0KWXEx0n5NhlhPV2VXYzJuY0FraBDU3hqeLzQJUVSSjkycn10R3nV1jMgo1TC9w
bmRpJXUJRU5Etdcg54UJhLteN5STfXSwBx0UzWdh1Psek3MYTZEeuH02VtYxewLyt6Rd0c35h6aHRt2VNCUm0mRqpaEJkRcvx3pUnklywJmZ1gbmJkc0uyjVJQJZXSxbtqTfUvQytrUv20rVgvcmveIxbaBhShkZ3VbzFrdhN1D80
c0tLS0tLS0tLSEtNfUtrRk10dQvRFLs0tLS0k
  server: https://172.31.40.131:6443
  name: kubernetes
contexts:
- context:
  cluster: kubernetes
  user: kubernetes-admin
  name: kubernetes-admin@kubernetes
current-context: kubernetes-admin@kubernetes
kind: Config
preferences: {}
users:
- name: kubernetes-admin
  user:
    client-certificate-data: LS0tLS1CRUJDTlBDRVJSUZJ00FURS0tLS0tCk1JSURJVEND0Wdzt0F3SUJBz0Lj0i9gYl12bwU0Vgd3RFZSktvklodmN00VFTEJR0xGVEVUTUJFR0ExVUJK0XhNS2EzVm1awEp1wlhsbGn60WVG
  dzB5TxBpNU1Ez3doakzTVRSYUZMph0ekETURVd0qTXlNWFjhTU1VeAp+kF5m0d0VkbTlVRDbsXoxw1hNeekfTv1jBaWEl132dFa1U1BMeDU3FHb01LM0RRR0JU8VVR0TR0K30XhZ0VlClkVs1JBUIN3THhcbG02T3hk1T3RjVUz7Vdcyrt1
  ZDj6R1Vjbp01YiN00982Mm0V0TE1Jb2ykpdyt0vYso2X203Lw082Yj1wHfFWkcd012G0mfCMh05b1l05zfscF1Bd2pLcJRCtU13dGoycW0b1dNtTh35fPMsD1TkwpsR0xGr1TwpXHeHb02c8zUFNNT3NFtKwdhNkY2Lzd2wQmUSehx
  K0R2zuJb010YiN00982Mm0V0TE1Jb2ykpdyt0vYso2X203Lw082Yj1wHfFWkcd012G0mfCMh05b1l05zfscF1Bd2pLcJRCtU13dGoycW0b1dNtTh35fPMsD1TkwpsR0xGr1TwpXHeHb02c8zUFNNT3NFtKwdhNkY2Lzd2wQmUSehx
  bmxtuEs93a1B1VXB0menczj1BbpoY24d4TNC0nLxEhloXojoYj0iRmTSVHTYsUz0qMh0N0kfBRp2XvEYTUE0R0exWMrEd0VCL3dRRUf3SUmUREFOCkjnT1ZUk1C0WY4R0JU0URBuIjgjTU1iR0ExWREZfX0k1TjkahWRCvnt2ZrLcJvceF9avdQ
  cWdd1JNkR1Nq0pVYK0md0VksnRUeakT2ndwmCmxjg1bVtVsLd01Vb0TR1j0kFkR0q0wvmsnfUEN6Wvpz03c131UU0xRvhsGxZtEJrS+TSGFgYmR0yGd0UTR1k55a9q1eYhRfCvVEnLvk84a1Vnfed
  ZKzYm9yUzFVUML2RJDD0kVc09oSDNC5HRYtH1LSC9J3zhWfT3ijFXXVhmlDdytHkw1pCmhd0Yn1k2rJHSUs2R0wKL0vavBraojPUtErYk1J5mY5L3W0KWXEx0n5NhlhPV2VXYzJuY0FraBDU3hqeLzQJUVSSjkycn10R3nV1jMgo1TC9w
  bmRpJXUJRU5Etdcg54UJhLteN5STfXSwBx0UzWdh1Psek3MYTZEeuH02VtYxewLyt6Rd0c35h6aHRt2VNCUm0mRqpaEJkRcvx3pUnklywJmZ1gbmJkc0uyjVJQJZXSxbtqTfUvQytrUv20rVgvcmveIxbaBhShkZ3VbzFrdhN1D80
  c0tLS0tLS0tLSEtNfUtrRk10dQvRFLs0tLS0k
  server: https://172.31.40.131:6443
  name: kubernetes
contexts:
- context:
  cluster: kubernetes
  user: kubernetes-admin
  name: kubernetes-admin@kubernetes
current-context: kubernetes-admin@kubernetes
kind: Config
preferences: {}
users:
- name: kubernetes-admin
  user:
    client-certificate-data: LS0tLS1CRUJDTlBDRVJSUZJ00FURS0tLS0tCk1JSURJVEND0Wdzt0F3SUJBz0Lj0i9gYl12bwU0Vgd3RFZSktvklodmN00VFTEJR0xGVEVUTUJFR0ExVUJK0XhNS2EzVm1awEp1wlhsbGn60WVG
  dzB5TxBpNU1Ez3doakzTVRSYUZMph0ekETURVd0qTXlNWFjhTU1VeAp+kF5m0d0VkbTlVRDbsXoxw1hNeekfTv1jBaWEl132dFa1U1BMeDU3FHb01LM0RRR0JU8VVR0TR0K30XhZ0VlClkVs1JBUIN3THhcbG02T3hk1T3RjVUz7Vdcyrt1
  ZDj6R1Vjbp01YiN00982Mm0V0TE1Jb2ykpdyt0vYso2X203Lw082Yj1wHfFWkcd012G0mfCMh05b1l05zfscF1Bd2pLcJRCtU13dGoycW0b1dNtTh35fPMsD1TkwpsR0xGr1TwpXHeHb02c8zUFNNT3NFtKwdhNkY2Lzd2wQmUSehx
  K0R2zuJb010YiN00982Mm0V0TE1Jb2ykpdyt0vYso2X203Lw082Yj1wHfFWkcd012G0mfCMh05b1l05zfscF1Bd2pLcJRCtU13dGoycW0b1dNtTh35fPMsD1TkwpsR0xGr1TwpXHeHb02c8zUFNNT3NFtKwdhNkY2Lzd2wQmUSehx
  bmxtuEs93a1B1VXB0menczj1BbpoY24d4TNC0nLxEhloXojoYj0iRmTSVHTYsUz0qMh0N0kfBRp2XvEYTUE0R0exWMrEd0VCL3dRRUf3SUmUREFOCkjnT1ZUk1C0WY4R0JU0URBuIjgjTU1iR0ExWREZfX0k1TjkahWRCvnt2ZrLcJvceF9avdQ
  cWdd1JNkR1Nq0pVYK0md0VksnRUeakT2ndwmCmxjg1bVtVsLd01Vb0TR1j0kFkR0q0wvmsnfUEN6Wvpz03c131UU0xRvhsGxZtEJrS+TSGFgYmR0yGd0UTR1k55a9q1eYhRfCvVEnLvk84a1Vnfed
  ZKzYm9yUzFVUML2RJDD0kVc09oSDNC5HRYtH1LSC9J3zhWfT3ijFXXVhmlDdytHkw1pCmhd0Yn1k2rJHSUs2R0wKL0vavBraojPUtErYk1J5mY5L3W0KWXEx0n5NhlhPV2VXYzJuY0FraBDU3hqeLzQJUVSSjkycn10R3nV1jMgo1TC9w
  bmRpJXUJRU5Etdcg54UJhLteN5STfXSwBx0UzWdh1Psek3MYTZEeuH02VtYxewLyt6Rd0c35h6aHRt2VNCUm0mRqpaEJkRcvx3pUnklywJmZ1gbmJkc0uyjVJQJZXSxbtqTfUvQytrUv20rVgvcmveIxbaBhShkZ3VbzFrdhN1D80
  c0tLS0tLS0tLSEtNfUtrRk10dQvRFLs0tLS0k
  server: https://172.31.40.131:6443
  name: kubernetes
contexts:
- context:
  cluster: kubernetes
  user: kubernetes-admin
  name: kubernetes-admin@kubernetes
current-context: kubernetes-admin@kubernetes
kind: Config
preferences: {}
users:
- name: kubernetes-admin
  user:
    client-certificate-data: LS0tLS1CRUJDTlBDRVJSUZJ00FURS0tLS0tCk1JSURJVEND0Wdzt0F3SUJBz0Lj0i9gYl12bwU0Vgd3RFZSktvklodmN00VFTEJR0xGVEVUTUJFR0ExVUJK0XhNS2EzVm1awEp1wlhsbGn60WVG
  dzB5TxBpNU1Ez3doakzTVRSYUZMph0ekETURVd0qTXlNWFjhTU1VeAp+kF5m0d0VkbTlVRDbsXoxw1hNeekfTv1jBaWEl132dFa1U1BMeDU3FHb01LM0RRR0JU8VVR0TR0K30XhZ0VlClkVs1JBUIN3THhcbG02T3hk1T3RjVUz7Vdcyrt1
  ZDj6R1Vjbp01YiN00982Mm0V0TE1Jb2ykpdyt0vYso2X203Lw082Yj1wHfFWkcd012G0mfCMh05b1l05zfscF1Bd2pLcJRCtU13dGoycW0b1dNtTh35fPMsD1TkwpsR0xGr1TwpXHeHb02c8zUFNNT3NFtKwdhNkY2Lzd2wQmUSehx
  K0R2zuJb010YiN00982Mm0V0TE1Jb2ykpdyt0vYso2X203Lw082Yj1wHfFWkcd012G0mfCMh05b1l05zfscF1Bd2pLcJRCtU13dGoycW0b1dNtTh35fPMsD1TkwpsR0xGr1TwpXHeHb02c8zUFNNT3NFtKwdhNkY2Lzd2wQmUSehx
  bmxtuEs93a1B1VXB0menczj1BbpoY24d4TNC0nLxEhloXojoYj0iRmTSVHTYsUz0qMh0N0kfBRp2XvEYTUE0R0exWMrEd0VCL3dRRUf3SUmUREFOCkjnT1ZUk1C0WY4R0JU0URBuIjgjTU1iR0ExWREZfX0k1TjkahWRCvnt2ZrLcJvceF9avdQ
  cWdd1JNkR1Nq0pVYK0md0VksnRUeakT2ndwmCmxjg1bVtVsLd01Vb0TR1j0kFkR0q0wvmsnfUEN6Wvpz03c131UU0xRvhsGxZtEJrS+TSGFgYmR0yGd0UTR1k55a9q1eYhRfCvVEnLvk84a1Vnfed
  ZKzYm9yUzFVUML2RJDD0kVc09oSDNC5HRYtH1LSC9J3zhWfT3ijFXXVhmlDdytHkw1pCmhd0Yn1k2rJHSUs2R0wKL0vavBraojPUtErYk1J5mY5L3W0KWXEx0n5NhlhPV2VXYzJuY0FraBDU3hqeLzQJUVSSjkycn10R3nV1jMgo1TC9w
  bmRpJXUJRU5Etdcg54UJhLteN5STfXSwBx0UzWdh1Psek3MYTZEeuH02VtYxewLyt6Rd0c35h6aHRt2VNCUm0mRqpaEJkRcvx3pUnklywJmZ1gbmJkc0uyjVJQJZXSxbtqTfUvQytrUv20rVgvcmveIxbaBhShkZ3VbzFrdhN1D80
  c0tLS0tLS0tLSEtNfUtrRk10dQvRFLs0tLS0k
  server: https://172.31.40.131:6443
  name: kubernetes
contexts:
- context:
  cluster: kubernetes
  user: kubernetes-admin
  name: kubernetes-admin@kubernetes
current-context: kubernetes-admin@kubernetes
kind: Config
preferences: {}
users:
- name: kubernetes-admin
  user:
    client-certificate-data: LS0tLS1CRUJDTlBDRVJSUZJ00FURS0tLS0tCk1JSURJVEND0Wdzt0F3SUJBz0Lj0i9gYl12bwU0Vgd3RFZSktvklodmN00VFTEJR0xGVEVUTUJFR0ExVUJK0XhNS2EzVm1awEp1wlhsbGn60WVG
  dzB5TxBpNU1Ez3doakzTVRSYUZMph0ekETURVd0qTXlNWFjhTU1VeAp+kF5m0d0VkbTlVRDbsXoxw1hNeekfTv1jBaWEl132dFa1U1BMeDU3FHb01LM0RRR0JU8VVR0TR0K30XhZ0VlClkVs1JBUIN3THhcbG02T3hk1T3RjVUz7Vdcyrt1
  ZDj6R1Vjbp01YiN00982Mm0V0TE1Jb2ykpdyt0vYso2X203Lw082Yj1wHfFWkcd012G0mfCMh05b1l05zfscF1Bd2pLcJRCtU13dGoycW0b1dNtTh35fPMsD1TkwpsR0xGr1TwpXHeHb02c8zUFNNT3NFtKwdhNkY2Lzd2wQmUSehx
  K0R2zuJb010YiN00982Mm0V0TE1Jb2ykpdyt0vYso2X203Lw082Yj1wHfFWkcd012G0mfCMh05b1l05zfscF1Bd2pLcJRCtU13dGoycW0b1dNtTh35fPMsD1TkwpsR0xGr1TwpXHeHb02c8zUFNNT3NFtKwdhNkY2Lzd2wQmUSehx
  bmxtuEs93a1B1VXB0menczj1BbpoY24d4TNC0nLxEhloXojoYj0iRmTSVHTYsUz0qMh0N0kfBRp2XvEYTUE0R0exWMrEd0VCL3dRRUf3SUmUREFOCkjnT1ZUk1C0WY4R0JU0URBuIjgjTU1iR0ExWREZfX0k1TjkahWRCvnt2ZrLcJvceF9avdQ
  cWdd1JNkR1Nq0pVYK0md0VksnRUeakT2ndwmCmxjg1bVtVsLd01Vb0TR1j0kFkR0q0wvmsnfUEN6Wvpz03c131UU0xRvhsGxZtEJrS+TSGFgYmR0yGd0UTR1k55a9q1eYhRfCvVEnLvk84a1Vnfed
  ZKzYm9yUzFVUML2RJDD0kVc09oSDNC5HRYtH1LSC9J3zhWfT3ijFXXVhmlDdytHkw1pCmhd0Yn1k2rJHSUs2R0wKL0vavBraojPUtErYk1J5mY5L3W0KWXEx0n5NhlhPV2VXYzJuY0FraBDU3hqeLzQJUVSSjkycn10R3nV1jMgo1TC9w
  bmRpJXUJRU5Etdcg54UJhLteN5STfXSwBx0UzWdh1Psek3MYTZEeuH02VtYxewLyt6Rd0c35h6aHRt2VNCUm0mRqpaEJkRcvx3pUnklywJmZ1gbmJkc0uyjVJQJZXSxbtqTfUvQytrUv20rVgvcmveIxbaBhShkZ3VbzFrdhN1D80
  c0tLS0tLS0tLSEtNfUtrRk10dQvRFLs0tLS0k
  server: https://172.31.40.131:6443
  name: kubernetes
contexts:
- context:
  cluster: kubernetes
  user: kubernetes-admin
  name: kubernetes-admin@kubernetes
current-context: kubernetes-admin@kubernetes
kind: Config
preferences: {}
users:
- name: kubernetes-admin
  user:
    client-certificate-data: LS0tLS1CRUJDTlBDRVJSUZJ00FURS0tLS0tCk1JSURJVEND0Wdzt0F3SUJBz0Lj0i9gYl12bwU0Vgd3RFZSktvklodmN00VFTEJR0xGVEVUTUJFR0ExVUJK0XhNS2EzVm1awEp1wlhsbGn60WVG
  dzB5TxBpNU1Ez3doakzTVRSYUZMph0ekETURVd0qTXlNWFjhTU1VeAp+kF5m0d0VkbTlVRDbsXoxw1hNeekfTv1jBaWEl132dFa1U1BMeDU3FHb01LM0RRR0JU8VVR0TR0K30XhZ0VlClkVs1JBUIN3THhcbG02T3hk1T3RjVUz7Vdcyrt1
  ZDj6R1Vjbp01YiN00982Mm0V0TE1Jb2ykpdyt0vYso2X203Lw082Yj1wHfFWkcd012G0mfCMh05b1l05zfscF1Bd2pLcJRCtU13dGoycW0b1dNtTh35fPMsD1TkwpsR0xGr1TwpXHeHb02c8zUFNNT3NFtKwdhNkY2Lzd2wQmUSehx
  K0R2zuJb010YiN00982Mm0V0TE1Jb2ykpdyt0vYso2X203Lw082Yj1wHfFWkcd012G0mfCMh05b1l05zfscF1Bd2pLcJRCtU13dGoycW0b1dNtTh35fPMsD1TkwpsR0xGr1TwpXHeHb02c8zUFNNT3NFtKwdhNkY2Lzd2wQmUSehx
  bmxtuEs93a1B1VXB0menczj1BbpoY24d4TNC0nLxEhloXojoYj0iRmTSVHTYsUz0qMh0N0kfBRp2XvEYTUE0R0exWMrEd0VCL3dRRUf3SUmUREFOCkjnT1ZUk1C0WY4R0JU0URBuIjgjTU1iR0ExWREZfX0k1TjkahWRCvnt2ZrLcJvceF9avdQ
  cWdd1JNkR1Nq0pVYK0md0VksnRUeakT2ndwmCmxjg1bVtVsLd01Vb0TR1j0kFkR0q0wvmsnfUEN6Wvpz03c131UU0xRvhsGxZtEJrS+TSGFgYmR0yGd0UTR1k55a9q1eYhRfCvVEnLvk84a1Vnfed
  ZKzYm9yUzFVUML2RJDD0kVc09oSDNC5HRYtH1LSC9J3zhWfT3ijFXXVhmlDdytHkw1pCmhd0Yn1k2rJHSUs2R0wKL0vavBraojPUtErYk1J5mY5L3W0KWXEx0n5NhlhPV2VXYzJuY0FraBDU3hqeLzQJUVSSjkycn10R3nV1jMgo1TC9w
  bmRpJXUJRU5Etdcg54UJhLteN5STfXSwBx0UzWdh1Psek3MYTZEeuH02VtYxewLyt6Rd0c35h6aHRt2VNCUm0mRqpaEJkRcvx3pUnklywJmZ1gbmJkc0uyjVJQJZXSxbtqTfUvQytrUv20rVgvcmveIxbaBhShkZ3VbzFrdhN1D80
  c0tLS0tLS0tLSEtNfUtrRk10dQvRFLs0tLS0k
  server: https://172.31.40.131:6443
  name: kubernetes
contexts:
- context:
  cluster: kubernetes
  user: kubernetes-admin
  name: kubernetes-admin@kubernetes
current-context: kubernetes-admin@kubernetes
kind: Config
preferences: {}
users:
- name: kubernetes-admin
  user:
    client-certificate-data: LS0tLS1CRUJDTlBDRVJSUZJ00FURS0tLS0tCk1JSURJVEND0Wdzt0F3SUJBz0Lj0i9gYl12bwU0Vgd3RFZSktvklodmN00VFTEJR0xGVEVUTUJFR0ExVUJK0XhNS2EzVm1awEp1wlhsbGn60WVG
  dzB5TxBpNU1Ez3doakzTVRSYUZMph0ekETURVd0qTXlNWFjhTU1VeAp+kF5m0d0VkbTlVRDbsXoxw1hNeekfTv1jBaWEl132dFa1U1BMeDU3FHb01LM0RRR0JU8VVR0TR0K30XhZ0VlClkVs1JBUIN3THhcbG02T3hk1T3RjVUz7Vdcyrt1
  ZDj6R1Vjbp01YiN00982Mm0V0TE1Jb2ykpdyt0vYso2X203Lw082Yj1wHfFWkcd012G0mfCMh05b1l05zfscF1Bd2pLcJRCtU13dGoycW0b1dNtTh35fPMsD1TkwpsR0xGr1TwpXHeHb02c8zUFNNT3NFtKwdhNkY2Lzd2wQmUSehx
  K0R2zuJb010YiN00982Mm0V0TE1Jb2ykpdyt0vYso2X203Lw082Yj1wHfFWkcd012G0mfCMh05b1l05zfscF1Bd2pLcJRCtU13dGoycW0b1dNtTh35fPMsD1TkwpsR0xGr1TwpXHeHb02c8zUFNNT3NFtKwdhNkY2Lzd2wQmUSehx
  bmxtuEs93a1B1VXB0menczj1BbpoY24d4TNC0nLxEhloXojoYj0iRmTSVHTYsUz0qMh0N0kfBRp2XvEYTUE0R0exWMrEd0VCL3dRRUf3SUmUREFOCkjnT1ZUk1C0WY4R0JU0URBuIjgjTU1iR0ExWREZfX0k1TjkahWRCvnt2ZrLcJvceF9avdQ
  cWdd1JNkR1Nq0pVYK0md0VksnRUeakT2ndwmCmxjg1bVtVsLd01Vb0TR1j0kFkR0q0wvmsnfUEN6Wvpz03c131UU0xRvhsGxZtEJrS+TSGFgYmR0yGd0UTR1k55a9q1eYhRfCvVEnLvk84a1Vnfed
  ZKzYm9yUzFVUML2RJDD0kVc09oSDNC5HRYtH1LSC9J3zhWfT3ijFXXVhmlDdytHkw1pCmhd0Yn1k2rJHSUs2R0wKL0vavBraojPUtErYk1J5mY5L3W0KWXEx0n5NhlhPV2VXYzJuY0FraBDU3hqeLzQJUVSSjkycn10R3nV1jMgo1TC9w
  bmRpJXUJRU5Etdcg54UJhLteN5STfXSwBx0UzWdh1Psek3MYTZEeuH02VtYxewLyt6Rd0c35h6aHRt2VNCUm0mRqpaEJkRcvx3pUnklywJmZ1gbmJkc0uyjVJQJZXSxbtqTfUvQytrUv20rVgvcmveIxbaBhShkZ3VbzFrdhN1D80
  c0tLS0tLS0tLSEtNfUtrRk10dQvRFLs0tLS0k
  server: https://172.31.40.131:6443
  name: kubernetes
contexts:
- context:
  cluster: kubernetes
  user: kubernetes-admin
  name: kubernetes-admin@kubernetes
current-context: kubernetes-admin@kubernetes
kind: Config
preferences: {}
users:
- name: kubernetes-admin
  user:
    client-certificate-data: LS0tLS1CRUJDTlBDRVJSUZJ00FURS0tLS0tCk1JSURJVEND0Wdzt0F3SUJBz0Lj0i9gYl12bwU0Vgd3RFZSktvklodmN00VFTEJR0xGVEVUTUJFR0ExVUJK0XhNS2EzVm1awEp1wlhsbGn60WVG
  dzB5TxBpNU1Ez3doakzTVRSYUZMph0ekETURVd0qTXlNWFjhTU1VeAp+kF5m0d0VkbTlVRDbsXoxw1hNeekfTv1jBaWEl132dFa1U1BMeDU3FHb01LM0RRR0JU8VVR0TR0K30XhZ0VlClkVs1JBUIN3THhcbG02T3hk1T3RjVUz7Vdcyrt1
  ZDj6R1Vjbp01YiN00982Mm0V0TE1Jb2ykpdyt0vYso2X203Lw082Yj1wHfFWkcd012G0mfCMh05b1l05zfscF1Bd2pLcJRCtU13dGoycW0b1dNtTh35fPMsD1TkwpsR0xGr1TwpXHeHb02c8zUFNNT3NFtKwdhNkY2Lzd2wQmUSehx
  K0R2zuJb010YiN00982Mm0V0TE1Jb2ykpdyt0vYso2X203Lw082Yj1wHfFWkcd012G0mfCMh05b1l05zfscF1Bd2pLcJRCtU13dGoycW0b1dNtTh35fPMsD1TkwpsR0xGr1TwpXHeHb02c8zUFNNT3NFtKwdhNkY2Lzd2wQmUSehx
  bmxtuEs93a1B1VXB0menczj1BbpoY24d4TNC0nLxEhloXojoYj0iRmTSVHTYsUz0qMh0N0kfBRp2XvEYTUE0R0exWMrEd0VCL3dRRUf3SUmUREFOCkjnT1ZUk1C0WY4R0JU0URBuIjgjTU1iR0ExWREZfX0k1TjkahWRCvnt2ZrLcJvceF9avdQ
  cWdd1JNkR1Nq0pVYK0md0VksnRUeakT2ndwmCmxjg1bVtVsLd01Vb0TR1j0kFkR0q0wvmsnfUEN6Wvpz03c131UU0xRvhsGxZtEJrS+TSGFgYmR0yGd0UTR1k55a9q1eYhRfCvVEnLvk84a1Vnfed
  ZKzYm9yUzFVUML2RJDD0kVc09oSDNC5HRYtH1LSC9J3zhWfT3ijFXXVhmlDdytHkw1pCmhd0Yn1k2rJHSUs2R0wKL0vavBraojPUtErYk1J5mY5L3W0KWXEx0n5NhlhPV2VXYzJuY0FraBDU3hqeLzQJUVSSjkycn10R3nV1jMgo1TC9w
  bmRpJXUJRU5Etdcg54UJhLteN5STfXSwBx0UzWdh1Psek3MYTZEeuH02VtYxewLyt6Rd0c35h6aHRt2VNCUm0mRqpaEJkRcvx3pUnklywJmZ1gbmJkc0uyjVJQJZXSxbtqTfUvQytrUv20rVgvcmveIxbaBhShkZ3VbzFrdhN1D80
  c0tLS0tLS0tLSEtNfUtrRk10dQvRFLs0tLS0k
  server: https://172.31.40.131:6443
  name: kubernetes
contexts:
- context:
  cluster: kubernetes
  user: kubernetes-admin
  name: kubernetes-admin@kubernetes
current-context: kubernetes-admin@kubernetes
kind: Config
preferences: {}
users:
- name: kubernetes-admin
  user:
    client-certificate-data: LS0tLS1CRUJDTlBDRVJSUZJ00FURS0tLS0tCk1JSURJVEND0Wdzt0F3SUJBz0Lj0i9gYl12bwU0Vgd3RFZSktvklodmN00VFTEJR0xGVEVUTUJFR0ExVUJK0XhNS2EzVm1awEp1wlhsbGn60WVG
  dzB5TxBpNU1Ez3doakzTVRSYUZMph0ekETURVd0qTXlNWFjhTU1VeAp+kF5m0d0VkbTlVRDbsXoxw1hNeekfTv1jBaWEl132dFa1U1BMeDU3FHb01LM0RRR0JU8VVR0TR0K30XhZ0VlClkVs1JBUIN3THhcbG02T3hk1T3RjVUz7Vdcyrt1
  ZDj6R1Vjbp01YiN00982Mm0V0TE1Jb2ykpdyt0vYso2X203Lw082Yj1wHfFWkcd012G0mfCMh05b1l05zfscF1Bd2pLcJRCtU13dGoycW0b1dNtTh35fPMsD1TkwpsR0xGr1TwpXHeHb02c8zUFNNT3NFtKwdhNkY2Lzd2wQmUSehx
  K0R2zuJb010YiN00982Mm0V0TE1Jb2ykpdyt0vYso2X203Lw082Yj1wHfFWkcd012G0mfCMh05b1l05zfscF1Bd2pLcJRCtU13dGoycW0b1dNtTh35fPMsD1TkwpsR0xGr1TwpXHeHb02c8zUFNNT3NFtKwdhNkY2Lzd2wQmUSehx
  bmxtuEs93a1B1VXB0menczj1BbpoY24d4TNC0nLxEhloXojoYj0iRmTSVHTYsUz0qMh0N0kfBRp2XvEYTUE0R0exWMrEd0VCL3dRRUf3SUmUREFOCkjnT1ZUk1C0WY4R0JU0URBuIjgjTU1iR0ExWREZfX0k1TjkahWRCvnt2ZrLcJvceF9avdQ
  cWdd1JNkR1Nq0pVYK0md0VksnRUeakT2ndwmCmxjg1bVtVsLd01Vb0TR1j0kFkR0q0wvmsnfUEN6Wvpz03c131UU0xRvhsGxZtEJrS+TSGFgYmR0yGd0UTR1k55a9q1eYhRfCvVEnLvk84a1Vnfed
  ZKzYm9yUzFVUML2RJDD0kVc09oSDNC5HRYtH1LSC9J3zhWfT3ijFXXVhmlDdytHkw1pCmhd0Yn1k2rJHSUs2R0wKL0vavBraojPUtErYk1J5mY5L3W0KWXEx0n5NhlhPV2VXYzJuY0FraBDU3hqeLzQJUVSSjkycn10R3nV1jMgo1TC9w
  bmRpJXUJRU5Etdcg54UJhLteN5STfXSwBx0UzWdh1Psek3MYTZEeuH02VtYxewLyt6Rd0c35h6aHRt2VNCUm0mRqpaEJkRcvx3pUnklywJmZ1gbmJkc0uyjVJQJZXSxbtqTfUvQytrUv20rVgvcmveIxbaBhShkZ3VbzFrdhN1D80
  c0tLS0tLS0tLSEtN
```

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:
                    sh 'kubectl apply -f deployment.yml'
                    sh 'kubectl apply -f service.yml'
                }
            }
        }
    }
}
```



Give the command after pipeline success

```
kubectl get all
```



Add Load balancer IP address to cluster ec2 instance security group and copy load balancer Link and open in a browser

You will see output like this.



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

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



Cluster deleted

Cluster name	Status	Kubernetes version	Provider
No clusters You do not have any clusters. Create cluster			

Delete the Ec2 instance & IAM role.

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

Congratulations on completing the journey of deploying your 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.

What's Next? Continue exploring the realm of DevSecOps and cloud technologies. Experiment with new tools, refine your deployment pipelines, and delve deeper into securing your applications effectively.

Stay Connected: We hope this guide has been insightful and valuable for your learning journey. Don't hesitate to reach out with questions, feedback, or requests for further topics. Follow us on [Your Social Media Handles] for more tech tutorials, guides, and updates.

Thank you for embarking on this DevSecOps journey with us. Keep innovating, securing, and deploying your applications with confidence!

[ansible](#)[Aws](#)[ci/cd](#)[devops](#)[DevSecOps](#)[Docker](#)[Git](#)[GitHubActions](#)[GitLab](#)[Jenkins](#)[kuberenetes](#)[snyk](#)[Terraform](#)[Trivy](#)

Ajay Kumar Yegireddi is a DevSecOps Engineer and System Administrator, with a passion for sharing real-world DevSecOps projects and tasks. **Mr. Cloud Book**, provides hands-on tutorials and practical insights to help others master DevSecOps tools and workflows. Content is designed to bridge the gap between development, security, and operations, making complex concepts easy to understand for both beginners and professionals.

Comments

24 responses to “DevSecOps CI/CD : Deploying a Secure Hotstar Clone (Even if You're Not a Pro)”

**Abdul Salam**

6 January 2024

Thank You for Creating Such valuable content project it really helps us to upgrade our skills and Thank you once again Mr. Cloud Book and keep doing it its our humble request from us.

[Reply](#)**mrcloudbook.com**

7 January 2024

Thanks bro
will do

[Reply](#)**Rajesh**

7 January 2024

Thank you Ajay for explaining such wonderfull ,in detail manner, the way you have presented is just awesome, keep up good work . I love to see you contributing so much to the community .

[Reply](#)**mrcloudbook.com**

7 January 2024

Thanks for support brother

[Reply](#)**Srinivas**

8 January 2024

Thanks Ajay. Wonderful demonstration. I have also implemented this project and learnt lot of things. If possible please do videos on GCP DevOps too. Thanks!

[Reply](#)

**mrcloudbook.com**

8 January 2024

will try bro

[Reply](#)**Jay**

9 January 2024

You're the EYE. Great job brother!

[Reply](#)**mrcloudbook.com**

9 January 2024

Thanks brother

[Reply](#)**Hieu Luong**

13 January 2024

Thanks for your great share. Keep your journey, bro 😊

[Reply](#)**mrcloudbook.com**

14 January 2024

definitely bro

[Reply](#)

**Shivasantosh**

14 January 2024

This is very much helpful... Thank you very much...Please provide details of different projects also.. Advance happy Sankranti to everyone...

[Reply](#)**mrcloudbook.com**

14 January 2024

okay bro happy sankranthi

[Reply](#)**Nagaraj**

14 January 2024

Hi Bro,

Fantastic information and explaination,

Regards,
Nagaraj CH

[Reply](#)**mrcloudbook.com**

14 January 2024

Thanks bro

[Reply](#)

**Ved**

11 February 2024

```
sh 'terraform ${action} -auto-approve'
```

i think this command should be

```
sh 'terraform apply ${action} -auto-approve'
```

Awesome Work Bro,Keep Going your journey.

[Reply](#)**Ved**

11 February 2024

```
sh 'terraform ${action} -auto-approve'
```

i think this command should be

```
sh 'terraform apply ${action} -auto-approve'
```

Awesome Work Bro,Keep goin your journey.

[Reply](#)**mrcloudbook.com**

12 February 2024

It will automcatically take from build
no need for apply in command

[Reply](#)**Alfiya**

21 February 2024

Nice!

[Reply](#)

mrcloudbook.com

6 March 2024

Thanks

[Reply](#)

Pardeep

13 March 2024

Thank you Ajay. i have completed the entire project after no. of tries. but in the end still after applying kubernetes stage. i got some error. in last stage what will be the serverurl? i added. stage('Deploy to kubernets') {

```
steps{
script{
dir('K8S') {
withKubeConfig(caCertificate: "", clusterName: 'EKS_Cloud', contextName: "",
credentialsId: 'K8s', namespace: 'default', restrictKubeConfigAccess: false, serverUrl:
'https://5A0A1F4D004A601558FEE65B5F7CF0F0.gr7.ca-central-
1.eks.amazonaws.com') {
sh 'kubectl apply -f deployment.yml --validate=false'
sh 'kubectl apply -f service.yml --validate=false'
}
}
}
}
}
}
```

i got error :

```
+ kubectl apply -f deployment.yml
error: error validating "deployment.yml": error validating data: failed to download
openapi: Get "https://2D9148EDEB83A1C2147C731BBA6D7104.gr7.ca-central-
1.eks.amazonaws.com/openapi/v2?timeout=32s": dial tcp: lookup
2D9148EDEB83A1C2147C731BBA6D7104.gr7.ca-central-1.eks.amazonaws.com on
127.0.0.53:53: no such host; if you choose to ignore these errors, turn validation off
with --validate=false
[Pipeline]
[kubernetes-cli] kubectl configuration cleaned up
[Pipeline] // withKubeConfig
```

```
[Pipeline] }
[Pipeline] // dir
[Pipeline] }
[Pipeline] // script
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
ERROR: script returned exit code 1
Finished: FAILURE
it shows this error
```

later on anyhow i overcome this error but when i deploy on kuberentes the loadbalancer ip address doesnot work.

[Reply](#)



Saketh

17 March 2024

Hi mr cloud book , can YOU suggest me some projects to keep in my resume

[Reply](#)



mrcloudbbook.com

17 March 2024

Do your own projects with the above process Brother
everyone uses same projects

[Reply](#)

**Umer Asif**

5 February 2025

Hi Bro,

I want to do these projects. However , I was curious if creating Ec2 etc would cost me money or it can be done in free aws account?

[Reply](#)

**36.01hw34k15gesbqggjtzbn6zd79@mail4u.life**

12 May 2024

velit error pariatur placeat natus. veritatis assumenda laborum quasi incidunt esse quia aut facilis. aut maxime ipsa a placeat quasi corporis. qui magnam saepe est id aliquam ipsa quis in deserunt la

[Reply](#)

Leave a Reply

Your email address will not be published. Required fields are marked *

Comment *

Name *

Email *

Website

Save my name, email, and website in this browser for the next time I comment.



I'm not a robot

reCAPTCHA
Privacy - Terms

Post Comment

Uncategorized

How to Automate Incident Response : How Q Developer Helped Me Automate a Daily Pain Point

22 July 2025

AI

How to Run Docker Model Runner on Ubuntu 24.04

11 July 2025

AI, DevOps

How to Install docker-ai on Ubuntu 24.04

15 June 2025

Upskill with Ajay: DevSecOps Mastery

Join Mr Cloud book to master DevSecOps through real-world projects. Learn CI/CD, security integration, automation, and more, gaining hands-on skills for industry-level challenges.



Important Links

[Privacy Policy](#)

[Terms & Conditions](#)

[Contact](#)

Resources

[Blog](#)

[YouTube Channel](#)

© 2024 · Powered by [Mr Cloud Book](#)

[Follow Us on YouTube](#)