

Mr Cloud Book

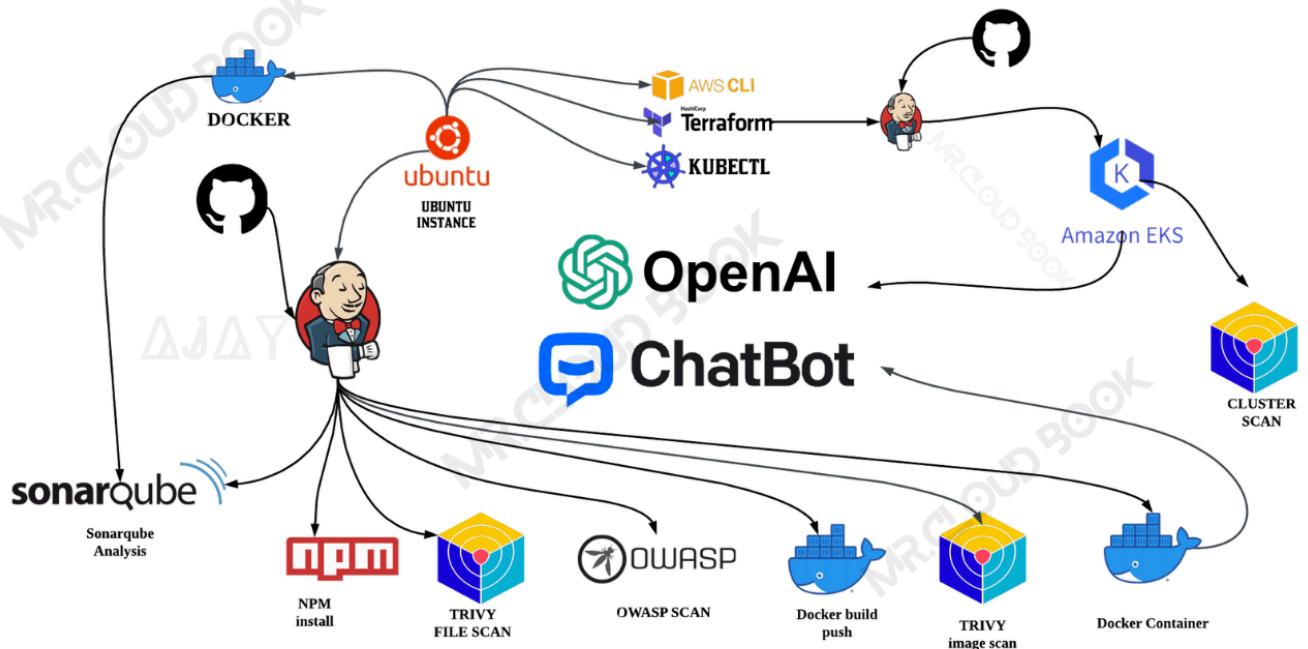
Home Blog DevSecOps Contact About Me Testimonials

Search Blogs

OpenAI Chatbot UI Deployment | DevSecOps



mrcloudbook.com · 6 March 2024



In today's digital landscape, user engagement is key to the success of any application. From websites to mobile apps, providing users with interactive and personalized experiences is essential. That's why we're thrilled to announce our latest endeavor:

What is ChatBOT?

ChatBOT is an AI language model trained on vast amounts of human conversation data. It's capable of generating human-like text responses based on the input it receives. From answering questions to engaging in conversation, ChatBOT can simulate natural language interactions with users, making it an invaluable tool for enhancing user engagement.

Why ChatBOT?

1. Personalized Interactions: ChatBOT can understand and respond to user queries in a natural and conversational manner, creating personalized interactions that keep users engaged.
2. 24/7 Availability: Unlike human agents, ChatBOT is available 24/7 to assist users, providing instant responses to their queries and ensuring a seamless user experience.
3. Scalability: With ChatBOT deployed in our application, we can handle a large volume of user interactions without compromising performance, ensuring scalability as our user base grows.

How We're Deploying ChatBOT

To deploy ChatBOT on our EKS, we're leveraging Jenkins as our CICD (Continuous Integration/Continuous Deployment) tool and deploying the chatbot within a Docker container. This approach allows us to automate the deployment process, ensuring efficient and reliable delivery of updates and enhancements to our users.

[CLICK HERE FOR GITHUB REPO](#)

**[HERE IS MY REPO WITH TERRAFORM FILES IN LEGACY BRANCH
EKS_TERRAFORM](#)**

Special thanks to [McKay Wrigley](#), the owner of Open Source, for his dedication to fostering innovation and collaboration in the open-source community.

We are grateful for his contributions to the development of technology and for making projects like ChatBOT UI possible.

Now, let's get started and dig deeper into each of these steps:-

Contents [hide]

[Launch an Ubuntu\(22.04\) T2 Large Instance](#)

[Create IAM role](#)

[Install Jenkins, Docker and Trivy](#)

Launch an Ubuntu(22.04) T2 Large Instance

Launch an AWS T2 Large Instance. Use the image as Ubuntu. You can create a new key pair or use an existing one. Enable HTTP and HTTPS settings in the Security Group and open all ports (not best case to open all ports but just for learning purposes it's okay).

The screenshot shows the AWS EC2 Instances page. At the top, there is a search bar with placeholder text "Find instance by attribute or tag (case-sensitive)". Below the search bar is a table header with columns: Name, Instance ID, Instance state, Instance type, Status check, Alarm status, Availability Zone, and Public IPv4 D. A single instance is listed: "CI-CD" with Instance ID "i-065c10200537a1eee", Instance state "Running", Instance type "t2.large", Status check "2/2 checks passed", Alarm status "No alarms", Availability Zone "ap-south-1a", and Public IPv4 "ec2-52-66-14".

Create IAM role

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

The screenshot shows the AWS search interface with "IAM" selected in the search bar. The search results for "IAM" are displayed, showing the "IAM" service card which is highlighted with a red box. The "Roles" tab under the IAM service card is also highlighted with a red box. Other services like "Amazon Redshift" and "AWS Fargate" are visible on the right side.

Click on Create Role

The screenshot shows the AWS IAM Roles page. The "Roles" tab is selected in the sidebar, and the main table shows two existing roles: "AWSServiceRoleForSupport" and "AWSServiceRoleForTrustedAdvisor". A red arrow points from the "Roles" tab in the sidebar to the "Create role" button at the top right of the main table.

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.
- Web identity**
Allows users federated by the specified external web identity provider to assume this role 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,you can use any name for 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      {
11        "Service": [
12          "ec2.amazonaws.com"
13        ]
14      }
15    ]
16  }

```

Role is created.

Identity and Access Management (IAM)

Roles (3) Info

An IAM role is an identity you can create that has specific permissions with credentials that are valid for short durations. Roles can be assumed by entities that you trust.

Role name	Trusted entities	Last activity
AWSServiceRoleForSupport	AWS Service: support (Service-Linker)	-
AWSServiceRoleForTrustedAdvisor	AWS Service: trustedadvisor (Service-Linker)	-
MARIO	AWS Service: ec2	-

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.

Instances (1/1) Info

Find Instance by attribute or tag (case-sensitive)

Instance state: running

Actions ▾

- Connect
- View details
- Manage instance state
- Instance settings
- Networking
- Security
- Image and templates
- Monitor and troubleshoot

Modify IAM role

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

The screenshot shows the 'Modify IAM role' page in the AWS IAM console. At the top, it says 'EC2 > Instances > i-0c92d797a5813a128 > Modify IAM role'. Below that, it says 'Modify IAM role' with an 'info' link. It says 'Attach an IAM role to your instance.' Under 'Instance ID', it shows 'i-0c92d797a5813a128 (Test)'. Under 'IAM role', it says 'Select an IAM role to attach to your instance or create a new role if you haven't created any. The role you select replaces any roles that are currently attached to your instance.' A dropdown menu is open, showing 'MARIO' with a red box around it. To the right of the dropdown are two buttons: 'Create new IAM role' and 'Update IAM role' (which has a blue box around it). At the bottom are 'Cancel' and 'Update IAM role' buttons.

Connect the instance to Mobaxtreme or Putty

Install Jenkins, Docker and Trivy

To Install Jenkins

Connect to your console, and enter these commands to Install Jenkins

vi jenkins.sh



```
#!/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.adoptium.net/artifactory/api/debian stable main" | sudo tee /etc/apt/sources.list.d/adoptium.list
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 | sudo tee /usr/share/keyrings/jenkins-keyring.asc &> /dev/null
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 -y
```

```
sudo apt-get install jenkins -y  
sudo systemctl start jenkins
```



```
sudo chmod 777 jenkins.sh  
sudo su    #move into root and run  
.jenkins.sh    # this will install jenkins
```



Once Jenkins is installed, you will need to go to your AWS EC2 Security Group and open Inbound Port 8080, since Jenkins works on Port 8080.

Nov



D U P I D A M



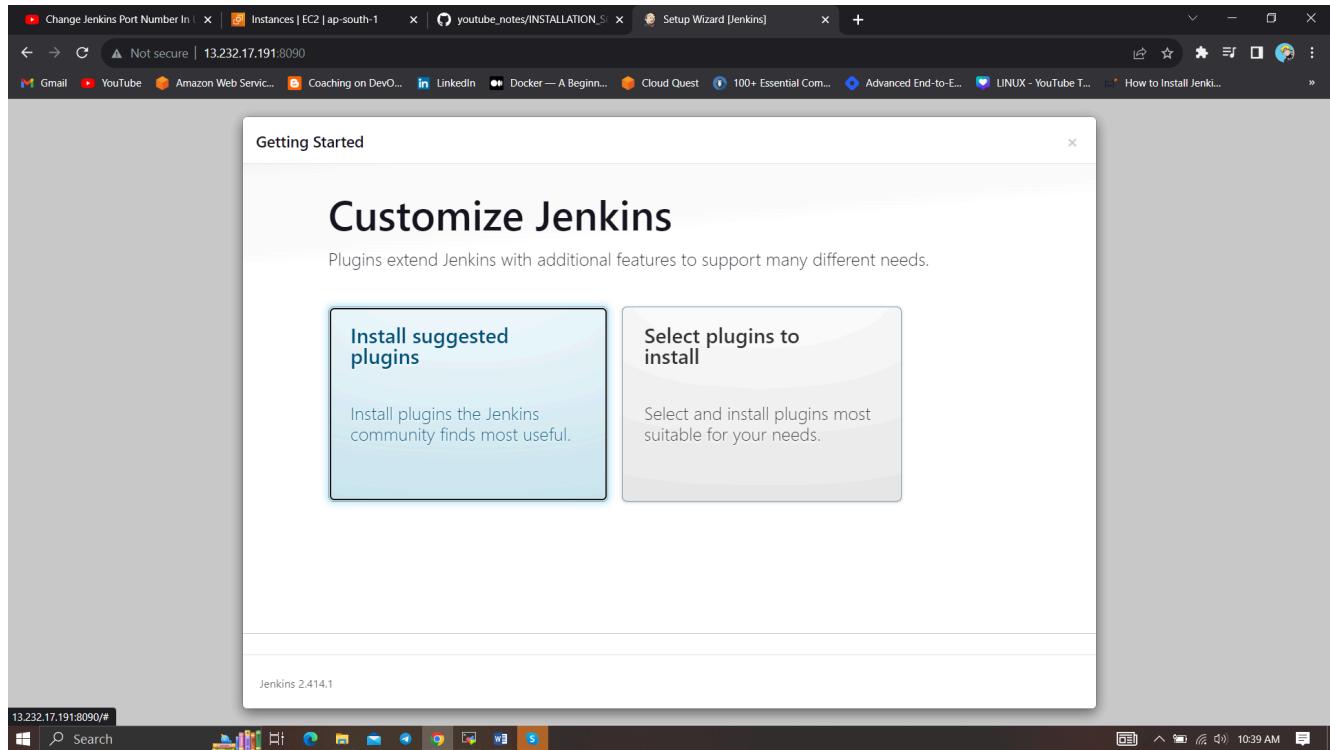
EC2 Public IP Address:8080&

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

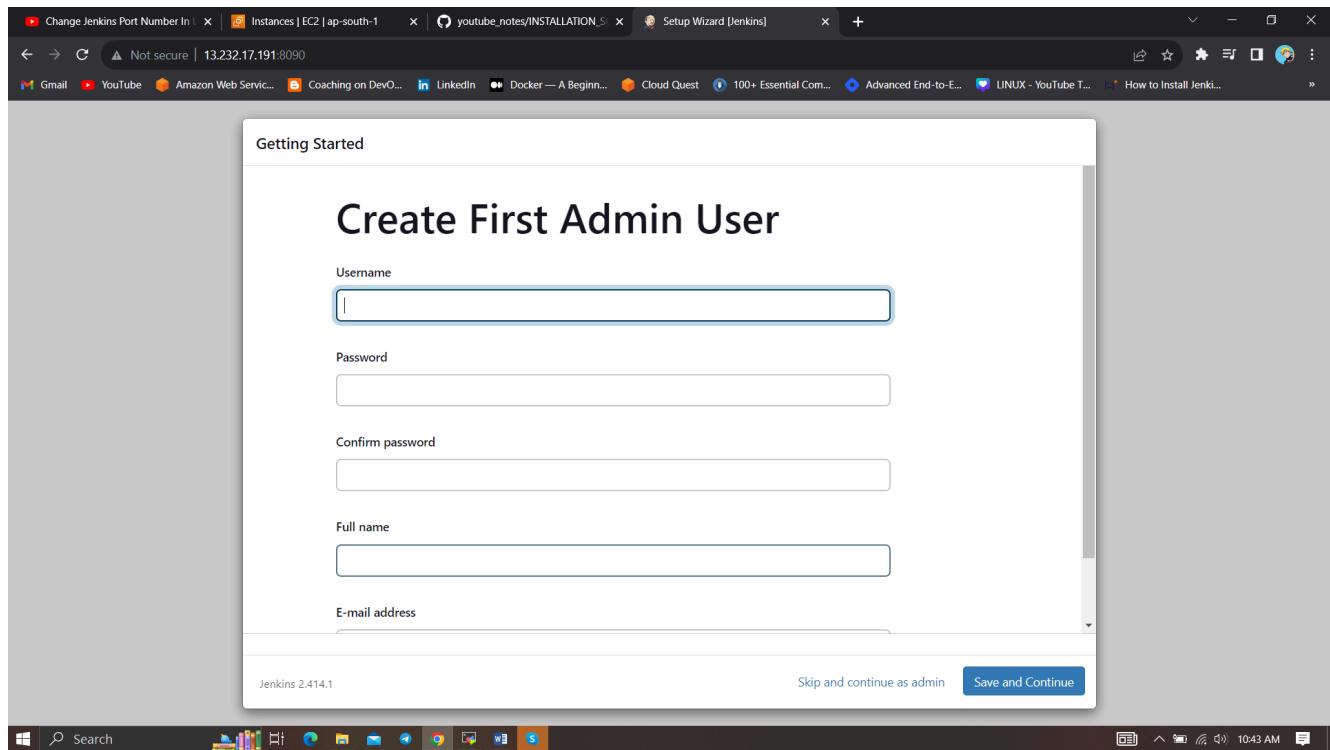


The screenshot shows a web browser window with the URL <http://13.232.17.191:8090/login?from=%2F>. The page title is "Getting Started" and the main heading is "Unlock Jenkins". The text on the page reads: "To ensure Jenkins is securely set up by the administrator, a password has been written to the log ([not sure where to find it?](#)) and this file on the server: `/var/lib/jenkins/secrets/initialAdminPassword`". Below this, it says "Please copy the password from either location and paste it below." There is an input field labeled "Administrator password" with a placeholder "Paste password here". At the bottom right of the form is a blue "Continue" button.

Unlock Jenkins using an administrative password and install the suggested plugins.



Jenkins will now get installed and install all the libraries.



Create a user click on save and continue.

Jenkins Getting Started Screen.

The screenshot shows the Jenkins dashboard. At the top, there's a navigation bar with various links like 'Change Jenkins Port Number In', 'Instances | EC2 | ap-south-1', 'youtube.notes/INSTALLATION_S...', 'Dashboard [Jenkins]', 'SonarQube', and a search bar. Below the navigation is the Jenkins logo and a 'Dashboard' link. On the left, there's a sidebar with links for 'New Item', 'People', 'Build History', 'Manage Jenkins', and 'My Views'. The main content area has a heading 'Welcome to Jenkins!'. It says, '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.' Below this, there's a 'Start building your software project' section with a 'Create a job' button and a 'Set up a distributed build' section with 'Set up an agent' and 'Configure a cloud' buttons, along with a 'Learn more about distributed builds' link.

Install Docker

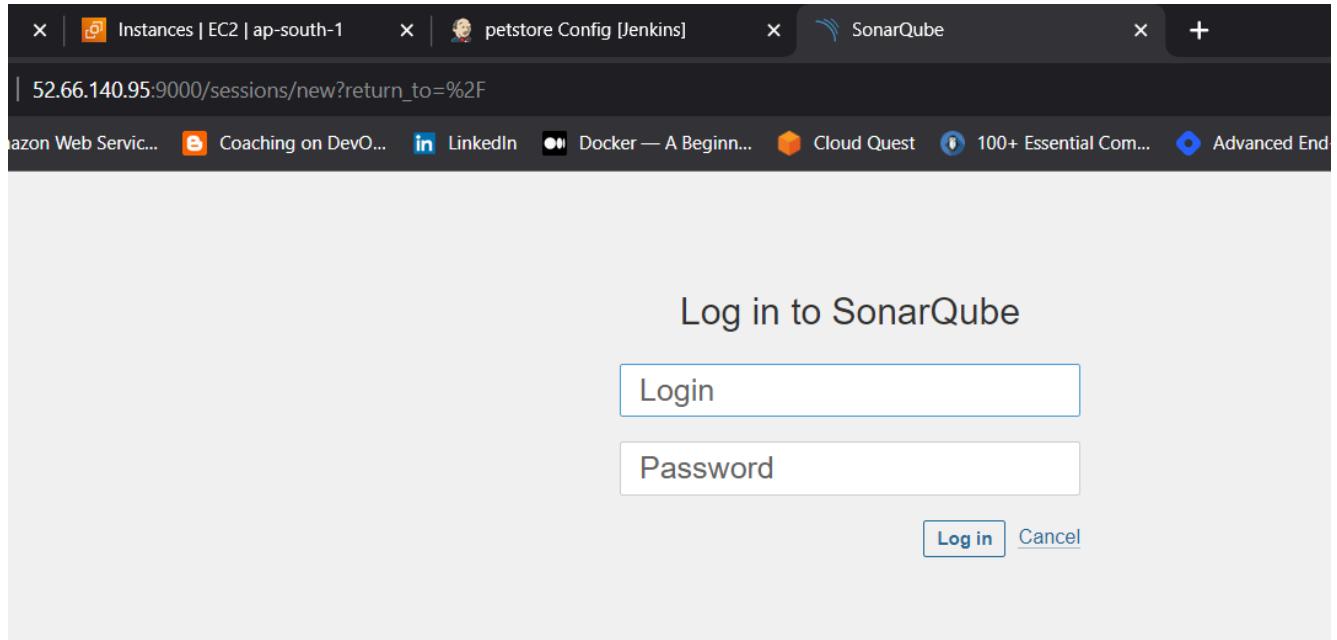
```
sudo apt-get update
sudo apt-get install docker.io -y
sudo usermod -aG docker $USER      #my case is ubuntu
newgrp docker
sudo chmod 777 /var/run/docker.sock
```

After the docker installation, we create a sonarqube container (Remember to add 9000 ports in the security group).

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

```
ubuntu@ip-172-31-42-253:~$ sudo chmod 777 /var/run/docker.sock
ubuntu@ip-172-31-42-253:~$ docker run -d --name sonar 9000:9000 sonarqube:lts-community
Unable to find image 'sonarqube:lts-community' locally
lts-community: Pulling from library/sonarqube
44ba2882fb8e: Pull complete
2cabec57fa36: Pull complete
c20481384b6a: Pull complete
bf7b17ee74f8: Pull complete
38617faac714: Pull complete
706f20f58f5e: Pull complete
65a29568c257: Pull complete
Digest: sha256:1a118f8ab960d6c3d4ea8b4455a5a6560654511c88a6816f1603f764d5dcc77c
Status: Downloaded newer image for sonarqube:lts-community
4b66c96bf9ad3d62289436af7f752fd804993892d0ca5065e2f2e32301b50139
ubuntu@ip-172-31-42-253:~$ docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
4b66c96bf9ad sonarqube:lts-community "/opt/sonarqube/dock..." 9 seconds ago Up 5 seconds 0.0.0.0:9000->9000/tcp, :::9000->9000/tcp sonar
ubuntu@ip-172-31-42-253:~$
```

Now our Sonarqube is up and running



Enter username and password, click on login and change password

```
username admin
password admin
```

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

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.

← → C 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 Set up global configuration	From Bitbucket Server Set up global configuration	From Bitbucket Cloud Set up global configuration	From GitHub Set up global configuration	From GitLab Set up global configuration
--	--	---	--	--

Install Trivy, Kubectl,Terraform



```
vi script.sh
```



```
sudo apt-get install wget apt-transport-https gnupg lsb-release -y
wget -qO - https://aquasecurity.github.io/trivy-repo/deb/public.key | gpg --dearmor
echo "deb [signed-by=/usr/share/keyrings/trivy.gpg] https://aquasecurity
sudo apt-get update
sudo apt-get install trivy -y

# 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
sudo apt update
sudo apt install curl -y
curl -LO https://dl.k8s.io/release/$(curl -L -s https://dl.k8s.io/releas
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 "awscl
sudo apt-get install unzip -y
unzip awscliv2.zip
sudo ./aws/install
```



Give permissions and run script



```
sudo chmod 777 script.sh
./script.sh
```



Next, we will log in to Jenkins and start to configure our Pipeline in Jenkins

Install Plugins like JDK, Sonarqube Scanner, NodeJs, OWASP Dependency Check

Install Plugin

Goto Manage Jenkins → Plugins → Available Plugins →

Install below plugins

Blue ocean

1 → Eclipse Temurin Installer

2 → SonarQube Scanner

3 → NodeJs Plugin

4 → Docker

5 → Docker commons

6 → Docker pipeline

7 → Docker API

8 → Docker Build step

9 → Owasp Dependency Check

10 → Kubernetes

11 → Kubernetes CLI

12 → Kubernetes Client API

13 → Kubernetes Pipeline DevOps steps

Search: kubernetes

Install

<input checked="" type="checkbox"/> Kubernetes Credentials 0.11	/	2 mo 12 days ago
kubernetes credentials		
Common classes for Kubernetes credentials		
<input checked="" type="checkbox"/> Kubernetes Client API 6.8.1-224.vd388fca_4db_3b_	/	2 mo 12 days ago
kubernetes Library plugins (for use by other plugins)		
Kubernetes Client API plugin for use by other Jenkins plugins.		
<input checked="" type="checkbox"/> Kubernetes 4054.v2da_8e2794884	/	1 mo 0 days ago
Cloud Providers Cluster Management kubernetes Agent Management		
This plugin integrates Jenkins with Kubernetes		
<input checked="" type="checkbox"/> Kubernetes CLI 1.12.1	/	2 mo 11 days ago
kubernetes		
Configure kubectl for Kubernetes		
<input checked="" type="checkbox"/> Kubernetes Credentials Provider 1.258.v95949f923a_a_e	/	22 days ago
kubernetes credentials		
Provides a read only credentials store backed by Kubernetes.		
<input checked="" type="checkbox"/> Kubernetes :: Pipeline :: DevOps Steps 1.6	/	4 yr 9 mo ago
pipeline kubernetes		

<input checked="" type="checkbox"/> Eclipse Temurin installer 1.5	Provides an installer for the JDK tool that downloads the JDK from https://adoptium.net	1 yr 1 mo ago
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"/> SonarQube Scanner 2.16.1	External Site/Tool Integrations Build Reports	1 mo 0 days ago
This plugin allows an easy integration of SonarQube , the open source platform for Continuous Inspection of code quality.		
<input checked="" type="checkbox"/> NodeJS 1.6.1	/	2 mo 25 days ago
npm		
NodeJS Plugin executes NodeJS script as a build step.		
<input checked="" type="checkbox"/> Docker 1.5	/	2 mo 6 days ago
Cloud Providers Cluster Management docker		
This plugin integrates Jenkins with Docker		

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.

NodeJS 1.6.1

[npm](#)

NodeJS Plugin executes [NodeJS](#) script as a build step.

Docker 1.5

[Cloud Providers](#) [Cluster Management](#) [docker](#)

Mr cloud book

This plugin integrates Jenkins with [Docker](#).

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.

Docker Pipeline 572.v950f58993843

[pipeline](#) [DevOps](#) [Deployment](#) [docker](#)

Build and use Docker containers from pipelines.

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.

docker-build-step 2.10

[Build Tools](#) [docker](#)

Mr cloud book

This plugin allows to add various docker commands to your job as build steps.

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.

Terraform 1.0.10

[Build Wrappers](#)

Mr cloud book

This plugin provides a build wrapper for [Terraform](#) to launch and destroy infrastructure.

Configure Java and Nodejs in Global Tool Configuration

Goto Manage Jenkins → Tools → Install JDK(17) and NodeJs(19)→ Click on Apply and Save

JDK installations

Add JDK

JDK

Name: X

Install automatically ?

Install from adoptium.net ?

Version: X

MR CLOUD BOOK

Add Installer

Add JDK

NodeJS installations

Add NodeJS

Name

Install automatically ?

Install from nodejs.org

Version

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

Force 32bit architecture

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

Groups
Global Permissions
Permission Templates

click on update Token

SCM Accounts	Last connection	Groups	Tokens
	< 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

! 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

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

Create

You will see this page once you click on create

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

ID	Name	Kind	Description	Action
	Sonar-token	sonar	Secret text	

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

Name: sonar-server Value: Mr cloud book

Server URL: Default is http://localhost:9000
http://13.232.165.223:9000

Server authentication token: Sonat-token

Add

Click on Apply and Save

The **Configure System** option is used in Jenkins to configure different server

Global Tool Configuration is used to configure different tools that we install using Plugins

We will install a sonar scanner in the tools.

Manage Jenkins -> Tools -> SonarQube Scanner

SonarQube Scanner

Name: sonar-scanner
Mr cloud book

Install automatically

Install from Maven Central

Version: SonarQube Scanner 5.0.1.3006

Add Installer

In the Sonarqube Dashboard add a quality gate also

Administration-> Configuration->Webhooks

Administration

Administration

General Settings
Encryption
Webhooks

Search by login or name...

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

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/

Server endpoint that will receive the webhook payload, for example:
"http://my_server/foo". If HTTP Basic authentication is used, HTTPS is recommended to avoid man in the middle attacks. Example:
"https://my.Login.myPassword@my_serverfoo"

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**

Get the most out of SonarQube!

Take advantage of the whole ecosystem by using SonarLint, a free IDE plugin that helps you find and fix issues earlier in your workflow

Learn More **Dismiss**

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

First, we configured the Plugin and next, we had to configure the Tool

Goto Dashboard → Manage Jenkins → Tools →

Dependency-Check installations

Add Dependency-Check

Dependency-Check

Name: DP-Check

Install automatically [?](#) **Mr cloud book**

Install from github.com

Version: dependency-check 9.0.9

Add Installer [▼](#)

Click on Apply and Save here.

Now, goto Dashboard → Manage Jenkins → Tools →

Docker

Name: docker **Mr cloud book**

Install automatically [?](#)

Download from docker.com

Docker version [?](#)

latest

Add Installer [▼](#)

Tools -> Terraform add this

In Jenkins



Terraform

Name: terraform **Mr cloud book**

Install directory: /usr/bin/

Install automatically [?](#)

Go to manage Jenkins -> Credentials

Add DockerHub Username and Password under Global Credentials

The screenshot shows the Jenkins Global Credentials configuration interface. The path in the top navigation bar is: Dashboard > Manage Jenkins > Credentials > System > Global credentials (unrestricted). The 'Kind' dropdown is set to 'Username with password'. The 'Scope' dropdown is set to 'Global (Jenkins, nodes, items, all child items, etc.)'. The 'Username' field contains 'sevenajay'. The 'Password' field contains a masked password. The 'ID' field contains 'docker'. The 'Description' field contains 'docker'. A blue 'Create' button is at the bottom left.

Create EKS Cluster from Jenkins

Go to Legacy Branch in Git Repository

There you can find folder called Eks-terrafrom there you can find backend file

CHANGE YOUR S3 BUCKET NAME IN THE BACKEND.TF

Now create a new job for the Eks provision

Enter an item name

Terraform-Eks
 » 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

This project is parameterized ?

Choice Parameter X

Name ?

Choices ? apply
destroy

Description ?

Plain text [Preview](#)

Save **Apply**

Let's add a pipeline



```
pipeline{
    agent any
    stages {
```

```
stage('Checkout from Git'){
    steps{
        git branch: 'legacy', url: 'https://github.com/Aj7Ay/chatbot-eks'
    }
}

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'
        }
    }
}
}
```

Pipeline Terraform-Eks

This build requires parameters:

action: **apply**

Build (highlighted with a red box)

Stage view it will take max 10mins to provision

Blue ocean output



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

EKS > Clusters

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	Put
Jenkins-ARGO	i-0323f37f837248e53	Running	t2.large	2/2 checks passed	No alarms	ap-south-1b	ec2
<input checked="" type="checkbox"/> Jenkins-ARGO	i-049634a401c64808b	Running	t2.medium	2/2 checks passed	No alarms	ap-south-1b	ec2

Create Job for chatbot clone

Add this stage to Pipeline Script



```

pipeline{
    agent any
    tools{
        jdk 'jdk17'
        nodejs 'node19'
    }
    environment {
        SCANNER_HOME=tool 'sonar-scanner'
    }
    stages {
        stage('Checkout from Git'){
            steps{
                git branch: 'legacy', url: 'https://github.com/Aj7Ay/chatbot-legacy'
            }
        }
        stage('Install Dependencies') {
            steps {
                sh "npm install"
            }
        }
        stage("Sonarqube Analysis"){
            steps{
                withSonarQubeEnv('sonar-server') {
                    sh ''' $SCANNER_HOME/bin/sonar-scanner -Dsonar.projectName=Chatbot -Dsonar.projectKey=Chatbot '''
                }
            }
        }
        stage("quality gate"){
            steps {
                script {
                    waitForQualityGate abortPipeline: false, credentialsId: 'jenkins-admin'
                }
            }
        }
        stage('OWASP FS SCAN') {
            steps {
                dependencyCheck additionalArguments: '--scan ./ --disableCheckForKnownVulnerabilities'
                dependencyCheckPublisher pattern: '**/dependency-check-report.html'
            }
        }
        stage('TRIVY FS SCAN') {
            steps {
                sh "trivy fs . > trivyfs.json"
            }
        }
    }
}

```

```

        }
    }
    stage("Docker Build & Push"){
        steps{
            script{
                withDockerRegistry(credentialsId: 'docker', toolName
                    sh "docker build -t chatbot ."
                    sh "docker tag chatbot sevenajay/chatbot:latest "
                    sh "docker push sevenajay/chatbot:latest "
                }
            }
        }
    }
    stage("TRIVY"){
        steps{
            sh "trivy image sevenajay/chatbot:latest > trivy.json"
        }
    }
    stage ("Remove container") {
        steps{
            sh "docker stop chatbot | true"
            sh "docker rm chatbot | true"
        }
    }
    stage('Deploy to container'){
        steps{
            sh 'docker run -d --name chatbot -p 3000:3000 sevenajay,
        }
    }
}

```

Apply and Save and click on Build

stage view



chatbot Passed

MR cloud book

Last analysis: 2 minutes ago

Bugs	Vulnerabilities	Hotspots Reviewed	Code Smells	Coverage	Duplications	Lines
1 D	0 A	0.0% E	36 A	0.0% E	10.2% E	5.3k S TypeScript...

You can see the report has been generated and the status shows as passed. You can see that there are 5.3k lines it scanned. To see a detailed report, you can go to issues.

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

Dependency-Check Results

SEVERITY DISTRIBUTION			
1	8	3	
File Name	Vulnerability	Severity	Weakness
axios:0.26.1	OSSINDEX CVE-2023-45857	Medium	CWE-352
follow-redirects:1.15.2	NVD CVE-2023-26159	Medium	CWE-601
get-func-name:2.0.0	OSSINDEX CVE-2023-43646	High	CWE-1333
next:13.2.4	OSSINDEX CVE-2023-46298	High	CWE-noinfo
postcss:8.4.21	NVD CVE-2023-44270	Medium	CWE-74
semver:6.3.0	OSSINDEX CVE-2022-25883	High	CWE-1333
semver:7.3.8	OSSINDEX CVE-2022-25883	High	CWE-1333
tough-cookie:4.1.2	NVD CVE-2023-26136	Critical	CWE-1321
ua-parser.js	NVD CVE-2022-25927	High	CWE-1333
vite:4.2.1	NVD CVE-2023-34092	High	CWE-706

Trivy Container scan report

[Save](#) [Copy](#)

SyntaxError: JSON.parse: unexpected character at line 2 column 1 of the JSON data

```
sevenajay/chatbot:latest (alpine 3.18.0)
=====
Total: 21 (UNKNOWN: 0, LOW: 0, MEDIUM: 16, HIGH: 2, CRITICAL: 3)
```

Library	Vulnerability	Severity	Status	Installed Version	Fixed Version	Title
busybox	CVE-2022-48174	CRITICAL	fixed	1.36.0-r9	1.36.1-r1	stack overflow vulnerability in ash.c leads to arbitrary code execution https://avd.aquasec.com/nvd/cve-2022-48174
busybox-binsh						
libcrypto3	CVE-2023-5363	HIGH		3.1.0-r4	3.1.4-r0	openssl: Incorrect cipher key and IV length processing https://avd.aquasec.com/nvd/cve-2023-5363
	CVE-2023-2650	MEDIUM			3.1.1-r0	openssl: Possible DoS translating ASN.1 object identifiers https://avd.aquasec.com/nvd/cve-2023-2650
	CVE-2023-2975				3.1.1-r2	openssl: AES-SIV cipher implementation contains a bug that causes it to ignore... https://avd.aquasec.com/nvd/cve-2023-2975
	CVE-2023-3446				3.1.1-r3	openssl: Excessive time spent checking DH keys and parameters https://avd.aquasec.com/nvd/cve-2023-3446
	CVE-2023-3817				3.1.2-r0	OpenSSL: Excessive time spent checking DH q parameter value https://avd.aquasec.com/nvd/cve-2023-3817
	CVE-2023-5678				3.1.4-r1	openssl: Generating excessively long X9.42 DH keys or checking excessively long X9.42... https://avd.aquasec.com/nvd/cve-2023-5678
	CVE-2023-6129				3.1.4-r3	openssl: POLY1305 MAC implementation corrupts vector registers on PowerPC https://avd.aquasec.com/nvd/cve-2023-6129
	CVE-2023-6237				3.1.4-r4	openssl: Excessive time spent checking invalid RSA public keys https://avd.aquasec.com/nvd/cve-2023-6237
	CVE-2024-0727				3.1.4-r5	openssl: denial of service via null dereference https://avd.aquasec.com/nvd/cve-2024-0727
libssl3	CVE-2023-5363	HIGH			3.1.4-r0	openssl: Incorrect cipher key and IV length processing https://avd.aquasec.com/nvd/cve-2023-5363

MR CLOUD
BOOK

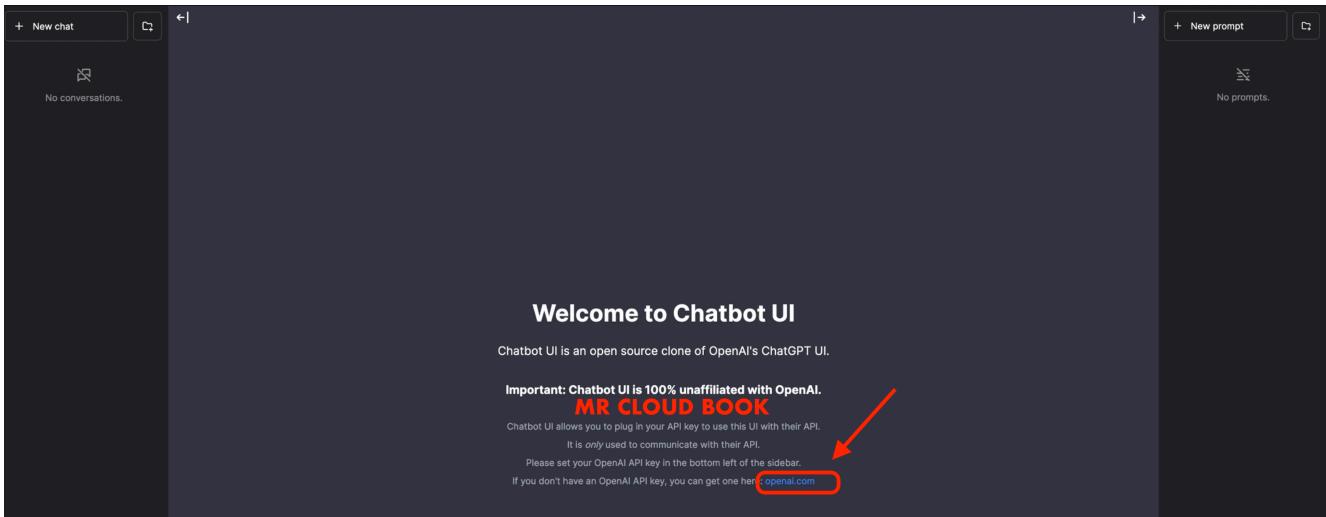
Trivy File scan report

```
package-lock.json (npm)
=====
Total: 6 (UNKNOWN: 0, LOW: 1, MEDIUM: 5, HIGH: 0, CRITICAL: 0)
```

Library	Vulnerability	Severity	Status	Installed Version	Fixed Version	Title	
axios	CVE-2023-45857	MEDIUM	fixed	0.26.1	1.6.0, 0.28.0	axios: exposure of confidential data stored in cookies https://avd.aquasec.com/nvd/cve-2023-45857	
follow-redirects	CVE-2023-26159			1.15.2	1.15.4	follow-redirects: Improper Input Validation due to the improper handling of URLs by... https://avd.aquasec.com/nvd/cve-2023-26159	
next	CVE-2023-46298	LOW	fixed	13.2.4	13.4.20-canary.13	Next.js missing cache-control header may lead to CDN caching empty reply https://avd.aquasec.com/nvd/cve-2023-46298	
postcss	CVE-2023-44270			8.4.14	8.4.31	An issue was discovered in PostCSS before 8.4.31. The vulnerability af..... https://avd.aquasec.com/nvd/cve-2023-44270	
tough-cookie	CVE-2023-26136	MEDIUM		4.1.2	4.1.3	tough-cookie: prototype pollution in cookie memstore https://avd.aquasec.com/nvd/cve-2023-26136	
word-wrap	CVE-2023-26115			1.2.3	1.2.4	word-wrap: ReDoS https://avd.aquasec.com/nvd/cve-2023-26115	

<Jenkins-public-ip:3000>

You will get this output



Click on openai.com to generate the API TOKEN

Click on Create new secret key

API keys

Your secret API keys are listed below. Please note that we do not display your secret API keys again after you generate them.

Do not share your API key with others, or expose it in the browser or other client-side code. In order to protect the security of your account, OpenAI may also automatically disable any API key that we've found has leaked publicly.

Enable tracking to see usage per API key on the [Usage page](#).

NAME	SECRET KEY	TRACKING	CREATED	LAST USED	PERMISSIONS
Test	sk-...bI1S	Enabled	1 Mar 2024	1 Mar 2024	All

+ Create new secret key

MR cloud book

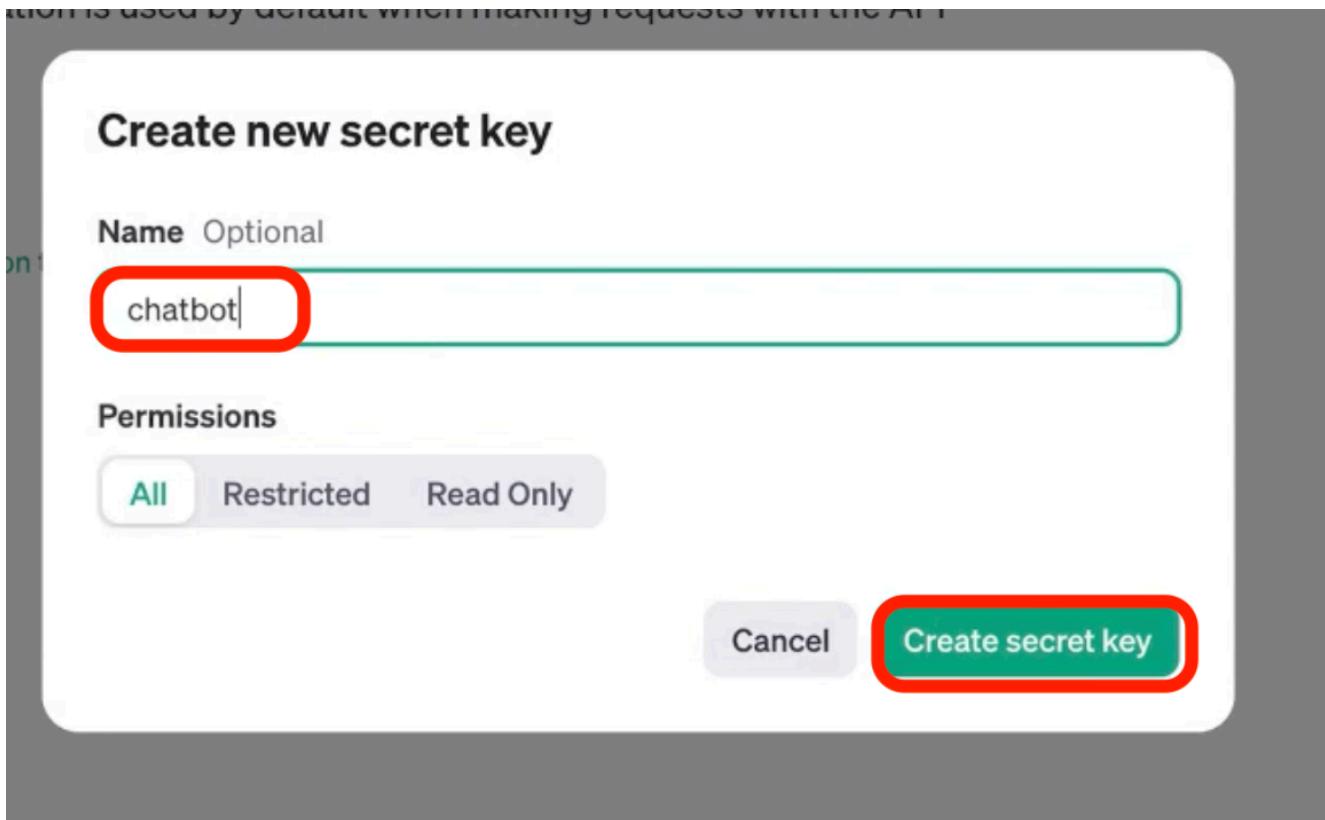
Default organization

If you belong to multiple organizations, this setting controls which organization is used by default when making requests with the API keys above.

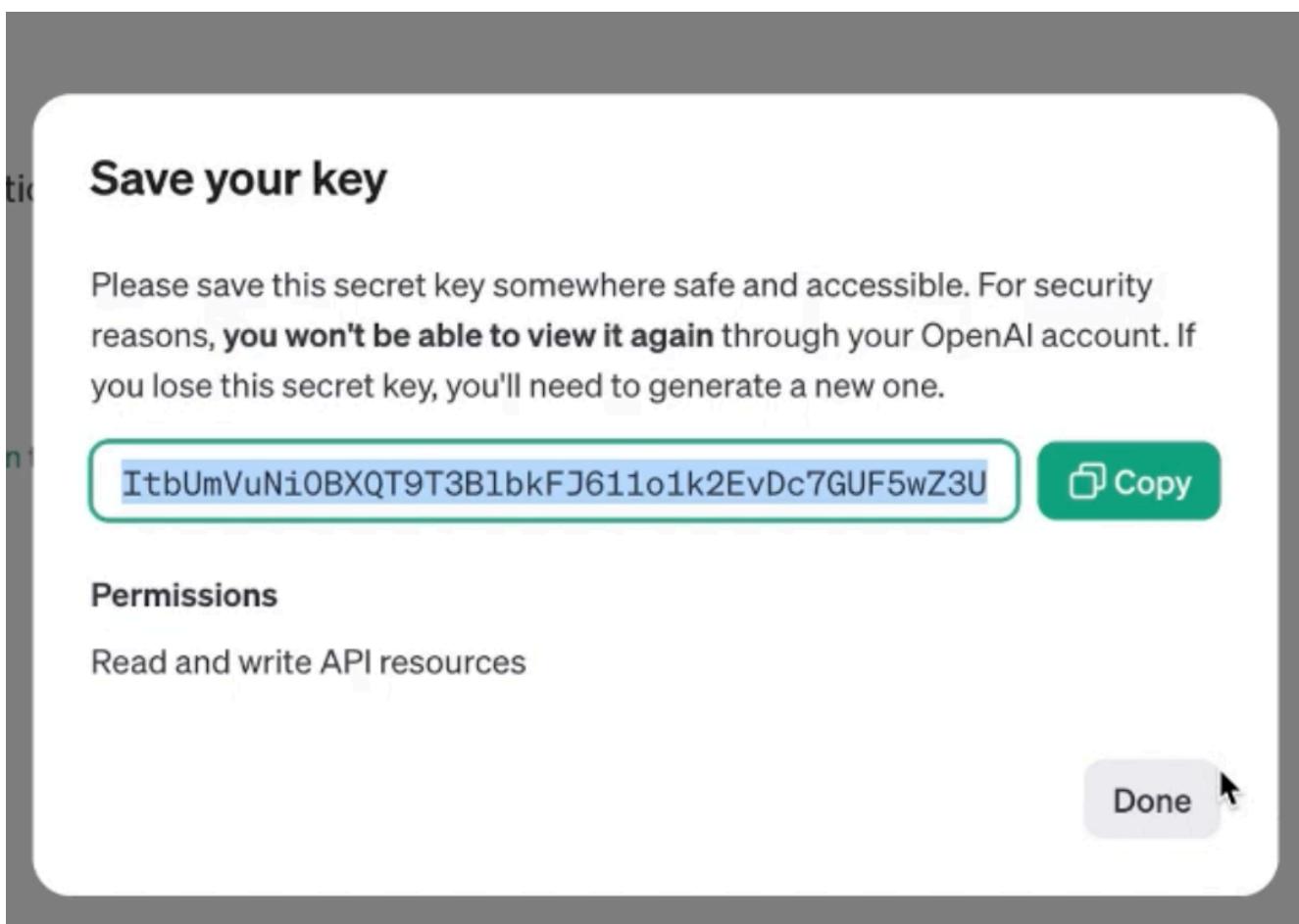
Personal

Note: You can also specify which organization to use for each API request. See [Authentication](#) to learn more.

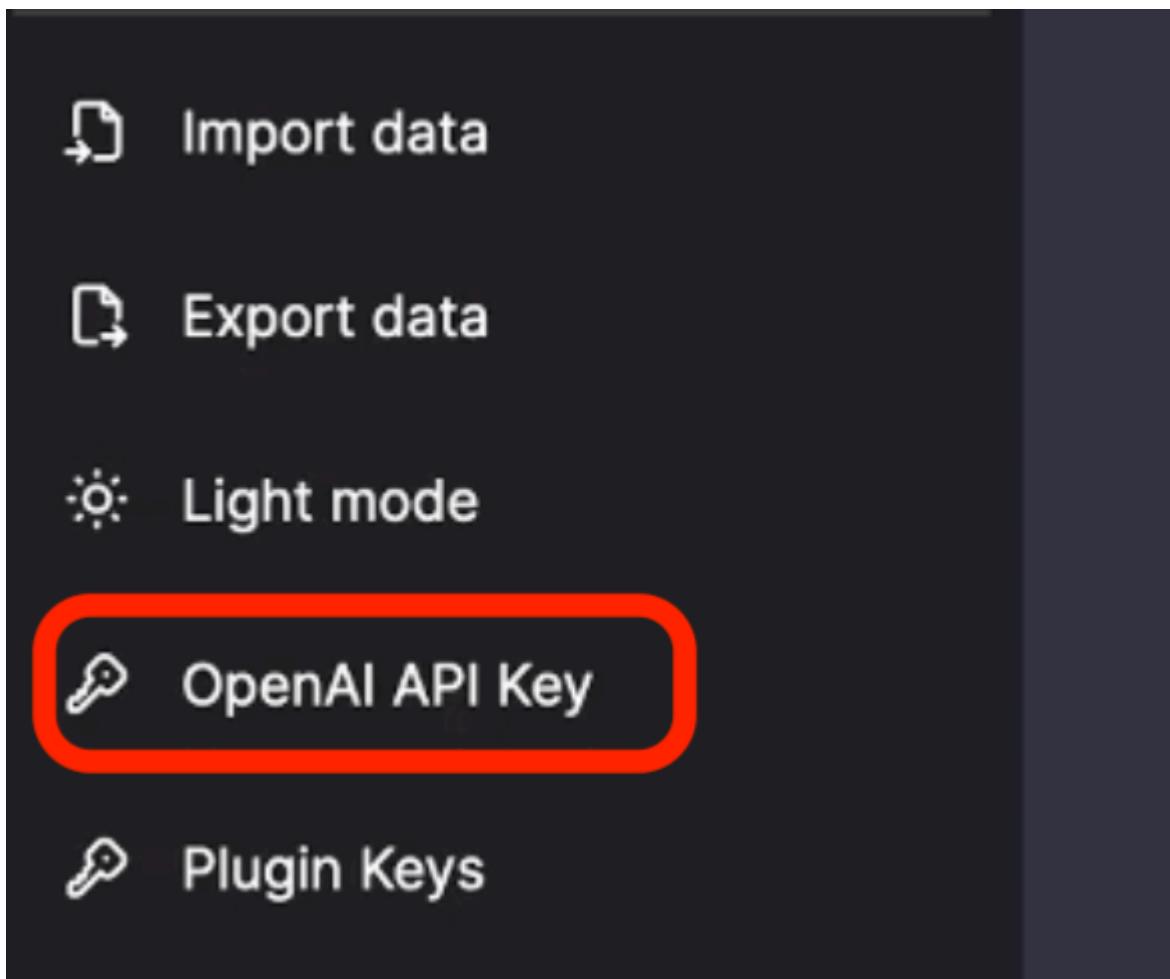
Give Name and click on Create



Copy Token and use

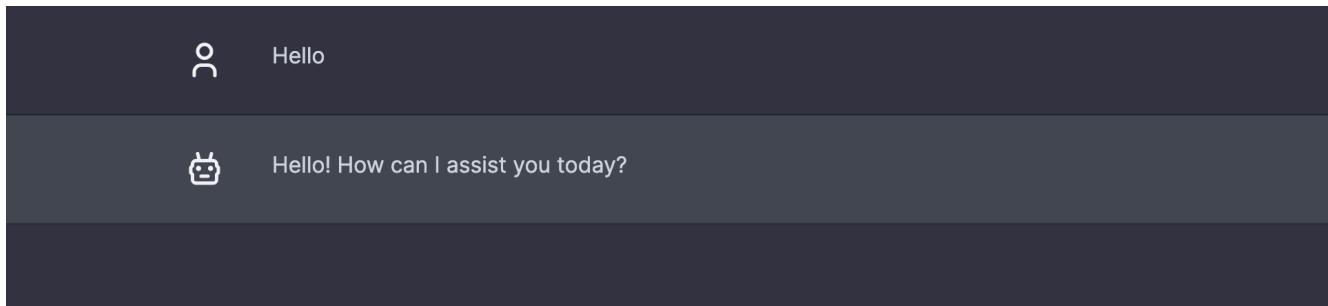


Come back to chatbot UI that we deployed and bottom of the page you will see OpenAI API key and give the Generated key and click on save (RIGHT MARK)



You will see page like this

You can ask questions Now



Or you can generate from [Rapidapi.com](#)

<https://rapidapi.com/hub>

Now In the Jenkins Instance

Give this command



```
eks update-kubeconfig --name <clusternamespace> --region <region>
```



It will Generate an Kubernetes configuration file

Here is the path for config file



```
cd .kube  
cat config
```

copy the file that generates



Save it in your local file explorer, at your desired location with any name as text file.

```
ubuntu@ip-172-31-45-90:~$ cd .kube/  
ubuntu@ip-172-31-45-90:~/kube$  
ubuntu@ip-172-31-45-90:~/kube$ cat config  
apiVersion: v1  
clusters:  
- cluster:  
  certificate-authority-data: LS0tLS1CRUJDTiBDRVJSUZJQ0FURS0tLS0tck1JSURCVENTD0WlyZ0F3SUJBZ0LJ00xHS0kSysrTFV3RFFZSktrWklodnN00VFTEJR0Xrd6VEVUTUJFR0ExVUJKQXhNS2EzVmLaWEp1WlhSbGN60WVGdZTxFeE1U0XoEZjM!pNekvATURj0d5UYTVNEZhTUjVeApFekFS0md0VkbJBTVPD0XwM!1keWt1VjBaWf13Zdrf!al1BMEcdDU3FHU0l1M0RRRUJBUVVb0TRj0k130Xdn20LckFvSUJBUIU3Y3ZkUH!p2bCNvNBZ2E1SVg4MGvPUnx5WtVwdlRk99015aZG40d2VYTU12MFbPq8l1MTBVsXQKH0pkaENYcnzYV2VqM1Z4EF02jdNdanY3hnB2UH!R6MEY30FVKC3F!NXAx0jo3aIMxV2150VNGS2zIvcLZxWk!vAp!tWhhMxF!ME51T0VWmd3eFM10ThCmJzbks3Z01wRzhdJ21dRZL12dxBV001EhmtvMlpJh!9QWmVa0m0wMztZCKR0FF0dm2bEEyV1uy!lgUVESHMjF-wLJ02TmVnVnLISHU09WhYK2V2bE1TTV0YhpySF2R01t!bXVfeG12T3EkxkLs21rZE11eXoxFVik92ljNc!dZrMG5MamNZa01TMVTVn1Nbnd!cdRDbt2dR0ptMThudlKry!ldMh!gapp!UgzerExob!9Fb1tXtjh!VtUxKwC9RELV0WZ00WN0NKFBR2!XyEjYTUE0R0ExWRE-d0VCL3dRR0F5UNREF0CKjntLz1uLc0MY4R0JU0UPB0UgjTU1wR0ExWREZ1FX0!KJS0VWrd6hmc2jNm!x1RjFKR6j4V245Z0svWmNU0YKqm0VkhSRUveakFnZ2dvcnRXSnxjbtVsZedWek!BMEdU3FHU0l1M0RRRUJ0d1VB0TRj0kFR01lqWgxtNWxaawp!TzhHRE5tb1R5M0hZ0ktuRkdzS0l3SDY4BtC2wLJnMjhpEVCYUgwRHMYFaMkdWw06SkwRlZ.NkZLc3j3V6t5D0kxTKF!Nm!9zatkVmh!_NRqTgzbk!z2mVn!30x0Z!b1b2z1dITUvstU0!duz!0WV86!ExrCk9sbXhuzK2CMC84cnTU!LZR!21SVHFK5XN2V8RvT1h!20FlnU!14TEtUyFaZgE1WDF4Nx0B21A0bTdmU!BUC-3oK0mtbhZ6dkhKZhJvC0!L50!  
  server: https://54.242.144.96:443/api/v3?region=ap-south-1  
  name: arn:aws:eks:ap-south-1:672618677785:cluster/EKS_CLOUD  
contexts:  
- context:  
  cluster: arn:aws:eks:ap-south-1:672618677785:cluster/EKS_CLOUD  
  user: arn:aws:eks:ap-south-1:672618677785:cluster/EKS_CLOUD  
  name: arn:aws:eks:ap-south-1:672618677785:cluster/EKS_CLOUD  
current-context: arn:aws:eks:ap-south-1:672618677785:cluster/EKS_CLOUD  
kind: Config  
preferences: {}  
users:  
- name: arn:aws:eks:ap-south-1:672618677785:cluster/EKS_CLOUD  
  user:  
    exec:  
      apiVersion: client.authentication.k8s.io/v1beta1  
      args:  
        - --region  
        - ap-south-1  
        - eks  
        - get-token  
        - --cluster-name  
        - EKS_CLOUD  
        - --output  
        - json  
      command: aws  
ubuntu@ip-172-31-45-90:~/kube$
```

If you Have any doubts refer the YouTube video

<https://youtu.be/aNHdarUeo5U> time stamp 39m:20 secs

Let's add the Final script with Kubernetes

Go to manage Jenkins -> manage credentials -> Click on Jenkins global -> add credentials

Select Kind as Secret file and choose the file that you saved in your local for kubernetes configuration.

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

Complete script

Dont forgot to update the Image in chatbot-ui yml file

```

pipeline{
    agent any
    tools{
        jdk 'jdk17'
        nodejs 'node19'
    }
    environment {
        SCANNER_HOME=tool 'sonar-scanner'
    }
    stages {
        stage('Checkout from Git'){
            steps{
                git branch: 'legacy', url: 'https://github.com/Aj7Ay/chatbot-legacy'
            }
        }
        stage('Install Dependencies') {
            steps {
                sh "npm install"
            }
        }
        stage("Sonarqube Analysis"){
            steps{
                withSonarQubeEnv('sonar-server') {
                    sh ''' $SCANNER_HOME/bin/sonar-scanner -Dsonar.projectName=Chatbot -Dsonar.projectKey=Chatbot '''
                }
            }
        }
        stage("quality gate"){
            steps {
                script {
                    waitForQualityGate abortPipeline: false, credentials: []
                }
            }
        }
        stage('OWASP FS SCAN') {
            steps {
                dependencyCheck additionalArguments: '--scan ./ --disableCheckForKnownVulnerabilities'
                dependencyCheckPublisher pattern: '**/dependency-check-report.html'
            }
        }
        stage('TRIVY FS SCAN') {
            steps {
                sh "trivy fs . > trivyfs.json"
            }
        }
    }
}

```

Apply and save , Run the build to deploy to eks

Declarative: Tool Install	clean workspace	Checkout from Git	Sonarqube Analysis	quality gate	Install Dependencies	OWASP FS SCAN	TRIVY FS SCAN	Docker Build & Push	TRIVY	Deploy to container	Deploy to kubernets
132ms	264ms	1s	25s	295ms	1min 49s	2min 38s	23s	1min 51s	1min 35s	1s	2s
133ms	261ms	1s	25s	284ms	1min 51s	2min 46s	23s	1min 23s	1min 52s	1s	1s

In the Jenkins give this command



```
kubectl get all
kubectl get svc #use anyone
```

Here it will generate loadbalancer DNS dont forgot



```
ubuntu@ip-172-31-40-131:~$ kubectl get all
NAME                                READY   STATUS    RESTARTS   AGE
pod/petshop-768578655f-kzcd9        1/1     Running   0          43s

NAME                TYPE            CLUSTER-IP      EXTERNAL-IP   PORT(S)        AGE
service/kubernetes  ClusterIP       10.96.0.1     <none>        443/TCP       58m
service/petshop     LoadBalancer   10.104.122.152  <pending>    80:30699/TCP  21m

NAME                READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/petshop   1/1     1           1           43s

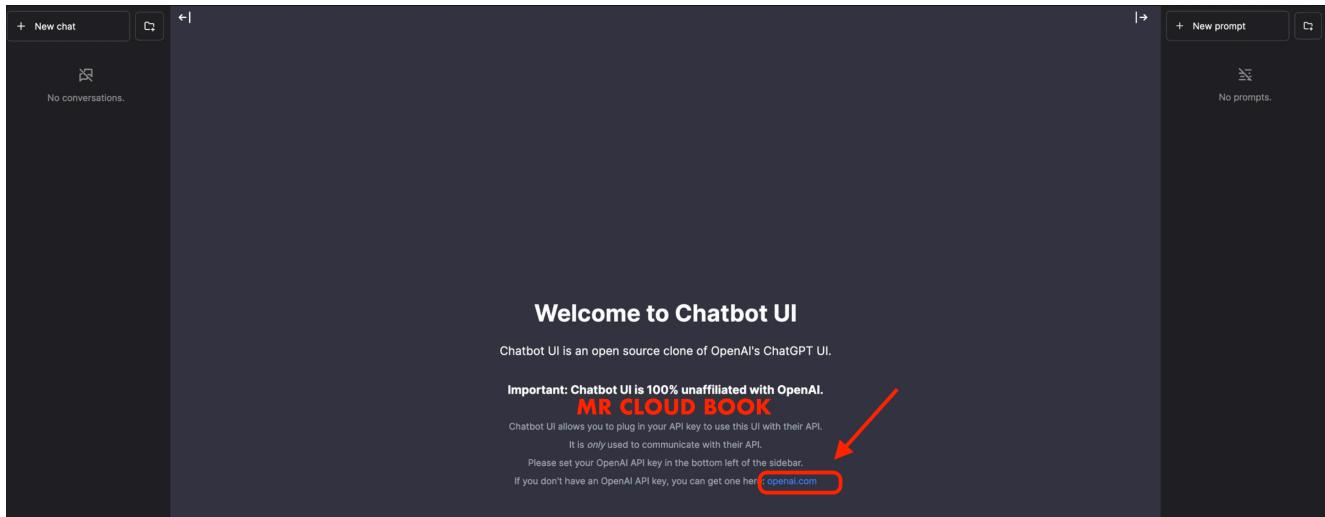
NAME                DESIRED  CURRENT  READY   AGE
replicaset.apps/petshop-768578655f  1        1        1        43s
ubuntu@ip-172-31-40-131:~$
```

Open the load-balancer port to the Cluster EC2 instance

otherwise it wont give output

EXTERNAL IP IN browser gives output

output:



Do the same process and add key to get output

Here is sample query output i used

```
bash
```

 Copy code 

```
docker network prune
```

5. Remove Docker Cache

To cleanup Docker cache or build cache, use:

```
bash
```

 Copy code 

```
docker builder prune
```

- Remove all (include dangling) build cache:

```
bash
```

 Copy code 

```
docker builder prune -a
```

6. Remove Everything

- To remove all unused data in one command:

```
bash
```

 Copy code 

```
docker system prune -a
```

You can also specify `**-f**` or `**--force**` in the command to bypass the confirmation prompt.

Note: Be very careful when using these commands and ensure you don't need the data you're removing, especially when using the `**-a**` (all) or `**-f**` (force)

DESTRUCTION

Update the pipeline with this code and Run again to remove the deployment and container



```
pipeline{
    agent any
    stages {
        stage ("Remove container") {
            steps{
                sh "docker stop chatbot | true"
                sh "docker rm chatbot | true"
            }
        }
        stage('Deploy to kubernets'){
            steps{
                script{
                    withKubeConfig(caCertificate: '', clusterName: '',
                        sh 'kubectl delete -f k8s/chatbot-ui.yaml'
                    )
                }
            }
        }
        stage('scan cluster'){
            steps{
                script{
                    withKubeConfig(caCertificate: '', clusterName: '',
                        sh 'trivy k8s --report summary cluster'
                    )
                }
            }
        }
    }
}
```



The above Script will remove Loadbalancer from Ec2 and removes deployment.

Go to EKS job to destroy cluster

Select Build with parameters and select destory and build

It will Take 6 Minutes to remove the EKS

Delete the EC2 instance at End.

Together, we're shaping the future of user engagement and paving the way for innovative solutions in the world of technology.

Thanks for Reading the blog.

[Aws](#)[awscli](#)[Chatbot](#)[ChatGpt](#)[devops](#)[DevSecOps](#)[Docker](#)[EKS](#)[Github](#)[kubectl](#)[kuberenetes](#)[Npm](#)[openAI](#)[Sonarqube](#)[Terraform](#)[Trivy](#)[Ubuntu](#)

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

12 responses to “Open Source Project: DevSecOps for OpenAI Chatbot UI Deployment | DevSecOps”

**Snehal Sengar**

15 March 2024

Really helpful and clear explanation to the project.

[Reply](#)**Snehal Sengar**

15 March 2024

Really helpful and clear explanation to the project.Thank you

[Reply](#)



Kelvin

22 March 2024

E0322 07:54:41.409163 20196 memcache.go:265] couldn't get current server API group list: the server has asked for the client to provide credentials

E0322 07:54:42.248879 20196 memcache.go:265] couldn't get current server API group list: the server has asked for the client to provide credentials

E0322 07:54:43.063662 20196 memcache.go:265] couldn't get current server API group list: the server has asked for the client to provide credentials

E0322 07:54:43.891800 20196 memcache.go:265] couldn't get current server API group list: the server has asked for the client to provide credentials

E0322 07:54:44.700283 20196 memcache.go:265] couldn't get current server API group list: the server has asked for the client to provide credentials

error: You must be logged in to the server (the server has asked for the client to provide credentials)

This a good project and I have followed the entire procedure but I'm getting the above error message.What could I be missing?

Thank you!

[Reply](#)



Kkrish

2 December 2024

Did u find a solution for it???

[Reply](#)



mrcloudbook.com

2 December 2024

which one ?

[Reply](#)**Zakład pogrzebowy Wola**

7 June 2024

Your writing is both insightful and accessible. I really enjoyed reading your article.

[Reply](#)**Wiaty Fotowoltaiczne**

15 June 2024

What a compelling read! Your points are well-articulated and thought-provoking.

[Reply](#)**Inwertery Solarne**

16 June 2024

Your post is a goldmine of information. It's evident you've put a lot of research and effort into it.

[Reply](#)**Lot Balonem Cennik**

8 July 2024

I'm always excited to read your posts, and this one exceeded my expectations. Excellent work!

[Reply](#)**Ile Kosztuje Lot Balonem**

9 July 2024

Your post was not just informative, but also beautifully written. Your talent is evident.

[Reply](#)



pardeep kaur

15 October 2024

Thank you for amazing project. but in repo there is no Dockerfile is available sir. can you please provide that one.

[Reply](#)



Abhinav Shrivastava

19 November 2024

Thanks for the blog.

[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](#)