Trivy Lab

Install docker

<https://www.openwriteup.com/?page_id=785>

Provide rights to Jenkins user to control docker

sudo usermod -a -G docker jenkins

sudo chown jenkins:docker /var/run/docker.sock

sudo chmod 660 /var/run/docker.sock

Setup Trivy

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

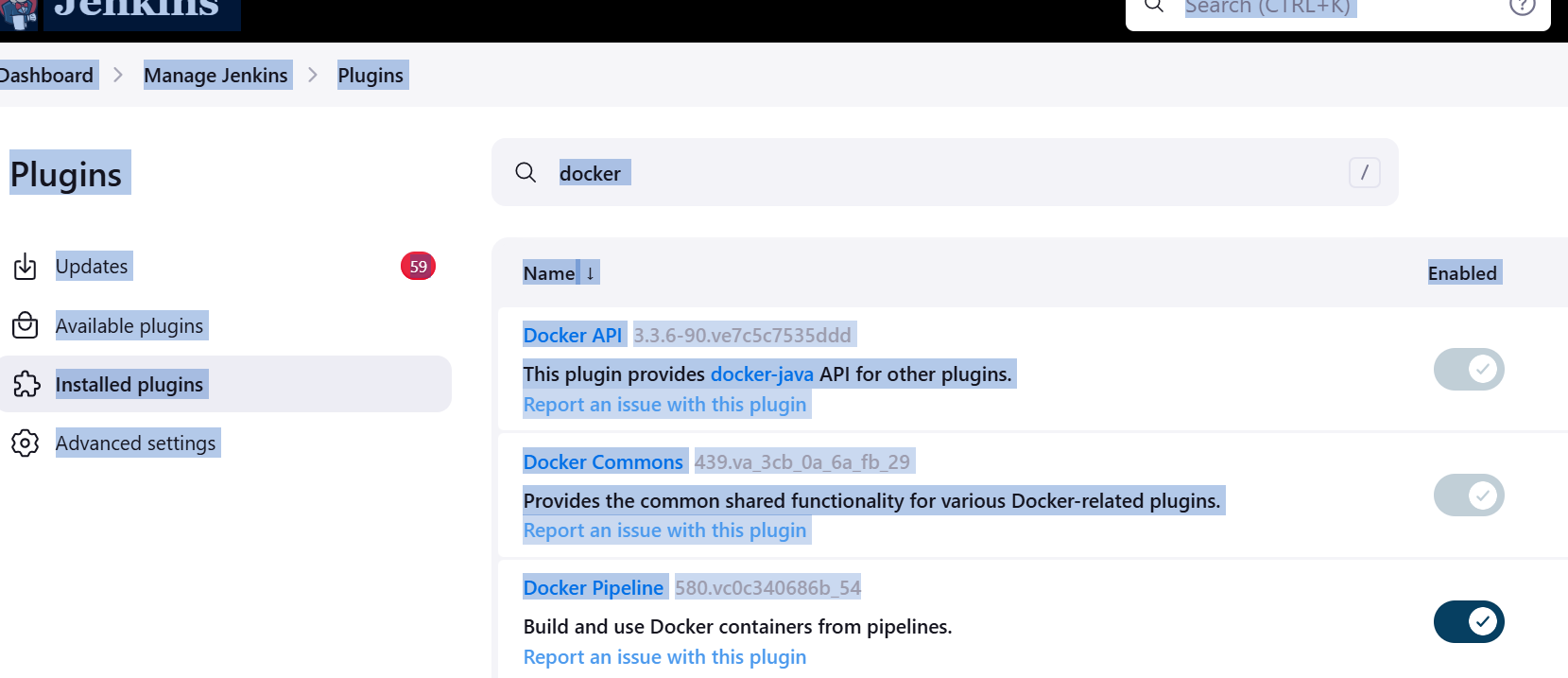
wget -qO - https://aquasecurity.github.io/trivy-repo/deb/public.key | gpg --dearmor | sudo tee /usr/share/keyrings/trivy.gpg > /dev/null

echo "deb [signed-by=/usr/share/keyrings/trivy.gpg] https://aquasecurity.github.io/trivy-repo/deb $(lsb\_release -sc) main" | sudo tee -a /etc/apt/sources.list.d/trivy.list

sudo apt-get update

sudo apt-get install trivy

Manage Jenkins-> plugin->install docker pipeline plugin



1. Install Docker Registry on a server or a cloud-based platform of your choice. You can use a Docker image to do this. For example:

docker run -d -p 5000:5000 --restart=always --name registry registry:2

This will start a Docker Registry container listening on port 5000.

1. Create a docker file in scm (github)

Create a dockerfile name (using vi editor)

FROM python:3.10

WORKDIR /app

COPY requirements.txt .

RUN pip install -r requirements.txt

COPY . .

CMD ["python", "app.py"]

requirements.txt (using vi editor)

SQLAlchemy==2.0.19

mysql-connector-python==8.0.33

gunicorn==21.2.0

requests==2.31.0

Flask>=2.2.5

Werkzeug>=2.2.0

Create a app.py file (using vi editor)

from flask import Flask, jsonify

app = Flask(\_\_name\_\_)

@app.route('/')

def home():

return jsonify({"message": "Backend is running successfully!"})

@app.route('/health')

def health():

return jsonify({"status": "healthy"}), 200

if \_\_name\_\_ == '\_\_main\_\_'

app.run(host='0.0.0.0', port=8000)

4)Create a jenkinsfile

pipeline {

agent any

environment {

GIT\_REPO = ' https://github.com/amitopenwriteup/upgrad.git'

GIT\_BRANCH = ‘main'

DOCKER\_REGISTRY = 'localhost:5000'

IMAGE\_NAME = 'myimage'

IMAGE\_TAG = 'latest'

DOCKERFILE\_PATH = 'dockerfile'

}

stages {

stage('Checkout the Git repository') {

steps {

git branch: "${GIT\_BRANCH}", url: "${GIT\_REPO}"

}

}

stage('Build Docker Image') {

steps {

script {

def dockerImage = docker.build("${DOCKER\_REGISTRY}/${IMAGE\_NAME}:${IMAGE\_TAG}", "-f ${DOCKERFILE\_PATH} .")

dockerImage.push()

}

}

}

stage('Trivy Scan') {

steps {

script {

// Run Trivy scan and output results to a file

def scanResult = sh(script: "trivy image --exit-code 1 --severity HIGH,CRITICAL ${DOCKER\_REGISTRY}/${IMAGE\_NAME}:${IMAGE\_TAG} | tee trivy\_report.txt", returnStatus: true)

// Check if vulnerabilities were found

if (scanResult == 1) {

error "Trivy scan found HIGH/CRITICAL vulnerabilities. Review the 'trivy\_report.txt' log for details."

}

}

}

}

stage('Deploy or Push to Production') {

when {

expression { currentBuild.result == null || currentBuild.result == 'SUCCESS' }

}

steps {

echo "No critical vulnerabilities found, proceeding with deployment."

// Add deployment steps here

}

}

}

post {

always {

archiveArtifacts artifacts: 'trivy\_report.txt', fingerprint: true

}

failure {

echo "Pipeline failed due to vulnerabilities. Check 'trivy\_report.txt'."

}

}

}

Create a Jenkins job :

* Type pipeline
* Pipeline from scm
  + Provide git repo
  + Branch
  + Name of jenkinsfile
* 