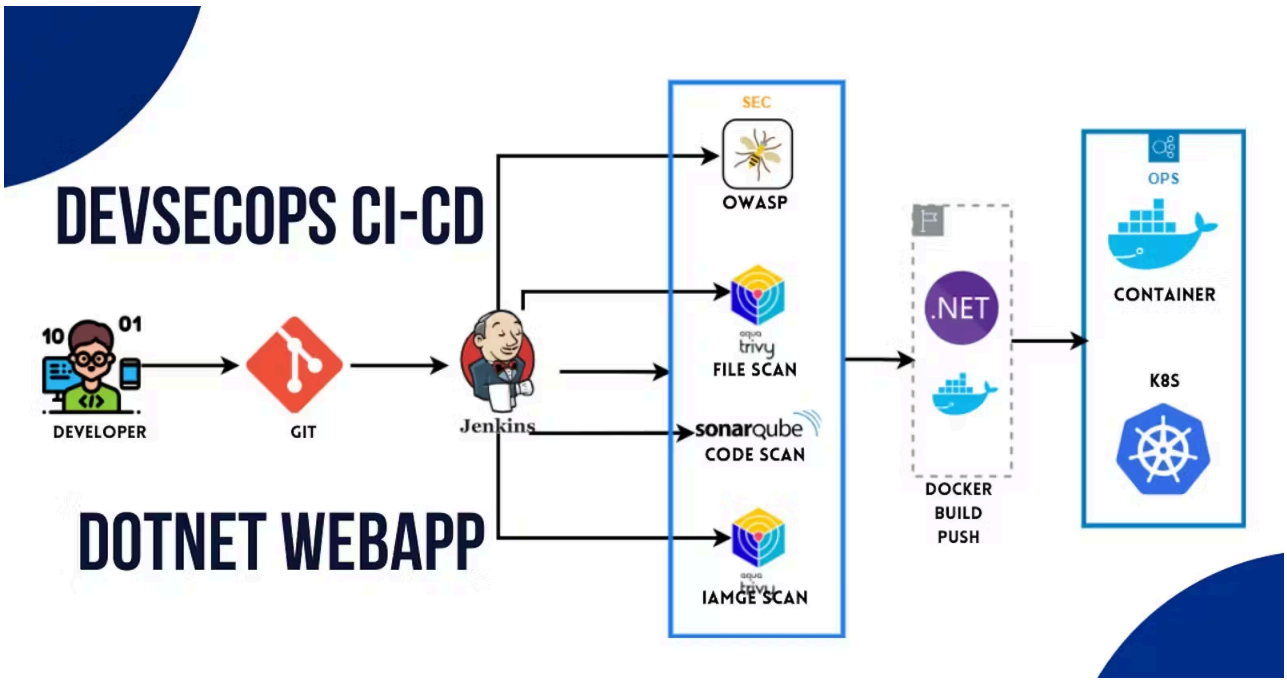
Narendra N's blog

Follow



Dockerised (.Net based) webapp deployment through Jenkins CI/CD

Narendra N

Sep 25, 2023 · 5 min read

#This blog is made by Inspired from Ajay Yogireddi .Thanx to Ajay

Github: https://github.com/Aj7Ay/DotNet-monitoring.git

Steps:-

Step 1 – Create an Ubuntu T2 Large Instance

Step 2 – Install Jenkins, Docker and Trivy. Create a Sonarqube Container using Docker Image of sonar

Step 3 – Install all needed Plugins [eclipse temurin JDK, Sonarqube Scanner, OWASP Dependency Check, docker, kubernetes]

Step 4 – Create a Pipeline Project in Jenkins using scriptive Pipeline

Step 5 – add all credentials in Manage Jenkins Tab

Step 6 – Configure all needed tools under Tools section in Manage Jenkins Tab

Step 7 – Install make to make the package

Step 8 – Docker Image Build and Push

Step 9 – Deploy the image using Docker and access it in browser

Step 10 – Terminate the AWS EC2 Instance

Procedure:

Jenkins, docker, trivy, ansible installation as follows:

COPY

```
#( in Jenkins Server Terminal T2 large ec2)
sudo vi jenkins
#enter the below code inside the jenkins.sh file
```

```
#!/bin/bash
sudo apt update -y
#sudo apt upgrade -y
wget -O - https://packages.adoptium.net/artifactory/api/gpg/key/pub
echo "deb [signed-by=/etc/apt/keyrings/adoptium.asc] https://packag
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
                  /usr/share/keyrings/jenkins-keyring.asc > /dev/nu
echo deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc] \
                  https://pkg.jenkins.io/debian-stable binary/ | su
                               /etc/apt/sources.list.d/jenkins.list
sudo apt-get update -y
sudo apt-get install jenkins -y
sudo systemctl start jenkins
sudo systemctl status jenkins
#  :wq! ----> to save & exit from VI editor
```

COPY

```
sudo chmod 777 jenkins.sh
./jenkins.sh    # this will installl jenkins
```

Note: give "All Traffic" and "any where (0.0.0.0/0)" in Security Group to access from any ports from any IP address since it just demo practice only

COPY

```
<EC2 Public IP /
sudo cat /var/lib/jenkins/secrets/initialAdminPassword
```

```
#copy & paste of the output password of above cat command into Jenk
#set the user credentials and default plug-ins in jenkins
```

COPY

```
#( in above Jenkins server Terminal T2 large ec2)
sudo apt-get update
sudo apt-get install docker.io -y
sudo usermod -aG docker $USER
sudo chmod 777 /var/run/docker.sock
sudo docker ps
docker run -d --name sonar -p 9000:9000 sonarqube:lts-community
```

now access sonarqube in browser with ec2publicIP:9000 and give username(admin), password(admin), next reset password(ex: admin123)

COPY

```
sudo vi trivy.sh
# copy the below script in above trivy.sh file
sudo apt-get install wget apt-transport-https gnupg lsb-release -y
wget -qO - https://aquasecurity.github.io/trivy-repo/deb/public.key
echo "deb [signed-by=/usr/share/keyrings/trivy.gpg] https://aquasec
sudo apt-get update
sudo apt-get install trivy -y
#  :wq! ----> to save & exit from VI editor
```

Goto Manage Jenk

Install below plugins

1 → OWASP ((Install without restart)

2 → SonarQube Scanner

3 → Eclipse Temurin Installer

4 → Docker, Docker Commons, Docker Pipeline, Docker API, Docker-build-step

Goto Manage Jenkins → Tools → Install JDK Click (jdk17, install automatically, select jdk-17.0.8.1+1 from "adoptium.net")

Goto Manage Jenkins → Tools → Dependency-Check Installations (DP-Check, install automatically, select dependecy-check-6.5.1 from "install from github.com")

In sonarqube browser:

Administration → Security → Users → Click on Tokens and Update Token → Give it a name → and click on Generate Token, Create a token with a name and generate, next copy & save the token in any notepad for further use

Goto Dashboard → Manage Jenkins → Credentials → Add Secret Text

kind: secret text, ID: Sonar-token, secret: <above generated token> create

Now, go to Dashboard → Manage Jenkins → Configure System

Sonarqube Installations:

Name: sonar-server, server url: http://<sonarqube browser IP link:9000, select sonar authentication token from drop down menu, apply & save

Goto Manage Jenkins → Tools → Sonarqube-Scanner Installations (Name: sonar-scanner, install automatically, select sonarqube-scanner-5.0.1.3006), apply & save

In sonarqube browser, Administration⟶ Configuration⟶Webhooks ⟶ create

Name: jenkins (any name), URL: <http://jenkins-public-ip:8080>/sonarqube-webhook/ -⟶ create

Now, goto Dashboard → Manage Jenkins → Tools → Docker Installations (Name: docker, install automatically, select latest from docker.com), apply & save

Manage Jenkins⟶Global credentials⟶ user name & password --→ user(dockerhub username), password(dockerhub personal access token or dockerhub password), ID: docker -⟶ create

Next, in Jenkins server Terminal ec2 T2 large

COPY

```
sudo apt install make
# to check version install or not
make -v
```

After run the pipeline job as shown below in Jenkins, access the web app by ec2publicIP

Jenkins Dashboard ⟶ create new item- ⟶ name(dotnet-app-cicd) ⟶ select pipeline- ⟶ OK

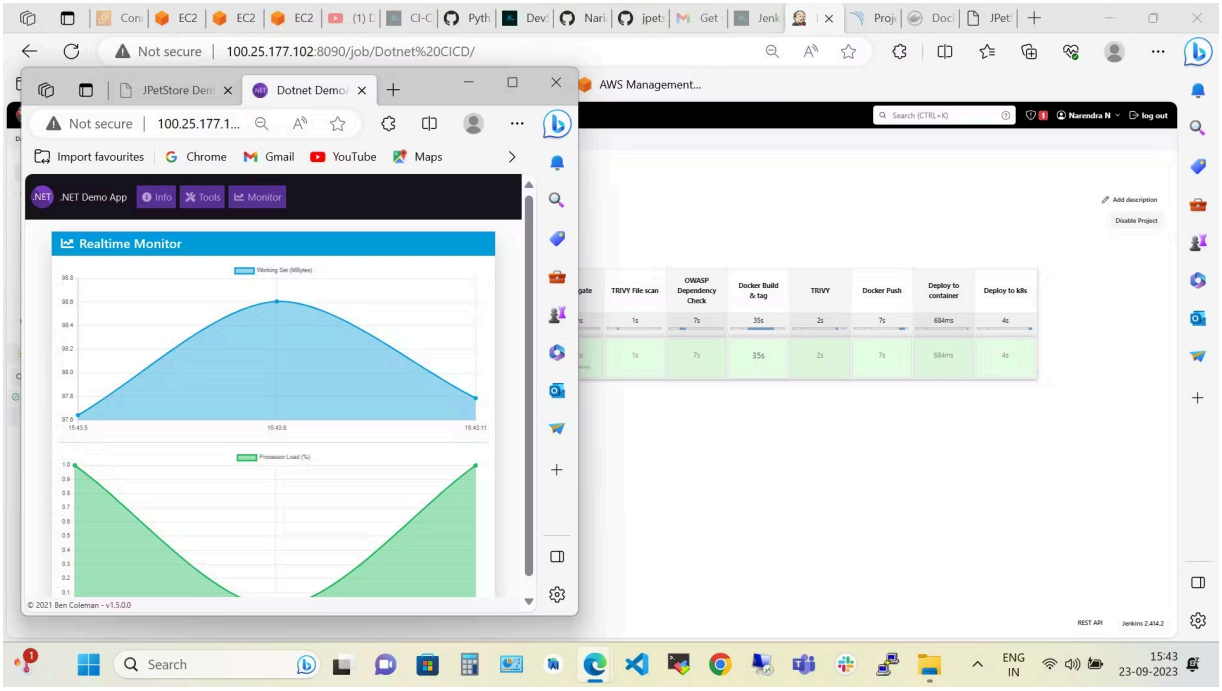
COPY

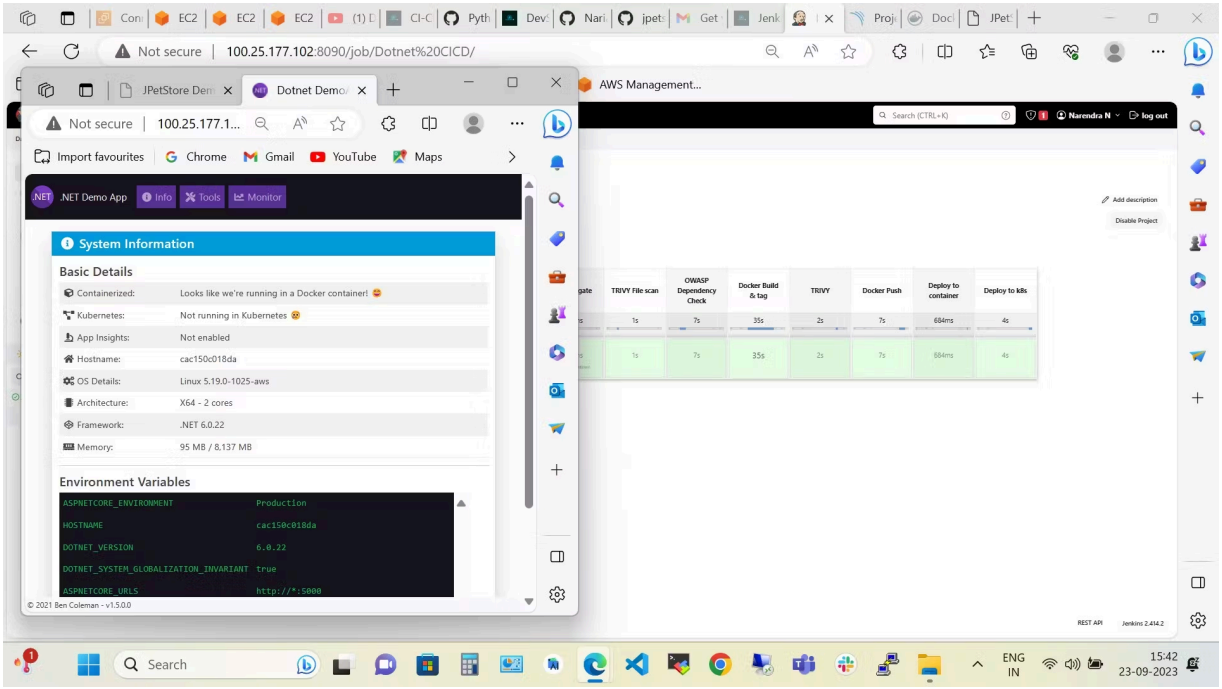
```
#complete pipeline to be paste in sriptive pipeline code in configu
pipeline{
    agent any
    tools{
        jdk 'jdk17'
    }
    environment {
        SCANNER_HOME=tool 'sonar-scanner'
    }
    stages {
        stage('clean workspace'){
            steps{
                cleanWs()
            }
        }
        stage('Checkout From Git'){
            steps{
                git branch: 'main', url: 'https://github.com/Aj7Ay/
            }
        }
        stage("Sonarqube Analysis "){
            steps{
                withSonarQubeEnv('sonar-server') {
                    sh ''' $SCANNER_HOME/bin/sonar-scanner -Dsonar.
                    -Dsonar.projectKey=Dotnet-Webapp '''
                }
            }
        }
        stage("
           steps {
```

```
            script {
                waitForQualityGate abortPipeline: false, creder
            }
        }
    }
    stage("TRIVY File scan"){
        steps{
            sh "trivy fs . > trivy-fs_report.txt"
        }
    }
    stage("OWASP Dependency Check"){
        steps{
            dependencyCheck additionalArguments: '--scan ./ --f
            dependencyCheckPublisher pattern: '**/dependency-ch
        }
    }
    stage("Docker Build & tag"){
        steps{
            script{
               withDockerRegistry(credentialsId: 'docker', tool
                    sh "make image"
                }
            }
        }
    }
    stage("TRIVY"){
        steps{
            sh "trivy image your dockerhub username/dotnet-moni
        }
    }
    stage("Docker Push"){
        steps{
                                                    sId: 'docker', tool
                    sh "make push"
```

```
                    }
                }
            }
        }
        stage("Deploy to container"){
            steps{
                sh "docker run -d --name dotnet -p 5000:5000 your c
            }
        }
        }
    }
}
```

Note: please replace your dockerhub username and some credentials as per you accordingly.....





Subscribe to my newsletter

Read articles from **Narendra N's blog** directly inside your inbox. Subscribe to the newsletter, and don't miss out.

anoopkmathew4u@gmail.cor SUBSCRIBE

.Net App deployment thru CICD pipeline

Written by

Narendra N

Follow

MORE ARTICLES

Narendra N

AKS-Tasks/Projects

Note: # Kubectl, Azure CLI (Ubuntu 22.04) & AKS cluster with NGINX deploy: sudo apt update curl -LO ...

Narendra N

Wordpress/PHP/MySQL-Projects

Note: 1. Install Apache server on Ubuntu sudo apt install apache2 2. Install php runtime and php my...

Narendra N

Jenkins-Projects

Note: 1. Netflix App deploy on k8s through Jenkins CI-CD [DevSecOps] create EC2(ubuntu 22.04, t2...

©2024 Narendra N's blog

Write on Hashnode

Powered by Hashnode - Home for tech writers and readers