*DevOps CI/CD AWS/SONARQUBE Runbook*

*Version: 1.1*

**SONARQUBE Installation with DB(POSTGRESQL)**

**Pre-requisites:**

Instance should have 2GB RAM. And mysql DB not suitable for sonarqube.

**SonarQube Architecture**

SonarQube have 3 components.

1. Sonar scanner
2. Sonar qube server
3. DB Server

**Installation Process:**

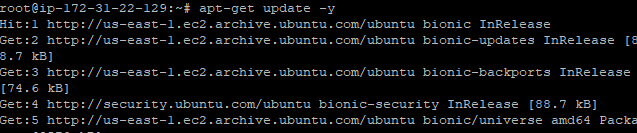
1. To create one t2.medium instance in AWS (Ubuntu machine)
2. Move to super user to root user

**sudo –i**



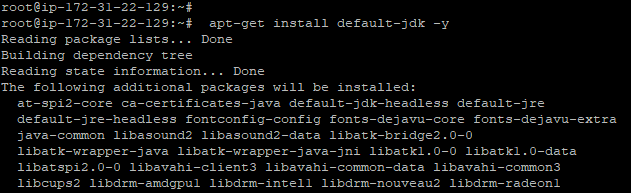
1. Update the machine

**Sudo apt-get update –y**



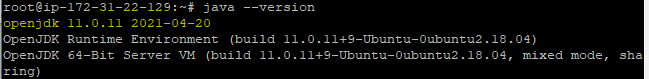
1. Install open jdk -11 in that machine

**apt-get install default-jdk –y**



1. Check the installation of jdk

**java –version**



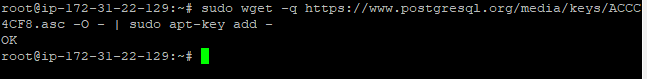
1. Postgres installation

**sudo sh -c 'echo "deb http://apt.postgresql.org/pub/repos/apt/ `lsb\_release -cs`-pgdg main" >> /etc/apt/sources.list.d/pgdg.list'**



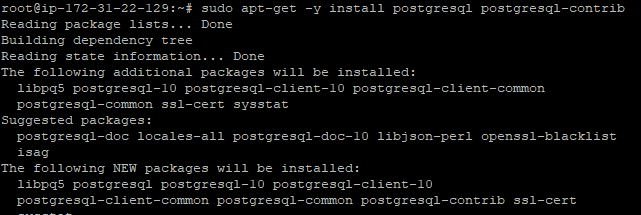
1. Key

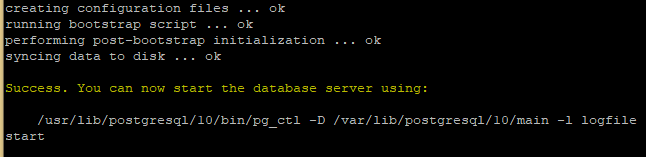
**sudo wget -q https://www.postgresql.org/media/keys/ACCC4CF8.asc -O - | sudo apt-key add**



1. Installation of postgres and postgres contrib

**sudo apt-get -y install postgresql postgresql-contrib**

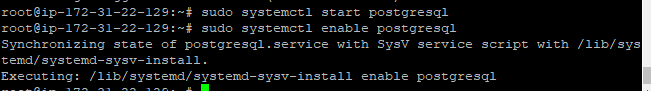




1. Start the postgresql

**sudo systemctl start postgresql**

**sudo systemctl enable postgresql**



1. Login to postgres user

**sudo su - postgres**

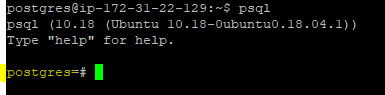


1. Create one user

**createuser sonar**

1. Switch to sql shell

**psql**



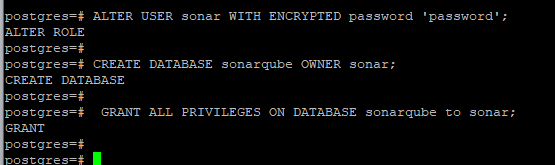
1. Create db using below commands

**ALTER USER sonar WITH ENCRYPTED password 'password';**

**CREATE DATABASE sonarqube OWNER sonar;**

**GRANT ALL PRIVILEGES ON DATABASE sonarqube to sonar;**

**\l(slash small L)**



1. Exit of the postgres

**\q**



1. And type the command for exit to come out of postgres user

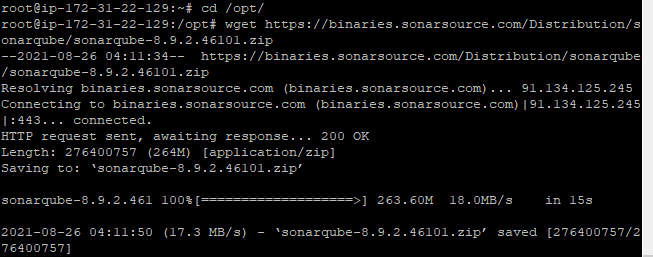
**exit**



**Download and install the sonarqube**

1. Download the sonarqube in opt directory

**wget** [**https://binaries.sonarsource.com/Distribution/sonarqube/sonarqube-8.9.2.46101.zip**](https://binaries.sonarsource.com/Distribution/sonarqube/sonarqube-8.9.2.46101.zip)



1. Unzip the sonarqube

**unzip sonarqube-8.9.2.46101.zip**

if unzip not install in the machine

**apt-get install unzip**

**mv sonarqube-8.9.2.46101 /opt/sonarqube**

1. Create group and user

**sudo groupadd sonarGroup**

1. Now add the user with directory access

**sudo useradd -c "user to run SonarQube" -d /opt/sonarqube -g sonarGroup sonar**

**sudo useradd -c "user to run sonarqube" -d /opt/sonarqube -g sonarGroup sonar**

**sudo chown sonar:sonarGroup /opt/sonarqube -R**

1. Modify sonar.properties file

**sudo vi /opt/sonarqube/conf/sonar.properties**

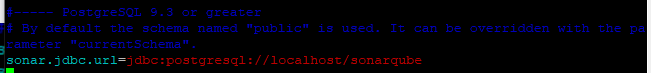
# The schema must be created first.

**sonar.jdbc.username=sonar**

**sonar.jdbc.password=password(sonar)**



**sonar.jdbc.url=jdbc:postgresql://localhost/sonarqube**



sonar.web.host=0.0.0.0

sonar.web.context=/sonar

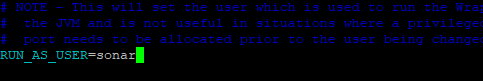
sonar.web.port=9000

1. Edit the sonar script file and set RUN\_AS\_USER

**sudo vi /opt/sonarqube/bin/linux-x86-64/sonar.sh**

**add enable the below line**

**RUN\_AS\_USER=sonar**



1. sudo vi /etc/systemd/system/sonar.service

[Unit]

Description=SonarQube service

After=syslog.target network.target

[Service]

Type=forking

ExecStart=/opt/sonarqube/bin/linux-x86-64/sonar.sh start

ExecStop=/opt/sonarqube/bin/linux-x86-64/sonar.sh stop

LimitNOFILE=131072

LimitNPROC=8192

User=sonar

Group=sonarGroup

Restart=always

[Install]

WantedBy=multi-user.target

1. Kernel system changes to make a few modifications a couple of kernel system limits files for sonarqube to work.

**sudo vi /etc/sysctl.conf**

In sysctl.conf folder to add below system limits

**vm.max\_map\_count=262144**

**fs.file-max=65536**

To edit limits.conf file at the end of the file to add below two lines

**sudo vi /etc/security/limits.conf**

**sonar - nofile 65536**

**sonar - nproc 4096**

1. Reload the system level changes without server boot

**sudo sysctl -p**

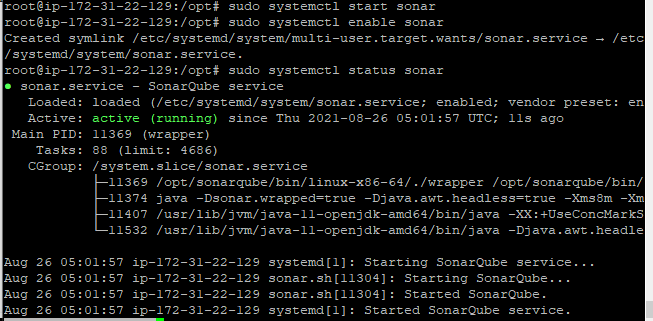


1. Start sonarqube now

**sudo systemctl start sonar**

**sudo systemctl enable sonar**

**sudo systemctl status sonar**



11) go to browser PublicIp of VM Instance:9000/sonar

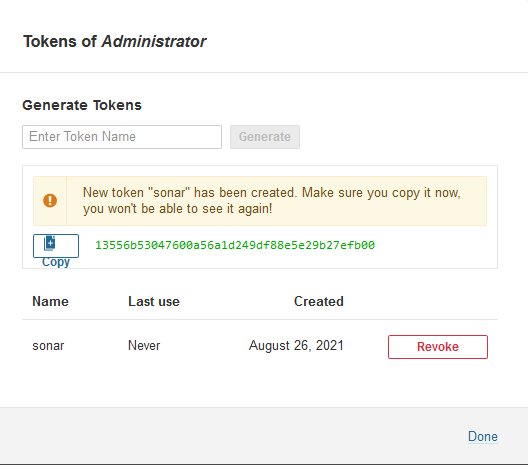
<http://3.12.84.85:9000/sonar>

Change password

Default user and password admin/admin

New User/password: admin/sonar

12) Go to Adminstration🡪Security-🡪 users-> click on Token and generate token



Token🡪 13556b53047600a56a1d249df88e5e29b27efb00

New token

Token🡪4c7d534a50186e49b7066ab763700f8f32e6a43d

13)Go to Jenkins dashboard Mange Jenkins , mange Plugins and Install

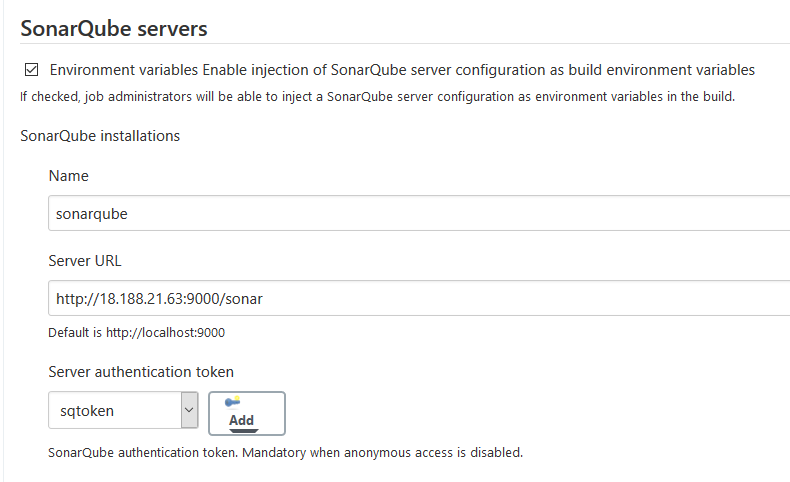
[SonarQube Scanner for Jenkins](https://plugins.jenkins.io/sonar)

14)

Go to manage Jenkins🡪 Configure system 🡪Sonarqube Servers\_ Enable Environment Variables

Server url: url of the sonar qube

Add🡪 secret text🡪 user sonarqube token which is created above



15) go to Jenkins VM🡪 Go to folder conf folder

Cd /opt/apache-maven-3.8.1/conf

vi settings.xml

<id>sonar</id>

<activation>

<activeByDefault>true</activeByDefault>

</activation>

<properties>

<sonar.host.url>http://3.12.84.85:9000/sonar/</sonar.host.url>

<sonar.login>13556b53047600a56a1d249df88e5e29b27efb00</sonar.login>

</properties>

</profile>

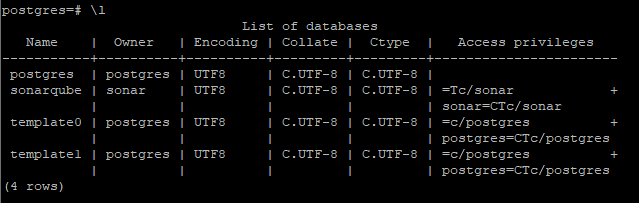
16) Jenkins Pipeline for Sonar qube setup

Check the jekins code below in last step

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Thank You\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

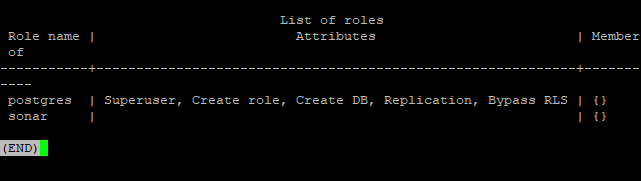
List of all the data bases

\l



List of roles attributes

\du



\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

pipeline {

agent any

// environment {

// AWS\_ACCOUNT\_ID="955470930558"

// AWS\_DEFAULT\_REGION="us-east-2"

// ECRURL='https://955470930558.dkr.ecr.us-east-2.amazonaws.com/ecr-dev-repo-jenkins-eks'

// ECRCRED='ecr:us-east-2:'cred-dev-ecr-jenkins-eks-new'

// IMAGE\_REPO\_NAME="ECR\_REPO\_NAME"

// IMAGE\_TAG="IMAGE\_TAG"

// REPOSITORY\_URI = "${AWS\_ACCOUNT\_ID}.dkr.ecr.${AWS\_DEFAULT\_REGION}.amazonaws.com/${IMAGE\_REPO\_NAME}"

// }

stages {

stage("Git Clone")

{

steps

{

git branch: 'master', url: 'https://github.com/chavaliInfy/java-web-app-docker.git'

}

}

stage("Maven clean package")

{

steps

{

script{

def mvnHome = tool name: 'maven', type: 'maven'

def mvnCMD = "${mvnHome}/bin/mvn"

sh "${mvnCMD} clean package"

withSonarQubeEnv(credentialsId: 'sqtoken')

// withSonarQubeEnv('sonar')

{

sh "${mvnCMD} sonar:sonar"

}

// sh "${mvnCMD} sonar:sonar"

}

}

}

stage("Build Docker Image")

{

steps

{

script{

def buildNumber= BUILD\_NUMBER

// sh "docker build -t ecr-dev-repo-jenkins-eks/java-web-app:${buildNumber} ."

sh "docker build -t ecr-dev-repo-jenkins-eks/java-web-app:latest ."

}

}

}

stage("ECR login")

{

steps

{

script

{

withCredentials([aws(accessKeyVariable: 'AWS\_ACCESS\_KEY\_ID', credentialsId: 'cred-dev-ecr-jenkins-eks-new', secretKeyVariable: 'AWS\_SECRET\_ACCESS\_KEY')])

// docker.withRegistry("https://955470930558.dkr.ecr.us-east-2.amazonaws.com", "ecr:us-east-2:cred-dev-ecr-jenkins-eks-new")

{

sh 'docker tag ecr-dev-repo-jenkins-eks/java-web-app:latest 955470930558.dkr.ecr.us-east-2.amazonaws.com/ecr-dev-repo-jenkins-eks:latest'

sh 'aws ecr get-login-password --region us-east-2 | docker login --username AWS --password-stdin 955470930558.dkr.ecr.us-east-2.amazonaws.com/ecr-dev-repo-jenkins-eks'

sh 'docker push 955470930558.dkr.ecr.us-east-2.amazonaws.com/ecr-dev-repo-jenkins-eks:latest'

}

}

}

}

}

}

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*