\*\*\*Use AWS Ubuntu 18.04 t2.medium\*\*\*

https://techexpert.tips/sonarqube/sonarqube-installation-ubuntu-linux/

Got a EC2 status check failure while implementing above , rebooted twice from the AwS console and issues resolved.

Access public fqdn oon 9000 and it will work. Got this working after 4 hours.

\*\*\*Sonar Scanner Installation & Execution\*\*\*

https://techexpert.tips/sonarqube/sonarqube-scanner-installation-ubuntu-linux/

https://github.com/saurabhjuneja/sonar-example.git

sonar-scanner -X \

-Dsonar.projectKey=Zabbix \

-Dsonar.sources=. \

-Dsonar.host.url=http://10.1.1.212:9000 \

-Dsonar.login=543d7f7ef70d140a76595fc11fc007e33176cda6

root@ip-10-1-1-212:/tmp/sonar-example/ws/manual-measure-ws# sonar-scanner -X -Dsonar.projectKey=Zabbix -Dsonar.sources=. -Dsonar.host.url=http://10.1.1.212:9000 -Dsonar.login=543d7f7ef70d140a76595fc11fc007e33176cda6

===========================================================================================

root@ip-10-1-1-212:/python# cat app.py

import boto3

import boto3

client = boto3.client('ec2',aws\_access\_key\_id='AKIA2QEFLENzzzzCE5S',

aws\_secret\_access\_key='F7GXkGPyi0Lsj7S99D8gSmjIXHznsssssBDNDN',

region\_name='us-east-1')

client = boto3.client('ec2',aws\_access\_key\_id='AKIA2QEFLENzzzzCE5S',

aws\_secret\_access\_key='F7GXkGPyi0Lsj7S99D8gSmjIXHznsssssBDNDN',

region\_name='us-east-1')

vpc = client.describe\_vpcs()

for keys in vpc['Vpcs']:

print(keys['CidrBlock'],'-->',keys['VpcId'])

#print(vpc)

nums = client.describe\_instances()

mantags = ["Env","CostCenter","Email","Name"]

for items in nums['Reservations']:

for holes in items['Instances']:

for mytags in holes['Tags']:

if mytags["Key"] in mantags:

print ("Tag",mytags["Key"],"Exists.")

else:

print("Lets shutdown ",holes['InstanceId'])

docker run -d --name sonarqube -p 9000:9000 -p 9092:9092 sonarqube

wget https://binaries.sonarsource.com/Distribution/sonar-scanner-cli/sonar-scanner-cli-4.6.2.2472-linux.zip

unzip sonar-scanner-cli-4.6.2.2472-linux.zip

mv sonar-scanner-cli-4.0.0.1744-linux.zip /var/opt

sonar-scanner -Dsonar.projectKey=python -Dsonar.sources=. -Dsonar.host.url=http://ec2-54-236-63-56.compute-1.amazonaws.com:9000 -Dsonar.login=4cd1ba2953c99b5503c2ebe46396bf41edb99931

clone dockertest1 and add app.py in it.

root@ip-10-40-1-76:~/sonar/dockertest1# /var/opt/sonar-scanner-4.0.0.1744-linux/bin/sonar-scanner -Dsonar.projectKey=test1 -Dsonar.sources=. -Dsonar.host.url=http://ec2-34-205-41-107.compute-1.amazonaws.com:9000/ -Dsonar.login=078e072012725b8ea75d3de9b4ffd620b6db2d50

Installing Sonar Scanner:

apt install -y unzip openjdk-8-jdk

wget https://binaries.sonarsource.com/Distribution/sonar-scanner-cli/sonar-scanner-cli-4.6.2.2472-linux.zip

unzip sonar-scanner-cli-4.6.2.2472-linux.zip

mv sonar-scanner-4.6.2.2472-linux/ /var/opt/sonar-scanner

root@ip-10-40-1-190:/mypythoncode# ll /var/opt/sonar-scanner-4.6.2.2472-linux/bin/

total 16

drwxr-xr-x 2 root root 4096 May 7 12:16 ./

drwxr-xr-x 6 root root 4096 May 7 12:16 ../

-rwxr-xr-x 1 root root 1822 May 7 12:15 sonar-scanner\*

-rwxr-xr-x 1 root root 662 May 7 12:15 sonar-scanner-debug\*

nano .bashrc

export PATH=$PATH:/var/opt/sonar-scanner/bin/

source .bashrc

root@ip-10-40-1-190:~# sonar-scanner --version

INFO: Scanner configuration file: /var/opt/sonar-scanner-4.6.2.2472-linux/conf/sonar-scanner.properties

INFO: Project root configuration file: NONE

INFO: SonarScanner 4.6.2.2472

INFO: Java 11.0.11 AdoptOpenJDK (64-bit)

INFO: Linux 5.4.0-1045-aws amd64

import boto3

import boto3

client = boto3.client('ec2',aws\_access\_key\_id='AKIA2QEFLENzzzzCE5S',

aws\_secret\_access\_key='F7GXkGPyi0Lsj7S99D8gSmjIXHznsssssBDNDN',

region\_name='us-east-1')

vpc = client.describe\_vpcs()

for keys in vpc['Vpcs']:

print(keys['CidrBlock'],'-->',keys['VpcId'])

nums = client.describe\_instances()

mantags = ["Env","CostCenter","Email","Name"]

for items in nums['Reservations']:

for holes in items['Instances']:

for mytags in holes['Tags']:

if mytags["Key"] in mantags:

print ("Tag",mytags["Key"],"Exists.")

else:

print("Lets shutdown ",holes['InstanceId'])

for items in nums['Reservations']:

for holes in items['Instances']:

for mytags in holes['Tags']:

if mytags["Key"] in mantags:

print ("Tag",mytags["Key"],"Exists.")

else:

print("Lets shutdown ",holes['InstanceId'])

#BUG

this-email-bug@gmail.com

#Code Smell

#https://towardsdatascience.com/5-python-code-smells-you-should-be-wary-of-c48cc0eb9d8b

def addElements(a=[]):

a.append(5)

return aaddElements()

# [5]

addElements()

# [5, 5]

sonar-scanner -Dsonar.projectKey=mywebapp2 -Dsonar.sources=. -Dsonar.host.url=http://ec2-3-239-19-7.compute-1.amazonaws.com:9000 -Dsonar.login=b836ce8b13498de0eff77d8b08d0a363c86e6187

For Testing Ven & Security :

resource "aws\_iam\_role\_policy" "test\_policy" {

name = "test\_policy"

role = aws\_iam\_role.test\_role.id

# Terraform's "jsonencode" function converts a

# Terraform expression result to valid JSON syntax.

policy = jsonencode({

Version = "2012-10-17"

Statement = [

{

Action = [

"iam:CreateAccessKey",

"iam:DeleteAccessKey",

]

Effect = "Allow"

Resource = "\*"

},

]

})

}