Practical Worksheet 2

Version: 1.0 Date: 10/04/2018 Author: David Glance

## Learning Objectives

1. Create an EC2 instance, security group and ssh keys using awscli and python/boto
2. Configure VirtualBox to allow for inbound IP traffic
3. Install and configure Docker and run a hello world application that can be called from the host machine

## Technologies Covered

Ubuntu

AWS

AWS EC2

Python/Boto/awscli/bash scripts

VirtualBox

Docker

**Note**: Do this from your VirtualBox VM – if you do it from any other platform (Windows, Mac – you will need to resolve any potential issues yourself)

## Create an EC2 instance using awscli

[1] Create a security group

aws ec2 create-security-group --group-name devenv-sg --description "security group for development environment"

Note: this will use the default VPC (you will learn about this later in the course) – if you want to specify another VPC, you would use **--vpc-id vpc-*xxxxxxxx***

Note the security group id that is created

[2] Authorise inbound traffic for ssh

aws ec2 authorize-security-group-ingress --group-name devenv-sg --protocol tcp --port 22 --cidr 0.0.0.0/0

[3] Create a key pair that will allow you to ssh to the EC2 instance

aws ec2 create-key-pair --key-name devenv-key --query 'KeyMaterial' --output text > devenv-key.pem

To use this key on Linux, copy the file to a directory ~/.ssh and change the permissions to:

chmod 400 devenv-key.pem

[4] Create the instance and note the instance id

aws ec2 run-instances --image-id ami-d38a4ab1--security-group-ids sg-<from above> --count 1 --instance-type t2.micro --key-name devenv-key --query 'Instances[0].InstanceId'

// 18.04 ami-176aa375

[5] Get the public IP address

aws ec2 describe-instances --instance-ids i-<instance id from above> --query 'Reservations[0].Instances[0].PublicIpAddress'

[6] Connect to the instance

ssh -i devenv-key.pem ubuntu@<IP Address>

[7] Look at the instance using the AWS console

[8] \*\*\*\*NOTE\*\*\*\*\* Once you have finished, log onto the console and terminate the instance **or**

aws ec2 terminate-instances --instance-ids i-<your instance id>

## Create an EC2 instance with Python Boto script

[1] Repeat the steps above using the equivalent Boto commands in a python script. The script should output the IP address to connect to.

[2] Submit the script you create

## Optional: Create an EC2 instance using the console interface. Are there any differences from doing through the command line?

## Install Docker

sudo apt install docker.io

You may have to

sudo systemctl start docker

sudo systemctl enable docker

Check the version

docker --version

## Build and run an httpd container

Create a directory called html

Edit a file index.html and add the following content

<html>

<head>

</head>

<body>

<p>Hello World!</p>

</body>

</html>

Create a file called Docker in the directory above with the following content:

FROM httpd:2.4

COPY ./html/ /usr/local/apache2/htdocs/

Build the docker image

docker build -t my-apache2 .

Run the image

docker run -p 80:80 -dit --name my-app my-apache2

Open a browser and access address <http://localhost> or <http://127.0.0.1>

Confirm you get Hello World!

Other commands

To check what is running

docker ps -a

To stop and remove the container

docker stop my-app

docker rm my-app

## Submission and Quiz

Submit the python code you wrote to create the EC2 instance

## Respond to the Quiz

[1] In dealing with AWS EC2 instances, chmod is a command that is used to:

[A] Change the mode of access to EC2 instances from ssh to ssh+

[B] Change the model of the EC2 machine that is started

[C] Change the permissions on a private key file

[D] The statement is false, it has no part in the creation of an EC2 instance

[2] An EC2 instance has traffic for port 22

[A] Allowed by creating an egress rule on an F5 firewall

[B] Allowed by creating an ingress rule on a security group

[C] Denied and is always blocked for security reasons

[D] Denied by creating an ingress rule on a security group