Step by step guide for REA Simple Sinatra App Demo

1. Install AWS CLI

Follow the steps in the following webpage

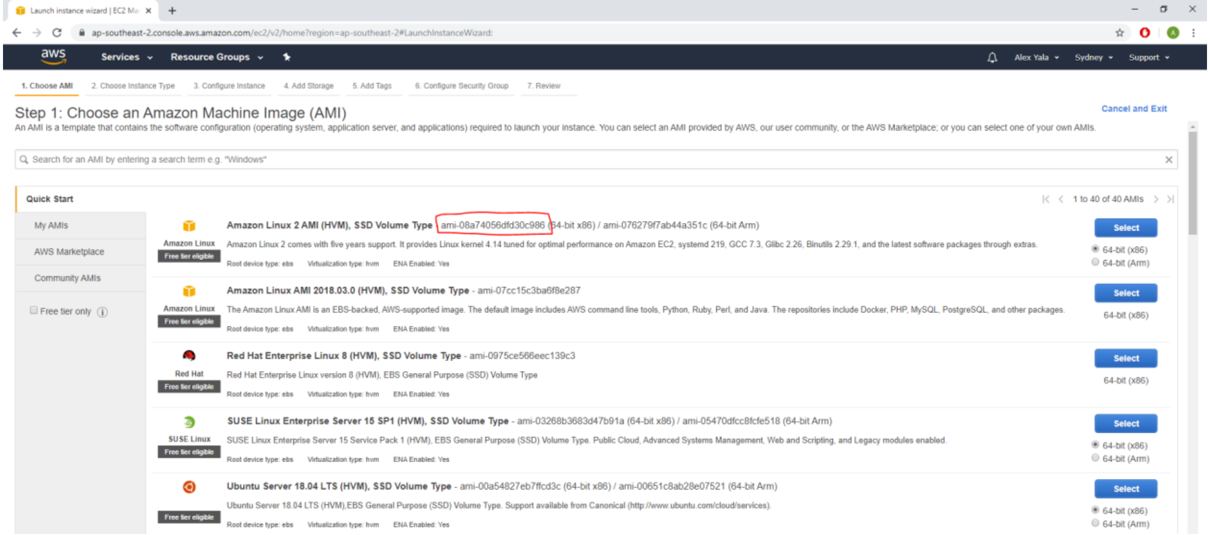
https://docs.aws.amazon.com/cli/latest/userguide/install-cliv2.html

For this demo, I installed it on Linux based OS

2. Configure user parameters:

```
[root@ip-172-31-15-122 ~]# aws configure

AWS Access Key ID [None]: AKIAYP5ZDDRWAT7VB66N

AWS Secret Access Key [None]:
Y8RcK8ZiqQQRRxki          dOyiOTKLhRONy (<< with hidden characters)

Default region name [None]: ap-southeast-2

Default output format [None]:
```

3. Launch Instance for a base VM



Step 1: Choose an Amazon Machine Image (AMI)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. You can select an AMI provided by AWS, our user community, or the AWS Marketplace; or you can select one of your own AMIs.

Search for an AMI by entering a search term e.g. "Windows"

□ Free tier only

Amazon Linux – Free tier eligible
Amazon Linux 2 AMI (HVM), SSD Volume Type - ami-08a74056dfd30c986 (64-bit x86) / ami-076279f7ab44a351c (64-bit Arm)
Amazon Linux 2 comes with five years support. It provides Linux kernel 4.14 tuned for optimal performance on Amazon EC2, systemd 219, GCC 7.3, Glibc 2.26, Binutils 2.29.1, and the latest software packages through extras.
Root device type: ebs Virtualization type: hvm ENA Enabled: Yes
● 64-bit (x86) ○ 64-bit (Arm)

Amazon Linux – Free tier eligible
Amazon Linux AMI 2018.03.0 (HVM), SSD Volume Type - ami-07cc15c3ba6f8e287
The Amazon Linux AMI is an EBS-backed, AWS-supported image. The default image includes AWS command line tools, Python, Ruby, Perl, and Java. The repositories include Docker, PHP, MySQL, PostgreSQL, and other packages.
Root device type: ebs Virtualization type: hvm ENA Enabled: Yes
64-bit (x86)

Red Hat – Free tier eligible
Red Hat Enterprise Linux 8 (HVM), SSD Volume Type - ami-0975ce566eec139c3
Red Hat Enterprise Linux version 8 (HVM), EBS General Purpose (SSD) Volume Type
Root device type: ebs Virtualization type: hvm ENA Enabled: Yes
64-bit (x86)

SUSE Linux – Free tier eligible
SUSE Linux Enterprise Server 15 SP1 (HVM), SSD Volume Type - ami-03268b3683d47b91a (64-bit x86) / ami-05470dfcc8fcfe518 (64-bit Arm)
SUSE Linux Enterprise Server 15 Service Pack 1 (HVM), EBS General Purpose (SSD) Volume Type. Public Cloud, Advanced Systems Management, Web and Scripting, and Legacy modules enabled.
Root device type: ebs Virtualization type: hvm ENA Enabled: Yes
● 64-bit (x86) ○ 64-bit (Arm)

Free tier eligible
Ubuntu Server 18.04 LTS (HVM), SSD Volume Type - ami-00a54827eb7ffcd3c (64-bit x86) / ami-00651c8ab28e07521 (64-bit Arm)
Ubuntu Server 18.04 LTS (HVM),EBS General Purpose (SSD) Volume Type. Support available from Canonical (http://www.ubuntu.com/cloud/services).
Root device type: ebs Virtualization type: hvm ENA Enabled: Yes
● 64-bit (x86) ○ 64-bit (Arm)

ami-08a74056dfd30c986 was used as the base AMI

4. Download keypair

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

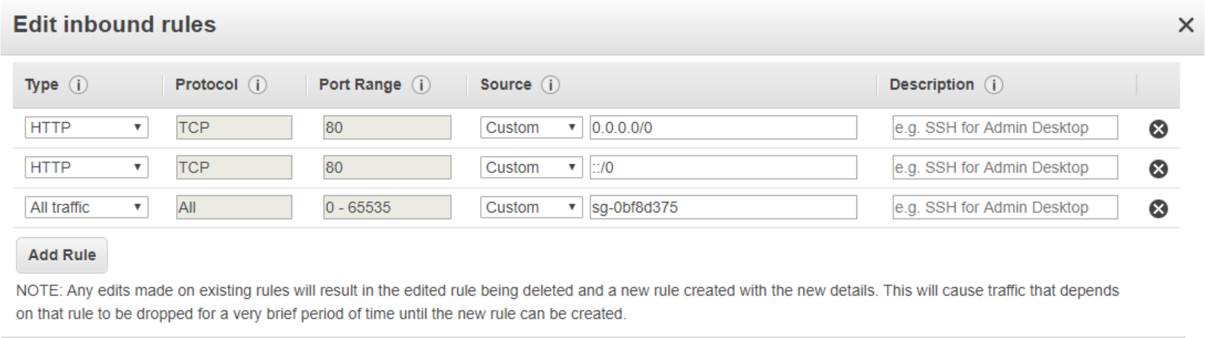
5. Download the VM setup file to configure the VM

```
curl https://raw.githubusercontent.com/4layxel4/REA_demo/master/vmsetup.txt
-o vmsetup.txt
```

Content of the script:

```
#!/bin/bash

yum update -y

yum install -y httpd

service httpd start

chkconfig httpd on

groupadd www

usermod -a -G www ec2-user

chown -R root:www /var/www

chmod 2775 /var/www

find /var/www -type d -exec chmod 2775 {} +

find /var/www -type f -exec chmod 0664 {} +

echo "<h1> Hello REA from $HOSTNAME. This instance was created on `date
+%d` `date +%b` `date +%Y` </h1>" > /var/www/html/index.html
```

6. Make sure the Default Security Group has Incoming HTTP port open from anywhere (0.0.0.0/0)



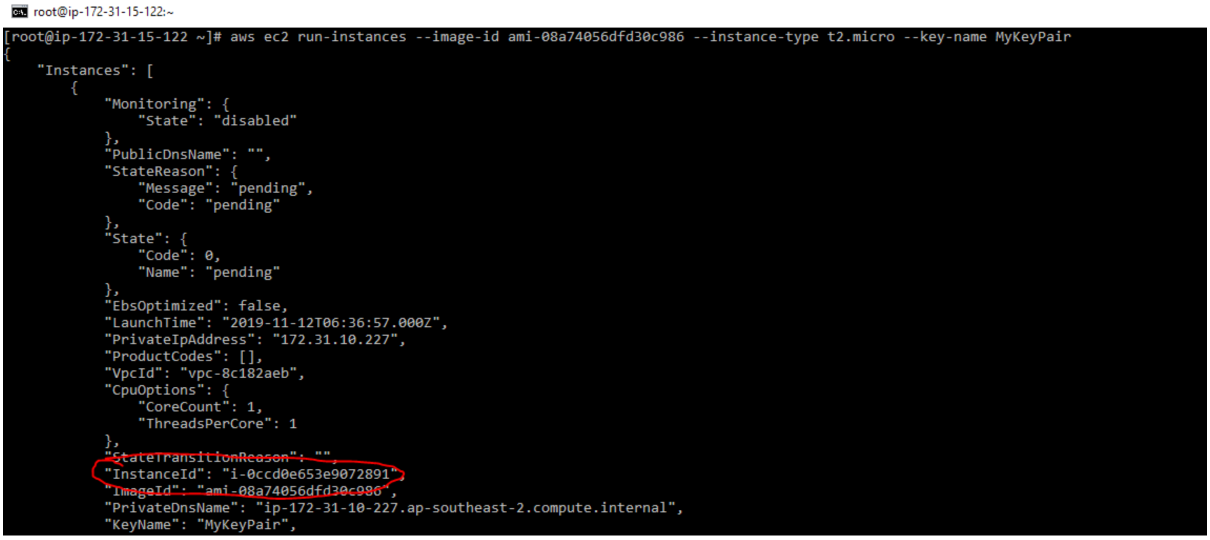
Edit inbound rules

Type	Protocol	Port Range	Source		Description	
HTTP	TCP	80	Custom	0.0.0.0/0	e.g. SSH for Admin Desktop	⊗
HTTP	TCP	80	Custom	::/0	e.g. SSH for Admin Desktop	⊗
All traffic	All	0 - 65535	Custom	sg-0bf8d375	e.g. SSH for Admin Desktop	⊗

NOTE: Any edits made on existing rules will result in the edited rule being deleted and a new rule created with the new details. This will cause traffic that depends on that rule to be dropped for a very brief period of time until the new rule can be created.

7. Create the base instance

```
aws ec2 run-instances --image-id ami-08a74056dfd30c986 --instance-type
t2.micro --key-name MyKeyPair
```



```
[root@ip-172-31-15-122 ~]# aws ec2 run-instances --image-id ami-08a74056dfd30c986 --instance-type t2.micro --key-name MyKeyPair
{
    "Instances": [
        {
            "Monitoring": {
                "State": "disabled"
            },
            "PublicDnsName": "",
            "StateReason": {
                "Message": "pending",
                "Code": "pending"
            },
            "State": {
                "Code": 0,
                "Name": "pending"
            },
            "EbsOptimized": false,
            "LaunchTime": "2019-11-12T06:36:57.000Z",
            "PrivateIpAddress": "172.31.10.227",
            "ProductCodes": [],
            "VpcId": "vpc-8c182aeb",
            "CpuOptions": {
                "CoreCount": 1,
                "ThreadsPerCore": 1
            },
            "StateTransitionReason": "",
            "InstanceId": "i-0ccd0e653e9072891",
            "ImageId": "ami-08a74056dfd30c986",
            "PrivateDnsName": "ip-172-31-10-227.ap-southeast-2.compute.internal",
            "KeyName": "MyKeyPair",
```

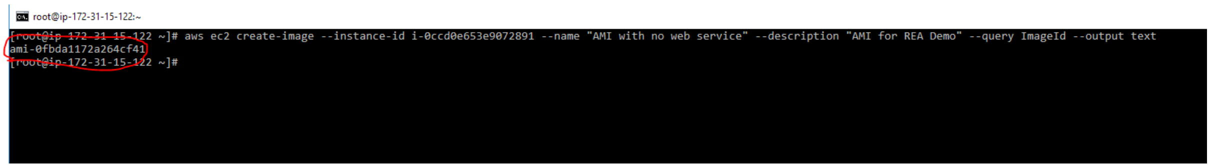
Instance Id = i-0ccd0e653e9072891

Note down the Instance ID

8. Personalise the VM according to the organisation IT policy

9. Create Personal AMI

```
aws ec2 create-image --instance-id i-0ccd0e653e9072891 --name "AMI with no
web service" --description "AMI for REA Demo" --query ImageId --output text
```



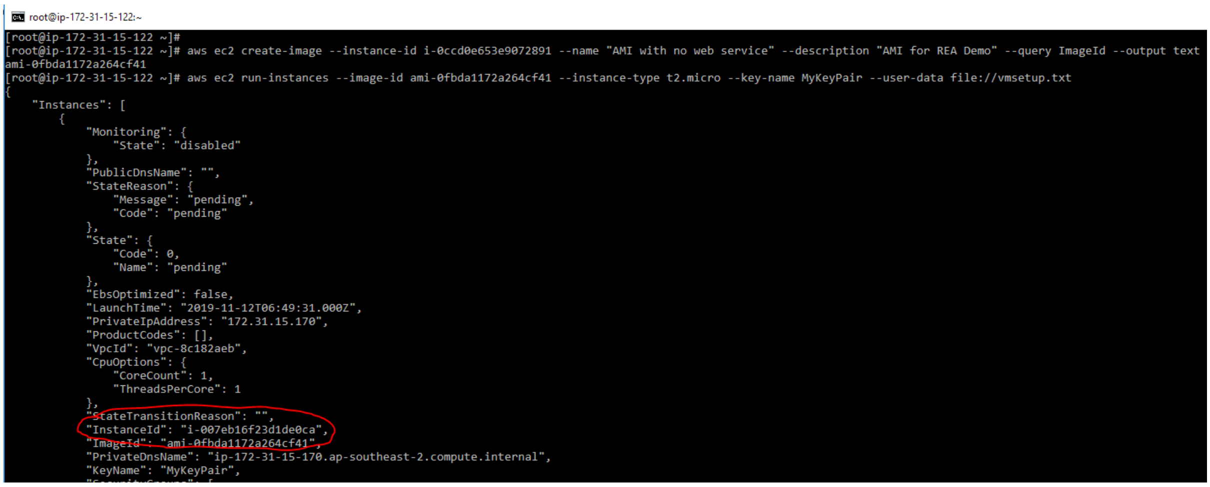
```
[root@ip-172-31-15-122 ~]# aws ec2 create-image --instance-id i-0ccd0e653e9072891 --name "AMI with no web service" --description "AMI for REA Demo" --query ImageId --output text
ami-0fbda1172a264cf41
[root@ip-172-31-15-122 ~]#
```

ami-0fbda1172a264cf41

10. Launching an instance using personal AMI

Create a new instance with webserver installed using the script.

aws ec2 run-instances --image-id ami-0fbda1172a264cf41 --instance-type t2.micro --key-name MyKeyPair --user-data file://vmsetup.txt



```
[root@ip-172-31-15-122 ~]#
[root@ip-172-31-15-122 ~]# aws ec2 create-image --instance-id i-0ccd0e653e9072891 --name "AMI with no web service" --description "AMI for REA Demo" --query ImageId --output text
ami-0fbda1172a264cf41
[root@ip-172-31-15-122 ~]# aws ec2 run-instances --image-id ami-0fbda1172a264cf41 --instance-type t2.micro --key-name MyKeyPair --user-data file://vmsetup.txt
{
    "Instances": [
        {
            "Monitoring": {
                "State": "disabled"
            },
            "PublicDnsName": "",
            "StateReason": {
                "Message": "pending",
                "Code": "pending"
            },
            "State": {
                "Code": 0,
                "Name": "pending"
            },
            "EbsOptimized": false,
            "LaunchTime": "2019-11-12T06:49:31.000Z",
            "PrivateIpAddress": "172.31.15.170",
            "ProductCodes": [],
            "VpcId": "vpc-8c182aeb",
            "CpuOptions": {
                "CoreCount": 1,
                "ThreadsPerCore": 1
            },
            "StateTransitionReason": "",
            "InstanceId": "i-007eb16f23d1de0ca",
            "ImageId": "ami-0fbda1172a264cf41",
            "PrivateDnsName": "ip-172-31-15-170.ap-southeast-2.compute.internal",
            "KeyName": "MyKeyPair",
```

Instance ID = i-007eb16f23d1de0ca

11. Confirm the web service is working

Follow the steps below:

1. Login to AWS Console
2. Go to EC2
3. Find the Instance ID from the last launch
4. Find the Public IP address of the instance
5. Browse to the public IP address to confirm that the website is working