

Step by step guide for REA Simple Sinatra App Demo

1. Install AWS CLI

2. Configure user parameters:

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

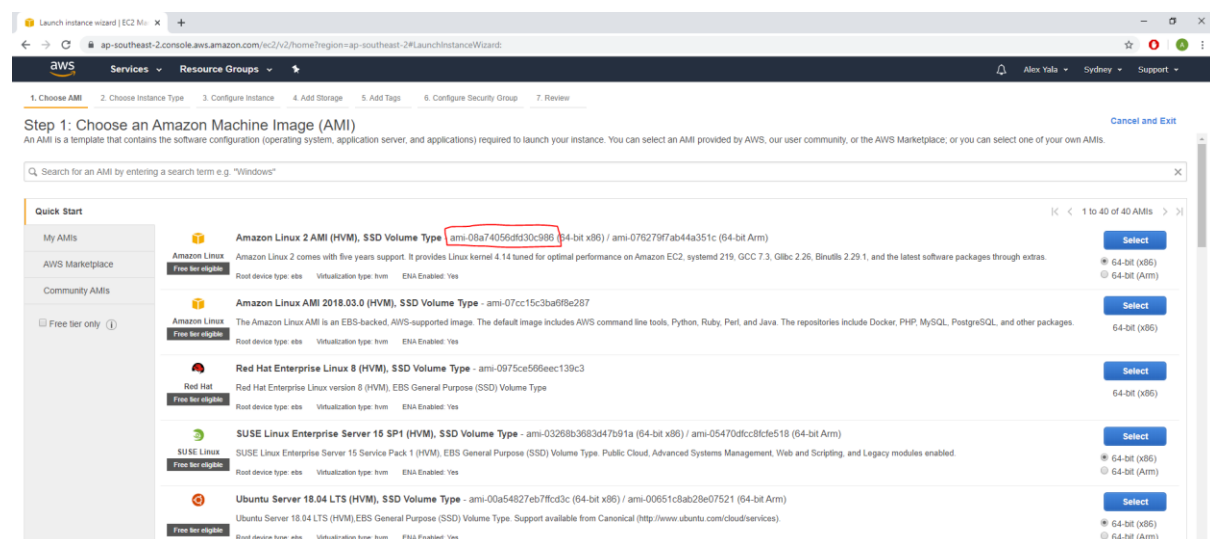
```
AWS Access Key ID [None]: AKIAYP5ZDDRWAT7VB66N
```

```
AWS Secret Access Key [None]:  
Y8RcK8ZiqQzzzzzzzz5KNGfaAtNdOyiOTKLhRONy
```

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

```
Default output format [None]:
```

3. Launch Instance for a base VM



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 <small>i</small> | Protocol <small>i</small> | Port Range <small>i</small> | Source <small>i</small> | Description <small>i</small> | |
|-----------------------|---------------------------|-----------------------------|-------------------------|------------------------------|---|
| 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 | ✕ |

Add Rule

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~  
[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-0cccd0e653e9072891",  
      "ImageId": "ami-08a74056dfd30c986",  
      "PrivateDnsName": "ip-172-31-10-227.ap-southeast-2.compute.internal",  
      "KeyName": "MyKeyPair",  
    }  
  ]  
}
```

Instance Id = i-0cccd0e653e9072891

Note down the Instance ID

8. Create Personal AMI

```
aws ec2 create-image --instance-id i-0cccd0e653e9072891 --name "AMI with no web service" --description "AMI for REA Demo" --query ImageId --output text
```

```
root@ip-172-31-15-122~  
[root@ip-172-31-15-122 ~]# aws ec2 create-image --instance-id i-0cccd0e653e9072891 --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

9. 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",  
      "SubnetId": "subnet-4a5b6c7d"  
    }  
  ]  
}
```

Instance ID = i-007eb16f23d1de0ca

10. Confirm the web service is working

Login to AWS Console > EC2 > find the Instance ID from the last launch > find the Public IP address > Browse to the public IP address