***Note****:*

*- For most of the EC2 labs, make sure you select Mumbai region.*

*- If you are selecting new region, make sure respective Security Groups and Keys are created.*

*- Make sure you select free tier wherever possible e.g. During Image or instance type selection*

**Lab 1: Creating Key Pair for EC2**

1. Select Services -> EC2

2. Click on Key Pairs -> Create Key Pair

3. Enter key Pair Name -> Create

4. File format – If you want to use Putty to connect instance then use ppk format, for all other tools use pem format. Please select accordingly.

5. One part of the key i.e. Private key gets downloaded on your computer, and other part i.e. public key gets created or saved. Make sure keep that private key or password with you.

***Note****: If you are providing above created key during VM creation then it will be used as password to login to VM. You can create pem format to ppk format. Additionally, if you have already key pair, then you can import the public key also by using action-> Import key pair.*

**Lab 2: Creating Security Group**

1. Services -> Security Group-> Create Security Group

2. Enter SG name (e.g. LinuxWebServerSG) and description. Keep the VPC by default

3. Click on Add Rule in Inbound Rules

4. Add SSH, HTTP rules and save it. Make sure, source should be 0.0.0.0/0 which is internet.

5. Similarly, create Windows security group. e.g. WindowsWebServerSG. Add RDP and HTTP rule in inbound section.

***Note****: No need to update anything in outbound section, as its by default open to all.*

**Lab 3: Provision a Windows VM and access it.**

1. Select Services -> EC2 -> Launch Instance

2. Select AMI as Windows 2k16 Base or Win 2k12 Base. You can select any windows under free tier.

3. Make sure instance type is selected as general purpose with free tier. Click Next

4. Keep all inputs by default as it is. Click Next

5. Keep storage as it is. Click Next

6. You can add any tag. e.g. Key as "Name" and put value as "Windows". You can skip this step as well. Click Next.

7. Select an existing group. It will show already created security group. e.g. WindowsWebServerSG. You can create new SG here as well and add required rules. Click Next and Launch

8. It will ask for key pair. Either you can create new keypair here or pass the existing key pair generated during Lab 1.

9. Click Launch. Click on view instances.

**Validation Steps:**

1. On instance or EC2 dashboard, you will Windows VM provision is in progress. After some time, it will show in running state.

2. Go through all the tabs like Description, status check, Monitoring etc

3. Select VM provisioned and select Action button.

4. Try stop, start, reboot options

5. To connect windows VM, click on Connect. Click on Get Password. Click on Choose File and provide the correct key file you downloaded at the time of key pair generation

6. Click On decrypt password. Copy that password. Click on Download Remote Desktop File and provide the copied password.

7. Please note that, you can also connect different way by launching existing remote desktop connection on windows. Default Username is **Administrator**.

8. You should be able to log in successfully.

**Lab 4: Provision a Linux VM and access it.**

1. Select Services -> EC2 -> Launch Instance

2. Select AMI as Amazon AMI. You can select any Linux AMI under free tier.

3. Make sure instance type is selected as general purpose with free tier. Click Next

4. Keep all inputs by default as it is. Click Next

5. Keep storage as it is. Click Next

6. You can add any tag. e.g. Key as "Name" and put value as "Linux". Its optional, you can skip this. Click Next.

7. Select an existing group. It will show already created security group. e.g. LinuxWebServerSG. You can create new SG here as well and add required rules. Click Next and Launch

8. It will ask for key pair. Either you can create new key pair here or pass the existing key pair generated during Lab 1.

9. Click Launch. Click on view instances.

**Validation Steps:**

1. On instance or EC2 dashboard, you will Linux VM provision is in progress. After some time, it will show in running state.

2. Go through all the tabs like Description, status check, Monitoring etc

3. Select VM provisioned and select Action button.

4. Try stop, start, reboot options

5. To connect or access Linux VM, there are multiple ways.

- **With Mobaxterm tool**

1. Open the mobaxterm tool.

2. Click on session -> SSH.

3. Enter the public IP in remote host

4. Enter username as ec2-user. This is default user name on AWS AMI. It may change as per AMI

5. Click on Advanced SSH Settings -> Click on Use Private Key and provide the path of private key.pem -> Ok

6. Now, you should be able to login successfully. Do 'sudo su' to login as root.

- **From other Linux or Mac OS**

1. If you want to access Linux from any other Linux or MAC, then below is the command to access it. You need private\_key.pem to be created with required content on VM from where you are accessing it. Either you can copy content as it is and create file or copy the same file from your computer to VM from where you are accessing the other Linux VM. Once you copied, please change the permission.

e.g. chmod 400 <private\_key.pem>

ssh <user>@<public\_ip> -i <private\_key.pem>

e.g. ssh ec2-user@1.2.3.4 -i key.pem

- **By Putty**

***Note****: Putty doesn't understand the pem format for key, hence we need to convert pem key into ppk format.*

1. Open the puttygen tool and click the load.

2. Select your private key file in pem format.

3. Click on save private key and for passphrase password dialogue click yes.

4. Save this key in ppk format now.

5. Open the putty.exe

6. Enter the public IP of Linux VM in hostname

7. Expand SSH on left side. Click on Auth and provide the ppk format key.

8. Provide the user name ec2-user and you should be able to login to Linux VM successfully.

- **[New] By Instance Connect**

***Note:*** *This is newly added functionality to connect EC2 instance. This is only available for Linux AMI 2 latest image.*

1. Provision new Linux Instance by using Amazon Linux 2 AMI (HVM)

2. Once Instance is in running state, click on button Connect

3. Click on EC2 instance connect (Browser based SSH connection)

4. No need to mention SSH key.

**Lab 5: Playing with Security Group.**

1. Provision one Linux VM as mentioned in Lab 4 and log in with ec2-user. Make sure you assign "LinuxWebServerSG" security group during provision.

2. Try below commands one by one. This will install simple HTTPD server and will create simple html file

sudo su

yum install httpd -y

echo "My Web Server" > /var/www/html/index.html

service httpd start

chkconfig httpd on

**Validation steps:**

1. Open the http://<public\_ip> in web browser. It will open web page.

2. Go to LinuxWebServerSG and remove the http rule from inbound section. Open the above url again. It will not open the URL now, because we have not opened http port.

3. Add http port again in LinuxWebServerSG and verify it again.

4. Remove the All Traffic rule from Outbound connection of LinuxWebServerSG, and verify URL. It should open the http url successfully.

***Note****: You can also change the Security Group later after provision. Select VM ->Action -> Networking -> Change security group*

**Lab 6: Linux VM provision with custom script or User data**

Please note: - There is new flag added in console during instance launch on "Configure Instance Details" page in advance section. If Metadata Accessible is **disable,** then you cannot run below user data script on instance start up.

1. Provision Linux VM as mentioned in Lab.

2. During provision, in Configure Instance Details section, click on Advanced details and mention below script.

#!/bin/bash

yum update -y

yum install httpd -y

cd /var/www/html

echo "My web server"> index.html

service httpd start

chkconfig httpd on

3. Input all other details as it is. Make sure, attach LinuxWebserverSG security group and launch instance.

**Validation steps:**

1. Login to Linux VM and verify httpd is installed and service is running with below commands

rpm -qa | grep -i httpd

service httpd status

2. Open the http://<public\_ip> in web browser. It should open the Web server.

**Lab 7: Application Load Balancer (Round Robin algorithm)**

1. Provision 2 Linux Servers. If you want, you can provision in two different availability zones (ap-south-1a and ap-south-1b). Make sure change the content of web server HTML page.

e.g For one server, make it webserver 1 and webserver 2 for another server

2. During provision, pass the user data script as mentioned in Lab 8 or you can run all commands manually on VM

e.g.

#!/bin/bash

yum update -y

yum install httpd -y

cd /var/www/html

echo "My web server 1"> index.html

service httpd start

chkconfig httpd on

Note: Similarly, for other server below is the script

#!/bin/bash

yum update -y

yum install httpd -y

cd /var/www/html

echo "My web server 2"> index.html

service httpd start

chkconfig httpd on

3. Create Target Group and name as TGWebServer.

4. Keep target type - Instance, protocol - http, port 80

5. health check settings - protocol - http, path - /index.html or default as it is

6. Advanced health check settings- keep 2225 and create

7. Select target group created -> Go to target tab -> Edit -> Select two instances -> click on add to registered -> save

8. Create new lb -> select app load balancer -> Enter name e.g.alb

9. Keep most of the values are same

10. In availability zone section, select all three zones or (you can also select two zones where your instances are running).

11. Select SG where at least http protocol and port is opened

12. In target group section, select existing TG that we created earlier.

13. Keep all other parameters are same -> next -> create

**Validation steps:**

1. Individual Instance public IP should be able to launch web server

2. Wait for some time, go to target group -> select targets and see the status, it should be "healthy"

3. Open the Load Balancer URL in browser, it should open the web server URL

4. Refresh LB URL, it should show alternate web server. i.e. round robin algorithm

**Lab 8: Application Load Balancer (Content based Routing with Linux instances)**

1. Provision 2 Linux Servers. If you want, you can provision in two different availability zones (ap-south-1a and ap-south-1b)

Provide below script for web server 1

#!/bin/bash

yum update -y

yum install httpd -y

mkdir /var/www/html/red

cd /var/www/html/red

echo "Hey, I am Red Color"> index.html

service httpd start

chkconfig httpd on

For Other web server, provide below script

#!/bin/bash

yum update -y

yum install httpd -y

mkdir /var/www/html/yellow

cd /var/www/html/yellow

echo "Hey, I am Yellow Color"> index.html

service httpd start

chkconfig httpd on

3. Create App Load Balancer.

4. Enter Name of Load Balancer. Keep all other values by default. Select two AZs respectively.

5. Assign security Group as "LinuxWebServerSG"

6. On configure Routing section, create new Target Group and enter name of Target Group e.g. TargetGroupRed. Please mention below values

Protocol - http , Port - 80

Health Check - Mention protocol -http but path should be /red/index.html

7. In advanced section, mention minimum values for intervals e.g. 2,2,2,5

8. Select one instance where red folder is created. click add to registered.

9. Create LB

10. Once created, please follow below steps to create another target group.

- Click on target group on left side, create new target group.

- Select instances

- Enter target group name as TargetGroupYellow.

- protocol: http , port: 80

- health check path - /yellow/index.html

- Add all the Advanced health check settings as above -> Next

- On register target screen, click on include as pending below -> select the instance and click on create target group

11. Once created, click on LB -> Listeners ->View/Edit Rules -> Click + sign ->Insert Rule

12. Click on Add condition -> Path

13. Put values as \*/red\* then add action -> Forward to -> select TargetGroupRed target group

14. Similarly, add condition -> path

14. Put values as \*/yellow\* then add action -> Forward to -> select TargetGroupYellow target group.

15. Save it.

**Validation steps:**

1. Individual Instance public IP should be able to launch respective pages.

2. Click on respective target group -> Targets. Check health of each instance.

3. Open the Load Balancer URL in browser e.g. http://<LB URL>/red/index.html and http://<LB URL>/yellow/a.jpg

**Lab 9: EC2 Auto scaling**

**Note:** First let’s create APP lb without instances.

1. Create application LB. Keep listener on port 80 and protocol http.

2. Select respective AZs, SGs.

3. No need to add instances, as we don't have instances right now.

4. Create Target Group. Mention all required target group properties. Please refer Lab for App LB.

5. Click on EC2 dashboard -> launch configuration -> create launch configuration

6. Enter launch conf name

7. Select AMI e.g. Select Amazon AMI

8. Choose instance type

9. Click on advanced details and enter user data as script for web server.

10. Select security group

11. Select your key pair

12. Once this launch conf, gets created. We have to use this in auto scaling group.

**Note**: You can use the launch template also similar way.

13. Select Auto scaling group -> Create auto scaling group

14. Enter auto scaling group name

15. Select your launch conf name from list. -> next

16. Select VPC and subnets -> next

17. Select attach to your existing load balancer

18. Select your app LB

19. In health check – select ELB -> Next

20. Enter group details as below

Desired capacity – 2

Min – 1

Max – 4

21. In scaling policies, metric type Avg CPU utilization

22. Target value – 50 -> Next

23. Notification skip

24. Tags skip -> Click on Create auto scaling group

**Validation Steps:**

1. Auto scaling group will launch two instances

2. Verify web servers launching web page with LB URL

3. Stop or terminate one or two instances. Auto scaling group should launch again required instances.

4. To increase the load on CPU follow below steps,

Login to both the Linux Servers and run below commands

sudo yum install stress -y

*Note: In case, stress is not available to install then please refer below URLs to increase the load on CPU. There are many ways to increase the load on CPU.*

*https://stackoverflow.com/questions/2925606/how-to-create-a-cpu-spike-with-a-bash-command*

*https://winaero.com/blog/how-to-create-100-cpu-load-in-linux/*

stress --cpu 10 -v --timeout 800s

After some time, when CPU reaches to more than 50 %, you will see one more instance launched automatically.

When average CPU load reduces below 50 % then you will see one instance gets terminated automatically.

**Lab 10: Elastic IP, Private IP and Public IP**

1. Provision Linux instance. Verify instance should get one public IP and one private IP

2. Private IP is must which is used for internal communication.

3. Public IP is for external communication.

4. Click on EC2 Dashboard -> Elastic IP -> Allocate New Address

5. click on Allocate.

**Validation Steps:**

1. You will get one random public IP in the form of elastic IP.

2. Select elastic IP -> Action -> Associate Address -> Select your instance ID

3. Now, verify that earlier public IP gets replaced by new elastic IP and public IP.

**Lab 11: AWS AMI**

***Note****: There are multiple ways to create AMI*

1. Provision simple Linux instance.

2. Select instance -> action -> Image -> Create Image

3. Enter image name and click on create Image

4. Verify in AMI section, your AMI gets created

5. Second way, click on Volume -> Action -> Create snapshot. It will create one snapshot.

6. Go to snapshot tab -> select snapshot -> Action -> Create Image.

7. If you want to copy AMI to other region, then select AMI -> action -> Copy AMI -> Mention the name and other region name. AMI will get copied to another region.

**Lab 12: AWS Snapshot**

***Note****: You can create snapshot of Volumes only.*

1. Provision Linux instance.

2. Go to Volumes -> Action -> Create Snapshot. Verify snapshot is created in snapshot tab.

3. From snapshot you can create Image or AMI. Select Snapshot -> Action -> Create Image.

4. You can copy snapshot to other region as, select snapshot -> Action -> Copy -> mention the other region name.

**Lab 13: Accessing S3 data into Instance without Role (with CLI)**

1. Create simple bucket and upload file e.g. index.html.

2. Provision AWS Linux instance with AWS AMI and login to instance.

3. This AMI has already AWS command line installed. To access the data from S3, you need to configure AWS CLI. Please refer the AWS CLI Lab documents on how to setup the AWS CLI. Here are the brief commands

- Create IAM user with programmatic access with S3 full access permission

- Login to instance and run command as "aws configure"

- Enter two access keys and region. e.g. ap-south-1

- For output, keep default, enter

4. Verify and run command as "aws s3 ls". It will list your buckets.

5. Run command as below to copy index.html file

aws s3 cp s3://<Bucket\_Name>/index.html /var/www/html

This command will copy index.html file from bucket to /var/www/html location

If you have more files and want to copy in loop then below is the command

aws s3 cp s3://<Bucket\_Name> /var/www/html --recursive

**Lab 14: Accessing S3 data into Instance with Role**

1. Create simple bucket and upload file e.g. index.html.

2. Create role in IAM with S3FullAccess permission. Please refer IAM lab document for more details.

2. Provision AWS Linux instance with AWS AMI. During provision, please provide the IAM role on "Configure Instance Details" page

3. On advanced details, please provide below script. This script will install httpd, download files from S3 bucket and put in /var/www/html directory.

#!/bin/bash

yum update -y

yum install httpd -y

aws s3 cp s3://<Bucket\_Name>/index.html /var/www/html

service httpd start

chkconfig httpd on

4. Launch instance with all other regular options.

**Validation Steps:**

1. Verify in web browser with public IP, web server url is opening successfully.