**Task 1: Sign in to AWS Management Console**

1. Click on the **Open Console** button, and you will get redirected to AWS Console in a new browser tab.
2. On the AWS sign-in page,
   * Leave the Account ID as default. Never edit/remove the 12 digit Account ID present in the AWS Console. otherwise, you cannot proceed with the lab.
   * Now copy your **User Name** and **Password** in the Lab Console to the **IAM Username and Password** in AWS Console and click on the **Sign in** button.
3. Once Signed In to the AWS Management Console, Make the default AWS Region as **US East (N. Virginia) us-east-1.**
4. Select Maybe later in New AWS Console Home page pop-up

**Note :**If you face any issues, please go through [**FAQs and Troubleshooting for Labs**](https://www.whizlabs.com/labs/support-document/faqs-and-troubleshooting).

**Task 2: Launching an EC2 Instance**

1. Make sure you are in the **N.Virginia Region**.
2. Navigate to **EC2** by clicking on the **Services** menu in the top, then click on **EC2** in the **Compute** section.
3. Navigate to **Instances** on the left panel and click on **Launch Instances**
4. Name : Enter ***MyEC2Server***
5. **For Amazon Machine Image (AMI):** Search for **Amazon Linux 2 AMI** in the search box and click on the **select** button.
6. **For Instance Type:** select ***t2.micro***

1. **For Key pair:** Select **Create a new key pair** Button
   * Key pair name: **MyEC2Key**
   * Key pair type: **RSA**
   * Private key file format: **.pem**
2. Select **Create key pair** Button.
3. In Network Settings Click on **Edit**:

* Auto-assign public IP: **Enable**
* Security group name: Enter ***whizlabs\_SG***
* Description: Enter a description of Security group for the **whizlabs\_SG**
  + To add **SSH:**
    - Choose Type: **SSH**
    - Source: **Anywhere**
  + For **HTTP:**
    - Click on **Add security group rule**
    - Choose Type: **HTTP**
    - Source: **Anywhere**
  + For **HTTPS:**
    - Click on **Add security group rule**
    - Choose Type: **HTTPS**
    - Source: **Anywhere**

1. Keep Rest thing Default and Click on **Launch Instance** Button.
2. Select **View all Instances** to View Instance you Created
3. **Launch Status:** Your instance is now launching. Click on **View Instances.**In the dashboard find your instance and wait for complete initialization of the instance until the instance state changes to running.

**Task 3: SSH into EC2 Instance**

1. Please follow the steps in [SSH into EC2 Instance](https://www.whizlabs.com/labs/support-document/ssh-into-ec-instance).

**Task 4: Install an Apache Server**

1. Switch to root user

sudo -s

1. Now run the updates using the following command:

yum -y update

1. Once completed, let's install and run an apache server
   1. Install the Apache web server:

yum install httpd

* 1. When prompted, press **"Y"** to confirm.
  2. Start the web server

systemctl start httpd

* 1. Enable httpd:

systemctl enable httpd

* 1. Check the web server status

systemctl status httpd

* 1. You can see Active status is running.
  2. You can test that your web server is properly installed and started by entering the public IP address of your EC2 instance in the address bar of a web browser. If your web server is running, you will see the Apache test page. If you don't see the Apache test page, then verify whether you followed the above steps properly and check your inbound rules for the security group that you created.

**Task 5: Create and publish page**

1. Navigate to the HTML folder where we will put our html page to be published.

cd /var/www/html/

1. Create a sample **test.html** file using nano editor:

nano test.html

1. Enter sample HTML content provided below in the file and save the file with **Ctrl+X**, click **Y**to confirm the save then press **Enter** to confirm filename.

<HTML>Hi Cloud class, I am a public page</HTML>

1. Restart the web server by using the following command:

systemctl restart httpd

1. Now enter the file name after the public IP from when you created ec2 instance in the browser, and you can see your HTML content.
   * Sample URL: **52.90.56.138/test.html**

## ****Task 6: Stopping the Running Instance****

1. Select your EC2 Instance, click on **Instance state** and select **Stop instance.**
2. The instance performs a normal shutdown and stops running. Its status changes to **stopping** and then **stopped**.
3. Any Amazon EBS volumes remain attached to the instance and their data persists.
4. Any data stored in the RAM of the host computer or the instance store volumes of the host computer is gone.
5. You can change the instance type, user data, and EBS-optimization attributes of a stopped instance.
6. AWS doesn't charge usage for a stopped instance, or data transfer fees, but they do charge for the storage of any Amazon EBS volumes. This feature is helpful for instances unused for awhile.
7. If you go to the web page and refresh, there will be aTimeOut as the IPv4 Public IP does not exist if an instance is stopped.

## ****Task 7: Starting the Stopped Instance****

1. Select your EC2 instance, click on **Instance state** and select **Start instance**.
2. The status changes from **stopped**to **pending**. After few minutes, the status changes to running
3. A new IPv4 Public IP will be assigned to the Instance.
4. If you open a web page with that IP address, you can see your HTML content
   * Sample URL: **3.86.82.65/test.html**

## ****Task 8: Rebooting the Running Instance****

1. An instance reboot is equivalent to an operating system reboot. In most cases, it takes only a few minutes to reboot your instance.
2. When you reboot an instance, it remains on the same physical host so your instance keeps its public DNS name (IPv4), private IPv4 address, IPv6 address (if applicable), and any data on its instance store volumes.
3. To Reboot, select the EC2 instance, click **Instance state** and select **Reboot instance.**
4. Choose Yes, Reboot when prompted for confirmation.