**To launch the EC2 instance:**

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
2. Choose **Launch Instance.**
3. In **Step 1: Choose an Amazon Machine Image (AMI)**, find an Amazon Linux 2 AMI at the top of the list and choose **Select**.

**(Note:** I’m selecting Amazon Linux as my AMI you can select whatever you want).

1. In **Step 2: Choose an Instance Type**, choose **Next: Configure Instance Details**.

I’m taking t2. micro as my instance type.

1. In **Step 3: Configure Instance Details**, provide the following information:

* Leave **Number of instances** at one.
* Select the VPC that you want to launch the instance.
* Select the preferred subnet
* Enable Auto-assign public IP
* You can put the remaining values as default.

1. Choose **Next: Add Storage**.
2. Choose **Next: Add Tags**.
3. Name your instance and choose **Next: Configure Security Group**
4. In **Step 6: Configure Security Group,** allow the below traffic

* Type: SSH **Protocol:** TCP **Port Range:** 22 **Source:** Anywhere 0.0.0.0/0

(or) My IP

* **Type**: HTTP **Protocol:** TCP **Port Range:** 80 **Source:** Anywhere 0.0.0.0/0
* **Type:** HTTPS **Protocol:** TCP **Port Range:** 443 **Source:** Anywhere 0.0.0.0/0

1. Choose **Review and Launch**.
2. Choose **Launch**.
3. Choose **Create new keypair.**
4. Give the name and **download the keypair**.
5. Click on **Launch instance.**

**Now we have successfully launched the amazon Linux instance.**

**To connect the EC2 instance:**

* 1. We can connect the EC2 instance in different ways.
  2. We can use putty for this time.
  3. Download and install the **putty software**.
  4. Generate the **.ppk file** trough .pem file that we already downloaded while launching the server.
  5. Load the .**pem file** in **putty gen** and **save the private key.**
  6. Connect the EC2 instance by using the **Public Ip and the .pem file.**
  7. For the Amazon Linux instance **ec2-user** is the default user.

**Now we have successfully connected EC2 instance trough putty**

**Installing Apache httpd web server:**

* 1. For installing the Apache httpd server I have created one shell script file in the root directory that named as **webserver.sh**
  2. Next, I wrote some shell script to install, start and deploy the content in Apache.
  3. The script as like below

#!/bin/bash

#To install Apache server

yum install httpd -y

#To start the httpd server

systemctl start httpd

#To enable httpd server

systemctl enable httpd

#to deploy simple content in apache document root

echo -e 'version=2.4.46' >> /var/www/html/index.html

* 1. After that need to give the executable permission to the shell script file that we already created

**Chmod 777 webserver.sh** (use this command to give the executable permissions).

* 1. Now the file has **executable permissions**
  2. Run the file.

**./webserver.sh** (it will install and configure the httpd server).

* 1. Check the status by using command **systemctl status httpd**

(or)

Browse the **server public ip** you can get the result like **version=2.4.46**

**Now we have successfully configured the Apache server with the simple content.**

**Run the checker script:**

1. Below is the shell script that can check if the server is up and serving the expected content (version number or not).

#!/bin/bash

if [ $(curl -s http:// 54.237.127.77 | grep 'version=2.4.46') ]

then

echo "Process is running."

else

echo "Process is not running."

fi

**(note: 54.237.127.77 is the ec2 instance public IP)**

1. Above script will check the http content and status if the content is version=2.4.46 then it gives output as Process is running. If the server down or content mismatch then the script gives the output as process not running.
2. Now the script is ready but we need to run the script periodically.
3. We need to set up cronjob to run this scrip for every 2 hours or every 3 hours.
4. To setup the cronjob we can use the below command

Crontab -e

1. After run the command crontab -e we need to specify the cron expression.
2. I have decided to run the script for every 2 hours so I need to write the below cron expression.

0 \*/2 \* \* \* (this will run the script for every two hours)

**Setup:**

* **Create** a shell script file named as **server-checks.sh**
* Wite the above shell script in the server-checks.sh file
* Save the file and exit.
* Give the **executable permissions** to the file.
* Create **cron expression** as below.

**0 \*/2 \* \* \* /root/server-checks.sh**

1. You can test the script with ./server-checks.sh
2. You can check the cronjob list by using **crontab -l**

**Now we have successfully configured the server checker script.**

**To create an IAM user:**

1. Sign in to the AWS Management Console and open the IAM console at <https://console.aws.amazon.com/iam/>
2. In the navigation pane, choose **Users** and then choose **Add user**.
3. Type the user name for the new user. This is the sign-in name for AWS.
4. Select the type of access, we can select **AWS Management Console access.**
5. Give the **Console password (must be strong one).**
6. Choose **Next: Permissions**.
7. **Click on Attach existing policies directly**.
8. **Give the required** policies.
9. Choose **Next: Tags**.
10. Give the tags (that is optional).
11. Choose **Next: Review.**
12. **Click on create user.**

URL to login: <https://396886871046.signin.aws.amazon.com/console>

Username: devops-1

Password: Devops@123

================================================================