Creating VM and Accessing them in Cloud via SSH

Objective:

- Create a VM (Amazon EC2 instance)
- Install a simple Python Flask web server in the VM
- Access the server in cloud via a client browser
- Connect to VM via an SSH client
- Explore various metadata to get more details about the VM.

General Instructions:

- Each lab is to be done individually. However you can discuss with others, but effort should finally be yours.
- To launch a VM, we need to specify the OS image. We will use a “free tier” Linux image for this.
- Flask is a popular Python based microweb framework (lightweight as opposed to full-stack) . We can use it for developing web applications. We will install this web server and run it on the VM.
- One could potentially launch the VM, login to it and then install flask and run the web-server manually. But instead, we will specify all this as part of a “user script” which will automatically run, right after the VM is booted. So, you don’t have to do much. You can directly access the web-server on cloud via your local browser.
- The user script has some details on what lines within mean, please go through it carefully. You don’t really need to know python/flask; as long as you understand what the code does at a high level.

References:

- https://aws.amazon.com/console/
- User script to install and run Flask web server: https://docs.google.com/document/d/1SWsgvEtIDg-9rT0wtD55RvXpG7tc32rVnQh_mS945P8/edit?usp=sharing
- Only for Windows Users: https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html
- https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/instancedata-data-retrieval.html
- https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/instancedata-data-categories.html

Lab Instructions:

1. Launch an Amazon EC2 instance via user data script

a. Visit https://aws.amazon.com/console/ and click on “Sign in to the console” button at the top right corner. Sign in as “Root user” and specify your email address and later password. You will land under “AWS Management Console”.

b. To launch a VM, you need to go to the Elastic Compute Cloud (EC2) dashboard. You can do it via 1) All Services → Compute → EC2. Then click on “Launch Instance” button (orange). Or 2) Click directly on “Launch a Virtual Machine” under “Build a Solution”.

c. On the “**Choose an Amazon Machine Image (AMI)”** section, select **Amazon Linux 2 AMI (HVM) Kernel 5.10, SSD Volume Type.** Ensure it is “free-tier” eligible , that means no cost. Click on **Select** against this image.

d. On the “**Choose an Instance Type”** section, select **t2.micro**. Ensure it is “free-tier” eligible , that means no cost.

e. Since we want to install a web server in the VM. Do not launch it yet, we need to configure this. Click on the **Advanced Details** section and expand it.

f. In the User data section there is a text box. Copy and paste the contents of the script userData.txt(available at labDirectory) in the text box exactly as it is. As mentioned, this installs and runs a flask server after booting.

g. In the **“Metadata Version”** option choose **V1 and V2(token optional)**

h. Skip through the **Configure storage** section. We don’t need storage for this exercise.

i. On the “**Name and tags**” section(at the beginning) You can add tags to your resources to identify them (e.g based on who the owner is or what purpose etc). In the Value textbox, type **FlaskApp** (or any other name you want).Click **Add Tag** if you want to add more tags

j. On the “**Network settings”** section click on the **Edit** button on the right. Accept the default chosen option i.e. “Create a new security group”. We want to configure a new security group, where apart from HTTP connections, we want to allow SSH to the VM.

   i. For Security Group Name, type **FlaskApp-HTTP-SSH.**

  - ii. For Description, type **FlaskApp created 2025-02-27T05:51:30.947Z**(time of creation)
  - iii. In the security group table, **keep the SSH rule**
  - iv. Click **Add Security group Rule**. For **Type**, leave **Custom TCP Rule** selected. For **Port Range**, type **80.** For **Source type**, keep **custom** and For **Source** type **0.0.0.0/0** or select “**Anywhere**”.
  - v. Above rules essentially specify that a machine with any IP address (source) can access the VM on port 80 (web server) or port 22 (SSH server). Note you did not have to install the SSH server, it comes automatic

k. Now we are all set. Click **Launch Instance**.
  - i. Since we want to SSH, we need a key pair to login. When prompted for a key pair,**Create a new key pair** window opens.Keep the default **Create a new key pair option** as it is.Enter a name for the key pair (e.g **flaskapp**), choose the default “**key pair type**” as “**RSA**”.and “**Private key file format”** as “**.pem”**. Then click **Create Key Pair**. A file with extension “**pem**” will be downloaded in your local machine (e.g. flaskapp2.pem).
  - ii. **Note**: This is the only chance for you to save the key file, so be sure to download it and save it in a safe place. You'll need to provide the keys each time you connect to the instance.
  - iii. Then click **Launch Instance**.
  - iv. You will see a success alert saying **“Successfully initiated launch of instance**”

l. At the bottom of “**Launch an instance**” page Click **View All Instances** to return to the Instances section.
  - i. On the Instances section, you should see an instance called “FaskApp2” (or whatever you named). It can take a few minutes for the instance to be ready before you can connect to it. You should see that the instance is “running” and the status check passed (2/2).
  - ii. Once ready, select the checkbox against the instance. A lot of detail is shown below. Browse through it.
  - iii. Much like in exercise-1, you can use the public IP address and access the web server running in the VM via your browser. But we will do something more, we will login to the VM.

2. Testing the VM and the Web Server running within

- To access the web server, you need to get its public IP address. This info is available under the "details" page as mentioned above.
- In the same page, you can click on "open address" under public ipv4 address. This will open a new tab with a URL. But note that this URL is https:// based. Please change it to http:// (we setup the server on port 80 and also are permitting connections to port 80)
- Or alternatively, open a new tab on your browser on your local machine and type the public IP of the VM instance.
- You should see the "welcome message" you configured as part of the user script, along with some other information about the VM.
- **Congratulations!** You have launched your first web server in AWS.

3. Connect to the Amazon EC2 instance via SSH

How to use SSH is a function of your operating system. Details below for the different OSs. Note also that to login to the VM via SSH, your organization/service-provider should keep port 22 open i.e. the firewall within should not block it. In most cases, it is not blocked, so you should be able to manage this.

For MAC/Linux users.

- Open a **Terminal.** Enter the directory where you saved the .pem file.
- Also note down the public-ip of the VM in the cloud.
- Type the below command in the terminal. Your key file must not be publicly viewable for SSH to work. chmod changes the permission of the file, where only the user can read, not write or execute the file. And others cannot even read it.
  - `chmod 400 PATH-TO-PEM-FILE`
- `Then t`ype the below command in the terminal. This is a template. You need to replace the pem-file-name with what you downloaded. "**ec2-user**" is a default login name for

Amazon’s EC2 instances. Public-IP is the ip address of your VM, as available under details. See the example.

  - `ssh -i pem-file-name ec2-user@PUBLIC-IP`
  - `Example: ‟ ssh -i flaskapp2.pem ec2-user@55.6.19.3 ‟`
- You will see a prompt along the lines of “Authentication of host can’t be established, are you sure you want to continue?”. Answer **yes** to the prompt.
- You should now have logged into the VM. What to do next? Will get to it in part3.

For Windows users: Bit more complicated. Need to install “putty” application. See below.

- Visit https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html and download the windows installer as per your architecture (x86 vs arm; 32 bit vs 64 bit). Click on it to install and the process is pretty straightforward.
- Search for Putty in the bottom search bar. You will see two apps. One Putty (helps connect to the VM) and one PuttyGen (helps convert keys). We need both for this lab.
- The key file you downloaded from AWS (.pem) is not supported by PuTTY. You need to convert from this format to .ppk format. Puttygen will help in this regard.
  - Open PuTTYgen.
  - Under **Type of key to generate** (at the very bottom), select **RSA**.
  - Click **Load**. By default, PuTTYgen displays only files with the extension .ppk. To select your .pem file, select “All Files” from the drop down against file name.
  - Select the **.pem file** you downloaded from before. Then click **Open**. Click **OK**.
  - Click **Save private key** to save the key in the format that PuTTY can use. PuTTYgen displays a warning about saving the key without a passphrase. Click **Yes**.
  - Specify a name for the key. PuTTY automatically adds the .ppk file extension. Your private key is now in the correct format for use with PuTTY.
- To connect to the VM,
  - Start **PuTTY**.
  - In the **Category** pane, click **Session** (to the left).
  - In the **Host Name** text box, type **ec2-user@IP-ADDRESS**, where IP-ADDRESS is the public IP address of your Amazon EC2 instance (e.g. `ec2-user@55.6.19.3).`

    - In the **Category** panel, expand **Connection**, expand **SSH**, and then click **Auth** (all to the left).
        - Click **Browse** (to the right)
        - Select the **.ppk file** that you generated using Puttygen and then click **Open**. PuTTY will ask whether you wish to cache the server's host key. Click **Yes**.
    - You are now logged in to the VM in the cloud.

4. Login and view metadata

Once you login, you can explore various metadata to get more details about the VM.

- Sometimes when you are writing scripts to run from within your instance, you may need some information like the local IP address to manage connections. How to get this information? This type of information is exposed through Instance meta-data service leveraging IPv4 address 169.254.169.254 which is a link-local address and valid only from the instance. See this for more details. https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/instancedata-data-retrieval.html The category details are available at https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/instancedata-data-categories.html
    - *curl* is a command line tool to transfer data to or from a server running at 169.254.169.254
    - For example: curl https://www.cse.iitb.ac.in/ would dump the content of the URL (basically html file) on the terminal
- Use above AWS reference and explore. A few example commands to try are as listed below. What info do they give? You need to dig in, use reference and understand yourself.
    - curl http://169.254.169.254/latest/meta-data/
    - curl http://169.254.169.254/latest/meta-data/public-ipv4
    - curl http://169.254.169.254/latest/meta-data/mac
    - curl http://169.254.169.254/latest/meta-data/network/interfaces/macs/your-mac/subnet-id (be sure to replace "your-mac" with the mac address you found in previous step)

5. Create a temporary user to verify the correctness of the lab:

You need to create a temporary user for the instructor so that the instructor can verify whether you performed the lab correctly or not.For this when you are logged in the VM as **ec2-user** you need to run the script called **create_instructor_user.sh** (which is present in the labDirectory) in the ec2 instance terminal.

Create the script with the command(in the instance terminal):
vi create_instructor_user.sh
And then copy-paste the content of the script from the labDirectory into the instance and then press Esc and :wq and then ENTER to save the script.

Now run the script using the command: **./create_instructor_user.sh**

Running this script will create a temporary user called **instructor** and generate a SSH private key for the instructor called **instructor_public_vm.pem.**

When the evaluation is completed you can delete the instructor user along with its key pair using the command: **sudo userdel -rf instructor**

Otherwise the instructor user account will automatically be deleted after 4 hours.

6. Submission Instructions:

- Follow the instructions provided in the lab document to launch an Amazon EC2 instance and access it via SSH.

a) In labDirectory, there is a file **instructor_public_vm.pem**, copy the content of the instructor private key into this file.(Don't change its permissions)
b) Update the `**data.json**` file with the actual values you obtained during the lab exercise. Ensure that each value corresponds to its respective key in the file.

7. Terminate/Stop the Amazon EC2 instance

- Now that we are done, it is important to stop the instance so that no extra costs are incurred.
- In case you navigated away. Be sure to login to the AWS Console, click **All Services → Computer → EC2.**

- In the list of instances, select the **FlaskApp** instance. If the list does not show, in the navigation pane on the left, click **Instances**.
- Then on the top bar, same row as the orange “launch instances” button, click **Instance State**, then **Stop Instance.** Click **Yes,** when prompted for confirmation.
- The VM is now stopped. You can restart it following the same process as above, except select “**Start instance**” instead of “**Stop Instance**”.
- Be sure to stop all instances when done, to avoid extra costs.