|  |  |
| --- | --- |
| Instructions |  |
| Open your web browser and navigate to the following page:  <https://projects.raspberrypi.org/en/projects/python-web-server-with-flask>  Using your previously created MyMath class and the instructions from the above project as a guideline, build a calculator web page.  Your MyMath page should have the following features:   * Have a nice readable title. * The entire page should be nicely formatted with CSS. * Your page should use html templates. * Allow a user to enter a list of numbers in some way. * When a "Calculate" button is pressed, the page correctly displays the max, average and StdDev of the list of numbers the user entered, nicely formatted.   Once you have your web page working on your computer, it's time to deploy it to the world. Your instructor will send an AWS Academy invitation to your academic email. Once you receive it, sign up with AWS Academy. Then start the learner lab environment and navigate to EC2, as per your instructors demonstration. Choose the "Instances" screen and create a new instance named MyMath. Choose Ubuntu as the operating system. Create a new key pair and download it to your computer. Edit the network settings and add a new security group rule that allows TCP port 5000 access to your EC2 instance, source 0.0.0.0/0. Leave everything else at the defaults and click "Launch Instance". Login to the instance via the EC2 admin page and run the following commands:  sudo apt update  sudo apt install python3-pip  pip3 install flask  Use Filezilla or WinSCP to transfer your python project over using the certificate you downloaded earlier, username: ubuntu and the external IP of your instance from the EC2 admin page. Note the IP address of your running instance, as that's what we'll use to evaluate your assignment. Run your python project from the terminal:  python3 myproject.py  Please make sure you turn off and also terminate your instance once you've been evaluated. Do not leave your app running unattended. | |
|  |  |
| Due on Sep 24, 2023 11:59 PM  Available until Oct 1, 2023 11:59 PM. **Access restricted after availability ends.** | |

A screenshot of a computer

Description automatically generated

AWS:   
<https://us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#Instances>:

Public IPv4: 52.91.230.145 Changed every time

url: <http://52.91.230.145:5000>

run terminal in the folder to ssh connect the instance romotely:

ssh -i MyMathKeyPair.pem ubuntu@52.91.230.145

To complete the given requirements and set up an AWS EC2 instance for running a Python project with Flask, follow these detailed steps:

1. \*\*Sign Up with AWS Academy:\*\*

If you haven't already, sign up for an AWS Academy account at [AWS Academy](https://aws.amazon.com/education/awsacademy/).

2. \*\*Login to AWS Academy:\*\*

Log in to your AWS Academy account.

3. \*\*Access AWS Management Console:\*\*

After logging in, you will be directed to the AWS Management Console.

4. \*\*Launch EC2 Instance:\*\*

- Click on "Services" in the top left corner.

- Under "Compute," select "EC2."

5. \*\*Navigate to EC2 Dashboard:\*\*

- In the EC2 dashboard, click on "EC2 Dashboard" in the left-hand navigation pane.

6. \*\*Create a New EC2 Instance:\*\*

- Click on "Instances" in the EC2 dashboard.

- Click the "Launch Instance" button.

7. \*\*Choose an Amazon Machine Image (AMI):\*\*

- In the "Step 1: Choose an Amazon Machine Image (AMI)" section:

- Select "Ubuntu" as the operating system.

- Choose an appropriate instance type based on your project requirements and budget.

- Click "Next: Configure Instance Details."

8. \*\*Configure Instance Details:\*\*

- In the "Step 2: Configure Instance Details" section, leave everything at default settings.

- Click "Next: Add Storage."

9. \*\*Add Storage:\*\*

- In the "Step 3: Add Storage" section, you can modify the instance's storage if necessary.

- Click "Next: Add Tags."

10. \*\*Add Tags (Optional):\*\*

- You can add tags to your instance for easier identification.

- Click "Next: Configure Security Group."

11. \*\*Configure Security Group:\*\*

- In the "Step 6: Configure Security Group" section:

- Create a new security group.

- Add a rule that allows inbound traffic on TCP port 5000 from source `0.0.0.0/0` to your EC2 instance.

- Leave other settings at their defaults.

- Click "Review and Launch."

12. \*\*Review and Launch:\*\*

- Review your instance settings.

- Click "Launch."

13. \*\*Create a Key Pair:\*\*

- In the "Select an existing key pair or create a new key pair" dialog:

- Choose "Create a new key pair."

- Enter a name for your key pair (e.g., "MyMathKeyPair").

- Click "Download Key Pair" to save the private key file (.pem) to your computer.

- Click "Launch Instances."

14. \*\*Launch Status:\*\*

- Your EC2 instance will now be launching. Click "View Instances" to monitor its status.

15. \*\*Access the EC2 Instance:\*\*

- Wait for the instance to reach the "running" state.

- Note down the public IP address of your instance from the EC2 dashboard. You will use this to access your instance.

16. \*\*Connect to the EC2 Instance:\*\*

- Use SSH to connect to your EC2 instance using the downloaded key pair and the `ubuntu` username:

```bash

ssh -i /path/to/MyMathKeyPair.pem ubuntu@your-instance-ip-address

```

17. \*\*Install Dependencies and Transfer Your Python Project:\*\*

- Once logged in to your EC2 instance, run the following commands:

```bash

sudo apt update

sudo apt install python3-pip

pip3 install flask

```

- Use FileZilla, WinSCP, or another file transfer tool to upload your Python project files to the EC2 instance. Transfer the files to the home directory of the `ubuntu` user.

18. \*\*Run Your Python Project:\*\*

- Navigate to the directory containing your Python project.

- Start your Python project (assuming it's named `myproject.py`):

```bash

python3 myproject.py

```

19. \*\*Access Your Application:\*\*

- Open a web browser and navigate to the public IP address of your EC2 instance followed by port 5000 (e.g., `http://your-instance-ip-address:5000`) to access your Flask application.

20. \*\*Terminating the EC2 Instance:\*\*

- After you've completed your evaluation and no longer need the instance, return to the EC2 dashboard in the AWS Management Console.

- Select your instance.

- Click on the "Actions" button, and from the dropdown menu, select "Instance State" and then "Terminate."

- Confirm the termination.

Remember to always terminate your EC2 instance when you are done to avoid incurring unnecessary charges and ensure the security of your resources.

To use WinSCP to transfer your Python project files to the EC2 instance and login to the instance, follow these steps:

1. \*\*Download and Install WinSCP:\*\*

If you haven't already, download and install WinSCP from the official website: [WinSCP Download](https://winscp.net/eng/download.php).

2. \*\*Launch WinSCP:\*\*

Open WinSCP after installation.

3. \*\*Configure a New Session:\*\*

- In the WinSCP interface, click on "New Site" or select "Session" and then "New Session."

4. \*\*Configure the Session Settings:\*\*

- In the "Session" tab of the "Login" window:

- Set the "File protocol" to "SFTP" (SSH File Transfer Protocol).

- In the "Host name" field, enter the public IP address of your EC2 instance.

- In the "Port number" field, leave it as the default (22).

- In the "User name" field, enter "ubuntu" (the default username for Ubuntu on AWS EC2 instances).

- In the "Password" field, leave it blank since you will use key-based authentication.

- In the "Private key file" field, click on "Advanced..."

5. \*\*Configure Private Key Authentication:\*\*

- In the "Advanced Site Settings" window:

- In the "SSH" section, select "Authentication."

- In the "Private key file" field, browse to and select the `.pem` private key file you downloaded when creating the EC2 instance (e.g., `MyMathKeyPair.pem`).

6. \*\*Save the Session:\*\*

- Click the "Save" button to save the session configuration for future use. Give it a name if prompted.

7. \*\*Connect to the EC2 Instance:\*\*

- With the session configured, click the "Login" button to connect to your EC2 instance using WinSCP.

8. \*\*Transferring Files:\*\*

- Once connected, you will see the remote directory (EC2 instance) on the right side of the WinSCP interface.

- Navigate to the local directory on the left side of the interface where your Python project files are located.

- To transfer files from your local machine to the EC2 instance, select the files you want to transfer and drag them from the left side to the right side.

9. \*\*Verify File Transfer:\*\*

- After transferring the files, you should see them in the remote directory (EC2 instance).

10. \*\*Close WinSCP:\*\*

- Once you've finished transferring files, you can close WinSCP.

Now, your Python project files should be in the home directory of the `ubuntu` user on your EC2 instance and ready for use. You can log in to your EC2 instance via SSH as previously mentioned in the instructions to run your Python project.