**Exercise 1**

 Bookmark this page

In this exercise, you will create and activate your AWS account. To begin, follow the steps below.

**1. Go to the**[**Amazon Web Services home page**](https://aws.amazon.com/)**and sign up for an account.**

* Go to the [Amazon Web Services home page](https://aws.amazon.com/).
* Click **Create a Free Account**.
* On the sign-in page, click **Create an Account**.  
  **Note**: This might be unavailable in your browser if you previously signed in to the AWS Management Console. In that case, click **Sign in to a different account**, and then click **Create a new account**.
* Type the requested account information, and then click **Continue**.
* Choose **Personal** or **Professional**.

**Note:** These two account types are identical in functionality. You can choose a personal account for your personal projects or choose professional for use within your company, an educational institution, or an organization.

* Type the requested company or personal information.
* Read the [AWS Customer Agreement](https://aws.amazon.com/agreement/), and then check the box.
* Click **Create Account and Continue**.

**2. Add a payment method.**

* On the **Payment Information** page, add a payment method by typing the requested information associated with your payment method.
* Click **Secure Submit**.

**3. Verify your phone number.**

* On the **Phone Verification** page, type a phone number that you can use to accept incoming phone calls.
* Enter the code displayed in the security check.
* When you're ready to receive a call, click **Call me now**. In a few moments, an automated system will call you.
* Type the provided PIN on your phone's keypad. After the process has completed, click **Continue**.

**4. Choose a support plan and sign in to your console.**

* On the **Select a Support Plan** page, select the **Basic Plan** included in the free tier.
* Click **Sign in to Console** to sign in to your console.

**Exercise 2**

 Bookmark this page

In this exercise, you will create your first Amazon EC2 instance and install a sample Python Flask app using user data. When you launch an instance in Amazon EC2, you can pass user data to the instance that can be used to perform common automated configuration tasks. User data is usually passed in the form of shell-scripts. You can pass this data into the launch wizard as plain text, as a file while launching via the command line tools, or as base64-encoded text for API calls.

To get started, follow the steps below.

**1. Launch an Amazon EC2 instance with a user data script.**

In this section, you will launch an Amazon EC2 instance with a user data script. If you are familiar with Amazon EC2, you may want to attempt to complete this section by using the properties below before reading the step-by-step instructions.

Region: Oregon (us-west-2)  
Amazon Machine Image (AMI): Amazon Linux AMI *(Do not use the Amazon Linux 2 AMI)*  
Instance Type: t2.micro  
User data script:[Download](https://s3-us-west-2.amazonaws.com/us-west-2-tcdev/courses/AWS-100-ADG/v1.1.0/exercises/UserDataScript.txt)  
Tag: SamplePythonFlaskApp  
Security group name: exercise2-sg  
Security group rules: Allow HTTP  
Key Pair: Proceed without a key pair

Expand for step-by-step instructions.

* In the AWS Console, click **Services**, then click **EC2** to open the **EC2 dashboard**.
* At the top-right corner, select the **US West (Oregon)** region.
* From the EC2 dashboard, click **Launch Instance**.
* On the **Choose an Amazon Machine Image (AMI)** page, select **Amazon Linux AMI** by clicking **Select**. This AMI is free-tier eligible.  
  **Note**: Do not select the Amazon Linux 2 AMI option.
* On the **Choose an Instance Type** page, you can select the hardware configuration of your instance. Select **t2.micro**.
* Click **Next: Configure Instance Details**.
* On the Configure Instance Details page, leave the defaults and scroll down to the **Advanced Details** section and expand it.
* In the User data section, leave **As text** selected.

[Download the user data script](https://s3-us-west-2.amazonaws.com/us-west-2-tcdev/courses/AWS-100-ADG/v1.1.0/exercises/UserDataScript.txt) and copy and paste the contents of the script in the text area.( #!/bin/bash -ex

sudo yum update -y

sudo pip install flask

sudo pip install requests

mkdir PythonWebApp

cd PythonWebApp

sudo cat >> flaskApp.py << EOF

from flask import Flask

import requests

app = Flask(\_\_name\_\_)

@app.route("/")

def main():

r = requests.get('http://169.254.169.254/latest/dynamic/instance-identity/document')

text = "Welcome! Here is some info about me!\n\n" + r.text

return text

if \_\_name\_\_ == "\_\_main\_\_":

app.run(host='0.0.0.0', port=80)

EOF

sudo python flaskApp.py

)

* Click **Next: Add Storage**. Skip through this page and click **Next: Add Tags**.
* Click **Add Tag**. Tags enable you to categorize your AWS resources in different ways - for example, by purpose, owner, or environment.
* In the Key textbox, type **Name**
* In the Value textbox, type **SamplePythonFlaskApp**
* Click **Next: Configure Security Group**. Note that the wizard gives you an option to create a new security group or select an existing one. For this exercise, accept the default chosen option, Create a new security group.
* For Security Group Name, type **exercise2-sg**
* In the security group table, **delete the SSH rule** by clicking the **X** button at the end of the row.
* Click **Add Rule**.
* For **Type**, leave **Custom TCP Rule** selected.
* For **Port Range**, type **80**
* For **Source**, type **0.0.0.0/0**
* Click **Review and Launch**.
* On the Review Instance Launch page, review the details and click **Launch**.
* When prompted for a key pair, select **Proceed without a key pair**.
* Select the acknowledgement check box, and then click **Launch Instances**.
* Click **View Instances** to return to the Instances page.
* On the Instances page, you can view the status of the launch. It can take a few minutes for the instance to be ready so that you can connect to it. Check that your instance has passed its status checks. You can view this information in the Status Checks column.  
  **Note**: It takes a few minutes for the status checks to pass. Wait until the status checks changes from **Initializing** to **2/2 checks passed**.
* Once the instance is ready, select the instance and write down the **IPv4 Public IP** found in the Descriptions tab at the bottom.

**2. Test the sample app running on your instance**

* Open a browser and type the public IP of the Amazon EC2 instance you copied earlier.
* You should see a sample Python app running on your Amazon EC2 instance.  
  **Congratulations!** You have launched your first web server in AWS.

**3. Terminate the Amazon EC2 instance**

In this section, you will terminate the Amazon EC2 instance by selecting the instance in EC2 dashboard and clicking **Actions -> Instance State -> Terminate**.

Expand for step-by-step instructions

* In the AWS Console, click **Services**, then click **EC2** to open the **EC2 dashboard**.
* In the navigation pane, click **Instances**. In the list of instances, select the **SamplePythonFlaskApp** instance.
* Click **Actions**, **Instance State**, **Terminate**.
* Click **Yes, Terminate** when prompted for confirmation.
* Amazon EC2 shuts down and terminates your instance. After your instance is terminated, it remains visible on the console for a short while, and then the entry is deleted.

## Exercise 3

 Bookmark this page

In this exercise, you will start laying the infrastructure for the course project by creating a VPC and subnets via an AWS CloudFormation template. You will create more AWS resources in these subnet in the subsequent exercises.  
You will also create an Amazon EC2 instance and learn how to connect to it using an SSH client. You use an SSH client for remote login and administration over an authenticated secure channel. You will then look at the log files on the instance and extract information from the instance metadata and user data script.

To get started, follow the steps below.

### 1. Launch an AWS CloudFormation template to create VPC and subnets.

In this section, you will create a VPC and subnets by launching an AWS CloudFormation template. If you are familiar with AWS CloudFormation, you may want to attempt to complete this section by using the properties below before reading the step-by-step instructions.

Region: Oregon (us-west-2)  
CloudFormation template: [Download template](https://s3-us-west-2.amazonaws.com/us-west-2-tcdev/courses/AWS-100-ADG/v1.1.0/exercises/templates/vpc.template)  
Name of the stack: edx-vpc-stack

Expand for step-by-step instructions.

* In the AWS Console, click **Services**, then click **CloudFormation** to open the **CloudFormation dashboard**.
* Make sure you are still in the **Oregon** AWS Region.
* Click **Create Stack**.
* [Download the AWS CloudFormation template](https://s3-us-west-2.amazonaws.com/us-west-2-tcdev/courses/AWS-100-ADG/v1.1.0/exercises/templates/vpc.template)to create a VPC and save it locally on your computer.
* To select the AWS CloudFormation template you just downloaded, click **Choose file**.
* Click **Next**.
* In the **Stack name** textbox, type **edx-vpc-stack**.
* Click **Next**. Skip the Options page and click **Next**.
* Click **Create**. You will notice that the status of the template is **CREATE\_IN\_PROGRESS.**The template should finish creating in a minute.
* In your AWS Management Console home page, in the **AWS services** search textbox at the top, type **VPC**, and then select **VPC** to open the VPC dashboard.
* In the VPC dashboard, on the left navigation menu, click **Your VPCs**. You will see a VPC named **edx-build-aws-vpc** in the list. Write down the vpc-id of the edx-build-aws-vpc. You will need the vpc-id in subsequent exercises.
* In the VPC dashboard, on the left navigation menu, click **Subnets**. You will see four subnets starting with edx-subnet-xxxx. Write down the subnet-id of edx-subnet-public-a. You will need the subnet-id in subsequent exercises.

### 2. Launch an Amazon EC2 instance with a user data script in a VPC.

In this section, you will launch an Amazon EC2 instance with an user data script. If you are familiar with Amazon EC2, you may want to attempt to complete this section by using the properties below before reading the step-by-step instructions.

Region: Oregon (us-west-2)  
Amazon Machine Image (AMI): Amazon Linux AMI (Do not use the Amazon Linux 2 AMI)  
Instance Type: t2.micro  
Network VPC: edx-build-aws-vpc  
Subnet: edx-subnet-public-a  
User data script:[Download](https://s3-us-west-2.amazonaws.com/us-west-2-tcdev/courses/AWS-100-ADG/v1.1.0/exercises/UserDataScript.txt)  
Tag: Ex3WebServer  
Security group name: exercise3-sg  
Security group rules: Allow HTTP and SSH  
Key Pair: Create a new key pair and save it for later use.

Expand for step-by-step instructions.

* In the AWS Console, click **Services**, then click **EC2** to open the **EC2 dashboard**.
* At the top right corner, select the **US West (Oregon)** region.
* From the EC2 dashboard, click **Launch Instance**.
* On the **Choose an Amazon Machine Image (AMI)** page, select **Amazon Linux AMI** by clicking **Select**. This AMI is free-tier eligible.  
  **Note**: Do not select the Amazon Linux 2 AMI option.
* On the **Choose an Instance Type** page, select **t2.micro**.
* Click **Next: Configure Instance Details**.
* For **Network**, select **edx-build-aws-vpc**.
* For **Subnet**, select **edx-subnet-public-a**.
* Leave the defaults and scroll down to the **Advanced Details** section and expand it.
* [Download the user data script](https://s3-us-west-2.amazonaws.com/us-west-2-tcdev/courses/AWS-100-ADG/v1.1.0/exercises/UserDataScript.txt) and copy and paste the contents of the script in the text area.
* Click **Next: Add Storage**. Skip through this page and click **Next: Add Tags**.
* Click **Add Tag**.
* In the **Key** textbox, type **Name**
* In the **Value** textbox, type **Ex3WebServer**
* Click **Next: Configure Security Group**. Note that the wizard gives you an option to create a new security group or select an existing one. For this exercise, leave the default chosen option, Create a new security group.
* For **Security group name**, type **exercise3-sg**
* Click **Add Rule**.
* For **Type**, leave **Custom TCP Rule** selected.
* For **Port Range**, type **80**
* For **Source**, type **0.0.0.0/0**
* **Note**: The inbound rule for SSH is added by default.
* Click **Review and Launch**.
* On the **Review Instance Launch** page, review the details and click **Launch**.
* When prompted for a key pair, select **Create a new key pair**, enter a name for the key pair, and then click **Download Key Pair**.

**Note**: This is the only chance for you to save the private key file, so be sure to download it. You will use the same key pair for all subsequent exercises in the course. Save the private key file in a safe place. You'll need to provide the name of your key pair when you launch an instance and the corresponding private key each time you connect to the instance.

* Select the acknowledgement check box, and then click **Launch Instances**.
* Click **View Instances** to return to the instances page.
* On the **Instances** page, you can view the status of the launch. It can take a few minutes for the instance to be ready so that you can connect to it. Check that your instance has passed its status checks. You can view this information in the **Status Checks** column.  
  **Note**: It takes a few minutes for the status checks to pass. Wait until the status checks changes from Initializing to 2/2 checks passed.
* Once the instance is ready, select the instance and note down the **IPv4 Public IP** found in the **Descriptions** tab at the bottom.
* Paste the public IP address of your instance in your web browser to display the welcome web page. This page is now displaying dynamic information about your server.

### 3. Connect to your Amazon EC2 instance.

In this section, you will connect to your Amazon EC2 instance via SSH. An SSH connection requires port 22 to be open on your network. You may need to contact your network administrator to ensure that this is open.

**For MAC/Linux users**:

* Open the **Terminal** application.
* Type the commands below. In both commands, replace **PATH-TO-PEM-FILE** with a reference to the .pem file that you downloaded while launching the instance. In the second command, replace **PUBLIC-IP** with the IPv4 Public IP of the instance.

chmod 400 PATH-TO-PEM-FILE  
ssh -i PATH-TO-PEM-FILE ec2-user@PUBLIC-IP

* You will see a prompt like the one below. Answer **yes** to the prompt.

The authenticity of host '54.201.7.240 (54.201.7.240)' can't be established. ECDSA key fingerprint is SHA256:TrCPkFBL0F+pTp3LH+UGFPhGjl7N4qaoLucu21RWsRM. Are you sure you want to continue connecting (yes/no)?

**For Windows users**:

Expand for step-by-step instructions.

* [Install PuTTY.](https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html)
* PuTTY does not natively support the private key format (.pem) generated by Amazon EC2. PuTTY has a tool named PuTTYgen, which can convert keys to the required PuTTY format (.ppk). You must convert your private key into this format (.ppk) before attempting to connect to your instance using PuTTY. To convert your private key (.pem file) using PuTTYgen, follow the instructions below:
  + Start PuTTYgen.
  + Under **Type of key to generate**, select **RSA**.
  + Click **Load**. By default, PuTTYgen displays only files with the extension .ppk. To locate your .pem file, select the option to display files of all types.
  + Select your **.pem file** for the key pair that you specified when you launched your instance, and then click **Open**. Click **OK** to dismiss the confirmation dialog box.
  + 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 the same name for the key that you used for the key pair (for example, my-key-pair). PuTTY automatically adds the .ppk file extension.  
    Your private key is now in the correct format for use with PuTTY. You can now connect to your instance using PuTTY's SSH client.
* Start **PuTTY**.
* In the **Category** pane, click **Session**.

In the **Host Name** text box, type **ec2-user@IP-ADDRESS**, where IP-ADDRESS is the public IP address of your Amazon EC2 instance.

* In the **Category** pane, expand **Connection**, expand **SSH**, and then click **Auth**.
* Click **Browse**.
* Select the **.ppk file** that you generated for your key pair, and then click **Open**.
* If you plan to start this session again later, you can save the session information for future use. In the **Category** tree, select **Session**, enter a name for the session in **Saved Sessions**, and then click **Save**.
* Click **Open** to start the PuTTY session. PuTTY will ask whether you wish to cache the server’s host key. Click **Yes**.

### 4. View log file, query instance metadata, and user data.

In this section, you will inspect the cloud-init logs to verify the steps in the UserData script executed on the Amazon EC2 Instance. Then, you will query the instance metadata service from the Amazon EC2 instance so that you can see how your application is printing information about itself on the welcome page.

* To view the log file, type the command below in your instance terminal.

cat /var/log/cloud-init-output.log

Explore the log file to see the log entries generated for installing the user data script.

* To view the instance metadata, type the command below:

curl http://169.254.169.254/latest/meta-data/

* Execute the command below to get the instance identity document of your instance:

curl http://169.254.169.254/latest/dynamic/instance-identity/document

* Execute the command below to get the instance public IP address:

curl http://169.254.169.254/latest/meta-data/public-ipv4

* Execute the command below to get the MAC address of the instance:

curl http://169.254.169.254/latest/meta-data/mac

* Execute the command below to get the VPC ID in which the instance resides. Make sure to replace **Your-MAC** in the command below with the MAC address of your instance:

curl http://169.254.169.254/latest/meta-data/network/interfaces/macs/Your-MAC/vpc-id

* Execute the command below to get the subnet-id in which the instance resides. Make sure to replace **Your-MAC** in the command below with the MAC address of your instance:

curl http://169.254.169.254/latest/meta-data/network/interfaces/macs/Your-MAC/subnet-id

* Execute the command below to get the instance user data:

curl http://169.254.169.254/latest/user-data

* Go to the **EC2 dashboard** in your console, locate the Amazon EC2 instance you created and verify the public IP address, the VPC ID and the subnet-id of the instance you just queried in the instance terminal. You should be able see this information in the **Description** tab at the bottom.

### 5. Terminate the Amazon EC2 instance.

In this section, you will terminate the Amazon EC2 instance by selecting the instance in the EC2 dashboard and clicking **Actions -> Instance State -> Terminate**.

Expand for step-by-step instructions.

* In the AWS Console, click **Services**, then click **EC2** to open the **EC2 dashboard**.
* In the navigation pane, click **Instances**. In the list of instances, select the **Ex3WebServer** instance.
* Click **Actions**, **Instance State**, **Terminate**.
* Click **Yes, Terminate** when prompted for confirmation.
* Amazon EC2 shuts down and terminates your instance. After your instance is terminated, it remains visible on the console for a short while, and then the entry is deleted.