Lab 1: Introducing Amazon Elastic Compute Cloud (EC2)

Exercise 1. Launch an EC2 Linux Instance and Log in Using SSH

1. From the EC2 Dashboard, click to launch a new instance and select a Linux AMI and instance type. Remember, the t2.micro is Free Tier–eligible if your AWS account is still within its first year.
2. Explore the Configure Instance Details, Add Storage, and Add Tags pages—although the default settings should work fine.
3. On the Configure Security Group page, make sure there's a rule permitting incoming SSH (port 22) traffic. It should be there by default.
4. Before letting you launch the instance, AWS will require you to select—or create—a key pair. Follow the instructions.
5. Once the instance is launched, you can return to the Instances Dashboard to wait until everything is running properly.
6. Click the Actions pull-down and then the Connect item for instructions on how to connect to the instance from your local machine. Then connect and take a look at your virtual cloud server.

Exercise 2. Create a launch Template

In this exercise, you'll create a launch template that installs and configures a simple web server. You'll then use the launch template to manually create an instance.

1. In the EC2 Dashboard, click Launch Templates.
2. Click the Create Launch Template button.
3. Give the launch template a name such as MyTemplate.
4. Click the Search For AMI link to locate one of the Ubuntu Server LTS AMIs. If you're in the us-east-1 region, you can use ami-0ac019f4fcb7cb7e6.
5. For Instance Type, select t2.micro.
6. Under Security Groups, select a security group that allows inbound HTTP access. Create a new security group if necessary.
7. Expand the Advanced Details section and input the following in the User Data text input field:

   ```
   #!/bin/bash
   apt-get update
   apt-get install -y apache2
   echo "Welcome to my website" > index.html
   cp index.html /var/www/html
   ```

8. Click the Create Launch Template button.
9. Click the Launch Instance From This Template link.
10. Under Source Template Version, select 1 (Default).
11. Click the Launch Instance From Template button.
12. After the instance boots, browse to its public IP address. You should see a web page that says "Welcome to my website."

Exercise 3. Install the AWS CLI in a Virtual Environment and Use It to Launch an EC2 Instance

1. Install virtualenv using pip.
   ```
   $ pip install --user virtualenv
   ```
2. Create a virtual environment and name it.
   ```
   $ virtualenv ~/aws-cli
   ```
   Alternatively, you can use the -p option to specify a version of Python other than the default.
   ```
   $ virtualenv -p /usr/bin/python3.5 ~/aws-cli
   ```
3. Activate your new virtual environment.
   ```
   $ source ~/aws-cli/bin/activate
   ```
4. Install the AWS CLI into your virtual environment.
   ```
   (aws-cli)~$ pip install --upgrade awscli
   ```
5. Verify that the AWS CLI is installed correctly.
   ```
   (aws-cli)~$ aws --version
   aws-cli/1.16.116 Python/3.6.8 Linux/4.14.77-81.59-
   amzn2.x86_64 botocore/1.12.106
   ```
   You can use the `deactivate` command to exit the virtual environment. Whenever you start a new session, you must reactivate the environment.
6. Configure the AWS CLI

   For general use, the `aws configure` command is the fastest way to set up your AWS CLI installation.
   ```
   $ aws configure
   AWS Access Key ID [None]: AKIAIOSFODNN7EXAMPLE
   AWS Secret Access Key [None]: wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
   Default region name [None]: us-west-2
   Default output format [None]: json
   ```
   To work with EC2 Instances using the AWS CLI you need to configure the following information: access key, secret access key, AWS Region, and output format.

   6.1. Create access and secret keys for an IAM user:
   - Open the AWS Management Console.
   - In the navigation pane, choose the name of the user whose keys you want to create, and then choose the Security credentials tab.
   - In the Access keys section, choose Create access key.
   - To view the new access key pair, choose Show. You will not have access to the secret access key again after this dialog box closes. Your credentials will look something like this:

     Access key ID: AKIAIOSFODNN7EXAMPLE
     Secret access key: wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
   - Insert keys to AWS Access Key ID and AWS Secret Access Key fields, respectively.

   6.2. Specify the AWS Region whose servers you want to send your requests to by default:

   6.3. Look at the Instances panel from the EC2 Dashboard to find Availability zone attribute (e.g., us-east-2c)

   6.4. Insert the value of availability zone attribute to the Default region name without the last letter (e.g., `us-east-2`).

   6.5. Type `json` to the Default output format.
7. Stop instances using the following commands:
   ```
   $ aws ec2 stop-instances --instance-ids i-5203422c
   ```
8. Look at the result of command in the EC2 Dashboard
9. Terminate instances using the following commands:
   ```
   $ aws ec2 terminate-instances --instance-ids i-5203422c
   ```
10. Look at the result of command in the EC2 Dashboard

Exercise 4. Launch a Simple CloudFormation Template

1. From the CloudFormation page, click Create Stack.
2. Click the Use a sample template radio button, and select the WordPress blog with a local MySQL databse in the Sample templates list.
3. Click the View In Designer link and spend a few minutes reading through the template text in the bottom panel to get a feel for how all the elements are organized. Values such as the password fields will be automatically populated in later steps.
   Change the “Default” attribute value to t2.micro in the InstanceType block.
4. When you’re done exploring and changing, click the Create Stack icon at the top of the page, which will take you back to the Select Template page. Make sure that the Amazon S3 URL is listed and click Next.
5. Provide a stack name, database name, two passwords, and a DBUser name.
6. Verify an instance type and provide the name of an existing EC2 key pair so you’ll have SSH access. Then click Next.
7. The defaults from the Options page are all fine for this exercise. Click Next, review your settings, and click Create.
8. It will take some time to launch the resources. While you’re waiting, you can view progress in the CloudFormation Dashboard. The Outputs tab in the Dashboard will include a website URL through which you can access the WordPress blog’s public page. Click to the link of your WordPress blog.
9. There might be an issue with a PHP version that is not supported by the WordPress. Fix this by connecting to the web server using SSH and install PHP 5.6 version with commands:
   - Remove OLD Apache
     ```
     sudo service httpd stop
     sudo yum erase httpd httpd-tools apr apr-util
     ```
   - Remove OLD PHP
     ```
     sudo yum remove php-*
     ```
   - Install PHP 5.6 (Apache 2.4 will be automatically installed with this)
     ```
     sudo yum install php56
     ```
   - Make sure all the required PHP extensions are installed
     ```
     yum list installed | grep php
     ```
   - If not then install them using
     ```
     sudo yum install php56-xml php56-xmlrpc php56-soap php56-gd
     ```
   - To list the other available php extensions
     ```
     yum search php56
     ```
   - PHP 5.6 MySQL extension (Assume you have already installed MySQL)
     ```
     sudo yum install php56-mysqlnd
     ```
     (NOTE: it is not php56-mysql)
   - Start / Restart Apache
     ```
     sudo service httpd start
     sudo service httpd restart
     ```
   - Check the version
     ```
     php -v
     httpd -v
     ```
   - Reload the WordPress blog’s page.
10. Set the initial settings and click Install WordPress.
11. Log In to admin page and customize your blog. Be creative!

Assignment:

1. Set up a public address in Elastic IPs tab and assign it to the web server on which WordPress is hosted.
2. Configure WebServer so that your blog page is accessible only through the public IP address (without specifying the /wordpress path). For example, you specify a public IP address in the address bar of the browser and gain access to the blog.
3. There may be problems with CSS. Search the web for how to fix broken CSS after changing the site URL.
4. In Moodle, submit your public IP address, which hosts the WordPress blog. After the TA has evaluated your assignment, terminate all instances and remove the CloudFormation stack.