Python on Amazon Web Services

[Instructions repurposed from CSI 101, Fall 2016]

Follow these steps to spin up your own Amazon EC2 (aka a virtual machine (VM)). A pre-configured EC2 image in the N. Virginia data center has been provided. It has pre-installed versions of Apache Web Server and Python 3.

Steps are also provided for connecting to this EC2 from a Web-based IDE called CodeAnywhere.

**Take care to follow instructions to stay within the AWS free tier!**

* Only 1 VM of size t2.micro should be running at one time to stay under the free tier compute hour limits. (750 hours per month)

EC2 Image Steps:

Step 1: Create EC2 using the AMI

In this step we will find a pre-made, pre-configured EC2 (virtual machine) image, and launch it.

1. Go to EC2 Dashboard
2. Click on AMIs under Images
3. Click the search dropdown that says Owned By Me, and change it to Public Images
4. Click in the Search Bar and click AMI Name
5. Enter UAlbany-CSI101-Fall2016-v2 (in the N. Virginia data center)
6. Click Launch
7. **Select t2.micro**
8. Click Review and Launch
9. Click Launch
10. Select a key pair
    1. If you don’t have one, select Create New Key Pair
    2. Name and download the new key pair[[1]](#footnote-1)
11. Click Launch Instances
12. Go back to the EC2 Dashboard, view instances, and verify that the EC2 is being set up. It may take Amazon several minutes to launch your EC2.

Step 2: Custom Security Group

In this step we make sure our security perimeter will allow us to connect to the running EC2 in a variety of ways using some specific network ports. We are being very free and loose with security. (You may choose more strict settings, but keep in mind your service must be publicly available.)

1. Click on the Instance that was created in the previous step
2. Find the security group that this instance is using (likely the “default” group). This is found at the bottom of the page once the instance is selected
3. On the EC2 Dashboard, click Security Groups under Network and Security
4. Select the security group that your instance is using
5. On the bottom of the screen, click Inbound and click edit
6. Click Add Rule and create the following rules

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Type** | **Protocol** | **Port Range** | **Source** |  |
| HTTP | TCP | 80 | Anywhere | 0.0.0.0/0 |
| All Traffic | All | 0-65535 | Anywhere | 0.0.0.0/0 |
| MySQL/Aurora | TCP | 3306 | Anywhere | 0.0.0.0/0 |
| Custom TCP Rule | TCP | 20-22 | Anywhere | 0.0.0.0/0 |
| Custom TCP Rule | TCP | 1024-1048 | Anywhere | 0.0.0.0/0 |

Step 3: Create an Elastic IP

Every time you re-launch / reboot the EC2, it may be assigned a new IP address and host name. This will make your life more complicated. Thus we will ask Amazon for a permanent address (the so-called strangely named Elastic IP). We will associate this Elastic IP with our EC2 so that the address will remain the same even after reboot.

1. On the EC2 Dashboard, Click Elastic IP under Network and Security
2. Click Allocate New Address
3. Click Yes
4. With the address selected, click Actions
5. Click Associate Address
6. For Instance, select your instance from the drop-down list
7. Click Associate

Step 4: Get CodeAnywhere

CodeAnywhere is a Web-based programming editor. It supports Python. We will use CodeAnywhere to edit Python code on our running EC2.

1. Go to the Chrome app store
2. Search for CodeAnywhere
3. Click add to Chrome
4. Open CodeAnywhere and sign up for a free account
5. Sign in

Step 5: Connect CodeAnywhere to EC2 Instance

1. In CodeAnywhere, Click File, New Connection, and FTP
2. For Connection Name, enter any text (it doesn’t matter what – just give it a name)
3. For Hostname, enter the instance’s public DNS found on the instances page of the EC2 Dashboard
4. For both Username and Password, enter csi101
5. Click connect
6. See in CodeAnywhere some basic HTML and Python pages we have created for you in advance to get you started.

1. You will not necessarily need this key pair. Do this step anyway. When the key pair downloads, keep it in a safe place in case we need it later. What it would be used for is to create a login session (a Unix command shell session) to this virtual machine. We the Instruction Team have gone to great lengths to not require you to need to do this as we’ve provided a pre-built, pre-configured EC2 image for you to use “as is”. [↑](#footnote-ref-1)