Kerry Sheldon Follow
May 6, 2018 · 5 min read · Listen

EC2 Exercise 1.1: Host a Static Webpage

(This post is part of the AWS for the Bootcamp Grad series. The series consists of exercises to build familiarity with various aspects of the AWS ecosystem. Again, all of these posts are "exercises" for introductory exposure to AWS — they are NOT represented as best practices.)

Background

As I began to learn about EC2, I had no background with terms like mount, volumes, and IOPS. The use of this terminology, and the assumed knowledge it implied, made me think EC2 concepts were too complex for my knowledge base. But as I stuck with the tutorials, I found myself understanding what was going on, even though I didn't always understand what was being said. I wished someone had simply said: "Creating an EC2 instance is like going to the store to pick out the right type of laptop to run your program." As soon as I built that mental model, the rest fell into place.

In this exercise, you will do the following:

- Launch an EC2 instance through the AWS console (i.e. build a "virtual laptop" to serve your website)
- SSH into to the EC2 instance and install a web server (i.e. use the terminal to "log in" to the "virtual laptop" and interact with it)
- Host a static webpage on the EC2 instance (i.e. "deploy" a static html file to the web)

If you haven’t created an AWS account yet, do that first.

Sign in to the AWS console and search for “EC2”. Navigate to the EC2 dashboard and click “Launch Instance”.

1. Choose AMI

In this step you will choose the operating system for your “virtual laptop” and the basic set of software that will come preinstalled.

Choose a free-tier eligible Linux option (**Amazon Linux AMI, SSD Volume Type**) and click “Select”

2. Choose Instance Type

In this step, you will choose how powerful the “virtual laptop” will be — things like the number of CPUs, amount of memory, type of hard disk/storage, and network speed.

Choose the option marked as free tier eligible (**General Purpose — t2.micro)**— it will be one of the smallest, least powerful options. Click next to configure instance details.

3. Configure Instance

In this step, you will accept all of the default options. Glance at the options to get a sense of the types of things that can be done here.

Click next to add storage.

4. Add Storage

In this step, you will choose the hard drive(s) for your “virtual laptop”, and how fast it will perform. By default, a hard drive is already attached (it is known as a root drive because it is capable of starting the operating system).

We will accept the default options and move to the next step. Click next to add tags.

5. Add Tags

In this step you can add "tags" to help you to manage and administer your AWS resources. We don't have a need for this, but add one for the experience.

Click Add Tag— enter "name" for the key and "WebServer" for the value.

Click Next to Configure the Security Group

6. Configure Security Group

In this step, you will choose the type of traffic your "virtual laptop" will allow from the outside. You need to allow two types of traffic — SSH (so we can "log in" to the virtual laptop) and HTTP (so we can view our webpage through the browser).

Select "Create a new security group". Enter anything you want for the name and description

Keep the SSH rule that is already listed.

Click "add rule". Select HTTP for the type and keep everything else as it is.

Click review and launch.

7. Review

Ignore the security warning (we will tear down this instance as soon as we are done) and click Launch.

A pop up window will ask you to select or create a key pair. A key pair is needed to securely SSH ("log in") to our new EC2 instance.

- Select 'Create a new key pair'
- Give the Key Pair a name — e.g. 'ec2-key-pair'

- Click View Instance to navigate back to the EC2 dashboard

SSH into the EC2 instance and Install a Web Server

You will see your new instance listed on the EC2 dashboard. Wait until the Instance State is 'running'.

Selecting the instance (click the button next to the instance) displays information about the instance below. In this area, you will see the IPv4 Public IP address of your instance. Copy it to your clipboard.

Navigate to your terminal and do the following:

1. Change the permissions on your key-pair file

First, save the downloaded key-pair .pem file to a directory of your choice (I put mine in my ~/.ssh/ directory). Then change the permissions with the following command. (If you don't do this, the key-pair is considered "too permissive", or unsafe, because it is ostensibly readable by other users. You won't be able to SSH into the EC2 instance as a result)

```
chmod 400 <path_to_key_pair_file>
```

2. SSH into your new EC2 instance

```
ssh -i <path_to_key_pair_file> ec2-user@<public_ip_from_dashboard>
```

Type yes to continue.

At this point, your terminal is now interacting directly with your EC2 instance (aka your "virtual laptop") — rather than your physical machine.

3. Elevate your privileges

4. Update all of the packages on the instance

```
yum update -y
```

(note: if you are familiar with using homebrew on your Mac, you can think of the way we are using `yum` here as similar to `brew`)

5. Install an apache webserver

```
yum install httpd -y
```

6. Start the webserver

```
service httpd start
```

7. Configure the web server to restart if it gets stopped

```
chkconfig httpd on
```

Add a static HTML file to be served

By default, the apache web server will display the index.html file found in `/var/www/html` directory in the root path of your website.

In this section you will create an index.html file to be served.

1. Navigate to the directory

Using your preferred editor (vi, nano, etc) create the index.html file:

```
nano index.html
```

Add valid html to the file (e.g.):

```
<html><body>My first EC2 instance</body></html>
```

Exit and save. Make sure that the file has content:

```
cat index.html
```

Navigate back to the EC2 dashboard in the AWS console and copy the Public DNS(IPV4) of your instance into your clipboard. Paste that address into your browser. If all went well, you will see the html that you just created!

Clean Up

Navigate to the EC2 dashboard, select your instance, and click on Actions. Select Instance State → Terminate. Confirm that you want to terminate and you're done. This will automatically kick you out of the SSH session in your terminal.

Next

EC2 Exercise 1.2: Host a Static Webpage with Content from Github