AWS Academy Introduction

Software Architecture

February 21, 2022

Brae Webb

1 Before Class

Prior to attending class, **select an open-source website** that you would be interested in deploying. In this guide, we will be deploying hextris [1], an online game similar to tetris. You may also deploy hextris, although, you can take the opportunity choose something which interests you. Note that deploying other websites is likely to be more involved than the example in the guide.

Potential ideas are:

- **Habitica:** https://github.com/HabitRPG/habitica
- **DailyNotes:** https://github.com/m0ngr31/DailyNotes/
- **kcal:** https://github.com/kcal-app/kcal

Alternatively, browse through the impressive collection here:
https://github.com/awesome-selfhosted/awesome-selfhosted

2 This Week

This week our goal is to get acquainted with AWS Academy. Throughout the course we will use AWS Academy to learn how to deploy and manage infrastructure with AWS. Specifically, this week you need to:

- Enrol in
  1. AWS Academy Cloud Foundations [10906] course;
  2. AWS Academy Learner Lab - Foundation Services [10909] course; and
- Navigate the AWS Academy interface.
- Enter the AWS Console from an AWS Academy lab.
- Use the AWS Console to provision an EC2 instance with a simple web interface.

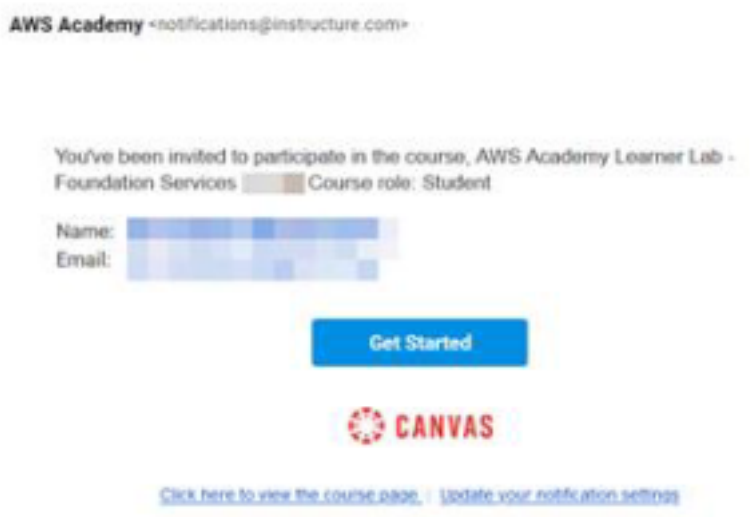
3 AWS Academy

AWS Academy is an educational platform to support teaching AWS services. In this course, we will be using it in two ways:

1. The Cloud Foundations course will be used as supplementary material to help cement your ability to use AWS. The use of Cloud Foundations is completely optional.
2. The Learner Lab provides access to an environment which will be used in these practicals to learn AWS.

4 Enrol in AWS Academy

1. Set up your AWS Academy account by responding to your email invitation and clicking **Get Started**. The email invitation will come from AWS Academy. Check your junk/spam folders.



2. Go to https://www.awsacademy.com/LMS_Login to login.
   (a) Press **Student Login**.
   (b) Use the email address that received the email invitation.

5 Exploring the Interface

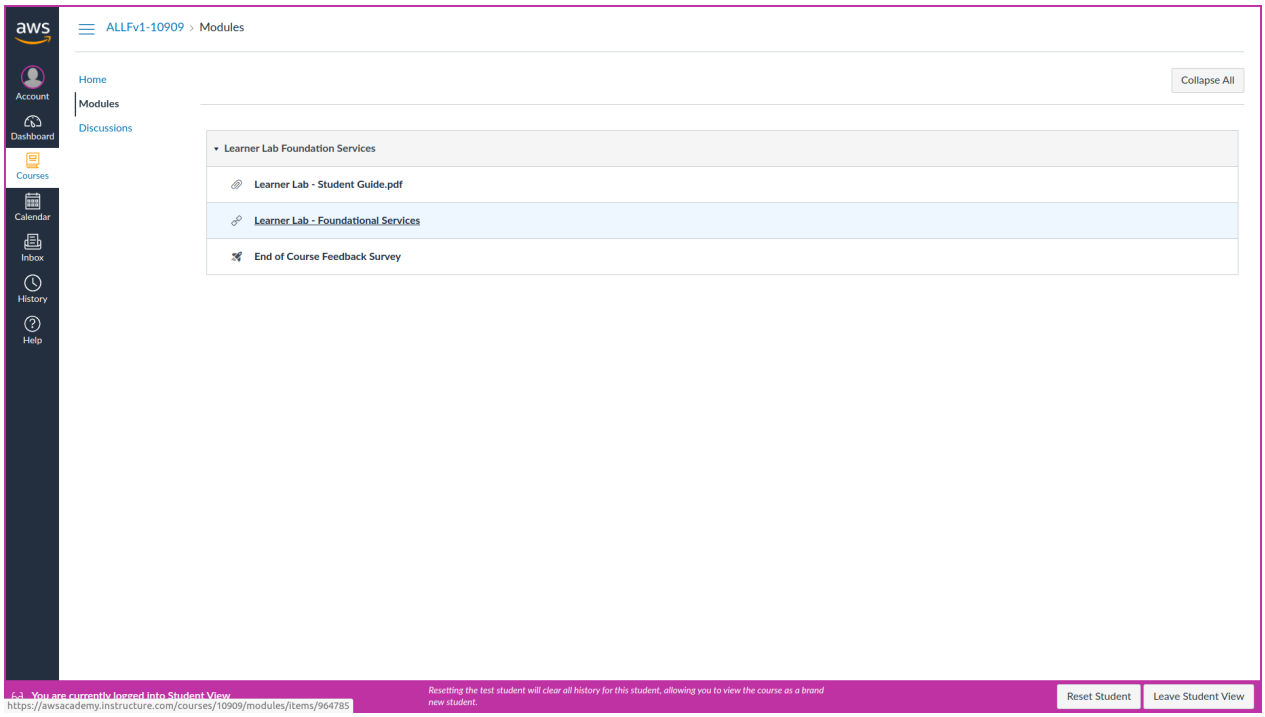
Aside

We will just be looking at the learner lab today, please ask on the slack if you need help using the cloud foundations course.

The following steps will enter the learner lab:

1. Once you have enrolled in the course, you should see this course page (minus everything in pink):

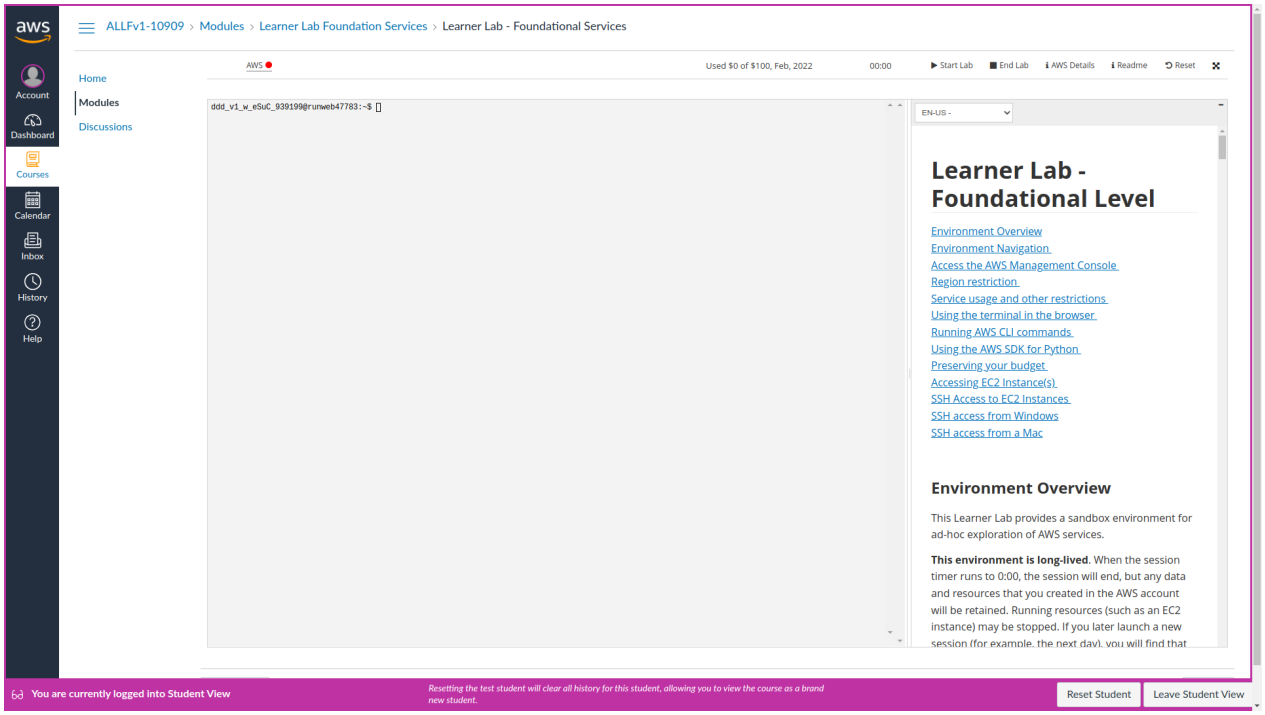
2. Navigate to the `Modules` tab and select the link for "Learner Lab - Foundational Services". You may also open and browse the "Learner Lab - Student Guide.pdf" link which covers some of the content of this guide.



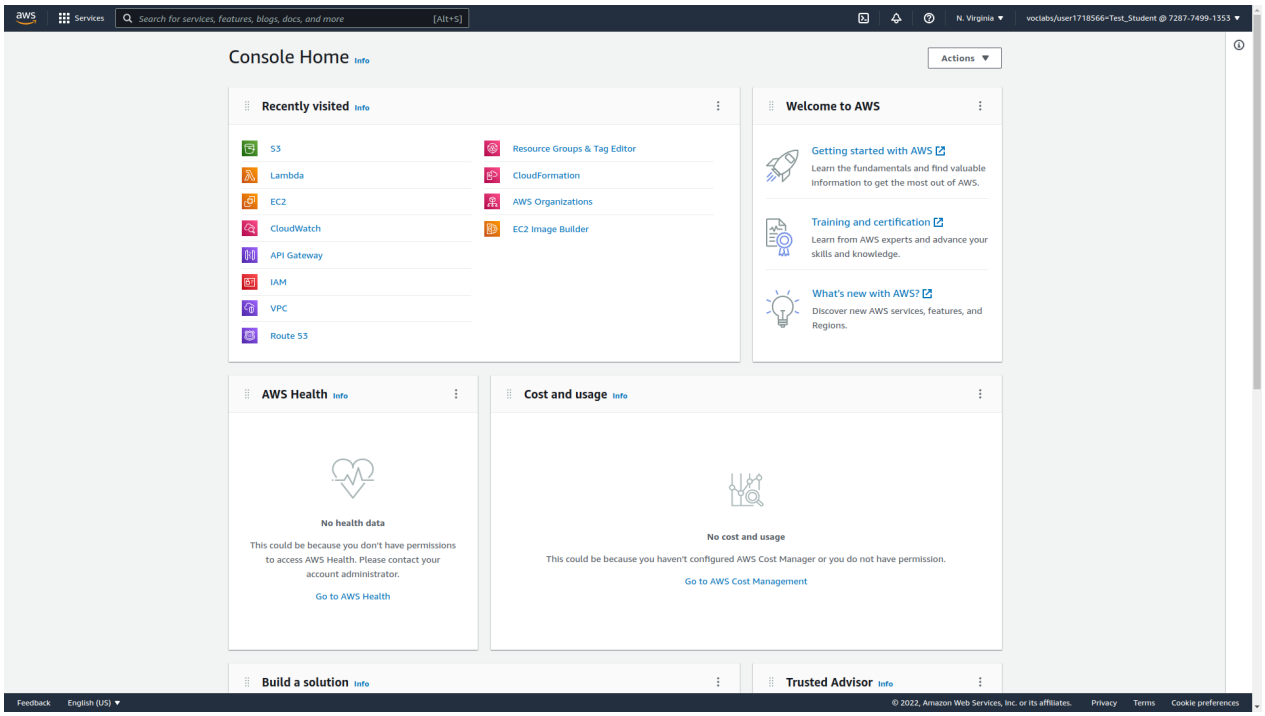
3. Now we have entered the main part of the interface, the learner lab.

   - The AWS text with the currently red circle is the link to open the AWS console.
   - You also see your budget, note that the budget is not updated in real-time.
   - The `00:00` indicates how long your lab has been activated. A lab can only remain active for 6 hours, after which it will be closed, unless you press start lab again before the 6 hours expires.
   - AWS details will become important later but are not needed now.
   - The README button will re-open the text panel currently on the right of the terminal interface.

- The terminal interface is not yet important.
- The right-hand has a lot of important information including what AWS services are available in the learner labs environment, please read it.



4. Go ahead and start the lab. It will take a few moments to get ready and the red circle will turn green once it is. Click on the green circle when it is available. This will open the AWS Console in a new tab.[1] If you end up working in a company which uses AWS, welcome to your new home.



[1] Your view will be different as you will not yet have any recently visited services.

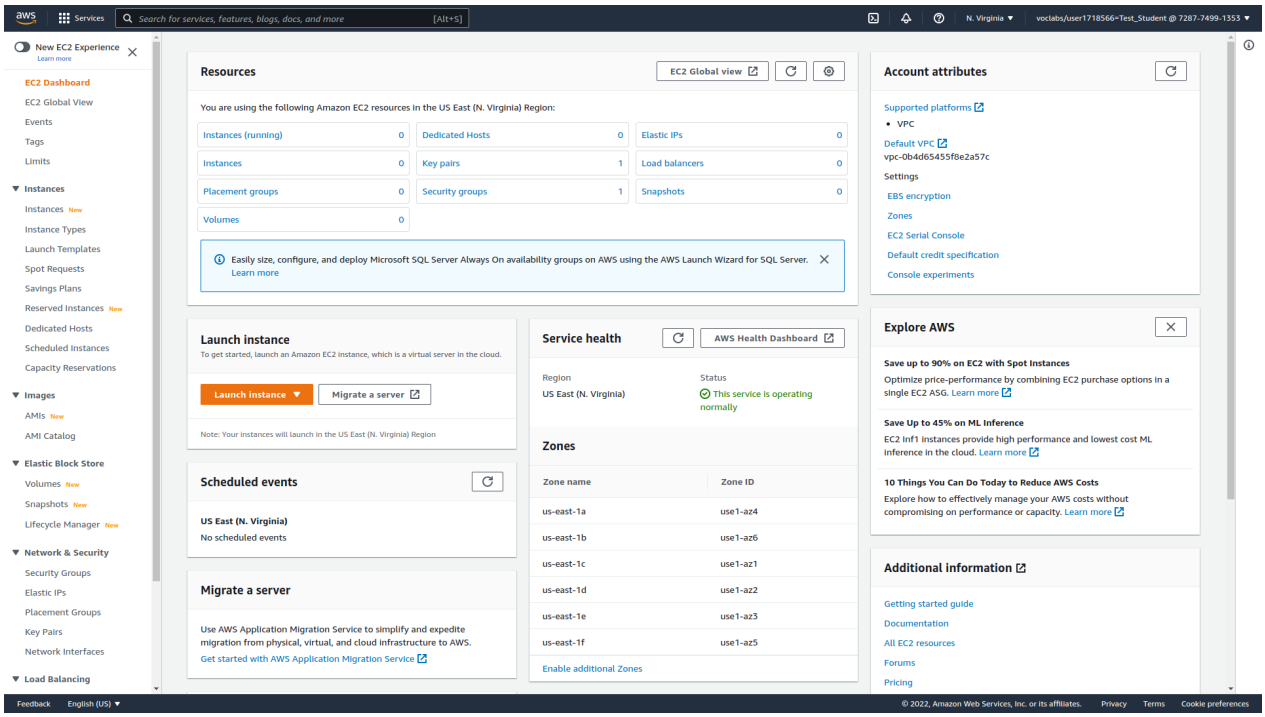
Aside

A short introduction to AWS: Amazon Web Services (AWS) is an Infrastructure as a Service (IaaS) and Software as a Service (SaaS) provider. They offer a collection of services which are helpful for development. For example, they offer virtual compute resources, database storage options, and networking to tie it all together. Services are offered with a pay as you go model, meaning you only pay for the seconds you use a service. We will get acquainted with some simple services offered by AWS now.

6 AWS EC2

Today we are going to focus on using AWS's EC2 service. Elastic Compute Cloud (EC2) is the primary compute service offered by AWS. It allows you to create a linux virtual machine on Amazon's infrastructure. You have full control over this machine and can configure it for whatever purpose you need.

Navigate to the search bar in the top left and find the EC2 service. You might find this interface overwhelming. It is important to note that since EC2 is one of the primary services offered by AWS, many smaller services we do not need are bundled into the service.



Today, we will just need the `Instances` dashboard. Navigate to there and select "Launch instances".

6.1 EC2 AMI

First we will need to select an Amazon Machine Image (AMI). An AMI is the template (cookie cutter) which provides instructions on how an instance should be provisioned. Amazon offers a range of built-in AMIs. There are also community AMIs or you can create your own. As we just want a simple server today, we will use one of the built-in AMIs.

Every AMI has a unique AMI code, something that looks like `ami-0a8b4cd432b1c3063`. We will use this AMI today[2] , it is considered one of the fundamental images. Search for it via the code and select it.

[2]Check the requirements for your chosen website, if your website requires a specific operating system, look for it as a community AMI. Ask for help if you aren't sure.

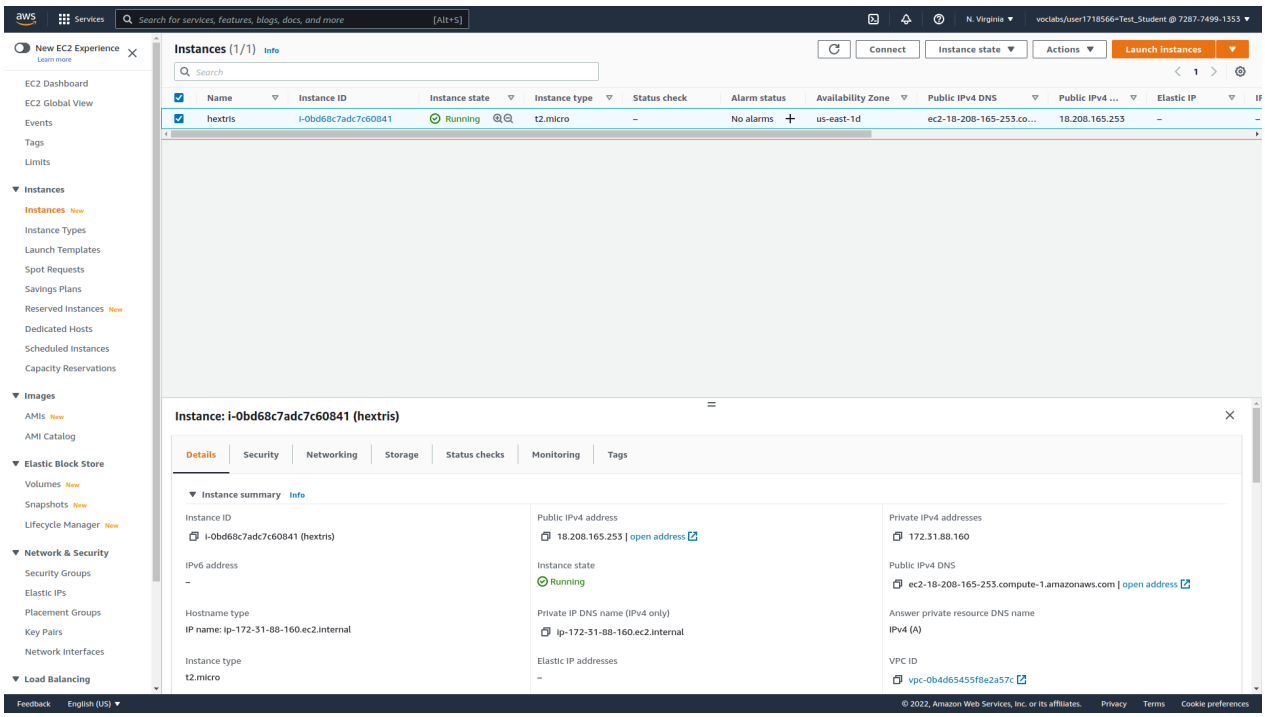
6.2 Instance Settings

The next few settings for configuring your instance are:

1. We do not need a large server, choose `t2.micro`.
2. Keep the 'Configure Instance' settings as default.
3. Keep the 'Add Storage' settings as default.
4. Add a 'Name' tag, call it the name of your website, e.g. `hextris`.
5. Choose 'Create a **new** security group'.
   (a) Give it a meaningful name, e.g. `hextris-http-ssh-access`.
   (b) Change the description to something meaningful, e.g. `Hextris HTTP and SSH access`.
   (c) Add a rule, select the `Type` as `HTTP`.
6. When prompted to choose a key pair (after the clicking `Launch` button), select the existing `vockey | RSA` option.

7 Accessing the Instance

Return to the `Instances` dashboard. You should now see a new instance created, its instance state might not yet be `Running`, if not, wait.



Note the public IPv4 address, we will need to use this to connect to the server.

Notice

For terminal examples in this section, lines that begin with a $ indicate a line which you should type while the other lines are example output that you should expect. Not all of the output is captured in the examples to save on space.

1. Return to the learner lab interface.
2. Run the following, replacing `127.0.0.1` with the public IP of your instance. This command uses the `vockey | RSA` key pair to gain SSH access to the machine.

```
$ ssh -i ~/.ssh/labsuser.pem ec2-user@127.0.0.1
```

example:

```
$ ssh -i ~/.ssh/labsuser.pem ec2-user@35.173.230.42
The authenticity of host '35.173.230.42 (35.173.230.42)' can not be established.
ECDSA key fingerprint is SHA256:W7OzzZm6nhwM9tB9Kb7enONPryo1a4259hJmSAZX7HQ.
Are you sure you want to continue connecting (yes/no)?

# At this point you will want to type yes and press enter
Warning: Permanently added '35.173.230.42' (ECDSA) to the list of known hosts.
Connection to 35.173.230.42 closed by remote host.
Connection to 35.173.230.42 closed.
```

8 Installing Hextris

Hextris is very simple to install, using an EC2 interface is perhaps overkill for it. It is an entirely client-side application which means we just have to serve the static files.

First, we will need to enable the basic serving of static files. We can install and start the `httpd` service for this. The AMI we have picked uses the yum package manager, so to install httpd we run:

```
$ sudo yum install httpd
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Resolving Dependencies
.....
.....
Total download size: 1.9 M
Installed size: 5.2 M
Is this ok [y/d/N]:

# enter y to install
.....
.....
Complete!
$ sudo systemctl enable httpd
Created symlink from /etc/systemd/system/multi-user.target.wants/httpd.service to
    /usr/lib/systemd/system/httpd.service.
$ sudo systemctl start httpd
```

All files in the `/var/www/html` directory will now be served when accessed via HTTP. Change to this directory and notice that it is currently empty.

Next, we need to download the static files to the server. We can use git for this, but first git needs to be installed on the machine.

```
$ sudo yum install git
```

Finally, let's double check we are in the `/var/www/html` directory.

```
$ cd /var/www/html
```

And clone the repository into that directory.

```
$ sudo git clone https://github.com/Hextris/hextris .
```

Now if you navigate to the **http** address of the public IP address (e.g. http://18.208.165.253), you should be able to see your newly deployed website.

9 Further Work

If you picked your own website to deploy, the steps for installation will likely be more involved. Track down those instructions and try your best to deploy the site. If you get stuck, ask your neighbour or your tutor.

Once you have worked out how to deploy your chosen website, please commit an installation script or installation instructions for others to re-use: https://github.com/CSSE6400/selfhost-cookbook

References

[1] L. Engstrom, G. Finucane, N. Moroze, and M. Yang, "hextris." https://github.com/hextris/hextris/, 2015.