Amazon Web Services (AWS) Workshop

Vitaly Ford | fordv@arcadia.edu | https://vford.me

Arcadia University

Table of Contents

NSF#: 1827050. Last update: January 27, 2019.

Executive Summary

This report introduces how to start with AWS Educate for students and faculty at Arcadia University. One of the major AWS services, EC2, will be described in detail in order to provide a better understanding how to create your own virtual machines in the cloud and connect to them using a terminal and graphical interface. Cloud9 will be mentioned at the end of the report. If there is a need for describing how to work with other AWS services, I will be glad to add that here as well – let me know.

Faculty will learn how to create AWS classrooms for their students.

I can run hands-on workshops a few times per semester for interested people or meet individually to work on / discuss a specific problem when time permits.

The bottom line why cloud matters:

- Process, store, and share big data
- Use machine learning and other software on scale
- Create publicly available servers
- Deploy mobile, web, and other apps on scale

Why Cloud Computing?

AWS is one of the major cloud computing providers (some other ones are Google Cloud, Microsoft Azure, and IBM Cloud). AWS Educate allows faculty to get $200 worth of free credits every year to use AWS in research and teaching. Additionally, faculty can easily create classrooms for students, allocating for each of them $50 of free credits for every class (you can request more if needed for larger projects). A few reasons why cloud computing is useful for computer science and math:

- Running multiple virtual machines that are much more powerful than anything we have to compute, research, mine and analyze big data, and run complex machine learning models including:
    - Tensor flow
    - Natural language processing and text analytics
    - AWS DeepLens (smart IP camera)
    - Conversational interfaces based on Alexa (e.g., for chatbots or virtual reality guides)
    - Text-to-lifelike speech, streaming it directly to apps
    - Existing cloud API for automated image/voice/video processing and analysis
    - Translation services
- Running your applications (web, mobile, commerce, research, testbeds) on servers that will be accessible from the Internet to anyone.
- Having encrypted, secure storage for your data in the cloud.
- Downloading/uploading big data and working on it in the cloud at unlimited storage and computational power.
- Building virtual reality and virtual robotics applications.
- Working with databases and using those in teaching (e.g., AWS DynamoDB is a NoSQL database that allows for 25GB per month for free at any given time). AWS PostgreSQL and MySQL are

wonderful as well for teaching students how to use SQL-based databases and connect them with the apps.

- Creating private networking infrastructure with multiple virtual machines interconnected among each other (with routing, firewalls, etc.)
- Serverless computation (running your code without a server, paying only when your users actually browse your web pages / use your mobile apps): AWS Lambda.
- Having a cloud Integrated Development Environment that will be automatically set up for you without the need to install any software on your computer: AWS Cloud9.

Who Can Benefit from Cloud Computing at Arcadia?

- Anyone who works with big data processing on campus will benefit from cloud computing.
- If you need to install a specific software that is not feasible to install on laptops and lab computers (e.g., galaxy server for bioinformatics), then cloud computing is the way to go, especially if you want to make it available at high speeds to all your students.
- Anyone who is interested in running machine learning models on any kinds of data (text, audio, video, images) but does not want to tinker with setting up the software on the personal or lab machines, especially if you are not comfortable with setting up and configuring machine learning software on your own (the cloud will do that for you).
- Anyone who has only one license for specific software and wants to allow others to use that software.
- Anyone who wants to create web and mobile servers and applications that would be publicly available on the Internet.
- Anyone who is going into the field of IT/CS.

Where Arcadia Students Learn about Cloud Computing

I teach students cloud computing (AWS Educate and sometimes Google Cloud) in CS358 Operating Systems and CS362 Computer Organization and Architecture courses. The students who took one of those courses will be prepared to work in the cloud on their own. This skill could be especially useful in the Capstone projects.

AWS Educate Registration

You can start the registration process here https://www.awseducate.com/registration. Choose Educator or Student depending on who you are at your Institution.

Students

Registration

During the registration process, students should choose a Starter account if they are applying separately from a classroom. They will be given $75 worth of AWS credits per year until they graduate.

Educators

Registration

Educators will be asked to add a credit card to the account. It will not be charged. However, you should be aware that if you run out of the $200 credits that are given to you every year and you continue using the AWS resources, you will be automatically charged on your credit card.

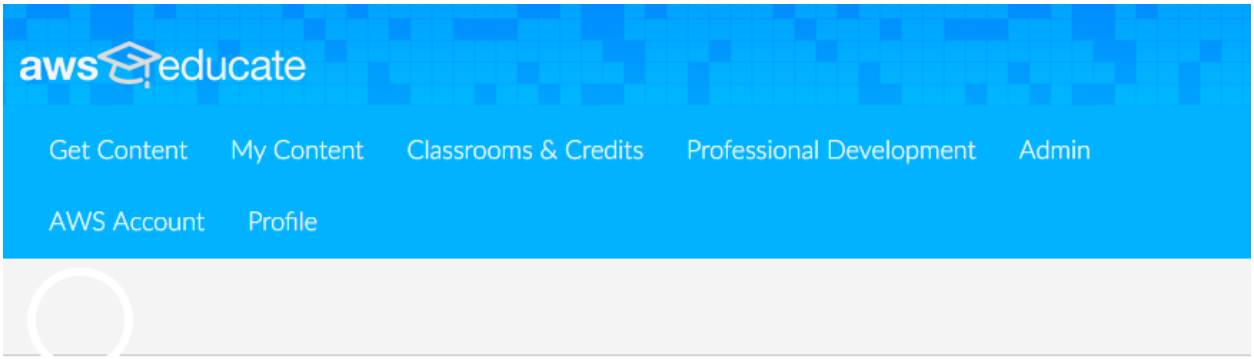
If the charges are accidental, you can dispute those and AWS has been lenient to withdrawing the charges if those happened by mistake (e.g., you forgot to shut down the server and it ran for a month). The bills come every month in case if your credit card is being charged.

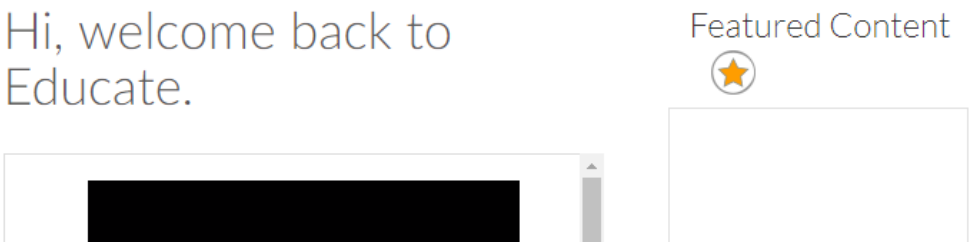
Classrooms

After registration, Educators can request classrooms in their accounts and when approved (it may take a few days), Educators can upload an email list of students in that classroom. Afterward, the students will receive separate emails from AWS with the instructions on how to proceed, without the need to register on awseducate.com beforehand.

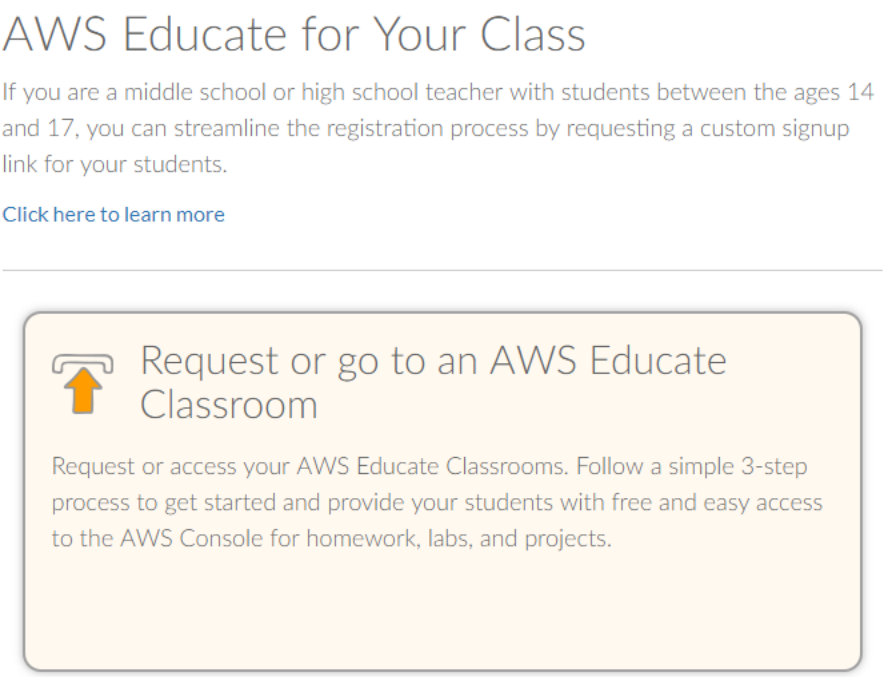
Create Classroom

Login to awseducate.com. Click on Classrooms & Credits.

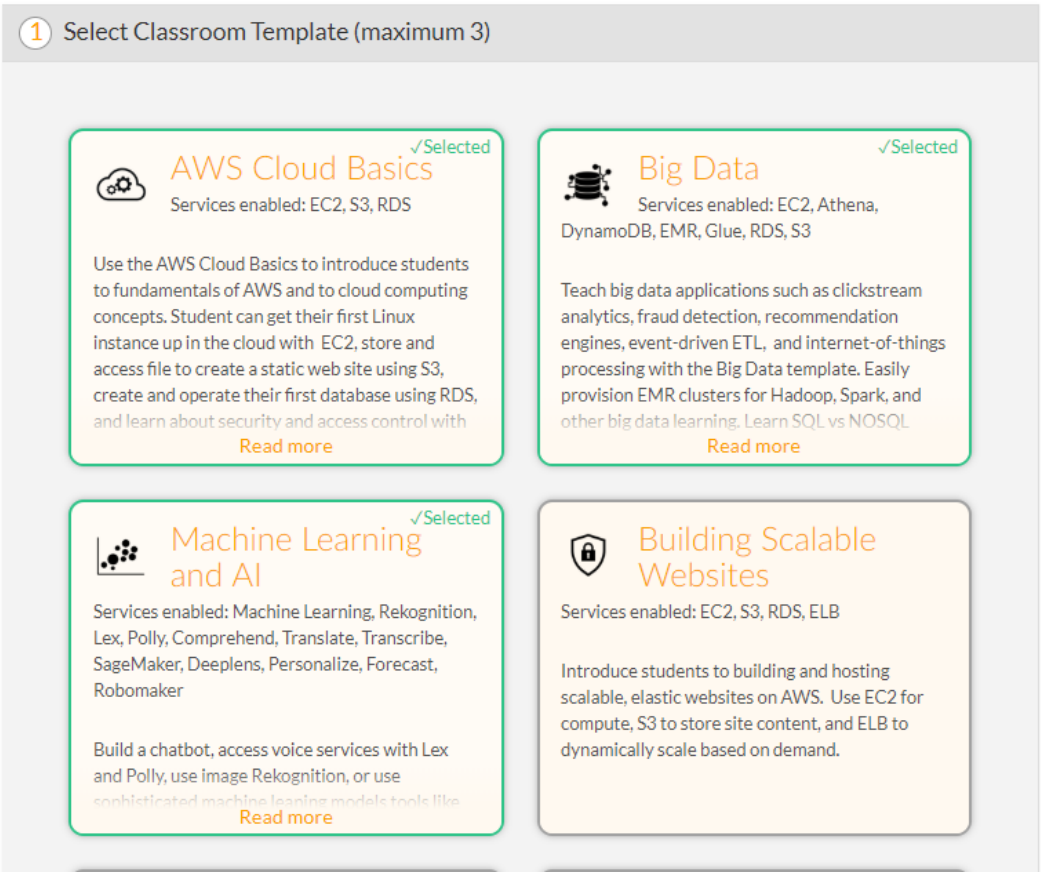




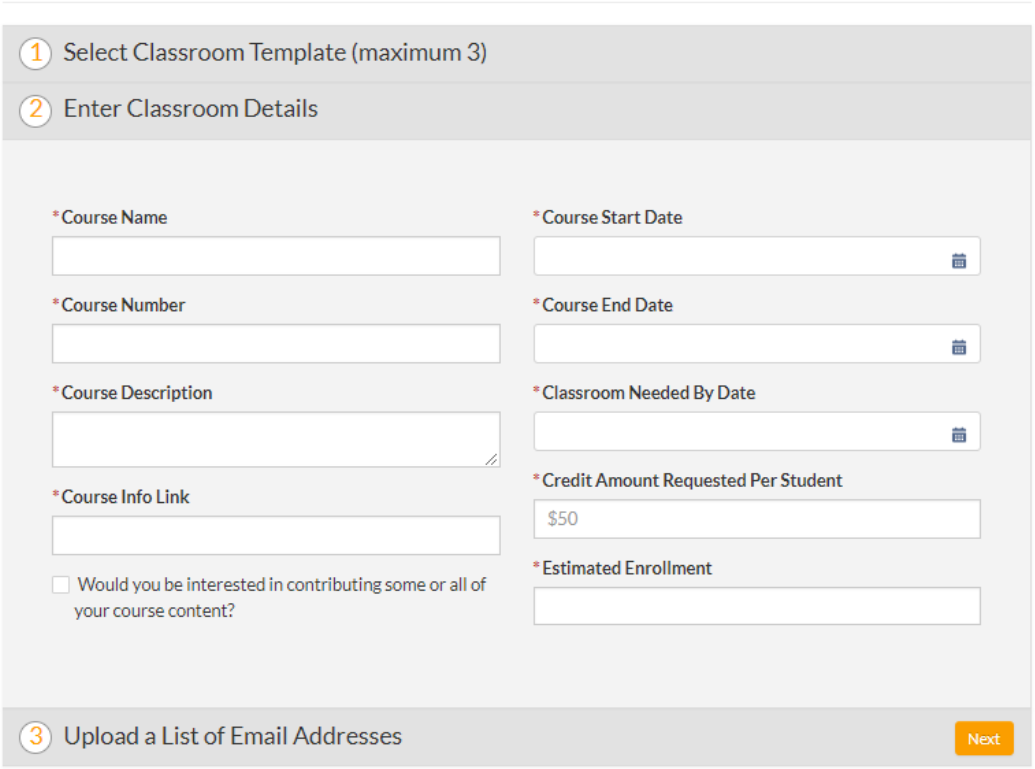
Click on “Request or go to an AWS Educate Classroom”.



Select up to 3 classroom templates that define what services will be allowed for students to use. E.g., AWS Cloud Basics, Big Data, and Machine Learning and AI.



Click on Next at the bottom and then fill up your classroom's information. You can have as many classrooms as you want but you will need to redo these steps separately for every classroom.

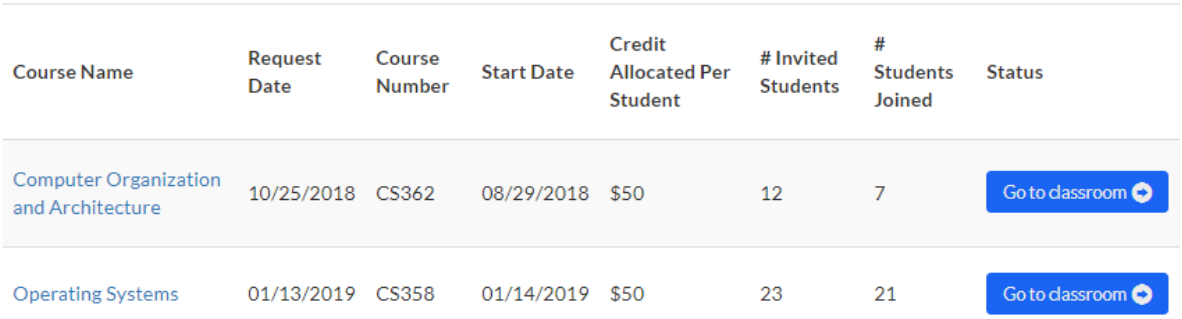


After you filled up all the information with $50 credit requested, click Next. Then download the template that they provide (which is just an Excel spreadsheet with one column, in which you will insert the email addresses of your students, one per row). Fill up the template and upload it on that page. Finish the application and you will receive an email when it is ready to go.

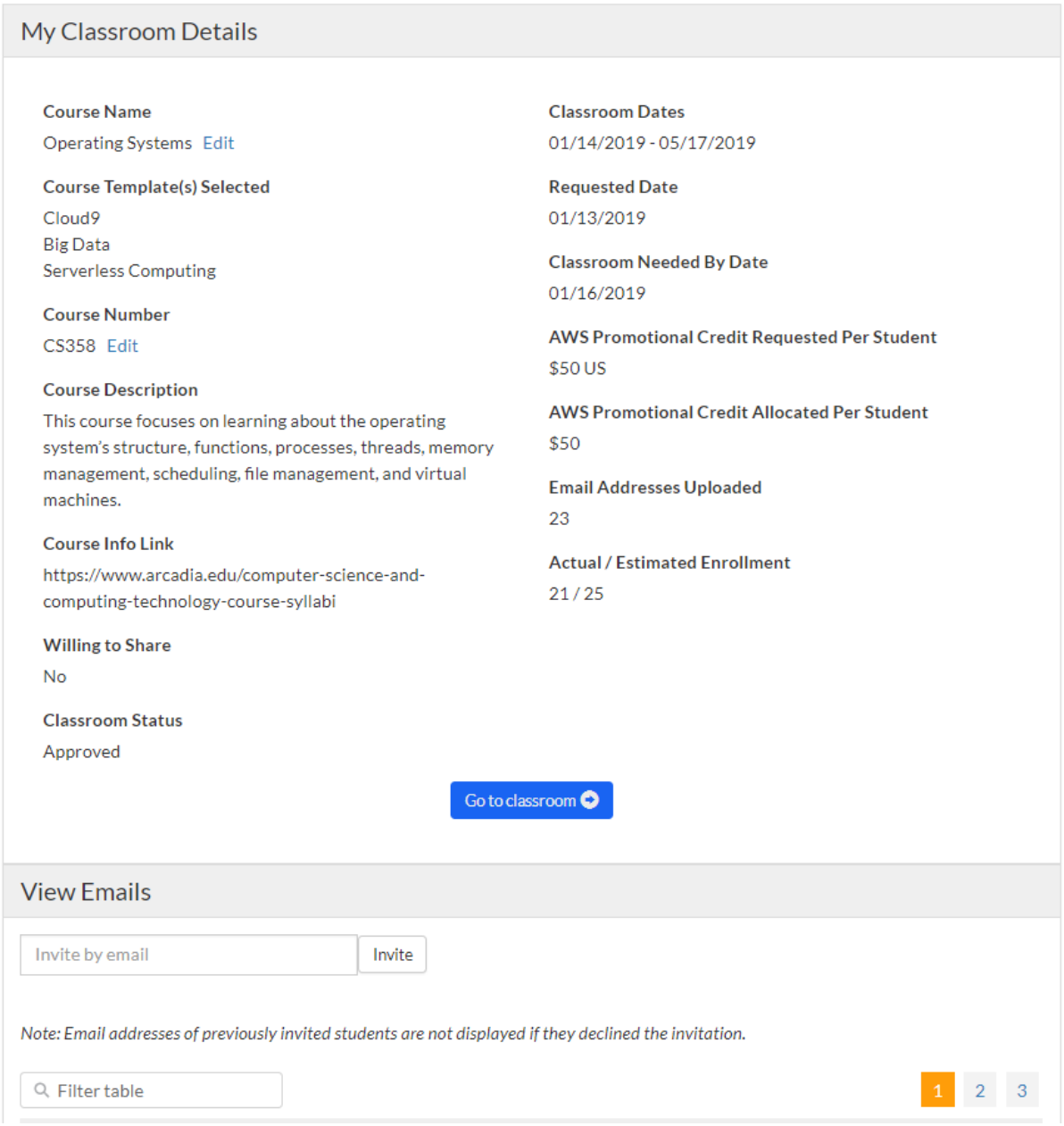
Add/Remove Students & Resend Invite

Your students will automatically receive an email from AWS but if someone did not or you want to add another student, you can do so as well in the created classroom. To do that, go to "Classroom & Credits" and select "Request or go to an AWS Educate Classroom". Click on "Go to my classrooms" link that is on the right side of the page.

Click on the appropriate classroom where you want to add the student. If you want to see how much money the students have used, you click on "Go to classroom". But if you want to add the student, you click on the name of the course (in my case, it is "Operating Systems").



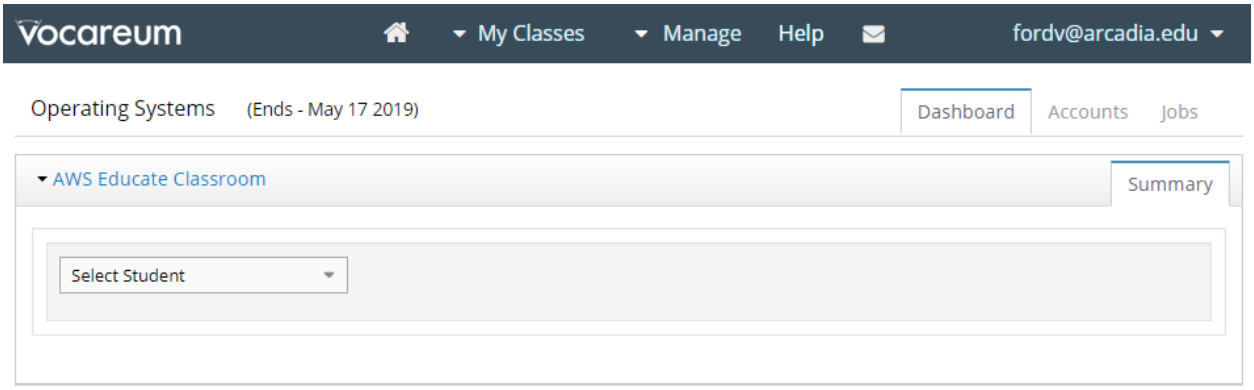
You should see something similar to the following unless you selected "Go to classroom" from the previous page:



At the bottom, you can invite by email, resend the invite, or remove students from the classroom. You will also be able to add student emails in bulk at the very bottom of that page.

View Resource Allocation and Credits Left per Student

If you clicked on "Go to classroom", it will tell that you will be redirected to Vocareum website. After you accept, it will show you your classroom.



If you want to see how much money every student used per day, you can click on Accounts.

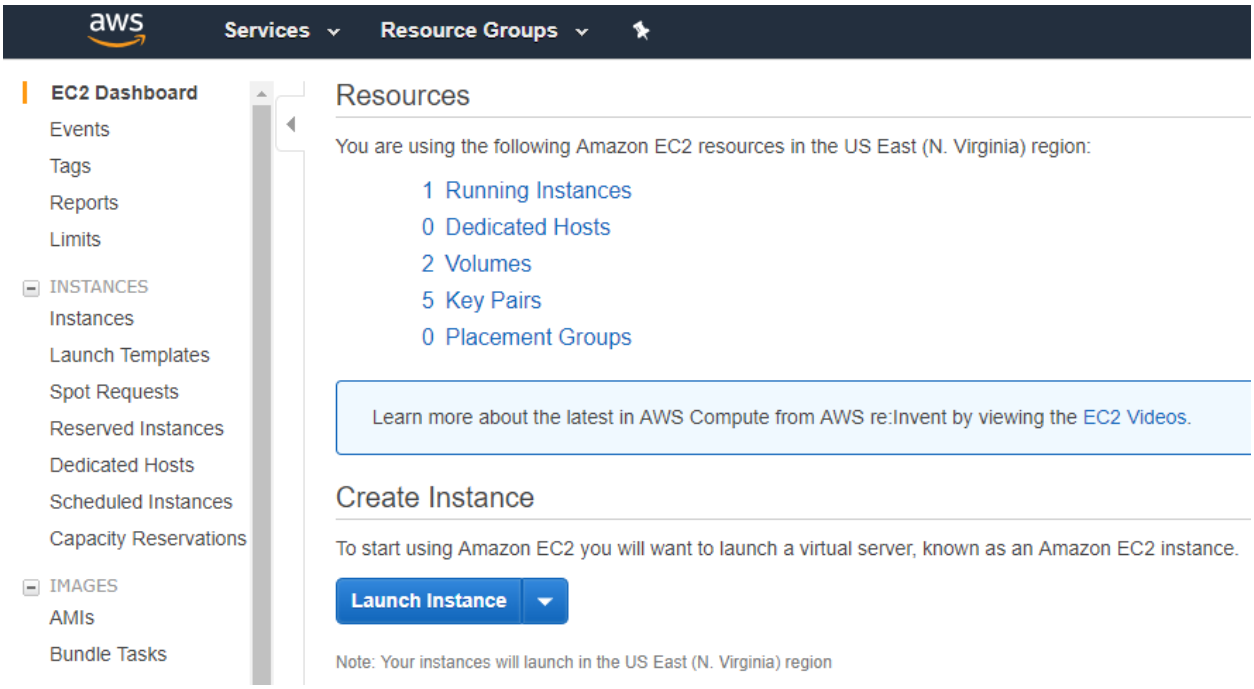
Computations and EC2 Virtual Machines

Introduction

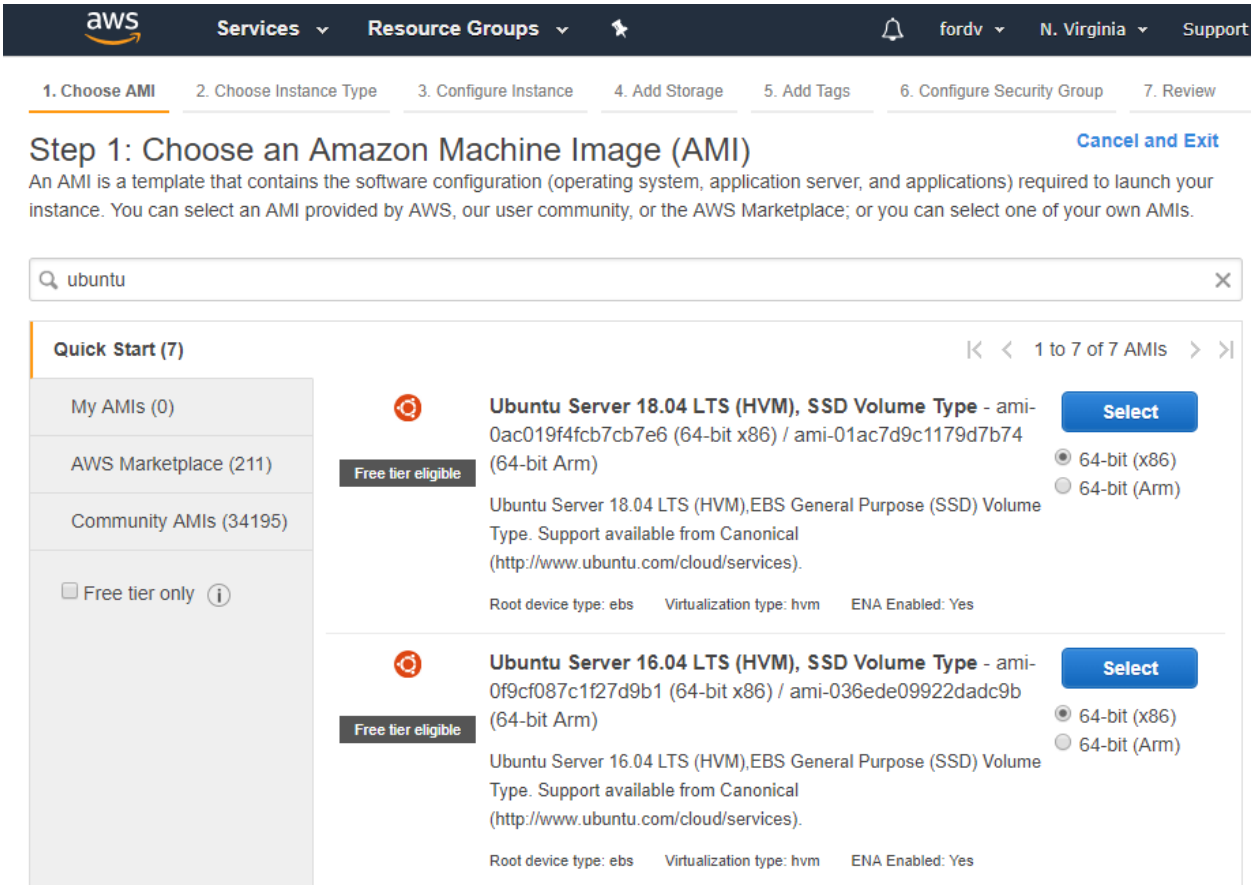
EC2 is a service that allows us to create virtual machines with almost any hardware configurations we want in a blink of an eye. Moreover, if at some point during our work we want to change the current hardware configuration without losing our data, it is as simple as shutting the machine down, changing its hardware properties to the ones we want, and booting it back up. It takes just a few minutes.

Launching an Instance

This is the EC2 dashboard that shows your current EC2 resource usage and allows you to launch a new virtual machine (instance):

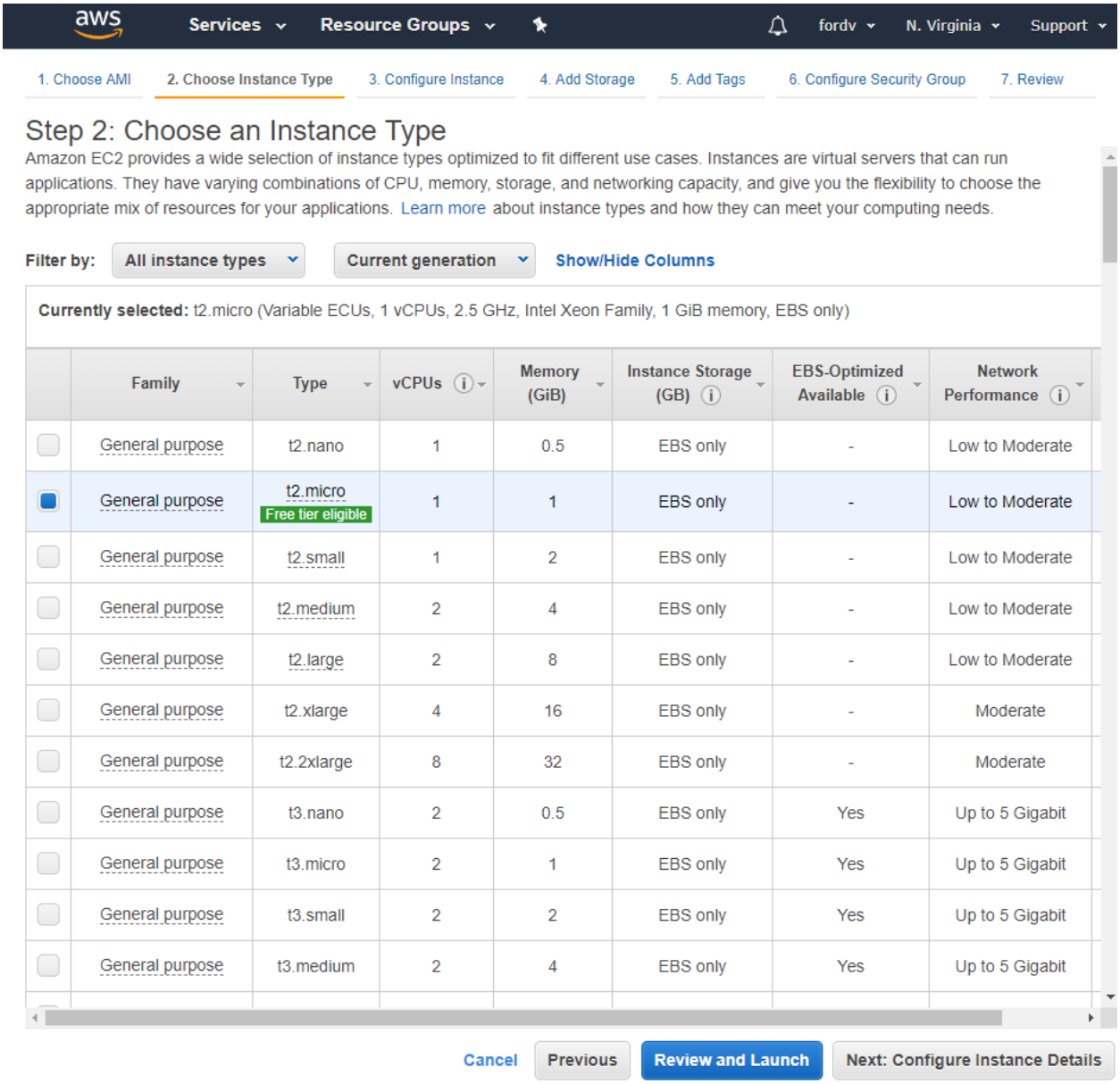


After clicking on Launch Instance, you are brought up with a page that asks you select a template to run your server. Most commonly used templates are Ubuntu Server (for Linux side) and Windows Server (for Windows side). We will select Ubuntu 18.04 by searching for it in the search field:

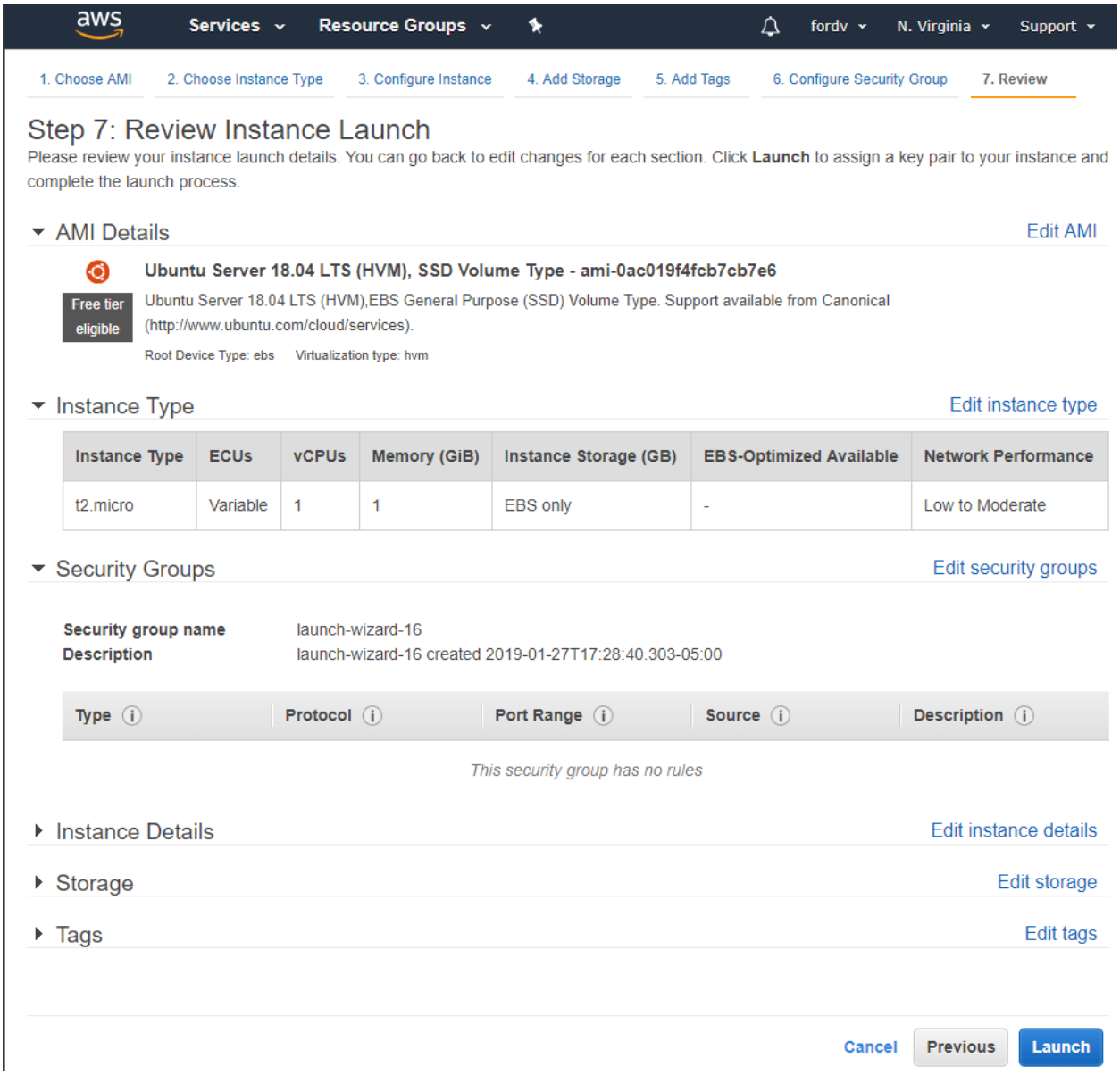


Next, you can select the hardware configuration that you want this server to use. By default, type t2.micro is selected that has 1 processor and 1 GB of RAM (hard disk / SSD will be chosen later). The types vary by the number of processors and RAM available on that machine. The more, the costlier. You can read more about pricing here: https://aws.amazon.com/ec2/pricing/

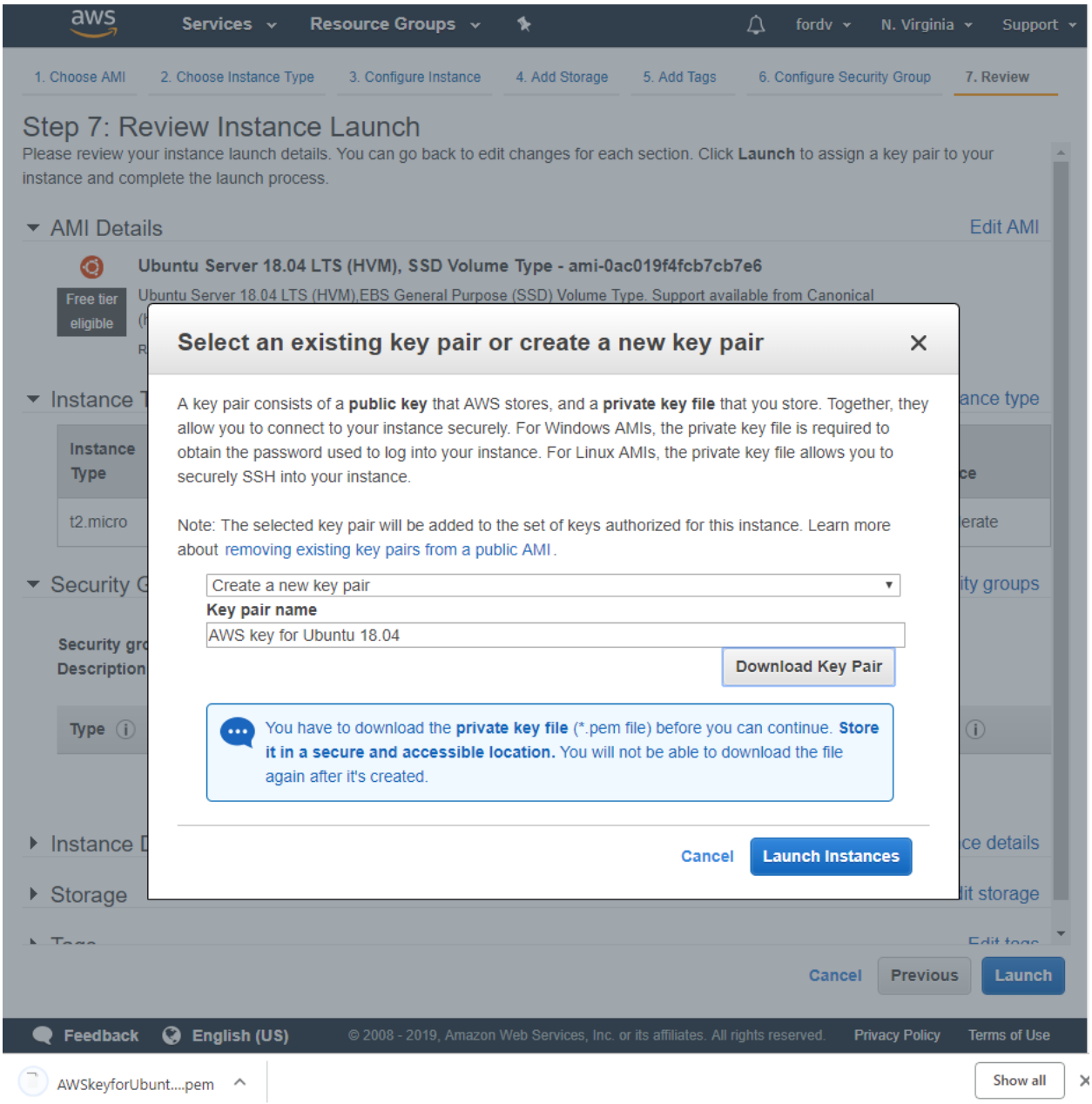
As an example, a machine with 1 processor and 1 GB RAM costs 0.0116/Hour whereas a machine with 96 processors and 384 GB RAM costs $4.608/Hour. We will go with the default configuration.



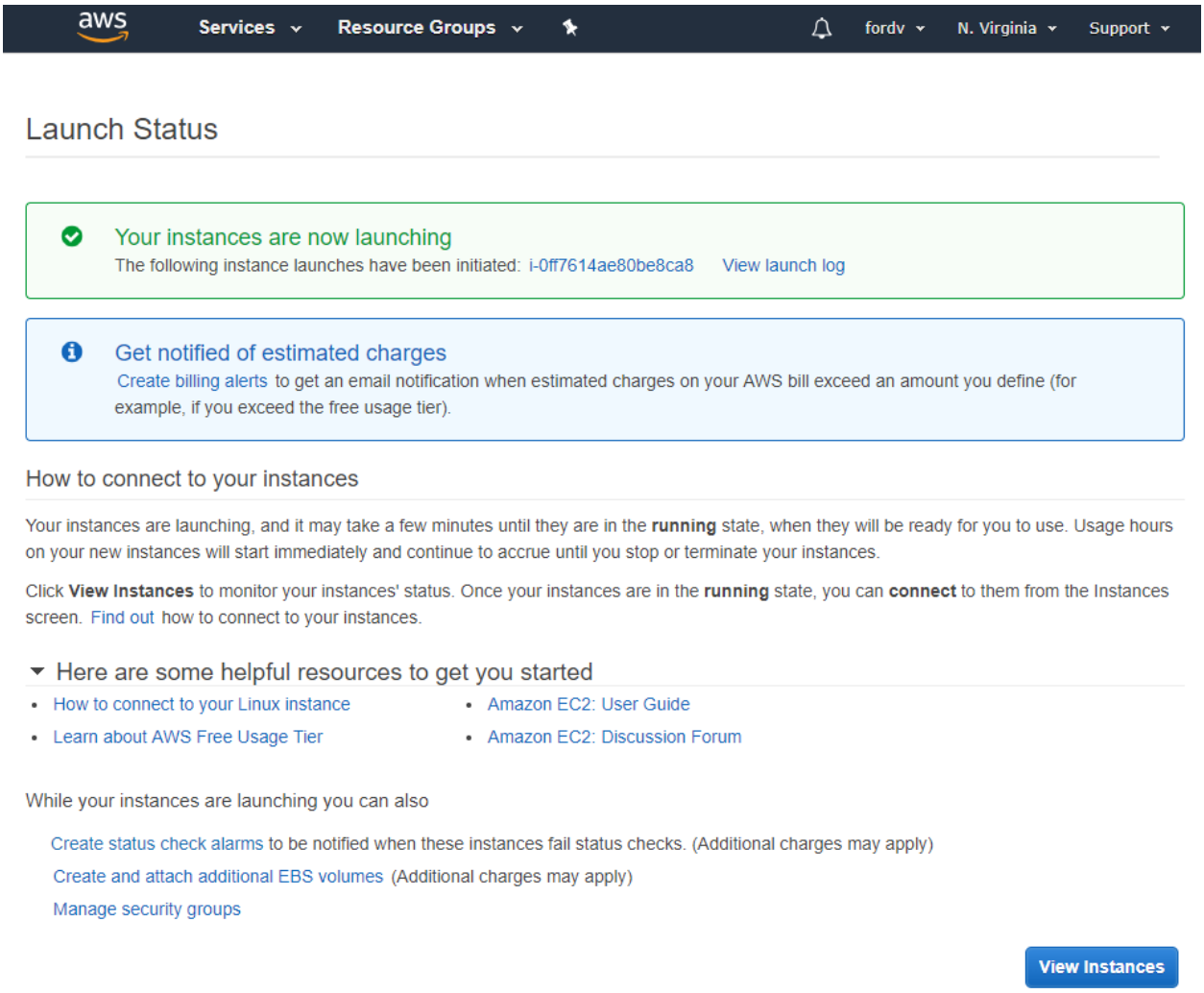
You can click on Review and Launch at this point unless you want to check out some of the other hardware configuration options like networking and storage space. Click on Launch when you are ready.



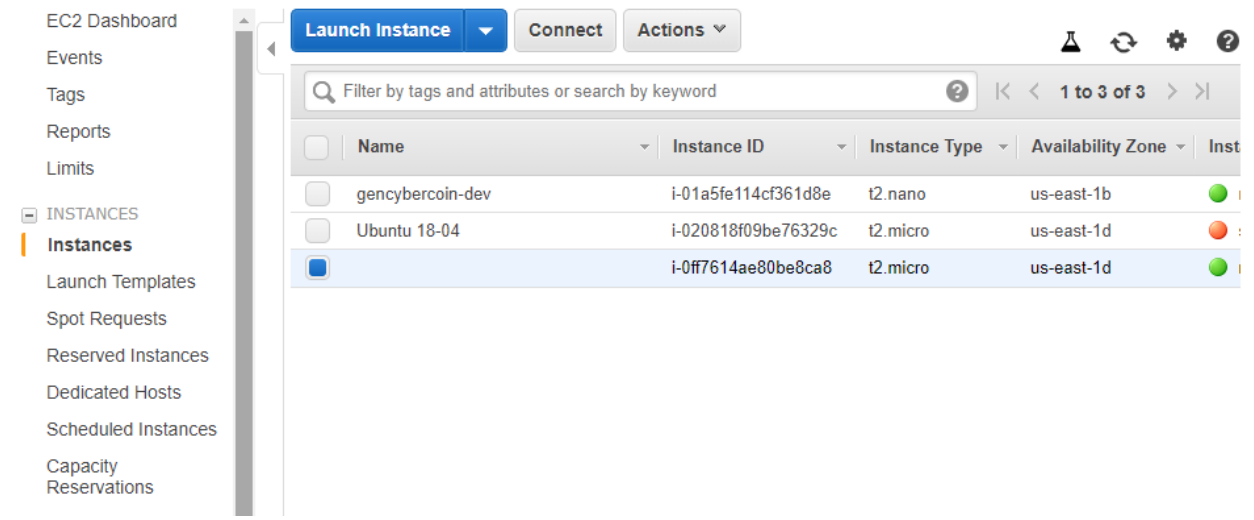
At this point, it will ask you to create and download a private key (unless you want to reuse the one you have already had). This key will be used to authenticate you on the server when you try to connect to it. Select an option to create a new key pair, give it any name that you want, and download it:



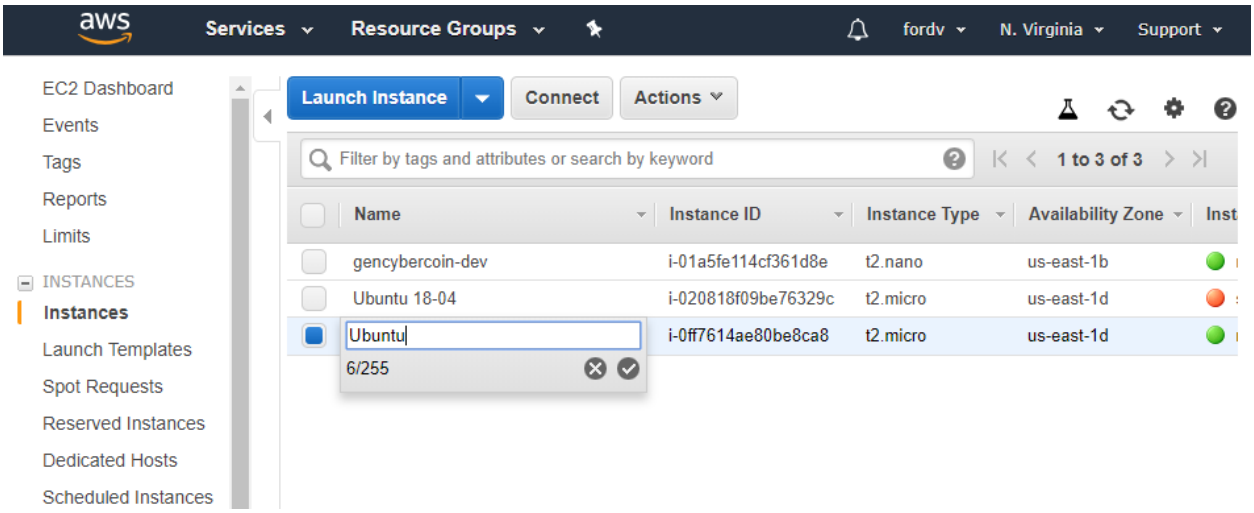
Launch the instance after you downloaded the key.



You can now click on View Instances and wait until the instance is created. It will bring you to the EC2 page and in the Menu on the left, you will see that AWS selected “Instances” for you. On this image, you will see 3 instances because I had previously created other machines.

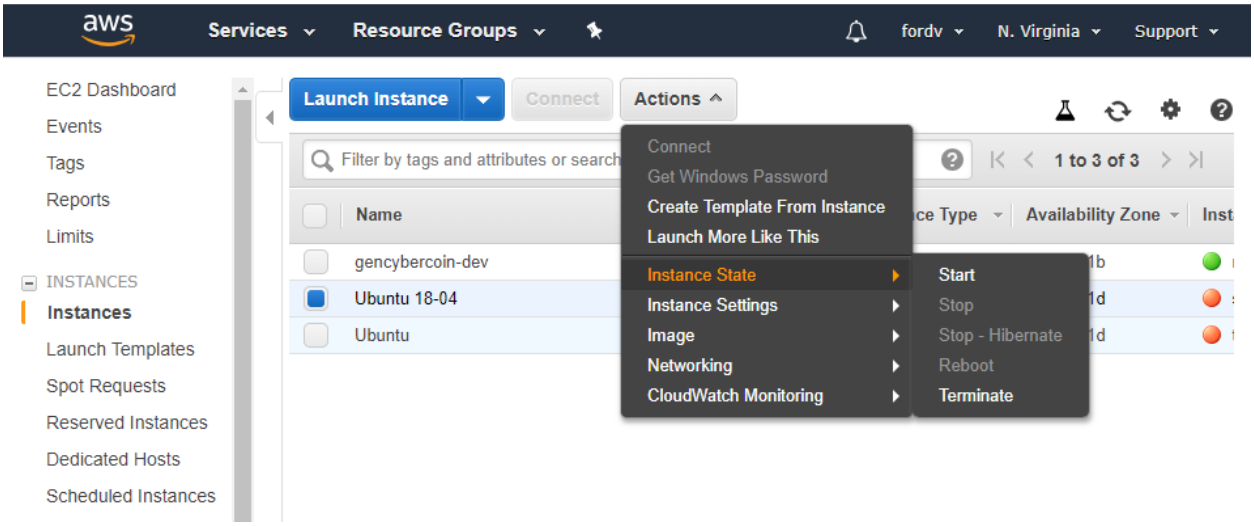


Our newly created machine does not have any name yet, so click on it and enter the name that you will recognize later:



Starting/Stopping/Terminating Instances

IMPORTANT: do not forget to stop your machine when you are not using it so that you will not incur any charges (storage is dirt-cheap on the cloud but computation does cost money). To do that, select the machine, then click on Actions. Select Instance State and Stop/Start depending on what you want to do at the moment. If you want to delete the machine completely (meaning that it will be gone forever), select Terminate in the Instance State.

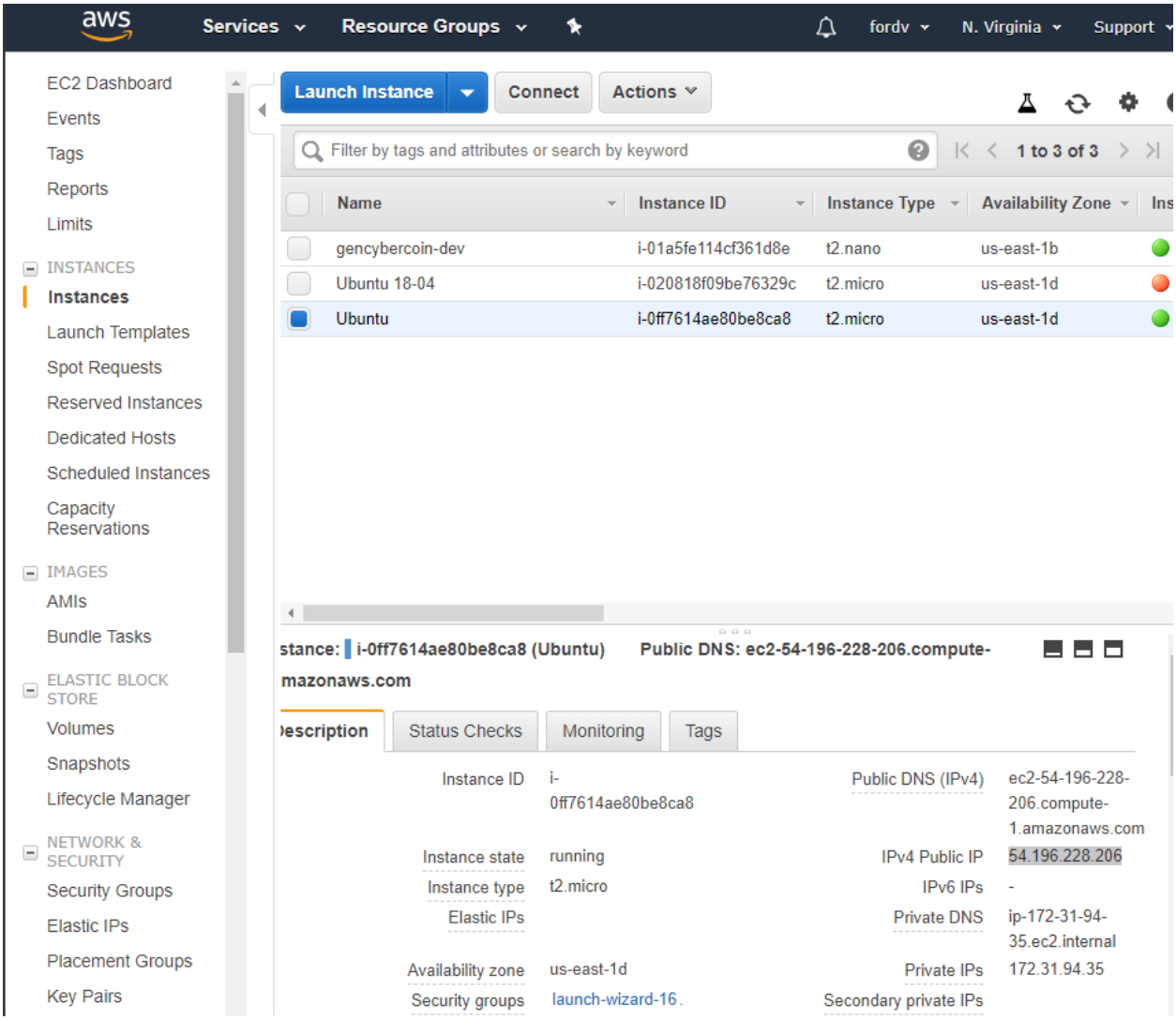


Access the Instance

Now that you created the instance and called it Ubuntu, you can connect to it by using the key you have previously downloaded and software called SSH (Secure Shell). If you are currently working on Mac OS or Linux, you will be able to enter the following command in the terminal (after navigating the terminal to the folder where your key is located):

```
ssh -i AWSkeyforUbuntu1804.pem ubuntu@public_ip_address
```

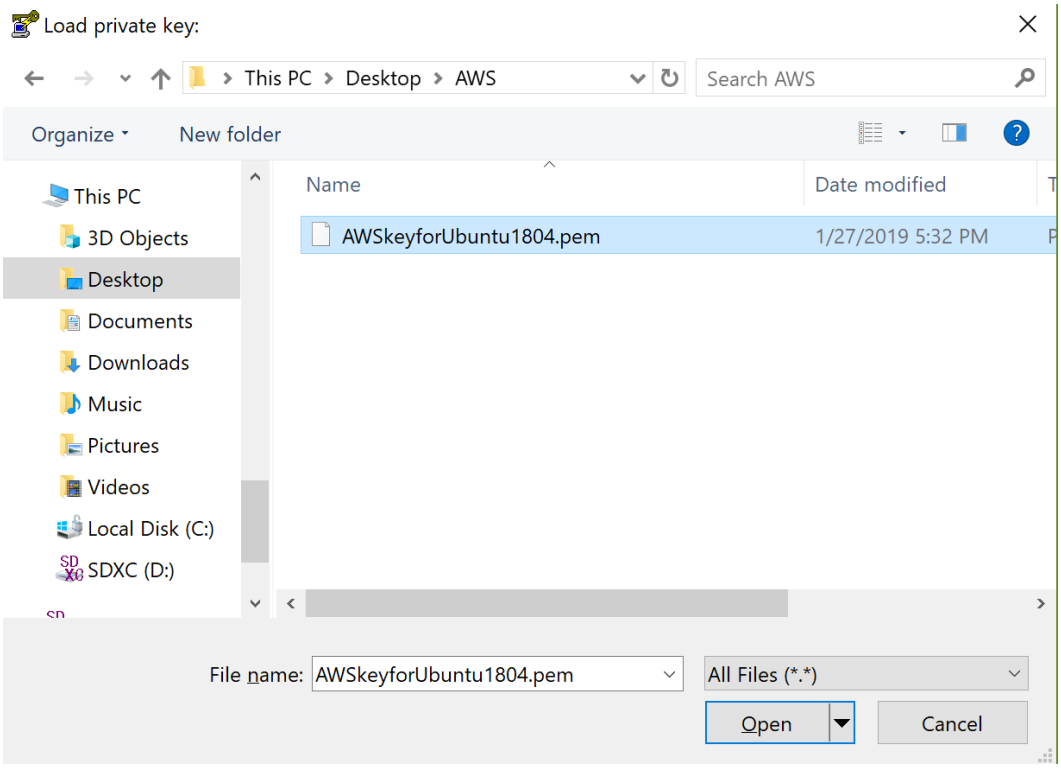
where `public_ip_address` is the machine's public IP address found at the bottom right when you select the machine in EC2 (in my case, it is 54.196.228.206); and ubuntu (preceding @ sign) is the default username.



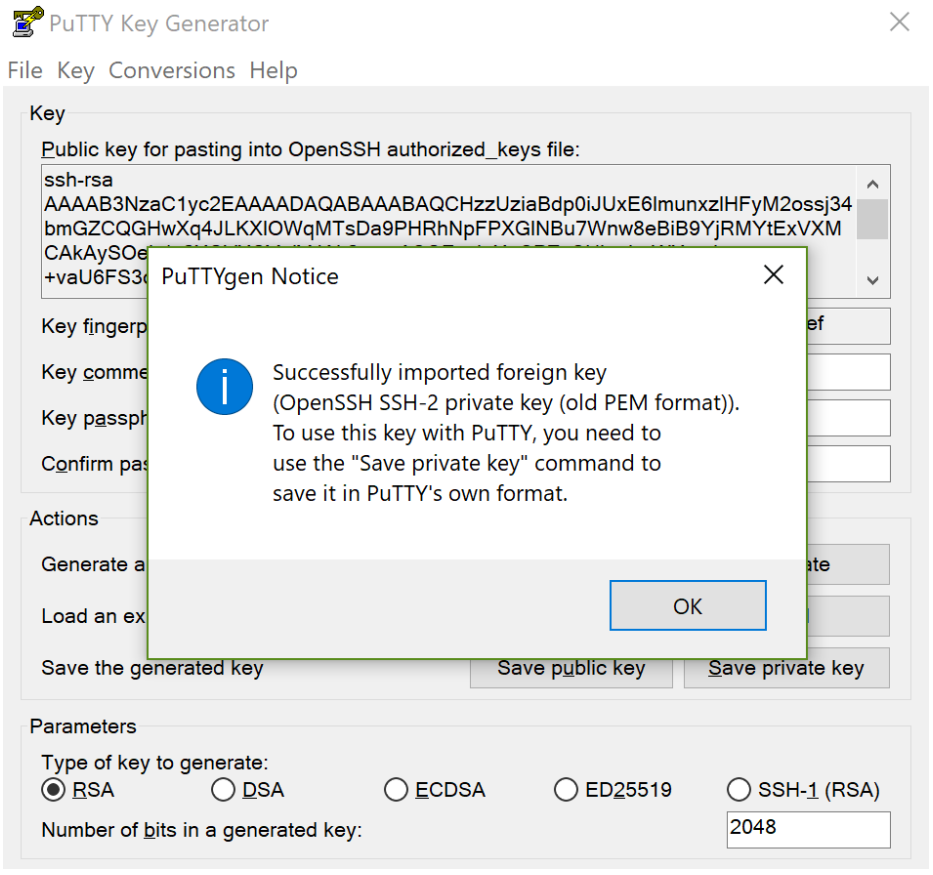
If you are on Windows, then you will need to either enable SSH in the Optional Features available in the Apps and Programs or use puttygen to generate the putty key and then use putty to connect as described below.

Download puttygen and putty from https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html (64-bit will be fine unless you are using an old machine that is 32-bit). Open puttygen and Load the key that you have downloaded from AWS when you created the Ubuntu instance.

Make sure that when you click on Load in puttygen, in the popped up window you select All filed in the bottom right corner to show all files (to actually see the key itself, otherwise it will not show it to you).



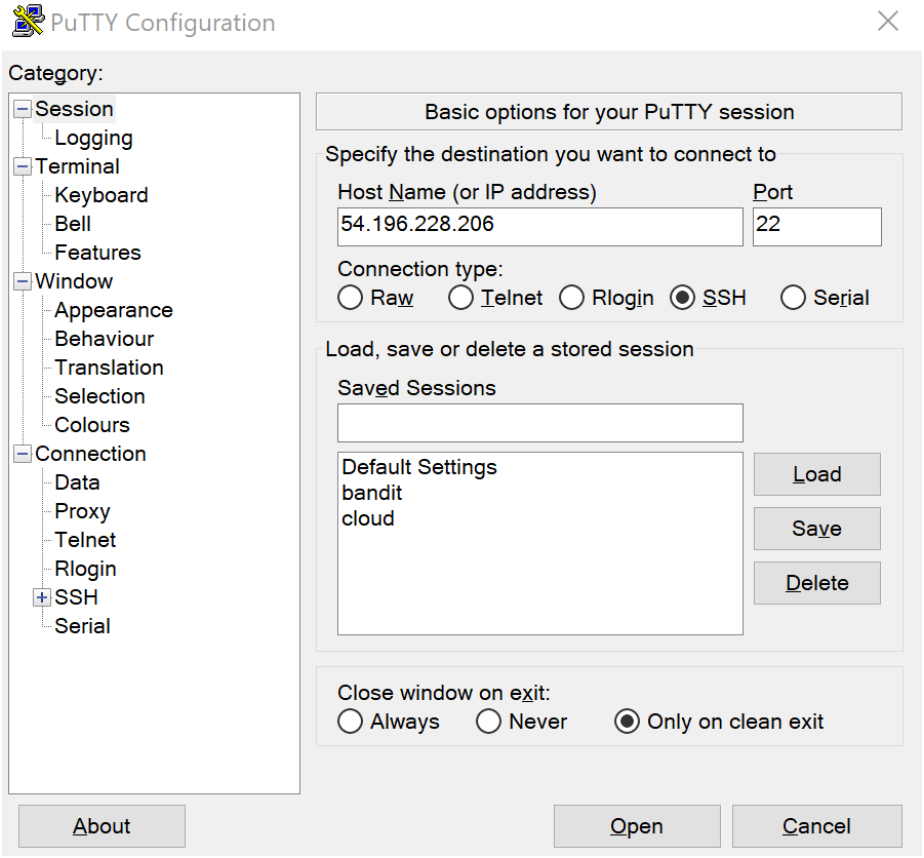
It should show you this window next:



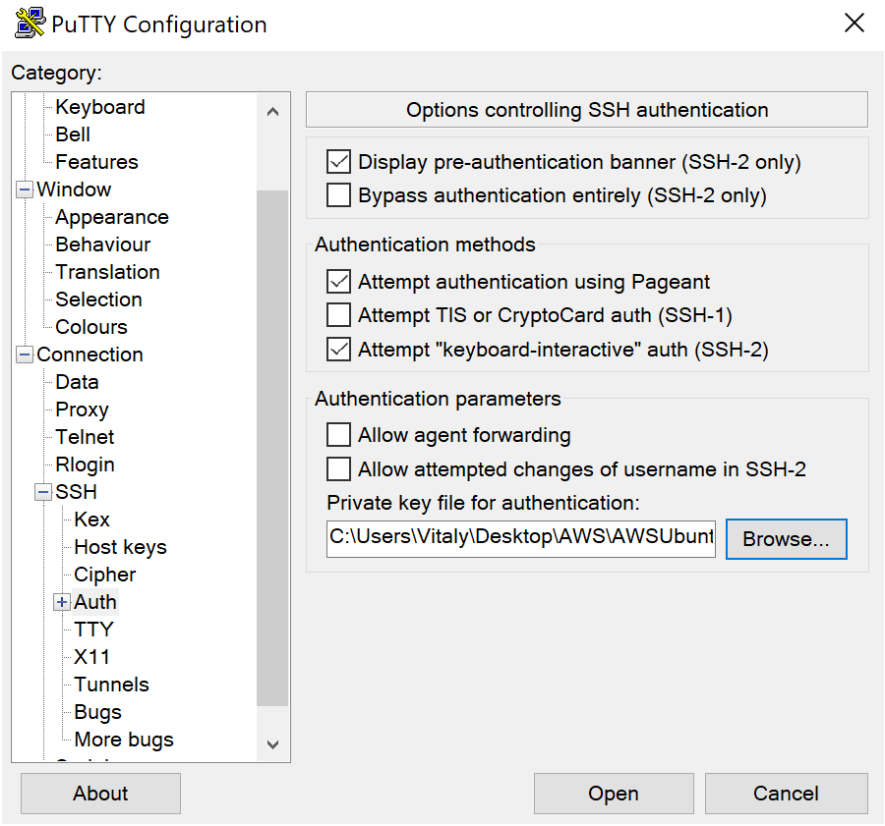
Click “OK” and then click on “Save private key” (you can save it to the same folder where your AWS key is because technically, they are the same keys but in a different format: one is formatted for SSH client (extension .pem) and another for putty client (extension .ppk).

It will ask you if you want to save it without the password, say Yes. You can save it with a different name like AWSUbuntu_putty.

Close puttygen and open putty. In putty, enter your machine’s IP address in the appropriate field:

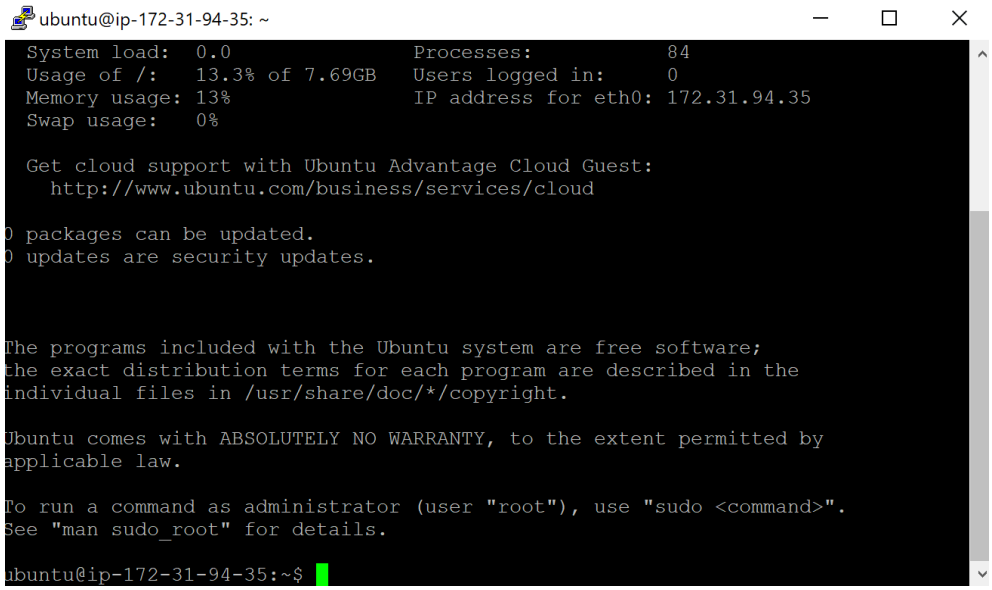


Then navigate to SSH on the left menu in putty, click on Auth and on the right side select “Private key for authentication” which should be your key generated by puttygen (AWSUbuntu_putty.ppk).



Click on open. Select “Yes” to the security alert. Enter the username as ubuntu:

Press Enter and you should be in:



Set up VNC and remote desktop for Ubuntu

If you want to set up a remote desktop with a graphical user interface for connecting to your machine instead of using the terminal, following this section. Otherwise, skip it.

Installation

Connect to your Ubuntu machine using putty on Windows or SSH command on Mac OS / Linux. VNC server is the software that allows you to connect remotely using a graphical user interface.

Type the following commands in the terminal to download the software and install it on your server:

```
sudo apt-get update
sudo apt-get install ubuntu-desktop
sudo apt-get install gnome-panel gnome-settings-daemon metacity nautilus gnome-termin
al
sudo apt-get install vnc4server
vncserver
```

Enter your password you will use for the remote desktop. After that, it will tell you that vncserver has started.

Configuration

Now, type the following to kill it and then edit the settings.

```
vncserver -kill :1
nano .vnc/xstartup
```

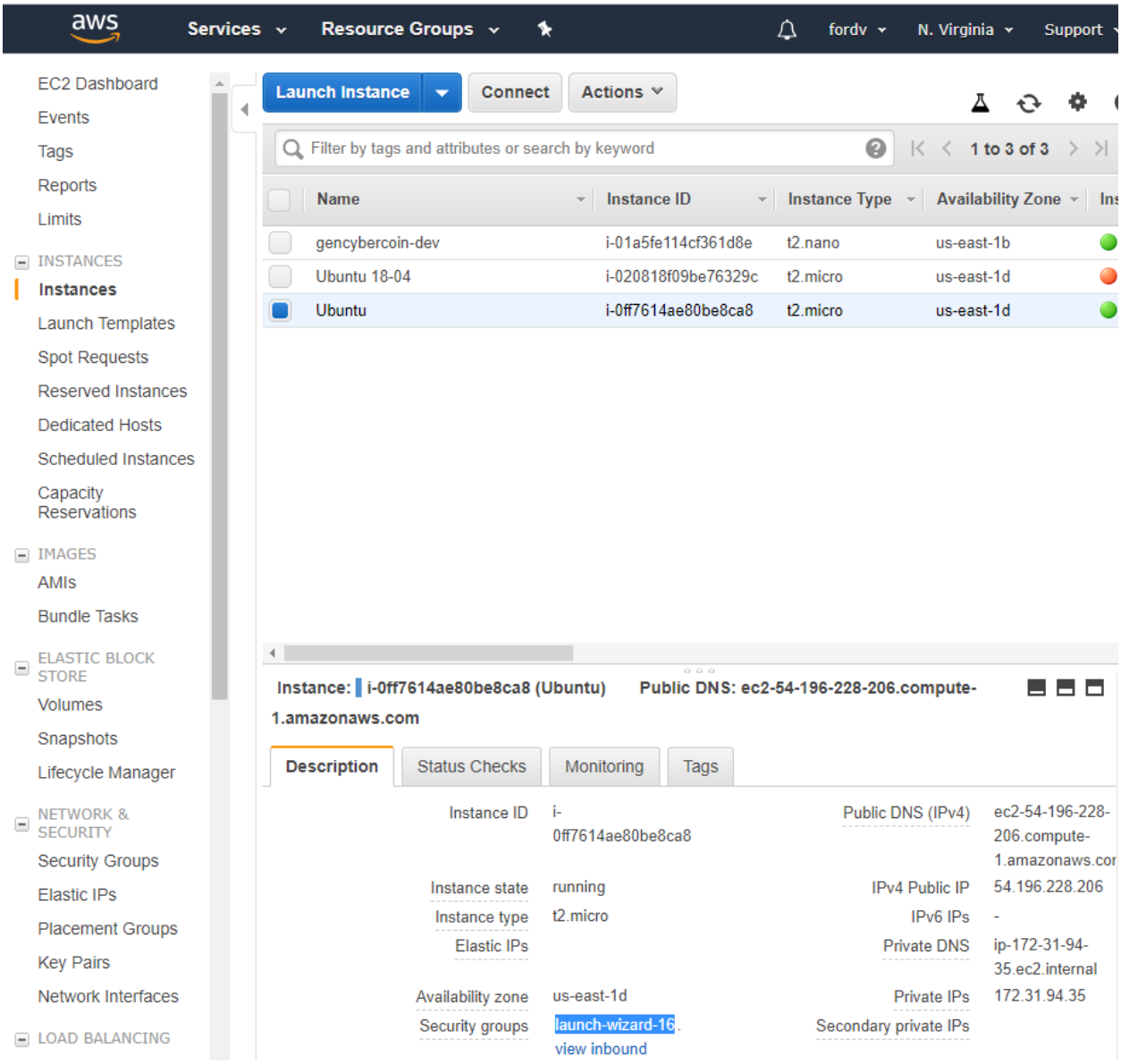
Delete everything from the file (using Delete or Backspace keys and keyboard-arrows to do it) and write the following in it (copy from here and right-click -> paste):

```
#!/bin/sh

export XKL_XMODMAP_DISABLE=1
unset SESSION_MANAGER
unset DBUS_SESSION_BUS_ADDRESS

[ -x /etc/vnc/xstartup ] && exec /etc/vnc/xstartup
[ -r $HOME/.Xresources ] && xrdb $HOME/.Xresources
xsetroot -solid grey
vncconfig -iconic &

gnome-panel &
gnome-settings-daemon &
metacity &
nautilus &
gnome-terminal &
```

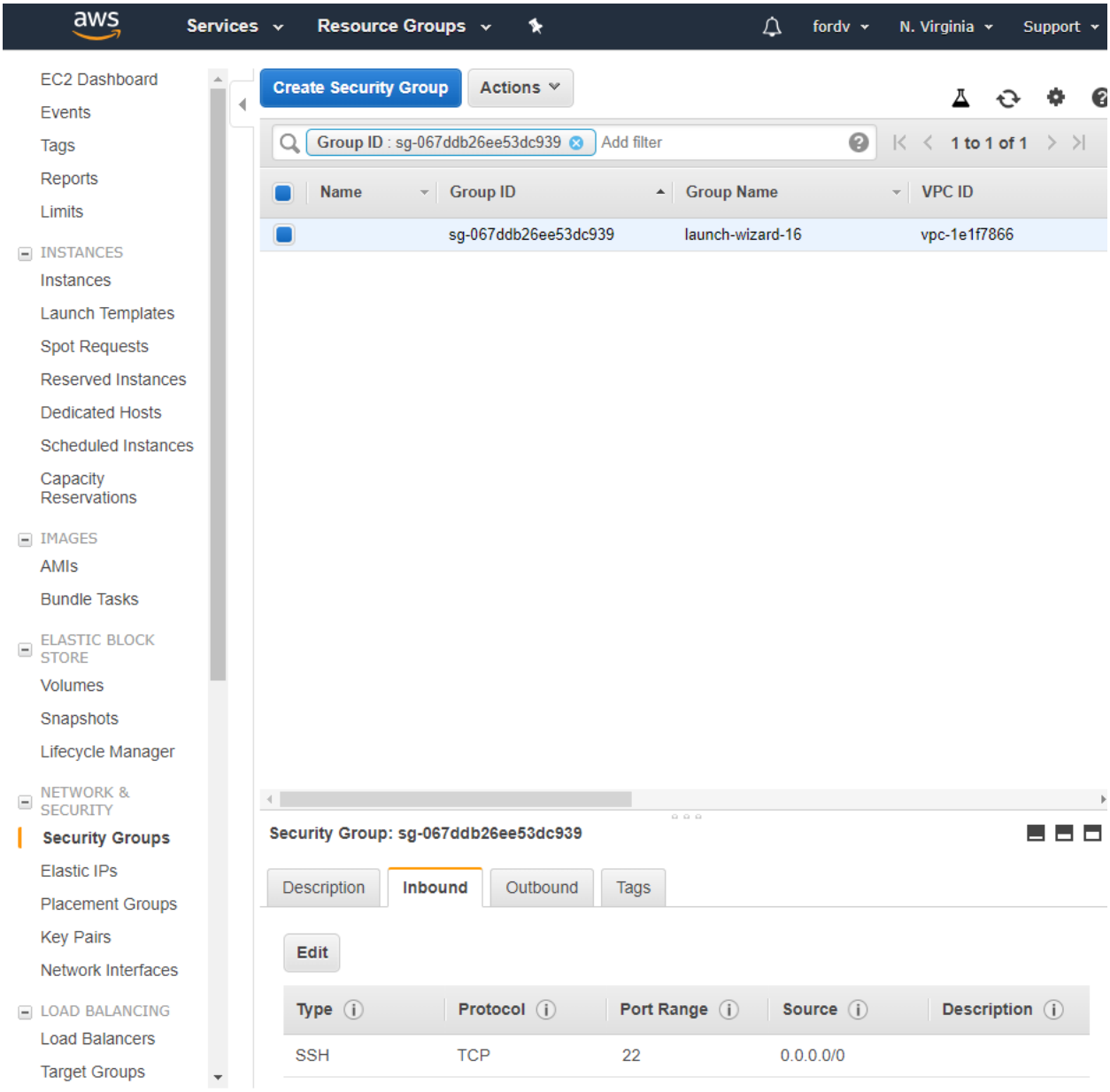
Press CTRL + X (without the plus, just CTRL X) on your keyboard. At the bottom of the terminal text editor, you should see a message asking if you want to save the changes. Press Y. Now, it should ask you where you want to save the file. By default, it will show you the name of this file in any case, so just press ENTER again.

Firewall Rule on EC2

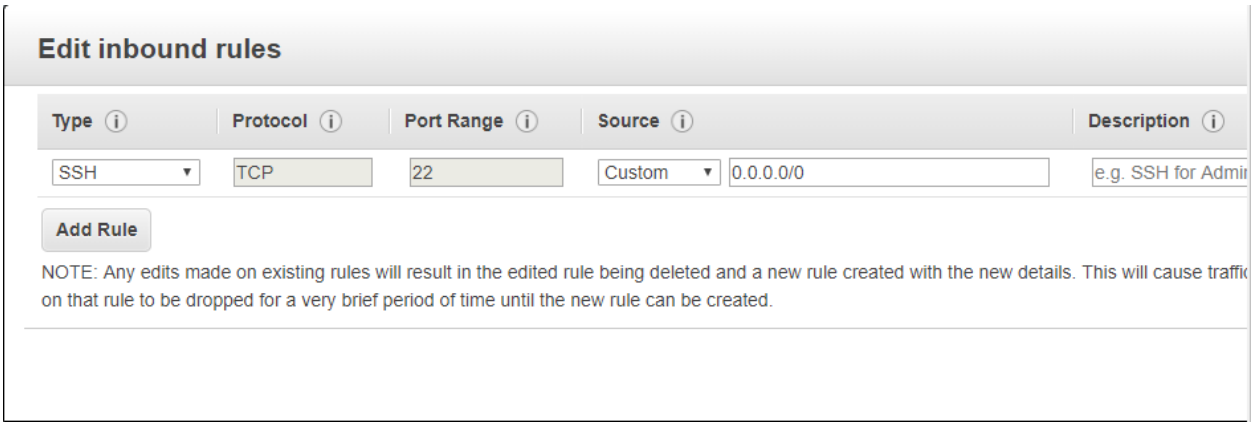
Add a firewall rule on EC2 for this machine to open port *tcp*:5901. To do that, open EC2 Instance, select your instance, and click on the “launch-wizard-..” at the bottom under Description -> Security groups:



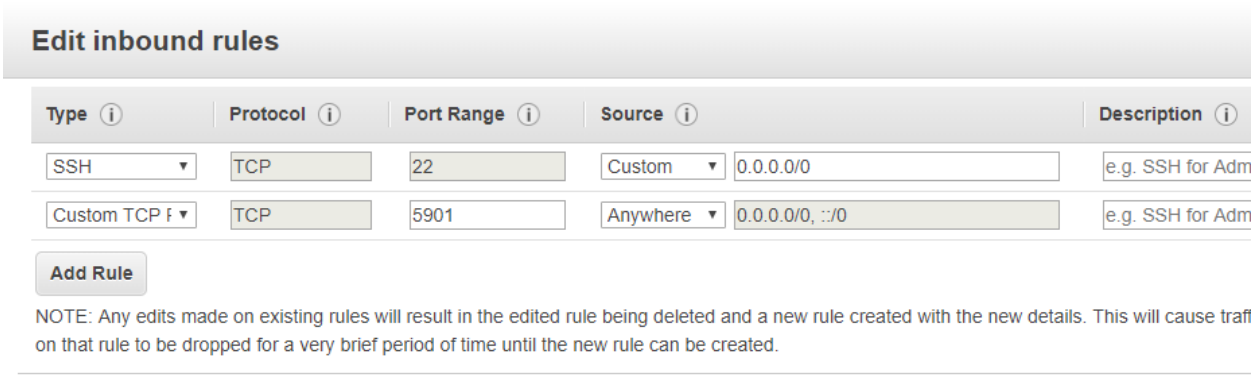
Next, you will see a page with the security group configurations. At the bottom, click on Inbound and select Edit:



After selecting Edit, you will see something like this:



Click on “Add Rule” to allow clients to connect to your VNCserver on your Ubuntu and add the following rule:



Click on Save.

Run VNC Server

Go back to the terminal where you have been working on installing the VNCserver and editing the configuration file on Ubuntu. Type the following command and press ENTER. It will launch vncserver again; this time you can also specify what resolution you want it to work at via '-geometry' parameter.

```
vncserver -geometry 1280x720
```

It should be up and running and waiting for you to connect to it. On your current machine, install Portable TightVNC (or RealVNC if you have admin privileges). You can use any other VNC client that is applicable to your operating system. Chrome has a VNS plugin that will also work as your browser app.

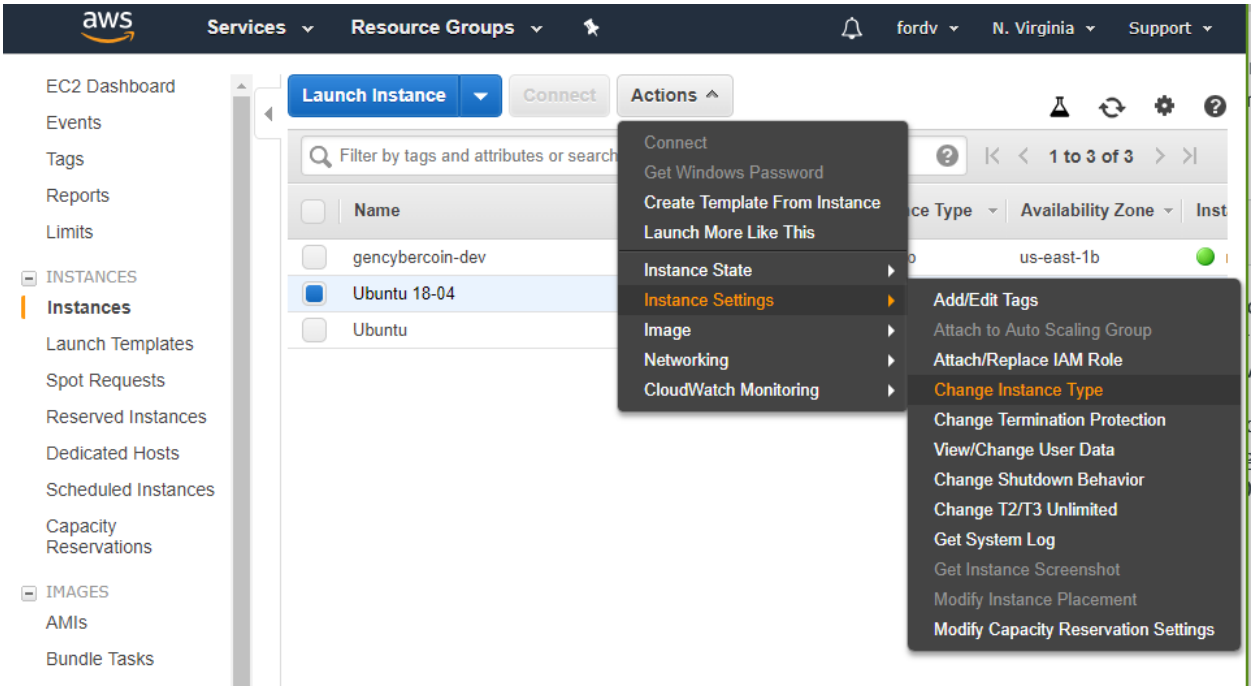
Connect to your ubuntu through the VNC client of your choice, use the public IP address of the Ubuntu machine bundled with **::5901** at the end (do not forget that double-colon in front of 5901). If double colon did not work, try just one colon in front of 5901.

EC2 Hardware Changes

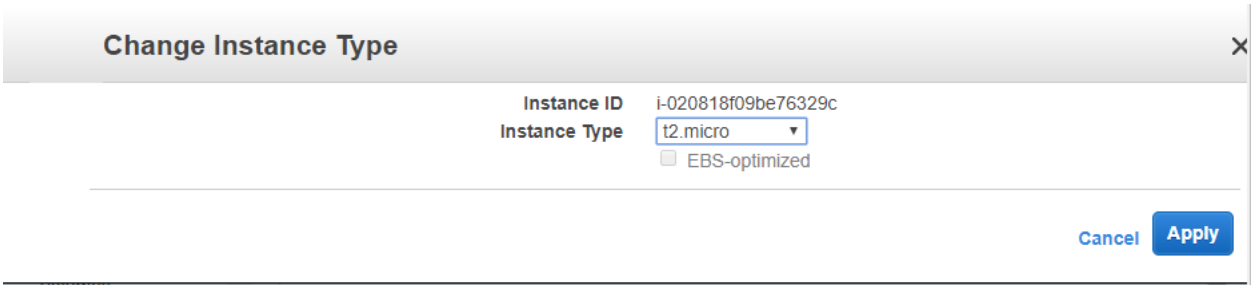
You can change the hardware configuration of your virtual machine instance at any time you want.

First, stop the instance for which you plan to update the hardware configuration.

Second, select the instance, navigate to Actions -> Instance Settings and click on Change Instance Type:



Select the instance type you want to change it to and click Apply. Done. You do not have to worry about your data getting lost, only the hardware will change:



To learn about different available instance types, their hardware specs, and cost, you can navigate to https://aws.amazon.com/ec2/pricing/on-demand/

Cloud9

From official AWS Cloud9 description: “AWS Cloud9 allows you to write, run, and debug your code with just a browser. With AWS Cloud9, you have immediate access to a rich code editor, integrated debugger, and built-in terminal with preconfigured AWS CLI. You can get started in minutes and no longer have to spend the time to install local applications or configure your development machine.”

The best part of Cloud9 is that you can collaborate with other people, editing the code at the same exact time, live! You do not have to worry about setting up the software on your machine, and you always have access to your projects that can be instantly tested and made publicly available on the Internet when you need it.

Cost Calculator

If you want to estimate the cost of all AWS services that you plan to use, navigate to https://calculator.s3.amazonaws.com/index.html

AWS has 3 different pricing models (pay-as-you-go, save when you reserve upfront, and pay less by using more). You can learn more about those here: https://aws.amazon.com/pricing/

Other AWS Cloud Services

For other services, it is actually easier to connect to and use them because there are extensive manuals and tutorials available.

Their user interface is very friendly. However, if you have any specific questions for a particular case, please email fordv@arcadia.edu and we will make it work.

What is Next?

Spread the word! Email me about any questions, especially if you are looking for a cloud solution that will help your teaching/research/capstones at Arcadia.

AWS provides a lot of free training material and videos that students and faculty can use to learn about cloud computing at their own pace. You will be able to find those materials in awseducate.com portal, under Professional Development.