Task Board API EC2 Setup Guide

(If you follow this guide correctly, you're team should be able to access the API from anywhere)

1. **Clone the Task API Repo**
    a. Run this git command exactly in whatever directory (folder) you want it in.
        i. *git clone https://gitlab.revaturelabs.com/revprotodosapi/todos-api.git*
        ii. **NOTE: linux systems hate spaces in folder names, try to avoid using any spaces in folder names that house pem files and the todos-api project folders.**

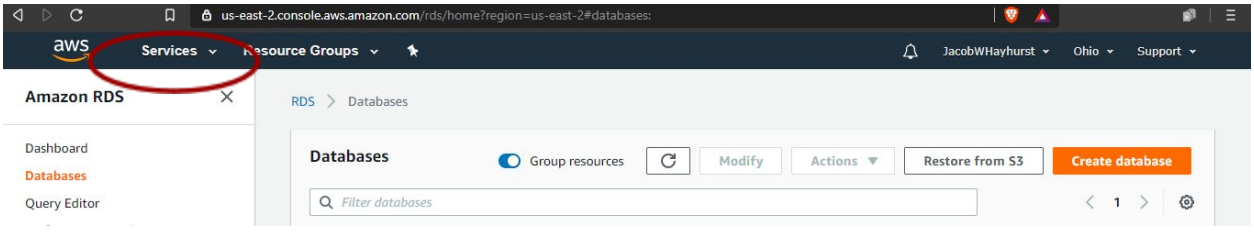
2. **Download Maven if you do not already have it on your desktop (not in STS)**
    a. Check to see if you have it already with mvn -v in the command line
    b. Download Maven Binary tar.gz archive from the link section at https://maven.apache.org/download.cgi
    c. Follow installation at https://maven.apache.org/install.html

3. **Build the todos-api project with Maven**
    a. Navigate to the folder in GitBash that holds the pom.xml aka todos-api folder and run the following command:
        i. *mvn package*

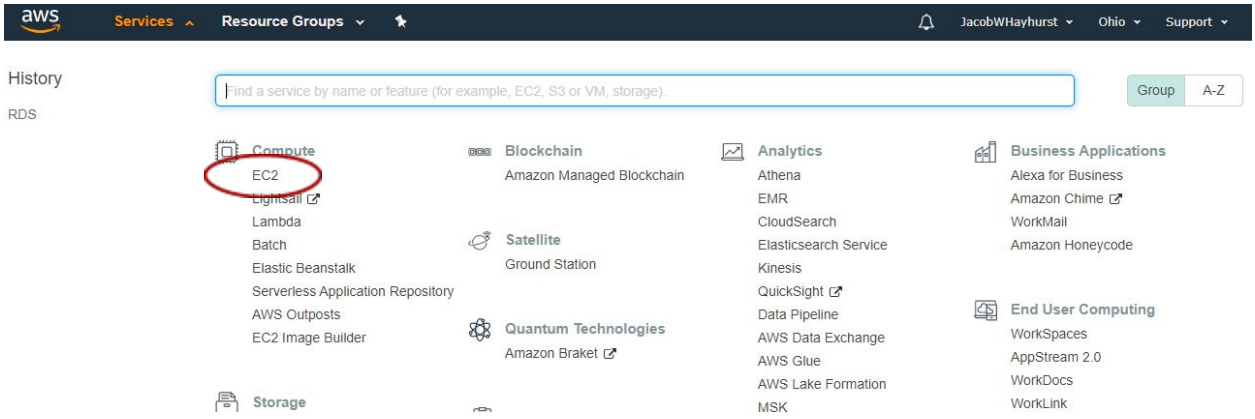
    b. Notice that in the todos-api/target folder there is a todos-api-1.0 executable JAR file

4. **Create the EC2 instance**

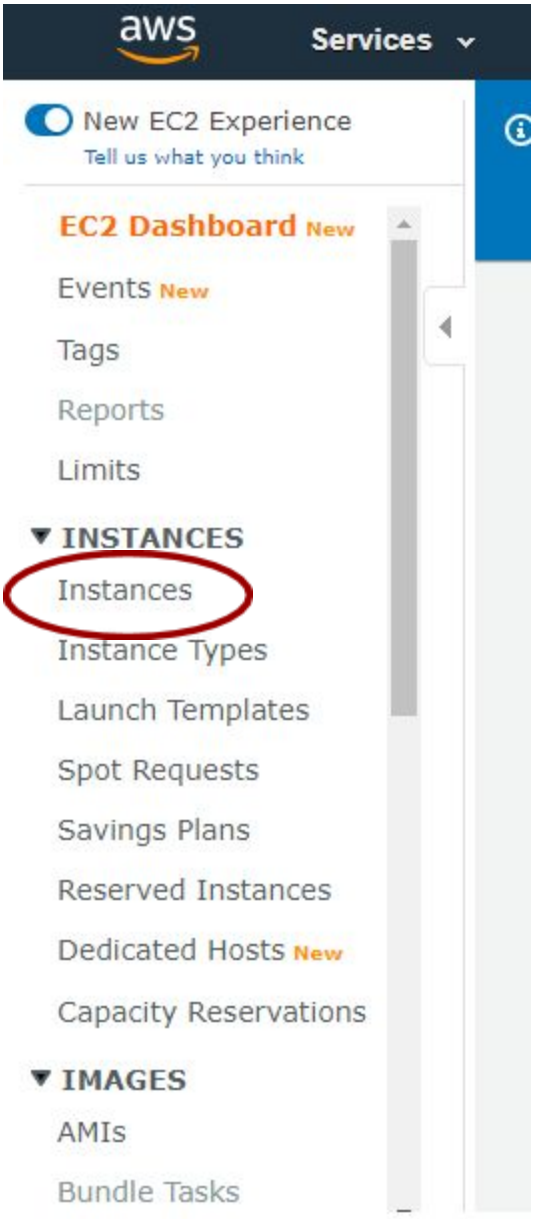
    1. Login in to AWS and navigate to the services tab as shown below:



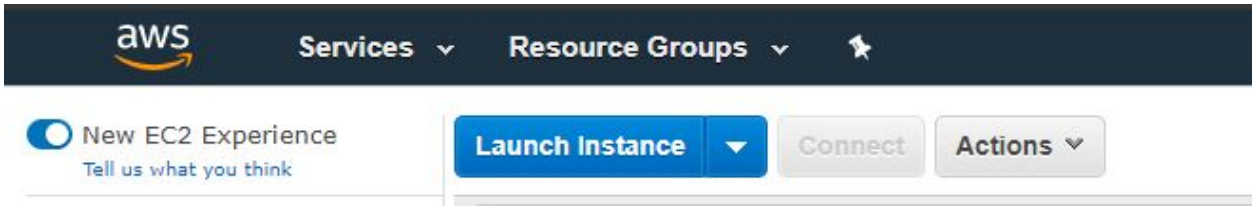
    2. Click EC2 on the drop down services list



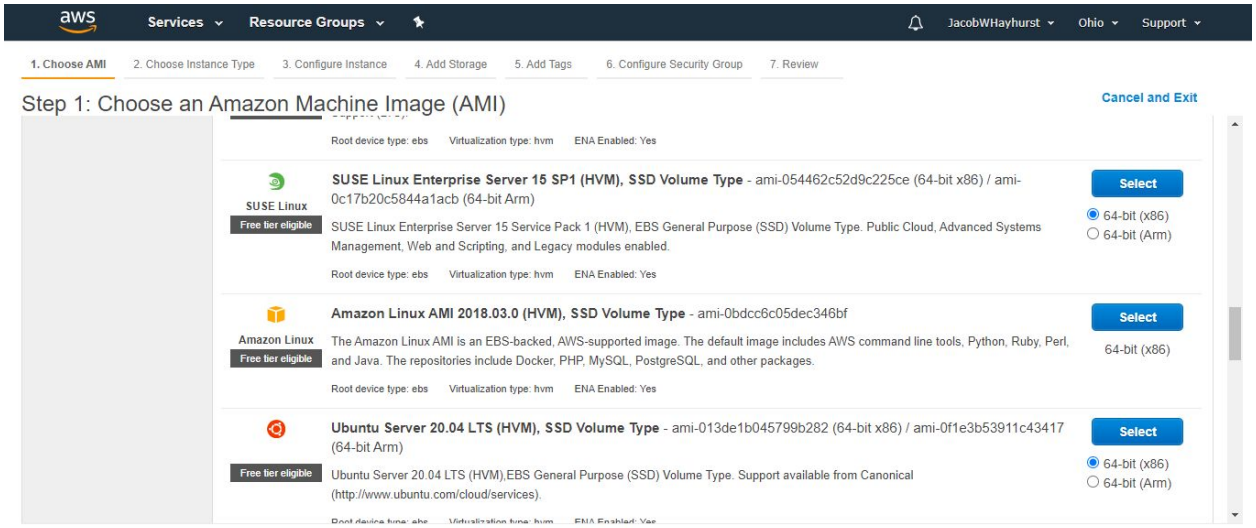
3. On the EC2 dashboard click “Instances” on the sidebar



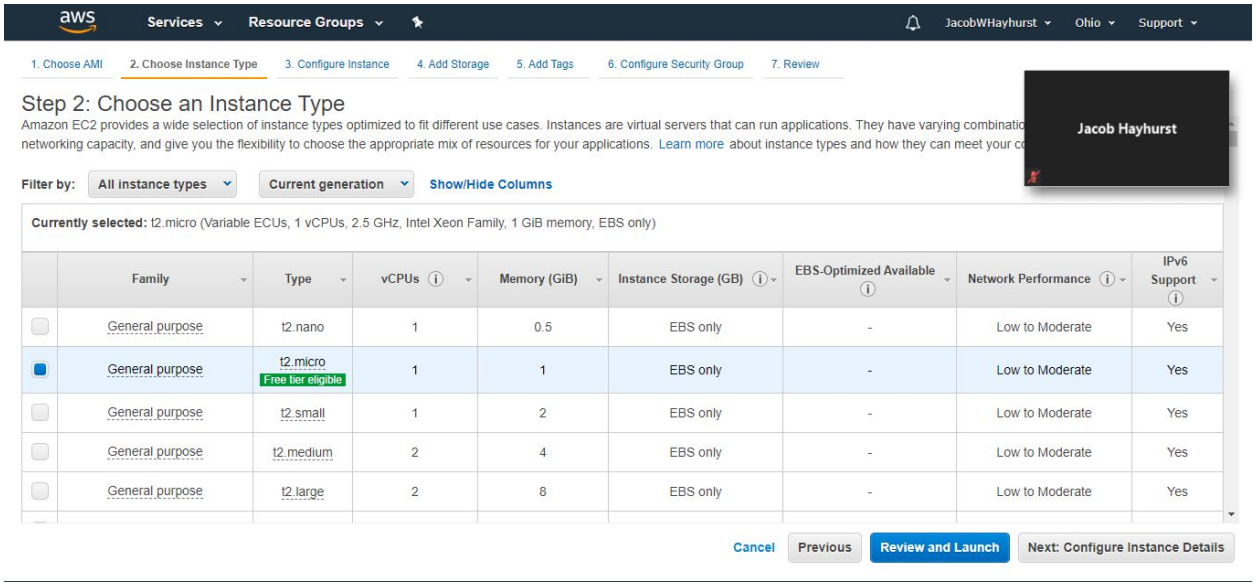
4. Then Click launch instance



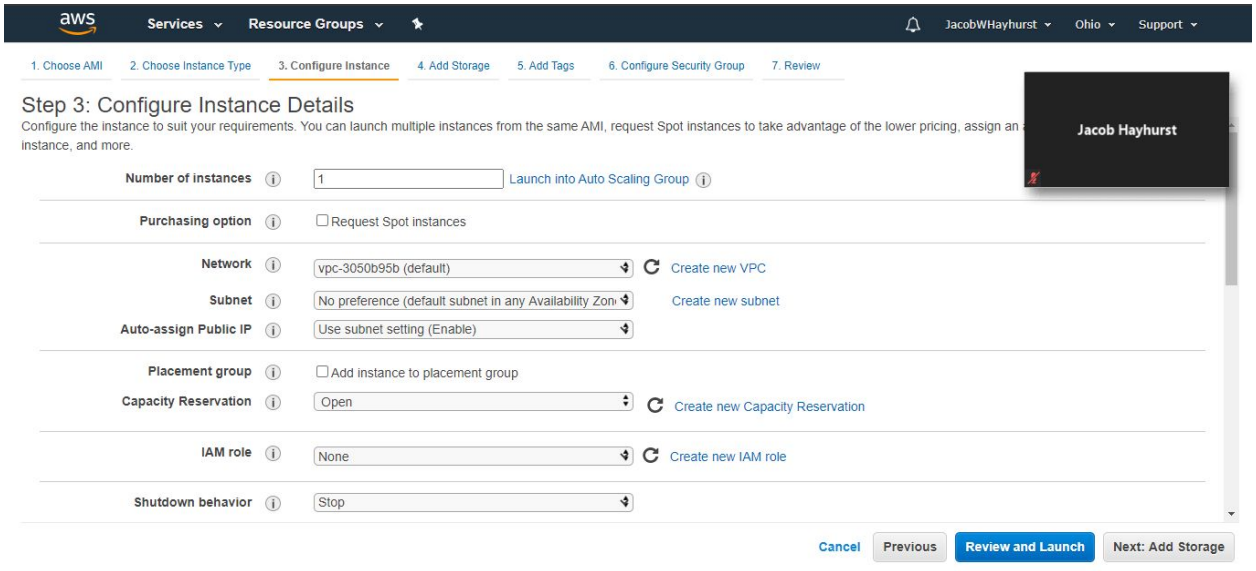
5. Select the **FREE TIER** Amazon Linux AMI as shown below



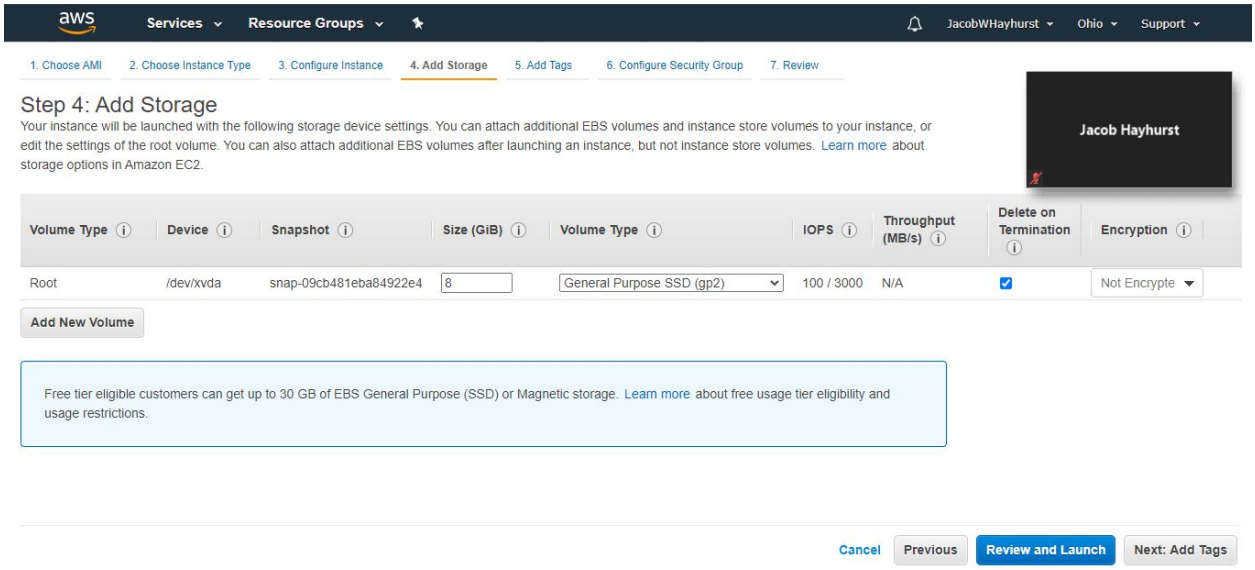
6. On choose an instance type page, leave it at free tier t2.micro and click next at the bottom right side of the page.



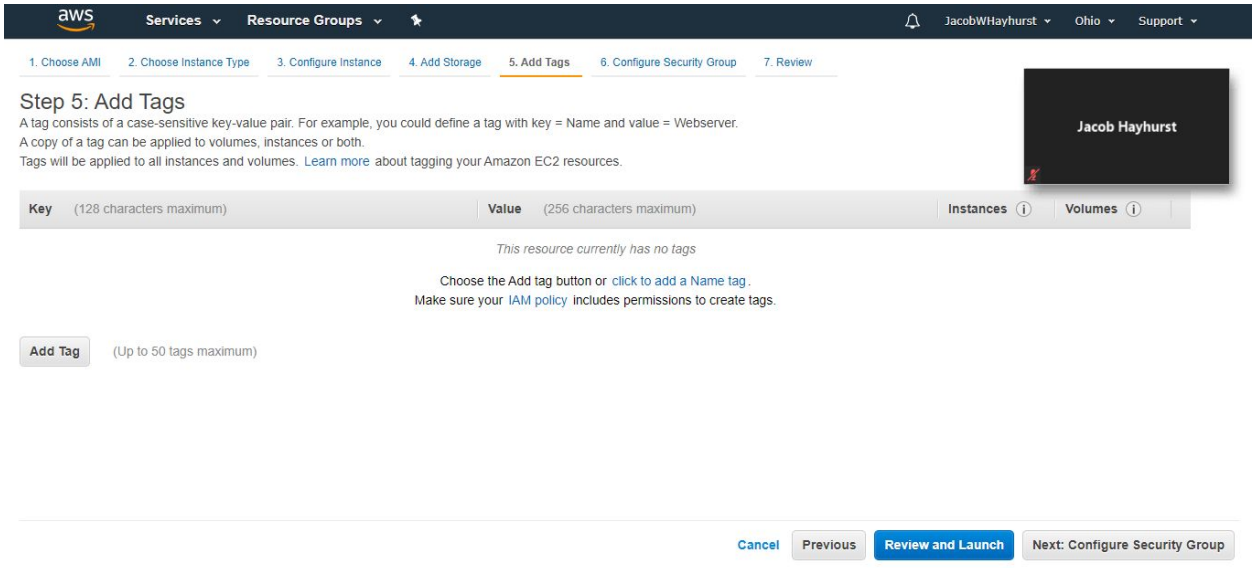
7. Keep the configuration instance details the same and clicked next again at the bottom right side of the page.



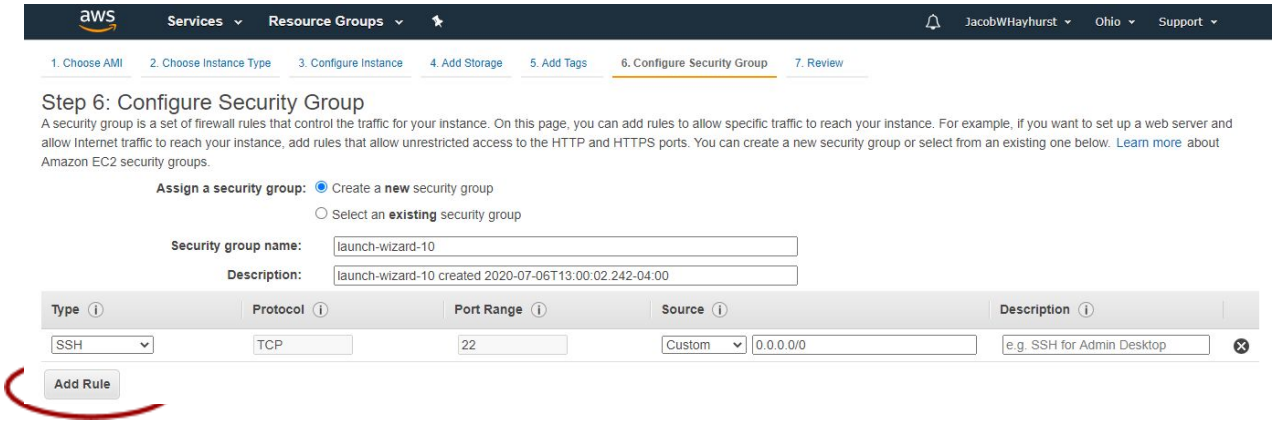
8. Leave storage settings the same, and just click next again



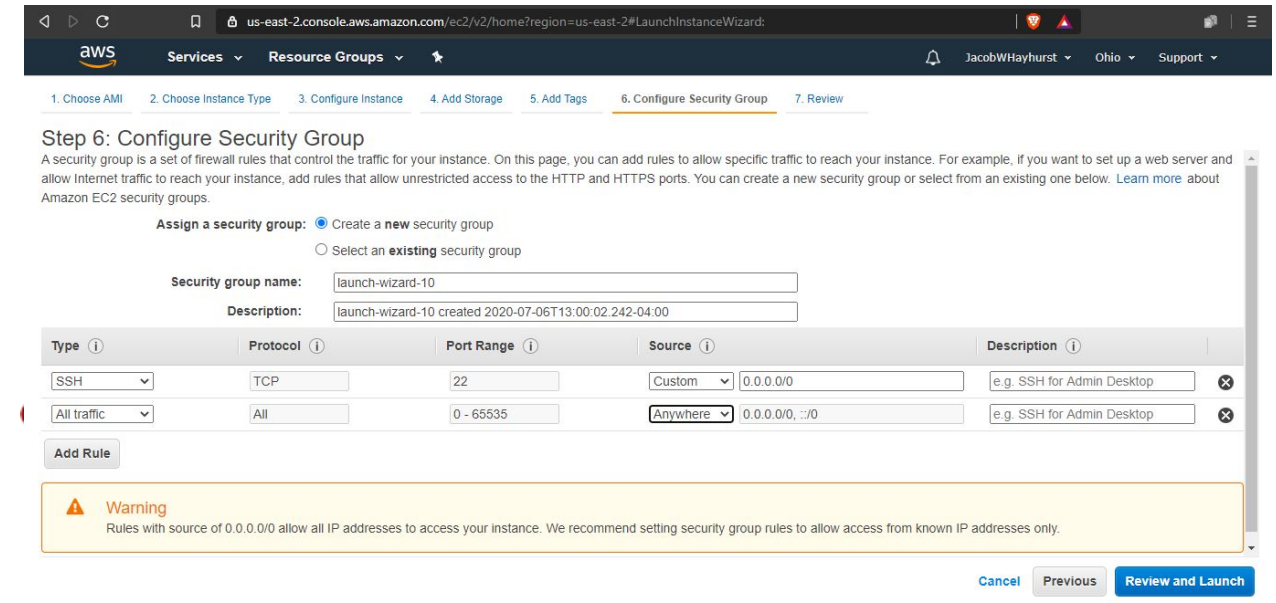
9. Don’t add any more tags and just click next again



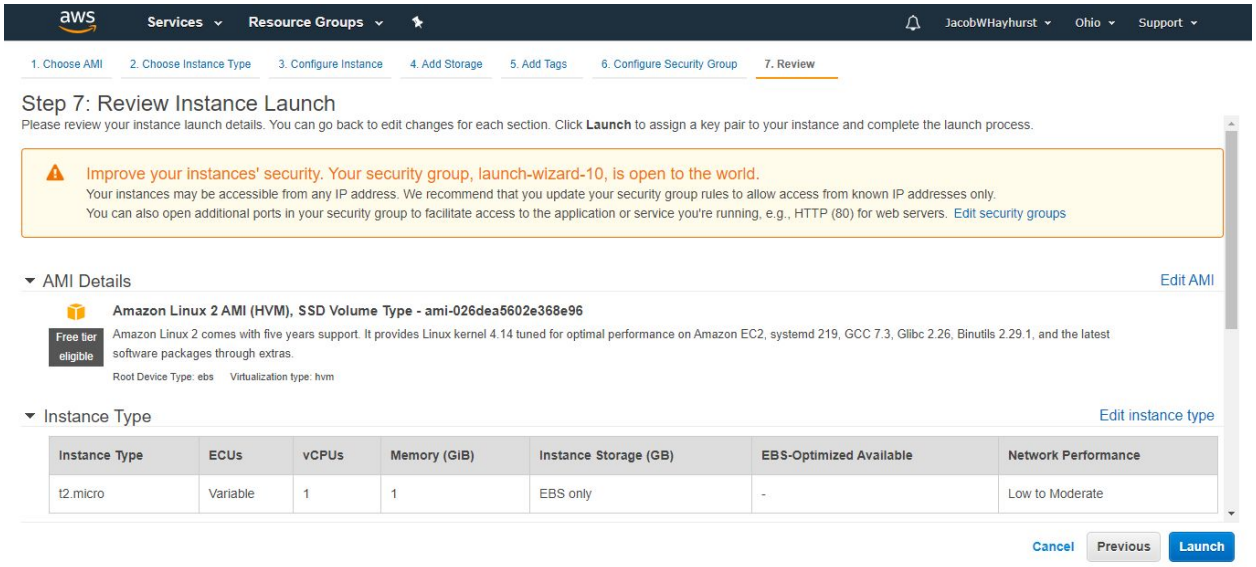
10. In the configure security Groups page, leave the current group untouched and click add rule at the bottom of the page.



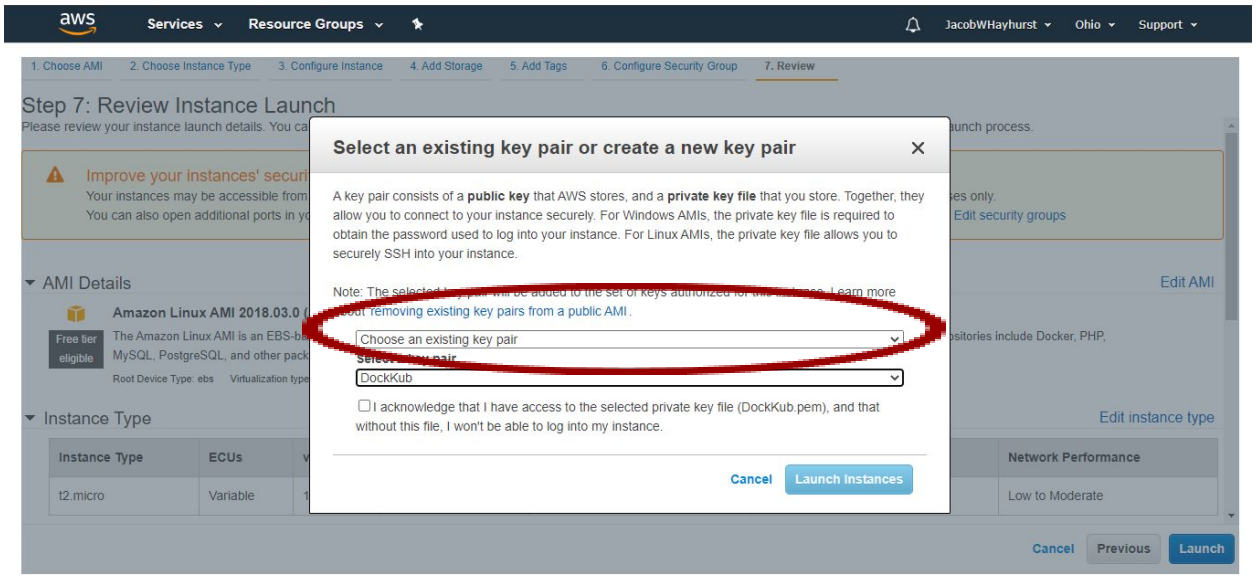
11. On the type dropdown, select the All Traffic option. On the source column in the table of the new all traffic rule, select anywhere from the dropdown. Then click review and launch. This will allow you to access the EC2 from anywhere on any port.



12. Click launch on the bottom right side of the page. **IMPORTANT, DO NOT RUSH THROUGH THE NEXT TWO STEPS AFTER HITTING LAUNCH**

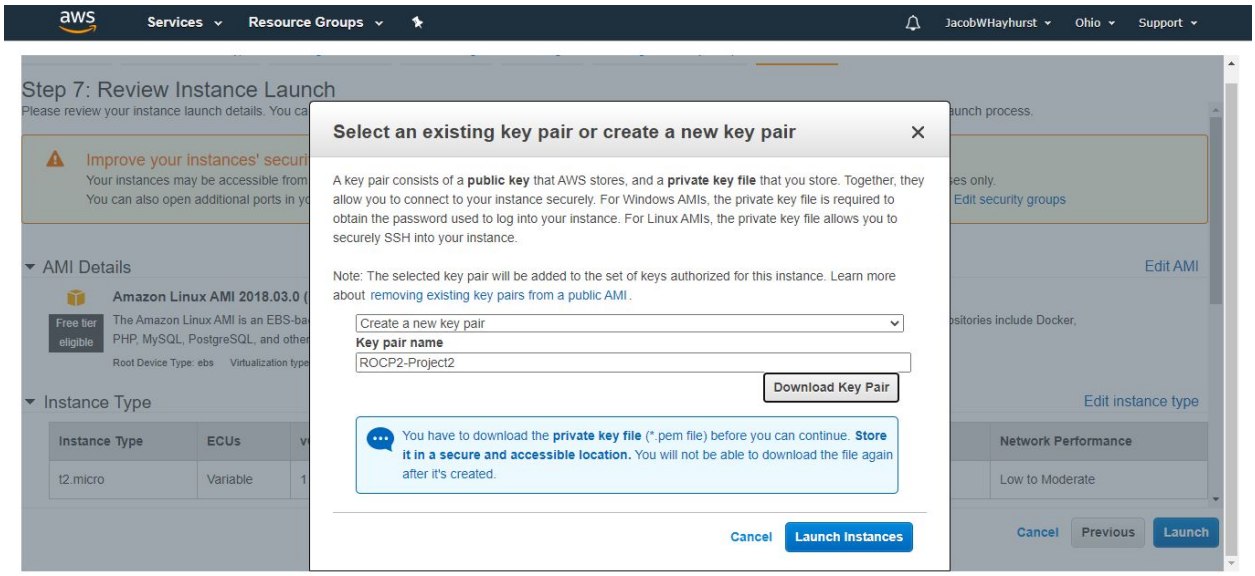


13. Make sure the first dropdown is on “create a new key pair” and name it appropriately.

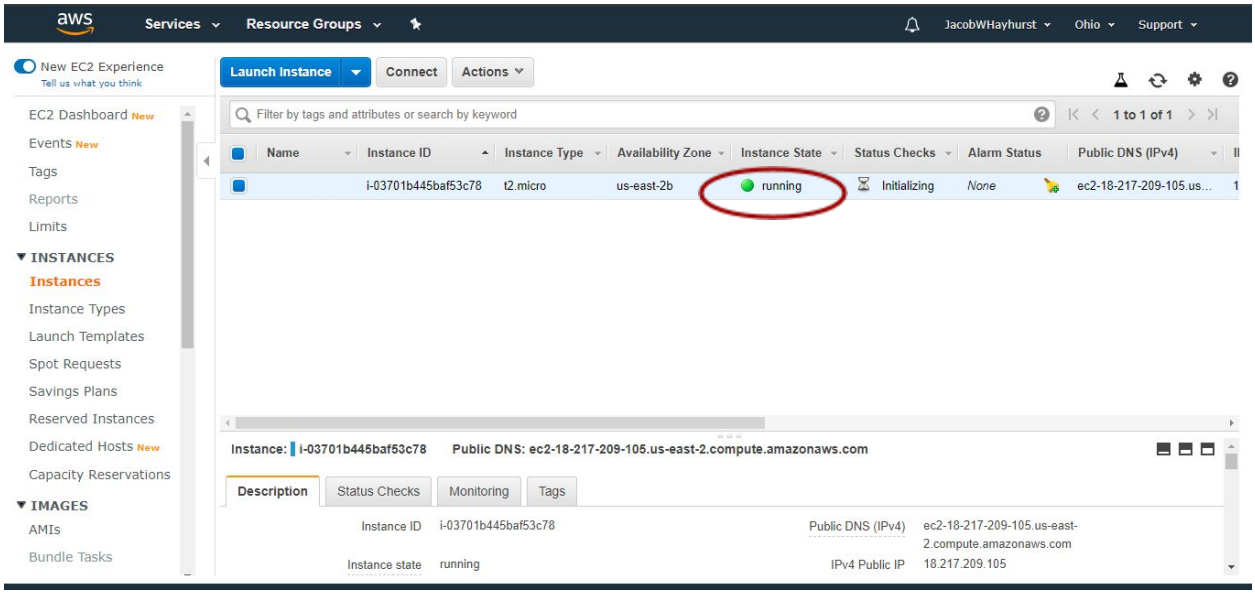


14. After naming it appropriately, **CLICK “DOWNLOAD KEY PAIR”** as this is the only time that you can download it **(NOTE: linux systems hate spaces in folder names, try to avoid using any spaces in folder names in the folder directories for pem files and the project folders)**. If you do not download it, you will have to create a new EC2. Save the key

in a place you can get to but not the desktop (this is due to permissions) and then click “launch Instances”.



15. Wait until the instance is running on the dashboard.



5. Access the EC2 by using ssh

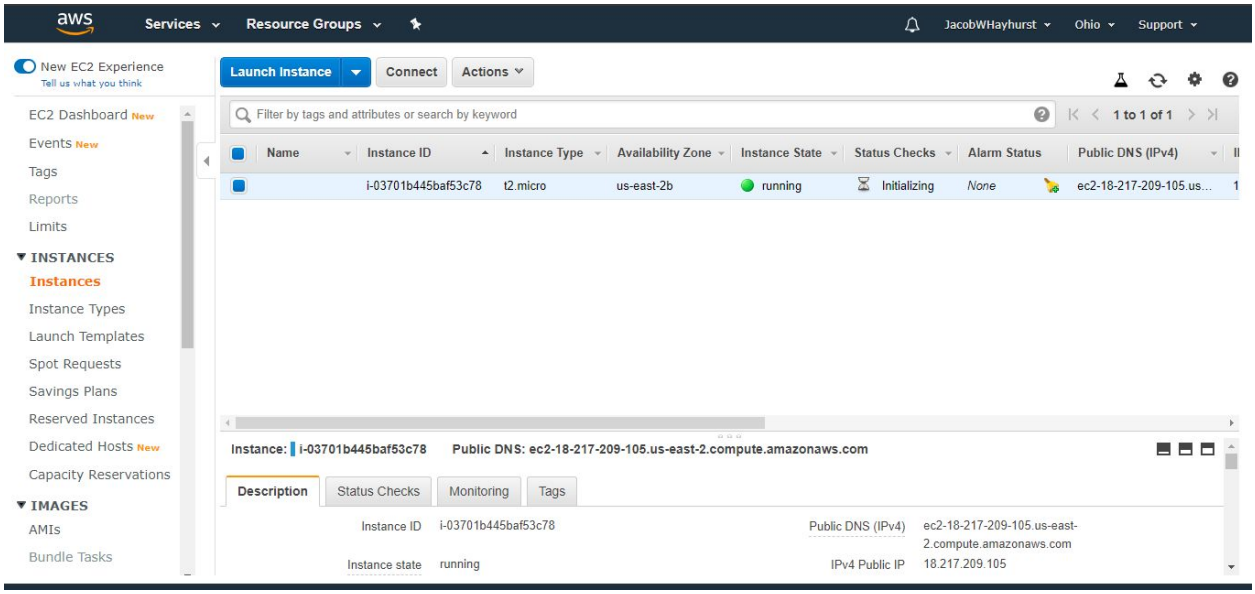
a. Open git BASH in the directory where you saved your EC2 key-pair, aka the pem file from step 14 above, if the directions below are not enough,

use this AWS resource to help as well: https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/AccessingInstancesLinux.html.

   i. **NOTE: linux systems hate spaces in folder names, try to avoid using any spaces in folder names in the folder directories for pem files and the project folders.**

b. The general structure of the command in git bash goes as follows:
   i. *ssh -i /path/my-key-pair.pem my-instance-user-name@my-instance-public-dns-name*

c. Notice that the above command is broken up into 4 parts
   i. *Ssh -i* (Secure SHell) being the command actually "tunnels" across a network to access the EC2 (the cloud computing service) remotely.
   ii. */path/my-key-pair.pem* is for the path to the Key-pair pem file downloaded locally on your computer **(there is a space after this path before the next part of the command)**.
   iii. *My-instance-user-name* is the EC2 user name, by default your username should be ec2-user.
   iv. *@my-instance-public-dns-name* is the actual end point to access your EC2 instance, you can find this on your AWS ec2 instance dashboard under the description tab at the bottom of the selected EC2 instance.



v. So the finished command should looks something like this:

```
Revature@DESKTOP-5K7MCMG MINGW64 ~/Documents/ROCP 2
$ ssh -i ROCP2-Project2.pem ec2-user@ec2-18-217-209-105.us-east-2.compute.amazon
aws.com
```

vi. Once you run this you should get prompted with the following:
   1. *Are you sure you want to continue connecting (yes/no)?*
   2. Which you should response with yes

vii. If everything is successful then you should see the following in the command line:

```
Revature@DESKTOP-5K7MCMG MINGW64 ~/Documents/ROCP 2
$ ssh -i ROCP2-Project2.pem ec2-user@ec2-18-217-209-105.us-east-2.compute.amazon
aws.com
Last login: Tue Jul  7 15:09:09 2020 from pool-71-178-235-202.washdc.fios.verizo
n.net

       __|  __|_  )
       _|  (     /   Amazon Linux AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-ami/2018.03-release-notes/
1 package(s) needed for security, out of 3 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-30-122 ~]$
```

6. **Copy the built todo-api into the EC2 using scp**
    a. Open a separate instance of git bash in the directory that the built todos-api-1.0 executable JAR file exists in (it should be in the target folder of the todos-api project folder as done in the bolded step 3).

    b. In git bash you will have use the scp command which has the structure of the following:
        i. scp -i /path/my-key-pair.pem /path/SampleFile.txt my-instance-user-name@my-instance-public-dns-name:~

    c. Notice that the following command is broken up into 5 parts:
        i. *scp -i* command which stands for Secure CoPy, which can copy files over a network to a remote system.
        ii. */path/my-key-pair.pem* is for the path to the Key-pair pem file downloaded locally on your computer **(there is a space after this path before the next part of the command)**.
        iii. */path/SampleFile.txt* is the file path to the file that you want to copy to the EC2, which in our case is the path to the todos-api-1.0 executable JAR file **NOTE that the file path has / (forward slash) not \ (backward slash) that will matter in the pathing of the jar file and the pem file. (there is a space after this path before the next part of the command)**.
        iv. *My-instance-user-name* is the EC2 user name, by default your username should be ec2-user.
        v. *@my-instance-public-dns-name* is the actual end point to access your EC2 instance, you can find this on your AWS ec2 instance dashboard under the description tab at the bottom of the selected EC2 instance**(there is a : after this dns name before the next part of the command)**.
        vi. :~ is the directory in the EC2 file system that you are copying too. We will go into this in much more depth in week 3, but the EC2 is essentially a separate computer that you are off loading processing power to. Since it is a Linux system that we are working with, we are going to copy to the ~ directory which is called the Root directory.

    d. The final command should look like the following:

```
Revature@DESKTOP-5K7MCMG MINGW64 ~/Documents/ROCP 2
$ scp -i ROCP2-Project2.pem "C:\Users\Revature\Documents\ROCP2-Project2\todos-api\target\todos
-api-1.0.jar" ec2-user@ec2-18-217-209-105.us-east-2.compute.amazonaws.com:~
C:\Users\Revature\Documents\ROCP2-Project2\todos-api\target 100%   47MB   3.6MB/s   00:12
```

e. If you look on the other gitbash terminal that is ssh into the EC2 you should be able to see the executable jar file in the ~ directory. (use ls command in that directory)

f. **NOTE: the file scribbled out in red is to be ignored(I messed up with using \ instead / in my scp command at first)**

7. **Install Java on the ec2 using yum**
   a. Use the following command in the ssh ec2 terminal to install Java:
      i. *sudo yum install -y java-1.8.0-openjdk.x86_64*
   b. Set the path variables for java via these commands
      i. *sudo /usr/sbin/alternatives --set java /usr/lib/jvm/jre-1.8.0-openjdk.x86_64/bin/java*
      ii. *sudo /usr/sbin/alternatives --set javac /usr/lib/jvm/jre-1.8.0-openjdk.x86_64/bin/javac*
   c. You may have to remove java 1.7 using the command
      i. *sudo yum remove java-1.7*

8. **Run the application in a detach screen, so that it runs in the background and doesn't time out.**
   a. In the ssh EC2 gitbash terminal, use the following command:
      i. *screen*
   b. Then run the command to run a java jar file:
      i. *java -jar todos-api-1.0.jar*
      ii. *Then hit ctrl+A on the keyboard*
      iii. *Then hit ntrl+D on the keyboard to detach the screen from the session and run in the background.*

9. **If no errors arise, you are now good to go**