

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)

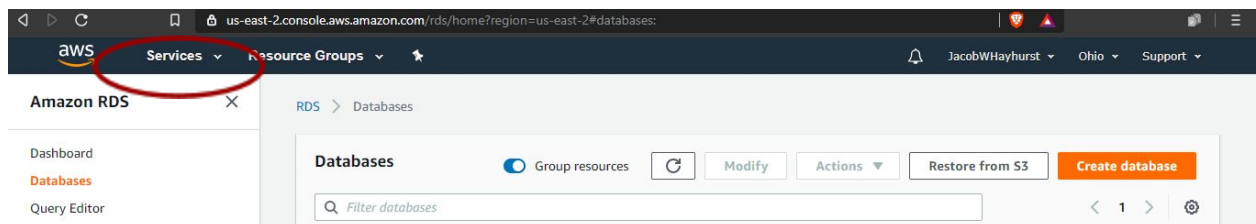
- a. Check to see if you have it already with `mvn -v` in the command line
- b. Download Maven Binary tar.gz archive from the link section at <https://maven.apache.org/download.cgi>
- c. Follow installation at <https://maven.apache.org/install.html>

3. Build the todos-api project with Maven

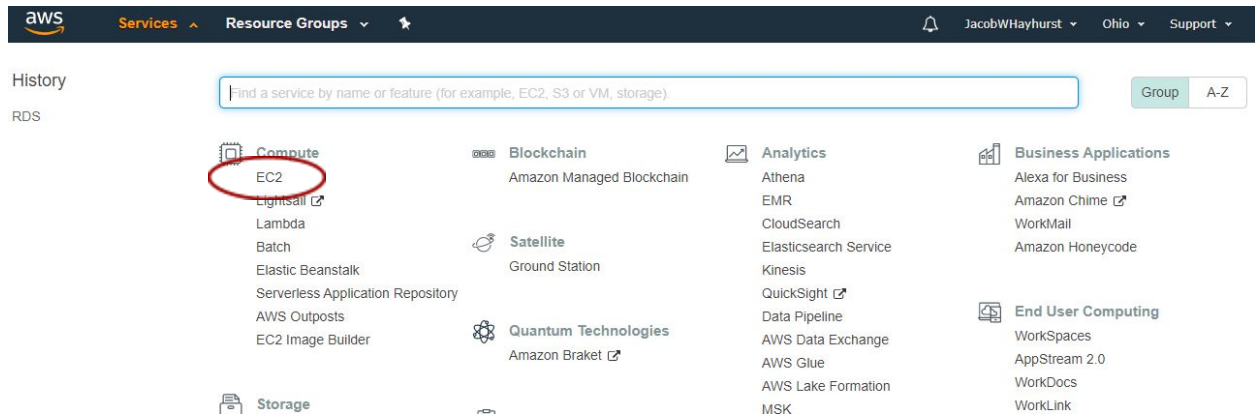
- a. Navigate to the folder in GitBash that holds the pom.xml aka todos-api folder and run the following command:
 - i. `mvn package`
- b. Notice that in the todos-api/target folder there is a todos-api-1.0 executable JAR file

4. Create the EC2 instance

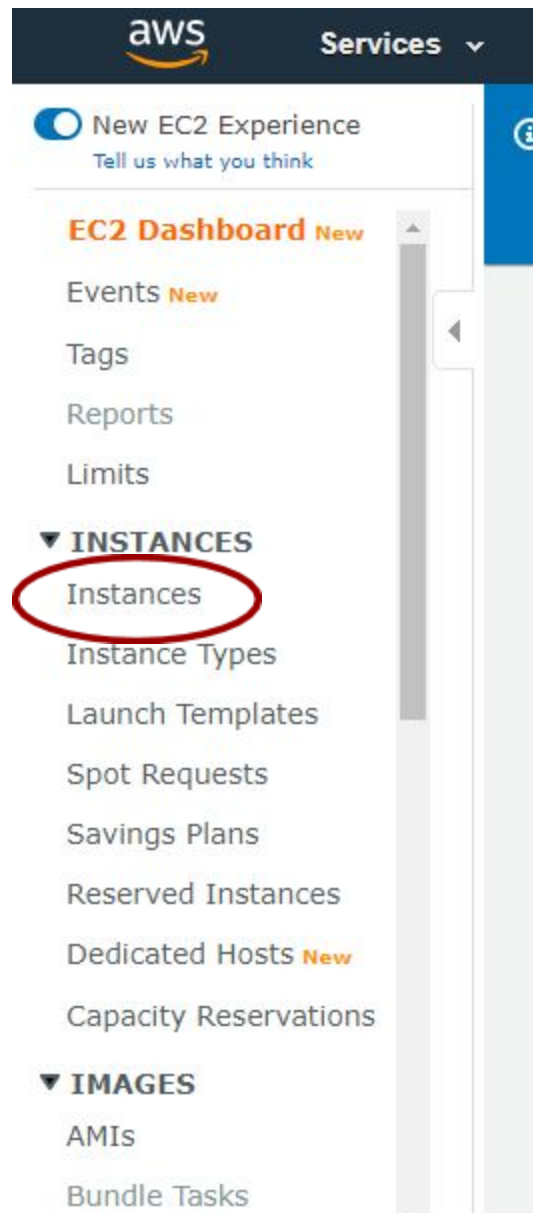
1. Login in to AWS and navigate to the services tab as shown below:



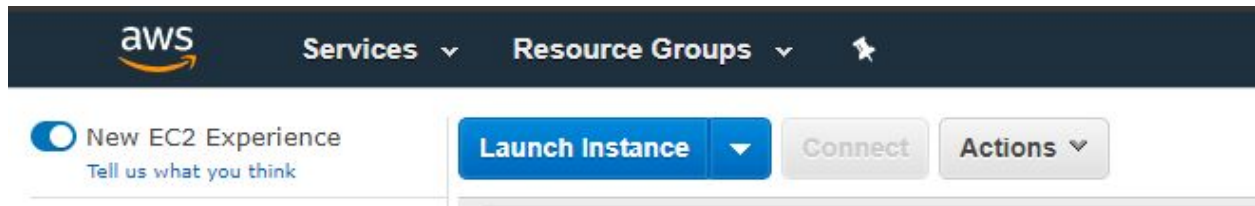
2. Click EC2 on the drop down services list



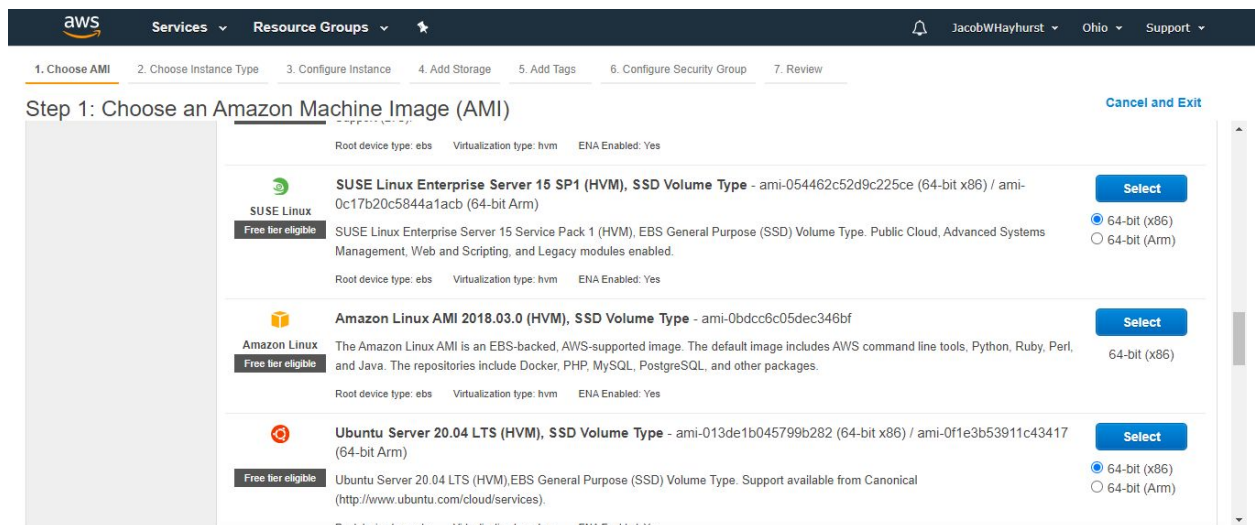
3. On the EC2 dashboard click “Instances” on the sidebar



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.

aws Services Resource Groups

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

Step 2: Choose an Instance Type

Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instances are virtual servers that can run applications. They have varying combinations of CPU, memory, storage, and networking capacity, and give you the flexibility to choose the appropriate mix of resources for your applications. [Learn more](#) about instance types and how they can meet your needs.

Filter by: All Instance types Current generation Show/Hide Columns

Currently selected: t2.micro (Variable ECUs, 1 vCPUs, 2.5 GHz, Intel Xeon Family, 1 GiB memory, EBS only)

	Family	Type	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance	IPv6 Support
<input type="checkbox"/>	General purpose	t2.nano	1	0.5	EBS only	-	Low to Moderate	Yes
<input checked="" type="checkbox"/>	General purpose	t2.micro Free tier eligible	1	1	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	General purpose	t2.small	1	2	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	General purpose	t2.medium	2	4	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	General purpose	t2.large	2	8	EBS only	-	Low to Moderate	Yes

Cancel Previous **Review and Launch** Next: Configure Instance Details

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

aws Services Resource Groups

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

Step 3: Configure Instance Details

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage of the lower pricing, assign an IAM role, and more.

Number of instances: 1 Launch into Auto Scaling Group

Purchasing option: ☐ Request Spot instances

Network: vpc-3050b95b (default) Create new VPC

Subnet: No preference (default subnet in any Availability Zone) Create new subnet

Auto-assign Public IP: Use subnet setting (Enable)

Placement group: ☐ Add instance to placement group

Capacity Reservation: Open Create new Capacity Reservation

IAM role: None Create new IAM role

Shutdown behavior: Stop

Cancel Previous **Review and Launch** Next: Add Storage

- Leave storage settings the same, and just click next again

aws Services Resource Groups

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

Step 4: Add Storage

Your instance will be launched with the following storage device settings. You can attach additional EBS volumes and instance store volumes to your instance, or edit the settings of the root volume. You can also attach additional EBS volumes after launching an instance, but not instance store volumes. [Learn more](#) about storage options in Amazon EC2.

Volume Type	Device	Snapshot	Size (GiB)	Volume Type	IOPS	Throughput (MB/s)	Delete on Termination	Encryption
Root	/dev/xvda	snap-09cb481eba84922e4	8	General Purpose SSD (gp2)	100 / 3000	N/A	<input checked="" type="checkbox"/>	Not Encrypted

[Add New Volume](#)

Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage. [Learn more](#) about free usage tier eligibility and usage restrictions.

[Cancel](#) [Previous](#) [Review and Launch](#) [Next: Add Tags](#)

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

aws Services Resource Groups

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

Step 5: Add Tags

A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = Webserver. A copy of a tag can be applied to volumes, instances or both. Tags will be applied to all instances and volumes. [Learn more](#) about tagging your Amazon EC2 resources.

Key (128 characters maximum)	Value (256 characters maximum)	Instances	Volumes
This resource currently has no tags			

Choose the [Add tag](#) button or [click to add a Name tag](#).
Make sure your [IAM policy](#) includes permissions to create tags.

[Add Tag](#) (Up to 50 tags maximum)

[Cancel](#) [Previous](#) [Review and Launch](#) [Next: Configure Security Group](#)

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

Step 6: Configure Security Group

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. [Learn more](#) about Amazon EC2 security groups.

Assign a security group: ☒ Create a new security group
☐ Select an existing security group

Security group name:

Description:

Type	Protocol	Port Range	Source	Description
SSH	TCP	22	Custom 0.0.0.0/0	e.g. SSH for Admin Desktop

Add Rule

- 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.

Step 6: Configure Security Group

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. [Learn more](#) about Amazon EC2 security groups.

Assign a security group: ☒ Create a new security group
☐ Select an existing security group

Security group name:

Description:

Type	Protocol	Port Range	Source	Description
SSH	TCP	22	Custom 0.0.0.0/0	e.g. SSH for Admin Desktop
All traffic	All	0 - 65535	Anywhere 0.0.0.0, ::/0	e.g. SSH for Admin Desktop

Add Rule

Warning

Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

[Cancel](#) [Previous](#) [Review and Launch](#)

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

Step 7: Review Instance Launch
Please review your instance launch details. You can go back to edit changes for each section. Click **Launch** to assign a key pair to your instance and complete the launch process.

Improve your instances' security. Your security group, `launch-wizard-10`, is open to the world.
Your instances may be accessible from any IP address. We recommend that you update your security group rules to allow access from known IP addresses only. You can also open additional ports in your security group to facilitate access to the application or service you're running, e.g., HTTP (80) for web servers. [Edit security groups](#)

AMI Details [Edit AMI](#)

Amazon Linux 2 AMI (HVM), SSD Volume Type - ami-026dea5602e368e96
Free tier eligible
Amazon Linux 2 comes with five years support. It provides Linux kernel 4.14 tuned for optimal performance on Amazon EC2, systemd 219, GCC 7.3, Glibc 2.26, Binutils 2.29.1, and the latest software packages through extras.
Root Device Type: ebs Virtualization type: hvm

Instance Type [Edit instance type](#)

Instance Type	ECUs	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance
t2.micro	Variable	1	1	EBS only	-	Low to Moderate

[Cancel](#) [Previous](#) [Launch](#)

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

Step 7: Review Instance Launch
Please review your instance launch details. You can go back to edit changes for each section. Click **Launch** to assign a key pair to your instance and complete the launch process.

Improve your instances' security. Your security group, `launch-wizard-10`, is open to the world.
Your instances may be accessible from any IP address. We recommend that you update your security group rules to allow access from known IP addresses only. You can also open additional ports in your security group to facilitate access to the application or service you're running, e.g., HTTP (80) for web servers. [Edit security groups](#)

AMI Details [Edit AMI](#)

Amazon Linux 2 AMI (HVM), SSD Volume Type - ami-026dea5602e368e96
Free tier eligible
Amazon Linux 2 comes with five years support. It provides Linux kernel 4.14 tuned for optimal performance on Amazon EC2, systemd 219, GCC 7.3, Glibc 2.26, Binutils 2.29.1, and the latest software packages through extras.
Root Device Type: ebs Virtualization type: hvm

Instance Type [Edit instance type](#)

Instance Type	ECUs	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance
t2.micro	Variable	1	1	EBS only	-	Low to Moderate

[Cancel](#) [Previous](#) [Launch](#)

Select an existing key pair or create a new key pair ✕

A key pair consists of a **public key** that AWS stores, and a **private key file** that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key file allows you to securely SSH into your instance.

Note: The selected key pair will be added to the set of keys authorized for this instance. [Learn more about removing existing key pairs from a public AMI.](#)

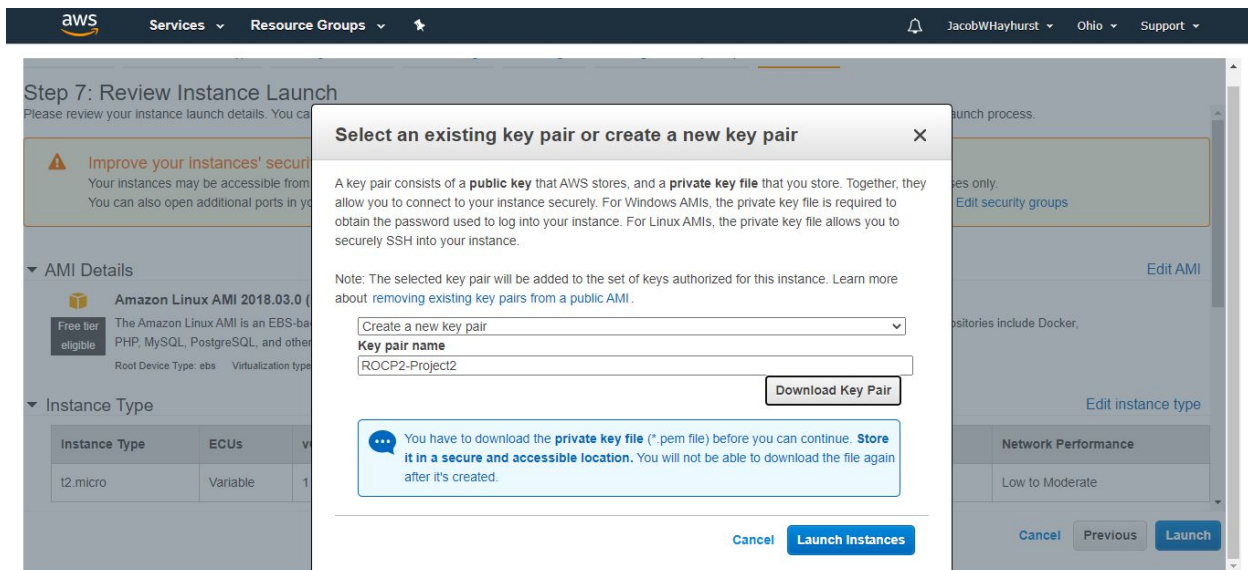
Choose an existing key pair ▼
 Select a new key pair
 DockKub ▼

☐ I acknowledge that I have access to the selected private key file (DockKub.pem), and that without this file, I won't be able to log into my instance.

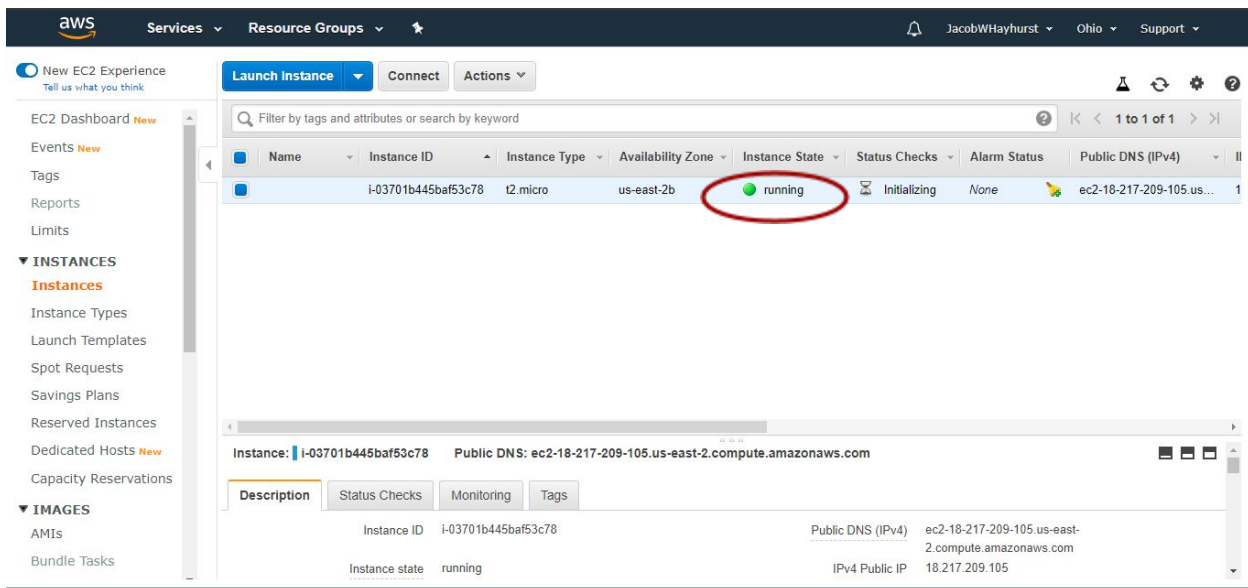
[Cancel](#) [Launch Instances](#)

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.



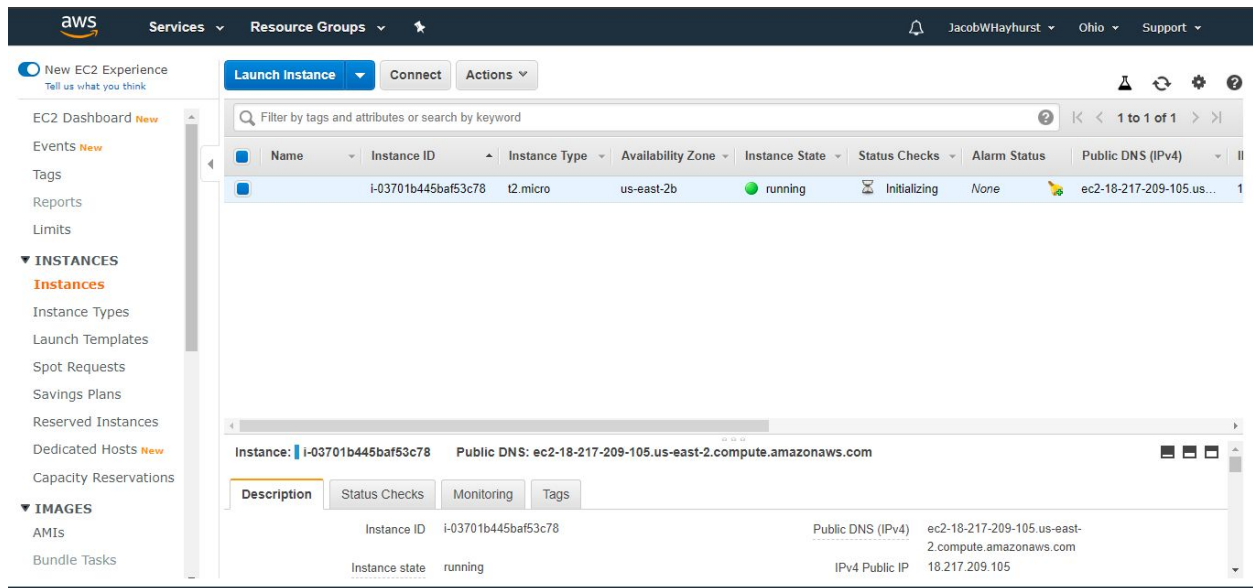
5. Access the EC2 by using ssh

- 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 look 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.amazonaws.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.amazonaws.com
Last login: Tue Jul 7 15:09:09 2020 from pool-71-178-235-202.washdc.fios.verizon.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)

```
[ec2-user@ip-172-31-30-122 ~]$ dir
C:\Users\Revature\Documents\ROCP2-Project2\todos-api\target\todos-api-1.0.jar
todos-api-1.0.jar
```

- 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 `ctrl+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