Deep Learning - Recitation 1

What is Amazon Web Services (AWS)

Cloud Based Computational Resource

What does AWS offer?

Many many things but here are the two main things to care about for DL...

EC2 - Compute Resources

Train the models

S3 - Data Storage

Store training data, models, etc

EC2 - What kinds of machines are available?

Different types and different subtypes (you can mix and match what you want)... Here are the ones you may care about

General Purpose:

T2 - Webservices

M3/M4 - Databases, Fileservers, etc

Compute Optimized:

C2 - Multiplayer Gaming Servers, scientific computing

C3/C4 - Ad serving machines, MMO servers, etc

GPU Optimized:

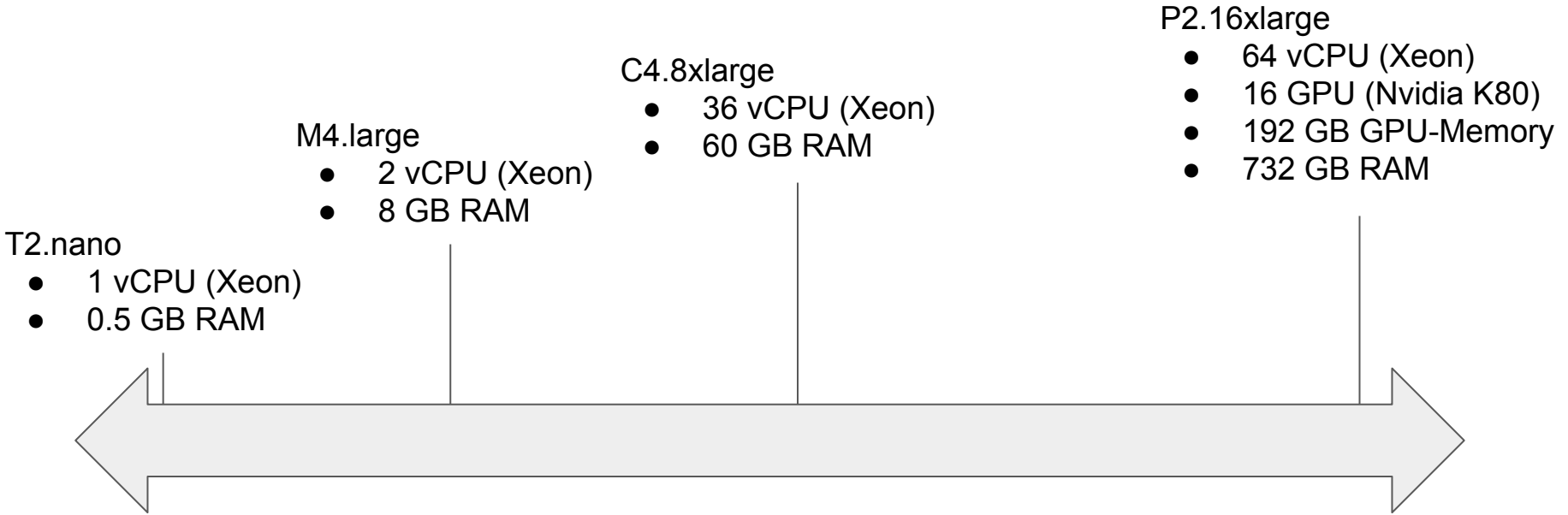
P3/P2 - Machine Learning

G3 - Fluid dynamics, graphics rendering, etc

Machine sizes - nano, micro, medium, large, xlarge, 2xlarge, …, 16xlarge

EC2 - What kinds of machines are available?

Different types and different subtypes (you can mix and match what you want)...



EC2 - So what do we put on these machines?

Amazon Machine Instances (AMIs)

- Virtual images of existing machines
  - You can create an image of your machine
    - Transfer it to a different machine
    - Save it as a backup
- Use cases
  - Software packages that are incredibly difficult to install
  - Need to create multiple different machines with the exact same data for parameters servers
  - Load balancing - create a new machine with the same AMI to be used in a different region depending on load

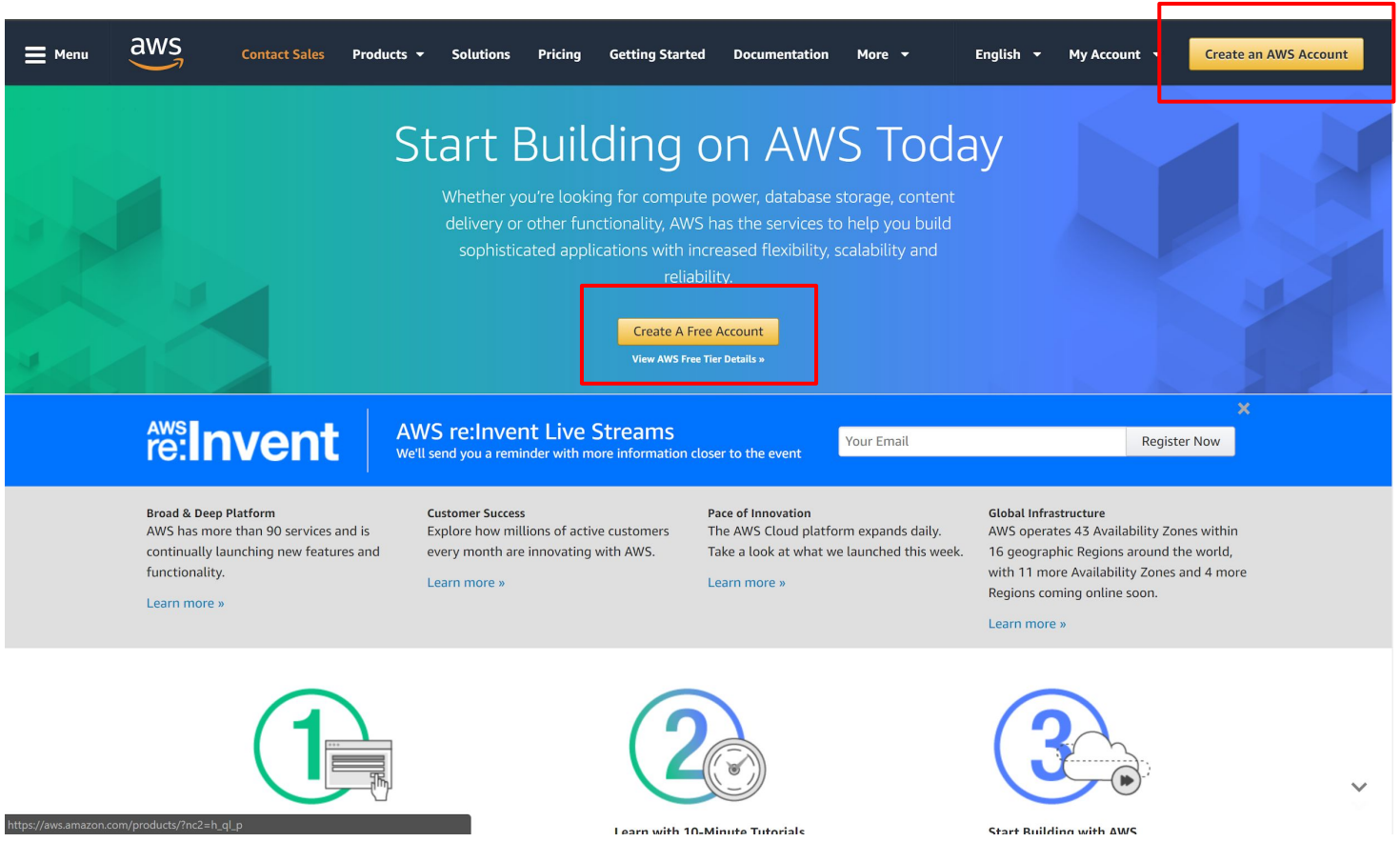
Now you know what AWS is and what you can do with it

Create An Account:

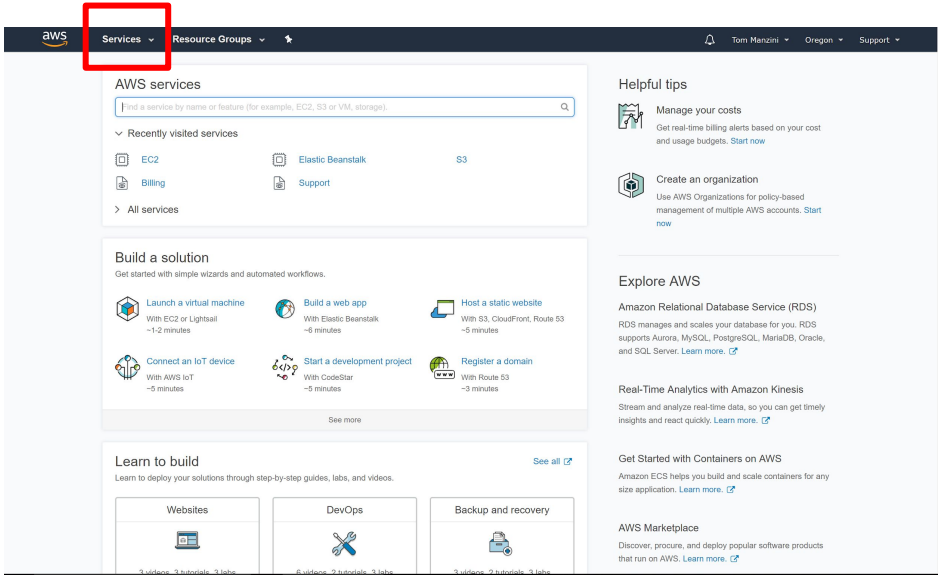
https://aws.amazon.com

Click “Sign Up” in the top right and follow the instructions

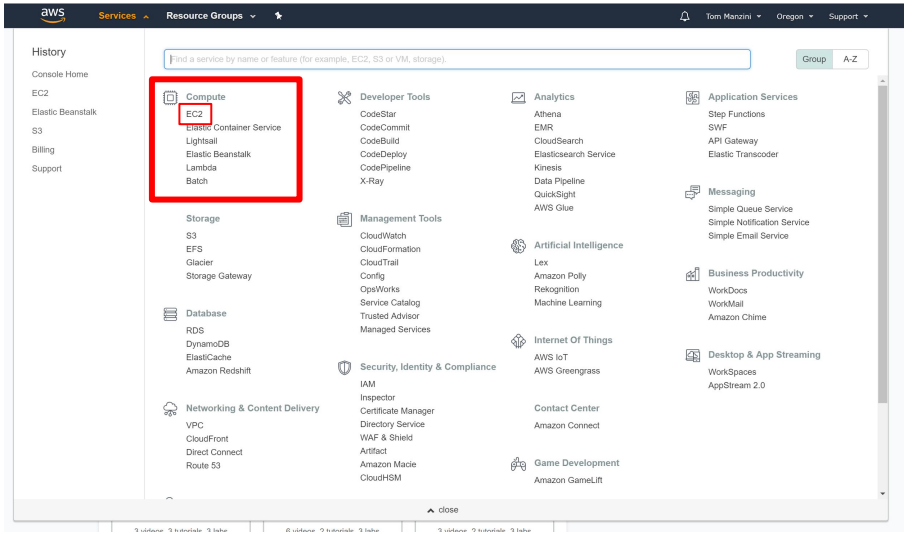
(If you already have an account you can skip this step)


Menu
aws
Contact Sales
Products
Solutions
Pricing
Getting Started
Documentation
More
English
My Account
Create an AWS Account
Start Building on AWS Today
Whether you're looking for compute power, database storage, content delivery or other functionality, AWS has the services to help you build sophisticated applications with increased flexibility, scalability and reliability.
Create A Free Account
View AWS Free Tier Details »
AWS re:Invent
AWS re:Invent Live Streams
We'll send you a reminder with more information closer to the event
Your Email
Register Now
Broad & Deep Platform
AWS has more than 90 services and is continually launching new features and functionality.
Learn more »
Customer Success
Explore how millions of active customers every month are innovating with AWS.
Learn more »
Pace of Innovation
The AWS Cloud platform expands daily. Take a look at what we launched this week.
Learn more »
Global Infrastructure
AWS operates 43 Availability Zones within 16 geographic Regions around the world, with 11 more Availability Zones and 4 more Regions coming online soon.
Learn more »
1
2
3
https://aws.amazon.com/products/?nc2=h_ql_p

Let’s set up a basic machine



Click on “Services” in the top left

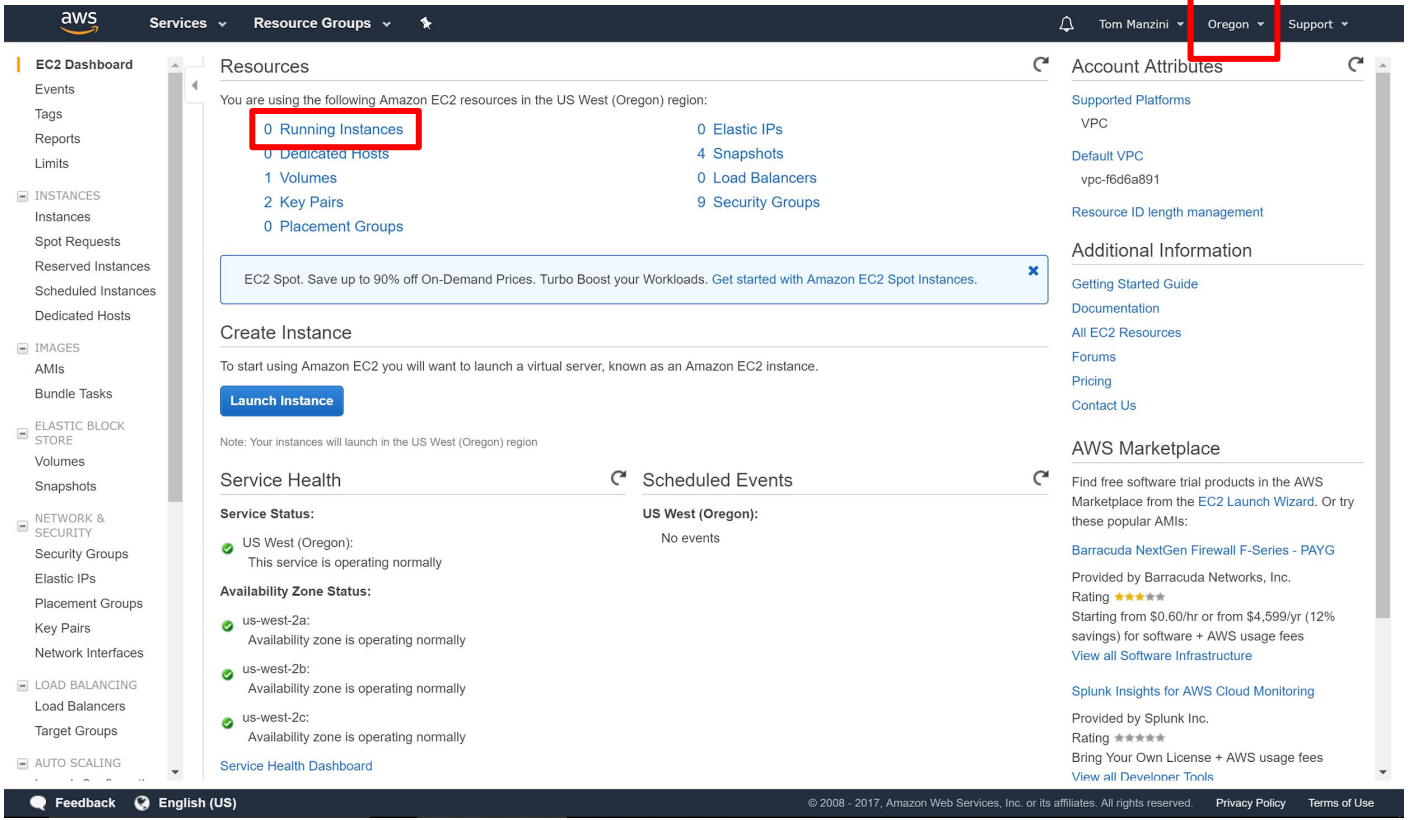


Then, under “Compute”, select “EC2”

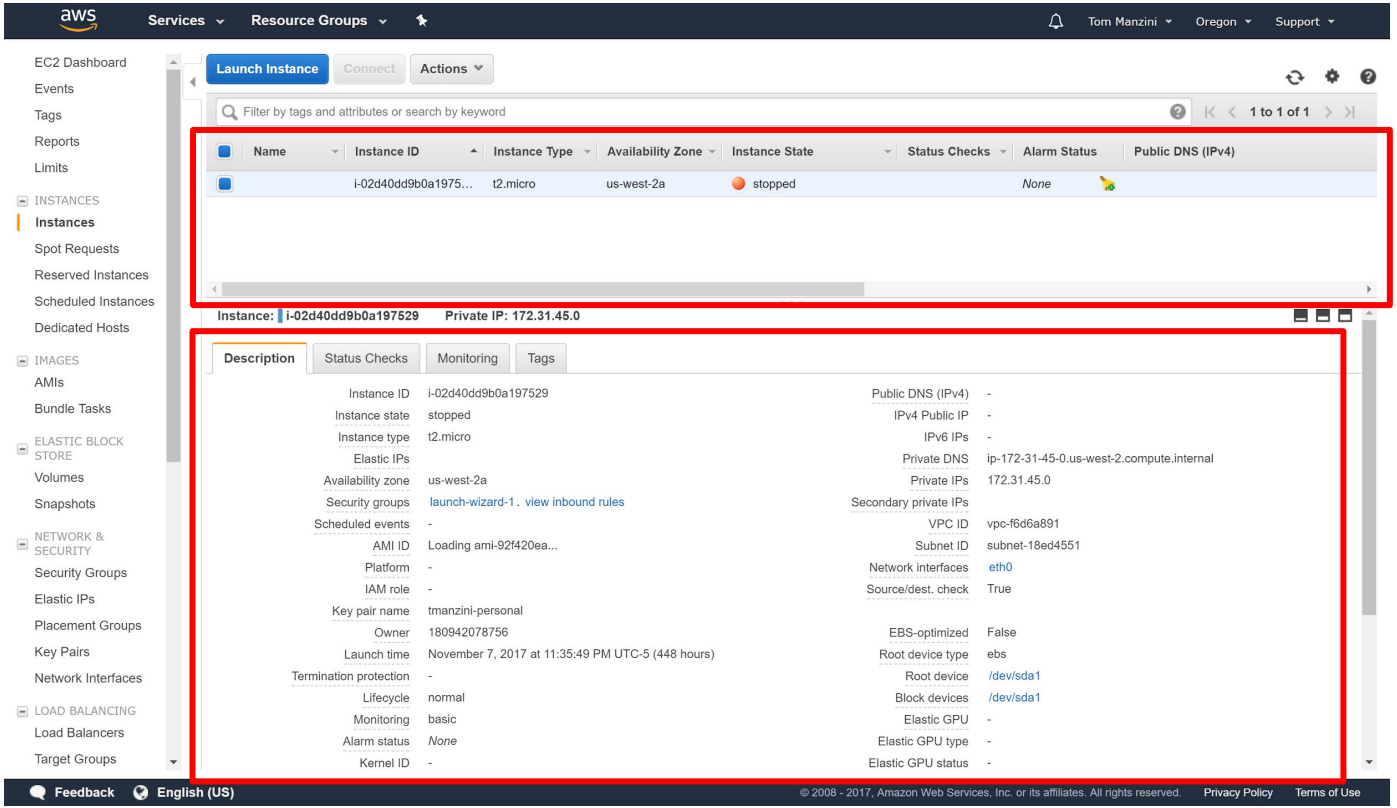
Let’s set up a basic machine

Make sure you are in the “US West (Oregon)” region

Then Click on “Running Instances”



Let’s set up a basic machine



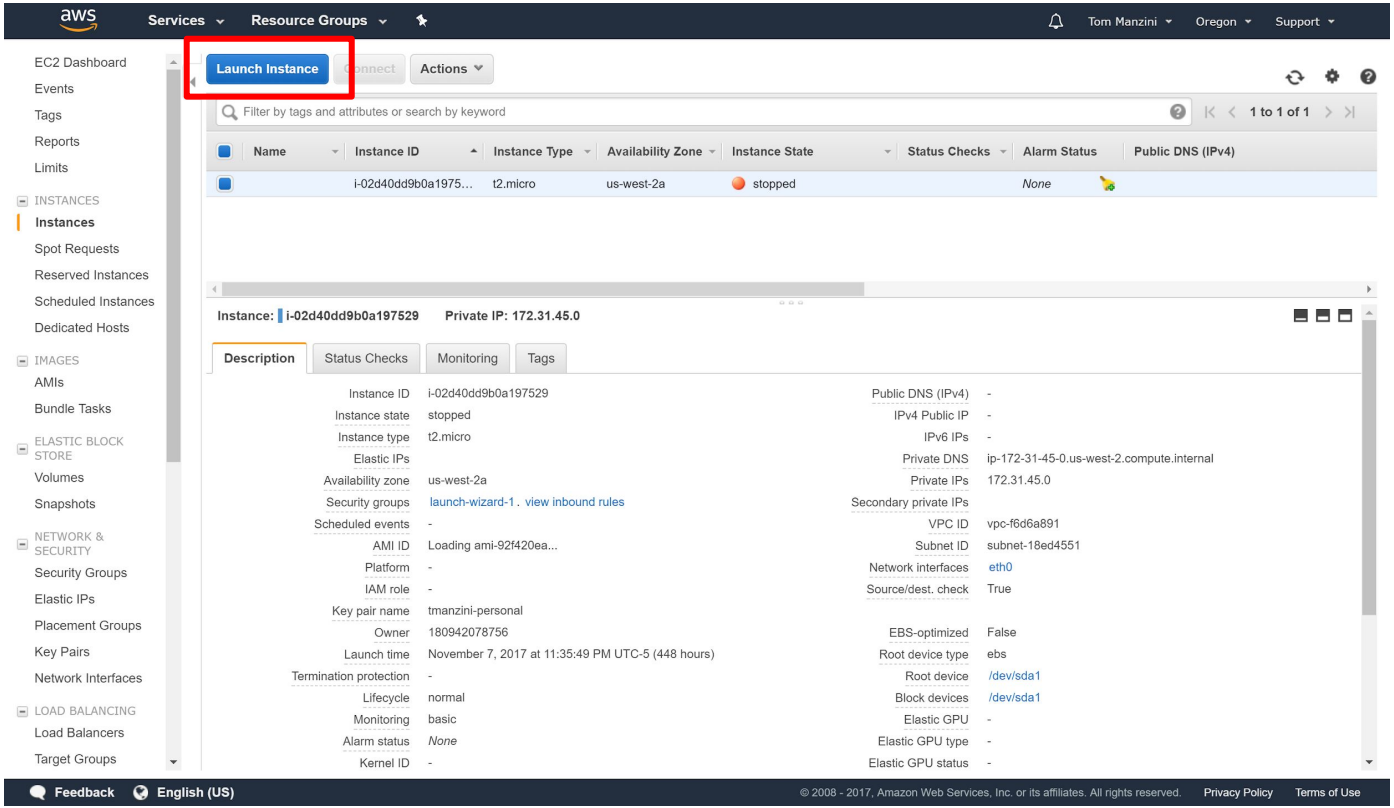
Here you see your current instances

(I have 1 that is stopped, you shouldn’t have any)

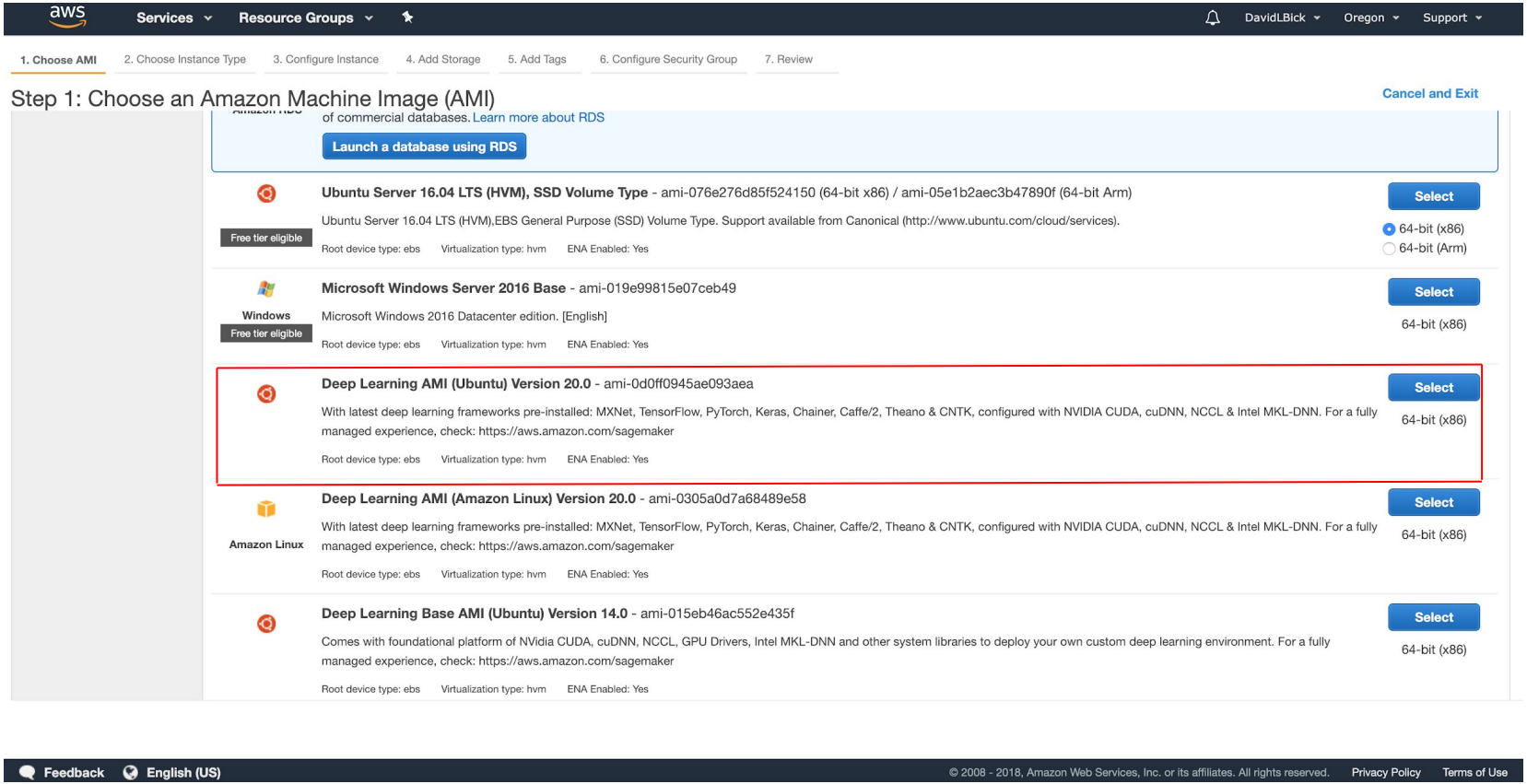
Here are the details of that instance

Let’s set up a basic machine

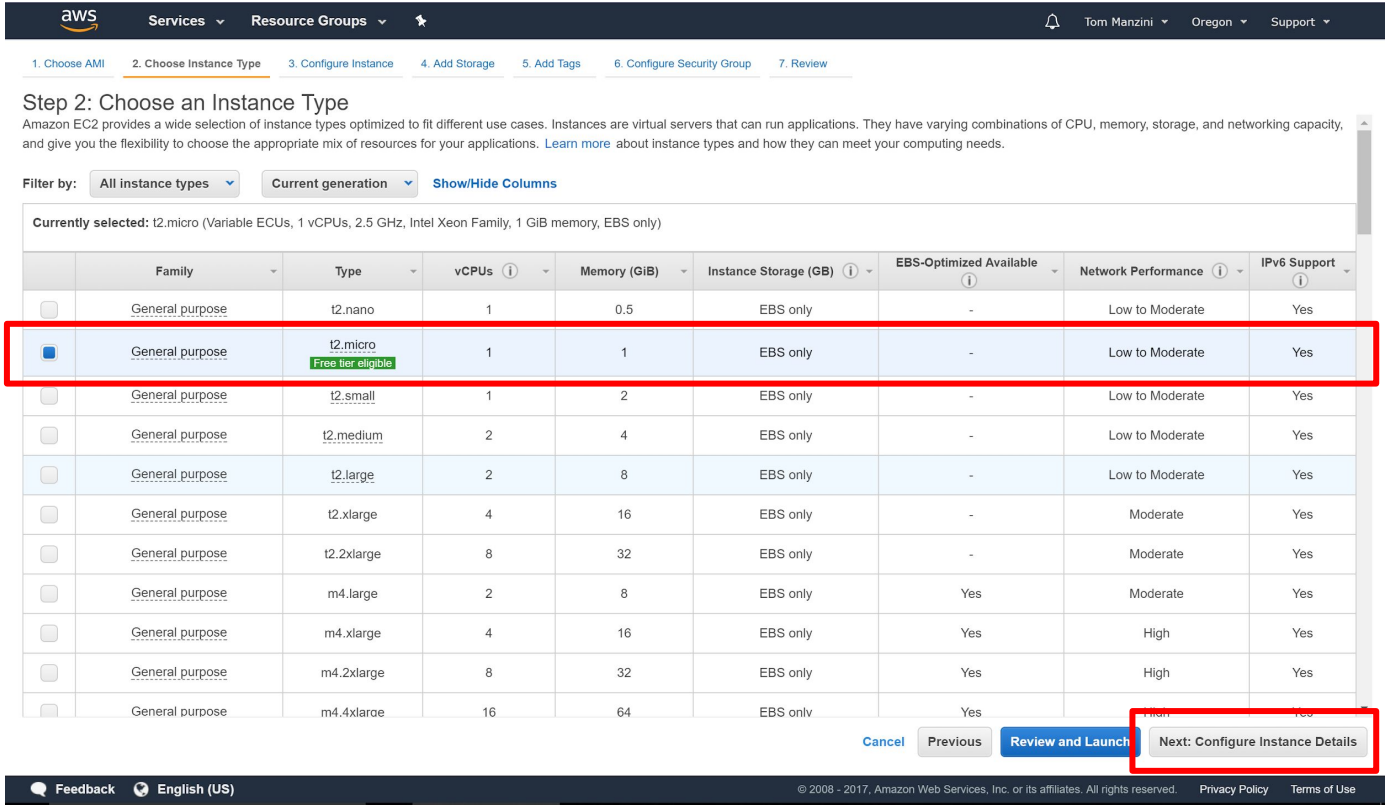
Click “Launch Instance”



Let’s set up a basic machine



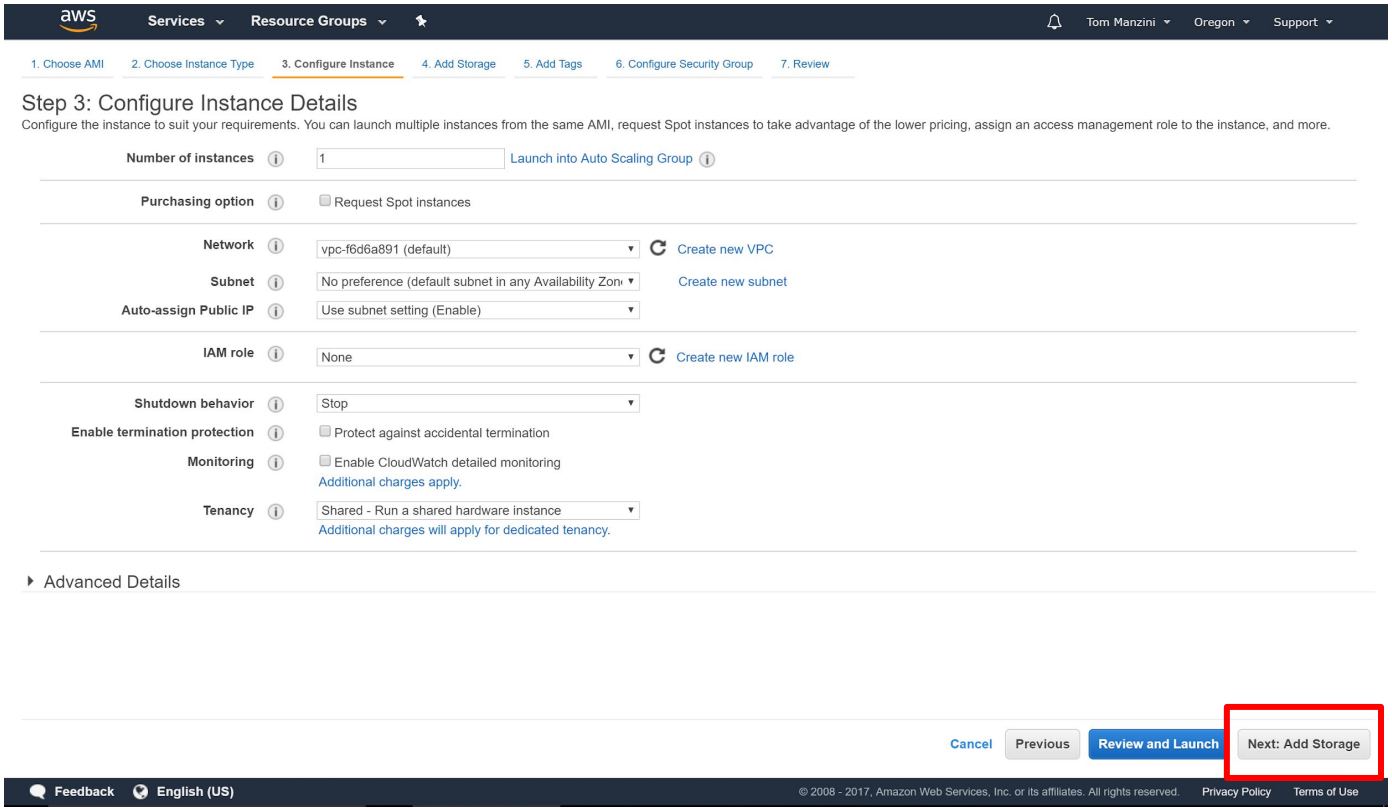
Let’s set up a basic machine



Select the t2-micro because it is “free tier eligible”

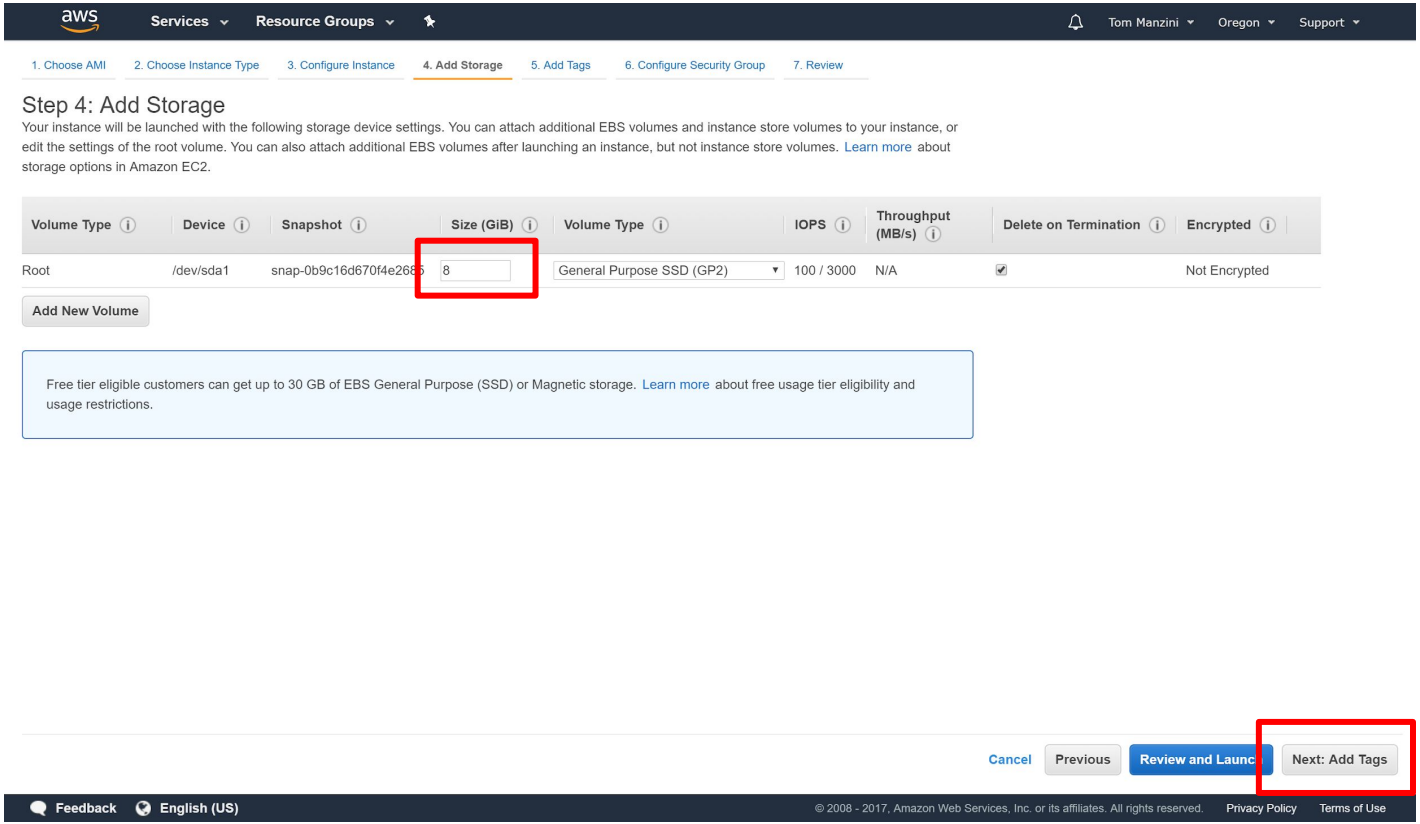
Select Next

Let’s set up a basic machine



Just select next

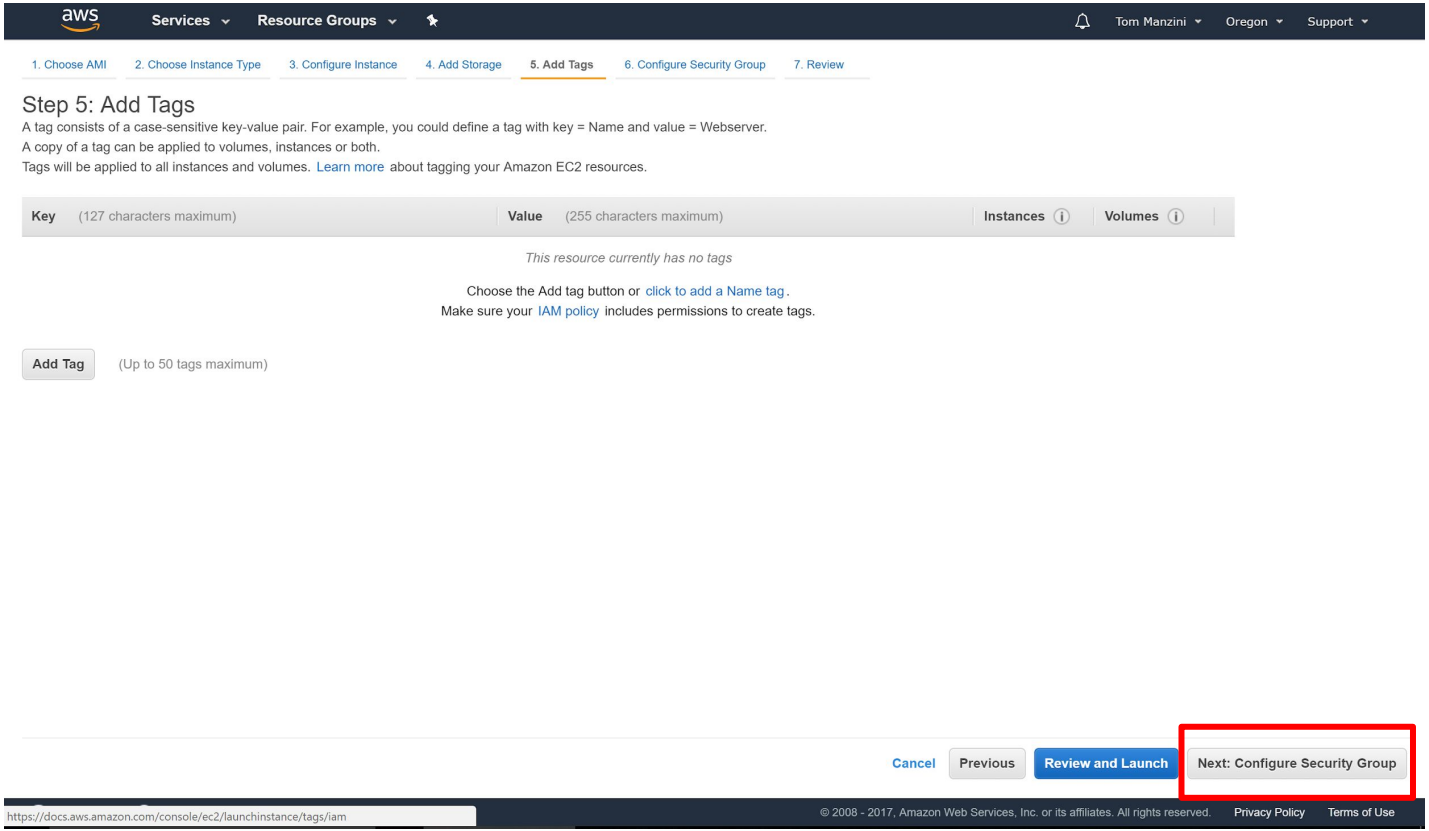
Let’s set up a basic machine



Make sure you choose 8 GB of SSD Storage Space

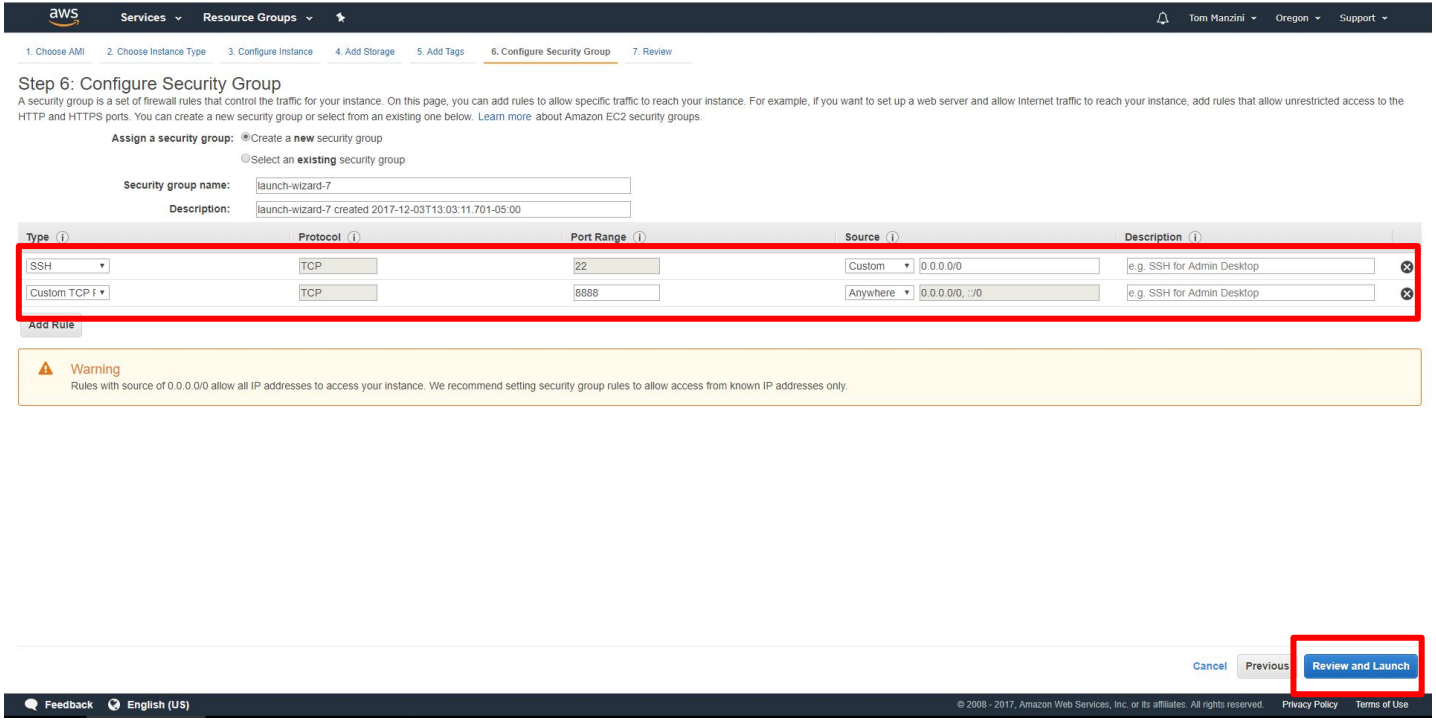
Select next

Let’s set up a basic machine



Select next

Let’s set up a basic machine

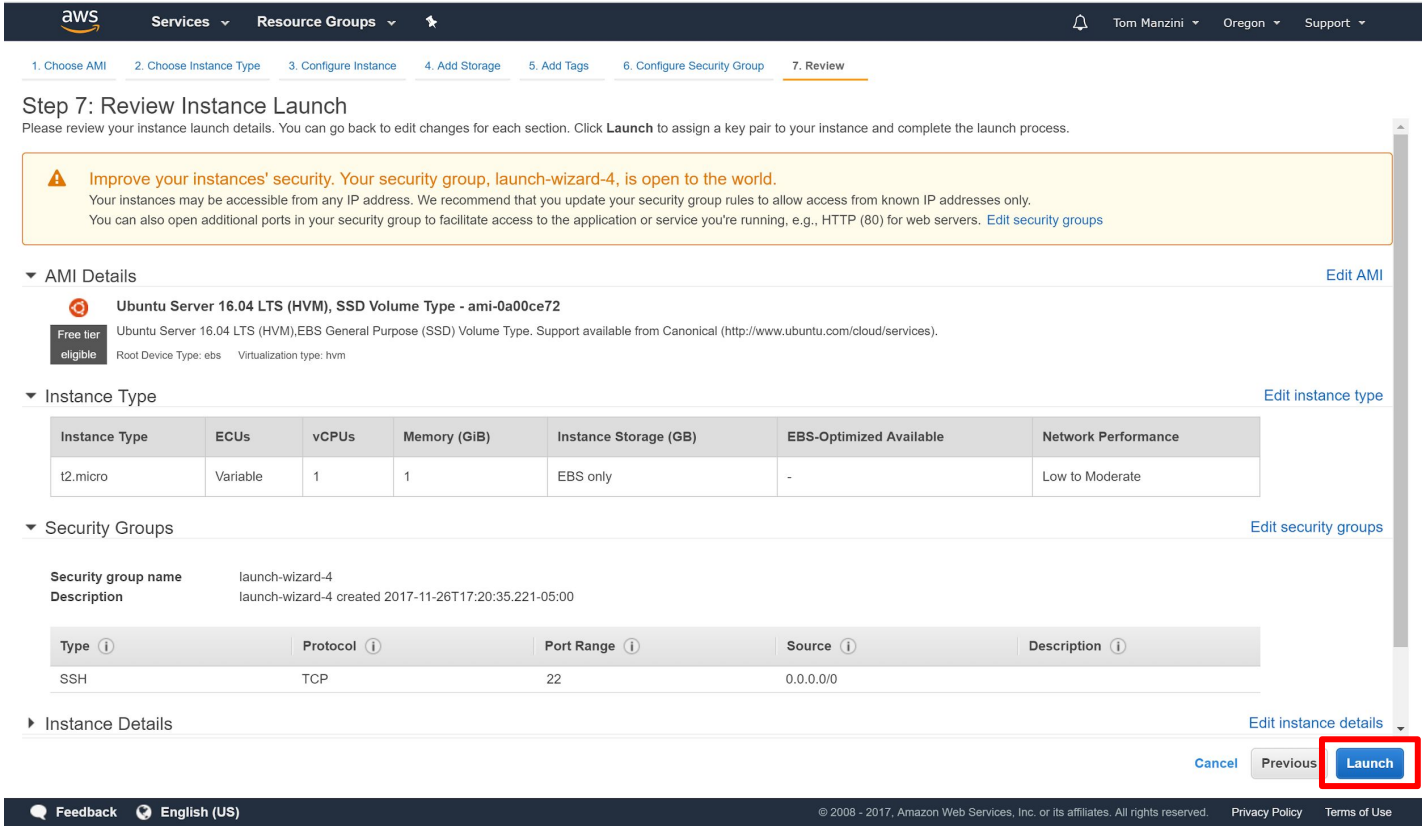


Make sure you have an SSH rule set (This should be default) & a TCP rule for 8888 so you can connect your Ipython Notebook

You can set HTTP or other rules here too if you want

Select Review and Launch

Let’s set up a basic machine

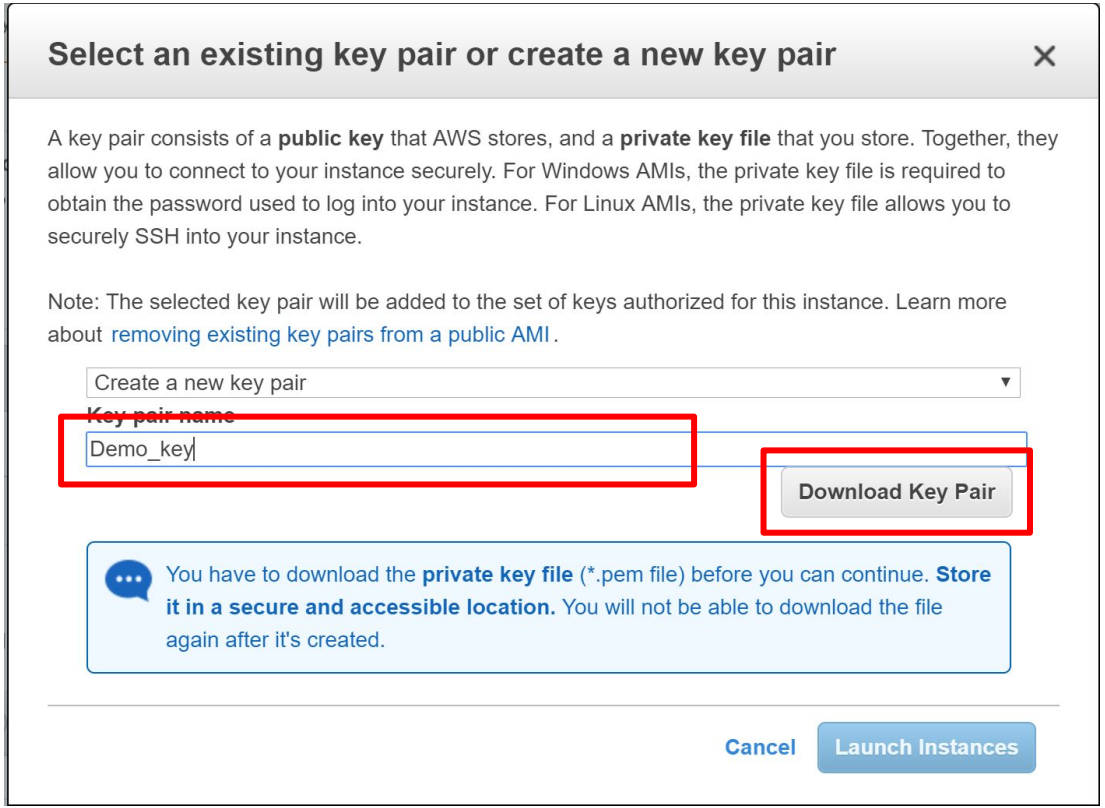


Take one last look to make sure you are happy with everything...

Select Launch

Let’s set up a basic machine

Enter a name for your private key



This window allows your to create a private key to access your machine… when you ssh you will need it

Download your key and save it

Make sure it’s in a place where it won’t get lost - this key is the only way to connect to this specific instance!

Now you have a basic AWS machine up and running

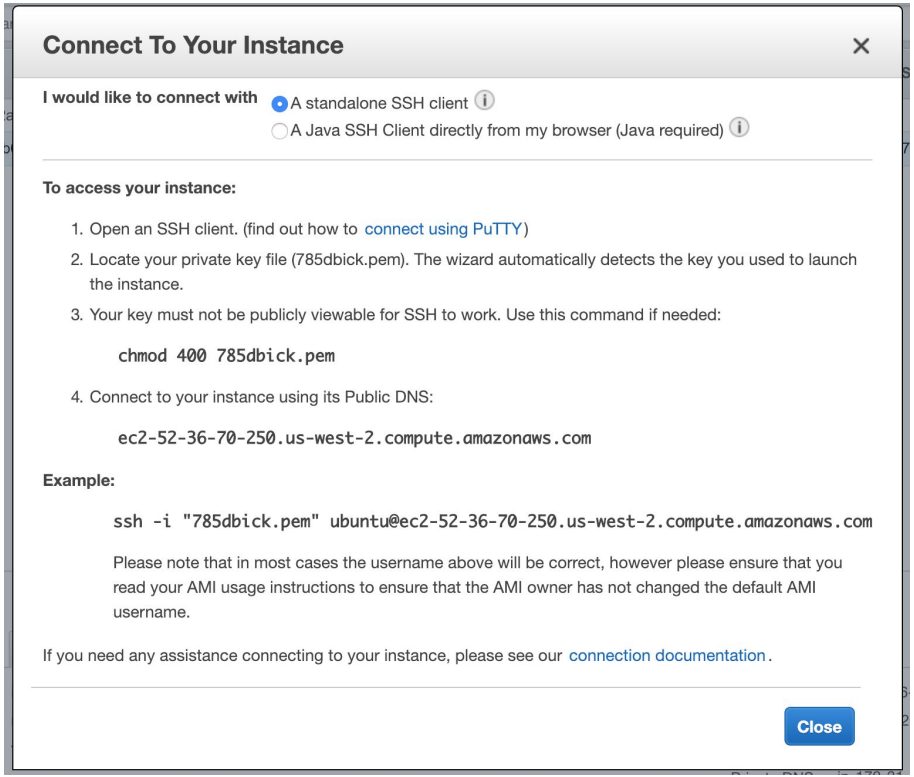
Now let’s connect to it

Your instance is now identified by its IP address. You can ssh to your instance by using this public IP address and your address key.

Let’s start by changing the instance type. For student accounts, t2.micro is free-tier eligible, but let’s still start on a t2.nano instance.

Remember your instance’s IP address changes every time you restart it

Connect to launched instance



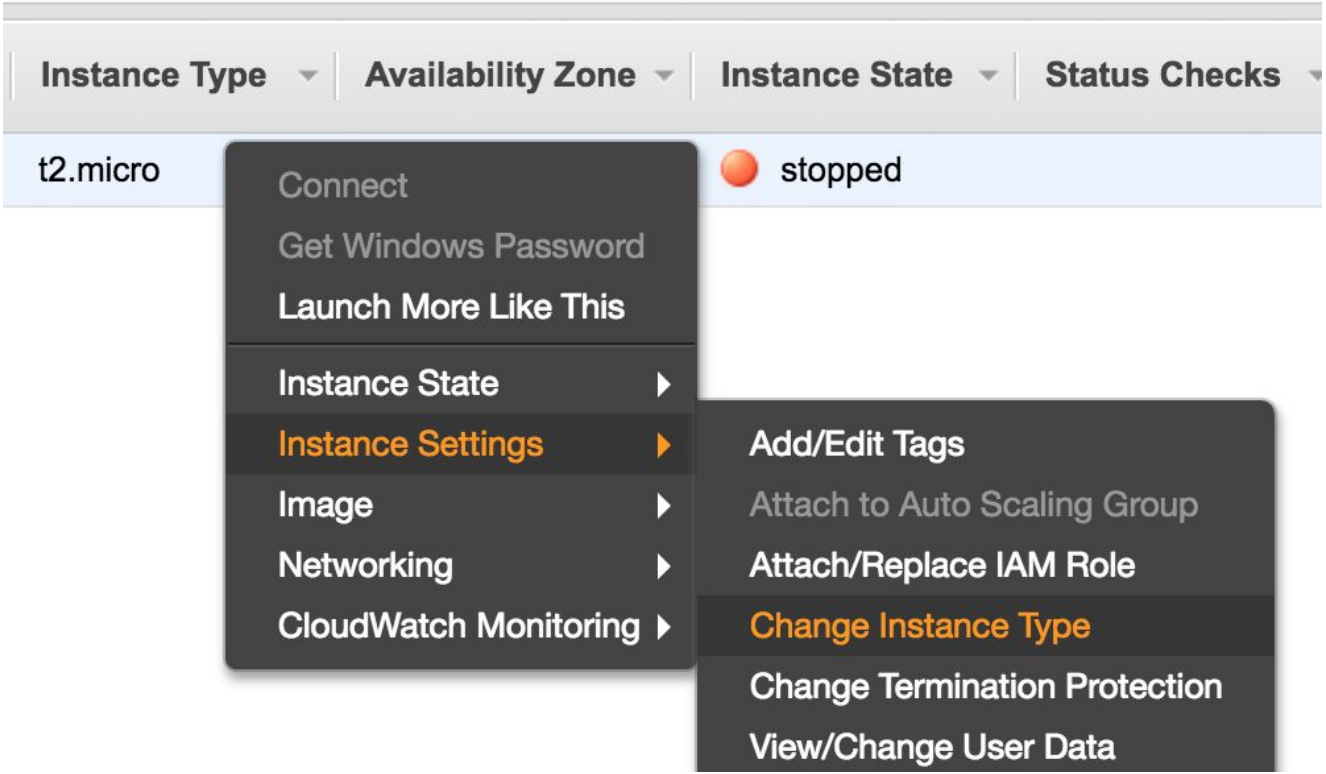
Right-click and click connect

The ssh command shown includes the currently assigned IP address.
This will change at each launch!

Note - the ssh command will use the path to the key associated with this instance.

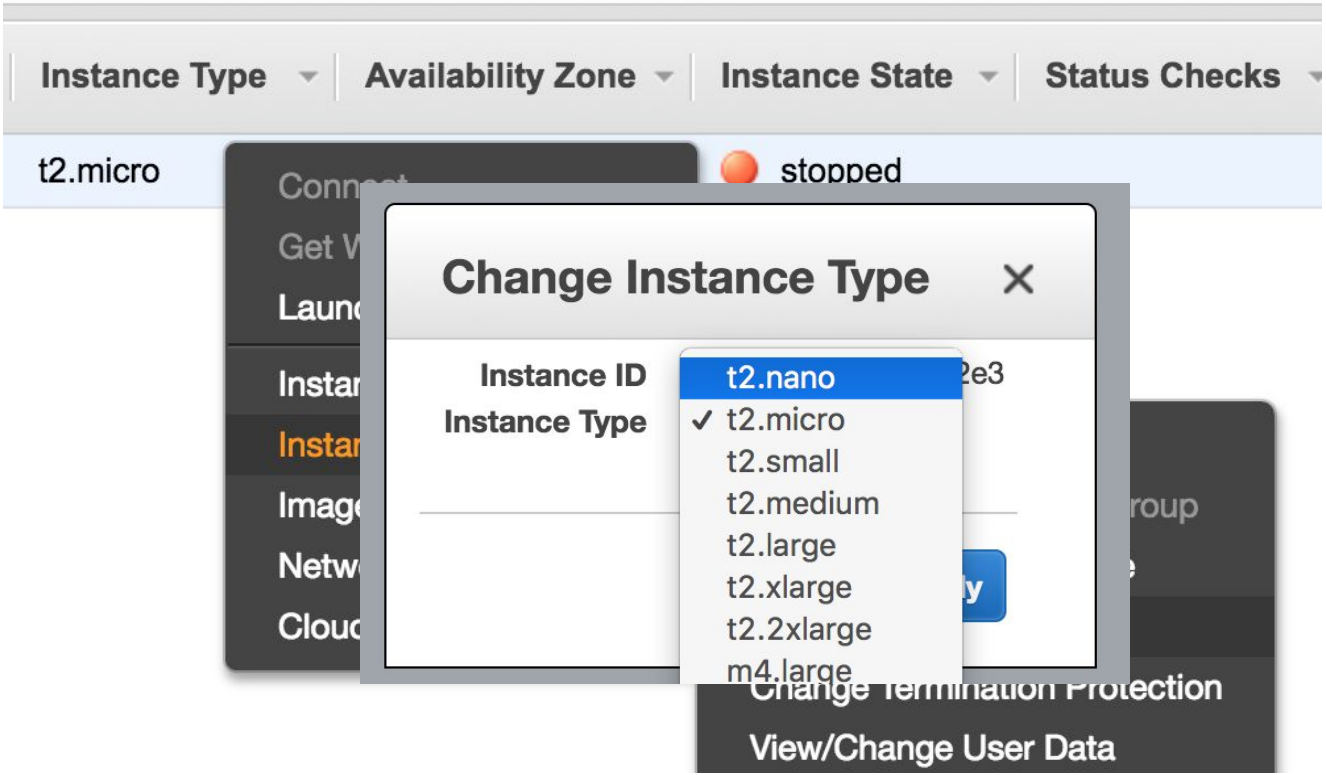
Important - will need to run chmod command every time you have a new key

Set instance type



Right-click on your instance under the instance tab, go to instance settings and change instance type to t2.nano

Set instance type



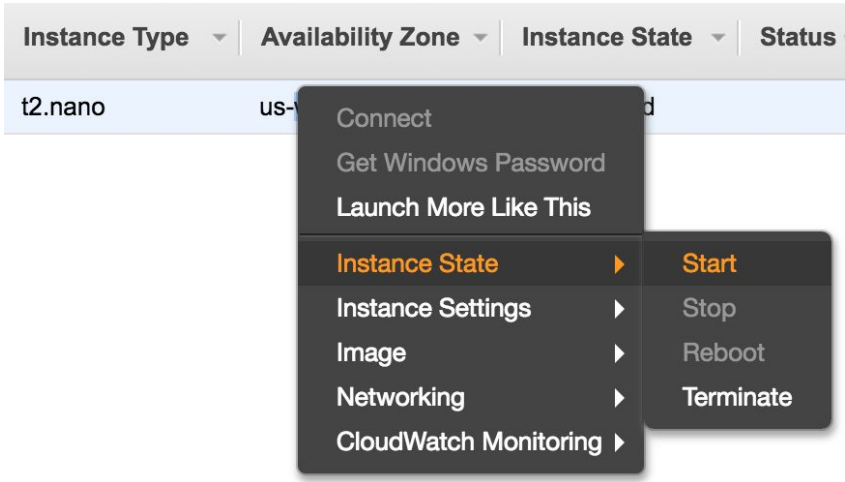
Right-click on your instance under the instance tab, go to instance settings and change instance type to t2.nano

Note

You can spin up a t2.micro for development (which allows access to all the packages in the AMI and can put all your data on the instance), and then change the instance type to a p2.xlarge for GPU access (training).

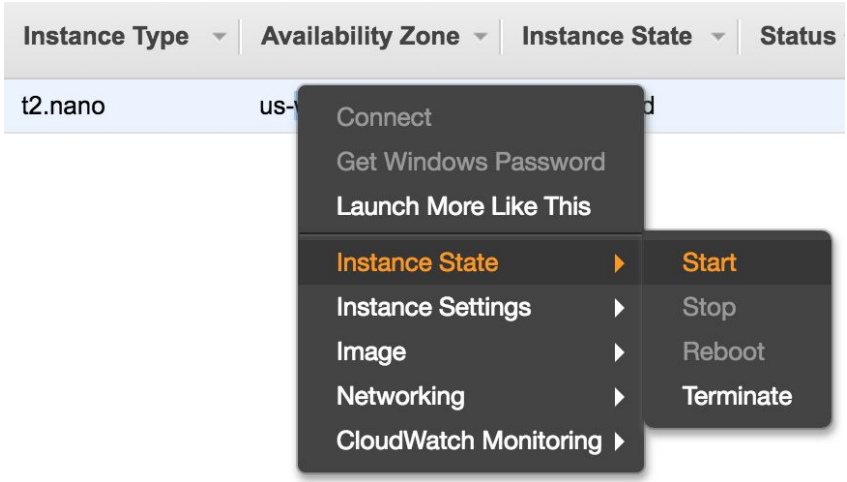
p2.xlarge is what I used for all my training, about $1 per hour, so with 3 credits of $50 that’s more than enough hours of training for almost allcases.

Launch instance



Right-click and Start your instance.

Launch instance



Right-click and Start your instance.

This will start the process of allocating resources to your instance. Once this is completed, your instance will be running and you can connect to it.

Stopping this instance removes the compute associated with the current session.

Do NOT terminate! This will wipe the slate clean. Terminate ONLY when you are sure of ‘throwing away’ the data.

Now you have a running AWS machine and you can connect

Now you know how to use machines on EC2

Let’s run a trivial little problem

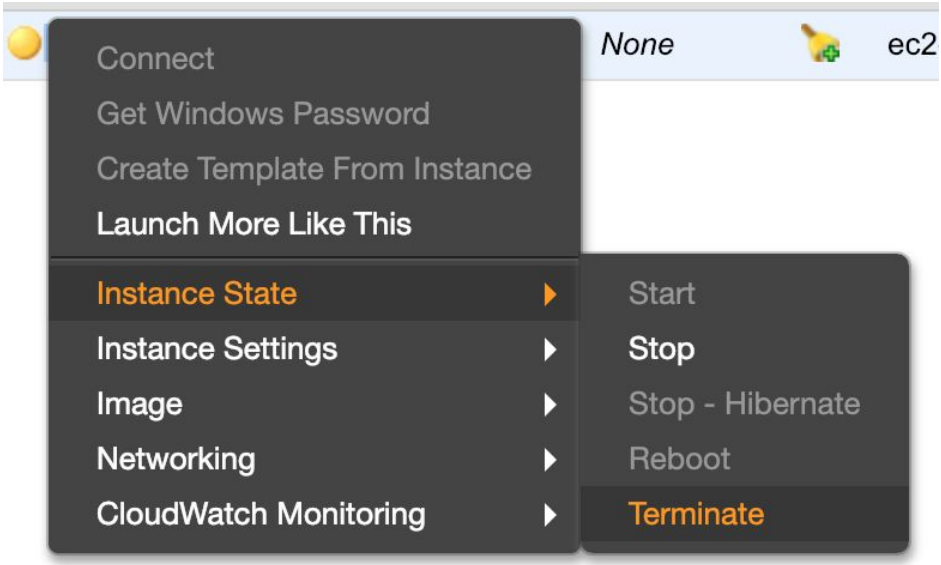
A Toy Problem to Test your AWS Instance

Will Test:

- Working instance
- Good Python installation
- Working PyTorch installed
- GPU / CUDA support
- Your ability to login and execute code on AWS

```
import torch


def main():
    GPU = torch.cuda.is_available()
    mat_size = (100, 100)
    cpu_mat_0 = torch.zeros(size = mat_size)
    cpu_mat_1 = torch.ones(size = mat_size)
    gpu_mat_0 = torch.zeros(size = mat_size)
    gpu_mat_1 = torch.ones(size = mat_size)


    if GPU:
        gpu_mat_0 = gpu_mat_0.cuda()
        gpu_mat_1 = gpu_mat_1.cuda()
        print("Using GPU")


    cpu_res = cpu_mat_0 + cpu_mat_1
    gpu_res = gpu_mat_0 + gpu_mat_1


    try:
        print(bool(torch.all(cpu_res == gpu_res)))
        return 0
    except:
        print("If using GPU, should be here")
    gpu_res = gpu_res.detach().cpu() # detach is for gradient computations
    print(bool(torch.all(cpu_res == gpu_res)))
    return 0

main()
```

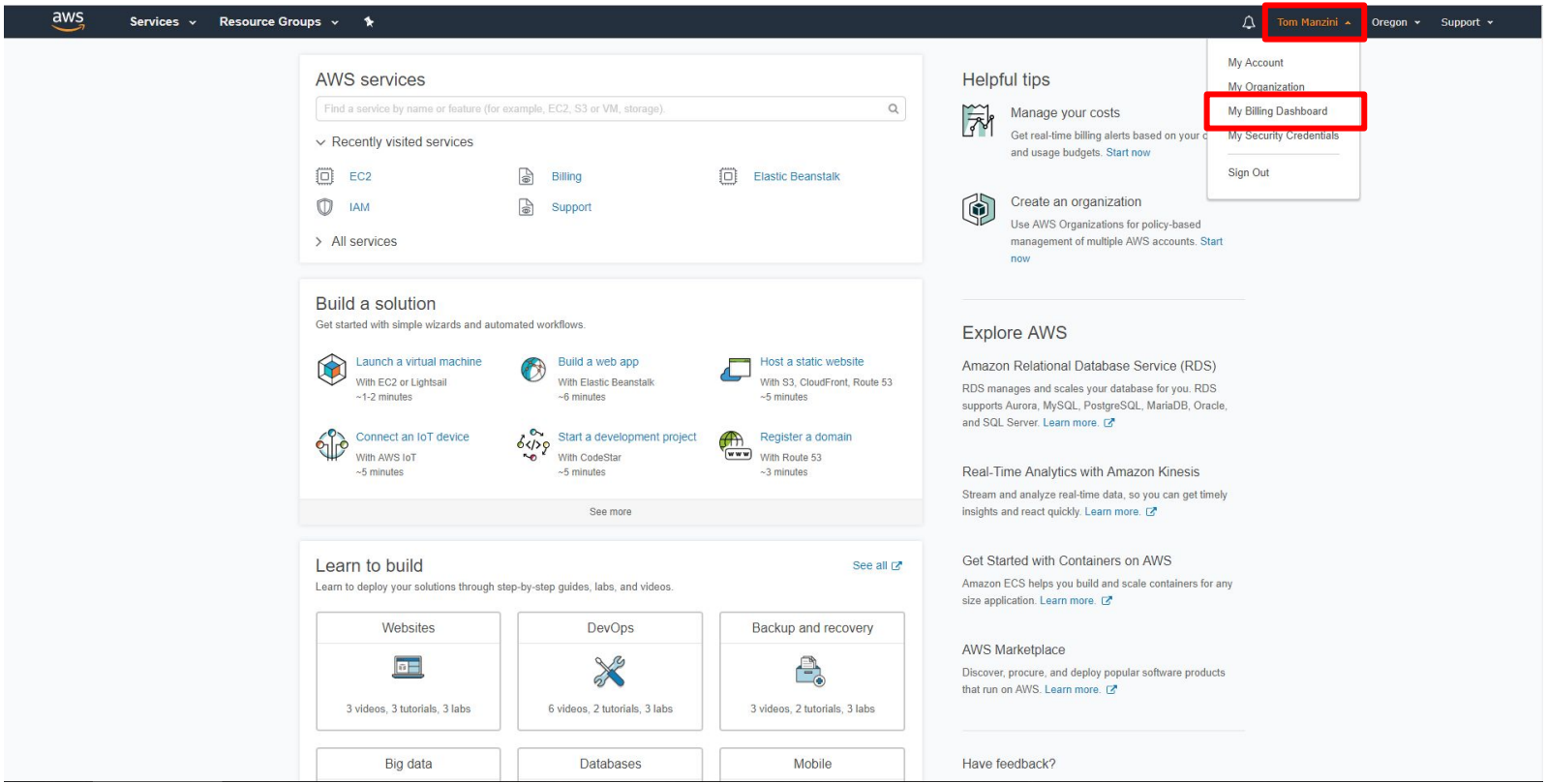
And let’s shut it down so you don’t get billed



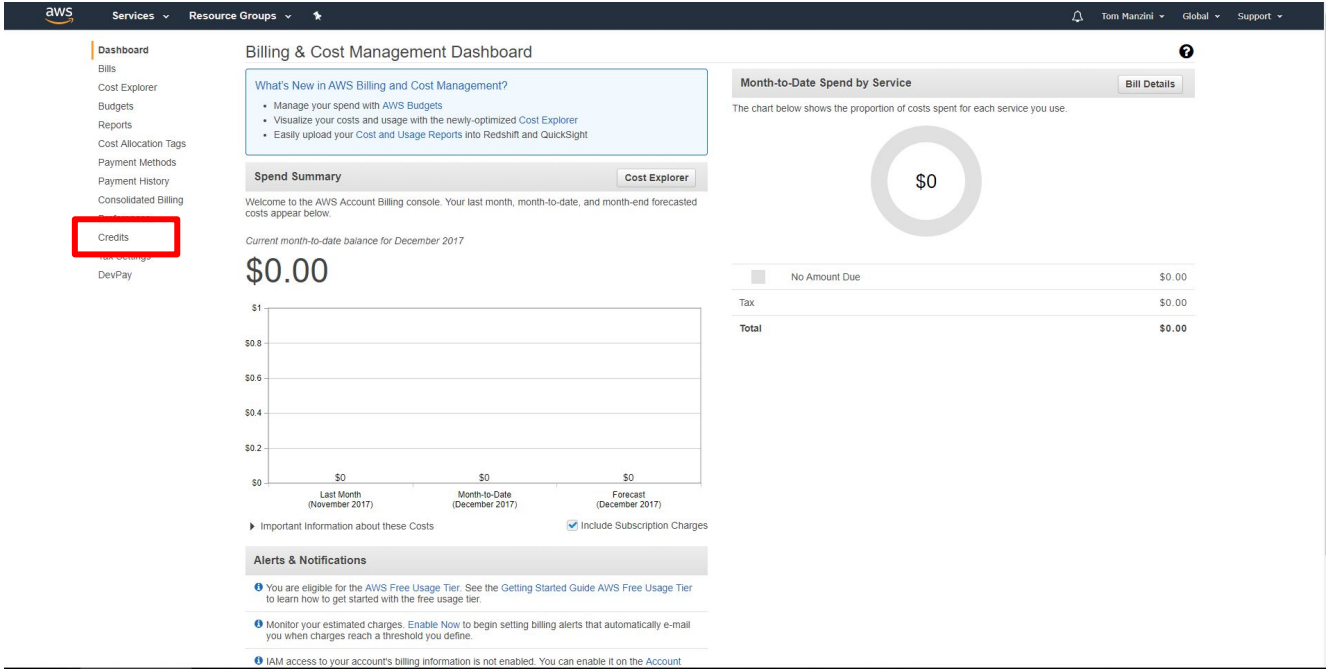
If you stop the instance you can just start the instance and resume whenever you want.

If you are done, terminate.

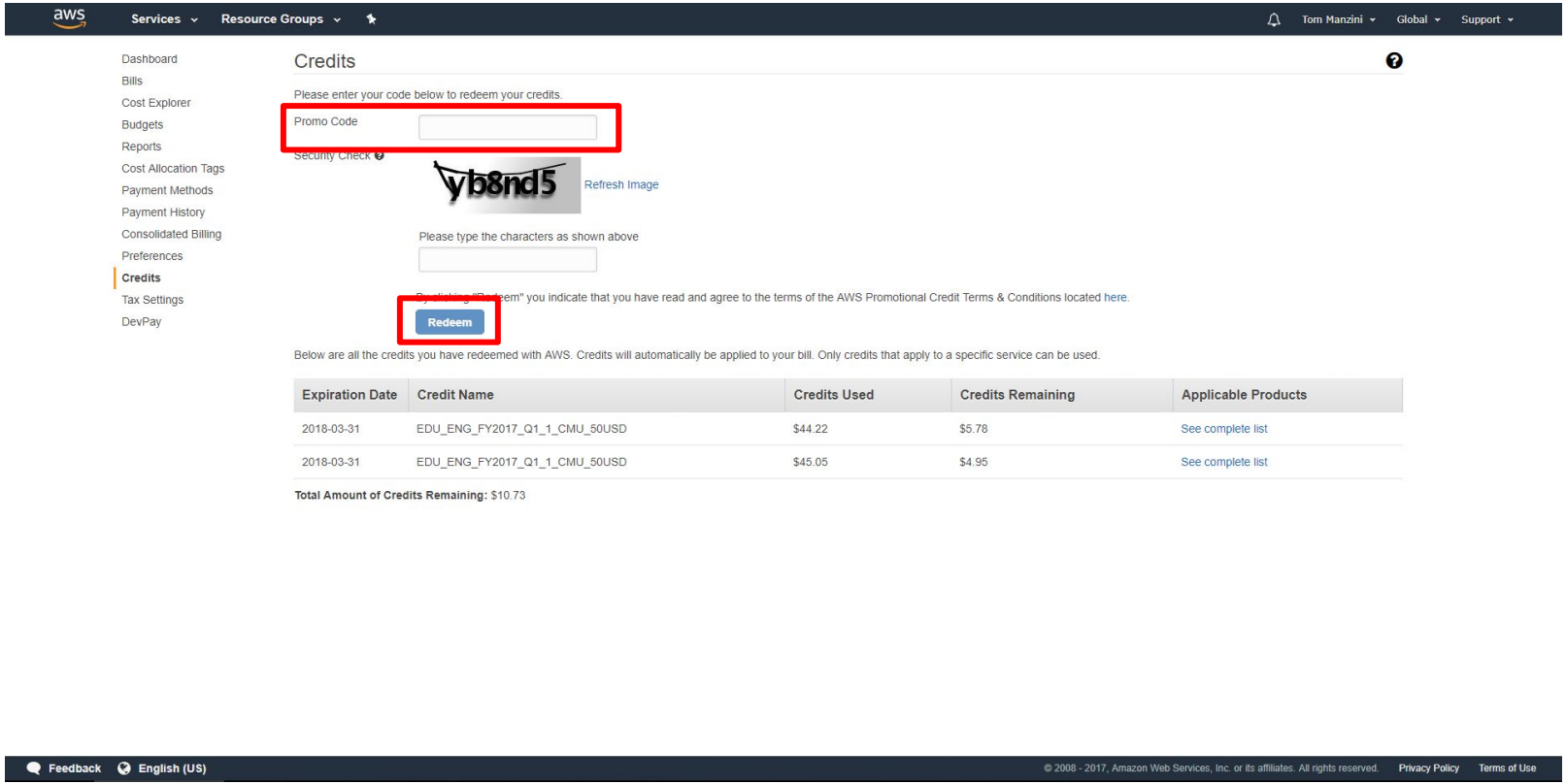
Now lets redeem some AWS credits



Now lets redeem some AWS credits



Now lets redeem some AWS credits



Now you know how to use machines on EC2 and you can afford to use the expensive ones

Finally, some useful tidbits

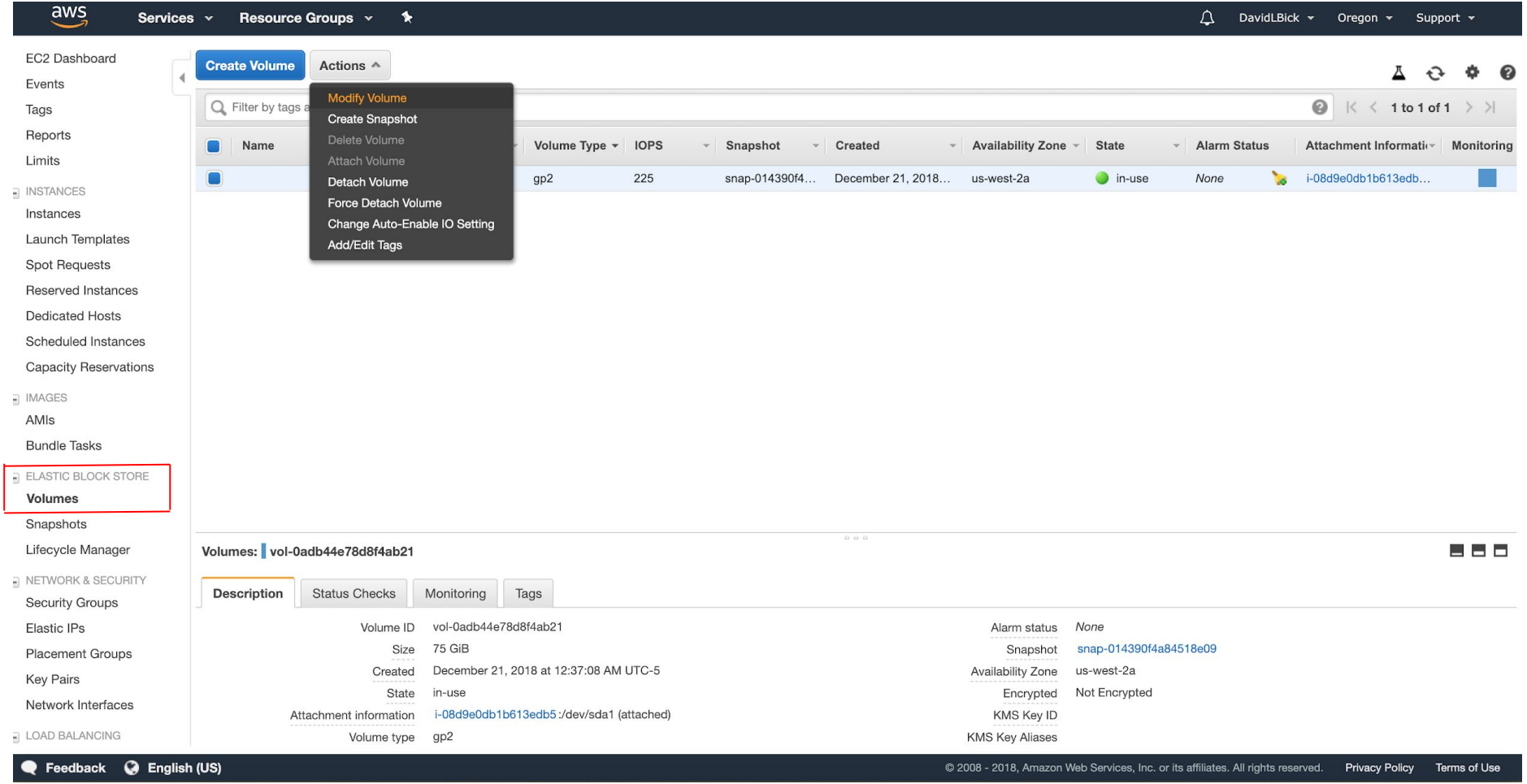
Remember to refer back here later in the semester, likely won’t remember all this right now

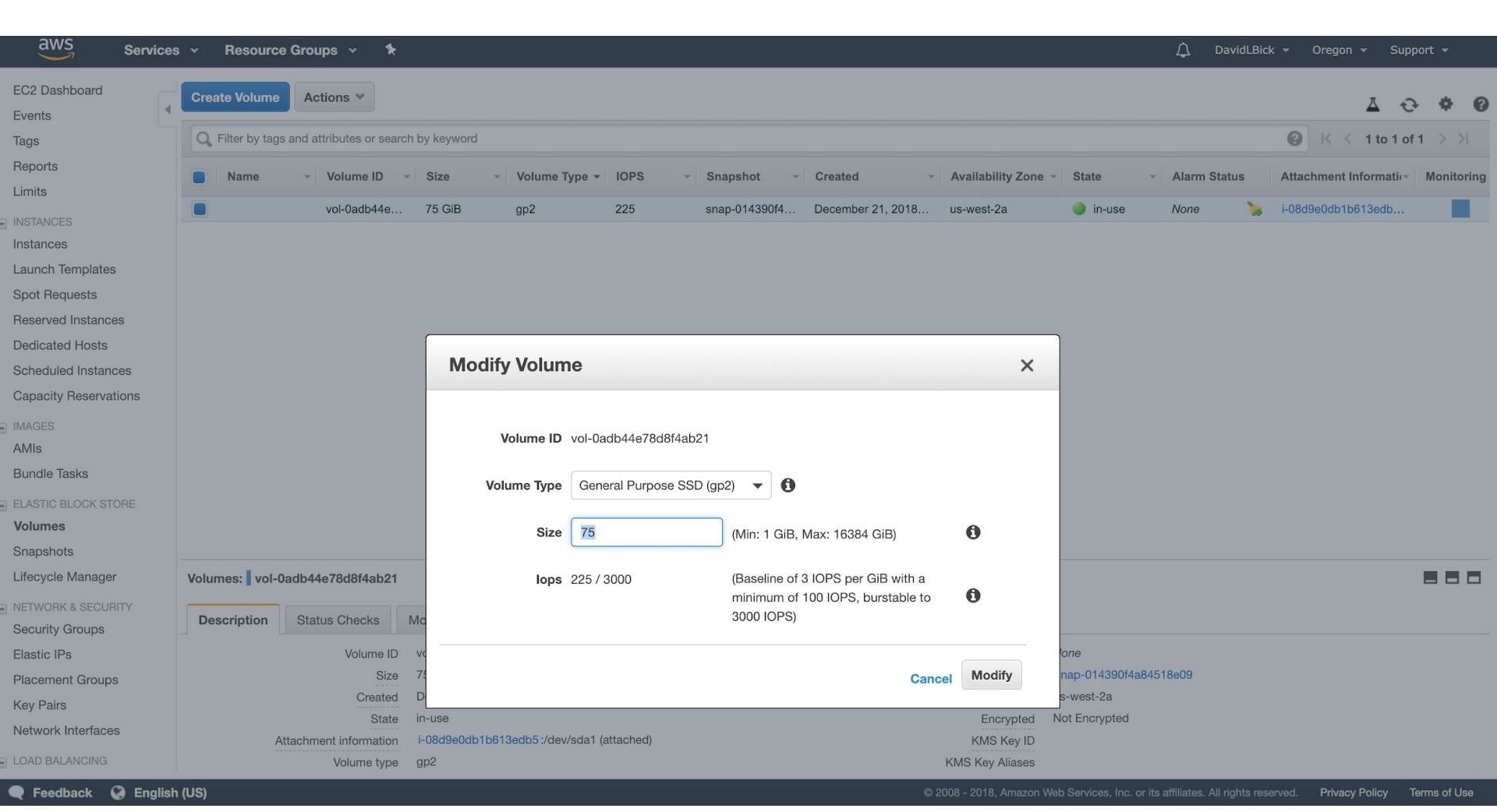
Editing Volume

If you run out of space, you can adjust your volume without shutting off the instance, happened to me on hw2p2

https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ebs-modify-volume.html

https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/console-modify.html


aws
Services
Resource Groups
DavidLBick
Oregon
Support
EC2 Dashboard
Events
Tags
Reports
Limits
INSTANCES
Instances
Launch Templates
Spot Requests
Reserved Instances
Dedicated Hosts
Scheduled Instances
Capacity Reservations
IMAGES
AMIs
Bundle Tasks
ELASTIC BLOCK STORE
Volumes
Snapshots
Lifecycle Manager
NETWORK & SECURITY
Security Groups
Elastic IPs
Placement Groups
Key Pairs
Network Interfaces
LOAD BALANCING
Create Volume
Actions
Modify Volume
Create Snapshot
Delete Volume
Attach Volume
Detach Volume
Force Detach Volume
Change Auto-Enable IO Setting
Add/Edit Tags
Filter by tags
1 to 1 of 1
Name
Volume Type
IOPS
Snapshot
Created
Availability Zone
State
Alarm Status
Attachment Informati
Monitoring
gp2
225
snap-014390f4...
December 21, 2018...
us-west-2a
in-use
None
i-08d9e0db1b613edb...
Volumes: vol-0adb44e78d8f4ab21
Description
Status Checks
Monitoring
Tags
Volume ID
vol-0adb44e78d8f4ab21
Size
75 GiB
Created
December 21, 2018 at 12:37:08 AM UTC-5
State
in-use
Attachment information
i-08d9e0db1b613edb5:/dev/sda1 (attached)
Volume type
gp2
Alarm status
None
Snapshot
snap-014390f4a84518e09
Availability Zone
us-west-2a
Encrypted
Not Encrypted
KMS Key ID
KMS Key Aliases
Feedback
English (US)
© 2008 - 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use


aws
Services
Resource Groups
DavidLBick
Oregon
Support
EC2 Dashboard
Events
Tags
Reports
Limits
INSTANCES
Instances
Launch Templates
Spot Requests
Reserved Instances
Dedicated Hosts
Scheduled Instances
Capacity Reservations
IMAGES
AMIs
Bundle Tasks
ELASTIC BLOCK STORE
Volumes
Snapshots
Lifecycle Manager
NETWORK & SECURITY
Security Groups
Elastic IPs
Placement Groups
Key Pairs
Network Interfaces
LOAD BALANCING
Create Volume
Actions
Filter by tags and attributes or search by keyword
1 to 1 of 1
Name
Volume ID
Size
Volume Type
IOPS
Snapshot
Created
Availability Zone
State
Alarm Status
Attachment Informati
Monitoring
vol-0adb44e...
75 GiB
gp2
225
snap-014390f4...
December 21, 2018...
us-west-2a
in-use
None
i-08d9e0db1b613edb...
Modify Volume
Volume ID vol-0adb44e78d8f4ab21
Volume Type General Purpose SSD (gp2)
Size 75
(Min: 1 GiB, Max: 16384 GiB)
Iops 225 / 3000
(Baseline of 3 IOPS per GiB with a minimum of 100 IOPS, burstable to 3000 IOPS)
Cancel
Modify
Volumes: vol-0adb44e78d8f4ab21
Description
Status Checks
Volume ID
Size
Created
State in-use
Attachment information i-08d9e0db1b613edb5 :/dev/sda1 (attached)
Volume type gp2
Encrypted Not Encrypted
KMS Key ID
KMS Key Aliases
Feedback
English (US)
© 2008 - 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use

Here’s How to Connect to Jupyter Notebook

Call ssh -N -L localhost:8888:localhost:8887 -i ~/path/to/key ubuntu@ip

Create new terminal window, ssh into instance

In ssh window, source activate pytorch_p36

Call jupyter notebook --no-browser --port=8887

Go to browser, enter localhost:8888

Parting wisdom

- Remember to shut down your machines
  - Just because you don’t have an open ssh connection doesn’t mean your machine is off
  - When you’re not running code, Stop; when you can get rid of the data, Terminate

- Use PyTorch on the previously given AMI
  - After you ssh into the instance, run command ‘source activate pytorch_p36’
  - Without this you cannot import torch
  - Unless you are already very comfortable with tensorflow, it is much easier to get help from TAs on PyTorch

Parting wisdom

- You need permission to launch a GPU instance
  - You will need to create a support ticket to launch GPU Machines
  - If you attempt to launch on you will be guided through the process
  - You are typically only allowed to launch 1 GPU machine at a time

- Only launch an expensive instance when it is time to train, not develop
  - Launch a basic instance with everything you need to develop and test your code
  - When it comes time to train your system for real, then launch a decked out instance
  - Saves you money

Parting wisdom

If you want to use a local IDE instead of VIM on your ssh window, you can develop locally and run an scp to transfer the file to the aws instance

scp -i ~/path/to/key ~/path/to/file/ ubuntu@ec2…

If you use jupyter notebook connected to aws, no need to worry about this.

While writing code, I kept a note with common commands that I could just change the IP address on, such as an ssh command, and scp of my local code

Requesting Instances

Must request access to instances from AWS in the Support Center section of AWS. Check the latest status on AWS, as they may have changed something or upgraded offerings. Currently, should request EC2 instance limit increase, but be sure that it has not changed.

How to Request

Instructions to apply for GPU access on AWS

Starting from aws.amazon.com
Top right corner, “My Account” -> “Account Settings”
At top right corner click “Support” -> Support Center
Click “Create case”
Click “Service Limit increase”
Limit type -> EC2 Instance
If Pitt, select Region US East (Ohio). If SV, select Region US West (Oregon)
New limit value -> 1 Instance type -> p2.xlarge (can also request other instance types)
Use case description -> Describe that you are in this class at CMU and you need a GPU to train deep learning models for the homeworks
Specify your contact method