How to use NGC Containers on AWS

Scott Ellis & Jeff Weiss GTC 2018 S8276

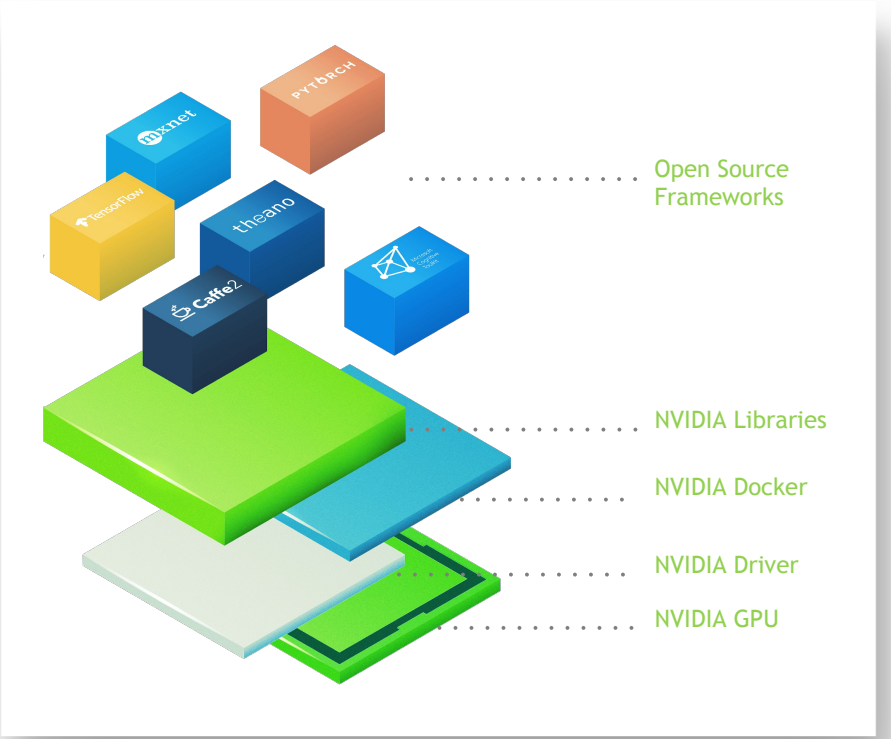
What is NGC? Why is NGC?

Challenges with Complex Software

Current DIY GPU-accelerated AI and HPC deployments are **complex** and **time consuming** to build, test and maintain

Development of software frameworks by the community is moving **very fast**

Requires high level of **expertise** to manage driver, library, framework dependencies



GPU-Accelerated Deep Learning Containers

Deep Learning Everywhere, for Everyone

Innovation for Every Industry

Quickly tap into the power of NVIDIA AI, from automotive, to healthcare, to fintech, and more

Say Goodbye to DIY

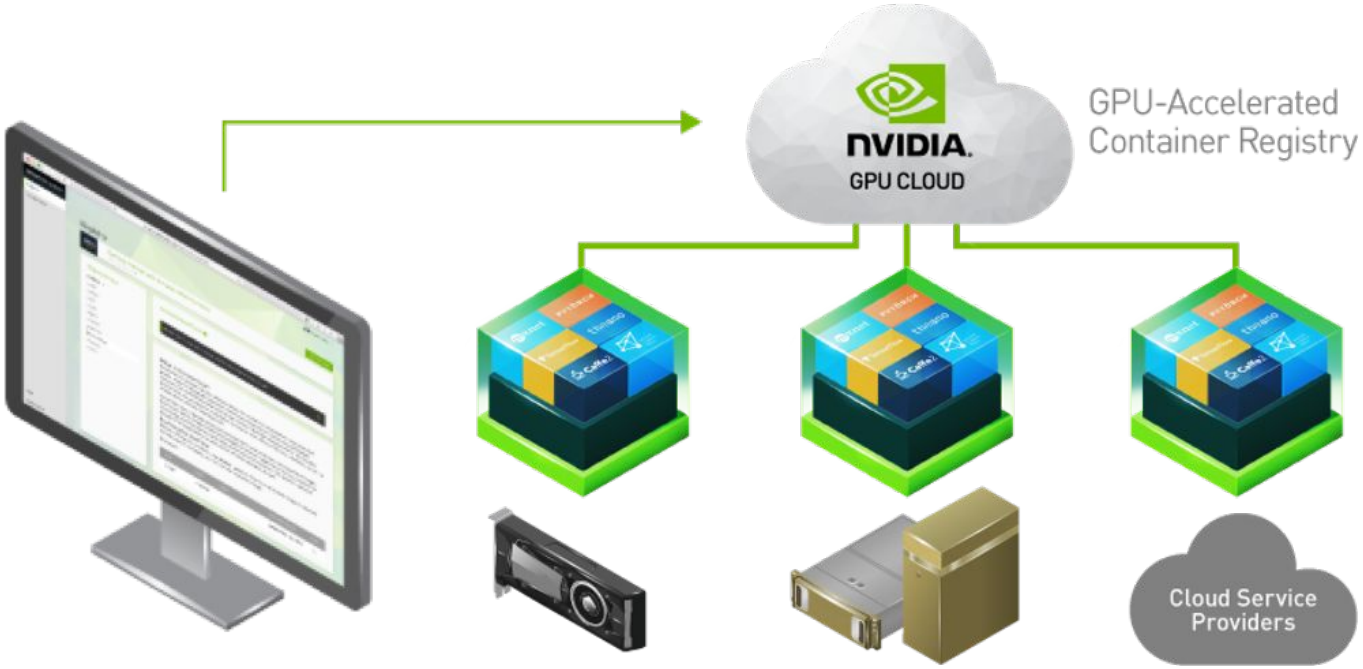
Deep learning software containers, tuned, tested, and certified by NVIDIA

Stay Up To Date

Monthly updates to deep learning containers

NVIDIA GPU Cloud integrates GPU-optimized deep learning frameworks, runtimes, libraries, and OS into a ready-to-run container, available at no charge

Deep Learning Across Platforms



NVIDIA TITAN (powered by NVIDIA Volta or NVIDIA Pascal)

NVIDIA DGX-1 and DGX Station

Amazon EC2 P3 instances with NVIDIA Volta

Three Steps To Deep Learning On Amazon EC2 with NGC

SIGN UP

To get an NGC account, go to:

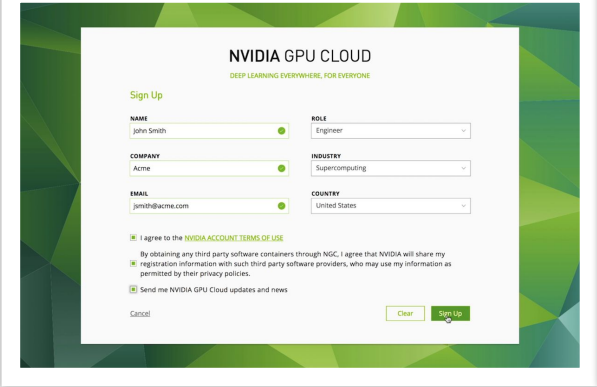
www.nvidia.com/ngcsignup

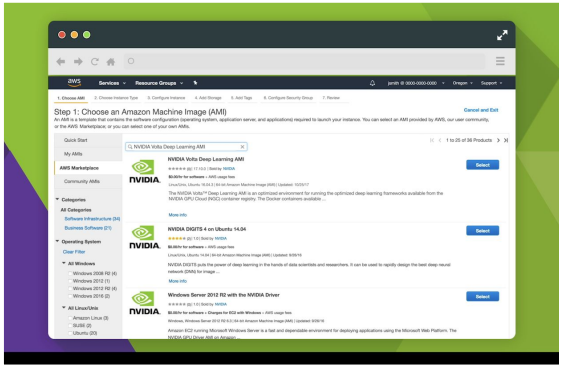
DEPLOY IMAGE

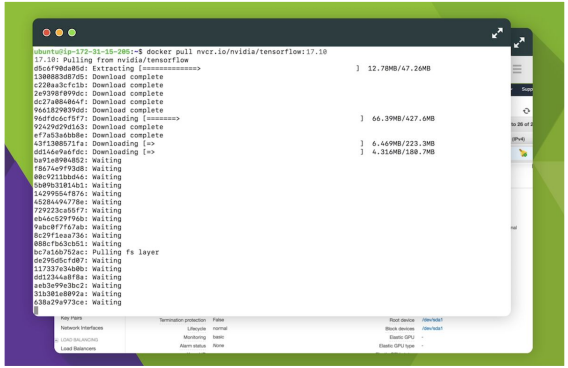
On Amazon EC2, choose a P3 instance and deploy the **NVIDIA Volta Deep Learning AMI for NGC**

PULL CONTAINER

Pick your desired framework (TensorFlow, PyTorch, MXNet, etc.), and pull the container into your instance



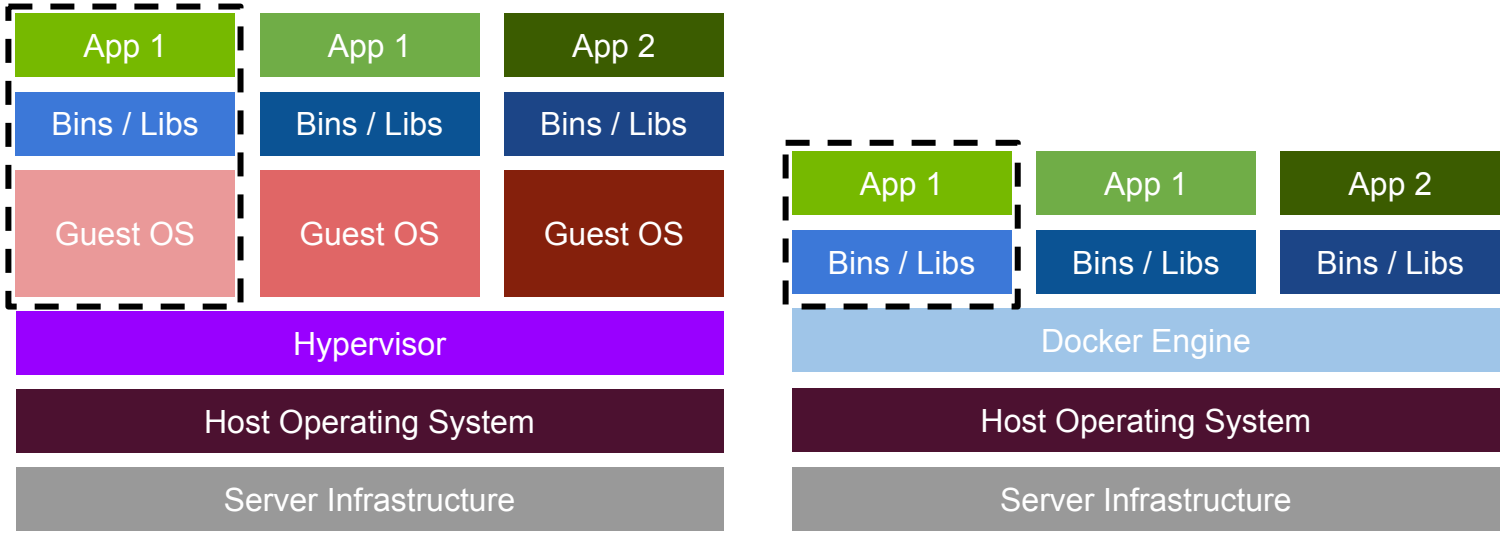




What is this container thing you speak of?

Virtual Machine vs. Container

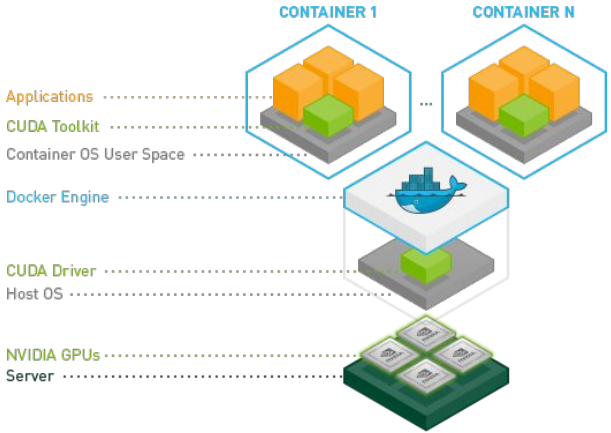
Not so similar



Virtual Machines

Containers

NVIDIA Container Runtime



https://github.com/NVIDIA/nvidia-docker

- ➔ Colloquially called “nvidia-docker”
- ➔ Docker containers are *hardware-agnostic* and *platform-agnostic*
- ➔ NVIDIA GPUs are specialized hardware that require the NVIDIA driver
- ➔ Docker does not natively support NVIDIA GPUs with containers
- ➔ NVIDIA Container Runtime makes the images agnostic of the NVIDIA driver
  - ◆ Required character devices and driver files are mounted when starting the container on the target machine
  - ◆ This makes Docker images portable while still leveraging NVIDIA GPUs

Docker Terms

Definitions

Image
Docker images are the basis of containers. An Image is an ordered collection of root filesystem changes and the corresponding execution parameters for use within a container runtime. An image typically contains a union of layered filesystems stacked on top of each other. An image does not have state and it never changes.

Container
A container is a runtime instance of a docker image.
A Docker container consists of

- A Docker image
- Execution environment
- A standard set of instructions

https://docs.docker.com/engine/reference/glossary/

Managing Images and Containers

Common Commands

List Images:

```
docker images
```

Remove an Image:

```
docker rmi imageID
```

```
docker rmi tensorRT
```

List Containers:

```
docker ps -a
```

Stop a running Container:

```
docker stop containerID
```

Remove a Container:

```
docker rm containerID
```

Note: imageID and containerID can be either a hash or a name

Running Containers

docker run and option

`docker run` Options

- ➔ `--runtime=nvidia` enable GPU capabilities
- ➔ `--rm` remove the container after it exits
- ➔ `-i -t` or `-it` interactive, and connect a "tty"
- ➔ `-d --detach` run in the background
- ➔ `--name` give the container a name
- ➔ `-p 8080:80` port map from host to container
- ➔ `-v ~/data:/data` map storage volume from host to container (bind mount) i.e. bind the `~data` directory in your home directory to `/data` in the container

Starts Tensorflow with ports, volumes, and console (All 1 line):

```
docker run
    --runtime=nvidia
    --rm -it
    --name MyCoolContainer
    -p 8888:80
    -v ~/data:/data
    nvcr.io/nvidia/tensorflow:18.01-py2
    examples/nvcnn.py
```

Navigating the NGC WebUI

NVIDIA GPU Cloud

How do we actually use it?

Our challenge:

- Sign up for a *free* NGC account at www.nvidia.com/ngcsignup
- Login to the WebUI
- Generate an API key for Docker to use

What is an API Key?

(And why do you need one?)

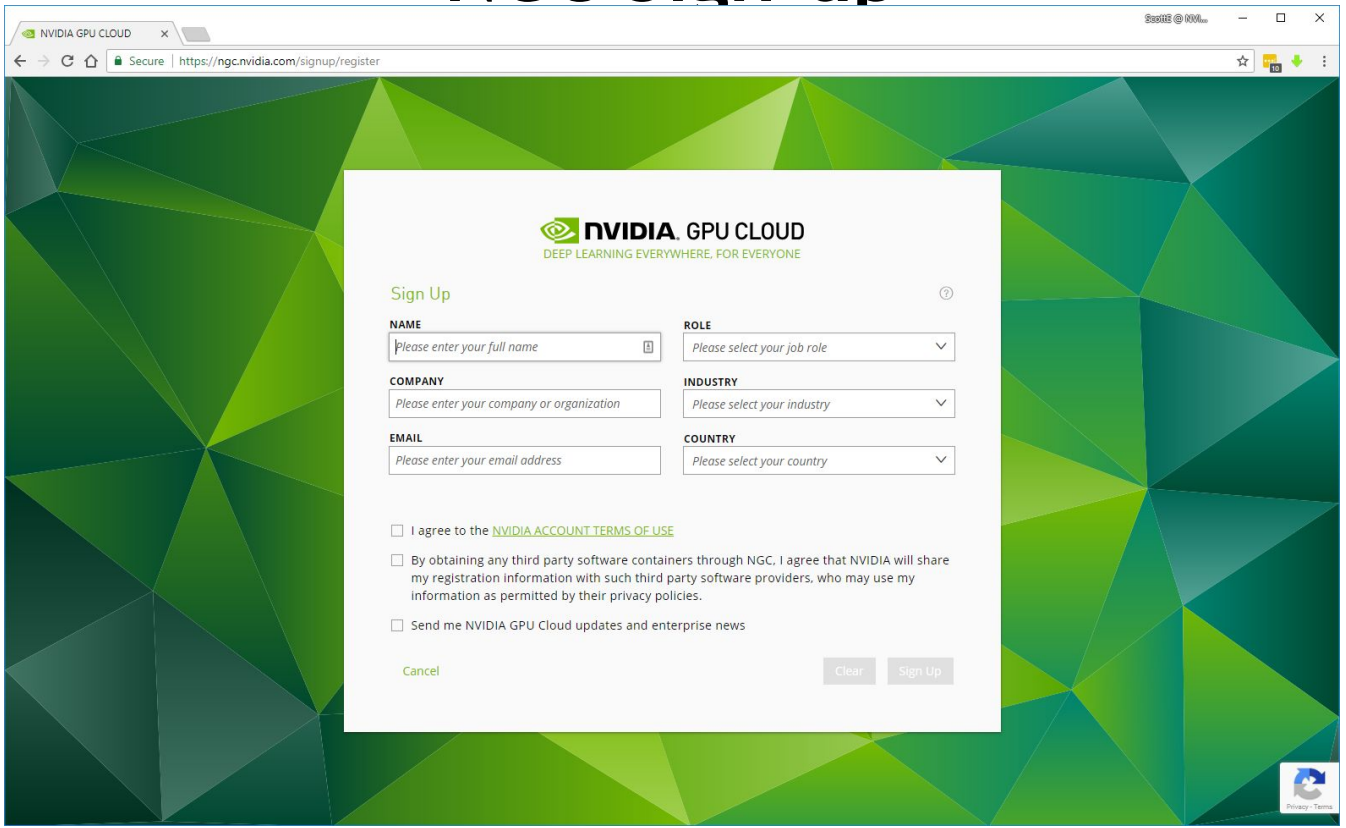
Your API key represents your credentials

- Used for programmatic interaction (e.g., docker, REST API, etc.)
- Uniquely identifies you (think “Username & Password”)
- There can be only one (regenerating your API key invalidates the old one)

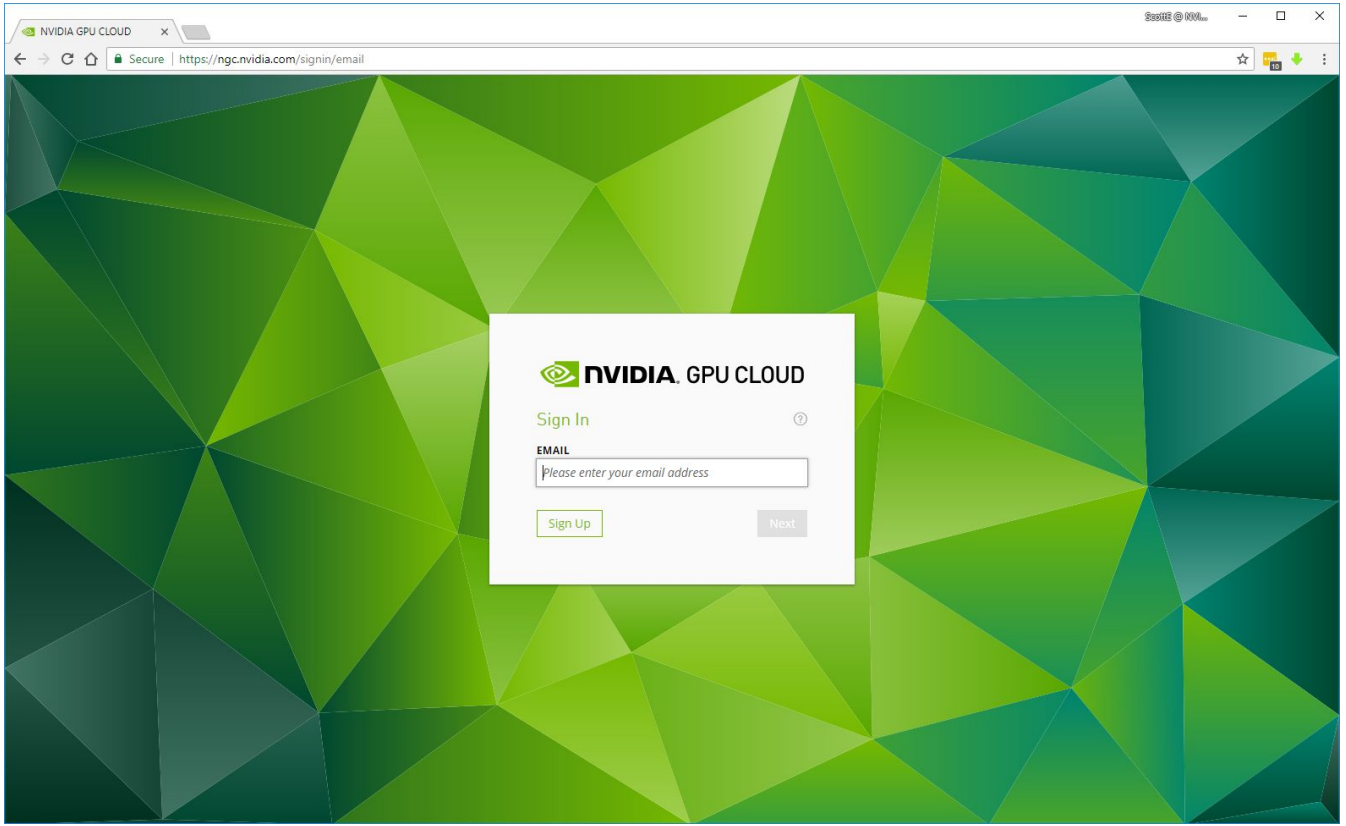
WebUI at ngc.nvidia.com: Use Username & Password

Programmatic interface at nvcr.io: Use API Key

NGC Sign-up


NVIDIA GPU CLOUD
Secure | https://ngc.nvidia.com/signup/register
NVIDIA GPU CLOUD
DEEP LEARNING EVERYWHERE, FOR EVERYONE
Sign Up
NAME
Please enter your full name
ROLE
Please select your job role
COMPANY
Please enter your company or organization
INDUSTRY
Please select your industry
EMAIL
Please enter your email address
COUNTRY
Please select your country
I agree to the NVIDIA ACCOUNT TERMS OF USE
By obtaining any third party software containers through NGC, I agree that NVIDIA will share my registration information with such third party software providers, who may use my information as permitted by their privacy policies.
Send me NVIDIA GPU Cloud updates and enterprise news
Cancel
Clear
Sign Up

NGC Access

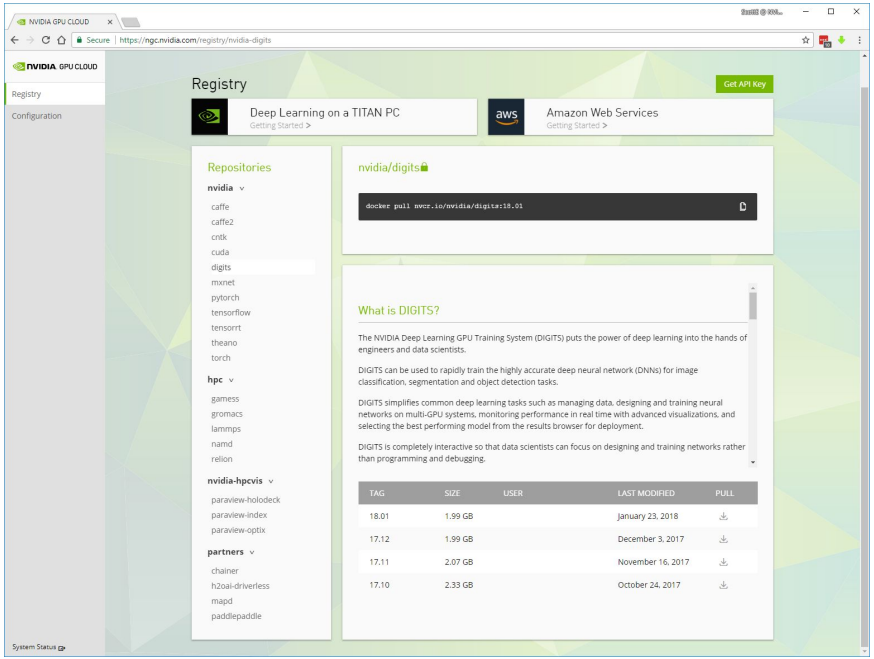

NVIDIA GPU CLOUD
Secure | https://ngc.nvidia.com/signin/email
NVIDIA GPU CLOUD
Sign In
EMAIL
Please enter your email address
Sign Up
Next

NGC WebUI

Where it all begins

When you login…

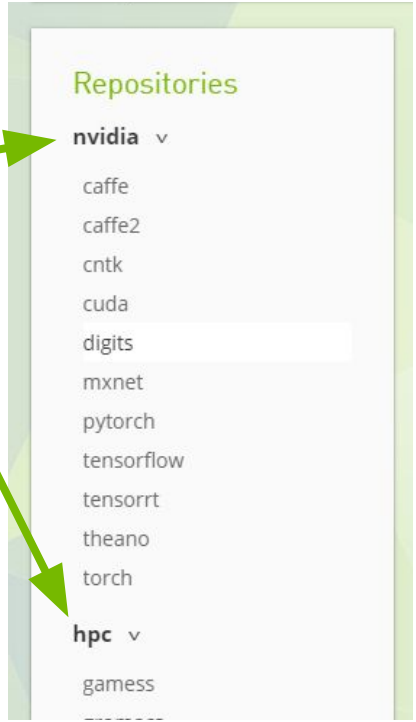
- How-to Pages
- Repository List
- Instructions and Info
- Image specifics
- Docker pull shortcut



NGC WebUI

Repository List

- Repositories
  - Used to group containers by functionality
- Containers
  - Descriptive container name
  - Details when you click on one

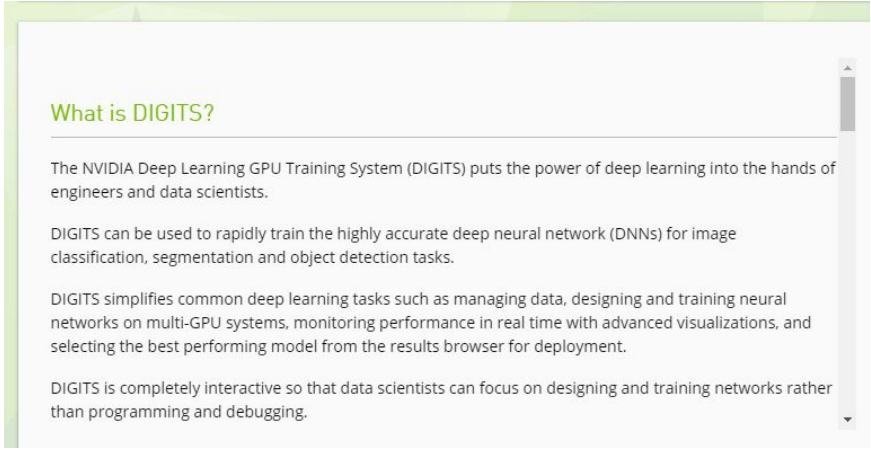


NGC WebUI

Instructions and Information

When you select a container

- Description on what the image contains
- Usually examples on running it
- Often has links for more information and tutorials

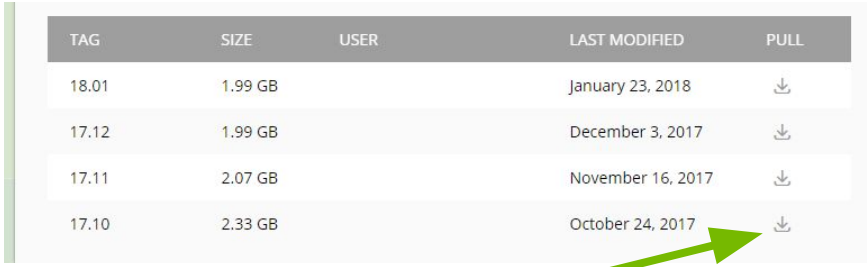


NGC WebUI

Image Specifics

List of images for that container

- Tag (`nvidia/caffe:18.01`)
  - Follows YY.MM format
- Creation date
  - Updated monthly
- Shortcut to copy `docker pull` command to clipboard



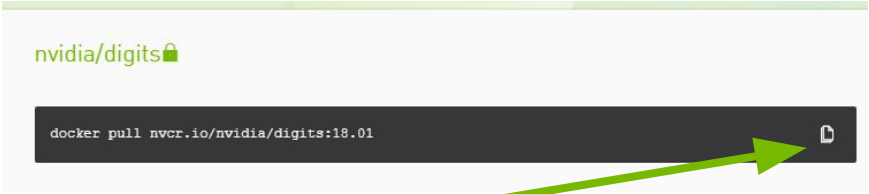
TAG	SIZE	USER	LAST MODIFIED	PULL
18.01	1.99 GB		January 23, 2018	
17.12	1.99 GB		December 3, 2017	
17.11	2.07 GB		November 16, 2017	
17.10	2.33 GB		October 24, 2017	

NGC WebUI

Docker pull shortcut

Shortcut to the latest at the top

- Shows full image name (`nvcr.io/nvidia/digits:18.01`)
- Icon to copy to clipboard
  - Same as in image details

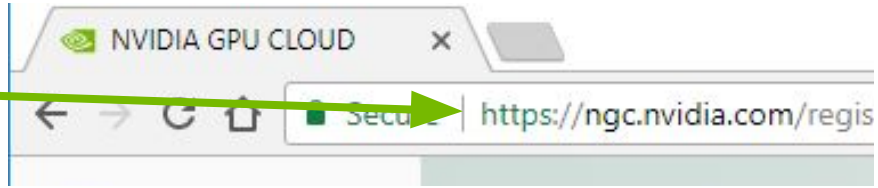


NGC vs. NVCR

Why are there two FQDNs?

ngc.nvidia.com

- NGC = NVIDIA GPU Cloud
- Used for administrative tasks
  - (a.k.a. The WebUI)

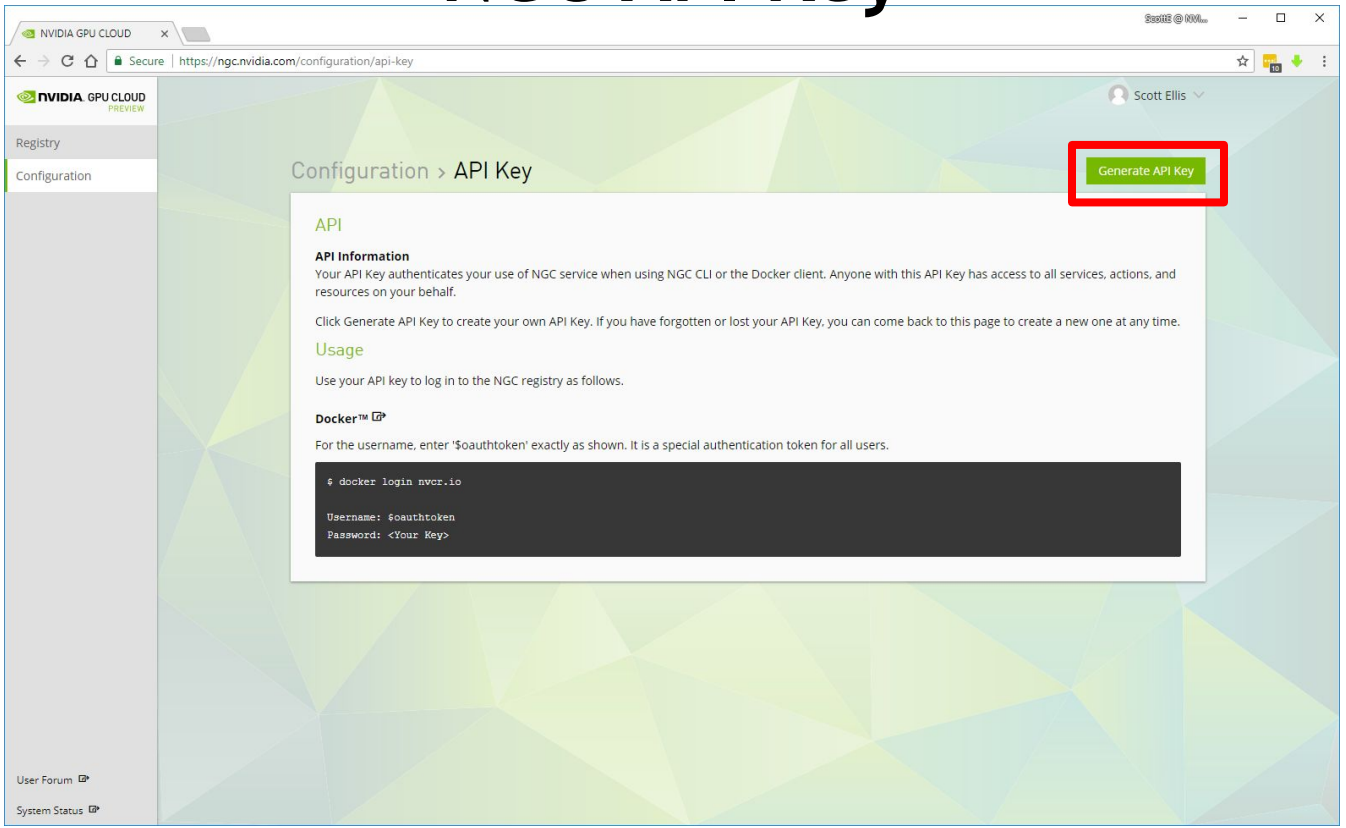



```
docker pull nvcr.io/nvidia/digits:18.01
```

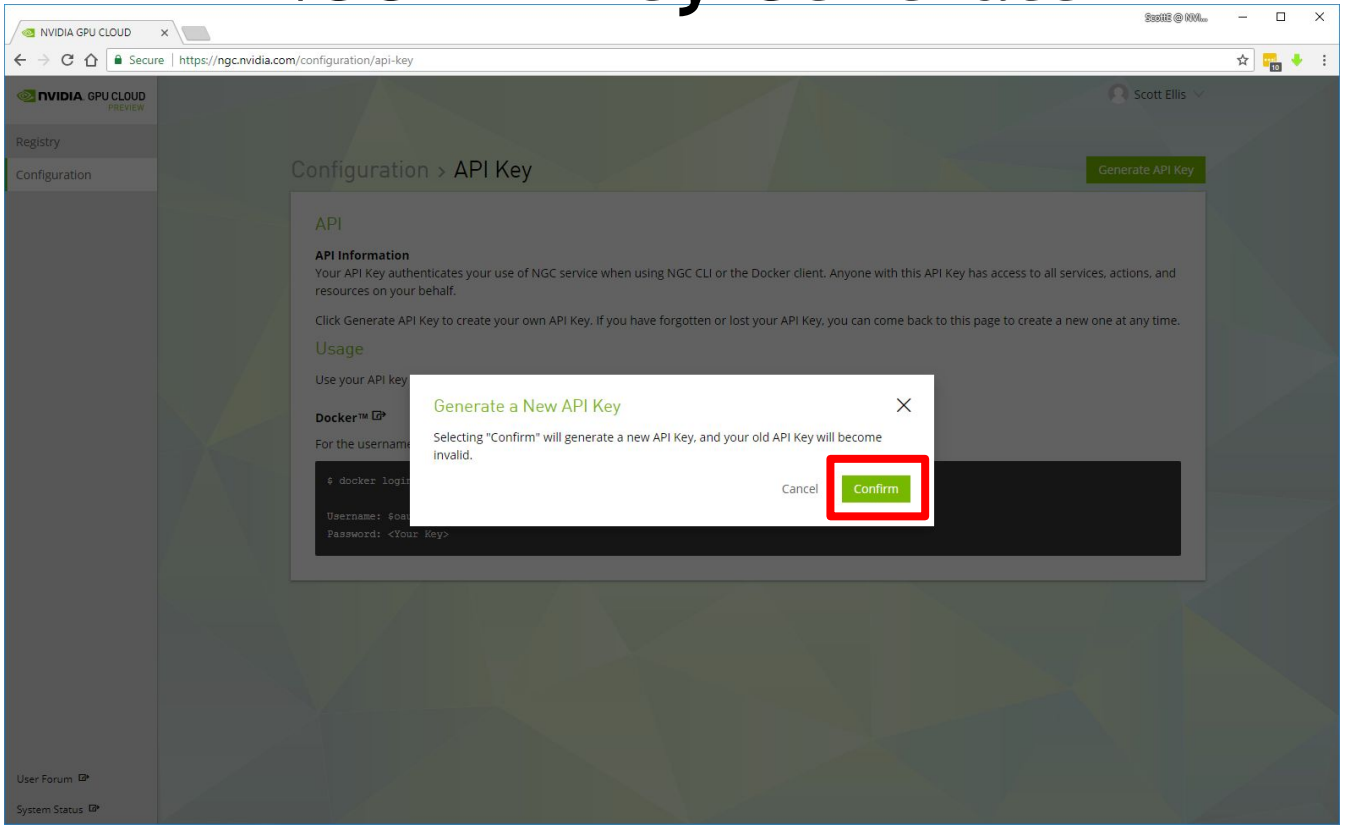
nvcr.io

- NVIDIA Container Repository
- Used for Docker tasks

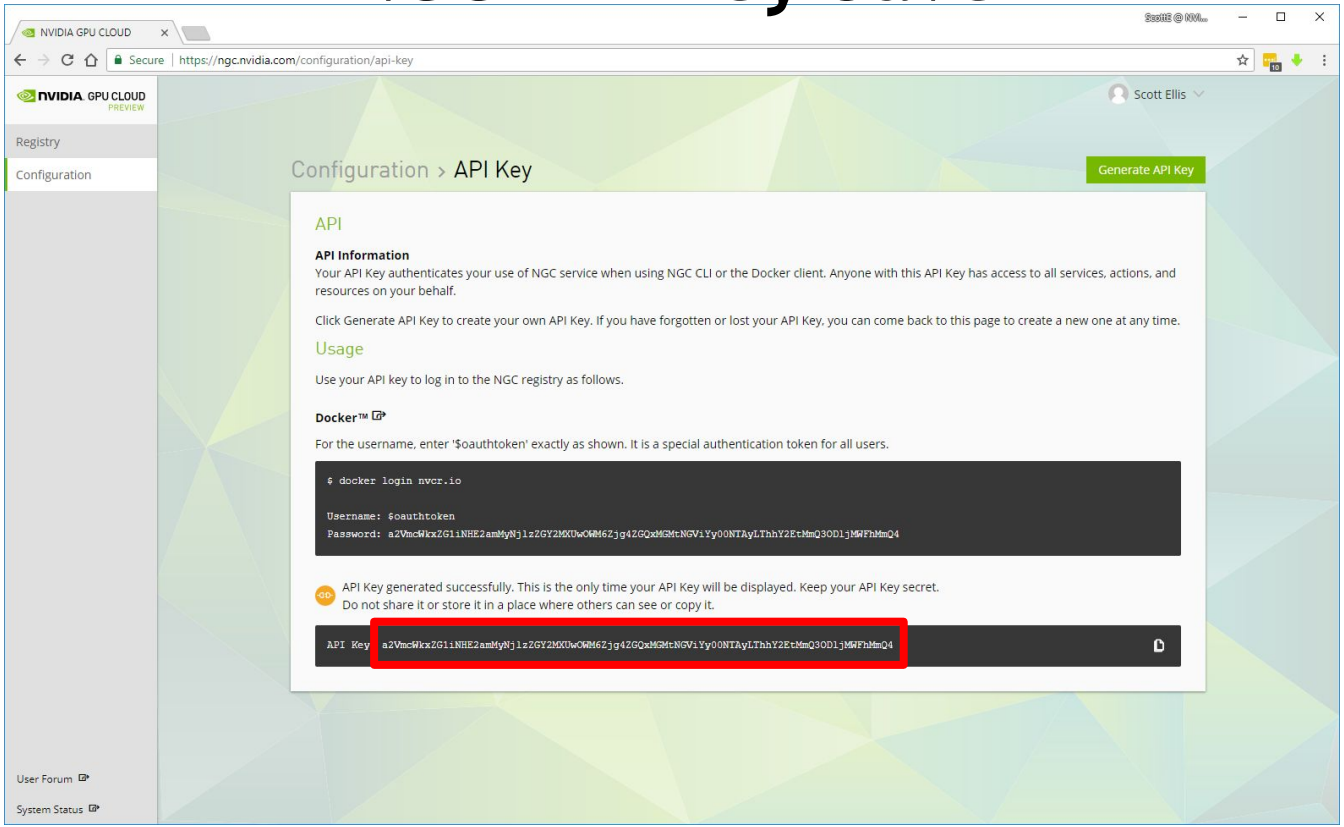
NGC API Key


NVIDIA GPU CLOUD
Secure | https://ngc.nvidia.com/configuration/api-key
NVIDIA GPU CLOUD
PREVIEW
Registry
Configuration
User Forum
System Status
Scott Ellis
Configuration › API Key
Generate API Key
API
API Information
Your API Key authenticates your use of NGC service when using NGC CLI or the Docker client. Anyone with this API Key has access to all services, actions, and resources on your behalf.
Click Generate API Key to create your own API Key. If you have forgotten or lost your API Key, you can come back to this page to create a new one at any time.
Usage
Use your API key to log in to the NGC registry as follows.
Docker™
For the username, enter '$oauthtoken' exactly as shown. It is a special authentication token for all users.
$ docker login nvcr.io
Username: $oauthtoken
Password: <Your Key>

NGC API Key Generate



NGC API Key Save



AWS Account Setup

Execution Environment

How does Amazon figure into this?

AWS is where we'll run NGC containers

...could also be your NVIDIA Titan-based workstation

...could also be your DGX server or workstation

...could also be almost anywhere with a GPU and docker

AWS

Getting Started

Before launching an Amazon instance (for the first time) we need to:

- Secure login access to our instance with an SSH key
- Secure network access with a set of connection rules

AWS

Getting Started

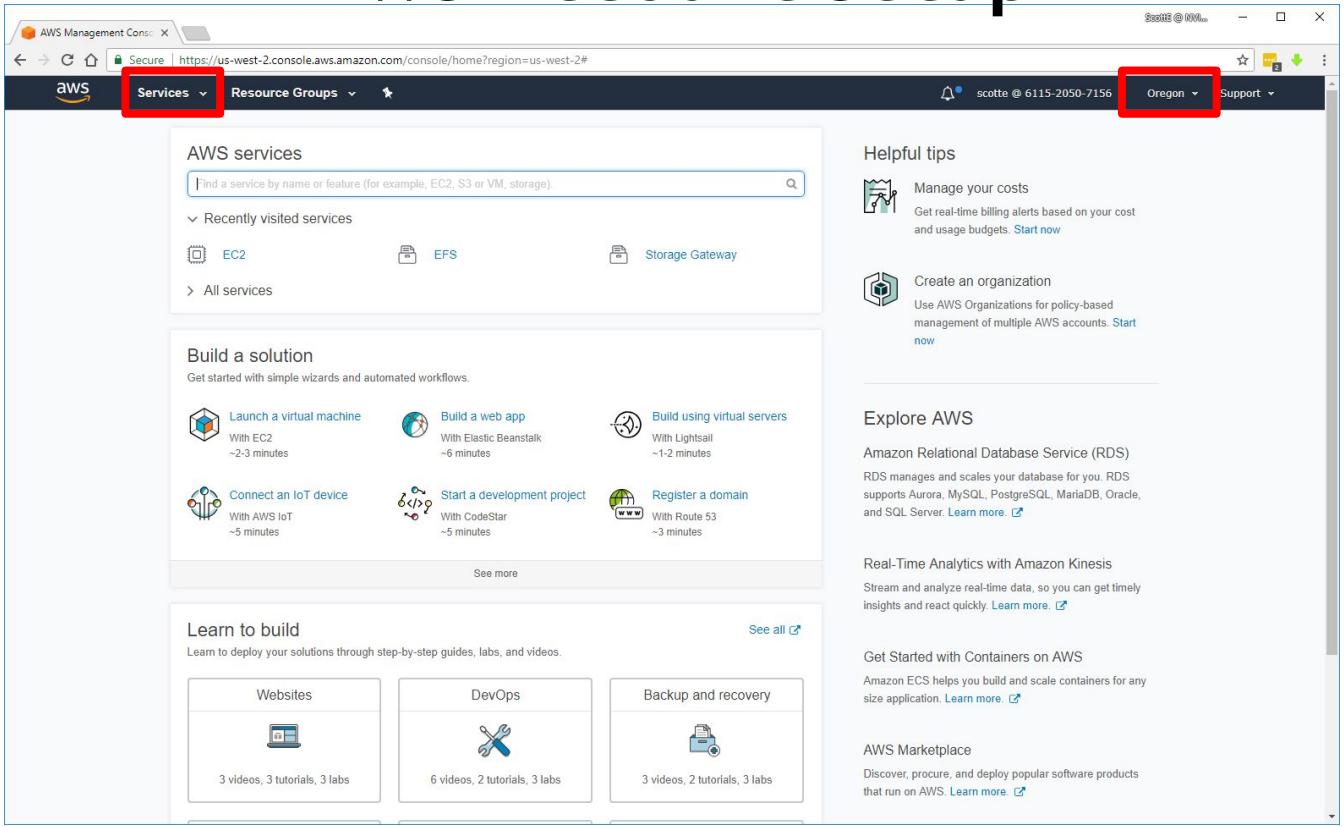
Our challenge:

- Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/
- Create Key Pair
- Create Security Group

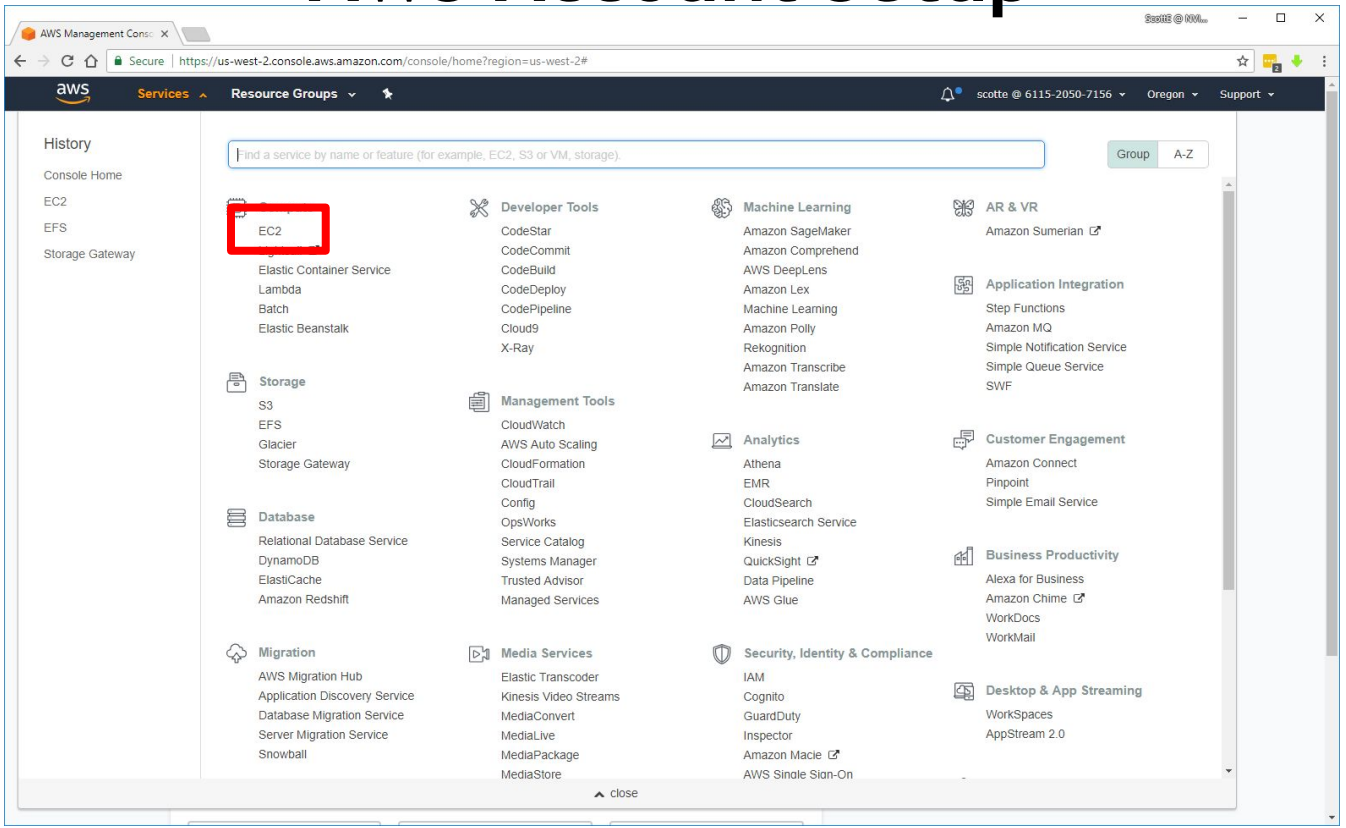
If AWS is managed for you, this might not be visible to you

(We'll use the WebUI, but there are hundreds of other ways to do this!)

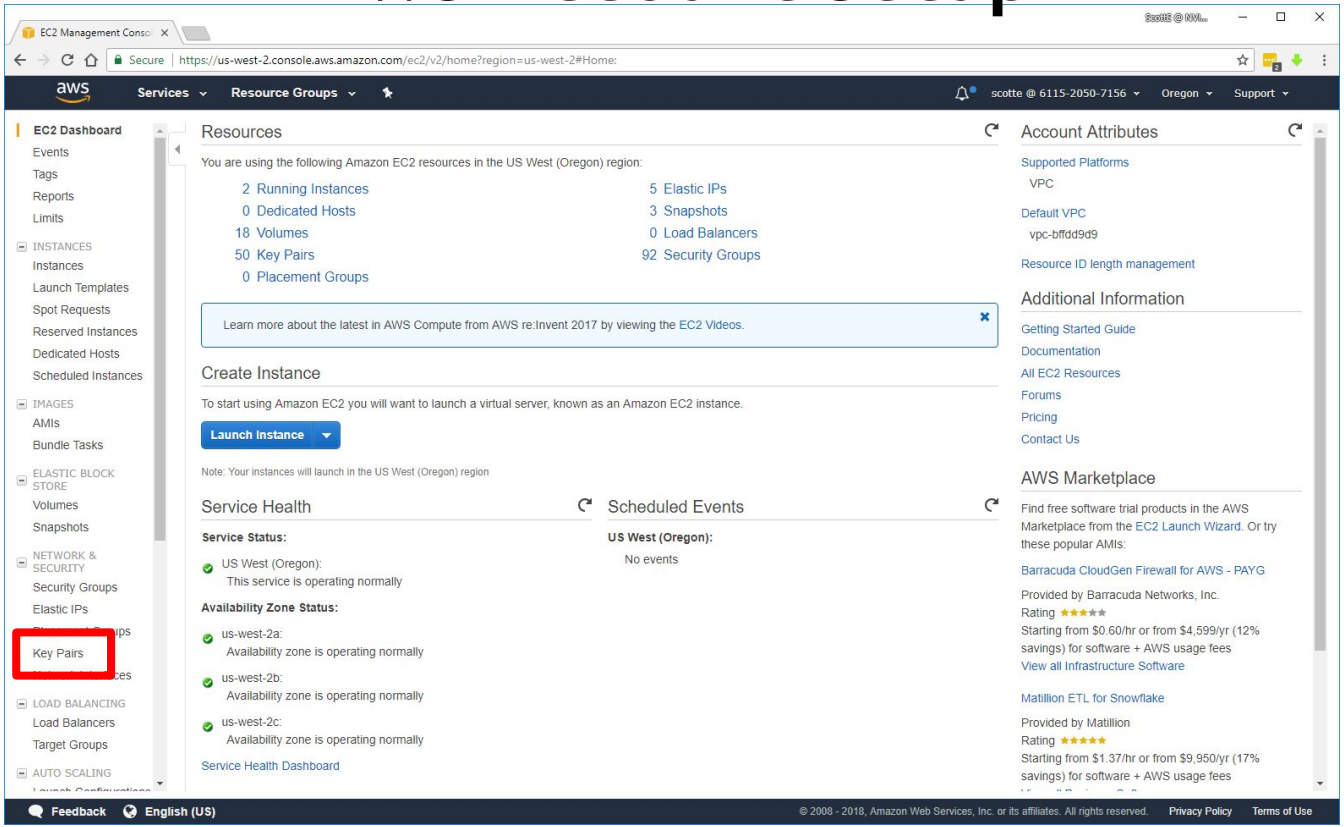
AWS Account Setup



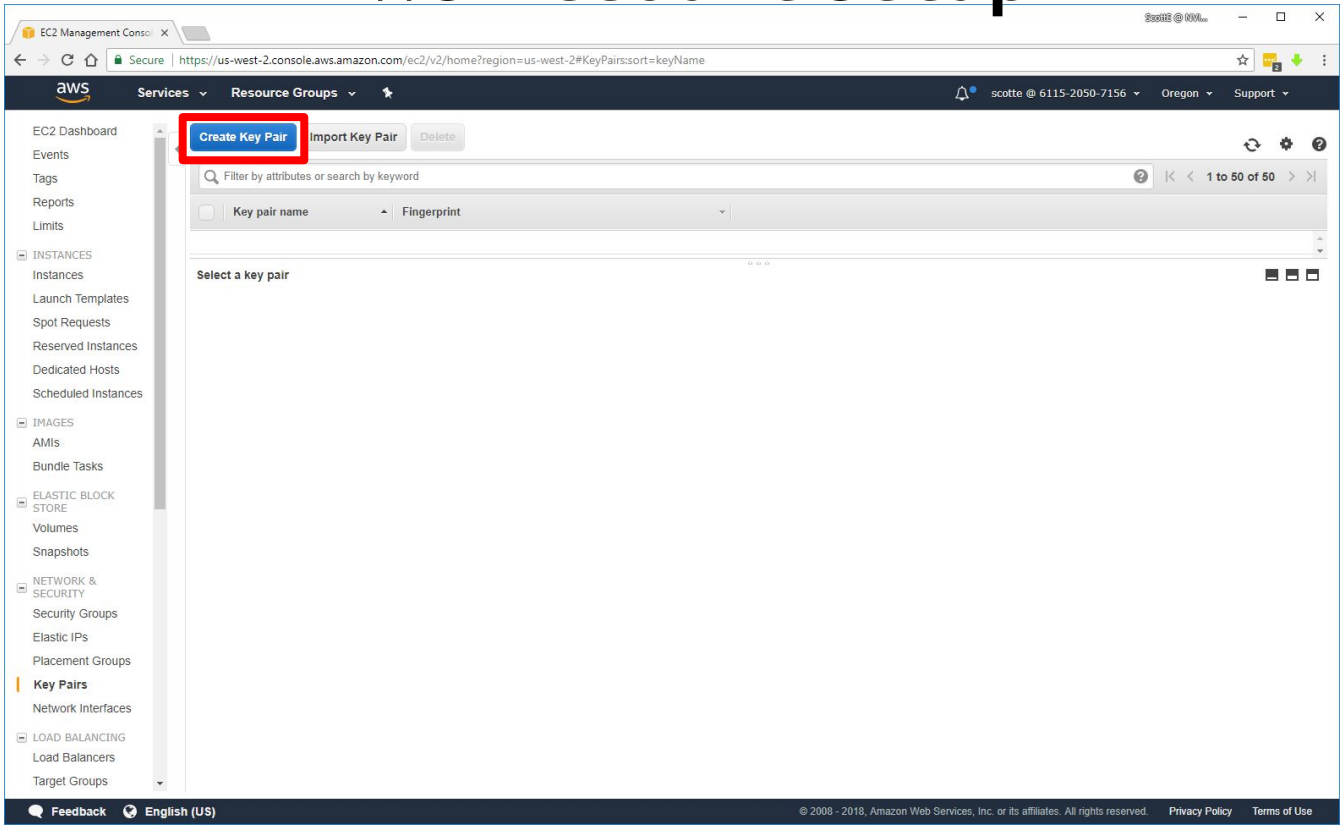
AWS Account Setup



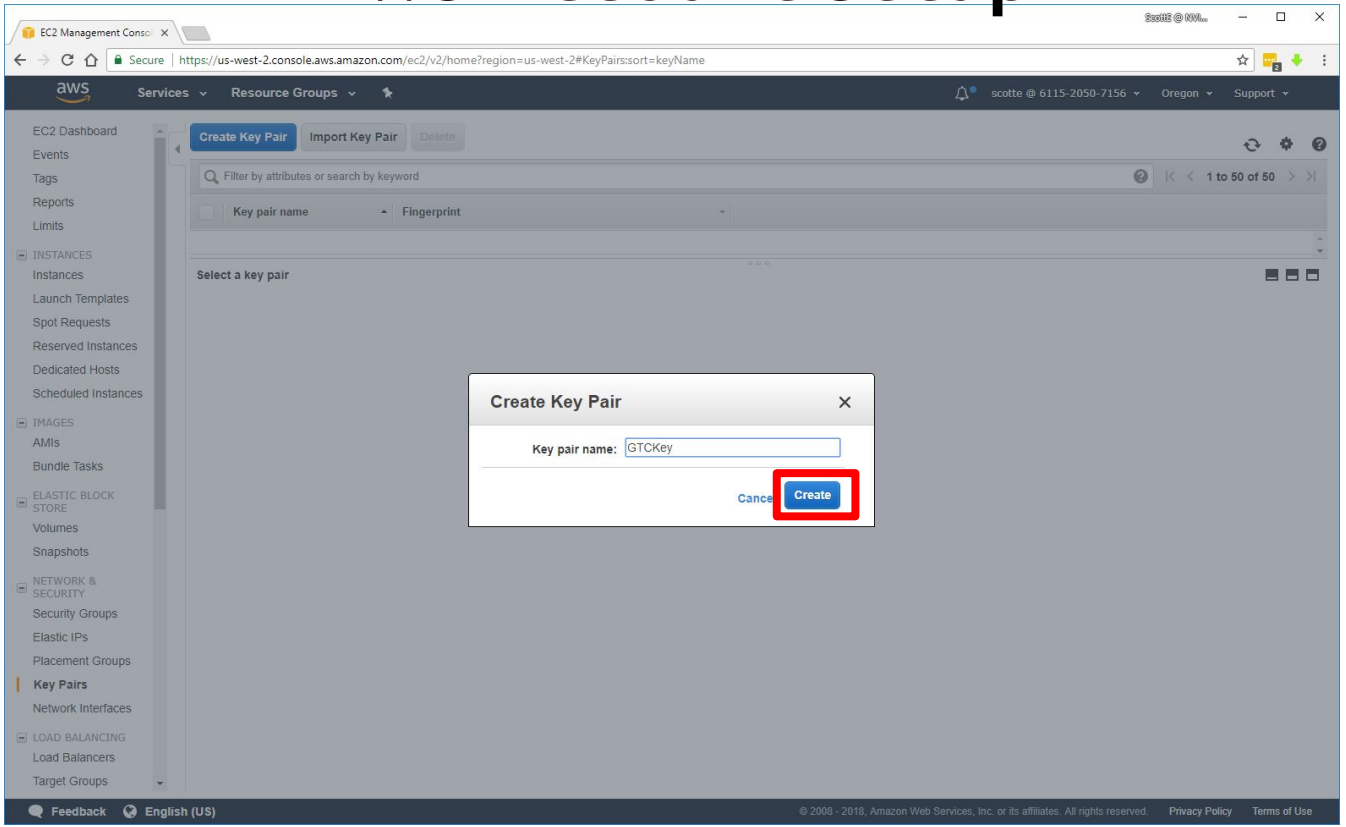
AWS Account Setup



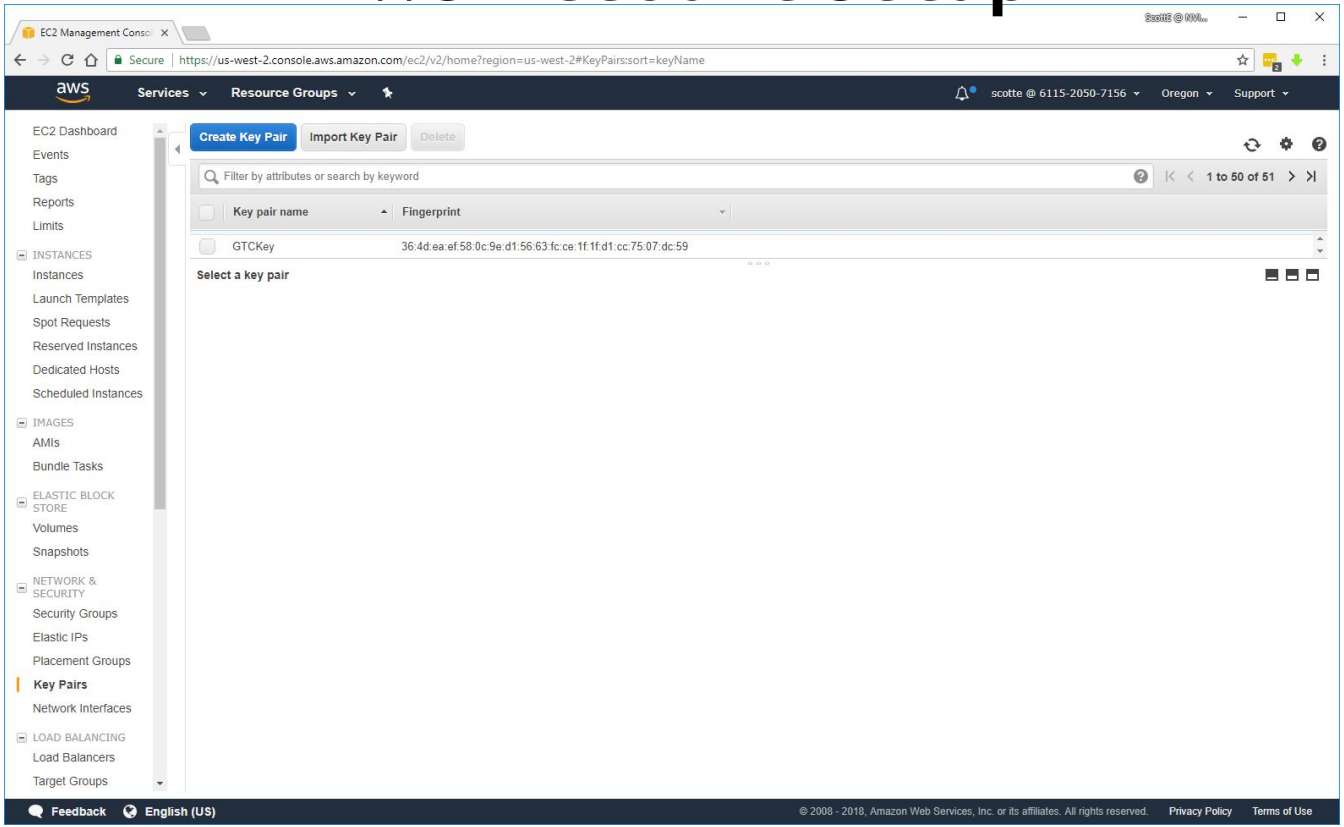
AWS Account Setup



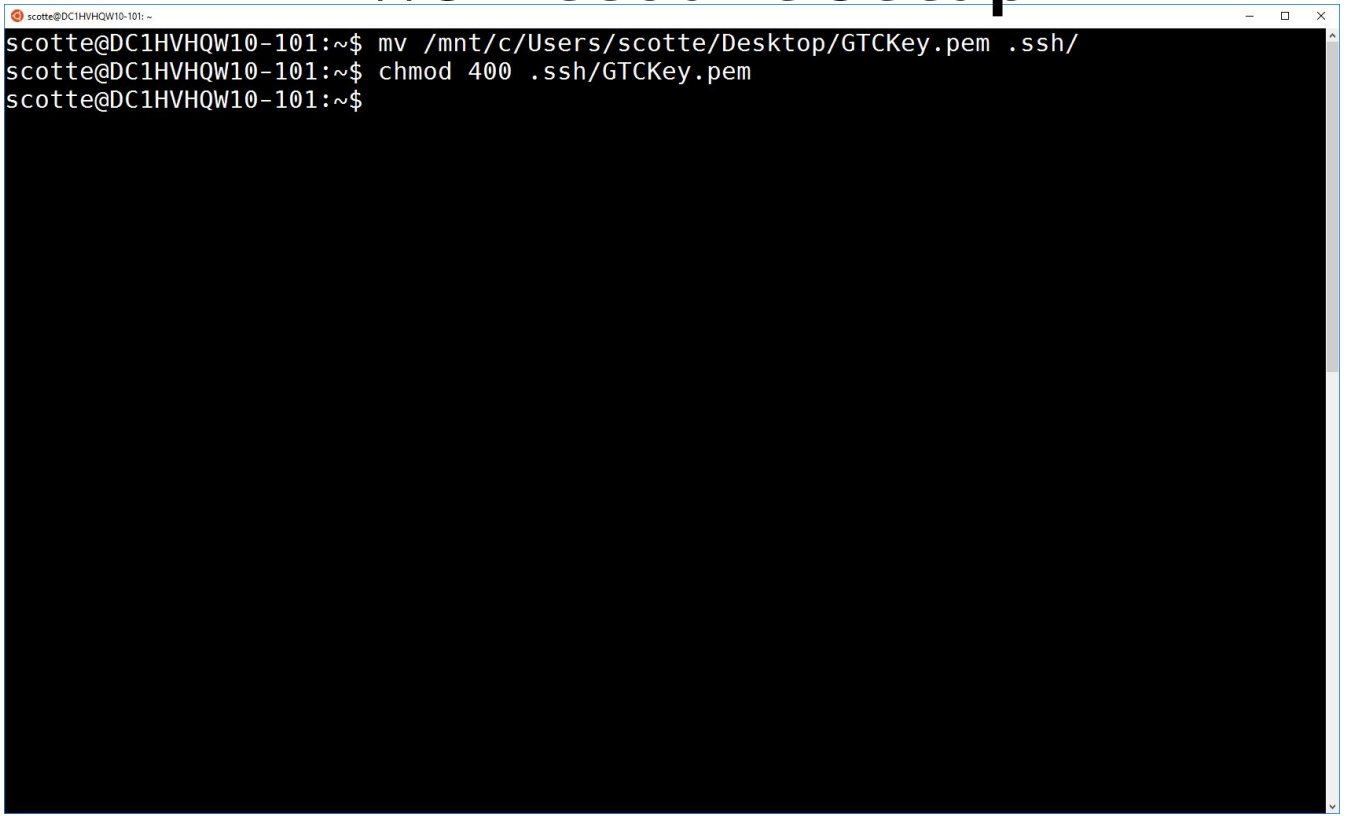
AWS Account Setup



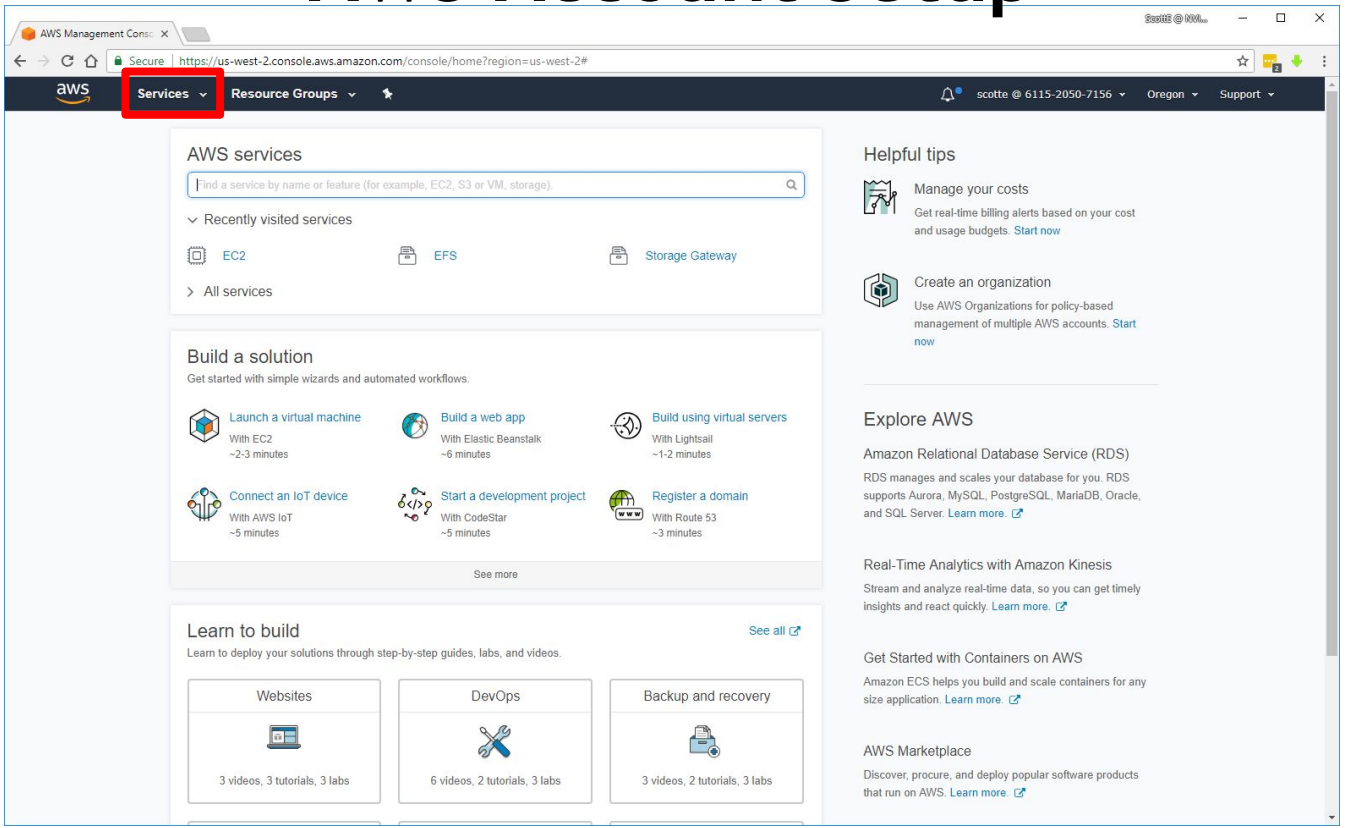
AWS Account Setup



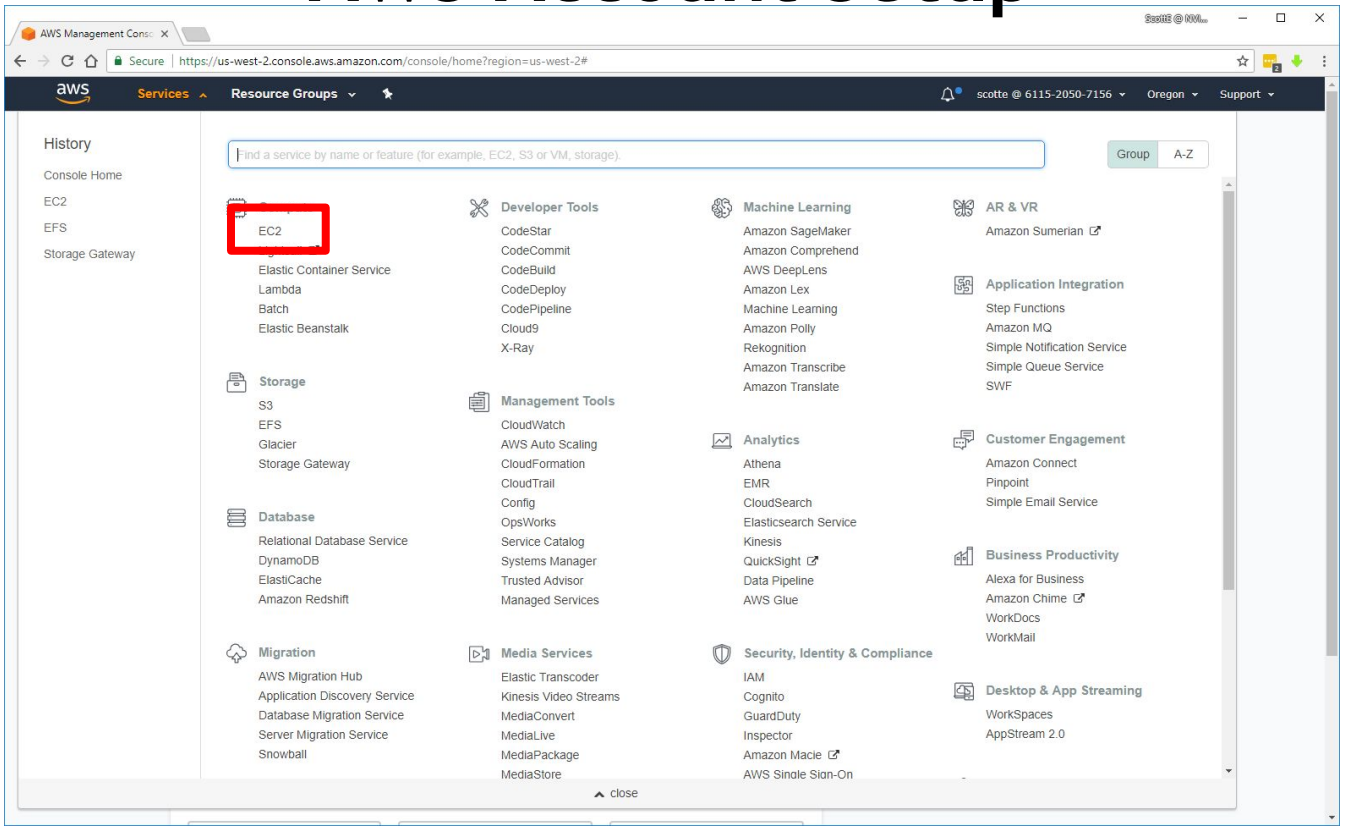
AWS Account Setup



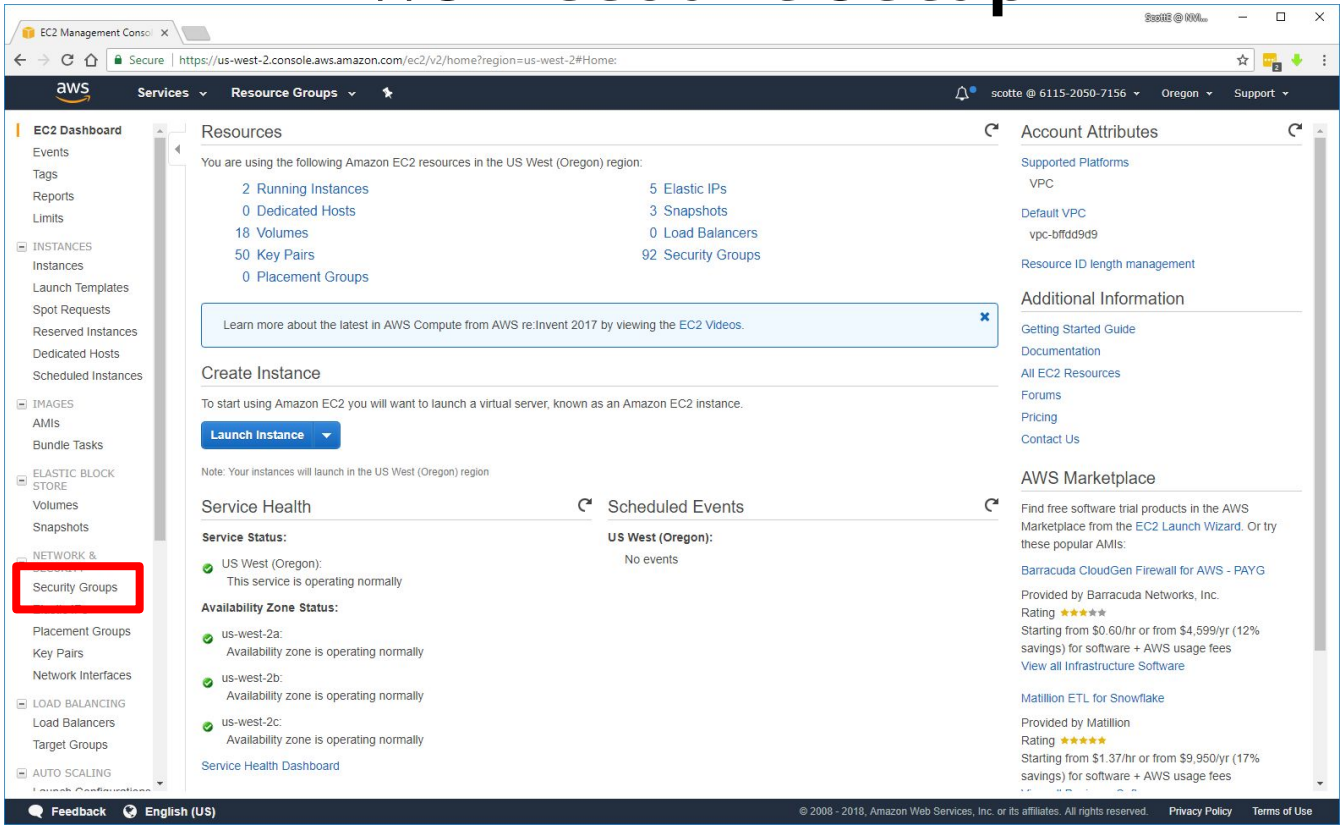
AWS Account Setup



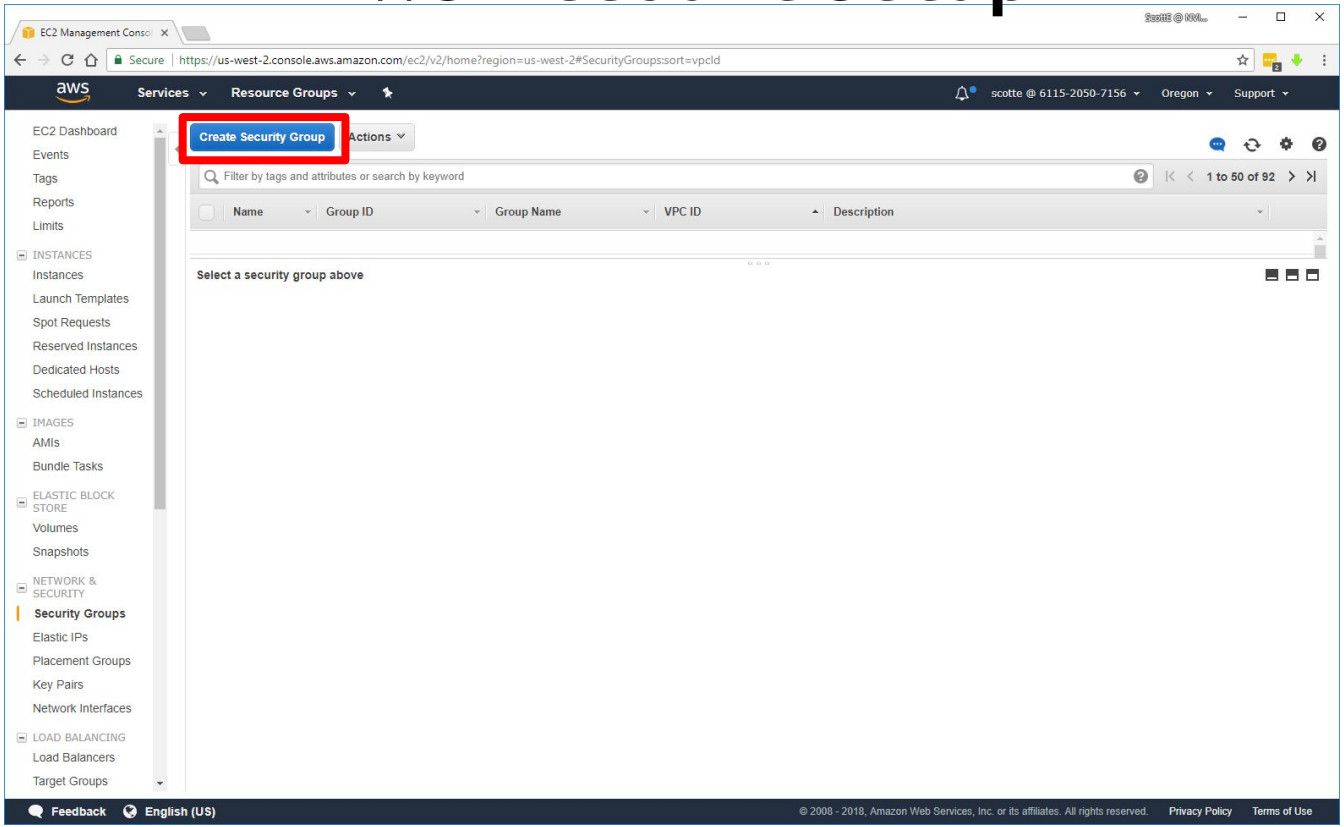
AWS Account Setup



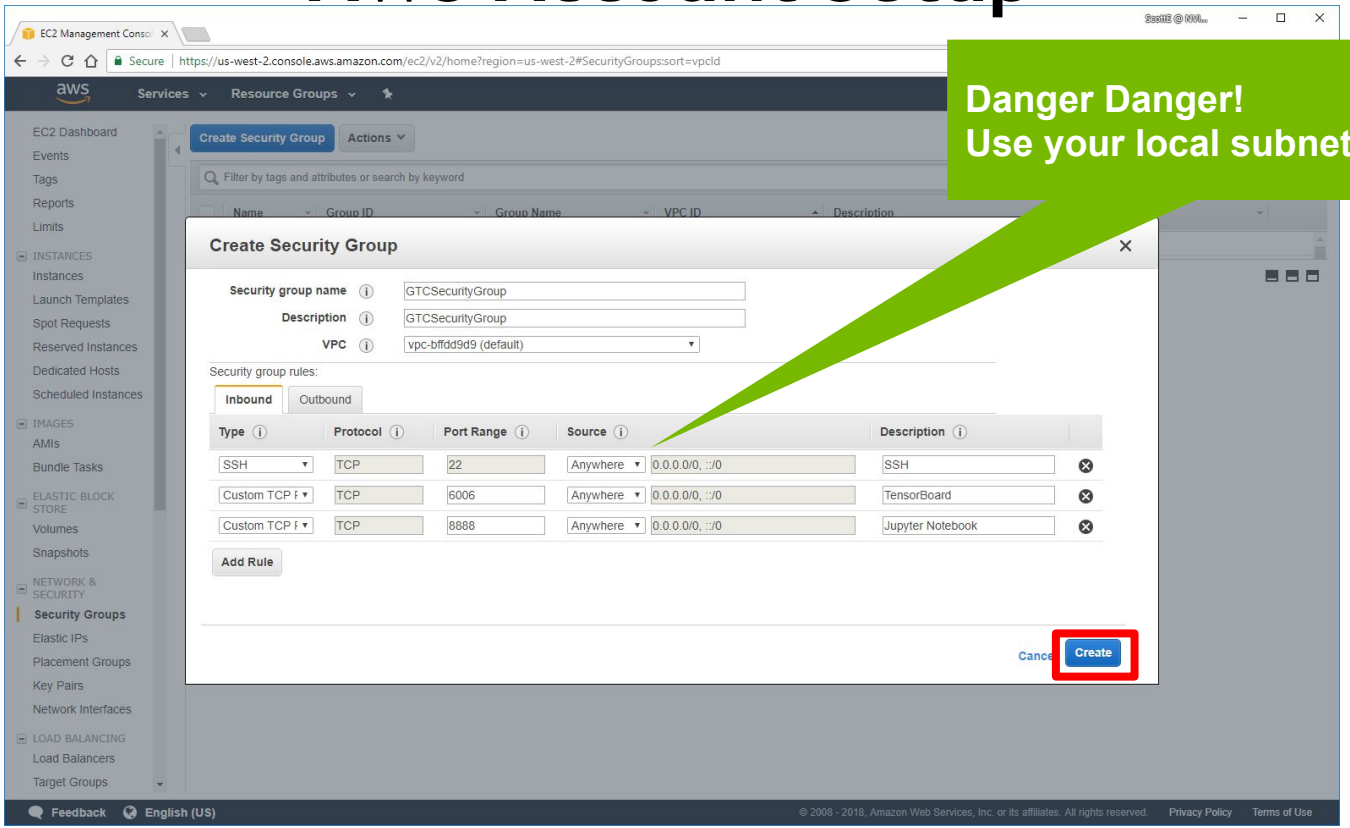
AWS Account Setup



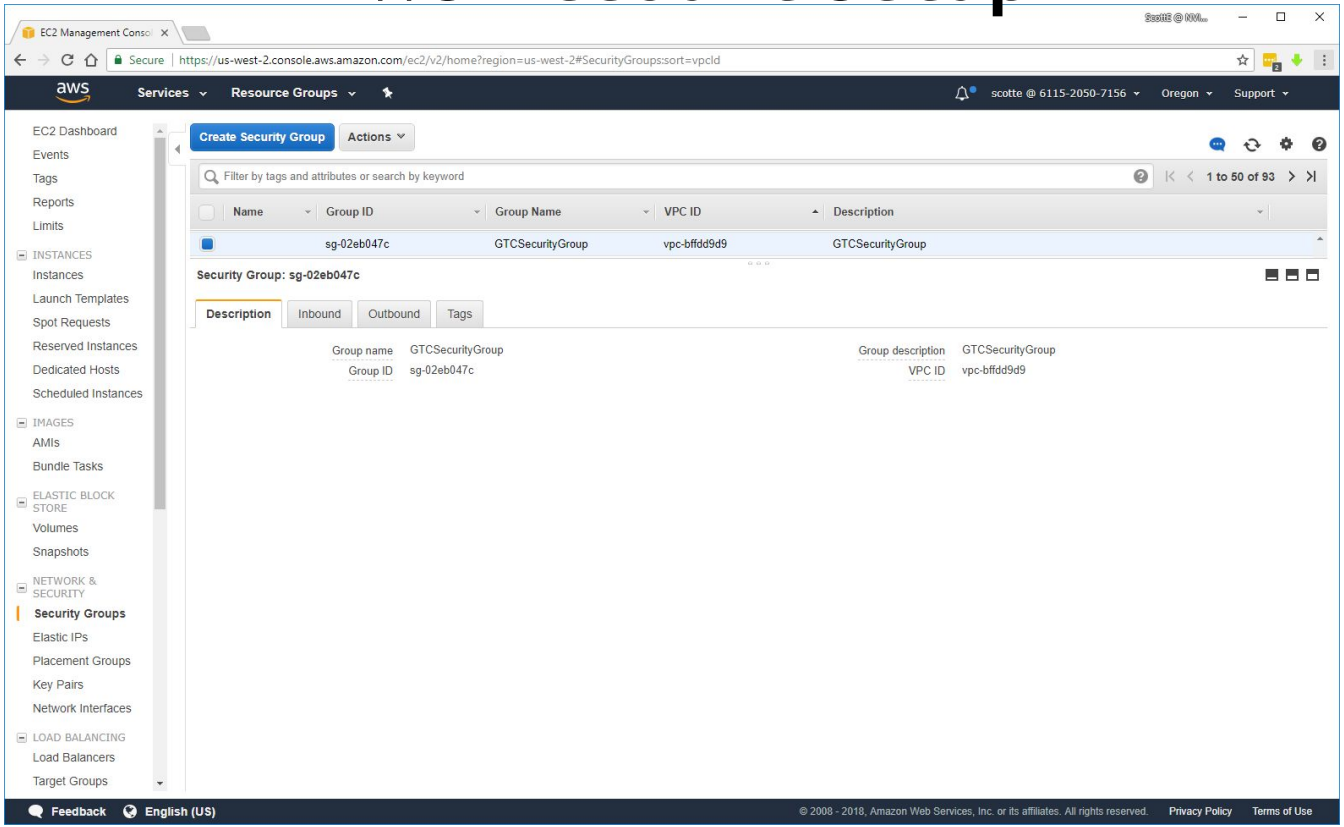
AWS Account Setup



AWS Account Setup



AWS Account Setup



AWS Launching an Instance

AWS AMI and Instances

What is an AMI? What is an Instance?

- Amazon Machine Instance (AMI)
  - Software to run in a Virtual Machine on AWS
  - Runs on an AWS EC2 Instance
- Instance
  - Hardware for running an AMI
  - Specifies CPU, GPU, RAM, Storage, Network, …

Instance Size	GPUs - Tesla V100	GPU Peer to Peer	GPU Memory (GB)	vCPUs	Memory (GB)	Network Bandwidth	EBS Bandwidth
p3.2xlarge	1	N/A	16	8	61	Up to 10 Gbps	1.5 Gbps

AWS AMI and Instances

What does NGC provide?

NGC Containers are best on...

- NVIDIA Volta Deep Learning AMI
  - Machine image with docker configured for GPUs
  - Automatic login to NGC on launch
  - Works with P3 instances (1x, 2x, or 8x Volta)

Details: http://docs.nvidia.com/ngc/ngc-ami-release-notes/

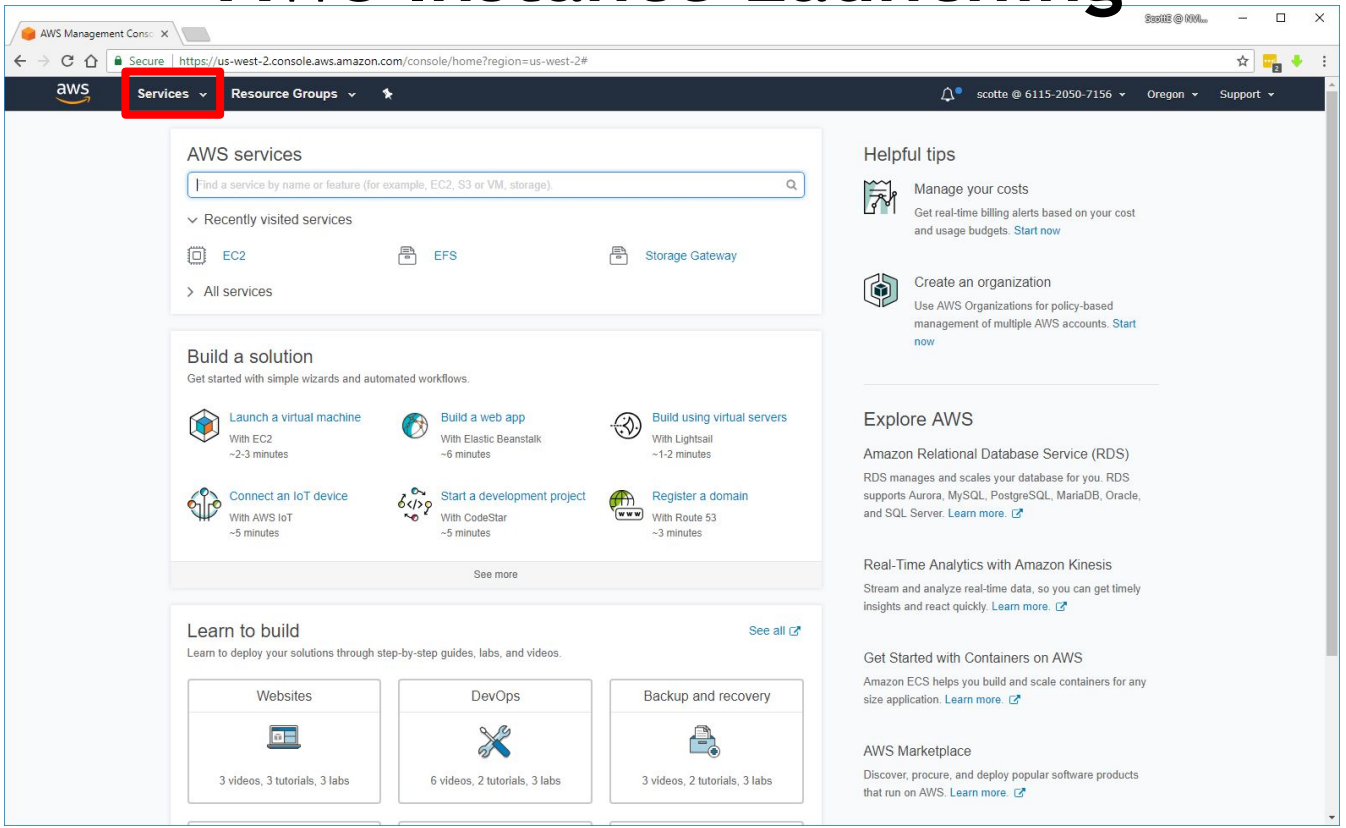
AWS AMI and Instances

Let’s do this

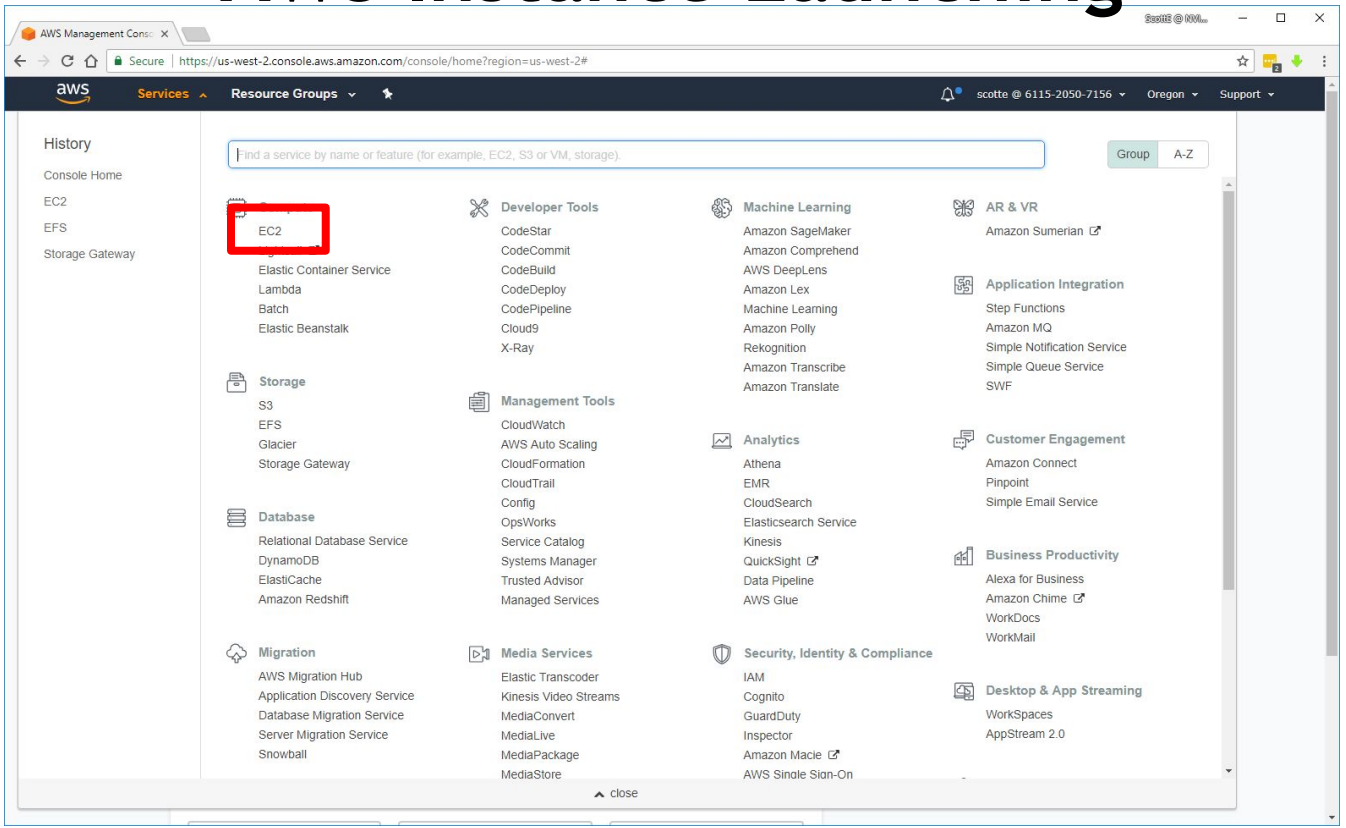
Our challenge:

- Launch an NVIDIA Volta Deep Learning AMI
- SSH into the instance
- Authenticate with NGC

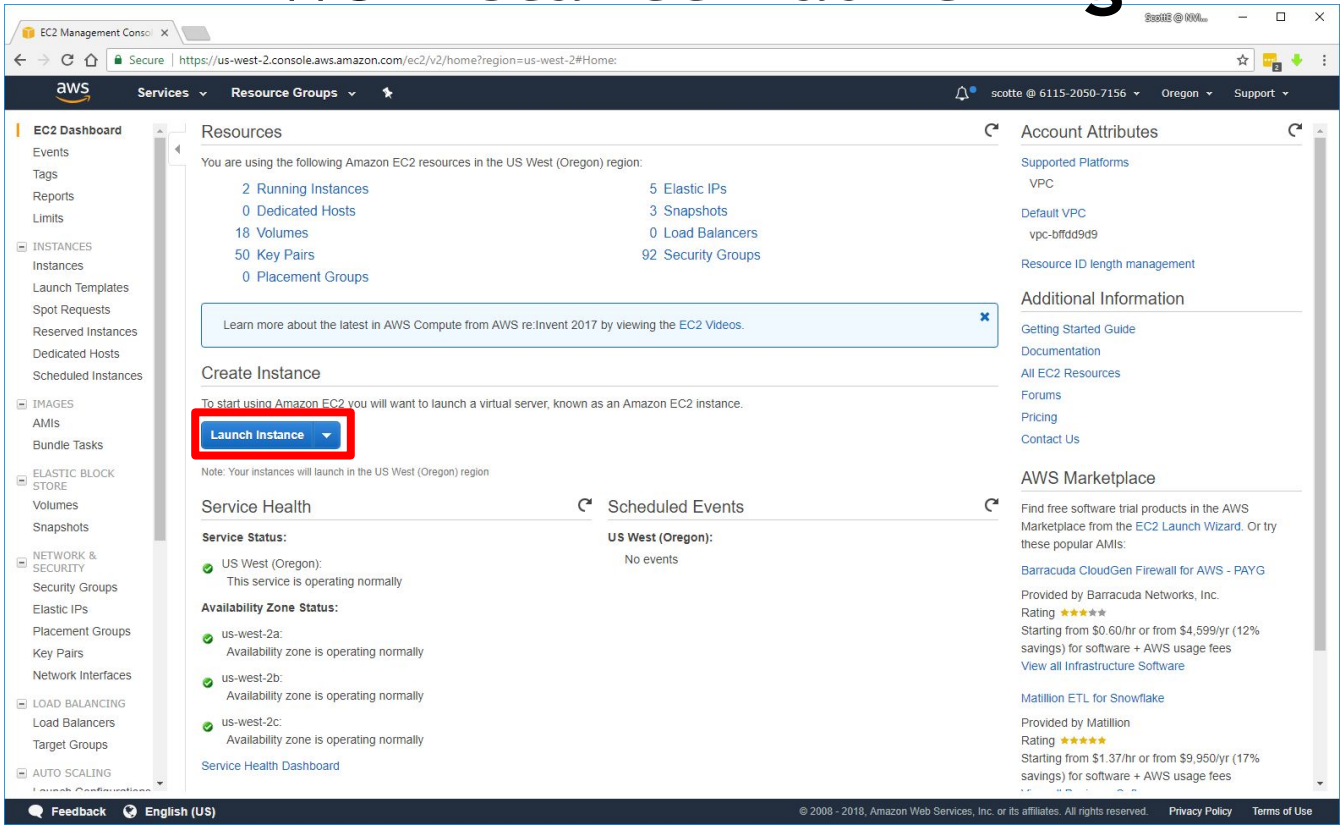
AWS Instance Launching



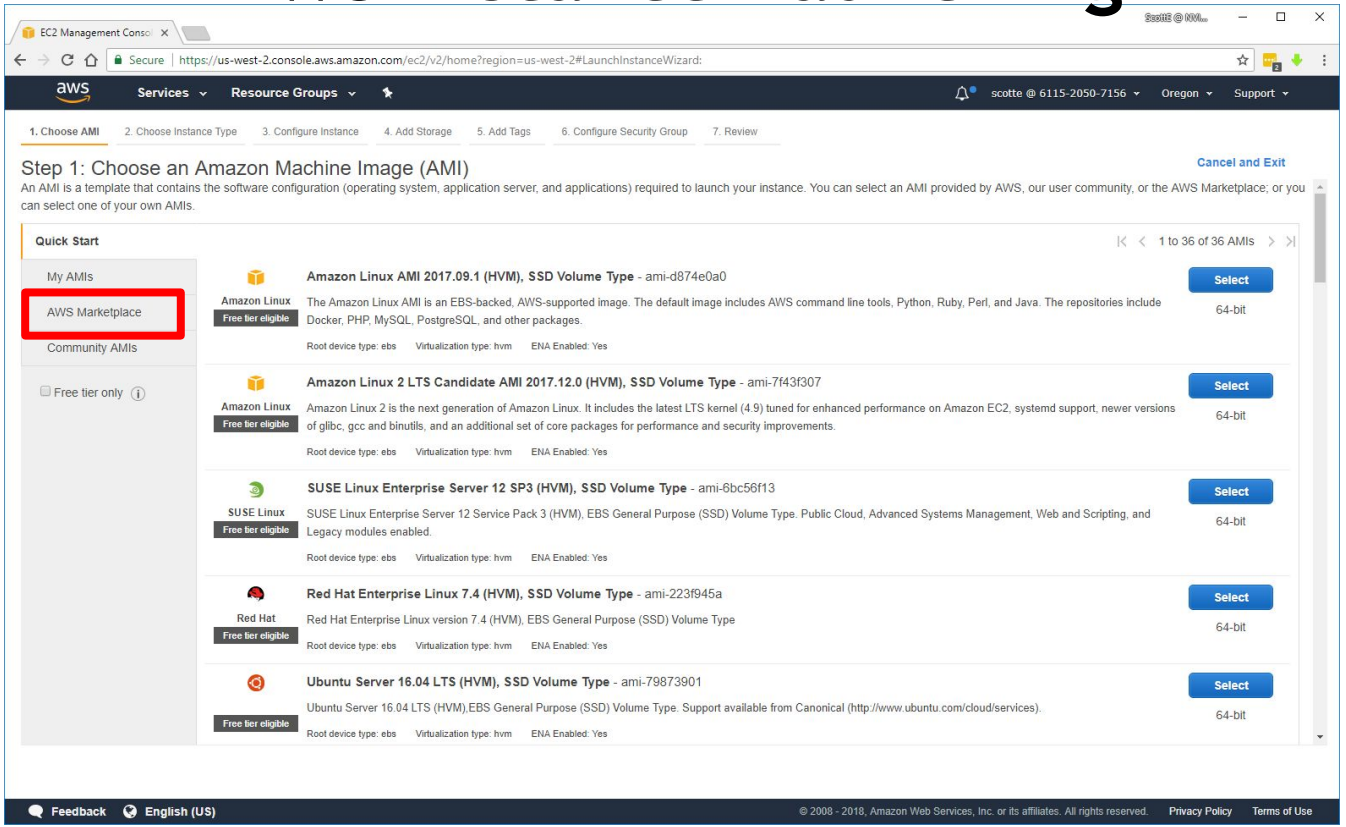
AWS Instance Launching



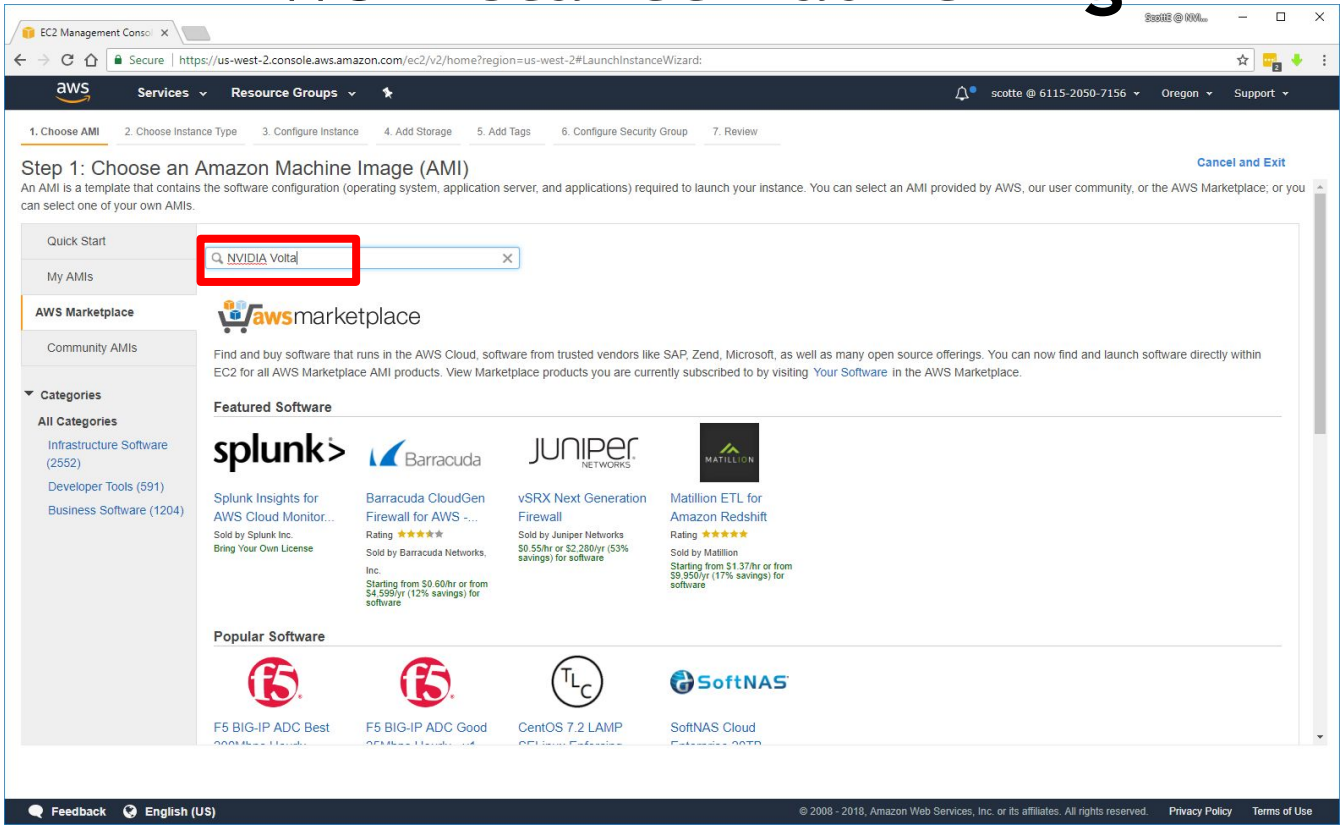
AWS Instance Launching



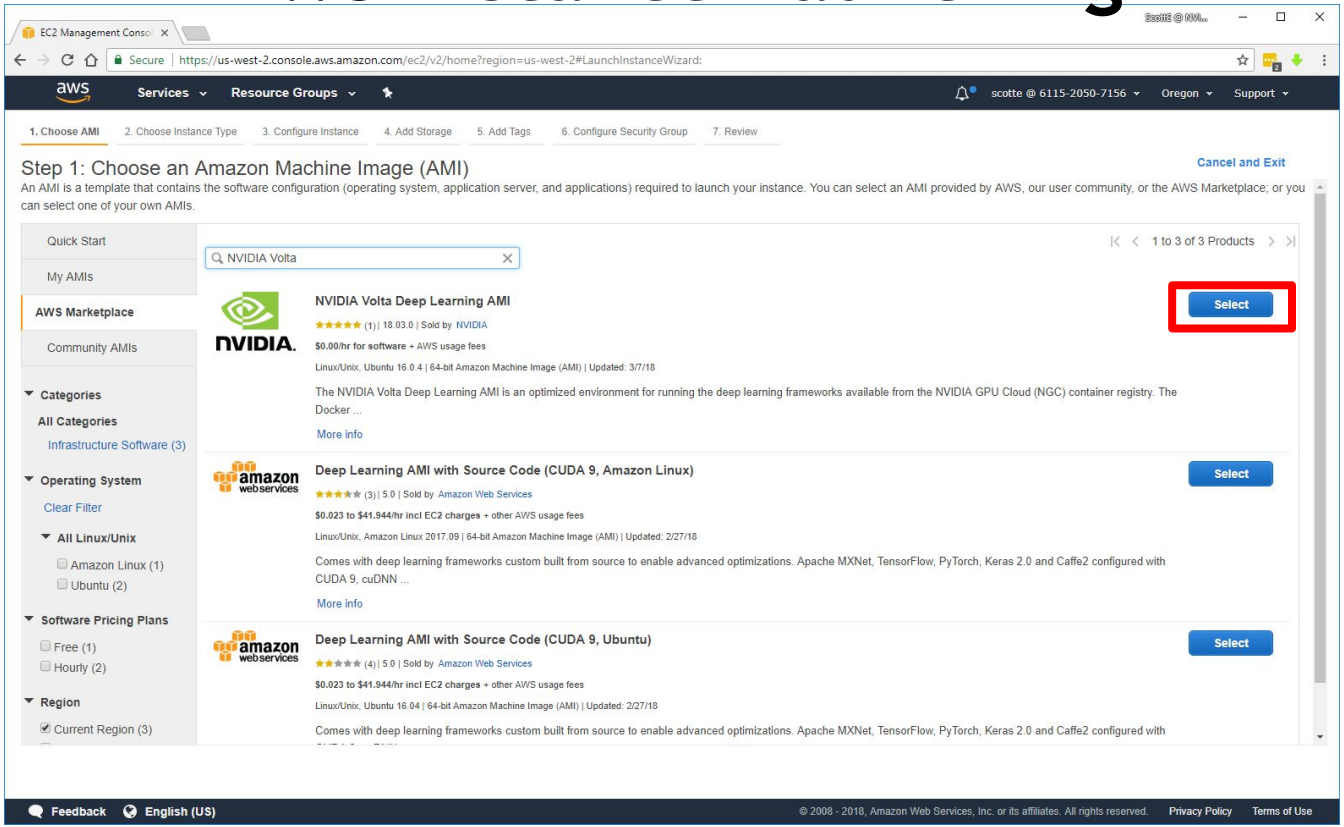
AWS Instance Launching



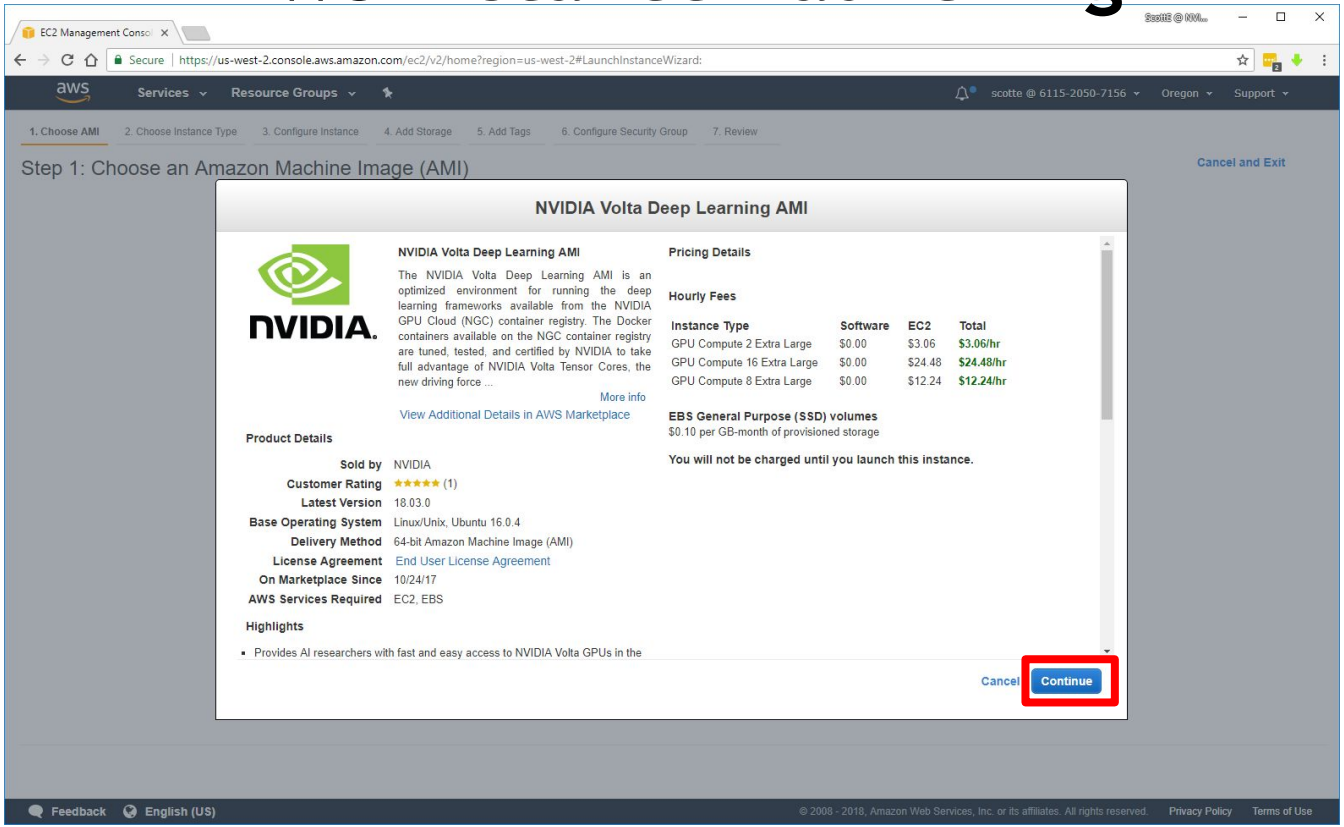
AWS Instance Launching



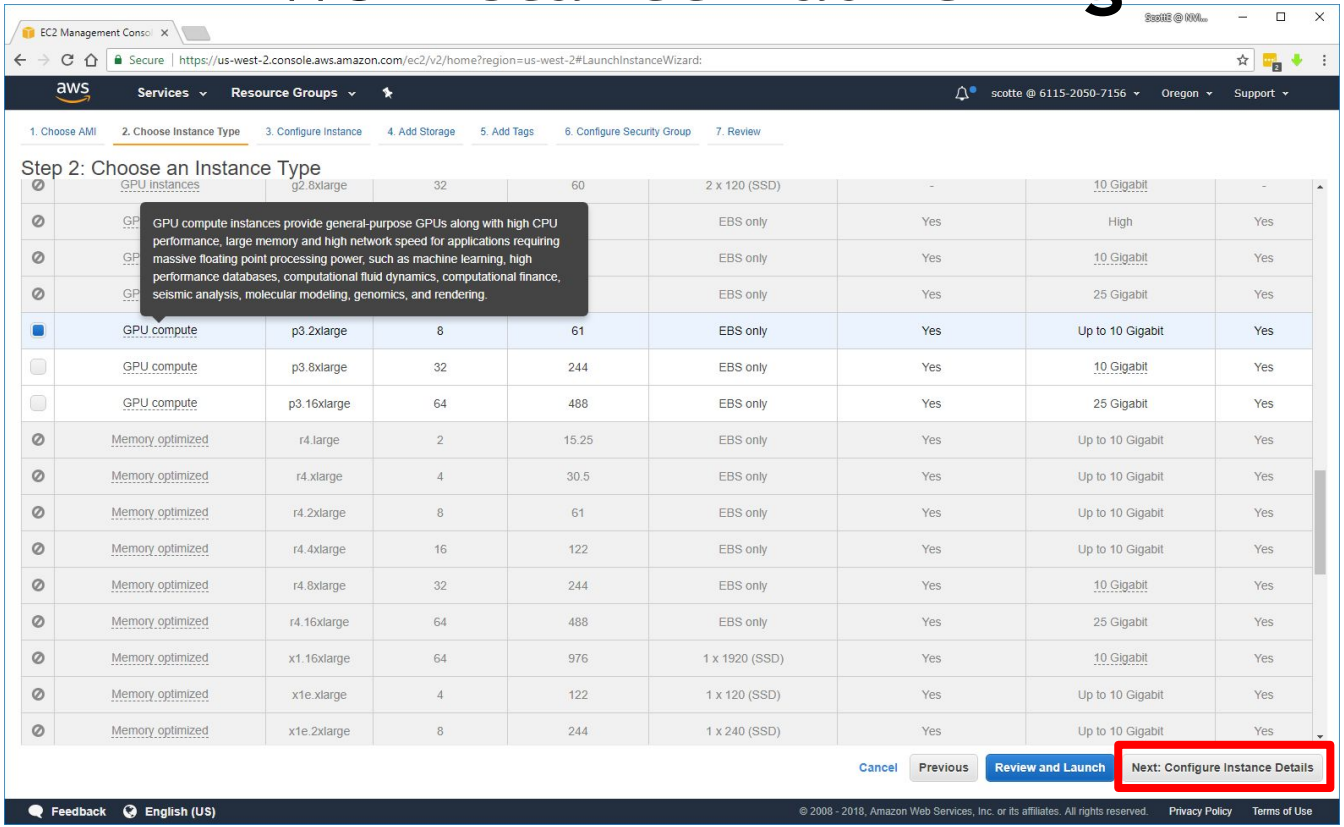
AWS Instance Launching



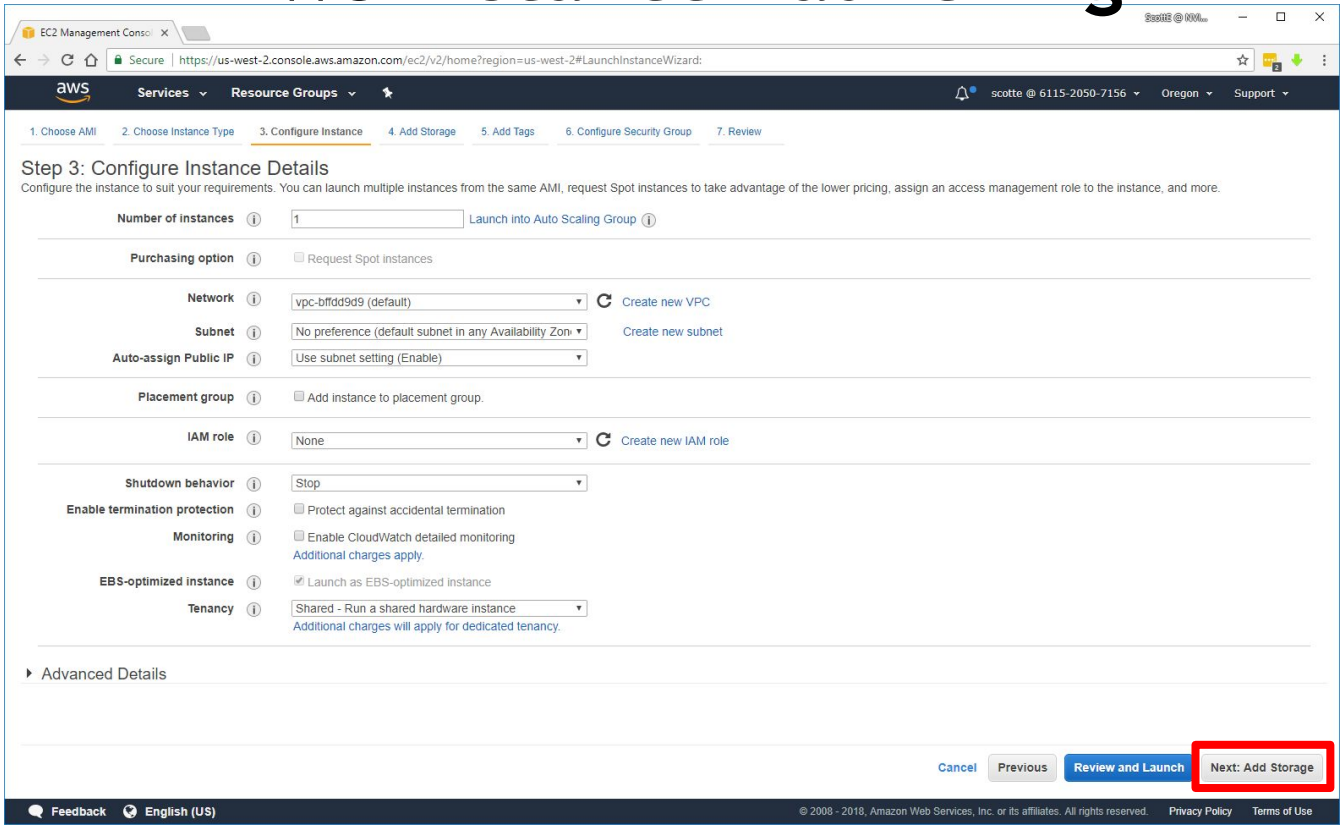
AWS Instance Launching



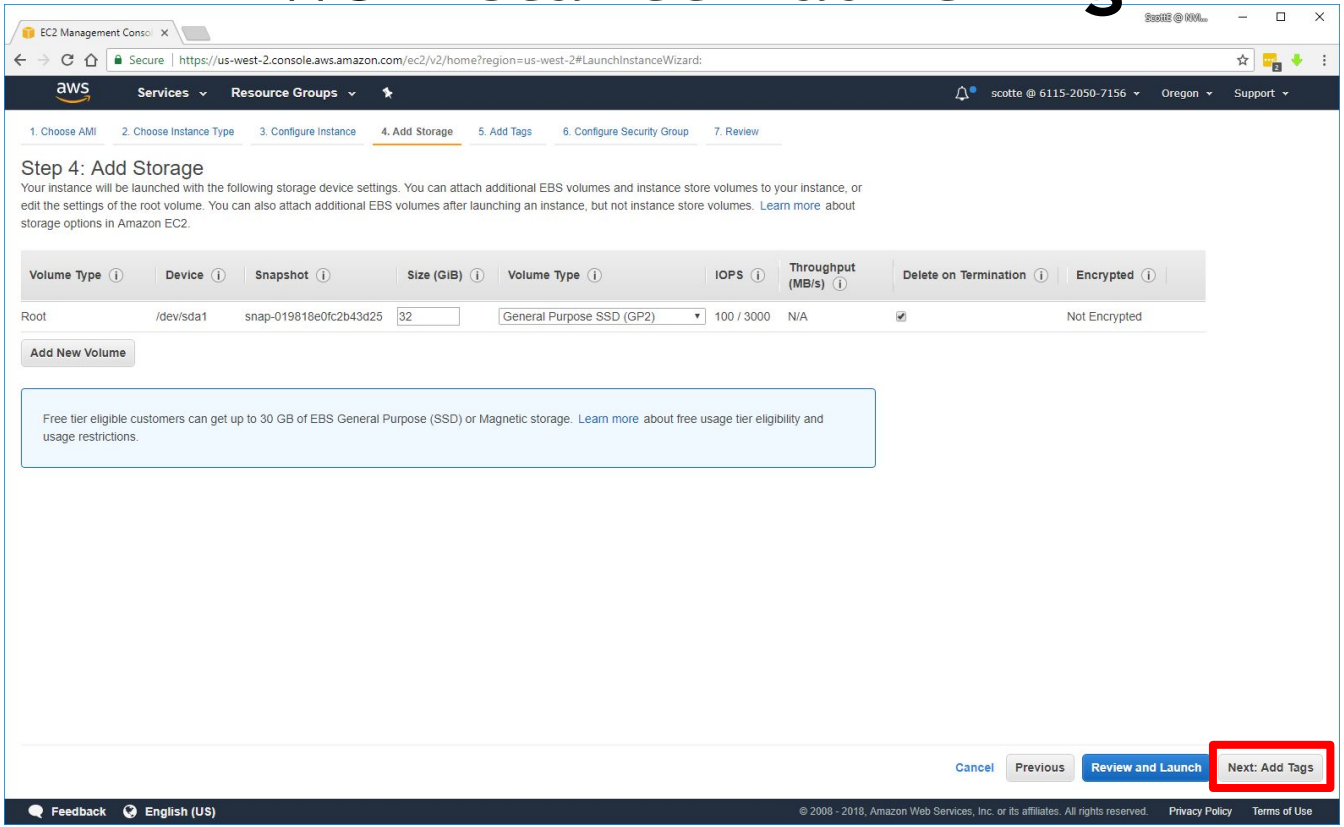
AWS Instance Launching



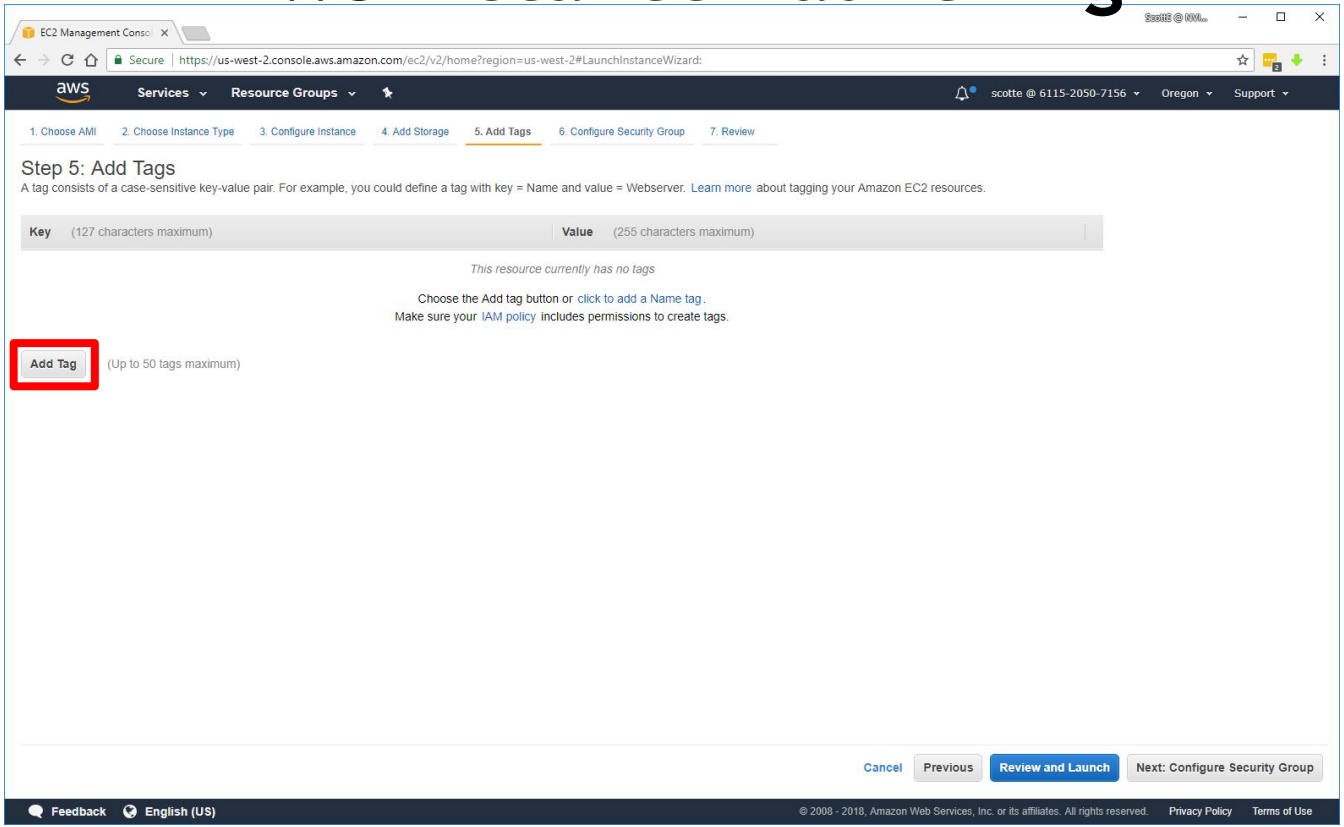
AWS Instance Launching



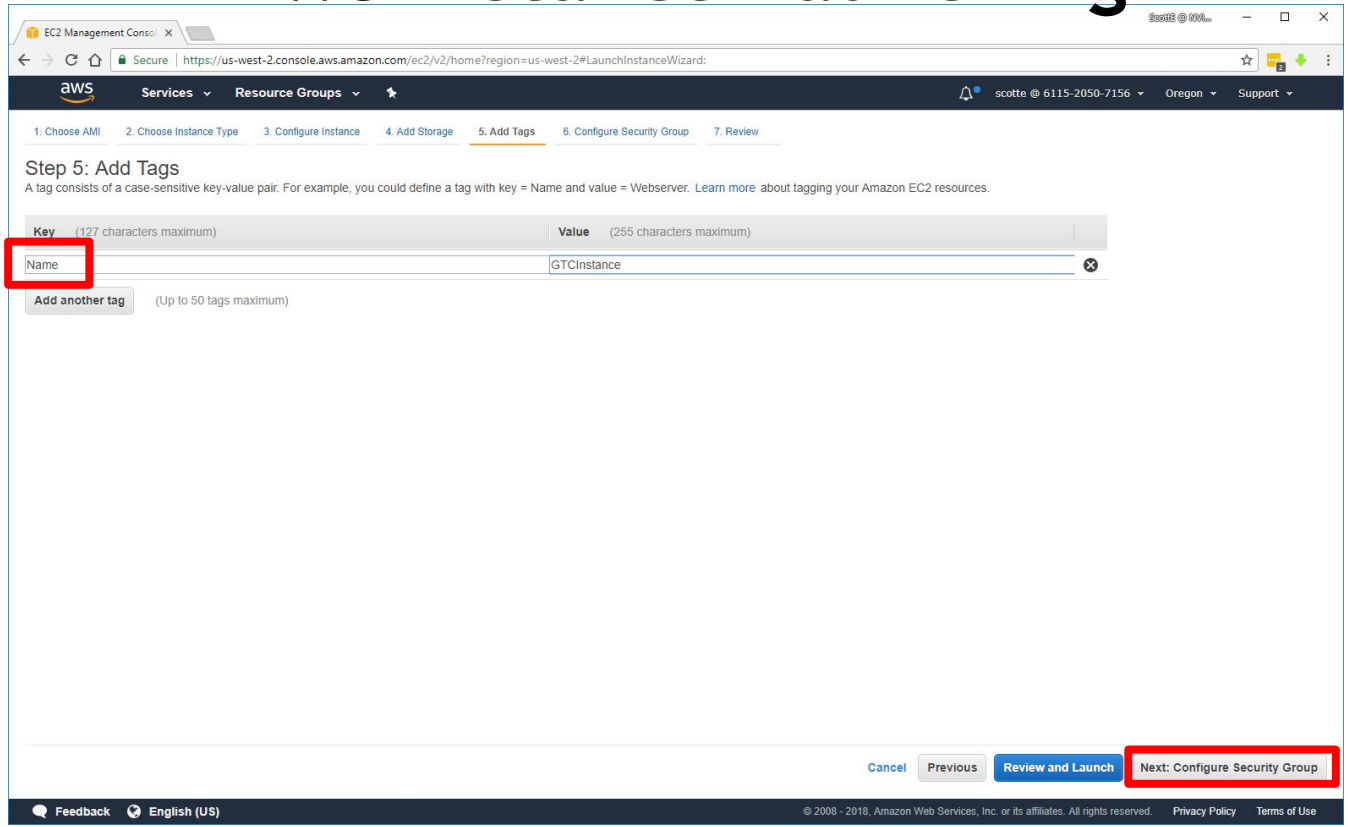
AWS Instance Launching



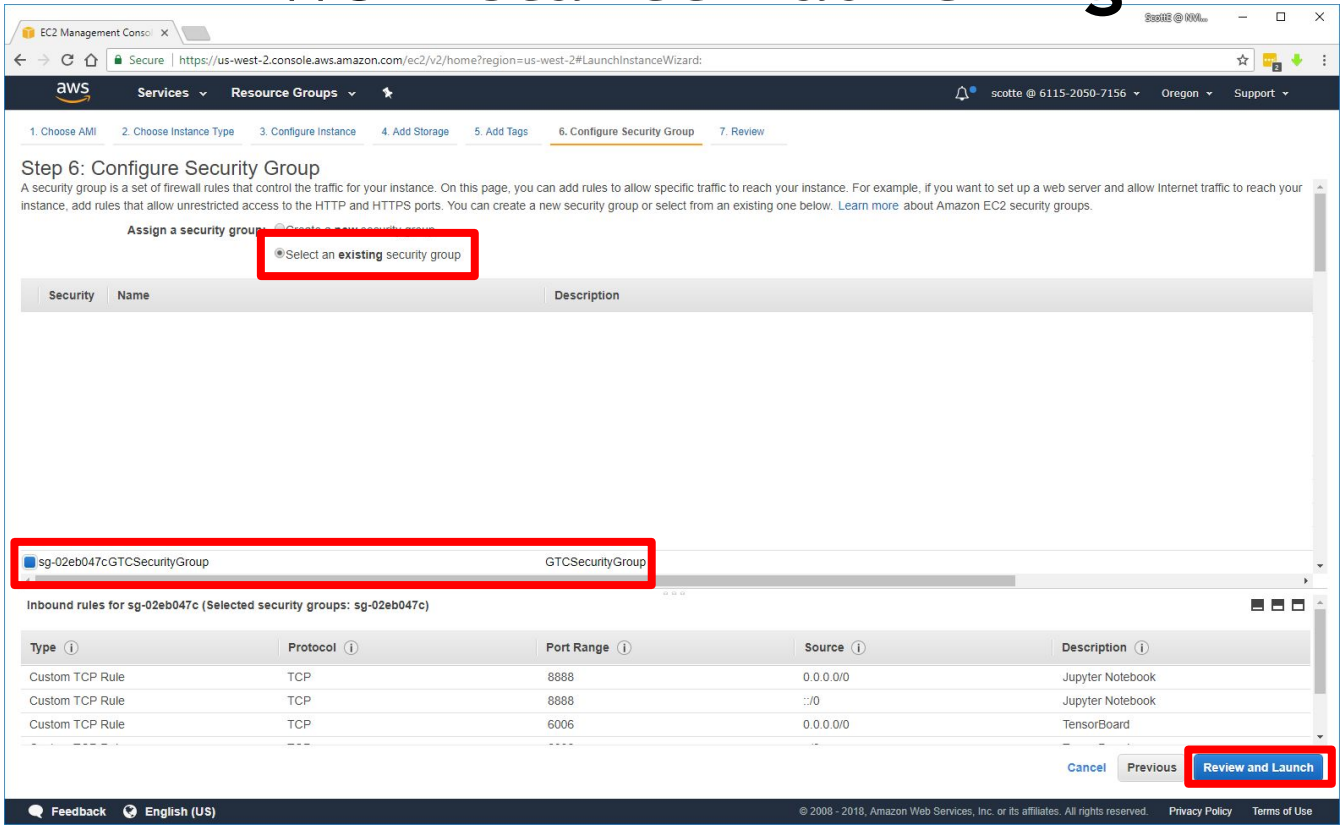
AWS Instance Launching



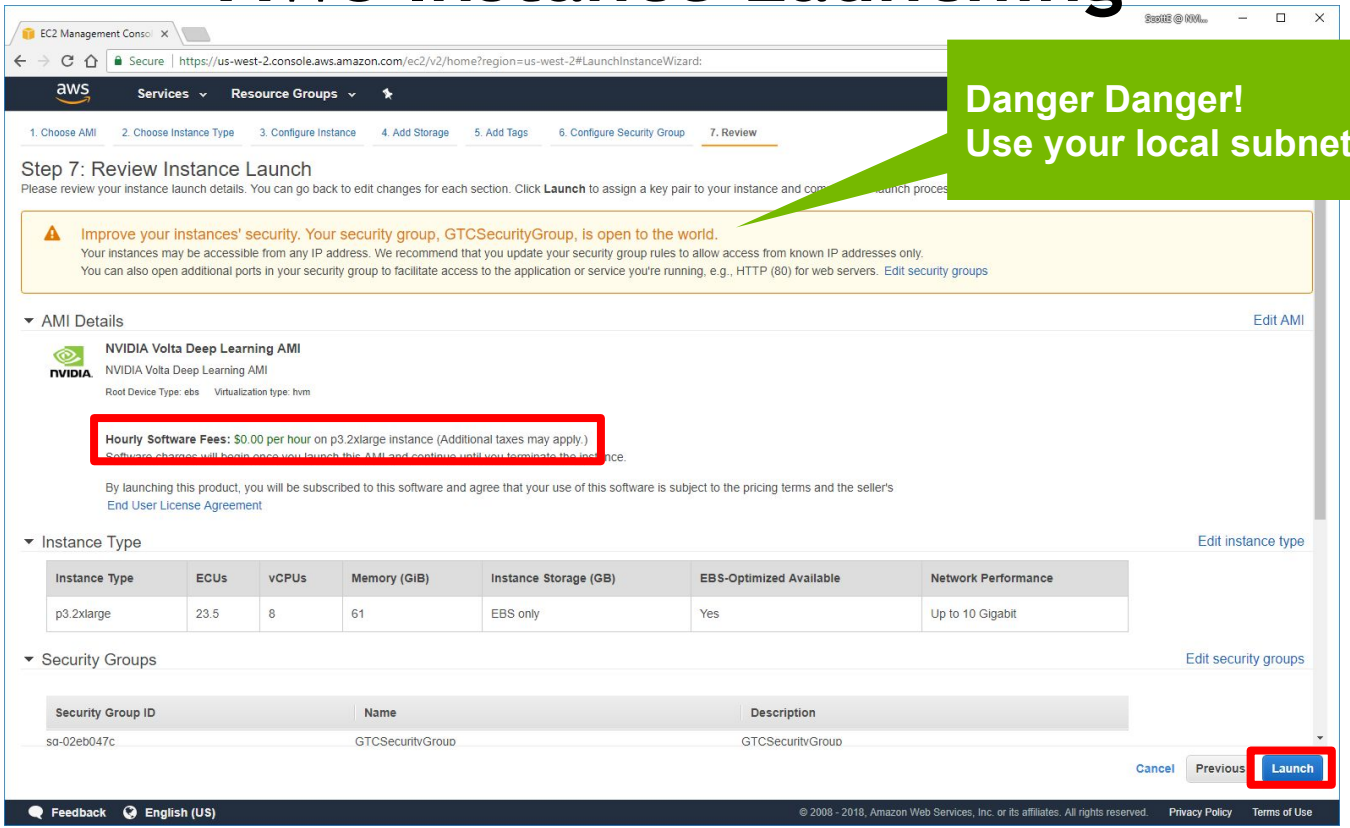
AWS Instance Launching



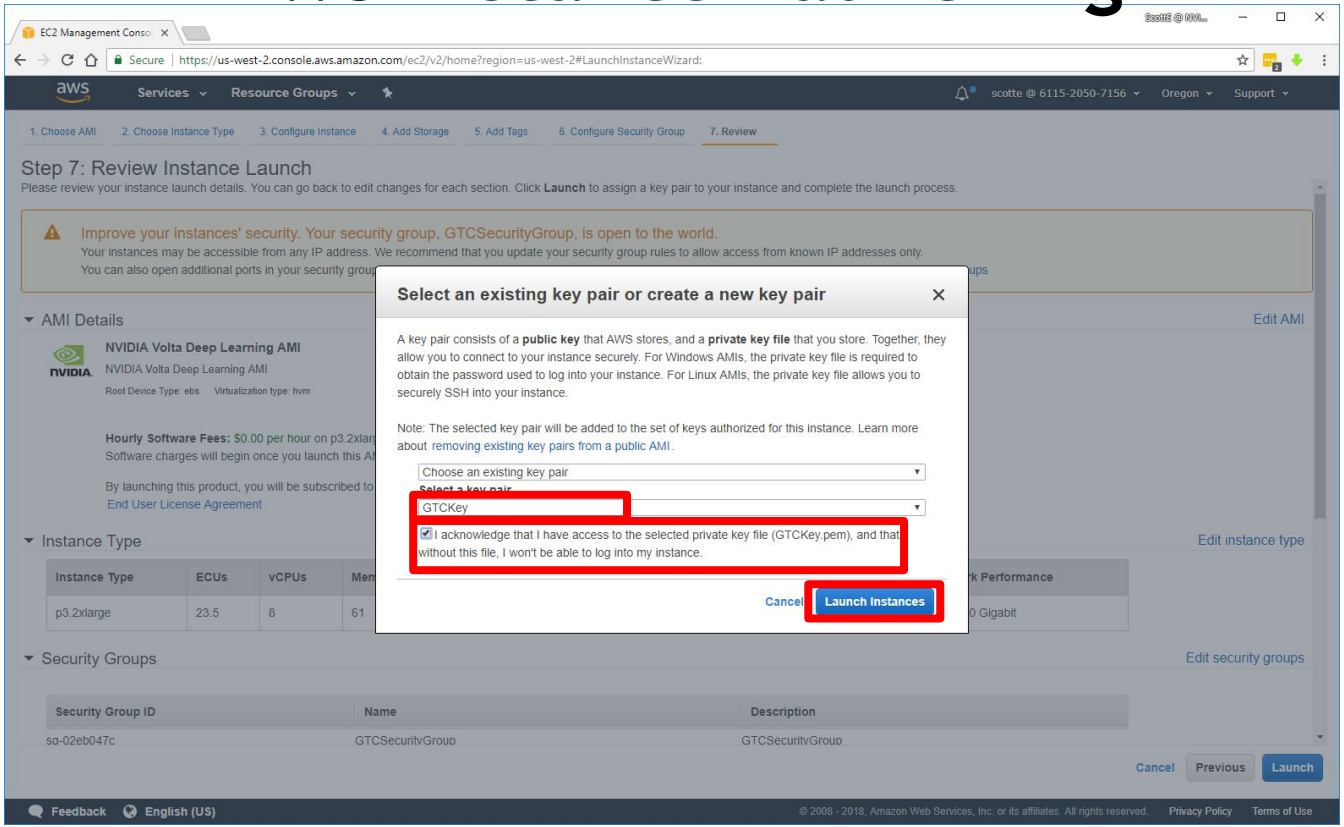
AWS Instance Launching



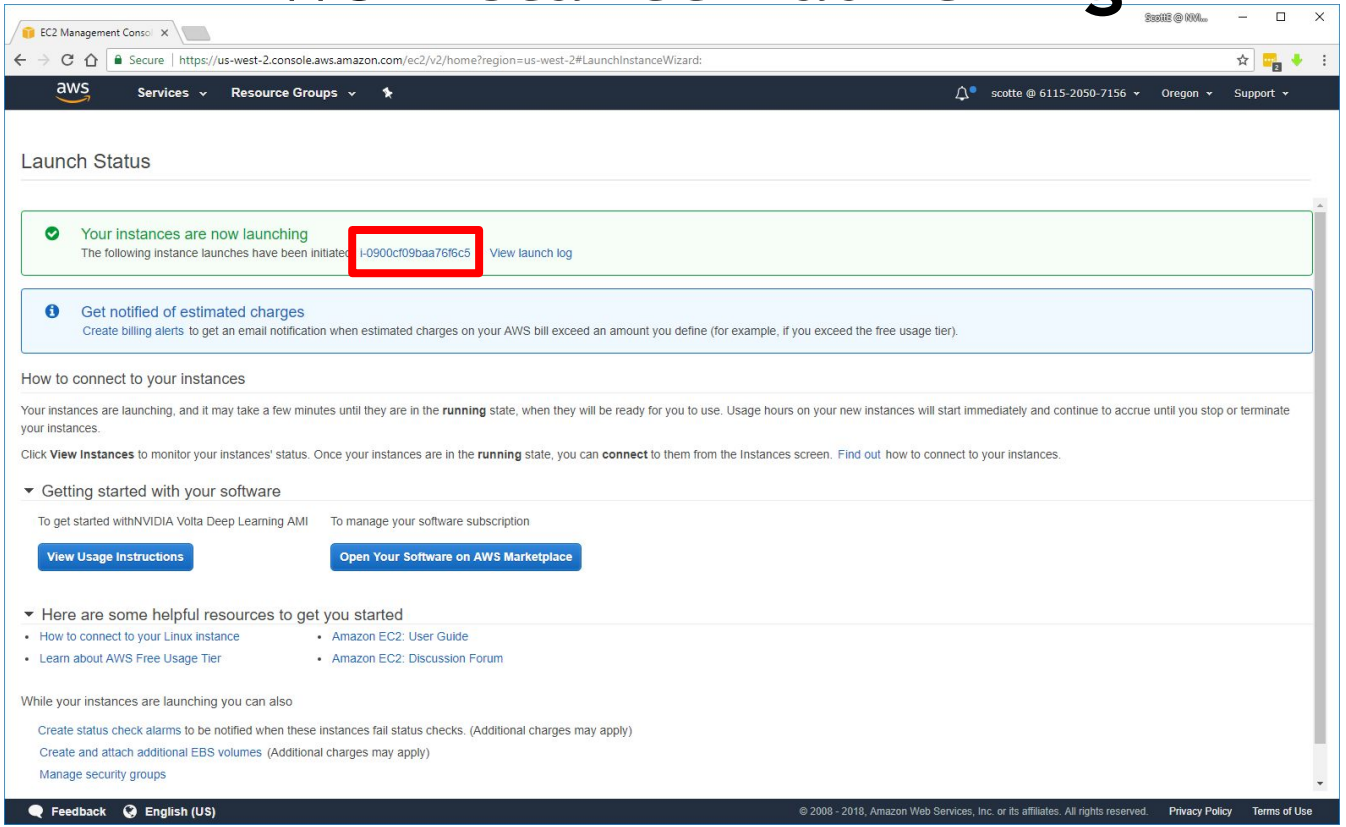
AWS Instance Launching



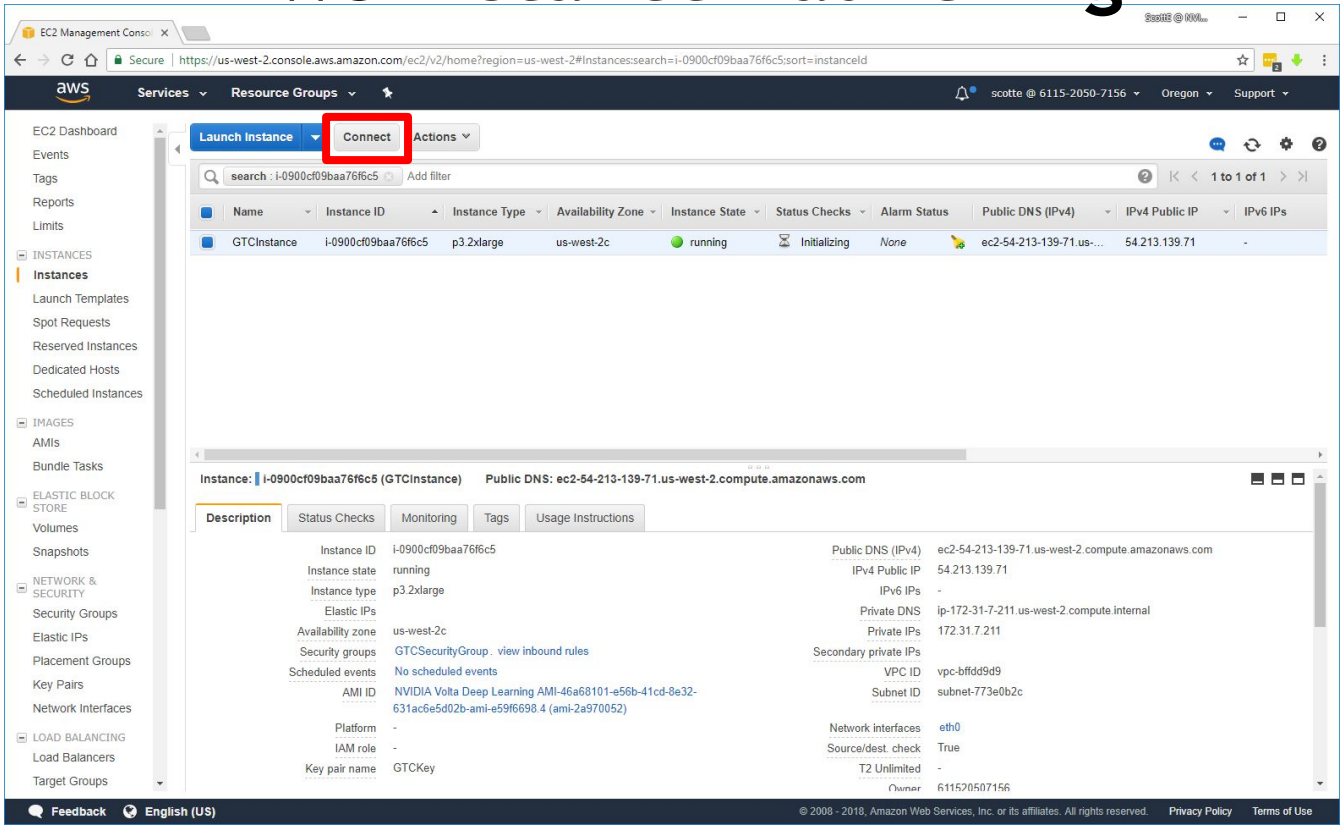
AWS Instance Launching



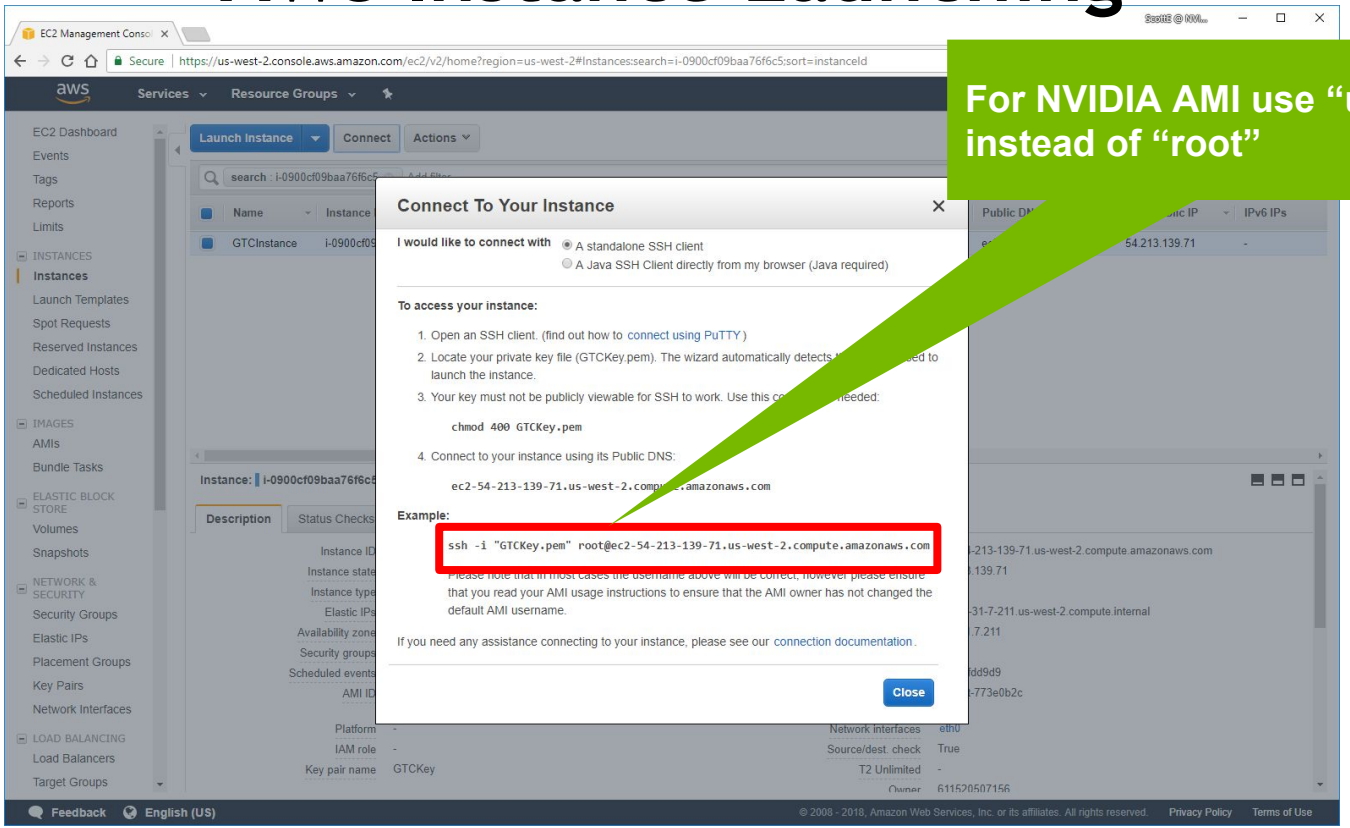
AWS Instance Launching



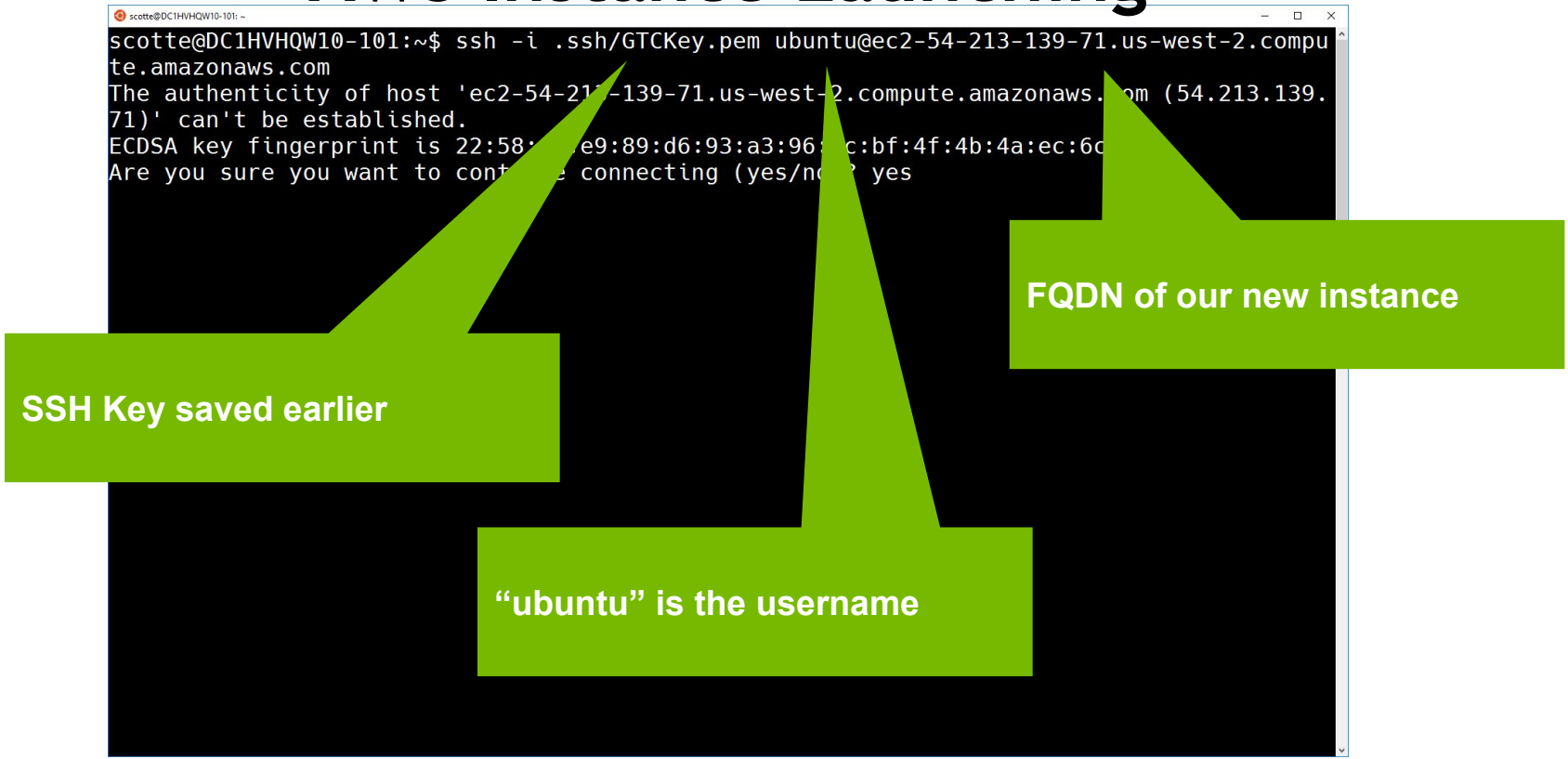
AWS Instance Launching



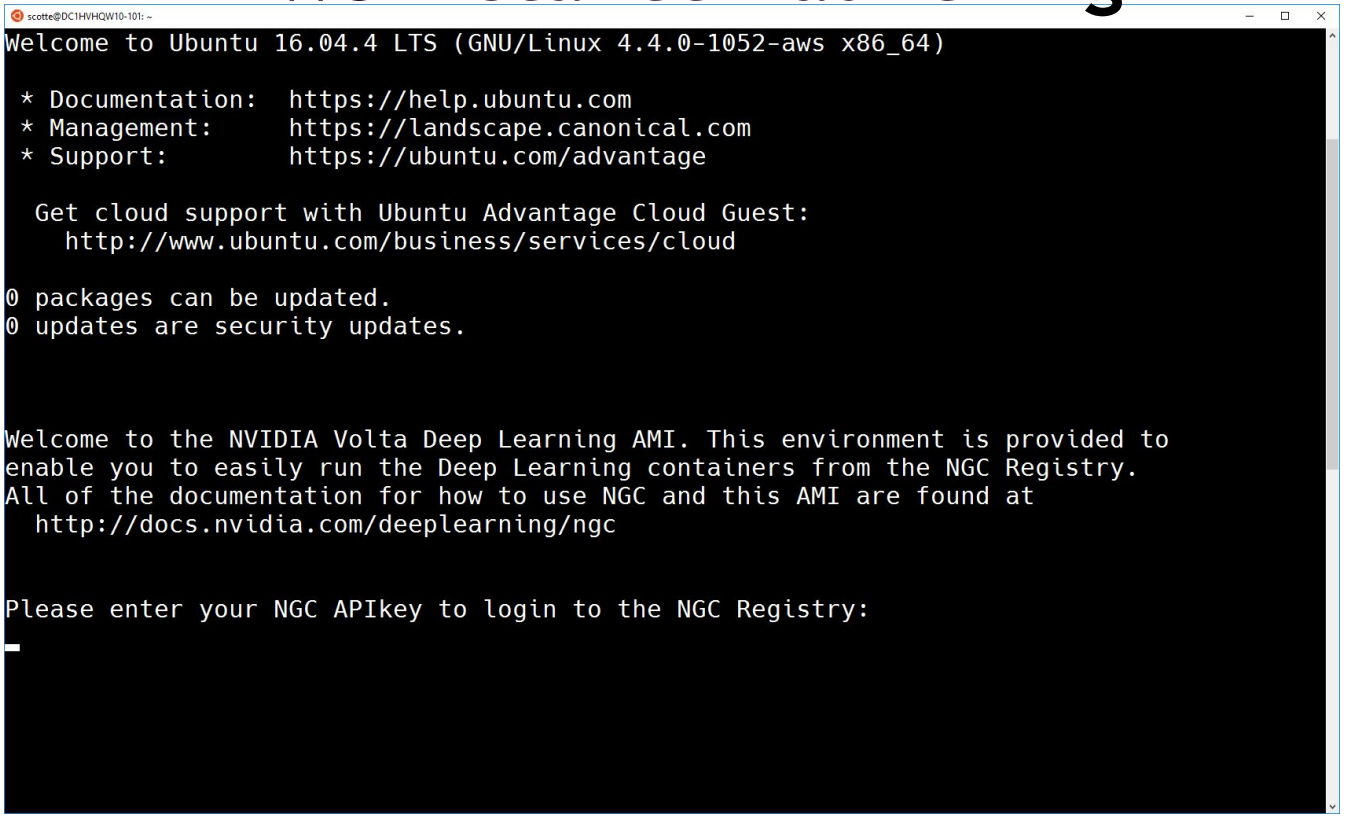
AWS Instance Launching



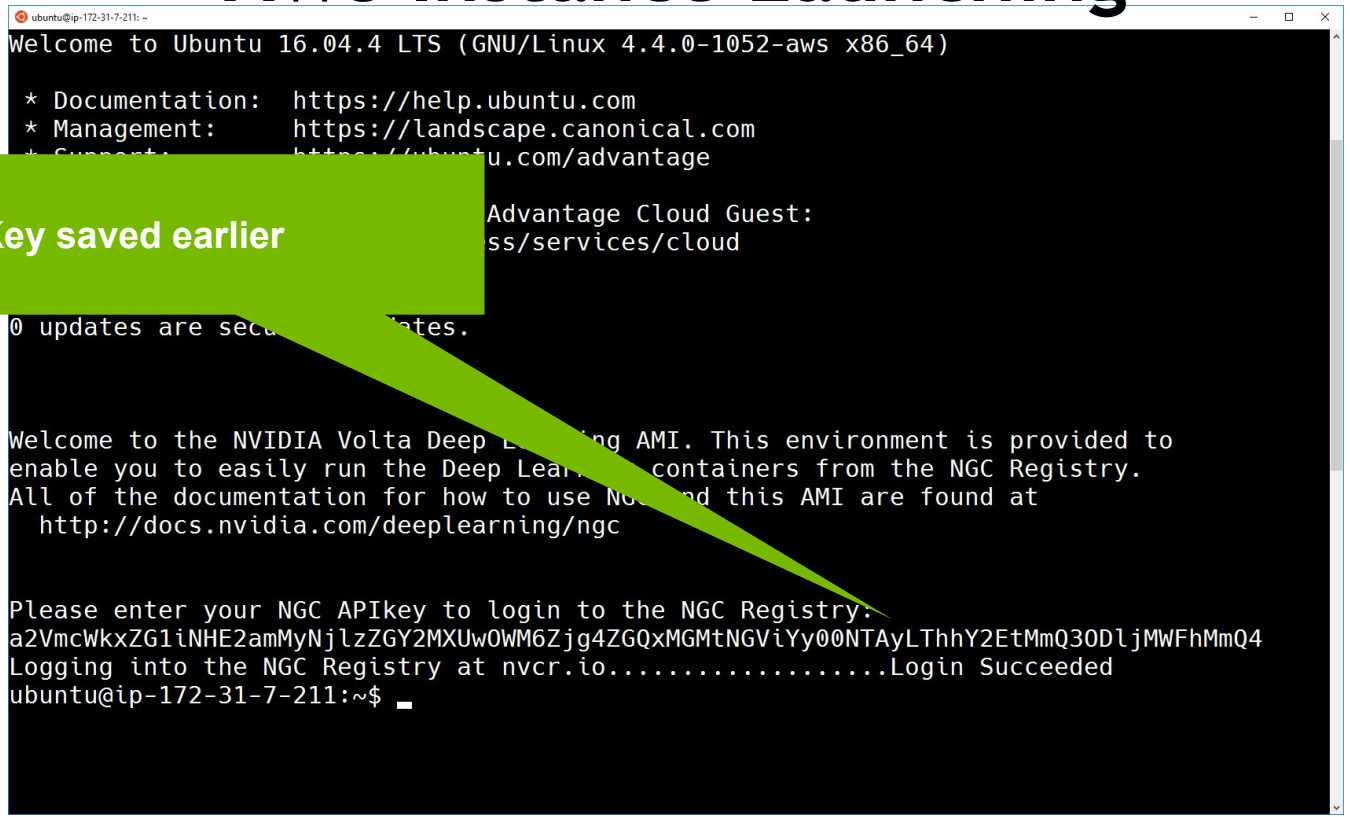
AWS Instance Launching



AWS Instance Launching



AWS Instance Launching



API Key saved earlier

Run a Container

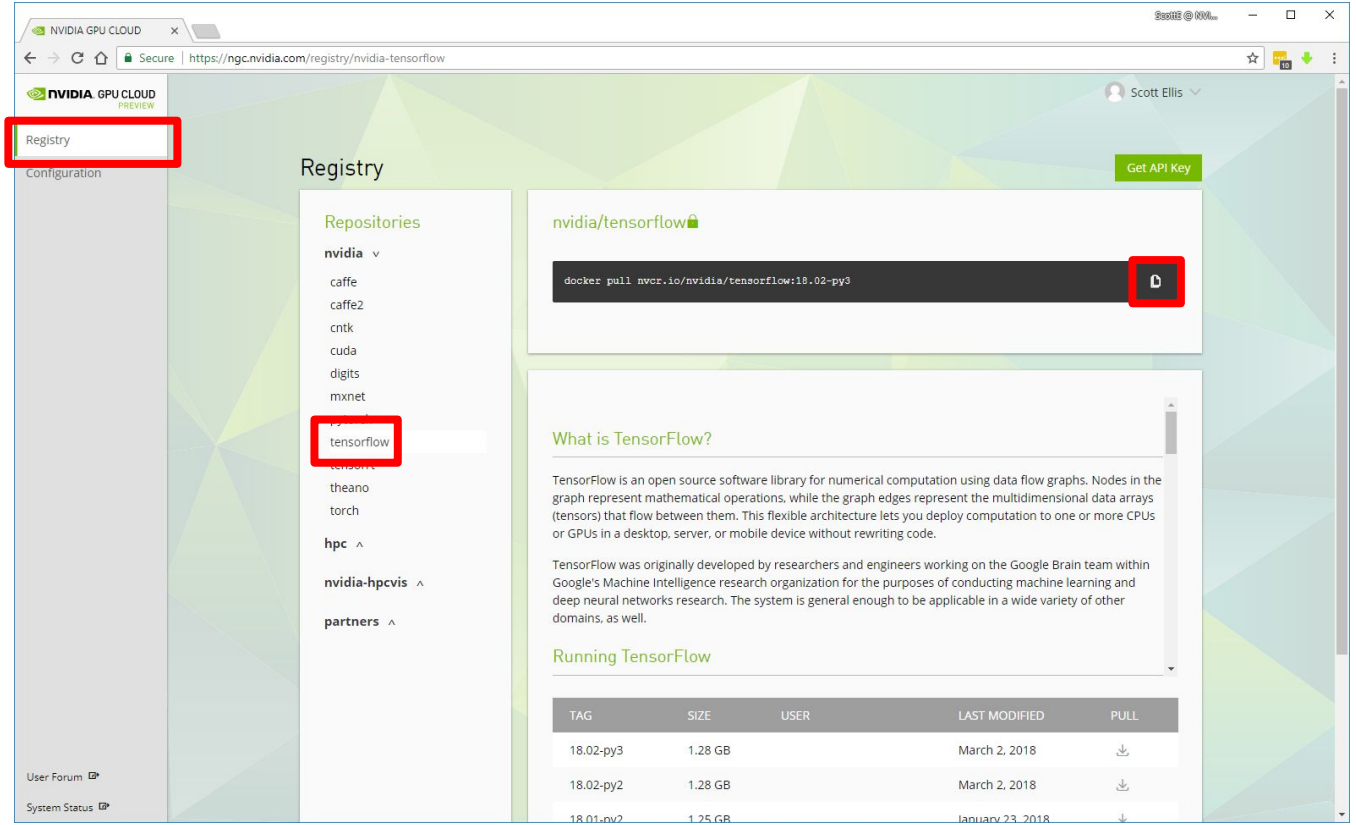
NGC Container Execution

Quick TensorFlow Run

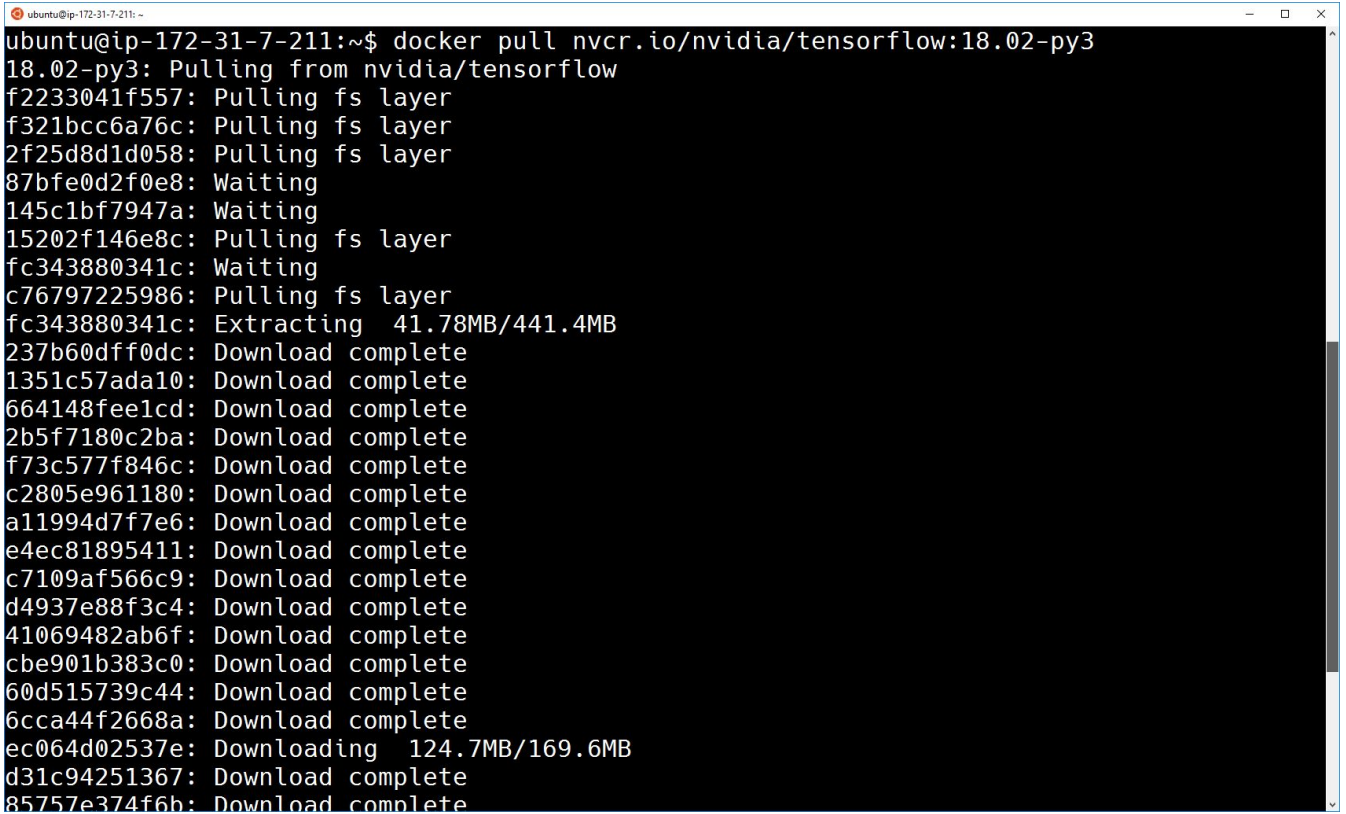
Our challenge:

- Download a TensorFlow container from NGC
- Run the container in our P3 instance

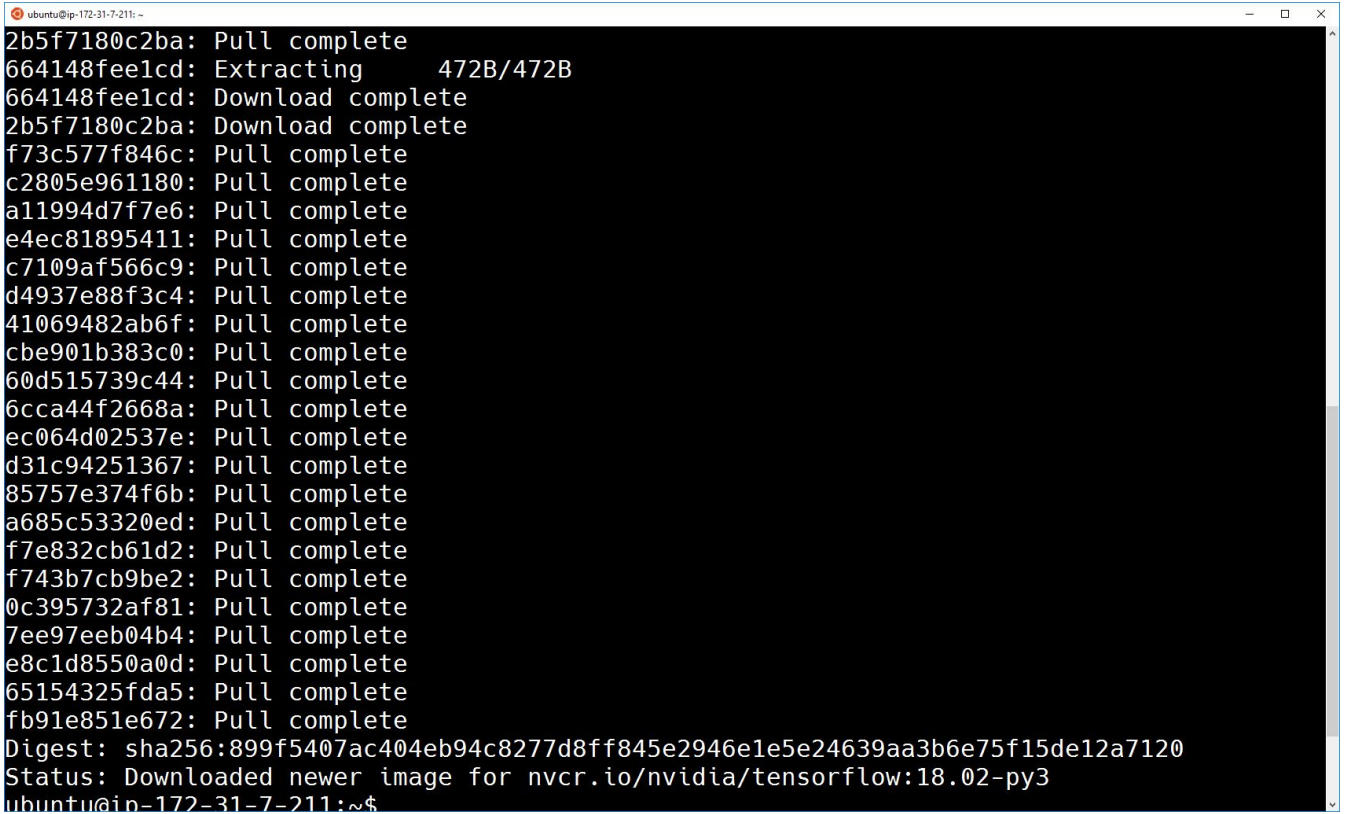
NGC Container Execution



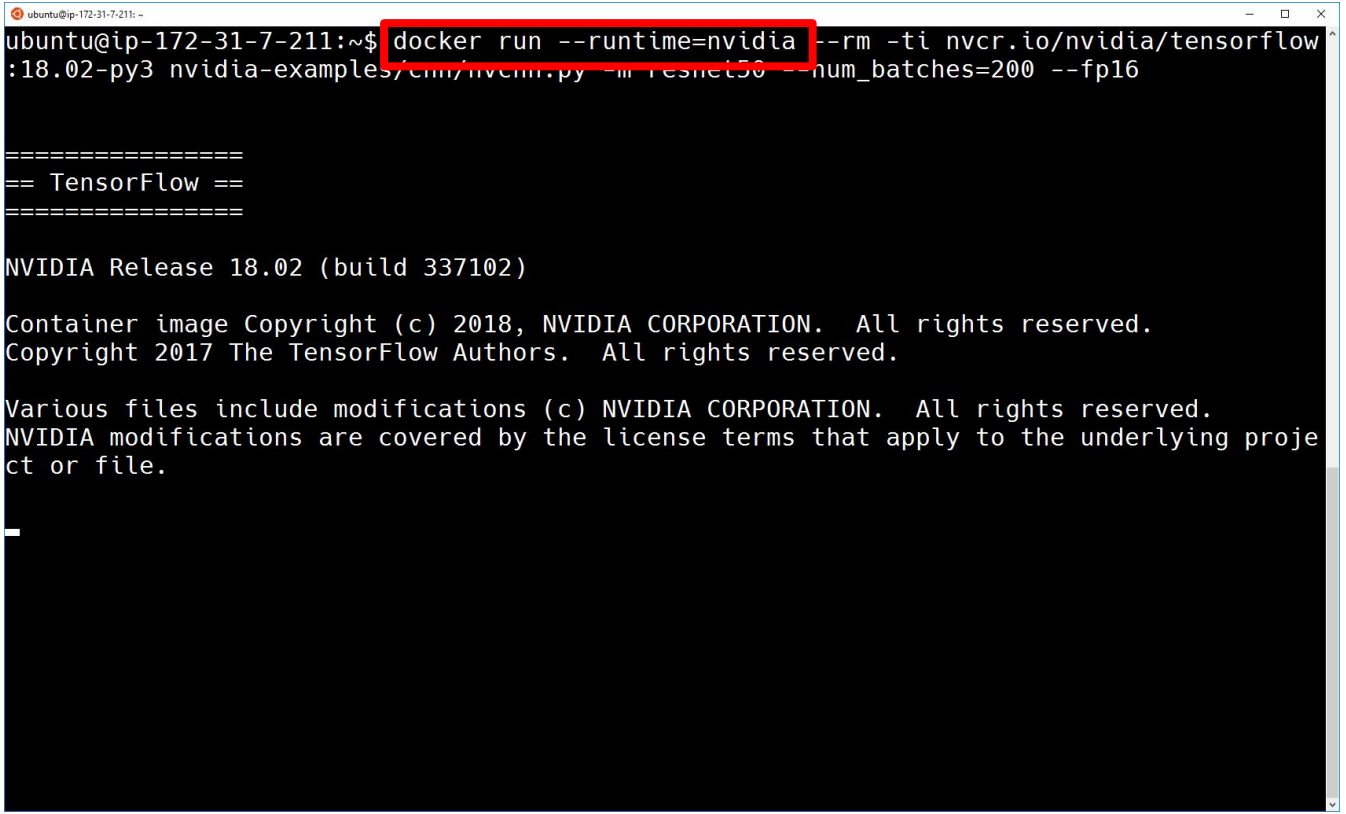
NGC Container Execution



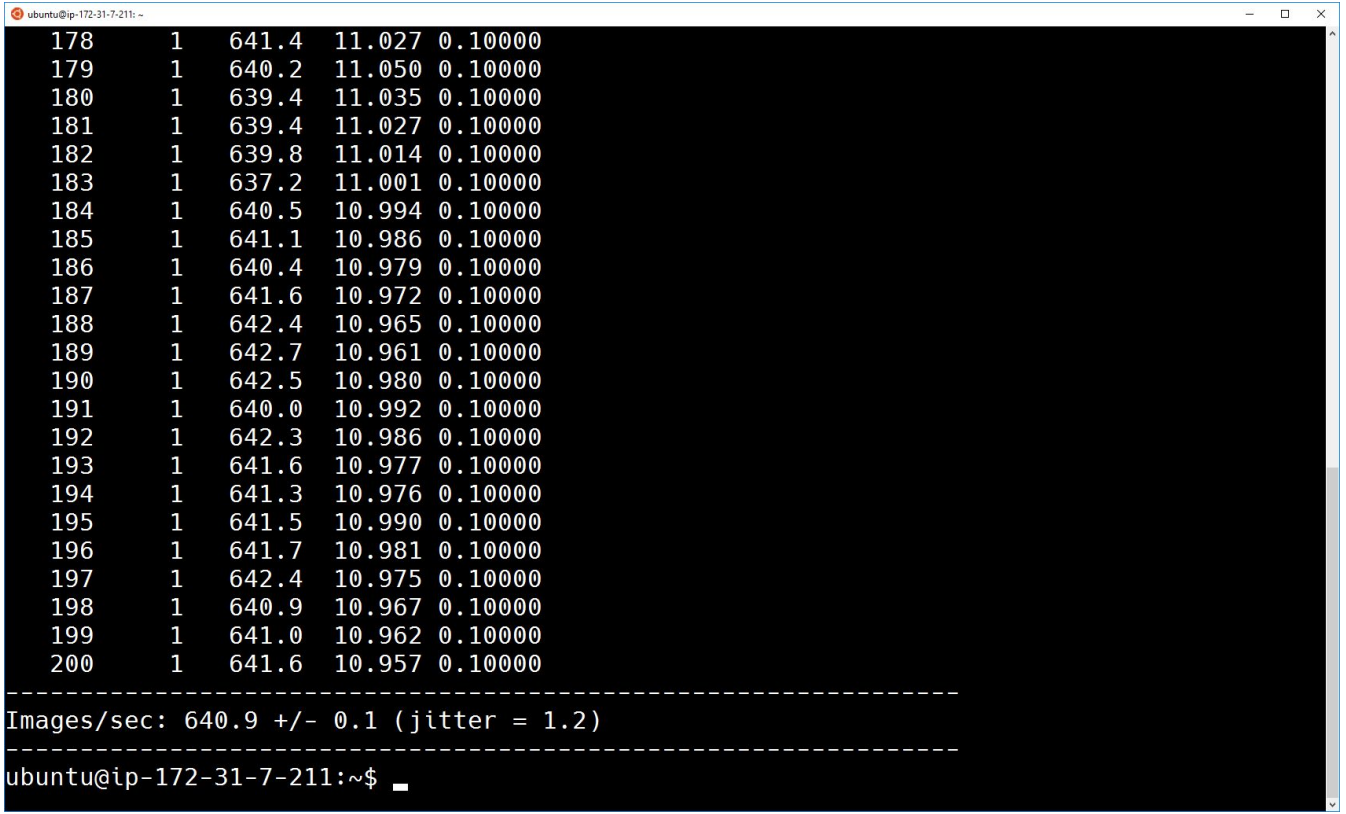
NGC Container Execution



NGC Container Execution



NGC Container Execution



Wait. What Just Happened?

That was too fast.

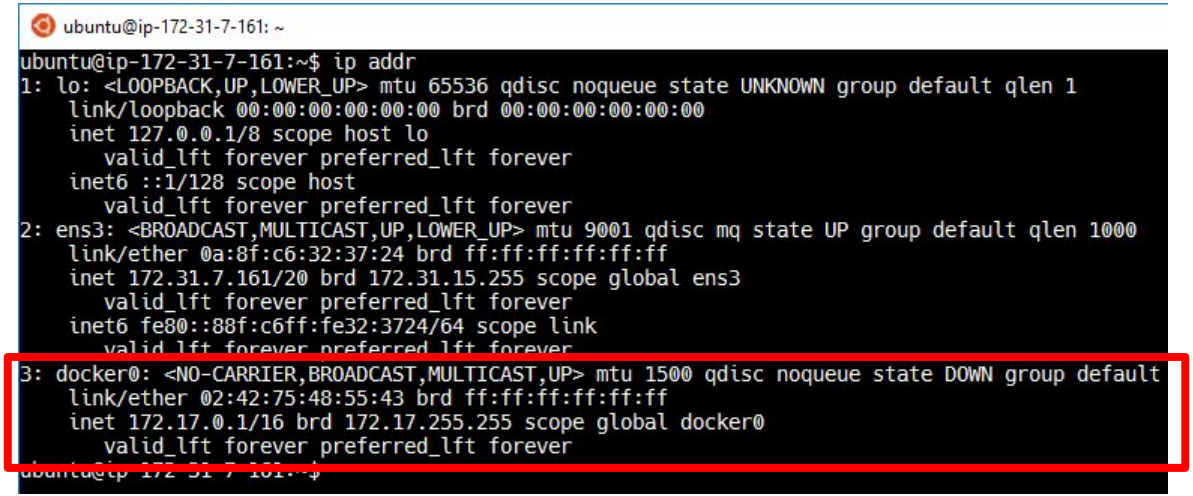
1. Logged into NGC and created an API key
2. Logged into AWS and setup our account
   a. Created SSH Key Pair
   b. Created Security Group
3. Launched AWS P3 instance with “NVIDIA Volta Deep Learning” AMI
   a. Logged into the P3 instance with SSH
   b. Logged into NGC with the API key
4. Downloaded the TensorFlow container from NGC
5. Ran nvcnn.py sample with Resnet-50 with synthetic ImageNet data for 200 epochs
6. Profit!

Exposing Ports

Accessing Container Services

What about things in the container?

Applications in a container are on their own network (‘docker0’ bridge)



Tell Docker you want to use them at runtime (remember `-p` ?)

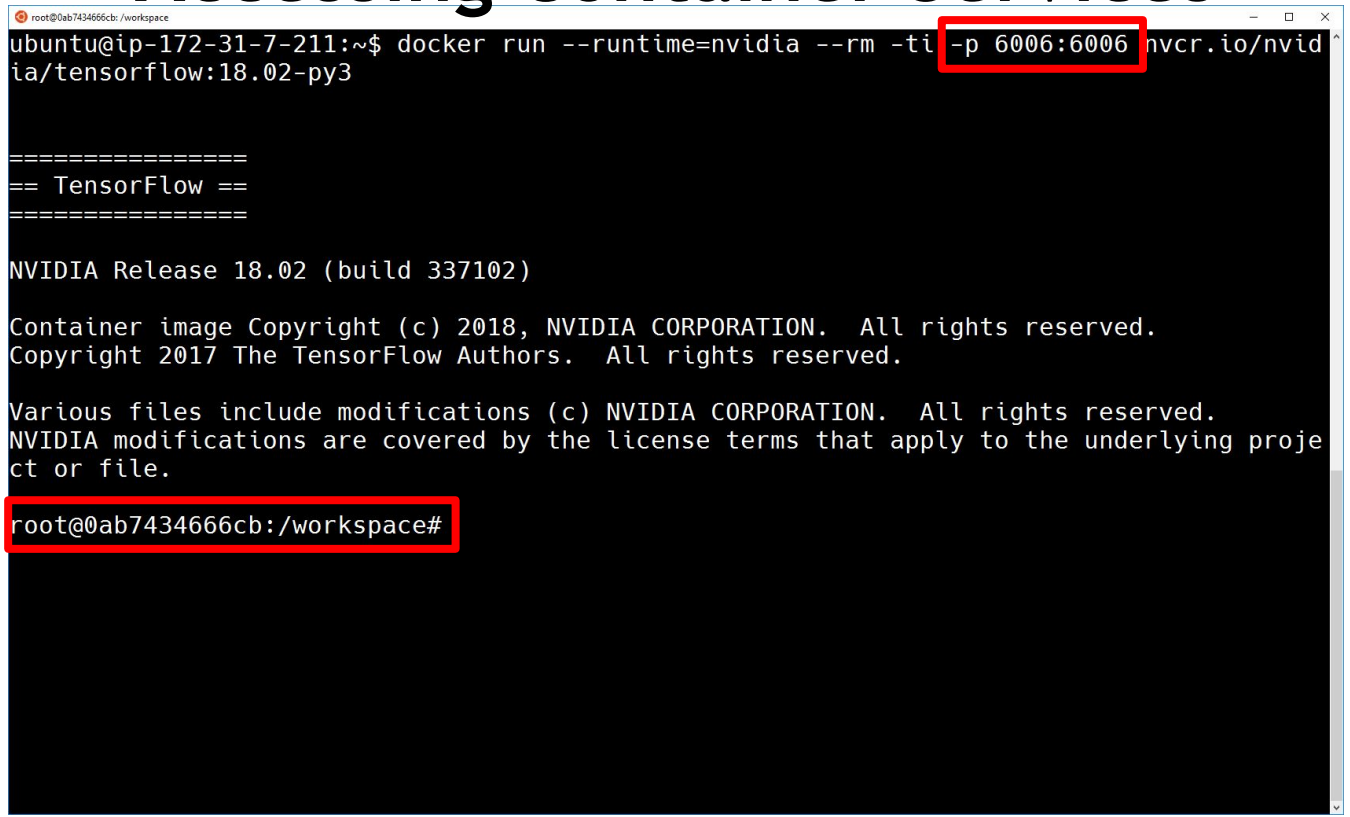
Accessing Container Services

Passing through a TCP port

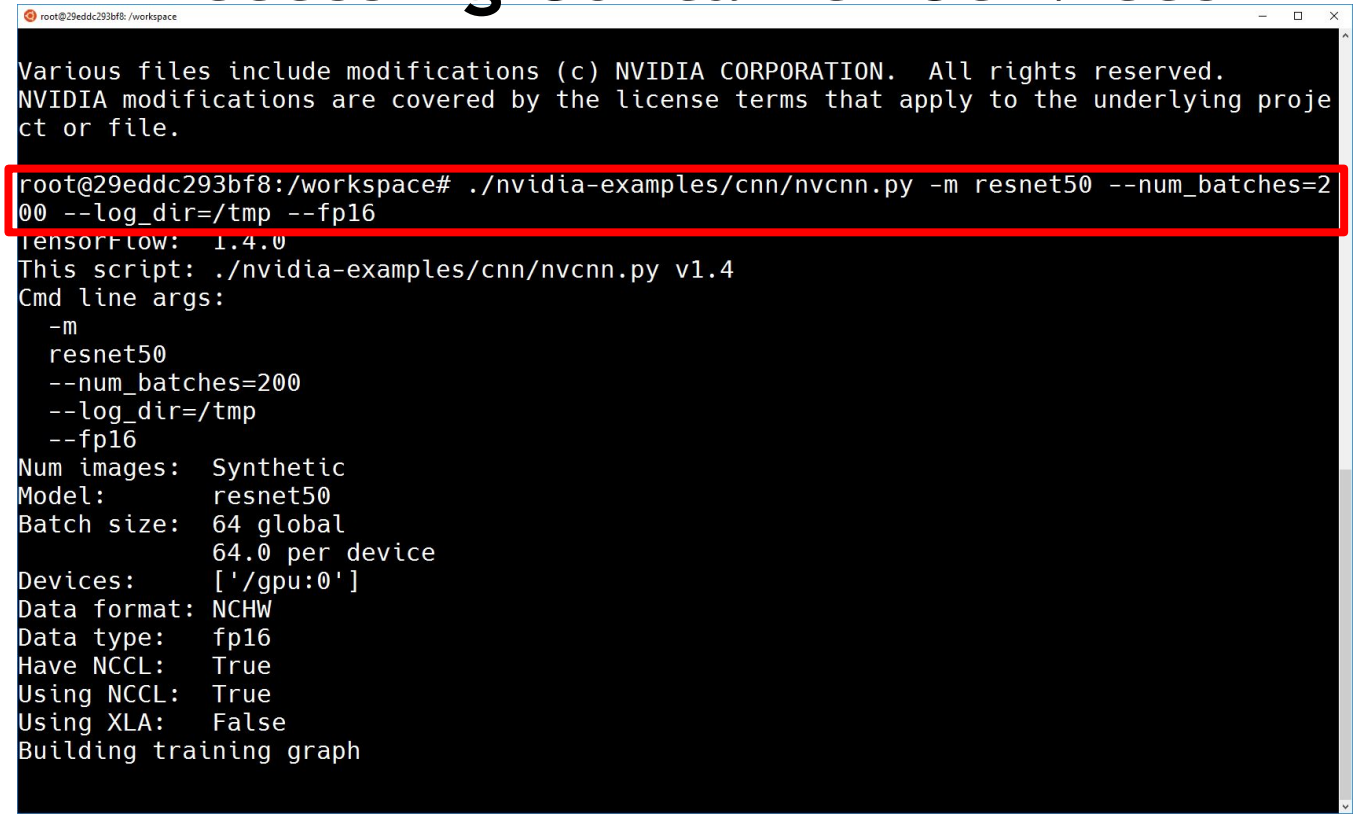
Our challenge:

- Launch the TensorFlow container
  - Run interactively (`-ti`)
  - Expose Tensorboard port 6006 (`-p`)
- Repeat our prior training run
  - Save log data to /tmp
- Run Tensorboard
- Visualize our training run

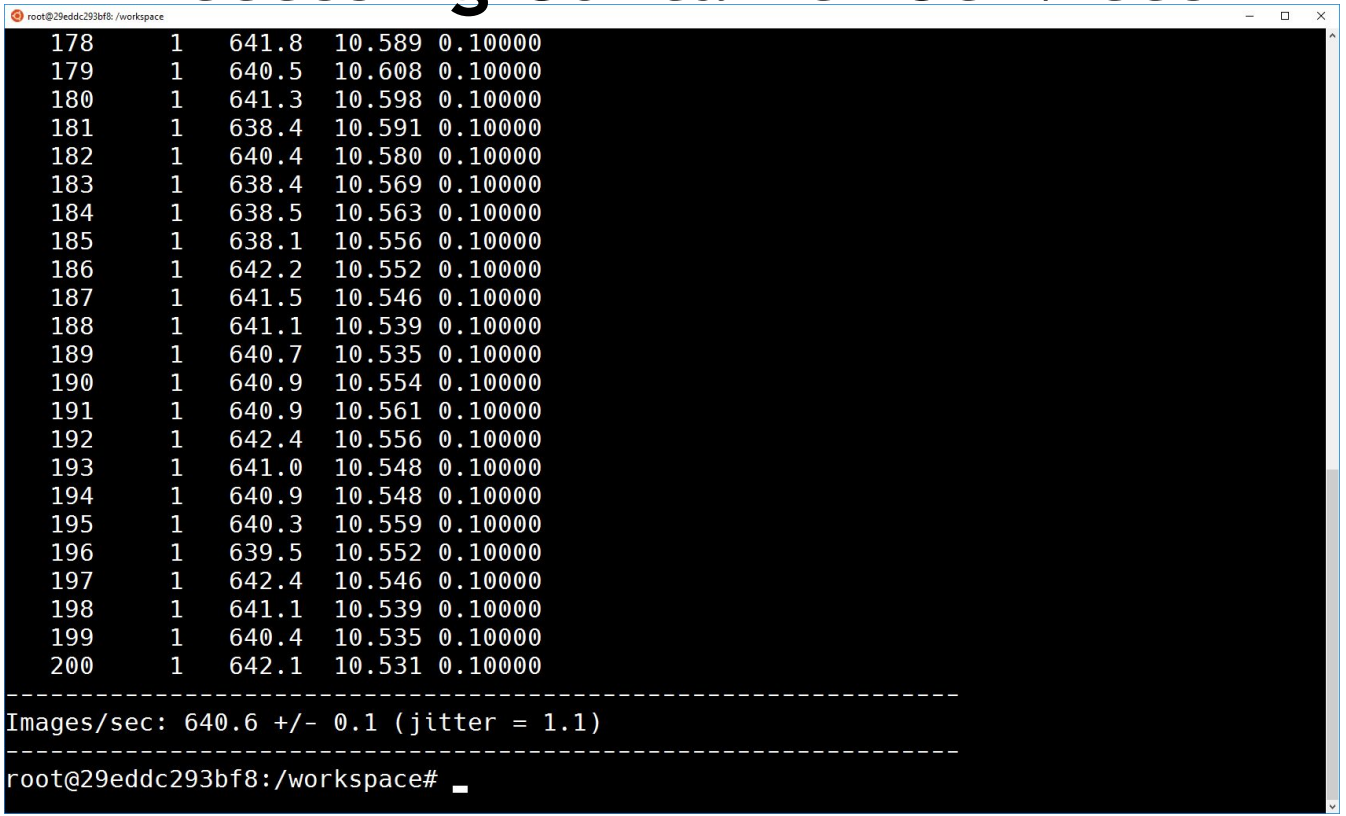
Accessing Container Services



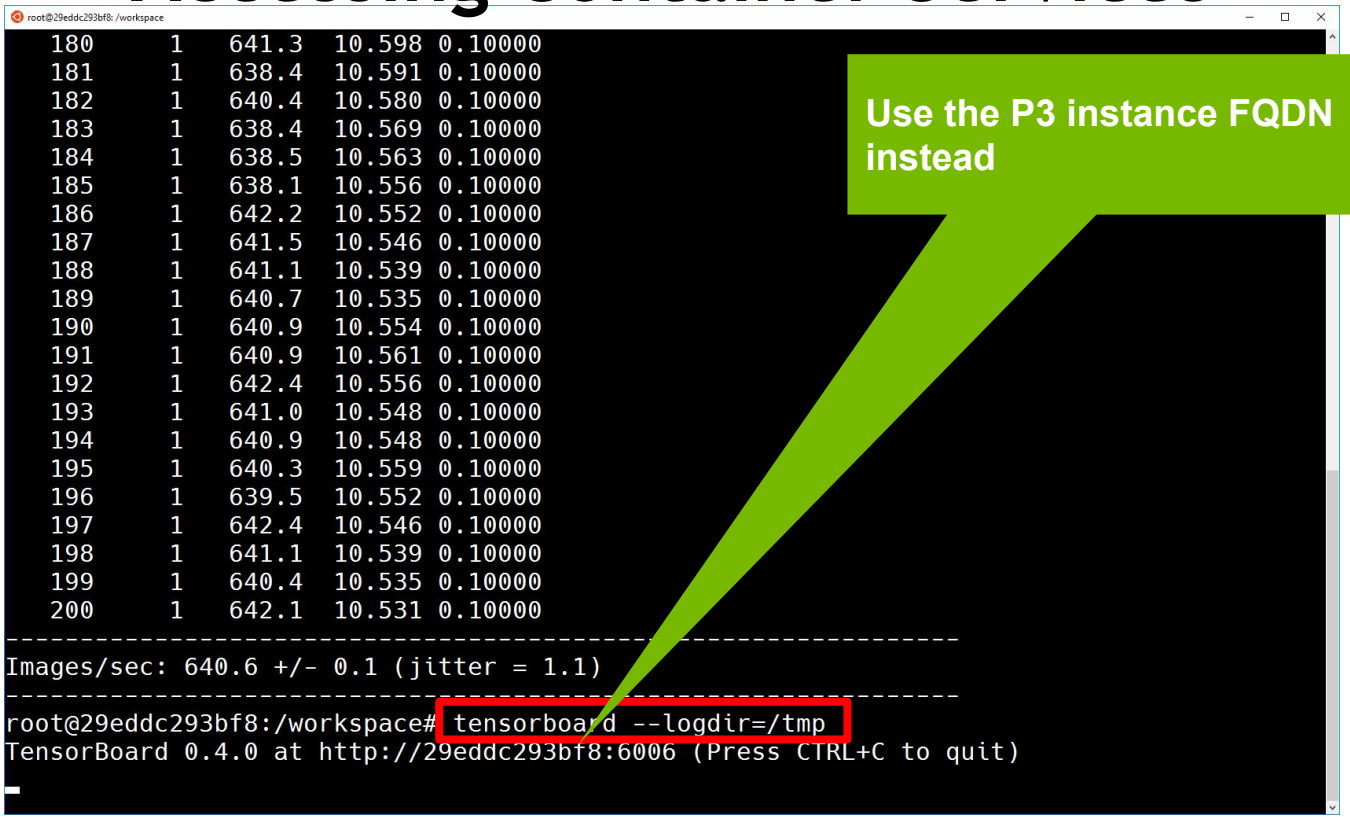
Accessing Container Services



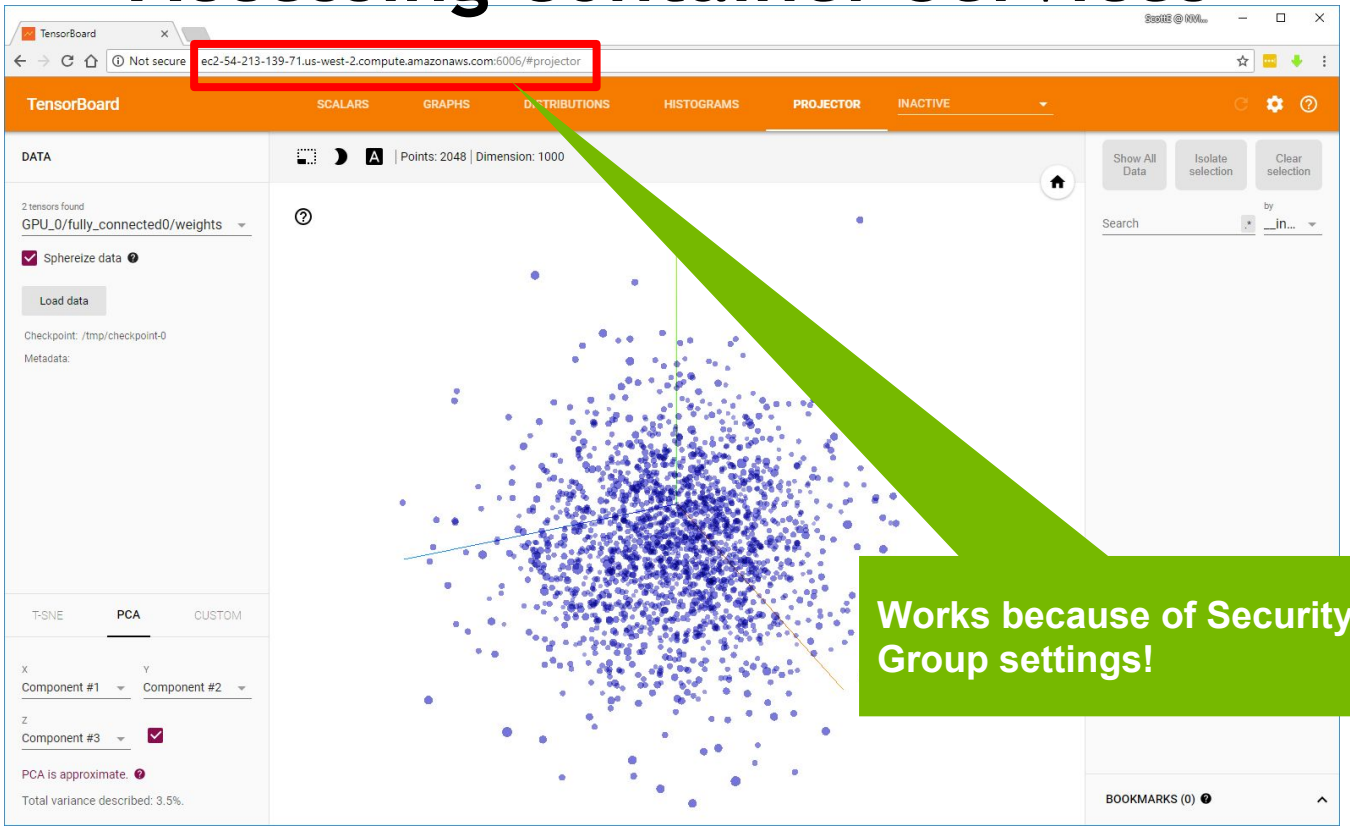
Accessing Container Services



Accessing Container Services



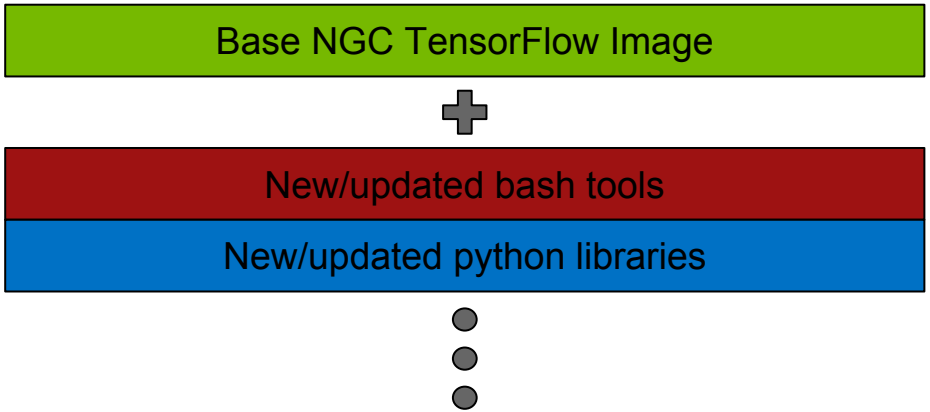
Accessing Container Services



Extending NGC Images

Extending NGC Images

Layers layers layers



- Often the out-of-the-box image is not enough
- Need extra tools/applications
  - Additional layers on top of base image

- Dockerfile allows for building custom images
  - `docker build` command creates new image from set of instructions

Extending NGC Images

A Dockerfile is a script that contains instructions to custom configure a container from a base image

Here are some common commands:

- FROM is Mandatory as the first instruction. It denotes the base image to be built from. Use a tag to specify the image.
- RUN = Creates a new layer with the output of the specified commands.
- WORKDIR = Directory the command will start it
- CMD = Default command executed when Docker container is started. Use only one CMD instruction in a Dockerfile.

```
1  FROM nvcr.io/nvidia/tensorflow:18.02-py3
2
3  RUN pip install jupyter
4
5  WORKDIR /notebooks
6
7  CMD jupyter notebook --allow-root --ip=0.0.0.0
```

Best practices for writing Dockerfiles

https://docs.docker.com/engine/userguide/eng-image/dockerfile_best-practices/

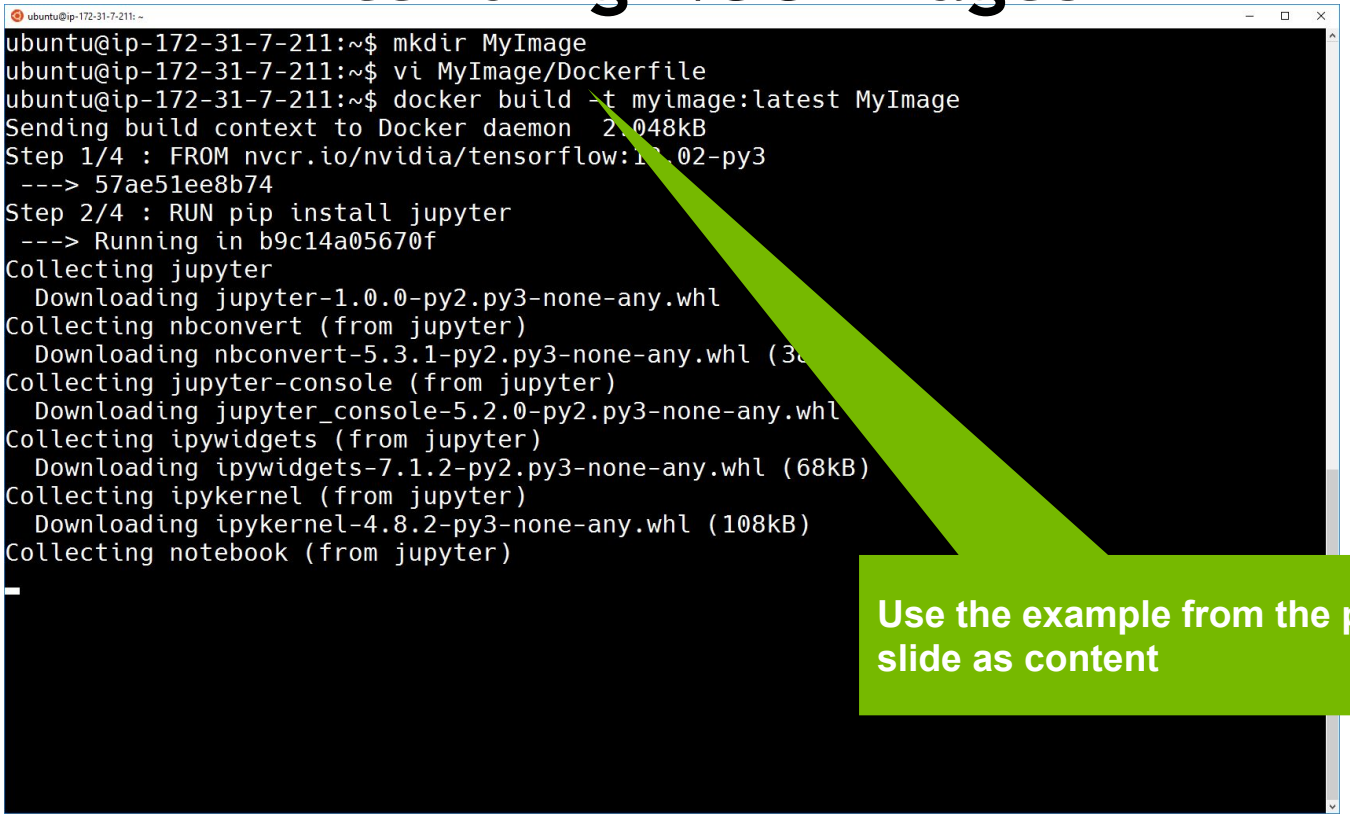
Extending NGC Images

Add Jupyter

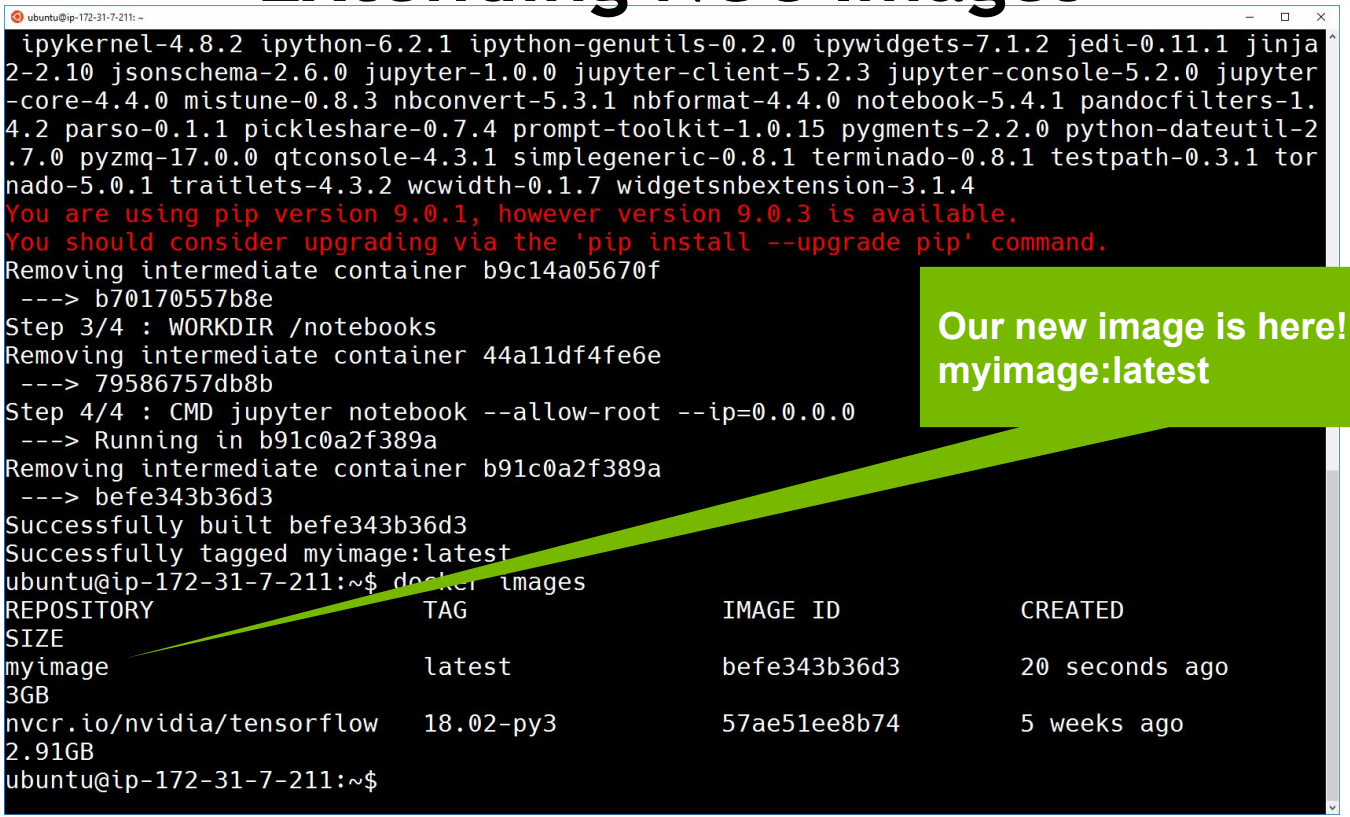
Our challenge:

- Add Jupyter to the NVIDIA TensorFlow Image
- Launch jupyter notebook automatically when the container starts
  - By default jupyter listens on port 8888
- Verify it worked!

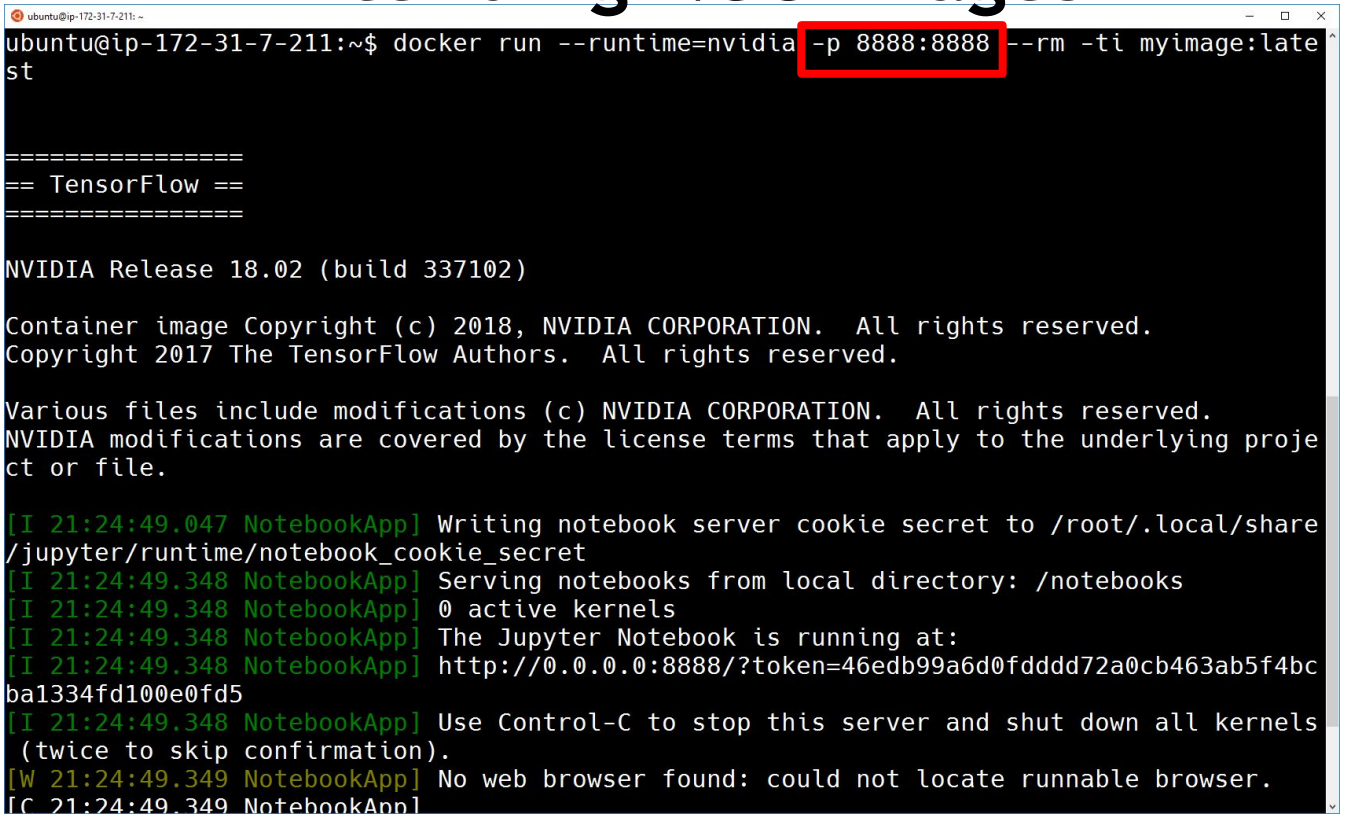
Extending NGC Images



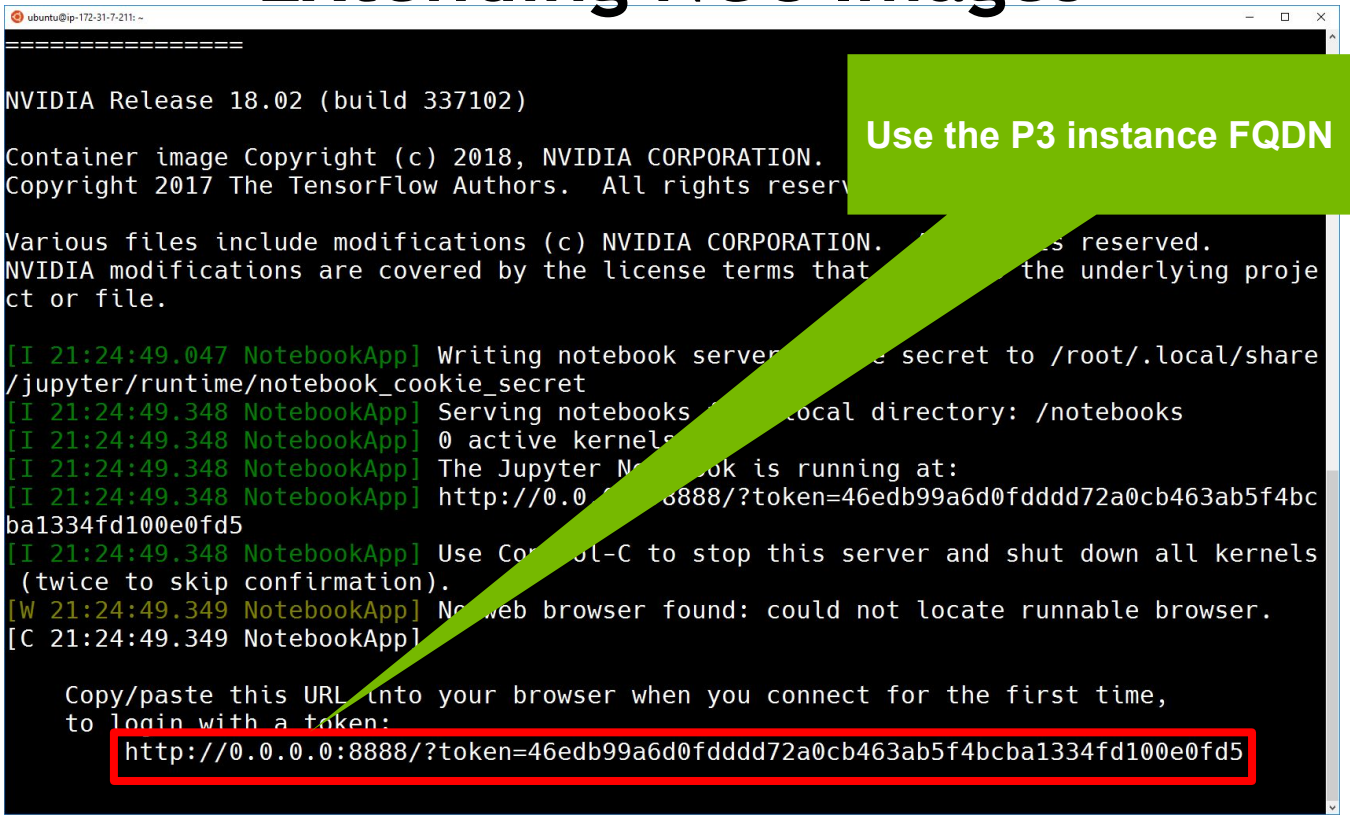
Extending NGC Images



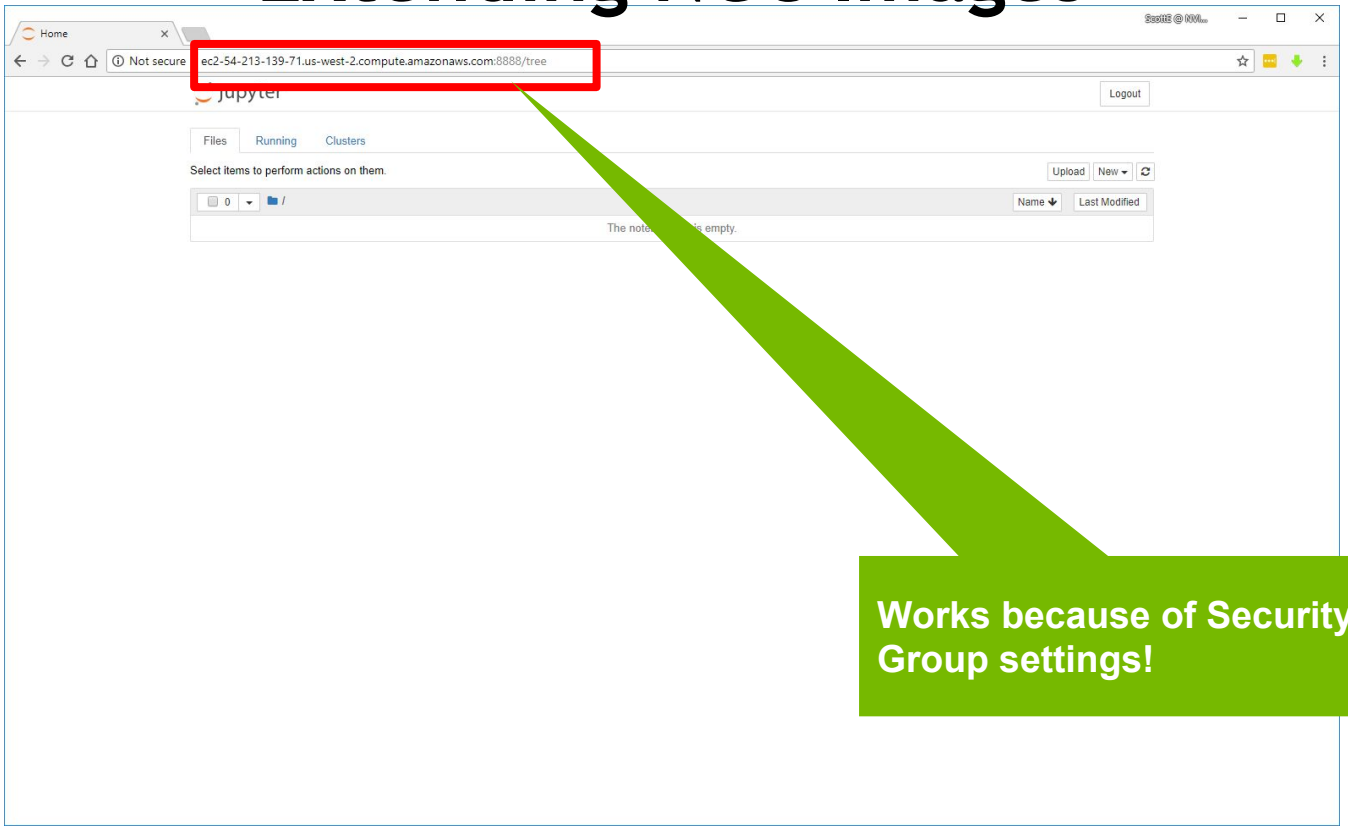
Extending NGC Images



Extending NGC Images



Extending NGC Images



Using Your Own Data

Using Your Own Data

How do you access actual data?

Many options for accessing data

- Object Storage via S3 Buckets
- Block Storage via Elastic Block Storage (EBS)
- File Storage via Elastic File Storage (EFS)
- 3rd party storage choices

We'll pick Elastic File Storage (EFS) for simplicity

Using Your Own Data

What to do

- Before we can use EFS we must (this should look familar)
  - Secure network access with a security profile

This is the same procedure as with our EC2 instance (so we'll skip it)

(Enable NFS as the port)

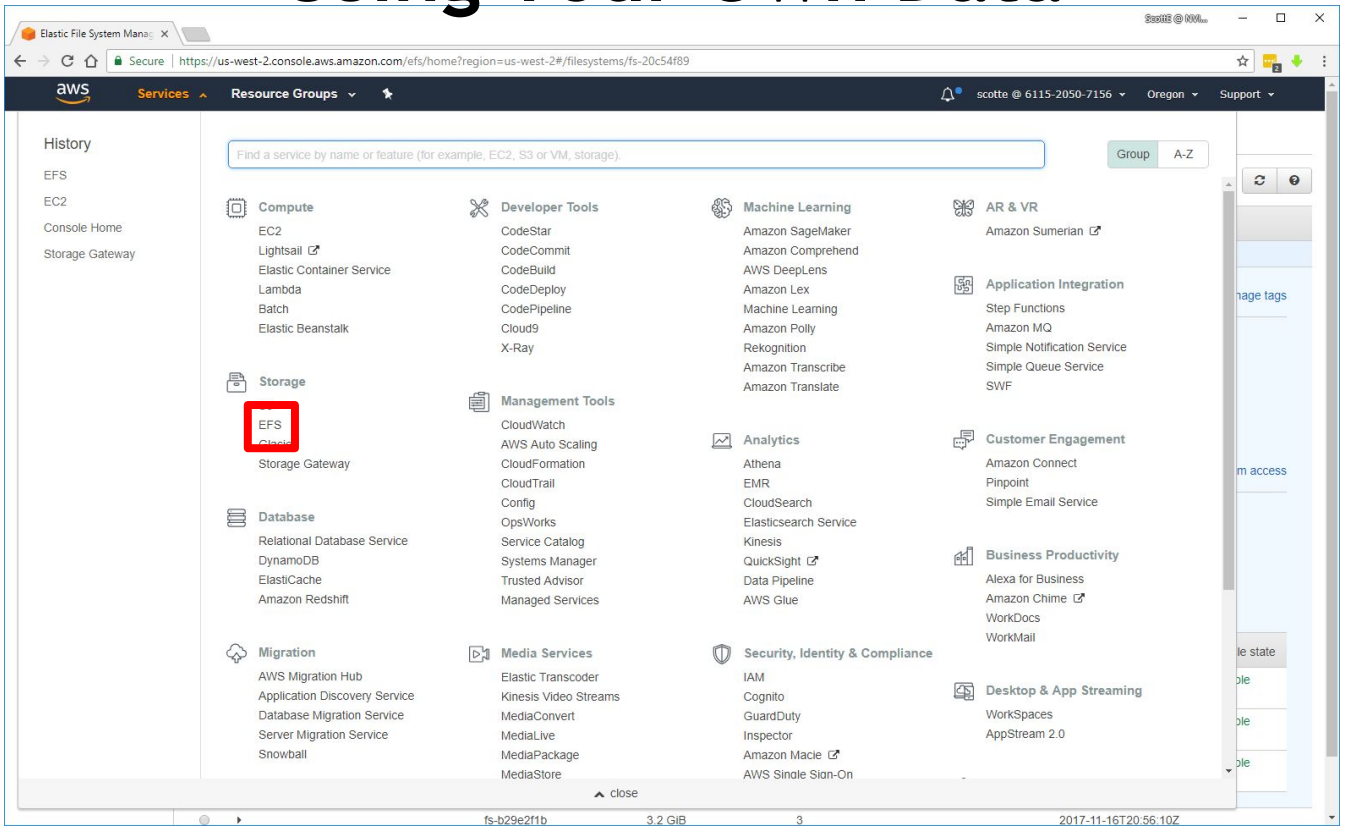
Using Your Own Data

What to do

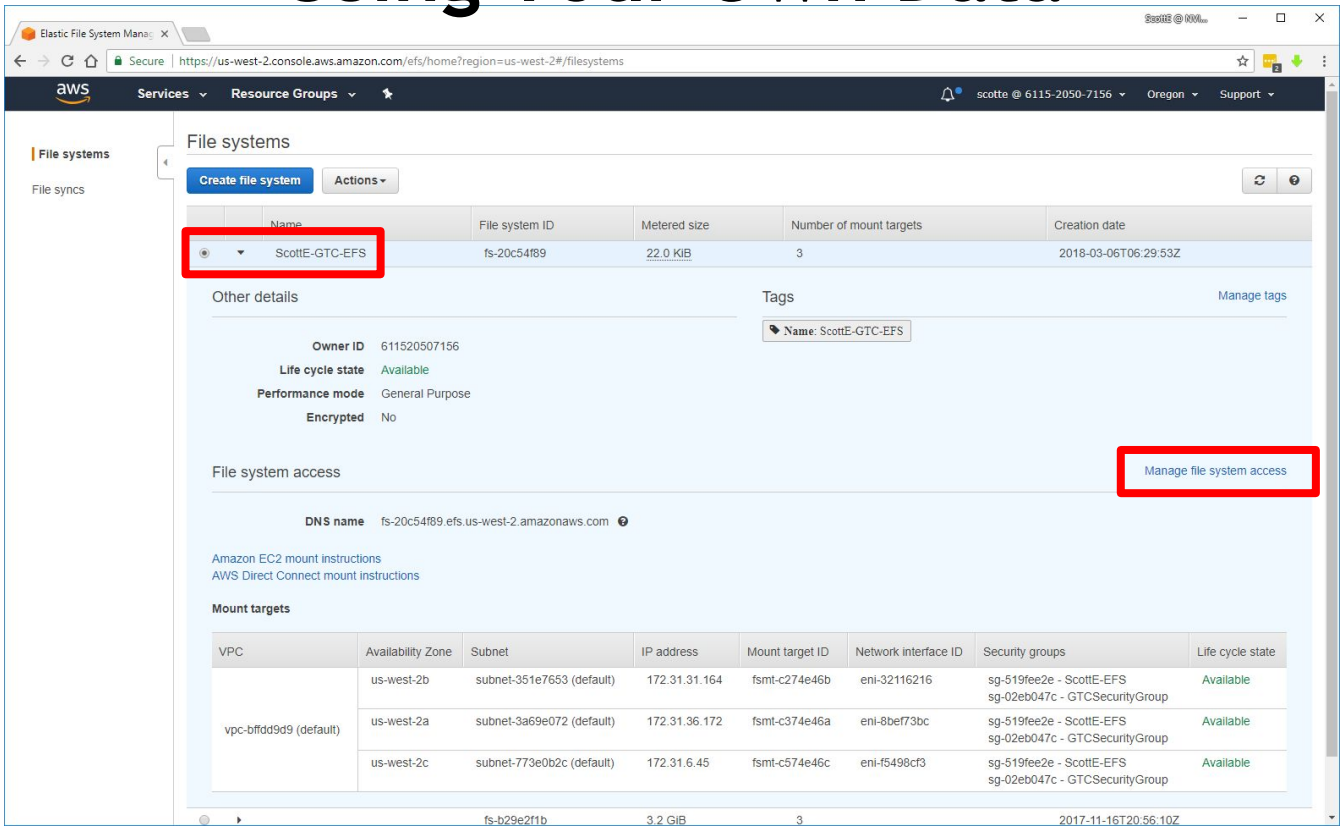
Our challenge:

- Mount the storage to our P3 instance
- Launch our container accessing the new data

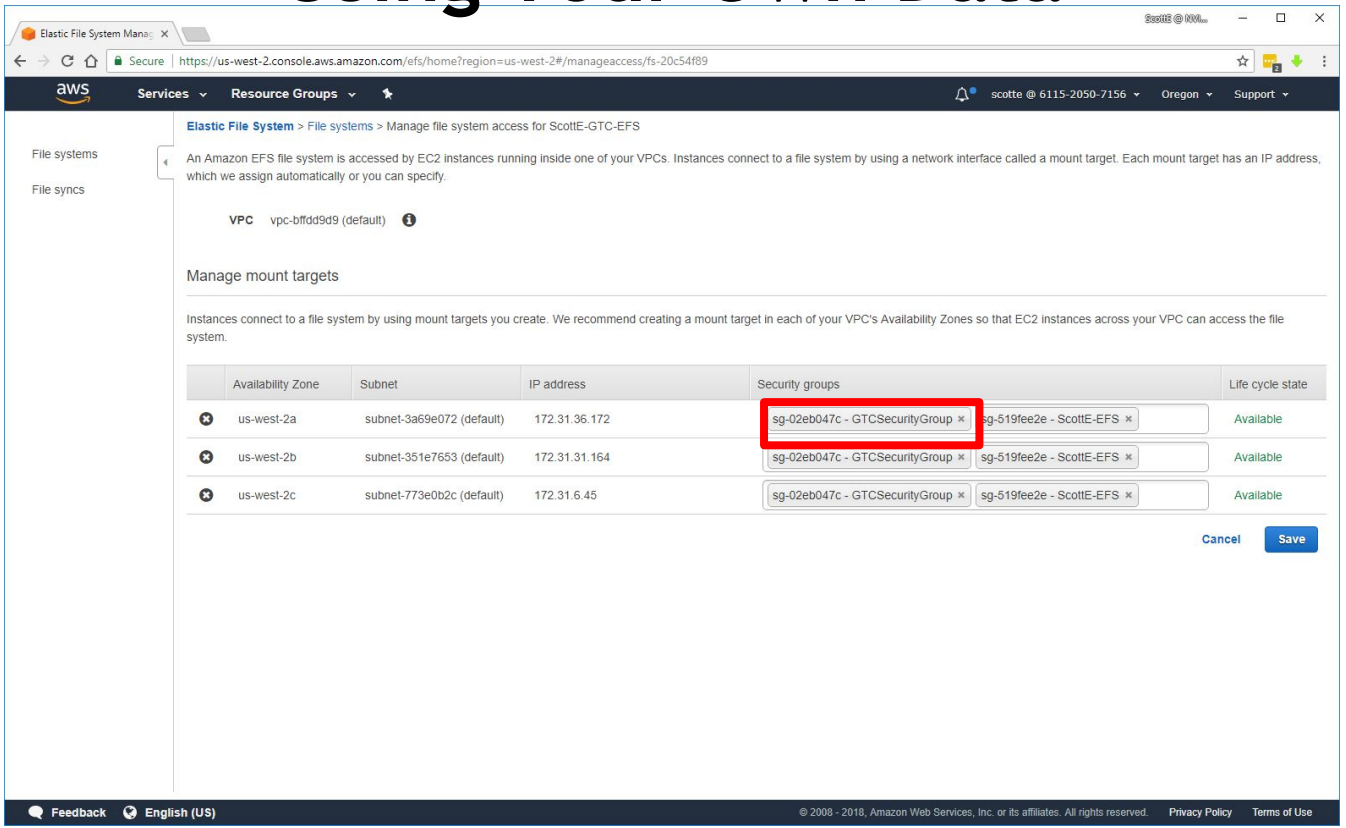
Using Your Own Data



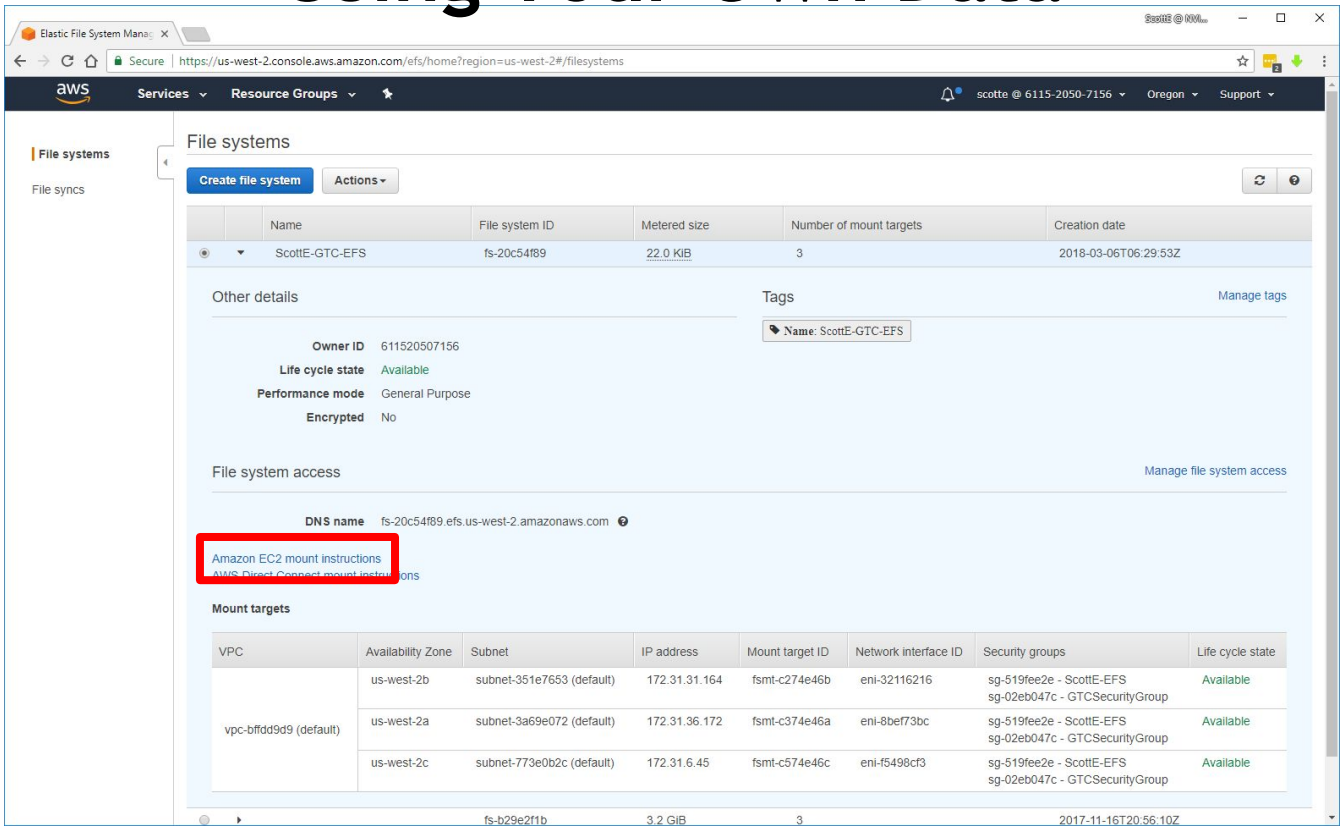
Using Your Own Data



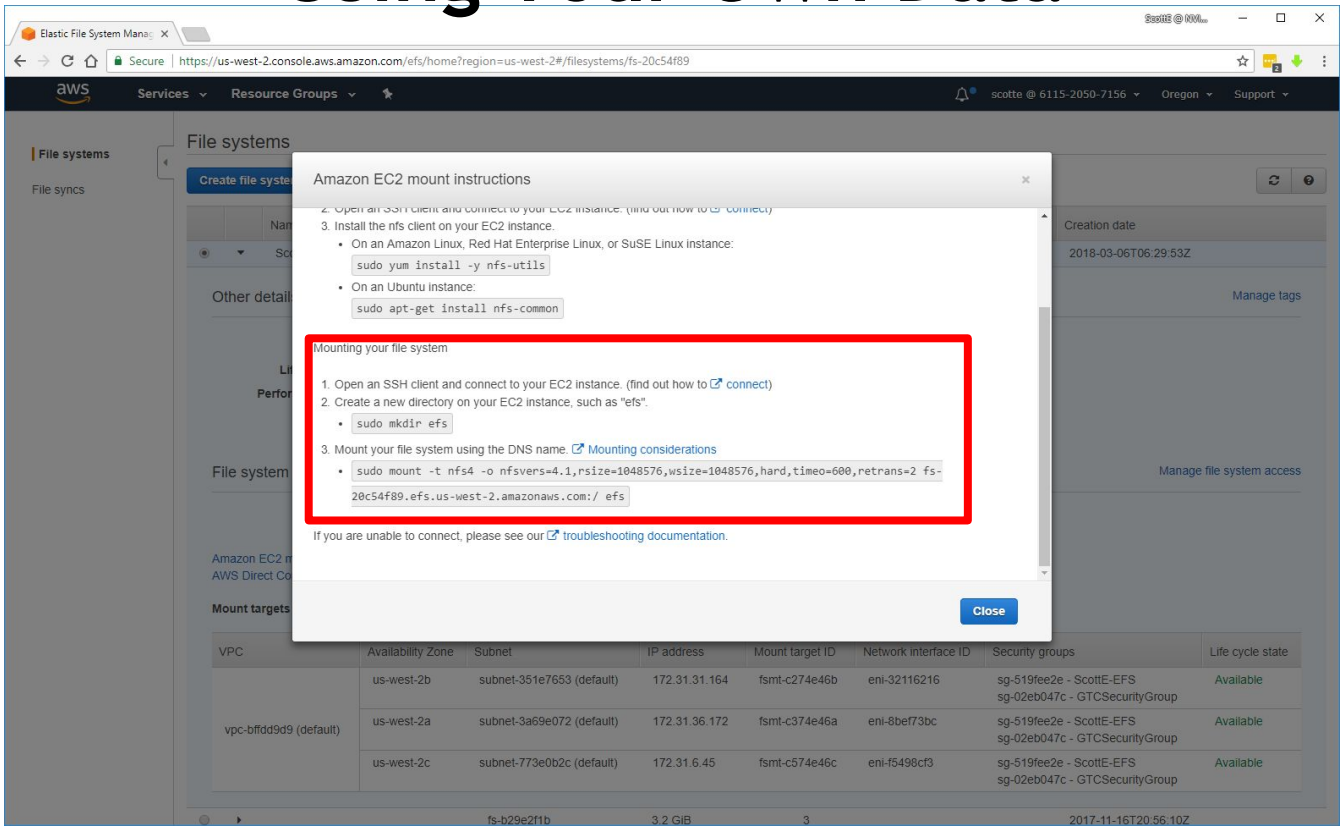
Using Your Own Data



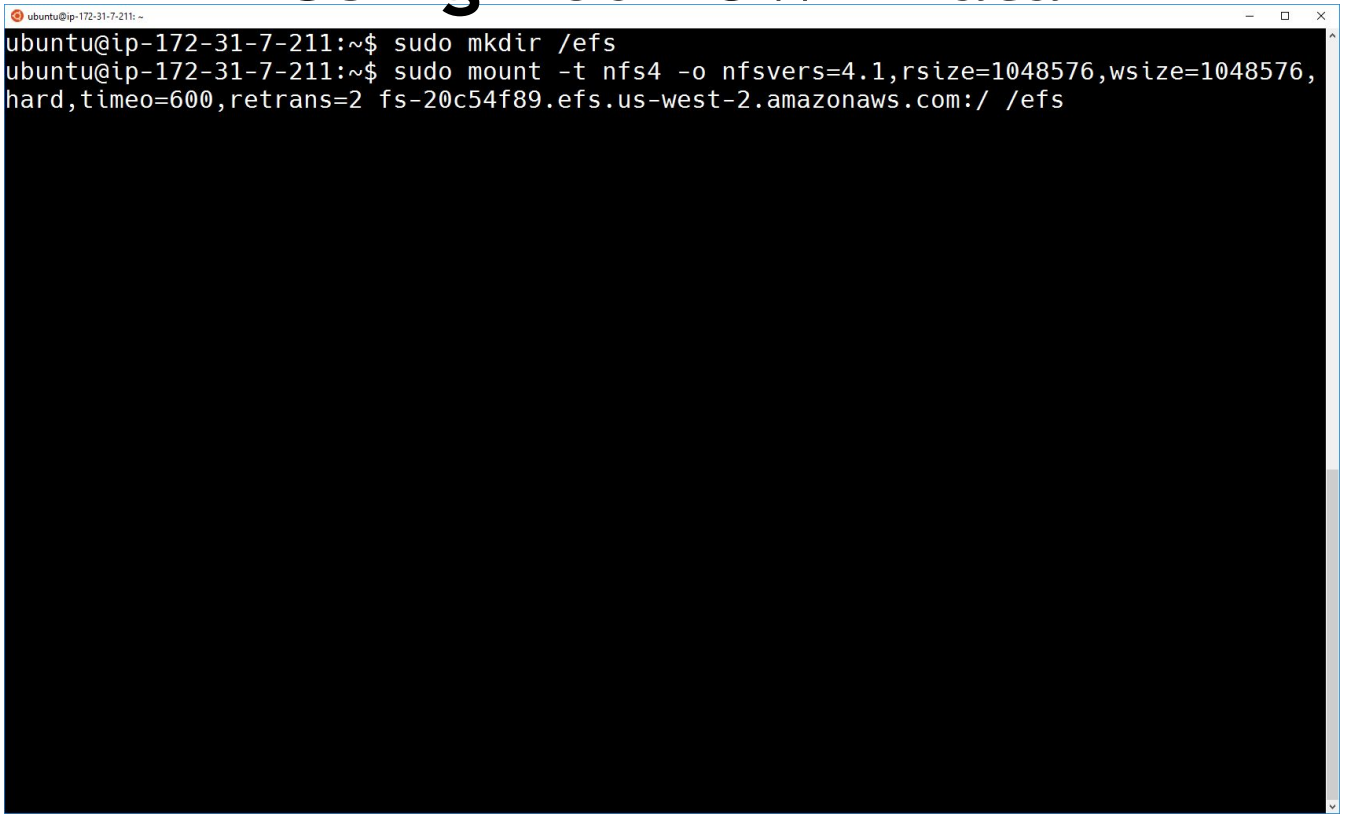
Using Your Own Data



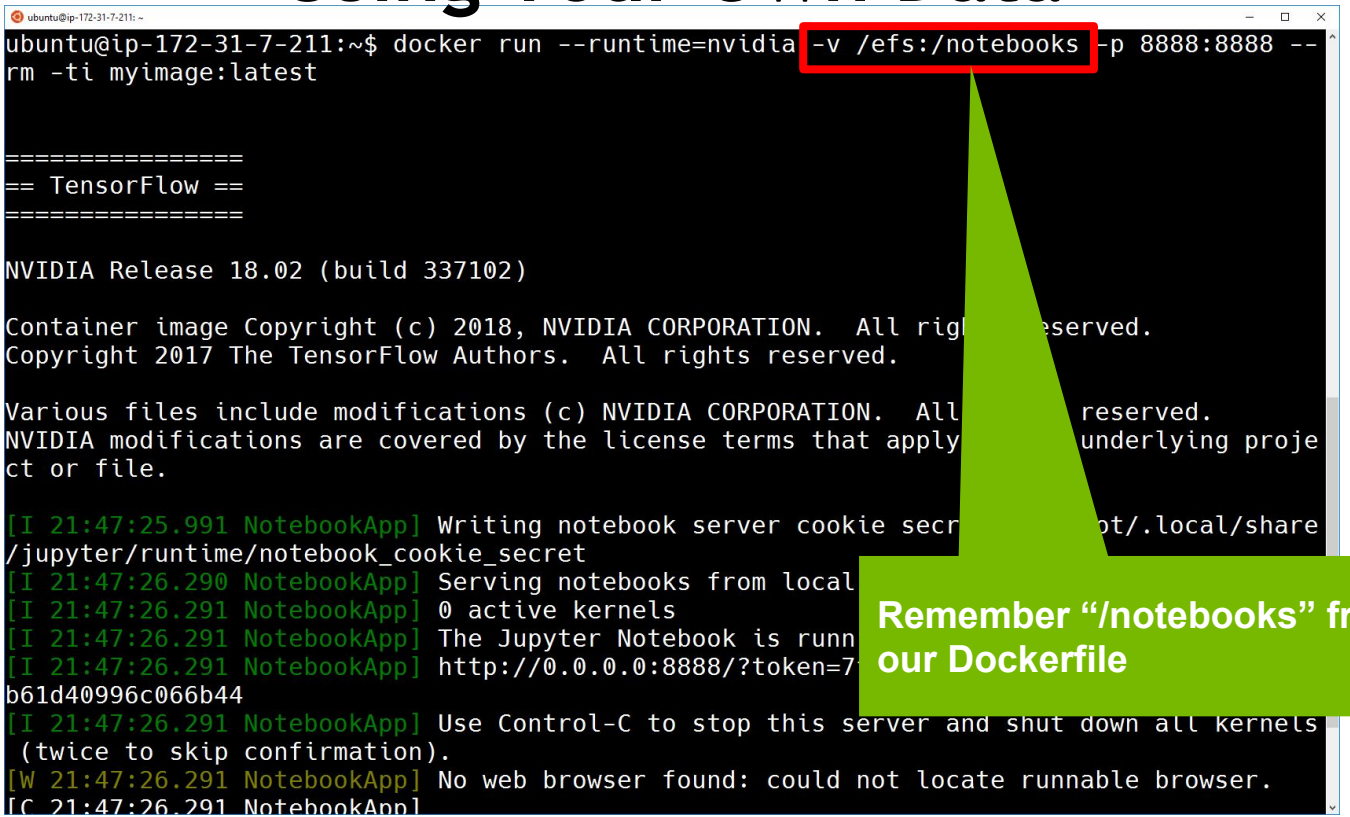
Using Your Own Data



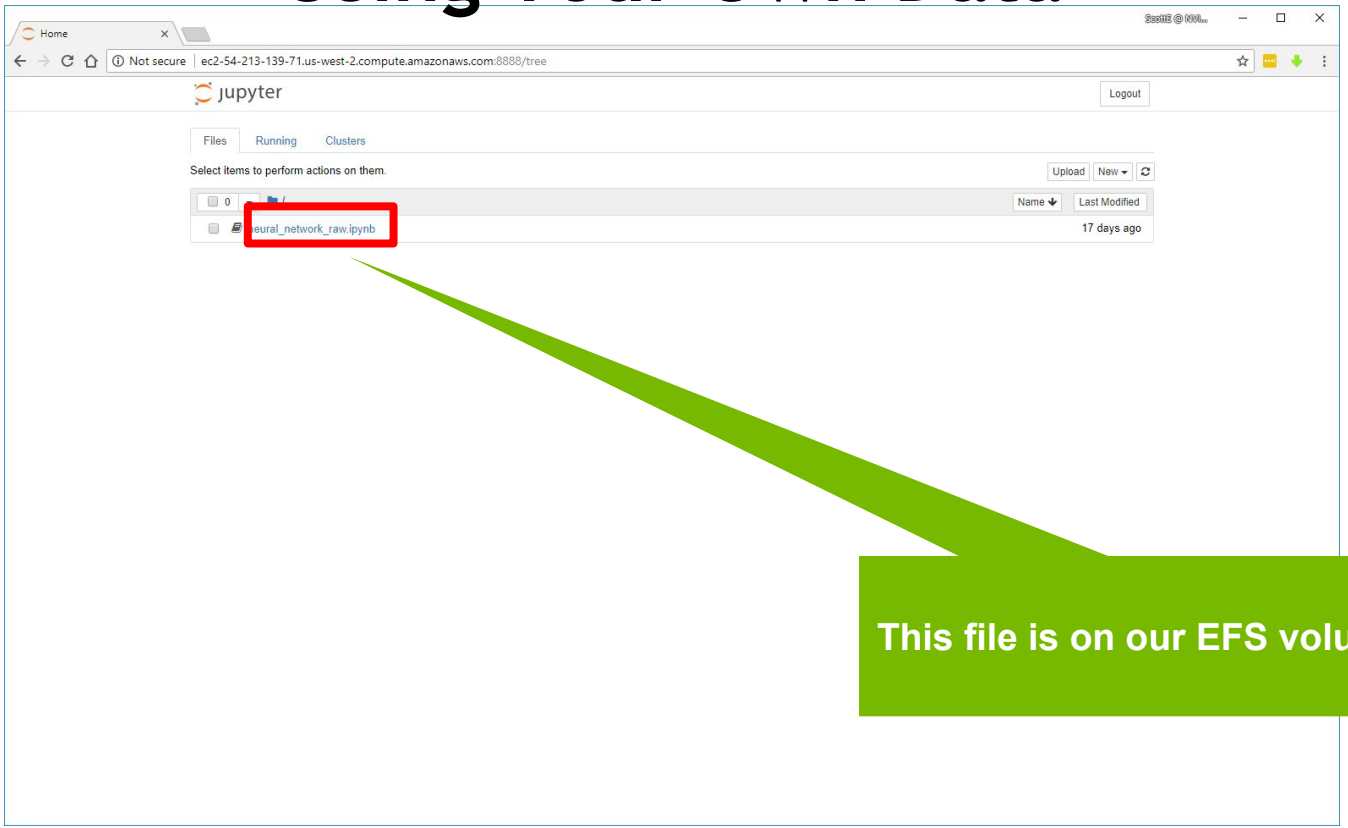
Using Your Own Data



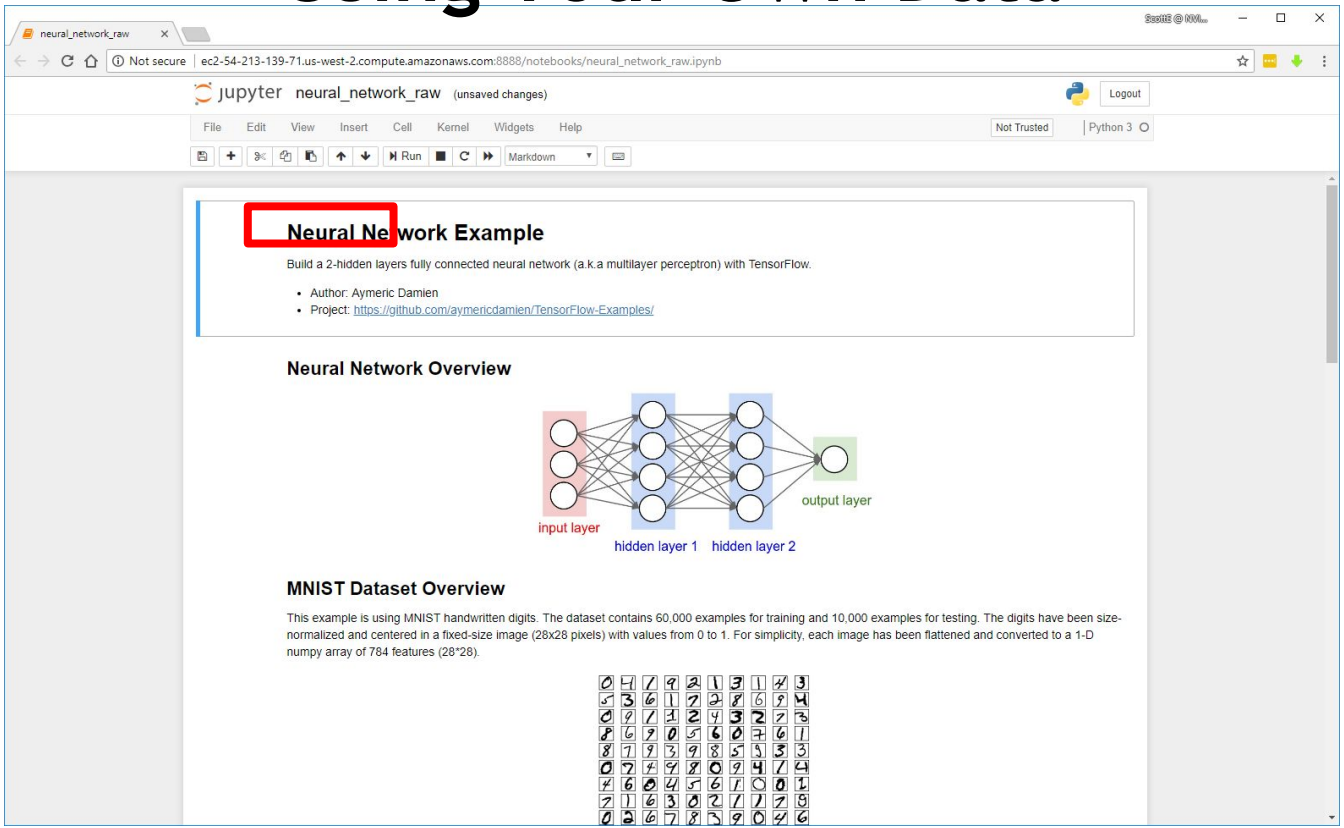
Using Your Own Data



Using Your Own Data



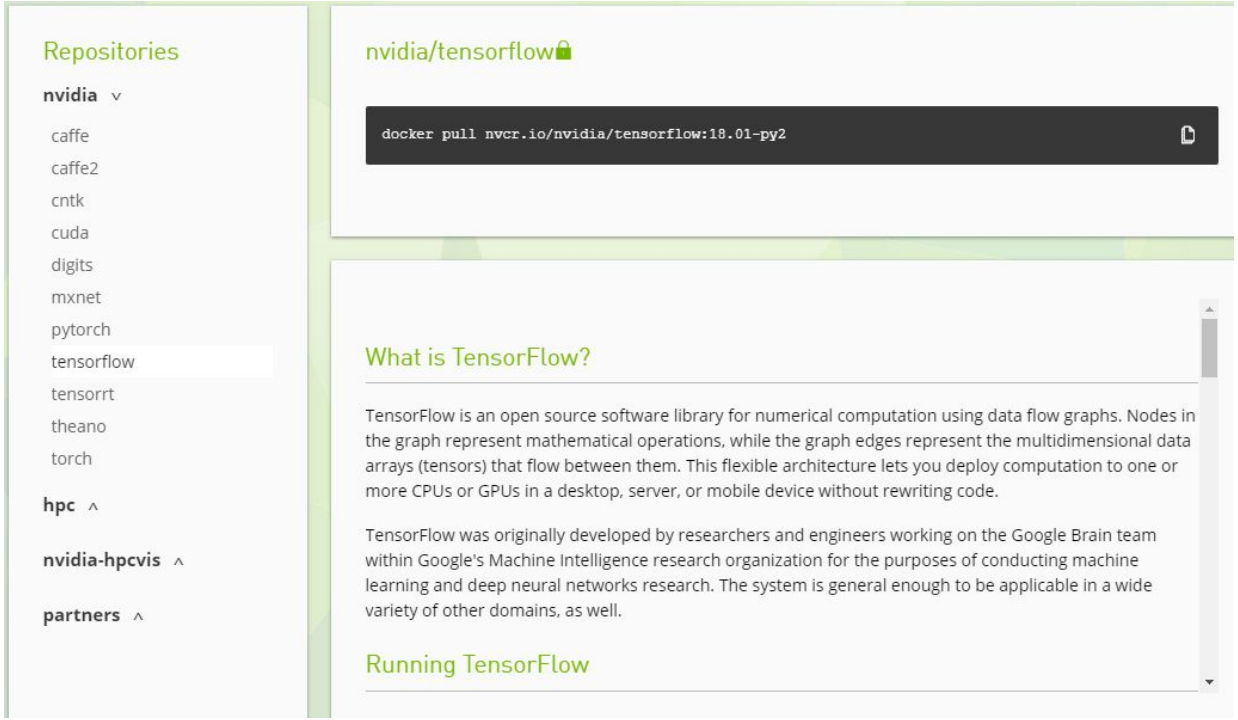
Using Your Own Data



Where do we go from here?

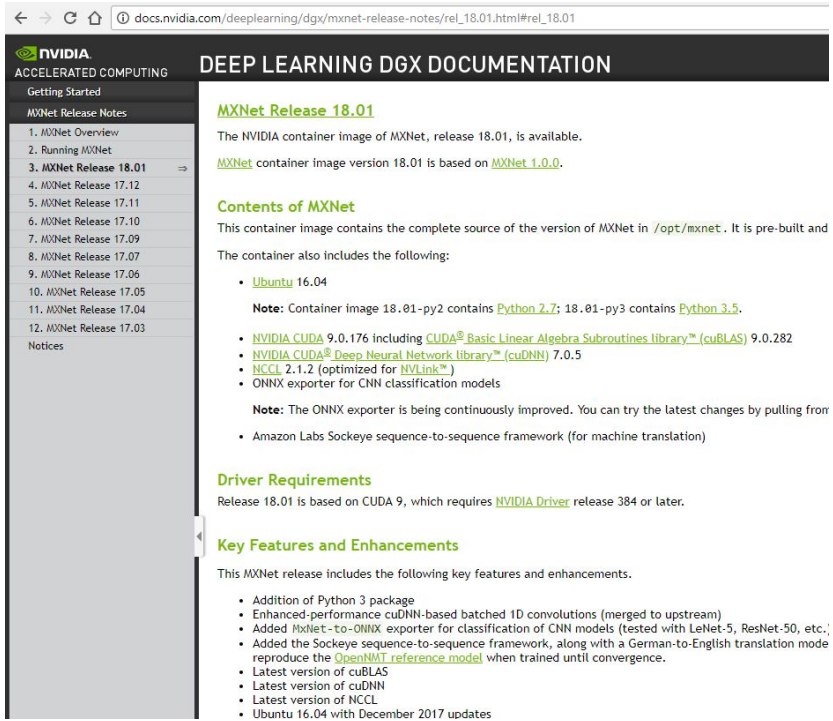
Framework Documentation

https://ngc.nvidia.com/



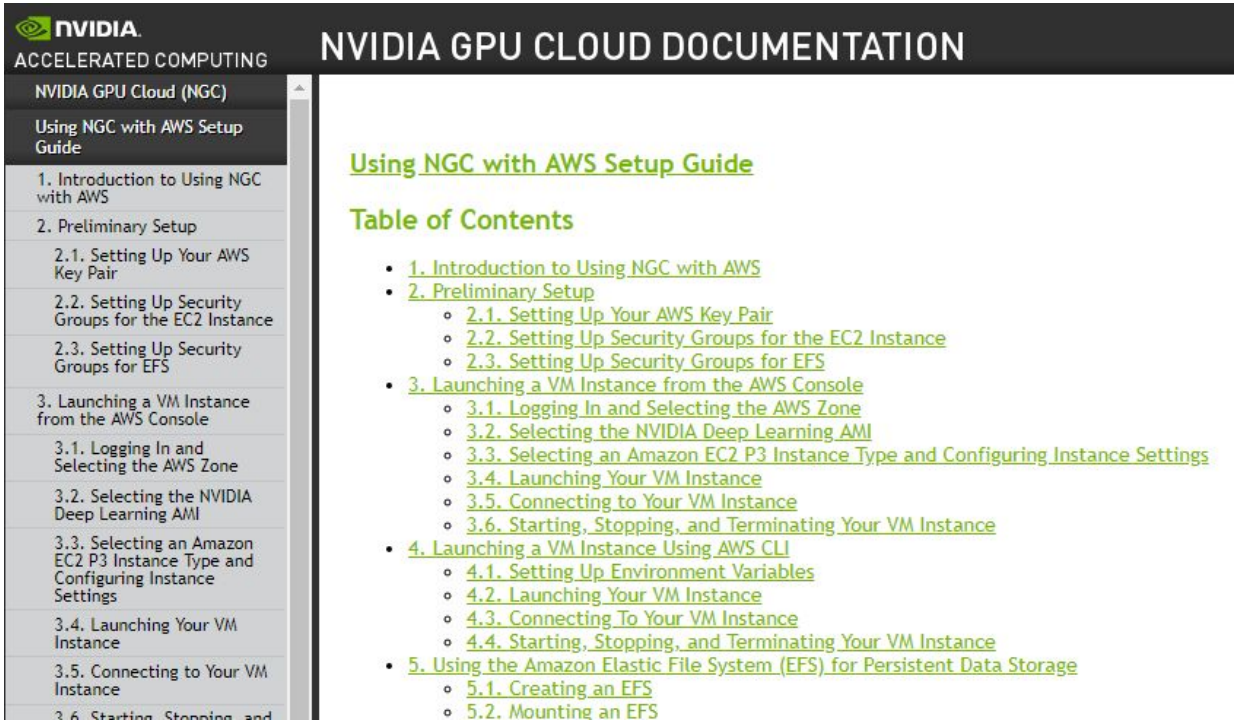
Container Documentation

http://docs.nvidia.com/deeplearning/dgx/



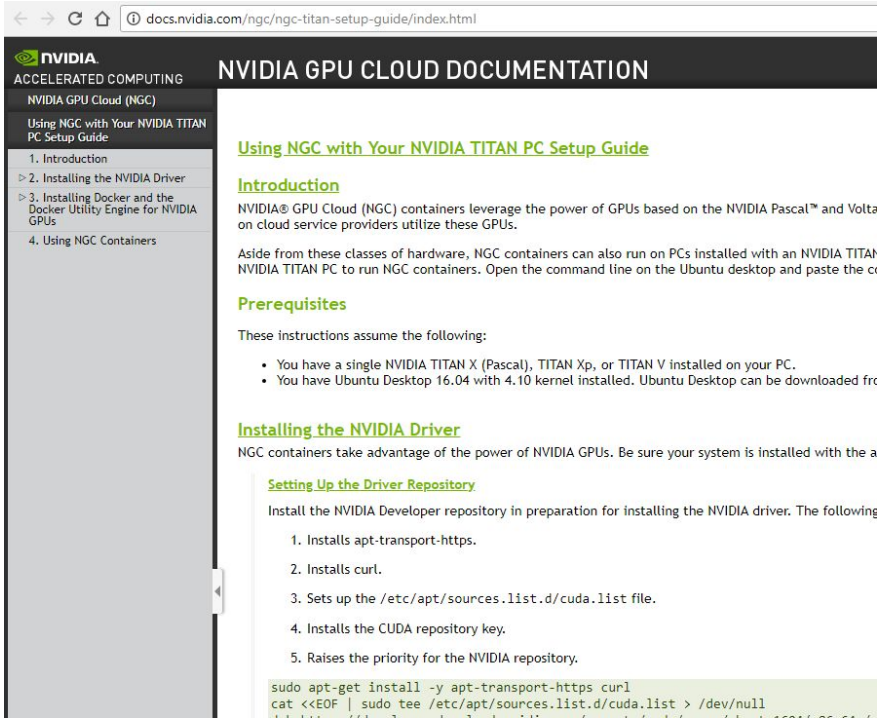
NGC + AWS

http://docs.nvidia.com/ngc/ngc-aws-setup-guide/



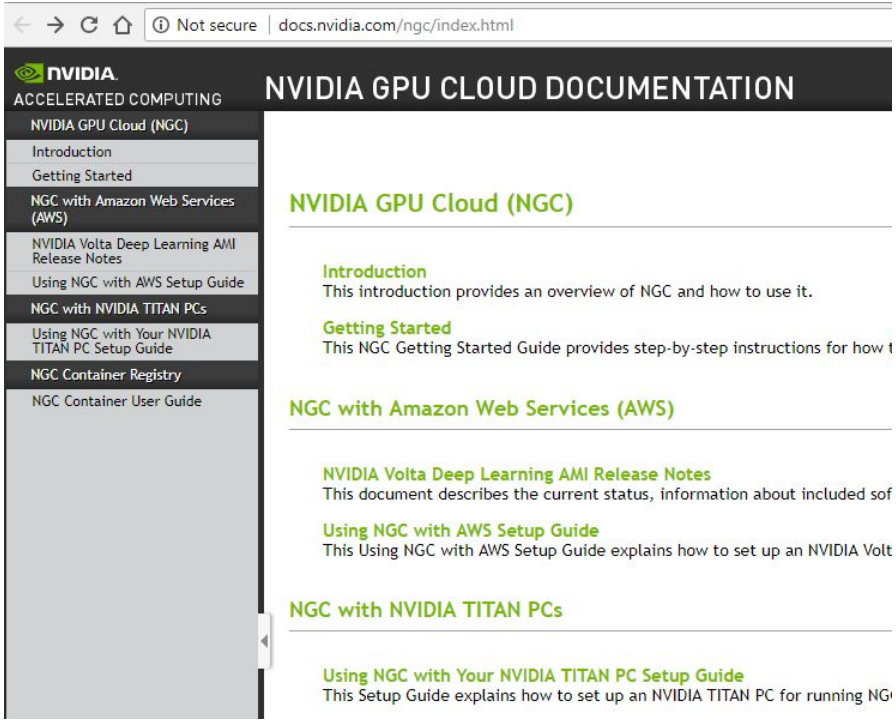
NGC + Real Hardware

http://docs.nvidia.com/ngc/ngc-titan-setup-guide/



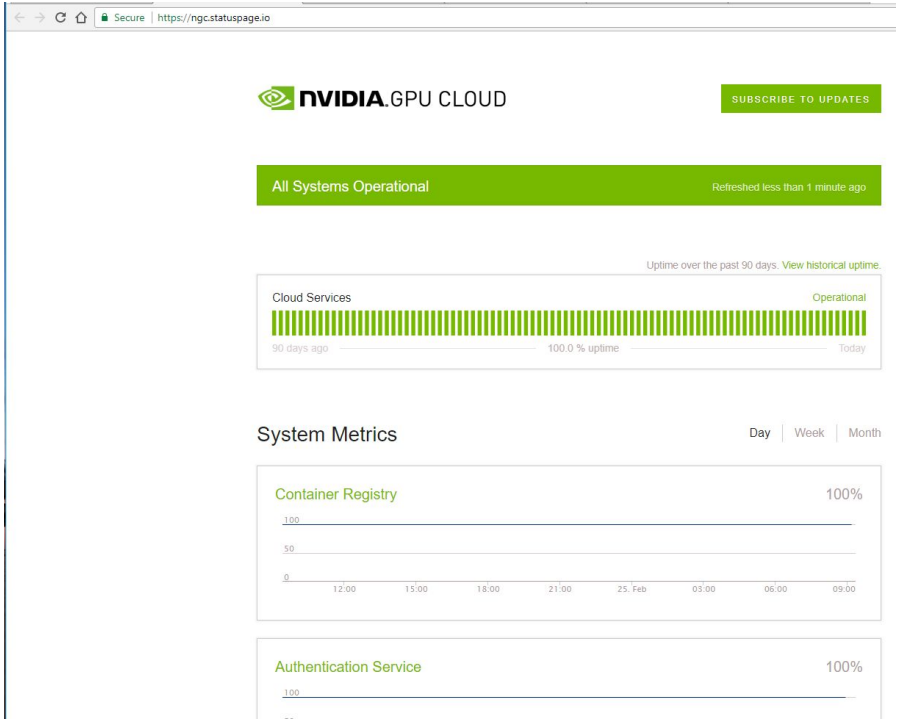
NGC Everywhere

http://docs.nvidia.com/ngc/



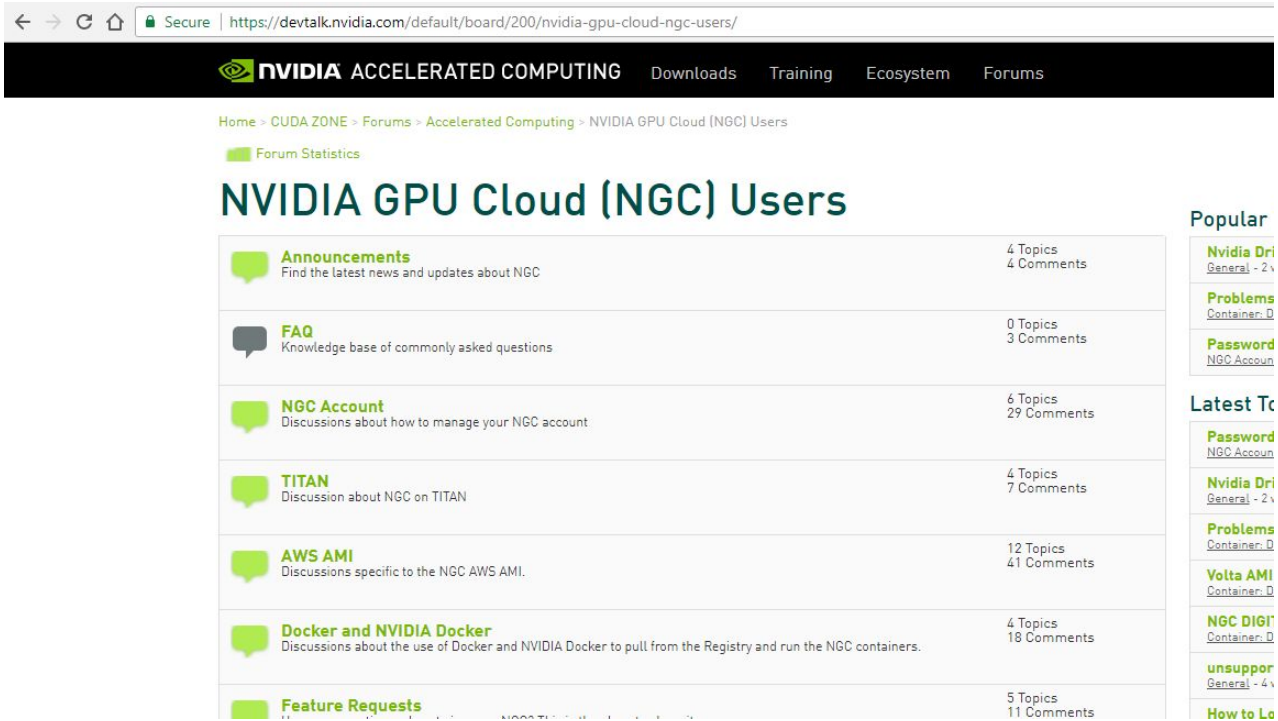
NGC Status

https://ngc.statuspage.io/



Help from Humans

https://devtalk.nvidia.com/



Thank You

Scott Ellis - scotte@nvidia.com
Jeff Weiss - jweiss@nvidia.com