TensorFlow GPU @AWS

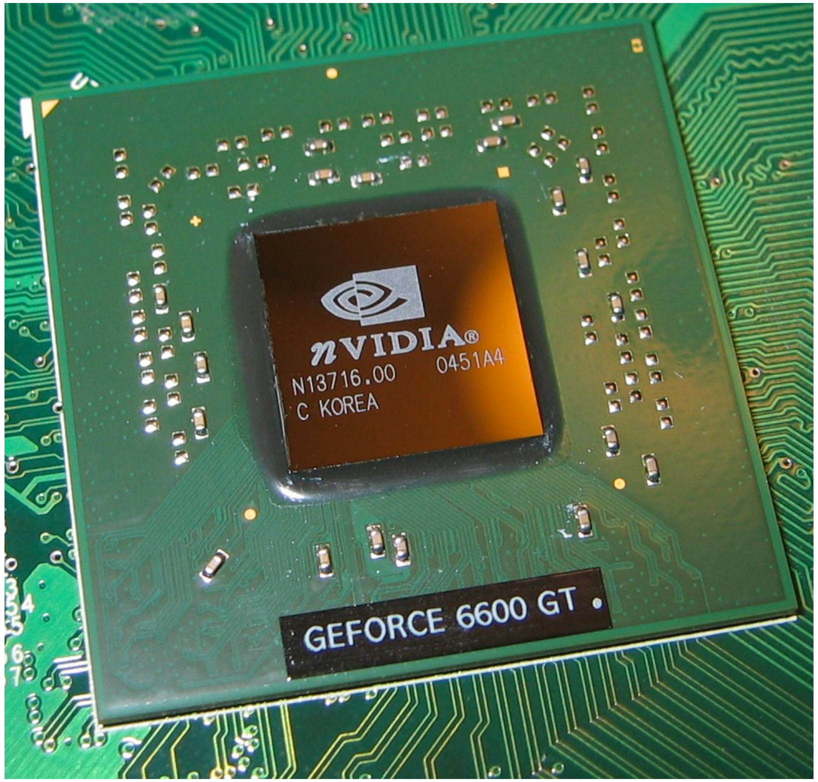
Sung Kim <hunkim+ml@gmail.com>
http://hunkim.github.io/ml/

Deep Network

- Takes a long time for training
    - Many forward/backward propagation and weight updates
    - Many metrics multiplications
- Very quick for testing and use in practice
    - One simple forward propagation

GPU

- A graphics processing unit (GPU), also occasionally called visual processing unit (VPU), is a specialized electronic circuit designed to rapidly manipulate and alter memory to accelerate the creation of images in a frame buffer intended for output to a display.



GPU version

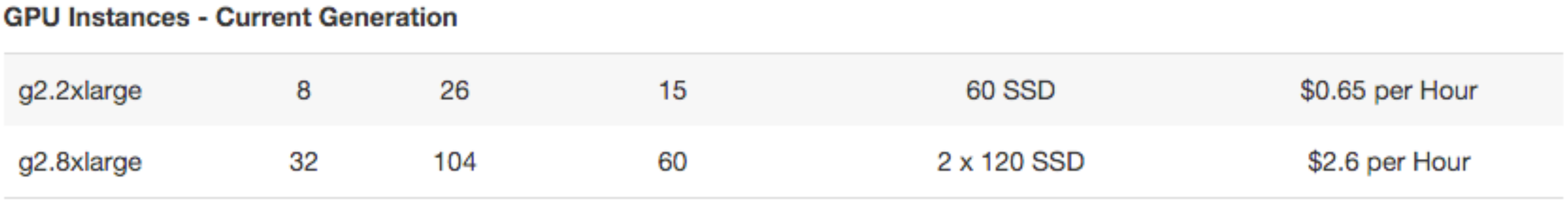
```
# Ubuntu/Linux 64-bit, GPU enabled, Python 2.7
# Requires CUDA toolkit 7.5 and CuDNN v4. For other versions, see "Install from sources" below.
$ export TF_BINARY_URL=https://storage.googleapis.com/tensorflow/linux/gpu/tensorflow-0.9.0rc0-cp27-none-linux_x86_64.whl

# Python 2
$ sudo pip install --upgrade $TF_BINARY_URL
```

http://solutionsreview.com/cloud-platforms/microsoft-beats-aws-google-on-cloud-storage-benchmark-test

amazon
web services™

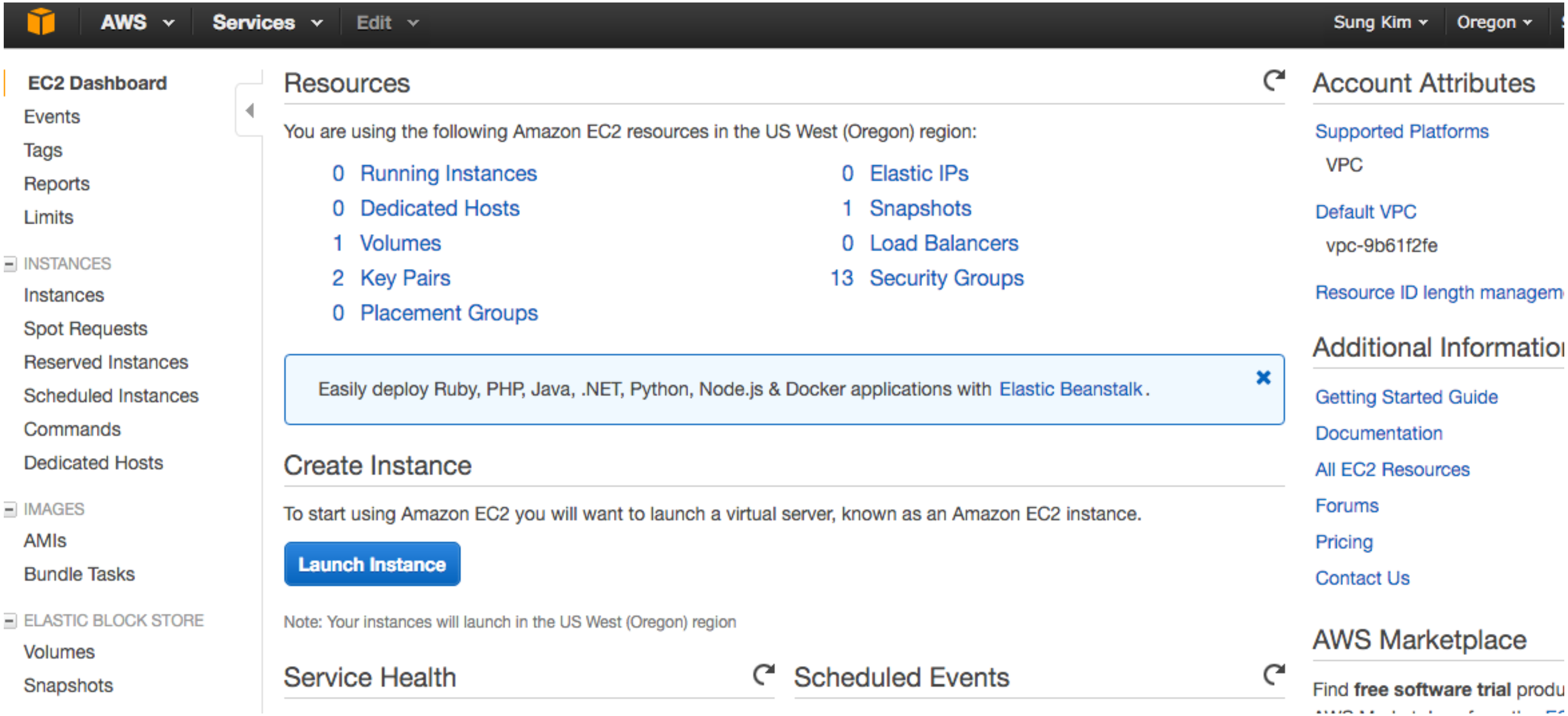
AWS GPU price in Oregon



GPU Instances - Current Generation

g2.2xlarge	8	26	15	60 SSD	$0.65 per Hour
g2.8xlarge	32	104	60	2 x 120 SSD	$2.6 per Hour

EC2 Console: Oregon



Resources

You are using the following Amazon EC2 resources in the US West (Oregon) region:

0	Running Instances	0	Elastic IPs
0	Dedicated Hosts	1	Snapshots
1	Volumes	0	Load Balancers
2	Key Pairs	13	Security Groups
0	Placement Groups		

Easily deploy Ruby, PHP, Java, .NET, Python, Node.js & Docker applications with Elastic Beanstalk.

Create Instance

To start using Amazon EC2 you will want to launch a virtual server, known as an Amazon EC2 instance.

Note: Your instances will launch in the US West (Oregon) region

Service Health

Scheduled Events

Account Attributes

Supported Platforms
VPC

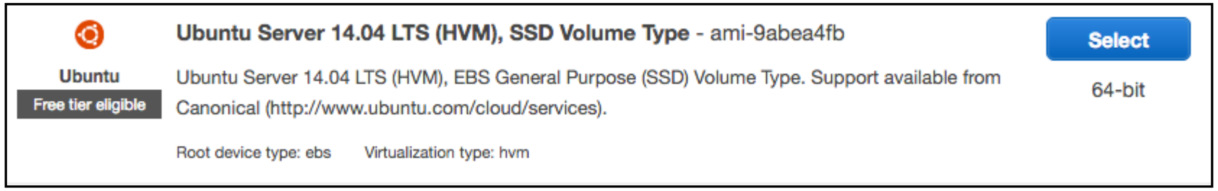
Default VPC
vpc-9b61f2fe

Resource ID length managem

Additional Informatio

AWS Marketplace

Find **free software trial** produ

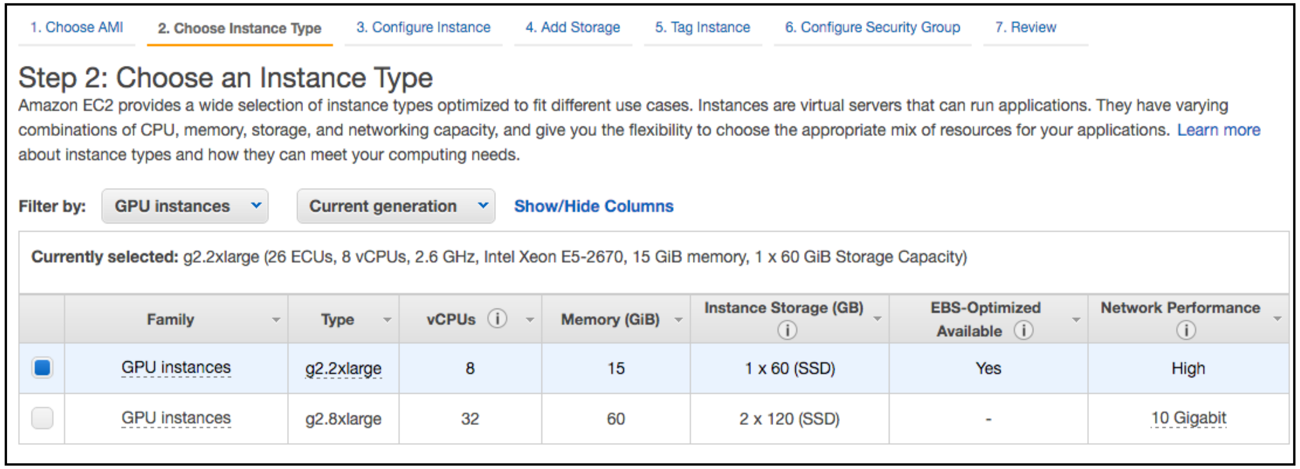


Ubuntu Server 14.04 LTS (HVM), SSD Volume Type - ami-9abea4fb

Ubuntu Server 14.04 LTS (HVM), EBS General Purpose (SSD) Volume Type. Support available from Canonical (http://www.ubuntu.com/cloud/services).

Root device type: ebs   Virtualization type: hvm

64-bit



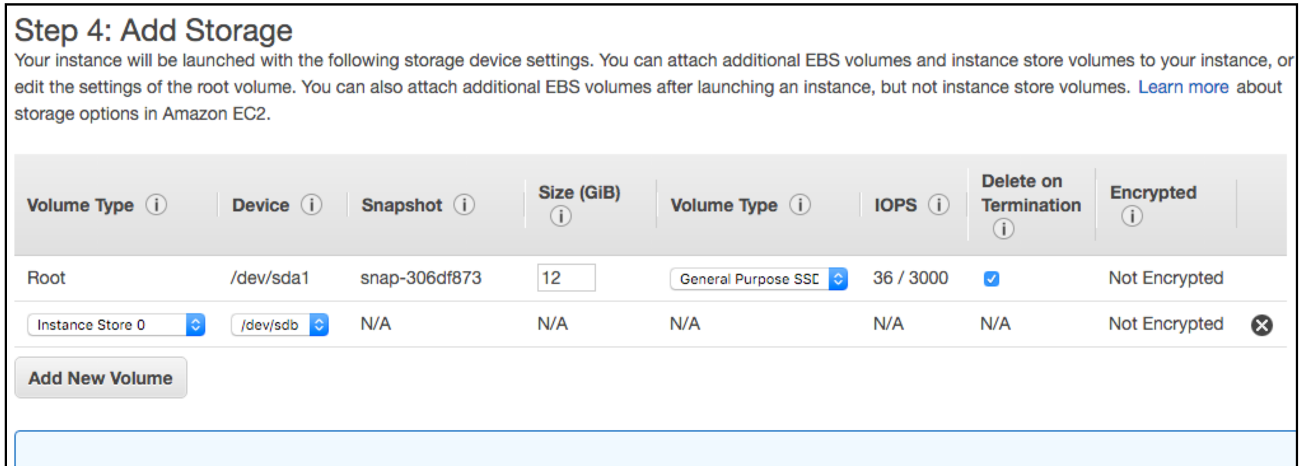
Step 2: Choose an Instance Type

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 computing needs.

Filter by: GPU instances   Current generation   **Show/Hide Columns**

Currently selected: g2.2xlarge (26 ECUs, 8 vCPUs, 2.6 GHz, Intel Xeon E5-2670, 15 GiB memory, 1 x 60 GiB Storage Capacity)

	Family	Type	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance
☑	GPU instances	g2.2xlarge	8	15	1 x 60 (SSD)	Yes	High
□	GPU instances	g2.8xlarge	32	60	2 x 120 (SSD)	-	10 Gigabit



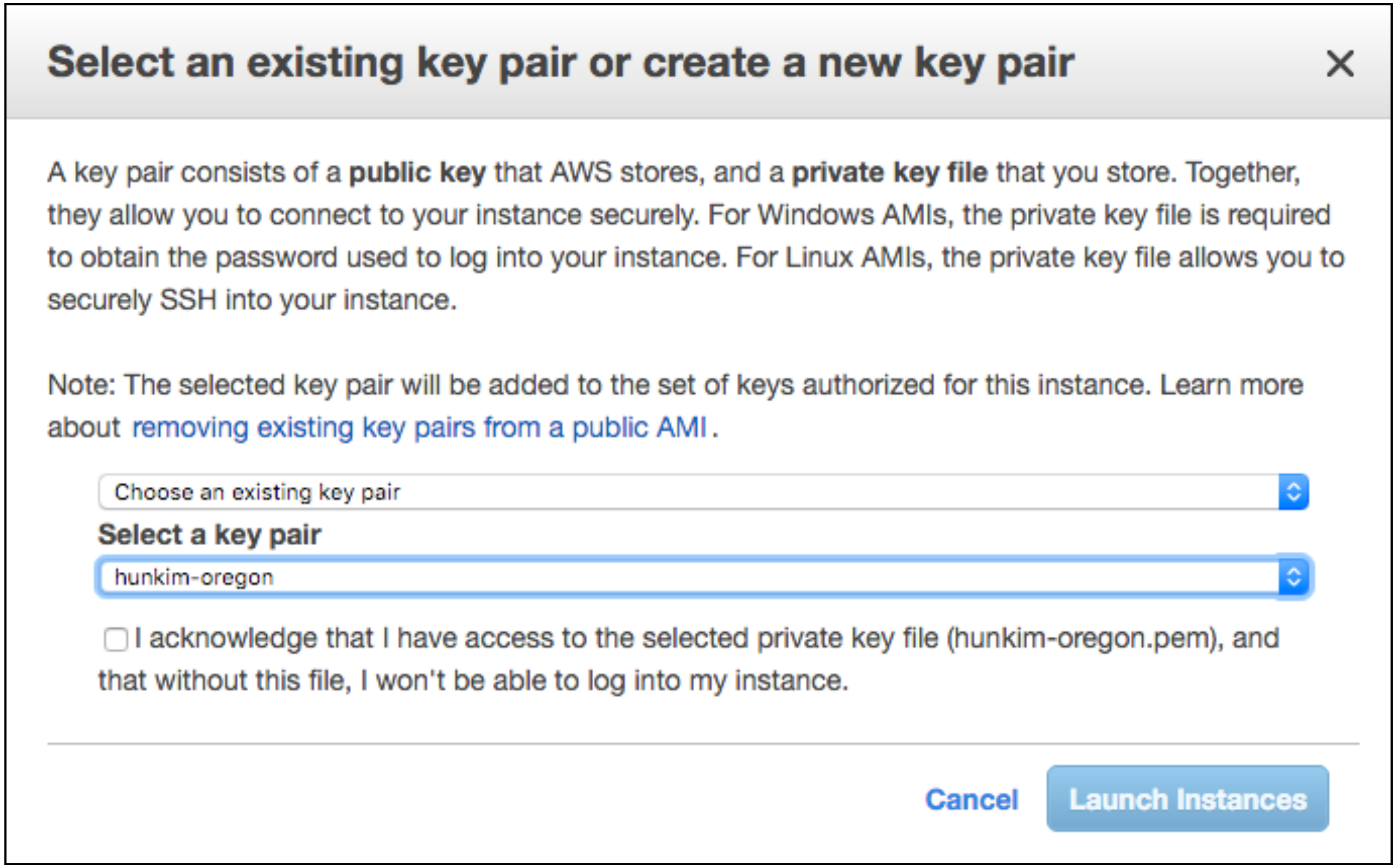
Step 4: Add Storage

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	Delete on Termination	Encrypted	
Root	/dev/sda1	snap-306df873	12	General Purpose SSD	36 / 3000	☑	Not Encrypted	
Instance Store 0	/dev/sdb	N/A	N/A	N/A	N/A	N/A	Not Encrypted	⊗

ubuntu,
GPU,
12G or more

key to access the server



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 .

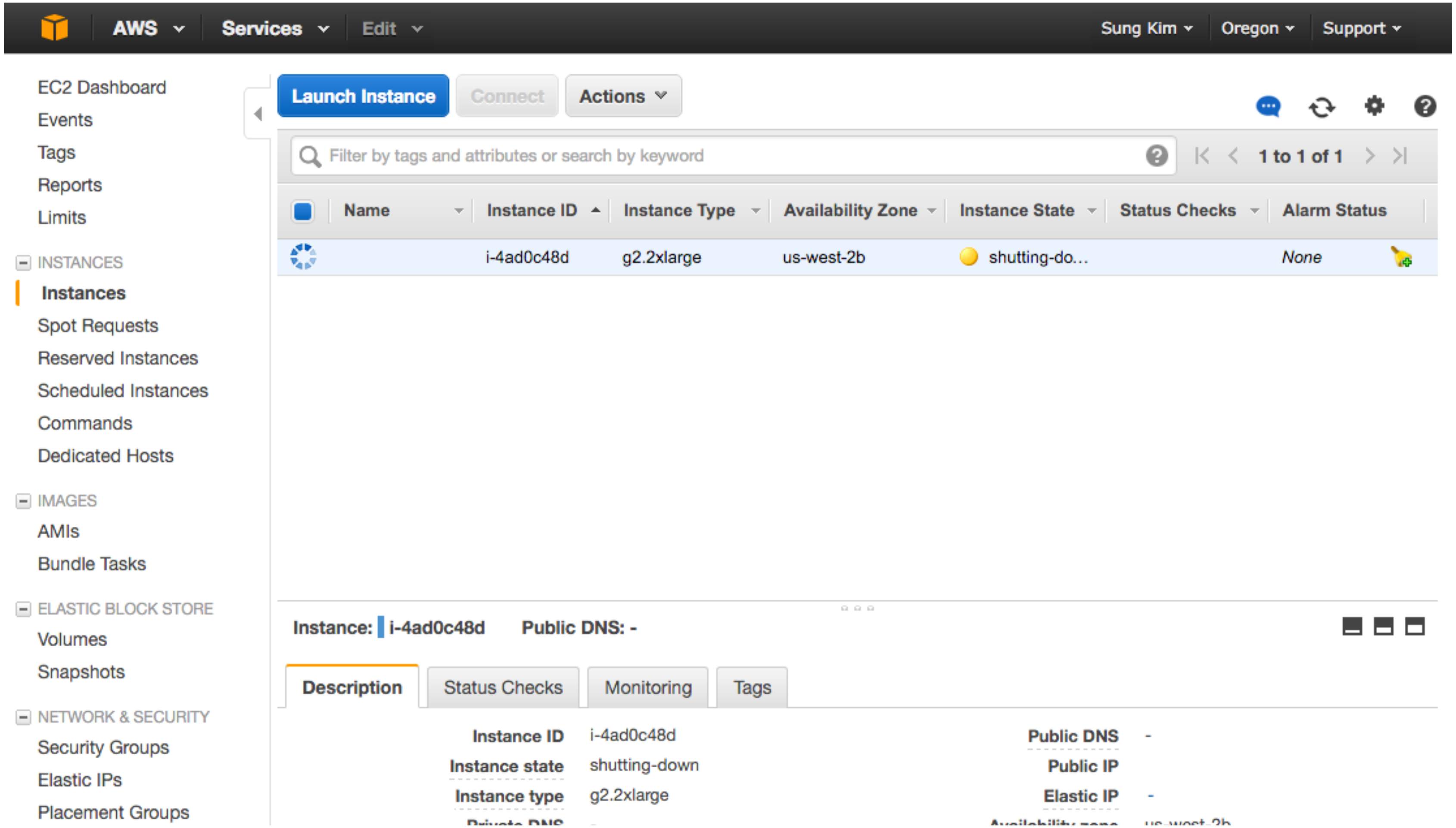
Choose an existing key pair

Select a key pair

hunkim-oregon

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

EC2: Create an instance

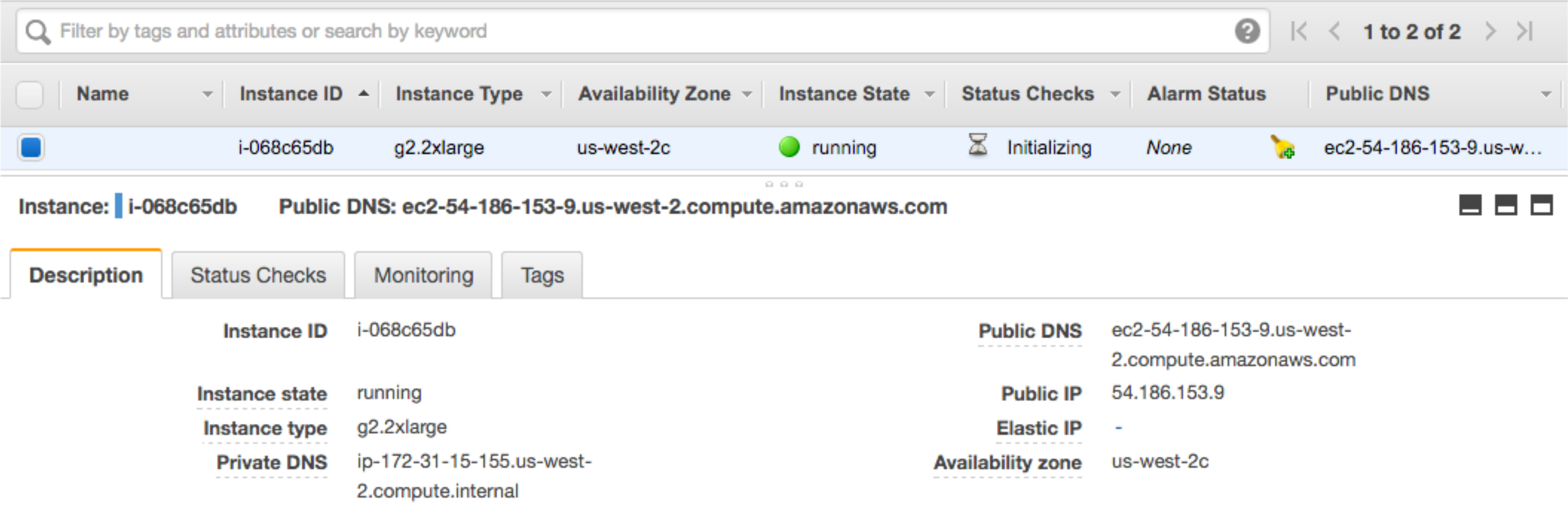


Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status
	i-4ad0c48d	g2.2xlarge	us-west-2b	● shutting-do...		*None*

Instance: i-4ad0c48d Public DNS: -

Instance ID	i-4ad0c48d	Public DNS	-
Instance state	shutting-down	Public IP	
Instance type	g2.2xlarge	Elastic IP	-
Private DNS	-	Availability zone	us-west-2b

It’s ready to ssh!



Filter by tags and attributes or search by keyword

1 to 2 of 2

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS
	i-068c65db	g2.2xlarge	us-west-2c	● running	Initializing	*None*	ec2-54-186-153-9.us-w...

Instance: i-068c65db   Public DNS: ec2-54-186-153-9.us-west-2.compute.amazonaws.com

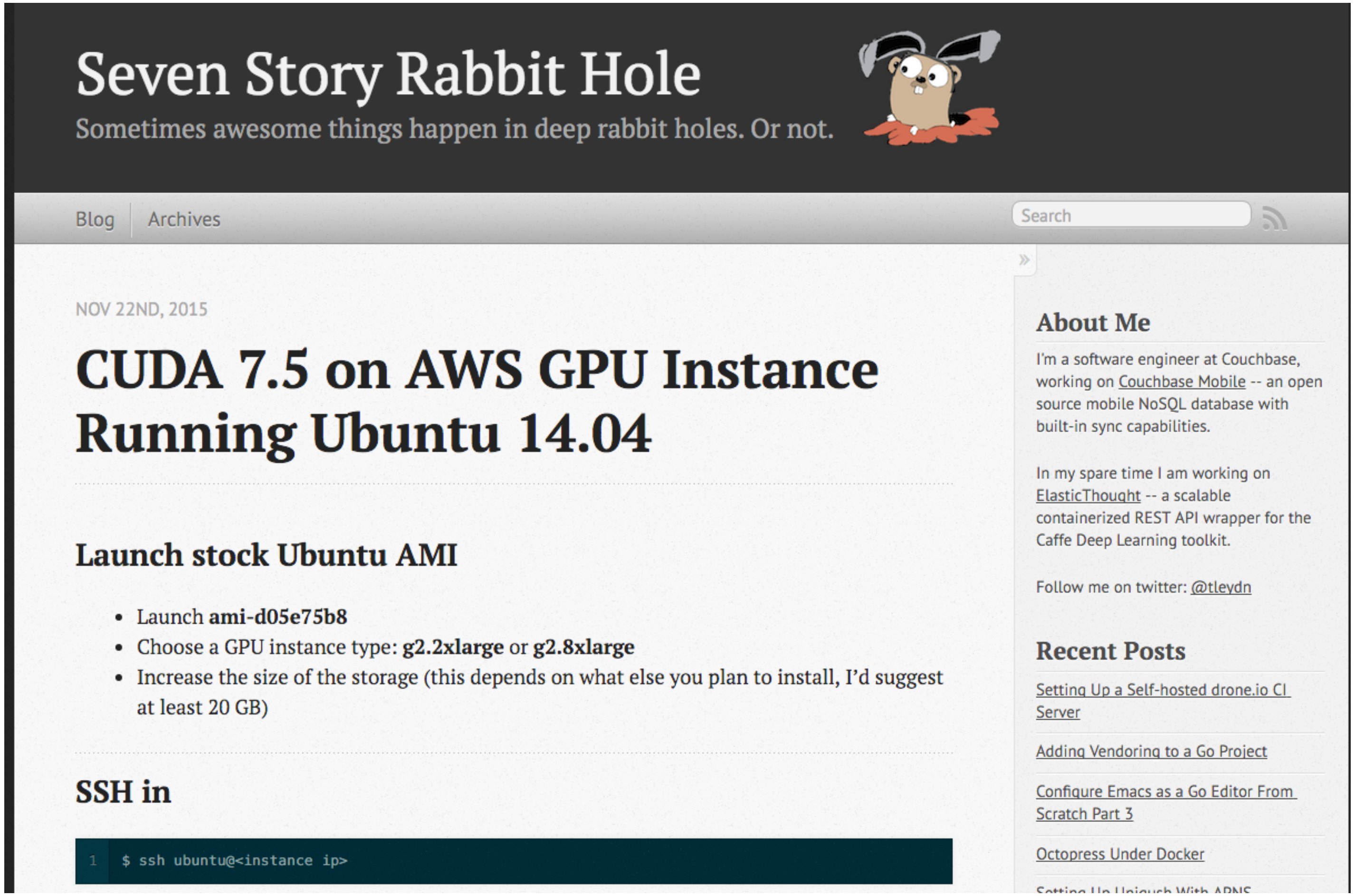
Instance ID	i-068c65db	**Public DNS**	ec2-54-186-153-9.us-west-2.compute.amazonaws.com
Instance state	running	**Public IP**	54.186.153.9
Instance type	g2.2xlarge	**Elastic IP**	-
Private DNS	ip-172-31-15-155.us-west-2.compute.internal	**Availability zone**	us-west-2c

Requires CUDA and CuDNN

```
# Ubuntu/Linux 64-bit, GPU enabled, Python 2.7
# Requires CUDA toolkit 7.5 and CuDNN v4. For other versions, see "Install from sources" below.
$ export TF_BINARY_URL=https://storage.googleapis.com/tensorflow/linux/gpu/tensorflow-0.9.0rc0-cp27-none-linux_x86_64.whl

# Python 2
$ sudo pip install --upgrade $TF_BINARY_URL
```

CUDA



Seven Story Rabbit Hole

Sometimes awesome things happen in deep rabbit holes. Or not.

NOV 22ND, 2015

CUDA 7.5 on AWS GPU Instance Running Ubuntu 14.04

Launch stock Ubuntu AMI

- Launch **ami-d05e75b8**
- Choose a GPU instance type: **g2.2xlarge** or **g2.8xlarge**
- Increase the size of the storage (this depends on what else you plan to install, I'd suggest at least 20 GB)

SSH in

```
$ ssh ubuntu@<instance ip>
```

About Me

I'm a software engineer at Couchbase, working on Couchbase Mobile -- an open source mobile NoSQL database with built-in sync capabilities.

In my spare time I am working on ElasticThought -- a scalable containerized REST API wrapper for the Caffe Deep Learning toolkit.

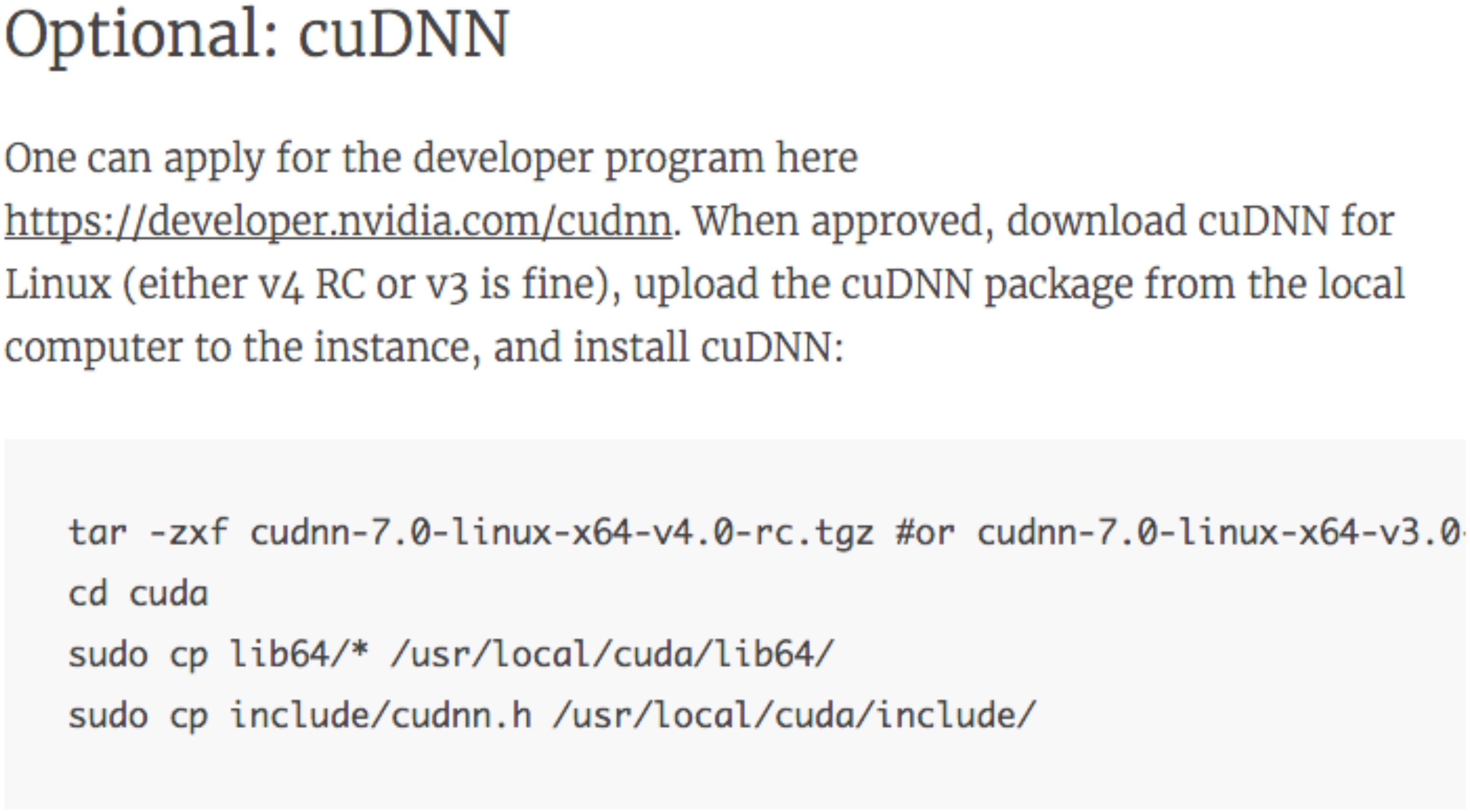
Follow me on twitter: @tleydn

Recent Posts

- Setting Up a Self-hosted drone.io CI Server
- Adding Vendoring to a Go Project
- Configure Emacs as a Go Editor From Scratch Part 3
- Octopress Under Docker

http://tleyden.github.io/blog/2015/11/22/cuda-7-dot-5-on-aws-gpu-instance-running-ubuntu-14-dot-04/

cuDNN



Optional: cuDNN

One can apply for the developer program here https://developer.nvidia.com/cudnn. When approved, download cuDNN for Linux (either v4 RC or v3 is fine), upload the cuDNN package from the local computer to the instance, and install cuDNN:

```
tar -zxf cudnn-7.0-linux-x64-v4.0-rc.tgz #or cudnn-7.0-linux-x64-v3.0
cd cuda
sudo cp lib64/* /usr/local/cuda/lib64/
sudo cp include/cudnn.h /usr/local/cuda/include/
```

https://no2147483647.wordpress.com/2016/01/16/setup-amazon-aws-gpu-instance-with-mxnet

16 commends

```
1 wget http://developer.download.nvidia.com/…/cuda-repo-ubuntu1404…
2 sudo dpkg -i cuda-repo-ubuntu1404_7.5-18_amd64.deb
3 sudo apt-get update
4 sudo apt-get upgrade -y
5 sudo apt-get install -y opencl-headers build-essential protobuf-compiler libprotoc-dev libboost-all-dev
libleveldb-dev hdf5-tools libhdf5-serial-dev libopencv-core-dev libopencv-highgui-dev libsnappy-dev
libsnappy1 libatlas-base-dev cmake libstdc++6-4.8-dbg libgoogle-glog0 libgoogle-glog-dev libgflags-dev
liblmdb-dev git python-pip gfortran
6 sudo apt-get clean
7 sudo apt-get install -y linux-image-extra-`uname -r` linux-headers-`uname -r` linux-image-`uname -r`
8 sudo apt-get install -y cuda
9 nvidia-smi
10 sudo apt-get install python-pip python-dev
11 sudo pip install --upgrade https://storage.googleapis.com/…/tensorflow-0.8.0rc0-cp27-n…
12 git clone https://github.com/nlintz/TensorFlow-Tutorials
13 cd TensorFlow-Tutorials/
14 vi ~/.profile # add PATH, LD PATH
15 source ~/.profile
16 python 06_autoencoder.py
```

Add Path

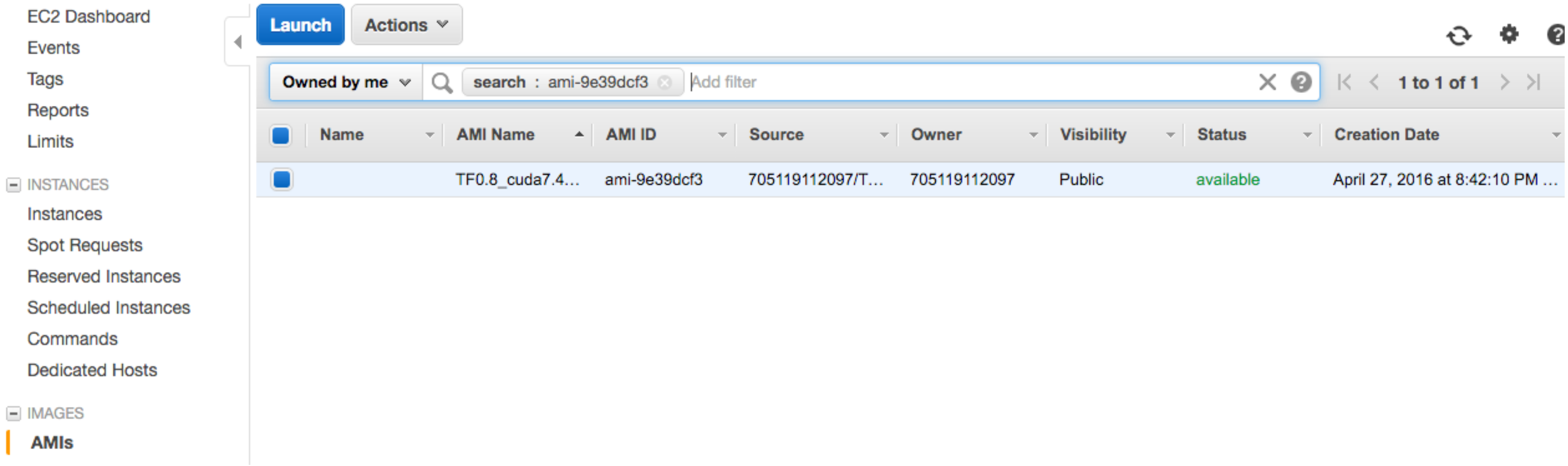
```
export PATH=/usr/local/cuda/bin:$PATH
export LD_LIBRARY_PATH=/usr/local/cuda/lib64:$LD_LIBRARY_PATH
```

Requires CUDA and CuDNN

```
# Ubuntu/Linux 64-bit, GPU enabled, Python 2.7
# Requires CUDA toolkit 7.5 and CuDNN v4. For other versions, see "Install from sources" below.
$ export TF_BINARY_URL=https://storage.googleapis.com/tensorflow/linux/gpu/tensorflow-0.9.0rc0-cp27-none-linux_x86_64.whl

# Python 2
$ sudo pip install --upgrade $TF_BINARY_URL
```

Reuse ami-9e39dcf3 (N.Virginia)
ami-38f60658 (oregon)

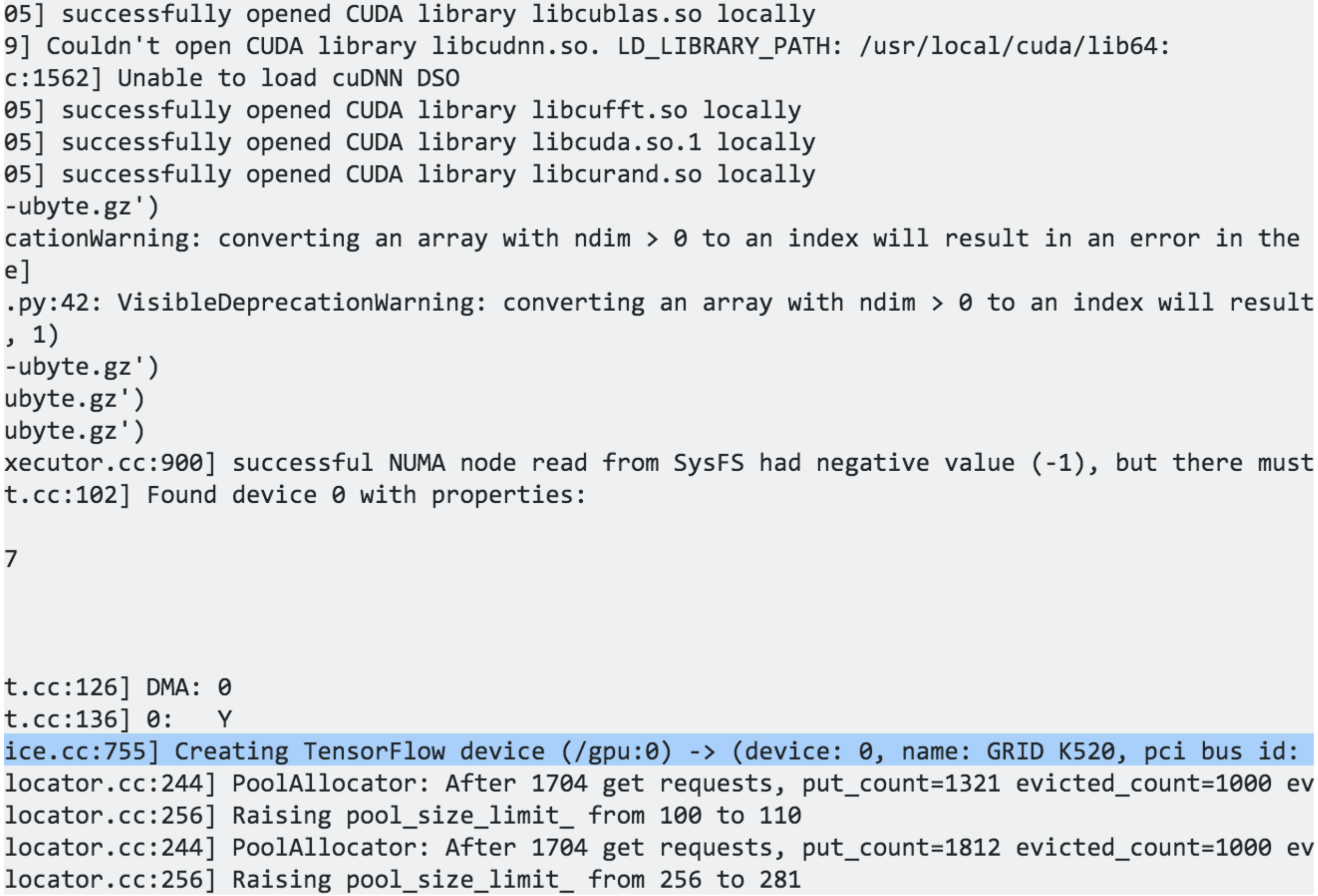


search : ami-9e39dcf3

■	Name	AMI Name	AMI ID	Source	Owner	Visibility	Status	Creation Date
■		TF0.8_cuda7.4...	ami-9e39dcf3	705119112097/T...	705119112097	Public	available	April 27, 2016 at 8:42:10 PM ...

Perhaps, it will be unavailable when later CUDA versions are out.

Creating TensorFlow device (/gpu:0)

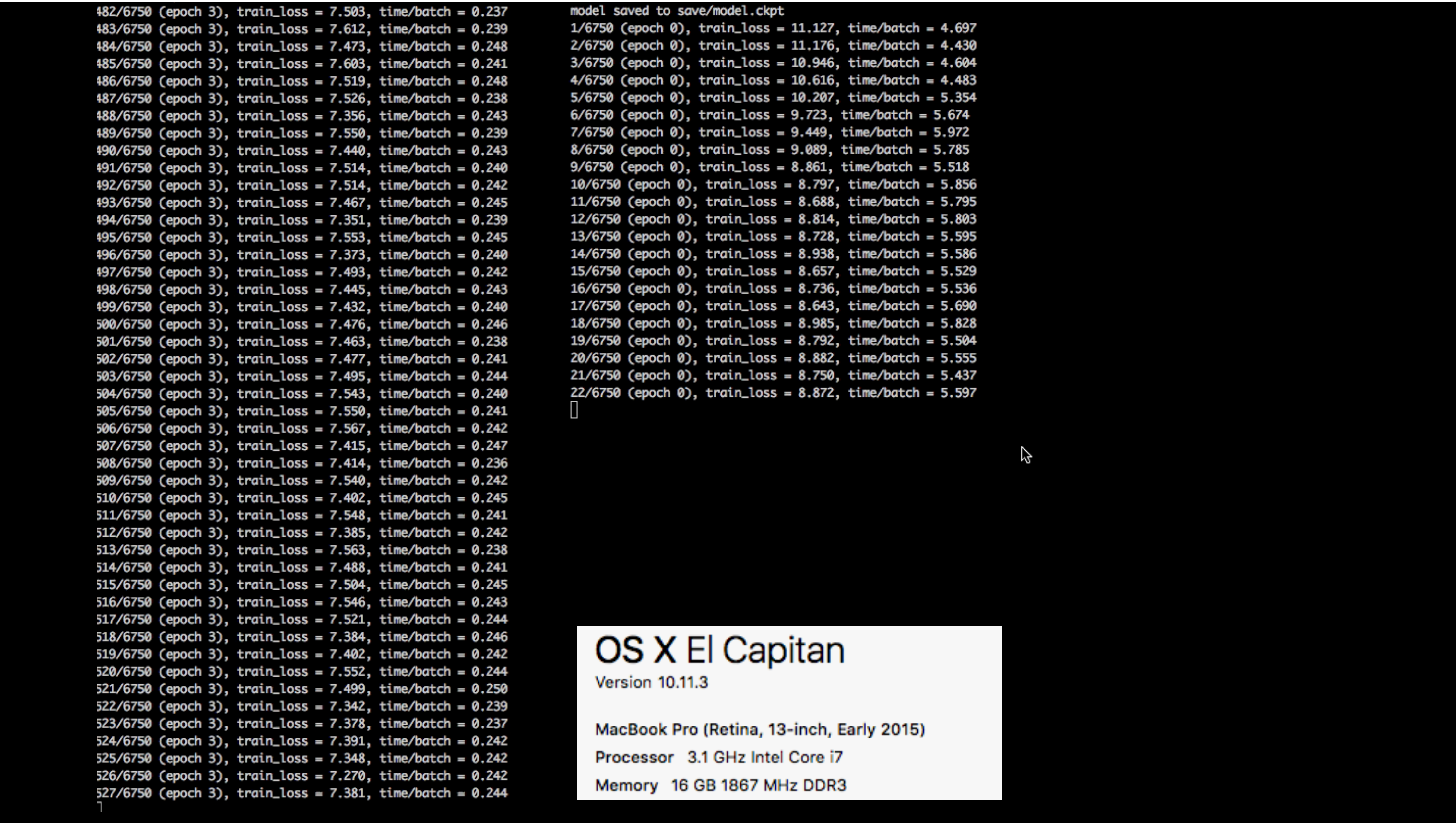


```
05] successfully opened CUDA library libcublas.so locally
9] Couldn't open CUDA library libcudnn.so. LD_LIBRARY_PATH: /usr/local/cuda/lib64:
c:1562] Unable to load cuDNN DSO
05] successfully opened CUDA library libcufft.so locally
05] successfully opened CUDA library libcuda.so.1 locally
05] successfully opened CUDA library libcurand.so locally
-ubyte.gz')
cationWarning: converting an array with ndim > 0 to an index will result in an error in the
e]
.py:42: VisibleDeprecationWarning: converting an array with ndim > 0 to an index will result
, 1)
-ubyte.gz')
ubyte.gz')
ubyte.gz')
xecutor.cc:900] successful NUMA node read from SysFS had negative value (-1), but there must
t.cc:102] Found device 0 with properties:

7



t.cc:126] DMA: 0
t.cc:136] 0:   Y
ice.cc:755] Creating TensorFlow device (/gpu:0) -> (device: 0, name: GRID K520, pci bus id:
locator.cc:244] PoolAllocator: After 1704 get requests, put_count=1321 evicted_count=1000 ev
locator.cc:256] Raising pool_size_limit_ from 100 to 110
locator.cc:244] PoolAllocator: After 1704 get requests, put_count=1812 evicted_count=1000 ev
locator.cc:256] Raising pool_size_limit_ from 256 to 281
```



```
482/6750 (epoch 3), train_loss = 7.503, time/batch = 0.237
483/6750 (epoch 3), train_loss = 7.612, time/batch = 0.239
484/6750 (epoch 3), train_loss = 7.473, time/batch = 0.248
485/6750 (epoch 3), train_loss = 7.603, time/batch = 0.241
486/6750 (epoch 3), train_loss = 7.519, time/batch = 0.248
487/6750 (epoch 3), train_loss = 7.526, time/batch = 0.238
488/6750 (epoch 3), train_loss = 7.356, time/batch = 0.243
489/6750 (epoch 3), train_loss = 7.550, time/batch = 0.239
490/6750 (epoch 3), train_loss = 7.440, time/batch = 0.243
491/6750 (epoch 3), train_loss = 7.514, time/batch = 0.240
492/6750 (epoch 3), train_loss = 7.514, time/batch = 0.242
493/6750 (epoch 3), train_loss = 7.467, time/batch = 0.245
494/6750 (epoch 3), train_loss = 7.351, time/batch = 0.239
495/6750 (epoch 3), train_loss = 7.553, time/batch = 0.245
496/6750 (epoch 3), train_loss = 7.373, time/batch = 0.240
497/6750 (epoch 3), train_loss = 7.493, time/batch = 0.242
498/6750 (epoch 3), train_loss = 7.445, time/batch = 0.243
499/6750 (epoch 3), train_loss = 7.432, time/batch = 0.240
500/6750 (epoch 3), train_loss = 7.476, time/batch = 0.246
501/6750 (epoch 3), train_loss = 7.463, time/batch = 0.238
502/6750 (epoch 3), train_loss = 7.477, time/batch = 0.241
503/6750 (epoch 3), train_loss = 7.495, time/batch = 0.244
504/6750 (epoch 3), train_loss = 7.543, time/batch = 0.240
505/6750 (epoch 3), train_loss = 7.550, time/batch = 0.241
506/6750 (epoch 3), train_loss = 7.567, time/batch = 0.242
507/6750 (epoch 3), train_loss = 7.415, time/batch = 0.247
508/6750 (epoch 3), train_loss = 7.414, time/batch = 0.236
509/6750 (epoch 3), train_loss = 7.540, time/batch = 0.242
510/6750 (epoch 3), train_loss = 7.402, time/batch = 0.245
511/6750 (epoch 3), train_loss = 7.548, time/batch = 0.241
512/6750 (epoch 3), train_loss = 7.385, time/batch = 0.242
513/6750 (epoch 3), train_loss = 7.563, time/batch = 0.238
514/6750 (epoch 3), train_loss = 7.488, time/batch = 0.241
515/6750 (epoch 3), train_loss = 7.504, time/batch = 0.245
516/6750 (epoch 3), train_loss = 7.546, time/batch = 0.243
517/6750 (epoch 3), train_loss = 7.521, time/batch = 0.244
518/6750 (epoch 3), train_loss = 7.384, time/batch = 0.246
519/6750 (epoch 3), train_loss = 7.402, time/batch = 0.242
520/6750 (epoch 3), train_loss = 7.552, time/batch = 0.244
521/6750 (epoch 3), train_loss = 7.499, time/batch = 0.250
522/6750 (epoch 3), train_loss = 7.342, time/batch = 0.239
523/6750 (epoch 3), train_loss = 7.378, time/batch = 0.237
524/6750 (epoch 3), train_loss = 7.391, time/batch = 0.242
525/6750 (epoch 3), train_loss = 7.348, time/batch = 0.242
526/6750 (epoch 3), train_loss = 7.270, time/batch = 0.242
527/6750 (epoch 3), train_loss = 7.381, time/batch = 0.244
```

```
model saved to save/model.ckpt
1/6750 (epoch 0), train_loss = 11.127, time/batch = 4.697
2/6750 (epoch 0), train_loss = 11.176, time/batch = 4.430
3/6750 (epoch 0), train_loss = 10.946, time/batch = 4.604
4/6750 (epoch 0), train_loss = 10.616, time/batch = 4.483
5/6750 (epoch 0), train_loss = 10.207, time/batch = 5.354
6/6750 (epoch 0), train_loss = 9.723, time/batch = 5.674
7/6750 (epoch 0), train_loss = 9.449, time/batch = 5.972
8/6750 (epoch 0), train_loss = 9.089, time/batch = 5.785
9/6750 (epoch 0), train_loss = 8.861, time/batch = 5.518
10/6750 (epoch 0), train_loss = 8.797, time/batch = 5.856
11/6750 (epoch 0), train_loss = 8.688, time/batch = 5.795
12/6750 (epoch 0), train_loss = 8.814, time/batch = 5.803
13/6750 (epoch 0), train_loss = 8.728, time/batch = 5.595
14/6750 (epoch 0), train_loss = 8.938, time/batch = 5.586
15/6750 (epoch 0), train_loss = 8.657, time/batch = 5.529
16/6750 (epoch 0), train_loss = 8.736, time/batch = 5.536
17/6750 (epoch 0), train_loss = 8.643, time/batch = 5.690
18/6750 (epoch 0), train_loss = 8.985, time/batch = 5.828
19/6750 (epoch 0), train_loss = 8.792, time/batch = 5.504
20/6750 (epoch 0), train_loss = 8.882, time/batch = 5.555
21/6750 (epoch 0), train_loss = 8.750, time/batch = 5.437
22/6750 (epoch 0), train_loss = 8.872, time/batch = 5.597
```

OS X El Capitan

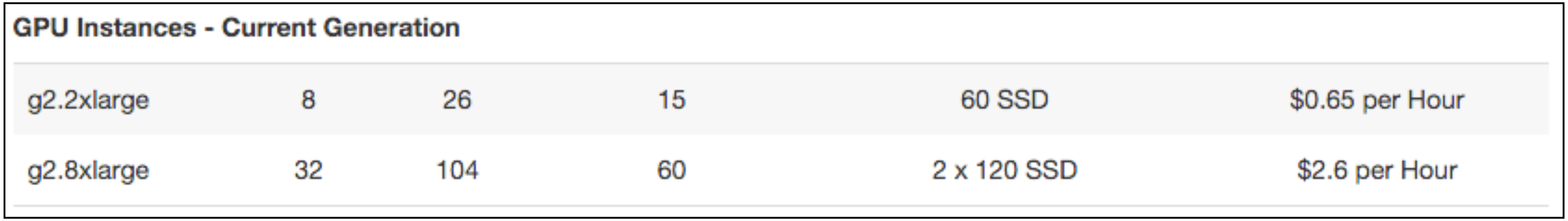
Version 10.11.3

MacBook Pro (Retina, 13-inch, Early 2015)

Processor 3.1 GHz Intel Core i7

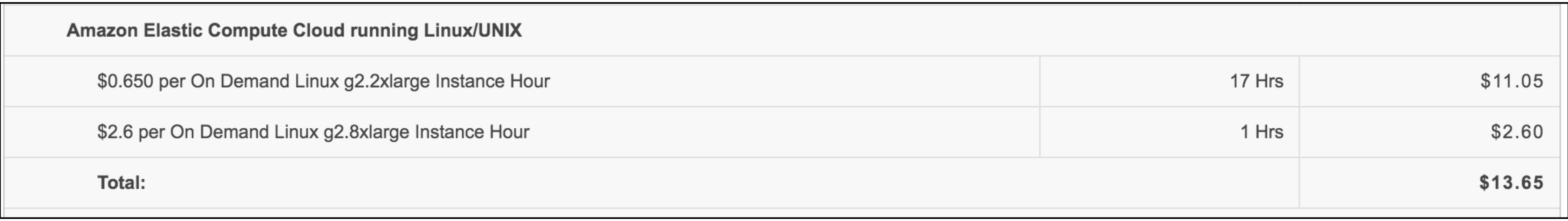
Memory 16 GB 1867 MHz DDR3

AWS GPU price in Oregon



GPU Instances - Current Generation

g2.2xlarge	8	26	15	60 SSD	$0.65 per Hour
g2.8xlarge	32	104	60	2 x 120 SSD	$2.6 per Hour

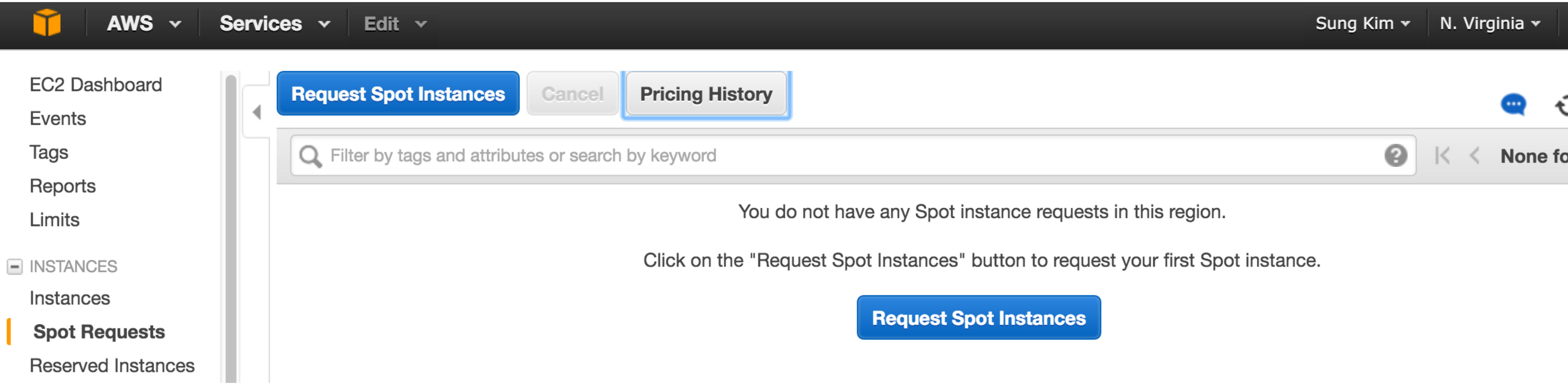


Amazon Elastic Compute Cloud running Linux/UNIX

$0.650 per On Demand Linux g2.2xlarge Instance Hour	17 Hrs	$11.05
$2.6 per On Demand Linux g2.8xlarge Instance Hour	1 Hrs	$2.60
Total:		**$13.65**

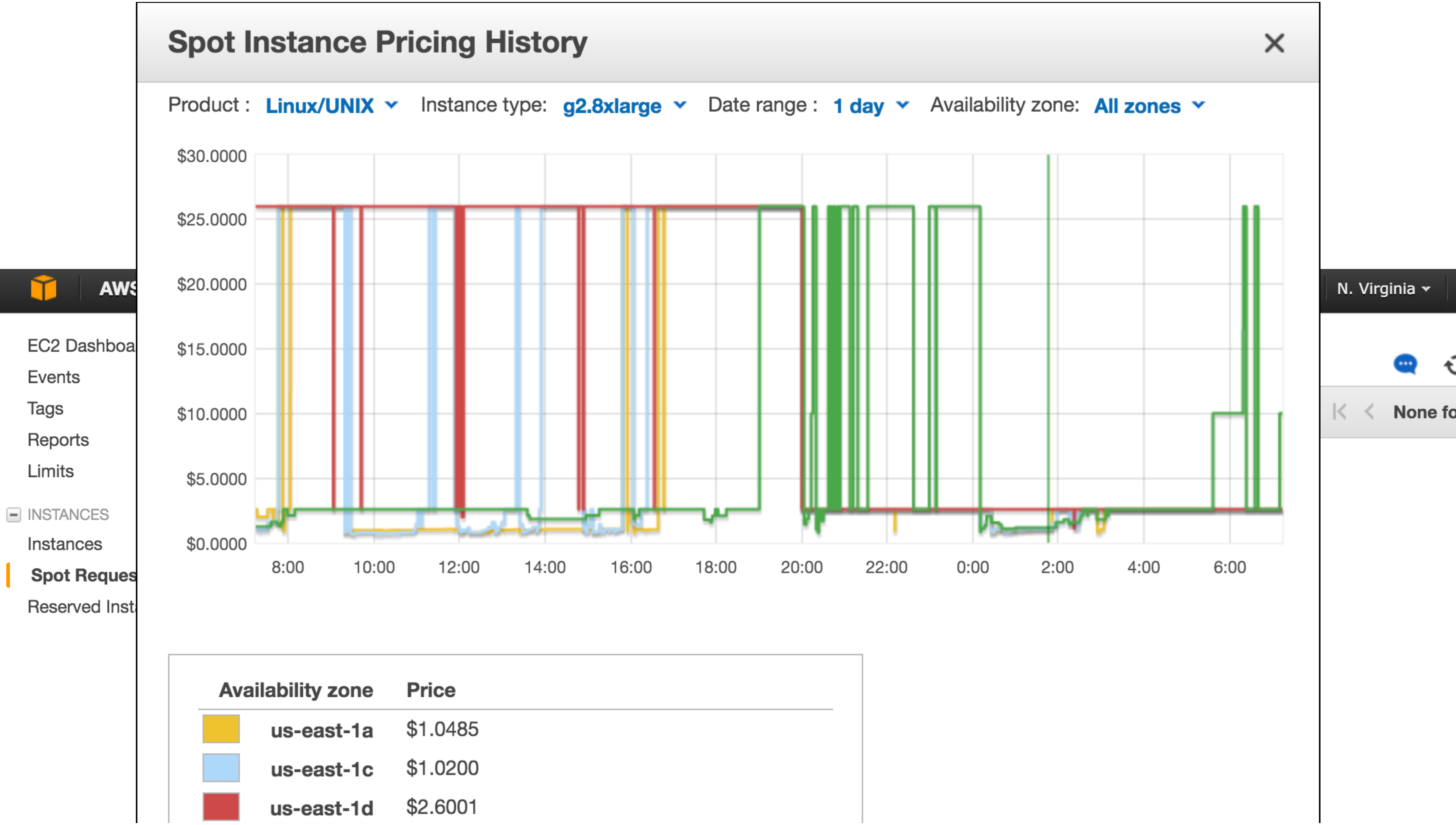
2.6 * 24 * 30 = 1,872 USD

Spot instances



You do not have any Spot instance requests in this region.

Click on the "Request Spot Instances" button to request your first Spot instance.

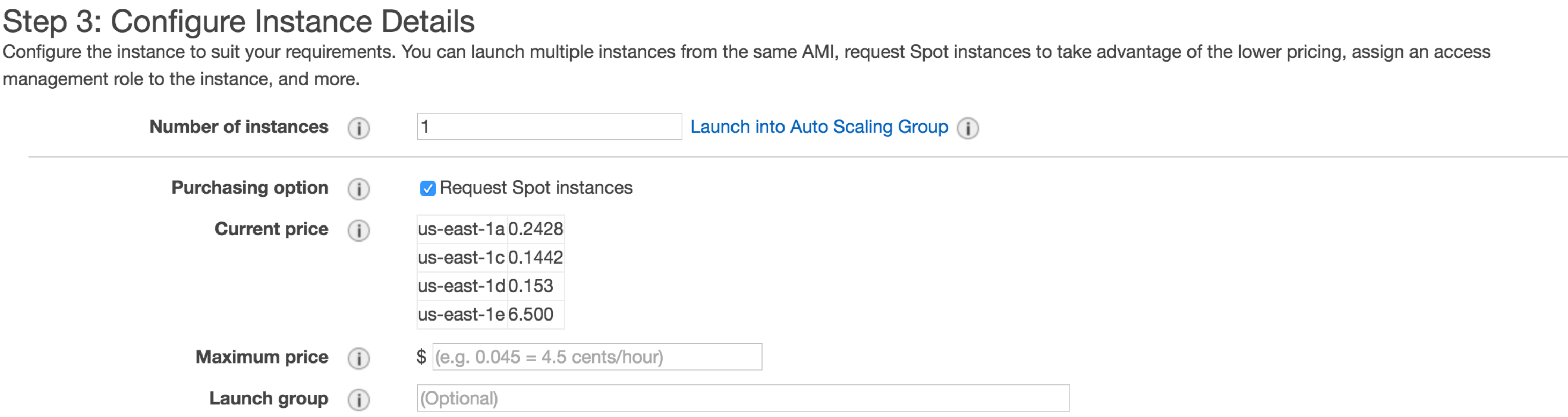


Spot Instance Pricing History

Product : Linux/UNIX Instance type: g2.8xlarge Date range : 1 day Availability zone: All zones

	Availability zone	Price
🟨	us-east-1a	$1.0485
🟦	us-east-1c	$1.0200
🟥	us-east-1d	$2.6001

Price bidding



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 access management role to the instance, and more.

Number of instances 1 Launch into Auto Scaling Group

Purchasing option ☑ Request Spot instances

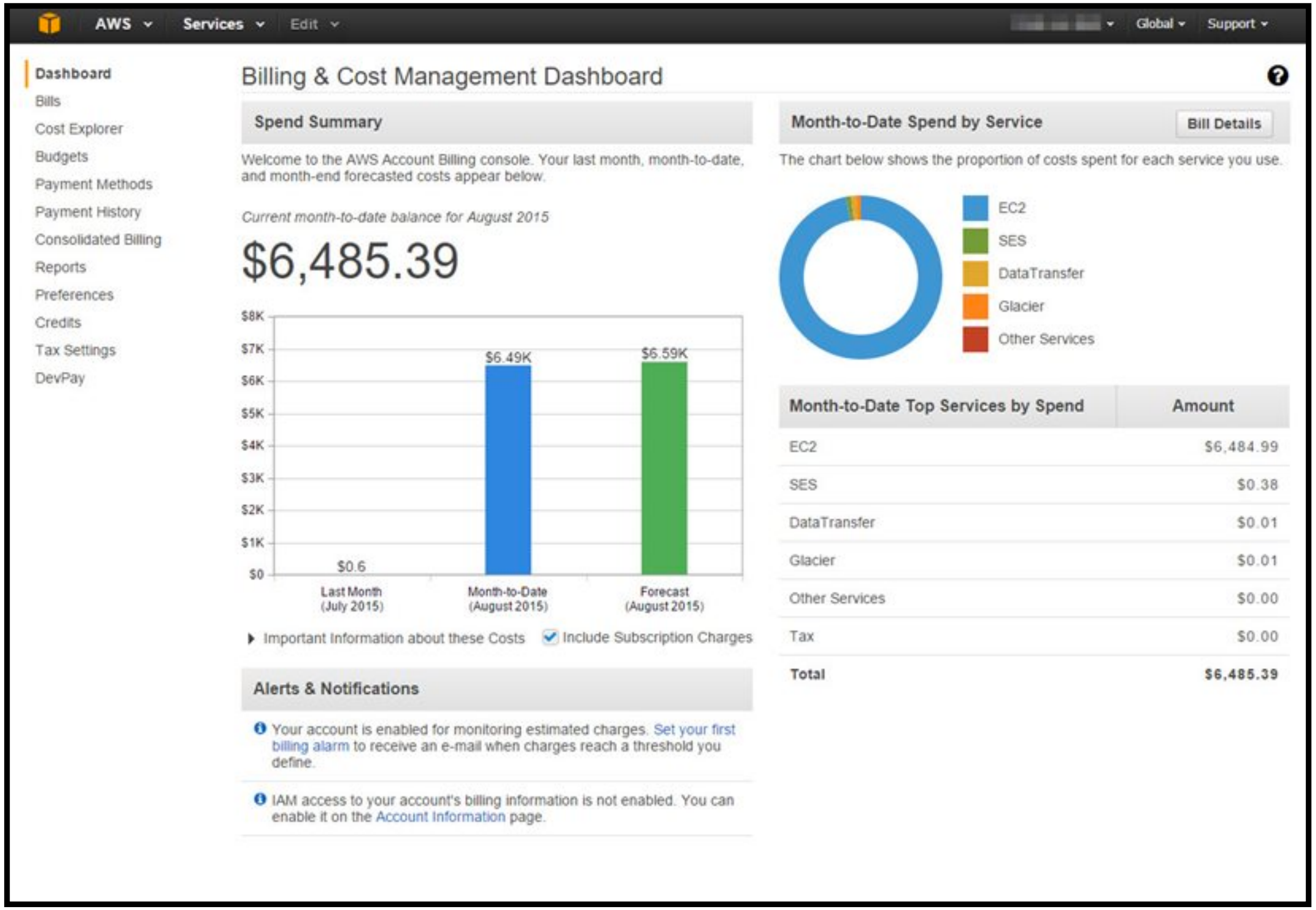
Current price

us-east-1a	0.2428
us-east-1c	0.1442
us-east-1d	0.153
us-east-1e	6.500

Maximum price $ (e.g. 0.045 = 4.5 cents/hour)

Launch group (Optional)

bill, bill, bill!



Dashboard
Bills
Cost Explorer
Budgets
Payment Methods
Payment History
Consolidated Billing
Reports
Preferences
Credits
Tax Settings
DevPay

Billing & Cost Management Dashboard

Spend Summary

Welcome to the AWS Account Billing console. Your last month, month-to-date, and month-end forecasted costs appear below.

Current month-to-date balance for August 2015

$6,485.39

▸ Important Information about these Costs ☑ Include Subscription Charges

Alerts & Notifications

Your account is enabled for monitoring estimated charges. Set your first billing alarm to receive an e-mail when charges reach a threshold you define.

IAM access to your account's billing information is not enabled. You can enable it on the Account Information page.

Month-to-Date Spend by Service

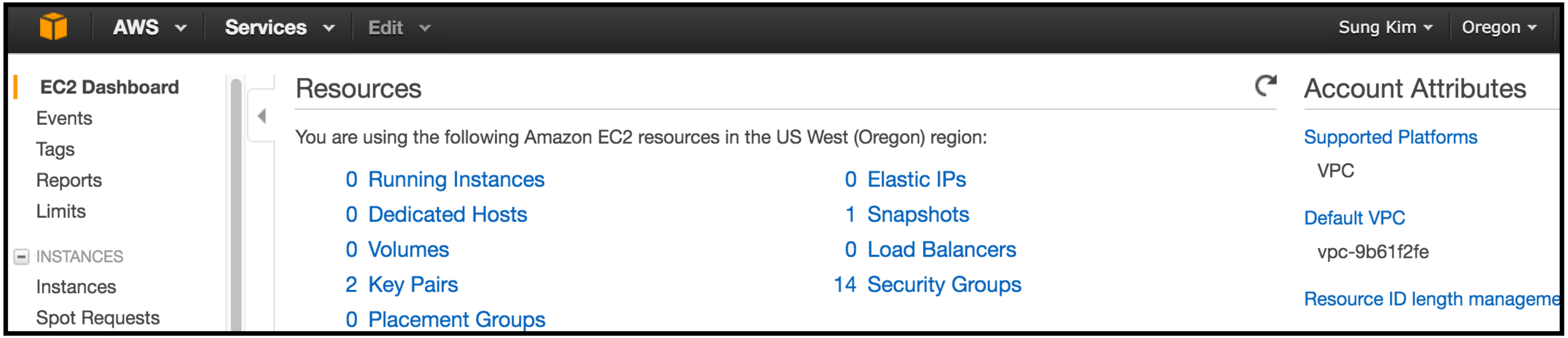
Bill Details

The chart below shows the proportion of costs spent for each service you use.

🟦 EC2
🟩 SES
🟨 DataTransfer
🟧 Glacier
🟥 Other Services

Month-to-Date Top Services by Spend	Amount
EC2	$6,484.99
SES	$0.38
DataTransfer	$0.01
Glacier	$0.01
Other Services	$0.00
Tax	$0.00
Total	**$6,485.39**

Check, stop, and terminate



Resources

You are using the following Amazon EC2 resources in the US West (Oregon) region:

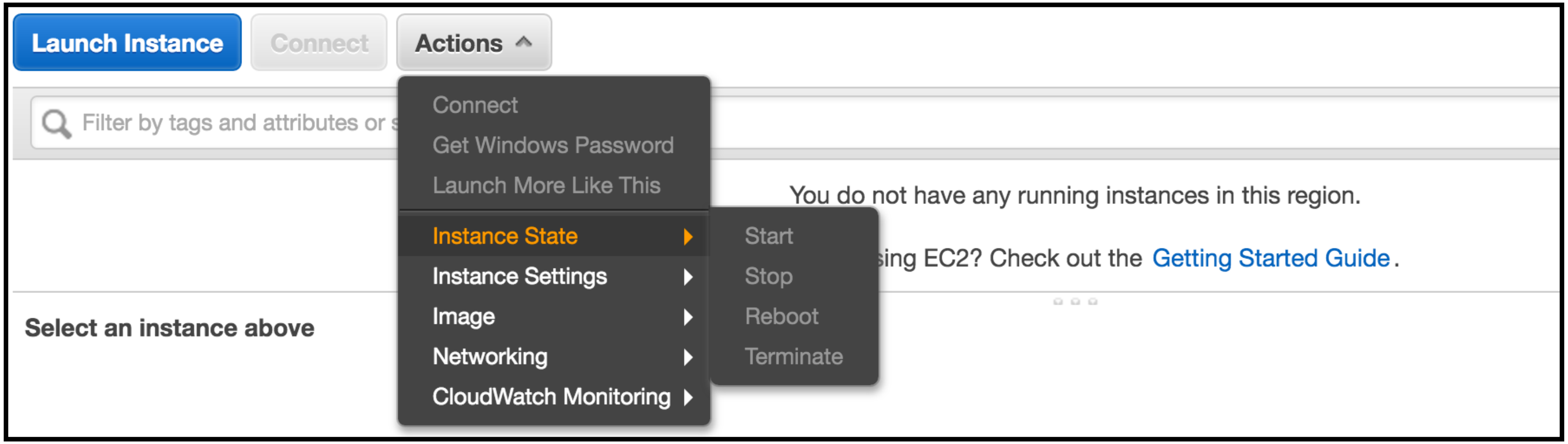
0 Running Instances	0 Elastic IPs
0 Dedicated Hosts	1 Snapshots
0 Volumes	0 Load Balancers
2 Key Pairs	14 Security Groups
0 Placement Groups	

Account Attributes

Supported Platforms
VPC

Default VPC
vpc-9b61f2fe

Resource ID length manageme

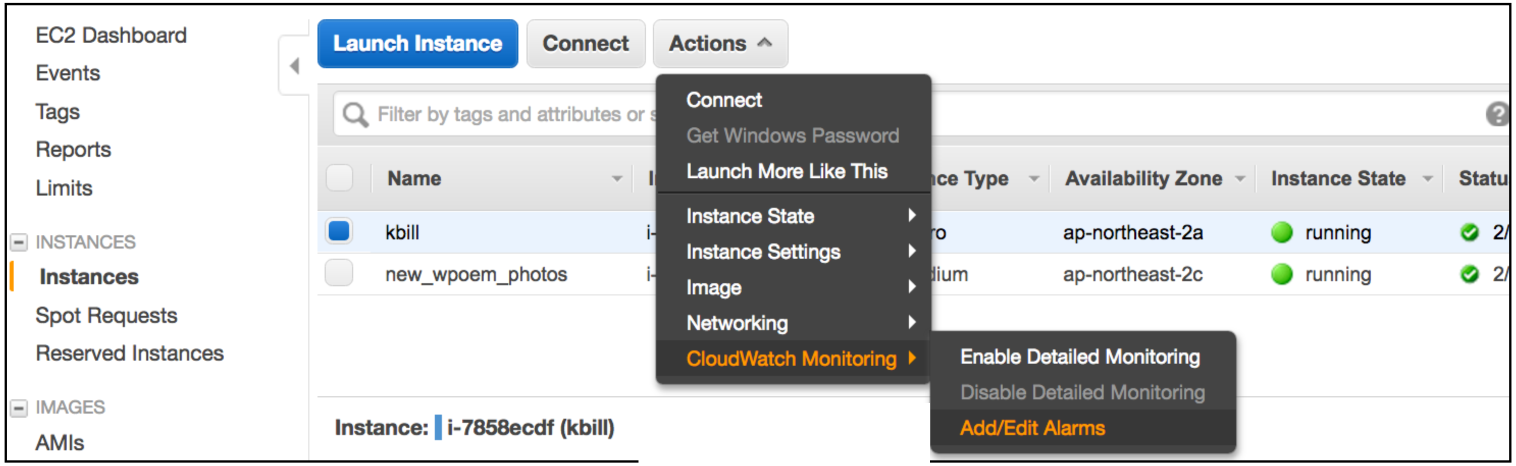


You do not have any running instances in this region.

sing EC2? Check out the Getting Started Guide.

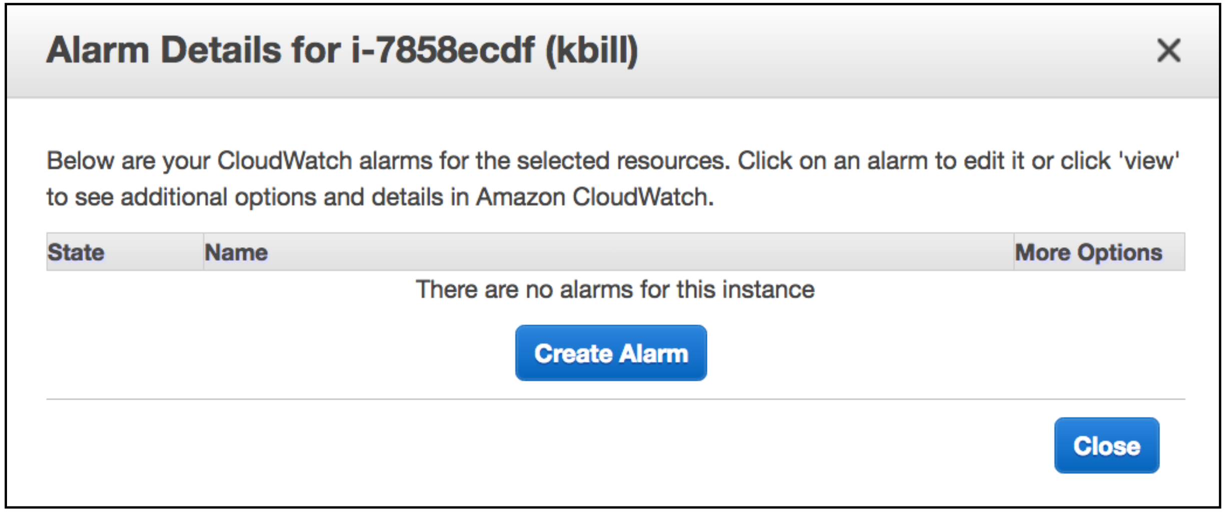
Select an instance above

Cloud Watch



	Name	I	ce Type	Availability Zone	Instance State	Statu
■	kbill	i-	ro	ap-northeast-2a	● running	✓ 2/
□	new_wpoem_photos	i-	lium	ap-northeast-2c	● running	✓ 2/

Instance: i-7858ecdf (kbill)



Alarm Details for i-7858ecdf (kbill)

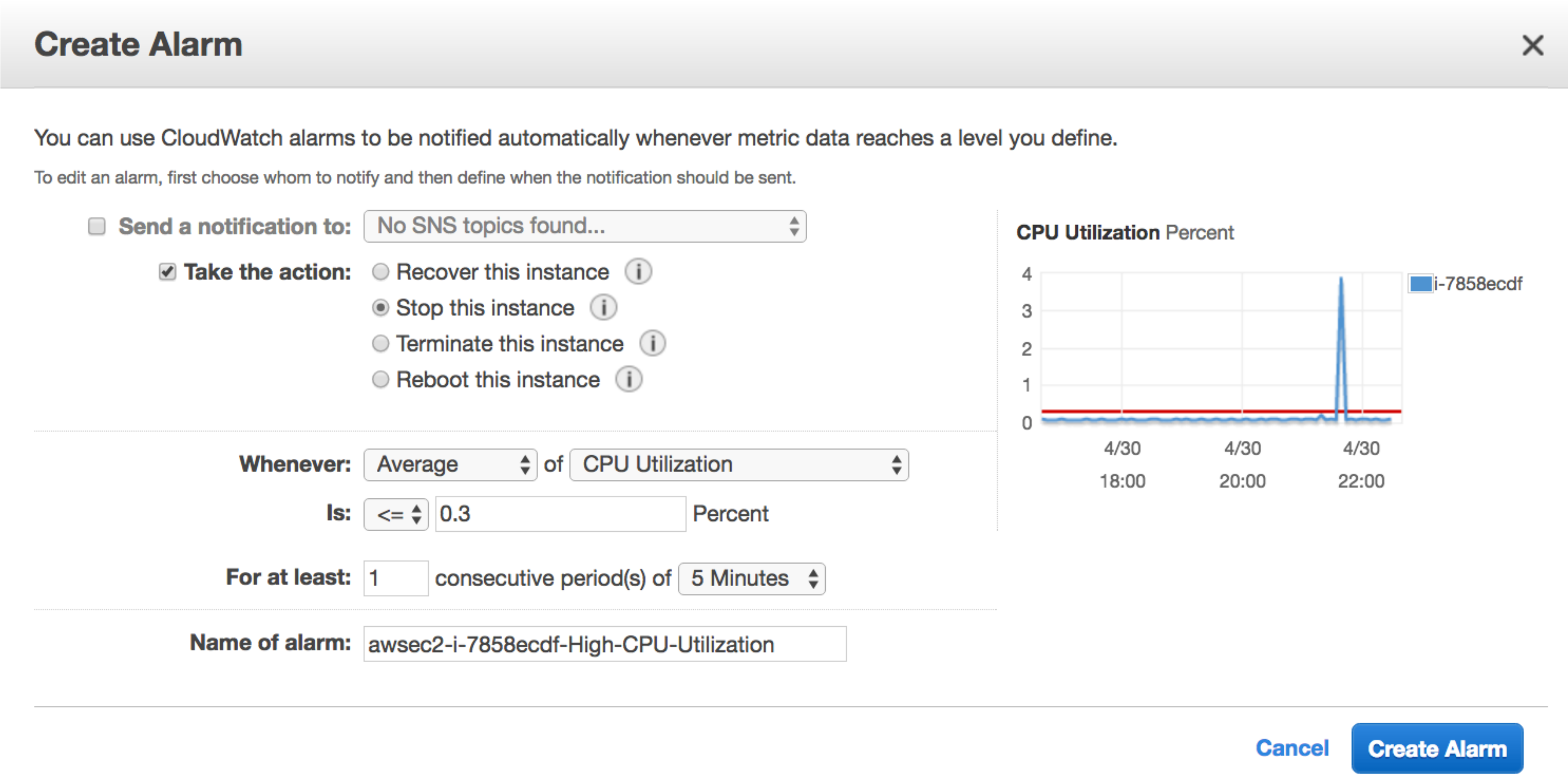
Below are your CloudWatch alarms for the selected resources. Click on an alarm to edit it or click 'view' to see additional options and details in Amazon CloudWatch.

State	Name	More Options

There are no alarms for this instance

http://docs.aws.amazon.com/AmazonCloudWatch/latest/DeveloperGuide/UsingAlarmActions.html#AddingStopActions

Stop when CPU utilization <= 0.3



Create Alarm

You can use CloudWatch alarms to be notified automatically whenever metric data reaches a level you define.

To edit an alarm, first choose whom to notify and then define when the notification should be sent.

□ **Send a notification to:** No SNS topics found...

☑ **Take the action:**
○ Recover this instance
● Stop this instance
○ Terminate this instance
○ Reboot this instance

Whenever: Average of CPU Utilization

Is: <= 0.3 Percent

For at least: 1 consecutive period(s) of 5 Minutes

Name of alarm: awsec2-i-7858ecdf-High-CPU-Utilization

Shutdown after training

```
$ screen


$ sudo -i
# python train.py; shutdown -h now
```

Deep learning for Everyone Season 2 coming soon!

Sung Kim <hunkim+ml@gmail.com>
http://hunkim.github.io/ml/