

TensorFlow GPU @AWS Spot Instances without losing data **+ 100 USD credit**

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

Consulted Channy Yun and DoHyun Jung

GPU

X10
X30

- 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.

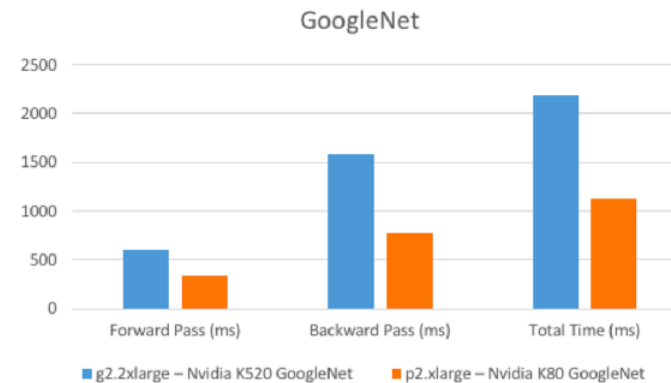
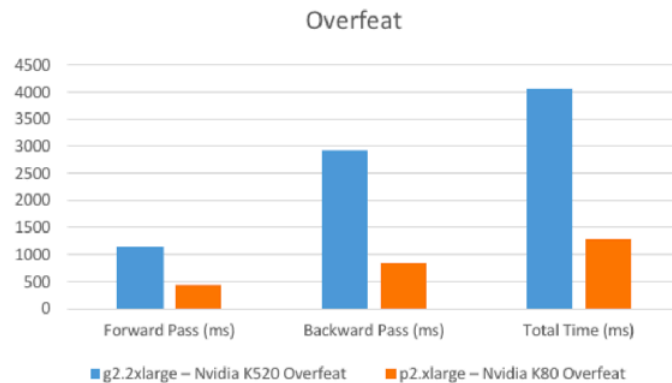
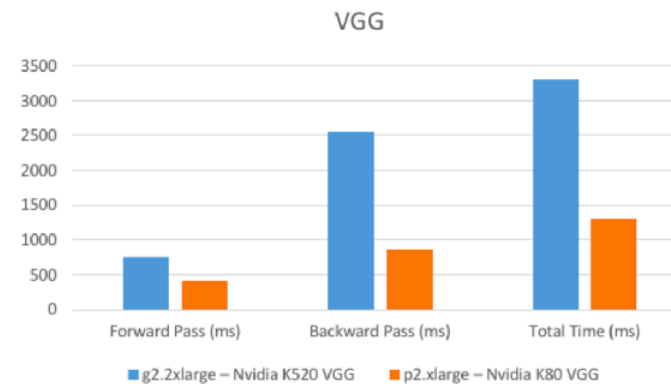
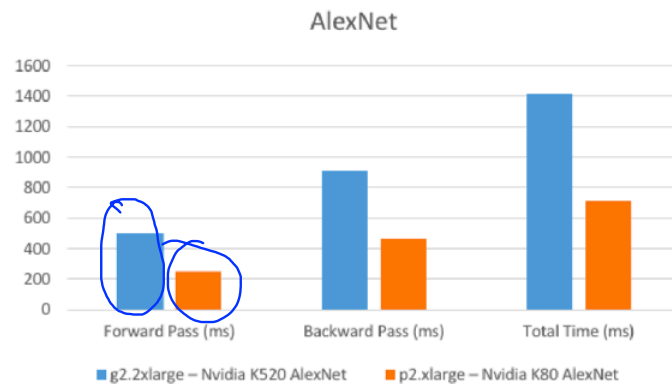


AWS G2 and P2 Instances

Model	GPUs	vCPU	Mem (GiB)	SSD Storage (GB)	GPU Memory
<u>g2.2xlarge</u>	1	8	15	1 x 60	4G (8G)
g2.8xlarge	4	32	60	2 x 120	4G (32G)

Instance Name	GPU Count	vCPU Count	Memory	Parallel Processing Cores	GPU Memory	Network Performance
<u>p2.xlarge</u>	1	4	61 GiB	2,496	12 GiB	High
p2.8xlarge	8	32	488 GiB	19,968	96 GiB	10 Gigabit
p2.16xlarge	16	64	732 GiB	39,936	192 GiB	20 Gigabit


AWS G2 and P2 Instances



TensorFlow GPU Performance: AlexNet, VGG, Overfeat, GoogleNet

<http://www.bitfusion.io/2016/11/03/quick-comparison-of-tensorflow-gpu-performance-on-aws-p2-and-g2-instances/>

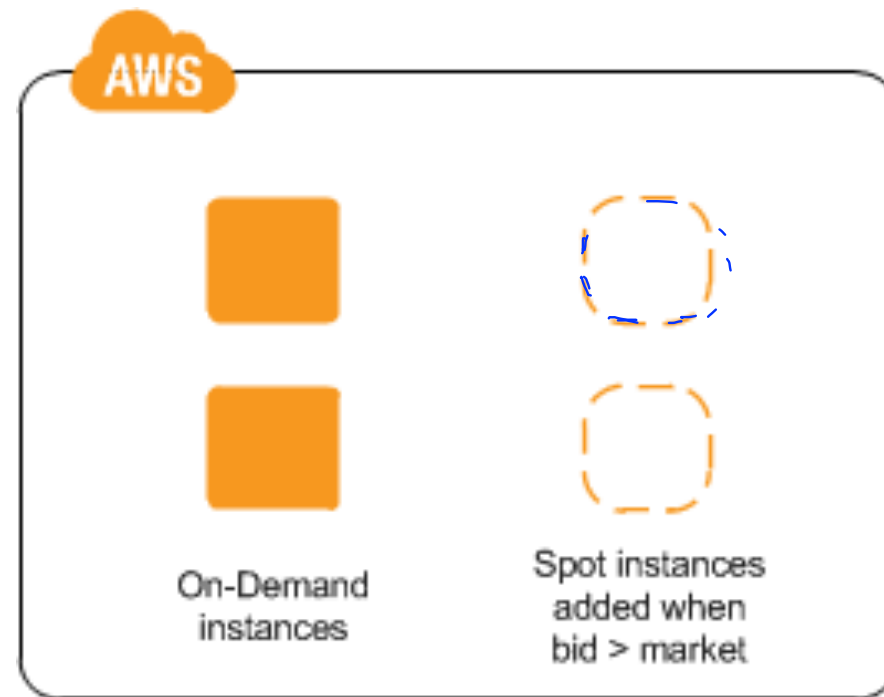
US EAST (N.Virginia) On-Demand Price

	vCPU	ECU	Memory (GiB)	Instance Storage (GB)	Linux/UNIX Usage
GPU Instances - Current Generation					
 <u>p2.xlarge</u>	4	12	61	EBS Only	<u>\$0.9 per Hour</u>
p2.8xlarge	32	94	488	EBS Only	\$7.2 per Hour
p2.16xlarge	64	188	732	EBS Only	\$14.4 per Hour
<u>g2.2xlarge</u>	8	26	15	60 SSD	<u>\$0.65 per Hour</u>
g2.8xlarge	32	104	60	2 x 120 SSD	\$2.6 per Hour

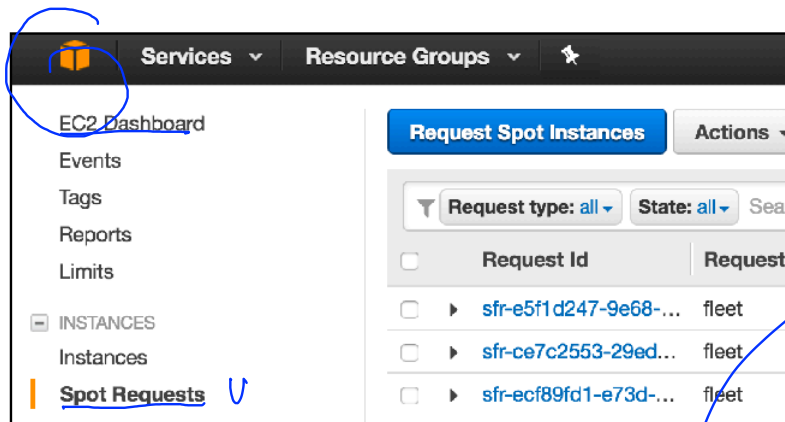
US EAST (N.Virginia) Spot Instance Price

g2.2xlarge	\$0.1564 per Hour	VS 0.6
g2.8xlarge	\$1.966 per Hour	VS 2.8
p2.xlarge	\$0.243 per Hour	<u>VS 0.9</u>
p2.8xlarge	\$3.583 per Hour	VS 7.2

On-Demand VS Spot Instances



Request spot instances and save \$



This screenshot shows the 'Select request type' form. The 'Request type' section has three radio buttons: 'Request' (selected), 'Request and Maintain', and 'Reserve for duration'. The 'Target capacity' field is set to '1' and is circled in blue. The 'AMI' dropdown menu is set to 'Amazon Linux AMI 2016.09.0.20161028 x86_64' and is also circled in blue. The 'Instance type(s)' section has a 'Select' button. A blue arrow points from the 'Request Spot Instances' button in the previous screenshot to the 'Request' radio button in this form.

Select request type

Request type ☒ Request
Submit a one-time Spot instance request

☐ Request and Maintain
Request a fleet of Spot instances to maintain your target capacity

☐ Reserve for duration
Request a Spot instance with no interruption for 1 to 6 hours (a Spot block)

Target capacity instances

AMI Select

Instance type(s) Select

Request spot instances and save \$

Select instance types

☐ Supports dedicated tenancy

Pricing History

Spot Bid Advisor

Instance type ▾	vCPUs ▾	Memory (GiB) ▾	Storage (GB) ▾	Network ▾	Current spot price ▾	Spot savings ▾
GPU compute ▴ ▾	1 ▴ ▾	(Any)	(Any)	Any network ▴ ▾		
<input checked="" type="checkbox"/> p2.xlarge	4	61	EBS only	High	\$0.1162	87%
<input type="checkbox"/> p2.8xlarge	32	488	EBS only	10 Gigabit	\$1.3821	81%
<input type="checkbox"/> p2.16xlarge	64	732	EBS only	20 Gigabit	\$4.5648	68%

Request spot instances ready (takes 2~3 min)

Launch Instance Connect Actions

Filter by tags and attributes or search by keyword

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)	IPv4 Public IP
	i-013b972e59561e7b3	g2.2xlarge	us-east-1a	pending	Initializing	None	ec2-52-90-17-8.compute-1.amazonaws.com	52.90.17.8
	i-0480cf73c8a91222e	g2.2xlarge	us-east-1d	terminated		None		-

Instance: i-013b972e59561e7b3 Public DNS: ec2-52-90-17-8.compute-1.amazonaws.com

Description Status Checks Monitoring Tags

Instance ID	i-013b972e59561e7b3	Public DNS (IPv4)	ec2-52-90-17-8.compute-1.amazonaws.com
Instance state	pending	IPv4 Public IP	52.90.17.8
Instance type	g2.2xlarge	IPv6 IPs	-
Elastic IPs		Private DNS	ip-172-31-18-62.ec2.internal
Availability zone	us-east-1a	Private IPs	172.31.18.62
Security groups	default view inbound rules	Secondary private IPs	
Scheduled events	-	VPC ID	vpc-3e020a5b
AMI ID	TF0.12_CUDA8 (ami-eb31c2fd)	Subnet ID	subnet-02bdcf5b

Work with screen

- ssh

- screen: open a new screen

- python train.py [Ⓢ]..[Ⓢ]; echo "Done" | mail -s "Finished" hunkim@gmail.com
- Ctrl-a d (to exit screen)

- screen -r: attach the screen

batch.ly

AWS COST
REDUCTION.
AUTOMATED.

AWS Spot Instance

***TERMINATION
NOTICES***



But spot instances can be terminated by AWS!

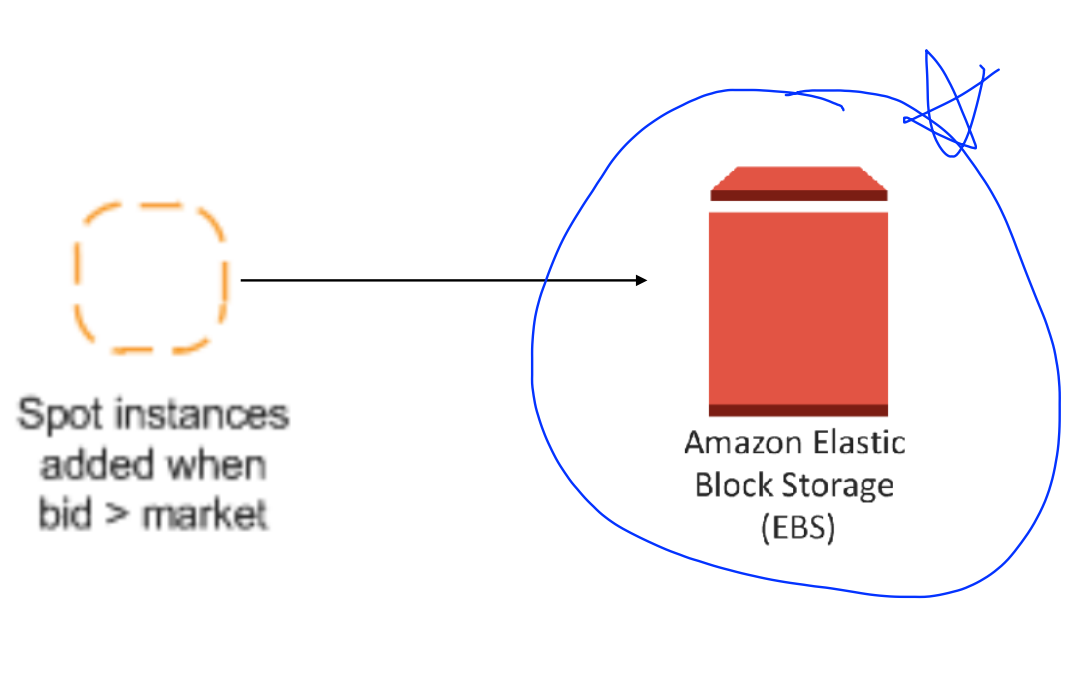
Solution: Spot Instance + EBS Volume



Spot instances
added when
 $\text{bid} > \text{market}$

Automatic termination after finishing job! (save \$)

Solution: Spot Instance + EBS Volume (do not delete on termination)



Automatic termination after finishing job! (save \$)

Don't delete the volume!

Spot instance launch wizard

Step 1: Find instance types
Step 2: Configure
Step 3: Review

Configure storage

Instance store ⓘ

☐ Attach at launch

EBS volumes ⓘ

Device ⓘ	Snapshot	Size (GiB)	Volume Type	IOPS	Delete ⓘ
Root: /dev/xvda	snap-fe8a3c04	8	General Purpose (SSD) ▾		<input type="checkbox"/> ⓘ

No additional EBS volumes configured

Add new volume

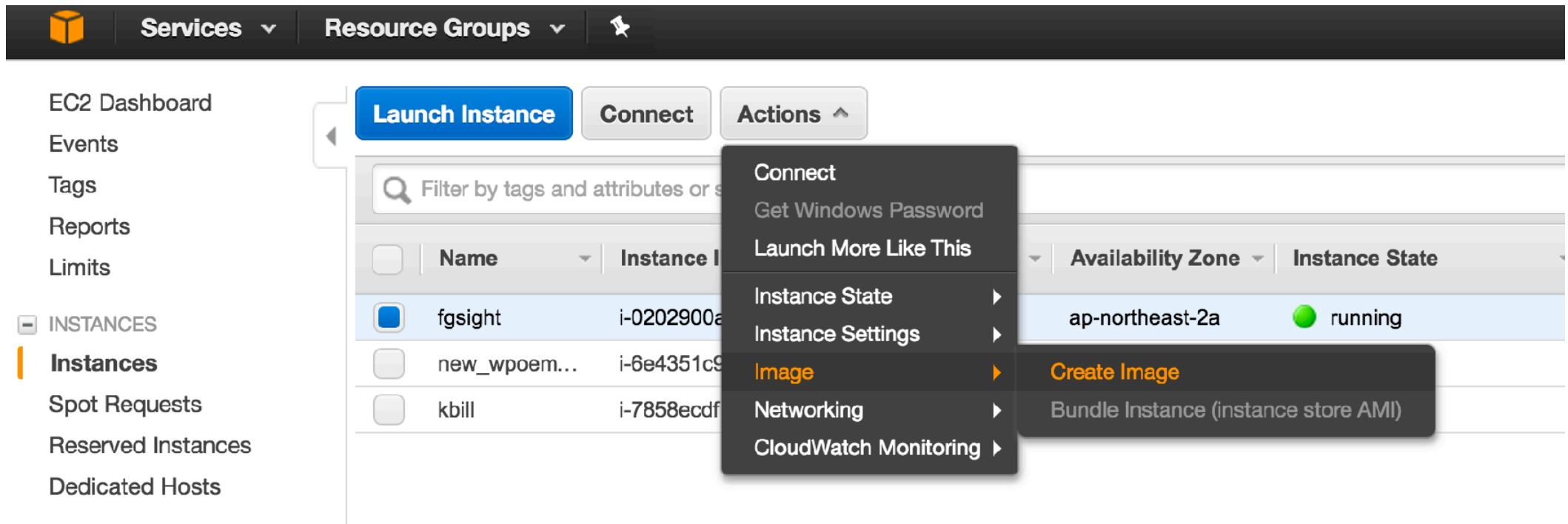
EBS-optimized ⓘ

☐ Launch EBS-optimized instances

Workflow

- Create a spot instance (from a AMI)
 - **Do not delete the main disk on termination**
- Run TF tasks; sudo shutdown now
 - run (in screen) and save results on the volume
 - terminate the instance (save \$)
- Create a (1) snapshot and (2) AMI from the leftover volume
- Create a new spot using the AMI
 - Delete old AMIs

0. Create AMI - TF/Cuda



EC2 Dashboard

Events

Tags

Reports

Limits

INSTANCES

Instances

Spot Requests

Reserved Instances

Dedicated Hosts

Launch Instance

Connect

Actions

Filter by tags and attributes or s

	Name	Instance I	Availability Zone	Instance State
<input checked="" type="checkbox"/>	fgsight	i-0202900a	ap-northeast-2a	running
<input type="checkbox"/>	new_wpoem...	i-6e4351c9		
<input type="checkbox"/>	kbill	i-7858ecdf		

Connect

Get Windows Password

Launch More Like This

Instance State

Instance Settings

Image

Networking

CloudWatch Monitoring

Create Image

Bundle Instance (instance store AMI)

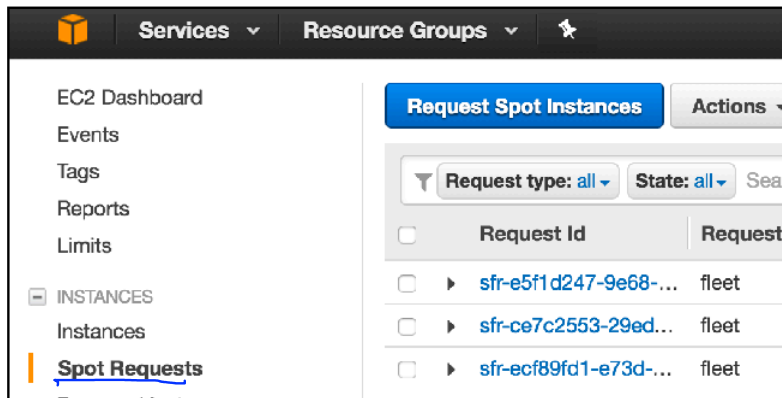
N. Virginia: ami-eb31c2fd (maybe outdated soon)

✓ <https://github.com/ritchieng/tensorflow-aws-ami>



<http://expressionflow.com/2016/10/09/installing-tensorflow-on-an-aws-ec2-p2-gpu-instance>

I. Request spot instances using AMI



Select request type

Request type

- ☒ **Request**
Submit a one-time Spot instance request
- ☐ **Request and Maintain**
Request a fleet of Spot instances to maintain your target capacity
- ☐ **Reserve for duration**
Request a Spot instance with no interruption for 1 to 6 hours (a Spot block)

Target capacity ⓘ 1 instances ▾

AMI ⓘ Amazon Linux AMI 2016.09.0.20161028 x86_64 ▾ **Select**

Instance type(s) ⓘ Select multiple instance types to find the lowest priced instances available **Select**


I. Request spot instances using AMI

Choose an Amazon Machine Image (AMI) x

ami-eb31c2fd

Community AMIs

« < 1 to 1 of 1 AMIs > »



TF0.12_CUDA8

ami-eb31c2fd

Root device type: ebs

Virtualization type: hvm

Owner: 705119112097

Select

64-bit

I. Request spot instances: check price

Select instance types

☐ Supports dedicated tenancy

Pricing History

Spot Bid Advisor

Instance type ▾	vCPUs ▾	Memory (GiB) ▾	Storage (GB) ▾	Network ▾	Current spot price ▾	Spot savings ▾
GPU compute ▴▾	1 ▴▾	(Any)	(Any)	Any network ▴▾		
<input checked="" type="checkbox"/> p2.xlarge	4	61	EBS only	High	\$0.1162	87%
<input type="checkbox"/> p2.8xlarge	32	488	EBS only	10 Gigabit	\$1.3821	81%
<input type="checkbox"/> p2.16xlarge	64	732	EBS only	20 Gigabit	\$4.5648	68%

I. Request spot instances: Don't delete the volume!

Spot instance launch wizard

[Step 1: Find instance types](#)

[Step 2: Configure](#)

[Step 3: Review](#)

Configure storage

Instance store ⓘ

☐ Attach at launch

EBS volumes ⓘ

Device ⓘ	Snapshot	Size (GiB)	Volume Type	IOPS	Delete ⓘ
Root: /dev/xvda	snap-fe8a3c04	8	General Purpose (SSD) ▾		<input type="checkbox"/> ⓘ

No additional EBS volumes configured

Add new volume

EBS-optimized ⓘ

☐ Launch EBS-optimized instances

I. Request spot instances: Success

Network ⓘ	vpc-3e020a5b (172.31.0.0/16) (default)
Availability Zones / Subnets ⓘ	No preference (launch in cheapest Availability Zone)
Security groups ⓘ	sg-c6c631a1
Auto-assign IPv4 Public IP ⓘ	Use subnet setting (Enable)

▼ Instance details

Success

Spot request with id: sfr-2224b5ea-2ff1-470f-9454-5a877ba574b7 successfully created.

OK

Root volume ⓘ	/dev/sda1: 20GiB (gp2), from snapshot snap-0d3380be
EBS-optimized ⓘ	no
Monitoring ⓘ	no
Tenancy ⓘ	default
Request valid from ⓘ	Now
Request valid until ⓘ	1 year from now
Terminate instances at expiration ⓘ	yes

Cancel Previous Launch

2. Run TF (with screen) and enjoy!

• ssh

- screen

- Start job: python3py; **sudo shutdown now**
- python3 main.py --maxLength=80 | tee out.log; sudo shutdown now
- Ctrl-a d (exit the screen, but Job is going)

- Checking progress: screen -r

- Ctrl-a d (exit the screen)

- When the job is done, the instances will be terminated (save \$)

- But the results will be in the volume

3. Spot Instance termination + EBS Volume



Automatic termination after finishing job! (save \$)

4. Create snapshot from the volume

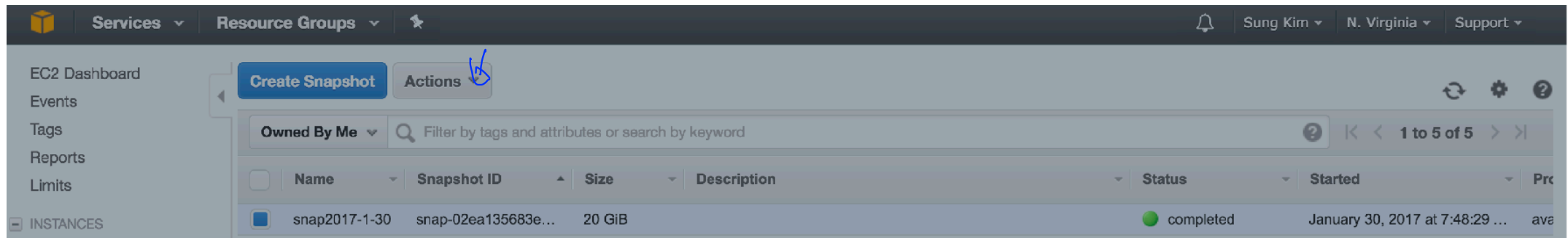
The screenshot shows the AWS Management Console interface for EBS volumes. A modal dialog titled "Create Snapshot" is open, allowing the user to create a snapshot from a selected volume. The background shows a table of EBS volumes with columns for Name, Volume ID, Size, Volume Type, IOPS, Snapshot, Created, Availability Zone, State, and Alarm Status. The first volume, "vol-0c74cb5896968b1b8", is selected and highlighted in blue. The "Create Snapshot" dialog has the following fields:

- Volume**: vol-0c74cb5896968b1b8
- Name**: snap2017-1-30
- Description**: (empty)
- Encrypted**: No

At the bottom of the dialog are "Cancel" and "Create" buttons. The background table shows three volumes:

Name	Volume ID	Size	Volume Type	IOPS	Snapshot	Created	Availability Zone	State	Alarm Status
forspot	vol-0c74cb5896968b1b8	20 GiB	gp2	100 / 3000	snap-c515822f	January 30, 2017 at...	us-east-1d	available	None
	vol-0aeebea...	20 GiB	gp2	100 / 3000	snap-c515822f	January 30, 2017 at...	us-east-1c	in-use	None
	vol-...						us-east-1a	available	None

5. Create AMI from the snapshot



The screenshot shows the AWS Management Console interface. In the top navigation bar, there are links for 'Services', 'Resource Groups', and a user profile 'Sung Kim'. On the left sidebar, there are links for 'EC2 Dashboard', 'Events', 'Tags', 'Reports', 'Limits', and 'INSTANCES'. The main content area shows a 'Create Snapshot' button, which is highlighted with a blue hand icon. Below this button is a table of snapshots. The table has columns for 'Name', 'Snapshot ID', 'Size', 'Description', 'Status', 'Started', and 'Progress'. A single snapshot is listed with the name 'snap2017-1-30', ID 'snap-02ea135683e...', size '20 GiB', and status 'completed'.

Name	Snapshot ID	Size	Description	Status	Started	Progress
snap2017-1-30	snap-02ea135683e...	20 GiB		completed	January 30, 2017 at 7:48:29 ...	ava

Create Image from EBS Snapshot

Name **Description**

Architecture **Virtualization type**

Root device name **Kernel ID**

RAM disk ID

Block Device Mappings

Volume Type	Device	Snapshot	Size (GiB)	Volume Type	IOPS	Throughput (MB/s)	Delete on Termination	Encrypted
Root	/dev/sda1	snap-02ea135683e1fb71d	20	General Purpose SSD (GP2)	100 / 3000	N/A	<input checked="" type="checkbox"/>	Not Encrypted

[Add New Volume](#)

[Cancel](#) [Create](#)


6. Create new spot instance using the AMI

Spot instance launch wizard

Step 1: Find instance types
Step 2: Configure
Step 3: Review

Choose an Amazon Machine Image (AMI)

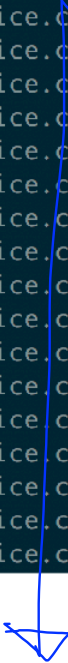
Search by AMI ID or keyword My AMIs 1 to 5 of 5 AMIs

 **AMI-2017-1-20 - ami-199a650f** [Select](#)

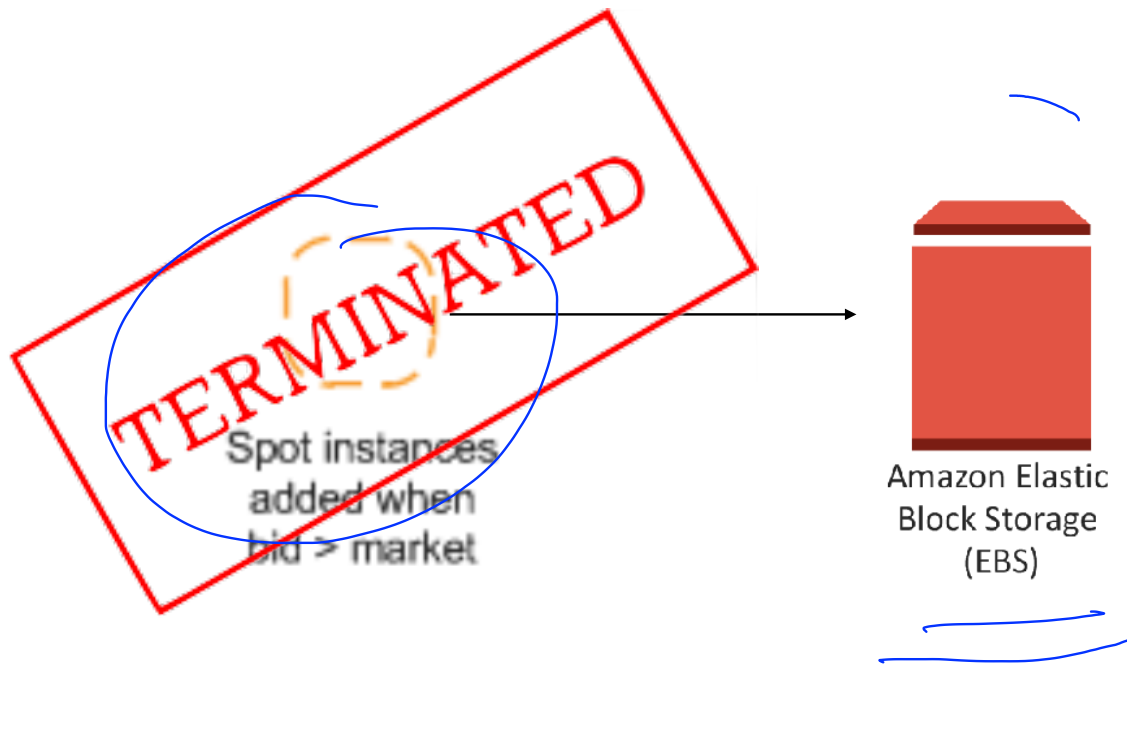
Root device type: ebs Virtualization type: paravirtual Owner: 705119112097 64-bit

GPU, GPU, GPU

```
total memory: 11.17GiB
free memory: 11.11GiB
tensorflow/core/common_runtime/gpu/gpu_device.cc:906] DMA: 0 1 2 3 4 5 6 7
tensorflow/core/common_runtime/gpu/gpu_device.cc:916] 0:  Y Y Y Y Y Y Y Y
tensorflow/core/common_runtime/gpu/gpu_device.cc:916] 1:  Y Y Y Y Y Y Y Y
tensorflow/core/common_runtime/gpu/gpu_device.cc:916] 2:  Y Y Y Y Y Y Y Y
tensorflow/core/common_runtime/gpu/gpu_device.cc:916] 3:  Y Y Y Y Y Y Y Y
tensorflow/core/common_runtime/gpu/gpu_device.cc:916] 4:  Y Y Y Y Y Y Y Y
tensorflow/core/common_runtime/gpu/gpu_device.cc:916] 5:  Y Y Y Y Y Y Y Y
tensorflow/core/common_runtime/gpu/gpu_device.cc:916] 6:  Y Y Y Y Y Y Y Y
tensorflow/core/common_runtime/gpu/gpu_device.cc:916] 7:  Y Y Y Y Y Y Y Y
tensorflow/core/common_runtime/gpu/gpu_device.cc:975] Creating TensorFlow device (/gpu:0) -> (device: 0, name: Tesla K80, pci bus id: 0000:00:17.0)
tensorflow/core/common_runtime/gpu/gpu_device.cc:975] Creating TensorFlow device (/gpu:1) -> (device: 1, name: Tesla K80, pci bus id: 0000:00:18.0)
tensorflow/core/common_runtime/gpu/gpu_device.cc:975] Creating TensorFlow device (/gpu:2) -> (device: 2, name: Tesla K80, pci bus id: 0000:00:19.0)
tensorflow/core/common_runtime/gpu/gpu_device.cc:975] Creating TensorFlow device (/gpu:3) -> (device: 3, name: Tesla K80, pci bus id: 0000:00:1a.0)
tensorflow/core/common_runtime/gpu/gpu_device.cc:975] Creating TensorFlow device (/gpu:4) -> (device: 4, name: Tesla K80, pci bus id: 0000:00:1b.0)
tensorflow/core/common_runtime/gpu/gpu_device.cc:975] Creating TensorFlow device (/gpu:5) -> (device: 5, name: Tesla K80, pci bus id: 0000:00:1c.0)
tensorflow/core/common_runtime/gpu/gpu_device.cc:975] Creating TensorFlow device (/gpu:6) -> (device: 6, name: Tesla K80, pci bus id: 0000:00:1d.0)
tensorflow/core/common_runtime/gpu/gpu_device.cc:975] Creating TensorFlow device (/gpu:7) -> (device: 7, name: Tesla K80, pci bus id: 0000:00:1e.0)
```



Solution: Spot Instance + EBS Volume



Automatic termination after finishing job! (save \$)

TensorFlow-KR 회원을 위한 AWS 크레딧 제공!



<http://bit.ly/awskr-feedback>

딥러닝 연구 및 학습
AWS 100달러 무료 크레딧 제공

등록하시면 패키지를 받으실 수 있는 URL 및 AWS 학습 정보를 이메일로 보내드립니다!