

```
*****  
**                               **  
*****  
                               Setup BEACLS for AWS  
*****
```

[ A. Use preinstalled image ]

1. Launch an instance of the preinstalled image

- Log in AWS
- Request a spot instance
  - Select ami-3cbc082a0 as AMI image
  - Select p2.xlarge as instance types
  - Make and download a key to access the instance
- Access to the P2 instance
  - Check the ip address of your instance in AWS console.
  - Log in the instance  
\$ ssh -i ~/path/yourprivatekey.pem ubuntu@[ipaddress]

GOTO step 5 in B scenario

[ B. Make the environment from scratch ]

1. Make AMI image for P2 instance

- Log in AWS and launch a P2 instance
  - Select N. Virginia DC (cheaper than other locations)
  - Use Ubuntu AMI ami-09b3691f (Ubuntu 16.04 LTS hvm:ebs-ssd)
  - Select p2.xlarge instance
  - Make key to access (or use existing key) the instance
  - Select SSD as much as you want, such as root storage size (20GB) and additional SSD(100GB)
- Access to the P2 instance
  - Check the ip address of your instance in AWS console.
  - Log in the instance  
\$ ssh -i ~/path/yourprivatekey.pem ubuntu@[ipaddress]
- Disk format for the additional SSD (Optional)
  - Check disk name (/dev/xvdba in this case) and mount it  
\$ sudo fdisk -l  
\$ sudo mkfs -t ext4 /dev/xvdba  
\$ sudo mount /dev/xvdba /home/ubuntu/simulation/  
\$ sudo chown -R ubuntu:ubuntu simulation

2. Install CUDA

- Install libraries for CUDA 8.0  
\$ sudo apt-get update

```

$ sudo apt-get upgrade
$ wget
http://developer.download.nvidia.com/compute/cuda/repos/ubuntu1604/x86_64/7fa2af80.pub
b
$ cat 7fa2af80.pub | sudo apt-key add -
$ wget
http://developer.download.nvidia.com/compute/cuda/repos/ubuntu1604/x86_64/cuda-repo-ubuntu1604_8.0.44-1_amd64.deb
$ sudo dpkg -i cuda-repo-ubuntu1604_8.0.44-1_amd64.deb
$ sudo apt-get update
$ sudo apt-get install linux-generic
$ wget
http://us.download.nvidia.com/XFree86/Linux-x86_64/352.99/NVIDIA-Linux-x86_64-352.99.run
un
$ sudo chmod +x NVIDIA-Linux-x86_64-352.99.run
$ sudo ./NVIDIA-Linux-x86_64-352.99.run
$ sudo apt-get install cuda
$ sudo reboot

$ sudo apt-get remove linux-virtual
$ sudo apt-get autoremove

- Set variables to .bashrc
$ vi ~/.bashrc
export PATH="/usr/local/cuda-8.0/bin:$PATH"
export LD_LIBRARY_PATH="/usr/local/cuda-8.0/lib64:$LD_LIBRARY_PATH"

- Disable display manager
$ vi /etc/default/grub # L12
GRUB_CMDLINE_LINUX="systemd.unit=multi-user.target"
$ sudo update-grub
$ sudo reboot

```

### 3 GPU test

- Check
 

```
$ nvidia-smi
```
- Performance test
 

```
$ cd /usr/local/cuda-8.0/samples
$ sudo make
$ cd bin/x86_64/linux/release
$ sudo ./bandwidthTest
```

### 4. Install libraries for BEACLS

- Install zlib, boost, OpenCV and hdf5
 

```
$ sudo apt-get update
```

```
$ sudo apt-get install libhdf5-dev libboost-dev libopencv-core-dev
libopencv-highgui-dev
$ sudo apt-get install libopencv-dev
```

- Make a patch for AWS  
\$ vi ~/update\_sources.sh

```
-----update_sources.sh-----
```

```
#!/bin/sh
```

```
sdir=./beacIs/sources
```

```
if [ "${sdir}" = "" ]; then
    echo "Source directory is not specified"
    exit
fi
```

```
if [ ! -e ${sdir} ]; then
    echo "No such a directory: ${sdir}"
    exit
fi
```

```
cd ${sdir}
```

```
find ./ -type f -print | grep Makefile$ | xargs grep -l nvidia-361 | xargs sed -i
"s/nvidia-361/nvidia-375/g"
```

```
find ./ -type f -print | grep Makefile$ | xargs grep -l sm_52 | xargs sed -i "s/GPU_ON =
N\r\n/GPU_ON = N\r\nGPU_SMS ?= 30 35 37 50 52 60\r\n/g"
```

```
find ./ -type f -print | grep Makefile$ | xargs grep -l sm_52 | xargs sed -i "s/NVCCFLAGS +=
-arch=sm_52 -maxrregcount=64/NVCCFLAGS += -maxrregcount=64\r\n\r\n$(foreach
sm,${GPU_SMS},${eval NVCCFLAGS += -gencode
arch=compute_$(sm),code=sm_$(sm)})/g"
```

```
find ./ -type f -print | grep Makefile$ | xargs grep -l sm_52 | xargs sed -i "s/NVLDLFLAGS +=
-arch=sm_52 -maxrregcount=64/NVLDLFLAGS += -maxrregcount=64/g"
```

```
-----
***** My AMI was made here (ami-3cbc082a0) through AWS console *****
My AMI image does not include the additional SSD because it costs about $0.05/GB-month
to keep the AMI image
*****
```

## 5. Execute BEACLS

- Get and compile BEACLS sources  
\$ mkdir ~/BEACLS; cd ~/BEACLS

```

$ git clone https://github.com/HJReachability/beacIs
$ ~/update_sources.sh
$ cd beacIs/sources
$ make GPU_ON=Y NVCC=/usr/local/cuda/bin/nvcc all

```

- Build and execute a sample
 

```

$ cd samples/DubinsCar_RS
$ make test

```

 if you want to use GPU,
 

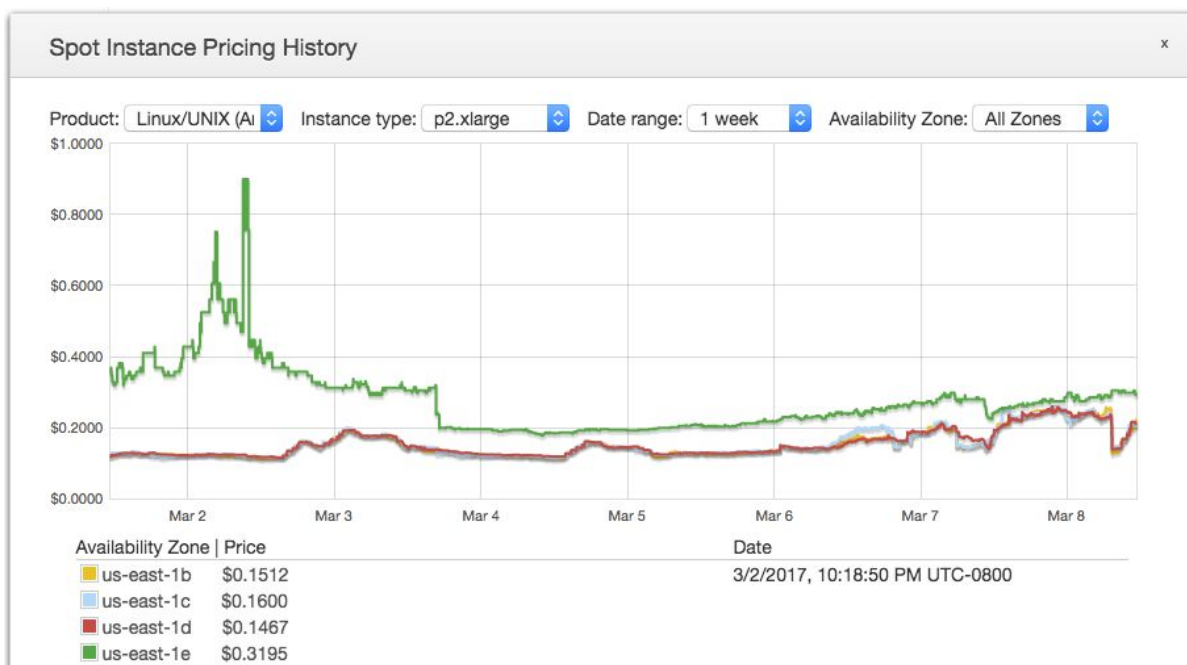
```

$ make USE_CUDA=1 test

```

\*\*\*\*\*  
 \*\* Appendix \*\*  
 \*\*\*\*\*

**- Spot Instance Pricing**



- **SSD pricing**  
 General Purpose SSD: \$0.12 (depends on region) per GB-month for storage you provision.  
 ex.) 100GB-SSD 2 hours everyday = \$0.8 per month

**- Bandwidth Test Result**

---

[CUDA Bandwidth Test] - Starting...  
Running on...

Device 0: Tesla K80  
Quick Mode

Host to Device Bandwidth, 1 Device(s)  
PINNED Memory Transfers  
Transfer Size (Bytes) Bandwidth(MB/s)  
33554432 10877.6

Device to Host Bandwidth, 1 Device(s)  
PINNED Memory Transfers  
Transfer Size (Bytes) Bandwidth(MB/s)  
33554432 12024.0

Device to Device Bandwidth, 1 Device(s)  
PINNED Memory Transfers  
Transfer Size (Bytes) Bandwidth(MB/s)  
33554432 156653.2

Result = PASS

---

#### - Single-precision Benchmark

---

```
$ ./nbody -benchmark -numbodies=200000 -numdevices=1
```

----

```
> Compute 3.7 CUDA device: [Tesla K80]  
Warning: "number of bodies" specified 200000 is not a multiple of 256.  
Rounding up to the nearest multiple: 200192.  
200192 bodies, total time for 10 iterations: 5232.908 ms  
= 76.586 billion interactions per second  
= 1531.723 single-precision GFLOP/s at 20 flops per interaction
```

---

#### - AWS GPU instance

---

G2 instance  
g2.2xlarge  
1GPU(4GB Mem), 8vCPU, 15GiB Mem  
Chip : NVIDIA Tesla K10  
FLOPS: 45.8/0.19 TFLOPS  
Price : \$0.12/hour spot instance, the price varies

\$0.65/hour for on-demand instance

P2 instance - new!

p2.xlarge

1GPU(12GB Mem), 4vCPU, 61GiB Mem

Chip : NVIDIA Tesla K80

FLOPS: 8.74/2.91 TFLOPS

Price : \$0.13/hour for spot instance, the price varies.

\$0.9/hour for on-demand instance

...

GPU of P2 instance is up to 16.

---

---

Development Memo

1. Git config

\$git config --global core.autoCRLF false