AWS GPU Instances Ubuntu 18.04

Install Ubuntu Packages:

```
$ sudo apt-get update
$ sudo apt-get install build-essential
$ sudo ln -n /usr/bin/python3.6 /usr/bin/python
$ sudo apt install ocl-icd-opencl-dev
$ sudo apt install -y opencl-headers
$ sudo apt install clinfo
$ sudo apt-get install linux-headers-$(uname -r)
```

Install AWS Packages and configure S3 credentials:

```
$ sudo apt  install awscli
$ aws configure
AWS Access Key ID [None]:
AWS Secret Access Key [None]:
Default region name [None]: eu-west-1
Default output format [None]: json
```

Install TornadoVM Prerequisites:

1. Install Openjdk 8:

```
$ mkdir env
$ wget
https://github.com/graalvm/graal-jvmci-8/releases/download/jvmci-20.2-b03/openj
dk-8u262+10-jvmci-20.2-b03-linux-amd64.tar.gz -P /tmp
$ tar -xf /tmp/openjdk-8u262+10-jvmci-20.2-b03-linux-amd64.tar.gz -C
/home/ubuntu/env/
```

2. Install Maven 3.6.3:

```
$ wget
https://www-us.apache.org/dist/maven/maven-3/3.6.3/binaries/apache-maven-3.6.3-
bin.tar.gz -P /tmp
wget
https://downloads.apache.org/maven/maven-3/3.9.2/binaries/apache-maven-3.9.2-bi
n.tar.gz -P /tmp
$ sudo tar xf /tmp/apache-maven-3.6.3-bin.tar.gz -C /opt
$ sudo ln -s /opt/apache-maven-3.6.3 /opt/maven
$ sudo vim /etc/profile.d/maven.sh

export JAVA_HOME=/home/ubuntu/env/openjdk1.8.0_262-jvmci-20.2-b03
export M2_HOME=/opt/maven
export MAVEN_HOME=/opt/maven
export PATH=${M2_HOME}/bin:${PATH}
```

```
$ sudo chmod +x /etc/profile.d/maven.sh
$ source /etc/profile.d/maven.sh
```
You can test the maven version:
```
$ mvn -version
```
This should report the following:
```
Apache Maven 3.6.3 (cecedd343002696d0abb50b32b541b8a6ba2883f)
Maven home: /opt/maven
Java version: 1.8.0_262, vendor: GraalVM Community, runtime:
/home/centos/env/openjdk1.8.0_262-jvmci-20.2-b03/jre
Default locale: en_GB, platform encoding: UTF-8
OS name: "linux", version: "3.10.0-1160.6.1.el7.x86_64", arch: "amd64", family:
"unix”
```
Create the m2 directory for maven:
```
$ mkdir ~/.m2
$ vim ~/.m2/settings.xml
```
Copy the following, and adjust and differences in the path of JAVA_HOME or the Java version:
```
<settings xmlns="http://maven.apache.org/SETTINGS/1.0.0"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://maven.apache.org/SETTINGS/1.0.0
        https://maven.apache.org/xsd/settings-1.0.0.xsd">
 <interactiveMode/>
 <usePluginRegistry/>
        <offline/>
                <pluginGroups/>
         <servers/>
        <mirrors/>
         <proxies/>
         <profiles>
         <profile>
                <id>tornado-jvmci</id>
                <activation>
                <activeByDefault>true</activeByDefault>
                </activation>
                <properties>
                        <!-- Your PATH TO JDK with JVMCI Support -->
<jvmci.root>/home/ubuntu/env/openjdk1.8.0_262-jvmci-20.2-b03</jvmci.root>
                        <jvmci.version>1.8.0_292</jvmci.version>
                </properties>
         </profile>
         </profiles>
         <activeProfiles/>
</settings>
```

3. Install cmake:

```
$ wget
https://github.com/Kitware/CMake/releases/download/v3.25.2/cmake-3.25.2-linux-x
86_64.tar.gz -P /tmp
$ sudo tar -xf /tmp/cmake-3.25.2-linux-x86_64.tar.gz -C /usr/local
$ vim ~/.bashrc
```
Add the following in the file:
```
export PATH="$PATH:/usr/local/cmake-3.25.2-linux-x86_64/bin"
```

```
$ source ~/.bashrc
$ cmake --version
```

Install TornadoVM and run on GPU:

A) Install TornadoVM

```
$ git clone https://github.com/beehive-lab/TornadoVM.git
$ cd TornadoVM
$ vim etc/sources.env
```

Add the following in the file:

```
#!/bin/bash
# OpenJDK
export JAVA_HOME=/home/ubuntu/env/openjdk1.8.0_262-jvmci-20.2-b03
export PATH=$PWD/bin/bin:$PATH    ## This directory will be automatically
generated during Tornado compilation
export TORNADO_SDK=$PWD/bin/sdk   ## This directory will be automatically
generated during Tornado compilation
export CMAKE_ROOT=/usr/local/cmake-3.19.8-Linux-x86_64            ## or
<path/to/cmake/cmake-3.10.2> (see step 4)
```

B) Build TornadoVM

```
$ source etc/sources.env
$ make
$ tornado --devices
```

Install GPU Driver on AWS

- Install GPU Driver on G4 instances

To install the AMD driver for ***g4ad.4xlarge*** (AMD Radeon Pro V520), do the following:
https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/install-amd-driver.html#download-amd-driver
Add the following in the `~/.bashrc` file:

```
export PATH=/opt/amdgpu-pro/bin/:$PATH
export LD_LIBRARY_PATH=/opt/amdgpu-pro/lib64/
```

To install the Nvidia driver for ***g4dn.4xlarge*** (Tesla T4 GPU), do the following:
https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/install-nvidia-driver.html
https://www.nvidia.com/Download/driverResults.aspx/139178/en-us

```
$ sudo sh NVIDIA-Linux-x86_64-450.119.04.run
```

For CUDA ToolKit:
https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/install-nvidia-driver.html#gpu-instance-install-cuda

```
$ wget
https://developer.download.nvidia.com/compute/cuda/repos/ubuntu1804/x86_64/cuda
-ubuntu1804.pin
$ sudo mv cuda-ubuntu1804.pin /etc/apt/preferences.d/cuda-repository-pin-600
$ sudo apt-key adv --fetch-keys
https://developer.download.nvidia.com/compute/cuda/repos/ubuntu1804/x86_64/7fa2
af80.pub
$ sudo add-apt-repository "deb
https://developer.download.nvidia.com/compute/cuda/repos/ubuntu1804/x86_64/ /"
$ sudo apt-get update
$ sudo apt-get -y install cuda
```

https://developer.nvidia.com/cuda-downloads?target_os=Linux&target_arch=x86_64&Distribution=Ubuntu&target_version=18.04&target_type=deb_network

```
$ export PATH=/usr/local/cmake-3.19.8-Linux-x86_64/bin:$PATH
$ export PATH=/usr/local/cuda-11.3/bin/:$PATH
$ export CUDA_PATH=/usr/local/cuda-11.3
$ export CPLUS_INCLUDE_PATH=${CUDA_PATH}/include:${CPLUS_INCLUDE_PATH}
```

Do the following to resolve problem with missing CUDA library:

```
$ cd /usr/lib/x86_64-linux-gnu/
$ sudo ln -s libcuda.so.465.19.01 libcuda.so.1
$ sudo ln -s libcuda.so.1 libcuda.so
```

- Install GPU Driver on G5g instances

To install the Nvidia driver for ***g5g.xlarge*** (Tesla NVIDIA T4G Tensor Core GPU), do the following as mentioned here:

- Download the driver for arm64 and install it.
- Then follow the link, to install CUDA:

TornadoVM installation for G5G instances

$ sudo apt install python3-pip -y
$ python3 -m pip install wget

Setup OpenCL

$ cd /usr/lib/aarch64-linux-gnu/
$ sudo ln -s libOpenCL.so.1 libOpenCL.so

CUDA Setup

$ cd /usr/lib/aarch64-linux-gnu/
$ sudo ln -s libcuda.so.1 libcuda.so

Install TornadoVM

```
tornado --devices
WARNING: Using incubator modules: jdk.incubator.vector, jdk.incubator.foreign

Number of Tornado drivers: 1
Driver: PTX
  Total number of PTX devices  : 1
  Tornado device=0:0  (DEFAULT)
    PTX -- PTX -- NVIDIA T4G
            Global Memory Size: 14.6 GB
            Local Memory Size: 48.0 KB
            Workgroup Dimensions: 3
            Total Number of Block Threads: [2147483647, 65535, 65535]
            Max WorkGroup Configuration: [1024, 1024, 64]
            Device OpenCL C version: N/A
```

Report PTX tests:

```
#AWS G5G instance (ARM + Tesla T4 GPU)
$ make tests

Number of Tornado drivers: 1
Driver: PTX
  Total number of PTX devices  : 1
  Tornado device=0:0  (DEFAULT)
   PTX -- PTX -- NVIDIA T4G
           Global Memory Size: 14.6 GB
           Local Memory Size: 48.0 KB
           Workgroup Dimensions: 3
           Total Number of Block Threads: [2147483647, 65535, 65535]
           Max WorkGroup Configuration: [1024, 1024, 64]
           Device OpenCL C version: N/A

=================================================
              Unit tests report
=================================================

{'[PASS]': 464, '[FAILED]': 15, '[UNSUPPORTED]': 48}
Coverage [PASS/(PASS+FAIL)]: 96.87%
Coverage [PASS/(PASS+FAIL+UNSUPPORTED)]: 88.05%

=================================================

Total Time(s): 176.08120608329773
```

```
```