Unity ML VM (Linux)

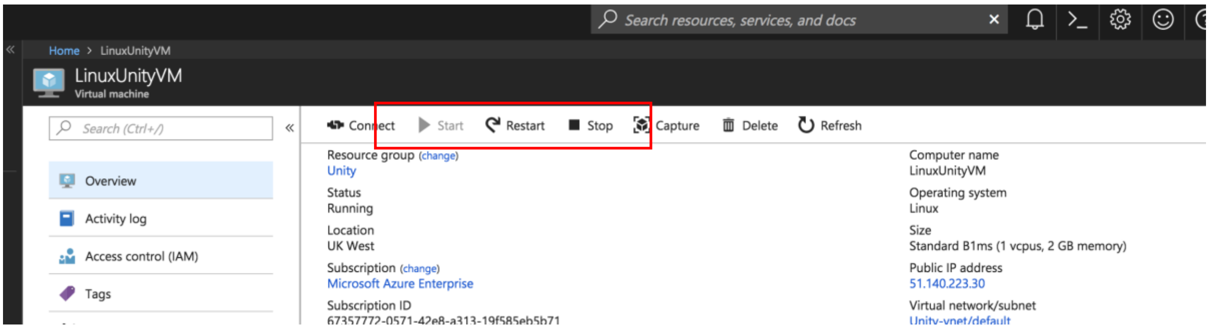
This document has been written assuming knowledge of basic ML agents knowledge. Make sure you have read through the basic guide page on the GitHub before you get to this stage.

Currently, although some initial testing and research was carried out, the easiest way to run a headless unity VM for ML model training is on a Linux box. A box is already setup on the team azure account titled ‘LinuxUnityVM’ which has all dependencies as well as unity editor [Linux] 2018.2.5f.

Azure Account details:
U: rnd.kainos@outlook.com
P: Oranges6

Details about the unity editor Linux versions can be found here. Instructions to install versions and new releases will be released on that page. All installs should preferably use command line only.

Once you have turned on the VM through the azure portal here



Resource group
Unity

Status
Running

Location
UK West

Subscription
Microsoft Azure Enterprise

Subscription ID
67357772-0571-42e8-a313-19f585eb5b71

Computer name
LinuxUnityVM

Operating system
Linux

Size
Standard B1ms (1 vcpus, 2 GB memory)

Public IP address
51.140.223.30

Virtual network/subnet
Unity-vnet/default

SSH into the machine by using the command in an SSH compatible program (Terminal for mac/linux, PuTTY for windows) rnd@51.140.223.30 and login using the credentials below.

VM login details:
U: rnd
P: Oranges6

To get your project onto the machine, I recommend just simply using git. All projects should already be a git repo and it’s simply a case of cloning them onto the machine after exporting a **64bit** Linux build to the repo. The build must be 64bit otherwise it will not run at all on the VM. All ML agents dependencies are installed on the machine already so there should be no additional dependencies required other than ones you have required yourself.

Once the repo is downloaded with the build you must be sure to have all require files/folders for your build and the ML agents python folder. The build should come out as a .x64 or .x86_64 file. You must make the build executable before we can do anything.

For those new to Linux, some files will be downloaded an require us to explicitly state that they are executable. To do this through command line, run the command

```
chmod +x ./file
```

To run ML agents training you need to simultaneously run the python learn as well as the unity build. To do this, we can use the screen function of Unix:

```
$ screen
[welcome screen appears]
$ cd <github clone>
$ [ctrl+A] [C]
$ cd <github clone>/python
```

Now you can use the command [ctrl + A][A] to switch between the two SSH screens. The alternative to this is to simply open up 2 SSH terminals using the same login.

After you are in both files, to run the training model, in the python terminal use the command:

```
$ learn.py --run-id=<num> --train
```

And in the terminal you wish to run unity, use the command:

```
$ ./file -nographics -batchmode
```

Once the brain has finished training, it is simply a case of pushing the new model to the GitHub repo and pulling it or downloading the file to wherever you need it.

Issues

Screenshot feature request:

3 main ways to try:

- Linux box as usual completely headless but force a render
  - Render texture not supported
  - Issues with unity player not detecting the [virtual] screen
- Windows box RDP
  - Will work but requires hands on RDP
- Windows box bash script to activate everything
  - May work but connecting to cmd of VM may be an issue
  - Retrieving files may be an issue as windows doesn’t believe in SSH
- ?