AWS and OpenAI gym Tutorial

10-703: Deep Reinforcement Learning: Recitation I
Objectives for Today

- What is AWS
- How do we use it (safely)!?
- What is OpenAI Gym
- How do we use it?
Objectives for Today

- What is AWS
- How do we use it (safely)!?
- What is OpenAI Gym
- How do we use it?
Amazon Web Services

- On Demand Cloud Computing Resource
  - Compute
  - Storage
  - Databases
  - Analytics
  - Networking
  - Mobile
  - Developer Tools
  - Management Tools
  - IoT
Amazon Web Services

- On Demand Cloud Computing Resource
  - Compute - EC2
  - Storage
  - Databases
  - Analytics
  - Networking
  - Mobile
  - Developer Tools
  - Management Tools
  - IoT
Login to your Account
  ○ use your Andrew ID, preferably
Amazon Web Services - EC2

- Click on Services
Amazon Web Services - EC2

- Search for “billing”
TODO

- Set spending alarm under the **Budget** section
- Add credit using the promo code we provide under the **Credits** section
- Check live spending under the **Bills** section
Amazon Web Services - Dashboard

- Go to Launch Instance
AMIs have pre-installed deep learning frameworks

- Go to Community AMIs and choose the one that fits your need.
Amazon Web Services - Choose Instance Type

- Which instance to choose?
# Amazon Web Services - Choose Instance Type

<table>
<thead>
<tr>
<th>Instance Name</th>
<th>Number/GPU Type</th>
<th>Price (on-demand) per hour</th>
<th>Architecture /Speed</th>
</tr>
</thead>
<tbody>
<tr>
<td>p2.xlarge</td>
<td>1 K80</td>
<td>$0.9</td>
<td>Kepler (Slow)</td>
</tr>
<tr>
<td>p2.8xlarge</td>
<td>8 K80</td>
<td>$7.2</td>
<td>Kepler (Slow)</td>
</tr>
<tr>
<td>p2.16xlarge</td>
<td>16 K80</td>
<td>$14.4</td>
<td>Kepler (Slow)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Instance Name</th>
<th>Number/GPU Type</th>
<th>Price (on-demand) per hour</th>
<th>Architecture/ Speed</th>
</tr>
</thead>
<tbody>
<tr>
<td>g3.4xlarge</td>
<td>1 M60</td>
<td>$1.14</td>
<td>Maxwell (medium speed)</td>
</tr>
<tr>
<td>g3.8xlarge</td>
<td>2 M60</td>
<td>$2.28</td>
<td>Maxwell (medium speed)</td>
</tr>
<tr>
<td>g3.16xlarge</td>
<td>4 M60</td>
<td>$4.56</td>
<td>Maxwell (medium speed)</td>
</tr>
</tbody>
</table>
## Amazon Web Services - Choose Instance Type

<table>
<thead>
<tr>
<th>Instance Name</th>
<th>Number/GPU Type</th>
<th>Price (on-demand) per hour</th>
<th>Architecture/Speed</th>
</tr>
</thead>
<tbody>
<tr>
<td>p3.2xlarge</td>
<td>1 V100</td>
<td>$3.06</td>
<td>Volta (fastest architecture so far!)</td>
</tr>
<tr>
<td>p3.8xlarge</td>
<td>4 V100</td>
<td>$12.24</td>
<td>Volta (fastest architecture so far!)</td>
</tr>
<tr>
<td>p3.16xlarge</td>
<td>8 V100</td>
<td>$24.48</td>
<td>Volta (fastest architecture so far!)</td>
</tr>
</tbody>
</table>
Amazon Web Services - Configure Instance

- Check “Request Spot Instances”
- Enter maximum price of $1
Amazon Web Services - Add Storage

Three types:

- EBS: Local storage per instance
- EFS: Common file sharing among instances
- S3 buckets

Configure EBS before launching the Instance

EFS can be configured/accessed once we are within the instance
Only three commands to mount EFS:

- `sudo apt-get install nfs-common`
- `sudo mkdir efs`
- `sudo mount -t nfs4 -o nfservers=4.1, rsize=1048576, wsize=1048576, hard, timeo=600, retrans=2 fs-ff38fd86..efs.us-east-2.amazon.aws.com:/ efs`
Keep in mind that ~50GB of the default 75GB is occupied by AMI Image files.
Amazon Web Services - Review & Launch

- Time to Launch
First time users: Select create a new key pair from drop down menu

Download key and put it in some location on your machine (e.g., ~/.aws/key_name.pem)
Amazon Web Services - Launch Instance

- Click on Connect
• Remember to change permissions for the key file
• ssh into the instance from your terminal: use the complete path of the key file
Objectives for Today

- What is AWS
- How do we use it (safely)!?
- What is OpenAI Gym
- How do we use it?
OpenAI Gym
To solve such a problem, you need the ability to:

- Define the environment
- Generate samples from the environment
- Sample an action from the action space
- Retrieve the next state after taking an action
- Retrieve the reward of taking an action
- Check if the episode has ended
- Reset the episode when the episode ends

OpenAI gym gives you the ability to:

- DO ALL OF THESE THINGS!
OpenAI Gym - Getting Started

Install directly using pip:
- pip install gym

Build from source
- git clone https://github.com/openai/gym
- cd gym
- pip install -e . #minimal install

Add new environment
- cd gym
- pip install -e [box2d] #installs all box2d environments

Install directly using pip:
- pip install gym
OpenAI Gym - Basic Concepts

Define the environment

- env = gym.make('LunarLander-v2')

Sample an action from the action space

- action = env.action_space.sample()

Reset the episode when the episode ends

- state = env.reset()

Retrieve the next state, reward and the indicator of the episode termination

- next_state, reward, done, info = env.step(action)
OpenAI Gym - Additional Features

Render the environment

- `env.render()`

Record the environment

- `env = gym.wrapper.Monitor(env, '.', force=True)`

Check out the state space and action space

- `Print (env.action_space)`
- `Print (env.observation_space)`
Let’s look at a demo.