

- We will use the Anaconda Distribution
 - simple to install
 - comprehensive
 - package and environment management
 - No more `pip install` followed by dependency chasing

Set up a ML environment

Several choices

- Anaconda on your own machine
- Anaconda on AWS (or Azure, Google)
- Turn-key, cloud solution: Floyhud
- Turn-key, cloud solution: Paperspace

Anaconda on your own machine

- Pro: cheapest
- Con: potential limited by memory and power of your machine

- Link: [Download Anaconda \(https://www.anaconda.com/download/#linux\)](https://www.anaconda.com/download/#linux), and run installer
 - if no browser available
 - save link, e.g., https://repo.continuum.io/archive/Anaconda3-2018.12-Linux-x86_64.sh (https://repo.continuum.io/archive/Anaconda3-2018.12-Linux-x86_64.sh).
 - use wget on the link:

wget

https://repo.continuum.io/archive/Anaconda3-2018.12-Linux-x86_64.sh

- run the downloaded file: `bash Anaconda3-2018.12-Linux-x86_64.sh`
 - accept defaults
 - allow your `.bashrc` to be updated:
 - or can do it later yourself: `source .bashrc`
-

Anaconda on AWS

- Same setup as Anaconda on your own machine **ONCE** you have knowledge of how to create machines on AWS
- Pro:
 - high potential: you can rent machines with increased power, memory and GPU !
 - knowing how to use a cloud services (AWS, Azure, Google) is a valuable skill !
- Con: Free-tier machine good to start but need to rent resources (i.e., money)

Links

- [Grant McKinnon \(http://www.grant-mckinnon.com/\)](http://www.grant-mckinnon.com/)
 - [Setting up AWS for Kaggle \(http://www.grant-mckinnon.com/?p=6%E2%80%8B\)](http://www.grant-mckinnon.com/?p=6%E2%80%8B)

Floydhub

- Pro:
 - **Turn-key** and cloud-based. No installation to start
- Con:
 - Best as a Jupyter notebook server, not as a full-service machine
 - You **WILL** want a text editor at some point, particularly as you develop Python Classes/Modules

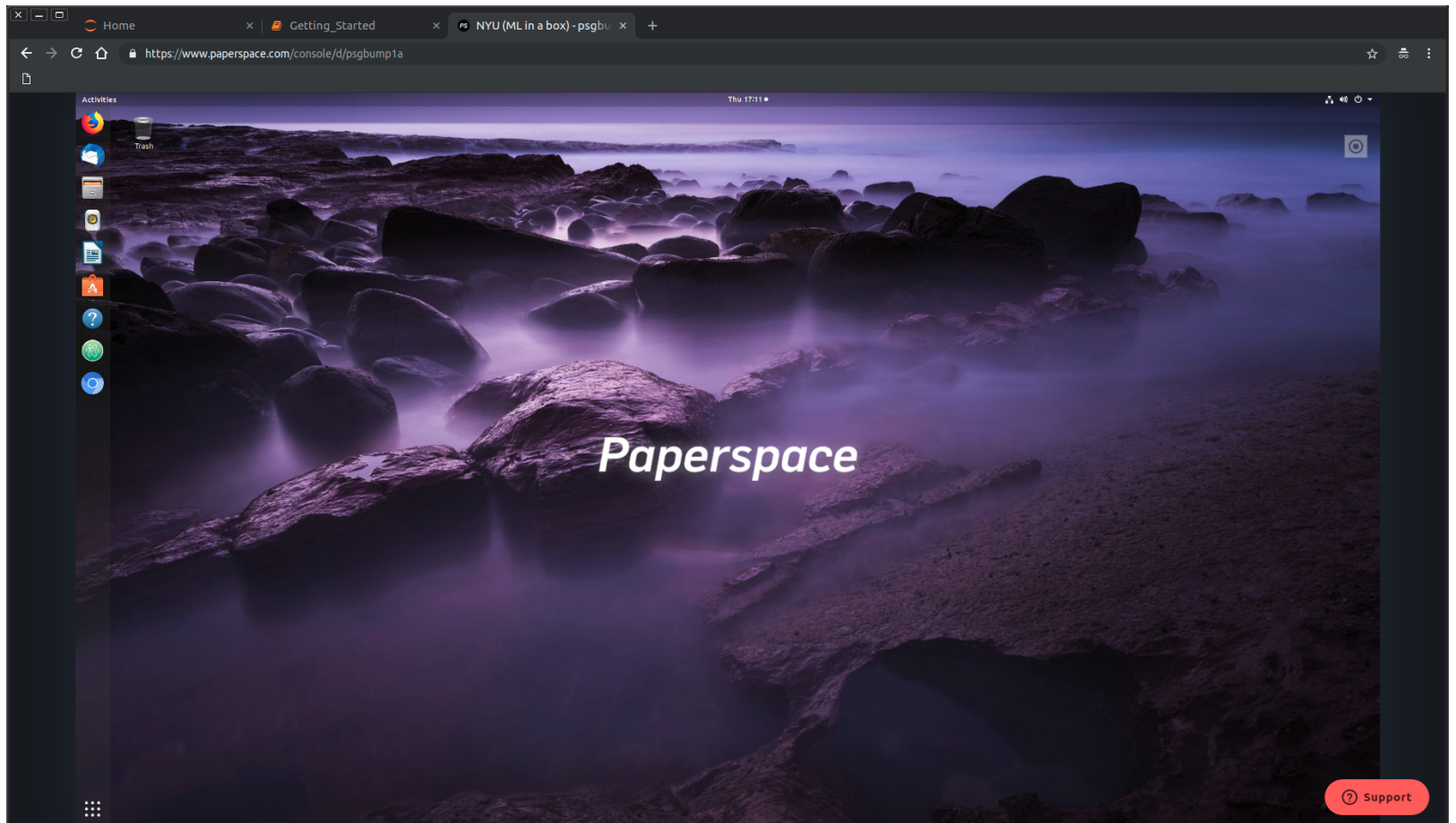
Paperspace

- Pro:
 - Notebook server is **Turn-key** and cloud-based. No installation to start
 - "ML in a box": Full featured machine (like AWS)
 - BUT only advanced frameworks (e.g., TensorFlow) not sklearn !
- Con:
 - Cost to rent is much higher than AWS
 - n.b., some providers use AWS as a back-end; you pay for convenience

- Create account on `paperspace.com`
- Select the "CORE" product: Create a Cloud VM
 - this gives you a full virtual machine (VM), that you can interact with in your local browser
- Select "New Machine" to create a new VM
 - Select a region (East Coast)
- Scroll down to find "Choose OS"
 - Click on "Public Templates" and choose "Ubuntu 18.04, ML-in-a-Box"
 - Scroll down to "Choose Machine (hourly)" and choose the cheapest machine
 - \$0.51/hour gets you a hefty machine (more than you need !)
 - 8 CPU's
 - GPU
 - 30GB RAM
- Scroll down to find "Choose Machine": hourly

- Choose Storage (I choose 50GB @ \$5/month)
- Default Network
- Auto Snapshot
 - you can choose to have snapshots of your machine taken at a frequency of your choosing
 - not really necessary but it might give you a feeling of safety
 - allows you to revert back to a previous state
 - it incurs storage charges (equal to size of your machine), so it's not free!
 - Each 50GB machine snapshot costs \$1/month

- You will stop your machine when you're finished
- It will stay around
- Next time: choose "Launch Console" from upper right of main page
- You will see your stopped machine. Launch it.
 - You may need to "Start" machine again by choosing from menu (circle on upper right)
 - If the desktop is not active:
 - Go back to Launch Console
 - Choose the setting widget (looks like a gear) on you machine
 - Choose to Launch Desktop



- Start a terminal
 - Click on "Show applications" (checkerboard on lower left)
 - Find "Terminal" and click
- You're in !