Reinforcement Learning for RecSys - from Bandits to Offline RL with Ray RLlib

Jun Gong jungong@anyscale.com
Kourosh Hakhamaneshi kourosh@anyscale.com

$whoami (Kourosh)

→ RL Team Lead @Anyscale.

→ Previously: PhD student at UC Berkeley working on RL in Robotics and design optimization

$whoami (Jun)

→ Senior Member of Engineering team @Anyscale.

→ Previously: Engineer at Google and Facebook. Distributed RL at Loon.

RLlib

ODSC

Anyscale

Who we are: Original creators of Ray, a unified framework for scalable, distributed computing. Part of that framework are our libraries for ML and data processing.

What we do: Scalable compute for AI and Python

Why we do it: Scaling is a necessity, scaling is hard; make distributed computing easy and simple for all developers.

Some of RLlib's Industry Users

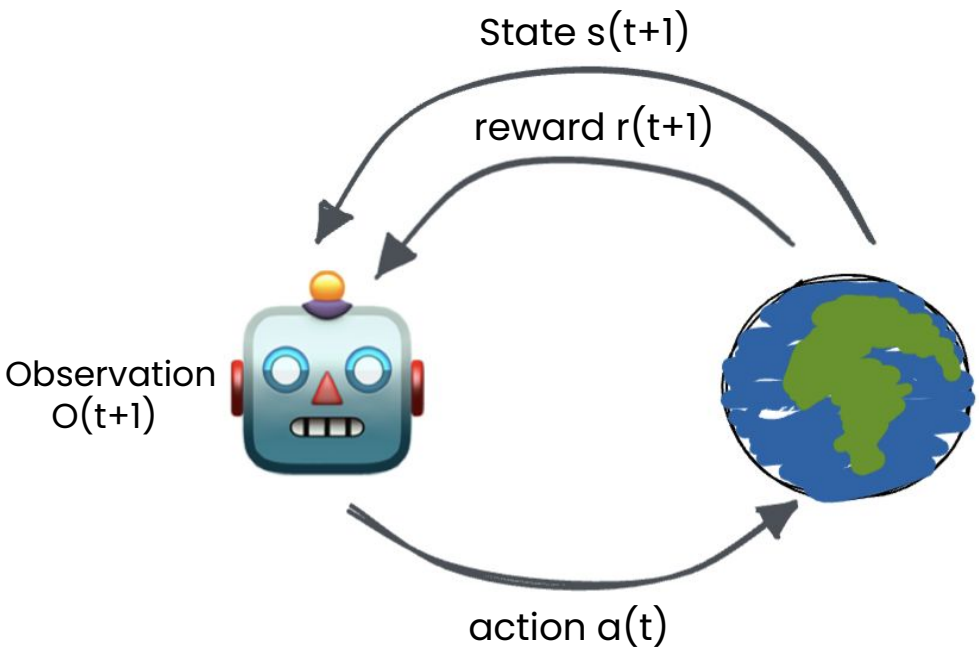
Overview of the tutorial

- Brief intro RL
- Brief intro RecSys
  - Traditional Approaches
  - Defining RecSys as an RL problem
- Online RL vs Offline RL
- Hands-on coding with python notebooks and scripts
- Thank you! Connect with us!

Goals – Understand:

- What are the advantages of using RL in RecSys?
- What are the pros and cons of offline RL in practical scenarios?

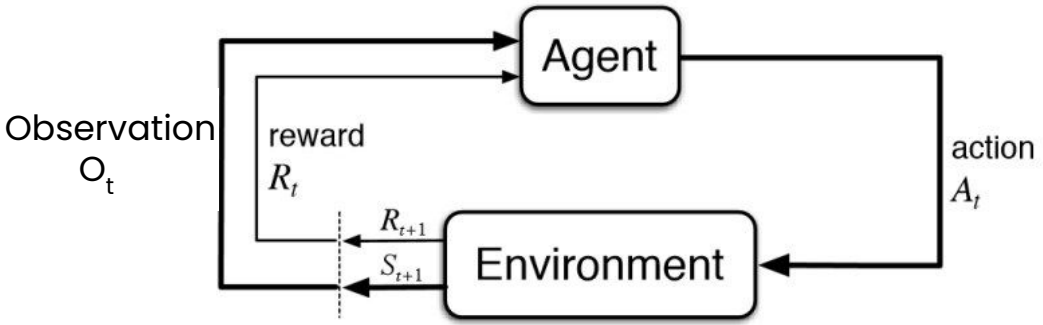
Brief intro RL



Algorithm

$$\max_{\pi} \mathbb{E}_{\pi} \left[\sum_{t} r(s_t, a_t) \right]$$

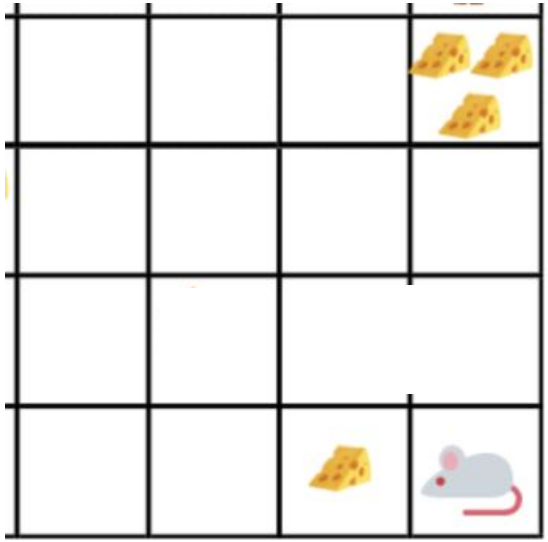
Brief intro RL formalization



$$(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma)$$

Discount factor γ in RL

- If γ **= 0**, the algorithm considers **1-step rewards only**.
- If γ **= 1,** the algorithm considers all future rewards equally.

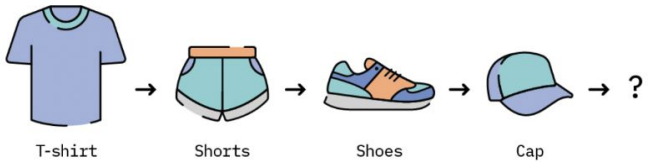


Brief intro RecSys

Companies want to recommend content.

ML: Pointwise recommendations.

RL: Combine pointwise recommendations with session based data.



Traditional ML in RecSys

- Traditional ML (collaborative filtering) models are **static with respect to time**.
  - Ignores the **sequence of interactions** with a given user.
- Static models can be:
  - Too short-sighted and **miss out on Long-term, delayed rewards**
  - **Overlook important and changing user intents** or business conditions such as seasonality or promotional campaigns

New way: RL in RecSys

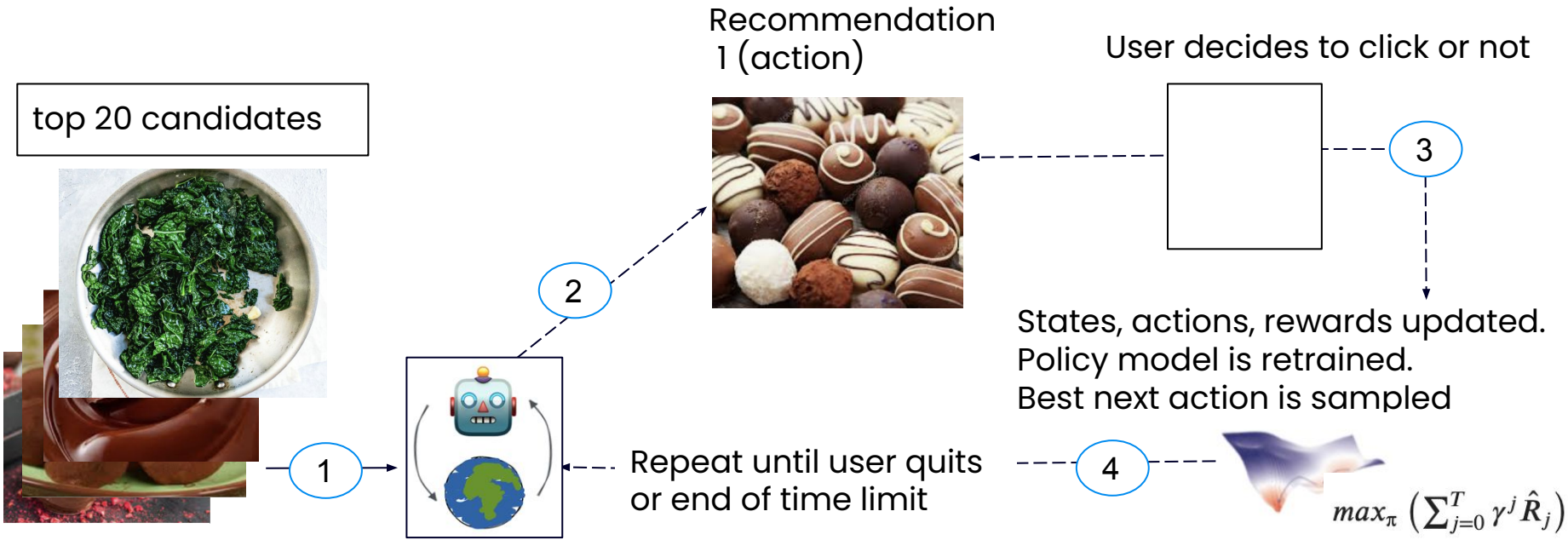
- By formulating each user's session history as a sequence of decisions, **the RecSys problem can be converted into a sequential decision-making problem**.
    - **documents** = items to be recommended
    - **States** = previous item features, user features
    - **Actions** = recommended items
    - **Rewards** = long term satisfaction
    - **Gamma** = 0 (bandits) or 1 (RL)
    - **Agent** = Recommendation system
    - **Env** = Google's RecSim
    - **Algorithm** = RLlib algorithm

RLlib

ODSC

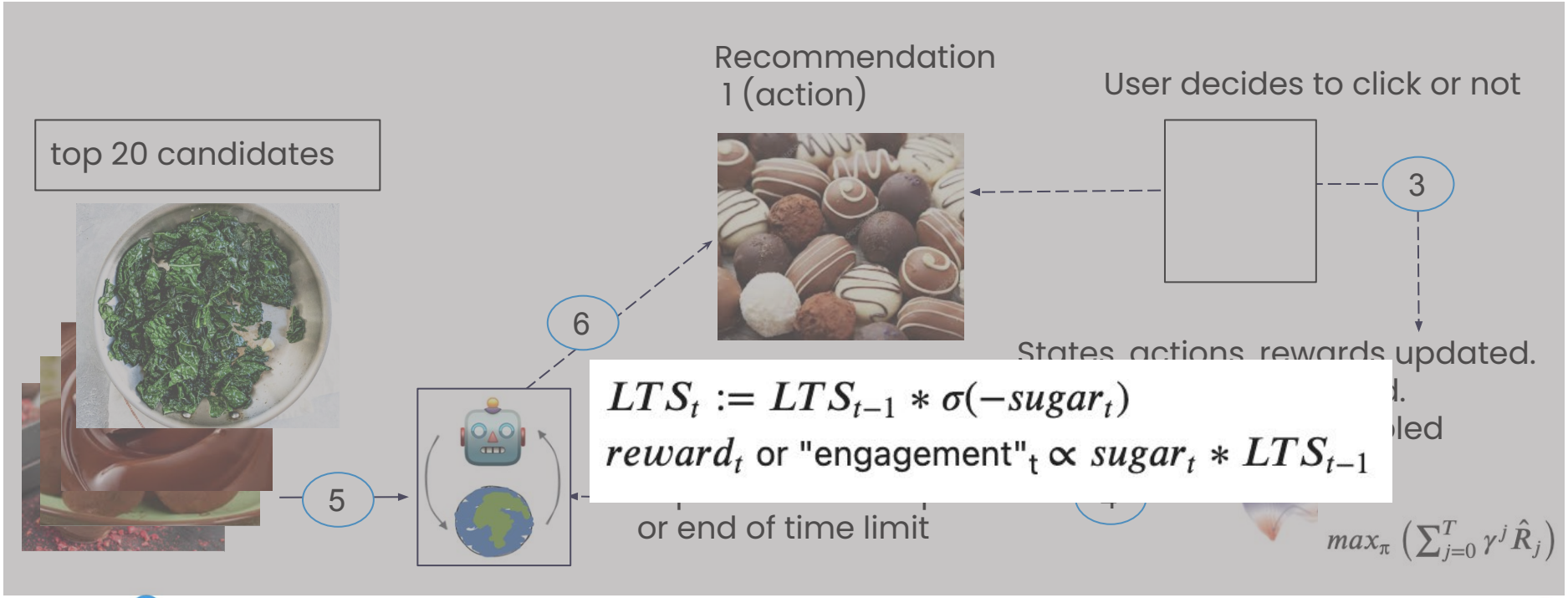
Environments Suitable for RL

Delayed Rewards & Long Term Satisfaction (LTS)

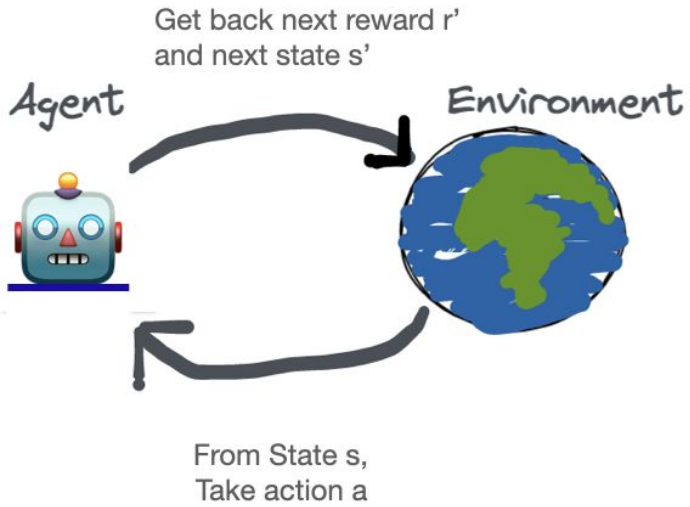
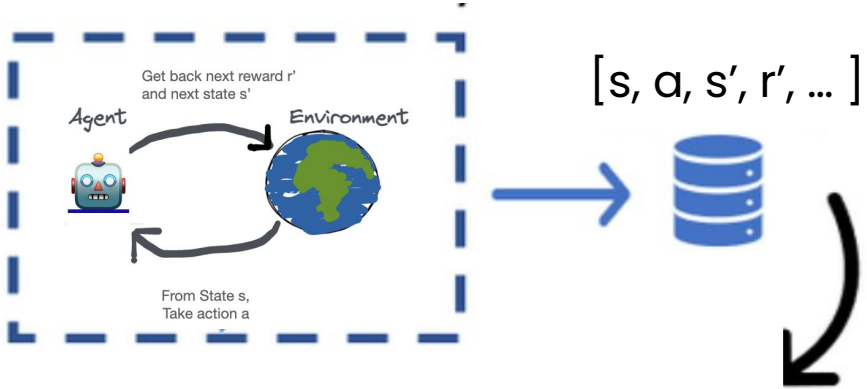


Environments Suitable for RL

Delayed Rewards & Long Term Satisfaction (LTS)



Online RL vs Offline RL

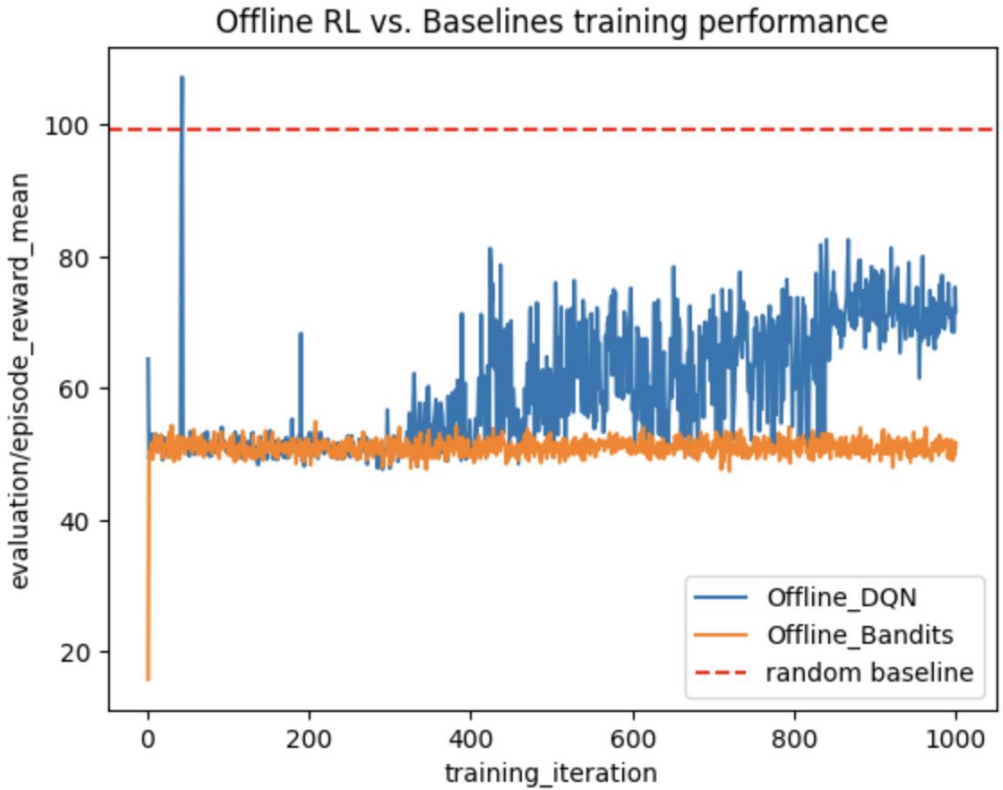
Online RL	Batch / Offline RL
	

Today's Dataset

- RecSim environment with a random agent.
- Offline Bandit
- Offline RL

https://bit.ly/rllib_odsc_west_2022

Sample result from notebook

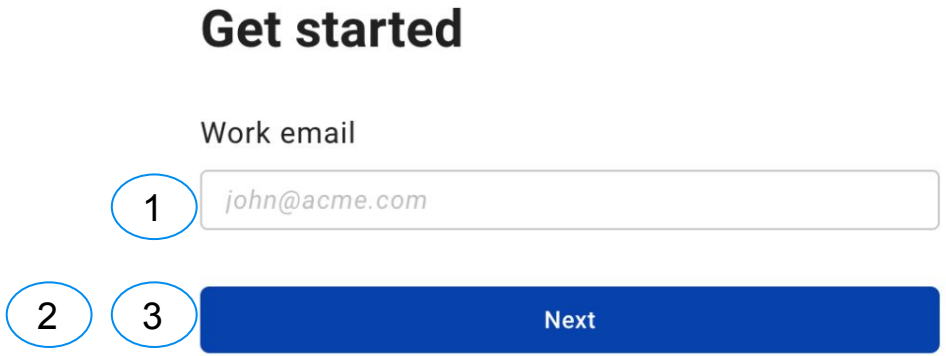


Your Anyscale Cluster

1. Claim username/password at https://bit.ly/rllib_odsc_west_2022
   a. Update the "**Status**" column to "**Claimed**"
2. Go to Console: http://console.anyscale.com/
3. Enter username (for the email) and password

anyscale

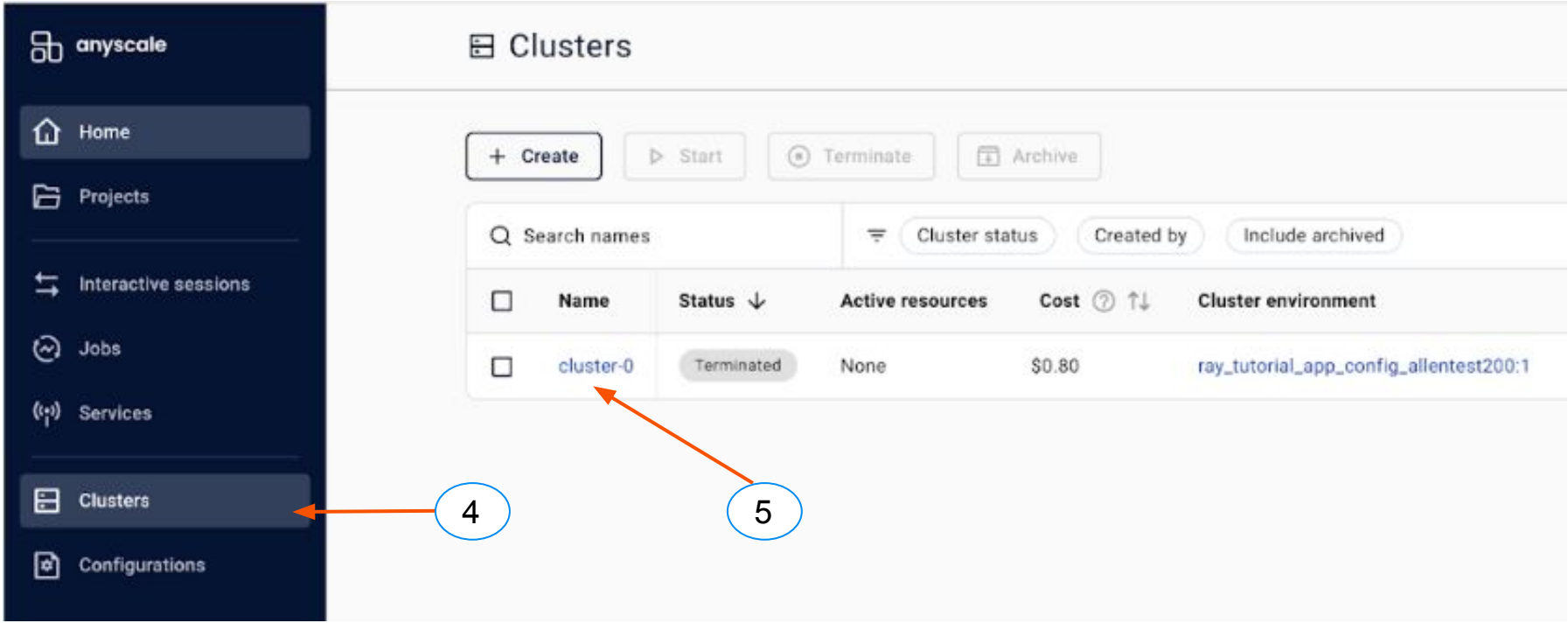
Scale your application from your laptop to the cloud



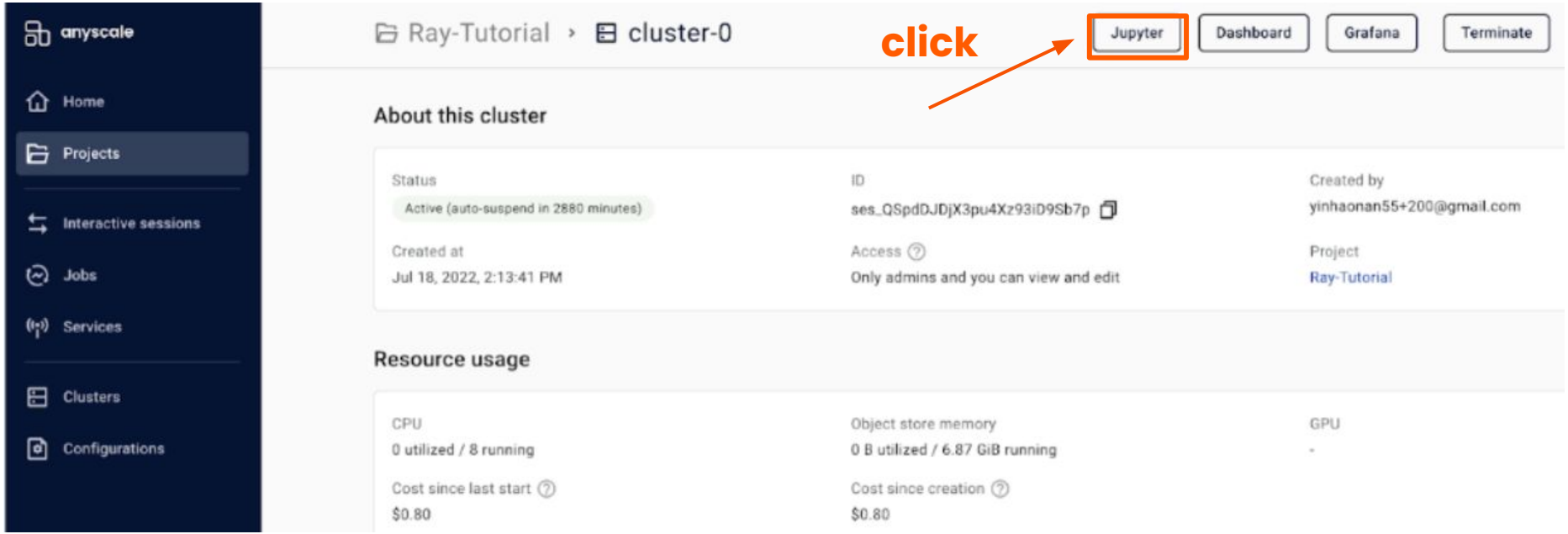
RLlib

ODSC

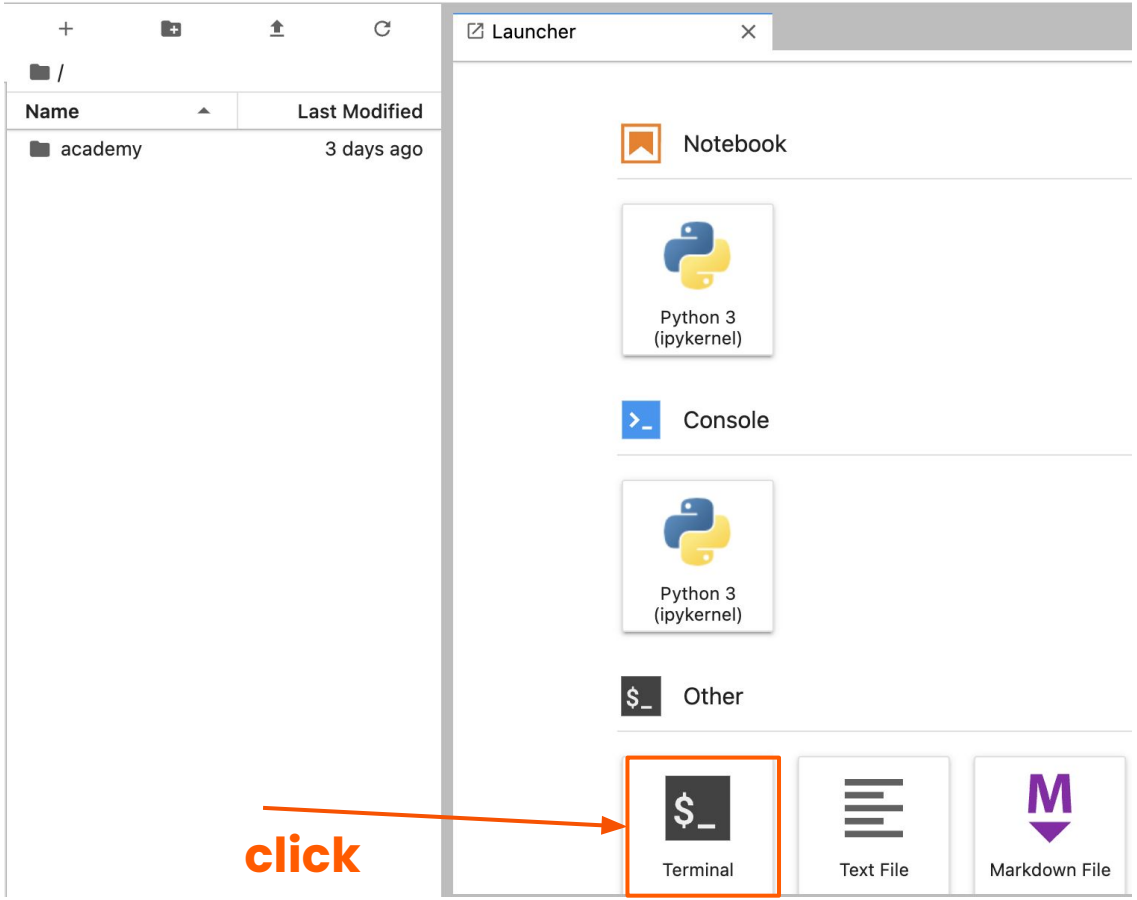
Your Anyscale Cluster

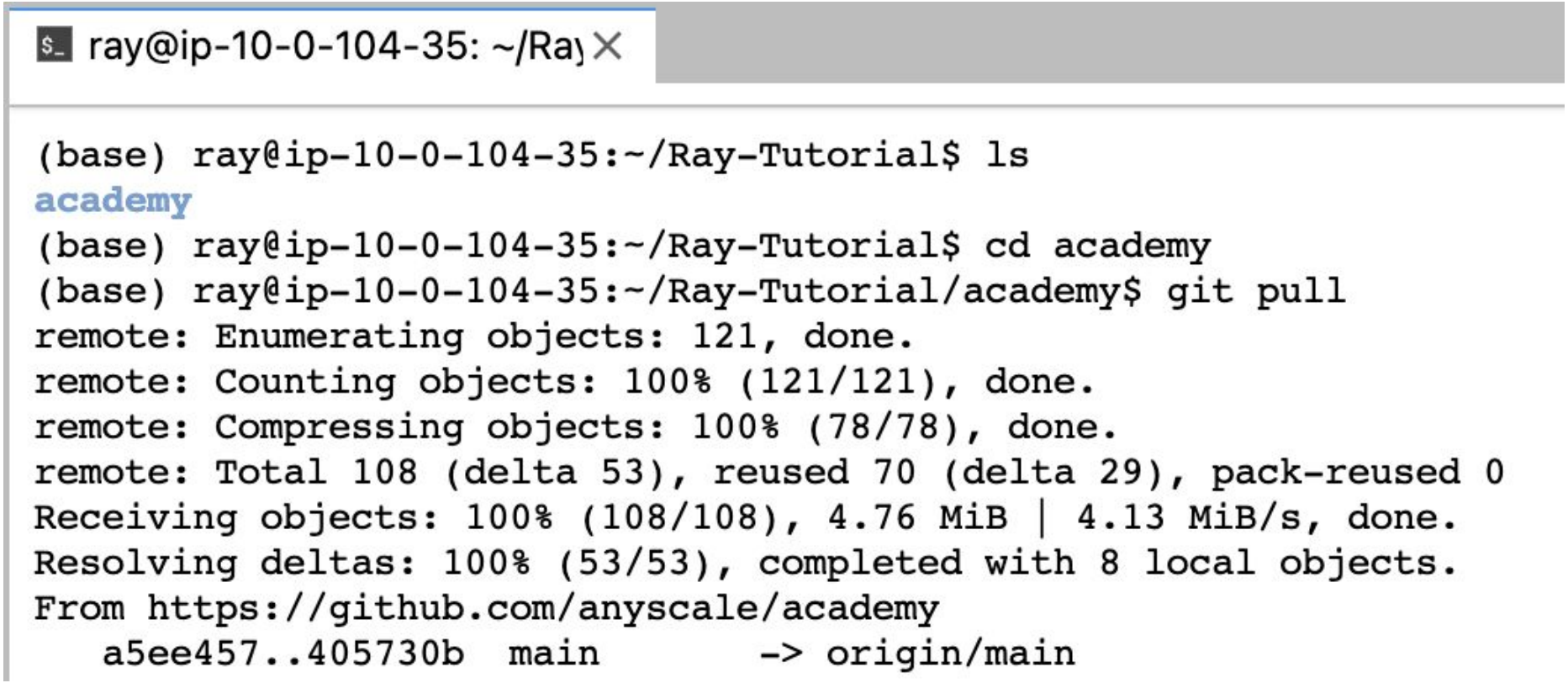


Your Anyscale Cluster

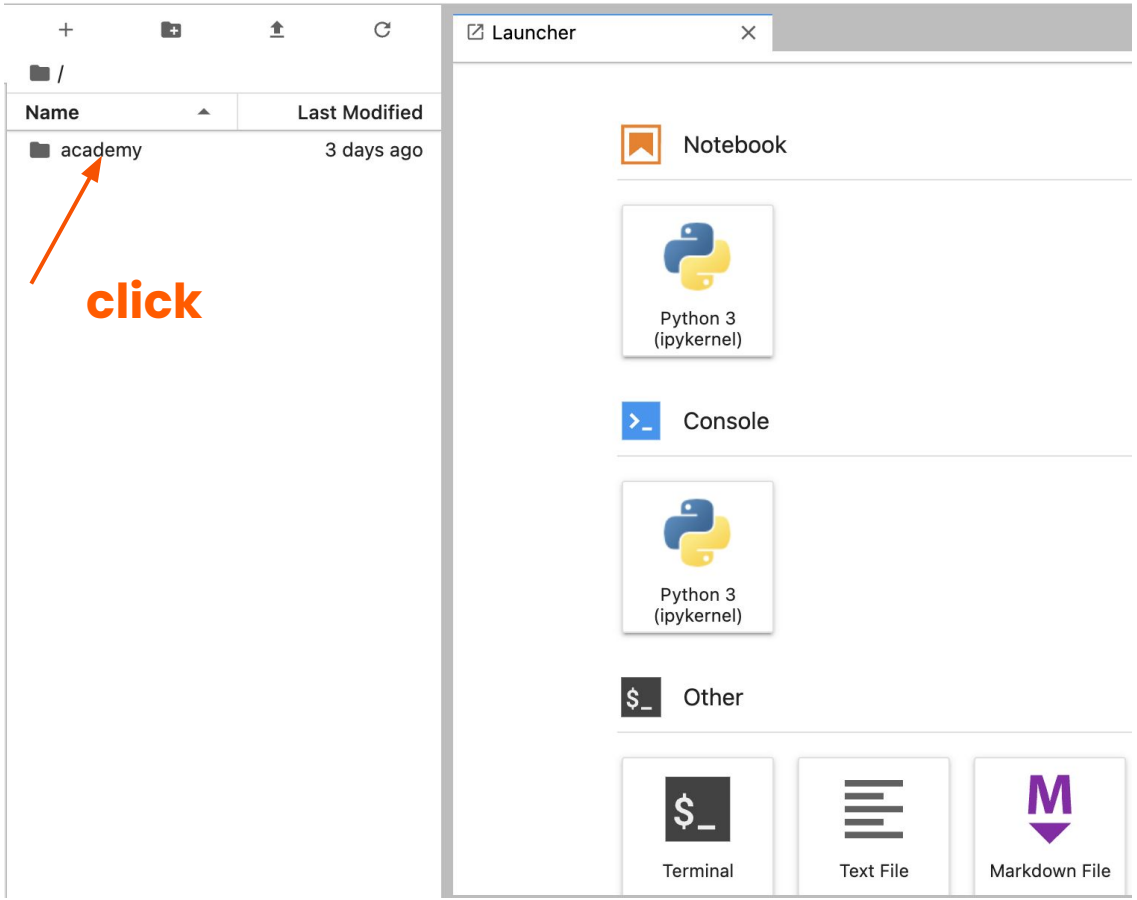


Your Anyscale Cluster

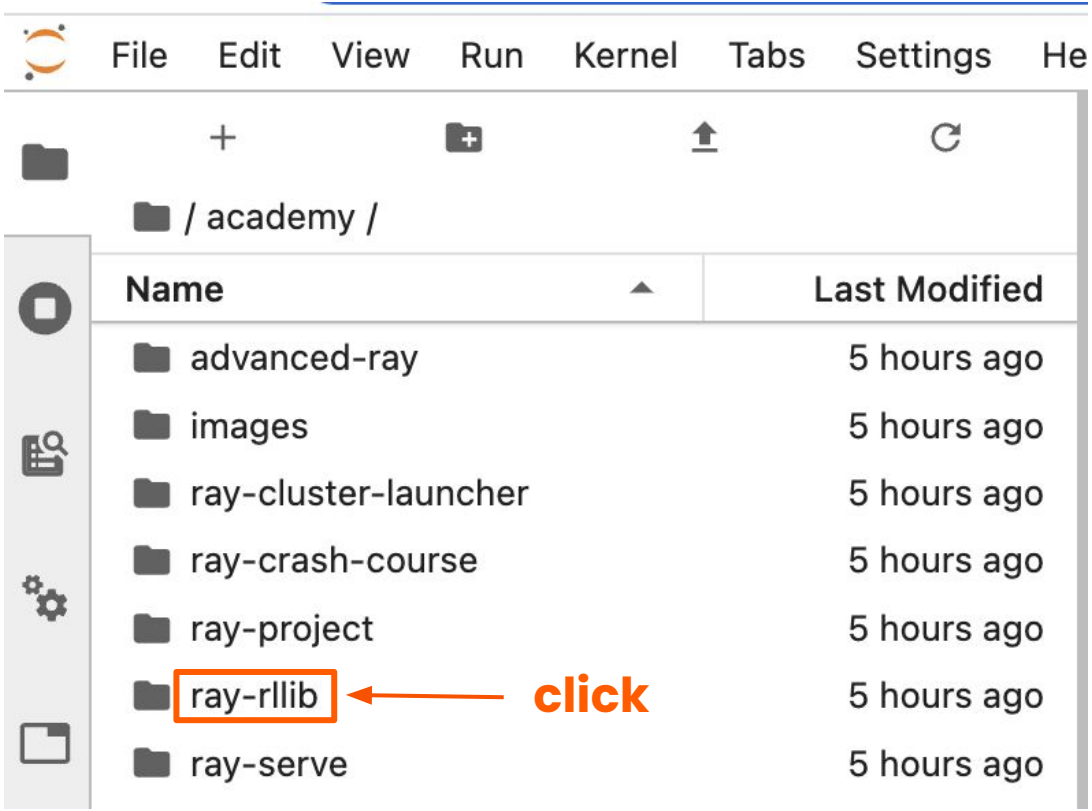



ray@ip-10-0-104-35: ~/Ra
(base) ray@ip-10-0-104-35:~/Ray-Tutorial$ ls
academy
(base) ray@ip-10-0-104-35:~/Ray-Tutorial$ cd academy
(base) ray@ip-10-0-104-35:~/Ray-Tutorial/academy$ git pull
remote: Enumerating objects: 121, done.
remote: Counting objects: 100% (121/121), done.
remote: Compressing objects: 100% (78/78), done.
remote: Total 108 (delta 53), reused 70 (delta 29), pack-reused 0
Receiving objects: 100% (108/108), 4.76 MiB | 4.13 MiB/s, done.
Resolving deltas: 100% (53/53), completed with 8 local objects.
From https://github.com/anyscale/academy
a5ee457..405730b main -> origin/main

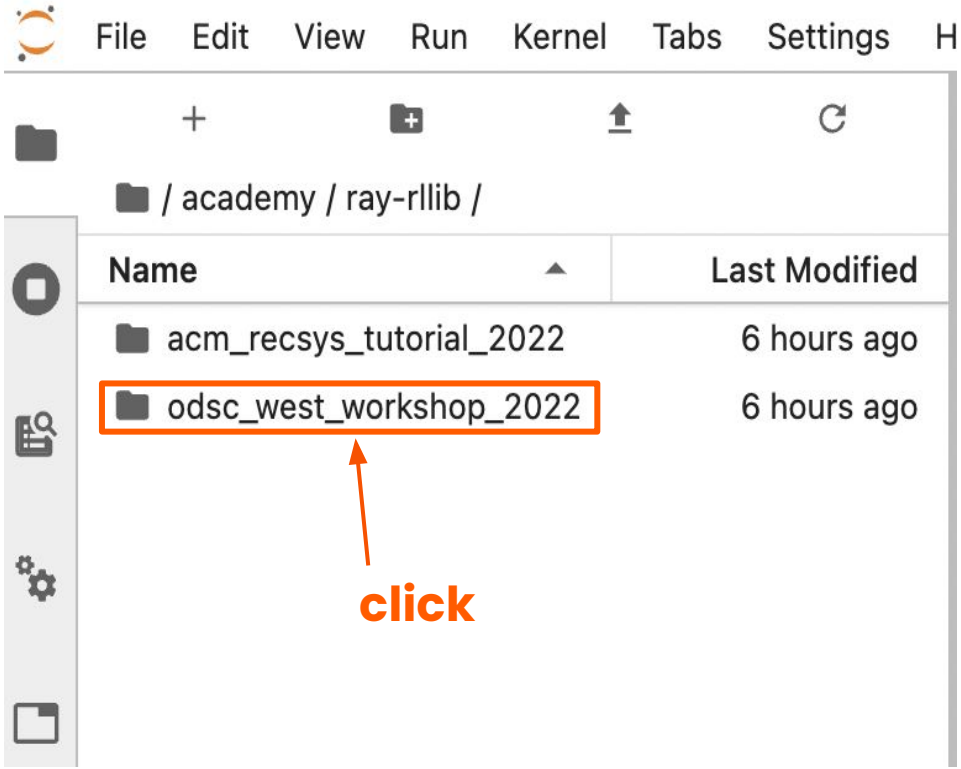
Your Anyscale Cluster



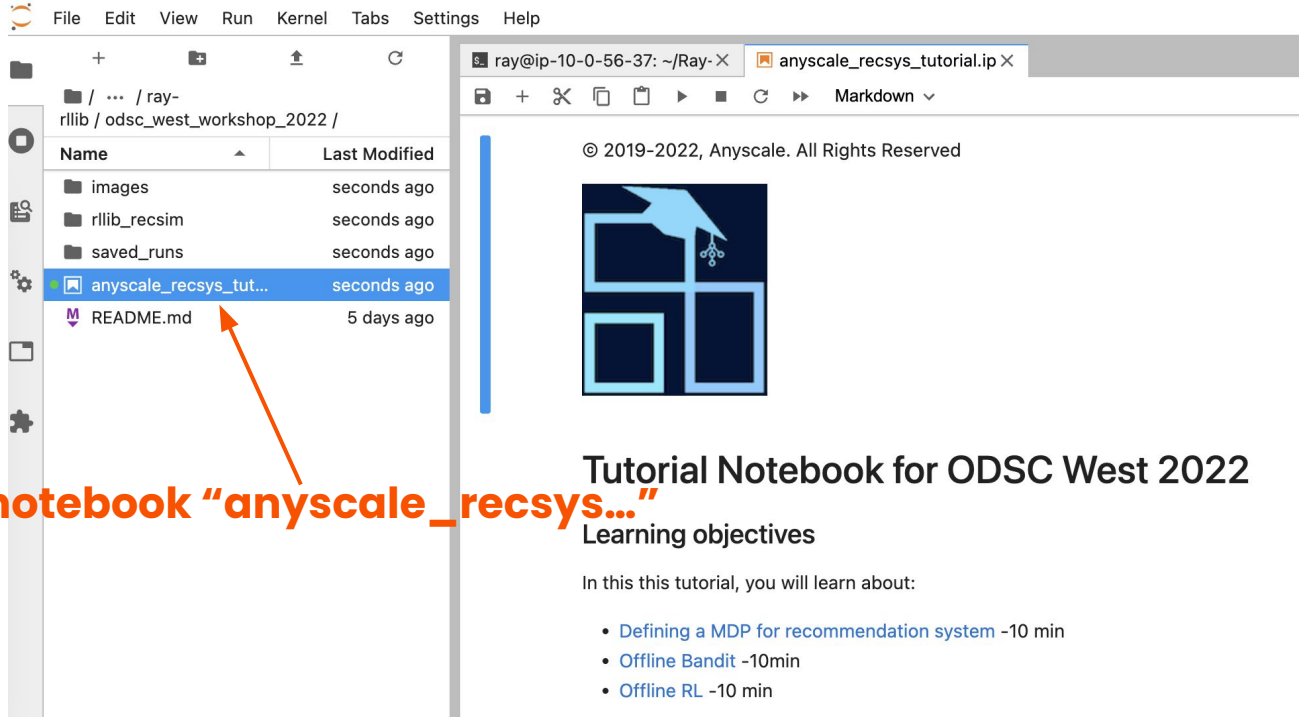
Your Anyscale Cluster



Your Anyscale Cluster



Your Anyscale Cluster



Open notebook "anyscale_recsys..."

Thank you.

We would love to connect with you!

Twitter - @anyscalecompute | @raydistributed

GitHub - https://github.com/ray-project/ray

Slack - https://www.ray.io/community

Discuss - https://discuss.ray.io/