

Deploy and Scale LLM Apps



Getting Started

Welcome to Anyscale! If you need any accommodations, let us know and we'll do our best to make you feel more comfortable in our space.

WiFi Network: Anyscale-Guest
WiFi Password: ProgramTheCloud

How to Participate

Join live polls, quizzes, and engage in Q&A by going to app.sli.do and enter code #LLM

You can also ask questions live by calling over an instructor.

How to Access Anyscale

Log-in to your cluster at console.anyscale.com

We have already made you an account. Check your email for your unique credentials for this workshop.

GitHub Repository

Everything you need will be mounted to your Anyscale cluster.

If you want to experiment locally after this event, all the materials live in github.com/ray-project/llms-in-prod-workshop-2023, and you can find bonus notebooks at github.com/ray-project/ray-educational-materials.

Free Resources

We'll be holding office hours next week. Come talk to us about extending your Anyscale cluster access to explore more with Ray!

Docs - docs.ray.io
Blog - anyscale.com/blog
YouTube - youtube.com/anyscale

Ray Summit 2023

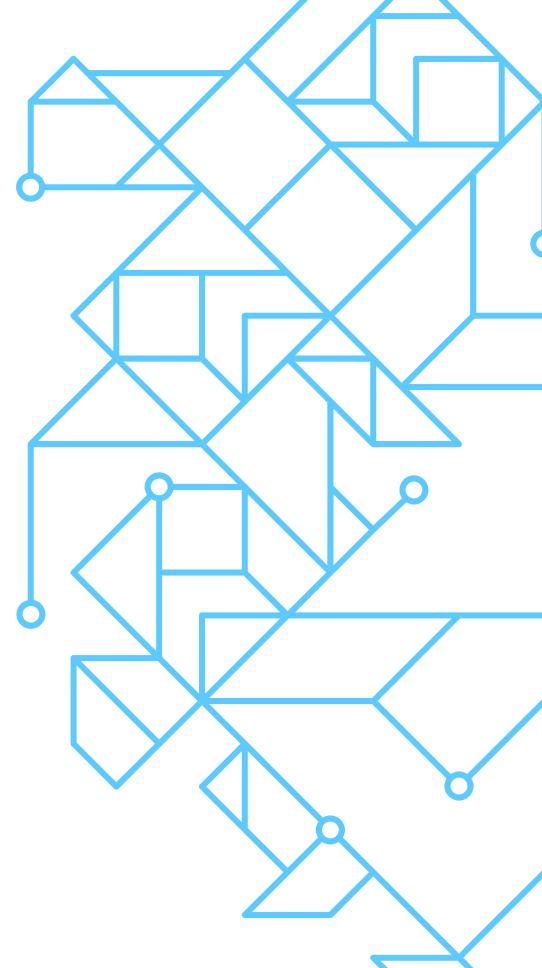
Join the Ray global community in San Francisco September 18-20 to explore the future of machine learning and scalable AI.

Register at raysummit.anyscale.com and get 15% off with code RAYMEETUP. Fill out the survey bit.ly/llms-feedback for 20% off!



Welcome!

We're happy to have you here.





Meet the team!



Emmy



Adam



Kamil



Our goals.

 Bridge the gap between dev and prod.

Introduce Ray for production-grade LLM systems.

 Learn by doing.

Hands-on, relevant coding examples.

 Reinforce through discussion.

Polls, live Q&A, conversation at tables.

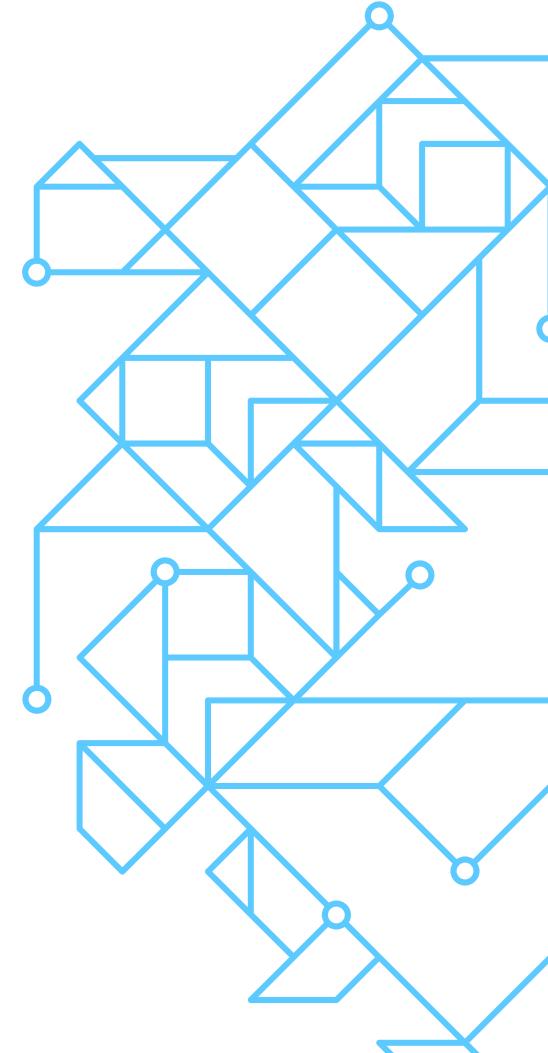
 Cultivate community.

Find friends, network, knowledge share.



The Plan

Here's what to expect today.





Today's agenda.

3:30pm (15 min)	Demo: Run an LLM App in 15 Minutes
3:45pm (20 min)	Talk: Introduction to Ray and Anyscale
4:05pm (15 min)	Coding Lab: Ray Serve for Scalable Deployments
4:20pm (10 min)	Coffee Break
4:30pm (50 min)	Coding Lab: LLMOps - Launching an LLM QA App in Production
5:20pm (10 min)	Talk: Resources for Further Exploration



Tech check.



Participating via app.sli.do

- Join with code **#LLM**
- Answer polls and quizzes.
 - Enter your name to compete!
- Ask questions.
 - Pose your own and upvote others.
 - TAs will be answering questions on a rolling basis.



Tech check.



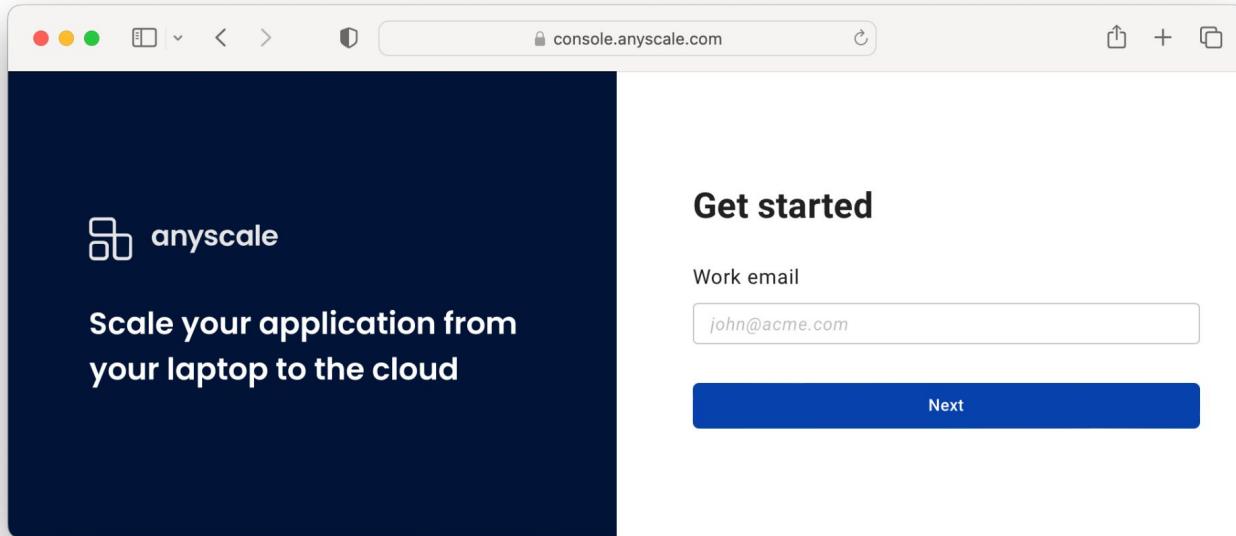
Accessing Anyscale clusters.

- All work will be in Anyscale provisioned clusters.
- Our GitHub repo will be mounted automatically.
- Access begins now.
 - Check your email for login information.
 - Step-by-step instructions to follow.



Anyscale Login

Link to Anyscale cluster: console.anyscale.com



Enter the
unique
credentials
sent to your
email!

1. Select
“Clusters”

The screenshot shows the Anyscale console interface. On the left, a dark sidebar menu lists various options: Home, Projects, Workspaces, Schedules, Jobs, Services, Clusters (which is highlighted with a yellow box and has a curved arrow pointing to it from the text), Configurations, Emmy, and Help. The main content area is titled "Clusters". It features a header with four buttons: "+ Create", "Start", "Terminate", and "Archive". Below the header is a search bar with a placeholder "Search names" and a filter button "Cluster status" set to "Created by is me". A table lists one cluster entry:

Name	Status	Active resources
Ilms-in-prod	Active (auto-terminates in 119 minutes)	0 gpu, 0 cpu

A blue box highlights the "Ilms-in-prod" cluster name. A curved arrow points from the text "2. Click on your cluster" to the highlighted cluster name. The bottom right corner of the screen has a blue circular icon with a white question mark.

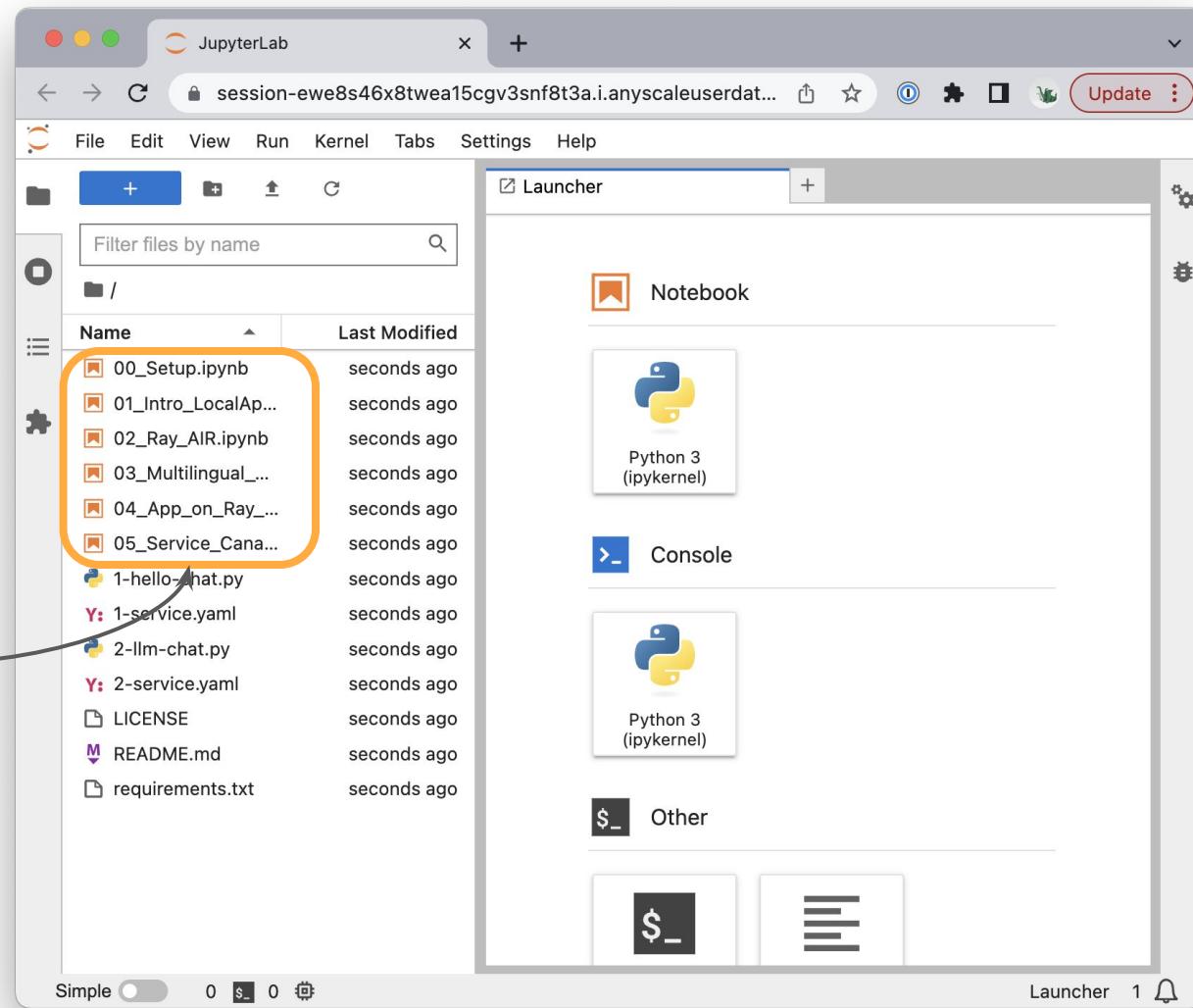
2. Click on
your cluster

3. “Start” the cluster

4. Then select “Jupyter”

The screenshot shows the Anyscale console interface. On the left, a sidebar menu lists "Home", "Projects" (selected), "Workspaces", "Schedules", "Jobs", "Services", "Clusters" (selected), "Configurations", "Emmy", and "Help". The main content area is titled "Ilms-in-prod" and displays the "About this cluster" section. This section includes fields for "Status" (Active (auto-terminates in 119 minutes)), "Created at" (Jun 12, 2023, 10:48:05 AM), "ID" (ses_ewe8s46x8twea15cgv3snf8t...), and "Access" (Everyone in your organization can view...). Below this is the "Resource usage" section, which shows CPU utilization (0 utilized / 32 running) and object store memory usage (0 GiB utilized / 35.84 GiB running). The "Configuration" section contains links for "Cluster environment" and "Compute config". At the top right of the main content area, there are three buttons: "Jupyter" (highlighted with an orange box and arrow), "Dashboard", and "Grafana". A large arrow points from the "Clusters" item in the sidebar to the "Jupyter" button.

5. View modules here



The screenshot shows a JupyterLab interface with a sidebar file browser and a main notebook editor.

File Browser:

- Shows a list of files in the current directory.
- File list:
 - 00_Setup.ipynb
 - 01_Intra_L...
 - 02_Multili...
 - 03_App_o...
 - 04_Service...
 - 1-hello-ch...
 - 1-service.y...
 - 2-llm-chat...
 - 2-service....
 - LICENSE
 - README.md
 - requireme...

Notebook Editor:

Warm-Up Notebook

You've made it to the first notebook in this workshop! As a good way to check that everything is running correctly, let's preload the model weights to save time later.

```
[ ]: import torch
      from transformers import pipeline
```

We'll be using `StableLM` throughout this workshop. To trigger the full download of the weights, set up a pipeline that caches the model with the fast NVMe storage on Anyscale.

```
[ ]: p = pipeline(model="stabilityai/stablelm-tuned-alpha-7b",
                 model_kwargs={'device_map': 'auto', 'torch_dty
```

Verify that the model is loaded into GPU memory.

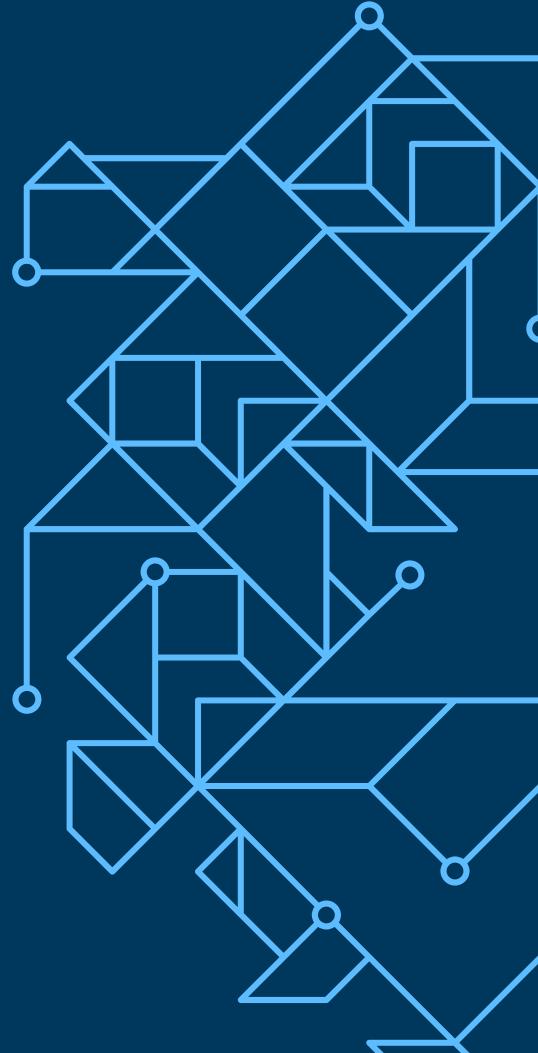
```
[ ]: ! nvidia-smi
```

In production situations, this memory should be freed when the process exits. However, in a notebook (or other long-running dev process environment), it can be useful to purge unneeded data

6. Run “00_Setup.ipynb”

LLM App in 15 Min

Run a question answering
application locally.



The screenshot shows a JupyterLab interface with a central notebook area and a sidebar.

File Explorer: On the left, a sidebar displays a file tree and a list of files. The list includes:

Name	Last Modified
00_Setup.ipynb	4 minutes ago
01_Intro_L...	7 minutes ago
02_Multili...	7 minutes ago
03_App_o...	7 minutes ago
04_Servic...	7 minutes ago
1-hello-ch...	7 minutes ago
Y: 1-service.y...	7 minutes ago
Y: 2-llm-chat...	7 minutes ago
Y: 2-service....	7 minutes ago
LICENSE	7 minutes ago
README.md	7 minutes ago
requireme...	7 minutes ago

Notebook Area: The main area displays the content of the selected notebook, **01_Intro_LocalApp.ipynb**.

Run an LLM App in 15 Minutes

To prime ourselves for the type of work ahead, we will start by creating a [question answering \(QA\)](#) service designed to run locally.

Large language models (LLMs), while very impressive at next token prediction, have no relationship to the truth. This is especially relevant when the topic falls outside of the model's training data. To help mitigate their hallucinatory tendencies, we can implement a pattern referred to as [retrieval QA](#). In this use case, we generate embeddings for domain-specific documents that the LLM can then use to construct a response to a user query.

After this short notebook, you will have set up a [document corpus](#) of [Taylor Swift's Eras Tour](#) and the [2023 XFL Season](#) for StableLM to use as context to supplement its generated answer.

Create a document corpus

First, you need to establish the pool of information from which the language model will draw its context. In this example, we'll be using a few modules from [LangChain](#) to facilitate this process. We'll be

01_Intro_LocalApp.ipynb

slido



Audience Q&A Session

- ⓘ Click **Present with Slido** or install our [Chrome extension](#) to show live Q&A while presenting.

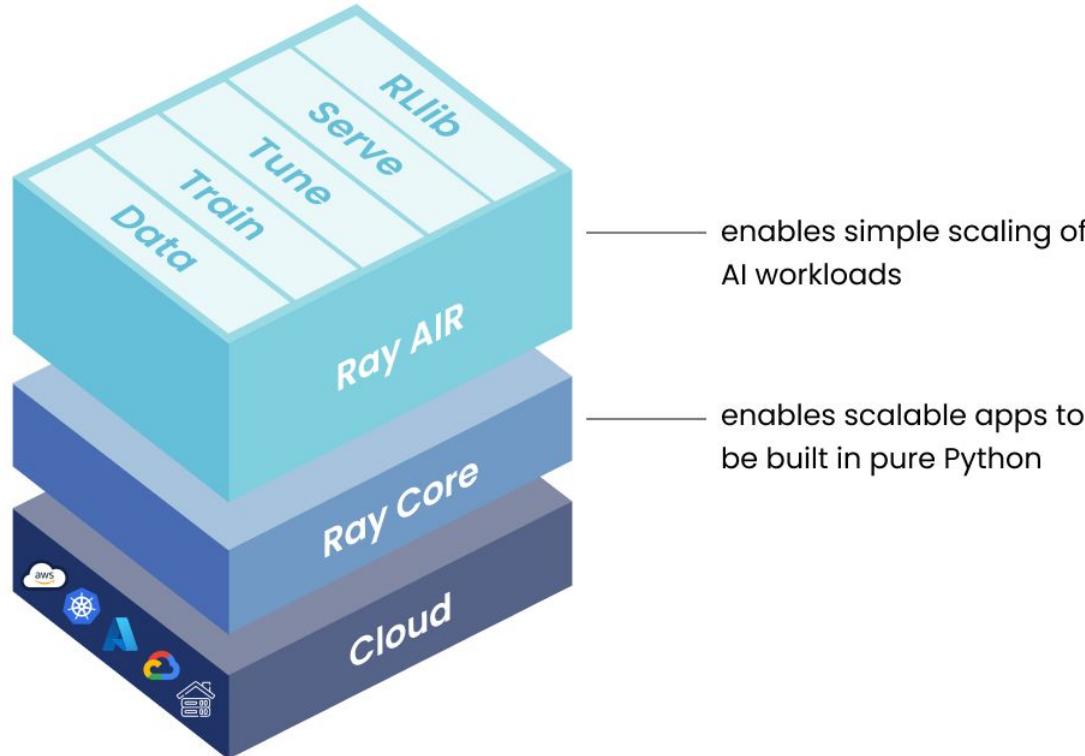
From local to cloud

An introduction to Ray and
Anyscale.





Where Ray AIR sits.



↔ End-to-end ML scaling.



slido



Audience Q&A Session

- ⓘ Click **Present with Slido** or install our [Chrome extension](#) to show live Q&A while presenting.

Scale Out

An introduction to Ray Serve
for scalable deployments.



02_Multiling... - JupyterLab

session-ewe8s46x8twea15cgv3snf8t3a.i.anyscaleuserdata.com/ju...

File Edit View Run Kernel Tabs Settings Help

02_Multilingual_Chat_with_Ray_Serve.ipynb

Python 3 (ipykernel)

Multilingual Chat with Ray Serve

```
[ ]: import ray
import requests, json
from starlette.requests import Request
from typing import Dict

from ray import serve
```

Ray Serve is a microservices framework for serving ML – the model serving component of Ray

Ray Serve provides resource management, scaling, a straightforward component framework, FastAPI compatibility ... and direct integration to the entire Ray ecosystem for scale-out compute.

```
[ ]: ray.init()
```

Chatbot using Huggingface LLM

Simple 0 Python 3 (ipykernel)... Mode: Com... Ln 1, C... 02_Multilingual_Chat_with_Ray_Serve_G... 1

02_Multilingual_Chat_with_Ray_Serve_GPU.ipynb

slido



Audience Q&A Session

- ⓘ Click **Present with Slido** or install our [Chrome extension](#) to show live Q&A while presenting.

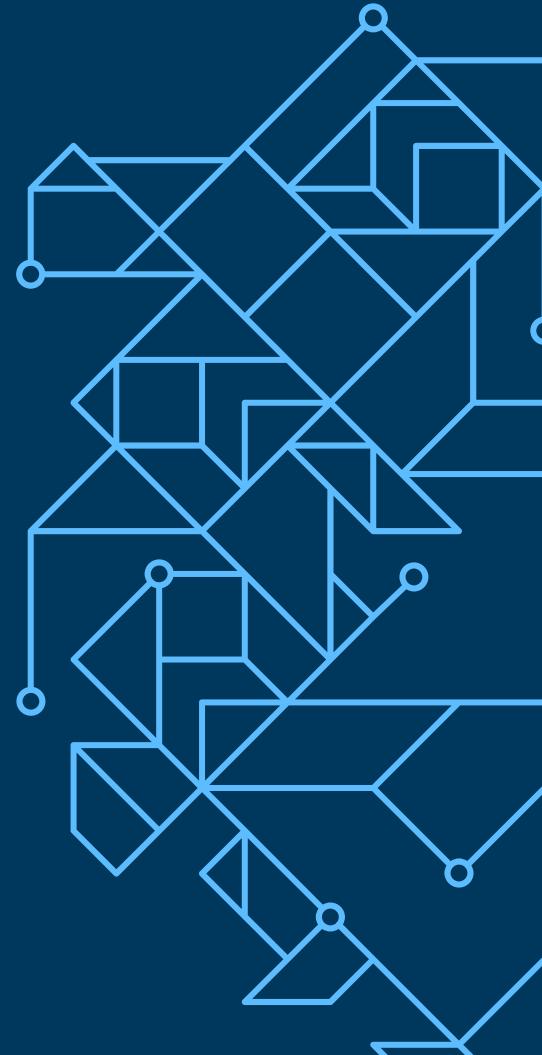


**Time for a
Break!**

10 minutes.

LLMOps

Launching an LLM QA
application in production



The screenshot shows a JupyterLab interface with a sidebar containing a file tree and a main notebook area. The notebook title is '03_App_on_Ray_Serve.ipynb'. The code cell contains the following imports:

```
[ ]: from typing import Optional, Any, Dict
from operator import add
import requests, json
from starlette.requests import Request
import numpy as np
import torch

from sentence_transformers import SentenceTransformer
from langchain.embeddings.base import Embeddings
from langchain.text_splitter import RecursiveCharacterTextSplitter
from langchain.vectorstores import FAISS
from langchain.document_loaders import WikipediaLoader
from langchain import HuggingFacePipeline
from langchain.chains.question_answering import load_qa_chain
from langchain.prompts import PromptTemplate

from transformers import pipeline as hf_pipeline

import ray
from ray import serve
```

Productionizing LLM Q&A Application with Ray Serve

In this notebook, we'll see how to productionize our Q&A application and its database service.

Simple 0 \$ 4 Python 3 (ipykernel) | I... Mode: Comma... 1 Ln 20, Col 22 03_App_on_Ray_Serve.ipynb 1 🔔

03_App_on_Ray_Serve.ipynb

The screenshot shows a JupyterLab interface with a file browser on the left and a notebook cell on the right.

File Browser:

- Path: /
- Items:

 - 00_Setup.ipynb (4 minutes ago)
 - 01_Intro_L... (7 minutes ago)
 - 02_Multili... (7 minutes ago)
 - 03_App_o... (7 minutes ago)
 - 04_Service_Canary_Rollout.ipynb (7 minutes ago) - Selected
 - 1-hello-chat.py (7 minutes ago)
 - Y: 1-service.yaml (7 minutes ago)
 - 2-llm-chat... (7 minutes ago)
 - Y: 2-service.... (7 minutes ago)
 - LICENSE (7 minutes ago)
 - README.md (7 minutes ago)
 - requireme... (7 minutes ago)

Notebook Cell (04_Service_Canary_Rollout.ipynb):

Anyscale Services + Canary Rollout Features

Anyscale Services is the part of the Anyscale platform which provides web endpoints to Ray Serve applications. Anyscale Services provides key production features including

- High availability (HA)
- Canary rollouts for new service versions
- Extensive monitoring/management
- Support for the entire Ray platform, FastAPI, and applications which go beyond Ray

Setup

The service versions are implemented in Python using standard Ray Serve APIs

- `1-hello-chat.py` - skeleton for a chat service, it generates a response in a trivial static manner
- `2-llm-chat.py` - our real LLM chat service

Each service version has a corresponding YAML file used to deploy

04_Service_Canary_Rollout.ipynb

slido

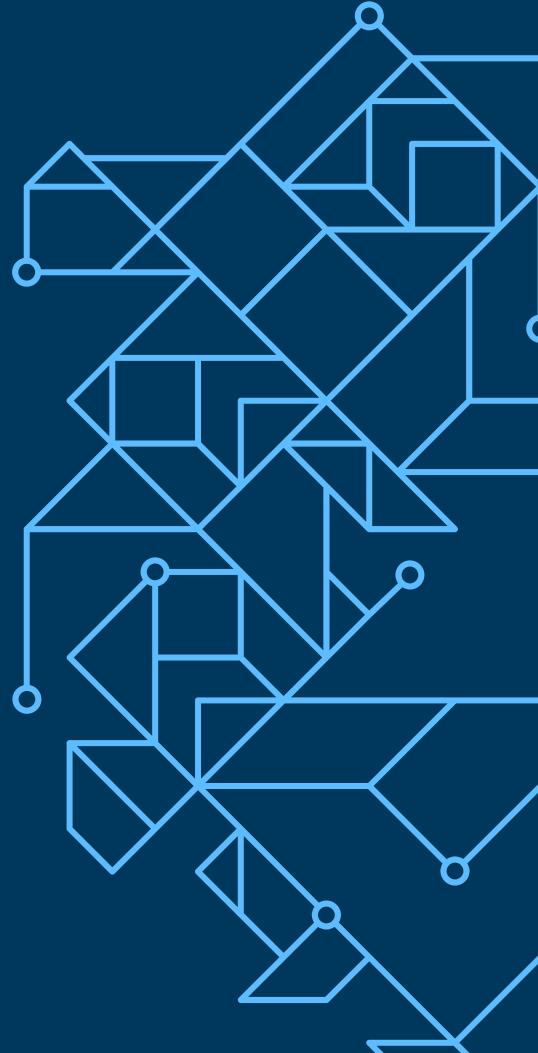


Audience Q&A Session

- ⓘ Click **Present with Slido** or install our [Chrome extension](#) to show live Q&A while presenting.

More Resources

For further exploration with
Ray, Anyscale, and LLMs.





Today we learned...



Introduction to Ray and Anyscale

Why distributed compute is necessary for AI.



Ray Serve for scalable deployments

Building robust application services that scale.



LLM Ops

Launching an LLM QA application in production.



Reading list.



[Ray Education GitHub](#)

Access bonus notebooks and scripts about Ray.



[Ray documentation](#)

API references and user guides.



[Anyscale Blogs](#)

Real world use cases and announcements.



[YouTube Tutorials](#)

Video walkthroughs about learning LLMs with Ray.



Upcoming events



Office Hours

June 21 at 4:00p.m. - follow-up for today's participants.



Ray Meetup

June 21 at 5:30p.m. ft. Ray data streaming & Pinterest ML

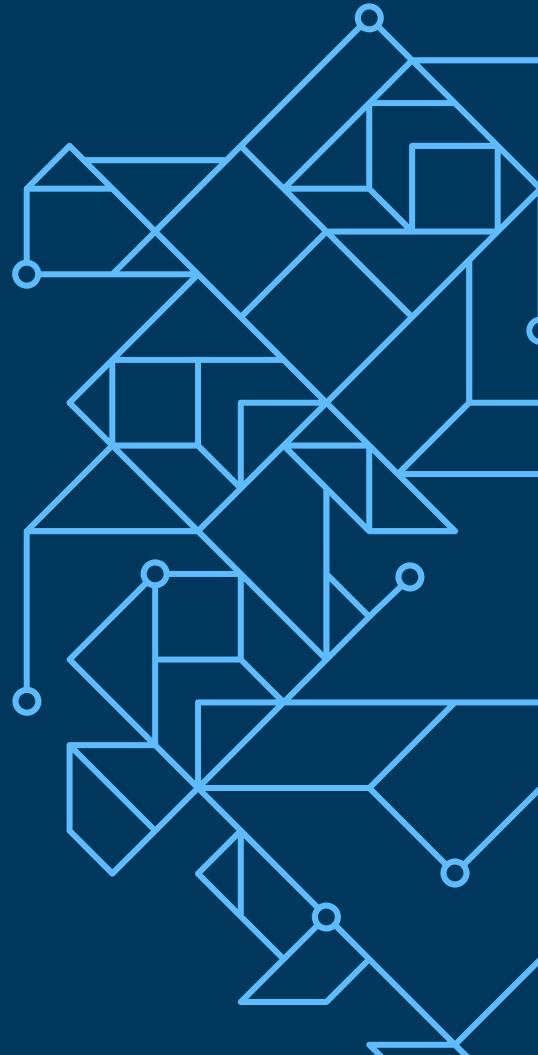


LangChain x Ray Meetup

June 22 at 5:30 ft. Robert Nishihara, Harrison Chase, Michel Tricot, Lianmin Zheng, & Charles Frye

Ray Aviary

Serving open source LLMs in production.



 Aviary Explorer Update

aviary-staging.anyscale.com

Aviary Explorer: A place to study stochastic parrots

Hosted on  anyscale | Powered by  RAY [Deploy your LLMs](#)

Compare Leaderboard Models About

LLM #1: LLM #2: LLM #3:

amazon/LightGPT lmsys/vicuna-13b-de mosaicml/mpt-7b-in

Select LLMs for me:    

Prompt

Write a description for a YouTube thumbnail that announces Ray Aviary.

Examples (Question Answering)

How do I make fried rice? What are the 5 best sci fi books?

What are the best places in the world to visit? Which Olympics were held in Australia?

Examples (Instruction Following)

Please describe a beautiful house. Generate 5 second grade level math problems.

Write a poem about shoes.

LLM #1

 Best answer is #1

The interior of the aviary is filled with birds of all sizes and shapes, some flying about the room, others perching on branches or sitting in cages. The walls are covered with bird paintings, each one different from the last.

Lat [s]	3.7
Cost [\$]	0.0010
Tokens (i/o)	88.0
Per 1K Tok [\$]	0.0119

LLM #2

 Best answer is #2

The thumbnail for Ray Aviary's YouTube channel features a stunning image of a majestic eagle in flight, its wings spread wide and its sharp talons outstretched. The background is a vibrant array of colors, with shades of blue, green, and purple blending together in a swirling pattern that gives the impression of movement and energy. The eagle is positioned in the center of the image, with the channel's logo – a stylized letter "R" in bold, modern font – superimposed over the top of the bird. The overall effect is striking and eye-catching, conveying the sense of freedom, power, and inspiration that defines Ray Aviary's brand.

Lat [s]	6.5
Cost [\$]	0.0018
Tokens (i/o)	222.0
Per 1K Tok [\$]	0.0081

LLM #3

 Best answer is #3

A black and white photo of two birds sitting on top of each other.

Lat [s]	1.4
Cost [\$]	0.0004

[Terms of Use](#) • [Privacy Policy](#)

aviary.anyscale.com

RAY SUMMIT 23



THE PLACE FOR EVERYTHING RAY

SEPT. 18–19 + SEPT. 20 TRAINING DAY

SAN FRANCISCO, CA

presented by  **anyscale**

Keynote Speakers for Ray Summit 2023



Albert Greenberg
VP Engineering
Uber



Brian McClendon
SVP Engineering
Niantic



Robert Nishihara
CEO
Anyscale



Ya Xu
Head of Data & AI
VP Engineering
LinkedIn



Ion Stoica
Co-Founder & President
Anyscale
Professor, U.C. Berkeley



Aidan Gomez
Co-founder and CEO
Cohere



John Schulman
Co-founder
OpenAI

raysummit.anyscale.com



Connect with the community.



Join the community

[Attend events](#), [subscribe to newsletter](#), [follow on Twitter](#).



Get support

[Join Ray Slack](#), [ask questions on forum](#), [open an issue](#).



Contribute to Ray

[Read contributor guide](#), [create a pull request](#).



Fill out the survey.

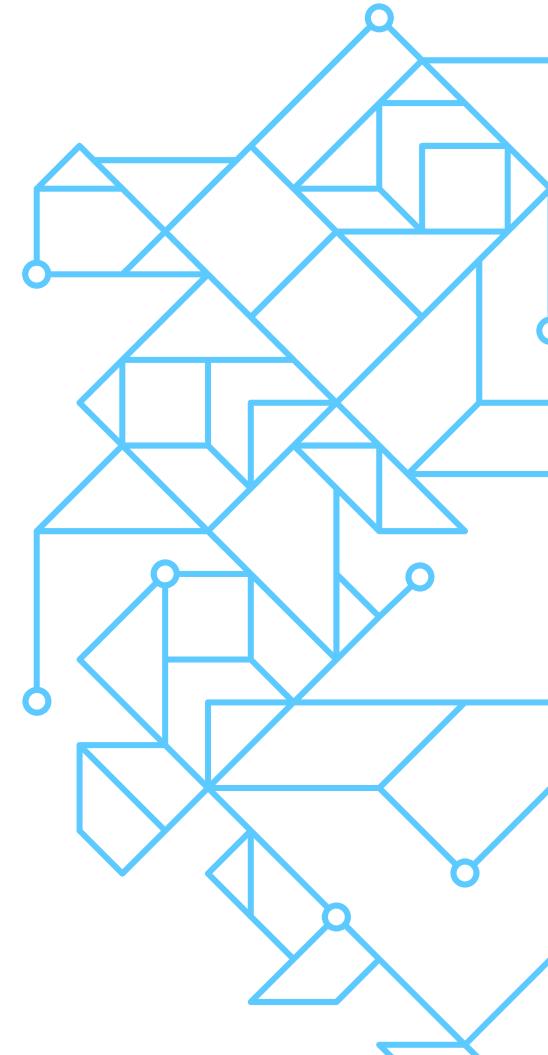


Go to bit.ly/lrms-feedback

We'll send all survey submitters a 20% discount code for Ray Summit tickets.

Thank you!

We hope to meet again.



slido



Audience Q&A Session

- ⓘ Click **Present with Slido** or install our [Chrome extension](#) to show live Q&A while presenting.