Update OpenEnv extra 24 hours deadline

Early OpenEnv Deadline + 24 hours to Monday, October 26th @ 7 PM PT

WELCOME BACK!

Schedule

Submission Deadlines:

- **Early OpenEnv Deadline: Monday, October 26th @ 7 PM PT** **See Slide 4**
- **Final Deadline: Wednesday, 29th October @ 7 PM PT**

Friday 24th October

- 7 PM PT Kickoff

Saturday 25th October:

- 10 AM PT: Opening talk Discord Event Link
- 10:30 AM PT: Daniel - **Reinforcement Learning** with OpenEnv & Unsloth. Discord Event Link
- 4 PM PT: Office hours. Discord Event Link

Sunday 26th October:

- 10 AM PT: OpenEnv and Synthetic Data Best Practises: Sanyam. Discord Event Link
- 4 PM PT: Office hours. Discord Event Link

Prizes

Synthetic-Data Track

$3000 + 1200 hrs of GPU credits
+ trophy

$1500 + 600 hrs of GPU credits
+ trophy

$900 + 300 hrs of GPU credits
+ trophy

New **OpenEnv RL** Track!

2 EARLY Submissions (7PM Monday PT) will win EXTRA!

2 x Each ($500 GPU Credits + 1 Ray-Ban Meta)

PLUS ALSO THE REGULAR TIMELINE:

3 extra Ray-Bans Meta (still deciding how to split)

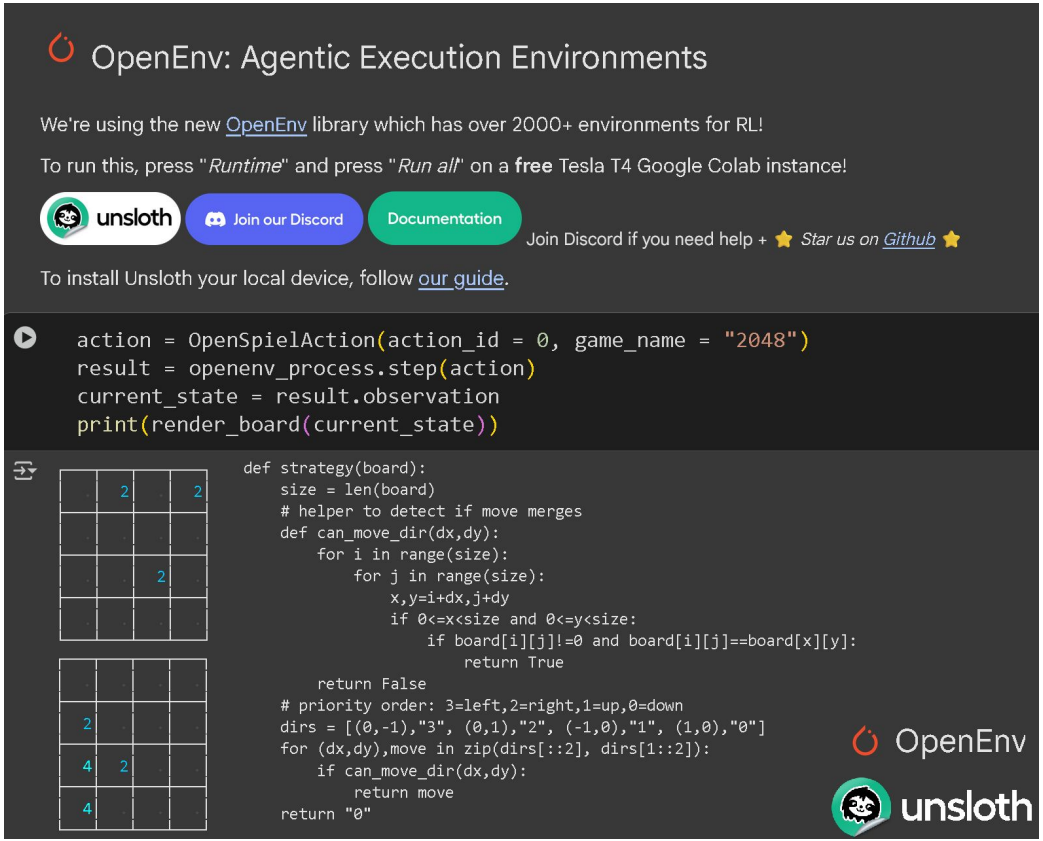
Early submissions qualify for the Regular Prizel!

$2500 GPU credits

$1000 GPU credits

$500 GPU credits

OpenEnv Reinforcement Learning



[OpenEnv](#) is a new PyTorch package offering 2000+ environments designed for reinforcement learning.

This special track will give **2 early submissions (Sunday 7PM PT) $500 + 1 Ray Ban each!**

Ideas:

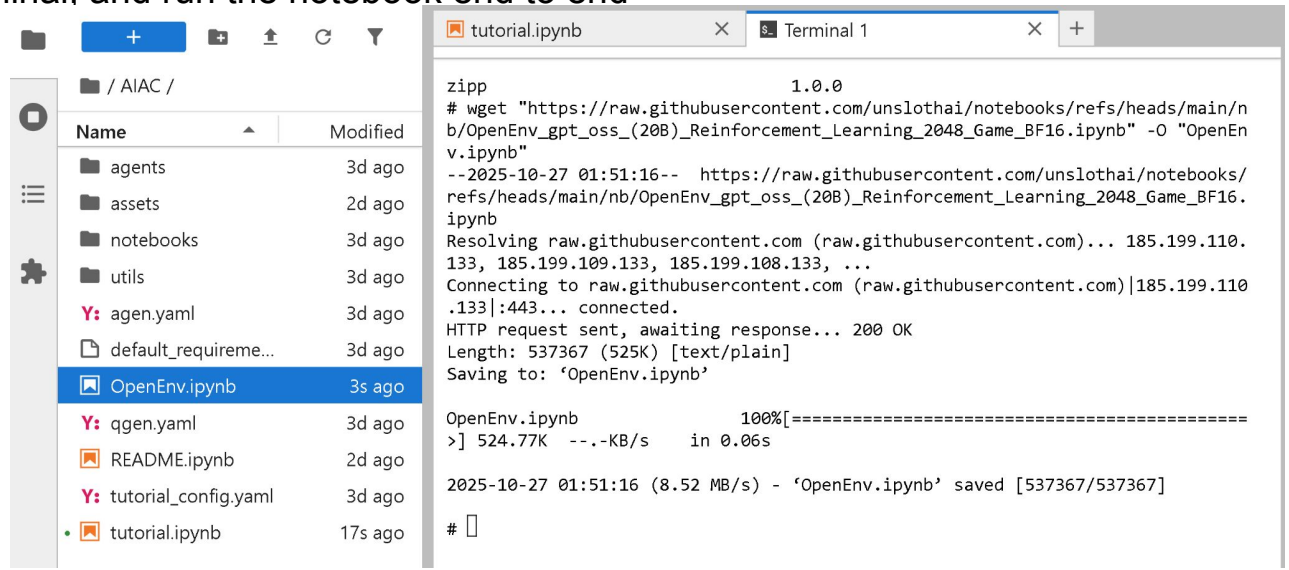
1. Customize [Unsloth + OpenEnv 2048 Game notebook](#) to work on other RL environments
2. Create universal reward functions for all 2000+ envs.
3. Design new environments using the OpenEnv spec.

OpenEnv Setup

!!! FIRST FOLLOW THE STEPS FROM SLIDE 8 ONWARDS

Then, please execute in a terminal, and run the notebook end to end

```
wget
"https://raw.githubu
sercontent.com/unslo
thai/notebooks/refs/
heads/main/nb/OpenEn
v_gpt_oss_(20B)_Rein
forcement_Learning_2
048_Game_BF16.ipynb"
-O "OpenEnv.ipynb"
```



https://github.com/unslothai/notebooks/blob/main/nb/OpenEnv_gpt_oss_(20B)_Reinforcement_Learning_2048_Game_BF16.ipynb

Weather environment / other environments RL FAQ

If you are making a custom environment, I would:

1. Ask an Open Source LLM to generate a weather environment - ie it should say today's weather (sunny, not sunny), with temperature, humidity, wind speed etc etc. It should be governed by some physical laws or mathematical transition process (Markov Chains)
2. Create a reward function to predict the weather and do RL on this!

Get faster inference with vLLM

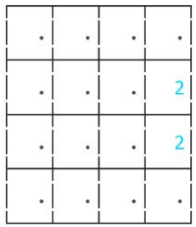
Unsloth supports faster inference for some models with fast_inference = True. For eg Llama, Qwen models. GPT-OSS will use Unsloth internal inference which is a little bit slower. See https://github.com/unslothai/notebooks/blob/main/nb/Advanced_Llama3_2_(3B)_GRPO_LoRA.ipynb Do:

```
import os
os.environ["UNSLOTH_VLLM_STANDBY"] = "1" # [NEW] Extra 30% context lengths!
from unsloth import FastLanguageModel
max_seq_length = 2048 # Can increase for longer reasoning traces
lora_rank = 64 # Larger rank = smarter, but slower
model, tokenizer = FastLanguageModel.from_pretrained(
    model_name = "meta-llama/Llama-3.2-3B-Instruct",
    max_seq_length = max_seq_length,
    load_in_4bit = False, # False for LoRA 16bit
    fast_inference = True, # Enable vLLM fast inference
    max_lora_rank = lora_rank,
)
```

ReadTimeoutError for OpenEnv

If you see the error below:

```
        else:  # right
            newb = [list(reversed(slide(list(reversed(r))))) for r in board]
        if newb != board:
            return str(m)
    return "0"
```




```
[2025-10-27 02:56:24] WARNING connectionpool.py:868: Retrying (Retry(total=2, connect=None, read=None, redirect=None, status=None)) after connection brok
n by 'ReadTimeoutError("HTTPConnectionPool(host='localhost', port=8081): Read timed out. (read timeout=15.0)")': /step
[2025-10-27 02:56:39] WARNING connectionpool.py:868: Retrying (Retry(total=1, connect=None, read=None, redirect=None, status=None)) after connection brok
n by 'ReadTimeoutError("HTTPConnectionPool(host='localhost', port=8081): Read timed out. (read timeout=15.0)")': /step
[2025-10-27 02:56:55] WARNING connectionpool.py:868: Retrying (Retry(total=0, connect=None, read=None, redirect=None, status=None)) after connection brok
n by 'ReadTimeoutError("HTTPConnectionPool(host='localhost', port=8081): Read timed out. (read timeout=15.0)")': /step
```

I added a fix since 1AM 27th October PT time. Please do:

1. **Update Unsloth** `pip install --upgrade --force-reinstall --no-deps --no-cache-dir unsloth unsloth_zoo`
2. **Re-get notebook as “OpenEnv_NEW.ipynb”:**

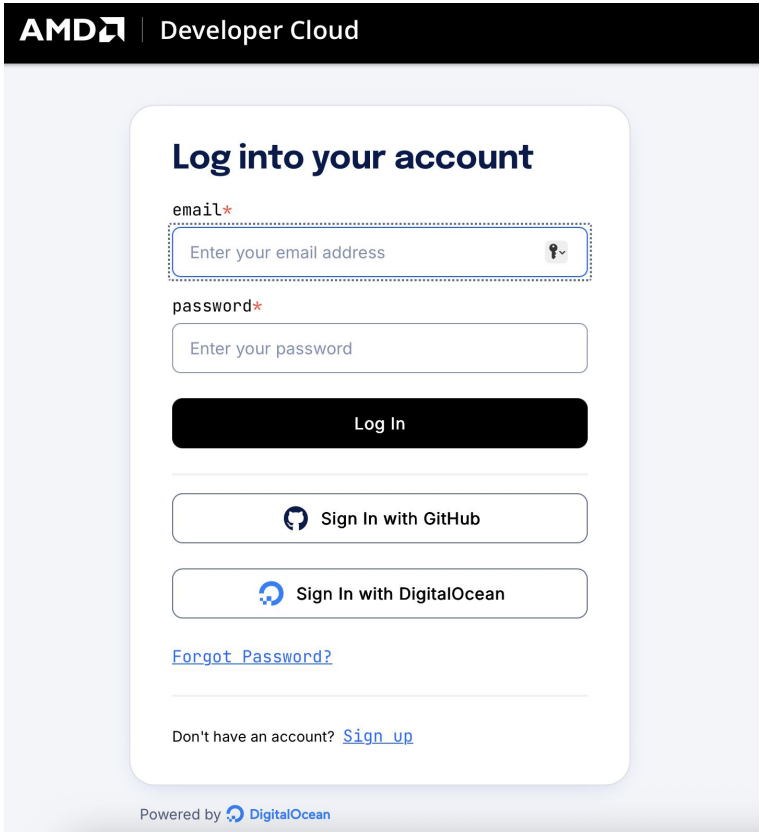
```
wget "https://raw.githubusercontent.com/unslothai/notebooks/refs/heads/main/nb/OpenEnv_gpt_oss_(20B)_Reinforcement_Learning_2048_Game_BF16.ipynb" -O "OpenEnv_NEW.ipynb"
```

How to form teams

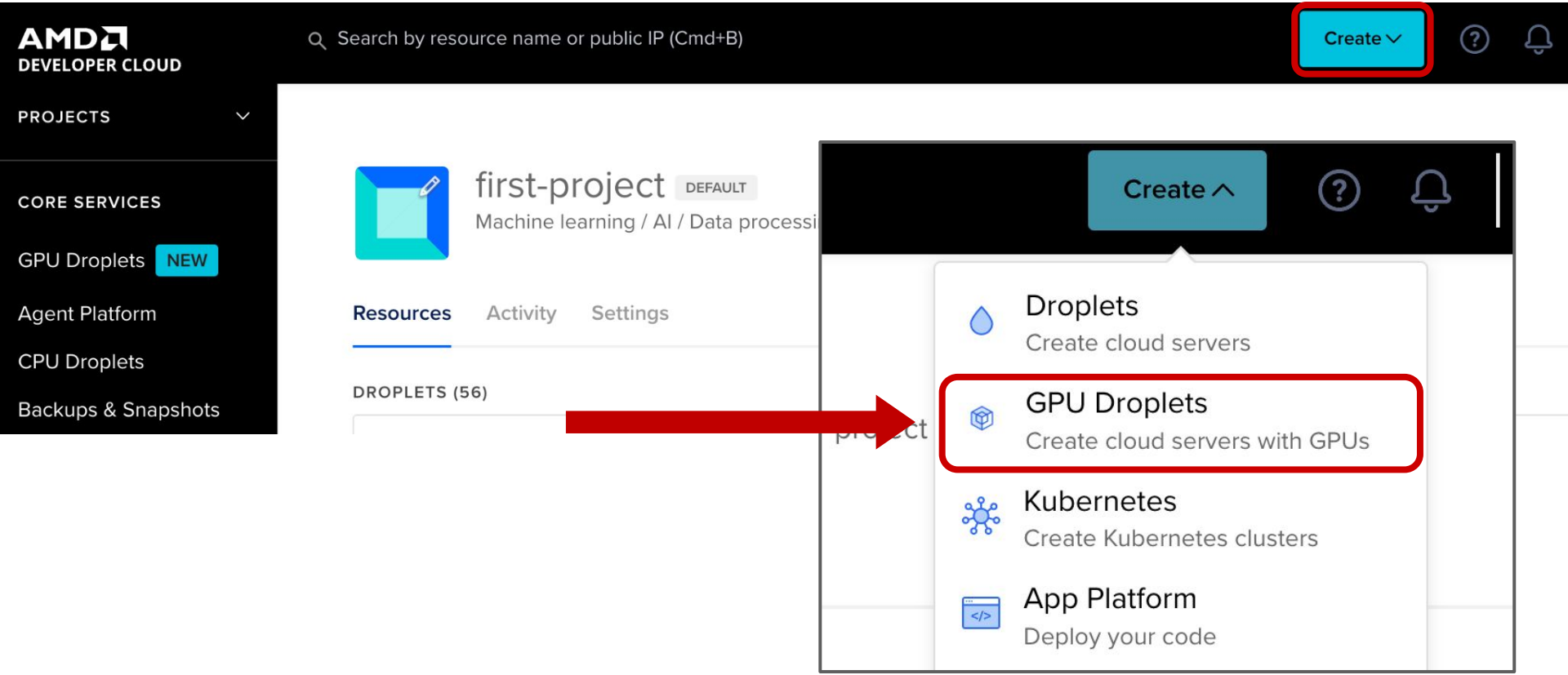
- Create an account on https://devcloud.amd.com/
- Fill out the Google form (one per team!) https://forms.gle/RPV7fURLNHDjz2yr9

Please see https://docs.unsloth.ai/new/unsloth-amd-pytorch-synthetic-data-hackathon

for debugging and more



Create a GPU Instance



Configure your instance

MI300X

Credits applicable 🎉

These GPU plans use your AMD GPU credits. You have AMD credits available for use.

This information is updated daily.

Note: After the credits have been used, it will charge your payment method. AMD credit only covers GPU access. All other services would be charged to your payment method.

YOU MUST USE ROCM 6.4.0 NOT 7

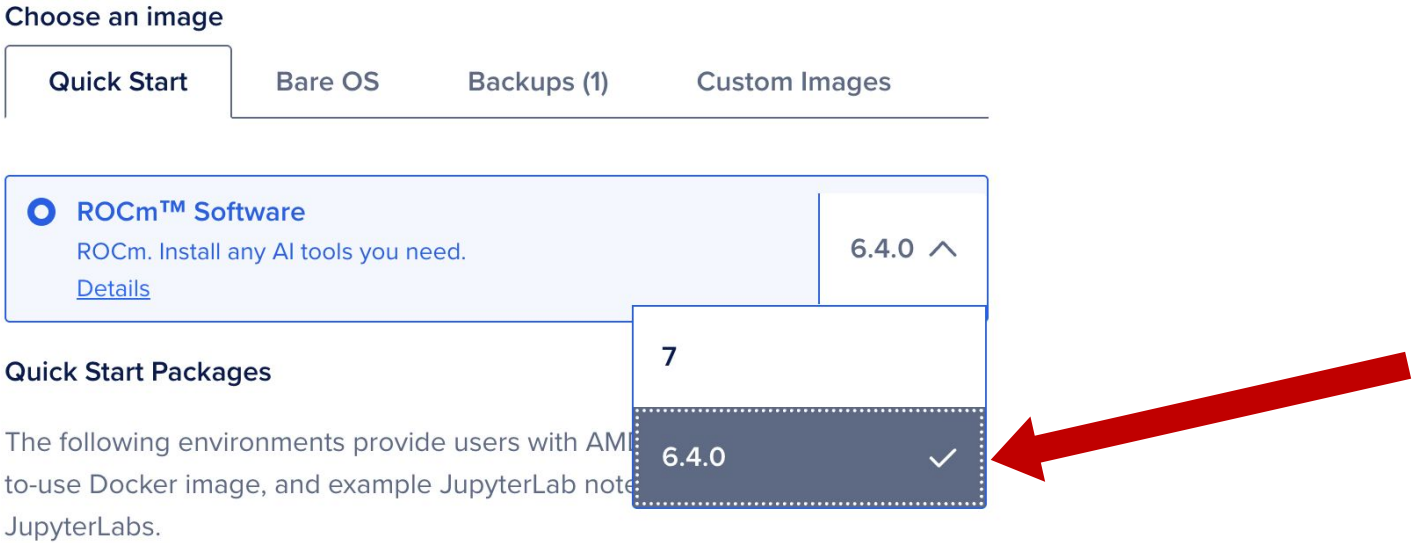
GPU Plans

MI300X x8
8 GPU - 1.5 TB VRAM - 160 vCPU - 1920 GB RAM
Boot disk: 2 TB NVMe- Scratch disk: 40 TB NVMe
$1.99/GPU/hr

MI300X
1 GPU - 192 GB VRAM - 20 vCPU - 240 GB RAM
Boot disk: 720 GB NVMe- Scratch disk: 5 TB NVMe
$1.99/GPU/hr

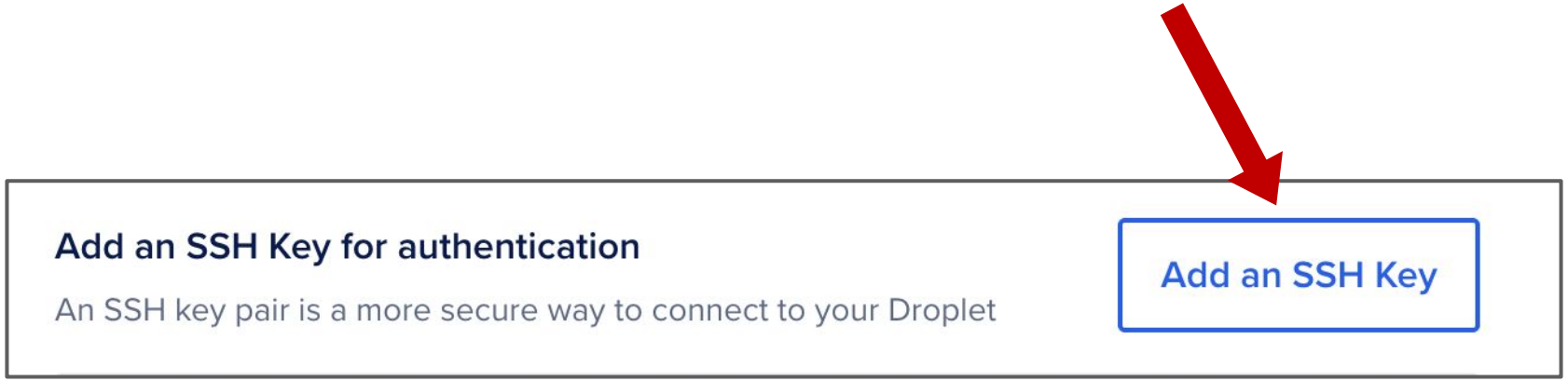
Configure your instance

YOU MUST USE ROCM 6.4.0 NOT 7

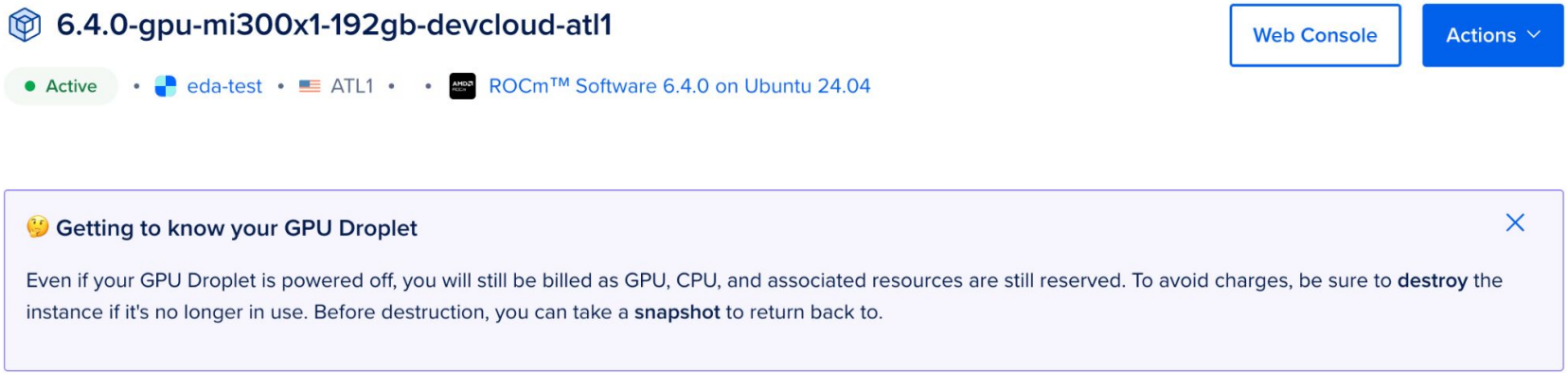


Configure your instance

If you're on a team, make sure to add your teammates ssh keys as well!

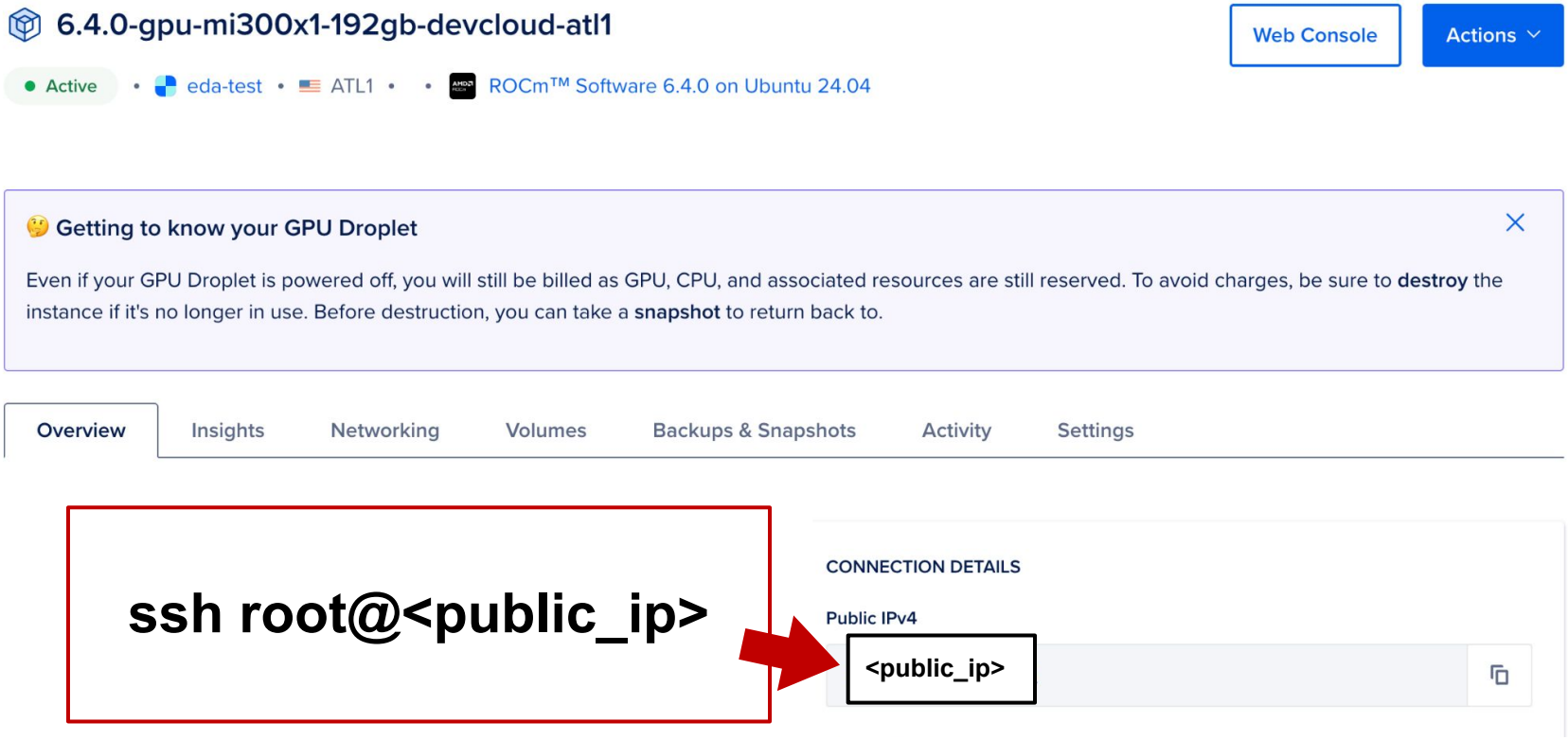


Configure your instance



Only start one instance or you will not have enough credits for submission!

Access your instance



For eg: Type **ssh root@134.XXX.XXX** into terminal

Start your workstation (Copy paste docker run)

In your terminal:

$ ssh root@<public_ip>

root@public_ip:~$

COPY PASTE BELOW:

```
docker run -it --rm \
  --network=host \
  --device=/dev/kfd \
  --device=/dev/dri \
  --group-add=video \
  --ipc=host \
  --cap-add=SYS_PTRACE \
  --security-opt seccomp=unconfined
  --shm-size 8G \
  -w /workspace \
  --name rocm-jupyter \
  edaamd/aiac:latest
```

Make sure you manually backup any work done in Jupyter! It will not persist if the Jupyter server or docker get kills

The Dockerfile is at https://github.com/edamamez/Unsloth-AMD-Fine-Tuning-Synthetic-Data/blob/main/AIAC/Dockerfile

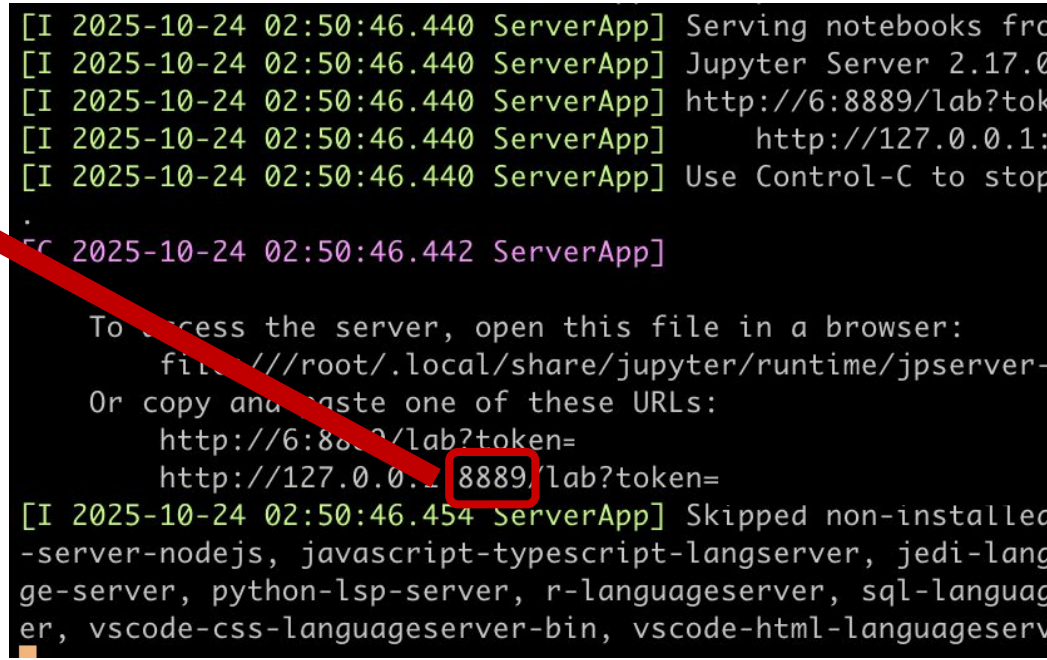
Accessing your workstation

In a **separate terminal**, start port forwarding:

$ ssh -N -v -L
localhost:8889:localhost:8889
root@<public_ip>

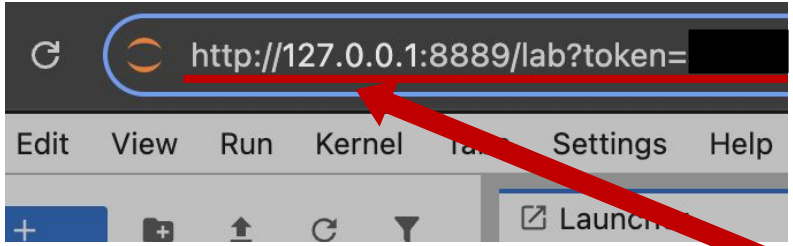
For eg in the new separate terminal:

```
ssh -N -v -L
localhost:8889:localhost:8
889 root@134.XXX.XXX.XXX
```



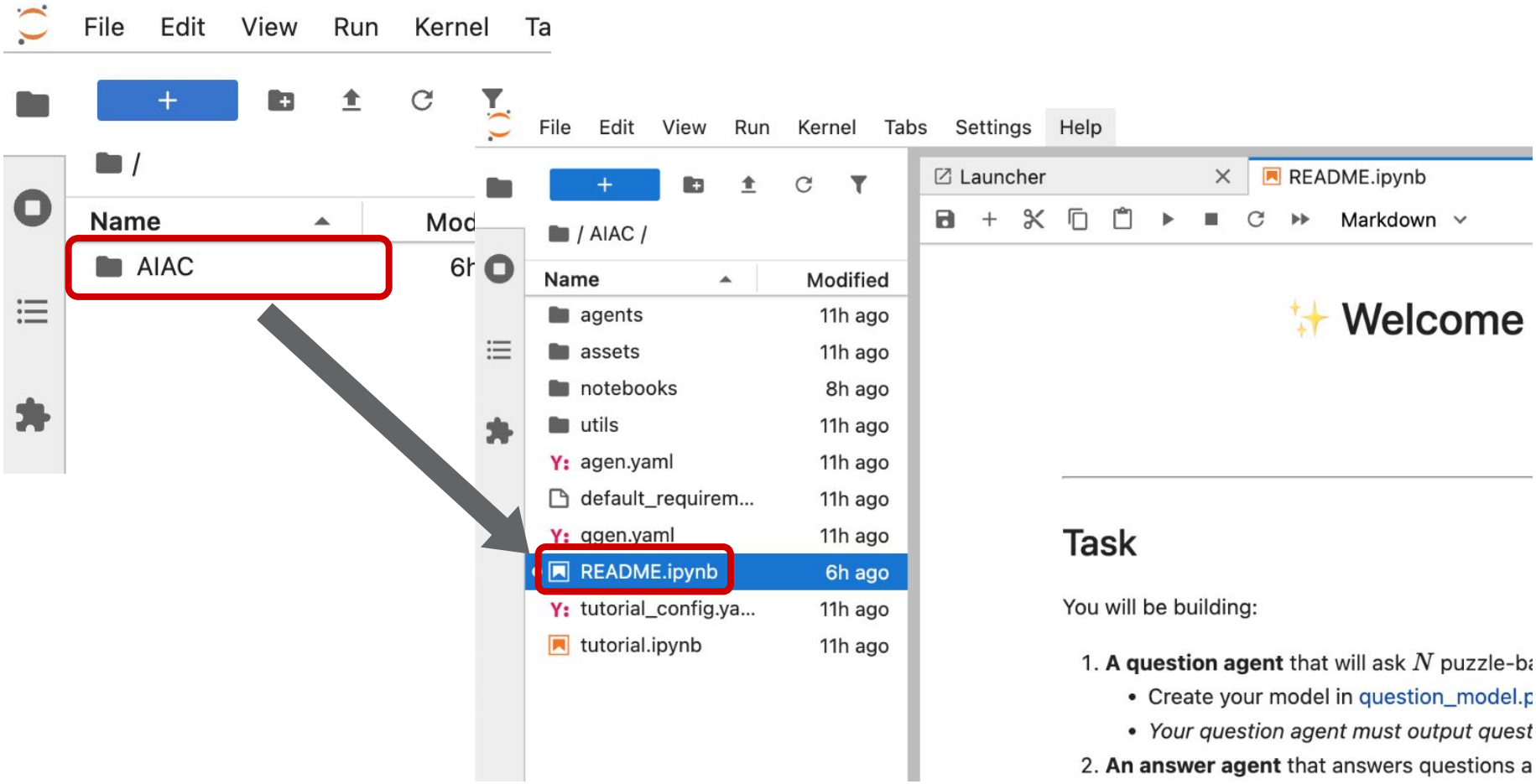
Accessing your workstation

Open jupyter server in your **local web browser**!

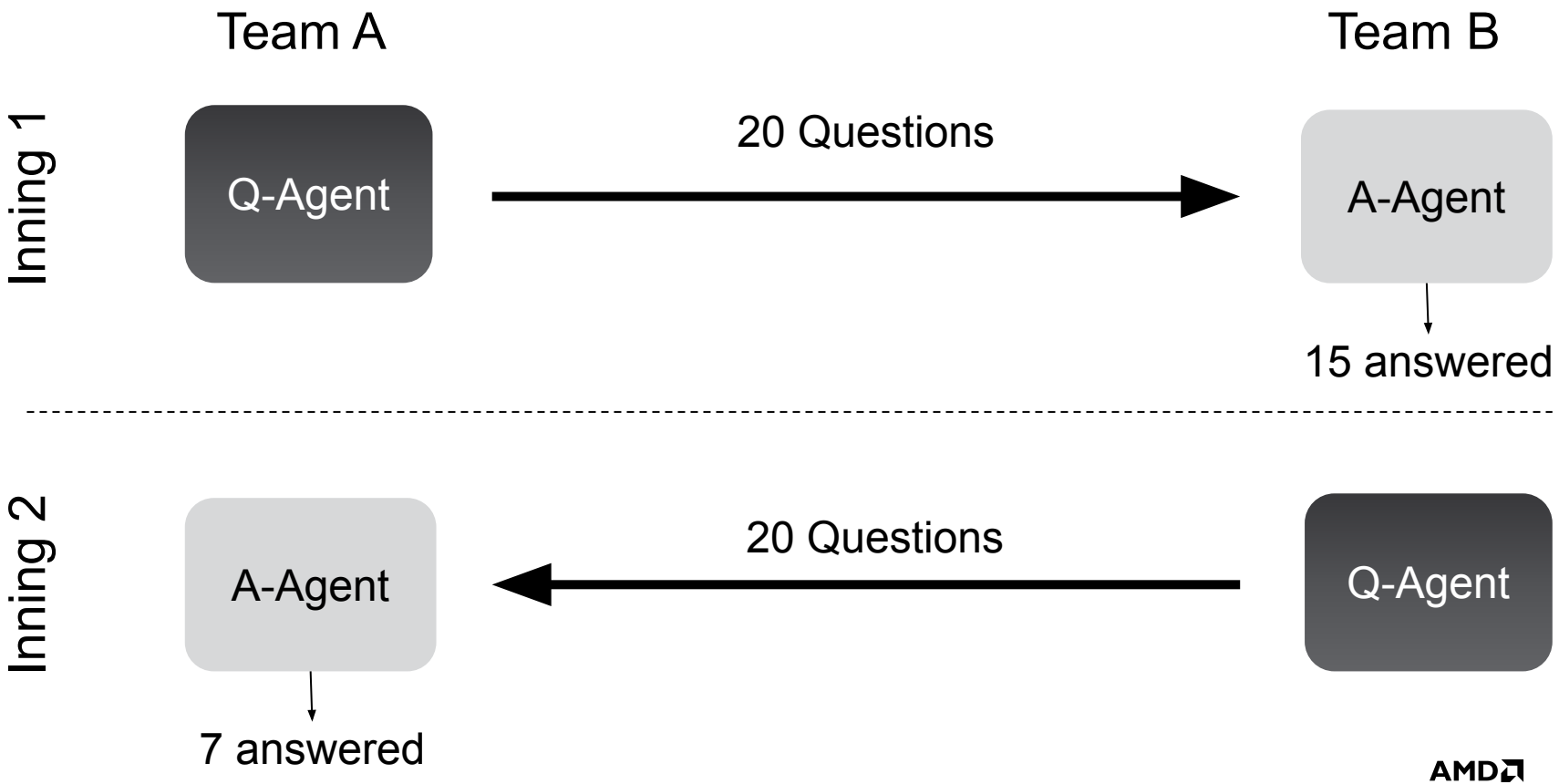



```
[I 2025-10-24 02:50:46.440 ServerApp] Serving notebooks fro
[I 2025-10-24 02:50:46.440 ServerApp] Jupyter Server 2.17.0
[I 2025-10-24 02:50:46.440 ServerApp] http://6:8889/lab?tok
[I 2025-10-24 02:50:46.440 ServerApp]     http://127.0.0.1:
[I 2025-10-24 02:50:46.440 ServerApp] Use Control-C to stop
.
[C 2025-10-24 02:50:46.442 ServerApp]

    To access the server, open this file in a browser:
        file:///root/.local/share/jupyter/runtime/jpserver-
    Or copy and paste one of these URLs:
        http://6:8889/lab?token=
        http://127.0.0.1:8889/lab?token=
[I 2025-10-24 02:50:46.454 ServerApp] Skipped non-installed
-server-nodejs, javascript-typescript-langserver, jedi-lang
ge-server, python-lsp-server, r-languageserver, sql-languag
er, vscode-css-languageserver-bin, vscode-html-languageserv
```



🏆AIAC Overview🏆



🏁AAIPL Score🏁

	Team A	Team B
Inning 1	For **Q**-Agent: 5/20	For A-Agent: 15/20
Inning 2	For A-Agent: 7/20	For **Q**-Agent: 13/20
	Team A Score: **12**	Team B Score: **28**

🥇 Team B Wins! 🥇

Guidelines

Format Overview

- Q-Agent: Given a topic, the Q-agent should generate questions in the specified JSON format:

```
{
"topic": "<Topic of the Question>",
"question": "<full question text>",
"choices": [
    "A) <choice A text>",
    "B) <choice B text>",
    "C) <choice C text>",
    "D) <choice D text>"
],
"answer": "<correct choice letter only>",
"explanation": "brief explanation within 100 words for why the answer is correct"
}
```

  from which we will extract **ONLY** the **"Question"** and **"Choices"** keys and feed it to the answer agent. The **"Topic"**, **"Question"**, **"Choices"**, and **"Answer"** will be verified for correctness from an Oracle.

- A-agent: Given a Question and Choices, A-agent should produce answer in the format of:

```
{
    "answer": "<correct choice letter only>",
    "reasoning": "brief reasoning within 100 words for why the answer is correct"
}
```

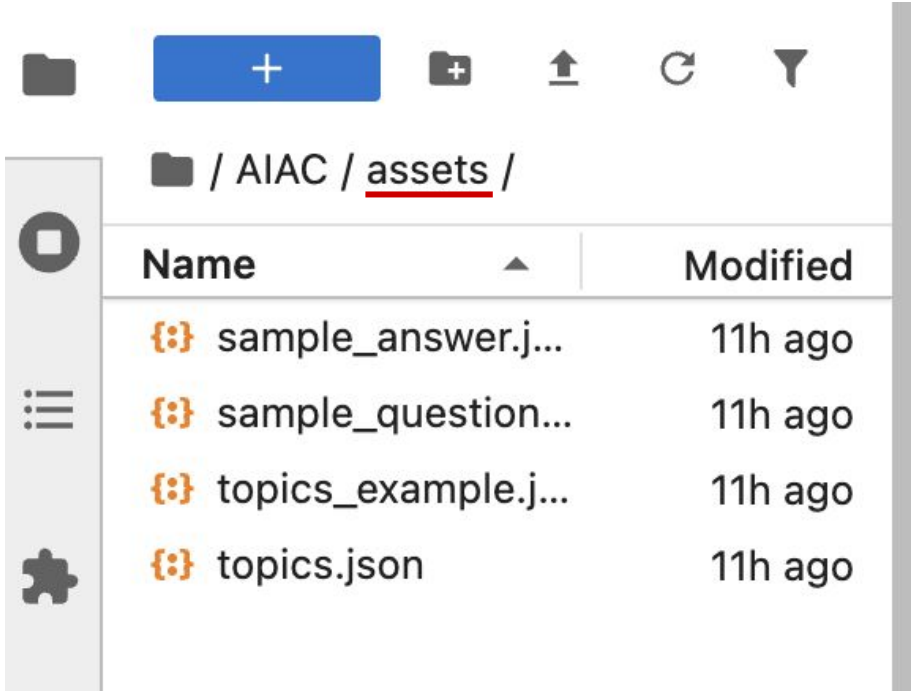
  where we will extract ONLY the **"Answer"** key and compare it with **"Answer"** from the opponent's question.

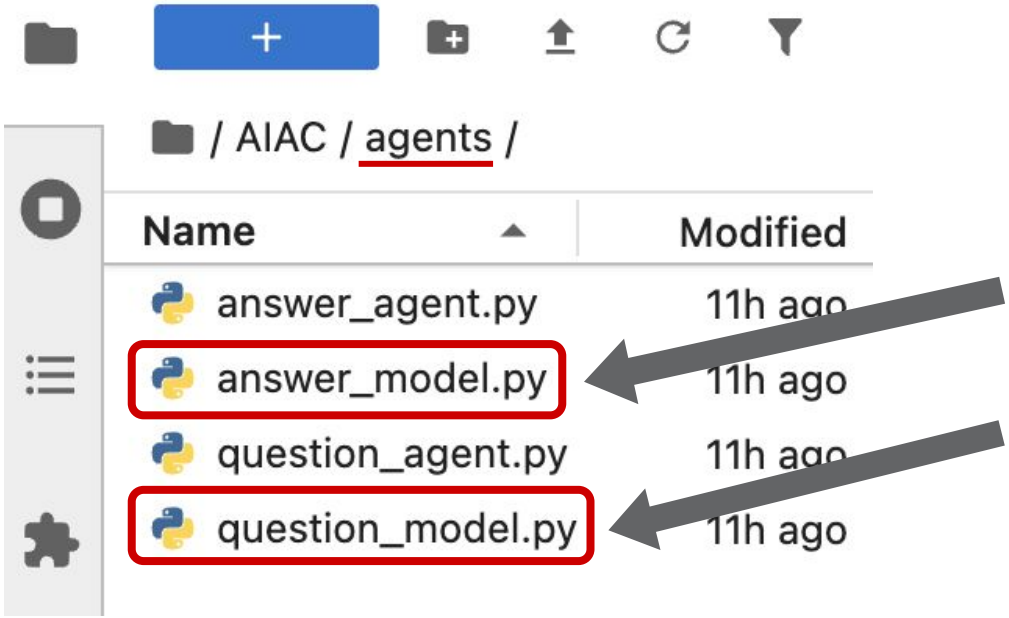
- *Remarks: Having explanation and reasoning is a plus. Not having them doesn't disqualify the question or answer being correct.*

Note: *We will only consider those responses from the Q-agent and the A-agent which follow the above format.*

Examples

1. *sample_answer.json* details how answers should look like.
2. *sample_questions.json* details how questions should look like.
3. topics.json consists of two topics.
   a. Seating Arrangements (Circular and Linear). *Don't include any numeric style seating arrangements questions, e.g., how many permutations such arrangements possible, etc.*
   b. Blood Relations and Family Tree
4. topics_example.json consists of **3** examples from each topic.





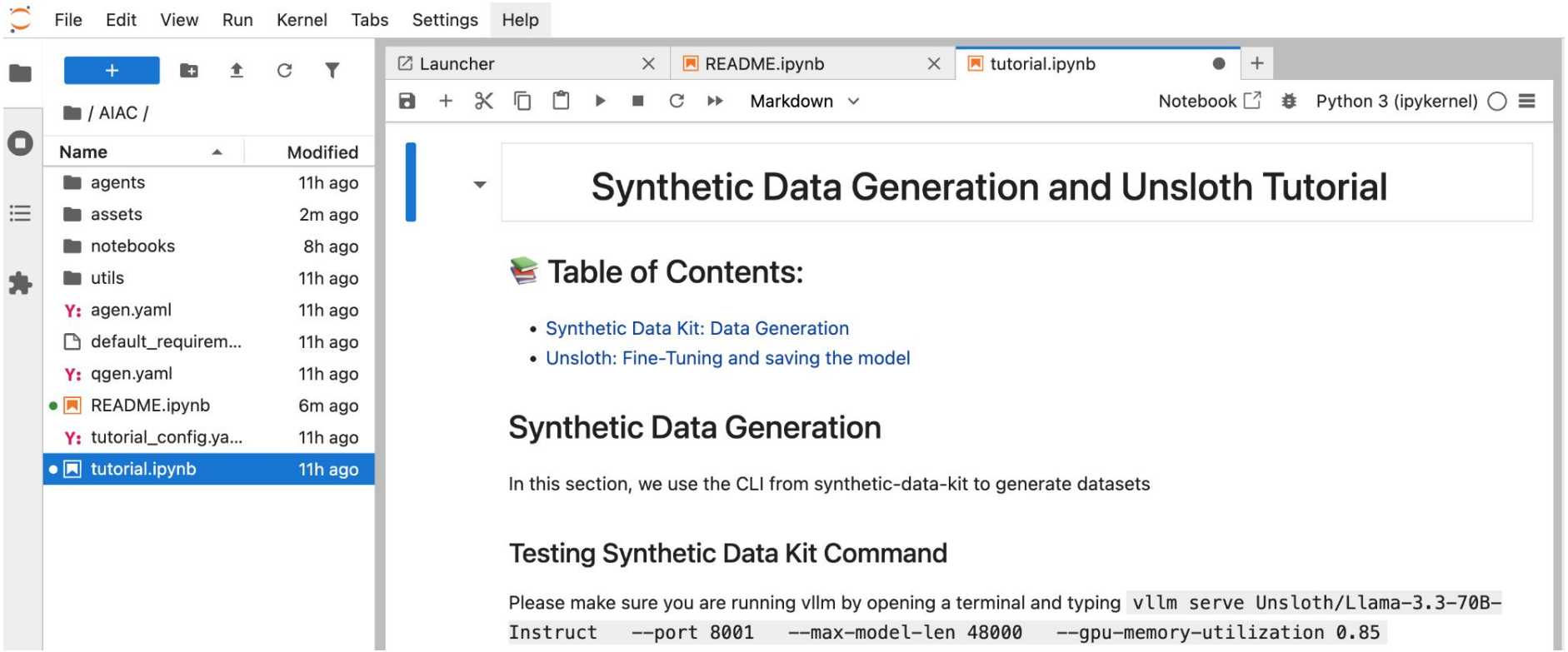
Put your code here!

Submission Check

```
# Run the following code to generate questions.
# For demo purpose, we have used the base Qwen3-4B model for Q-Agent.
!python -m agents.question_agent \
    --output_file "outputs/questions.json" \
    --num_questions 20 \
    --verbose
```

```
# Same instructions apply for the answer agent.
# For demo purpose, we have used the base Qwen3-4B model for A-agent.
!python -m agents.answer_agent \
    --input_file "outputs/filtered_questions.json" \
    --output_file "outputs/answers.json" \
    --verbose
```

Synthetic Data and Fine Tuning



What will you submit?

Submission

You need to submit your code which should contain these main files:

1. All work must be within the `AIAC` folder. Do NOT change the folder name.
2. No need to upload anything anywhere, we'll collect your agent code at from your Jupyter Server.
    A. The agents will be called by `python -m agents.question_agent` and `python -m agents.answer_agent`, respectively.
3. ENSURE model checkpoint(s) (e.g., `model.safetensors` or `.pt` or `.pth`) is(are) loading and expected files are getting generated from Q-agent and A-agent, when inference is done.
    A. Outputs must be saved to `outputs/questions.json` and `outputs/answers.json`, respectively.

You can test your submission by running the commands in the Getting Started section.

Restrictions

Deadline: **7 pm PT, Wednesday Oct 29**

1. **NO LAST Minute Submission**: The submission deadline is strict. Any changes to your code after the deadline may disqualify your submission.
2. RAG (Retrieval Augmented Generation) techniques are not allowed.
3. Adversarial approaches will lead to disqualification, e.g. making A-agents hallucinate.
4. Only English language is allowed for both Q-agent and A-agent.
5. Strictly stay within the `max_tokens` limits specified in `agen.yaml` & `qgen.yaml`. Other parameters can be changed.
6. Questions must pertain to the topics listed in `topics.json`.
7. Each question should be generated under `10 secs`. Questions exceeding this limit will not be considered.
8. Each answer should be generated under `6 secs`. Answers exceeding this limit will not be considered.

Feel free to reach out in the Discord channel for any clarifications or questions!

Takeaways

- We encourage everyone to **follow the rules** and the **format** strictly, otherwise your questions and answers won't be considered.
- Also very importantly, not just the format, but also the quality. **Question**, **Choices,** and **Answer** correctness of Q-agent.
- Further, give **equal** importance to your answer agent, i.e., A-agent.
- Ensure **four** `.py` files in `agents/` folder for both Q-agent and A-agent.
- Do NOT cross the RESTRICTIONS as all of them will be enforced.
- Feel free to ask any questions in the Discord!

Ground Rules

- Be Respectful of Participants
- Use AMD GPUs (Each team gets MI300X Access)
- Use Synthetic-Data (Synthetic-Data-Kit from Meta usage is encouraged)
- Think about maximising GPU Memory Usage: 192GB is a lot!
- Remember to have fun and Drink Chai!

Resources

- Everything will be shared in #challenge-news
- AMD Blog: https://www.amd.com/en/developer/resources/technical-articles/2025/10x-model-fine-tuning-using-synthetic-data-with-unsloth.html
- Unsloth Fine-Tuning Notebook: https://github.com/unslothai/notebooks
- Synthetic-Data-Kit: https://github.com/meta-llama/synthetic-data-kit/
- OpenEnv Reinforcement Learning 2048 Example: https://github.com/unslothai/notebooks/blob/main/nb/OpenEnv_gpt_oss_(20B)_Reinforcement_Learning_2048_Game_BF16.ipynb
- See https://docs.unsloth.ai/new/unsloth-amd-pytorch-synthetic-data-hackathon for debugging

Weather environment / other environments RL FAQ

If you are making a custom environment, I would:

1. Ask an Open Source LLM to generate a weather environment - ie it should say today's weather (sunny, not sunny), with temperature, humidity, wind speed etc etc. It should be governed by some physical laws or mathematical transition process (Markov Chains)
2. Create a reward function to predict the weather and do RL on this!

Get faster inference with vLLM

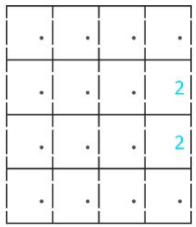
Unsloth supports faster inference for some models with fast_inference = True. For eg Llama, Qwen models. GPT-OSS will use Unsloth internal inference which is a little bit slower. See https://github.com/unslothai/notebooks/blob/main/nb/Advanced_Llama3_2_(3B)_GRPO_LoRA.ipynb Do:

```
import os
os.environ["UNSLOTH_VLLM_STANDBY"] = "1" # [NEW] Extra 30% context lengths!
from unsloth import FastLanguageModel
max_seq_length = 2048 # Can increase for longer reasoning traces
lora_rank = 64 # Larger rank = smarter, but slower
model, tokenizer = FastLanguageModel.from_pretrained(
    model_name = "meta-llama/Llama-3.2-3B-Instruct",
    max_seq_length = max_seq_length,
    load_in_4bit = False, # False for LoRA 16bit
    fast_inference = True, # Enable vLLM fast inference
    max_lora_rank = lora_rank,
)
```

ReadTimeoutError for OpenEnv

If you see the error below:

```
        else:  # right
            newb = [list(reversed(slide(list(reversed(r))))) for r in board]
        if newb != board:
            return str(m)
    return "0"
```




```
[2025-10-27 02:56:24] WARNING connectionpool.py:868: Retrying (Retry(total=2, connect=None, read=None, redirect=None, status=None)) after connection brok
n by 'ReadTimeoutError("HTTPConnectionPool(host='localhost', port=8081): Read timed out. (read timeout=15.0)")': /step
[2025-10-27 02:56:39] WARNING connectionpool.py:868: Retrying (Retry(total=1, connect=None, read=None, redirect=None, status=None)) after connection brok
n by 'ReadTimeoutError("HTTPConnectionPool(host='localhost', port=8081): Read timed out. (read timeout=15.0)")': /step
[2025-10-27 02:56:55] WARNING connectionpool.py:868: Retrying (Retry(total=0, connect=None, read=None, redirect=None, status=None)) after connection brok
n by 'ReadTimeoutError("HTTPConnectionPool(host='localhost', port=8081): Read timed out. (read timeout=15.0)")': /step
```

I added a fix since 1AM 27th October PT time. Please do:

1. **Update Unsloth** `pip install --upgrade --force-reinstall --no-deps --no-cache-dir unsloth unsloth_zoo`
2. **Re-get notebook as “OpenEnv_NEW.ipynb”:**

```
wget "https://raw.githubusercontent.com/unslothai/notebooks/refs/heads/main/nb/OpenEnv_gpt_oss_(20B)_Reinforcement_Learning_2048_Game_BF16.ipynb" -O "OpenEnv_NEW.ipynb"
```

vLLM GPT-OSS 20B / 120B

DO NOT UPDATE vLLM in the Docker!!

Do: **`pip install --upgrade numba numpy`**
Then serve it:

```
vllm serve unsloth/gpt-oss-20b \
  --no-enable-prefix-caching \
  --compilation-config '{"full_cuda_graph": true}' \
  --port 8001 \
  --max-model-len 48000 \
  --gpu-memory-utilization 0.85
```

For 120b:

```
vllm serve unsloth/gpt-oss-120b \
  --no-enable-prefix-caching \
  --compilation-config '{"full_cuda_graph": true}' \
  --port 8001 \
  --max-model-len 24000 \
  --gpu-memory-utilization 0.9
```

```
(APIServer pid=8919) INFO 10-27 08:49:23 [launcher.py:46] Route: /v1/responses, Methods: POST
(APIServer pid=8919) INFO 10-27 08:49:23 [launcher.py:46] Route: /v1/responses/{response_id}, Methods: GET
(APIServer pid=8919) INFO 10-27 08:49:23 [launcher.py:46] Route: /v1/responses/{response_id}/cancel, Methods
(APIServer pid=8919) INFO 10-27 08:49:23 [launcher.py:46] Route: /v1/chat/completions, Methods: POST
(APIServer pid=8919) INFO 10-27 08:49:23 [launcher.py:46] Route: /v1/completions, Methods: POST
(APIServer pid=8919) INFO 10-27 08:49:23 [launcher.py:46] Route: /v1/embeddings, Methods: POST
(APIServer pid=8919) INFO 10-27 08:49:23 [launcher.py:46] Route: /pooling, Methods: POST
(APIServer pid=8919) INFO 10-27 08:49:23 [launcher.py:46] Route: /classify, Methods: POST
(APIServer pid=8919) INFO 10-27 08:49:23 [launcher.py:46] Route: /score, Methods: POST
(APIServer pid=8919) INFO 10-27 08:49:23 [launcher.py:46] Route: /v1/score, Methods: POST
(APIServer pid=8919) INFO 10-27 08:49:23 [launcher.py:46] Route: /v1/audio/transcriptions, Methods: POST
(APIServer pid=8919) INFO 10-27 08:49:23 [launcher.py:46] Route: /v1/audio/translations, Methods: POST
(APIServer pid=8919) INFO 10-27 08:49:23 [launcher.py:46] Route: /rerank, Methods: POST
(APIServer pid=8919) INFO 10-27 08:49:23 [launcher.py:46] Route: /v1/rerank, Methods: POST
(APIServer pid=8919) INFO 10-27 08:49:23 [launcher.py:46] Route: /v2/rerank, Methods: POST
(APIServer pid=8919) INFO 10-27 08:49:23 [launcher.py:46] Route: /scale_elastic_ep, Methods: POST
(APIServer pid=8919) INFO 10-27 08:49:23 [launcher.py:46] Route: /is_scaling_elastic_ep, Methods: POST
(APIServer pid=8919) INFO 10-27 08:49:23 [launcher.py:46] Route: /invocations, Methods: POST
(APIServer pid=8919) INFO 10-27 08:49:23 [launcher.py:46] Route: /metrics, Methods: GET
(APIServer pid=8919) INFO:     Started server process [8919]
```

AMD

Good Luck!