

FROM DEVOPS TO LLMOPS

GENERATIVE AI FOR INFRA/OPS ENGINEERS

MATT FRANZ

<https://www.linkedin.com/in/matthewdfranz/>

<https://infosec.exchange/@mdfranz>

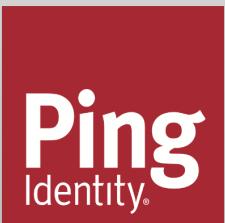
<https://github.com/mdfranz>

AGENDA

- OBJECTIVES & INTRO
- WHAT IS AI/LLMOPS?
 - LESSONS ON "OPS" FROM PREVIOUS "DISRUPTIONS"
 - SOME JOB DESCRIPTIONS AND AI HALLUCINATIONS
- **PAUSE FOR QUESTIONS**
- LLM FOUNDATIONS AND DEPLOYMENT MODELS
 - MODEL SELF-HOSTING
 - SAAS AND IAAS LLM OFFERINGS
- CHOOSE YOUR OWN OWN GEN AI OPS ADVENTURE
- **FINAL QUESTIONS & DISCUSSION?**

WHO AM I?

- TRANSITIONED TO TECH AFTER 3.5 YEARS AS MIDDLE SCHOOL TEACHER AND A DECADE AS USAR SIGINT ANALYST IN TEXAS IN THE LATE 90S
 - HELD VARIOUS LEADERSHIP ROLES FROM EM TO VP
 - SURVIVED 5 M&A BUT STILL NOT FIRE
- ENJOYS HISTORY, ASIAN/EUROPEAN TRAVEL AND EAST COAST HIKING
- CURRENTLY A PRINCIPAL SECURITY ENGINEER AT DCC



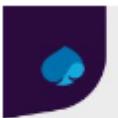
WHY ARE WE HERE?

We are <u>NOT</u> going to!	But you will!
Use LLMs to generate IAC (Ansible, Terraform, etc.) or talk about DevOpsGPT!	Understand how operational practices , tools and infrastructure platforms intersect with Gen AI
Learn whether RAG or Finetuning is a better approach for your app!	Gain awareness of LLM solution space and tooling ecosystem to identify a learning path based on your interests, abilities & experience
Become an amazing prompt engineer in just 30 minutes or develop multi-modal agents or custom ChatGPT's	

ARE AIOPS SRES THE NEW "UNICORNS?"

What is AIOps/LLMOps?

- Using AI to Do Ops
- The Ops & INFRA necessary to run AI



AIOps Site Reliability Engineer

Sogeti · Chicago, IL · 4 days ago · Over 100 applicants

Remote · Full-time · Mid-Senior level

10,001+ employees · IT Services and IT Consulting

18 school alumni work here

2 of 3 skills match your profile - you may be a good fit

- Machine Learning in IT Operations: ML algorithms, data preprocessing and feature engineering.
- AI/ML Tools: (i.e. TensorFlow, Keras, PyTorch) for development and deployment of efficient models.
- Incident Management: rapidly identifying, diagnosing, and resolving IT issues using automated and AI-driven tools.
- Data Analysis and Processing: working with large datasets and applying statistical techniques to understand patterns; manipulate and analyze data in creating predictive models.
- Integrated Observability: consolidating and analyzing data across systems to provide a holistic view of IT health and performance.
- Performance Optimization: leveraging AI insights to fine-tune IT systems for peak operational efficiency.
- Building and Managing AIOps Platforms: designing, implementing, and maintaining platforms that leverage AI to automate and enhance IT operations.
- Documentation: creating clear, concise, and comprehensive guides and records for AIOps processes and configurations.
- Programming: programming languages such as Python, Java, and others for writing AI and ML algorithms, integrating AI tools, and customizing solutions.

- Experience with web server configuration, monitoring/observability, trending, network design, high availability.
- Skilled at enhancing SRE by leveraging AI and machine learning capabilities to automate tasks, providing predictive analytics, and ultimately improving system reliability.
- Collecting and analyzing data on user behavior, website traffic, and server logs; providing valuable insights into trends, system bottlenecks, and potential security threats.
- 5+ years of experience as a Site reliability, Operations, Software engineer, AI/ML Ops SRE.
- Proficiency in AI/ML Tools (i.e. TensorFlow, Keras, PyTorch).
- Proficiency in a scripting language.
- Practical, solid knowledge of shell scripting and at least one higher-level language (Python or Ruby preferred).
- Expert level understanding of Linux servers, specifically RHEL/CentOS.
- Comfortable configuring DNS, DHCP, and LAN/WAN technologies.

MORE DEVOPS + ML HYBRID JD'S

Senior MLOPS/Devops Engineer

Xylo Technologies · United States · 8 hours ago · 13 applicants

 Remote · Full-time · Entry level

 51-200 employees · IT Services and IT Consulting

 Skills: Scala, DevOps, +8 more

 Am I a good fit for this job?

 How can I best position myself for this >

 Apply

 Saved

About the job

Title: Senior AI/Client Engineer (SENIOR MLOPS/DEVOPS ENGINEER)

Client: Mayo Clinic

100% remote

Required Technical Skills

Proficiency in programming languages such as Python, R, or Scala.

In-depth knowledge of machine learning frameworks and libraries (e.g., TensorFlow, PyTorch, scikit-learn).

Strong understanding of DevOps principles and practices.

Knowledge of infrastructure as code (IaC) tools like Ansible, Terraform, or CloudFormation.

Experience with monitoring and logging tools such as Prometheus, Grafana, ELK stack (Elasticsearch, Logstash, Kibana), or similar.

Understanding of networking, security, and data management principles.

Ability to work with big data technologies (e.g., Hadoop, Spark, Kafka).



Required Experience

Strong problem-solving and troubleshooting skills.

5-7 years of relevant experience in roles involving machine learning, software development, DevOps, or a combination thereof.

Experience in progressively challenging roles with demonstrated leadership, project management, and the ability to architect and implement complex systems.

Required Domain Expertise

Deep understanding of machine learning concepts, algorithms, and workflows.

Familiarity with various types of machine learning models (e.g., supervised learning, unsupervised learning, reinforcement learning).

Experience with real-world machine learning applications and challenges, including data preprocessing, feature engineering, model selection, and evaluation.

Knowledge of industry-specific requirements and compliance standards (e.g., GDPR, HIPAA, HITRUST) a plus

Bachelor's degree in Computer Science or a related field with 7 years of experience. Or Master's degree in Computer Science or a related field with 5 years of experience.

BACK TO BROOKLYN / SUMMER 2016

<https://charity.wtf/2016/05/31/wtf-is-operations-serverless/>

WHAT IS OPERATIONS?

Let's talk about what "operations" actually means, in the year 2016, assuming a reasonably high-functioning engineering environment.



What is operations?

Operations is the constellation of your org's technical skills, practices, and cultural values around designing, building and maintaining systems, shipping software, and solving problems with technology.

Core competencies of operations engineering

1. Scalability
 2. Resiliency
 3. Availability
 4. Maintainability
 5. Simplicity in complex systems
 6. Instrumentation and visibility
 7. Graceful degradation
-
- A graphic of a rainbow arching across the bottom right corner. Above the rainbow are several small, colorful hearts (green, yellow, purple, pink) of different sizes, some with outlines and some solid.

LET'S ASK THE LLMS (MISTRAL LITE, M1)

LLMOps is a **community-driven platform** for **operations and maintenance of Large Language Models (LLMs) that can be deployed on cloud infrastructure**. It aims to provide developers, researchers, and organizations with the tools and resources needed to **efficiently manage and scale their LLM deployments**.

The platform is designed to be flexible and scalable, allowing users to customize their deployment based on their specific needs and requirements. Additionally, it provides a user-friendly interface that makes it **easy for anyone to set up and maintain their own LLM environments without having to worry about the underlying infrastructure** or hardware.

LLMOps also offers a range of **monitoring and analytics tools to help users track performance, identify bottlenecks, and optimize their deployments for maximum efficiency**. This includes real-time data visualization and analysis, as well as the ability to set up alerts and notifications for critical events.

By providing these services in one place, LLMOps aims to **simplify the process of managing and scaling LLM deployments** and make it easier for organizations to take full advantage of the capabilities of large language models.

See <https://ollama.com/library/mistrallite>
(7B Parameter, 4 Bit Quantization)

LET'S ASK THE LLMS (COMMAND R, M3 PRO)

LLM Ops, short for Large Language Model Operations, refers to the practices and processes involved in **managing, optimizing, and maintaining large language models**. It encompasses the operational aspects related to the **deployment, monitoring, scaling, and updating of these complex models**. As large language models require significant computational resources, expertise, and infrastructure, **LLM Ops focuses on ensuring their effective and efficient utilization**. Organizations dedicated to developing and deploying language AI often have **dedicated teams responsible for LLM Ops**. These teams work closely with data scientists and engineers to streamline the deployment process, maximize model uptime, and provide reliable access to the models for internal or external users.

Some **key aspects of LLMOps** include:

1. Infrastructure Management
2. Model Monitoring and Performance Optimization
3. Scalability Planning
4. Model Updates and Versions
5. Security and Access Control
6. User Experience and Feedback
7. Cost Management
8. Incident Response

<https://ollama.com/library/command-r>

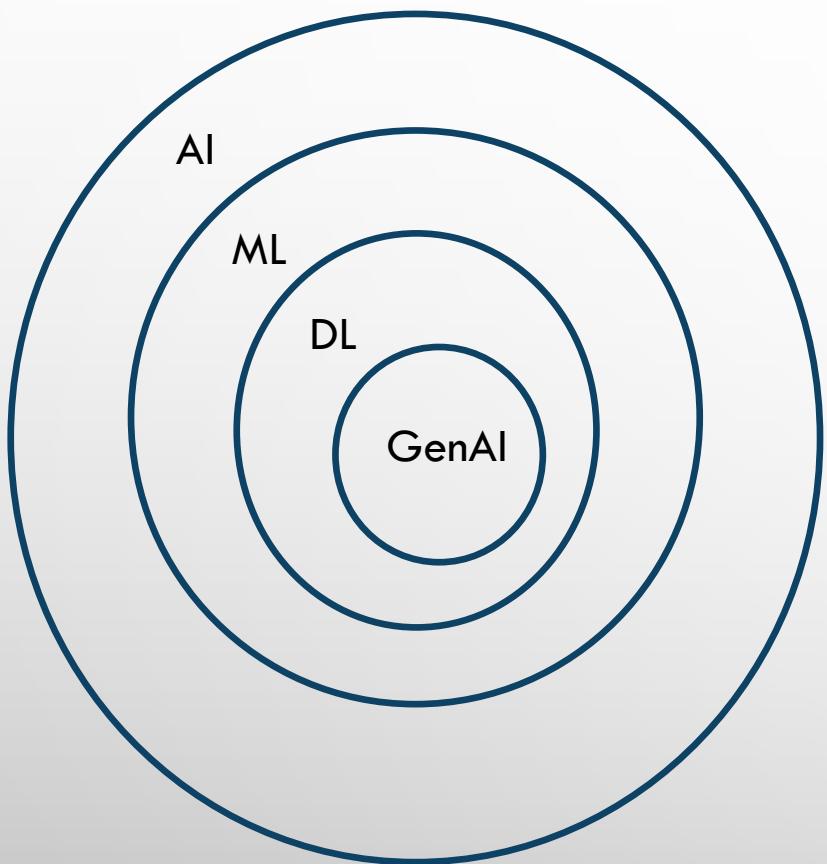
(35 Billion Parameter, 4-bit Quantization)

PLATFORM CONCERNS



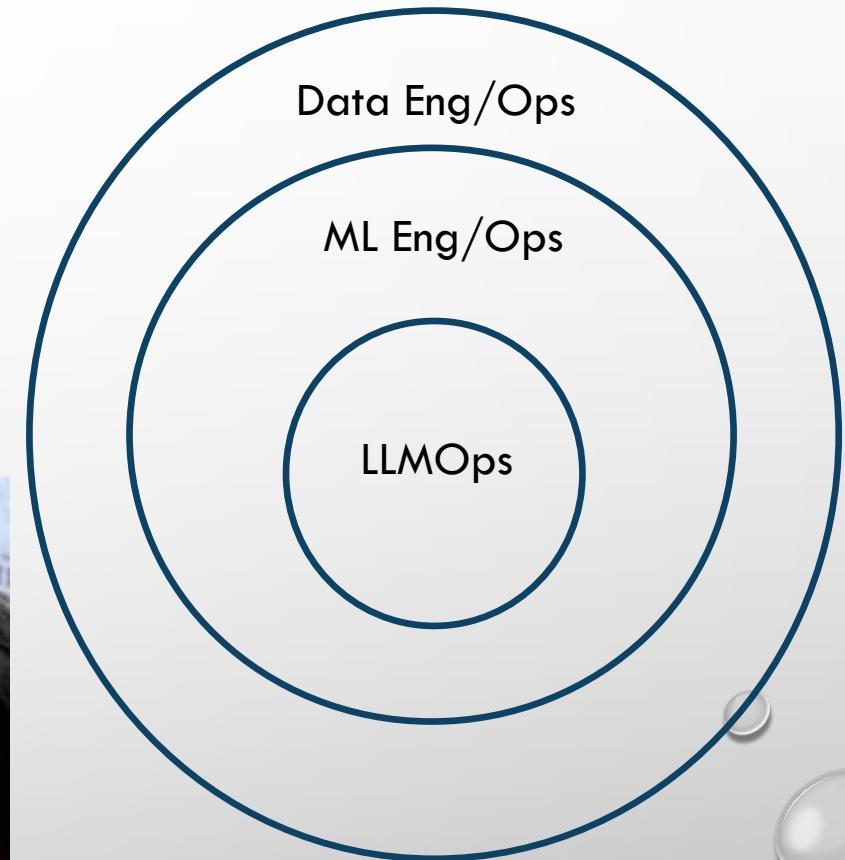
- Versioning & Artifact Management
- Configuration and Secrets Stores
- Environments & Pipelines & Promotions
- Deployment (rollback/forward)
- Logging & Observability
- Underlying Infrastructure Governance

ORIENTING OURSELVES



How do we apply the good engineering practices (versioned repeatable delivery, observability) of software delivery and deployment to data management, machine learning jobs, model training, etc?

How do we avoid AI ClickOps and Prompt Engineers gone wild?



Worldcue:
2020-01-23T03:23Z
Safety Verification
Request from T Rowe
Price: According to our
records, you might be
impacted by the Critical
Alert US - Police Activity
Jan. 22 - Seattle Upd. 1.
Please Confirm your
safety status by: AWARE
1=OK AWARE 2=Need
Assistance



DELIVERYCONF

Seattle, WA | Jan 21 & 22, 2020

Continuous Delivery For Machine Learning: Patterns And Pains

Implementing Continuous Delivery in a software context is already a challenge, but when it comes to building data-driven systems with machine learning, complex data architectures, and evolving demands, achieving a true Continuous Delivery state is even more difficult. From a lens of extensive experience in developing continuous delivery practices in software contexts, this talk will explore where things stand in achieving CD for data systems in practice, look at common pain points, and explore the patterns we regularly see in addressing these challenges.

Even before COVID it
was hard

And before LLMs and
GenAI and Vector DB

CI/CD + ML == MLOps - The Way To Speed Bringing Machine Learning To Production

Have you ever struggled with having different environments to build, train and serve ML models, and how to orchestrate between them? While DevOps and GitOps have made huge traction in recent years, many customers struggle to apply these practices to ML workloads.

This talk will focus on ways to effectively infuse AI into production-grade applications through establishing practices around model reproducibility, validation, versioning/tracking, and safe/compliant deployment. We will demonstrate how to run an E2E machine learning system using nothing more than Git. This will integrate DevOps, data and ML pipelines together, and show how to use multiple workload orchestrators together.

While the examples will be run using Azure Pipelines, Azure ML and Kubeflow, we will also show how to extend these platforms to any orchestration tool.

LLMS... FINALLY

GEN-AI FROM THE BOTTOM UP



MY BET ABOUT GEN AI AND OUR FUTURE (IN OPS)

- **JUST LIKE WITH CLOUD, IT WILL BE A MULTI-POLAR WORLD**

- MULTIPLE MODELS, ESPECIALLY OPEN SOURCE
- MULTIPLE BIG AI PROVIDERS , NOT JUST OPENAI

- **PRIVACY, SECURITY, AND DATA RESIDENCY WILL REMAIN CRITICAL REQUIREMENTS**

- PROBABILISTIC SECURITY AND PRIVACY RISKS
- CUSTOMERS WILL WANT TO RUN CSP (OR OPEN SOURCE MODELS) IN THE SAFETY OF THEIR VPC OR VNET

- **MODELS WILL CONTINUE TO BE SMALLER (QUANTIZED) AND MORE AND MORE CUSTOMIZED (FINE TUNED) FOR SPECIFIC USE CASES**

THERE WILL BE A PLACE FOR TESTING, DEPLOYING, AND MONITORING LLM-BASED APPLICATIONS



Nothing to see here.
Please disperse.

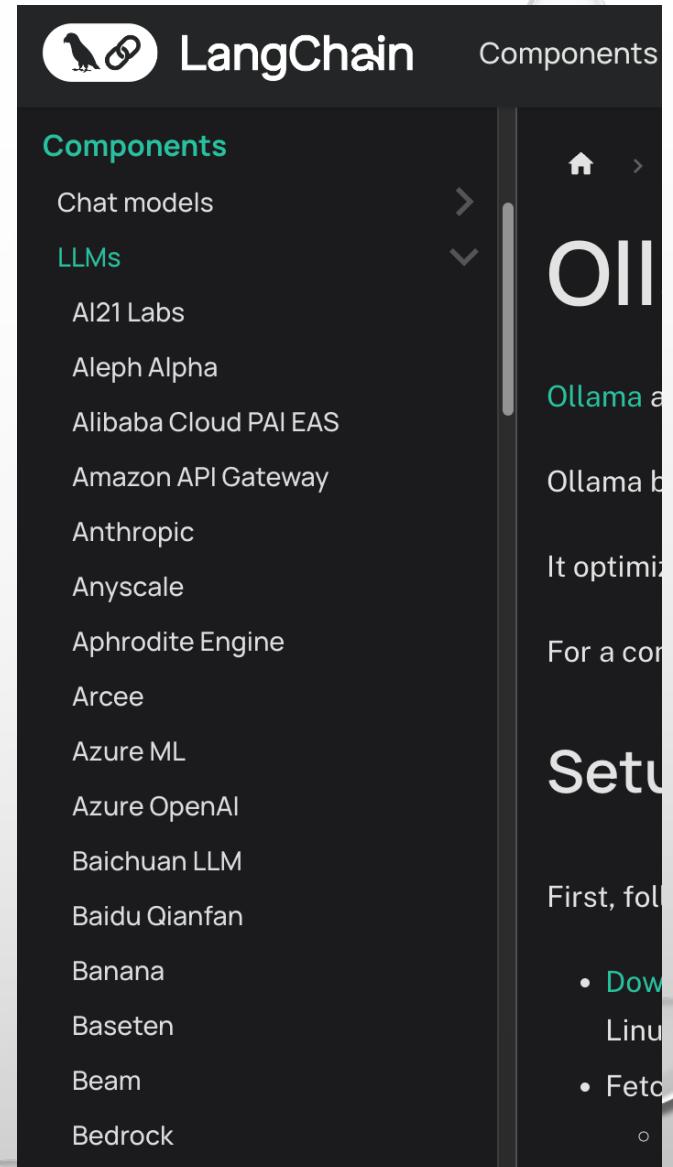
SELF HOSTED

- OPENLLM & VLLM
- OLLAMA
- LLAMA.CPP
- GPT4ALL
- LLAMAFILE

WHY LANGCHAIN IS HELPFUL (EVEN FOR OPS FOLKS)

SAAS AND IAAS

- AWS BEDROCK & SAGEMAKER
- OPENAI & AZURE ML
- COHERE
- GOOGLE (VERTEX & GEMINI)
- HUGGINGFACE
- CLOUDFLARE WORKERS



“ENTRY-LEVEL” GENAI PLATFORM ECOSYSTEM



Hugging Face



Ollama

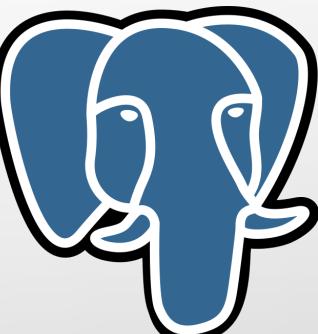


ANTHROPIC



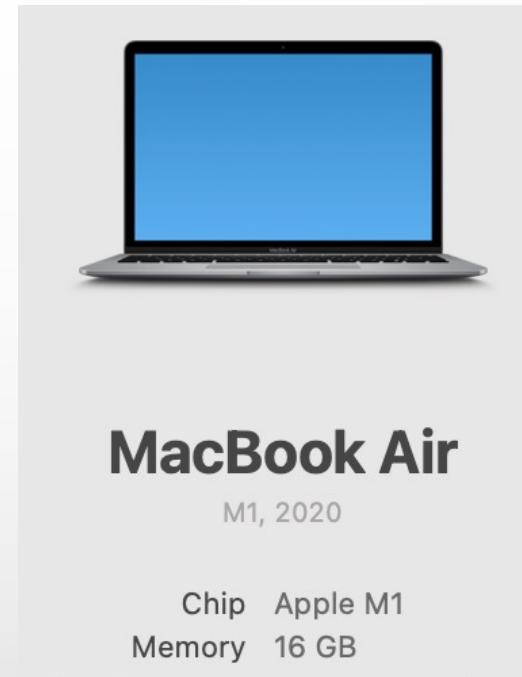
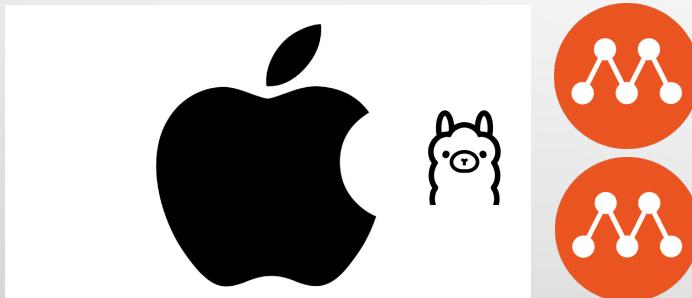
CUDA

LLaMA^{c+}



LOCAL HARDWARE OPTIONS (GPU-ONLY)

- APPLE SILICON FOR OLLAMA
 - 2020 MBA (16GB, M1, 7 CORE)
 - 2023 MBP (36GB, M3 PRO, 11 CORE)
- LINUX (ARM64) VMS WITH MULTIPASS



MINIMAL GCE WORKSTATION FOR OLLAMA (AND AWS INSTANCE TYPES, TOO)

(See <https://cloud.google.com/compute/docs/gpus>)

Graphics processing units (GPUs) accelerate specific workloads on your instances such as machine learning and data processing. [Learn More](#)

GPU type: NVIDIA T4 Number of GPUs: 1

Enable Virtual Workstation (NVIDIA GRID)

Series	Description	vCPUs	Memory	Platform
<input checked="" type="radio"/> N1	Balanced price & performance	1 - 96	1.8 - 624 GB	Intel Skylake

Machine type

Choose a machine type with preset amounts of vCPUs and memory that suit most workloads. Or, you can create a custom machine for your workload's particular needs. [Learn more](#)

PRESET **CUSTOM**

n1-standard-4 (4 vCPU, 2 core, 15 GB memory)

vCPU	Memory
4 (2 cores)	15 GB

ADVANCED CONFIGURATIONS

Availability policies

VM provisioning model: **Spot**

Monthly estimate

\$139.93

That's about \$0.19 hourly

Pay for what you use: no upfront costs and per second billing

Item	Monthly estimate
4 vCPU + 15 GB memory	\$27.73
1 NVIDIA T4	\$102.20
100 GB balanced persistent disk	\$10.00
Logging	Cost varies
Monitoring	Cost varies
Total	\$139.93

[Compute Engine pricing](#)

[Cloud Operations pricing](#)

[▲ LESS](#)

[Amazon EC2 G6 instances](#) are designed to accelerate graphics-intensive applications and machine learning inference.

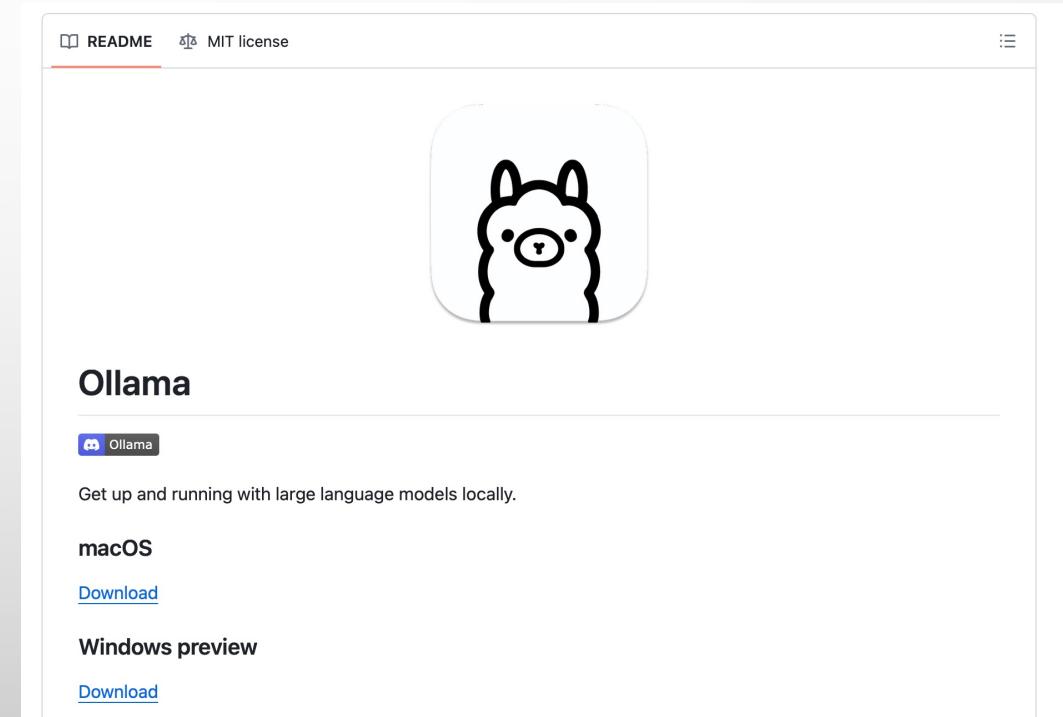
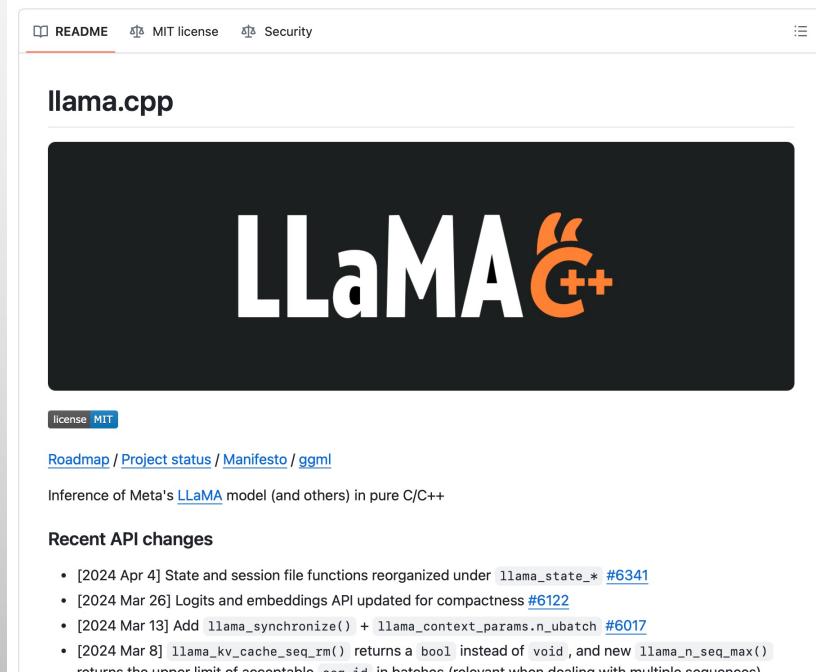
Features:

- 3rd generation AMD EPYC processors (AMD EPYC 7R13)
- Up to 8 NVIDIA L4 Tensor Core GPUs
- Up to 100 Gbps of network bandwidth
- Up to 7.52 TB of local NVMe local storage

Instance Name	vCPUs	Memory (GiB)	NVIDIA L4 Tensor Core GPU	GPU Memory (GiB)	Network Bandwidth (Gbps)***	EBS Bandwidth (Gbps)
g6.xlarge	4	16	1	24	Up to 10	Up to 5
g6.2xlarge	8	32	1	24	Up to 10	Up to 5

YOUR FIRST TWO TOOLS (ASSUMING YOU CAN GPU)

- [HTTPS://GITHUB.COM/GGERGANOV/LLAMA.CPP](https://github.com/ggerganov/llama.cpp)
- [HTTPS://GITHUB.COM/OLLAMA/OLLAMA](https://github.com/ollama/ollama)

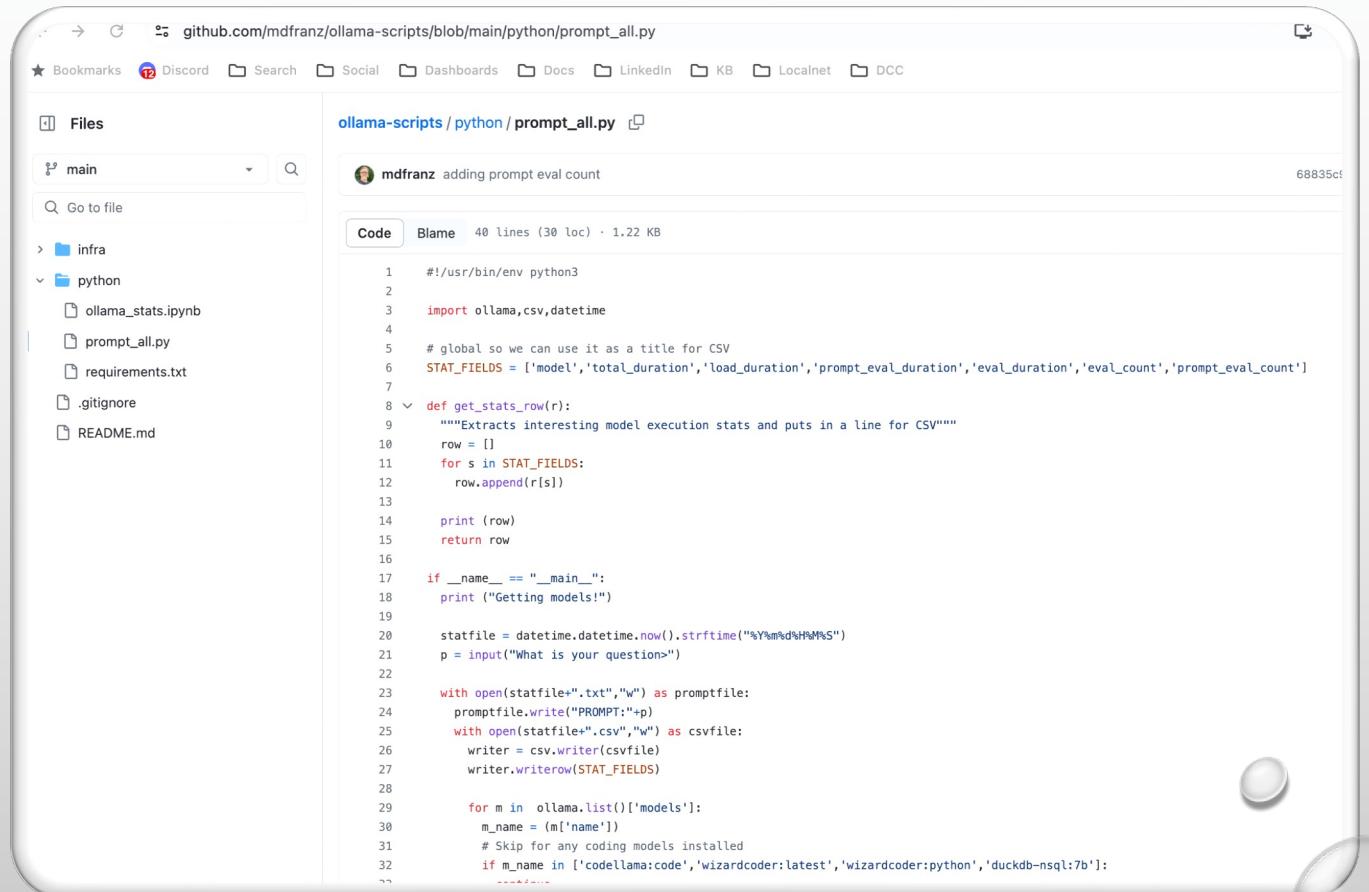


ALSO LEARN MODEL CARDS AND FILE FORMATS

The screenshot shows the Hugging Face platform interface for the Mixtral-8x22B-v0.1 model card. At the top, there's a search bar and navigation links for Models, Datasets, Spaces, Posts, Docs, and Pricing. Below the header, the model card for 'mistral-community/Mixtral-8x22B-v0.1' is displayed. It includes tabs for Model card, Files and versions (which is selected), and Community (with 17 members). There are buttons for Train, Deploy, and Use in Transformers. A license link for apache-2.0 is also present. The 'Files and versions' section lists several files: .gitattributes, README.md, RELEASE, config.json, convert.py, generation_config.json, and three large safetensors files (model-00001-of-00059.safetensors, model-00002-of-00059.safetensors, and model-00003-of-00059.safetensors). Each file entry includes a download icon, size, commit message, and timestamp (either 1 day ago or 8 days ago). A note at the top of the list mentions official weights (#15) and a VERIFIED badge.

- GGML
- GGUF
- SAFETENSORS

WRITE SOME PYTHON TO GENERATE INFERENCES (AND LEARN THE APIs)



The screenshot shows a GitHub browser window with the URL github.com/mdfranz/ollama-scripts/blob/main/python/prompt_all.py. The left sidebar shows a file tree for the 'main' directory, including 'infra', 'python' (containing 'ollama_stats.ipynb', 'prompt_all.py', 'requirements.txt'), '.gitignore', and 'README.md'. The right pane displays the content of 'prompt_all.py'. The code is as follows:

```
#!/usr/bin/env python3
import ollama, csv, datetime
# global so we can use it as a title for CSV
STAT_FIELDS = ['model', 'total_duration', 'load_duration', 'prompt_eval_duration', 'eval_duration', 'eval_count', 'prompt_eval_count']

def get_stats_row():
    """Extracts interesting model execution stats and puts in a line for CSV"""
    row = []
    for s in STAT_FIELDS:
        row.append(r[s])
    print (row)
    return row

if __name__ == "__main__":
    print ("Getting models!")

statfile = datetime.datetime.now().strftime("%Y%m%d%H%M%S")
p = input("What is your question>")

with open(statfile+".txt","w") as promptfile:
    promptfile.write("PROMPT:" + p)
with open(statfile+".csv","w") as csvfile:
    writer = csv.writer(csvfile)
    writer.writerow(STAT_FIELDS)

for m in ollama.list()['models']:
    m_name = (m['name'])
    # Skip for any coding models installed
    if m_name in ['codellama:code','wizardcoder:latest','wizardcoder:python','duckdb-nsql:7b']:
        continue
    else:
        r = ollama.stats(m_name)
        row = get_stats_row()
        writer.writerow(row)
```

THEN MOVE ONTO OPENLLM

(And add more GPU memory
and fight with Python
dependencies!)

README Code of conduct Apache-2.0 license Security



OpenLLM: Self-Hosting Large Language Models Made Easy

[PyPI v0.4.44](#) [PyPI Nightly](#) [Follow Us @bentomlai](#) [Join Us OpenLLM](#) [Continuous Integration failing](#)

[pre-commit.ci passed](#)

[Python 3.8 | 3.9 | 3.10 | 3.11 | 3.12](#) [Hatch](#) [code style Google](#) [Ruff](#) [types mypy](#) [types pyright](#)

Run any open-source LLMs, such as Llama 2 and Mistral, as OpenAI-compatible API endpoints, locally and in the cloud.

LEARN TOOLS (AND TERRAFORM) RESOURCES FOR YOUR CLOUD OF CHOICE

- AWS SAGEMAKER AND BEDROCK
- AZURE MACHINE LEARNING
- GOOGLE VERTEX AI



CONCLUSIONS (OR WHY YOU DON'T NEED TO WORRY ABOUT CHATGPT AND AGENTS!)

- YOU DON'T NEED TO BE A DATA SCIENTIST, KNOW ALL THE PYTHON ML FRAMEWORKS TO DO LLMOPS – MANY OF THE SKILLS YOU HAVE (APPS & INFRA & SAAS) ARE RELEVANT
- CHOOSE SOME REALISTIC KNOWLEDGE (OR CODE/DATA GENERATION) TASKS AND EXPERIMENT WITH THE RESULTS (AND GPU) AND PERFORMANCE
 - START TO USE [HTTPS://LLM.DATASETTE.IO/EN/STABLE/](https://llm.datasette.io/en/stable/) INSTEAD OF CHATGPT
- EXPERIMENT WITH VARIOUS MODEL HOSTING

FOR MORE INFO

- [HTTPS://GITHUB.COM/MDFRANZ/CHEETSHEETZ/BLOB/MAIN/AI/LLM/LLMOPS.MD](https://github.com/mdfranz/cheetsheetz/blob/main/AI/LLM/LLMOPS.md)

The screenshot shows a GitHub repository page for the user 'mdfranz' with the repository name 'cheetsheetz'. The 'Code' tab is selected. On the left, there is a sidebar titled 'Files' showing a tree view of files and a search bar. The main area displays a list of commits for the branch 'ai'. Each commit includes the author's profile picture, the commit message, and the date it was made.

Name	Last commit message	Last commit date
...		
books	more llmops	last month
llm	Sunday Monday	4 days ago
README.md	I hate VSC	last month
adoption.md	rebuff	last year
agents.md	done done ping	2 months ago
conversational.md	More LLM refactoring	last month
devtools.md	tweaks	last year
infra.md	more llmops	last month
keras.md	frameworks	10 months ago
models.md	Ops is not dead	last month