Overview

- Intro to Paperspace
- Developing with Language Models on Paperspace
- Paperspace GPU offerings
- The Serge application
    - What does it do?
    - How does it use LangChain?
    - Additional useful information
- How to use Serge with Paperspace
    - Setting up Paperspace
    - Creating a Deployment through the UI
    - Interacting with your deployment through the API Endpoint
- Demo
- Deploying with LangServe and Paperspace

Paperspace provides GPU power for AI model training and deployment at scale

Develop

Launch a notebook to build a proof of concept

Train

Train or fine-tune AI models with machines

Deploy

Convert models into scalable API endpoints

Built for AI Developers
key technical differentiation

GPU Acceleration

Access powerful GPU instances, such as NVIDIA H100s, Tesla, and Quadro GPUs, which are specifically designed for AI and machine learning workloads.

Pre-configured Deep Learning Containers

Run pre-configured Docker containers with popular deep learning frameworks like TensorFlow, PyTorch, and Keras.

Powerful Machine Learning Platform: Gradient

Gradient is Paperspace's machine learning platform, which provides a suite of tools and services to streamline the development, training, and deployment of AI models.

Collaboration and Sharing

Paperspace provides collaboration and sharing capabilities, allowing AI developers to work together on projects more effectively. Developers can easily share and collaborate on experiments, models, and data with their teammates, accelerating the development and iteration process.

How Customers use Paperspace products
understanding the solution difference

Notebook

Launch a notebook to build a proof of concept

Notebook provides a cloud-based integrated development environment (IDE) for data science and machine learning. It offers pre-installed packages and libraries commonly used and allows users to write, run, and collaborate on code directly within the browser.

Machines

Train or fine-tune AI models with machines

Virtual machines can be easily provisioned and accessed for various computing tasks. These machines are pre-configured with popular deep learning frameworks and can be used for training and running complex machine learning models.

Deployments

Convert models into scalable API endpoints

Deployments enable users to deploy and run a variety of software applications in the cloud. The solution offers pre-configured application stacks for tasks like web development, gaming, and AI inference.

Cost Advantage with Paperspace

Paperspace is uniquely positioned to help **SMBs and Startups** build and deploy GPU centric AI applications in an easy and affordable way.

GPU	Paperspace	AWS
A100-80Gx1	$3.18	NA
A100-80Gx2	$6.36	NA
A100-80Gx4	$12.72	NA
A100-80Gx8	$25.44	$32.77

GPU	Paperspace	Azure
A100-80Gx1	$3.18	$3.70
A100-80Gx2	$6.36	$7.34
A100-80Gx4	$12.72	$14.69
A100-80Gx8	$25.44	$32.7

AWS GPUs are 30% more expensive

Azure GPUs are 15-30% more expensive

Paperspace can provide total savings of up to 50% or more when usage extends to storage, bandwidth and other product utilization

Developing with Language Models on Paperspace

- It's easy to collaborate within a Teamspace across various projects using Notebooks, and then, after placing applying that work in a Docker container, deploy the application with Paperspace Deployments
- You can also access 8xH100 or A100-80G GPUs on our vm platform Core with quick SSH access to begin training/fine-tuning models, on demand
- Integrations with HuggingFace, AWS S3, and DigitalOcean Spaces make it straightforward to plug and play different models with the same templatized application as well

| Notebooks	Core Machine	Deployments
Test your code, data engineering, application development, etc.	Get full VM access to train models and build/test Docker images	Deploy containerized as API endpoint

Paperspace GPU offerings

Name	GPU (GB)	vCPUs	CPU RAM (GB)	Price (hourly)	Linux	Windows	Regions
GPU+ (M4000)	8	8	30	$0.45/hr	✅	✅	NY2 CA1
P4000	8	8	30	$0.51/hr	✅	✅	All
P5000	16	8	30	$0.78/hr	✅	✅	All
P6000	24	8	30	$1.10/hr	✅	✅	All
RTX4000	8	8	30	$0.56/hr	✅	✅	All
RTX5000	16	8	30	$0.82/hr	✅	✅	NY2
A4000	16	8	45	$0.76/hr	✅	✅	NY2
A5000	24	8	45	$1.38/hr	✅	✅	NY2
A6000	48	8	45	$1.89/hr	✅	✅	NY2
V100	16	8	30	$2.30/hr	✅		NY2 CA1
V100-32G	32	8	30	$2.30/hr	✅		NY2
A100	40	12	90	$3.09/hr	✅		NY2
A100-80G	80	12	90	$3.19/hr	✅		NY2
H100	80	20	250	$5.95/hr	✅		NY2

What is Serge?

- Per the Github repo: "Serge is a chat interface crafted with llama.cpp for running GGUF models. No API keys, entirely self-hosted!"
- In practice, it is an application for serving LLMs with a SvelteKit frontend, Redis chat history & storage management, and a FastAPI + LangChain API to wrap calls to llama.cpp using Python bindings
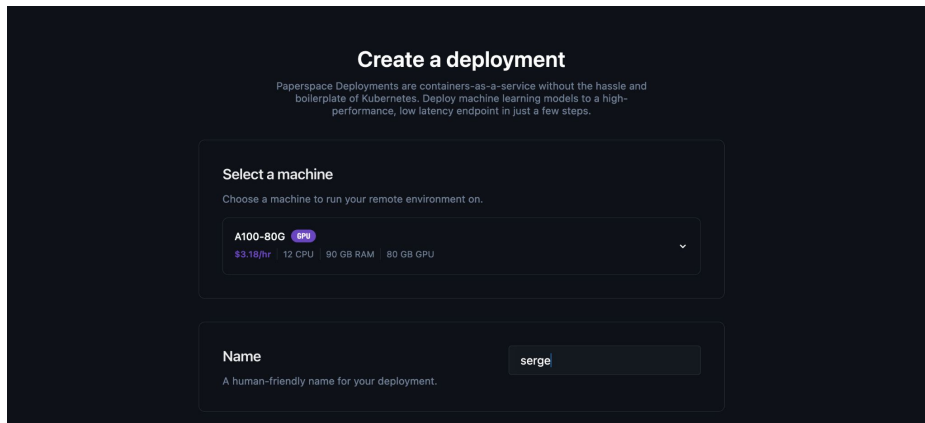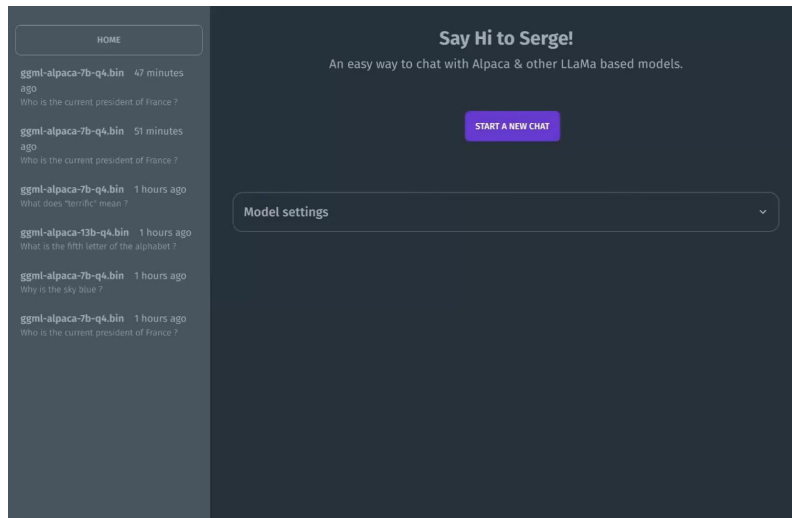
How does Serge use LangChain?

- Looking deeper into the application files (namely at chat.py and stream.py), we can see that the `langchain` package is used specifically to enable the RedisChat Message History for memory management of the chat & several schema related functions like AIMessage, HumanMessage, and SystemMessage, that help organize our chat history
- Together, these LangChain features work to functionally log the interactions between the user, system, and model, enable quick access to previous chats logged by the Redis database, and enable other quality of life features like history purging in an application

How to deploy Serge with Paperspace

- In the Github repo, follow the instructions on the README for the 'paperspace' subdirectory in Lab 5, and walkthrough it together

Interacting with your deployment through the API Endpoint

- Clicking on the API Endpoint URL linked in the top right corner of the deployment details page will give you access to your Deployed UI
- We then need to download some models, of which a wide variety of open source options are available
- Then we can set out parameters and chat!

Demo

- Application demo
  - Github
- Deployment spec: paste these values into gradient deployments to test the application.
  - image: serge-chat/serge:latest
  - port: 8008
  - resources:
  - replicas: 1
  - instanceType: A100-80/g <- I recommend this machine. See list of GPUs for options.

Deploying with LangServe and Paperspace

- Since LangServe allows devs to deploy LangChain chains as a REST API
- LangServe is helpfully written with integration to the familiar FastAPI, and comes with several useful features:
  - "Input and Output schemas automatically inferred from your LangChain object, and enforced on every API call, with rich error messages"
  - The /invoke/, /batch/, and /stream/ endpoints allow for many concurrent requests on a single server, which can massively save costs on scaling projects
  - Javascript client
- Next session, we will look at how LangServe can be used to develop a chat application with open source models from scratch