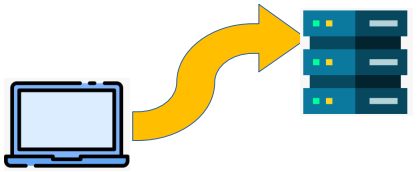
Home-lab-2-prod-ai-golden-path

Sep 2025

1- Objective

- Play with AI with easy entry level concepts from SRE - Platform or Data Engineer – Architect – DevOps **eyes**.
- Create a POC to play like a **backbone** that can be enhanced little by little.
- Create a technological **golden path**, opinionated shortcut, that can be reused from home lab to PROD.
- Mini state of the art: from ML towards MCP agents.

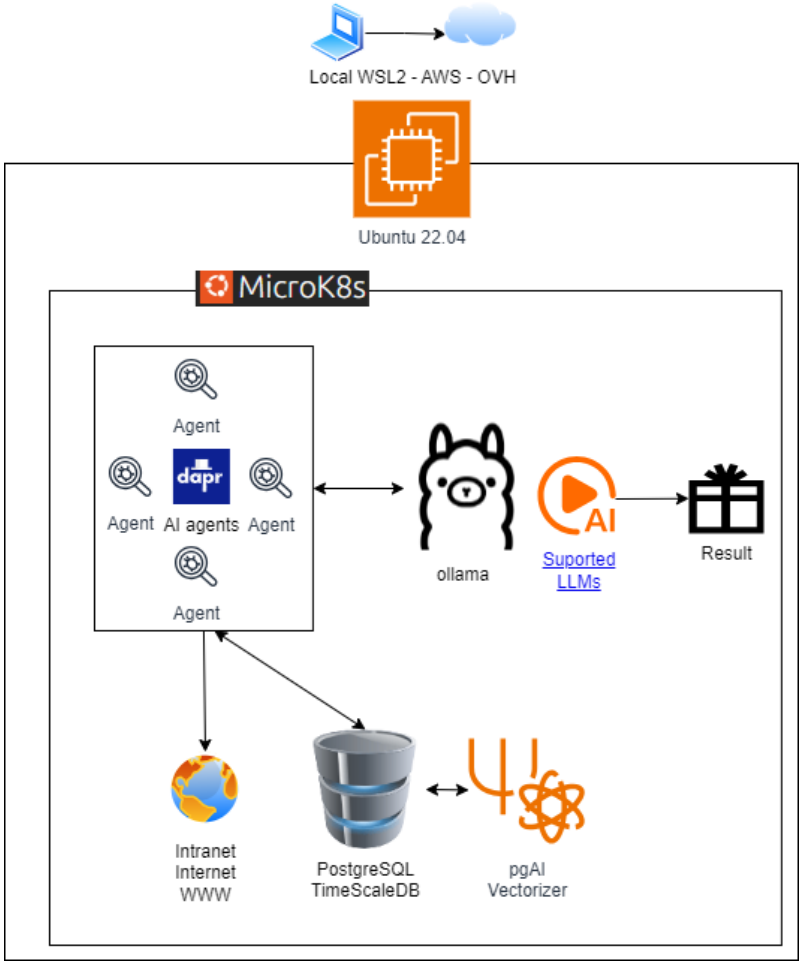


1- Objective: golden path

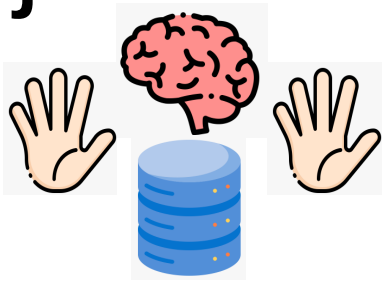
- Select and integrate OSS projects curated by CNCF to respect vendor neutrality.
- Privacy: local AI and local data.
- Technological independence, Europe first.

Tarif 15 %

Source glamour.com



1- Objective: create a 1st POC



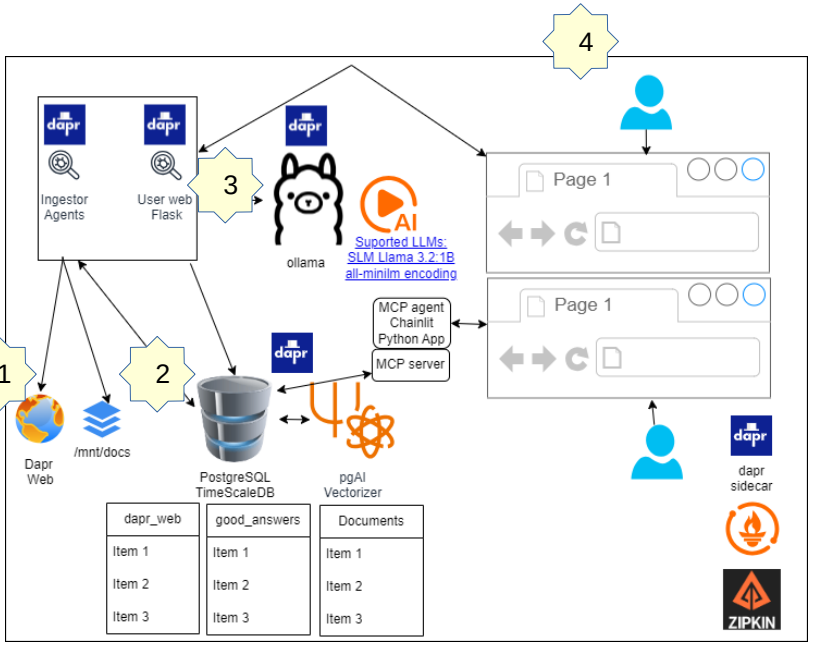
- Broad definition of **AI agents + RAG** : AI Agents (hands, gets data) - LLM (brain, process data and answers) – RAG database (authoritative knowledge base).

1 • Python program scraper reads web content and save into PostgreSQL DB . Another agent vectorizes documents.

2 • Pg AI chunks and vectorizes inserted data **automatically**, this is our RAG database where semantic search is done. Pg AI can vectorize sources from S3 or PostgreSQL and can integrate with Ollama or OpenAI.

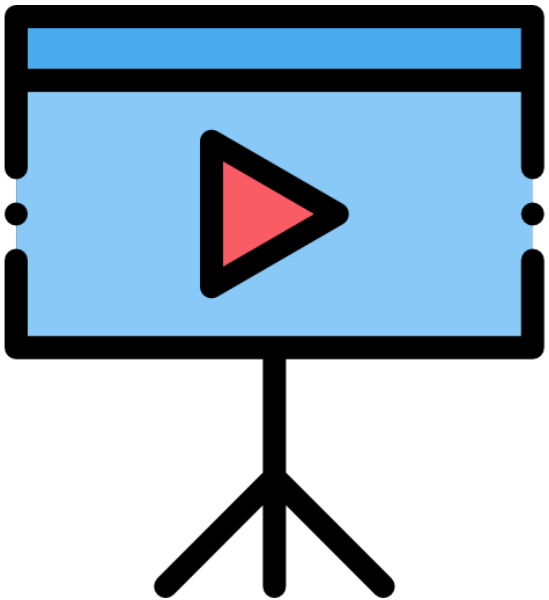
3 • Ollama allows you to run large language models (LLMs) locally on your own computer.

4 • Python Flask user web where user can write prompt and receive and answer based on local RAG database and local Ollama LLM, optionally configured with OpenAI as well. Good answers can be saved back to RAG database. MCP arch.



2- Demo time

- LLM (Large Language Model) vs SLM (Small): trained with less parameters so we need less hardware resources to run it. Demo running in a laptop WSL2 Ubuntu with 10GB RAM & 4 cores assigned.
- TimescaleDB/PostgreSQL RAG database vectorizer encoding is setup to all-minilm which is OSS and maps sentences & paragraphs to a dimensional dense vector space and can be used for semantic search. **Performance**: Ollama cpu, pg db chunk size, original data language, save contexts and metrics (tokens, time …) to integrate with Prometheus.
- Ollama LLM selection: **llama3.2:1b** is an SLM which **license** is not fully OSS but is very permissive, supports all-minilm encoding, honesty temperature parameter / prompt, can use RAG (authoritative knowledge base) and supports different language support (original RAG data language matters in semantic search). **TinyLlama (1.1B)** is fully OSS but performance is a little behind **llama3.2:1b**.
- Dapr microservices building blocks – maximizes standardization and code reuse changing configuration files only. Dapr is also evolving into an AI Agent framework.
- Dapr eases Open Telemetry monitoring Zipkin tracing – Prometheus metrics.
- Tilt deploy directly into K8S – mini demo with a small change.



2- Demo time: project structure

- Github – MIT license. "Built with Llama" extra license. TimescaleDB license is against cloud Hiperscalers only.
- Detailed explanations in README.md in each folder.
- 1-IaC: shell script to install in WSL2 - OpenTofu IaC for AWS, OVH or Hetzner (U know, Europe first). OpenTofu supports many providers.

```
├── 1-IaC: we choose kubernetes microK8S implementation as the neutral vendor platform to run differents POCs
│   ├── AWS: TF/OpenTofu scripts to run it in AWS Ubuntu 22.04
│   └── WSL2: steps/shell script how to install it locally in WSL2 - Ubuntu 22.04
│   └── OVH-Hetzner: TF/OpenTofu scripts to run it in Hetzner Ubuntu 22.04 European provider
├── 2-mandatory-k8s-services: mandatory services to install in K8S
│   ├── dapr: distributed application runtime helping microservices standardization and agent implementation
│   ├── ollama: local LLM offering an API
│   ├── timescaleDB: postgreSQL with vector database support, our RAG database, pgAI vectorizer
├── 3-dapr-microservices-agents: microservices and agents inserting data into database and passing it trought to LLM
│   ├── 1-user-web
│   ├── 2-injection-agent-web-dapr
│   ├── 3-injection-agent-docs
│   ├── 4-MCP
└── 4-optional-k8s-services: monitoring with Zipkin, Prometheus & Grafana - DONE.
│                            Jupyter Notebooks, MLflow vs KubeFLow, ArgoCD - Pending.
└── Docs
```

3- Mini state of the art

- SLM are ideal for specific domains knowledge and the **future of Agentic AI according to NVIDIA**. Install and play, have your own internal RAG database filled with your web/intranet or documents and expose it to your own users.
- Depending on your data type can work better with semantic data search (understand meaning of natural language) vs vectorscale pg extension search (image similarity & anomaly detection sensor data).
- This POC could work in Raspberry Pi industrial environments on the edge.
- This POC is suitable for Lab/Dev environments & Educational environments to gain a hands on experience.
- You can play with Machine Learning installing **MLFlow** or **Kubeflow** or **both**(MLFlow better for tracking experiments VS Kubeflow better for pipelines) on K8S.
- AI agent concept is a rapidly evolving trend and Dapr offers its framework to it – **Dapr AI Agents framework** and **MCP Quickstarts**, see **MCP architecture image**.

4- Use the force, Luke!

- MIT report 95% of GenAI projects are getting zero retun. Systems do not learn. Focus on ROI business cases. User context & remembering and AI agents iceberg is more infra than programming.
- Take a look to from lab to PROD hard path: IMHO technological change management best practice should be this way, see image.
- Data quality is the most important and still may be undervalued factor in AI. There is a huge Diogenes syndrome on companies data. Do you want to be a data engineer hero? - AI and synthetic data pros and cons.
- Quality AI services are paid + Cloud Technofeudalism.
- Too many AI news, all is AI. Impostor syndrome. Cognitive overload. Gartners's Hype Cycle
- Use the force, Luke!
  - What you hear, you will forget - What you do, you will understand - Explain to others, you will understand even deeper.
  - Do not be a perfectionist and fail fast to learn fast. Play hard to work smart
  - If you want to be fast, do it alone. If you want to go far, do it with company. If the company is not good, it is better to go alone.

5 - Q&A - Contribute

- Share your experience or thoughts.
- Tell me if you need any help with your lab.
- This project is a learning by doing experience - If something is wrong or there is an error, comment it but do not be a hater ;) .
- Feedback points: YouTube or GitHub or LinkedIn.

6- Some references

- Most of the icons in this presentation are from flaticom.com, flaticom license (Free for personal and commercial purpose with attribution)
- Wizard of Oz 1st slide image link
- Use the force Luke link