November 2024

PaaS: A Scalable and Cost-Effective Private Cloud

Case Study by Ovais Quraishi

Objective

A Product Engineering team of 6 needs a low-cost yet performant platform to experiment with building Agentic-AI based applications and services. The team is looking to build and deploy an MVP to raise additional post-seed funding. Depending on time and cost constraints, the team is willing to assemble a solution by putting something together. Wherever possible, open-source solutions should be used. Ultimately, both time and cost are of the essence.

Key Requirements:

- AI infrastructure relies mainly on open-source LLMs (no training/fine-tunning)
- User Apps: Data visualization, Jenkins, SonarQube, Web UI for chat/prompt running LLMS
- LLMs are cached and performant
- Should be able to interface with OpenAI and AnthropicAI APIs at a later stage
- Ensures that application data, and Agentic-AI data is stored and stays local
- Platform is accessible from anywhere
- Easily serves a team of 6 full stack developers
- Be able to build and deploy using Jenkins and GitHub
- Platform is extensible
- Is very cost effective - monthly spend should remain under $9k

Deliverables:

- A functional platform or solution that meets the above requirements
- Documentation and support for deployment and maintenance of the platform

Challenges

Building the desired MVP using commercial cloud services and AI API providers would require over a million dollars—funding that isn't currently available. The client believes the MVP will secure that funding. Additionally, all input and output data must remain private. The data privacy is the most important requirement.

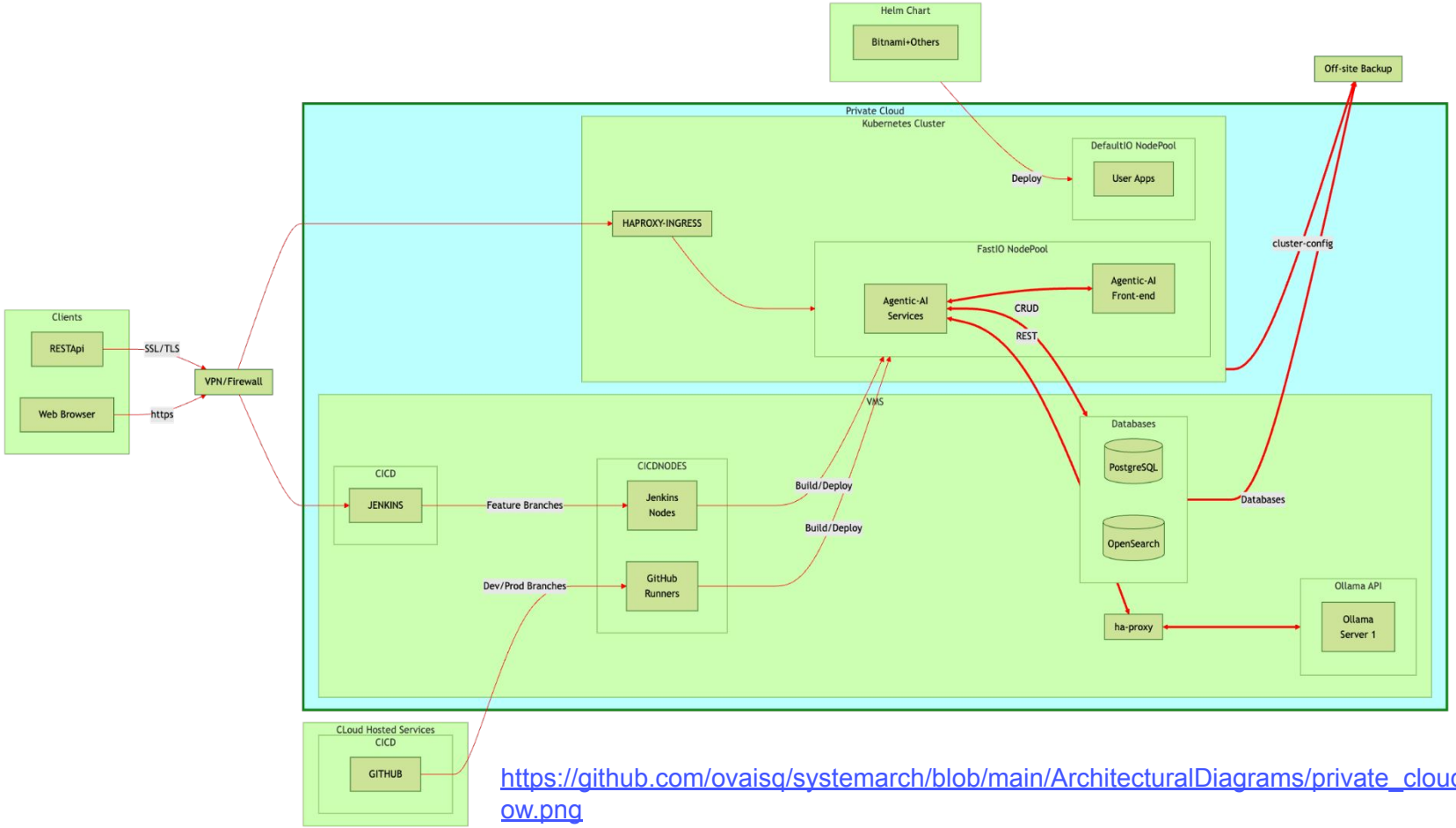
We need to start small, focusing on a setup where we can rapidly build, deploy, and confirm the feature set within the next 6 to 9 months. The solution must also be capable of scaling when necessary.

The team has experimented with various open-source AI/LLM solutions and feels that the quality of the answers, while not as polished as commercially available options, is sufficient for their needs. Moreover, open-source solutions addresses the pressing data privacy needs.

Solution

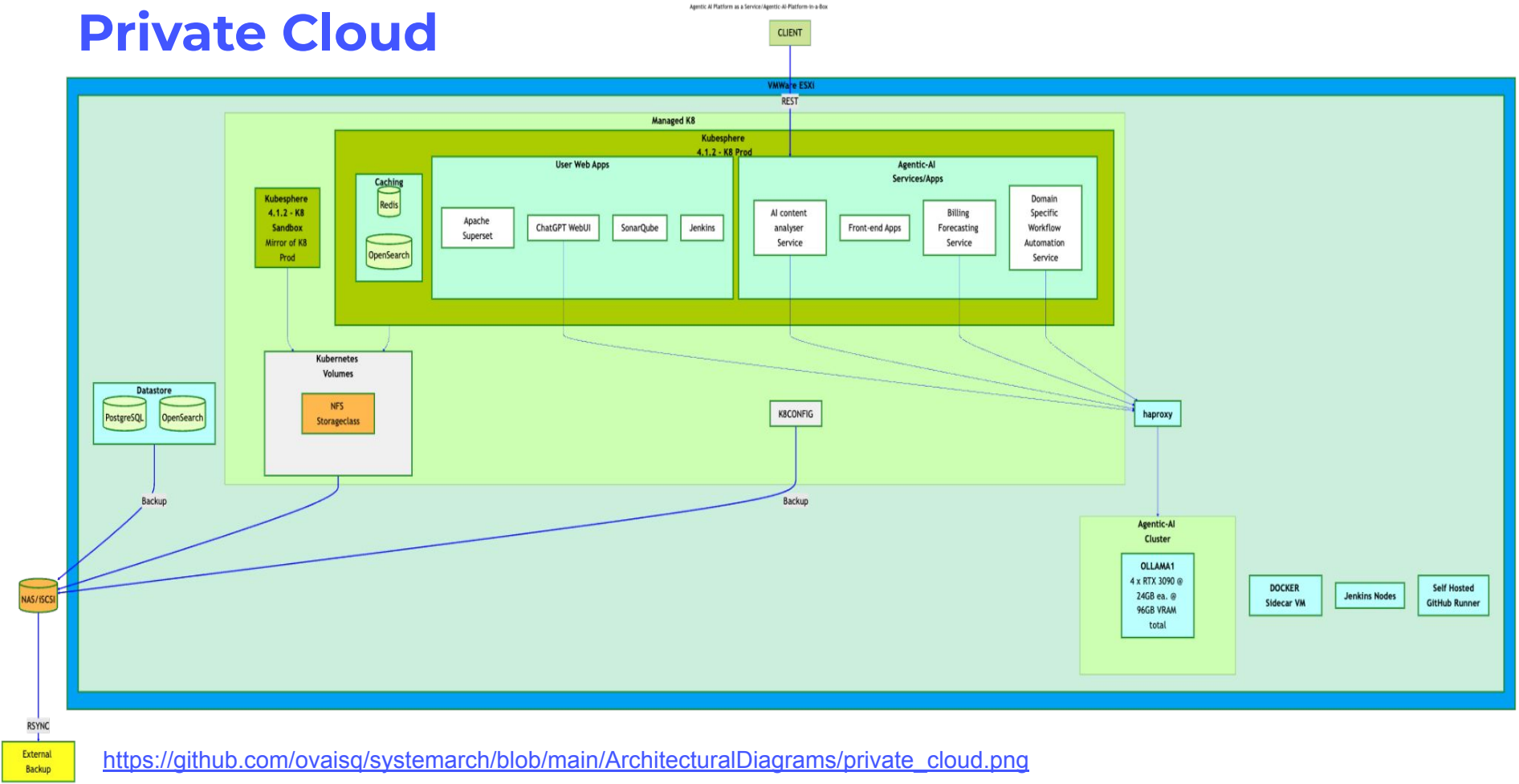
The resulting platform - Private Cloud platform - is a single-server based private cloud that handles high-volume, secure build/deploy activities for developing and testing Agentic-AI applications and services. Built on VMware ESXi, Kubernetes, and other open-source tools, this setup integrates seamlessly with public cloud providers, and provides secure VPN access to all key services. The platform is designed to be extensible. With costs significantly lower than AnthropicAI or OpenAI. The solution achieves AI ambitions without breaking the bank.

Technical Implementation Strategy



https://github.com/ovaisq/systemarch/blob/main/ArchitecturalDiagrams/private_cloud_workflow.png

Private Cloud



https://github.com/ovaisq/systemarch/blob/main/ArchitecturalDiagrams/private_cloud.png

Private Cloud

Physical Server:

- AMD Epyc 7452 32 Cores/64 Threads
- 256GB DDR4 3200MHz ECC RAM
- 2 x 2TB NVMe SSD Storage, 20TB RAID10 (4 x 10TB) NAS Storage
- 4 x NVIDIA RTX 3090 24GB VRAM ea. GPUs - totalling 96GB VRAM

Virtual Server

- VMware ESXi 8U3 Server (Standard - 32 Core License)

Dedicated VMs - all Debian 12 (64 bit):

- DB Server: PostgreSQL Server, OpenSearch Server
- Ollama Server: Runs Gemma 2, Llama 3.1, 3.2, and various open-source LLMs locally
  4 x NVIDIA RTX 3090 24GB VRAM GPUs in Passthrough Mode
- 2 x Kubernetes Clusters (4 Nodes each) managed by open-source multi-cloud management called Kubesphere.
- Misc: haproxy, Local Docker container Registry

Alerting and Monitoring:

- LogicMonitor's Container/K8 monitoring -

Private Cloud

Kubernetes Hosted Apps and Services:
User Apps:

- Jenkins - CICD platform (custom Branch builds; use GitHub for Dev and Prod branches)
- SonarQube - static code quality and secure analysis tool
- Apache Superset - Data visualization platform
- Open WebUI - Web UI for chat-prompting - ChatGPT clone but for locally hosted LLMs
- Redis and Opensearch used for caching content by some of the aforementioned apps
- Dedicated Local Nodes for Jenkins and GitHub

Locally built Agentic-AI based Services:

- AI content analyser Service
- Billing Forecasting Service
- Domain Specific Workflow Automation Service
- Django Based Frontend

Release Strategy

Platform:

- Initial install: Deploy it at a colocation facility. With on-site support staff, confirm outside access
- **Server BIOS/Hardware update**: Notify team of a planned downtime, then apply updates.
- **Update/Upgrades**:
  - K8 major version update: Notify team of the planned downtime, bring down the service or cluster if needed, and apply upgrade. Verify update, bring up service/cluster. Notify team.
  - User App (Helm charts), and OS updates - apply security patches immediately, if downtime required, notify users of planned downtime, apply patch. Notify users after.
- **K8 Services** - incremental roll-out

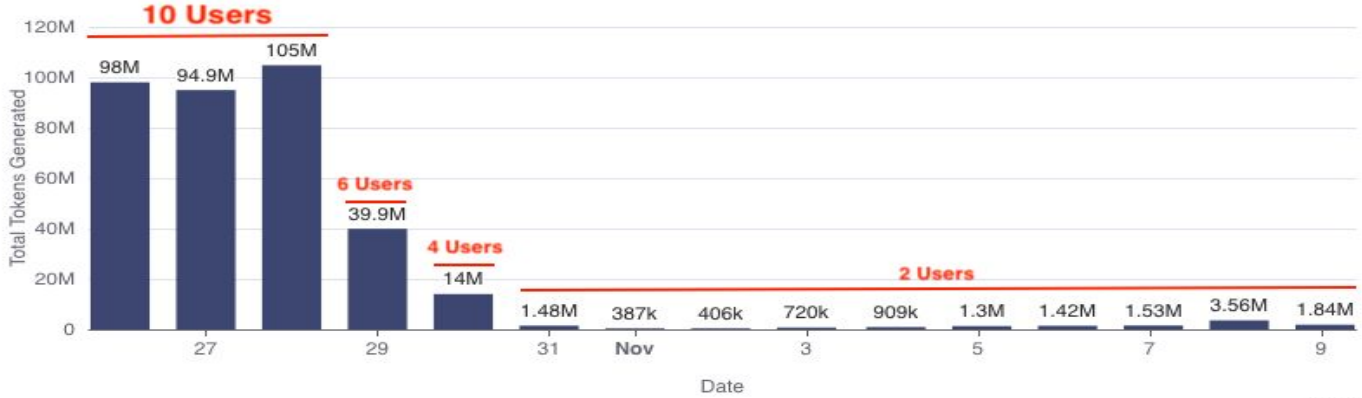
Results/Impact

- Huge cost Savings:
  - 1st year CAPEX @ $26k with subsequent @ $766/mo cost.
  - No rate limiting nor any overages
- All data is stored on hosted platform.
- Dozens of daily builds and deploys
  - Facilitated rapid development
  - Jenkins Feature branch build/deploys are under 1 minute
  - GitHub build/deploys are slightly above 1 min
- Performant
  - Millions of input tokens
  - Tens of millions of tokens output
  - Hundreds of thousands of successful prompts completions
  - Thousands of concurrent requests per minute
  - Not bound by API rate limitation
- Extensible
  - Seamless integration with AWS/GCP/Azure (hybrid-cloud)

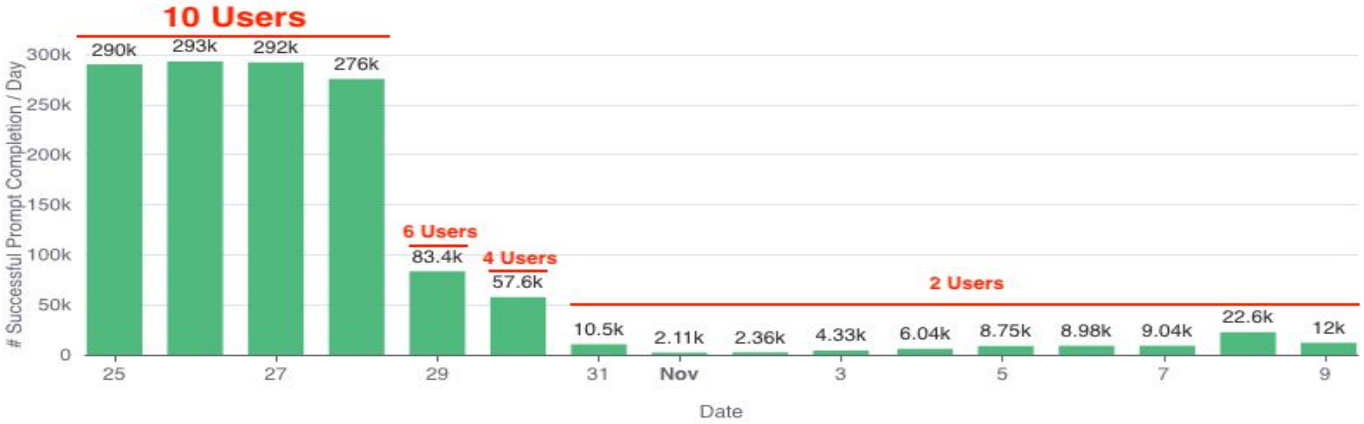
- Multi-cloud management
- Highly scalable
  - Vertical Scaling
    - Server: Add CPU
    - Server: Add GPUs (up to 4 additional)
    - Server: Add RAM
    - Server: Add Disk Space
    - VMs: Add more vCPUs and RAM to K8 Nodes
    - VMs: Add more vCPUs and RAM to VMs
  - Horizontal Scaling (K8)
    - Autoscale K8 Cluster
    - Use Node Pools to distribute I/O intensive workloads
    - HPA - autoscale num pods based on utilization
    - Up Replica count
    - Add more K8 nodes

Total Number of Tokens Generated Per Day

Output in MTok



Successful Prompt Completions Per Day



Server/Colocation Cost

Server:
4U 32C/64T, 256GB Ram, 2 x 2TB NVME Gen4, 4 x RTX 3090 24GVRAM = $17k

Colocation cost:
4U, 3AMPs, 1GBPS = $450/mo @ $5.4k/yr

ESXi Standard License Cost/Yr: $2.6k/yr

LogicMonitor License: $100/mo @$1200/yr

W/ESXi **1st year Initial cost** = $26.2k/yr

W/ESXi **subsequent yearly cost** = $766/mo @ $9.1k/yr

Anthropic API Usage Cost

Claude 3.5 Haiku:
$5 x 105MTok = $525/day @ $15k/mo @ $180k/yr
$5 x 14MTok = $70/day @ $2.1k/mo @ $25.2k/yr

Claude 3.5 Sonnet:
$15 x 105MTok = $1,575/day @ $47k/mo @ $564k/yr
$15 x 14MTok = $210/day @ $6.3k/mo @ $75.6k/yr

Claude 3 Opus:
$75 x 105MTok = $7,875/day @ $236k/mo @ $2.8m/yr
$75 x 14MTok = $1,050/day @ $31.5k/mo @ $378k/yr

Note: Estimates are for MTok Output only. Input MTok, Input token caching costs are extra, and are not reflected in totals.

OpenAI API Usage Cost

gpt-4o:
$10 x 105MTok = $1,050/day @ $31.5k/mo
$10 x 14MTok = $140/day @ $4.2k/mo

Note: Estimates are for MTok Output only. Input MTok, Input token caching costs are extra, and are not reflected in totals.

Private Cloud recoups its investment within a few months for almost all cases, but one.