Revolutionizing AI Deployment with the Nefos AI Power Kube

A Business-Centric Perspective

Executive Summary

The Nefos AI Power Kube empowers organizations to build a scalable, secure, and autonomous AI infrastructure on-premises. By eliminating reliance on third-party cloud providers, businesses gain full control over their AI models and sensitive data, ensuring compliance and operational agility.

Introduction

Artificial Intelligence is reshaping industries, but traditional cloud-dependent solutions introduce security risks, latency, and escalating costs. The Nefos AI Power Kube solves these challenges by delivering a fully managed, on-premises AI cluster optimized for seamless model deployment and operational efficiency.

Key Business Benefits

- **Cost Efficiency** – Reduce long-term cloud expenses and avoid vendor lock-in.
- **Data Sovereignty** – Maintain full control over sensitive data, ensuring compliance.
- **Scalability** – Deploy AI models flexibly across edge environments without performance bottlenecks.
- **Enhanced Security** – Operate AI workloads in a secure, private infrastructure.

Trust & Compliance: Ensuring AI Integrity

With increasing regulatory demands, businesses must balance innovation with strict compliance requirements. Power Kube ensures **data sovereignty** with full **GDPR**

compliance, eliminating reliance on external cloud providers. Advanced **encryption**, **access control policies**, and **containerized AI deployments** protect sensitive data. The system is built on **open-source models**, ensuring transparency in AI workflows, ethical governance, and bias mitigation.

Technical Overview

Power Kube is built on cutting-edge containerized architectures, supporting open-source AI models and seamless integration with business-critical applications. The **Model Context Protocol (MCP)** enables real-time interaction with external data sources, enhancing model intelligence and responsiveness.

Use Cases & Industry Applications

- **Financial Services** – Secure AI-powered fraud detection and risk analysis.
- **Healthcare** – Privacy-compliant patient data analysis and diagnostics.
- **Manufacturing** – AI-driven predictive maintenance for industrial machinery.
- **Retail & E-Commerce** – Personalized customer experiences with real-time AI analytics.

Competitive Advantage

Unlike traditional AI deployments, Power Kube provides **unmatched flexibility**, ensuring businesses tailor AI solutions to their unique operational needs while maintaining full ownership and customization of their models.

Conclusion

With Nefos AI Power Kube, businesses unlock the full potential of AI while safeguarding their data, optimizing costs, and enhancing strategic decision-making. This revolutionary AI cluster is the key to autonomous, future-ready AI deployments.

Hardware

To run a powerful AKS (Azure Kubernetes Service) On-Premises Cluster, the following Mini PC requirements are recommended:

- Processor: A multi-core processor, preferably Intel Core i7 or AMD Ryzen 7, with a base clock speed of at least 3.0 GHz for efficient processing power.
- Memory: A minimum of 32 GB RAM to handle containerized workloads and ensure smooth operation.
- Storage: SSD storage with at least 1 TB capacity for fast read/write speeds and sufficient space for container images, logs, and data storage.
- Network Interface: Gigabit Ethernet connectivity to ensure high-speed communication between nodes in the cluster.
- Graphics Processing Unit (Optional): For AI and machine learning workloads, a dedicated GPU such as NVIDIA RTX or A100 series for accelerated computations.
- Operating System: Linux-based OS, such as Ubuntu Server 22.04 LTS, optimized for Kubernetes environments.
- Power Supply: Reliable and uninterrupted power supply to maintain cluster availability during critical operations.

These specifications ensure optimal performance, scalability, and reliability for deploying and managing an AKS On-Premises Cluster tailored to demanding AI-driven applications.

Multi-Mini PC Cluster for On-Premises AI

This configuration outlines how to build a robust, local cluster for hosting AI models and agents using containerization technologies.

---

1. Hardware Foundation: Mini PC Nodes

- **Minimum Nodes:** 3 to 5 Mini PCs for a high-availability (HA) Kubernetes control plane and worker nodes. This allows for failover and rolling updates. More nodes can be added for increased capacity.
- **Mini PC Specifications (per node):**
    - **CPU:** Modern multi-core processor (e.g., AMD Ryzen 7/9 or Intel Core i7/i9, latest generations if possible).
    - **RAM:** 32GB to 64GB DDR4/5 (more is better for running multiple models/agents).
    - **Storage:**
        - **Boot/OS Drive:** 256GB+ NVMe SSD (for speed).
        - **Data Drive:** 1TB+ NVMe SSD (for container images, persistent volumes, and distributed storage).
    - **Networking:** 1GbE or 2.5GbE Ethernet port (2.5GbE preferred for better storage and inter-node communication).
    - **Optional:** Some Mini PCs support Coral AI TPUs via M.2 or USB for accelerating specific TensorFlow Lite models, if applicable to your workload. Discrete GPUs are rare in this form factor and significantly increase cost/complexity; this guide assumes primarily CPU-based inference or modest iGPU usage.

---

2. Software Stack

- **Operating System:** A lightweight, server-focused Linux distribution on each Mini PC (e.g., Ubuntu Server, Debian, or Fedora Server).
- **Container Orchestration: K3s (Lightweight Kubernetes)** is highly recommended.
    - **Control Plane:** Run on 3 of the Mini PCs for HA.
    - **Worker Nodes:** All nodes (including control plane nodes, by default in K3s) can run workloads.
- **Distributed Storage: Longhorn** (CNCF project) integrated with K3s.
    - It uses the local NVMe data drives on each node to create replicated, distributed block storage for persistent volumes.
    - Supports snapshots, backups, and automated volume replication.
    - Alternatives: Rook (Ceph) for more advanced needs (more complex), or GlusterFS.

---

3. AI Model & Agent Deployment

- **Containerization:** All AI models and agents should be packaged as Docker containers.
- **Serving Runtimes:**
    - **NVIDIA Triton Inference Server** (supports multiple frameworks, CPU/GPU).
    - **Seldon Core** (advanced model serving and orchestration).
    - **BentoML** (for building and deploying AI applications).
    - Custom Python applications using frameworks like FastAPI or Flask, containerized.
- **Resource Management:** Define CPU and memory requests/limits for your Kubernetes deployments to ensure fair resource allocation and prevent resource starvation.

- **GPU Acceleration (if applicable):** If using Mini PCs with capable integrated GPUs (e.g., Intel Iris Xe, AMD Radeon) or eGPUs, configure Kubernetes device plugins to make them available to containers. CPU-based inference will be the primary method for most Mini PC clusters.

---

4. Networking 🌐

- **Cluster Network:** A dedicated, reliable switch connecting all Mini PCs (2.5GbE if possible).
- **Service Exposure & Load Balancing:**
    - **Klipper (default with K3s) or MetalLB:** For exposing services with LoadBalancer type on a bare-metal cluster.
    - **Ingress Controller:** Nginx Ingress or Traefik (comes with K3s) for L7 load balancing, SSL termination, and path-based routing to your AI services.
- **Internal DNS:** CoreDNS (default in K3s) for service discovery within the cluster.

---

5. Storage Architecture

- **Persistent Volumes (PVs):** Provided by **Longhorn**.
    - AI models (if large and not baked into images) can be stored on PVs.
    - Databases or state stores for AI agents (e.g., Vector DBs, Redis for agent memory) will use PVs.
    - Longhorn typically defaults to 3 replicas for data resilience across nodes.
- **Container Image Registry:**
    - **Local Registry:** Deploy a private Docker registry (e.g., Harbor, Docker Registry) within the cluster on a persistent volume to store your custom AI model/agent images. This avoids reliance on external registries and speeds up deployments.

- **Object Storage (Optional):** For unstructured data (e.g., training datasets, large model artifacts), consider deploying MinIO within the cluster on Longhorn volumes, or connect to an external NAS if available.

---

6. Failover Strategy

- **Kubernetes Self-Healing:**
    - **Pods:** If a pod (container instance) crashes, Kubernetes will automatically restart it.
    - **Nodes:** If a Mini PC node fails, Kubernetes will reschedule its pods (that are part of a ReplicaSet, Deployment, or StatefulSet) onto healthy nodes. This requires your applications to be stateless or to use persistent storage correctly.
- **Stateful Applications:** Use Kubernetes **StatefulSets** for applications that require stable network identifiers and persistent storage per instance (e.g., databases). Longhorn PVs will ensure data persistence.
- **Control Plane HA:** K3s with an embedded etcd datastore supports HA with 3 or 5 server nodes. If one server node goes down, the cluster remains operational.
- **Longhorn Data Redundancy:** Longhorn automatically replicates volume data across multiple nodes. If a node with a replica fails, Longhorn will use the remaining replicas and create new ones on available nodes.

---

7. Disaster Recovery (DR) Plan

- **Backups:**
    - **Kubernetes Cluster State:**
        - **K3s:** Regularly back up the K3s state (etcd snapshot if using embedded etcd, or external DB backup). K3s has built-in snapshot capabilities.

        - **Velero:** For backing up Kubernetes cluster resources and persistent volumes to an external location (e.g., a separate NAS, an offsite server, or compatible S3 storage).
    - **Longhorn Volumes:**
        - Utilize Longhorn's built-in snapshot and backup features. Backups can be sent to an external S3-compatible target or an NFS share (e.g., hosted on a separate NAS or another machine outside the primary cluster).
    - **Model Artifacts & Configuration:** Store original model files, agent configurations, and Dockerfiles in a version control system (e.g., Git) and back them up regularly to a separate location.

- **Recovery Steps (High-Level):**

1. **Hardware Replacement:** If Mini PCs are damaged, replace them.
2. **OS & K3s Installation:** Re-install the base OS and K3s on new/repaired nodes.
3. **Cluster Restore:**
    - Restore K3s cluster state from backup.
    - Restore Longhorn volumes from external backups using Velero or Longhorn's backup restore mechanism.
4. **Application Deployment:** Redeploy AI model/agent containers (images should be available from your local or backed-up registry).

- **Documentation:** Maintain a clear, step-by-step DR plan document. Test the DR plan periodically.
- **Off-Cluster Backup Target:** Crucially, ensure backups (Velero, Longhorn, model artifacts) are stored *outside* the Mini PC cluster itself. This could be another server, a robust NAS, or even a secure cloud storage location if your policy allows.

Key Considerations & Best Practices

- **Monitoring:** Implement monitoring and alerting using tools like Prometheus, Grafana, and Alertmanager (can be deployed within the Kubernetes cluster). Monitor node health, resource usage, Longhorn status, and application performance.
- **Security:**
    - Harden the OS on each Mini PC.
    - Use Kubernetes network policies to restrict traffic between pods.
    - Manage secrets using Kubernetes Secrets.
    - Regularly update OS, K3s, and all software components.
- **Scalability:** Start with the minimum number of nodes and scale out by adding more Mini PCs as your workload grows.
- **Power Management:** Ensure a stable power supply, ideally with a UPS (Uninterruptible Power Supply) for the cluster and network switch to handle short power outages gracefully.
- **Cooling:** Ensure Mini PCs have adequate ventilation, especially when clustered together under load.

This configuration provides a solid foundation for an on-premises AI platform with a good balance of performance, resilience, and manageability using cost-effective Mini PC hardware.

NEFOS
INNOVATION
AI