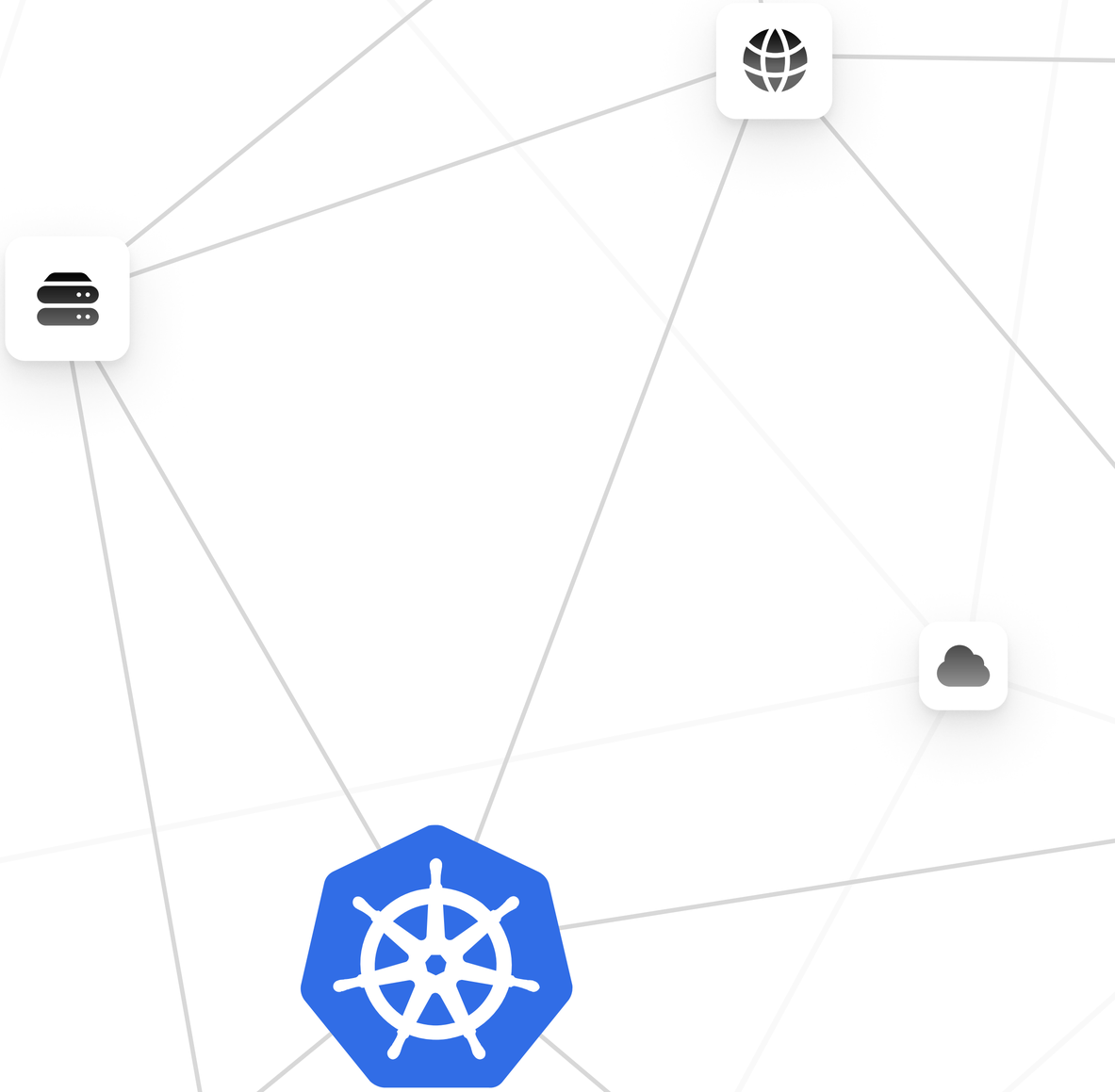
morphius

Case Study

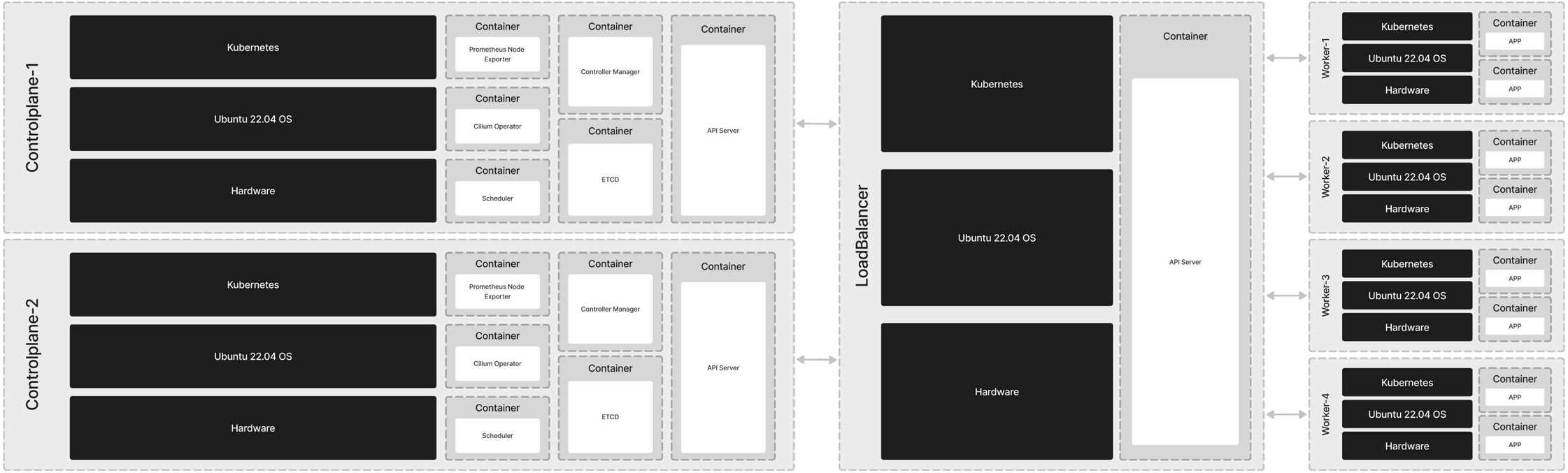
Kubernetes High Availability (Stacked ETCD) on-premises deployment

Involved deploying and managing a high-availability Kubernetes cluster (Stacked ETCD) on-premises for a cloud distributor company.



Jakarta, Indonesia +62 851-3490-4867 morphius.app

Overview



This project involved deploying and managing a high-availability Kubernetes cluster (Stacked ETCD) on-premises for a cloud distributor company. The cluster was deployed on seven virtual machines running on LXD using **kubeadm** and was configured for networking security, persistent storage, and monitoring.

The setup integrated **Cilium** for networking, **Istio** for service mesh, **ArgoCD** for GitOps, **NFS** for storage, **Velero** for backup and migration, and **Prometheus** for alerting and monitoring.

The Challenge

The cloud distributor company was a relatively new business, operating for about a year. Their DevOps team was small, consisting of only two engineers: one responsible for infrastructure and the other for development. They required an infrastructure that was highly available, scalable, and easy to manage while minimizing operational overhead.

They needed an environment where workloads could be efficiently orchestrated, deployed, and managed without excessive manual intervention. Kubernetes became the natural choice for container orchestration, providing flexibility, scalability, and reliability.

Project Objectives

- Deploy a resilient and scalable Kubernetes cluster using **kubeadm**.
- Ensure **high availability** by distributing workloads across multiple virtual machines.
- Implement secure networking using **Cilium**.
- Implement service mesh using **Istio**.
- Implement **GitOps** using **ArgoCD**.
- Enable persistent storage using an **NFS-backed** storage solution.
- Implement a robust backup and disaster recovery solution with **Velero**.
- Provide comprehensive monitoring and alerting with **Prometheus** and **Grafana**.
- Facilitate external access to services securely using **Cilium Gateway API** and **Istio**.

Features

High Availability Kubernetes Cluster: Spread across seven LXD virtual machines to ensure redundancy.

Cilium CNI: Advanced networking security and observability.

Istio: Service mesh for traffic management, security, and observability.

ArgoCD: Continuous deployment for managing Kubernetes applications.

Persistent Storage: NFS-backed PVs for seamless data management.

Gateway API: Facilitates secure traffic management and service discovery.

Velero Backup & Recovery: Enables workload migration and disaster recovery.

Monitoring & Alerting: Prometheus and Grafana provide real-time observability.

Deployment Process

01 Cluster Setup

- Installed kubeadm on six LXD virtual machines (**two control planes and 4 worker nodes**)
- Installed HaProxy as a load balancer in one server. This **load balancer** distributes traffic to all healthy control plane nodes in its target list.

02 Networking Configuration

- Installed and configured **Cilium as the CNI**.
- Implemented network policies for enhanced security.
- Configured **LoadBalancer** and **Gateway API** for external traffic routing.
- Deployed **Istio** for service mesh capabilities and traffic management.

03 Storage Integration

- Set up an **NFS** server to handle persistent storage.
- Configured the **NFS CSI** driver for dynamic volume provisioning.

04 Deployment Automation

- Installed **ArgoCD** for continuous deployment of Kubernetes applications.
- Configured **GitOps** workflows for automated application updates.

05 Backup and Migration

- Deployed **Velero** for backup and restoration of Kubernetes workloads.
- Verified **PVC and PV** restoration to ensure data consistency.

06 Monitoring and alerting

- Deployed **Prometheus** and **Grafana** for cluster monitoring.
- Configured **alerts for node failures**, high resource utilization, and service downtime.
- Exposed the monitoring stack via **Ingress**.

The Result

The implementation of the high-availability Kubernetes cluster provided significant improvements for the company's infrastructure:

- **Improved reliability**: Workloads can continue running even if a node fails.
- **Scalability**: Easily scale applications across multiple nodes without downtime.
- **Cost efficiency**: Running on their own infrastructure reduces reliance on expensive cloud solutions.

- **Portability**: Kubernetes' API enables seamless application migration across namespaces and environments.
- **Faster deployments**: Setting up development and staging environments in separate namespaces is quicker and more efficient.
- **Enhanced security and observability**: Istio and Cilium provide deep network insights and secure service communication.
- **Automated deployments**: ArgoCD ensures that applications remain in their desired state with minimal manual intervention.

morphius

Let’s talk!

+62 851-3490-4867

admin@morphius.cloud