

Building a Kubernetes Cluster with Vagrant & Ansible

This leaflet provides an overview of how to set up a lightweight Kubernetes cluster using K3s, automating the provisioning process with Vagrant and Ansible. The project enables developers to experiment with Kubernetes locally before deploying to a production environment.

Tools Used in the Project

K3s:

A lightweight Kubernetes distribution designed for minimal resource consumption.

Vagrant:

A tool for managing virtual environments to create repeatable development setups.

VirtualBox:

A virtualisation software used to run virtual machines for the Kubernetes cluster.

Ansible:

An automation tool used for configuring and deploying software across multiple servers.

kubectl:

A command-line tool for managing Kubernetes clusters and deployments.

Helm:

A package manager for Kubernetes applications, simplifying deployment and management.

Prometheus:

A monitoring tool used to collect and visualise cluster performance metrics.

Loki:

A log aggregation system used to track and analyse system logs.

How the Components Work Together

Vagrant & VirtualBox:

Vagrant is used to define and provision the virtual machines (VMs) needed for the cluster. These VMs run on VirtualBox, providing an isolated environment for Kubernetes.

K3s:

K3s is installed on the virtual machines to form a Kubernetes cluster. A single node acts as the master, while additional nodes serve as workers.

Ansible:

Ansible automates the installation of K3s across the nodes, ensuring a smooth and repeatable setup

process.

kubectl:

kubectl is used to interact with the Kubernetes cluster, deploy applications, manage resources, and monitor system performance.

Helm:

Helm simplifies the deployment of applications within Kubernetes by managing package dependencies.

Prometheus & Loki:

Prometheus collects and visualises system metrics, while Loki gathers logs to monitor the health and performance of the cluster.

Summary & Benefits

This project demonstrates how a Kubernetes cluster can be set up in a local environment using automation tools.

It provides a controlled environment for learning, experimentation, and testing before moving to a production-grade cluster.

The automation ensures consistency, reducing the risk of misconfigurations and speeding up deployments.

Key Benefits:

- Automated deployment reduces manual setup errors.
- Local virtualisation enables experimentation without cloud costs.
- Ansible ensures reproducibility across multiple environments.
- Monitoring and logging improve visibility into system health.
- Lightweight K3s offers a fast and efficient Kubernetes experience.