Sara Rasool

Metal3 Project

Goals

What drove us to create Metal3?

**Kubernetes Native API**

The Metal³ project aims to provide a Kubernetes-native way of managing bare metal infrastructure, meaning that it integrates tightly with Kubernetes and leverages its API and tooling to manage bare metal hosts. This approach allows the Cube admins to use the same tools to administer the infrastructure underneath the cluster that they used to administer the application that runs on top of it that makes it easier to manage and automate the deployment of Kubernetes clusters on bare metal infrastructure.

**Self-Hosted**

The Metal³ project aims to be self-hosted, meaning that it can run (in the cluster) on top of Kubernetes itself. This allows users to manage their bare metal infrastructure using the same tools and workflows that they use to manage their Kubernetes clusters, further simplifying the management clusters and avoiding the need for an external dependency.

**Self-Managed**

The Metal³ project aims to be self-managed, meaning that its Integration with Machine API from Sigma Cluster- Lifecycle automates the provisioning, configuration, and management of bare metal hosts just like your cloud-based clusters. This automation helps to reduce the manual effort required to manage bare metal infrastructure, making it easier and more efficient to deploy and maintain Kubernetes clusters on bare metal hardware.

Components

**Bare Metal Actuator**

Bare Metal Actuator is a component of the Metal³ project that provides a set of Kubernetes custom resources, command-line tools, and a web-based user interface for managing bare metal hosts. The Actuator interacts with the bare metal operator to perform hardware management tasks such as provisioning, configuring, and deprovisioning bare metal hosts. The Actuator allows users to automate the management of bare metal hosts using standard Kubernetes workflows and tools, providing a powerful set of APIs and tools for managing bare metal infrastructure in a Kubernetes-native way.

**Bare metal operator**

The bare metal operator is responsible for managing the lifecycle of bare metal hosts, including provisioning, configuration, and deprovisioning. It runs as a Kubernetes operator inside the Kubernetes cluster and interacts with the underlying hardware platform to perform these tasks. The operator is responsible for hardware management and is designed to be highly available and scalable.

**Bare metal management pods**

Bare metal management pods typically include an operator that manages a set of pre-configured tools and services for controlling the power on the host, monitoring the host status, and provisioning images to the host. These tools run inside the pod with the operator, and do not require any configuration by the user.