Introduction

Prometheus Operator is a Kubernetes Operator that provides Kubernetes native deployment and management of Prometheus and related monitoring components.

The Prometheus operator includes, but is not limited to, the following features:

- **Kubernetes Custom Resources**: Use Kubernetes custom resources to deploy and manage Prometheus, Alertmanager, and related components.
- **Simplified Deployment Configuration**: Configure the fundamentals of Prometheus like versions, persistence, retention policies, and replicas from a native Kubernetes resource.
- **Prometheus Target Configuration**: Automatically generate monitoring target configurations based on familiar Kubernetes label queries; no need to learn a Prometheus specific configuration language.

Prometheus Operator provides a set of Custom Resource Definitions(CRDs) that allows you to configure your Prometheus and related instances. Currently, the CRDs provided by Prometheus Operator are:

- Prometheus
- Alertmanager
- ThanosRuler
- ServiceMonitor
- PodMonitor
- Probe
- PrometheusRule
- AlertmanagerConfig
- PrometheusAgent
- ScrapeConfig

> Check the Design page for an overview of all the resources provided by Prometheus Operator.

Goals

- To significantly reduce the effort required to configure, implement and manage all components of Prometheus based monitoring stack.
- **Automation** - Automate the management of Prometheus monitoring targets, ultimately increasing efficiency. This automation is performed by the use of Kubernetes Custom Resource Definition. The Operator introduces custom resources like `Prometheus`, `Alertmanager`, `ThanosRuler`, and others, which help automate the deployment and configuration of these resources.
- **Configuration Abstraction and Validation** - Instead of learning and manually writing Prometheus Relabeling rules (which can be time consuming), you can simply use Kubernetes Label Selectors. `ServiceMonitor`, `PodMonitor` and `Probe` custom resources provide this abstraction. The Operator also removes the complexity of validating the configuration of `AlertmanagerConfig` and `PrometheusRule` objects.
- **Scaling** - There are many scaling-related features provided by the Operator like ThanosRuler custom resource for rule evaluation, workload distribution across multiple Prometheus instances using scrape target sharding, and running Thanos sidecar in Prometheus instance for long-term storage.

Next Steps

By now, you have the basic idea about Prometheus Operator!!

Take a look at these guides to get into action with Prometheus Operator.

Getting-Started

Get started with Prometheus-Operator. →

API Reference

Reference for different fields of Custom Resources in Prometheus-Operator. →

Platform Guide

Set up, configure and manage instances of Prometheus-Operator, Prometheus, Alertmanager and ThanosRuler resources. →

Developer Guide

→

Learn how to configure scraping, alerting, and recording rules for your applications.

Community

Join and interact with Prometheus-Operator community.

Installing Prometheus Operator →