

## Using Prometheus with KumoScale in Managed Mode

This section provides an overview to how to set up the Prometheus custom resource file to configure and create components used in data collection and display.

The Prometheus Stack includes the following:

- **Prometheus Operator** provides Kubernetes native deployment and management of Prometheus and related monitoring components.
- **Prometheus Service** provides a web user interface (UI) that enables you to view simple graphs, Prometheus configurations and rules, and the state of the monitoring endpoints.
- **Alert Manager** handles alerts sent by client applications such as the Prometheus server.
- **Prometheus Node Exporter** reads system-level statistics about bare-metal nodes or virtual machines and exports them to Prometheus.
- **Kube State Metrics** is a simple service that listens to the Kubernetes API server and generates metrics about the state of the objects. It focuses on the health of the various objects, such as deployments, nodes and pods.
- **Grafana** supports querying Prometheus. It is an interactive visualization web application that provides charts, graphs, and alerts. (The initial Grafana credentials are: admin/ksAdmin and you will need to log into the UI after setting up the stack.)

To configure the Prometheus stack, KumoScale software provides a **Prometheus Stack CRD**, a template of which is in `kumoscale.kioxia.com_v1_prometheusstack_cr.yaml`.

1. Make a copy of `kioxia.com_v1_prometheusstack_cr.yaml` for customizing, (e.g., `myapp_prometheusstack_cr.yaml`)
2. Configure the Prometheus stack by specifying the appropriate parameter values in yaml. See [Configuring the Prometheus Stack in the KumoScale User Guide](#) for details on how to configure the CRD.
3. After specifying your custom values, install the Prometheus Stack with **kubect**`l create`. Using our example file `yaml`:

```
kubectl create -f myapp_prometheusstack_cr.yaml
```

NOTE: If Prometheus fails to install, with the error "*Prometheus: Unable to create mmap-ed active query log*" or similar, try the following:

1. Edit the storage class used by Prometheus (**provisioner-storage-class.yaml**)

2. Explicitly set **fstype** by uncommenting the line:

```
csi.storage.k8s.io/fstype: "ext4"
```

3. Delete and then recreate the storage class with the new `yaml`

To uninstall Prometheus use **kubect**`l delete` as in:

```
kubectl delete -f myapp_prometheusstack_cr.yaml
```

To update Prometheus with new values, use **kubect**`l apply` as in:

```
kubectl apply -f myapp_prometheusstack_cr.yaml
```

