**Monitor your Azure Kubernetes Cluster (AKS) with Prometheus and Grafana**

We will go through the Open Source options available for monitoring Azure Kubernetes Service (AKS) clusters.

We will have a closer look at the Open Source tool kit widely used by the Kubernetes Community: Prometheus, and how it can be used and deployed on the clusters

The following will be covered:

* Deploying and configuring the Prometheus stack using Helm Charts and the Prometheus Operator

*Pre-requisites:*

1. Azure subscription
2. Azure CLI
3. Basic knowledge of Kubernetes commands and components
4. An AKS cluster provisioned and is in Running state

* az aks get-credentials -g {resource-group} -n {aks-cluster-name}

1. Install Helm for your respective OS - <https://github.com/helm/helm/releases>

Prometheus and Grafana are open source monitoring tools for Kubernetes clusters. It allows users and developers to capture metrics and telemetry data for applications running inside the cluster, allowing deeper insights into application performance and reliability at the container level.

**Introduction to Prometheus**

Prometheus is an open-source system monitoring and alerting toolkit originally built at SoundCloud.

Prometheus has been adopted since its inception, by many companies and organizations, and the project has a very active developer and user community. It is now a standalone open source project and maintained independently of any company.

Prometheus joined the Cloud Native Computing Foundation in 2016 as the second hosted project, after Kubernetes.

Prometheus provides a set of applications which collect monitoring data from your applications, containers, and nodes by scraping a specific endpoint. These services collate this data and provide a way to query and alert on the data.

In addition to the applications to collect monitoring data, Prometheus provides an easy way for you to expose metrics from your own applications using a set of client libraries and a standardized format for presenting the metrics. Prometheus does not Provide a dashboard, it relies on Grafana for visualizing data.

**Prometheus Components**

**Prometheus Operator**

In late 2016, CoreOS introduced the Operator pattern and released the Prometheus Operator as a working example of the pattern.

The Prometheus Operator is simple to install with a single command line and enables the configuration and management of Prometheus instances using simple declarative configuration that will, in response, create, configure, and manage Prometheus monitoring instances.

The Operator deploys the following components into the cluster:

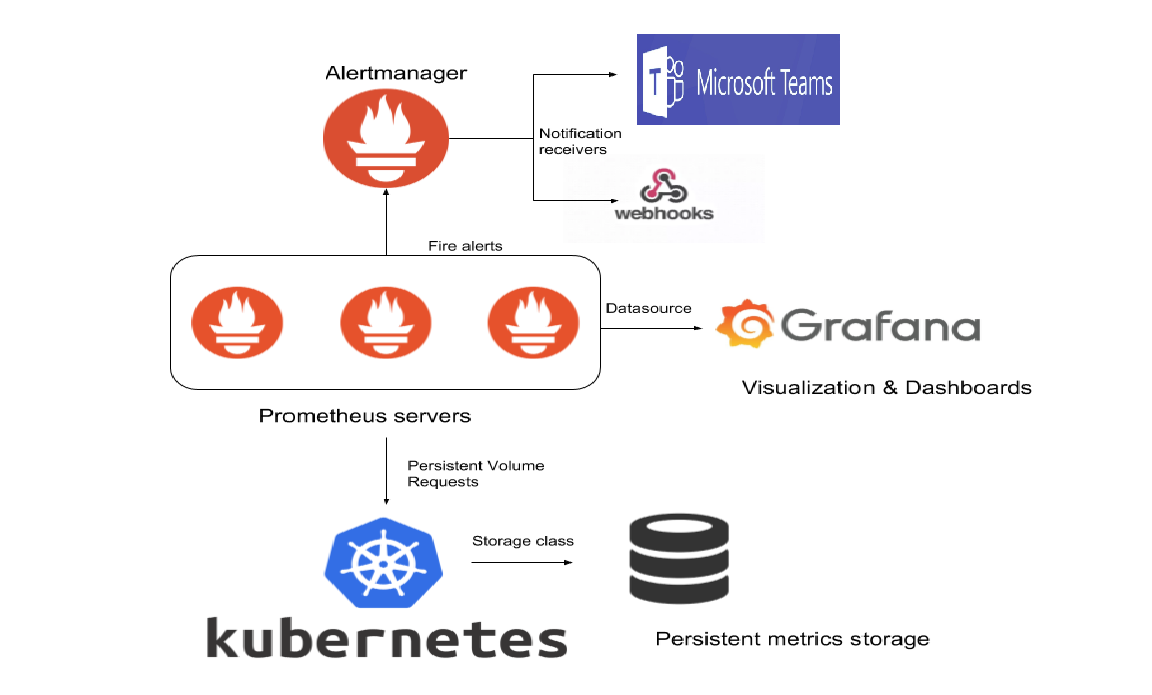
* Prometheus, which defines a desired Prometheus deployment. The Operator always ensures that a deployment matching the resource definition is running.
* Service Monitor, which declaratively specifies how groups of services should be monitored. The Operator automatically generates Prometheus scrape configuration based on the definition.
* Alertmanager, which defines a desired Alertmanager deployment. The Operator always ensures that a deployment matching the resource definition is running. It is used to aggregate alerts and send them to customly configured receivers.

These receivers can be: Webhooks for systems like Slack, Microsoft teams, SMTP emailing system.

**Grafana**

Grafana is used to visualize any data. It can be used to create multiple dashboards that support a variety of beautiful graphs and visualizations for different users and stakeholders. It has support for the integration different type of data sources like Prometheus, Azure Monitor, Elasticsearch, AWS Cloud Watch etc.

Grafana, it is used to visualize the metrics collected by Prometheus and exposes them on prebuilt Dashboards.



**Deployment**

Helm, the Package Manager for Kubernetes.

Helm is a tool for managing Charts. Charts are packages of pre-configured Kubernetes resources.

We will deploy Prometheus using the stable helm chart for [Prometheus Operator](https://github.com/helm/charts/tree/master/stable/prometheus-operator)

**Connect and setup HELM**

Connect to your cluster by running: az login

**List your subscriptions by running:**

az account list

**Select the subscription your AKS cluster is in by running:**

az account set --subscription <subscription-id>

**Get the credentials required for kubectl to connect to your AKS cluster by running:**

az aks get-credentials --name <aks-cluster-name> --resource-group <aks-cluster-resource-group>

If you do not have HELM installed locally, download the appropriate binaries from this GitHub repository <https://github.com/helm/helm/releases>

If you do not have HELM installed yet in your cluster run: helm init

The scripts needed to follow along can be found in [GitHub Repo](https://github.com/souravjha20/aks-prometheus)

**Clone the GitHub Repo using following command**

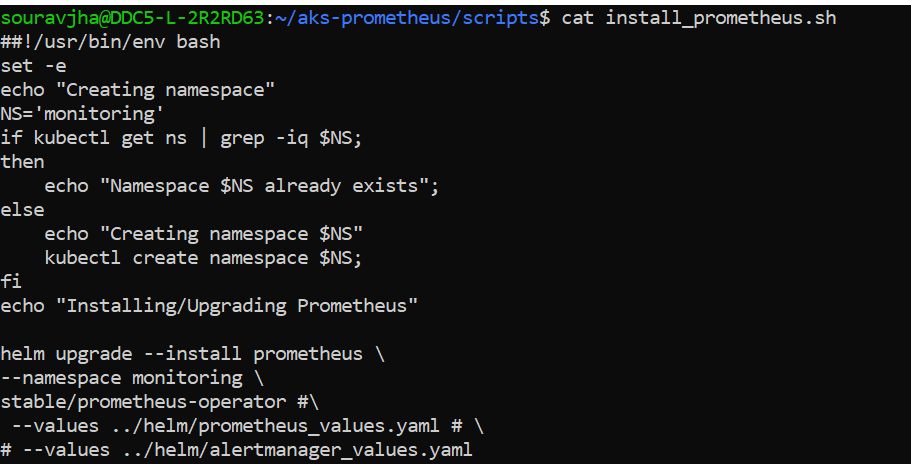
git clone <https://github.com:souravjha20/aks-prometheus.git>

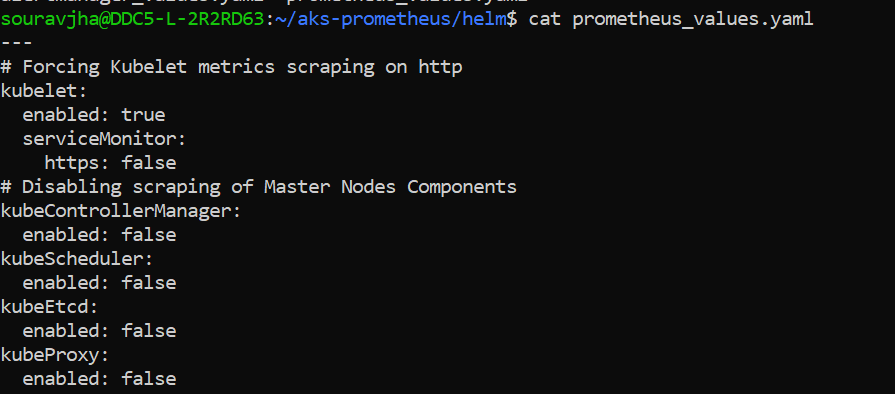
Then run the script,

cd scripts

./install\_prometheus.sh

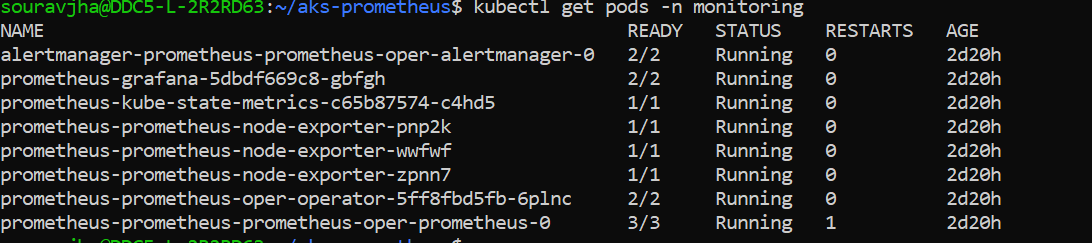






The script performs the following:

* Creates the namespace “monitoring” if it does not exist
* Installs Prometheus and all its components within the monitoring namespace
* If a Helm release named prometheus exists already in the monitoring namespace, Helm will upgrade it
* Please note the use of \*\_values.yaml files after the –values parameter
* These values files help override the Helm Chart templates with values specific to the configuration



**AKS specific configuration**

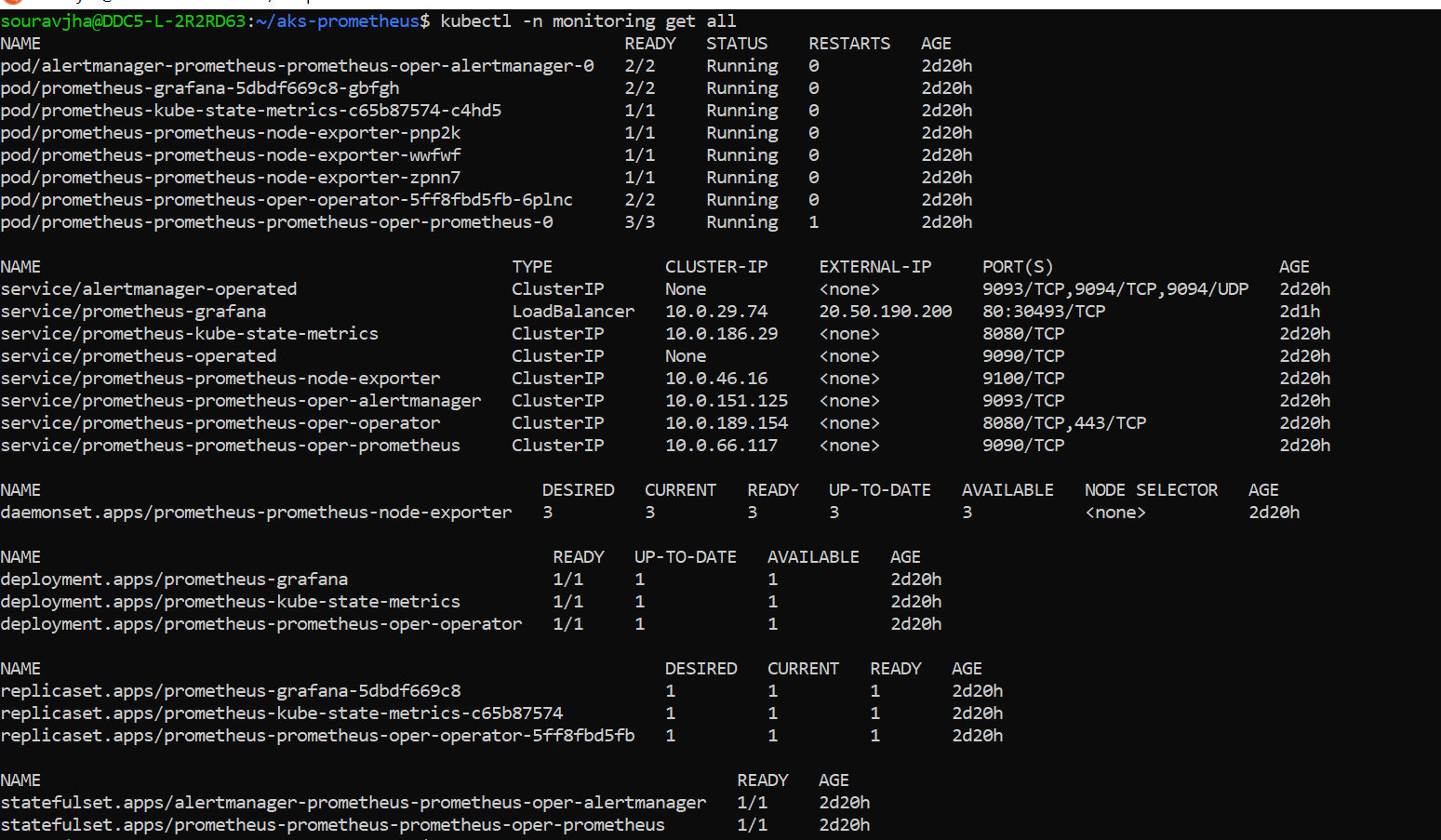
Microsoft AKS being a managed Cloud service, Access to some of the components of the Cluster is not possible (Kubernetes Control Plane), mainly the following components running on the Master nodes:

* kube Controller Manager
* kube Scheduler
* kube Etcd
* kube Proxy

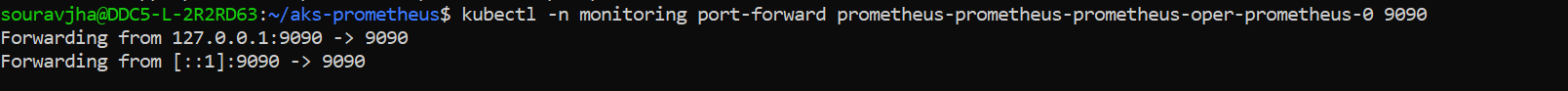
In addition to this, Kubelet, which is running on the Worker nodes is exposing its metrics on http, whereas Prometheus is configured to scrape its metrics on https.

Prometheus deploys the whole shebang with one line of code:

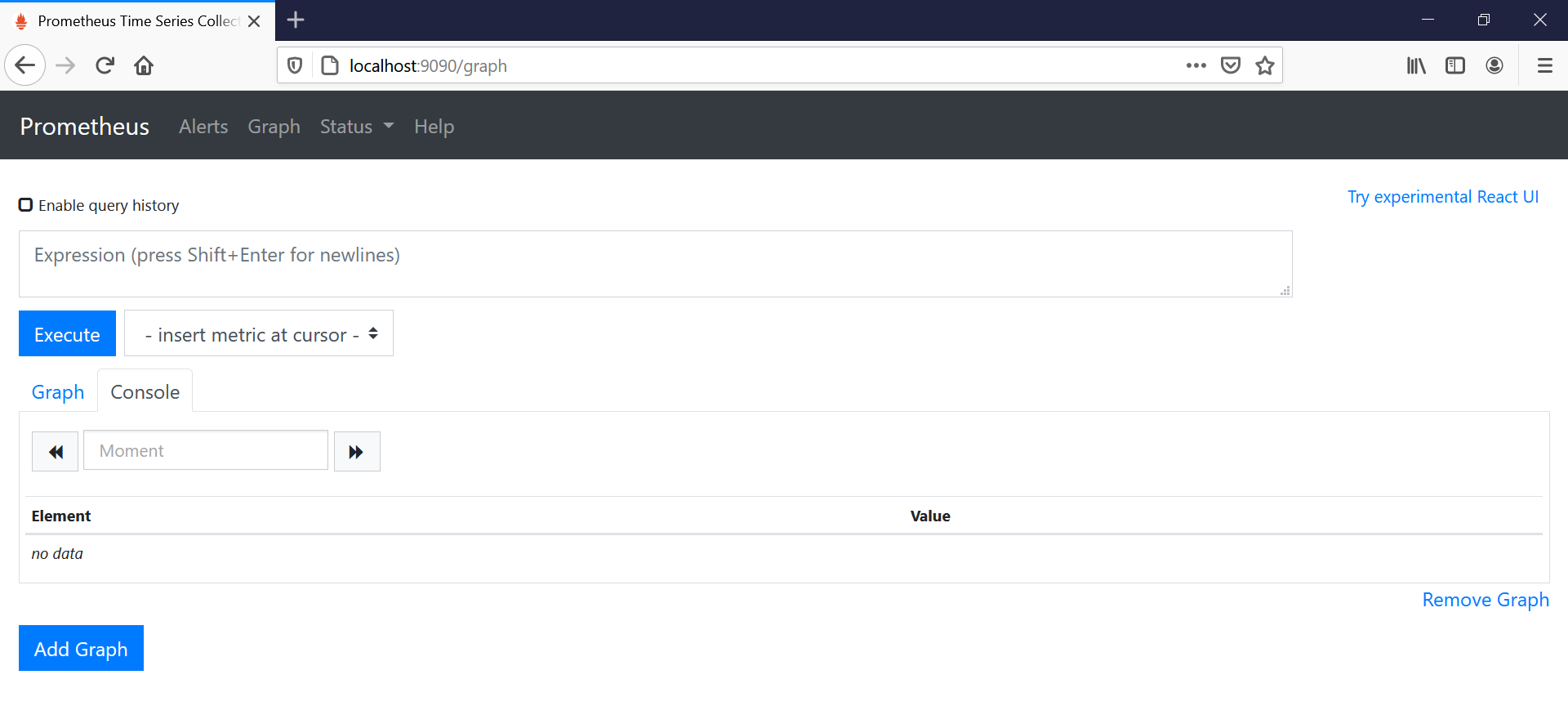
* Deployments
* Services
* Statefulsets
* Replicasets
* Pods



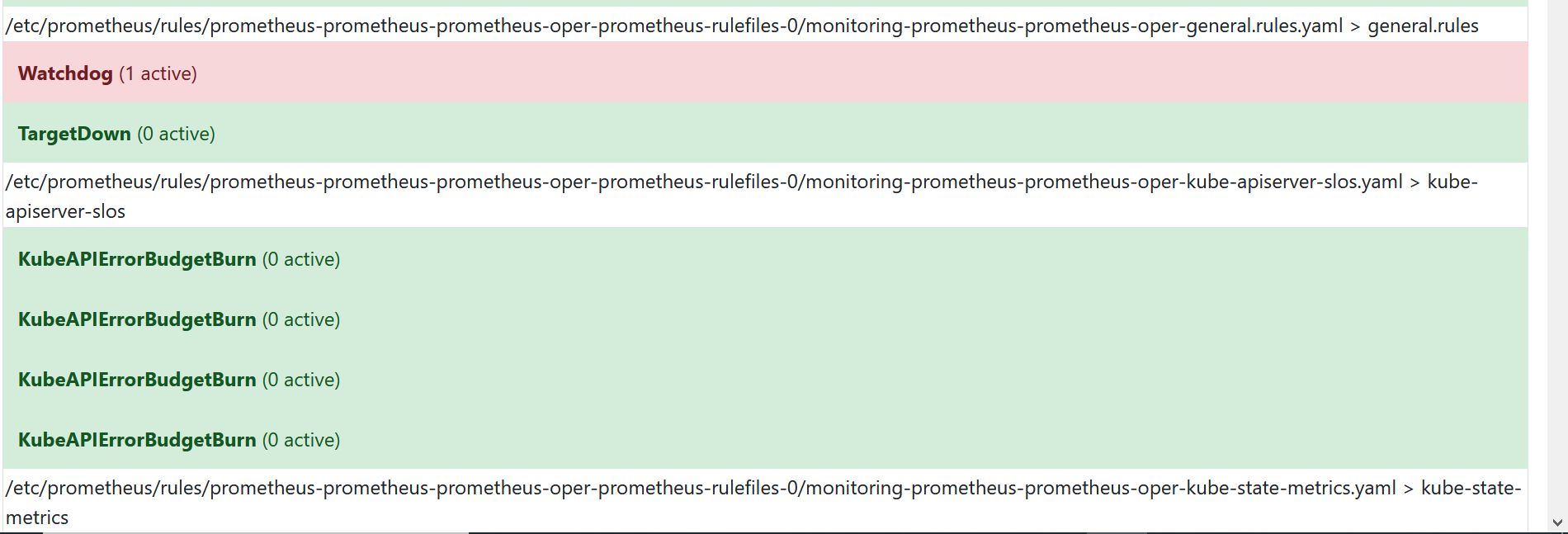
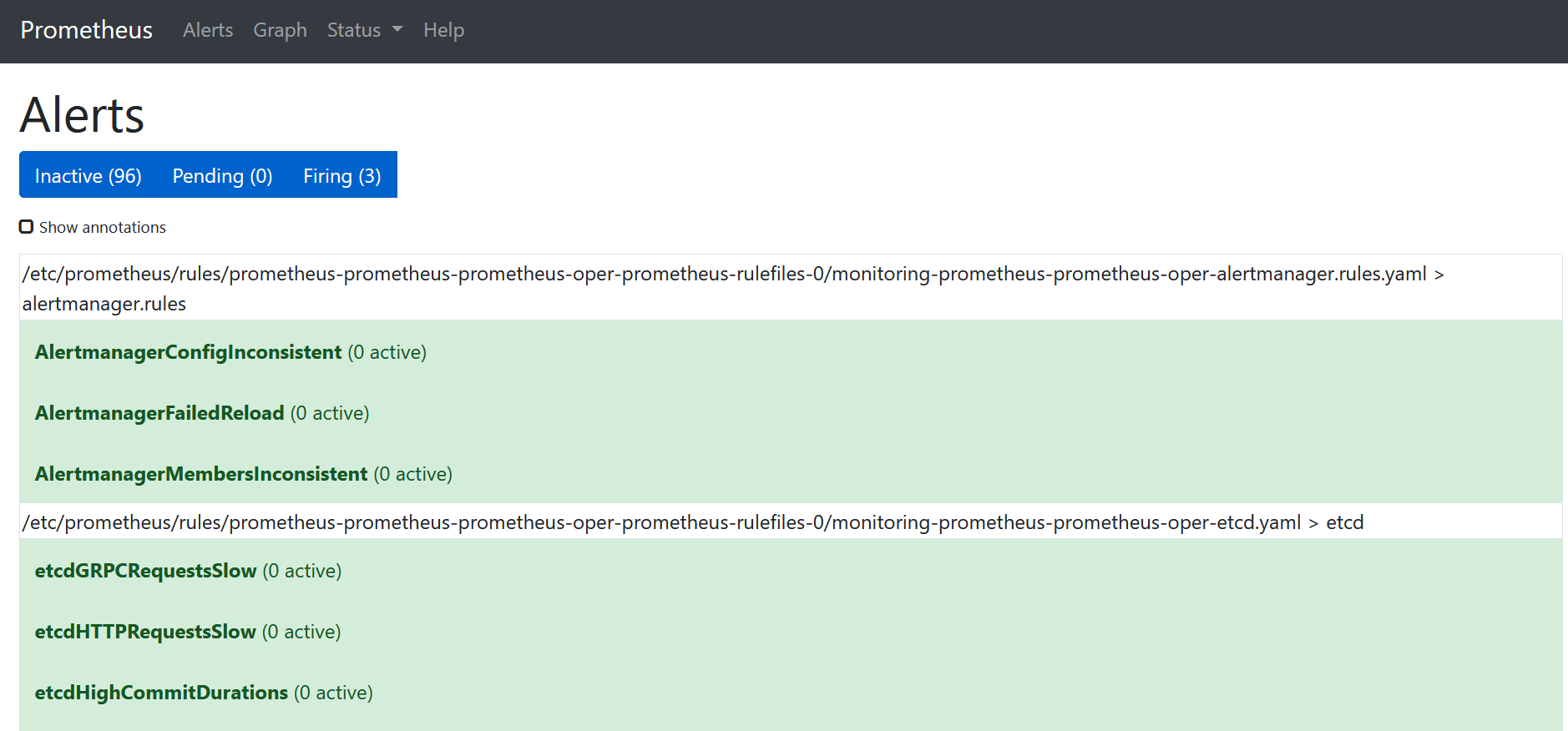
To check the Prometheus page, we have to execute Kubectl with port-forward



Then navigate to http://localhost:9090 and select Alerts

Here we go, all alerts are green except the watchdog alert -which is normal- and the kublet metrics are being scraped properly on http

Watchdog, this is an alert meant to ensure that the entire alerting pipeline is functional. This alert is always firing therefore, it should always be firing in Alertmanager and always fire against a receiver. There are integrations with various notification mechanisms that send a notification when this alert is not firing.



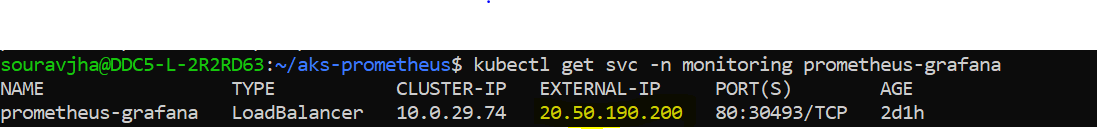
**Grafana**

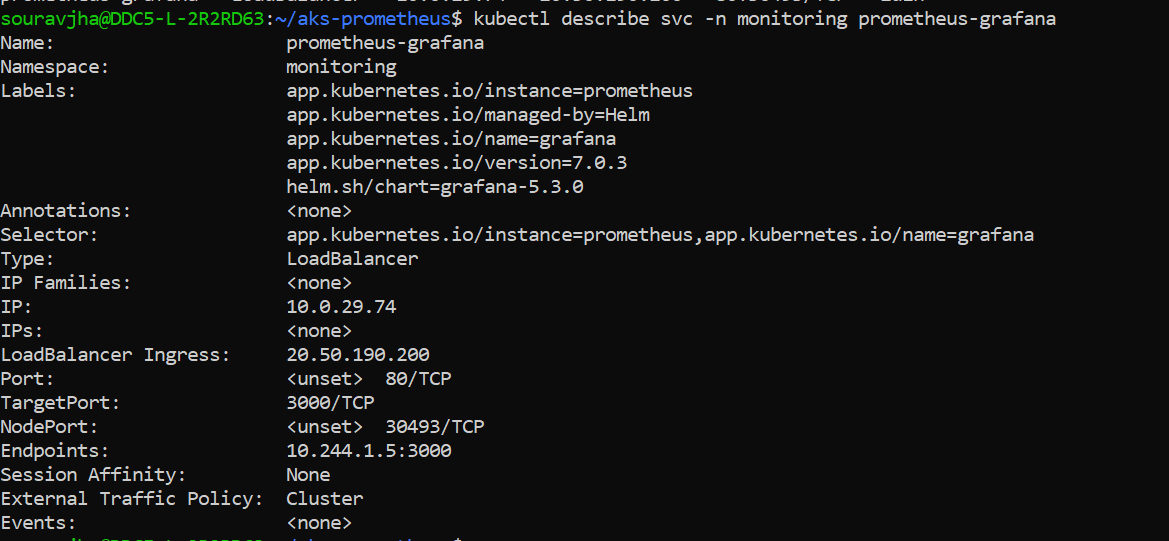
Grafana Dashboard can be accessed by performing port-forward on the pod:



Then navigate to http://localhost:3000 to access Grafana Web Site - admin | password: prom-operator

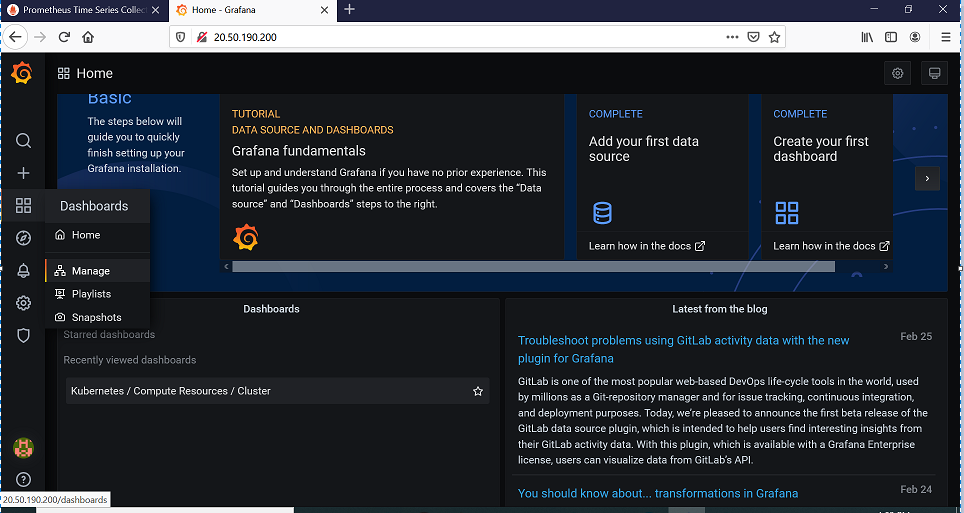


We can also create an external endpoint to connect the Grafana using Load Balancer Service 

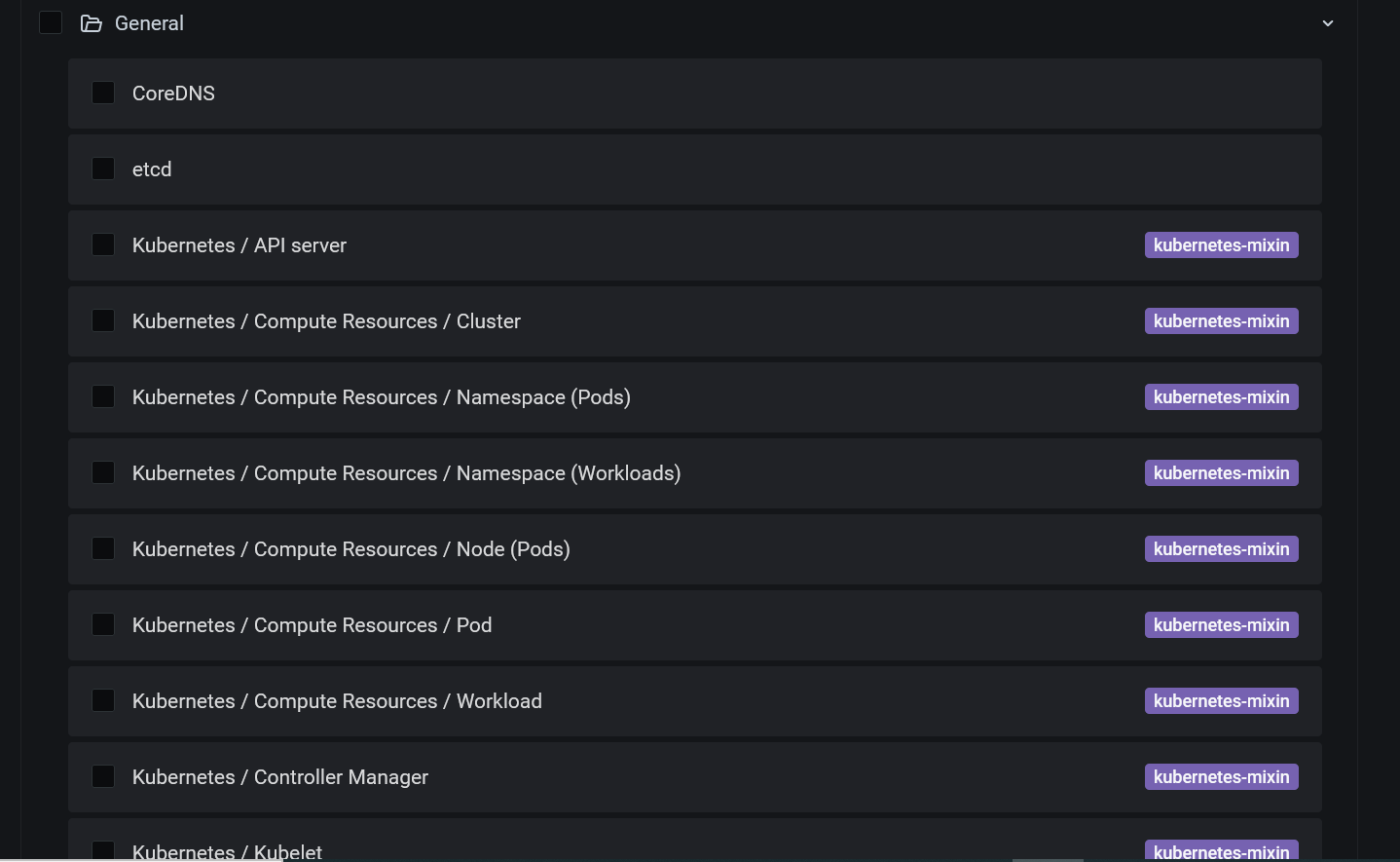


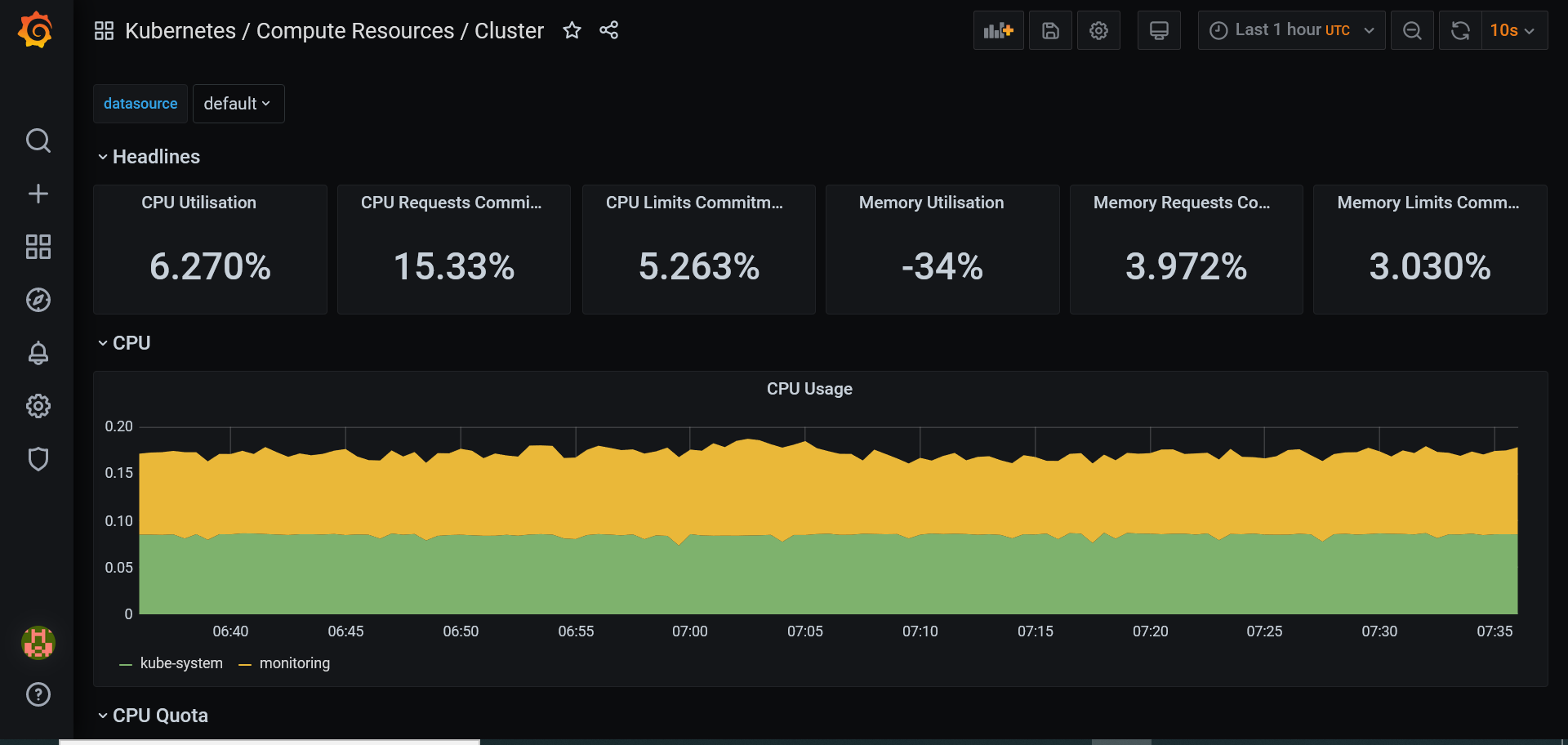
Navigate to external endpoint <http://20.50.190.200> to connect with Grafana with default username and password.

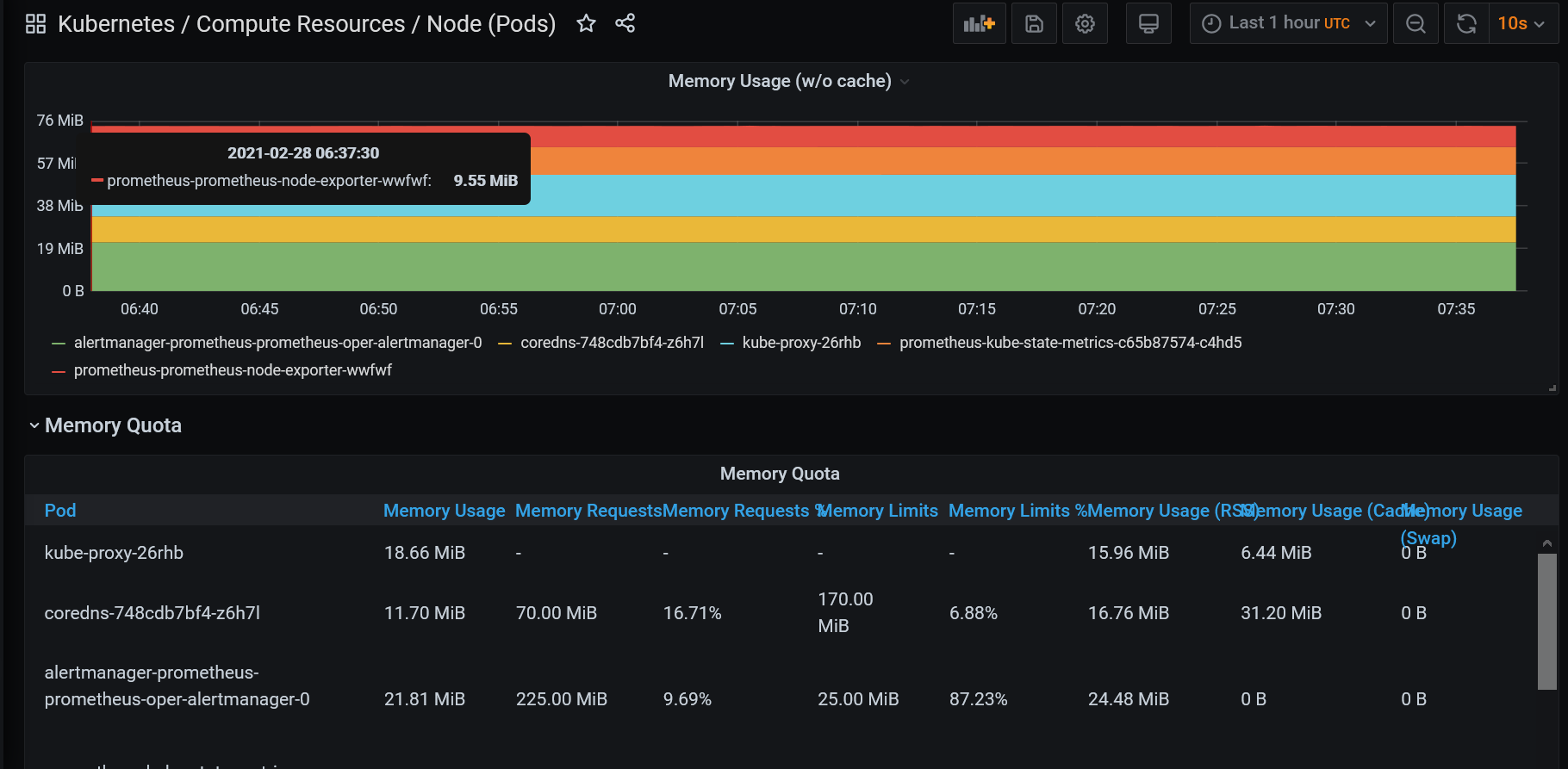
Click on Dashboards, and manage further



**List of Dashboard will fetch up to view the metrics as per cluster**

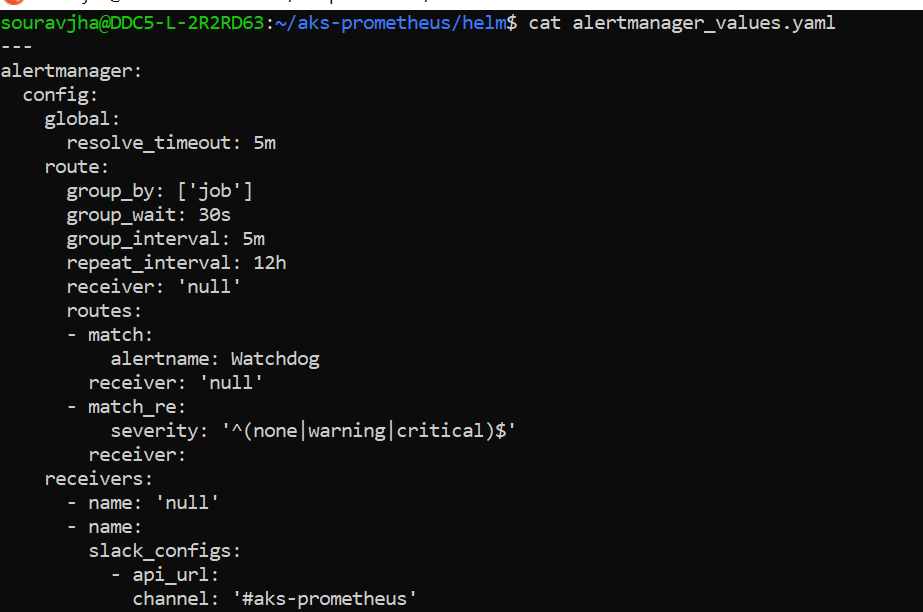






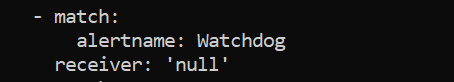
We are all set with Prometheus and Grafana on Azure Kubernetes Cluster (AKS).

To configure the Alerts and Receivers for Alertmanager, helm alertmanager\_values.yaml file is created as below:

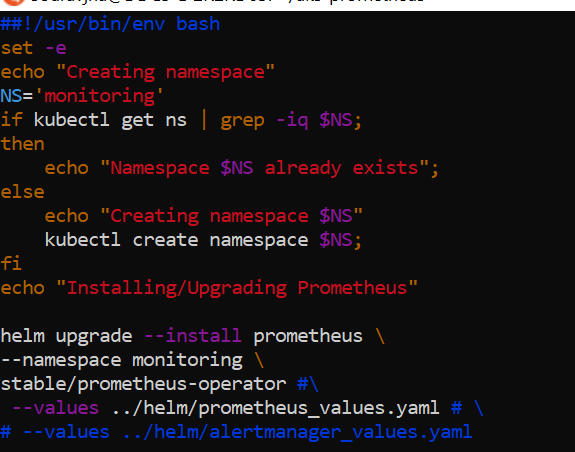


In this, a receiver have to be configured like Microsoft team to use a Team webhook in order to send alerts to a Microsoft Team channel.

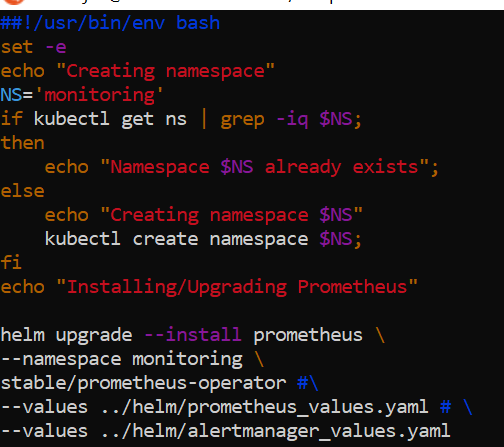
Watchdog Alerts are configured to be sent to the receiver which is a good way to test that the alert process is working and that the webhook is working as well.



Further uncomment the command in install



Execute it as follows to update Helm with the new values.





**Conclusion**

We can discover a lot of other dashboards that gives us valuable insights about the health of AKS cluster, resource usage patterns of application pods, network traffic flow across the cluster, and much more.

Prometheus and Grafana are powerful tools for monitoring your Kubernetes cluster, and with Helm it’s so easy to setup and running in few minutes.