How-To: Install Prometheus

Description

Here we showcase installing Prometheus, https://prometheus.io/docs/introduction/overview/, in Minikube, https://kubernetes.io/docs/setup/learning-environment/minikube/.

Environment Variables

```
declare -x APP_NAME=p8s
declare -x CONTEXT=minikube
declare -x NAMESPACE=prometheus
```

Minikube

Installation and Launch

```
brew install minikube
minikube version
```

> ```
> minikube version: v1.5.2
> commit: 792dbf92a1de583fcee76f8791cff12e0c9440ad
> ```

```
minikube start
```

Explore Default Set-Up

```
kubectl --context="${CONTEXT:-minikube}" get --all-namespaces namespaces,all,configmaps,secrets
```

Prometheus

Installation

- https://github.com/helm/charts/tree/master/stable/prometheus-operator, ef7618975bcfb0557835bb44e77eb2431e2e56e4
- https://github.com/coreos/prometheus-operator/tree/master/example/prometheus-operator-crd/, 53cee6376f5f7c45a2ff8a15a60dd342eb662ddc

```
helm repo add stable https://kubernetes-charts.storage.googleapis.com/
helm repo update

helm search repo prometheus-operator

helm template stable/prometheus-operator | grep --extended-regex ^kind: | nl -s'. '
helm template stable/prometheus-operator | grep --extended-regex ^kind: | sort --unique | nl -s'. '

helm template "$APP_NAME" stable/prometheus-operator --namespace="$NAMESPACE" --set=kubeEtcd.enabled=false --
set=kubeProxy.enabled=false --set=prometheusOperator.createCustomResource=false --set="prometheusOperator.
namespaces.additional=default\,kube-system\,$NAMESPACE" > "${APP_NAME}.yaml"
```

> ```
> # For managed clusters like AliCloud, you may need to disable scraping kubeControllerManager
> /kubelet/kubeScheduler in addition to kubeEtcd/kubeProxy, so use the following instead of the
> above:
> helm template "$APP_NAME" stable/prometheus-operator --namespace="$NAMESPACE" --
> set=kubeControllerManager.enabled=false --set=kubeEtcd.enabled=false --set=kubelet.enabled=false
> --set=kubeProxy.enabled=false --set=kubeScheduler.enabled=false --set=prometheusOperator.
> createCustomResource=false --set= --set=prometheusOperator.kubeletService.enabled=false >
> "${APP_NAME}.yaml"
> ```

```
for crd in alertmanagers podmonitors prometheuses prometheusrules servicemonitors; do
  kubectl --context="${CONTEXT:-minikube}" apply -f https://raw.githubusercontent.com/coreos/prometheus-operator
/master/example/prometheus-operator-crd/monitoring.coreos.com_$crd.yaml
done

kubectl --context="${CONTEXT:-minikube}" create namespace "$NAMESPACE"
kubectl --context="${CONTEXT:-minikube}" apply -f "${APP_NAME}.yaml"
```

Explore Default Set-Up

```
kubectl --context="${CONTEXT:-minikube}" get --all-namespaces namespaces,all,configmaps,secrets | grep -F "$NAMESPACE"
```

Launch

```
kubectl --context="${CONTEXT:-minikube}" get --namespace="$NAMESPACE" svc
kubectl --context="${CONTEXT:-minikube}" port-forward --namespace="$NAMESPACE" "svc/${APP_NAME}-prometheus-operator-prometheus" 9090

open http://localhost:9090/targets
```

Grafana

Credentials

```
kubectl --context="${CONTEXT:-minikube}" get --namespace="$NAMESPACE" "secret/${APP_NAME}-grafana" --output=json | jq --raw-output '.data["admin-user"]' | base64 --decode; echo
kubectl --context="${CONTEXT:-minikube}" get --namespace="$NAMESPACE" "secret/${APP_NAME}-grafana" --output=json | jq --raw-output '.data["admin-password"]' | base64 --decode; echo
```

Launch

```
kubectl --context="${CONTEXT:-minikube}" get --namespace="$NAMESPACE" svc
kubectl --context="${CONTEXT:-minikube}" port-forward --namespace="$NAMESPACE" "svc/${APP_NAME}-grafana" 8080:80

open http://localhost:8080/
```

Uninstallation

```
kubectl --context="${CONTEXT:-minikube}" delete -f "${APP_NAME}.yaml"
kubectl --context="${CONTEXT:-minikube}" delete namespace "$NAMESPACE"

for crd in alertmanagers podmonitors prometheuses prometheusrules servicemonitors; do
  kubectl --context="${CONTEXT:-minikube}" delete -f https://raw.githubusercontent.com/coreos/prometheus-operator/master/example/prometheus-operator-crd/monitoring.coreos.com_$crd.yaml
done
```