Install Prometheus Operator

This is the helper that will extend Kubernetes API, and help us to deploy monitoring.

Prometheus Operator

We are going to download the original Prometheus Operator from git, and do just one change. We are going to change the namespace in its configs.

```
cd
git clone https://github.com/prometheus-operator/prometheus-operator.git
cd prometheus-operator/
# Check current setting for namespaces in bundle.yaml
grep namespace: bundle.yaml
  namespace: default
  namespace: default
  namespace: default
  namespace: default

#We will change that to monitoring:
sed -i 's/namespace: default/namespace: monitoring/g' bundle.yaml

#Check again:
grep namespace: bundle.yaml

  namespace: monitoring
  namespace: monitoring
  namespace: monitoring
  namespace: monitoring
```

Now apply `bundle.yaml` to your Kubernetes cluster:

```
root@control01:/home/ubuntu/prometheus-operator# kubectl apply -n monitoring -f bundle.yaml

customresourcedefinition.apiextensions.k8s.io/alertmanagerconfigs.monitoring.coreos.com created
customresourcedefinition.apiextensions.k8s.io/alertmanagers.monitoring.coreos.com created
customresourcedefinition.apiextensions.k8s.io/podmonitors.monitoring.coreos.com created
customresourcedefinition.apiextensions.k8s.io/probes.monitoring.coreos.com created
customresourcedefinition.apiextensions.k8s.io/prometheuses.monitoring.coreos.com created
customresourcedefinition.apiextensions.k8s.io/prometheusrules.monitoring.coreos.com created
customresourcedefinition.apiextensions.k8s.io/servicemonitors.monitoring.coreos.com created
customresourcedefinition.apiextensions.k8s.io/thanosrulers.monitoring.coreos.com created
clusterrolebinding.rbac.authorization.k8s.io/prometheus-operator created
clusterrole.rbac.authorization.k8s.io/prometheus-operator created
deployment.apps/prometheus-operator created
serviceaccount/prometheus-operator created
service/prometheus-operator created
```

This has created a bunch of `custom resource definitions`, which now extends our Kubernetes API and deploys one `prometheus-operator` into the namespace `monitoring`. Also created is the `service account` for this deployment, and `service`.

Check if they are deployed; it can take a few minutes to come up.

```
root@control01:/home/ubuntu/prometheus-operator# kubectl get pods -n monitoring
NAME                                   READY   STATUS    RESTARTS   AGE
prometheus-operator-6cdb7d79fb-trj6h   1/1     Running   0          21s

root@control01:/home/ubuntu/prometheus-operator# kubectl get deploy -n monitoring
NAME                  READY   UP-TO-DATE   AVAILABLE   AGE
prometheus-operator   1/1     1            1           57s

root@control01:/home/ubuntu/prometheus-operator# kubectl get svc -n monitoring
NAME                  TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)    AGE
prometheus-operator   ClusterIP   None         <none>        8080/TCP   70s
```

Before we deploy the actual Prometheus instance, we should first prepare the service monitors. I mean, it doesn't really matter, but you will have to tell Prometheus which service monitors to scrape. Therefore it's better to have them ready first.

Continue with Service Monitors

Last update: October 20, 2021

Comments