Experiment 14: Jaeger Operator

Installation notes

The Jaeger Operator will be installed in a k3d cluster

Note: The Jaeger Operator only works for the number of recent versions of k8s

While multiple operators might coexist watching the same set of namespaces, which operator will succeed in setting itself as the owner of the CR is undefined behavior. Automatic injection of the sidecars might also result in undefined behavior. Therefore, it's highly recommended to have at most one operator watching each namespace. Note that namespaces might contain any number of Jaeger instances (CRs).

The Jaeger Operator version tracks one version of the Jaeger components (Query, Collector, Agent). When a new version of the Jaeger components is released, a new version of the operator will be released that understands how running instances of the previous version can be upgraded to the new version.

Install modes

The Jaeger Operator can be installed to watch for new Jaeger custom resources (CRs) either in the whole cluster or in specific namespaces. When configured for cluster-mode, the operator can:

- watch for events related to Jaeger resources in all namespaces
- watch the namespaces themselves looking for the `sidecar.jaegertracing.io/inject` annotation
- watch all deployments, to inject or remove sidecars based on the `sidecar.jaegertracing.io/inject` annotation
- create cluster role bindings, when necessary

When not using the cluster-wide resources (`ClusterRole` and `ClusterRoleBinding`), set the `WATCH_NAMESPACE` to the comma-separated list of namespaces that the Jaeger Operator should watch for events related to Jaeger resources. It is possible to have the Jaeger Operator running in a given namespace (like, `observability`) and manage Jaeger resources in another (like, `myproject`). For that, use a `RoleBinding` like the following for each namespace the operator should watch for resources:

```
kind: RoleBinding
apiVersion: rbac.authorization.k8s.io/v1
metadata:
  name: jaeger-operator-in-myproject
  namespace: myproject
subjects:
- kind: ServiceAccount
  name: jaeger-operator
  namespace: observability
```

```
roleRef:
  kind: Role
  name: jaeger-operator
  apiGroup: rbac.authorization.k8s.io
```

Installing the Operator

Create our cluster

```
C:\k3d> k3d cluster create local
[36mINFO[0m[0000] Created network 'k3d-local'
[36mINFO[0m[0000] Created volume 'k3d-local-images'
[36mINFO[0m[0001] Creating node 'k3d-local-server-0'
[36mINFO[0m[0001] Creating LoadBalancer 'k3d-local-serverlb'
[36mINFO[0m[0007] Cluster 'local' created successfully!
[36mINFO[0m[0007] You can now use it like this:
kubectl cluster-info
```

Update our KUBECONFIG_FILE for this new cluster

```
C:\k3d> set KUBECONFIG_FILE=C:\k3d\.kube\local
```

Pull our local kubernetes configuration to create our KUBECONFIG_FILE reference

```
C:\k3d> k3d kubeconfig get local > %KUBECONFIG_FILE%
```

Update our KUBECONFIG environment variable to reference our experiment cluster

```
C:\k3d> set KUBECONFIG=%KUBECONFIG_FILE%
```

Verify the cluster information

```
C:\k3d> kubectl cluster-info
Kubernetes master is running at https://0.0.0.0:52508
CoreDNS is running at https://0.0.0.0:52508/api/v1/namespaces/kube-system/services/kube-
dns:dns/proxy
Metrics-server is running at https://0.0.0.0:52508/api/v1/namespaces/kube-
system/services/https:metrics-server:/proxy

To further debug and diagnose cluster problems, use 'kubectl cluster-info dump'.
```

The following instructions will create the `observability` namespace and install the Jaeger Operator there. By default, the operator will watch the same namespace in which it has been installed.

Create the namespace for our new Jaeger Operator

```
kubectl create namespace observability
```

Install the “Custom Resource Definition” for the `apiVersion: jaegertracing.io/v1`

```
kubectl create -f https://raw.githubusercontent.com/jaegertracing/jaeger-operator/master/deploy/crds/jaegertracing.io_jaegers_crd.yaml

kubectl create -n observability -f https://raw.githubusercontent.com/jaegertracing/jaeger-operator/master/deploy/service_account.yaml

kubectl create -n observability -f https://raw.githubusercontent.com/jaegertracing/jaeger-operator/master/deploy/role.yaml

kubectl create -n observability -f https://raw.githubusercontent.com/jaegertracing/jaeger-operator/master/deploy/role_binding.yaml

kubectl create -n observability -f https://raw.githubusercontent.com/jaegertracing/jaeger-operator/master/deploy/operator.yaml
```

The operator will activate extra features if given cluster-wide permissions.

Enable the operator

```
kubectl create -f https://raw.githubusercontent.com/jaegertracing/jaeger-operator/master/deploy/cluster_role.yaml

kubectl create -f https://raw.githubusercontent.com/jaegertracing/jaeger-operator/master/deploy/cluster_role_binding.yaml
```

Note: You’ll need to download and customize the `cluster_role_binding.yaml` if you are using a namespace other than `observability`. You may also want to download and customize the `operator.yaml`, setting the env var `WATCH_NAMESPACES` to have an empty value, so that it can watch for instances across all namespaces.

At this point, there should be a `jaeger-operator` deployment available. You can view it by running the following command:

View our Jaeger Operator Deployment

```
$ kubectl get deployment jaeger-operator -n observability

NAME            DESIRED   CURRENT   UP-TO-DATE   AVAILABLE   AGE
jaeger-operator   1         1         1            1          48s
```

The operator is now ready to create Jaeger instances

Tear down our cluster

```
C:\k3d\kpt-repo> k3d cluster delete local
```

```
[36mINFO[0m[0000] Deleting cluster 'local'
[36mINFO[0m[0000] Deleted k3d-local-serverlb
[36mINFO[0m[0001] Deleted k3d-local-server-0
[36mINFO[0m[0001] Deleting cluster network
'bd7bd4bd8ec595f0bbcc402f5f1090db29db7d27428ed2fa5877bc97a2189367'
[36mINFO[0m[0001] Deleting image volume 'k3d-local-images'
[36mINFO[0m[0001] Removing cluster details from default kubeconfig...
[36mINFO[0m[0001] Removing standalone kubeconfig file (if there is one)...
[36mINFO[0m[0001] Successfully deleted cluster local!
```