Environment Setup for Gemfire Cluster on Kubernetes

Create Gemfire Cluster

We are going to work in the default namespace. There is a more complete example that can be found with the docs at VMware Tanzu Gemfire. In this set of labs we are going to deploy the Tanzu GemFire cluster to the `default` namespace.

By using the default namespace we save on typing for the lab.

Apply the CRD for your Tanzu GemFire cluster, as in this development environment example:

```
$ cat << EOF | kubectl apply -f -
apiVersion: core.geode.apache.org/v1alpha1
kind: GeodeCluster
metadata:
  name: gemfire1
spec:
  exposeExternalManagement: true
  locators:
    replicas: 1
  servers:
    replicas: 2
EOF
```

or you can simply create a yaml file from the contents like gemfire-cluster.yaml:

```
apiVersion: core.geode.apache.org/v1alpha1
kind: GeodeCluster
metadata:
  name: gemfire1
spec:
  exposeExternalManagement: true
  locators:
    replicas: 1
  servers:
    replicas: 2
```

and create the gemfire-cluster with the following command:

```
kubectl apply -f gemfire-cluster.yaml
```

check the creation status of the Tanzu GemFire cluster:

```
 kubectl  get GeodeClusters
```

and you should see an output that looks similar to this:

```
NAME       LOCATORS   SERVERS
gemfire1   2/2        1/2
```

Connect to the Tanzu GemFire Cluster

```
kubectl  exec -it gemfire1-locator-0 -- gfsh
```

Verify Gemfire is working

Since the cluster is deployed for us we need only connect. Do the following:

```
gfsh>connect
```