Creating a Kubernetes Cluster with Jenkins X and Importing App

This lesson provides the link to gists that you can use to create a Kubernetes cluster with Jenkins X. Moreover, it also gives some steps to follow before we continue with the chapter.

We'll cover the following

- Create a new cluster and install Jenkins X
  - GKE
  - EKS
  - AKS
- Install Jenkins X in an existing serverless cluster
- Importing the application

If you kept the cluster from the previous chapter, you can skip this section. Otherwise, we'll need to create a new Jenkins X cluster.

> All the commands from this chapter are available in the 14-upgrade.sh Gist.

Create a new cluster and install Jenkins X

For your convenience, the Gists from the previous chapter are available below as well.

GKE

Create a new serverless **GKE** cluster: gke-jx-serverless.sh

EKS

Create a new serverless **EKS** cluster: eks-jx-serverless.sh

AKS

Create a new serverless **AKS** cluster: aks-jx-serverless.sh

Install Jenkins X in an existing serverless cluster

Use an **existing** serverless cluster: install-serverless.sh

- Create a new static **GKE** cluster: gke-jx.sh
- Create a new serverless **GKE** cluster: gke-jx-serverless.sh
- Create a new static **EKS** cluster: eks-jx.sh
- Create a new serverless **EKS** cluster: eks-jx-serverless.sh
- Create a new static **AKS** cluster: aks-jx.sh
- Create a new serverless **AKS** cluster: aks-jx-serverless.sh
- Use an **existing** static cluster: install.sh
- Use an **existing** serverless cluster: install-serverless.sh

Importing the application

> The commands that follow will reset your *go-demo-6* `master` branch with the contents of the branch that contain all the changes we did so far. Please

> execute them only if you are unsure whether you did all the exercises correctly.

> Depending on whether you're using static or serverless Jenkins X flavor, we'll need to restore one branch or the other. The commands that follow will restore `extension-model-jx` if you are using static Jenkins X, or `extension-model-cd` if you prefer the serverless flavor.

```
NAMESPACE=$(kubectl config view \
    --minify \
    --output jsonpath="{..namespace}")

cd go-demo-6

git pull

git checkout extension-model-$NAMESPACE

git merge -s ours master --no-edit

git checkout master

git merge extension-model-$NAMESPACE

git push

cd ..
```

> If you destroyed the cluster at the end of the previous chapter, you'll need to import the *go-demo-6* application again. Please execute the commands that follow only if you created a new cluster specifically for the exercises from this chapter.

```
cd go-demo-6

jx import --pack go --batch-mode

cd ..
```

Now we are ready to talk about validating upgrades and backing up clusters.