Defining Multiple Objects in the Same YAML file

In this lesson, we will define all the objects used thus far in a single YAML file.

Defining Multiple Objects In The Same YAML file

The `vfarcic/go-demo-2` and `mongo` images form the same stack. They work together and having four YAML definitions is confusing. It would get even more confusing later on since we are going to add more objects to the stack. Things would be much simpler and easier if we would move all the objects we created thus far into a single YAML definition. Fortunately, that is very easy to accomplish.

Let's take a look at yet another YAML file.

```
cat svc/go-demo-2.yml
```

We won't display the output since it is the same as the contents of the previous four YAML files combined. The only difference is that each object definition is separated by three dashes (`---`).

Let's create the objects defined in that file.

```
kubectl create -f svc/go-demo-2.yml
kubectl get -f svc/go-demo-2.yml
```

The **output** of the latter command is as follows.

NAME   DESIRED   CURRENT   READY   AGE

```
NAME                                 DESIRED   CURRENT   READY   AGE
replicaset.apps/go-demo-2-db   1         1         1       11s

NAME                     TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)     AGE
service/go-demo-2-db   ClusterIP   10.103.63.57    <none>        27017/TCP   11s

NAME                                  DESIRED   CURRENT   READY   AGE
replicaset.apps/go-demo-2-api   3         3         2       11s

NAME                      TYPE       CLUSTER-IP      EXTERNAL-IP   PORT(S)          AGE
service/go-demo-2-api   NodePort   10.101.92.202   <none>        8080:30643/TCP   11s
```

The two ReplicaSets and the two Services were created, and we can rejoice in replacing four files with one.

Finally, to be on the safe side, we'll also double check that the stack API is up-and-running and accessible.

```
PORT=$(kubectl get svc go-demo-2-api \
    -o jsonpath="{.spec.ports[0].nodePort}")
curl -i "http://$IP:$PORT/demo/hello"
```

The response is `200` indicating that everything works as expected.

Before we finish the discussion about Services, we might want to go through the discovery process of Services.