**SSH to your AWS Workstation**

**ssh devops@<public-ip-addr**> of your Workstation  
Password is : **Dev0p$!!/**

**Replace <your-name> with your name throughout the lab.**

You can run an application by creating a Kubernetes Deployment object, and you can describe a Deployment in a YAML file.

1. Run the below commands on your AWS-Workstation

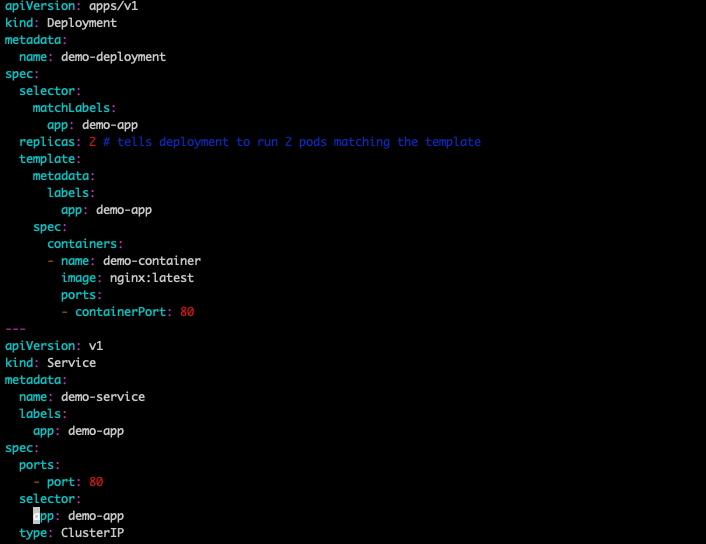
|  |
| --- |
| $ cd /home/devops $ mkdir /home/devops/stateless $ cd /home/devops/stateless  $ curl -f <https://pastebin.com/raw/eTJTi1d0> > <your-name>-app.yaml |

|  |
| --- |
| $ vim <your-name>app.yaml |

Press ‘i’ to start the edit mode in the vim editor. Update <your-name> with your name.

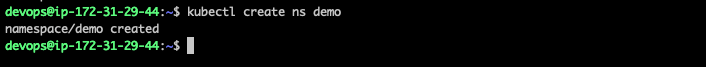
Save and exit by pressing the **ECS key** and type **:wq** and press **enter** to exit.

Example



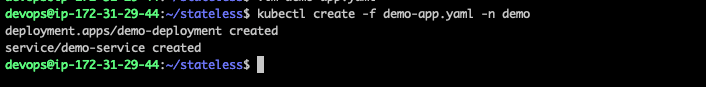
2. Create a Namespace for your deployments in Kubernetes

|  |
| --- |
| $ kubectl create ns <your-name> |



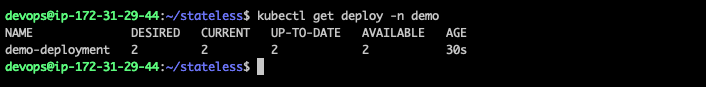
2. Create a Deployment based on the YAML file:

|  |
| --- |
| **$ kubectl create -f <your-name>-app.yaml -n <your-name>** |



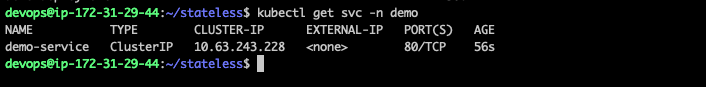
3. Check the Status of the Deployment.

|  |
| --- |
| $ kubectl get deploy -n <your-name> |



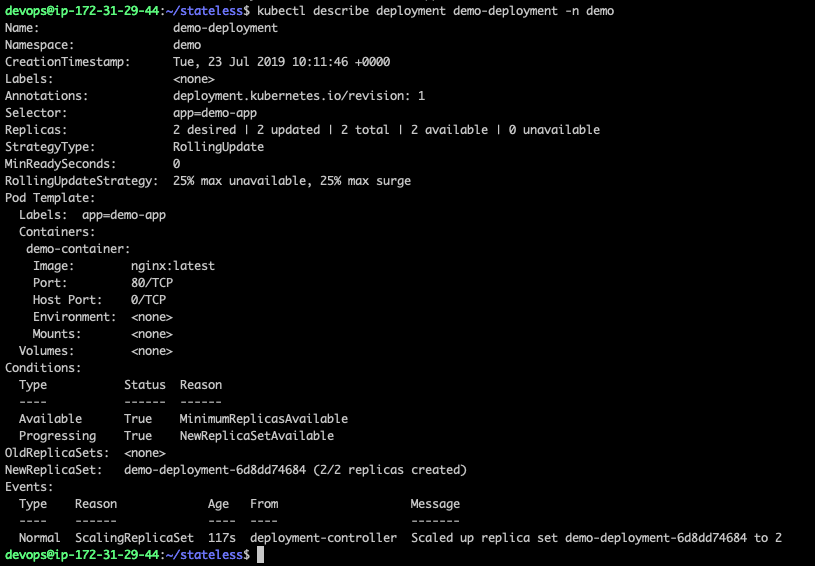
4. Check the Service Status.

|  |
| --- |
| $ kubectl get svc -n <your-name> |



5. Display information about the Deployment:

|  |
| --- |
| **$ kubectl describe deployment <your-name>-deployment -n <your-name>** |



Since the App is not exposed, we will not be able to access it internally or from outside.

And, in order to access the application we will use kubectl port forwarding to check if we are able to access the Application.

kubectl port-forward allows using resource name, such as a pod name, to select a matching pod to port forward

6. Run the below command to port forward the POD.

|  |
| --- |
| $ kubectl port-forward -n <your-name> svc/<your-name>-service 8080:80 & |



Press Ctrl+z to exit.

7. Now, run the below command to check the application.

|  |
| --- |
| $ curl http://127.0.0.1:8080 |

