**Deploying Stateless Application with Deployment Objects.**

Kubectl - kubectl controls the Kubernetes cluster manager

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

**1. Login to your AWS Workstation and make a dir /home/devops/application**

|  |
| --- |
| $ sudo su  # cd /home/devops # mkdir application # cd application/ |

|  |
| --- |
| # vim <your-name>-deployment.yaml |

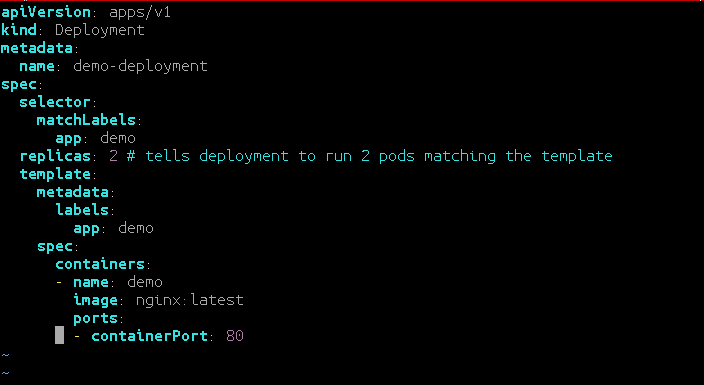
# Paste the below text in the vim editor

#Note : press ‘i’ to start the edit mode in the vim editor. Update <your-name> with your name.

|  |
| --- |
| apiVersion: apps/v1 kind: Deployment metadata:  name: <your-name>-deployment spec:  selector:  matchLabels:  app: <your-app-name>  replicas: 2 # tells deployment to run 2 pods matching the template  template:  metadata:  labels:  app: <your-app-name>  spec:  containers:  - name: <your-container-name>   image: nginx:latest  ports:  - containerPort: 80 |

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

Example



2. Create a Deployment based on the YAML file:

|  |
| --- |
| **# kubectl apply -f <your-name>-deployment.yaml** |

3. Display information about the Deployment:

|  |
| --- |
| **# kubectl describe deployment <your-name>-deployment** |

4. List the pods created by the deployment:

|  |
| --- |
| **# kubectl get pods -l app=<your-app-name>** |

The output is similar to this:

|  |
| --- |
| NAME READY STATUS RESTARTS AGE demo-deployment-1471416983-7o5ac 1/1 Running 0 16h demo-deployment-1541148254-318ad 1/1 Running 0 16h |

5. To display information about a pod:

|  |
| --- |
| **# kubectl describe pod <pod-name>** |

6. Expose the Deployment with the below command.

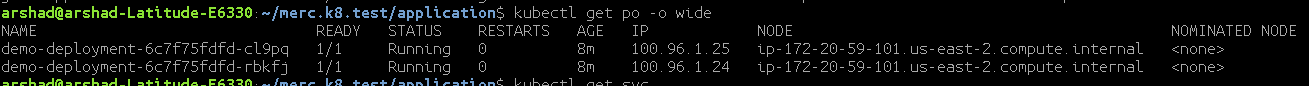
**Update <your-service-name> with your name**

|  |
| --- |
| **# kubectl expose deployment <your-deployment-name> --type=NodePort --name=<your-service-name>** |



7. Check the **NODE** where your app has been deployed.

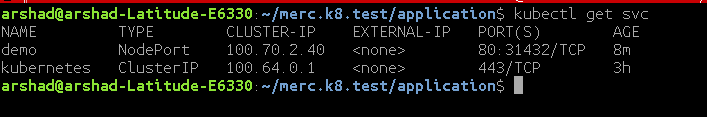
|  |
| --- |
| # kubectl get po -o wide |



In this example the app has been deployed to the NODE **ip-172-20-59-101.us-east-2.compute.internal**

5. Check the NODEPORT on which the application has been exposed

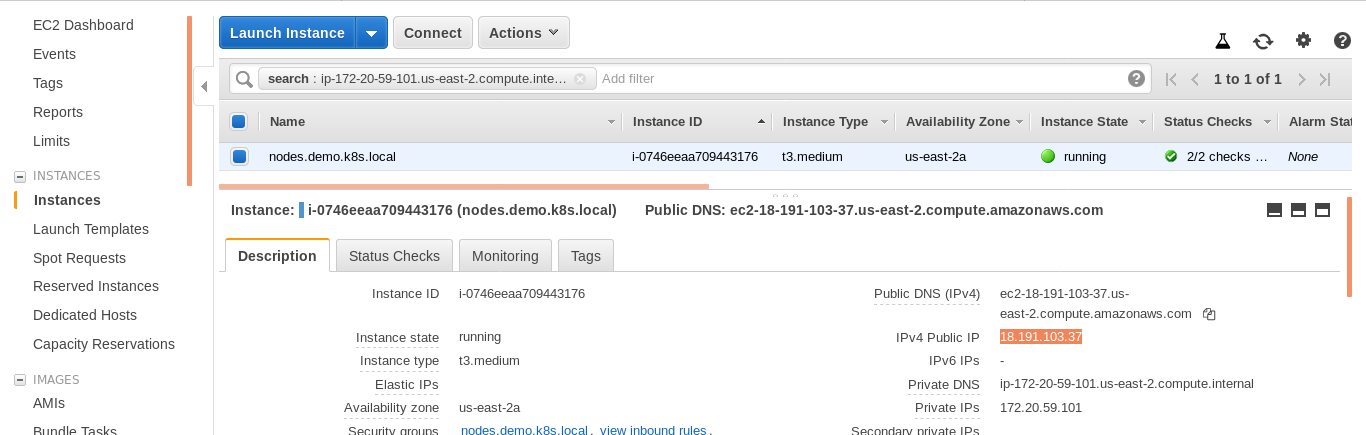
|  |
| --- |
| # kubectl get svc |

****

**Example**

**I**n this example the demo application has been exposed on port **31432** as shown in the below

6. Login to the **AWS** portal and check the public IP of the NODE (**ip-172-20-59-101.us-east-2.compute.internal**) to access your application web page from the NODE Public IP address and Node Port on which it is exposed at.



**http://<NODE-PUBLIC-IP>:NODEPORT**

**Search for the NODE where your application has been deployed on the AWS EC2 Dashboard as shown below.**

**Access the application from the public IP of the NODE and the NodePort as shown below**

Example.

[**http://18.191.103.37:31432/**](http://18.191.103.37:31432/)

