Chapter 6 - Kubernetes 101/201 (Reduced version)

Objective

- Verify the functionality of created cluster in chapter 5
  - Kubernetes control plane
  - Kube-DNS
  - Kube-Proxy
- Manage a Deployment
- Manage a Service

Deployment Management

Create an nginx Deployment:

```
kubectl create -f https://k8s.io/examples/application/dep
```

Check your deployment

List all Deployments:

```
kubectl get deployment
```

List the Pods created by the Deployment:

```
kubectl get pods -l app=nginx
```

Services

Create an nginx Service:

```
kubectl create -f https://k8s.io/examples/service/nginx-s
```

List all services:

```
kubectl get services
```

Get the service IP and port

Provided the service IP is accessible, you should be able to access its http endpoint with wget on the exposed port:

```
export SERVICE_IP=$(kubectl get service nginx-service -o 
export SERVICE_PORT=$(kubectl get service nginx-service -
```

Check `$SERVICE_IP` and `$SERVICE_PORT`:

```
echo "$SERVICE_IP:$SERVICE_PORT"
```

Verify the service

Then, create a busybox Pod:

```
kubectl run busybox --generator=run-pod/v1 --image=busybo

u@busybox$ wget -qO- http://$SERVICE_IP:$SERVICE_PORT # R
u@busybox$ wget -qO- http://nginx-service.default:$SERVIC
u@busybox$ exit # Exit the busybox container
```

After verification, delete the busybox Pod

```
kubectl delete pod busybox # Clean up the busybox Pod
```

Delete the nginx Service

To delete the Service by name:

```
kubectl delete service nginx-service
```

Delete the nginx Deployment by name:

```
kubectl delete deployment nginx-deployment
```

Key Takeaways

- Deployments manage Pods lifecycle
- Services manage access of Pods