Hands On Labs

Look at the composition of the cluster:

```
kubectl get node
kubectl get nodes -o wide
kubectl get no -o yaml
kubectl cluster-info
```

View details about a resource with:

```
kubectl describe nodes
```

View the definition for a resource type with:

```
kubectl explain nodes
```

List the services on our cluster:

```
kubectl get services
```

This would also work:

```
kubectl get svc
```

List pods on our cluster:

```
kubectl get pods
```

List the namespaces on the cluster

```
kubectl get namespaces
```

List the pods in the `kube-system` namespace:

```
kubectl -n kube-system get pods
```

Create a simple pod:

```
kubectl run --image=nginx:1.18 nginx-pod
```

List most resource types:

```
kubectl get all
```

Create a simple deployment:

```
kubectl create deploy --image=nginx nginx-deploy
```

List deployment:

```
kubectl get deploy
```

Port forwarding:

```
kubectl get pods
kubectl port-forward po/nginx-deploy-****   8080:8080  --address 0.0.0.0
```

Scale deployment:

```
kubectl scale –replicas 2 deploy/nginx-deploy
```

List deployment with more info:

```
kubectl get deploy -o wide
```

View the latest logs:

```
kubectl logs deploy/nginx-deploy
```

In a separate window, list pods, and keep watching them:

```
kubectl get pods -w
```

Clean up your deployment

```
kubectl delete deploy/nginx-deploy
```

Start a bunch of ElasticSearch containers:

```
kubectl create deploy elastic --image=elasticsearch:8.6.2 --replicas=4
```

Watch them being started:

```
kubectl get pods -o wide -w
```

Expose the ElasticSearch HTTP API port:

```
kubectl expose deploy/elastic --port 9200
```

Look up which IP address was allocated:

```
kubectl get svc
```

the IP address that was allocated for our service:

```
IP=$(kubectl get svc elastic -o go-template --template '{{ .spec.clusterIP }}')
```

Send a few requests:

```
curl http://$IP:9200/
```

let’s clean it up

```
kubectl delete deploy/elastic
```