Day 4

Info - Route vs Ingress

```
- Ingress is a forwarding rule
- Ingress is not a service
- Ingress helps us forwarding the calls to multiple services based on path
  For example
   - assume the base url or home page of your web site is www.tektutor.org
   - when users attempt to login with url www.tektutor.org/login, the call
should be forwarded to login clusterip service.
   - when user attempt to logout with url www.tektutor.org/logout, the call
should be forwarded to logout clusterip service.
- Route forward the call to only a single service unlike Ingress
- Route is a new feature added in Openshift
- Route is based on Kubernetes ingress
```

Lab - Deploying an application from GitHub source using docker strategy

The new-app command creates deployment and service.

```
oc new-project jegan
oc new-app https://github.com/tektutor/spring-ms.git --strategy=docker
```

We need to expoxe the service to create a public route as shown below. Route is a new feature added in openshift but based on Kubenetes ingress.

```
oc expose svc/spring-ms
oc get routes
oc get route
```

To check build log

```
oc logs -f bc/spring-ms
```

Lab - Deploying an application from GitHub source using source strategy

Before proceeding, you need delete the previous deployment to avoid errors

```
oc delete deploy/spring-ms svc/spring-ms route/spring-ms bc/spring-ms
```

Now you may proceed deploy application using source strategy

```
oc new-project jegan
oc new-app registry.access.redhat.com/ubi8/openjdk-
11~https://github.com/tektutor/spring-ms.git --strategy=source
oc expose svc/spring-ms
```

To check build log

```
oc logs -f bc/spring-ms
```

Lab - Deploying an application using Container image from Docker Hub Registry

```
oc new-app --name=nginx bitnami/nginx:latest
oc expose svc/nginx
```

You may check the status of the application in the webconsole developer topology. In order to access the application, you can click on the route url arrow pointing diagonally upward.

Lab - Deploying nginx in declarative style

Delete any deployment you may have created

```
oc delete project jegan
oc new-project jegan
```

You may proceed now as shown below

```
oc create deployment nginx --image=bitnami/nginx:1.18 --replicas=3 -o yaml
--dry-run=client
oc create deployment nginx --image=bitnami/nginx:1.18 --replicas=3 -o yaml
--dry-run=client > nginx-deploy.yml
cat nginx-deploy.yml
oc apply -f nginx-deploy.yml

oc get deploy,rs,po
```

Lab - Creating a clusterip internal service in declarative style

```

```

```
oc expose deploy/nginx --type=ClusterIP --port=8080 -o yaml --dry-
run=client
oc expose deploy/nginx --type=ClusterIP --port=8080 -o yaml --dry-
run=client > nginx-clusterip-svc.yml
oc apply -f nginx-clusterip-svc.yml

oc get svc
oc describe svc/nginx
```

To delete the clusterip service declaratively

```
oc delete -f nginx-clusterip-svc.yml
oc get svc
```

Lab - Creating a nodeport external service in declarative style

```
oc expose deploy/nginx --type=NodePort --port=8080 -o yaml --dry-run=client
oc expose deploy/nginx --type=NodePort --port=8080 -o yaml --dry-run=client
> nginx-nodeport-svc.yml
oc apply -f nginx-nodeport-svc.yml

oc get svc
oc describe svc/nginx
```

To delete the nodeport service declaratively

```
oc delete -f nginx-nodeport-svc.yml
oc get svc
```

Lab - Creating a loadbalancer external service in declarative style

```
oc expose deploy/nginx --type=LoadBalancer --port=8080 -o yaml --dry-
run=client
oc expose deploy/nginx --type=LoadBalancer --port=8080 -o yaml --dry-
run=client > nginx-lb-svc.yml
oc apply -f nginx-lb-svc.yml

oc get svc
oc describe svc/nginx
```

To delete the loadbalancer service declaratively

```
oc delete -f nginx-lb-svc.yml
oc get svc
```