

# Docker Compose to Kubernetes

- [Converting from Docker-Compose](#)
  - [Install kompose](#)
  - [Startup without Converting](#)
  - [Exporting](#)
  - [Create Kubernetes pods in Cluster](#)
- [Pulling image from Docker Registry](#)
  - [Create a Secret in the cluster that holds your authorization token](#)
  - [Revise deployment.yaml files](#)
  - [Setup SSH Tunnel](#)
- [References](#)

## Converting from Docker-Compose

The first step will be to install kompose and convert our docker-compose.yml to kubernetes yaml files.

### Install kompose

```
> brew install kompose
```

### Startup without Converting

For simple docker-compose files, we can skip the steps of converting and start up directly using kompose up.

```
> kompose up
```

### Exporting

For more complicated docker-compose files, we will need to export and then modify the converted files.

```
> kompose convert
```

```
$ kompose convert
WARN Volume mount on the host "/Users/john.mehan/projects/cloud/deployment/is-config/IND" isn't supported -
ignoring path on the host
INFO Kubernetes file "cloudservice-service.yaml" created
INFO Kubernetes file "isservice-service.yaml" created
INFO Kubernetes file "postgres-service.yaml" created
INFO Kubernetes file "redis-service.yaml" created
INFO Kubernetes file "cloudservice-deployment.yaml" created
INFO Kubernetes file "isservice-deployment.yaml" created
INFO Kubernetes file "isservice-claim0-persistentvolumeclaim.yaml" created
INFO Kubernetes file "postgres-deployment.yaml" created
INFO Kubernetes file "db-volume-persistentvolumeclaim.yaml" created
INFO Kubernetes file "redis-deployment.yaml" created
```

## Create Kubernetes pods in Cluster

We are using minikube for our cluster. Let's start by making sure it is running:

```
$ minikube start
```

Using the yaml files created by kompose export, let's create our pods:

```
$ kubectl create -f cloudservice-service.yaml,isservice-service.yaml,postgres-service.yaml,redis-service.yaml,
cloudservice-deployment.yaml,isservice-deployment.yaml,isservice-claim0-persistentvolumeclaim.yaml,postgres-
deployment.yaml,db-volume-persistentvolumeclaim.yaml,redis-deployment.yaml

service/cloudservice created
service/isservice created
service/postgres created
service/redis created
deployment.extensions/cloudservice created
deployment.extensions/isservice created
persistentvolumeclaim/isservice-claim0 created
deployment.extensions/postgres created
persistentvolumeclaim/db-volume created
deployment.extensions/redis created
```

## Pulling image from Docker Registry

Kubernetes will pull docker images from docker hub by default. For our own custom images, we will pull them from our local docker registry.

For this to work we need to

- create a secret in the cluster that holds our authorization token
- update our deployment yaml files to include this secret
- setup an ssh tunnel between our localhost running docker registry and our cluster (minikube)

### Create a Secret in the cluster that holds your authorization token

```
> kubectl create secret docker-registry regcred --docker-server=<your-registry-server> --docker-username=<your-
name> --docker-password=<your-pword> --docker-email=<your-email>
```

```
$ kubectl create secret docker-registry regcred --docker-server=http://localhost:5000/ --docker-
username=username --docker-password=password --docker-email=john.mehan@irdeto.com
secret/regcred created
```

### Revise deployment.yaml files

Add imagePullSecrets section to your deployment yaml files:

```
imagePullSecrets:
- name: regcred
```

example:

```
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  ...
spec:
  replicas: 1
  strategy:
    type: Recreate
  template:
    metadata:
      creationTimestamp: null
      labels:
        io.kompose.service: isservice
    spec:
      containers:
        - image: localhost:5000/safa_is:0.1
          name: isservice
          ports:
            - containerPort: 8080
          resources: {}
          tty: true
          volumeMounts:
            - mountPath: /home/irdeto/CONF/ISF/IND
              name: is-volume
      restartPolicy: Always
      volumes:
        - name: is-volume
          persistentVolumeClaim:
            claimName: is-volume
      imagePullSecrets:
        - name: regcred
status: {}
```

## Setup SSH Tunnel

We will setup an SSH Tunnel in order to allow kubernetes to pull images from our local docker registry.

```
$ ssh -i ~/.minikube/machines/minikube/id_rsa docker@$(minikube ip) -R 5000:localhost:5000
```

## References

Reference	URL
Translate docker compose to Kubernetes	<a href="https://kubernetes.io/docs/tasks/configure-pod-container/translate-compose-kubernetes/">https://kubernetes.io/docs/tasks/configure-pod-container/translate-compose-kubernetes/</a>
Katacoda Kubernetes Playground	<a href="https://www.katacoda.com/courses/kubernetes/playground">https://www.katacoda.com/courses/kubernetes/playground</a>
Pull an image from a Private Registry	<a href="https://kubernetes.io/docs/tasks/configure-pod-container/pull-image-private-registry/">https://kubernetes.io/docs/tasks/configure-pod-container/pull-image-private-registry/</a>