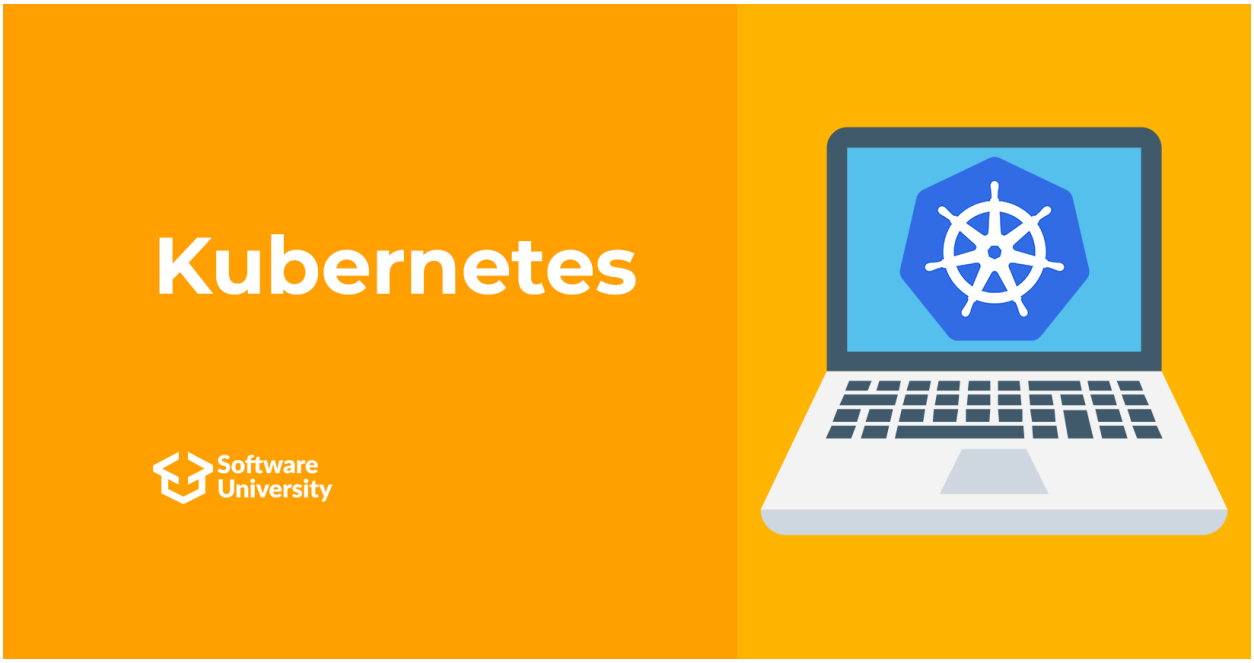


Kubernetes

June 2022

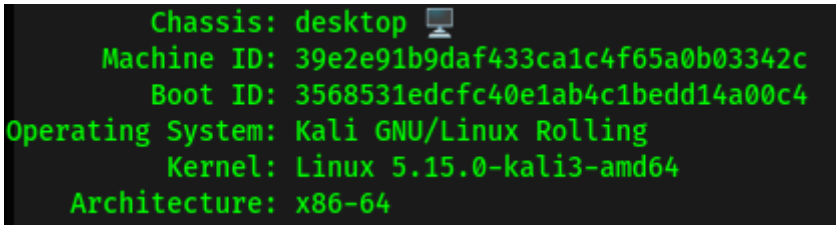
Advanced Concepts

Home Work

Stefan Veselinov

Environment:

Host



Tasks Solution

1. **Init Containers**
   a. Create a set of two **init containers** and **one app container** *(in fact modify/extend the example shown during the practice)*
   b. The **first init container** should generate the following two lines with **10 seconds** delay
   dd-mm-yyyy hh:mi:ss => begin initialization …
   dd-mm-yyyy hh:mi:ss => … done
   Please note that the dd-mm-yyyy hh:mi:ss should reflect the actual time the event is taking place
   c. The **second init container** should add one more line like the following
   dd-mm-yyyy hh:mi:ss => launching the application …
   Please note that the dd-mm-yyyy hh:mi:ss should reflect the actual time the event is taking place
   d. The **app container** should be **nginx** based and should **display the three lines** generated by the init containers instead of the **default index page**

```
vagrant@node1:/vagrant/homework$ kubectl apply -f init-cnt-pod.yaml
pod/init-cnt-pod created
service/svc-init created
```

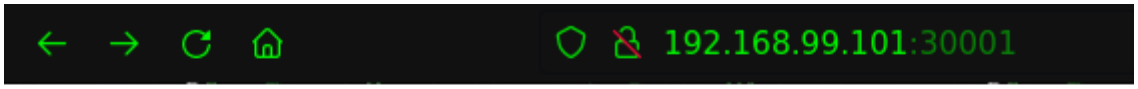
```
vagrant@node1:/vagrant/homework$ kubectl get svc,pod
NAME                 TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)        AGE
service/kubernetes   ClusterIP   10.96.0.1       <none>        443/TCP        26h
service/svc-init     NodePort    10.104.97.186   <none>        80:30001/TCP   34s

NAME               READY   STATUS    RESTARTS   AGE
pod/init-cnt-pod   1/1     Running   0          34s
```

```
Init Containers:
  init-cont-10s:
    Container ID:  docker://85a97f6dacd2c5f5f01c88c391461b54138affd04e3d4249b1ef147336f0fa38
    Image:         alpine
    Image ID:      docker-pullable://alpine@sha256:686d8c9dfa6f3ccfc8230bc3178d23f84eeaf7e457f36f271ab1acc53015037c
    Port:          <none>
    Host Port:     <none>
    Command:
      /bin/sh
      -c
    Args:
      while true; do echo $(date +'%d-%m-%Y %H:%M:%S') '=> begin initialization ...' '<br />' >> /data/index.html; sleep 10; echo $(date +'%d-%m-%Y %H:%M:%S')
 '=> ... done'  '<br />' >> /data/index.html; break; done
    State:          Terminated
      Reason:       Completed
```

```
init-cont-start-up:
  Container ID:  docker://df975b232c678fedd52418ed0b177cc8ee756b6a2c61ca0942d2aace75228c71
  Image:         alpine
  Image ID:      docker-pullable://alpine@sha256:686d8c9dfa6f3ccfc8230bc3178d23f84eeaf7e457f36f271ab1acc
  Port:          <none>
  Host Port:     <none>
  Command:
    /bin/sh
    -c
  Args:
    echo $(date +'%d-%m-%Y %H:%M:%S') '=> launching the application ...' '<br />' >> /data/index.html;
  State:          Terminated
    Reason:       Completed
```

```
Containers:
  app-container:
    Container ID:   docker://84fd9b27eabf54a767f53674bde04a07c68
    Image:          nginx
    Image ID:       docker-pullable://nginx@sha256:10f14ffa93f8d
    Port:           80/TCP
    Host Port:      0/TCP
    State:          Running
      Started:      Fri, 08 Jul 2022 13:37:11 +0300
    Ready:          True
```



08-07-2022 10:36:57 => begin initialization ...
08-07-2022 10:37:07 => ... done
08-07-2022 10:37:09 => launching the application ...

2. **Ingress** and **TLS**
   a. Using either **NGINX** or **HAProxy** *(the implementation should not differ significantly)* **ingress controller** try to modify/extend the **fan out** example shown in the practice to **handle TLS traffic**
   b. Note, that you will need to create a **self-signed certificate** and store it in a **secret** which then to be **used in the ingress**

- Creating TLS certificate/secret

- **mkdir cert/**
- **cd cert/**
- **openssl genrsa -out node.key 2048**
- **openssl req -new -key node.key -out node.csr -subj "/CN=node/O=node"**
- **`openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout node.key -out node.crt -subj "/CN=node/O=node"`**
- **`kubectl create secret tls node --key node.key --cert node.crt`**

```
vagrant@node1:~/cert$ kubectl get secret
NAME                  TYPE                                  DATA   AGE
default-token-nkjxk   kubernetes.io/service-account-token   3      45h
node                  kubernetes.io/tls                     2      24s
```

- Setup and Create NGINX Ingress

    - git clone https://github.com/nginxinc/kubernetes-ingress.git --branch v2.2.2

    - cd kubernetes-ingress/deployments

    - kubectl apply -f common/ns-and-sa.yaml

    - kubectl apply -f rbac/rbac.yaml

    - kubectl apply -f common/default-server-secret.yaml

    - kubectl apply -f common/nginx-config.yaml

    - kubectl apply -f common/ingress-class.yaml

    - kubectl apply -f common/crds/k8s.nginx.org_virtualservers.yaml

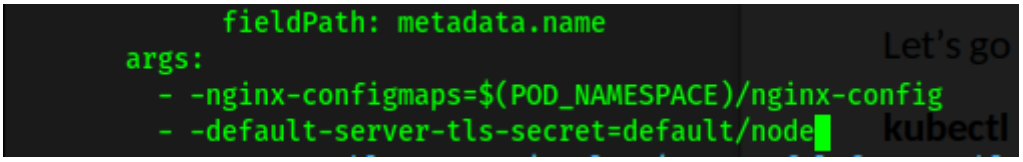
    - kubectl apply -f common/crds/k8s.nginx.org_virtualserverroutes.yaml

    - kubectl apply -f common/crds/k8s.nginx.org_transportservers.yaml

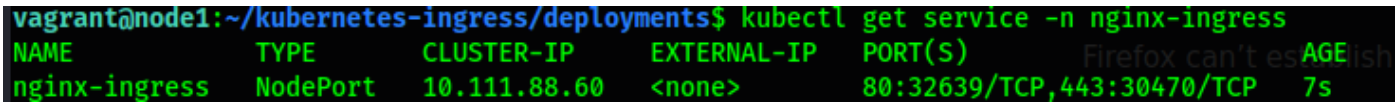
    - kubectl apply -f common/crds/k8s.nginx.org_policies.yaml

    - kubectl apply -f deployment/nginx-ingress.yaml

- Edit configuration to use TLS secret



    - kubectl create -f service/nodeport.yaml



- Applying manifests