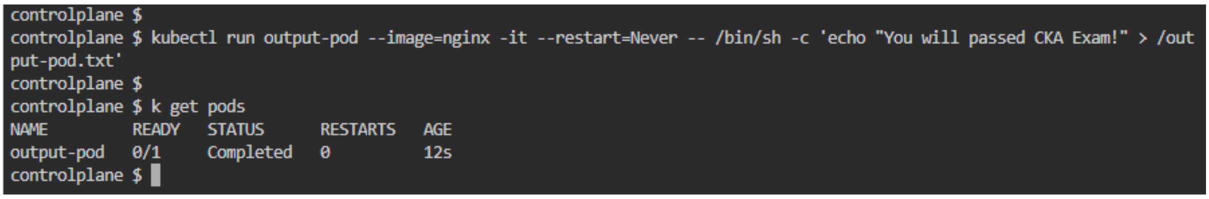
Exercise - Pods and Nodes

Name: Haffiz Hissham

Date: 21 September 2024

Q1

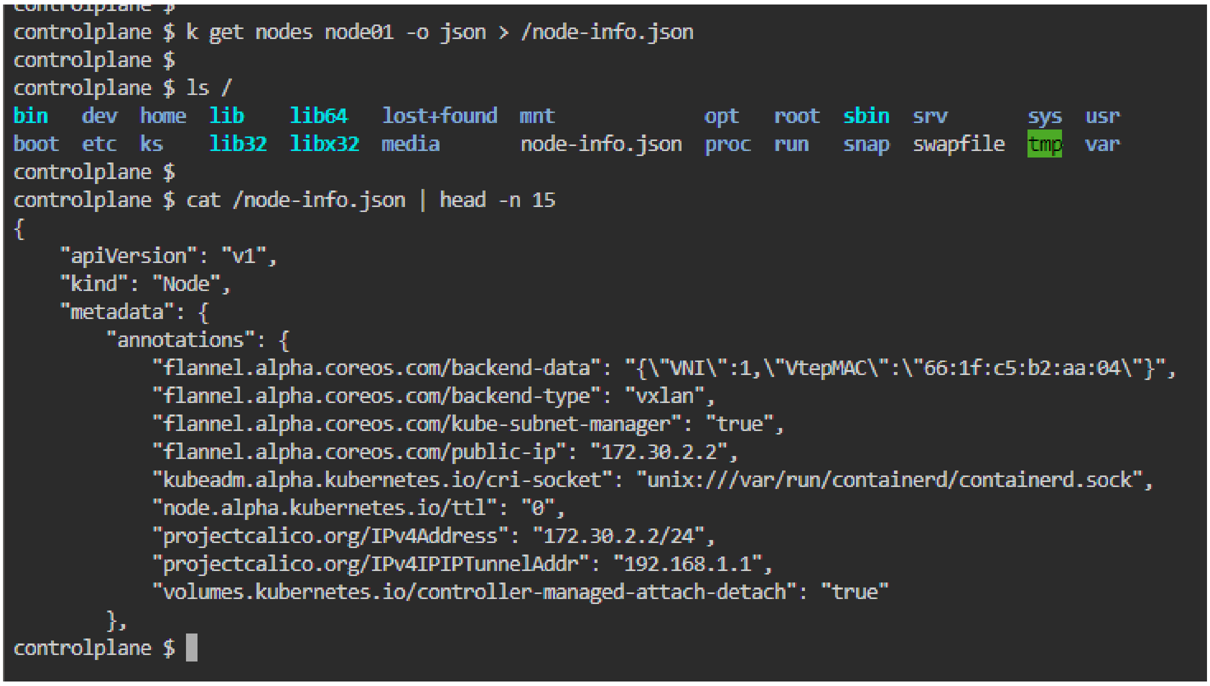
Create a pod output-pod which write "You will passed CKA Exam!" into a file "output-pod.txt"
The Pod output-pod should be deleted automatically after writing the text to the file.



```
controlplane $
controlplane $ kubectl run output-pod --image=nginx -it --restart=Never -- /bin/sh -c 'echo "You will passed CKA Exam!" > /out
put-pod.txt'
controlplane $
controlplane $ k get pods
NAME         READY   STATUS      RESTARTS   AGE
output-pod   0/1     Completed   0          12s
controlplane $
```

Q2

Get the node node01 in JSON format and store it in a file at /node-info.json



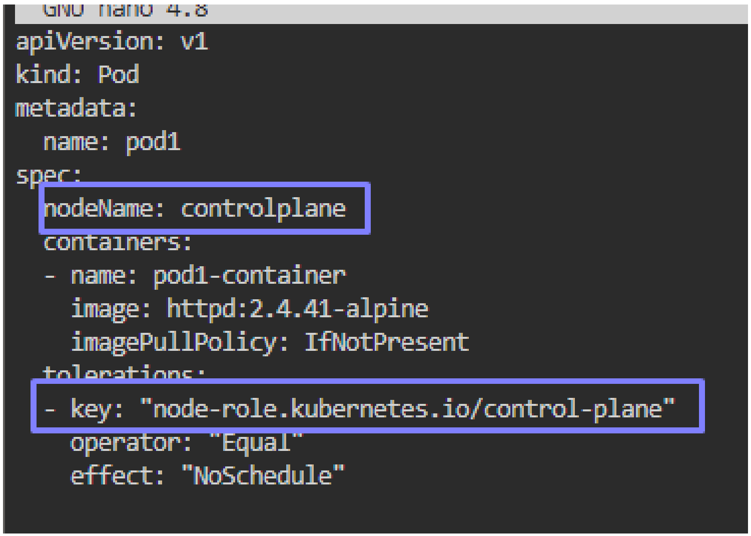
```
controlplane $
controlplane $ k get nodes node01 -o json > /node-info.json
controlplane $
controlplane $ ls /
bin   dev  home  lib    lib64   lost+found  mnt             opt   root  sbin  srv       sys  usr
boot  etc  ks    lib32  libx32  media       node-info.json  proc  run   snap  swapfile  tmp  var
controlplane $
controlplane $ cat /node-info.json | head -n 15
{
    "apiVersion": "v1",
    "kind": "Node",
    "metadata": {
        "annotations": {
            "flannel.alpha.coreos.com/backend-data": "{\"VNI\":1,\"VtepMAC\":\"66:1f:c5:b2:aa:04\"}",
            "flannel.alpha.coreos.com/backend-type": "vxlan",
            "flannel.alpha.coreos.com/kube-subnet-manager": "true",
            "flannel.alpha.coreos.com/public-ip": "172.30.2.2",
            "kubeadm.alpha.kubernetes.io/cri-socket": "unix:///var/run/containerd/containerd.sock",
            "node.alpha.kubernetes.io/ttl": "0",
            "projectcalico.org/IPv4Address": "172.30.2.2/24",
            "projectcalico.org/IPv4IPIPTunnelAddr": "192.168.1.1",
            "volumes.kubernetes.io/controller-managed-attach-detach": "true"
        },
controlplane $
```

Q3

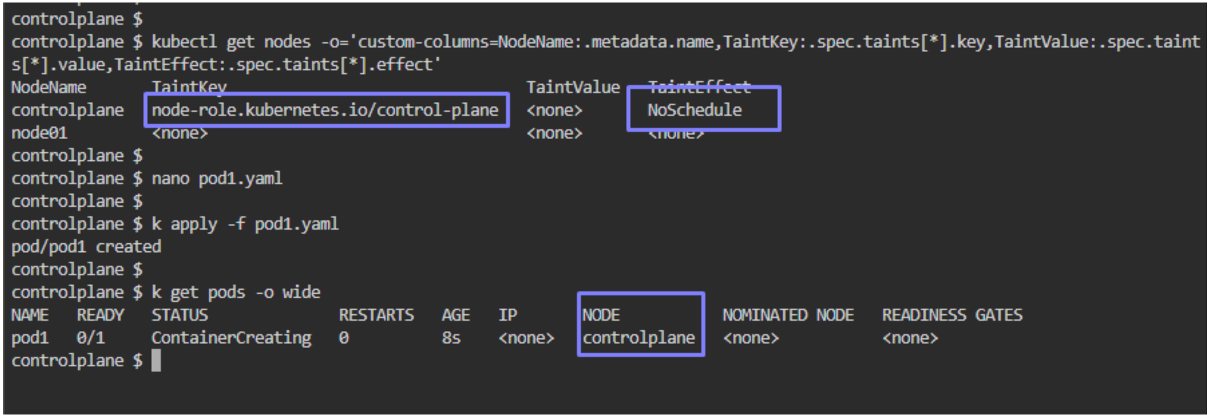
Create a single Pod of image httpd:2.4.41-alpine in Namespace default.

The Pod should be named pod1 and the container should be named pod1-container.
This Pod should only be scheduled on controlplane nodes using tolerations as well .

Do not add new labels to any nodes.



```
  GNU nano 4.8
apiVersion: v1
kind: Pod
metadata:
  name: pod1
spec:
  nodeName: controlplane
  containers:
  - name: pod1-container
    image: httpd:2.4.41-alpine
    imagePullPolicy: IfNotPresent
  tolerations:
  - key: "node-role.kubernetes.io/control-plane"
    operator: "Equal"
    effect: "NoSchedule"
```



```
controlplane $
controlplane $ kubectl get nodes -o='custom-columns=NodeName:.metadata.name,TaintKey:.spec.taints[*].key,TaintValue:.spec.taint
s[*].value,TaintEffect:.spec.taints[*].effect'
NodeName       TaintKey                               TaintValue   TaintEffect
controlplane   node-role.kubernetes.io/control-plane   <none>       NoSchedule
node01         <none>                                 <none>       <none>
controlplane $
controlplane $ nano pod1.yaml
controlplane $
controlplane $ k apply -f pod1.yaml
pod/pod1 created
controlplane $
controlplane $ k get pods -o wide
NAME   READY   STATUS              RESTARTS   AGE   IP       NODE           NOMINATED NODE   READINESS GATES
pod1   0/1     ContainerCreating   0          8s    <none>   controlplane   <none>           <none>
controlplane $
```