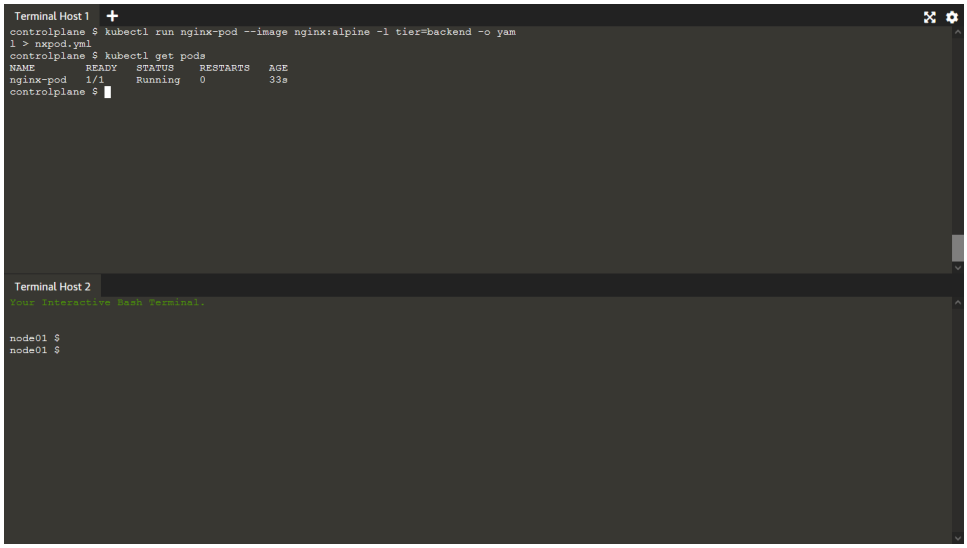
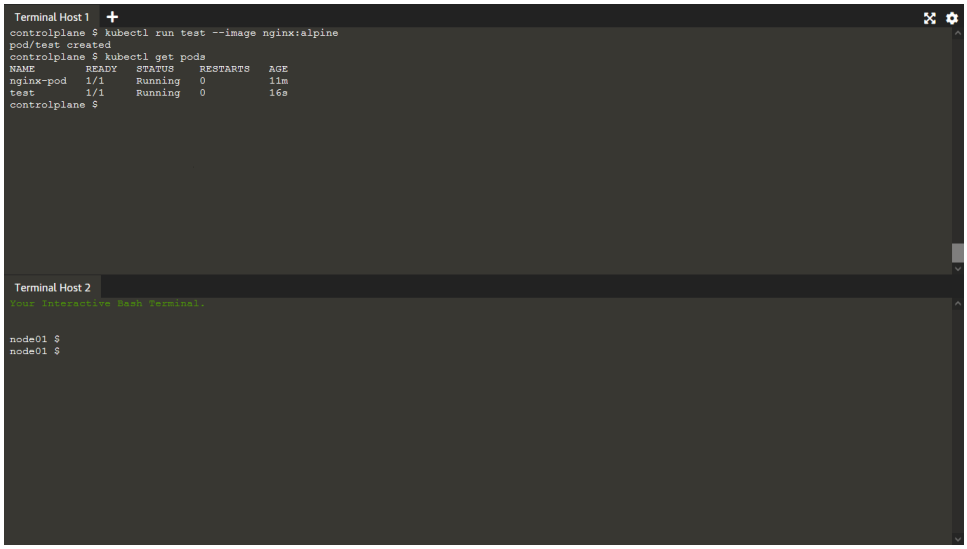
1-


Terminal Host 1
controlplane $ kubectl run nginx-pod --image nginx:alpine -l tier=backend -o yam
l > nxpod.yml
controlplane $ kubectl get pods
NAME READY STATUS RESTARTS AGE
nginx-pod 1/1 Running 0 33s
controlplane $
Terminal Host 2
Your Interactive Bash Terminal.
node01 $
node01 $

2-


Terminal Host 1
controlplane $ kubectl run test --image nginx:alpine
pod/test created
controlplane $ kubectl get pods
NAME READY STATUS RESTARTS AGE
nginx-pod 1/1 Running 0 11m
test 1/1 Running 0 16s
controlplane $
Terminal Host 2
Your Interactive Bash Terminal.
node01 $
node01 $

Terminal Host 1

```
apiVersion: v1
kind: Pod
metadata:
  creationTimestamp: null
  labels:
    tier: backend
  name: nginx-pod
spec:
  containers:
  - image: nginx:alpine
    name: nginx-pod
    resources: {}
  dnsPolicy: ClusterFirst
  restartPolicy: Always
status: {}
~
~
~
~
```

Terminal Host 2

```
Your Interactive Bash Terminal.


node01 $
node01 $
```

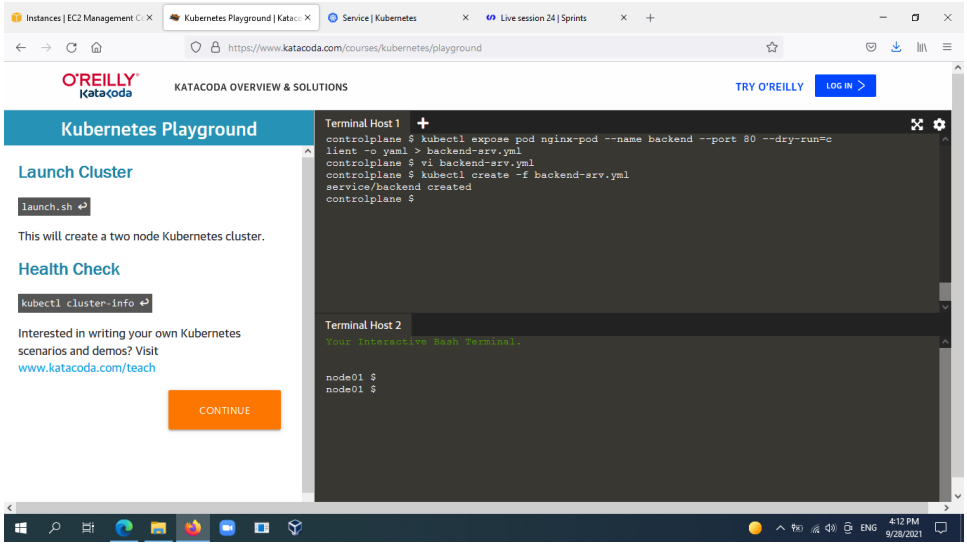
3-

Terminal Host 1

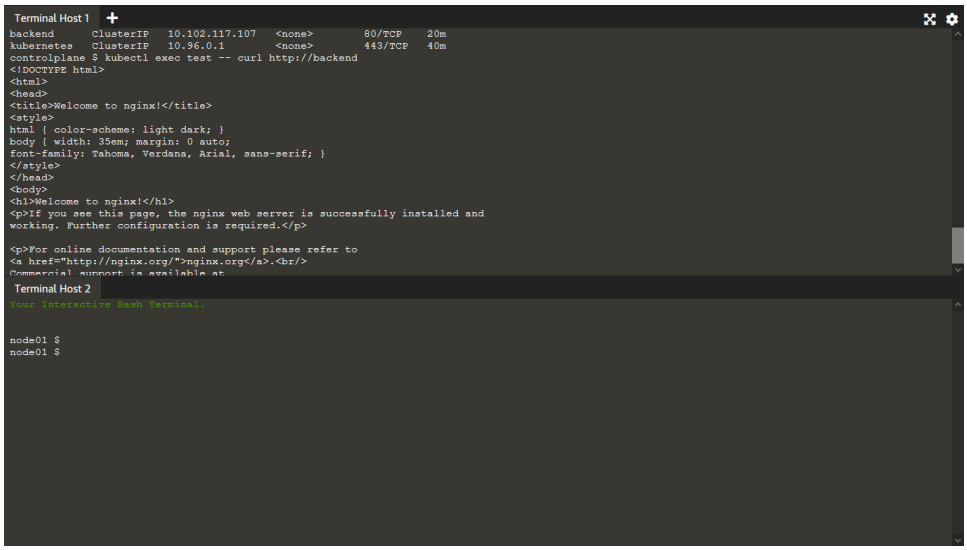
```
apiVersion: v1
kind: Service
metadata:
  creationTimestamp: null
  labels:
    tier: backend
  name: backend
spec:
  ports:
  - port: 80
    protocol: TCP
    targetPort: 80
  selector:
    tier: backend
status:
  loadBalancer: {}
~
~
~
```

Terminal Host 2

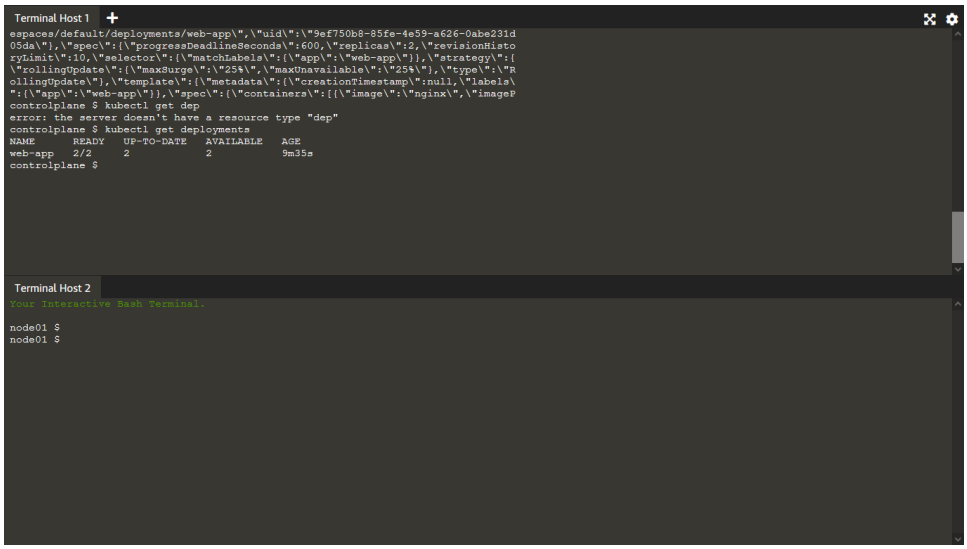
```
Your Interactive Bash Terminal.


node01 $
node01 $
```

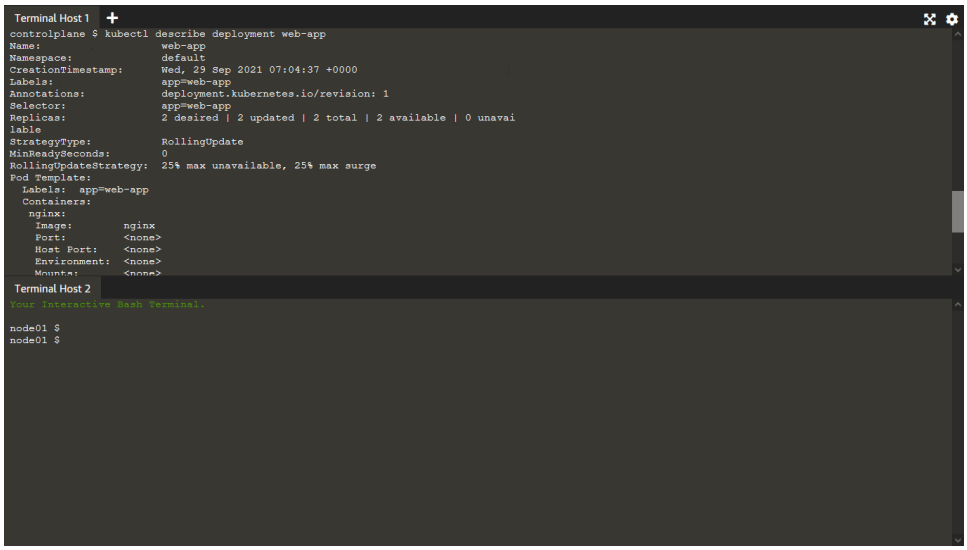

Instances | EC2 Management
Kubernetes Playground | Katac
Service | Kubernetes
Live session 24 | Sprints
https://www.katacoda.com/courses/kubernetes/playground
O'REILLY Katacoda
KATACODA OVERVIEW & SOLUTIONS
TRY O'REILLY
LOG IN
Kubernetes Playground
Launch Cluster
launch.sh
This will create a two node Kubernetes cluster.
Health Check
kubectl cluster-info
Interested in writing your own Kubernetes scenarios and demos? Visit
www.katacoda.com/teach
CONTINUE
Terminal Host 1
controlplane $ kubectl expose pod nginx-pod --name backend --port 80 --dry-run=c
lient -o yaml > backend-srv.yml
controlplane $ vi backend-srv.yml
controlplane $ kubectl create -f backend-srv.yml
service/backend created
controlplane $
Terminal Host 2
Your Interactive Bash Terminal.
node01 $
node01 $
ENG
4:12 PM
9/28/2021

4-

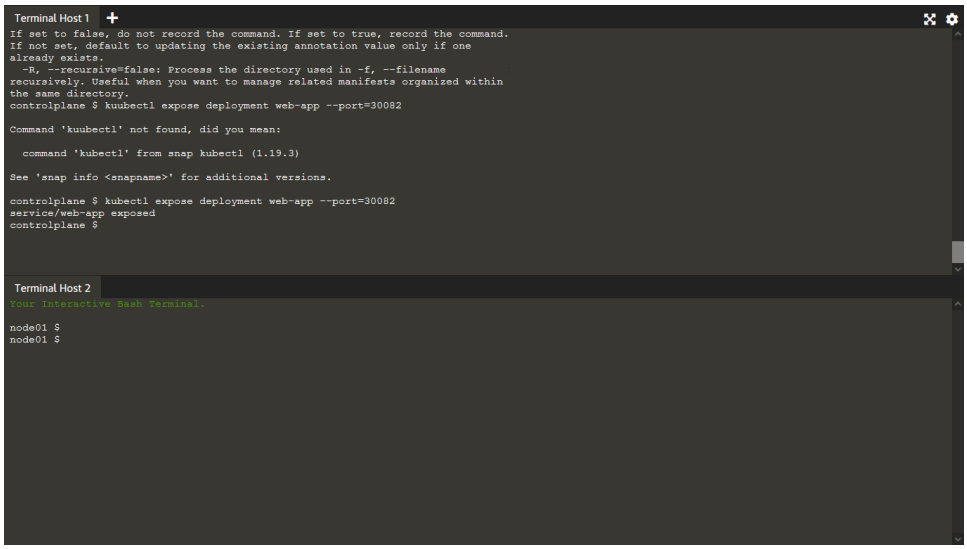

Terminal Host 1
backend ClusterIP 10.102.117.107 <none> 80/TCP 20m
kubernetes ClusterIP 10.96.0.1 <none> 443/TCP 40m
controlplane $ kubectl exec test -- curl http://backend
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
html { color-scheme: light dark; }
body { width: 35em; margin: 0 auto;
font-family: Tahoma, Verdana, Arial, sans-serif; }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.
For online documentation and support please refer to
nginx.org.
Commercial support is available at
Terminal Host 2
Your Interactive Bash Terminal.
node01 $
node01 $

5-


Terminal Host 1
espaces/default/deployments/web-app\",\"uid\":\"9ef750b8-85fe-4e59-a626-0abe231d
05da\"},\"spec\":{\"progressDeadlineSeconds\":600,\"replicas\":2,\"revisionHisto
ryLimit\":10,\"selector\":{\"matchLabels\":{\"app\":\"web-app\"}},\"strategy\":{
\"rollingUpdate\":{\"maxSurge\":\"25%\",\"maxUnavailable\":\"25%\"},\"type\":\"R
ollingUpdate\"},\"template\":{\"metadata\":{\"creationTimestamp\":null,\"labels\
":{\"app\":\"web-app\"}},\"spec\":{\"containers\":[{\"image\":\"nginx\",\"imageP
controlplane $ kubectl get dep
error: the server doesn't have a resource type "dep"
controlplane $ kubectl get deployments
NAME   READY   UP-TO-DATE   AVAILABLE   AGE
web-app   2/2   2   2   9m35s
controlplane $
Terminal Host 2
Your Interactive Bash Terminal.
node01 $
node01 $

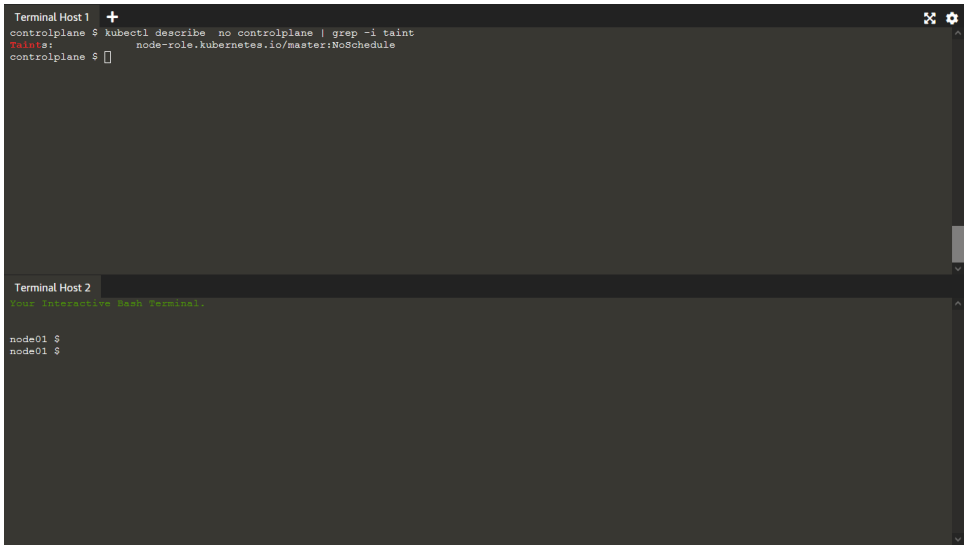

Terminal Host 1
controlplane $ kubectl describe deployment web-app
Name: web-app
Namespace: default
CreationTimestamp: Wed, 29 Sep 2021 07:04:37 +0000
Labels: app=web-app
Annotations: deployment.kubernetes.io/revision: 1
Selector: app=web-app
Replicas: 2 desired | 2 updated | 2 total | 2 available | 0 unavai
lable
StrategyType: RollingUpdate
MinReadySeconds: 0
RollingUpdateStrategy: 25% max unavailable, 25% max surge
Pod Template:
  Labels: app=web-app
  Containers:
   nginx:
    Image: nginx
    Port: <none>
    Host Port: <none>
    Environment: <none>
    Mounts: <none>
Terminal Host 2
Your Interactive Bash Terminal.
node01 $
node01 $

6-

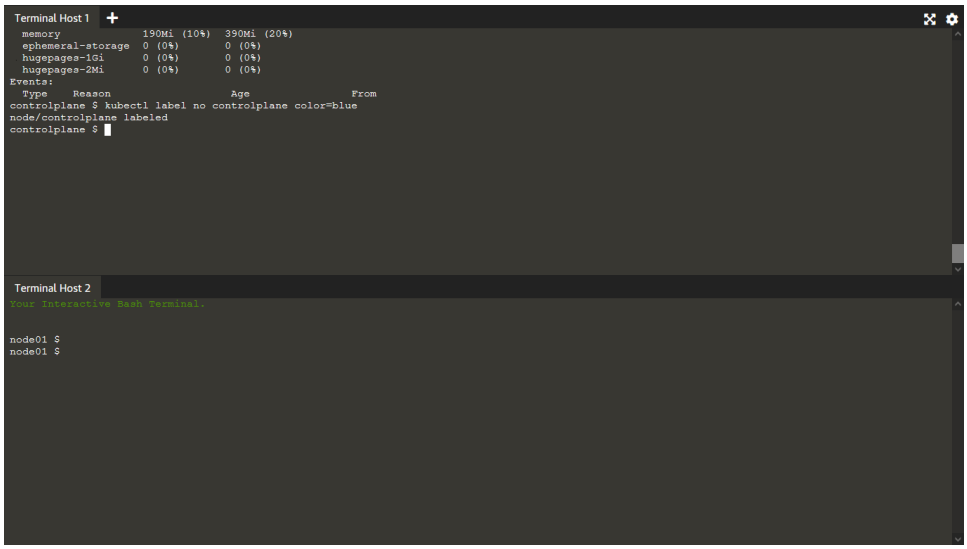


7&8- need another env.

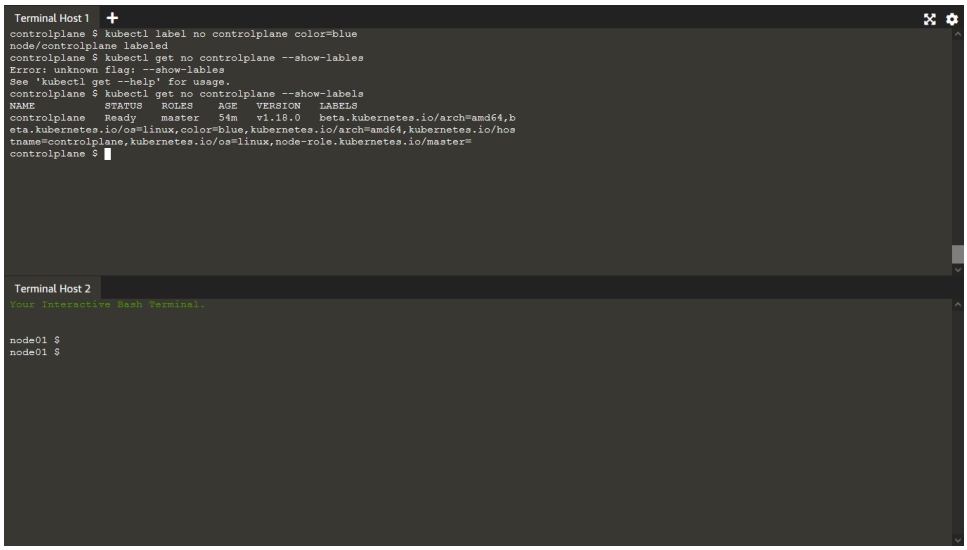
9-

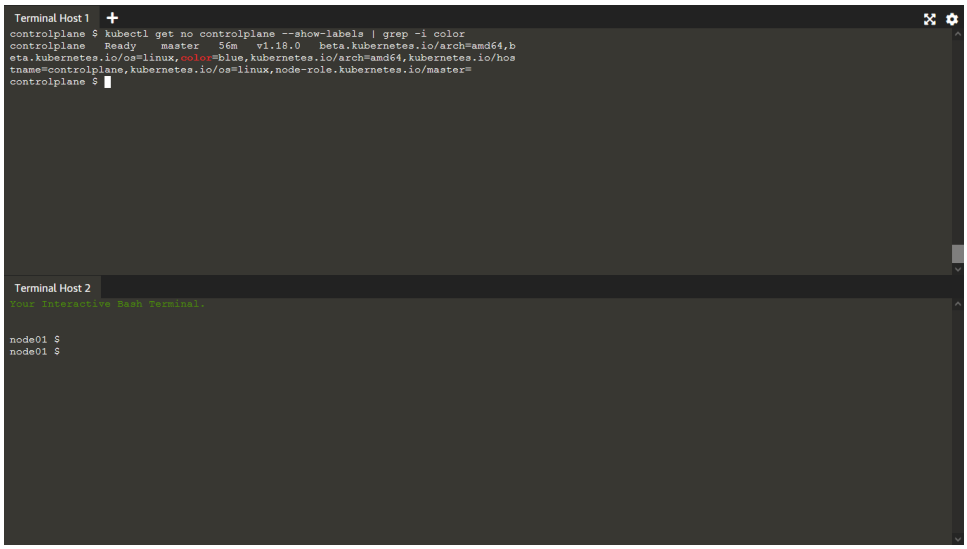


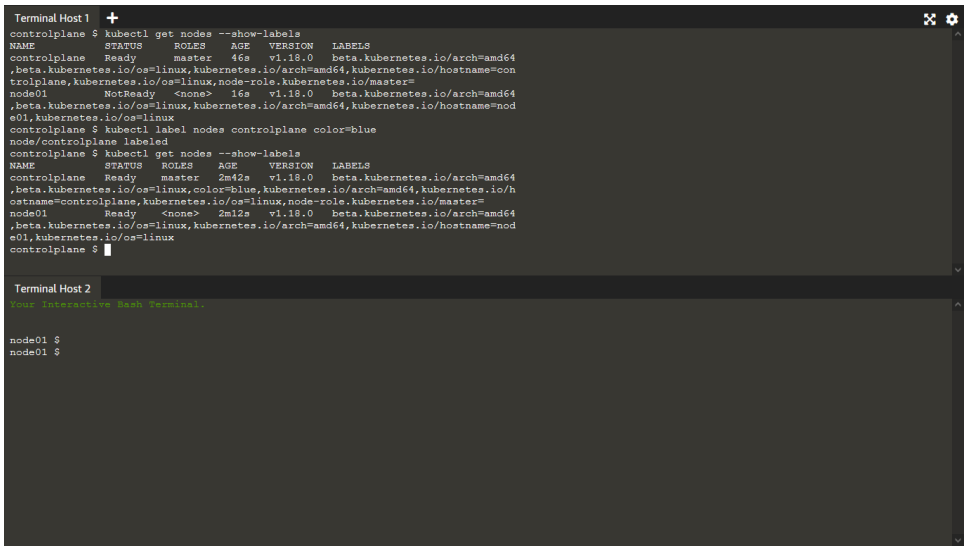
10-

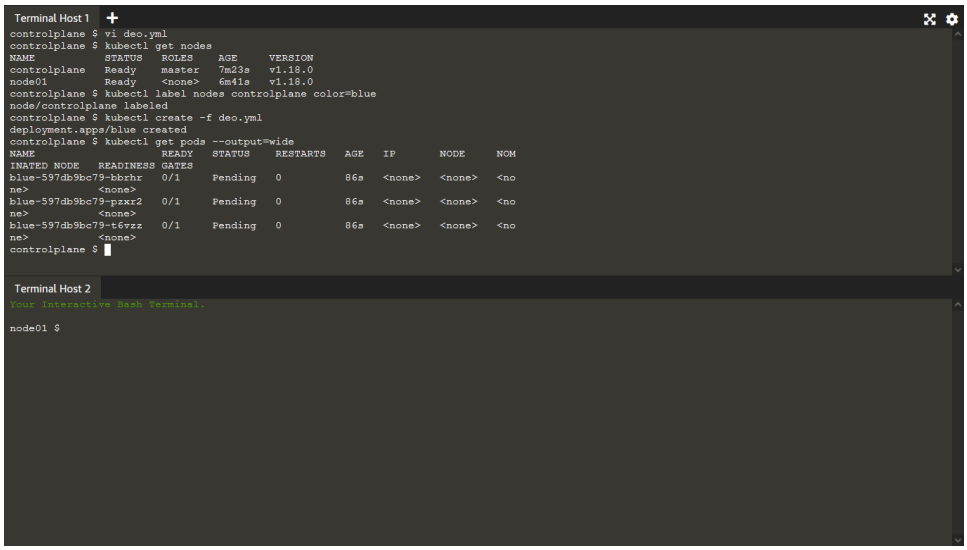

Terminal Host 1
memory 190Mi (10%) 390Mi (20%)
ephemeral-storage 0 (0%) 0 (0%)
hugepages-1Gi 0 (0%) 0 (0%)
hugepages-2Mi 0 (0%) 0 (0%)
Events:
Type Reason Age From
controlplane $ kubectl label no controlplane color=blue
node/controlplane labeled
controlplane $
Terminal Host 2
Your Interactive Bash Terminal.
node01 $
node01 $

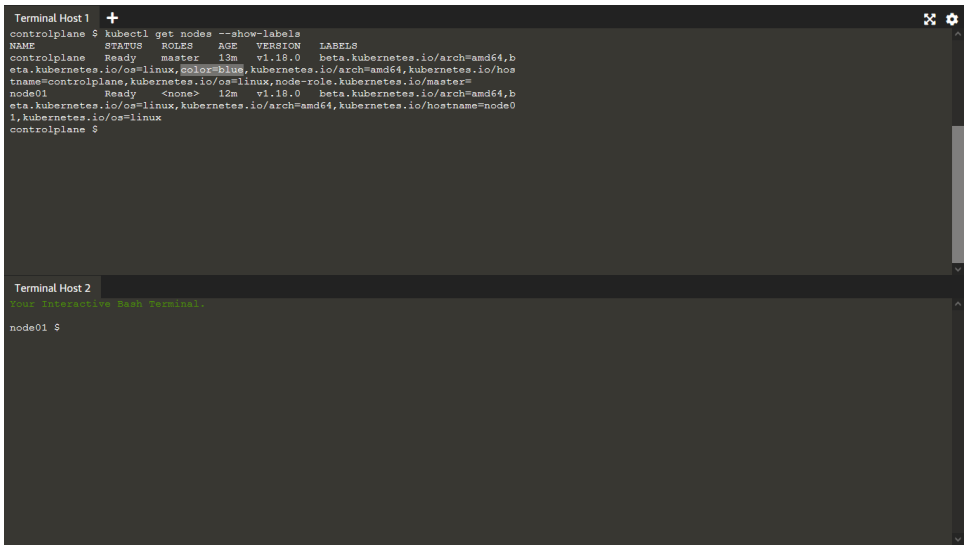
11-

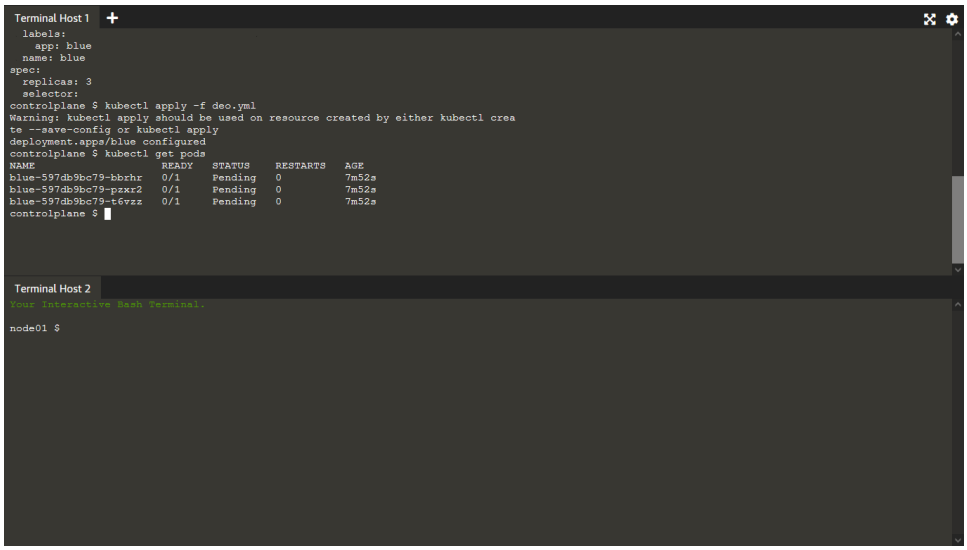

Terminal Host 1
controlplane $ kubectl label no controlplane color=blue
node/controlplane labeled
controlplane $ kubectl get no controlplane --show-lables
Error: unknown flag: --show-lables
See 'kubectl get --help' for usage.
controlplane $ kubectl get no controlplane --show-labels
NAME STATUS ROLES AGE VERSION LABELS
controlplane Ready master 54m v1.18.0 beta.kubernetes.io/arch=amd64,b
eta.kubernetes.io/os=linux,color=blue,kubernetes.io/arch=amd64,kubernetes.io/hos
tname=controlplane,kubernetes.io/os=linux,node-role.kubernetes.io/master=
controlplane $
Terminal Host 2
Your Interactive Bash Terminal.
node01 $
node01 $


Terminal Host 1
controlplane $ kubectl get no controlplane --show-labels | grep -i color
controlplane   Ready   master   56m   v1.18.0   beta.kubernetes.io/arch=amd64,b
eta.kubernetes.io/os=linux,color=blue,kubernetes.io/arch=amd64,kubernetes.io/hos
tname=controlplane,kubernetes.io/os=linux,node-role.kubernetes.io/master=
controlplane $
Terminal Host 2
Your Interactive Bash Terminal.
node01 $
node01 $

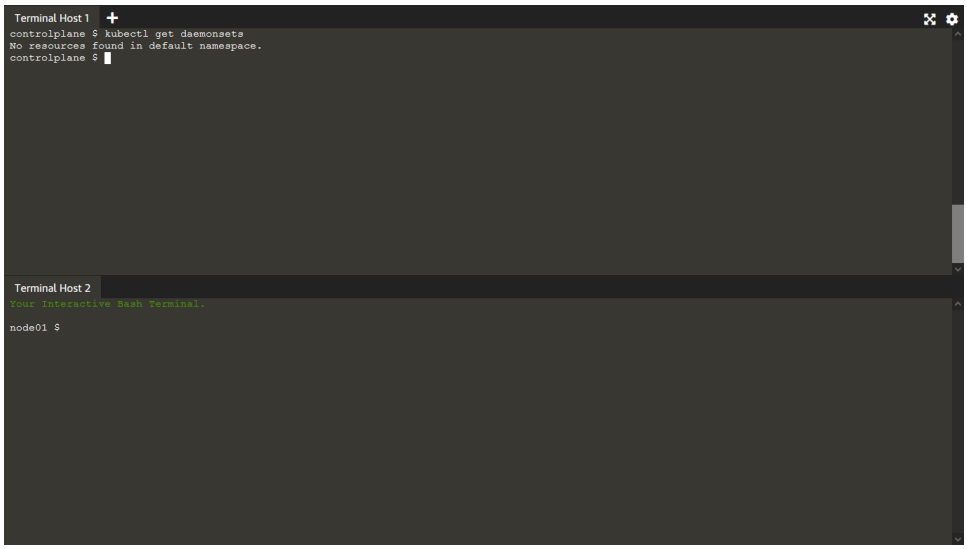

Terminal Host 1
controlplane $ kubectl get nodes --show-labels
NAME   STATUS   ROLES   AGE   VERSION   LABELS
controlplane   Ready   master   46s   v1.18.0   beta.kubernetes.io/arch=amd64
,beta.kubernetes.io/os=linux,kubernetes.io/arch=amd64,kubernetes.io/hostname=con
trolplane,kubernetes.io/os=linux,node-role.kubernetes.io/master=
node01   NotReady   <none>   16s   v1.18.0   beta.kubernetes.io/arch=amd64
,beta.kubernetes.io/os=linux,kubernetes.io/arch=amd64,kubernetes.io/hostname=nod
e01,kubernetes.io/os=linux
controlplane $ kubectl label nodes controlplane color=blue
node/controlplane labeled
controlplane $ kubectl get nodes --show-labels
NAME   STATUS   ROLES   AGE   VERSION   LABELS
controlplane   Ready   master   2m42s   v1.18.0   beta.kubernetes.io/arch=amd64
,beta.kubernetes.io/os=linux,color=blue,kubernetes.io/arch=amd64,kubernetes.io/h
ostname=controlplane,kubernetes.io/os=linux,node-role.kubernetes.io/master=
node01   Ready   <none>   2m12s   v1.18.0   beta.kubernetes.io/arch=amd64
,beta.kubernetes.io/os=linux,kubernetes.io/arch=amd64,kubernetes.io/hostname=nod
e01,kubernetes.io/os=linux
controlplane $
Terminal Host 2
Your Interactive Bash Terminal.
node01 $
node01 $


Terminal Host 1
controlplane $ vi deo.yml
controlplane $ kubectl get nodes
NAME   STATUS   ROLES   AGE   VERSION
controlplane   Ready   master   7m23s   v1.18.0
node01   Ready   <none>   6m41s   v1.18.0
controlplane $ kubectl label nodes controlplane color=blue
node/controlplane labeled
controlplane $ kubectl create -f deo.yml
deployment.apps/blue created
controlplane $ kubectl get pods --output=wide
NAME   READY   STATUS   RESTARTS   AGE   IP   NODE   NOM
INATED NODE   READINESS GATES
blue-597db9bc79-bbrhr   0/1   Pending   0   86s   <none>   <none>   <no
ne>   <none>
blue-597db9bc79-pzxr2   0/1   Pending   0   86s   <none>   <none>   <no
ne>   <none>
blue-597db9bc79-t6vzz   0/1   Pending   0   86s   <none>   <none>   <no
ne>   <none>
controlplane $
Terminal Host 2
Your Interactive Bash Terminal.
node01 $

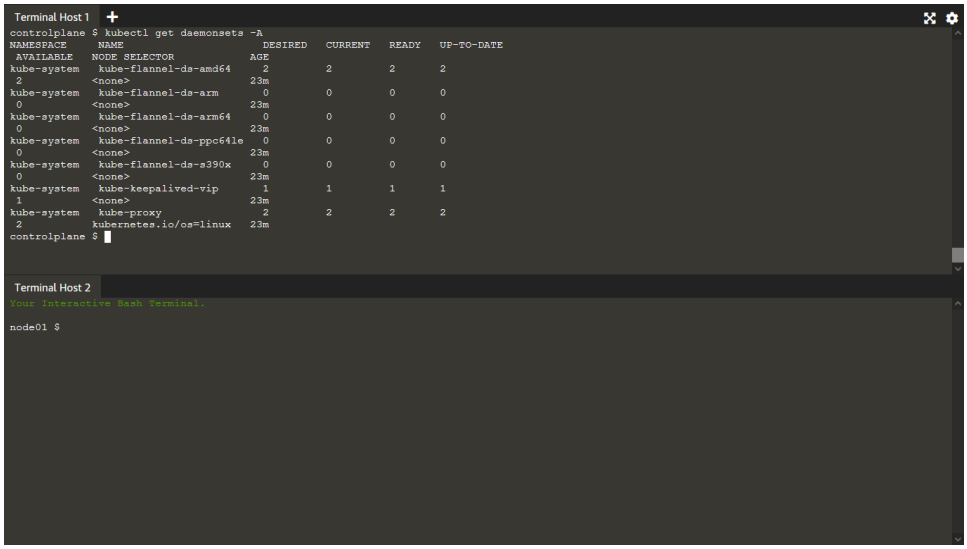

Terminal Host 1
controlplane $ kubectl get nodes --show-labels
NAME STATUS ROLES AGE VERSION LABELS
controlplane Ready master 13m v1.18.0 beta.kubernetes.io/arch=amd64,b
eta.kubernetes.io/os=linux,color=blue,kubernetes.io/arch=amd64,kubernetes.io/hos
tname=controlplane,kubernetes.io/os=linux,node-role.kubernetes.io/master=
node01 Ready <none> 12m v1.18.0 beta.kubernetes.io/arch=amd64,b
eta.kubernetes.io/os=linux,kubernetes.io/arch=amd64,kubernetes.io/hostname=node0
1,kubernetes.io/os=linux
controlplane $
Terminal Host 2
Your Interactive Bash Terminal.
node01 $


Terminal Host 1
labels:
app: blue
name: blue
spec:
replicas: 3
selector:
controlplane $ kubectl apply -f deo.yml
Warning: kubectl apply should be used on resource created by either kubectl crea
te --save-config or kubectl apply
deployment.apps/blue configured
controlplane $ kubectl get pods
NAME READY STATUS RESTARTS AGE
blue-597db9bc79-bbrhr 0/1 Pending 0 7m52s
blue-597db9bc79-pzxr2 0/1 Pending 0 7m52s
blue-597db9bc79-t6vzz 0/1 Pending 0 7m52s
controlplane $
Terminal Host 2
Your Interactive Bash Terminal.
node01 $

12-


Terminal Host 1
controlplane $ kubectl get daemonsets
No resources found in default namespace.
controlplane $
Terminal Host 2
Your Interactive Bash Terminal.
node01 $

13-

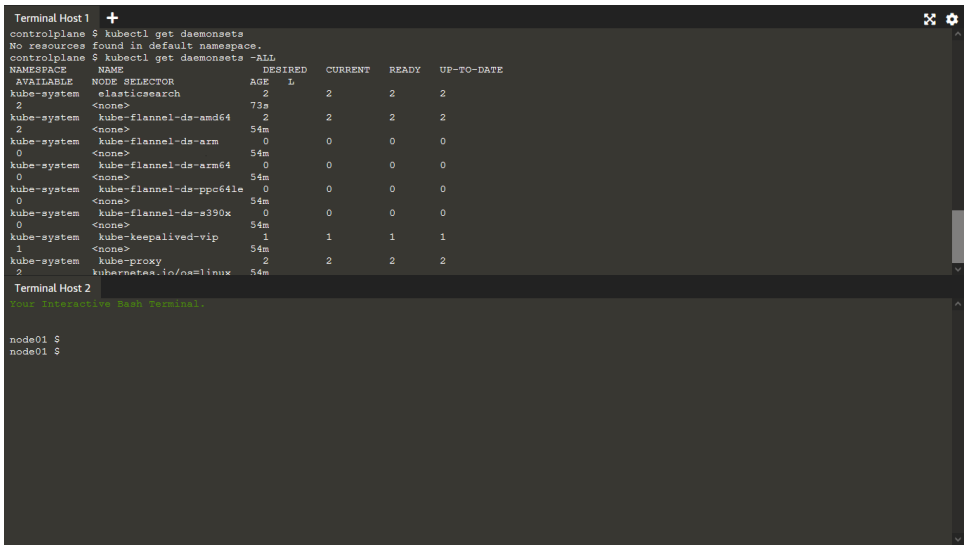

Terminal Host 1
controlplane $ kubectl get daemonsets -A
NAMESPACE NAME DESIRED CURRENT READY UP-TO-DATE
AVAILABLE NODE SELECTOR AGE
kube-system kube-flannel-ds-amd64 2 2 2 2
2 <none> 23m
kube-system kube-flannel-ds-arm 0 0 0 0
0 <none> 23m
kube-system kube-flannel-ds-arm64 0 0 0 0
0 <none> 23m
kube-system kube-flannel-ds-ppc64le 0 0 0 0
0 <none> 23m
kube-system kube-flannel-ds-s390x 0 0 0 0
0 <none> 23m
kube-system kube-keepalived-vip 1 1 1 1
1 <none> 23m
kube-system kube-proxy 2 2 2 2
2 kubernetes.io/os=linux 23m
controlplane $
Terminal Host 2
Your Interactive Bash Terminal.
node01 $

14-

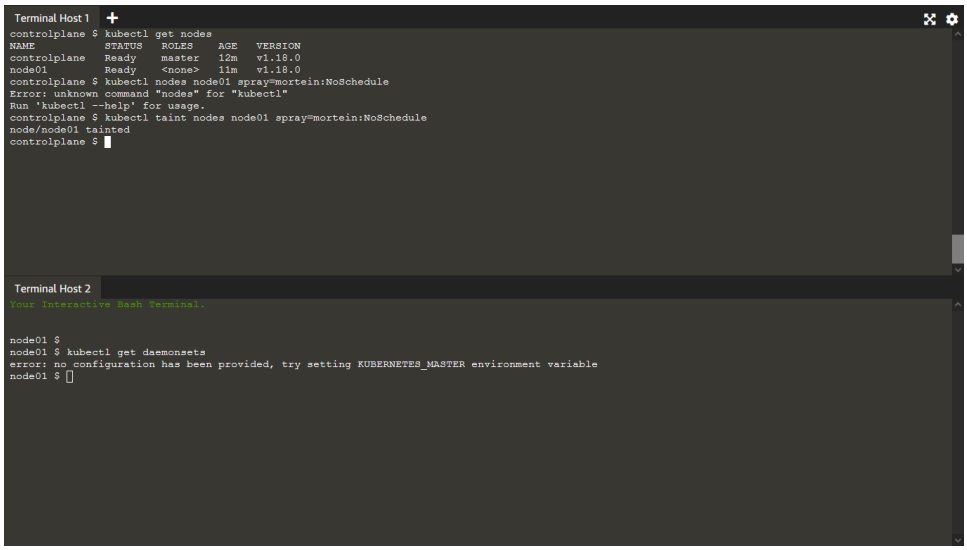
```
Terminal Host 1
controlplane $ kubectl get namespaces
NAME              STATUS   AGE
default           Active   11m
kube-node-lease   Active   11m
kube-public       Active   11m
kube-system       Active   11m
controlplane $ kubectl get daemonsets -A
NAMESPACE     NAME                         DESIRED   CURRENT   READY   UP-TO-DATE
 AVAILABLE   NODE SELECTOR            AGE
kube-system   kube-flannel-ds-amd64        2         2         2       2
 2           <none>                   13m
kube-system   kube-flannel-ds-arm          0         0         0       0
 0           <none>                   13m
kube-system   kube-flannel-ds-arm64        0         0         0       0
 0           <none>                   13m
kube-system   kube-flannel-ds-ppc64le      0         0         0       0
 0           <none>                   13m
kube-system   kube-flannel-ds-s390x        0         0         0       0
 0           <none>                   13m
kube-system   kube-keepalived-vip          1         1         1       1
 1           <none>                   13m
Terminal Host 2
Your Interactive Bash Terminal.


node01 $
node01 $
```

15-

```
Terminal Host 1
apiVersion: apps/v1
kind: DaemonSet
metadata:
  name: elasticsearch
  namespace: kube-system
  labels:
    k8s-app: elasticsearch
spec:
  selector:
    matchLabels:
      name: elasticsearch
  template:
    metadata:
      labels:
        name: elasticsearch
  containers:
      - name: elasticsearch
        image: k8s.gcr.io/fluentd-elasticsearch:1.20
~
Terminal Host 2
Your Interactive Bash Terminal.


node01 $
node01 $
```


Terminal Host 1
controlplane $ kubectl get daemonsets
No resources found in default namespace.
controlplane $ kubectl get daemonsets -ALL
NAMESPACE NAME DESIRED CURRENT READY UP-TO-DATE
AVAILABLE NODE SELECTOR AGE L
kube-system elasticsearch 2 2 2 2
2 <none> 73s
kube-system kube-flannel-ds-amd64 2 2 2 2
2 <none> 54m
kube-system kube-flannel-ds-arm 0 0 0 0
0 <none> 54m
kube-system kube-flannel-ds-arm64 0 0 0 0
0 <none> 54m
kube-system kube-flannel-ds-ppc64le 0 0 0 0
0 <none> 54m
kube-system kube-flannel-ds-s390x 0 0 0 0
0 <none> 54m
kube-system kube-keepalived-vip 1 1 1 1
1 <none> 54m
kube-system kube-proxy 2 2 2 2
2 kubernetes.io/os=linux 54m
Terminal Host 2
Your Interactive Bash Terminal.
node01 $
node01 $

16-


Terminal Host 1
controlplane $ kubectl get nodes
NAME STATUS ROLES AGE VERSION
controlplane Ready master 12m v1.18.0
node01 Ready <none> 11m v1.18.0
controlplane $ kubectl nodes node01 spray=mortein:NoSchedule
Error: unknown command "nodes" for "kubectl"
Run 'kubectl --help' for usage.
controlplane $ kubectl taint nodes node01 spray=mortein:NoSchedule
node/node01 tainted
controlplane $
Terminal Host 2
Your Interactive Bash Terminal.
node01 $
node01 $ kubectl get daemonsets
error: no configuration has been provided, try setting KUBERNETES_MASTER environment variable
node01 $

17&18-

Terminal Host 1

```
controlplane $ kubectl get nodes
NAME           STATUS   ROLES    AGE   VERSION
controlplane   Ready    master   12m   v1.18.0
node01         Ready    <none>   11m   v1.18.0
controlplane $ kubectl nodes node01 spray=mortein:NoSchedule
Error: unknown command "nodes" for "kubectl"
Run 'kubectl --help' for usage.
controlplane $ kubectl taint nodes node01 spray=mortein:NoSchedule
node/node01 tainted
controlplane $ kubectl run mosquito --image nginx
pod/mosquito created
controlplane $ kubectl get pods
NAME       READY   STATUS    RESTARTS   AGE
mosquito   0/1     Pending   0          14s
controlplane $
```

Terminal Host 2

```
Your Interactive Bash Terminal.


node01 $
node01 $ kubectl get daemonsets
error: no configuration has been provided, try setting KUBERNETES_MASTER environment variable
node01 $
```

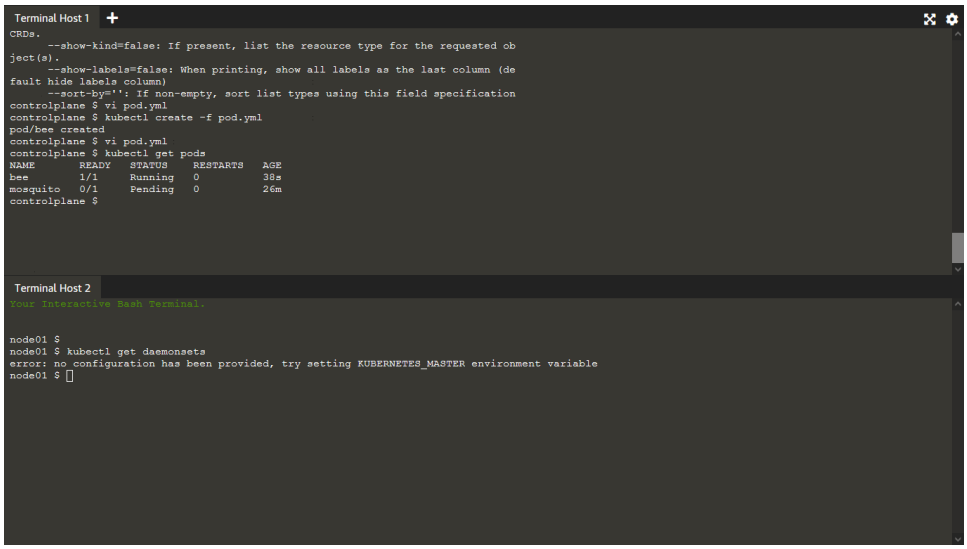
19-

Terminal Host 1

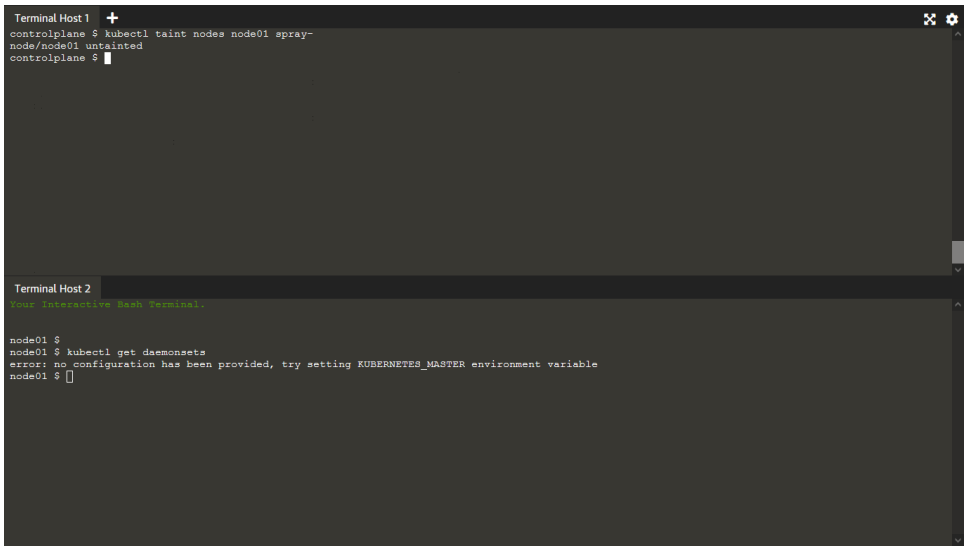
```
apiVersion: v1
kind: Pod
metadata:
  creationTimestamp: null
  labels:
    run: bee
  name: bee
spec:
  containers:
  - image: nginx
    name: bee
    resources: {}
  tolerations:
  - key: "spray"
    operator: "Equal"
    value: "mortein"
    effect: "NoSchedule"

~
```

Terminal Host 2

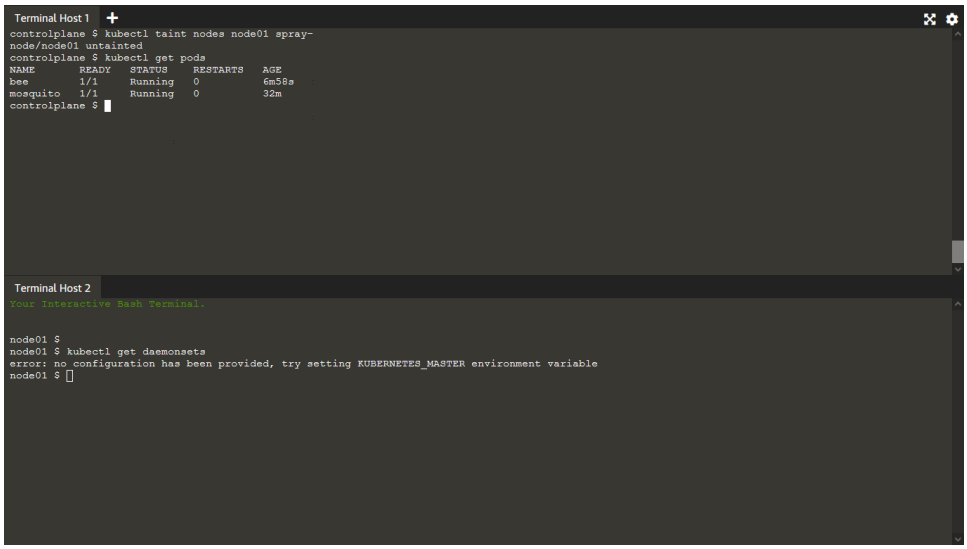
```
Your Interactive Bash Terminal.


node01 $
node01 $ kubectl get daemonsets
error: no configuration has been provided, try setting KUBERNETES_MASTER environment variable
node01 $
```


Terminal Host 1
CRDs.
--show-kind=false: If present, list the resource type for the requested ob
ject(s).
--show-labels=false: When printing, show all labels as the last column (de
fault hide labels column)
--sort-by='': If non-empty, sort list types using this field specification
controlplane $ vi pod.yml
controlplane $ kubectl create -f pod.yml
pod/bee created
controlplane $ vi pod.yml
controlplane $ kubectl get pods
NAME READY STATUS RESTARTS AGE
bee 1/1 Running 0 38s
mosquito 0/1 Pending 0 26m
controlplane $
Terminal Host 2
Your Interactive Bash Terminal.
node01 $
node01 $ kubectl get daemonsets
error: no configuration has been provided, try setting KUBERNETES_MASTER environment variable
node01 $

20-


Terminal Host 1
controlplane $ kubectl taint nodes node01 spray-
node/node01 untainted
controlplane $
Terminal Host 2
Your Interactive Bash Terminal.
node01 $
node01 $ kubectl get daemonsets
error: no configuration has been provided, try setting KUBERNETES_MASTER environment variable
node01 $

21-


Terminal Host 1
controlplane $ kubectl taint nodes node01 spray-
node/node01 untainted
controlplane $ kubectl get pods
NAME READY STATUS RESTARTS AGE
bee 1/1 Running 0 6m58s
mosquito 1/1 Running 0 32m
controlplane $
Terminal Host 2
Your Interactive Bash Terminal.
node01 $
node01 $ kubectl get daemonsets
error: no configuration has been provided, try setting KUBERNETES_MASTER environment variable
node01 $