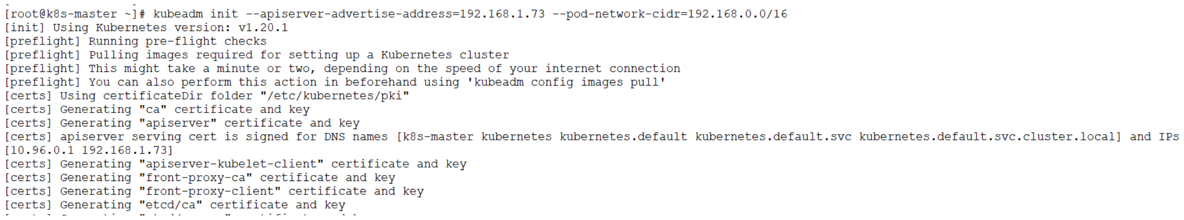
Network LAB

10 January 2021 12:04

Demo

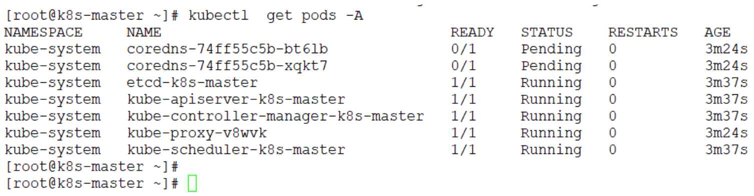
- If you were deploying on a Vmware environment NSX -T may be a good option.
- If you look at the play-with-k8s labs they use WeaveNet.
- In our demos in the course we used Calico.
- Depending on your environment and after evaluating the Pros and Cons of each of these, you may chose the right networking sol ution.

- **Reset your cluster:**



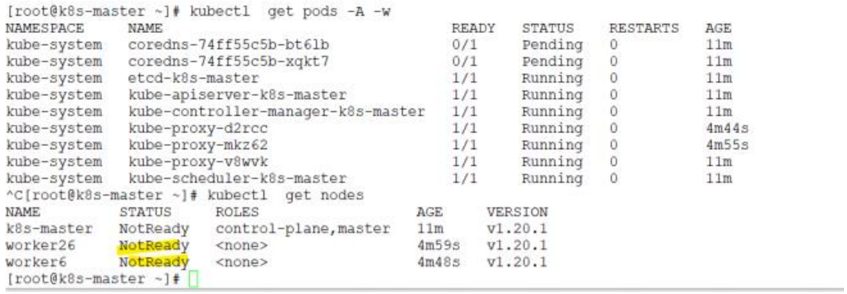
```
[root@k8s-master ~]# kubeadm init --apiserver-advertise-address=192.168.1.73 --pod-network-cidr=192.168.0.0/16
[init] Using Kubernetes version: v1.20.1
[preflight] Running pre-flight checks
[preflight] Pulling images required for setting up a Kubernetes cluster
[preflight] This might take a minute or two, depending on the speed of your internet connection
[preflight] You can also perform this action in beforehand using 'kubeadm config images pull'
[certs] Using certificateDir folder "/etc/kubernetes/pki"
[certs] Generating "ca" certificate and key
[certs] Generating "apiserver" certificate and key
[certs] apiserver serving cert is signed for DNS names [k8s-master kubernetes kubernetes.default kubernetes.default.svc kubernetes.default.svc.cluster.local] and IPs
[10.96.0.1 192.168.1.73]
[certs] Generating "apiserver-kubelet-client" certificate and key
[certs] Generating "front-proxy-ca" certificate and key
[certs] Generating "front-proxy-client" certificate and key
[certs] Generating "etcd/ca" certificate and key
```

- **Initialize Cluster**



```
[root@k8s-master ~]# kubectl  get pods -A
NAMESPACE     NAME                                 READY   STATUS    RESTARTS   AGE
kube-system   coredns-74ff55c5b-bt6lb              0/1     Pending   0          3m24s
kube-system   coredns-74ff55c5b-xqkt7              0/1     Pending   0          3m24s
kube-system   etcd-k8s-master                      1/1     Running   0          3m37s
kube-system   kube-apiserver-k8s-master            1/1     Running   0          3m37s
kube-system   kube-controller-manager-k8s-master   1/1     Running   0          3m37s
kube-system   kube-proxy-v8wvk                     1/1     Running   0          3m24s
kube-system   kube-scheduler-k8s-master            1/1     Running   0          3m37s
[root@k8s-master ~]#
[root@k8s-master ~]#
```

- **Join workers without installing CNI and notice status:**



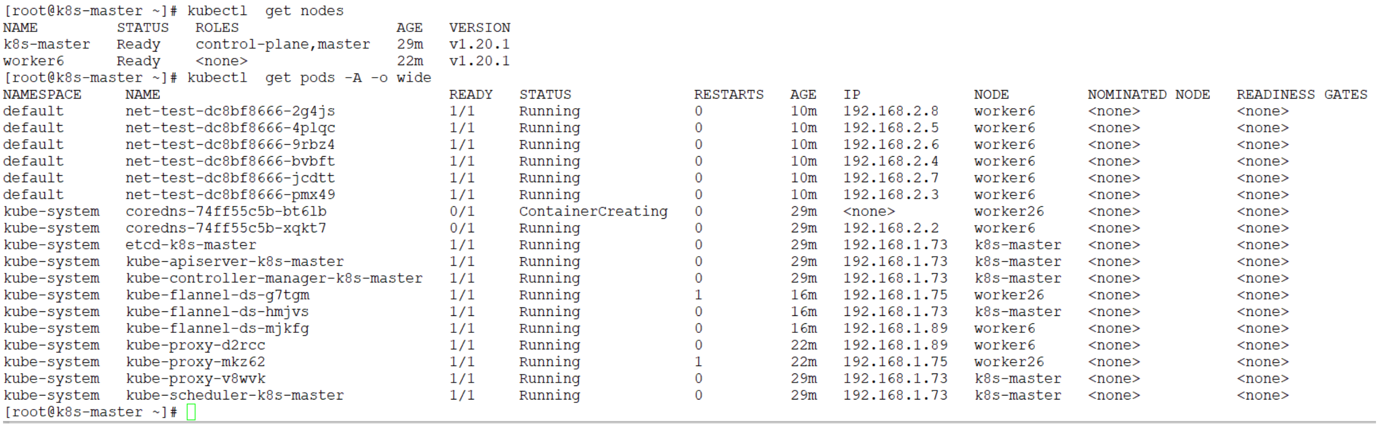
```
[root@k8s-master ~]# kubectl  get pods -A -w
NAMESPACE     NAME                                 READY   STATUS    RESTARTS   AGE
kube-system   coredns-74ff55c5b-bt6lb              0/1     Pending   0          11m
kube-system   coredns-74ff55c5b-xqkt7              0/1     Pending   0          11m
kube-system   etcd-k8s-master                      1/1     Running   0          11m
kube-system   kube-apiserver-k8s-master            1/1     Running   0          11m
kube-system   kube-controller-manager-k8s-master   1/1     Running   0          11m
kube-system   kube-proxy-d2rcc                     1/1     Running   0          4m44s
kube-system   kube-proxy-mkz62                     1/1     Running   0          4m55s
kube-system   kube-proxy-v8wvk                     1/1     Running   0          11m
kube-system   kube-scheduler-k8s-master            1/1     Running   0          11m
^C[root@k8s-master ~]# kubectl  get nodes
NAME         STATUS     ROLES                  AGE     VERSION
k8s-master   NotReady   control-plane,master   11m     v1.20.1
worker26     NotReady   <none>                 4m59s   v1.20.1
worker6      NotReady   <none>                 4m48s   v1.20.1
[root@k8s-master ~]#
```

- **Install network drivers**

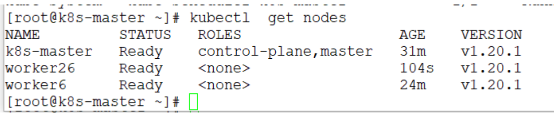
FLANNEL

kubectl apply -f https://raw.githubusercontent.com/coreos/flannel/master/Documentation/kube-flannel.yml

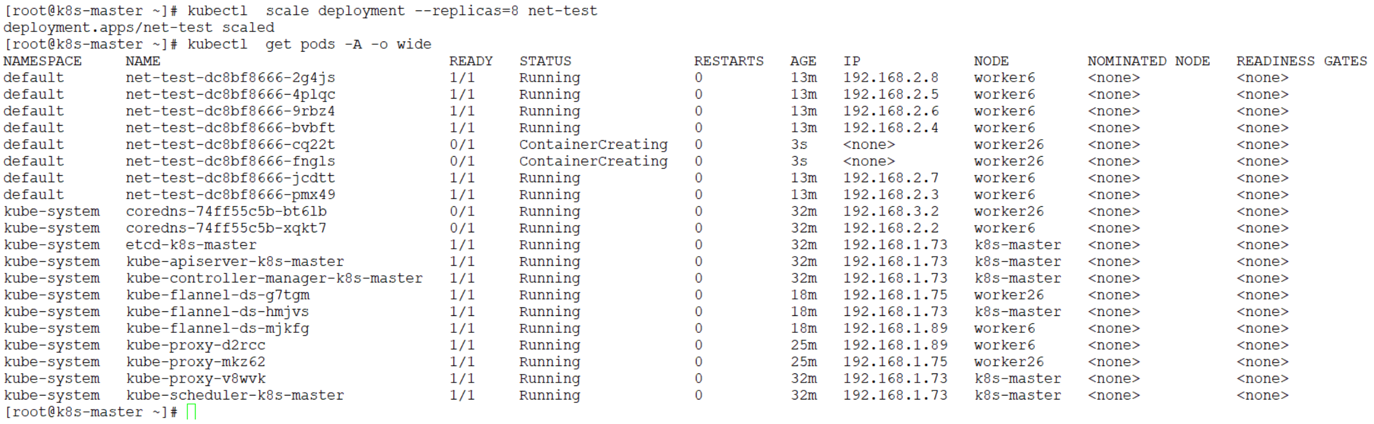
- **Kubernetes Scheduling:**



```
[root@k8s-master ~]# kubectl  get nodes
NAME         STATUS   ROLES                  AGE   VERSION
k8s-master   Ready    control-plane,master   29m   v1.20.1
worker6      Ready    <none>                 22m   v1.20.1
[root@k8s-master ~]# kubectl  get pods -A -o wide
NAMESPACE     NAME                                 READY   STATUS              RESTARTS   AGE   IP             NODE         NOMINATED NODE   READINESS GATES
default       net-test-dc8bf8666-2g4js             1/1     Running             0          10m   192.168.2.8    worker6      <none>           <none>
default       net-test-dc8bf8666-4plqc             1/1     Running             0          10m   192.168.2.5    worker6      <none>           <none>
default       net-test-dc8bf8666-9rbz4             1/1     Running             0          10m   192.168.2.6    worker6      <none>           <none>
default       net-test-dc8bf8666-bvbft             1/1     Running             0          10m   192.168.2.4    worker6      <none>           <none>
default       net-test-dc8bf8666-jcdtt             1/1     Running             0          10m   192.168.2.7    worker6      <none>           <none>
default       net-test-dc8bf8666-pmx49             1/1     Running             0          10m   192.168.2.3    worker6      <none>           <none>
kube-system   coredns-74ff55c5b-bt6lb              0/1     ContainerCreating   0          29m   <none>         worker26     <none>           <none>
kube-system   coredns-74ff55c5b-xqkt7              0/1     Running             0          29m   192.168.2.2    worker6      <none>           <none>
kube-system   etcd-k8s-master                      1/1     Running             0          29m   192.168.1.73   k8s-master   <none>           <none>
kube-system   kube-apiserver-k8s-master            1/1     Running             0          29m   192.168.1.73   k8s-master   <none>           <none>
kube-system   kube-controller-manager-k8s-master   1/1     Running             0          29m   192.168.1.73   k8s-master   <none>           <none>
kube-system   kube-flannel-ds-g7tgm                1/1     Running             1          16m   192.168.1.75   worker26     <none>           <none>
kube-system   kube-flannel-ds-hmjvs                1/1     Running             0          16m   192.168.1.73   k8s-master   <none>           <none>
kube-system   kube-flannel-ds-mjkfg                1/1     Running             0          16m   192.168.1.89   worker6      <none>           <none>
kube-system   kube-proxy-d2rcc                     1/1     Running             0          22m   192.168.1.89   worker6      <none>           <none>
kube-system   kube-proxy-mkz62                     1/1     Running             1          22m   192.168.1.75   worker26     <none>           <none>
kube-system   kube-proxy-v8wvk                     1/1     Running             0          29m   192.168.1.73   k8s-master   <none>           <none>
kube-system   kube-scheduler-k8s-master            1/1     Running             0          29m   192.168.1.73   k8s-master   <none>           <none>
[root@k8s-master ~]#
```



```
[root@k8s-master ~]# kubectl  get nodes
NAME         STATUS   ROLES                  AGE    VERSION
k8s-master   Ready    control-plane,master   31m    v1.20.1
worker26     Ready    <none>                 104s   v1.20.1
worker6      Ready    <none>                 24m    v1.20.1
[root@k8s-master ~]#
```



```
[root@k8s-master ~]# kubectl  scale deployment --replicas=8 net-test
deployment.apps/net-test scaled
[root@k8s-master ~]# kubectl  get pods -A -o wide
NAMESPACE     NAME                                 READY   STATUS              RESTARTS   AGE   IP             NODE         NOMINATED NODE   READINESS GATES
default       net-test-dc8bf8666-2g4js             1/1     Running             0          13m   192.168.2.8    worker6      <none>           <none>
default       net-test-dc8bf8666-4plqc             1/1     Running             0          13m   192.168.2.5    worker6      <none>           <none>
default       net-test-dc8bf8666-9rbz4             1/1     Running             0          13m   192.168.2.6    worker6      <none>           <none>
default       net-test-dc8bf8666-bvbft             1/1     Running             0          13m   192.168.2.4    worker6      <none>           <none>
default       net-test-dc8bf8666-cq22t             0/1     ContainerCreating   0          3s    <none>         worker26     <none>           <none>
default       net-test-dc8bf8666-fngls             0/1     ContainerCreating   0          3s    <none>         worker26     <none>           <none>
default       net-test-dc8bf8666-jcdtt             1/1     Running             0          13m   192.168.2.7    worker6      <none>           <none>
default       net-test-dc8bf8666-pmx49             1/1     Running             0          13m   192.168.2.3    worker6      <none>           <none>
kube-system   coredns-74ff55c5b-bt6lb              0/1     Running             0          32m   192.168.3.2    worker26     <none>           <none>
kube-system   coredns-74ff55c5b-xqkt7              0/1     Running             0          32m   192.168.2.2    worker6      <none>           <none>
kube-system   etcd-k8s-master                      1/1     Running             0          32m   192.168.1.73   k8s-master   <none>           <none>
kube-system   kube-apiserver-k8s-master            1/1     Running             0          32m   192.168.1.73   k8s-master   <none>           <none>
kube-system   kube-controller-manager-k8s-master   1/1     Running             0          32m   192.168.1.73   k8s-master   <none>           <none>
kube-system   kube-flannel-ds-g7tgm                1/1     Running             0          18m   192.168.1.75   worker26     <none>           <none>
kube-system   kube-flannel-ds-hmjvs                1/1     Running             0          18m   192.168.1.73   k8s-master   <none>           <none>
kube-system   kube-flannel-ds-mjkfg                1/1     Running             0          18m   192.168.1.89   worker6      <none>           <none>
kube-system   kube-proxy-d2rcc                     1/1     Running             0          25m   192.168.1.89   worker6      <none>           <none>
kube-system   kube-proxy-mkz62                     1/1     Running             0          25m   192.168.1.75   worker26     <none>           <none>
kube-system   kube-proxy-v8wvk                     1/1     Running             0          32m   192.168.1.73   k8s-master   <none>           <none>
kube-system   kube-scheduler-k8s-master            1/1     Running             0          32m   192.168.1.73   k8s-master   <none>           <none>
[root@k8s-master ~]#
```

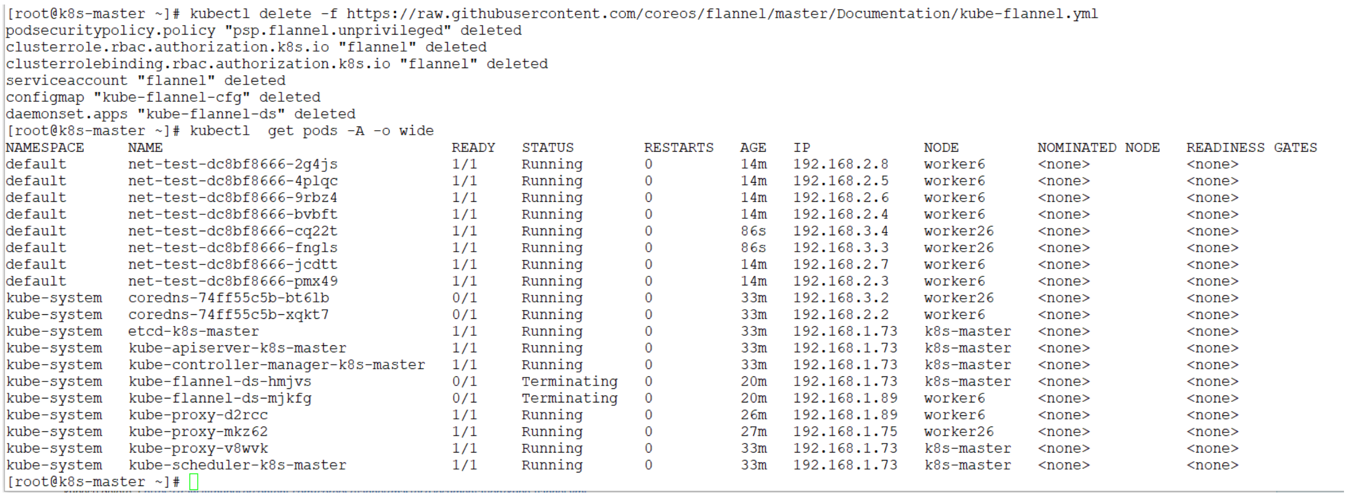
Change the Network:

CALICO

- **Delete flannel**

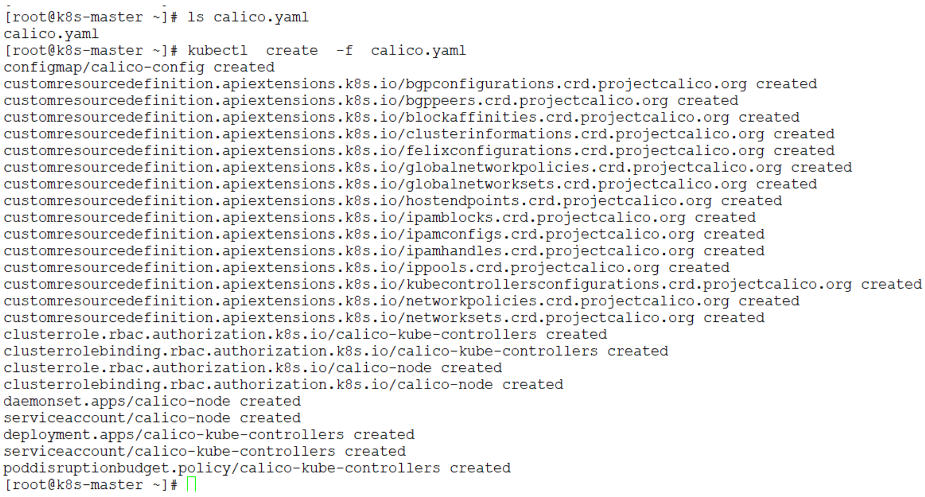
  kubectl delete -f https://raw.githubusercontent.com/coreos/flannel/master/Documentation/kube-flannel.yml

- **Check status of pods**



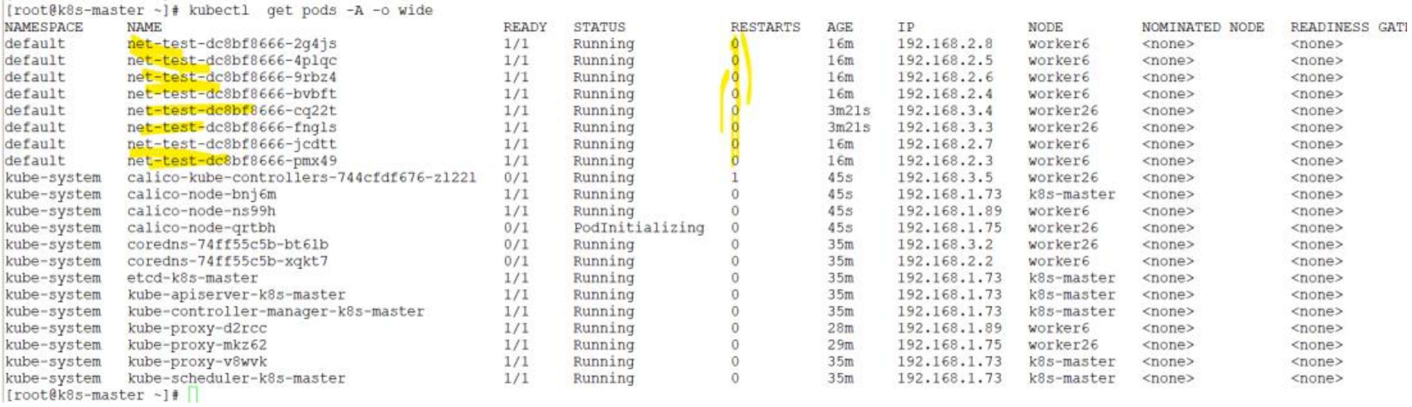
```
[root@k8s-master ~]# kubectl delete -f https://raw.githubusercontent.com/coreos/flannel/master/Documentation/kube-flannel.yml
podsecuritypolicy.policy "psp.flannel.unprivileged" deleted
clusterrole.rbac.authorization.k8s.io "flannel" deleted
clusterrolebinding.rbac.authorization.k8s.io "flannel" deleted
serviceaccount "flannel" deleted
configmap "kube-flannel-cfg" deleted
daemonset.apps "kube-flannel-ds" deleted
[root@k8s-master ~]# kubectl  get pods -A -o wide
NAMESPACE     NAME                                 READY   STATUS        RESTARTS   AGE   IP             NODE         NOMINATED NODE   READINESS GATES
default       net-test-dc8bf8666-2g4js             1/1     Running       0          14m   192.168.2.8    worker6      <none>           <none>
default       net-test-dc8bf8666-4plqc             1/1     Running       0          14m   192.168.2.5    worker6      <none>           <none>
default       net-test-dc8bf8666-9rbz4             1/1     Running       0          14m   192.168.2.6    worker6      <none>           <none>
default       net-test-dc8bf8666-bvbft             1/1     Running       0          14m   192.168.2.4    worker6      <none>           <none>
default       net-test-dc8bf8666-cq22t             1/1     Running       0          86s   192.168.3.4    worker26     <none>           <none>
default       net-test-dc8bf8666-fngls             1/1     Running       0          86s   192.168.3.3    worker26     <none>           <none>
default       net-test-dc8bf8666-jcdtt             1/1     Running       0          14m   192.168.2.7    worker6      <none>           <none>
default       net-test-dc8bf8666-pmx49             1/1     Running       0          14m   192.168.2.3    worker6      <none>           <none>
kube-system   coredns-74ff55c5b-bt6lb              0/1     Running       0          33m   192.168.3.2    worker26     <none>           <none>
kube-system   coredns-74ff55c5b-xqkt7              0/1     Running       0          33m   192.168.2.2    worker6      <none>           <none>
kube-system   etcd-k8s-master                      1/1     Running       0          33m   192.168.1.73   k8s-master   <none>           <none>
kube-system   kube-apiserver-k8s-master            1/1     Running       0          33m   192.168.1.73   k8s-master   <none>           <none>
kube-system   kube-controller-manager-k8s-master   1/1     Running       0          33m   192.168.1.73   k8s-master   <none>           <none>
kube-system   kube-flannel-ds-hmjvs                0/1     Terminating   0          20m   192.168.1.73   k8s-master   <none>           <none>
kube-system   kube-flannel-ds-mjkfg                0/1     Terminating   0          20m   192.168.1.89   worker6      <none>           <none>
kube-system   kube-proxy-d2rcc                     1/1     Running       0          26m   192.168.1.89   worker6      <none>           <none>
kube-system   kube-proxy-mkz62                     1/1     Running       0          27m   192.168.1.75   worker26     <none>           <none>
kube-system   kube-proxy-v8wvk                     1/1     Running       0          33m   192.168.1.73   k8s-master   <none>           <none>
kube-system   kube-scheduler-k8s-master            1/1     Running       0          33m   192.168.1.73   k8s-master   <none>           <none>
[root@k8s-master ~]#
```

- **Install Calico**



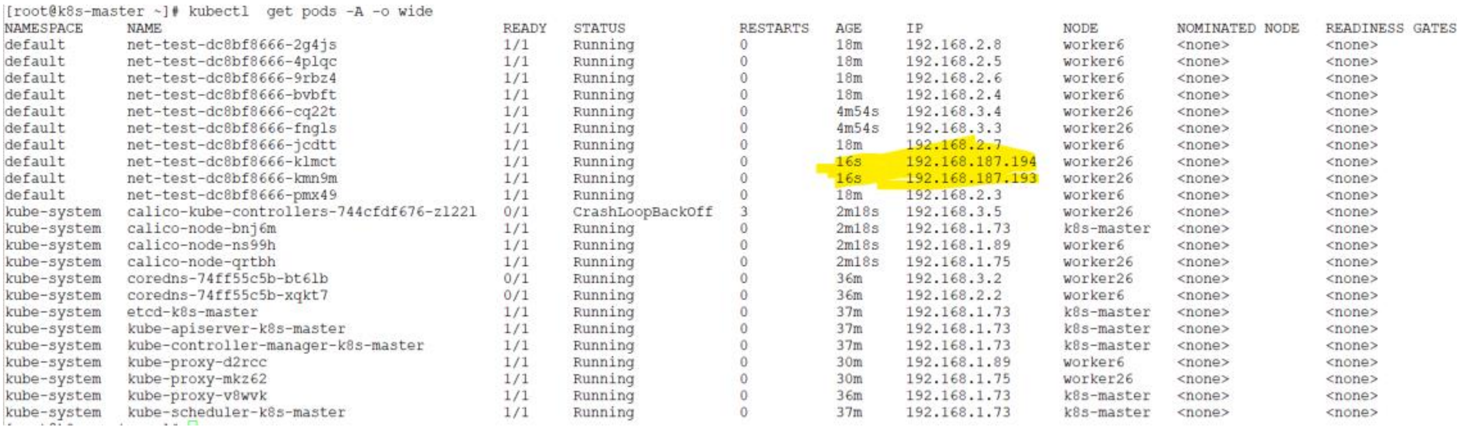
```
[root@k8s-master ~]# ls calico.yaml
calico.yaml
[root@k8s-master ~]# kubectl  create  -f  calico.yaml
configmap/calico-config created
customresourcedefinition.apiextensions.k8s.io/bgpconfigurations.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/bgppeers.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/blockaffinities.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/clusterinformations.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/felixconfigurations.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/globalnetworkpolicies.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/globalnetworksets.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/hostendpoints.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/ipamblocks.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/ipamconfigs.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/ipamhandles.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/ippools.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/kubecontrollersconfigurations.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/networkpolicies.crd.projectcalico.org created
customresourcedefinition.apiextensions.k8s.io/networksets.crd.projectcalico.org created
clusterrole.rbac.authorization.k8s.io/calico-kube-controllers created
clusterrolebinding.rbac.authorization.k8s.io/calico-kube-controllers created
clusterrole.rbac.authorization.k8s.io/calico-node created
clusterrolebinding.rbac.authorization.k8s.io/calico-node created
daemonset.apps/calico-node created
serviceaccount/calico-node created
deployment.apps/calico-kube-controllers created
serviceaccount/calico-kube-controllers created
poddisruptionbudget.policy/calico-kube-controllers created
[root@k8s-master ~]#
```

- **Check status of Pods**



```
[root@k8s-master ~]# kubectl  get pods -A -o wide
NAMESPACE     NAME                                       READY   STATUS            RESTARTS   AGE     IP             NODE         NOMINATED NODE   READINESS GATI
default       net-test-dc8bf8666-2g4js                   1/1     Running           0          16m     192.168.2.8    worker6      <none>           <none>
default       net-test-dc8bf8666-4plqc                   1/1     Running           0          16m     192.168.2.5    worker6      <none>           <none>
default       net-test-dc8bf8666-9rbz4                   1/1     Running           0          16m     192.168.2.6    worker6      <none>           <none>
default       net-test-dc8bf8666-bvbft                   1/1     Running           0          16m     192.168.2.4    worker6      <none>           <none>
default       net-test-dc8bf8666-cq22t                   1/1     Running           0          3m21s   192.168.3.4    worker26     <none>           <none>
default       net-test-dc8bf8666-fngls                   1/1     Running           0          3m21s   192.168.3.3    worker26     <none>           <none>
default       net-test-dc8bf8666-jcdtt                   1/1     Running           0          16m     192.168.2.7    worker6      <none>           <none>
default       net-test-dc8bf8666-pmx49                   1/1     Running           0          16m     192.168.2.3    worker6      <none>           <none>
kube-system   calico-kube-controllers-744cfdf676-zl221   0/1     Running           1          45s     192.168.3.5    worker26     <none>           <none>
kube-system   calico-node-bnj6m                          1/1     Running           0          45s     192.168.1.73   k8s-master   <none>           <none>
kube-system   calico-node-ns99h                          1/1     Running           0          45s     192.168.1.89   worker6      <none>           <none>
kube-system   calico-node-qrtbh                          0/1     PodInitializing   0          45s     192.168.1.75   worker26     <none>           <none>
kube-system   coredns-74ff55c5b-bt6lb                    0/1     Running           0          35m     192.168.3.2    worker26     <none>           <none>
kube-system   coredns-74ff55c5b-xqkt7                    0/1     Running           0          35m     192.168.2.2    worker6      <none>           <none>
kube-system   etcd-k8s-master                            1/1     Running           0          35m     192.168.1.73   k8s-master   <none>           <none>
kube-system   kube-apiserver-k8s-master                  1/1     Running           0          35m     192.168.1.73   k8s-master   <none>           <none>
kube-system   kube-controller-manager-k8s-master         1/1     Running           0          35m     192.168.1.73   k8s-master   <none>           <none>
kube-system   kube-proxy-d2rcc                           1/1     Running           0          28m     192.168.1.89   worker6      <none>           <none>
kube-system   kube-proxy-mkz62                           1/1     Running           0          29m     192.168.1.75   worker26     <none>           <none>
kube-system   kube-proxy-v8wvk                           1/1     Running           0          35m     192.168.1.73   k8s-master   <none>           <none>
kube-system   kube-scheduler-k8s-master                  1/1     Running           0          35m     192.168.1.73   k8s-master   <none>           <none>
[root@k8s-master ~]#
```

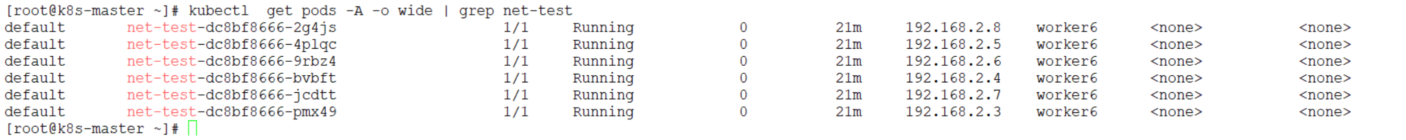
- **Scale up your deployment**
- **Check IP address of new pods**



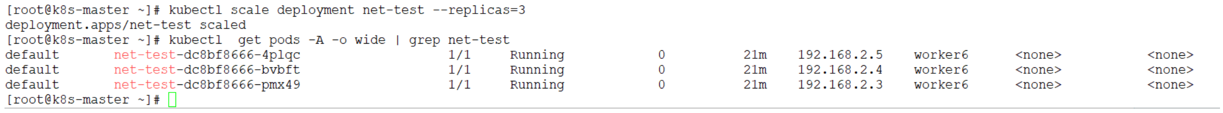
```
[root@k8s-master ~]# kubectl  get pods -A -o wide
NAMESPACE     NAME                                       READY   STATUS             RESTARTS   AGE     IP                NODE         NOMINATED NODE   READINESS GATES
default       net-test-dc8bf8666-2g4js                   1/1     Running            0          18m     192.168.2.8       worker6      <none>           <none>
default       net-test-dc8bf8666-4plqc                   1/1     Running            0          18m     192.168.2.5       worker6      <none>           <none>
default       net-test-dc8bf8666-9rbz4                   1/1     Running            0          18m     192.168.2.6       worker6      <none>           <none>
default       net-test-dc8bf8666-bvbft                   1/1     Running            0          18m     192.168.2.4       worker6      <none>           <none>
default       net-test-dc8bf8666-cq22t                   1/1     Running            0          4m54s   192.168.3.4       worker26     <none>           <none>
default       net-test-dc8bf8666-fngls                   1/1     Running            0          4m54s   192.168.3.3       worker26     <none>           <none>
default       net-test-dc8bf8666-jcdtt                   1/1     Running            0          18m     192.168.2.7       worker6      <none>           <none>
default       net-test-dc8bf8666-klmct                   1/1     Running            0          16s     192.168.187.194   worker26     <none>           <none>
default       net-test-dc8bf8666-kmn9m                   1/1     Running            0          16s     192.168.187.193   worker26     <none>           <none>
default       net-test-dc8bf8666-pmx49                   1/1     Running            0          18m     192.168.2.3       worker6      <none>           <none>
kube-system   calico-kube-controllers-744cfdf676-zl221   0/1     CrashLoopBackOff   3          2m18s   192.168.3.5       worker26     <none>           <none>
kube-system   calico-node-bnj6m                          1/1     Running            0          2m18s   192.168.1.73      k8s-master   <none>           <none>
kube-system   calico-node-ns99h                          1/1     Running            0          2m18s   192.168.1.89      worker6      <none>           <none>
kube-system   calico-node-qrtbh                          1/1     Running            0          2m18s   192.168.1.75      worker26     <none>           <none>
kube-system   coredns-74ff55c5b-bt6lb                    0/1     Running            0          36m     192.168.3.2       worker26     <none>           <none>
kube-system   coredns-74ff55c5b-xqkt7                    0/1     Running            0          36m     192.168.2.2       worker6      <none>           <none>
kube-system   etcd-k8s-master                            1/1     Running            0          37m     192.168.1.73      k8s-master   <none>           <none>
kube-system   kube-apiserver-k8s-master                  1/1     Running            0          37m     192.168.1.73      k8s-master   <none>           <none>
kube-system   kube-controller-manager-k8s-master         1/1     Running            0          37m     192.168.1.73      k8s-master   <none>           <none>
kube-system   kube-proxy-d2rcc                           1/1     Running            0          30m     192.168.1.89      worker6      <none>           <none>
kube-system   kube-proxy-mkz62                           1/1     Running            0          30m     192.168.1.75      worker26     <none>           <none>
kube-system   kube-proxy-v8wvk                           1/1     Running            0          36m     192.168.1.73      k8s-master   <none>           <none>
kube-system   kube-scheduler-k8s-master                  1/1     Running            0          37m     192.168.1.73      k8s-master   <none>           <none>
```

- **Scale down replica**

]# kubectl scale deployment net-test --replicas=6
deployment.apps/net-test scaled

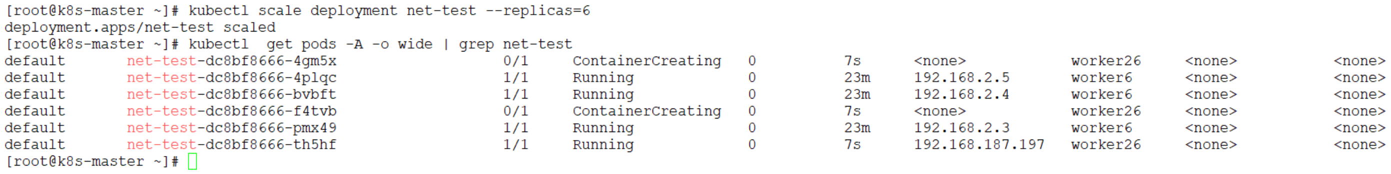


```
[root@k8s-master ~]# kubectl  get pods -A -o wide | grep net-test
default       net-test-dc8bf8666-2g4js                   1/1     Running            0          21m     192.168.2.8     worker6    <none>           <none>
default       net-test-dc8bf8666-4plqc                   1/1     Running            0          21m     192.168.2.5     worker6    <none>           <none>
default       net-test-dc8bf8666-9rbz4                   1/1     Running            0          21m     192.168.2.6     worker6    <none>           <none>
default       net-test-dc8bf8666-bvbft                   1/1     Running            0          21m     192.168.2.4     worker6    <none>           <none>
default       net-test-dc8bf8666-jcdtt                   1/1     Running            0          21m     192.168.2.7     worker6    <none>           <none>
default       net-test-dc8bf8666-pmx49                   1/1     Running            0          21m     192.168.2.3     worker6    <none>           <none>
[root@k8s-master ~]#
```



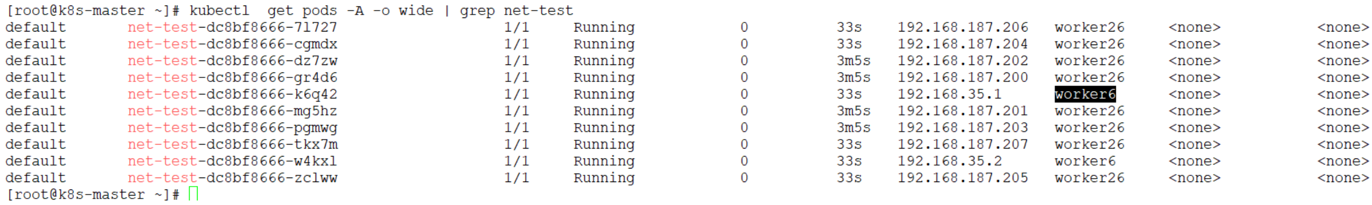
```
[root@k8s-master ~]# kubectl scale deployment net-test --replicas=3
deployment.apps/net-test scaled
[root@k8s-master ~]# kubectl  get pods -A -o wide | grep net-test
default       net-test-dc8bf8666-4plqc                   1/1     Running            0          21m     192.168.2.5     worker6    <none>           <none>
default       net-test-dc8bf8666-bvbft                   1/1     Running            0          21m     192.168.2.4     worker6    <none>           <none>
default       net-test-dc8bf8666-pmx49                   1/1     Running            0          21m     192.168.2.3     worker6    <none>           <none>
[root@k8s-master ~]#
```

- **Scale up replicas again**



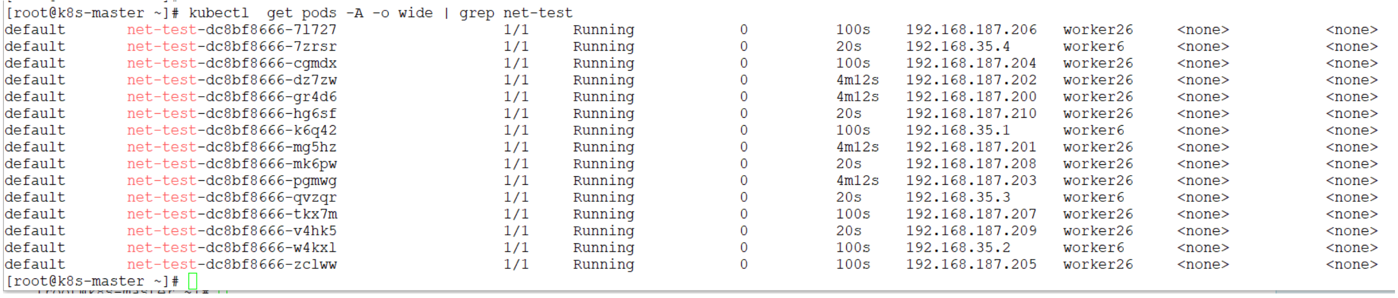
```
[root@k8s-master ~]# kubectl scale deployment net-test --replicas=6
deployment.apps/net-test scaled
[root@k8s-master ~]# kubectl  get pods -A -o wide | grep net-test
default       net-test-dc8bf8666-4gm5x                   0/1     ContainerCreating   0          7s      <none>            worker26   <none>           <none>
default       net-test-dc8bf8666-4plqc                   1/1     Running             0          23m     192.168.2.5       worker6    <none>           <none>
default       net-test-dc8bf8666-bvbft                   1/1     Running             0          23m     192.168.2.4       worker6    <none>           <none>
default       net-test-dc8bf8666-f4tvb                   0/1     ContainerCreating   0          7s      <none>            worker26   <none>           <none>
default       net-test-dc8bf8666-pmx49                   1/1     Running             0          23m     192.168.2.3       worker6    <none>           <none>
default       net-test-dc8bf8666-th5hf                   1/1     Running             0          7s      192.168.187.197   worker26   <none>           <none>
[root@k8s-master ~]#
```

- **Bring replicas to 0 and then to 10**



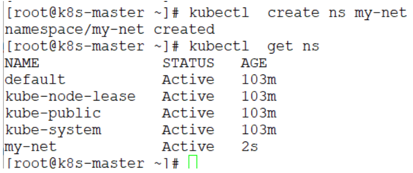
```
[root@k8s-master ~]# kubectl  get pods -A -o wide | grep net-test
default       net-test-dc8bf8666-71727                   1/1     Running            0          33s     192.168.187.206   worker26   <none>           <none>
default       net-test-dc8bf8666-cgmdx                   1/1     Running            0          33s     192.168.187.204   worker26   <none>           <none>
default       net-test-dc8bf8666-dz7zw                   1/1     Running            0          3m5s    192.168.187.202   worker26   <none>           <none>
default       net-test-dc8bf8666-gr4d6                   1/1     Running            0          3m5s    192.168.187.200   worker26   <none>           <none>
default       net-test-dc8bf8666-k6q42                   1/1     Running            0          33s     192.168.35.1      worker6    <none>           <none>
default       net-test-dc8bf8666-mg5hz                   1/1     Running            0          3m5s    192.168.187.201   worker26   <none>           <none>
default       net-test-dc8bf8666-pgmwg                   1/1     Running            0          3m5s    192.168.187.203   worker26   <none>           <none>
default       net-test-dc8bf8666-tkx7m                   1/1     Running            0          33s     192.168.187.207   worker26   <none>           <none>
default       net-test-dc8bf8666-w4kxl                   1/1     Running            0          33s     192.168.35.2      worker6    <none>           <none>
default       net-test-dc8bf8666-zclww                   1/1     Running            0          33s     192.168.187.205   worker26   <none>           <none>
[root@k8s-master ~]#
```

- **Scale up to 15**



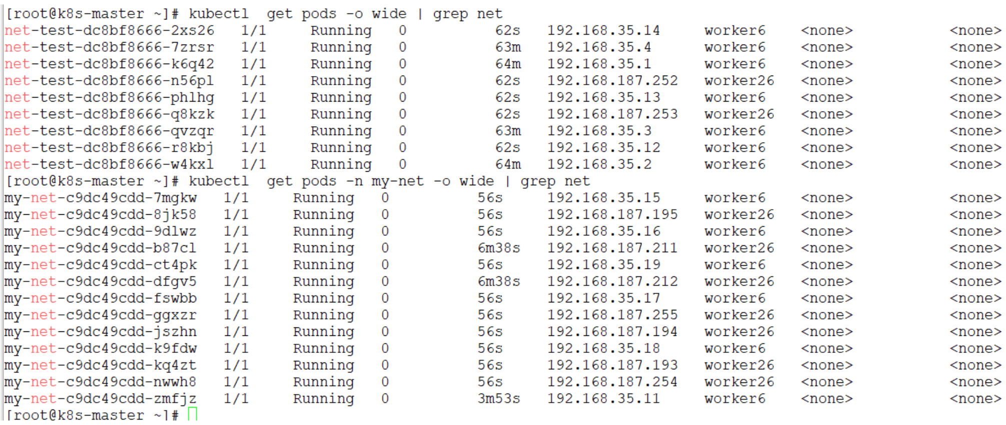
```
[root@k8s-master ~]# kubectl  get pods -A -o wide | grep net-test
default       net-test-dc8bf8666-71727                   1/1     Running            0          100s    192.168.187.206   worker26   <none>           <none>
default       net-test-dc8bf8666-7zrsr                   1/1     Running            0          20s     192.168.35.4      worker6    <none>           <none>
default       net-test-dc8bf8666-cgmdx                   1/1     Running            0          100s    192.168.187.204   worker26   <none>           <none>
default       net-test-dc8bf8666-dz7zw                   1/1     Running            0          4m12s   192.168.187.202   worker26   <none>           <none>
default       net-test-dc8bf8666-gr4d6                   1/1     Running            0          4m12s   192.168.187.200   worker26   <none>           <none>
default       net-test-dc8bf8666-hg6sf                   1/1     Running            0          20s     192.168.187.210   worker26   <none>           <none>
default       net-test-dc8bf8666-k6q42                   1/1     Running            0          100s    192.168.35.1      worker6    <none>           <none>
default       net-test-dc8bf8666-mg5hz                   1/1     Running            0          4m12s   192.168.187.201   worker26   <none>           <none>
default       net-test-dc8bf8666-mk6pw                   1/1     Running            0          20s     192.168.187.208   worker26   <none>           <none>
default       net-test-dc8bf8666-pgmwg                   1/1     Running            0          4m12s   192.168.187.203   worker26   <none>           <none>
default       net-test-dc8bf8666-qvzqr                   1/1     Running            0          20s     192.168.35.3      worker6    <none>           <none>
default       net-test-dc8bf8666-tkx7m                   1/1     Running            0          100s    192.168.187.207   worker26   <none>           <none>
default       net-test-dc8bf8666-v4hk5                   1/1     Running            0          20s     192.168.187.209   worker26   <none>           <none>
default       net-test-dc8bf8666-w4kxl                   1/1     Running            0          100s    192.168.35.2      worker6    <none>           <none>
default       net-test-dc8bf8666-zclww                   1/1     Running            0          100s    192.168.187.205   worker26   <none>           <none>
[root@k8s-master ~]#
```

- **Scaled down the deployment to 4**

- **And deployed one more deployment in different namespace**
  - 

```
[root@k8s-master ~]# kubectl  create ns my-net
namespace/my-net created
[root@k8s-master ~]# kubectl  get ns
NAME              STATUS   AGE
default           Active   103m
kube-node-lease   Active   103m
kube-public       Active   103m
kube-system       Active   103m
my-net            Active   2s
[root@k8s-master ~]#
```

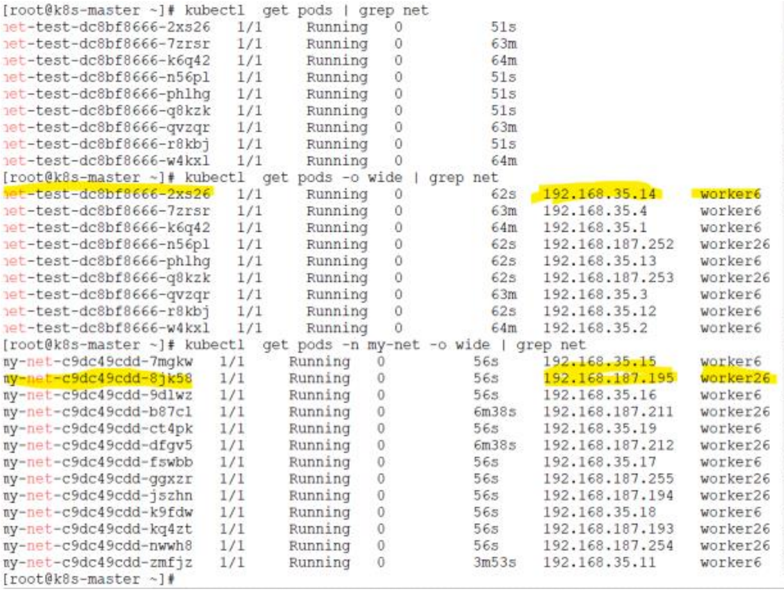
- **Scale up/down both the replicas so we get pods on both the workers**
-



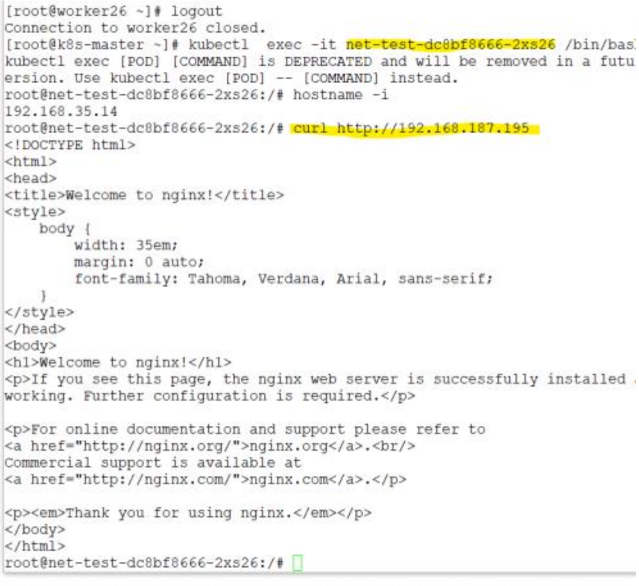
```
[root@k8s-master ~]# kubectl  get pods -o wide | grep net
net-test-dc8bf8666-2xs26   1/1     Running   0          62s   192.168.35.14     worker6    <none>           <none>
net-test-dc8bf8666-7zrsr   1/1     Running   0          63m   192.168.35.4      worker6    <none>           <none>
net-test-dc8bf8666-k6q42   1/1     Running   0          64m   192.168.35.1      worker6    <none>           <none>
net-test-dc8bf8666-n56pl   1/1     Running   0          62s   192.168.187.252   worker26   <none>           <none>
net-test-dc8bf8666-phlhg   1/1     Running   0          62s   192.168.35.13     worker6    <none>           <none>
net-test-dc8bf8666-q8kzk   1/1     Running   0          62s   192.168.187.253   worker26   <none>           <none>
net-test-dc8bf8666-qvzqr   1/1     Running   0          63m   192.168.35.3      worker6    <none>           <none>
net-test-dc8bf8666-r8kbj   1/1     Running   0          62s   192.168.35.12     worker6    <none>           <none>
net-test-dc8bf8666-w4kxl   1/1     Running   0          64m   192.168.35.2      worker6    <none>           <none>
[root@k8s-master ~]# kubectl  get pods -n my-net -o wide | grep net
my-net-c9dc49cdd-7mgkw   1/1     Running   0          56s     192.168.35.15     worker6    <none>           <none>
my-net-c9dc49cdd-8jk58   1/1     Running   0          56s     192.168.187.195   worker26   <none>           <none>
my-net-c9dc49cdd-9dlwz   1/1     Running   0          56s     192.168.35.16     worker6    <none>           <none>
my-net-c9dc49cdd-b87cl   1/1     Running   0          6m38s   192.168.187.211   worker26   <none>           <none>
my-net-c9dc49cdd-ct4pk   1/1     Running   0          56s     192.168.35.19     worker6    <none>           <none>
my-net-c9dc49cdd-dfgv5   1/1     Running   0          6m38s   192.168.187.212   worker26   <none>           <none>
my-net-c9dc49cdd-fswbb   1/1     Running   0          56s     192.168.35.17     worker6    <none>           <none>
my-net-c9dc49cdd-ggxzr   1/1     Running   0          56s     192.168.187.255   worker26   <none>           <none>
my-net-c9dc49cdd-jszhn   1/1     Running   0          56s     192.168.187.194   worker26   <none>           <none>
my-net-c9dc49cdd-k9fdw   1/1     Running   0          56s     192.168.35.18     worker6    <none>           <none>
my-net-c9dc49cdd-kq4zt   1/1     Running   0          56s     192.168.187.193   worker26   <none>           <none>
my-net-c9dc49cdd-nwwh8   1/1     Running   0          56s     192.168.187.254   worker26   <none>           <none>
my-net-c9dc49cdd-zmfjz   1/1     Running   0          3m53s   192.168.35.11     worker6    <none>           <none>
[root@k8s-master ~]#
```

==

- **Check inter pod communication across namespaces:**
  - **Pod from namespace default namespace are able to access service running on pod under namespace "my-net" namespace**



```
[root@k8s-master ~]# kubectl  get pods | grep net
net-test-dc8bf8666-2xs26   1/1     Running   0          51s
net-test-dc8bf8666-7zrsr   1/1     Running   0          63m
net-test-dc8bf8666-k6q42   1/1     Running   0          64m
net-test-dc8bf8666-n56pl   1/1     Running   0          51s
net-test-dc8bf8666-phlhg   1/1     Running   0          51s
net-test-dc8bf8666-q8kzk   1/1     Running   0          51s
net-test-dc8bf8666-qvzqr   1/1     Running   0          63m
net-test-dc8bf8666-r8kbj   1/1     Running   0          51s
net-test-dc8bf8666-w4kxl   1/1     Running   0          64m
[root@k8s-master ~]# kubectl  get pods -o wide | grep net
net-test-dc8bf8666-2xs26   1/1     Running   0          62s   192.168.35.14     worker6
net-test-dc8bf8666-7zrsr   1/1     Running   0          63m   192.168.35.4      worker6
net-test-dc8bf8666-k6q42   1/1     Running   0          64m   192.168.35.1      worker6
net-test-dc8bf8666-n56pl   1/1     Running   0          62s   192.168.187.252   worker26
net-test-dc8bf8666-phlhg   1/1     Running   0          62s   192.168.35.13     worker6
net-test-dc8bf8666-q8kzk   1/1     Running   0          62s   192.168.187.253   worker26
net-test-dc8bf8666-qvzqr   1/1     Running   0          63m   192.168.35.3      worker6
net-test-dc8bf8666-r8kbj   1/1     Running   0          62s   192.168.35.12     worker6
net-test-dc8bf8666-w4kxl   1/1     Running   0          64m   192.168.35.2      worker6
[root@k8s-master ~]# kubectl  get pods -n my-net -o wide | grep net
my-net-c9dc49cdd-7mgkw   1/1     Running   0          56s     192.168.35.15     worker6
my-net-c9dc49cdd-8jk58   1/1     Running   0          56s     192.168.187.195   worker26
my-net-c9dc49cdd-9dlwz   1/1     Running   0          56s     192.168.35.16     worker6
my-net-c9dc49cdd-b87cl   1/1     Running   0          6m38s   192.168.187.211   worker26
my-net-c9dc49cdd-ct4pk   1/1     Running   0          56s     192.168.35.19     worker6
my-net-c9dc49cdd-dfgv5   1/1     Running   0          6m38s   192.168.187.212   worker26
my-net-c9dc49cdd-fswbb   1/1     Running   0          56s     192.168.35.17     worker6
my-net-c9dc49cdd-ggxzr   1/1     Running   0          56s     192.168.187.255   worker26
my-net-c9dc49cdd-jszhn   1/1     Running   0          56s     192.168.187.194   worker26
my-net-c9dc49cdd-k9fdw   1/1     Running   0          56s     192.168.35.18     worker6
my-net-c9dc49cdd-kq4zt   1/1     Running   0          56s     192.168.187.193   worker26
my-net-c9dc49cdd-nwwh8   1/1     Running   0          56s     192.168.187.254   worker26
my-net-c9dc49cdd-zmfjz   1/1     Running   0          3m53s   192.168.35.11     worker6
[root@k8s-master ~]#
```

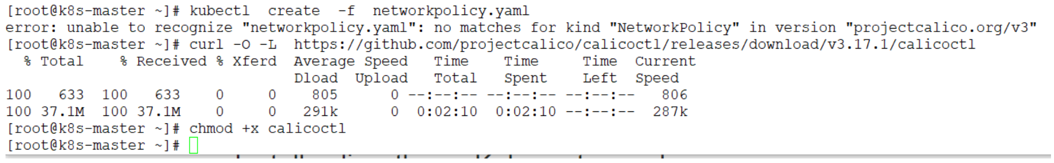


```
[root@worker26 ~]# logout
Connection to worker26 closed.
[root@k8s-master ~]# kubectl  exec -it net-test-dc8bf8666-2xs26 /bin/bas
kubectl exec [POD] [COMMAND] is DEPRECATED and will be removed in a futu
ersion. Use kubectl exec [POD] -- [COMMAND] instead.
root@net-test-dc8bf8666-2xs26:/# hostname -i
192.168.35.14
root@net-test-dc8bf8666-2xs26:/# curl http://192.168.187.195
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
    body {
        width: 35em;
        margin: 0 auto;
        font-family: Tahoma, Verdana, Arial, sans-serif;
    }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed .
working. Further configuration is required.</p>

<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>

<p><em>Thank you for using nginx.</em></p>
</body>
</html>
root@net-test-dc8bf8666-2xs26:/#
```

- **Calico now manages the networks and IPs in my nodes and assigns a different network address for each node.**

Network Policy

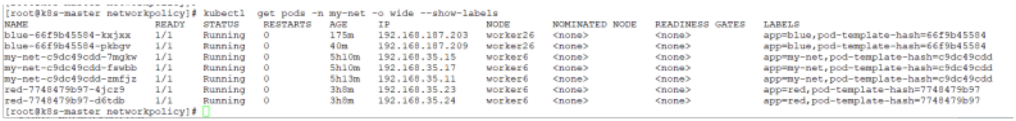
Install calicoctl



```
[root@k8s-master ~]# kubectl  create  -f  networkpolicy.yaml
error: unable to recognize "networkpolicy.yaml": no matches for kind "NetworkPolicy" in version "projectcalico.org/v3"
[root@k8s-master ~]# curl -O -L  https://github.com/projectcalico/calicoctl/releases/download/v3.17.1/calicoctl
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   633  100   633    0     0    805      0 --:--:-- --:--:-- --:--:--   806
100 37.1M  100 37.1M    0     0   291k      0  0:02:10  0:02:10 --:--:--  287k
[root@k8s-master ~]# chmod +x calicoctl
[root@k8s-master ~]#
```

# copy command to /usr/local/bin

Scenario:



```
[root@k8s-master networkpolicy]# kubectl  get pods -n my-net -o wide --show-labels
NAME                     READY   STATUS    RESTARTS   AGE    IP                NODE       NOMINATED NODE   READINESS GATES   LABELS
blue-66f9b45584-kxjxx    1/1     Running   0          175m   192.168.187.203   worker26   <none>           <none>            app=blue,pod-template-hash=66f9b45584
blue-66f9b45584-pkbgv    1/1     Running   0          40m    192.168.187.209   worker26   <none>           <none>            app=blue,pod-template-hash=66f9b45584
my-net-c9dc49cdd-7mgkw   1/1     Running   0          5h10m  192.168.35.15     worker6    <none>           <none>            app=my-net,pod-template-hash=c9dc49cdd
my-net-c9dc49cdd-fswbb   1/1     Running   0          5h10m  192.168.35.17     worker6    <none>           <none>            app=my-net,pod-template-hash=c9dc49cdd
my-net-c9dc49cdd-zmfjz   1/1     Running   0          5h13m  192.168.35.11     worker6    <none>           <none>            app=my-net,pod-template-hash=c9dc49cdd
red-7748479b97-4jcz9     1/1     Running   0          3h8m   192.168.35.23     worker6    <none>           <none>            app=red,pod-template-hash=7748479b97
red-7748479b97-d6tdb     1/1     Running   0          3h8m   192.168.35.24     worker6    <none>           <none>            app=red,pod-template-hash=7748479b97
[root@k8s-master networkpolicy]#
```

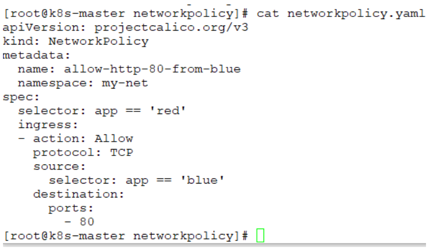
2 pods = label "app=blue"
3 pods = label "app=my-net"
2 pods = label "app=red"

**** NS = my-net**

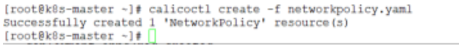
1) Apply policy within NS

ingress traffic to endpoints in the **namespace: my-net** with label **app: red** is allowed, only if it comes from a pod in the same namespace with **app: blue**, on port **80**.

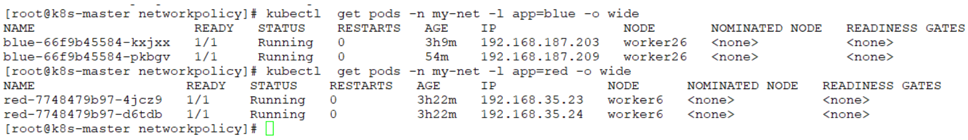
- Allow traffic within default NS from pods:
  - □ Incoming traffic coming **to** pods labeled "app: red"
  - □ Incoming traffic coming **from** pods labeled "app: blue"



```
[root@k8s-master networkpolicy]# cat networkpolicy.yaml
apiVersion: projectcalico.org/v3
kind: NetworkPolicy
metadata:
  name: allow-http-80-from-blue
  namespace: my-net
spec:
  selector: app == 'red'
  ingress:
  - action: Allow
    protocol: TCP
    source:
      selector: app == 'blue'
    destination:
      ports:
        - 80
[root@k8s-master networkpolicy]#
```



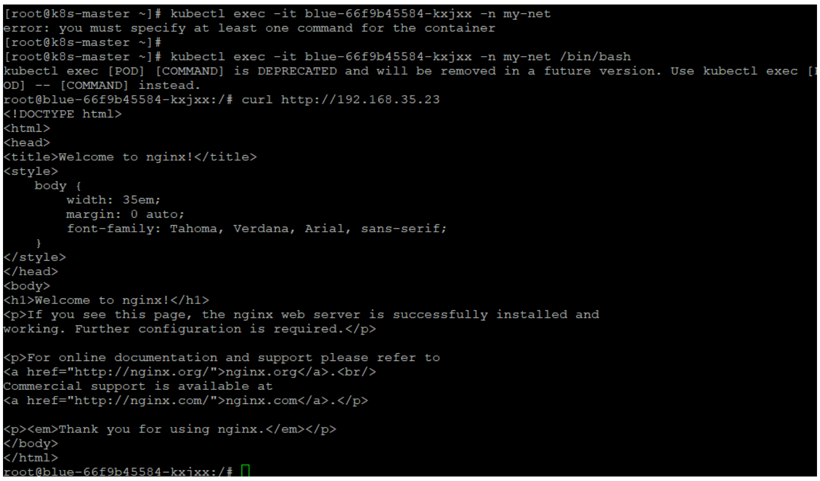
```
[root@k8s-master ~]# calicoctl create -f networkpolicy.yaml
Successfully created 1 'NetworkPolicy' resource(s)
[root@k8s-master ~]#
```



```
[root@k8s-master networkpolicy]# kubectl  get pods -n my-net -l app=blue -o wide
NAME                    READY   STATUS    RESTARTS   AGE    IP                NODE       NOMINATED NODE   READINESS GATES
blue-66f9b45584-kxjxx   1/1     Running   0          3h9m   192.168.187.203   worker26   <none>           <none>
blue-66f9b45584-pkbgv   1/1     Running   0          54m    192.168.187.209   worker26   <none>           <none>
[root@k8s-master networkpolicy]# kubectl  get pods -n my-net -l app=red -o wide
NAME                   READY   STATUS    RESTARTS   AGE     IP              NODE      NOMINATED NODE   READINESS GATES
red-7748479b97-4jcz9   1/1     Running   0          3h22m   192.168.35.23   worker6   <none>           <none>
red-7748479b97-d6tdb   1/1     Running   0          3h22m   192.168.35.24   worker6   <none>           <none>
[root@k8s-master networkpolicy]#
```

Results: within NS

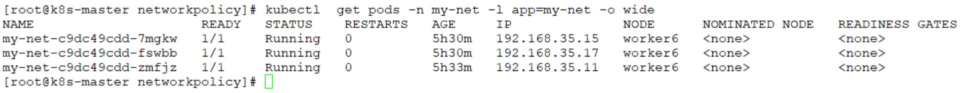
- **Source - app: blue --> destination - app: red**
  - **Traffic allowed**



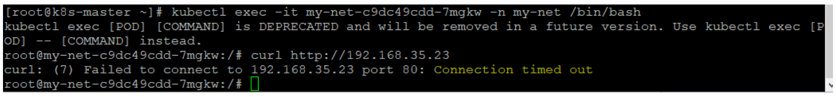
```
[root@k8s-master ~]# kubectl exec -it blue-66f9b45584-kxjxx -n my-net
error: you must specify at least one command for the container
[root@k8s-master ~]#
[root@k8s-master ~]# kubectl exec -it blue-66f9b45584-kxjxx -n my-net /bin/bash
kubectl exec [POD] [COMMAND] is DEPRECATED and will be removed in a future version. Use kubectl exec [P
OD] -- [COMMAND] instead.
root@blue-66f9b45584-kxjxx:/# curl http://192.168.35.23
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
    body {
        width: 35em;
        margin: 0 auto;
        font-family: Tahoma, Verdana, Arial, sans-serif;
    }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>

<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>

<p><em>Thank you for using nginx.</em></p>
</body>
</html>
root@blue-66f9b45584-kxjxx:/#
```

  - Once policy is applied, only specific traffic is allowed

- **Source - app: my-net --> destination - app: red**
  - **Traffic NOT allowed**



```
[root@k8s-master networkpolicy]# kubectl  get pods -n my-net -l app=my-net -o wide
NAME                      READY   STATUS    RESTARTS   AGE     IP              NODE      NOMINATED NODE   READINESS GATES
my-net-c9dc49cdd-7mgkw    1/1     Running   0          5h30m   192.168.35.15   worker6   <none>           <none>
my-net-c9dc49cdd-fswbb    1/1     Running   0          5h30m   192.168.35.17   worker6   <none>           <none>
my-net-c9dc49cdd-zmfjz    1/1     Running   0          5h33m   192.168.35.11   worker6   <none>           <none>
[root@k8s-master networkpolicy]#
```



```
[root@k8s-master ~]# kubectl exec -it my-net-c9dc49cdd-7mgkw -n my-net /bin/bash
kubectl exec [POD] [COMMAND] is DEPRECATED and will be removed in a future version. Use kubectl exec [P
OD] -- [COMMAND] instead.
root@my-net-c9dc49cdd-7mgkw:/# curl http://192.168.35.23
curl: (7) Failed to connect to 192.168.35.23 port 80: Connection timed out
root@my-net-c9dc49cdd-7mgkw:/#
```

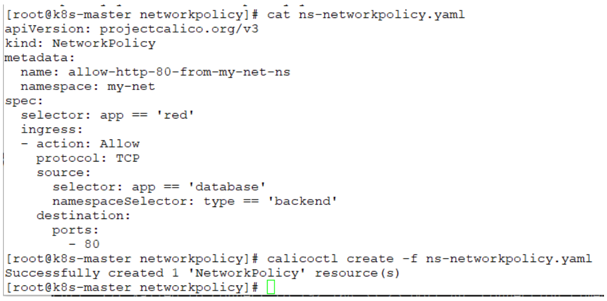
2) Apply policy across NS

*** Previous policy has blocked traffic across NS

ingress traffic to endpoints in the **namespace: my-net** with label **app: red** is allowed, if it comes from a pod in the namespace with **type: backend**, on port **80**.

□ Allow traffic within default NS from pods:

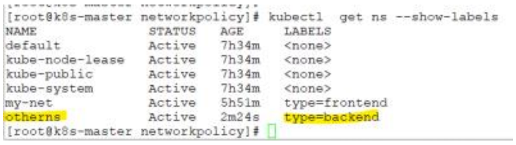
- ◇ Incoming traffic coming **to** pods labeled "app: red" in *my-net* NS
- ◇ Incoming traffic coming **from** pods labeled "app: database" in *otherns* NS



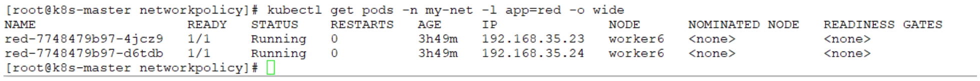
```
[root@k8s-master networkpolicy]# cat ns-networkpolicy.yaml
apiVersion: projectcalico.org/v3
kind: NetworkPolicy
metadata:
  name: allow-http-80-from-my-net-ns
  namespace: my-net
spec:
  selector: app == 'red'
  ingress:
  - action: Allow
    protocol: TCP
    source:
      selector: app == 'database'
      namespaceSelector: type == 'backend'
    destination:
      ports:
        - 80
[root@k8s-master networkpolicy]# calicoctl create -f ns-networkpolicy.yaml
Successfully created 1 'NetworkPolicy' resource(s)
[root@k8s-master networkpolicy]#
```

Scenario:

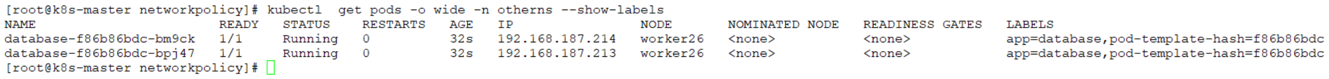
- **2 pod - app: database [NS - otherns]**
- **2 pod - app: red [NS - my-net]**



```
[root@k8s-master networkpolicy]# kubectl  get ns --show-labels
NAME                STATUS   AGE     LABELS
default             Active   7h34m   <none>
kube-node-lease     Active   7h34m   <none>
kube-public         Active   7h34m   <none>
kube-system         Active   7h34m   <none>
my-net              Active   5h51m   type=frontend
otherns             Active   2m24s   type=backend
[root@k8s-master networkpolicy]#
```



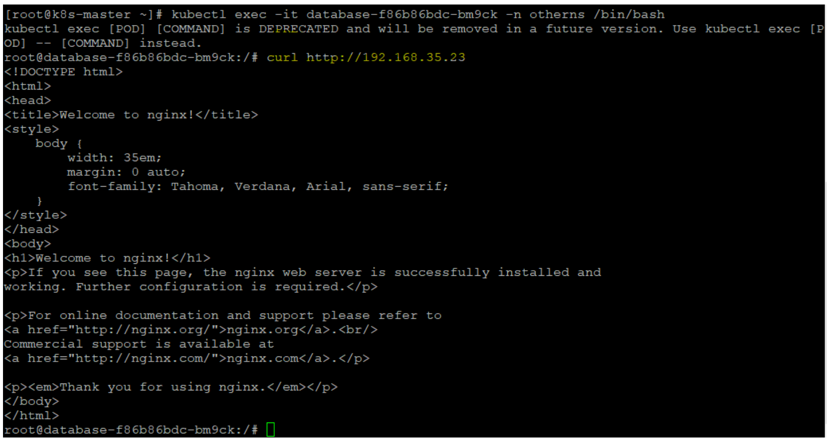
```
[root@k8s-master networkpolicy]# kubectl get pods -n my-net -l app=red -o wide
NAME                   READY   STATUS    RESTARTS   AGE     IP              NODE      NOMINATED NODE   READINESS GATES
red-7748479b97-4jcz9   1/1     Running   0          3h49m   192.168.35.23   worker6   <none>           <none>
red-7748479b97-d6tdb   1/1     Running   0          3h49m   192.168.35.24   worker6   <none>           <none>
[root@k8s-master networkpolicy]#
```



```
[root@k8s-master networkpolicy]# kubectl  get pods -o wide -n otherns --show-labels
NAME                        READY   STATUS    RESTARTS   AGE   IP                NODE       NOMINATED NODE   READINESS GATES   LABELS
database-f86b86bdc-bm9ck    1/1     Running   0          32s   192.168.187.214   worker26   <none>           <none>            app=database,pod-template-hash=f86b86bdc
database-f86b86bdc-bpj47    1/1     Running   0          32s   192.168.187.213   worker26   <none>           <none>            app=database,pod-template-hash=f86b86bdc
[root@k8s-master networkpolicy]#
```

Results: across NS

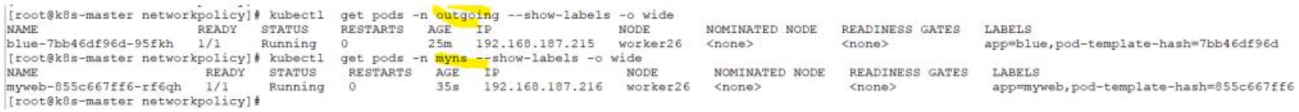
- **Source - app: database (otherns) --> destination - app: red (my-net)**
  - **Traffic IS ALLOWED allow NOW**



```
[root@k8s-master ~]# kubectl exec -it database-f86b86bdc-bm9ck -n otherns /bin/bash
kubectl exec [POD] [COMMAND] is DEPRECATED and will be removed in a future version. Use kubectl exec [P
OD] -- [COMMAND] instead.
root@database-f86b86bdc-bm9ck:/# curl http://192.168.35.23
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
    body {
        width: 35em;
        margin: 0 auto;
        font-family: Tahoma, Verdana, Arial, sans-serif;
    }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>

<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>

<p><em>Thank you for using nginx.</em></p>
</body>
</html>
root@database-f86b86bdc-bm9ck:/#
```

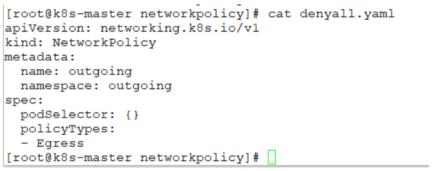
3) Apply deny all EGRESS policy on NS

Scenario:

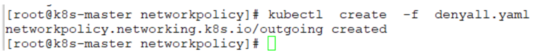


```
[root@k8s-master networkpolicy]# kubectl  get pods -n outgoing --show-labels -o wide
NAME                    READY   STATUS    RESTARTS   AGE   IP                NODE       NOMINATED NODE   READINESS GATES   LABELS
blue-7bb46df96d-95fkh   1/1     Running   0          25m   192.168.187.215   worker26   <none>           <none>            app=blue,pod-template-hash=7bb46df96d
[root@k8s-master networkpolicy]# kubectl  get pods -n myns --show-labels -o wide
NAME                     READY   STATUS    RESTARTS   AGE   IP                NODE       NOMINATED NODE   READINESS GATES   LABELS
myweb-855c667ff6-rf6qh   1/1     Running   0          35s   192.168.187.216   worker26   <none>           <none>            app=myweb,pod-template-hash=855c667ff6
[root@k8s-master networkpolicy]#
```

Now apply deny all egress policy on outgoing ns

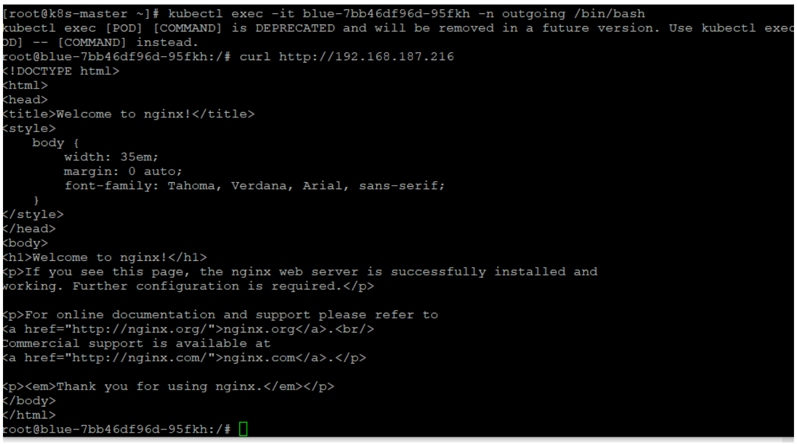


```
[root@k8s-master networkpolicy]# cat denyall.yaml
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  name: outgoing
  namespace: outgoing
spec:
  podSelector: {}
  policyTypes:
  - Egress
[root@k8s-master networkpolicy]#
```



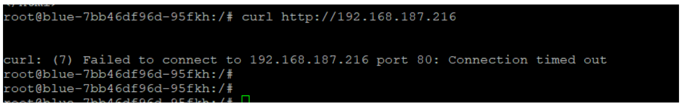
```
[root@k8s-master networkpolicy]# kubectl  create  -f  denyall.yaml
networkpolicy.networking.k8s.io/outgoing created
[root@k8s-master networkpolicy]#
```

Before the policy



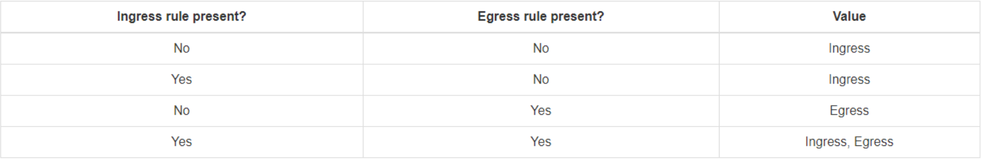
```
[root@k8s-master ~]# kubectl exec -it blue-7bb46df96d-95fkh -n outgoing /bin/bash
kubectl exec [POD] [COMMAND] is DEPRECATED and will be removed in a future version. Use kubectl exec
OD] -- [COMMAND] instead.
root@blue-7bb46df96d-95fkh:/# curl http://192.168.187.216
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
    body {
        width: 35em;
        margin: 0 auto;
        font-family: Tahoma, Verdana, Arial, sans-serif;
    }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>

<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>

<p><em>Thank you for using nginx.</em></p>
</body>
</html>
root@blue-7bb46df96d-95fkh:/#
```

After deny all policy:



```
</html>
root@blue-7bb46df96d-95fkh:/# curl http://192.168.187.216


curl: (7) Failed to connect to 192.168.187.216 port 80: Connection timed out
root@blue-7bb46df96d-95fkh:/#
root@blue-7bb46df96d-95fkh:/#
root@blue-7bb46df96d-95fkh:/#
```

==



Ingress rule present?	Egress rule present?	Value
No	No	Ingress
Yes	No	Ingress
No	Yes	Egress
Yes	Yes	Ingress, Egress