Metal-LB

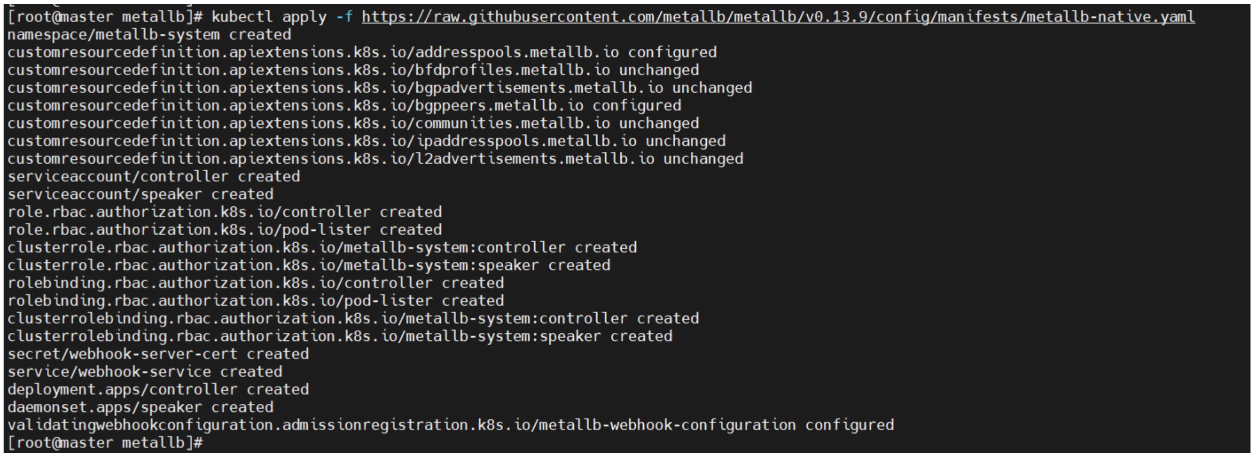
Deploy Metal-lb in Kubernetes cluster using YAMl.

Prerequisite:

Kubernetes cluster ready with v1.19.0+ later version

Deploy metallb yaml in k8s cluster

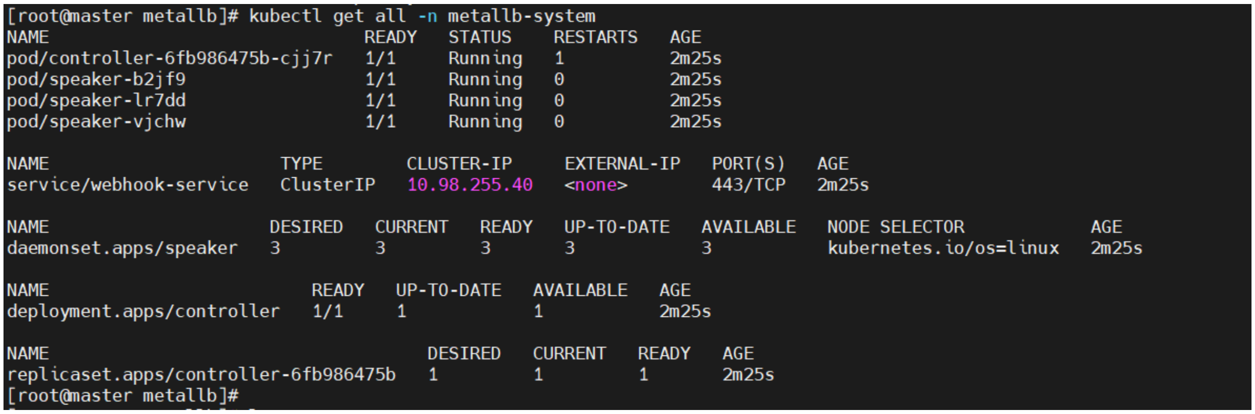
kubectl apply -f
https://raw.githubusercontent.com/metallb/metallb/v0.13.9/config/manifests/metallb-native.yaml



```
[root@master metallb]# kubectl apply -f https://raw.githubusercontent.com/metallb/metallb/v0.13.9/config/manifests/metallb-native.yaml
namespace/metallb-system created
customresourcedefinition.apiextensions.k8s.io/addresspools.metallb.io configured
customresourcedefinition.apiextensions.k8s.io/bfdprofiles.metallb.io unchanged
customresourcedefinition.apiextensions.k8s.io/bgpadvertisements.metallb.io unchanged
customresourcedefinition.apiextensions.k8s.io/bgppeers.metallb.io configured
customresourcedefinition.apiextensions.k8s.io/communities.metallb.io unchanged
customresourcedefinition.apiextensions.k8s.io/ipaddresspools.metallb.io unchanged
customresourcedefinition.apiextensions.k8s.io/l2advertisements.metallb.io unchanged
serviceaccount/controller created
serviceaccount/speaker created
role.rbac.authorization.k8s.io/controller created
role.rbac.authorization.k8s.io/pod-lister created
clusterrole.rbac.authorization.k8s.io/metallb-system:controller created
clusterrole.rbac.authorization.k8s.io/metallb-system:speaker created
rolebinding.rbac.authorization.k8s.io/controller created
rolebinding.rbac.authorization.k8s.io/pod-lister created
clusterrolebinding.rbac.authorization.k8s.io/metallb-system:controller created
clusterrolebinding.rbac.authorization.k8s.io/metallb-system:speaker created
secret/webhook-server-cert created
service/webhook-service created
deployment.apps/controller created
daemonset.apps/speaker created
validatingwebhookconfiguration.admissionregistration.k8s.io/metallb-webhook-configuration configured
[root@master metallb]#
```

Check All resource are up an running or not

# kubectl get all -n metallb-system



```
[root@master metallb]# kubectl get all -n metallb-system
NAME                             READY   STATUS    RESTARTS   AGE
pod/controller-6fb986475b-cjj7r   1/1     Running   1          2m25s
pod/speaker-b2jf9                 1/1     Running   0          2m25s
pod/speaker-lr7dd                 1/1     Running   0          2m25s
pod/speaker-vjchw                 1/1     Running   0          2m25s

NAME                      TYPE        CLUSTER-IP     EXTERNAL-IP   PORT(S)   AGE
service/webhook-service   ClusterIP   10.98.255.40   <none>        443/TCP   2m25s

NAME                     DESIRED   CURRENT   READY   UP-TO-DATE   AVAILABLE   NODE SELECTOR            AGE
daemonset.apps/speaker   3         3         3       3            3           kubernetes.io/os=linux   2m25s

NAME                         READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/controller   1/1     1            1           2m25s

NAME                                    DESIRED   CURRENT   READY   AGE
replicaset.apps/controller-6fb986475b   1         1         1       2m25s
[root@master metallb]#
```

Now deploy address pool

Create a file IPaddresspool.yaml and add below content in it

# apiVersion: metallb.io/v1beta1

kind: IPAddressPool

metadata:

  name: first-pool

  namespace: metallb-system

spec:

  addresses:

  - <free source ip range> - <destination ip range>

Example :

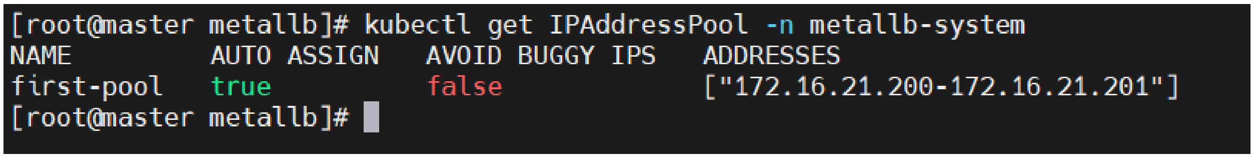
- 172.16.21.100 – 172.16.21.110

Apply IPaddresspool.yaml file

# kubectl apply -f IPaddresspool.yaml

Check IPaddresspool deployed in cluster

```
# kubectl get IPAddressPool -n metallb-system
```



```
[root@master metallb]# kubectl get IPAddressPool -n metallb-system
NAME          AUTO ASSIGN   AVOID BUGGY IPS   ADDRESSES
first-pool    true          false             ["172.16.21.200-172.16.21.201"]
[root@master metallb]# 
```

Now deploy L2Advertisement

Create new file l2advertisement.yaml and add below content in it

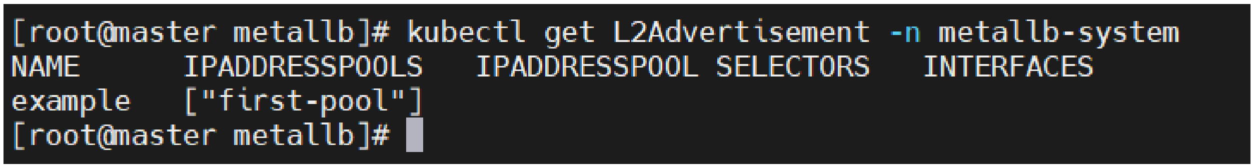
```
# apiVersion: metallb.io/v1beta1
kind: L2Advertisement
metadata:
  name: example
  namespace: metallb-system
spec:
  ipAddressPools:
  - first-pool
```

Apply the l2advertisement.yaml file

```
# kubectl apply -f l2advertisement.yaml
```

Check deployed resource

```
# kubectl get L2Advertisement -n metallb-system
```

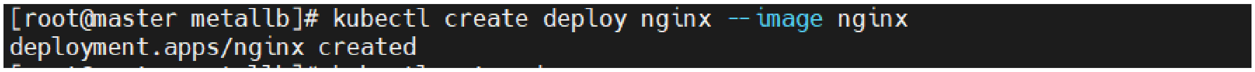


```
[root@master metallb]# kubectl get L2Advertisement -n metallb-system
NAME      IPADDRESSPOOLS   IPADDRESSPOOL SELECTORS   INTERFACES
example   ["first-pool"]
[root@master metallb]#
```

Deploy sample nginx Application

Execute below command to deploy nginx application

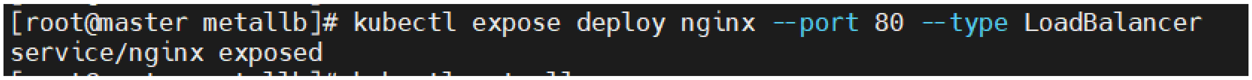
kubectl create deploy nginx --image nginx



```
[root@master metallb]# kubectl create deploy nginx --image nginx
deployment.apps/nginx created
```

Create service for nginx application with service type Load Balancer

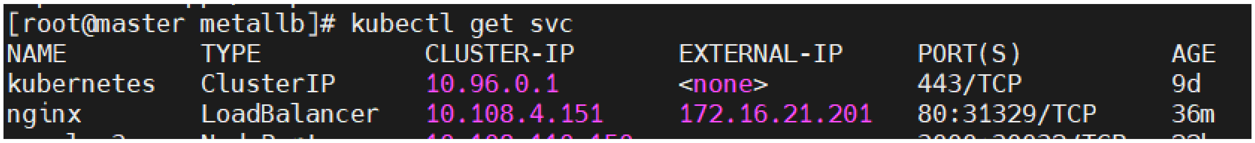
kubectl expose deploy nginx --port 80 --type LoadBalancer



```
[root@master metallb]# kubectl expose deploy nginx --port 80 --type LoadBalancer
service/nginx exposed
```

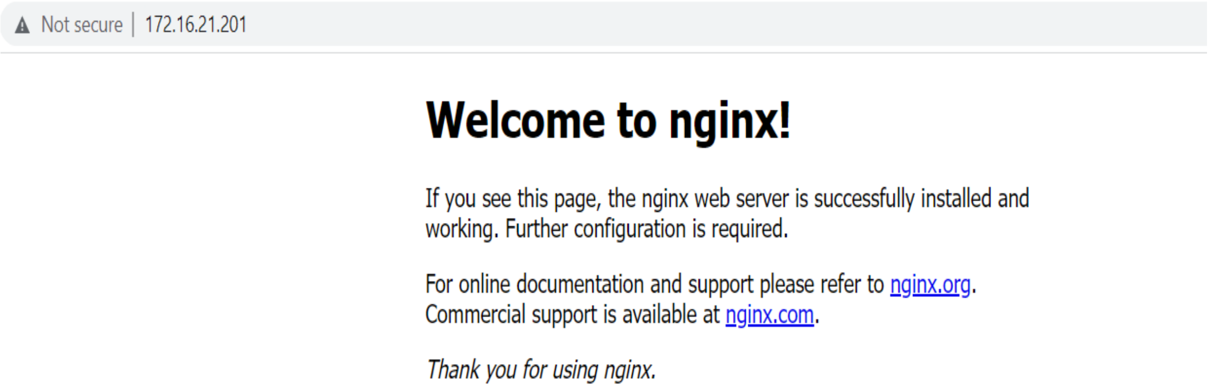
Now check Ip address assign to nginx service from metal-lb ip range

kubectl get svc



```
[root@master metallb]# kubectl get svc
NAME         TYPE           CLUSTER-IP      EXTERNAL-IP     PORT(S)         AGE
kubernetes   ClusterIP      10.96.0.1       <none>          443/TCP         9d
nginx        LoadBalancer   10.108.4.151    172.16.21.201   80:31329/TCP    36m
```

Now check Load Balancer service working or not



Not secure | 172.16.21.201

Welcome to nginx!

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 nginx.com.

Thank you for using nginx.