# MetalLB come loadBalancer

La pagina ufficiale dell'installazione: https://metallb.universe.tf/installation/

Ho eseguito l'operazione preliminare “If you’re using kube-proxy in IPVS mode, since Kubernetes v1.14.2 you have to enable strict ARP mode.”

Edito la configMap di kube-proxy:

```
ubuntu@kubectl:~$ kubectl edit configmap -n kube-system kube-proxy
apiVersion: v1
data:
  config.conf: |-
    apiVersion: kubeproxy.config.k8s.io/v1alpha1
...
...
    #mode: "iptables"
    mode: "ipvs"
    ipvs:
      strictARP: true
    kind: KubeProxyConfiguration
```

A questo punto è tutto abbastanza semplice, usando il deploy tramite `manifest`:

https://metallb.universe.tf/installation/#installation-by-manifest

```
kubectl apply -f
https://raw.githubusercontent.com/metallb/metallb/v0.14.8/config/manifests/m
etallb-native.yaml
```

Adesso che è installato nel suo namespace:

```
ubuntu@kubectl:~$ kubectl get all -n metallb-system
NAME                                READY   STATUS    RESTARTS   AGE
pod/controller-8694df9d9b-8b9zc   1/1     Running   0          145m
pod/speaker-9wxnp                 1/1     Running   0          145m

NAME                                TYPE        CLUSTER-IP      EXTERNAL-IP
PORT(S)   AGE
service/metallb-webhook-service   ClusterIP   10.105.26.211   <none>
443/TCP   145m

NAME                     DESIRED   CURRENT   READY   UP-TO-DATE   AVAILABLE
NODE SELECTOR            AGE
daemonset.apps/speaker   1         1         1       1            1
kubernetes.io/os=linux   145m

NAME                         READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/controller   1/1     1            1           145m
```

```
NAME                                       DESIRED   CURRENT   READY   AGE
replicaset.apps/controller-8694df9d9b   1         1         1       145m
```

creiamo un pool di IP (in questo caso 2, per test)

```
ubuntu@kubectl:~$ cat metalLB/metallb-pool.yaml
apiVersion: metallb.io/v1beta1
kind: IPAddressPool
metadata:
  name: public-pool
  namespace: metallb-system
spec:
  addresses:
  - 185.91.188.40-185.91.188.41
```

Ci sono due modi di utilizzo di questo Load Balancer: Layer 2 e BGP. Il Layer 2 funziona con il protocollo `arp` ed è di gestione più semplice.

```
ubuntu@kubectl:~$ cat metalLB/metallb-L2.yaml
apiVersion: metallb.io/v1beta1
kind: L2Advertisement
metadata:
  name: ingress
  namespace: metallb-system
spec:
  ipAddressPools:
  - public-pool
```

a questo punto è immediato un test:

```
ubuntu@kubectl:~$ kubectl create deploy nginx1 --image nginx:latest
ubuntu@kubectl:~$ kubectl expose deploy nginx1 --port 80 --type LoadBalancer
ubuntu@kubectl:~$ kubectl create deploy nginx2 --image nginx:latest
ubuntu@kubectl:~$ kubectl expose deploy nginx2 --port 80 --type LoadBalancer
ubuntu@kubectl:~$ kubectl get all
NAME                                    READY   STATUS    RESTARTS   AGE
pod/nginx1-c95765fd4-wxkds              1/1     Running   0          1m
pod/nginx2-774db4988c-m2vh5             1/1     Running   0          1m

NAME                           TYPE           CLUSTER-IP       EXTERNAL-IP
PORT(S)          AGE
service/nginx1                 LoadBalancer   10.109.150.219
185.91.188.40   80:31721/TCP     3m
                               ^^^^^^^^^^^^
^^^^^^^^^^^^^
service/nginx2                 LoadBalancer   10.109.150.219
185.91.188.41   80:31721/TCP     3m
```

ATTENZIONE: Il ping non risponde perché metalLB non gestisce le risposte ICMP.