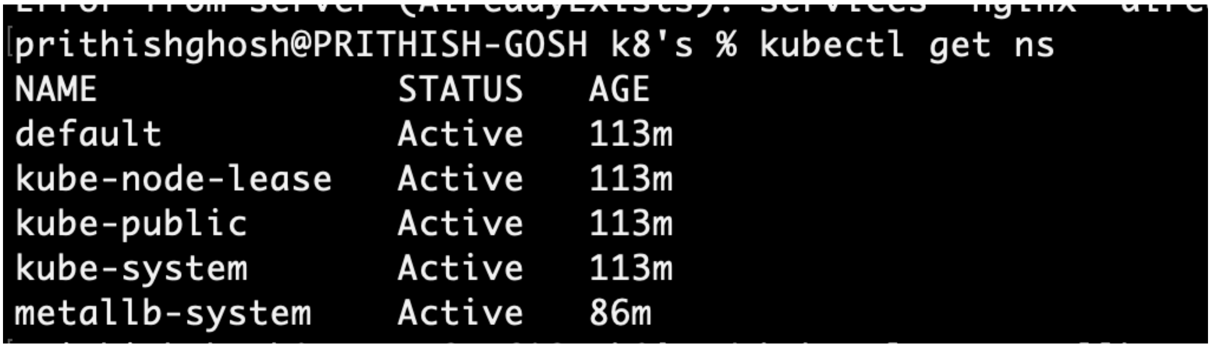
Host Nginx website on minikube Cluster as a load balancer with certificate manager :

Step 1. Install Metallb using the metallb manifest file :

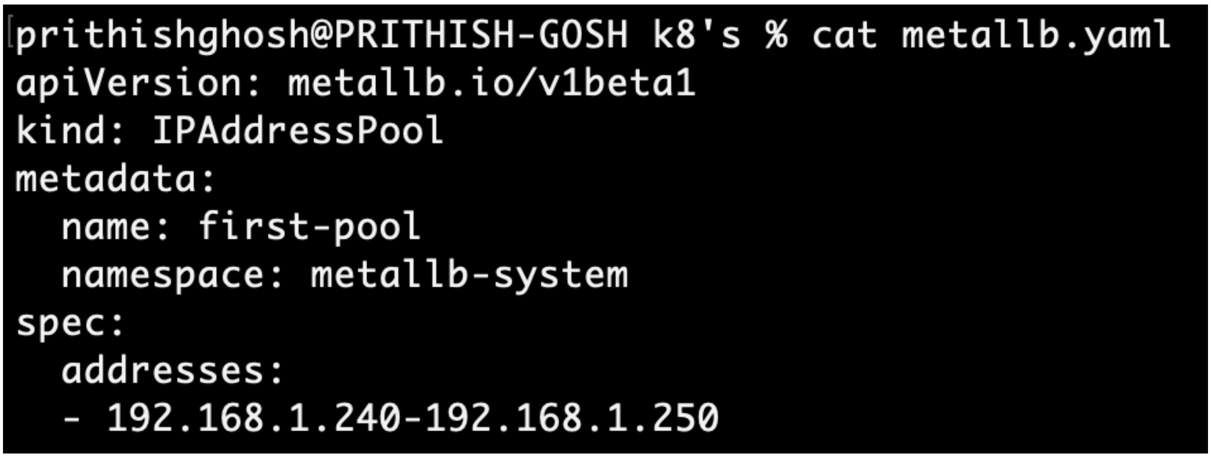
```
kubectl apply -f
https://raw.githubusercontent.com/metallb/metallb/v0.13.12/config/manifests/metallb-nativ
e.yaml
```

This has installed all the required CDR’s in a specific namespace called : **metallb-system**



```
Error from server (AlreadyExists): services "nginx" alre
prithishghosh@PRITHISH-GOSH k8's % kubectl get ns
NAME                STATUS   AGE
default             Active   113m
kube-node-lease     Active   113m
kube-public         Active   113m
kube-system         Active   113m
metallb-system      Active   86m
```

Step 2 . created a metallb config file where defined IPs from `192.168.1.240` to `192.168.1.250`, and configured Layer 2 mode as we don’t need any protocol-specific configuration, only IP addresses that will be assigned to LB .



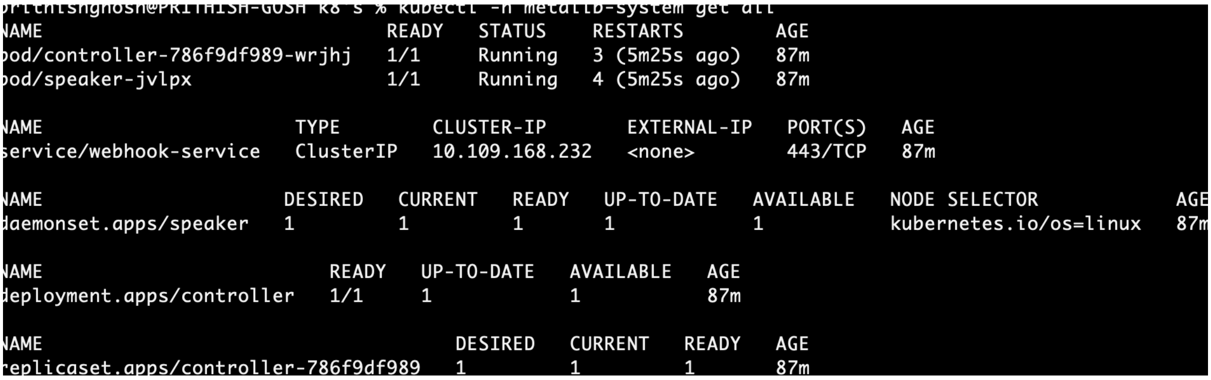
```
prithishghosh@PRITHISH-GOSH k8's % cat metallb.yaml
apiVersion: metallb.io/v1beta1
kind: IPAddressPool
metadata:
  name: first-pool
  namespace: metallb-system
spec:
  addresses:
  - 192.168.1.240-192.168.1.250
```

Kubectl apply -f metallb.yaml

Step 3. Now for testing our metallb used a Nginx container deployment to test the handing out of IP addresses from MetalLB using

```
kubectl create deploy nginx --image nginx:latest
```

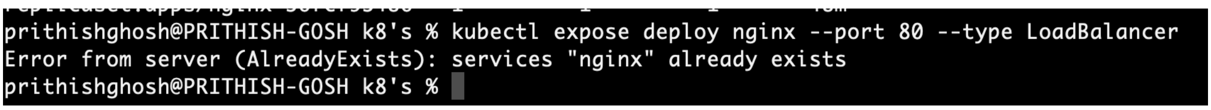
Now checked all pods : kubectl get all -n metallb-system



```
prithishghosh@PRITHISH-GOSH k8's % kubectl -n metallb-system get all
NAME                             READY   STATUS    RESTARTS        AGE
pod/controller-786f9df989-wrjhj  1/1     Running   3 (5m25s ago)   87m
pod/speaker-jvlpx                1/1     Running   4 (5m25s ago)   87m

NAME                      TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)   AGE
service/webhook-service   ClusterIP   10.109.168.232   <none>        443/TCP   87m

NAME                     DESIRED   CURRENT   READY   UP-TO-DATE   AVAILABLE   NODE SELECTOR            AGE
daemonset.apps/speaker   1         1         1       1            1           kubernetes.io/os=linux   87m

NAME                         READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/controller   1/1     1            1           87m

NAME                                    DESIRED   CURRENT   READY   AGE
replicaset.apps/controller-786f9df989   1         1         1       87m
```

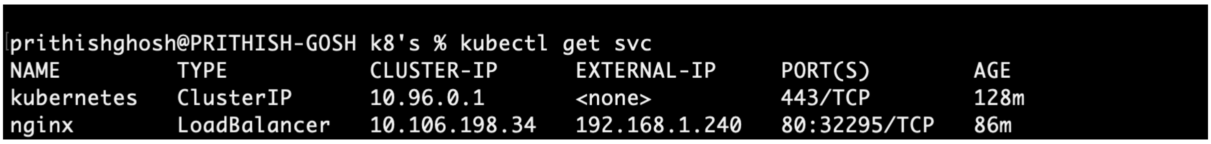
Now deployed Nginx test pod, can be exposed the deployment using the type **LoadBalancer**.

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



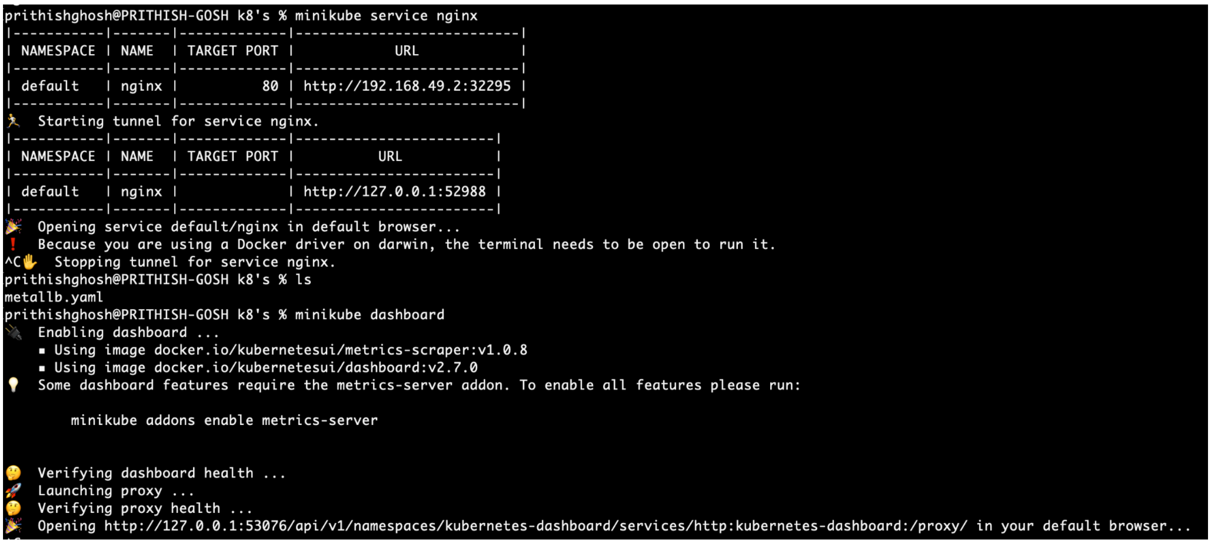
```
prithishghosh@PRITHISH-GOSH k8's % kubectl expose deploy nginx --port 80 --type LoadBalancer
Error from server (AlreadyExists): services "nginx" already exists
prithishghosh@PRITHISH-GOSH k8's %
```

And now checked the svc load balancer assigned ip is showing up

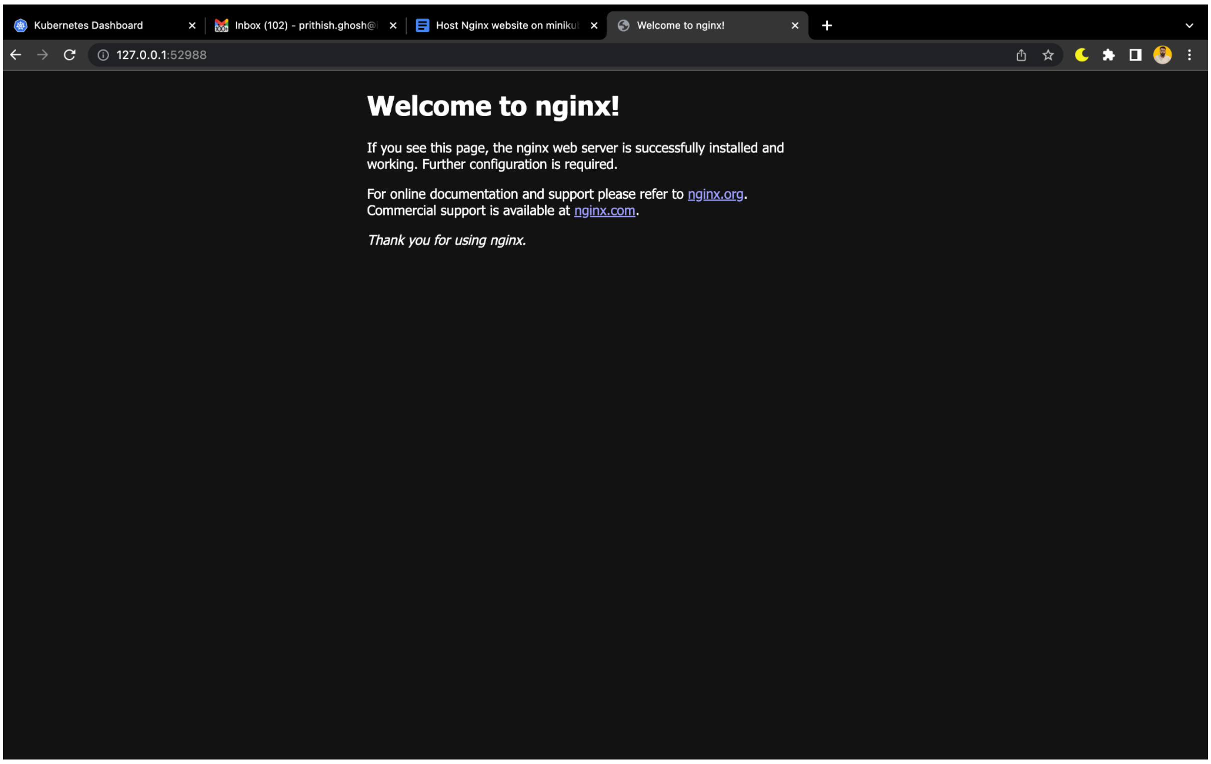


```
prithishghosh@PRITHISH-GOSH k8's % kubectl get svc
NAME         TYPE           CLUSTER-IP      EXTERNAL-IP     PORT(S)        AGE
kubernetes   ClusterIP      10.96.0.1       <none>          443/TCP        128m
nginx        LoadBalancer   10.106.198.34   192.168.1.240   80:32295/TCP   86m
```

Now let's test the svc using tunnel : **minikube service nginx**



```
prithishghosh@PRITHISH-GOSH k8's % minikube service nginx
|-----------|-------|-------------|---------------------------|
| NAMESPACE | NAME  | TARGET PORT |            URL            |
|-----------|-------|-------------|---------------------------|
| default   | nginx |          80 | http://192.168.49.2:32295 |
|-----------|-------|-------------|---------------------------|
🏃  Starting tunnel for service nginx.
|-----------|-------|-------------|------------------------|
| NAMESPACE | NAME  | TARGET PORT |          URL           |
|-----------|-------|-------------|------------------------|
| default   | nginx |             | http://127.0.0.1:52988 |
|-----------|-------|-------------|------------------------|
🎉  Opening service default/nginx in default browser...
❗  Because you are using a Docker driver on darwin, the terminal needs to be open to run it.
^C✋  Stopping tunnel for service nginx.
prithishghosh@PRITHISH-GOSH k8's % ls
metallb.yaml
prithishghosh@PRITHISH-GOSH k8's % minikube dashboard
🔌  Enabling dashboard ...
    ▪ Using image docker.io/kubernetesui/metrics-scraper:v1.0.8
    ▪ Using image docker.io/kubernetesui/dashboard:v2.7.0
💡  Some dashboard features require the metrics-server addon. To enable all features please run:

        minikube addons enable metrics-server


🤔  Verifying dashboard health ...
🚀  Launching proxy ...
🤔  Verifying proxy health ...
🎉  Opening http://127.0.0.1:53076/api/v1/namespaces/kubernetes-dashboard/services/http:kubernetes-dashboard:/proxy/ in your default browser...
```



Welcome to nginx!

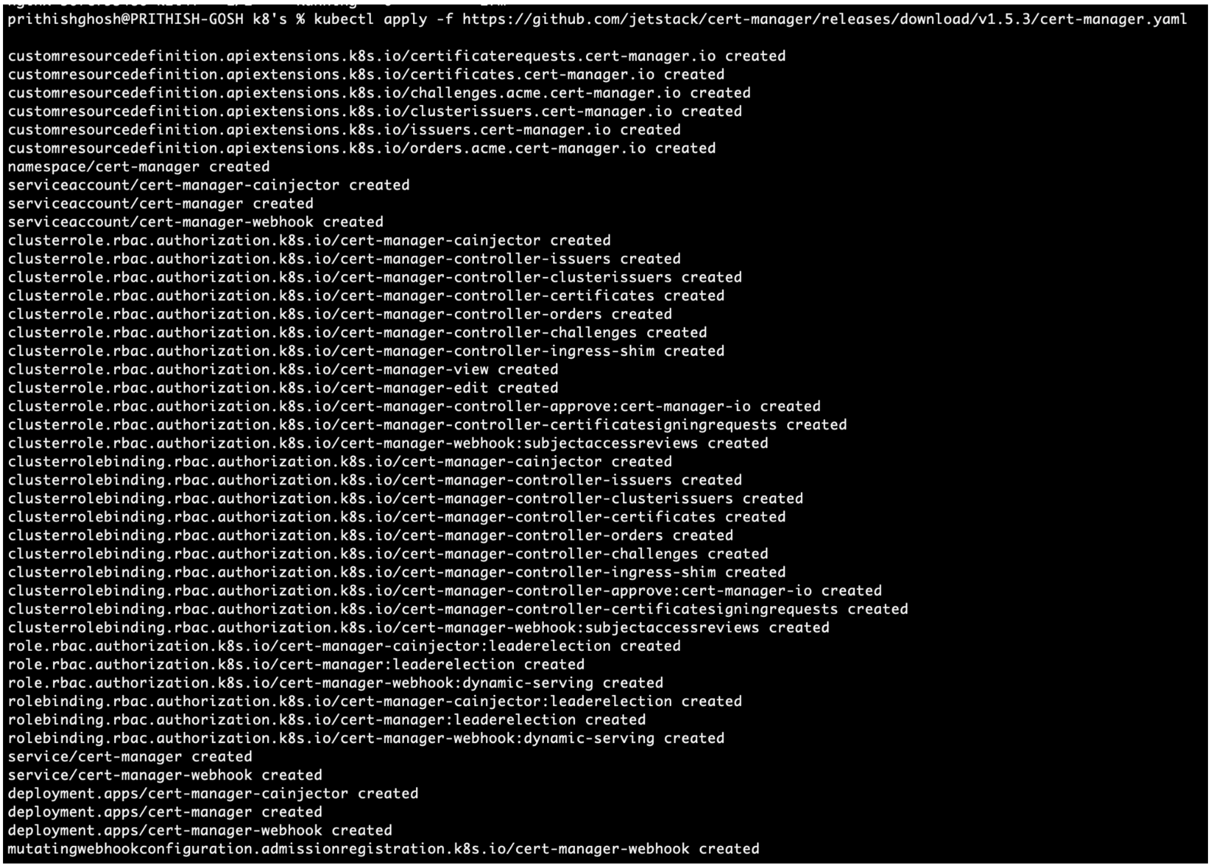
If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

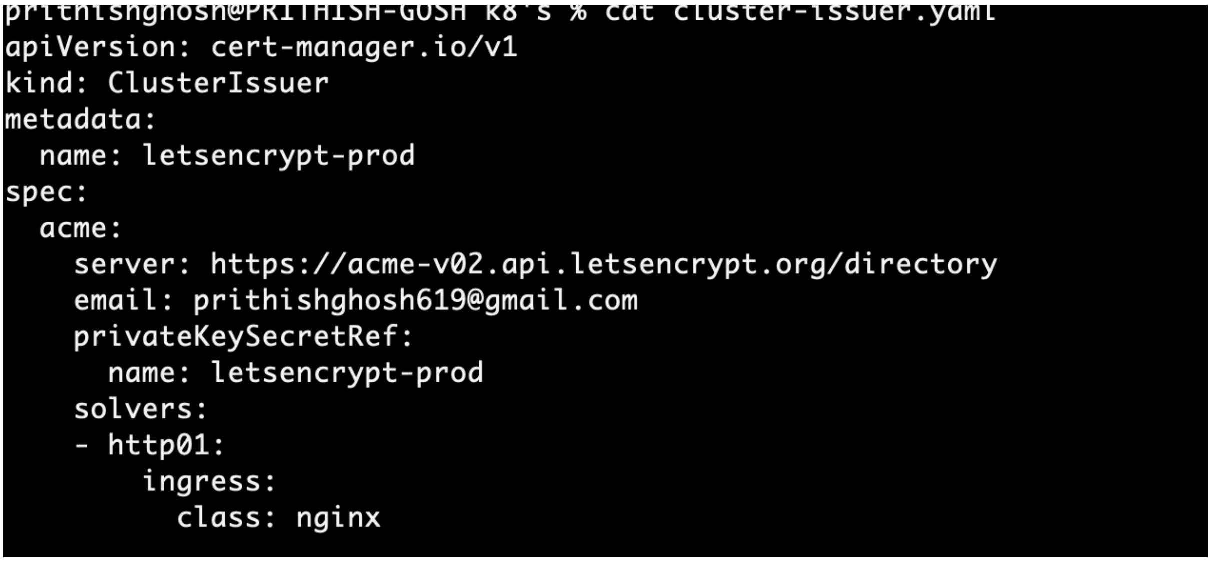
Now installed cert manager CRD’s :

```
kubectl apply -f
https://github.com/jetstack/cert-manager/releases/download/v1.5.3/cert-manager.yaml
```



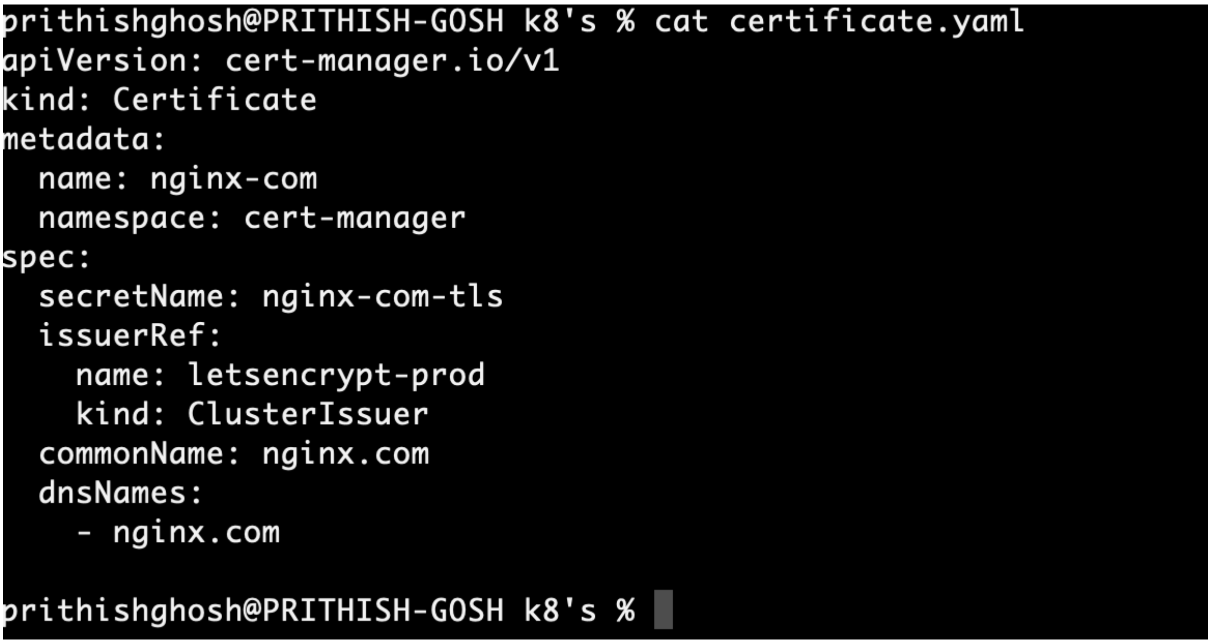
```
prithishghosh@PRITHISH-GOSH k8's % kubectl apply -f https://github.com/jetstack/cert-manager/releases/download/v1.5.3/cert-manager.yaml

customresourcedefinition.apiextensions.k8s.io/certificaterequests.cert-manager.io created
customresourcedefinition.apiextensions.k8s.io/certificates.cert-manager.io created
customresourcedefinition.apiextensions.k8s.io/challenges.acme.cert-manager.io created
customresourcedefinition.apiextensions.k8s.io/clusterissuers.cert-manager.io created
customresourcedefinition.apiextensions.k8s.io/issuers.cert-manager.io created
customresourcedefinition.apiextensions.k8s.io/orders.acme.cert-manager.io created
namespace/cert-manager created
serviceaccount/cert-manager-cainjector created
serviceaccount/cert-manager created
serviceaccount/cert-manager-webhook created
clusterrole.rbac.authorization.k8s.io/cert-manager-cainjector created
clusterrole.rbac.authorization.k8s.io/cert-manager-controller-issuers created
clusterrole.rbac.authorization.k8s.io/cert-manager-controller-clusterissuers created
clusterrole.rbac.authorization.k8s.io/cert-manager-controller-certificates created
clusterrole.rbac.authorization.k8s.io/cert-manager-controller-orders created
clusterrole.rbac.authorization.k8s.io/cert-manager-controller-challenges created
clusterrole.rbac.authorization.k8s.io/cert-manager-controller-ingress-shim created
clusterrole.rbac.authorization.k8s.io/cert-manager-view created
clusterrole.rbac.authorization.k8s.io/cert-manager-edit created
clusterrole.rbac.authorization.k8s.io/cert-manager-controller-approve:cert-manager-io created
clusterrole.rbac.authorization.k8s.io/cert-manager-controller-certificatesigningrequests created
clusterrole.rbac.authorization.k8s.io/cert-manager-webhook:subjectaccessreviews created
clusterrolebinding.rbac.authorization.k8s.io/cert-manager-cainjector created
clusterrolebinding.rbac.authorization.k8s.io/cert-manager-controller-issuers created
clusterrolebinding.rbac.authorization.k8s.io/cert-manager-controller-clusterissuers created
clusterrolebinding.rbac.authorization.k8s.io/cert-manager-controller-certificates created
clusterrolebinding.rbac.authorization.k8s.io/cert-manager-controller-orders created
clusterrolebinding.rbac.authorization.k8s.io/cert-manager-controller-challenges created
clusterrolebinding.rbac.authorization.k8s.io/cert-manager-controller-ingress-shim created
clusterrolebinding.rbac.authorization.k8s.io/cert-manager-controller-approve:cert-manager-io created
clusterrolebinding.rbac.authorization.k8s.io/cert-manager-controller-certificatesigningrequests created
clusterrolebinding.rbac.authorization.k8s.io/cert-manager-webhook:subjectaccessreviews created
role.rbac.authorization.k8s.io/cert-manager-cainjector:leaderelection created
role.rbac.authorization.k8s.io/cert-manager:leaderelection created
role.rbac.authorization.k8s.io/cert-manager-webhook:dynamic-serving created
rolebinding.rbac.authorization.k8s.io/cert-manager-cainjector:leaderelection created
rolebinding.rbac.authorization.k8s.io/cert-manager:leaderelection created
rolebinding.rbac.authorization.k8s.io/cert-manager-webhook:dynamic-serving created
service/cert-manager created
service/cert-manager-webhook created
deployment.apps/cert-manager-cainjector created
deployment.apps/cert-manager created
deployment.apps/cert-manager-webhook created
mutatingwebhookconfiguration.admissionregistration.k8s.io/cert-manager-webhook created
```

And created a cluster issuer config file as well for certificate authorization .



```
prithishghosh@PRITHISH-GOSH k8's % cat cluster-issuer.yaml
apiVersion: cert-manager.io/v1
kind: ClusterIssuer
metadata:
  name: letsencrypt-prod
spec:
  acme:
    server: https://acme-v02.api.letsencrypt.org/directory
    email: prithishghosh619@gmail.com
    privateKeySecretRef:
      name: letsencrypt-prod
    solvers:
    - http01:
        ingress:
          class: nginx
```

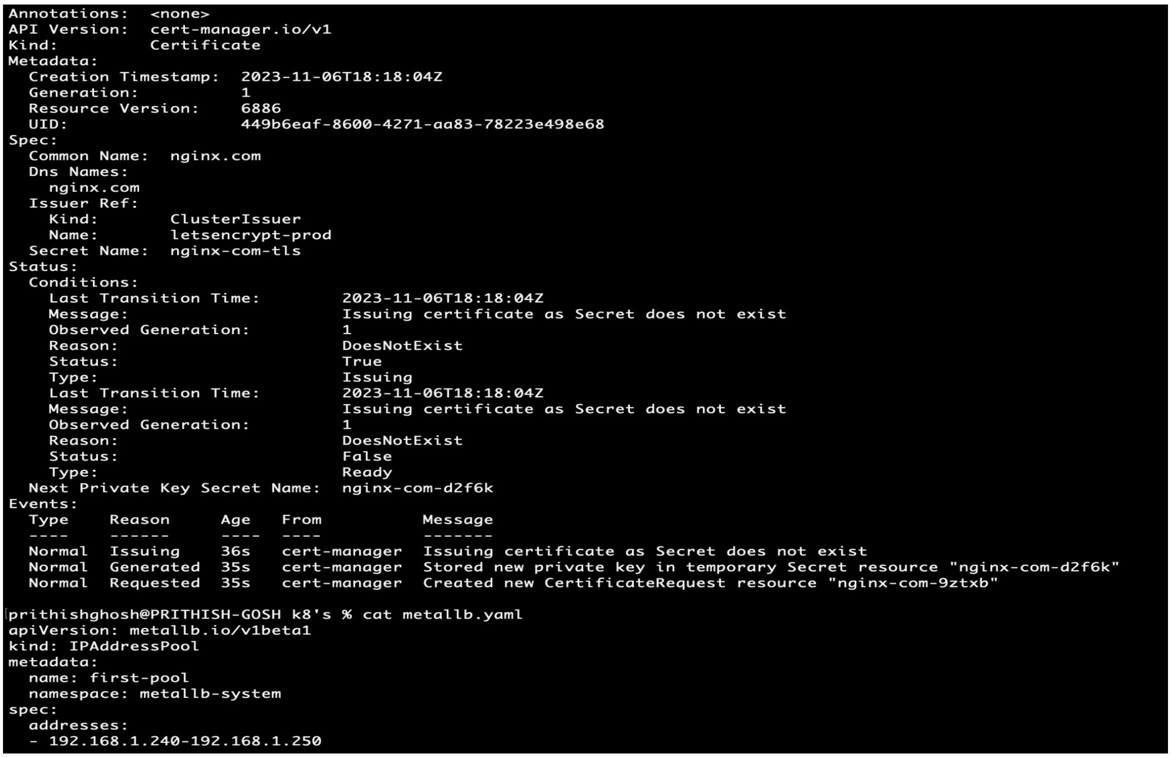
In order to request a certificate, made a certificate resource config file .



```
prithishghosh@PRITHISH-GOSH k8's % cat certificate.yaml
apiVersion: cert-manager.io/v1
kind: Certificate
metadata:
  name: nginx-com
  namespace: cert-manager
spec:
  secretName: nginx-com-tls
  issuerRef:
    name: letsencrypt-prod
    kind: ClusterIssuer
  commonName: nginx.com
  dnsNames:
    - nginx.com

prithishghosh@PRITHISH-GOSH k8's %
```

Now describe the certificate manager details :

```
kubectl describe certificate nginx -n cert-manager
```



```
Annotations:  <none>
API Version:  cert-manager.io/v1
Kind:         Certificate
Metadata:
  Creation Timestamp:  2023-11-06T18:18:04Z
  Generation:          1
  Resource Version:    6886
  UID:                 449b6eaf-8600-4271-aa83-78223e498e68
Spec:
  Common Name:  nginx.com
  Dns Names:
    nginx.com
  Issuer Ref:
    Kind:       ClusterIssuer
    Name:       letsencrypt-prod
  Secret Name:  nginx-com-tls
Status:
  Conditions:
    Last Transition Time:        2023-11-06T18:18:04Z
    Message:                     Issuing certificate as Secret does not exist
    Observed Generation:         1
    Reason:                      DoesNotExist
    Status:                      True
    Type:                        Issuing
    Last Transition Time:        2023-11-06T18:18:04Z
    Message:                     Issuing certificate as Secret does not exist
    Observed Generation:         1
    Reason:                      DoesNotExist
    Status:                      False
    Type:                        Ready
  Next Private Key Secret Name:  nginx-com-d2f6k
Events:
  Type    Reason     Age   From          Message
  ----    ------     ----  ----          -------
  Normal  Issuing    36s   cert-manager  Issuing certificate as Secret does not exist
  Normal  Generated  35s   cert-manager  Stored new private key in temporary Secret resource "nginx-com-d2f6k"
  Normal  Requested  35s   cert-manager  Created new CertificateRequest resource "nginx-com-9ztxb"

prithishghosh@PRITHISH-GOSH k8's % cat metallb.yaml
apiVersion: metallb.io/v1beta1
kind: IPAddressPool
metadata:
  name: first-pool
  namespace: metallb-system
spec:
  addresses:
  - 192.168.1.240-192.168.1.250
```