

CREATING AN INGRESS

Steps for Master

Visit this website


<https://kubernetes.github.io/ingress-nginx/deploy/#bare-metal>

Step 1: Copy and paste the command from the above website and paste it in the terminal

Bare-metal

Using NodePort:

```
kubectl apply -f https://raw.githubusercontent.com/kubernetes/ingress-nginx/control-plane
```

 **Tip**

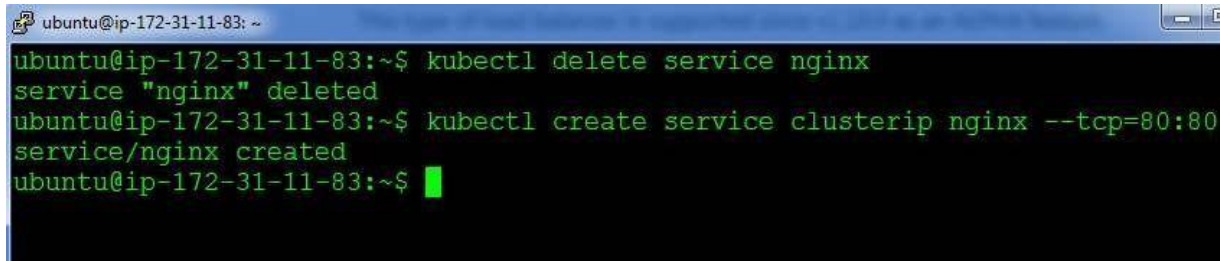
For extended notes regarding deployments on bare-metal, see [Bare-metal considerations](#).

```
ubuntu@ip-172-31-11-83: ~$ kubectl apply -f https://raw.githubusercontent.com/kubernetes/ingress-nginx/master/deploy/mandatory.yaml
namespace/ingress-nginx created
configmap/nginx-configuration created
configmap/tcp-services created
configmap/udp-services created
serviceaccount/nginx-ingress-serviceaccount created
clusterrole.rbac.authorization.k8s.io/nginx-ingress-clusterrole created
role.rbac.authorization.k8s.io/nginx-ingress-role created
rolebinding.rbac.authorization.k8s.io/nginx-ingress-role-nisa-binding created
clusterrolebinding.rbac.authorization.k8s.io/nginx-ingress-clusterrole-nisa-binding created
deployment.apps/nginx-ingress-controller created
ubuntu@ip-172-31-11-83:~$
```

Step 3: Next, since ingress routes to only cluster-ip services, let us delete our previously created nginx nodeport service and create a service with clusterip for nginx. Use the following commands:

```
$ kubectl delete service nginx
```

```
$ kubectl create service clusterip nginx --tcp=80:80
```

A screenshot of a terminal window showing the execution of two kubectl commands. The first command deletes the 'nginx' service, and the second command creates a new 'nginx' service of type 'clusterip' on port 80. The terminal output shows the service being deleted and then created successfully.

```
ubuntu@ip-172-31-11-83: ~  
ubuntu@ip-172-31-11-83:~$ kubectl delete service nginx  
service "nginx" deleted  
ubuntu@ip-172-31-11-83:~$ kubectl create service clusterip nginx --tcp=80:80  
service/nginx created  
ubuntu@ip-172-31-11-83:~$ █
```

Step 4: Next, we will have to create an ingress rule, create an *ingress.yaml* file with the below code:

```
apiVersion: extensions/v1beta1  
  
kind: Ingress  
  
metadata:  
  name: simple-fanout-example  
  
  annotations:  
    nginx.ingress.kubernetes.io/rewrite-target: /  
  
spec:  
  rules:  
  - http:  
    paths:  
    - path: /nginx  
      backend:  
        serviceName: nginx  
        servicePort: 80
```

Step 5: Finally, create the ingress rule using the following command:

```
$ kubectl create -f ingress.yaml
```

```
ubuntu@ip-172-31-11-83: ~  
ubuntu@ip-172-31-11-83:~$ kubectl create -f ingress.yaml  
ingress.extensions/simple-fanout-example created  
ubuntu@ip-172-31-11-83:~$
```

Step 6: Let's verify if ingress is working or not, by checking the nodeport of the ingress service, for checking the nodeport use the following command:

```
$ kubectl get svc -n ingress-nginx
```

```
ubuntu@ip-172-31-11-83: ~  
ubuntu@ip-172-31-11-83:~$ kubectl get svc -n ingress-nginx  
NAME                TYPE        CLUSTER-IP    EXTERNAL-IP    PORT(S)  
ingress-nginx        NodePort    10.98.234.221  <none>         80:30778/TCP,443:31672/TCP  
TCP 10m  
ubuntu@ip-172-31-11-83:~$
```

Step 7: Finally verify by browsing to `https://<IP-address-of-master or slave>:<nodeport>/nginx`

`https://18.219.111.151:31672/nginx`

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.