**Nginx-Ingress Controller Demo**

1. Clone the Ingress controller repo and change into the deployments folder:

**$ git clone https://github.com/nginxinc/kubernetes-ingress/**

**$ cd kubernetes-ingress/deployments**

**$ git checkout v1.10.0**

2. Create a namespace and a service account for the Ingress controller:

**$ kubectl apply -f common/ns-and-sa.yaml**

3. Create a cluster role and cluster role binding for the service account:

**$ kubectl apply -f rbac/rbac.yaml**

4. Create a secret with a TLS certificate and a key for the default server in NGINX:

**$ kubectl apply -f common/default-server-secret.yaml**

5. Create a config map for customizing NGINX configuration:

**$ kubectl apply -f common/nginx-config.yaml**

6. Create an IngressClass resource (for Kubernetes >= 1.18):

**$ kubectl apply -f common/ingress-class.yaml**

7. Create custom resource definitions for VirtualServer and VirtualServerRoute, TransportServer and Policy resources:

**$ kubectl apply -f common/crds/k8s.nginx.org\_virtualservers.yaml**

**$ kubectl apply -f common/crds/k8s.nginx.org\_virtualserverroutes.yaml**

**$ kubectl apply -f common/crds/k8s.nginx.org\_transportservers.yaml**

**$ kubectl apply -f common/crds/k8s.nginx.org\_policies.yaml**

8. Create a custom resource definition for GlobalConfiguration resource:

**$ kubectl apply -f common/crds/k8s.nginx.org\_globalconfigurations.yaml**

9. Create a GlobalConfiguration resource:

**$ kubectl apply -f common/global-configuration.yaml**

10. Run the ingress controller

**$ kubectl apply -f daemon-set/nginx-ingress.yaml**

11. Run the following command to make sure that the Ingress controller pods are running:

**$ kubectl get pods –namespace=nginx-ingress**

Congrats the nginx-Ingress controller is running perfectly file

Now to test it  
  
1) Create the Replication controller, svc and ingress rule in the kubernetes cluster

# vim rc.yml

apiVersion: v1

kind: ReplicationController

metadata:

name: rc1

spec:

replicas: 2

selector:

red: green

template:

metadata:

labels:

red: green

spec:

containers:

- name: cont1

image: nginx

save an quit the file

# kubectl create -f rc.yml

# vim svc.yml

apiVersion: v1

kind: Service

metadata:

name: svc1

spec:

selector:

red: green

ports:

- name: port1

protocol: TCP

port: 8080

targetPort: 80

save and quit the file

# kubectl create -f svc.yml

# vim ingress.yml

apiVersion: networking.k8s.io/v1

kind: Ingress

metadata:

name: ing1

annotations:

kubernetes.io/ingress.class: nginx

spec:

rules:

- host: www.vishal.com

http:

paths:

- path: /

pathType: Prefix

backend:

service:

name: svc1

port:

number: 8080

# save and quit the file

# kubectl create -f ingress.yml

Set your worker node ip on your system hosts file and then open [www.vishal.com](http://www.vishal.com/) on you brower, you should be able to see the content