

Ingress

» Ingress

Deployment & Service

First, create a manifest to deploy an application and expose it via a service.

```

cat > microbot-deployment.yaml <<EOF
apiVersion: apps/v1beta1
kind: Deployment
metadata:
  name: microbot
  labels:
    app: microbot
spec:
  replicas: 3
  template:
    metadata:
      labels:
        app: microbot
        role: frontend
    spec:
      containers:
        - name: microbot
          image: dontrebootme/microbot:v1
          ports:
            - containerPort: 80
---
apiVersion: v1
kind: Service
metadata:
  name: microbot
  labels:
    app: microbot
spec:
  ports:
    - port: 80
      protocol: TCP
      targetPort: 80
  type: NodePort
  selector:
    app: microbot
    role: frontend
EOF

```

Create the Deployment and Service.

```
kubectl create -f microbot-deployment.yaml
```

Verify the Deployment and Service were successfully created.

```
kubectl get deployment,svc -l app=microbot
```

Ingress

Save the hostname for the lab cluster into a shell variable for easy use. Remember to replace `labXX` with your actual lab number.

```
export CLUSTER=labXX.coreostrain.me
echo $CLUSTER
```

Obtain one of the IP addresses for the lab cluster load balancer.

```
host $CLUSTER
```

Save an IP address returned into a shell variable.

```
export ELB_IP=`host $CLUSTER | awk 'NR==1{print $4}'`
echo $ELB_IP
```

Define a shell variable for the DNS name for the application.

```
export MICROBOT_HOST=microbot.$ELB_IP.xip.io
echo $MICROBOT_HOST
```

Create a manifest file that will define an Ingress resource to be used by the Tectonic Nginx Ingress Controller. This controller is automatically provisioned during Tectonic installation.

```
cat > microbot-ingress.yaml <<EOF
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  name: microbot
  annotations:
    kubernetes.io/ingress.class: "tectonic"
spec:
  rules:
    - host: $MICROBOT_HOST
      http:
        paths:
          - path: /
            backend:
              serviceName: microbot
              servicePort: 80
EOF
```

Create the Ingress resource.

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

Verify that the Ingress resource was successfully created.

```
kubectl get ing
kubectl describe ing microbot
```

Open up your web browser and navigate to `http://<MICROBOT_HOST>/` .

Logs

Follow the logs from the Tectonic Ingress Controller pod to see your request in real-time.

The following command will list all of the pods inside the `tectonic-system` namespace.

```
kubectl get pods --namespace tectonic-system
```

Filter the list down to just the ingress controller pod.

```
kubectl get pods -n tectonic-system -l component=ingress-controller
POD=$(kubectl get pods -n tectonic-system -l component=ingress-controller --output jsonpath={.items..metadata.name})
echo $POD
```

View the logs for the Tectonic Ingress Controller pod.

```
kubectl logs -n tectonic-system -f $POD
```

For the curious, check out the `nginx.conf` file that resides inside the Tectonic Ingress Controller pod.

```
kubectl exec -n tectonic-system -it $POD cat /etc/nginx/nginx.conf
```

Clean Up

Clean up the `microbot` resources.

```
kubectl delete deploy,svc,ing microbot
```

View any remaining resources.

```
kubectl get all
```