Terminal

```
Your Interactive Bash Terminal. A safe place to learn and execute commands.

controlplane $ launch.sh
Waiting for Kubernetes to start...
Kubernetes started
controlplane $
controlplane $
controlplane $ kubectl apply -f clusterip.yaml
service/webapp1-clusterip-svc created
deployment.extensions/webapp1-clusterip-deployment created
controlplane $ cat clusterip.yaml
apiVersion: v1
kind: Service
metadata:
  name: webapp1-clusterip-svc
  labels:
    app: webapp1-clusterip
spec:
  ports:
  - port: 80
  selector:
    app: webapp1-clusterip
---
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  name: webapp1-clusterip-deployment
spec:
  replicas: 2
  template:
    metadata:
      labels:
        app: webapp1-clusterip
    spec:
      containers:
      - name: webapp1-clusterip-pod
        image: katacoda/docker-http-server:latest
        ports:
        - containerPort: 80
---
controlplane $ kubectl get pods
NAME                                          READY   STATUS    RESTARTS   AGE
webapp1-clusterip-deployment-669c7c65c4-bmfm2   1/1     Running   0          8s
```

Terminal

```
controlplane $ kubectl get pods
NAME                                          READY   STATUS    RESTARTS   AGE
webapp1-clusterip-deployment-669c7c65c4-bmfm2   1/1     Running   0          8s
webapp1-clusterip-deployment-669c7c65c4-xfshf   1/1     Running   0          8s
controlplane $ kubectl get svc
NAME                    TYPE        CLUSTER-IP    EXTERNAL-IP   PORT(S)   AGE
kubernetes              ClusterIP   10.96.0.1     <none>        443/TCP   2m47s
webapp1-clusterip-svc   ClusterIP   10.98.82.65   <none>        80/TCP    11s
controlplane $ kubectl describe svc/webapp1-clusterip-svc
Name:              webapp1-clusterip-svc
Namespace:         default
Labels:            app=webapp1-clusterip
Annotations:       kubectl.kubernetes.io/last-applied-configuration:
                     {"apiVersion":"v1","kind":"Service","metadata":{"annotations":{},"labels":{"app":"webapp1-clusterip"},"name":"webapp1-clusterip-
svc","name...
Selector:          app=webapp1-clusterip
Type:              ClusterIP
IP:                10.98.82.65
Port:              <unset>  80/TCP
TargetPort:        80/TCP
Endpoints:         10.32.0.5:80,10.32.0.6:80
Session Affinity:  None
Events:            <none>
controlplane $ export CLUSTER_IP=$(kubectl get services/webapp1-clusterip-svc -o go-template='{{(index .spec.clusterIP)}}')
controlplane $ echo CLUSTER_IP=$CLUSTER_IP
CLUSTER_IP=10.98.82.65
controlplane $ curl $CLUSTER_IP:80
<h1>This request was processed by host: webapp1-clusterip-deployment-669c7c65c4-bmfm2</h1>
controlplane $ curl $CLUSTER_IP:80
<h1>This request was processed by host: webapp1-clusterip-deployment-669c7c65c4-xfshf</h1>
controlplane $ kubectl apply -f clusterip-target.yaml
service/webapp1-clusterip-targetport-svc created
deployment.extensions/webapp1-clusterip-targetport-deployment created
controlplane $ cat clusterip-target.yaml
apiVersion: v1
kind: Service
metadata:
  name: webapp1-clusterip-targetport-svc
  labels:
    app: webapp1-clusterip-targetport
spec:
  ports:
  - port: 8080
```

Terminal

```
    app: webapp1-clusterip-targetport
spec:
  ports:
  - port: 8080
    targetPort: 80
  selector:
    app: webapp1-clusterip-targetport
---
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  name: webapp1-clusterip-targetport-deployment
spec:
  replicas: 2
  template:
    metadata:
      labels:
        app: webapp1-clusterip-targetport
    spec:
      containers:
      - name: webapp1-clusterip-targetport-pod
        image: katacoda/docker-http-server:latest
        ports:
        - containerPort: 80
---
controlplane $ kubectl get svc
NAME                              TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)    AGE
kubernetes                        ClusterIP   10.96.0.1       <none>        443/TCP    3m20s
webapp1-clusterip-svc             ClusterIP   10.98.82.65     <none>        80/TCP     44s
webapp1-clusterip-targetport-svc  ClusterIP   10.103.70.206   <none>        8080/TCP   10s
controlplane $ kubectl describe svc/webapp1-clusterip-targetport-svc
Name:              webapp1-clusterip-targetport-svc
Namespace:         default
Labels:            app=webapp1-clusterip-targetport
Annotations:       kubectl.kubernetes.io/last-applied-configuration:
                     {"apiVersion":"v1","kind":"Service","metadata":{"annotations":{},"labels":{"app":"webapp1-clusterip-targetport"},"name":"webapp1
-clusterip...
Selector:          app=webapp1-clusterip-targetport
Type:              ClusterIP
IP:                10.103.70.206
Port:              <unset>  8080/TCP
TargetPort:        80/TCP
Endpoints:         10.32.0.7:80,10.32.0.8:80
```

Terminal

```
TargetPort:        80/TCP
Endpoints:         10.32.0.7:80,10.32.0.8:80
Session Affinity:  None
Events:            <none>
controlplane $ export CLUSTER_IP=$(kubectl get services/webapp1-clusterip-targetport-svc -o go-template='{{(index .spec.clusterIP)}}')
controlplane $ echo CLUSTER_IP=$CLUSTER_IP
CLUSTER_IP=10.103.70.206
controlplane $ curl $CLUSTER_IP:8080
<h1>This request was processed by host: webapp1-clusterip-targetport-deployment-5599945ff4-xgd8g</h1>
controlplane $ curl $CLUSTER_IP:8080
<h1>This request was processed by host: webapp1-clusterip-targetport-deployment-5599945ff4-xgd8g</h1>
controlplane $ kubectl apply -f nodeport.yaml
service/webapp1-nodeport-svc created
deployment.extensions/webapp1-nodeport-deployment created
controlplane $ cat nodeport.yaml
apiVersion: v1
kind: Service
metadata:
  name: webapp1-nodeport-svc
  labels:
    app: webapp1-nodeport
spec:
  type: NodePort
  ports:
  - port: 80
    nodePort: 30080
  selector:
    app: webapp1-nodeport
---
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  name: webapp1-nodeport-deployment
spec:
  replicas: 2
  template:
    metadata:
      labels:
        app: webapp1-nodeport
    spec:
      containers:
      - name: webapp1-nodeport-pod
        image: katacoda/docker-http-server:latest
```

Terminal

```
      app: webapp1-nodeport
  spec:
    containers:
    - name: webapp1-nodeport-pod
      image: katacoda/docker-http-server:latest
      ports:
      - containerPort: 80
---
controlplane $ kubectl get svc
NAME                            TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)        AGE
kubernetes                      ClusterIP   10.96.0.1       <none>        443/TCP        3m49s
webapp1-clusterip-svc           ClusterIP   10.98.82.65     <none>        80/TCP         73s
webapp1-clusterip-targetport-svc   ClusterIP   10.103.70.206   <none>        8080/TCP       39s
webapp1-nodeport-svc            NodePort    10.97.219.195   <none>        80:30080/TCP   12s
controlplane $ kubectl describe svc/webapp1-nodeport-svc
Name:                     webapp1-nodeport-svc
Namespace:                default
Labels:                   app=webapp1-nodeport
Annotations:              kubectl.kubernetes.io/last-applied-configuration:
                            {"apiVersion":"v1","kind":"Service","metadata":{"annotations":{},"labels":{"app":"webapp1-nodeport"},"name":"webapp1-node
port-svc","namesp...
Selector:                 app=webapp1-nodeport
Type:                     NodePort
IP:                       10.97.219.195
Port:                     <unset>  80/TCP
TargetPort:               80/TCP
NodePort:                 <unset>  30080/TCP
Endpoints:                10.32.0.10:80,10.32.0.9:80
Session Affinity:         None
External Traffic Policy:  Cluster
Events:                   <none>
controlplane $ curl 172.17.0.32:30080
<h1>This request was processed by host: webapp1-nodeport-deployment-677bd89b96-4vzvx</h1>
controlplane $ sed -i 's/HOSTIP/172.17.0.32/g' externalip.yaml
controlplane $ cat externalip.yaml
apiVersion: v1
kind: Service
metadata:
  name: webapp1-externalip-svc
  labels:
    app: webapp1-externalip
spec:
  ports:
```

Terminal

```
    app: webapp1-externalip
spec:
  ports:
  - port: 80
  externalIPs:
  - 172.17.0.32
  selector:
    app: webapp1-externalip
---
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  name: webapp1-externalip-deployment
spec:
  replicas: 2
  template:
    metadata:
      labels:
        app: webapp1-externalip
    spec:
      containers:
      - name: webapp1-externalip-pod
        image: katacoda/docker-http-server:latest
        ports:
        - containerPort: 80
---
controlplane $ kubectl apply -f externalip.yaml
service/webapp1-externalip-svc created
deployment.extensions/webapp1-externalip-deployment created
controlplane $ kubectl get svc
NAME                              TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)        AGE
kubernetes                        ClusterIP   10.96.0.1       <none>        443/TCP        5m1s
webapp1-clusterip-svc             ClusterIP   10.98.82.65     <none>        80/TCP         2m25s
webapp1-clusterip-targetport-svc  ClusterIP   10.103.70.206   <none>        8080/TCP       111s
webapp1-externalip-svc            ClusterIP   10.102.108.7    172.17.0.32   80/TCP         4s
webapp1-nodeport-svc              NodePort    10.97.219.195   <none>        80:30080/TCP   84s
controlplane $ kubectl describe svc/webapp1-externalip-svc
Name:              webapp1-externalip-svc
Namespace:         default
Labels:            app=webapp1-externalip
Annotations:       kubectl.kubernetes.io/last-applied-configuration:
                     {"apiVersion":"v1","kind":"Service","metadata":{"annotations":{},"labels":{"app":"webapp1-externalip"},"name":"webapp1-externali
p-svc","na...
```

Terminal

```
webapp1-nodeport-svc              NodePort    10.97.219.195   <none>        80:30080/TCP   84s
controlplane $ kubectl describe svc/webapp1-externalip-svc
Name:              webapp1-externalip-svc
Namespace:         default
Labels:            app=webapp1-externalip
Annotations:       kubectl.kubernetes.io/last-applied-configuration:
                     {"apiVersion":"v1","kind":"Service","metadata":{"annotations":{},"labels":{"app":"webapp1-externalip"},"name":"webapp1-externali
p-svc","na...
Selector:          app=webapp1-externalip
Type:              ClusterIP
IP:                10.102.108.7
External IPs:      172.17.0.32
Port:              <unset>  80/TCP
TargetPort:        80/TCP
Endpoints:         10.32.0.11:80,10.32.0.12:80
Session Affinity:  None
Events:            <none>
controlplane $ curl 172.17.0.32
<h1>This request was processed by host: webapp1-externalip-deployment-6446b488f8-b9h82</h1>
controlplane $ kubectl apply -f cloudprovider.yaml
daemonset.extensions/kube-keepalived-vip configured
configmap/vip-configmap configured
deployment.apps/keepalived-cloud-provider created
controlplane $ kubectl get pods -n kube-system
NAME                                      READY   STATUS    RESTARTS   AGE
coredns-fb8b8dccf-59vnc                   1/1     Running   0          5m7s
coredns-fb8b8dccf-ptwh5                   1/1     Running   0          5m7s
etcd-controlplane                         1/1     Running   0          4m4s
katacoda-cloud-provider-7c895f7d5-7zp8t   1/1     Running   0          5m6s
keepalived-cloud-provider-78fc4468b-t5vn4 1/1     Running   0          4s
kube-apiserver-controlplane               1/1     Running   0          4m16s
kube-controller-manager-controlplane      1/1     Running   0          4m9s
kube-keepalived-vip-csdbn                 1/1     Running   0          4m20s
kube-proxy-cx57m                          1/1     Running   0          5m7s
kube-scheduler-controlplane               1/1     Running   0          4m18s
weave-net-mwqqm                           2/2     Running   1          5m7s
controlplane $ kubectl apply -f loadbalancer.yaml
service/webapp1-loadbalancer-svc created
deployment.extensions/webapp1-loadbalancer-deployment created
controlplane $ cat loadbalancer.yaml
apiVersion: v1
kind: Service
metadata:
```

Terminal

```
controlplane $ kubectl apply -f loadbalancer.yaml
service/webapp1-loadbalancer-svc created
deployment.extensions/webapp1-loadbalancer-deployment created
controlplane $ cat loadbalancer.yaml
apiVersion: v1
kind: Service
metadata:
  name: webapp1-loadbalancer-svc
  labels:
    app: webapp1-loadbalancer
spec:
  type: LoadBalancer
  ports:
  - port: 80
  selector:
    app: webapp1-loadbalancer
---
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  name: webapp1-loadbalancer-deployment
spec:
  replicas: 2
  template:
    metadata:
      labels:
        app: webapp1-loadbalancer
    spec:
      containers:
      - name: webapp1-loadbalancer-pod
        image: katacoda/docker-http-server:latest
        ports:
        - containerPort: 80
---
controlplane $ kubectl get svc
NAME                              TYPE           CLUSTER-IP      EXTERNAL-IP   PORT(S)        AGE
kubernetes                        ClusterIP      10.96.0.1       <none>        443/TCP        5m33s
webapp1-clusterip-svc             ClusterIP      10.98.82.65     <none>        80/TCP         2m57s
webapp1-clusterip-targetport-svc  ClusterIP      10.103.70.206   <none>        8080/TCP       2m23s
webapp1-externalip-svc            ClusterIP      10.102.108.7    172.17.0.32   80/TCP         36s
webapp1-loadbalancer-svc          LoadBalancer   10.110.120.41   172.17.0.32   80:31459/TCP   11s
webapp1-nodeport-svc              NodePort       10.97.219.195   <none>        80:30080/TCP   116s
controlplane $ kubectl describe svc/webapp1-loadbalancer-svc
```

Terminal

```
      - name: webapp1-loadbalancer-pod
        image: katacoda/docker-http-server:latest
        ports:
        - containerPort: 80
---
controlplane $ kubectl get svc
NAME                              TYPE           CLUSTER-IP      EXTERNAL-IP   PORT(S)        AGE
kubernetes                        ClusterIP      10.96.0.1       <none>        443/TCP        5m33s
webapp1-clusterip-svc             ClusterIP      10.98.82.65     <none>        80/TCP         2m57s
webapp1-clusterip-targetport-svc  ClusterIP      10.103.70.206   <none>        8080/TCP       2m23s
webapp1-externalip-svc            ClusterIP      10.102.108.7    172.17.0.32   80/TCP         36s
webapp1-loadbalancer-svc          LoadBalancer   10.110.120.41   172.17.0.32   80:31459/TCP   11s
webapp1-nodeport-svc              NodePort       10.97.219.195   <none>        80:30080/TCP   116s
controlplane $ kubectl describe svc/webapp1-loadbalancer-svc
Name:                     webapp1-loadbalancer-svc
Namespace:                default
Labels:                   app=webapp1-loadbalancer
Annotations:              kubectl.kubernetes.io/last-applied-configuration:
                            {"apiVersion":"v1","kind":"Service","metadata":{"annotations":{},"labels":{"app":"webapp1-loadbalancer"},"name":"webapp1-
loadbalancer-svc"...
Selector:                 app=webapp1-loadbalancer
Type:                     LoadBalancer
IP:                       10.110.120.41
LoadBalancer Ingress:     172.17.0.32
Port:                     <unset>  80/TCP
TargetPort:               80/TCP
NodePort:                 <unset>  31459/TCP
Endpoints:                10.32.0.14:80,10.32.0.15:80
Session Affinity:         None
External Traffic Policy:  Cluster
Events:
  Type    Reason                Age   From                Message
  ----    ------                ----  ----                -------
  Normal  CreatingLoadBalancer  16s   service-controller  Creating load balancer
  Normal  CreatedLoadBalancer   16s   service-controller  Created load balancer
controlplane $ export LoadBalancerIP=$(kubectl get services/webapp1-loadbalancer-svc -o go-template='{{(index .status.loadBalancer.ingress 0).ip}}')
controlplane $ echo LoadBalancerIP=$LoadBalancerIP
LoadBalancerIP=172.17.0.32
controlplane $ curl $LoadBalancerIP
<h1>This request was processed by host: webapp1-loadbalancer-deployment-f45b8d9cd-p6hkb</h1>
controlplane $ curl $LoadBalancerIP
<h1>This request was processed by host: webapp1-loadbalancer-deployment-f45b8d9cd-p6hkb</h1>
controlplane $
```