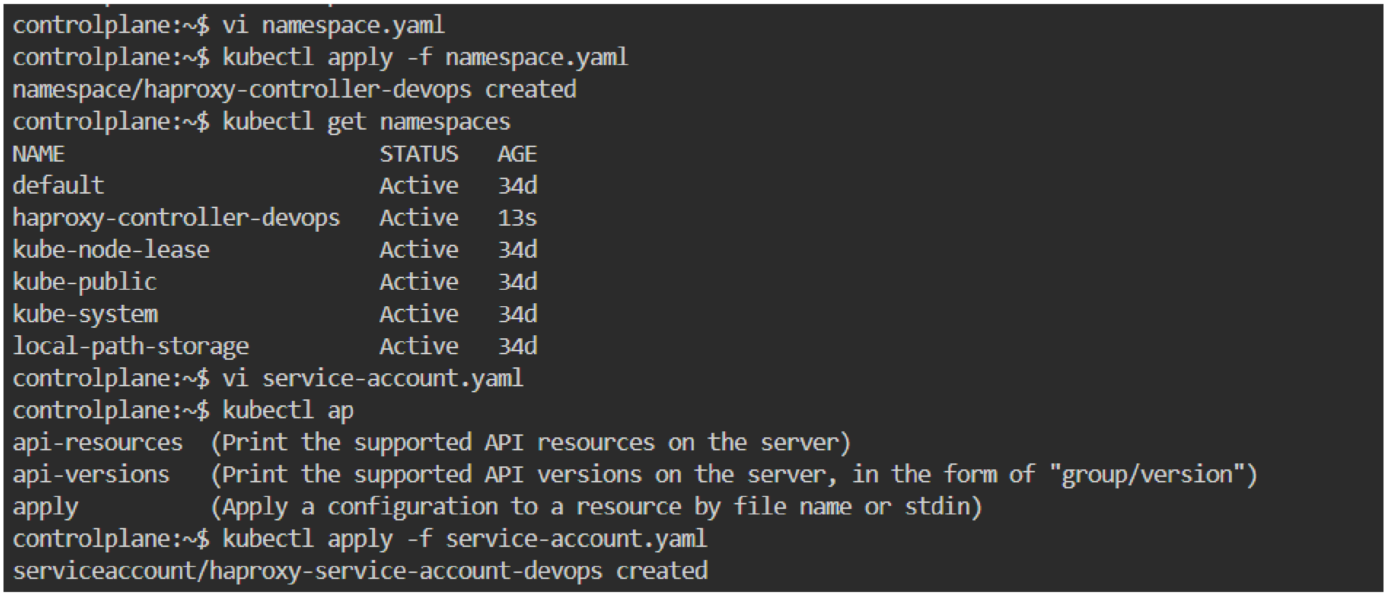
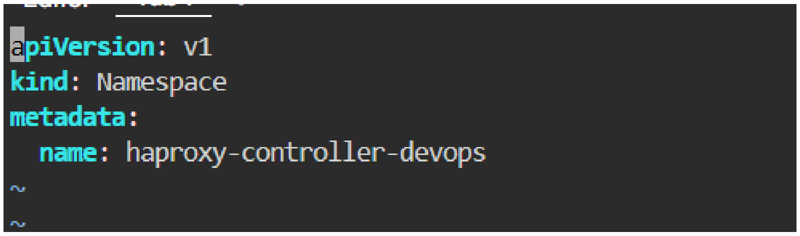
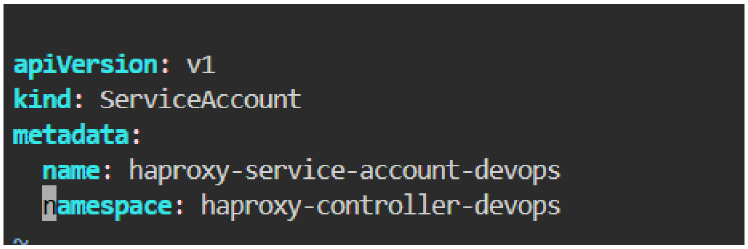


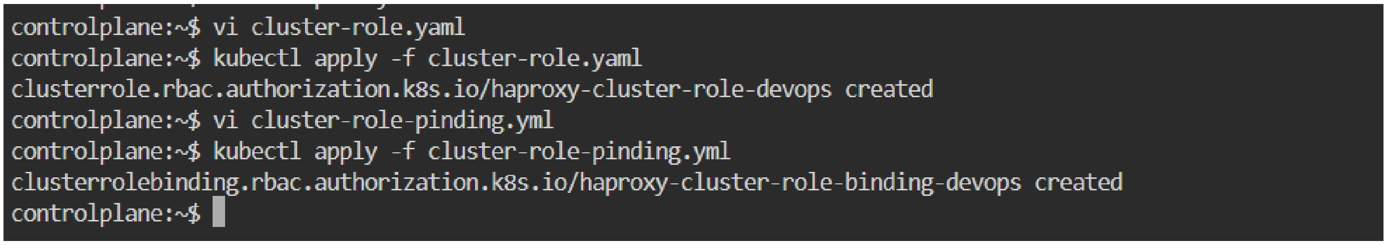
```
controlplane:~$ vi namespace.yaml
controlplane:~$ kubectl apply -f namespace.yaml
namespace/haproxy-controller-devops created
controlplane:~$ kubectl get namespaces
NAME                        STATUS   AGE
default                     Active   34d
haproxy-controller-devops   Active   13s
kube-node-lease             Active   34d
kube-public                 Active   34d
kube-system                 Active   34d
local-path-storage          Active   34d
controlplane:~$ vi service-account.yaml
controlplane:~$ kubectl ap
api-resources  (Print the supported API resources on the server)
api-versions   (Print the supported API versions on the server, in the form of "group/version")
apply          (Apply a configuration to a resource by file name or stdin)
controlplane:~$ kubectl apply -f service-account.yaml
serviceaccount/haproxy-service-account-devops created
```



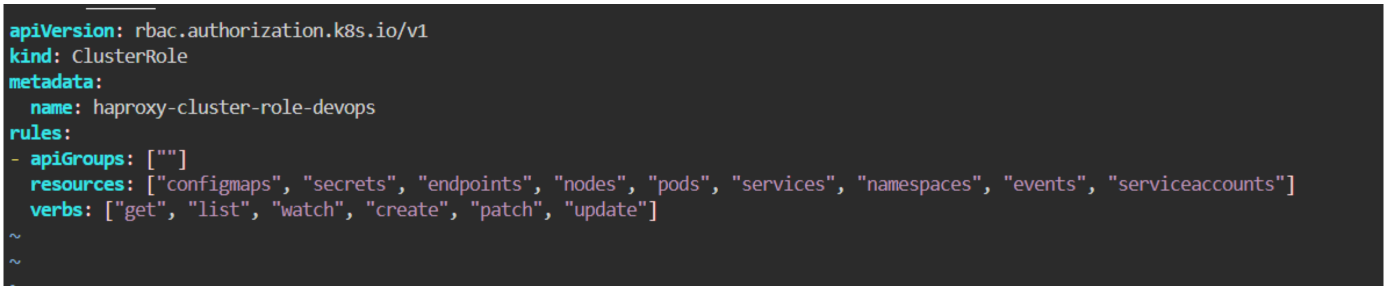
```
apiVersion: v1
kind: Namespace
metadata:
  name: haproxy-controller-devops
~
~
```



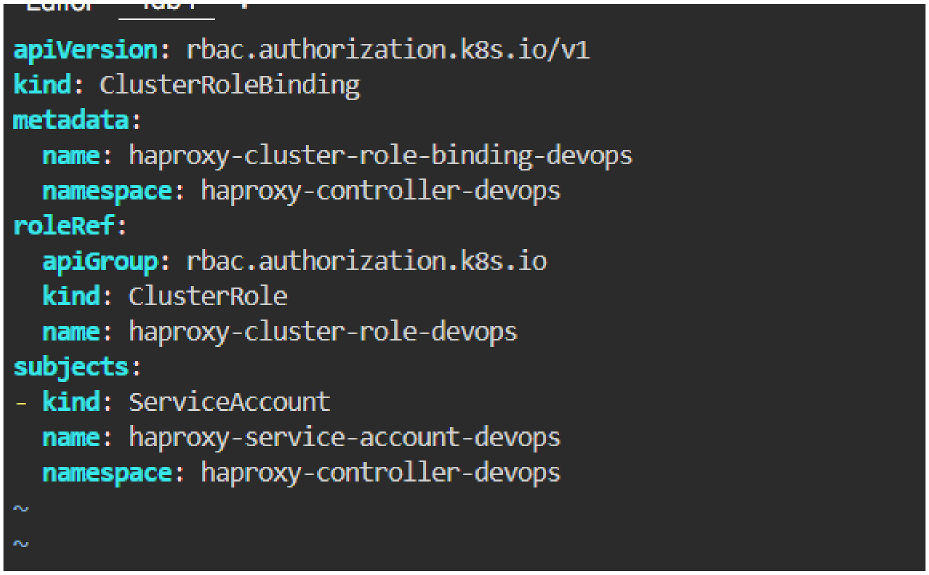
```
apiVersion: v1
kind: ServiceAccount
metadata:
  name: haproxy-service-account-devops
  namespace: haproxy-controller-devops
~
```



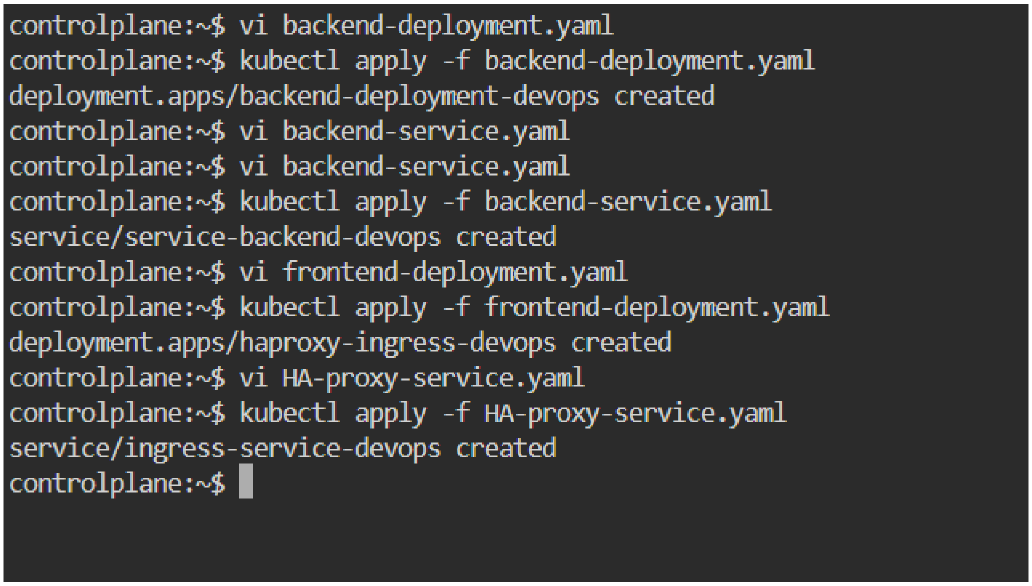
```
controlplane:~$ vi cluster-role.yaml
controlplane:~$ kubectl apply -f cluster-role.yaml
clusterrole.rbac.authorization.k8s.io/haproxy-cluster-role-devops created
controlplane:~$ vi cluster-role-pinding.yml
controlplane:~$ kubectl apply -f cluster-role-pinding.yml
clusterrolebinding.rbac.authorization.k8s.io/haproxy-cluster-role-binding-devops created
controlplane:~$ 
```



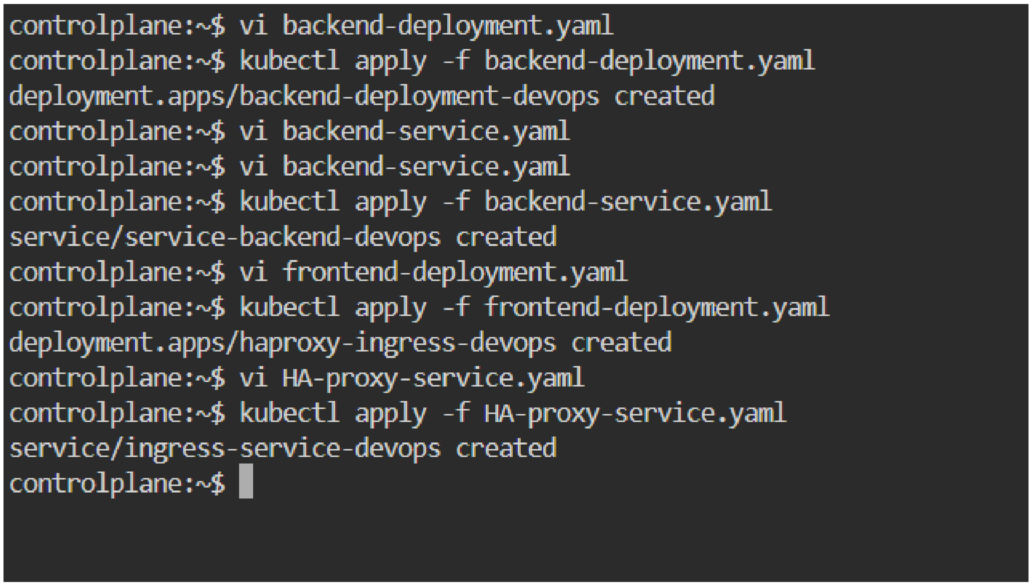
```
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: haproxy-cluster-role-devops
rules:
- apiGroups: [""]
  resources: ["configmaps", "secrets", "endpoints", "nodes", "pods", "services", "namespaces", "events", "serviceaccounts"]
  verbs: ["get", "list", "watch", "create", "patch", "update"]
~
~
```



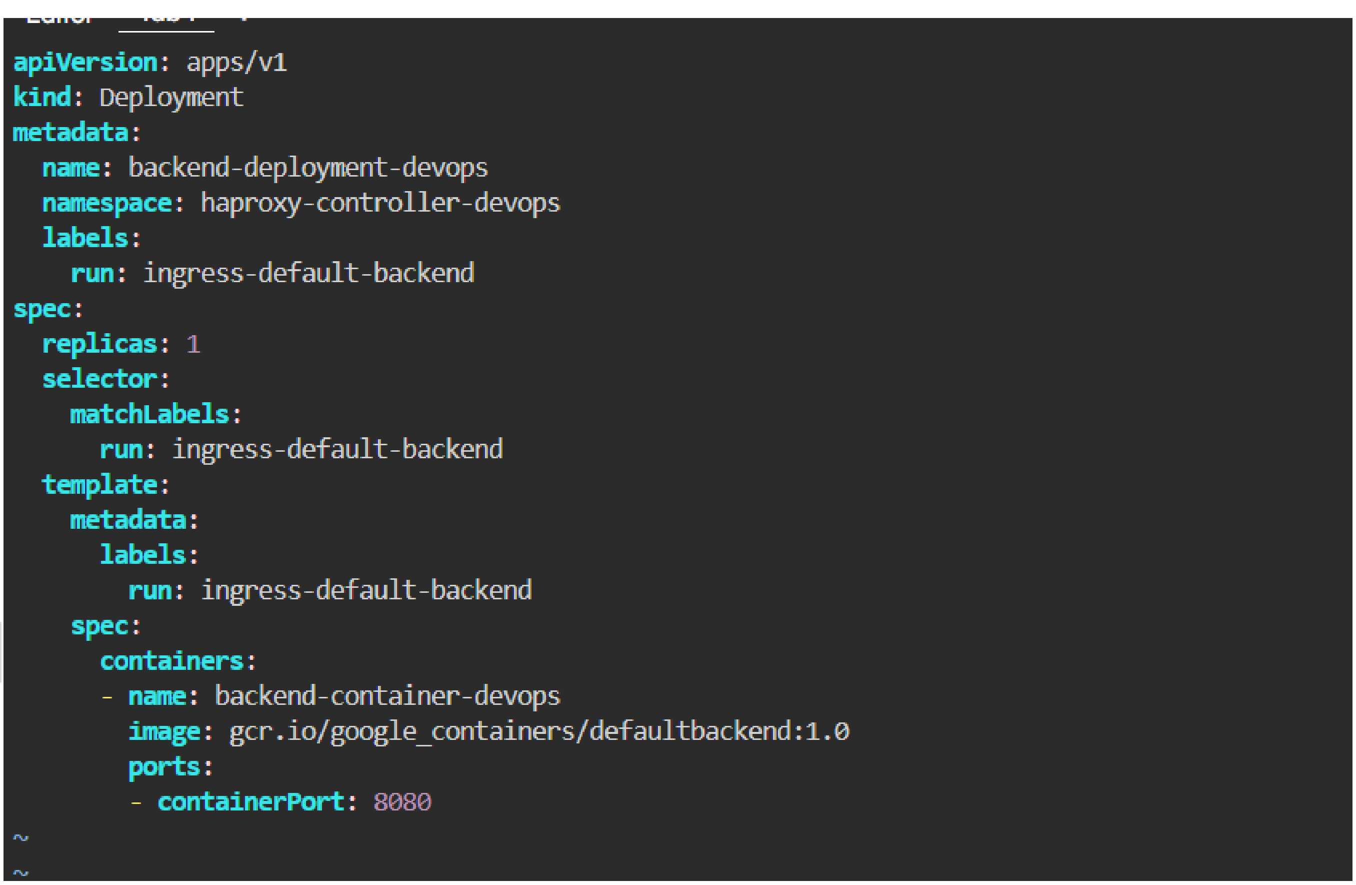
```
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  name: haproxy-cluster-role-binding-devops
  namespace: haproxy-controller-devops
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: haproxy-cluster-role-devops
subjects:
- kind: ServiceAccount
  name: haproxy-service-account-devops
  namespace: haproxy-controller-devops
~
~
```



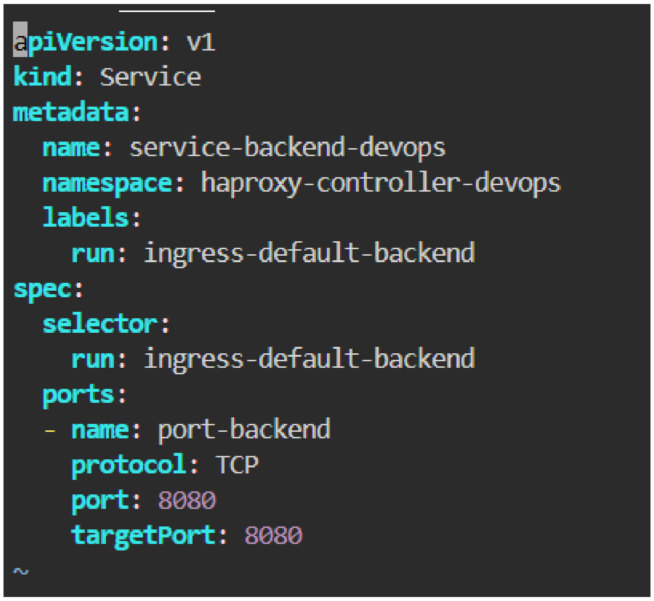
```
controlplane:~$ vi backend-deployment.yaml
controlplane:~$ kubectl apply -f backend-deployment.yaml
deployment.apps/backend-deployment-devops created
controlplane:~$ vi backend-service.yaml
controlplane:~$ vi backend-service.yaml
controlplane:~$ kubectl apply -f backend-service.yaml
service/service-backend-devops created
controlplane:~$ vi frontend-deployment.yaml
controlplane:~$ kubectl apply -f frontend-deployment.yaml
deployment.apps/haproxy-ingress-devops created
controlplane:~$ vi HA-proxy-service.yaml
controlplane:~$ kubectl apply -f HA-proxy-service.yaml
service/ingress-service-devops created
controlplane:~$ 
```



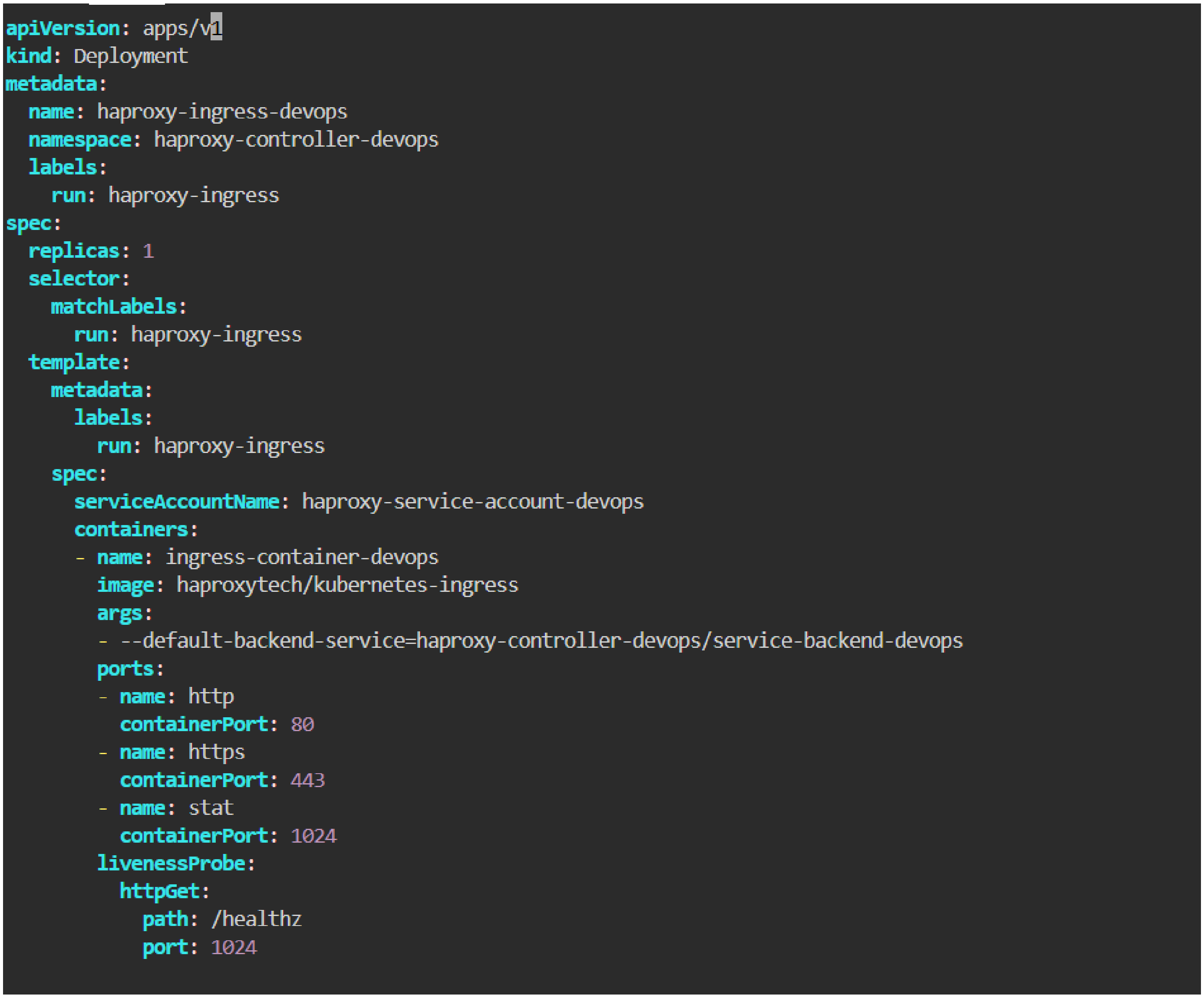
```
controlplane:~$ vi backend-deployment.yaml
controlplane:~$ kubectl apply -f backend-deployment.yaml
deployment.apps/backend-deployment-devops created
controlplane:~$ vi backend-service.yaml
controlplane:~$ vi backend-service.yaml
controlplane:~$ kubectl apply -f backend-service.yaml
service/service-backend-devops created
controlplane:~$ vi frontend-deployment.yaml
controlplane:~$ kubectl apply -f frontend-deployment.yaml
deployment.apps/haproxy-ingress-devops created
controlplane:~$ vi HA-proxy-service.yaml
controlplane:~$ kubectl apply -f HA-proxy-service.yaml
service/ingress-service-devops created
controlplane:~$ 
```



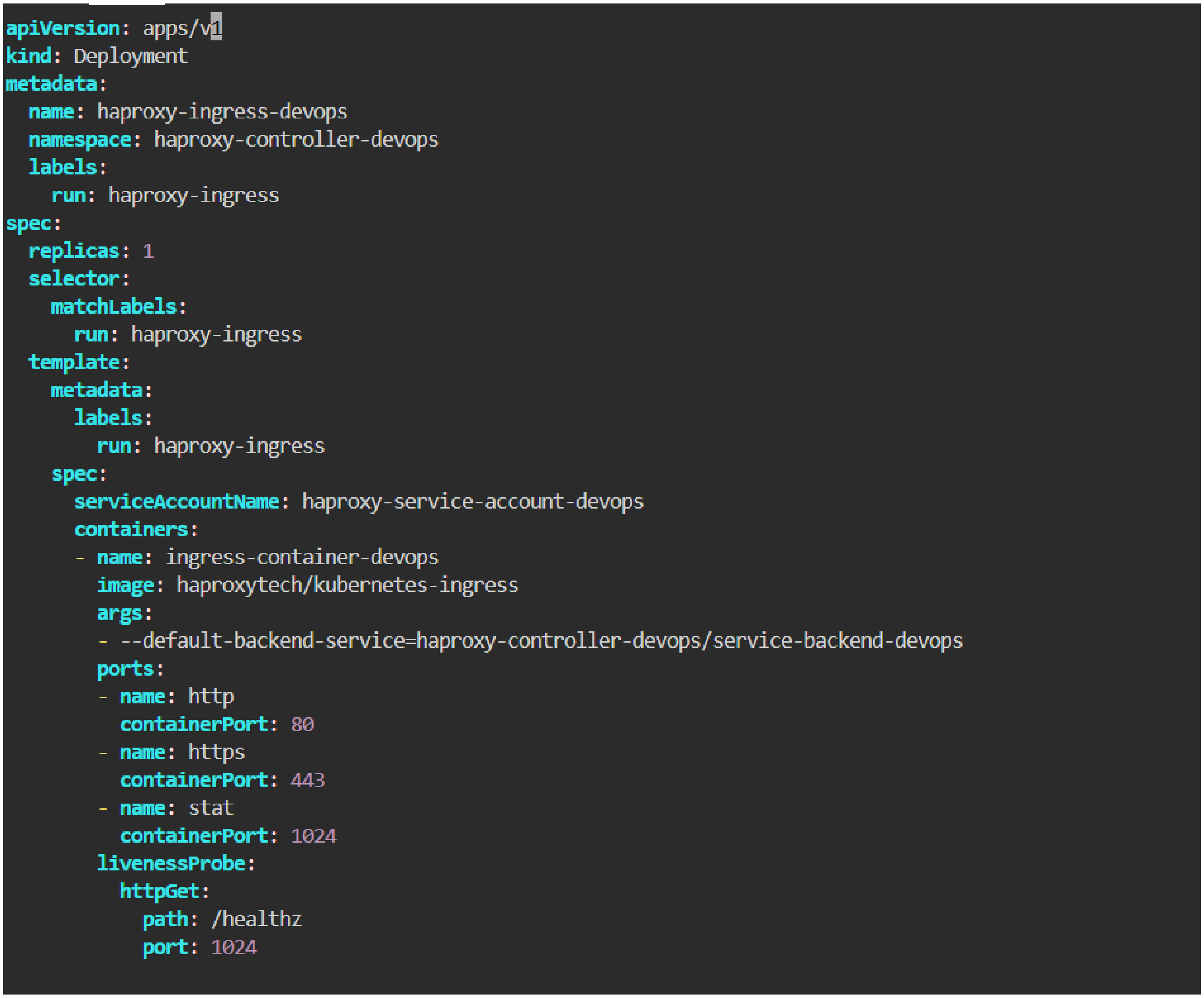
```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: backend-deployment-devops
  namespace: haproxy-controller-devops
  labels:
    run: ingress-default-backend
spec:
  replicas: 1
  selector:
    matchLabels:
      run: ingress-default-backend
  template:
    metadata:
      labels:
        run: ingress-default-backend
    spec:
      containers:
      - name: backend-container-devops
        image: gcr.io/google_containers/defaultbackend:1.0
        ports:
        - containerPort: 8080
~
~
```



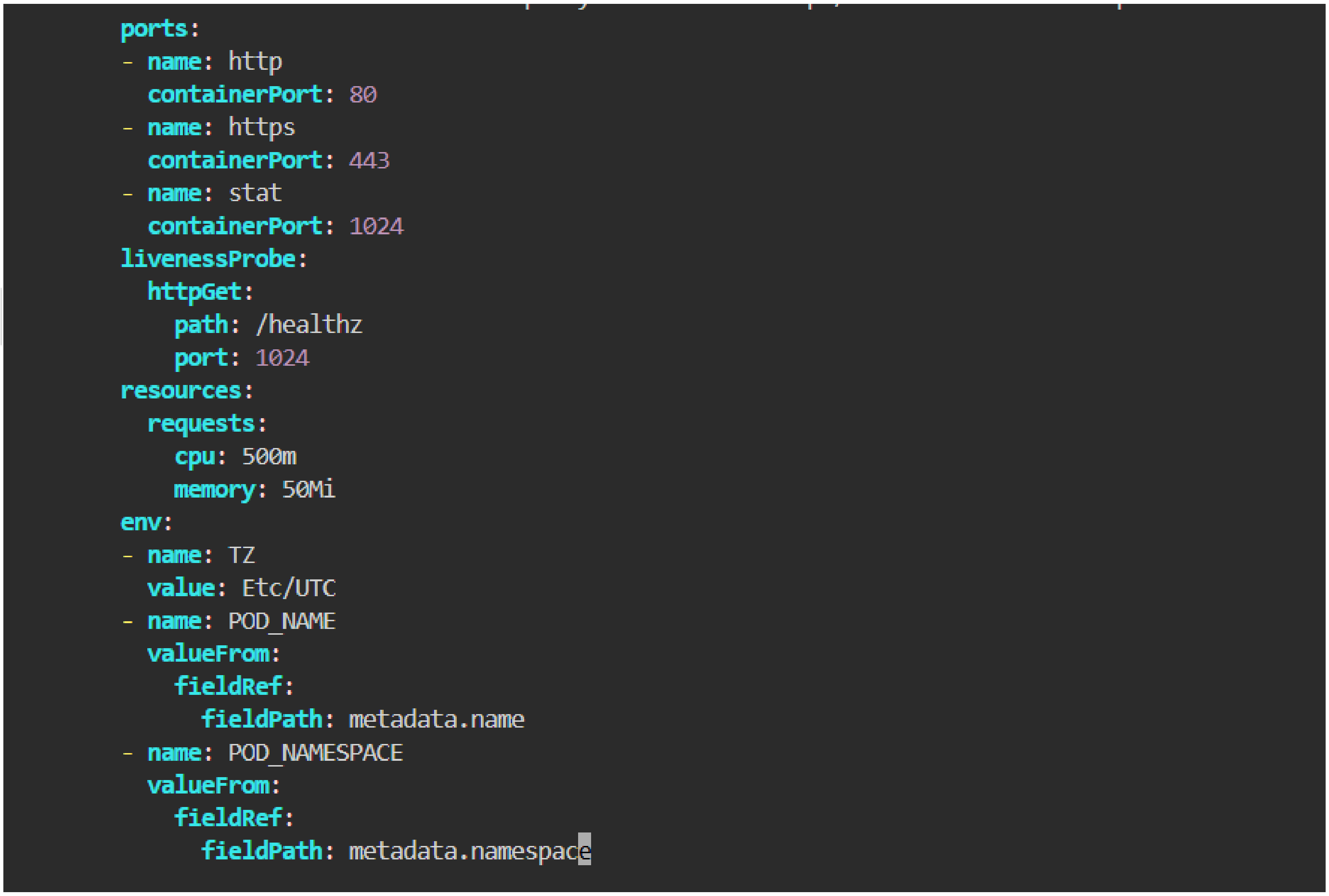
```yaml
apiVersion: v1
kind: Service
metadata:
  name: service-backend-devops
  namespace: haproxy-controller-devops
  labels:
    run: ingress-default-backend
spec:
  selector:
    run: ingress-default-backend
  ports:
  - name: port-backend
    protocol: TCP
    port: 8080
    targetPort: 8080
~
```



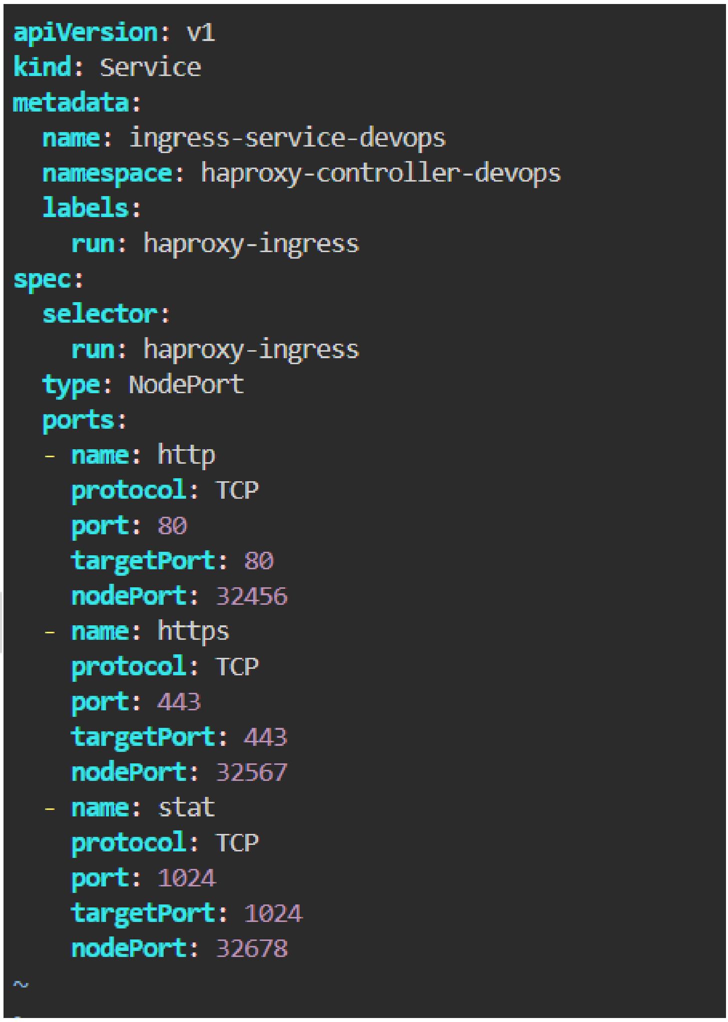
```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: haproxy-ingress-devops
  namespace: haproxy-controller-devops
  labels:
    run: haproxy-ingress
spec:
  replicas: 1
  selector:
    matchLabels:
      run: haproxy-ingress
  template:
    metadata:
      labels:
        run: haproxy-ingress
    spec:
      serviceAccountName: haproxy-service-account-devops
      containers:
      - name: ingress-container-devops
        image: haproxytech/kubernetes-ingress
        args:
        - --default-backend-service=haproxy-controller-devops/service-backend-devops
        ports:
        - name: http
          containerPort: 80
        - name: https
          containerPort: 443
        - name: stat
          containerPort: 1024
        livenessProbe:
          httpGet:
            path: /healthz
            port: 1024
```



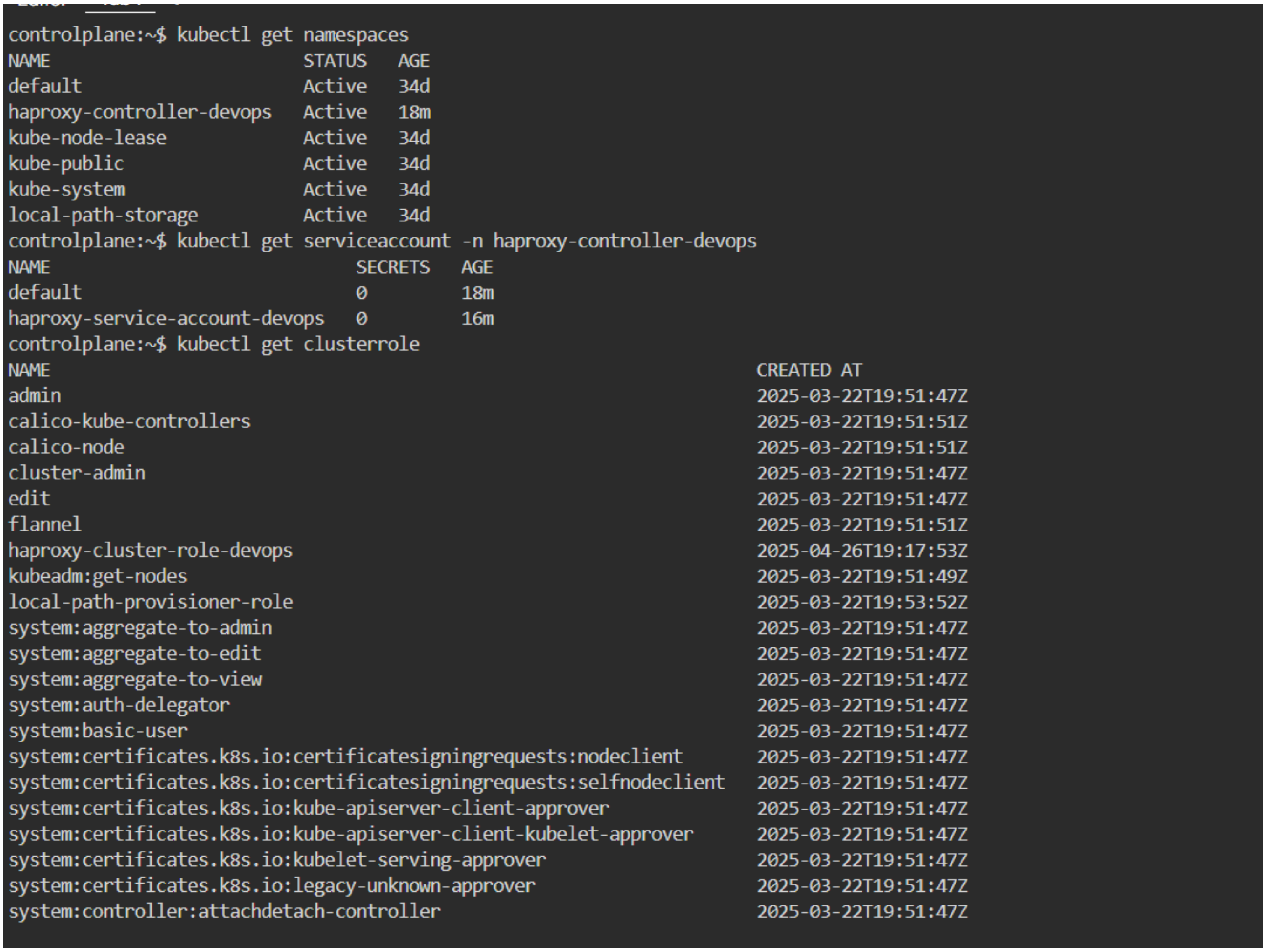
```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: haproxy-ingress-devops
  namespace: haproxy-controller-devops
  labels:
    run: haproxy-ingress
spec:
  replicas: 1
  selector:
    matchLabels:
      run: haproxy-ingress
  template:
    metadata:
      labels:
        run: haproxy-ingress
    spec:
      serviceAccountName: haproxy-service-account-devops
      containers:
      - name: ingress-container-devops
        image: haproxytech/kubernetes-ingress
        args:
        - --default-backend-service=haproxy-controller-devops/service-backend-devops
        ports:
        - name: http
          containerPort: 80
        - name: https
          containerPort: 443
        - name: stat
          containerPort: 1024
        livenessProbe:
          httpGet:
            path: /healthz
            port: 1024
```



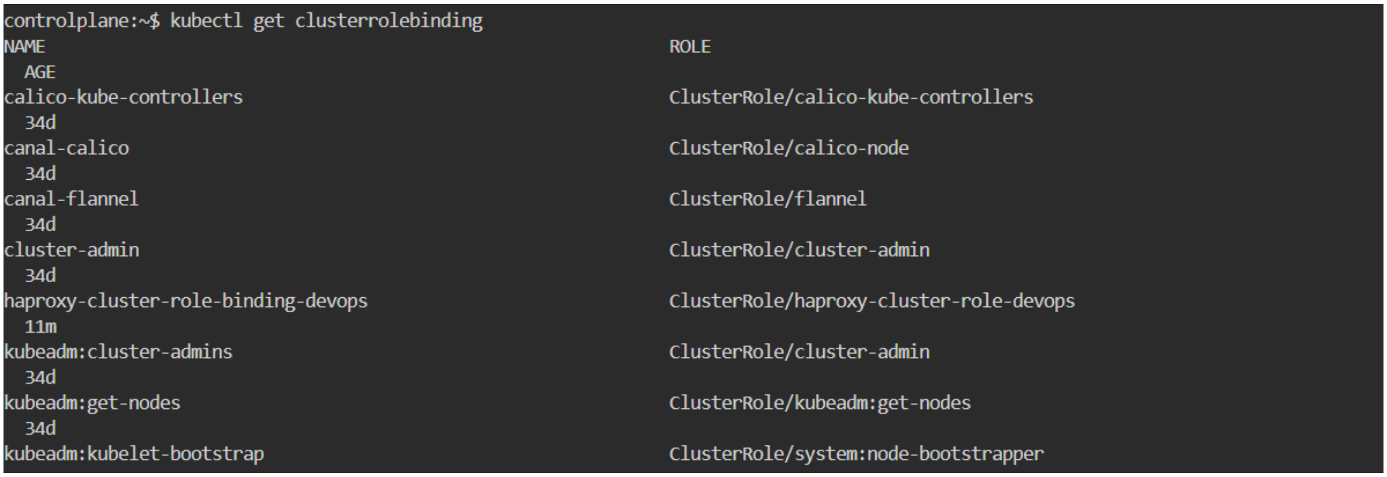
```yaml
        ports:
        - name: http
          containerPort: 80
        - name: https
          containerPort: 443
        - name: stat
          containerPort: 1024
        livenessProbe:
          httpGet:
            path: /healthz
            port: 1024
        resources:
          requests:
            cpu: 500m
            memory: 50Mi
        env:
        - name: TZ
          value: Etc/UTC
        - name: POD_NAME
          valueFrom:
            fieldRef:
              fieldPath: metadata.name
        - name: POD_NAMESPACE
          valueFrom:
            fieldRef:
              fieldPath: metadata.namespace
```



```
apiVersion: v1
kind: Service
metadata:
  name: ingress-service-devops
  namespace: haproxy-controller-devops
  labels:
    run: haproxy-ingress
spec:
  selector:
    run: haproxy-ingress
  type: NodePort
  ports:
  - name: http
    protocol: TCP
    port: 80
    targetPort: 80
    nodePort: 32456
  - name: https
    protocol: TCP
    port: 443
    targetPort: 443
    nodePort: 32567
  - name: stat
    protocol: TCP
    port: 1024
    targetPort: 1024
    nodePort: 32678
~
```



```
controlplane:~$ kubectl get namespaces
NAME                        STATUS   AGE
default                     Active   34d
haproxy-controller-devops   Active   18m
kube-node-lease             Active   34d
kube-public                 Active   34d
kube-system                 Active   34d
local-path-storage          Active   34d
controlplane:~$ kubectl get serviceaccount -n haproxy-controller-devops
NAME                             SECRETS   AGE
default                          0         18m
haproxy-service-account-devops   0         16m
controlplane:~$ kubectl get clusterrole
NAME                                                                   CREATED AT
admin                                                                  2025-03-22T19:51:47Z
calico-kube-controllers                                                2025-03-22T19:51:51Z
calico-node                                                            2025-03-22T19:51:51Z
cluster-admin                                                          2025-03-22T19:51:47Z
edit                                                                   2025-03-22T19:51:47Z
flannel                                                                2025-03-22T19:51:51Z
haproxy-cluster-role-devops                                            2025-04-26T19:17:53Z
kubeadm:get-nodes                                                      2025-03-22T19:51:49Z
local-path-provisioner-role                                            2025-03-22T19:53:52Z
system:aggregate-to-admin                                              2025-03-22T19:51:47Z
system:aggregate-to-edit                                               2025-03-22T19:51:47Z
system:aggregate-to-view                                               2025-03-22T19:51:47Z
system:auth-delegator                                                  2025-03-22T19:51:47Z
system:basic-user                                                      2025-03-22T19:51:47Z
system:certificates.k8s.io:certificatesigningrequests:nodeclient       2025-03-22T19:51:47Z
system:certificates.k8s.io:certificatesigningrequests:selfnodeclient   2025-03-22T19:51:47Z
system:certificates.k8s.io:kube-apiserver-client-approver              2025-03-22T19:51:47Z
system:certificates.k8s.io:kube-apiserver-client-kubelet-approver      2025-03-22T19:51:47Z
system:certificates.k8s.io:kubelet-serving-approver                    2025-03-22T19:51:47Z
system:certificates.k8s.io:legacy-unknown-approver                     2025-03-22T19:51:47Z
system:controller:attachdetach-controller                              2025-03-22T19:51:47Z
```



```
controlplane:~$ kubectl get clusterrolebinding
NAME                                                            ROLE
  AGE
calico-kube-controllers                                         ClusterRole/calico-kube-controllers
  34d
canal-calico                                                    ClusterRole/calico-node
  34d
canal-flannel                                                   ClusterRole/flannel
  34d
cluster-admin                                                   ClusterRole/cluster-admin
  34d
haproxy-cluster-role-binding-devops                             ClusterRole/haproxy-cluster-role-devops
  11m
kubeadm:cluster-admins                                          ClusterRole/cluster-admin
  34d
kubeadm:get-nodes                                               ClusterRole/kubeadm:get-nodes
  34d
kubeadm:kubelet-bootstrap                                       ClusterRole/system:node-bootstrapper
```