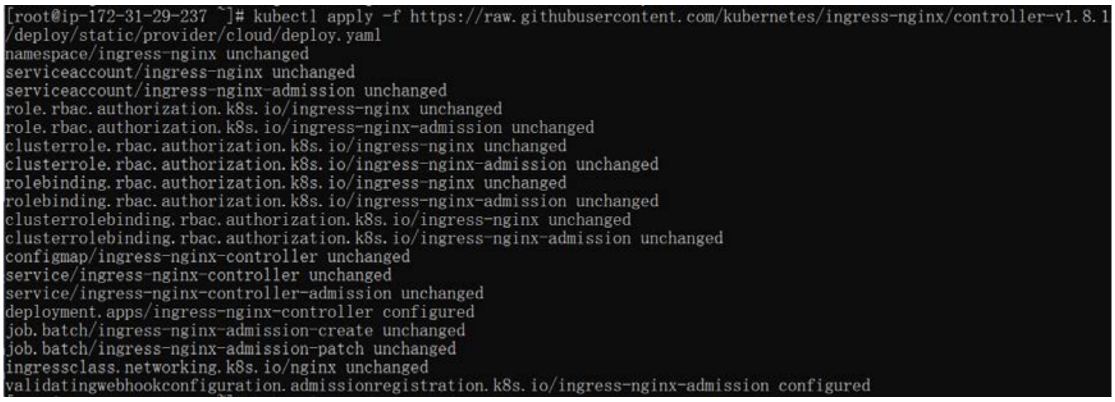
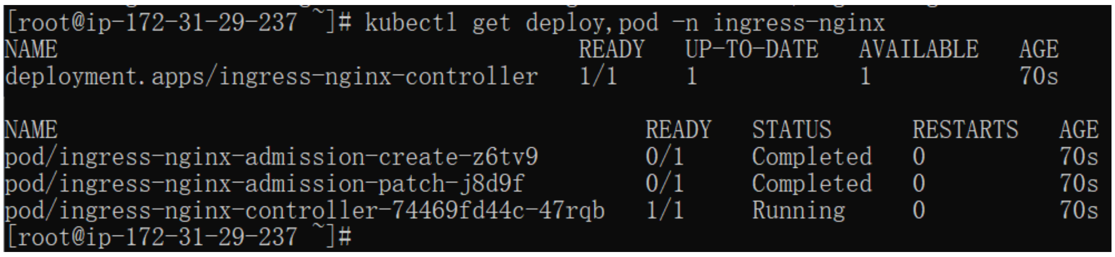
Install ingress.

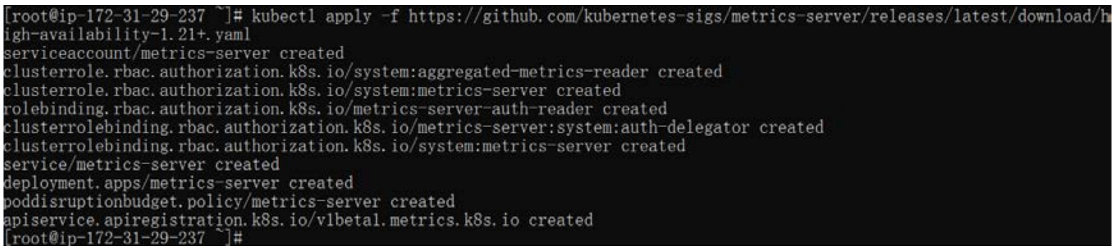

```
[root@ip-172-31-29-237 ~]# kubectl apply -f https://raw.githubusercontent.com/kubernetes/ingress-nginx/controller-v1.8.1
/deploy/static/provider/cloud/deploy.yaml
namespace/ingress-nginx unchanged
serviceaccount/ingress-nginx unchanged
serviceaccount/ingress-nginx-admission unchanged
role.rbac.authorization.k8s.io/ingress-nginx unchanged
role.rbac.authorization.k8s.io/ingress-nginx-admission unchanged
clusterrole.rbac.authorization.k8s.io/ingress-nginx unchanged
clusterrole.rbac.authorization.k8s.io/ingress-nginx-admission unchanged
rolebinding.rbac.authorization.k8s.io/ingress-nginx unchanged
rolebinding.rbac.authorization.k8s.io/ingress-nginx-admission unchanged
clusterrolebinding.rbac.authorization.k8s.io/ingress-nginx unchanged
clusterrolebinding.rbac.authorization.k8s.io/ingress-nginx-admission unchanged
configmap/ingress-nginx-controller unchanged
service/ingress-nginx-controller unchanged
service/ingress-nginx-controller-admission unchanged
deployment.apps/ingress-nginx-controller configured
job.batch/ingress-nginx-admission-create unchanged
job.batch/ingress-nginx-admission-patch unchanged
ingressclass.networking.k8s.io/nginx unchanged
validatingwebhookconfiguration.admissionregistration.k8s.io/ingress-nginx-admission configured
```

Check the installation status.

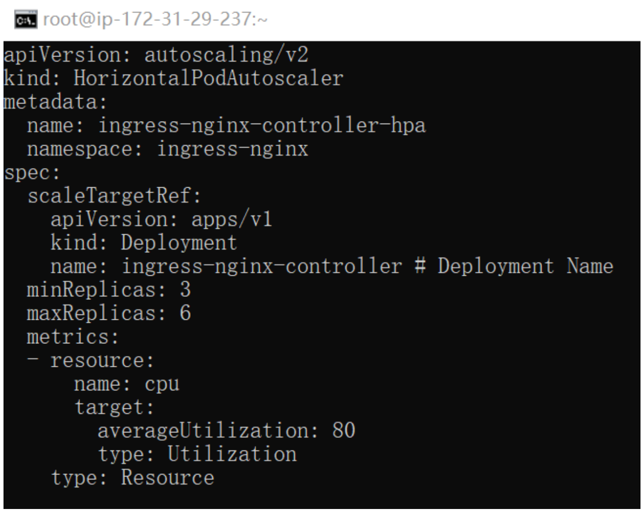

```
[root@ip-172-31-29-237 ~]# kubectl get deploy,pod -n ingress-nginx
NAME                                       READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/ingress-nginx-controller   1/1     1            1           70s

NAME                                           READY   STATUS      RESTARTS   AGE
pod/ingress-nginx-admission-create-z6tv9       0/1     Completed   0          70s
pod/ingress-nginx-admission-patch-j8d9f        0/1     Completed   0          70s
pod/ingress-nginx-controller-74469fd44c-47rqb  1/1     Running     0          70s
[root@ip-172-31-29-237 ~]#
```

The above status indicates that the installation is successful.

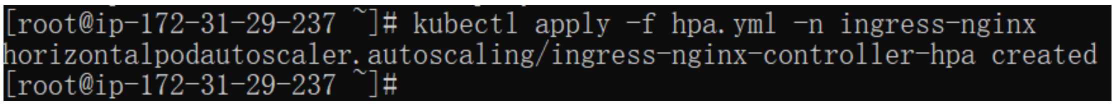
Under normal circumstances, Ingress has enough capacity to cope with business emergencies. In order to avoid insufficient demand under high load conditions, we can horizontally expand Ingress through HPA.


```
[root@ip-172-31-29-237 ~]# kubectl apply -f https://github.com/kubernetes-sigs/metrics-server/releases/latest/download/h
igh-availability-1.21+.yaml
serviceaccount/metrics-server created
clusterrole.rbac.authorization.k8s.io/system:aggregated-metrics-reader created
clusterrole.rbac.authorization.k8s.io/system:metrics-server created
rolebinding.rbac.authorization.k8s.io/metrics-server-auth-reader created
clusterrolebinding.rbac.authorization.k8s.io/metrics-server:system:auth-delegator created
clusterrolebinding.rbac.authorization.k8s.io/system:metrics-server created
service/metrics-server created
deployment.apps/metrics-server created
poddisruptionbudget.policy/metrics-server created
apiservice.apiregistration.k8s.io/v1beta1.metrics.k8s.io created
[root@ip-172-31-29-237 ~]#
```

Create HPA.



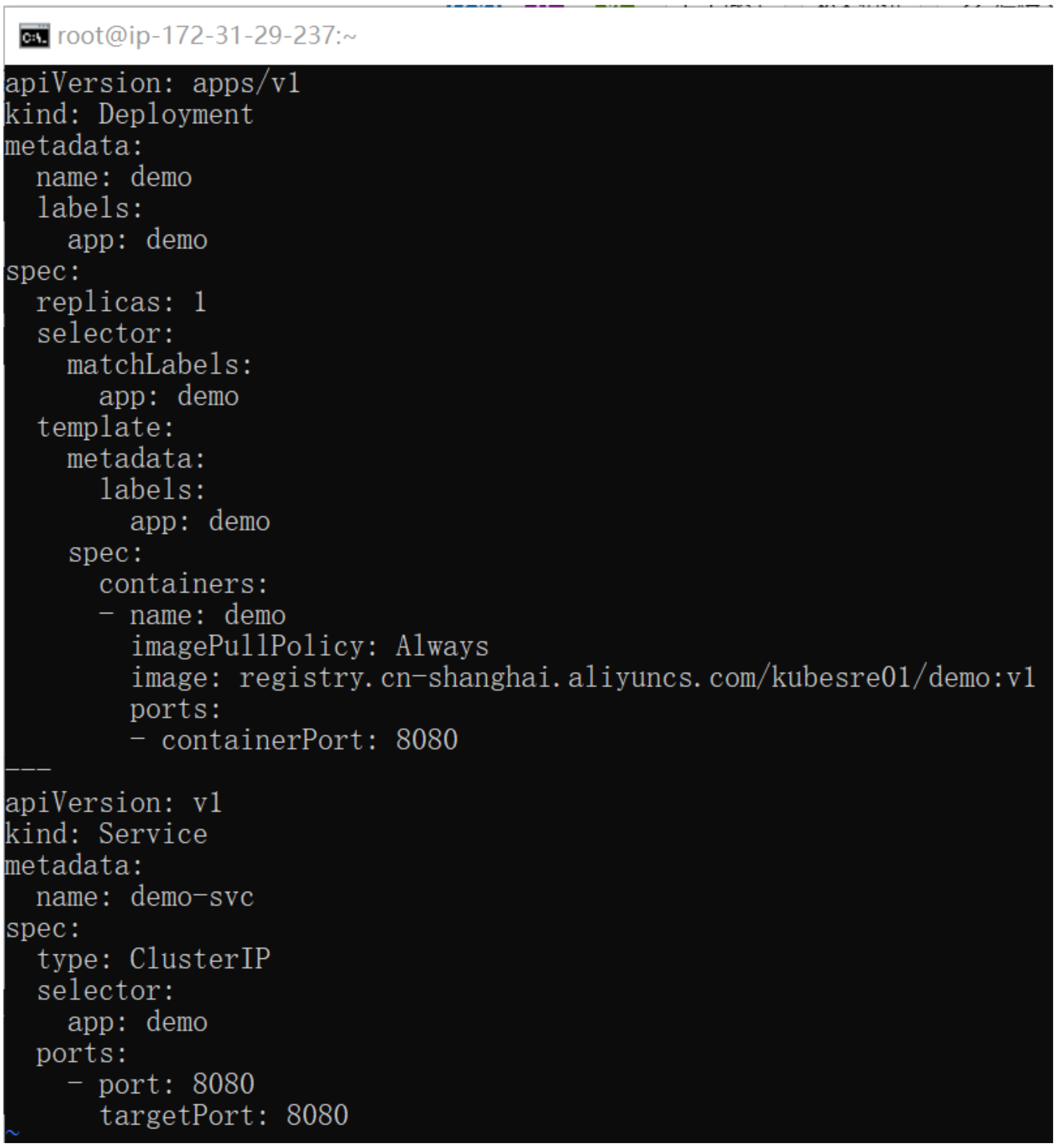
```yaml
apiVersion: autoscaling/v2
kind: HorizontalPodAutoscaler
metadata:
  name: ingress-nginx-controller-hpa
  namespace: ingress-nginx
spec:
  scaleTargetRef:
    apiVersion: apps/v1
    kind: Deployment
    name: ingress-nginx-controller # Deployment Name
  minReplicas: 3
  maxReplicas: 6
  metrics:
  - resource:
      name: cpu
      target:
        averageUtilization: 80
        type: Utilization
    type: Resource
```



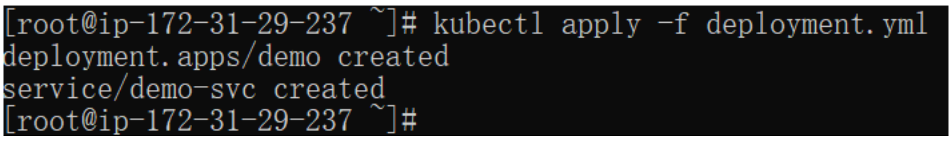
```
[root@ip-172-31-29-237 ~]# kubectl apply -f hpa.yml -n ingress-nginx
horizontalpodautoscaler.autoscaling/ingress-nginx-controller-hpa created
[root@ip-172-31-29-237 ~]#
```

Verify that Ingress is functioning normally.

Deploy the application.

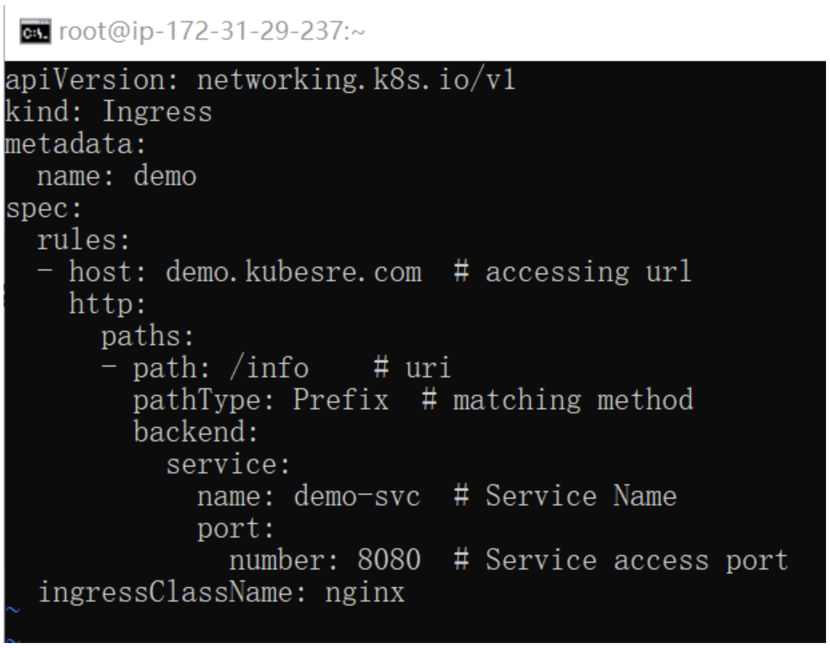


```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: demo
  labels:
    app: demo
spec:
  replicas: 1
  selector:
    matchLabels:
      app: demo
  template:
    metadata:
      labels:
        app: demo
    spec:
      containers:
      - name: demo
        imagePullPolicy: Always
        image: registry.cn-shanghai.aliyuncs.com/kubesre01/demo:v1
        ports:
        - containerPort: 8080
---
apiVersion: v1
kind: Service
metadata:
  name: demo-svc
spec:
  type: ClusterIP
  selector:
    app: demo
  ports:
    - port: 8080
      targetPort: 8080
~
```

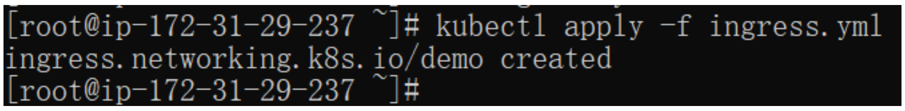


```
[root@ip-172-31-29-237 ~]# kubectl apply -f deployment.yml
deployment.apps/demo created
service/demo-svc created
[root@ip-172-31-29-237 ~]#
```

Create an ingress object.



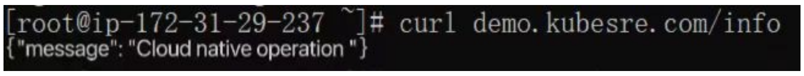
```
root@ip-172-31-29-237:~
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: demo
spec:
  rules:
  - host: demo.kubesre.com  # accessing url
    http:
      paths:
      - path: /info    # uri
        pathType: Prefix  # matching method
        backend:
          service:
            name: demo-svc  # Service Name
            port:
              number: 8080  # Service access port
  ingressClassName: nginx
~
```



```
[root@ip-172-31-29-237 ~]# kubectl apply -f ingress.yml
ingress.networking.k8s.io/demo created
[root@ip-172-31-29-237 ~]#
```

pathType parameter description:
ImplementationSpecific: For this path type, the matching method depends on the IngressClass. Implementations may treat this as a separate pathType or the same as the Prefix or Exact types.
Exact: Matches the URL path exactly and is case-sensitive.
Prefix: Match based on URL path prefix separated by /. Matching is case-sensitive and is done element-by-element in the path. A path element refers to a list of tags in a path separated by the / delimiter. A request matches path p if each p is an element prefix of request path p.

Access:



```
[root@ip-172-31-29-237 ~]# curl demo.kubesre.com/info
{"message": "Cloud native operation "}
```

The above information appears, indicating that the access is successful.