

Terminal



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

```
controlplane $
controlplane $ launch.sh
Waiting for Kubernetes to start...
Kubernetes started
controlplane $
controlplane $
controlplane $ cat deployment.yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: webapp1
spec:
  replicas: 1
  selector:
    matchLabels:
      app: webapp1
  template:
    metadata:
      labels:
        app: webapp1
    spec:
      containers:
      - name: webapp1
        image: katacoda/docker-http-server:latest
        ports:
        - containerPort: 80
---
apiVersion: apps/v1
kind: Deployment
metadata:
  name: webapp2
spec:
  replicas: 1
  selector:
    matchLabels:
      app: webapp2
  template:
    metadata:
      labels:
        app: webapp2
    spec:
```

Terminal



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

```
controlplane $
controlplane $ launch.sh
Waiting for Kubernetes to start...
Kubernetes started
controlplane $
controlplane $
controlplane $ cat deployment.yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: webapp1
spec:
  replicas: 1
  selector:
    matchLabels:
      app: webapp1
  template:
    metadata:
      labels:
        app: webapp1
    spec:
      containers:
      - name: webapp1
        image: katacoda/docker-http-server:latest
        ports:
        - containerPort: 80
---
apiVersion: apps/v1
kind: Deployment
metadata:
  name: webapp2
spec:
  replicas: 1
  selector:
    matchLabels:
      app: webapp2
  template:
    metadata:
      labels:
        app: webapp2
    spec:
```

Terminal



```
    app: webapp2
  spec:
    containers:
      - name: webapp2
        image: katacoda/docker-http-server:latest
        ports:
          - containerPort: 80
```

```
apiVersion: apps/v1
kind: Deployment
metadata:
```

```
  name: webapp3
```

```
spec:
```

```
  replicas: 1
```

```
  selector:
```

```
    matchLabels:
```

```
      app: webapp3
```

```
  template:
```

```
    metadata:
```

```
      labels:
```

```
        app: webapp3
```

```
    spec:
```

```
      containers:
```

```
        - name: webapp3
```

```
          image: katacoda/docker-http-server:latest
```

```
          ports:
```

```
            - containerPort: 80
```

```
apiVersion: v1
```

```
kind: Service
```

```
metadata:
```

```
  name: webapp1-svc
```

```
  labels:
```

```
    app: webapp1
```

```
spec:
```

```
  ports:
```

```
    - port: 80
```

```
  selector:
```

```
    app: webapp1
```

```
apiVersion: v1
```

```
kind: Service
```

```
metadata:
```

Terminal



```
apiVersion: v1
kind: Service
metadata:
  name: webapp1-svc
  labels:
    app: webapp1
spec:
  ports:
    - port: 80
  selector:
    app: webapp1
---
apiVersion: v1
kind: Service
metadata:
  name: webapp2-svc
  labels:
    app: webapp2
spec:
  ports:
    - port: 80
  selector:
    app: webapp2
---
apiVersion: v1
kind: Service
metadata:
  name: webapp3-svc
  labels:
    app: webapp3
spec:
  ports:
    - port: 80
  selector:
    app: webapp3
controlplane $ kubectl apply -f deployment.yaml
deployment.apps/webapp1 created
deployment.apps/webapp2 created
deployment.apps/webapp3 created
service/webapp1-svc created
service/webapp2-svc created
service/webapp3-svc created
controlplane $ kubectl get deployment
```

Terminal

+

NAME	READY	UP-TO-DATE	AVAILABLE	AGE
webapp1	0/1	1	0	3s
webapp2	0/1	1	0	3s
webapp3	0/1	1	0	3s

```
controlplane $ cat ingress.yaml
apiVersion: v1
kind: Namespace
metadata:
  name: nginx-ingress
---
apiVersion: v1
kind: Secret
metadata:
  name: default-server-secret
  namespace: nginx-ingress
type: kubernetes.io/tls
data:
  tls.crt: LS0tLS1CRUdJTiBDRVJUSUZJQ0FURSB0tLS0tCk1JSUN2akNDQWFzQ0NRREFPRj10THNhWfHEQU5CZ2txaGtpRz13MEJBUXNGOURBaE1SOHdIUUVlEVlFRRRERCWk8KUjBsTldFbHVVaM0
psYzNORGiYNTBjbTlZykdWeU1CNFhEVEU0TURreE1qRTRNRE16TlZvd0lURWZNQjBHQTFFVRUF3d1dUa2RKVGxoSmJtZHLawe56UTI5dWRISnZiR3hsY2pdQ0F
TSXdeUUVlKcKtvWklodmNOQVVFQkJRQURNZ0VQQUURDQ0FRb0NnZ0VCQUwvN2hiUETFWGRMdjNyaUM3QlBrMTNpWkt5eTlyQ08KR2xzUXYyK2EzUDF0azIrS3YwVGF5aGRcbDRrcnNUcTzzm8vWUk1
Y2Vhbkw4WGM3U1pyQkVRYm9EN2REBws1Qgo4eDZLS2xHWU5lWlg0Rm5Uz0VPaStlM2ptTFFxRlBSY1kzVnNPazFFeUZBL0JnWlJVbkNHZUtGeERSN0tQdGhyCmtqSXVuektURXUyaDU4Tlp0S2l3Sc
UJHdDEwctNTRYzhzT3ExM2FnbmovUWRjc0ZYYTJnMjB1K1lYzDdoZ3krZksKwk4vVUkxQUQ0YzZyM1lma1ZUWUvHd1lxQVp1WXN2V0RKbW1GNWRwdEMzN01lcDBPRUxVTEExSakZJOTZXXNXIwSaaS1Tm
dPc25NWFJNV1hYVlpiNWRxT3R0SmRtS3PhZ25T2lJQQVpQN2MwQjFQU2FqYzZjNGZRVXpNQ0F3RUFbVEFOckJna3Foa2lHOXcwQkFRc0ZBQU9DQVFFQWpLb2tRdGRPcEsrtZhibWVPc3lySmdJSXJ
ycVFVY2ZOUitjb0hzVUoKdGhrYnhITFMzR3VBTWI5dm15VExPY2xxeC9aYzJPblEwMEJCLz1Tb0swcitFz1U2U1VrRWtWcitTTFA3NTdUWgozZWl4dmdPdEduMS9ienM3bzNBaS9kclkrCUi5Q2k1
S3lPc3FHTG1US2xPaUtOYkcyR1ZyTWxjS0ZYQU80YTY3Cklnc1hzYktNbTQwV1U3cG9mcGltU1ZmaxFSDkV5YmN3N0NYODF6cFErUyt1eHRYK2VBZ3V0NHh3VlI5d2IyVXYKelhuZk9HbWhWNThDd
ldIQnKa0kxNXhaa2VUWXdsN0diaEFMSkZUUKk3dkhvQXprTWIzbjAxQjQyWjNrN3RXNQpJUDFmTlplOFUvOWxiUHN0T21FRFZkdjF5ZytVRVJxbStGSis2R0oxeFJGcGZnPT0KLS0tLS1FTkQgQ0
VSVELGSUNBEU0tLS0tLQo=
  tls.key: LS0tLS1CRUdJTiBSU0EgUUFJJVJkFURSBURVktLS0tLQpNSU1FcEFJQkFBS0NBUEVBdi91RWM4b1JkMHUvZXBVJTHNFk1RYZUprcKxMMnNjNGFWaEMvYjVYy9XMIrINHEvClJocktGME
dYaVn1eE9ycXgrajlNamx4NXFjdnhkenRKbXNFUk1Z1B0ME9hVGtIekhbv3FVWmcwZGxmZ1dkT0EKUTZMNTdlTl10Q29VOUz4amRXdzZUVVVRJUVUQ4R0JsRlNjSVo0b1hFTkhzbysyR3VTTWk2Zk1
wTVM3YUhudzFtMApxWkdvRWEzWFNyzEJ6eGc2clhkcUNlUDlCMXl3VmRyYURiUzc1aGQzdUdETDU4cGszOVFqVUFQaHpxdmRoK1JWCl2GNGJCaW9CbTVpeTlZTW1hWVhsMm0wTGZzeTzuUTRRdFFz
dEdNVWozcGJtdlFmazJBNNljeGRFeFpkZFZsdmwKMM82MjBjBm1lxcHFDZEtCRThCay90elFIVTlKcU56cHpoOUJUTXdJREFRQUJBb0lCQVFDZklHbXowOHhRVmorNwpLznZJUXQwQ0YzR2MxNld6e
DhVNml4MHg4Mm15dlkxUUNlL3BzWE9LZlRxt1h1SEnyUlp5TnUvZ2IvUuQ4bUFOCMxOMjRzTW10TWRJODg5TEZoTkp3QU5OODJDeTczckM5bzVvUdlkazAvYzRiBjAzSskVYNzZ5QjgzQm9rR1FvYk
sKMjhmNk0rdHUzUmFqNjd6Vmc2d2szaEhrU0pXSzBwV1YrSjdrUkRWYmhDYUzhNk5nMUZNRWxhTlozVDhhUutyQgpDUDNDeEFTdjYxWtk5TEI4KzNXWVFIK3NYaTVGM01pYVNBZ1BkQUk3WEh1dXF
ET1lvMU5PL0JoSGt1aVg2QnRtCnorNTZud2p2My8yUytSRmNBc3JMTnIwMDJZzi9oY0IraVldNzVWYmcydVd6WTY3TWdOTGQ5VW9RU3BDRkYrVm4KM0cyUnhybnhBb0dCQU40U3M0ZVlPU2huMVPQ
QjdhTUZsY0k2RHR2S2ErTGZTTFXfYy2pOZjJlSEpZNNhubmxKdgpGenpGL2RiVWVtbWxsSekR0Wkd1cXZXaHFI8y9iTjIyeWJhOU1WMD1RQ0JFTk5jNmtWajJTVHpuWkJVbEx4QzYrCk93Z0wyZHhKe
ndWelU0VC84ajdhAlRUN05BZVpFS2FvRHfYRG5BYWkyaW5oZU1JVVWZHRXFGKzJyQW9HQkFOMVAKK0tZL0lss3RWRzRKSklQNzBjUis3RmpyeXJpY05iWctQVzUvOXFHaWxnY2grZ3l4b25BWLbpd2
NpeDN3QVpGdwpazC96ZFB2aTbKWEppc1BSZjRMazg5b2pCumpiRmRmc2l5UmJYbyt3TFU4NUhRU2NGMnN5aUFPaTVBRHdVU0FkCm45YWFweUNweEFkREtERHdObit3ZFhtaT20OHRpSFRkK3RoVDh
kaVpBb0dCQU0t6Wis1bG9OOTBTYlF4VvH5YUwKMjF8Um9tMGJjcndsTmCaWNfSmlzaEhYa2xpSVVxZ3hSzk1NM2hhUVRUcklKZENFaHFsV01aV0xPb2I2NTNyZgo3aFlMSXM1ZUtka3o0aFRVdnp1
dm9TMHVXcm9CV2x0VHlGanIrsWhKZnZUc0hpOGdsU3FkbXgyS8kjhZUFVWUNXCndNdlQ4NmNLclNyNkQrZG8wS05FZzFzL0FvR0FlmkFvDhVfBnqLzBmRzgrv3hHc1RFV1Jqc1RNUzRSUjhRWXQKe
XdjdFA4aDzXGxKUTRCWGxQU05rMXZLTmtOUkxIb2p2T2pCQTViYjhixbXNVU1BlV09NNENoaFJ4QnlHbmr2eAphYkJDRkFwY0IvbEg4d1R0a1VZYLN5T294ZGt5OEpe0ek90ajJhS0FiZhd6N1ArWD
ZD0dhjZmxYVfo5MWpYL3RMCjF3TmRKS2tDZ1lCbyt0UzB5TzJ2SWFmK2UwSkN5TGhzVDQ5cTN3zis2QWVqWGx2WDJ1VnRYejN5QTznXo5aCsKcDNlK2JMRUxwb3BOWFhNdUFRR0xhUkcrYlNncjR
5dERYbE5Z8ndUeThXczNKY3dlSTdgZVp2b0ZpbmNvVlVIMwphdmx0TUUVCRGYxSjltSDB5cDBWUNaS2ROdHNvZEZtQktzVETQMjJhTmtsVVhCS3gyZzR6cFE9PQotLS0tLUVORCBSU0EgUUFJJVJkFU
RSBLRVktLS0tLQo=
```

Terminal



```
XdjdfA4aDZxTGxKUTrCWGxQU05rMXZLTmtOUkxIb2pZT2pCQTviYjhibXNVU1BlV09NNENoaFJ4QnlHbmR2eAphYkJDkFwY0IvbEg4d1R0a1VZyln5T294ZGt5OEpk90ajJhS0FiZHd6NlArWD
ZDODhjZmxYVFo5MWpYL3RMCjF3TmRKS2tDZ1lCbYt0UzB5TzJ2SWFmK2UwSkN5TGhzVDQ5cTN3Zis2QWVqWGx2WDJlVnRYejN5QTZnbXo5aCsKcDNlK2JMRUxwb3B0WFhNdUFRR0xhUkcrYlNNcjR
5dERYbE5ZSndUeThXczNKY3dlSTdqZVp2b0ZpbmNvVlVIMwphdmxoTUVCRGYxSjltSDB5cDBwWUNaS2ROdHNvZEZtQktzVEtQMjJhTmtsVvVhCS3gyZzR6cFE9PQotLS0tLUVORCBSU0EgUFJJVkJFU
RSBLRVktLS0tLQo=
---
apiVersion: v1
kind: ServiceAccount
metadata:
  name: nginx-ingress
  namespace: nginx-ingress
---
kind: ConfigMap
apiVersion: v1
metadata:
  name: nginx-config
  namespace: nginx-ingress
data:
---
# Described at: https://docs.nginx.com/nginx-ingress-controller/installation/installation-with-manifests/
# Source from: https://github.com/nginxinc/kubernetes-ingress/blob/master/deployments/common/ingress-class.yaml
apiVersion: networking.k8s.io/v1beta1
kind: IngressClass
metadata:
  name: nginx
  # annotations:
  #   ingressclass.kubernetes.io/is-default-class: "true"
spec:
  controller: nginx.org/ingress-controller
---
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-ingress
  namespace: nginx-ingress
spec:
  replicas: 1
  selector:
    matchLabels:
      app: nginx-ingress
  template:
    metadata:
      labels:
        app: nginx-ingress
```

Terminal



```
metadata:
  labels:
    app: nginx-ingress
spec:
  serviceAccountName: nginx-ingress
  containers:
  - image: nginx/nginx-ingress:edge
    imagePullPolicy: Always
    name: nginx-ingress
    ports:
    - name: http
      containerPort: 80
    - name: https
      containerPort: 443
    env:
    - name: POD_NAMESPACE
      valueFrom:
        fieldRef:
          fieldPath: metadata.namespace
    - name: POD_NAME
      valueFrom:
        fieldRef:
          fieldPath: metadata.name
    args:
    - -nginx-configmaps=$(POD_NAMESPACE)/nginx-config
    - -default-server-tls-secret=$(POD_NAMESPACE)/default-server-secret
```

```
apiVersion: v1
kind: Service
metadata:
  name: nginx-ingress
  namespace: nginx-ingress
spec:
  type: NodePort
  ports:
  - port: 80
    targetPort: 80
    protocol: TCP
    name: http
  - port: 443
    targetPort: 443
    protocol: TCP
    name: https
```

Terminal



```
http:
  paths:
  - path: /webapp1
    backend:
      serviceName: webapp1-svc
      servicePort: 80
  - path: /webapp2
    backend:
      serviceName: webapp2-svc
      servicePort: 80
  - backend:
      serviceName: webapp3-svc
      servicePort: 80

controlplane $ kubectl create -f ingress-rules.yaml
ingress.extensions/webapp-ingress created
controlplane $ kubectl get ing
NAME          CLASS    HOSTS          ADDRESS          PORTS    AGE
webapp-ingress  nginx   my.kubernetes.example  172.17.0.22      80       3s
controlplane $ curl -H "Host: my.kubernetes.example" 172.17.0.22/webapp1
<html>
<head><title>404 Not Found</title></head>
<body>
<center><h1>404 Not Found</h1></center>
<hr><center>nginx/1.19.8</center>
</body>
</html>
controlplane $ curl -H "Host: my.kubernetes.example" 172.17.0.22/webapp2
<html>
<head><title>404 Not Found</title></head>
<body>
<center><h1>404 Not Found</h1></center>
<hr><center>nginx/1.19.8</center>
</body>
</html>
controlplane $ curl -H "Host: my.kubernetes.example" 172.17.0.22
<html>
<head><title>404 Not Found</title></head>
<body>
<center><h1>404 Not Found</h1></center>
<hr><center>nginx/1.19.8</center>
</body>
</html>
controlplane $
```