Getting started

Instruction

Control plane vs worker nodes

Control Plane:

- The control plane is installed on your master node
- Can be both a control plane node and a worker node
- It houses the API server, scheduler, and controller manager settings

Worker Nodes:

- This is where the kubelet and kube-proxy are installed
- You can use the kubeadm join command to join workers to the master node to form the cluster

First Test

New Pod

shell-demo.yaml

```
apiVersion: v1
kind: Pod
metadata:
  name: shell-demo
spec:
  volumes:
  - name: shared-data
    emptyDir: {}
  containers:
  - name: nginx
    image: nginx
    volumeMounts:
    - name: shared-data
```

```
      mountPath: /usr/share/nginx/html
  hostNetwork: true
  dnsPolicy: Default
```

Create a Pod

```
kubectl apply -f https://k8s.io/examples/application/shell-demo.yaml
```

Verify that the container is running

```
kubectl get pod shell-demo

# Get a shell to the running container
kubectl exec -it shell-demo -- /bin/bash
```

New Deployment

nginx-deployment.yaml:

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
  labels:
    app: nginx
spec:
  replicas: 3
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx
        image: nginx:1.14.2
        ports:
        - containerPort: 80
```

Create a Deployment

```
kubectl apply -f https://k8s.io/examples/controllers/nginx-deployment.yaml

# Without a yaml file
kubectl apply -f - <<EOF
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
  labels:
    app: nginx
spec:
  replicas: 3
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx
        image: nginx:1.14.2
        ports:
        - containerPort: 80
EOF
```

Verify

```
kubectl get deployments
kubectl get pods --show-labels
```

Networking

Inbound Rules for K3s Nodes

Protocol	Port	Source	Destination	Description

TCP	2379-2380	Servers	Servers	Required only for HA with embedded etcd
TCP	6443	Agents	Servers	K3s supervisor and Kubernetes API Server
UDP	8472	All nodes	All nodes	Required only for Flannel VXLAN
TCP	10250	All nodes	All nodes	Kubelet metrics
UDP	51820	All nodes	All nodes	Required only for Flannel Wireguard with IPv4
UDP	51821	All nodes	All nodes	Required only for Flannel Wireguard with IPv6
TCP	5001	All nodes	All nodes	Required only for embedded distributed registry (Spegel)
TCP	6443	All nodes	All nodes	Required only for embedded distributed registry (Spegel)

Typically, all outbound traffic is allowed.

Network access to other pods

- Different Namespace: `http://<service-name>.<namespace>:<port>`
- Same Namespace: `http://<service-name>:<port>`

Network access within the same pod

- http://localhost:<port>
- ?? container ????? port

Revision #89
Created 2025-01-24 10:01:24 CST by A-Lang (Admin)
Updated 2026-02-06 16:14:26 CST by A-Lang (Admin)