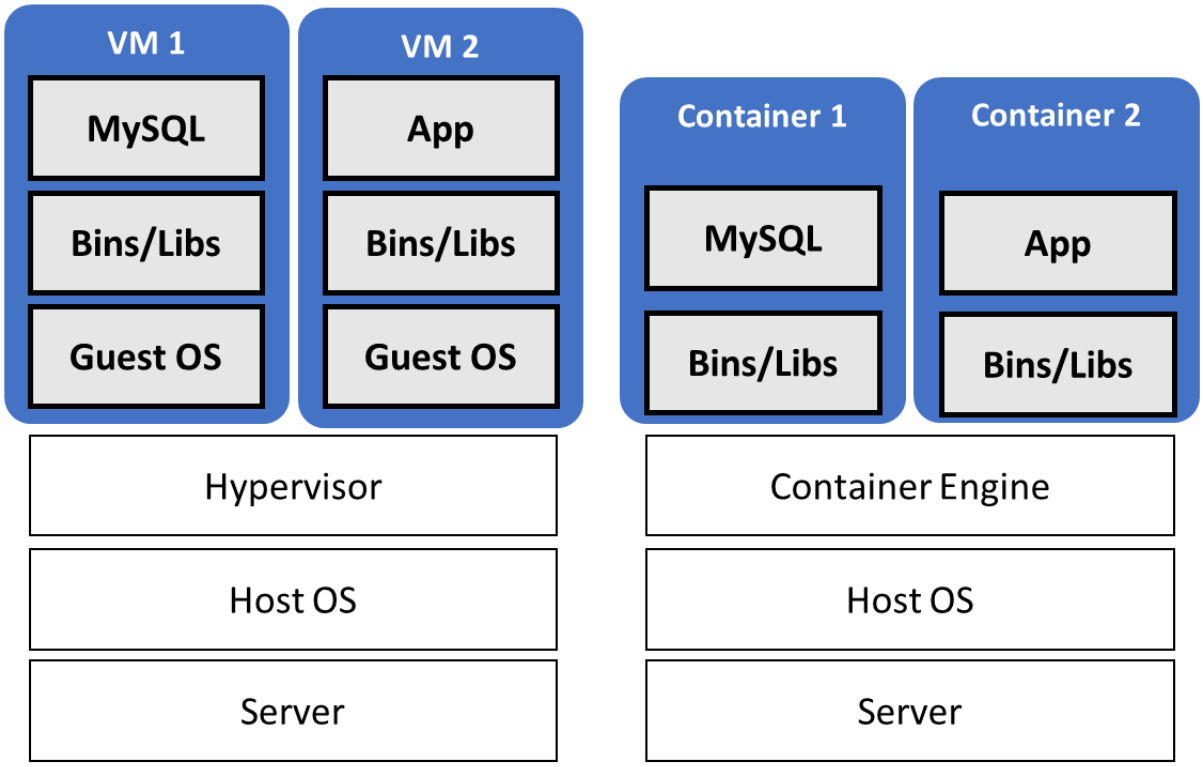
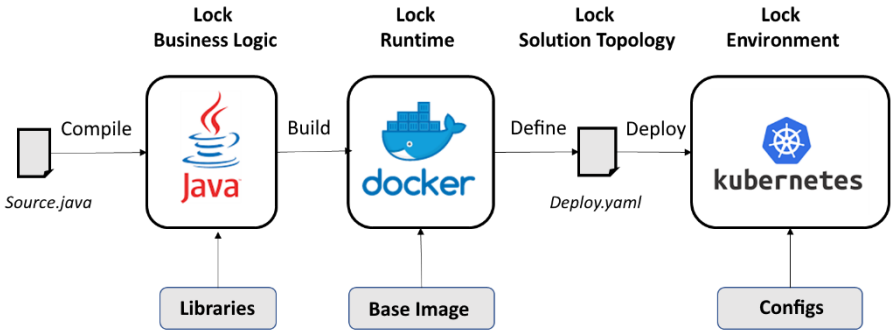
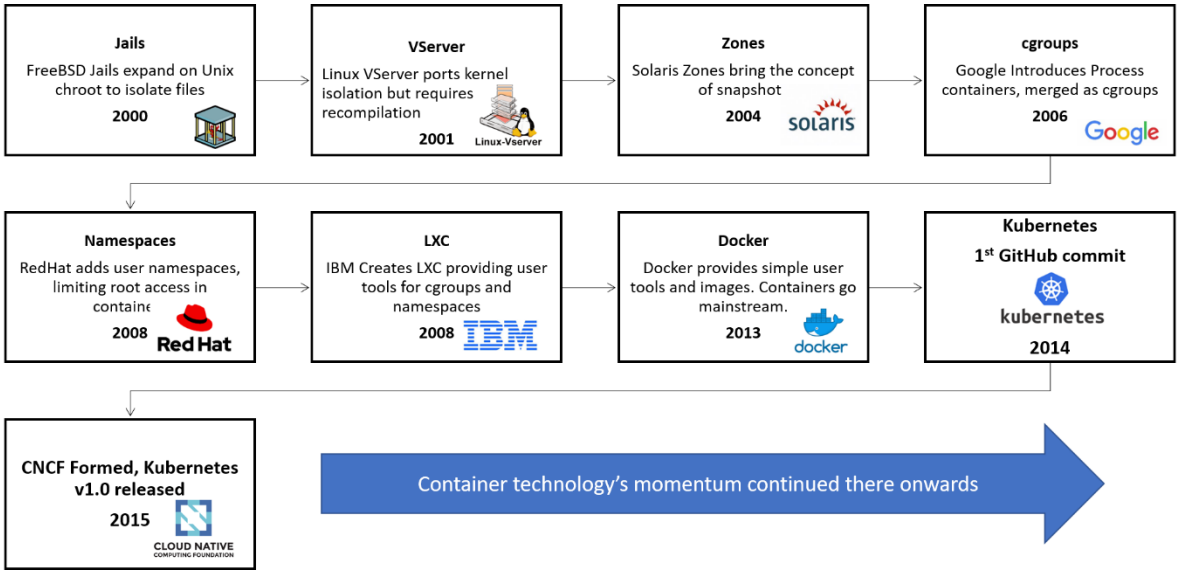
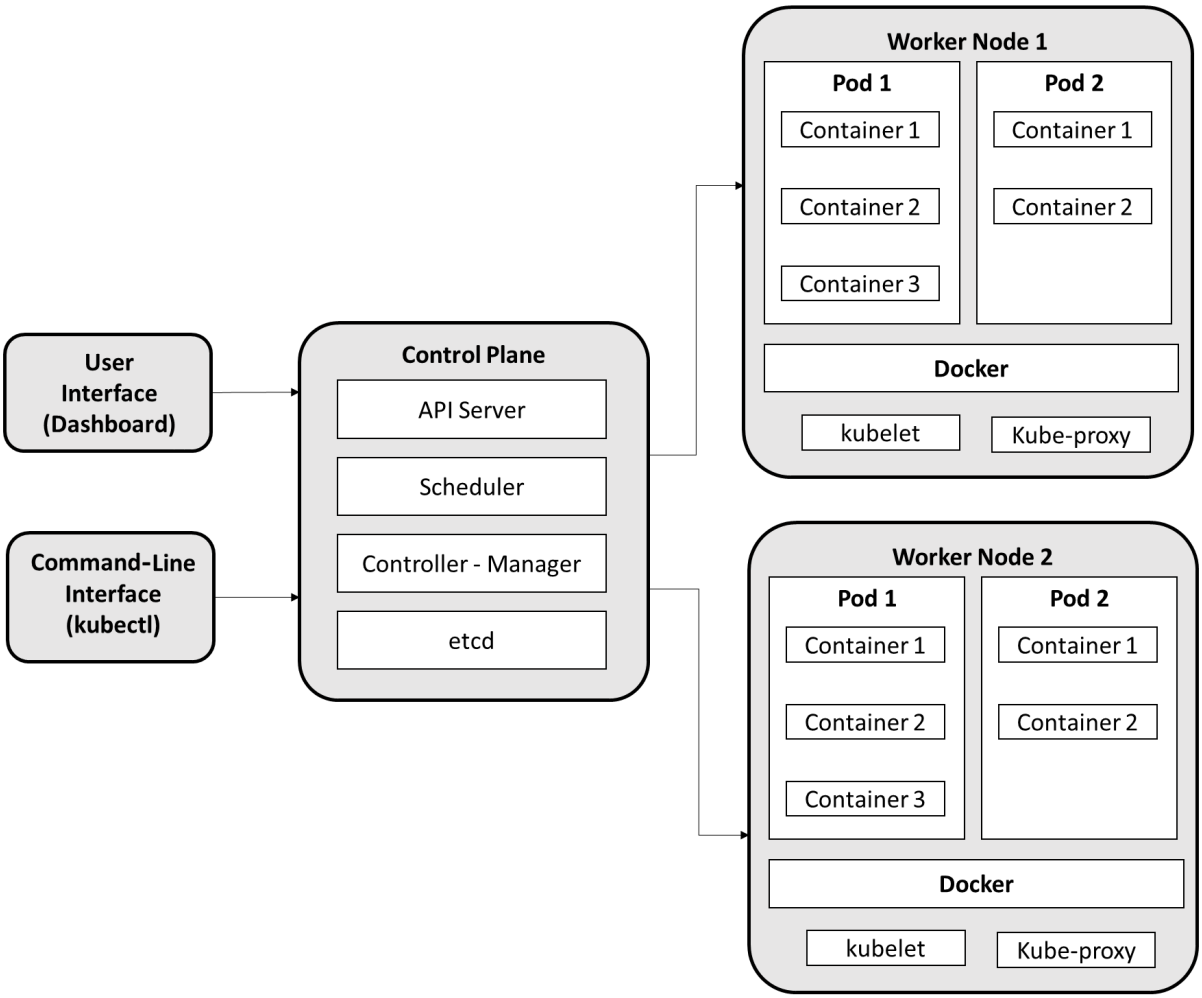
Chapter 1: Getting Started with Kubernetes

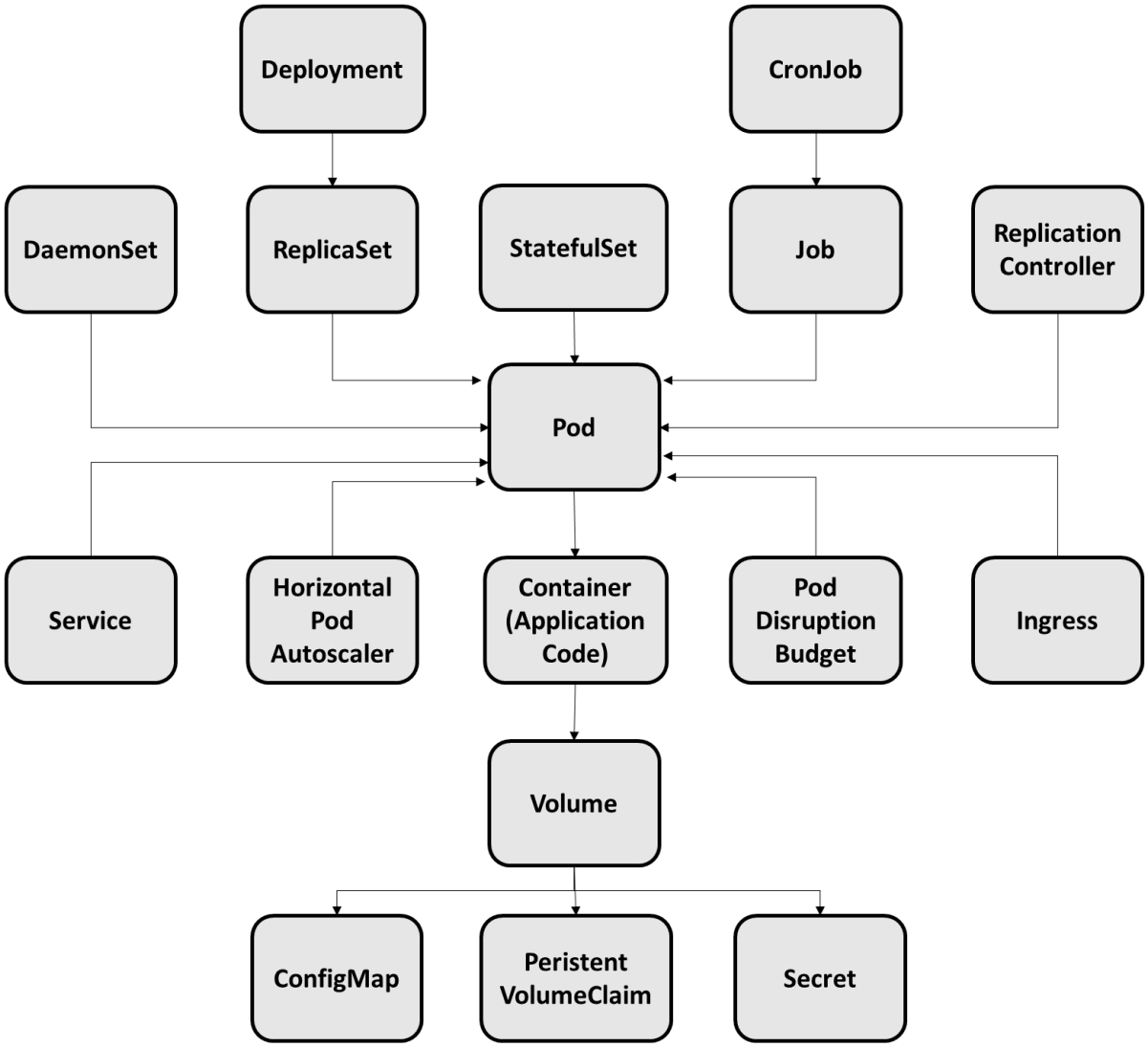


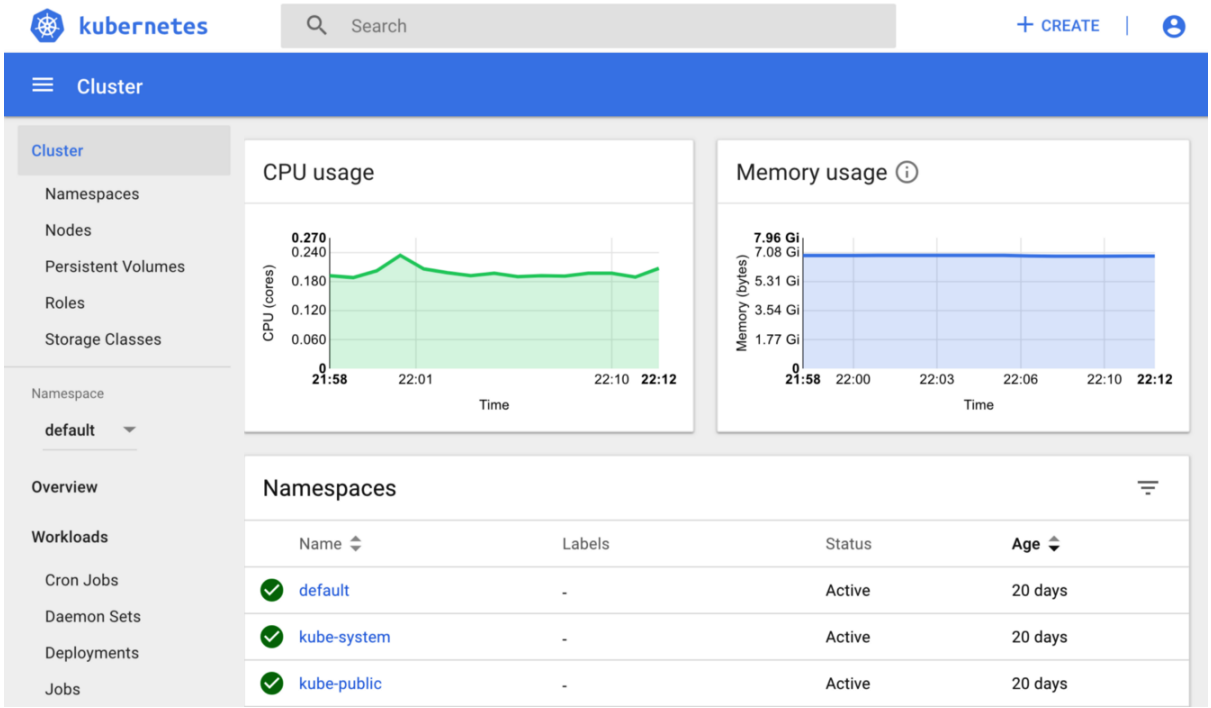
Virtual Machines **Containers**

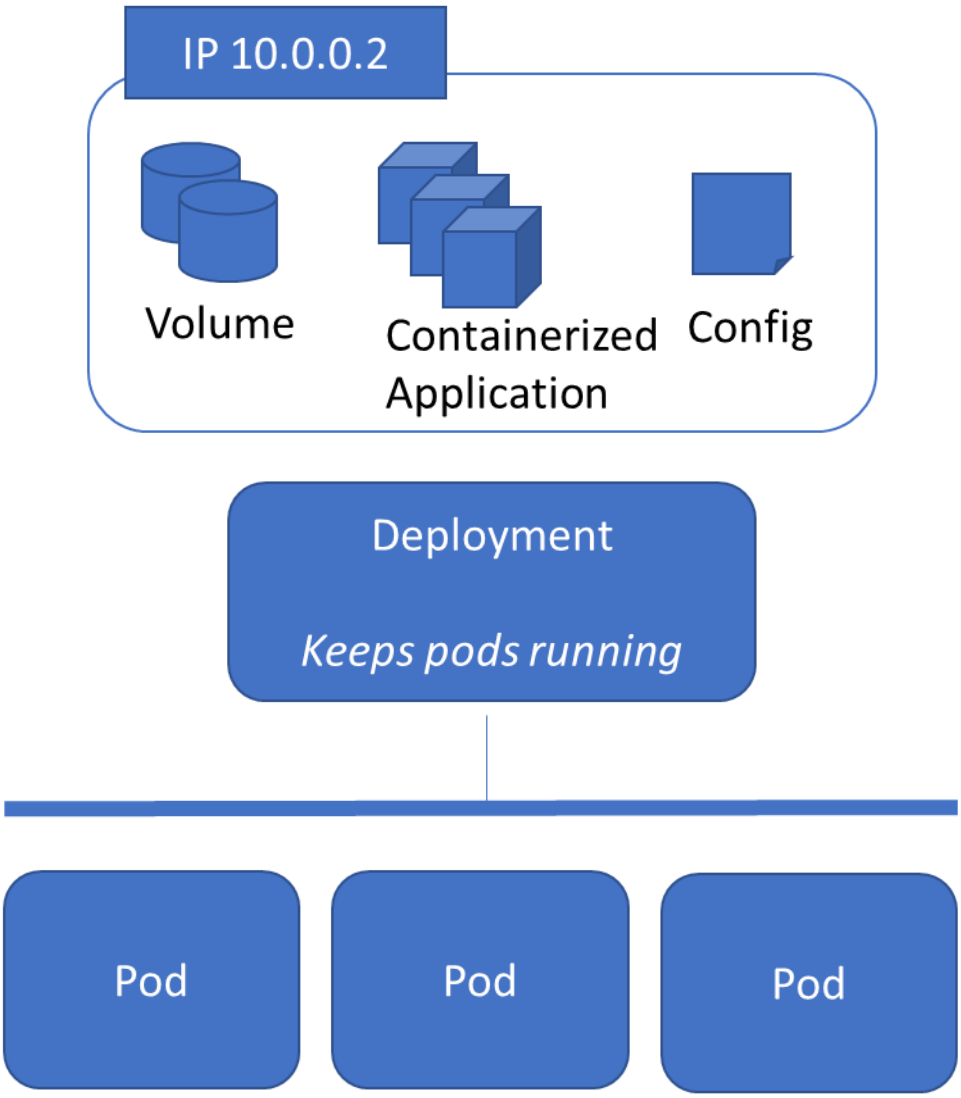



Jails
FreeBSD Jails expand on Unix chroot to isolate files
2000
VServer
Linux VServer ports kernel isolation but requires recompilation
2001
Linux-Vserver
Zones
Solaris Zones bring the concept of snapshot
2004
solaris
cgroups
Google Introduces Process containers, merged as cgroups
2006
Google
Namespaces
RedHat adds user namespaces, limiting root access in containe
2008
Red Hat
LXC
IBM Creates LXC providing user tools for cgroups and namespaces
2008
IBM
Docker
Docker provides simple user tools and images. Containers go mainstream.
2013
docker
Kubernetes
1st GitHub commit
kubernetes
2014
CNCF Formed, Kubernetes v1.0 released
2015
CLOUD NATIVE
COMPUTING FOUNDATION
Container technology's momentum continued there onwards


User Interface (Dashboard)
Command-Line Interface (kubectl)
Control Plane
API Server
Scheduler
Controller - Manager
etcd
Worker Node 1
Pod 1
Container 1
Container 2
Container 3
Pod 2
Container 1
Container 2
Docker
kubelet
Kube-proxy
Worker Node 2
Pod 1
Container 1
Container 2
Container 3
Pod 2
Container 1
Container 2
Docker
kubelet
Kube-proxy


Deployment
CronJob
DaemonSet
ReplicaSet
StatefulSet
Job
Replication Controller
Pod
Service
Horizontal Pod Autoscaler
Container (Application Code)
Pod Disruption Budget
Ingress
Volume
ConfigMap
Peristent VolumeClaim
Secret


kubernetes
Search
+ CREATE
Cluster
Cluster
Namespaces
Nodes
Persistent Volumes
Roles
Storage Classes
Namespace
default
Overview
Workloads
Cron Jobs
Daemon Sets
Deployments
Jobs
CPU usage
CPU (cores)
0.270
0.240
0.180
0.120
0.060
0
21:58
22:01
22:10
22:12
Time
Memory usage
Memory (bytes)
7.96 Gi
7.08 Gi
5.31 Gi
3.54 Gi
1.77 Gi
0
21:58
22:00
22:03
22:06
22:10
22:12
Time
Namespaces
Name
Labels
Status
Age
default
-
Active
20 days
kube-system
-
Active
20 days
kube-public
-
Active
20 days


IP 10.0.0.2
Volume
Containerized Application
Config
Deployment
Keeps pods running
Pod
Pod
Pod

Chapter 2: Introducing MicroK8s

```
azureuser@microk8s-vm:~$ sudo snap install microk8s --classic
microk8s (1.22/stable) v1.22.2 from Canonical✓ installed
azureuser@microk8s-vm:~$
```

```
azureuser@microk8s-vm:~$ sudo snap alias microk8s.kubectl kubectl
Added:
  - microk8s.kubectl as kubectl
azureuser@microk8s-vm:~$
```

```
azureuser@microk8s-vm:~$ kubectl get nodes
Insufficient permissions to access MicroK8s.
You can either try again with sudo or add the user azureuser to the 'microk8s' group:

    sudo usermod -a -G microk8s azureuser
    sudo chown -f -R azureuser ~/.kube

After this, reload the user groups either via a reboot or by running 'newgrp microk8s'.
azureuser@microk8s-vm:~$
```

```
azureuser@microk8s-vm:~$ sudo usermod -a -G microk8s azureuser
azureuser@microk8s-vm:~$     sudo chown -f -R azureuser ~/.kube
azureuser@microk8s-vm:~$
```

```
azureuser@microk8s-vm:~$ kubectl get nodes
NAME          STATUS   ROLES    AGE     VERSION
microk8s-vm   Ready    <none>   9m10s   v1.22.2-3+9ad9ee77396805
azureuser@microk8s-vm:~$
```

```
azureuser@microk8s-vm:~$ kubectl get nodes
NAME          STATUS   ROLES    AGE     VERSION
microk8s-vm   Ready    <none>   9m10s   v1.22.2-3+9ad9ee77396805
azureuser@microk8s-vm:~$ kubectl describe node microk8s-vm
Name:               microk8s-vm
Roles:              <none>
Labels:             beta.kubernetes.io/arch=amd64
                    beta.kubernetes.io/os=linux
                    kubernetes.io/arch=amd64
                    kubernetes.io/hostname=microk8s-vm
                    kubernetes.io/os=linux
                    microk8s.io/cluster=true
Annotations:        node.alpha.kubernetes.io/ttl: 0
                    projectcalico.org/IPv4Address: 10.1.0.4/24
                    projectcalico.org/IPv4VXLANTunnelAddr: 10.1.254.64
                    volumes.kubernetes.io/controller-managed-attach-detach: true
CreationTimestamp:  Thu, 14 Oct 2021 08:40:48 +0000
Taints:             <none>
Unschedulable:      false
Lease:
  HolderIdentity:  microk8s-vm
```

```
azureuser@microk8s-vm:~$ kubectl create deployment nginx --image=nginx
deployment.apps/nginx created
azureuser@microk8s-vm:~$
```

```
azureuser@microk8s-vm:~$ kubectl get pods
NAME                    READY   STATUS    RESTARTS   AGE
nginx-6799fc88d8-s2z8n  1/1     Running   0          68s
azureuser@microk8s-vm:~$
```

```
azureuser@microk8s-vm:~$ kubectl expose deployment nginx \
> --port 80 \
> --target-port 80 \
> --type ClusterIP \
> --selector=run=nginx \
> --name nginx
service/nginx exposed
azureuser@microk8s-vm:~$
```

```
azureuser@microk8s-vm:~$ kubectl get svc
NAME         TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)   AGE
kubernetes   ClusterIP   10.152.183.1     <none>        443/TCP   87m
nginx        ClusterIP   10.152.183.108   <none>        80/TCP    2m32s
azureuser@microk8s-vm:~$
```

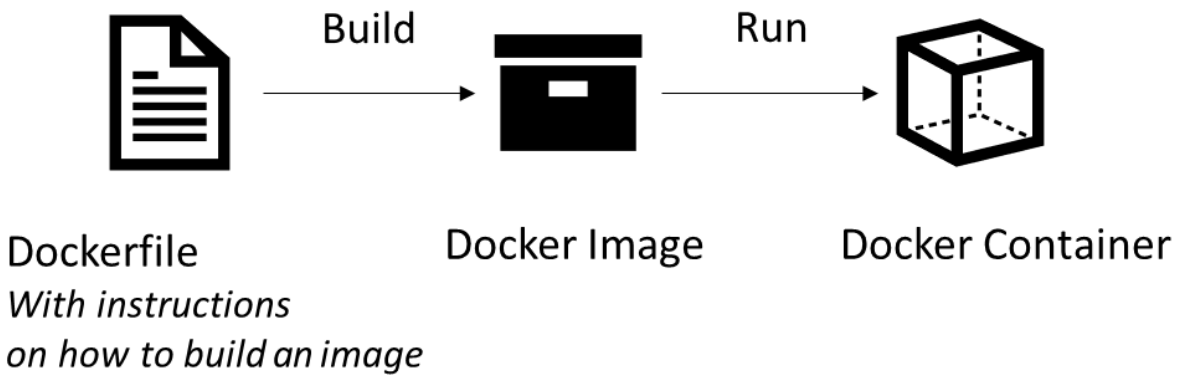
```
azureuser@microk8s-vm:~$ microk8s enable dns storage
Enabling DNS
Applying manifest
serviceaccount/coredns created
configmap/coredns created
Warning: spec.template.metadata.annotations[scheduler.alpha.kubernetes.
ctional in v1.16+; use the "priorityClassName" field instead
deployment.apps/coredns created
service/kube-dns created
clusterrole.rbac.authorization.k8s.io/coredns created
clusterrolebinding.rbac.authorization.k8s.io/coredns created
Restarting kubelet
DNS is enabled
Enabling default storage class
deployment.apps/hostpath-provisioner created
storageclass.storage.k8s.io/microk8s-hostpath created
serviceaccount/microk8s-hostpath created
clusterrole.rbac.authorization.k8s.io/microk8s-hostpath created
clusterrolebinding.rbac.authorization.k8s.io/microk8s-hostpath created
Storage will be available soon
azureuser@microk8s-vm:~$
```

```
azureuser@microk8s-vm:~$ kubectl get all --all-namespaces
NAMESPACE     NAME                                       READY   STATUS    RESTARTS   AGE
default       pod/nginx-6799fc88d8-s2z8n                 1/1     Running   0          46m
kube-system   pod/calico-kube-controllers-6f896476f5-qnjsh   1/1     Running   0          129m
kube-system   pod/coredns-7f9c69c78c-87btk               1/1     Running   0          33m
kube-system   pod/calico-node-msctq                      1/1     Running   0          129m
kube-system   pod/hostpath-provisioner-5c65fbdb4f-wznsp  1/1     Running   0          32m
```

```
azureuser@microk8s-vm:~$ microk8s status
microk8s is running
high-availability: no
  datastore master nodes: 127.0.0.1:19001
  datastore standby nodes: none
addons:
  enabled:
    dns                  # CoreDNS
    ha-cluster           # Configure high availability on the current node
    registry             # Private image registry exposed on localhost:32000
    storage              # Storage class; allocates storage from host directory
  disabled:
    ambassador           # Ambassador API Gateway and Ingress
    cilium               # SDN, fast with full network policy
    dashboard            # The Kubernetes dashboard
    fluentd              # Elasticsearch-Fluentd-Kibana logging and monitoring
    gpu                  # Automatic enablement of Nvidia CUDA
    helm                 # Helm 2 - the package manager for Kubernetes
    helm3                # Helm 3 - Kubernetes package manager
    host-access          # Allow Pods connecting to Host services smoothly
    ingress              # Ingress controller for external access
    istio                # Core Istio service mesh services
```

```
azureuser@microk8s-vm:~$ microk8s stop
Stopped.
azureuser@microk8s-vm:~$
```

```
azureuser@microk8s-vm:~$ microk8s start
Started.
```




```
azureuser@microk8s-vm:~$ docker images
REPOSITORY     TAG       IMAGE ID       CREATED        SIZE
nginx1.21      local     87a94228f133   13 days ago    133MB
nginx          latest    87a94228f133   13 days ago    133MB
hello-world    latest    feb5d9fea6a5   4 weeks ago    13.3kB
azureuser@microk8s-vm:~$
```

```
azureuser@microk8s-vm:~$ docker save nginx1.21 > nginxlocal.tar
azureuser@microk8s-vm:~$ microk8s ctr image import nginxlocal.tar
unpacking docker.io/library/nginx1.21:local (sha256:f090a2b152845c78ed6b9eac73ffbc267abc0b06a8717ca798e89d9589e70511)...do
ne
azureuser@microk8s-vm:~$
```

```
azureuser@microk8s-vm:~$ microk8s ctr image ls | grep nginx1.21
docker.io/library/nginx1.21:local
ation/vnd.docker.distribution.manifest.v2+json      sha256:f090a2b152845c78ed6b9eac73ffbc267abc0b06a871
1 51.3 MiB  linux/amd64
-containerd.image=managed
azureuser@microk8s-vm:~$
```

```
  GNU nano 4.8
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
  labels:
    app: nginx
spec:
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx
        image: nginx1.21:local
        imagePullPolicy: Never
        ports:
        - containerPort: 80
```

```
azureuser@microk8s-vm:~$ microk8s kubectl apply -f nginx.local
deployment.apps/nginx-deployment created
azureuser@microk8s-vm:~$
```

```
azureuser@microk8s-vm:~$ microk8s kubectl get deployment
NAME                READY   UP-TO-DATE   AVAILABLE   AGE
nginx               1/1     1            1           10d
nginx-deployment    1/1     1            1           6m10s
azureuser@microk8s-vm:~$
```

```
azureuser@microk8s-vm:~$ microk8s enable registry:size=40Gi
Addon storage is already enabled.
Enabling the private registry
Applying registry manifest
namespace/container-registry created
persistentvolumeclaim/registry-claim created
deployment.apps/registry created
service/registry created
configmap/local-registry-hosting configured
The registry is enabled
The size of the persistent volume is 40Gi
azureuser@microk8s-vm:~$
```

```
azureuser@microk8s-vm:~$ docker tag 87a94228f133 localhost:32000/nginx1.21:registry
```

```
azureuser@microk8s-vm:~$ docker push localhost:32000/nginx1.21:registry
The push refers to repository [localhost:32000/nginx1.21]
9959a332cf6e: Pushed
f7e00b807643: Pushed
f8e880dfc4ef: Pushed
788e89a4d186: Pushed
43f4e41372e4: Pushed
e81bff2725db: Pushed
registry: digest: sha256:7250923ba3543110040462388756ef099331822c6172a050b12c7a38361ea46f size: 1570
azureuser@microk8s-vm:~$
```

```
azureuser@microk8s-vm:~$ docker images
REPOSITORY                  TAG        IMAGE ID       CREATED       SIZE
nginx1.21                   local      87a94228f133   13 days ago   133MB
nginx                       latest     87a94228f133   13 days ago   133MB
localhost:32000/nginx1.21   registry   87a94228f133   13 days ago   133MB
hello-world                 latest     feb5d9fea6a5   4 weeks ago   13.3kB
azureuser@microk8s-vm:~$
```

```
  GNU nano 4.8
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-builtin-registry-deployment
  labels:
    app: nginx
spec:
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx
        image: localhost:32000/nginx1.21:registry
        ports:
        - containerPort: 80
```

```
azureuser@microk8s-vm:~$ microk8s kubectl apply -f nginx.builtin
deployment.apps/nginx-builtin-registry-deployment created
azureuser@microk8s-vm:~$
```

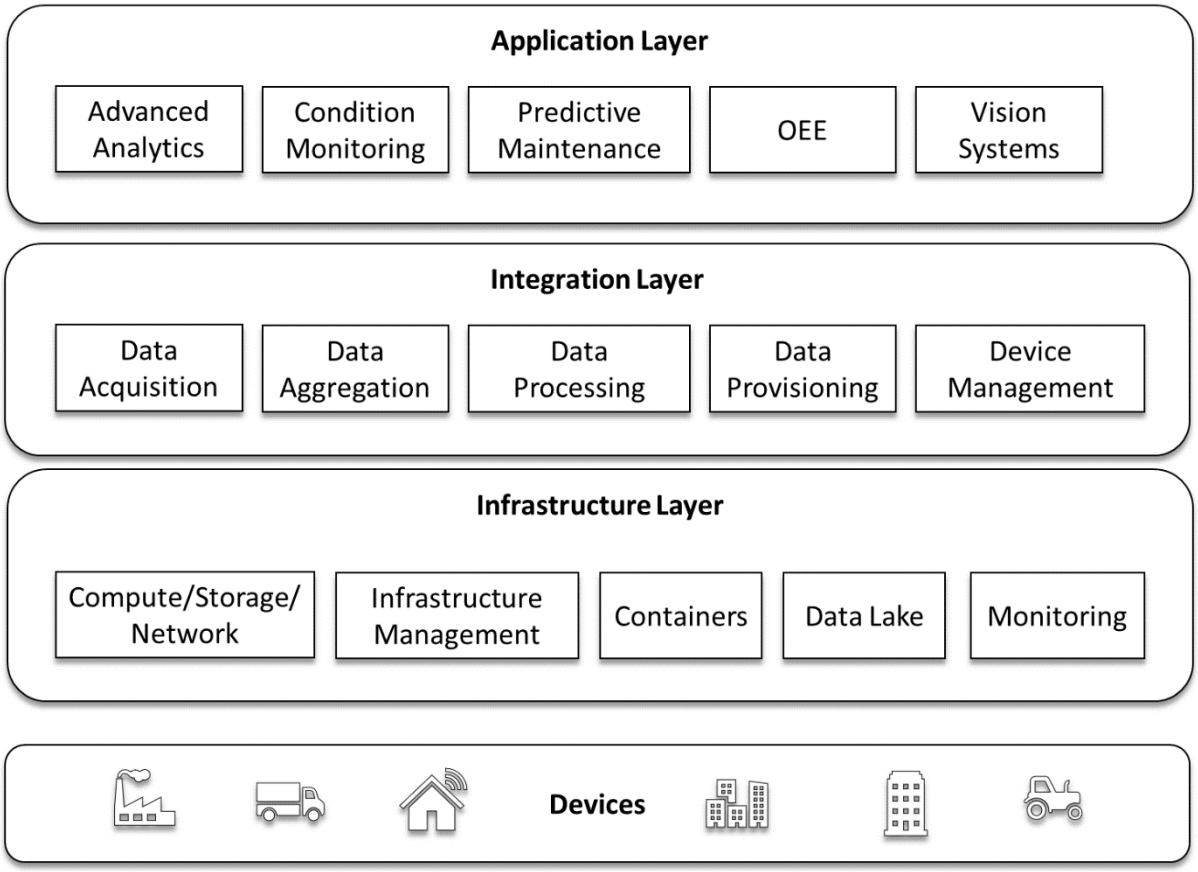
```
azureuser@microk8s-vm:~$ microk8s kubectl get deployment
NAME                                READY   UP-TO-DATE   AVAILABLE   AGE
nginx                               1/1     1            1           11d
nginx-deployment                    1/1     1            1           167m
nginx-builtin-registry-deployment   1/1     1            1           68s
azureuser@microk8s-vm:~$
```

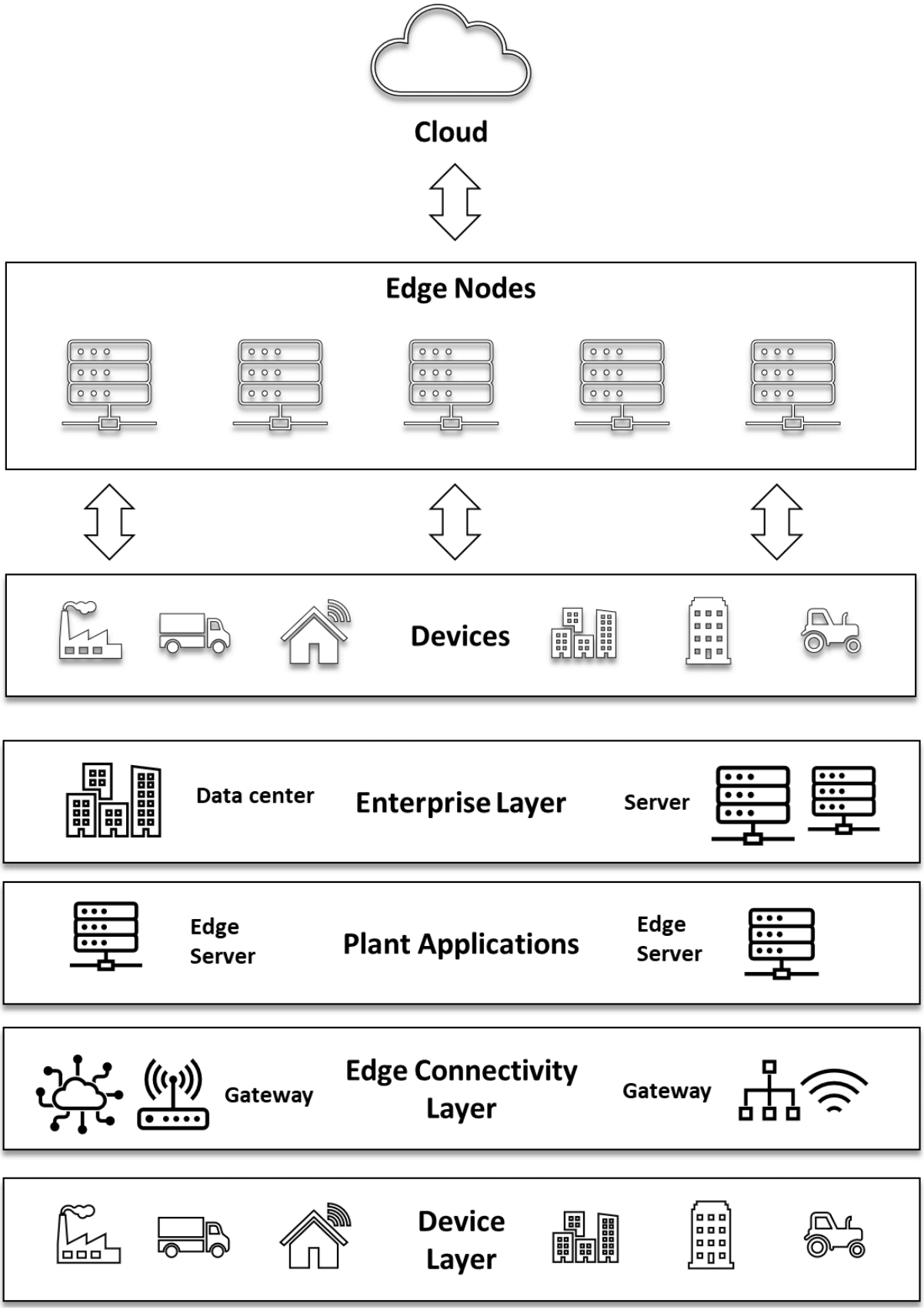
```
  GNU nano 4.8
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-private-or-public-registry-deployment
  labels:
    app: nginx
spec:
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx
        image: 10.131.231.155:32000/nginx1.21:registry
        ports:
        - containerPort: 80
```

```
azureuser@microk8s-vm:~$ kubectl describe pods nginx-deployment-5476f8c6df-ttkzz
Name:         nginx-deployment-5476f8c6df-ttkzz
Namespace:    default
Priority:     0
Node:         microk8s-vm/10.1.0.4
Start Time:   Mon, 25 Oct 2021 08:00:08 +0000
Labels:       app=nginx
              pod-template-hash=5476f8c6df
Annotations:  cni.projectcalico.org/podIP: 10.1.254.89/32
              cni.projectcalico.org/podIPs: 10.1.254.89/32
Status:       Running
IP:           10.1.254.89
IPs:
  IP:           10.1.254.89
Controlled By:  ReplicaSet/nginx-deployment-5476f8c6df
Containers:
  nginx:
    Container ID:   containerd://597bc9b01fdffd9b6dce344f62f9c746b3c71ac53929f2cfe267181af
    Image:          nginx1.21:local
    Image ID:       docker.io/library/nginx@sha256:644a70516a26004c97d0d85c7fe1d0c3a67ea8a
    Port:           80/TCP
    Host Port:      0/TCP
    State:          Running
      Started:      Thu, 28 Oct 2021 08:09:24 +0000
    Last State:     Terminated
      Reason:       Unknown
      Exit Code:    255
```

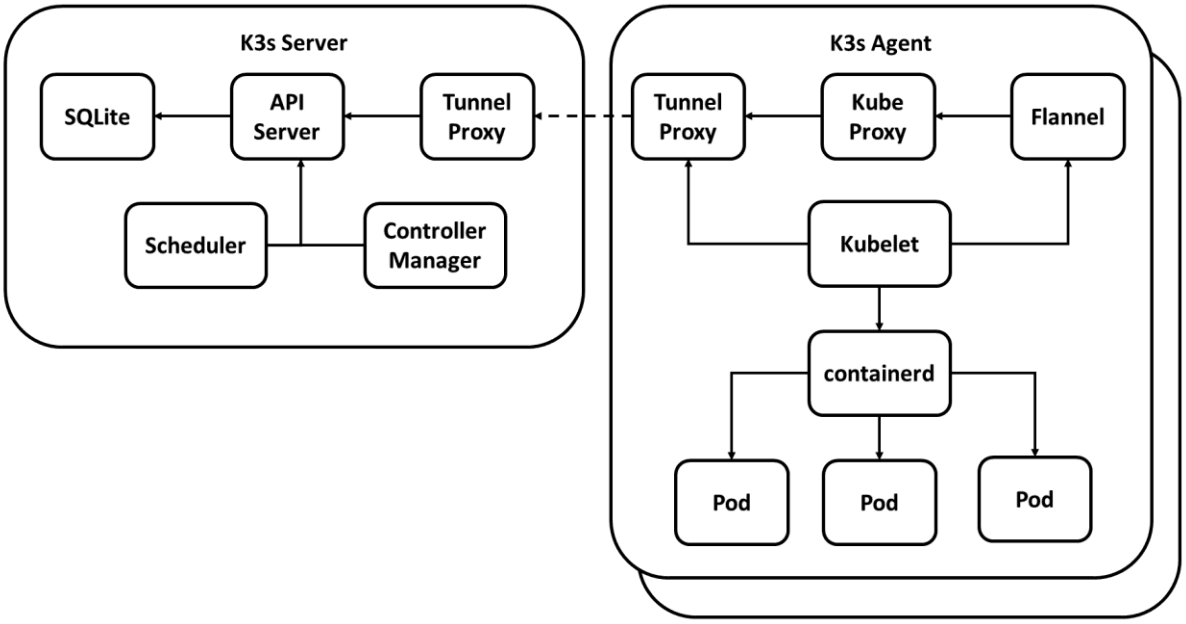
```
azureuser@microk8s-vm:~/snap/microk8s$ sudo microk8s inspect
Inspecting Certificates
Inspecting services
  Service snap.microk8s.daemon-cluster-agent is running
  Service snap.microk8s.daemon-containerd is running
  Service snap.microk8s.daemon-apiserver-kicker is running
  Service snap.microk8s.daemon-kubelite is running
  Copy service arguments to the final report tarball
Inspecting AppArmor configuration
Gathering system information
  Copy processes list to the final report tarball
  Copy snap list to the final report tarball
  Copy VM name (or none) to the final report tarball
  Copy disk usage information to the final report tarball
  Copy memory usage information to the final report tarball
  Copy server uptime to the final report tarball
  Copy current linux distribution to the final report tarball
  Copy openSSL information to the final report tarball
  Copy network configuration to the final report tarball
Inspecting kubernetes cluster
  Inspect kubernetes cluster
Inspecting juju
  Inspect Juju
```

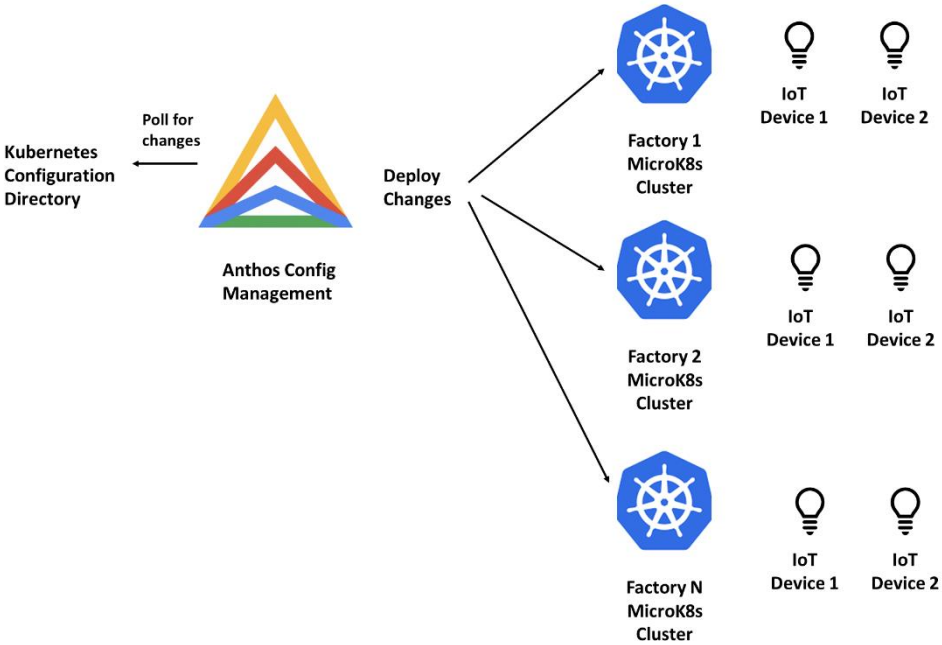
Chapter 3: Essentials of IoT and Edge Computing

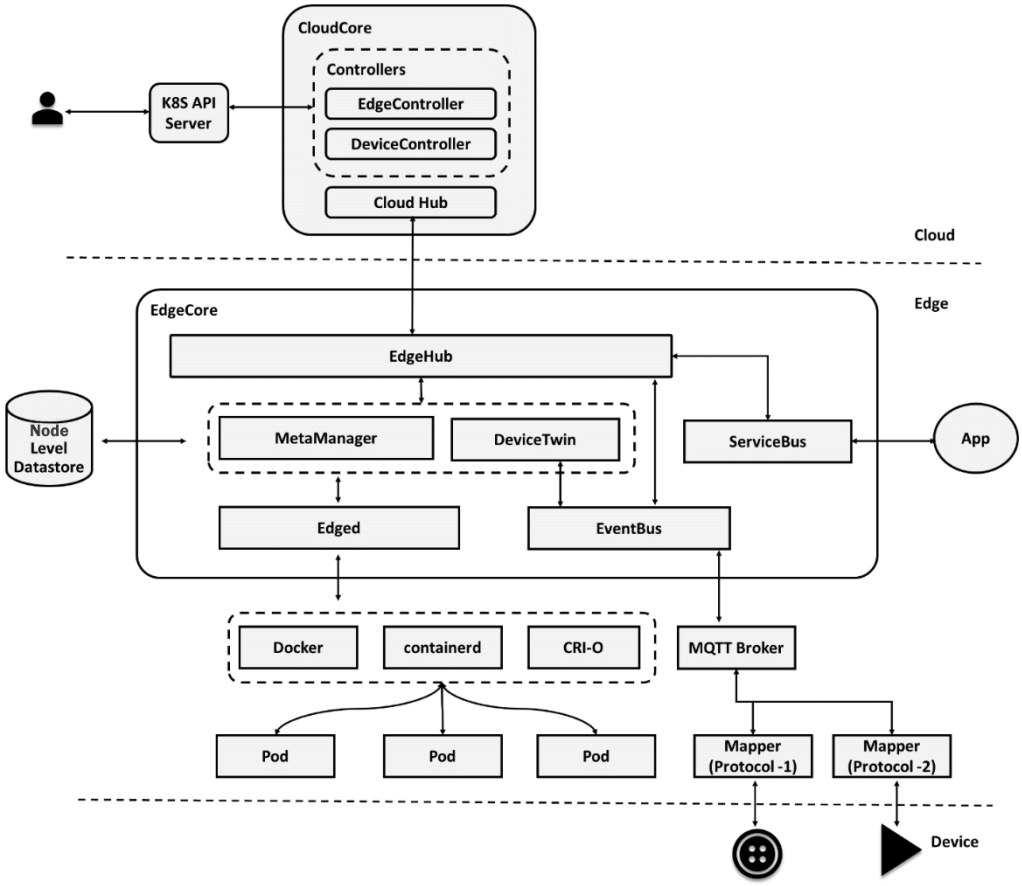


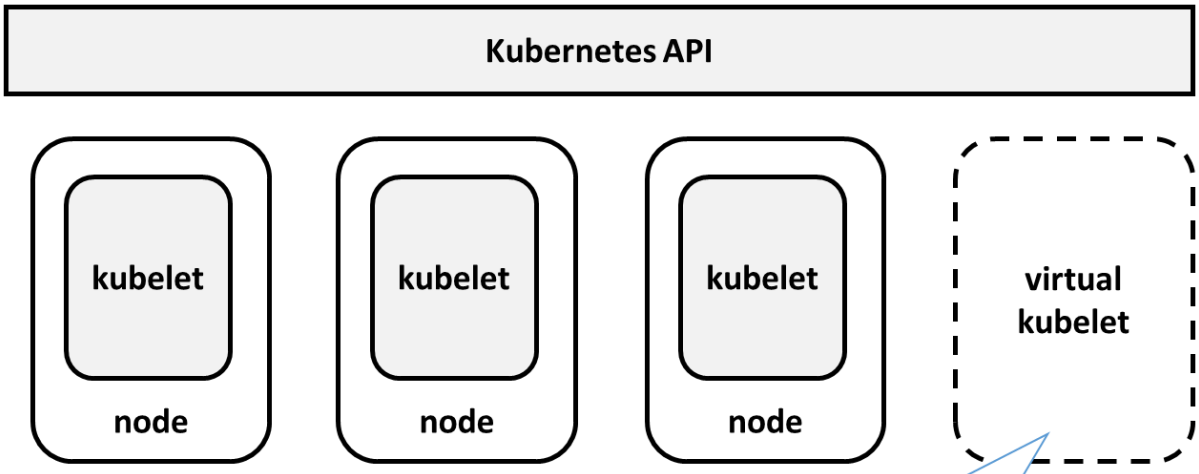

Cloud
Edge Nodes
Devices
Data center
Enterprise Layer
Server
Edge Server
Plant Applications
Edge Server
Gateway
Edge Connectivity Layer
Gateway
Device Layer

Chapter 4: Handling the Kubernetes Platform for IoT and Edge Computing



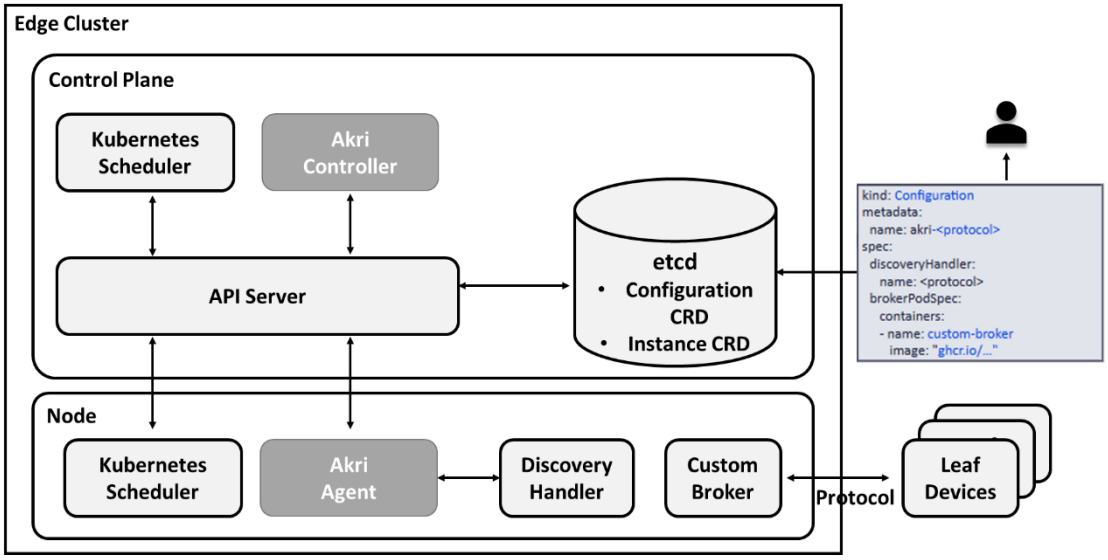




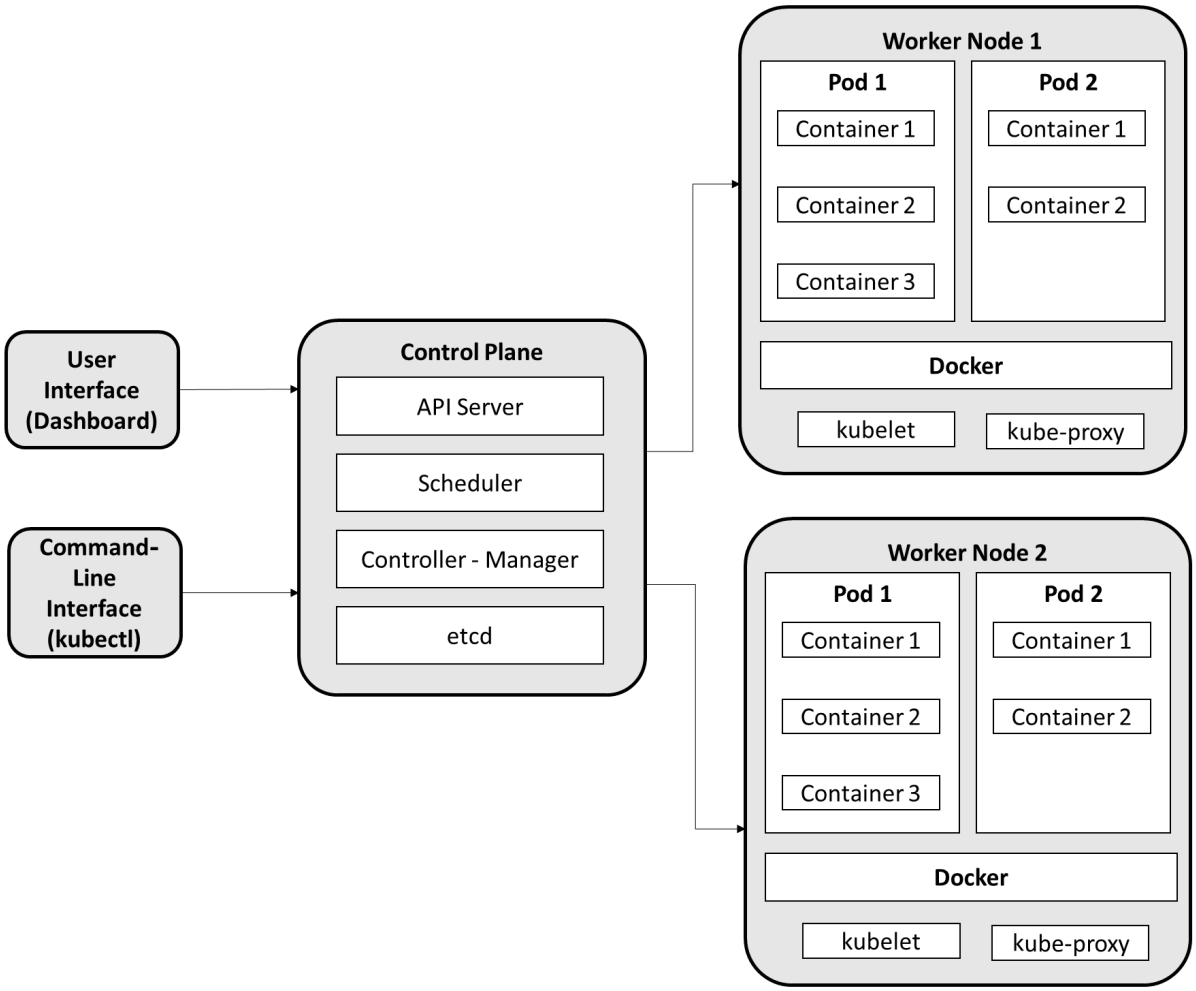


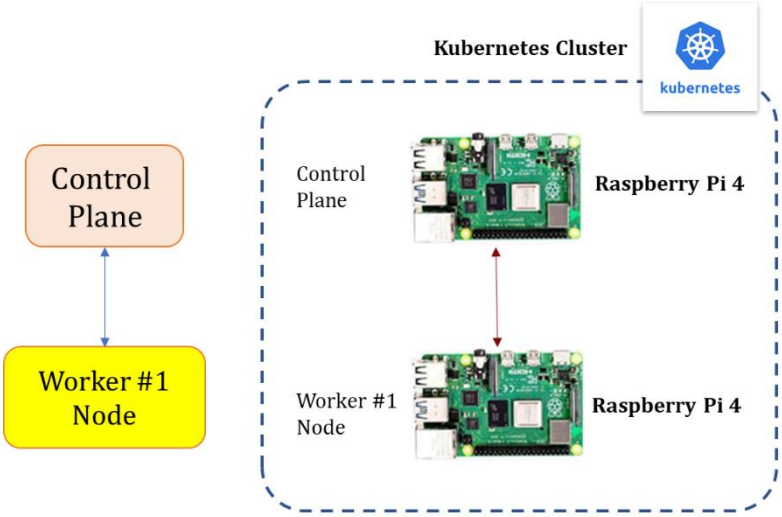
Virtual Kubelet registers itself as a “node” and allows to deploy pods and containers with their own APIs.

Virtual Kubelet works with Azure Container Instances, Azure IoT Edge, and AWS Fargate control plane.

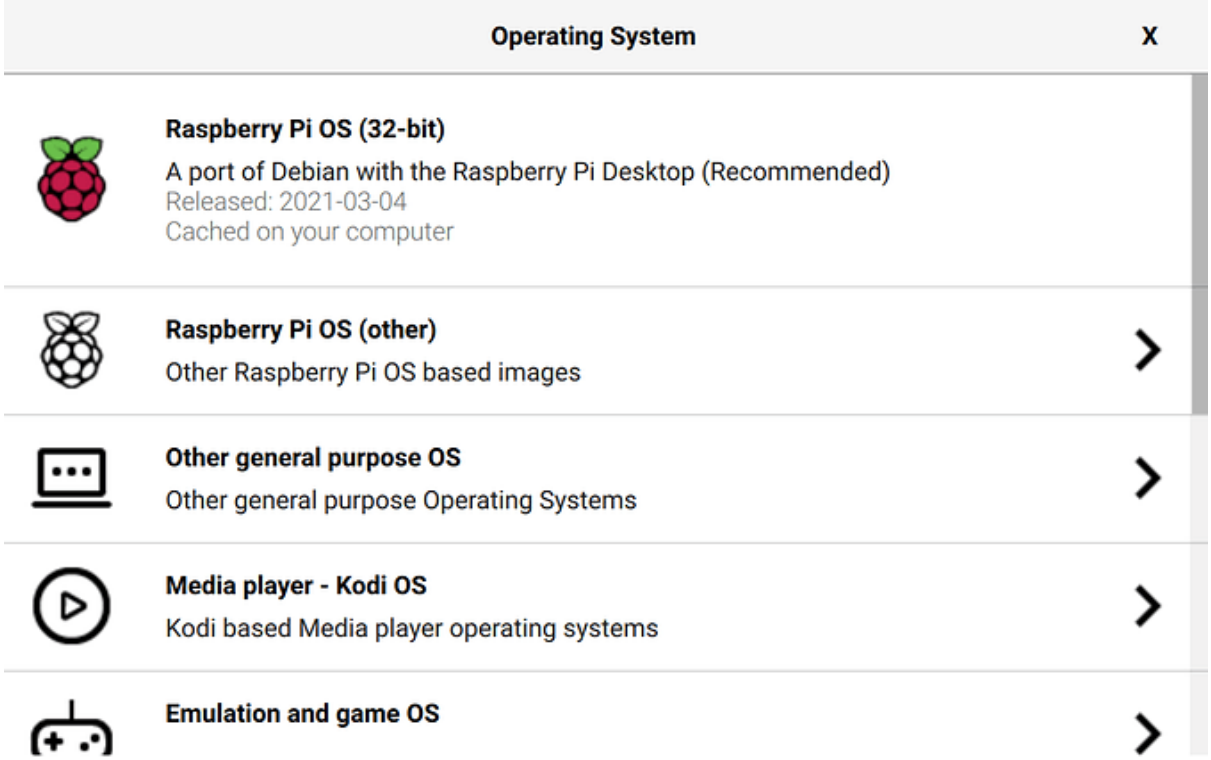

Edge Cluster
Control Plane
Kubernetes Scheduler
Akri Controller
API Server
etcd
• Configuration CRD
• Instance CRD
kind: Configuration
metadata:
name: akri-<protocol>
spec:
discoveryHandler:
name: <protocol>
brokerPodSpec:
containers:
- name: custom-broker
image: "ghcr.io/..."
Node
Kubernetes Scheduler
Akri Agent
Discovery Handler
Custom Broker
Protocol
Leaf Devices

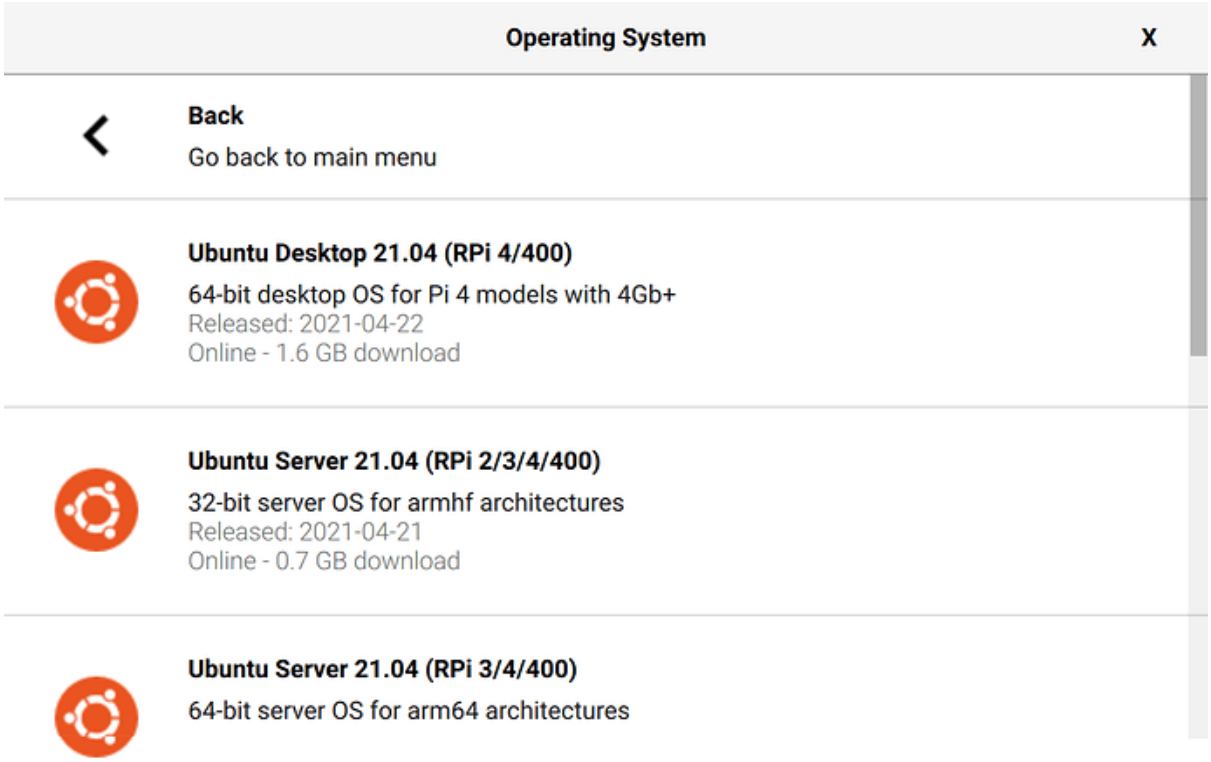
Chapter 5: Creating and Implementing Updates on a Multi-Node Raspberry Pi Kubernetes Clusters



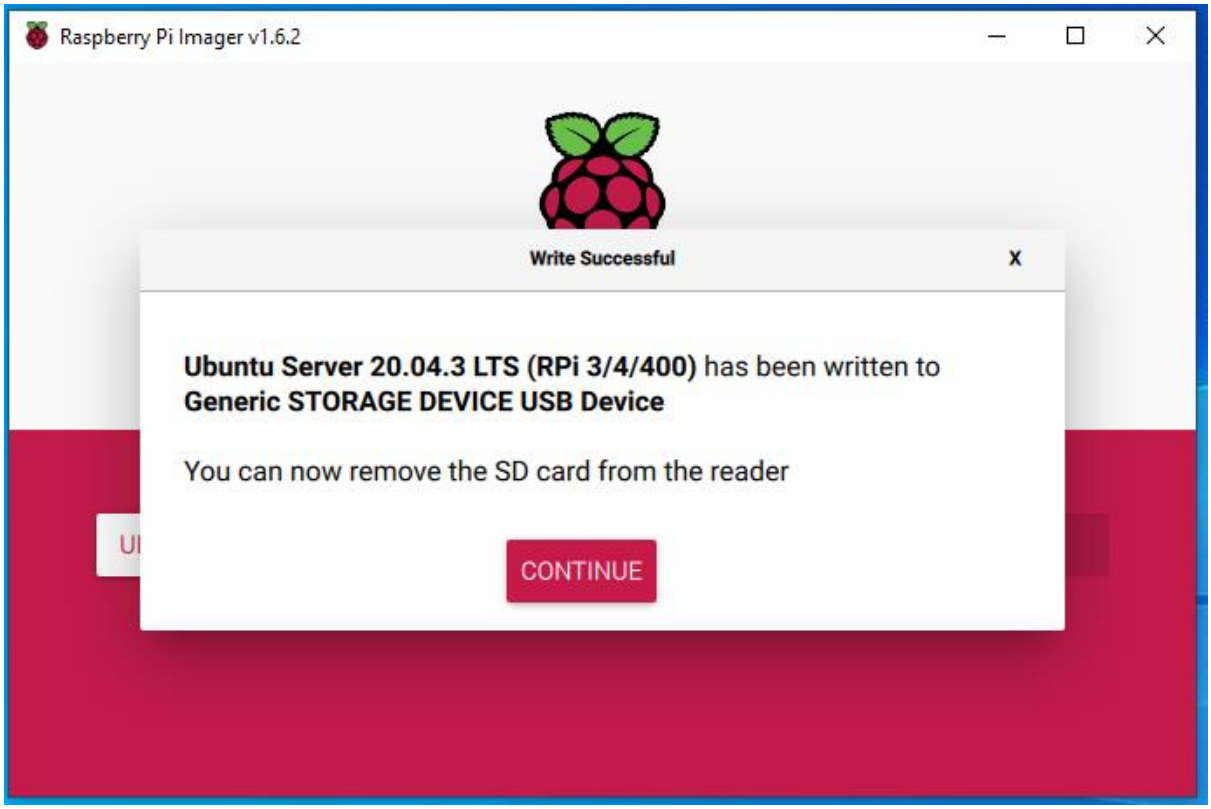


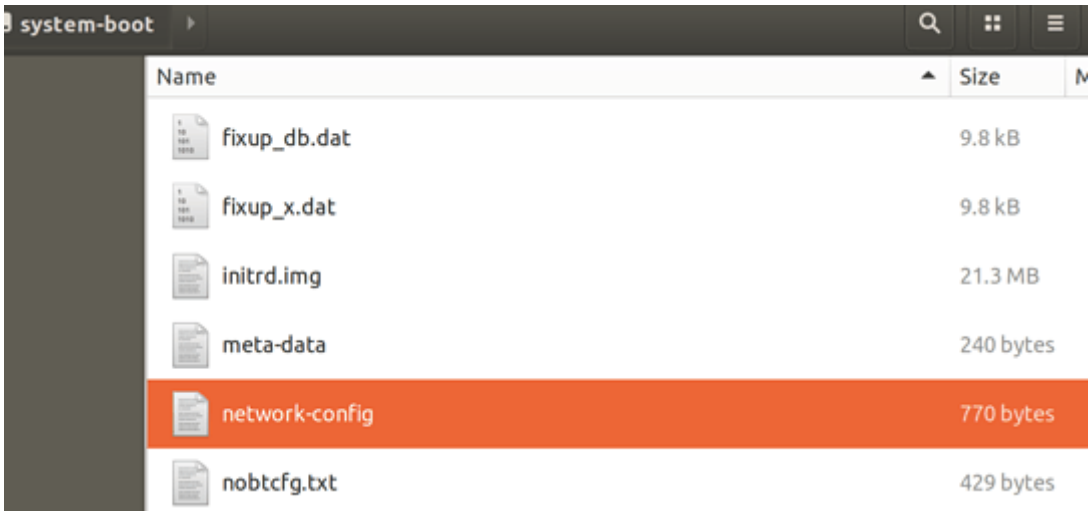

Raspberry Pi Imager v1.6
Raspberry Pi
Operating System
Storage
CHOOSE OS
CHOOSE STORAGE
WRITE

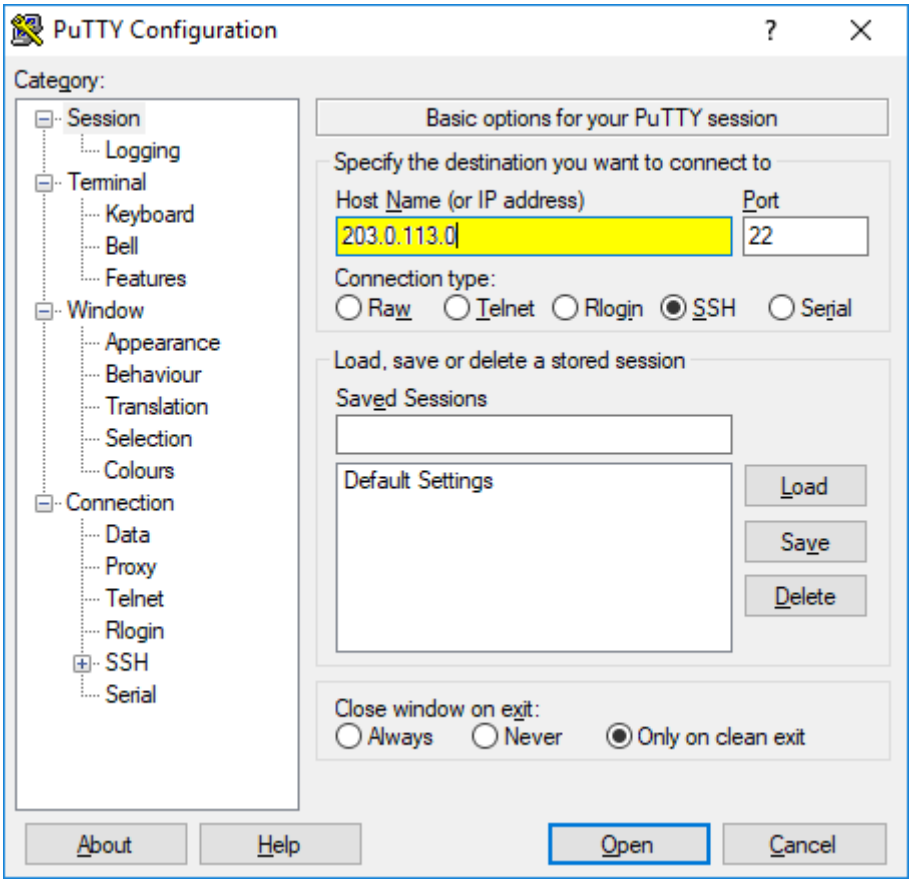

Operating System
X
Raspberry Pi OS (32-bit)
A port of Debian with the Raspberry Pi Desktop (Recommended)
Released: 2021-03-04
Cached on your computer
Raspberry Pi OS (other)
Other Raspberry Pi OS based images
Other general purpose OS
Other general purpose Operating Systems
Media player - Kodi OS
Kodi based Media player operating systems
Emulation and game OS


Operating System
X
Back
Go back to main menu
Ubuntu Desktop 21.04 (RPi 4/400)
64-bit desktop OS for Pi 4 models with 4Gb+
Released: 2021-04-22
Online - 1.6 GB download
Ubuntu Server 21.04 (RPi 2/3/4/400)
32-bit server OS for armhf architectures
Released: 2021-04-21
Online - 0.7 GB download
Ubuntu Server 21.04 (RPi 3/4/400)
64-bit server OS for arm64 architectures


Raspberry Pi Imager v1.6.2
Raspberry Pi
Operating System
Storage
UBUNTU SERVER 20.04.3 LTS (RPI 3/4/400)
GENERIC ST...
WRITE
Writing... 0%
CANCEL WRITE


Raspberry Pi Imager v1.6.2
Write Successful
X
Ubuntu Server 20.04.3 LTS (RPi 3/4/400) has been written to
Generic STORAGE DEVICE USB Device
You can now remove the SD card from the reader
CONTINUE


system-boot
Name
Size
fixup_db.dat
9.8 kB
fixup_x.dat
9.8 kB
initrd.img
21.3 MB
meta-data
240 bytes
network-config
770 bytes
nobtcfg.txt
429 bytes




```
ubuntu@controlplane:~$ sudo snap install microk8s --classic
microk8s (1.23/stable) v1.23.3 from Canonical✓ installed
ubuntu@controlplane:~$
```

```
ubuntu@controlplane:~$ microk8s status
Insufficient permissions to access MicroK8s.
You can either try again with sudo or add the user ubuntu to the 'microk8s' grou
p:

    sudo usermod -a -G microk8s ubuntu
    sudo chown -f -R ubuntu ~/.kube

After this, reload the user groups either via a reboot or by running 'newgrp mic
rok8s'.
```

```
ubuntu@controlplane:~$ microk8s status
microk8s is running
high-availability: no
  datastore master nodes: 127.0.0.1:19001
  datastore standby nodes: none
addons:
  enabled:
    ha-cluster           # Configure high av
  disabled:
    dashboard            # The Kubernetes da
    dashboard-ingress    # Ingress definitio
```

```
ubuntu@controlplane:~$ sudo snap alias microk8s.kubectl kubectl
Added:
  - microk8s.kubectl as kubectl
ubuntu@controlplane:~$
```

```
ubuntu@controlplane:~$ kubectl get nodes
NAME           STATUS   ROLES    AGE   VERSION
controlplane   Ready    <none>   12m   v1.23.3-2+0d2db09fa6fbbb
ubuntu@controlplane:~$
```

```
ubuntu@worker1:~$ sudo snap install microk8s --classic
microk8s (1.23/stable) v1.23.3 from Canonical✓ installed
```

```
ubuntu@worker1:~$ microk8s status
microk8s is running
high-availability: no
  datastore master nodes: 127.0.0.1:19001
  datastore standby nodes: none
addons:
  enabled:
    ha-cluster           # Configure high av
  disabled:
    dashboard            # The Kubernetes da
    dashboard-ingress    # Ingress definitio
```

```
ubuntu@worker1:~$ kubectl get nodes
NAME       STATUS   ROLES    AGE   VERSION
worker1    Ready    <none>   13m   v1.23.3-2+0d2db09fa6fbbb
ubuntu@worker1:~$
```

```
ubuntu@controlplane:~$ sudo microk8s.add-node
From the node you wish to join to this cluster, run the following:
microk8s join 192.168.1.7:25000/fba12c2f1bce9fbe70208443565aaa04/3e2f115c73d6

Use the '--worker' flag to join a node as a worker not running the control plane,
microk8s join 192.168.1.7:25000/fba12c2f1bce9fbe70208443565aaa04/3e2f115c73d6 --wo

If the node you are adding is not reachable through the default interface you can
he following:
microk8s join 192.168.1.7:25000/fba12c2f1bce9fbe70208443565aaa04/3e2f115c73d6
ubuntu@controlplane:~$
```

```
ubuntu@worker1:~$ sudo microk8s join 192.168.1.7:25000/fba12c2f1bce9fbe702084435
65aaa04/3e2f115c73d6 --worker
Contacting cluster at 192.168.1.7

The node has joined the cluster and will appear in the nodes list in a few secon
ds.

Currently this worker node is configured with the following kubernetes API serve
r endpoints:
    - 192.168.1.7 and port 16443, this is the cluster node contacted during the
join operation.

If the above endpoints are incorrect, incomplete or if the API servers are behin
d a loadbalancer please update
/var/snap/microk8s/current/args/traefik/provider.yaml

ubuntu@worker1:~$
```

```
ubuntu@controlplane:~$ kubectl get nodes
NAME           STATUS   ROLES    AGE   VERSION
controlplane   Ready    <none>   17m   v1.23.3-2+0d2db09fa6fbbb
worker1        Ready    <none>   27s   v1.23.3-2+0d2db09fa6fbbb
ubuntu@controlplane:~$
```


```
ubuntu@controlplane:~$ kubectl apply -f deployment.yaml
deployment.apps/nginx-deployment created
```

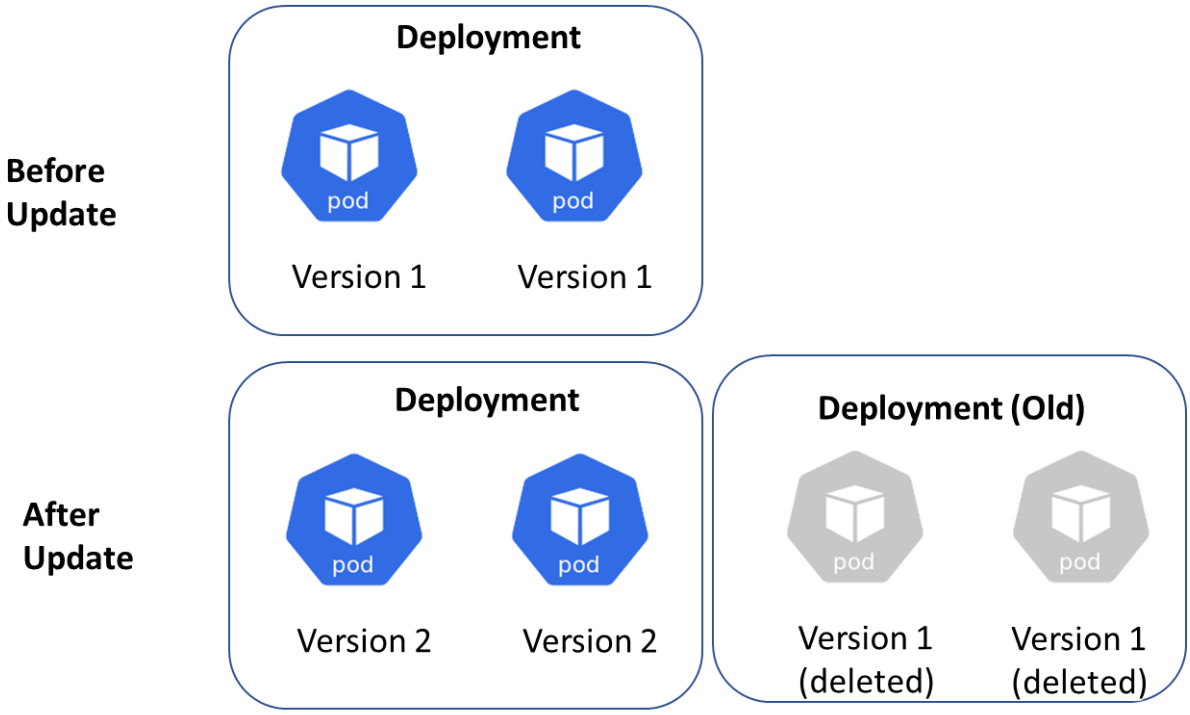
```
ubuntu@controlplane:~$ kubectl describe deployment nginx-deployment
Name:                   nginx-deployment
Namespace:              default
CreationTimestamp:      Sat, 05 Mar 2022 12:11:03 +0000
Labels:                 <none>
Annotations:            deployment.kubernetes.io/revision: 1
Selector:               app=nginx
Replicas:               2 desired | 2 updated | 2 total | 2 available | 0 unavaila
StrategyType:           RollingUpdate
MinReadySeconds:        0
RollingUpdateStrategy:  25% max unavailable, 25% max surge
Pod Template:
  Labels:  app=nginx
  Containers:
   nginx:
    Image:        nginx:1.14.2
    Port:         80/TCP
    Host Port:    0/TCP
    Environment:  <none>
    Mounts:       <none>
  Volumes:        <none>
Conditions:
  Type           Status  Reason
  ----           ------  ------
  Available      True    MinimumReplicasAvailable
  Progressing    True    NewReplicaSetAvailable
OldReplicaSets:  <none>
NewReplicaSet:   nginx-deployment-57d554699f (2/2 replicas created)
Events:
  Type    Reason             Age   From                   Message
  ----    ------             ----  ----                   -------
  Normal  ScalingReplicaSet  2m5s  deployment-controller  Scaled up replica set ng
ubuntu@controlplane:~$
```

```
ubuntu@controlplane:~$ kubectl get pods -l app=nginx
NAME                                READY   STATUS    RESTARTS   AGE
nginx-deployment-57d554699f-clxd5   1/1     Running   0          3m9s
nginx-deployment-57d554699f-8hjbv   1/1     Running   0          3m9s
ubuntu@controlplane:~$
```

```
ubuntu@controlplane:~$ kubectl get pods -l app=nginx -o wide
NAME                                READY   STATUS    RESTARTS   AGE    IP            NODE
S
nginx-deployment-57d554699f-clxd5   1/1     Running   0          8m56s  10.1.49.70    controlplane
nginx-deployment-57d554699f-8hjbv   1/1     Running   0          8m56s  10.1.235.129  worker1
ubuntu@controlplane:~$
```

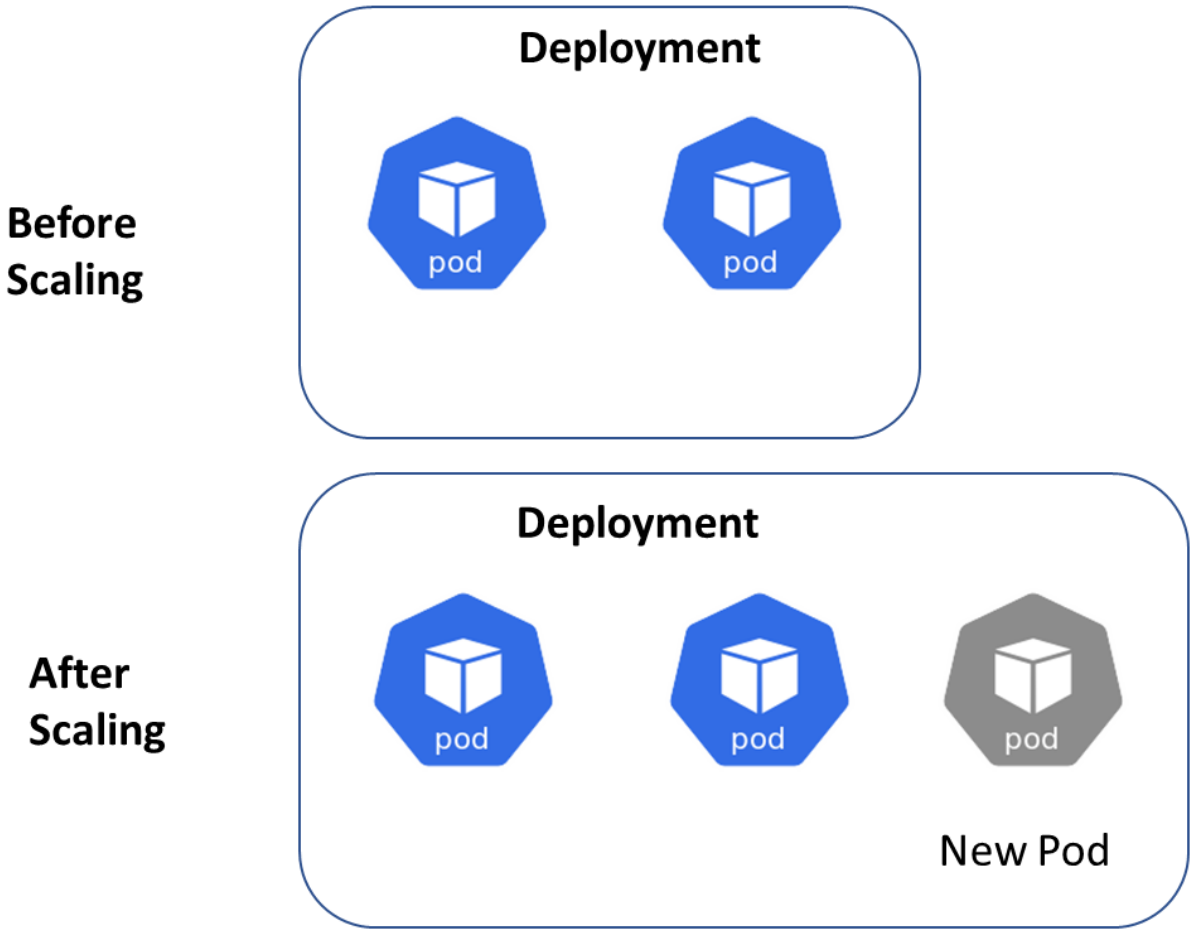
```
ubuntu@controlplane:~$ kubectl apply -f deployment-update.yaml
deployment.apps/nginx-deployment configured
ubuntu@controlplane:~$
```

```
ubuntu@controlplane:~$ kubectl get pods -l app=nginx
NAME                                READY   STATUS    RESTARTS   AGE
nginx-deployment-6796bd85dd-lf4zb   1/1     Running   0          7m36s
nginx-deployment-6796bd85dd-6fgtp   1/1     Running   0          95s
```



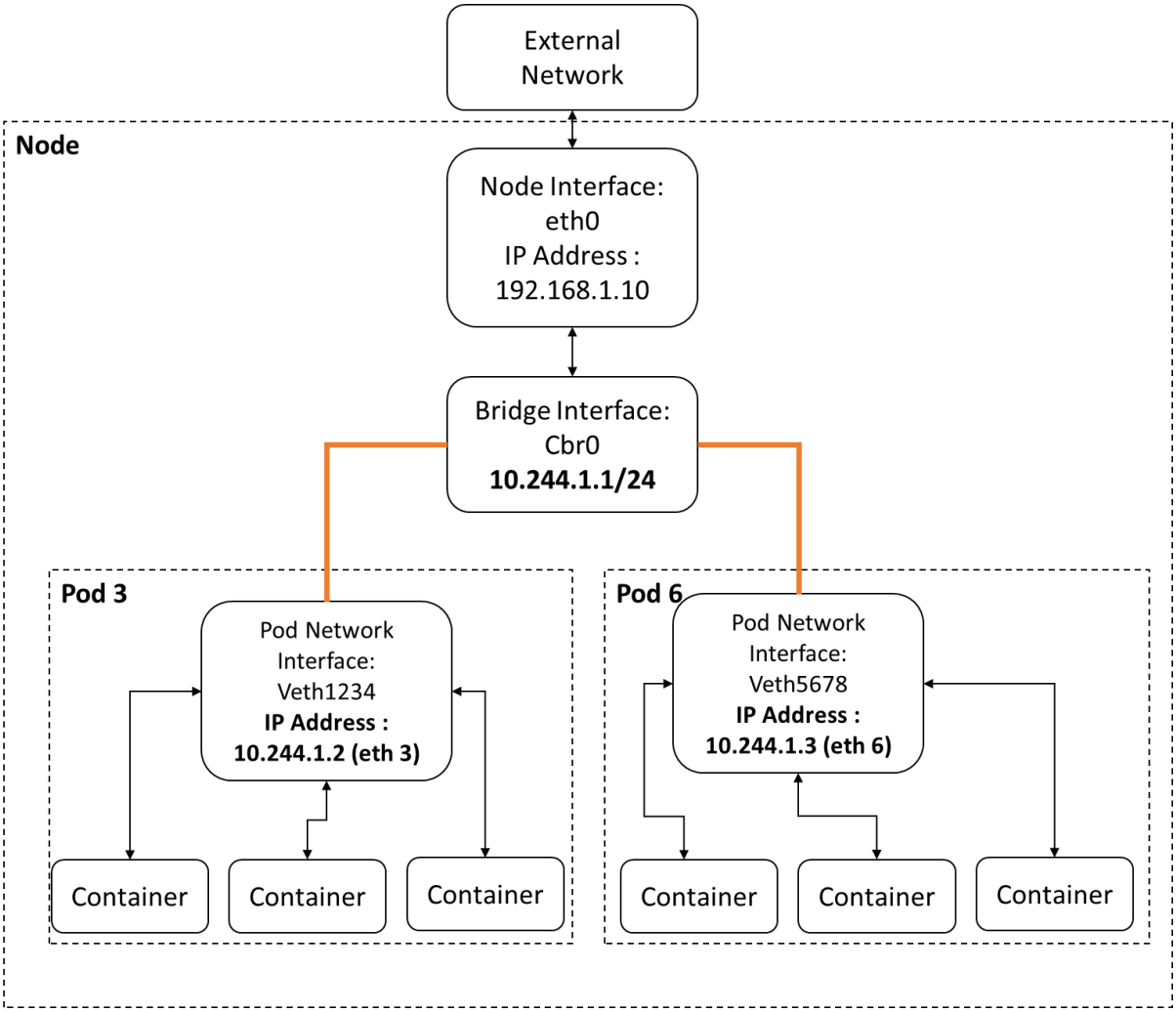

```
ubuntu@controlplane:~$ kubectl apply -f deployment-scale.yaml
deployment.apps/nginx-deployment configured
ubuntu@controlplane:~$
```

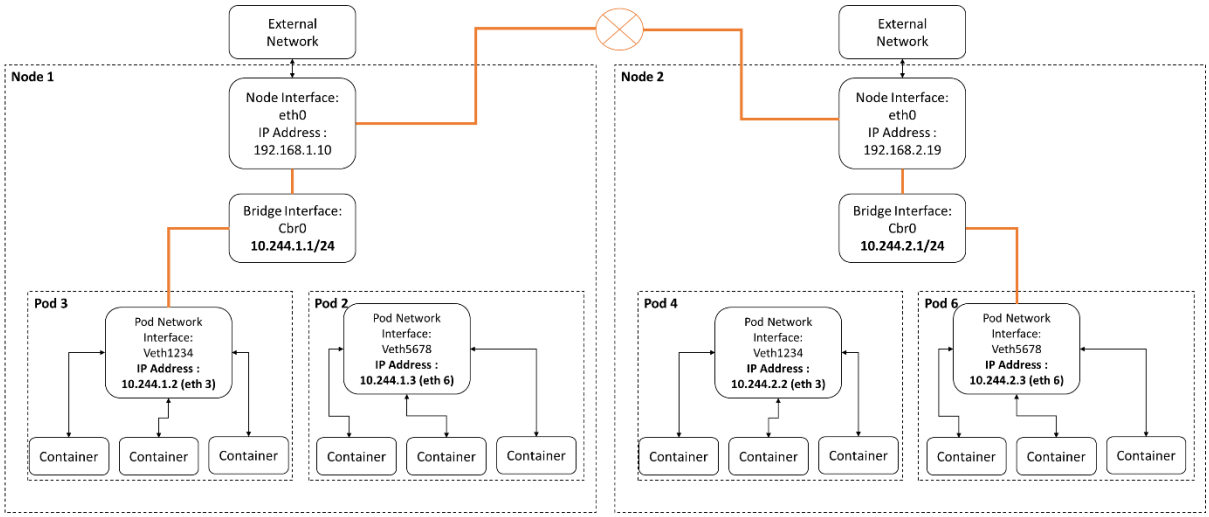
```
ubuntu@controlplane:~$ kubectl get pods -l app=nginx
NAME                                READY   STATUS    RESTARTS   AGE
nginx-deployment-9456bbbf9-sgj9b    1/1     Running   0          2m48s
nginx-deployment-9456bbbf9-fmhnb    1/1     Running   0          2m48s
nginx-deployment-9456bbbf9-xsnfj    1/1     Running   0          2m22s
nginx-deployment-9456bbbf9-r6w58    1/1     Running   0          2m22s
ubuntu@controlplane:~$
```

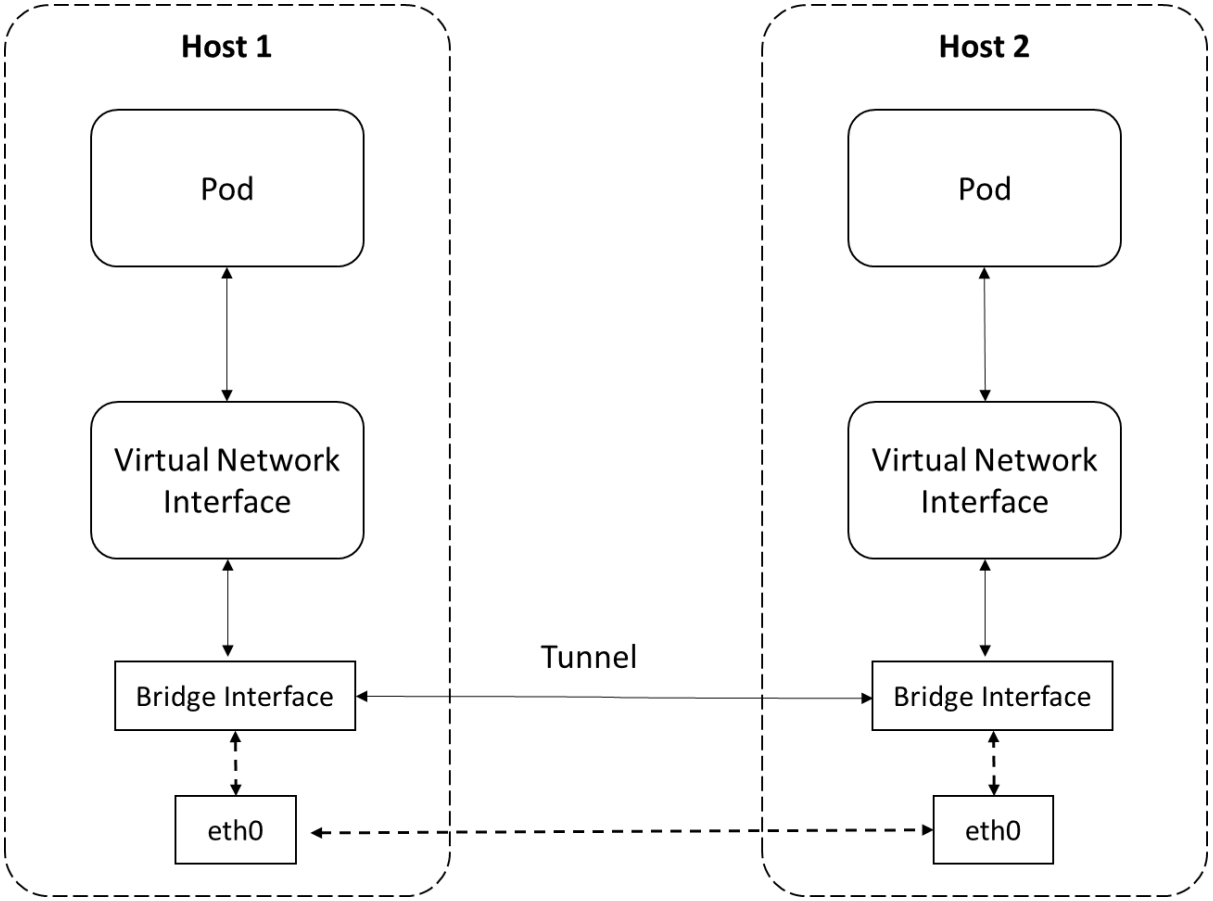


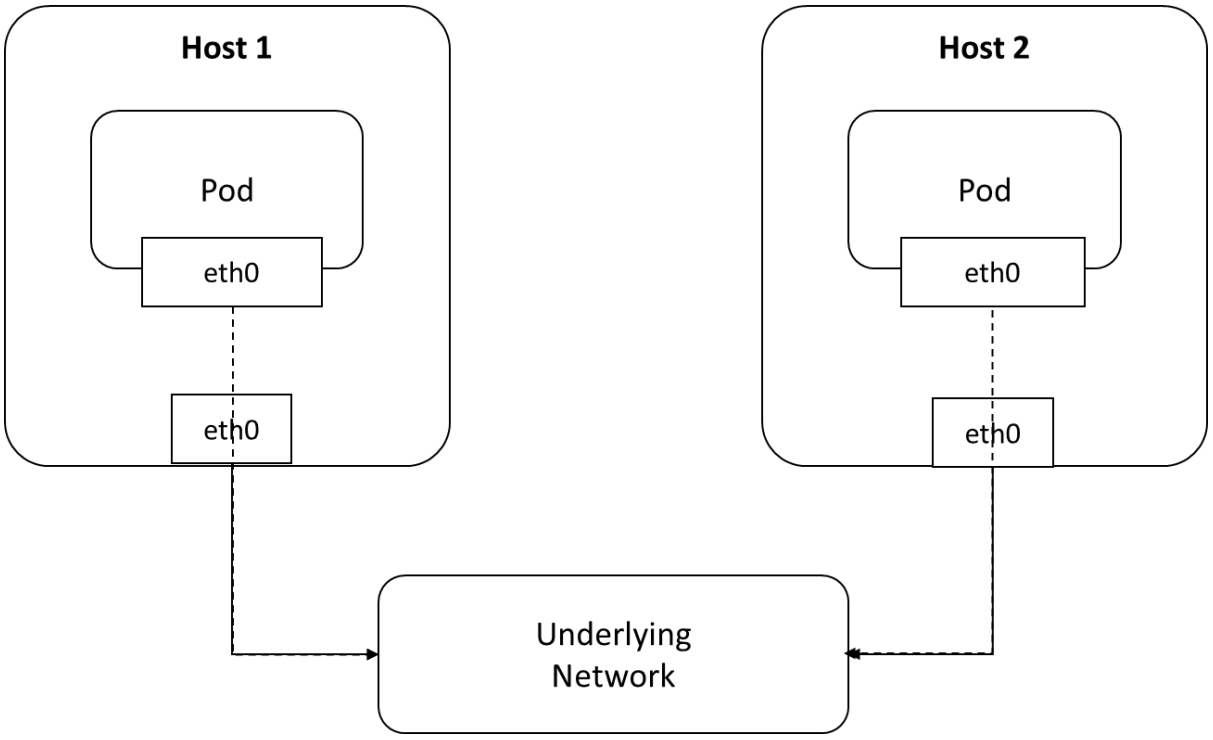

```
ubuntu@controlplane:~$ kubectl get pods -o wide
NAME                                READY   STATUS    RESTARTS   AGE     IP             NODE
nginx-deployment-9456bbbf9-sgj9b    1/1     Running   0          4m54s   10.1.49.70     controlplane
nginx-deployment-9456bbbf9-fmhnb    1/1     Running   0          4m54s   10.1.235.133   worker1
nginx-deployment-9456bbbf9-xsnfj    1/1     Running   0          4m28s   10.1.49.71     controlplane
nginx-deployment-9456bbbf9-r6w58    1/1     Running   0          4m28s   10.1.235.134   worker1
ubuntu@controlplane:~$
```

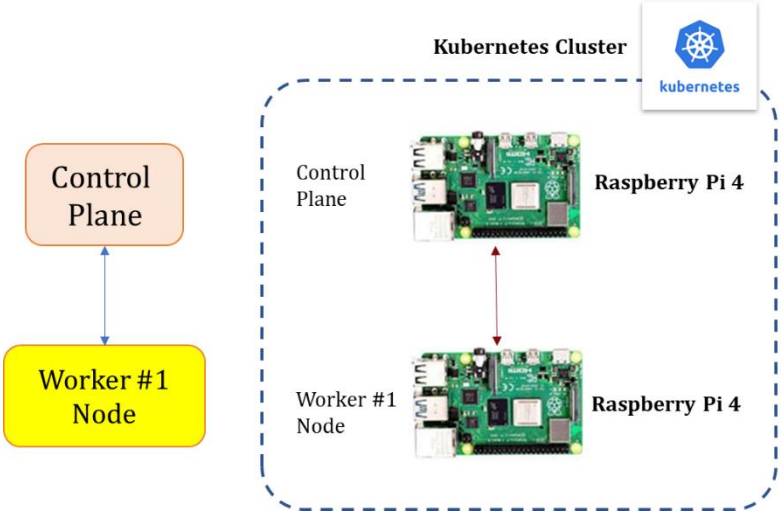
Chapter 6: Configuring Connectivity for Containers






Host 1
Pod
Virtual Network Interface
Bridge Interface
eth0
Tunnel
Host 2
Pod
Virtual Network Interface
Bridge Interface
eth0


Host 1
Pod
eth0
eth0
Host 2
Pod
eth0
eth0
Underlying Network




```
ubuntu@controlplane:~$ kubectl get pods -A |grep calico
kube-system      calico-kube-controllers-548d5485bf-tfnhc   1/1     Running
kube-system      calico-node-n6xkg                          1/1     Running
ubuntu@controlplane:~$
```

```
ubuntu@controlplane:~$ kubectl create deployment nginx --image=nginx
deployment.apps/nginx created
ubuntu@controlplane:~$
```

```
ubuntu@controlplane:~$ kubectl expose deployment nginx --port=80
service/nginx exposed
ubuntu@controlplane:~$
```

```
ubuntu@controlplane:~$ kubectl get svc | grep nginx
nginx                  ClusterIP      10.152.183.117   <none>        80/TCP        37s
ubuntu@controlplane:~$
```

```
ubuntu@controlplane:~$ kubectl run access --rm -ti --image busybox /bin/sh
If you don't see a command prompt, try pressing enter.
/ #
```

```
/ # wget -q nginx -O -
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
html { color-scheme: light dark; }
body { width: 35em; margin: 0 auto;
font-family: Tahoma, Verdana, Arial, sans-serif; }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>

<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>

<p><em>Thank you for using nginx.</em></p>
</body>
</html>
/ #
```

```
ubuntu@controlplane:~$ kubectl apply -f calico-policy.yaml
networkpolicy.networking.k8s.io/default-deny created
ubuntu@controlplane:~$
```

```
ubuntu@controlplane:~$ kubectl run access --rm -ti --image busybox /bin/sh
If you don't see a command prompt, try pressing enter.
/ #
```

```
wget: download timed out
/ # wget -q nginx -O -
```

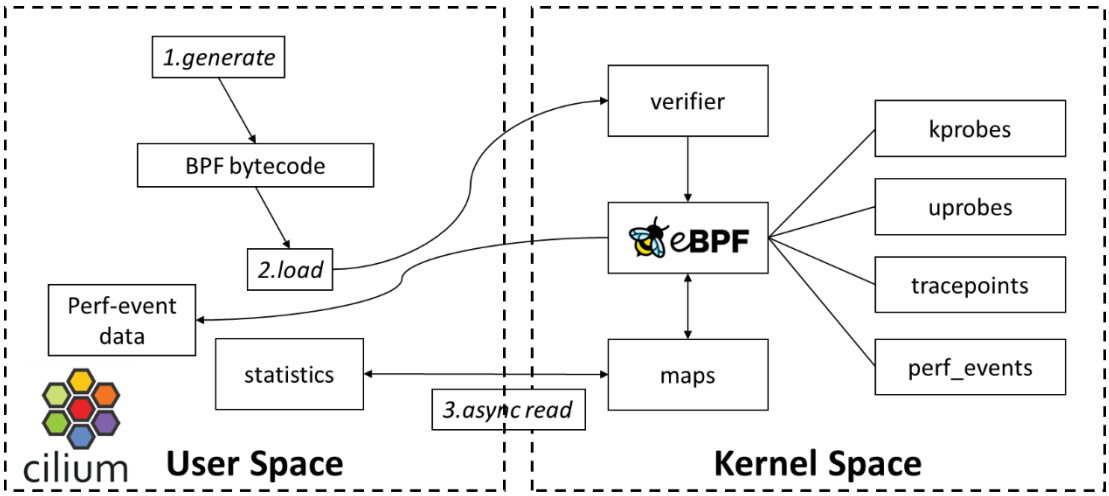
```
/ # wget -q --timeout=5 nginx -O -
wget: download timed out
```

```
ubuntu@controlplane:~$ kubectl apply -f calico-policy.yaml
networkpolicy.networking.k8s.io/access-nginx created
ubuntu@controlplane:~$
```

```
/ # wget -q nginx -O -
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
html { color-scheme: light dark; }
body { width: 35em; margin: 0 auto;
font-family: Tahoma, Verdana, Arial, sans-serif; }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>

<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>

<p><em>Thank you for using nginx.</em></p>
</body>
</html>
/ #
```

```
ubuntu@controlplane:~$ kubectl run access1 --rm -ti --image busybox /bin/sh
If you don't see a command prompt, try pressing enter.
/ #
```

```
ubuntu@Master:~$ kubectl run access1 --rm -ti --image busybox /bin/sh
If you don't see a command prompt, try pressing enter.
/ #  wget -q nginx -O -
^C
/ #  wget -q --timeout=5 nginx -O -
wget: download timed out
/ #
```



- Network Policy
- Services & Load Balancing
- Bandwidth Management
- Flow & Policy Logging
- Metrics

```
ubuntu@microk8s2-worker1:~$ microk8s enable cilium
Enabling Helm 3
Fetching helm version v3.5.0.
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 11.7M  100 11.7M    0     0  7403k      0  0:00:01  0:00:01 --:--:-- 7403k
Helm 3 is enabled
Restarting kube-apiserver
Restarting kubelet
Enabling Cilium
Fetching cilium version v1.10.
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   131    0   131    0     0    433      0 --:--:-- --:--:-- --:--:--   436
100 32.2M    0 32.2M    0     0  6773k      0 --:--:--  0:00:04 --:--:-- 7318k
Deploying /var/snap/microk8s/2948/actions/cilium.yaml. This may take several minutes.
serviceaccount/cilium created
serviceaccount/cilium-operator created
configmap/cilium-config created
clusterrole.rbac.authorization.k8s.io/cilium created
clusterrole.rbac.authorization.k8s.io/cilium-operator created
clusterrolebinding.rbac.authorization.k8s.io/cilium created
clusterrolebinding.rbac.authorization.k8s.io/cilium-operator created
Warning: spec.template.spec.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringE
[0].key: beta.kubernetes.io/os is deprecated since v1.14; use "kubernetes.io/os" instead
Warning: spec.template.metadata.annotations[scheduler.alpha.kubernetes.io/critical-pod]:
assName" field instead
daemonset.apps/cilium created
deployment.apps/cilium-operator created
Waiting for daemon set spec update to be observed...
```

```
clusterrole.rbac.authorization.k8s.io "calico-kube-controllers" deleted
clusterrolebinding.rbac.authorization.k8s.io "calico-kube-controllers" deleted
clusterrole.rbac.authorization.k8s.io "calico-node" deleted
clusterrolebinding.rbac.authorization.k8s.io "calico-node" deleted
daemonset.apps "calico-node" deleted
serviceaccount "calico-node" deleted
deployment.apps "calico-kube-controllers" deleted
serviceaccount "calico-kube-controllers" deleted
Warning: policy/v1beta1 PodDisruptionBudget is deprecated in v1.21+, unavailable in v1.25+; use
poddisruptionbudget.policy "calico-kube-controllers" deleted
Cilium is enabled
ubuntu@microk8s2-worker1:~$
```

```
ubuntu@microk8s:~$microk8s.cilium status
KVStore:                Ok   Disabled
Kubernetes:             Ok   1.23+ (v1.23.3-2+d441060727c463) [linux/amd64]
Kubernetes APIs:        ["cilium/v2::CiliumClusterwideNetworkPolicy", "cilium/v2::CiliumEndpoint", "cilium/v2::CiliumNetworkPolicy", "
cilium/v2::CiliumNode", "core/v1::Namespace", "core/v1::Node", "core/v1::Pods", "core/v1::Service", "discovery/v1::EndpointSlice", "ne
tworking.k8s.io/v1::NetworkPolicy"]
KubeProxyReplacement:   Disabled
Cilium:                 Ok   1.10.7 (v1.10.7-3e77756)
NodeMonitor:            Listening for events on 1 CPUs with 64x4096 of shared memory
Cilium health daemon:   Ok
IPAM:                   IPv4: 2/254 allocated from 10.0.0.0/24,
BandwidthManager:       Disabled
Host Routing:           Legacy
Masquerading:           IPTables [IPv4: Enabled, IPv6: Disabled]
Controller Status:      18/18 healthy
Proxy Status:           OK, ip 10.0.0.205, 0 redirects active on ports 10000-20000
Hubble:                 Ok   Current/Max Flows: 100/4095 (2.44%), Flows/s: 0.15   Metrics: Disabled
Encryption:             Disabled
Cluster health:         1/1 reachable   (2022-02-17T13:13:35Z)
ubuntu@microk8s:~$
```

```
ubuntu@microk8s:~$microk8s enable dns
Enabling DNS
Applying manifest
serviceaccount/coredns created
configmap/coredns created
deployment.apps/coredns created
service/kube-dns created
clusterrole.rbac.authorization.k8s.io/coredns created
clusterrolebinding.rbac.authorization.k8s.io/coredns created
Restarting kubelet
DNS is enabled
```

```
ubuntu@microk8s:~$kubectl create deployment nginx-cilium --image=nginx
deployment.apps/nginx-cilium created
ubuntu@microk8s:~$
```

```
ubuntu@microk8s:~$kubectl expose deployment nginx-cilium --port=80
service/nginx-cilium exposed
ubuntu@microk8s:~$
```

```
ubuntu@microk8s:~$kubectl run access --rm -ti --image busybox /bin/sh
If you don't see a command prompt, try pressing enter.
/ # wget -q nginx-cilium -O -
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
html { color-scheme: light dark; }
body { width: 35em; margin: 0 auto;
font-family: Tahoma, Verdana, Arial, sans-serif; }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>

<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>

<p><em>Thank you for using nginx.</em></p>
</body>
</html>
/ #
```

```
ubuntu@microk8s:~$kubectl apply -f cilium-policy.yaml
networkpolicy.networking.k8s.io/cilium-deny created
ubuntu@microk8s:~$
```

```
ubuntu@microk8s:~$microk8s cilium policy get
[
  {
    "endpointSelector": {
      "matchLabels": {
        "k8s:io.kubernetes.pod.namespace": "default"
      }
    },
    "ingress": [
      {}
    ],
    "labels": [
      {
        "key": "io.cilium.k8s.policy.derived-from",
        "value": "NetworkPolicy",
        "source": "k8s"
      },
      {
        "key": "io.cilium.k8s.policy.name",
        "value": "cilium-deny",
        "source": "k8s"
      },
      {
        "key": "io.cilium.k8s.policy.namespace",
        "value": "default",
        "source": "k8s"
      },
      {
        "key": "io.cilium.k8s.policy.uid",
        "value": "46c9e099-1fc0-49b8-9441-32471f47624d",
        "source": "k8s"
      }
    ]
  }
]
Revision: 2
ubuntu@microk8s:~$
```

```
/ # wget -q --timeout=5 nginx-cilium -O -
wget: download timed out
/ #
```

```
ubuntu@microk8s:~$kubectl apply -f cilium-policy.yaml
networkpolicy.networking.k8s.io/cilium-deny configured
ubuntu@microk8s:~$
```

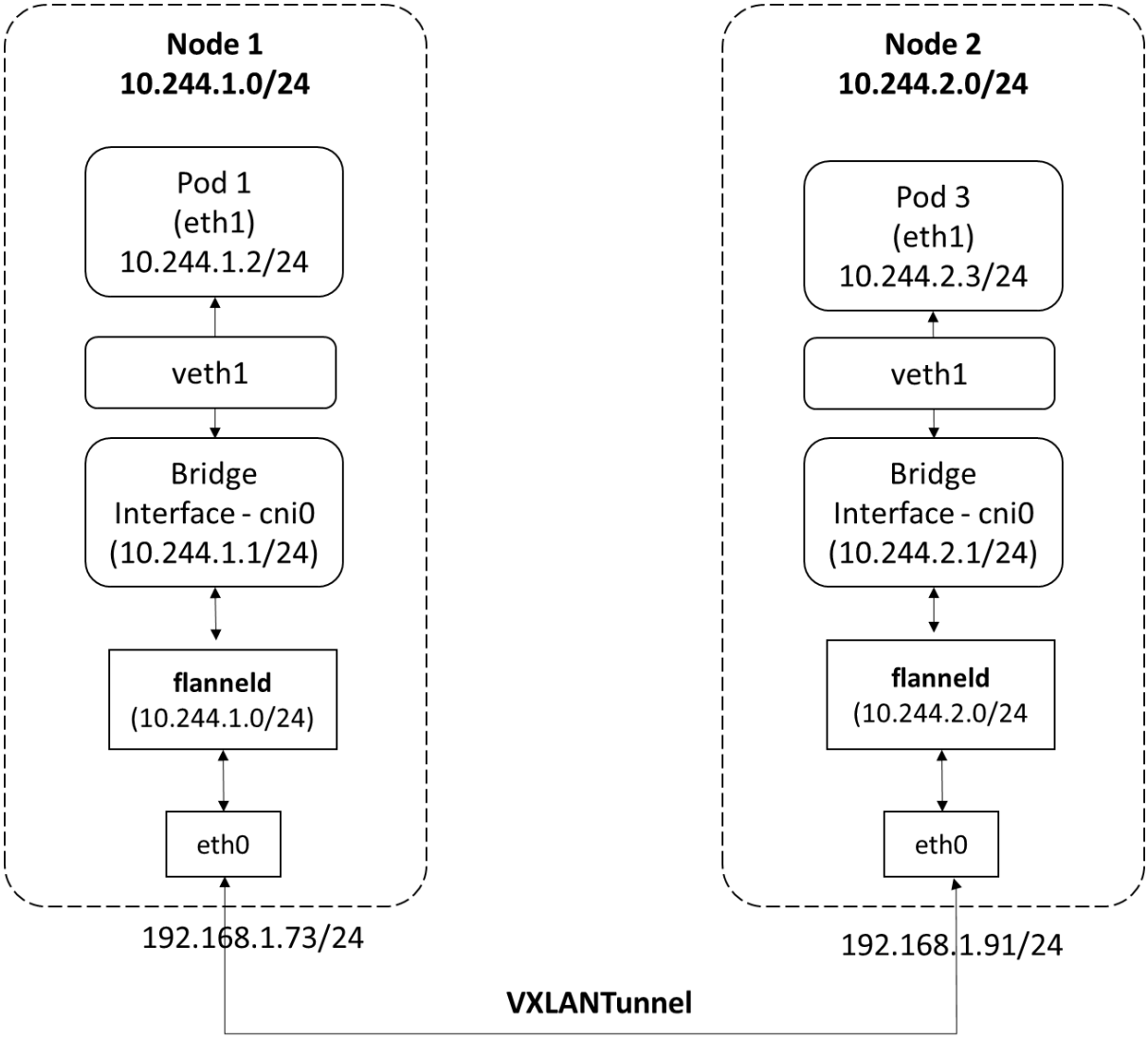
```
ubuntu@microk8s:~$microk8s cilium policy get
[
  {
    "endpointSelector": {
      "matchLabels": {
        "k8s:app": "nginx-cilium",
        "k8s:io.kubernetes.pod.namespace": "default"
      }
    },
    "ingress": [
      {
        "fromEndpoints": [
          {
            "matchLabels": {
              "k8s:io.kubernetes.pod.namespace": "default",
              "k8s:run": "access"
            }
          }
        ]
      }
    ],
    "labels": [
      {
        "key": "io.cilium.k8s.policy.derived-from",
        "value": "NetworkPolicy",
        "source": "k8s"
      },
      {
        "key": "io.cilium.k8s.policy.name",
        "value": "cilium-deny",
        "source": "k8s"
      },
      {
        "key": "io.cilium.k8s.policy.namespace",
        "value": "default",
```

```
/ # wget -q --timeout=5 nginx-cilium -O -
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
html { color-scheme: light dark; }
body { width: 35em; margin: 0 auto;
font-family: Tahoma, Verdana, Arial, sans-serif; }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>

<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>

<p><em>Thank you for using nginx.</em></p>
</body>
</html>
/ #
```

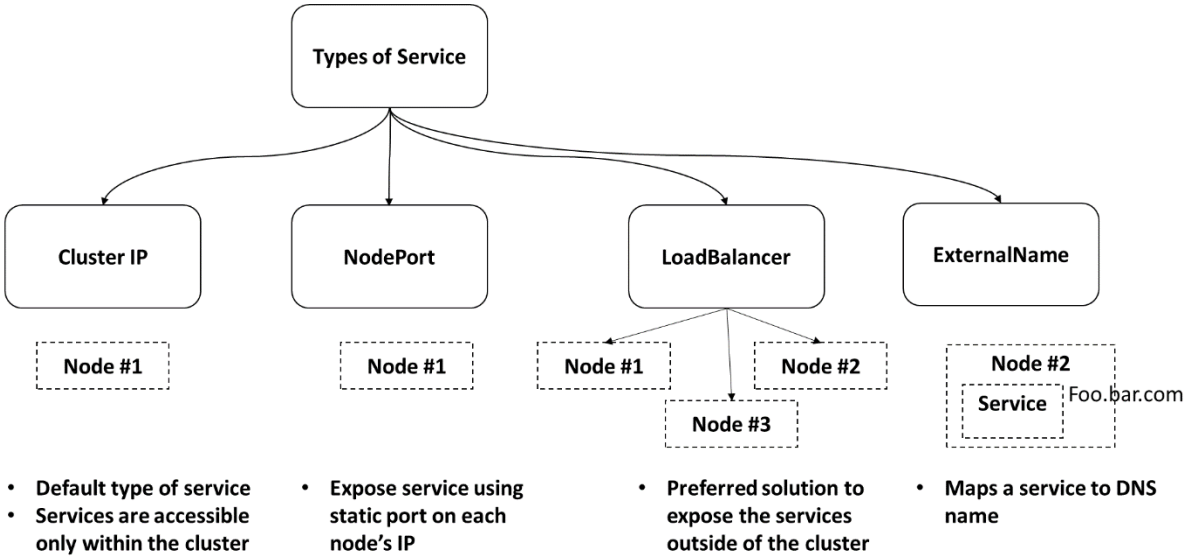
```
ubuntu@microk8s:~$microk8s disable cilium
Disabling Cilium
serviceaccount "cilium" deleted
serviceaccount "cilium-operator" deleted
configmap "cilium-config" deleted
clusterrole.rbac.authorization.k8s.io "cilium" deleted
clusterrole.rbac.authorization.k8s.io "cilium-operator" deleted
clusterrolebinding.rbac.authorization.k8s.io "cilium" deleted
clusterrolebinding.rbac.authorization.k8s.io "cilium-operator" deleted
daemonset.apps "cilium" deleted
deployment.apps "cilium-operator" deleted
12: cilium_vxlan: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1440 qdisc noqu
    link/ether 0a:e4:e4:70:61:ee brd ff:ff:ff:ff:ff:ff
Deleting old cilium_vxlan link
Restarting default cni
configmap/calico-config created
```

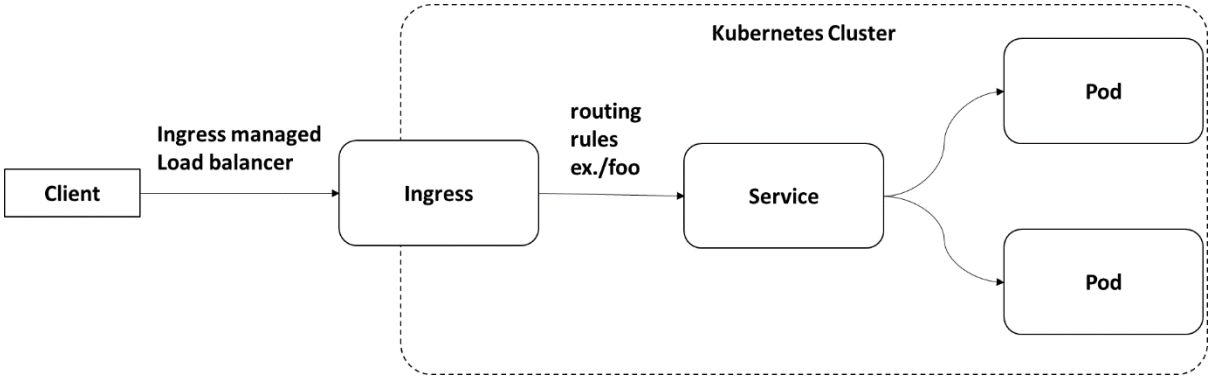
```
ubuntu@microk8s:~$kubectl get pods -o wide -n kube-system
NAME                                       READY   STATUS    RESTARTS
INESS GATES
coredns-64c6478b6c-xm78p                   1/1     Running   0
e>
calico-node-nd4pn                          1/1     Running   0
e>
calico-kube-controllers-6966456d6b-94wpt   1/1     Running   0
e>
ubuntu@microk8s:~$
```

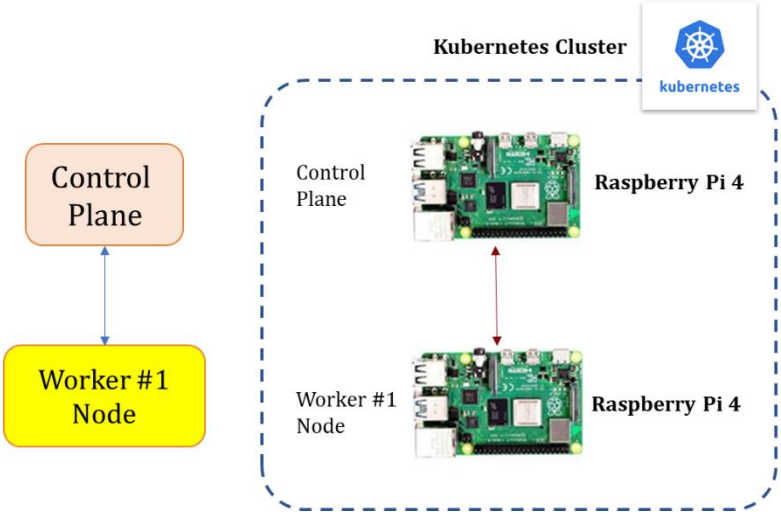
```
ubuntu@microk8s:~$microk8s disable ha-cluster
Disabling HA will reset your cluster in a clean state.
Any running workloads will be stopped and any cluster configuration will be lost.
As this is a single node cluster and this is a destructive operation,
please use the '--force' flag.
ubuntu@microk8s:~$microk8s disable ha-cluster --force
Reverting to a non-HA setup
Generating new cluster certificates.
Waiting for node to start. .
Enabling flanneld and etcd
HA disabled
ubuntu@microk8s:~$
```



Chapter 7: Setting Up MetalLB and Ingress for Load Balancing








```
ubuntu@controlplane:~$ microk8s enable metallb 192.168.1.10-192.168.1.15
Enabling MetalLB
Applying Metallb manifest
namespace/metallb-system created
secret/memberlist created
Warning: policy/v1beta1 PodSecurityPolicy is deprecated in v1.21+, unavailable in v1.25+
podsecuritypolicy.policy/controller created
podsecuritypolicy.policy/speaker created
serviceaccount/controller created
serviceaccount/speaker created
clusterrole.rbac.authorization.k8s.io/metallb-system:controller created
clusterrole.rbac.authorization.k8s.io/metallb-system:speaker created
role.rbac.authorization.k8s.io/config-watcher created
role.rbac.authorization.k8s.io/pod-lister created
clusterrolebinding.rbac.authorization.k8s.io/metallb-system:controller created
clusterrolebinding.rbac.authorization.k8s.io/metallb-system:speaker created
rolebinding.rbac.authorization.k8s.io/config-watcher created
rolebinding.rbac.authorization.k8s.io/pod-lister created
Warning: spec.template.spec.nodeSelector[beta.kubernetes.io/os]: deprecated since v1.14;
daemonset.apps/speaker created
deployment.apps/controller created
configmap/config created
MetalLB is enabled
ubuntu@controlplane:~$
```

```
ubuntu@controlplane:~$ microk8s status
microk8s is running
high-availability: no
  datastore master nodes: 127.0.0.1:19001
  datastore standby nodes: none
addons:
  enabled:
    dns                  # CoreDNS
    ha-cluster           # Configure high availability on the current node
    metallb              # Loadbalancer for your Kubernetes cluster
  disabled:
    dashboard            # The Kubernetes dashboard
    dashboard-ingress    # Ingress definition for Kubernetes dashboard
```

```
ubuntu@controlplane:~$ kubectl get all -n metallb-system
NAME                                  READY   STATUS    RESTARTS   AGE
pod/speaker-zwz98                     1/1     Running   0          3m9s
pod/controller-558b7b958-t4r78        1/1     Running   0          3m9s

NAME                     DESIRED   CURRENT   READY   UP-TO-DATE   AVAILABLE   NODE SELECTOR                   AGE
daemonset.apps/speaker   1         1         1       1            1           beta.kubernetes.io/os=linux     3m9s

NAME                            READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/controller      1/1     1            1           3m9s

NAME                                       DESIRED   CURRENT   READY   AGE
replicaset.apps/controller-558b7b958       1         1         1       3m9s
ubuntu@controlplane:~$
```

```
ubuntu@Master:~$ kubectl get nodes
NAME       STATUS   ROLES    AGE   VERSION
worker1    Ready    <none>   10d   v1.23.3-2+0d2db09fa6fbbb
master     Ready    <none>   10d   v1.23.3-2+0d2db09fa6fbbb
ubuntu@Master:~$
```

```
ubuntu@controlplane:~$ kubectl apply -f webserver-deploy.yaml
namespace/web created
deployment.apps/web-server created
service/web-server-service created
ubuntu@controlplane:~$
```

```
ubuntu@controlplane:~$ kubectl get deployments -n web
NAME          READY   UP-TO-DATE   AVAILABLE   AGE
web-server    1/1     1            1           76s
ubuntu@controlplane:~$
```

```
ubuntu@controlplane:~$ kubectl get all -n web
NAME                                  READY   STATUS    RESTARTS   AGE
pod/web-server-54c7c6444c-vn2fj       1/1     Running   0          101s

NAME                          TYPE           CLUSTER-IP       EXTERNAL-IP    PORT(S)        AGE
service/web-server-service    LoadBalancer   10.152.183.214   192.168.1.10   80:31323/TCP   100s

NAME                          READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/web-server    1/1     1            1           101s

NAME                                     DESIRED   CURRENT   READY   AGE
replicaset.apps/web-server-54c7c6444c    1         1         1       101s
ubuntu@controlplane:~$
```

```
ubuntu@controlplane:~$ curl 192.168.1.10
<html><body><h1>It works!</h1></body></html>
ubuntu@controlplane:~$
```

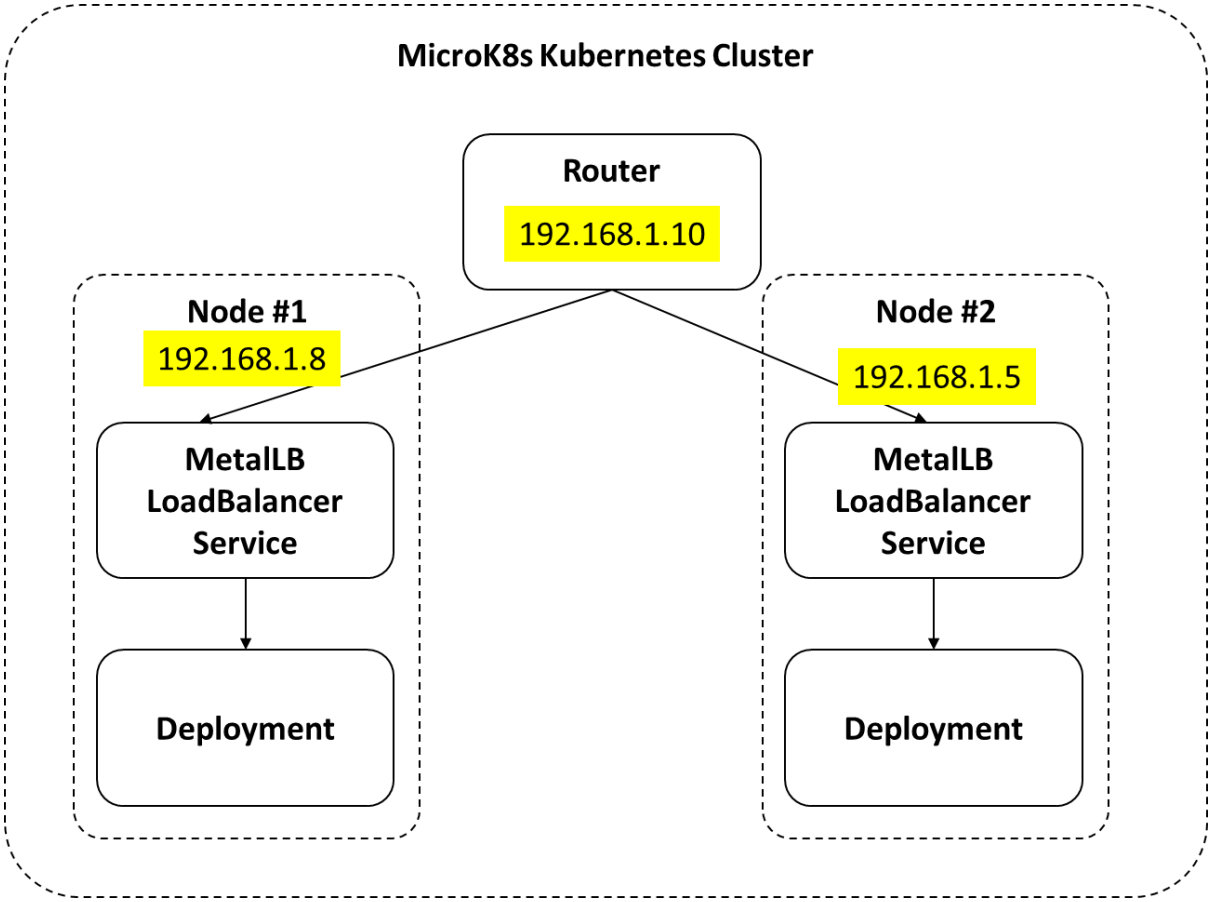
```
ubuntu@controlplane:~$ kubectl scale deployments/web-server --replicas=5 -n web
deployment.apps/web-server scaled
ubuntu@controlplane:~$
```

```
ubuntu@controlplane:~$ kubectl get deployments -n web
NAME          READY   UP-TO-DATE   AVAILABLE   AGE
web-server    5/5     5            5           3m34s
ubuntu@controlplane:~$
```

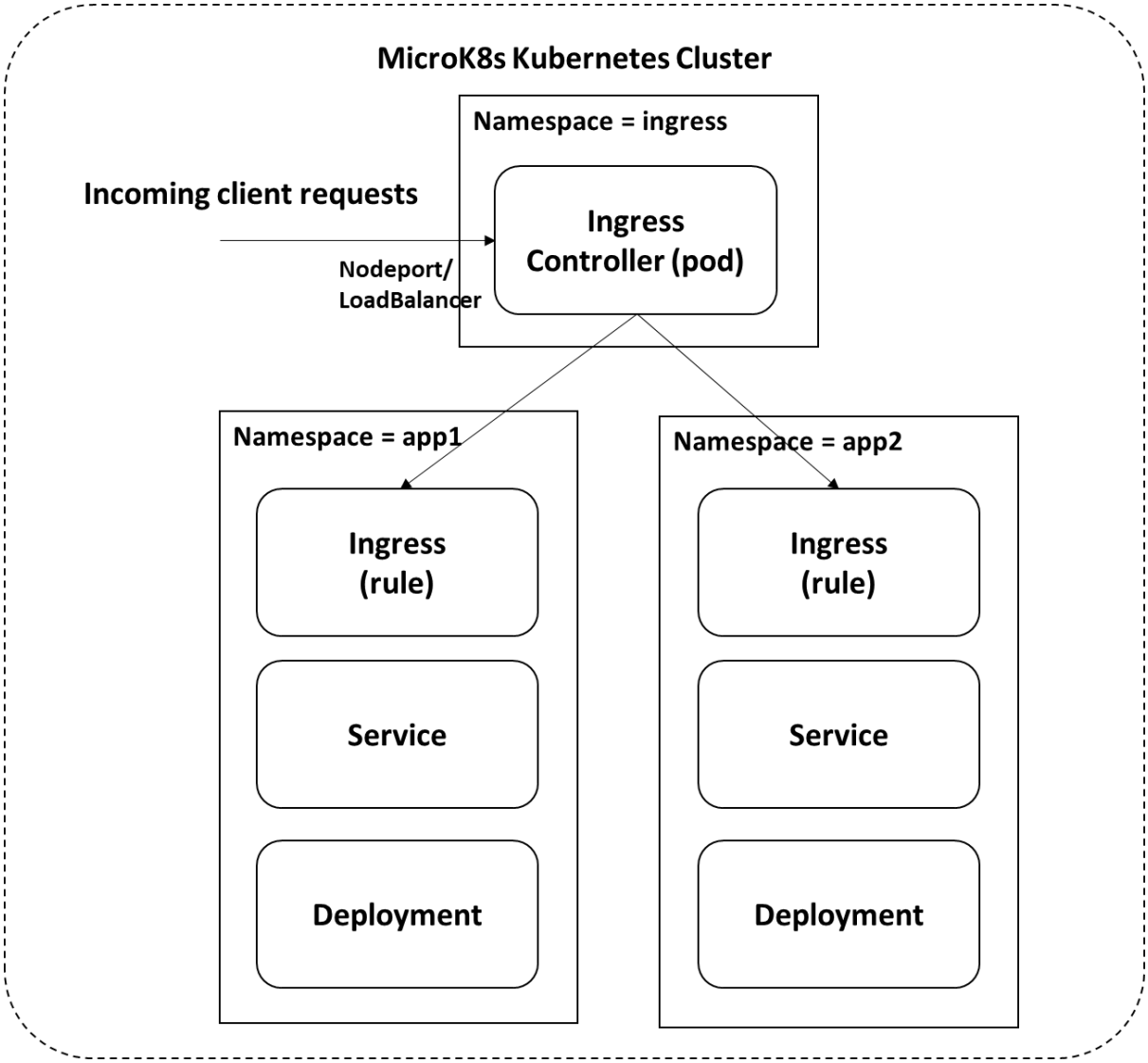
```
ubuntu@Master:~$ kubectl get pods -n web -o wide
NAME                          READY  STATUS   RESTARTS  AGE    IP            NODE     NOMINATED NODE  READINESS GATES
web-server-54c7c6444c-4q8wt   1/1    Running  0         7m46s  10.1.219.75   master   <none>          <none>
web-server-54c7c6444c-j82v4   1/1    Running  0         118s   10.1.219.76   master   <none>          <none>
web-server-54c7c6444c-56w5k   1/1    Running  0         118s   10.1.219.77   master   <none>          <none>
web-server-54c7c6444c-4p2zz   1/1    Running  0         118s   10.1.235.135  worker1  <none>          <none>
web-server-54c7c6444c-c2gkz   1/1    Running  0         118s   10.1.235.136  worker1  <none>          <none>
ubuntu@Master:~$
```

```
ubuntu@controlplane:~$ kubectl get all -n web
NAME                              READY   STATUS    RESTARTS   AGE
pod/web-server-54c7c6444c-vn2fj   1/1     Running   0          4m37s
pod/web-server-54c7c6444c-8hk4v   1/1     Running   0          93s
pod/web-server-54c7c6444c-mmzjp   1/1     Running   0          93s
pod/web-server-54c7c6444c-9vt7q   1/1     Running   0          93s
pod/web-server-54c7c6444c-rpv25   1/1     Running   0          92s

NAME                         TYPE           CLUSTER-IP       EXTERNAL-IP    PORT(S)        AGE
service/web-server-service   LoadBalancer   10.152.183.214   192.168.1.10   80:31323/TCP   4m36s

NAME                         READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/web-server   5/5     5            5           4m37s

NAME                                    DESIRED   CURRENT   READY   AGE
replicaset.apps/web-server-54c7c6444c   5         5         5       4m37s
ubuntu@controlplane:~$
```

```
ubuntu@controlplane:~$ curl 192.168.1.10
<html><body><h1>It works!</h1></body></html>
ubuntu@controlplane:~$
```




```
ubuntu@controlplane:~$ kubectl describe configmap config -n metallb-system
Name:         config
Namespace:    metallb-system
Labels:       <none>
Annotations:  <none>

Data
====
config:
----
address-pools:
- name: default
  protocol: layer2
  addresses:
  - 192.168.1.10-192.168.1.15


BinaryData
====

Events:  <none>
ubuntu@controlplane:~$
```




```
ubuntu@controlplane:~$ microk8s enable ingress
Enabling Ingress
ingressclass.networking.k8s.io/public created
namespace/ingress created
serviceaccount/nginx-ingress-microk8s-serviceaccount created
clusterrole.rbac.authorization.k8s.io/nginx-ingress-microk8s-clusterrole created
role.rbac.authorization.k8s.io/nginx-ingress-microk8s-role created
clusterrolebinding.rbac.authorization.k8s.io/nginx-ingress-microk8s created
rolebinding.rbac.authorization.k8s.io/nginx-ingress-microk8s created
configmap/nginx-load-balancer-microk8s-conf created
configmap/nginx-ingress-tcp-microk8s-conf created
configmap/nginx-ingress-udp-microk8s-conf created
daemonset.apps/nginx-ingress-microk8s-controller created
Ingress is enabled
ubuntu@controlplane:~$
```

```
ubuntu@controlplane:~$ kubectl apply -f whoami-deployment.yaml
deployment.apps/whoami-deployment created
service/whoami-service created
ingress.networking.k8s.io/whoami-ingress created
ubuntu@controlplane:~$
```

```
ubuntu@controlplane:~$ kubectl get deployments
NAME                 READY   UP-TO-DATE   AVAILABLE   AGE
whoami-deployment    1/1     1            1           69s
ubuntu@controlplane:~$
```

```
ubuntu@controlplane:~$ kubectl get ingress
NAME              CLASS    HOSTS   ADDRESS     PORTS   AGE
whoami-ingress    public   *       127.0.0.1   80      91s
ubuntu@controlplane:~$
```

```
ubuntu@Master:~$ kubectl describe ingress whoami-ingress
Name:             whoami-ingress
Labels:           <none>
Namespace:        default
Address:          127.0.0.1
Default backend:  default-http-backend:80 (<error: endpoints "default-http-backend" not found>)
Rules:
  Host        Path  Backends
  ----        ----  --------
  *
              /whoami   whoami-service:80 (10.1.219.79:80,10.1.219.80:80,10.1.235.138:80 + 2 more...)
Annotations:  <none>
Events:       <none>
ubuntu@Master:~$
```

```
ubuntu@worker1:~$ curl 192.168.1.8/whoami
Hostname: whoami-deployment-57fb67548c-mh4zp
IP: 127.0.0.1
IP: ::1
IP: 10.1.235.138
IP: fe80::1c3a:79ff:fe9d:bf29
RemoteAddr: 10.1.219.78:40750
GET /whoami HTTP/1.1
Host: 192.168.1.8
User-Agent: curl/7.68.0
Accept: */*
X-Forwarded-For: 192.168.1.7
X-Forwarded-Host: 192.168.1.8
X-Forwarded-Port: 80
X-Forwarded-Proto: http
X-Forwarded-Scheme: http
X-Real-Ip: 192.168.1.7
X-Request-Id: 03a64ad7607dee3bd7e28604561d62e9
X-Scheme: http
```

```
ubuntu@controlplane:~$ kubectl scale deployments/whoami-deployment --replicas=5
deployment.apps/whoami-deployment scaled
ubuntu@controlplane:~$
```

```
ubuntu@Master:~$ kubectl get pods -o wide
NAME                                 READY  STATUS   RESTARTS  AGE    IP            NODE     NOMINATED NODE  READINESS GATES
whoami-deployment-57fb67548c-mh4zp   1/1    Running  0         6m52s  10.1.235.138  worker1  <none>          <none>
whoami-deployment-57fb67548c-dnsc5   1/1    Running  0         2m43s  10.1.235.139  worker1  <none>          <none>
whoami-deployment-57fb67548c-gpp6h   1/1    Running  0         2m43s  10.1.235.140  worker1  <none>          <none>
whoami-deployment-57fb67548c-lsk6q   1/1    Running  0         2m43s  10.1.219.79   master   <none>          <none>
whoami-deployment-57fb67548c-zndn7   1/1    Running  0         2m43s  10.1.219.80   master   <none>          <none>
ubuntu@Master:~$
```

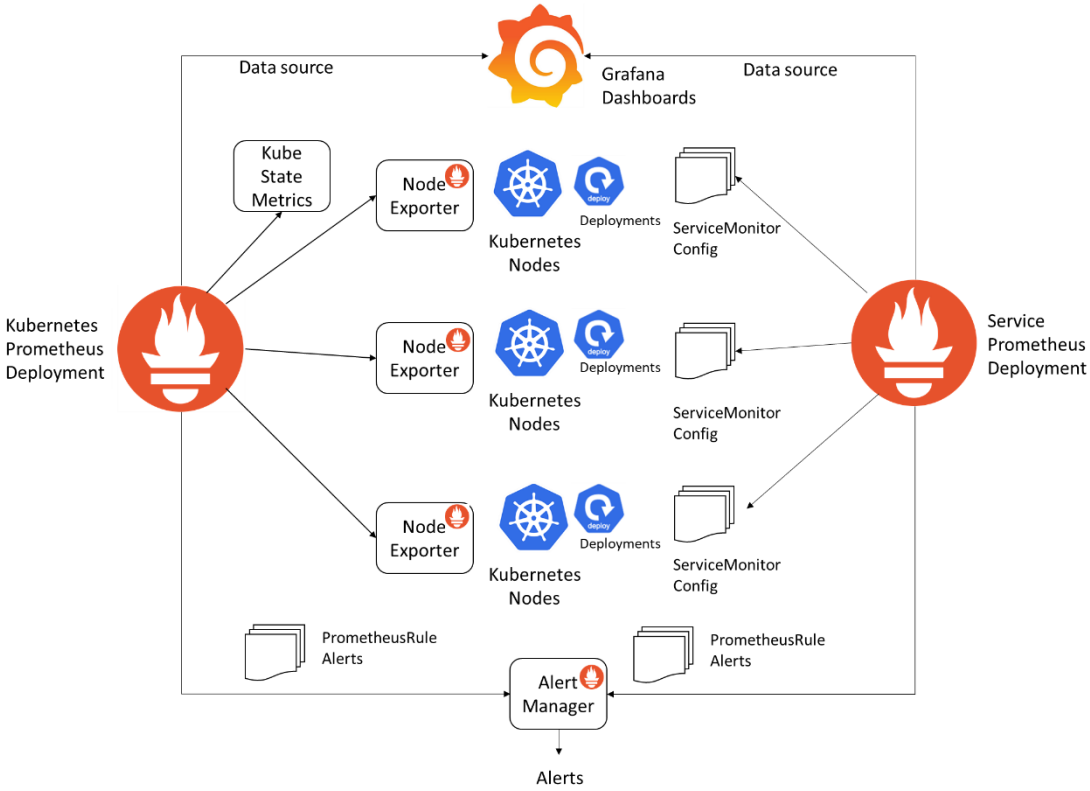
```
ubuntu@worker1:~$ curl 192.168.1.8/whoami
Hostname: whoami-deployment-57fb67548c-lsk6q
IP: 127.0.0.1
IP: ::1
IP: 10.1.219.79
IP: fe80::1481:7ff:fe9b:1578
RemoteAddr: 10.1.219.78:51910
GET /whoami HTTP/1.1
Host: 192.168.1.8
User-Agent: curl/7.68.0
Accept: */*
X-Forwarded-For: 192.168.1.7
X-Forwarded-Host: 192.168.1.8
X-Forwarded-Port: 80
X-Forwarded-Proto: http
X-Forwarded-Scheme: http
X-Real-Ip: 192.168.1.7
X-Request-Id: 7aee251014f64f663bf54c5af19b5b43
X-Scheme: http
```

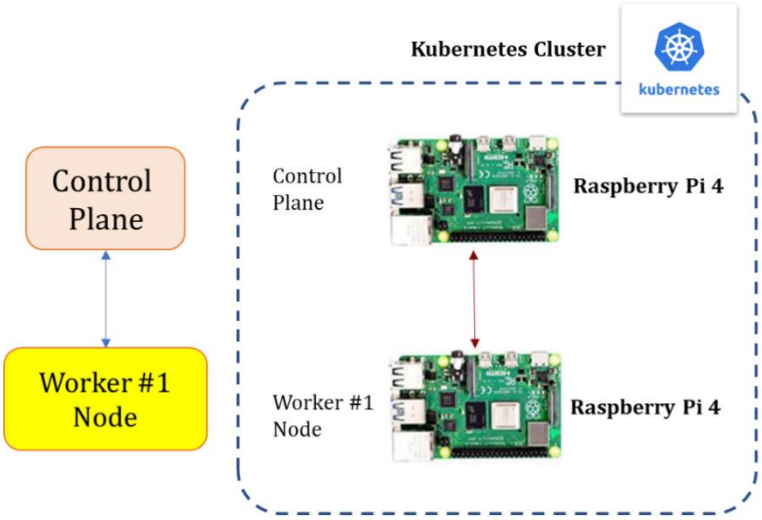
```
ubuntu@controlplane:~$ kubectl apply -f loadbalancer.yaml
service/metallb-load-balancer created
ubuntu@controlplane:~$
```

```
ubuntu@controlplane:~$ kubectl get svc
NAME                  TYPE           CLUSTER-IP       EXTERNAL-IP    PORT(S)        AGE
kubernetes            ClusterIP      10.152.183.1     <none>         443/TCP        31d
whoami-service        ClusterIP      10.152.183.28    <none>         80/TCP         6m18s
metallb-load-balancer LoadBalancer   10.152.183.213   192.168.1.11   80:31633/TCP   118s
ubuntu@controlplane:~$
```

```
ubuntu@controlplane:~$ curl 192.168.1.11/whoami
Hostname: whoami-deployment-57fb67548c-skpll
IP: 127.0.0.1
IP: ::1
IP: 10.1.49.98
IP: fe80::7cef:2cff:fe4c:56dc
RemoteAddr: 192.168.1.6:61445
GET /whoami HTTP/1.1
Host: 192.168.1.11
User-Agent: curl/7.68.0
Accept: */*

ubuntu@controlplane:~$
```

Chapter 8: Monitoring the Health of Infrastructure and Applications






```
ubuntu@controlplane:~$ microk8s enable dashboard prometheus
Enabling Kubernetes Dashboard
Enabling Metrics-Server
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-delegato
r created
clusterrolebinding.rbac.authorization.k8s.io/system:metrics-server created
service/metrics-server created
deployment.apps/metrics-server created
apiservice.apiregistration.k8s.io/v1beta1.metrics.k8s.io created
clusterrolebinding.rbac.authorization.k8s.io/microk8s-admin created
Metrics-Server is enabled
Applying manifest
```

```
prometheus.monitoring.coreos.com/k8s created
prometheusrule.monitoring.coreos.com/prometheus-k8s-prometheus-rules created
rolebinding.rbac.authorization.k8s.io/prometheus-k8s-config created
rolebinding.rbac.authorization.k8s.io/prometheus-k8s created
rolebinding.rbac.authorization.k8s.io/prometheus-k8s created
rolebinding.rbac.authorization.k8s.io/prometheus-k8s created
role.rbac.authorization.k8s.io/prometheus-k8s-config created
role.rbac.authorization.k8s.io/prometheus-k8s created
role.rbac.authorization.k8s.io/prometheus-k8s created
role.rbac.authorization.k8s.io/prometheus-k8s created
service/prometheus-k8s created
serviceaccount/prometheus-k8s created
servicemonitor.monitoring.coreos.com/prometheus-k8s created
The Prometheus operator is enabled (user/pass: admin/admin)
ubuntu@controlplane:~$
```

```
ubuntu@controlplane:~$ kubectl apply -f dashboard-adminuser.yaml
serviceaccount/admin-user created
clusterrolebinding.rbac.authorization.k8s.io/admin-user created
ubuntu@controlplane:~$
```

```
ubuntu@controlplane:~$ kubectl proxy &
[1] 61742
ubuntu@controlplane:~$ Starting to serve on 127.0.0.1:8001
```

kubernetes

Search

+ CREATE

Cluster

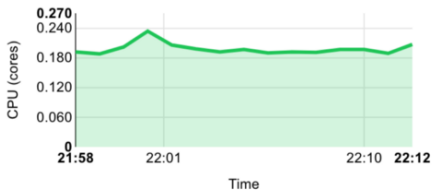
Cluster
Namespaces
Nodes
Persistent Volumes
Roles
Storage Classes

Namespace
default

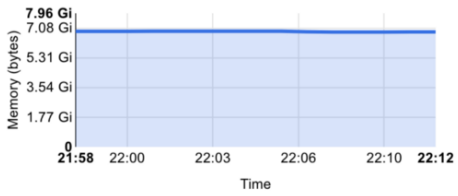
Overview

Workloads
Cron Jobs
Daemon Sets
Deployments
Jobs

CPU usage



Memory usage



Namespaces

Name	Labels	Status	Age
default	-	Active	20 days
kube-system	-	Active	20 days
kube-public	-	Active	20 days

kubernetes

Search

+ CREATE

Workloads > Pods

Cluster
Namespaces
Nodes
Persistent Volumes
Roles
Storage Classes

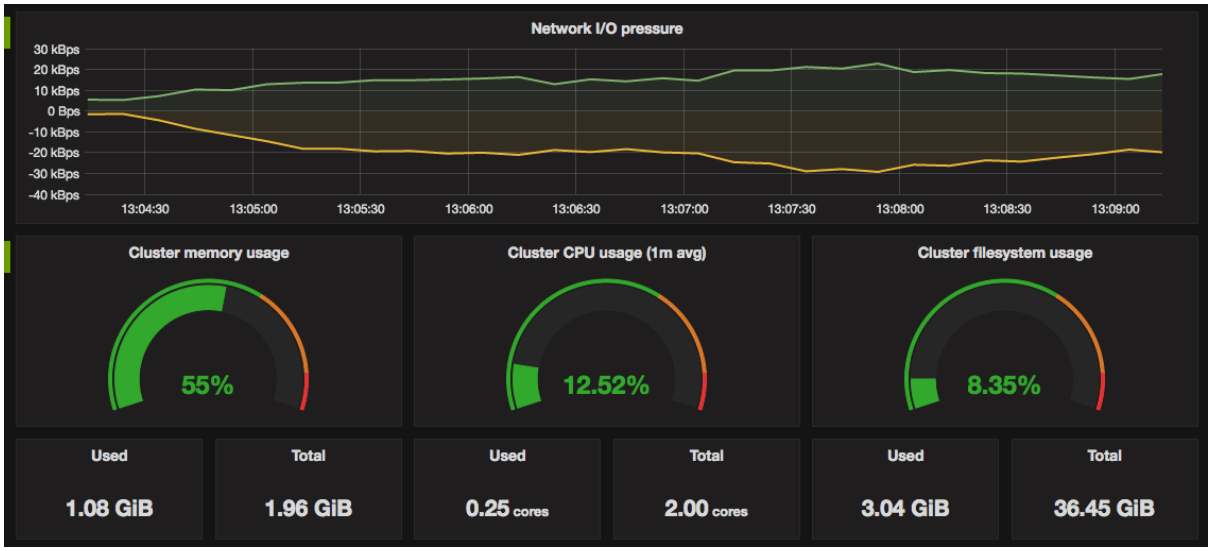
Namespace
kube-system

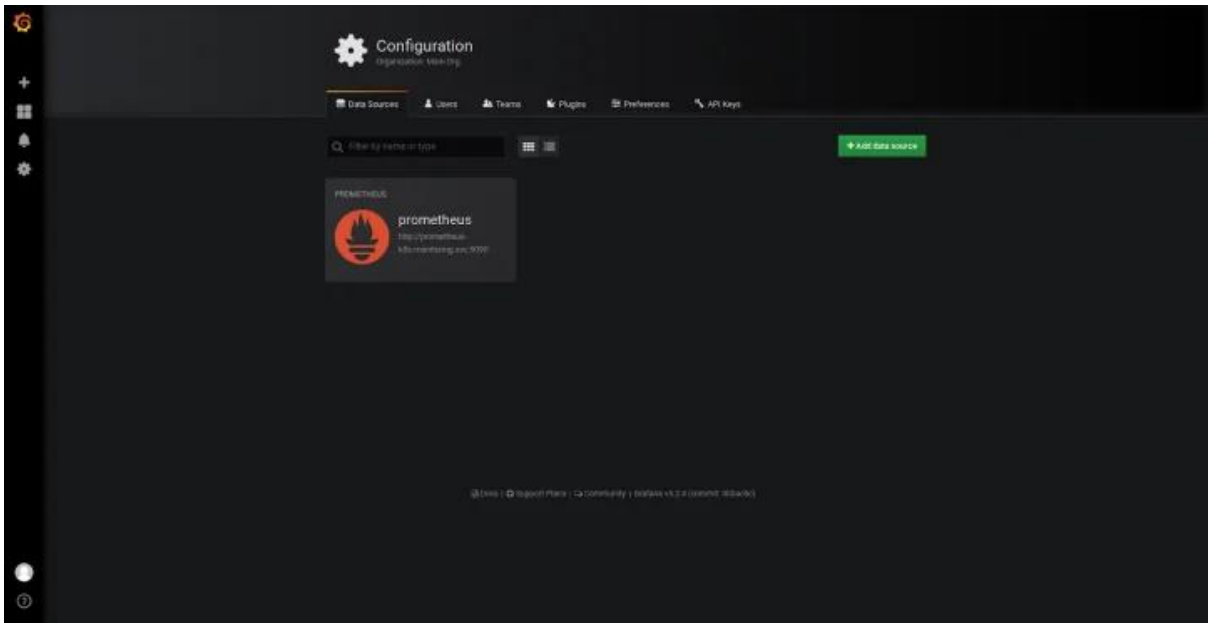
Workloads
Daemon Sets
Deployments
Jobs
Pods
Replica Sets
Replication Controllers

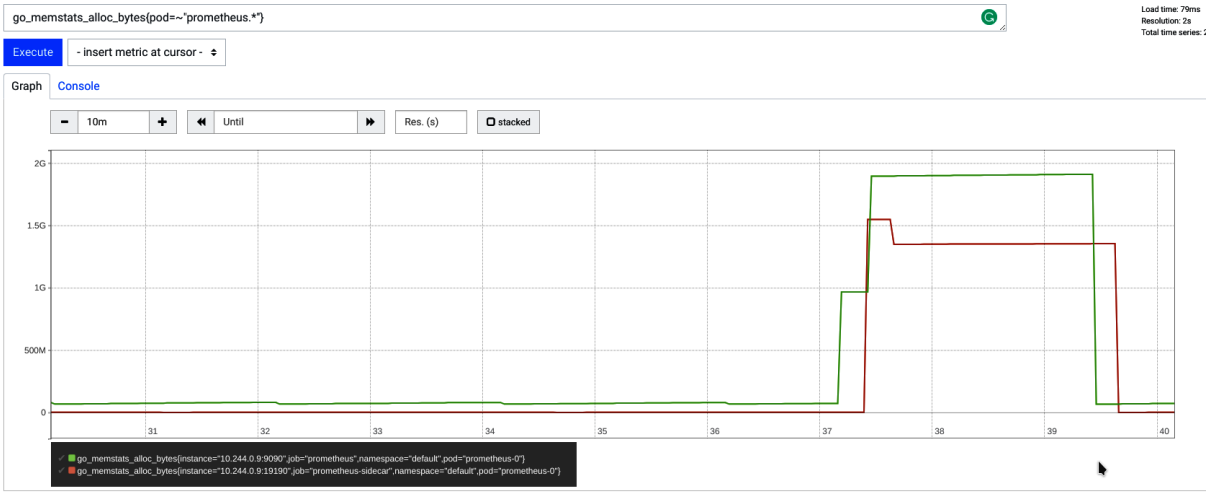
Pods

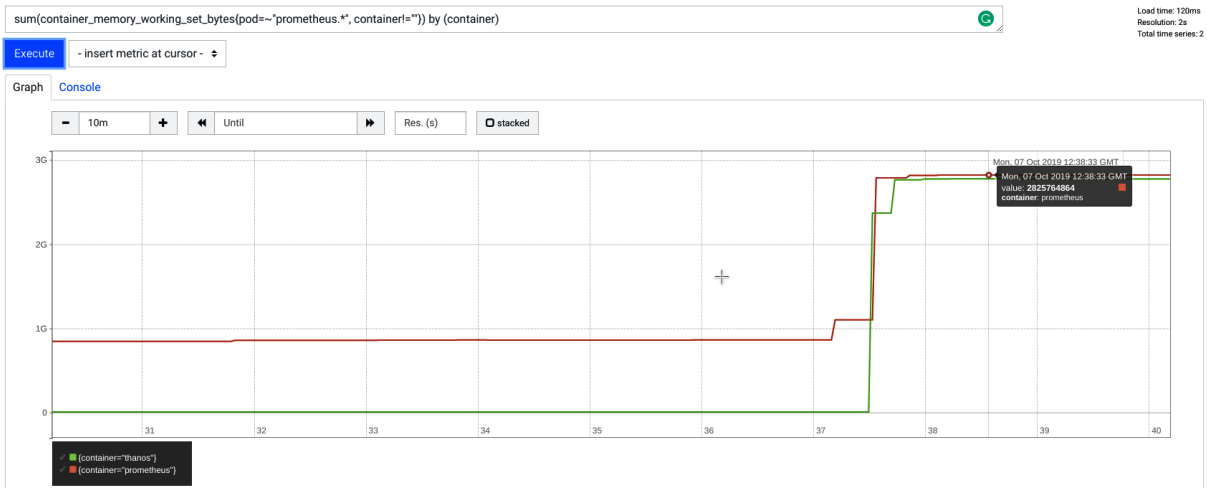
Name	Status	Restarts	Age
kubernetes-dashboard-2039414953-pplhm	Running	0	3 minutes
kube-proxy-ip-172-20-48-146.ec2.internal	Running	0	8 minutes
kube-proxy-ip-172-20-56-251.ec2.internal	Running	0	10 minutes
kube-proxy-ip-172-20-32-47.ec2.internal	Running	0	13 minutes
kube-dns-1321724180-sr5z1	Running	0	16 minutes
kube-dns-1321724180-894r4	Running	0	16 minutes
kube-dns-autoscaler-265231812-v9mls	Running	0	16 minutes
kube-proxy-ip-172-20-49-11.ec2.internal	Running	0	17 minutes
kube-scheduler-ip-172-20-37-69.ec2.internal	Running	0	19 minutes
etcd-server-ip-172-20-37-69.ec2.internal	Running	0	19 minutes

1 - 10 of 15


Network I/O pressure
30 kBps
20 kBps
10 kBps
0 Bps
-10 kBps
-20 kBps
-30 kBps
-40 kBps
13:04:30
13:05:00
13:05:30
13:06:00
13:06:30
13:07:00
13:07:30
13:08:00
13:08:30
13:09:00
Cluster memory usage
55%
Used
1.08 GiB
Total
1.96 GiB
Cluster CPU usage (1m avg)
12.52%
Used
0.25 cores
Total
2.00 cores
Cluster filesystem usage
8.35%
Used
3.04 GiB
Total
36.45 GiB


Configuration
Data Sources
Users
Teams
Plugins
Preferences
API Keys
+ Add data source
PROMETHEUS
prometheus


go_memstats_alloc_bytes{pod=~"prometheus.*"}
Load time: 79ms
Resolution: 2s
Execute
- insert metric at cursor -
Graph
Console
10m
Until
Res. (s)
stacked
2G
1.5G
1G
500M
0
31
32
33
34
35
36
37
38
39
40
go_memstats_alloc_bytes{instance="10.244.0.9:9090",job="prometheus",namespace="default",pod="prometheus-0"}
go_memstats_alloc_bytes{instance="10.244.0.9:19190",job="prometheus-sidecar",namespace="default",pod="prometheus-0"}



Targets

All | Unhealthy

kubernetes-apiservers (1/1 up) show less

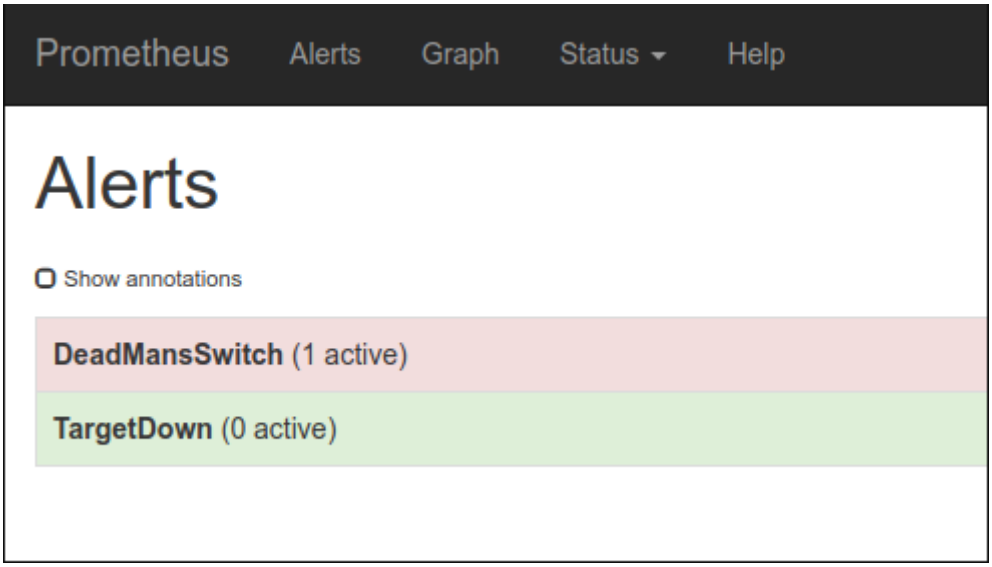
Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
https://192.168.99.100:8443/metrics	UP	instance="192.168.99.100:8443" job="kubernetes-apiservers"	10.085s ago	104.9ms	

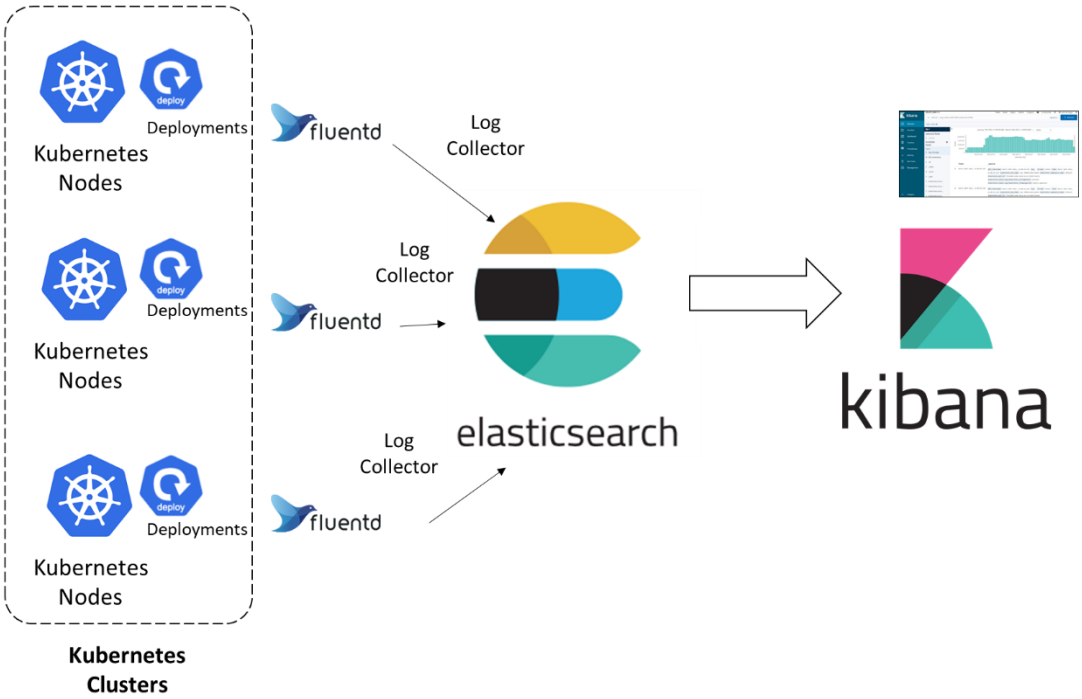
kubernetes-nodes (1/1 up) show less

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
https://kubernetes.default.svc:443/api/v1/nodes/minikube/proxy/metrics	UP	beta_kubernetes_io_arch="amd64" beta_kubernetes_io_os="linux" dedicated="cortex" dedicated_memcached="cortex-memcached" instance="minikube" job="kubernetes-nodes" kubernetes_io_arch="amd64" kubernetes_io_hostname="minikube" kubernetes_io_os="linux" node_role_kubernetes_io_master=""	27.384s ago	65.1ms	

kubernetes-nodes-cadvisor (1/1 up) show less

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
https://kubernetes.default.svc:443/api/v1/nodes/minikube/proxy/metrics/cadvisor	UP	beta_kubernetes_io_arch="amd64" beta_kubernetes_io_os="linux" dedicated="cortex" dedicated_memcached="cortex-memcached" instance="minikube" job="kubernetes-nodes-cadvisor" kubernetes_io_arch="amd64" kubernetes_io_hostname="minikube" kubernetes_io_os="linux" node_role_kubernetes_io_master=""	46.692s ago	141.4ms	


Prometheus
Alerts
Graph
Status
Help
Alerts
Show annotations
DeadMansSwitch (1 active)
TargetDown (0 active)

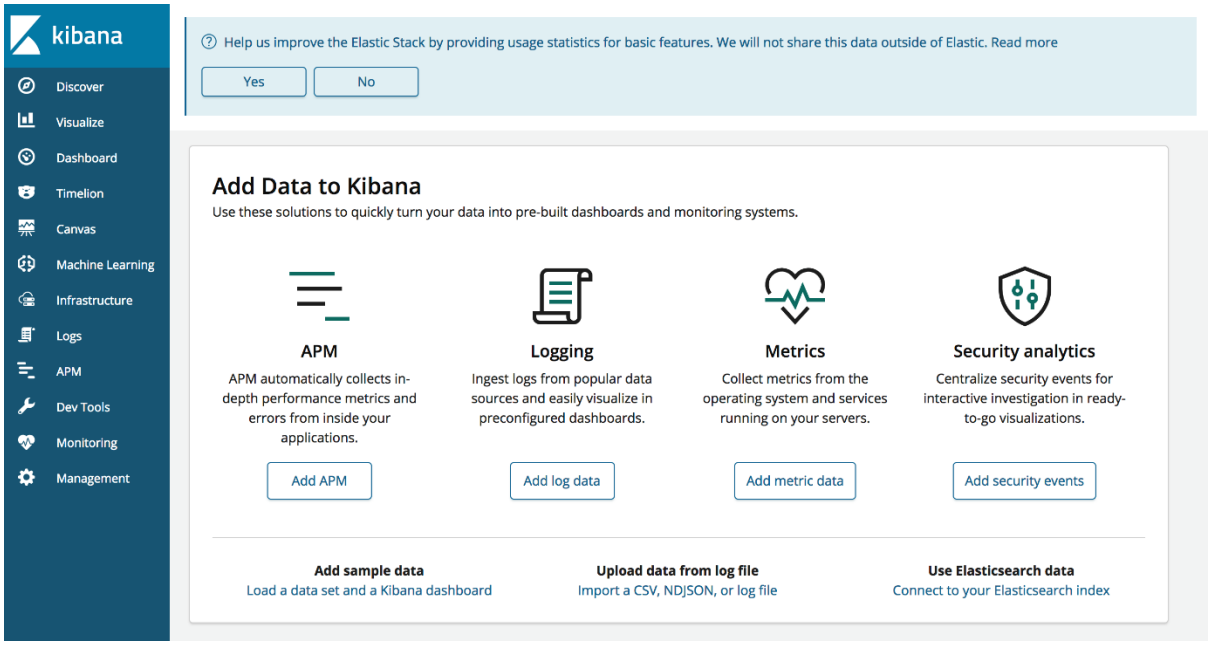

Deployments
Kubernetes Nodes
Deployments
Kubernetes Nodes
Deployments
Kubernetes Nodes
Kubernetes Clusters
fluentd
fluentd
fluentd
Log Collector
Log Collector
Log Collector
elasticsearch
kibana

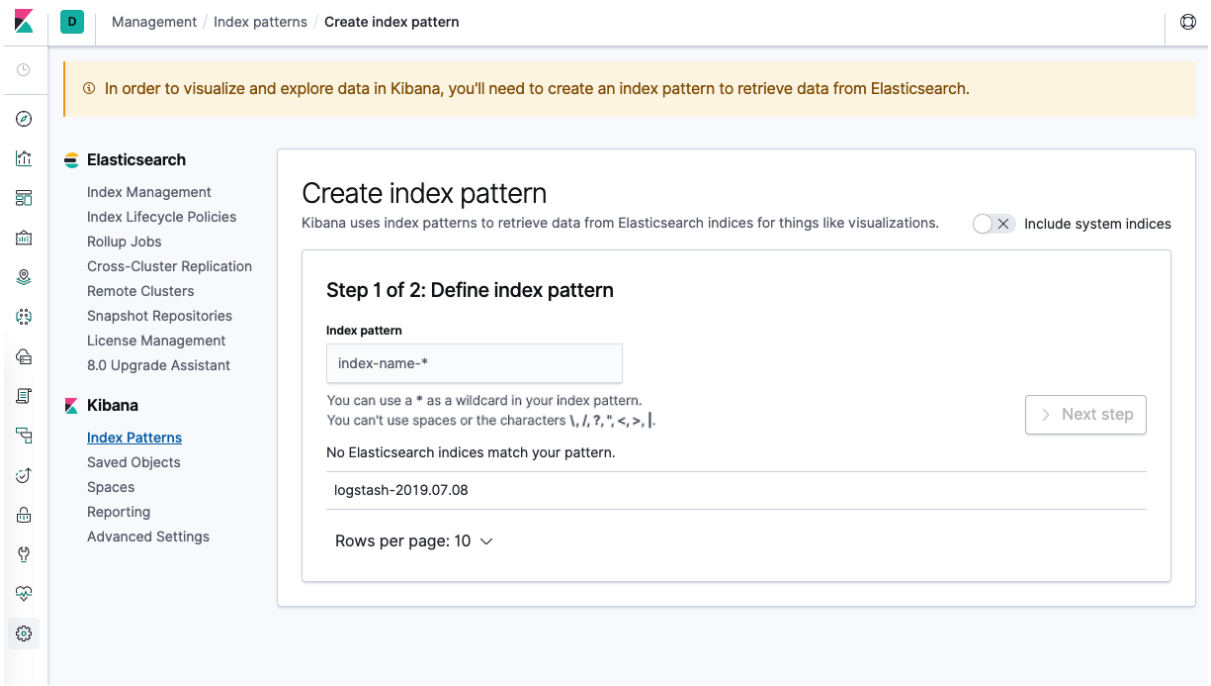
```
$ microk8s enable fluentd
Enabling Fluentd-Elasticsearch
Labeling nodes
node/host01 labeled
Addon dns is already enabled.
--allow-privileged=true
service/elasticsearch-logging created
serviceaccount/elasticsearch-logging created
clusterrole.rbac.authorization.k8s.io/elasticsearch-logging created
clusterrolebinding.rbac.authorization.k8s.io/elasticsearch-logging created
statefulset.apps/elasticsearch-logging created
configmap/fluentd-es-config-v0.2.0 created
serviceaccount/fluentd-es created
clusterrole.rbac.authorization.k8s.io/fluentd-es created
clusterrolebinding.rbac.authorization.k8s.io/fluentd-es created
daemonset.apps/fluentd-es-v3.1.0 created
deployment.apps/kibana-logging created
service/kibana-logging created
Fluentd-Elasticsearch is enabled
$
```

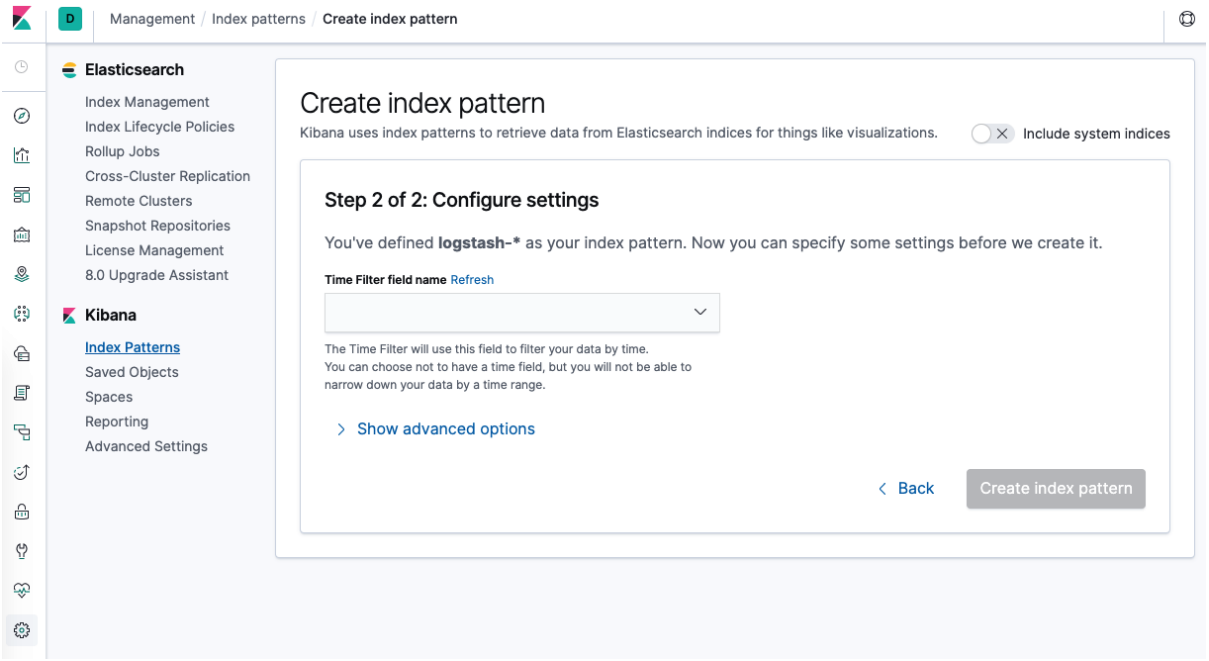
```
$ microk8s status
microk8s is running
high-availability: no
  datastore master nodes: 127.0.0.1:19001
  datastore standby nodes: none
addons:
  enabled:
    dashboard            # The Kubernetes dashboard
    dns                  # CoreDNS
    fluentd              # Elasticsearch-Fluentd-Kibana logging and monitoring
    ha-cluster           # Configure high availability on the current node
    metrics-server       # K8s Metrics Server for API access to service metrics
  disabled:
    ambassador           # Ambassador API Gateway and Ingress
```

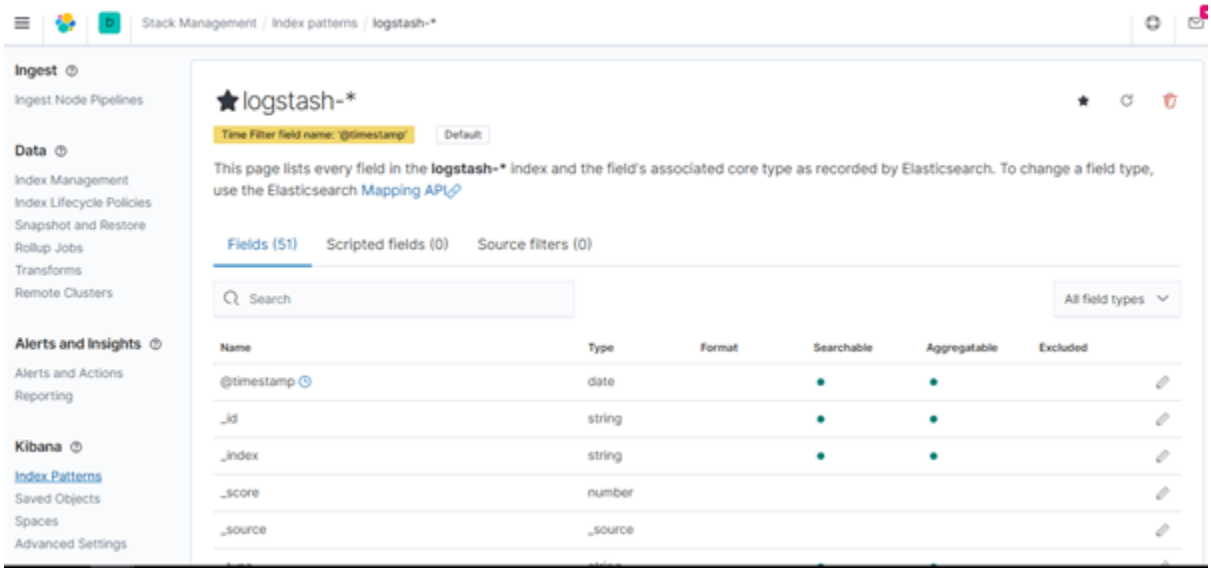
```
$ microk8s kubectl get svc -n kube-system
NAME                        TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)                  AGE
kube-dns                    ClusterIP   10.152.183.10    <none>        53/UDP,53/TCP,9153/TCP   6m8s
metrics-server              ClusterIP   10.152.183.109   <none>        443/TCP                  5m14s
kubernetes-dashboard        ClusterIP   10.152.183.129   <none>        443/TCP                  4m56s
dashboard-metrics-scraper   ClusterIP   10.152.183.151   <none>        8000/TCP                 4m55s
elasticsearch-logging       ClusterIP   None             <none>        9200/TCP,9300/TCP        2m7s
kibana-logging              ClusterIP   10.152.183.100   <none>        5601/TCP                 2m5s
$
```

```
$ microk8s kubectl port-forward -n kube-system service/kibana-logging 8181:5601
Forwarding from 127.0.0.1:8181 -> 5601
Forwarding from [::1]:8181 -> 5601
```


kibana
Discover
Visualize
Dashboard
Timelion
Canvas
Machine Learning
Infrastructure
Logs
APM
Dev Tools
Monitoring
Management
Help us improve the Elastic Stack by providing usage statistics for basic features. We will not share this data outside of Elastic. Read more
Yes
No
Add Data to Kibana
Use these solutions to quickly turn your data into pre-built dashboards and monitoring systems.
APM
APM automatically collects in-depth performance metrics and errors from inside your applications.
Add APM
Logging
Ingest logs from popular data sources and easily visualize in preconfigured dashboards.
Add log data
Metrics
Collect metrics from the operating system and services running on your servers.
Add metric data
Security analytics
Centralize security events for interactive investigation in ready-to-go visualizations.
Add security events
Add sample data
Load a data set and a Kibana dashboard
Upload data from log file
Import a CSV, NDJSON, or log file
Use Elasticsearch data
Connect to your Elasticsearch index


D
Management / Index patterns / Create index pattern
In order to visualize and explore data in Kibana, you'll need to create an index pattern to retrieve data from Elasticsearch.
Elasticsearch
Index Management
Index Lifecycle Policies
Rollup Jobs
Cross-Cluster Replication
Remote Clusters
Snapshot Repositories
License Management
8.0 Upgrade Assistant
Kibana
Index Patterns
Saved Objects
Spaces
Reporting
Advanced Settings
Create index pattern
Kibana uses index patterns to retrieve data from Elasticsearch indices for things like visualizations.
Include system indices
Step 1 of 2: Define index pattern
Index pattern
index-name-*
You can use a * as a wildcard in your index pattern.
You can't use spaces or the characters \, /, ?, ", <, >, |.
Next step
No Elasticsearch indices match your pattern.
logstash-2019.07.08
Rows per page: 10


Management / Index patterns / Create index pattern
Elasticsearch
Index Management
Index Lifecycle Policies
Rollup Jobs
Cross-Cluster Replication
Remote Clusters
Snapshot Repositories
License Management
8.0 Upgrade Assistant
Kibana
Index Patterns
Saved Objects
Spaces
Reporting
Advanced Settings
Create index pattern
Kibana uses index patterns to retrieve data from Elasticsearch indices for things like visualizations.
Include system indices
Step 2 of 2: Configure settings
You've defined logstash-* as your index pattern. Now you can specify some settings before we create it.
Time Filter field name Refresh
The Time Filter will use this field to filter your data by time.
You can choose not to have a time field, but you will not be able to narrow down your data by a time range.
Show advanced options
Back
Create index pattern


Stack Management / Index patterns / logstash-*
Ingest
Ingest Node Pipelines
Data
Index Management
Index Lifecycle Policies
Snapshot and Restore
Rollup Jobs
Transforms
Remote Clusters
Alerts and Insights
Alerts and Actions
Reporting
Kibana
Index Patterns
Saved Objects
Spaces
Advanced Settings
logstash-*
Time Filter field name: '@timestamp'
Default
This page lists every field in the logstash-* index and the field's associated core type as recorded by Elasticsearch. To change a field type, use the Elasticsearch Mapping API
Fields (51)
Scripted fields (0)
Source filters (0)
Search
All field types
Name
Type
Format
Searchable
Aggregatable
Excluded
@timestamp
date
_id
string
_index
string
_score
number
_source
_source

Discover

1,326 hits

New Save Open Share Inspect

Filters Search KQL Last 15 minutes Show dates Refresh

+ Add filter

logstash-*

Selected fields

? _source

Available fields

@timestamp

t _id

t _index

_score

t _type

t docker.container_id

t kubernetes.container_im...

t kubernetes.container_im...

t kubernetes.container_na...

t kubernetes.host

t kubernetes.labels.app

t kubernetes.labels.control...

t kubernetes.labels.k8s-app

t kubernetes.labels.kubern...

t kubernetes.labels.pod-te...

t kubernetes.labels.pod-te...

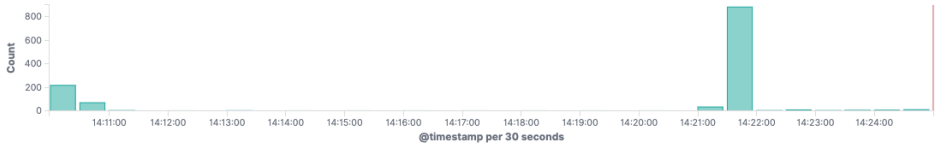
t kubernetes.labels.statefu...

t kubernetes.master_url

t kubernetes.namespace_id

t kubernetes.namespace_...

Jul 8, 2019 @ 14:09:59.102 - Jul 8, 2019 @ 14:24:59.102 — Auto



Time	_source
Jul 8, 2019 @ 14:24:52.143	log: {"type":"response","@timestamp":"2019-07-08T18:24:52Z","tags":[],"pid":1,"method":"get","statusCode":200,"req":{"url":"/bundles/4b5a84aaf1c9485e060c503a0ff8cadb.woff2","method":"get","headers":{"host":"localhost:5601","connection":"keep-alive","origin":"http://localhost:5601","user-agent":"Mozilla/5.0 (Macintosh; Intel Mac OS X 10_14_5) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/75.0.3770.100 Safari/537.36","accept":"*/*","referer":"http://localhost:5601/bundles/commons.style.css","accept-encoding":"gzip,
Jul 8, 2019 @ 14:24:51.779	log: {"type":"response","@timestamp":"2019-07-08T18:24:51Z","tags":[],"pid":1,"method":"put","statusCode":200,"req":{"url":"/api/saved_objects/index-pattern/ab5e9f30-a1ad-11e9-a730-792a99dd4f03","method":"put","headers":{"host":"localhost:5601","connection":"keep-alive","content-length":"11710","origin":"http://localhost:5601","kbn-version":"7.2.0","user-agent":"Mozilla/5.0 (Macintosh; Intel Mac OS X 10_14_5) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/75.0.3770.100 Safari/537.36","content-
Jul 8, 2019 @ 14:24:51.221	log: {"type":"response","@timestamp":"2019-07-08T18:24:51Z","tags":[],"pid":1,"method":"get","statusCode":200,"req":{"url":"/api/index_patterns/_fields_for_wildcard?pattern=logstash-*&meta_fields=_source&meta_fields=_id&meta_fields=_type&meta_fields=_index&meta_fields=_score","method":"get","headers":{"host":"localhost:5601","connection":"keep-alive","accept":"application/json, text/plain, */*","kbn-version":"7.2.0","user-agent":"Mozilla/5.0 (Macintosh; Intel Mac OS X 10_14_5) AppleWebKit/537.36 (KHTML, like Gecko)
Jul 8, 2019 @ 14:24:51.061	log: {"type":"response","@timestamp":"2019-07-08T18:24:51Z","tags":[],"pid":1,"method":"post","statusCode":200,"req":{"url":"/api/saved_objects/_bulk_get","method":"post","headers":{"host":"localhost:5601","connection":"keep-alive","content-length":"70","origin":"http://localhost:5601","kbn-version":"7.2.0","user-agent":"Mozilla/5.0 (Macintosh; Intel Mac OS X 10_14_5) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/75.0.3770.100 Safari/537.36","content-type":"application/json","accept":"*/*","referer":"http://localhost:5601/app/kibana","accept-encoding":"gzip, deflate,

Discover

64 hits

New Save Open Share Inspect

Filters kubernetes.pod_name:counter KQL ~ 15 minutes ago → now Refresh

+ Add filter

logstash-*

Selected fields

? _source

Available fields

@timestamp

t _id

t _index

_score

t _type

t docker.container_id

t kubernetes.container_im...

t kubernetes.container_im...

t kubernetes.container_na...

t kubernetes.host

t kubernetes.labels.app

t kubernetes.labels.control...

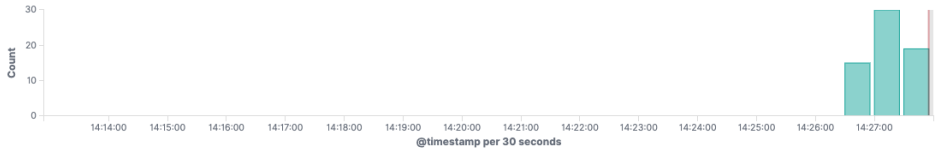
t kubernetes.labels.io_ciliu...

t kubernetes.labels.k8s-app

t kubernetes.labels.kubern...

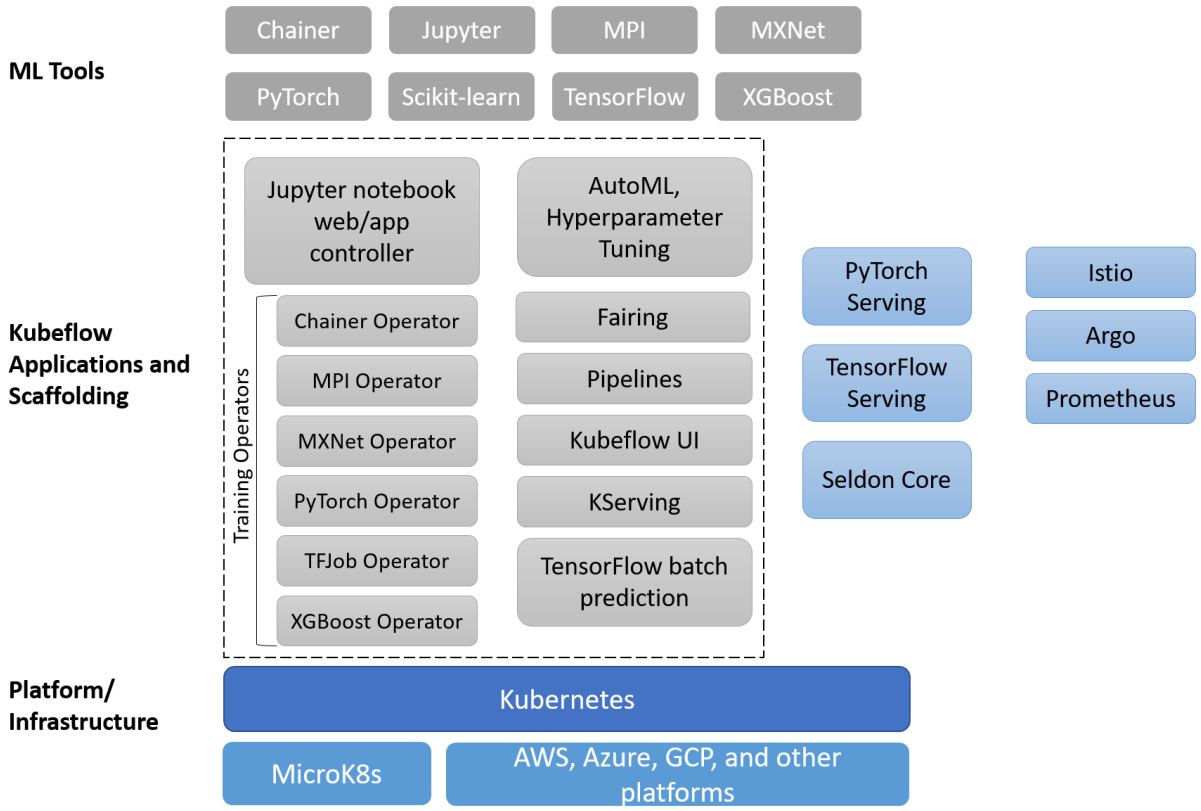
t kubernetes.labels.name

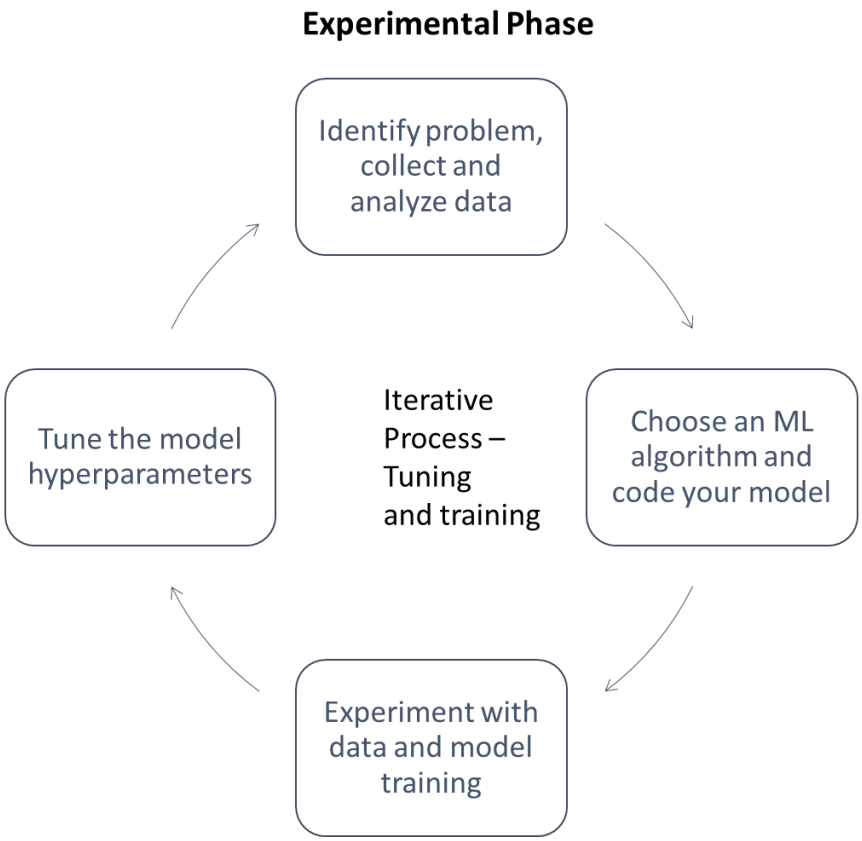
Jul 8, 2019 @ 14:12:54.384 - Jul 8, 2019 @ 14:27:54.384 — Auto

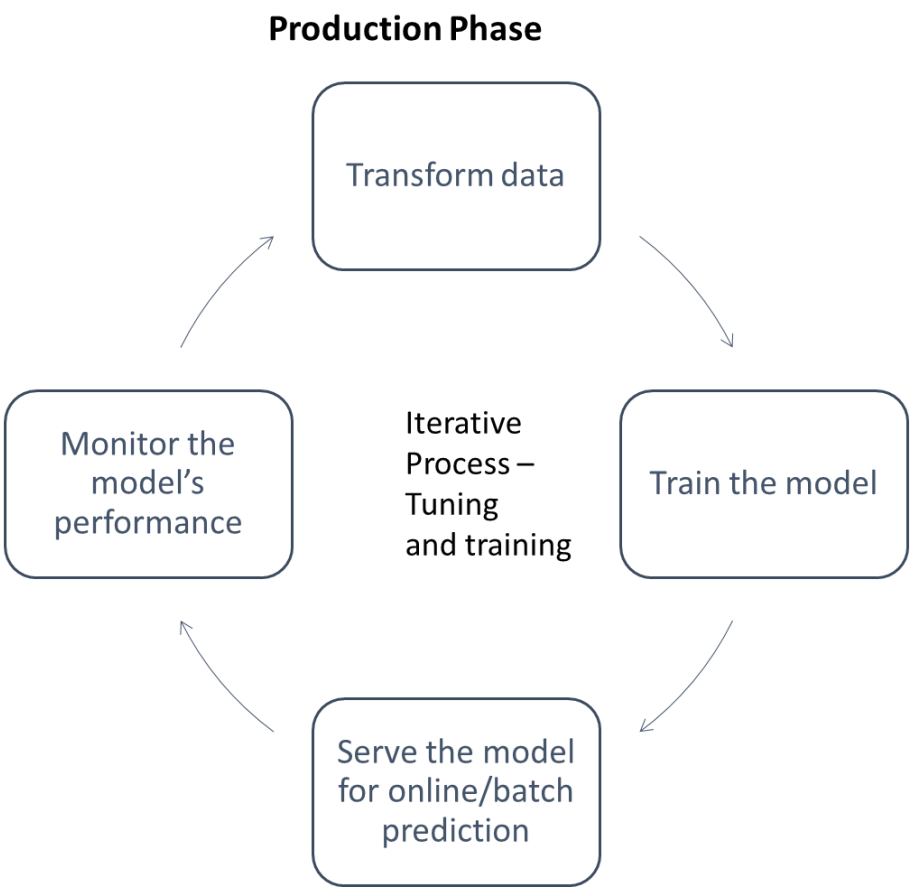


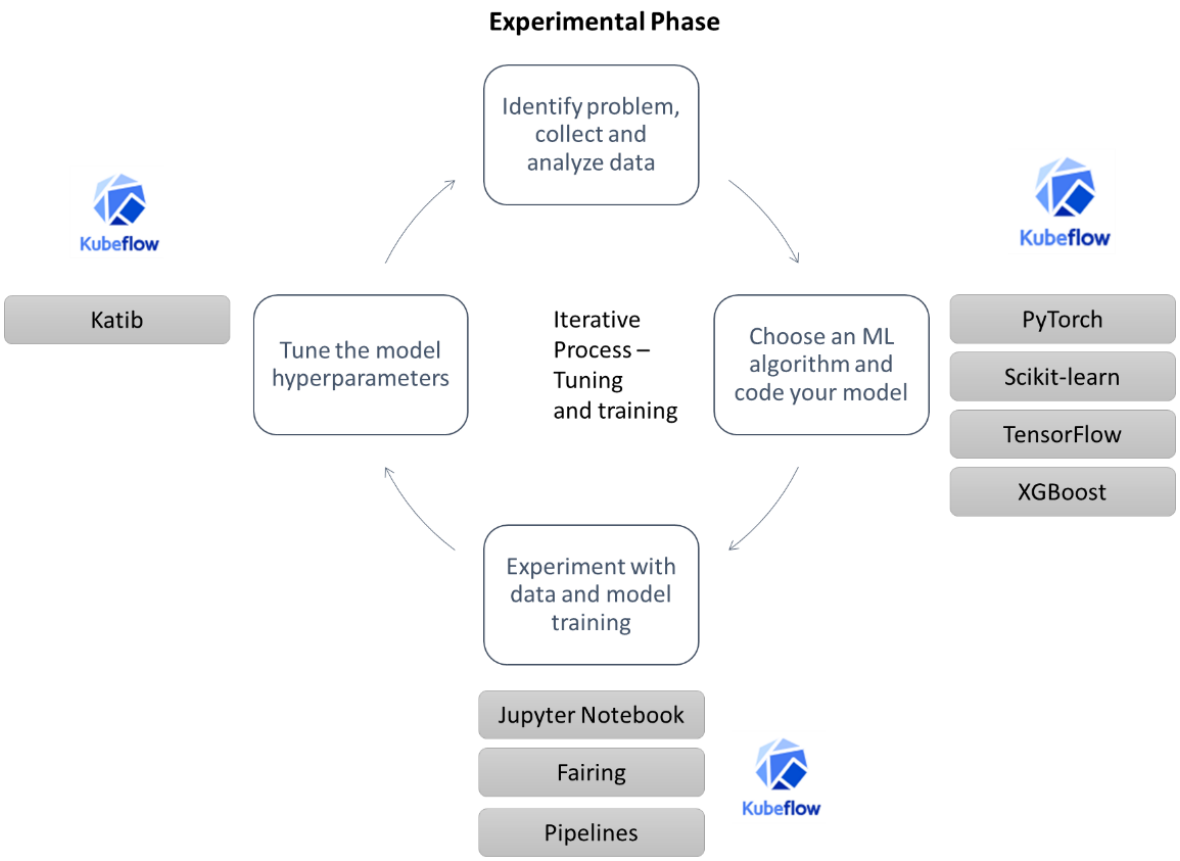
Time	_source
Jul 8, 2019 @ 14:27:48.897	kubernetes.pod_name: counter log: 63: Mon Jul 8 18:27:48 UTC 2019 stream: stdout docker.container_id: 98beea3f30e5fa9939f91f8bf1eb7cccdef5be4c90dc96b7167865db630bb9e9 kubernetes.container_name: count kubernetes.namespace_name: default kubernetes.container_image: busybox:latest kubernetes.container_image_id: docker-pullable://busybox@sha256:c94cf1b87ccb80f2e6414ef913c748b105060debda482058d2b8d0fce39f11b9 kubernetes.pod_id: efd1ee59-a1ad-11e9-8401-f6147c25455e kubernetes.host: pool-7nwspiw6h-odh2 kubernetes.master_url: https://10.245.0.1:443/api
Jul 8, 2019 @ 14:27:47.896	kubernetes.pod_name: counter log: 62: Mon Jul 8 18:27:47 UTC 2019 stream: stdout docker.container_id: 98beea3f30e5fa9939f91f8bf1eb7cccdef5be4c90dc96b7167865db630bb9e9 kubernetes.container_name: count kubernetes.namespace_name: default kubernetes.container_image: busybox:latest kubernetes.container_image_id: docker-pullable://busybox@sha256:c94cf1b87ccb80f2e6414ef913c748b105060debda482058d2b8d0fce39f11b9 kubernetes.pod_id: efd1ee59-a1ad-11e9-8401-f6147c25455e kubernetes.host: pool-7nwspiw6h-odh2 kubernetes.master_url: https://10.245.0.1:443/api
Jul 8, 2019 @ 14:27:46.894	kubernetes.pod_name: counter log: 61: Mon Jul 8 18:27:46 UTC 2019 stream: stdout docker.container_id: 98beea3f30e5fa9939f91f8bf1eb7cccdef5be4c90dc96b7167865db630bb9e9 kubernetes.container_name: count kubernetes.namespace_name: default kubernetes.container_image: busybox:latest kubernetes.container_image_id: docker-pullable://busybox@sha256:c94cf1b87ccb80f2e6414ef913c748b105060debda482058d2b8d0fce39f11b9 kubernetes.pod_id: efd1ee59-a1ad-11e9-8401-f6147c25455e kubernetes.host: pool-7nwspiw6h-odh2 kubernetes.master_url: https://10.245.0.1:443/api

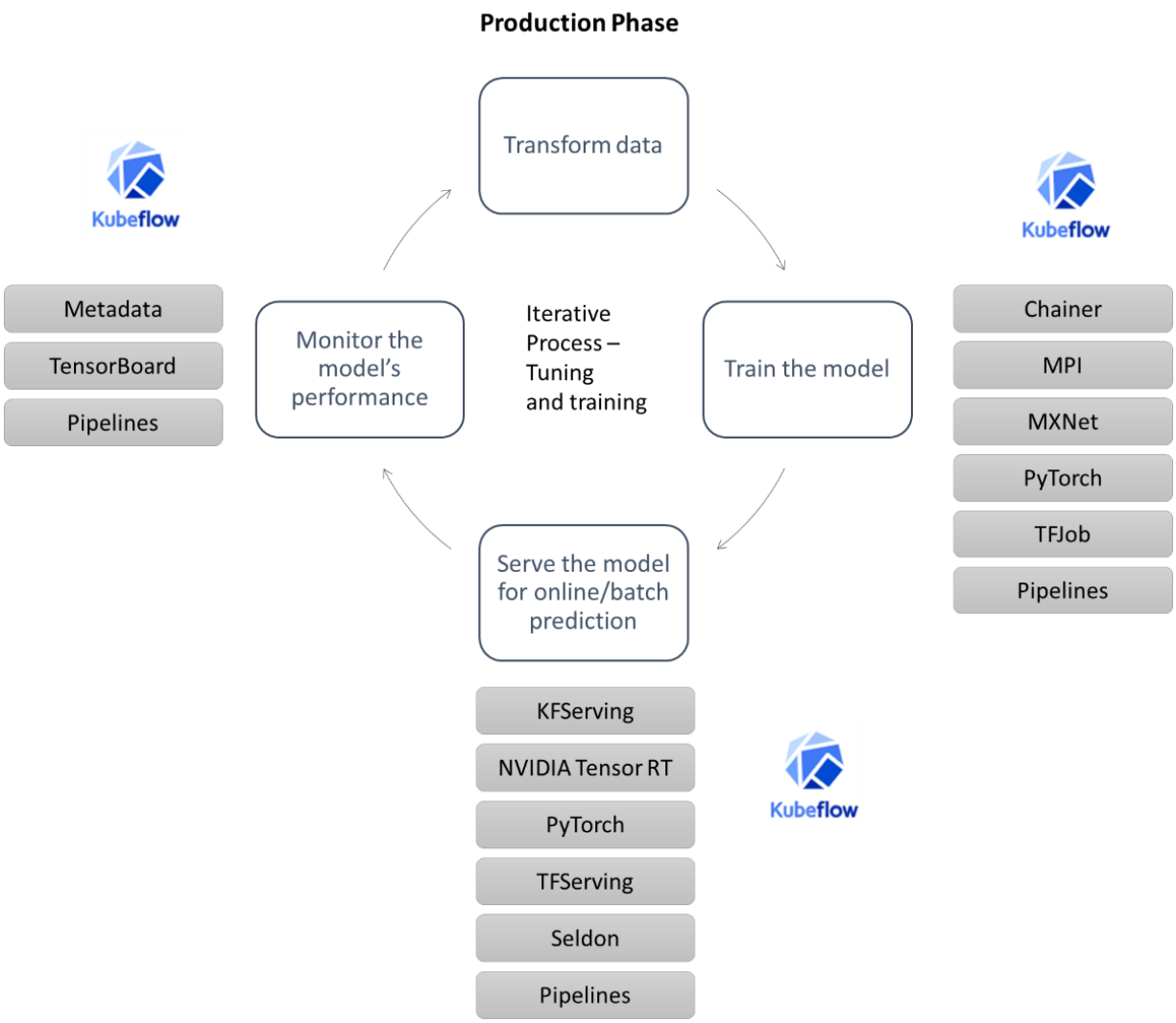
Chapter 9: Using Kubefl ow to Run AI/MLOps Workloads






Production Phase
Transform data
Train the model
Serve the model for online/batch prediction
Monitor the model's performance
Iterative Process – Tuning and training


Experimental Phase
Identify problem, collect and analyze data
Choose an ML algorithm and code your model
Experiment with data and model training
Tune the model hyperparameters
Iterative Process – Tuning and training
Kubeflow
PyTorch
Scikit-learn
TensorFlow
XGBoost
Jupyter Notebook
Fairing
Pipelines
Kubeflow
Katib
Kubeflow




```
ubuntu@microk8s:~/Desktop$ snap install microk8s --classic --chann
el=1.21/stable
microk8s (1.21/stable) v1.21.10 from Canonical* installed
ubuntu@microk8s:~/Desktop$
```

```
ubuntu@microk8s:~/Desktop$ microk8s status
microk8s is running
high-availability: no
  datastore master nodes: 127.0.0.1:19001
  datastore standby nodes: none
addons:
  enabled:
    ha-cluster           # Configure high availability o
  disabled:
    ambassador           # Ambassador API Gateway and In
    cilium               # SDN, fast with full network p
    dashboard            # The Kubernetes dashboard
    dns                  # CoreDNS
```

```
ubuntu@microk8s:~/Desktop$ microk8s enable dns storage ingress metallb:10.64.14
0.43-10.64.140.49
Infer repository core for addon dns
Infer repository core for addon storage
Infer repository core for addon ingress
Infer repository core for addon metallb
Addon core/dns is already enabled
Addon core/storage is already enabled
Enabling Ingress
ingressclass.networking.k8s.io/public created
namespace/ingress created
serviceaccount/nginx-ingress-microk8s-serviceaccount created
clusterrole.rbac.authorization.k8s.io/nginx-ingress-microk8s-clusterrole create
d
role.rbac.authorization.k8s.io/nginx-ingress-microk8s-role created
clusterrolebinding.rbac.authorization.k8s.io/nginx-ingress-microk8s created
rolebinding.rbac.authorization.k8s.io/nginx-ingress-microk8s created
```

```
clusterrole.rbac.authorization.k8s.io/metallb-system:speaker
role.rbac.authorization.k8s.io/config-watcher created
role.rbac.authorization.k8s.io/pod-lister created
clusterrolebinding.rbac.authorization.k8s.io/metallb-system:c
clusterrolebinding.rbac.authorization.k8s.io/metallb-system:s
rolebinding.rbac.authorization.k8s.io/config-watcher created
rolebinding.rbac.authorization.k8s.io/pod-lister created
Warning: spec.template.spec.nodeSelector[beta.kubernetes.io/o
ce v1.14; use "kubernetes.io/os" instead
daemonset.apps/speaker created
deployment.apps/controller created
configmap/config created
MetalLB is enabled
ubuntu@microk8s:~/Desktop$
```

```
ubuntu@microk8s:~/Desktop$ sudo snap install juju --classic
juju 2.9.25 from Canonical* installed
ubuntu@microk8s:~/Desktop$
```

```
ubuntu@microk8s:~/Desktop$ juju bootstrap microk8s --agent-version="2.9.22"
Since Juju 2 is being run for the first time, downloaded the latest public clou
d information.
Creating Juju controller "microk8s-localhost" on microk8s/localhost
Bootstrap to Kubernetes cluster identified as microk8s/localhost
Fetching Juju Dashboard 0.8.1
Creating k8s resources for controller "controller-microk8s-localhost"
Downloading images
Starting controller pod
Bootstrap agent now started
Contacting Juju controller at 10.152.183.18 to verify accessibility...

Bootstrap complete, controller "microk8s-localhost" is now available in namespa
ce "controller-microk8s-localhost"

Now you can run
        juju add-model <model-name>
to create a new model to deploy k8s workloads.
ubuntu@microk8s:~/Desktop$
```

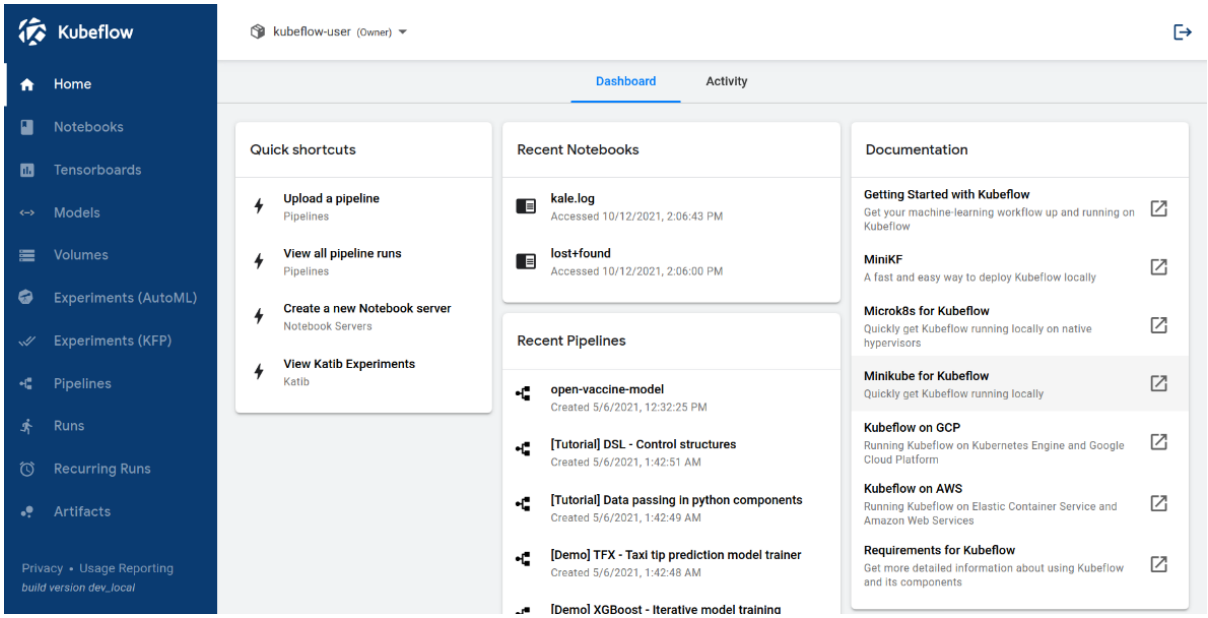
```
ubuntu@microk8s:~/Desktop$ juju add-model kubeflow
Added 'kubeflow' model on microk8s/localhost with credential 'microk8s' for use
r 'admin'
ubuntu@microk8s:~/Desktop$
```

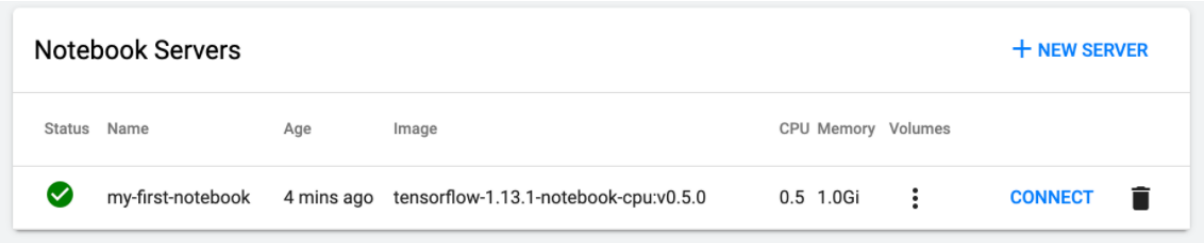
```
ubuntu@microk8s:~/Desktop$ juju deploy kubeflow-lite --trust
Located bundle "kubeflow-lite" in charm-hub, revision 60
Located charm "admission-webhook" in charm-hub, channel stable
Located charm "argo-controller" in charm-hub, channel stable
Located charm "dex-auth" in charm-hub, channel 2.28/stable
Located charm "envoy" in charm-hub, channel stable
Located charm "istio-gateway" in charm-hub, channel 1.5/stable
Located charm "istio-pilot" in charm-hub, channel 1.5/stable
Located charm "jupyter-controller" in charm-hub, channel stable
Located charm "jupyter-ui" in charm-hub, channel stable
Located charm "kfp-api" in charm-hub, channel stable
Located charm "charmed-osm-mariadb-k8s" in charm-hub, channel stable
Located charm "kfp-persistence" in charm-hub, channel stable
Located charm "kfp-profile-controller" in charm-hub, channel stable
Located charm "kfp-schedwf" in charm-hub, channel stable
```

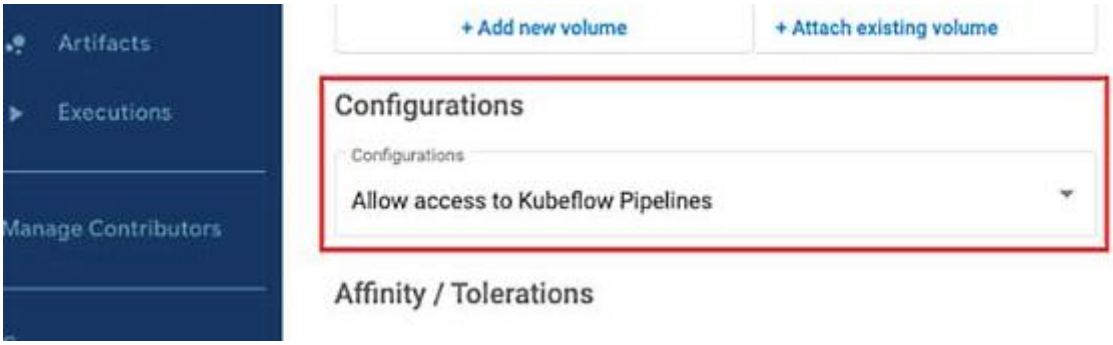
```
- add relation kfp-api:kfp-api - kfp-ui:kfp-api
- add relation kfp-api:kfp-viz - kfp-viz:kfp-viz
- add relation kfp-api:object-storage - minio:object-storage
- add relation kfp-profile-controller:object-storage - minio:object-storage
- add relation kfp-ui:object-storage - minio:object-storage
- add relation kubeflow-profiles - kubeflow-dashboard
- add relation mlmd:grpc - envoy:grpc
Deploy of bundle completed.
ubuntu@microk8s:~/Desktop$
```

```
ubuntu@microk8s:~/Desktop$ juju config dex-auth public-url
=http://10.64.140.43.nip.io
ubuntu@microk8s:~/Desktop$ juju config oidc-gatekeeper pub
lic-url=http://10.64.140.43.nip.io
```

```
ubuntu@microk8s:~/Desktop$ juju config dex-auth static-use
rname=admin
ubuntu@microk8s:~/Desktop$ juju config dex-auth static-pas
sword=admin
```







From Notebook to Kubeflow Pipeline using Fashion MNIST

In this notebook, we will walk you through the steps of converting a machine learning model, which you may already have on a jupyter notebook, into a Kubeflow pipeline. As an example, we will make use of the fashion we will make use of the fashion MNIST dataset and the Basic classification with Tensorflow example.

In this example we use:

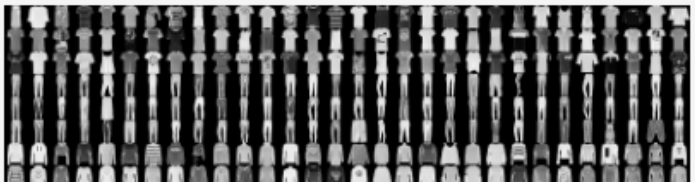
- **Kubeflow pipelines -** Kubeflow Pipelines is a machine learning workflow platform that is helping data scientists and ML engineers tackle experimentation and productionization of ML workloads. It allows users to easily orchestrate scalable workloads using an SDK right from the comfort of a Jupyter Notebook.
- **Microk8s -** Microk8s is a service that gives you the ability to spin up a lightweight Kubernetes cluster right on your local machine. It comes with Kubeflow built right in.

Note: This notebook is to be run on a notebook server inside the Kubeflow environment.

Section 1: Data exploration (as in here)

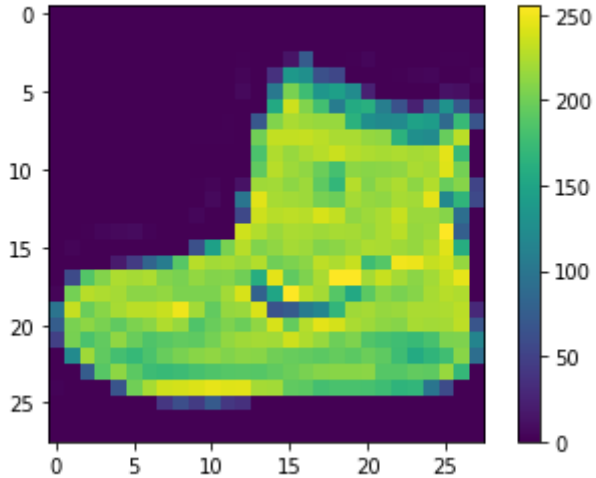
The Fashion MNIST dataset contains 70,000 grayscale images in 10 clothing categories. Each image is 28x28 pixels in size. We chose this dataset to demonstrate the funtionality of Kubeflow Pipelines without introducing too much complexity in the implementation of the ML model.

To familiarize you with the dataset we will do a short exploration. It is always a good idea to understand your data before you begin any kind of analysis.



In [ ]:
```
plt.figure()
plt.imshow(train_images[0])
plt.colorbar()
plt.grid(False)
plt.show()
```

In [ ]:
```
train_images = train_images / 255.0

test_images = test_images / 255.0
```



In [ ]:
```
plt.figure(figsize=(10,10))
for i in range(25):
    plt.subplot(5,5,i+1)
    plt.xticks([])
    plt.yticks([])
    plt.grid(False)
    plt.imshow(train_images[i], cmap=plt.cm.binary)
    plt.xlabel(class_names[train_labels[i]])
plt.show()
```

In [ ]:
```
!pip install --user --upgrade kfp
```

```
# Run a training job with specified number of epochs
model.fit(train_images, train_labels, epochs=10)

# Evaluate the model and print the results
test_loss, test_acc = model.evaluate(test_images,  test_labels, verbose=2)
print('Test accuracy:', test_acc)

# Save the model to the designated
model.save(f'{data_path}/{model_file}')

# Save the test_data as a pickle file to be used by the predict component.
with open(f'{data_path}/test_data', 'wb') as f:
    pickle.dump((test_images,test_labels), f)
```

```
# Define a Softmax layer to define outputs as probabilities
probability_model = tf.keras.Sequential([model,
                                         tf.keras.layers.Softmax()])
```

```
# See https://github.com/kubeflow/pipelines/issues/2320 for explanation on this line.
image_number = int(image_number)

# Grab an image from the test dataset.
img = test_images[image_number]

# Add the image to a batch where it is the only member.
img = (np.expand_dims(img,0))

# Predict the label of the image.
predictions = probability_model.predict(img)
```

In [ ]:
```
# Create train and predict lightweight components.
train_op = comp.func_to_container_op(train, base_image='tensorflow/tensorflow:latest-gpu-py3')
predict_op = comp.func_to_container_op(predict, base_image='tensorflow/tensorflow:latest-gpu-py3')
```

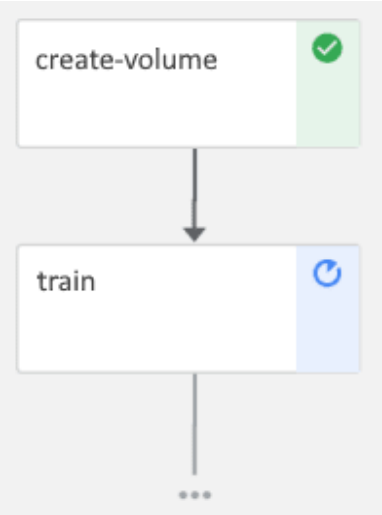
In [ ]:
```
client = kfp.Client(host='pipelines-api.kubeflow.svc.cluster.local:8888')
```

In [ ]:
```python
# Define the pipeline
@dsl.pipeline(
   name='MNIST Pipeline',
   description='A toy pipeline that performs mnist model training and prediction.'
)

# Define parameters to be fed into pipeline
def mnist_container_pipeline(
    data_path: str,
    model_file: str,
    image_number: int
):

    # Define volume to share data between components.
    vop = dsl.VolumeOp(
    name="create_volume",
    resource_name="data-volume",
    size="1Gi",
    modes=dsl.VOLUME_MODE_RWM)

    # Create MNIST training component.
    mnist_training_container = train_op(data_path, model_file) \
                                   .add_pvolumes({data_path: vop.volume})

    # Create MNIST prediction component.
    mnist_predict_container = predict_op(data_path, model_file, image_number) \
                                   .add_pvolumes({data_path: mnist_training_container.pvolume})
```

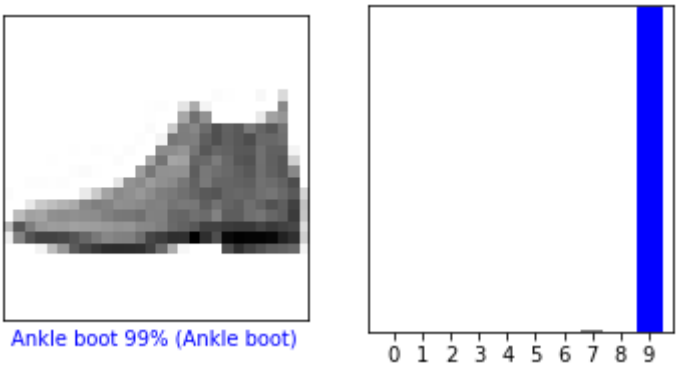

```python
# Create MNIST training component.
mnist_training_container = train_op(data_path, model_file) \
                                   .add_pvolumes({data_path: vop.volume})

# Create MNIST prediction component.
mnist_predict_container = predict_op(data_path, model_file, image_number) \
                                   .add_pvolumes({data_path: mnist_training_container.pvolume})
```


```python
# Print the result of the prediction
mnist_result_container = dsl.ContainerOp(
    name="print_prediction",
    image='library/bash:4.4.23',
    pvolumes={data_path: mnist_predict_container.pvolume},
    arguments=['cat', f'{data_path}/result.txt']
)
```

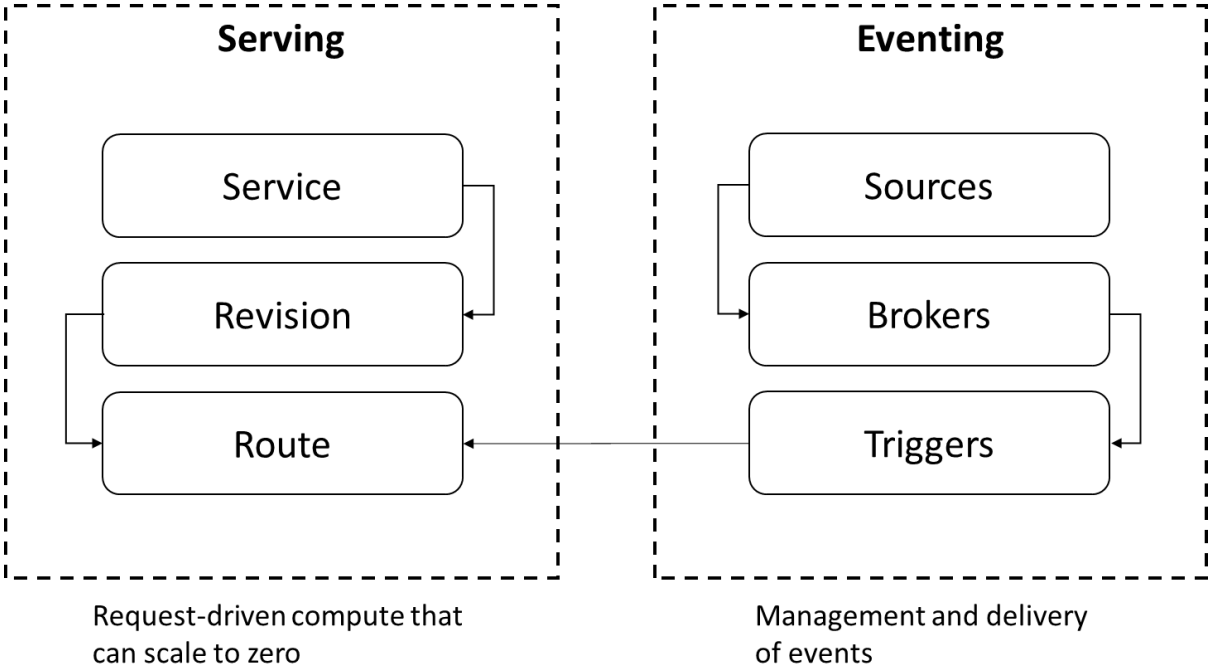
In [ ]:
```
experiment_name = 'fashion_mnist_kubeflow'
run_name = pipeline_func.__name__ + ' run'

arguments = {"data_path":DATA_PATH,
             "model_file":MODEL_PATH,
             "image_number": IMAGE_NUMBER}

# Compile pipeline to generate compressed YAML definition of the pipeline.
kfp.compiler.Compiler().compile(pipeline_func,
  '{}.zip'.format(experiment_name))

# Submit pipeline directly from pipeline function
run_result = client.create_run_from_pipeline_func(pipeline_func,
                                                  experiment_name=experiment_name,
                                                  run_name=run_name,
                                                  arguments=arguments)
```




```
1875/1875 [==============================] - 3s 2ms/step - loss: 0.2949 - accuracy: 0.8913
Epoch 6/10
1875/1875 [==============================] - 3s 2ms/step - loss: 0.2776 - accuracy: 0.8977
Epoch 7/10
1875/1875 [==============================] - 3s 2ms/step - loss: 0.2669 - accuracy: 0.9022
Epoch 8/10
1875/1875 [==============================] - 3s 2ms/step - loss: 0.2552 - accuracy: 0.9046
Epoch 9/10
1875/1875 [==============================] - 3s 2ms/step - loss: 0.2463 - accuracy: 0.9089
Epoch 10/10
1875/1875 [==============================] - 3s 2ms/step - loss: 0.2376 - accuracy: 0.9117
<keras.callbacks.History at 0x7f5f2c785110>
```



Chapter 10: Going Serverless with Knative and OpenFaaS Frameworks




```
$ microk8s enable knative
Enabling Knative
Enabling Istio
Fetching istioctl version v1.10.3.
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   668  100   668    0     0   2241      0 --:--:-- --:--:-- --:--:--  2241
100 21.3M  100 21.3M    0     0  15.3M      0  0:00:01  0:00:01 --:--:-- 31.1M
istio-1.10.3/
istio-1.10.3/manifests/
istio-1.10.3/manifests/charts/
istio-1.10.3/manifests/charts/istio-operator/
```

```
service/broker-ingress created
deployment.apps/mt-broker-controller created
Warning: autoscaling/v2beta2 HorizontalPodAutoscaler is deprecated in v1.23+, unavaila
 autoscaling/v2 HorizontalPodAutoscaler
horizontalpodautoscaler.autoscaling/broker-ingress-hpa created
horizontalpodautoscaler.autoscaling/broker-filter-hpa created


Visit https://knative.dev/docs/admin/ to customize which broker channel
implementation is used and to specify which configurations are used for which namespac


$
```

```
$ kubectl get pods -n knative-serving
NAME                                    READY   STATUS    RESTARTS   AGE
autoscaler-74f697b6c6-bt4sl             1/1     Running   0          2m31s
controller-84f98b57b-jd2q7              1/1     Running   0          2m30s
activator-6b8d5bccb4-7sqrs              1/1     Running   0          2m31s
domain-mapping-69479cf66f-8742d         1/1     Running   0          2m30s
net-istio-controller-6b876996dc-9dkl6   1/1     Running   0          2m15s
net-istio-webhook-d45dbdcb6-l5xs7       1/1     Running   0          2m15s
default-domain-fk62r                    1/1     Running   0          2m7s
domainmapping-webhook-bb67b5f65-8hx6p   1/1     Running   0          2m30s
webhook-5dcd765485-wf8ck                1/1     Running   0          2m29s
$
```

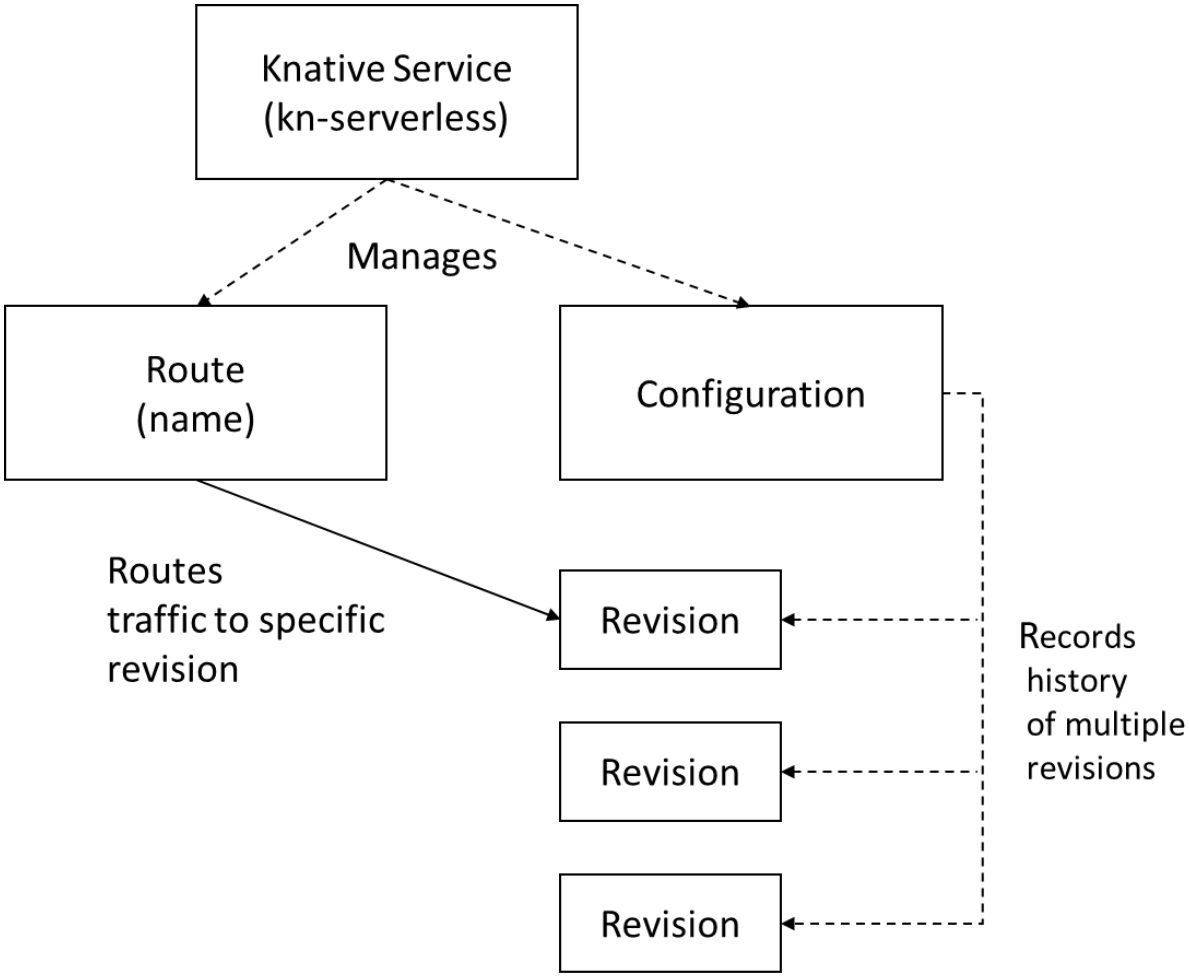
```
$ kubectl get pods -n knative-eventing
NAME                                    READY   STATUS    RESTARTS   AGE
eventing-controller-7bfd95cc79-ztx4z    1/1     Running   0          2m39s
eventing-webhook-c7998d8f9-gcj55        1/1     Running   0          2m38s
imc-dispatcher-dd5bbb4d7-vrqj7          1/1     Running   0          2m13s
imc-controller-6f74957b95-g5x9s         1/1     Running   0          2m18s
mt-broker-controller-658f88d698-j7wtd   1/1     Running   0          89s
mt-broker-filter-5fd68bd989-d7dcp       1/1     Running   0          91s
mt-broker-ingress-5bd6749895-kdlmk      1/1     Running   0          90s
$
```

```
$ sudo curl -o /usr/local/bin/kn -sL https://github.com/knative/client/releases/download/knative-v1.3.1
kn-linux-amd64
$ sudo chmod +x /usr/local/bin/kn
$
```

```
$ kn version
Version:      v1.3.1
Build Date:   2022-03-11 18:43:10
Git Revision: a591c0c0
Supported APIs:
* Serving
  - serving.knative.dev/v1 (knative-serving v1.3.0)
* Eventing
  - sources.knative.dev/v1 (knative-eventing v1.3.0)
  - eventing.knative.dev/v1 (knative-eventing v1.3.0)
$
```

```
$ microk8s config > $HOME/.kube/config
```

```
$ kn service create kn-serverless --image gcr.io/knative-samples/helloworld-go --env TARGET=upnxtblog.com
Creating service 'kn-serverless' in namespace 'default':

  0.083s The Route is still working to reflect the latest desired specification.
  0.105s Configuration "kn-serverless" is waiting for a Revision to become ready.
  0.118s ...
 45.781s ...
 45.873s Ingress has not yet been reconciled.
 45.952s Waiting for Envoys to receive Endpoints data.
 46.260s Waiting for load balancer to be ready
 46.519s Ready to serve.

Service 'kn-serverless' created to latest revision 'kn-serverless-00001' is available at URL:
http://kn-serverless.default.example.com
$
```




```
$ curl http://$SERVICE_IP:$INGRESS_PORT/ -H 'Host: kn-serverless.default.example.com'
Hello upnxtblog.com!
$
```

```
Every 2.0s: kubectl get pods
0:00 Fri Apr  1 06:19:45 2022
NAME                                          READY   STATUS        RESTARTS
kn-serverless-00001-deployment-57bf78bd47-tz4gc   2/2     Terminating   0
s
```

```
$ kubectl get pods
NAME                                                  READY   STATUS    RESTARTS   AGE
kn-serverless-00001-deployment-57bf78bd47-tz4gc       2/2     Running   0          109s
$
```

```
$ microk8s enable dns
Enabling DNS
Applying manifest
serviceaccount/coredns created
configmap/coredns created
deployment.apps/coredns created
service/kube-dns created
clusterrole.rbac.authorization.k8s.io/coredns created
clusterrolebinding.rbac.authorization.k8s.io/coredns created
Restarting kubelet
DNS is enabled
$
```

```
$ microk8s enable registry
The registry will be created with the default size of 20Gi.
You can use the "size" argument while enabling the registry, eg microk8s.enabl
Enabling default storage class
deployment.apps/hostpath-provisioner created
storageclass.storage.k8s.io/microk8s-hostpath created
serviceaccount/microk8s-hostpath created
clusterrole.rbac.authorization.k8s.io/microk8s-hostpath created
clusterrolebinding.rbac.authorization.k8s.io/microk8s-hostpath created
Storage will be available soon
Applying registry manifest
namespace/container-registry created
persistentvolumeclaim/registry-claim created
deployment.apps/registry created
service/registry created
configmap/local-registry-hosting configured
The registry is enabled
$
```

```
$ microk8s enable openfaas
Addon dns is already enabled.
Enabling Helm 3
Fetching helm version v3.5.0.
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 11.7M  100 11.7M    0     0  12.8M      0 --:--:-- --:--:-- --:--:-- 12.8M
Helm 3 is enabled

Enabling OpenFaaS
Operator: false
Basic Auth enabled: true
WARNING: Kubernetes configuration file is group-readable. This is insecure. Loca
3052/credentials/client.config
```

```
REVISION: 1
TEST SUITE: None
NOTES:
To verify that openfaas has started, run:

  kubectl -n openfaas get deployments -l
To retrieve the admin password, run:

  echo $(kubectl -n openfaas get secret ba
ecode)
OpenFaaS has been installed
$
```

```
$ kubectl -n openfaas get deployments -l "release=openfaas, app=openfaas"
NAME                 READY   UP-TO-DATE   AVAILABLE   AGE
prometheus           1/1     1            1           92s
nats                 1/1     1            1           92s
queue-worker         1/1     1            1           92s
basic-auth-plugin    1/1     1            1           92s
alertmanager         1/1     1            1           92s
gateway              1/1     1            1           92s
$
```

```
$ curl -sSL --insecure https://cli.openfaas.com | sudo -E sh
Finding latest version from GitHub
0.14.2
Downloading package https://github.com/openfaas/faas-cli/releas
li
Download complete.

Running with sufficient permissions to attempt to move faas-cli
New version of faas-cli installed to /usr/local/bin
Creating alias 'faas' for 'faas-cli'.

 ___                   _____           ____
/ __\ _ __   ___ _ __ |_   _|__  ___  / ___|
| |  | '_ \ / _ \ '_ \| |/ _` |/ _ \ \___ \
| |_| | |) |  __/ |  | | | (_| | (_| |___) |
 \___/| .__/ \___|_| |_|_| \__,_|\__,_|____/
      |_|


CLI:
 commit:  b1c09c0243f69990b6c81a17d7337f0fd23e7542
 version: 0.14.2
$
```

```
$ faas-cli new --lang python --prefix localhost:32000 openfaas-serverless
Folder: openfaas-serverless created.
 ___                   _____           ____
 /   \ _ __   ___ _ __ |_   _|__  ___ _/___|
| | | | '_ \ / _ \ '_ \| | / _ `|/ _ `\___ \
| |_| | |_) |  __/ | | | | (_| | (_| |___) |
 \___/| .__/ \___|_| |_|_| \__,_|\__,_|____/
      |_|


Function created in folder: openfaas-serverless
Stack file written: openfaas-serverless.yml
$
```

```
$ ls
openfaas-serverless  openfaas-serverless.yml  snap  template
$
```

```
$ faas-cli build -f ./openfaas-serverless.yml
[0] > Building openfaas-serverless.
Clearing temporary build folder: ./build/openfaas-
Preparing: ./openfaas-serverless/ build/openfaas-s
Building: localhost:32000/openfaas-serverless:late
Sending build context to Docker daemon  8.192kB
Step 1/31 : FROM --platform=${TARGETPLATFORM:-linu
chdog
 ---> 6f97aa96da81
Step 2/31 : FROM --platform=${TARGETPLATFORM:-linu
2.7-alpine: Pulling from library/python
aad63a933944: Pulling fs layer
259d822268fb: Pulling fs layer
10ba96d218d3: Pulling fs layer
44ba9f6a4209: Pulling fs layer
44ba9f6a4209: Waiting
```

```
 ---> 3202cbddcb56
Step 31/31 : CMD ["fwatchdog"]
 ---> Running in bd9c86d0aaf9
Removing intermediate container bd9c86d0aaf9
 ---> b3af5c4a4e49
Successfully built b3af5c4a4e49
Successfully tagged localhost:32000/openfaas-serve
Image: localhost:32000/openfaas-serverless:latest
[0] < Building openfaas-serverless done in 35.92s.
[0] Worker done.

Total build time: 35.92s
$
```

```
$ faas-cli push -f ./openfaas-serverless.yml
[0] > Pushing openfaas-serverless [localhost:32000/o
The push refers to repository [localhost:32000/openf
5c917ca7243b: Pushed
4ef521ff4222: Pushed
680a1beb2e40: Pushed
c59cdfad7930: Pushed
240550aa0dec: Pushed
dac9a8959481: Pushed
444490860c1b: Pushed
34a6d15eaa0f: Pushed
3e304456f938: Pushed
01e115009001: Pushed
65909e40f7e4: Pushed
d3a87395ac2b: Pushed
879c0d8666e3: Pushed
20a7b70bdf2f: Pushed
3fc750b41be7: Pushed
beee9f30bc1f: Pushed
latest: digest: sha256:4034411878816583b1db6458ed570
[0] < Pushing openfaas-serverless [localhost:32000/o
[0] Worker done.
```

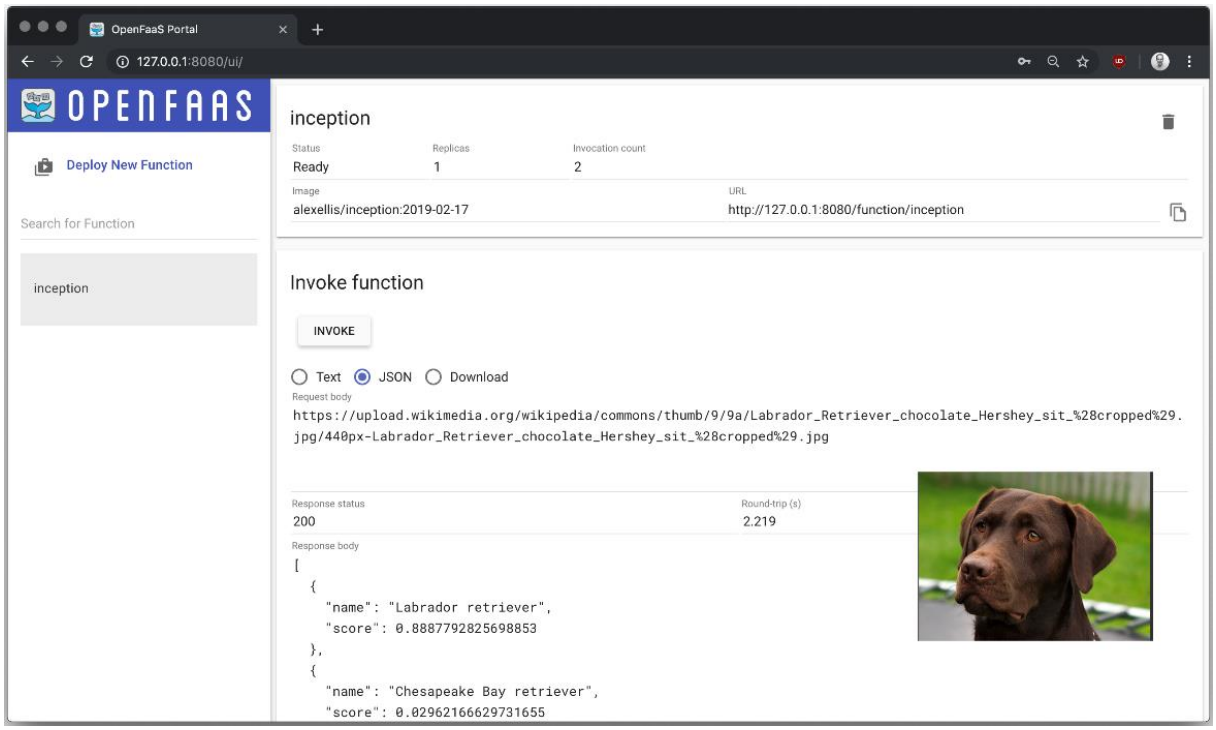
```
$ export OPENFAAS_URL=http://127.0.0.1:31112
$
```

```
$ echo $(kubectl -n openfaas get secret basic-auth -o jsonpath="{.data.basic-auth-password}" | base64 --d
ecode)
V5NjmSMhIb2q
```

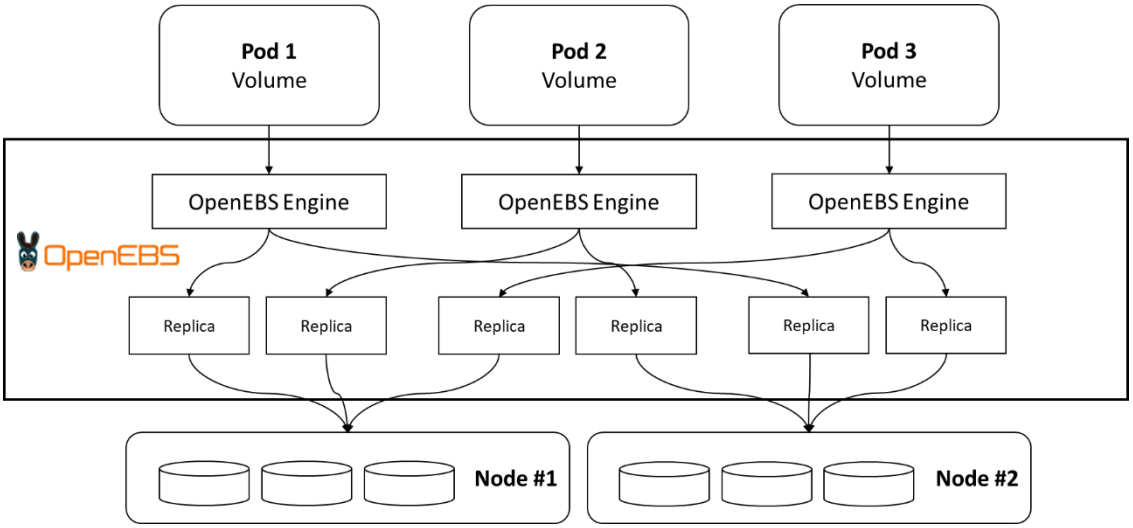
```
$ faas-cli login admin
Must provide a non-empty password via --password or --password-stdin
$ faas-cli login admin --password V5NjmSMhIb2q
WARNING! Using --password is insecure, consider using: cat ~/faas_pass.
sword-stdin
Calling the OpenFaaS server to validate the credentials...
credentials saved for admin http://127.0.0.1:31112
$
```

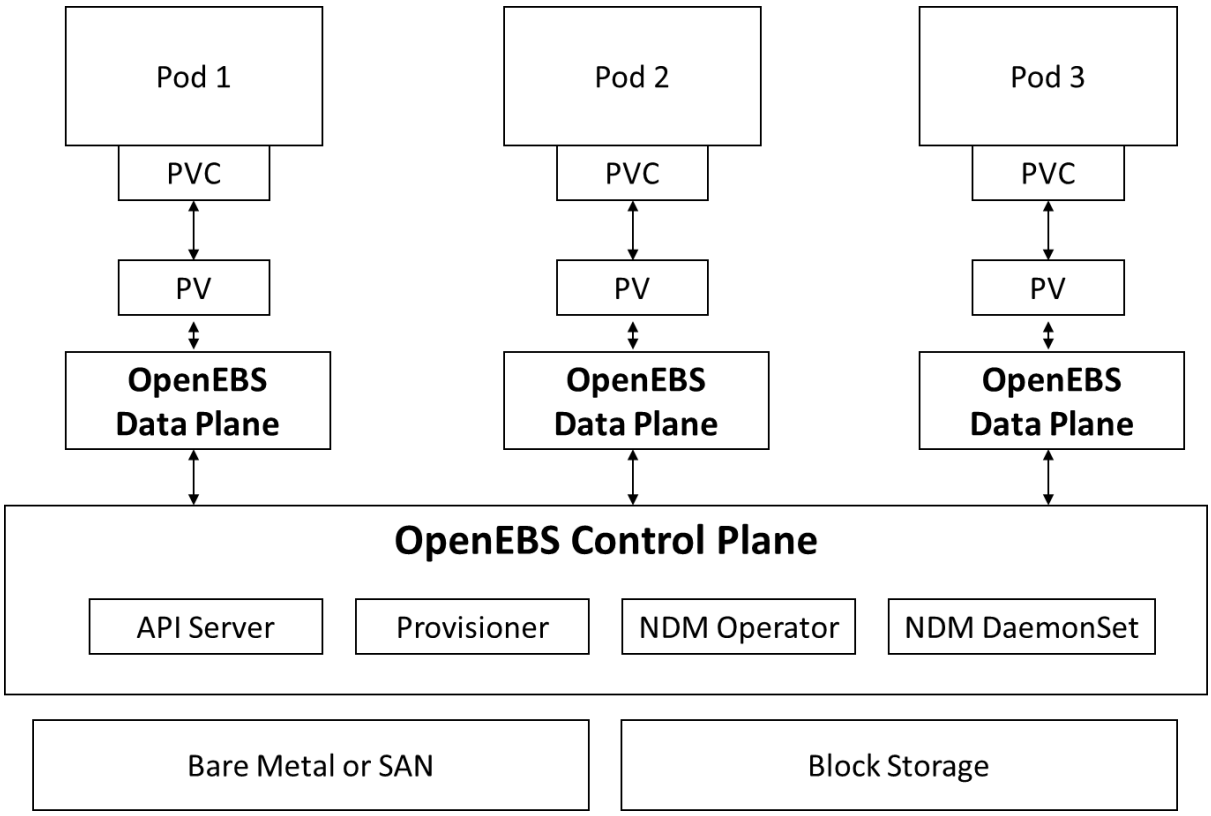
```
$ faas-cli deploy -f ./openfaas-serverless.yml
Deploying: openfaas-serverless.

Deployed. 202 Accepted.
URL: http://127.0.0.1:31112/function/openfaas-serverless

$
```

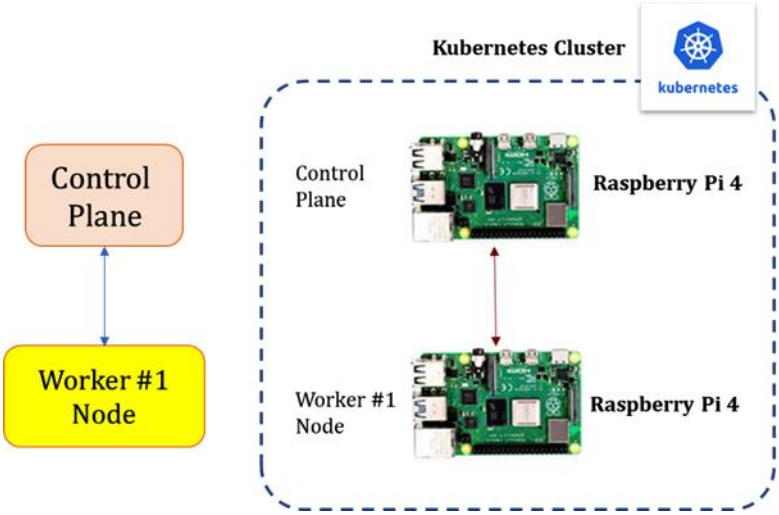
```
$ echo "Hello Upnxtblog.com" | faas-cli invoke -f openfaas-serverless.yml openfaas-serverless
Hello Upnxtblog.com

$
```



Chapter 11: Managing Storage Replication with OpenEBS





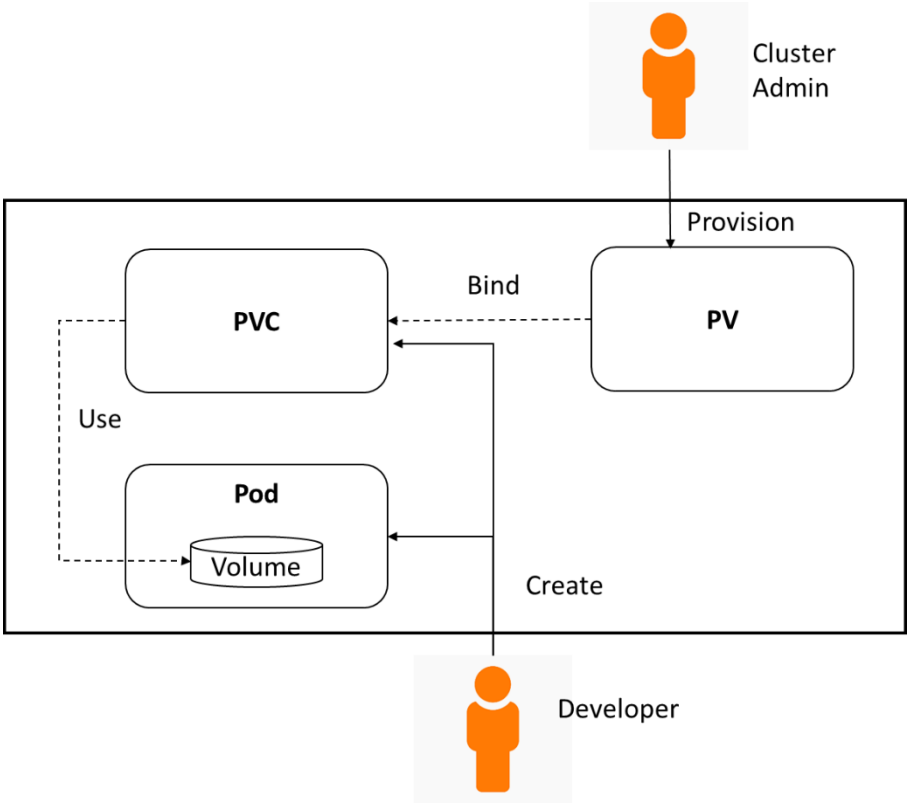



```
ubuntu@controlplane:~$ microk8s enable openebs
iscsid is not available or enabled.  Make sure iscsi is installed on all nodes.
To enable iscsid:
      sudo systemctl enable iscsid
Please refer to the OpenEBS prerequisites (https://docs.openebs.io/docs/next/prerequisites.html)
ubuntu@controlplane:~$
```

```
ubuntu@controlplane:~$ sudo systemctl enable iscsid
Synchronizing state of iscsid.service with SysV service script with /lib/systemd/syst
Executing: /lib/systemd/systemd-sysv-install enable iscsid
Created symlink /etc/systemd/system/sysinit.target.wants/iscsid.service → /lib/system
ubuntu@controlplane:~$
```

```
ubuntu@controlplane:~$ microk8s enable openebs
Addon dns is already enabled.
Enabling Helm 3
Fetching helm version v3.5.0.
  % Total    % Received % Xferd  Average Speed   Time    Time     Tim
                                 Dload  Upload   Total   Spent    Lef
100 10.4M  100 10.4M    0     0  2201k      0  0:00:04  0:00:04 --:--
Helm 3 is enabled
WARNING: Kubernetes configuration file is group-readable. This is ins
client.config
"openebs" has been added to your repositories
WARNING: Kubernetes configuration file is group-readable. This is ins
client.config
Hang tight while we grab the latest from your chart repositories...
...Successfully got an update from the "openebs" chart repository
Update Complete. □Happy Helming!□
WARNING: Kubernetes configuration file is group-readable. This is ins
client.config
NAME: openebs
LAST DEPLOYED: Mon Apr 11 14:05:39 2022
NAMESPACE: openebs
STATUS: deployed
REVISION: 1
TEST SUITE: None
NOTES:
Successfully installed OpenEBS.

Check the status by running: kubectl get pods -n openebs
```

```
ubuntu@controlplane:~$ kubectl get pods -n openebs
NAME                                          READY   STATUS    RESTARTS   AGE
openebs-ndm-9p4p7                             1/1     Running   0          12m
openebs-ndm-operator-7bd6898d96-7ml9c         1/1     Running   0          12m
openebs-cstor-cspc-operator-55f9cc6858-xltr8  1/1     Running   0          12m
openebs-jiva-operator-564964cb67-7qvs7        1/1     Running   0          12m
openebs-cstor-admission-server-5754659f4b-jnhxw 1/1   Running   0          12m
openebs-cstor-cvc-operator-754f9cb6b7-8vsjc   1/1     Running   0          12m
openebs-localpv-provisioner-658895c6c9-6jlvs  1/1     Running   0          12m
openebs-cstor-csi-node-9z2kk                  2/2     Running   0          12m
openebs-jiva-csi-node-h6rqx                   3/3     Running   0          12m
openebs-cstor-csi-controller-0                6/6     Running   0          12m
openebs-jiva-csi-controller-0                 5/5     Running   0          12m
ubuntu@controlplane:~$
```




```
ubuntu@controlplane:~$ kubectl get sc
NAME                        PROVISIONER           RECLAIMPOLICY   VOLUMEBINDINGMODE      ALLOWVO
openebs-jiva-csi-default    jiva.csi.openebs.io   Delete          Immediate              true
openebs-device              openebs.io/local      Delete          WaitForFirstConsumer   false
openebs-hostpath            openebs.io/local      Delete          WaitForFirstConsumer   false
ubuntu@controlplane:~$
```

```
ubuntu@controlplane:~$ kubectl apply -f postgres.yaml
persistentvolume/postgres-pv created
persistentvolumeclaim/postgres-pv-claim created
ubuntu@controlplane:~$
```

```
ubuntu@controlplane:~$ kubectl get pv
NAME          CAPACITY   ACCESS MODES   RECLAIM POLICY   STATUS   CLAIM                      STORAGECLASS
EASON   AGE
postgres-pv   5Gi        RWO            Retain           Bound    default/postgres-pv-claim  openebs-jiva-csi-default
        53s
ubuntu@controlplane:~$ kubectl get pvc
NAME                STATUS   VOLUME        CAPACITY   ACCESS MODES   STORAGECLASS               AGE
postgres-pv-claim   Bound    postgres-pv   5Gi        RWO            openebs-jiva-csi-default   57s
ubuntu@controlplane:~$
```

```
ubuntu@controlplane:~$ kubectl apply -f postgres-config.yaml
configmap/postgres-configuration created
ubuntu@controlplane:~$
```

```
ubuntu@controlplane:~$ kubectl describe configmap postgres-configuration
Name:         postgres-configuration
Namespace:    default
Labels:       app=postgres
Annotations:  <none>

Data
====
POSTGRES_DB:
----
postgresdb
POSTGRES_PASSWORD:
----
postgrespassword
POSTGRES_USER:
----
postgres

BinaryData
====

Events:  <none>
ubuntu@controlplane:~$
```

```
ubuntu@controlplane:~$ kubectl apply -f postgres-deployment.yaml
statefulset.apps/postgres-statefulset created
service/postgres-service created
ubuntu@controlplane:~$
```

```
ubuntu@controlplane:~$ kubectl get pods | grep post
postgres-statefulset-0   1/1     Running   0          3m22s
ubuntu@controlplane:~$
```

```
ubuntu@controlplane:~$ kubectl get svc | grep post
postgres-service        NodePort       10.152.183.6     <none>        5432:32367/TCP   3m50s
ubuntu@controlplane:~$
```

```
ubuntu@controlplane:~$ kubectl get pods -o wide | grep post
postgres-statefulset-0   1/1     Running   0          16m    10.1.49.81     controlplane   <none>
postgres-statefulset-1   1/1     Running   0          107s   10.1.235.151   worker1        <none>
ubuntu@controlplane:~$
```

```
ubuntu@controlplane:~$ kubectl exec -it postgres-statefulset-0 -- psql -U postgres
psql (12.10 (Debian 12.10-1.pgdg110+1))
Type "help" for help.

postgres=#
```

```
postgres=# CREATE DATABASE inventory_mgmt;
CREATE DATABASE
postgres=#
```

```
postgres-# \c inventory_mgmt
You are now connected to database "inventory_mgmt" as user "postgres".
inventory_mgmt-#
```

```
inventory_mgmt=# CREATE TABLE products_master (
    product_no integer,
    name text,
    price numeric
);
CREATE TABLE
inventory_mgmt=#
```

```
inventory_mgmt=# INSERT INTO products_master VALUES (1, 'Chair', 119.99);
INSERT 0 1
inventory_mgmt=#
```

```
inventory_mgmt=# INSERT INTO products_master VALUES (2, 'Work Table', 199.99);
INSERT 0 1
inventory_mgmt=#
```

```
inventory_mgmt=# SELECT * FROM products_master;
 product_no |     name     | price
------------+--------------+--------
          1 | Chair        | 119.99
          2 | Work Table   | 199.99
(2 rows)

inventory_mgmt=#
```

```
ubuntu@controlplane:~$ kubectl get pods -o wide | grep post
postgres-statefulset-0   1/1     Running   0          16m    10.1.49.81     controlplane   <none>
postgres-statefulset-1   1/1     Running   0          107s   10.1.235.151   worker1        <none>
ubuntu@controlplane:~$
```

```
ubuntu@controlplane:~$ kubectl cordon worker1
node/worker1 cordoned
ubuntu@controlplane:~$
```

```
ubuntu@controlplane:~$ kubectl drain --force --ignore-daemonsets worker1
node/worker1 already cordoned
error: unable to drain node "worker1" due to error:cannot delete Pods with local storage (use
erride): openebs/openebs-jiva-csi-controller-0, continuing command...
There are pending nodes to be drained:
 worker1
cannot delete Pods with local storage (use --delete-emptydir-data to override): openebs/opene
```

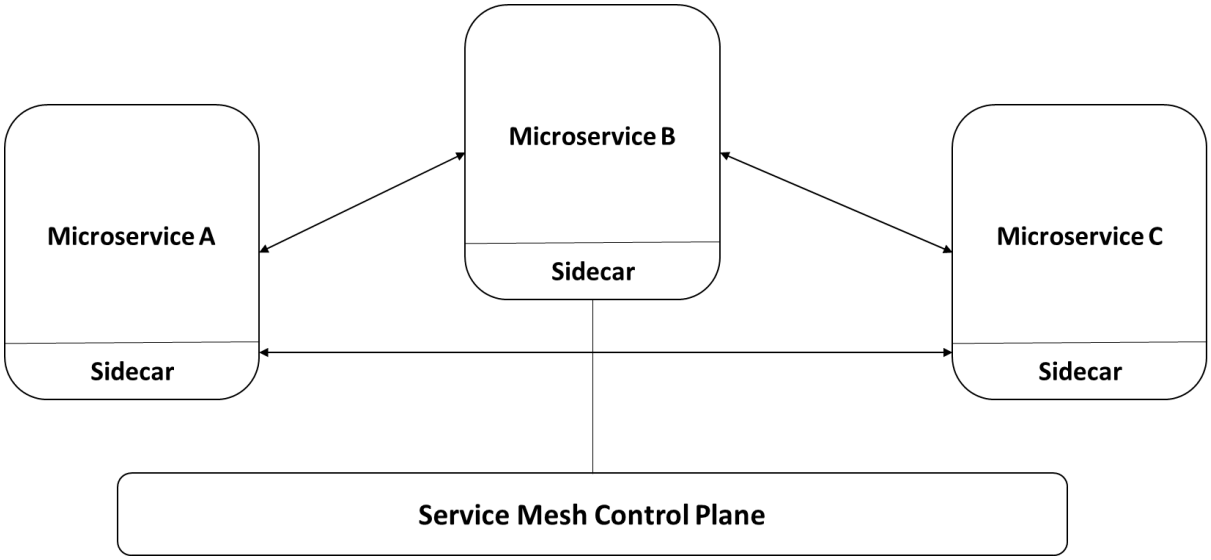
```
ubuntu@controlplane:~$ kubectl delete pod postgres-statefulset-1
pod "postgres-statefulset-1" deleted
ubuntu@controlplane:~$
```

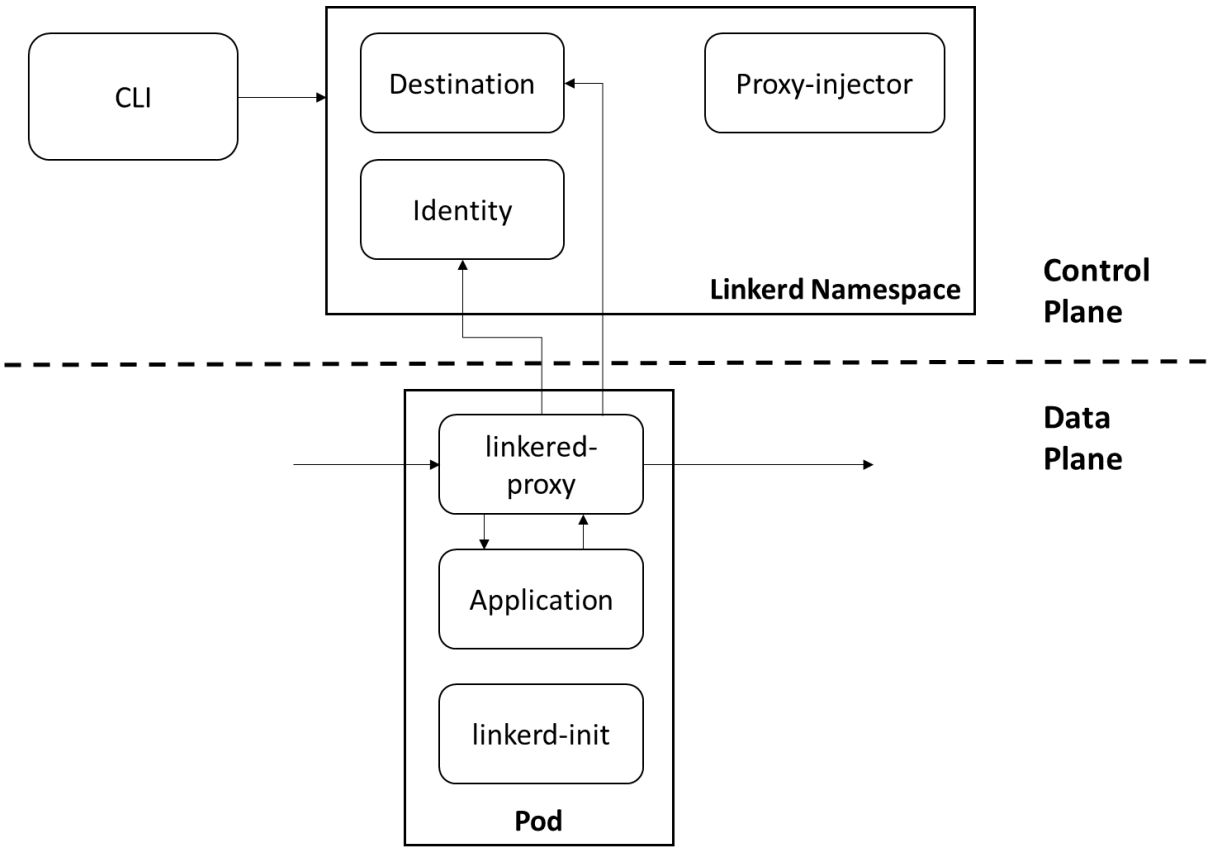
```
ubuntu@controlplane:~$ kubectl get pods -o wide | grep post
postgres-statefulset-0   1/1     Running   0          24m    10.1.49.81     controlplane   <none>           <none>
postgres-statefulset-1   1/1     Running   0          13s    10.1.49.69     controlplane   <none>           <none>
ubuntu@controlplane:~$
```

```
ubuntu@controlplane:~$ kubectl exec -it postgres-statefulset-1 -- psql -U postgres
psql (12.10 (Debian 12.10-1.pgdg110+1))
Type "help" for help.

postgres=#
```

```
ubuntu@controlplane:~$ kubectl exec -it postgres-statefulset-1 -- psql -U postgres
psql (12.10 (Debian 12.10-1.pgdg110+1))
Type "help" for help.

postgres=# \c inventory_mgmt;
You are now connected to database "inventory_mgmt" as user "postgres".
inventory_mgmt=# select * from products_master;
 product_no |    name    | price
------------+------------+--------
          1 | Chair      | 119.99
          2 | Work Table | 199.99
(2 rows)

inventory_mgmt=#
```

Chapter 12: Implementing Service Mesh for Cross-Cutting Concerns





```
$ microk8s enable linkerd
Fetching Linkerd2 version v2.11.1.
2.11.1
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   677  100   677    0     0   2763      0 --:--:-- --:--:-- --:--:--  2763
100 48.6M  100 48.6M    0     0  18.1M      0  0:00:02  0:00:02 --:--:-- 22.6M
Enabling Linkerd2
Enabling DNS
Applying manifest
serviceaccount/coredns created
configmap/coredns created
deployment.apps/coredns created
service/kube-dns created
clusterrole.rbac.authorization.k8s.io/coredns created
clusterrolebinding.rbac.authorization.k8s.io/coredns created
Restarting kubelet
DNS is enabled
namespace/linkerd created
```

```
mutatingwebhookconfiguration.admissionregistration.k8s.io/linkerd-tap-injec
service/tap-injector created
deployment.apps/tap-injector created
server.policy.linkerd.io/tap-injector-webhook created
serverauthorization.policy.linkerd.io/tap-injector created
service/web created
deployment.apps/web created
serviceprofile.linkerd.io/metrics-api.linkerd-viz.svc.cluster.local created
serviceprofile.linkerd.io/prometheus.linkerd-viz.svc.cluster.local created
serviceprofile.linkerd.io/grafana.linkerd-viz.svc.cluster.local created
Linkerd is starting
$
```

```
$ kubectl get pods -n linkerd
NAME                                      READY   STATUS
linkerd-identity-54795b9f9f-gh9cl         2/2     Running
linkerd-destination-8468749f6f-4jv5f      4/4     Running
linkerd-proxy-injector-6db5c59479-hr4gw   2/2     Running
$
```

```
$ kubectl apply -f  https://k8s.io/examples/application/deployment.yaml
deployment.apps/nginx-deployment created
$
```

```
$ kubectl get pods
NAME                                READY   STATUS
nginx-deployment-9456bbbf9-8v4s8    1/1     Running
nginx-deployment-9456bbbf9-cm59w    1/1     Running
$
```

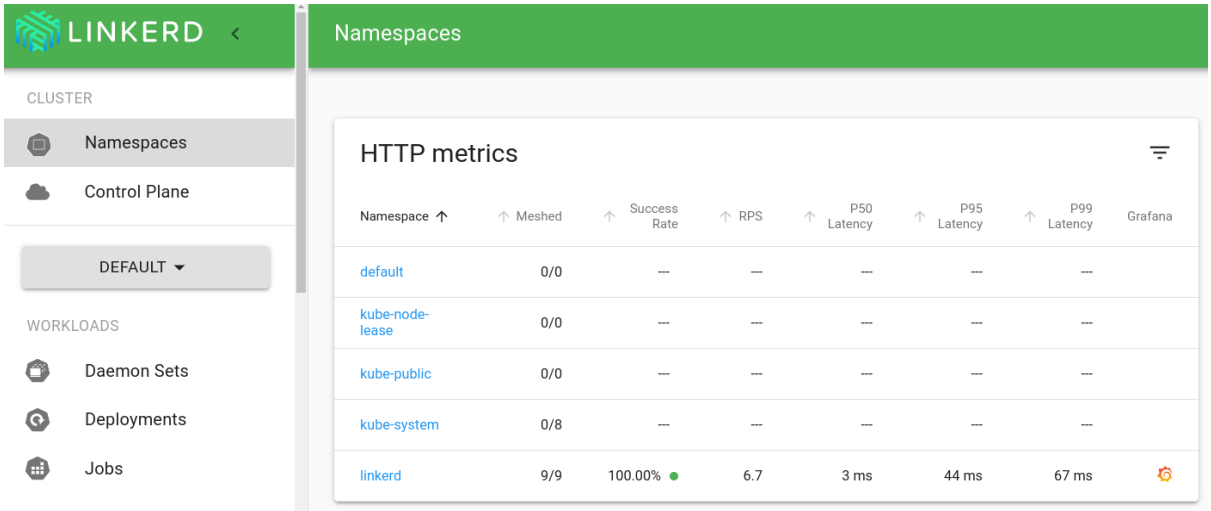
```
$   kubectl get deployment nginx-deployment -n default -o yaml \
>   | microk8s linkerd inject - \
>   | kubectl apply -f -

deployment "nginx-deployment" injected

deployment.apps/nginx-deployment configured
$
```

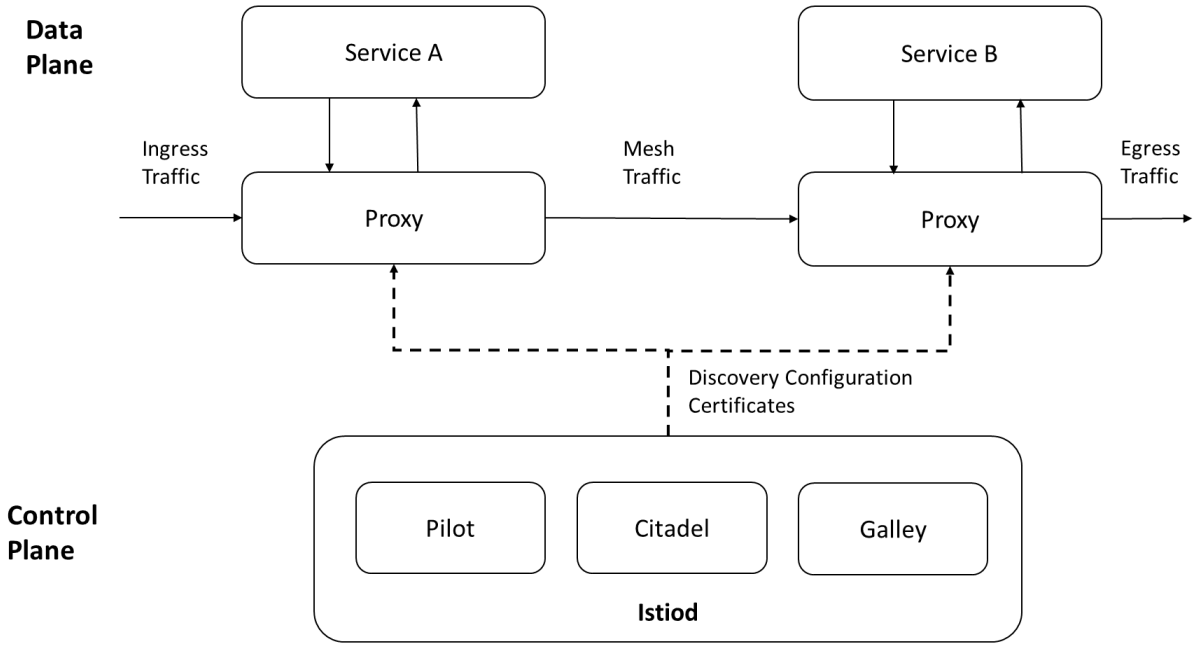
```
$ microk8s linkerd check --proxy
Linkerd core checks
===================

kubernetes-api
--------------
√ can initialize the client
√ can query the Kubernetes API

kubernetes-version
------------------
√ is running the minimum Kubernetes API version
√ is running the minimum kubectl version

linkerd-existence
-----------------
√ 'linkerd-config' config map exists
√ heartbeat ServiceAccount exist
√ control plane replica sets are ready
√ no unschedulable pods
√ control plane pods are ready
```

```
linkerd-viz
-----------
√ linkerd-viz Namespace exists
√ linkerd-viz ClusterRoles exist
√ linkerd-viz ClusterRoleBindings exist
√ tap API server has valid cert
√ tap API server cert is valid for at least 60
√ tap API service is running
√ linkerd-viz pods are injected
√ viz extension pods are running
√ viz extension proxies are healthy
√ viz extension proxies are up-to-date
√ viz extension proxies and cli versions match
√ prometheus is installed and configured correc
√ can initialize the client
√ viz extension self-check

Status check results are √
$
```

```
$ kubectl describe deployment nginx-deployment
Name:                   nginx-deployment
Namespace:              default
CreationTimestamp:      Sun, 17 Apr 2022 12:05:28 +0000
Labels:                 <none>
Annotations:            deployment.kubernetes.io/revision: 2
Selector:               app=nginx
Replicas:               2 desired | 2 updated | 2 total | 2
StrategyType:           RollingUpdate
MinReadySeconds:        0
RollingUpdateStrategy:  25% max unavailable, 25% max surge
Pod Template:
  Labels:       app=nginx
  Annotations:  linkerd.io/inject: enabled
  Containers:
   nginx:
    Image:        nginx:1.14.2
    Port:         80/TCP
    Host Port:    0/TCP
    Environment:  <none>
    Mounts:       <none>
```

```
$ microk8s linkerd viz dashboard &
[1] 19577
$ Linkerd dashboard available at:
http://localhost:50750
Grafana dashboard available at:
http://localhost:50750/grafana
Opening Linkerd dashboard in the default browser
Failed to open Linkerd dashboard automatically
Visit http://localhost:50750 in your browser to view the dashboard
```






```
$ microk8s enable istio
Enabling Istio
Fetching istioctl version v1.10.3.
  % Total    % Received % Xferd  Average Speed
                                 Dload  Upload
100   668  100   668    0     0   2891      0 --
100 21.3M  100 21.3M    0     0  15.3M      0  0
istio-1.10.3/
istio-1.10.3/manifests/
istio-1.10.3/manifests/charts/
istio-1.10.3/manifests/charts/istio-operator/
istio-1.10.3/manifests/charts/istio-operator/fil
istio-1.10.3/manifests/charts/istio-operator/fil
istio-1.10.3/manifests/charts/istio-operator/val
```

```
Restarting kubelet
DNS is enabled
✓ Istio core installed
✓ Istiod installed
✓ Ingress gateways installed
✓ Egress gateways installed
✓ Installation complete
Thank you for installing Istio 1.10.  Please
perience!  https://forms.gle/KjkrDnMPByq7akr
Istio is starting

To configure mutual TLS authentication consu
$
```

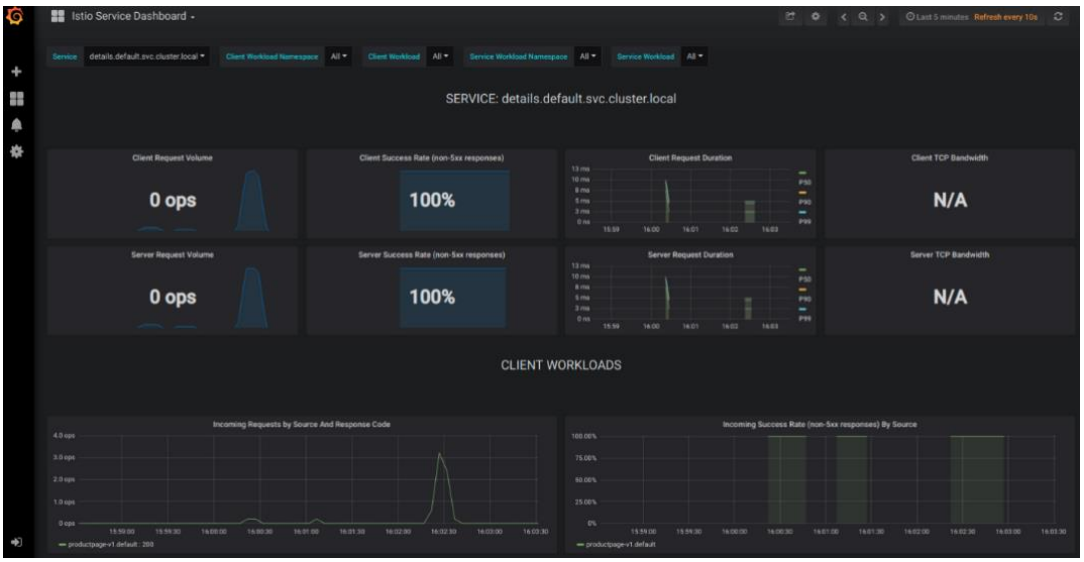
```
$ kubectl get pods -n istio-system
NAME                                    READY   STATUS
istiod-6f94fb9786-zfjn6                 1/1     Running
istio-ingressgateway-f9cd5d59d-kx9w8    1/1     Running
istio-egressgateway-77c5c9d46d-gfhk4    1/1     Running
$
```

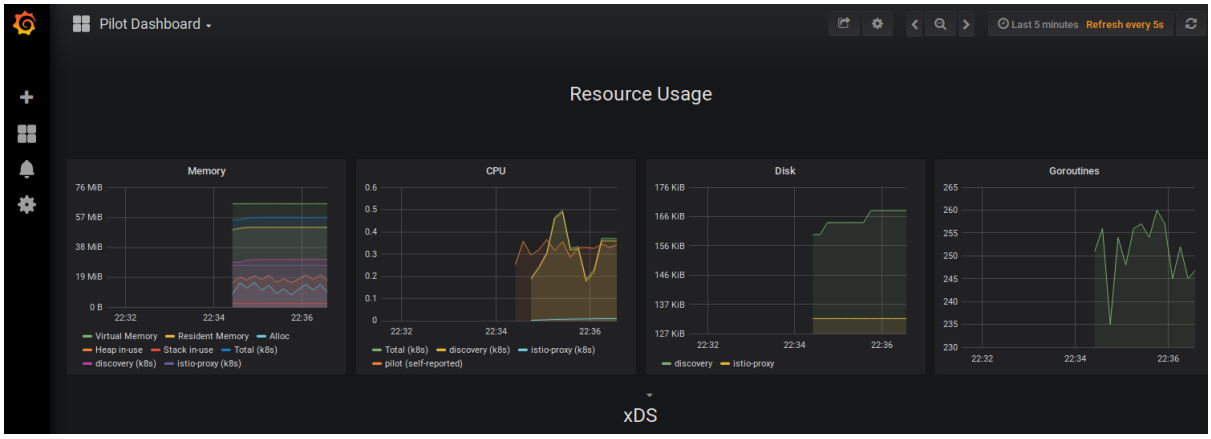
```
$ microk8s kubectl label namespace default istio-injection=enabled
namespace/default labeled
```

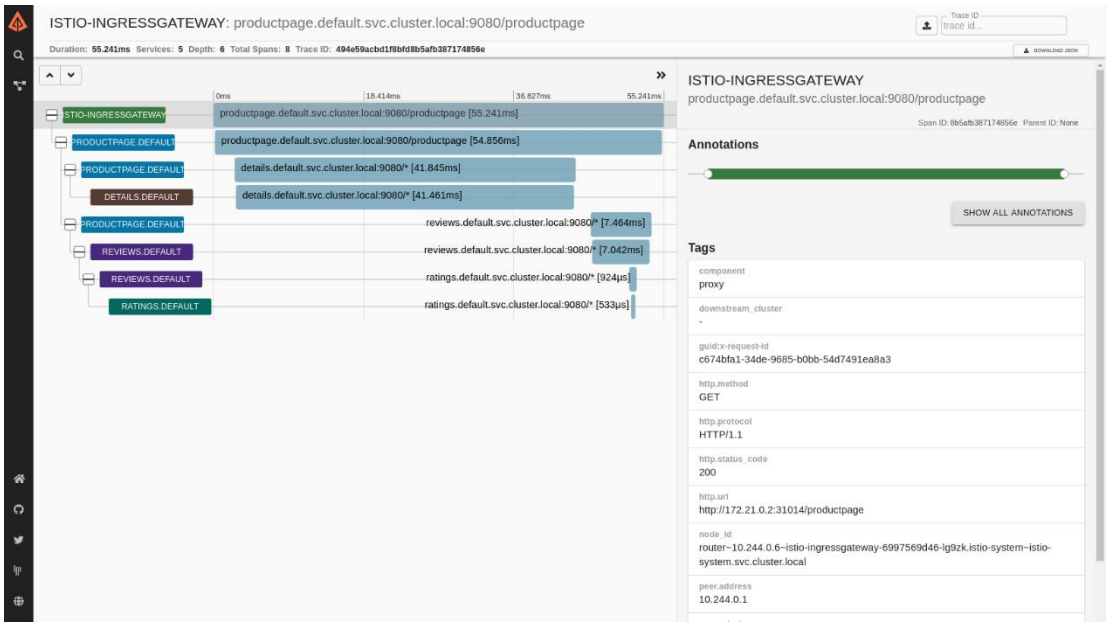
```
$ kubectl apply -f https://k8s.io/examples/application/deployment.yaml
deployment.apps/nginx-deployment created
$
```

```
$ kubectl describe pod nginx-deployment-9456bbbf9-w2ltq
Name:         nginx-deployment-9456bbbf9-w2ltq
Namespace:    default
Priority:     0
Node:         host01/10.0.0.21
Start Time:   Sun, 17 Apr 2022 12:39:51 +0000
Labels:       app=nginx
              istio.io/rev=default
              pod-template-hash=9456bbbf9
              security.istio.io/tlsMode=istio
              service.istio.io/canonical-name=nginx
              service.istio.io/canonical-revision=latest
Annotations:  cni.projectcalico.org/podIP: 10.1.239.199/32
              cni.projectcalico.org/podIPs: 10.1.239.199/32
              kubectl.kubernetes.io/default-container: ngin
              kubectl.kubernetes.io/default-logs-container:
              prometheus.io/path: /stats/prometheus
              prometheus.io/port: 15020
              prometheus.io/scrape: true
              sidecar.istio.io/status:
                {"initContainers":["istio-init"],"container
istio-data","istio-podinfo","istio-token","istiod-...
Status:       Running
```

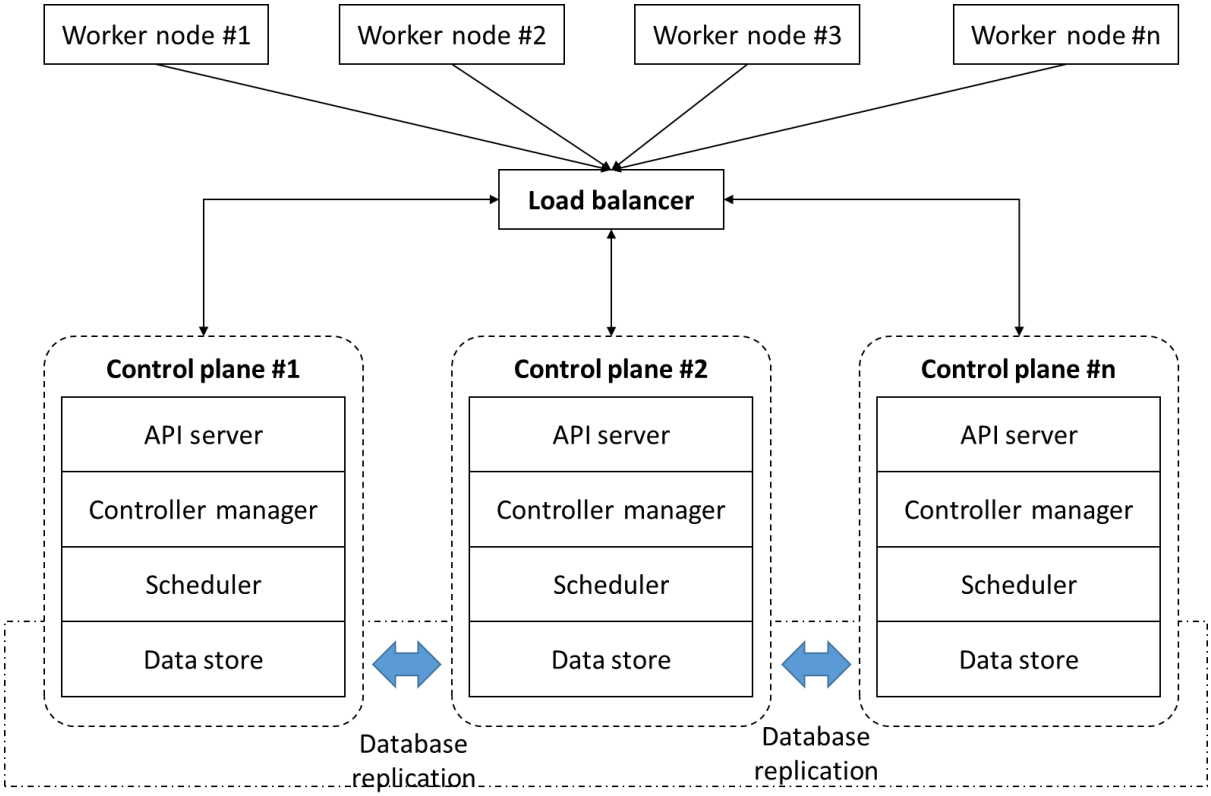
```
$ microk8s istioctl proxy-status
NAME                                                   CDS        LDS        EDS        RDS
OD                        VERSION
istio-egressgateway-77c5c9d46d-gfhk4.istio-system      SYNCED     SYNCED     SYNCED     NOT SENT
od-6f94fb9786-zfjn6       1.10.3
istio-ingressgateway-f9cd5d59d-kx9w8.istio-system      SYNCED     SYNCED     SYNCED     NOT SENT
od-6f94fb9786-zfjn6       1.10.3
nginx-deployment-9456bbbf9-8z6dm.default               SYNCED     SYNCED     SYNCED     SYNCED
od-6f94fb9786-zfjn6       1.10.3
nginx-deployment-9456bbbf9-w2ltq.default               SYNCED     SYNCED     SYNCED     SYNCED
od-6f94fb9786-zfjn6       1.10.3
$
```

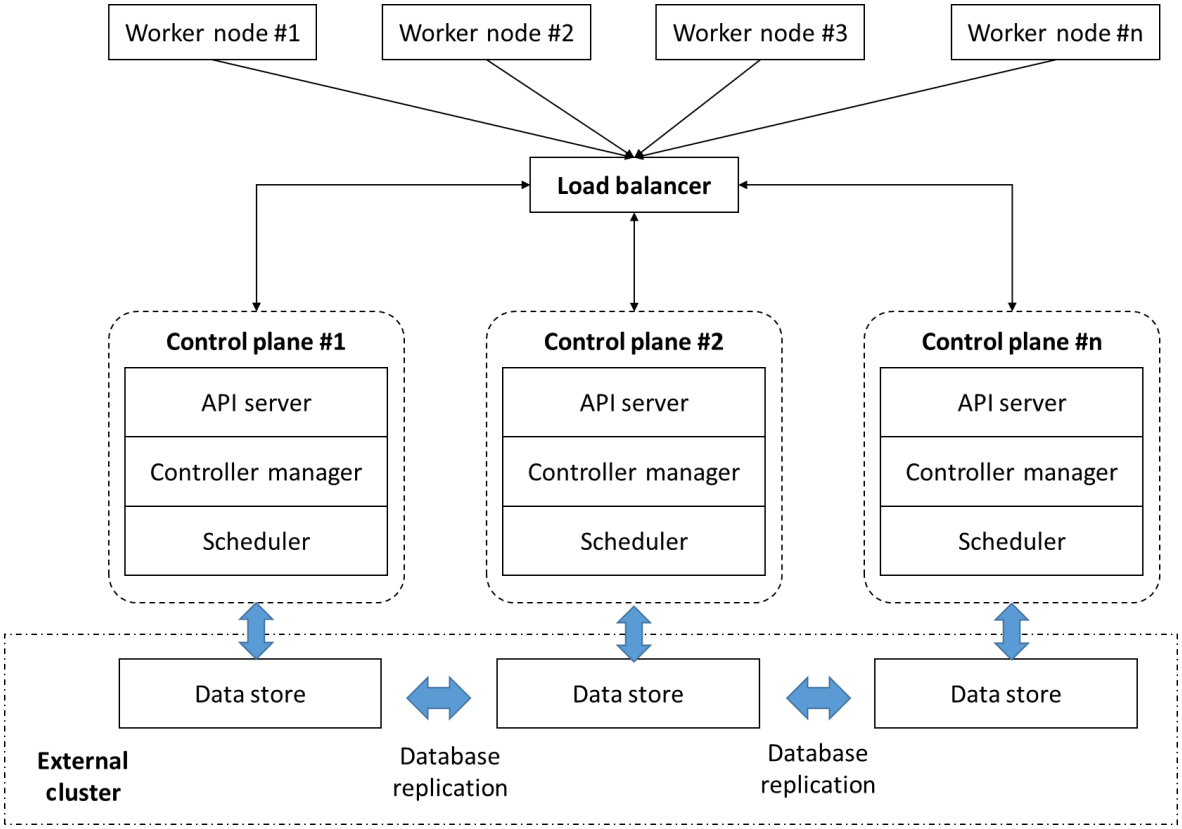


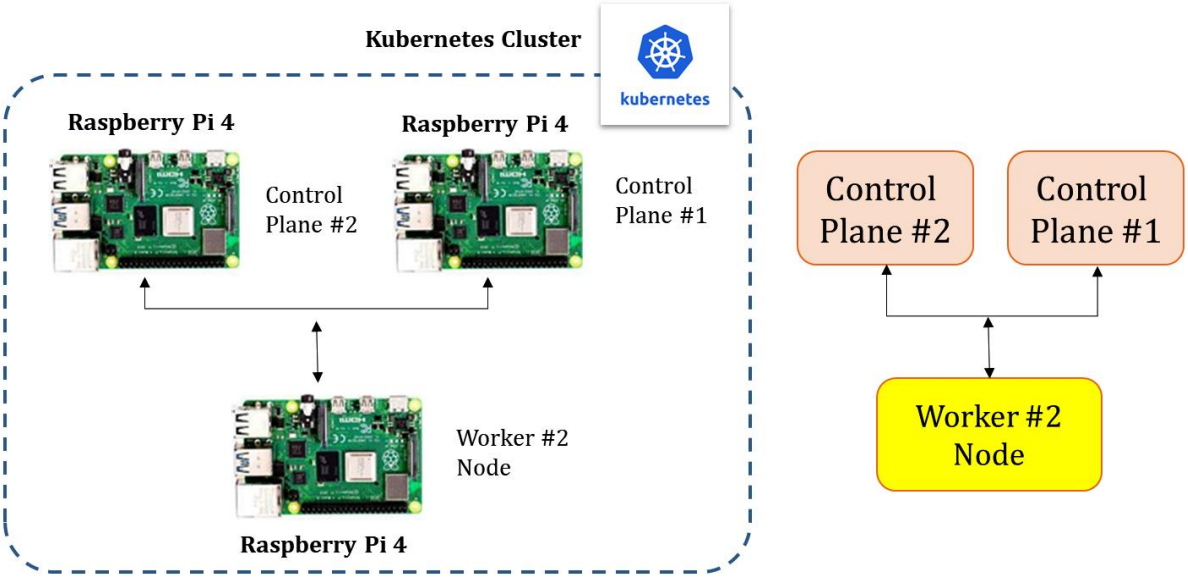

Pilot Dashboard
Last 5 minutes
Refresh every 5s
Resource Usage
Memory
76 MiB
57 MiB
38 MiB
19 MiB
0 B
22:32
22:34
22:36
Virtual Memory
Resident Memory
Alloc
Heap in-use
Stack in-use
Total (k8s)
discovery (k8s)
istio-proxy (k8s)
CPU
0.6
0.5
0.4
0.3
0.2
0.1
0
Total (k8s)
discovery (k8s)
istio-proxy (k8s)
pilot (self-reported)
Disk
176 KiB
166 KiB
156 KiB
146 KiB
137 KiB
127 KiB
discovery
istio-proxy
Goroutines
265
260
255
250
245
240
235
230
xDS


ISTIO-INGRESSGATEWAY: productpage.default.svc.cluster.local:9080/productpage
Duration: 55.241ms Services: 5 Depth: 6 Total Spans: 8 Trace ID: 494e59acbd1f8bfd8b5afb387174856e
Trace ID
trace id...
DOWNLOAD JSON
0ms
18.414ms
36.827ms
55.241ms
ISTIO-INGRESSGATEWAY
productpage.default.svc.cluster.local:9080/productpage [55.241ms]
PRODUCTPAGE.DEFAULT
productpage.default.svc.cluster.local:9080/productpage [54.856ms]
PRODUCTPAGE.DEFAULT
details.default.svc.cluster.local:9080/* [41.845ms]
DETAILS.DEFAULT
details.default.svc.cluster.local:9080/* [41.461ms]
PRODUCTPAGE.DEFAULT
reviews.default.svc.cluster.local:9080/* [7.464ms]
REVIEWS.DEFAULT
reviews.default.svc.cluster.local:9080/* [7.042ms]
REVIEWS.DEFAULT
ratings.default.svc.cluster.local:9080/* [924µs]
RATINGS.DEFAULT
ratings.default.svc.cluster.local:9080/* [533µs]
ISTIO-INGRESSGATEWAY
productpage.default.svc.cluster.local:9080/productpage
Span ID: 8b5afb387174856e Parent ID: None
Annotations
SHOW ALL ANNOTATIONS
Tags
component
proxy
downstream_cluster
-
guid:x-request-id
c674bfa1-34de-9685-b0bb-54d7491ea8a3
http.method
GET
http.protocol
HTTP/1.1
http.status_code
200
http.url
http://172.21.0.2:31014/productpage
node_id
router~10.244.0.6~istio-ingressgateway-6997569d46-lg9zk.istio-system~istio-system.svc.cluster.local
peer.address
10.244.0.1

Chapter 13: Resisting Component Failure Using HA Clusters








```
ubuntu@controlplane:~$ sudo snap install microk8s --classic
microk8s (1.23/stable) v1.23.3 from Canonical✓ installed
ubuntu@controlplane:~$
```

```
ubuntu@controlplane:~$ microk8s status
microk8s is running
high-availability: no
  datastore master nodes: 127.0.0.1:19001
  datastore standby nodes: none
addons:
  enabled:
    ha-cluster           # Configure high av
  disabled:
    dashboard            # The Kubernetes da
    dashboard-ingress    # Ingress definitio
```

```
ubuntu@controlplane:~$ kubectl get nodes
NAME           STATUS   ROLES    AGE   VERSION
controlplane   Ready    <none>   12m   v1.23.3-2+0d2db09fa6fbbb
ubuntu@controlplane:~$
```

```
ubuntu@controlplane:~$ sudo microk8s.add-node
From the node you wish to join to this cluster, run the following:
microk8s join 192.168.1.7:25000/fba12c2f1bce9fbe70208443565aaa04/3e2f115c73d6

Use the '--worker' flag to join a node as a worker not running the control plane,
microk8s join 192.168.1.7:25000/fba12c2f1bce9fbe70208443565aaa04/3e2f115c73d6 --wo

If the node you are adding is not reachable through the default interface you can
he following:
microk8s join 192.168.1.7:25000/fba12c2f1bce9fbe70208443565aaa04/3e2f115c73d6
ubuntu@controlplane:~$
```

```
ubuntu@controlplane1:~$ sudo microk8s join 192.168.1.7:25000/fba12c2f1bce9fbe
65aaa04/3e2f115c73d6
Contacting cluster at 192.168.1.7

The node has joined the cluster and will appear in the nodes list in a few se
ds.

Currently this worker node is configured with the following kubernetes API se
r endpoints:
    - 192.168.1.7 and port 16443, this is the cluster node contacted during
join operation.

If the above endpoints are incorrect, incomplete or if the API servers are be
d a loadbalancer please update
/var/snap/microk8s/current/args/traefik/provider.yaml
```

```
ubuntu@worker2:~$ sudo microk8s join 192.168.1.7:25000/fba12c2f1bce9fbe702084435
65aaa04/3e2f115c73d6
Contacting cluster at 192.168.1.7

The node has joined the cluster and will appear in the nodes list in a few secon
ds.

Currently this worker node is configured with the following kubernetes API serve
r endpoints:
    - 192.168.1.7 and port 16443, this is the cluster node contacted during the
join operation.

If the above endpoints are incorrect, incomplete or if the API servers are behin
d a loadbalancer please update
/var/snap/microk8s/current/args/traefik/provider.yaml
```

```
ubuntu@controlplane:~$ microk8s kubectl get no
NAME            STATUS   ROLES    AGE     VERSION
controlplane    Ready    <none>   5h12m   v1.24.0-2+f76e51e86eadea
controlplane1   Ready    <none>   4h23m   v1.24.0-2+f76e51e86eadea
worker2         Ready    <none>   13m     v1.24.0-2+59bbb3530b6769
ubuntu@controlplane:~$
```

```
ubuntu@controlplane:~$ microk8s status
microk8s is running
high-availability: yes
  datastore master nodes: 192.168.1.7:19001 192.168.1.6:19001 192.168.1.8:19001
  datastore standby nodes: none
addons:
  enabled:
    dns                  # (core) CoreDNS
    ha-cluster           # (core) Configure high availability on the current no
  disabled:
    community            # (core) The community addons repository
```

```
ubuntu@controlplane:~$ kubectl apply -f https://k8s.io/examples/controllers/nginx-deployment.yaml
deployment.apps/nginx-deployment created
ubuntu@controlplane:~$
```

```
ubuntu@controlplane:~$ kubectl get deployments
NAME               READY   UP-TO-DATE   AVAILABLE   AGE
nginx-deployment   3/3     3            3           4m7s
ubuntu@controlplane:~$
```

```
ubuntu@controlplane:~$ kubectl get pod -o=custom-columns=NODE:.spec.nodeName,NAME:.metadata.name
NODE           NAME
controlplane   nginx-deployment-6595874d85-k44jr
controlplane1  nginx-deployment-6595874d85-d9zbd
worker2        nginx-deployment-6595874d85-kql6z
ubuntu@controlplane:~$
```

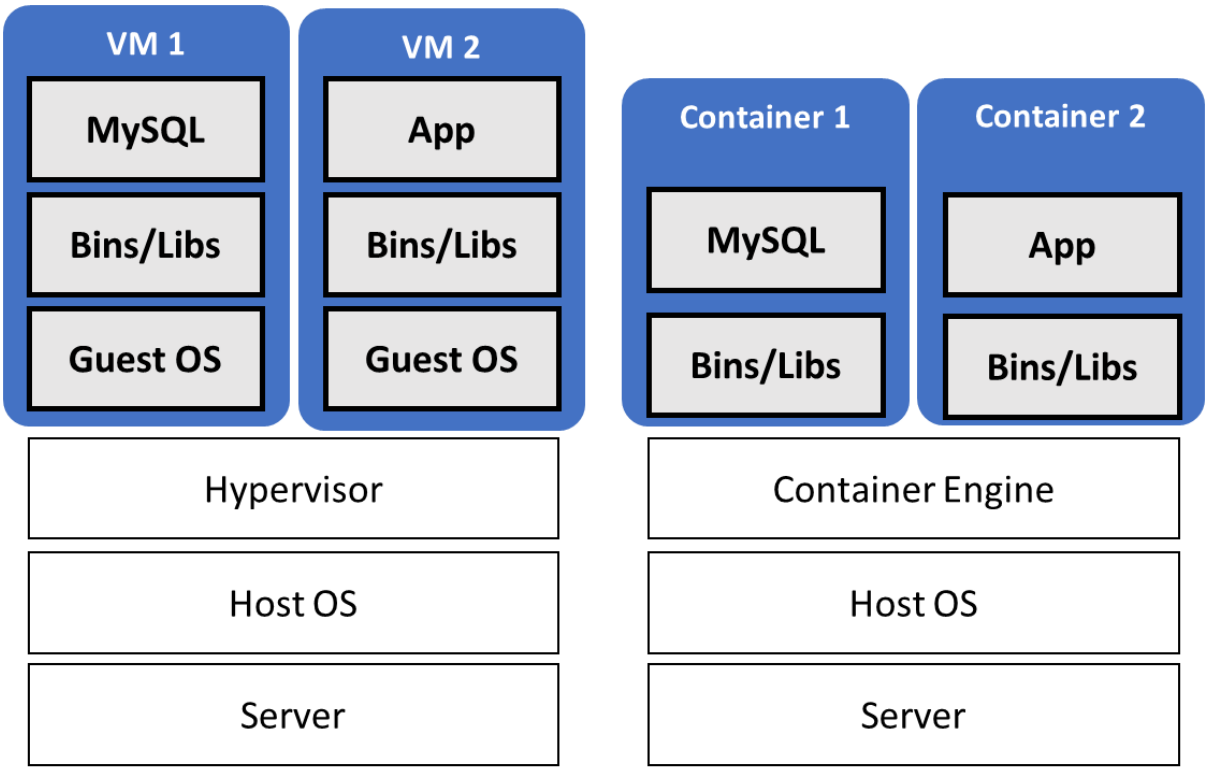
```
ubuntu@controlplane:~$ kubectl cordon controlplane1
node/controlplane1 cordoned
ubuntu@controlplane:~$
```

```
ubuntu@controlplane:~$ kubectl drain controlplane1 --ignore-daemonsets
node/controlplane1 already cordoned
WARNING: ignoring DaemonSet-managed Pods: kube-system/calico-node-gkxps
evicting pod default/nginx-deployment-6595874d85-d9zbd
pod/nginx-deployment-6595874d85-d9zbd evicted
node/controlplane1 drained
ubuntu@controlplane:~$
```

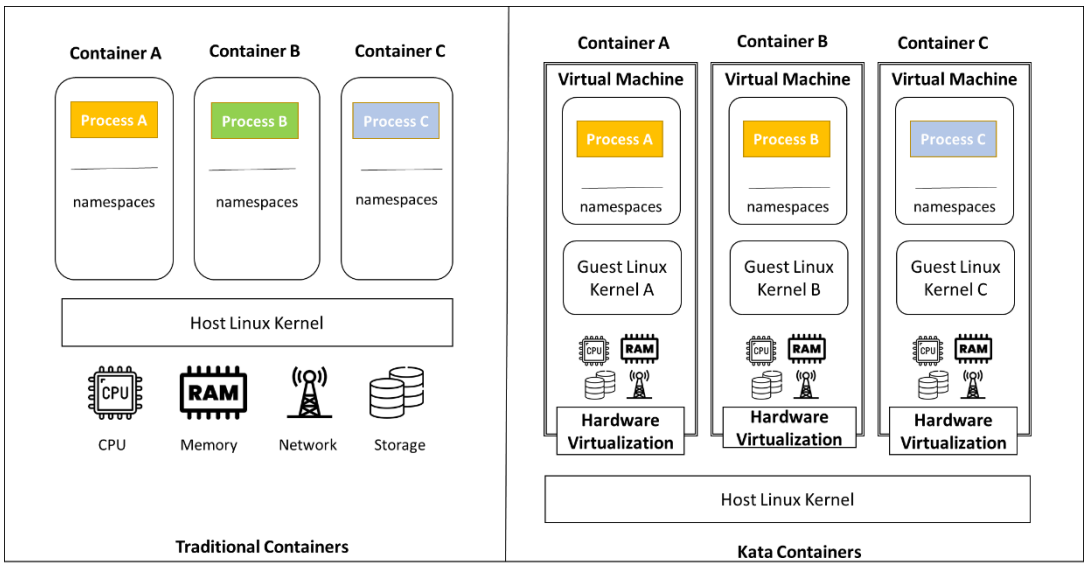
```
ubuntu@controlplane:~$ kubectl get pod -o=custom-columns=NODE:.spec.nodeName,NAME:.metadata.name
NODE           NAME
controlplane   nginx-deployment-6595874d85-k44jr
worker2        nginx-deployment-6595874d85-kql6z
controlplane   nginx-deployment-6595874d85-kptt2
ubuntu@controlplane:~$
```

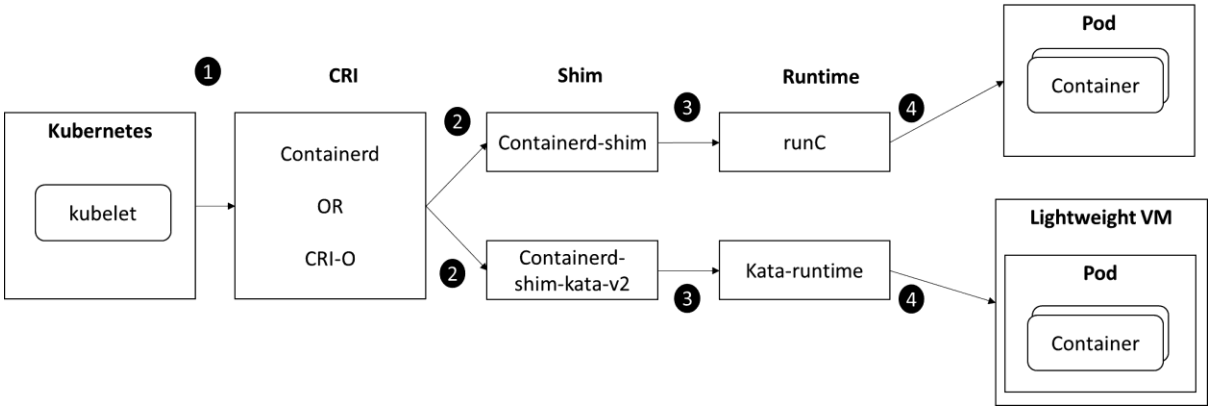
```
ubuntu@controlplane:~$ kubectl get deployments
NAME               READY   UP-TO-DATE   AVAILABLE   AGE
nginx-deployment   3/3     3            3           8m55s
ubuntu@controlplane:~$
```

Chapter 14: Hardware Virtualization for Securing Containers



Virtual Machines **Containers**






```
$ microk8s enable community
Infer repository core for addon community
Cloning into '/var/snap/microk8s/common/addons/community'...
done.
Community repository is now enabled
$
```

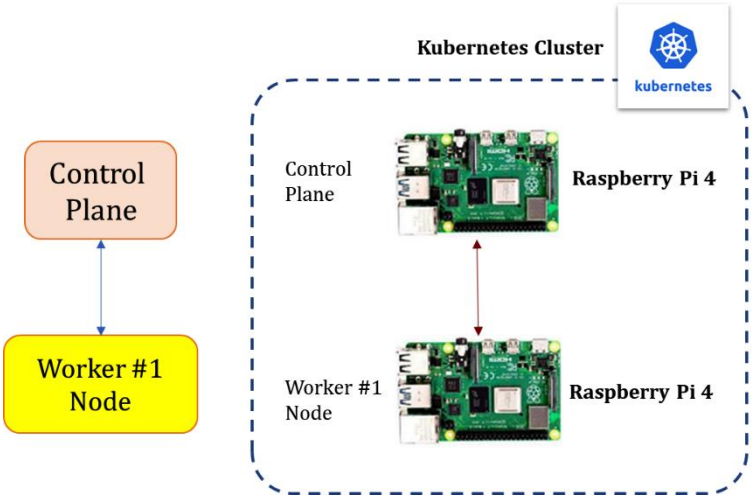
```
$ microk8s enable kata
Infer repository community for addon kata
Installing kata-containers snap
kata-containers 2.4.1 from Katacontainers (katacontainers✓) installed
Restarting containerd
Warning: node.k8s.io/v1beta1 RuntimeClass is deprecated in v1.22+, unavailable in v1.25+
runtimeclass.node.k8s.io/kata created


To use the kata runtime set the 'kata' runtimeClassName, eg:

kind: Pod
metadata:
  name: nginx-kata
spec:
  runtimeClassName: kata
  containers:
  - name: nginx
    image: nginx

$
```

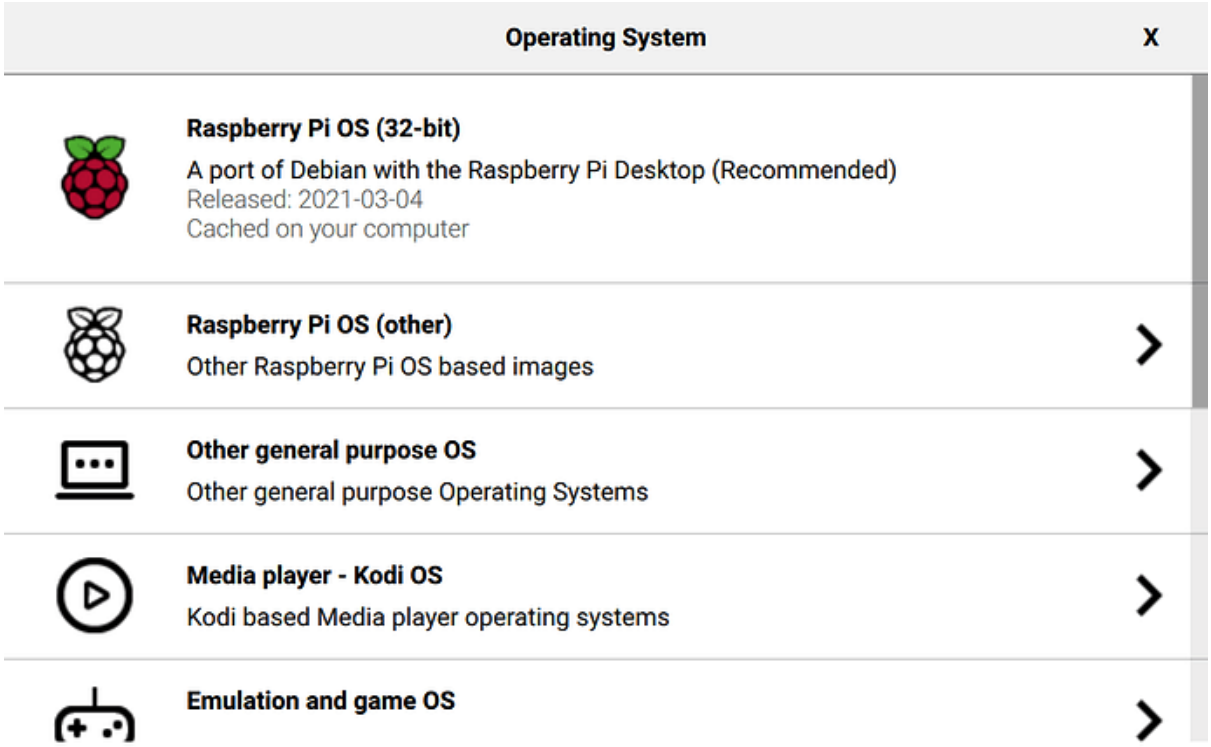
```
$ microk8s status
microk8s is running
high-availability: no
  datastore master nodes: 127.0.0.1:19001
  datastore standby nodes: none
addons:
  enabled:
    kata                 # (community) Kata Containers is a s
    community            # (core) The community addons reposi
    ha-cluster           # (core) Configure high availability
```

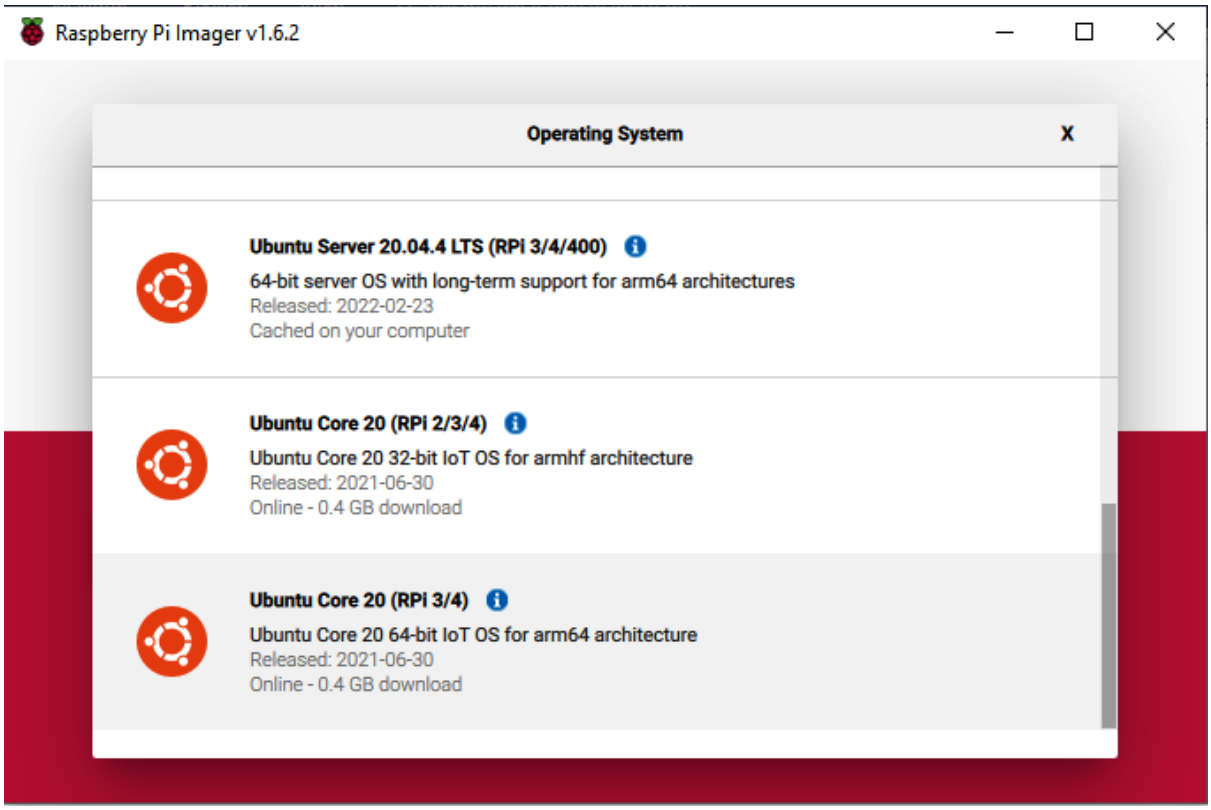
```
$ kubectl apply -f kata-nginx.yaml
pod/kata-nginx created
$
```

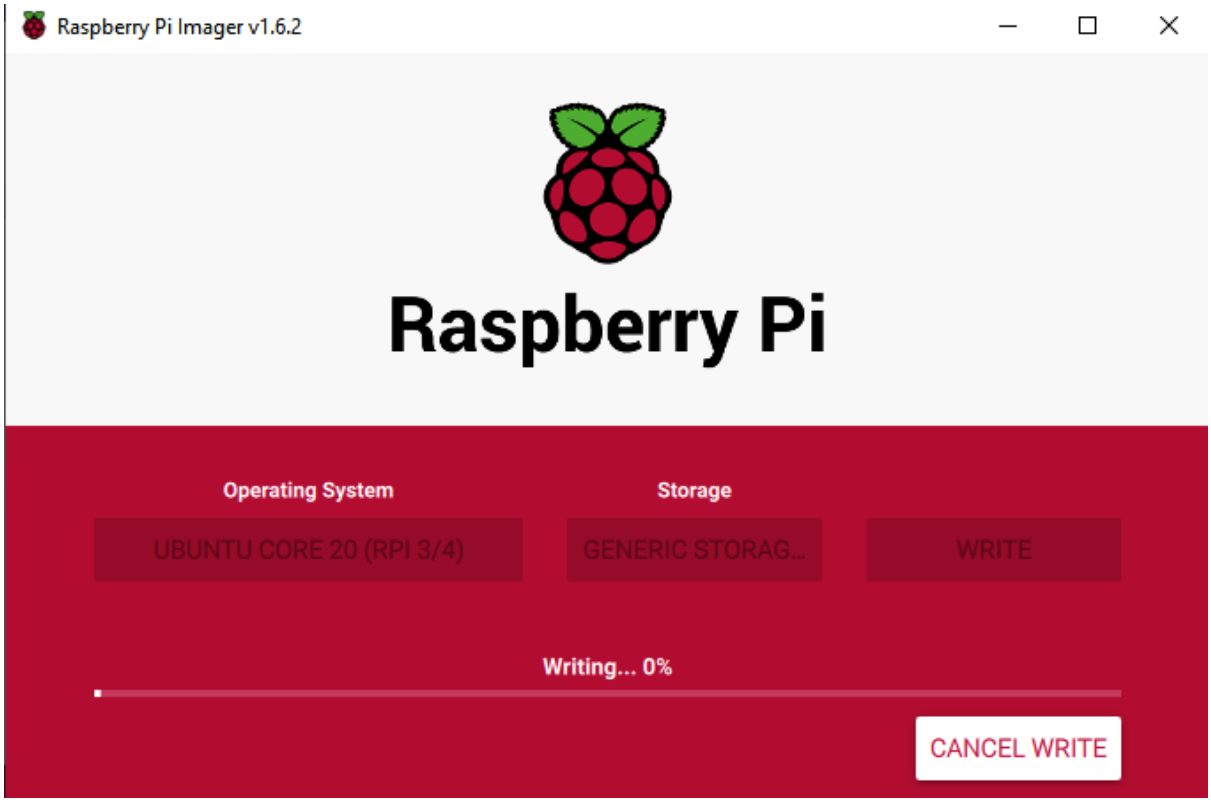
```
$ kubectl get pods
NAME          READY   STATUS    RESTARTS   AGE
nginx-kata    1/1     Running   0          12s
$
```

Chapter 15: Implementing Strict Confi nement for Isolated Containers




Operating System
X
Raspberry Pi OS (32-bit)
A port of Debian with the Raspberry Pi Desktop (Recommended)
Released: 2021-03-04
Cached on your computer
Raspberry Pi OS (other)
Other Raspberry Pi OS based images
Other general purpose OS
Other general purpose Operating Systems
Media player - Kodi OS
Kodi based Media player operating systems
Emulation and game OS


Raspberry Pi Imager v1.6.2
Operating System
X
Ubuntu Server 20.04.4 LTS (RPi 3/4/400)
64-bit server OS with long-term support for arm64 architectures
Released: 2022-02-23
Cached on your computer
Ubuntu Core 20 (RPi 2/3/4)
Ubuntu Core 20 32-bit IoT OS for armhf architecture
Released: 2021-06-30
Online - 0.4 GB download
Ubuntu Core 20 (RPi 3/4)
Ubuntu Core 20 64-bit IoT OS for arm64 architecture
Released: 2021-06-30
Online - 0.4 GB download




```
PS C:\WINDOWS\system32> ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (C:\Users\Admin/.ssh/id_rsa): ubuntu-core-rpi
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in ubuntu-core-rpi.
Your public key has been saved in ubuntu-core-rpi.pub.
The key fingerprint is:
SHA256:pYBRIoeikEJQh9b70eZN62Vq+A0K5fzzXXWwE61MPmo admin@DESKTOP-5RAQJ1L
The key's randomart image is:
+---[RSA 3072]----+
|++o+=..          |
|= ++.+         . |
|+o  ....  .   + .|
|.   . ..oo.  + = |
|      . +So .  B o|
|       .+. o o. oo|
|       . oo.+E   .|
|        ..o=+ . . |
|         .oooo .  |
+----[SHA256]-----+
PS C:\WINDOWS\system32>
```

Name	Date modified
ubuntu-core-rpi	6/20/2022 10:46 AM
ubuntu-core-rpi.pub	6/20/2022 10:46 AM

ubuntu one

Karthik Log out

My account

- Personal details
- Authentication devices
- Applications
- SSH keys
- Account activity

SSH keys

Import new SSH key

Insert the contents of your public key (usually `~/.ssh/id_rsa.pub`, `~/.ssh/id_dsa.pub`, `~/.ssh/id_ecdsa.pub` or `~/.ssh/id_ed25519.pub`).

Note: Only SSH v2 keys are supported.

Public SSH Key:

mBMOkN8L+hbNUt8cW15RlQdRGr2NX+LyvgUoLaT+Hp+YLsWqxPlxfztIObA4s10rt3xAje8Uv1Oy1NGokCts+Rn
qF4H2GMJUqBS6PXb0OnE2J9lfStrBFv/l1oFJAe4fiyNzbqqiqJ5zcARX1dP/G83+qmzXvVZUVrlYo7nhMtNZWniqz
Sgvv8exw/o8lyzoMQISs= admin@DESKTOP-5RAQJ1L

ubuntu one

My account

- Personal details
- Authentication devices
- Applications
- SSH keys
- Account activity

The following key was added to your account: `admin@DESKTOP-5RAQJ1L`

SSH keys

admin@DESKTOP-5RAQJ1L

+ Key details

Delete selected keys

```
ubuntu@controlplane:~$ sudo snap install microk8s --channel=latest/edge/strict
2022-06-21T06:05:14Z INFO Waiting for automatic snapd restart...
microk8s (edge/strict) v1.24.1 from Canonical✓ installed
ubuntu@controlplane:~$
```

```
ubuntu@controlplane:~$ microk8s status
microk8s is running
high-availability: no
  datastore master nodes: 127.0.0.1:19001
  datastore standby nodes: none
addons:
  enabled:
    ha-cluster           # Configure high av
  disabled:
    dashboard            # The Kubernetes da
    dashboard-ingress    # Ingress definitio
```

```
root@master:~# snap connections microk8s
Interface              Plug                          Slot                  Notes
account-control        microk8s:account-control      :account-control      -
cifs-mount             microk8s:cifs-mount           -                     -
content                -                             microk8s:microk8s     -
docker-support         microk8s:docker-privileged    :docker-support       -
docker-support         microk8s:docker-unprivileged  :docker-support       -
firewall-control       microk8s:firewall-control     :firewall-control     -
fuse-support           microk8s:fuse-support         -                     -
hardware-observe       microk8s:hardware-observe     :hardware-observe     -
home                   microk8s:home                 :home                 -
home                   microk8s:home-read-all        :home                 -
kernel-crypto-api      microk8s:kernel-crypto-api    -                     -
```

```
ubuntu@controlplane:~$ kubectl get nodes
NAME            STATUS   ROLES    AGE   VERSION
controlplane    Ready    <none>   12m   v1.23.3-2+0d2db09fa6fbbb
ubuntu@controlplane:~$
```

```
 channels:
   1.24/stable:      v1.24.0  2022-05-13 (3272) 230MB classic
   1.24/candidate:   v1.24.0  2022-05-13 (3272) 230MB classic
   1.24/beta:        v1.24.0  2022-05-13 (3272) 230MB classic
   1.24/edge:        v1.24.2  2022-06-20 (3475) 230MB classic
   latest/stable:    v1.24.0  2022-05-13 (3272) 230MB classic
   latest/candidate: v1.24.0  2022-05-13 (3273) 230MB classic
   latest/beta:      v1.24.0  2022-05-13 (3273) 230MB classic
   latest/edge:      v1.24.2  2022-06-21 (3479) 230MB classic
   dqlite/stable:    -
   dqlite/candidate: -
```

```
ubuntu@worker1:~$ sudo snap install microk8s --channel=latest/edge/strict
microk8s (edge/strict) v1.24.1 from Canonical✓ installed
ubuntu@worker1:~$
```

```
ubuntu@worker1:~$ microk8s status
microk8s is running
high-availability: no
  datastore master nodes: 127.0.0.1:19001
  datastore standby nodes: none
addons:
  enabled:
    ha-cluster           # Configure high av
  disabled:
    dashboard            # The Kubernetes da
    dashboard-ingress    # Ingress definitio
```

```
ubuntu@worker1:~$ kubectl get nodes
NAME       STATUS   ROLES    AGE   VERSION
worker1    Ready    <none>   13m   v1.23.3-2+0d2db09fa6fbbb
ubuntu@worker1:~$
```

```
ubuntu@controlplane:~$ sudo microk8s.add-node
From the node you wish to join to this cluster, run the following:
microk8s join 192.168.1.7:25000/fba12c2f1bce9fbe70208443565aaa04/3e2f115c73d6

Use the '--worker' flag to join a node as a worker not running the control plane,
microk8s join 192.168.1.7:25000/fba12c2f1bce9fbe70208443565aaa04/3e2f115c73d6 --wo

If the node you are adding is not reachable through the default interface you can
he following:
microk8s join 192.168.1.7:25000/fba12c2f1bce9fbe70208443565aaa04/3e2f115c73d6
ubuntu@controlplane:~$
```

```
ubuntu@worker1:~$ sudo microk8s join 192.168.1.7:25000/fba12c2f1bce9fbe702084435
65aaa04/3e2f115c73d6 --worker
Contacting cluster at 192.168.1.7

The node has joined the cluster and will appear in the nodes list in a few secon
ds.

Currently this worker node is configured with the following kubernetes API serve
r endpoints:
    - 192.168.1.7 and port 16443, this is the cluster node contacted during the
join operation.

If the above endpoints are incorrect, incomplete or if the API servers are behin
d a loadbalancer please update
/var/snap/microk8s/current/args/traefik/provider.yaml

ubuntu@worker1:~$
```

```
ubuntu@controlplane:~$ kubectl get nodes
NAME           STATUS   ROLES    AGE   VERSION
controlplane   Ready    <none>   17m   v1.23.3-2+0d2db09fa6fbbb
worker1        Ready    <none>   27s   v1.23.3-2+0d2db09fa6fbbb
ubuntu@controlplane:~$
```

```
ubuntu@controlplane:~$ kubectl apply -f https://k8s.io/examples/controllers/nginx-deployment.yaml
deployment.apps/nginx-deployment created
ubuntu@controlplane:~$
```

```
ubuntu@controlplane:~$ kubectl get pods -l app=nginx
NAME                                READY   STATUS    RESTARTS   AGE
nginx-deployment-57d554699f-clxd5   1/1     Running   0          3m9s
nginx-deployment-57d554699f-8hjbv   1/1     Running   0          3m9s
ubuntu@controlplane:~$
```

Chapter 16: Diving into the Future

No images…