
digikala

Kubernetes Course Session - 1

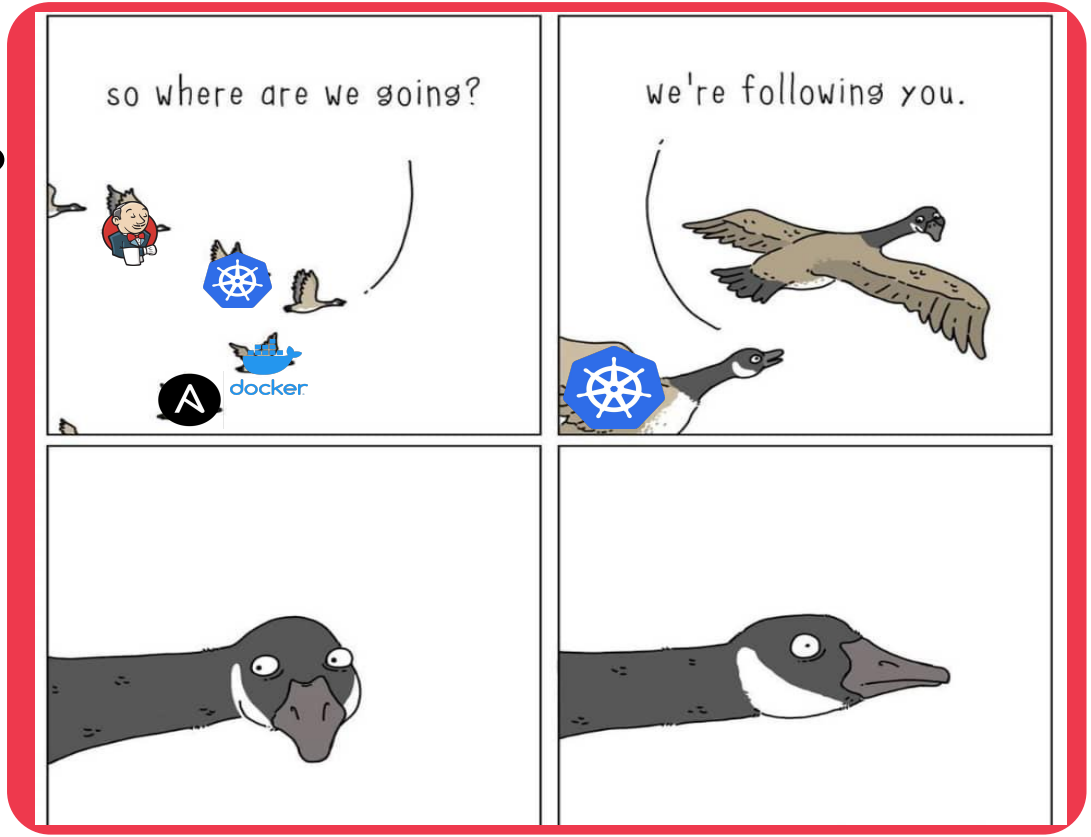
Contents

Main content

- Why kubernetes
- Learning basics of containers with docker & docker compose
- Learning basics of application deployment on kubernetes

Why kubernetes?

- What is Kubernetes?
- Why Kubernetes?
- Where to use
- kubernetes?
- Kubernetes vs …



What is kubernetes? Search it!

Kubernetes, also known as K8s, is an open-source system for automating deployment, scaling, and management of containerized applications.

Kubernetes (/ˌkuːbərˈnɛtɪs, -ˈnɛtɪs, -ˈnɛtiːz, -ˈnɛtiːz/, commonly abbreviated **K8s**[3]) is an open-source container orchestration system for automating software deployment, scaling, and management.[4][5] Originally designed by Google, the project is now maintained by the Cloud Native Computing Foundation.

The name *Kubernetes* originates from Ancient Greek, meaning 'helmsman' or 'pilot'. *Kubernetes* is often abbreviated as *K8s*, counting the eight letters between the *K* and the *s* (a numeronym).[6]

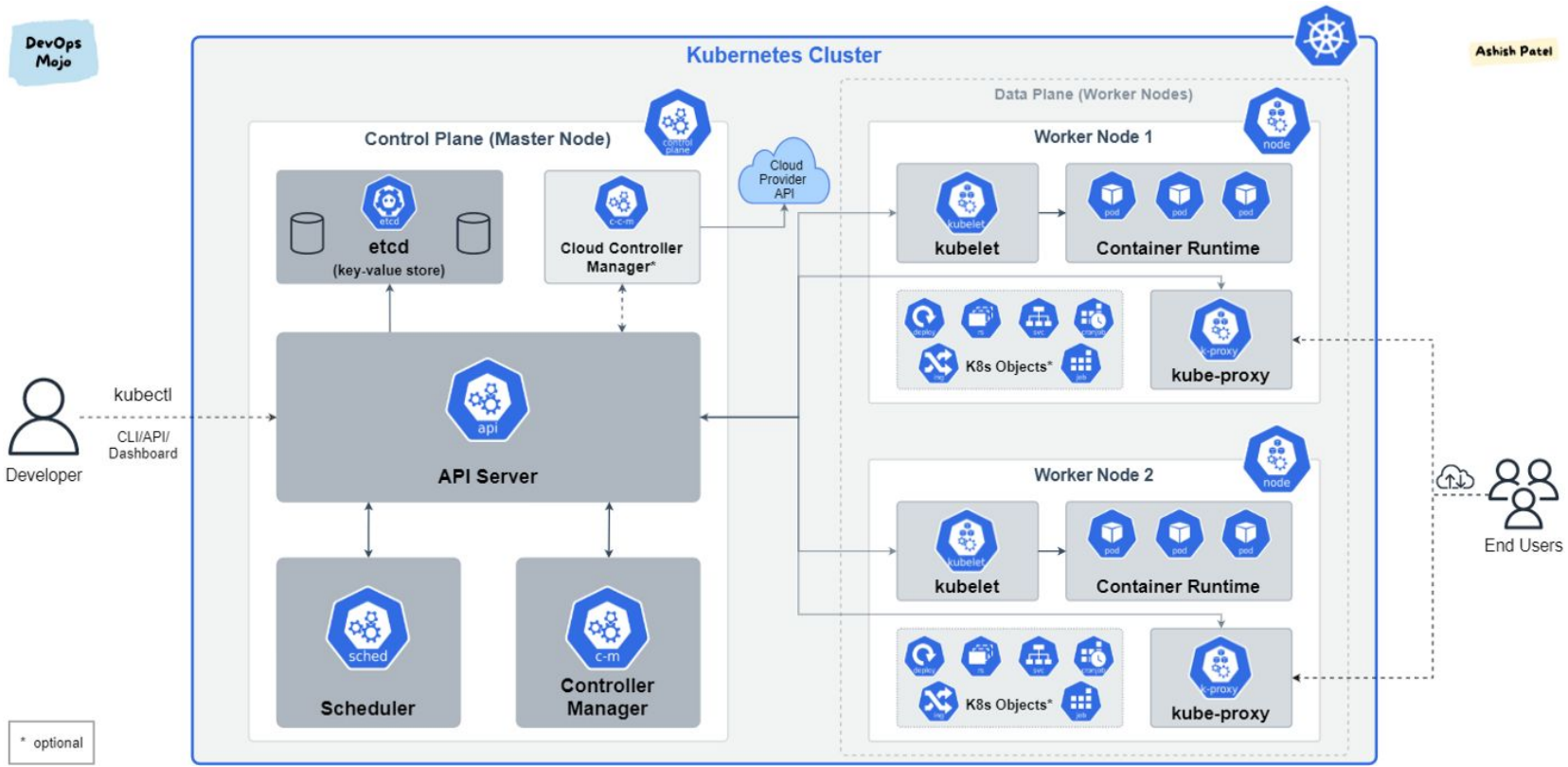
Kubernetes works with various container runtimes, such as containerd and CRI-O.[7] Its suitability for running and managing large cloud-native workloads has led to its widespread adoption in the data center. There are multiple distributions of this platform – from independent software vendors (ISVs) as well as hosted-on-cloud offerings from all the major public cloud vendors.[8]

Overview

Kubernetes (also known as k8s or “kube”) is an open source container orchestration platform that automates many of the manual processes involved in deploying, managing, and scaling containerized applications.

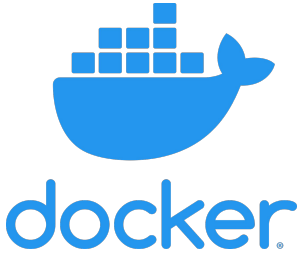
Originally developed and designed by engineers at Google as the Borg project, Kubernetes was donated to the Cloud Native Computing Foundation (CNCF) in 2015. Red Hat® was one of the first companies to work with Google on Kubernetes, even prior to launch, and has become the 2nd-most leading contributor to the Kubernetes upstream project.

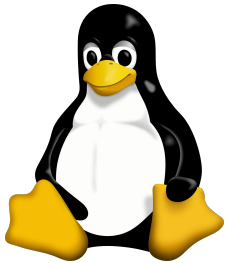
How kubernetes works?



What?!

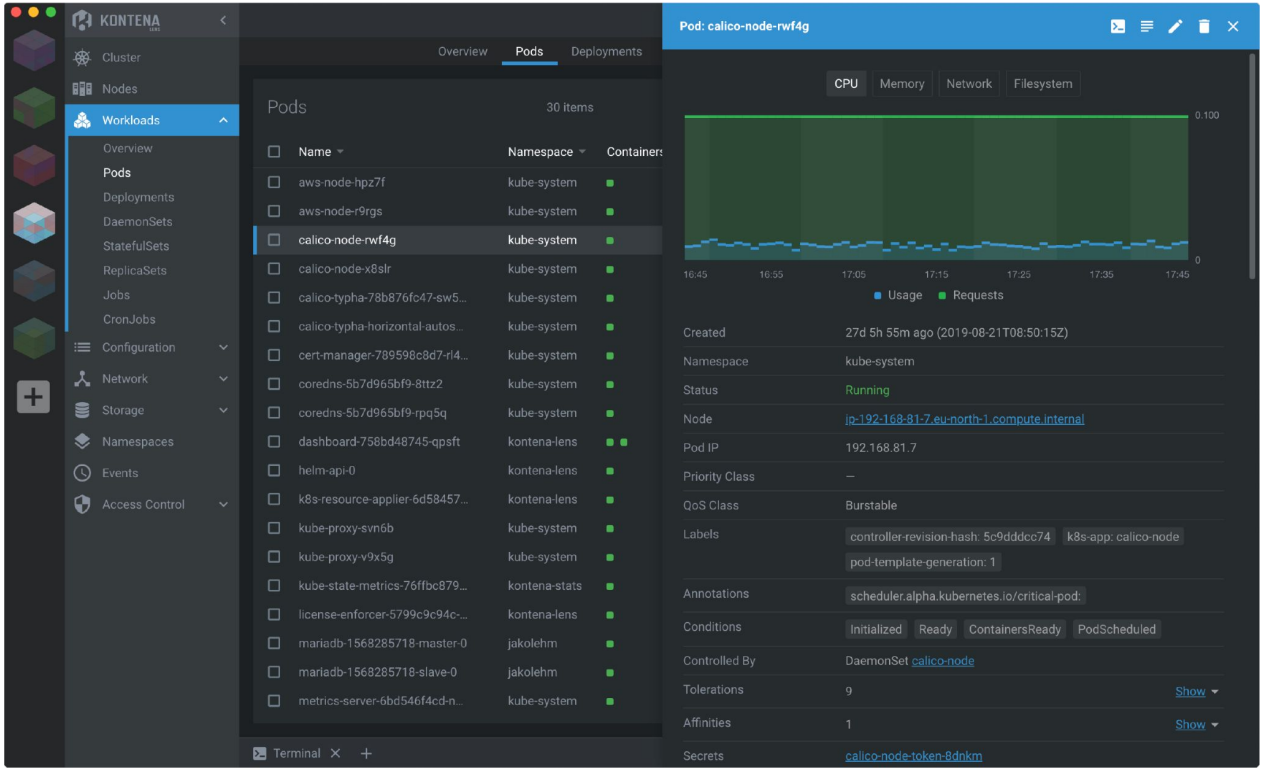
Pre Requirements





What is forbidden during the course?

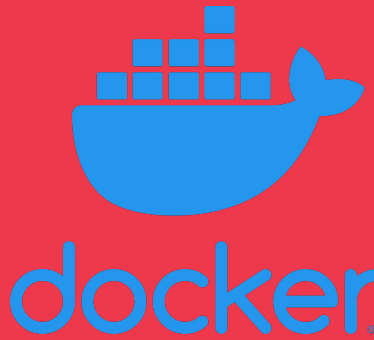
Any GUI utils!
Specially lens



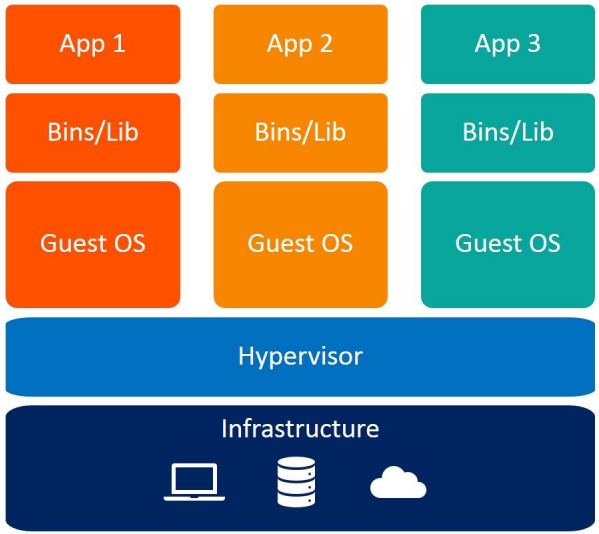
Deploy application

Using docker

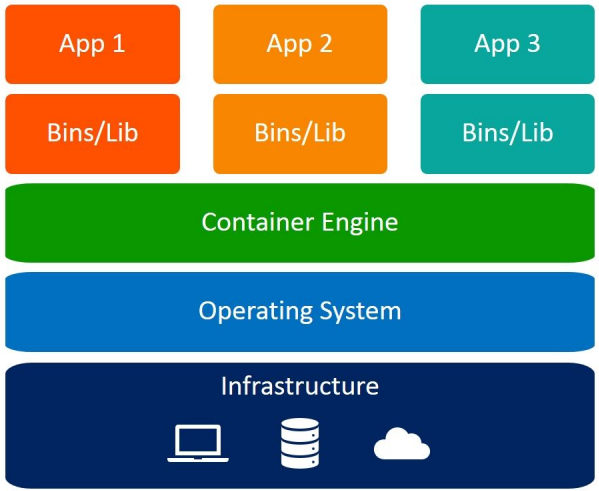
find out why docker compose



Full virtualization vs Shared kernel



Virtual Machines

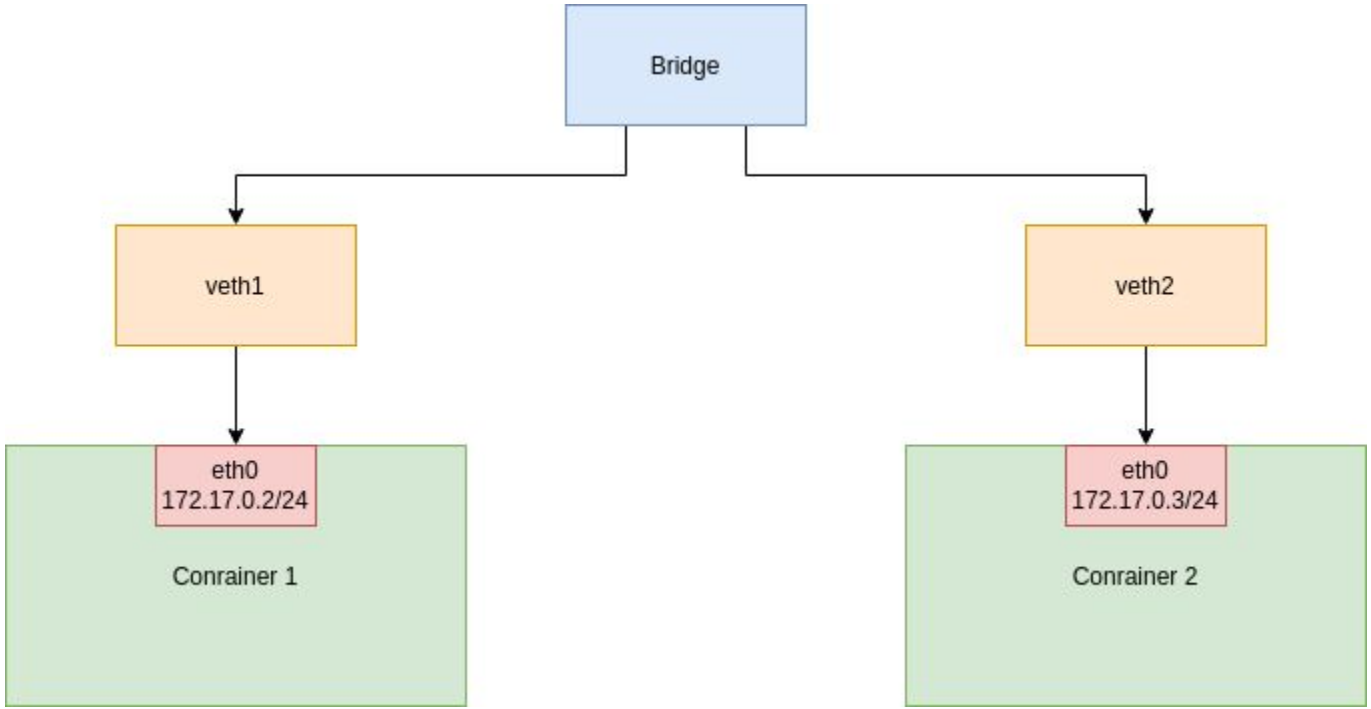


Containers

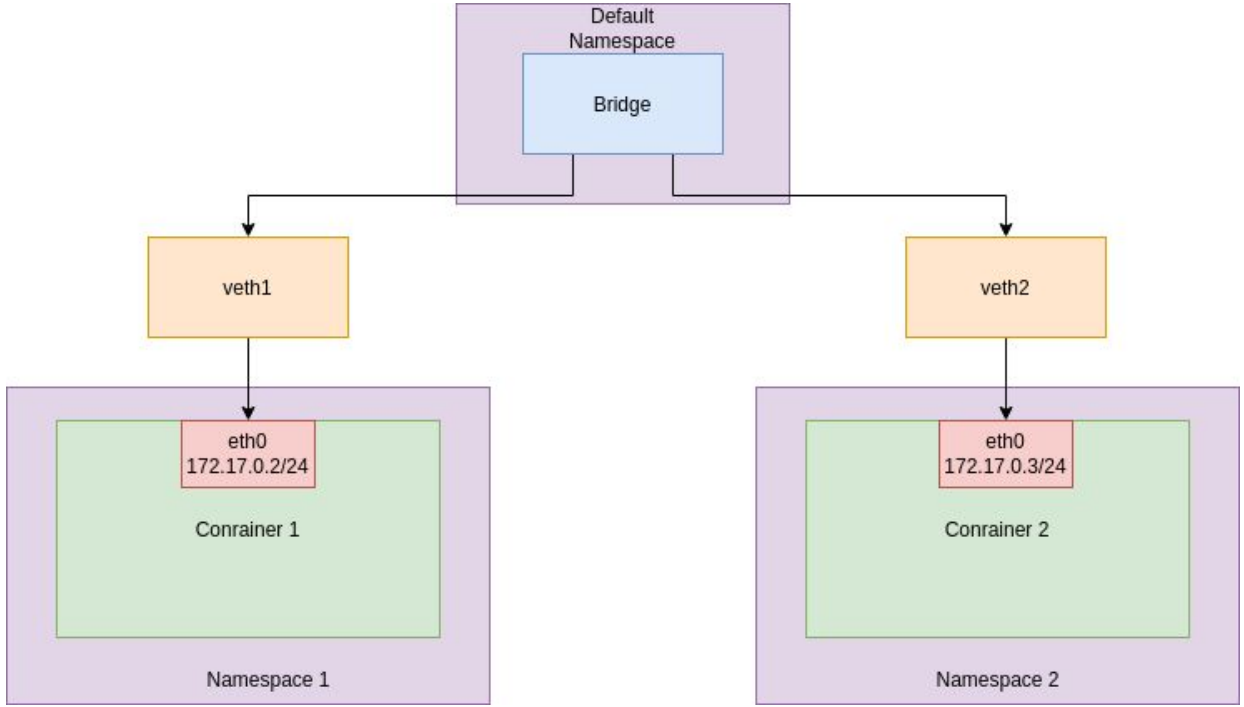
Docker overview

- Network namespace & bridge
- Filesystem isolation
- Resource management
- Onion Filesystem
- Management by cli interface

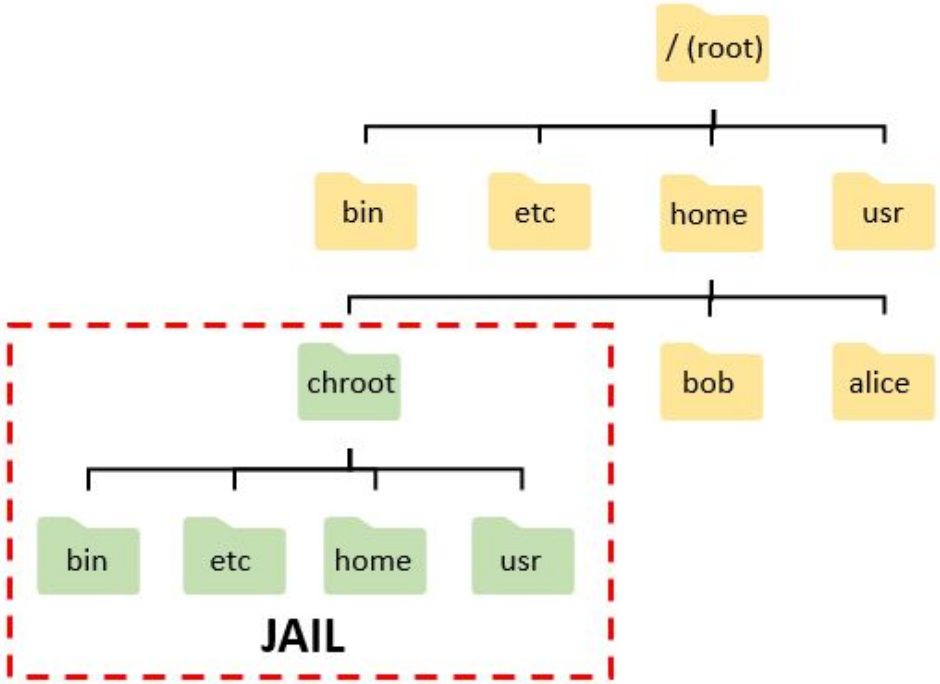
Linux bridges



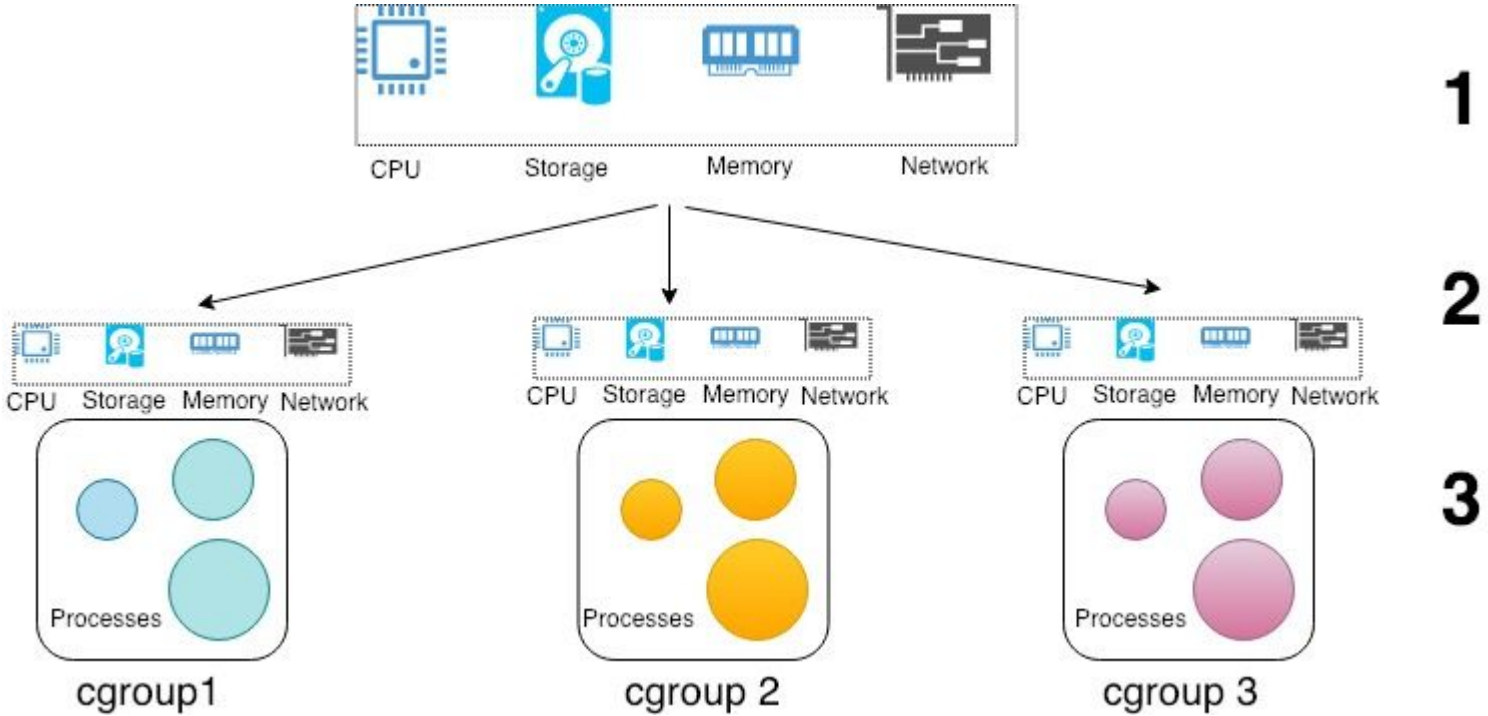
Network namespaces



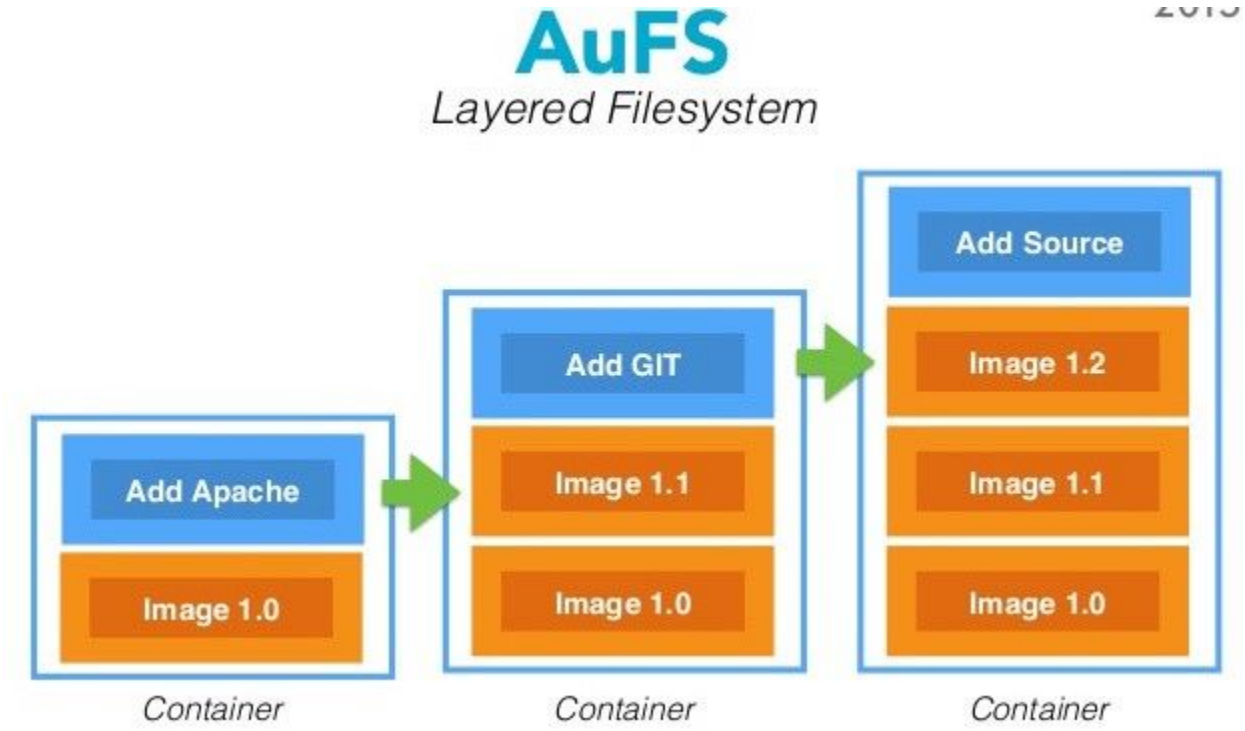
Filesystem isolation



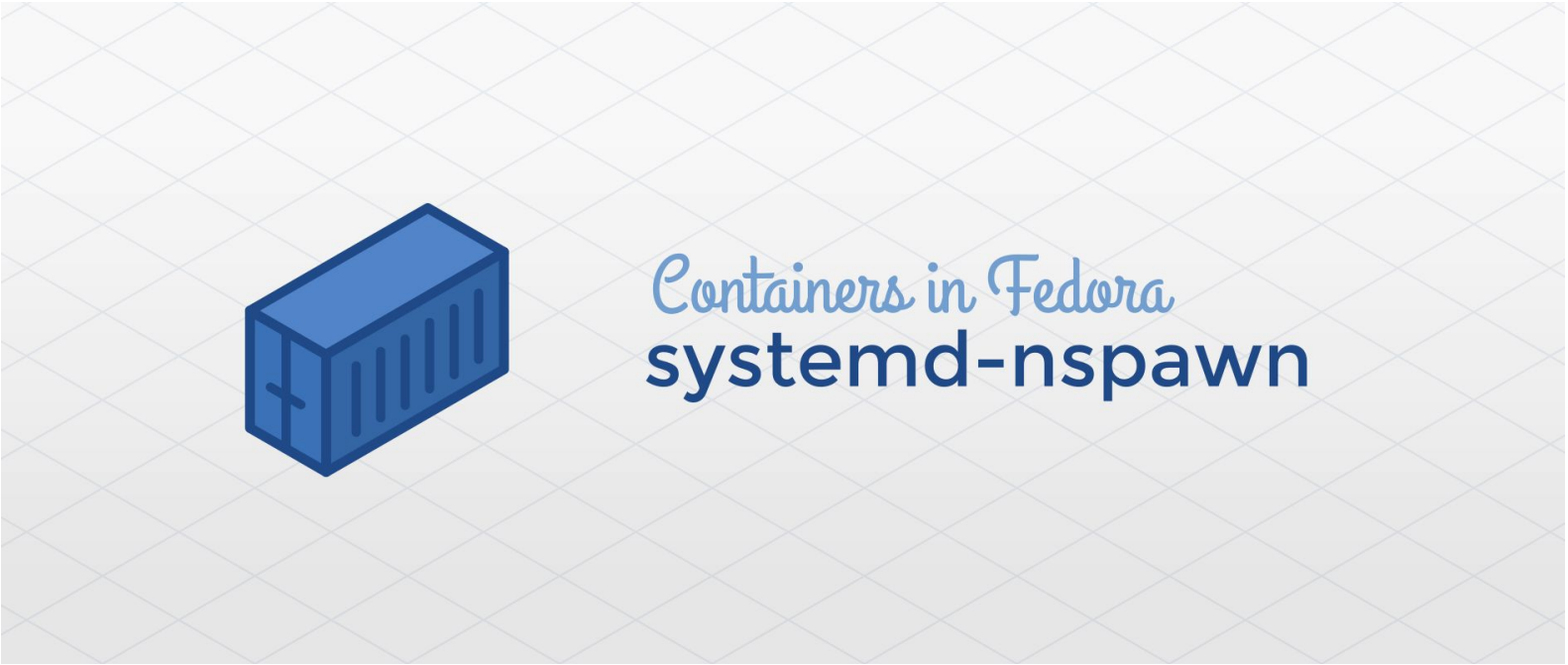
Resource management



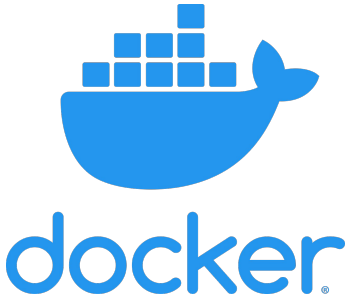
Onion filesystem

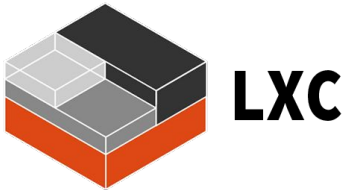


CLI Management



Container technologies





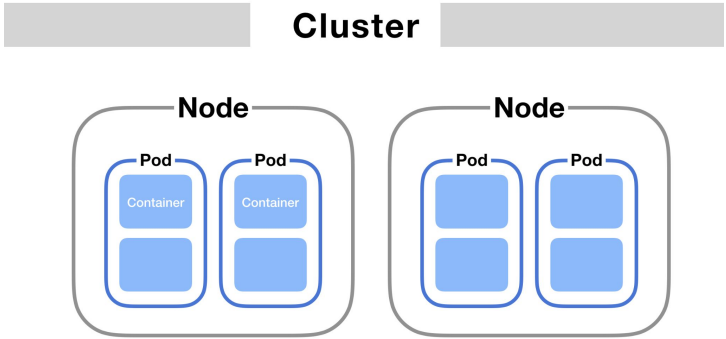
Deploy application

In kubernetes

Pod

Pods are the smallest deployable units of computing that you can create and manage in Kubernetes.

- Group of one or more containers, it contains one or more application containers which are relatively tightly coupled
- Shared storage and network resources, and a specification for how to run the containers



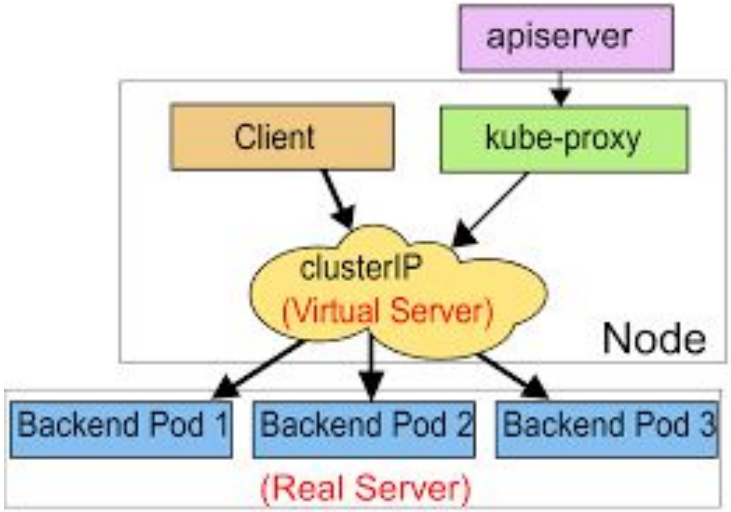
Pod manifest

```yaml
apiVersion: v1
kind: Pod
metadata:
  name: nginx
  labels:
    environment: production
    app: nginx
spec:
  containers:
  - name: nginx
    image: nginx:1.21.6
```

Service

An abstract way to expose an application running on a set of Pods as a network service

- Make a DNS name with a single ip address
- Load balance requests into the pods

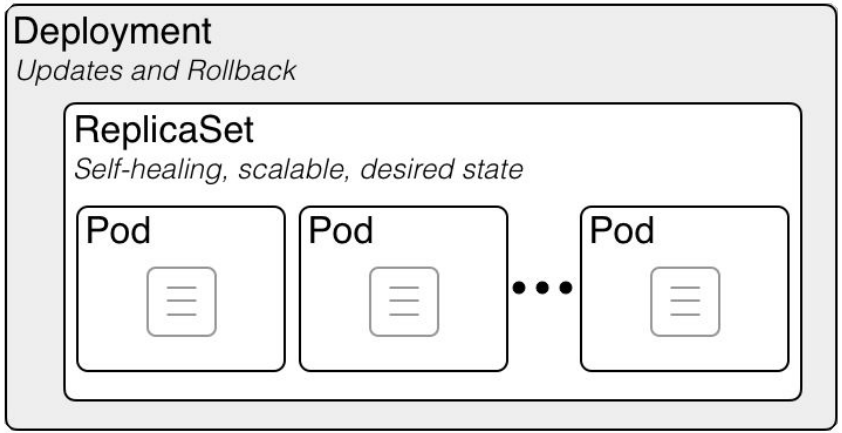


Service manifest

```
apiVersion: v1
kind: Service
metadata:
  labels:
        app: nginx
        environment: production
  name: nginx
  namespace: default
spec:
  ports:
  - nodePort: 30537
        port: 80
        protocol: TCP
        targetPort: 80
  selector:
        app: nginx
        environment: production
  type: NodePort
```

Deployment

A Deployment is one of the Kubernetes objects that is used to manage Pods in a declarative way. It provides updates, control as well as rollback functionalities. This means you can update or downgrade an application to the desired version without experiencing a user blackout as well as roll back to the previous version

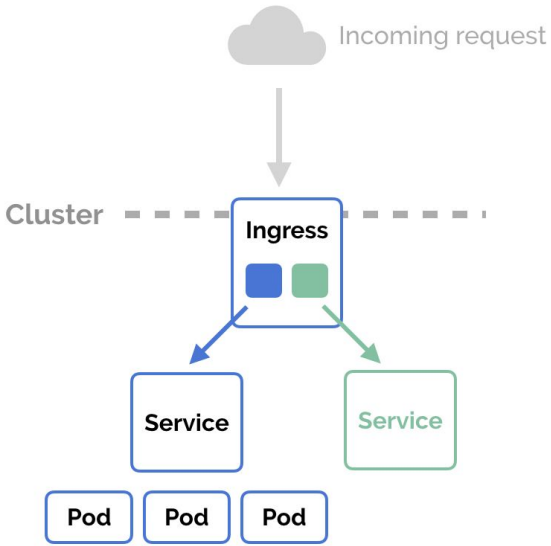


Deployment manifest

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx
  labels:
    app: nginx
spec:
  replicas: 1
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
        environment: production
    spec:
      containers:
        - name: app
          image: khafan_app:v1
```

Ingress

Ingress exposes HTTP and HTTPS routes from outside the cluster to services within the cluster. Traffic routing is controlled by rules defined on the Ingress resource.



Ingress manifest

```yaml
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: nginx
spec:
  rules:
    - host: nginx.local
      http:
        paths:
        - path: /
          pathType: Prefix
          backend:
            service:
              name: nginx
              port:
                number: 80
```

Q&A

Q&A

Thank You

www.digikala.com