

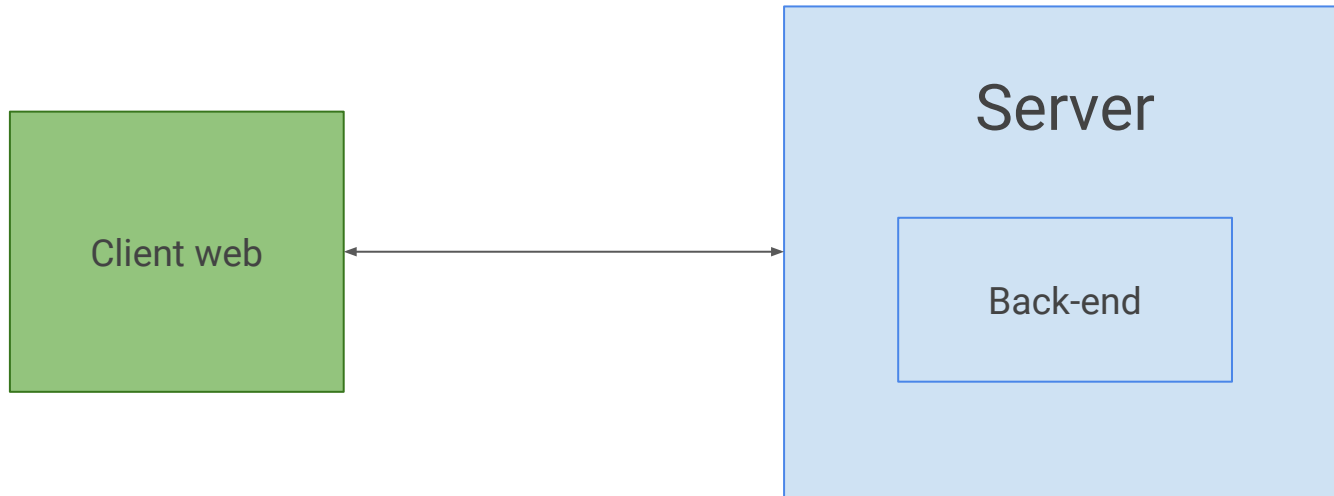


Introduzione a Kubernetes

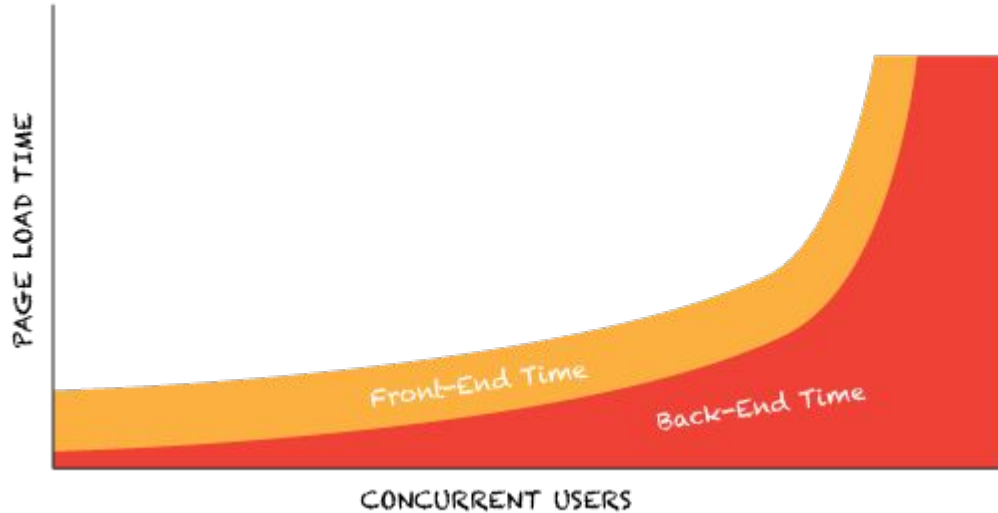
Cos'è, come funziona e a cosa serve

Fabio Da Soghe - Cognito

Web Performance: 80-20



Web Performance: 80-20?

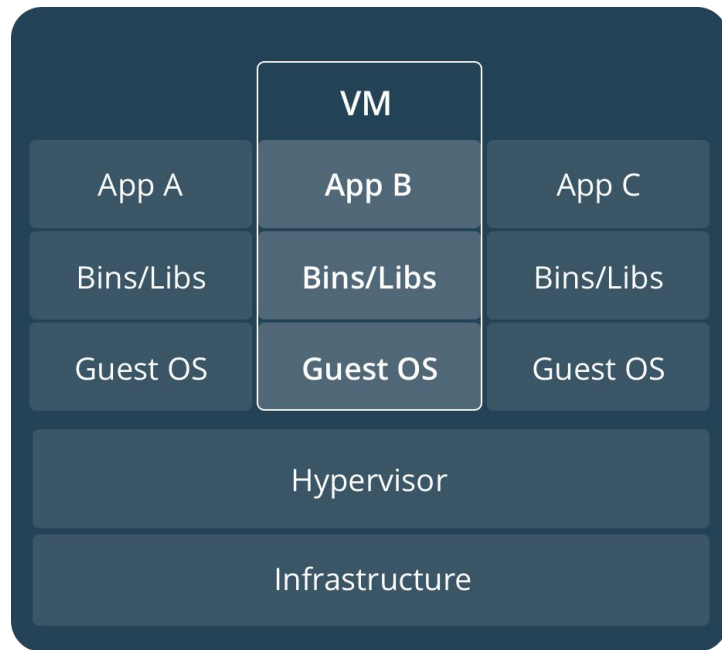


Fonte: <https://www.speedawarenessmonth.com/when-8020-becomes-2080/>

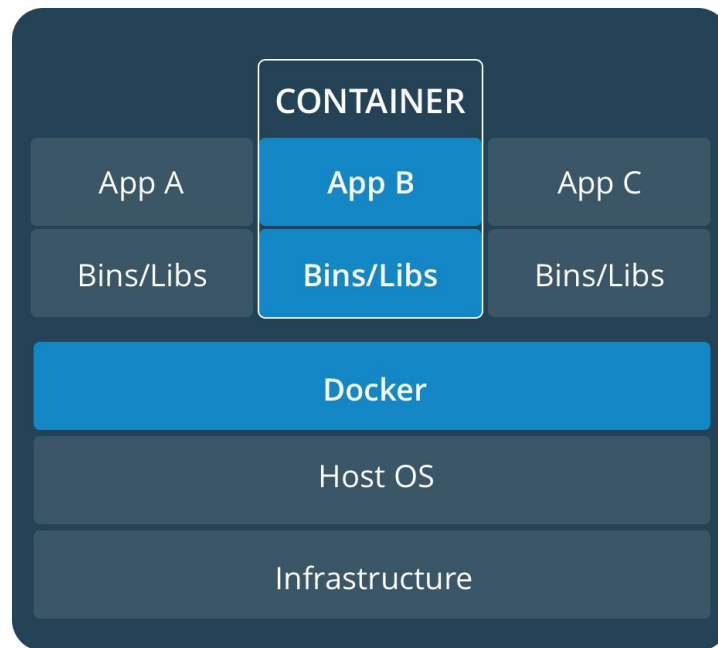
A chi è rivolto il talk

- Non conosce o non ha ancora usato Kubernetes
- Usa o almeno conosce i container (Docker)

Virtual Machine



Container



Cos'è Kubernetes

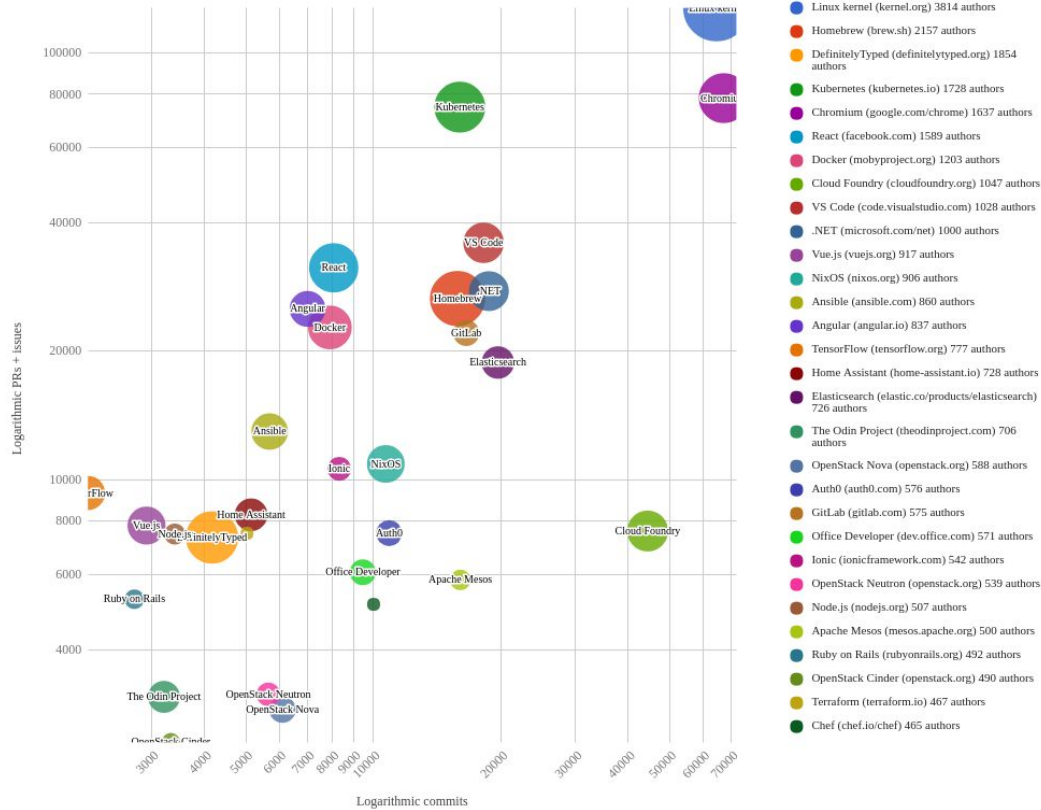
Orchestratore di container:

- Cluster di nodi (fisici o virtuali)
- Gestione dell'infrastruttura (computing e network)
- Automazione

Creato da Google

- Risultato di 15 anni di esperienza in Google
- Progetto Open Source donato da Google alla CNCF (Luglio 2015)
- Uno dei progetti Open Source più attivi
- Leader container orchestrator

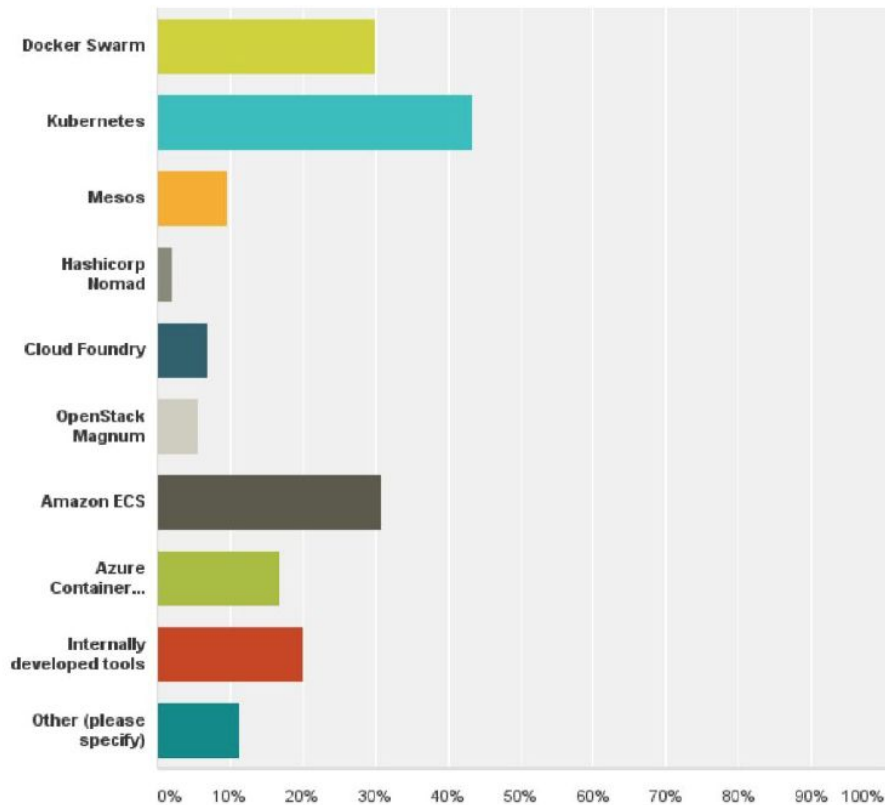
Top 30 Projects 05/2016 - 04/2017



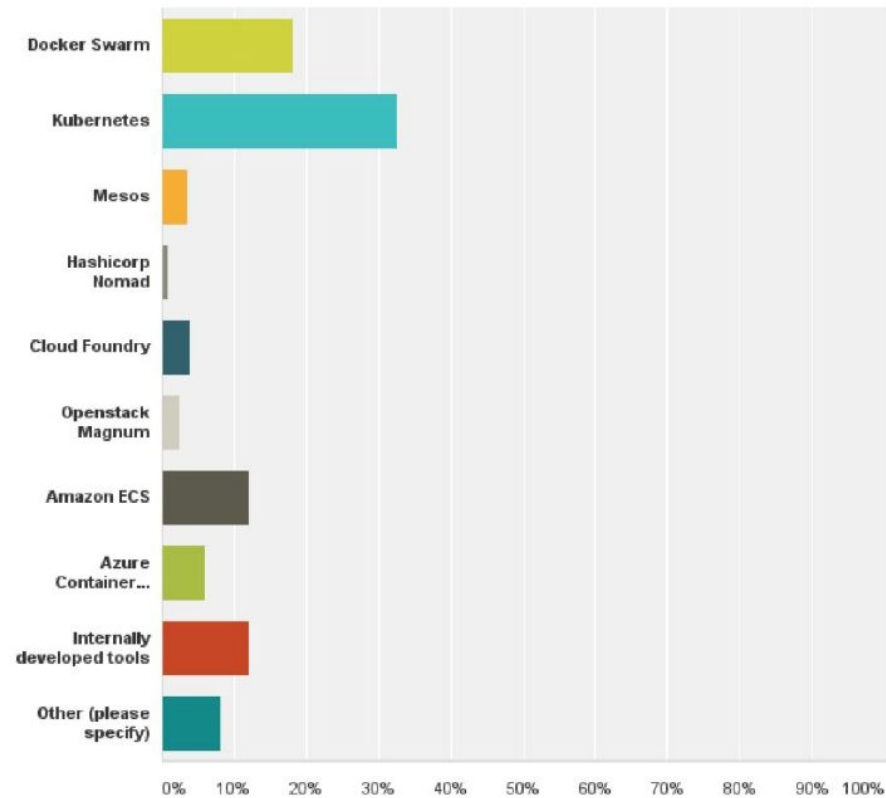
Fonte: CNCF

<https://www.cncf.io/blog/2017/06/05/30-highest-velocity-open-source-projects/>

Orchestratori usati in azienda



Orchestratore usato più spesso in azienda



Fonte: Portworx <https://portworx.com/2017-container-adoption-survey/>

Benefici di Kubernetes

Paradigma
dichiarativo

Distribuzione
automatica

Scaling
orizzontale e
verticale

Rollout e
rollback

Gestione storage

Self-Healing

Service
discovery

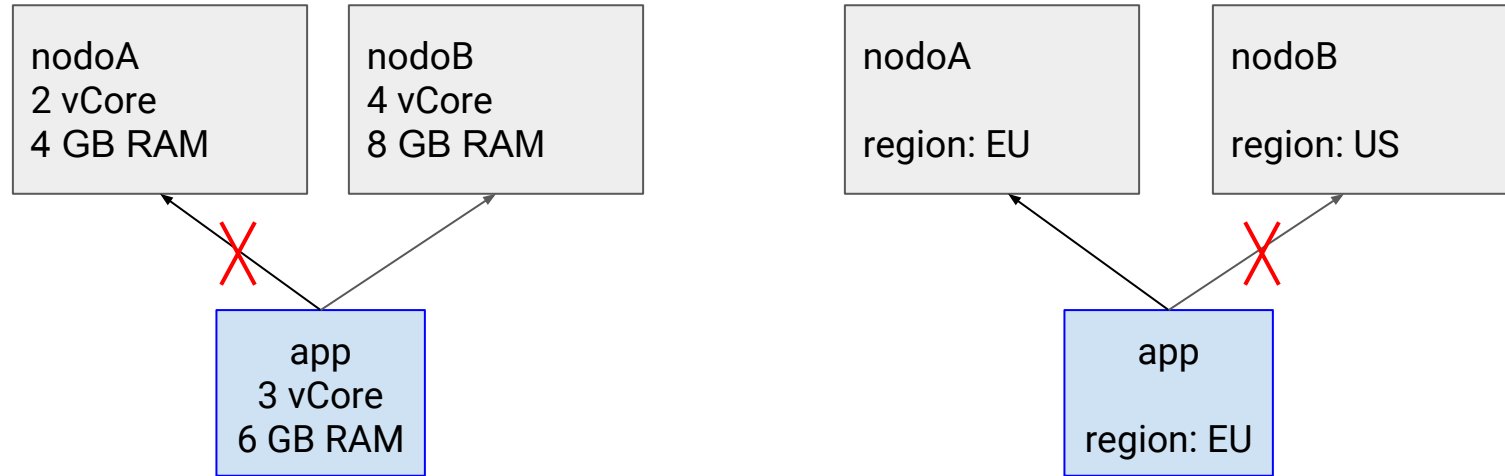
Gestione secret e
configurazione

Paradigma dichiarativo

- Descrizione dello stato desiderato dell'applicazione (YAML)
- Non come, ma cosa
- Definizione dell'intera infrastruttura necessaria all'applicazione in produzione

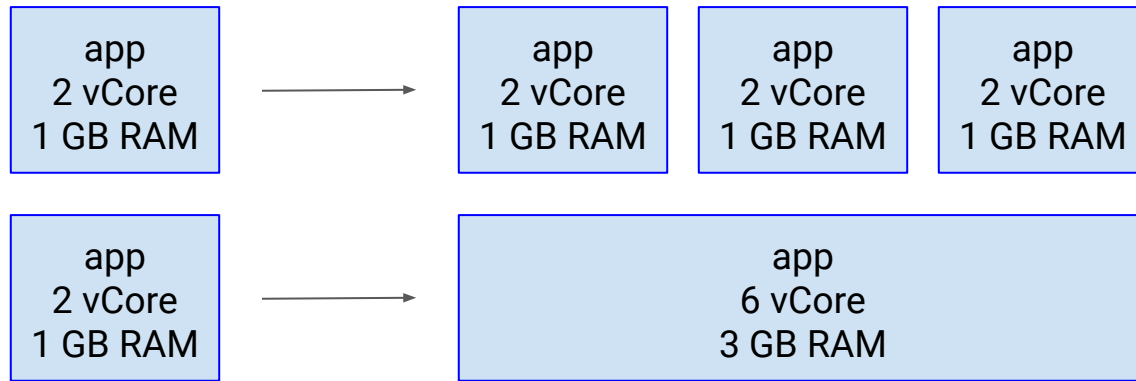
Distribuzione automatica

- Collocamento automatico dei runtime sui nodi
- Rispetto di requirement e constraint
- Applicazione di politiche di massima disponibilità e bilanciamento del carico



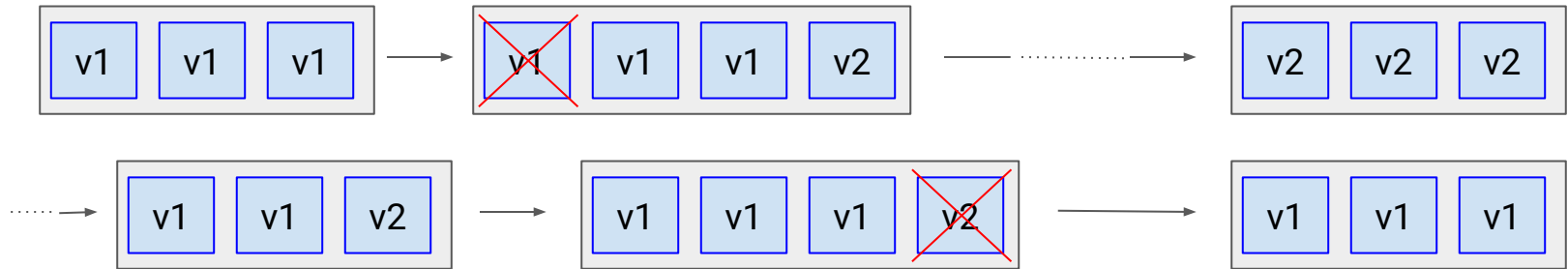
Scaling orizzontale e verticale

- Aumento o diminuzione container in esecuzione (scaling orizzontale)
- Aumento o diminuzione risorse cpu/ram (scaling verticale)
- Manuale e automatico, in base a soglie prestabilite



Rollout e rollback

- Applicazione progressiva di modifiche e aggiornamenti (supporto a canary release)
- Monitoraggio costante dello stato
- Processo automatico
- Rollback in caso di problemi

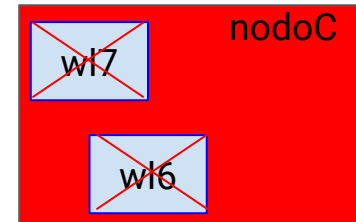
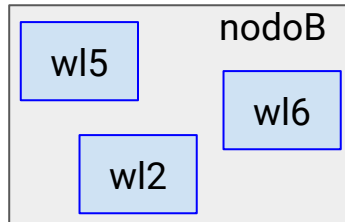
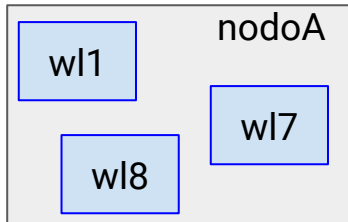
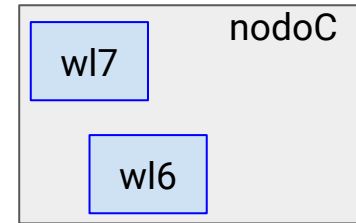
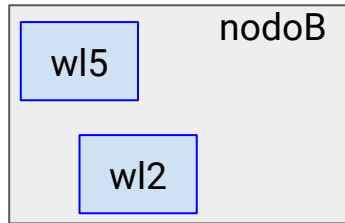
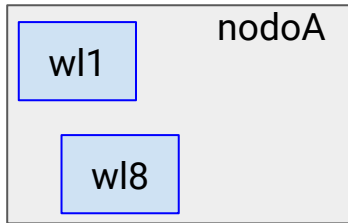


Gestione storage

- Supporto a diversi storage provider: locale, network (NFS, iSCSI, Gluster, Ceph, Cinder, Flocker, ...), cloud (GCP, AWS, Azure)
- Estendibile da terze parti
- Manuale o auto-provisioned

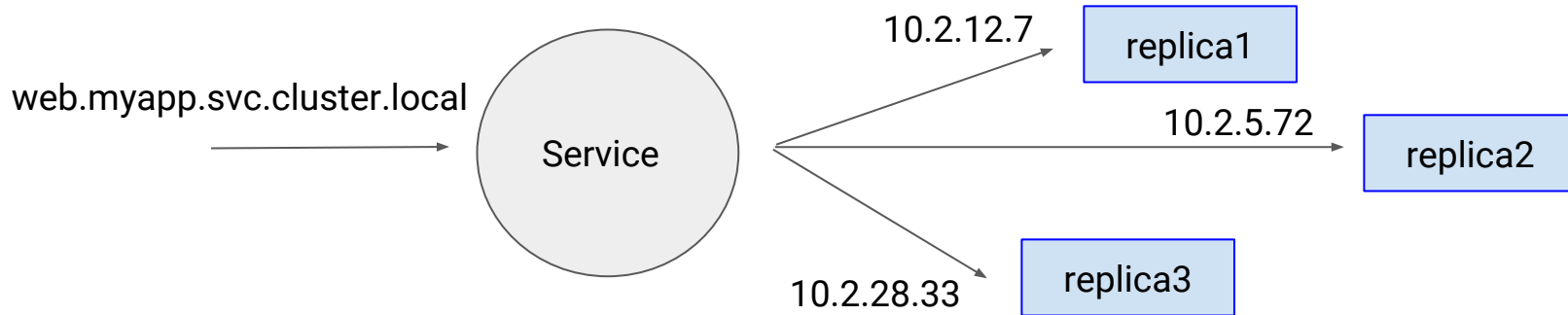
Self-healing

- Riavvio container in errore (possibile ricollocamento)
- Ricollocamento automatico in caso crash nodi
- Health check definiti dall'utente



Service discovery

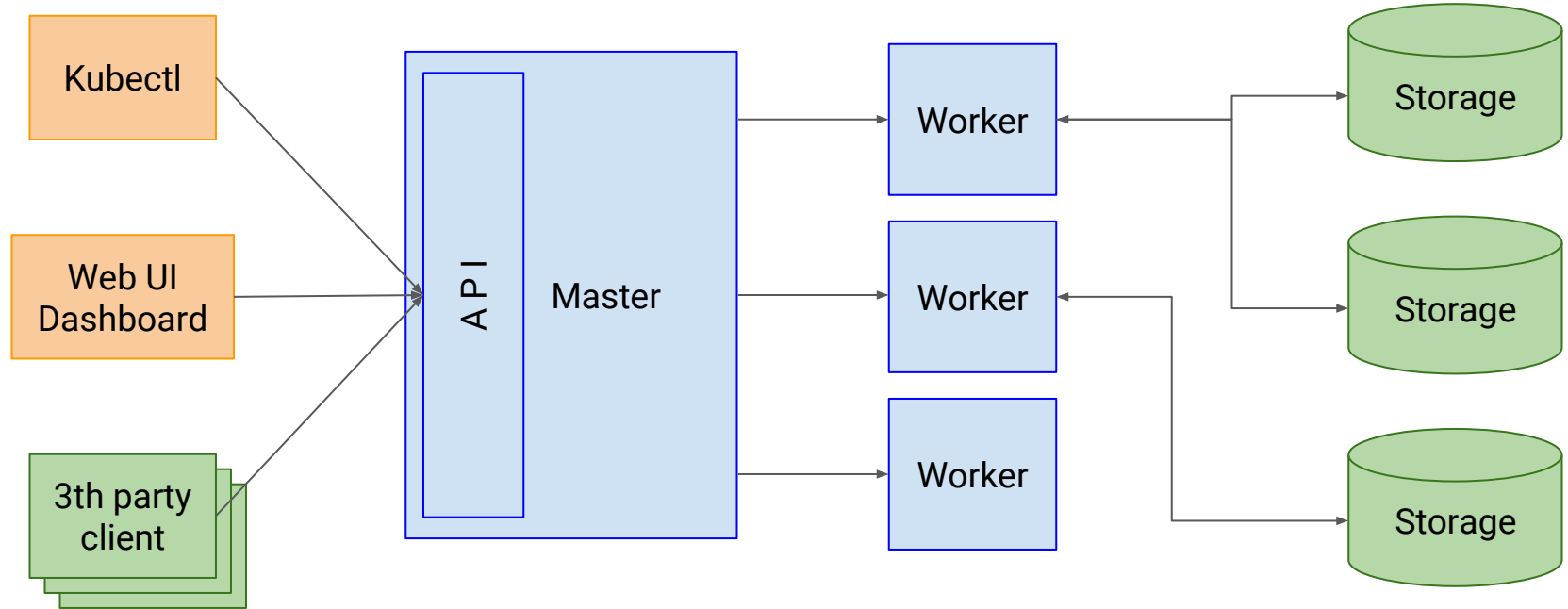
- Meccanismi built-in per pubblicazione servizi ed endpoint
- DNS interno per assegnare ai container gli IP
- Networking virtuale sull'intero cluster
- Load Balancing dei servizi esposti



Gestione secret e configurazione

- Gestione esplicita informazioni sensibili (secret)
- Disaccoppiamento esplicito configurazione dai container

Architettura generale



Oggetti Kubernetes

Workload:

- Deployment
- Replica Set
- Stateful Set
- Daemon Set
- Job / Cron Job
- **Pod**

Networking:

- Service
- Ingress

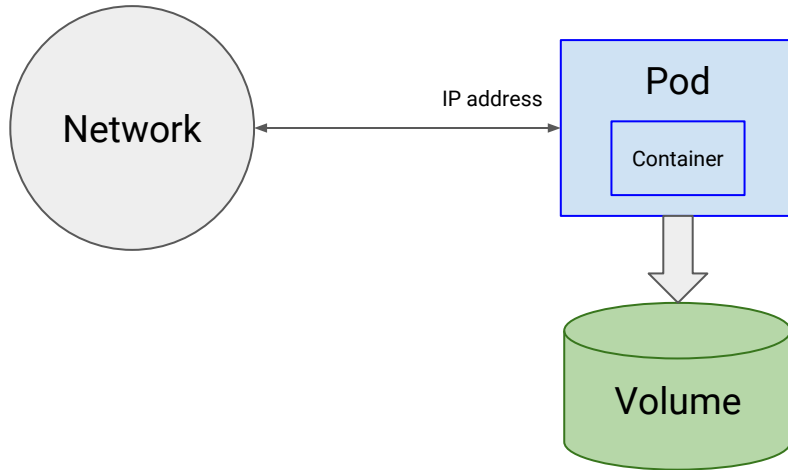
Storage:

- Volume
- Persistent Volume
- Persistent Volume Claim

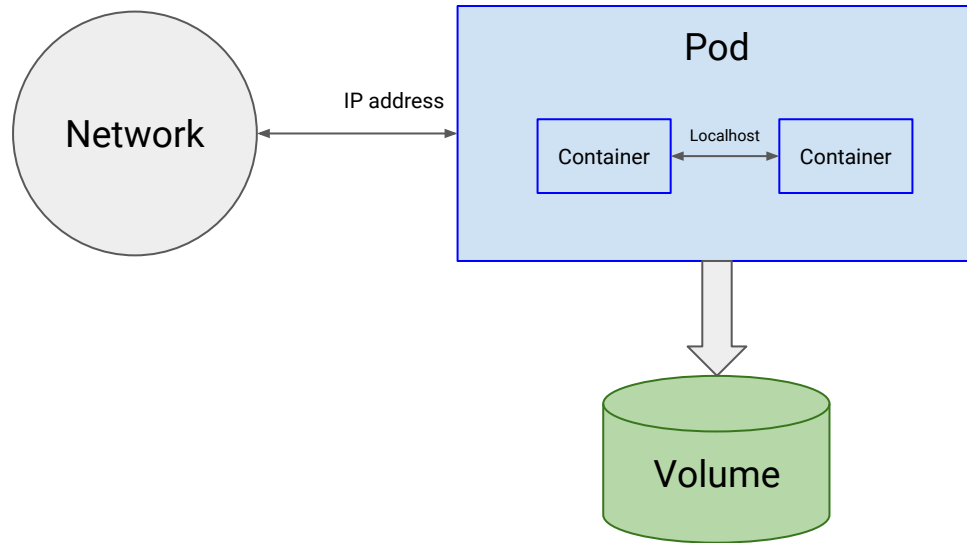
Configuration:

- Namespace
- Secret
- Config Map

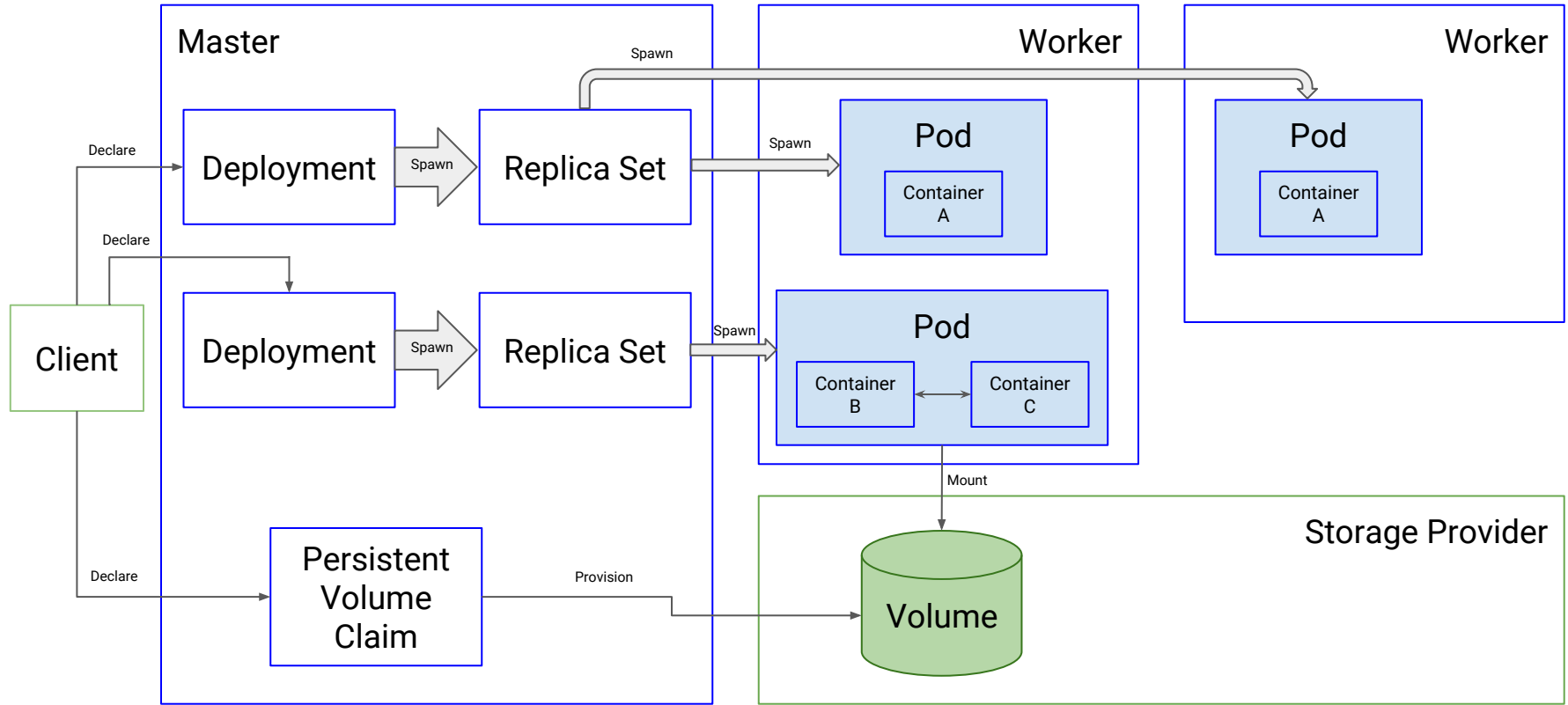
Pod: l'unità di elaborazione



Pod: l'unità di elaborazione



Struttura (core) di un'applicazione

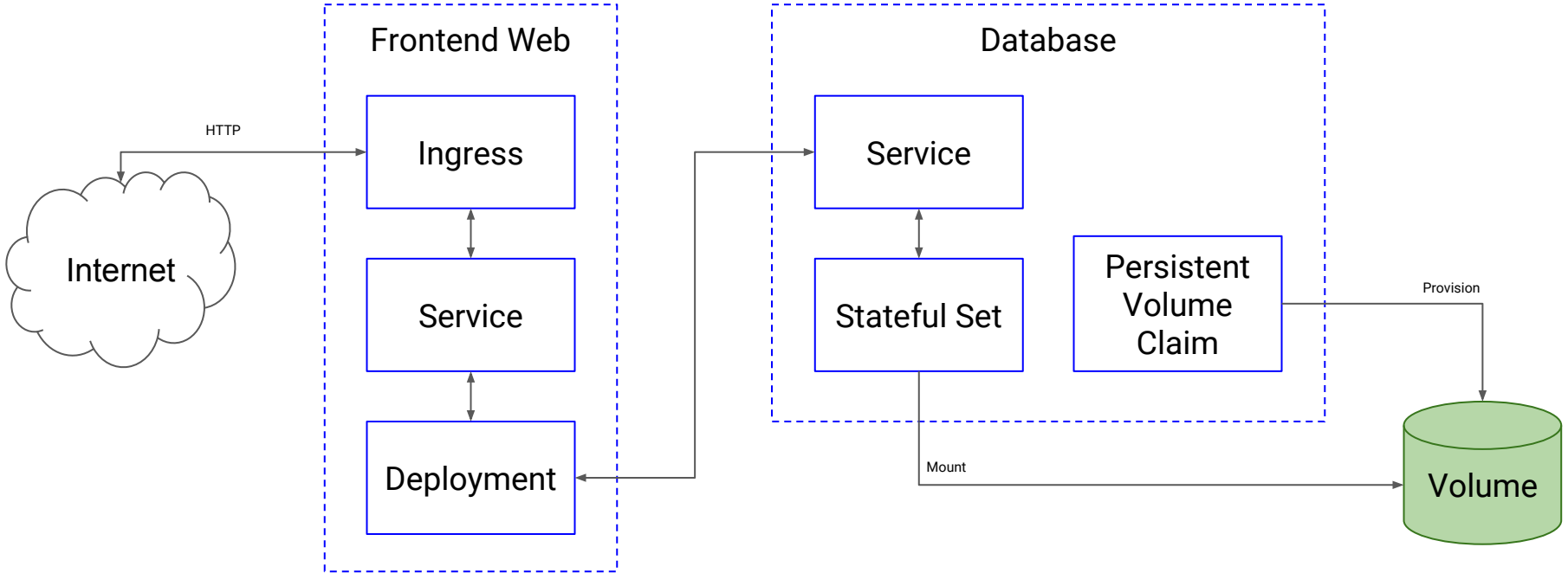


It's demo time!

Applicazione web:

- Frontend web Meteor (-> Node.js)
- Database MongoDB

Architettura logica K8s Demo



YAML frontend web (1/2)

kind: Ingress

```
apiVersion: extensions/v1beta1
metadata:
  name: k8s-demo
spec:
  backend:
    serviceName: k8s-demo
    servicePort: 80
```

kind: Service

```
apiVersion: v1
metadata:
  name: k8s-demo
spec:
  type: NodePort
  ports:
    - port: 80
      targetPort: http
  selector:
    app: k8s-demo
```

YAML frontend web (2/2)

```
kind: Deployment
apiVersion: apps/v1beta1
metadata:
  name: nodejs
spec:
  replicas: 1
  strategy:
    type: RollingUpdate
    rollingUpdate:
      maxSurge: 2
      maxUnavailable: 1
  template:
    metadata:
      labels:
        app: k8s-demo
    spec:
```

```
containers:
- image: "eu.gcr.io/myrepo/k8s-demo"
  name: nodejs
  resources:
    limits:
      cpu: "0.1"
    requests:
      cpu: "0.01"
  env:
- name: MONGO_URL
  value: "mongodb://mongo-0.mongo-27017/k8s-demo?replicaSet=rs0"
- [...]
  ports:
- containerPort: 3000
  name: http
  readinessProbe:
    httpGet:
      path: /rest/data/add?size=10&numDocs=0&work=0
      port: 3000
    initialDelaySeconds: 10
    periodSeconds: 1
    failureThreshold: 1
```

YAML database (1/3)

```
kind: Service
apiVersion: v1
metadata:
  name: mongo
  labels:
    name: mongo
spec:
  ports:
    - port: 27017
      targetPort: 27017
  clusterIP: None
  selector:
    role: mongo
```

YAML database (2/3)

```
kind: StatefulSet
apiVersion: apps/v1beta1
metadata:
  name: mongo
spec:
  serviceName: "mongo"
  replicas: 1
  updateStrategy:
    type: RollingUpdate
  template:
    metadata:
      labels:
        role: mongo
    spec:
      terminationGracePeriodSeconds: 10
      containers:
```

```
- name: mongo
  image: mongo
  resources:
    limits:
      cpu: "0.1"
    requests:
      cpu: "0.01"
  command:
    - "mongod"
    - "--replSet"
    - "rs0"
    - "--smallfiles"
    - "--noprealloc"
  ports:
    - containerPort: 27017
  volumeMounts:
    - name: mongo-persistent-storage
      mountPath: /data/db
```

YAML database (3/3)

```
- name: mongo-sidecar
  image: cvallance/mongo-k8s-sidecar
  env:
    - name: MONGO_SIDECAR_POD_LABELS
      value: "role=mongo"
    - name: KUBERNETES_MONGO_SERVICE_NAME
      value: "mongo"

  volumeClaimTemplates:
    - metadata:
        name: mongo-persistent-storage
        annotations:
          volume.beta.kubernetes.io/storage-class:
"standard"
      spec:
        accessModes: [ "ReadWriteOnce" ]
        resources:
          requests:
            storage: 10Gi
```

Kubernetes in Cognitio



Plenus:
Platform-as-a-Service
basato su Kubernetes

www.plenus.cloud

