Introduction to Kubernetes

SciLifeLab

CONTAINERS

- “Environment” in a running operating system
- Not a virtual machine
- Kernel namespaces
  - Processes
  - Network interfaces
  - Filesystem mount points
  - ...
- Docker

DOCKERFILE

```
FROM node:latest as build

RUN yarn global add @quasar/cli
COPY ./frontend/package.json /package.json
WORKDIR /
RUN yarn install
COPY ./frontend /code
RUN mv /node_modules /code/
WORKDIR /code
RUN quasar build


FROM nginx:alpine

COPY --from=build /code/dist/spa/ /usr/share/nginx/html/
COPY ./k8s/nginx.conf /etc/nginx/nginx.conf

EXPOSE 80
```

Kubernetes

- Container orchestration
- Originates from Google
- Kubernetes $\rightarrow$ K + 8 letters + s $\rightarrow$ k8s (“Kates”)

Basics

- `kubectl`
  - `kubectl help -f ...`
  - `kubectl apply -f ...`
  - `kubectl delete -f ...`
  - `kubectl get ...`
  - `kubectl describe ...`
- “Death by `YAML`”

EXAMPLE

```yaml
apiVersion: v1
kind: Pod
metadata:
  name: menu-frontend
  labels:
    id: menu-frontend

spec:
  containers:
    - name: menu-frontend
      image: scilifelabdatacentre/menu-frontend:latest
      ports:
        - containerPort: 80
```

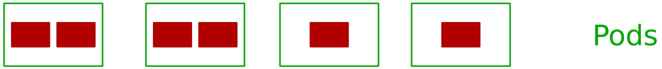
Example

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: dsw-client
  labels:
    name: dsw-client
    type: prod
  namespace: dc-dsw
spec:
  selector:
    matchLabels:
      name: dsw-client
      type: prod
  replicas: 1
  template:
    metadata:
      labels:
        name: dsw-client
        type: prod
        version: 2.5.0
    spec:
      containers:
      - name: dsw-client
        image: dsw/wizard-client:2.5.0
        resources:
          limits:
            cpu: 600m
            memory: 200Mi
          requests:
            cpu: 200m
            memory: 40Mi
        ports:
          - containerPort: 80
        readinessProbe:
          httpGet:
            path: /
            scheme: HTTP
            port: 80
        volumeMounts:
          - name: client-conf
            mountPath: /src/scss/_variables.scss
            subPath: _variables.scss
          - name: client-conf
            mountPath: /src/scss/_overrides.scss
            subPath: _overrides.scss
          - name: custom-assets
            mountPath: /usr/share/nginx/html/assets
        env:
          - name: API_URL
            valueFrom:
              configMapKeyRef:
                name: prod-client
                key: API_URL
      volumes:
        - name: client-conf
          configMap:
            name: prod-client
        - name: custom-assets
          configMap:
            name: custom-assets
```

NAMESPACE

- Internal grouping
- DNS
  - `<service>.<namespace>.svc.cluster.local`
- Access/resource control

Pods



Pods

- “Basic unit”
- One or more containers in each pod
- A deleted pod won’t restart

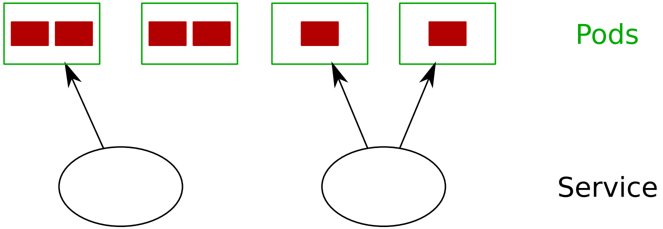
VOLUMES

- Persistent storage
- `PersistentVolume`
  - Storage, e.g. a folder on a hd or nfs
  - Global
- `StorageClass`
  - Storage pool
  - Global
- `PersistentVolumeClaim`
  - The “claim” for storage used by the pod.
  - Namespaced

Configuration Variables and Files

- `env:`
- ConfigMap
- Secret
  - Not encrypted
  - Base64
- Use cases:
  - Environment variables
  - Arguments
  - Files

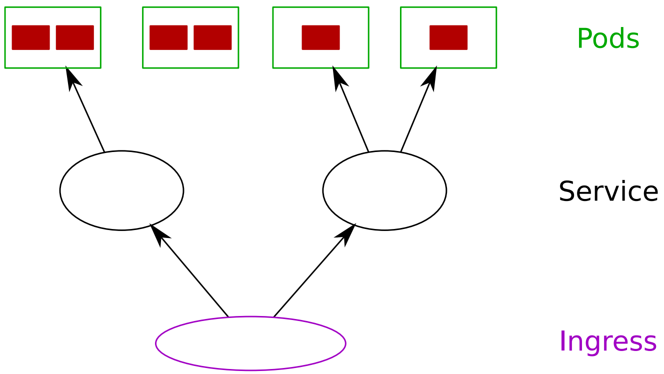
SERVICES



SERVICES

- Allow easy access (“DNS”) to groups of pods
- Selecting pods using labels
- Can be used to expose pods to external access
- `ClusterIP`, `NodePort`, `LoadBalancer`

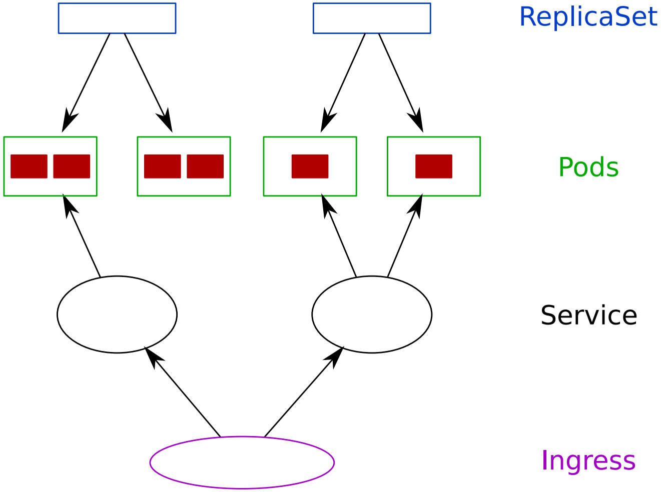
INGRESS



Ingress

- Http(s) access to containers
- Certmanager
  - Certification management
  - Integration with Let's Encrypt
- Interface
  - No official implementation
- Multiple available
  - Nginx
  - Ingress-GCE
  - AWS ALB Ingress Controller
  - Traefik
  - ...

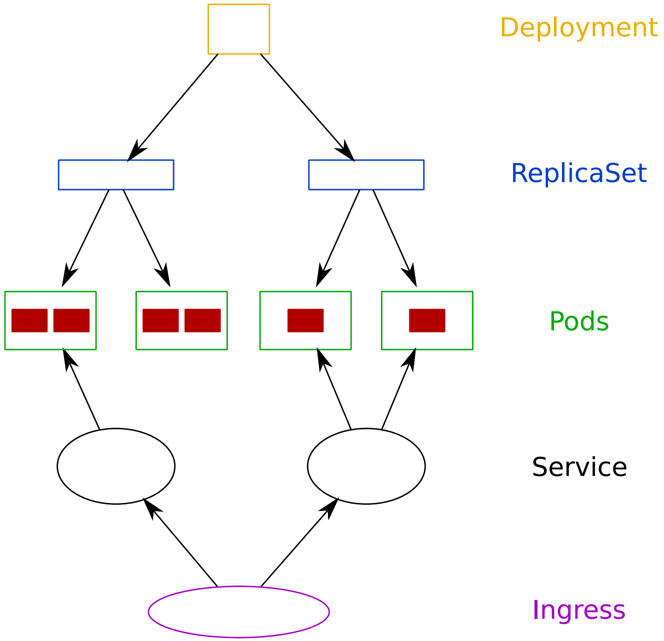
Replica Sets



Replica Sets

- Defines replication of pods
- Will start/stop pods to match the wanted number of `replicates`

Deployments



DEPLOYMENTS

- Deployments of replica sets
- Allows progressive deployment of new containers

OTHER USEFUL THINGS

- Certmanager
- `readinessProbe`
- `StatefulSet`
- `kustomization.yml`

Linus Östberg
SciLifeLab Data Centre

talavis
@lostb

https://github.com/talavis/coffee-k8s

SciLifeLab