

# Kubernetes

Servisno orijentisane arhitekture

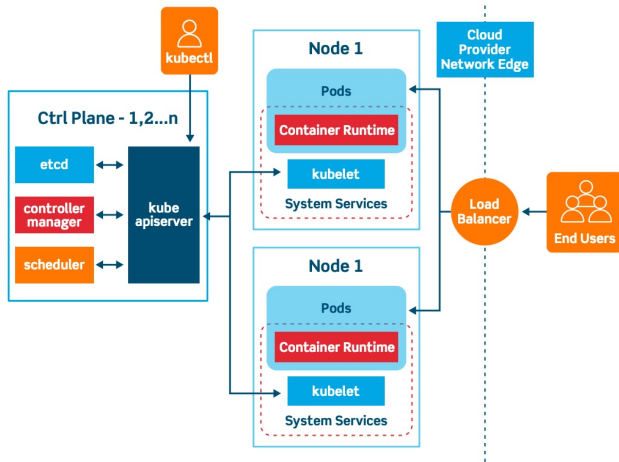


**Univerzitet u Novom Sadu**  
**Fakultet Tehničkih Nauka**

# Kubernetes (k8s)

- ▶ Inicijalno razvijen od strane Google-a
- ▶ Open source implementacija in house orkestratora Borg
- ▶ Koristan alat za rad za kontejnerima
- ▶ Omogućio upotrebu mikroservisnih aplikacija i automatizaciju posla
- ▶ Vrlo bitan faktor za razvoj cloud native aplikacija!
- ▶ Stanje se opisuje deskriptivno koristeći YAML – sistem se snalazi sa ostalim stvarima
- ▶ Vodiće računa da se trenutno i željeno stanje poklope – reconcile loop

# Arhitektura



(Kubernetes Architecture and Concepts)

# Resursi

- ▶ Osnovni resurs kojim Kubernetes upravlja naziva se **pod** — jedan ili više kontejnera koji mogu direktno da komuniciraju
- ▶ Sistem labela nam olakšava referenciranje različitih resursa
- ▶ Ostali tipovi resursa (koje ćemo mi raditi):
  - ▶ **Service** - Komunikacija sa pod-ovima treba da se obavlja preko servisa
  - ▶ **Deployment** - Pod repliciran n puta i njihova deployment strategija
  - ▶ **ConfigMap** - Konfiguracija koja se može proslediti pod-ovima (preko fajlova ili env varijabli)
  - ▶ **PersistentVolume** - Deo raspoloživog diska koje pod-ovi mogu koristiti
  - ▶ **PersistentVolumeClaim** - Zahtev za povezivanje sa nekim dostupnim persistent volume-om
- ▶ Pored ovoga postoji još veliki broj tipova resursa, a moguće je definisati i nove tipove – CustomResourceDefinition (CRD)

# Instalacija i pokretanje

- ▶ Potrebno je instalirati sledeće alate:
  - ▶ minikube
  - ▶ kubectl
- ▶ minikube će za nas kreirati k8s klaster sa jednim virtuelnim čvorom
- ▶ Osnovne minikube komande:
  - ▶ `minikube start` - Pokreće klaster
  - ▶ `minikube ip` - IP adresa virtuelnog čvora (na tu adresu šalјemo sve zahteve)
  - ▶ `minikube stop` - Zaustavlja klaster



# Učitavanje kontejner slika u minikube

- ▶ `docker build . -t <image-name:tag>`
- ▶ `minikube image load <image-name:tag>`
- ▶ `minikube image ls`
- ▶ `minikube image rm <image-name:tag>`

# Pod

- ▶ `kubectl apply -f pod.yml`
- ▶ `kubectl get pods`
- ▶ `kubectl delete pod my-first-pod`

**Pitanje:** Da li možemo da kontaktiramo pod direktno?

# Deployment

- ▶ `kubectl apply -f deployment.yml`
- ▶ `kubectl get deployments`
- ▶ `kubectl get pods`
- ▶ `kubectl delete <pod-name>`
- ▶ `kubectl get pods`

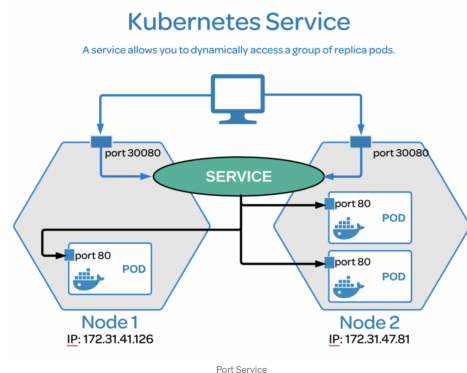
**Pitanje:** Zašto se pod nije obrisao?



# Service

- ▶ `kubectl apply -f service.yml`
- ▶ `kubectl get services`

**Pitanje:** Koju adresu i port ćemo koristiti da pristupimo servisu?



# ConfigMap

- ▶ `kubectl apply -f configmap.yml`
- ▶ `kubectl apply -f deployment-env.yml`
- ▶ `kubectl apply -f deployment-file.yml`

**Pitanje:** Da li će naša aplikacija videti novu *db.name* vrednost bez restarta?

# PersistentVolume/PersistentVolumeClaim

- ▶ `kubectl delete all -all`
- ▶ `kubectl apply -f deployment-mongo.yml`
- ▶ `kubectl delete all -all`
- ▶ `kubectl delete pvc -all`
- ▶ `kubectl delete pv -all`
- ▶ `kubectl delete configmap -all`

