



UNIVERZITET U NOVOM SADU
FAKULTET TEHNIČKIH NAUKA
KATEDRA ZA PRIMENJENE RAČUNARSKE NAUKE

Računarstvo u oblaku

ms Helena Anišić

Zimski semester 2025/2026.

Studijski program: Računarstvo i automatika

Modul: Računarstvo visokih performansi

Umrežavanje

Servis

- Servis je objekat koji omogućava mrežni pristup skupu Pod-ova unutar klastera.
- Servis definiše kako aplikacije ili korisnici mogu pristupiti Podovima, čime se omogućava konzistentna i pouzdana komunikaciju.
- Svaki Pod ima internu IP adresu (kao što i Docker kontejneri imaju svoje interne IP adrese)
 - PROBLEMI:
 - Ne može se koristiti interna IP adresa da se pristupi od spolja
 - IP adresa se menja svaki put kad se Pod ponovo kreira
- Servis grupiše Pod-ove i dodeljuje im zajedničku IP adresu koja se ne menja i na taj način se može Pod-ovima pristupiti i od spolja.
- **Unutar klastera** je **jako teško** pristupiti Podovima ako se ne koristi Servis.
- **Izvan klastera** je **nemoguće** pristupiti Podu ako se ne koristi Servis.

Servis

- Svaki servis u Kubernetes klasteru ima svoju IP adresu i DNS ime, što pruža stalnu tačku pristupa za komunikaciju sa podovima, bez obzira na to gde se podovi nalaze u klasteru.
- Kubernetes podržava različite tipove servisa:
 - ClusterIP (unutrašnji pristup unutar klastera),
 - NodePort (spoljašnji pristup preko IP adrese čvora),
 - LoadBalancer (integracija sa spoljašnjim balanserom opterećenja) i
 - ExternalName (mapiranje servisa na eksterni DNS).

Pod-internal komunikacija

- Dva kontejnera koja se nalaze u istom Pod-u mogu da komuniciraju kao da su im portovi namapirani na lokalnu mašinu:
 - adresa je: `localhost:<PORT>`

Pod-to-pod komunikacija

- Hardkodiranje interne IP adrese servisa (CLUSTER-IP)
- Automatski generisane environment varijable
- Automatski generisane domenske nazive

Pod-to-pod komunikacija

- Hardkodiranje interne IP adrese servisa (CLUSTER-IP)
 - IP adresa servisa unutar čvora
 - Ne može da se koristi za spoljnu komunikaciju
 - npr. za pristup tasks servisima adresa je <http://10.104.195.239:8000/>

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
auth-service	ClusterIP	10.111.207.137	<none>	80/TCP	60s
frontend-service	LoadBalancer	10.101.12.142	<pending>	3000:31587/TCP	22s
kubernetes	ClusterIP	10.96.0.1	<none>	443/TCP	29d
tasks-service	LoadBalancer	10.104.195.239	<pending>	8000:31742/TCP	45s
users-service	LoadBalancer	10.107.137.136	<pending>	8080:32110/TCP	33s

Pod-to-pod komunikacija

- Automatski generisane environment varijable
 - Podovi u okviru klastera imaju pristup određenim environment varijablama
 - Environment varijabla: `<NAZIV_SERVISA_SERVICE_HOST>` automatski se pretvara u CLUSTER_IP
 - zaobilazi se hardkodiranje
 - npr. za pristup tasks servisu adresa je [http://\\${process.env.TASKS_SERVICE_SERVICE_HOST}:8000/](http://${process.env.TASKS_SERVICE_SERVICE_HOST}:8000/)

CoreDNS: DNS i Service Discovery

- CoreDNS je fleksibilan i proširiv DNS server koji se često koristi u cloud-native okruženjima, kao što je Kubernetes.
- Kreira naziv domena koji je poznat unutar klastera
 - adresa je: `<naziv_servisa>.<naziv_namespace-a>`
 - npr. umesto <http://10.104.195.239:8000/> adresa tasks servisa je je <http://tasks-service.default:8000/>
- Predefinisani namespace u klasteru je: **default**

Reverse proxy

- Reverse proxy je vrsta servera koji prima zahteve sa interneta i prosleđuje ih drugim serverima u pozadini.
- Poznati alati za kreiranje reverse proxy-ja su:
 - nginx, apache, caddy,..
- Reverse proxy štiti identitet pozadinskih servera tako što preusmerava zahteve, čime ovi serveri ostaju nevidljivi za krajnje korisnike.
- Može raspodeliti dolazne zahteve na više servera, čime se poboljšavaju performanse i pouzdanost sistema.
- Reverse proxy se često koristi u velikim web infrastrukturama kako bi se poboljšala sigurnost, performanse i pouzdanost.

Forward proxy vs Reverse proxy

- **Forward proxy:**
 - **Klijent je svestan:** Korisnici su svesni i namerno koriste forward proxy (npr. postavljanje proxyja u pretraživaču).
 - **Služi klijentima:** Forward proxy deluje kao posrednik između korisnika (klijenta) i interneta. Korisnički zahtevi prvo idu do forward proxyja, koji zatim šalje zahteve internetu.
 - **Skrivanje korisnika:** Glavna svrha je sakriti identitet korisnika od internetskih servera, pružajući anonimnost.
 - **Kontrola i filtriranje:** Često se koristi u organizacijama za kontrolu pristupa internetu, filtriranje sadržaja i monitoring korišćenja interneta.

Forward proxy vs Reverse proxy

- **Reverse proxy:**
 - **Klijent nije svestan:** Korisnici obično nisu svesni prisustva reverse proxyja; za njih izgleda kao da direktno pristupaju serveru.
 - **Služi serverima:** Reverse proxy deluje kao posrednik između interneta i servera koji se nalaze iza njega. Kada korisnici šalju zahteve serveru, oni prvo stižu do reverse proxyja.
 - **Skrivanje servera:** Cilj je sakriti postojanje i identitet pozadinskih servera, povećavajući bezbednost.
 - **Optimizacija i balansiranje opterećenja:** Koristi se za balansiranje opterećenja među više servera, keširanje sadržaja, SSL enkripciju i optimizaciju performansi.

Primer - *reverse proxy*

- Upotrebom reverse proxy-ja dobijamo pristup environment varijablama i CoreDNS servisu unutar klastera

Primer - *nginx.conf*

```
server {  
    listen 3000;  
  
    location /api/ {  
        proxy_pass http://tasks-service.default:8000/;  
    }  
    location / {  
        root /usr/share/nginx/html;  
        index index.html index.htm;  
        try_files $uri $uri/ /index.html =404;  
    }  
    include /etc/nginx/extra-conf.d/*.conf;  
}
```

Ingress

- Ako u okviru klastera postoji frontend koji komunicira sa ostalim servisima u klasteru, onda ti drugi servisi mogu da budu tipa ClusterIP
 - Ne može da im se pristupi spolja
 - Povećana bezbednost
- Ako ne postoji frontend unutar klastera ovakva dodatna bezbednost se može omogućiti uvođenjem Ingress-a
- Nije alternativa za upotrebu bilo kog servisa, već za upotrebu servisa koji omogućava eksterni pristup (NodePort, LoadBalancer)
- Link do dokumentacije: <https://kubernetes.io/docs/concepts/services-networking/ingress/>

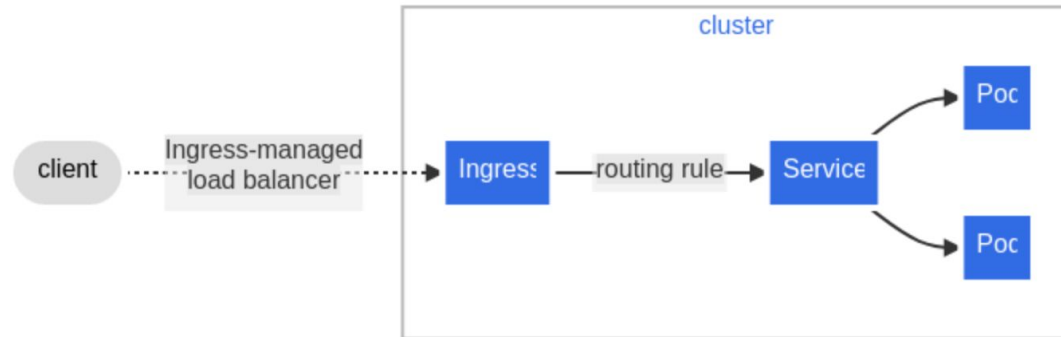
Ingress

- Bezbedno rešenje koje nam omogućava pristup servisima unutar klastera
- Upotrebom Ingress-a servisi nemaju eksterni pristup (ne izlaže se port)
- Zašto bi se koristio Ingress umesto servisa koji omogućava eksterni pristup?
 - Povećana bezbednost klastera
 - Bolje rutiranje saobraćaja
 - Poboljšan load balancing
 - Automatsko skaliranje

Ingress

- Ingress se sastoji iz:
 - **Ingress Objekta**
 - resurs koji definiše pravila za usmeravanje spoljnog saobraćaja ka servisima unutar klastera
 - definiše pravila rutiranja koja upravljaju načinom na koji se spoljni zahtevi usmeravaju ka internim servisima.
 - **Ingress Kontrolera**
 - zadužen za ispunjavanje zahteva Ingress Objekta
 - služi kao gateway, a može da izvršava load balancing, autentifikaciju, autorizaciju, itd.
 - Primer Ingress kontrolera: Nginx, Traefic, HAProxy, ..

Ingress



Ingress

- Dodavanje ingress kontrolera unutar minikube:
 - naredba: `minikube addons enable ingress`
- Provera da je dodato uspešno:
 - naredba: `kubectl get namespace`
- Primena ingress.yaml fajla:
 - naredba: `kubectl apply -f ingress.yaml`
- Prikaz IP adrese ingress servera:
 - naredba: `kubectl get ingress`
- Dodatno - mapiranje IP adrese na hostname:
 - `sudo nano /etc/hosts`
 - upisati IP adresu i hostname unutar fajla, npr. 192.168.59.100 tasks-server.com

Primer - *ingress.yaml*

```
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: my-ingress
  labels:
    name: my-ingress
spec:
  rules:
  - host: tasks-server.com
    http:
      paths:
      - pathType: Prefix
        path: "/" #propuštanje svih putanji
      # path: "/tasks" #propuštanje samo tasks putanje
    backend:
      service:
        name: tasks-service
        port:
          number: 8000
```

Dodatni materijali

- <https://github.com/ramitsurana/awesome-kubernetes>
- <https://cloud.google.com/kubernetes-engine/kubernetes-comic/>
- <https://kubernetes.io/docs/home/>