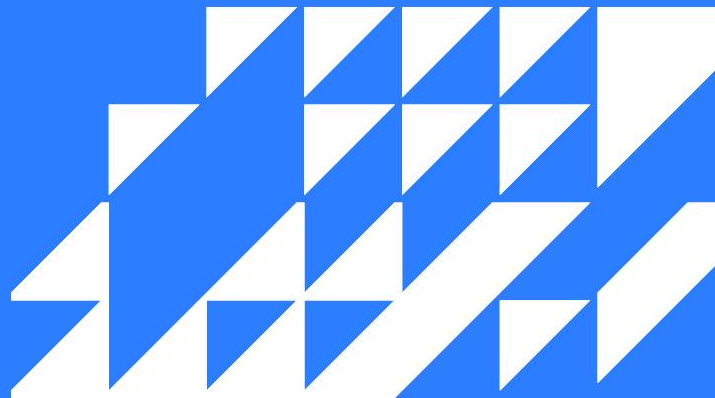


entigo

Kubernetese põhitõed

Martin Vool

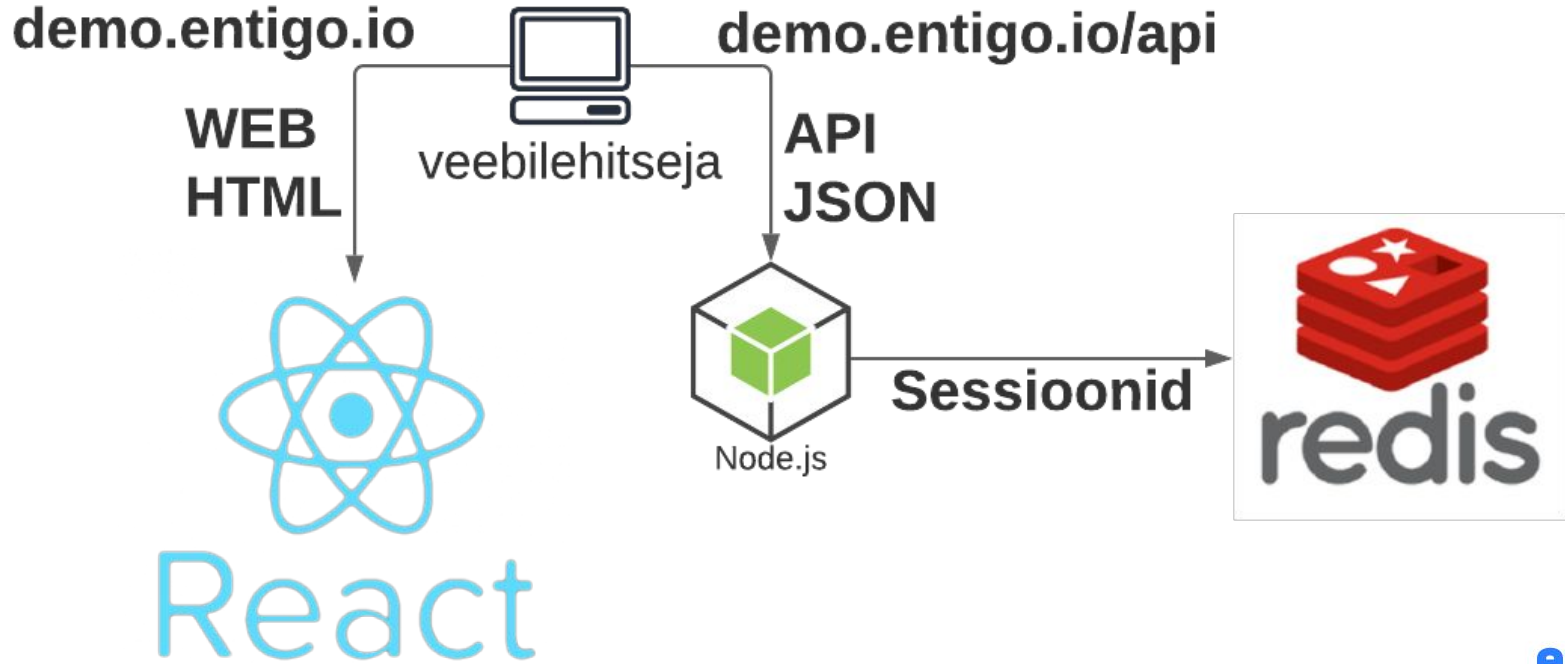
Tallinn 2021



Rakendus

- **WEB**
Nodejs + react, staatiline html+javascript browseris
- **API**
Nodejs API, vastab json formaadis
- **Redis**
noSQL andmebaas sessiooni hoidmiseks

Rakendused



Paketeerimine Dockeriga

- Levinud tarkvara on juba paketeeritud
- Efektiivne ruumikasutus
- Ei ole sõltuvuste põrgut
- Töötab minu arvutis ja ka sinu arvutis

Paketeerimine Dockeriga

- API readme.txt

Node 16. Vajab Redis andmebaasi. Ehitamiseks kasuta

\$ npm run build

Käivitamiseks

\$ npm start

Seadista keskkonna muutujad:

SESSION_SECRET ja REDIS_HOST

Paketeerimine Dockeriga

- API Dockerfile

FROM node:16	#Kasutame node 16
WORKDIR /app	#Määrame töötamise kausta
ENV SESSION_SECRET="CHANGEME"	#Määrame vaike väärtuse
ENV REDIS_HOST="localhost"	#Määrame vaike väärtuse
CMD npm start	#Rakenduse käivitamise käsk
COPY . /app	#Pane koodi /app kausta
RUN npm run build	#Ehitame rakenduse

Paketeerimine Dockeriga

- Redis

Kasutame ametliku redise dockeri konteinerit

https://hub.docker.com/_/redis



redis ☆

Docker Official Images

Redis is an open source key-value store that functions as a data structure server.

Paketeerimine Dockeriga

- WEB readme.txt

Node 16. Ehitamiseks

\$ npm install && npm run build

Serveerida "build" kausta tekkiv HTML

API komponent peab vastama relatiivse kausta /api kaudu.

Paketeerimine Dockeriga

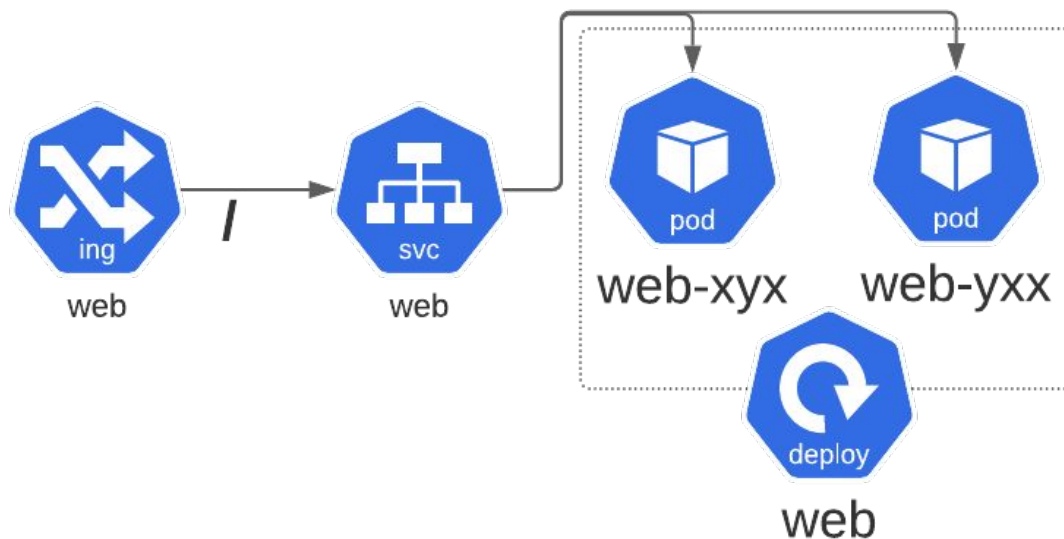
- WEB Dockerfile

```
FROM node:16-slim as build          #Kasutame node 16|
WORKDIR /app                        #Määrame töötamise kausta
COPY . /app                         #Pane kood /app kausta
RUN npm install && npm run build     #Ehitame HTML
```

```
FROM nginx:1.20-alpine              #Kasutame nginx'i
#Kopeerime eelmises konteineris ehitatud HTMLi
COPY --from=build /app/build /usr/share/nginx/html
```

Käivitamine Kuberneteses

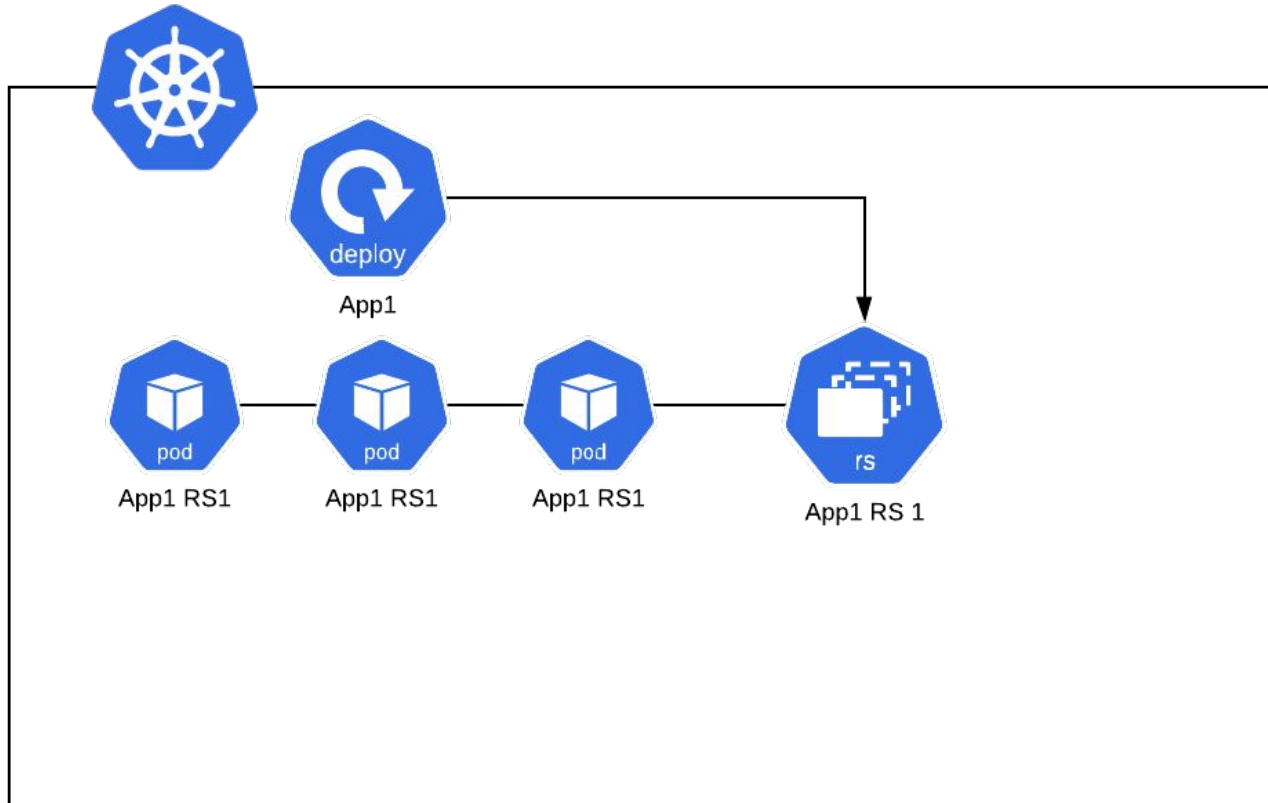
- WEB jaoks vajalikud objektid



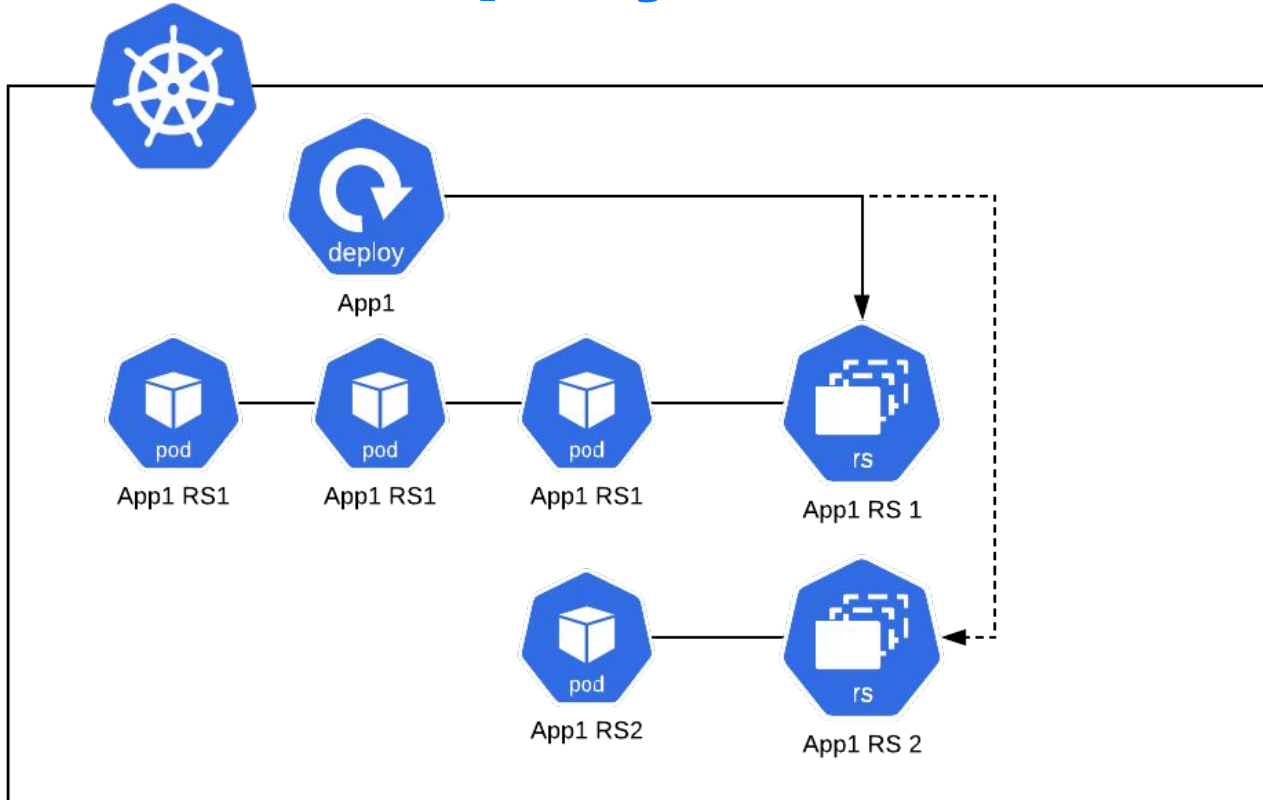
Deployment

- Määrab **Pod**-ide arvu, ehk skaleerimine
- Teostab uuendusi ja määrab strateegia(Recreate, Rolling)
- Muudatuste tagasikerimine, ehk “rollback”
- Kasutatakse “stateless” rakenduste jaoks
- Tekitab ja haldab ReplicaSet objekte
- Sisaldab **Pod**-i kirjeldust

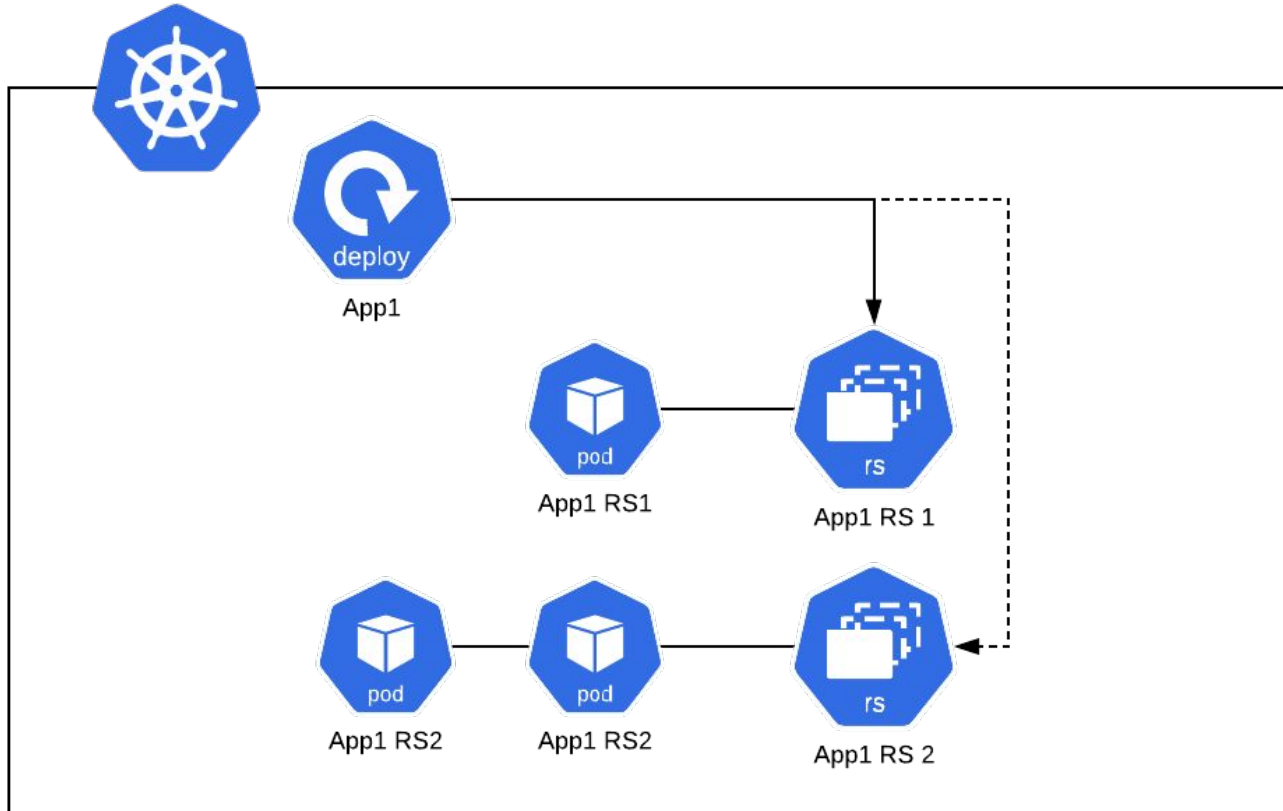
Deployment



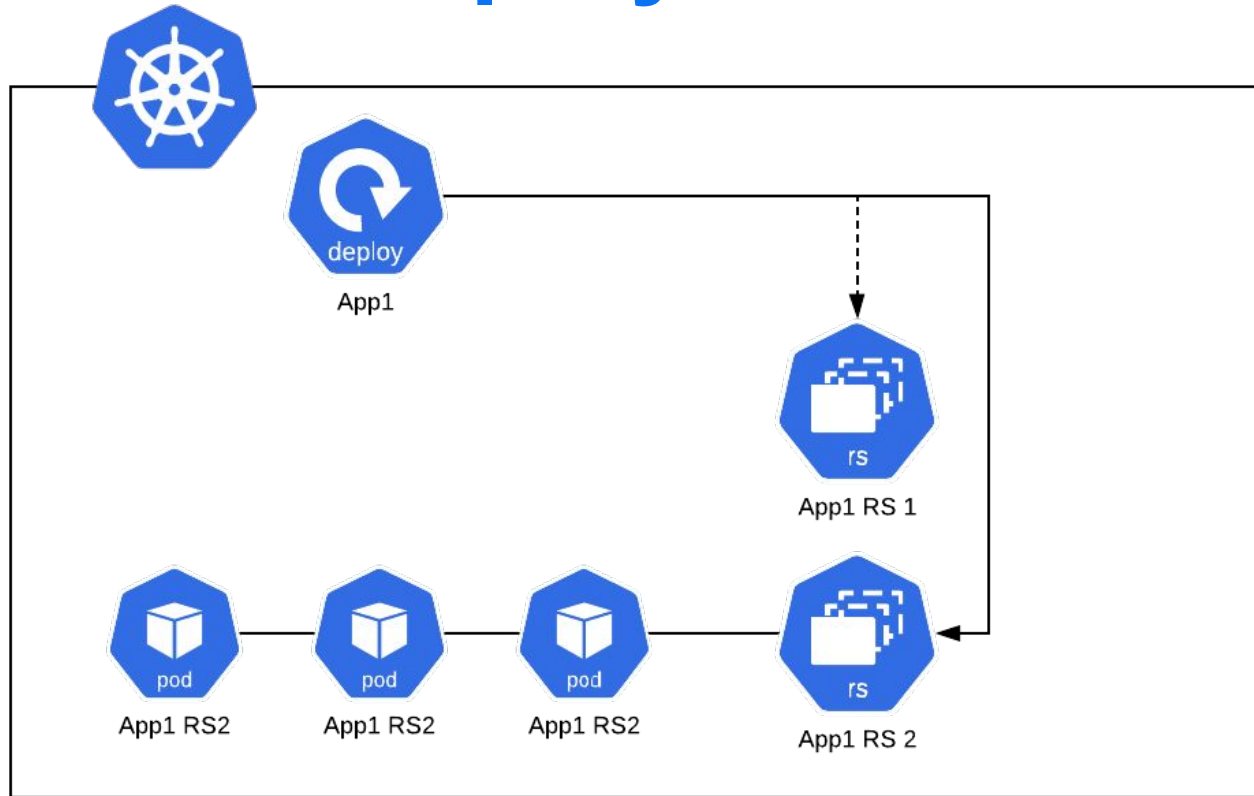
Deployment



Deployment

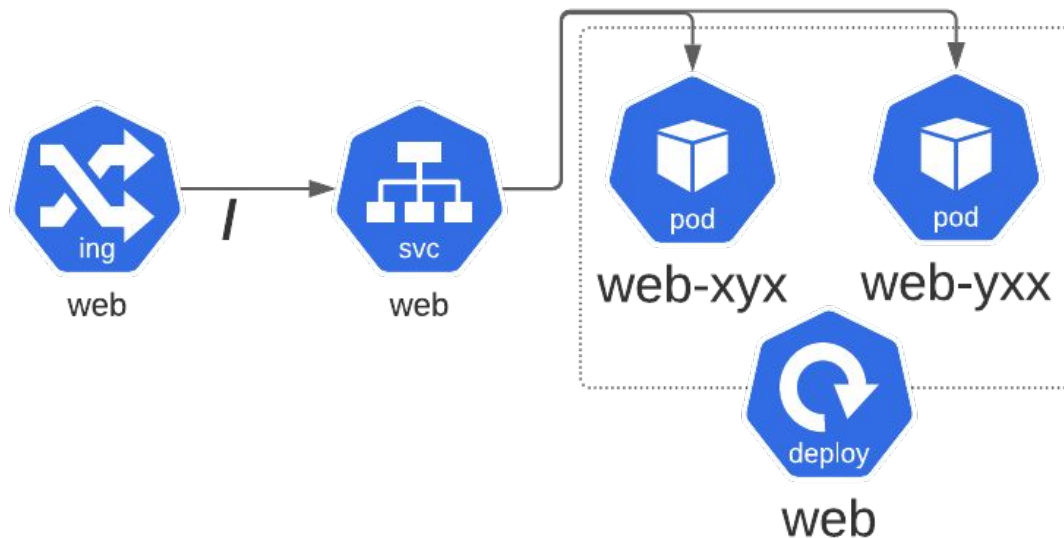


Deployment



Käivitamine Kuberneteses

- WEB jaoks vajalikud objektid



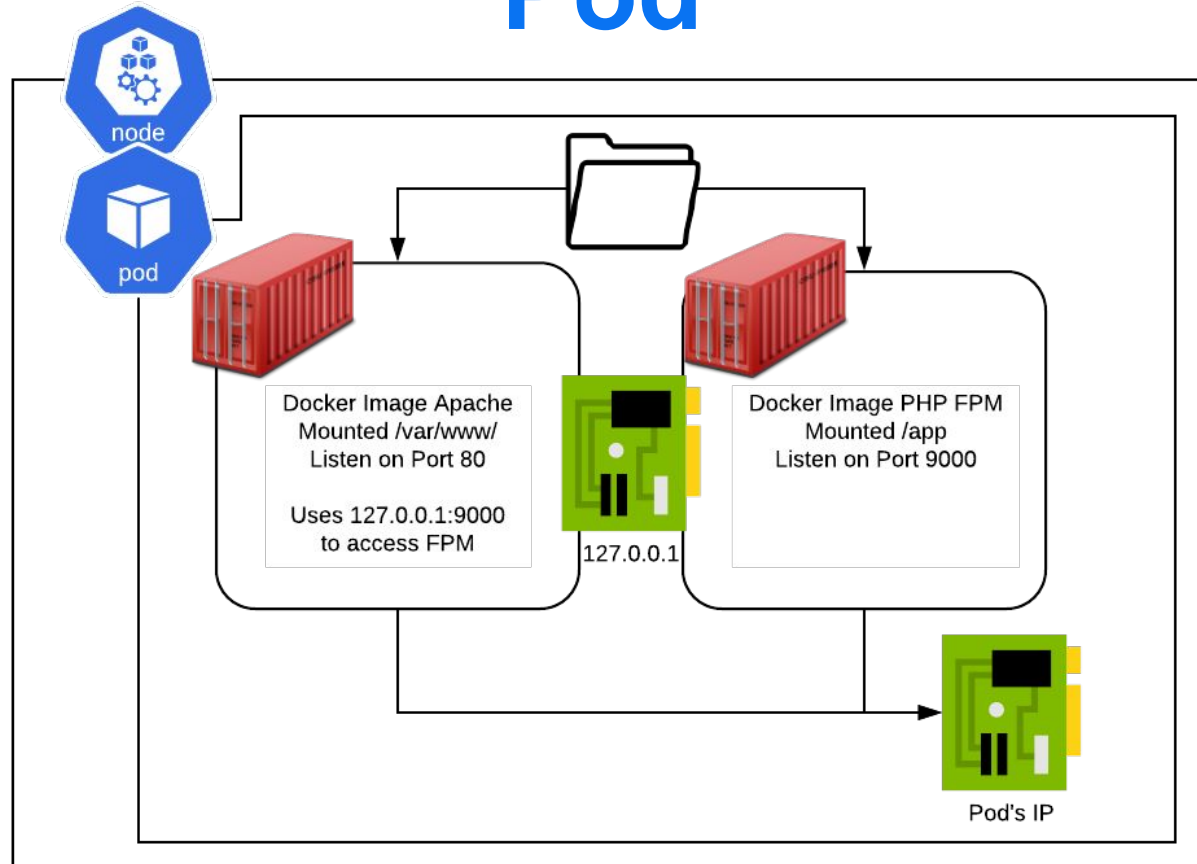
Pod

- Omab unikaalset IP aadressi
- Töötab ühe kindla serveri peal oma elu lõpuni
- Skaleeritakse pod-e, mitte konteinereid podis
- Uuenduste puhul luuakse uus pod ja vana kustutatakse
- Koosneb ühest kuni mitmest konteinerist

Pod

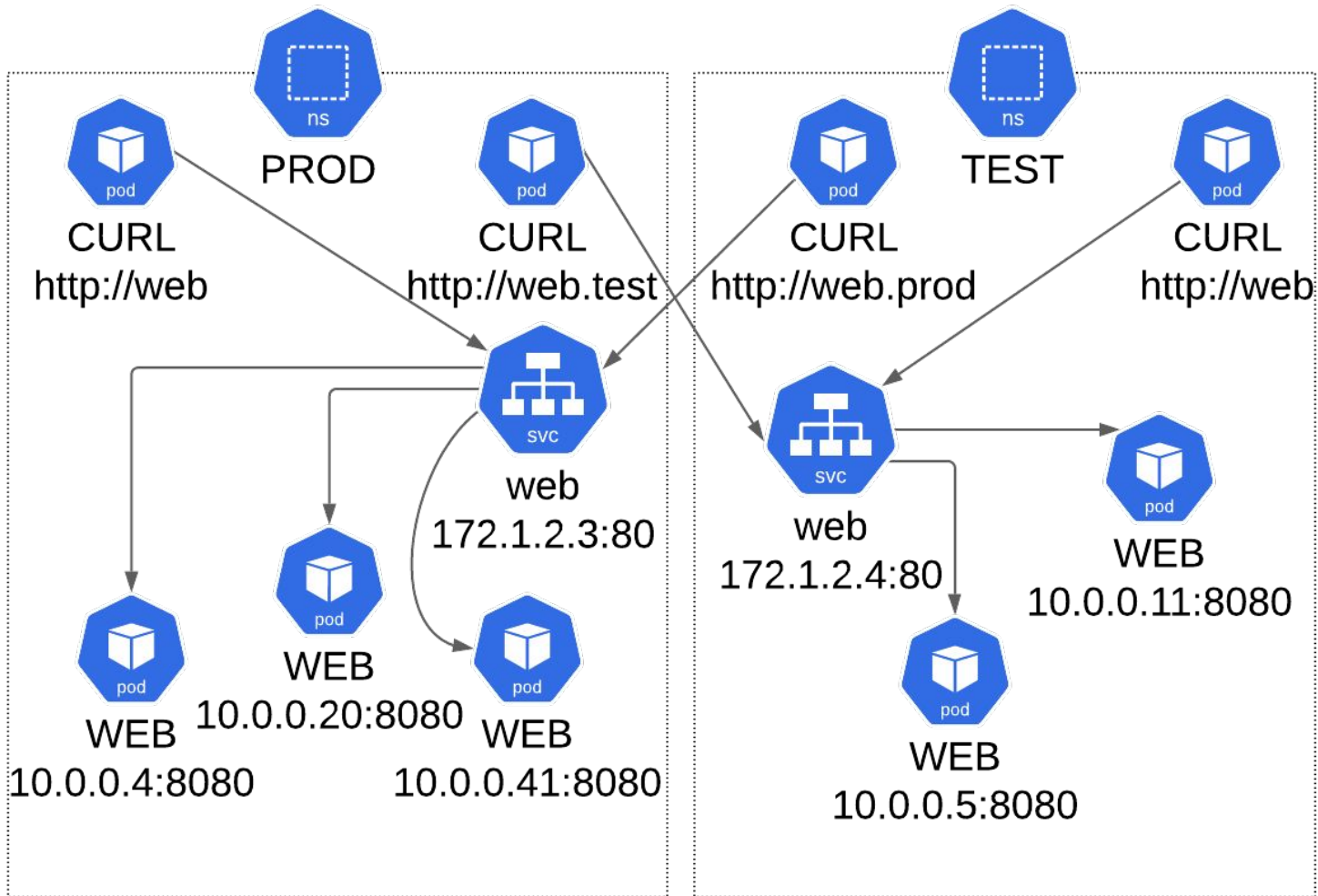
- Kui Pod sisaldab mitut konteinerit
 - Igal konteineril on oma protsessi tabel
 - Ei näe teiste konteinerite protsesse
 - Omavad enda failisüsteemi
 - Võivad omada jagatud kaustu teiste konteineritega
 - Jagavad ühte IP aadressi

Pod



Service

- Suunab võrgu liiklust Pod-ide suunas kindlatel võrgu portidel
- Nimeserveri kanne mis on service-iga sama nimega
- Teenuse hosti nimed on kujul <service nimi>.<namespace nimi>
- Samas namespace-is oleva teenuse puhul piisab teenuse nime kasutamisest hosti nimena.
- Erinevad liigid: **ClusterIP, NodePort, LoadBalancer, ExternalName**



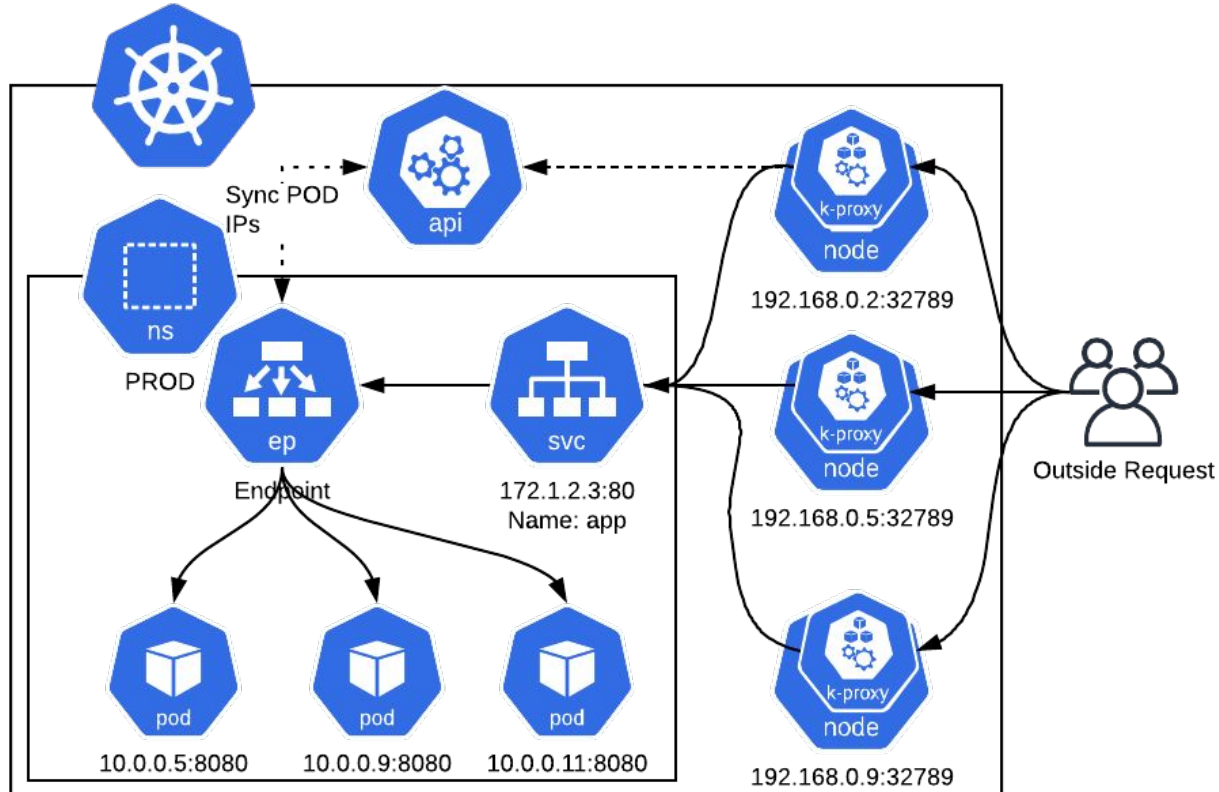
Service

- Igas kubernetese klastris on oma nimeserveri teenus.
- Iga Service sõltumata tüübist saab omanimelise kande selles nimeserveris.
- Kõikidele konteineritele seadistatakse kubernetese nimeserverid automaatselt.
- Kui kubernetese siseselt ei suudeta nime lahendada, siis küsitakse serverite nimeserverilt.

Service

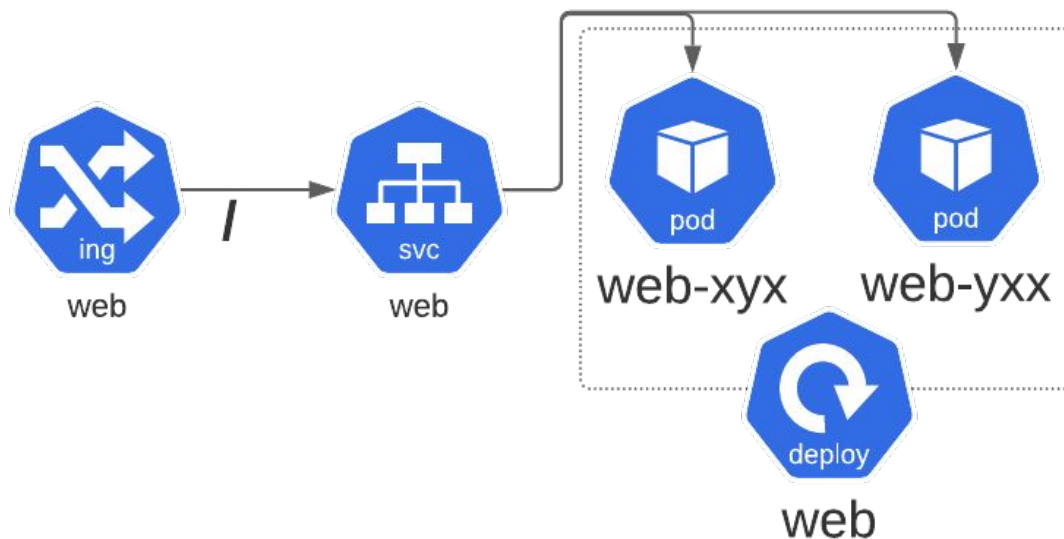
- **ClusterIP** kõige levinum tüüp. Tekitab koormusjaoturi ja nimelahenduse **pod**-ide pihta.
- **NodePort** paneb serverite võrgu peal kindla pordi kuulama ja edastab selle **pod**-ide pihta.
- **LoadBalancer** eeldab integratsiooni välise koormusjaoruriga. Näiteks AWS või Azure pilves. Automaatselt tekitatakse koormusjaotur koos seadistusega.
- **ExternalName** tekitab klasteri sees nimeserveri kande, mis on vabalt valitud CNAME tüüpi viide.

Service



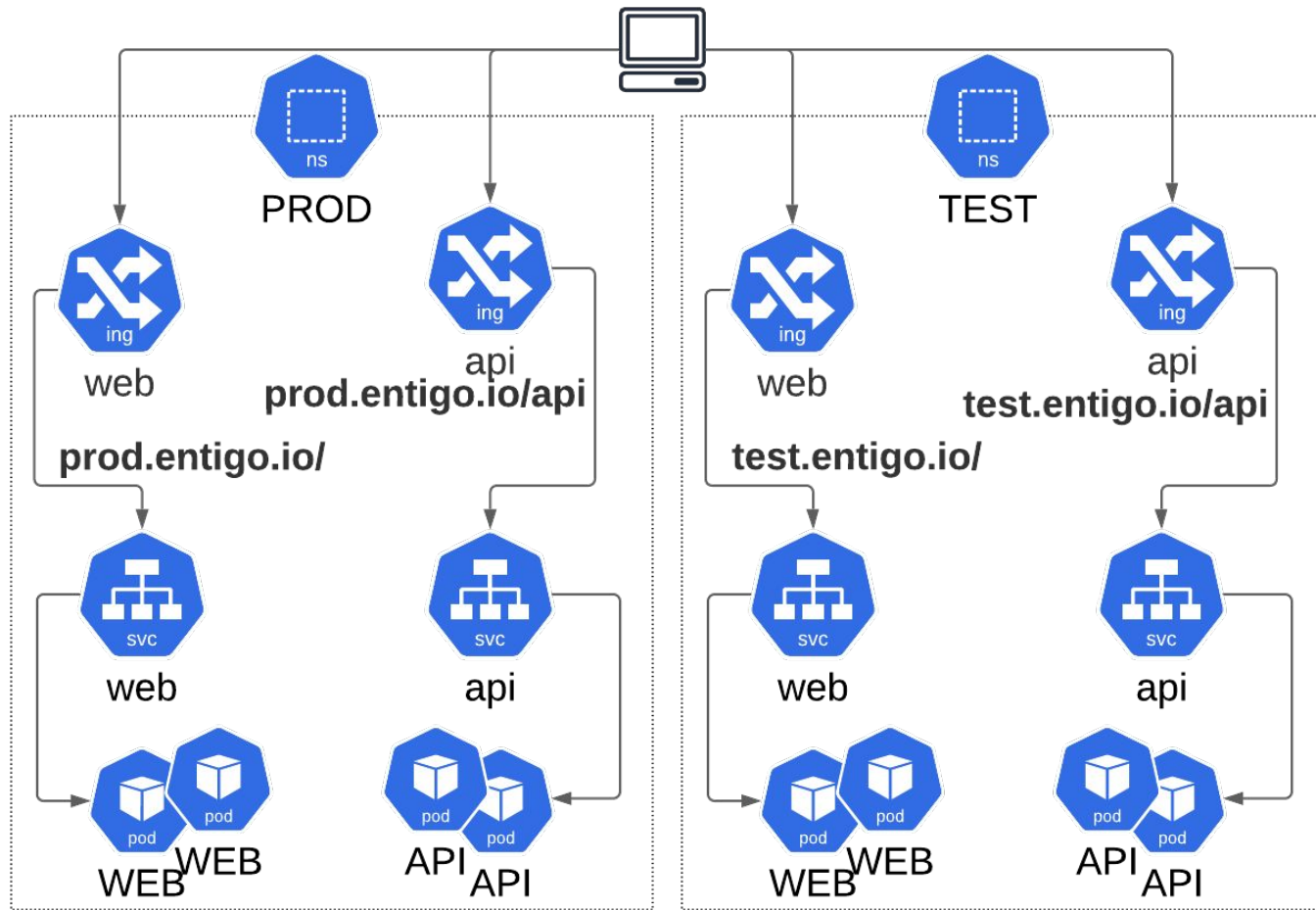
Käivitamine kuberneteses

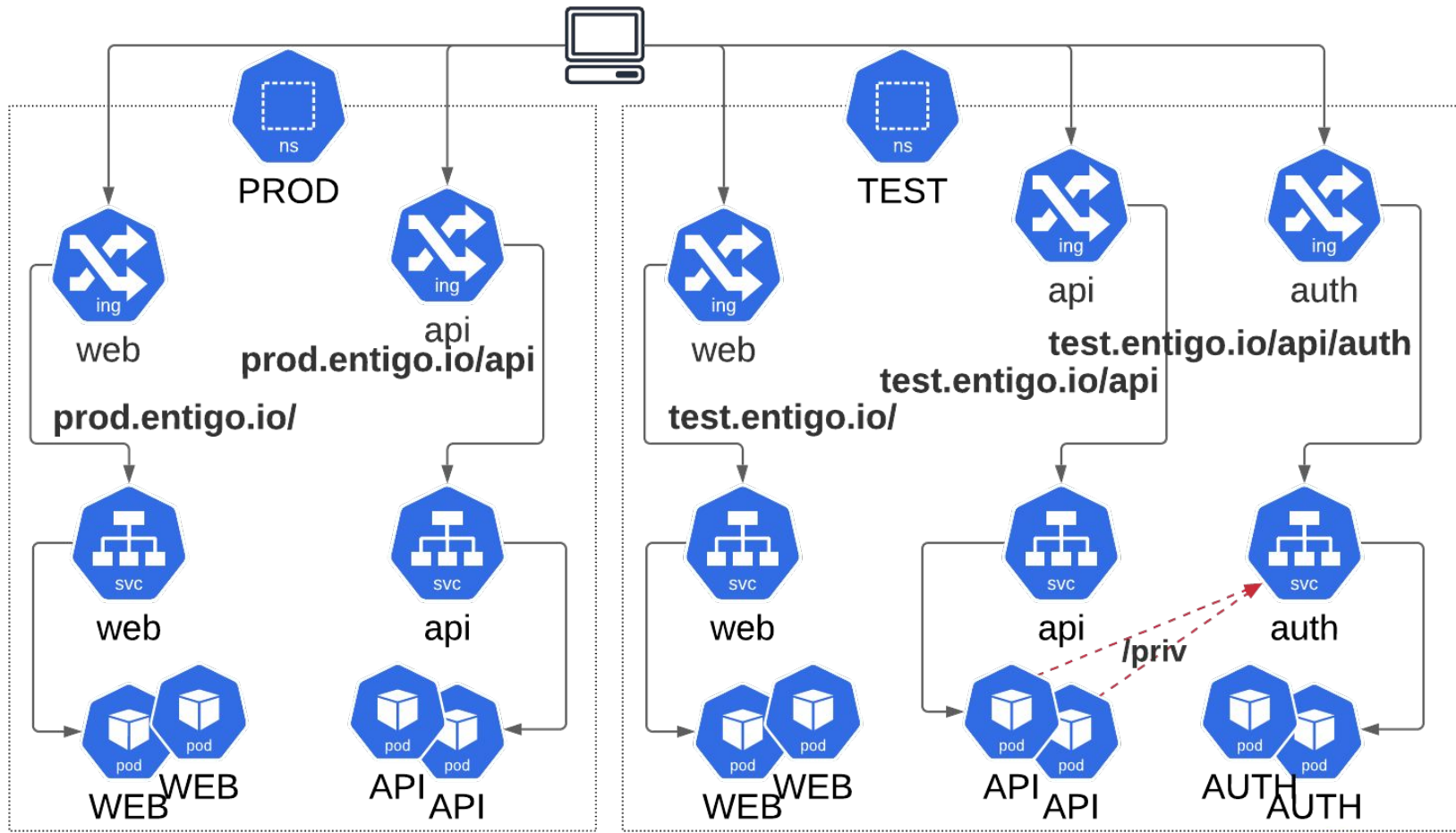
- WEB jaoks vajalikud objektid



Ingress

- Objekt on kirjeldus väljaspool kubernetest pakutavast teenusest
- Tavaliselt seadistatakse hosti nimi ja kaust millele vastab mingi kubernetese service
- Sageli määratakse ka millist SSL sertifikaati kasutada.
- Näiteks:
API ingress: demo.entigo.com/api* suuna api service 80 pordi pihta
WEB ingress: demo.entigo.com/* suuna web service 80 pordi pihta



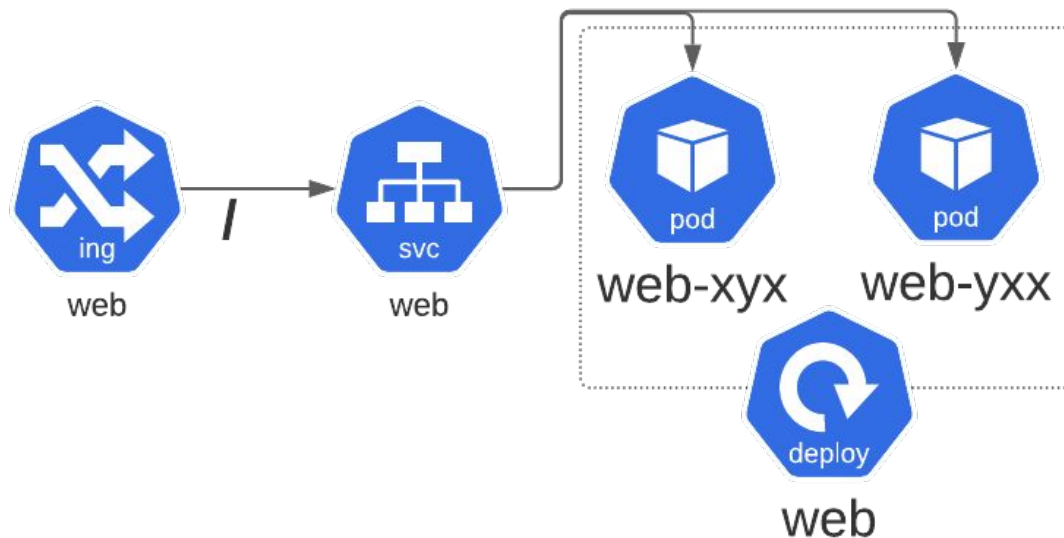


Ingress

- Ingressi objekte loevad erinevad koormusjaoturi tarkvarad, näiteks haproxy või nginx kontrollid
- Objektidest saadud info põhjal koostatakse automaatselt koormusjaoturi seadistus
- Backendiks on pod-ide IP aadressid
- Enamasti teostatakse HTTPS koormusjaoturites

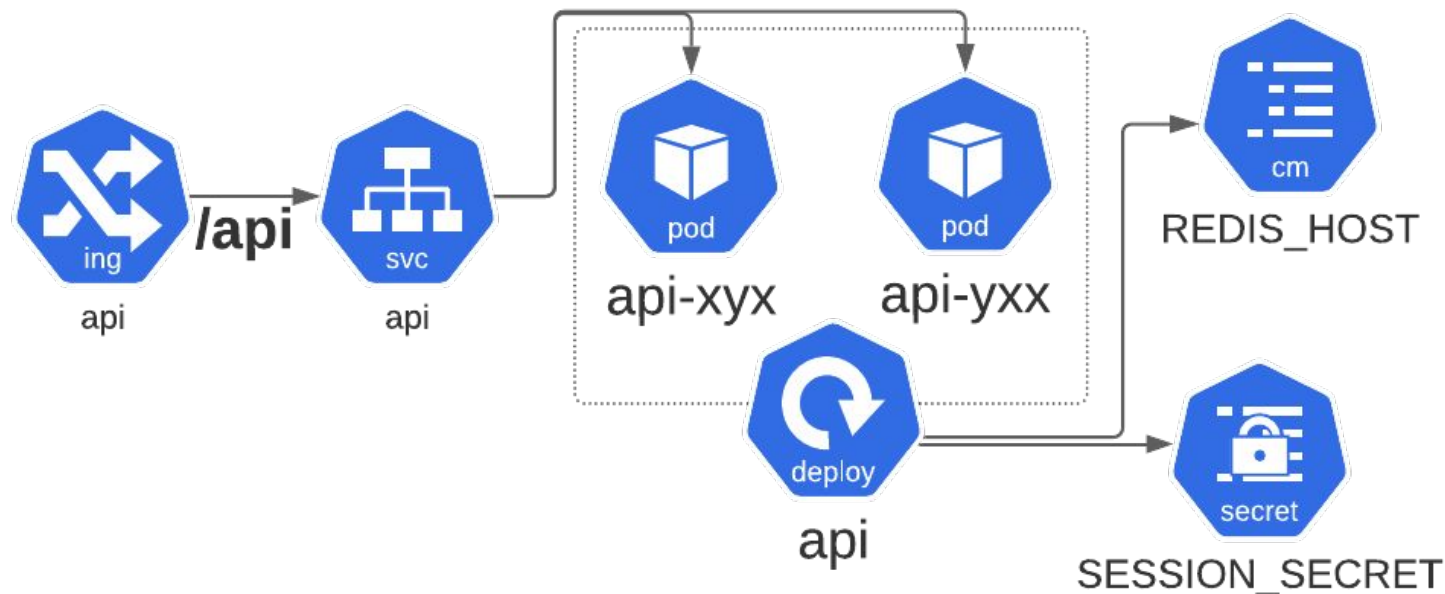
Käivitamine Kuberneteses

- WEB jaoks vajalikud objektid



Käivitamine Kuberneteses

- API jaoks vajalikud objektid



Configmap ja Secret

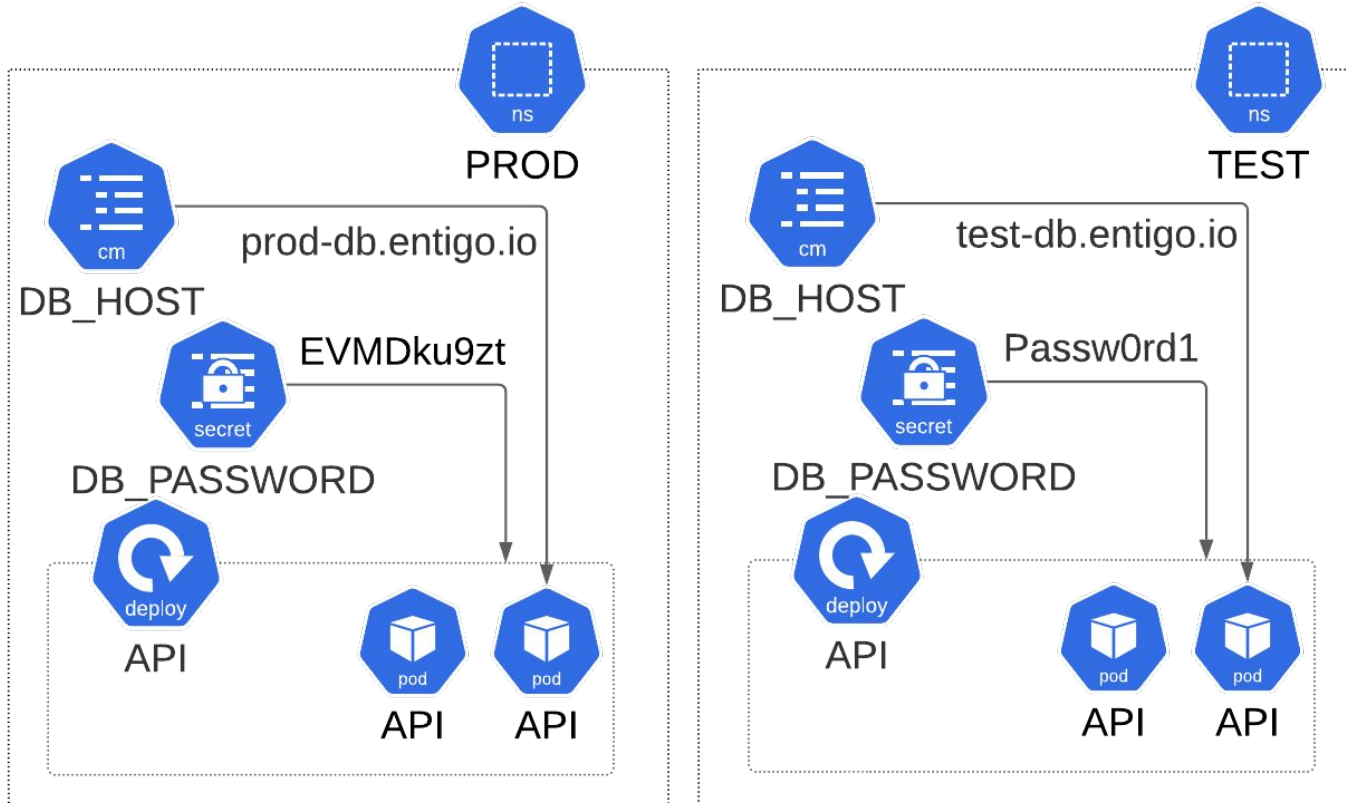
- **Configmap** sisaldab endas keskkonna muutujaid või faile
- Mitmed erinevad rakendused saavad kasutada ühiseid seadistusi või nende osi
- Erinevates keskkondades on sama nimega, aga erinevate väärtustega seadistused

Configmap ja Secret

- **Secret** väga sarnane configmap-iga.
- Sisaldab endast paroole, sertifikaate, võtmeid
- Sensitiivse info eristamiseks
- Sageli on erinevad kasutaja õigused

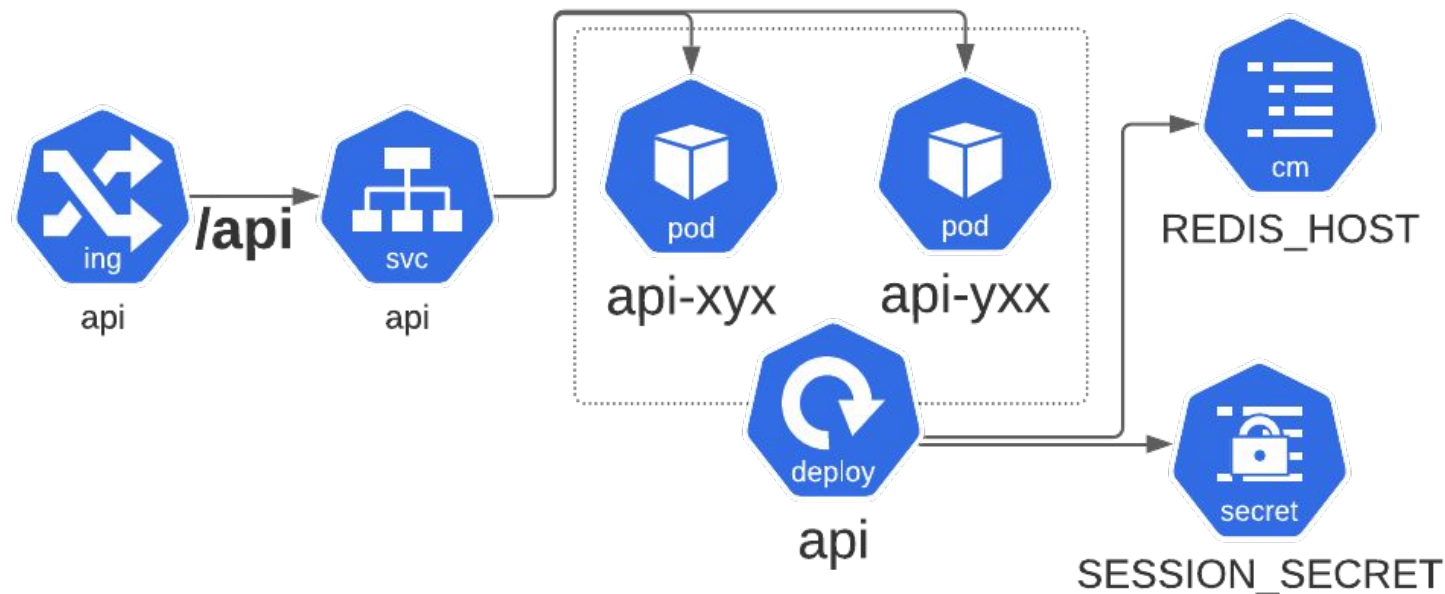
Näiteks võib olla õigus lugeda configmap-e, aga mitte secret-eid

Configmap ja Secret



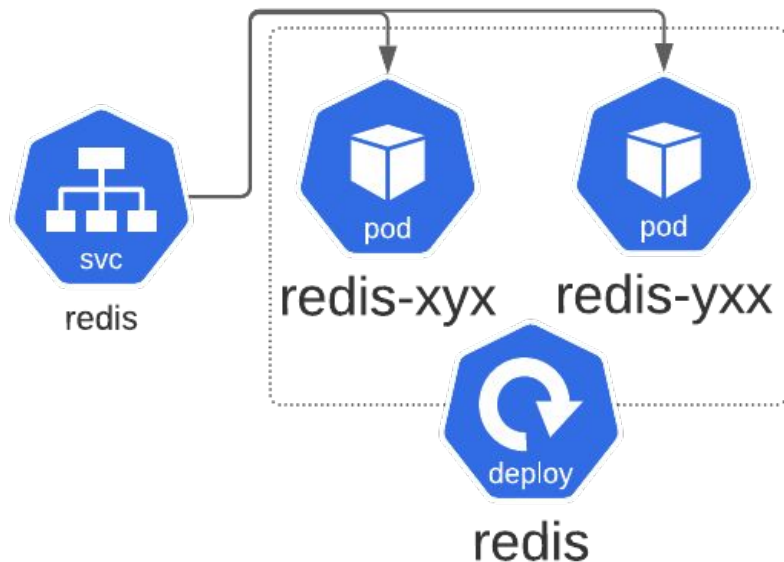
Käivitamine Kuberneteses

- API jaoks vajalikud objektid



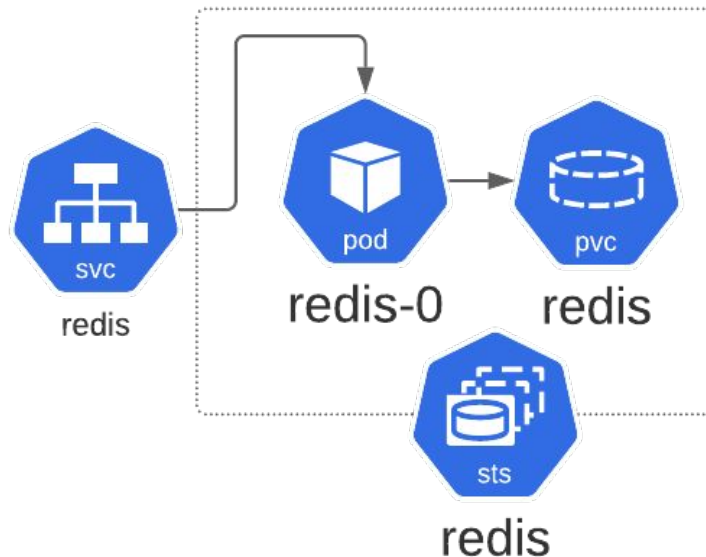
Käivitamine Kuberneteses

- Redis jaoks vajalikud objektid(Ei sobi)



Käivitamine Kuberneteses

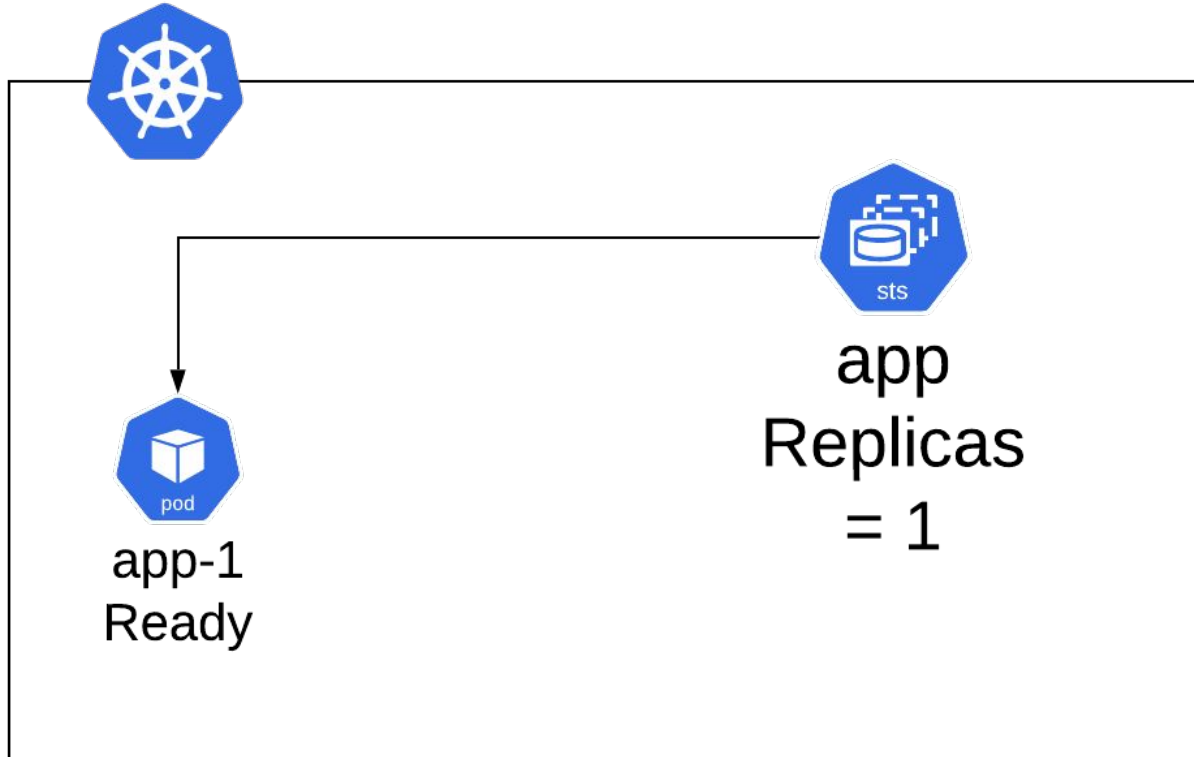
- Redis jaoks vajalikud objektid



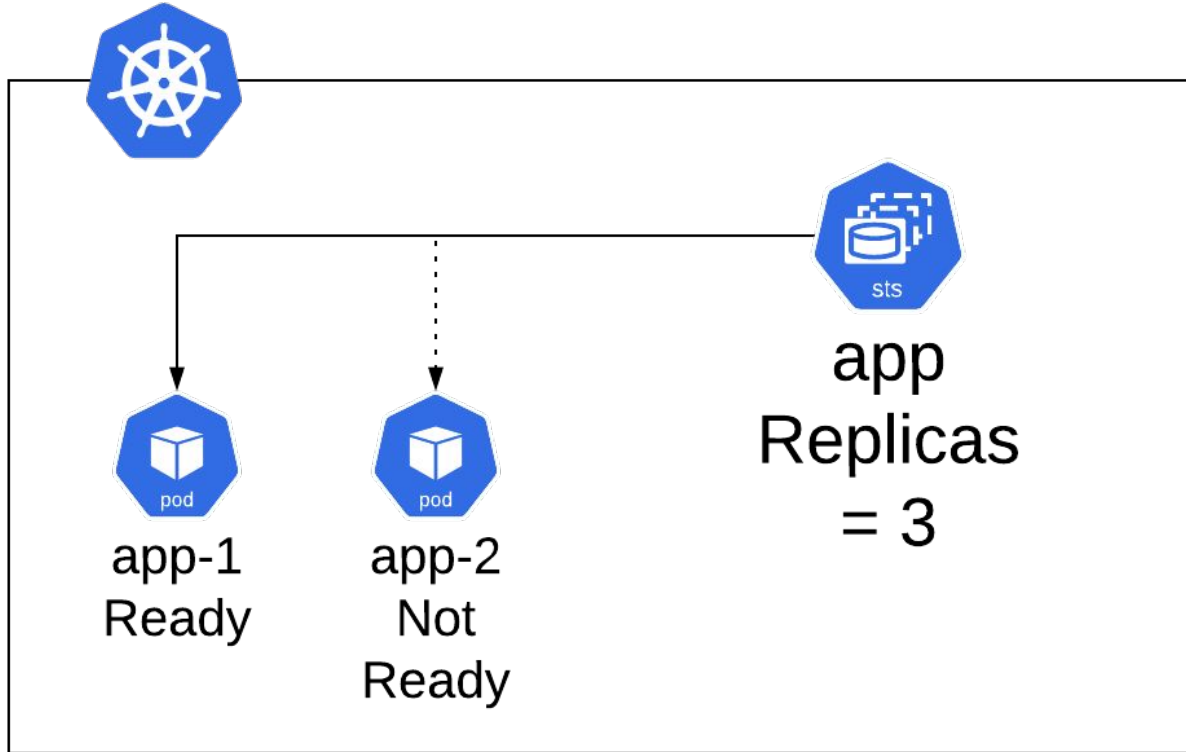
Statefulset

- Sisaldab **Pod**-i kirjeldust
- Loob ja hävitab pod-e kindlas järjekorras
- Püsivad nimed ja kettad
- Mitte kunagi ei ole rohkem pode kui määratud
- Kasutatakse “stateful” rakenduste puhul
Näiteks: redis, mysql

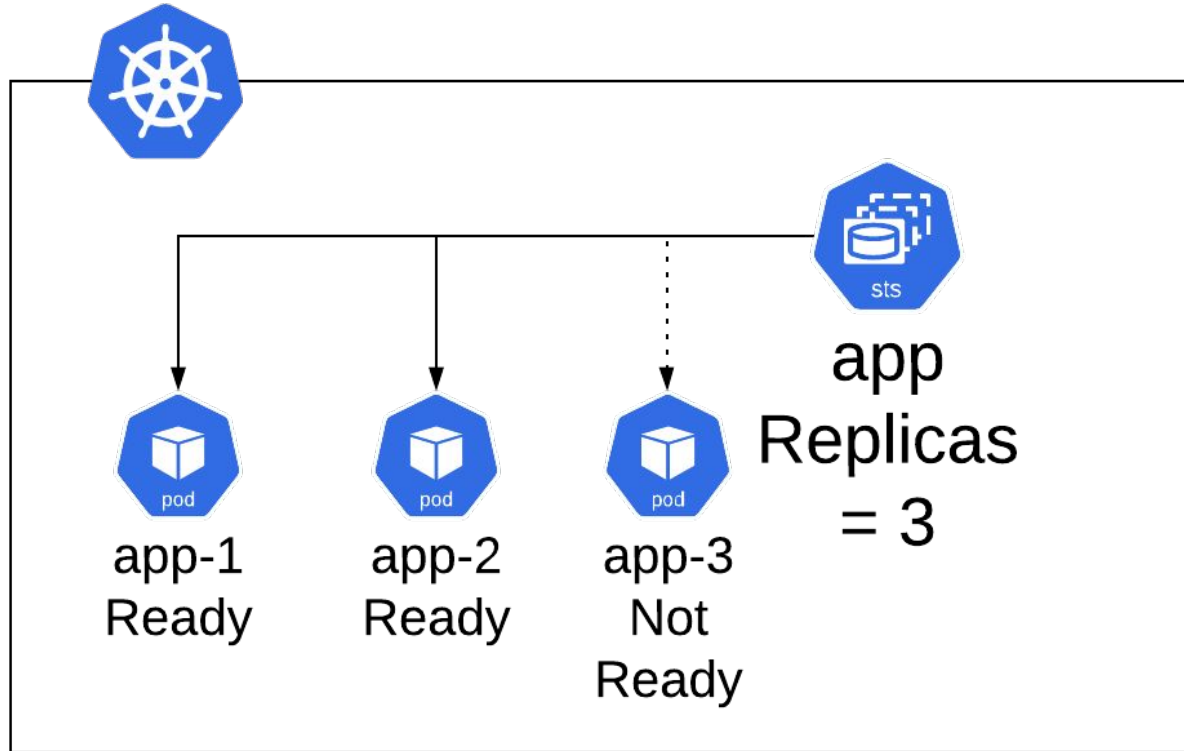
Statefulset



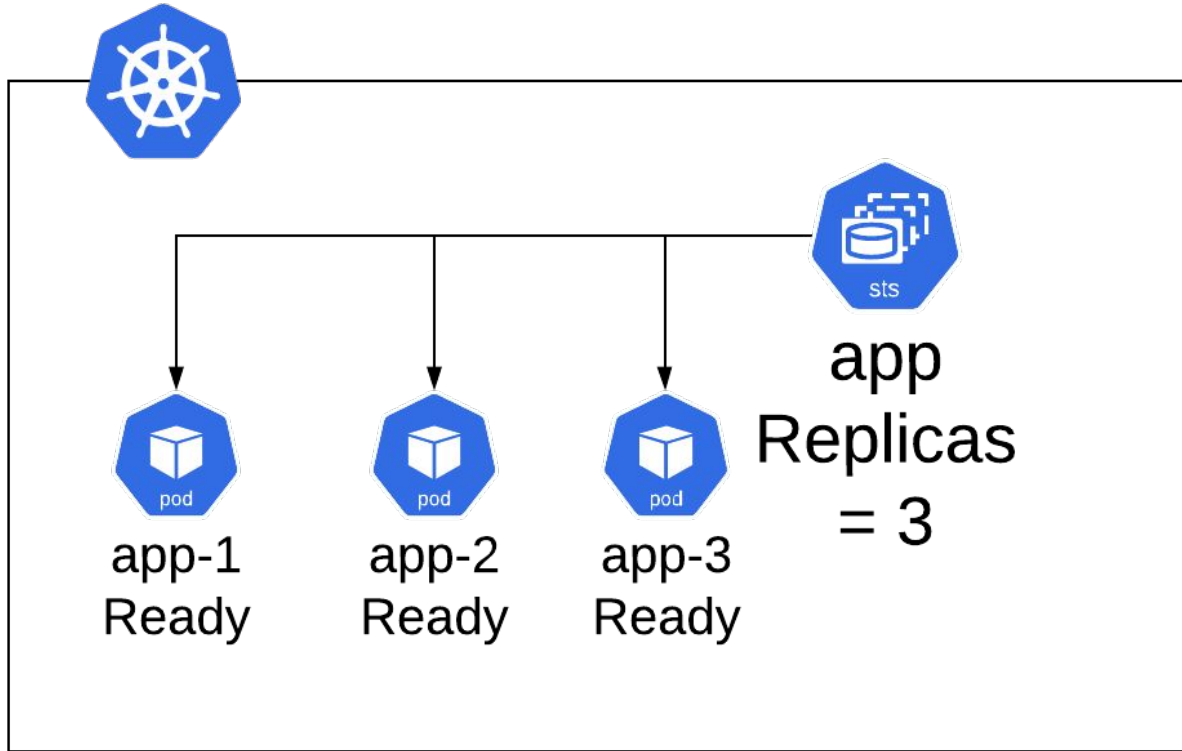
Statefulset



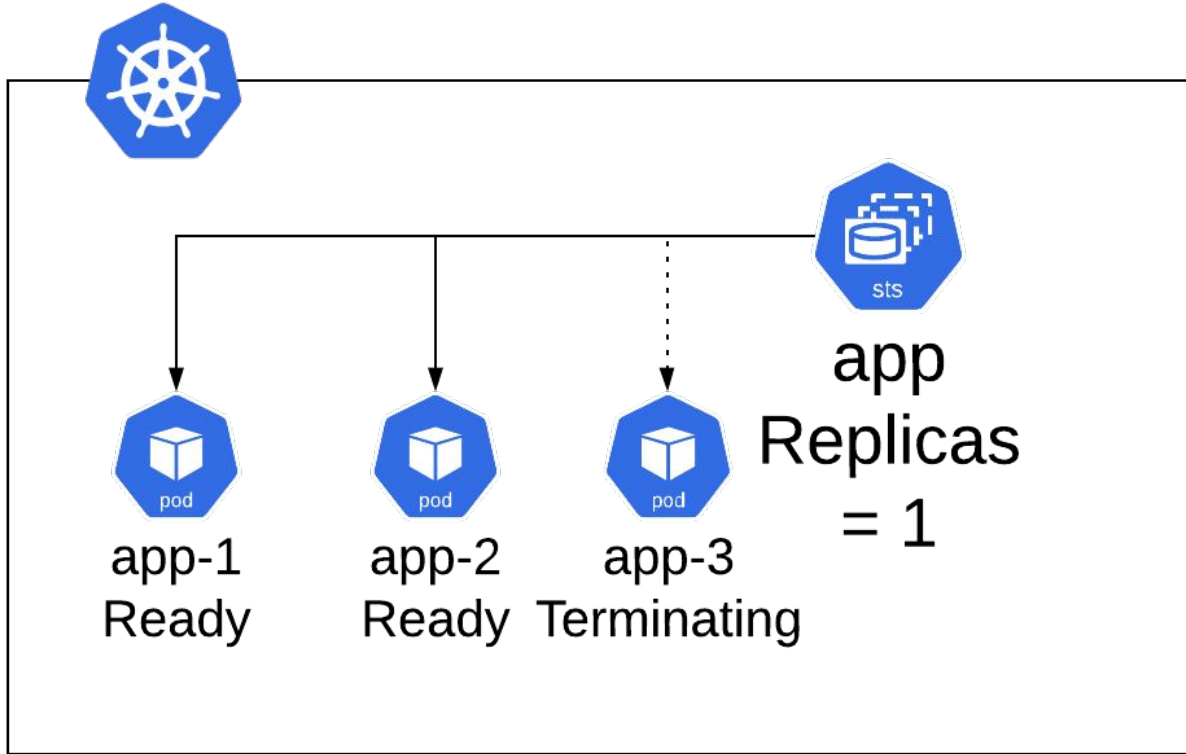
Statefulset



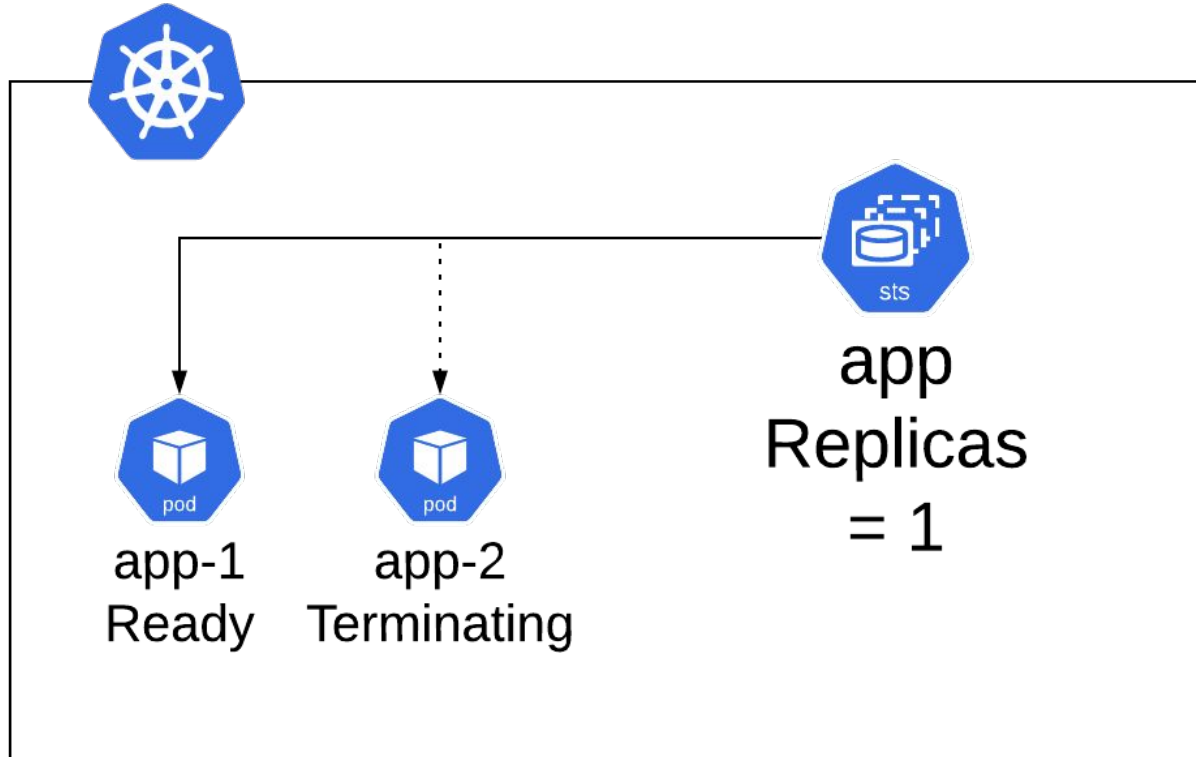
Statefulset



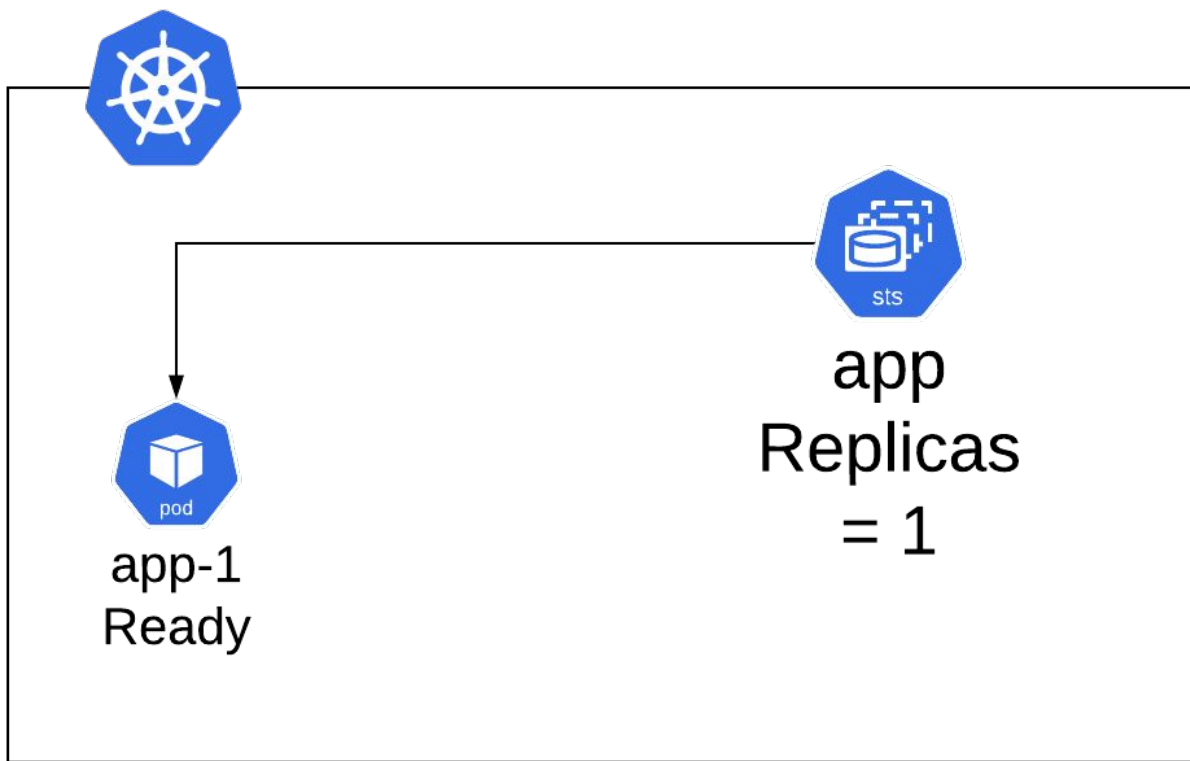
Statefulset



7. Statefulset



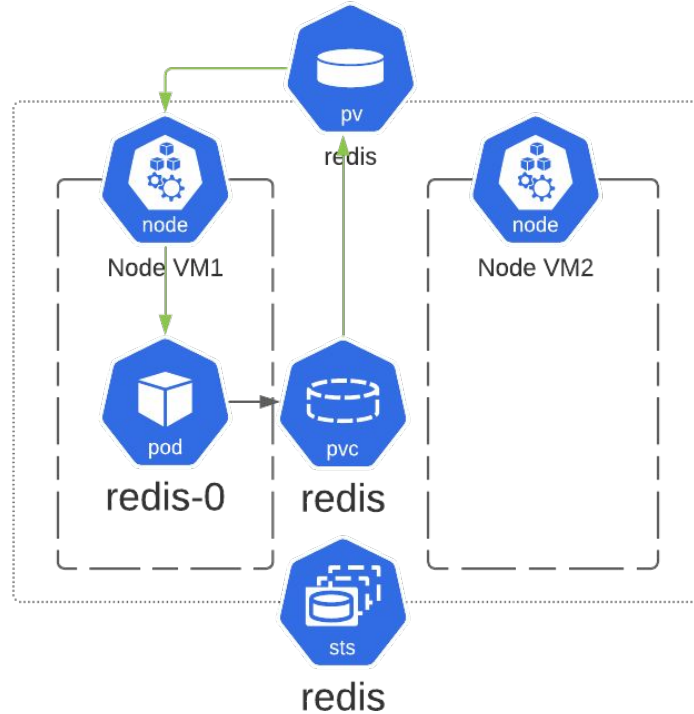
Statefulset



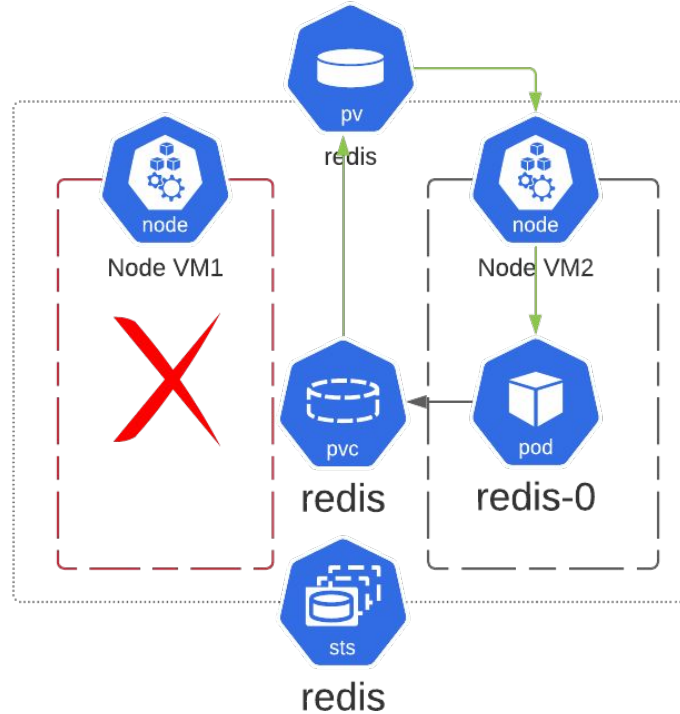
PersistentVolumeClaim

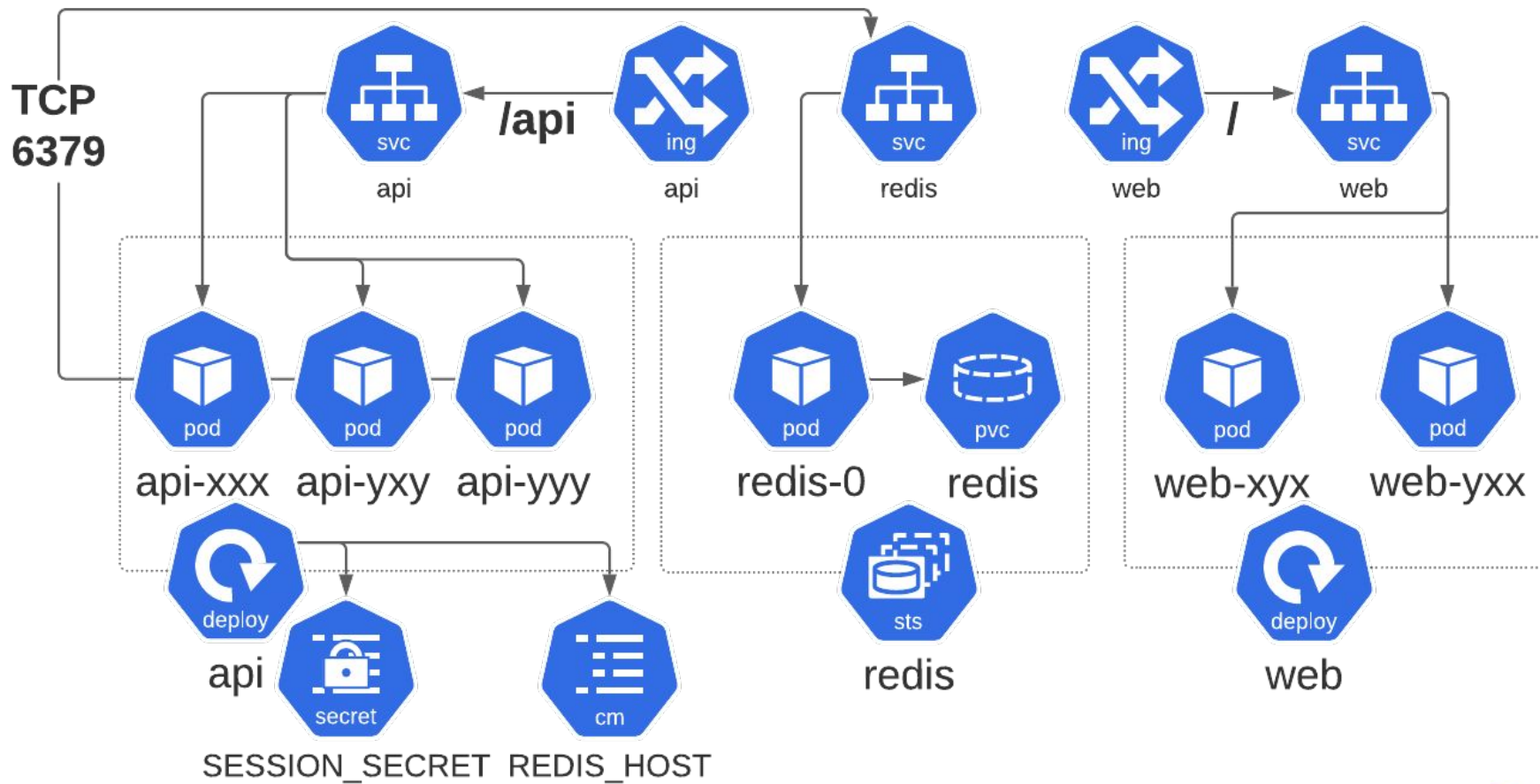
- Määrab ketta tüübi ja mahu
- Tegeleb ketta suurendamisega
- Sageli määratakse ka varundus strateegia
- Võimaldab siduda Podi kindla kettaga
- Mis saab kettast kui teda enam vaja ei ole

PersistentVolumeClaim



PersistentVolumeClaim





Rakenduse struktuur

- API/ Dockerfile + tarkvara kood
- web/ Dockerfile + tarkvara kood
- yaml/ Kubernetes objektide kirjeldused yaml formaadis
- scaffold.yaml Skaffold tööriista seadistused

Rakenduse struktuur

- `yaml/api`
`ingress.yaml` `service.yaml` `deployment.yaml` `secret.yaml` `configmap.yaml`
- `yaml/web`
`ingress.yaml` `service.yaml` `deployment.yaml`
- `yaml/redis`
`service.yaml` `statefulset.yaml`

Demo

- Rakenduste lähtekood ja kubernetese seadistus

<https://github.com/entigolabs/>

riigipilve-koolituspaev-21-kubernetese-pohitoed

- Demo video

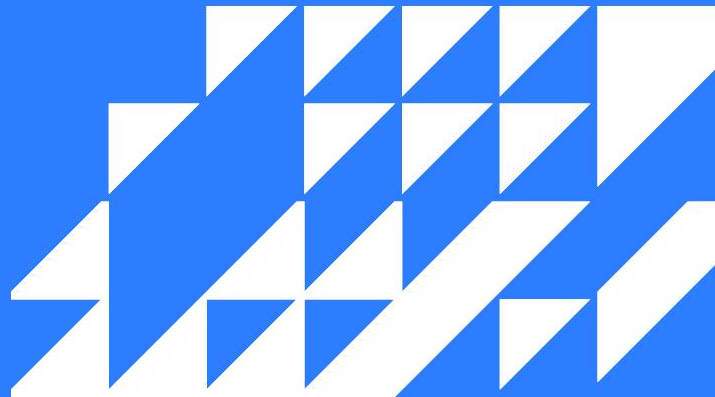
<https://youtu.be/VgDG7zknpnk>

entigo

Täna!

Martin Vool

Tallinn 2021



entigo

