Resurshantering i Kubernetes

Resurstyper

- CPU
- Minne (RAM)
- Tillfällig lagring
- Utökade resurstyper
  - T ex GPU

Resurstilldelning

- Rancher tilldelar resurser på projekt- eller namespacenivå
- Ett projekt kan ha flera namespaces
- Poddar kan endast starta om det finns tillräckligt med resurser tillgängliga

Resources

```
spec:
  containers:
    - resources:
        requests:
          cpu: "1"
          memory: 2Gi
        limits:
          cpu: "2"
          memory: 2Gi
              ephemeral-storage: 100Mi
```

`resources.requests`

- Garanterade resurser
- Används vid schemaläggning
- Poddar kan ej schemaläggas om inte resurserna är tillgängliga
  - Rullande uppdatering kräver resurser för gamla och nya poddar

`resources.limits`

- Gränser för hur mycket resurser en pod får använda
- Tilldelas om resurserna finns tillgängliga

CPU

- "Stateless resource"
- `requests`: CPU som podden alltid kan använda
- `limits`: CPU som podden får använda
  - Anti-pattern

CPU - rekommendation

- Sätt `requests` till ett "genomsnittsvärde"
- Använd `limits` för "bursting"
- `limits` kan oftast lämnas tom

Minne

- "Stateful resource"
- `requests`: RAM som podden alltid kan använda
- `limits`: RAM som podden får använda
  - Podden dödas om någon annan pod behöver minnet för `requests`

Minne - rekommendation

- Sätt `limits.requests` till den mängd RAM din applikation behöver vid vanligt bruk
- Var försiktig med `limits.memory` - utmärkt för "burst", risk för långtidsanvändning
  - Kom ihåg att vissa program använder allt minne de kommer åt för cachning

Tillfällig lagring

- T ex lagring i `emptyDir`
- "Stateful resource"
- `requests`: RAM som podden alltid kan använda
- `limits`: RAM som podden får använda
  - Podden dödas om någon annan pod behöver lagringen för `requests`
  - Podden dödas om `limits` passeras

Tillfällig lagring - rekommendationer

- Används oftast inte
- Sätt `limits.ephemeral-storage` till ett värde som inte nås vid vanligt bruk

Sammanfattning

- Resurser tilldelas till projekt
- Lås inte resurserna om de inte behöver vara tillgängliga hela tiden
  - `requests` för resurserna som måste finnas tillgängliga hela tiden
  - `limits` för att hantera "burst"
  - Utvärdera resursbehoven med Grafana (metrics)
- Rullande uppdateringar kräver "dubbla" resurser
  - Gamla och nya poddar samtidigt