Klusterin asennus

> ⚠ **WANHA!**
>
> Ajantasaset ohjeet löytyy tuolta https://github.com/tracon/ansible-tracon/tree/master/kubernetes

Karkeat asennusohjeet QB-klusterille siltä varalta, että se täytyy joskus asentaa tyhjästä uusiksi. Tätä vois automatisoida enemmänkin.

Esivaatimukset

Vähintään 3 kpl Ubuntu-koneita. Ks. CPU- ja RAM-speksi Kyypän etusivulta.

ansible-tracon kloonattuna Linux- tai macOS-koneelle. Päivitä hosts-tiedosto.

Salasanaton SSH-pääsy rootille tai sudolliselle käyttäjälle.

K3s master

Lisää tarvittaessa Ansible-komentoihin *-bK* sudoa varten.

Työasemalla

```
ansible-playbook -l k3s-masters tracon.yml
```

Tällä pitäisi nousta yhden noden tyhjä K3s-"klusteri" master-koneelle. Ota sinne SSH ja käy huutelemassa

Master-koneella

```
mkdir -p .kube
sudo install -o omakäyttäjä -m 0600 /etc/rancher/k3s/k3s.yml .kube/config
kubectl get node
kubectl get all --all-namespaces
```

Pitäis tulla jotain ehjän näköistä.

K3s nodet

Ota master-koneelta talteen tiedoston */var/lib/rancher/k3s/server/node-token* sisältö. Laita se Ansible Vaultiin avaimelle *vault_k3s_node_token*:

Työasemalla

```
ansible-vault edit group_vars/k3s-nodes/vault
```

Sit me konffataan vähän nodeja:

Työasemalla

```
ansible-playbook -l k3s-nodes tracon.yml
```

Nyt näiden pitäs näkyä node-listauksessa:

Master-koneella

```
kubectl get node
```

helm

Klusterissa on RBAC oletuksena päällä, joten noudatetaan ohjeita Helmin asentamiseksi RBAC-enabloituun ympäristöön. Nappaa YAML-listaus kohdasta *Example: Service account with cluster-admin role* tiedostoon tai leikepöydälle ja loihe

Master-koneella

```
kubectl apply -f rbac-config.yaml
helm init --service-account tiller --history-max 200
```

nginx-ingress

K3s:n oletus on Traefik, mutta koska Japsu on laiska eikä jaksa opetella miten Traefik toimii cert-managerin kanssa, em. stepit jättää traefikin asentamatta. Asennetaan sen sijaan *ingress-nginx* koska sitä Japsu osaa jo käyttää.

Master-koneella

```
helm install --name nginx-ingress --namespace nginx-ingress --set rbac.create=true,controller.kind=DaemonSet,
controller.service.type=ClusterIP,controller.service.externalIPs="{91.105.252.81,91.105.252.82,91.105.252.83}"
stable/nginx-ingress

kubectl get all -n nginx-ingress
```

Podin nimeltä *tiller-deploy* pitäisi nousta sinne hiljakseen.

cert-manager

webhook.enabled=false tarvitaan, koska K3s:stä puuttuu joku kokeellinen rajapinta jota cert-damager haluais muuten käyttää. Vaikutus on vissiin se, että cert-damagerin CRD:itä ei validoida applytessä tjsp.

Master-koneella

```
helm install --name cert-manager --namespace cert-manager --version v0.9.1 --set webhook.enabled=false jetstack
/cert-manager
kubectl get all -n cert-manager
```

Seuraavaksi konffataan *ClusterIssuer* jota käytetään myöhemmin ingress-resurssien serttien tilaamiseen. Konffitiedosto *roles/k3s-server/kubernetes/cert-manager-cluster-issuer.yaml* löytyy reposta.

Master-koneella

```
kubectl apply -f roles/k3s-server/kubernetes/cert-manager-cluster-issuer.yaml
```