k3s

k3s ist eine Kubernetes Distribution.

- Loadbalancing
- Installation via Ansible

Loadbalancing

Wie funktionierts?

Standard mässig verwendet k3s die Software "klippert" als Loadbalancer, dieser erstellt sich als dämon set auf dem Cluster, wenn ein Service via Loadbalancer veröffentlicht wird.

Verwendung eines anderen Loadbalancer

Wenn ein anderer Loadbalancer wie Beispielsweise Metallb verwendet werden möchte, muss klippert entfernt, bzw. deaktiviert werden.

Klippert deaktivieren

Da k3s kubernetes in einem Binary ist, sollte Klippert beim Start des Servers (Binary) mittels Argument deaktiviert werden. Verwenden Sie dazu das extra Argument "`--disable servicelb`". `Dies können Sie im SystemD-Service File unter "ExecStart" hinzufügen.`

`/etc/systemd/k3s.service`

`Das zusatz Argument finden Sie auf der 10ten Zeile.`

```
[Unit]
Description=Lightweight Kubernetes
Documentation=https://k3s.io
After=network-online.target

[Service]
Type=notify
ExecStartPre=-/sbin/modprobe br_netfilter
ExecStartPre=-/sbin/modprobe overlay
ExecStart=/usr/local/bin/k3s server --data-dir /var/lib/rancher/k3s --disable servicelb
```

```
KillMode=process
Delegate=yes
# Having non-zero Limit*s causes performance problems due to accounting overhead
# in the kernel. We recommend using cgroups to do container-local accounting.
LimitNOFILE=1048576
LimitNPROC=infinity
LimitCORE=infinity
TasksMax=infinity
TimeoutStartSec=0
Restart=always
RestartSec=5s

[Install]
WantedBy=multi-user.target
```

Installation via Ansible

Das k3s Projekt verfügt über ein öffentliches Github-Repository, mit welchem man einfach sein Eigenes Kubernetes Cluster erstellen kann.

Voraussetzungen

- Server mit SSH-Zugriff
- Vorkonfigurierte Ansible Infrastruktur

Installation

Kurzzusammenfassung der Installation

1. Git repo klonen
2. Custom Ansible Inventory
3. Ansible Playbook ausführen
4. Kube-Config auf Client Kopieren

Git Repository klonen

Klonen Sie als erstes das Github-Repository in ihr gewünschter Ordner. Verwenden Sie dazu git (`git clone https://github.com/k3s-io/k3s-ansible.git`). Alternativ können Sie auch das Repository als Zip herunterladen und auspacken. Navigieren Sie anschliessend in das Lokal geklonte Repository(`cd k3s-ansible`).

Custom Ansible Inventory erstellen

Im k3s-ansible Repository existiert bereits ein Beispiel Inventory, dieses wird kopiert und anschliessend bearbeitet.

Kopieren des Beispiel Inventory

In diesem Beispiel wird das Beispiel-Inventory(./inventory/sample) in den Ordner my-cluster(./inventory/my-cluster) kopiert. Verwenden Sie dazu den Befehl "`cp ./inventory/sample/`

`./inventory/my-cluster -r`" Sie können auch mehrere Ordner erstellen, falls Sie mehrere Server Gruppen Administrieren.

Ansible-Inventory mit Hosts ergänzen

Bearbeiten Sie die Host-Datei("hosts.ini") im neu erstellten Ordner(./inventory/my-cluster). Bearbeiten Sie die Textdatei mit ihrem gewünschten Text editor. Bsp.: `vim ./inventory/my-cluster/hosts.ini`

Hier ist mein Beispiels Inventory

```
[master]
srvk3s01.voser.local ansible_port=122

[node]
srvk3s02.voser.local ansible_port=222
srvk3s03.voser.local ansible_port=322
srvk3s04.voser.local ansible_port=422
srvk3s21.voser.local ansible_port=2122
srvk3s22.voser.local ansible_port=2222
srvk3s24.voser.local ansible_port=2322

[k3s_cluster:children]
master
node

[k3s_cluster:vars]
ansible_host=home.voser.cloud
ansible_user=ansible-user
```

Ansible-Gruppen Variablen bearbeiten

Der Inventory Ordner beinhaltet einen Unterordner namens: "group_vars", in diesem befindet sich eine Datei namens: "all.yml". Diese Datei beinhaltet allgemeine Informationen zum kubernetes cluster, bearbeiten Sie diese nach Ihrer Infrastruktur. Das einzige was gesetzt werden muss ist die Master IP, sowie der Ansible-User, je nach Infrastruktur möchten Sie jedoch auch die anderen Informationen überarbeiten.

Bsp.: `vim ./inventory/my-cluster/group_vars/all.yml`

```
---
k3s_version: v1.21.7+k3s1
```

```
ansible_user: ansible-user
systemd_dir: /etc/systemd/system
master_ip: "172.22.0.101"
#master_ip: "{{ hostvars[groups['master'][0]]['ansible_host'] | default(groups['master'][0]) }}"
extra_server_args: "--disable servicelb"
extra_agent_args: ""
```

Ansible Playbook ausführen

Es existieren zwei Ansible-Playbooks in diesem Repository (site.yml & reset.yml)

Mit dem site.yml wird ein Cluster Installiert, sowie aktualisiert, und mit dem reset.yml wird alles von den Hosts entfernt (Deinstallation).

Führen Sie die Playbooks mit dem normalen Befehl "ansible-playbook", sowie dem extra Parameter "-i <Inventoryfile>" aus. Der Parameter "-i" steht für Inventoryfile.

Bsp. Installation: `ansible-playbook -i ./inventory/my-cluster/hosts.ini site.yml`

Bsp. Deinstallation: `ansible-playbook -i ./inventory/my-cluster/hosts.ini reset.yml`

Kube-Config auf Cleint Kopieren

Wenn das Cluster erfolgreich Installiert ist finden Sie die Kube-Config Datei auf dem Master server unter (/etc/rancher/k3s.yml) kopieren Sie den Inhalt auf Ihren gewünschten Host und bearbeiten Sie die Variable "server:" in der Datei. Anschliessend können Sie ihr Kubernetes Cluster via Kubectl auf Ihrem Computer ansteuern. Wenn Sie das Cluster öffentlich nicht erreichbar machen wollen, können Sie kubectl auch auf dem Master verwenden.