

NWT Labor Protokoll - Nickel

Name	Kirschner und Kudler
Klasse	5CN
Datum der Übung	01.10.2019 - Fortlaufend
Datum der Abgabe	12.01.2020
Übungsnummer	01
Auftraggeber	NIC
Thema der Übung	Kubernetes

Inhaltsverzeichnis

1	Aufgabenstellung	3
1.1	Kubernetes	3
1.1.1	Aufbau und Architektur	3
2	Installation	4
2.1	Validierung bevor es losgeht	4
3	Installation	5
3.1	Docker	5
3.2	Kubernetes	6
4	Erstellung eines Clusters	7
4.1	Einstellungen	7
4.2	Kubeadm	7
5	Troubleshooting	8

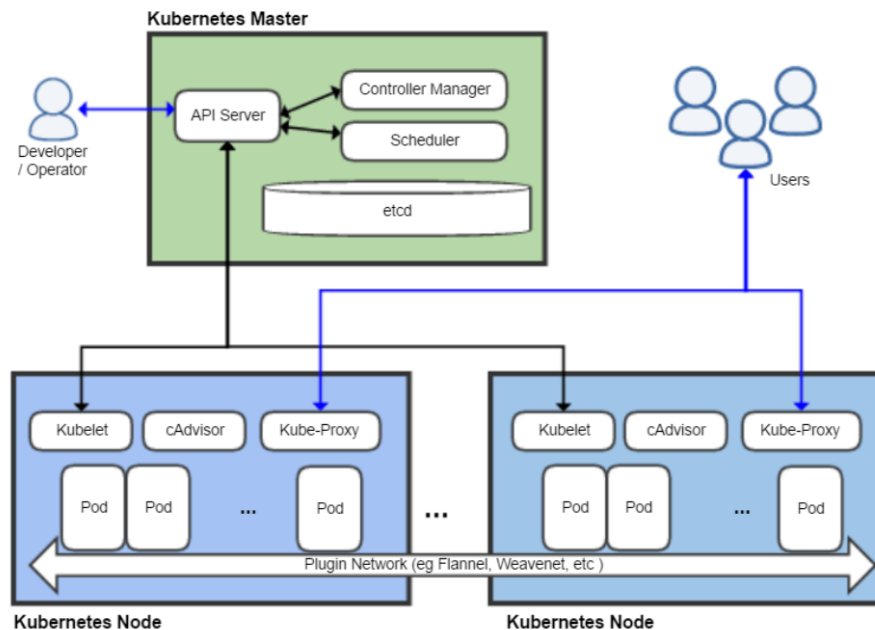


Abbildung 1: Kubernetes Architektur

1 Aufgabenstellung

Die Idee des Projektes ist es einen Kubernetes Cluster auf 3 VMS zu deployen und die Möglichkeiten die dadurch entstehen zu erkunden.

Ein optionales Ziel ist es, diese Infrastruktur auf eine Web-Applikation mit Anbindung einer Datenbank anzuwenden und Aktivitäten zu überprüfen.

1.1 Kubernetes

Kubernetes ist ein Open-Source Orchestrierungs Tool, das dazu dient Container automatisiert Bereitzustellen und zu verwalten.

Der Name Kubernets kommt aus dem griechischen und steht für Steuermann.

Wir verwenden in unserem Projekt Docker als Container Runtime.

1.1.1 Aufbau und Architektur

Der Vorteil bei Kubernetes liegt darin, dass die es *Pods* orchestriert. Sie stellen die kleinstmögliche Steuerbare Einheit im Kubernetes Universum dar. Sie laufen auf *Nodes* - also VMs oder physischen Maschinen). Ein Pod kann einen oder mehrere Container beinhalten.

Die Architektur ist auf dem Master-Slave System aufgebaut.

Der Master ist die *Control Plane*, auf ihr wird Inventur über alle Objekte in einem Cluster geführt. Der Master steuert außerdem alle Slaves (Minions). Wir haben in unserem Fall nur einen Master-Node. Es können aber auch - zwecks Redundanz - mehr Master-Nodes in einem Kubernetes Cluster konfiguriert werden.

Auf jedem Minion muss zusätzlich auch die zu verwendende Container-Runtime installiert werden.

2 Installation

2.1 Validierung bevor es losgeht

Bei jedem Node (VM oder physische Maschine), die im Cluster verwendet werden soll, müssen sich folgende Eigenschaften unterscheiden:

Eigenschaft	Befehl zum Prüfen
MAC Adresse	<code>ip link</code>
Produkt UUID	<code>cat /sys/class/dmi/id product_uuid</code>

Da wir für dieses Beispiel Debian 10 verwenden, müssen wir zusätzlich noch iptables in den legacy-mode schalten.

```
1 update-alternatives --set iptables /usr/sbin/iptables-legacy
2 update-alternatives --set ip6tables /usr/sbin/ip6tables-legacy
3 update-alternatives --set arptables /usr/sbin/arptables-legacy
4 update-alternatives --set ebtables /usr/sbin/ebtables-legacy
```

3 Installation

3.1 Docker

Als Container Runtime benutzen wir für dieses Beispiel Docker.

Zuerst müssen wir alle Dependencies von Docker installieren.

```
1 apt update
2 apt -y install apt-transport-https ca-certificates curl gnupg2 software-properties-
  common
```

Anschließend fügen für den offiziellen GPG-Key von Docker zu unserem Package-Manager hinzu. Mit diesem Schlüssel sind die Pakete signiert.

```
1 curl -fsSL https://download.docker.com/linux/debian/gpg | apt-key add -
```

Nun können wir die offiziellen Docker-Repositories hinzufügen.

```
1 add-apt-repository \
2     "deb [arch=amd64] https://download.docker.com/linux/debian \
3     $(lsb_release -cs) \
4     stable"
```

Die Installation selbst ist nun ziemlich einfach.

```
1 apt-get update && apt-get install -y \
2     containerd.io=1.2.10-3 \
3     docker-ce=5:19.03.4~3-0~debian-$(lsb_release -cs) \
4     docker-ce-cli=5:19.03.4~3-0~debian-$(lsb_release -cs)
```

Zu guter Letzt passen wir noch die Einstellungen für Docker an.

```
1 cat > /etc/docker/daemon.json <<EOF
2 {
3     "exec-opts": ["native.cgroupdriver=systemd"],
4     "log-driver": "json-file",
5     "log-opts": {
6         "max-size": "100m"
7     },
8     "storage-driver": "overlay2"
9 }
10 EOF
11
12 mkdir -p /etc/systemd/system/docker.service.d
```

Einen kleinen Restart brauchen wir noch.

```
1 systemctl daemon-reload
2 systemctl restart docker
```

Wie jeder gute IT-Admin werden verifizieren wir natürlich noch am Ende die gelungene Installation.

```
1 docker version
```

Falls hier vernünftige Informationen angezeigt werden und keine Fehlermeldung ist die Installation gelungen.

3.2 Kubernetes

Als nächsten Schritt installieren wir Kubernetes. Oder besser gesagt die 3 Services, aus denen unsere Kubernetes Installation bestehen wird.

- kubeadm
- kubelet
- kubect1

Dafür benutzen wir einen ähnlichen Ablauf, wie bei Docker. Der Einfachheit halber ist nun nicht jeder Schritt kommentiert, sondern ein fertiges Skript zu sehen.

```
1 apt-get update && sudo apt-get install -y apt-transport-https curl
2 curl -s https://packages.cloud.google.com/apt/doc/apt-key.gpg | sudo apt-key add -
3 cat <<EOF | sudo tee /etc/apt/sources.list.d/kubernetes.list
4 deb https://apt.kubernetes.io/ kubernetes-xenial main
5 EOF
6 apt-get update
7 apt-get install -y kubelet kubeadm kubect1
8 apt-mark hold kubelet kubeadm kubect1
```

Auch hier folgt natürlich wieder die Verifikation.

```
1 kubeadm version
2 kubelet --version
3 kubect1 version
```

4 Erstellung eines Clusters

4.1 Einstellungen

Ein kleiner Schritt hält uns noch vom Cluster ab: Wir müssen swap abschalten.

```
1 swapoff -a
2 cp /etc/fstab /etc/fstab.orig
3 cat /etc/fstab.orig | grep -v 'swap' > /etc/fstab
```

Jetzt noch ein Reboot.

```
1 reboot 0
```

4.2 Kubeadm

Bevor man den Cluster installiert muss man sich für ein pod network add-on entscheiden. Die Auswahl ist da groß, weswegen wir uns schlichtweg für das beliebteste Add-On entschieden haben: Flannel

Auf unserem Master führen wir nun folgenden Befehl aus:

```
1 kubeadm init --pod-network-cidr=10.244.0.0/16
```

Mit den anderen Nodes können wir jetzt joinen. Den Befehl dafür finden wir im Output von kubeadm init am Master.

```
1 kubeadm join 10.0.0.88:6443 --token <token> \
2 --discovery-token-ca-cert-hash sha256:<hash>
```

5 Troubleshooting

Bei der Verifizierung am Master mit `kubectl get nodes` kann es zu folgendem Fehler kommen:

```
1 The connection to the server localhost:8080 was refused - did you specify the right  
   host or port?
```

Das lässt sich mit den folgenden Befehlen beheben. (Credits an den GitHub post von user csarora)

```
1 cp /etc/kubernetes/admin.conf $HOME/  
2 chown $(id -u):$(id -g) $HOME/admin.conf  
3 export KUBECONFIG=$HOME/admin.conf
```