

Installation Cluster

Vorwort

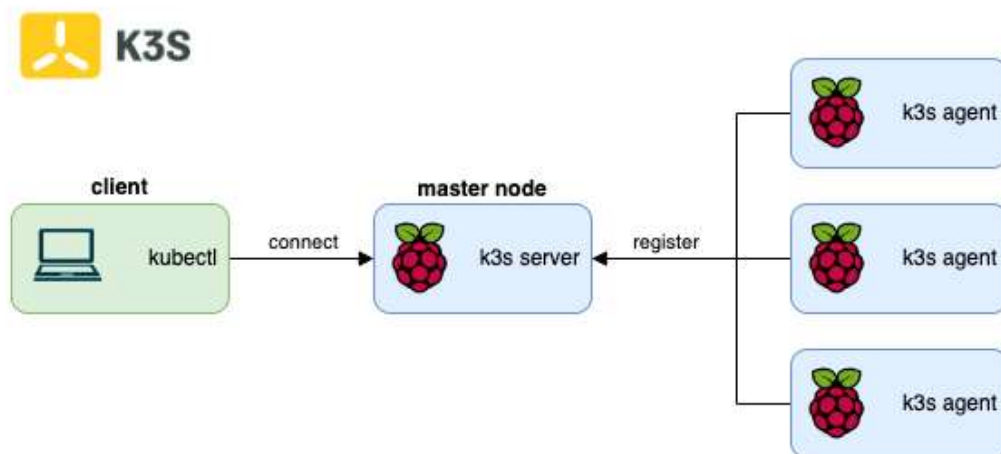
Auf die RasPis wurde das Raspberry Pi OS aufgespielt. Die aktuelle Kombination der Software – k3s, Kubectl, Helm – wurde ausgewählt, da die Installationsanleitung im Internet einfach zu befolgen und verständlich war.

Installation k3s

Die Installation von k3s entstammt größtenteils dem Folgenden Artikel:

<https://medium.com/thinkport/how-to-build-a-raspberry-pi-kubernetes-cluster-with-k3s-76224788576c>

Auf die RasPis wird das OS aufgespielt. Über den Raspberry Pi Imager können ein Name und eine WiFi Verbindung vor dem flashen des Images konfiguriert werden. Die folgende Architektur wurde aufgebaut, allerdings mit nur einem Agenten



Auf dem Master-Node wird der k3s-Server installiert:

```
curl -sL https://get.k3s.io | sh -
```

Die Installation wird nicht fehlerfrei Verlaufen. Im Anschluss muss im boot-Verzeichnis eine Änderung vorgenommen werden.

```
sudo nano /boot/cmdline.txt
```

In dem geöffneten Dokument muss am Ende der ersten Zeile folgendes eingefügt werden:

```
cgroup_memory=1 cgroup_enable=memory
```

Wichtig ist, dass die erste Zeile nicht verlassen und keinen Zeilenumbruch eingefügt wird. Mit strg X und y + Enter wird die Änderung gespeichert. Im Anschluss muss der RaspPi neugestartet werden.

```
Sudo reboot
```

Um zu überprüfen, ob der Master-Node korrekt aufgesetzt wurde, kann der folgende Befehl ausgeführt werden:

```
Sudo k3s kubectl get nodes
```

NAME	STATUS	ROLES	AGE	VERSION
raspinodel	Ready	<none>	27h	v1.25.3+k3s1
raspimaster	Ready	control-plane,master	27h	v1.25.3+k3s1

Um einen Agenten oder Worker-Node wird die IP-Adresse des k3s Server und der Token benötigt. Befehl für die IP-Adresse:

```
hostname -I | awk '{print $1}'
```

Befehl für das Servertoken:

```
sudo cat /var/lib/rancher/k3s/server/node-token
```

Die resultierende Zeichenkette enthält die Zeichenkette „::string:“. Die gesamte Zeichenkette muss zwischengespeichert werden. Auf dem Agenten wird jetzt k3s installiert. IP-Adresse und Token werden als Parameter mitgegeben und müssen im folgenden String ersetzt werden:

```
curl -sL https://get.k3s.io | K3S_URL=https://<kmaster_IP_from_above>:6443  
K3S_TOKEN=<token_from_above> sh -
```

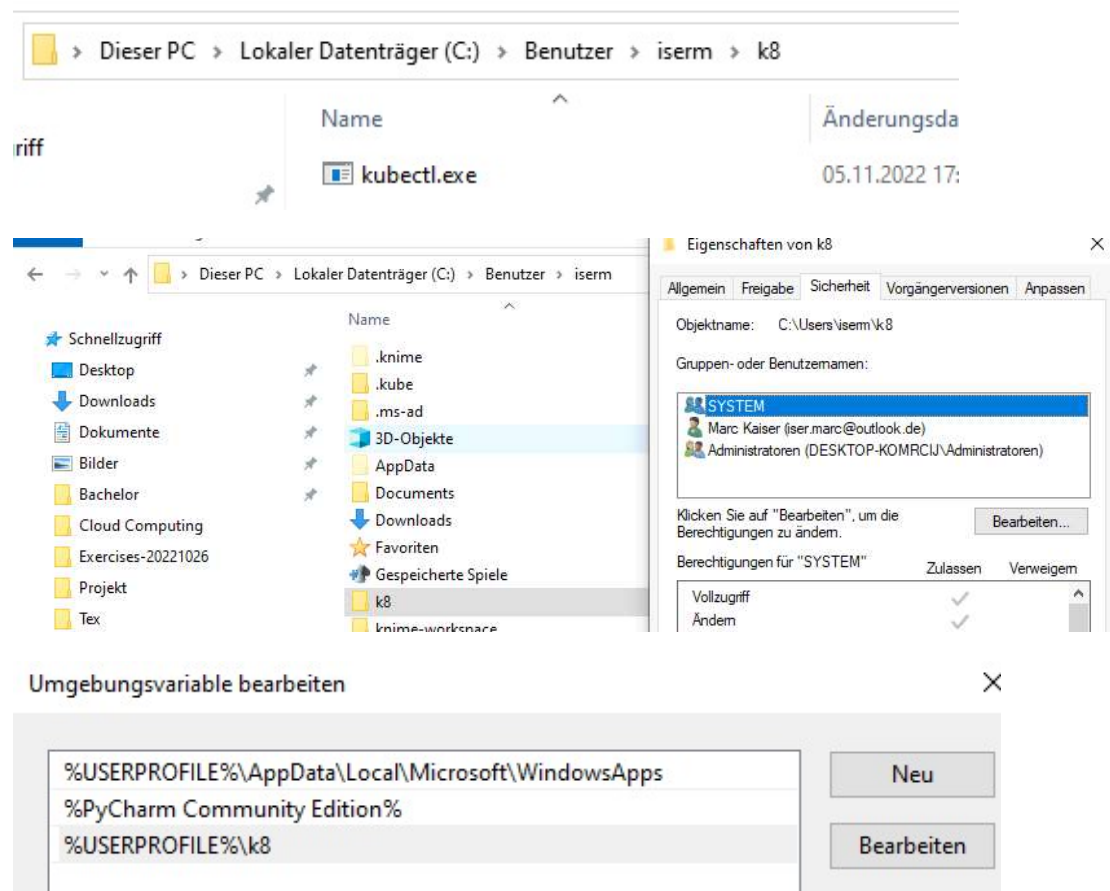
Die Korrektur im Boot-Verzeichnis muss auch bei jedem der Agenten vorgenommen werden. Nach der Korrektur müssen die RasPis neugestartet werden. Durch den folgenden Befehl kann die erfolgreiche Installation verifiziert werden.

```
Sudo k3s kubectl get nodes
```

Installation Kubectl

Kubectl wird auf dem Gerät installiert, mit dem man dem Cluster Befehle geben möchte. Im Folgenden wird leicht von dem Artikel abgewichen. Die nächste Installation wurde auf Windows durchgeführt. Hier sind alle Anleitungen zur Installation von kubectl <https://kubernetes.io/docs/tasks/tools/>. Für Windows wird die aktuelle Version heruntergeladen. Es handelt sich bei der Anwendung um eine Command Line Anwendung, daher muss die Anwendung in die Path Umgebungsvariable hinzugefügt werden. Im

Folgendes ist ein Beispiel aufgeführt.



Weiterhin muss eine Konfigurationsdatei aus dem Master-Node kopiert werden:

```
sudo cat /etc/rancher/k3s/k3s.yaml
```

Greift man über SSH auf den RasPi zu, kann man mit strg + einfg kopiert werden. Die resultierende Datei muss zwischengespeichert und bearbeitet werden. Allerdings darf auf dem Master Node keine Änderung vorgenommen werden. Die IP-Adresse des Master Nodes muss durch die tatsächliche Adresse ersetzt werden. Im Homeverzeichnis des Benutzers muss das Verzeichnis ./kube eingefügt werden. Die bearbeitete Datei muss dort unter dem Namen „config“ (ohne Endung) eingefügt werden.

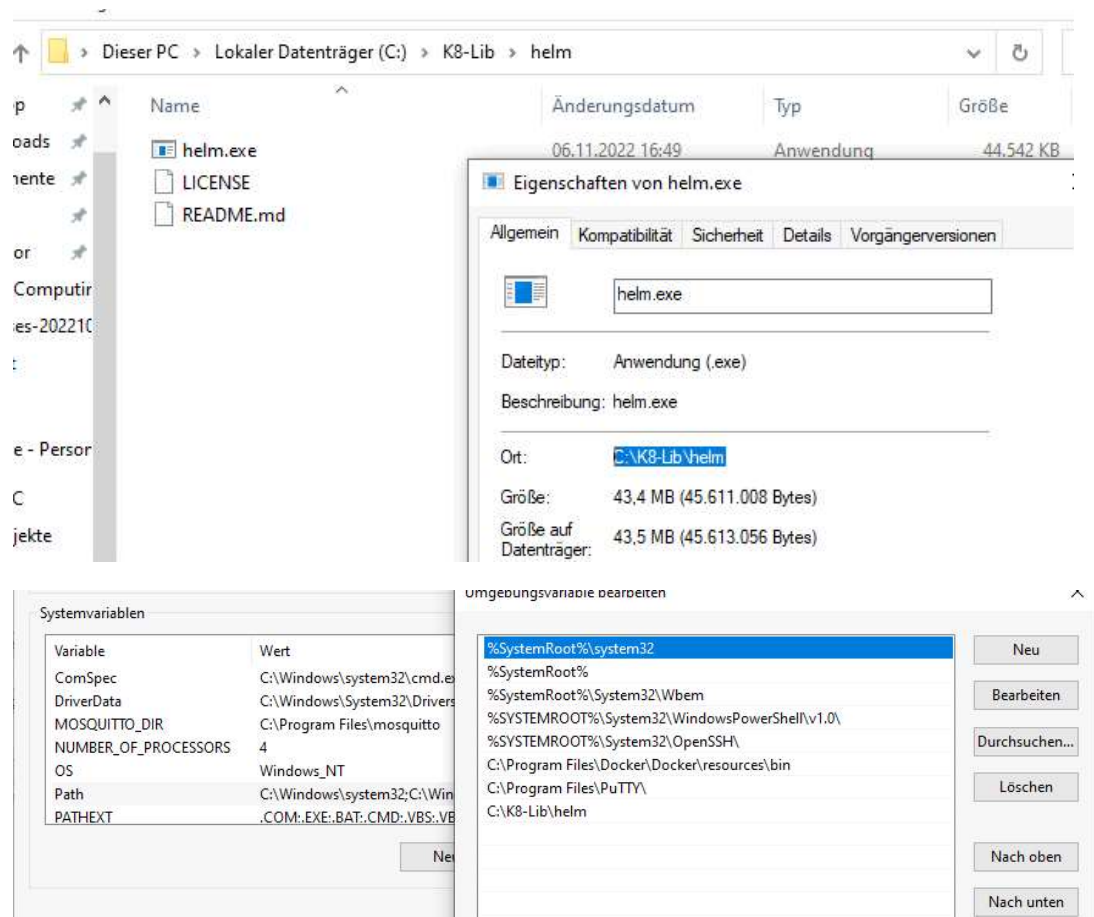
Mit dem folgenden Befehl kann die Installation überprüft werden

```
kubectl config use-context
kubectl get nodes
```

```
C:\Users\iserm>kubectl get nodes
NAME                STATUS    ROLES                  AGE    VERSION
raspimaster         Ready    control-plane,master   27h    v1.25.3+k3s1
raspinode1          Ready    <none>                 27h    v1.25.3+k3s1
```

Installation Helm

Helm bietet eine Möglichkeit das automatische Erstellen, Paketieren, Konfigurieren und Einsetzen von Anwendungen in Kubernetes. Hier muss zunächst die gewünschte Version heruntergeladen werden: <https://helm.sh/docs/intro/install/>. Helm wird auf dem gleichen System wie Kubectl installiert. Hier muss ebenfalls eine Umgebungsvariable angelegt werden, dieses Mal aber als System-Umgebungsvariable.



Mit dem Folgendem Befehl kann überprüft werden, ob Helm installiert ist.

```
helm version
```

Um Mosquitto zu installieren, müssen die folgenden Befehle ausgeführt werden.

```
helm repo add k8s-at-home https://k8s-at-home.com/charts/
```

```
helm repo update
```

create a dedicated namespace

```
kubectl create namespace mqtt
```

install mosquitto

```
helm install mosquitto k8s-at-home/mosquitto -n mqtt
```

check installation

```
helm list -n mqtt
```

check the resources installed

```
kubectl get all -n mqtt
```