

Automatisiertes Aufsetzen eines Kubernetes-Clusters auf Raspberry Pis mithilfe von Ansible-Playbooks

KL

Seminararbeit (Bachelor)

Betreuer: RH, HB

Agenda

- Einleitung
- Docker, Kubernetes, Raspberry Pi, Ansible
- Umsetzung
- Live-Demo + Austausch
- Fazit

Einleitung

- Kubernetes ermöglicht skalierbare, verteilte Applikationen
- IoT-Projekte machen Fog Computing nötig (Datenmengen, Latenzen, Verfügbarkeit)
- Aufwändige Einrichtung mit Ansible automatisieren

Docker

- Container-virtualisierung
- Isolation von Ressourcen
- Dockerfiles

```
FROM debian:buster-slim
```

```
RUN set -x \  
    && addgroup nginx \  
    && adduser nginx \  
    && apt-get update [...]
```

```
EXPOSE 80
```

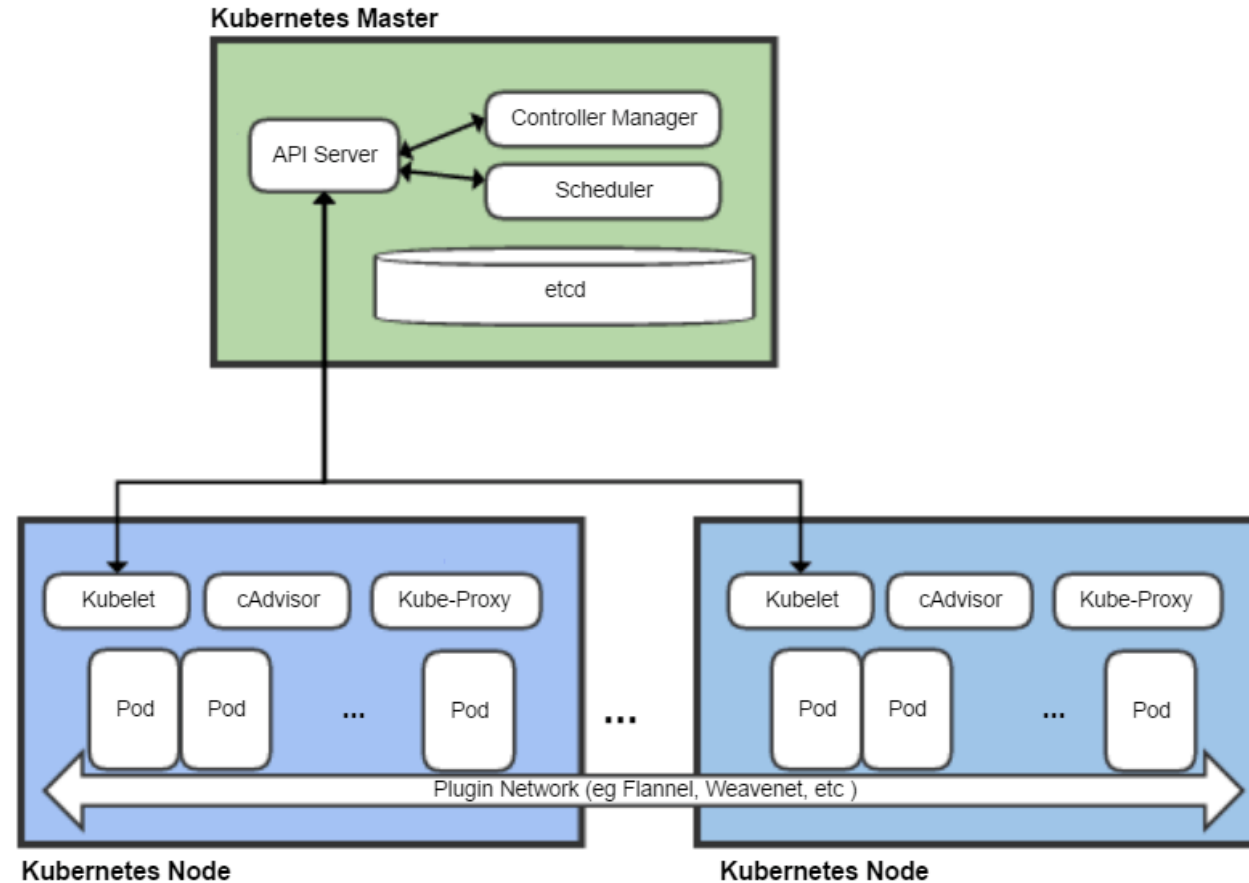
```
STOPSIGNAL SIGTERM
```

```
CMD ["nginx", "-g", "daemon off;"]
```

<https://github.com/nginxinc/docker-nginx/blob/master/stable/buster/Dockerfile>

Kubernetes

- Skalierung und Verwaltung von Containern
- Verteilte Hosts/Cloud
- Lastverteilung



Von Khtan66 - Eigenes Werk, CC BY-SA 4.0, <https://commons.wikimedia.org/w/index.php?curid=53571935>

Kubernetes

- Skalierung und Verwaltung von Containern
- Verteilte Hosts/Cloud
- Lastverteilung

```
apiVersion: apps/v1
metadata:
  name: nginx-deployment
spec:
  selector:
    matchLabels:
      app: nginx
  replicas: 2
  spec:
    containers:
      - name: nginx
        image: nginx:1.14.2
        ports:
          - containerPort: 80
```

<https://kubernetes.io/docs/tasks/run-application/run-stateless-application-deployment/>

Raspberry Pi

- Einplatinencomputer
- Günstig und sparsam
- Für IoT geeignet



Ansible

- Automatisierte Provisionierung von entfernten Geräten
- Infrastructure as code
- YAML-Playbooks
- Module

```
---  
- hosts: all  
  become: yes  
  tasks:  
    - name: Set master flag  
      set_fact: master=yes  
      when: ansible_host == masterIp  
  
    - name: Set timezone  
      timezone:  
        name: "{{ timezone }}"  
  
    - name: Upgrade apt packages  
      apt:  
        upgrade: yes  
        update_cache: yes
```

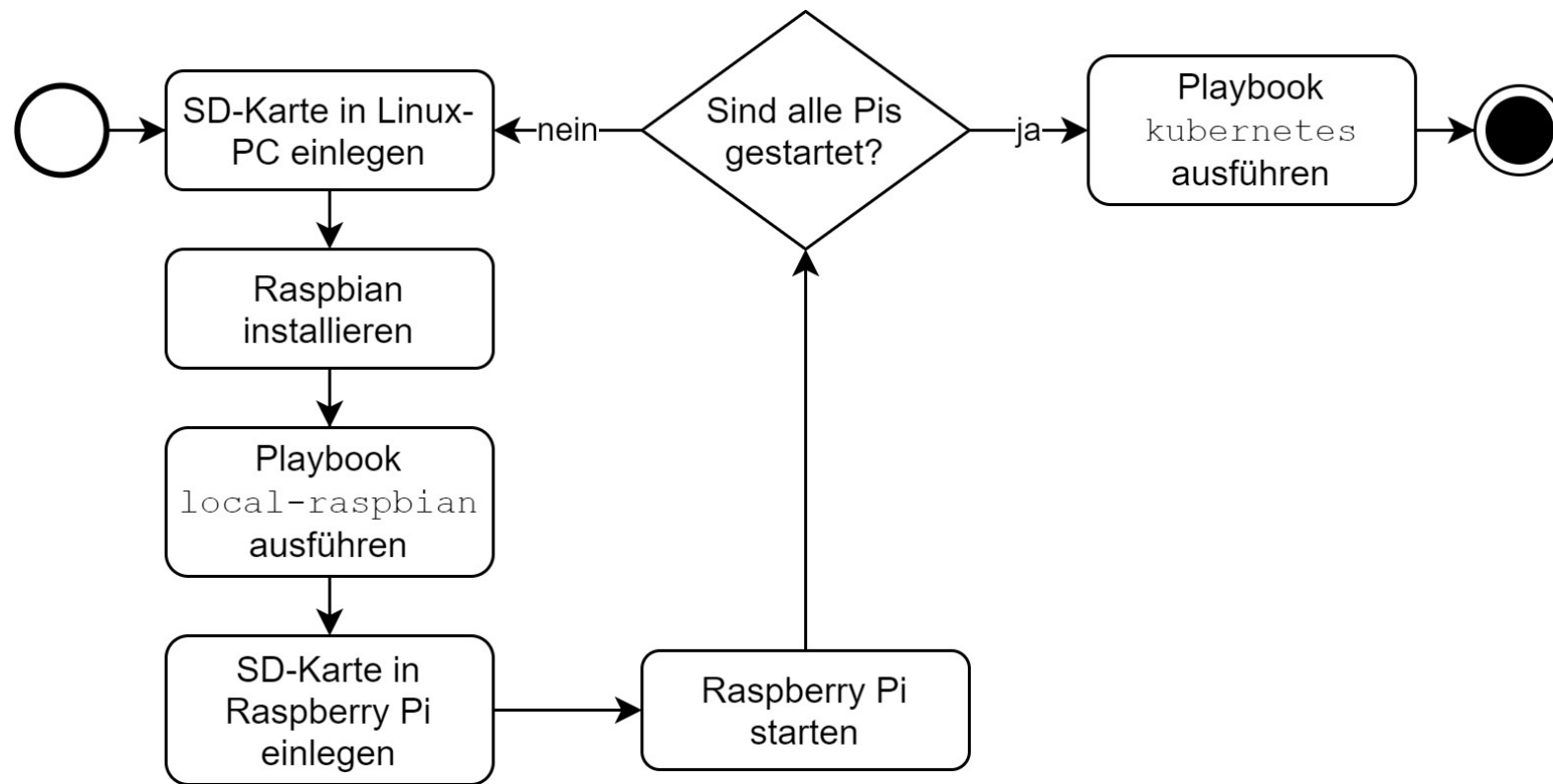

Aufgaben

- Raspbian installieren
- WiFi einrichten (Zugangsdaten, statische IP)
- Swapfile deaktivieren
- Control Groups aktivieren
- SSH-Keys hinterlegen
- Softwarepakete aktualisieren
- Docker installieren
- Kubernetes installieren
- Master Node initialisieren
- Weitere Nodes zum Cluster hinzufügen
- Virtuelles Netzwerk installieren

`local-raspbian.yaml`

`kubernetes.yaml`

Ablauf der Einrichtung



„Live“-Demo

Fazit

- Kubernetes-Cluster in 35 Minuten
- Raspbian-Update nicht reibungslos

Fragen?

Danke!