etcd - Das Gehirn des Kubernetes Clusters

Oder was Flöße und die Ägäis damit zu tun haben

Gabriel Wustmann

4. Juli 2026

Agenda

1. Motivation
2. Kubernetes-Architektur
3. Aufbau etcd
4. Exkurs Consensus
5. Funktionsumfang etcd
6. Integration etcd in Kubernetes

Motivation

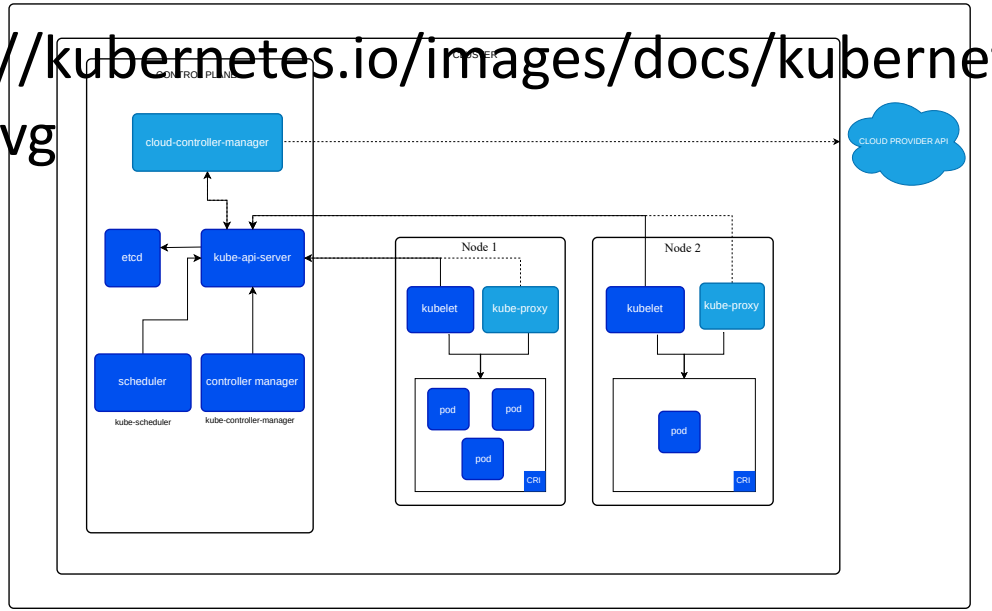
- Tieferer Blick in den inneren Aufbau von Kubernetes
- Limits bestimmen für Größenskalierung
- » ■ 

Who I am

- Gabriel Wustmann
- Studium der (Wirtschafts-)Informatik
- seit 2003 mit IT befasst und beschäftigt
- seit 2015 Container
- aktuell IT-Architekt

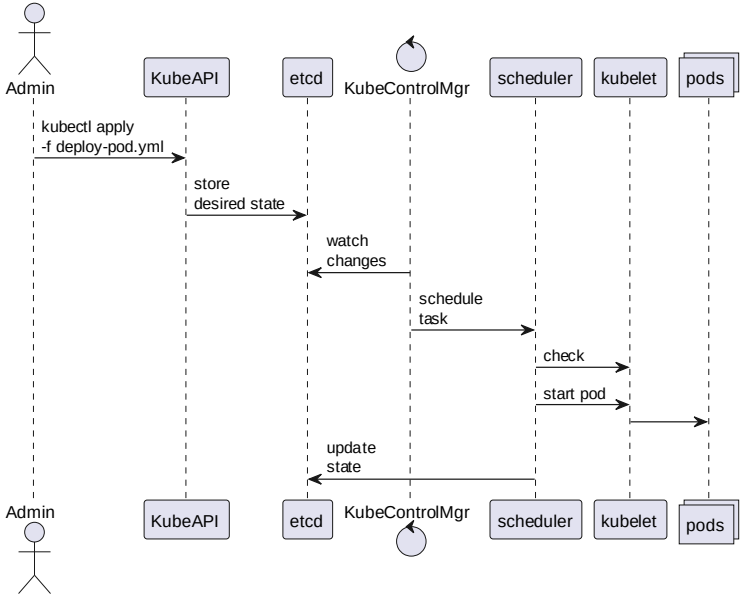
Architektur

Quelle: https://kubernetes.io/images/docs/kubernetes-cluster-architecture.svg



KubeArc

Ablauf POD-Deployment



Create-POD

Was ist etcd



- Key/Value-Store
- Verteilte Datenreplikation
- Bietet Ausfallsicherheit
- Bestandteil nicht nur von Kubernetes

Projekt etcd

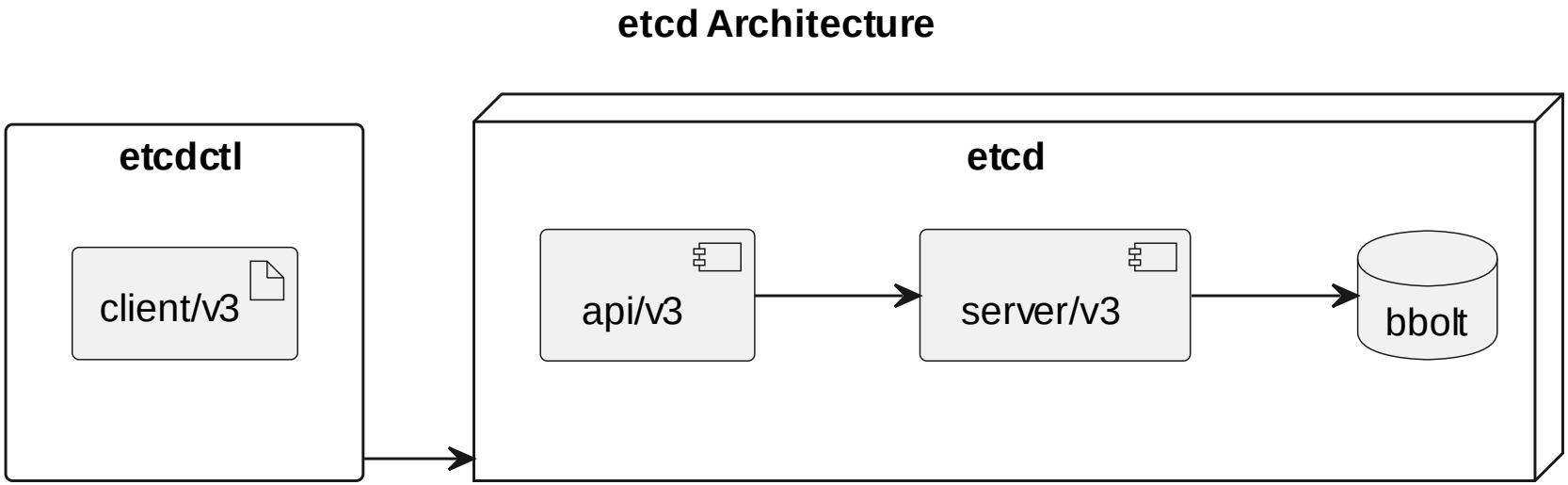
- OpenSource: https://github.com/etcd-io/etcd
- Lizenz: Apache 2.0
- ~ 950 Beitragende
- 285 Release
- Ursprünge im CoreOS Projekt ~ 2013
- 2014 Teil von Kubernetes
- Inzwischen unter dem Dach der CNCF

Komponenten etcd

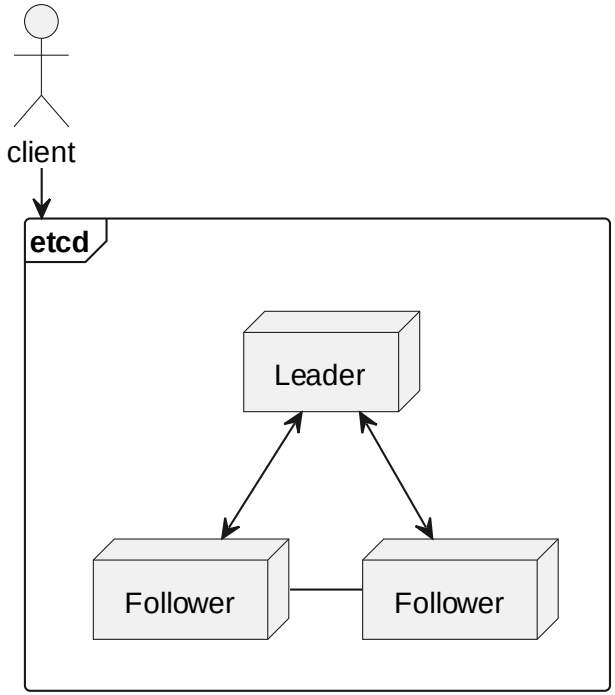
- Golang
- etcd API
- BBolt
- RAFT



Schematischer Aufbau



Typischer Betrieb



Verteiltes System

> A distributed system is a collection of independent computers that appears to its users as a single coherent system.

Nach Andrew S. Tanenbaum, Maarten Van Steen

> A distributed system is a collection of autonomous computers that collaborate with each other on a message-passing network to achieve a common goal.

Nach Imran Bashir

Exkurs Consensus

Consensus-Algorithmen sind Grundlage von Verteilten Systemen

Consensus: Historie

- 1980er Goldenes Jahrzehnt der Forschung
- 1982 “Byzantine Generals” L. Lamport 1982
- 1987 3 Phase Commit

FLP Paper

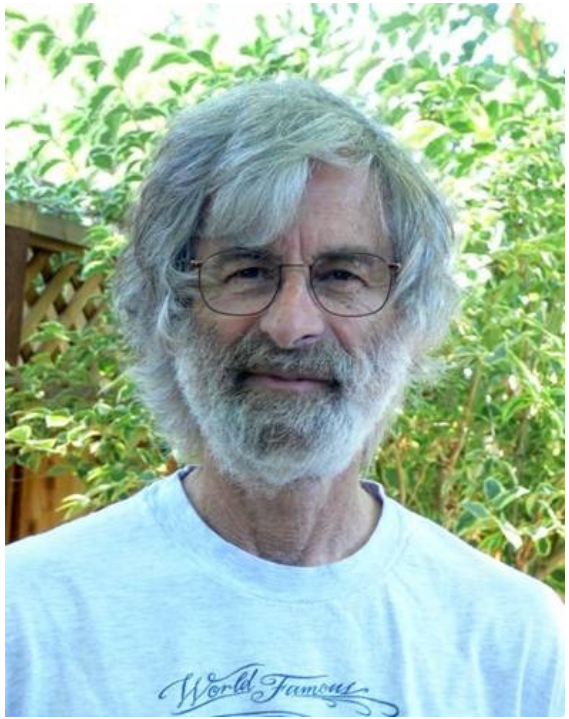
“Impossibility of Distributed Consensus with One Faulty Process”

1985 veröffentlicht von

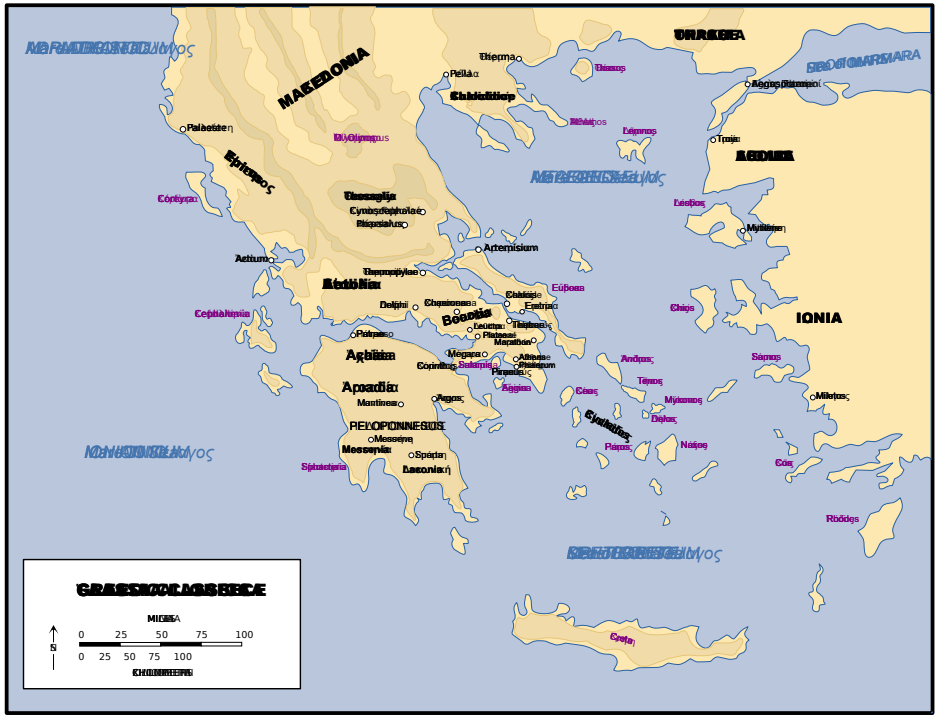
- Michael Fischer
- Nancy Lynch
- Michael Paterson

Paxos

- 1990 erstellt von Leslie Lamport
- 1998 veröffentlicht als “The Part-Time Parliament”
- 2001 überarbeitet, veröffentlicht als “Paxos Made Simple”



Paxos Insel


MACEDONIA
THRACE
AEOLIS
IONIA
Pella
Thessaloniki
Mount Olympus
Epirus
Thessaly
Larissa
Lesbos
Mytilene
Chios
Samos
Euboea
Chalcis
Boeotia
Thebes
Delphi
Thermopylae
Artemisium
Athens
Piraeus
Attica
Peloponnesus
Corinth
Argos
Sparta
Laconia
Messenia
Arcadia
Andros
Tinos
Mykonos
Delos
Naxos
Paros
Ios
Milos
Rhodes
Crete
Troy
Aegae
Pergamon
Ithaca
Cephallonia
Actium
Patras
Olympus
Pieria
Chalkidiki
Lemnos
Imbros
Thasos
Samothrace
Aegean Sea
Ionian Sea
Mediterranean Sea
Black Sea
GREECE
MICRA
0 25 50 75 100
N
0 25 50 75 100
CHOMMEEN

Paxos Details

- Leader election
- Fast/Slow Read
- Phasen: Prepare, Promise, Accept
- Besonders: Rolle des Lernenden

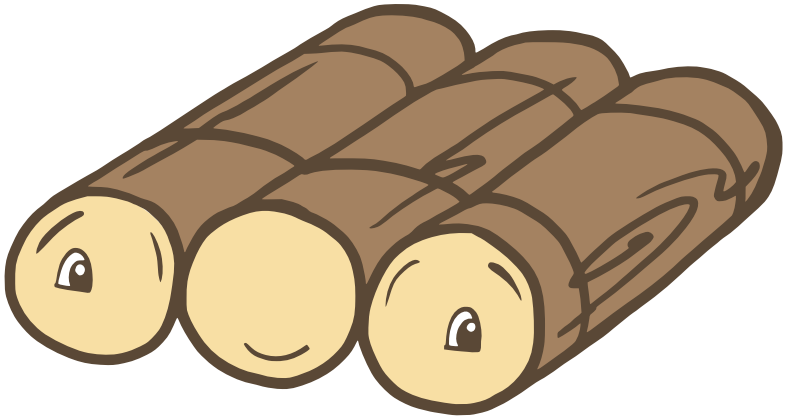
RAFT

Paxos zu komplex

Schwer verständlich

einfachere Implementierung

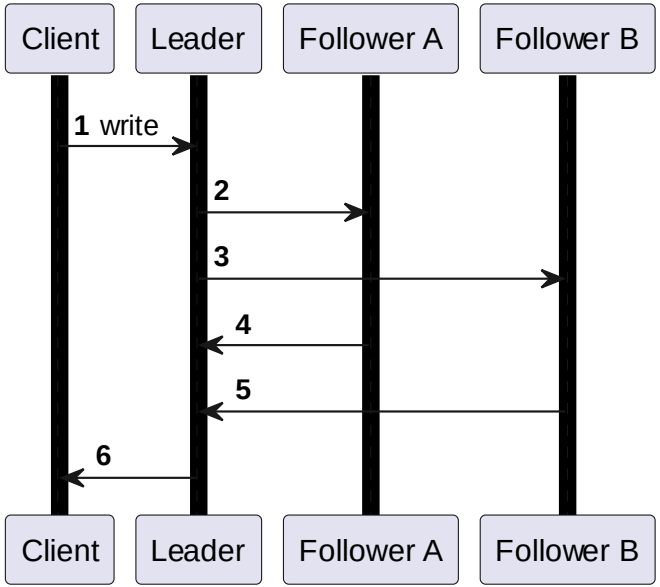
2014 Gegenvorschlag in “In Search of an Understandable Consensus Algorithm”



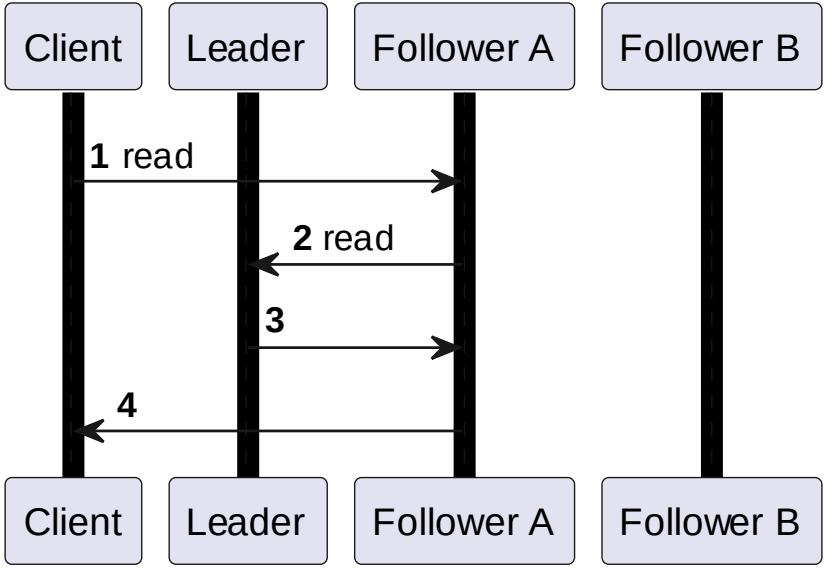
RAFT Details

- min. 3 Nodes, bzw. ungerade
- 2 echte Rollen:
  - Leader
  - Follower
- 2 Phasen:
  - Leader-Election
- Read/Write

Write / Read Sequence



write

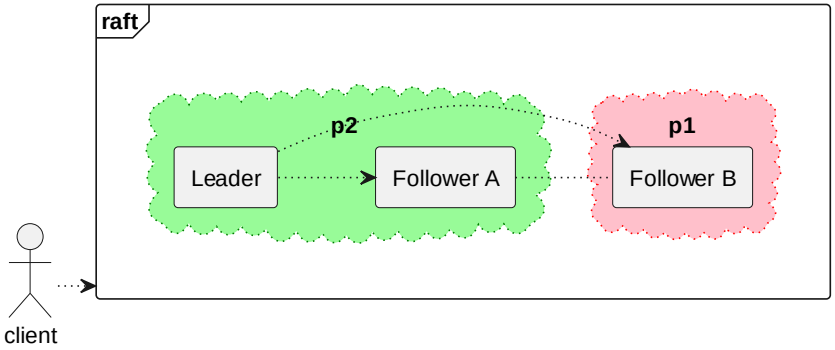


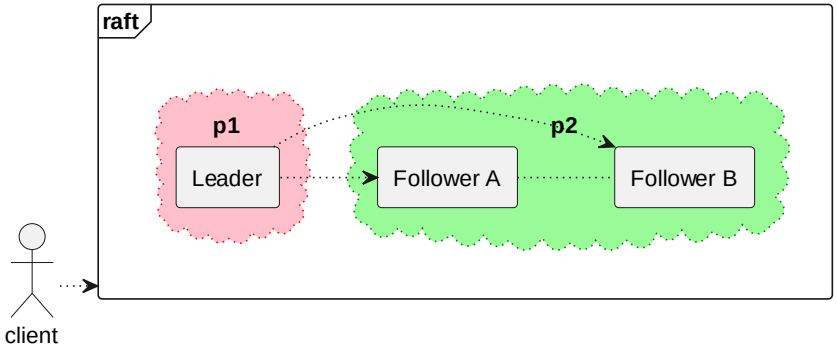
read

Leader-Election

- Follower -> Candidate
- Electionperiodecounter + 1
- Message an andere Follower

Split-Szenarien





Weitere Algorithmen

- Multi-Paxos
- KRaft
- PBFT
- Blockchain
  - Bitcoin
  - Etherum

Exkurs Ende

zurück zu etcd

Demo etcd



Funktionsumfang

- Befehlsumfang:
  - Key/Value CRUD
  - Watching
  - Versioning/History
  - Keep-Alive
  - Cluster-Messages
  - Auth
  - Maintenance
- Snapshots

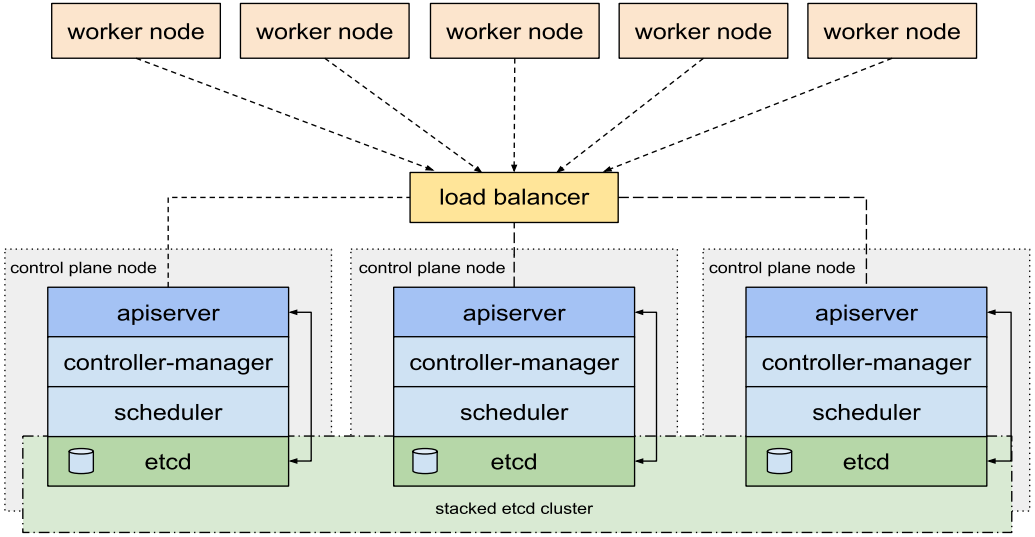
Integration etcd Kubernetes

- Verteilte Installation (min 3, empf. 5 Instanzen)
- Loadbalancer als Zugang
- Absicherung der Kommunikation via
  - TLS (Client)Zertifikate
  - Beschränkung auf den Kube-API Server
- Backup-Plan einrichten
- Defragmentation 
- als Standard auf jedem control plane node
- externe etcd Instanz möglich

Variante 1

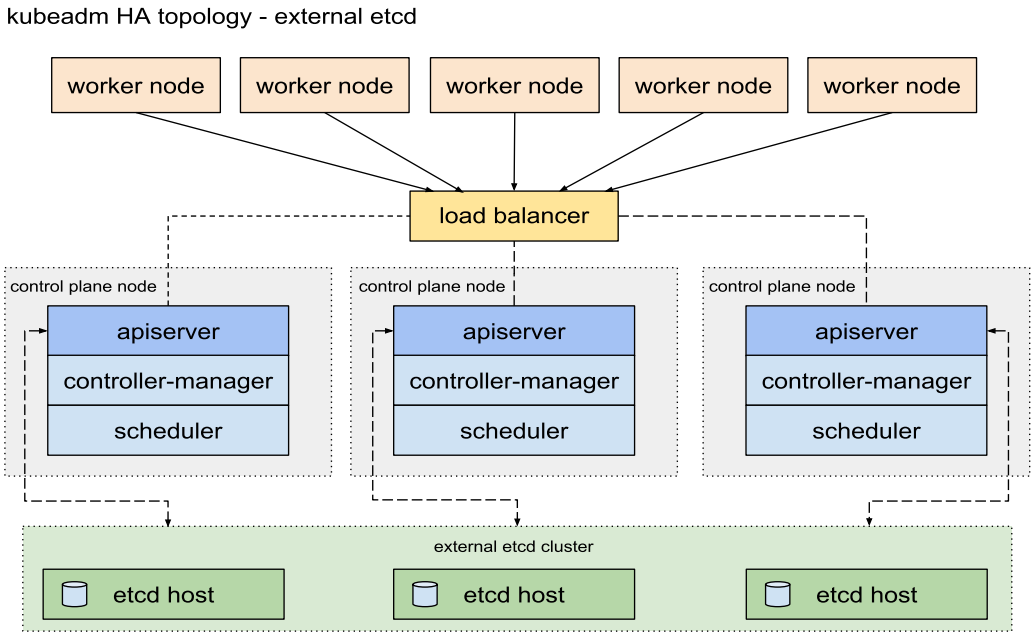
Teil der Controlplane

kubeadm HA topology - stacked etcd



Variante 2

Specialized Nodes



Limits

- Max. Request Size 1.5 MiB (max-request-bytes Flag)
- Max Storage: 2 GiB (quota-backend-bytes, 8 GiB gilt als Obergrenze)

Hilfsprogramme

- Backup-Plan
- Defragmentation (zB. mit etcd-defrag)

```
etcdctl snapshot save
```

The End

Vielen Dank!

Fragen 💬

Präsentation erstellt mit



+ + + Pandoc

Literatur

Bildnachweis:

- https://commons.wikimedia.org/wiki/File:Go_gopher_favicon.svg
- https://commons.wikimedia.org/wiki/File:Logo_of_etcd.svg
- https://commons.wikimedia.org/wiki/File:Leslie_Lamport.jpg
- https://commons.wikimedia.org/wiki/File:Classical_Greece_Map_-_USMA.svg
- https://commons.wikimedia.org/wiki/File:Raft_Consensus_Algorithm_Mascot_on_transparent_background.svg
- https://commons.wikimedia.org/wiki/File:Deepin_Icon_Theme_%E2%80%93_text-x-markdown_(14).svg
- https://commons.wikimedia.org/wiki/File:Plantuml_Logo.svg
- https://kubernetes.io/images/docs/kubernetes-cluster-architecture.svg
- https://kubernetes.io/images/kubeadm/kubeadm-ha-topology-stacked-etcd.svg
- https://kubernetes.io/images/kubeadm/kubeadm-ha-topology-external-etcd.svg

Papers:

- https://www.cs.rochester.edu/u/scott/courses/458/notes/FLP.pdf
- https://lamport.azurewebsites.net/pubs/byz.pdf
- https://lamport.azurewebsites.net/pubs/pubs.html#lamport-paxos
- https://lamport.azurewebsites.net/pubs/pubs.html#paxos-simple
- https://raft.github.io/raft.pdf