

CLOUD- PIZZASYSTEM

Simon Biewald

Erik Schippmann

Lennard Rudolph



AGENDA

- ▶ Demonstration Pizzasystem
- ▶ Zielstellung
- ▶ Architekturentwurf
- ▶ Deployment
- ▶ Bewertung

DEMONSTRATION CLOUDPIZZA

ZIELSTELLUNG

ZIELSTELLUNG

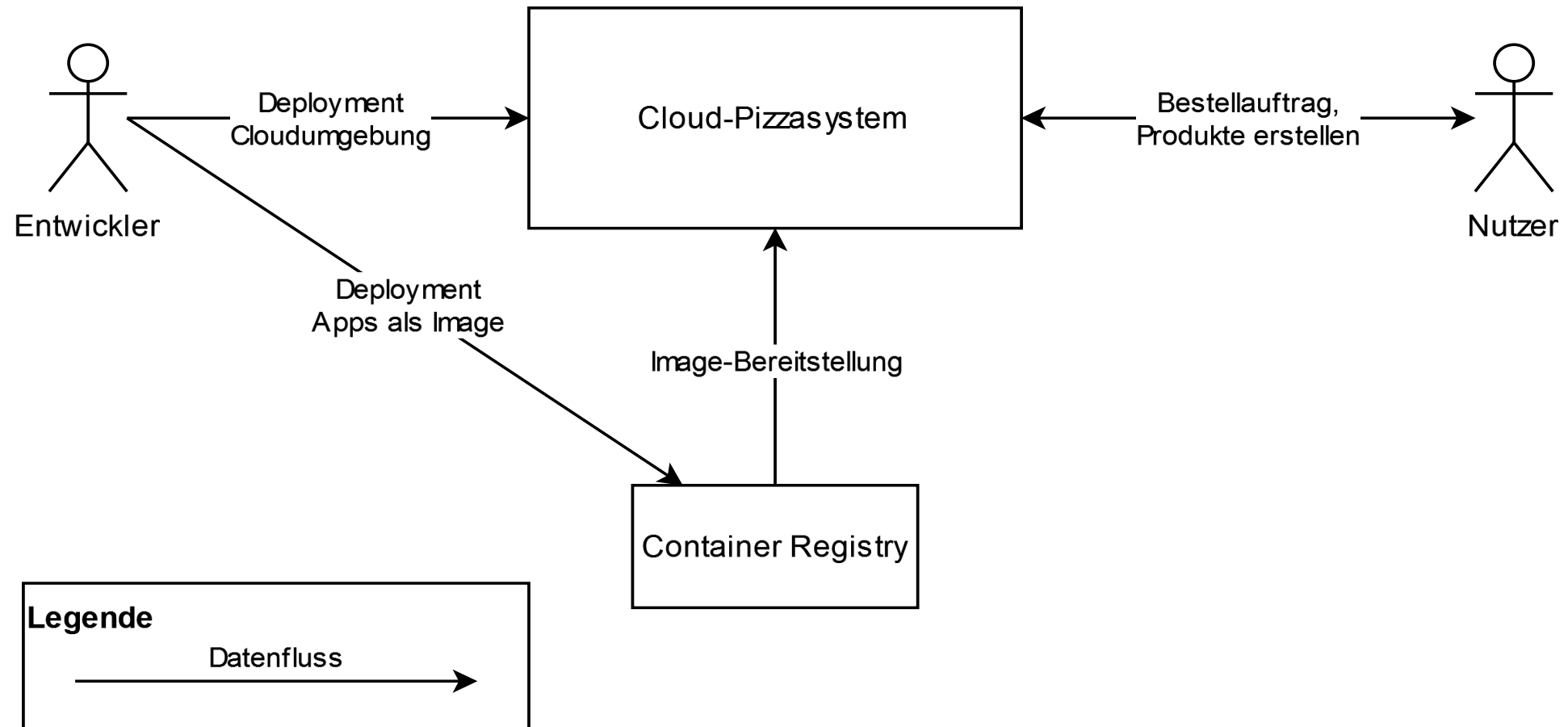
- ▶ Funktionierendes Kubernetes-Cluster in einem Cloudprovider
- ▶ Beispielapplikation Pizzasystem mit einigen Microservices
- ▶ Vorgaben
 - ▶ K8s-Distribution: kubeadm / k3s
 - ▶ Cloudprovider: Azure
 - ▶ IAC-Tool: Terraform

ARCHITEKTURENTWURF

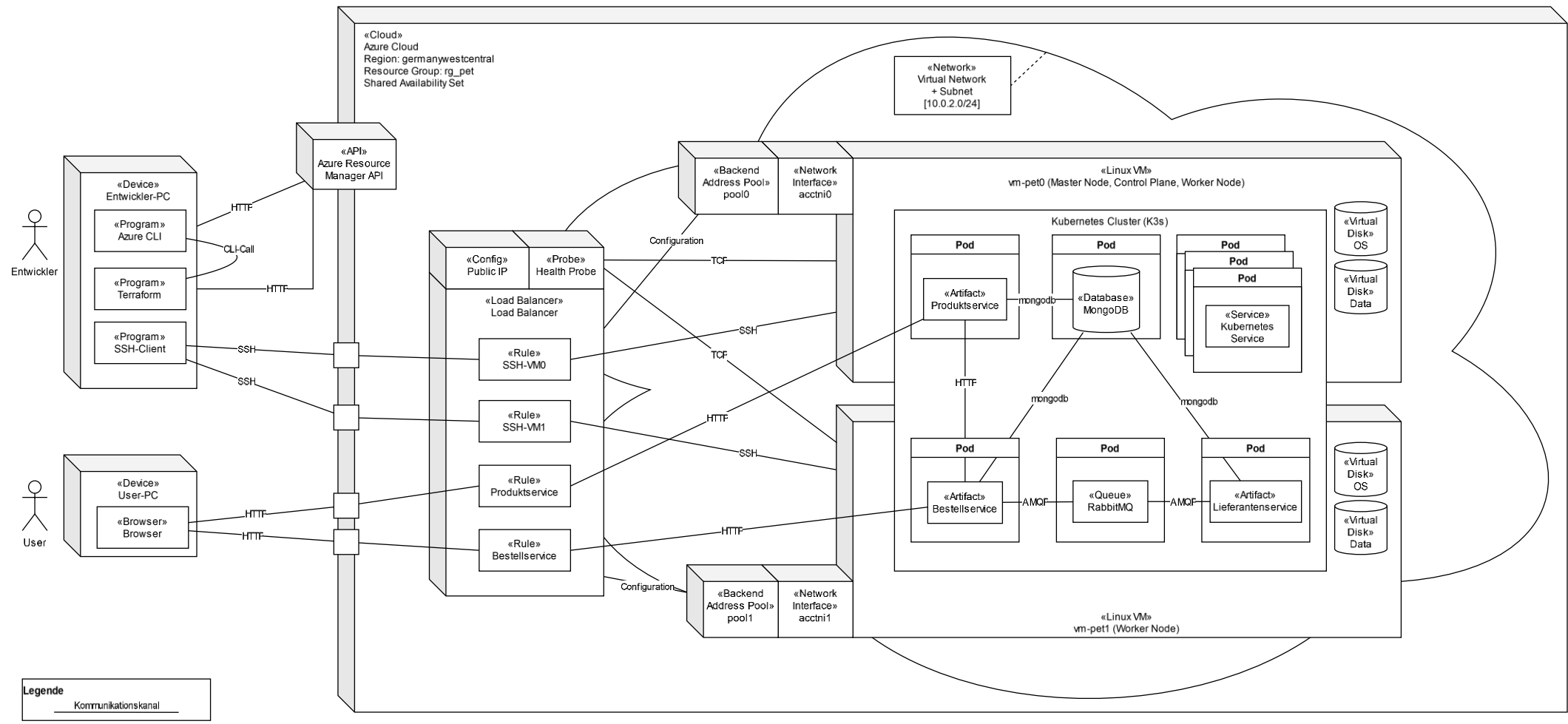
ARCHITEKTURENTWURF

- ▶ 2 VMs
- ▶ 3 Microservices
 - ▶ Bestellservice
 - ▶ Produktservice
 - ▶ Lieferantenservice
- ▶ Nötige Support-Infrastruktur

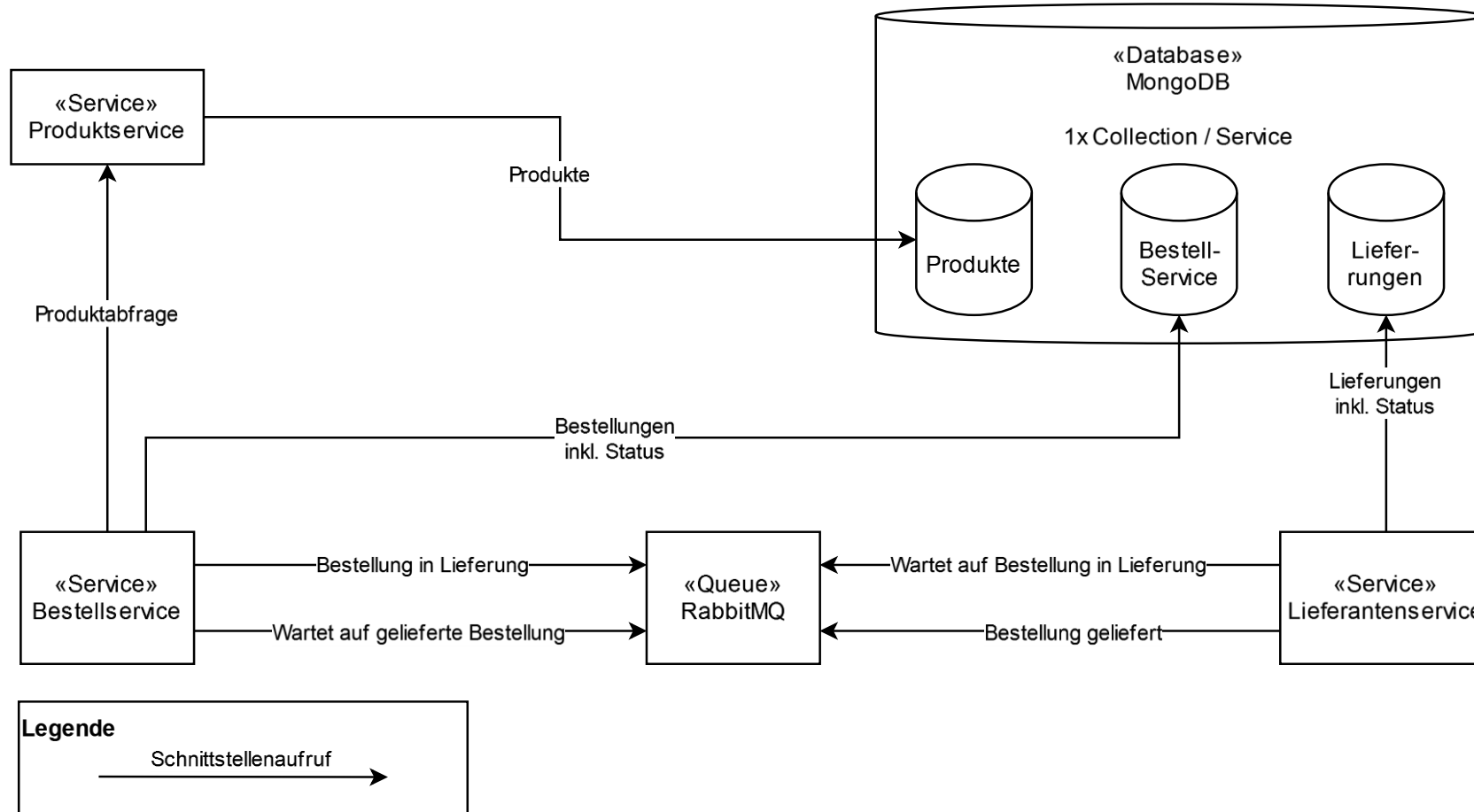
ARCHITEKTURENTWURF – KONTEXTSICHT



ARCHITEKTURENTWURF – VERTEILUNGSSICHT



ARCHITEKTURENTWURF – BAUSTEINSICHT

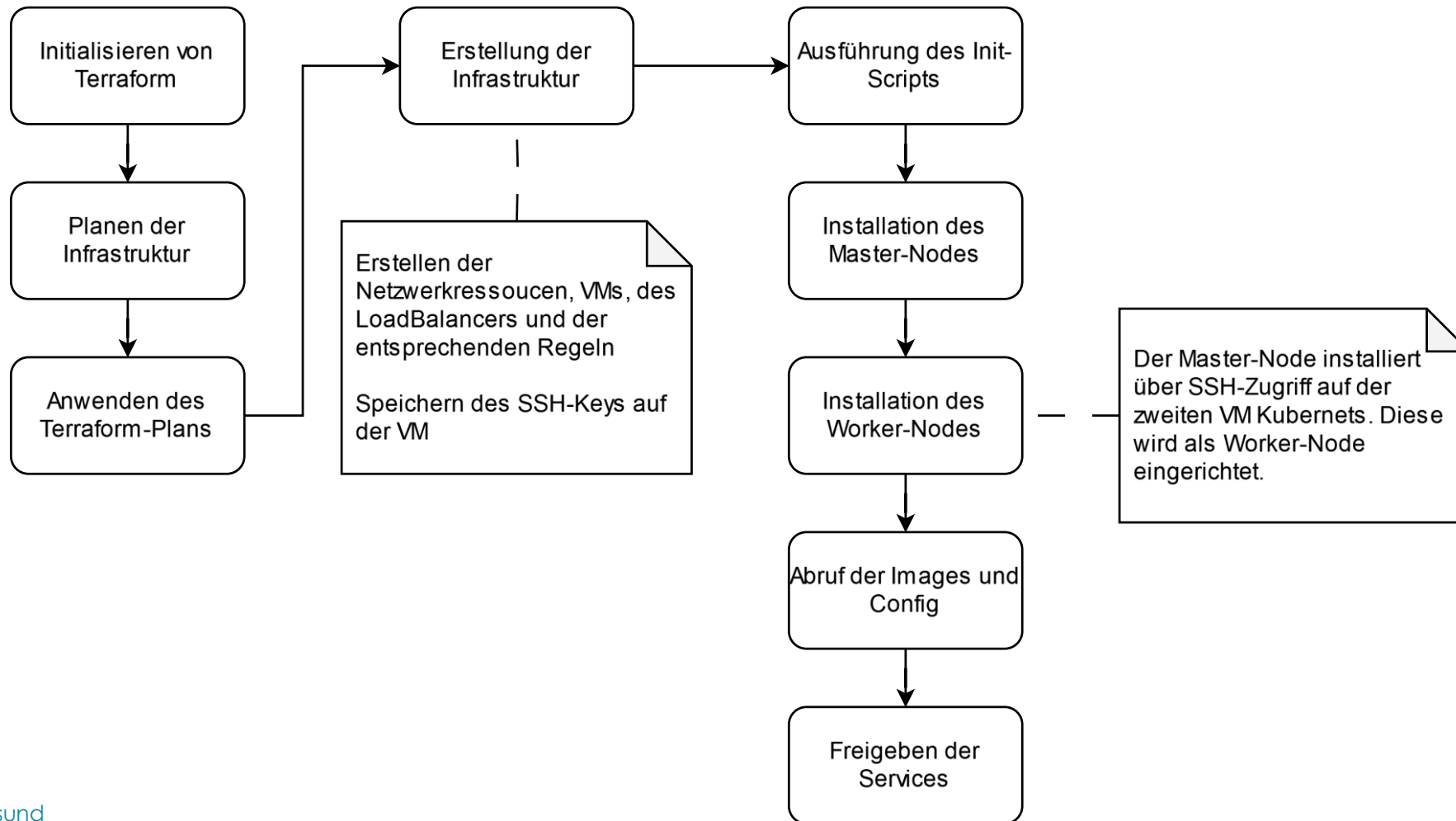


DEPLOYMENT

DEPLOYMENT – DEMONSTRATION

Starten des Deployments

DEPLOYMENT – ABLAUFDIAGRAMM



DEPLOYMENT – DEMONSTRATION

Auswertung des Deployments

BEWERTUNG

BEWERTUNG

- ▶ Deployment
 - ▶ Azure
 - ▶ Terraform
- ▶ Kubernetes
 - ▶ Kubeadm
 - ▶ K3s
- ▶ Gesamtsystem

BEWERTUNG – AZURE

- + Übersichtlicher, logischer als AWS
- + Besseres Free-Tier, gut zu evaluieren
- Sehr komplex, viele verschachtelte Ressourcen
 - Erfordert viel Einarbeitung, Erfahrung
- Langsames Deployment aus der Weboberfläche
- Rechtemanagement (500+ Rollen)
- Doppelungen Konten EntraID / Azure

BEWERTUNG – TERRAFORM

- + Gute Syntax, sinnvolle Struktur
- + Lösung für alle Probleme
- + Solides Laufzeitverhalten
- Dokumentation von Azure lückenhaft, verteilt
- Umständliches Debuggen
 - Destroy/Deploy der Infrastruktur ~2-5 min, teilweise späte Fehler
 - Fehler stoppen nicht restliches Deployment

Insgesamt besser als manuelle Konfiguration

BEWERTUNG – DEPLOYMENT

- ▶ Setup per Cloud-Init-Skript nur bedingt sinnvoll
 - ▶ Keine Zustandserfassung
 - ▶ Keine einfache Fehlerbehebung
 - ▶ Fragil
 - ▶ Kein Zugriff auf Provider-Inventar (Azure, K8s)
 - ▶ Skalierbarkeit
- ▶ Configuration Management dafür sinnvoller (z.B. Ansible, Chef)

BEWERTUNG – KUBEADM

Warum kein kubeadm?

- ▶ Kubeadm benötigt Netzwerkprovider (z.B. Calico)
- ▶ Calico ausprobiert, funktionierte aber nicht
 - ▶ Möglicherweise Versionsinkompatibilität mit Kubernetes 1.30
 - ▶ Komplexe Fehlersuche
- ▶ Umfangreiches Setup im Vergleich zu K3s



BEWERTUNG – K3S

Erfahrungen mit K3s

- + Einfaches Setup
- + Funktionierte direkt
- Wenig gelernt

BEWERTUNG – KUBERNETES INSGESAMT

- ▶ Sehr komplex, nur oberflächlich benutzt:
 - ▶ Einfacher Service & Deployment
 - ▶ Voreingestelltes managed Netzwerk
- ▶ Noch nicht betrachtet:
 - ▶ Stateful Containers & Volumes
 - ▶ Security (Segmentierung, Rollen, etc.)
 - ▶ Netzwerkmanagement
 - ▶ Upgrades
 - ▶ Datenbank-Replikationen, Datenbank-Performance
 - ▶ ...

BEWERTUNG – GESAMTSYSTEM

- ▶ Für kleines System zufrieden
- ▶ Azure, Terraform, K3s größtenteils gut zu benutzen
- ▶ Echte Skalierbarkeit/Redundanz erfordert großen Mehraufwand
- ▶ Weit von Production-Ready entfernt

FRAGEN?