



*Herzlich willkommen zum
4. Azure Rosenheim Meetup*

CONTAINER ON AZURE

AGENDA

18:30 Uhr: Ankunft und Begrüßung

19:00 Uhr: Überblick zu Container-Diensten auf Azure

- Basics (Was sind Container und wichtige Begriffe)
- Azure Container Registry (ACR)
- Azure Kubernetes Service (AKS)
- Azure Container Instances (ACI)

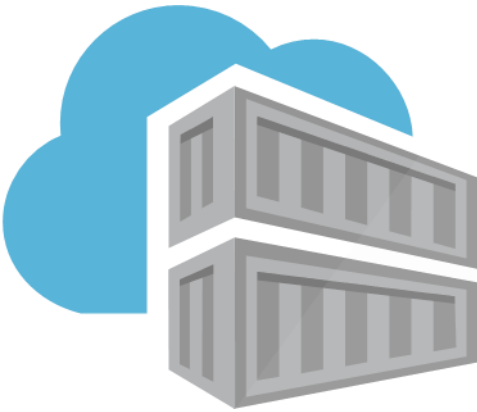
19.30 Uhr: Live-Demo

- Docker-Container erstellen und in AKS deployen
- Continuous Integration und Continuous Delivery (CI/CD Pipelines)
- Monitoring von Containern in Azure

20:00 Uhr: Azure DevOps

20:30 Uhr: Diskussion & Networking

21:00 Uhr: Ende der Veranstaltung



ÜBER UNS

white duck

- Spezialisiert auf Cloud-Computing, Software-Engineering und Data Analytics / BI -Technologien
- Ganzheitliches Angebot als Cloud Solution Provider - von der Entwicklung bis zum Betrieb von Cloud-Lösungen
- Gegründet 2012 mit Sitz in Rosenheim, derzeit 17 Mitarbeiter
- Erfahrung aus mehr als 15 Jahren Softwareentwicklung
- Technologie-Fokus: Microsoft Azure Cloud, .NET C#, .NET CORE, REST, Angular, ASP.NET
- Konzeption, Implementierung und Betrieb von SaaS-, Web-, Mobile- und IoT-Anwendungen

technologie

kompetenz

teamwork

white duck

Entwicklung, Beratung und Coaching rund
um Cloud-Computing



WAS TREIBT UNS AN?

- Aufbau und Etablierung einer Azure Community in der Region Südost-Bayern
- Azure KnowHow in der Region verteilen und vertiefen
- Azure Anwender zusammen bringen
- Neuigkeiten zu Azure besprechen und diskutieren
- Austausch zu Azure Projekten fördern

A large, irregular teal-colored ink splash or watercolor blotch serves as the background for the title. It has a textured, painterly appearance with darker and lighter shades of teal. Scattered around the main splash are numerous small, dark teal ink droplets and splatters, giving the overall design a dynamic and artistic feel.

GRUNDLAGEN

BUZZWORDS

Containerisierung

- „Virtualisierung der Virtualisierung“ (Auf Betriebssystemebene)
- Abstrahiert von Anwendungen zur Laufzeit (Erstellen von Containern)
- Anwendungen nutzen selben Kernel wie das OS

Container Image

- Anwendung + OS Abhängigkeiten

Container

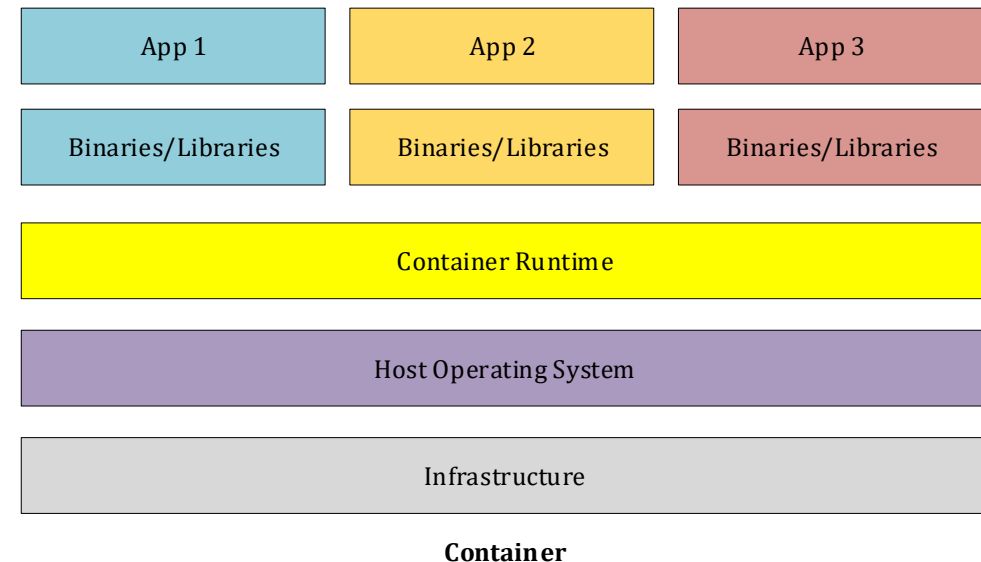
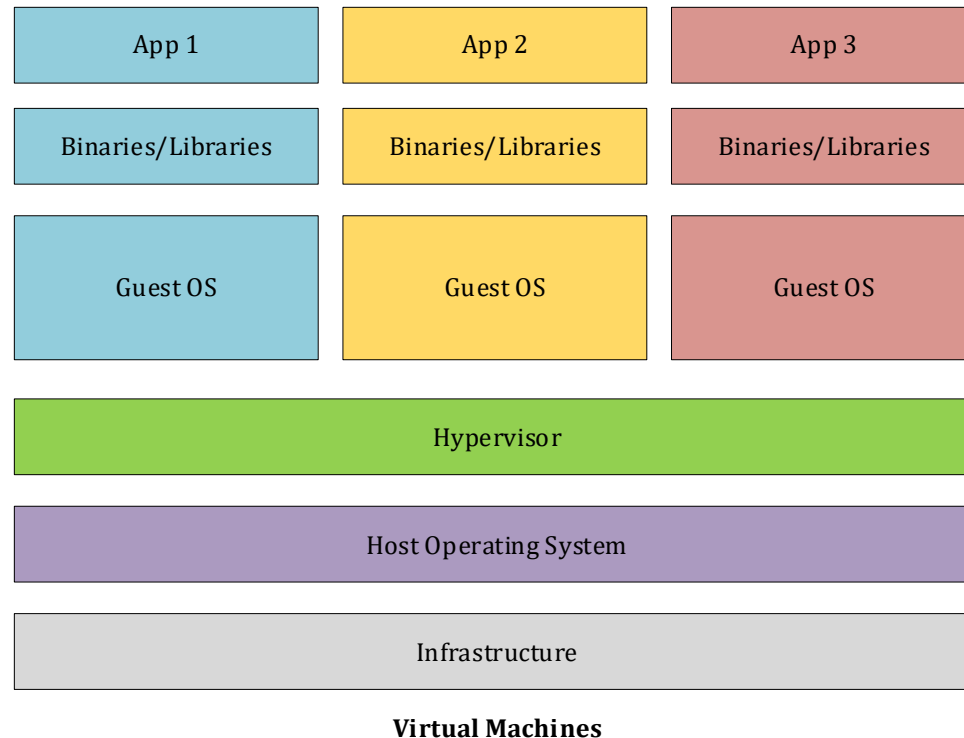
- Software Anwendung und Abhängigkeiten in Filesystem gebündelt (code, runtime, system tools, libraries, OS dependencies)
- Isolierte Umgebung
- Hilft gegen „dependency hell“

Container Registry

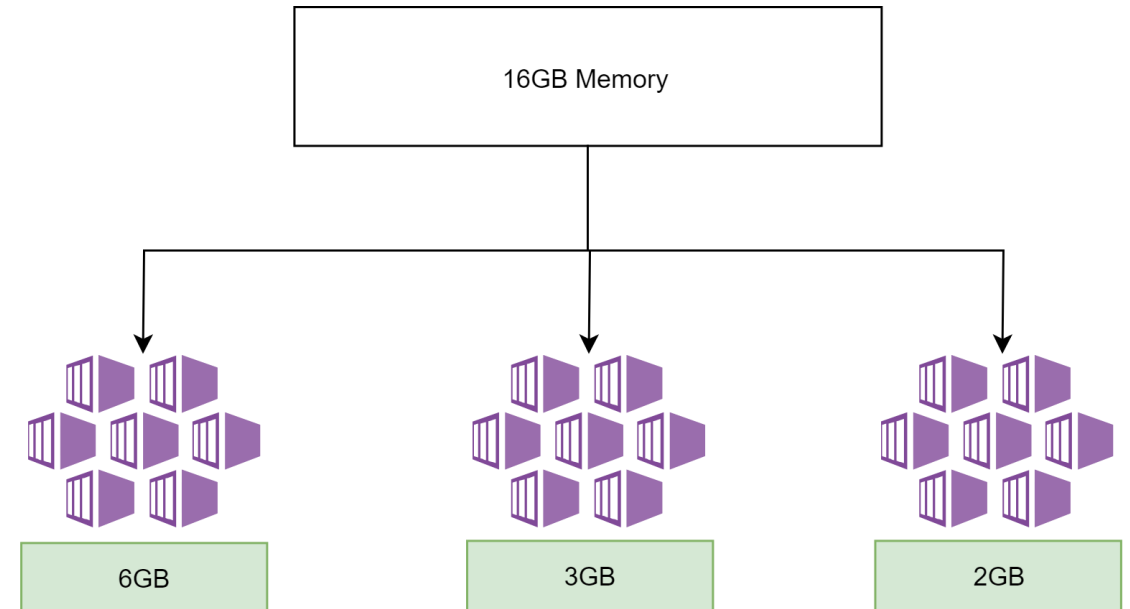
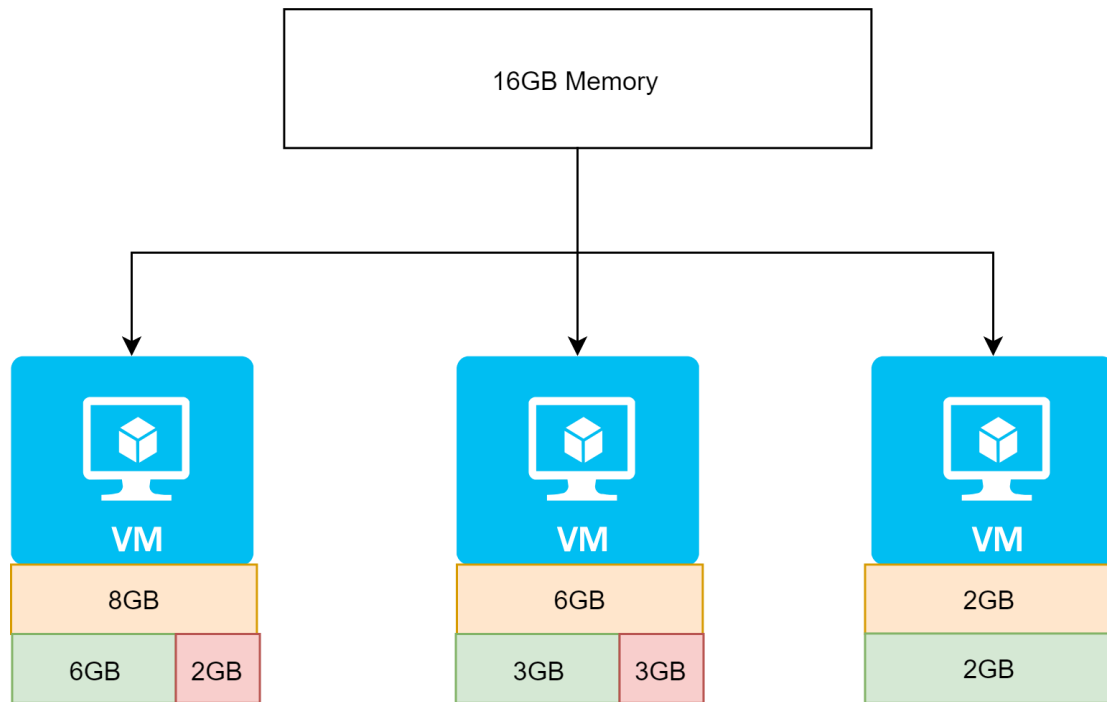
- Container Images speichern und verwalten
- Bereitstellung

VERGLEICH ZWISCHEN VM CONTAINER

- VM: Workloads teilen sich Host Ressourcen
- Container: Workloads teilen sich OS Ressourcen



BEISPIEL: NUTZEN



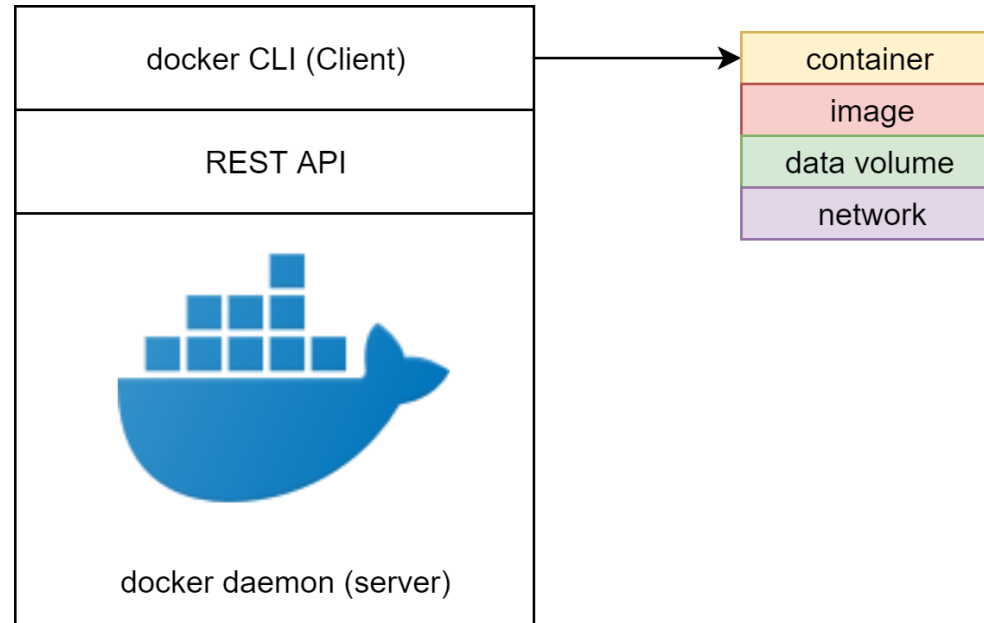
WAS IST DOCKER?

- Open Source Software
- Erlaubt es Anwendungen in Container zu überführen (Containervirtualisierung)
- Erleichtert Container zu erstellen
- Isoliert Anwendungen/Prozesse von der Infrastruktur
- Vereinfachung z.B mit Dockerfiles



DOCKER-ENGINE

- Client-Server Anwendung
- Docker daemon: Engine auf dem Host Rechner
- REST API: Kommunikation mit dem docker-daemon
- Docker client: CLI um mit dem daemon zu interagieren



VORTEILE VON DOCKER

Skalierbarkeit

- Leichtgewichtig
- Schneller start/stop

Portierbarkeit

- Snapshot von Environment
- Upload zur Registry
- Neue Container erstellen

Deployment

- Können fast überall eingesetzt werden
- Any App, any language, any stack
- Geringer overhead

Density

- Mehrere unterschiedliche Container können parallel auf einer Maschine laufen
- Reduziert Lizenz- und Ressourcenkosten

WAS MACHT EIN ORCHESTRATOR?

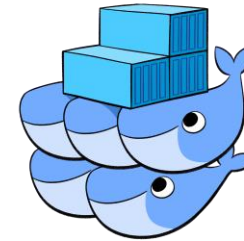
Automatisiert die Verwaltung von Containern

- Shutdown, Suspend, Clone, Spin
- Schnittstelle nach „außen“
- Steuer Zugriff auf Ressourcen (Netzwerk, Speicher)
- Load Balancing
- Einfaches „bewegen“ von Containern
- Health check

Cluster von VMs (Nodes)

- Orchestrator entscheidet welcher Container auf welchem Node läuft
- Verteilt mehrere Instanzen über den Cluster

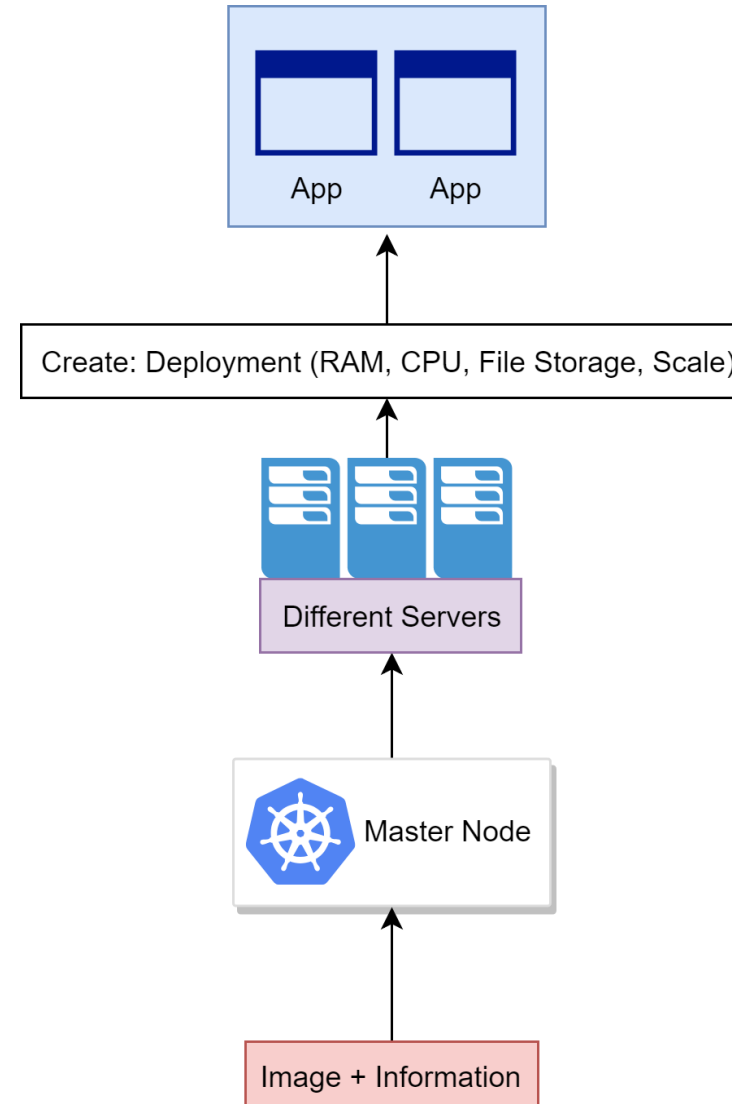
Beispiele: AKS, Kubernetes, Docker Swarm, Mesosphere, DC/OS



MEHR BUZZWORDS

Kubernetes

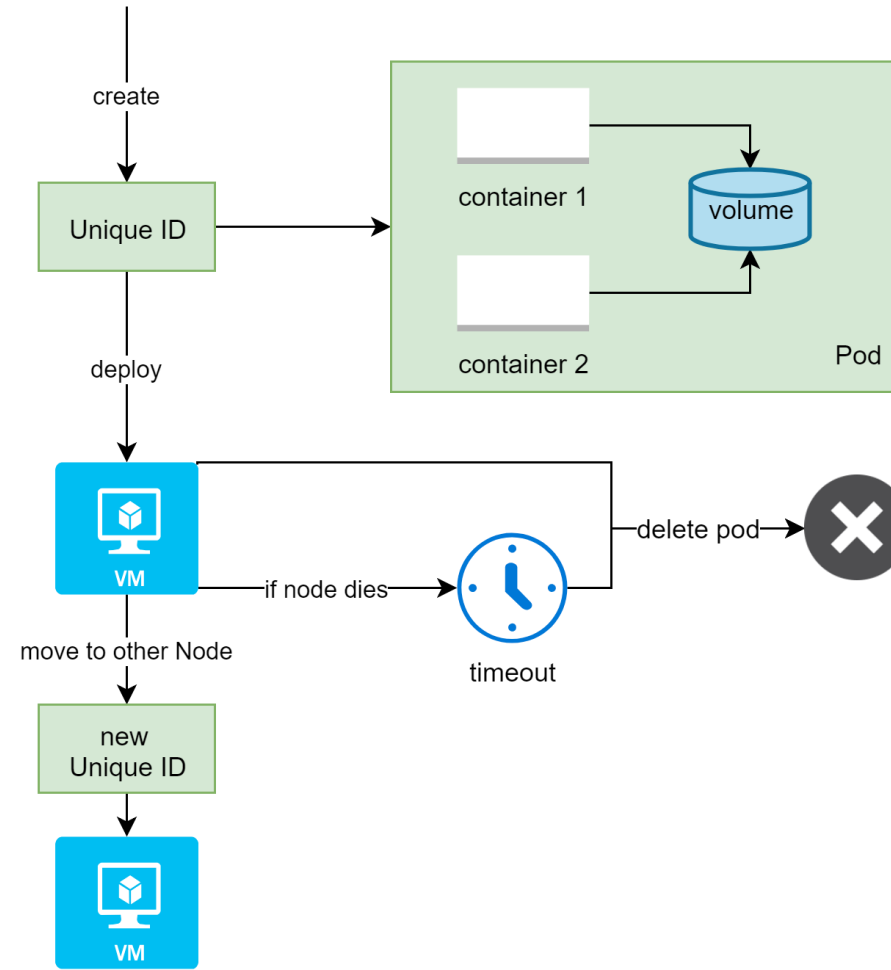
- Orchestrator
- Plattform für Containerverwaltung
- Automatisiert Management / Deployment
- Bietet
 - Skalierung
 - Load Balancing
 - Monitoring / Auto Recovery
 - Aufteilung in Service (Set aus Pods + Access policy)
 - Health check



MEHR BUZZWORDS

Pod

- Gruppe aus einem oder mehrerer Container
- Jeder Pod hat eine unique ID
- Container innerhalb teilen sich eine IP und Port range
- Container innerhalb über localhost erreichbar
- Inhalte haben shared content (namespaces, csgroups, ..)
- Shared volumes



A large, irregular teal-colored ink splash or paint blotch serves as the background for the title text. The splash has a textured, watercolor-like appearance with darker and lighter shades of teal.

AZURE CONTAINER REGISTRY(ACR)

AZURE CONTAINER REGISTRY (ACR)

- Image: Read only Container snapshot
- ACR erlaubt das managen und "aufbewahren" von Container images
- Aus diesen Images können neue Container erstellt werden (ACR build)
 - Automatischer rebuild und update von images
 - Für DC/OS, Docker Swarm, Kubernetes sowie Azure-Diensten wie Azure App Services, Batch, Service Fabric
- Pro Subscription mehrere ACR möglich
- Repository: Gruppe aus Container images
- Registry: 1-N Repositories



A large, irregular teal-colored ink splash or paint blotch serves as the background for the title text. The splash has a textured, watercolor-like appearance with darker and lighter shades of teal.

AZURE KUBERNETES SERVICE (AKS)

AZURE KUBERNETES SERVICE (AKS)



- Managed Kubernetes Service on Azure
 - Verwaltet gehostete Kubernetes Umgebung
 - Kubernetes API as a Service
 - Deployment und Management von Containeranwendungen
 - Übernimmt Health Monitoring
 - Automatische Updates und Patches
 - Leichte Skalierung des Clusters
 - Keine kosten für Master Nodes oder das Verwalten des Kubernetes Clusters nur für die Agents
- Eigenschaften
 - Nutzbar über CLI, Portal, Templates (ARM)
 - Easy to use (In nur 3 Befehlen lauffähig)
 - Zugriffssteuerung mit RBAC
 - AAD integration möglich
- Netzwerk
 - Basic: Die Netzwerkkonfiguration wird von Azure verwaltet
 - Advanced: Pods in eigens Konfiguriertes VNET

```
$ az aks create -g myResourceGroup -n myCluster --generate-ssh-keys  
\ Running ..
```

```
$ az aks install-cli  
Downloading client to /usr/local/bin/kubect1 ..
```

```
$ az aks get-credentials -g myResourceGroup -n myCluster  
Merged "myCluster" as current context ..
```

```
$ kubectl get nodes
```

NAME	STATUS	AGE	VERSION
k8s-mycluster-36851231-0	Ready	4m	v1.8.1
k8s-mycluster-36851231-1	Ready	4m	v1.8.1
k8s-mycluster-36851231-2	Ready	4m	v1.8.1

A large, irregular teal-colored ink splash or paint blotch serves as the background for the title text. The splash has a textured, watercolor-like appearance with darker and lighter shades of teal.

AZURE CONTAINER INSTANCES (ACI)

WAS SIND AZURE CONTAINER INSTANCES (ACI) ?



- Schneller Einstieg in Container
- Normalerweise: Container Hosting Infrastruktur zuerst
 - Container environment erstellen
 - VM provisionieren und managen
 - Image auswählen und richtig deployen
 - Wissen wie Orchestrierung/Verwalten/Skalieren funktioniert
- ACI: Container serverless betreiben
 - Azure verwaltet die Host VMs
 - Es muss nur das Container Image spezifiziert werden
 - Pay per second / only if running

EIGENSCHAFTEN VON ACI

- Starten von Containern in Sekunden
- Pricing
 - Pay per second
 - Günstig für “occasional workloads” (Ausprobieren, Load testing, batch jobs, load spikes)
 - Höhere Kosten für konstanten Betrieb / große Anwendungen
- Erlaubt genaue Spezifizierung von CPU und Speicher
- Jeder Container hat IP Adresse und FQDN (name.region.azurecontainer.io)
- Linux und Windows
- Integration von Azure CLI, PowerShell, c# fluent SDK, ARM templates
- Restart policies, Environment variables for containers, access logs, monitoring

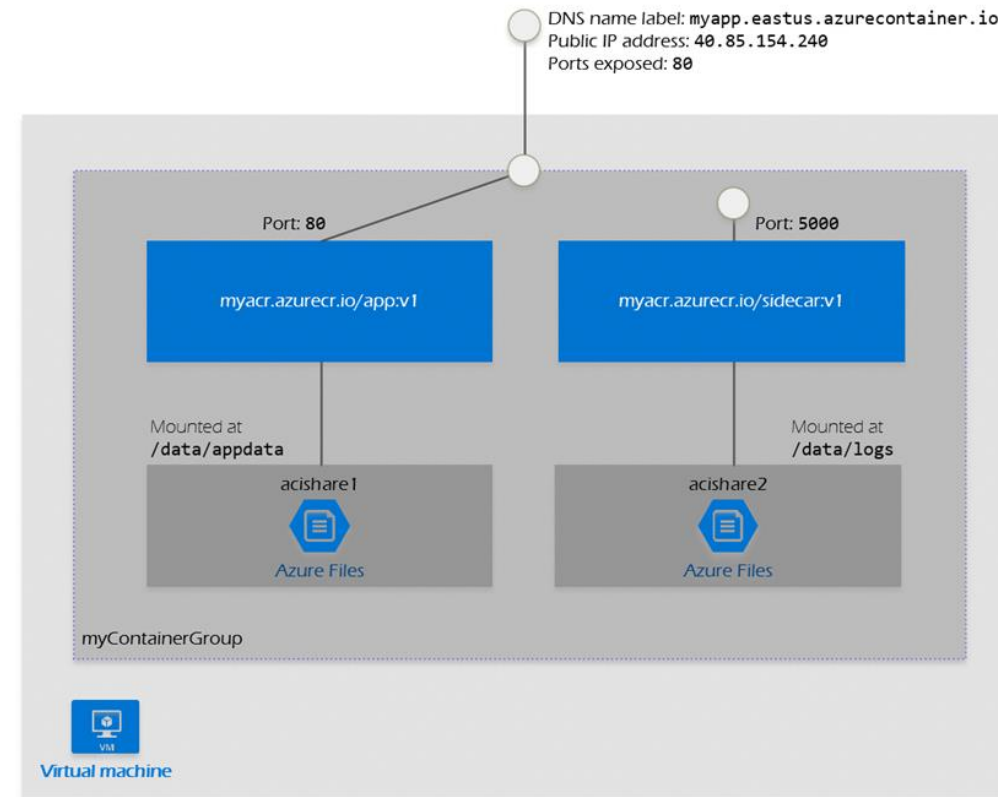
CONTAINER GROUPS

- Container die sich auf der selben host Maschine befinden
- Teilen sich ein lokales Netzwerk, Storage und Lebenszyklus
- Vergleichbar mit einem Kubernetes Pod
- Bestehen aus einem oder mehreren Containern
 - Teilen sich IP Adresse und Port namespace
- Erlauben das Nutzen des sidecar-patterns

Beispiel:

Application Container + Logging Container + Monitoring Container

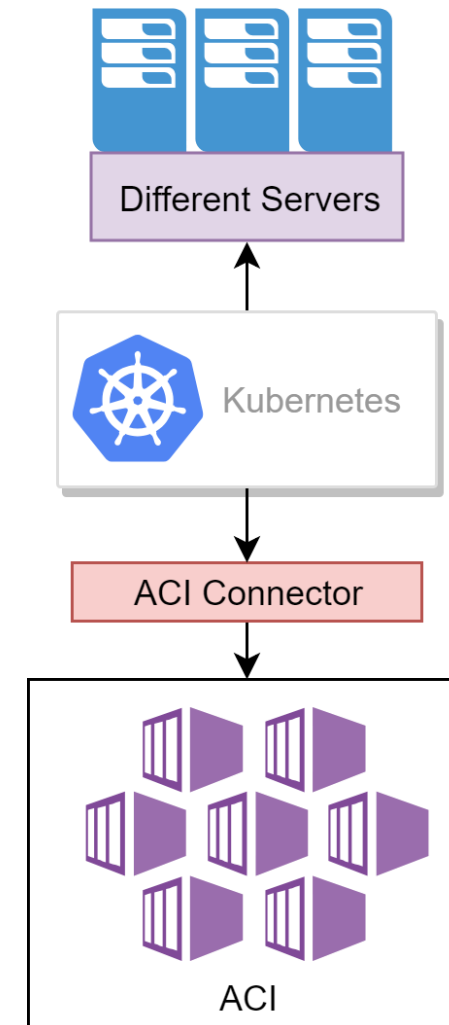
- Application Container enthält die Anwendung
- Logging Container sammelt Logs/Metriken und schreibt in storage
- Monitoring Container prüft Anwendung und Altered



<https://docs.microsoft.com/en-us/azure/container-instances/container-instances-container-groups>

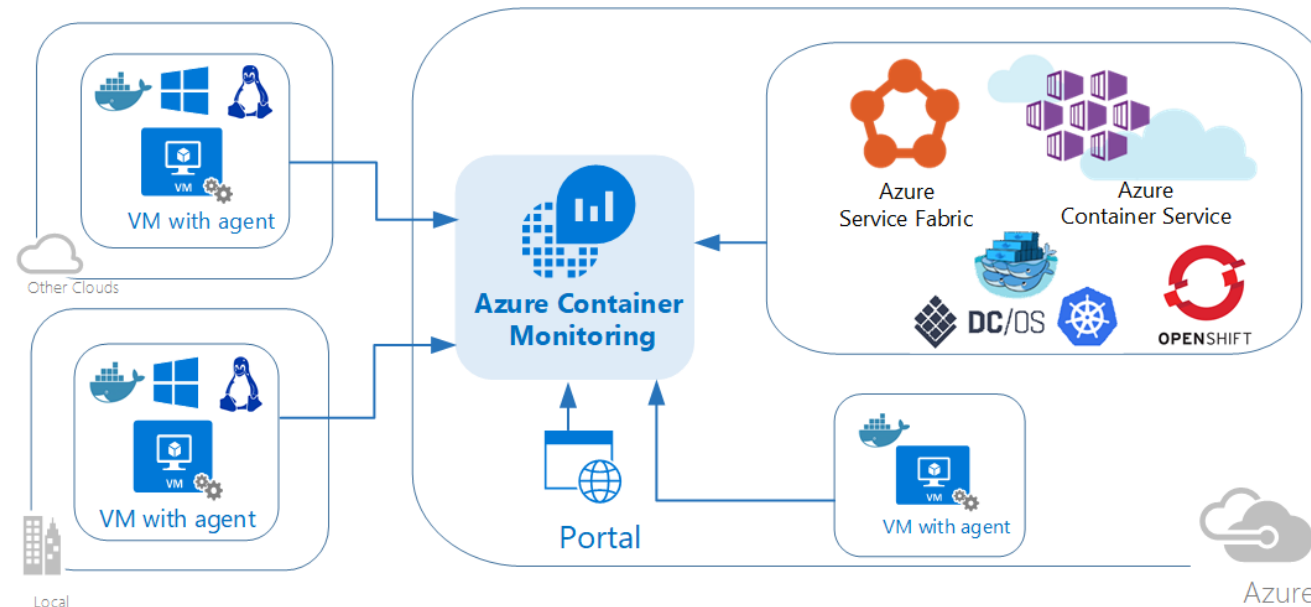
ACI UND ORCHESTRATOR

- Kein Ersatz für einen Container Orchestrator
- Kann aber die Funktionalitäten erweitern
 - Testen von experimentellen Containern
 - Kann Lastspitzen ausgleichen
- ACI connector für Kubernetes
 - ACI kann als virtuelles Node in einem Kubernetes Cluster genutzt werden
 - Z.b Abfangen von „burst loads“



CONTAINER MONITORING WITH LOG ANALYTICS

- Container Health
 - Identifiziert Container und sammelt Daten über Auslastung und Speicher
 - Gut zum Identifizieren von Bottlenecks
 - Gibt Einblick wo der Container betrieben wird
 - Performance Berechnungen
 - Cluster Auslastung
 - Zeigt ungewöhnliches Verhalten auf
- Container & Log Analytics



<https://docs.microsoft.com/en-us/azure/log-analytics/log-analytics-containers>

NÜTZLICHE LINKS



- Docker Dokumentation: <https://docs.docker.com/engine/docker-overview/>
- Azure Kubernetes Service: <https://docs.microsoft.com/en-us/azure/aks/intro-kubernetes>
- Azure Container Instances: <https://docs.microsoft.com/en-us/azure/container-instances/>
- Azure Container Registry: <https://docs.microsoft.com/en-us/azure/container-registry/>
- Kubernetes: <https://kubernetes.io/>
- Monitoring AKS: <https://docs.microsoft.com/en-us/azure/monitoring/monitoring-container-health>
- Monitor Container in Log Analytics: <https://docs.microsoft.com/en-us/azure/log-analytics/log-analytics-containers>
- Preisrechner: <https://azure.microsoft.com/de-de/pricing/calculator/>

- White duck Repository: <https://github.com/whiteducksoftware/azure-meetup-rosenheim>
- White duck Website & Blog: <https://whiteducksoftware.com/blog/>



A large, thick, teal-colored brushstroke arc that starts from the top right and curves around the right side of the slide, ending at the bottom right. It has a rough, hand-painted texture.

LIVE-DEMO