



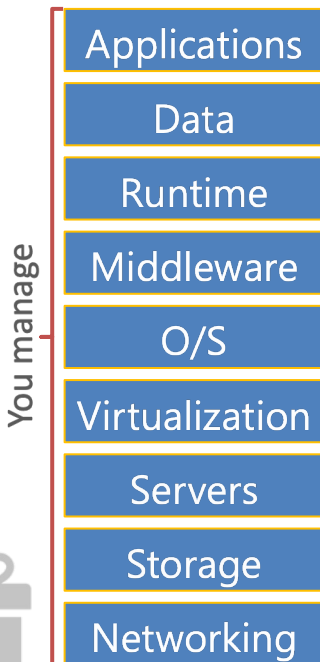
# Cloud Anbieter und Methoden

Heiter bis wolkig

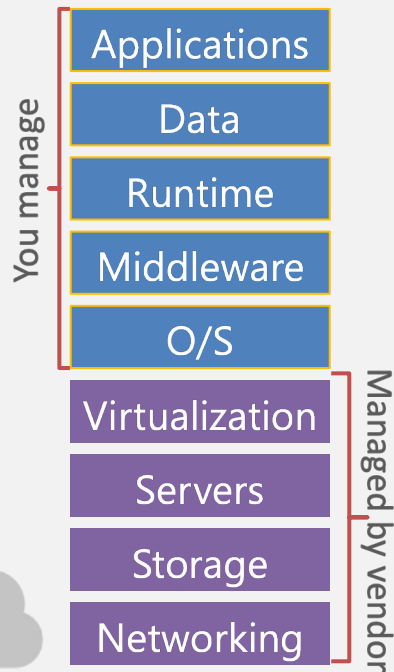


# Überblick

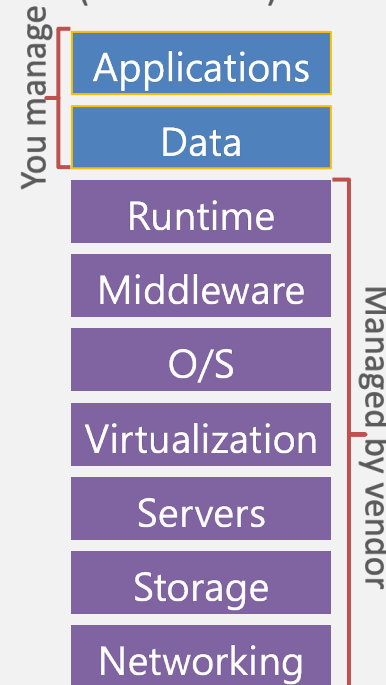
## Packaged Software



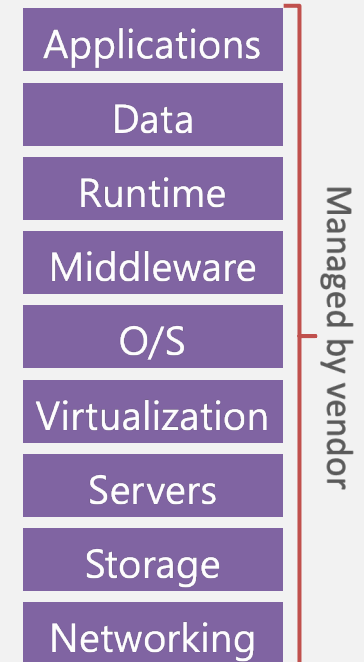
## Infrastructure (as a Service)



## Platform (as a Service)



## Software (as a Service)





# IaaS, Deployment Level 1

Entwickler

0) Wahl Angebot...



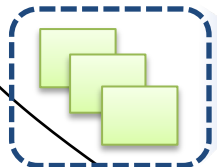
2) OS Image Wahl

3) DB Provisionierung

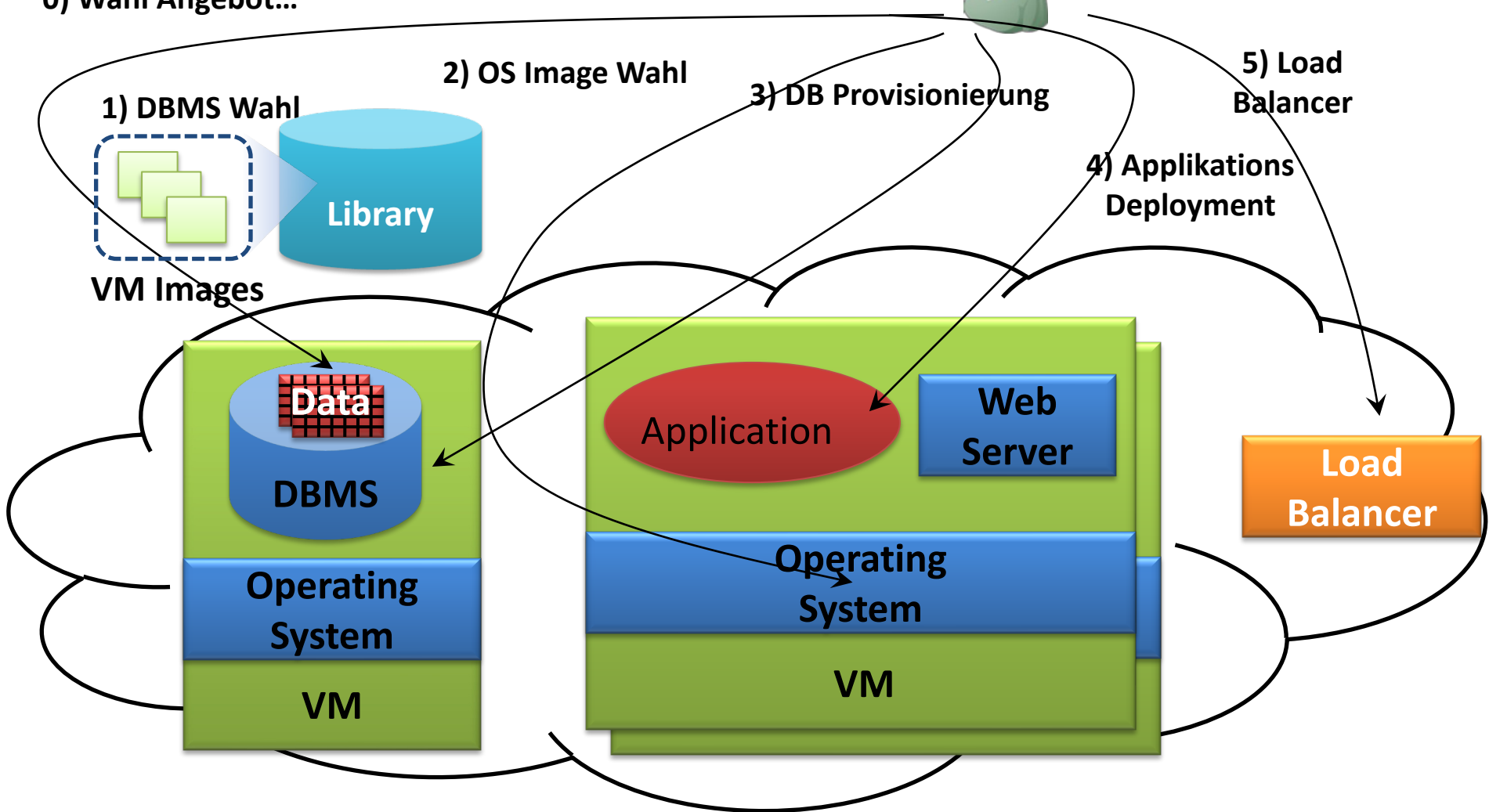
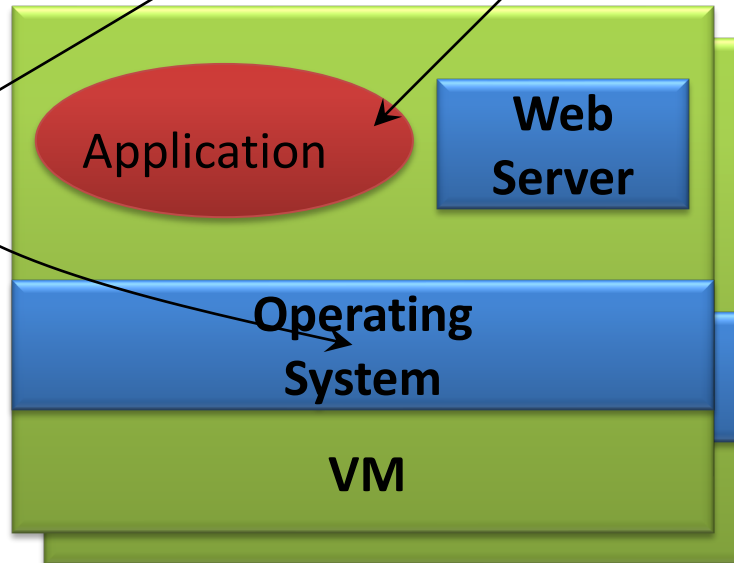
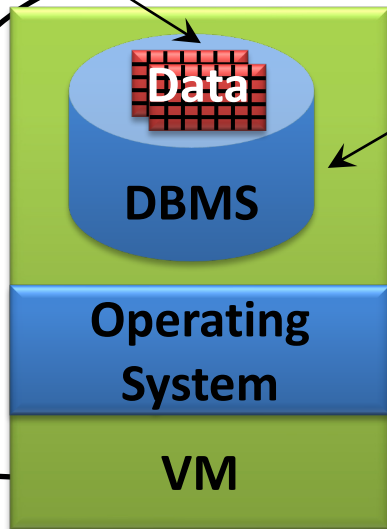
5) Load Balancer

1) DBMS Wahl

4) Applikations Deployment



VM Images





# IaaS, Deployment Level 2 & 3

- Deployment-Automatisierung:
  - z.B. Puppet, Chef, Ansible, ...
- Orchestrierung
  - Z.B. Ubuntu Juju, Kubernetes, OpenShift
- Auf Cloud-Anwendungsebene
  - Provider-Autoscaling
  - APIs direkt verwendet zur Skalierung
    - Infrastructure as Code



# IaaS #1 unsere Kollegen

z.B. 1&1 Hosting

Konfigurieren Sie sich Ihren Dynamic Cloud Server:

Keine Vertragslaufzeit  
**FLEX**

Sie haben Bedarf an maximaler Flexibilität und wollen Ihren Cloud-Server nur von Zeit zu Zeit aktivieren. Bis zu 6 Monate kostenloser Sleep Mode.

12 Monate Vertragslaufzeit  
**SPECIAL**

Sie benötigen Ihren Cloud-Server dauerhaft und wollen eine feste Basisleistung mit hoher Flexibilität kombinieren.

**Basis-Leistungen**  
Monatlicher Festpreis

**Flexible Leistungen**  
Jederzeit änderbar bei stundengenauer Abrechnung

vCore-Anzahl: 0-8  
Arbeitsspeicher (GB): 0-32  
Festplattenspeicher (GB): 0-800

Linux (CentOS 6 Minimalssystem (64 Bit))  
Microsoft Windows

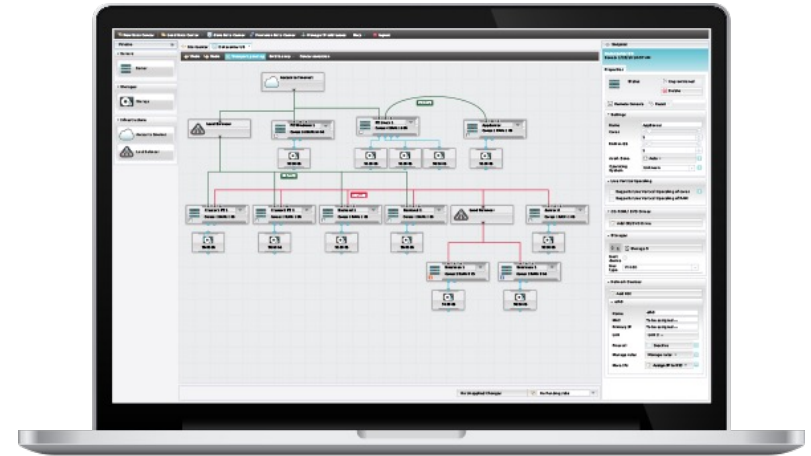
Aktion verlängert! Alle Betriebssysteme bis 31.12.2014 KOSTENLOS!

Gesamt: Monatspreis (Beispiel: 30 Tage) **0,7** €/Monat

Gesamt: Monatspreis (Beispiel: 30 Tage) **999** €/Monat

auswählen

z.B. Profitbricks



=> Live DEMO

- + Cloudflare CDN
- + MyWebsite SaaS
- + Cloud Storage
- + ...



# IaaS #2: Amazon AWS

- Juli 2002: AWS
- März 2006: S3
- August 2006: EC2
- ...

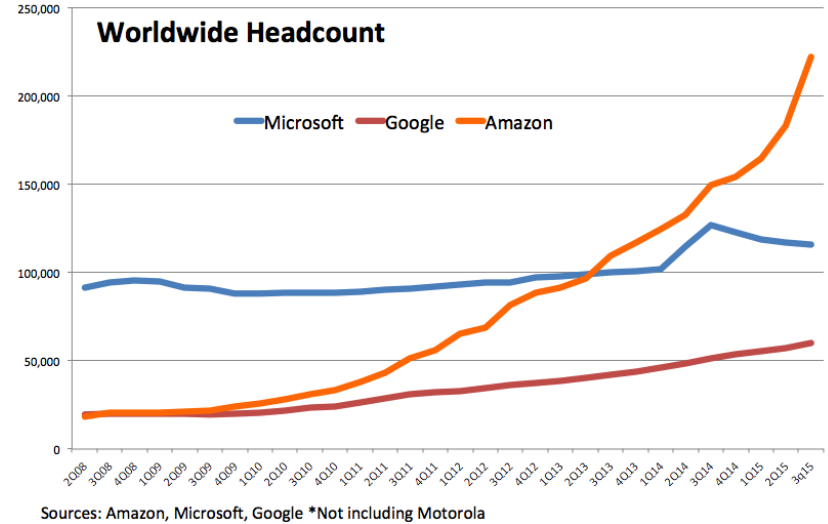
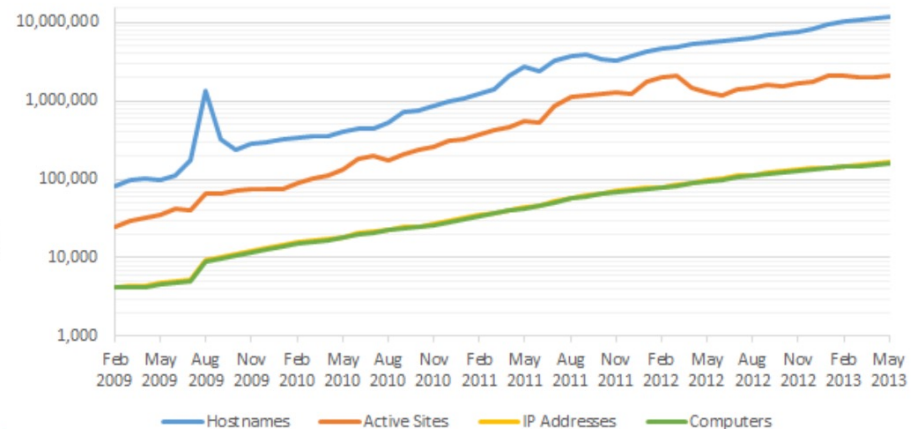
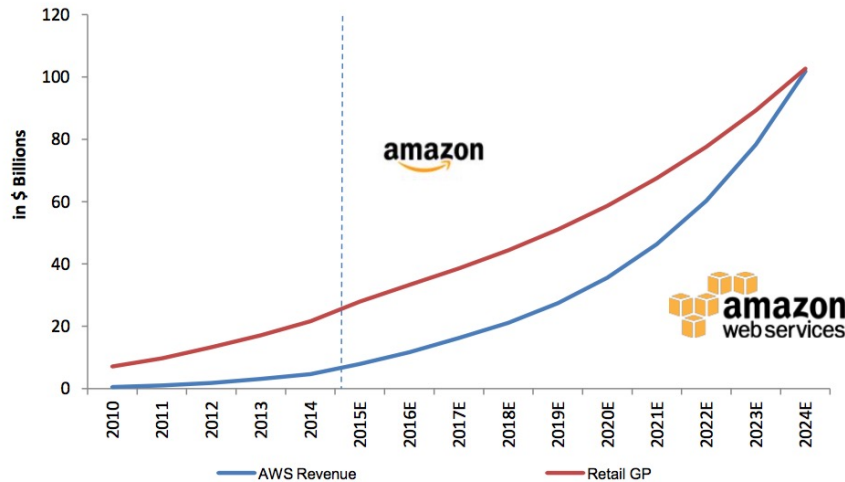


Figure 11: When Will AWS Become Larger than Amazon's Core Retail Business?

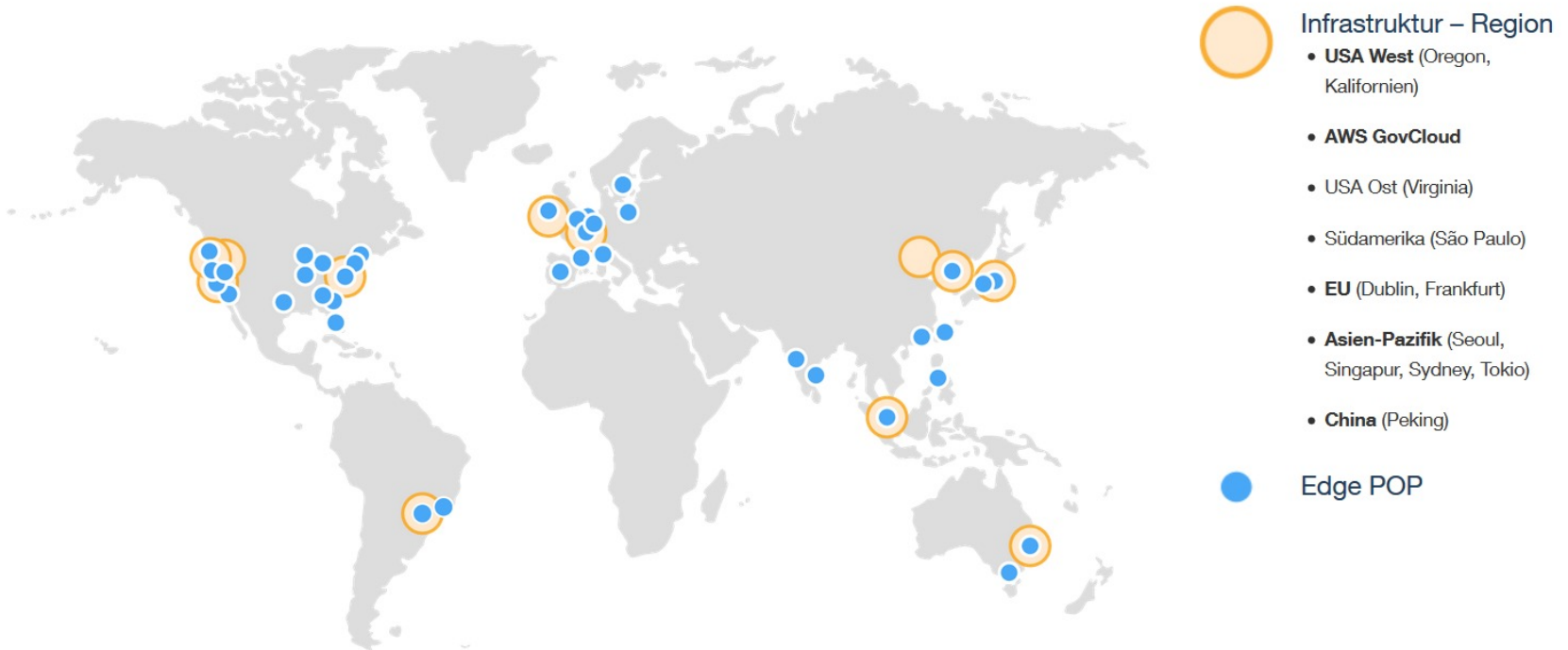


Source: Deutsche Bank, Amazon



# Amazon AWS Standorte

Globales Netzwerk von Regionen und Edge-Standorten





# AWS Komponenten



## Database

### DynamoDB

Predictable and Scalable NoSQL Data Store

### ElastiCache

In-Memory Cache

### RDS

Managed Relational Database

### Redshift

Managed Petabyte-Scale Data Warehouse

## Storage & CDN

### S3

Scalable Storage in the Cloud

### EBS

Networked Attached Block Device

### CloudFront

Global Content Delivery Network

### Glacier

Archive Storage in the Cloud

### Storage Gateway

Integrates On-Premises IT with Cloud Storage

### Import Export

Ship Large Datasets

## Cross-Service

### Support

Phone & email fast-response 24x7 Support

### Marketplace

Buy and sell Software and Apps

### Management Console

UI to manage AWS services

### SDKs, IDE kits and CLIs

Develop, integrate and manage services

## Analytics

### Elastic MapReduce

Managed Hadoop Framework

### Kinesis

Real-Time Data Stream Processing

### Data Pipeline

Orchestration for Data-Driven Workflows

## Compute & Networking

### EC2

Virtual Servers in the Cloud

### VPC

Virtual Secure Network

### ELB

Load balancing Service

### WorkSpaces

Virtual Desktops in the cloud

### Auto Scaling

Automatically scale up and down

### DirectConnect

Dedicated Network Connection to AWS

### Route 53

Scalable Domain Name System

## Deployment & Management

### CloudFormation

Templated AWS Resource Creation

### CloudWatch

Resource and Application Monitoring

### Elastic Beanstalk

AWS Application Container

### IAM

Secure AWS Access Control

### CloudTrail

User Activity Logging

### OpsWorks

DevOps Application Management Service

### CloudHSM

Hardware-based key storage for compliance

## App Services

### CloudSearch

Managed Search Service

### Elastic Transcoder

Easy-to-use Scalable Media Transcoding

### SES

Email Sending Service

### SNS

Push Notification Service

### SQS

Message Queue Service

### SWF

Workflow Service for Coordinating App Components

### AppStream

Low-latency Application Streaming

**AWS Global Physical Infrastructure**  
(Geographical Regions, Availability Zones, Edge Locations)

[http://media.amazonwebservices.com/AWS\\_Overview.pdf](http://media.amazonwebservices.com/AWS_Overview.pdf)





# AWS Komponenten: EC2

- Virtuelle Instanzen für das Ausführen von AMIs (Amazon Machine Images)
- Instanztypen
  - m: Allgemein
  - c: Prozessorpower
  - g: GPU-power
  - i: I/O
  - + Size (micro..8xlarge)

<https://aws.amazon.com/de/ec2/instance-types/>

<http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/instance-types.html>

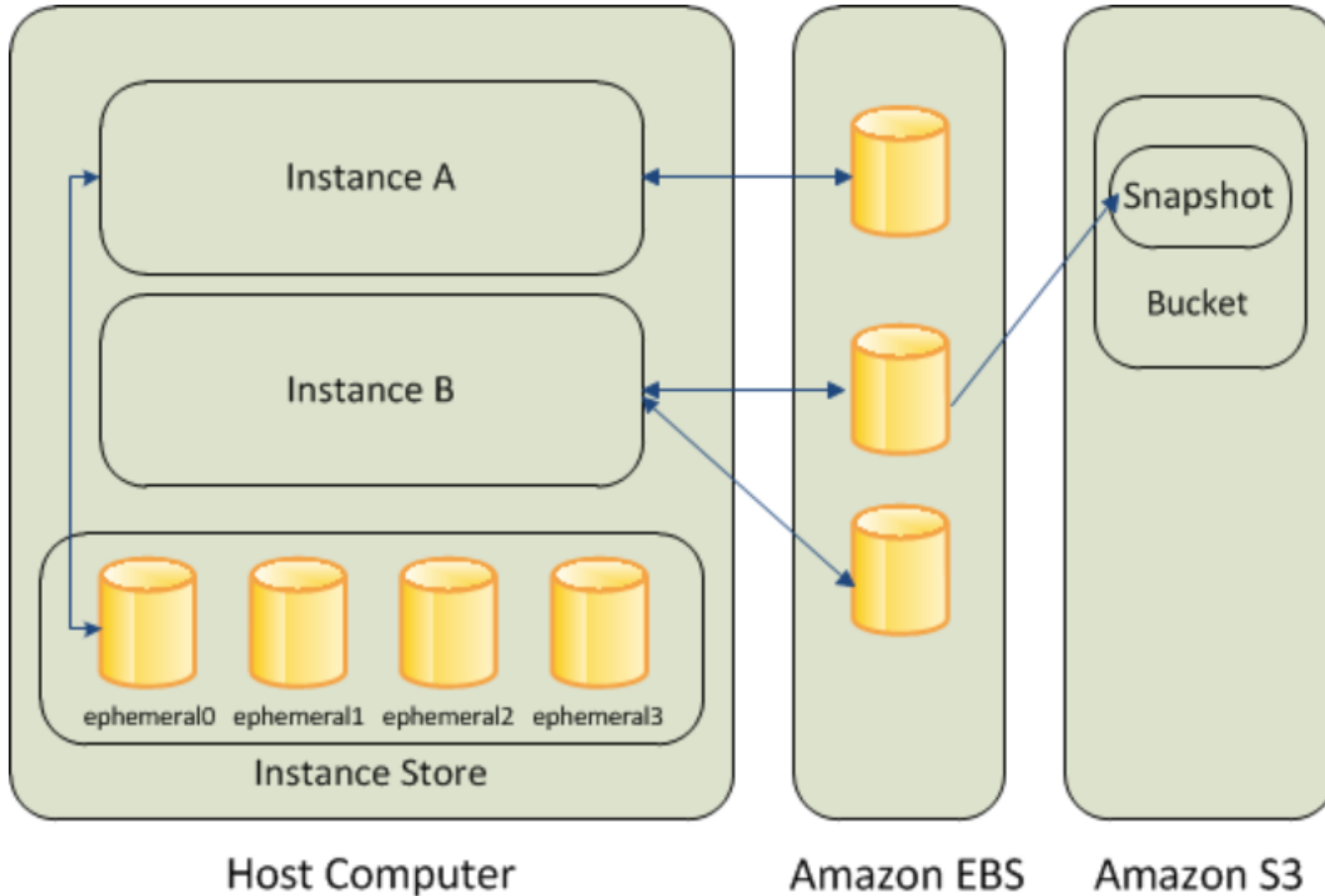


# AWS Komponenten: EC2

- Besondere Instanzklassen:
  - Reserved
  - Spot
- Granularität:
  - Regions
  - Availability Zones
  - Placement Groups



# AWS Storage



(<http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/Storage.html>)



# AWS Komponenten: S3

- Storage Objekte bis 5TB
- Buckets in Regions
- REST / SOAP zur Verwaltung, HTTP oder Bittorrent zum Download
- Redundanzen führen zu einem SLA von
  - 99,999999999% Zuverlässigkeit (bzgl. Verlust)
  - 99,99% Verfügbarkeit
- Alternative: EBS, Zuverlässigkeit höher als bei Festplatten, günstiger, Block Device



# AWS API, Tools und Architekturen

APIs in Folien abkupfern ist doof.

Daher springen wir direkt hier hin:

- <http://docs.aws.amazon.com/general/latest/gr/Welcome.html>
- <http://docs.aws.amazon.com/AWSEC2/latest/APIReference/making-api-requests.html>

Dasselbe gilt für Tools-Übersichten:

- <https://aws.amazon.com/de/tools/>

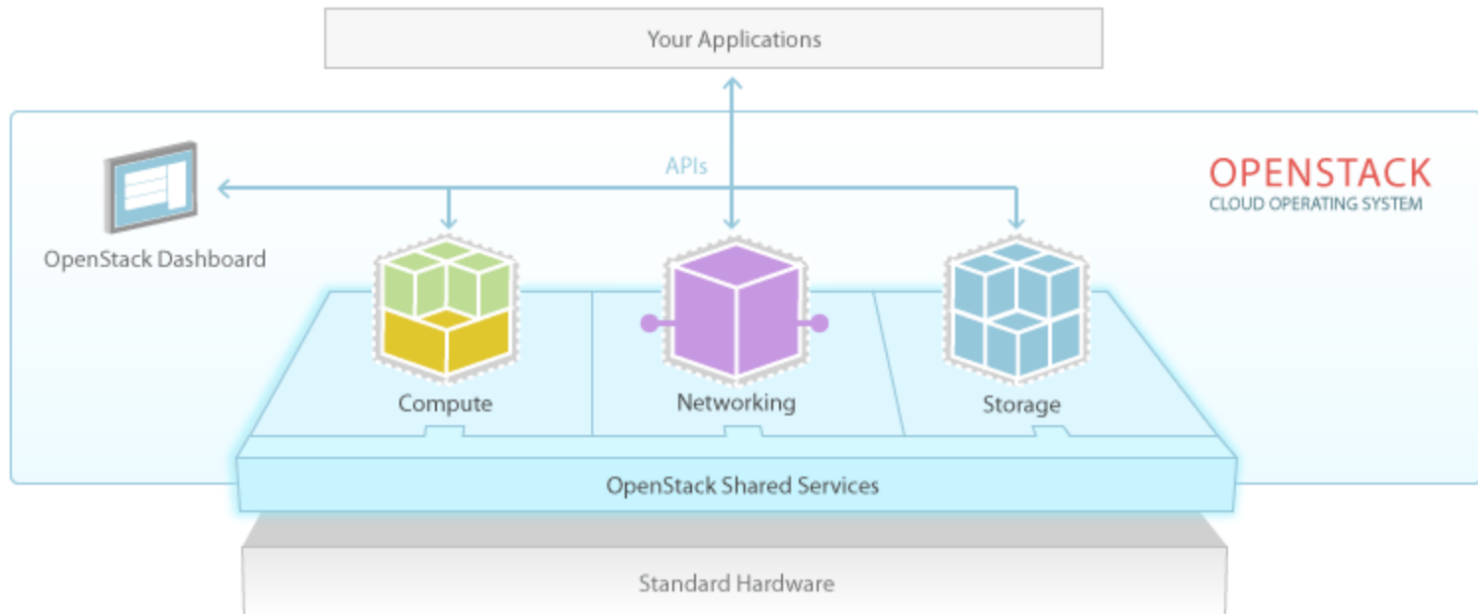
Auch für Beispielarchitektur ist AWS toll:

- <http://aws.amazon.com/de/architecture/>



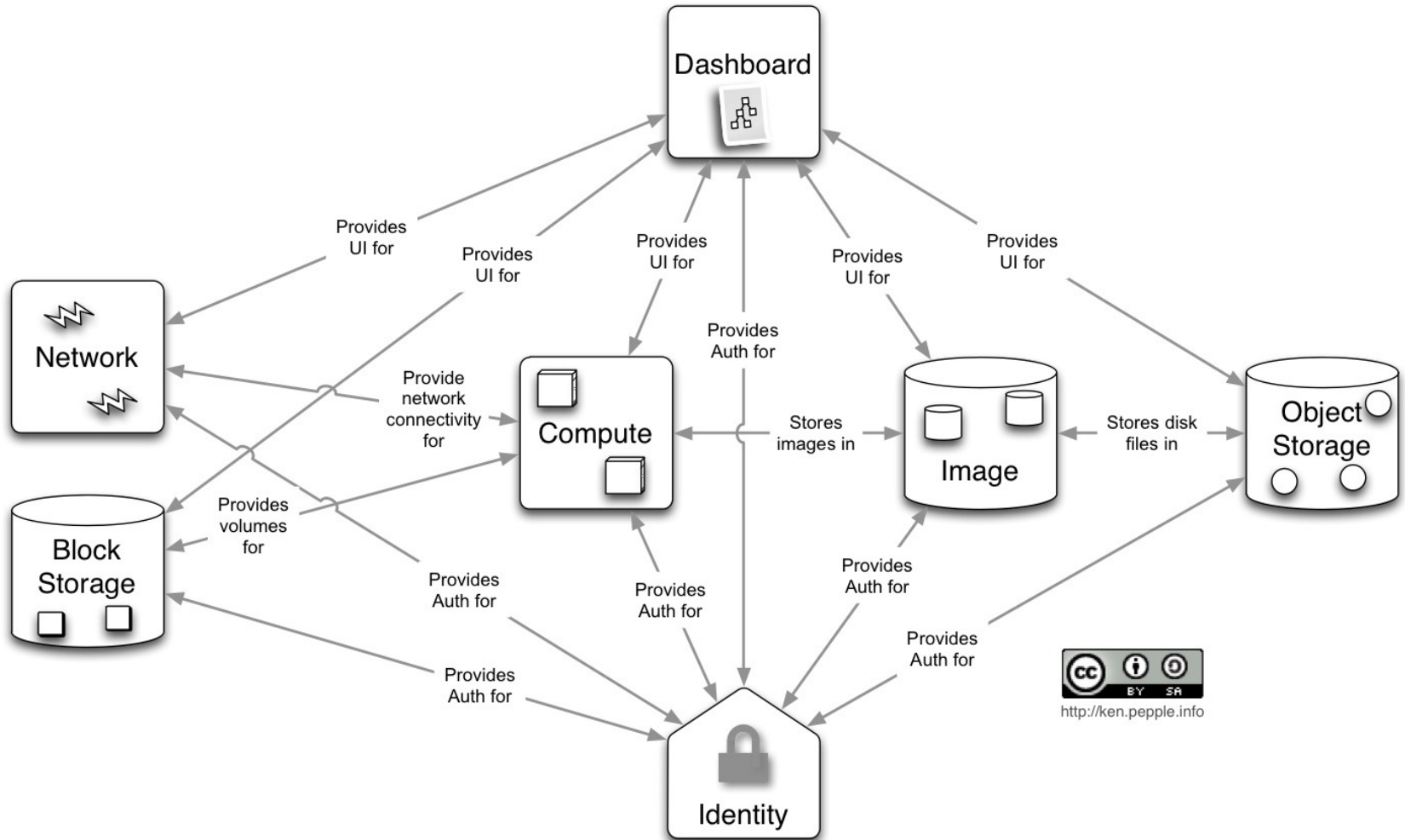
# IaaS #2: Openstack

- State Of The Art Open Source Cloud Software
- Sehr starke Industriebeteiligung
  - Sogar NSA... (<http://www.youtube.com/watch?v=NgahKksMZis>)



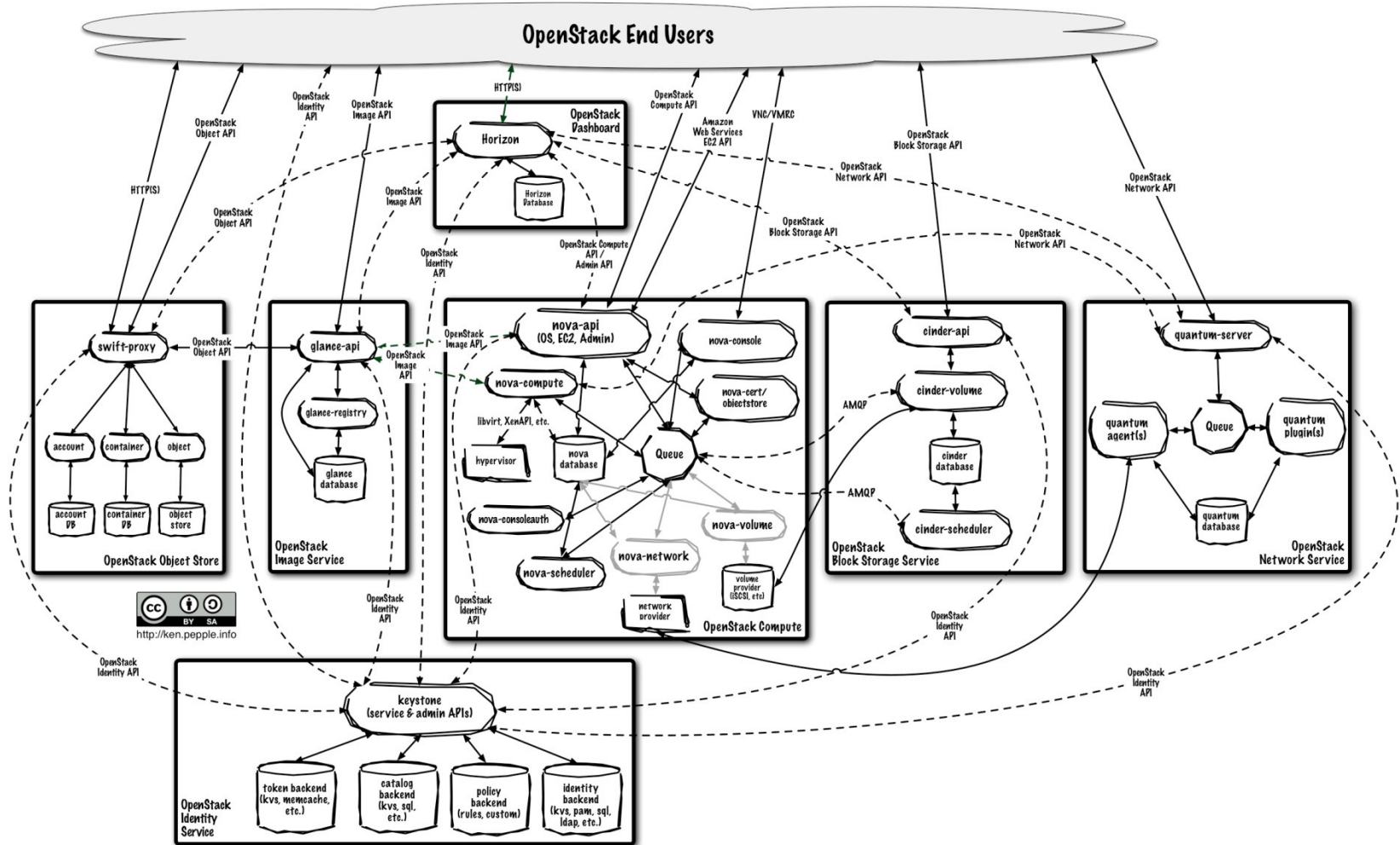


# Openstack Gesamtkonzept





# Openstack logische Architektur







# Openstack: Verfügbarkeit

- Source via git
- Linux Distributionen
  - Dediziert
  - enthalten
- Installer
  - Chef, Puppet, Juju, Developer Installer...
- Cloud Provider: public clouds
- Kommerzielle Distributionen



# Openstack Dashboard: Horizon

Project Users - OpenStack D... x

Logged in as: admin [Settings](#) [Sign Out](#)

## Users for Project: demo

Remove Users

<input type="checkbox"/>	ID	User Name	Email	Enabled	Actions
<input type="checkbox"/>	779d6a4a2bfe407caa62256d3e9fb4ba	admin	admin@example.com	True	<a href="#">Remove User</a>
<input type="checkbox"/>	fb9e9667d6eb4ba59ac2bbc885d7d890	demo	demo@example.com	True	<a href="#">Remove User</a>

Displaying 2 items

## Add New Users

ID	User Name	Email	Enabled	Actions
32d92034862d4c73ad25b83f22335479	nova	nova@example.com	True	<a href="#">Add To Project</a>
c8e76d5da6474adba9cb2161802105df	glance	glance@example.com	True	<a href="#">Add To Project</a>
4b35949bd96d4804aac81c55d196193b	swift	swift@example.com	True	<a href="#">Add To Project</a>
e2b1ab40b9234a5889c91f11f7f8cc52	scott	-	True	<a href="#">Add To Project</a>
0f8f6378ebe24b8290f6ff80cf5683d3	jesse	-	True	<a href="#">Add To Project</a>
750bbf9ac4534c2ebdd496811be2f4cf	dolph	-	True	<a href="#">Add To Project</a>



# Openstack Compute: Nova

- Unterstützt diverse Hypervisoren:
  - Citrix XenServer, Xen, KVM, VMWare ESX, LXC, QEMU, UML, (Microsoft Hyper-V)
  - (<https://wiki.openstack.org/wiki/HypervisorSupportMatrix>)
- EC2-API kompatibel
- Mandantenfähig, stark skalierbar
- Block Storage ansprechbar über
  - AOE: ATA over Ethernet
  - IET iSCSI
  - LVM RBD: Rados Block Device, netzwerk block device mit Objekten in Ceph
  - Sheepdog: verteiltes Speichersystem für KVM
  - Solaris iSCSI: auf Solaris/ZFS
  - HP SAN: HP StorageWorks P4000 SAN target



# Openstack Storage: Swift, Cinder, Glance, ...

- Swift: Object Storage
  - URL Adressierung
  - Eventual Consistent
    - Auditors & Replicators
  - RESTful API
  - Hierarchie: Accounts, Containers, Objects
  - Regions, Zones, Partitions, Ringe (consistent hashing)
- Cinder: Block Storage
  - Logische Volumes, viele Backend driver
  - (<https://wiki.openstack.org/wiki/CinderSupportMatrix>)
- Glance: VM Image Storage
- Ephemeral Storage



# Openstack Sonstige: Neutron, Keystone

- Neutron (Quantum): Netzwerkmodell
  - Diverse Backends, von VLAN bis SDN
  - Durch Plugins erweiterbar z.B. mit IDS, LB, VPN  
Deployments, siehe  
[https://wiki.openstack.org/wiki/Neutron#Using\\_Neutron](https://wiki.openstack.org/wiki/Neutron#Using_Neutron)
  - Architektur:  
<https://docs.openstack.org/security-guide/networking/architecture.html>
- Keystone:
  - Identity, Token, Catalog, Policy Services
  - Authentication, Authorization, Users, Groups,  
Permissions



# IaaS #3: kubernetes

- Container Orchestrierungssystem
- Unterstützt Docker und andere
- Open Source, in Go geschrieben
- Grundlage: Google
- Erstveröffentlichung: 2014/15
- Darauf Aufbauende Projekte sprießen, wichtig: HELM



# IaaS #3:

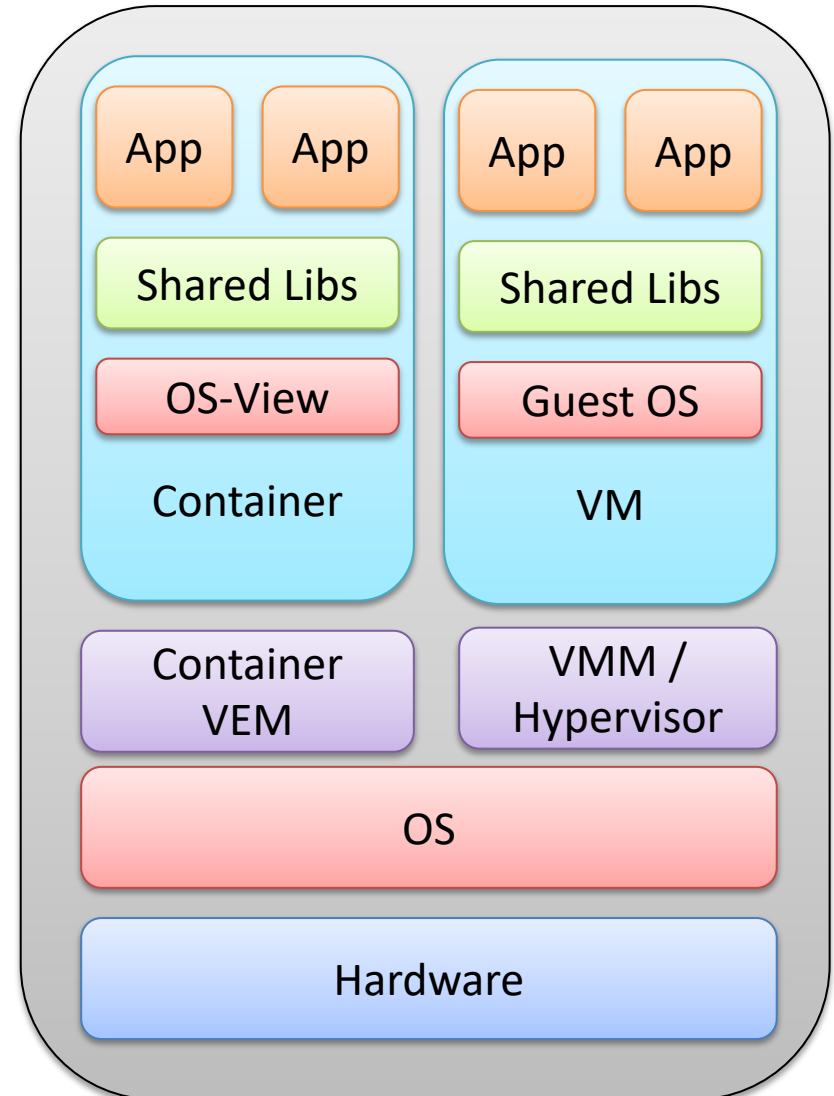
## Erinnerung: Container / VM

### Container:

- Ressource schonend
- Startet schnell
- Verwendet „Host Kernel“
- Isolierung über eigenen Namespace für Prozesse, Netzwerk, Mountpoints etc.
- Ressource Verwaltung über Containerruntime (VEE) / OS Methoden (cgroups)
- Packaging: „host-fs“ / Docker File

### VM:

- Höhere Isolation
- Eigene Kernel
- Ressource Verwaltung über VMM
- Packaging: AMI, ISO, VMDK, VDI,... (Virtual Disks)





# IaaS #3: Ausflug: Docker

- Single Prozess Philosophie
- Tooling / Verwaltung / API
- Packaging & „Image“ Management:
  - Read Only, keine „lokale“ Persistenz
  - In Layern „gestapelt“
  - „Bauanleitung“ -> Docker Files
  - Registry
- => Dadurch optimal für CI/CD Pipelines
- Konfiguration einfach (build time & run time)



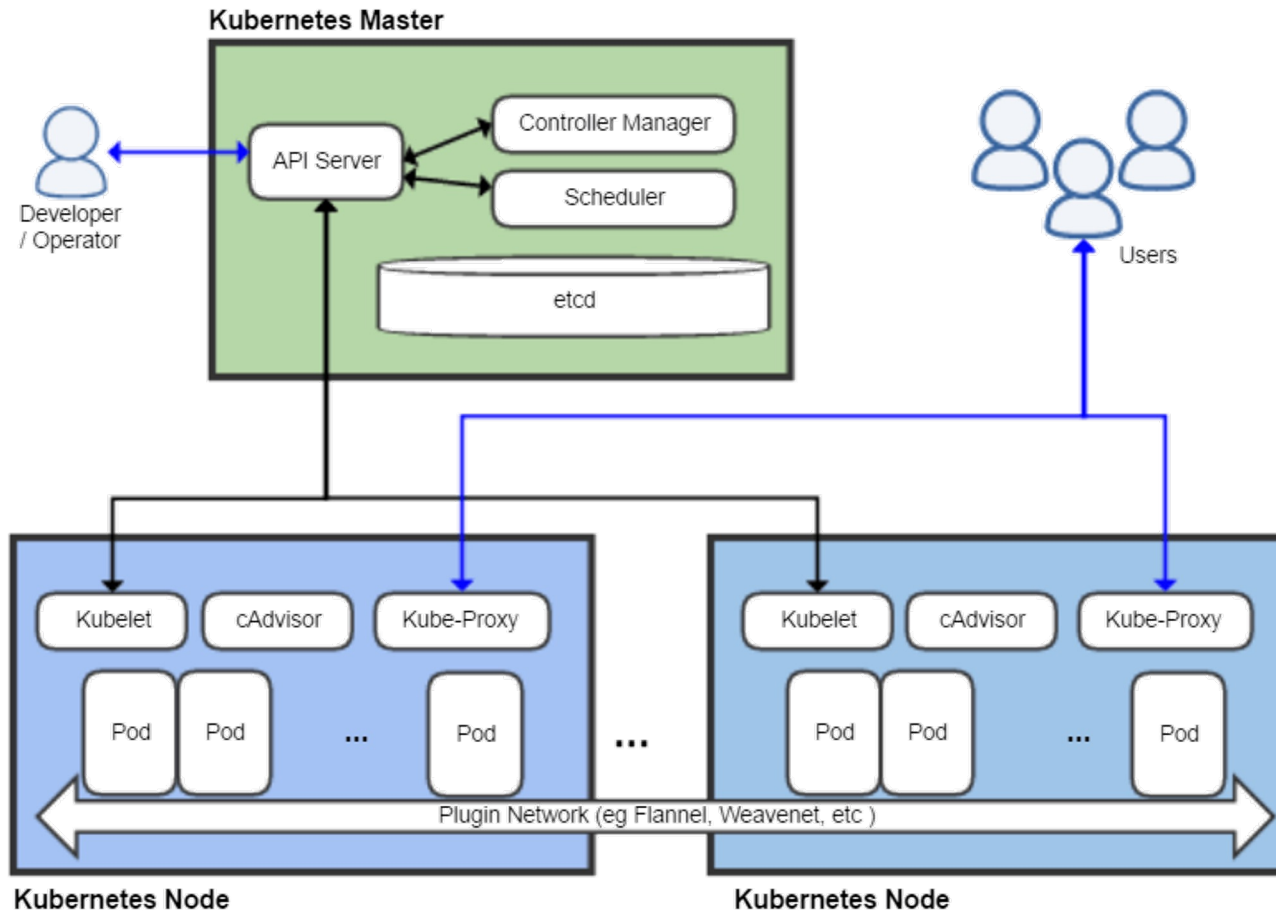


# IaaS #3: K8s – welche Probleme werden gelöst?

1 Container kommt selten allein  
1 Dockerinstanz ist keine „Cloud“

## kubernetes

- Container als Cluster – horizontal skalierbare Knoten (Minions)
  - Verfügbarkeit / Zustand & Ressourcen (CPU, Speicher) der Knoten
  - Konfiguration
  - Monitoring
- Orchestrierung
  - Abhängigkeiten zwischen verschiedenen Containern („Micro-“ Services)
  - Scheduling - Verteilung von Containern (Ressource Management) / Selbstheilung / Automatische Skalierung / Verfügbarkeit (Replicasets)
  - Definition von „Regeln“
- Services
  - Service Discovery
  - Netzwerk: Erreichbarkeit / Loadbalancing von Services
  - Persistent & Ephemeral Storage



Source: wikipedia



# IaaS #3: Pods

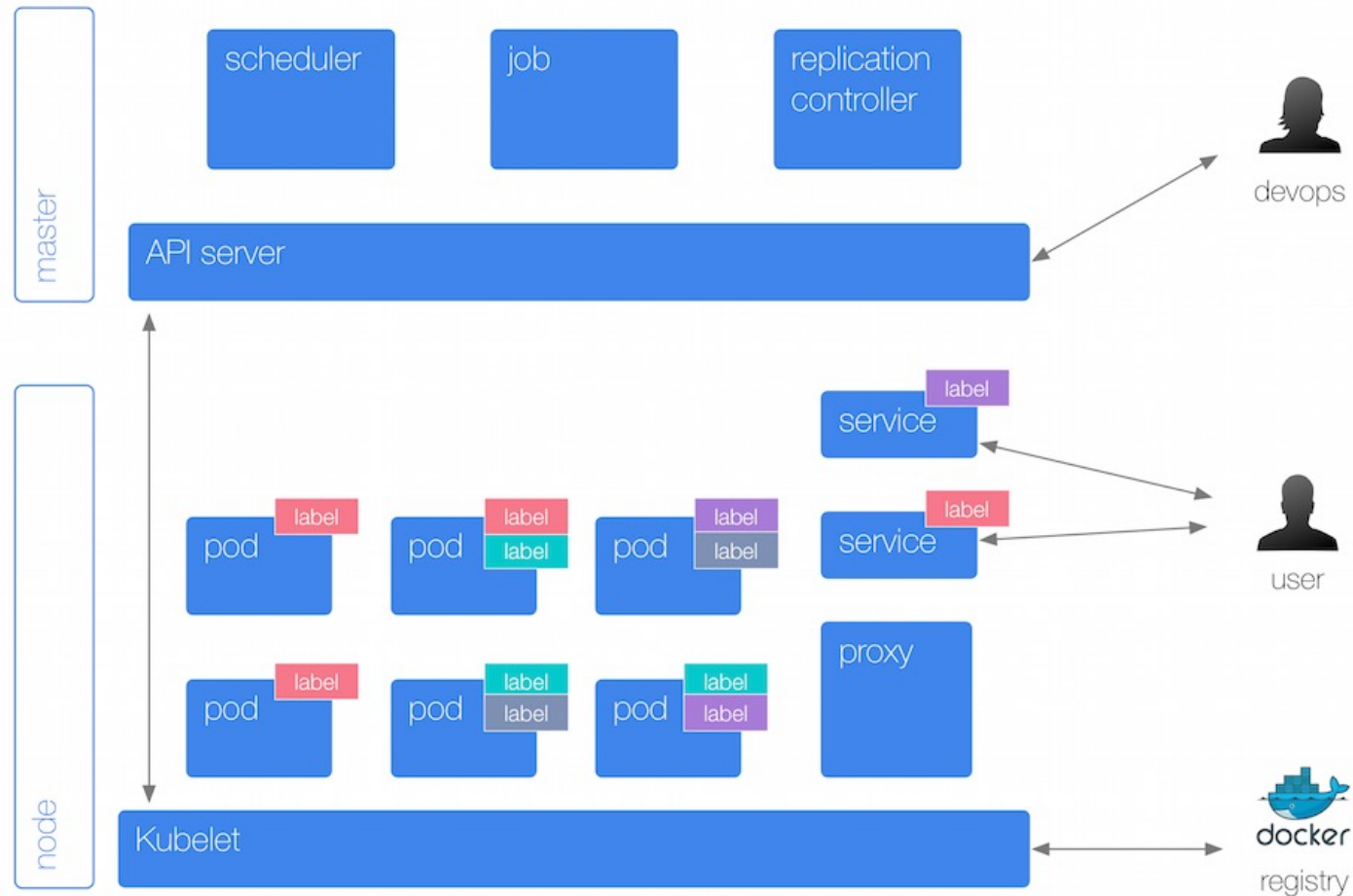


kubernetes

- Logische atomare Einheit in K8s
- Kann 1 oder mehrere Container enthalten
- Teilen sich:
  - Den gleichen Host (Node)
  - Networking (IP / Port Range)
  - Storage / Volumes
  - Lifecycle (start/stop/deletion/scaling)



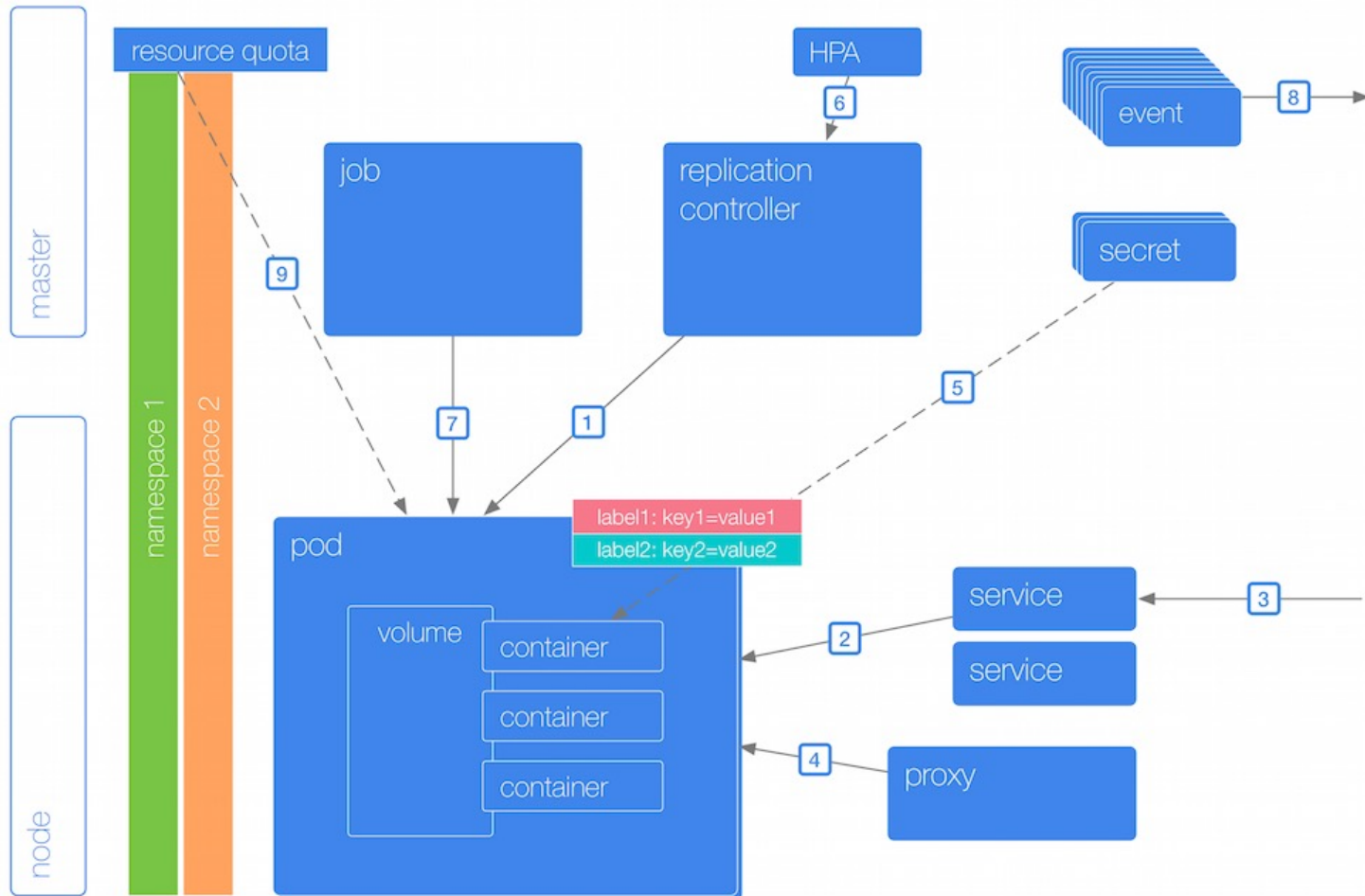
# IaaS #3: Kubernetes abstrakte Sicht



Source: <http://k8s.info/cs.html>



# IaaS #3: Kubernetes Details



Source: <http://k8s.info/cs.html>



# Zusammenfassung IaaS

- Typische Komponenten
  - Computing Einheiten
  - Storage Einheiten
- Diverse optionale Komponenten
  - z.B. Networking
  - Tendenz zu Mehrwertdiensten -> PaaS
- Scale-Out & Scale-Up Support
- Diverse Business- und SLA Aspekte
- Tendenz zu API Standards der Platzhirsche
- private, public & hybrid IaaS Clouds