universität wien

Computing Infrastructure

Chapter 2

VU Information Management & Systems Engineering

Wolfgang Klas

Planned Kahoot Sessions

Subject to change (see Moodle)

Typ	Date	Loc	Comment / Tentative Schedule	Kahoot
L	Fri 07.03. 13:15-14:45	HS 1	Kickoff + Organization	Kahoot-Test
A	Tue 11.03. 15:00-16:30	online	Assignment-Instruction Session	
L	Fri 14.03. 13:15-14:45	HS 1	Data Engineering 1	
L	Fri 21.03. 13:15-14:45	HS 1	Data Engineering 2	Kahoot
A	Tue 25.03. 15:00-16:30	online	Q/A Milestone 1	
L	Fri 28.03. 13:15-14:45	HS 1	Data Engineering 3	Kahoot
L	Fri 04.04. 13:15-14:45	HS 1	Data Engineering 4	Kahoot
L	Fri 11.04. 13:15-14:45	HS 1	Computing Infrastructure 1	Kahoot
S	Fri 11.04. 13:00	online	Milestone 1 (Submission Deadline)	
	Fri 18.04.		Easter break	
	Fri 25.04.		Easter break	
L	Fri 02.05. 13:15-14:45	HS 1	Computing Infrastructure 2	Kahoot
T	Fri 09.05. 13:15-14:45	tba.	Test 1	
A	Tue 13.05. 15:00-16:30	online	Docker Tutorial	
L	Fri 16.05. 13:15-14:45	HS 1	Security Engineering 1	Kahoot
A	Tue 20.05. 15:00-16:30	online	Q/A Milestone 2	
L	Fri 23.05. 13:15-14:45	HS 1	Security Engineering 2	Kahoot
L	Fri 30.05. 13:15-14:45	HS 1	- Reserve	
T	Fri 06.06. 13:15-14:45	tba.	Test 2	
T	Fri 13.06. 13:15-14:45	tba.	- Reserve (Test)	
S	Mon 16.06. 13:00		Milestone 2 (Submission Deadline)	
A	from 17.06.	online	Final Presentations	

Computing Infrastructure

Contents

- Vision: Utility Computing
- Computational Models
- Virtualization
  - Types and Techniques of Virtualization
  - DevOps and Docker
- Cloud Computing
  - Characteristics, Service- and Deployment-Model
  - To use or not to use

The Vision: Utility Computing

- An old Computing Vision in Science Fiction Literature
- “The Last Question” by Isaac Asimov, **1956**
  https://xpressenglish.com/the-last-question/
  - *"The Last Question" is a thought-provoking science fiction short story by Isaac Asimov, first published in 1956. The story explores themes of entropy, the quest for knowledge, and the fate of the universe. It raises deep questions about the nature of existence and the role of technology in our lives. Over time, various advanced computers (Multivac, Microvac, Galactic AC, Cosmic AC [totally in hyperspace and is said to be made of something that is neither matter nor energy], …) are asked how to reverse entropy and prevent the universe's end. After the universe's demise, the final computer discovers the solution and creates a new universe. [Summary by GPT]*
  - *“He stared somberly at **his small AC-contact**: It was only **two inches cubed** and nothing in itself, but it was **connected through hyperspace with the great Galactic AC** that served all mankind”*

The Vision: Utility Computing

IT resources as Utility

- *"Computing may someday be organized as a public utility just as the telephone system is a public utility."*
  - John McCarthy at MIT's centennial celebration in 1961
  - Then, first approaches by **Virtualization**: memory, disk, CPUs

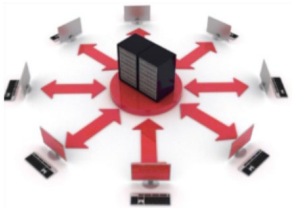
- Today: **Cloud Computing** and **Containerization**

Computational Models

1970's – 1980's

Mainframe Era

- Monolithic
- Powerful computers used mainly by large organizations for critical applications
- Centralized computing
- No communication with outside
- Time-sharing



1990's – 2000's

Client – Server Era

- There is more than one computer in the world
- Common hardware becomes more powerful
- Distributed computing
- Web

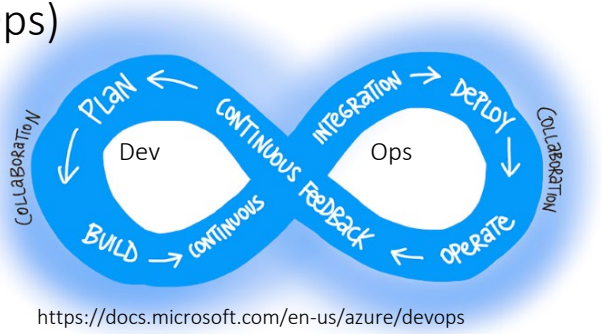
2010 +

Cloud Era

- Increased network bandwidth and reliability
- No need to invest large amounts of money in hardware
- Large data centers
- Virtualized Resources

DevOps and Virtualization

- **DevOps** is a software development methodology that combines and automates software **Development** (Dev) and information technology **Operations** (Ops)
  - To shorten the systems development life cycle
  - Delivering features, fixes, and updates frequently
  - In close alignment with business objectives
- (One) **Tool** to achieve DevOps: Virtualization
- **Virtualization** refers to the act of creating a virtual (rather than actual) version of something, including virtual computer hardware platforms, storage devices, and computer network resources
- Many **different forms** of virtualization, as Hardware, Desktop, …
- Culmination in new Computing Paradigm: **Cloud Computing**

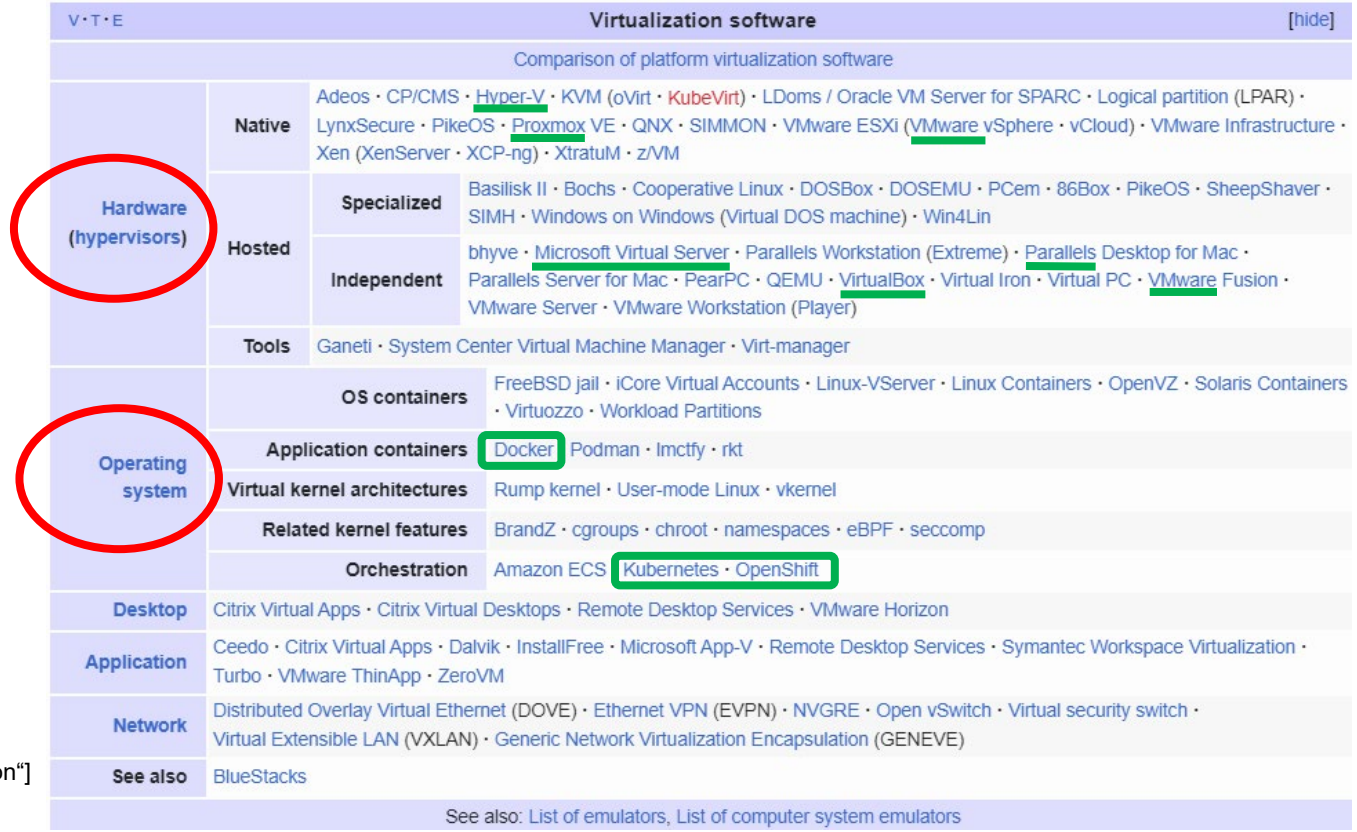


https://docs.microsoft.com/en-us/azure/devops

(Some) Forms of Virtualization

- **Hardware virtualization** or platform virtualization refers to the creation of a virtual machine that acts like a real computer with an operating system
- **OS-level virtualization**, also known as **containerization**, refers to an operating system paradigm in which the kernel allows the existence of **multiple isolated user-space instances**
- **Desktop virtualization** separates the **logical desktop** from the physical machine (e.g., game streaming Stadia, Remote Desktop)
- **Application Virtualization** (also known as Process Virtualization) is software technology that **encapsulates computer programs** from the underlying operating system on which it is executed (e.g. sandboxing)
- **Network virtualization** is the process of combining hardware and software network resources and network functionality into a single, software-based administrative entity (e.g. overlay virtual network)

Techniques for Virtualization

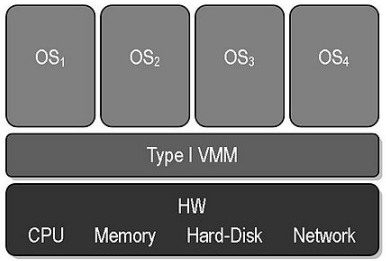


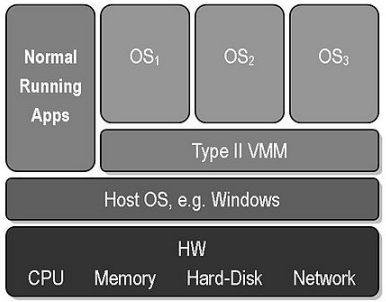
V·T·E	Virtualization software		[hide]
Comparison of platform virtualization software			
Hardware (hypervisors)	Native	Adeos · CP/CMS · Hyper-V · KVM (oVirt · KubeVirt) · LDoms / Oracle VM Server for SPARC · Logical partition (LPAR) · LynxSecure · PikeOS · Proxmox VE · QNX · SIMMON · VMware ESXi (VMware vSphere · vCloud) · VMware Infrastructure · Xen (XenServer · XCP-ng) · XtratuM · z/VM	
	Hosted	Specialized	Basilisk II · Bochs · Cooperative Linux · DOSBox · DOSEMU · PCem · 86Box · PikeOS · SheepShaver · SIMH · Windows on Windows (Virtual DOS machine) · Win4Lin
		Independent	bhyve · Microsoft Virtual Server · Parallels Workstation (Extreme) · Parallels Desktop for Mac · Parallels Server for Mac · PearPC · QEMU · VirtualBox · Virtual Iron · Virtual PC · VMware Fusion · VMware Server · VMware Workstation (Player)
	Tools		Ganeti · System Center Virtual Machine Manager · Virt-manager
Operating system	OS containers		FreeBSD jail · iCore Virtual Accounts · Linux-VServer · Linux Containers · OpenVZ · Solaris Containers · Virtuozzo · Workload Partitions
	Application containers		Docker · Podman · lmctfy · rkt
	Virtual kernel architectures		Rump kernel · User-mode Linux · vkernel
	Related kernel features		BrandZ · cgroups · chroot · namespaces · eBPF · seccomp
	Orchestration		Amazon ECS · Kubernetes · OpenShift
Desktop			Citrix Virtual Apps · Citrix Virtual Desktops · Remote Desktop Services · VMware Horizon
Application			Ceedo · Citrix Virtual Apps · Dalvik · InstallFree · Microsoft App-V · Remote Desktop Services · Symantec Workspace Virtualization · Turbo · VMware ThinApp · ZeroVM
Network			Distributed Overlay Virtual Ethernet (DOVE) · Ethernet VPN (EVPN) · NVGRE · Open vSwitch · Virtual security switch · Virtual Extensible LAN (VXLAN) · Generic Network Virtualization Encapsulation (GENEVE)
See also			BlueStacks
See also: List of emulators, List of computer system emulators			

[Wikipedia „Virtualization“]

Hardware Virtualization

- **Virtual machine**, that acts like a real computer with operating system
- **Hypervisor** necessary, i.e., computer software, firmware or hardware that creates and runs virtual machines
  - **Type I: bare-metal or native hypervisor** that are deployed directly over the host's system hardware without any underlying operating systems or software, e.g., **Microsoft Hyper-V hypervisor**, VMware ESXi, IBM z/VM, KVM, Xen
    - e.g., **Xen** used by Amazon EC2, Rackspace Cloud …
  - **Type II: hosted hypervisor** that runs as a software layer within a physical operating system, e.g. **Oracle Virtualbox**, Parallels Desktop, VMware Player
  - Some solutions blur these boundaries



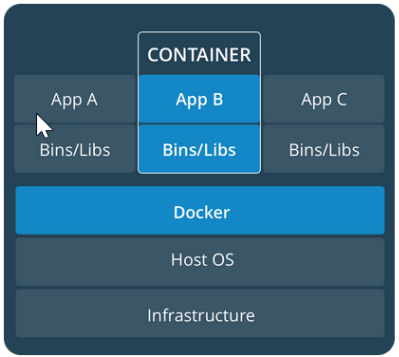


OS-level Virtualization

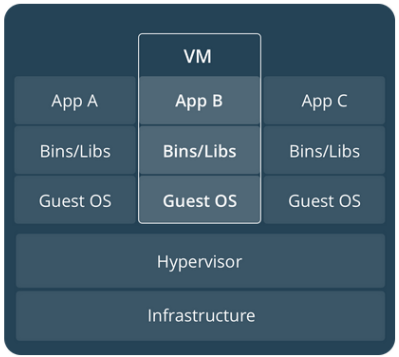
- Host OS-kernel allows **multiple isolated user-space instances**
  - Such instances, called **containers** (zones, jails, etc.), may look like real computers from the point of view of programs running in them
- A **container** is a runtime instance of an executable package that includes everything needed to run an application
  - As code, runtime libraries, environment variables, configuration file
- Comparison

Container is lightweight
Example : Docker

OS-level virtualization



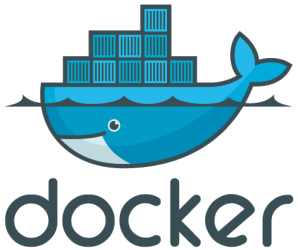
Hardware virtualization



A virtual machine (**VM**) runs a **full-blown "guest" operating system** with virtual access to host resources through a hypervisor

Docker: It's all about shipping!

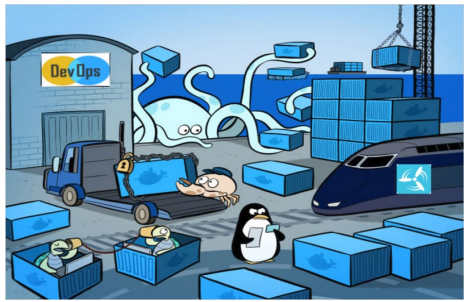
- **Docker** is a platform for Developers and Operators (Sysadmins) to **automate the**
  - **development,**
  - **deployment,** and
  - **running of applications**
- with containers: **Tool for DevOps**



- **Containers leverage** the **low-level mechanics** of the host operating system, to **provide most of the isolation of virtual machines** at a fraction of the computing power
  - Implements a high-level API to provide lightweight containers that run processes in isolation
- Developer: "Build once … (finally) run anywhere"
- Operator: "Configure once … run anything"

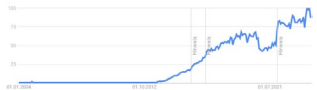
Image and Container

- An **Image** is an executable package that includes **everything** needed to run an application
  - Code, runtime libraries, environment variables, and configuration files
- A **Container** is what the image becomes in memory when executed (that is, an image with state, or a user process)
- Containerization is increasingly popular because containers are:
  - **Flexible**: Even the most complex applications can be containerized
  - **Lightweight**: Containers leverage and share the host kernel
  - **Interchangeable**: You can deploy updates and upgrades on-the-fly
  - **Portable**: You can build locally, deploy to the cloud, and run anywhere
  - **Scalable**: You can increase and automatically distribute container replicas
  - **Stackable**: You can stack services vertically and on-the-fly

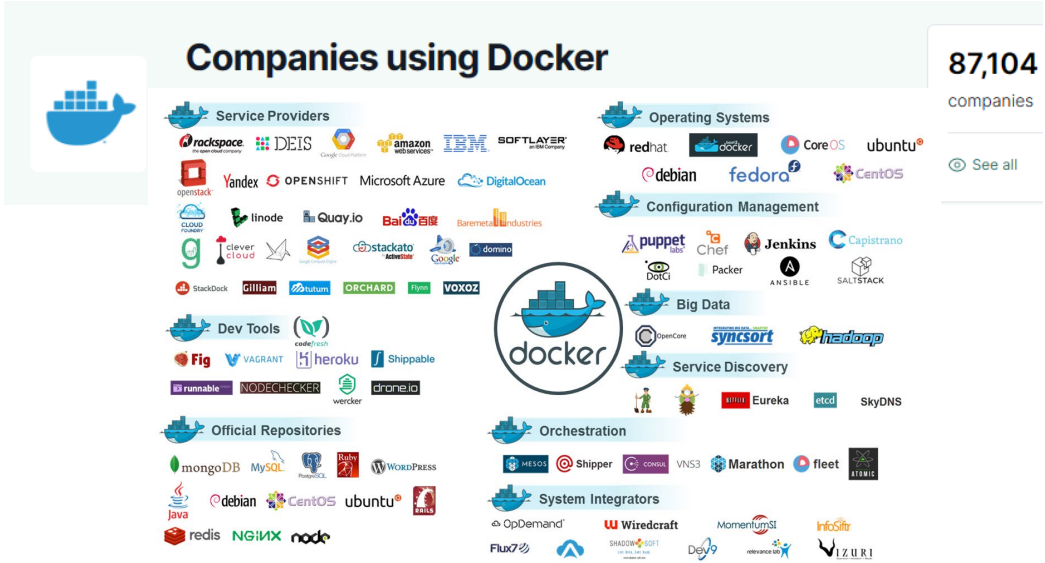


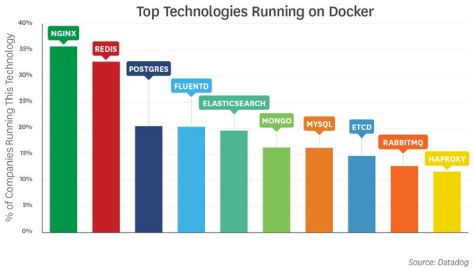
Is it popular?

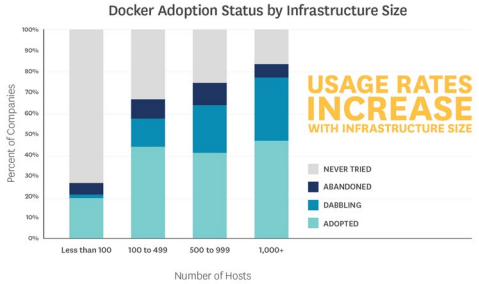
- Well-known commercial providers
  - Amazon Web Services
  - Microsoft Azure
  - Digital Ocean
  - Google Compute Engine
  - OpenStack
  - Rackspace
  - … (many, many more)
- Increased attention since 2013/2014



Google Trends







Virtualization as Paradigm: Cloud Computing

- Starting point "Distributed Systems"
  - „A distributed System is one in which (hardware or software) components located at networked computers communicate and coordinate their actions only by passing messages“
- "The largest supercomputer of the world is the Internet"
  - Pool of resources: CPUs (work power), Disks (storage), Networks (infrastructures), Knowledge (competence), …
- Different computing paradigms in the last few years
  - Distributed and Parallel Computing (HPC, Supercomputer, …)
  - Cluster
  - Web computing, Internet of Services and Things
  - Grid Computing
  - Cloud Computing

Cloud Computing Hype?

- *The most important significant change in the IT priorities of ESG's 2011 survey is the increased importance attached to cloud computing services*. - ESG Research Thomson Reuters
- IT spending on cloud computing (Gartner)

Worldwide Public Cloud Service Revenue Forecast (Billions of U.S. Dollars)	2017	2018	2019	2020	2021
Cloud Business Process Services (BPaaS)	42.6	46.4	50.1	54.1	58.4
Cloud Application Infrastructure Services (PaaS)	11.9	15.0	18.6	22.7	27.3
Cloud Application Services (SaaS)	60.2	73.6	87.2	101.9	117.1
Cloud Management and Security Services	8.7	10.5	12.3	14.1	16.1
Cloud System Infrastructure Services (IaaS)	30.0	40.8	52.9	67.4	83.5
Total Market	**153.5**	**186.4**	**221.1**	**260.2**	**302.5**

- Estimate for 2023 was: 591 Billions US$
  - https://siliconangle.com/2022/10/31/gartner-spending-public-cloud-services-will-exceed-591b-2023/
- Believe it or not, you are using cloud computing already for a long time!

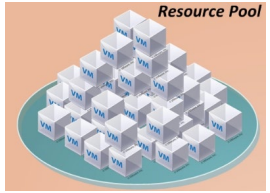
Definition Cloud Computing

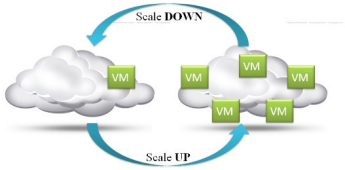
*"Cloud computing is a model for enabling convenient, on-demand network access to a **shared pool of configurable computing resources** (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. This cloud model promotes availability and is composed of **five essential characteristics**, **three service models**, and **four deployment models**."*

- From the National Institute of Standards and Technology: http://csrc.nist.gov/groups/SNS/cloud-computing/index.html

Five Characteristics

- Resources Pooling
  - Resources are shared by multiple tenants
  - Sense of location independence
- Rapid elasticity
  - Resources can scale Up/Down
  - Any quantity at any time
- On-demand Self-service
  - Unilaterally (self) provision of computing capabilities as needed
  - Within mere minutes or seconds
- Broad Network Access
  - Access from anywhere
  - Standard mechanisms
- Measured Service
  - Pay-as-you-go Service
  - Metering capability





Software Platform Infrastructure (SPI) Service Models

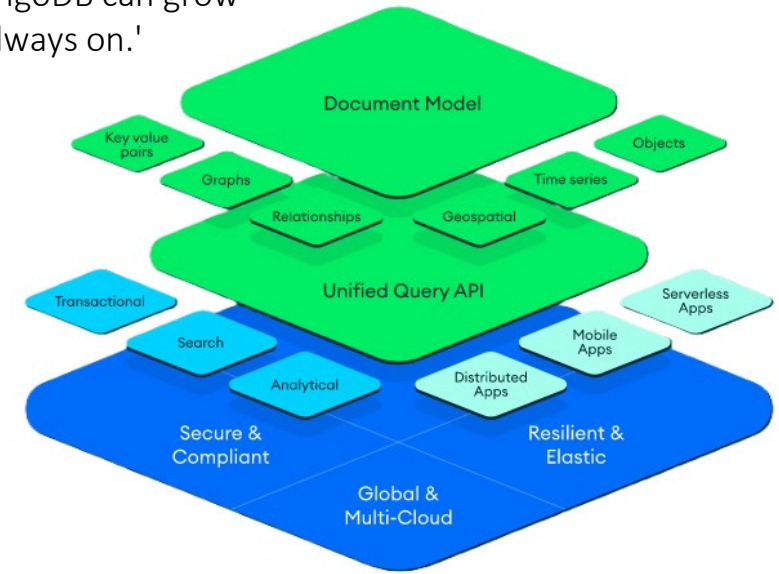
- Infrastructure as a Service (IaaS)
  - Provision of computer infrastructure as a service
    - Hardware, as cores, disks, network, OSs
- Platform as a Service (PaaS)
  - Solution stack as a service
  - Provides computational resources via a platform upon which application and services can be developed and hosted
- Software as a Service (SaaS)
  - Model of software deployment whereby a provider licenses an application to consumer for use as a service on demand
    - Google Docs, GMail, Salesforce CRM
    - Microsoft 365, OneDrive, OneNote, Teams, Word, …

Service Model: Cloud Database

- A **Cloud Database** is a database that typically runs on a cloud computing platform, access to it is provided as a service
  - Virtual machine Image: Users execute their own machine image with a database installed
  - Database Container
  - DBaaS
- **Database as a Service** (DBaaS)
  - Application owners do not have to maintain the database themselves
  - Database service provider takes responsibility for installing and maintaining the database system and database
  - Application owners are charged according to their usage of the service
- Examples:
  - MongoDB Atlas: MongoDB as a service
    - https://www.mongodb.com/cloud

MongoDB Atlas Benefits

- Cloud Database: flexible, affordable, and scalable database management.
- DBaaS (Database-as-a-Service): fully managed cloud platform; no installation needed.
- Scalability: As an application grows in size and complexity, MongoDB can grow
- 24/7 Availability: built-in backup and recovery, ensuring it is 'always on.'
- Handling Large Datasets
- Flexible Document Schemas
- Powerful Querying and Analytics
- Multi-cloud Deployment
- Cost Efficiency
- Security
- Automated Backup and Recovery
- Data Visualization Tool
- Easy Installation and Integration
- Data API



Examples: Databases as Amazon Web Services

Database type	Examples	AWS service
Relational	Traditional applications, enterprise resource planning (ERP), customer relationship management (CRM), ecommerce	Amazon Aurora, Amazon RDS, Amazon Redshift
Key-value	High-traffic web applications, ecommerce systems, gaming applications	Amazon DynamoDB
In-memory	Caching, session management, gaming leaderboards, geospatial applications	Amazon ElastiCache, Amazon MemoryDB for Redis
Document	Content management, catalogs, user profiles	Amazon DocumentDB (with MongoDB compatibility)
Wide column	High-scale industrial apps for equipment maintenance, fleet management, and route optimization	Amazon Keyspaces
Graph	Fraud detection, social networking, recommendation engines	Amazon Neptune
Time series	Internet of Things (IoT) applications, DevOps, industrial telemetry	Amazon Timestream
Ledger	Systems of record, supply chain, registrations, banking transactions	Amazon Ledger Database Services (QLDB)

Examples: Databases as Microsoft Azure Services

Microsoft Azure

Azure SQL

Migrate, modernize, and innovate on the modern SQL family of cloud databases

Azure Cosmos DB

Build or modernize scalable, high-performance apps

Azure SQL Database

Build apps that scale with managed and intelligent SQL database in the cloud

Azure Database for PostgreSQL

Fully managed, intelligent, and scalable PostgreSQL

Azure SQL Managed Instance

Modernize SQL Server applications with a managed, always-up-to-date SQL instance in the cloud

Azure Database for MySQL

Fully managed, scalable MySQL Database

SQL Server on Azure Virtual Machines

Migrate SQL Server workloads to the cloud at lower total cost of ownership (TCO)

Azure Cache for Redis

Accelerate apps with high-throughput, low-latency data caching

Azure Database Migration Service (classic)

Accelerate your data migration to Azure

Azure Managed Instance for Apache Cassandra

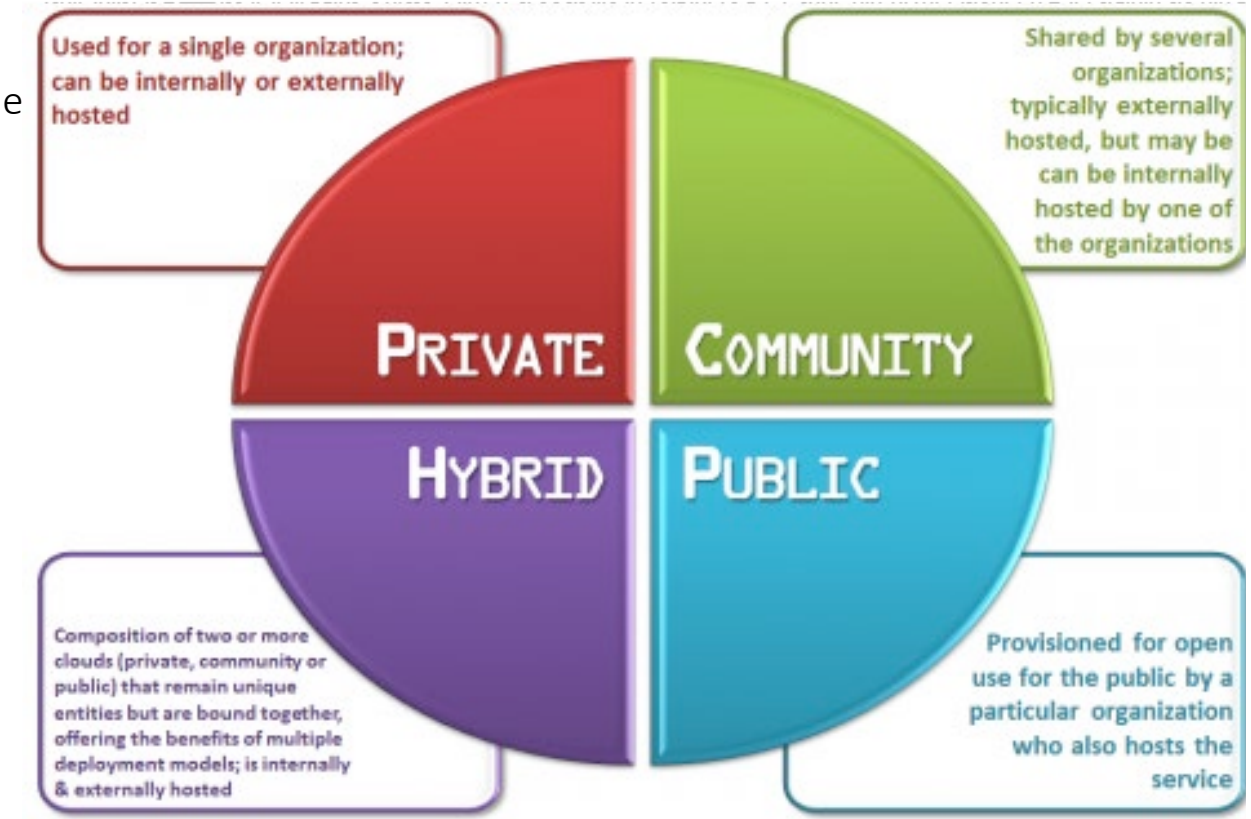
Modernize Cassandra data clusters with a managed instance in the cloud

Azure Database for MariaDB

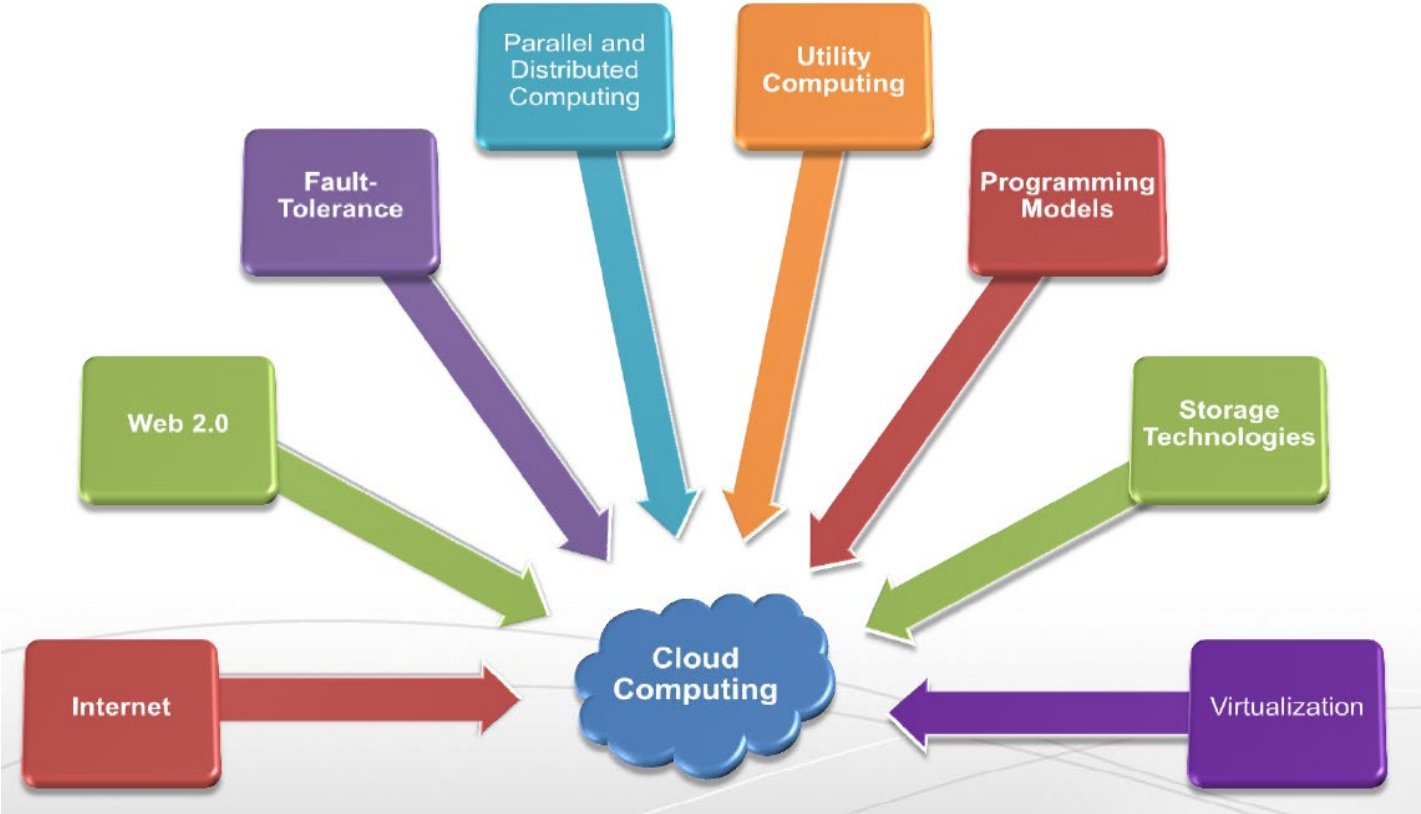
Deploy applications to the cloud with enterprise-ready, fully managed community MariaDB

Four Deployment Models

- **Public Cloud**
  - Cloud solution used by anyone
- **Private Cloud**
  - Cloud solution owned by the organization running the cloud
- **Hybrid Cloud**
  - Creating solutions based on more than one of the models above
- **Community Cloud**
  - Cloud solution used by a group of organizations



Many Influencing IT Aspects

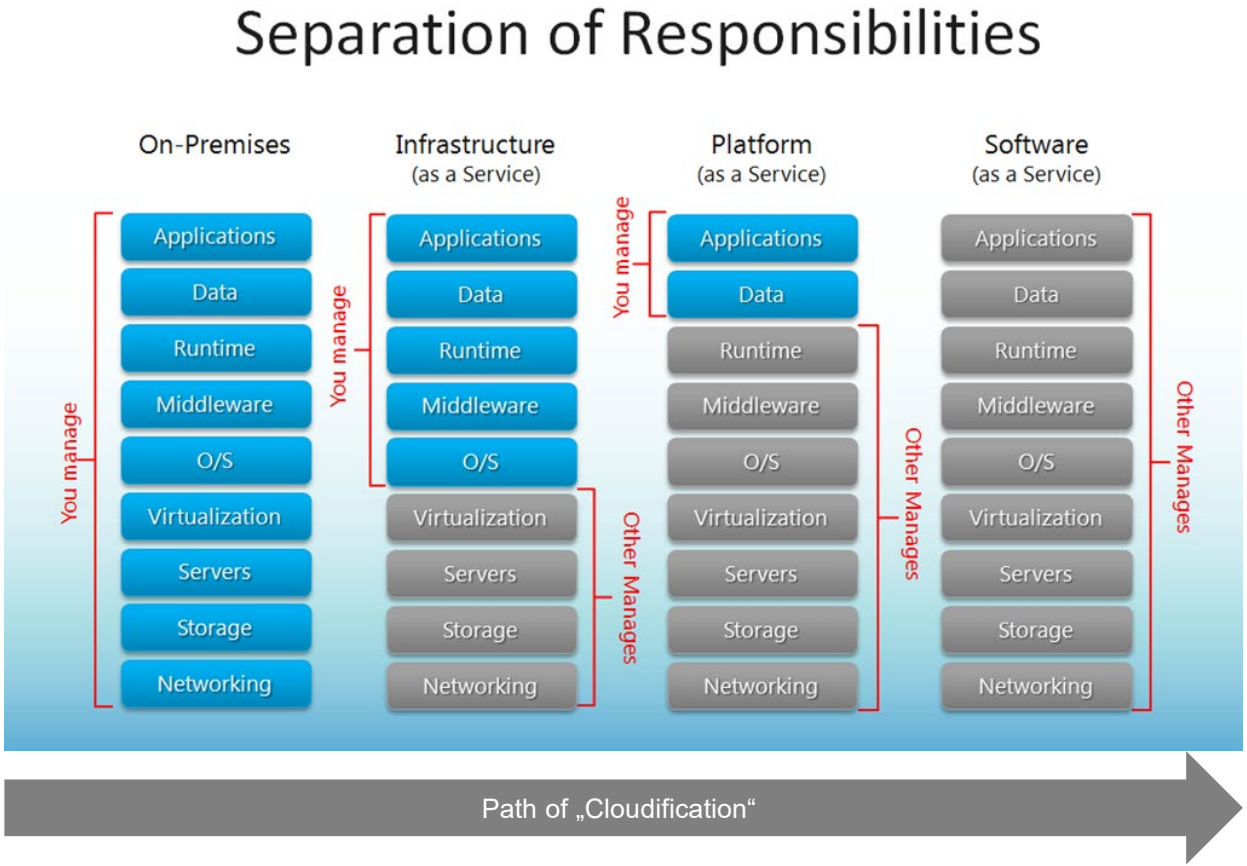


Summarizing Promises of the Cloud

Clouds for Startup/SME, for Rapid Prototyping

- Infrastructure- and platform-services very convenient for **Startups** or **SMEs and rapid prototyping of solutions**
- Advantage of the **Pay-per-use** cost model
- Allows to roll out business model with **limited financial effort**
  - No costly SW and HW necessary
  - high initial investments often lost, if idea does not fly
- Allows SME to **focus on core business**
  - Computer center services are „out-sourced“
- What about **out-sourcing of data?**
  - Often refused due to security, privacy and legal issues
  - Company and customer data

Cloud „Outsourcing“ path of companies



Changes for the user / developer

not easy

- Moving applications to the cloud is **non-trivial**
- Needs often **large internal changes**
  - Change of APIs or user interfaces
  - Cloud services are continually developed
    - „Perpetual Beta“
- User/developer must react to **changes rapidly**
  - Outage of cloud services
  - Even cancellation of services
    - Mitigation plan necessary
  - Contracts have short change cycles
    - About 30 days, different to classical out-sourcing
- **Cost monitoring** necessary by user
  - Different cost model
    - Pay-per-use, flatrate, …
- Cloud Services need **Internet**
  - What happens if connection loss?

Motivation for Cloud Computing in the future

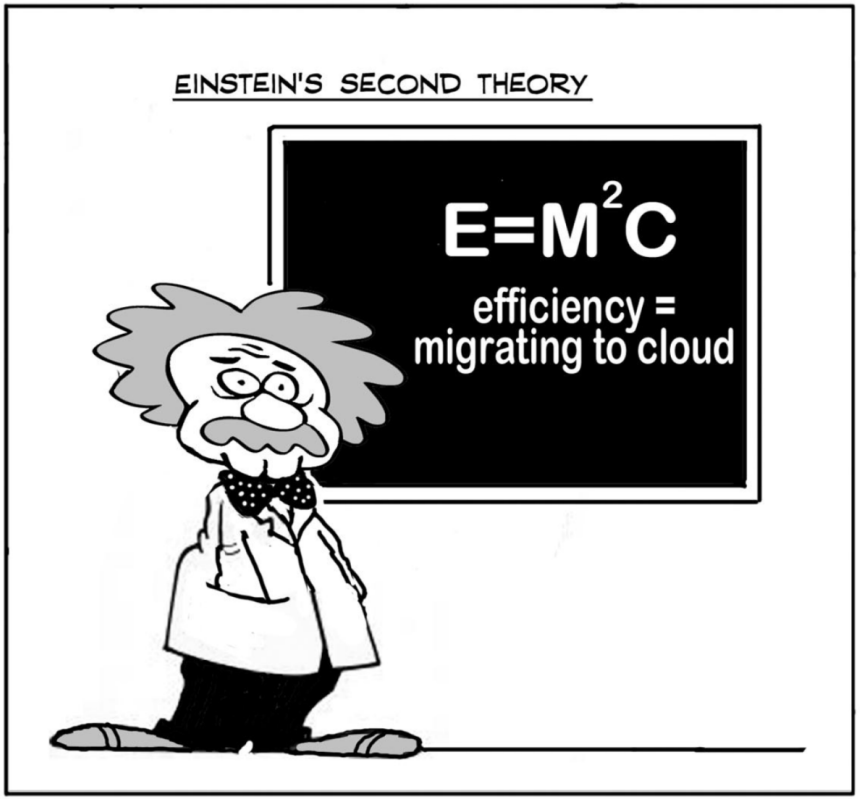
- Mobile interactive applications
  - Increased use of tablets and smartphones with Apps, in real-time accessible from everywhere
  - Parallel batch processing
  - Parallelization of tasks by a pay-per-use model
    - better 100 virtual CPUs for 1 hour
      than 100h of a single CPU
- Number Crunching and Data analysis (Big Data)
  - Processing of huge data sets
  - Business Intelligence in real time
- Internet of Things (IOT)
  - Support of low performant devices by power from the cloud
  - Thus, increase of energy/battery performance

No Cloud!

- Systems with **low latency requirements** can not be easily transferred to the cloud
  - Latency is the delay from input into a system to desired outcome, e.g. network latency (Internet latency), interrupt latency, OS latency, …
    - Examples
      - Electronic stock market
      - Real-time systems
      - Life critical systems
      - Power plants or systems with emergency shutdown

- Systems with specific **security and/or privacy needs**
  - However, work on cloud security issues ongoing (many international projects)
  - **BUT**: there is motivation for Cloud usage, e.g. EU-Datenschutz-Grundverordnung (DSGVO, General Data Protection Regulation) in place since May 25th, 2018

There is no way around Clouds!



http://www.cloudtweaks.com