Cloud Computing

Deployment Models and Service Models in Cloud Computing

Slide set 3

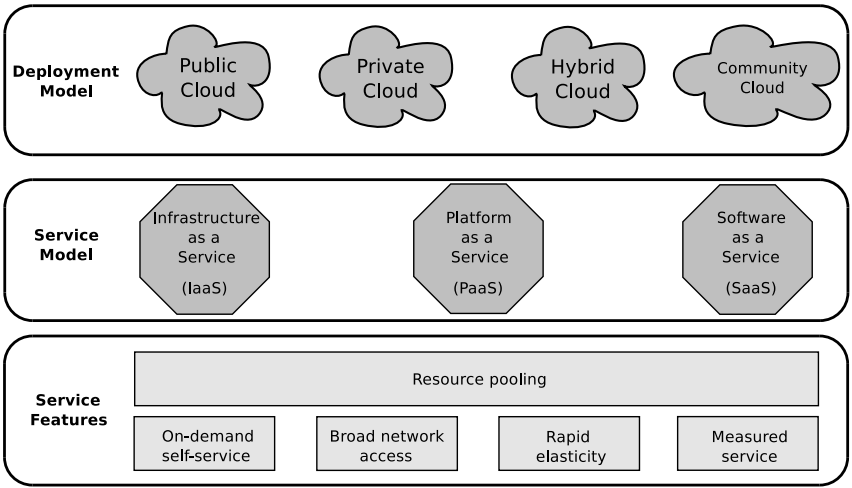
Henry-Norbert Cocos
cocos@fb2.fra-uas.de

Computer Science
Department of Computer Science and Engineering
Frankfurt University of Applied Sciences

Agenda

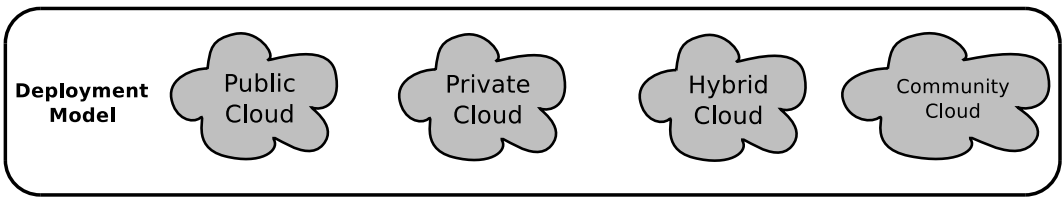
FRANKFURT UNIVERSITY OF APPLIED SCIENCES

NIST definition of Cloud Computing



FRANKFURT UNIVERSITY OF APPLIED SCIENCES

Deployment Models in Cloud Computing



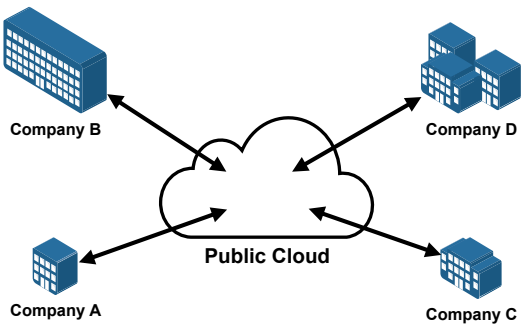
Deployment Models in Cloud Computing

Definition of deployment models

Cloud services are usually divided into **private and public cloud** models. The **hybrid** and **community** models that also exist are often less present in the public debate, presumably because they can hardly play to their strengths in service computing.

FRANKFURT UNIVERSITY OF APPLIED SCIENCES

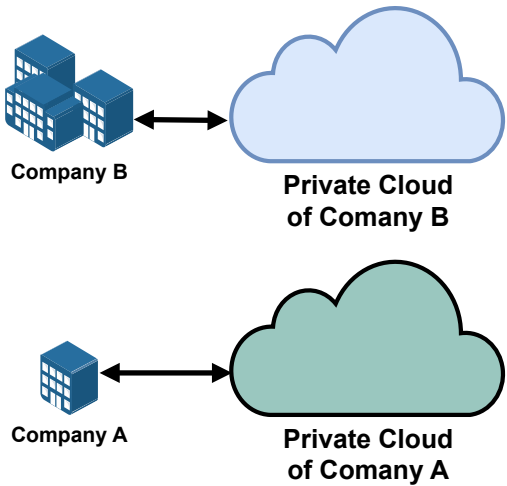
Public Cloud



A public cloud is a cloud infrastructure for open use by the general public. It can be **owned, managed and operated by a business**, academic or government organization, or a combination thereof. It is located on the **cloud provider's premises** (i.e. off-premise for cloud users).

FRANKFURT UNIVERSITY OF APPLIED SCIENCES

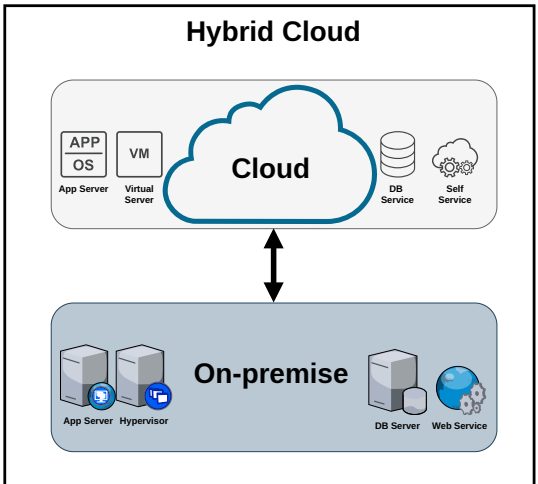
Private Cloud



A private cloud, on the other hand, is a cloud infrastructure that **is operated for the exclusive use of a single organization** with multiple consumers (e.g. business units). It can be **owned by the organization, a third party or a combination of both**. It is irrelevant whether the infrastructure is located on the organization's premises (i.e. on-premise for the cloud users) or not.

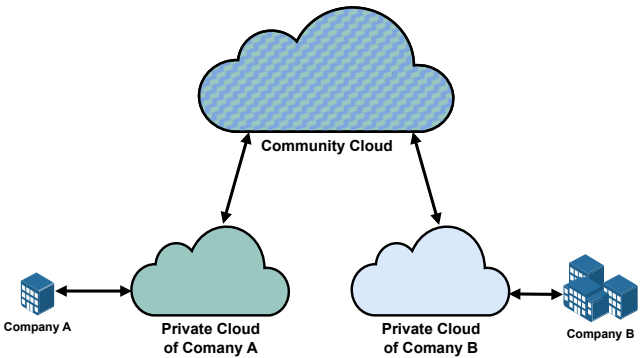
FRANKFURT UNIVERSITY OF APPLIED SCIENCES

Hybrid Cloud



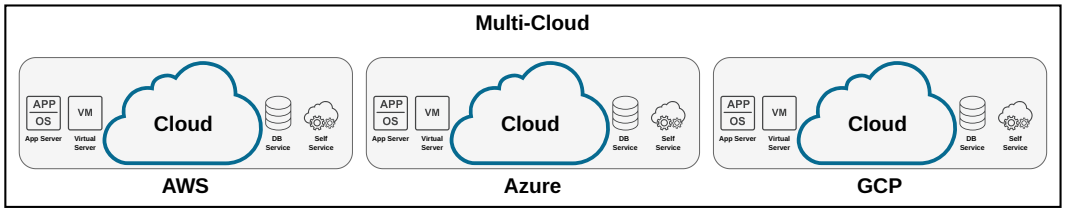
A hybrid cloud is a cloud infrastructure that forms a composition of **two or more** of the above-mentioned **cloud infrastructure forms (private, public, community)**. These remain independent units, but are connected to each other using standardized or proprietary technology that enables the portability of data and applications (e.g. cloud bursting for load balancing between cloud infrastructures).

Community Cloud



The lesser-known form of community cloud refers to a cloud infrastructure that is **operated for the exclusive use of a specific community of consumers** from organizations. This community usually has common concerns (e.g. mission, security requirements, guidelines and compliance considerations). It may be owned, managed and operated by one or more organizations in the community, a third party or a combination of them. Community clouds can be operated both on-premise and off-premise.

Multi Cloud



A multi-cloud involves **using multiple cloud services from different providers** to meet various organizational needs. Unlike a single-cloud approach, which relies on one cloud service provider, a multi-cloud leverages various providers' unique strengths and capabilities to **optimize performance, cost, and resilience**. This approach offers flexibility, reduces dependency on a single vendor, and can enhance the overall effectiveness of cloud operations. ($\Rightarrow$ More in slide set 4)

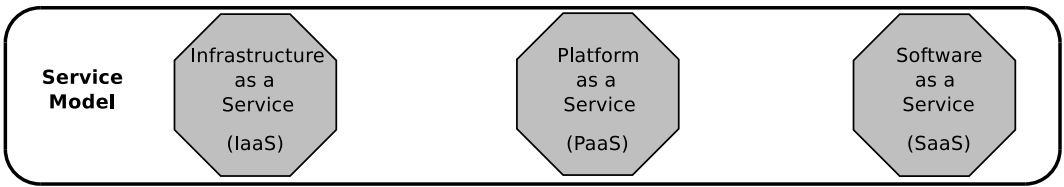
Notes on Public Cloud

Notes on public clouds

Although public cloud computing can be very beneficial in many cases, there are also use cases that are considered problematic and where it is difficult to take advantage of the public cloud deployment model, as the following examples show:

- **Critical infrastructures:** In areas such as energy supply, healthcare or public safety, critical infrastructures are operated whose failure can have serious consequences.
- **Data protection and compliance:** Companies that process personal data (GDPR) in particular must ensure that their data is secure and meets the applicable data protection and compliance requirements.
- **Costs:** Although public clouds can in many cases be more cost-effective than providing and managing your own IT infrastructure, there are also use cases in which using the public cloud can be uneconomical. This is particularly true if the application requires high requirements in terms of performance, storage space or bandwidth.

Service Models in Cloud Computing



Service Models in Cloud Computing

Service Models

Cloud computing can be used to outsource parts of the IT-based value chain to external cloud service providers (CSP). The scope of outsourcing is often divided into the categories of **Infrastructure as a Service (IaaS)**, **Platform as a Service (PaaS)** and **Software as a Service (SaaS)**.

Level of abstraction and risks

The extent of outsourcing also increases the potential dependency **(vendor lock-in)** of a customer to a cloud provider also increases. A lock-in effect generally refers to close customer dependency to a CSP's products/services in the form of a technical-functional customer dependency. This makes it difficult for customers to switch from one provider to the service of another provider. In cloud computing, this effect is usually caused by non-standardized cloud service APIs of the individual providers.

Service Models in Cloud Computing. . . There is even more!!!

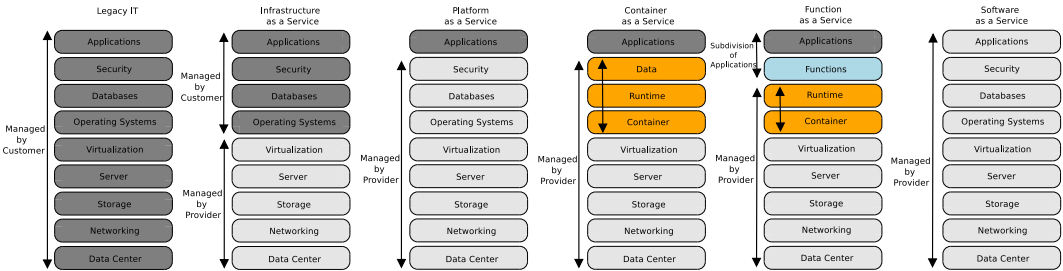


Figure: Service Models in Cloud Computing

Infrastructure as a Service

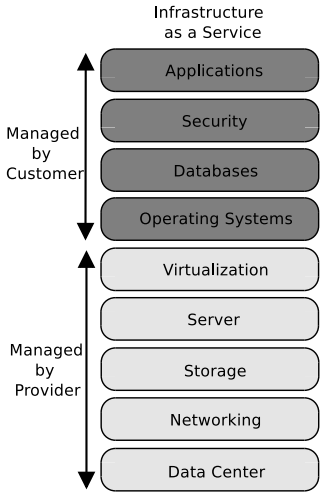


Figure: Infrastructure as a Service

Types of resources:

- **Compute**
  - CPU, GPU and RAM
  - Needed for the execution of arbitrary tasks
- **Storage**
  - Block-level ⇒ stores data in blocks on SSD or hard disk.
  - File-level ⇒ stores data in files in a file system.
  - Object-storage ⇒ stores data as unstructured objects.
- **Networking**
  - IaaS also includes network resources such as routers, switches and load balancers.

Infrastructure as a Service – characteristics

IaaS Characteristics

In the IaaS model, a provider offers physical and virtual hardware such as **servers, storage**, and **network infrastructure** that can be quickly provisioned and decommissioned via a self-service interface. This makes it possible, for example, to provide IT resources flexibly and, above all, load-driven as part of periodic workloads with recurring peak loads.

IaaS resources

Resources for the customer consists of the fast and elastic provision of **processing, storage, network**, and other **basic computing resources** on which the customer can deploy and run any software, including operating systems and applications. While the customer does not manage or control the underlying cloud infrastructure, the customers control operating systems, storage, and deployed applications.

Infrastructure as a Service – elastic infrastructures

Elastic infrastructures

The associated service offering are an **elastic infrastructure** to provide **virtual servers**, **persistent storage**, and **network connectivity**. An elastic infrastructure usually offers **preconfigured** virtual server images, persistent storage, and network connectivity that customers can request via a self-service interface. The provider also provides load and use data to inform resource utilization, which is required for traceable billing and automation of management tasks.

Platform as a Service

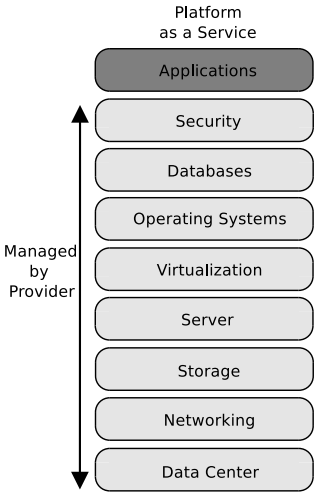


Figure: Platform as a Service

PaaS

- The PaaS platform provides runtime environment and/or development environment for applications.
- Hosting of (often) web applications.
- The underlying hardware and software is provided by the platform provider.
- Support for essential parts of the software lifecycle from development, testing and delivery through the operation of applications via the internet.

Platform as a Service

Deployment of Applications

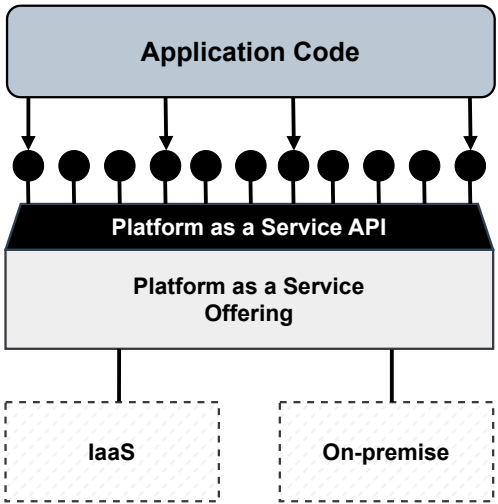
PaaS services **offer an ad-hoc development and operating platform** for the entire life cycle of applications. Applications are deployed as an **application package or as source code**. In most cases, no image is required for a technical infrastructure.

APIs of platform

The application only sees programming or access interfaces of a PaaS-specific runtime environment. The PaaS runtime environment can also ensure **automatic scaling of the application** by means of automatic provisioning of the infrastructure.

FRANKFURT UNIVERSITY OF APPLIED SCIENCES

Platform as a Service – APIs

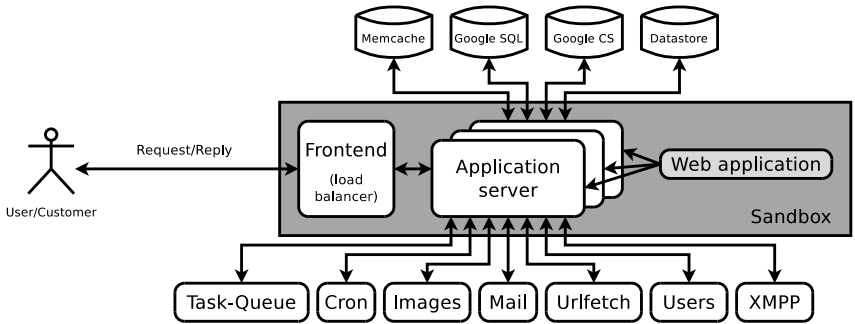


PaaS API

- The API is defined by the PaaS platform
- Could be implemented in public and private platform
- Could also be implemented on self-hosted hardware

FRANKFURT UNIVERSITY OF APPLIED SCIENCES

Platform as a Service example — Google AppEngine (1/4)

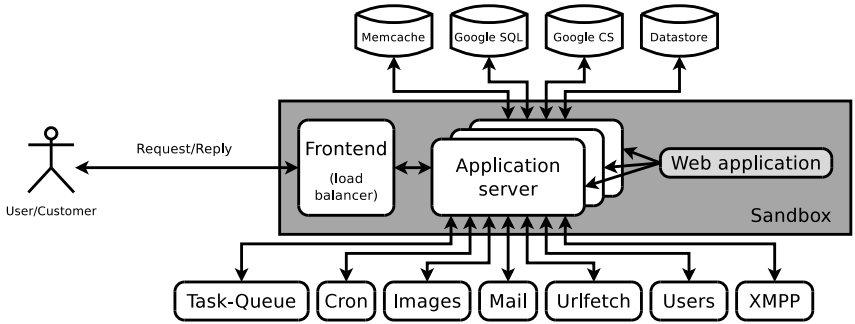


Google AppEngine

Google's PaaS offering. GAE is a good example type representative for Platform as a Service offerings. All other platforms and services follow similar considerations and technical restrictions. With GAE, applications run within the Google infrastructure.

Platform as a Service example

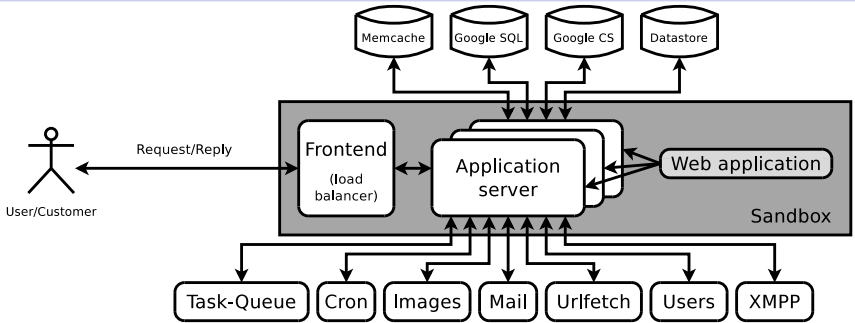
Google AppEngine (2/4)



Features

- Support for Node.js, Java, Ruby, C#, Go, Python, or PHP.
- Flexible environment instances are Compute Engine virtual machines.
- Customer specifies amount of CPU and memory of application.
- App Engine automatically scales applications based on incoming load.

Platform as a Service example — Google AppEngine (3/4)



Problems in PaaS

One of the main problems with PaaS since the early days of this service model has been the lack of standards. This applies in particular to the...

- deployment format of the applications to be hosted
- and the PaaS runtime interface.

Platform as a Service – Limitations of platforms Google AppEngine (4/4)

Limitations in PaaS

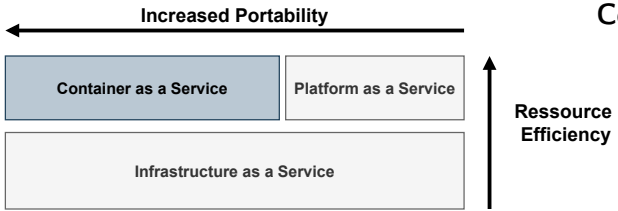
GAE has the following restrictions, for example:

- Applications cannot open their own threads and have no access to the runtime environment.
- Communication with other web applications is restricted and is only permitted via channels such as URL fetch, XMPP or e-mail.
- There are usually size limits for requests and responses (in the case of GAE, these may not be larger than 1 MB).
- It is often necessary to rely on special provider-specific libraries for applications, which cannot be transferred to other providers.

Solution

Standardization of deployment units ⇒ **Containers!**

Standardization of deployment units (container)



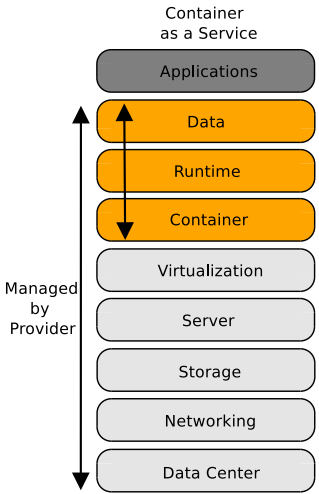
Container plattforms

- Applications and their dependencies can be packaged into containers.
- Containers have a small resource footprint and are more efficient.
- Containers can scale up to demands faster than VMs and react to changes faster

Future development

It has the potential to slowly displace the PaaS model due to better portability and better standardization.

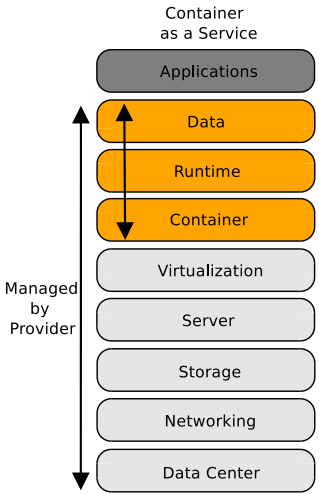
Container as a Service



CaaS

- Container as a Service (CaaS) is a cloud computing model that allows container-based virtualization to be used as a service from the cloud.
- As with all other service models, CaaS can be obtained as a managed service (e.g., as part of a public cloud) or as a self-hosted service (e.g., as part of a private cloud).

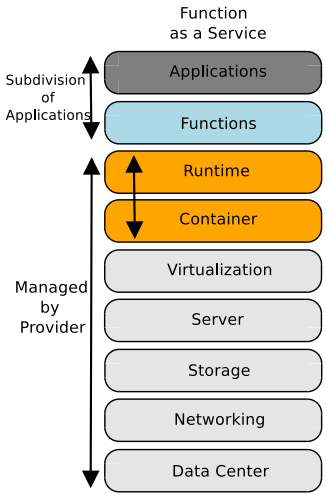
Container as a Service



Container platforms orchestrators

- **Kubernetes** – is an open-source tool for the automated deployment, scaling, and management of container applications on distributed IT infrastructures. Unlike Swarm, Kubernetes does not only support Docker containers. (More details ⇒ Slide Set 2!)
- **Docker Swarm** – open-source cluster management and orchestration tool developed by Docker as a native tool for managing Docker clusters and container operations (outdated)
- **Apache Mesos** – is a open-source cluster manager, provided by Mesosphere as an operating system for data centers under an open-source license. Mesos uses Linux cgroups to provide isolation for CPU, memory, I/O and file system.

Function as a Service

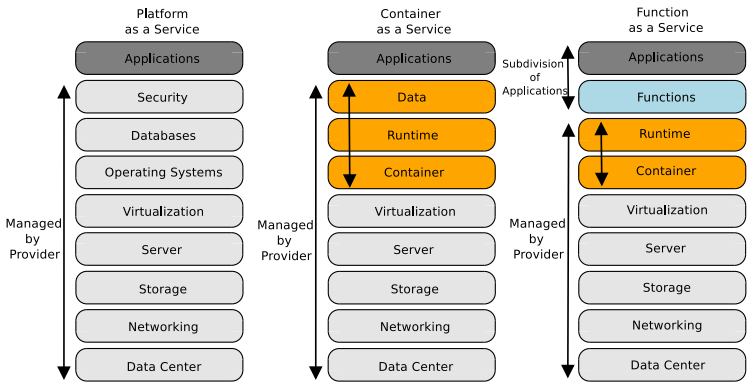


FaaS aka Serverless Computing

- It simplifies the provision of applications in the cloud, which can be easily divided into many fine-grained functions.
- Business logic is provided and operated as functions on a FaaS platform.
- This platform executes the functions on demand without needing infrastructure provisioning or maintenance from the customer's perspective.

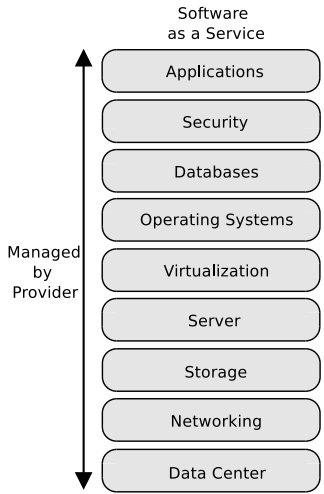
FRANKFURT UNIVERSITY OF APPLIED SCIENCES

Difference between PaaS, CaaS and FaaS



The infrastructural Ops aspect, which characterizes IaaS and is **partially abstracted away** in **PaaS** and **container orchestration/CaaS**, is **completely** absent in **FaaS** from a dev perspective. The focus is on the function and less on the execution environment of the function.

Software as a Service



SaaS

- Human-usable application software for customers.
- No need for operating infrastructure or platform by customers.
- Applications are usually accessed from various client devices, such as a web browser.
- User only manages user-specific configuration settings of the application

FRANKFURT UNIVERSITY OF APPLIED SCIENCES

Software as a Service – Characteristics

SaaS Characteristics

In the SaaS model, providers make IT resources available to customers in the form of **human-usable application software** for customers to enable **self-service, rapid elasticity, and pay-per-use pricing**. Small and medium-sized enterprises often need more workforce and expertise to develop custom software applications. Furthermore, many applications have become commoditized and are used by many companies, but more is needed to differentiate themselves from competitors. This includes, for example, office suites, collaboration software, or communication software.

SaaS responsibility

The capability provided to the consumer in SaaS consists of using a **provider's applications** without operating the necessary infrastructure or platform. The applications are usually accessed from various client devices, such as **a web browser** (e.g., web-based email) or via **a program interface**.

Software as a Service – Usage

Saas usage

The consumer does not manage or control the underlying **cloud infrastructure or platform**, including the network cloud platform, network, server, operating system, storage, or even individual application functions. However, **user-specific configuration settings** are possible - usually to a minimal extent (e.g., adaptation of the user interface to company style guide specifications).

FRANKFURT UNIVERSITY OF APPLIED SCIENCES

Public Cloud Computing offerings

Public Cloud Computing offerings

In the last 15 years many public Cloud Service offerings have become available and the number of Cloud Service Providers (CSPs) is huge! There are also german offerings e.g. **IONOS** and **Telekom**!

Hyperscalers

Since the number of CSPs is very high, this lecture focuses on the offering of the **three** biggest Hyperscalers on the market!

The three Hyperscalers are...

- **Amazon Web Services**
- **Google Cloud Platform**
- **Microsoft Azure**

FRANKFURT UNIVERSITY OF APPLIED SCIENCES

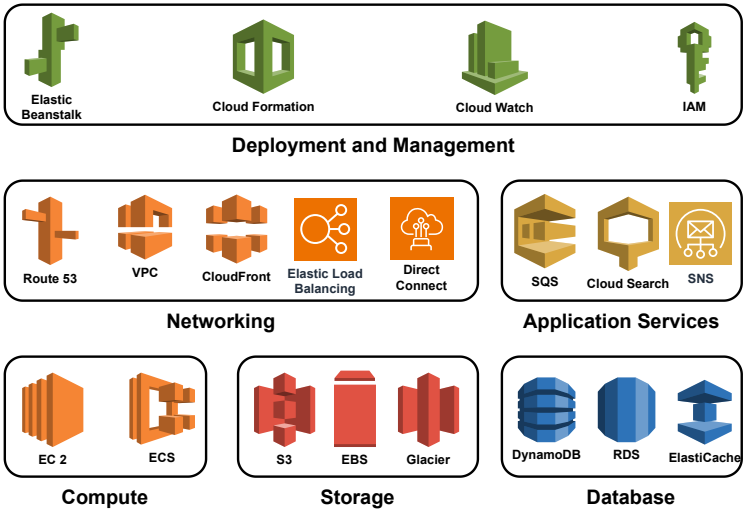
Amazon Web Services

- AWS date back to the early 2000s.
- `merchant.com` → an e-commerce-as-a-service platform for third-party retailers to create their own web stores.
- Amazon pursued a service-oriented architecture to scale its technical operations.

AWS origins

Amazon created „*a shared IT platform*“, because its technical organizations were spending 70% of their time on IT and infrastructure issues. Also handling unusual **traffic spikes**, especially during the **vacation season**, by migrating services to **commodity Linux hardware** and using **open source software** was an issue.

Amazon Web Services – Portfolio



FRANKFURT UNIVERSITY OF APPLIED SCIENCES

Amazon Web Services - Compute

Elastic Compute Cloud (Amazon EC2)

- Compute platform with choice for different processors, storage, networking, operating systems.
- Support for Intel, AMD, and Arm processors.
- Support for different RAM sizes and different Storage technologies (e.g. HDD or SSD)

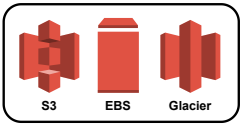
Elastic Container Service (Amazon ECS)

- is a fully managed container orchestration service for deploying, managing, and scaling containerized applications.
- It is deeply integrated into the AWS environment (interoperability with AWS services).
- It provides an easy-to-use solution for running containerized workloads in the cloud.

Amazon Web Services - Storage

Amazon Simple Storage Service (Amazon S3)

- S3 is an object storage service.
- Data is stored as an **object** in a **bucket**.
- An object is a file and all the metadata that describes this file.
- A bucket is a container for objects.
- Each object has a key, which is the unique identifier for the object in the bucket.



Storage

Amazon Elastic Block Store (Amazon EBS)

- Amazon EBS is a block storage service designed for Amazon EC2.
- Offers different classes of block storage devices (HDD and SSD).
- Attachment possible to existing or new EC2 instances.

Amazon S3 Glacier

- Amazon S3 Glacier storage is built for data archiving in the cloud.
- Three archive storage classes exist with different prices:
  - Instant Retrieval – fast retrieval (within milliseconds)
  - Flexible Retrieval – slow retrieval (within minutes, up to 12h)
  - Deep Archive – slowest retrieval (12h up to 48h)

Amazon Web Services - Database

Amazon DynamoDB

- DynamoDB is a NoSQL database and data is stored in Tables as **items**, and can be queried using **indices**.
- Items consist of a number of attributes which can belong to a number of data types,
- The **Key** that is expected to be unique across the Table.

Database

Amazon Relational Database Service

- Amazon RDS is a distributed and managed relational database service.
- Administration processes like patching the database software, backing up databases are managed automatically.

Amazon ElastiCache

- Amazon ElastiCache is a fully managed in-memory data store and cache service.
- It uses in-memory caches, instead of relying entirely on slower disk-based databases.
- The service supports two open-source in-memory caching engines:
  - Memcached and Redis

Amazon Web Services - Networking (1/2)

Amazon Route 53

- Amazon Route 53 provides highly available and scalable cloud services for Domain Name System (DNS).
- Provides reliable way to route end users to Internet applications by translating domain names.
- Implementation of your routing policies, and you can acquire and manage domain names and automatically configure DNS settings.

Amazon Virtual Private Cloud (Amazon VPC)

- Amazon VPC launches AWS resources in a logically isolated virtual network.
- The virtual network resembles a traditional network to operate for a private data center.

Amazon CloudFront

- Amazon CloudFront is a content delivery network (CDN) operated by Amazon Web Services.
- The CDN provides a globally-distributed network of proxy servers to cache content.
- It improves access speed for downloading the content.

Amazon Web Services - Networking (2/2)

Networking

Elastic Load Balancing (ELB)

- Elastic Load Balancing (ELB) automatically distributes incoming application traffic to multiple targets.
- ELB distributes traffic to virtual appliances in one or more Availability Zones (AZs).

AWS Direct Connect

- AWS Direct Connect connects services directly to the customers network.
- The network traffic remains in transit on the global AWS network and is not routed through the public Internet.
- It reduces the likelihood of bottlenecks or unexpected latency increases.

Amazon Web Services - Application Services

Amazon Simple Queue Service (Amazon SQS)

- Amazon SQS is a message queuing service for sending, storing, and receiving messages.
- It offers a secure, durable, and available hosted queue.
- It integrates and decouples distributed software systems and components.

Application Services

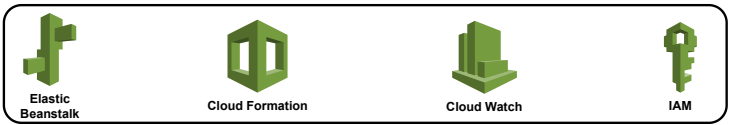
Amazon CloudSearch

- Amazon CloudSearch is a managed search solution service for websites or applications.

Amazon Simple Notification Service (Amazon SNS)

- Amazon SNS is a managed Publish/Subscribe service to Application-to-application (A2A) and Application-to-person (A2P) messages.
- Amazon SNS acts as a single message bus that can message to a variety of devices and platforms.

Amazon Web Services - Deployment and Management (1/2)



Deployment and Management

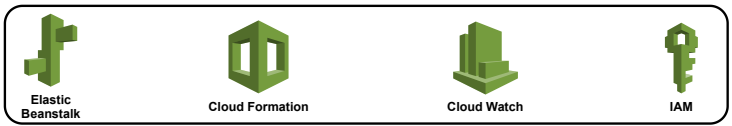
Elastic Beanstalk

- Elastic Beanstalk is a service for deploying and scaling web applications and services.
- It automatically handles the deployment—from capacity provisioning, load balancing, and auto scaling to application health monitoring.

CloudFormation

- AWS CloudFormation offers an automated management of infrastructures in AWS.
- It provides a way to model an entire AWS infrastructure in a text file (similar to Terraform).
- It allows version-controlling and the usage of templates.

Amazon Web Services - Deployment and Management (2/2)



Deployment and Management

CloudWatch

- AWS CloudWatch is a managed monitoring tool for services and resources in AWS.
- It responds to performance changes, optimizes resource use, and provides insights into operational health.

Identity and Access Management (IAM)

- AWS IAM is a service for the management of identities and permissions for services.
- It provides a fine-grained permissions and attribute-based access control.

Google Cloud Platform and Microsoft Azure

Because the big CSPs have similar or comparable services as AWS, we will not discuss every CSP in this lecture!

FRANKFURT UNIVERSITY OF APPLIED SCIENCES

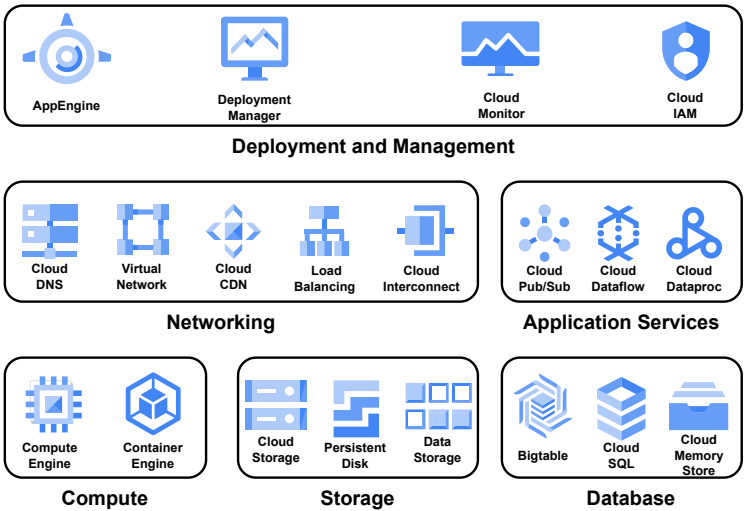
Google Cloud Platform

Google Cloud Platform

- Google announced the PaaS service **App Engine** (see slide 21) in April 2008 for developing and hosting web applications
- This was the first cloud service of Google
- Since then, Google has expanded their offer of services and has evolved to one of the biggest hyperscalers

FRANKFURT UNIVERSITY OF APPLIED SCIENCES

Google Cloud Platform – Portfolio

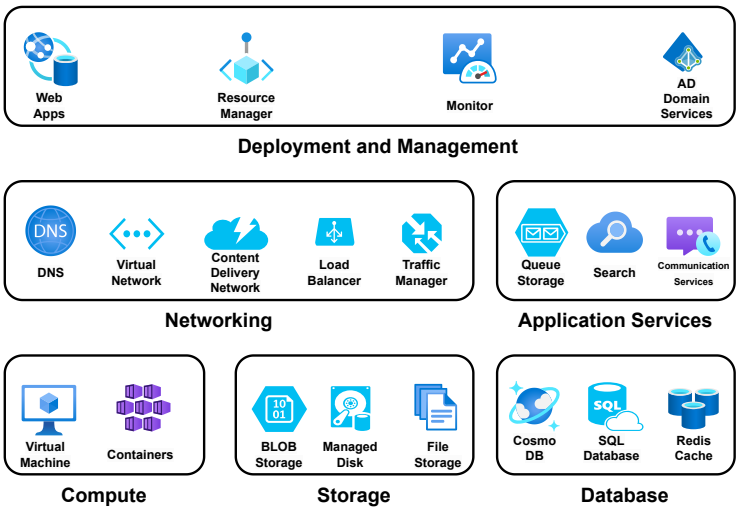


Microsoft Azure

- Azure was announced in October 2008 under the name „Project Red Dog“.
- It was officially launched as Windows Azure in February 2010.
- It was renamed to Microsoft Azure in March 2014.

FRANKFURT UNIVERSITY OF APPLIED SCIENCES

Microsoft Azure – Portfolio



FRANKFURT UNIVERSITY OF APPLIED SCIENCES

Private Cloud Computing offerings

Private Cloud Computing offerings

There are a large number of products available, that allow the construction of own, self-hosted cloud services. The categories are various, ranging from IaaS to SaaS platforms. There are also offerings, which are licensed by companies and require fees. This lecture only discusses open-source solutions for the creation of services!

Plattforms

Since the number of platforms is very high, we only discuss some of the most prominent examples. This lecture presents offerings for IaaS, PaaS, CaaS and SaaS setups.

FRANKFURT UNIVERSITY OF APPLIED SCIENCES

Private Cloud Computing offerings – IaaS

Private IaaS platforms

Private Iaas platforms provide components for the setup of infrastructure services, like compute, networking and storage resources.

Open-source platforms are:

- **OpenStack**
- **OpenNebula**

FRANKFURT UNIVERSITY OF APPLIED SCIENCES

Rackspace OpenStack (1/4)

- OpenStack began in 2010 as a joint project of Rackspace Hosting and NASA.
- Since 2012 it is managed by the **OpenStack Foundation**.
- It is an open-source offering for the setup of private IaaS offerings.

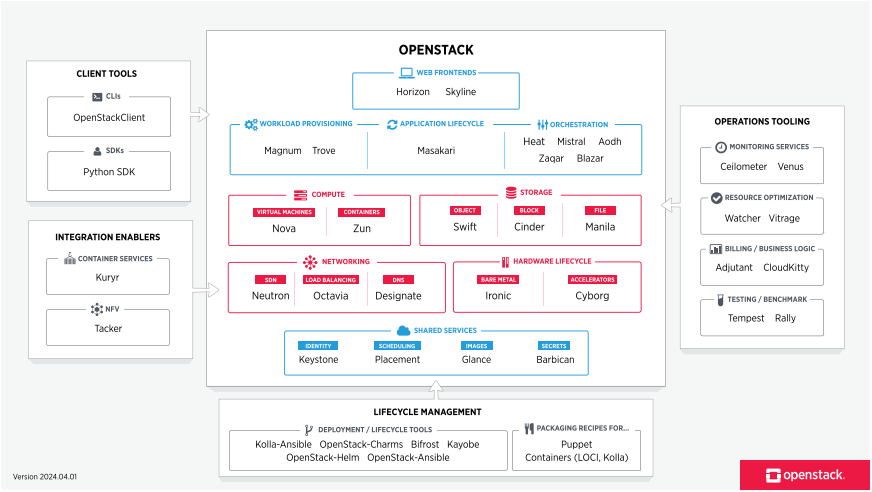
OpenStack

OpenStack can be deployed as IaaS in both public and private setups, where virtual servers and other resources are made available to users. The software platform offers components that control diverse, multi-vendor hardware pools of processing, storage, and networking resources in a data center.

FRANKFURT UNIVERSITY OF APPLIED SCIENCES

Rackspace OpenStack

image source:https://www.openstack.org/software/(2/4)



Rackspace OpenStack (3/4)

- **Components for computing**
  - **Nova** for provisioning compute instances. Nova supports creating virtual machines, baremetal servers.
  - **Zun** is Container service. It provides an API service for running application containers without the need to manage servers or clusters.
- **Components for hardware**
  - **Ironic** is a component which provisions bare metal machines. It integrates with Compute (nova), Network (neutron), Image (glance), and Object (swift) services.
  - **Cyborg** is a general management framework for accelerators

OpenStack (4/4)

- **Components for storage**
  - **Swift** is a highly available, distributed, eventually consistent object/blob store.
  - **Cinder** is the Block Storage service for providing volumes to Nova virtual machines, Ironic bare metal hosts, containers and more.
  - **Manila** is the Shared Filesystems service for providing Shared Filesystems as a service.
- **Components for networking**
  - **Neutron** provides network connectivity between interface devices (e.g., vNICs) managed by other OpenStack services (e.g., nova). It implements the OpenStack Networking API.
  - **Octavia** is a load balancer in OpenStack.
  - **Designate** is a multi-tenant DNSaaS service for OpenStack. It provides a REST API. It can be configured to auto-generate records based on Nova and Neutron actions.

OpenNebula (1/3)

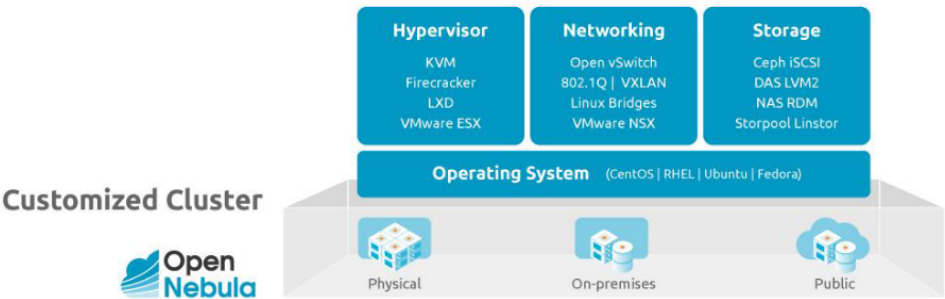
- OpenNebula started as a research project in 2005 by Ignacio M. Llorente and Ruben S. Montero.
- The first public release of the software occurred in 2008.
- The goals were to create solutions for managing virtual machines on distributed infrastructures.
- Since March 2010 part of OpenNebula Systems, which provides professional services to enterprises.

OpenNebula

OpenNebula manages on-premises and remote virtual infrastructure to build private, public, or hybrid implementations of IaaS and multi-tenant Kubernetes deployments. The two primary uses of the OpenNebula platform are data center virtualization and cloud deployments based on the KVM hypervisor.

OpenNebula

image source:https://support.opennebula.pro/hc/en-us/article_attachments/7124526151825 (2/3)



OpenNebula (3/3)

Components

OpenNebula can be used with open-source or commercial products for hypervisors, networking and storage.

Deployment

OpenNebula supports the deployment on bare-metal or in on-premise premise setups. It can also be deployed on hosted environments of CSPs (AWS, GCP, Azure).

Private Cloud Computing offerings – PaaS

Private PaaS platforms

Private Iaas platforms provide components for the setup of platform services, like runtimes for applications and APIs to other services from public CSPs.

Open-source platforms are:

- **AppScale**
- **CloudFoundry**

FRANKFURT UNIVERSITY OF APPLIED SCIENCES

AppScale

AppScale

- AppScale started as research project at the University of California.
- AppScale systems was founded in 2012 for the commercial support of the platform.
- AppScale ATS is a managed hybrid cloud infrastructure software platform that emulates the core AWS APIs.
- AppScale GTS is an open source platform for building, deploying and running web applications.

AppScale GTS

AppScale GTS automatically deploys and scales unmodified Google App Engine applications on-premises clusters. AppScale is modeled on the App Engine APIs and supports Go, Java, PHP, and Python applications.

AppScale GTS

It has the following components:

- **Datastore API:** Apache Cassandra (database) and Apache ZooKeeper (configuration information)
- **Memcache API:** memcached
- **Task Queue API:** RabbitMQ and Celery
- **Messaging API:** ejabberd for Extensible Messaging and Presence Protocol (XMPP)
- **Blobstore API:** Apache Cassandra and Apache ZooKeeper
- **Images API:** Python Imaging Library (PIL)
- **Cron API:** Crontab (standard Linux)
- **Proxy and Load balancing:** HAProxy

AppScale GTS

AppScale GTS decouples app logic from its service ecosystem. This allows the development of web applications with fault-tolerance, and auto-scaling capabilities.

NKFURT
VERSITY
CIENCES

Cloud Foundry (1/2)

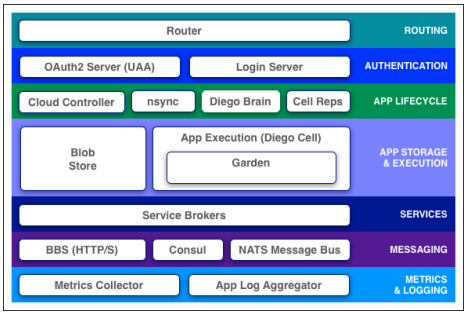
- Founded in 2009, it was designed and developed by a small team at VMware.
- Cloud Foundry is an open source platform as a service (PaaS) governed by the Cloud Foundry Foundation.
- Since 2015 it is part of the Linux Foundation Collaborative Project.

Cloud Foundry

Cloud Foundry supports the full application development lifecycle, from initial development through all testing stages to deployment. Cloud Foundry's container-based architecture runs apps and supports various programming languages like Java, Ruby and Python.

Cloud Foundry

image source:https://docs.cloudfoundry.org/concepts/architecture/ (2/2)



- **Router:** Internal Routing and Load Balancing.
- **Authentication:** Access and Authorization.
- **App Lifecycle:** Management of lifecycle.
- **Execution:** Runtime of application inside a container (Diego). BLOB store for the package data.
- **Broker:** Service broker for communication of service components.
- **Messaging:** messages of the application internally.

Source

A good source fo information:
Winn, D. C. E. (2017). **Cloud Foundry: The Definitive Guide: Develop, Deploy, and Scale. United States**: O'Reilly Media.

BOSH

Cloud Foundry uses BOSH. It is an open-source software project that offers a toolchain for release engineering, software deployment and application lifecycle management of large-scale distributed services.

Private Cloud Computing offerings – FaaS

Private FaaS platforms

There are platforms for setting up Function as a Service offerings on-premise, which are also open-source and available for the construction of serverless platforms in a private context.

Open-source platforms are:

- **OpenFaaS**
- **Apache OpenWhisk**

FRANKFURT UNIVERSITY OF APPLIED SCIENCES

OpenFaaS (1/2)

OPENFAAS

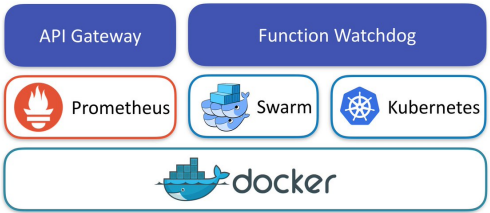
- OpenFaas was developed as an independent open-source project originally created by Alex Ellis in 2016.
- It provides a platform for the development and deployment of event-driven applications.
- It package the function code or an existing binary in a Docker image.
- Support for OCI-compatible images for scalable endpoints with auto-scaling and metrics.

OpenFaaS

OpenFaas provides a platform for the implementation of functions in different programming languages like Go, Python or Java. It provides autoscaling features for functions deployed on the platform by using a API Gateway.

OpenFaaS (2/2)

Functions as a Service



Open Source*

OpenFaaS offers three models:

- **Community Edition:** Limited version for personal-use and experimentation.
- **OpenFaaS Standard:** Designed for production workload.
- **OpenFaaS for Enterprises:** Highest level of service for enterprises and multi-tenant hosting.

- **API Gateway:** scales functions according to demand by altering the service replica count.
- **Function Watchdog:** entrypoint allowing HTTP requests to be forwarded to the target process via `STDIN`.
- **Prometheus/Grafana:** monitoring and metrics collection.
- **Runtime and Orchestration:** it uses Kubernetes for orchestration and OCI as a runtime environment for workloads.

Cloud-Native

OpenFaaS provides an opportunity to deploy and operate cloud-native applications, which are follow the development paradigm of DevOps.

Apache OpenWhisk (1/3)

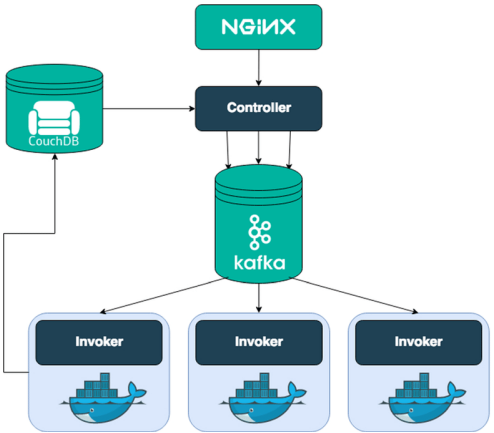
- Apache OpenWhisk was developed by Apache and is also part of IBM cloud functions
- It is an open source, distributed serverless platform for functions.
- It manages the infrastructure, servers and scaling using Docker containers.
- It provides a programming model for the implementation of event-driven, cloud-native applications

OpenWhisk

Apache OpenWhisk is designed to provide a serverless platform for large scale service offerings and supports programming languages such as Go, Java, NodeJS, .NET, PHP, Python, Ruby, etc.

OpenWhisk

image source: https://openwhisk.apache.org/documentation.html (2/3)



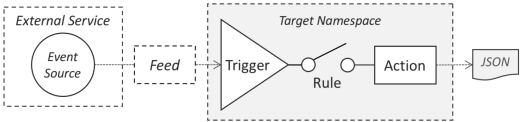
- **Nginx:** Loadbalancer for incoming requests and forwarding requests to the controller.
- **Controller:** Checks incoming requests and controls the further action.
- **Apache Kafka:** Publish-Subscribe Messaging Service, which queues the requests.
- **CouchDB:** Responsible for authentication of requests (permission checking) and storing information on the imported functions.
- **Invoker:** Docker container(s) running the function and each Invoker can be paused for faster request (later) fulfillment.

Cloud-Native

OpenWhisk provides an opportunity to deploy and operate cloud-native applications, which are follow the development paradigm of DevOps.

OpenWhisk

image source: https://openwhisk.apache.org/documentation.html (3/3)



- **Event:** An action caused by an external event.
- **Triggers:** The external events **trigger** the execution of an **Action**, which is associated by a **Feed** or from HTTP requests.
- **Rules:** Definition of conditions that are associated with **Triggers** and **Actions**.
- **Actions:** Functional logic is called an **Action** and responsible for the execution of the code.
- **JSON:** The result of the function execution is returned as a JSON.

Programming model

The OpenWhisk platform supports an event-driven programming model in which developers write functional logic and deploy it in the platform. The project includes a REST API-based Command Line Interface (CLI) along with other tooling to support packaging, catalog services and many container deployment options.

Private Cloud Computing offerings – SaaS

Private SaaS offerings

Deploying and hosting private SaaS applications is basically hosting a web application on-premise! There are many tools for this purpose, but they cannot be associated exclusively to cloud computing.

Hosting private SaaS

By deploying a web application on a public platform (e.g. **IaaS** → OpenStack, OpenNebula or e.g. **PaaS** → AppScale, Cloud Foundry or **Faas** → OpenWhisk) and hosting the application, it becomes a private SaaS application.

FRANKFURT UNIVERSITY OF APPLIED SCIENCES

Summary

In this lecture we have discussed the following topics:

- The different deployment models in cloud computing (private, public, community and hybrid cloud)
- The different service models in cloud computing (IaaS, PaaS, CaaS, FaaS, SaaS)
- The Offerings of three of the biggest public CSPs on the market (AWS →**in detail!**, GCP and Azure only Portfolio)
- Different private platforms for cloud computing setups (Iaas, PaaS, FaaS)

FRANKFURT UNIVERSITY OF APPLIED SCIENCES

Outlook

~~1st part:~~ ~~**Introduction**~~

~~2nd part:~~ ~~**Technological foundations**~~

3rd part: **Service models, deployment models** $\Longleftarrow$ *This slide set*

4th part: **Adoption and strategy**

5th part: **Architectures and applications**

6th part: **Cloud-Native applications**

7th part: **Current and future trends**

FRANKFURT UNIVERSITY OF APPLIED SCIENCES

4th part: Adoption and strategy

Topics:

- Cloud adoption in a business perspective
- Cloud strategy, key terms, methods and general knowledge
- Multi-Cloud and the differences to Hybrid Cloud

FRANKFURT UNIVERSITY OF APPLIED SCIENCES

Thank You For Your Attention!

Henry-Norbert Cocos, M.Sc
Frankfurt University of Applied Sciences
Room 1-230
☎ +49 69 1533-2699
✉ cocos@fb2.fra-uas.de
www.henrycocos.de