Cloud Foundry Glossary

Contents

Glossary

General Terms

Backing Service

A backing service is any service an application consumes over the network as part of its normal operation. Examples include datastores (such as HANA, Postgresql or CouchDB), messaging/ queuing systems (such as RabbitMQ or Beanstalkd), SMTP services for outbound email (such as Postfix), and caching systems (such as Memcached or Redis). The backing service term has a major role in the 12 Factor manifesto.

BLOB Store - Binary Large Object Store

A BLOB Store is a software component to save binary large object data (images, videos, assets, …). Most BLOB Store implementations provide a HTTP REST interface for managing BLOBs and implement mechanism for data replication and high availability. Prominent examples for BLOB Store implementations are Amazon S3 [4] and OpenStack Swift [5]. In the modern Cloud context BLOB Stores are used as a replacement for traditional filesystem storage.

IaaS - Infrastructure as a Service

Infrastructure as a Service is a form of cloud computing that provides virtualized computing resources over the Internet. IaaS layers are installed on top of hardware and provide an API to manage virtual machines (VMs), networks and storage volumes on demand. Prominent examples for IaaS systems are Amazon Web Services [9], OpenStack [10] and VmWare VSphere [11].

PaaS - Platform as a Service

Platform as a Service systems provide a development platform for application developers within the Cloud context. Applications and Services are the main citizens of these environments in contrast to virtual machines in IaaS layers. PaaS systems expose an API for managing application lifecycles in big distributed environments by providing the following core features:

- On demand self-service
- Scalability
- Automated provisioning
- Multi tenancy
- Monitoring & Reporting
- Self healing

In most cases PaaS systems sit on top of IaaS layers for orchestrating resource provisioning. Prominent examples for PaaS systems are Cloud Foundry [7], OpenShift [8] and Heroku [6].

Scaling

Scaling is the ability of a computer application or product (hardware or software) to continue to function well as it (or its context) is changed in size or volume in order to meet a user need.

Modern Cloud systems provide well defined mechanisms to scale hosted software components on demand to adopt rapid growth in resource consumption.

12 Factors / 12 Factor Manifesto

The 12 Factor manifest [3] defines 12 rules for application development in PaaS contexts. The 12 Factors form the contract between applications and Platform as a Service layers.

Cloud Foundry Terms (CF)

Cloud Foundry [7] is the leading open source Platform as a Service framework by the time of this writing. Cloud Foundry was initially released in 2011 by VmWare and Pivotal Software and is now maintained by the Cloud Foundry Foundation under guidance of the Linux Foundation [12]. The major goal is to establish an open PaaS standard. Cloud Foundry and its components are released under the Apache License 2.0 [13].

App/Application

In the context of Cloud Foundry an application is a piece of software running within the Cloud Foundry runtime. Each application has a unique identifier and can consist of multiple application instances. Applications are run within containers (DEA & Warden) to allow fast startup and isolation from other applications. Applications can be created via the Cloud Foundry api [14] by uploading source code or precompiled assets (e.g. war files). The uploaded bits are processed using programming language specific buildpacks in a staging phase. After finishing the staging phase successfully the resulting droplets are distributed over the runtime containers [16]. Applications are the first class citizens of the Cloud Foundry runtime.

Application/App Deployment

Application deployment is the process of rolling out applications to the Cloud Foundry hosting platform.

Application/App Instance

An application instance is one running process of an application. Application instances are Application droplets started within containers. Each application can be hosted by multiple application instances for scaling the application as needed.

Application Logs

Application logs are log messages emitted by running applications. Application logs are aggregated by the Cloud Foundry runtime.

Buildpack

Buildpacks [15] provide framework and runtime support for applications by processing user provided artifacts (source code, war files, …) into runnable droplets. Buildpacks are used during the staging phase of an application.

cf CLI

The cf command line interface [18] is an application allowing users to interact with the Cloud Foundry API. It is written in the go programming language and provided as binaries for all common operating systems.

Cloud Controller

The Cloud Controller [17] exposes the Cloud Foundry API [14] to consumers and operators of the PaaS offering. It forms the main entry point for Cloud Foundry interactions and stores system information in an underlying SQL database.

DEA - Droplet Execution Agent

The DEA is the component in the Cloud Foundry runtime responsible for staging and hosting applications. It provides containers for running application instances. The current DEA implementation will be substituted by the follow up implementation called Diego [19].

Domain

The Cloud Foundry runtime allows mapping domains to multiple spaces. Those domains can be routed to hosted applications to enable external access via web domains.

Droplet

A droplet is an archive within the Cloud Foundry runtime that contains an application ready to run on a DEA. A droplet is the result of the application staging process.

HM9000 - Cloud Foundry Health Manager

The Cloud Foundry health manager is the component responsible for checking application health in the CF PaaS and triggering countermeasures on application instance outages.

Organization/Org

An org is the top-most meta object within the Cloud Foundry runtime. It can represent companies or departments in case of an enterprise Cloud Foundry installation. Organizations can contain multiple spaces.

Route

A route is a mapping of a url consisting of a hostname and a domain to an application. Routes are processed using the Cloud Foundry router.

Router

The router is the component in the Cloud Foundry runtime responsible for routing requests on hosted applications to the according application instances.

Service

A service is a “factory” which produces service instances. A service provides software for usage within the Cloud Foundry system (e.g databases, message queues, BLOB stores, …).

Service Binding

A service binding [22] forms the connection between a service instance and an application instance in the Cloud Foundry runtime. It is represented by a set of credentials provided to bound applications as runtime environment variables.

Service Broker

A service broker forms the connection between the Cloud Foundry runtime and a service offering. Service brokers have to be implemented according to the service broker API specification [23].

Service Instance

A service instance is a resource provisioned by a service. The resource provisioned will differ by service and could be a database, a database cluster or an account on a multi-tenant application.

Space

A space is an organizational unit for Cloud Foundry organizations. A space can contain applications and service instances. In most cases spaces are used for separating development environments (e.g. testing and production spaces) or departments from each other.

Staging

During the staging phase of an application user provided artifacts (e.g. source code, war files) are processed into droplets. The staging phase is performed by executing a buildpack within a container of a DEA instance.

UAA - User Account and Authentication Server

The UAA [21] is the component in the Cloud Foundry runtime responsible for authenticating and authorizing users with the system.

Warden

Warden is a software tool that handles containerization on Cloud Foundry's DEAs. Warden ensures application instance isolation and a fair share of computing resources.

Links

[1] – not used –

[2] – not used –

[3] 12 Factor manifesto - http://12factor.net/

[4] Official OpenStack Swift website - http://docs.openstack.org/developer/swift/

[5] Amazon S3 documentation - http://aws.amazon.com/de/documentation/s3/

[6] Heroku - https://www.heroku.com/

[7] Cloud Foundry - http://cloudfoundry.org/about/index.html

[8] RedHat OpenShift - https://www.openshift.com/

[9] Amazon Web Services (AWS) - http://aws.amazon.com/de/

[10] OpenStack - https://www.openstack.org/

[11] VmWare VSphere - http://www.vmware.com/de/products/vsphere

[12] Press Release Cloud Foundry Foundation / Linux Foundation - http://www.linuxfoundation.org/news-media/announcements/2014/12/cloud-foundry-foundation-established-advance-platform-service

[13] Apache License 2.0 - http://www.apache.org/licenses/LICENSE-2.0.html

[14] Cloud Foundry API documentation - http://apidocs.cloudfoundry.org/207/

[15] Cloud Foundry Buildpack documentation - http://docs.cloudfoundry.org/buildpacks/

[16] Cloud Foundry application deployment documentation - http://docs.cloudfoundry.org/devguide/deploy-apps/deploy-app.html

[17] Cloud Controller documentation - http://docs.cloudfoundry.org/concepts/architecture/cloud-controller.html

[18] CF CLI documentation - http://docs.cloudfoundry.org/devguide/installcf/install-go-cli.html

[19] Cloud Foundry - Diego Explained - http://www.activestate.com/blog/2014/09/cloud-foundry-diego-explained-onsi-fakhouri

[20] Cloud Foundry Health manager documentation - http://docs.pivotal.io/pivotalcf/concepts/architecture/#hm9k

[21] Cloud Foundry UAA documentation - http://docs.cloudfoundry.org/concepts/architecture/uaa.html

[22] Service binding documentation - http://docs.cloudfoundry.org/devguide/services/bind-service.html

[23] Service broker API specification - http://docs.cloudfoundry.org/services/api.html

[24] Vagrant documentation - https://docs.vagrantup.com/v2/

[25] Cloud Foundry BOSH release repository - https://github.com/cloudfoundry/cf-release

[26] Official Cloud Foundry Glossary - http://docs.pivotal.io/pivotalcf/concepts/glossary.html

[27] Official BOSH Glossary - https://bosh.io/docs/terminology.html

[28] Docker website - http://www.docker.com/