API Application Programming Interface.

Artifact
An artifact is a deployable unit that is supported by a specific cartridge type (e.g., a PHP file in a PHP cartridge, a web app `.war` in an Application Server Cartridge).

Auto-scaling policy
An auto-scaling policy is a policy that determines the auto-scaling process, based on the load thresholds. The load thresholds in turn are determined based on the requests in flight, memory consumption and load average.

Availability zone
An availability zone is an isolated location within a region used to define one or more partitions. Thereby, availability zones provide physical separation (not geographical separation) and redundancy.

Cartridge
A cartridge is a container for an application, framework or data management system that could be deployed in a platform as a service (PaaS) for scalability. In Stratos, service runtimes are created by cartridge runtimes. For example, if you want to have a PHP runtime, you can use a PHP cartridge to get the runtime to deploy your PHP application.

Cartridge Agent
Cartridge Agent is a component that resides within a cartridge instance and handles the communication between the cartridge and Stratos.

Cartridge Group
A cartridge group is a metadata that you can define by grouping several cartridges together. Composite applications in Stratos, support nested groups within a group as inter-dependencies among group members. Cartridge groups define the relationship among a set of groups and a set of cartridges. The relationship among the children of a group can be start up order, termination behavior and any scalable dependencies. Writing a group definition provides the ability to re-use the same frequently used group as needed in different composite applications.

Cartridge image
A cartridge image is a virtual machine (VM) image of a cartridge in Apache Stratos. The terminology used for VM images will vary based on the IaaS. For example, in EC2 and OpenStack VM images are referred to as Amazon Machine Images (AMIs) and images respectively.
Cartridge instance

A cartridge instance is an instance of a cartridge image.

Deployment policy

A deployment policy defines how (i.e., what partition algorithm to use) and where to spawn cartridge instances, and it also defines the maximum instances that are allowed in a service cluster. DevOps define deployment policies based on the deployment patterns. The network partitions, which correspond to a deployment policy, need to be added before the deployment policy is added.

Deployment synchronization

Deployment Synchronization is the process of synchronizing the deployed artifacts, which are in the remote Git repository, with the relevant cartridge instances.

Deployment synchronizer

Deployment Synchronizer is the component that carries out Deployment Synchronization.

Instance type

An instance type defines the following specifications of an instance: memory, CPU, storage capacity, and hourly cost for an instance. There can be one or more instance sizes based on a particular instance type. For example, m3.medium, m3.large, m3.xlarge etc.

Key pair

A key pair consists of a public key and a private key, and is used to prove your identity.

LAMP stack

This acronym is short for Linux, Apache, MySQL and PHP. This refers to an open-source Web development platform, also referred to as a Web stack, which uses Linux as the operating system, Apache as the Web server, MySQL as the RDBMS and PHP as the object-oriented scripting language.

LXC

LinuX Containers (LXC) is an operating system–level virtualization method that allows multiple isolated Linux systems (containers) to run on a single control host.

Partition algorithm

A partition algorithm in Apache Stratos is the effective method used in spawning instances between network partitions. The available partition algorithms are Round Robin and One after the other. When using the Round Robin algorithm, instances will be spawned in each of the partitions sequentially. When using the One after the other algorithm, instances will be spawned in the same partition until the maximum number of instances for the respective partition is reached, before spawning instances in the next partition. For example, if there are 2 partitions named P1 and P2, if Round Robin is applied, after an instance is spawned in P1, it will spawn the next instance in P2. If One after the other algorithm is applied, instances will be spawned in the same partition until the maximum number of instances are reached, before spawning instances in P2.
Network partition

A network partition is a network bounded area in a IaaS, where private IPs can be used for communication. Network partitions are also referred to as partition groups. The partition algorithm that is applied within a network partition may differ from the partition algorithm that is applied between different network partitions.

Partition

A partition depicts the division in an IaaS. A partition can be made at one of the following levels: provider level, region level or zone level. A partition should at least have a provider defined. The Auto-scaler makes decisions based on the defined partitions. The partitions are defined in the partitions.json file in the registry. The partitions.json file contains configurations on sample partitions.

Region

A region demarcates a geographically separate area used to define one or more partitions. Thereby, regions provide geographical separation and redundancy.

Scaling down

Scaling down refers to the auto-scaling system automatically shutting down the additional service instances during a non-peak-time.

Scaling up

Scaling up refers to the auto-scaling system automatically spawning up the service instances during the rush time.

Security group

A security group defines a set of IP filter rules for an instance. Each security group includes a list of protocols, ports, and IP address ranges. A security group can be applied to multiple instances; while, multiple security groups can be applied for a single instance.

SOAP