4.1.x Features

Apache Stratos includes polyglot language and environment support together with the ability to run on top of multiple IaaS runtimes. The following are the key features available in Apache Stratos 4.1.x:

Composite Application support | Docker Support | Cartridge Agent | Update artifacts at run-time | Cloud Controller | Multi-factored auto-scaling | Smart policies | Multiple IaaS support | Multi-cloud bursting | Controlling IaaS resources | Loosely coupled communication | Multi-tenancy | Cartridges | Pluggable architecture support for cartridges | Cartridge automation using Puppet | Support for third party load balancers | Artifact distribution coordination | Stratos Manager Console | Stratos REST API | Interactive CLI Tool | Logging | Monitoring and metering | Persistent volume support for cartridges | Gracefully shutdown instances | MQTT and AMQP support | Stratos Mock IaaS

- Composite Application support

  This allows applications, which requires different service runtimes with their relationship and dependencies, to be deployed. Furthermore, each of the service runtimes in the application can scale independently or jointly with the dependent services. The following operations are supported in composite application support:

  - Starting up instances using the StartupOrder, which is defined in the cartridge group definition.
  - Starting up instances using the StartupOrder, which is defined in the application definition. After the relevant clusters and groups get activated according to the startup order, the application itself gets activated.
  - Termination based on the TerminationBehaviour, which is defined in the cartridge group definition.
  - Termination based on the TerminationBehaviour, which is defined in the application definition.
  - Sharing information between instances when one instance is dependent on another.

  For detailed information on how composite applications work in Stratos, see Composite Applications.

- Docker Support
  Docker support using Google Kubernetes and CoreOS. Thereby, Stratos can also leverage the use of Docker in a PaaS. The following aspects are supported for Docker:
  - Auto-scaling Docker Containers.
  - Manual scaling Docker Containers.
  - Git based artifact deployment for Docker.
  - CLI support for Docker deployments.
  - VM LB support for dockers
  - Private Docker registry
- Cartridge Agent

  A Python based and Java based cartridge agent is available in Stratos.
- Update artifacts at run-time

  After an application is deployed, Stratos allows users to update the following artifacts, which directly effects the runtime.

  - Auto-scaling policy definition
  - Deployment policy definition
  - Application definition
- Cloud Controller
  Cloud Controller (CC) leverages Apache jclouds' APIs and provides a generic interface to communicate with different IaaSs.
- Multi-factored auto-scaling
  The Auto-scaler uses a Complex Event Processor (CEP) for real-time decision making, and it integrates both real-time and rule-base decision making to provide better control over scaling of platforms. Stratos allows users to define auto-scaling policies with multiple factors,i.e., requests in flight, memory consumption and load average, which are considered when scaling up or down. The Auto-scaler also supports scaling for non-HTTP transport.
- Smart policies
  The Auto-scaler in Stratos uses two smart policies when making auto-scaling decisions: auto-scaling policy and deployment policy. The instances will be automatically spawned based on the smart policies that are

applied to the application. For more information on auto-scaling and deployment policy scaling policies, see Smart Policies.

- Multiple IaaS support
  Apache Stratos is tested on the following IaaS providers: AWS EC2 and OpenStack. However, it is very easy to extend Apache Stratos to support any IaaS that is supported by Apache jclouds (i.e., Google cloud, CloudStack etc.).
- Multi-cloud bursting

  Apache Stratos supports multiple IaaSs. When the maximum limit of instances have been reached in an IaaS, instances are spawned on another IaaS, which is in another network partition. Thereby, this will enable resource peak times to be off-loaded to another cloud.
- Controlling IaaS resources
  It is possible for DevOps to define partitions in a network partition, to control IaaS resources. Thereby, Apache Stratos can control resources per cloud, region, and zone. Controlling of IaaS resources provide a high availability and solves disaster recovery concerns. For more information, see Cloud Partitioning.
- Loosely coupled communication
  Stratos uses the Advanced Message Queuing Protocol (AMQP) messaging technology for communication among all its components . Apache Stratos uses an AMQP Message Broker (MB), namely ActiveMQ, to communicate in a loosely coupled fashion. However, it is possible to use any MB, which supports AMQP, with Stratos.
- Multi-tenancy
  Stratos supports in-container multi-tenancy. Thereby, this helps to optimize the resource utilization.
- Cartridges
  In Stratos, service runtimes are created by cartridge runtimes. Currently, Stratos provides the following cartridges: Node.js, Wordpress, PHP, MySQL, WSO2 Application Server, Java, jBoss, Ruby and Tomcat. However, users can easily use Stratos to create **any cartridge**, which is either a framework, data, application or load balancer cartridge, i.e., MongoDB, Ruby on Rails, .NET, Spring, Joomla, Struts, PostgreSQL, etc.
- Pluggable architecture support for cartridges
  A cartridge is a package of code that includes a Virtual Machine (VM) image plus additional configuration, which can be plugged into Stratos to offer a new PaaS service. Stratos supports single tenant and multi-tenant cartridges. If needed, tenants can easily add their own cartridges to Stratos. For more information on how Stratos uses cartridges, see Cartridge.
- Cartridge automation using Puppet
  Cartridges can be easily configured with the use of an orchestration layer such as Puppet.
- Support for third party load balancers
  Stratos supports third-party load balancers (LBs), i.e, HAProxy, NGINX. Thereby, if required, users can use their own LB with Stratos.
- Artifact distribution coordination
  The Artifact Distribution Coordinator is responsible for the distribution of artifacts. Artifacts can be uploaded using `git push`. When a trigger event happens the ADC will find the correct matching cluster for that event from the topology and send notifications to appropriate Cartridge instances. ADC supports external Git repositories and GitHub repositories based deployment synchronization. Users are able to use their own Git repository to sync artifacts with a service instance. For more information, see Artifact Distribution Coordinator.
- Stratos Manager Console
  Administrators and tenants can use the Stratos Manager console, which is a web-based UI management console in Stratos, to interact with Stratos. For more information, see Stratos Manager.
- Stratos REST API
  DevOps can use REST APIs to carry out various administering functions (e.g., adding a tenant, adding a cartridge, etc.). For more information, see the Stratos API Reference Guide.
- Interactive CLI Tool
  Command Line Interface (CLI) tool provides users an interface to interact with Stratos and manage your applications. For more information, see the CLI Tool and the CLI Guide.
- Logging
  Stratos provides centralized logging. Any Thrift receiver can be used for this purpose.
- Monitoring and metering
  Apache Stratos provides centralized monitoring and metering. The level of resource utilization in Stratos is measured using metering.
- Persistent volume support for cartridges
  If required, the DevOps can enable a persistent volume for cartridges. If persistent volume is enabled, Apache Stratos automatically attaches a volume when a new cartridge instance is created. For more information, see Persistence Volume Mapping.

- Gracefully shutdown instances
  Before terminating an instance, when scaling down, the Auto-scaler will allow all the existing requests to the instance to gracefully shutdown, and not accepting any new requests for that instance.
- MQTT and AMQP support
- Stratos Mock IaaS
  Stratos mock IaaS simulates the basic features that Stratos requires from an IaaS. Thereby, this provides a cost effective way of trying out Stratos. For more information, see Apache Stratos Mock IaaS.

Related Links

- 4.1.x Apache Stratos Mock IaaS
- 4.1.x Cartridge
- 4.1.x CLI Tool
- 4.1.x Cloud Partitioning
- 4.1.x Composite Applications
- 4.1.x Load Balancer Extensions
- 4.1.x Logging
- 4.1.x Persistence Volume Mapping
- 4.1.x Smart Policies