ConPaaS User Manual

Ismail El Helw Guillaume Pierre

Version 1.0.0

Contents

1 Introduction

ConPaaS is an open-source runtime environment for hosting applications in the cloud which aims at offering the full power of the cloud to application developers while shielding them from the associated complexity of the cloud.

ConPaaS is designed to host both high-performance scientific applications and online Web applications. It runs on a variety of public and private clouds, and is easily extensible. ConPaaS automates the entire life-cycle of an application, including collaborative development, deployment, performance monitoring, and automatic scaling. This allows developers to focus their attention on application-specific concerns rather than on cloud-specific details.

ConPaaS is organized as a collection of **services**, where each service acts as a replacement for a commonly used runtime environment. For example, to replace a MySQL database, ConPaaS provides a cloud-based MySQL service which acts as a high-level database abstraction. The service uses real MySQL databases internally, and therefore makes it easy to port an cloud application to ConPaaS. Unlike a regular centralized database, however, it is self-managed and fully elastic: one can dynamically increase or decrease its processing capacity by requesting it to reconfigure itself with a different number of virtual machines.

ConPaaS currently contains six services:

- **Two Web hosting services** respectively specialized for hosting PHP and JSP applications;
- **MySQL database** service;
- **Scalarix service** offering a scalable in-memory key-value store;
- **MapReduce service** providing thewell-known high-performance computation framework;
- **TaskFarming service** high-performance batch processing.

ConPaaS applications can be composed of any number of services. For example, a bio-informatics application may make use of a PHP and a MySQL service to host a Web-based frontend, and link this frontend to a MapReduce backend service for conducting high-performance genomic computations on demand.

2 ConPaaS usage overview

Most operations in ConPaaS can be done using the ConPaaS frontend, which gives a Web-based interface to the system. The front-end allows users to register, create services, upload code and data to the services, and configure each service.

- The Dashboard page displays the list of services currently active in the system. Beware: each active service uses credits, even if it is in “stopped” state. To stop using credits you must terminate the services completely.

- Each service comes with a separate page which allows one to configure it, upload code and data, and scale it up and down.

All the functionalities of the frontend are also available using a command-line interface. This allows one to script commands for ConPaaS. The command-line interface also features additional advanced functionalities, which are not available using the front-end.

3 The PHP Web hosting service

The PHP Web hosting service is dedicated to hosting Web applications written in PHP. It can also host static Web content.

3.1 Basic usage with the front-end

Create a service. Click on "create new service", then select "PHP service". This operation starts a new "Manager" virtual machine. The manager is in charge of taking care of the service, but it does not host Web applications itself. To make use of the PHP service you need to start it.

Start a service. Click on "start", this will create a new virtual machine which can host your Web application.

Rename the service. By default all new PHP services are named "New PHP service." To give it a meaningful name, click on this name in the service-specific page and enter a new name.

Upload an application to the service. Applications can be uploaded in the form of a `tar` or `zip` archive. Attention: the archive must expand *in the current directory* rather than in a subdirectory. The service does not immediately use new applications when they are uploaded. The frontend shows the list of versions that have been uploaded; choose one version and click "make active" to activate it. Note that the frontend only allows uploading archives smaller than a certain size. To upload large archives, you must use the command-line tool (see Advanced usage).

Access your application. The frontend gives a link to the running application. This URL will remain valid as long as you do not stop the service.

Scale the service up and down. The front-end shows the list of virtual machine instances used to run a service. The minimum configuration to run a PHP application requires two instances: one manager instance, and one instance containing a load balancer, a web server and a PHP server. To scale the capacity up or down, click the numbers below to request adding or removing servers. The system reconfigures itself without any service interruption.

Stop the service. When you do not need to run the application any more, click “stop” to stop the service. This stops all instances except the manager which keeps on running. Beware that the manager still uses credit while it is running. To stop using credit you must terminate the service.

Terminate the service. Click “terminate” to terminate the service. At this point all the state of the service manager will be lost.

3.2 Advanced usage with the command-line tool

The PHP service can also be controled using the `cpsclient.web` script. This command-line tool can issue the same operations as the front-end, except creating a new service. It also has additional functionalities which are useful for advanced usage.

To use the `cpsclient.web` tool you must first set the PYTHONPATH variable to contain the full path name of the “src” subdirectory within the ConPaaS directory:

```
$ export PYTHONPATH=/path/to/conpaas/src
```

The `cpsclient.web` tool always takes the URL of the service manager as its first argument. This URL is provided by the front-end.

List all options of the command-line tool.

```
$ ./cpsclient.web help
```

Start the service.

```
$ ./cpsclient.web http://x-x-x-x/ startup
```

Stop the service.

```
$ ./cpsclient.web http://x-x-x-x/ shutdown
```

Upload a new version of the application

```
$ ./cpsclient.web http://x-x-x-x/ upload_code_version path/to/archive.zip
```

Note that this operation allows one to upload bigger archive files than the frontend.

Scale the service up and down.

```
$ ./cpsclient.web http://x-x-x-x/ add_nodes -h
Usage: add_nodes

Options:
  -h, --help            show this help message and exit
  -p PROXY, --proxy=PROXY
  -w WEB, --web=WEB
  -b BACKEND, --backend=BACKEND

$ ./cpsclient.web http://x-x-x-x/ add_nodes -w 1 -b 1

$ ./cpsclient.web http://x-x-x-x/ remove_nodes -h
Usage: remove_nodes

Options:
  -h, --help            show this help message and exit
  -p PROXY, --proxy=PROXY
  -w WEB, --web=WEB
  -b BACKEND, --backend=BACKEND

$ ./cpsclient.web http://x-x-x-x/ remove_nodes -w 1 -b 1
```

Set the service in debug mode. By default the PHP service does not display anything in case PHP errors occur while executing the application. This setting is useful for production, when you do not want to reveal internal information to external users. While developing an application it is however useful to let PHP display erors.

```
$ ./cpsclient.web http://x-x-x-x/ update_php_configuration \
  -p "display_errors=On"
```

4 The Java Web hosting service

The Java Web hosting service is dedicated to hosting Web applications written in Java using JSP or servlets. It can also host static Web content.

4.1 Basic usage with the front-end

Create a service. Click on “create new service”, then select “Java service”. This operation starts a new “Manager” virtual machine. The manager is in charge of taking care of the service, but it does not host Web applications itself. To make use of the Java service you need to start it.

Start a service. Click on “start”, this will create a new virtual machine which can host your Web application.

Rename the service. By default all new Java services are named "New Java service." To give it a meaningful name, click on this name in the service-specific page and enter a new name.

Upload an application to the service. Applications can be uploaded in the form of a `war` file. The service does not immediately use new applications when they are uploaded. The frontend shows the list of versions that have been uploaded; choose one version and click "make active" to activate it. Note that the frontend only allows uploading archives smaller than a certain size. To upload large archives, you must use the command-line tool (see Advanced usage).

Access your application. The frontend gives a link to the running application. This URL will remain valid as long as you do not stop the service.

Scale the service up and down. The front-end shows the list of virtual machine instances used to run a service. The minimum configuration to run a Java application requires two instances: one manager instance, and one instance containing a load balancer, a web server and a Java server. To scale the capacity up or down, click the numbers below to request adding or removing servers. The system reconfigures itself without any service interruption.

Stop the service. When you do not need to run the application any more, click "stop" to stop the service. This stops all instances except the manager which keeps on running. Beware that the manager still uses credit while it is running. To stop using credit you must terminate the service.

Terminate the service. Click "terminate" to terminate the service. At this point all the state of the service manager will be lost.

4.2 Advanced usage with the command-line tool

The Java service can also be controled using the `cpsclient.web` script. This command-line tool can issue the same operations as the front-end, except creating a new service. It also has additional functionalities which are useful for advanced usage.

To use the `cpsclient.web` tool you must first set the PYTHONPATH variable to contain the full path name of the "src" subdirectory within the ConPaaS directory:

```
$ export PYTHONPATH=/path/to/conpaas/src
```

The `cpsclient.web` tool always takes the URL of the service manager as its first argument. This URL is provided by the front-end.

List all options of the command-line tool.

```
$ ./cpsclient.web help
```

Start the service.

```
$ ./cpsclient.web http://x-x-x-x/ startup
```

Stop the service.

```
$ ./cpsclient.web http://x-x-x-x/ shutdown
```

Upload a new version of the application

```
$ ./cpsclient.web http://x-x-x-x/ upload_code_version path/to/archive.war
```

Note that this operation allows one to upload bigger archive files than the frontend.

Scale the service up and down.

```
$ ./cpsclient.web http://x-x-x-x/ add_nodes -h
Usage: add_nodes

Options:
  -h, --help            show this help message and exit
  -p PROXY, --proxy=PROXY
  -w WEB, --web=WEB
  -b BACKEND, --backend=BACKEND

$ ./cpsclient.web http://x-x-x-x/ add_nodes -w 1 -b 1

$ ./cpsclient.web http://x-x-x-x/ remove_nodes -h
Usage: remove_nodes

Options:
  -h, --help            show this help message and exit
  -p PROXY, --proxy=PROXY
  -w WEB, --web=WEB
  -b BACKEND, --backend=BACKEND

$ ./cpsclient.web http://x-x-x-x/ remove_nodes -w 1 -b 1
```

5 The MySQL database service

6 The Scalarix key-value store service

7 The MapReduce service

8 The TaskFarming service

9 Building new types of services

The architecture of ConPaaS allows developers to build new types of services. To learn how to do this, please check the “Internals” ConPaaS documentation.

10 About this document

INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.