

Heterogeneous Cloud Computing for Research

Guillaume Pierre

Managing Cloud Services with ConPaaS

For this hands-on session we will use the ConPaaS online installation that runs on Amazon EC2. You can access ConPaaS online at <https://online.conpaas.eu/>. Create an account by clicking on “register”, and log in.

Note: you will probably not have time to finish all the exercises :-(

In this session we have designed two different installation processes for the famous WordPress content management system (www.wordpress.org) in ConPaaS. We will first install WordPress manually using the ConPaaS services. Then we will set up WordPress using a pre-defined application manifest.

Exercise-1: WordPress is implemented in PHP using a MySQL database so we will need to create an application that uses a PHP and a MySQL service in ConPaaS.

1. Create a new application (or enter the “new application” provided by default).
2. Create a MySQL service, start it, reset its password. Copy the IP address of the master node somewhere, we will need it in step 4.
3. Create a PHP service, start it.
4. Download a Wordpress tarball from <http://www.wordpress.org>, and expand it in your computer.
5. Copy file `wordpress/wp-config-sample.php` to `wordpress/wp-config.php` and edit the `DB_NAME`, `DB_USER`, `DB_PASSWORD` and `DB_HOST` variables to point to the database service. You can choose any database name for the `DB_NAME` variable as long as it does not contain any special character. We will reuse the same name in step 6.
6. Rebuild a tarball of the directory such that it will expand *in the current directory* rather than in a wordpress subdirectory. Upload this tarball to the PHP service, and make the new version active.

7. Create a SQL script which creates a new database with the same name as in step. This script needs exactly one line (after replacing “databasename” with the actual name of your database):

```
CREATE DATABASE databasename;
```

Upload the script using the “load database from file” button of the MySQL service.

8. Open the page of the PHP service, and click “access application”. Your browser will redirect to a web page where you have to register because the application is not fully installed yet. Fill in the requested information (site name etc).
9. That’s it! The application should now work.

- **Question 1:** Why is it necessary to copy-paste database parameters in the PHP application in step 4? Which new feature could be provide in ConPaaS to make this step superfluous?
- **Question 2:** Add two more PHP instances in your PHP service. Then in WordPress, create a new post (in directory `/wp-admin/` within the application, click on “+New ⇒ Post’ in the top-side bar). In your new post, upload an image file by clicking on “Add Media”. Insert the image in your post, then save the post. View your post, and reload it several times. What is going on? Can you explain the reason why things do not work as expected, and what could be done to fix it?

Exercise-2: To facilitate and automatize the installation process, we will use a ready-made application manifest for WordPress in ConPaaS.

1. Stop the PHP and MySQL services deployed in the **Exercise-1**.
 2. Go to the main dashboard and click on “Deploy ready-made application”, then start WordPress.
 3. Open the page of the PHP service, and click “access application”. Your browser will redirect to your WordPress site which was pre-configured using “*admin*” and “*conpaas*” as username and password, respectively.
 4. Create and save a post as done in *Step-8* of the **Exercise-1**.
 5. Installation completed! The application should now work and you should be able to add images in your posts.
- **Question 3:** Check the XtreamFS website. What does it do, and what are its unique features compared to other cloud storage services?

- **Question 4:** How did ConPaaS address the problem with images in WordPress? Make a figure to show how the different components interact.
- **Question 5:** In the PHP service, read the content of the startup script which is executed every time a new instance is created. What does this script do, exactly?

— the end —