Administrator Manual: ConPaaS Web Hosting Service

August 15, 2011

1 Using Amazon Web Services

The Web Hosting Service is capable of running over the Elastic Compute Cloud (EC2) of Amazon Web Services (AWS). This section describes the process of configuring an AWS account to run the Web Hosting Service.

1.1 Create an EBS backed AMI on Amazon EC2

The Web Hosting Service requires the creation of an Amazon Machine Image(AMI) to contain the dependencies of it's processes. The easiest method of creating a new Elastic Block Store backed Amazon Machine Image is to start from an already existing one, customize it and save the resulting filesystem as a new AMI. The following steps explains how to setup an AMI using this methodology.

1. Search the public AMIs for a Debian squeeze EBS AMI and run an instance of it.
2. Download conpaas_web_deps script and run it inside the instance. This script will install all of the dependencies of the manager and agent processes as well as create the necessary directory structure.
3. Clean the filesystem by removing any temporary files you may have created.
4. Go to the EC2 administration page at the AWS website, right click on the running instance and click on "Create Image (EBS AMI)". AWS documentation is at http://docs.amazonwebservices.com/AWSEC2/latest/UserGuide/index.l

1.2 Create a Security Group

An AWS security group is an abstraction of a set of firewall rules to limit inbound traffic. The default policy of a new group is to deny all inbound traffic.

Therefore, one needs to specify a whitelist of protocols and destination ports that are accesible from the outside. The Web Hosting Service uses TCP ports 80, 8080 and 9000. All three ports should be open for all running instances. AWS documentation is at http://docs.amazonwebservices.com/AWSEC2/latest/UserGuide/index.l network-security.html.

2 Setup ConPaaS Front-end

The ConPaaS Front-end is a web application that allows users to manager their ConPaaS services. Users can create, configure and terminate services through it. The Front-end supports running ConPaaS over AWS only. This section describes the process of setting up a ConPaaS Front-end.

2.1 Create a MySQL Database

The ConPaaS frontend uses a MySQL database to store data about users and their services. The script located in "frontend/scripts/frontend-db.sql" creates a new user DB_USER with password DB_PASSWD and a database DB_NAME. It grants all access permissions to user DB_USER on the new database. Finally, it creates the database schema. Update the first four lines to change DB_USER, DB_PASSWD and DB_NAME to reasonable values.

You can create the database schema using this command:

```
mysql -u ADMIN -p ADMINPASSWORD  < frontend-db.sql
```

2.2 Configure the Front-end

The ConPaaS Front-end code is a collection of PHP scripts. It can run on any PHP-enabled Web server. The following instructions detail the configuration of the Front-end.

1. Place the PHP code found in directory "frontend/www" at the document root of the target web server such that the file named __init__.php is directly underneath it.

2. Edit the CONF_DIR variable in __init__.php such that it points to the configuration directory path.

3. Copy all files from the "frontend/conf" directory to a location *outside* of the Web server's document root. A good location could be for example /etc/conpaas. Edit each of these files to setup the required configuration parameters. Each variable should be described in the config file itself.

4. Download the AWS sdk for PHP from http://aws.amazon.com/sdkforphp/. Extract the sdk directory and rename it to aws-sdk. Place it under the lib directory of the front-end source code such that lib/aws-sdk/ contains a file named config-sample.inc.php.

5. Rename lib/aws-sdk/config-sample.inc.php to lib/aws-sdk/config.inc.php and fill in AWS_KEY, AWS_SECRET_KEY, AWS_ACCOUNT_ID and AWS_CANONICAL_ID as instructed in the file's documentation.

At this point, your front-end should be working!

3 Sandboxing

The default ConPaaS configuration creates strong snadboxing so that applications cannot open sockets, access the file system, execute commands, etc. This makes the platform relatively secure against malicious applications. On the other hand, it strongly restricts the actions that ConPaaS applications can do. To reduce these security measures to a more usable level, you need to edit two files:

- To change restrictions applied to PHP applications, edit file "web-servers/etc/fpm.tmpl" to change the list of "disable_functions". Do not forget to recreate a file ConPaaSWeb.tar.gz out of the entire "web-servers" directory, and to copy it at the URL specified in file "frontend/conf/manager-user-data".

- To change restrictions applied to Java applications, edit file "web-servers/etc/tomcat-catalina.policy". Do not forget to recreate a file ConPaaSWeb.tar.gz out of the entire "web-servers" directory, and to copy it at the URL specified in file "frontend/conf/manager-user-data".