CHOReOS Enactment Engine Installation Guide

Leonardo Leite (IME - USP)

February 20, 2013

1 Introduction

The CHOReOS Enactment Engine provides a Platform as a Service (PaaS) that automates the distributed deployment of service choreographies in cloud environments. This document provides instructions about how to install, configure, and run the Enactment Engine.

The Enactment Engine is composed by the components pictured on Figure 1, from which Choreography Deployer and Deployment Manager are provided by the Enactment Engine, whereas the Chef components and the Cloud Gateway are third-party used by the Enactment Engine.

In this guide we assume that both Choreography Deployer and Deployment Manager components will be executed on the deployer machine. Deployer is the human operator responsible by the deployment process. For while, a Choreography Deployer instance is limited to use only one Deployment Manager.

2 Requirements

Before you run Enactment Engine, you will need:

- SVN;
- Java 6 (we are using OpenJDK);
- Maven 3 (`http://maven.apache.org/download.html`);
- a Cloud Gateway access, as detailed in Section 3;
- a Chef account, as detailed in Section 4.

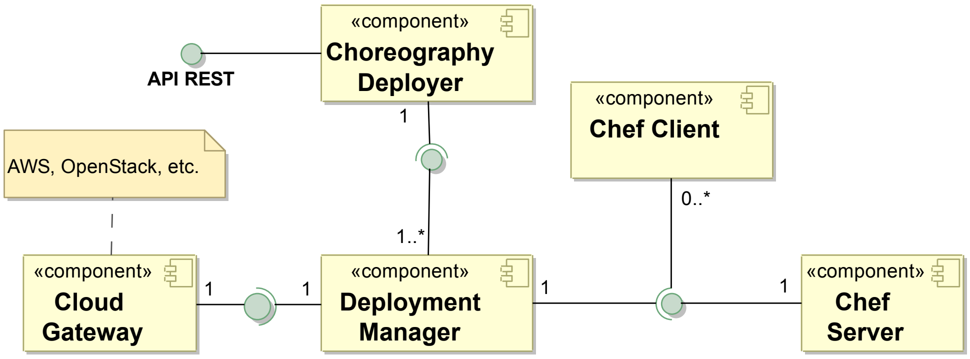

Figure 1: Enactment Engine architecture

3 Cloud Gateway

3.1 Amazon EC2

3.2 OpenStack

3.3 Fixed cloud provider

4 Chef

Deployment Manager will properly configure the nodes using Chef, an open-source configuration management system. With Chef you can specify a resources set to be deployed into cloud nodes. These resources are described by a Ruby-like DSL (Domain Specific Language), and include: systems, files, scripts execution, and others.

You can setup your own Chef server, or create an Hosted Chef server account. Hosted Chef is offered by Opscode[1]. Although Hosted Chef frees you from setting the Chef Server in the infrastructure of your organization, it allows only a limited number of nodes to be managed by Chef.

Make sure your `knife.rb` file be something like the example on Listing 1.

Listing 1: knife.rb example

```
%log_level                :info
%log_location             STDOUT
%node_name                "lleite"
%client_key               "#{current_dir}/lleite.pem"
%validation_client_name   "choreos-verao-validator"
%validation_key           "#{current_dir}/choreos-verao-validator.pem"
%chef_server_url          "https://api.opscode.com/organizations/choreos-verao"
%cache_type               'BasicFile'
%cache_options( :path => "#{ENV['HOME']}/.chef/checksums" )
%cookbook_path            ["#{current_dir}/../cookbooks"]
```

Here, "lleite" is also my Opscode user name, and "choreos-verao" is the organization name configured on Hosted Chef. And don't forget the "cookbook_path" property.

You will also need to upload to your Chef Server all the cookbooks from our cookbook folder: https://github.com/choreos/choreos_middleware/tree/master/chef-repo/cookbooks.

5 Checkout and Compilation

To checkout the code: `svn checkout http://ow2.forge ...`

After installing Maven 3, open the terminal at the `choreos_middleware` folder, and run the `build.sh` script. It can take several minutes. Internet access is necessary during compilation.

6 Configuration

Open the folder `DeploymentManager/src/main/resources`, and create a `deployment.properties` file by copying the `deployment.properties.template` file. The new properties file must be created in the same folder.

Open the just created properties file and edit it following instructions on the template file. The Listing 2 shows an example.

Listing 2: deployment.properties example

```
NODE_POOL_MANAGER_PORT=9100
SERVICE_DEPLOYER_PORT=9101
NODE_SELECTOR=ROUND_ROBIN
CLOUD_PROVIDER=AWS
FIXED_VM_IP=192.168.56.102
FIXED_VM_HOSTNAME=choreos-node
FIXED_VM_PRIVATE_SSH_KEY=/home/leonardo/.ssh/nopass
FIXED_VM_USER=choreos
AMAZON_ACCESS_KEY_ID=AKIAIIT213ISasdSECRETEFJH6Q
AMAZON_SECRET_KEY=N+KzHQITasdIS123ALSOwzAj9MiSECRETE0UPuwyD
```

[1] `http://www.opscode.com/`

```
AMAZON_KEY_PAIR=leofl
AMAZON_PRIVATE_SSH_KEY=/home/leonardo/.ssh/leofl.pem
CHEF_CONFIG_FILE=/home/leonardo/chef/chef-repo/.chef/knife.rb
CHEF_REPO=/home/leonardo/chef/chef-repo
```

NODE_SELECTOR options...

7 Execution

After compiling the project, to run both Choreography Deployer and Deployment Manager you have just to run the main method, on the following classes: org.ow2.choreos.deployment.rest.DeploymentManagerServer and org.ow2.choreos.chors.rest.ChorDeployerServer.

This task can be easier accomplished if you import the Enactment Engine projects in the Eclipse IDE. After importing the project, open the menu `Window>>Preferences>>Java>>Build Path>>Classpath` variables, and set the `M2_REPO` variable pointing to your Maven repository folder, usually the `.m2/repository` folder within your home folder. Obs: we have used the Eclipse Indigo version.

If you successfully start the Choreography Deployer and the Deployment Manager you must see the following messages on their respective consoles:

```
Choreography Deployer has started [http://localhost:9100/choreographydeployer/]
Deployment Manager has started [http://localhost:9101/deploymentmanager/]
```

To verify if it is everything OK, run the org.ow2.choreos.chors.SimpleChorEnactmentTest. This test will deploy a simple choreography composed of two services and try to invoke it.