ENSIMAG Training

Document Contents

This document purpose is to describe the training delivered to ENSIMAG IS students

Document writer

Bruno.Cornec@hpe.com

Table of content

Overview of the Assessment

Objectives

The goal of this document is to describe the use case that will be used for the training of ENSIMAG students

Reference documents

The main **OpenStack** entry point is at http://www.openstack.org

v1.0:   2016-09-29 – First version

v1.1:   2016-11-28 – Update with grade per part

Customer Story

Objectives

The goal of this training is to realize a promotional lottery for an e-commerce site

Details

Customer of a big e-commerce site will received a promotional email with a link to earn a price if they are lucky. The application should detect whether the player already played or not, and whether he won already or not. Each status should be kept with the date when it was performed. The application will provide a button allowing the customer to play, in case he didn't already, and the result of the computation which will happen behind the scene will give to the customer the nature of the article he has won, and display the corresponding image in the interface. Mails will be sent to admins when a winner is found.

Students will first have to automate the management of a virtual infrastructure (Infrastructure as code concept). Within this part we are using Openstack as our Infrastructure As A Service platform but this is an implementation choice. The same principles can be implemented on top of another platform like Eucalyptus, Cloudstack, Amazon AWS, Azure, Docker. The HPE team will provide two physical infrastructures for that purpose, one which will be pre-installed with Helion OpenStack in which a tenant per group of student will be available for application development, and another one (set of 15 servers) which will be pre-installed with just a CentOS 7 Operating System. Students will configure the IaaS part on top of it to study the various components that they want to test (**this part will be evaluated on 4 points**).

Then they will have to create a cloud native application to support the marketing activity. That application will consist of:

- 1 Web page with 5 parts/micro-services: I, S, B, W and P.

- I(dentification) service: receives http request from customer (link with customer ID) and look for it into DB (script to build it provided separately by HPE team).

- S(tatus) service: detect whether customer already played or not, status stored in the DB. (DB choice left to students - HA to be seen). Can have a message bus to buffer requests.

- B(utton) service: button widget allowing the customer to play. Only when not already done.

- W(orker) service that computes whether the customer won or not (slow service provided by the HPE team with a REST API interface), called by B. If won, post an image representing what has been won into OpenStack Swift with customer ID. Then post by e-mail via an external provider a message to admins (using a message bus if possible). Button is gray if the customer has already played.

- P(icture) service: Look into Swift with customer ID to display the image of the customer, empty if no image.

This part will be evaluated on 6 points

The content provided by the HPE team will be referenced on the etherpad used to follow the exchanges on the project (https://etherpad.openstack.org/p/ENSIMAG_2016)

Goals & constraints

The app must have the following goals & constraints:

1. IaaS platform chosen is OpenStack. The install has to provide the following services: Nova, Glance, Keystone, Cinder, Swift, Heat, Neutron. One tenant per group has to be created. 2 users (user, account used for automation and admin) have to be created per tenant.

2. The W micro-service is a third party software (provided by HPE team) and cannot be changed by the students.

3. Each part of the web page has to be implemented as a micro-service

4. Using automation, students should be able to redeploy their application on another tenant or another OpenStack instance. (Think to initial conditions, some steps can still remain manual)

5. DB should be persistent with regards to VMs failures.

6. DB and W service should be on a separate private network

7. Application should be publicly available on the external network. Only http[s] (ports 80 and 443) will be available from outside.

8. Application should support nicely the death of any one of the 5 micro-services. (Page should still be displayed anyway, printing N/A is OK, having an error isn't). As well it should be scalable in case of insufficient resources

9. All materials should be kept on public git repository (github e.g.) with an Open Source license (Cf: https://opensource.org/licenses prefer the popular ones). App should be deployed from Git up to the infra, with either ansible, heat, or any other automation tool/script

This part will be evaluated on 10 points (automation will be particularly important)

Bonus goals and constraints

In addition, The app may have the following goals & constraints:

1. The DB info should resist to failure and not loose records (Using a message bus e.g. to handle failures)

2. The application provided has some latency. Use OpenStack to scale it up (msg bus/ha proxy) in order to be able to exceed the 4 reqs on a lapse of 10 seconds. A LBaaS (Load Balancer as a Service) can also be used and provided as part of the solution.

3. Monitoring, Security, Error case, Full HA, Full install/conf automation, Automatic Scalability

This part will be evaluated on 4 additional bonus points

Deliverables

- On Gihub:
    - Heat template/ansible playbooks/scripts for Infra group
    - Performance and tests results
    - Application code
- Application design document
- Present the (almost !) automatic deployment of the application in a new empty tenant and make reliability checks.

HPE support

HPE will provide a remote access to the pre-production platform (set of servers preinstalled with a CentOS 7 distribution) that the IaaS team will have to setup from an OpenStack perspective and deliver so the DevOps teams could deploy their application on it. HPE will also provide a production platform running Helion OpenStack as a development environment for the application developers for testing purposes.

A mailing list has been created to provide remote support to students during the whole period. Various people from HPE will so be available to answer questions and help with regards to platform setup and access. The mailing-list address is ensimag-openstack@lists.osp.hpe.com

Agenda

Lundi 21 novembre : 14h00-17h00

- Project explanation: Overall Goals & method (groups, prod platform, TD systems for tests). The goal is to provide a cloud native application, providing redundancy and scalability, using an OpenStack infrastructure. No formal solution will be directly given, students will have to build the solution by themselves. Many approaches are possible. The HPE team role will be after the 2 first sessions and generic presentations on all concepts to help them in the realization of that application and its setup.
- Cloud fundamentals: IaaS (Bruno)
- OpenStack architecture & example (Bruno)
- DevStack installation - stackrc + env var - image shared between groups (on ENSIMAG systems - nested with prepared VM or BM in B10) (Jérôme - Alexis - René)
- Production platform exaplanation (Bruno)
- Home work:
    - Continue to explore OpenStack from both UI & CLI
    - Create 9 groups of 5 students (one country per group), assign specialization (ops, devs, middleware, ... + backup), tool choice left to students, but licensing should be correct (prefer Open Source)

Mardi 29 novembre : 15h30-18h30

- Cloud fundamentals: DevOps (Christophe)

- 12 factors apps - micro services (Nicolas)

- Infrastructure as Code: OpenStack API as TD

- Project explanation: Architecture of the use case - Specifications - Design Constraints & Goals (GitHub, automation,)

- Home work: Continue to explore OpenStack API (Dev), Start Prod Infra setup

Mardi 6 décembre : 15h30-18h30

- Prod infra setup continued: Ability to launch a VM from an image using a heat template, attached to a network and a storage, and an object storage. private and public net are available and a floating IP attached to the VM.

- Application architecture done: microservices identified, HA solved, Scalability solved. Design done (Paper work, UML ?).

Mardi 13 décembre : 15h30-18h30

- Prod infra setup finished: Ability to launch a VM from an image using a heat template, attached to a network and a storage, and an object storage. private and public net are available and a floating IP attached to the VM.

- Application development ongoing

Mardi 3 janvier : 15h30-18h30

- Prod infra setup review if needed

- Application development ongoing

Mardi 10 janvier : 15h30-18h30

- Prod infra setup review if needed

- Application development done

Mardi 17 janvier : 15h30-18h30

- Synthesis and integration

- Group Presentation

-

Mardi 24 janvier (TBC) evaluation of the projects