CNSM4 Distributed & Cloud Computing: Assignment 1 - 10%

Due: Wednesday 25th January 2016 @ 12:55pm

John Jennings

LIT

ZeroMQ Chat Server

Introduction

This assignment tests your ability to create a set of Vagrant managed VMs running Python based client/server application that uses the ZeroMQ messaging library to pass string messages.

This project should be contained in a Vagrant based development environment as described below and also needs to be hosted in a private Git repository on http://www.bitbucket.org.

Requirements

Project Setup

- You are required to create a project folder called "K00000000_CNSM4_DSCC_Ass01_2016".
- This folder should contain a Vagrant File and all the Python and other code that you have developed.
- You are also required to publish your project to a private Git based repository in https://bitbucket.org/ and give full access to your Lecturer (https://bitbucket.org/johnfjennings/). The project must be named K00000000_CNSM4_Ass01
- You must include a **README.md** file in your project folder that includes:
  - your student and course details
  - a description of the project and instructions for running your code
  - The content of this file should be targetted at users that have no knowledge of this course or you.
  - This file should be formatted in the Markdown format (see https://confluence.atlassian.com/display/BITBUCKET/_MarkdownOverview)

Vagrant Specification

General

- You are required to build your own Vagrantfile configuration to achieve the following specification:
  - Create a multi-machine[1] configuration
    - One VM should be the **server**
    - One VM should be the **client**
  - The VMs should be based on the **ubuntu/trusty32** box
  - The following packages should be installed if not already pre-installed:
    - Apache2

[1] https://www.vagrantup.com/docs/multi-machine/

        - Nginx
        - PHP5
        - MySQL
        - Python 3.4
        - ZeroMQ

Networking

```
- Create a private network for the 2 VMs and configure as required
    - Set server static IP to 192.168.101.1
    - Set server static IP to 192.168.101.10
```

Git Repository

- You are required to commit your code at least once a week but there will be more marks awarded for more frequent, well commented commits of significant updates.

Server Specification

- You are asked to create a server application that complies with the following specification:

    1. Name one file **server.py**
    2. The server should use a **ZeroMQ** socket listening on port 3456.
    3. The server process should accept a string message from the client
    4. The server process must then append the IP address of the client to the message.
    5. The server process must then append the current time as a timestamp to this message. The current time must be correct and if possible have been synchronized with an external NTP source
    6. The transformed message with IP and timestamp must then be appended to a plain text file called **messagestore1.txt** on the server in the location **/vagrant/data/messagestore1.txt**.
    7. The transformed message with IP and timestamp must also be appended to an XML document called **messagestore2.xml** on the server in the location **/vagrant/data/messagestore2.xml**.
    8. You will be provided with a web local root folder from your lecturer that contains some html and javascript code
    9. You need to setup a web server using this local root folder and include/link your messagestore2.xml

Client Specification

- You are asked to create a client application that complies with the following specification:
    1. Name the file **send.py**
    2. The client application should be run from the client VM
    3. The client application should accept a single string parameter and send this string parameter to the client as a ZeroMQ message. Suggested usage is as follows:

```
$ python send.py --message="This is a sample message" --toip=192.168.100.1 --toport=3456
```

Submission

- Remember to name your project folder in the following convention, K00000000_CNSM4_DSCC_Ass01_2016 , remembering to replace K00000000 with your K number.
- Zip your project folder.
- Upload your project folder to Moodle before the due date and time. This is a backup of your work. The Git based project will the one that is graded

- Late submissions will incur a penalty of 5% deducted from your grade per day that you are late.