|  |  |  |
| --- | --- | --- |
| **Module Code**  **COMP5850M** | School of Computing  University of Leeds  **Coursework 1 - Report** | University of Leeds logo |

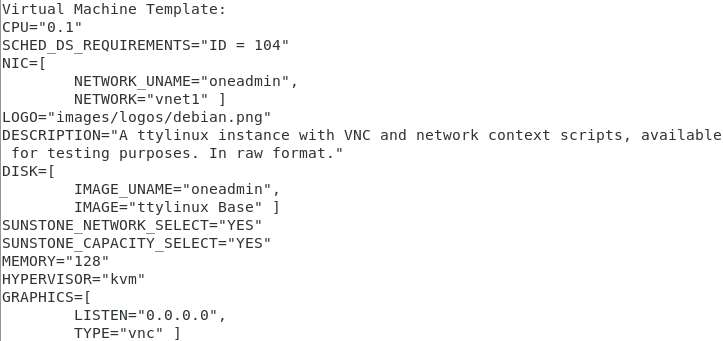
Full Name: Username:

Coursework Title: VIM Deadline Date: 09/03/2020

**Part 1: Java OpenNebula Cloud API (OCA) (10 marks)**

*Provide an explanation of the implementation of this task. The inclusion of the entire code is not required but you may include snippets if you wish.*

*VM template (1 mark)*

**

*Information OpenNebula provides about the VM (1 mark)*



*Measure the time it takes to instantiate/delete the VM. To get these measurements you are expected to run the experiments n times (e.g. n = 5). A statistical analysis (average, standard deviation) is expected. (2 marks)*

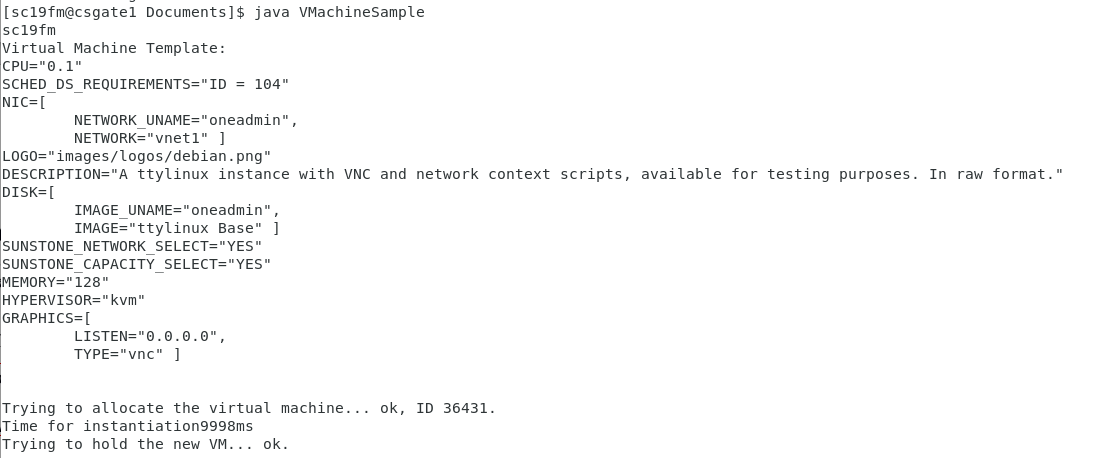
|  |  |  |
| --- | --- | --- |
| **Run No.** | **VM instantiation time** | **VM deletion time** |
| **1** | 8934 ms | 157 ms |
| **2** | 8486 ms | 128 ms |
| **3** | 9998 ms | 126 ms |
| **4** | 11203 ms | 135 ms |
| **5** | 8111 ms | 167 ms |
| **Average** | 9346.4 ms | 142.6 ms |
| **Standard Deviation** | 1123.22 ms | 16.43 ms |

*Explain how you have obtained these measurements (2 marks)*

Use currentTimeMillis() method to get the current system time (startTime) before instantiation a VM. Then I write a loop, keep checking the vm’s status. Until the vm is up running, break the loop. Then use the same method to get the time (endTime) after instantiation. And the instantiation time is the endTime minus the startTime.

Then use the same way to get the VM deletion time. Set startTimeDelete before delete. Set endTimeDelete after finalizeVM() method and make sure there is nothing wrong. Finally use these two numbers to calculate the deletion time.

*Evidence of successful run, e.g. screenshot (4 marks)*





**Part 2: VM Migration (15 marks)**

*Provide an explanation of the implementation of this task. The inclusion of the entire code is not required but you may include snippets if you wish.*

*Requirements (2 marks)*

*Solution Design (2 marks)*

*Implementation (2 marks)*

*Measure the time it takes to migrate the VM. (2 marks)*

|  |  |
| --- | --- |
| **Run No.** | **VM migration time** |
| **1** |  |
| **2** |  |
| **3** |  |
| **4** |  |
| **5** |  |
| **Average** |  |
| **Standard Deviation** |  |

*Evidence of successful run, e.g. screenshot (3 marks)*

*Discussion of the results (4 marks)*

**Part 3: Resource Scaling and Performance/Energy Consumption Trade-Off**

**(10 - 25 marks, depending on application and challenge)**

*Details of the application considered (stress, MPI, Hadoop, other) (1-3 marks)*

*Design of the experiments (1-4 marks)*

*Implementation of the experiments (1-4 marks)*

*Discussion of results (3-10 marks))*

*Evidence of successful run, e.g. screenshot (4 marks)*

**Any other comments:**