

Computer Assignment II

Course: Applied Cloud Computing

Course-ID:

1TD265

Instructors:

Andreas Hellander, Andreas.Hellander@it.uu.se

Salman Toor, Salman.Toor@it.uu.se

Task 1

This task is to be completed by everyone and is a requirement for a passing grade on this assignment (1 point).

In this task you will familiarize yourself with the OpenStack command line clients (CLIs) and Python APIs for interacting with the IaaS layer.

Part 1: Install Ipython and IPython Notebook on your client (laptop), or, if this is not possible or if you prefer to use the computer lab rooms, you can start an Ubuntu VM in the SNIC Cloud and start your Notebook server on that VM.

Part 2: The IPython Notebook 'OpenStack-API-Lab2-PartI.ipynb' (available in the student portal) contains exercises and questions. Open the notebook and complete it, then export it to pdf and upload that in the student portal.

Part 3: In this task, you will do a simple performance comparison between the VM and the physical host. Access your VM and execute the following command:

```
time echo "scale=5000; a(1)*4 " | bc -l
```

The command will calculate the value of PI upto 5000 digits. It is a CPU intensive task. Record the time taken to execute the command. The time on the physical machine is 32.317 seconds.

Question: Explain your findings in part 3. Are VMs slower than the physical machine? If yes, explain the reason. Are there alternative to VMs? How do you compare them with VMs? Keep the answer fairly short, limit it to a few sentences, max ¼ of a page.

Task 2

This task is not a requirement to pass the assignment, but can give up to 1 point, counting towards higher grades (see grading table).

In this task, you will build on your experiences in Task 1 and extend the MOLNs software (mentioned in Lecture 1 and 3) with some more functionality for OpenStack Providers.

Part 1:

Set up a GitHub account and 'fork' [www.github.com/ahellander/molns](https://github.com/ahellander/molns) to your account. Then clone the repo to your local client. Check out a new branch (call it 'lab2_assignment'), and follow the instructions in the README (look on the github page of the repo) to configure a provider for SMOG cloud, and then setup a controller. Start a MOLNs controller and access the IPython Notebook frontend.

Part 2: 'molns help' will show a list of the available commands. Your task is now to extend molnsclient (the CLI is implemented in molns.py) with the following commands (only implement it for the OpenStackProvider):

molns controller restart

Should restart the controller.

In the current implementation, MOLNs always allocates a new IP from the floating IP pool when it starts a new cluster controller or worker VM. But in OpenStack, there might already be IPs available to the Tenant. See this issue:

<https://github.com/Molns/molns/issues/28>

Investigate, by looking in the OpenStack APIs, if it is possible to fix this problem, and if, so provide an implementation.

After completing the above two tasks, send a pull request back to the the original repository, name it in such a way that we can easily see your First and Last name.

Note that this is mainly an exercise in understanding the basic APIs. By reading through some of the implementations in MolnsLib/OpenStackProvider.py, you will see more example of API usage. We will most likely not merge your PR.

Part3: Cloud interoperability is needed to expand applications across the boundaries of a single cloud infrastructure, for example in combined use of private and public clouds. It allows scalability and high availability for cloud applications. Write a ~1 page mini-essay where you highlight some of the key issues that can hinder cloud interoperability and also suggest possible solutions to those issues. Cite the references you build on in your discussion. We have posted a few articles in the student portal to get you started, but you can and should seek additional sources.