



Cloud Computing

Summer Term 2017

{Anton.Gulenko | Marcel.Wallschlaeger}@tu-berlin.de

Project Assignment No. 2

Due: 02.07.2017 23:59

This assignment focuses on two major topics: benchmarking the cloud operation itself, and using a cloud-based orchestration engine to manage cloud resources.

Important notes:

- Work in the same groups as for assignment 1.
- Use the same cit-cloud access credentials from assignment 1.
- For this assignment you will not need Amazon AWS. Please make sure all your VMs are shut down or deleted. You can delete your AWS account entirely to ensure that your credit card will not accidentally be charged in the future.

1. Cloud Benchmark

Benchmark the OpenStack cloud provided by the CIT department. In contrast to assignment 1, this benchmark should cover the cloud API operations themselves, not the VM resources. Use the `openrc` file from assignment 1 and make sure you deploy your virtual machines within the "Cloud Computing 2017" availability zone. When designing the benchmark, take the quotas of your OpenStack project into account. The benchmark should cover the following scenarios:

- Querying the list of your VMs (Command: `openstack server list`)
- Creation time of VMs (including the boot time)

Note: the first scenario can also return an empty list.

Each scenario should be executed in three different time slots (day, evening, night).

The OpenStack landscape provides a number of tools for cloud benchmarks, like [Rally](#). Pick one benchmarking tool and describe the methodology of your benchmark. Alternatively, you can design your own benchmark based completely on the OpenStack CLI tools. Plot and describe your results. Include any code (scripts, YAML files, etc.) you used for your benchmark.

2. Introducing Heat

This task introduces the OpenStack orchestration engine [Heat](#). Heat is an example for the [infrastructure-as-code](#) paradigm.

Tasks:

- Make yourself familiar with Heat, the Heat Orchestration Template language (HOT), and the command-line tools necessary to manage Heat stacks.

- Download the file `server.yml` provided on ISIS. This HOT file launches a single virtual machine within your OpenStack project using the given parameters.
- Instantiate a Heat stack using the `server.yml` template.
- Test the created VM by assigning a floating IP to it, connecting to it via SSH and testing its internet connectivity.
- Delete the instantiated stack and check that the VM has been deleted.
- Keep a listing of all commands you execute for the tasks above, as they are part of your submission.

3. Advanced Heat Templates

In this task you will extend the provided `server.yml` template to create a more complex scenario. Create a Heat template called `server-landscape.yml` that creates the resources as described below. This Heat template will be base for the third assignment. Therefore, there is no specific software to be installed on the virtual machines.

Network:

- One network
- One underlying subnet
- One router connecting the private network with the public `tu-internal` network

One frontend server:

- Instantiate the `server.yml` as a nested stack. Use Heats [template composition](#) feature.
- Flavor: Cloud Computing
- Availability Zone: Cloud Computing 2017
- Image: ubuntu-16.04
- Security Group: a custom security group managed by Heat that opens the port 80 and 22 to the outside world
- Connect the server to the new subnet
- Associate a FloatingIP
 - The resulting IP address must be available as an output variable named `floating_ip`.

Two equal backend servers:

- Instantiate the `server.yml` as a nested stack (see frontend server). Use Heats [Resource Group](#) feature to create two equal instances.
- Flavor, Availability Zone, Image: see frontend server
- Connect the servers to the new subnet
- The backend servers do NOT have an associated floating IP and should have no traffic allowed from outside networks. You can log into the backend servers by first logging into your frontend server. You will need to place your private SSH key on the frontend server in order to connect to the backends (or check out the `-A` flag of the `ssh` program).

Tasks:

- Instantiate a Heat stack using your `server-landscape.yml` template (one invocation of the `heat` command).

- Extract the `floating_ip` output variable of your stack using a second `heat` command.
- Test the created VMs: log into the frontend server and test its internet connectivity. From there, log into the two backend servers and test their internet connectivity, respectively.
- Keep a listing of all commands you execute for the tasks above, as they are part of your submission.

3. Submission Deliverables

Your submission on ISIS must be a single .pdf file with the following contents:

- Detailed description of your cloud benchmarking methodology, including any scripts or other code (*10 points*)
- Benchmarking results of the six different combinations of scenarios and time slots, including plots and interpretation of the results (*12 points*)
- Commented listing of commands you executed for Task 2 (*5 points*)
- The contents of your `server-landscape.yml` file (*15 points*)
- Commented listing of commands you executed to test your advanced Heat template (*5 points*)

Total points: 47