

# Report: VM Contextualization

Group 16: You Hu 12036234, Wenchen Lai 12482625

OpenNebula is a tool for private clouds deployment. Its API allows us to manage virtual resources. The command `“onevnet list”` shows available virtual networks with ID. The command `“oneimage list”` shows available images with ID. In this assignment, we mainly use `“onevm”` command to manage our virtual machines (VMs). We create a VM based on a VM template file using `“onevm create [VM_template_file]”`. The VM state will change from pending, to prolog, to boot, to running. The command `“onevm list”` can list existing VM with ID and show its state. The IP address of a VM can be found by `“onevm show [VM_ID]”`. The VM states need to turn to running before we log into our VM. We also can stop our VM using `“onevm poweroff [VM_ID]”` and delete our VM using `“onevm shutdown [VM_ID]”`.

VM Contextualization enables us to send some information to a VM at boot time. This information includes network configuration, user credentials and execute scripts. When a new VM is created, it will be configured based on this information. In OpenNebula, VM Contextualization can be done by CONTEXT section in VM template file.

Our task is to contextualize a VM that it can deploy the URL Shortener Web Service automatically at the start up. We create a VM template file (ubuntu-disk.one). The SSH\_PUBLIC\_KEY in CONTEXT section is changed to our own key and the IMAGE\_ID in DISK section is changed to an image id that we can use. Then we can create a VM and log into this VM through ssh. We deploy our service in this VM and call our service outside this VM. All needed commands for setting up the environment and starting our service are written into a script file (start\_script.sh). After that, we shut down this VM and modify the CONTEXT section in our VM template file again. The paths of the script file and our service implementation file (app.py) are added to FILES parameter. The command to execute the script file is added to START\_SCRIPT parameter. Then we can use this new VM template to create a VM, and test our service.

We encountered two problems in this assignment. The first one is that we cannot install python-pip when using the default image centos 5. We changed the image to ubuntu to avoid this problem. Another problem is that we cannot access to our service outside the VM. We logged in the VM and checked the networking. We found that when Flask runs on debug mode, the server is only accessible from the VM, not from any other in the network. Thus the debug parameter should be disabled and the host ip address is changed to 0.0.0.0. Then we should allow the traffic using `“iptables -A INPUT -p tcp --dport 5000 -j ACCEPT”`. After that, our problems are solved.