Report - Assignment 2.1 *(Deploying the URL Shortener in a VM)*

Experience using ONE API

Onevm is a very flexible tool and quite easy to use. By creating the template file, it was straightforward to define which files should be used in the VM as well the initialization script that should be used.

One of the aspects that we found not so good while using onevm is the fact that while the VM is being initialized/deployed is not possible to see what is happening "behind the scenes", deploying a simple VM without any special customization has little chance of failures, but when the template file becomes more complex and with more steps, in case of failures during the VM instantiation it's difficult to identify the cause of the issue. In order to circuvein this limitation the group decided to create one script to invoke the deployment script which has its standard output redirected to a log file (*sh /mnt/deployment.sh > /home/deployment.log*).

Contextualization of the Virtual Machine

For this assignment it was necessary to install Oracle JDK, Apache Tomcat and deploy the war. In order to do that, the tar.gz's containing the JDK and the Tomcat were added to the list of files in the template file as well as the .war file - this way once the VM is instantiate the files are ready to be used by the deployment script (please refer to section 'FILES' in the template file).

The final template file invokes the boot.sh script that mounts the filesystem and invokes the deployment.sh script which is responsible for: 1st - Installing the JDK; 2nd - Installing Tomcat; 3rd - Starting Tomcat; 4th Opening the firewall for the port 8080; 5th deploying the application (by moving the .war file to the webapps directory in the Tomcat installation folder) - please refer to the file deployment.sh for the whole list of actions/commands being performed.

Using the command onevm show <VM_Id> is possible to check the IP address associated to the Virtual Machine and with this information is possible to 'ssh' to the VM.

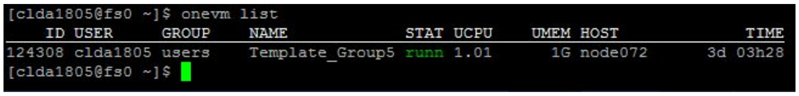
In order to test the VM, curl commands were used example:

POST (new url)

curl -X POST http://10.141.0.139:8080/Restul-UrlShortner/service/targetresource/www.google.com

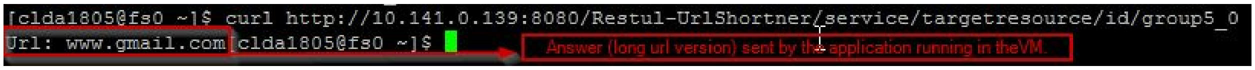
Normal GET (obtaining the long url version from the url short version):

curl http://localhost:8080/Restul-UrlShortner/service/targetresource/id/group5_0



```
[clda1805@fs0 ~]$ onevm list
    ID USER      GROUP    NAME            STAT UCPU    UMEM HOST             TIME
124308 clda1805 users    Template_Group5 runn 1.01      1G node072       3d 03h28
[clda1805@fs0 ~]$
```

VM running



```
[clda1805@fs0 ~]$ curl http://10.141.0.139:8080/Restul-UrlShortner/service/targetresource/id/group5_0
Url: www.gmail.com[clda1805@fs0 ~]$
```

Answer (long url version) sent by the application running in theVM.

Generating a request from DAS4 to the VM and obtaining an answer from the URL_Shortner Application "www.gmail.com"