CS454/554 - ASSIGNMENT #3 25 POINTS

TEAM MEMBERS: **TEAM NUMBER:** _____

_____

_____

_____

print only this coversheet for grading

PROJECT GOAL

The objective of this assignment is to interface with OpenStack's RESTful API to authenticate to the system and launch a VM. Mastery of these steps would enable one to develop software to manage and configure a cloud system without using the dashboard.

OVERVIEW

OpenStack consists of several service components that *listen* to specific ports on your installed devstack system. For example, on your manage node ip, port 8774 is the port that nova listens to. You would make calls to the nova API to create VMs. Port 5000 is the keystone port. Keystone manages authentication to your system. When you need to request a session token to allow you to interface with nova, you would send a request to keystone at that port. The steps to call the API functions are provided at the end of this document. The sections 13.4.1 and 13.4.2 in your textbook explain the format of the cURL commands used to interface with keystone and nova.

PROJECT REQUIREMENTS

Requirement	Completion (To Be Verified by Instructor or TA) *Grader Comments*
1. Obtain an Authentication Token and Retrieve your Project ID from devstack	*10 points*
2. Get an OS image and Flavor	*5 points*
3. Create a VM Instance	*5 points* *You should be able to go to your dashboard and see your new VM, if successful*
4. Login to your new instance using the IP netns commands from project 2	*5 points*

STEP 1: GET AN AUTHENTICATION TOKEN AND PROJECT ID

1. Log into your Linux node.
2. Copy over or create the get_token.json to your Manage node.
3. Update the information in the get_token.json with your information

```
{
    "auth": {
        "identity": {
            "methods": ["password"],
            "password": {
                "user": {
                    "name": "teamPUT_YOUR_TEAM_NUMBER+10_HERE",
                    "domain": {
                        "name": "default"
                    },
                    "password": "teamPUT_YOUR_ TEAM_NUMBER+10_HERE_pass"
                }
            }
        },
        "scope": {
            "project": {
                "name": "teamPUT_YOUR_TEAM_NUMBER+10_HERE",
                "domain": {
                    "name": "default"
}   }   }   }   }
```

get_token.json

4. Once you have the correct info in the json run the command:

curl -i -H "Content-Type: application/json" -d @get_token.json http://192.168.122.251:5000/v3/auth/tokens

5. This will return a large JSON file. The header will contain an Item called “X-Subject-Token” Copy that token into an environment variable in your current shell to use later.

   **The token will expire after one hour (default) so if you are having issues with authentication in the next steps, rerun the curl command to get a new token.*

6. The other item of interest in the JSON file is the project ID. You should see a line that looks similar to this:

"project": {"domain": {"id": "default", "name": "Default"}, "id": "de700e268c104347a0408ef1192d1dee", "name": "demo"}

That long string of characters is your PROJECT ID. This also needs to be copied into an environment variable for ease of use later.

STEP 2: GET AN OPERATING SYSTEM IMAGE AND FLAVOR TO USE WITH YOUR VM

7. Once you have your projects token and project id you must get a list of the images and flavors to use. To get a list of the Images run the following command:

curl -s http://192.168.122.251:8774/v2/$YOUR_PROJECT_ID_VARIABLE/images -X GET -H "X-Auth-Project-Id: admin" -H "Accept: application/json" -H "X-Auth-Token: $YOUR_TOKEN_VARIABLE" | python -m json.tool

This should all be on one line. The pipe into the python module will give you a better format for the output of this command.

8. The output of the last curl command will give you a list of images. We want to us the image named "cirros-0.3.4-x86_64-uec" Make a note of its ID.

9. To get a list of the flavors run the following command:

curl -s http://192.168.122.251:8774/v2/$YOUR_PROJECT_ID_VARIABLE/flavors -X GET -H "X-Auth-Project-Id: admin" -H "Accept: application/json" -H "X-Auth-Token: $YOUR_TOKEN_VARIABLE" | python -m json.tool

Again, this should all be on one line.

10. The output of the last curl command will give you a list of flavors. We want to us the flavor named "m1.tiny" Make a note of its ID.

STEP 3: CREATE A VM INSTANCE

11. Next we need a keypair to associate with the new instance. If you still have your key from the last project use it. Otherwise you can create a new one in the dashboard and copy it to your Manage node. [*OR! Look up the API command and use cURL to generate a new keypair*].

12. To create an instance we must set up a new JSON file that has all the information for the instance. Fill out the following template to use.

```
{
    "server": {
        "flavorRef": "FLAVOR_ID",
        "imageRef": "IMAGE_ID",
        "key_name": "NAME_OF_YOUR_KEYPAIR",
        "max_count": 1,
        "min_count": 1,
        "name": "ANY_NAME_TO_GIVE_YOUR_INSTANCE",
        "security_group": "default"
    }
}
```

13. Run the following command to create a new instance using the JSON file you just made.

curl -i http://192.168.122.251:8774/v2/$YOUR_PROJECT_ID_VARIABLE/servers -X POST -H "X-Auth-Project-Id: admin" -H "Content-Type: application/json" -H "Accept: application/json" -H "X-Auth-Token: $YOUR_TOKEN_VARIABLE" -d @create_instance.json

14. If the command runs successfully you will get back a JSON file containing the metadata of the Instance you just created.
15. Log into your dashboard to verify that the instance has been created and get the IP address of the new instance.

STEP 4: VERIFICATION

16. In browser go to http://192.168.122.251/dashboard/. Login credentials are the same as in step 3.
17. Verify the image and VM instance have been created correctly on CS Cloud.
18. **If you can successfully log into your new instance you have completed the project.**