SCHOOL OF ENGINEERING

VANDERBILT UNIVERSITY

Principles of Cloud Computing
Ansible Installation
Needed for PA2

CS 4287/5287

Dr. Vikash Singh

Credits to Dr. Aniruddha Gokhale of Vanderbilt University for many slides

After class demo

- We did many of the settings in this document while running ansible in our class demo.
- I still think you should verify all the setup discussed in the slides is in place before running cloud related playbooks.

- I have added many sample playbooks on Brightspace.
  - Many of these playbooks will help you get started on Cloud VM
  - For testing the automation on cloud VM (Chameleon cloud)
    - Make sure that you are not destroying the chameleon cloud VM yet.
  - SSH into your local VM from your host terminal.
  - Then run playbooks that connect to chameleon VMs using ansible-playbook command.

Summary of Needed Artifacts

- These instructions are for your manually created VirtualBox VM on your laptop where you are going to try out the different sample playbooks
  - The auto-generated VM thru Vagrant will also need similar artifacts for your Assignment #2
- Python version 3 and pip version 3
- Ansible version 2.13.x
- Python OpenStackSDK
- Ansible Galaxy openstack.cloud collection
- clouds.yaml file in a specific location
  - This is needed for OpenStack platform like Chameleon
  - AWS, GCP will need a different way

Ansible Installation

- Use either the sudo apt install or pip3 install approaches
- sudo apt install ansible
  - This approach will use the traditional approach to install ansible
  - On Ubuntu 18.04, Ansible may still use Python2 and so we may need to force it to use Python3

OR

- sudo -H pip3 install --upgrade ansible
  - Using Pip3 to install Ansible forces it to use Python3
  - Sudo will install it in systems folder but if you are using Python virtualenv, install it in your virtualenv
  - The -H flag was needed (as I found it) to overcome some warnings with ownership issues caused due to use of sudo
  - On Ubuntu 18.04, it may actually be worthwhile to do it this way which at the time of these slides installs Ansible version 2.10.x

Python OpenStack SDK

- Make sure that pip, setuptools and wheel are up to date

  sudo python3 -m pip install --upgrade pip setuptools wheel

- Python OpenStack SDK is a dependency that must be satisfied in order to use Ansible OpenStack modules
- Install it as (if installing in system folder)

  sudo python3 -m pip install --upgrade openstacksdk

OR

  sudo pip3 install --upgrade openstacksdk

- Otherwise install it in your virtualenv

Possible Annoying Issues on Ubuntu 18.04

- You shouldn’t get this issue in Ubuntu 20.04
- Your Ubuntu 18.04 may still include Python 2.x and /usr/bin/python may point to Python version 2
- So no matter how much we ask Ansible to use Python 3, it appears to discover /usr/bin/python and start using it (which becomes version 2)
- So we use the update-alternatives tool to set the version we want as shown below

```
gokhale@asg-ubuntu18:~$ python --version
Python 2.7.17
gokhale@asg-ubuntu18:~$ python3 --version
Python 3.6.9
gokhale@asg-ubuntu18:~$ sudo update-alternatives --install /usr/bin/python pytho
n /usr/bin/python2.7 1
update-alternatives: using /usr/bin/python2.7 to provide /usr/bin/python (python
) in auto mode
gokhale@asg-ubuntu18:~$ sudo update-alternatives --install /usr/bin/python pytho
n /usr/bin/python3.6 2
update-alternatives: using /usr/bin/python3.6 to provide /usr/bin/python (python
) in auto mode
gokhale@asg-ubuntu18:~$ sudo update-alternatives --set python /usr/bin/python3.6
gokhale@asg-ubuntu18:~$
gokhale@asg-ubuntu18:~$ which python
/usr/bin/python
gokhale@asg-ubuntu18:~$
gokhale@asg-ubuntu18:~$ python --version
Python 3.6.9
gokhale@asg-ubuntu18:~$
```

Ignoring Warnings on Ubuntu 18.04

- You may still see these warnings when executing your playbooks
- However, now that our /usr/bin/python points to version 3.6, we can ignore this warning

```
TASK [Gathering Facts] *********************************************************
[DEPRECATION WARNING]: Distribution Ubuntu 18.04 on host 127.0.0.1 should use
/usr/bin/python3, but is using /usr/bin/python for backward compatibility with
prior Ansible releases. A future Ansible release will default to using the
discovered platform python for this host. See https://docs.ansible.com/ansible/
2.10/reference_appendices/interpreter_discovery.html for more information. This
 feature will be removed in version 2.12. Deprecation warnings can be disabled
by setting deprecation_warnings=False in ansible.cfg.
ok: [127.0.0.1]
```

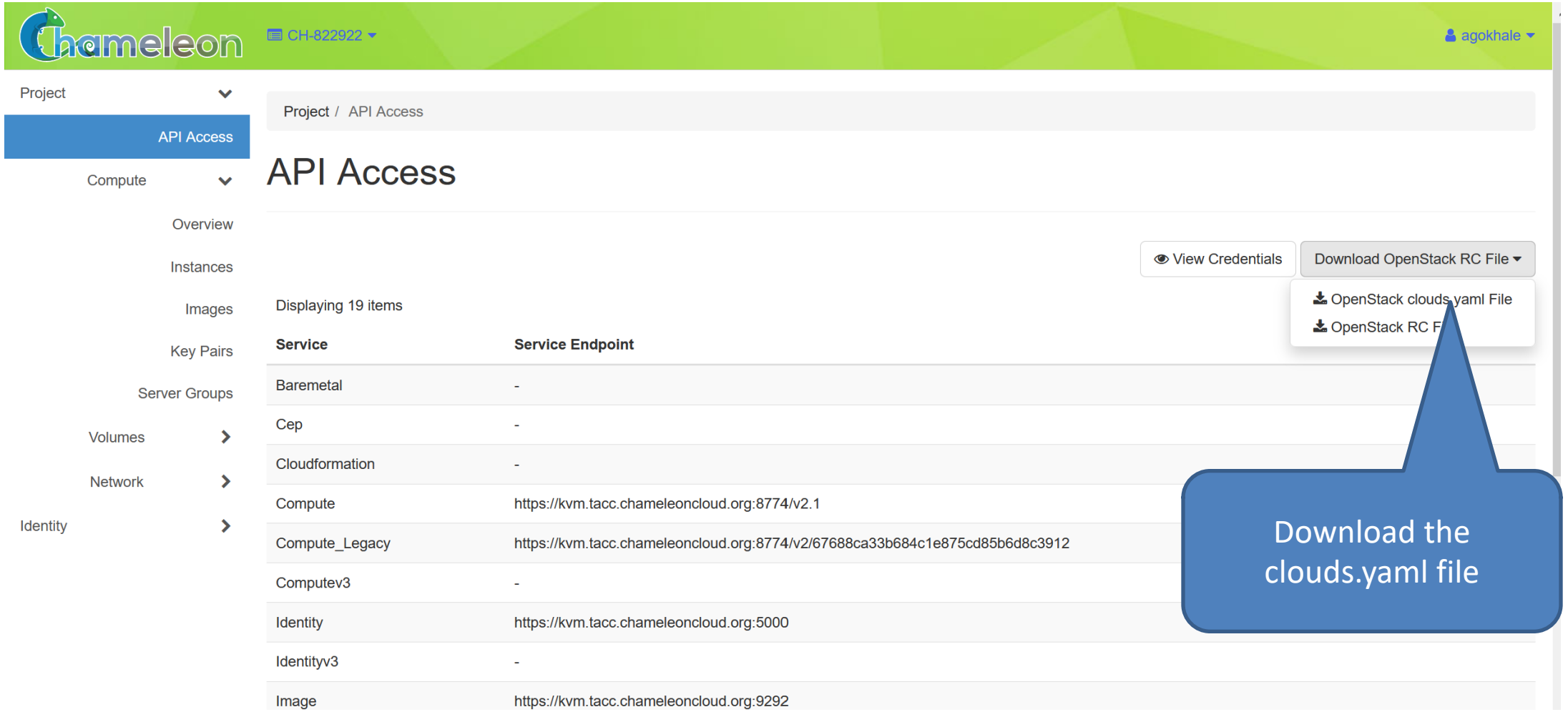
Installing openstack.cloud Collection

- Starting Ansible 2.9, the OpenStack module is now part of Ansible Galaxy collection as a plugin
- Playbooks will now include the desired collection which is akin to importing packages in Python
- Install as follows:

```
ansible-galaxy collection install openstack.cloud
```

- Note that the above step will install the plugin locally and not system-wide (use sudo if you want system-wide)
  - Local installation is fine for our purposes

You will need clouds YAML “auth” file. See next few slides.



clouds.yaml file

- The new approach now expects to see a clouds.yaml file in your ${HOME}/.config/openstack directory
- First create the directory (most likely the .config directory already exists else first create it and then the child directory called openstack)
  - mkdir ~/.config/openstack
  - Place the file called clouds.yaml we downloaded from Chameleon in this directory
  - The contents are shown on the next slide

Contents of clouds.yaml File

Notice the 4 levels of indentation here

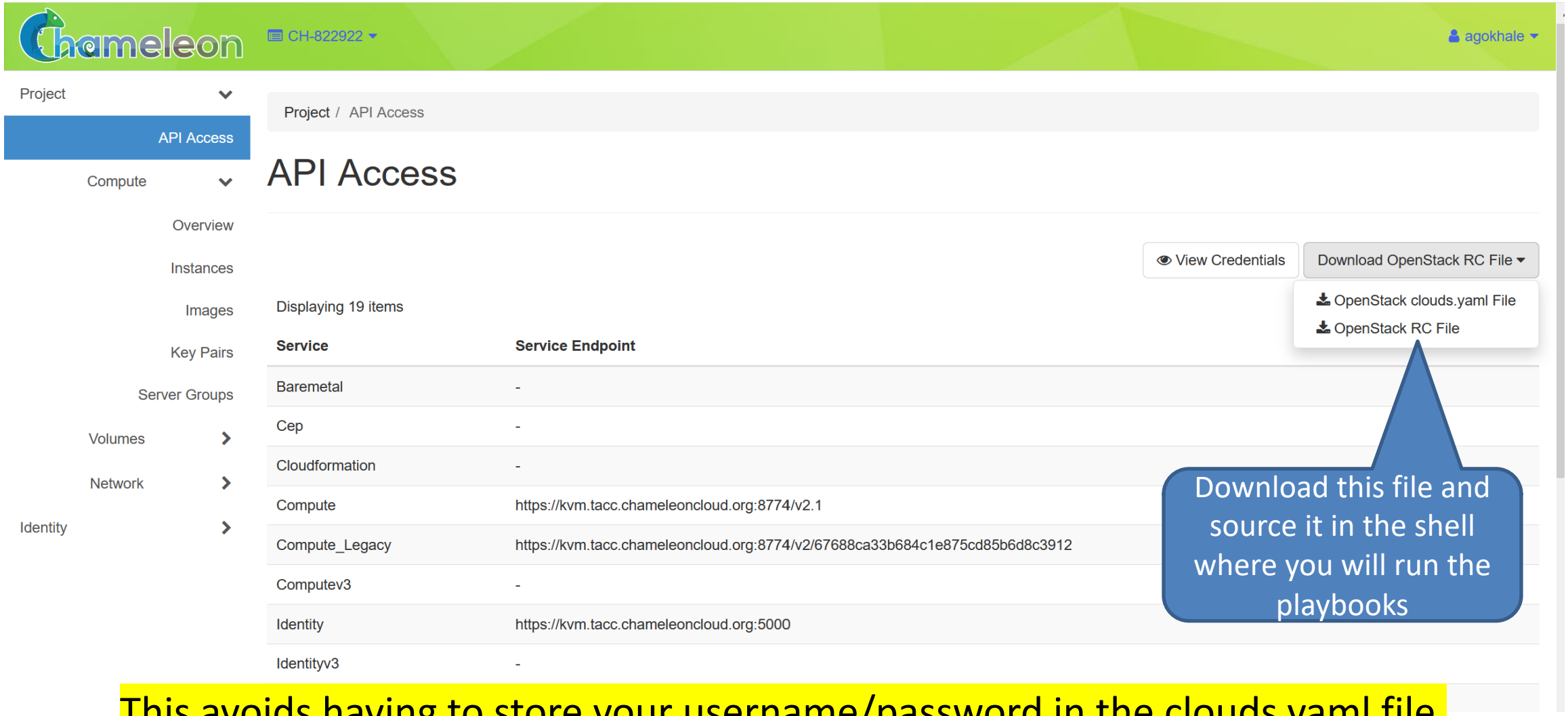
You might need to add /v3 after 5000

```
# ********************* IMPORTANT ***********************
# To set password go to https://chameleoncloud.readthedocs.io/en/latest/technica
# See "CLI authentication" section on this page.
# USe the generated password in this file.
# Note this password is not for GUI of Chameleon cloud.
#
# If you are a member of multiple projects, when invoking the CLI, provide
# either the env variable OS_CLOUD=<project> or flag --os-cloud=<project> to
# target your operation to the desired project, where <project> is the name
# (or nickname, if set) of your project.
clouds:
  chameleon: # This is the alias you use in your tasks yml file/s. I changed it to chameleon
    auth_type: v3oidcpassword
    auth:
      auth_url: https://kvm.tacc.chameleoncloud.org:5000/v3
      username: replace_with_your_username
      password: replace_with_your_password # Create your CLI password and use here. See abov
      protocol: openid
      identity_provider: chameleon
      discovery_endpoint: https://auth.chameleoncloud.org/auth/realms/chameleon/.well-known/
      client_id: keystone-kvm-prod
      access_token_type: access_token
      client_secret: none
    region_name: "KVM@TACC"
    interface: public
    identity_api_version: 3
    project_name: "CHI-221026"
    project_domain_name: chameleon
```

auth and region_name are at same level

Sample YAML file on Brightspace is like this. Modify accordingly and add this to your Vagrant created VM in ~/.config/openstack directory

Alternate Approach for “auth” Credentials



This avoids having to store your username/password in the clouds.yaml file.
Read openstack documentation of you want to use this method.

Documentation

- https://docs.ansible.com/ansible/latest/user_guide/index.html
  - Vast amount of info
  - My suggestion is to search for info as the need arises
  - Use sample playbooks to get started
- Built in modules
  - https://docs.ansible.com/ansible/latest/collections/ansible/builtin/
- Openstack modules
  - https://docs.ansible.com/ansible/latest/collections/openstack/cloud/
- AWS modules
  - https://docs.ansible.com/ansible/latest/collections/amazon/aws/
- Docker modules
  - https://docs.ansible.com/ansible/latest/collections/community/general/docker_container_module.html