

Name: Maxine Audrey D. Pulao	Date Performed: Dec 11 2022
Course/Section: CPE31S2	Date Submitted: Dec 12 2022
Instructor: Dr. Jonathan Taylar	Semester and SY: 2022-2023
Activity 15: OpenStack Installation (Neutron, Horizon, Cinder)	
1. Objectives	
Create a workflow to install OpenStack using Ansible as your Infrastructure as Code (IaC).	
2. Intended Learning Outcomes	
<ol style="list-style-type: none"> 1. Analyze the advantages and disadvantages of cloud services 2. Evaluate different Cloud deployment and service models 3. Create a workflow to install and configure OpenStack base services using Ansible as documentation and execution. 	
3. Resources	
<p>Oracle VirtualBox (Hypervisor)</p> <p>1x Ubuntu VM or Centos VM</p>	
4. Tasks	
<ol style="list-style-type: none"> 1. Create a new repository for this activity. <pre>pulao@pulao-VirtualBox:~/ansible\$ git clone git@github.com:MaxinePulao/Activity-15. Cloning into 'Activity-15'... remote: Enumerating objects: 3, done. remote: Counting objects: 100% (3/3), done. remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 Receiving objects: 100% (3/3), done.</pre> 2. Create a playbook that converts the steps in the following items in https://docs.openstack.org/install-guide/ <ol style="list-style-type: none"> a. Neutron 	

```
1 - name: installing the components for neutron
2 apt:
3   name: neutron-linuxbridge-agent
4   when: ansible_distribution == "Ubuntu"
5
6 - name: configuring RabbitMQ message queue access
7 copy:
8   dest: /etc/neutron/neutron.conf
9   content: |
10     [DEFAULT]
11     transport_url = rabbit://openstack:1234@controller
12 - name: configuring Identity service access (1)
13 copy:
14   dest: /etc/neutron/neutron.conf
15   content: |
16     [DEFAULT]
17     auth_strategy = keystone
18     [keystone_authtoken]
19     www_authenticate_uri = http://controller:5000
20     auth_url = http://controller:5000
21     memcached_servers = controller:11211
22     auth_type = password
23     project_domain_name = default
24     user_domain_name = default
25     project_name = service
26     username = neutron
27     password = 1234
28 - name: configuring the lock path
29 copy:
30   dest: /etc/neutron/neutron.conf
31   content: |
32     [oslo_currency]
33     lock_path = /var/lib/neutron/tmp
34 ## Configuring the compute service to use networking service
35
36 - name: configuring the access parameters
37 copy:
38   dest: /etc/nova/nova.conf
39   content: |
40     [neutron]
41     auth_url = http://controller:5000
42     auth_type = password
43     project_domain_name = default
44     user_domain_name = default
45     region_name = RegionOne
46     project_name = service
47     username = neutron
48     password = 1234
49 - name: restarting the compute service
```

```

50 shell: service nova-compute restart
51 when: ansible_distribution == "Ubuntu"
52
53 - name: restarting the linux bridge agent
54   shell: service neutron-linuxbridge-agent restart
55   when: ansible_distribution == "Ubuntu"
56
57 - block:
58   - name: Verifying if already running and active the nova-compute.
59     shell: systemctl status nova-compute
60     register: novacompute_service
61
62   - debug:
63     msg="{ { novacompute_service } }"
64
65 - block:
66   - name: Verifying if already running and active the neutron-linuxbridge-agent
67     shell: systemctl status neutron-linuxbridge-agent
68     register: neutron_service
69
70   - debug:
71     msg="{ { neutron_service } }"
72

```

b. Horizon

```
1 - name: configuring the dashboard to use Openstack services
2   copy:
3     dest: /etc/openstack-dashboard/local_settings.py
4     content: |
5       OPENSTACK_HOST = "controller"
6 - name: Allowing all hosts to access dashboard
7   copy:
8     dest: /etc/openstack-dashboard/local_settings.py
9     content: |
10      ALLOWED_HOSTS = ['*']
11 - name: configure the memcached session storage service
12   copy:
13     dest: /etc/openstack-dashboard/local_settings.py
14     content: |
15       SESSION_ENGINE = 'django.contrib.sessions.backends.cache'
16       CACHES = {
17         'default': {
18           'BACKEND': 'django.core.cache.backends.memcached.MemcachedCache', 'LOCATION':
19             'controller:11211',
20         }
21       }
22 - name: enable the identity API version 3
23   copy:
24     dest: /etc/openstack-dashboard/local_settings.py
25     content: |
26       OPENSTACK_KEYSTONE_URL = "http://%s/identity/v3" % OPENSTACK_HOST
27 - name: enable support for domains
28   copy:
29     dest: /etc/openstack-dashboard/local_settings.py
30     content: |
31       OPENSTACK_KEYSTONE_MULTIDOMAIN_SUPPORT = True
32 - name: configure API versions
33   copy:
34     dest: /etc/openstack-dashboard/local_settings.py
35     content: |
36       OPENSTACK_API_VERSIONS = {
37         "identity": 3,
38         "image": 2,
39         "volume": 3,
40       }
41 - name: configure default as the default domain for users that you create via the dashboard
42   copy:
43     dest: /etc/openstack-dashboard/local_settings.py
44     content: |
45       OPENSTACK_KEYSTONE_DEFAULT_DOMAIN = "Default"
46 - name: configure user as the default role for users that your create via the dashboard
47   copy:
48     dest: /etc/openstack-dashboard/local settings.py
```

```

48     content: |
49         OPENSTACK_KEYSTONE_DEFAULT_ROLE = "user"
50 - name: if you choose network 1, disable support for layer-3 networking services
51 copy:
52     dest: /etc/openstack-dashboard/local_settings.py
53     content: |
54         OPENSTACK_NEUTRON_NETWORK = {
55             'enable_router': False,
56             'enable_quotas': False,
57             'enable_ipv6': False,
58             'enable_distributed_router': False,
59             'enable_ha_router': False,
60             'enable_fip_topology_check': False
61         }
62 - name: add the following line if not added yet
63 copy:
64     dest: /etc/openstack-dashboard/local_settings.py
65     content: |
66         CACHES = {
67             'default': {
68                 'BACKEND': 'django.core.cache.backends.memcached.MemcachedCache', 'LOCATION':
192.168.56.137:11211',
69             },
70         }
71         SESSION_ENGINE = "django.contrib.sessions.backends.cache"
72         OPENSTACK_HOST = "192.168.56.124"
73         OPENSTACK_KEYSTONE_URL = "http://%s/identity/v3" % OPENSTACK_HOST
74         OPENSTACK_KEYSTONE_URL = "http://192.168.56.137:5000/v3"
75         TIME_ZONE = "Asia/Tokyo"
76         OPENSTACK_KEYSTONE_MULTIDOMAIN_SUPPORT = True
77         OPENSTACK_KEYSTONE_DEFAULT_DOMAIN = 'Default'
78         OPENSTACK_API_VERSIONS = {
79             "identity": 3,
80             "volume": 3,
81             "compute": 2,
82         }
83         WSGIApplicationGroup %{GLOBAL}
84 - name: install apache2
85 apt:
86     name: apache2
87
88 - name: finalize installation by starting the apache2 service
89 shell: sudo systemctl start apache2.service
90
91 - name: finalize installation by reloading the apache2 service
92 shell: sudo systemctl reload apache2.service
93
94 - block:
95     - name: Verifying the apache2.service
96       shell: systemctl status apache2.service
97       register: apache2_service
98
99 - debug:

```

c. Cinder

```
1 - name: install the packages for cinder (1) controller node
2 apt:
3   name: cinder-api
4
5 - name: install cinder scheduler
6 shell: sudo apt install cinder-scheduler
7
8 - name: configure database access
9 copy:
10  dest: /etc/cinder/cinder.conf
11  content: |
12    [database]
13    connection = mysql+pymysql://cinder:1234@controller/cinder
14 - name: configure RabbitMQ message queue access
15 copy:
16  dest: /etc/cinder/cinder.conf
17  content: |
18    [DEFAULT]
19    transport_url = rabbit://openstack:1234@controller
20 - name: configure identity services access
21 copy:
22  dest: /etc/cinder/cinder.conf
23  content: |
24    [DEFAULT]
25    auth_strategy = keystone
26    [keystone_authtoken]
27    www_authenticate_uri = http://controller:5000
28    auth_url = http://controller:5000
29    memcached_servers = controller:11211
30    auth_type = password
31    project_domain_name = default
32    user_domain_name = default
33    project_name = service
34    username = cinder
35    password = 1234
36 - name: configure my_ip option to use the management interface IP add of controller node
37 copy:
38  dest: /etc/cinder/cinder.conf
39  content: |
40    [DEFAULT]
41    my_ip = 192.168.56.137
42 - name: configure the lock path
43 copy:
44  dest: /etc/cinder/cinder.conf
45  content: |
46    [oslo_concurrency]
47    lock_path = /var/lib/cinder/tmp
48 - name: populate the block storage database
```

```
49  shell: su -s /bin/sh -c "cinder-manage db sync" cinder
50
51  ### Configure compute to use block storage
52
53  - name: adding the following line for block storage
54    copy:
55      dest: /etc/nova/nova.conf
56      content: |
57        [cinder]
58          os_region_name = RegionOne
59  - name: install nova-api
60    shell: sudo apt install nova-api
61
62  - name: restarting the compute API service
63    shell: service nova-api start
64
65  - name: restart the block storage services (1)
66    shell: service cinder-scheduler start
67
68  - name: restart the block storage services (2)
69    shell: sudo systemctl start apache2
70
71  ## Install and configure components (Storage Node)
72
73  - name: install the supporting utility packages
74    apt:
75      name:
76        - lvm2
77        - thin-provisioning-tools
78
79  - name: create the LVM physical volume /dev/sdb
80    file:
81      path: /dev/sdb
82      state: directory
83
84  - name: create the LVM volume group cinder-volume
85    shell: sudo touch cinder-volumes /dev/sdb
86
87  ## add a filter that accepts the /dev/sdb in /etc/lvm/lvm.conf
88
89  - name: install the packages for cinder (2) storage node
90    apt:
91      name:
92        - cinder-volume
93        - tgt
94
95  #configure database access
96  #configure RabbitMQ message queue access
```

```

97 #configure identity services access
98 #configure the my_ip option
99
100 - name: configure the LVM backend with the LVM driver
101   copy:
102     dest: /etc/cinder/cinder.conf
103     content: |
104       [lvm]
105       volume_driver = cinder.volume.drivers.lvm.LVMVolumeDriver
106       volume_group = cinder-volumes
107       target_protocol = iscsi
108       target_helper = tgtadm
109 - name: enabling the LVM backend
110   copy:
111     dest: /etc/cinder/cinder.conf
112     content: |
113       [DEFAULT]
114       enabled_backends = lvm
115 - name: configuring the location of the image service API
116   copy:
117     dest: /etc/cinder/cinder.conf
118     content: |
119       [DEFAULT]
120       glance_api_servers = http://controller:9292
121 - name: configuring the lock path
122   copy:
123     dest: /etc/cinder/cinder.conf
124     content: |
125       [oslo_concurrency]
126       lock_path = /var/lib/cinder/tmp
127 - name: Restarting the block storage volume service including its dependencies (1)
128   shell: service tgt restart
129
130 - name: Restarting the block storage volume service including its dependencies (2)
131   shell: service cinder-volume restart
132
133 ## Install and configure components (Backup Service)
134
135 - name: install the packages for cinder (3) backup service
136   apt:
137     name: cinder-backup
138
139 - name: configuring the backup options
140   copy:
141     dest: /etc/cinder/cinder.conf
142     content: |
143       [DEFAULT]
144       backup_driver = cinder.backup.drivers.swift.SwiftBackupDriver
145       backup_swift_url = SWIFT_URL
146 #- name: replacing the SWIFT_URL with the URL of the Object Service service
147 # shell: openstack catalog show object-store
148
149 #finalize installation
150
151 - name: restart the block storage backup service
152   shell: service cinder-backup start
153
154 - block:
155   - name: Verifying the cinder-backup.
156     shell: systemctl status cinder-backup
157     register: cinder_service
158
159 - debug:
160   msg="{{ cinder_service }}"

```


- d. Create different plays in installing per server type (controller, compute etc.) and identify it as a group in the Inventory file.

```
ansible.cfg
~/ansible/Activity-15

1 |
2 [defaults]
3
4 inventory = inventory
5 host_key_checking = False
6
7 deprecation_warnings= False
8
9 remote_user = pulao
10 private_key_file = ~/.ssh/id_rsa
```

```
*inventory
~/ansible/Activity-15

1 [controller_node]
2 192.168.56.138
```

```
main.yml
~/ansible/Activity-15

1 ---
2
3 - hosts: all
4   become: true
5   pre_tasks:
6
7   - name: Dpkg fixing in Ubuntu Servers
8     shell: |
9       dpkg --configure -a
10    when: ansible_distribution == "Ubuntu"
11
12  - name: Update and Upgrade remote in Ubuntu servers
13    apt:
14      update_cache: yes
15      upgrade: yes
16      when: ansible_distribution == "Ubuntu"
17
18  - hosts: controller_node
19    become: true
20    roles:
21      - neutron
22      - horizon
23      - cinder
```

- e. Add, commit and push it to your GitHub repo.

```
pulao@pulao-VirtualBox:~/ansible/Activity-15$ git commit -m "Activity-15"
[main 3ef64ad] Activity-15
 9 files changed, 367 insertions(+)
 create mode 100644 .ansible.cfg.swp
 create mode 100644 .inventory.swp
 create mode 100644 .main.yml.swp
 create mode 100644 ansible.cfg
 create mode 100644 inventory
 create mode 100644 main.yml
 create mode 100644 roles/cinder/tasks/main.yml
 create mode 100644 roles/horizon/tasks/main.yml
 create mode 100644 roles/neutron/tasks/main.yml
pulao@pulao-VirtualBox:~/ansible/Activity-15$ git push origin main
Enumerating objects: 19, done.
Counting objects: 100% (19/19), done.
Compressing objects: 100% (11/11), done.
Writing objects: 100% (18/18), 4.20 KiB | 186.00 KiB/s, done.
Total 18 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), done.
To github.com:MaxinePulao/Activity-15.git
   cb02d59..3ef64ad  main -> main
```

5. Output (screenshots and explanations)

```
maxine@local:~/ansible/Activity-15$ ansible-playbook --ask-become-pass main.yml
BECOME password:

PLAY [all] *****

TASK [Gathering Facts] *****
ok: [192.168.56.140]

TASK [Dpkg fixing in Ubuntu Servers] *****
changed: [192.168.56.140]

TASK [Update and Upgrade remote in Ubuntu servers] *****
[WARNING]: The value "True" (type bool) was converted to "'True'" (type
string). If this does not look like what you expect, quote the entire value to
ensure it does not change.
ok: [192.168.56.140]

PLAY [controller_node] *****

TASK [Gathering Facts] *****
ok: [192.168.56.140]

TASK [neutron : installing the components for neutron] *****
ok: [192.168.56.140]

TASK [neutron : configuring RabbitMQ message queue access] *****
changed: [192.168.56.140]

TASK [neutron : configuring Identity service access (1)] *****
changed: [192.168.56.140]

TASK [neutron : configuring the lock path] *****
changed: [192.168.56.140]

TASK [neutron : configuring the access parameters] *****
ok: [192.168.56.140]

TASK [neutron : restarting the compute service] *****
[WARNING]: Consider using the service module rather than running 'service'. If
you need to use command because service is insufficient you can add 'warn:
false' to this command task or set 'command_warnings=False' in ansible.cfg to
get rid of this message.
changed: [192.168.56.140]

TASK [neutron : restarting the linux bridge agent] *****
changed: [192.168.56.140]

TASK [neutron : Verifying if already running and active the nova-compute.] *****
changed: [192.168.56.140]
```

```
TASK [neutron : restarting the compute service] *****
[WARNING]: Consider using the service module rather than running 'service'. If
you need to use command because service is insufficient you can add 'warn:
false' to this command task or set 'command_warnings=False' in ansible.cfg to
get rid of this message.
```

```
changed: [192.168.56.140]
```

```
TASK [neutron : restarting the linux bridge agent] *****
```

```
changed: [192.168.56.140]
```

```
TASK [neutron : Verifying if already running and active the nova-compute.] *****
```

```
changed: [192.168.56.140]
```

```
TASK [neutron : debug] *****
```

```
ok: [192.168.56.140] => {
```

```
  "msg": {
    "changed": true,
    "cmd": "systemctl status nova-compute",
    "delta": "0:00:00.026680",
    "end": "2022-12-12 09:41:31.880816",
    "failed": false,
    "rc": 0,
    "start": "2022-12-12 09:41:31.854136",
    "stderr": "",
    "stderr_lines": [],
    "stdout": "●nova-compute.service - OpenStack Compute\n    Loaded: load
ed (/lib/systemd/system/nova-compute.service; enabled; vendor preset: enabled)\n
    Active: active (running) since Mon 2022-12-12 09:41:29 PST; 2s ago\n    Main
PID: 3322 (nova-compute)\n    Tasks: 1 (limit: 21504)\n    Memory: 37.5M\n
    CPU: 571ms\n    CGroup: /system.slice/nova-compute.service\n
└─3322 /usr/bin/python3 /usr/bin/nova-compute --config-file=/etc/nova/nova.conf
--config-file=/etc/nova/nova-compute.conf --log-file=/var/log/nova/nova-compute.
log\n\nDec 12 09:41:29 local systemd[1]: Started OpenStack Compute.\nDec 12 09:4
1:31 local nova-compute[3322]: Modules with known eventlet monkey patching issue
s were imported prior to eventlet monkey patching: urllib3. This warning can usu
ally be ignored if the caller is only importing and not executing nova code.",
    "stdout_lines": [
      "●nova-compute.service - OpenStack Compute",
      "    Loaded: loaded (/lib/systemd/system/nova-compute.service; enab
led; vendor preset: enabled)",
      "    Active: active (running) since Mon 2022-12-12 09:41:29 PST; 2s
ago",
      "    Main PID: 3322 (nova-compute)",
      "    Tasks: 1 (limit: 21504)",
      "    Memory: 37.5M",
      "    CPU: 571ms",
      "    CGroup: /system.slice/nova-compute.service",
      "    └─3322 /usr/bin/python3 /usr/bin/nova-compute --config
-file=/etc/nova/nova.conf --config-file=/etc/nova/nova-compute.conf --log-file=/
```

```

    "stdout_lines": [
        "●nova-compute.service - OpenStack Compute\n      Loaded: load
ed (/lib/systemd/system/nova-compute.service; enabled; vendor preset: enabled)\n
      Active: active (running) since Mon 2022-12-12 09:41:29 PST; 2s ago\n      Main
PID: 3322 (nova-compute)\n      Tasks: 1 (limit: 21504)\n      Memory: 37.5M\n
      CPU: 571ms\n      CGroup: /system.slice/nova-compute.service\n
└─3322 /usr/bin/python3 /usr/bin/nova-compute --config-file=/etc/nova/nova.conf
--config-file=/etc/nova/nova-compute.conf --log-file=/var/log/nova/nova-compute.
log\n\nDec 12 09:41:29 local systemd[1]: Started OpenStack Compute.\nDec 12 09:4
1:31 local nova-compute[3322]: Modules with known eventlet monkey patching issue
s were imported prior to eventlet monkey patching: urllib3. This warning can usu
ally be ignored if the caller is only importing and not executing nova code.",
        "stdout_lines": [
            "●nova-compute.service - OpenStack Compute",
            "      Loaded: loaded (/lib/systemd/system/nova-compute.service; enab
led; vendor preset: enabled)",
            "      Active: active (running) since Mon 2022-12-12 09:41:29 PST; 2s
ago",
            "      Main PID: 3322 (nova-compute)",
            "      Tasks: 1 (limit: 21504)",
            "      Memory: 37.5M",
            "      CPU: 571ms",
            "      CGroup: /system.slice/nova-compute.service",
            "      └─3322 /usr/bin/python3 /usr/bin/nova-compute --config
-file=/etc/nova/nova.conf --config-file=/etc/nova/nova-compute.conf --log-file=/
var/log/nova/nova-compute.log",
            "",
            "Dec 12 09:41:29 local systemd[1]: Started OpenStack Compute.",
            "Dec 12 09:41:31 local nova-compute[3322]: Modules with known eventl
et monkey patching issues were imported prior to eventlet monkey patching: urlli
b3. This warning can usually be ignored if the caller is only importing and not
executing nova code."
        ]
    ]
}
}

```

TASK [neutron : Verifying if already running and active the neutron-linuxbridge-agent] ***

changed: [192.168.56.140]

TASK [neutron : debug] *****

ok: [192.168.56.140] => {

```

    "msg": {
        "changed": true,
        "cmd": "systemctl status neutron-linuxbridge-agent",
        "delta": "0:00:00.021840",
        "end": "2022-12-12 09:41:33.447471",
        "failed": false,
        "rc": 0,
        "start": "2022-12-12 09:41:33.425631"
    }
}

```

```

    "end": "2022-12-12 09:41:33.447471",
    "failed": false,
    "rc": 0,
    "start": "2022-12-12 09:41:33.425631",
    "stderr": "",
    "stderr_lines": [],
    "stdout": "●neutron-linuxbridge-agent.service - Openstack Neutron Linux
Bridge Agent\n    Loaded: loaded (/lib/systemd/system/neutron-linuxbridge-agen
t.service; enabled; vendor preset: enabled)\n    Active: active (running) since
Mon 2022-12-12 09:41:30 PST; 2s ago\n    Process: 3364 ExecStartPre=/bin/mkdir
-p /var/lock/neutron /var/log/neutron /var/lib/neutron (code=exited, status=0/SU
CCESS)\n    Process: 3366 ExecStartPre=/bin/chown neutron:neutron /var/lock/neut
ron /var/log/neutron /var/lib/neutron (code=exited, status=0/SUCCESS)\n    Proce
ss: 3368 ExecStartPre=/sbin/modprobe br_netfilter (code=exited, status=0/SUCCESS
)\n    Main PID: 3375 (neutron-linuxbr)\n    Tasks: 1 (limit: 21504)\n    Memo
ry: 33.9M\n    CPU: 611ms\n    CGroup: /system.slice/neutron-linuxbridge-ag
ent.service\n        └─3375 /usr/bin/python3 /usr/bin/neutron-linuxbridge-a
gent --config-file=/etc/neutron/neutron.conf --config-file=/etc/neutron/plugins/
ml2/linuxbridge_agent.ini --log-file=/var/log/neutron/neutron-linuxbridge-agent.
log\n\nDec 12 09:41:30 local systemd[1]: Starting Openstack Neutron Linux Bridge
Agent...\nDec 12 09:41:30 local systemd[1]: Started Openstack Neutron Linux Bri
dge Agent.",
    "stdout_lines": [
        "●neutron-linuxbridge-agent.service - Openstack Neutron Linux Bridg
e Agent",
        "    Loaded: loaded (/lib/systemd/system/neutron-linuxbridge-agent.
service; enabled; vendor preset: enabled)",
        "    Active: active (running) since Mon 2022-12-12 09:41:30 PST; 2s
ago",
        "    Process: 3364 ExecStartPre=/bin/mkdir -p /var/lock/neutron /var
/log/neutron /var/lib/neutron (code=exited, status=0/SUCCESS)",
        "    Process: 3366 ExecStartPre=/bin/chown neutron:neutron /var/lock
/neutron /var/log/neutron /var/lib/neutron (code=exited, status=0/SUCCESS)",
        "    Process: 3368 ExecStartPre=/sbin/modprobe br_netfilter (code=ex
ited, status=0/SUCCESS)",
        "    Main PID: 3375 (neutron-linuxbr)",
        "    Tasks: 1 (limit: 21504)",
        "    Memory: 33.9M",
        "    CPU: 611ms",
        "    CGroup: /system.slice/neutron-linuxbridge-agent.service",
        "        └─3375 /usr/bin/python3 /usr/bin/neutron-linuxbridge-a
gent --config-file=/etc/neutron/neutron.conf --config-file=/etc/neutron/plugins/
ml2/linuxbridge_agent.ini --log-file=/var/log/neutron/neutron-linuxbridge-agent.
log",
        "",
        "Dec 12 09:41:30 local systemd[1]: Starting Openstack Neutron Linux
Bridge Agent...",
        "Dec 12 09:41:30 local systemd[1]: Started Openstack Neutron Linux B
ridge Agent "
    ]
}
PLAY RECAP *****
192.168.56.140 : ok=42   changed=34   unreachable=0   failed=0   s
kipped=0   rescued=0   ignored=0

```

Reflections:

Answer the following:

1. Describe Neutron, Horizon and Cinder services

Neutron is an OpenStack project to provide “Network connection as a Service (NaaS)”. The OpenStack Networking service (neutron) provides an API that allows users to build rich networking topologies, set up and define network connectivity, configure advanced network policies and addressing in the cloud. While OpenStack Horizon is a web-based graphical user interface for managing OpenStack compute, storage, and networking services, it is accessible by cloud administrators and users. Lastly, Cinder is a Block Storage service for OpenStack. It virtualizes the management of block storage devices and gives end users access to a self-service API that allows them to request and use those resources without having to know where or what kind of device their storage is actually deployed on.

Conclusions:

In this activity, I have mastered on how to create a playbook to install and update my ubuntu server and openstack packages. This activity also showed me what could services are and what is their purpose. However, it may have some issues regarding on security since some of it is public and accessible. To keep their cloud environment secure, the majority of cloud service providers implement pertinent security standards and industry certifications. However, storing data and business-critical files in virtual data centres can potentially open you up to risks. Common risks are data loss and theft. This activity also taught me the components and what makes up a cloud service and its purpose to my servers. There are four cloud deployment models. Public, Private, Community, and Hybrid. Each deployment model is defined by where the environment's infrastructure resides. There are three main cloud service models: Software as a Service, Platform as a Service, Infrastructure as a Service. After performing this activity, I hope I could be more comfortable and try to master managing servers like Ubuntu and CentOS.