Empowering Undergraduate Students With Cloud Computing Skills: A Proposal for OpenStack-Centric Education

Dr. Emil H Salib
Computer Science Department, CISE, James Madison University
Harrisonburg, VA 22801, USA
salibeh@jmu.edu

***Abstract*—This innovative practice full paper presents the incorporation of cloud computing into Information Technology (IT) undergraduate programs and curricula which has become increasingly vital for upgrading and enhancing the delivery of teaching materials reliant on virtual machines, such as networking and network security lab exercises. Furthermore, the inclusion of cloud computing-focused courses in these programs will soon be imperative to address the rising demand in the job market for IT graduates equipped with cloud computing knowledge and skills.**

However, accessing cloud computing resources poses challenges. Public cloud computing services such as AWS, Azure, and Google Cloud impose significant cost and accessibility barriers on educational institutions and students. Conversely, conventional operator deployment of private cloud platforms like OpenStack presents hurdles in terms of the need for specialized expertise, training, and resource management and administration.

This paper introduces a solution that simplifies the process of deploying small-scale OpenStack environments, making it comparable to the installation and setup efforts required for standalone virtualization environments like VMware Workstation (WS). Leveraging the kolla-ansible framework and tools, this innovative solution streamlines the deployment, configuration, and management of OpenStack, resulting in significant time and resource savings without the need for dedicated and highly trained resources.

The paper provides our experience in migrating networking and network security lab exercises of two courses from the standalone virtualization environment, VMware WS, to our own private cloud computing platform based on OpenStack's All-In-One (AIO) deployment using kolla-ansible customization, configuration and automation tools. A detailed account of the updates and modifications of the lab exercises network setups and instructions required for the migration is also presented, including specific details of best practices and their configurations and implementation.

Additionally, it offers a qualitative assessment of the students' experience in utilizing OpenStack compared to VMware WS for executing networking and network security-related lab exercises.

***Index Terms*—private cloud computing, OpenStack, undergraduate, networking, network security, migration, VMware Workstation, lab exercise.**

I. Introduction

The integration of cloud computing into IT undergraduate curricula, enabling students to conduct their networking and network security lab exercises from anywhere and at any time, presents both an opportunity and a challenge. The major challenge lies in the significant cost and accessibility barriers imposed by current public cloud platforms such as AWS, Azure, and Google Cloud on educational institutions and students. This is based on our experience in conducting other class projects. Another potential issue is the reliance on proprietary methods and algorithms inherent in these platforms, limiting the scope of what educators can cover with students beyond practitioner-focused exercises. In the case of a private cloud computing approach, such as an open-source, cost-effective platform like OpenStack [1], the major challenge, until now, has been the heavy demand for specialized resources to deploy and maintain the private cloud on educational institution premises.

Our proposed solution to these challenges revolves around leveraging innovative tool, specifically known as kolla-ansible [2], to customize, deploy, and run OpenStack as a private cloud computing platform. This was accomplished with the same level of personnel resources typically required for currently deploying virtualization platforms such as standalone VMware Workstation (WS) [3]. By embracing OpenStack, IT undergraduate students gain access to a platform that not only facilitates the exploration of cloud computing concepts, basic architecture, and services, but also enables potential enhancements. Additionally, it enables educators to migrate their networking and network security lab exercises, which require students to be in a computer lab using a standalone virtualization environment such as VMware WS, to a cloud environment that allows them to perform their lab exercises at any time and from anywhere. All of this is achieved without the financial burdens typically associated with proprietary cloud providers. As an added benefit, introducing a private cloud platform into an educational institution will enable some educators to develop and deliver new teaching materials (including lab exercises) on the fundamentals of cloud computing architecture and inner workings into IT undergraduate curricula. This is desperately needed to meet the demand for cloud-savvy professionals, which is rapidly increasing.

This paper presents a successful experiment conducted at our institution, focusing on migrating the lab exercises of two (2) IT undergraduate courses: IT 461 Inter-networking and IT 460 Advanced Network Security from VMware WS to an OpenStack Private Cloud Computing environment. We view this as the first step in the process of migrating our IT courses currently based on VMware WS to a cloud computing platform.

Central to this solution is the creation and management of the OpenStack deployment and updates to the hands-on lab exercises. These updates include instructions on how to use OpenStack cloud services in general, and more specifically, how to exercise and utilize many of the OpenStack capabilities, such as creating internal and external networks as part of conducting the lab exercises. As part of the updates, we demonstrated how to integrate hardware devices such as Wireless Access Points with instances (virtual machines) running on OpenStack. We also demonstrated how educators could leverage a unique cloud computing environment, granting them simultaneous access to students' instances. This facilitates real-time troubleshooting opportunities and allows for the inclusion of VMs in course assignments and exams from anywhere, removing location constraints observed with standalone VMware WS setups.

The structure of this paper is as follows: Section II presents a brief review of related works. In Section III, we present an overview of OpenStack concepts and services, the role of kolla-ansible in the solution design, and specific design considerations related to deployment options and the hardware resources adopted. The detailed implementation of the proposed solution is provided in Section IV, which includes the network arrangement adopted and the customization configurations applied. Section V provides a detailed account of the lab exercises updates and upgrades required for the successful migration from VMware WS to the OpenStack virtualization environment. Finally, Section VI engages in a discussion encompassing our students' reflections on their experience and observations of using the OpenStack virtualization solution, along with our conclusion and potential future work.

II. Related Work

There have been interests, from the late 2000s to the early 2010s, to introduce OpenStack for educational purposes, as described in [4], [5], and [6]. Recently, we have seen renewed interest in using OpenStack for education purposes, as described in [7]. Another interesting related work is the proposal of a system to automate the deployment of a cyber range [8]. In that article, the authors leveraged the OpenStack cloud platform as the deployment platform, thereby enhancing the applicability of the cyber range. In [9], the authors share their experience in constructing a scalable hosting architecture tailored to educational settings. They describe an experimental virtualization setup implemented within an educational scenario, along with guidelines for configuring OpenStack. From a general perspective, [10] presents the benefits and challenges of cloud computing in education.

III. Solution Design Considerations

In this section, we will provide an overview of OpenStack concepts and the design considerations of our solution. OpenStack is an open-source cloud software that consists of a series of allied projects controlling large pools of computing, storage, and network resources in a data center while managed through a dashboard. See Fig. 1.

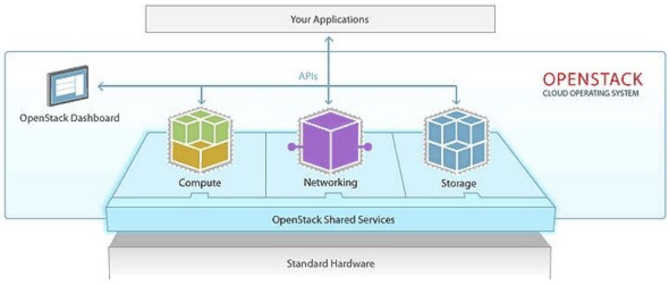


Fig. 1: OpenStack Architecture Overview [11].

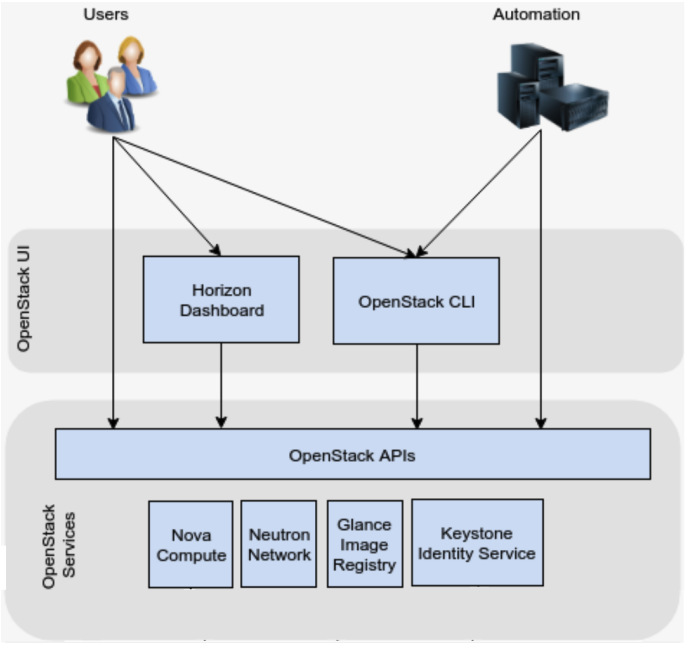


Fig. 2: OpenStack All-In-One Components and Services [12].

OpenStack is composed of a variety of interdependent components and services [13] that work together to provide the aforementioned resources. In this project, the components and services used (see Fig. 2) includes:

- **Compute (Nova)**: A component that provides scalable, on-demand access to computing resources, including instances/virtual machines (VMs) and bare metal servers.
- **Networking (Neutron)**: A component that provides Networking-as-a-Service (NaaS) for use with instances/VMs and other cloud resources.
- **Identity (Keystone)**: A component that provides authentication and authorization services for all other components.
- **Image (Glance)**: A component that provides a repository for instance/VM images.
- **Dashboard (Horizon)**: A web-based graphical user interface (GUI) for managing OpenStack resources.

OpenStack, a cloud computing platform, provides many types of resources, but the most basic types include the following:

- **Project**: A project is a group of zero or more users. In Compute, a project owns virtual machines. Users can be

associated with more than one project. Each project and user pairing can have a role associated with it [14].

- **Instance**: A virtual machine that runs software and containers. An instance can be used as a workstation, a server, or as a worker node in a larger cluster.
- **Flavor**: In OpenStack, flavors define the compute, memory, and storage capacity of nova computing instances [15].
- **Volume**: A virtual hard disk drive that stores data. It can be small (under a gigabyte) or very large (many terabytes).
- **Port**: A port is a connection point for attaching a single device, such as the Network Interface Card (NIC) of an instance, to a network. The port also describes the associated network configuration, such as the MAC ID/EUI-48 and IP addresses to be used on that port [16].
- **Security Group**: Security groups are sets of IP filter rules that are applied to all instances, defining networking access to the instance. Group rules are project-specific; project members can edit the default rules for their group and add new rule sets. They are modelled as an attribute of ports rather than servers/instances [17].
- **Floating IP address**: Each instance has a private, fixed IP address and can also have a floating IP address. Floating IP addresses allow a user to direct ingress network traffic to their OpenStack instances. They are associated with instances in addition to their fixed IP addresses [18].
- **Networks**: There are two types of network: project (commonly referenced as Internal) and provider (commonly referenced as External) networks [19]. It is possible to share these types of networks among projects as part of the network creation process as shown in Fig. 3.

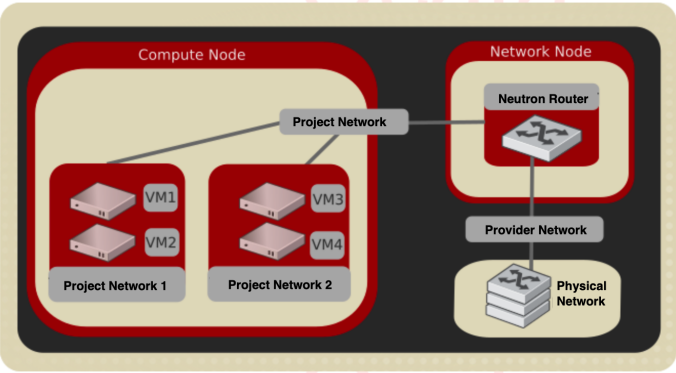


Fig. 3: OpenStack Project Network Services [20].

One of the major challenges with OpenStack is the complexity of the platform and its flexibility in terms of the many different deployment options, components, and services that an operator could choose from. That typically translates into the need for human resources, specialized knowledge, and a lot of time. With kolla-ansible [2] which deploys OpenStack services and infrastructure components in Docker containers, OpenStack deployment is made easy and straightforward. kolla-ansible provides Ansible playbooks to deploy the Kolla images and containers. It allows for complete customization and thus permits operators with little experience to deploy OpenStack quickly, and as experience grows, modify the OpenStack configuration to suit the operator's exact requirements [2]. The bottom line is that kolla-ansible simplifies the deployment process through automation, making it easier for administrators to create a robust OpenStack cloud deployment.

As part of our solution design, we opted for the simplest kolla-ansible inventory option for OpenStack deployment, which is the All-In-One (AIO) inventory. Unlike the multi-node inventory case, the AIO is ready for deploying a single-node OpenStack cloud. Additionally, we have ensured to select the minimum but operational set of customization parameters and configurations (see Section IV for details). We also took advantage of having two (2) 400-level networking-focused courses with low enrollment (less than 10 junior and senior students each). Both are lab-based courses. The first is titled 'Inter-networking,' which makes use of GNS3 [21], and the other is 'Advanced Network Cybersecurity' which makes use of virtual machines (VMs), containers, and hardware. Each delivers six (6) lab exercises in a semester. Since

TABLE 1: Physical Server Specifications

Item	Description
Hardware Model	Precision Tower 3420
Memory	32 GB
Processor	Intel(R) Core(TM) i7-7700 CPU @ 3.60GHz (cores=4, threads=8, 8 vCPUs)
Disks Capacity	1.38 TiB
OS Name	Ubuntu 22.04.3 LTS (Jammy Jellyfish)
OS Type	64-bit

we did not have access to a physical server with sufficient RAM and processing power for 20 students, we decided to dedicate ten (10) desktops for rolling out our experiment. This approach, with our limited experience in running OpenStack deployment, made it easy for us to troubleshoot issues on one deployment without impacting the others. The specifications of the physical machines (desktops) are provided in TABLE 1. On each machine, we planned to have a full OpenStack AIO deployment dedicated to each 2-student group. We also planned on performing the deployment once on one machine and cloning it on the others (see further details in Section IV).

IV. Solution Implementation

In this section, we will present the implementation and deployment of the OpenStack AIO platform on 10 desktops. kolla-ansible requires a few networking options to be set. It needs at least two (2) network interfaces on each machine. The first interface is the default interface for multiple management-type networks, including access to the OpenStack Dashboard (Horizon) locally and remotely. We assigned this interface a static private IPv4 address (from the Internal Access Point AP1 DHCP server). The second required interface is dedicated to Neutron external (commonly referenced as provider or physical) networks. This interface is active without an IPv4 address. If not, instances won't be able to have access to the external networks. This interface configuration allows us to select a pool of IPv4 addresses (known as floating IPv4 addresses and

managed by the External Access Point AP2 DHCP server). See Fig. 4 for our solution network arrangement.

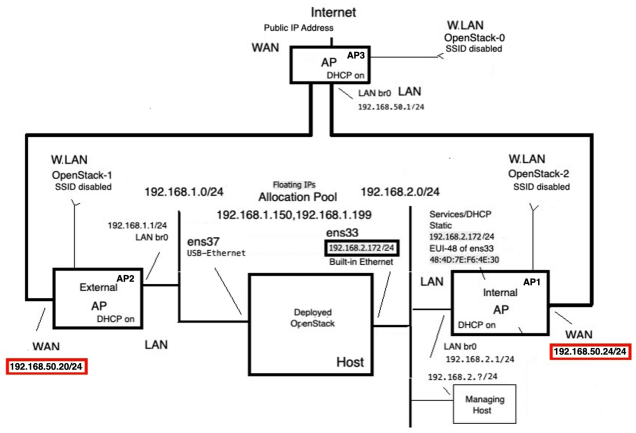


Fig. 4: OpenStack Private Cloud AIO Network Arrangement.

To access the OpenStack deployment's internal network and external network (via floating IPv4 addresses associated with the instances) over the internet, we decided to use a third Access Point AP3. The WAN interfaces of the internal and external APs, AP1 and AP2, respectively, are terminated on LAN ports of AP3 and assigned private static IPv4 addresses from AP3's DHCP server. This arrangement enabled us to make use of a single public IPv4 address assigned to the WAN interface of AP3 for multiple deployments through the implementation of port forwarding on all three APs.

With the well-documented kolla-ansible procedure for OpenStack AIO deployment given in [22], we were able to have a deployment up and running in less than 20 minutes. The critical task is the selection of the deployment customization parameters in the 'globals.yml' file. The 'globals.yml' is the main configuration file for kolla-ansible. There are a few options that are required to deploy OpenStack AIO by kolla-ansible [22]. Here are the minimal configuration parameters we populated in the 'globals.yml' to get started. The full list of the 'globals.yml' parameters can be checked out in [23].

```
---
#Base
workaround_ansible_issue_8743: "yes"
kolla_base_distro: "ubuntu"
openstack_release: "2023.1" # Antelope
#Networks
kolla_internal_vip_address: "<IPv4 addr>"
network_interface: "ens33" # internal
neutron_external_interface: "ens37"
#Services
enable_neutron_provider_networks: "yes"
 #Public Provider
enable_haproxy: "no"
nova_compute_virt_type: "kvm"
```

In the early stage of our experimentation, we populated the 'kolla_internal_vip_address' parameter with an explicit IPv4 address, which is the same as that assigned statically to the internal network interface (ens33 in our case). To secure access to the OpenStack Horizon Dashboard using SSL/TLS, the 'globals.yml' was then updated to include the parameters given below.

```
---
#Base
workaround_ansible_issue_8743: "yes"
kolla_base_distro: "ubuntu"
openstack_release: "2023.1" # Antelope
#Networks
kolla_internal_fqdn: "mykolla.net"
kolla_internal_vip_address: "<IPv4 addr>"
network_interface: "ens33" # internal
kolla_external_fqdn: "mykolla.net"
kolla_external_vip_address: "<IPv4 addr>"
neutron_external_interface: "ens37"
#Services
enable_neutron_provider_networks: "yes"
enable_haproxy: "yes"
nova_compute_virt_type: "kvm"
#TLS
kolla_enable_tls_internal: "yes"
kolla_enable_tls_external: "yes"
kolla_copy_ca_into_containers: "yes"
kolla_enable_tls_backend: "yes"
openstack_cacert:
    "/etc/ssl/certs/ca-certificates.crt"
kolla_admin_openrc_cacert:
    "/etc/ssl/certs/ca-certificates.crt"
```

The key concepts that enabled us to successfully deploy SSL/TLS on OpenStack are: (a) enabling load balancing using HAProxy and (b) assigning an additional virtual IPv4 address (VIP) to the internal network interface [24]. HAProxy provides load balancing services and SSL/TLS termination when hardware load balancers (like in our case) are not available for high availability OpenStack-ansible deployment options. Enabling HAProxy for SSL/TLS termination required us to assign a second IPv4 address to the internal network interface, known as internal VIP, which is different from that assigned to the physical internal network interface (ens33 in our case). Also, we executed the init-runonce script which downloads a CirrOS image and registers it. Then it configures networking including the provider/physical network.

Once we had the OpenStack AIO deployment up and running on our first machine, we created two OpenStack projects and a user account on each project. We also created an internal network and a router between the internal and the admin created provider network for each user and allocated a specific range of floating IP addresses for each project. However, they were not granted administrative privileges. For example, they do not have control over the external network. The administrator account, reserved for the educator or instructor, has visibility and control over all projects and users. With that arrangement and the physical machines residing on our institution's premises, the

students had access to their assigned accounts as long as they were connected to the institution's internal network. To allow remote access to these deployments, we were granted Virtual Private Network (VPN) access for our students to the public IPv4 address (and a corresponding domain name) assigned to the WAN interface of the third Access Point AP3. In addition, we were granted access to a range of transport (TCP/UDP) ports needed for the following access and associated port forwarding rules (the example below is for one physical machine):

(a) OpenStack Horizon Dashboard account login:
AP3 - <WAN IP address>:4431 to 192.168.50.24:4431,
AP1 - 192.168.50.24:4431 to 192.168.2.240:443.

(b) Instance/VM console login:
AP3 - <WAN IP address>:6081 to 192.168.50.24:6081,
AP1 - 192.168.50.24:6081 to 192.168.2.240:6080.

(c) SSH admin access to the physical machine/desktop:
AP3 - <WAN IP address>:2201 to 192.168.50.24: 2201,
AP1 - 192.168.50.24:2201 to 192.168.2.240:22.

(d) Remote access (using floating IPv4 addresses on the external network) to the instances/VMs network interface:
Authentication to FreeRADUIS server instance
AP3 - <WAN IP address>:7181 to 192.168.50.20:7181,
AP2 - 192.168.50.20:7181 to 192.168.1.81:1812.
SSH to instance
AP3 - <WAN IP address>:2211 to 192.168.50.20:2211,
AP2 - 192.168.50.20:2211 to 192.168.1.81:22.
The IPv4 address 192.168.1.81 in this item is the floating IPv4 address associated with the instance running the ssh and radius servers.

The default port number for the instance console is 6080. Since we have used port forwarding to remotely access the instance console, the novncproxy_base_url must be redefined through Nova configuration (/etc/kolla/config/nova.conf) as follows:

```
[vnc]
novncproxy_base_url = https://
<deployment public IPv4 address>:6081/
vnc_lite.html
```

One of the challenges we faced early on in completing the OpenStack AIO deployment on the first machine/desktop is acquiring or creating OpenStack base images. These images are managed under a service known as Glance (see Section III), which is deployed by default, along with Keystone, Horizon, Nova, and Neutron components. Our goal was to create the smallest possible base image due to the limited resources we have on the physical machines/desktops. That's why it is recommended to use the network installation ISO. One can always install missing packages and create an updated image using the snapshot functionality of OpenStack. We successfully followed the procedure described in [25]. Once we had a server version (no GUI) image, we then added the desktop package for the desktop version. We prepared Glance base images for Ubuntu Server and Desktop releases 18.04, 20.04, and 22.04. In case you do not have resource limitations that mandate a smaller image footprint, you can leverage the already available Ubuntu Server cloud images, which you can find in [26].

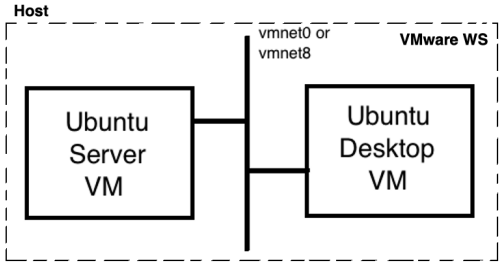
The next challenge we faced in rolling out the OpenStack deployment was to clone the deployment created on the first machine to the remaining machines, using tools like Clonezilla [27]. In order to accomplish the cloning process successfully, we had to replace the 'kolla_internal_vip_address' parameter with the 'kolla_internal_fqdn' parameter. This was our solution to the issue we faced, where currently there is no means in the kolla-ansible framework to reconfigure a deployment with a new static IPv4 address for the internal network interface. That is, the IPv4 address of the internal network interface cannot be reconfigured using kolla-ansible's 'reconfigure' or 'deploy' utilities. Rather than making the IPv4 address change directly in the 'globals.yml', we made the change to the '/etc/hosts' file by updating the IPv4 address corresponding to the 'mykolla.net' hostname. This approach was successful as long as the RabbitMQ and MariaDB containers were healthy and running. In some cases, we had to restart the RabbitMQ and/or MariaDB containers. This saved us a considerable amount of time and resources in rolling out our experiment.

V. Migration of Lab Exercises

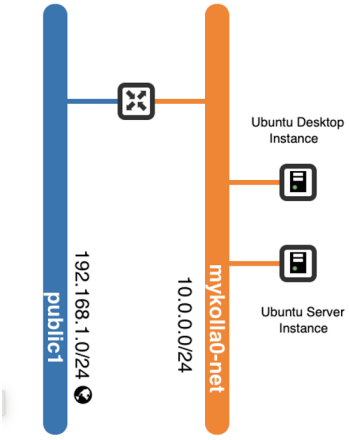
In this section, we will detail how we migrated a number of existing hands-on networking related lab exercises from being Linux native or VMware WS based to our Private Cloud Platform, OpenStack AIO deployment. This includes one lab that required external hardware integration to an OpenStack instance running FreeRADIUS server. We were able to successfully have 11 out of the 12 lab exercises (6 lab exercises of each course) migrated to OpenStack private cloud. To accomplish that required significant lab instructions and procedure updates. And in the case involving hardware integration required new network setup. These are described below. The 12th lab exercise required far more vCPUs, Disk space and RAM resources than available on the hardware available to us and specified in TABLE 1. We will start with what we found as best practices for the educators and the students in the process of migrating and conducting the lab exercises. These are:

A. Best Practices

- It is recommended that the admin/educator provide the necessary Glance images and make them shareable with all projects, and designate them for public use.
- Once students launch an instance, they should check internal and external network connectivity, as well as name resolution availability.
- Students are encouraged to use a CirrOS image and instance to check the health of the platform.
- Before creating new OpenStack instances, students should ensure that all previously created instances are shelved to release their allocated vCPUs.
- Students should be instructed to take snapshots of their instances at critical points for potential restoration.



(a) Using VMware WS.



(b) Using OpenStack Private Cloud.

Fig. 5: IT 460 Lab 1 Network Arrangement.

- The educator, acting as an admin, should create student projects, usernames, and passwords, and ensure that students change their passwords.

B. IT 460 Lab Exercises

Lab 1: Network Security OpenSSL

In this Lab Exercise, students were provided with the necessary credentials to log into the Horizon Dashboard. They were also instructed on how to: (a) change their account password, (b) navigate to their project, (c) create internal network/subnet, (d) create a router between the internal and external networks, (e) create their first instance using the CirrOS image provided, (f) log into the CirrOS instance and (g) check connectivity to the Internet and the success of name resolution (e.g., www.google.com). CirrOS is a minimal Linux distribution designed for use as a test image on clouds including OpenStack Compute [28]. Next, students were instructed to shelve the CirrOS instance to reclaim the vCPUs, disk space, and RAM resources, and then created two instances: ubuntu-desktop-20-04 and ubuntu-server-20-04 (see Fig. 5), using m1.medium flavor (2 vCPUs, 40 GB Disk, and 4096 MB RAM.)

When creating an instance (virtual machine), it's crucial for both the educator and students to utilize the cloud-init service [29] for instance customization. Cloud-init is the service installed inside the instance, and cloud-config consists of a set of scripts executed as soon as the instance starts. Cloud-config provides the language for the scripts that cloud-init executes. For example, we used cloud-config to enable password authentication for a given user, configure hostname, install packages on an instance, run scripts, and adjust instance console screen resolution.

In this lab, we encountered an issue regarding the instance hostname. Although the instance was assigned the name 'CompanyServer.com' during the creation process, when we launched and logged into the instance, the hostname displayed was 'CompanyServer-com'. This behavior is not typical for the Ubuntu Linux distribution. However, it did not impact the outcome of executing the lab exercise.

Lab 2: Network Access Control – User Authentication - What is Kerberos?

In this lab exercise, we encountered the challenge of assigning a static IPv4 address to a network port on an instance. To address this, we followed a procedure similar to that described in [30]. Specifically, we configured networking ports in OpenStack with pre-assigned IPv4 addresses and launched the instances using the '--port' option instead of the '--network' option. Additionally, in our case, name resolution did not function until we modified the DNS Name servers list in the subnet details to include the gateways of the internal and external networks, as well as 8.8.8.8, in that specific order. Fig. 6 (a) and (b) illustrate the network arrangement of this exercise, before and after migration, respectively.

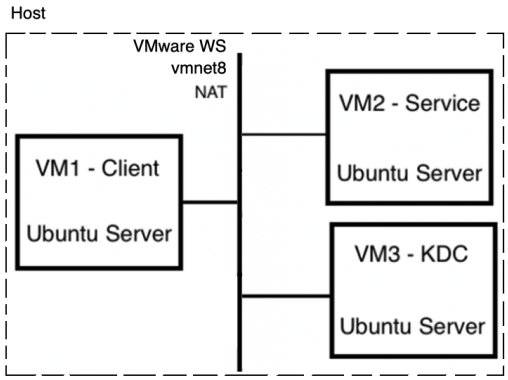
Lab 3: Network Security Boundary Protection - What is VPN?

Fig. 7 (a) and (b) depict the network arrangements outlined in the lab instructions for Lab 3 before and after migration, respectively. In this lab exercise, as well as another titled 'What is a Firewall?', we encountered an issue while setting up gateway and firewall-based instances. These instances function as packet pass-through points, meaning the packets do not originate from or are destined to the instance itself. By default, Neutron, the OpenStack Network Component, enforces port security on a per-port basis. This includes:

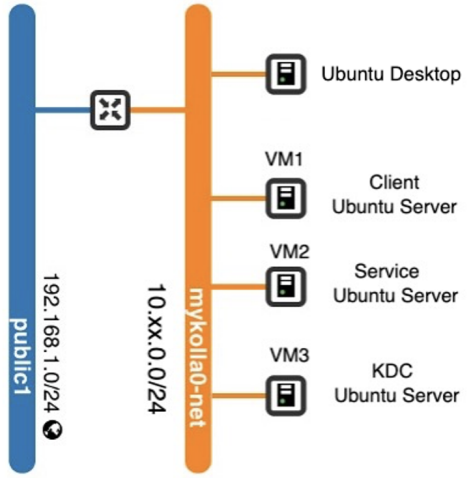
- **Security Groups** - By default, all incoming and outgoing traffic is blocked for ports connected to an instance unless a 'Security Group' has been applied [31].
- **Anti-Spoofing** - Neutron's security group implementation includes anti-spoofing rules that prevent a VM from sending or receiving traffic with a MAC/EUI-48 or IP address that does not belong to its Neutron port [32].

To address this in the lab exercises, we needed to disable port security (packet filtering) [32] on certain ports of the gateway and firewall instances, such as the ports on M1 and M3 (refer to Fig. 7).

Lab 4: Network Access Control – User Authentication -



(a) Using VMware WS.



(b) Using OpenStack Private Cloud.

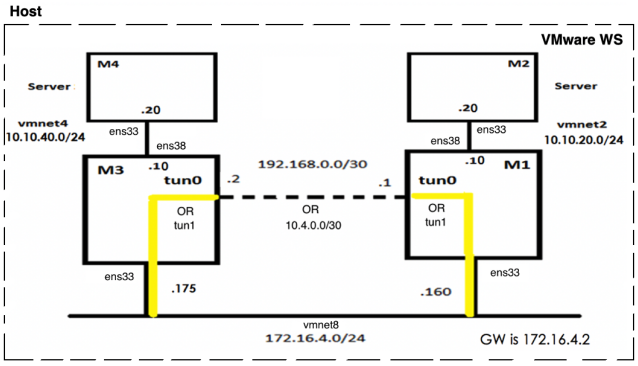
Fig. 6: IT 460 Lab 2 Network Arrangement.

What is 802.1X?

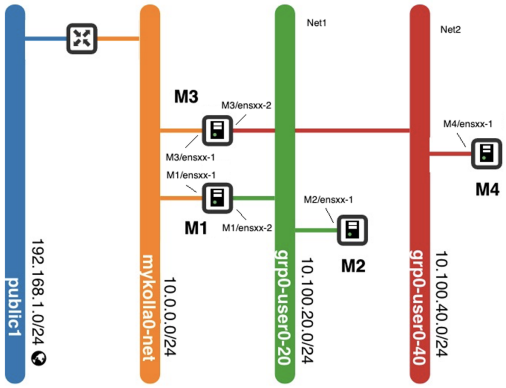
One of the most intriguing challenges we encountered involved integrating hardware devices, such as a Wireless 802.11 Access Point (AP), with the OpenStack platform. Lab 4, which focused on the 802.1X topic, required integrating the authentication server (a FreeRADIUS implementation) running on an OpenStack instance with an AP.

Unlike the pre-migration network arrangement (refer to Fig. 8 (a)), in the post-migration network arrangement (see Fig. 8 (b)), the AP LAN and the FreeRADIUS server were in different networks. We managed to interconnect them through a Ubiquiti Networks EdgeRouter X (U-ERX) Router [33].

Additionally, we had to add port forwarding rules to AP3 and AP2 and a firewall rule in the group security of the project to allow ingress traffic to UDP/1812 port. It is worth noting that we are currently exploring a solution that would enable us to connect a WAN port on the AP to the FreeRADIUS server over the Internet without the need for the U-ERX router.

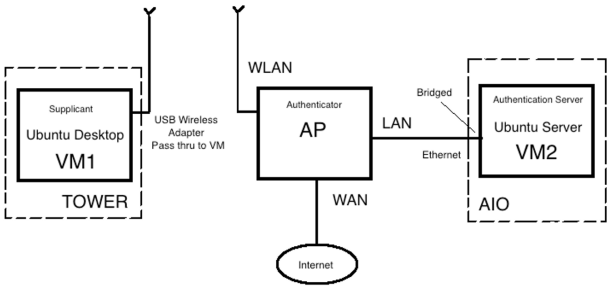


(a) Using VMware WS.

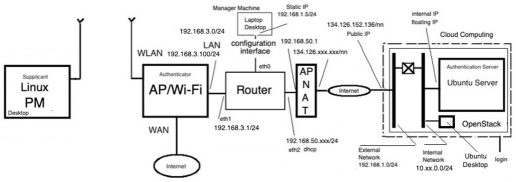


(b) Using OpenStack Private Cloud.

Fig. 7: IT 460 Lab 3 Network Arrangement.



(a) Using VMware WS.



(b) Using OpenStack Private Cloud.

Fig. 8: IT 460 Lab 4 Network Arrangement.

C. IT 461 Lab Exercises

For the IT 461 course, we encountered two main migration challenges: (a) transitioning from running GNS3 on a native Linux physical machine to running it on a virtual machine/instance, and (b) moving from integrating GNS3 with VMware WS (on the native Linux host) to an OpenStack virtualization environment. We successfully prepared an OpenStack image (Ubuntu Desktop 20.04 based) with GNS3 (release 2.2.44) installed and created a baseline instance with sufficient resources using the m1.large flavor: 4 vCPUs, 80 GB Disk, and 8192 MB RAM, thus completing the first migration. For the second migration, Lab 2 and Lab 6 required specific upgrades to function effectively in the OpenStack environment.

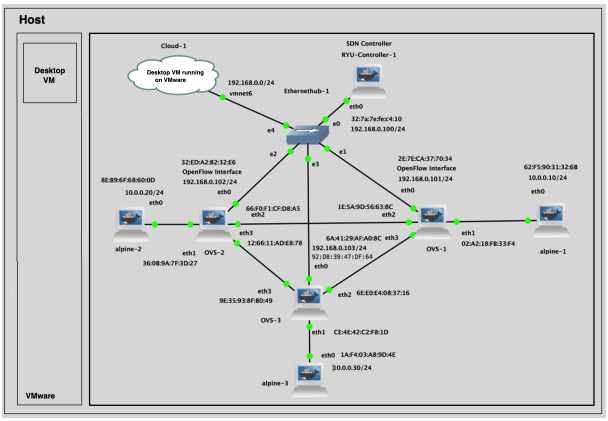
Lab 2: CIDR/VLSM, Static and RIP Dynamic Routing & Redistribution

In this lab exercise, we introduced students to the Cisco Cloud Services Router (CSR) 1000V [34]. The Cisco CSR 1000V Cloud Services Router is a cloud-based virtual router deployed on a virtual machine (VM) on x86 server hardware. It supports a subset of Cisco IOS XE software features and technologies, offering Cisco IOS XE security and switching features on a virtualization platform [35].

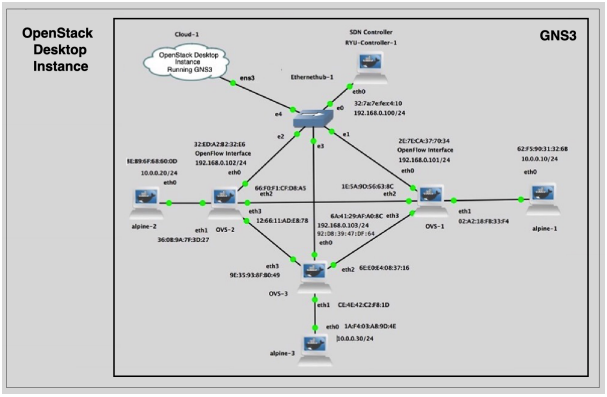
Using the qemu supported emulation environment within the GNS3 application, we successfully installed, created, configured, launched, and ran a GNS3 CSR-1000V object using the Cisco CSR v10000 qcow2 image. This demonstrates nested virtualization. To enable this functionality, the physical machine (referred to as HOST), where the OpenStack deployment is hosted, must have Intel VT-x or AMD-V enabled in the BIOS. Additionally, the HOST must support KVM. We installed KVM using the following commands: 'sudo modprobe kvm_intel' module and 'sudo apt install virt-manager' package.

Lab 6: SDN – Software Defined Network – Part II

In Lab 6, which focused on SDN, we previously launched a desktop VM on VMware WS running on the HOST, which was then integrated with the GNS3 environment using a GNS3 Cloud object. However, in the OpenStack environment, we eliminated this step, saving resources and time. With OpenStack, we gained an advantage over VMware WS because we could achieve access to a Desktop VM for Wireshark packet capturing through a GNS3 Cloud object by associating the Cloud object with the OpenStack instance running the GNS3 server. Fig. 9 (a) and Fig. 9 (b) illustrate the pre-migration and post-migration environments, respectively.



(a) Using VMware WS.



(b) Using OpenStack Private Cloud.

Fig. 9: IT 461 Lab 6 Network Arrangement.

VI. Discussion, Conclusion & Future Work

A. Discussion

The primary intent of this paper is to share with other educators (a) our approach of deploying the private cloud computing platform, OpenStack, as a replacement and enhanced virtualization environment over the current standalone VMware Workstation, (b) the procedures and results of migrating eleven (11) lab exercises from the current virtualization environment to the cloud computing based environment, and (c) the reflections and observations of the students who conducted and executed these lab exercises using OpenStack. Below are some verbatim statements made by the fifteen (15) senior and junior IT students who participated in the project during Spring 2024 semester, in response to a request to share their experiences with OpenStack.

'OpenStack enabled not only more efficient work but also higher quality outcomes, as I could better manage my time with the flexibility to work remotely.'
'I found it to be extremely easy to use and very robust.'
'I greatly improved my knowledge when it comes to networking. OpenStack's 'network topology' capabilities allowed me to visualize the network setup I was interacting with.'
'OpenStack's visualization of all of its components include but are not limited to: instances, vCPUs, RAM, Security groups, Networks, etc. allowed me to learn how to manage my resources when creating instances.'

'I would have never thought that we would be able to get to the point this semester where the interaction between an instance on the OpenStack environment and physical hardware components would be capable.'
'It has allowed me to better manage my time and tasks. OpenStack has given me the benefit of executing labs remotely, allowing me leverage as to when and where I can complete my assignments.'

B. Conclusion and Future Work

In this paper, we have demonstrated the viability of using OpenStack Private Cloud Computing platform to migrate lab based networking and network security lab exercises from standalone VMware WS virtualization environment to OpenStack private cloud computing platform . This enabled the students to perform the lab exercises from anywhere without the constraint of having to always be in the computer lab on campus. We also have demonstrated how straightforward the deployment of OpenStack using the kolla-ansible project. We believe that this experiment paves the way for us to advance our IT undergraduate curriculum to create and offer lab exercises on the inner workings, management and administration of the cloud computing architecture and services. Below is a brief list of topics that we see ourselves pursuing in the near future:

- Introduce OpenStack to large-sized classes and conduct quantitative assessments of its effectiveness as an alternative to VMware WS standalone environment.
- Roll out an OpenStack multi-node deployment.
- Introduce additional OpenStack services such as Zun Containers, Magnum Kubernetes/Container Orchestration, Cinder Block Storage/Volumes, Swift Object Storage, or Ceph Block, Object, and File Storage.

VII. Acknowledgment

The author would like to thank the College of Science and Engineering (CISE) and Computer Science Department at James Madison University (JMU) for their support throughout the development and migration of the lab exercises and the creation of this paper. Special thanks to Casey Alexander and Katherine Botticelli for their precursor work on the OpenStack FA23 Capstone Project, and the students in the IT 460 and IT 461 classes of Spring 2024.

References

[1] "The Most Widely Deployed Open Source Cloud Software in the World," https://www.openstack.org/, May 18, 2024.
[2] "Kolla Ansible," https://github.com/openstack/kolla-ansible, Nov 2016.
[3] "VMware Desktop Hypervisors," https://www.vmware.com/products/desktop-hypervisor.html, May 18, 2024.
[4] S. Bonner, C. Pulley, I. Kureshi, V. Holmes, J. Brennan, and Y. James, "Using OpenStack to improve student experience in an H.E. environment," in *2013 Science and Information Conference*, 2013, pp. 888–893.
[5] G. Bhatia, I. A. Noutaki, S. A. Ruzeiqi, and J. A. Maskari, "Design and implementation of private cloud for higher education using openstack," in *2018 Majan International Conference (MIC)*, 2018, pp. 1–6.
[6] Kyrre Begnum, "Using OpenStack for Education," https://www.usenix.org/conference/sesa14/summit-program/presentation/begnum_open_stack, Nov 11, 2014.
[7] F. Mechraoui, P. Martins, and F. Caldeira, "Openstack: a virtualisation overview," *Int. J. Inf. Technol. Manage.*, vol. 23, no. 1, p. 1–12, jan 2024. [Online]. Available: https://doi.org/10.1504/ijitm.2024.136181
[8] S. Zhou, J. He, T. Li, X. Lan, Y. Wang, H. Zhao, and Y. Li, "Automating the Deployment of Cyber Range with OpenStack," *The Computer Journal*, vol. 67, no. 3, pp. 851–863, 04 2023. [Online]. Available: https://doi.org/10.1093/comjnl/bxad024
[9] M. Abbasi, F. Cardoso, J. Silva, and P. Martins, "Exploring openstack for scalable and cost-effective virtualization in education," in *New Trends in Disruptive Technologies, Tech Ethics and Artificial Intelligence*, D. H. de la Iglesia, J. F. de Paz Santana, and A. J. López Rivero, Eds. Cham: Springer Nature Switzerland, 2023, pp. 135–146.
[10] Christopher Pappas, "5 Benefits And Challenges Of Cloud Computing In Education," https://elearningindustry.com/benefits-and-challenges-of-cloud-computing-in-education, April 26, 2024.
[11] Huawei Enterprise Support Community, "OpenStack and components," https://forum.huawei.com/enterprise/en/openstack-and-components/thread/667245611331764224-667213860102352896, June 17, 2021.
[12] Ales Nosek, "18 Months with Openstack, Our Experience, Part I," https://alesnosek.com/blog/2018/02/19/18-months-with-openstack-our-experience-part-i/, Feb. 21, 2018.
[13] Sagar Nangare, "Software Deployment Options in OpenStack: A Quick Guide," https://superuser.openinfra.dev/articles/software-deployment-options-in-openstack-a-quick-guide/, February 10, 2023.
[14] "Manage projects, users, and roles," https://docs.openstack.org/keystone/latest/admin/cli-manage-projects-users-and-roles.html, May 4, 2024.
[15] "Manage flavors," https://docs.openstack.org/nova/latest/admin/flavors.html, May 4, 2024.
[16] "Port," https://docs.openstack.org/python-openstackclient/latest/cli/command-objects/port.html, May 12, 2024.
[17] "Security Groups," https://docs.openstack.org/nova/latest/user/security-groups.html, May 12, 2024.
[18] "Manually associating and dissociating floating IPs of instances – OpenStack," https://mindmajix.com/openstack/manually-associating-dissociating-floating-ips-tenants, May 4, 2024.
[19] "OpenStack Networking," https://docs.openstack.org/neutron/pike/admin/intro-os-networking.html, May 12, 2024.
[20] Huawei Enterprise Support Community, "OpenStack Networking," https://docs.openstack.org/neutron/2024.1/admin/intro-os-networking.html, July 5, 2023.
[21] "The software that empowers network professionals," https://www.gns3.com/, May 12, 2024.
[22] "Quick Start for deployment/evaluation," https://docs.openstack.org/kolla-ansible/2023.1/user/quickstart.html, April 2024.
[23] "kolla-ansible/etc/kolla /globals.yml," https://github.com/openstack/kolla-ansible/blob/master/etc/kolla/globals.yml, April 2024.
[24] "Configuring HAProxy," https://docs.openstack.org/openstack-ansible-haproxy_server/2023.1/configure-haproxy.html, April 29, 2024.
[25] "Example: Ubuntu image," https://docs.openstack.org/image-guide/ubuntu-image.html, April 28, 2024.
[26] "Ubuntu Cloud Images (RELEASED)," https://cloud-images.ubuntu.com/releases/, April 28, 2024.
[27] "Clonezilla The Free and Open Source Software for Disk Imaging and Cloning," https://clonezilla.org/, April 29, 2024.
[28] "Get images," https://docs.openstack.org/image-guide/obtain-images.html, May 15, 2024.
[29] Aditya Bhuyan, "A Comprehensive Guide to Cloud-Init: Automating Cloud Instance Initialization," https://faun.pub/a-comprehensive-guide-to-cloud-init-automating-cloud-instance-initialization-2af37516f81e, Oct 10, 2023.
[30] "Static Addresses in OpenStack Networking," https://www.madboa.com/blog/2023/05/17/openstack-static/, April 29, 2024.
[31] "Managing Port Level Security in OpenStack," https://superuser.openinfra.dev/articles/managing-port-level-security-openstack/, April 21, 2017.
[32] "Port Security and MAC Spoofing," https://docs.openstack.org/developer/dragonflow/specs/mac_spoofing.html, May 3, 2024.
[33] "EdgeRouter X Datasheet," https://dl.ubnt.com/datasheets/edgemax/EdgeRouter_X_DS.pdf, May 15, 2024.
[34] "Cisco Cloud Services Router 1000v," https://www.cisco.com/c/en/us/support/routers/cloud-services-router-1000v/model.html, May 15, 2024.
[35] "Cisco IOS XE," https://www.cisco.com/c/en/us/products/ios-nx-os-software/ios-xe/index.html, May 15, 2024.