International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 09 Issue: 04 | Apr 2022 www.irjet.net p-ISSN: 2395-0072

Creating a Cloud Architecture for Machine Learning and Artificial Intelligence Applications

Rahul Maurya[1], Jayesh Rane[2], Tanmay Hinge[3], Arjun Nair[4], and Prof. Varunakshi Bhojane[5]

[1,2,3,4]*Student, Department of Computer Engineering, Pillai College of Engineering, Navi Mumbai, Maharashtra, India*

[5]*Professor, Department of Computer Engineering, Pillai College of Engineering, Navi Mumbai, Maharashtra, India*

Abstract - *Many software applications require heavy computing and most individuals cannot afford such an expensive system. Some applications even require a very high amount of resources. Everybody cannot afford such a system, and one does not require a whole setup all the time. We can build a cloud system where multiple systems with powerful Graphical Processing Units (GPUs) can be set up to solve this problem. Any individual who needs a powerful system to run heavy computing software for various purposes like Artificial Intelligence and Machine Learning can access these systems remotely. To build this cloud system, the OpenStack platform will be used. OpenStack is an open-source platform that can be used to deploy cloud. With the help of OpenStack, we can provide virtual services and other resources to users. It also provides a dashboard for seamless management of networking and storage allocation.*

Key Words: Cloud Architecture, Cloud Computing, GPU, Remote Connection, Machine Learning and Artificial Intelligence Applications.

1. INTRODUCTION

Every industry needs various online applications for automation of work and simplification of work-related software. When there is a requirement for an online application, there is a necessary focus on security, management, and data backup. Most online applications store data on the cloud, which is much more secure and accessed from anywhere. In the case of high complexity and high computing applications in the domain of Machine Learning and Artificial Intelligence, individuals need heavy computing systems. Many individuals, especially students, cannot afford a powerful system. Users of such a system do not require full-time access to the whole system, and they either need the whole system for a small duration of time or a small number of resources for a long duration of time, which means users can share the total available resources. The OpenStack platform will be used for deploying the cloud system. OpenStack provides a variety of components and a dashboard to manage every available resource. We can allocate memory, manage networking, create instances and authenticate users using this platform.

2. LITERATURE SURVEY

2.1 Public, Private and Hybrid Cloud Service

Prakash Gopalakrishnan describes various cloud models like Private, Community, Public, and Hybrid cloud [1]. Public cloud is distributed by a third party and provided to various organizations and companies. The resources and systems available in the Private cloud are provided to a single user. A hybrid cloud is a combination of Public and Private clouds; it allows data and applications to be shared.

2.2 Security Assessment of OpenStack

Ionel Gordin, Adrian Graur, Alin Potorac and Doru Balan discussed the security issues with the private cloud [2]. Before any security assessment, it is important to know the network structure in a system. The information provided from this step can help to build a secure system. The system should not be able to get accessed without authorization. Various port scanning tools can be used to determine which ports are open and vulnerable.

2.3 The Analysis of OpenStack

Daniel Grzonka provides information on OpenStack and its various cloud services[3]. Analysis and performance measurement of OpenStack needs to be done to find the right allocation of virtualization resources and setting up. The improvement in the system after applying a particular setting needs to be carried out. A proper analysis of the system helps in identifying the flaws, and rectify those to make an efficient system.

2.4 Openstack Cloud for High Performance Computing

Pavle Ivanovic and Harald Richter discuss an efficient way to perform high performance computing [4]. It is done using a physical shared memory between VMs on the

same host server. It also provides analysis of factors influencing tuning techniques that can increase performance. The more efficient the system is, the more work it is able to perform. Proper tuning results in efficient utilization of the system.

2.5 Summary of Related Work

The summary of methods used in literature is given in Table -2.1.

Table -2.1: Summary of literature survey

SN	Paper Name	Summary
1.	Providing Storage as a Service on Cloud using OpenStack (2017) by Chetan Gaikwad, Bhoomika Churi, Kanad Patil and Tatwadarshi P. N.[5]	This paper discusses storage of data on the cloud. In this, the user has to authenticate themselves, then only they can access their data and perform various actions on it.. Various components of OpenStack like Nova, Swift, Keystone etc are used in this system. The proposed system fulfills the goal of security i.e Confidentiality, Integrity and Availability. The proposed system will provide Storage as a Service on cloud by using OpenStack.
2.	Openstack- Paradigm Shift to Open Source Cloud Computing & Its Integration (2016) by Shubham Awasthi, Anay Pathak, Lovekesh Kapoor. [6]	This paper discusses OpenStack in detail. OpenStack setup up, architecture and functionalities are mentioned in detail. There are various tests conducted for different use cases. This paper revolves around a combination of Public and Private Cloud i.e Hybrid Cloud.
3.	A Comparative Study of OpenStack and CloudStack (2015) by Jaison Paul Mulerikkal,Yed hu Sastri.[7]	This paper majorly discusses the analysis of OpenStack and Cloudstack. Various benchmarking systems like unixbench Unixbench and Bonnie++ benchmarks are used. This paper concludes that OpenStack performs better than CloudStack in a single node environment. OpenStack has better overall stability but is more complex to install than CloudStack.
4.	OpenStack of open source cloud computing in colleges and universities' computer room (2017) by Lei Wang and Dandan Zhang.[8]	This paper discusses open source cloud computing, the main function and architecture of OpenStack. It also discusses various issues of computer labs in colleges and universities. The performance of OpenStack in a university computer lab is determined in this paper.
5.	OpenStack Platform and its Application in Big Data Processing (2015) by Cen Shao, Bo Liang, Feng Wang, Hui Deng, Wei Dai, Shoulin Wei, Xiaoli Zhang and Zhi Yuan.[9]	This paper provides details related to the architecture of OpenStack. High computing requirement of a system such as Big Data processing is mentioned in this paper. Large scale data handling process on OpenStack is also provided in this paper.
6.	A Survey Paper on Cloud Computing (2012) by Shyam Patidar, Dheeraj Rane, Pritesh Jain.[10]	Provides various definitions of Cloud Computing. Architecture and various services like IaaS, SaaS and PaaS are discussed in this paper. Various types of cloud like Private, Public and Hybrid cloud are discussed.

3. PROPOSED WORK

In this section, the flaws in existing system architecture and proposed systems to overcome this architecture will be discussed.

3.1 Existing System Architecture

In the existing system shown in Figure 3.1, Students have to use their specific systems and deploy all their applications physically creating a barrier. Due to pandemic situations, students are not able to use these applications

for their practical uses. On top of that, installing new applications and softwares can be a hassle for first time users.

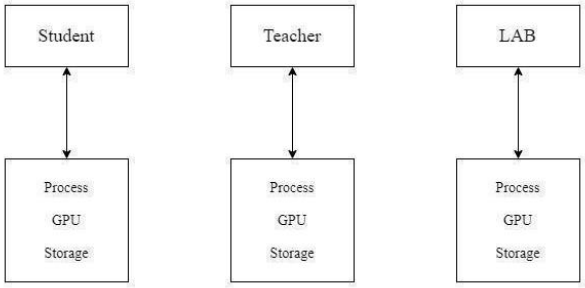

Fig -3.1: Existing System

3.2 Proposed System Architecture

In the proposed system, any entity i.e students, teachers or any user with proper internet connectivity would be able to access the cloud remotely. The proposed system is presented in Figure 3.2.

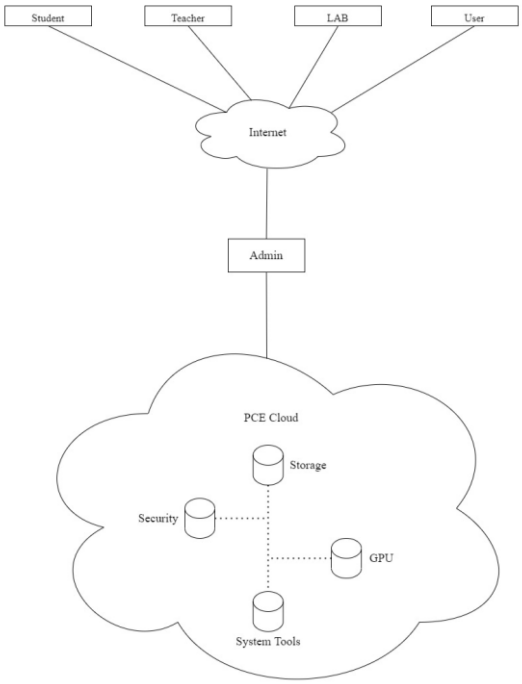

Fig -3.2: Proposed System

For building the cloud system, we will be using OpenStack. OpenStack is an open source cloud management platform. OpenStack helps in managing large amounts of resources like storage, RAM, networking etc.

Now that the cloud system is built in a local server, the system also needs to be opened for accessing remotely. For accessing this local server remotely, we will be using the Ngrok application. Ngrok is an application used to securely open local server ports to the internet.

3.3 Openstack

In this section we will discuss more on the OpenStack software platform. The OpenStack platform is used for deploying the cloud system. It provides a variety of components and a dashboard to manage every available resource like storage, RAM, networking, create instances etc.

3.3.1 Openstack Principles

Open development model: OpenStack is open source software. All the code of OpenStack is freely available

Open design process: Every six months the development community gathers people from various industries to understand the requirements and to make appropriate changes in the software.

Open community: OpenStack has a very active community. All processes are open and transparent.

3.3.2 Openstack Architecture

The OpenStack architecture is presented in this section. The classification of various services provided by OpenStack is given in Figure 3.3.

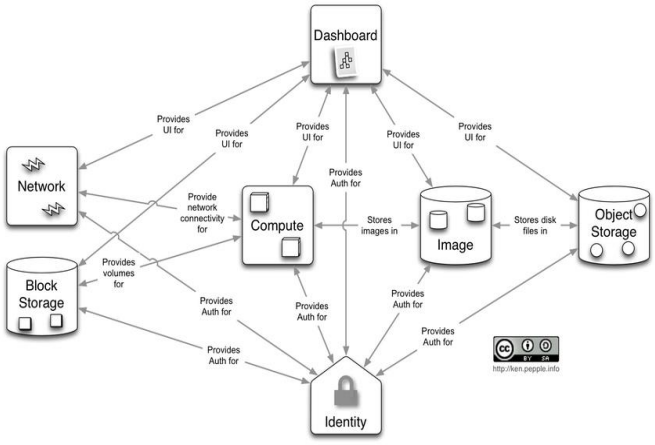

Fig -3.3: Openstack Dashboard Structure [11]

3.3.3 Openstack Components

There are various components shown in Figure 3.4 that are provided by OpenStack for ease of development and management of the cloud services.

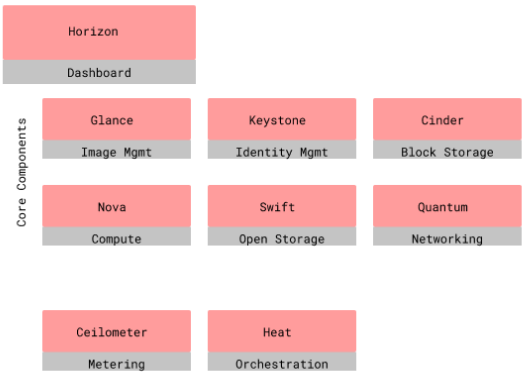

Fig -3.4: OpenStack Components

3.3.3.1 Compute (Nova)

Compute help in managing computing resources and architecture. It provides cloud design flexibility without the need for proprietary software and hardware. It also manages multiple instances that perform various computational tasks.

3.3.3.2 Image Service (Glance)

Image services help in registering, discovering or restoring virtual machines images. It is built upon client server model and provides a user with a REST API that allows them to query virtual machine image metadata as well as retrieve the actual picture. Glance, one of Nova's virtual machine instances, uses the images stored as templates. Virtualization, bare metal, and high-performance compute architectures are used while installing new systems. Glance supports Hyper-V (VHD), VirtualBox (VDI), VMWare (VMDK, OVF), etc.

3.3.3.3 Object Storage (Swift)

Swift helps in creating redundant and scalable data storage. The data can be retrieved and updated as per the need. It helps in scalability and performance. It also helps in organizing data safely and efficiently. Shift also ensures that replication and distribution of data over various systems, which make it cost effective.

3.3.3.4 Dashboard (Horizon)

Horizon is the only graphical interface for cloud-based resources, and it is the only permitted implementation of OpenStack's Dashboard. It provides external services such as monitoring, networking and other management tools for admins.

3.3.3.5 Identity Service (Keystone)

Keystone gives them access to a common list of users who are mapped to all OpenStack services. It connects to current backend services while also serving as a general system for authentication in the cloud. It supports various types of authentication like username with password, token based systems etc. It provides a list of services running in the OpenStack cloud

3.3.3.6 Networking (Neutron)

Neutron is used for networking related tasks. It is used to manage networks and Internet Protocol(IP) addresses. It helps in the creation of networks and connects devices or servers. Neutron provides various extension frameworks, which supports deploying and managing other network services. These other network services include virtual private networks (VPN), firewalls, load balancing, and intrusion detection system (IDS).

3.3.3.7 Block Storage (Cinder)

Cinder helps in determining block level storage devices for applications with OpenStack compute instances. The cloud user can manage their storage needs by integrating block storage volumes with Dashboard and Nova. Cinder can use storage platforms such as Linus server, IBM, etc. It is helpful for expandable file systems and database storage.

3.3.3.8 Orchestration (Heat)

Heat is a prime factor in OpenStack. Without heat, customers can not create cloud additives like instances, networks, and so forth in the form of codes.

3.4 Ngrok

Ngrok is an application which is used for securely opening a local server to the internet. This software makes the locally hosted server appear as if hosted on a subdomain of its website, for example xxxx.ngrok.com. Ngrok is able to bypass NAT Mapping and firewall restrictions by creating a long-lived TCP tunnel from a randomly generated subdomain on ngrok.com (e.g. 3gf892ks.ngrok.com) to the local machine [12]. After specifying the port that your web server listens on, the ngrok client program initiates a secure connection to the ngrok server and then anyone can make requests to your local server with the unique ngrok tunnel address.

3.5 Implementation Details

After gaining some information about Openstack architecture and components, now let's start the

installation and setting up a cloud architecture using the OpenStack platform.

3.5.1 Installation of OpenStack

For installation and configuration of the Openstack system, Devstack is used. DevStack is a series of extensible scripts used to quickly bring up a complete OpenStack environment based on the latest versions of everything.

For smooth installation, Ubuntu Operating System(OS) will be required to be installed on the system. Now, we need to open terminal and run the following commands:

```
sudo apt-get update && sudo apt-get upgrade -y
sudo useradd -s /bin/bash -d /opt/stack -m stack
echo "stack ALL=(ALL) NOPASSWD: ALL" | sudo tee /etc/sudoers.d/stack
sudo su - stack
sudo apt install git -y
git clone https://opendev.org/openstack/devstack
cd devstack
Install vim
sudo apt install vim
vim local.conf
[[local|localrc]]
# Password for KeyStone, Database, RabbitMQ and Service
```

Further, we need to make changes in the local.conf file as per our system's details using any text editing software, here we have used VIM. The changes are as follows:

```
ADMIN_PASSWORD=<Admin Password of your choice>
DATABASE_PASSWORD=<Database Password of your choice>
RABBIT_PASSWORD=<Any Password of your choice>
SERVICE_PASSWORD=<Any Password of your choice>
# Host IP - get your Server/VM IP address from ip addr command
HOST_IP=<Host IP of the system>

FLOATING_RANGE=<Range of IP address of which the system is a part>
FORCE=yes ./stack.sh
```

On successful completion of the above steps, the OpenStack login window will be introduced. Here, the admin can login using credentials setted in local.conf file. The admin will then be introduced to a dashboard with various options to manage various resources like RAM, storage, networking etc.

3.5.2 Resource selection for creating an instance

Now that the cloud system is successfully built, instances will be required. Instance is a small virtual server with various resources running on a cloud. A single cloud can host multiple instances. Multiple users can use multiple instances hosted on the same cloud system. Before creating instances, selection of various parameters or resources will be needed.

After login, the admin will be provided with many options to creating and customizing the resources available. For creating and adding resources to the instance, we need to set various parameters. These parameters consist of compute, volumes and network, each of these are further divided into sub parameters. Parameters and the options selected are mentioned in Table -3.1.

Table -3.1: Setting of various parameters

Parameter	Creation /addition of resources	Sub parameters	Additional settings (if any)
Compute	Creating an Image	OS: Truly Ubuntu Flavor: ds2G	Storage as per admins requirement
Network	Create Network	Enable Admin state and create subnet	Subnet: Systems IP address
Network	Create Router	Enable Admin state	External Network: public
Network	Add interface	subnet: select private net with previously entered IP address	-
Volume	Create Volume	Image: Trusty Ubuntu	Size: Instance size as per users need

3.5.3 Creating a Key Pair for SSH connection

To connect one system to another, we need to establish a SSH (Secure Shell Protocol) connection between two systems. For establishing a SSH connection, we need a Key Pair. For that, the admin needs to go to the 'Key Pairs' option in the 'Compute' parameter.

Save the key pair properly. When establishing an SSH connection, this key can be used. This key is needed to be saved for further process. There are various algorithm used for encryption. For SSH, RSA key with SHA256 is used. RSA is a public key cryptography which is used for secure data transmission. SHA-256 is Secure Hash Algorithm 256-bit and it's used for security in cryptography.

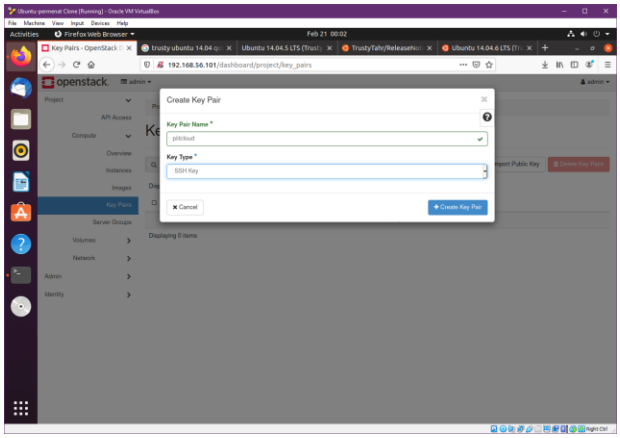

Fig -3.5: Key Pair Creation

The key has been added successfully as we can see in the screenshot below.

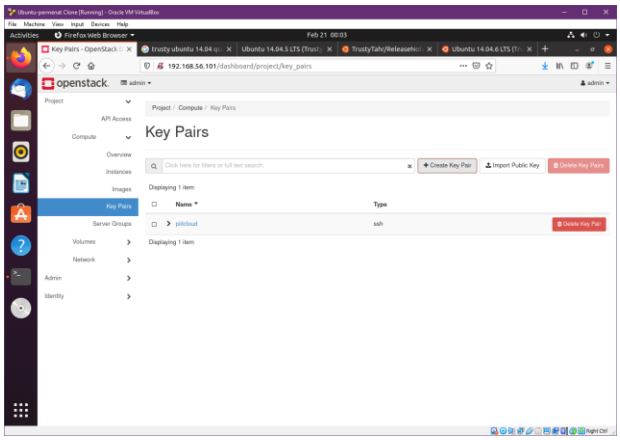

Fig -3.6: Key added successfully

3.5.4 Managing Security Group:-

Now, the admin needs to manage the security group. For that, they need to go to the 'Security groups' option in the 'Networks' parameter.

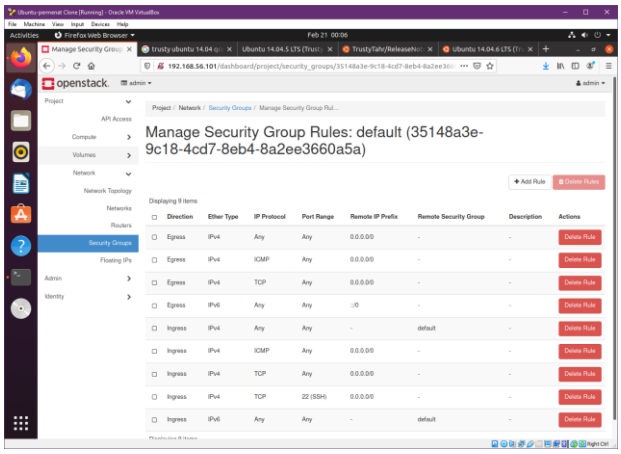

Fig -3.7: Management Menu for Security of Rules

For this project purpose, we have added TCP ICMP rules in both ingress and egress along with custom TCP at port 22.

In this section, there are various terminologies used as following:-

a. ICMP:-

ICMP is Internet Control Message Protocol, it is mainly used by devices like a router. It is used when a communication issue arises and there is a need to communicate with the sender of the data packets.

b. Ingress:-

Ingress is the authority provided to an entity to access a particular system. It means the right to enter a system. In cloud computing, Ingress means giving access to the client from outside the service area to resources within the service area.

c. Egress:-

Ingress is the authority provided to an entity to leave a particular system. It means the right to exit a particular system. In cloud computing, it means to leave a private network to go to the public internet.

d. Transmission Control Protocol(TCP):-

TCP is a transport layer protocol that allows packets to be sent from one location to another. It is used for movement of packets in a network. There is a need to establish a connection before the communication takes place, therefore it's a connection-oriented protocol. TCP collects the data from the application layer. It then separates the data into many packets, assigns an identification number to each packet, and sends the packets to their target location.. TCP, on the other side, reassembles the packets before sending them to the application layer. Until the

whole process of sending and receiving is completed, the connection will be maintained.

3.5.5 Launching an Instance:-

Various resources are created, now allocation of these resources to the instances will take place. The creation of resources have already been done in previous steps, now only allocation will take place.

Now, the admin needs to launch an instance. For that, they need to go to the 'Instances' option in the 'Compute' parameter.

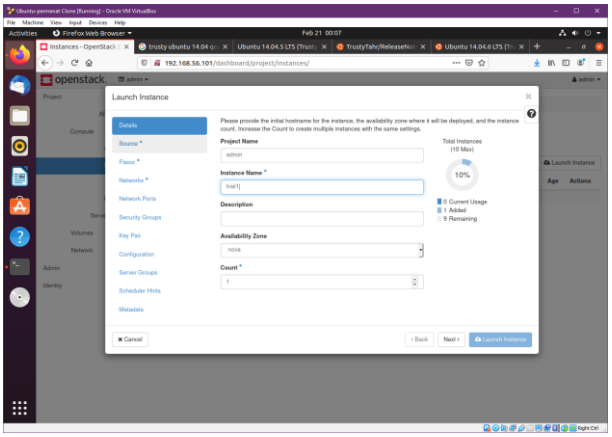

Fig -3.8: Instance Launch

Now, the admin needs to allocate an image, here 'Trusty Ubuntu' is used.

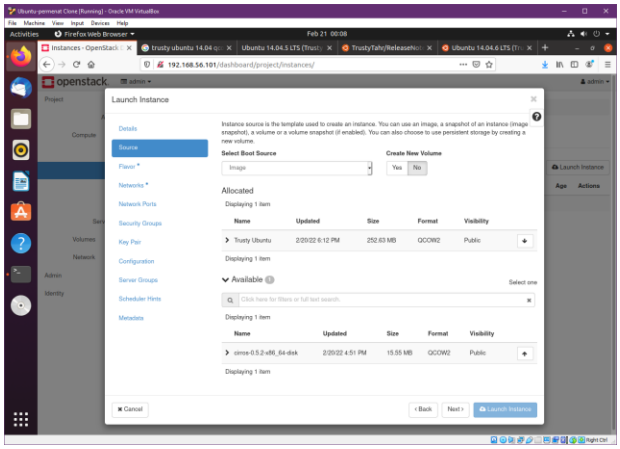

Fig -3.9: Allocation of Image

For demonstration purposes we have allocated the flavor as 'ds2G', as shown in the image below. The admin can use any flavor of their choice.

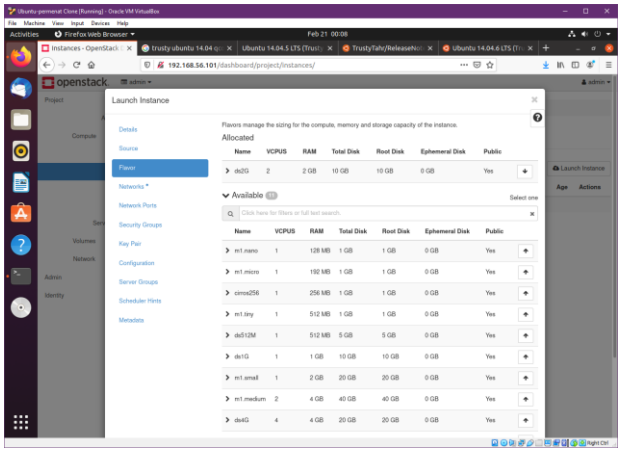

Fig -3.10: Flavor allocation

Allocate a network. Network provides the communication channel for instrances in the cloud.

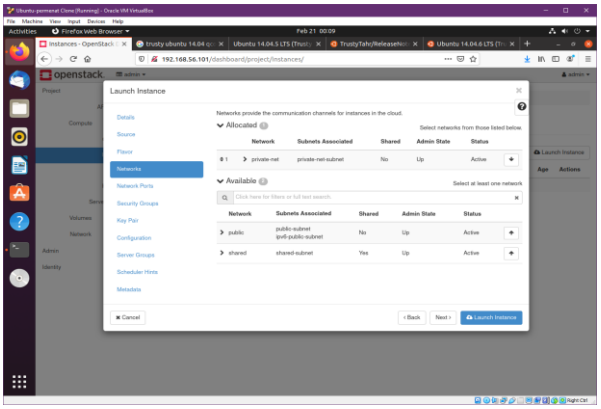

Fig -3.11: Network Allocation

Set the default Security Group as previous settings which we have configured before.

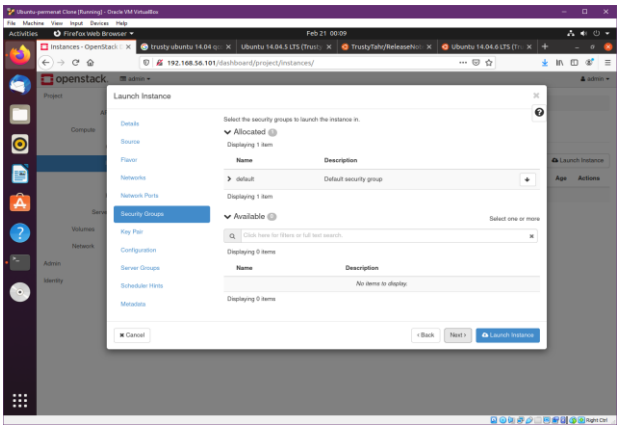

Fig -3.12: Security Group set to default

Set the right key pair as shown in the image below. A key pair allows us to SSH into newly created instances.

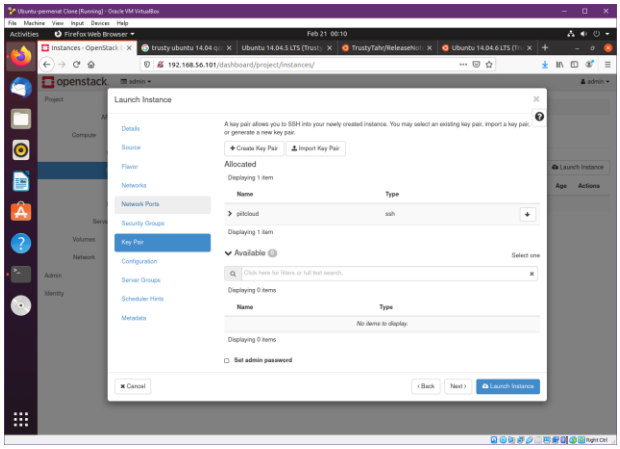

Fig -3.13: Key Pair Setting

Instance has been added successfully, which can be accessed by clicking on the instance name and then going to the console window.

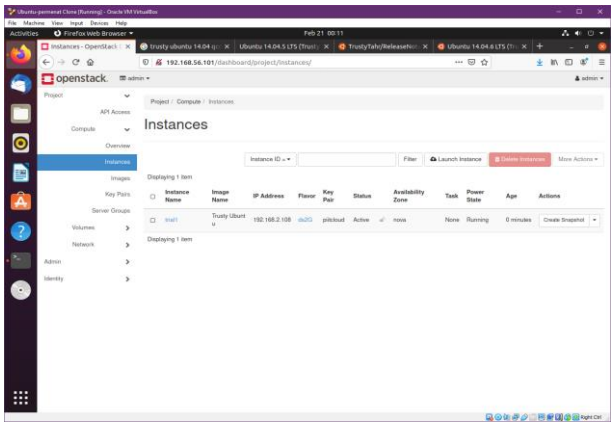

Fig -3.14: Successfully added the instance

3.5.6 Allocating Floating IP

Now, the admin needs to allocate floating IP. For that, they need to go to the 'Floating IPs' option in the 'Networks' parameter. Admin needs to select Public Pool.

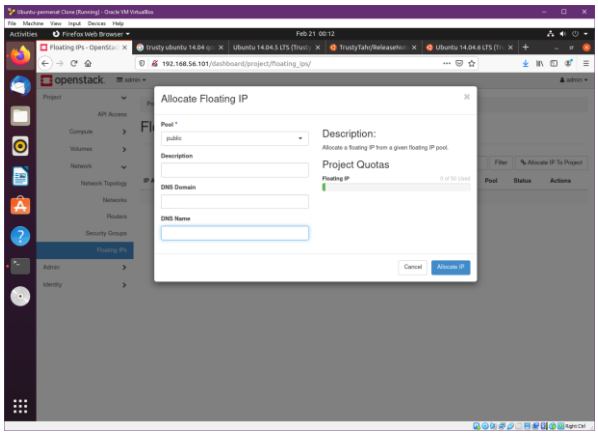

Fig -3.15: IP allocation

Set the IP Address as the admin would like to associate with their selected instance or port.

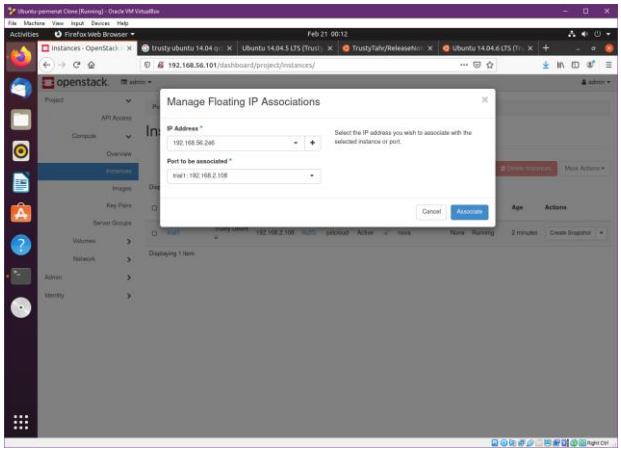

Fig -3.16: Given IP Address set to admin permissions

3.5.7 IP Forwarding

When you want the system to operate as a router and move IP packets from one network to another, you should enable IP forwarding.

Consider a server with two physical ethernet ports, each of which is intended to connect to two distinct networks (say your internal network and the outside world as provided by a DSL modem). The system can communicate on either network if you merely connect and set up those two interfaces. However, because forwarding is disabled, packets from one network cannot transit to the other.

Consider the case of the 'route add' command. You'll need at least two routes, one for each network interface, if you have two network interfaces. When the kernel is deciding where to send a network packet, it will choose the most particular route and deliver it to that interface.

If forwarding is off, the kernel will first determine which interface the packet originated from. The kernel will discard it if it does not originate from the same interface.

IP Forwarding commands:

```
sudo bash
echo 1 > /proc/sys/net/ipv4/ip_forward
echo 1 > /proc/sys/net/ipv4/conf/br-ex/proxy_arp
echo 1 > /proc/sys/net/ipv4/conf/enp4s0/proxy_arp
iptables -t nat -A POSTROUTING -o enp4s0 -j
MASQUERADE
```

Reviving devstack in account of System Failure:
sudo systemctl restart devstack@*

3.5.8 Connecting with the instance:-

The admin has now successfully setted up the cloud, but now the admin needs to establish a connection with the instance. For establishing a connection to the instance, SSH will be used. This process will be done on the system on which OpenStack cloud is built. The SSH connection to the instance and requirement to use Ngrok to open a port for remote connection by the user will be fulfilled by the following commands:

sudo apt update
sudo apt upgrade
sudo apt install python3-pip
python3 -m pip install jupyter
sudo su
python3 -m pip install jupyterlab
sudo apt install jupyter core
ssh -i ~/.ssh/authorized_keys <username>@<Host IP>
jupyter notebook --no-browser --port=<any port> --ip=0.0.0.0
wget https://bin.equinox.io/c/4VmDzA7iaHb/ngrok-stable-linux-amd64.tgz
sudo tar xvzf ngrok-stable-linux-amd64.tgz -C /usr/local/bin
ngrok http <Host IP>:<Port Number>
./ngrok http <Host IP>:<Port Number>

Pillai College of Engineering(PCE) Cloud Notebook 1
Link: <example.ngrok.io format link received in previous step>

Token: <token from jupiter>

Authentication : ngrok authtoken <Enter authentication code received from ngrok>

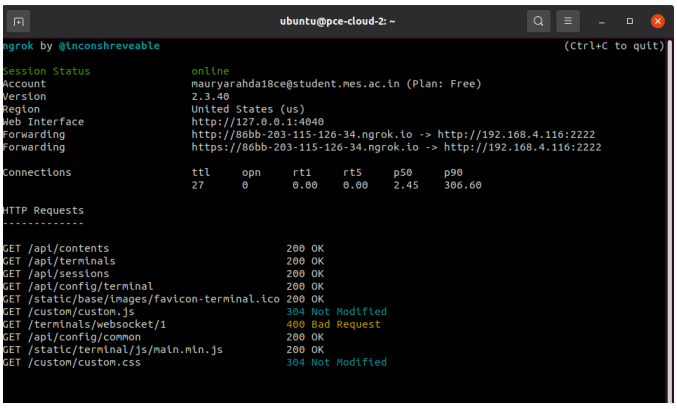

Fig -3.17: Ngrok Application

Similarly many instances and corresponding notebooks can be created. The link to these instances can be incorporated into a simple web application with buttons for different instances as shown in Figure 3.18. A token can be used as the password.

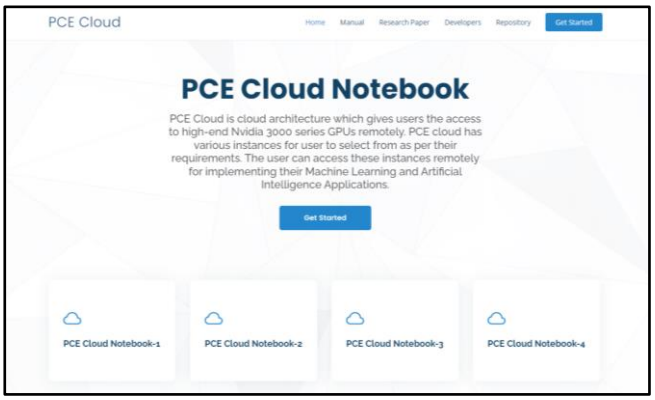

Fig -3.18: Jupyter Notebook Login Screen

The button 'Get Started' gives instructions to users on connecting the admin. The user can click on the button 'PCE Cloud Notebook -1 ' and enter the token on the next screen as shown in Figure 3.19. Users receive the token from the admin to get access to the Jupyter notebook. The users can run their Machine Learning and Artificial Intelligence application on this Jupyter Notebook remotely.

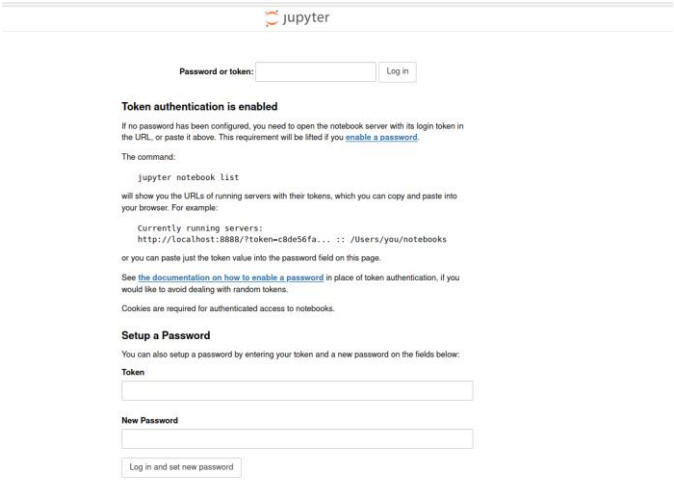

Fig -3.19: Jupyter Notebook Login Screen

3.6 Use Case Diagram

The Figure 3.20 below shows the overall outline of the system, admin and its users[13].

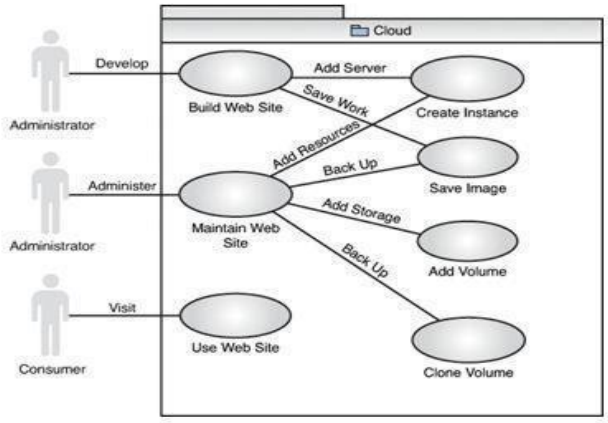

Fig -3.20: Use Case Diagram

3.7 Flow Diagram

The Figure 3.21 below shows the outline of the cloud system built. The user will visit the link and ask the admin for the password (token). After successful login, the user will be introduced to the Jupyter notebook for running the Machine Learning and Artificial Intelligence application of their choice.

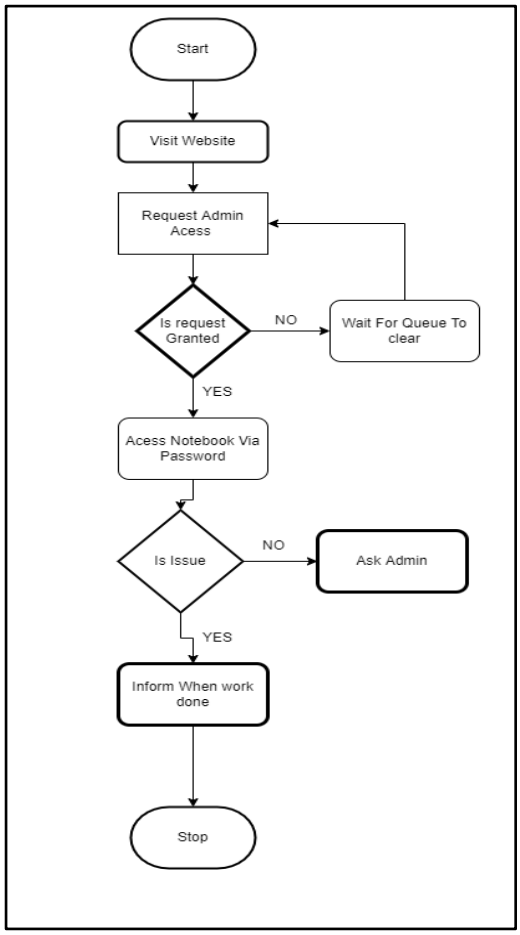

Fig -3.21: Flow Diagram

4. REQUIREMENT ANALYSIS

The cloud system is carried out on a computer system which has the different software and hardware specifications as given in Table 4.1 and Table 4.2 respectively.

4.1 Software

Table -4.1: Software details

Operating System	Linux (Ubuntu 20)
Programming Language	Shell Script/ Python
Other	Jupyter Notebook

Above mentioned requirement specifications are the minimum software requirements for efficient working of the system.

4.2 Hardware

Table -4.2: Hardware details

GPU'S	Nvidia 3000 series GPUs
HDD	1 TB
RAM	32 GB x 4

Above mentioned requirement specifications are the minimum hardware requirements for efficient working of the system.

5. APPLICATIONS

There are various applications of this cloud system. The applications are listed here.

5.1 Remote Integrated Development Environment (IDE)

The user can access the cloud from anywhere and thus edit the code on the go. The user need not worry about backing up the code until the resources are kept allotted to the user. Therefore the user can access the system through any devices and make the required changes to the code.

5.2 Machine Learning Applications on Cloud

The system can run various data sets related to any domain. The users log in to the cloud system to access the various resources available on the cloud as they do not have to physically access a particular system.

5.3 Artificial Intelligence Applications on Cloud

The users can run high computing software of Artificial Intelligence remotely from anywhere and anytime. The user simply needs to login to the system and avail the resources allotted. The user can request the admin to update resources as per their needs.

6. CONCLUSION

To conclude, the admin will be able to set up a cloud system using OpenStack. The admin can allocate various resources like RAM, Storage etc to the users of the instances. The user can connect to these instances using a token provided by the admin, thus allowing them to use high-end GPUs to run their Machine Learning and Artificial Intelligence applications. The user can access the system only using the using token provided by the admin, thus securing the system from unauthorized users.

REFERENCES

[1] P. Gopalakrishnan, B. U. Maheswari, Research On Enterprise Public and Private Cloud Service (2019)

[2] I. Gordin, A. Graur, A. Potorac, D. Balan, Security Assessment of OpenStack cloud using outside and inside software tools (2018)

[3] D. Grzonka, The Analysis of OpenStack Cloud Computing Platform: Features and Performance (2015)

[4] P. Ivanovic, H. Richter, Openstack Cloud Tuning for High Performance Computing (2018)

[5] C. Gaikwad, B. Churi, K. Patil, Tatwadarshi P. N., Providing Storage as a Service on Cloud using OpenStack(2017)

[6] S. Awasthi, A. Pathak, L. Kapoor, Openstack- Paradigm Shift to Open Source Cloud Computing & Its Integration (2016)

[7] J. P. Mulerikkal, Y. Sastri, A Comparative Study of OpenStack and CloudStack (2015)

[8] L. Wang, D. Zhang, OpenStack of open source cloud computing in colleges and universities' computer room (2017)

[9] C. Shao, B. Liang, F. Wang, H. Deng, W. Dai, S. Wei, X. Zhang, Z. Yuan, OpenStack Platform and its Application in Big Data Processing (2015)

[10] S. Patidar, D. Rane, P. Jain, A Survey Paper on Cloud Computing (2012)

[11] IBM 2014, accessed on 24 February 2022, <https://www.ibm.com/blogs/cloud-computing/2014/08/06/quick-overview-openstack-technology/>

[12] PubNub, accessed on 7 April 2022, <https://www.pubnub.com/learn/glossary/what-is-ngrok/>

[13] Informit 2012, accessed on 10 April 2022, <https://www.informit.com/articles/article.aspx?p=1927741&seqNum=2>