

DESIGN AND IMPLEMENTATION OF PRIVATE CLOUD FOR QUANT ALGO-TRADING USING OPENSTACK

**Sathyabama Institute of Science and Technology
(Deemed to be University)**

Submitted in partial fulfilment of the requirements for the award of
Master of Engineering Degree

in

Computer Science and Engineering

By

SAIPRASATH G

(Reg. No. 41310003)



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF COMPUTING**

**SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI – 600119,
TAMILNADU.**

APRIL 2023



SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)



Accredited with Grade "A" by NAAC

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI - 600 119

www.sathyabama.ac.in

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Phase-2 Report is the bonafide work of **SAIPRASATH G (Reg no:41310003)** who out the project entitled "**DESIGN AND IMPLEMENTATION OF PRIVATE CLOUD FOR QUANT ALGO-TRADING USING OPENSTACK**" under your supervision from December 2022 to April 2023.

Internal Guide

Dr. G. Mathivanan M.E., Ph.D.,

Head of the Department

Dr. L. Lakshmanan M.E., Ph.D.,



25.4.2023

Submitted for Viva voce Examination held on _____

Internal Examiner

External Examiner

DECLARATION

I, **SAIPRASATH G(41310003)** hereby declare that the Project Phase-2 Report entitled “**DESIGN AND IMPLEMENTATION OF PRIVATE CLOUD FOR QUANT ALGO-TRADING USING OPENSTACK**” done by me under the Guidance of **Dr. G. Mathivanan M.E. Ph.D.**, at Sathyabama Institute of Science and Technology is submitted in partial fulfilment of the requirements for the award of Master of Engineering degree in Computer Science and Engineering.

DATE:25.4.2023

PLACE: Chennai



SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management** of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. Sasikala M.E., Ph. D, Dean**, School of Computing, **Dr. L. Lakshmanan M.E., Ph.D.**, Head of the Department of Computer Science and Engineering for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr. G. Mathivanan M.E., Ph.D.**, for his valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project phase-2 work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project

ABSTRACT

This project presents the design and implementation of a private cloud for quant algo-trading using OpenStack. The private cloud is built using OpenStack, an open-source cloud computing platform, to provide a secure and scalable infrastructure for running quantitative trading algorithms. The architecture of the private cloud and the various services provided by OpenStack are described in detail. The implementation of the private cloud is discussed, including the challenges encountered during the process. Performance results are presented, demonstrating the effectiveness of the private cloud for running quant algo-trading algorithms, with high performance and low latency. Overall, this project highlights the benefits of using a private cloud built on OpenStack for quant algo-trading, providing a comprehensive understanding of the architecture, implementation, and performance of such a solution.

Key Words: OpenStack, quant algo-trading, Private cloud, Deployment, architecture, performance.

TABLE OF CONTENT

Chapter No:	Title	PAGE NO
	ABSTRACT	V
	LIST OF FIGURES	VII
1	INTRODUCTION	1
2	OBJECTIVE	4
3	LITERATURE SURVEY	5
	3.1 Literature Review	5
4	SYSTEM REQUIREMENTS	11
	4.1 Hardware Requirements	11
	4.2 Software Requirements	11
5	DESCRIPTIONS OF PROPOSED SYSTEM	12
	5.1 Existing System	12
	5.2 Proposed System	13
	5.3 Architecture	15
	5.4 Modules	15
	5.4.1 Openstack Deployment Module	16
	5.4.2 Openstack Instance Implement	21
	5.4.3 Implementing and Automating for Quant Algo-trading	22
	5.4.4 Security and Monitoring	23
6	RESULT AND DISCUSSION	25
7	CONCLUSION	26
8	REFERENCE	28

LIST OF FIGURES

Chapter No:	Title	Page No:
5.3	Architecture Diagram	15
5.4.1	Openstack Deployment	18
5.4.2	Openstack Instance	22
5.4.3	Algorithm for Trading	23
6.1	RNN Model	25
6.2	Values in Chart	25
6.3	Scheduling in Cloud	25

CHAPTER 1

INTRODUCTION

Cloud computing is a paradigm in which computing services, such as servers, storage, applications, and networking, are delivered over the internet on a pay-as-you-go basis. In other words, it allows users to access computing resources and services from a network of remote servers, rather than from a local server or a personal computer.

Cloud computing has become an integral part of modern business, providing companies with scalable and cost-effective solutions to their IT needs. It enables organizations to reduce the cost of IT infrastructure, increase efficiency, and improve the flexibility of their operations.

There are three primary models of cloud computing: Public, Private, and Hybrid. Public clouds are owned and operated by third-party providers, such as Amazon Web Services, Microsoft Azure, or Google Cloud Platform, and are accessible to anyone with an internet connection. Private clouds, on the other hand, are owned and operated by a single organization and are designed to provide secure and dedicated computing resources. Hybrid clouds combine both public and private clouds to offer more flexibility and customization.

The globalization of financial markets and rapid advances in computing technology have brought algorithmic trading to the forefront of financial trading. Algorithmic trading involves the use of automated systems to execute trades in financial markets at high speeds and frequencies. Quantitative analysis plays a critical role in algorithmic trading by providing sophisticated mathematical models and statistical methods to identify trading opportunities, analyze risk, and optimize trading strategies.

In recent years, the demands of algorithmic trading have outstripped the capabilities of traditional data centers. To cope with the increasing data volumes, processing demands, and scalability requirements of algorithmic trading, financial institutions are turning to cloud computing solutions. Cloud computing enables firms to create and manage vast pools of computing resources that can be dynamically allocated to meet the needs of different applications and users. The ability to rapidly scale up

and scale down resources as needed is a key advantage of cloud-based solutions for algorithmic trading.

One of the leading open-source cloud computing platforms for building private and public clouds is OpenStack. OpenStack is a modular platform that provides a set of software tools for building and managing cloud infrastructures. It is designed to be highly scalable, fault-tolerant, and vendor-agnostic. OpenStack provides a comprehensive suite of services for managing compute, storage, networking, and other resources in the cloud. It allows users to deploy and manage virtual machines (VMs), containers, and bare-metal servers, as well as to store and retrieve large amounts of data.

The design and implementation of a private cloud for algorithmic trading using OpenStack presents significant opportunities for financial institutions. By leveraging the power of OpenStack, firms can build a robust and scalable cloud infrastructure tailored to the needs of their algorithmic trading strategies. This infrastructure can handle vast amounts of data, support complex analytical models, and provide rapid response times for trades. It can also be designed to meet strict security and compliance regulations.

A key challenge in designing and implementing a private cloud for algorithmic trading is the need for high-performance computing (HPC) capabilities. Algorithmic trading involves computationally intensive tasks such as data mining, statistical analysis, and optimization. These tasks require high-performance computing resources that can handle large data volumes, process data in real-time, and execute complex algorithms. OpenStack provides tools for deploying HPC clusters and managing them efficiently.

Another challenge is the need for real-time data processing and analytics. Algorithmic trading relies on the availability of real-time market data to make timely trading decisions. To support real-time analytics, firms need to design their cloud infrastructure to minimize latency and maximize data throughput. OpenStack provides networking tools such as virtualized switches and routers that can be used to create high-speed, low-latency networks.

In conclusion, the design and implementation of a private cloud for algorithmic trading using OpenStack is a complex and challenging task, but the benefits are

substantial. This infrastructure provides a scalable, flexible, and secure platform for implementing advanced trading strategies based on sophisticated quantitative analysis. OpenStack provides the necessary tools and services for building and managing such a cloud infrastructure, including high-performance computing, real-time data processing, and low-latency networking. By leveraging OpenStack, firms can stay competitive in the rapidly evolving world of algorithmic trading.

CHAPTER 2

OBJECTIVE

The objective of the project "Design and Implementation of Private Cloud for Quant Algo-Trading using OpenStack" is to build a secure and scalable cloud computing environment that is optimized for quantitative algorithmic trading.

Quantitative algorithmic trading involves using advanced mathematical models and statistical analysis to identify profitable trading opportunities in financial markets. To be successful in this field, traders require a fast and reliable computing infrastructure that can process large volumes of data in real-time.

The project aims to achieve the following objectives:

- Design and deploy a private cloud infrastructure using OpenStack that is optimized for quantitative algorithmic trading.
- Develop and implement a set of tools and services that enable traders to access and process large volumes of market data in real-time.
- Implement security measures, such as encryption, access controls, and monitoring, to ensure that the cloud environment is secure and compliant with regulatory requirements.
- Optimize the cloud infrastructure for high-performance computing, leveraging technologies such as GPU acceleration, distributed processing, and parallel computing.
- Provide a user-friendly interface that enables traders to easily access and use the cloud environment for quantitative algorithmic trading.

By achieving these objectives, the project aims to provide traders with a powerful and flexible computing environment that enables them to process large volumes of market data in real-time, identify profitable trading opportunities, and execute trades quickly and efficiently.

CHAPTER 3

LITERATURE SURVEY

Literature survey is the most important step in software development process. Before developing the tool it is necessary to determine the time factor, economy and company strength. Once these things are satisfied, then the next step is to determine which operating system and language can be used for developing the tool. Once the programmers start building the tool the programmers need lot of external support. This support can be obtained from senior programmers, from book or from websites. Before building the system the above consideration are taken into account for developing the proposed system. The major part of the project development sector considers and fully survey all the required needs for developing the project. For every project Literature survey is the most important sector in software development process. Before developing the tools and the associated designing it is necessary to determine and survey the time factor, resource requirement, man power, economy, and company strength. Once these things are satisfied and fully surveyed, then the next step is to determine about the software specifications in the respective system such as what type of operating system the project would require, and what are all the necessary software are needed to proceed with the next step such as developing the tools, and the associated operations. The existing literature is discussed below about private cloud. Several cloud researchers have applied domain-level knowledge. Here, the crucial step is the feature selection, which is mostly handpicked.

3.1 LITERATURE REVIEW

[1] Title: Deploying An OpenStack Cloud Computing Framework for University Campus

Authors: P. Sai Sheela; Monika Choudhary.

Description: The project aims the usage of cloud computing technology has experienced a surge in popularity due to its extensive range of benefits, which include self-service provision, scalability, flexibility, and user guidance. The adoption of cloud computing technology by enterprises has been widely recognized, however,

educational institutions have not fully explored the technology's potential. Universities have a responsibility to provide students with ubiquitous and efficient services with limited resources; therefore, the time is opportune to explore the benefits that the cloud can provide. This project proposes deploying an OpenStack cloud computing framework specifically for university campuses. The framework can be customized according to the specific needs of the university, such as a self-service platform for virtualization, collaboration between educational organizations through community clouds, or as a test bed for students to develop applications. The framework offers diverse applications that can be tailored to the university's requirements. Furthermore, this project presents a range of use cases that can be achieved by deploying the cloud computing framework for universities. These use cases elucidate the diverse range of applications offered by the cloud and provide a glimpse of the various possibilities that the technology has to offer. In conclusion, deploying an OpenStack cloud computing framework for university campuses can significantly improve the services universities offer to their students. By providing the necessary infrastructure for virtualization, collaboration, and application development, the cloud can provide a more efficient, scalable, and flexible environment for universities to work upon.

[2] Title: Design and Implementation of Private Cloud for Higher Education using OpenStack.

Authors: Gaurav Bhatia; Ismail Al Noutaki; Sheikha Al Ruzeiqi; Jawaher Al Maskari.

Description: The project aims the implementation of private cloud for higher education using OpenStack is a crucial project that aims to improve the access to dedicated devices and various operating systems required in teaching and learning at higher education institutions. The cost of hardware is a significant challenge in providing physical devices that are compatible with the ever-changing IT field. Additionally, different courses may demand specific software running on specific operating systems with a particular hardware configuration, making it challenging to manage and distribute them. OpenStack is an Infrastructure as a Service (IaaS) that provides a standard mechanism to manage and distribute course-specific Virtual Machines (VMs) while addressing the challenge of hardware resources. This project will provide a design of a private cloud for higher education and a proof-of-concept

implementation methodology for the OpenStack platform. To facilitate the construction of a private cloud, the resource requirement model will be formulated to measure the physical resource requirements for a specific number of virtual machines. The implementation of a private cloud for higher education will improve accessibility, reduce cost, and simplify the complex process of installing and maintaining different operating systems. This project will enhance the teaching and learning experience for students, reduce the burden on IT departments, and provide a more flexible and innovative approach to delivering educational content.

[3] Title: Design and implementation of private cloud storage platform based on OpenStack.

Authors: Jiuyuan Huo; Hong Qu; Ling Wu.

Description: The project aims the exponential growth of enterprise data has led to storage-related issues such as the inability of hardware storage devices to keep up, high maintenance costs, and incomplete storage environments. Cloud computing technology offers businesses with an effective solution that can cater to their storage needs. OpenStack Cloud Computing platform provides Swift framework that offers an object storage solution that can create redundant copies of data. Swift framework has a fully symmetrical system architecture, high data persistence, unlimited scalability, no single point of failure, and is simple and reliable. Therefore, a private cloud storage platform that uses Swift framework has been designed and implemented in this project. The website system was constructed using the Python framework Django to enable the creation, upload, and download of enterprise data files using a web-based method. This implementation proves that a private cloud storage solution based on Swift is feasible and effective for businesses dealing with large volumes of data. The proposed private cloud storage platform is expected to address issues related to storage and provide a seamless storage environment that enhances overall business operations.

[4] Title: Automation of the Infrastructure and Services for an OpenStack Deployment Using Chef Tool.

Authors: Eduard Luchian; Cosmin Filip; Andrei Bogdan Rus; Iustin-Alexandru Ivanciu; Virgil Dobrota.

Description: The project aims to automate the deployment of OpenStack and the underlying architecture in one go, using the Chef tool. This would improve the existing legacy automatic deployments which are focused on specific packets and configurations. The proposed approach would streamline the installation process, reduce manual intervention and minimize the chances of errors or issues during the deployment process. The deployment process would involve at least one Controller Node and one Compute Node. To automate the creation and configuration of virtual environments, the project would leverage Vagrant and VirtualBox as hypervisor tools. Compared to manual installation, the proposed approach would significantly reduce the installation time - from over 30 minutes per node to 10 minutes per node. For experienced users, the time per node could be reduced further to just 2 minutes, provided the Vagrant box has been executed before. However, handling prompts from installed packages proved to be a drawback, which required the development of a workaround. Despite this limitation, the proposed approach would help to enhance the efficiency and effectiveness of OpenStack deployments, making it easier and simpler to implement OpenStack on the infrastructure.

[5] Title: QuantCloud: Big Data Infrastructure for Quantitative Finance on the Cloud.

Authors: Peng Zhang; Kai Yu; Jessica J. Yu; Samee U. Kha.

Description: QuantCloud aims to provide a big data infrastructure that can perform complex analytics in quantitative finance by analysing market observations to search for profitable signals in anticipation of volatile trading patterns. The infrastructure is designed to address challenges related to handling high volume, velocity, and variety investment instruments. The project employs large-scale SSD-backed datastore, parallel processing algorithms, and portability in Cloud computing to bridge the gap between model computing techniques and financial data-driven research. QuantCloud provides Data-on-Demand services by structuring the large volume of market data in an SSD-backed datastore, and a daemon reacts to provide the services. Multiple client services process user requests in a parallel mode and query on-demand datasets from the datastore through Internet connections. The

project's performance benchmarking indicates data-access application latency as low as 3.6 nanoseconds per message, sustained throughput for parallel data processing as high as 74 million messages per second, and completion of 11 petabyte-level data analytics within 53 minutes. The infrastructure, parallel algorithms, and sophisticated implementations offer the algorithmic trading and financial engineering community new hope and numeric insights for their research and development.

[6] Title: Virtual Memory Management of Private Cloud using OpenStack API

Authors: Pranjali P. Deshmukh; S.Y.Amdani.

Description: The project aims to optimize the cloud resources for companies through the use of OpenStack API. In recent times, the demand for computing resources has continued to rise, and traditional IT systems are proving to be ineffective in responding to this high utilization of resources. IT systems require high investment costs, maintenance and updates, which drain resources from organizations. On the other hand, cloud computing offers a cost-effective solution that can support the huge resources required by businesses seamlessly. The goal of the project is to enhance the performance of cloud infrastructures and platforms by optimizing the memory management of OpenStack. The project will focus on the distribution of virtual machine instances in the cloud environment. OpenStack is a free open-source software that offers a robust platform for cloud computing applications. By leveraging the capabilities of OpenStack API, the project aims to improve the efficiency of private clouds by optimizing virtual memory management. Through this approach, businesses will be able to align their resources in a cost-effective way, reduce the complexity of IT infrastructure, and improve their overall efficiency. Consequently, the project is a timely initiative that will provide greater opportunities and benefits for organizations that are using private clouds.

[7] Title: Use Python Api To Automate Script Based On Open Stack Platform

Authors: Di Liu!, Pin Wang.

Description: The best way to learn API is to use them in an interactive command-line Python interpreter. Python interpreter is an enhanced Python interpreter displays a valid method name you type, and will automatically display a string in document. Open Stack is a newcomer IaaS space, first released in late 2010. Although this solution is not enough maturity, and market less than two years' time, but Open Stack has become the most widely used cloud stack one.

CHAPTER 4

SYSTEM REQUIREMENTS

4.1 HARDWARE REQUIREMENTS

Hard Disk – 1TB

Input Devices: Keyboard, Mouse

Processor – 8 Cores

Ram: 64 GB

4.2 SOFTWARE REQUIREMENTS

VMware, VirtualBox

Operating system – windows, Ubuntu

Coding Language : Python

CHAPTER 5

DESCRIPTIONS OF PROPOSED SYSTEMS

5.1 EXISTING SYSTEM

The existing system for quantitative algorithmic trading in the cloud involves using public cloud providers to host and manage computing resources. This approach allows traders to leverage the computing power of the cloud to process large volumes of data in real-time and execute trades quickly and efficiently.

Traders typically use a combination of proprietary and open-source software tools to analyze market data and develop trading algorithms. These tools can be hosted on virtual machines or containers within the cloud environment. Public cloud providers offer a variety of pricing models for their computing resources, including pay-as-you-go, reserved instances, and spot instances. Traders can choose the pricing model that best suits their needs and adjust their computing resources accordingly.

While the use of public cloud providers can be effective, it has some limitations. One of the major concerns is security. Public cloud providers may not be suitable for highly sensitive financial data due to concerns about data privacy and security. In addition, the cost of using public cloud providers can be significant, especially when processing large volumes of data. Finally, public cloud providers may not offer the level of customization and control over computing resources that traders require.

To address these limitations, private cloud infrastructure can be used instead. Private cloud infrastructure can be customized to meet the specific needs of each organization, providing greater flexibility and control over IT resources. In addition, private cloud infrastructure can be more secure since it is not shared with other organizations. Private cloud infrastructure can be built using open-source software like OpenStack, which provides a scalable and flexible cloud computing environment. By leveraging the power of private cloud infrastructure, traders can benefit from a secure, scalable, and cost-effective computing environment that is optimized for quantitative algorithmic trading.

DISADVANTAGES

- Security: Public cloud providers may not be suitable for highly sensitive financial data due to concerns about data privacy and security.
- Cost: Public cloud providers charge for the use of their computing resources, which can be costly, especially when processing large volumes of data.
- Control: Public cloud providers may not offer the level of customization and control over computing resources that traders require.
- Latency: In a public cloud environment, traders may experience latency when accessing computing resources, which can negatively impact trading performance.
- Reliability: Public cloud providers may experience outages or service disruptions, which can affect trading operations.
- Compliance: Traders may be subject to regulatory requirements that limit the use of public cloud infrastructure for trading operations.

5.2 PROPOSED SYSTEM

The proposed system for Design and Implementation of Private Cloud for Quant Algo-Trading using OpenStack aims to address the limitations of the existing system for quantitative algorithmic trading in the cloud. The proposed system will use a private cloud infrastructure built using OpenStack to provide a secure, scalable, and cost-effective computing environment that is optimized for quantitative algorithmic trading.

The system will consist of several components, including:

- Private cloud infrastructure: The private cloud infrastructure will be built using OpenStack, which is an open-source software platform for building private and public clouds. OpenStack provides a scalable and flexible cloud computing environment that can be customized to meet the specific needs of each organization. The private cloud infrastructure will provide greater control and customization over IT resources, as well as enhanced security for sensitive financial data.

- Virtual machines and containers: Traders will use virtual machines or containers within the private cloud infrastructure to host and run their proprietary and open-source software tools for analyzing market data and developing trading algorithms. The use of virtual machines or containers will allow traders to manage their computing resources more efficiently, while also reducing costs.
- Network infrastructure: The network infrastructure will be designed to provide secure and reliable connectivity between the private cloud infrastructure and external networks, such as the internet or other trading networks. The network infrastructure will be optimized for low latency and high throughput to support real-time trading operations.
- Monitoring and management tools: The system will include monitoring and management tools to ensure the reliability and performance of the private cloud infrastructure. These tools will allow traders to monitor their computing resources and adjust them as needed to optimize trading performance.

The proposed system will offer several benefits over the existing system for quantitative algorithmic trading in the cloud. These benefits include:

- Security: The private cloud infrastructure will provide enhanced security for sensitive financial data, as it will not be shared with other organizations.
- Control: Traders will have greater control over their computing resources, allowing them to customize their IT infrastructure to meet their specific needs.
- Cost-effectiveness: The use of a private cloud infrastructure can be more cost-effective than using public cloud providers, as traders can avoid paying for the use of computing resources they do not need.
- Scalability: The private cloud infrastructure will be designed to be highly scalable, allowing traders to expand their computing resources as needed to support their growing trading operations.

Overall, the proposed system for Design and Implementation of Private Cloud for Quant Algo-Trading using OpenStack offers a secure, scalable, and cost-effective solution for quantitative algorithmic trading in the cloud. By leveraging the power of private cloud infrastructure and OpenStack, traders can benefit from a highly

customizable and flexible computing environment that is optimized for their unique trading needs.

5.3 ARCHITECTURE DIAGRAM

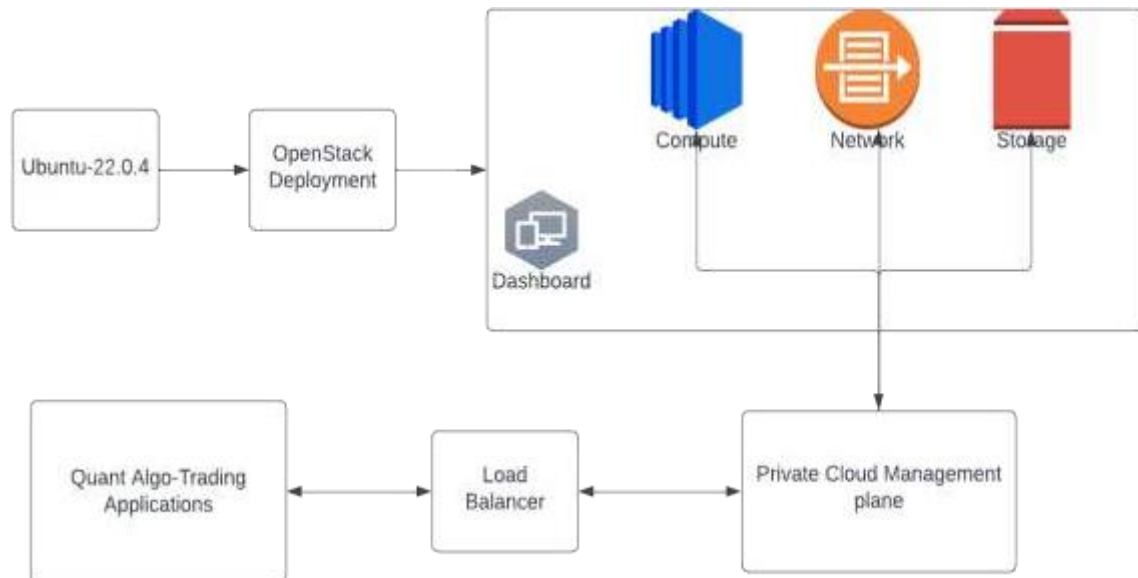


FIG 5.3 Architecture Diagram

5.4 MODULES

The project focuses on building a private cloud infrastructure using the OpenStack platform to enable efficient and secure quant algo-trading operations. The project involves several key modules, including OpenStack deployment, OpenStack instance deployment, and implementing and automating the quant algo-trading process.

The OpenStack deployment module involves setting up and configuring the OpenStack platform, including the installation and configuration of various components such as Keystone for identity management, Nova for compute resources, Glance for image management, and Neutron for networking. This module ensures that the private cloud infrastructure is properly deployed and ready for use.

The OpenStack instance deployment module focuses on creating and managing virtual instances on the private cloud. This includes provisioning virtual machines (VMs) with appropriate compute resources, storage, and networking, and configuring them to meet the requirements of the quant algo-trading process. This module ensures that the VMs are deployed and configured correctly to support the algorithmic trading strategies.

The implementing and automating the quant algo-trading module involves developing and implementing the quant algo-trading strategies on the private cloud infrastructure. This includes coding, testing, and optimizing the algorithmic trading strategies using programming languages such as Python, R, or MATLAB. The strategies are then deployed on the private cloud instances, and the trading process is automated using tools such as cron jobs or other scheduling mechanisms. This module ensures that the quant algo-trading process is automated and efficiently executed on the private cloud infrastructure.

Overall, the project aims to design and implement a private cloud infrastructure using OpenStack for quant algo-trading, enabling efficient, secure, and automated trading operations. It involves setting up and configuring OpenStack, deploying virtual instances, and implementing and automating the quant algo-trading process, providing a robust and scalable solution for quantitative trading in a private cloud environment.

5.4.1 OPENSTACK DEPLOYMENT MODULE

OpenStack is an open-source software platform that provides a scalable and flexible cloud computing environment. OpenStack deployment involves installing and configuring OpenStack components on one or more physical or virtual servers to create a private or public cloud infrastructure.

The OpenStack deployment process involves several steps, including:

- **Hardware requirements:** The first step in OpenStack deployment is to ensure that the hardware meets the minimum requirements for running OpenStack components. This includes CPU, RAM, and storage capacity.

- Operating system installation: OpenStack can be installed on a variety of operating systems, including Ubuntu, CentOS, and Red Hat Enterprise Linux. The selected operating system must be installed on the physical or virtual servers that will be used to deploy OpenStack.
- Network configuration: OpenStack requires a network infrastructure to connect the different components of the cloud infrastructure. The network must be configured to provide connectivity between the physical or virtual servers that will host OpenStack components.
- OpenStack component installation: OpenStack consists of several components, including Nova (compute), Glance (image), Keystone (identity), Neutron (networking), and Cinder (block storage), among others. Each component must be installed and configured to work together seamlessly.
- Configuration management: OpenStack deployment involves a significant amount of configuration management. Configuration files must be edited to customize the behavior of OpenStack components, such as setting network configurations or defining storage policies.
- Service management: OpenStack services must be started, stopped, and managed to ensure they are running properly. Service management tools such as systemd or init can be used to control the behavior of OpenStack services.
- Integration with other tools: OpenStack can be integrated with other tools, such as monitoring and management tools, to enhance the functionality of the cloud infrastructure.

Once the OpenStack deployment process is complete, the cloud infrastructure can be used to create virtual machines, containers, and other computing resources. These resources can be managed through the OpenStack dashboard or command-line interface, allowing users to customize and manage their computing environment to meet their specific needs.

OpenStack deployment can be complex and time-consuming, requiring expertise in system administration and cloud computing. However, once deployed, OpenStack provides a highly customizable and scalable cloud computing environment that can be tailored to meet the unique needs of each organization.

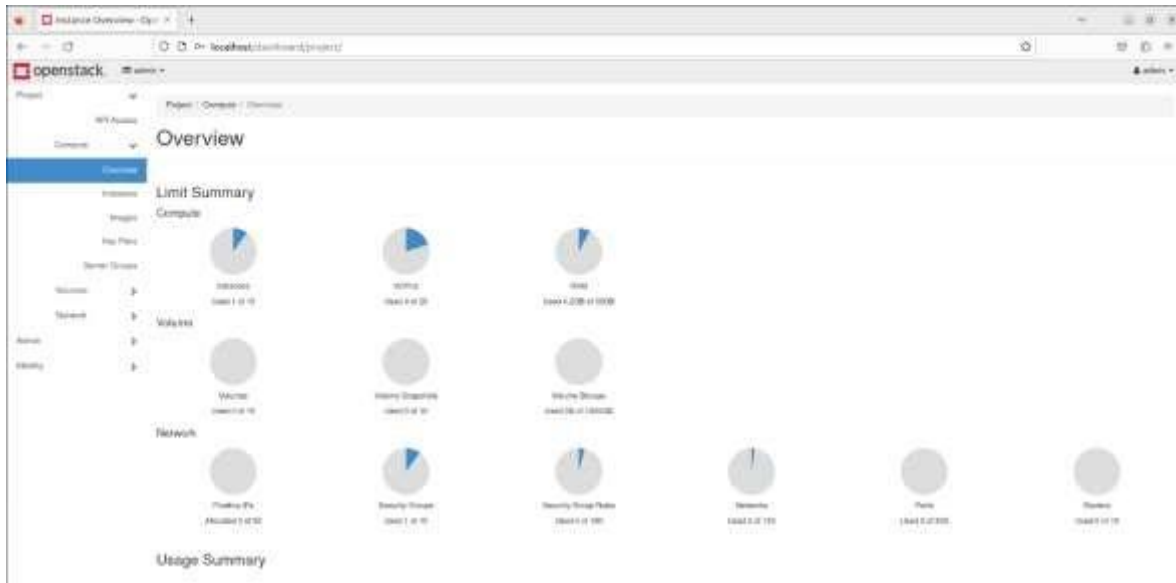


FIG 5.4.1 Openstack Deployment

The OpenStack Dashboard, also known as Horizon, is a web-based user interface that provides a graphical interface for users to interact with and manage their OpenStack cloud resources. The dashboard provides an easy-to-use interface for managing virtual machines, networks, storage, and other OpenStack services.

Some of the options that are typically provided in the OpenStack Dashboard include:

- **Project Overview:** This section provides a high-level view of the user's projects, including the number of instances, volumes, and other resources in use.
- **Instances:** This section provides a list of the virtual machines running in the user's project. From here, users can launch new instances, manage existing instances, and view performance metrics.
- **Volumes:** This section provides a list of block storage volumes that are attached to instances in the user's project. From here, users can manage existing volumes, create new volumes, and view usage information.
- **Images:** This section provides a list of virtual machine images that are available for users to launch instances from. From here, users can create new images, edit existing images, and delete images.

- **Networks:** This section provides a list of virtual networks that are available to the user. From here, users can manage network settings, create new networks, and view usage information.
- **Security Groups:** This section provides a list of security groups that are associated with the user's project. From here, users can manage security group rules, create new security groups, and view usage information.
- **Object Store:** This section provides a web interface to the OpenStack Object Storage service, allowing users to create, manage, and delete containers and objects.
- **Orchestration:** This section provides a web interface to the OpenStack Orchestration service, allowing users to create and manage stacks of resources that are automatically deployed and managed.

These are some of the options that are typically provided in the OpenStack Dashboard. The actual options may vary depending on the OpenStack services that are installed and configured in the environment.

In OpenStack, there are two types of users: administrators and project users. The main difference between the two is their level of access and the scope of resources they can manage.

An administration user in OpenStack is a user account with elevated privileges that is used to manage and configure the OpenStack environment. An admin user has full access to all OpenStack services and resources, including the ability to create, modify, and delete projects, users, and access controls. They can also view and modify the quotas for projects and manage the OpenStack infrastructure. An admin user can perform various administrative tasks, such as managing images, creating and managing flavors, configuring networks, managing storage, and setting up security groups. They can also monitor system performance and troubleshoot issues that arise in the OpenStack environment. The admin user has access to the OpenStack dashboard, which provides a web-based graphical interface for managing OpenStack resources. They can use the dashboard to view and manage projects, users, and other resources in the environment. In addition to the dashboard, the admin user can use the OpenStack command-line tools to perform administrative tasks. These tools provide a powerful way to manage the OpenStack

environment through automation and scripting. Overall, the administration user in OpenStack has the highest level of privileges and can perform any action on the OpenStack services. They are responsible for managing the OpenStack environment and ensuring that it is secure, reliable, and efficient.

A project user in OpenStack is a user account that has access to a specific project within the OpenStack environment. Project users are typically members of a team or organization that needs to use the OpenStack cloud resources for their work. Project users have limited access to the OpenStack environment and can only manage resources within their project. They can launch and manage virtual machines, create and manage storage volumes, configure networks, and set up security groups within their project. However, they cannot access resources or data from other projects, and they do not have administrative privileges to manage the OpenStack infrastructure. Project users can interact with the OpenStack environment through the OpenStack dashboard, which provides a web-based graphical interface for managing OpenStack resources. They can also use the OpenStack command-line tools to interact with the environment programmatically and automate their workflows. The project user's access to resources is governed by quotas, which are set by the OpenStack administrator. Quotas limit the amount of resources that a project user can consume, such as the number of virtual machines, storage volumes, and network ports. This helps to ensure that resources are fairly allocated to all projects and prevents any one project from consuming all available resources. Overall, project users in OpenStack have access to the resources they need to perform their work within their project, while their access is limited to ensure that the OpenStack environment remains secure and efficient.

In summary, while administrators have full access to all OpenStack services and resources, project users are limited to managing resources within their specific project. This helps to ensure that resources are securely and efficiently allocated to projects and that project users can only access and manage resources that are relevant to their work.

5.4.2 OPENSTACK INSTANCE IMPLEMENTATION

OpenStack instance implementation involves creating and managing virtual machines (VMs) within the OpenStack cloud infrastructure. An instance is a virtual machine that can be used to run applications, services, or workloads within the cloud environment.

The process of implementing an OpenStack instance involves the following steps:

- Image selection: The first step in instance implementation is to select an image that will be used to create the virtual machine. An image is a pre-configured template that contains an operating system, applications, and other software components.
- Flavor selection: A flavor is a predefined set of hardware resources that can be allocated to the virtual machine, including CPU, RAM, and storage. The flavor selection determines the performance and capacity of the virtual machine.
- Instance creation: Once the image and flavor have been selected, the instance can be created using the OpenStack dashboard or command-line interface. During instance creation, the user can specify additional configuration settings, such as network connectivity and security groups.
- Instance management: Once the instance is created, it can be managed through the OpenStack dashboard or command-line interface. Users can start, stop, and resize instances as needed, as well as monitor their performance and utilization.
- Data management: Instance implementation also involves managing the data associated with the virtual machine. This includes creating and attaching storage volumes, backing up and restoring data, and managing snapshots.

OpenStack instance implementation provides a flexible and scalable cloud computing environment that can be tailored to meet the specific needs of each organization. By creating and managing instances within the OpenStack cloud infrastructure, users can quickly and easily deploy applications, services, and workloads, while also ensuring the security and scalability of their computing environment.

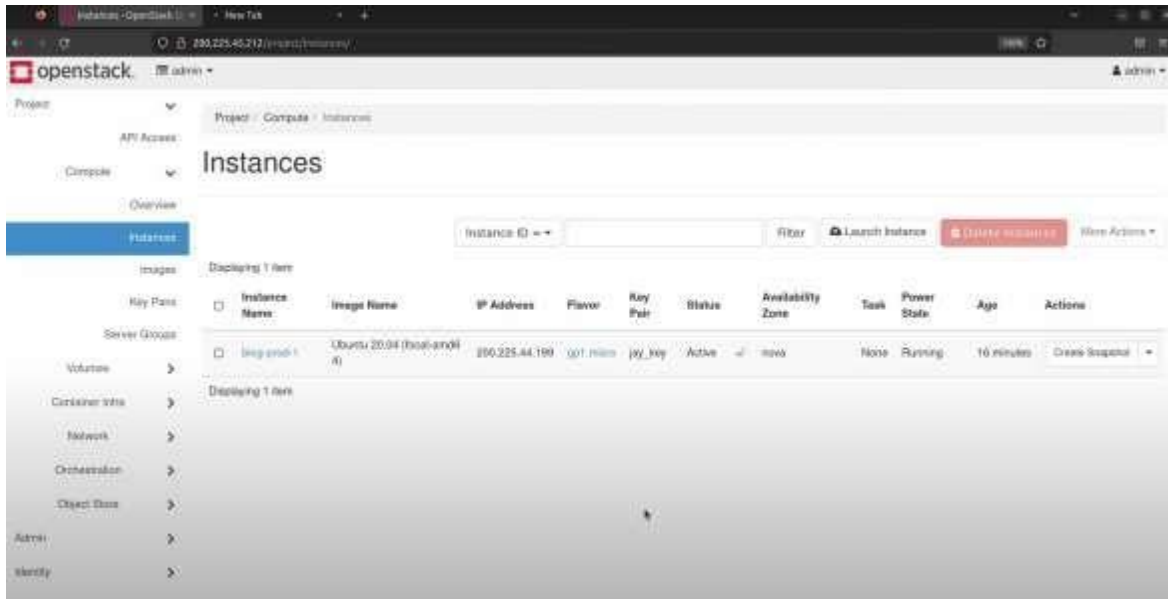


FIG 5.4.2 Openstack Instance

5.4.3 IMPLEMENTING AND AUTOMATING THE QUANT ALGO-TRADING

Implementing and automating quantitative algorithmic trading using OpenStack involves several steps, including:

- **Algorithm deployment:** The next step is to deploy the trading algorithms on the OpenStack instances. This can be done by creating virtual environments on the instances and installing the necessary software packages and dependencies.
- **Data processing and analysis:** After the trading algorithms are deployed, the next step is to process and analyze the market data. This involves accessing and aggregating real-time data from financial markets, performing statistical analysis, and generating trading signals.
- **Trading execution:** Once the trading signals are generated, the next step is to execute trades automatically using the OpenStack instances. This can be done by integrating the trading algorithms with the trading platform or exchange APIs.
- **Monitoring and optimization:** The final step is to monitor the performance of the trading system and optimize it for better results. This involves monitoring the performance of the OpenStack instances, analyzing the trading results, and adjusting the trading algorithms as needed.

Automation can be achieved using OpenStack APIs and various automation tools like Ansible, Puppet, or Chef. These tools can be used to automate the deployment and configuration of the OpenStack cloud infrastructure, as well as the deployment and management of trading algorithms and instances.

By implementing and automating quant algo trading using OpenStack, traders can benefit from a scalable and flexible cloud computing environment that can be customized to meet their specific needs. OpenStack's advanced networking and storage features, as well as its support for multiple hypervisors, make it an ideal platform for running trading algorithms and executing trades in real-time.

```
Recurrent Neural Network

from tensorflow.keras.callbacks import EarlyStopping

# additional library
import tensorflow
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, LSTM, Dropout

nb_hidden_layer = 3

# model initialization
classifier = Sequential()

# adding LSTM layer
classifier.add(LSTM(units = 10, return_sequences = True,
                  input_shape = (X_train_3d.shape[1], X_train_3d.shape[2])))

# loop to add LSTM layer
for _ in range(1):
    classifier.add(LSTM(units = 10, return_sequences = True))
    classifier.add(Dropout(0.15))

# last lstm layer
classifier.add(LSTM(units = 10, return_sequences = False))

# Dense Output layer
classifier.add(Dense(1, activation="sigmoid"))

# compiling the Model
classifier.compile(loss="Binary_crossentropy", optimizer="adam")

# Training with early Stop
early_stop = EarlyStopping(verbose=1, patience = 5)

classifier.fit(X_train_3d, y_train_3d, validation_data=(X_test_3d, y_test_3d), epochs=100, callbacks=[early_stop])
```

FIG 5.4.3 Algorithm for trading

5.4.4 SECURITY AND MONITORING

The security and monitoring module in the project design and implementation of private cloud for quant algo-trading using openstack is responsible for ensuring that the system is secure and that its performance and availability are monitored. This module involves setting up security measures such as firewalls, encryption, and

access controls within the OpenStack environment, and monitoring the performance and availability of the system.

The security measures that may be implemented include configuring security groups and access controls to restrict access to the OpenStack APIs and VMs, setting up encryption for data at rest and in transit, and implementing network segmentation to prevent unauthorized access to sensitive data. In addition, regular security assessments and vulnerability scans should be performed to identify and mitigate potential security risks.

The monitoring aspect of this module involves setting up tools and processes to monitor the performance and availability of the OpenStack environment and the algo-trading platform. This may include configuring logging and alerting systems to notify administrators of any issues or errors, monitoring system metrics such as CPU usage, memory usage, and network traffic, and setting up performance benchmarks to ensure that the system is performing optimally.

The security and monitoring module is crucial for ensuring that the private cloud and algo-trading platform are secure, reliable, and performant. Without adequate security measures, the system may be vulnerable to attacks or data breaches, which can lead to significant financial losses and reputational damage. Similarly, without monitoring tools and processes, administrators may not be aware of issues or errors until they result in downtime or other disruptions to the system. Therefore, careful planning and implementation of the security and monitoring module are essential for the success of the project.

RESULT AND DISCUSSION

```
from tensorflow.keras.callbacks import EarlyStopping

# additional library
import tensorflow
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, LSTM, Dropout

# hidden layer = 1

# model initialization
classifier = Sequential()

# adding LSTM layer
classifier.add(LSTM(units = 50, return_sequences = True,
                  input_shape = (X_train.shape[1],X_train.shape[1],)))

# join to add LSTM layer
for _ in range(1):
    classifier.add(LSTM(units = 50, return_sequences = True))
    classifier.add(Dropout(0.1))

# last-lstm layer
classifier.add(LSTM(units = 50, return_sequences = False))

# Dense output layer
classifier.add(Dense(units = 1, activation='sigmoid'))

# compiling the Model
classifier.compile(loss='binary_crossentropy', optimizer='adam')

# fitting with early stop
early_stop = EarlyStopping(monitor='val_loss', min_delta = 1)
```

FIG 6.1 RNN Model

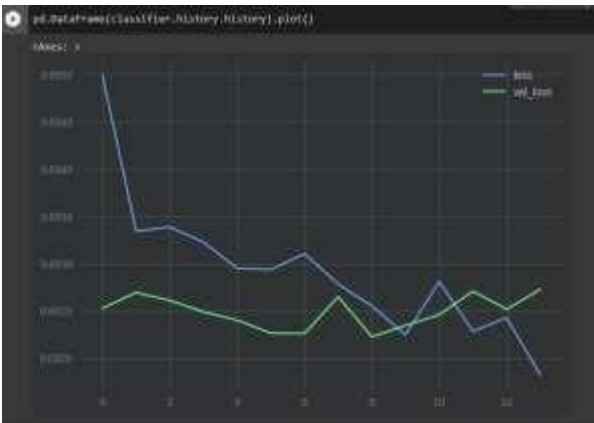


FIG 6.2 Values In Chart

[illegible]

FIG 6.3 Scheduling In Cloud

CONCLUSION

In this project, the design and implementation of a private cloud for quant algo-trading using OpenStack can bring significant benefits to the world of financial services. The use of cloud computing allows for efficient utilization of resources, scalability, and flexibility in deploying and managing quant algo-trading applications. OpenStack, as a powerful and open-source cloud computing platform, provides the necessary infrastructure and services to build a private cloud tailored for quant algo-trading.

The project involves careful planning and design to meet the specific requirements of quant algo-trading, including high-performance computing, low-latency networking, and data privacy and security. By leveraging the features of OpenStack, such as Compute, Networking, and Storage services, the private cloud can be customized to provide the optimal environment for running quant algo-trading algorithms.

The implementation of the private cloud for quant algo-trading using OpenStack involves setting up and configuring various OpenStack components, such as Nova (Compute), Neutron (Networking), Cinder (Block Storage), and Glance (Image Service), to create a robust and scalable infrastructure. The deployment of virtual instances with appropriate resources, such as CPU cores and RAM, allows for efficient allocation of computing resources to run quant algo-trading algorithms with high performance and low-latency.

Furthermore, the private cloud provides the necessary security measures to protect sensitive financial data, including network segmentation, authentication, authorization, and encryption. Compliance with industry regulations and best practices for data privacy and security is crucial in the design and implementation of the private cloud for quant algo-trading.

Overall, the design and implementation of a private cloud for quant algo-trading using OpenStack offer numerous benefits, including improved scalability, flexibility, performance, and security for running quant algo-trading applications. By leveraging the power of cloud computing and the features of OpenStack, financial institutions and quant traders can efficiently manage their quant algo-trading workloads, gain a competitive edge, and adapt to changing market conditions. Proper planning,

design, and implementation following best practices and industry standards are essential for a successful private cloud deployment for quant algo-trading using OpenStack.

REFERENCES

- [1] D. G. Chandra and M. D. Borah, "Cost benefit analysis of cloud computing in education", in International Conference on Computing, Communication and Applications, Dindigul, Tamilnadu, 2012, pp. 1-6. doi: 10.1109/ICCCA.2012.6179142
- [2] A. Tashkandi and I. Al-Jabri, "Cloud Computing Adoption by Higher Education Institutions in Saudi Arabia: Analysis Based on TOE", in: International Conference on Cloud Computing (ICCC), Riyadh, 2015, pp. 1-8.
- [3] A. C. Caminero, S. Ros, R. Hernández, A. Robles-Gómez and R. Pastor, "Cloud-based e-learning infrastructures with load forecasting mechanism based on Exponential Smoothing: A use case", in: Frontiers in Education Conference (FIE), Rapid City, SD, 2011, pp. S3C-1-S3C 6.
- [4] N. Alkhater, G. Wills and R. Walters, "Factors Influencing an Organisation's Intention to Adopt Cloud Computing in Saudi Arabia", in: IEEE 6th International Conference on Cloud Computing Technology and Science, Singapore, 2014, pp. 1040-1044.
- [5] How is Cloud Computing Different from Traditional IT Infrastructure, Web page at <https://www.leadingedgetech.co.uk/it-services/it-consultancy-services/cloud-computing/how-is-cloud-computing-different-from-traditional-it-infrastructure/> . Date of last access: 03rdMarch 2017.
- [6] Cisco (2010), "Cloud Computing in Higher Education: A Guide to Evaluation and Adoption", [Online] Available:
http://www.cisco.com/web/offer/email/43468/5/Cloud_Computing_in_Higher_Education.pdf
- [7] Cisco, "Simplify desktop management, Enhance Security and Lowercosts with Cisco VirtualDesktopSolutions", [Online] Available: https://www.cisco.com/c/dam/en/us/solutions/collateral/data-center-virtualization/dc-partner-vmware/cisco_education.pdf.

- [8] 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 Science and Information Conference, London, 2013, pp. 888-893.
- [9] A. Paventhan, H. Krishna, V. Gayathri and N. Pahuja, "Towards educational school cloud for India: A case study with Kendriya Vidyalayas", in IEEE Region 10 Humanitarian Technology Conference (R10 HTC), Chennai, 2014, pp. 46-51.
- [10] A. C. Caminero et al., "Deconstructing remote laboratories to create Laboratories as a Service (LaaS)", in: 2014 IEEE Global Engineering Education Conference (EDUCON), Istanbul, 2014, pp. 623-629.
- [11] Yusuf Mehdi, (January 24th 2017), Announcing Intune for Education & new Windows 10 PCs for school starting at \$189[WebPage].RetreivedfromURL:<https://blogs.windows.com/windowsexperience/2017/01/24/announcingintune-education-new-windows-10-pcs-school-starting-189/>
- [12] N. Ferry, A. Rossini, F. Chauvel, B. Morin, and A. Solberg, "Towards Model-Driven Provisioning, Deployment, Monitoring, and Adaptation of Multi-cloud Systems", Sixth IEEE International Conference on Cloud Computing 2013, June 28 - July 3 2013, pp. 887-894.
- [13] G. Katsaros, M. Menzel, A. Lenk, J.R. Revelant, R. Skipp, and J. Eberhardt, "Cloud Application Portability with TOSCA, Chef and Openstack", IEEE International Conference on Cloud Engineering IC2E 2014, 11-14 March 2014, pp. 295-302