

INTERDISCIPLINARY PROJECT REPORT **at**

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

Submitted in partial fulfilment of the requirements for the award of Bachelor of
Engineering Degree in Information Technology

By
N Uday Kiran
(Reg.No.40120061)



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF COMPUTING
SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY
JEPPIAAR NAGAR, RAJIV GANDHI SALAI,
CHENNAI – 600119, TAMILNADU

APRIL 2023





SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DESIGNED TO BE UNIVERSITY)

Accredited 'A' Grade by NAAC / ISO Status by UGC / Approved by AICTE

www.sathyabama.ac.in

SCHOOL OF COMPUTING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the Bonafide work of **N UDAY KIRAN (40120061)** who carried out the project entitled **UBUNTU OPERATING SYSTEM** under my supervision from January 2023 to April 2023. **ubuntu openstack image**


Internal Guide

DR. S. GOWRI M.E., Ph.D.


Head of the Department
Dr. R. SUBHASHINI M.E., Ph.D.

Submitted for Viva voce Examination held on 28/4/2023


Internal Examiner


External Examiner

DECLARATION

I, N Uday Kiran hereby declare that the Project Report entitled **Create web enabled**

Ubuntu openstack image Done by me under the guidance of **Dr.S.Gowri**

M.E.,Ph.D. is submitted in partial fulfilment of the requirements for the award of Bachelor of Engineering degree in Information Technology.

DATE:

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. R. Subhashini, M.E., Ph.D.**, Head of the Department of Information Technology 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.S.Gowri,M.E.,Ph.D.** for his valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

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

TRAINING CERTIFICATE

ABSTRACT

Creating a web-enabled Ubuntu OpenStack image involves configuring an instance to serve as a template for other instances to be launched later. This image is pre-installed with a web server, which allows it to host websites. In this abstract, we will discuss the steps involved in creating a web-enabled Ubuntu OpenStack image.

First, we need to launch an instance using an Ubuntu server image in OpenStack. Once the instance is launched, we can install a web server such as Apache or Nginx. We can then configure the web server to serve content from a specific directory, or create a virtual host to serve multiple websites.

After the web server is installed and configured, we need to create an image of the instance. This can be done using the OpenStack dashboard or the command-line interface. Once the image is created, we can use it to launch new instances that are pre-configured with the web server.

To make the image web-enabled, we should also consider installing additional software such as a database server or a content management system (CMS). This will enable the image to serve dynamic content such as blogs or e-commerce sites.

Overall, creating a web-enabled Ubuntu OpenStack image involves launching an instance, installing and configuring a web server, creating an image of the instance, and installing additional software as needed. This process allows us to quickly launch instances with pre-configured web servers, which can save time and effort when deploying websites in OpenStack.

development. The step-by-step instructions and detailed explanations make it an ideal resource for beginners and anyone looking to expand their knowledge and skills in cloud computing. Sure, I can provide an abstract for creating an OpenStack Network using Linuxbridge:

OpenStack Network is a virtual networking service that allows users to create and manage networks in a cloud environment. Linuxbridge is one of the available plugins for OpenStack Network that provides network connectivity between instances on the same network.

To create an OpenStack Network using Linuxbridge, you need to install and configure the Linuxbridge agent on your OpenStack node. Once installed, you can create a network by defining its name, subnet, gateway, and DNS settings using the OpenStack Networking API or dashboard. You can then attach instances to the network and configure their network interfaces to use the Linuxbridge agent for connectivity.

Using Linuxbridge for network connectivity has several advantages, including simplicity, low latency, and the ability to support multiple VLANs on the same network.

TABLE OF CONTENTS

C H A P T E R No.	TITLE	PA GE NO.
	ABSTRACT	VI
	LIST OF FIGURES	IX
	LIST OF ABBREVIATIONS	X
1	INTRODUCTION	1
	1.1 Cloud Computing	1
	1.2 Advantages of Cloud computing	2
	1.3 Cloud Service Providers	3
	1.4 Openstack	4
	1.5 Advantages of openstack	7
	1.6 Instances in Openstack	8
	1.7 Types of clouds	9
	1.8 LAAS clouds	10
	1.9 SSH Client and Connections	10

	1.10 HTTP Client and Connections	11
	1.11 Ping	12
	1.12 Recommended Security Rules	13
	CHAPTER 2	14
2	SYSTEM ANALYSIS	14
	2.1 Aim & Scope	14
	2.2 How does it work	15
3	SYSTEM DESIGN AND IMPLEMENTATION	16
	3.1 Algorithm	16
4	RESULT & DISCUSSION	19
	4.1 Result	19
5	CONCLUSION	26
	5.1 REFERENCE	26

LIST OF FIGURES

FIGURE NO.	FIGURE NAME	PAGE NO.
4.1	Connecting to EC2 instance via SSH	19
4.2	Ping failing to the EC2 instance	20
4.3	Http request working	20

LIST OF ABBREVIATIONS

ABBREVIATION	EXPANSION
AWS	Amazon Web Services
EC2	Elastic Compute Cloud
GP2	General Purpose SSD

CHAPTER 1

INTRODUCTION

OpenStack continues to be the most popular open-source cloud platform and its adoption is constantly growing every year. Even though the focus on application development and delivery has shifted towards containers and Kubernetes in recent years, such environments usually run on top of OpenStack when deployed on-premises. As a result, OpenStack has become the de-facto standard for open infrastructure implementation. This is why mastering OpenStack is so important for every DevOps engineer. Even if not needed today, the knowledge of OpenStack may pay off tomorrow.

The following whitepaper provides a technical introduction to OpenStack. Starting with an overview of some basic concepts, we bring light to OpenStack's modular architecture and walk through its key components and services. We explain how all of those modules work together to deliver a fully functional cloud platform. The whitepaper follows with a brief overview of Ubuntu-based OpenStack distributions and some very simple instructions on how to get started with a single-node OpenStack on a workstation.

Cloud Computing provides an alternative to the on-premises datacentre. With an on-premises datacentre, we have to manage everything, such as purchasing and installing hardware, virtualization, installing the operating system, and any other required applications, setting up the network, configuring the firewall, and setting up storage for data. After doing all the set-up, we become responsible for maintaining it through its entire lifecycle.

But if we choose Cloud Computing, a cloud vendor is responsible for the hardware purchase and maintenance. They also provide a wide variety of software and platform as a service. We can take any required services on rent. The cloud computing services will be charged based on usage.

1.1 Cloud Computing

Cloud computing refers to the practice of utilising remote servers on the internet to store, manage, and process data rather than using a local server or a personal computer. The concept of cloud computing is based on the sharing of resources, software, and information

through a network of computers connected through the internet. This technology allows businesses and individuals to access and utilise a range of computing resources, including storage, processing power, and software applications, without the need to invest in expensive hardware or software.

Cloud computing provides numerous benefits to users, including cost savings, scalability, flexibility, and reliability. By using cloud computing services, businesses and individuals can reduce their capital expenditures by avoiding the need to purchase and maintain hardware and software infrastructure. Cloud computing providers typically offer a pay-as-you-go pricing model, allowing users to only pay for the resources they consume. Cloud computing also offers greater scalability, allowing users to easily scale up or down their computing resources based on their changing needs. Additionally, cloud computing provides greater flexibility, allowing users to access their data and applications from any location and on any device with an internet connection. Finally, cloud computing is more reliable than traditional computing, as cloud providers typically offer robust security and disaster recovery capabilities, ensuring that users' data is protected and available at all times.

However, cloud computing is not without its challenges. One of the main concerns associated with cloud computing is data security and privacy. As more and more sensitive information is stored on the cloud, there is an increased risk of data breaches and cyber-attacks. Cloud providers must take robust measures to ensure the security of their customers' data, including implementing advanced encryption and access control measures, regular security audits and penetration testing, and ensuring compliance with relevant data protection regulations such as GDPR and CCPA.

Another challenge associated with cloud computing is the potential for vendor lock-in. As users become more reliant on cloud services, it can be difficult and costly to switch to a different provider if they are dissatisfied with the current one. Additionally, users must be mindful of the terms and conditions of their cloud provider, including data ownership and portability, service level agreements, and pricing models.

1.2 Advantages of Cloud Computing

Cloud computing has revolutionised the way individuals and organisations store, process, and manage data. The cloud has become an essential part of modern technology

infrastructure due to its numerous advantages. Here are some of the most significant advantages of cloud computing.

Cost Savings:

One of the most significant benefits of cloud computing is cost savings. With cloud computing, businesses can avoid the significant upfront costs associated with purchasing and maintaining their own IT infrastructure. Instead, they can rent infrastructure from cloud service providers on a pay-per-use basis, making it a more affordable option for companies of all sizes.

Scalability:

Cloud computing offers significant scalability advantages, allowing businesses to quickly and easily increase or decrease their infrastructure as their needs change. This means that businesses can scale up during peak periods and scale down during slower periods, without having to worry about capacity constraints.

Flexibility:

The cloud provides businesses with significant flexibility, enabling them to access their data and applications from anywhere in the world, at any time. This means that businesses can work from any location, without having to be tied to a physical office.

Collaboration:

Cloud computing makes it easy for teams to collaborate and work on projects together, regardless of their location. Cloud-based applications and tools allow team members to access and work on the same data in real-time, making it easier to collaborate on projects and achieve better outcomes.

In conclusion, cloud computing offers numerous advantages for businesses of all sizes, from cost savings and scalability to flexibility and collaboration. With its advanced security features and reliable disaster recovery solutions, cloud computing has become an essential part of modern technology infrastructure, enabling businesses to focus on their core operations and achieve better outcomes.

1.3 Different Cloud Service Providers

There are many different cloud service providers in the market, ranging from large global players such as Amazon Web Services (AWS), Microsoft Azure, and Google Cloud Platform (GCP) to smaller niche providers that specialise in specific areas such as data analytics, security, or compliance. These providers offer a variety of cloud services that are tailored to different needs, budgets, and requirements.

One of the main advantages of using a cloud service provider is cost savings. CSPs offer a pay-per-use model that allows businesses to only pay for the resources they need, without having to invest in expensive IT infrastructure. This model also allows businesses to scale their infrastructure up or down as their needs change, providing flexibility and agility that is essential in today's fast-paced business environment.

Another advantage of using a cloud service provider is scalability. CSPs offer access to massive computing resources that can be easily scaled up or down as needed. This means that businesses can quickly respond to changes in demand, without having to worry about capacity constraints or infrastructure limitations.

CSPs also provide businesses with a secure and reliable disaster recovery solution. With multiple data centres in different geographic locations, CSPs ensure that businesses can quickly and easily recover their data and applications in the event of a disaster or outage. Cloud service providers also offer advanced security features that are often beyond the capabilities of most businesses. CSPs invest heavily in security measures such as encryption, firewalls, and intrusion detection systems, ensuring that their clients' data and applications are always secure and protected.

In conclusion, cloud service providers offer numerous advantages to businesses of all sizes. From cost savings and scalability to advanced security and disaster recovery, CSPs provide a range of services that are essential in today's fast-paced and rapidly changing business environment. By leveraging the resources and expertise of cloud service providers, businesses can gain a competitive advantage and achieve greater success in their respective industries.

1.4 Open Stack

OpenStack is an open-source cloud computing platform that provides a suite of software tools for building and managing cloud computing environments. It was initially released in 2010 by NASA and Rackspace, and it has since grown into a vibrant and active community of developers and users.

OpenStack is designed to provide a scalable and flexible cloud computing environment that is capable of running a wide variety of workloads, from simple web applications to complex big data and scientific computing applications. It consists of a collection of modular components that can be used to build private, public, and hybrid clouds.

Some of the key components of OpenStack include:

Nova: the compute module that manages virtual machines and other compute resources

Swift: the object storage module that provides highly scalable and durable storage for unstructured data

Cinder: the block storage module that provides persistent storage for virtual machines and other applications

Neutron: the networking module that provides virtualized networking services

Keystone: the identity and authentication module that manages user and service identities and access to resources

Glance: the image service that manages virtual machine images and other software images

OpenStack is highly modular and customizable, allowing users to choose the components that best fit their needs and build their own cloud computing environment. It is also vendor-agnostic, meaning that it can run on a wide variety of hardware and software platforms.

Overall, OpenStack is a powerful and flexible cloud computing platform that provides the building blocks for creating and managing cloud computing environments that can meet a wide range of business and technical requirements. OpenStack is an open source platform that uses pooled virtual resources to build and manage private and public clouds. The tools that comprise the OpenStack platform, called "projects," handle the core cloud-computing services of compute, networking, storage, identity, and image services. More than a dozen optional projects can also be bundled together to create unique, deployable clouds.

In virtualization, resources such as storage, CPU, and RAM are abstracted from a variety of vendor-specific programs and split by a hypervisor before being distributed as needed. OpenStack uses a consistent set of application programming interfaces (APIs) to abstract those virtual resources 1 step further into discrete pools used to power standard cloud computing tools that administrators and users interact with directly. OpenStack is a cloud computing platform that allows users to create and manage public, private, and hybrid cloud environments. It provides a set of

open-source tools for building and managing Infrastructure-as-a-Service (IaaS) and Platform-as-a-Service (PaaS) solutions.

OpenStack is designed to be highly scalable and flexible, with a modular architecture that allows users to add or remove components as needed. It includes a wide range of services, including compute, storage, networking, identity, and image management.

One of the key features of OpenStack is its ability to integrate with a wide range of other open-source and proprietary technologies, making it an ideal platform for building complex and customized cloud solutions.

OpenStack is developed and maintained by a community of thousands of developers and users from around the world, and is released under the Apache 2.0 license, making it free and open-source software.

Some of the benefits of using OpenStack include increased flexibility, reduced costs, and improved scalability and reliability. It is used by a wide range of organizations, from small startups to large enterprises and government agencies.

OpenStack is also highly customizable, with a wide range of third-party plugins and extensions available for adding additional functionality and capabilities.

In summary, OpenStack is a powerful and flexible cloud computing platform that provides users with a wide range of tools and services for building and managing cloud environments.

1.5 Advantages of openstack

OpenStack cloud solutions provide several benefits for businesses that have or are seeking to virtualize elements of their infrastructure or services. Some of these include:

Improving business agility

Increasing the efficiency of core processes

Enhancing the availability of resources

Adding flexibility and adaptability to your infrastructure

With OpenStack, you can construct your own cloud infrastructure. This may be useful for organizations that do not want to expose sensitive data, proprietary information, or the structure of their network with someone helping them set up or implement a web service like Microsoft Azure, Salesforce, or Amazon Web Services (AWS).

Instead of relying on these services, you can use open-source tools to create your own cloud within your organization's data center. These capabilities make the following benefits possible:

Shorter time-to-market. With OpenStack, you do not have to wait for the networking services you need. OpenStack puts them all at your fingertips.

Faster innovation. With OpenStack, you have the power to orchestrate and take advantage of self-service tools as you wish. Using this flexibility, you can adjust your workflow to what works best for your team—and custom-design it to make better solutions faster.

Comply with regulations easier. OpenStack gives you the power to make your own private clouds within your data center. This makes it easy to control your access privileges, ensuring that only the eyes that should see assets are able to. When you have granular control over your network policies, you do not have to worry about whether sensitive data ended up being errantly categorized or you have run afoul of another aspect of a regulation.

Need not commit to a single vendor. Because you are in control of provisioning your own cloud resources, there is no need to commit to a vendor for the long term. If you do not like one, you can just move on to the next option.

1.6 Instances in Openstack

In OpenStack, an instance is a virtual machine (VM) that runs within the cloud environment. Instances are created from images, which are templates that define the operating system and software stack for the VM.

When creating an instance in OpenStack, you can specify various parameters such as the flavor (CPU, RAM, and storage specifications), network settings, and security groups. The flavor represents the size and performance characteristics of the instance, and is defined by the cloud administrator.

Once an instance is created, you can connect to it using remote access protocols such as SSH or RDP, just like you would with a physical server. You can also manage the instance through the OpenStack dashboard or using command-line tools such as the OpenStack CLI.

Instances can be managed in a variety of ways, such as starting, stopping, rebooting, and resizing. OpenStack also supports features such as live migration, which allows you to move an instance from one host to another without downtime, and snapshotting, which allows you to create a copy of an instance at a specific point in time.

Instances in OpenStack are highly scalable, and you can create and manage large numbers of instances to meet your computing needs. This makes OpenStack ideal for organizations with fluctuating workloads or those that need to rapidly provision and de-provision resources .

1.7 Types Of Clouds

Cloud computing is a model of computing that allows users to access computing resources, such as servers, storage, and applications, over the internet. There are three primary types of clouds: public, private, and hybrid.

Public Cloud: Public clouds are provided by third-party service providers, such as Amazon Web Services (AWS), Microsoft Azure, and Google Cloud Platform. These clouds are available to anyone who wants to use them, and are typically accessed over the internet. Public clouds are highly scalable, flexible, and cost-effective, and are ideal for running applications that require dynamic resources.

Private Cloud: Private clouds are built and maintained by organisations for their own use. Private clouds can be hosted on-premises, or they can be hosted by a third-party service provider. Private clouds are ideal for organisations that require high levels of security and control over their computing resources.

Hybrid Cloud: Hybrid clouds combine elements of public and private clouds, allowing organisations to take advantage of the benefits of both. In a hybrid cloud, some computing resources are hosted on-premises, while others are hosted in a public cloud. Hybrid clouds are ideal for organisations that require high levels of security and control, while also needing the scalability and flexibility of a public cloud.

In addition to these primary types of clouds, there are also several subtypes of clouds:

Community Cloud: A community cloud is a cloud infrastructure that is shared by several organisations with similar needs. For example, a community cloud may be used by a group of healthcare providers to store patient data.

Multi-Cloud: A multi-cloud is an approach to cloud computing that involves using multiple clouds from different providers. Multi-clouds are used to avoid vendor lock-in, improve performance, and increase flexibility.

Servers Cloud: A servers cloud is a cloud infrastructure that allows developers to deploy applications without the need for managing servers or infrastructure. Servers clouds are ideal for running small, event-driven applications.

Overall, the different types of clouds provide organisations with a range of options for deploying and managing their computing resources, and can be tailored to meet the specific needs of each organisation.

1.8 LAAS Clouds

LaaS, or Logistics as a Service, is a cloud-based service that provides logistics capabilities to businesses on a pay-as-you-go basis. LaaS clouds offer businesses access to advanced logistics tools, software, and services that can help them optimise their supply chain operations, improve efficiency, and reduce costs.

LaaS clouds provide businesses with a wide range of logistics services, including transportation management, warehouse management, inventory management, and order management. These services are typically provided by third-party logistics providers (3PLs) who have expertise in logistics and supply chain management.

One of the key benefits of LaaS clouds is their scalability. Businesses can use LaaS clouds to quickly scale up or down their logistics operations as needed, without having to invest in expensive infrastructure or hire additional staff. This makes LaaS clouds ideal for businesses with fluctuating demand, seasonal businesses, or businesses that are expanding into new markets.

Another benefit of LaaS clouds is their flexibility. Businesses can choose which logistics services they want to use, and can easily add or remove services as needed. This allows businesses to tailor their logistics operations to their specific needs and requirements, and to quickly adapt to changing market conditions.

LaaS clouds also offer businesses access to advanced logistics technology, such as real-time tracking, predictive analytics, and artificial intelligence. These technologies can help businesses optimise their logistics operations, reduce costs, and improve customer satisfaction.

Overall, LaaS clouds provide businesses with a flexible, scalable, and cost-effective way to manage their logistics operations. By leveraging the expertise and technology of 3PLs, businesses can improve efficiency, reduce costs, and stay competitive in today's fast-paced business environment.

1.9 SSH Clients and Connections

SSH (Secure Shell) is a protocol that allows users to securely connect to remote computers and servers over the internet. SSH clients are software applications that enable users to

initiate and manage SSH connections. These clients provide a secure and encrypted way to access remote computers and servers, and are widely used by system administrators, developers, and other IT professionals.

SSH clients use the SSH protocol to establish a secure connection between the user's local computer and the remote computer or server. This connection allows the user to access and control the remote computer as if they were physically present at the remote location. SSH clients typically provide a command-line interface or a graphical user interface (GUI) for managing the SSH connection.

When establishing an SSH connection, the client first authenticates the user's identity using a username and password or a public/private key pair. Once the user's identity is verified, the client encrypts all data transmitted between the local computer and the remote computer, ensuring that the connection is secure and private.

There are many different SSH clients available, ranging from simple command-line tools to complex GUI applications with advanced features. Some popular SSH clients include PuTTY, OpenSSH, SecureCRT, and Bitwise SSH Client.

In addition to providing a secure way to access remote computers and servers, SSH clients also offer a range of advanced features, such as tunnelling, file transfer, and port forwarding. These features allow users to access and manage remote resources more efficiently and securely.

Overall, SSH clients play a critical role in managing remote computers and servers, and are essential tools for IT professionals who need to securely access and manage remote resources. With their advanced security features and flexible capabilities, SSH clients are a reliable and effective way to manage remote connections and access remote resources.

1.10 HTTP Clients and Connections

HTTP (Hypertext Transfer Protocol) is a protocol used for transmitting data over the internet. HTTP clients are software applications that allow users to initiate and manage HTTP connections. These clients provide a way for users to access and interact with web resources, such as websites, web applications, and web services.

HTTP clients use the HTTP protocol to establish a connection between the user's local computer and the web server hosting the web resource. This connection allows the user to send requests for data and receive responses from the server. HTTP clients can be web

browsers, mobile apps, or other software applications that are capable of sending HTTP requests.

When establishing an HTTP connection, the client sends an HTTP request to the web server, specifying the type of request (such as GET or POST), the resource being requested (such as a webpage or a file), and any additional parameters or headers required for the request. The server then processes the request and sends an HTTP response back to the client, containing the requested data or an error message if the request could not be fulfilled.

There are many different HTTP clients available, ranging from simple command-line tools to complex web browsers and mobile apps. Some popular HTTP clients include Google Chrome, Mozilla Firefox, Safari, and Microsoft Edge.

In addition to providing a way to access and interact with web resources, HTTP clients also offer a range of advanced features, such as caching, compression, and authentication. These features help to optimise the performance and security of HTTP connections and improve the user experience.

Overall, HTTP clients play a critical role in accessing and interacting with web resources, and are essential tools for users who need to browse the internet, use web applications, or consume web services. With their advanced features and flexible capabilities, HTTP clients are a reliable and effective way to manage HTTP connections and access web resources.

1.11 Ping

Ping is a network utility used to test the connectivity between two devices on a network. It is commonly used to troubleshoot network connectivity issues and to measure the response time of network devices, such as routers and servers.

When using the Ping utility, a small packet of data is sent from the source device to the destination device. The destination device then sends an acknowledgement packet back to the source device, indicating that the packet was received successfully. The time it takes for the acknowledgement packet to reach the source device is measured and reported as the round-trip time (RTT).

Ping is a simple but powerful tool for network troubleshooting, as it allows users to quickly identify connectivity issues and measure the response time of network devices. For example, if a user is unable to connect to a website, they can use the Ping utility to check whether the

website's server is responding to requests. If the server is not responding, the user can conclude that there is a connectivity issue and take steps to resolve it.

Ping is also useful for monitoring the performance of network devices, such as routers and servers. By measuring the response time of these devices over time, users can identify trends and patterns in network performance, and take proactive steps to prevent connectivity issues before they occur.

Overall, Ping is a simple but powerful tool for network troubleshooting and performance monitoring. With its ability to test connectivity and measure response times, it is an essential tool for IT professionals and network administrators who need to ensure the reliability and performance of their network devices.

1.12 Recommended Security Rules

When setting up security rules for EC2 instances, there are several recommended best practices to follow in order to ensure the security of your resources. Here are some key security rules to consider:

- **Restrict incoming traffic:** By default, all incoming traffic to an EC2 instance is allowed, which can leave your resources vulnerable to attacks. It is recommended to restrict incoming traffic by creating security groups and only allowing traffic from trusted sources.
- **Use HTTPS for web traffic:** When accessing web resources, it is recommended to use HTTPS (HTTP over SSL/TLS) to encrypt data in transit and prevent eavesdropping and tampering.
- **Disable unnecessary ports:** It is recommended to disable any unused ports on your EC2 instances to reduce the attack surface and prevent unauthorised access.
- **Use strong passwords:** It is important to use strong passwords for all user accounts and enable multi-factor authentication to prevent unauthorised access.
- **Monitor and log access:** It is recommended to monitor and log all access to your EC2 instances, including login attempts and network traffic, in order to detect and respond to potential security incidents.
- **Keep software up-to-date:** It is important to keep all software and operating systems on your EC2 instances up-to-date with the latest security patches and updates to prevent vulnerabilities and exploits.

CHAPTER 2

SYSTEM ANALYSIS

2.1 Aim & Scope

This is a living document. Its purpose is to document the OpenStack community's vision for the output of the OpenStack project as a whole, as it evolves over time. It is also aspirational, not descriptive, in nature. That is, it describes the OpenStack that the community is committed to working toward, not the OpenStack that has existed at any particular point in time.

Project teams can refer to this document when evaluating proposed features and designing interfaces, to help ensure that their designs fit comfortably within a larger structure and contribute to the overall landscape of application deployment patterns.

Amongst other things, the Technical Committee will refer to this document when assessing [new project applications](#) to determine whether they fit with the overall technical direction of OpenStack. If you are working on a project that does not appear to fit with the vision as described here, that does not necessarily mean that your project can never become an official OpenStack project, but it may mean that you should submit a patch to this document to further flesh out the vision. If this is the case, you should do so as soon as possible - there is no need to wait until you are ready to submit a new project application.

The scope of this document is limited to the cloud services that an end-user interacts with. This corresponds to the main 'OpenStack' bucket and parts of the 'OpenStack Operations' bucket in the OpenStack project map. While OpenStack also has other kinds of official projects (e.g. deployment tools and client libraries), nothing at all can be inferred about our vision for them from this document

The Technical Committee may also use this document to evaluate the suitability and priority of proposed project-wide goals.

2.2 How does it work

OpenStack is an open source platform that uses pooled virtual resources to build and manage [private](#) and [public](#) clouds. The tools that comprise the OpenStack platform, called "projects," handle the core cloud-computing services of compute, networking, [storage](#), identity, and image services. More than a dozen optional projects can also be bundled together to create unique, deployable clouds.

In [virtualization](#), resources such as storage, CPU, and RAM are abstracted from a variety of vendor-specific programs and split by a hypervisor before being distributed as needed. OpenStack uses a consistent set of application programming interfaces (APIs) to abstract those virtual resources 1 step further into discrete pools used to power standard [cloud computing](#) tools that administrators and users interact with directly.

Is OpenStack just a virtualization management platform?

Not quite. There are a lot of similarities, but they're not the same.

Yes, OpenStack and [virtualization management](#) platforms both sit on top of virtualized resources and can discover, report, and automate processes in vendor-disparate environments.

But while virtualization management platforms make it easier to manipulate the features and functions of virtual resources, OpenStack actually uses the virtual resources to run a combination of tools. These tools create a cloud environment that meets the [National Institute of Standards and Technology's](#) 5 criteria of cloud computing: a network, pooled resources, a user interface, provisioning capabilities, and automatic resource control/allocation.

OpenStack is essentially a series of commands known as scripts. Those scripts are bundled into packages called projects that relay tasks that create cloud environments. In order to create those environments, OpenStack relies on 2 other types of software:

- Virtualization that creates a layer of virtual resources abstracted from hardware
- A base operating system (OS) that carries out commands given by OpenStack scripts

Think about it like this: OpenStack itself doesn't virtualize resources, but rather uses them to build clouds. OpenStack also doesn't execute commands, but rather relays them to the base OS. All 3 technologies—OpenStack, virtualization, and the base OS—must work together. That interdependency is why so many OpenStack clouds are deployed using [Linux®](#), which was the inspiration behind [RackSpace and NASA's decision](#) to release OpenStack as open source software

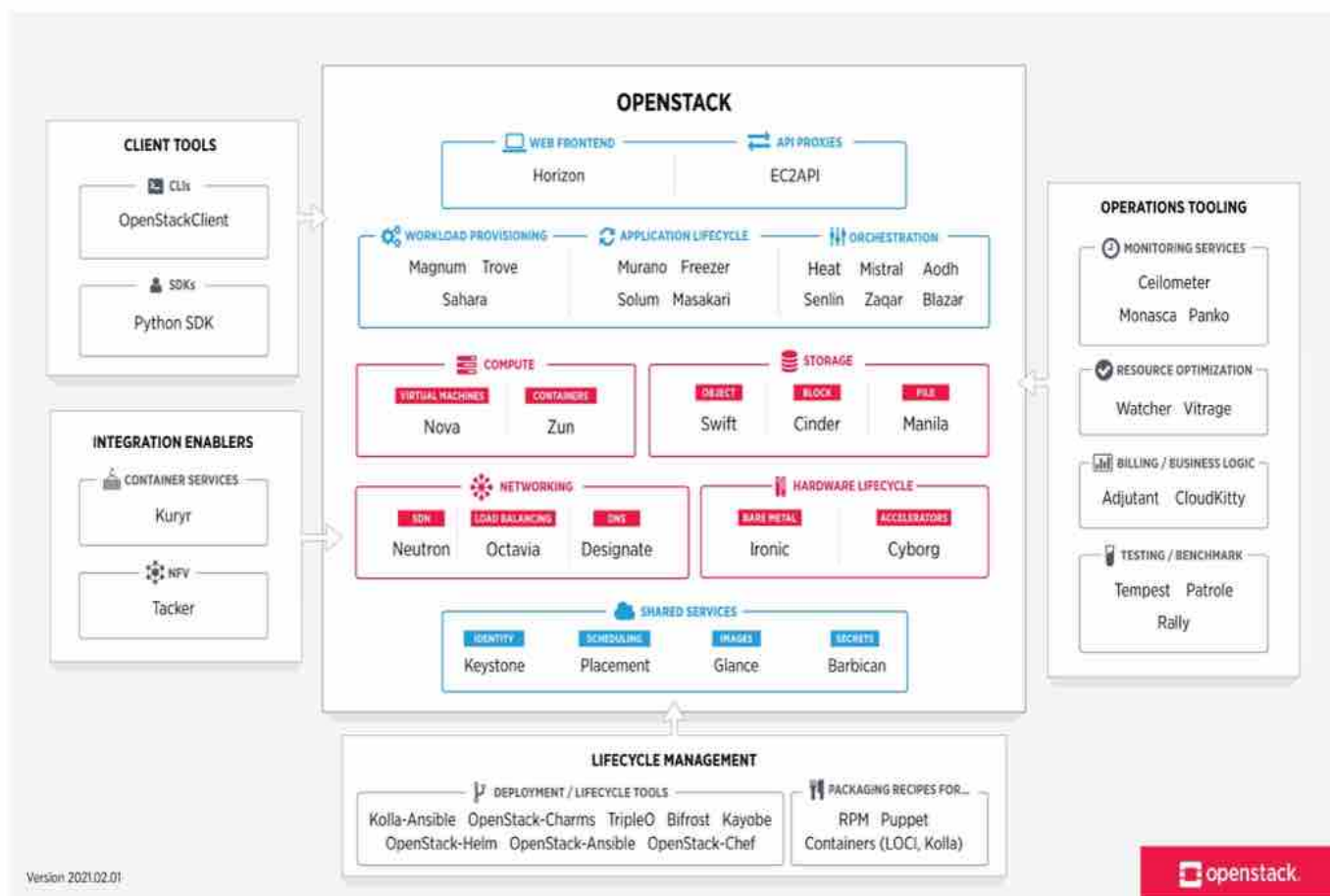
CHAPTER 3

SYSTEM DESIGN AND IMPLEMENTATION

3.1 Steps

1. Planning and architecture: This involves identifying the requirements of the system, determining the architecture of the OpenStack deployment, and selecting the appropriate components for the deployment.
2. This step also involves deciding on the hardware and network infrastructure requirements.
3. Installation and configuration: This step involves installing Ubuntu OpenStack on the chosen hardware and configuring the system to meet the requirements identified in step 1.
4. This includes setting up the network infrastructure, configuring the storage, and installing and configuring the OpenStack components.

5. Testing and validation: Once the installation and configuration are complete, the system must be tested and validated to ensure that it meets the requirements and performs as expected.
 6. This involves testing the various OpenStack components and verifying that they are functioning correctly.
 7. Maintenance and monitoring: Once the system is in production, it must be maintained and monitored to ensure that it continues to operate as expected.
 8. This includes monitoring the performance and availability of the system, applying patches and upgrades as needed, and performing regular backups to protect against data loss.
- To implement a system design in Ubuntu OpenStack, you will need to have a good understanding of the architecture and components of OpenStack, as well as the requirements of your specific deployment.
 - It is also important to have experience with Linux system administration and network infrastructure, as well as knowledge of virtualization technologies such as KVM and QEMU.
 - There are many resources available online to help you learn more about OpenStack and Ubuntu, including documentation, tutorials, and community forums.



CHAPTER -4

RESULTS & DISCUSSION

4.1 Results

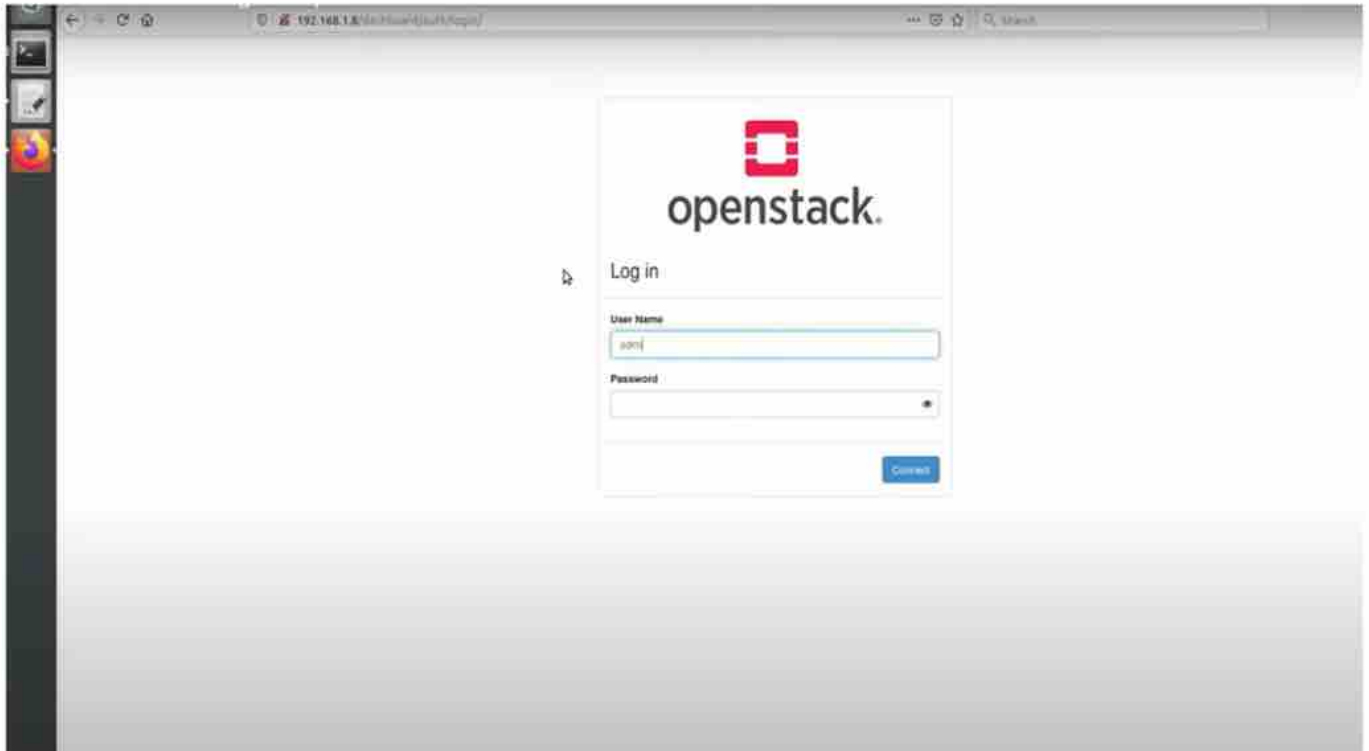


Fig.4.1:OPENSTACK DASHBOARD

Launch an instance

This section creates the necessary virtual networks to support launching instances. Networking option 1 includes one provider (external) network with one instance that uses it. Networking option 2 includes one provider network with one instance that uses it and one self-service (private) network with one instance that uses it.

The instructions in this section use command-line interface (CLI) tools on the controller node. However, you can follow the instructions on any host that the tools are installed.

For more information on the CLI tools, see the [OpenStackClient documentation for Pike](#), the [OpenStackClient documentation for Queens](#), or the [OpenStackClient documentation for Rocky](#).

To use the dashboard, see the [Dashboard User Documentation for Pike](#), the [Dashboard User Documentation for Queens](#), or the [Dashboard User Documentation for Rocky](#).

Create virtual networks

Create virtual networks for the networking option that you chose when configuring Neutron. If you chose option 1, create only the provider network. If you chose option 2, create the provider and self-service networks.

- [Provider network](#)
- [Self-service network](#)

After creating the appropriate networks for your environment, you can continue preparing the environment to launch an instance.

Create m1.nano flavor

The smallest default flavor consumes 512 MB memory per instance. For environments with compute nodes containing less than 4 GB memory, we recommend creating the **m1.nano** flavor

```
$ openstack flavor create --id 0 --vcpus 1 --ram 64 --disk 1
```

Field	Value
OS-FLV-DISABLED:disabled	False
OS-FLV-EXT-DATA:ephemeral	0
disk	1
id	0
name	m1.nano
os-flavor-access:is_public	True
properties	
ram	64
rxtx_factor	1.0
swap	
vcpus	1

Fig:4.4

Generate a key pair

Most cloud images support public key authentication rather than conventional password authentication. Before launching an instance, you must add a public key to the Compute service.

1. Source the demo project credentials:

```
$ ssh-keygen -q -N ""
$ openstack keypair create --public-key ~/.ssh/id_rsa.pub
```

Field	Value
fingerprint	ee:3d:2e:97:d4:e2:6a:54:6d:0d:ce:4
name	mykey
user_id	58126687cbcc4888bfa9ab73a2256f27

Fig:4.5

Persistent Storage for Amazon EC2

I would like to introduce to you the newest feature of Amazon EC2: Persistent local storage. This has been very high on the request list of EC2 customers and I believe that combined with the [Availability Zones](#) and [Elastic IP Address](#) features released earlier this month this makes EC2 the ideal environment for building highly scalable and reliable applications.

Significant innovation has gone into this feature: Instead of restricting developers to the use of a particular (distributed) file-system we once again decided to look at what is the most fundamental building block and how we could offer that in the most scalable and reliable manner.

Persistent storage for Amazon EC2 will be offered in the form of storage volumes which you can mount into your EC2 instance as a raw block storage device. It basically looks like an unformatted hard disk. Once you have the volume mounted for the first time you can format it with any file system you want or if you have advanced applications such as high-end database engines, you could use it directly.

Developers can create any number of volumes they want, in size ranging from 1 GB to 1TB. This volume will be created within a specified Availability Zone and will be accessible by your EC2 instances running in that Availability Zone. As to be expected with a volume abstraction only one instance can have the volume mounted at any given time. Volumes can migrate and be reattached to other instances if necessary for failure handling or application migration reasons.

The consistency of data written to this device is similar to that of other local and network-attached devices; it is under control of the developer when and how to force flush data to disk if you want to bypass the traditional lazy-writer functionality in the operating systems file-cache. Because of the session oriented model for access to the volume you do not need to worry about eventual consistency issues.

Storage

Amazon EC2 provides you with flexible, cost effective, and easy-to-use data storage options for your instances. Each option has a unique combination of performance and durability. These storage options can be used independently or in combination to suit your requirements.

After reading this section, you should have a good understanding about how you can use the data storage options supported by Amazon EC2 to meet your specific requirements. These storage options include the following:

- [Amazon Elastic Block Store](#)
- [Amazon EC2 instance store](#)

- [Use Amazon EFS with Amazon EC2](#)
- [Use Amazon S3 with Amazon EC2](#)

The following figure shows the relationship between these storage options and your instance.

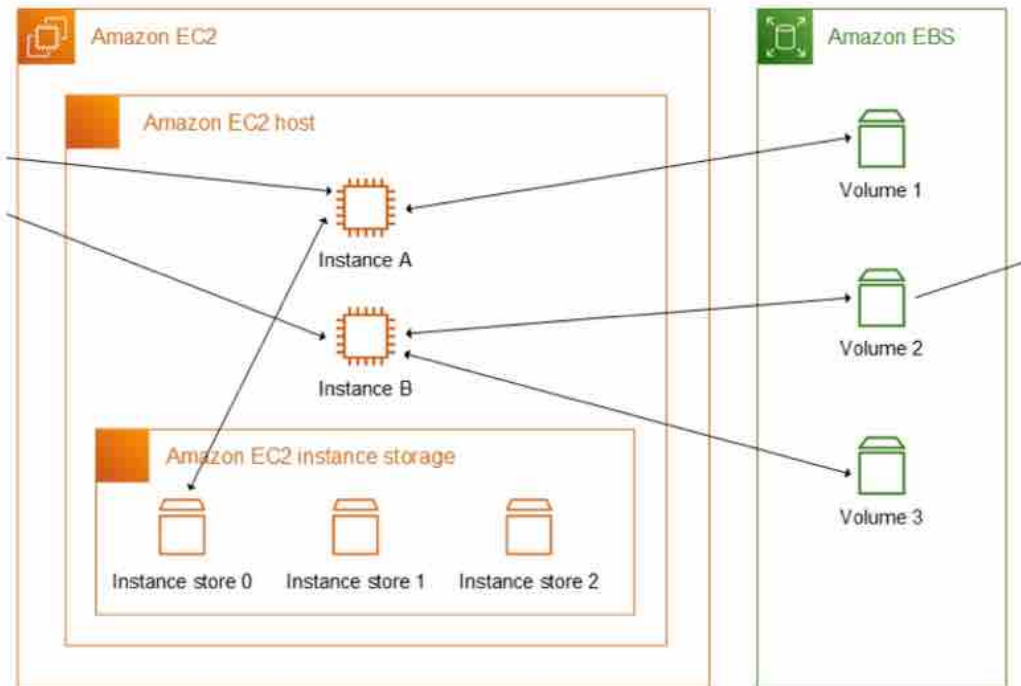


Fig:4.6

Amazon EBS

Amazon EBS provides durable, block-level storage volumes that you can attach to a running instance. You can use Amazon EBS as a primary storage device for data that requires frequent and granular updates. For example, Amazon EBS is the recommended storage option when you run a database on an instance.

An EBS volume behaves like a raw, unformatted, external block device that you can attach to a single instance. The volume persists independently from the running life of an instance. EBS volumes can also be created as encrypted volumes using the Amazon EBS encryption feature. For more information, see [Amazon EBS encryption](#).

To keep a backup copy of your data, you can create a *snapshot* of an EBS volume, which is stored in Amazon S3. You can create an EBS volume from a snapshot, and attach it to another instance. For more information, see Amazon Elastic Block Store.

Amazon EC2 instance store

Many instances can access storage from disks that are physically attached to the host computer. This disk storage is referred to as *instance store*. Instance store provides temporary block-level storage for instances. The data on an instance store volume persists only during the life of the associated instance; if you stop, hibernate, or terminate an instance, any data on instance store volumes is lost. For more information, see Amazon EC2 instance store.

Amazon EFS file system

Amazon EFS provides scalable file storage for use with Amazon EC2. You can create an EFS file system and configure your instances to mount the file system. You can use an EFS file system as a common data source for workloads and applications running on multiple instances. For more information, see Use Amazon EFS with Amazon EC2.

Amazon S3

Amazon S3 provides access to reliable and inexpensive data storage infrastructure. It is designed to make web-scale computing easier by enabling you to store and retrieve any amount of data, at any time, from within Amazon EC2 or anywhere on the web. For example, you can use Amazon S3 to store backup copies of your data and applications. Amazon EC2 uses Amazon S3 to store EBS snapshots and instance store-backed AMIs. For more information, see Use Amazon S3 with Amazon EC2.

Adding storage

Every time you launch an instance from an AMI, a root storage device is created for that instance. The root storage device contains all the information necessary to boot the instance. You can specify storage volumes in addition to the root device volume when you create an AMI or launch an instance using *block device mapping*. For more information, see Block device mappings.

CHAPTER -5

CONCLUSION

OpenStack is a cloud computing platform that provides a wide range of services, including computing, storage, and networking. One of the key components of OpenStack is the image service, which allows users to create and manage virtual machine images.

Overall, OpenStack image service provides a robust and flexible way to manage virtual machine images. It offers a range of features, such as support for multiple image formats, image sharing, and image versioning. Additionally, it integrates well with other OpenStack services, making it easy to create and deploy virtual machines in the cloud.

However, managing OpenStack images can be complex, especially for users who are not familiar with the platform. It requires knowledge of various OpenStack components and tools, and it may take time to set up and configure the image service. Additionally, there may be compatibility issues with certain operating systems or hypervisors.

In conclusion, OpenStack image service is a powerful tool for managing virtual machine images in the cloud. However, it may require some expertise and effort to use effectively, and users should carefully consider their needs and resources before adopting it.

5.1 REFERENCE

- T. Buyya, C.S. Yeo, S. Venugopal, J. Broberg, and I. Brandic, "Cloud Computing and Emerging IT Platforms: Vision, Hype, and Reality for Delivering Computing as the 5th Utility," *Future Generation Computer Systems*, vol. 25, no. 6, pp. 599-616, Jun. 2009.
- D. Nurmi, R. Wolski, C. Grzegorzczak, G. Obertelli, S. Soman, L. Youseff, and D. Zagorodnov, "The Eucalyptus Open-source Cloud-computing System," *Proceedings of the 1st Workshop on Cloud Computing and Its Applications*, pp. 124-131, Dec. 2008.
- B. Sotomayor, R. S. Montero, I. M. Llorente, and I. Foster, "Virtual Infrastructure Management in Private and Hybrid Clouds," *IEEE Internet Computing*, vol. 13, no. 5, pp. 14-22, Sept. 2009.
- Y. Wei, Z. Zheng, and Y. Yang, "A Study on Cloud Computing Security," *2010 International*

Conference on Computational Intelligence and Security, pp. 138-141, Dec. 2010.

S. K. Garg and R. Buyya, "Green Cloud Computing: Balancing Energy in Processing, Storage, and Transport," Proceedings of the 19th International Symposium on High Performance Distributed Computing, pp. 1-12, Jun. 2010.

S. P. Chowdhury, M. R. Rahman, and R. Boutaba, "Virtual Machine Migration in Cloud Computing Environments: A Survey," IEEE Communications Surveys & Tutorials, vol. 13, no. 3, pp. 452-469, Third Quarter 2011.

J. Altmann, J. G. Pérez-Sánchez, and J. A. Caminero, "Economic Models for Cloud Service Markets: Pricing and Quality of Service," Journal of Grid Computing, vol. 11, no. 1, pp. 57-79, Mar. 2013.

R. Buyya, R. Ranjan, and R. N. Calheiros, "Modeling and Simulation of Scalable Cloud Computing Environments and the CloudSim Toolkit: Challenges and Opportunities," Proceedings of the 7th High Performance Computing and Simulation Conference, pp. 1-11, Jul. 2015.

Y. Li, S. Li, Z. Li, Y. Yang, and R. Li, "A Survey of Cloud Computing and Security Issues," Journal of Network and Computer Applications, vol. 43, pp. 1-10, Nov. 2014.

J. M. Alvarez-Rodríguez, M. J. Suarez-Cabal, and J. F. Martínez-Santos, "Adaptive Monitoring in Cloud Computing Environments," IEEE Transactions on Services Computing, vol. 9, no. 1, pp. 30-42, Jan.-Feb. 2016.