I. J. Computer Network and Information Security, **2018, 3, 1-7**
Published Online March 2018 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijcnis.2018.03.01

An Implementation of Software Routing for Building a Private Cloud

Rawezh Ziad Kamla[1], Tara Yahiya[1]
[1]University of Kurdistan- Hewl ĝ / Department of Computer Science and Engineering, Erbil, 44001, KRG-Iraq
E-mail: {r.ziad3, t.ibrahim1}@ukh.edu.krd

Nashwan B. Mustafa[2]
[2]Health Queensland/Queensland Government, Brisbane, 4001, Australia
E-mail: Nashwan.mustafa@health.qld.gov.au

Received: 23 November 2017; Accepted: 17 January 2018; Published: 08 March 2018

***Abstract*—**The demand on cloud computing is increasing, more organizations tend to use it to store and process their data. In this article, we address some challenges starting by building a private cloud from our own company's old devices, and then implementing some functionalities that a private cloud can offer to its users. Since cloud computing is a paradigm which is based mainly on a virtualized environment, therefore we used Proxmox Virtual Environment which is an open source free server virtualization technology for this purpose. Then we deployed software routers on the virtual routers through Quagga software to perform the routing functionality among the virtual machines. Finally, and in order to show the real use of our private cloud, an open source Nextcloud service is installed which is a free file sharing software that is used to show Software as a Service (SaaS) usage of our private cloud. We tested our implementation of private cloud computing through two case studies that showed a successful access of a user to the Nextcloud service. In the same time, we tested the routing functionality of the private cloud through the use of Quagga software router without using a physical router. As a result, our private cloud is fully oriented open source, cost effective and reliable.

***Index Terms*—**Private Cloud, Quagga, Nextcloud, Proxmox.

I. Introduction

Over the past decade, cloud computing became more popular and widely usable as it is started to take the characteristics of a ubiquitous service. Cloud computing has a significant impact on the worldwide technology industry and business corporation.

Nowadays cloud is considered as a place to store and process our data. In most of the cases, one cannot know where and how these data have been kept and processed, this might not be critical for personal data but when it comes to organizational critical data, it really matters to know exactly where and how the data was stored and who has the access to the data. Using a public cloud might not be an efficient and a good solution in terms of performance, security in terms of data protections and economical in terms of cost solution for a medium or large scale business organization. It's more practical to use a private cloud which the organization itself owns the infrastructure, physical and software components of that private cloud. The aim of this article is to implement a private cloud to address the previously mentioned objectives. It's clear that cloud is reducing operational and maintenance costs through using virtualization technologies and software defined solutions instead of using dedicated hardware for computing, routing and network switching solutions. The main objectives of this article is to assist small companies or any other institutions to build their private cloud in less cost, best performance, higher security issue.

Accordingly, the main contributions of this work can be represented by enabling the organizations to build their own private cloud with a lowest cost, which can be performed through using open source, free hypervisor such as Proxmox Virtual Environment to implement the server virtualization technology with , and reusing used physical equipment's such as server nodes and network switches. Another key solution is to implement the software based router and virtual switching solutions to perform the routing functionalities on a virtualized environment, this also can be done using open source free software tools such as Quagga routing suite and Linux Bridge.

The rest of this paper is organized as follows section II introduces the Background and Literature Survey, section III Describes the Data Center Architecture, section IV Tackles the Implementation and Case Studies and section V concludes the article.

II. Background and Literature Survey

A cloud is defined as a place, over network infrastructure, where information technology (IT) and computing resources such as computer hardware, operating systems, networks, storage, databases and even entire applications are available instantly, on demand. It

facilitates sharing of infrastructure (server space), platform and services. Basically cloud technology can be separated into three main types in terms of Infrastructure used to run the service and the entity which own, manage and support the cloud service [1].

Public Cloud is an attractive solution to small and mid-size business owners, as it eliminates the cost of building their own computing, networking and storage infrastructure in the case of small and medium businesses, as they can use cloud resources and pay per their usage needs, nowadays there are many cloud providers but the top ones are Amazon Web Services, Google Computing Engine and Microsoft Azure [2]. Private Cloud is hosted by the same company that uses it therefore; it provides more security than a public cloud [3]. In our research we focus on building a private cloud using the software router technologies as well as discussing the main required equipment's in building data center infrastructure. Hybrid Cloud is a combination of public and private cloud when they are used together. For example, an organization can send an on-premises private cloud to host sensitive or critical workloads, but utilize a third-party public cloud supplier, for example, Google Compute Engine or Amazon Web Services, to have less-basic assets, for example, test and development workloads [4].

We can classify the clouds in terms of services that they are providing into the following services. Infrastructure as a Service (IaaS) Provides users with processors, storage, networks and other computing resources as a service and the users do not have to control or manage the infrastructure but they do have control over the operating systems (OS), applications and programming frameworks. Platform as a Service (PaaS) provides hardware services with operating systems. User just has to install the required applications on those hardware resources [5]. Software as a Service (SaaS) Such kind of service allows multiple end users to access applications which are running on cloud infrastructure through a web browser. The end users do not manage or control the software in the cloud.

In the Internet world, the demands on online real- time services are increasing, nowadays everything is moved to the cloud, the pictures which we upload to a social media network, the files that we share over Internet online shopping agencies and etc. This huge amount of online services requires a huge amount of (i) computing (ii) memory (iii) storage and (iv) network resources which means using more server nodes, network equipment's and storage units that obviously will increase the power consumption, occupied space in Data Centers and requires more qualified IT engineers and technical to monitor, maintain and operate these services. The two most important technologies that have been applied in cloud computing industry are Server Virtualization and Virtual Networking which we will go through them in the next sections of this article.

Virtualization is the ability to create a virtual version of a device or resource such as server, storage, network devices rather than using the actual physical devices. A key use of virtualization technology in cloud computing is server virtualization, which uses a software layer called a hypervisor to emulate the underlying hardware as its shown in Figure 1. This often includes the CPU's memory, I/O and network traffic.

The hypervisor enables multiple OS to reside on a single physical server node, which provides each OS to interact with a CPU and memory and often the guest OS has no idea that it's on a virtualized hardware [4].

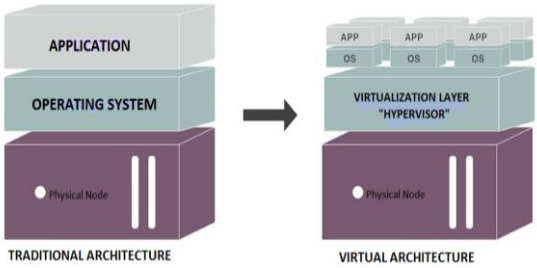

Fig.1. Virtual vs Traditional Architecture

There are different types of hypervisor which provide server virtualization technology such as VMware vSphere ESXi, Cetrix XenServer and Proxmox Virtual Environment.

The implementation of a virtualized environment requires a different type of networking than the traditional model which is not well suited for a virtualized environment, which is Virtual Networking and the two key elements of virtual networking are Virtual Switches such as VMware Virtual Switch and Proxmox Linux Bridge and Software Routers such as XORP, BIRD and Quagga. In this article we a aim to show the implementation of Quagga software router for a private cloud.

There are some research works achieved in the same domain as ours and close to our topic. The authors in [6] show the implementation of Quagga with OpenFlow, Quagga which is an open source routing suit and open flow is an open standard based on an Ethernet switch. In this structure, Openflow is responsible for a fast packet forwarding (data path) and Quagga is responsible for high level routing decisions (control path). And the results state that routes from Quagga are properly converted into flow entries and data plane packets are correctly forwarded.

In [7] authors discuss the need for higher performance and time consuming in switching time in network architecture is increasing due to diffusion of information and communication technologies and new applications that more rely on audio and video streaming. In this paper, the performance parameter has been introduced in terms of switching time when a network topology changes occur. The test methodology to measure the performance parameter is done on a personal computer with Linux operating system equipped with Quagga 0.98 Routing Software.

In another research the authors [8] worked on the implementation of software defined networking (SDN) in Openstack cloud platform through using Quantum

network manager which was beneficial through increasing emphasis of features of networking in OpenStack open source cloud computing platform.

Authors in [9] describe the necessity of virtual networking and SDN, in their paper they survey how Software Defined Networks evolved to be one of the most preferred technology of contemporary times. And in [10] authors show the study of different types of hypervisors which they compare the ESXi 4.1, XenServer 6.0 and KVM for guest isolation capacity evaluation.

III. Data Center Architecture

When the term "Cloud Computing" comes into mind, software based solution can be envisioned for such environment, which is partially true but there would be also a missing part which is represented by the physical equipment's that run the cloud services and provide the infrastructure for the access to these services. Basically there are three main categories of the cloud physical components namely (i) Computing units (Servers) (ii) Networking infrastructure (Router, Switch and Firewall) and (iii) Storage and backup solutions. For all of these equipment's, we should have a place which can keep and integrate all these different parts together. It's a Data Center which is a facility made out of arranged networked, computation, storage, power and cooling systems used by organizations or different associations in order to process, store and spread large amount of data. Data Center consists of two main components; Passive components such as Data Center physical building, Raised Floor, Ceiling Solution, Cooling System, Fire Suppression System and Power Solutions, and Active components such as Internet Gateway Router, Internet Switch, Firewall, Core switch and Server nodes, as its shown in Figure 2 .We will discuss them in details in the following sections.

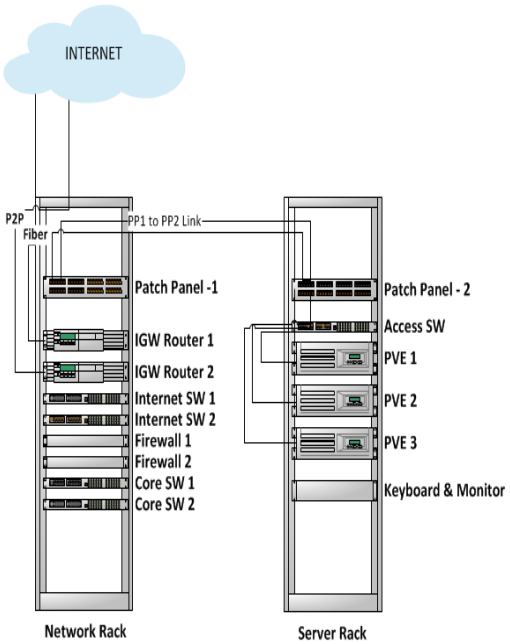

Fig.2. Physical Components Diagram.

A. Data Center Network Infrastructure

Internet Gateway Routers, the data center is composed of two main Internet Gateway (IGW) Routers, which are connected to the Internet through fiber and Point to point (PPP) connections consecutively. The reason behind the use of the PPP is to keep a redundant connection to Internet in case the first IGW failed. Open Shortest Path First (OSPF) which is a dynamic routing protocol is configured on both IGWs for their connection to the Internet Service Provider (ISP), (cf. section A of Figure 3).

Internet Switches, as previously mentioned that the data center has fiber and PPP internet connection in addition to a primary and secondary Internet Gateway Routers to support fully redundant internet connectivity. Indeed, it is well known that routers and Firewalls (FW) are limited in terms of physical Interfaces, that's why a network switch is required to work as an intermediate node to establish the physical connectivity between the IGWs and FWs (cf. section B of Figure 3).

Firewall, Network security is one of the most critical issues in data centers. Firewalls are the key solution to keep the internal network fully separated from the outside (Internet) network, firewall allows a controlled exchange of data between these two networks based on predefined access rules, IP address filtering, port security, protocol filtering, etc. In our data center a redundant high performance firewalls have been used, the firewalls work as intermediate node between internal network and external network, (cf. C section of Figure 3).

Core Switch, the core switches stand between the firewall and access switch, the two core switches have been used for redundancy and high availability. The two switches are connected back-to-back via stack cable, (cf. section D of Figure 3).

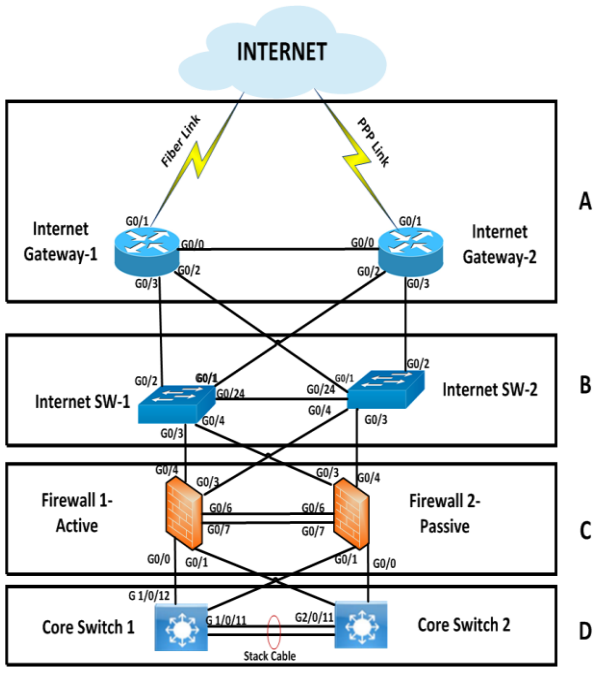

Fig.3. Network Architecture

B. Computing Component

The computing part consists of three Dell PowerEdge 2950 servers which are connected to each other via the access switch. The logical Diagram is depicted in Figure 4.

Characteristics of the server nodes:

- Model: Dell PowerEdge 2950
- CPU: two quad-core or dual-core Intel Xeon® 5400 series standard up to 3.16GHz
- Memory: 16 GB 4* 4GB 667MHz
- Hard Drive: 1 TB SATA
- Network Interface Card: Dual embedded Broadcom NetXtreme II™ 5708 Gigabit Ethernet NIC

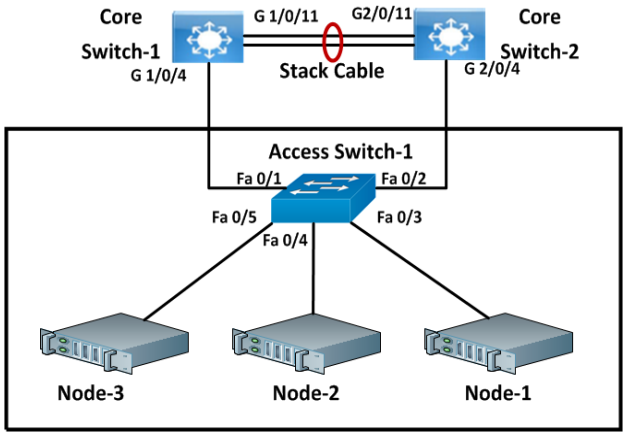

Fig.4. Computing Nodes.

IV. Implementation

In this section we will describe the steps required to build cloud for an enterprise using a fully open source technologies from virtualization, operating system, software routing and cloud services. To build an on-premises cloud, we must have standardized – and documented -- procedures for operating, deploying and maintaining the cloud environment. There are many reasons to push organizations to build their own private cloud instead of using a public cloud such as Security Increase, Cost Reduction, Data Center Visibility Increase and Performance Increase. In the following section we are going through the building and implementation parts of the private cloud.

A. Server Virtualization Technology Tools

Virtualization is the process of creating a software-based (or virtual) representation of hardware rather than a physical one. In our implementation we focus more on server virtualization as it's been discussed in technical background section. There are many technologies or hypervisors which provide server virtualization, in the following section we focus on the hypervisor which we used in our implementation.

1. Proxmox Virtual Environment

It is a complete open source server virtualization management software. It is based on Kernal Based Virtual Machine (KVM) virtualization. Kernal Based Virtual Machine is an open source hypervisor; KVM is a full virtualization solution for Linux on x86 hardware containing virtualization extensions (Intel VT or AMD-V). In addition Proxmox VE is based on Linux containers (LXC) which is an operating-system-level virtualization environment for running multiple isolated Linux systems on a single Linux control host, storage, virtualized networks, and HA clusters. Proxmox VE uses a Linux kernel and is based on the Debian GNU/Linux Distribution. The source code of Proxmox VE is released under the GNU Affero General Public License, version 3 (GNU AGPL, v3). This means that we are free to inspect the source code at any time or contribute to the project using our source codes [11]. We chose Proxmox VE, which has many brilliant features which inspired us to use it in our private cloud project just like the following:

- Open source software
- No vendor lock-in
- Linux kernel, with which we won't have any compatibility issue running Linux serves as virtual machine and then implementing Quagga software router which is also Linux based.
- Fast installation
- Web-based management interface
- Huge active community from forums, FAQ and free support.
- Low administration costs and simple deployment

The first step is to install Proxmox VE on our three servers by enabling Virtualization mode on their CPUs, then follow the installation steps. The second step is to perform network configuration (Host Name, IP Address and etc.).

Then after installing the Proxmox VE on each server node now it's time to create a Proxmox VE cluster from the three nodes which we have, Proxmox VE 4.x (and all versions above) cluster enables central management of multiple physical servers. A Proxmox VE Cluster consists of several nodes (up to 32 physical nodes, probably more, depend ending on network latency).

2. Virtual Machines

A virtual machine is an operating system which is installed on a hypervisor which is previously installed on a physical hardware; on one physical host we can install multiple Virtual Machines (VMs). The hypervisor is responsible for controlling and allocating the physical resources (CPU, Memory, Storage and Network interfaces) to the VMs. Now our host servers and the cluster are ready to install the first Virtual Machine, the first step is to prepare the resources (CPU, Memory, Hard Drive and NIC) for each virtual machine then to upload the ISO image and install the OS. In our implementation we use Ubuntu server 16.04 LTS.

B. Virtual Switching

As its mentioned in section II that bridges are like physical network switches implemented in software on the Proxmox VE host. PVE bridges can be created and configured from both command line and web interfaces. The networking configurations are stored in /etc/network/interfaces [12].

C. Virtual Routing

One of the network key solutions for a virtualized environment is software routing. We mentioned three different types of software routing tools. In our implementation, we use Quagga Routing Suite to perform the network connectivity between VMs each on different network subnet. We chose Quagga because it is one of the most popular software routing tools, its command line is almost the same as Cisco Internetworking Operating System (IOS) which is a leading company in network solutions. The first stem to implement Quagga Routing is to download the Quagga files and then install it on one of the virtual machines then the second step is to create the configuration files then to enable the desired routing protocol (OSPF, RIP or BGP) [13]. In our implementation, we used OSPF protocol, then we can start configuring the router and initialize the IP subnets on the routers interfaces. Figure 5 depicts the case of using the virtual routers which enables the virtual machines to communicate with different network subnets.

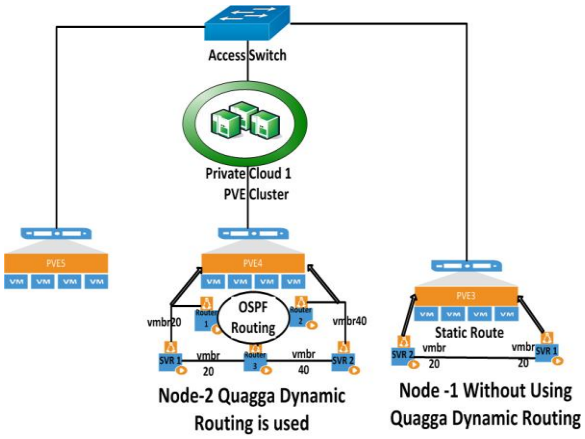

Fig.5. Virtual Routing

D. NextCloud Services

In order to demonstrate the practical part of the SaaS implementation in our architecture, *Nextcloud* service is used as an example of SaaS. *NextCloud* is a free and open source cloud software suite for creating file sharing services like documents and pictures in a centralized location, it's much similar to Dropbox but company or organization can install it and use it on the private cloud. We installed *Nextcloud* version -10.0.1 on Ubuntu server 16.04 LTS, in order to be able installing *NextCloud*. First of all we have to install a web server, database and PHP to be properly functional. We used Apache2 Server, MySQL database and PHP). After preparing the server with all prerequisites that *Nextcloud* needs, then we started installing *Nextcloud* service [14].

E. Case Studies

In this section we show the applicability of our private cloud computing environment and its use. Like any system implementation, we have to test its practicability through some use cases that show how the system is working. For our environment, we choose two case studies. Since private cloud is adapted to be used by enterprises, then there will be two kinds of users to access the cloud; namely the employee and the administrator of the cloud. The employee can access the SaaS from outside the enterprise using VPN tunnel. The SaaS in our case is represented by *Nextcloud*.

1. Case Study 1

This case study demonstrates the use of Nextcloud service which is the implementation of using our private cloud for SaaS, the following steps are valid for this case:

An employee is trying to connect to the private cloud using a VPN tunnel, for doing so, the employee logs into the VPN client profile using username and password, the VPN user authentication is checked inside the data center firewalls, if the login was successful the remote client is connected to the private cloud network, if not the user should retry the user authentication requirements again.

The authenticated user requests the IP address of the server where Nextcloud is hosted. The served resides in TEST VLAN on the data center firewall. The request will be directed to the Core Switch on TEST VLAN and then the user reaches the private cloud server. In this stage the user enters the IP address and the port number of the VM where Nextcloud server resides. If the entered combination of IP address and the port number is valid then the user can access the web interface of Nextcloud service. Otherwise the user should retry entering a valid combination. Once the web interface of Nextcloud appears it requires the user to login with his/her Nextcloud account which requires a user name and password then the user will be able to start using the Nextcloud services.

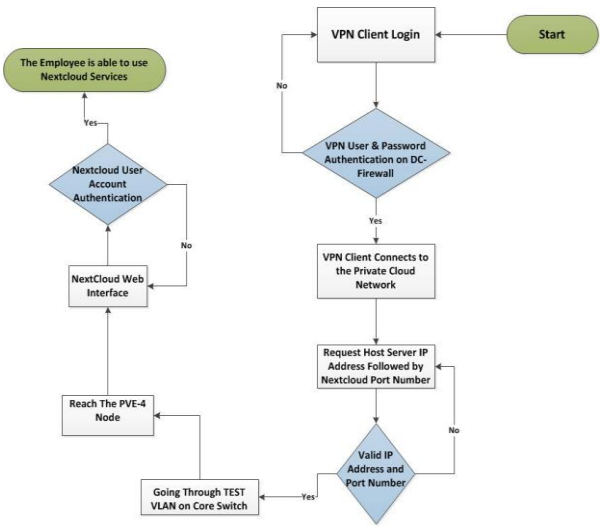

Fig.6. Case Study 1.

2. Case Study 2

This case study shows clearly the implementation of IaaS of our private cloud, the first requirement is that we should have a VPN client profile in order to be able to reach the private cloud network, once the VPN user authentication is done successfully, the cloud admin will be connected to the network which the private cloud is using, as it's discussed before that the VPN user authentication is done on the data center firewall. Now the VPN connection is established and the cloud admin requests the PVE cluster's IP address followed by PVE port number from his/her own computer web browser, the PVE cluster which resides on the TEST VLAN network will forward the request to the data center Core switch and then to the PVE cluster. In this step the PVE cluster web interface requires the admin authentication of PVE cluster user account using username and password, after the successful authentication the admin will be able to see the web interface management of the PVE cluster and then can access the virtual machines which are created to work as a software router inside the virtual machine. The admin can access software routes via command line interface (CLI) which requires the router's password. This is the final step of authentication requirement, now the admin has a full access to configure the routers and perform modification in network topology of the VMs.

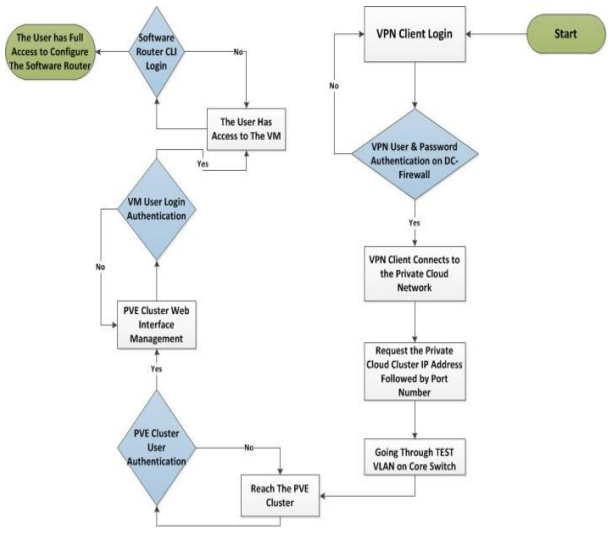

Fig.7. Case Study 2

V. Impementation and Test

To implement a virtual routing using a software router such as Quagga, we built a testbed environment from three physical servers, installed PVE hypervisor on each of the servers then virtual machines have been created. As it was shown in Figure 5 on Node-2 , three of the virtual machines configured to function as a router using Quagga software and OSPF Dynamic routing configuration which enabled the other virtual machine that run ordinary services to communicate with each other even though they are on different network subnet.

VI. Conclusion

The key issues with using public cloud by medium and large business organizations are security, cost and performance. Most of the large organizations prefer keeping their data in their hands specially the organizations which have critical data such as governmental entities. In this article, we implemented a real private cloud which can be a good prototype for any educational, business or governmental organization to build their own private cloud on the basis of this article. As for the cost, we could build our private cloud using old equipment's from servers and other networking components. Besides, the reduced cost is represented by the use of free open source software solutions such as Proxmox Virtual Environment as a server virtualization technology. Proxmox has an efficient resource management as well as a practical single interface management for all the server nodes which the private cloud is required to use. Open source Ubuntu server is installed as an OS for the virtual machines on the Proxmox VE and on the top of the virtual machines we implemented the software routing using Quagga which is also open source software routing solution. We deployed OSPF dynamic routing protocol just to show that Quagga software routers are functional on the virtualized environment.

List of Acronyms

BGP: Border Gateway Protocol
BIRD: Bible Internet Routing Daemon
DC: Data Center
IaaS: Infrastructure as a Service
IGW: Internet Gateway
IP: Internet Protocol
IS-IS: Intermediate System – Intermediate System
ISP: Internet Service Provider
KVM: Kernal Based Virtual Machine
OSPF: Open Shorts Path First
PaaS: Platform as a Service
PPP: Point- to- Point
PVE: Proxmox Virtual Environment
RIP: Routing Information Protocol
SaaS: Software as a Service
SDN: Software Defined Networking
VMBR: Virtual Machine Bridge
XORP: eXtensible Open Routing Protocol

Acknowledgment

The authors acknowledge the Department of Information Technology, in Kurdistan Regional Government (DIT-KRG) for their continuous support, and providing the facilities needed to implement this project.

References

[1] H.Wang, D. He, and Sh. Tang. "Identity-based proxy-oriented data uploading and remote data integrity checking in public cloud." IEEE Transactions on Information Forensics and Security 11, no. 6 (2016):

1165-1176.
[2] G. Garg, S. Sabharwal, and A. Jain. "BASICS OF CLOUD COMPUTING." (2016).
[3] P.Amaral."Implementation of a Private Cloud." Faculdade de Ciências e Tecnologia da Universidade Nova de Lisboa.(2016).
[4] Margaret Rouse. May 2012. Raised Floor. [online]. [Accessed 5 April 2017]. Available from World Wide Web:<http://searchdatacenter.techtarget.com/definition/raised-floor/>
[5] K. Gupta, R. Beri, V. Behal. "Cloud Computing: A Survey on Cloud Simulation Tools." International Journal for Innovative Research in Science & Technology (IJIRST) 2, no. 11 (2016).
[6] M. Nascimento, Ch. Rothenberg, M. Salvador, and M. Magalhães. "Quagflow: partnering quagga with openflow." In ACM SIGCOMM Computer Communication Review, vol. 40, no. 4, pp. 441-442. ACM, 2010.
[7] V. Eramo, M. Listanti, and A. Cianfrani. "Switching time measurement and optimization issues in Gnu Quagga routing software." In Global Telecommunications Conference, 2005. GLOBECOM'05. IEEE, vol. 2, pp. 6-pp. IEEE, 2005.
[8] M. Banikazemi, D. Olshefski, A. Shaikh, J. Tracey, and G. Wang. "Meridian: an SDN platform for cloud network services." IEEE Communications Magazine 51, no. 2 (2013): 120-127.
[9] S. Revathi, and A. Geetha. "A Survey of Applications and Security Issues in Software Defined Networking." International Journal of Computer Network and Information Security 9, no. 3 (2017): 21.
[10] P. Reddy, V. Vijaya, and R. Lakshmi. "Hypervisors' Guest Isolation Capacity Evaluation in the Private Cloud Using SIAGR Framework." (2015).
[11] Proxmox1. 2017. Download & Documentation Files-Important Downloads. [online]. [Accessed 14 April 2017]. Available from World Wide Web: https://www.proxmox.com/en/downloads
[12] Proxmox2.2017. Proxmox Network Configuration. [online]. [Accessed 25May 2017]. Available from World Wide Web: <https://pve.proxmox.com/wiki/Network_Configuration>
[13] P. Jakma, D. Lamparter. Introduction to the quagga routing suite. IEEE Network. 2014 Mar;28(2):42-8.
[14] Nextcloud.2017. Nextcloud Features. [online]. [Accessed 25May 2017]. Available from World Wide Web: <https://nextcloud.com/secure/>

Authors' Profiles

Rawezh Ziad Kamla is holding master degree in Computer System Engineering in University of Kurdistan- Hewler. He has more than four years of experience in Datacenter Networking, Virtualization and Storage Systems. Currently working in KRG national Datacenter as Network and Systems Engineer.

Tara Yahiya is assistant professor in the university of Kurdistan-Hewler, and holds an (Habilitation a diriger des recherches) which is a qualification for getting full professorship from the university of Paris Sud 11 (France), a PhD in mobile and wireless networks from the university of Paris 6 (Pierre & Marie Curie) and a MSc from the university of Marne-La-Vallee (Paris Est). Her main research interests include cloud computing, mobile communication and QoS.

Nashwan B. Mustafa About seven years of industry experience mainly in Networking, Virtaulization and cloud computing. Holding double master's degree in Information Technology, Network and Network Management. Twice award-winning for achieving excellent academic record in Science and Engineering faculty at Queensland University of Technology, one of the leading universities in Australia. Currently working for eHealth Queensland, Queensland Government in Australia

How to cite this paper: Rawezh Ziad Kamla, Tara Yahiya, Nashwan B. Mustafa,"An Implementation of Software Routing for Building a Private Cloud", International Journal of Computer Network and Information Security(IJCNIS), Vol.10, No.3, pp.1-7, 2018.DOI: 10.5815/ijcnis.2018.03.01