International Journal of Multidisciplinary Research and Growth Evaluation

International Journal of Multidisciplinary Research and Growth Evaluation
ISSN: 2582-7138
Received: 15-09-2021; Accepted: 18-10-2021
www.allmultidisciplinaryjournal.com
Volume 2; Issue 6; November-December 2021; Page No. 341-348

Design and development of multi tenancy security issues in cloud computing

Dr. A Yashwanth Reddy [1*], Dr. M Upendra Kumar [2]
[1] Head of the Department-CSE, Sree Dattha Group of Institutions, Telangana, India
[2] Professor of CSE, MJCET- OU Hyderabad, Telangana, India

Corresponding Author: **Dr. A Yashwanth Reddy**

DOI: https://doi.org/10.54660/.IJMRGE.2021.2.6.341-348

Abstract

With the advent of technology cloud computing has become the next generation of network computing where cloud computing can deliver both software and hardware as on-demand services over the Internet. Cloud computing has enabled small organizations to build web and mobile apps for millions of users by utilizing the concept of "pay-as-you-go" for applications, computing, network and storage resources as on-demand services. These services can be provided to the tenants in different categories: Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS). In order to decrease the costs for the cloud users and increase resource utilization, cloud providers try to share the resources between different organizations (tenants) through a shared environment which is called Multi-Tenancy. Even though multi-tenancy's benefits are tremendous for both cloud providers and users, security and privacy concerns are the primary obstacles to Multi- Tenancy. Since Multi-Tenancy dramatically depends on resource sharing, many experts have suggested different approaches to secure Multi-Tenancy. One of the solutions is resource allocation and isolation techniques. In most cases, resource allocation techniques consider but are not sufficient for security. OpenStack community uses a method to isolate the resources in a Multi-Tenant environment. Even though this method is based on a smart filtering technique to segregate the resources in Compute nodes (the component that the instances are running on it in OpenStack), this method is not flawless. The problem comes up in the Cinder nodes where the resources are not isolated. This failure can be considered as a security concern for a Multi- Tenant environment in OpenStack. In order to solve this problem, this system explores a method to secure Multi- Tenancy for both sides in the Compute node and for backend where Block Storage devices for the instances can be isolated as well and with advanced software technologies and will address the societal changes due to cloud computing.

Keywords: Cloud computing, OpenStack, Multi- Tenancy, Security, Multi-Tenancy Isolation

Introduction

With the advent of cloud computing technology, cloud security has become a big issue in the cloud. Security needs to be considered as one of the most serious concerns for cloud customers such as enterprises and companies. That big issue mostlyis driven by Multi-Tenancy that refers to sharing the resources in cloud computing that leads to integrity and confidentiality risks. In order to conquer the security issues in cloud computing and propose solutions, it is necessary to know more about the architecture of Multi-Tenancy, different attack vectors and attack surfaces [1].

Multi-Tenancy is a way of trying to achieve an economic gain in Cloud Computing by utilizing virtualization and resource sharing. Multi-Tenancy implies different meanings from different points of view and services. In SaaS, Multi-Tenancy implies; when two or more users use the same software or application that is provided by the Cloud Service Provider irrespective of the resources [2, 3]. In PaaS, Multi-Tenancy happens; when a platform or VM is shared between two users (Developers) or more. In IaaS, Multi-Tenancy happens when two or more VMs belong to different users are sharing the same resources (physical machines) [5-8]. In order to have a better concept of multi-tenancy, Figure 1 shows the benefits of multi-tenancy.

Figure 1 clearly shows all the possible benefits of Multi-Tenancy in cloud computing. As tree shows, all these benefits lead to either virtualization or resource sharing or a mix of them. In other words, it can be said "Multi-Tenancy = virtualization + resource sharing". By the way of the example for its benefits, the separation of hardware failures from software failures is possible by virtualization; or resource sharing brings a reduction in energy consumption that finally leads to a reduction of emission gasses and costs. These two inseparable features have a direct impact on VM mobility and over-provisioning [9, 10]. With VM mobility cloud providers are able to reallocate the VMs into clusters and minimize the number of the used servers through high resource utilization. In addition, by over-provisioning cloud service providers are able to sell more than the resource‟s capacity to the users [1]. Multi-Tenancy has pros and cons; even though Multi-Tenancy is imagined as a great chance for the developers, security experts see the Multi-Tenancy as a vulnerability that can be considered as a threat to confidentiality. It is very important for the cloud service providers to try to keep both features, VM mobility and over provisioning and any security solution needs to consider those features.

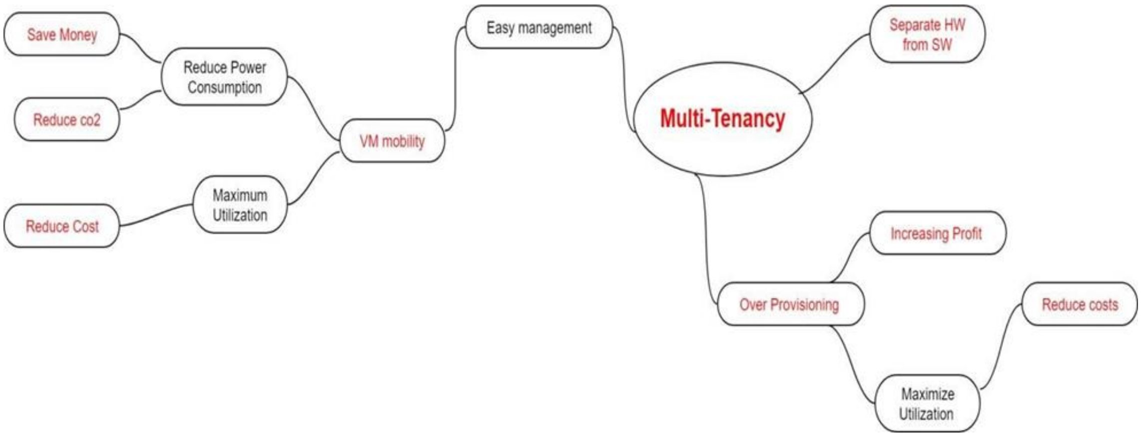


Fig 1: Benefits of Multi-Tenancy tree [1]

Cloud computing can be defined as data center resource sharing through virtualization. Put it
multitasking or multi-processing share common differently, multiple users use the same resource through different services. This system provides broad network access, scalability, and virtualization as the essential characteristics of cloud computing as a service to the customers (Pay-as-you-Go). The biggest challenge through the security in Multi- Tenancy refers to the tradeoff between security and costs. Tim Watson says although one provider may offer a wonderful secure service and another may not, if the latter charges half the price, the majority of organizations will opt for it as they have no real way of telling the difference [2].

Security has a significant role in cloud computing regardless of the service type (IaaS-PaaS- SaaS) that is dedicated to the users. By the way of example in the case of SaaS, cloud service providers provide software or application for users to use. In that case, if a malicious user gets access to the software‟s location, then the attacker has the potential to make inaccessible the resources for all other users. This type of attack can be extended for all other services [11-13]. In the following part, you can see the comparison between different services and the suggested solutions.

A. Multi-Tenancy Security Issues in Cloud Computing

There are many good reasons for Multi- Tenancy to be considered as a security threat in cloud computing; First of all, confidentiality and privacy can be menaced by multi-tenancy. Even though users are separated from each other at the virtual level, the hardware is not isolated and users share the hardware. Multi-Tenancy is relevant to multi-tasking in operating systems whereprocessing resources such as a CPU. Multi-tenancy, like multitasking, directs to a large number of privacy and confidentiality threats [3, 14, 15].

Secondly, Subashini et al suggest using isolation for both in physical layer and application which this segregation needs to be enough intelligent to isolate the data from different users. Subashini believes the Intrusion of data of one host by a malicious user is possible due to the shared feature of multi-tenancy. This intrusion can be done either by hacking through the loopholes in the application or by injecting client code into the SaaS system [4].

Thirdly, Azeez et al describe security as the big challenge in a Multi-Tenant environment, and they tried to build a secure architecture for (SOA framework) where users are enabled to move their applications to a Multi-Tenant environment with almost changes to those applications [5]. To increase security in the cloud, the VR team has suggested a layer of security and depicts virtualization as an issue of the hosting layer because of hosting different virtual machines in the same physical machine [6]. The last but not the least, Cloud Security Alliance (CSA) describes the problem with current security solutions for making a Multi-Tenant environment secure due to the separated location of the cloud environment. They do believe Multi- Tenancy has increased the potential of intrusion in the cloud [7].

The unique feature about Multi-Tenancy from a security perspective is in a Multi- Tenant environment both attackers and victims are on the same side, for example, they are sharing one VM.

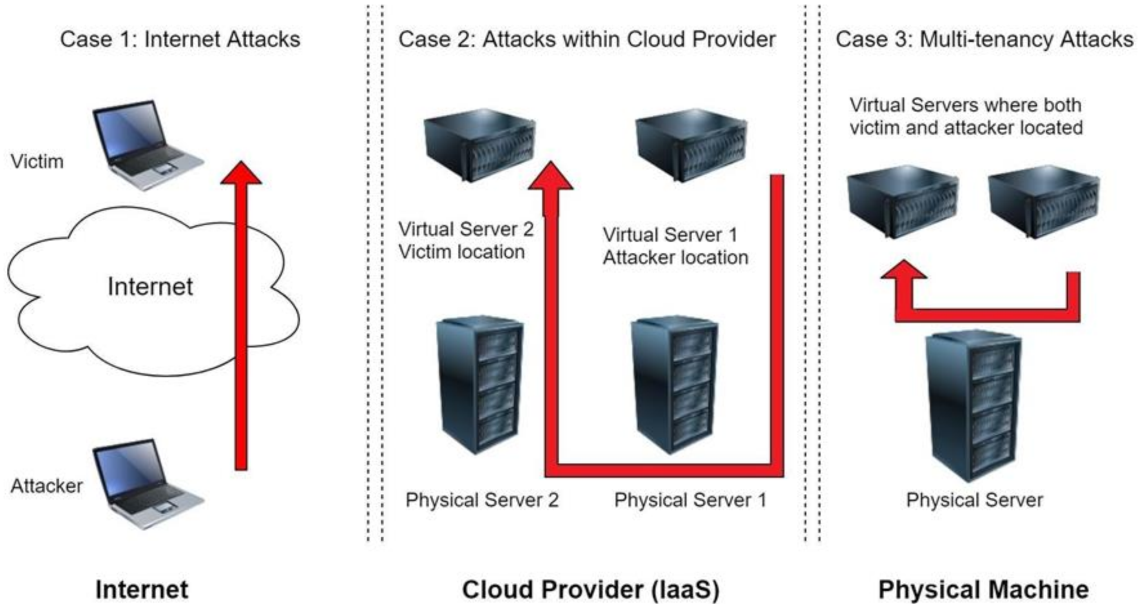


Fig 2: Difference between Multi-Tenancy and other networks [1]

Figure 2 depicts the different cases of attacker and victim and networking between them.As shown by Figure 2, in case one as the simplest case of cyber- attack, both attacker and victim are internet users. In order to prevent such attacks, users can use traditional cyber-security techniques. In case two both attackers and victims are the tenants of a cloud service provider, but they are located on separated physical servers. Due to the usage of a virtualization layer on top of physical servers, there is a need for utilization of virtual network security devices for providing security in the cloud or a way to isolate the shared resources. Case three clearly shows the problem that this project is trying to mention. As can be observed, case three illustrates Multi- Tenancy where both attackers and victims are cloud‟s tenants, and they are sharing the same physical server.

The network communication between the attack‟s VM and victim‟s VM through the physical server imposes one of the most security challenges in cloud computing. Something that makes it hard for security experts to overcome is in this case traffic will not leave the physical machine and it makes it harder to use virtualized network security devices that are used in case two [1].

Cloud computing is known as one of the most popular and widely exploited technologies that gives this opportunity to all small and big enterprises to access system resources via the internet. A wide range of users‟ needs such as data storage, processor power, and software via outside sources with the concepts of pay-per-use is fulfilled by Cloud Computing. It means customers (users) can use the resources as long as they pay for it as a tenant. Cloud computing brings great advantages for customers such as high flexibility and performance without requiring complicated maintenance tasks [27]. In order to take full advantage of cloud computing, Multi- Tenant architecture is designed with the goal of maximizing resource sharing among users. Not only Multi-Tenancy provides full resource utilization for the cloud providers, but also it decreases the cost for the clients. Multi-Tenancy can be described as an architectural structure that allows all resources to be shared by multiple users and sub-users at the same time [27].

Even though Multi-Tenancy brings many advantages both for service providers and customers, it is not flawless and it has its own security issues. Multi- Tenancy security issues are related to integrity and confidentiality risks in sharing resources in cloud computing. When multiple users are sharing the same resources, a malicious user can take the advantage to get access to all other users‟ resources by using some tricks [1].

Network security experts suggest different solutions to overcome Multi-Tenancy security issues. Some suggest using resource allocation techniques due to the nature of Multi-Tenancy [27]. Other security experts, on the other hand, are of the opinion that automated security control can be the best option for cloud providers to protect their network from malicious users. They offer Intrusion Detection System [16-20]. Host-based IDS and Network-based IDS are two types of IDS that can be deployed in a cloud environment. However, for a Multi-Tenant environment, Network-based IDS cannot be useful since it can only address attacks from outsiders, not insiders [21-28]. Host-based IDS can be useful for checking inside attacks where both attackers and victims are located in the same place. In order to solve this problem and avoid imposing security tasks to the customers, this system explores a method for isolating the tenants in a shared environment and shows the importance of the automation of that method where there are lots of users and nodes. According to the National Institute of Standards and Technology (NIST), “Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction” [14]. The cloud model consists of five essential characteristics, three service models, and four deployment models; you can see the general cloud architecture in Figure 3.

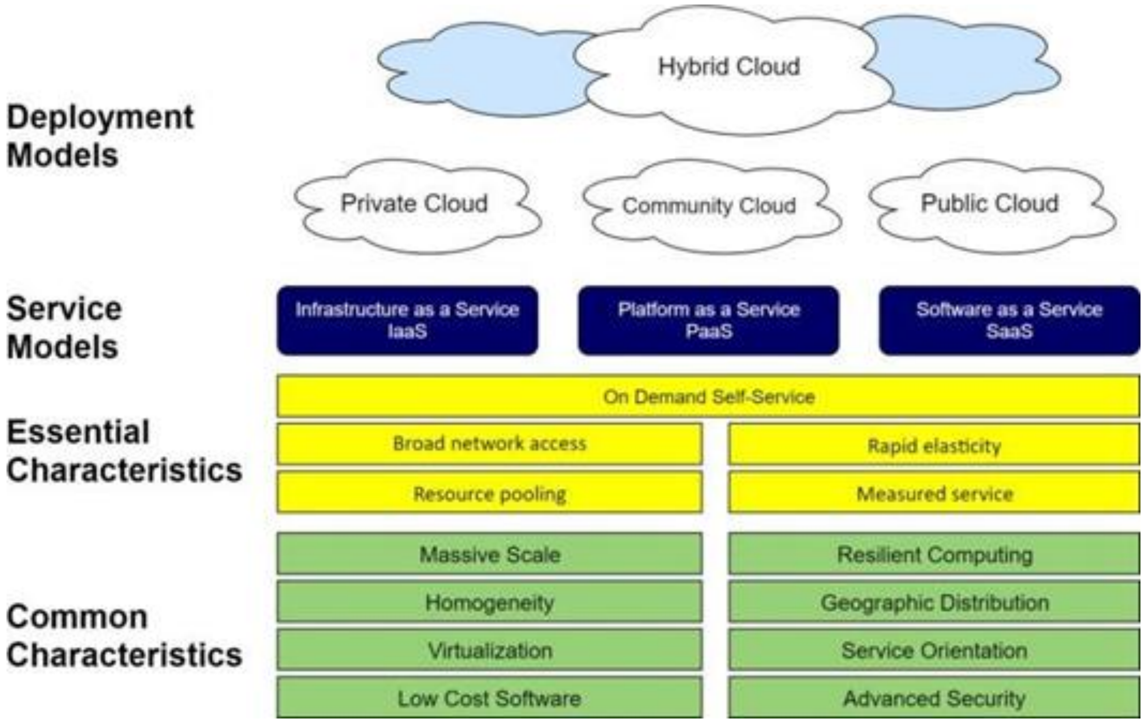


Fig 3: Cloud computing definition [17]

As Figure 4 shows, Cloud Computing offers three types of services:

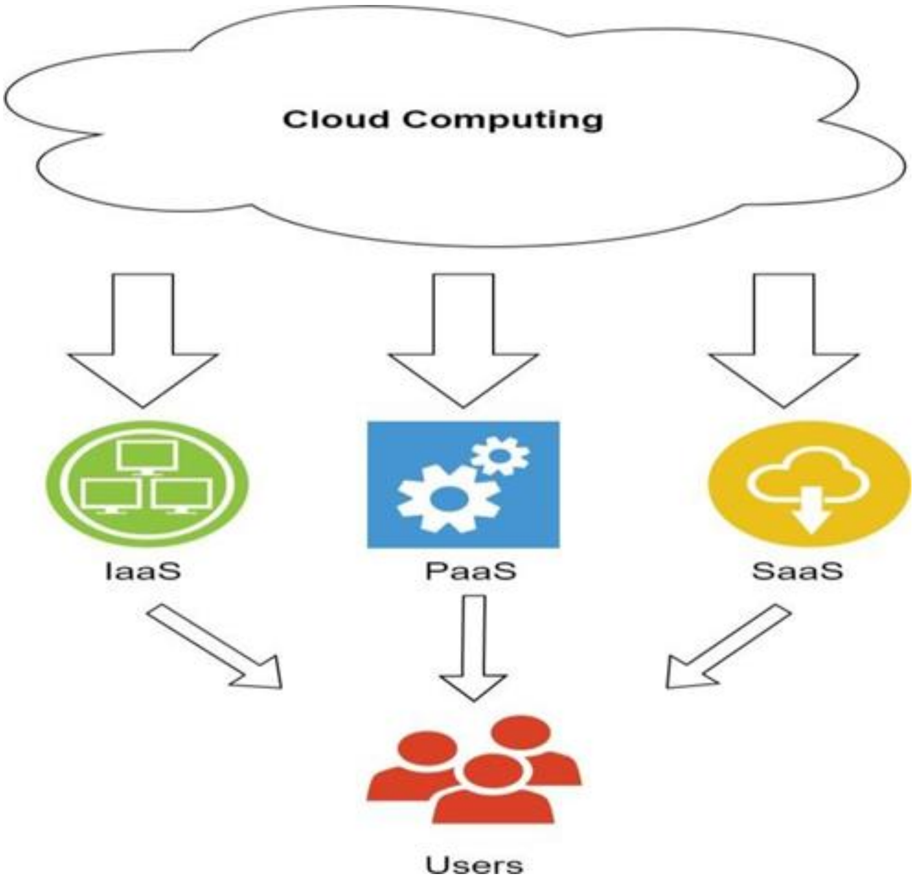


Figure 4: Cloud Computing Services [13]

Software as a Service (SaaS): In SaaS, CSP or Cloud Service Provider is responsible for providing the applications. In other words, an instance of the software will be provided to the users over the internet. In SaaS, the software has been shared between multiple users since it is considered as single-to-many. In SaaS, users are not able to control or monitor the infrastructures, so management and maintenance are centralized and is a duty of the cloud providers. SaaS is a network- based service, and cloud service providers need to make sure the service is up and running all the time. This service ensures the latest version of the software is available to the users where there is no need for users to buy expensive software. The examples of this service are Google Apps, NetSuite [8, 13, 14].

SaaS serves unique characteristics; first of all, users have access to the application via an internet-based interface which is typically run from a web browser. This feature easily provides scalability for adding new users. SaaS supports Multi-Tenancy where each user can share access to the software with the others, so users can opt either increasing the scale for the cost or remain as a single tenant and have greater security and privacy. The SaaS model has a systematic model to support the software instead of maintenance and releasing patches to the subscribers, so users can use the benefits of the latest technological features provided by vendors without any disruption or cost for updating and upgrading.

The traditional method of using software is based on installing software on the user‟s computer locally and buying the license for authorization, but with the SaaS, consumers don‟t need to install the software locally and they just need to pay for the subscription (it also can be free), and software will be accessible via internet. The best example for SaaS is Google Docs, an online word processing application where consumers can access it via the internet and create their own document. In other words, Google provides an application that users can use but not alter directly. It looks like the traditional model, but is used from the internet [15].

Platform as a Service (PaaS): In PaaS, the entire hardware is dedicated to the user as a virtualized environment, and the user can run its own applications on that virtualized environment. PaaS supports scalability where users can ask for more hardware resources, moreover, it supports multiple programming languages that are available to developers (users) in different platforms. In PaaS, users are not allowed to control the resources such as server, network or host operating system, but users are able to manage their own infrastructure by using programming provided tools [8, 13], [14].

As a compare with SaaS, in SaaS, the application that is owned by a cloud provider is ready to use, but PaaS. Provides a platform to create and modify the applications where the PaaS model provides infrastructure as an operational and development environment for the deployment of the applications.

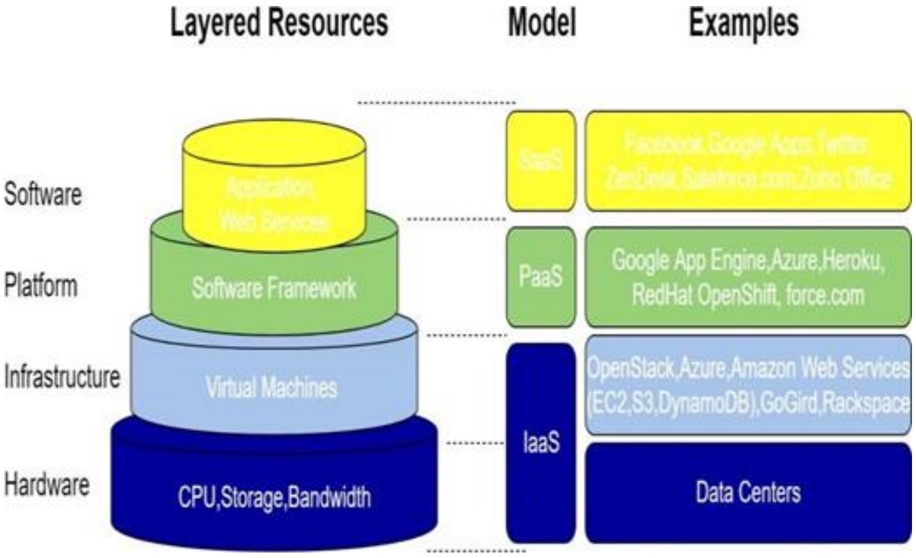

Layered Resources
Model
Examples
Software
Platform
Infrastructure
Hardware
Application,
Web Services
Software Framework
Virtual Machines
CPU,Storage,Bandwidth
SaaS
PaaS
IaaS
Facebook,Google Apps,Twitter,
ZenDesk,Salesforce.com,Zoho Office
Google App Engine,Azure,Heroku,
RedHat OpenShift, force.com
OpenStack,Azure,Amazon Web Services
(EC2,S3,DynamoDB),GoGird,Rackspace
Data Centers

Fig 5: Layered cloud computing models and examples [16]

As an example, as one of the most famous development platforms, Google Apps Engine can be mentioned. Google Apps Engine helps developers by providing different tools like programming languages and APIs. From a security perspective, PaaS leverages the strong facilities of cryptography to secure storage and user‟s confidential data. These facilities provide domain security that protects user‟s data from unauthorized access in a cloud environment; in addition, PaaS provides a privacy feedback process that users will be informed about any risks that endanger their sensitive and confidential data [15].

Infrastructure as a Service (IaaS): In IaaS, a pool of resources that are necessary for running high-performance applications such as servers, routers, storage and switches are dedicated to the user. The user is able to prepare compute, storing and network resourcing and control these resources without managing the infrastructures. In other words, the user has control over the devices and the applications and interacts with the infrastructure, but functions are provisioned by the cloud service provider. In IaaS, an individual user is free to deploy and run every arbitrary software, containing applications or operating systems and control of selected networking components (e.g., host firewalls or host IDS) [8, 13, 14].

IaaS enjoys different standards and architectures from an organization to another, but one single solution is not designed for all. IaaS is a foundation for all delivery models; for example, Kubernetes can be installed on top of it in order to cluster the containers. IaaS is composed of different components: Physical and Virtual Servers, Storage Systems, Network connectivity and Network segmentation (Network blocs and virtual network areas), Network equipment (routers, switches, firewalls, etc.) , DHCP and DNS servers, virtualized platform, billing system, security equipment such as hardware-based or VM-based firewalls, Intrusion Detection and Prevention System. IaaS can be considered as a column for a cloud computing architecture where PaaS and SaaS are built on top of it. This architecture is clearly shown in Figure 5; as can be seen from Figure 5, on the left side the layered architecture is visible, and on the right side, different services and the related examples are shown [15].

In general, what it differs PaaS from IaaS is, in PaaS user has no control over the virtualization instance or network configuration of the server, and from the other side, in PaaS, the user has no control on the hardware that application is running on it or application itself or network configuration [15].

Methodology and Method

The various steps carried out in this research project are shown in order in Figure 6. The project started with gaining basic knowledge of cloud computing and the general advantage and disadvantage of moving to cloud computing. It was followed by learning OpenStack, Kubernetes and Multi-Tenancy in the cloud environment and its security issues. In parallel a case study was done at Ericsson to know more about Ericsson‟s cloud structure, and what has been done to provide Multi- Tenancy isolation for OpenStack.

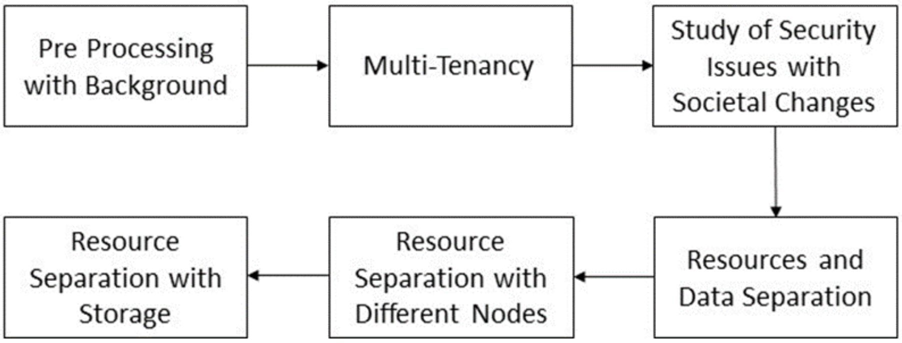
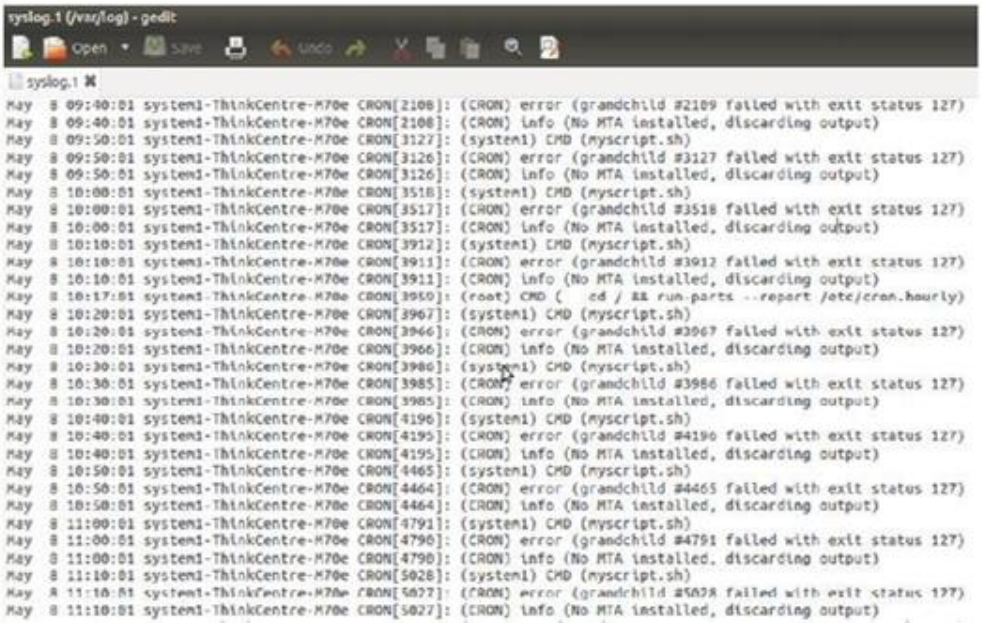

Pre Processing with Background
Multi-Tenancy
Study of Security Issues with Societal Changes
Resources and Data Separation
Resource Separation with Different Nodes
Resource Separation with Storage

Fig 6: Process steps security and privacy of both compute nodes and

After the necessary background knowledge was acquired, the current implementation of Multi- Tenancy in both cloud computing and OpenStack had examined. It was followed by doing the Multi- Tenancy isolation for compute node and in next level was extended to the block storage. The first level of implementation acquired by different resources and the second level of resource isolation was gained by a new method. It leads to the optimal path to increase security in Multi-Tenancy in OpenStack.

Multi-Tenancy Isolation for Compute nodes and Backend

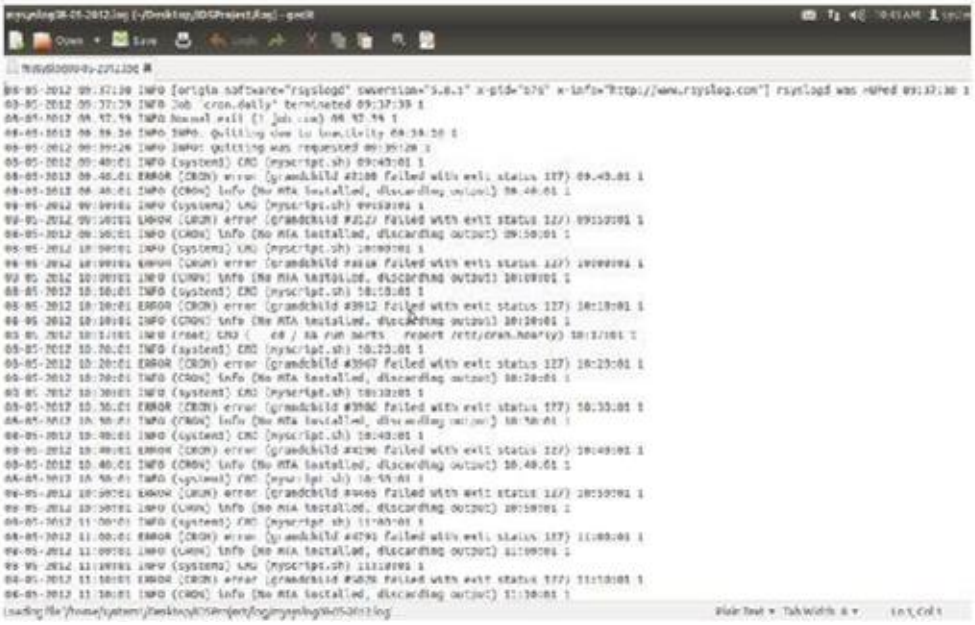


```
May  8 09:40:01 system1-ThinkCentre-M70e CRON[2108]: (CRON) error (grandchild #2109 failed with exit status 127)
May  8 09:40:01 system1-ThinkCentre-M70e CRON[2108]: (CRON) info (No MTA installed, discarding output)
May  8 09:50:01 system1-ThinkCentre-M70e CRON[3127]: (system1) CMD (myscript.sh)
May  8 09:50:01 system1-ThinkCentre-M70e CRON[3126]: (CRON) error (grandchild #3127 failed with exit status 127)
May  8 09:50:01 system1-ThinkCentre-M70e CRON[3126]: (CRON) info (No MTA installed, discarding output)
May  8 10:00:01 system1-ThinkCentre-M70e CRON[3518]: (system1) CMD (myscript.sh)
May  8 10:00:01 system1-ThinkCentre-M70e CRON[3517]: (CRON) error (grandchild #3518 failed with exit status 127)
May  8 10:00:01 system1-ThinkCentre-M70e CRON[3517]: (CRON) info (No MTA installed, discarding output)
May  8 10:10:01 system1-ThinkCentre-M70e CRON[3912]: (system1) CMD (myscript.sh)
May  8 10:10:01 system1-ThinkCentre-M70e CRON[3911]: (CRON) error (grandchild #3912 failed with exit status 127)
May  8 10:10:01 system1-ThinkCentre-M70e CRON[3911]: (CRON) info (No MTA installed, discarding output)
May  8 10:17:01 system1-ThinkCentre-M70e CRON[3950]: (root) CMD (   cd / && run-parts --report /etc/cron.hourly)
May  8 10:20:01 system1-ThinkCentre-M70e CRON[3967]: (system1) CMD (myscript.sh)
May  8 10:20:01 system1-ThinkCentre-M70e CRON[3966]: (CRON) error (grandchild #3967 failed with exit status 127)
May  8 10:20:01 system1-ThinkCentre-M70e CRON[3966]: (CRON) info (No MTA installed, discarding output)
May  8 10:30:01 system1-ThinkCentre-M70e CRON[3986]: (system1) CMD (myscript.sh)
May  8 10:30:01 system1-ThinkCentre-M70e CRON[3985]: (CRON) error (grandchild #3986 failed with exit status 127)
May  8 10:30:01 system1-ThinkCentre-M70e CRON[3985]: (CRON) info (No MTA installed, discarding output)
May  8 10:40:01 system1-ThinkCentre-M70e CRON[4196]: (system1) CMD (myscript.sh)
May  8 10:40:01 system1-ThinkCentre-M70e CRON[4195]: (CRON) error (grandchild #4196 failed with exit status 127)
May  8 10:40:01 system1-ThinkCentre-M70e CRON[4195]: (CRON) info (No MTA installed, discarding output)
May  8 10:50:01 system1-ThinkCentre-M70e CRON[4465]: (system1) CMD (myscript.sh)
May  8 10:50:01 system1-ThinkCentre-M70e CRON[4464]: (CRON) error (grandchild #4465 failed with exit status 127)
May  8 10:50:01 system1-ThinkCentre-M70e CRON[4464]: (CRON) info (No MTA installed, discarding output)
May  8 11:00:01 system1-ThinkCentre-M70e CRON[4791]: (system1) CMD (myscript.sh)
May  8 11:00:01 system1-ThinkCentre-M70e CRON[4790]: (CRON) error (grandchild #4791 failed with exit status 127)
May  8 11:00:01 system1-ThinkCentre-M70e CRON[4790]: (CRON) info (No MTA installed, discarding output)
May  8 11:10:01 system1-ThinkCentre-M70e CRON[5028]: (system1) CMD (myscript.sh)
May  8 11:10:01 system1-ThinkCentre-M70e CRON[5027]: (CRON) error (grandchild #5028 failed with exit status 127)
May  8 11:10:01 system1-ThinkCentre-M70e CRON[5027]: (CRON) info (No MTA installed, discarding output)
```

Fig 7: Syslog before Pre-processing by ALP

Proper full resource isolation improves the

1. Full resource isolation helps to ensure that the quality of service standards is met.
2. Full resource isolation helps to fully isolate the data streams between the tenants through the authentication and authorization of the tenants in the entire OpenStack environment.
3. From a security perspective, full resource isolation allows ensuring a high standard of security by countering different denial-of- service attacks.



```
08-05-2012 09:37:30 INFO [origin software="rsyslogd" swVersion="5.8.1" x-pid="676" x-info="http://www.rsyslog.com"] rsyslogd was HUPed 09:37:30 1
08-05-2012 09:37:39 INFO Job `cron.daily' terminated 09:37:39 1
08-05-2012 09:37:39 INFO Normal exit (1 job run) 09:37:39 1
08-05-2012 09:39:20 INFO INFO: quitting due to inactivity 09:39:20 1
08-05-2012 09:39:20 INFO INFO: quitting was requested 09:39:20 1
08-05-2012 09:40:01 INFO (systemd) CMD (myscript.sh) 09:40:01 1
08-05-2012 09:40:01 ERROR (CRON) error (grandchild #2100 failed with exit status 127) 09:40:01 1
08-05-2012 09:40:01 INFO (CRON) info (No MTA installed, discarding output) 09:40:01 1
08-05-2012 09:50:01 INFO (systemd) CMD (myscript.sh) 09:50:01 1
08-05-2012 09:50:01 ERROR (CRON) error (grandchild #2127 failed with exit status 127) 09:50:01 1
08-05-2012 09:50:01 INFO (CRON) info (No MTA installed, discarding output) 09:50:01 1
08-05-2012 10:00:01 INFO (systemd) CMD (myscript.sh) 10:00:01 1
08-05-2012 10:00:01 ERROR (CRON) error (grandchild #[illegible] failed with exit status 127) 10:00:01 1
08-05-2012 10:00:01 INFO (CRON) info (No MTA installed, discarding output) 10:00:01 1
08-05-2012 10:10:01 INFO (systemd) CMD (myscript.sh) 10:10:01 1
08-05-2012 10:10:01 ERROR (CRON) error (grandchild #3912 failed with exit status 127) 10:10:01 1
08-05-2012 10:10:01 INFO (CRON) info (No MTA installed, discarding output) 10:10:01 1
08-05-2012 10:17:01 INFO (root) CMD (   cd / && run-parts --report /etc/cron.hourly) 10:17:01 1
08-05-2012 10:20:01 INFO (systemd) CMD (myscript.sh) 10:20:01 1
08-05-2012 10:20:01 ERROR (CRON) error (grandchild #3967 failed with exit status 127) 10:20:01 1
08-05-2012 10:20:01 INFO (CRON) info (No MTA installed, discarding output) 10:20:01 1
08-05-2012 10:30:01 INFO (systemd) CMD (myscript.sh) 10:30:01 1
08-05-2012 10:30:01 ERROR (CRON) error (grandchild #3986 failed with exit status 127) 10:30:01 1
08-05-2012 10:30:01 INFO (CRON) info (No MTA installed, discarding output) 10:30:01 1
08-05-2012 10:40:01 INFO (systemd) CMD (myscript.sh) 10:40:01 1
08-05-2012 10:40:01 ERROR (CRON) error (grandchild #4190 failed with exit status 127) 10:40:01 1
08-05-2012 10:40:01 INFO (CRON) info (No MTA installed, discarding output) 10:40:01 1
08-05-2012 10:50:01 INFO (systemd) CMD (myscript.sh) 10:50:01 1
08-05-2012 10:50:01 ERROR (CRON) error (grandchild #4465 failed with exit status 127) 10:50:01 1
08-05-2012 10:50:01 INFO (CRON) info (No MTA installed, discarding output) 10:50:01 1
08-05-2012 11:00:01 INFO (systemd) CMD (myscript.sh) 11:00:01 1
08-05-2012 11:00:01 ERROR (CRON) error (grandchild #4791 failed with exit status 127) 11:00:01 1
08-05-2012 11:00:01 INFO (CRON) info (No MTA installed, discarding output) 11:00:01 1
08-05-2012 11:10:01 INFO (systemd) CMD (myscript.sh) 11:10:01 1
08-05-2012 11:10:01 ERROR (CRON) error (grandchild #[illegible] failed with exit status 127) 11:10:01 1
08-05-2012 11:10:01 INFO (CRON) info (No MTA installed, discarding output) 11:10:01 1
Loading file '/home/system/Desktop/IDSProject/log/mysyslog/08-05-2012.log'
```

Fig 8: mysyslog.log after processing by ALP Subsystem

Table 1: Resource Isolation in Compute node and Block Storage

Solutions/Services	Resource Isolation in Compute Node	ResourceIsolation in Block Storage
Node	Compute	Cinder
Scheduler	Nova-Scheduler	Cinder-Scheduler
Host Groups	Host Aggregates	Multiple Storage
Filter	*Aggregate Instance Extra Specs Filter Aggregate Multi Tenancy Isol*	cinder.scheduler filter_scheduler FilterScheduler
Authentication	Metadata	Volume Type
Authorization	Filter-tenant-id	Cinder-quota
Partition	Flavor	Volume

3. Conclusions

As time goes on, cloud computing has become one of the most popular and widely exploited technologies that provides the chance for all small and big enterprises to access system resources via the internet. The final aim of cloud computing is to provide a pool of resources for all enterprises to have their own data center in a virtual way. This can be considered as the best practice for small companies to be developed through cost reduction. This cost reduction can be extended with Multi- Tenancy where the pool of resources is shared between more than one customer. Multi-Tenancy has both pros and cons; from the positive angle it leads to cost-saving for the customers and increase in resource utilization, but it has its drawbacks in security and privacy. This paper has aimed to overcome security issues with Multi- Tenancy in OpenStack. It uses some suggested solutions by different experts and academic works, and finally, it uses it is own method to fully support the customers via resource isolation. This project uses some features of OpenStack to develop its idea, and implement a new method. For example, Mirantis implements its own filter (PlacementFilter) in nova-scheduler instead of using the default filter of OpenStack, but from the other side, this project extends resource isolation for backend(s) as well that is not implemented by Mirantis. It might be a good idea for researchers who are interested in this topic to work on new filters and compare the results with default filters by OpenStack. Moreover, using Host Aggregate was the work area that this project used however it could be interesting to find a way for using Availability Zone to compare with Host Aggregate.

References

1. AlJahdali, Hussain, et al. "Multi-Tenancy in cloud computing." 2014 IEEE 8th International Symposium on Service-Oriented System Engineering. IEEE, 2014.
2. Khorshed, Md Tanzim, ABM Shawkat Ali, and Saleh A. Wasimi. "A survey on gaps, threat remediation challenges and some thoughts for proactive attack detection in cloud computing." Future Generation computer systems. 2012; 28.6:833-851.
3. Zissis, Dimitrios, and Dimitrios Lekkas. "Addressing cloud computing security issues." Future Generation computer systems 28.3 (2012):583-592.
4. Subashini, Subashini, and Veeraruna Kavitha. "A survey on security issues in service delivery models of cloud computing." Journal of network and computer applications. 2011; 34.1:1-11.
5. Azeez, Afkham, et al. "Multi-Tenant SOA middleware for cloud computing." 2010 IEEE 3rd international conference on cloud computing. IEEE, 2010.
6. Team, Verizon RISK. "2015 data breach investigations report, 2015.
7. Pearson, Siani, and Azzedine Benameur. "Privacy, security and trust issues arising from cloud computing." 2010 IEEE Second International Conference on Cloud Computing Technology and Science. IEEE, 2010.
8. Saripalli, Prasad, and Ben Walters. "Quirc: A quantitative impact and risk assessment framework for cloud security." 2010 IEEE 3rd international conference on cloud computing. Ieee, 2010.
9. Amazon EC2 Abuse Report,

https://security.stackexchange.com/questions/ 195164/amazon-ec2-abuse-report, Jul/2019.

10. Brown, Wayne J., Vince Anderson, and Qing Tan. "Multitenancy-security risks and countermeasures." 2012 15Th International Conference on Network-Based Information Systems. IEEE, 2012.
11. Tsai, Wei-Tek, and Qihong Shao. "Role-based access-control using reference ontology in clouds." 2011 Tenth International Symposium on Autonomous Decentralized Systems. IEEE, 2011.
12. John Rhoton, Cloud Computing Explained Second Edition, Recursive Publishing, 2011
13. Jasti, Amarnath, et al. "Security in Multi- Tenancy cloud." 44th Annual 2010 IEEE International Carnahan Conference on Security Technology. IEEE, 2010.
14. Mell, Peter, and Tim Grance. "The NIST definition of cloud computing, 2011.
15. Goyal, Sumit. "Public vs private vs hybrid vs community-cloud computing: a critical review." International Journal of Computer Network and Information Security. 2014; 6.3:20.
16. Rashid Mijumbi, Joan Serrat, Network Function Virtualization: State-of-the-Art and Research Challenges, https://www.semanticscholar.org/paper/Network-Function-Virtualization%3A-State-of-the-Art-Mijumbi-Serrat/fb38375adf84f909e7784f9736192Aafc3c9f7a9/figure/4 , Jul/2019
17. Baiju Joseph, Cloud testing, https://www.slideshare.net/baijuglad/cloud- testing-16617639, Jul/2019
18. WHAT IS OPENSTACK?, https://www.OpenStack.org/software/, Jul/2019
19. Rosado, Tiago, and Jorge Bernardino. "An overview of OpenStackck architecture." Proceedings of the 18th International Database Engineering & Applications Symposium. ACM, 2014.
20. Kominos, Charalampos Gavriil, Nicolas Seyvet, and Konstantinos Vandikas. "Bare-metal, virtual machines and containers in OpenStack." 2017 20th Conference on Innovations in Clouds, Internet and Networks (ICIN). IEEE, 2017.
21. Docker Containers vs. Virtual Machines, https://www.aquasec.com/wiki/display/ containers/ Docker+ Containers+ vs.+Virtual+Machines , Jul/2019
22. Sehgal, Anuj. "Introduction to OpenStack." Running a Cloud Computing Infrastructure with OpenStack, University of Luxembourg, 2012.
23. Installing Across Multiple Systems for a Multi- node Havana OpenStack Configuration,https://docs.oracle.com/cd/E36784_01/HTml/E54155/installmulti.ht ml#scrolltoc , Jul/2019
24. Sahasrabudhe, Shalmali Suhas, and Shilpa S. Sonawani. "ComparinOpenStackck aVMwarere." 2014 International Conference on Advances in Electronics Computers and Communications. IEEE, 2014.
25. Ashoor, Asmaa Shaker, and Sharad Gore. "Difference between intrusion detection system (IDS) and intrusion prevention system (IPS)." International Conference on Network Security and Applications. Springer, Berlin, Heidelberg, 2011.
26. Singh, Amrit Pal, and Manik Deep Singh. "Analysis of host-based and network-based intrusion detection system." IJ Computer Network and Information Security 8 (2014): 41-47.
27. Karataş, Gözde, et al. "Multi-Tenant architectures in the cloud: a systematic mapping study." 2017 International Artificial Intelligence and Data Processing Symposium (IDAP). IEEE, 2017.
28. Nikolai, Jason, and Yong Wang. "Hypervisor- based cloud intrusion detection system." 2014 International Conference on Computing, Networking and Communications (ICNC). IEEE, 2014.