World Journal of Advanced Research and Reviews
eISSN: 2581-9615 CODEN (USA): WJARAI
Cross Ref DOI: 10.30574/wjarr
Journal homepage: https://wjarr.com/

(RESEARCH ARTICLE)

Check for updates

Multi-tenant data isolation techniques in public clouds assessing the effectiveness of isolation mechanisms

Karthik Venkatesh Ratnam [1] and Rajashekar Reddy Yasani [2,*]

[1] *Southern Methodist University, USA.*
[2] *Murray State University, USA.*

World Journal of Advanced Research and Reviews, 2021, 12(01), 529–539

Publication history: Received on 02 August 2021; revised on 22 October 2021; accepted on 26 October 2021

Article DOI: https://doi.org/10.30574/wjarr.2021.12.1.0402

Abstract

Cloud computing is characterized by its capacity for multiple tenants to share a single cloud infrastructure, which is known as multi-tenancy. Even while it offers cost-efficiency and resource optimization, implementing multi-tenancy raises a number of distinct security challenges. With an emphasis on isolation and access control methods, this paper aims to offer a thorough investigation into the security of multi-tenancy in cloud computing. In this study, the potential dangers that are linked with co-located tenants are investigated, and a variety of measures that can be used to guarantee strong isolation between renters are discussed. Furthermore, the research examines access control methods including Attribute-Based Access Control (ABAC) and Role-Based Access Control (RBAC) to enforce finer-grained access permissions. Both cloud service providers and tenants can enhance cloud security by learning and using effective isolation and access control mechanisms. This will help with multi-tenant setups and overall cloud security.

Keywords: Multi-tenancy security; Cloud computing; Isolation; Access control; Co-located tenants

1 Introduction

The use of cloud storage services, such as Amazon Simple Storage Service (S3), Microsoft Azure, and Google Cloud Storage, has recently exploded in popularity. Public cloud storage solutions have replaced private data centers for many businesses and organizations. Operators of cloud storage services aim to optimize revenues by evaluating the efficacy of resource usage1 [1, 2], 3]. This will allow them to concentrate on latency management and resource management, two aspects of system efficiency. Both resource utilization and system tail delay models are necessary for evaluating the efficiency of resource consumption in multi-tenant cloud storage systems [4]. Firstly, the tail latency issue has proven that the distribution of access delays is inherently random. Although the majority of tenants do not require exact worst-case latency assurances, this method provides a more accurate assessment of user latency requirements by focusing on latency Service Level Objectives (SLOs) at lower percentiles, like 99th and 100 ms. [5].

Contrarily, resource utilization is concerned with the probability of resource usage variations due to task intensity variability. It gives a clear picture of how efficient the resources are. To characterize the overall efficacy of resource utilization, nevertheless, it is necessary to consider both of them. This is because, in cloud storage systems that support numerous tenants, tail latency and resource utilization are two competing concerns.

For instance, cloud facilities frequently run at extremely low utilization to attain low tail latency, which leads to a significant loss of resource use [6]. Equally detrimental to user experience and profitability is increased resource usage at the cost of excessive tail latency [7], [8]. The ideal scenario involves managing tail latency and resource utilization simultaneously; nevertheless, evaluating resource efficiency in multi-tenant cloud storage systems is problematic. The

* Corresponding author: Rajashekar Reddy Yasani

first thing that needs changing is that there isn't a clear mechanism for accurately determining each tenant's tail latency. Factors such as workload burstiness, price, and competition for shared resources (such as networks, CPUs, storage, etc.) are among the many that potentially impact tail latency [9]. Because of these variables, the length of the request arrival/service backlog is more difficult to measure, which exacerbates the measurement challenges.

Secondly, it is a challenging task to correctly assess resource use when dealing with bursty workloads. There is no normal distribution for resource use due to the sporadic nature of workloads [10]. Therefore, optimal resource usage evaluation cannot be supported by conventional approaches that rely on mean value calculations of resource use. As a last and most crucial point, there is currently no reliable system in place to coordinate tail latency with resource use as a whole, making it impossible to differentiate and evaluate the efficacy of resource usage. The relationship between tail delay and resource use has been the subject of multiple studies. While methods for measuring resource utilization are often appropriate for assessing stable workloads, those for tail latency evaluation are limited to network-level tail latency control [5, 11,12]. Finding an optimal trade-off between tail latency and resource consumption in cloud systems has also been the subject of earlier studies [3,13, 14, 15].

No matter how good these methods are at improving resource efficiency under tail latency SLO limits, they can't be utilized to measure how well resources are being used because they all focus on isolation techniques or provisioning strategies. Our proposed framework, SMEA, stands for Stochastic Model-based Effectiveness Analyzer, and it is designed to assess the efficacy of resource consumption in multi-tenant cloud storage systems while taking tail latency and resource utilization into account in two dimensions.

Below is a summary of the main contributions:

- We evaluate the time it takes to access cloud storage services by "treating the computer as a network" [6] and expanding the model for analyzing network latency that is based on Stochastic Network Calculus (SNC) [16]. Network transfer delay, central processing unit (CPU) latency, and disk read/write latency inside storage servers are all components of this latency.
- •We propose a new metric we call resource-productivity to learn about the resource-usage efficiency of multi-tenant cloud storage systems. A connection function that integrates tail latency and resource consumption officially defines it.
- We apply SMEA to cloud storage systems that house several tenants. It starts by applying a Markov-modulated Poisson process (MMPP) to correctly characterize the behavior of workloads with burstinesses. Then, it builds a calculator for resource productivity and two predictors for accurate measurement of tail latency and resource utilization.
- To ensure the accuracy of SMEA, thorough trials are carried out. It is shown that the analytical results agree with the actual experimental results. The relative errors between SMEA and real experiment findings for tail latency, resource consumption, and resource usage effectiveness are about 11%, 7%, and 13%, respectively.

2 Literature review

2.1 Cloud Computing

According to the National Institute of Standards and Technology (NIST) [17], cloud computing enables users to tap into a shared pool of flexible computing resources like networks, servers, storage, apps, and services. These resources can be easily assembled and released with little management work or communication with service providers. You can see the overall cloud architecture in Figure 1. Along with its three service types and four deployment techniques, the cloud model encompasses five fundamental qualities.

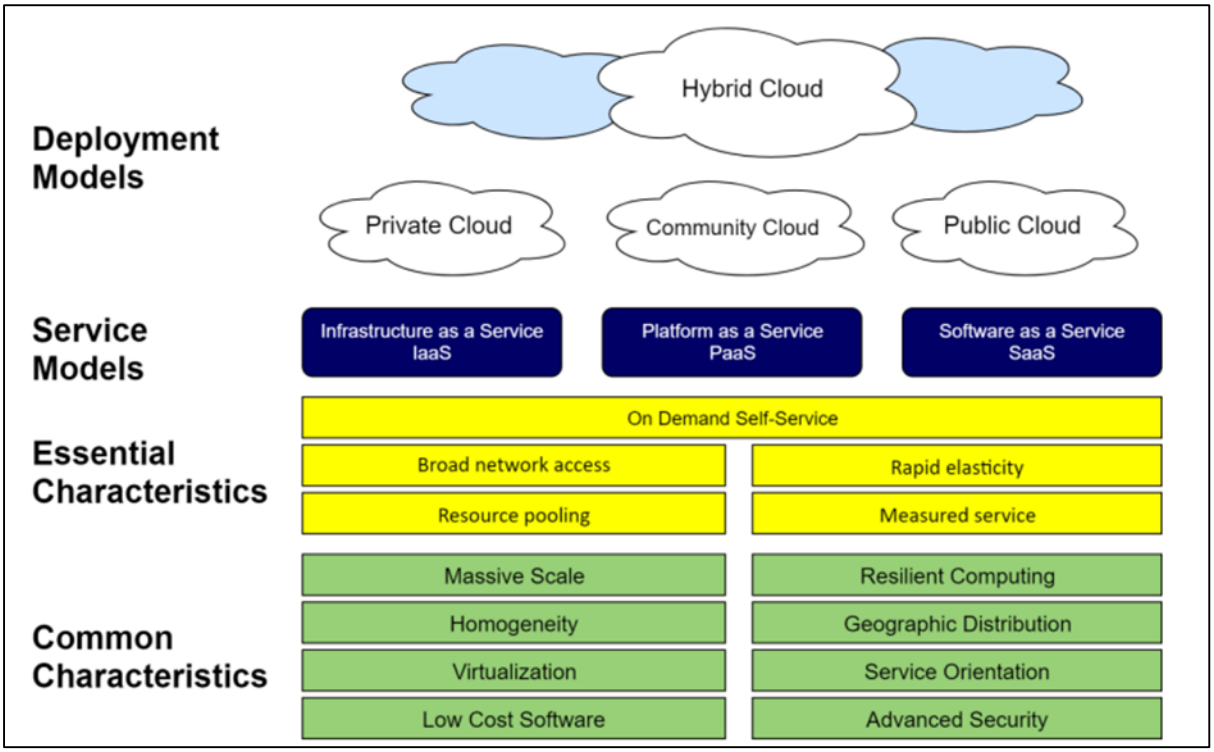

Figure 1 Cloud computing definition

2.2 Essential characteristics

When using on-demand self-service, users can automate the preparation of computing features and capabilities in response to a one-way request [18]. The complete network can be accessed using common procedures that are made available through user platforms like computers, mobiles, and tablets, and the features are ready to use. Pooling resources: In a MultiTenant architecture, various physical and virtual resources can be shared amongst several users based on their need. Even though users may be able to learn about the physical location of resources (like storage, memory, network bandwidth, and processors) at a higher level (like the country or data center)—they still do not have control over or knowledge of the specifics of where these resources are located or the equipment that makes them up [19]. The released capabilities need to be automated and flexible so that the demands may be scaled rapidly. That is to say, the capabilities ought to be accessible in any amount and at any moment [20]. Service measurement: Depending on the service, cloud providers are in charge of controlling and optimizing resources using metrics like pay-per-use or charge-per-use. Providers and users alike can keep tabs on resource consumption and report back on reported service utilization [21].

2.3 Deployment models

A public cloud is one in which any entity, including businesses, educational institutions, the government, or a combination of these, owns, operates, and controls the underlying cloud infrastructure in order to make it available to the general public. To rephrase, a third party provides the services in public clouds, which are open to the public. The data that users create and upload to the server is stored by third parties [22]. A "public cloud" is a type of cloud computing in which users only pay for the resources they really utilize.

Users can scale their consumption as needed and don't have to spend a fortune on hardware. Scalability and constant availability are the two main benefits of public cloud. When discussing its drawbacks, privacy and security should be taken into account. One issue with public clouds is reliability, which comes from not knowing where data is stored, how it is stored, or how easy it is to retrieve. Companies should research their potential public cloud provider's ability to handle their specific privacy and security needs. Public clouds include services like Amazon Web Services (AWS), Microsoft Azure, and Google Cloud Platform. In a "private cloud" setup, one company or organization makes its resources available to several consumers rather than the public cloud. The owner or administrator of a private cloud can be an organization, an outside entity, or even both [23]. The services offered by a private cloud, which is located in the data center of a certain organization or firm, are accessible only to users within that organization or firm. Increased privacy and security, less complexity, and resource-related cost reductions are just a few of the benefits that private

clouds offer over public clouds. They also bring resources that aren't being exploited to their full potential, which can be shared with partners to make the most of them.

The infrastructure and computing resources can be better managed with private cloud computing. The fact that a business must set aside a sizable sum to purchase hardware, software, and other components is the biggest drawback of the private cloud. Community cloud: This type of cloud service is designed for groups of users within a business or organization who share common concerns, like security regulations or norms. This cloud model allows for a combination of ownership, organization, and control by many organizations or even a third party [24]. A community cloud combines public and private cloud features; it mimics the appearance of a private cloud while actually allowing numerous companies to share the same infrastructure and computational resources while adhering to the same privacy and security protocols. The architecture offers less cost than the private cloud because the devoted funding is distributed among several organs.

By entrusting the management and control duties to a reliable third party, users of community clouds can reduce the complexity of their cloud infrastructure and save time on maintenance. Community clouds, on the other hand, have higher prices and less security capabilities than public clouds because users share bandwidth and storage. Hybrid cloud: In a hybrid cloud, elements of both public and private clouds work together to provide users with personalized service while facilitating the transfer of data and applications. Simply said, a hybrid cloud consists of multiple independent clouds that are linked via a common technology or standard that permits the transfer of data and applications between them [25]."A vendor has a private cloud and forms a partnership with a public cloud provider or a public cloud provider forms a partnership with a vendor that provides private cloud platforms" is one way to describe a hybrid cloud. In a hybrid cloud, some resources can be housed inside the cloud while others can be hosted outside.

3 Methodology and method

This section provides an overview of the research approach taken for this project and we cover the study technique, we discuss the data gathering plan. The experimental design of the study is presented in below Section and the validity and dependability of the data are described.

3.1 Research process

Figure 2 shows the sequential order of the numerous steps that were carried out in this study effort. Learning the fundamentals of cloud computing as well as the pros and cons of making the switch was the first step in the endeavor. After that, we moved on to studying cloud security challenges related to OpenStack, Kubernetes, and multi-tenancy. Concurrently, Ericsson's cloud architecture and the processes used to implement Multi-Tenancy isolation for OpenStack were studied through a case study.

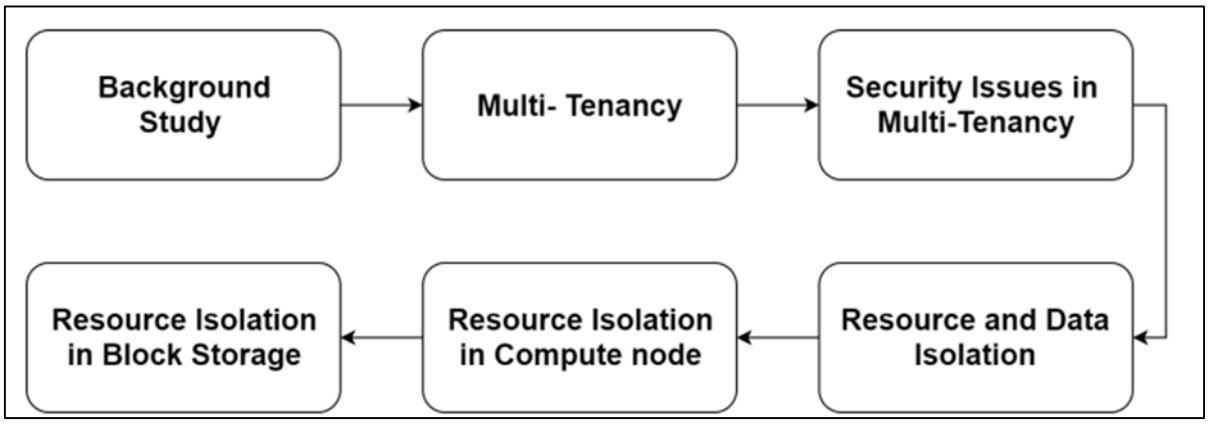

Figure 2 Steps in conducting research

The next stage, after gathering the necessary background knowledge, was to examine the current state of Multi-Tenancy in cloud computing and OpenStack. After the compute node was multi-tenant isolated, it was extended to the block storage level. We successfully isolated our resources on two levels: the first by utilizing various resources, and the second by employing a novel approach. Finding the best way to strengthen OpenStack's Multi-Tenancy security is what it's all about.

3.2 Data Collection

Using an inductive approach, this research gathered data that would serve as facts and best practices for bolstering Multi-Tenancy security using the isolation strategy. Basic ideas of OpenStack features from various suppliers or experts were used to finish the work, ensuring that all the data acquired was suitable for the review.

3.3 Experimental Design

In this section, we will go over the steps used to conduct the environmental experiments for this project. Both the software and the hardware that is used are part of this.

3.3.1 Hardware Platform

One OPNFV Linux Foundation x86_64 physical system with two separate CPU sockets and eighty-eight CPU cores is used for this project. It was an HP Enterprise CPU. The hard drive is 894 GB SSD and the RAM is 503 G.

3.3.2 Software Platform

Ubuntu 16.04.5 LTS (Xenial Xerus) served as the operating system for all of the project's trials. The current version of OpenStack is Kilo. The OpenStack services were run in this cloud environment using LXC containers. The scripts on OpenStack are written in Python, and this project also utilized an Ansible Playbook to automate the process, which is also written in Python.

3.4 Reliability and Validity

Experts from Ericsson or OPNFV who worked with OpenStack were interviewed and discussed for the case study, and the data was compared with that from the literature review.

3.4.1 Reliability

Concerning the information source for this project, it should be noted that the data pertaining to future plans can be verified up until the time the data was collected.

3.4.2 Validity

There is a chance that changing surroundings would cause different outcomes or possibly introduce errors if the identical procedures (in Chapter 4) were to be repeated. In addition, the project's foundation was a qualitative approach, which meant that other researchers may use other methods to achieve the same goal.

4 Multi-tenancy isolation in OpenStack

Managing multi-tenancy in OpenStack is a crucial aspect of cloud computing, ensuring that different tenants can safely coexist on the same physical infrastructure without interference or security breaches. In this detailed exploration, we will delve into the concept of multi-tenancy in OpenStack, its security implications, and the best practices for managing it effectively.

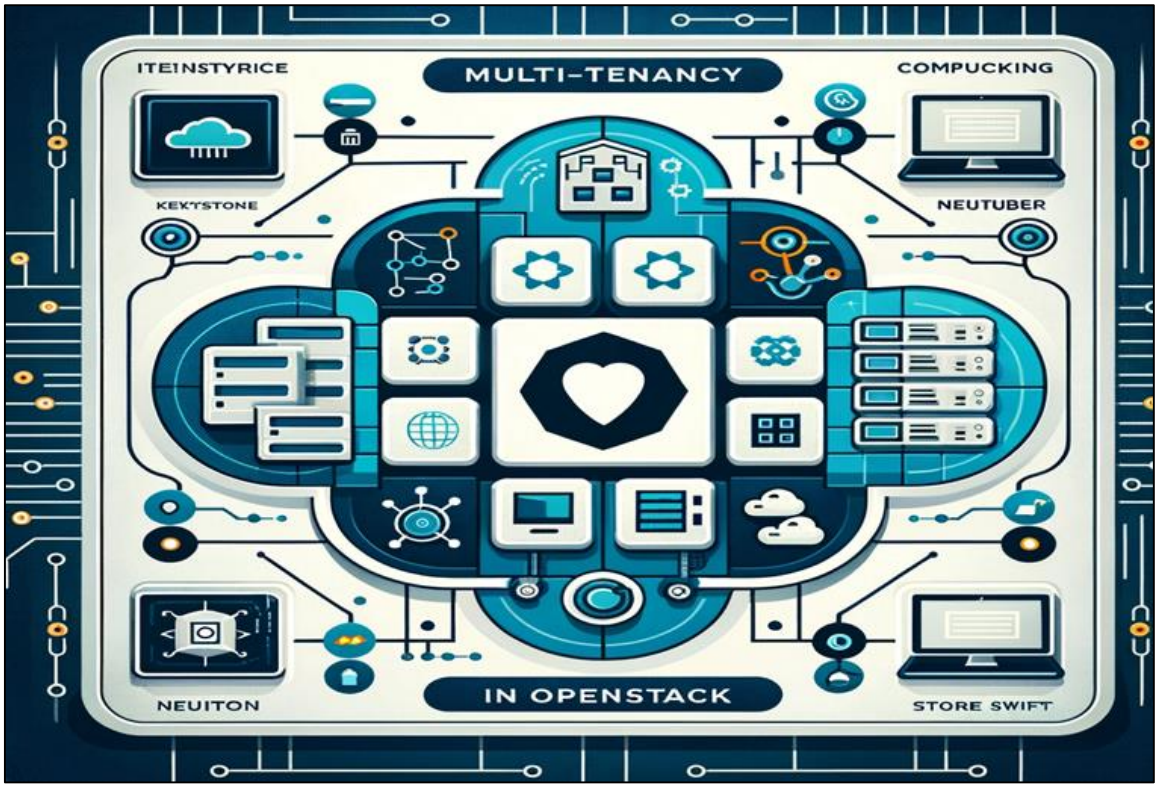

Figure 3 Multi-tenancy isolation in open stack

4.1 Understanding Multi-Tenancy in OpenStack

In information technology, "multi-tenancy" means that a single instance of software can support numerous users, groups, or organizations (tenants), with the data of each tenant kept separate and unobservable by the others. With OpenStack's multi-tenancy feature, a cloud provider can make available the same set of resources and services to numerous customers while keeping the same set of hardware and middleware components. By distributing the workload among its many components, OpenStack is able to support multiple tenants in a single cloud:

- Keystone: Serves as OpenStack's central point for authentication and permission by providing identity services. The system ensures that access controls are strictly implemented by managing users and their permissions.
- Nova: Oversees the progress of computing instances during their lifetime. In order to facilitate multi-tenancy, Nova uses host and project-specific security groups to partition user instances.
- Neutron: Provides OpenStack with networking services and manages the creation of tenant-isolated network resources such as subnets, routers, and IPs.
- Cinder and Swift: Make available object storage and block storage services with tools for secure per-tenant management of storage resources.

4.2 Security Implications of Multi-Tenancy

Multiple security issues arise from the fact that resources are shared in multi-tenant setups. These issues mainly include data breaches, leaks of resources, and attacks that target multiple tenants at once. A multi-tenant OpenStack environment presents several important security challenges: Isolation Failed: If this fails, one tenant could be able to access another tenant's data or resources due to setup errors or security flaws.

- Resource Contention: A "noisy neighbor" problem exists when one tenant's excessive demands interfere with the capacity of other tenants to work or access necessary supplies.
- Tenant Privilege Escalation: A tenant may be able to obtain illegal access to other tenants' data or more system functionality if permissions are not properly configured.

4.3 Best Practices for Managing Multi-Tenancy in OpenStack

Policy, management techniques, and technical controls can help administrators manage these risks and make a multi-tenant OpenStack infrastructure more secure: Robust Authorization and Authentication: Incorporate strong authentication methods (such as two-factor authentication) and establish transparent, minimum permissions for every user role using Keystone.

- Network Segmentation and Firewalling: Neutron allows you to separate networks for various tenants. Protect these networks from possible interference from other tenants by limiting traffic to and from them using firewalls and security groups.
- Regular Audits and Compliance Checks: Keep an eye on the OpenStack environment's settings, logs, and permissions. To make sure policies are implemented and comply with regulations, tools like OpenStack's Security Group and Neutron's firewall as a service (FWaaS) are useful.
- Resource Allocation and Monitoring: To reduce the likelihood of resource depletion, set quotas and restrictions on how much a tenant can use. Abnormal activity or surges in resource utilization can be flagged by monitoring systems as possible security risks.
- Encryption: Particularly when data is being transferred between different tenant environments, encrypt it both while it is in transit and at rest. Secure Sockets Layer (SSL/TLS) and the built-in encryption capabilities of Cinder and Swift are all part of this.
- Vulnerability Management: Keep OpenStack components up-to-date and patched to prevent known vulnerabilities. Critically important is the implementation of a proactive program for managing vulnerabilities, which should include vulnerability scanning tools.
- Incident Response and Forensic Capabilities: Create a strategy for dealing with incidents that is ideal for buildings with several tenants. As part of this, procedures should be put in place to identify affected systems and isolate them so that they do not affect other tenants.

4.4 OpenStack Tenancy and Resource Isolation

Cloud administrators can manage rights in VMware Integrated OpenStack using project definitions, users, and groups. Datapath isolation is also possible across networking, storage, and computing.

One important feature of VMware Integrated OpenStack is that it allows numerous users to share the VMware SDDC environment while yet assuring total separation. In VIO, renters can share virtual resources while yet enjoying full isolation when necessary.

There are other ways in which VIO isolates tenants, such as:

- Authorization and authentication for the Keystone API and Dashboard.
- Glance Image Catalog private photographs.
- VRF allows for the separation of network traffic for different user groups.
- Isolate your computer.

In order to create a multitenant environment for VNF deployment, the accompanying figure shows how to use a fully integrated VMware Integrated OpenStack Compute Zone, NSX segments, a tenant virtual data center (VDC), and Tier-1 gateways.

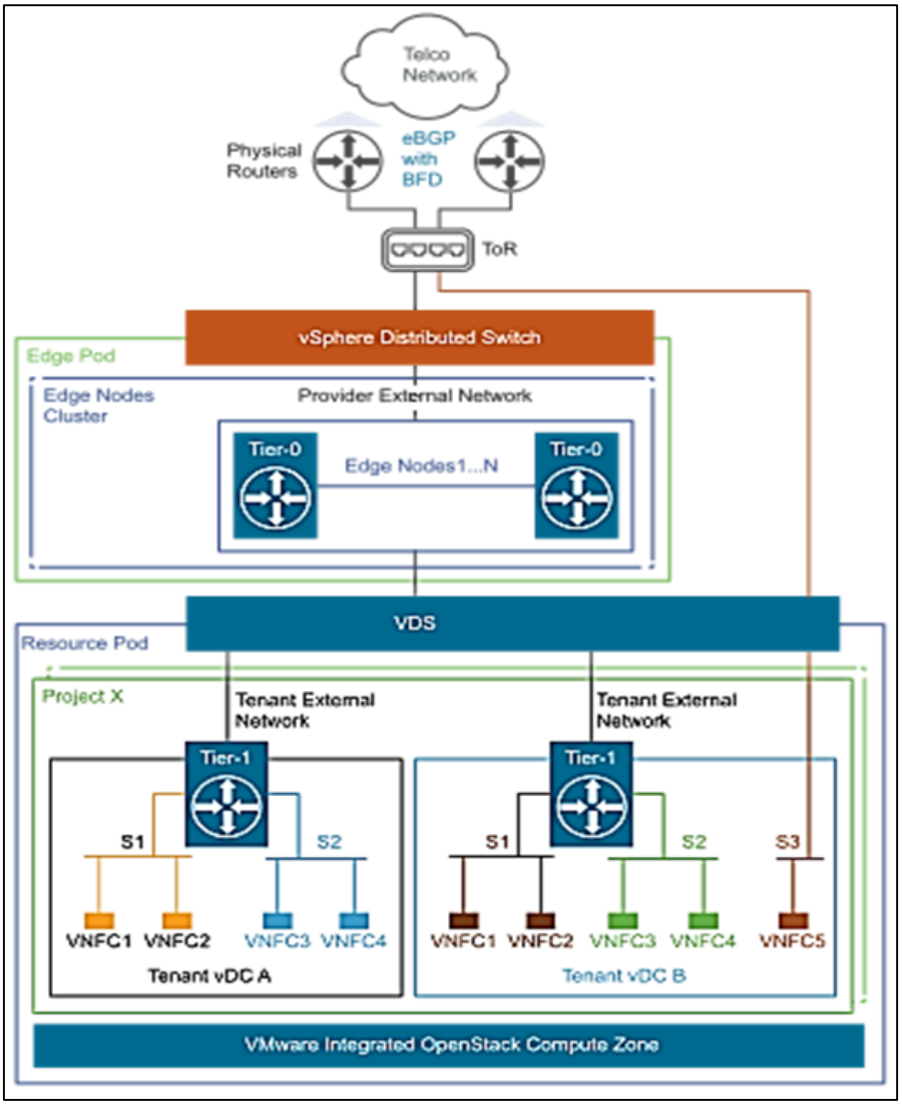

Figure 4 Private Cloud with Multiple Tenants

Keystone user isolation is described in this section. The design of OpenStack services, including resource isolation, is detailed in the sections that follow.

4.5 Identity

One way to manage authentication, authorization, and a catalog of services is with the OpenStack Keystone Identity service. The identity provider will provide a user with an authorization token once Keystone authentication is complete. According to the authorization rules (policy.json), an end-user can access additional OpenStack services using the authorization token. You have the option of using an internal database for the Identity service or integrating it with third-party user management systems.

Using the enterprise Active Directory (AD) Lightweight Directory Authentication Protocol (LDAP) services for authentication is an alternative to keeping an independent local keystone user database that VIO provides. It is also feasible to integrate federated identities with external identity providers through SAML2 or OIDC, in addition to LDAP.

4.6 OpenStack Projects

Telco Cloud Infrastructure tenants are analogous to OpenStack projects. A project is a management container for deploying and overseeing telecom workloads. Users and groups are assigned to an OpenStack project. You can limit resources like virtual CPUs, RAM, instances, networks, ports, subnets, and more in OpenStack using a construct called Quotas. You can set quotas in OpenStack at the project level. There is no correlation between the quota values and the

capability of the current infrastructure. The configuration of quotas on OpenStack is platform-agnostic. Services provided by OpenStack, including Nova, Neutron, Cinder, and Glance, are subject to quota enforcement.

Domains are a way to classify projects, users, and groups at a high level. Delegating control of OpenStack resources is possible through the usage of domains. With the right authorization, a user can belong to more than one domain. Default is the default domain that Keystone offers. Users and groups can be adequately isolated with just one Domain for the majority of telecom use cases.

4.7 Tenant VDC

With a Tenant VDC, many telco workloads can share the same virtual data center infrastructure, but with their own dedicated compute nodes and service level agreements (SLAs). Tenant virtual data centers (VDCs) offer resource assurances to tenants and enable reservations and restrictions on a specific class of resources (such as compute nodes) depending on the current capacity, in contrast to project quotas that limit OpenStack resources generally. To avoid noisy neighbor scenarios entirely in a multitenant environment, the vSphere platform limits and guarantees resources assigned to a tenant on a given compute node. We back three distinct policy styles:

- Pay-as-you-go
- Reservation Pool
- Allocation Pool

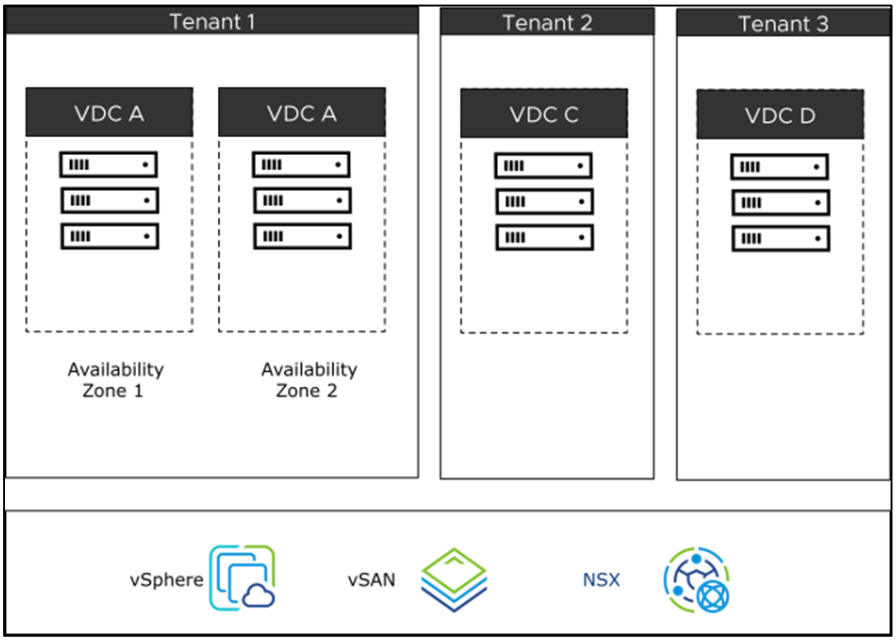

Figure 5 Tenant VDC for VNF Resource Allocation

4.8 Tenancy Design Recommendations

Here's a well-organized table summarizing the design recommendations for VMware Integrated OpenStack:

Table 1 The design recommendations, their justifications, and implications

Design Recommendation	Design Justification	Design Implication
The VIO Identity service can be integrated with third-party user databases like Active Directory.	Makes it possible to add or remove users worldwide without having to go to each deployment individually, and it centralizes the management of users.	Must have extensive knowledge of the company's user directory in order to onboard just necessary individuals into VIO.

When external users need management access, utilize the OpenStack domain. Otherwise, don't use it.	The absence of domain-level usage quota options and the complexity of OpenStack domains make user administration a nightmare.	None
To ensure tenants have access to services and prevent situations when neighbors are too noisy, use Tenant VDCs.	Boosts SLA per VNF and ensures the availability of virtualized infrastructure resources.	Make a fresh default VDC in Tenant VDC that does not have a reservation and assign all of the default flavors to it.
Save the OpenStack policy updates for later.Please let me know if there are any other ways to accomplish the same result using JSON.	Ensures that upgrades will work without a hitch, since non-default changes may not be compatible with previous versions.	None

This table should help in understanding the design recommendations, their justifications, and implications more clearly.

5 Conclusion

Implementing a well-rounded strategy for managing OpenStack multi-tenancy calls for a combination of stringent security measures, proactive monitoring, and quick incident management. In order to get the most of OpenStack's capabilities while keeping all tenants' data and resources safe, businesses should follow these best practices. Maintaining trust in multi-tenant situations and keeping up with new threats in the cloud will need constantly improving and adapting security measures. Cloud computing's multi-tenancy security is, to sum up, an essential component that calls for meticulous planning and strong execution. In shared cloud environments, numerous tenants share the same infrastructure. This article reviews the literature on the topic and discusses the pros and cons of these settings. To protect data, applications, and resources in a multitenant environment, the results stress the significance of strong isolation and access control measures. Security measures like virtualization, containerization, and resource partitioning are essential in a multi-tenant environment to avoid data breaches and unauthorized access. These safeguards allow cloud providers to isolate their tenants' environments logically, which in turn reduces the likelihood of data leakage and privilege escalation.

Compliance with ethical standards

Disclosure of conflict of interest

No conflict of interest to be disclosed.

References

[1] Amazon Athena. (2020). Retrieved from https://aws.amazon.com/athena/

[2] Amazon Firecracker. (2020). Retrieved from https://aws.amazon.com/about-aws/whats-new/2018/11/firecracker-lightweight-virtualization-for-serverless-computing/

[3] Amazon RDS Multi-AZ. (2020). Retrieved from https://aws.amazon.com/rds/features/multi-az/

[4] AWS Redshift. (2020). Retrieved from https://aws.amazon.com/redshift/

[5] Amazon Aurora Serverless. (2020). Retrieved from https://aws.amazon.com/rds/aurora/serverless/

[6] Apache Hadoop. (2020). Retrieved from http://hadoop.apache.org

[7] A Technical Overview of Azure Cosmos DB. (2020). Retrieved from https://azure.microsoft.com/en-us/blog/a-technical-overview-of-azure-cosmos-db/

[8] Azure SQL DB Automatic Tuning. (2020). Retrieved from https://docs.microsoft.com/en-us/sql/relational-databases/automatic-tuning/automatic-tuning

[9] Antonopoulos, P., A. Budovski, C. Diaconu, A. Hernandez Saenz, J. Hu, H. Kodavalla, D. Kossmann, S. Lingam, U. F. Minhas, N. Prakash, et al. (2019). "Socrates: The New SQL Server in the Cloud". In: Proceedings of the 2019 International Conference on Management of Data. ACM. 1743–1756.

[10] Appuswamy, R., G. Graefe, R. Borovica-Gajic, and A. Ailamaki. (2019). "The five-minute rule 30 years later and its impact on the storage hierarchy". Communications of the ACM. 62(11): 114–120.

[11] Bacon, D. F., N. Bales, N. Bruno, B. F. Cooper, A. Dickinson, A. Fikes, C. Fraser, A. Gubarev, M. Joshi, E. Kogan, et al. (2017). "Spanner: Becoming a SQL system". In: Proceedings of the 2017 ACM International Conference on Management of Data. ACM. 331–343.

[12] Banerjee, I., F. Guo, K. Tati, and R. Venkatasubramanian. (2013). "Memory overcommitment in the ESX server". VMware Technical Journal. 2(1): 2–12.

[13] Barham, P., R. Isaacs, R. Mortier, and D. Narayanan. (2003). "Magpie: Online Modelling and Performance-aware Systems." In: HotOS. 85–90.

[14] Burns, B., B. Grant, D. Oppenheimer, E. Brewer, and J. Wilkes. (2016). "Borg, Omega, and Kubernetes". Queue. 14(1): 10.

[15] Chang, F., J. Dean, S. Ghemawat, W. C. Hsieh, D. A. Wallach, M. Burrows, T. Chandra, A. Fikes, and R. E. Gruber. (2008). "Bigtable: A distributed storage system for structured data". ACM Transactions on Computer Systems (TOCS). 26(2): 1–26.

[16] Google BigQuery. (2020). Retrieved from https://cloud.google.com/bigquery

[17] MarketResearch. (2019). https://www.marketsandmarkets.com/MarketReports/cloud-database-as-a-service-dbaas-market-1112.html.

[18] Corbett, J. C., J. Dean, M. Epstein, A. Fikes, C. Frost, J. J. Furman, S. Ghemawat, A. Gubarev, C. Heiser, P. Hochschild, et al. (2013). "Spanner: Google's globally distributed database". ACM Transactions on Computer Systems (TOCS). 31(3): 8.

[19] Curino, C., D. E. Difallah, C. Douglas, S. Krishnan, R. Ramakrishnan, and S. Rao. (2014). "Reservation-based scheduling: If you're late don't blame us!" In: Proceedings of the ACM Symposium on Cloud Computing. ACM. 1–14.

[20] Curino, C., E. P. Jones, S. Madden, and H. Balakrishnan. (2011). "Workload-aware database monitoring and consolidation". In: Proceedings of the 2011 ACM SIGMOD International Conference on Management of data. ACM. 313–324.

[21] AlJahdali, Hussain, et al. "Multi-Tenancy in cloud computing." 2014 IEEE 8th International Symposium on Service-Oriented System Engineering. IEEE, 2014.

[22] Khor shed, 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 28.6 (2012): 833-851.

[23] Zissis, Dimitrios, and Dimitrios Lekkas. "Addressing cloud computing security issues." Future Generation computer systems 28.3 (2012): 583-592.

[24] Subashini, Subashini, and Veeraruna Kavitha. "A survey on security issues in service delivery models of cloud computing." Journal of network and computer applications 34.1 (2011): 1-11.

[25] Azeez, Afkham, et al. "Multi-Tenant SOA middleware for cloud computing." 2010 IEEE 3rd international conference on cloud computing. IEEE, 2010.