Universiti Teknologi MARA
Cawangan Perlis
Jurnal Intelek

Volume 14 Issue 2 (December 2019)

A Deployment Scenario: A Taxonomy Mapping and Keyword Searching for the Application Relation.

Sharipah Setapa [1], Shahrol Hisham Baharom [1], Luke Jing Yuan[1]

[1]Advanced Computing Lab, Mimos Berhad, Kuala Lumpur, Malaysia

sharipah@mimos.my, shahrol.baharom@mimos.my, jyluke@mimos.my
Received Date: 18 September 2019
Accepted Date: 9 October 2019

ABSTRACT

Upgrading and patching is a method to strengthen the host and virtualisation devices from malware. In the deployment to different entity's clients, different scenarios are faced to support their business process. A host and virtual machine (VM) is dependent on each other to provide high efficiency. A good design of virtual resources in a physical host can maintain the host's efficiency. If the host is to be upgraded or patched, then the relationship with VM needs to be explored to avoid missing or malfunctioning of certain application on the host or VM. A relationship how to check whether the application scenario is working as expected is not being derived. This paper will discuss scenarios that we face and how it can be converted into taxonomy to provide a strategic approach when upgrading a specific item. With that, an analogy can be based on how the application scenario can be established as a model and converted into taxonomy for troubleshooting when execution is facing an error.

Keywords: Design scenario, strategic way, taxonomy, keyword, application

INTRODUCTION

Reviewing periodically a resource's infrastructure is a step to check any holes in the system. Why does it need a periodical upgrade and patch? Let's take some analogy by imagining every single item surrounding us in our environment, for example cars, infrastructure buildings, gardens, and kindergartens. The longer the item is, the more it needs to be polished or observed. In the environment, especially for a network that consists of a computer, virtual machine, gateway and application, full cycle connectivity needs to be reviewed in certain period of time. A scanning for network penetration to get information for any vulnerability is necessary to avoid a black hole in the network. For example, software or application can become obsolete, and then new software will be updated through a patch. Initially, network vulnerability protection is done through routine patching when the user is alerted at a certain period time (Zhou et al., 2010). Virtualisation layer in client infrastructure can cause complicity for patching. Physical host and device such as virtual machine (VM) need to be checked and balanced for the patch to work accordingly. This is one of the ways to reduce network vulnerability. Each client use software which is different specification hardware and software which can cause deployment of patch is not smooth as be intended.

The number of companies utilising virtualisation are increasing. The number of users utilising and creating the VM for various purposes is also increasing. Therefore, managing the VM and reviewing the activity of VM often is critical. Regularly reviewing the VM can optimise the resources to support all

ISSN: 2231-7716
DOI: http://10.24191/ji.v14i2.224

existing and upcoming projects to reduce the cost for maintenance. The cost of managing the VM every month is dependent on the type of VM being looked at. As an example, for our case, the cost to manage four VMs can cost a hundred dollar.

Table 1: VM specification

ID	NAME	IP	VCPU	MEMORY	STAT
1982	Wiki	10.1.70.147	2	1024M	Running
2086	DB	10.1.70.221	2	1024M	Running
2093	MBIS2	10.1.70.227	4	8G	Running
2106	DB1	10.1.70.20	4	8G	Running

Each activity in the host is monitored to avoid wasting resources or any inactive VM. This table shows four VMs that are created, with their respective VCPU and memory. All the VMs are running but a notification will be sent often as a reminder to take any necessary action if the VMs are inactive. An email or notification is sent to the owner to review the VM applicability. There is a checklist for patching such as (Scarfone et al., 2005):

- Preparation by standardising the configuration and providing awareness to client
- Vulnerability identification
- Identifying patch for specific vulnerability
- Patch testing before deploying to production
- Patch deployment to client if the testing is a success

Client business model, physical host patching and VM purposes are another extra checklist that need to be included if the patching involved a virtualisation layer. Other advisory for vulnerability can be provided by CERT Coordination Center or vendor through websites and mailing lists. Shown in Table 2 below are some samples of patch type based on vendor severity.

Table 2: Patch type

Item	Vendor severity	Patch type
1	Critical	Security patches
2	important	Non-Security Patch
3	low	Security tool

There are different criteria for a patch. A patch can be critical, important or low based on their vendor severity. Categorisation of patch type is based on security, non-security and security tool. With this type of properties involved it can be converted into taxonomy for deployment and execution of the application involved. A classification of different type of application's functionality can make the application be executed as intended. The purpose of the second layer of application is to test the result and at the same time to identify whether the first application is working as expected.

The virtualisation layer, which is created in the host, causes security to become complex with a new paradigm. The new paradigm creates multi tenancy application with various users. Each user is able to access the application that resides in the host.

The rest of this paper is organised as follows: The first section discusses the problem statement and objectives of our proposed work. The second section discusses cloud orchestra ecosystem while the third section discusses a methodology to backup existing configuration and relation with host and VM for observation. Next, the fourth section discusses the related works that are relevant and providing

background knowledge to our work. Finally, last section concludes the proposed research work and explores the feasibility of future work.

PROBLEM STATEMENT

A routine task of upgrading and patching in certain period of time is necessary in order to provide a service and checkup of each machines and virtualisation resources. The other reason for upgrading is to support the application execution for specific tasks. During deployment of the application, which is related with other applications, it does not convert into a structure for observation when the execution occurs. This lacking can cause a problem for troubleshooting when execution face a failure for existing library resources or operating system which not follow the standard specification. There is no guideline or standard strategies to formulate the errors that exist, linkage properties between application and specification and tool which involved during testing. Any update or upgrade on the multi-tenancy application, which operated in silo or non-silo, can cause impact to the host. The virtualisation layer that is created from a physical host can contribute to the attack of the physical host through the flexibility of multi tenancy application under infrastructure as a service (IAAS).

CLOUD ORCHESTRA

Each of the entities has a different ecosystem from one another. For example, if the client is interconnected with each other, as shown in Figure 1, then a middleman is needed to ensure that if anything happens to client A, it will not impact client B.

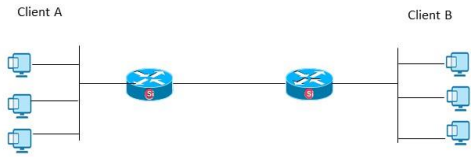


Figure 1: Network client

Each client consists of a respective cloud portal to manage the virtual resources. Information must be acquired through an audit in order to verify the resources before the process of upgrading can happen. A detailed framework of taxonomy is not provided to show how the mapping between application, virtual resources, client A and client B, if the system which be updated have a relation with other devices.
The compatibility and complexity task of upgrading the operating system (OS) and application in various situation which involved virtualization will not be discussed. The interoperability or impact to other setup after upgrading at host level to virtual machine will not be formulated into a taxonomy structure as a reference for next upgrade. For example, the change at the host level can cause instability of the virtualisation layer especially for connectivity of virtual machine. It is important to backup existing configuration of network, application or database as a reference for the error impact which is caused by the new deployment for later audit (Ishizu et al., 2008).

In the cloud there are variants of VM that reside in it, which are active and non-active, but not to be deleted by the owner. This can cause a degradation of optimisation of another VM. There is a management portal which allows the user to manage virtual resources under single interface as show in Figure 2. Cloud orchetsrator consists of:

- Front-end – provisions and operates services requested by the user. Network Virtualisation Manager shall install in front-end
- Node – physical host that will run the virtual resources
- Secure Authentication system which allows user to sign in, using just one 'identity', to various systems.

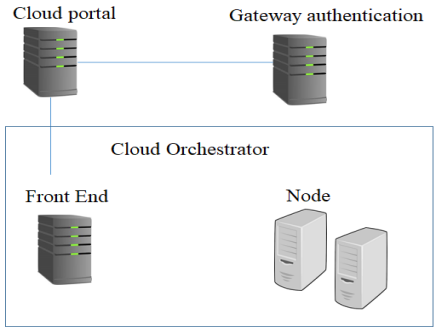


Figure 2: Cloud and gateway authentication

Upgrading Operating System Scenario

Each upgrading and patching effect can be different on host and VM which cause disturbance in the ecosystem of cloud computing. In Figure 2, existing host specification is utilised but it is involved in formatting the operating system only. The interface which is called the gateway authentication is also involved with the upgrading at the same time by upgrade the new version to support the latest OS.

Updating Security Patches, Bug Fixes and Application Upgrades

This standard patching and upgrading are based on the operating system. For ubuntu the command as below can support the update.

Example for *updated* Commands
sudo apt update && sudo apt upgrade -y
- It will updating the package

METHODOLOGY

The new deployment when upgrading or patching is tested with physical connectivity as a reference for the condition of the environment. Then a test is carried to see whether the host-to-host connectivity is working as expected as shown in Figure 4. In this Figure, a virtual network is created with multi tenancy. First, the breakdown of the categories into different clusters is shown below:

- Host under different cluster communicate with VM under different cluster
- Host to VM and VM to VM in same cluster

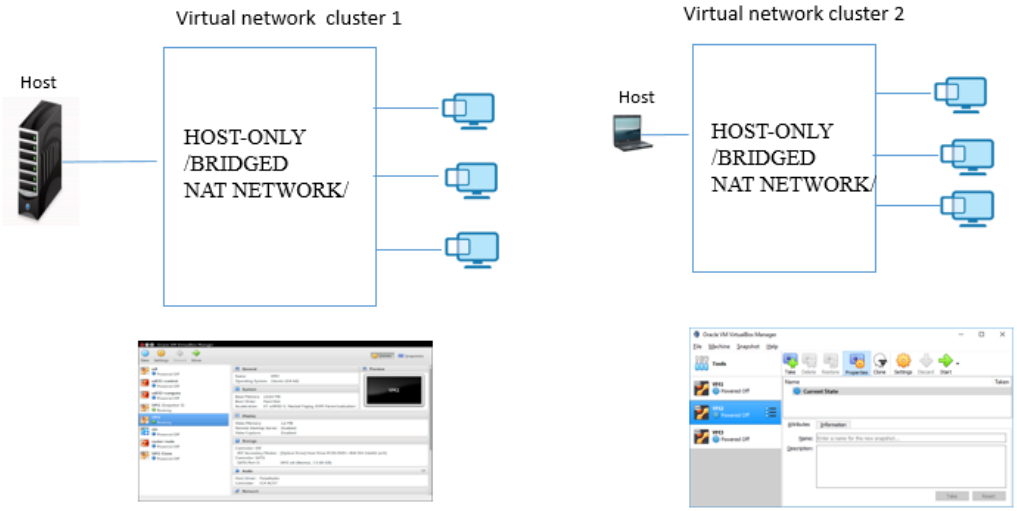


Figure 4: Cluster 1 and cluster 2 with virtual network

Physical and virtual device configuration is kept as a reference. Prior to that, a backup for all configurations before any activity of upgrading or patching is done. This is done as a precaution if the new deployment is not working as expected.

Testing Method

The connectivity is be classified between host-to-host, host-to-VM and VM-to-VM. First, test the connectivity between different clusters and secondly, between same cluster.

- Connectivity between different clusters

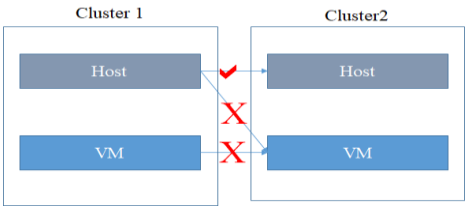


Figure 5: Connectivity between different clusters of the network

- Test the connectivity of VM which reside in the host on same cluster

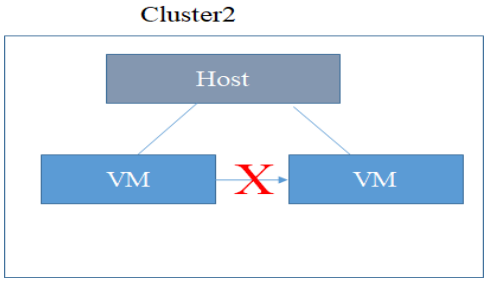


Figure 6: Connectivity between the same cluster of the network

This observation can be formulated into a strategy selection for specific properties' relationships to identify the operation of the application, which reside on the cloud portal. In this situation, an analogical model of the application's properties and how it can be converted into a structure will be discussed when the result of the testing is obtained.

RELATED WORKS

Morsy et al. (2010) analyses existing challenges and issues involving cloud computing security problem with multi-tenancy application and elasticity. Each multi-tenancy application with different approach technique has a risk that can impact the memory and hardware of the host. It also causes issues in securing the VM and workload in one of the devices inside the virtualisation layer from common security threats that affect the host such as malware and viruses. Priya Iyer et al. (2014) analyses a security issue faced by both the cloud supplier and customers though it provides many services and has several advantages. Basu et al. (2017) proposed a quantitative methodology to compute individual risk associated with the assets. Chakraborty et al. (2012) studied business strategies related to parliamentary government's departments. In addition, cluster strategies were identified which belong to respective of the taxonomy. A classification strategy develops a tool to facilitate a role of strategy formulator for business strategy.

EXPERIMENTAL RESULT

An application, which needs to be upgraded in the system, is studied. Then the process of how to formulate the structure application which reside in virtual resources such as VM is also studied. Upgrading the operating system without specific application that resides in the device is easier compared to when the machine has an application which is related with other applications. Upgrading or updating the library can cause imbalance to other application. It is recommended to back up the application configuration from the perspective of the network communication or software configuration. Upgrading the host without considering the VM can cause VM's appliance to stop working properly.

A specific application called Crawling needs to be installed. Unfortunately, the system cannot find certain packages, which can execute the crawling. Once the error is given, then a new technique and solution is wasted by try and error without a proper linkage and structure to solve the issue. Breakdown of the error into different categories show the relation with the web. It is still unclear why a combination of Selenium with Python does not match up during execution. At the end, this study was unable to get

Selenium with Python to execute a Firefox web browser. There is no clear guideline until a mapping taxonomy is done to ease the situation when the error exists, as shown below:

Example error
Selenium.common.exceptions.WebDriverException:Message:Unable to find a set of capabilities

A link and relation of the application is given as a step to analyse the chronology of the situation (Doty et.al., 1994). This is a result of existing machine using an older version of Firefox. A dependency happened and needed to be updated. A package named Python, an operating system with Ubuntu 14.04LTS is utilised with an older version of Firefox. A selected package is updated though a package that includes Python, dist-package, build-essential, libpq-dev, libssl-dev opnessl libffi-dev, zlibig-dev. At the end, the portal needed to be updated with the new version if the existing Firefox is not supported as shown in Figure 7.

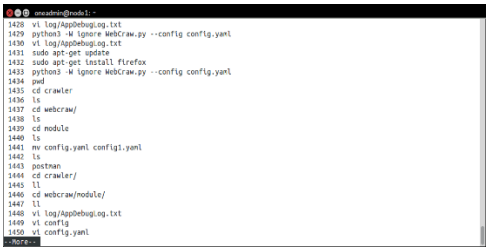


Figure 7: Update and install new portal

Application 1 is given an error as below. It shows that the first error mentioned about Selenium and relation with the web driver. An arrangement of criteria is based on certain specifications. The criteria are characters of structure application and functionality, which relate with the library and browser. Application's criteria are highlighted by the breakdown into library, operating system and browser as shown in Figure 8.

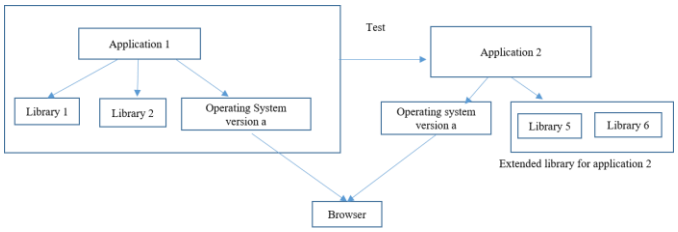


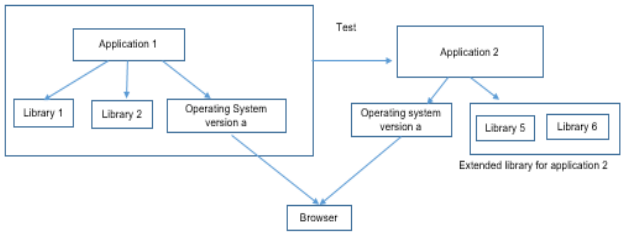
Figure 8: Taxonomy guideline for upgrading

A prerequisite installation for the new version of Python and Geckodriver will impact the Firefox browser due to their relationship with each other. Library 1 and library 2 on Application 1 contribute to the browser capability. When Application 2 is installed to test whether Application 1 is working or not, it shows a relationship between the two libraries.

Based on the test strategy, a classification is developed between the libraries, supporting tool and multiple applications thus identifying the taxonomy relationship. A property taxonomy of the relationship between the first and second application can show how the linkage are arranged between the two applications. The second application is proposed for troubleshooting of the first application.

RECOMMENDATION

The taxonomy relationship can improve the domain functionality by adding some modification in the middle for the query classification (Bruce Croft et al., 2010) or by adding classification optimisation. A long keyword sometimes can give an accurate hit but the traffic is low and it will depend on the specific application. A low traffic for a specific application which have a simple keyword is based on the application error exist during troubleshooting.
A regular keyword for a specific application can create a heavy traffic but not significant result. For example, when giving selenium as a keyword during troubleshooting, it will show an involvement of the web. By using a different keyword style, if the keyword is web and selenium, used together, the amount hit will randomly give a result without specific solution to the error that exists. A specific keyword can cause multiple results but unfortunately, it will not give a significant hit. A long keyword sometimes can give an accurate hit, but the traffic is low and will depend on the specific application. A low traffic for a specific application with a simple keyword is based on the application error during trouble shooting. Shown below in Figure 9 is the short keyword and long keyword. The significant hit can be different for each keyword. It is categorised as low, medium and high as a threshold during searching the relevant information for specific cases. This categorisation can optimise the process of finding a specific solution.



Significant hit's number			
Keyword	Low	Medium	High
Crawler		✔	
Selenium		✔	
Python	✔		
Web	✔		
Unable to find a matching set of capabilities			✔
selenium.common.exceptions			✔

Figure 9: Taxonomy chronology mapping for test the application.

CONCLUSION

Although all the upgrade and update had already been done, a process to ease the troubleshooting can be provided based on the taxonomy classification and search engine strategy. A mapping taxonomy for upgrading is done to reduce the risk and give awareness and decision suggestion for incoming situation when the errors still exist. In this case study, a complex situation is faced when the environment of the application is not upgraded and updated properly for deployment. The decision can give an early warning by identifying and analysing possible future events, which will happen if the execution is selected with a

specific library. A test strategy provides a standardised checking through a classification of taxonomy and searching strategy. This formulated strategy and taxonomy can be expanded to approach a complex virtualisation layer.

REFERENCES

Zhou, W., Ning, P., Zhang, X., Ammons, G., Wang, R., & Bala, V. 2010. Always up-to-date – Scalable offline patching of VM images in a compute cloud. Paper presented at Twenty-Sixth Annual Computer Security Applications Conference, ACSAC 2010, Austin, Texas, USA.

Scarfone, K., Soupp, M., & Johnson, P. M. 2005. Guidance for securing Microsoft Windows XP systems for IT professionals: A NIST Security Configuration Checklist. Recommendations of the National Institute of Standards and Technology. NIST Special Publication 800-68.

Morsy, M. A., Grundy, J., & Müller, I. 2010. An analysis of the cloud computing security problem. Presented at Asia Pacific Cloud Workshop, Colocated with APSEC2010, Australia.

Priya Iyer, K.B., Priya, P., & Anusha, R. 2014. Analysis on Cloud Computing Security Issues, Threats and Solutions. Presented at International Conference on Communication, Computing and Information Technology (ICCCMIT-2014).

Basu, S., Sengupta, A., & Mazumdar, C. 2017. A Quantitative Methodology for Cloud Security Risk Assessment. Presented at 7th International Conference on Cloud Computing and Services Science.

Chakraborty, A., & Stewart, G. 2012. Strategy Taxonomy and Classification System Development – Study of two State Governments. Presented at Sixth IEEE International Conference on Management of Innovations and Technology.

Doty, D. H., & Glick, W.H., 1994. Typologies as a unique form of theory building: Toward improved understanding and modelling. Journal Academy of Management,19, p.230.

Bruce Croft, W., Metzler, D., & Strohman, T. 2010. Search Engines: Information retrieval in practice. Boston, Massachusetts: Pearson Education.