Enhanced Framework To Migrate Virtual Machines To The Container In Cloud Environment

[1]S. Raja , [2]Dr. S.S. Manikandasaran

[1] Research Scholar, PG and Research Department of Computer Science, Adaikalamatha College, Vallam, Thanjavur, Tamil Nadu, India. Affiliated with Bharathidasan University, Trichy.

[2] Assistant Professor, PG and Research Department of Computer Science, Adaikalamatha College, Vallam, Thanjavur, Tamil Nadu, India. Affiliated with Bharathidasan University, Trichy.

Abstract

Cloud Computing is an extension of virtualization concepts that can include 'Service vice'. It has extensive resource scaling and automation capability to integrate everything in a holistic view. In recent days, Cloud Computing has had multiple enhancements and is maturing with more flexibility. Still, security issues such as Confidentiality, Integrity and Availability remain, and cloud adoption is happening drastically even if issues persist. A virtual machine is a building block for the application landscape, and application architecture is monolithic. The container is the next level of abstraction layer in cloud computing. To migrate the existing virtual machine workload into container-based will get into many security issues. This paper proposes a secure architecture to migrate the virtual machine workload to containers in Cloud Computing to align with security parameters. Current industry trends and challenges migrate the workload into the cloud with server-less architecture and containerization.

1. Introduction

Cloud Computing is the underpinning platform which gives priority to security. The entire information technology transformation is adapted from virtual machine technology to container strategy. The container is an extensive technology and is an optimized piece of the virtual machine that will do process isolation in either bare metal hardware or a virtual machine. In this paper, cloud adoption takes place from virtual machine to container with a strong commitment to security parameters which comprise confidentiality, integrity and availability. The application migration will be progressive since the cloud adoption process should not impact existing application transactions. The application migration from monolithic to micro-service architecture required more effort with strong security around the wall. Monolithic application architecture defines that heritage method and is tightly coupled. Micro-service architecture is a modern way of managing the application with full-fledged automation capability. The twelve-factor methodology can be applied to apps written in any programming language and use any backing services, such as database, queue, memory cache etc., in micro-service-based applications [1]. When customers intend to migrate legacy applications into micro-service architecture based, security is an eminent factor.

This paper is organized as follows. Section 2 presents the Components of Cloud Computing, and section 3 narrates the related research works. Section 4 defines the problem of the proposed work. The methodology is presented in Section 5, and Section 6 explains the proposed security architecture. Analysis of the Proposed work is presented in Section 7. Finally, the conclusion is given in Section 8.

2. Components of Cloud Computing

There are two types of components in Cloud Computing. They are Cloud Enabled and Cloud Native, defined in this section.

2.1 **Cloud Enabled**

Cloud Enabled is the application that was moved to the cloud, but it was originally developed in a traditional data centre. Therefore, some application characteristics had to be changed or customized for the cloud. But on the other hand, it is an application developed with the cloud principles of multi-tenancy, elastic scaling and easy integration and administration in its design. In a nutshell, a cloud-enabled environment has the semi-automation capability and full automation capability covered in the Infrastructure as a Service (IaaS) [2].

2.2 Cloud Native

Cloud-native is the term for fully cloud-based applications which operate globally and can be accessed anywhere, even if the internet is not working. The application services and data have been replicated in multiple locations. It can be hosted anywhere in the private, public, or hybrid cloud. In cloud-native applications, deployment and functionality changes happen rapidly, which is impossible in legacy applications. Often, it is called continuous innovation, which aligns with agile methodology. Cloud Enabled and Cloud Native Components in Cloud Computing are presented in Figure 1.

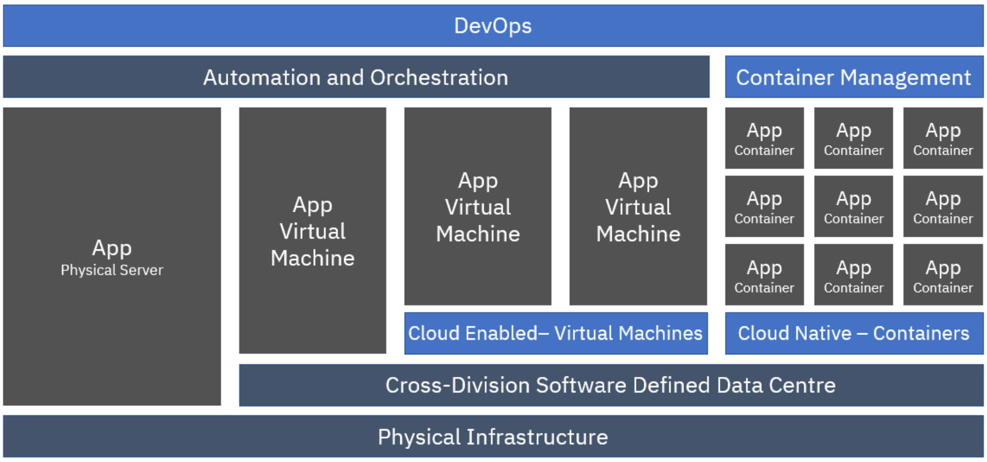

Figure 1. Cloud Enabled and Cloud Native Components in Cloud Computing

3. Related Research Works

Security remains the biggest objection to cloud computing and the number one inhibitor of broad-scale adoption. Information Technology leaders are expected to enable the business to innovate and do more for less and cloud computing presents this opportunity. However, IT departments are concerned with reduced visibility into cloud datacentre, less control over security policies, new and unknown threats facing shared environments and the complexity of demonstrating compliance. In traditional perimeter-based security, the population is divided into Trusted and Untrusted users. The network is used to create a demilitarized zone to keep trusted users behind the firewall and make the enterprise applications accessible behind the firewall. Likewise, this demilitarized zone keeps untrusted users out because they are untrusted and not part of the enterprise.

Dennis et al. defined the term "Cloud Native" [3]. Cloud Native is a modern platform as a service with scalability and operates globally. Its well-defined parallelism has been achieved in Cloud Computing to handle multiple users concurrently. The first transformation initiative was in the industry to adopt Infrastructure as a service in both on-premises and cloud with extensive capability of automation to spin the instance in one click button. Micro-service is a new design pattern in the cloud environment that is easy to manage, replicate, scale, upgrade and deploy independently. The communication mechanism differs little from the heritage application architecture in Micro-service application architecture. It uses remote procedure call mechanisms and advanced message queueing protocol. Linux operating system provides complete encapsulation of process by the name of the namespace. The author described the serverless platform that operated on event-driven beyond the micro-service. Most cloud service providers have an offering related to serverless platforms. It is often called a "Fully Managed" platform.

Salman basset et al. discussed the point of view of container security [4]. The authors defined the benefits of container applications; for The containers have visibility of very limited resources rather than controlling the entire operating system. As a result, the container has an efficient way of handling application requests, simplified management, portability and less attack surface for the workloads. The author described core components of containers like namespace, control groups, Linux capabilities, second, Linux security modules, selinux, appArmour and user namespace. However, the author did not describe concrete security parameters that were defined and described, which are related to the container.

Ashif Khan defined the virtual machine as a container transformation toward the serverless platform [5]. The author explained the building blocks of application agility: container encapsulates operating system processes that allow own private namespace and computations resource limits, including memory and CPU. The author has demonstrated container orchestration capabilities. Containers can run on multiple platforms. In a container clustering platform, the Orchestration layer is responsible for maintaining the cluster state, which is important to run operations. It has scheduling capability, container backups, garbage collection, file consolidation, and index rebuilds such as binning and instance affinity. Security is the most important factor in container orchestration, ensuring the deployed services' high-security standards and integrity. The key difference stated between the physical platform and container platform. The author has defined the various stages of the process in the micro-service architecture. The service registry is responsible for containing the network location and is highly available to avoid a single point of failure. This author has defined types of discovery methods, i.e. client-based discovery methods able to determine the network location of available instances and load balancer to handle them. Server-based discovery method, the client requests service via a load balancer. The load balancer is responsible for querying against the service registry and router each request to an available instance. Continuous delivery and deployment is the development practice when code changes happen. It has been automatically built, tested and prepared for production release. Monitoring and governance have been differentiated between physical infrastructure and container platforms. In the Container platform, Tools is responsible for providing the white box tracing logs, logging events to an approved logging store and monitoring container performance.

Kennedy et al. discussed a new approach for designing security as a service for cloud-native-based applications [6]. The author re-designed security architecture from traditional web applications to cloud-native-based applications. It defined how security assessment had been taken care of before the continuous integration process kicked in but did not express countermeasures if the assessment failed. The author discussed the security requirements of the cloud native applications: application security, Network security and data security. The key things highlighted in application refactoring when converting traditional applications to cloud-native patterns. The secure architecture implemented and captured the charts and response time but did not mention the countermeasure.

Jeeva et al. discussed techniques of security containers from distributed denial of service (DDoS) [7]. The authors defined virtualization and containers. It has been highlighted that container spawn techniques. When the container is created, the virtual Ethernet interface will be connected through the bridge interface. The networking component plays a vital role in the Denial of Service (DoS) attacks. The container requires root privilege to control the process and can connect to the host kernel. Hackers can easily exploit using memory allocation techniques.

4. Problem Definition

In the recent Cloud Computing transformation strategy, the end users face different challenges when converting their workload from a monolithic architecture to micro-service architecture design patterns. There are multiple methods to convert from monolithic to micro-service, like re-writing the entire application's code, which will take more effort, and another one is migrating the existing workload to micro-service patterns, which will have more challenges in all the efforts. In addition, security is still a million-dollar question regarding micro-service-based cloud applications.

5. Methodology

Cloud computing promotes serverless architecture andInfrastructure as a code (IaC). However, the end user faces challenges. This paper proposes an architecture to migrate the application from monolithic to micro-service. Previous work involves migrating the traditional workload from on-premises into the cloud and enhancing the current architecture to converting the traditional application into modern architecture with cloud-native capability.

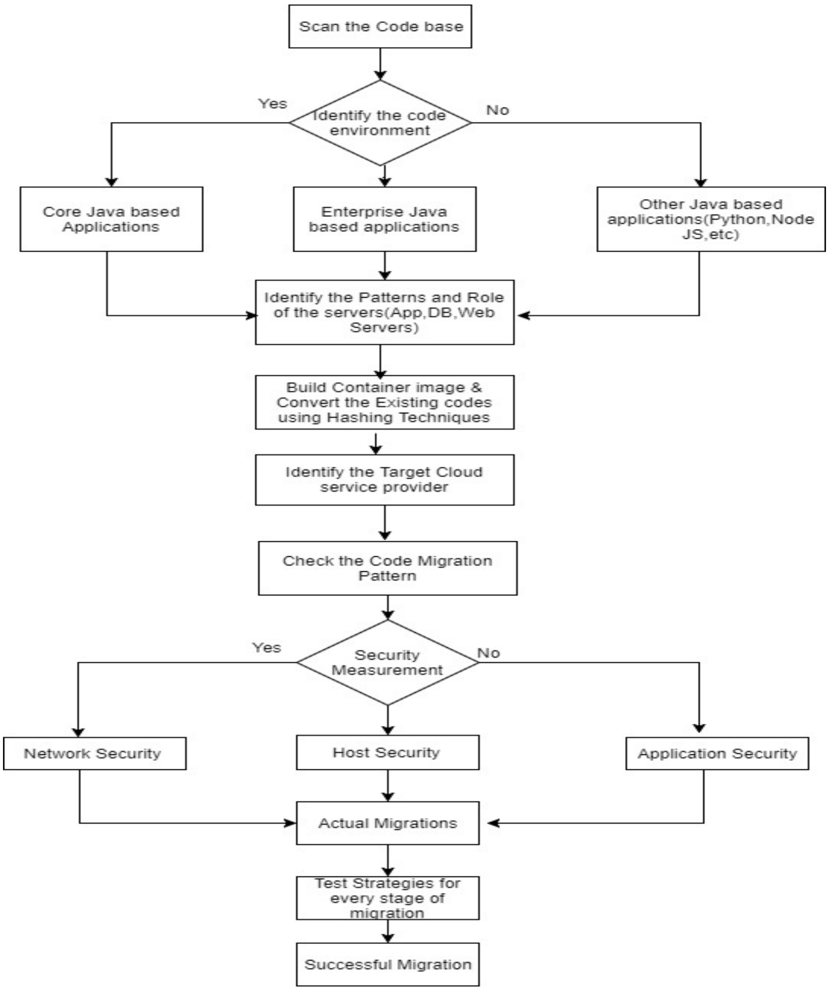
Scan the Code base
Yes
Identify the code environment
No
Core Java based Applications
Enterprise Java based applications
Other Java based applications(Python,Node JS,etc)
Identify the Patterns and Role of the servers(App,DB,Web Servers)
Build Container image & Convert the Existing codes using Hashing Techniques
Identify the Target Cloud service provider
Check the Code Migration Pattern
Yes
Security Measurement
No
Network Security
Host Security
Application Security
Actual Migrations
Test Strategies for every stage of migration
Successful Migration

Figure 2. Migration Methodological Diagram

Figure 2 Methodological diagram represents multiple stages of migration. It has involved converting the existing monolithic architecture to micro-service architecture among these phases. Discovery is a minimum requirement for detailed analysis to understand the application landscape.

```
Step 1:     Scan the codebase
Step 2:     Identify the code environment
            If lang=" J2SE" || lang=" J2EE" || lang=" Python" || lang=" NodeJS"
                ServerRole=" AppServer"
            Else
                ServerRole=" WebServer|DBServer"
Step 3:     Build the Containerized image and store the Trusted Container
        Repository
Step 4:     Convert the code into Hashing technique
Step 5:    Identify the Cloud Service Provider
Step 6:    Check the Code Migration Pattern
Step 7:    Checking Security Parameter
                    {Network Security
                 {Defined DDoS Prevention Parameter
                 Check Security policy for container and Bridge network
                 }  }
           Host Security {
                 Check Host is not vulnerable
                  Port Forwarding and firewall enabled
                 }
           Application Security {
                 Check Code is not vulnerable
                 TLS is enabled for application communication
                 API end is secured
                 }
Step 8:    Perform the migration
Step 9:    Perform test validation upon successful migration
```

Table 1. Algorithm for Virtual Machine to Container Migration

6. The Proposed Security Architecture

Existing work involved migrating virtual machines to virtual machines into on-premises Cloud data centres [7]. In this paper, the proposed secure architecture is to migrate from monolithic architecture workload to micro-service architecture, which comprises minimum refactoring of application code. Monolithic architecture is hosted in traditional Infrastructure, which is slow to change, complex e fragile, difficult to scale, and lacks resiliency and flexibility. Micro-service architecture has Table 1 provides details steps on how to migrate from monolithic to microservice architecture.

This architecture is deployed on self-service, elastic, cloud computing infrastructure, preferable containers, autoscaling and resiliency. The Micro-service architecture meets all requirements but requires extra focus on security vulnerabilities, confidentiality, integrity and availability. Container images are basic building blocks and important elements in the software development lifecycle. The proposed architecture will prevent code vulnerabilities in container-based platforms since shared kernel architectures must be accessed via standard configuration and container profiles. Container orchestration is responsible for providing coarse-grained and fine-grained access control for the container. Networking is the key factor for container orchestration platform efficiency. The container created three networks host, bridge and none for communication. In this paper, container communication explains how the container gains IP address during the entire life cycle. Service discovery is the factor application required to know the network location.

The proposed architecture meets the security requirements when converting the workload from on-premises to cloud computing as the micro-service pattern. The container images and Host / Hypervisor underlying infrastructure must be hardened.

- The data volumes are shared across containers. Therefore, it must be encrypted when containers mount the data volume during the transaction.
- Port forwarding is strictly handled since a virtual network interface will be created when the container spawns, and network packet sniffing should be disabled.
- User access should be handled as non-privileged users to prevent exploitation.
- Logs will be shipped properly during the container transaction.

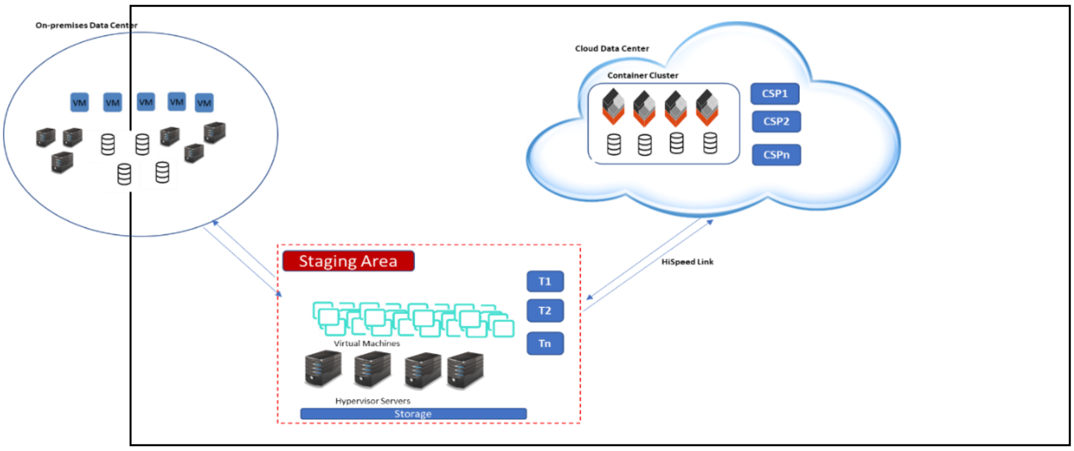

Figure 3. Secure Architecture for Virtual Machine to Container Migration

7. Analysis of Proposed work

The defined architecture will help customers migrate monolithic application into micro-service architecture with less code refactoring. It mainly focuses on converting the application from a tightly coupled environment to a loosely coupled environment. The proposed architecture in the cloud computing environment intends to use a non-privilege container since a non-privilege container is potentially unsafe. A unique identifier (UID) is mapped into the container host root user id. The container image should be built from a trusted registry. The architecture covers the Cgroup limitation, which inherits from the parent. The normal user who uses a container can reasonably DoS Host by running a consistent number of fork processes to generate Payload. Due to this unlimited Payload, the host kernel will run out of memory. The user limitation is inherited from the parent, and the limit is tied to the kernel's name. The resource sharing possibly includes ID maps, common kernel UID, and PID maps. This paper proposes architecture for migrating monolithic application into micro-service with this consideration.

Our proposed architecture will have well-defined east-west traffic in the Container cluster. East-west traffic is Layer 2 network communication within the environment which will be connected to the bridge interface in the container host operating system. The namespace is the isolation for the Linux namespaces and deeply describes isolation through system calls. The differentiation between the container and virtual machine is how to process maps in the operating system since the container shares the same kernel space and is easy to exploit if the Host has vulnerabilities. This is one of the main factors and will lead easily to potential exploitation if we don't handle it properly. All container definitions will be defined when we scan the source codebase environment. The data volume is encrypted when containers mount the volume during the transaction. The proposed architecture is enforced to use AppArmor for enhancing security. As Continuous Integration and Continuous Delivery are important factors in bringing agility to the micro-service architecture, these properties are default in the proposed architecture.

8. Conclusion

The Cloud Computing transformation journey is in the next stage. Business users face many challenges in each technology adoption stage and bring more attention to the researchers. The cloud service providers created a platform to design the new application but focused less on the existing workload migration and conversion into the cloud without impacting the current environment. The proposed new security architecture for the virtual machine to container migration involves multiple stages of discovery and defines the target cloud environment.

This paper has detailed the migration of workload into the fully automated environment in cloud computing. In Future, the proposed model will be implemented and achieved for better results. Furthermore, it will focus on researching how code-level security and data security will be carried out once the services are converted into monolithic microservices.

Conflicts of interest

The authors have no conflicts of interest to declare.

9. References

[1] Twelve-Factor methods https://12factor.net/ accessed on 05 October 2018.

[2] Arockiam, L., S. Monikandan, and G. Parthasarathy. "Cloud computing: A survey." Journal of Computer and Communication Technology: Vol 8.1 (2017): 4, pp. 21-28.

[3] D. Gannon, R. Barga and N. Sundaresan 2018 Cloud-Native Applications IEEE Cloud Computing, **4,** 5, 16-21.

[4] Salman Baset, Stefan Berger, James Bottomley, Canturk Isci, Nataraj Nagaratnam, Dimitrios Pendarakis, J. R. Rao, Gosia Steinder and Jayashree Ramanatham Docker and Container Security White Paper

[5] A. Khan Key Characteristics of a Container Orchestration Platform to Enable a Modern Application IEEE Cloud Computing **4** 5 42-48

[6] K. A. Torkura, M. I. H. Sukmana, F. Cheng and C. Meinel 2017 Leveraging Cloud Native Design Patterns for Security-as-a-Service Applications IEEE International Conference on Smart Cloud (SmartCloud) New York 90-97.

[7] Manikandasaran S. S. and Raja S 2018 Security Architecture for multi-Tenant Cloud Migration, International Journal of Future Computer and Communication **7** 2 42-45.

[8] Shu, Rui & Gu, Xiaohui & Enck, William. 2017 A Study of Security Vulnerabilities on Docker Hub. 269-280. 10.1145/3029806.3029832.

[9] IaaS, PaaS and SaaS – IBM Cloud service models. [Online]. Available: https://www.ibm.com/cloud/learn/iaas-paas-saas accessed on 05 October 2018

[10] L. Arockiam and S. Monikandan 2013 Data security and privacy in cloud storage using hybrid symmetric encryption algorithm International Journal of Advanced Research in Computer and Communication Engineering **2** 8 3064-3070.

[11] S.S. Manikandasaran, K. Balaji and S. Raja 2018 Infrastructure Virtualization Security Architecture Specification for Private Cloud International Journal of Computer Sciences and Engineering **06** 02 10-14.

[12] https://www.ibm.com/blogs/cloud-computing/2014/08/13/deploy-cloud-enabled-cloud-centric-application/.

[13] Monikandan, S., and L. Arockiam. "Confidentiality technique to enhance the security of data in public cloud storage using data obfuscation." Indian Journal of Science and Technology 8.24 (2015): 1, pp. 88-97.

Mr. S Raja is Research Scholar in PG and Research Department of Computer Science, Adaikala Matha College, Vallam, Thanjavur, Tamil Nadu, India. He has 17 years of experience in IT industry. He is completed his M.Sc. in Bharathidasan University, Tiruchirappalli in 2005. Hehas attended many International and National Conferences, Seminars and Workshops. He has delivered more than 5 international and national conferences. His research interest is cloud computing, cloud security, Cloud IoT and Big data analytics.

Email: rajasjc@gmail.com

Manikandasaran S. S. is working as Associate Director in PG and Research Department of Computer Science, Adaikala Matha College, Vallam, Thanjavur, Tamil Nadu, India. He has 14 years of experience in teaching and 13 years of experience in research. He completed his MCA and M.Tech in Bharathidasan University, Tiruchirappalli, in 2007 and 2009, respectively, and completed his PhD in Manonmaniam Sundaranar University Tirunelveli in 2015. Now he is pursuing Post Doctoral Fellowship at Srinivas University, Karnataka,India. He has attended many International and National Conferences, Seminars, and Workshops. He has published 56 research articles in the International / National Conferences and Journals. He has delivered more than 45 lecturers in various National and International level seminars, workshops, and conferences. He is a author of a book. He has published two Indian Patent. His research interest is Cloud computing, Network Security, Cloud Security, IoT, and Web Technology.

Email:ssmanikandasaran@gmail.com