Indonesian Journal of Electrical Engineering and Computer Science
Vol. 21, No. 2, February 2021, pp. 874~885
ISSN: 2502-4752, DOI: 10.11591/ijeecs.v21.i2.pp874-885

A secure cloud service deployment framework for DevOps

P Ravinder Rao[1], V.Sucharita[2]
[1]Research Scholar in Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram, AP, India
[2]Supervisor in Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram, Narayana Engineering, College AP, India

Article Info

Article history:

Received Jun 13, 2020
Revised Aug 11, 2020
Accepted Aug 21, 2020

Keywords:

DevOps
Variable length encryption change management
VM deployment
VM migration security

ABSTRACT

The advancements in cloud computing and leveraging the benefits from cloud computing to the service providers have increased the deployment of traditional applications to the cloud. The applications once deployed on the cloud, due to various reasons, need migration from development infrastructure to operational infrastructure, one operational instance to other operational instances due to load balancing and the cycle continues due to the use of DevOps as development strategies for cloud computing applications. Advocates of hybrid and public clouds observe cloud computing makes it possible for organizations to avert or minimize upfront IT infrastructure expenses. Proponents also assert that cloud computing systems permit businesses to receive their software up and running faster, using improved manageability and less maintenance, so it empowers IT teams to rapidly adapt tools to meet the varying and unpredictable requirements. DevOps is a lot of practices that mechanizes the procedures between programming improvement and IT groups, all together that they can fabricate, test, and discharge programming quicker and even more dependably. The idea of DevOps is established on building a culture of a joint effort between groups that generally worked in relative siloes. The guaranteed advantages incorporate expanded trust, quicker programming discharges, capacity to explain basic issues rapidly and better oversee impromptu work. Thus, this work identifies the need for providing multiple security protocols during the complete life cycle of cloud application development and deployment. This work proposes a novel framework for automatic selection and deployment of the security protocols during cloud service deployments. The framework identifies the need for security aspects and selects the appropriate security algorithms for virtual machines. The proposed framework demonstrates nearly 80% improvement over the security policy deployment time.

Corresponding Author:

P Ravinder Rao
Department of Computer Science and Engineering
Koneru Lakshmaiah Education Foundation
Vaddeswaram, AP, India
Email: ravisri.sunny@gmail.com

1. INTRODUCTION

Motivated by the benefits, such as measurable advantages, cost improvements, and the scalability benefits of migrating the applications on the cloud, a good number of applications are migrated or hosted directly on the cloud. Leveraging the advantages from the cloud, the service providers or the application developers can take the benefits of the platform, infrastructure, or the build-in services from the cloud

providers as PaaS, IaaS, or the SaaS. Also, the application developers can take the freedom of deployment capabilities with application infrastructure or the maintainability, or the accessibilities of the application by hosting the applications on public, private, hybrid or community cloud hosting plans. Nevertheless, the migrating the cloud applications on the cloud also brings in multiple challenges. Especially during the deployment of the cloud services, the infrastructure or the build-in services may differ from the development environment to the operation or deployment environment. These dependencies can lead to defects or sometimes data and application security vulnerabilities.

However, the security protocols deployed on the cloud servers can protect the data at the cost of additional time complexity, which increases the response time of the applications and may lead to a decrease in customer satisfaction. Thus, several research attempts were made to reduce the time complexity of the security protocols or the algorithms without reducing the effectiveness to fight against the attacks on the cloud. Thus, this research also attempts to reduce the time complexity of the security protocols by deploying a novel change detection and variable-length encryption algorithm for the virtual machine data.

The existing system primarily focuses on the infrastructure-based security for the cloud-services. That means it can only deal with the security issues after deployment. However, the proposed system [Paper – 2] can provide security for the cloud services from development to testing to deployment, in all phases. The major drawbacks of these researches to ignore the modern software application development strategies, change management in the source code and finally the growing or shrinking volume of the virtual machines, responsible for application hosting and used for migration. This proposed framework identifies the advantages of detecting change management, produced by the DevOps process, as a scope for reducing the time complexity. Also, the virtual machine fundamental capabilities as variable length virtual machines can also help in reducing the time complexity further.

Literature review : the research outcomes from the parallel researches are analyzed. Motivated by the needs of various application developers to provide the best security to their applications, business logic, and the data, a good number of parallel researches are carried out. Most of the parallel researches have identified the need for robust algorithms to secure the applications and protect the privacy of the business users and attempted to accommodate the neediness.

The work of P. J. Bruening et al. [1] elaborates on the privacy issues of the cloud-based applications to a greater extend. Also, the notable survey and analysis by M. Ouedraogo et al. [2] define the importance of cloud service provider selections considering the growth in cybercrime reports. Nonetheless, the migration of the cloud applications must be taken with the highest priority to maintain the market capitalization and building highly customers satisfiable application. This migration demands highly automated tools and strategies. The work by A. Khajeh-Hosseini et al. [3] elaborates on the advantages of various tools to support the migration process and critical decision support situations. These same advantages are also well discussed by P. Jamshidi et al. [4].

For a long, the application developers have hesitated to deploy or migrate their applications or services on to the cloud as the business applications and the data generated by these applications are hosted on third-party infrastructure as one of the primary concerns. The early days' concerns and the traditional security measures are well documented in the work carried out by C. Kalloniatis et al. [5]. Also, yet another concern of the user privacy, the consumers using these deployed applications, was the triggering factor as well. The application developers were anxious to find out the depth of the security provided by the cloud providers. Nonetheless, the lack of availability of the security mechanism strength analysis measures was a real bottleneck. The work of S. Pearson et al. [6] highlighted the gaps and the FedRAMP [7] as an automated and independent organization has formulated the guidelines. These guidelines ensured the compulsory security measures to be taken by all cloud providers for all applications. These made the application developers increase their trust in hosting the applications on the cloud. The overall scenario is well furnished in the work of P. J. Bruening et al. [1]. As per the report by P. J. Bruening et al. [1], the majority of the cloud-based applications are vulnerable to open APIs, leakage of business data, improper data management, and malicious access, making the applications highly insecure. Thus, the researchers have attempted to provide proactive security measures to avoid or prevent these security breaches. Analyzing the risks of open access APIs, the research by M. Theoharidou et al. [8] opens newer dimensions of the works. Also, the contribution by H.Takabi et al. [9] lists the significant benchmarks for making the security algorithms better. Not limiting to open API defects, the work of M. Gregg et al. [10] elaborates on securing the application data from a trojan or the man-in-the-middle attacks.

The security concerns differ from the scenarios to scenarios of cloud service providers and the application types. Regardless to mention, many of the applications must provide open access to the business data due to the process flow of the applications and also, many of the applications, in the other hand, must provide multiple levels of security layers for the authenticated users to access the business data, encouraging

the higher time response times. The differential security measures are well demonstrated by the work of S. Pearson et al. [6].

The business data, generated by the application hosted on the cloud, is often replicated to provide the parallelism of the applications. Due to various business requirements, the replicated data in various locations are secured with different security policies. The security policies in different locations can be identified as weak or very strong depending on the requirements. The attackers can identify these vulnerabilities and make the data tampering. The solution to this problem was provided by M. Mulazzani et al. [11].

Many of the instances, the application developers and cloud service providers are criticized for making the security extremely high and compromise on the time complexity. A generic framework was proposed by S. De Capitani di Vimercati et al. [12] to identify the desired complexity level of security. This framework is well accepted in researchers and practitioner's communities.

The virtual machines and the applications running on these virtual machines can also become vulnerable to the attacks as the virtual machines migrate as plain text over the internet. The initial strategy for securing the virtual machine data was proposed by D. G. Rosado et al. [13] and the work is further enhanced by M. Klems et al. [14] for e-business systems.

The virtual machines are vulnerable during the migration process for many possible migrations and other tasks [15-19]. As a result, the Service provider will become out of service whenever many packets are flooded by attackers. Various technologies are being used for handling such attacks [20-25].

The technology which has a boom in the field of computer science nowadays is considered as Cloud Computing. Various research projects are being undertaken by many researchers in this field. Researchers are developing different Custom-Driven applications under cloud environments. Many IT Corporate Companies like Google, IBM, Oracle, Microsoft have started to enhance the integration of their respective products with the emerging Cloud Computing Technology. But, It has become a tedious task to integrate their specific products in Cloud Computing environments. Interoperability is the key to become the success of Cloud Computing in future endeavors. One of the ideas of this research is to introduce Cloud Interoperability in the field of Cloud Computing [21-24].

The load in Cloud Data Center will be imbalance whenever a huge volume of sequential incoming function requests to physical hosts. Execution inefficiency for tasks is carried out in the long run even some existing works balance the load with optimization algorithms in selecting the optimal host for achieving instantaneous load balancing [22-23].

Nonetheless, the present time complexity of the security policies is higher and signifies a possibility for reduction. The reduction in time complexity must address the continuous changes in the source code of the applications and the variable length of the virtual machines deployed over the cloud instances. Henceforth, with a detailed understanding of the research outcomes from the parallel researches, this work provides a novel framework to address various needs for cloud-based application deployment securities.

2. FUNCTIONAL OUTCOMES FROM DEVOPS

The functional resuts from the DevOps process are identified. These results will help to realize the security needs for the cloud application service. DevOps is the blend of social theories, practices, and devices that expand an association's capacity to convey applications and administrations at high speed: This speed empowers associations to more readily serve their clients and content even more successfully in the market.

Under the DevOps model, improvement and activities groups are no more drawn-out "siloed." Sometimes, these two groups are converted into a solitary group where the architects work over the whole application lifecycle, from advancement and test to send to the task and build up a scope of abilities not constrained to a un accompanied capacity as shown in Figure 1. In some DevOps models, quality affirmation and security groups may turn out to be more firmly coordinated with improvement and activities and all through the application lifecycle. At this point, when security is the main emphasis of everybody's on a DevOps group, this is to be alluded to as DevSecOps.

These groups use practices to convey forms that verifiably have been manual and moderate. They utilize an innovation stack and tooling, which enable them to work and develop applications rapidly and dependably. These apparatuses likewise help to build autonomously achieve assignments (for instance, sending code or provisioning framework) that typically would have required assistance from different groups, and this further expands a group's speed as shown in Figure 2.

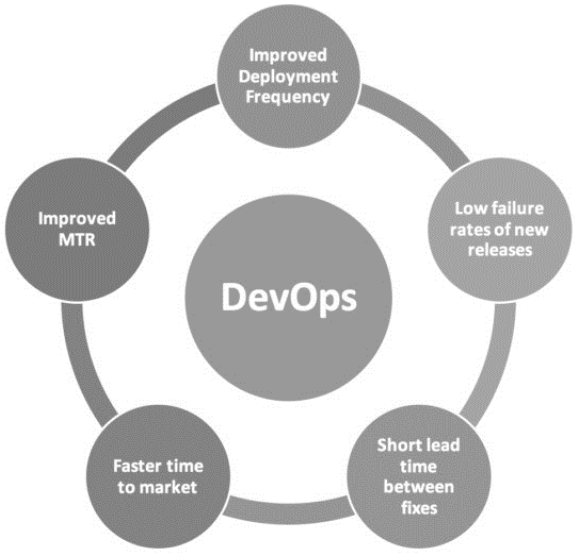


Figure 1. DevOps capabilities

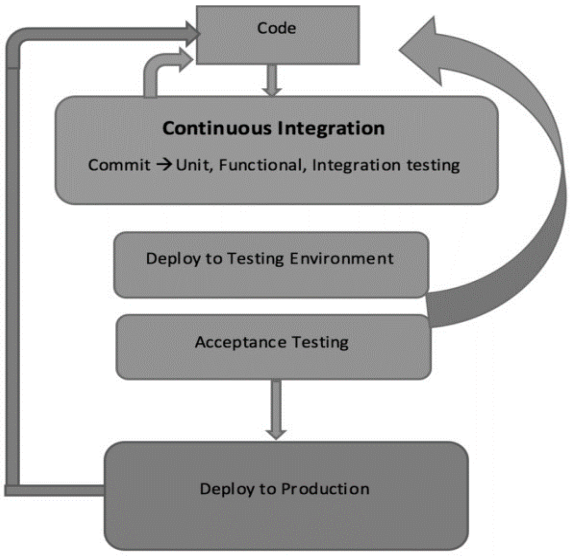


Figure 2. Process oriented DevOps

2.1. Create process - people - tools ecosystem

DevOps incorporates individuals, procedures, and devices together to change an association into a solitary substance.

2.2. Integration for development

When the entire association is ready, the developers center should swing to the DevOps group itself. Here, the fundamental issue is to give developers precise, a la mode data about the generation condition so they can design the organization properly.

2.3. Testing

The snappier the company gets input on changes, the better developer's programming quality will be. In the customary cascade process, the total code moves from improvement to the testing region and is then pushed to generation on the off chance that it effectively finishes the test. If not, the code is sent back to advancement for altering. It takes additional time and is less dependable.

2.4. Deployment for heterogeneous systems

Constant sending broadens nonstop conveyance. Each form that effectively finishes a full test cycle is naturally sent. It kills the requirement for human intercession to choose what to convey and when to send. With nonstop sending, associations can rapidly convey new highlights and updates while proactively making changes to the item. To decrease downtime and moderate dangers, associations can think about what's known as a blue/green organization. In this technique, when a change is made and another sending (blue) is activated, it is conveyed in parallel to the former one (green).

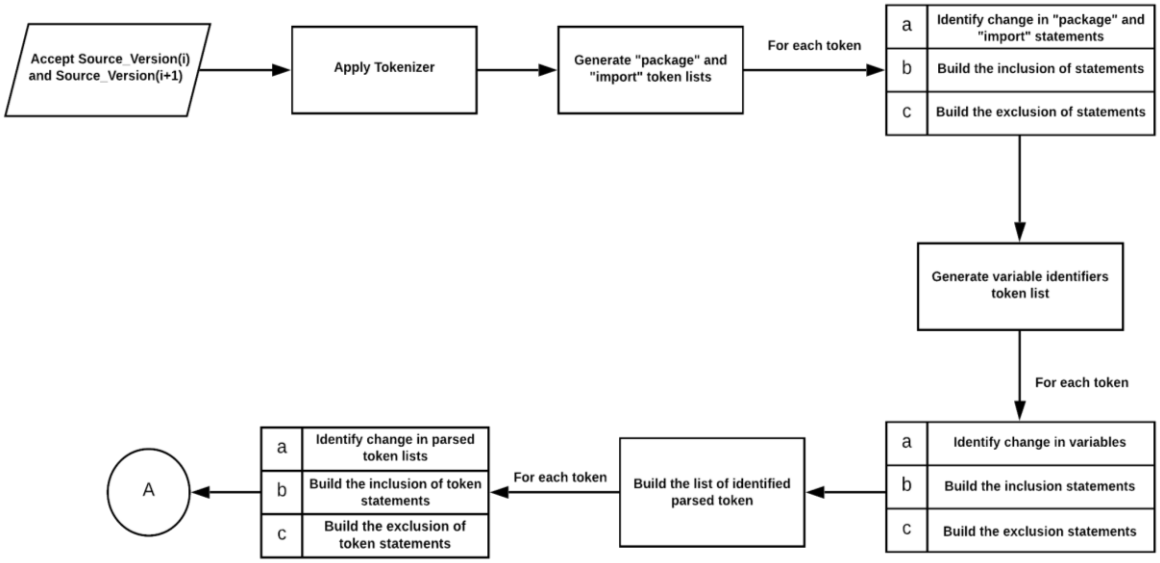
2.5. Performance monitoring

In a robotized situation, execution checking is critical, and there are a few instruments to help. Before the company picks the developer's instruments, the company needs to recognize the key measurements the company needs to screen. Agile and DevOps serve distinct functions: several conventional DevOps clinics like automated build and evaluation, continuous integration, and also continuous shipping started from the planet, that moves (informally) into the 1990s, and also officially to 2001. Agile could be looked at addressing communication gaps between clients and programmers, whereas DevOps addresses differences between programmers and IT operations/infrastructure. Additionally, DevOps has given attention to the installation of developed applications, while it's developed via Agile or alternative systems.

While continuous monitoring is cantered on automating the procedures in computer software shipping, DevOps additionally targets the organizational shift to encourage great cooperation between the various purposes required.

3. PROPOSED CHANGE BASED VM REGION SECURITY ALGORITHM

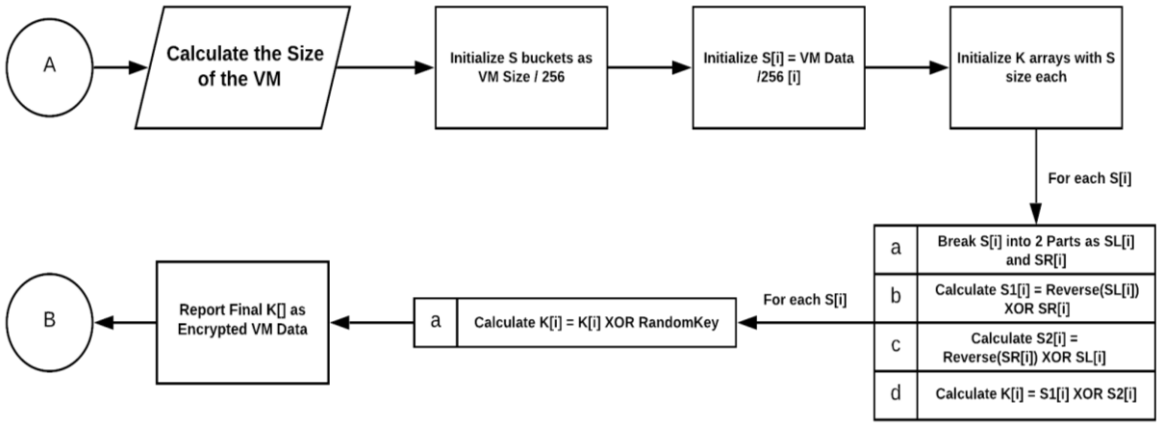
The proposed algorithms are elaborated. The proposed algorithm is built into three parts as change detection in the source code during the DevOps process, encryption of the changed part and decryption of the total VM data. The first part of the algorithm is furnished here:



Algorithm Flow - 1: Source Code Change Detection Algorithm (SCCD)

The proposed SCCD algorithm is designed to detect the changes in three major parts. Firstly, include and package statements to denote the dependency changes. The include and package statements are generally used to support the application code with external or internal reusable codes, and also any changes in the code dependencies. Secondly, the identifiers in the code blocks are used to do functional works in the code. Often the loops or the control statements. Any change in the identifiers can significantly demonstrate a change in the functionalities. Finally, the tokens are used in the source code for maintaining the data. Any changes in the tokens can signify the change in the data flow process in the source code and can identify the way data is changing. The purpose of the SCCD algorithm is to identify the changes in the source code during the DevOps process and only apply security algorithms for the changed parts to reduce the time complexity of the complete process.

The second part of the algorithm is furnished here:

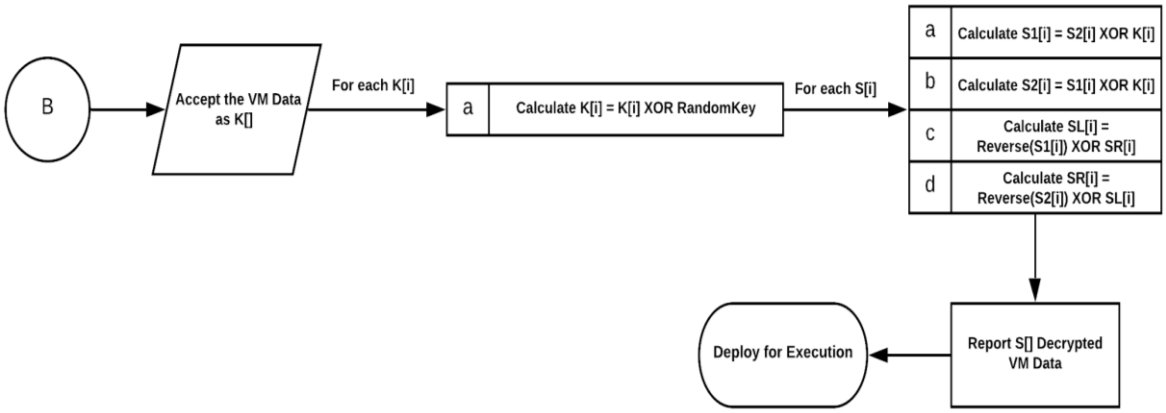


Algorithm Flow - 2: Variable Length VM Encryption Algorithm (VLVE)

The proposed VLVE algorithm is proven to be one of the best algorithms to encrypt the virtual machine data as the virtual machine space calculations are done based on the total blocks allocated on the cloud server. Nevertheless, the virtual machines can be configured to grow based on an increase in the

application or shrink based on the decrease in the application size, which is often observed during any DevOps process. The standard parallel algorithms can not take advantage of this phenomenon, thus cannot cater to the best needs for VM security. Whereas, the proposed variable-length algorithm divides the virtual machines into multiple blocks and further applies the novel security algorithm. This improvement over the other algorithms improves the space complexity of the security algorithms.

The third part of the algorithm is furnished here:



Algorithm Flow - 3: Variable Length VM Decryption Algorithm (VLVD)

Regardless to mention, the purpose of the VLVD algorithm is to reverse the VLVE algorithm. The proposed algorithms are combined into a single process flow and the combined process flow is discussed in the next section of the work.

4. PROPOSED SECURITY FRAMEWORK

In this section of the work, the proposed framework is elaborated with all the function component descriptions as shown in Figure 3. The working flow of the proposed framework is elaborated here. The Code Repository is configured as part of the DevOps process. A source code repository is a file and web facilitating location where a lot of source code, for programming or web pages, is kept, either freely or secretly. They are regularly utilized by open-source programming ventures and other multi-engineer activities to deal with different adaptations. They enable engineers to submit patches of code in a composed manner. Frequently these sites bolster form control, bug following, discharge the board, mailing records, and wiki-based documentation. The code pre-processor as part of the repository manages the version control. Once the pre-processor analyses the versions and selects the most recent two versions from the repository, the version files are given as inputs to the SCCD algorithm module. The change recognition systems endeavor to identify and find regions of source code, which have changed between at least two perceptions of a similar code block. These progressions can be of various kinds, with various causes, and of various lengths. These permit to recognize various types of dependencies, functional, and data flow changes. The security algorithm, proposed in this work, is applied to the changed code, which reduces the time complexity to a greater extend.

Further, once the change is detected, the changed part of the Code, which is to be deployed on the virtual machines are only encrypted using the proposed VLVE algorithm block. Any encryption is the process of encoding a VM data or information in such a way that only authorized parties can access it and those who are not authorized cannot. Encryption does not itself prevent interference but denies the intelligible content to a would-be interceptor. The variable-length VLVE algorithm can identify the block size and can apply encryption to the selected part of the virtual machine data, which reduces the time complexity further.

Furthermore, as part of the DevOps process, the deployment descriptors are prepared for automation in the deployment and provisioning process. A deployment descriptor alludes to a setup document for an ancient rarity that is sent to some compartment/motor. In the Java Platform, Enterprise Edition, a deployment descriptor depicts how a part, module, or application ought to be sent. It guides a deployment device to send a module or application with explicit compartment alternatives, security settings, and depicts explicit design necessities. XML is utilized for the punctuation of these deployment descriptor records.

It natural to realize that the virtual machine deployments are not static, rather vary based on the load balancing strategies of the cloud service provider. Cloud load balancing is a sort of load balancing that is performed in cloud registering. Cloud load balancing is the way toward dispersing workloads over various

figuring assets. Cloud load balancing diminishes costs related to archive the executives' frameworks and amplifies the accessibility of assets. It is a kind of load balancing and not to be mistaken for Domain Name Service (DNS) load balancing. While DNS load balancing utilizes programming or equipment to play out the capacity, cloud load balancing utilizes administrations offered by different PC organize organizations. Because of this module in the proposed framework, the source and destination nodes are identified, and the encrypted virtual machines are migrated from the source nodes to the destination nodes. Once the virtual machines are delivered to the destination node, the VLVD algorithm module reverses the decryption process and the virtual machines are provisioned with all dependencies by the VM Starter module.

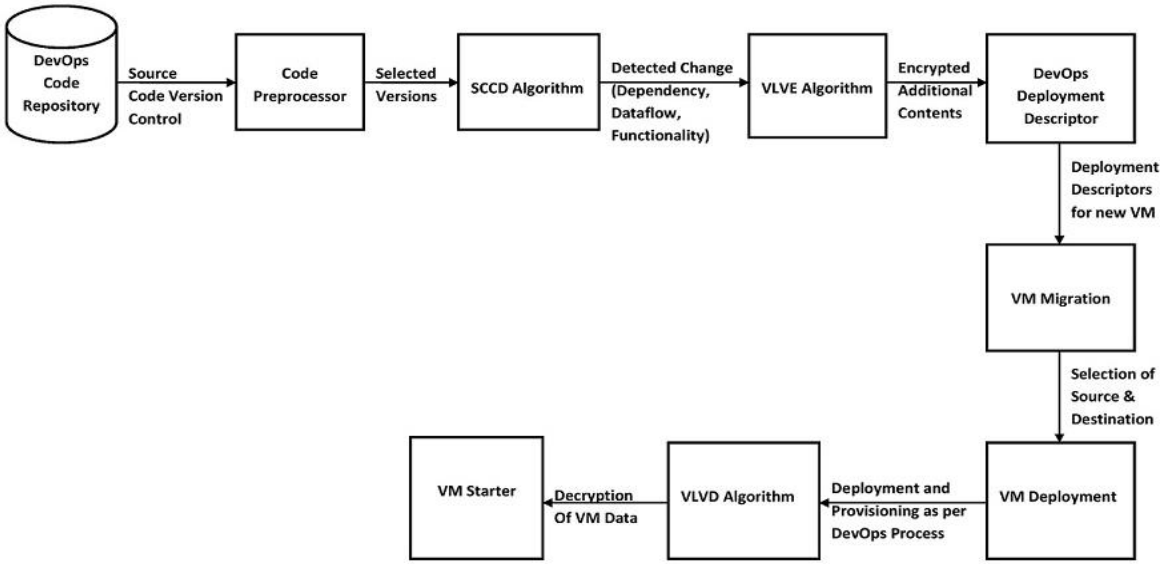


Figure 3. Proposed framework for secure service deployment

5. RESULTS AND DISCUSSION

The results obtained from the proposed framework are highly satisfactory. In this section the results are furnished as initial VM allocation results, VM dependency and code size change tracing, the encryption time and finally the decryption time.

5.1. Initial VM allocation

The virtual machines for the cloud services are allocated on cloud service providers. The initial allocation status is furnished here as shown in Table 1. The results are analysed graphically here as shown in Figure 4.

Table 1. VM Initial allocation

Test Run Number	Provisioning VM Name	Application Size (GB)	Data Size (GB)
1	TestData1.vm	36	35
2	TestData2.vm	36	35
3	TestData3.vm	36	35
4	TestData4.vm	36	35
5	TestData1.vm	36	35
6	TestData2.vm	36	35
7	TestData3.vm	36	35
8	TestData4.vm	36	35
9	TestData1.vm	36	35
10	TestData2.vm	36	35
11	TestData3.vm	36	35
12	TestData4.vm	36	35

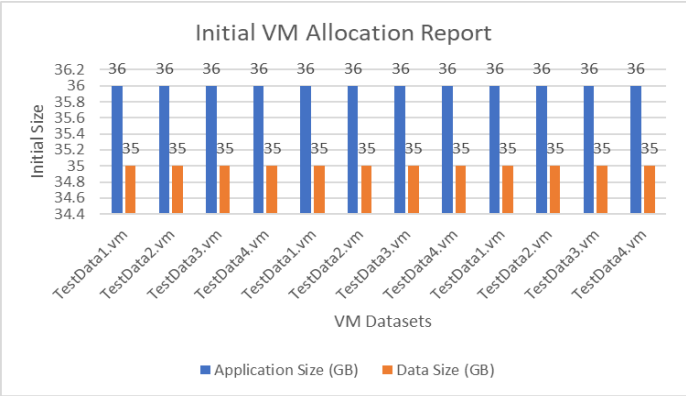


Figure 4. Initial VM allocation

The virtual machines are allocated to IBM Bluemix cloud providers. Virtual servers are adaptable and accompanied devoted centre and memory designations. They are an extraordinary alternative if developers are searching for figure assets, that can be included minutes, with access to highlights like picture layouts. The hypervisor is completely overseen by IBM Cloud, and developers can perform design and the broad undertakings by utilizing both the IBM Cloud client entry and the API. Virtual servers are sent to the equivalent VLANs as physical servers, enabling developers to spread outstanding tasks at hand crosswise over virtual servers and uncovered metal servers while looking after interoperability. Virtual servers are completely adjustable when developers order them, with choices to scale up as their register needs develop.

5.2. Change detection results

Secondly the source code dependency and the functional change results are analysed. The change results are received from the SCCD algorithm. The results are furnished here as shown in Table 2. The results are analysed graphically here as shown in Figure 5.

Table 2. Change detection

Test Run Number	Provisioning VM Name	Changed Dependency Size (GB)	Changed Code Size (MB)
1	TestData1.vm	1547	52
2	TestData2.vm	975	14
3	TestData3.vm	2504	36
4	TestData4.vm	4087	67
5	TestData1.vm	1547	27
6	TestData2.vm	975	18
7	TestData3.vm	2504	124
8	TestData4.vm	4087	52
9	TestData1.vm	1547	56
10	TestData2.vm	975	24
11	TestData3.vm	2504	51
12	TestData4.vm	4087	200

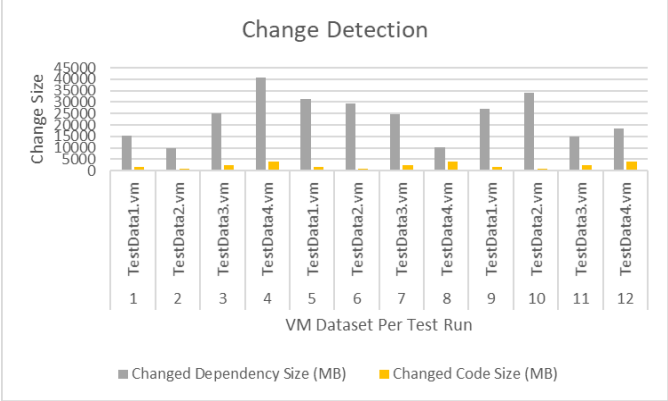


Figure 5. Change detection results

5.3. Encryption time analysis

Further, the encryption time for the changed part of the application with dependencies are analyzed here as shown in Table 3. The results are analysed graphically here as shown in Figure 6.

Table 3. Encryption time analysis

Test Run Number	Provisioning VM Name	Encryption Time (Sec)
1	TestData1.vm	52
2	TestData2.vm	14
3	TestData3.vm	36
4	TestData4.vm	67
5	TestData1.vm	27
6	TestData2.vm	18
7	TestData3.vm	124
8	TestData4.vm	52
9	TestData1.vm	56
10	TestData2.vm	24
11	TestData3.vm	51
12	TestData4.vm	200

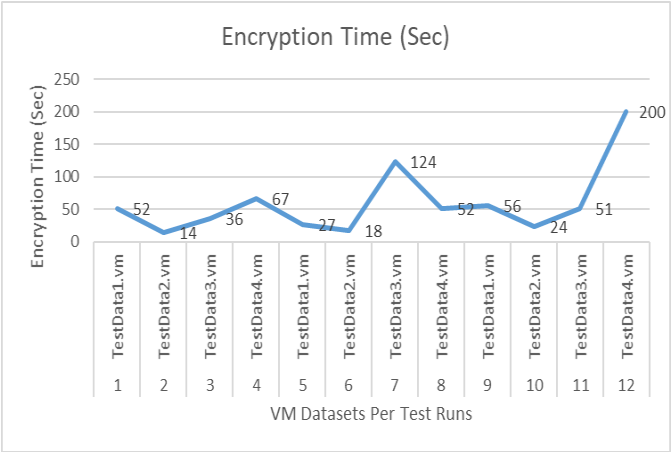


Figure 6. Encryption time analysis

It is natural to realize that, the dependencies are nested in package format. Hence, the time for encryption is not only dependent on the size of the code, rather also depending on the nested levels of the packages.

5.4. Decryption time analysis

Further, the decryption time of the application with dependencies are analysed here as shown in Table 4. The results are analysed graphically here as shown in Figure 7. Furthermore, with the detailed analysis of the results from the proposed framework, in the next section, the comparative analysis is carried out.

Table 4. Decryption time analysis

Test Run Number	Provisioning VM Name	Decryption Time (Sec)
1	TestData1.vm	786
2	TestData2.vm	857
3	TestData3.vm	8
4	TestData4.vm	7
5	TestData1.vm	15
6	TestData2.vm	14
7	TestData3.vm	7
8	TestData4.vm	7
9	TestData1.vm	16
10	TestData2.vm	15
11	TestData3.vm	15
12	TestData4.vm	15

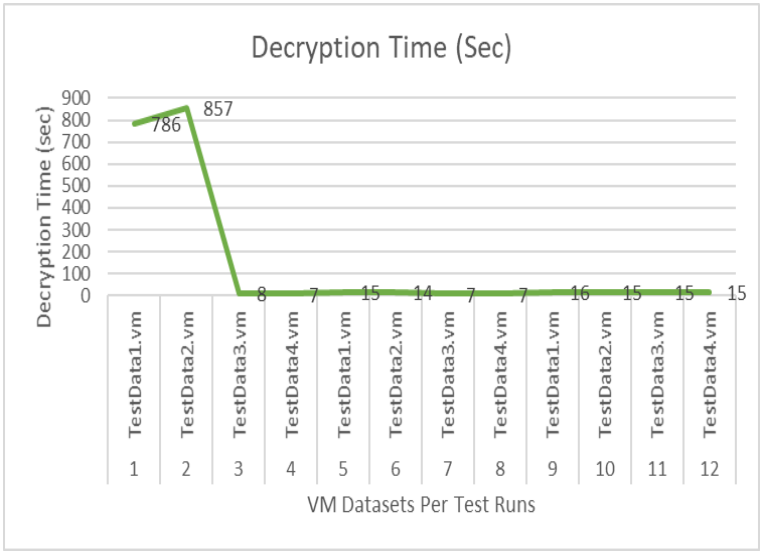


Figure 7. Decryption time analysis

6. COMPARATIVE ANALYSIS

In this section of the work, the comparative analysis with the standard VM encryption and decryption is analysed as shown in Table 5. During the comparative analysis, the standard encryption and decryption algorithms are applied to the virtual machines, and the least time complexity algorithm is selected.

The time complexity for encryption and decryption is stable for the traditional algorithm as the traditional algorithms do not cater to the need for change management and variable-length VM encryption policies as demonstrated by the proposed framework. The advantages, which are reducing the time complexity, are already elaborated in the previous sections of this work.

From the observed results, it is natural to realize that for the encryption a nearly 88.77% and for the decryption a nearly 79.41% improvement over the standard algorithms is achieved. After realizing the results and achieving satisfactory improvements on the time complexity. The results are analysed graphically here as shown in Figure 8.

Table 5. Comparative encryption & decryption time analysis

		Proposed System		Traditional Security Algorithms		Improvements	
Test Run Number	Provisioning VM Name	Encryption Time (Sec)	Decryption Time (Sec)	Encryption Time (Sec)	Decryption Time (Sec)	Encryption Time (%)	Decryption Time (%)
1	TestData1.vm	52	786	535	713	90.28	-10.24
2	TestData2.vm	14	857	535	713	97.38	-20.20
3	TestData3.vm	36	8	535	713	93.27	98.88
4	TestData4.vm	67	7	535	713	87.48	99.02
5	TestData1.vm	27	15	535	713	94.95	97.90
6	TestData2.vm	18	14	535	713	96.64	98.04
7	TestData3.vm	124	7	535	713	76.82	99.02
8	TestData4.vm	52	7	535	713	90.28	99.02
9	TestData1.vm	56	16	535	713	89.53	97.76
10	TestData2.vm	24	15	535	713	95.51	97.90
11	TestData3.vm	51	15	535	713	90.47	97.90
12	TestData4.vm	200	15	535	713	62.62	97.90

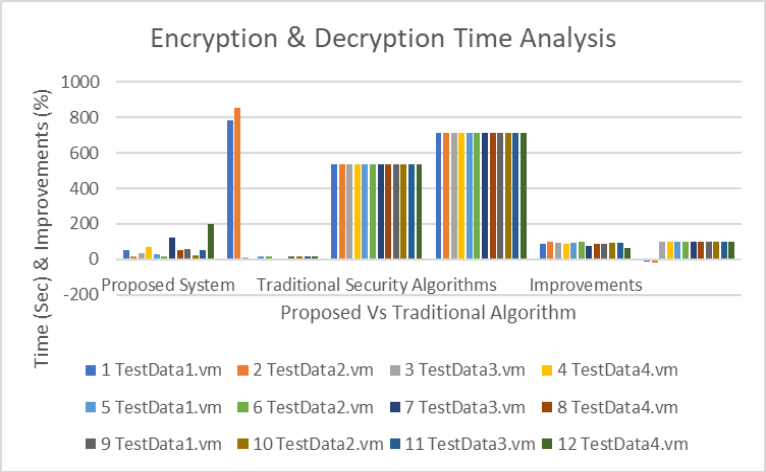


Figure 8. Encryption and decryption time comparative analysis

7. CONCLUSION

The security is the vital concern of the cloud-based application for decades, but several research attempts are being forced now. The security solutions provided by various researchers are criticized by the research communities because of the higher time complexities and for less secure. The major drawbacks of these researches are to ignore the modern software application development strategies, change management in the source code, and finally the growing or shrinking volume of the virtual machines, responsible for application hosting and used for migration.This proposed framework identifies the advantages of detecting change management, produced by the DevOps process, as a scope for reducing the time complexity. Also, the virtual machine fundamental capabilities as variable length virtual machines can also help in reducing the time complexity further. Thus, this work formulates a novel framework to utilize the benefits from these two aspects of cloud-based applications and demonstrates significant reduction in time complexity without compromising the security strengths for making the cloud application even more responsive and highly secure.

8. FUTURE ENHANCEMENT

The data confidentiality, less effort is paid to protect users' identity privacy during those interactive protocols. Users' identities, which are described with their attributes, are generally disclosed to key issuers, and the issuers issue private keys according to their attributes. But it seems natural that users are willing to keep their identities secret while they still get their private keys.

REFERENCE

[1] P. J. Bruening, B. C. Treacy, "Privacy & security law report: privacy", 2009.
[2] M. Ouedraogo, H. Mouratidis, "Selecting a cloud service provider in the age of cybercrime", *Comput. Security Special Issue Cybercrime Digital Economy*, vol. 38, pp. 3-13, 2013.
[3] Khajeh-Hosseini, I. Sommerville, J. Bogaerts, P. Teregowda, "Decision support tools for cloud migration in the enterprise", *Proc. IEEE Int. Conf. Cloud Comput*., pp. 541-548, 2011.
[4] P. Jamshidi, A. Ahmad, C. Pahl, "Cloud migration research: A systematic review", *IEEE Trans. Cloud Comput*., vol. 1, no. 2, pp. 142-157, 2013.
[5] C. Kalloniatis, H. Mouratidis, M. Vassilisc, S. Islam, S. Gritzalis, E. Kavaklif, "Towards the design of secure and privacy-oriented information systems in the cloud: Identifying the major concepts", *Comput. Standards Interfaces*, vol. 36, no. 4, pp. 759-775, 2014.
[6] S. Pearson, A. Benameur, "Privacy security and trust issues arising from cloud computing", *Proc. 2nd IEEE Int. Conf. Cloud Comput. Technol. Science*, pp. 693-702, 2010.
[7] Federal Risk and Authorization Management Program (FedRAMP). https://fedramp.gov/
[8] M. Theoharidou, N. Papanikolaou, S. Pearson, D. Gritzalis, "Privacy risk security accountability in the cloud", *Proc. IEEE 5th Int. Conf. Cloud Comput. Technol. Sci*., pp. 177-184, 2013.
[9] H. Takabi, J. Joshi, G. Ahn, G. Security, "Privacy challenges in cloud computing environments", Nov./Dec. 2010.
[10] M. Gregg, "10 Security Concerns for Cloud Computing". 2010.

[11] M. Mulazzani, S. Schrittwieser, M. Leithner, M. Huber, E. Weippl, "Dark clouds on the horizon: Using cloud storage as attack vector and online slack space", *Proc. 20th USENIX Conf. Security*, pp. 5, 2011.
[12] S. De Capitani di Vimercati, S. Foresti, S. Jajodia, S. Paraboschi, P. Samarati, "Over-encryption: Management of access control evolution on outsourced data", *Proc. 33rd Int. Conf. Very Large Databases*, pp. 123-134, 2007.
[13] D. G. Rosado, R. Gomez, D. Mellado, E. Fernández-Medina, "Security analysis in the migration to cloud environments", *Future Internet*, vol. 4, no. 2, pp. 469-487, 2012.
[14] M. Klems, J. Nimis, S. Tai, "Do clouds compute? A framework for estimating the value of cloud computing designing e-business systems", *Proc. 7th Workshop E-Bus. Syst. Markets Services Netw*., pp. 110-123, 2009.
[15] Singh, S. P., Nayyar, A., Kaur, H., Singla, A. "Dynamic Task Scheduling using Balanced VM Allocation Policy for Fog Computing Platforms". *Scalable Computing: Practice and Experience*, vol. 20, no. 2, pp. 433-456, 2019.
[16] Kaur, A., Gupta, P., Singh, M., Nayyar, A. "Data Placement in Era of Cloud Computing: a Survey, Taxonomy and Open Research Issues". *Scalable Computing: Practice and Experience*, vol. 20, no. 2, pp. 377-398, 2019.
[17] Singh, P., Gupta, P., Jyoti, K., Nayyar, A. "Research on Auto-Scaling of Web Applications in Cloud: Survey, Trends and Future Directions". *Scalable Computing: Practice and Experience*, vol. 20, no. 2, pp. 399-432, 2019.
[18] Singh, S. P., Nayyar, A., Kumar, R., Sharma, A. "Fog computing: from architecture to edge computing and big data processing". *The Journal of Supercomputing*, vol. 75, no. 4, pp. 2070-2105, 2019.
[19] Nayyar, A. "Private virtual infrastructure (pvi) model for cloud computing". *International Journal of Software Engineering Research and Practices*, vol. 1, no. 1, pp. 10-14, 2011.
[20] Nayyar, A. "Handbook of Cloud Computing: Basic to Advance research on the concepts and design of Cloud Computing". *BPB Publications*, 2019.
[21] Nayyar, A. "Interoperability of cloud computing with web services". *International Journal of Electro Computational World & Knowledge Interface*, 2011.
[22] Swathy, R., Vinayagasundaram, B., Rajesh, G., Nayyar, A., Abouhawwash,M., & Abu Elsoud, M. "Game theoretical approach for load balancing using SGMLB model in cloud environment". *Plos one*, vol. 15, no. 4, 2020, e0231708.
[23] G.Balakrishna and Moparthi Nageshwara Rao," "ESBL: Design and Implement A Cloud Integrated Framework for IoT Load Balancing" *International Journal Of Computers Communications & Control* ISSN 1841-9836, e-ISSN 1841-9844, vol. 14, no. 4, pp. 459-474, 2019.
[24] P Ravinder Rao and Dr. V.Sucharita, "A Framework to Automate Cloud based Service Attacks Detection and Prevention" *International Journal of Advanced Computer Science and Applications (IJACSA)*, vol. 10, no. 2, 2019. http://dx.doi.org/10.14569/IJACSA.2019.0100232
[25] G. Balakrishna* and Nageswara Rao Moparthi, "An Optimal IoT Device Placement Strategy for Agro-IoT using Edge Computing", *Recent Advances in Computer Science and Communications*, vol. 13, no. 1, 2020.