iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 8 **Issue:** VII **Month of publication:** July 2020

DOI: http://doi.org/10.22214/ijraset.2020.7069

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 8 Issue VII July 2020- Available at www.ijraset.com

Libraries and Impact of Implementing Cloud Native Applications in replacement to Premise Applications

Dr. Mohammed Imtiaz Ahmed[1], Mr. Vijayendra Singh Rana[2], Dr. Naresh Puri[3]
[1]*University Librarian, Pandit Deendayal Upadhyay Memorial Health Science & Ayush University of Chhattisgarh*
[2]*Assistant Librarian, Pandit Deendayal Upadhyay Memorial Health Science & Ayush University of Chhattisgarh*
[3]*Librarian, Government Arts and Commerce Girls College, Devendra Nagar, Raipur*

Abstract: ***Cloud computing is a well-known worldview to figuring assets. Cloud-native technologies incorporate the application system in a dynamic cloud environment, which may include private, public, or hybrid. It focuses on activity and agility. This paper speaks about implementing cloud-native applications in replacement to on-premise applications. We implement Platform as a Service (PaaS) for Application Streaming through the premise, rather than incorporating the conventional procedure of setting up on-premise application software on each system. The paper also tries to provide an insight into the advantages of a native-cloud application over an on-premise one.***

Keywords: PaaS - Platform as a Service; CNA - Cloud Native Applictaions; VM - Virtual Machine; API – Application Programming Interface; AWS – Amazon Web Services; VPC – Virtual Private Cloud

I. INTRODUCTION

Cloud computing is increasingly a widespread criterion for access to computing resources. Cloud service providers prefer to provide services that can get categorized into three groups in practice: service software, service network, and service infrastructure. The objective is to deploy a PaaS (Platform as a Service) solution that delivers on-demand usage of powerful and graphic-intense applications through a web browser. We also use a Virtual Private Cloud (VPS). For this purpose, we deploy the AppStream 2.0 provided by AWS. Amazon Web Services (AWS) is a division of Amazon that offers on-demand cloud hosting tools and APIs to customers, businesses and governments on a pay-as-you-go basis. AWS technology is implemented in server farms around the world and maintained by the Amazon subsidiary. Amazon Web Services (AWS) is a division of Amazon that recommendations on-demand cloud encouraging gadgets and APIs to customers, associations and governments on a premise of the pay-as-you-go. AWS innovation gets actualized in server cultivates the world over and kept up by the Amazon auxiliary. AppStream 2.0 is an overseen administration that lets clients be able to stream work area applications over the cloud to any HTML5-skilled program on a PC or PC, regardless of whether it runs Windows, Linux, macOS or Chrome OS. A few organizations around the world have moved their innovation improvement practices and procedures to the cloud through PaaS to change the conventional advancement draws near and to get the full profit by the condition of the adaptable craftsmanship administrations. Users spend much money on buying a powerful computing unit for their requirements. Instead, we can run the same applications at a cheaper cost on an online platform. This methodology also reduces the need for periodic up-gradation of computing components in the users' side. The significant of the cloud-based virtual machine is developing with new advancement in virtualization innovation. The gushing of huge estimated virtual machines tends to the issue of restrictive privileges of the product and cost and support issues on the customer end. It is not doable to stream enormous estimated virtual apparatuses in a single go. Indeed, even with better system associations, any interference can result in download disappointment. PaaS merchants face difficulties in proficiently offering types of assistance with the development of their contributions. PaaS sellers use compartments as methods for facilitating Apps. The PaaS Use case and the present selection of Container-based PaaS models with the current sellers. We take a gander at how every one of these usage handles Process, FileSystem, and Namespace detachment. The remarkable highlights of every compartment are comparable, and some of them reuse base Linux Container execution or contrast from it. The on-request prefetching of virtual packs significantly decreases the transmitting time and transmission capacity utilization of the system. The weight on the handling intensity of inside hubs likewise decreases with the decrease in the size of information. Applications require a substantial foundation for consistent types of assistance to encourage clients. The customary customer server model is efficient for mixed media benefits because of its bottleneck and a predetermined number of simultaneous client's help.

©IJRASET: All Rights are Reserved

II. RELATED WORKS

A. Gannon Dennis, Barga Roger, Sundaresan Neel; 2017; Cloud-Native Applications

In this article, there includes a portrayal of fundamental properties of valuable cloud applications including dynamic adaptability, extraordinary adaptation to internal failure, consistent upgradeability and support and security. To make it conceivable to fabricate applications that meet these necessities, we portray the microservice engineering and serverless registering establishment that are integral to the cloud-native structure.

B. Eweoya Ibukun, Daramola Olawande; 2015; A systematic literature review of mobile cloud computing

This paper conducts an orderly survey, offering organised, deliberate, and thorough way to deal with the comprehension of the pattern of research in the domain, and other explored issue get introduced. The primary target of such an examination is providing dependable manual, and thus far, the issues of design, setting mindfulness and information the board have been averagely inspected. In contrast, inquiries on activities, end-clients, administration and applications have gotten a great deal of consideration in writing.

C. Vassil Roussev, Shane McCulley; 2016; Forensic analysis of cloud-native artifacts

In this work, the idea of dissecting cloud-local computerised antiquities information questions that keep up the steady condition of web/SaaS applications is presented. Utilising Google Docs as a contextual investigation, the way that such antiquities can have a unique structure was illustrated. Further, cloud-local items have no normalised outside portrayal, which brings up issues concerning their drawn-out protection and understanding.

D. Pahl; 2015; Containerization and the PaaS Cloud

This article examines the necessities that emerge from encouraging applications through disseminated multi-cloud platforms. Besides displaying benefits over traditional virtual machines in Cloud, holders are particularly crucial for Platform as a Service (PaaS) clouds to oversee and organize applications through compartments as an application bundling system.

E. F. Paraiso, N. Haderer, P. Merle, R. Rouvoy and L. Seinturier; 2012; A Federated Multi-cloud PaaS Infrastructure

In this paper, the combined multi-cloud PaaS foundation for tending to confronted difficulties was introduced. This system relies upon three foundations: an open help model used to structure and actualise both our multi-cloud PaaS and the SaaS applications running on it, a configurable plan of the joined PaaS, and some establishment administrations for directing both our multi-cloud PaaS and the SaaS applications.

F. Dimpi Rani, Rajiv Kumar Ranjan; 2014; A Comparative Study of SaaS, PaaS and IaaS in Cloud Computing.

This paper illustrated a study in cloud processing services, concentrating on the extensive rundown services gave by driving organisations. It depicts the relative investigation of cloud services. Cloud offers agreeable situations toward its clients and different services, which includes Infrastructure as a Service(IaaS), Platform as a service(PaaS), and Software as a service(SaaS). Analysts, despite everything, have more work to do; the paper was considered as a beginning stage distinguishing open doors for future research.

G. S. García-Gómez et al.; 2012; 4CaaSt: Comprehensive Management of Cloud Services through a PaaS

The task targets on building up a PaaS system that empowers the adaptable definition, showcasing, sending and board the Cloud-based services and applications. The noteworthy advancements proposed are the plan and its lifecycle the board, a one-stop-look toward Cloud services and the PaaS level asset the board including versatility. It additionally gives an arrangement of prepared to utilise Cloud-local services and Cloud-mindful worker advancements.

H. B. Martínez-Pérez, I. de la Torre-Díez, M. López-Coronado and J. J. P. C. Rodrigues; 2015; Are Mobile Health Cloud Apps Better Than Native?

This paper plays out a similar investigation of the highlights of local and versatile cloud applications for social insurance. With that reason, a scholarly and non-scholastic audit of highlights, preferences and drawbacks of the two kinds of applications were created. It was reasoned that in spite of the versatile cloud applications offer noteworthy focal points, which conquer the ones of local applications in certain situations, there are different situations where local applications are the central arrangement. Consequently, the two kinds need to exist together for some time.

©IJRASET: All Rights are Reserved

I. *Kamateri E. et al.; 2013; Cloud4SOA: A Semantic-Interoperability PaaS Solution for Multi-cloud Platform Management and Portability.*

This paper presents an answer which is a versatile way to deal with semantically interconnecting different PaaS contributions across various Cloud suppliers that share a similar innovation. The plan of the arrangement, broadly introduced in this work, involves a lot of interlinked teaming up programming parts and models to furnish engineers and platform suppliers with a few centre abilities: matchmaking, the executives, checking and relocation of uses.

J. *Pahl and H. Xiong; 2013; Migration to PaaS clouds - Migration process and architectural concerns*

The paper viewed programming design and programming worries with regards to relocation to PaaS arrangements, for example, the change of platform frameworks from on-reason to cloud arrangements. Best-practice approaches in cloud-mindful coding as examples and details these as a relocation procedure were researched. While balanced mappings of Software from on-reason to cloud platforms are conceivable, statelessness and information externalisation from stateful meetings and applications rise as arrangements if cloud advantages, for example, flexibility and execution are focused.

K. *D. Gannon, R. Barga, and N. Sundaresan; 2017; Cloud-Native Applications*

This paper portrayed cloud-native ideas and represented them with models. It also looked at the innovation slants that may give us a thought regarding the eventual fate of cloud applications. The central properties that many cloud-native applications share for all intents and purpose have been spoken about. After portraying, the depiction was made on how these properties arise out of the specialised plan designs.

L. *Giove F., Longoni D., Shokrolahi Yancheshmeh M.; Ardagna D. and Di Nitto E.; 2013; An Approach for the Development of Portable Applications on PaaS Clouds.*

In this paper, a methodology has been proposed for the advancement of compact applications for Platform as a Service (PaaS) frameworks. This framework depends on a Java library uncovering a seller free API that gives a theoretical intermediation layer to the essential middleware benefits regularly offered by PaaS frameworks. The present form of library underpins the movability of uses across Java platforms for Google App Engine and Windows Azure. The analyses exhibit that such disintegration is not critical.

M. *N. Kratzke and R. Peinl; 2016; ClouNS - a Cloud-Native Application Reference Model for Enterprise Architects*

Endeavour designers ought to know that cloud-local applications are helpless against merchant lock-in. The paper examined cloud-local application plan standards, open cloud specialist co-ops, and modern cloud norms. All outcomes demonstrate that most cloud administration classes appear to cultivate merchant lock-in circumstances which may be particularly dangerous for significant business structures. The paper also presented a reference model for cloud-local applications that depends just on a little subset of very much normalised IaaS administrations.

N. *I. Pelle, J. Czentye, J. Dóka and B. Sonkoly; 2019; Towards Latency Sensitive Cloud Native Applications: A Performance Study on AWS.*

A generally utilised and adaptable cloud platform, Amazon Web Services (AWS), and uncover the postpone attributes of crucial segments and administrations which sway the general execution of delicate inactivity applications is addressed. To begin with, the characterisation by a point by point estimation approach for CaaS/FaaS (Container/Function as a Service) platforms, explicitly for AWS is made. Second, a complete investigation is done of AWS segments concentrating on postponing qualities. Third, the endeavour to change an automaton control application to the platform and examines the presentation on the present framework.

O. *L. Yang, J. Cao, S. Tang, T. Li and A. T. S. Chan; 2012; A Framework for Partitioning and Execution of Data Stream Applications in Mobile Cloud Computing*

Three methodologies have been proposed for versatile cloud applications: 1) stretching out the entrance to cloud administrations to cell phones; 2) empowering cell phones to work cooperatively as cloud asset suppliers; 3) increasing the execution of versatile applications on convenient gadgets utilizing cloud assets. In this paper, the centre is around the third methodology in supporting versatile information stream applications. All the more explicitly, the paper studies the calculation dividing, which targets upgrading the parcel of an information stream application among portable and cloud with the end goal that the application has most significant speed/throughput in handling the spilling information.

©IJRASET: All Rights are Reserved

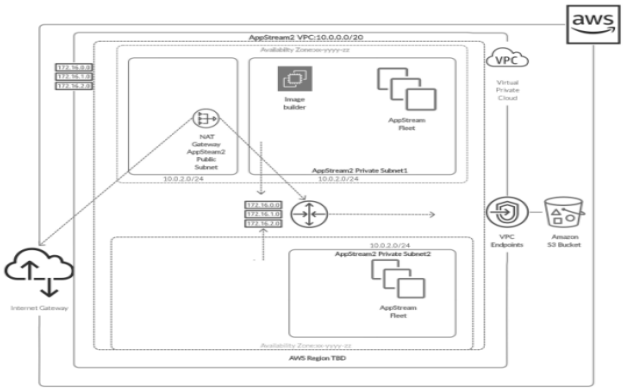


Fig. 1.Architecture Model of AppStream

III. PROPOSED SYSTEM

Clients can begin AppStream 2.0 gushing meetings by utilizing an internet browser or by utilizing the AppStream 2.0 customer application for an upheld gadget. On the off chance that they utilize an internet browser, they can utilize an HTML5-fit program, for example, Google Chrome, Mozilla Firefox, Safari, Microsoft Edge, and Microsoft Internet Explorer variant 11 or later. No program augmentations or modules are required to utilize an internet browser for AppStream 2.0 meetings.

The AppStream 2.0 customer is a local application that intends for clients who require extra usefulness during their AppStream 2.0 spilling meetings. This customer is reasonable for clients who require support for different screens, utilize their USB gadgets with applications gushed through AppStream 2.0, and use alternate console ways during their spilling meetings.

IV. MATHEMATICAL PROOF

Jan –> Dec is a named range – 12 Time Units
Microsoft RDS SAL = $0.44 / user / month
RDS Cost per Year = 20 labs * 60 users *12 months * $0.44 /user/month = $36060
Graphics Instance = $0.5/hour * 24 * 365 * 5 instances = $21900
Thin Client = $150*1200 = $180,000
Therefore, the total cost of using appstream for an institution is,
Total Cost aws = $237960
But in local App streaming,
Total cost = (No of computers/ device used * cost of the computer) + Maintenance cost = 1200 * $570 + $86500
Therefore, the total cost for local App streaming, Total Costlocal = 686500
Profit = Total Costaws - Total Costlocal = $686500 - $237960 = $448540
Profit within five years is ~$500000

V. EXPERIMENTAL SETUP

To set up this project, primarily, an AWS account is required. It must also be noted that a broadband Internet connection with at least 1 Mbps bandwidth is active. A browser capable of supporting HTML5 or AppStream Client App is required to run and handle the cloud app.

First, Log onto your AWS account. Select AppStream 2.0. You will be redirected to the dashboard. Here, select Image builder and either choose a preexisting one or build your own from scratch. Second, go to Stacks, and recreate the template. Now, create a Fleet. Third, associate the stack to the fleet to execute it. Finally, select the URL option to generate a client usable URL.

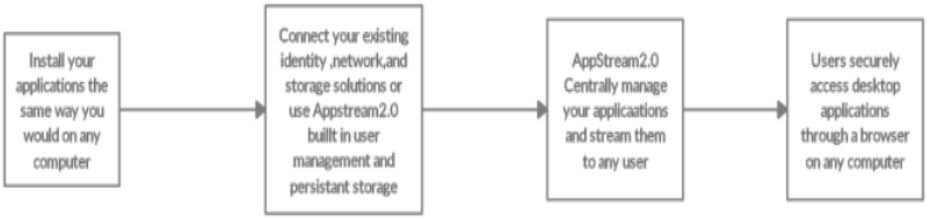


Fig. 2.WorkFlow chart of AppStream 2.0

©IJRASET: All Rights are Reserved

VI. RESULTS AND DISCUSSION

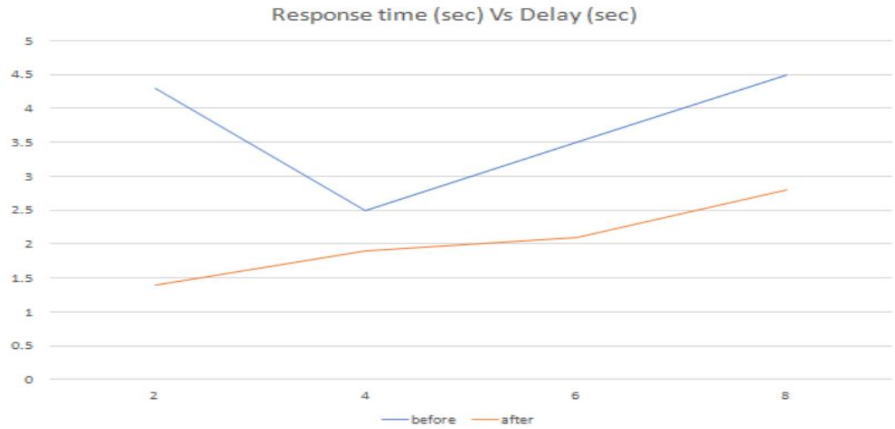


Fig. 3.Response Time vs Delay

After the streaming URL is created and published to users, upon accessing the link, it will ask for a signup, and later the homepage will look as in the following image. Clicking upon any app icon will start the application in the server, and near-real-time streaming to the user's browser is done. Users can plug in their USB drives and can access the apps connected to their laptop/ desktop systems. As we see in this graph, local app streaming getting much latency(delay) since the response time is taking too much time. By using the AppStream 2.0, we can reduce the latency also the time taken to get the response back.

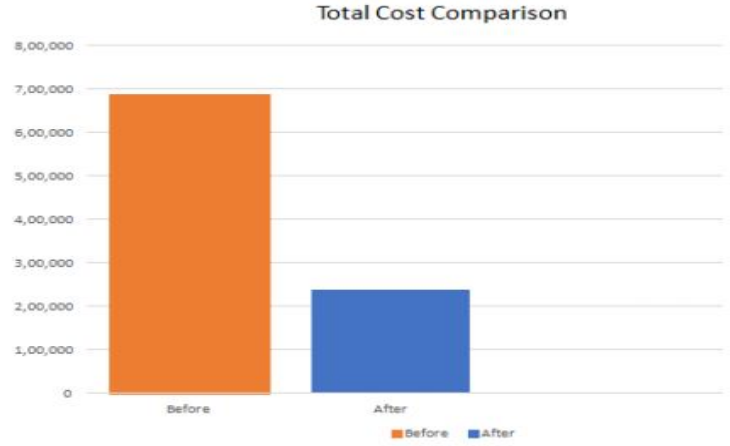


Fig. 4.Total Cost Comparison

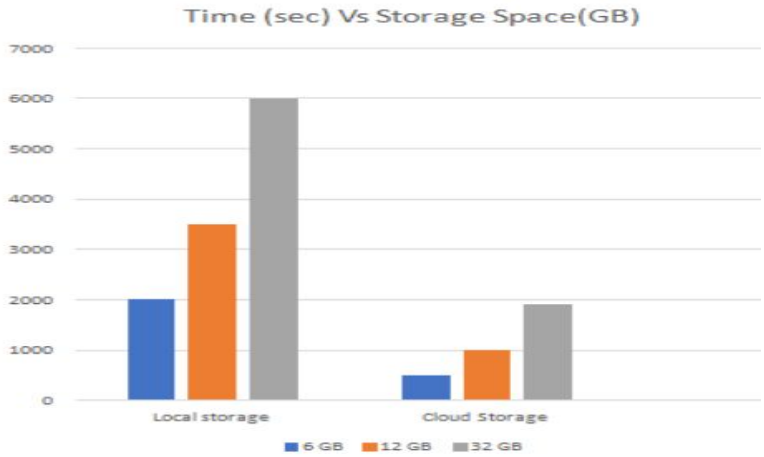


Fig. 5.Time vs Storage Space

©IJRASET: All Rights are Reserved

In this graph, we have compared the results with the storage and time. We have found in both cases disk storage is proportional to the performance time. Except, by using Appstream 2.0, we can produce a more efficient solution than by using local app streaming.

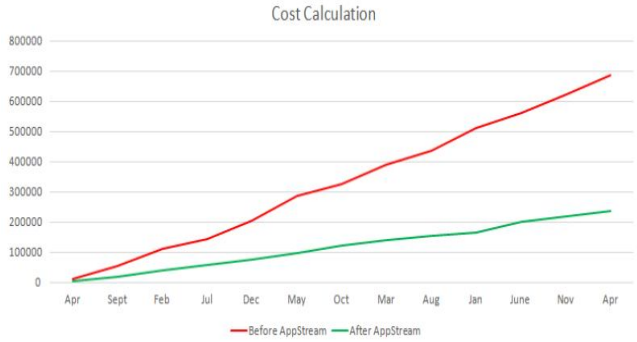


Fig. 6.Cost of System affect by App Stream

By comparing the local app streaming and AWS Appstream 2.0, we concluded that by using AWS Appstream 2.0, we could save the total cost up to five lakhs in 5 years.

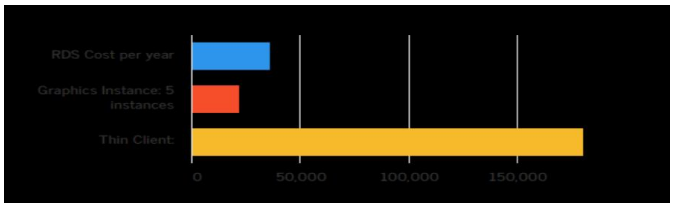


Fig. 7.Cost split up of each module in AppStream

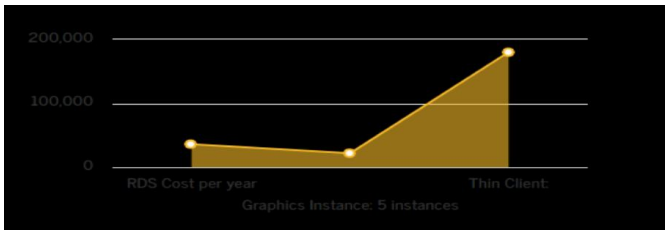


Fig. 8.Cost of using AppStream over 5 years

VII. CONCLUSION

A quick calculation compared the cost of running applications in user's systems vs app streaming on our university campus. Taking cloud computing is comparatively a new technological development into account; it can significantly affect the world. It has numerous advantages that it provides to its clients and businesses. Especially incorporating native applications, as proven, can impact on cost and in return, development. Nevertheless, there are other challenges the cloud computing must overcome. Intended to flourish in a dynamic, virtualized cloud condition, native application frameworks utilized Platform as a Service (PaaS) figure foundation and oversaw administrations. They treat the hidden foundation as dispensable - provisioned in minutes and resized, scaled, moved, or pulverized on request – employing mechanization. However, cloud computing has a proven scope to revolutionize the future.

VIII. FUTURE SCOPE

After In the future works of the project, there is a scope to range out this implementation to the SaaS stage. Software as a Service (SaaS) is an item course model where a pariah provider has application and makes them open to customers over the Internet. SaaS has a differential with respect to costs since it, when in doubt, stays in a norm or multitenant condition where the hardware and programming license costs are low differentiated and the regular model. Not exactly equivalent to the ordinary model, in SaaS the item (application) is starting at now presented and orchestrated. The customer has the advantage of provisioning the server for an event in the cloud, and a few hours, they can have the application arranged for use.

©IJRASET: All Rights are Reserved

IX. ACKNOWLEDGMENT (HEADING 5)

We would like to thank our respective families for the support they aided us with. We would also like to extend our thanks to our university for providing the infrastructure with which our paper was inspired.

REFERENCES

[1] Gannon, Dennis & Barga, Roger & Sundaresan, Neel. (2017). Cloud-Native Applications. IEEE Cloud Computing. 4. 16-21. 10.1109/MCC.2017.4250939.

[2] Eweoya, Ibukun & Daramola, Olawande. (2015). A systematic literature review of mobile cloud computing. 10. 135-152. 10.14257/ijmue.2015.10.12.15.

[3] Vassil Roussev, Shane McCulley, Forensic analysis of cloud-native artifacts, Digital Investigation, Volume 16, Supplement, 2016, Pages S104-S113, ISSN 1742-2876, /10.1016/j.diin.2016.01.013.

[4] C. Pahl, "Containerization and the PaaS Cloud," in IEEE Cloud Computing, vol. 2, no. 3, pp. 24-31, May-June 2015, doi: 10.1109/MCC.2015.51.

[5] F. Paraiso, N. Haderer, P. Merle, R. Rouvoy and L. Seinturier, "A Federated Multi-cloud PaaS Infrastructure," 2012 IEEE Fifth International Conference on Cloud Computing, Honolulu, HI, 2012, pp. 392-399, doi: 10.1109/CLOUD.2012.79.

[6] Dimpi Rani, Rajiv Kumar Ranjan, "A Comparative Study of SaaS, PaaS and IaaS in Cloud Computing," 2014 International Journal of Advanced Research in Computer Science and Software Engineering, Kangara

[7] S. García-Gómez et al., "4CaaSt: Comprehensive Management of Cloud Services through a PaaS," 2012 IEEE 10th International Symposium on Parallel and Distributed Processing with Applications, Leganes, 2012, pp. 494-499, doi: 10.1109/ISPA.2012.72.

[8] B. Martínez-Pérez, I. de la Torre-Díez, M. López-Coronado and J. J. P. C. Rodrigues, "Are mobile health cloud apps better than native?," 2015 IEEE International Conference on Communications (ICC), London, 2015, pp. 518-523, doi: 10.1109/ICC.2015.7248374.

[9] Kamateri E. et al. (2013) Cloud4SOA: A Semantic-Interoperability PaaS Solution for Multi-cloud Platform Management and Portability. In: Lau KK., Lamersdorf W., Pimentel E. (eds) Service-Oriented and Cloud Computing. ESOCC 2013. Lecture Notes in Computer Science, vol 8135. Springer, Berlin, Heidelberg

[10] C. Pahl and H. Xiong, "Migration to PaaS clouds - Migration process and architectural concerns," 2013 IEEE 7th International Symposium on the Maintenance and Evolution of Service-Oriented and Cloud-Based Systems, Eindhoven, 2013, pp. 86-91, doi: 10.1109/MESOCA.2013.6632740.

[11] Giove, F.; Longoni, D.; Shokrolahi Yancheshmeh, M.; Ardagna, D. and Di Nitto, E. (2013). An Approach for the Development of Portable Applications on PaaS Clouds.In Proceedings of the 3rd International Conference on Cloud Computing and Services Science - Volume 1: CI, (CLOSER 2013) ISBN 978-989-8565-52-5, pages 591-601. DOI: 10.5220/0004511605910601

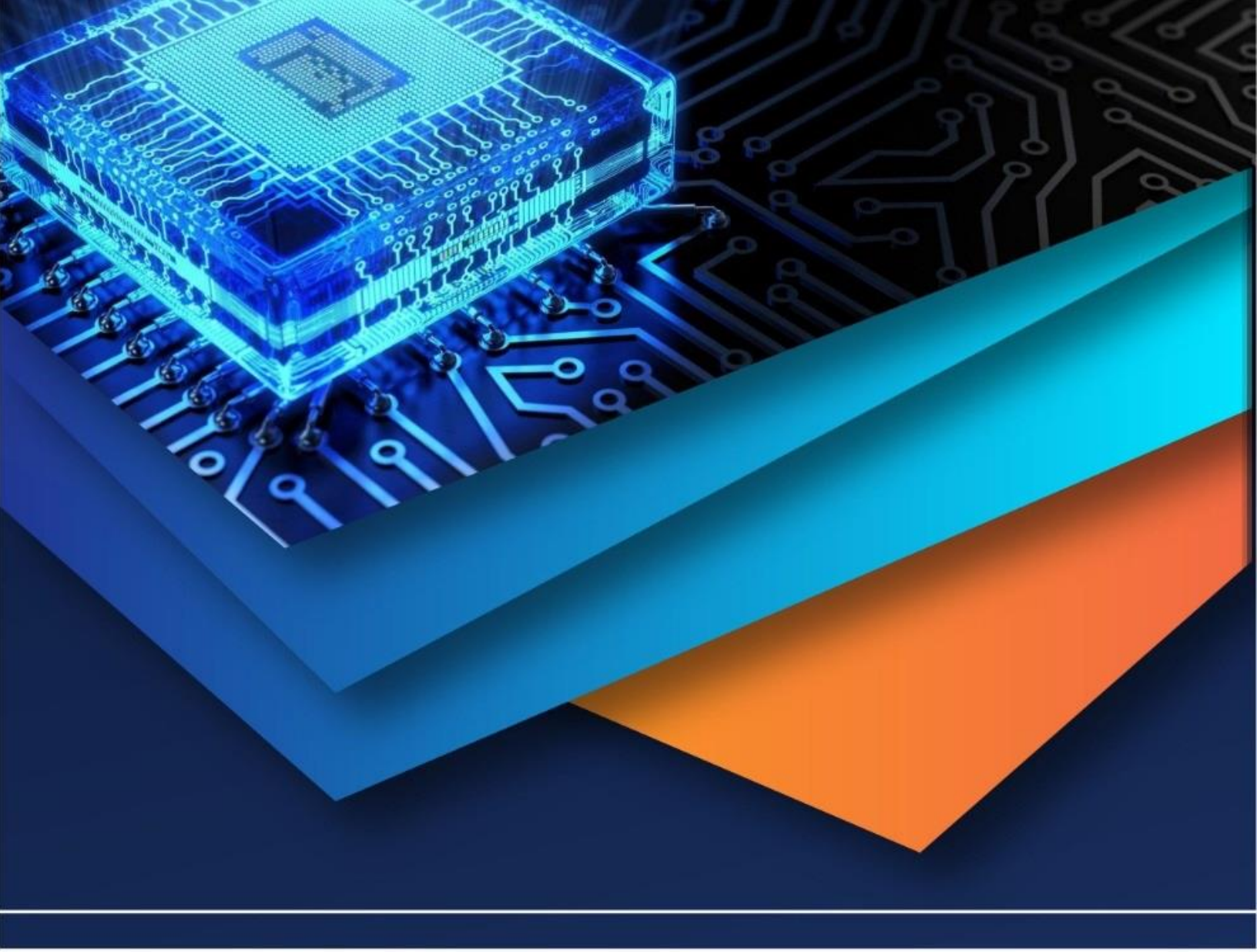
[12] N. Kratzke and R. Peinl, "ClouNS - a Cloud-Native Application Reference Model for Enterprise Architects," 2016 IEEE 20th International Enterprise Distributed Object Computing Workshop (EDOCW), Vienna, 2016, pp. 1-10, doi: 10.1109/EDOCW.2016.7584353.

[13] D. Gannon, R. Barga and N. Sundaresan, "Cloud-Native Applications," in IEEE Cloud Computing, vol. 4, no. 5, pp. 16-21, September/October 2017, doi: 10.1109/MCC.2017.4250939.

[14] I. Pelle, J. Czentye, J. Dóka and B. Sonkoly, "Towards Latency Sensitive Cloud Native Applications: A Performance Study on AWS," 2019 IEEE 12th International Conference on Cloud Computing (CLOUD), Milan, Italy, 2019, pp. 272-280, doi: 10.1109/CLOUD.2019.00054.

[15] L. Yang, J. Cao, S. Tang, T. Li and A. T. S. Chan, "A Framework for Partitioning and Execution of Data Stream Applications in Mobile Cloud Computing," 2012 IEEE Fifth International

©IJRASET: All Rights are Reserved



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)