Cloud DevOps Tools and Environment for Cloud-based Applications

18bce103@nirmauni.ac.in

Maharsh Shah

18bce135@nirmauni.ac.in

Nirav Madhani

Abstract: If we had a world with programmers for the cloud-hosted apps that possessed exceptional technical capability and patience, then we could say that we would have not required the DevOps Environment (a.k.a. Platform-as-a-Service or PaaS). Thus, these developers dream for the usable and reliable DevOps environment that can facilitate the development and operational processes. The DevOps scenario today has applications like Google App Engine, Docker, Kubernetes, Mesos, and others. Hence, we can say that DevOps is the bridge between the all-vivid IT developers and the inflexible cloud systems. In this paper we will be giving comprehensive reviews and examinations to the PaaS solutions across the different tiers of the global Cloud Architecture. Hence, we will see the different philosophies and methodologies that have got the consensus and also the diversities in them. We won't miss to explore the solutions that have brought us to the state-of-the-art full-stack DevOps environment. We expect you, the readers to grab the updates, on-goings and the future of PaaS.

Keywords: Platform-as-a-Service(PaaS);cloudcomputing;DevOps;operation;development; environment

---

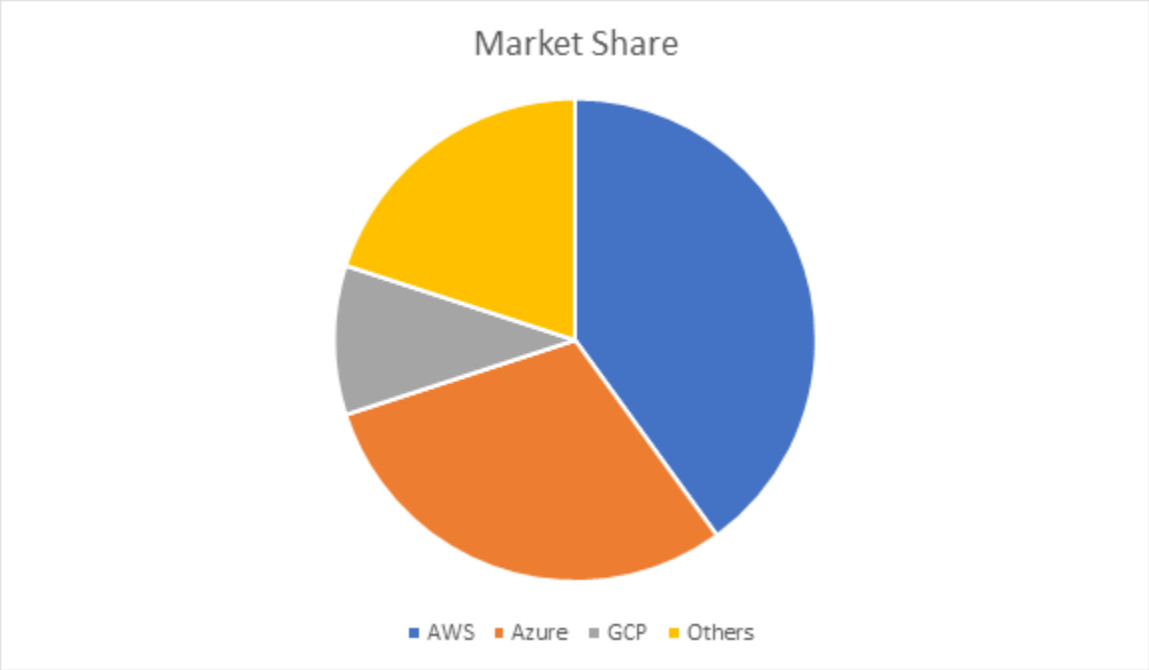
I. Introduction

Cloud Computing has taken over the Information Technology scenario in the recent years. It is widely recognized to have three layers: 1) Infrastructure-as-a-Service (IaaS) 2) Platform-as-a-Service (PaaS) 3) Software-as-a-Service (SaaS). Among these 3 layers, IaaS systems, for example Amazon Web Services, Microsoft Azure, Google Cloud, provide infrastructure utilities like Virtual Machines (VMs) and Storage Solutions, and have been the most popular till now. As of the current scenario, IaaS standardization has matured, and in contrast various SaaS systems are available on the internet. For example, Salesforce, Gmail, Office 365. As SaaS systems are the closest to the users being the highest end of the 3 layers, most Internet users would know about their existence. In between the IaaS and SaaS are the PaaS systems (e.g.: Google App Engine, Google Borg, Docker) which act as the bridge between the IaaS and the SaaS. This has been realised to have an increased importance in the Cloud Computing Industry as well. Most of the new businesses and start-ups are focussing on Cloud based products. Not to forget the older organisations with the hardware at hand either have come out with their own PaaS and IaaS service to make good use of their infrastructure and also have shifted their resources to the Cloud.

DevOps is an abbreviation for Development and Operations which in a technical perspective is said to be the PaaS system. In a nutshell PaaS is the one responsible to provide tools that help IT Engineers build Cloud Application in no time, as compared to the hard built systems previously. Cloud was originally introduced in order to reduce the hassle for the programmers to take off the details with respect to the servers, routers and switches. Ironically, they still had to suffer due to the complexities of cloud computing.

Cloud Computing covers almost all areas of computer science and technology and has full stack techniques available to maintain the same. But, as we know full stack engineers are rare in the market, since most of the engineers have expertise in only one tier of the market such as web services. Thus, the cloud computing engineers have to deal with a lot of complicated configurations, deployment pipelines, performance and other things that are related to the hardware. Operating System (OS), containers (e.g.: cgroups, Docker, Rocket), Database Management, TCP/IP, DNS, Firewall and the list still goes on. Hence, what the engineers today envision is a single DevOps environment that will facilitate and simplify all the Development and Operational processes As a result, getting a fluent system that will bridge the gap between the Developers and the Cloud Computing Engineers.

PaaS or the DevOps Environment for cloud-based applications must consist of 1) App Coding, Build, Testing and QA, 2) VM or container monitoring, 3) Resources Management, 4) Task Scheduling, 5) Concurrency Coordination, 6) System-Logs, 7) Information Visualisation and Analytics.



There are other advanced functions such as the RESTful Application Programming Interfaces (REST = Representational State Transfer), Team Collaboration, Service Integration, Security, Privacy. In short, PaaS is a refined and updated experience of the initial cloud computing engineers in terms of the development and operations for the same. The main goal is to make DevOps environment work in such a way that it becomes an assistant to the developers using the cloud to help build and launch the cloud applications. It should also have some great properties like scalability, extensibility, reliability, security, concurrency, and cost efficiency.

This paper is an introduction to PaaS, as well as a concise survey of the current and upcoming scenario of PaaS techniques. First, we will start with a list of State-of-the-art PaaS solutions e.g.: Google App Engine, Google Borg, AWS Elastic Beanstalk, MS Azure Stack, Openstack Horizon and Magnum, VMware Cloud Foundry, Docker, Zookeeper, Puppet, Hadoop YARN across the various layers of Cloud. From this we will be able to have areas of consensus and diversity in the philosophies and methods. We will also be exploring Google Kubernetes, Apache Mesos, Heroku and Cloud Studio, followed by the conclusion in the last section.

II. State of the Art PaaS Solutions

State of the art means that it is popular in the user base or is of the latest technology. We will be examining DevOps in perspective of the 3 International Market giants – Google, Microsoft Azure and Amazon Web Services. We will also be presenting open-source solutions from OpenStack, VMware and dotCloud. We also have an introduction included for classic tools for large scale system coordination, configuration, and resource management, i.e., Puppet, ZooKeeper, Hadoop YARN.

Google App Engine

The main advantage of GAE is in the simplification of the development and deployment phases of the web services. It has special SDK libraries for the cloud applications, GAE also has multiple functions that enable useful functions that can help cloud applications get online quickly. These functions are: 1) Automatic Resource Scaling, 2) Distributed Caching, 3) Task and Message Queuing. It also has the facility of providing users to manage versions and also access them simultaneously. These Client Requests can be forwarded to the different versions in order to manage the traffic load. This helps in facilitating A/B or split testing that is a common stage in Software Development.

Google Borg

It is known as the “efficiency weapon” that has been used in most of the Google Services, it wasn’t brought public until 2015 with the publication of a paper in ACM. It indicated the clear importance of Borg to Google. Borg is aimed to have efficient management and optimised utilization of large-scale distributed server clusters. Borg is the service that has been responsible for running thousands of Google applications at ease. To enhance the utilization of each physical server, Borg makes 3 choices of the performances in order to bring operations for the application. 1) It doesn’t employ Virtual Machines

which can severely degrade the working of the server instead it takes help of lightweight Linux containers or cgroups. 2) The jobs running on Borg are classified into 2 categories – end user facing services and batch jobs, for the application of different resource management and task scheduling strategies. The apps deployed on the server are of the mixed manner, multiple applications that are logically isolated on the same server instead of a single application having multiple servers dedicated for itself.

AWS Elastic Beanstalk

Amazon is considered to be both – the inventor and the dominance in the cloud computing industry. AWS among its number of services provides Elastic Beanstalk system to help developers conveniently and rapidly deploy their applications on top of AWS. It is similar in look as compared to GAE, moreover apart from Elastic Beanstalk, AWS has more PaaS tools offered for the developers like CodeCommit, CodeDeploy, CodePipeline as well as PaaS tool for the System Operators, such as CloudWatch, CloudFormation, CloudTrail, Config, Console, OpsWork and the Service Catalog.

Microsoft Azure Stack

Despite the late entry of Microsoft into the Cloud industry it has established itself well enough in the users' minds, and brings itself second to AWS. It claims to be a full stack DevOps Environment for the cloud hosted applications. Yet, from a PaaS perspective, there are at least 10 services in Azure that are worthy of attention. They supply many tools for the developers: Visual Studio Team Services, Visual Studio Application Insights, DevTest Labs, Xamarin (for faster cloud based mobile applications), and Storage Explorer which are a few of the many services covered under Microsoft Azure. Apart from these developer tools, we also have management tools available on the Platform: Azure Resource Manager, Scheduler, Log Analytics, Automation and Site Recovery.

OpenStack Horizon

It has become the standard for the open-source cloud computing solutions. It has also joined the PaaS Sector. It has provided an OpenStack horizon which is a tool to visualize the information and resources you have for your cloud systems. A user can clearly see how many servers they own, the occupation of the resources and the health of the operations. It also acts as a console for the whole system resource management, enabling one-click start up, shut down, stand by, hibernate operations for the users. Recently the community has also released a tool called Magnum to efficiently support the running of containers.

VMware Cloud Foundry

The world known virtualization company VMware has shifted its motive from open source computing. In 2011, to the world's surprise VMware announced its first open-source fully fledged PaaS solution known as Cloud Foundry. It is based upon Ruby on rails and acts as a distributed system composed of several other sub systems which can communicate with each other via message passing. With an identical code base this same system can be used to deploy on larger scale systems or mega data centres or a couple of personal computers or a bundle of public cloud Virtual machines. Those familiar with OpenStack may find Cloud Foundry quite similar.

Docker

Due to the extreme popularity Docker has gained in the recent days, people have missed on to the name of the company that has created it – dotCloud. It has been the most trending cloud computing technique and hosts applications by letting them run in logically isolated containers by using a series of Linux Kernel features like cgroups and namespace. Since many containers share the same Linux kernel the resource utilization and efficiency increases drastically as compared to the conventional VMs. The design philosophy is much similar to the Google Borg Project. Information technology in most cases have efficiency and compatibility at stake which is also the case in Docker. Despite the efficiency, Docker has some issues regarding compatibility like no support for 32-bit architecture or windows servers. Also, increased sharing of dockers results in less isolation of the containers such that security has been an issue in many cases.

Puppet and ZooKeeper

Imagine you are a cloud computing engineer and you have hundreds of VM hosted on Elastic clouds. You will have to often run commands that will be repetitive and common to all the Virtual Machines. At such time you won't have to run multiple shells and give commands iteratively each and every time, instead you can make a puppet script (command.pp <- file extension) and run it at once on all the VMs. Puppet is a centralized configuration management system that is applicable to almost all major operation systems, Linux, Unix, and Windows platforms. System Admins and Operation staff can easily manage the massive repetitive configuration details with the help of Puppet. It uses a simplified Client/Server architecture to organize the nodes of the system so that even a newbie engineer can manage the same at any organisation. However, this centralized system can become a bottleneck at times.

Distributed coordination and configuration management is being recently handled by this service named ZooKeeper which has been very popular these days. Though it is more complicated than puppet it is simpler than other services called Paxos. It provides various functions in distributed systems such as distributed lock, message queues, master node election and dynamic configuration for scaling and load balancing. In theory of distributed systems, the ideal scheme is the Paxos protocol that has been designed and implemented by Leslie Lamport, and Paxos has been implemented in the Google Chubby System. Paxos is however considered to be very complicated considering the comprehensive options available for the various scenarios. It is hence considered to be an over-provision. This is where ZooKeeper wins the battle which implements the basic principles of Paxos.

This was done by implementing the ZooKeeper Atomic Broadcast (ZAB) protocol. As of now, many PaaS Systems make use of ZooKeeper such as resource manager for Apache Mesos and the message queue of Apache Kafka.

Hadoop YARN

YARN stands for – Yet Another Resource Negotiator, and so it has been developed for the sole purpose of Hadoop's resource management. The reason they use "Yet Another" is that prior to the release of YARN, they already had a resource manager that had limited scalability. Hadoop's server cluster

previously could barely accommodate 4000 nodes and unoptimized resource utilization. “Necessity is the mother of invention”, was yet again justified when the engineers designed a two-layer resource scheduler to solve the problem. The two layers are ResourceManager + ApplicationMaster, where in the scheduling strategies for the tasks were determined by their respective applications, rather than the single entity. This radically enhanced Hadoop’s performance and the scaling problem for resource utilization.

III Consensus and Diversity

We defined a number of universal values compatible with most of them on the basis of these state-of-the-art PaaS solutions. Below are some key points of agreement on cloud-hosted apps creation and operation

There is no PaaS solution that meets or fulfils every requirement, even if it is full or complete. As such, the cloud app operators and developers must instead of trying to deliver a single panacea solution, create multiple solutions for specific scenarios.

Main performance-metrics in operation are usability and reliability. Though university solutions (for example, Spark) will seem to be much better than industrial solutions (for example, Hadoop), the former will not soon replace them. UI, security safeguards, code reviews, debugging issues and group maturity also are the main factors in real-world scenarios. None of the VMs or boxes function as real physical devices. Nearly every VM provider (e.g. AWS-EC2 and Aliyun-ECS) warns that VMs can only be used statelessly, i.e. any time persistence is not stored in a VM, as they may crash or even vanish at any time. Containers are undoubtedly much more fragile.
The entire VM market is unlikely to capture containers, so there will probably be a dynamic balance between each market share. Containers are generally related to productivity and use, whereas VMs are more compatible and more isolated. In reality, running containers in VMs can become a common paradigm for cloud-hosted applications.

In general, SLAs are more important than vendors for customers. SLAs are not rules, and people in some circumstances do not obey even certain laws. In addition, few cloud clients were able to identify and quantify the performance measures described in SLAs accurately. Most customers never calculate their efficiency measurements. SLAs, therefore may not be taken too seriously for a cloud user, but can be used as a guideline and a guidance in negotiations with a cloud provider.

Public clouds make startups easier and veterans profit from private clouds. AWS EC2 plus S3 plus RDS will save considerable money and time on infrastructure maintenance for a start-up enterprise. A large number of chances will however be lost to optimise machine performance, as the infrastructure is in this case a 'black box.' This is why Dropbox relied entirely on Amazon S3 for file contents storage during its early years, but since then has moved to its Magic Pocket, a private object cloud, for the vast majority of its file contents.

In terms of their popularity, accessibility, maturity, and transparency, however, PaaS solutions were considerably diverse. As any given PaaS solution's service becomes more sophisticated, its restrictions on education applications increase as a result. As such, no commonly accepted PaaS solution can meet the requirements of all parties. In some metrics or tiers a PAAS solution does well; normally in others it does not work properly. At both ends, PaaS solutions appear constrained and free.

A limited PaaS solution will use underlying resources completely or primarily, i.e. computing, stocking and network resources. However, app developers typically need to adhere to a particular/private collection of data and API restrictions. Often, software developers can be limited to those languages of programming. GAE is the PaaS leader. If programmers develop GAE-based applications, they may benefit from the Google Cloud platform techniques and tools, but they have a number of limitations such as restricted language and libraries support, HTTP styled APIs and lack of continued session status. On the other hand, open PaaS solutions do not place invasive limitations on software codes, but enable application developers to maintain their original programming languages, system structures, components and containers as much as possible. Heroku (see Section 4 below) represents the open solution supporting all common programming and related "minority" languages such as Ruby. There are some PaaS semi-open solutions between these two extremes, which render their source codes, such as Cloud Foundry, Docker, and OpenStack publicly accessible. For example, dotCloud opens the Docker software and maintains a centralised repository for all Docker users to upload their Docker images free of charge and easily find other Docker images.

IV Frontier-Solutions Exploration

With the conceptual and methodological levels of consensus and diversity found in state of the art PaaS solutions, we are now exploring a range of frontier solutions towards a more fine-grained, comprehensive DevOp setting. **"Kubernetes"** is not an ordinary English term, so many people call it K8. The word 'Kubernetes' is an ancient *Greek* term that refers to a *pilot* or *steering man*. Google is said for a subtle reason to have used this name. As Docker plans for himself as a whale that moves by sea with containers, Kubernetes leads this "container age." While presented only one year ago, Kubernetes now has a significantly more market-share than Docker and has strong support from many cloud computing giants. At the present time, most people prefer to consider Kubernetes as a Docker upper layer. That is, the Kubernetes team has developed a Docker-based service-centered distribution system that can scale automatically and diagnose a service when needed. At the same time, Kubernetes also supports another competitive container technology, Rocket, built by CoreOS, in order to relieve its dependency on Docker. Kubernetes is also regarded as Google Borg's open-source edition. Kubernetes has made great strides not only in its open-source aspect, but also in its transparency. For example, regardless of the language used to write a particular app, the application can be moved directly to a Kubernetes service which communicates via standard TCP-based protocols with the Internet or other services.More notably, Kubernetes introduces an entirely new Pod layer, between the server node and the node containers. Many containers will run in the same pood at the same time, effectively increasing the efficiency of data transfer between these containers. There has recently arisen a new, hot concept of a microservice in the IT industry, where an interconnected service is composed in a variety of separate, network-connected

micro-services. Kubernetes is designed to serve micro-services in this way. Apache Mesos. Apache Mesos. Mesos is more transparent and has a finer granularity than Docker and Kubernetes, and is thus referred to as "the kernel of distributed systems." Mesos was originally developed and was commonly used on Twitter by the well-known AMPLab at UC Berkeley. In the following year, its founder, Ben Hindman, visited Twitter with his UC Berkeley team. Just eight Twitter engineers were there at that time and Ben Hindman was disappointed. Three of these 8 engineers were subsequently added to the Mesos team, all former Google employees.

Mesos has two main improvements compared with Docker and Kubernetes in order to achieve more transparency. Mesos specifically distinguishes the management of resources from the planning of tasks in order to allow applications to get their requested resources better. Second, Mesos offers resilient application interfaces for other systems such as Spark and Marathon to fit frameworks. Mesos team also proposed a new allocating strategy called DRF, namely, Dominant Resource Fairness, to enhance resource management. The concept for DRF was derived from comments by many IT engineers to concentrate on the dominant share of resources needed by users, in the presence of many different types of resources, an acceptable resource allocation strategy. Suppose, for example, that Mesos allocates resources from the physical server concurrently to two users, A and B, where user A performs high-performance CPU tasks and user B runs high-performance work. DRF will then try to assign additional CPU resources to A than B.

Heroku. - Heroku. Founded in 2007 and bought by Salesforce in 2010, Heroku has been fighting in the cloud computing industry for many years. Heroku, however, has been taking note of its free, neutral PaaS platform recently. Heroku built and introduced a highly portable PaaS platform "compatible with all mainstream PaaS solutions", to resolve or mitigate an invasion of too many PaaS solutions into the application codes of users and the resultant "cloud-based lock-in" problem. In addition, the Heroku team has refined 12 factors required for the intelligent development and operation of cloud-hosted apps with their detailed observations and techniques. In 2011, particularly, Ruby's inventor joined Heroku as his chief architect, reflecting the programming community's positive attitude towards Heroku.
Cloud Studio (currently in alpha phase). Large enterprises such as Google and Amazon have built almost all the mainstream PaaS systems. This does not say, however, that PaaS can't contribute to small companies and organisations. In reality, small teams may develop unique PaaS solutions to meet special

needs.. A number of useful DevOps tools for cloud-hosted applications have recently been released by Cloud Studio (thucloud.com) ; Virtualpool (for VMs and container accommodations), iDashboard (for displayed and user operations), dual cloud acceleration (for particularly Google, Gmail, Github, and Dropbox), cloud disc, iRecommend, etc.

V Conclusion

The cloud computer industry today is rapidly developing, with fierce competition, also true for the DevOps or PaaS world. No standardised PaaS APIs or data formats have yet been developed and the 3 market giants – Google Cloud, AWS, and Microsoft Azure – have adhered to their PaaS standards. Heroku has thus yet to realise its vision of a unified PaaS. Many DevOps teams are using state of the art PaaS solutions as references and using open source PaaS-related software to build their own optimal PaaS solutions in order to create a suitable 'sweet-spot' solution. For today's IT engineers who need solutions, is this too complicated?

The response is, we believe, NO. As we have said repeatedly, the innovations of cloud computing evolve quickly and significantly, and thus create new ideas and resources continuously. For instance, Spark was introduced after IT engineers had only mastered Hadoop (MapReduce) which reputedly surpasses Hadoop by 100. Also, Kubernetes and Mesos came to provide finer granularity and higher overallity when IT technicians had mastered the Docker operations. In order to learn these new methods, engineers will soon be exhausted.

Fortunately, while cloud computing has provided a new market, few truly new techniques are needed. Once we look at the cloud computing stack, we find that most of our cloud computing activities include device design and resource management, for which computing, storage and networking are still fundamental resource concerns. In other words, main IT technology has never essentially changed. These include programme compilation, link, loading and execution; memory and disc I/O management by the OS; and TCP/IP protocols. Wise engineers can detect the basics of the dizzying range of innovations and then adapt instead of being hijacked by them.

References

1) Daniel Cukier. 2013. DevOps patterns to scale web applications using cloud services. In *Proceedings of the 2013 companion publication for conference on Systems, programming, & applications: software for humanity* (*SPLASH '13*). Association for Computing Machinery, New York, NY, USA, 143–152. DOI:https://doi.org/10.1145/2508075.2508432
2) Z. Li, Y. Zhang and Y. Liu, "Towards a full-stack devops environment (platform-as-a-service) for cloud-hosted applications," in *Tsinghua Science and Technology*, vol. 22, no. 01, pp. 1-9, February 2017, doi: 10.1109/TST.2017.7830891.

3) Mohammad, Sikender Mohsienuddin, Streamlining DevOps Automation For Cloud Applications (October 4, 2018). International Journal of Creative Research Thoughts (IJCRT), ISSN:2320-2882, Volume.6, Issue 4, pp.955-959, October-2018, Available at SSRN: https://ssrn.com/abstract=3655574

4) M. Rajkumar, A. K. Pole, V. S. Adige and P. Mahanta, "DevOps culture and its impact on cloud delivery and software development," *2016 International Conference on Advances in Computing, Communication, & Automation (ICACCA) (Spring)*, 2016, pp. 1-6, doi: 10.1109/ICACCA.2016.7578902.