Paas Providers And Their Offerings

Deepali Bajaj, Urmil Bharti , Anita Goel, S. C. Gupta

Abstract: Platform-as-a-Service (PaaS) is a cloud service model that provides an environment to develop, deploy and operate software applications. PaaS provides preconfigured capabilities like, programming languages, supported frameworks, hosting and runtime environments, application development tools, and database servers managed for developing web applications as well as mobile applications. For an organization, selection of PaaS provider is a crucial task. Since PaaS providers offer a variety of development and deployment features, it becomes difficult to select from among them. Also, the organization may get locked to a particular provider and it is not easy to migrate to a different provider due to legal constraints and technical incompatibilities. In this paper, we present a study of currently available PaaS offerings. We identify the important PaaS characteristics that must be assessed before selecting a PaaS offering. We present a comparative study of different PaaS providers on basis of the identified characteristics. Our study benefits enterprise architects, IT managers and development teams to select PaaS offering best suited to their requirements.

Index Terms: PaaS, Cloud Foundry, AWS Elastic Beanstalk, Heroku, Google AppEngine, Red Hat OpenShift Online, Appfog, Application Portability

1. INTRODUCTION

National Institute of Standards and Technology (NIST) outlines three service layers to render cloud computing capabilities: Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS) and Software-as-a-Service (SaaS). PaaS presents the most significant impact over other service aspect of cloud computing because it provides tailored development of software on the cloud [1]. NIST explains PaaS as: “Service capability provided to its consumer to deploy consumer created and acquired applications onto the cloud infrastructure using various programming languages and other tools supported by the provider.” PaaS is a collection of related services that provides the developers with fast and convenient ways to create, deploy and manage software applications onto the cloud infrastructure. It provides abstraction to software developers from underlying difficulties of installing and configuring applications on hardware systems. PaaS has many benefits including rapid prototyping, lower costs, faster time-to-market and lower risks [2]. Initial PaaS offerings provided limited feature set, like, forms, databases and simple APIs. With the maturity of cloud technologies, PaaS offerings also manage functions, like, user subscription, user security, role-based security, resource metering, workflow, reporting and other shared services [3]. The understanding of features and functions of PaaS is important for selection and utilization of PaaS in an efficient manner. PaaS provider competition is intense because this technology is so much in demand among the developers’ community [4]. However, selecting an appropriate PaaS from an immense number of competitors is a challenging and complex task for developers.

- *Deepali Bajaj1, Urmil Bharti2, , Anita Goel3, S. C. Gupta4 Department of Computer Science, Shaheed Rajguru College of Applied Sciences for Women,*
- *University of Delhi, Delhi, India, deepali.bajaj@rajguru.du.ac.in Department of Computer Science, Shaheed Rajguru College of Applied Sciencesfor Women, University of Delhi, Delhi, India, ubharti@hotmail.com*
- *Department of Computer Science, Dyal Singh College University of Delhi, Delhi, India, goel.anita@gmail.com Department of Computer Science, Indian Institute of Technology Delhi, India, scgupta@cse.iitd.ac.in*

This is because PaaS can significantly impact performance, availability and flexibility of an application. For an organization, PaaS selection is an important process because the selected provider shall continue to partner with them. So, there is a need to evaluate various parameters of PaaS to ensure that it fits the requirement of an organization. Currently there is no standard list of guiding parameters for PaaS selection. Based on our findings, we have identified ten important and guiding PaaS parameters that vary from one provider to another. These parameters are - Programming Languages, Supported Frameworks, Scaling capabilities, Deployment Models, Runtime environment, Database Management Systems, Development Tools, Underlying IaaS, Application Portability and Pricing Model. These key parameters need to be considered when selecting a PaaS provider, as they directly impact cloud applications development and deployment. In this paper, we present a comparative study of the PaaS providers on basis of the key features supported by them. We identify and expound ten guiding parameters that are important for selecting a PaaS provider. The key open source PaaS providers have been identified, namely, Cloud Foundry, Amazon Web Services Elastic Beanstalk, Heroku, Google App Engine, Red Hat OpenShift Online and Appfog. The PaaS providers have been evaluated on the identified parameters and a comparative summarization of this evaluation is presented. Our study is for use by the developer, when selecting a PaaS offering.

2. RELATED WORK

Academically, very few research papers were found, related to current PaaS offerings. A previous published work categorized PaaS providers into two broad areas - (i) closed source development model and (ii) open source development model and their result indicates that in terms of cost, use of open source model is better than closed source model [5]. A detailed study is provided [6] on high level overview of two leading open source PaaS providers - OpenShift and Cloud Foundry. The authors highlight supported features, architecture comparison at component level, operational compatibility and deployment principles for the two PaaS offerings. Nandimandalam et al. [7] have given detailed study on architectural component aspects of four open PaaS packages namely AppScale, Cloud Foundry, Cloudify, and OpenShift. Gorai et al. [8] propose key features and limitations of PaaS providers like Google

App Engine (GAE), Microsoft Azure, and Amazon AWS. They describe some considerations and steps for selecting the PaaS. Intel IT center [9] highlights six high level steps that an organization should consider while planning for PaaS. It points out that PaaS solution must support multiple languages, application services and data technologies. Though several researchers have worked in similar areas, but our work focuses on critical PaaS characteristics like scaling capabilities, underlying IaaS and Application portability.

3. BACKGROUND

Platform-as-a-Service is a type of cloud computing service offering in which a provider offers a platform to its customers and enables them to develop, deploy, run, and manage their business applications over the cloud. PaaS services are hosted in cloud and accessible to its users via web browsers. PaaS eliminates the need to build and maintain the infrastructure that is typically required in any software development processes. PaaS permits developers to develop applications using pre-defined software components thereby offering features, like, scalability, high availability and multi tenancy capabilities [10]. In order to qualify as a PaaS provider, a product must offer (i) databases creation tools (ii) development platform (iii) application management tools and (iv) capabilities to deploy applications. PaaS comprises of various underlying infrastructure components, like, servers, operating systems, networking equipment and storage. PaaS provides pre-coded application components like directory services, workflow, security mechanisms, search and other development tools that help in reducing overall application development time. When using PaaS, the user need not buy expensive hardware, sophisticated software development tools, business intelligence and analytics tools. PaaS provides development options for variant platforms, like, desktop computers, laptops, mobile devices and web browsers, thus making cross platform apps faster and easier to develop. PaaS provides all the competence that is required in a web application development lifecycle i.e. building, testing, deploying, updating and managing an application. It also enables geographically distributed collaborative application development where development teams work together on the same application built from remote geographical locations [11]. There are some fundamental drawbacks of PaaS. Customer captivity, also known as vendor lock-in, is a drawback of PaaS. Customers can be locked in to a particular PaaS vendor and get attached on a single provider technology implementation. According to Gartner research, due to dearth of standardized APIs, tools and languages in PaaS, cloud services have severe problem of lock-in. In this scenario, customers cannot easily migrate to a different vendor in near future without suffering from significant costs, legal restriction or other technical incompatibilities. Integration of PaaS services with the existing system and applications could also pose a challenge and can trigger an increased complexity. For a developer of web based applications or software, PaaS is a flexible solution to decrease preliminary startup cost for hardware infrastructure. If these drawbacks are taken care appropriately, PaaS will provide a great potential to gain ground in cloud market [12].

4. EVALUATION OF PAAS OFFERINGS

Platform-as-a-Service(PaaS) model of cloud computing has emerged as a major element of the modern IT environment. PaaS adaptability is evolving in real time. However, finding and deploying cloud application on PaaS can be a daunting challenge. The best approach to choose a PaaS solution is to start with the detailed study of leading PaaS providers. Below is an analysis of the main functions and features of existing open source PaaS offerings.

4.1 Cloud Foundry

Cloud Foundry is the world's first open PaaS offering originally developed in-house at VMware. Pivotal Software, a joint project of EMC, VMware, and General Electric, is now owned by Cloud Foundry. Cloud Foundry provides an application execution engine, an automation engine for application deployment and easy management, command line interface (CLI) for application services interface. One of the important tenets of Cloud Foundry is its independence on underlying infrastructure which gives users the option to use their existing infrastructure like desktop, datacenters and private Clouds whilst still leveraging all the advantages of PaaS. Organizations can deploy this PaaS offering either on their own infrastructure or on cloud providers' infrastructure like Amazon Web Services (AWS) or OpenStack. Cloud Foundry is extremely customizable that allows developers to code in several languages and frameworks like Spring for Java developers and other JVM languages/frameworks, including Rails, Sinatra, Node.js etc. Developers prefer Cloud Foundry because it can easily be integrated with their existing tools and code. Cloud Foundry is suitable for anyone interested in lowering the cost and complexity of configuring infrastructure for their apps. Developers can deploy their applications on Cloud Foundry using its tools and without any modification to their code [20].

4.2 AWS Elastic Beanstalk

AWS Elastic Beanstalk gives developers and systems administrators a simplified, faster way to deploy, control and manage their applications without worrying about core elements of underlying AWS infrastructure [13]. Beanstalk decreases management complexity without reducing choice or control. Developers can use the AWS platform services without creating or maintaining their own application servers. It enables Java developers using Apache Tomcat software stack to direct upload of a J2EE application. AWS readily supports Java, Python, .NET, Node.js Ruby and different types of container for each language as well. A container characterizes infrastructure and software stack that is available for a given environment. MySQL, Oracle and SQL Server can be set up and managed easily in Beanstalk. AWS also presents RDS (Remote Desktop) web service, which completely eliminates tasks of database administration. It also supports the applications which are originally not written to be used on Web. Developers can take leverage of Amazon Beanstalk in automatic load balancing, auto-scaling, managing of peaks in the application's workload and application health monitoring [17]. Once an application is deployed, Beanstalk may provisions other AWS resources, if required, such as Amazon EC2 instances. Developers can communicate with

AWS Beanstalk using the AWS Management Console, AWS CLI (AWS Command Line Interface), or using a high level CLI 'EB' which is especially designed for Elastic Beanstalk. Users pay only for the underlying AWS resources that an application uses. There is no surplus charge for using Elastic Beanstalk

4.3 Heroku

Heroku is a PaaS platform based on abstract computing environment known as dynos, that are Unix style virtualized containers. Dynos run processes in isolated environments and allows users to run apps inside these containers [14]. Heroku takes care of other thing that are required to run apps including container orchestration, load balancing, logging, configuration, failovers, security, and many others. It has a powerful ecosystem needed for deploying and running modern apps. Heroku is a polyglot PaaS platform as it allows the developer to build, deploy, run and scale applications across multiple languages. It supports Ruby, Scala, Clojure, Python, PHP, Go etc. Heroku is well integrated with Git for deploying applications. Using Git, a single command can push application to the remote Heroku repository. There are many other ways of deploying applications like GitHub integration, Dropbox Sync, or Heroku API to build and release apps [19]. Heroku combines application's source code and its dependencies like packages, modules and libraries that must be available in the runtime environment. It compiles source code, find out dependencies and bundles all these resources in a structure called slug. The slug contains all the resources required to run an application.

4.4 Google App Engine (GAE)

Google App Engine is a PaaS offering designed to host distributed Web applications. Developers can quickly build small mobile and Web back-ends applications locally (on developer machines) using programming language, frameworks, runtimes, and libraries and later deploy them on to the Cloud. After deployment, applications run in the same environment that powers other Google applications. This PaaS offering delivers an execution environment which allows applications to run on a virtualized technology that ensures automatic on demand scaling. Google App Engine offers very fast development, deployment, administration, monitoring, troubleshooting, and other helpful developer tools. Currently App Engine supports languages like Java, Node, PHP, Python, Ruby and Go. In Google App Engine appropriate Software Development Kit is available for each supported language. It supports both SQL and NoSQL databases [15]. Other PaaS providers generally support IaaS like features that enable its users to highly customize their environment. But this is not achievable with Google App Engine and App Engine is frequently criticized for not rendering transparency to its user to allow control of the infrastructure they uses. Developers get managed hardware infrastructure and runtime environments which are guaranteed to auto-scale. Developers are not equipped to have direct control over resources and their allocation since the underlying hardware resources are hidden by Google App Engine layer. Google takes the burden of infrastructure while developers work on their codes.

4.5 Red Hat OpenShift Online

Red Hat OpenShift Online is a Platform-as-a-Service application development and hosting platform. It facilitates developers and IT organizations to provision, manage and scale applications in cloud environment [18]. OpenShift Online provides a wide choice of programming languages and frameworks as well to its developers, such as Node.js, .NET, Java, PHP and Python. Other integrated development tools, such as JBoss Developer Studio, Eclipse integration, and Jenkins support the complete application life cycle. OpenShift Online is a multitenant cloud platform, which enables developers to develop, deploy and run containerized applications. It also makes use of Kubernetes for container orchestration and management which simplifies operational requirements such as deployment, scaling, service discovery and health management. Developer and operations-centric tools enables accelerated application development. It allows swift management of multiple projects using a Web-dashboard. OpenShift makes use of quick start templates, which allow developers to do one-click deployments of application frameworks. It provides sets of services, build configurations and deployment configurations needed to build an application.

4.6 AppFog

AppFog is a public PaaS offering that makes fast and easy deployment of scalable, robust, high performance cloud-ready applications. Using AppFog, developers can target on writing web applications rather than to worry about management of underlying infrastructure. AppFog aims to deploy applications on a swift, geographically dispersed, and elastic cloud based platform [16]. With AppFog, developers only need to manage their code and data, and rest like operating systems, virtualization, runtime, middleware, servers, networking, storage are taken care by PaaS. Developers can concentrate on their application rather than provisioning and configuring Web servers, or setting up databases. AppFog runs on multiple clouds and this is its most attractive and appreciated option. Currently it runs on Amazon, OpenStack and Microsoft Azure. AppFog is built on the top of Cloud Foundry that may be attractive to those developers who favor the VMware platform. It also allows easy migration of applications if they are previously hosted on Cloud Foundry. Currently AppFog does not provide persistent file storage so its users may use Amazon S3 or other object storage systems that levies additional charges. It supports many languages like Java, Node.js, PHP, Python, and Ruby. AppFog supports multiple runtimes environments and frameworks, auto scaling, load balancing and more.

5. COMPARISON OF PAAS OFFERINGS

Currently there are plenty of PaaS providers available in the market having their unique strengths and weaknesses. In order to evaluate and choose a PaaS provider, there is a need to consider various parameters that a vendor supports, like, Programming Languages, Supported Frameworks, Scaling capabilities, Deployment Models, Runtime environment, Database Management Systems, Development Tools, Underlying IaaS, Application Portability and Pricing Model. Here we briefly discuss these parameters:

1. **Programming Languages:** Leading PaaS offerings are "polyglot" which means that multiple programming languages are supported by the PaaS. It enables developer to code in any language of their expertise like, Java, Ruby, Python, PHP, Perl and Javascript.

2. **Supported Frameworks:** Developers must choose the best framework according to their enterprise needs. The framework provides the software environment to create, build and deploy web and mobile applications without doing everything from scratch. It may happen that developer is comfortable with a particular framework, but the same is not available on a chosen PaaS offering. So supported frameworks becomes an important parameter for the selection of PaaS vendor. Thus PaaS providers often follow the strategy of supporting as many languages and frameworks as possible in order to attract multiple programmer communities.

3. **Scaling:** Scalability is one of the key benefits that comes with cloud computing. It means that every application or hardware infrastructure should be expanded to manage increased load. Scalability in web application ensures that they can scale-up to handle the increased load and will not crash. PaaS vendors supports three types of scaling to manage the system load - i) Scale-Up (Vertical scaling), ii) Scale-Out (Horizontal scaling) and iii) Auto-scaling. Scaling up a system indicates adding more power to an existing infrastructure instance. This means more powerful processors (CPUs), high end memory (RAM) and faster storage like Solid State Drives (SSDs). Scaling out the systems horizontally adds additional computing power or memory capacity to an overall system by adding additional servers. Auto-scaling (also called automatic elasticity) refers to automatically adding or removing compute resources depending upon actual usage.

4. **Deployment Models:** Today's PaaS offerings are also "Polyhost". It means they support a variety of underlying cloud hosting environments on top of which the PaaS is operated. There are three major deployment models that most PaaS providers offer to their customers: public, private and virtual private. In Public cloud deployment, cloud services are offered publicly to multiple customers among whom resources offered by the cloud service provider are shared. In Private cloud deployment, cloud services are used exclusively by a single customer. Due to reduced chances for resource sharing and increased cost, private deployment may results in decreased efficiency than public cloud deployment. This deployment model may require the customers to have expert in-house ability for PaaS installation and operation. In virtual private cloud, deployment is done within a data center which is owned and controlled by a cloud service provider. Resources provided by a datacenter are dedicated to one customer only and are kept isolated from resources offered to other customers.

5. **Runtime Environment:** Execution environment offered by operating system to an application or software is called runtime environment. It bundles resources like software libraries, environment variables and system variables. It also provides vital services and support to the application processes involved in the execution. Software developers need a runtime environment to test their application's functioning. In current PaaS offerings Apache Tomcat, Jetty, IIS, Dynos Runtime are examples of common run time environment.

6. **Database:** A PaaS database runs on a cloud platform. Access to PaaS database is provided as a service that takes care of high availability and scalability of the database [21]. In current PaaS offerings, MongoDB, Redis, MySQL, PostgreSQL are some examples of common PaaS databases.

7. **Development Tools:** Development tools automate and streamline all development and deployment activities for developers so as to save their time. The most basic development tools are consoles, command-line tools, IDE, code editors, web UIs, continuous integrations and testing libraries.

8. **Underlying IaaS:** PaaS is always build on the IaaS model because PaaS uses underlying cloud infrastructure components like server, storage, network and virtualization to provide its services [24]. All PaaS offerings have some underlying IaaS on the top of which PaaS services are built. Some examples of underlying IaaS are OpenStack, Google Compute Engine(GCE), Amazon's EC2 cloud servers, Amazon Web Services, OpenStack.

9. **Application Portability:** Application portability refers to the feasibility of a PaaS application to be ported across different cloud platforms. If a developer creates and deploys an application on a typical cloud platform, it should be feasible to port the application to a different platform, with a least set of modifications, if required. Many proprietary PaaS vendors are currently using technologies allows portability of cloud applications. Heroku uses Buildpacks, while OpenShift use Cartridges for cloud app portability [22].

10. **Pricing Model:** Service provider uses various pricing models to determine the price. The strategy of pricing can be categorized into two common pricing models such as fixed pricing model and dynamic pricing model [23].

The key open source PaaS providers have been identified. The providers are studied for their support to the key parameters discussed above. A comparative study of the PaaS providers on basis of the key parameters supported

by them has been performed. Table 1 shows the key consideration applied to the studied six leading open source PaaS providers

Table1. *Comparative analysis of leading PaaS Providers*

Key considerations	*Cloud Foundry*	*AWS Elastic Beanstalk*	*Heroku*	*Google AppEngine*	*OpenShift Online*	*Appfog*
Programming Languages	*Go Groovy Ruby Scala Java Node PHP Python*	*Node PHP Python Ruby .Net Go Java*	*Java Node PHP Python Clojure Go Groovy Ruby Scala*	*PHP Python Ruby Go Java Node*	*.Net Java PHP Python Node Perl Ruby*	*Java Node PHP Python Ruby*
Supported Frameworks	*CakePHP Grails Play Rails Sinatra Spring Symfony*	*Spring*	*Django Flask Grails Play Rails*	*Django, Webapp2*	*Django Drupal Flask Rails Switchyard Vertex*	*Django Flask Rails*
Scaling Capabilities	*Scale-Up Scale-Out*	*Scale-Up Scale-Out AutoScale*	*Scale-Up Scale-Out AutoScale*	*Scale-Out Autoscale*	*Scale-Up Scale-Out AutoScale*	*Scale-Up Scale-Out*
Deployment Models	*Private*	*Public*	*Public Virtual Private*	*Public*	*Public Virtual Private*	*Public Private*
Runtime Environment	*Apache Tomcat*	*Apache Tomcat Phusion Passenger Puma-webserver for Ruby Microsoft IIS Nginx*	*Dyno Runtime*	*Jetty*	*Apache Tomcat*	*Gunicorn- WSGI server, Apache Tomcat*
Database	*MySQL, Redis, MongoDB*	*Amazon RDS Amazon SimpleDB Microsoft SQL Server, Oracle*	*PostgreSQL MongoDB Cloudant and Redis*	*BigTable Cloud SQL*	*MySQL SQLite PostgreSQL MongoDB*	*Mongodb MySQL PostgreSQL Redis*
Development Tools	*Cloud Foundry Eclipse Tools (CF), IBM Eclipse Tools for Bluemix* Cloud Foundry Integration for Eclipse* HP Helion Development Platform Eclipse Plugin*	*AWS Toolkit Elastic Beanstalk CLI (EB)*	*Git, GitHub and Docker*	*Google Cloud SDK Google Cloud Shell, Cloud Source Repository, Cloud Tools for Eclipse, Cloud Tools for Visual Studio, Cloud Tools*	*Eclipse integration JBoss Developer Studio and Jenkins*	*Magento*

				for Android Studio, Cloud Tools for IntelliJ, Cloud Tools for Power Shell, Cloud Deployment Manager, Maven App Engine Plugin, Gradel App Engine Plugin		
Underlying IaaS	*Amazon Web Services, OpenStack, vSphere*	*Amazon Elastic Compute Cloud (EC2)*	*Amazon's EC2 cloud servers*	*Google Compute Engine(GCE)*	*OpenStack*	*CenturyLink Cloud*
Application Portability	*Buildpacks*	*Docker*	*Buildpacks*	*Google Cloud Client Libraries*	*Cartridges*	*Cloud Foundry Core compatible*
Pricing Model	*Free usage*	*Pay per use monthly bills free usage*	*Pay per use monthly bills free usage*	*Pay per use monthly billingfree usage*	*Hybrid usage, monthly billing or fixed usage, annually bills or free usage.*	*Fixed usage, monthly billing*

* vendor specific implementations 6.

6 CONCLUSION

Platform-as-a-service is cloud service model that offers developers with hosted software development kits, database management tools and complete application management capabilities. PaaS vendors support virtual resources to build, deploy and finally launch web and mobile applications thus minimizing the requirement for backend software development. Developers must understand the features and functions offered by various PaaS providers, for selecting a suitable PaaS for their business requirement. We have identified ten guiding parameters which are critical for selecting an appropriate PaaS offering. These parameters are discussed with respect to six preferred open source PaaS offerings. A comparative study of important features of these PaaS providers is presented and the outcome is summarized in a table. Our study provides guiding rules for organizations in deciding a well suited PaaS vendor as per their business requirements.

7 REFERENCES

[1] Ebert, N., Weber, K., & Koruna, S. (2017). Integration Platform as a Service. Business & Information Systems Engineering, 59(5), 375-379.

[2] Kriz, P. (2014, July). Survey on open source platform-as-a-service solutions for education. In Proceedings of the 18th International Database Engineering & Applications Symposium (pp. 176-184). ACM.

[3] Paraiso, F., Haderer, N., Merle, P., Rouvoy, R., &Seinturier, L. (2012, June). A federated multi-cloud PaaS infrastructure. In Cloud Computing

(CLOUD), 2012 IEEE 5th International Conference on (pp. 392-399). IEEE.
[4] Varma, M. K., & Choi, E. (2016). COMPARATIVE STUDY OF VARIOUS PLATFORM AS A SERVICE FRAMEWORKS.
[5] Salih, N. K., &Zang, T. (2012). Survey and comparison for Open and closed sources in cloud computing. arXiv preprint arXiv:1207.5480.
[6] Lomov, A. (2014). OpenShift and Cloud Foundry PaaS: High-level Overview of Features and Architectures. white paper, Altoros.
[7] Nandimandalam, M. K. V., & Choi, E. (2014). COMPARISON OF OPEN-SOURCE PAAS ARCHITECTURAL COMPONENTS
[8] Gorai, A. D.,&Goswami B. (2014).Comparison of Several PaaS Cloud Computing Platforms . International Journal for Scientific Research & Development Vol. 2/Issue 09 ,457-462
[9] https://www.intel.com/content/dam/www/public/us/en/documents/white-papers/cloud-computing-what-is-paas-cloud-demand-paper.pdf (as retrieved on 20 May, 2019)
[10] Soni, M. (2014). Cloud computing basics—platform as a service (PaaS). Linux Journal, 2014(238), 4.
[11] Hossny, E., Khattab, S., Omara, F., & Hassan, H. (2013, December). A case study for deploying applications on heterogeneous PaaS platforms. In Cloud Computing and Big Data (CloudCom-Asia), 2013 International Conference on (pp. 246-253). IEEE.
[12] Li, A., Yang, X., Kandula, S., & Zhang, M. (2011). Comparing public-cloud providers. IEEE Internet Computing, 15(2), 50-53.
[13] Narula, S., & Jain, A. (2015, February). Cloud computing security: Amazon web service. In Advanced Computing & Communication Technologies (ACCT), 2015 Fifth International Conference on (pp. 501-505). IEEE.
[14] https://www.upguard.com/articles/heroku-appengine (as retrieved on 15 May, 2019)
[15] Ciurana, E. (2009). Google app engine. Developing with Google App Engine, 1-10.
[16] https://www.ctl.io/blog/post/appfog-and-twelve-factor-apps-explained/ (as retrieved on 20 Mar, 2019)
[17] https://aws.amazon.com/elasticbeanstalk/ (as retrieved on 2 Jun, 2019)
[18] https://www.openshift.com/features/index.html (as retrieved on 30 Jun, 2019)
[19] Middleton, N., &Schneeman, R. (2013). Heroku: Up and Running: Effortless Application Deployment and Scaling. " O'Reilly Media, Inc.".
[20] Bernstein, D. (2014). Cloud Foundry aims to become the OpenStack of PaaS. IEEE Cloud Computing, 1(2), 57-60.
[21] Gelogo, Y. E., & Lee, S. (2012). Database management system as a cloud service. International Journal of Future Generation Communication and Networking, 5(2), 71-76.
[22] Pahl, C. (2015). Containerization and the paas cloud. IEEE Cloud Computing, 2(3), 24-31.
[23] Mazrekaj, A., Shabani, I., &Sejdiu, B. (2016). Pricing Schemes in Cloud Computing: An Overview. International Journal of Advanced Computer Science and Applications, 7(2), 80-86.
[24] https://www.gartner.com/en/information-technology/glossary/platform-as-a-service-paas(as retrieved on 22 Jun, 2019)