A Framework for Platform-as-a-Service Selection and Provision

Thesis submitted in partial fulfillment of the requirements for the award of degree of

Master of Engineering
in
Computer Science and Engineering

Submitted By
Madhu Dhingra
(Roll No. 821132007)

Under the supervision of:
Dr. Damandeep Kaur
Assistant professor

COMPUTER SCIENCE AND ENGINEERING DEPARTMENT
THAPAR UNIVERSITY
PATIALA – 147004

June 2014

CERTIFICATE

I hereby certify that the work which is being presented in the thesis entitled, "*A Framework for Platform-as-a-Service Selection and Provisioning* ", in partial fulfillment of the requirements for the award of degree of Master of Engineering in *Computer Science and Engineering* submitted in Computer Science and Engineering Department of Thapar University, Patiala, is an authentic record of my own work carried out under the supervision of *Dr. Damandeep Kaur* and refers other researcher's work which are duly listed in the reference section.

The matter presented in the thesis has not been submitted for award of any other degree of this or any other University.

Madhu

(Madhu Dhingra)

This is to certify that the above statement made by the candidate is correct and true to the best of my knowledge.

Daman

(Dr. Damandeep Kaur)
Asst. Professor, CSED

Countersigned by

(Dr. Deepak Garg)
Head
Computer Science and Engineering Department
Thapar University
Patiala

(Dr. S. K. Mohapatra)
Dean (Academic Affairs)
Thapar University
Patiala

Acknowledgement

No volume of words is enough to express my gratitude towards my guide Dr. Damandeep Kaur, Asst. Professor, Department of Computer Science & Engineering, Thapar University, Patiala, who has been very concerned and has aided for all the material essential for the preparation of this thesis report. She has helped me to explore this vast topic in an organized manner and provided me all the ideas on how to work towards a research-oriented venture.

I am also thankful to Dr. Deepak Garg, Head of Computer Science Engineering Department for the motivation and inspiration triggered me for the thesis work.

I would like to thank all other respected faculty members fo assistance and suggestions during research period.

Most importantly, I would like to thank my parents and the for showing me the right direction out of the blue, to help m in the oddest of times and keep moving even at times wh no hope.

(Madhu Dhingra)

Abstract

As cloud computing is increasingly transforming the information technology landscape, organizations and businesses are exhibiting strong interest in Platform-as-a-Service (PaaS) offerings that can help them increase business agility, flexibility and reduce their operational costs. They increasingly demand services that can meet their functional and non-functional requirements. Given the abundance and the variety of PaaS Vendors available, we propose, in this research work, a framework for PaaS provisioning, which relies on the parameters for their evaluation. These parameters helps service consumers find the right PaaS providers that can fulfill their functional and non-functional requirements. The proposed selection process ranks potential PaaS providers by matching their offerings against the requirements of the service consumer.

Table of Contents

Chapter-1

INTRODUCTION TO CLOUD COMPUTING

The entire computing spectrum is overwhelmed with cloud computing services these days. It is considered as the turning movement in computing industry. The advent of this technology offers unprecedented benefits to an organization's IT processing and administrative capabilities unlike those available in traditional in-house infrastructure.

As defined by The National Institute of Standards and Technology (NIST):

Cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources that can be rapidly provisioned and released with minimal management efforts or service provider interaction [1].

This recent fad in IT moves computing and data away from desktop and portable PCs top large data centers enabling them to operate like internet by accessing and sharing computing resources as virtual resources in a secure and scalable manner. The mainstream adoption of cloud computing is gaining momentum day by day and this is because of the accelerants that meet the demands of the business owners. These accelerants offer organization, opportunities to lower their cost and free their best technology managers to focus on creating strategic differentiation. These accelerants include:

1. Elasticity: The elastic environment allows capacity to be added, moved or removed almost instantaneously in response to dynamically changing processing needs and this attracts large and small organizations to a large extent.
2. Pay-per-Use vs. Install-and-Own: Cloud eliminates the need to install the application onto the user's system instead it allows to use it on-premise without the need of installation or setup. The pay-per-use pricing schemes employed by the cloud are equally attractive.
3. Cost savings: A report by Computer Economics finds that 15% of IT budgets can be saved by migrating IT infrastructure to the cloud.
4. Effective utilization of infrastructure: Virtualization of hardware and software resources and sharing them among a collection of users allows efficient utilization of infrastructure. Network efficiency is also increased through global load balancing.
5. Public Investments: As NIT com Security report states that nearly 60% companies and businesses all over the world(North America, Japan, Europe, Asia etc.) are enthusiastic about moving to the cloud to help them gain a competitive advantage for their businesses, reduce costs and do more with less[2]. Governments worldwide are migrating their own IT infrastructure to cloud services in an effort to lead by example.
6. Security: With "Security-as-a-Service" model emerging, the in-house ITs greatest non-value adding challenge will be eliminated. Numerous security vendors are now leveraging cloud based models to deliver security solutions.

7. Emerging Standards: Standards for cloud computing are still coming into existence and once standardized, it will reduce or eliminate the risks involved in adoption of the cloud.
8. The Cloud Brokers: The Brokers simplify an organization's adoption of cloud by helping to overcome the issues such as security, privacy, interoperability across multiple clouds.
9. Risk of lagging-behind: The organizations that do not go in for cloud adoption risk missing out on the benefits such as flexibility and agility afforded by on-demand services and access to the latest version of technologies.

1.1 Working of a Cloud Service

When Cloud computing was not there, your software organization requires

- A bunch of computers for all your employees with appropriate software and required hardware.
- Software licenses to give employees the tools they require.

Soon, there emerged an alternative-Instead of installing a suite of software for each computer, you'd only have to load one application. That application would allow workers to log into a Web-based service which hosts all the programs the user would need for his or her job. Remote machines owned by another company would run everything from e-mail to word processing to complex data analysis programs. It's called **cloud computing**, and it could change the entire computer industry. It basically uses networks of large groups of servers running low cost PC technology with specialized connections containing large pools of systems linked together. A virtual group of resources eg. Networks, CPU, storage is built to carry out user's resource requirements.

In a cloud computing system:

- Local computers no longer have to do all the heavy lifting when it comes to running applications.
- The network of computers that make up the cloud handles them instead.
- Hardware and software demands on the user's side decrease.

- The only thing the user's computer needs to be able to run is the cloud computing systems' **interface software**, which can be as simple as a Web browser, and the cloud's network takes care of the rest.

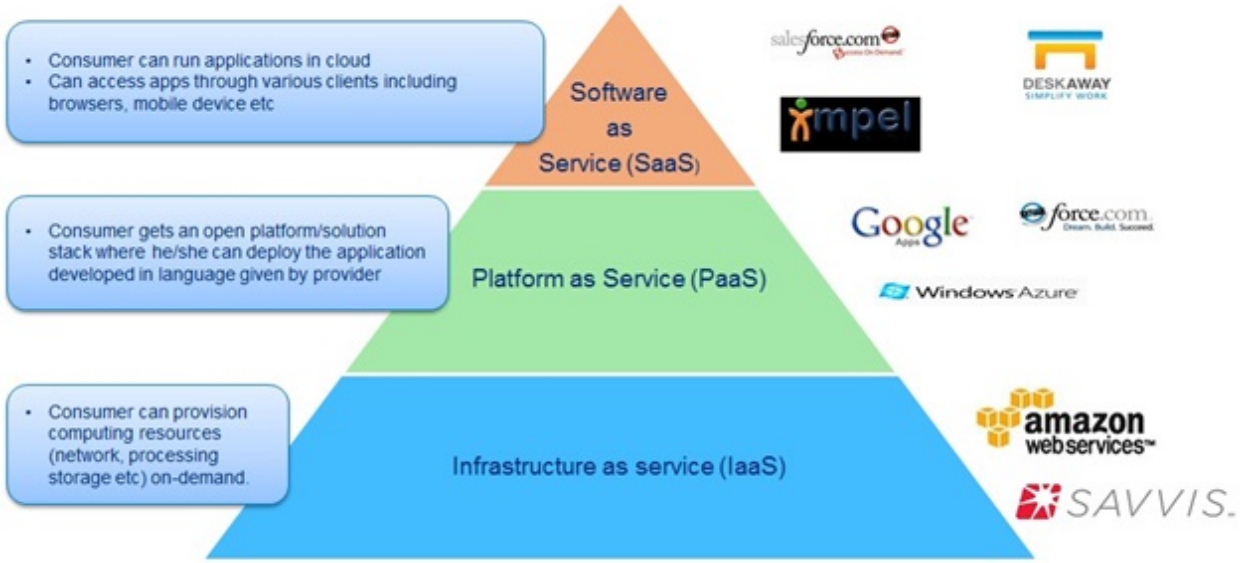
When talking about a cloud computing system, it's helpful to divide it into two sections: the **front end** and the **back end**. They connect to each other through a network, usually the Internet. The front end is the side the computer user, or client, sees. The back end is the "cloud" section of the system. The front end includes the client's computer (or computer network) and the application required to access the cloud computing system. Not all cloud computing systems have the same user interface. Services like Web-based e-mail programs leverage existing Web browsers like Internet Explorer or Firefox. Other systems have unique applications that provide network access to clients. On the back end of the system are the various computers, servers and data storage systems that create the "cloud" of computing services. In theory, a cloud computing system could include practically any computer program you can imagine, from data processing to video games. Usually, each application will have its own dedicated server.

A central server administers the system, monitoring traffic and client demands to ensure everything runs smoothly [4]. It follows a set of rules called **protocols** and uses a special kind of software called **middleware**. Middleware allows networked computers to communicate with each other. Most of the time, servers don't run at full capacity. That means there's unused processing power going to waste. It's possible to fool a physical server into thinking it's actually multiple servers, each running with its own independent operating system. The technique is called server virtualization. By maximizing the output of individual servers, server virtualization reduces the need for more physical machines.

If a cloud computing company has a lot of clients, there's likely to be a high demand for a lot of storage space. Some companies require hundreds of digital storage devices. Cloud computing systems need at least twice the number of storage devices it requires to keep all its clients' information stored. That's because these devices, like all computers, occasionally break down. A cloud computing system must make a copy of all its clients' information and store it on other devices. The copies enable the central server to access backup machines to retrieve data that otherwise would be unreachable. Making copies of data as a backup is called **redundancy**.

1.2 Service Models

There are 3 potential layers of service in Cloud, together termed as "The SPI Model" of Cloud. (Software, Platform, Infrastructure) [5]. These layers are:



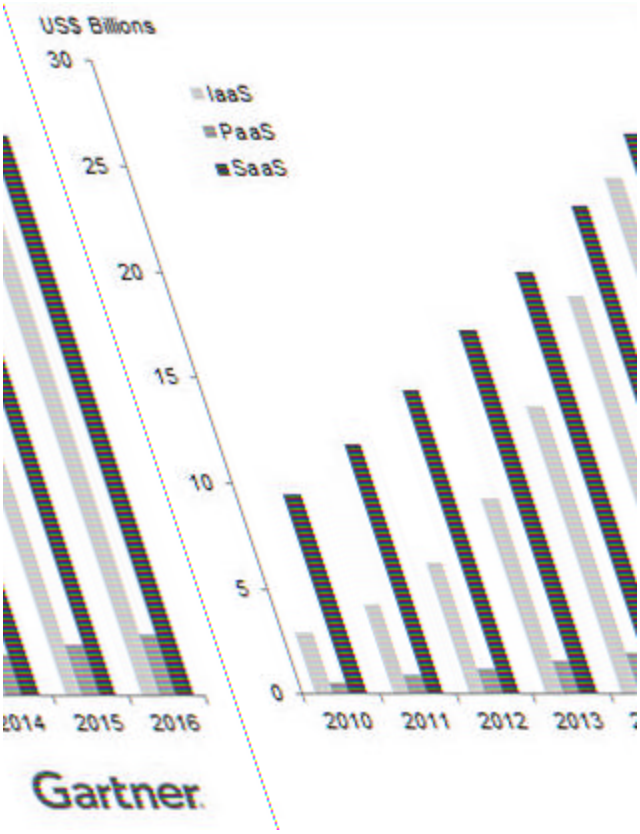
1. SaaS (Software-as-a-Service): General applications such as word processing, Spreadsheet and specialized applications like ERM (Enterprise Resource Management) and CRM (Customer Relationship Management) are delivered over the internet eliminating the need to install or run it on cloud user's system. Sometimes "On-demand Pay-as-you-Go" model is used as a synonym for SaaS. The infrastructure and platform is managed in the background by the Cloud Service Providers (CSPs).

2. IaaS (Infrastructure-as-a-Service): This model provides access to the lowest level resources including servers, data centers or network resources in an easy-to-consume way with no need to purchase them. It is an alluring way to make IT operations scalable and efficient. Also customers get faster service delivery with less cost. But IaaS is also the most susceptible model when it comes to security risks.

3. PaaS (Platform-as-a-Service): It provides self-service on-demand tools, resources, automation and a hosted platform runtime container to facilitate application development and deployment. An installation kit is no longer needed. It is another significant opportunity for enabling the developers to rapidly consume the IaaS built on the PaaS framework. PaaS developers can streamline the building of custom applications and deploy them onto the infrastructure. PaaS automates and provisions for application development, testing, deployment, scaling and monitoring.

1.3 Growing interest in PaaS

PaaS market is in its infancy today and is poised for huge growth as Enterprises are turning to cloud adoption and needs to construct and modernize their development process in a simplified manner. According to Gartner, global market for PaaS will grow from $1.5 billion in 2013 to more than $2.9 billion in 2016 [26]. This tends to become a self-fulfilling prediction.



It comes as no surprise that the PaaS competitive landscape is still in flux but many of the largest enterprise software vendors are on the cusp of entering the PaaS market with their own offerings and this will turn the prophecy for 2016 into reality.

1.3.1 Characteristics of PaaS

There are a number of takes on what constitutes PaaS but some basic characteristics include:

- Multi-Tenant Architecture where a single instance of development application serves multiple concurrent customers.
- Web based user interface creation tools that assist the creation, modification, testing and deployment of different UI scenarios.
- A set of services to develop, test, deploy, host and maintain applications.
- Built-in scalability along with load balancing and failover.
- Integration with web services and databases via common standards.

1.3.2 The PaaS Foredeal

PaaS provides IT with significant benefits:

I. Faster time to market: With access to a broad set of automated tools and technologies, developers can accelerate the designing and production of new cloud applications. This reduction in development cycle time enables more new products to reach the market faster.

II. Re-hosting of legacy applications to run in the cloud: PaaS provides the ability to re-architect the existing applications to fit into the cloud with minimal changes, increased agility and broader reach. IT operational costs are also cut down to a great extent.

III. Lower Costs: By providing the underlying software infrastructure, PaaS reduces the organization costs. The savings on server and storage overhead are also realized. The burden of software maintenance in terms of huge expenses and time is also released as the platform vendor manages all the patches and updates for the hardware and software.

IV. Streamlined application management: In addition to providing commoditized platform such as Caching, PaaS allows you to manage all your applications from a central place eliminating the concern of being outside IT governance. It streamlines application lifecycle management from building to removing it end of life by automating the steps and functionality associated with each milestone.

V. Address application integration issues: Building cloud-aware applications specifically for dynamic environments can increase the cloud adoption internally.

VI. Internal entrepreneurship: developing through PaaS combined with faster deployment on infrastructure online, empowers visionaries in any part of the company.

1.3.3 Working of PaaS

PaaS or Platform-as-a-service is the new wave of computing, computing in the cloud. But how exactly does it work?

A command-line interface (CLI) or an interactive development environment (IDE) pushes an application to the PaaS cloud using a plug-in. PaaS matches its resource requirements and provides a suitable runtime container to host it. PaaS can even instantiate its multiple copies in the same or multiple clouds. Much like a public utility, you pay for what and how much you use.

There are ofcourse, different types of PaaS solutions:

1. Social networking applications like facebook.
2. Business application platforms which are becoming more popular as we speak.
3. Web application platforms such as those provided by Google apps.
4. Computing platforms e.g. Bandwidth, Storage and other traditional computing on a service-by-service basis.

1.3.4 Where PaaS Makes Sense

1. When multiple developers will be working on a development project.
2. To facilitate interaction between external parties and the development process.
3. To support legacy data source applications while leverage that data.
4. Automation of testing and deployment services.

1.3.5 Where PaaS does not make sense

Though PaaS is the buzzword these days but there are certain situations where it may not be ideal.

1. Where the application needs to be highly portable.
2. Where the development process would be impacted by the patent approaches.
3. Where customization of the underlying resources is required.
4. Where concerns are raised about movement to another provider.

1.3.6 Top PaaS market trends in India

- **Supplier explosion:** Current fashion of supply exceeding demand will continue as the internet makes it more convenient to distribute new applications. In that context, boom of PaaS providers will fuel this explosion further.
- **Exponential adoption by SMBs:** SMBs (small and medium businesses) show keen interest in PaaS platform, and this is expected to continue in the near future.
- **Resolution of data security and bandwidth issues:** The concerns around security of data and lack of bandwidth are restricting market growth. Joint effort by PaaS platform vendors and ISVs will result in suitable solution in the coming years.

Chapter-2
LITERATURE REVIEW

2.1 Barriers to PaaS cloud adoption

In spite of the numerous benefits that PaaS (Platform-as-a-Service) cloud brings along with it, there are several concerns and issues which need to be addressed before ubiquitous adoption of this computing paradigm happens.

Simple-to-use and inexpensive PaaS services adds complexity to the businesses of many established companies entering the cloud market, whether as service providers or users. Cloud services brings with them issues such as complex and uncertain security, privacy, tax and related compliance and control consequences for cloud computing users and providers alike. However, these are not impenetrable obstacles and can be transformed into accelerants by the PaaS vendors.

Still these are the very reasons that act as barriers to cloud adoption and make it a long, difficult curve, especially for large organizations and governments. Among the inhibitors are:

1. Corporate culture shock: Keeping focus on differentiating value-add for the organization and separation from the underlying technical implementation, is a big shift for IT managers. The shift from being an internal provider to manager of external service providers is all the more daunting.
2. Loss of control: Cloud users are managing their infrastructure through their relationship with the providers and SLAs (Service Level Agreements) in spite of exercising direct control over the implementation of technical specifications that they define. The skills required for this are not processed by IT organizations yet and so they need to reinvent themselves.

3. Information Security: Cloud makes your data available when you need it and safeguards it from unauthorized access by others, but is it a little more difficult to protect it on cloud than doing so on your own? The opinion about this is still divided [7].
4. Privacy concerns: Cloud complicates how the personally identifiable information of customers, partners and employees is safeguarded. It is mandatory both for fulfilling the legal and ethical requirements of the organization and privacy regulations of the cloud.
5. Regulatory compliance: Virtualization of hardware and software that could theoretically be located anywhere in the world raises a new question about whose rules must be followed. Complying with regulations may be difficult especially when talking about cross-border issues [4]. This affects the performance adversely.
6. Lack of standards: Standards are required to simplify issues like interoperability among cloud providers, portability etc., but few exist. This lack of standards poses obstacles to data recovery, whether it is required for legal discovery or migration from one CSP to another. Customers also risk loss of data as its gets locked into proprietary formats termed as Vendor lock-in; another key concern in PaaS [3].
7. Provider outage: In case of suspension of provider, either you have to wait for the vendor to fix things or you need to have a backup plan for your application, that constitutes a mirror image of your application (and likely its databases as well), running on separate architecture. You may also need some kind of mechanism that can detect failures and automatically swap over to your backup architecture. A PaaS app loses its 100% value if there comes a moment when the user needs to rely on failovers or fallbacks [7].
8. Need of isolated applications: A service or application should not interact with any other on the PaaS unless it is specifically allowed to, in order to prevent application's files and processes from being breached.
9. Data Encryption: Slow system performance is a big time risk in PaaS and encryption of data accounts for it. Encrypting every piece of data eats up organizations' CPU cycles and slow things down.
10. Long term viability: Validity of data in all situations is one thing that needs to be guaranteed. Gartner says: Ask potential providers how we would get our data back and if it would be in a format that we could import into a replacement application.

11. Consumption based Pricing: Though PaaS uses pay-per-use pricing schemes but CSPs still face a daunting task in articulating to potential customers how they create palpable economic value so as to sustain a CSP's service over the long term. Moreover, leveraging cost benefits may not always be possible as organizations do have tax disadvantages.
12. Appropriate abstractions: The central idea behind PaaS is that the CSP provides a development and deployment environment to the user. PaaS abstracts the underlying infrastructure but the main goal of PaaS is to abstract the development details. Unfortunately, building an abstraction at this level is extraordinarily difficult because developers require hands-on control of their development environments.
13. Dark as a Blackbox: PaaS is quite similar to a tunnel where there is light at both the ends, but its pitch dark inside. Once you enter, you cannot see a thing. A global survey of 740 senior IT professionals about cloud computing adoption, Compuware (CPWR) and Research In Action (RIA) found 73 percent of businesses believed their cloud services providers (CSPs) could be hiding infrastructure or performance problems. But business owners do not simply invest in a system which lacks visibility [8].
14. One size may not fit All: PaaS offers great solutions, but this is not obvious that it may work for all developers. Customization is one attribute that a developer may want to have but with PaaS offerings, you get struck with the setup a vendor can provide. Making changes to installations or setups is seemingly an impossible task as of now.

2.2 Meeting Real-world Challenges

PaaS is not a cure-all for every technology challenge but it does provide a range of solutions for the aforesaid problems. Auspiciously, these concerns are not necessarily deal-breakers, as long as business owners evaluate potential PaaS providers carefully. Different CSPs have different ways to deal with these issues and selecting the right one proves as the self-healing mechanism itself.

CSP's Considerations

CSPs are in the enviable position of being first movers in a movement that is rapidly becoming mainstream [9]. They need to incorporate anything and everything to be at the topmost level amongst others. While meeting the customers' requirements they encounter the following challenges:

1. Setting prices sounds simple but becomes complex as the complete economic picture does not deal with the costs alone. To make provisions for access to new technology and use innovative approaches to create a difference should also be factored into the economic equation.
2. Standards, mainly to facilitate interoperability among clouds, are another area of uncertainty. Enforcing real standards may emerge as a winning strategy but it is again a difficult choice because of the trade-off with vendor lock-in.
3. A decision, as to whether the infrastructure is to be built in such a manner so as to enable compliance with rules from different regions or to provide a homogeneous cloud, needs to be carefully taken.
4. The level of transparency to be allowed to the customers is another issue to be considered.

CSPs can transform each of these challenges into a new opportunity and an adoption driver in the near future. Already specialized "compliant clouds" have emerged recently offering compliance with specific regulations and a promise to store and manage data within the borders of a given nation. Moreover, Forrester research projects that CSPs are expected to invest far more in development of their security infrastructure in the coming five years.

Chapter-3

PROBLEM STATEMENT

PaaS facilitates the construction and deployment of applications, frees you from the cost and complexity of buying and managing the underlying hardware/software and provision of hosting capabilities. It provides all the facilities to support the complete life cycle of building and delivering web applications and services. Different PaaS offerings provide different combinations of services to support the application lifecycle.

After deciding that PaaS platform is the appropriate choice according to the requirements, the decision about which PaaS provider to pick is to be made. But getting into this tricky task of navigating the fast-growing PaaS landscape is nerve-wrecking.

3.1 Objectives

Following objectives have been framed for this thesis work:

1. To study the importance, benefits and working areas of PaaS.
2. To study the barriers to adoption of this platform.
3. To study the parameters for evaluation of a PaaS Vendor.
4. To create a framework to facilitate the Selection process.
5. To provide a tool that would point towards the best PaaS providers according to the user's requirements amongst a list of options.

Chapter-4

A FRAMEWORK FOR EVALUATING PaaS VENDORS

As more and more applications find their way to the cloud, users will likely use different types of services and will certainly have numerous Vendors looking for them. In such a scenario, to get onto the path that leads to the best Vendor seems to be quite chaotic. Therefore, to ease the situation a little, it would be convenient to first gather some insights about the key elements of an enterprise PaaS.

4.1 A PaaS Provider Must Haves

1. Application and Services centric lifecycle API: An enterprise PaaS must express itself in terms of applications and services. The REST (Representational State Transfer) API is available for use in any context such as the command line tool, a web console etc. Such an API can lead to a huge jump in productivity [10].
2. Buildpack Support: Every CSP is looked out for this buildpack that supports a broad array of languages and frameworks. This ecosystem ensures version updates and other maintenance activities for virtually any language. Rather than requiring a developer to instruct how to run an application, buildpack can be relied upon to detect the application type, to download and configure the appropriate runtimes, containers and libraries.
3. High performance dynamic routing: Unbounded horizontal scaling is a requirement of many modern applications and the platform must support it. The dynamic router supports the update of deployed app versions, allowing multiple versions to share the same URL. This enables high value capabilities like A/B testing.

4. Linux Container management: Linux containers provide a variety of benefits to a modern PaaS. These include isolation of one application from another, rapid economical provisioning of robust development environments.
5. Data and web services Brokers: To facilitate discovery of available data and web services, a generalized mechanism is required.
6. Role Based Access and Teams: Medium or large teams with role based control to different stages of an application lifecycle are critical for success of an enterprise. Different teams might be responsible for development, testing, staging and deployment of an application.
7. Standards based authentication and authorization: In order to ensure security and protection against unauthorized access, integration with authentication and authorization systems supporting enterprise standards like LDAP and SAML, is required.
8. Active application health management: With thousands of applications running simultaneously on the platform, adjusting and reporting application state throughout its lifecycle is a tedious technical challenge. While IaaS systems often depend on VM monitoring and operator intervention, PaaS systems must eliminate this operational overhead.
9. Multi-provider ecosystem: Development teams should be able to acquire the system on multiple clouds, from more than one provider.
10. Integrated real time logging API: Maintaining application platform, internal network, as well as user action logs is required for healthy functioning of an enterprise PaaS. Therefore, Real time logging aggregation should be a standard service to any application on the platform.

4.2 Parameters for Evaluation

When evaluating and choosing a PaaS provider, the following parameters should be kept in consideration:

1. Programming frameworks: Different CSPs provide access to different programming frameworks. Choosing a programming language and server side technology is the first and foremost step in any application development project. The programming language chosen can strongly influence programming paradigms, tools and components to be deployed. Server side technologies also strongly influence the designing of applications.

Another factor to be considered when choosing a PaaS is how likely you are to continue the current programming framework. If you are strongly committed to it, then working with a language-specific PaaS makes sense but if your development team works with multiple languages and server side technologies then a language-agnostic platform is likely a better fit.

2. Target applications: Typical applications targeted by different CSPs are web-base, general purpose, OS-specific or Storage-specific. The type of application to be developed is also an important parameter while selecting a CSP.
3. Native database systems: Data storage options are increasing. Different CSPs support different database systems. Developers and Database designers typically choose between custom-file based storage and relational databases. When choosing a PaaS, consider your data storage needs. Several of the CSPs can be filtered out in the first step if they do not support the required database systems.
4. Resource pooling: A typical requirement for high performance computing (HPC) applications is effective resource utilization. To run multiple instances of an application, pooling of resources (like memory, database connections, libraries etc.) is required to provide consistency, space efficiency, support faster inter-process communication, reduce network traffic etc.
5. Tool & Application Integration and Support: Support for developer tools (such as Eclipse and Visual Studio) and code management tools (such as Git), is another consideration when evaluating a PaaS vendor. Solid integration helps reduce time and overhead associated with uploading and managing code between developers' machines and the PaaS servers. It also reduces the risk of errors when deploying code to PaaS servers. Several of Cloud customers exploit public Cloud services in extension to their private Clouds. These customers require a seamless integration of their private and public infrastructures. Also consider how an application in the PaaS cloud integrates with other applications? What mechanism is employed to share the data store with other applications? Can data be automatically replicated to another database?
6. Root access: Root access refers to administrative rights to access the hardware. Root access is required sometimes to meet applications' special requirements of hardware configurations.

7. Data backup: CSPs can also be compared on the basis of the data backup services they offer. In some cases, the data-backup is performed on-demand and in others, it is performed automatically.
8. Fault tolerance: Fault tolerance capabilities provided by the CSPs are of special interest for the applications that require large pool of resources, for example HPC applications and applications with long run times. Such capabilities include services for backup, replication, etc.
9. Configurability: Some CSPs have constraints and there needs to be an assessment of those constraints and if they meet the needs of the enterprise. There should be a sense of freedom amongst the users in the way they want to work.
10. Pricing plans/subscription type: Every CSP has a different pricing model, for example, pay-as-you-go model, hourly or monthly packages, discounts, etc.
11. Costs and budgets: The cost of running the application in a PaaS is another important consideration. Also consider options for optimizing the PaaS configuration to balance the needs in order to manage the budget.
12. Cost of data transfer/bandwidth: Applications dealing with large data sets involve numerous inbound and outbound data transfers. The cost of these operations is measured in terms of bandwidth allocated to the cloud user.
13. User/ control interface: Cloud services are accessed by means of a method or tool. For example, GUI, web browsers, APIs, command line tools. The more the tools available, the easier it will be to use the cloud resources [13]. It also adds to the flexibility of the CSP.
14. Security features: Several dimensions of security include email/password security, firewall, backups, authorization, encryption, data protection and failover features etc. One of the most important factors to be considered and it should also be kept in mind to check whether these services are free of cost or not.
15. Service Level Agreement (SLA): A definition of the types and level of services to be provided by the vendor and user's consent to these services against a cost. It covers different factors like performance guaranteed, service outage credit, professional support services etc.

16. Customer support services: Customer support is provided by the CSPs by means of live chat, discussion forums, email, tutorials, etc.
17. Certifications: the security and compliance-related certifications are off typical interest to cloud customers as it depicts the strength of the vendor's services. Examples include PCI, provision of MPLS network service, DDoS protection service etc.
18. Compatibility: The framework, code vault, defect tracking mechanism, versioning and other application lifecycle management tools should be compatible with the enterprise.
19. Risk assessment: Risks, such as vendor lock-in where application developed in the CSP's cloud cannot be moved to another infrastructure, should be assessed carefully and what mechanisms are used by the CSPs to mitigate such risks must be studied carefully. Enterprises should seek out PaaS vendors supporting open protocols and open standards in order to avoid issues such as lock-in.
20. Financial viability: It is significant to make sure that the CSP chosen has financial viability and that it is going to be around.
21. Portability and Interoperability: Does the CSP make it easy to take applications and data out of their platform, be it for the purpose of migration to another cloud (portability) or to be shared among different CSPs (interoperability)? This is one important question that needs to be satisfactorily answered.
22. Load balancing and Scaling: Another essential requirement out of a CSP is automatic load balancing and scaling. The PaaS service monitors the load on your application, can add servers if needed, and distribute work across a set of servers. It also improves the fault-tolerance and can also be configured to manage auto-scaling as well.

Chapter-5
SELECTION PROCESS

Nobody is perfect. One size doesn't fit all. These sayings go for everything and here as well they are making complete sense as there does not exist even a single CSP that fulfils all the requirements of a user and gains a10 on 10 in the evaluation process.

As different CSPs exhibit different characteristics, an enterprise may enter a state of trade-off where it has to forego some of its requirements. In this a scenario one needs to ask a certain set of questions to take the decision about what can be left behind. And a pre-requisite for this introspection is a deep understanding of internal business requirements.

5.1 A Self-Introspection

Analyze carefully and look out for the following:

1. The Stability and Character takes the center stage.
2. Financial Viability (Monetary Stability) can be a deal-breaker.
3. Private, Public or Hybrid?
4. Compatibility of the provider's framework and tools with the Enterprise.
5. Language-specific or Language-agnostic?
6. Database, Framework support?
7. Check for Security in SLA.
8. Consider risks such as lock-ins.
9. On-time delivery?
10. Check out the Pricing model.
11. Transparency?
12. Integration support?
13. The Key Differentiators.
14. Identify functionality provided.
15. Support for legacy apps?

The answer sheet will provide a roadmap for finding exactly the right PaaS vendor.

5.2 Evaluation Framework

PaaS offerings can be evaluated under the following categories:

1. Cloud Dimensions:
    - Deployment Model (private, public or hybrid)
    - Service Model (SaaS, PaaS or IaaS)
2. Overview:
    - Product Description
    - Notable Customers
3. Programming Model:
    - Server Os types
    - Runtimes
    - Frameworks
    - Middleware
    - Native Databases
    - Native Services

4. Pricing Model:
    - Base Plan Price
    - Virtual CPU cores
    - RAM
    - Disk Space
    - Subscription Options

5. Specifications
    - Development Tools
    - Control Interface

6. Features
    - Features
    - Writable Storage
    - Remote Access
7. Standards
    - Administration and Compliance details
8. Services:
    - Support Services available.
    - Email hosting

9. Company Information
    - Year founded
    - Website

Cloud Dimensions
Deployment Model
Service Model

Overview
Product Description
Notable Customers

Company information
Year Founded
Website

Programming Model
OS Server types
Runtimes
Frameworks
Middleware
Native Databases

Services
Support Services available
Email hosting
Native Services

Pricing Model
Base Plan Price
Virtual CPU cores
RAM
Disk Space
Subscription Options

Cloud Features
Features
Writable storage
Remote Access

Administration and Compliance details

Specifications
Development Tools
Control Interface

5.3 A Few PaaS Providers

- **Amazon AWS**

Although Amazon Web Services is primarily an IaaS, many of the services available in AWS are similar to PaaS offerings. You can use the platform services available in AWS without having to create or maintain your own application servers. AWS readily supports Java, Python, Ruby, Perl and other languages. Oracle, MySQL and SQL Server can be set up and managed, but AWS offers RDS web service as well, which eliminates database administration tasks. Developers can take advantage of Amazon Elastic Beanstalk for automatic load balancing, auto-scaling, and application health monitoring.

Key Features:	No limit to the languages, databases or server side technologies you can install and run.
Limitations:	Requires more management overhead than other PaaS options.
Pricing:	Complex mixture based on instances, storage, application services and data egress charges. Amazon, however, offers a monthly calculator to help estimate your costs.
Bonus:	New users can get 750 hours, 30GB storage and 15GB bandwidth for free with AWS's Free Usage Tier.

Features:

1. IaaS & PaaS in One: Amazon Web Services (AWS) is probably not the first cloud service that comes to mind when you think of Platform-as-a-Service. AWS is known primarily as an Infrastructure-as-a-Service platform (IaaS) and with good reason. The key distinguishing feature between an IaaS and a PaaS is the type of service offered. In an IaaS customers typically work with virtual machines that they configure themselves; in a PaaS customers work with services created and maintained by the PaaS provider. AWS has a mix of both.
2. AWS Customization: One way to distinguish PaaS providers is by the languages and server side technologies they support. Amazon is largely an IaaS and hence there is virtually no limit to the languages and server side technologies you can install and run**.** If you want to run .NET, PHP, Django or Spring you can do that as well [22]. Talking about database technologies, along with others you have an option though with AWS, make use of Amazon's Relational Database Service (RDS)**.** Rather than create your own database

instance, you can use the RDS web service to provide access to an Amazon managed database. There are a number of advantages to this approach. Amazon takes care of all the database administration tasks, like patching software and creating backups. RDS also supports multi-zone replication so you get high availability features without the level of administrative overhead you would have if you ran databases in your own instances. The RDS service also allows you to provision IOPS storage for those applications that need consistent I/O performances. Pricing for RDS is based on the size of the database instance you choose. A micro-DB instance without replication costs $0.025 per hour; and a quadruple extra large DB instance will cost $3.77 per hour.

3. Load Balancing and Scaling: Another common feature of PaaS offerings is automatic load balancing and auto-scaling**.** The PaaS service monitors the load on your application, can add servers if needed, and distribute work across a set of servers. AWS offers the Elastic Load Balancing service that performs similar functions. Since any traffic to the domain name mapped to the load balancer is distributed across the cluster, you gain improved fault tolerance. You can also configure the load balancing service to manage auto-scaling as well. Elastic load balancing is in some ways a small step toward PaaS like services. If you want even more PaaS-like functionality while still retaining a substantial degree of control over the underlying infrastructure, then you may want to consider using Amazon Elastic Beanstalk. Like more traditional PaaS services, you upload your application and publish your application to Amazon Elastic Beanstalk. .NET developers can use AWS Toolkit for Visual Studio to push their applications. Java developers can upload Web Application Archive (WAR) files. Developers working with Python, Ruby and other languages can create a Git repository for deploying their code. Amazon Elastic Beanstalk manages load balancing and virtual machines, but developers can still control lower level infrastructure configuration if needed. For example, developers can specify virtual machine instance types, a particular relational database for storage, replication for an application in multiple zones, and modify environment settings if needed. An added benefit for developers is that there is no additional charge to run Amazon Elastic Beanstalk. Charges are based on the underlying resources, such as EC2 instances and storage.
4. PaaS-Like Services with AWS: AWS includes a number of services commonly used in large scale distributed applications, including search, workflow, and messaging queues. These are

all services you can establish for yourself in the AWS cloud but if you prefer a more PaaS-like option, you have one available to you. For example, the Amazon Cloud Search service allows you to offer search services on your site or application without having to administer a Lucene or Solr search server yourself. The service will automatically scale the number of search instances to meet the demand of current load. As with other Amazon services, you pay for what you use. In the case of search, there are charges based on the number of instances running search services, the number of documents indexed, and the number of search requests. PaaS providers typically offer a mix of compute services, storage, and application service. All of these are available in the Amazon AWS. Developers may be hesitant to move their application to a PaaS that does not allow them some level of control over the underlying infrastructure. The process model established by a PaaS may not be a 90% solution to developers' needs but the missing 10% is enough to dissuade the developer from deploying to the PaaS. By providing PaaS-like services, such as RDS, Elastic Beanstalk, search, workflow and messaging queues, developers can have a mix of IaaS and PaaS benefits together.

- **Google App Engine**

Google App Engine is designed for distributed web applications and developers using Java, Python, PHP and Go [24]. The Java environment supports other languages that make use of the JRE and there is a SDK for each of the four main supported languages as well as a plugin for Eclipse. The PaaS offers managed infrastructure and runtime environments that are guaranteed to scale, but only if the applications fit the restrictions of Google App Engine. The Datastore, a transactional, schema-less data store based on key-value pairs, handles the complex management of data that's accessible to multiple machine instances.

Key Features:	Google App Engine uses a sandbox model which isolates processes thereby reducing the risk that a rogue process on a physical server will disrupt operations of other processes on that server.

Limitations:	Programming languages are limited to Java, Python, Go and PHP.
Pricing:	Based on usage. $0.08/hour per on-demand instance or $0.05/hour for a reserved instance. The Datastore is billed at $0.18/GB per month; bandwidth costs $0.12/GB. Other service costs may apply.
Bonus:	The first 28 instance hours, 1GB of storage and 1GB of inbound/outbound traffic per application are free each day.

Features:

1. Back to Coding: Google App Engine is a Platform-as-a-Service cloud offering designed for distributed Web applications. As with other PaaS offerings, the Google App Engine manages many of the implementation details about maintaining virtual servers, operating systems and development tools. Some PaaS options, such as RedHat OpenShift, offer IaaS-like features that allow users to customize their environment to a greater degree. This is not the case with Google App Engine. With Google's PaaS, you get managed infrastructure and runtime environments that are guaranteed to scale as long as your applications follow the Google App Engine playbook. For developers who want to focus on their application code without the routine hassles of systems and have applications that are either interactive, modeled on processing operations in a task queue, or some combination of the two, then Google App Engine is a good option for getting back to coding and away from system administration. If your application does not fit the restrictions of Google App Engine, you would probably be better served by another PaaS rather than trying to force Google App Engine into your process model. The first step in assessing Google App Engine for your application is to review the list of supported programming languages. The Java environment supports other languages that make use of the JRE, e.g. Ruby through the JRuby on Google App Engine project. There is a SDK for each of the four main supported languages as well as a plugin for Eclipse. Java and Python are fully supported but Go and PHP are still considered

experimental. Google App Engine runs version 2.7 and does not support many C language extensions. Fortunately, for compute intensive operations developers can turn to the NumPy library. Web developers can make use of the Django framework for their applications. Google App Engine limitations are not solely linked to languages.

2. Google App Engine Customization: Programs run within a sandbox that is logically isolated from other applications, the operating system and underlying hardware. They cannot write to the local file system and can access other machines on the Internet through URL fetch and email functions provided in the environment. Another key restriction is that code only runs in response to a Web request, an event in the task queue or a task scheduled as a cron job. When the code does run in response to one of these triggers, it must return a result within 60 seconds. At first glance these may seem like arbitrary restrictions, especially to those who are accustomed to programming with the design patterns and models used on infrastructure they control. The sandbox model is designed to isolate processes and that reduces the risk that a rogue process on a physical server will disrupt the operations of other processes on that same server. The 60 seconds limit on computation encourages modular, limited units of computation on the Google App Engine. If you need to run long extraction, transformation and load operations on large data sets, then an IaaS service is probably a better option for you. Alternatively, you could design your program to break your data set into subunits that can be processed within the 60 second limitation. Programs designed this way are well suited for the distributed Google infrastructure. Keep in mind that autoscaling is provided only for applications with low latency, i.e. respond to requests in less than 1 second. Google App Engine includes a transactional, schema-less data store based on key-value pairs**.** Known as the Datastore**,** this service handles complex management of data that must be accessible to multiple machine instances. Entities are organized hierarchically into entity groups. Queries within an entity group are always up to date and consistent but queries over multiple entity groups may return stale data since these queries depend on replicated data that is only eventually consistent. Developers and application administrators can query entities in the Datastore using that runs in the administrator console and in the Python runtime. Google App Engine developers also have access to relational database services in Google Cloud SQL, which is based on MySQL. If you need to store files or large objects, you can make use of Google Cloud Storage. Developers manage their storage and applications through

the Google App Engine Dashboard. Developers have access to details about their instances, logs, cron jobs, and task queues. The dashboard also presents details about the Datastore, including details on indexes, text searches, memcache and Datastore statistics. Access Control and permissions are based on user's Google account, which can be either a Gmail account or a Google Apps account. Google App Engine now also allows authentication based on OpenID but this is still considered an experimental feature. Users can be restricted to members of your Google Apps domain as well.

3. Google App Engine Pricing: The services are billed based on usage. Google offers for free up to 28 instance hours, 1 GB of datastore storage as well as 1 GB of inbound and outbound traffic per application per day. After you exceed your free quotas you will be billed $0.08 per hour per on-demand instance or $0.05 per hour for a reserved instance hour. The Datastore is billed at $0.18 per GB per month for storage in excess of 1 GB and outgoing bandwidth is $0.12 per GB. Other services, such as object storage, email API, and dedicated memcache incur per usage charges as well.
4. Google App Engine Gets You Back to Coding**:** Google App Engine is a development stack that includes support for multiple languages, caching, persistent data storage and basic access controls. If you can design your application within the restrictions imposed by the PaaS, you will be able to leverage key components of highly scalable applications. Google App Engine is a good option for developers who want to focus on coding their applications and not worry about system administration.

➢ Engine Yard

Engine Yard is designed for developing web applications using Ruby on Rails (RoR), PHP and Node.js. Engine Yard provides a set of services on top of Amazon AWS. In fact, Engine Yard runs its platform in the Amazon cloud. Engine Yard takes care of key operations tasks such as performing backups, managing snapshots, managing clusters, administering databases and load balancing.

Key Features:	Dedicated instances with no multi-tenancy at VM level. More control over your virtual machine instances than other PaaS providers. Both Public and Private Git repositories are integrated.

Limitations:	Supported languages are limited to Ruby, JRuby, Node.js, and PHP.
Pricing:	Prices range from $0.05/hour per instance to $2.19/hour per instance, depending on your configuration.

Features:

1. Orchestration and Management: Maintaining development and production environments rarely provides for a competitive advantage. Software development firms are known more for their products than their development and operations management procedures. Similarly, developers are typically judged on how well they deliver quality solutions on time and on budget. It's no surprise that some developers are willing to turn over responsibility for platform management to someone else. Engine Yard is a Platform-as-a-Service designed for Web application developers who want the advantages of cloud computing without all the operations management responsibility. It provides a set of services on top of Amazon AWS. Engine Yard runs its platform in the Amazon cloud and manages the application stack for you. Its customers can deploy instances in eight Amazon regions using normal, high memory and high CPU instances. When a customer starts an instance in Engine Yard, multiple software components are configured and started according to their particular needs. In addition to Ruby, PHP, and Node.js, the stack includes HAProxy load balancer, Ngnix and Rack Web servers, Passenger and Unicorn app servers, as well as MySQL and PostgreSQL relational databases [25]. Developers have the option of running these same components in the Amazon cloud themselves so the value of Engine Yard rests more with orchestration and management than with providing software components. The value-add services start with VM configuration and imaging, configuration management, resource optimization and auto scaling. Once systems are configured and deployed, we need to perform standard operations management procedures. Engine Yard takes care of key operations tasks such as performing backups, managing snapshots, managing clusters, administering databases and load balancing.
2. Engine Yard Customization: Engine Yard gives customers more control over their VM instances than some other PaaS providers. Its instances are dedicated instances that are not shared with other customers. Since there is no multi-tenancy at the virtual machine level, each customer can have greater control over instances without creating significant additional

risk to others. There are a number of ways to customize your configuration in Engine Yard. The PaaS supports Chef Configuration management recipes which you can customize. This give some level of operations management to customers who need additional components or want more control over some aspects of the standard Engine Yard stack. For those that need more control over the deployment process, Engine Yard provides deployment hooks. These are Ruby scripts that are run during the deployment process to perform additional operations. Deployment hooks can invoke shell commands as well. These scripts also have access to several deploy hook variables such as path names and account names. Although PaaS providers can take on some of the operations management tasks associated with development, managing code is one task that will likely remain well within the control of developers. If you want a managed service for an application not supported by Engine Yard you might find what you need from one of the add-on services available through the PaaS. For example, if MySQL and PostgreSQL don't meet your database needs, you have the option to work with several vendors providing MongoDB, Riak and Hadoop data services. If you need additional performance management tools, you have the choice of application management, load and performance testing, user management and log management services.

3. Engine Yard Pricing: The pricing model used of Engine Yard is similar to Amazon's "pay for what you use" model. At the low end, a small instance with 1 ECU, 1.7GB RAM and 160 GB of non-persistent storage costs $0.05/hour. A high memory XL instance with 6.5 ECU, 17.1 GB RAM and 420 GB non-persistent storage is $0.68/hour. At the high end, a 26 ECU, 64.8 GB RAM, and 1,690 GB non-persistent storage will cost you $1.99/hour. There are several other configurations available as well. If you use additional Amazon services, such as S3 storage or reserved IP addresses then Engine Yard will pass on Amazon's charges plus a 20% fee. As an incentive to try the PaaS, Engine Yard offers 500 hours of free service which can be spread out over six months.

   The prices listed include standard support but a premium support option is available as well. The standard support services include phone support, access to community forums, assistance with upgrades and configurations, and database administration support. The premium support plan includes all the features of the standard plan plus application analysis, proactive monitoring and response to alerts, 24x7 production emergency support and a 99.9% SLA. Premium support costs $150 per month plus a charge based on monthly usage. The

additional support fee is 15% of monthly charges when spending less than $5,000 per month but drops to 9% when monthly charges exceed $50,000 [25].

4. Engine Yard Gives You More Control: Engine Yard is a PaaS for Web application developers that strike a balance between performing operations management tasks for developers and giving developers control over their development and production environments. Engine Yard provides standardized open platforms that run in the Amazon cloud. This PaaS is designed to help you get your platform up and running (and keep it running) without getting in your way.

➢ Windows Azure Cloud Services

Again with Windows Azure, Microsoft is blurring the lines between IaaS and PaaS. Windows Azure Cloud Services supports .NET, Node.js, PHP, Python, Java and Ruby. In addition to software development kits, developers can use Visual Studio for creating and deploying applications. Developers can choose between a SQL Database, Tables and Blobs when it comes to persistent storage. Applications are administered through the Windows Azure dashboard or through a command line interface.

Key Features:	Since Windows Azure is an IaaS and PaaS in one, developers can mix and match IaaS components with PaaS offerings giving you more control.
Limitations:	Minimalist administration portal.
Pricing:	Prices range from $0.02/hour (768MB of RAM & 1 shared virtual core) to $0.64/hour (14GB of RAM & 8 virtual cores). Pricing for high memory machines is also available.
Bonus:	A Free 30-day trial with a limit of up to $200 is available for new users.

Features:

1. IaaS & PaaS in One: Windows developers were fortunate to have an early Platform-as-a-Service (PaaS) in the form of Microsoft Windows Azure. Microsoft eventually added

Infrastructure-as-a-Service (IaaS) functionality to Azure and included Linux servers in the IaaS line-up as well as Windows operating systems. Cloud providers like Microsoft, Amazon and Google are trying to blur the lines between PaaS and IaaS with PaaS-like offerings, such as databases, messaging queues and caching, as well as IaaS core services, such as virtual machines. Developers are free to mix and match IaaS components with services found in PaaS offerings making it even easier to balance the need for control offered by IaaS with the higher-level services of PaaS.

2. Azure Cloud Services Customization: Windows Azure Cloud Services has software development kits (SDKs) for several languages and server side technologies. The SDKs include tools to help you run application in the Windows Azure PaaS. Applications are deployed with three configuration files: a service definition file, a service configuration file, and the third service package. The service definition file defines the service model characteristics, such as the number of roles. Windows Azure defines two roles: a Worker role and a Web role. They are both Microsoft Server operating systems but the Web role includes an IIS server to handle Web-based requests. The configuration files also include supporting details to enable services such as SSL and Remote Desktop. In addition to the SDKs, Visual Studio can be used for creating and deploying applications. A benefit of using Visual Studio is that is supports local debugging of application code. Visual Studio includes a trace facility, storage account diagnostics and troubleshooting features. Visual Studio includes version control services as well. The Windows Azure management console provides an application deployment function. It requires a developer to specify a URL and a region to deploy the application. When you deploy code to an application you have the option of deploying to a staging area or to production. When deployed to the staging area, the application is accessed through a URL based on the applications globally unique identifier (e.g. GUID.cloudapp.net) rather than a user-friendly DNS name [18]. You can administer your application through the Windows Azure dashboard or through a command line interface. Microsoft has a Windows Azure PowerShell utility as well as a command-line interface for the Mac OS X and Linux operating systems. Once your application is deployed, you can scale resources to meet demand. If you prefer to manually scale Web and Worker servers, you can do that through the management console. Manual scaling is the default mode but automatic scaling is available as well. Automatic scaling is based on either the number of

queue messages or average CPU usage. When average CPU or the number of queue messages cross a threshold, the number of servers in a particular role are increased or decreased as needed. There may be cases where you may want to monitor your application even though you turn over most administrative tasks to the cloud provider. Windows Azure Cloud Services provide a minimal set of performance metrics, including CPU percentage, data in and out, along with disk read/write throughput. Additional metrics are available if you configure your roles in verbose monitoring mode.

Developers have three basic options with regards to persistent storage: SQL Database, Tables or Blobs. SQL Database is a Database-as-a-Service offering that provides many of the features of SQL Server**.** SQL Database supports Transact SQL so developers familiar with coding for SQL Server should find SQL Database familiar [23]. Note, not all Transact SQL procedures are supported so familiarize yourself with the subset of Transact SQL supported.

If you prefer to use a NoSQL database such as MongoDB but do not want to manage a data store service, then the Tables data storage option may be the best fit for you. Tables supports up to 200 terabytes of simply structured data. Tables are manipulated using either REST commands or API functions. Data in Tables are automatically partitioned to enable scalability. Blobs, or binary large objects, are unstructured storage objects designed to store arbitrary binary data. This data storage system supports up to 200 TB of data and is also accessed through REST or API commands.

3. Azure Cloud Services Pricing: Pricing for Windows Azure Cloud Services is based on the size of instances running your application. There are five standard instances: extra small, small, medium, large and extra large. The small instance includes 768 MB of RAM and a shared virtual CPU core at a cost of $0.02 per hour. A medium instance includes 3.5 GB or RAM and 2 virtual CPU cores at a cost of $0.16 per hour. The largest of the standard instances, the extra large, comes with 14 GB of RAM and 8 CPU virtual cores at a cost of $0.64 per hour. If the standard servers are not sufficient for your memory intensive applications, Windows Azure Cloud Services offers the A6 and A7 servers with 28 GB or RAM and 56 GB or RAM, respectively. The prices for the high memory machines are $0.90 per hour for the A6 and $1.80 per hour for the A7. Like other PaaS vendors, Microsoft provides additional services for large scale applications including:

- Business analytic services, such as SQL Reporting and HDInsight, a Hadoop service;
- Queues and services bus for messaging;
- Caching and Content delivery network (CDN) services to improve application performance;
- Identity management services;
- Media services.

Microsoft Windows Azure Cloud Services offers a simple PaaS model based on Web and Worker instances, three different persistent storage services, and support for commonly used programming languages and web development frameworks.

➢ Red Hat OpenShift

Red Hat OpenShift is based on open source applications and offers a wide variety of languages, databases and components. The PaaS is highly customizable and offered in three forms: OpenShift Online (a cloud-based hosting service), OpenShift Enterprise (a private PaaS that runs in your data center) and OpenShift Origin (the open source application hosting platform). OpenShift automates system administration tasks such as virtual server provisioning, configuration and scaling and supports git repositories for code management.

Key Features:	Developers can interface with OpenShift through a web console, the command line or through an integrated development environment.
Limitations:	Works well with Git, but non-Git deployments might require additional steps.
Pricing:	OpenShift Online pricing is based on the number and types of components (called gears) you deploy. Gear prices range from $0.02/hour to $0.10/hour, depending on the size: 512MB (small), 1GB (medium), or 2GB (large). The Silver support plan is $20/month plus usage costs.

Bonus:	A limited number of resources are available as a trial; 1GB of storage per gear and 3 small gears are free.

Features:

1. Cartridges, Gears & Nodes: Red Hat may be best known for its enterprise Linux, but it is not letting the movement to go without a response. As a software company with a business model based on supporting open source software in the enterprise, it is no surprise that its Platform-as-a-Service offering is based on open source applications. It is also not surprising that Red Hat OpenShift offers a variety of languages, databases and components to developers. After an extended beta period, Red Hat opened OpenShift to the public in June 2013 with a wide array of development options. OpenShift is offered in three forms:
   OpenShift Online -- a cloud based, hosting service for application developers
   OpenShift Enterprise -- a PaaS platform designed to run within an organization's data center.
   OpenShift Origin -- the open source application hosting platform underlying OpenShift Online and OpenShift Enterprise [19]. Like other PaaS platforms, OpenShift is also designed for developers. Many typical systems administration tasks are automated so developers can spend more time on code and less time on configuring operating systems and installing libraries. The most important of the automated administration tasks are virtual server provisioning, configuration, and scaling. Developers work with applications and components. In OpenShift, an application is a combination of code, configurations and application components called cartridges. Cartridges are high level services, such as Web servers, databases, as well as logging and monitoring tools. Cartridges are logically isolated from one another so multiple cartridges can run on the same server. OpenShift uses the terms "gears" and "nodes" to refer to the abstract structures that isolate components. In OpenShift parlance, nodes hold gears that contain one or more cartridges. A broker manages provisioning and application management processes and communications with nodes over a message bus.
2. OpenShift Customization: One of the most immediately apparent features of OpenShift is the number of component options**.** Component options span the application stack from front end to backend services**.** Developers can choose to code their applications in Java, PHP, Perl,

Ruby, Python and Node.js. OpenShift supports server side technologies commonly used with these languages. For example, Java developers can choose between JBoss Enterprise Application Server and JBoss Enterprise Web Server/Tomcat according to their needs. Python developers can build on frameworks including Django, Pylons, TurboGears, Bottle and Zope. Ruby on Rails and Sinatra are the supported frameworks for Ruby developers. For applications that require a relational database, developers have the choice of using MySQL or PostgreSQL. If a NoSQL data store is a better option, developers can work with MongoDB. The Perl mantra "there is more than one way to do it" is applicable to Red Hat OpenShift. In addition to multiple languages and application stacks, there are a few different ways to interface with the OpenStack platform. Developers create and manage applications from a Web console, the command line or through an integrated development environment. With a simple local installation, developers can use the Red Hat command line tools to create applications, clone copies of git repositories, display status information, and add cartridges. Eclipse developers do not have to leave the IDE environment if they don't want to. Installing the JBoss Tools in Eclipse gives developers the ability to deploy and work with applications within the development environment. OpenShift supports the use of git repositories**.** Git is so well integrated, developers can deploy an application with a single command that specifies needed cartridges (e.g. language and database) and a URL to a git repository with application code. The Git model of source control works well with OpenShift. OpenShift supports a number of hooks, which are points in the deployment process when custom scripts can be run**.** Scripts can be run at five stages: pre-receive, pre-build, build, deploy, and post-deploy. These scripts give developers control over ancillary processes that may need to run at varying points in the deployment process. For developers that prefer to work with a continuous integration model, OpenShift supports Jenkin**s** as well. OpenShift cartridges are the basis for integrating other components into the PaaS environment. In addition to the languages, databases and application server cartridges, OpenShift has cartridges for Jenkins for continuous integration, Cron for scheduling jobs, and SwitchYard for managing service oriented applications.

3. OpenShift Pricing: The Pricing model is based on the number and types of gears (i.e. components) you deploy. Red Hat offers a limited amount of resources for free but they are enough to run a small Web site. The free tier includes 3 small gears, each of which is enough

to run a scripting language or a MySQL database, and 1 GB of storage per gear. For applications requiring more resources, developers can sign up the Sliver Plan for $20 per month. This plan includes professional Red Hat support and the ability to run up to 16 small or medium gears. Medium gears are required to run Java, PostgreSQL and MongoDB. The Silver Plan includes 3 small gears for free and after that customers pay $0.04 per hour for small gears and $0.10 per hour for medium gears. With the Silver Plan you can store up to 6GB per gear. Red Hat OpenShift is currently available in the United States, Canada and Europe.

4. RedHat OpenShift Gives You Lots of Options: As with any PaaS, developers have to work within the logical abstractions established by the application hosting platform. In the case of Red Hat OpenShift, this means understanding the roles of cartridges and gears. OpenShift cartridges enable the integration of other components into the PaaS environment. Developers who already use git will appreciate the streamlined integration with the source code management system.

Chapter-6

A COMPARATIVE STUDY

A Comparison amongst a few PaaS providers is conducted to facilitate the Evaluation & Selection Process.

6.1 Comparison Snap Shots

Companies / Parameters	AWS Elastic Beanstalk	Salesforce1 Platform	AgileApps Live Platform
Cloud Dimensions			
Deployment Model	Public Cloud	Public Cloud	Public
Service Model	PaaS, IaaS	PaaS	PaaS
Overview			
Product Description	Amazon Elastic Beanstalk of AWS allows developers to host, deploy, and manage applications in the cloud.	PaaS platform offerings of Salesforce.com include Social Application Platforms, Raw Compute Platforms, Web Application Platforms, and Business Application Platforms	AgileApps Live is a business process management and application platform-as-a-service (bpm + aPaaS) that allows the subject matter experts, as well as developers, to visually build and deploy process-driven, application solutions.
Notable customers	Mendeley Snapdeal Adobe Twiitch	Facebook WellsFargo Ge Philips	HP Nielsen Newtex Gannett

Company Information			
Year Founded	2011	2009	2008
Website	aws.amazon.com	salesforce.com	agileappslive.com

Table1: Comparison of Amazon AWS Elastic Beanstalk, Salesforce1 and Agileapps Live on the basis of their Company Profile.

Companies / **Parameters**	**AWS Elastic Beanstalk**	**Salesforce1 Platform**	**AgileApps Live Platform**
Programming Model			
Server Os types	Windows Amazon Linux		C
Runtimes	.NET Java Python Ruby PHP Node.js	Ruby Python Scala Java Node.js Clojure	Java C Perl PHP .NET
Frameworks	Django Spring	Spring RoR	
Middleware	HTTP Server IIS Passenger TomCat Nginx		Tomcat
Native Databases	Amazon RDS Amazon DynamoDB Amazon SimpleDB Microsoft SQL Server Oracle IBM DB2 Informix	MySQL PostGreSQL SQLite	MySQL
Specifications			
Development Tools	Git Java Web Application Archive		
Control Interface	Command Line	API GUI	API
Pricing Model			

Base Plan Price	$0.02 per Hour	$25 per Month	Free Trial
Virtual CPU Cores			
RAM	1 GB		
Disk Space	8 GB		
Subscription options	Hourly	Monthly/Fixed	On Demand

Table2: Comparison of Amazon AWS Elastic Beanstalk, Salesforce1 and Agileapps Live on the basis of their Architecture.

Companies / **Parameters**	**AWS Elastic Beanstalk**	**Salesforce1 Platform**	**AgileApps Live Platform**
Services			
Native Services		Redis	
Support services available	Forums	Phone	
Email Hosting	No	No	No
Cloud Features			
Features	Auto Scaling Capacity Provisioning Load Balancing Application Health Monitoring	Horizontal scaling Load Balancing	Multi-tenant architecture Load Balancing 99% Uptime Data Recovery
Writable Storage	No		
Remote Access	SFTP SSH		
Standards			
Administration and Compliance details	FedRAMP FIPS 140-2 HIPAA Compliant ISO 27001 Certified PCI Compliant SOC 2 SSAE16 Audited Facility		

Table 3: Comparison of Amazon AWS Elastic Beanstalk, Salesforce1 and Agileapps Live on the basis of their Working Culture.

Companies / Parameters	Windows Azure	Openshift	Cloud Foundry
Cloud Dimensions			
Deployment Model	Public Cloud	Public, Private	Public Cloud
Service Model	PaaS, IaaS	PaaS	PaaS
Overview			
Product Description	Microsoft Azure is a cloud platform and infrastructure, for building, deploying and managing applications and services through a global network of Microsoft-managed datacenters.	Based on open source applications and offers a wide variety of languages, databases and components. The PaaS is offered in three forms.	Cloud Foundry is the industry's Open PaaS and provides a choice of clouds, frameworks and application services.
Notable customers	Icertis Webzen Telenor Toyota Milliman Bmw Latin America	Digistarters Telefonica Digital	SAP Baidu OpenCredo Verizon
Company Information			
Year Founded	2010	2011	2011
Website	microsoft.com	openshift.com	cloudfoundry.org
Specifications			
Development Tools	Microsoft Visual Sudio Git Eclipse	Git	
Control Interface	Web-Based Application/ Control Panel API Command –Line	Web-Based API Command Line	API Command Line
Pricing Model			

Base Plan Price	$0.02 per Hour	$0.02-$0.10 per Hour	$0.02 per Hour
Virtual CPU Cores		3 Gears	
RAM	768 MB		
Disk Space	20 GB	1 GB	1 GB
Subscription options	Hourly	Free Plan Hourly (Otherwise)	Hourly

Table 4: Comparison of Microsoft Azure, RedHat Openshift and VMWare CloudFoundry on the basis of their Company Profile.

Companies / **Parameters**	**Windows Azure**	**Openshift**	**Cloud Foundry**
Programming Model			
Server Os types	Linux Windows	Linux	Linux
Runtimes	.NET Java Python Ruby PHP Node.js	Java Python Ruby PHP Node.js Perl	Go Java Python Ruby PHP Node.js Scala
Frameworks	Drupal Symfony	Rack Django Node.js PSGI Django Ruby on Rails	Play
Middleware	Windows Server	Jboss Tomcat	Tomcat Jetty Jboss Resin Glassfish
Native Databases	Microsoft SQL MySQL MongoDB OracleDB	MySQL PostGreSQL MongoDB	MySQL PostGreSQL MongoDB
Services			
Native Services	Hadoop	Jenkins	Couchbase Filesystem
Support services available	Forums Live Chat Phone	Customer portal	
Email Hosting	No	No	No
Standards			

Administration and Compliance details	FedRAMP HIPAA Compliant ISO 27001 Certified PCI Compliant SOC 2 SSAE16 Audited Facility Network Uptime Guarantee		

Table 5: Comparison of Microsoft Azure, RedHat Openshift and VMWare CloudFoundry on the basis of their Architecture.

Companies / **Parameters**	**Windows Azure**	**Openshift**	**Cloud Foundry**
Cloud Features			
Features	Auto Scaling Vertical scaling Horizontal scaling Load Balancing Block Storage Cloud Storage Disaster recovery Database as a Service Messaging services Content delivery Network Deploy Servers	Fast Service Delivery Sreamlined Application Management Automated Scaling	Horizontal Scaling Vertical Scaling Auto scaling Application Health Monitoring Maintaining Logs
Writable Storage	No	No	No
Remote Access	FTP		SSH

Table 6: Comparison of Microsoft Azure, RedHat Openshift and VMWare CloudFoundry on the basis of their Working Culture.

Companies / Parameters	Google App Engine	CloudBees	Engine Yard
Cloud Dimensions			
Deployment Model	Public Cloud	Public Cloud	Public Cloud
Service Model	PaaS	PaaS	PaaS
Overview			
Product Description	Google App Engine (often referred to as GAE or App Engine) is a platform as a service (PaaS) cloud computingplatform for developing and hosting web applications in Google-managed data centers.	This Platform features Jenkins in the cloud for its core development service, DEV@cloud, Also offers Jenkins Ent., a subscription-based offering built on top of Jenkins CI.	This platform uses open source technologies and ideologies to automate quick and effective deployment configuration, and management of mobile and web applications.
Notable customers	Appogee Best Buy CloudLock Ubisoft News Limited	Choose Digital Global Bank jclouds Neyflix	3 Play Media Access Health Group Afar Audi Appboy
Company Information			
Year Founded	2008	2010	2006
Website	appengine.google.com	cloudbees.com	engineyard.com
Specifications			
Development Tools	Eclipse IntellIJ Maven Git Jenkins PyCharm	Eclipse Git SVN Maven	Git
Control Interface	Web-Based Application/ Control Panel API	API	Web-Based Application/ Control Panel API

Pricing Model			
Base Plan Price	$0.05 per Hour	Free Plan	$40 per Month
Virtual CPU Cores			1 VCPU
RAM			1740 MB
Disk Space		2 GB	160 GB
Subscription options	Monthly/Fixed	Monthly (otherwise)	Monthly/Fixed

Table 7: Comparison of Google APP Engine, CloudBees and Engine yard on the basis of their Company Profile.

Companies / **Parameters**	**Google App Engine**	**CloudBees**	**Engine Yard**
Programming Model			
Server Os types		Mac OS	Linux
Runtimes	Java Python PHP Go	Java Ruby PHP C Scala Clojure Python Node.js	Ruby Jruby Node.js PHP
Frameworks	Django Grails Pyramid Struts 1 & 2 Flask Spring	Play Ruby On Rails Grails Groovy	Ruby On Rails
Middleware	Django Flask Spring Webapp2	Tomcat Jboss 7	Unicorn HAProxy Passenger Rack Nginx Jetty Tomcat
Native Databases	Datasore NoSQL Cloud SQL	AWS RDS Cloudant EnterpriseDB MongoHQ WebSolr	MySQL PostGreSQL MongoDB
Services			
Native Services	Node.js C++ Scala, Hadoop	Jenkins	Memcached Redis Riak SendGrid

	MongoDB Redis		
Support services available	Forums Phone	Phone	Forums Live Chat Phone
Email Hosting	Yes	No	No
Standards			
Administration and Compliance details		CAMP	SOC 2 HIPAA Compliant

Table 8: Comparison of Google APP Engine, CloudBees and Engine yard on the basis of their Architecture.

Companies / **Parameters**	**Google App Engine**	**CloudBees**	**Engine Yard**
Cloud Features			
Features	Database administration Server configuration Sharing and load balancing Traffic Splitting Multitenancy support	Role base access control High availability	Horizontal Scaling Vertical Scaling Server Performance Database Efficiency
Writable Storage	No	No	
Remote Access		SSH	SSH

Table 9: Comparison of Google APP Engine, CloudBees and Engine yard on the basis of their Working Culture.

Chapter-7

CONCLUSION AND FUTURE SCOPE

Given the wide cloud adoption and the growing number of PaaS providers, Enterprises will certainly face the challenge of finding the appropriate PaaS vendor that can satisfy their functional and non-functional requirements.

In this research work, a framework for evaluation and selection is presented. It relies on the various parameters that affect the performance of a given PaaS for a given set of requirements. As a future work, it is intended to create a selection tool that will provide best results for some real scenarios.

References

[1] P. Mell and T. Grance, "The NIST Definition of Cloud Computing," National Institute of Standards and Technology, Special Publication 800-145, 2011.

[2] S. K. Garg, et al., "Smicloud: A framework for comparing and ranking cloud services," In proc. Of the 2011 Fourth IEEE International Conference on Utility and Cloud Computing (UCC), pp. 210-218, 2011.

[3] F. K. Hussain and O. K. Hussain, "Towards Multi-Criteria Cloud Service Selection," In Proc. Of the 2011 Fifth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), pp. 44-48, 2011.

[4] A. Li, et al., "CloudCmp: comparing public cloud providers," In Proc. of the 10th annual conference on Internet measurement, pp. 1-14, 2010.

[5] Z. ur Rehman, et al., "A Framework for User Feedback Based Cloud Service Monitoring," presented at the 2012 Sixth International Conference on Complex,Intelligent and Software Intensive Systems (CISIS), 2012.

[6] Yashpalsinh Jadeja and Kirit Modi, “Cloud Computing - Concepts, Architecture and Challenges,” In Proc. Of the 2012 International Conference on Computing, Electronics and Electrical Technologies [ICCEET], 2012.

[7] Cloud Security Alliance, 2009, Security Guidance for Critical Areas of Focus in Cloud Computing, http://www.cloudsecurityalliance.org (accessed 9 June 2010)

[8] Muhammad Baqer Mollah, Kazi Reazul Islam, Sikder Sunbeam Islam, “Next Generation of Computing through Cloud Computing Technology,” In Proc. Of the 2012 25th IEEE Canadian Conference on Electrical and Computer Engineering (CCECE) 978-1-4673-1433-6/12/$31.00 ©2012 IEEE.

[9] Byron Ludwig, et al., “A Comparison of PaaS clouds with a detailed reference to security and Geoprocessing services,” In Proc. Of the 1st International workshop on Pervasive Web Mapping, Geoprocessing and Services 2010.[10] Sean Carlin and Kevin Curran, “Cloud Computing Technologies,” International Journal of Cloud Computing and Services Science (IJ-CLOSER) Vol.1, No.2, June 2012, pp. 59~65ISSN: 2089-3337.

[11] Kuyoro S. O., Ibikunle F. & Awodele O., “Cloud Computing Security Issues and Challenges,” International Journal of Computer Networks (IJCN), Volume (3) : Issue (5) : 2011.

[12] Intel IT center White paper, “Platform-as-a-Service,” August, 2013.

[13] K.Bhavya , K.Yamini and V.Sreenivas, “Cloud Services Portability for secure migration,” International Journal of Computer Trends and Technology (IJCTT) - volume4Issue4 –April 2013.

[14] “PaaS Providers List: 2014,” [online]. Available: http://www.tomsitpro.com/articles/paas-providers,1-1517.html

[15] “Force.com,” [online]. Available: http://en.wikipedia.org/wiki/Force.com

[16] “10-steps to avoid Vendor lock-in,” [online]. Available: http://www.zdnet.com/10-steps-to-avoid-cloud-vendor-lock-in-7000017971/

[17] ”Openshift under the hood,” [online]. Avaialble: http://cloudmechanic.blogspot.in/2012/11/openshift-back-end-services-messaging.html

[18] “Azure: Microsoft’s Cloud Platform,” [online]. Available: http://azure.microsoft.com/en-us/

[19] “Openshift Platform as a Service,” [online]. Available: https://www.openshift.com/products

[20] “Salesforce.com,” [online]. Available: http://en.wikipedia.org/wiki/Salesforce.com

[21] “Agile apps Live Pricing,” [online]. Available: http://www.agileappslive.com/shop.htm

[22] “ Amazon AWS Elastic Beanstalk,’ [online]. Available: http://www.agileappslive.com/shop.htm

[23] “ Cloud foundry core,” [online]. Available: core.cloudfoundry.org

[24] “Overview of App Engine features,” [online]. Available: http://www.agileappslive.com/shop.htm

[25] “Engine Yard Stacks and capabilities,” [online]. Available: http://www.agileappslive.com/shop.htm

[26] **“**Gartner Says Worldwide Platform as a Service Revenue Is on Pace to Reach $1.2 Billion,” [online]. Available: http://www.gartner.com/newsroom/id/2242415