"Focus on Your Application Not Plumbing"

[1] *Mrs. Smita Bajpai Asst.Professor* [2], **Mr.Rohit Saxena Asst.Professor**

[1] **CS Dept.RIET Collage (GBTU Code 293) Rama Group Kanpur (U.P.)**
[2] **H.O.D.Asst.Professor CS Dept.RIET Collage (GBTUCode293) Rama Group Kanpur (U.P.)**

:smitalaxmi.bajpai@gmail.com,rohitsaxena9@gmail.com

ABSTRACT: *Cloud Foundry allows developers to focus on applications, not machines or middleware. Traditional application deployments require developers to configure and patch systems, maintain middleware and worry about network topologies. Cloud Foundry allows you to focus on your application, not infrastructure, and deploy and scale applications in seconds. Cloud Foundry is a cloud-era application platform which includes a self-service application execution engine, an automation engine for application deployment and lifecycle management, a scriptable command line interface (CLI), integration with development tools to ease development and deployment processes and an open architecture for quick development framework integration, application services interface and cloud provider interface. Cloud Foundry is ideal for any developer interested in removing the cost and complexity of configuring infrastructure and runtime environments for their applications without locking themselves into a single cloud. Developers can deploy their applications to Cloud Foundry using their existing tools and with zero modification to their code. In this paper I am discussing here, techniques of Cloud Foundry, how Cloud Foundry is good for any developer.*

Keywords— Cloud foundry, developer, cost, security, management, service provider etc

I: INTRODUCTION: NIST* DEFINITION: "… DEPLOY ONTO THE CLOUD INFRASTRUCTURE CONSUMER-CREATED OR ACQUIRED APPLICATIONS CREATED USING

Programming languages, libraries, services, and tools supported by the provider". Consumer responsible only for writing application. Consumer responsible only for writing application. Vendor gives sandboxed environment to develop/deploy applications. Multiple consumers share the platform

*NIST = National Institute of Standards and Technology

1.1: Platform as a Service for developer: : Platform as a Service (PaaS) is a delivery of a computing platform over the web. PaaS enables we create web applications quickly, without the cost and complexity of buying and managing the underlying software/hardware.PaaS provides all the facilities required to support the complete life cycle of building and delivering web applications entirely on the web.

As Platform-as-a-Service (PaaS) is available as a service, the developer and ISV's get full control of the application development and deployment. PaaS enables developers and ISV's to create custom web applications and deliver it quickly, as many of the hassles like setting up hosting, servers, databases, user interaction process and frameworks are prepackaged.

1.2: **Pass Benefits for Business Users:**

Platform as a Service (PaaS) help business users to minimize operational costs and increase their productivity.

- Time to Market
- Requires no up-front investments
- Minimize operational costs
- Centralized information management
- Enhanced productivity
- Access to information anywhere, anytime
- Easy collaboration Secured customized access

1.3: Figure1Platform for developer:

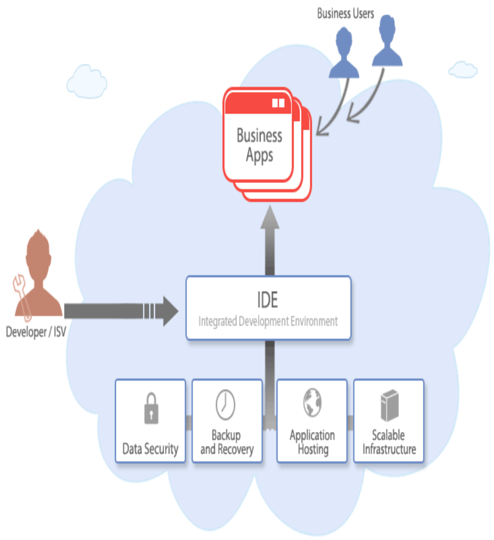


Figure1

1.4: PaaS Benefits for Developers:

Platform as a Service (PaaS) enables developers to focus only on innovation that provide real business value instead of infrastructure set-up.

- Zero Infrastructure
- Lower Risk
- Lower cost and improved profitability
- Easy and quick development
- Monetize quickly
- Reusable code and business logics
- Integration with other web services

II.Types:

1.Add-on development facilities: These facilities allow customization of existing software-as-a-service (SaaS) applications, and in some ways are the equivalent of macro language customization facilities provided with packaged software applications such as Lotus Notes, or Microsoft Word. Often these require PaaS developers and their users to purchase subscriptions to the co-resident SaaS applicationStand alone development environments Stand-alone PaaS environments do not include technical, licensing or financial dependencies on specific SaaS applications or web services, and are intended to provide a generalized development environment.

2. Application delivery-only environments: Delivery-only PaaS offerings do not include development, debugging and test capabilities as part of the service, though they may be supplied offline. The services provided generally focus on security and on-demand scalability.

3. Open platform as a service: This type of PaaS does not include hosting as such; rather it provides open source software to allow a PaaS provider to run applications. For example, AppScale allows a user to deploy some applications written for Google App Engine to their own servers, providing datastore access from a standard SQL or NoSQL database. Some open platforms let the developer use any programming language, any database, any operating system, any server, etc. to deploy their applications.

III: Key Characteristic: : for applying pass services, Allows only provider supported programming languages, tools, APIs and components for building applications.

1. No control of underlying infrastructure.
2. Network, servers, operating systems, or storage.
3. Can only control deployed application and possibly.
4. Its hosting environment configurations.
5. Effort needed to setup/management is lower than IaaS, But at the cost of flexibility.

1. Multi-tenant architecture: PaaS offerings typically attempt to support use of the application by many concurrent users, by providing concurrency management, scalability, fail-over and security. The architecture enables defining the "trust relationship" between users in security, access, distribution of source code, navigation history, user (people and device) profiles, interaction history, and application usage.

2.Integration with web services and databases: Support for **SOAP** and REST interfaces allow PaaS offerings to create compositions of multiple web services, sometimes called "**mashups**" as well as access databases and re-use services maintained inside private networks. Support for keeping the user/relationships (if multiple users)/device context and profile through the mashup across web services, databases and networks.

IV: Full development for PaaS **characteristics:** Services to develop, test, deploy, host and maintain applications in the same integrated development environment**. Support for development team collaboration:** The ability to form and share code with ad-hoc or pre-defined or distributed teams could potentially enhance the productivity of PaaS offerings. In some cases, schedules, objectives, teams, action items, owners of different areas of responsibilities, roles (designers, developers, tester, QC) can be defined, updated and tracked based on access rights.

Utility-grade instrumentation. PaaS offerings provide developers some insight into the inner workings of their applications, and the behavior of their users. Some PaaS offerings use information about user behavior to enable pay-per-use billing. Historical usage and logs may help:

- Determine whether services are of value to users/customers.
- Compare the value of different services.
- Track activity based costs and revenues.
- Visualization tools could show usage patterns, exposing functional or correlational relationships between services and/or user interactions.
- The value to the user or users.
- The cost of alternative service paths such as web, mobile browser or mobile applications.
- Architecture of Pass: Application Hosting Environment: Applications run in a secure sandbox environment. Limited access to the underlying operating system Allows GAE to load balance requests for application across multiple servers, and automatically scale the servers But some restrictions apply Sandbox Restrictions:
- An app can only access other computers on the Internet through the provided URL fetch and email services.
- Other computers can only connect to the application via HTTP(S) requests on standard ports.
- Cannot write to the file system, cannot spawn a sub-process or execute code after the response has been sent.
- Can read only files bundled with application code
- Application code only runs in response to a web request, a queued task, or a scheduled task must return response data within 60 seconds in any case.

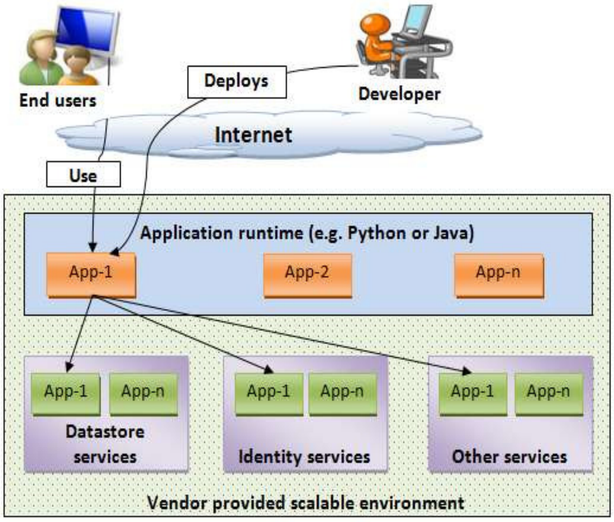
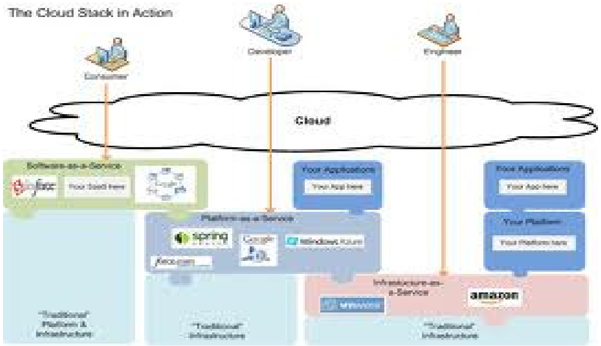


Fig 2. **Vendor Provide Environment for developer.**



Fig,3 .Vendor Provide Platform

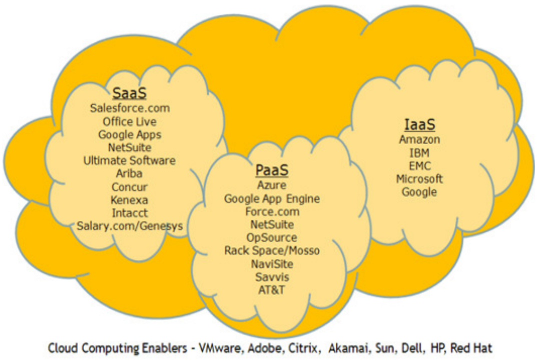


Fig 4. .Example Of Popular services

V: • Popular PaaS:

Several players in the field, Google apps,Engine, Microsoft Azure, VMware, cloud Foundry etc.

VI: CONCLUSION: IT research and consulting firm Frost & Sullivan recently predicted that the platform as a service (PaaS) market will be the next area of keen competition for cloud innovators because the infrastructure as a service (IaaS) and software as a service (SaaS) spaces have been commoditized. As most software available from the cloud is standardized, enterprises are looking to leverage PaaS offerings as it will be the only stack where a service provider can create differentiation. For the record, PaaS is a category of cloud computing services that provides a computing platform and a solution stack as a service for enterprises. Along with SaaS and IaaS, PaaS is a fast-growing service model of cloud computing. In this model, the PaaS customer creates the software using tools and libraries from the provider. The customer also controls software deployment and configuration settings. The provider delivers the networks, servers and storage. PaaS offerings facilitate the deployment of applications without the cost and complexity of buying and managing the underlying hardware and software. Various types of PaaS vendor offerings are extensive and include the application hosting, development, testing and deployment environment, along with extensive integrated services that consist of scalability, maintenance and versioning.PaaS eases some of the tasks for developers Don’t need to worry about underlying infrastructure but also brings in some restrictions. Provider gives commonly needed application services

Language runtimes, storage engines, identity, management, async tasks etc.

PaaS provider takes care of low level issues such as scalability, OS and other S/W patches and updates etc.

REFRENCES

[1"The NIST Definition of Cloud Computing". National Institute of Science and Technology. http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf. Retrieved 24 July 2011.

2.Google angles for business users with 'platform as a service'.

3.Comparing Amazon’s and Google’s Platform-as-a-Service (PaaS) Offerings | Enterprise Web 2.0 | ZDNet.com

4."Cloud Platform as a Service (PaaS) in Cloud Computing Services".CloudComputingSec. 2011.http://cloudcomputingsec.com/296/cloud-platform-as-a-service-paas-in-cloud-computing-services.html.Retrieved 2011-12-15.

5. Erin Kutz, “Platform-as-a-Service Startup Mendix Pockets $13M Series A,” *Economy*, October 2011\

6. Danielson, Krissi (2008-03-26). "Distinguishing Cloud Computing from Utility Computing". Ebizq.net. Retrieved 2010-08-22.

7."Recession Is Good For Cloud Computing – Microsoft Agrees". CloudAve. Retrieved 2010-08-22.