Kartik Padave

70362019039

A022

Practical 3

PaaS

Platform as a Service is a type of cloud computing service mainly used by developers to develop, test and deploy applications. This method does not include the complexity of building and maintaining the infrastructure for development and environment. This makes the development, testing and deployment process relatively easier for the developers.

A PaaS service can be provided in three ways:

a. Via a service provider. In this case the consumer is given access to a public cloud where he/she can control the software and application development with minimum configuration to infrastructure, while the provider provides and maintains the networking, storage, OS and other complicated services.
b. As a private service behind a firewall.
c. As software deployed on public infrastructure as a service.

Advantage of using PaaS:

- Allows high level programming with reduced complexity making development of application more effective and efficient.
- Has built in self-up and down infrastructure resources i.e., the resources required can be easily increased or decreased.
- Maintenance and enhancement of applications and softwares is thus easier.

Disadvantages of PaaS:

- Increased pricing at larger scales.

- Lack of operational features.
- Reduced control.

Types of PaaS:

1. Public PaaS
   Public PaaS is originally derived from Software as a Service (SaaS) and is situated in between SaaS and IaaS (Infrastructure as a Service). The SaaS part provides hosting in the cloud so that no hard drive space is taken in user's computer.

2. Private PaaS
   Private PaaS can either be downloaded and installed in a company's data center. Once installed on all machines private PaaS arranges the application and database components into a single hosting platform.

3. Hybrid PaaS
   Hybrid PaaS has services of both public and private PaaS.

4. Communication PaaS (CPaaS)
   CPaaS is a cloud-based platform that enables developers to add real time communication features like voice, video and messaging in their application without building backend infrastructure from scratch. Applications using CPaaS are very sensitive to latency and packet loss.

5. Mobile PaaS (MPaaS)
   MPaaS provides development capabilities for mobile app developers and designers.

6. Open PaaS
   Open PaaS provides open-source software allowing a PaaS consumer to run applications in an open-source environment, like Google App Engine. Some open platforms let the developer use any programming language, database, operating system or server to deploy their applications.

PaaS Providers:

1. AWS - Amazon Web Services

2. Salesforce

3. Microsoft Azure

4. Bluemix - IBM

5. RedHat

6. Google App Engine

7. Heroku

Use of PaaS by a company that designs and manufactures diesel and natural gas-powered engines:

The company was facing a significant challenge, in need of a common sourcing process that would support their direct purchasing needs around the world throughout their divisions. They needed built-in capabilities in purchasing functions to allow all relevant parties to collaborate and therefore make confident decisions.

This comprehensive solution would need to facilitate their ability to efficiently source a large number of items, given that the company has hundreds and thousands of parts. Just one single sourcing involves hundreds, even thousands, of the company's 700,000 parts, with a high number of attributes and cost factors to consider.

Employees were using multiple systems to manage huge amounts of data, and enterprise leaders realized they needed to create a system whereby team members could rely upon a single source of information, one that delivered what was needed to the right people at the right time.

The overall solution involved Cloud strategic procurement to E-Business including PaaS extensions integrated to Cloud sourcing. More specifically, PaaS extensions focused on:

- high volume uploading
- price breaks
- line level attributes

From a functional standpoint, we needed to assess and create solutions in the following areas:

- creating and editing a supplier negotiation
- creating and editing supplier negotiation lines with:
    - multiple line level attributes
    - multiple price breaks
- uploading mass negotiation lines from Excel
- cloning negotiation lines

From a technical standpoint, it took two months to develop the highly complex solution that included the following elements:

- Java Cloud service SaaS extension provisioned in the same identity domain as the Oracle Procurement Cloud
- Oracle ADF for development of the application UI
- Procurement Cloud web services as a data model

Companies are increasingly needing to import or otherwise mass load items into sourcing events. With direct purchasing companies, there are typically large attribute lists associated with each item, especially when engineering part numbers are involved. Uploading lines in bulk doesn't really help direct purchasing companies if they can't also load price breaks and line level attributes.

There was, however, no option available to create a PaaS solution that solely focuses on the line level, and it isn't practical to drill into each line, one by one, and individually add data. To solve this problem, the entire negotiation needs to be created, and then additional functionality needs to be layered in with multiple levels for cost factors and price breaks.

It can be challenging to address these types of issues without negatively impacting the bells and whistles built into Cloud sourcing functionality. CSS consultants, in this case, needed to focus on fixing gaps rather than attempting to rebuild entire modules in PaaS. RFQs, as just one example, needed to be pushed into Cloud sourcing to take full advantage of the sourcing module.

The company could now effectively maintain global contacts and manage contract development documents, as well as complete risk assessments, audits, and certification tracking for suppliers. CSS was able to transform the company' operations through a hybrid Cloud and EBS solution with PaaS extensions.

PaaS Extensions developed using Java Cloud service and database Cloud service allowed the enterprise to benefit from a supplier negotiation process that was tailored specifically for their needs. It supports unique price break and line level attribute requirements and seamlessly serves as an extension of Oracle Cloud.

Google App Engine (GAE)

Google App Engine is Platform as a Service used for developing and hosting web applications. Applications are sandboxed and run across multiple servers. The engine offers scaling up of web application and its resources with increasing users and demands. Google App Engine has support for Go, PHP, Java, Python, Node.js, .NET and Ruby. Other languages can also be used by configuring the custom runtimes. This service is free with limited resources and in a standard environment. Fees are charged for the additional storage, bandwidth required on top of the free service. Google App Engine requires a Google account to get started and an account may allow the developer to register up to 25 free applications and an unlimited number of paid applications. It defines usage quotas for free applications. Extensions to these quotas can be requested, and application authors can pay for additional resources.

Features of Google App Engine:

1. Diverse Frameworks

   Any Python framework that supports WSGI (Web Server Gateway Interface) using the CGI adapter can be used to create an application. The frameworks can be uploaded with the developed application. Third-party libraries written in pure Python can also be uploaded. Django web framework and applications can be used in the app engine with modifications. Django - nonrel allows Django to work with non-relational databases.

   Google App Engine also has supports for many Java frameworks. With accompanying technologies such as JSP. A newer release of App Engine Standard Java in Beta supports Java8, Servlet 3.1 and Jetty9.

2. Reliable

   The app engine has a 99.95% uptime SLA. App engine is designed is designed and built in a way such that multiple downtimes of datacentres can be handled seamlessly. Even google engineers provide support and guidance for platform to people with premier accounts.

3. Bulk Downloads

The open-source Python projects gaebar, approcket, and gawsh allow users to download and back up App Engine data. Currently there is no method for bulk downloading data from the engine using Java.

4. Pay as you use

The app engine works on a pay-as-you-go model, i.e., you only pay for what you use. The app engine automatically scales up resources when the application traffic picks up and vice-versa.

Difference between GAE and other cloud hosting:

- App Engine provides more infrastructure which makes it easy to develop scalable applications
- Since, Google handles deploying code to a cluster, monitoring, failover, and launching application instances as necessary. This removes many of the system administration and development challenges of building applications to scale to hundreds of requests per second and beyond.
- App engine can run only a limited range of applications designed for the infrastructure.
- Other services lets users install and configure nearly any *NIX compatible softwares but App Engine requires developers to use only its supported languages, APIs, and frameworks. Its Current APIs allows storing and retrieving data from the Google Cloud Datastore database, making HTTP requests, sending e-mail, manipulating images and caching. Google Cloud SQL can be used for applications requiring a relational MySQL compatible database backend.

Benefits of using GAE:

1. All Time Availability
While developing and deploying web applications on the cloud, you get remote access for your applications. Google App Engine is the right choice that lets the developers develop applications remotely, while the cloud service manages the infrastructure needs.

2. Easy to Use

The developers only require to write code. With zero configuration and server management, you eliminate all the burden to manage and deploy the code. Google App Engine makes it easy to use the platform, which offers the flexibility to focus on other concurrent web applications and processes. The best part is that GAE automatically handles the traffic increase through patching, provisioning, and monitoring.

3. Diverse APIs

Google App Engine has several built-in APIs and services that allow developers to build robust and feature-rich apps.
These features include:

- Access to the application log
- Blobstore, serve large data objects
- Google App Engine Cloud Storage
- SSL Support
- Page Speed Services
- Google Cloud Endpoint, for mobile application
- URL Fetch API, User API, Memcache API, Channel API, XXMP API, File API

4. Highly Scalable

Google App Engine cloud development platform is automatically scalable. Whenever the traffic to the web application increases, GAE automatically scales up the resources, and vice-versa.

Use of GAE to build a Micro Blogging Site:

A standard micro blogging site has the following features:

- User registration and secure login
- Client-server model
- Dynamic updates and data persistence
- Events and notifications
- Email communication

The goal is to implement the features listed above using the tools and APIs provided by GAE development platform. This effort will provide an insight on the level of usability, scalability, portability, and flexibility in developing web applications on the GAE's platform. The secondary motivation for developing the application is to design components for a micro blogging framework. The goal is to develop a set of modular components and building blocks for developing micro blogging sites. The use of micro blogs has made into corporate life where the tool helps create communities within the company environment. The messaging and notification functionality of blogging tools help the exchange of information among colleagues at work. The exchange of messages does not need to be between people. Systems can be integrated to the blogging system so that it can notify users of events. For example, a blogging system could be used by a software engineering team to monitor the development efforts. When new code is checked into the Version Control System (VCS), a text message is sent to the software architect or developer about certain events (e.g., build or compilation errors). Depending on the type 15 of business the requirements for a micro blogging application can vary and commercial products may not support them out of the box. Hence, there is a need for a platform for quick development of customized features.

There is no doubt that cloud computing has opened new business opportunities and has started a new Internet revolution. Creative companies can take advantage of the cloud services to deploy applications that before were too expensive to develop or maintain. There is the opportunity of getting rid of the IT department in the company all together resulting in savings and less overheads. For cloud providers, there is a new business opportunity where commoditized computer resources can be sold as pay-as-you go model. From the start of the cycle to the end when the cloud computing platform stabilizes, users will most likely be winners since the capital to experiment on technologies such as GAE is low. GAE has proven to be a well thought-out and designed platform. The use of standard APIs in GAE allows the development of well-designed and attractive web applications. The experience on GAE shows that users can focus on the business logic while developing flexible and portable web applications.