Platform as a Service (PaaS)

Outline

- **PaaS**
  - What is PaaS..?
  - Traditional model v/s PaaS model
  - Advantages & Disadvantages of PaaS
  - PaaS Providers
- **DaaS**
  - What is Relational database..?
  - Problem with relational database
  - Features of non- relational database
  - Cloud database Providers

What is PaaS..?

- PaaS provides a full or partial development platform for which development tool itself will be over the cloud.
- With PaaS, developers can:
  - **build web applications** without installing any tools on their computer.
  - **deploy those applications** without any specialized systems administration skills.

Traditional Model
Vs
PaaS Model

Traditional Model

- Building and running on-premise applications have always been **complex and expensive**.
- Each application requires:
  - Hardware
  - Database
  - Web Servers etc.

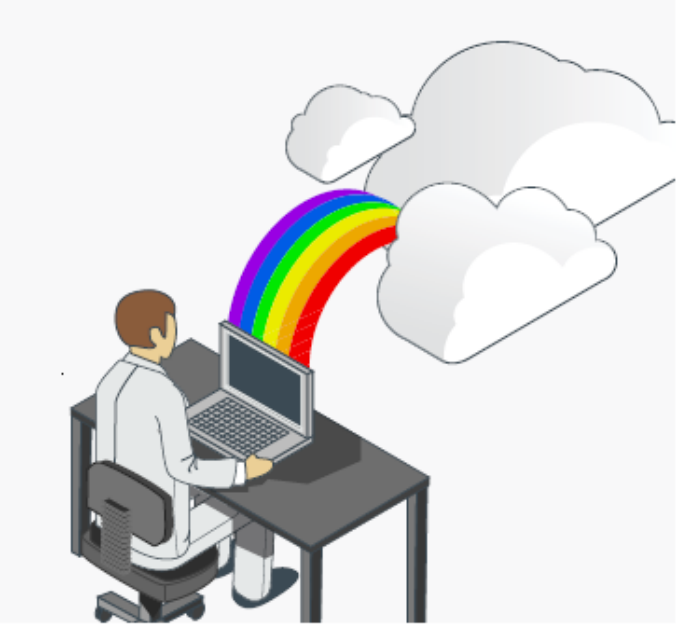
- Once the stack was assembled,
  - A team of developers had to navigate complex programming models like J2EE and .NET.
  - A team of network, database, and system management experts was needed to keep everything up and running.

Problem with Traditional Model

- Any **change in business requirement** will require a change to the application, which again starts lengthy development, test, and redeployment cycle.
- Software and hardware **upgrades** must be managed.
- Enormous amount of **electricity** is also needed to power the servers as well as for system cooling.
- Data must be **replicated** at different data centers so that it can be restored in case of disaster.

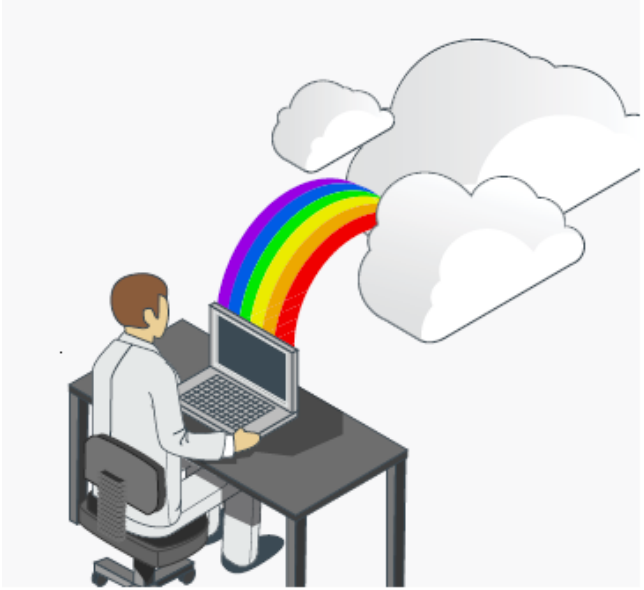
New Model - PaaS

- PaaS offers a faster, more cost-effective model for application **development and delivery**. PaaS provides all the infrastructure needed to run applications over the Internet.
- It is delivered in the same way as a utility like **electricity or water**. Users simply "tap in" and will not worry about the complexity. Also PaaS is based on a metering or subscription model so users only **pay for what they use**.
- With PaaS, corporate IT



PaaS Model

- All the infrastructure to run applications will be over the Internet.
- Developers do **not need to worry** about the storage or hosting.
- Developers write the code and the PaaS provider uploads that code & present it on the internet.
- The PaaS provider manages upgrades, patches, and other routine system maintenance.



Advantages of PaaS

- **Simplified Deployment**

  Developers can focus on development and innovation without worrying about the infrastructure.

- **Prebuilt Business Functionality**

  Some PaaS vendors also provide prebuilt business functionality so that users can avoid building everything from scratch hence helping jump-start projects.

- **Lower Risk**

  No requirements of up-front investment in hardware and software. Developers only need a PC and an Internet connection to start building applications.

- **Instant Community**

  PaaS vendors frequently offer online communities where developers can get ideas, share best practices, and seek advice from others.

- **Pay-per-use model**

  We have to pay only for what we use so there will be reduction in cost.

- **Scalability**

  Applications deployed can scale from one to tens of thousands of users without any changes to the application.

Disadvantages of PaaS

- **Vendor lock-in**

  One have to write applications according to the platform provided by PaaS provider so migrating an application to other PaaS provider would be a problem.

- **Data Privacy**

  Corporate data, whether it can be critical or not, will be private so if it is not located within the walls of the company there can be risk in terms of privacy of data.

- **Integration with the rest of the systems applications**

  It may happen that some applications are local and some are in cloud. So there will be increased complexity when we want to use data which is in cloud with the local data.

What is NOT PaaS…?

- **Browser-based development studio**

  If you have to install something on your computer to develop applications, ***then it's not PaaS..!***

- **Seamless deployment to hosted runtime environment**

  A developer should be able to deploy a PaaS application with one click. If you have to talk to a person to get your app deployed, ***then it's not PaaS..!***

- **Pay as you go billing**

  avoiding upfront costs has made PaaS popular. If you can't pay with your credit card based on usage, ***then it's not PaaS..!***


dotCloud


Windows Azure


AppHarbor


cloudControl
web - application - platform


aS


LONGJUMP


id phpfog


SOFTWARE
force.com
cloud platform


Google
app engine


CloudBees

Google App Engine

- To build application in GAE, you set up an account and get access to the App Engine Software Development Kit (SDK).
- Since this is Google , the platform does not support Microsoft Technologies…!!
- Google App Engine mainly two programming languages:
  1. App Engine's **Java** runtime environment includes standard Java technologies like JVM, Java Servlets, and the Java language.
  2. App Engine's **Python** runtime environment includes a fast Python interpreter and the Python standard library.

Cont…

- **Google Datastore:** App Engine provides a distributed data storage service which features a query engine and transactions.
- **GQL:** App engine also provides query language for querying data.
- **Intended for small applications:** so any request that takes longer than 30 seconds to complete or that sends more than 10MB of data is immediately terminated.

Microsoft Azure

- Azure is Microsoft's equivalent of the GAE so it is slightly more focused on the use of **.NET components.**
- Microsoft Azure mainly provides three components:
  1. **Windows Azure** provides a Windows based environment to develop applications on Microsoft's data centers.
  2. **SQL Azure** provides access to a relational database hosted at Microsoft's' data centers.
  3. **.NET Services** provide access to applications running on the cloud.
- Azure also supports non-Microsoft development languages including **PHP and Python**.

Salesforce.com

- The force.com allows users to access a application development and execution platform from a browser.
- The force.com provides support for language **Apex**. Apex has Java like syntax. But it is not possible to run any Java or .NET programs on the force.com platform.
- According to Salesforce, more than **100,000 applications** have been built on Salesforce platform.

Database as a Service (DaaS)

What is Relational Database..?

- **Tables**: group of tables where each table has columns and rows.
- **Constraints:** tables have constraints, and relationships defined between them.
- **SQL**: query language to query database
- **Joins:** to access multiple tables in a single query
- **Normalization:** data-structuring model which ensures data consistency and removes data duplication.
- **Examples:** Oracle, SQL Server, MySQL, Sybase, DB2 etc.

Problem with Relational Database

- RDBMS have provided database with simplicity, robustness, flexibility, performance and compatibility.
- But today, more and more applications with massive workloads are launched & their scalability requirements can grow very large and can change very quickly.
- When capacity of that single node is reached, we need to scale out and distribute that load across multiple server nodes.

Cont…

- In relational database, if data is across multiple server nodes then it is very hard to maintain **ACID properties** and **Locking & Commitment protocols** for transactions.
- Cloud platform without a scalable data store is not much of a platform at all.
- So for cloud services to be viable, vendors have to provide a scalable place to store data.
- Vendors have to implement a new type of database system that focuses on scalability that is **Non relational database** or **NoSQL database**.

Features of NoSQL database

- Schema free
- Elastic Scalability
- Replication
- Eventual Consistency
- Version History
- De-normalization
- Multi-valued Properties
- …

- **Schema-free**
  - In **relation database**, the schema defines the tables, fields in each table and the relationships between fields and tables.
  - In **non relational database**, no strict schema needs to be defined in advance before using the database.
  - So we can change data which is to be inserted at any time without expensive table rewrites.
  - This provides **greater flexibility** in storing structured data.

Elastic Scalability

- Non relational database can be easily distributed across multiple hosts as load increases.
- During seasonal demand for a product in an e-commerce company or during an exponential growth phase for a social networking Website.
- Ex- Animoto Company

Data Replication

- Keeps multiple copies of the same content in different storage devices and places.
- This ensures high availability of data and support disaster recovery.

Eventual Consistency

- If there is a change in attribute then those changes may not be reflected in read operations that immediately follow.
- All changes will **eventually** propagate through the system and all the replicas will be consistent.

Version History

- Suppose we store value 123 in an attribute, and later change it to 234, data store actually now contains both values, each with a timestamp.
- Ex- Google Docs

Categories of NoSQL database

Category	Description	Name of the database
Document Oriented	Data is stored as documents. An example format may be like - FirstName="Arun", Address="St. Xavier's Road", Spouse= [{Name:"Kiran"}], Children= [{Name:"Rihit", Age:8}]	CouchDB, Jackrabbit, MongoDB, OrientDB,
XML database	Data is stored in XML format	BaseX, eXist, MarkLogic Server etc.
Graph databases	Data is stored as a collection of nodes, where nodes are analogous to objects in a programming language. Nodes are connected using edges.	AllegroGraph, DEX, Neo4j, FlockDB, Sones GraphDB etc.
Key-value store	In Key-value-store category of NoSQL database, an user can store data in schema-less way. A key may be strings, hashes, lists, sets, sorted sets and values are stored against these keys.	Cassandra, Riak, Redis, memcached, BigTable etc.

Cloud Database Providers

Google Datastore

- It is built on **BigTable**, Google's internal storage system for handling structured data.
- Can be used only if we are building applications within the app engine.
- It provides GQL (Google Query language) for querying data.
- Data is stored in form of **entities** & properties where entity can have **multi valued properties.**
- It uses a **strong consistency** model.
- It provides **timestamp versioning.**

GAE Datastore vs. RDBMS

- Joins are not supported
  - Queries with joins are not supported.
  - **Reference property** is used to implement similar thing like foreign keys in RDBMS.
- Uniqueness of data
  - There is no concept of Primary key in data store.
  - Entities are uniquely identified by entity key.
  - Entity key is unique to make entity unique among all the entities of all applications

Microsoft SQL Azure

- SQL Azure is a fully managed cloud database **built on SQL Server** technologies.
- SQL Azure uses Microsoft SQL Server as a backend, but it exposes only a subset of the data types — including string, numeric, date and Boolean.
- SQL Azure provides **T-SQL** query language for querying data.
- It uses an XML-based format for data transfer like Microsoft SQL Server.

Amazon SimpleDB

- SimpleDB is **an attribute-oriented key/value database** available on the Amazon Web Services platform.
- So each item contains attributes & each attribute can have multiple values.
- In SimpleDB, a query can only execute for a maximum of **5 seconds.**
- There are no data types apart from **strings.** Everything is stored, retrieved, and compared as a string.
- The maximum size of any string is limited to **1024 bytes**.
- One key feature of SimpleDB is that it uses an **eventual consistency** model.

Amazon SimpleDB vs. RDBMS

- No schema required, so data modeling is eliminated.
- Supports automatic indexing unlike RDBMS.
- Better querying capabilities
  - Amazon SimpleDB does not store raw data. It takes data as input and creates indices across multiple dimensions to quickly query that data.
- Items are stored in hierarchical structures not in tables.

Thank You..!