BITS Pilani
Hyderabad Campus

BITS Pilani presentation

D. Powar
Lecturer,
BITS-Pilani, Hyderabad Campus

BITS Pilani
Hyderabad Campus

SSZ G527

Cloud Computing

Agenda:

- Introduction to PaaS
- PaaS examples
- Introduction Windows Azure
- Introduction to Drupal and Wolf Frameworks
- Introduction to Force.com PaaS
- 5 Principles of UI Design - AWS PaaS
- Introduction google app engine
  - Google app engine demo

Introduction to PaaS

- The PaaS model provides the tools within an environment needed to create applications that can run in a Software as a Service model
- PaaS is application middleware offered as a service to developers, integrators, and architects.
- Development and Operation teams use PaaS to design, build, and deliver customized applications or information services.
- Instead of relying on standardized SaaS, teams using PaaS have more control over solution architecture, quality of service, user experience, data models, identity, integration, and business logic.

PaaS (contd..)

- In PaaS you are given a toolkit to work with, a virtual machine to run your software on, and it is up to you to design the software and its user-facing interface in a way that is appropriate to your needs.
- So PaaS systems range from full-blown developer platforms like Windows Azure to systems like Drupal, Squarespace, Wolf, and others where the tools are modules that are very well developed and require almost no coding.

PaaS (contd..)

- PaaS solution will ensure the availability of the application despite downtime of the underlying virtual machine by automatically creating a new instance of the application on a new virtual machine when the machine goes down.
- PaaS systems can be used to host a variety of cloud services
  - Online portal-based applications like Facebook that need to scale to thousands of users
  - Startup who wants to host their new application in a Software-as-a-Service model
  - Can also be used for massively parallel computations
  - Enterprises can deploy their Line-of-Business applications in the cloud, taking advantage of the scale and availability while still maintaining security and privacy of data

PaaS examples

- Windows Azure
- Google App Engine
- Hadoop platform

Drupal
Wolf Frameworks
Force.com
VMWare Cloud Foundry
Bluemix (IBM)
Eccentex
AppBase
LongJump
SquareSpace
WaveMaker etc

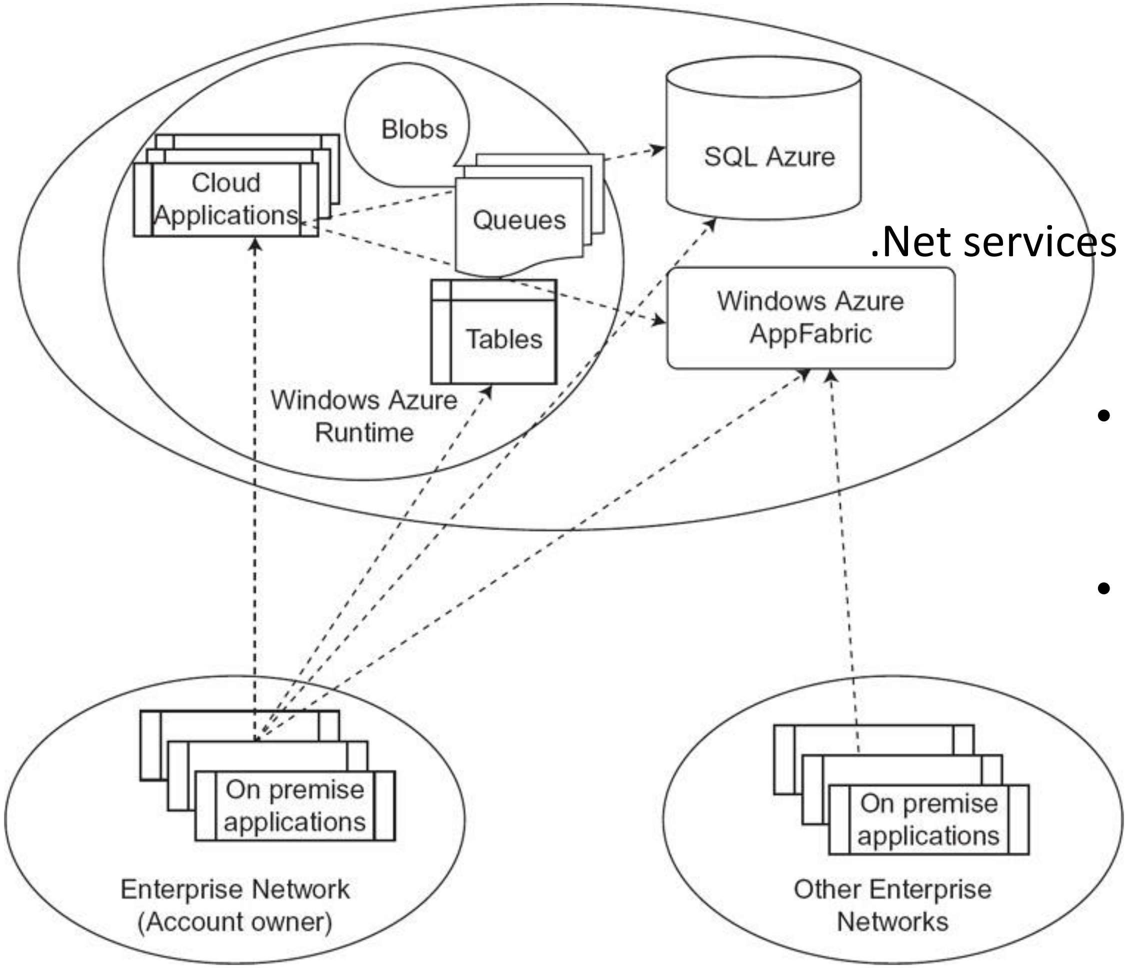
PaaS examples

force.com
platform as a service
amazon web services
S3 & EC2
facebook
Google
MySpace Developer Platform
A Place For Developers
Rollbase beta
Roll your own business apps
LONGJUMP
Ning
Drupal
dabble db
Intuit QuickBase
Coghead
WOLF
eccentex
Knowledge Worker Apps
WaveMaker
CLOUD FOUNDRY
No Obstacles: Deploy and Scale Your Applications in Seconds
IBM
hadoop
Windows Azure

Windows Azure

- The **Azure Services Platform** is a popular application platform for the cloud that allows Windows applications and web-services to be hosted and run in Microsoft datacenters
- A simplistic view - a cloud deployment platform for applications developed for Windows using .NET
- While Azure is primarily designed for PaaS capabilities, it also includes certain features for **Data-as-a-Service** (**DaaS**) and Infrastructure-as-a-Service (IaaS)

Windows Azure (Contd..)



- Windows Azure can be thought of as a Cloud Operating System over Microsoft blade servers
- SQL Azure, AppFabric allow that on-premise applications to interoperate with applications hosted in the cloud with secure connectivity, messaging, and identity management.

Schematic diagram of Azure platform services

Windows Azure (contd..)

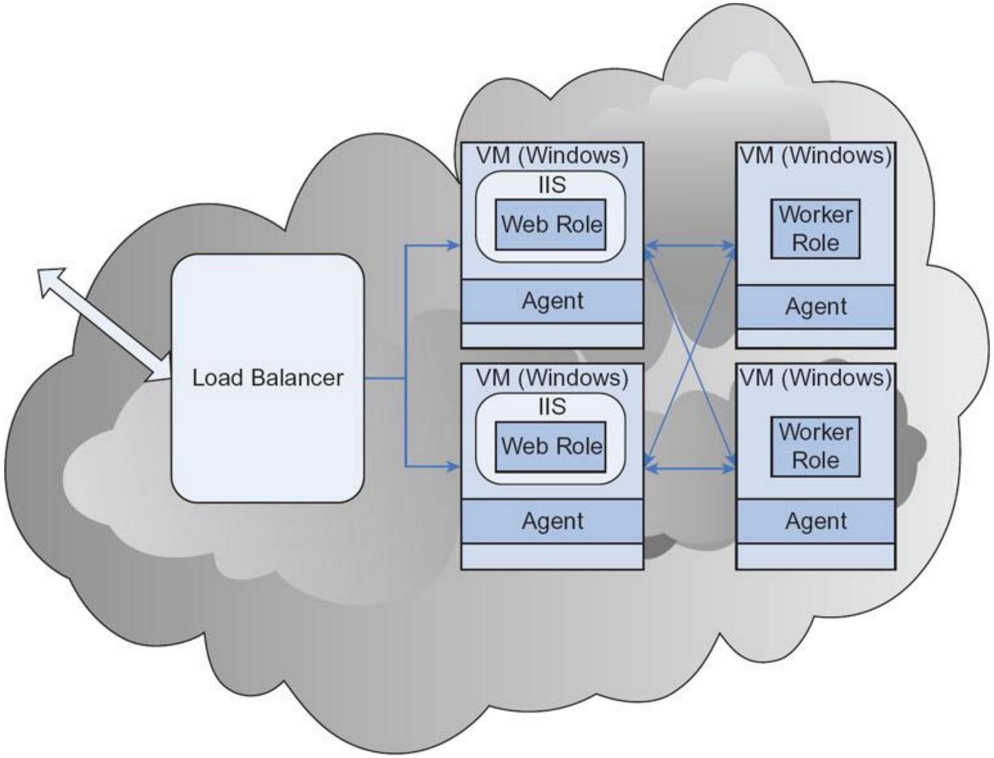
- Like any other cloud-hosted application, Windows Azure applications typically have multiple instances running on different virtual machines.
- Developers need not create nor manage these virtual machines explicitly
- Developers just write applications either as a **Web role** and/or **Worker role**, and tell Windows Azure the number of instances of each role that should be created

Web role Vs Worker role

- Applications are commonly composed of both Web and Worker roles
- You can think of Compute as being a container for web roles and worker roles
- Web roles enable web applications to be hosted on IIS server and provide external interactivity
- You can quickly and easily deploy web applications to Web Roles and then scale your Compute capabilities up or down to meet demand
- Worker roles let you host any type of application, including Apache Tomcat and Java Virtual Machines (JVM)
- A common implementation in Windows Azure takes input from a Web role, sends those requests through a Queue to a Worker role, then processes the requests and stores the output

Compute service with two Web roles and two Worker roles

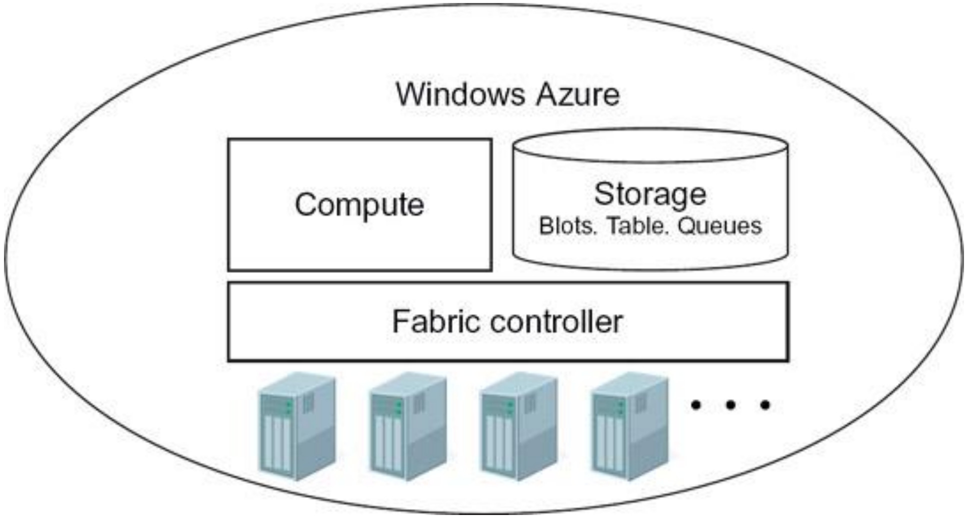
- Worker roles are used for the more heavy-duty processing that is typically done by Windows services in on-premise applications.
- Web roles and Worker roles communicate by either using message queues or by setting up direct connections via WCF or other technologies



- Microsoft provides Visual Studio project templates for creating Windows Azure Web roles, Worker roles, and combinations of the two, and developers are free to use any programming language supported on Windows. Eclipse for Java and PHP development are supported via plug-ins

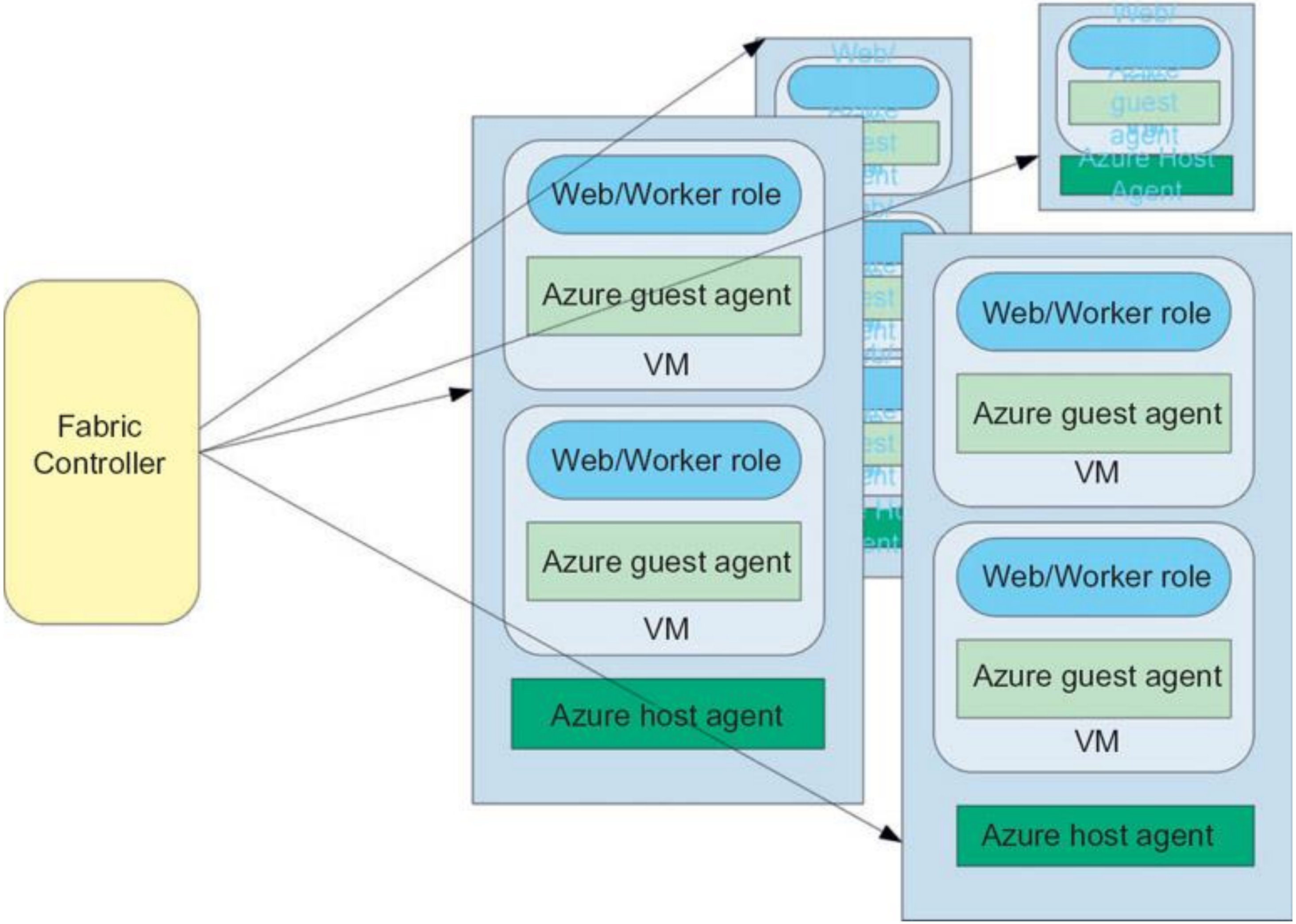
Windows Azure (Contd..)

- Windows Azure runs on a large number of machines, all maintained in Microsoft data centers.
- The hosting environment of Azure is called the **Fabric Controller**
- **FC** is a distributed program that manages the hardware and applications in a cluster internally used by Azure
- The key task of **FC** is to assign the appropriate resources to an application



Windows Azure runtime environment components

Fabric Controller architecture



Azure Cloud Storage Services

Blob service: For large binary and text data

Azure Drives: To use as mounted file systems

Table Service: For structured storage of non-relational data

Queue Service: For message passing between components

Blob service

- Blobs (Binary Large Objects)
- It provides file-level storage and is similar to Amazon S3
- Applications deal with blobs as a whole, although they might read/write parts of a blob
- Blobs are always stored under **containers**, which are similar to AWS buckets
- Every storage account must have at least one container, and containers can have blobs within them.
- Container names can contain the directory separator character ("/") – this gives developers the facility to create hierarchical "file-systems" similar to those on disks
- The blob service defines 2 kinds of blobs to store text and binary data: A **page blob** and a **block blob**.

- Windows **Azure Drives** are used for mounting an NTFS volume to be accessed by an application, and are similar to Amazon EBS.

Table service:

- Structured forms of storage using key-value pairs
- Uses a NoSQL model based on key-value pairs for querying structured data that is not in a typical database
- These tables are not relational in nature, nor are table schemas enforced by the Azure framework

Std_name	ID	DOB
Aryan	2015A7101	12.10.1985
Bhuvan	2015A7105	12/01/1980
Suman	123	April, 12 1972

Table service (contd..)

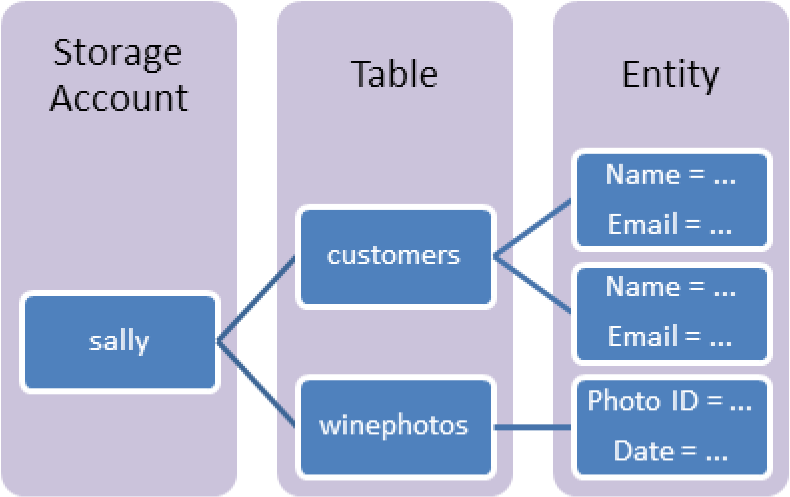
- Data stored in Azure tables is partitioned horizontally and distributed across storage nodes for optimized access.
- Every table has a property called the **Partition Key**, which defines how data in the table is partitioned across storage nodes – rows that have the same partition key are stored in a partition.
- In addition, tables can also define **Row Keys** which are unique within a partition and optimize access to a row within a partition.
- The pair {partition key, row key} uniquely identifies a row in a table

Table service (contd..)

Table: A table is a collection of entities. Single table can contain entities that have different sets of properties. The number of tables that a storage account can contain is limited only by the storage account capacity limit.

Entity: An entity is a set of properties, similar to a database row.

Properties: A property is a name-value pair. Each entity can include up to 255 properties to store data.



Queue Service

- Provide reliable message delivery within and between services
- A storage account can have unlimited number of queues, and each queue can store an unlimited number of messages
- Queues are used by Web roles and Worker roles for inter-application communication, and by applications to communicate with each other

Wolf Frameworks

WOLF Platform-as-a-Service

WOLF Platform-as-a-Service

- Founded in 2006, in Bangalore.
- World’s first 100% AJAX, XML & .NET based Business Solutions -platform available as a service.
- WOLF based solutions are integrated currently with Quickbooks, Microsoft Office Suite, SalesForce, SAP– have more than 1300 plus users using WOLF Platform

Offering Wolf Platform as a Service?

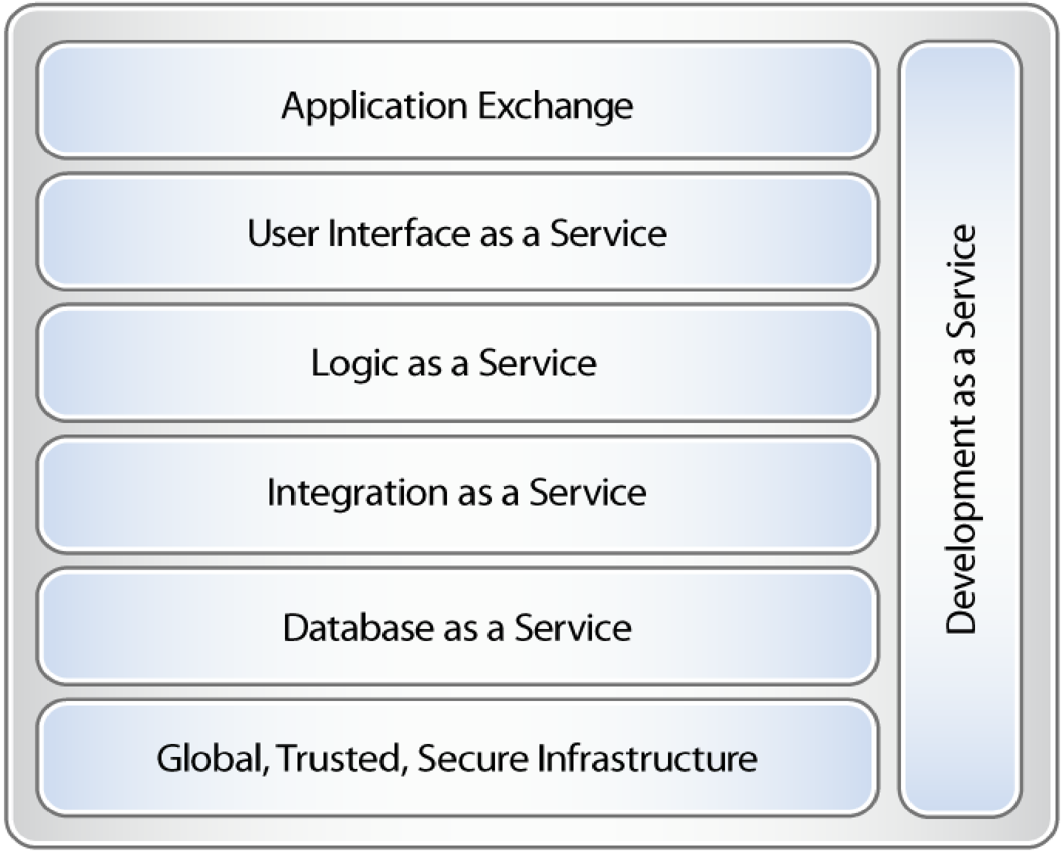
- Wolf Platform as a Service (PaaS) enable users to design & deploy web-based applications without the cost and complexity of buying, managing servers, writing technical code, setting database, middleware and other system software. It supports end-to-end life cycle of building and delivering web applications and services entirely available from the Internet and is an example of the everything as a service trend.
- The Wolf Platform as a Service model enables and manages the end-to-end life cycle of
  - Design
  - Development
  - Integration
  - Testing
  - Deployment
  - Storage
  - Back up
  - Error handling
  - Hosting of a SaaS Web application on the Internet.

force.com™
platform as a service

- PaaS lets developers focus on what their apps do, rather than the software and infrastructure to run them

Force.com			You
Infrastructure Services	**Application Services**	**Operations Services**	**Build Your Apps!**
✓	✓	✓	

Definition of Platform as a Service

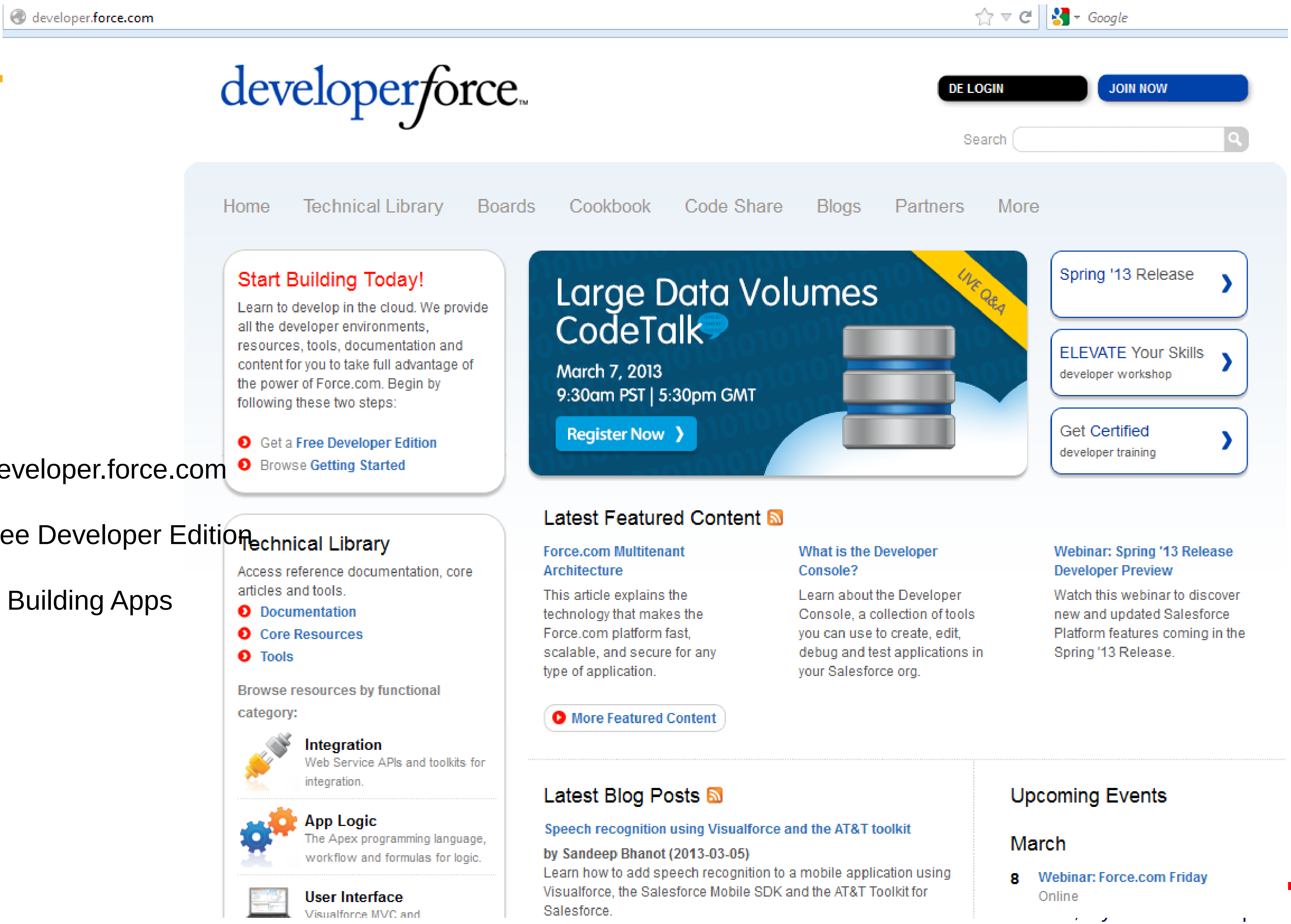


Getting Started Is Easy!

1.Visit developer.force.com

2.Get Free Developer Edition

3. Begin Building Apps



developer.force.com

developerforce

Start Building Today!

Learn to develop in the cloud. We provide all the developer environments, resources, tools, documentation and content for you to take full advantage of the power of Force.com. Begin by following these two steps:

- Get a Free Developer Edition
- Browse Getting Started

Large Data Volumes CodeTalk

LIVE Q&A

March 7, 2013

9:30am PST | 5:30pm GMT

Register Now ❯

Spring '13 Release ❯

ELEVATE Your Skills ❯
developer workshop

Get Certified ❯
developer training

Technical Library

Access reference documentation, core articles and tools.

- Documentation
- Core Resources
- Tools

Browse resources by functional category:

Integration
Web Service APIs and toolkits for integration.

App Logic
The Apex programming language, workflow and formulas for logic.

User Interface
Visualforce MVC and

Latest Featured Content

Force.com Multitenant Architecture
This article explains the technology that makes the Force.com platform fast, scalable, and secure for any type of application.

What is the Developer Console?
Learn about the Developer Console, a collection of tools you can use to create, edit, debug and test applications in your Salesforce org.

Webinar: Spring '13 Release Developer Preview
Watch this webinar to discover new and updated Salesforce Platform features coming in the Spring '13 Release.

More Featured Content

Latest Blog Posts

Speech recognition using Visualforce and the AT&T toolkit

by Sandeep Bhanot (2013-03-05)

Learn how to add speech recognition to a mobile application using Visualforce, the Salesforce Mobile SDK and the AT&T Toolkit for Salesforce.

Upcoming Events

March

8 Webinar: Force.com Friday
Online

Drupal

- Powerful ready-to-go platform to produce web services and applications
- Open source - Free to use and free to modify
- Open API’s provides easy expandability
- Tested and safe. Regular security updates
- Lots of prepared core functions to ease simple coding tasks
- Lots of ready-made modules and code snippets available
- Most basic CMS features are included
- Highly Customizable
  - Plenty of Modules
  - Good Architecture and API
  - Good Documentation
- Performance
- Good Community

Drupal (contd..)

Use:

- ✓ Community Web Sites
- ✓ Weblogs
- ✓ Forums
- ✓ Social Network Sites
- ✓ Wiki/Knowledge Base
- ✓ Business Web Applications

System requirements:

- PHP
- Apache/IIS
  - AppServ is recommended for Windows
  - Lighttpd works as well
- MySQL/PostgreSQL
  - Oracle support
  - IBM is working on DB2 support

Installation (Drupal)

- Download and install xampp server
- Start Apache & MySQL services using xampp control panel
- Go to http://localhost/xampp/ to create database for CM
- Download and import core drupal into xampp’s /htdocs folder
- Run drupal's browser based installer http://localhost/drupal
- Make few configurations
- Your development platform is ready:

  http://localhost/drupal/

5 Principles of Good UI by AWS

MADPO Principles:

- Make it simple
- Automate
- Don’t Please Everyone
- Provide Progressive Help
- Offer translation

MADPO Principles

- M: Page layout editor
- A: Validation rules, Filter lookups, default values
- D: Record types
- P: Custom field help
- M: Translation tools

Introduction google app engine

- Google App Engine is a PaaS solution that enables users to host their own applications on the Google data centers similar to Google Docs, Google Maps and other popular Google services.
- It enables users to develop and host applications written using Java, Python, Go, JRuby, JavaScript (Rhino), Scala etc.
- The applications hosted on Google App Engine can scale both in compute and storage just like other Google products

Google app engine

Developing and Deploying app on Google App Engine

1. Download and install Java EE
2. Add plug-ins: Google plugin for eclipse(SDK)
3. Create a new "Web Application Project"
4. Configure the application
5. Develop code
6. Test in simulated App Engine environment
7. Deploy to Google App Engine

Google app engine demo

Summary

- Introduction to PaaS
- PaaS examples
- Introduction Windows Azure
- Introduction to Drupal and Wolf Frameworks
- Introduction to Force.com PaaS
- 5 Principles of UI Design - AWS PaaS
- Introduction google app engine
    - Google app engine demo