



# Developing Java Based Web Applications in Google App Engine

Tahir Akram  
19<sup>th</sup> December 2009

3<sup>rd</sup> International Conference on Open Source Systems and Technologies  
at University of Engineering and Technology, Lahore

# Today's Goal!!!

To understand the Java application hosting solution on cloud by Google. And to discover its features and to see how it could be useful to develop apps.

# What we will learn?

- Understanding of Google App Engine and cloud computing
- What benefits we can grab and what are the limitations
- From where to get GAE SDK and Eclipse plugin
- Creating our first project and running it on production
- The App Engine sandbox
- Incorporating Google accounts for authentication
- Saving, updating and deleting record in App Engine datastore using JPA
- Sending emails from your application
- Billing and free quotas

# Intended Audience

- FYP Students
- Developers
- Software Configuration Management Engineers
- Web Hosting Geeks

Lets know our  
audience?



# Areas to Discuss

- Hosting Java web apps traditionally
- Introduction to Google App Engine
- Overview of Runtime Environment
- Scalable Services
- Advantages and Limitations
- Billing and Free Quotas
- Demo

# Hosting Java web apps traditionally



# Hosting Java web apps traditionally

- Not so popular except enterprise
- High rates as compared to PHP hosting
- Shared Tomcat instance among users
- Restrictions on any time deployments due to shared server
- Dedicated hosts works fine but they are costly

# You end up with all this



No Worries!!!  
Let Google App Engine  
come into an action



# What is Google App Engine

- It is a platform for hosting web applications in Google-managed data centers. It is **cloud** computing technology which virtualizes applications across multiple servers and data centers.

*-Wikipedia*

What is  
cloud computing?



# Cloud Computing

- In which dynamically scalable resources are provided as a service over the Internet
- Users need not have knowledge of expertise in or control over the technology infrastructure in the cloud

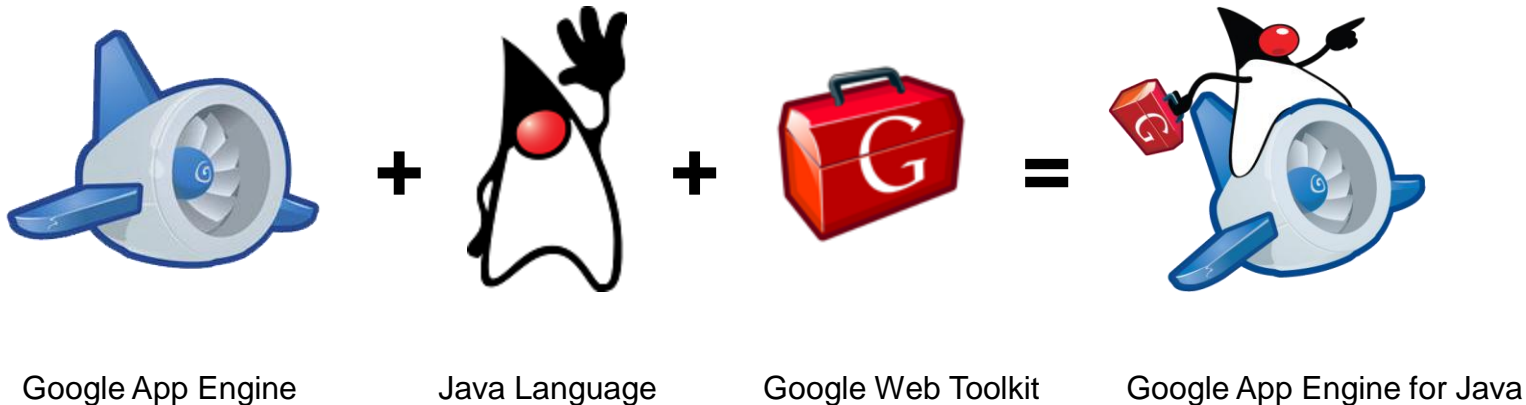
*-Wikipedia*

# Introduction to Google App Engine contd....



# Introduction to Google App Engine

- Was released on April 08 with Python support. Java included on August 09



# Google App Engine

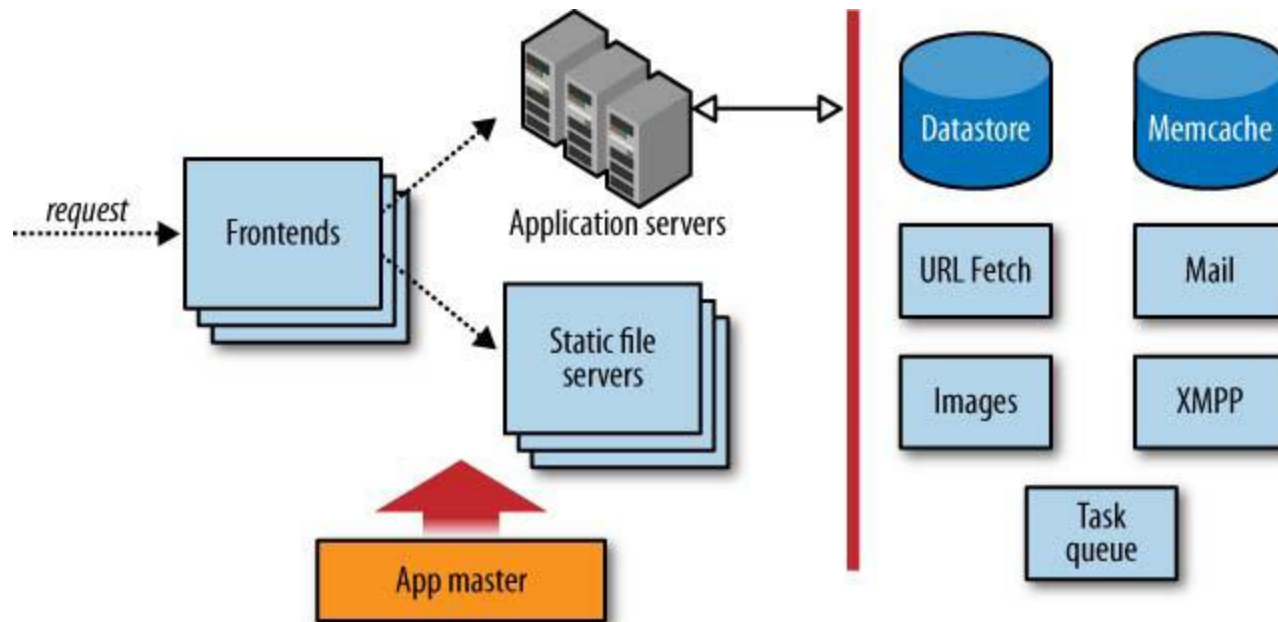
- Easy to use, scale and manage
- Run your application on Google's infrastructure
- Forget worries of managing your servers
- Think about developing more features for your web, let Google manage the rest
- No server restart, no network issues



## Google Datacenters at Dallas, Oregon

Picture credit: Portland Java User Group 18 Aug 2009

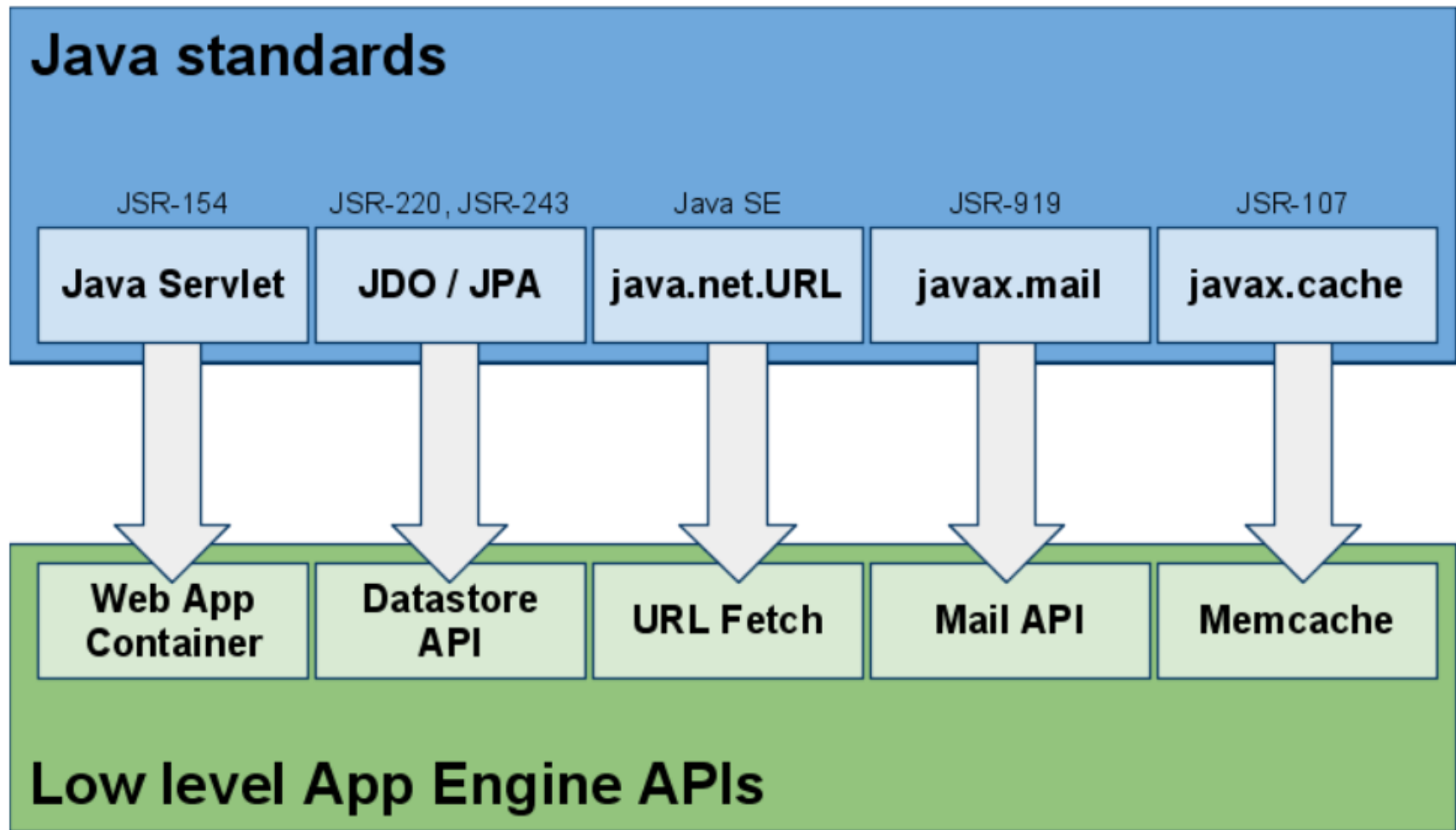
# GAE Architecture



# GAE Java Runtime Environment

- Java 6 VM
- Servlet 2.5 Container
- HTTP Session support (need to enable explicitly)
- JDO/JPA for Datastore API
- JSR 107 for Memcache API
- `javax.mail` for Mail API
- `javax.net.URLConnection` for URLFetch API

# Java Standards on GAE



# Services by App Engine

- **Memcache API** – high performance in-memory key-value cache
- **Datastore** – database storage and operations
- **URLFetch** – invoking external URLs
- **Mail** – sending mail from your application
- **Task Queues** – for invoking background processes
- **Images** – for image manipulation
- **Cron Jobs** – scheduled tasks on defined time
- **User Accounts** – using Google accounts for authentication

<http://code.google.com/appengine/docs/java/apis.html>

# Memcache API

- Better than Datastore
- Storage on memory rather on disk
- Key-value pair mapping
- It implements JCache interface

```
import static java.util.Collections.emptyMap;  
import javax.cache.*;
```

```
CacheFactory cacheFactory =  
    CacheManager.getInstance().getCacheFactory();
```

```
Cache cache = cacheFactory.createCache(emptyMap());
```

```
cache.put(key, value);  
cache.get(key);
```

<http://code.google.com/appengine/docs/java/memcache/>

# Datastore API

- Storing data and manipulation
- Based on Bigtable
- Not a relational database
- GQL (Google Query Language)
- Need to use JDO/JPA

<http://code.google.com/appengine/docs/java/datastore/>

# URLFetch API

- Invoking external URLs from your application over HTTP and HTTPS
- We are already using it in our other Java apps, here it got some more capabilities

```
import java.net.*;
import java.io.*;
```

```
URL url = new URL("http://...");
InputStream inputStream = new
    InputStreamReader(url.openStream());
BufferedReader reader = new BufferedReader(inputStream);
String line;
while ((line = reader.readLine()) != null) {
    //do something
}
reader.close();
```

<http://code.google.com/appengine/docs/java/urlfetch/>

# Mail API

- Send emails on the behalf of app administrator to the Google account user
- You can not receive emails

```
import javax.mail.*;

Session session = Session.getDefaultInstance(new
    Properties(), null);
InternetAddress admins = new InternetAddress("admins");
Message msg = new MimeMessage(session);

msg.setFrom(admins);
msg.addRecipient(Message.RecipientType.TO, admins);
msg.setSubject("subject");
msg.setText("text");

Transport.send(msg);
```

# Task Queues API

- **Perform background processes by inserting tasks into queues.**
- **Instructions need to be mention in file queue.xml, in the WEB-INF/ dir**

```
import com.google.appengine.api.labs.taskqueue.Queue;
import com.google.appengine.api.labs.taskqueue.QueueFactory;
import com.google.appengine.api.labs.taskqueue.TaskOptions;

// ...
TaskOptions taskOptions =
TaskOptions.Builder.url("/send_invitation_task")
.param("address", "juliet@example.com")
.param("firstname", "Juliet");
Queue queue = QueueFactory.getDefaultQueue();
queue.add(taskOptions);
```

<http://code.google.com/appengine/docs/java/config/queue.html>

# Images API

- Manipulation of images
- Transformation of images
- Changing image formats

<http://code.google.com/appengine/docs/java/images/>

# Cron Jobs

- Up to 20 scheduled tasks per app
- Cron jobs (scheduled tasks) supported in cron.xml in WEB-INF dir
- Schedule instructions contain Englis-like format

```
<?xml version="1.0" encoding="UTF-8"?>
<cronentries>
<cron>
<url>/listbooks</url>
<description>Repopulate the cache every day at
5am</description>
<schedule>every day 05:00</schedule>
</cron>
</cronentries>
```

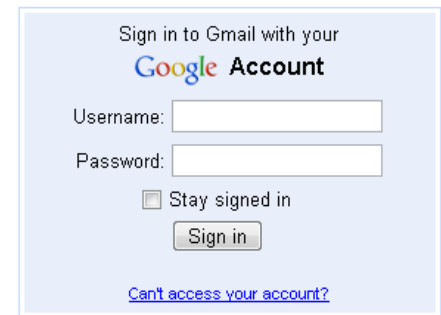
<http://code.google.com/appengine/docs/java/config/cron.html>

# Users API

- Using Google accounts authentication in your application

```
import com.google.appengine.api.users.*;
```

```
UserService userService =  
    UserServiceFactory.getUserService();  
User user = userService.getCurrentUser();  
String navBar;  
if (user == null) {  
navBar = "<p>Welcome! <a href=\"\" +  
    userService.createLoginURL(\"/\") + \"\">Sign in or  
    register</a> to customize.</p>";  
} else {  
    navBar = "<p>Welcome, " + user.getEmail() + "! You can  
    <a href=\"\" + userService.createLogoutURL(\"/\") + \"\">sign  
    out</a>.</p>";  
}
```



Sign in to Gmail with your  
**Google Account**

Username:

Password:

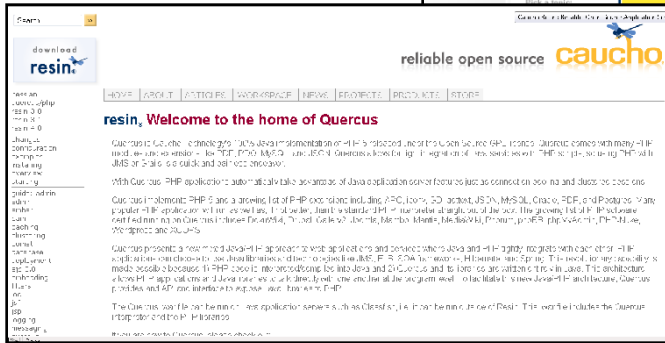
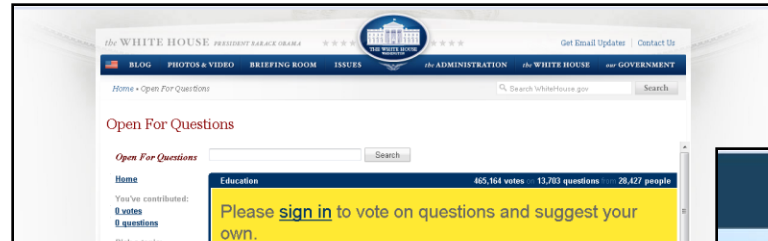
Stay signed in

[Can't access your account?](#)

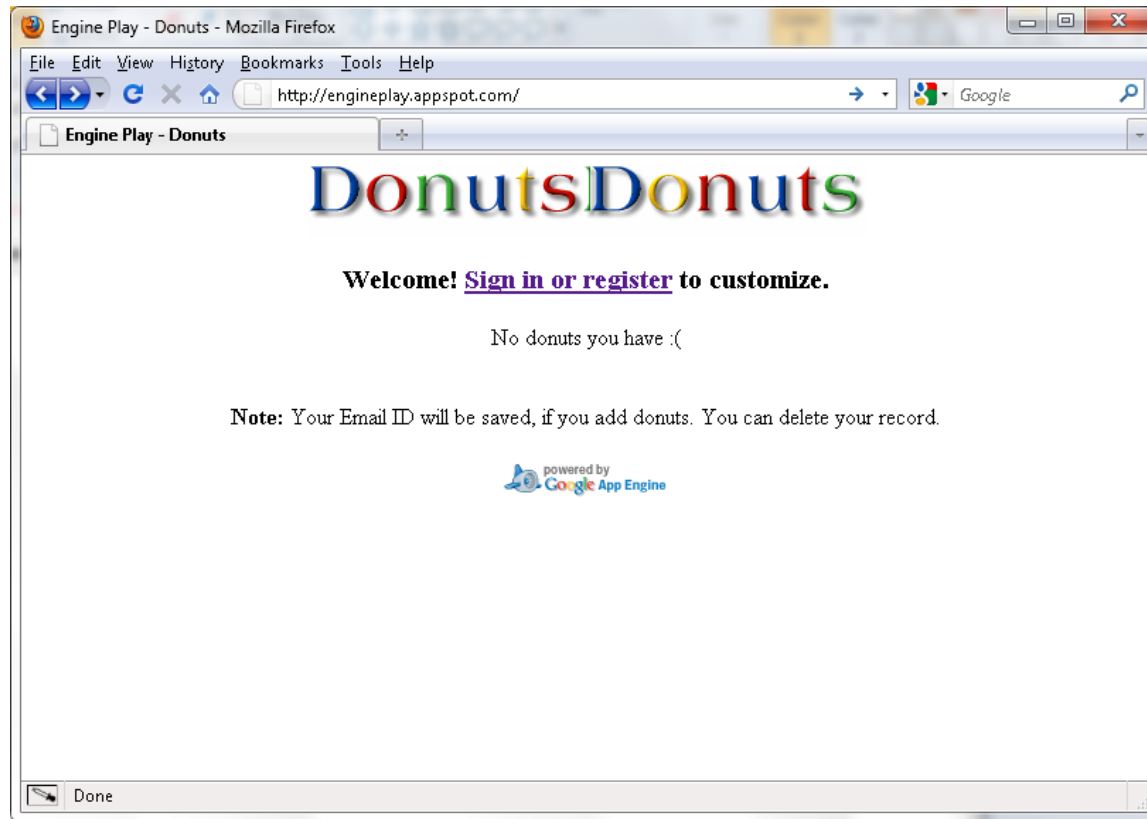
# Limitations

- No HTTPS for custom domains. Only for your-app-id.appspot.com domains
- No streaming and long term connections
- A web request must respond in 30 seconds otherwise GAE throw DeadlineExceededException
- Application will not work with your naked custom domain. i.e <http://yourdomain.com>
- Only HTTP and HTTPS. Client can not connect to GAE through FTP
- First request will be slow
- Application can not connect to socket
- You can not create threads
- Only 1000 files per application

# Who is using GAE?



# Demo



<http://engineplay.appspot.com/>

# Questions, Suggestions and Feedback



# Contact Me

Tahir Akram

Email: tahirakramch [AT] gmail [DOT] com

Read my blog: <http://pakzilla.com>

Follow me on Twitter: <http://twitter.com/tahirakram>

# Resources

Following resources are used to prepare these slides. You must have a look for more insights of Google App Engine

- <http://googleappengine.blogspot.com/2009/04/seriously-this-time-new-language-on-app.html>
- <http://googleappengine.blogspot.com/2009/02/back-to-future-for-data-storage.html>
- <http://code.google.com/p/googleappengine/issues/list>
- [http://code.google.com/appengine/kb/adminconsole.html#delete\\_app](http://code.google.com/appengine/kb/adminconsole.html#delete_app)
- <http://code.google.com/p/googleappengine/issues/detail?id=335>
- <http://code.google.com/eclipse/docs/download.html>
- <http://cjedaudio.wordpress.com/2009/06/19/google-appengine-limitations-workarounds/>
- <http://blog.frankel.ch/tech/dev/java/jee/google-appengine-limitations>
- <http://groups.google.com/group/google-appengine-java/web/will-it-play-in-app-engine>
- <http://java.sys-con.com/node/1199833>
- [http://www.youtube.com/view\\_playlist?p=DA31F43DE4107B05](http://www.youtube.com/view_playlist?p=DA31F43DE4107B05)
- <http://gaejexperiments.wordpress.com/2009/10/29/episode-5-upgrading-to-google-app-engine-1-2-6/>
- <http://code.google.com/appengine/docs/quotas.html>
- <http://code.google.com/appengine/docs/java/config/appconfig.html>
- <http://www.techcrunch.com/2009/03/24/white-house-using-google-moderator-for-town-hall-meeting/>