Lecture 12 – Cloud Deployment

Web Application Development

February 23, 2017

Jeffrey L. Eppinger
Professor of the Practice
School of Computer Science

Lecture Schedule – 1st Half

(subject to change)

#1	Intro	#9	Files & Images
#2	HTML & CSS	#10	AJAX
#3	JavaScript & DOM	#11	jQuery
#4	HTTP & Django	#12	Cloud Deployment
#5	Cookies & Sessions	#13	S3 & Databases
#6	Models	3/2	TBD
#7	Transactions	3/7	Project Proposals
#8	Forms & Templates	3/9	Project Proposals

Agenda

→ Course Administration

Cloud Deployment

Administrative Issues

- HW#4 Grades
  - Most have been posted
  - The TAs had problems grading a few of these
- HW#6 Spec Posted
  - Due on Monday, 2/27
- HW#7 Spec will be posted next week
  - Due Monday, 3/6
  - It's the last homework!
- Project Proposal Spec Posted Now
  - Due Thursday, 3/9, right before break
  - Just one page long (or longer)
  - Ungraded, but we will give you feedback
  - We will give you classtime on Tues, 3/7 and Thurs, 3/9 to finish it

HW#6

Enhance your social network app:

- AJAX Stream Refresh
- Add comments to posts
  - Comments must be added using AJAX

Any questions on HW#6?

Agenda

- ✓ Course Administration
- → The Cloud

The Evolution of the Web

1989 – HTML Specified
- A subset of the SGML, derived from 1960s GML (IBM)

1990 – HTTP Protocol
- First version had one method (GET)

1993 – Common Gateway Interface (CGI)
- Allowed programs to dynamically generate HTML

1994 – Cookies

1995 – SSL

1996 – JavaScript, Java, Flash in the Browser

1999 – AJAX

2008 – Start of the standardization of HTML5

Many Web Browsers

1992 – Mosaic

1994 – Netscape and Opera

1995 – Internet Explorer

2002 – Safari

2004 – Firefox

2008 – Chrome

Centralized vs Personal Computing

1960s – Mainframe Computers

- John McCarthy predicts
  - "computation … organized as a public utility"

1980s – Personal Computing

- Xerox Alto Computer (1973)

2000s – Cloud Computing

- Starting with Common Gateway Protocol (1993)

Cloud Computing

1960s – Transaction Processing Systems

1990s – Web Servers

- Common Gateway Protocols
  - Applications were scripts
- Web Application Servers
  - Initially C programs
  - Then Java
  - Now many choices: Ruby, Python, JavaScript, …

Software as a Service

- Utilizing servers in data centers for complete application function
- Interface on “client-side” is a web browser
- Notable SaaS applications
  - Salesforce.com (1999)
  - Google Apps (2006)
    - Gmail, Groups, Calendar, Docs, Sites

Amazon Web Services (AWS)

- Amazon.com is largest online retail store
- Eventually their infrastructure was overwhelmed
- Led to creation of Amazon Web Services
  - Selling of Amazon's infrastructure to others (2006)
- Many services, including
  - Elastic Compute Cloud (EC2) – Virtual private servers
  - Relational Database Service (RDS)
  - Simple Storage Service (S3)
  - Elastic Beanstalk – Deploy applications
    - Java Servlets, PHP, Python, Ruby, .NET
- Free year for new users

Heroku

- Started development in 2007
- Bought by SalesForce.com in 2010
- Initially allowed app deployment for Ruby
- Now allows Java, Python, etc
- Uses Postgres DB
- Runs on AWS Servers
- Scalable
  - Number of server instances grows as needed
- Trial usage is free, with capacity limit

Google AppEngine

- Google enters the market (2008)
- Deploy apps using Java Servlet, Python, PHP, Go
- Utilize relational or non-relational DB
- Use Gmail authentication or Oauth
- Scalable
  - Number of server instances grows as needed
- Trial usage is free, with capacity limit

Organizing these Concepts

- SaaS – Software as a Service
  - E.g., Salesforce.com, Google Apps
- PaaS – Platform as a Service
  - E.g, Heroku, Amazon's Elastic Beanstalk, Google AppEngine
  - Free trials available
- IaaS – Infrastructure as a Service
  - E.g., Amazon EC2, Microsoft Azure, Google Compute Engine
  - Free trials available

IaaS Example

- Amazon's EC2
  - I have some EC2 virtual private servers (can be less than $2/mo)
    - Currently, you can get 1 yr free with a new account
  - See http://aws.amazon.com for info
  - Runs Ubuntu Linux, or Red Hat, or Windows, etc
  - I can configure it – I have to configure it
    - Python, pip, Django
    - Apache HTTP Server, wsgi_mod
    - MySQL
    - DNS name (e.g., webapp.jeffeppinger.com)
  - Strong Security
    - Use SSH to contact (with .pem file)

PaaS Example - 1

- Heroku
  - heroku.com
  - ancient-springs-2252.heroku.com
    - Installed intro & addrbook examples
    - Configured to use Postgres DB
  - Many steps, but it's all in their step-by-step guide
    - Install and configure virtualenv
    - Install Heroku's django-toolbelt
    - Set up project
    - Configure databases
    - Use git for deployment of files to server
    - (See dyno running: heroku ps)

PaaS Example - 2

- Google AppEngine
  - appengine.google.com
  - These are in Java
    - webapp-todolist.appspot.com
    - eppinger-homepage.appspot.com
    - eppinger-addrbook.appspot.com
  - Eclipse Plug-in
  - Google Authentication
  - Free w/daily usage limit
  - Administration via the web

SaaS Examples

- There's no need to demo these
- You use them all the time

Homework #7

- You get to deploy to the cloud
- We are working out the details
  - Which platform or platforms
  - What features are required
    - Databases?
    - Pictures?
    - New features?