

# Google App Engine

(Susan Philipose)

# Google App Engine

- Runs web applications on Google's infrastructure
- Competes with Amazon EC2
- Python Run time Environment
- Sandbox
- App Engine Services – URL fetch, mail, memcache and Image manipulation

# Getting Started

- **Application.yaml** –configuration file  
application: helloworld version: 1 runtime:  
python api\_version: 1 handlers: - url: /\*  
script: helloworld.py

## **helloworld.py**

```
print 'Content-Type: text/plain'  
print "  
print 'Hello, world!'
```

# Test the apps

- **Start the webserver using**  
`google_appengine/dev_appserver.py`  
`helloworld/`
- **Browse the webpage**  
`http://localhost:8080/`

# Webapps Framework

- one or more RequestHandler classes that process requests and build responses
- a WSGIApplication instance that routes incoming requests to handlers based on the URL
- a main routine that runs the WSGIApplication using a CGI adaptor

# Hello, webapp!

- ```
from google.appengine.ext import webapp
from google.appengine.ext.webapp.util import run_wsgi_app
```

```
class MainPage(webapp.RequestHandler):
    def get(self):
        self.response.headers['Content-Type'] = 'text/plain'
        self.response.out.write('Hello, webapp World!')
```

```
application = webapp.WSGIApplication(
    [('/', MainPage)],
    debug=True)
```

```
def main():
    run_wsgi_app(application)
```

```
if __name__ == "__main__":
    main()
```

# User Services

- Integrate with Google user accounts

# Data Store

- GQL provides access to the App Engine datastore query engine's features using a familiar SQL syntax



# Uploading Application

- `appcfg.py update helloworld/`