BangPypers June Meetup 2012

Google App Engine with Python

Deepak Garg
Citrix R&D, Bengaluru

deepakgarg.iitg@gmail.com

Contents

- What my App needs ?
- App Engine ?
- Python for App Engine
- Set Up Your Env
- app.yaml
- Wsgi
- Webapp2 framework
- Templates
- Static Files
- DataStore
- Admin & DashBoard
- App Engine Feature Set
- FAQ

What my App needs ?

- App = Business Logic + IT
- IT =>
  - deploying and maintaining web and db, server
  - backup, HA, scale out/in, LB etc. etc..
  - Upgrades, licenses
  - Monitoring
- IT efforts remain consistent and similar across different apps, Business Logic differs
- Developers want to focus on the business logic

App Engine ?

- **Google App Engine** (aka GAE) is a Platform as a Service (PaaS) cloud computing platform for developing and hosting web applications in Google-managed data centers
- Applications are sandboxed and run in Google's best-of-breed data centers around the world
- App Engine offers automatic scaling for web applications—as the number of requests increases for an application
- App Engine automatically allocates more resources for the web application to handle the additional demand
- GAE is ***free*** up to a certain level of consumed resources
- Fees are charged for additional storage, bandwidth, or instance hours

Python for App Engine

- Python → 2.5, 2.7
- Frameworks → Django, web2py, webapp, cherryPy, pylons
- External Libraries → jinja2, lxml, numpy, webob, yaml, pil, pycrypto, setuptools etc..

Set up Your Env

- Install Python :-)
- Download App Engine SDK →
  - a web server application that simulates the App Engine env
  - ***`dev_appserver.py myapp`*** → for a local webserver
  - ***`appcfg.py update myapp`*** → to deploy to the cloud
- Install Eclipse [Optional]
- Install Pydev and App Engine Eclipse Plugin [Optional]

https://developers.google.com/appengine/downloads

app.yaml

- specifies runtime configuration → versions, url handlers etc.

```
application: helloworld
version: 1
runtime: python27
api_version: 1
threadsafe: true

handlers:
- url: /.*
  script: helloworld.app
```

application identifier is "helloworld", name used to register your application with App Engine

Uploading application with diff version no. creates different versions on the server

code runs in the python27 runtime environment, version "1"

app is threadsafe or not

URL Routers → urls matching the regex are captured

WSGI

A WSGI application is a callable with the following signature:

```
def application(environ, start_response)
```

environ is a dictionary which contains everything in os.environ plus variables which contain data about the HTTP reqeust.

start_response is a callable which the application calls to set the status code and headers for a response, before returning the body of the response.

```
start_response(status, response_headers, exc_info=None)
```

The ***return value*** for the WSGI application is an iterable object. Each item in the iterable is a chunk of the HTTP response body as a byte string.

Webapp2 framework

- A webapp2 application has 2 parts:
  - one or more *RequestHandler* classes that process requests and build responses
  - a *WSGIApplication* instance that routes incoming requests to handlers based on the URL

```
import webapp2

class MainPage(webapp2.RequestHandler):
    def get(self):
        self.response.headers['Content-Type'] = 'text/plain'
        self.response.out.write('Hello, webapp World!')

app = webapp2.WSGIApplication([('/', MainPage)],
                              debug=True)
```

Templates

- Import jinja2 in *app.yaml*

```
libraries:
- name: jinja2
  version: latest
```

- Point jinja2 to your template dir

```
jinjadir = jinja2.Environment(loader=jinja2.FileSystemLoader(template_dir))
```

- Load your templates

```
fom = jinjadir.get_template('form.template')
```

- Render your templates

```
fom.render({'items':['a', 'b', 'c']})
```

Static Files

- create your static directory
- point to it in app.yaml →

```
- url: /mystatic
  static_dir: mystatic
```

Deploy

- register your *app_id* → https://appengine.google.com/
- point to it in app.yaml →

```
application: app_id
version: 1
```

- update version no.
- upload your app →

```
appcfg.py update myapp
```

- browse your app :)

http://app_id.appspot.com/

DataStore

- Google's Bigtable high-performance distributed database
- Not a Relational DB – scales awesomely well !
- Replicate data across multiple DCs
- Data is written to the Datastore in objects known as entities. Each entity has a key that uniquely identifies it
- Every entity has a unique identifier
- An entity can optionally designate another entity as its parent and thus form a hierarchy

Object Oriented	Relational DB	DataStore
Class	Table	Kind
Object	Record	Entity
Attribute	Column	Propery

https://developers.google.com/appengine/docs/python/datastore/entities

DataStore

- define your entity class
- create object (entity)
- put it in db
- check it in DB viewer

```
from google.appengine.ext import db

class PyperDB(db.Model):
    name = db.StringProperty(multiline=True);
    status = db.BooleanProperty();
    user_id = db.UserProperty();
    time = db.DateTimeProperty(auto_now_add=True)
```

```
us = PyperDB(name=val, status=False, user_id=uid)
us.put();
```

- Model defines a unique id by default for each entity
- Fetch data using GQL queries
- **Delete**

```
key = us.key().id_or_name()
```

```
result = db.GqlQuery("SELECT * "
                     "FROM PyperDB "
                     "WHERE ANCESTOR IS :1 "
"ORDER BY date DESC LIMIT 10",key_value)
```

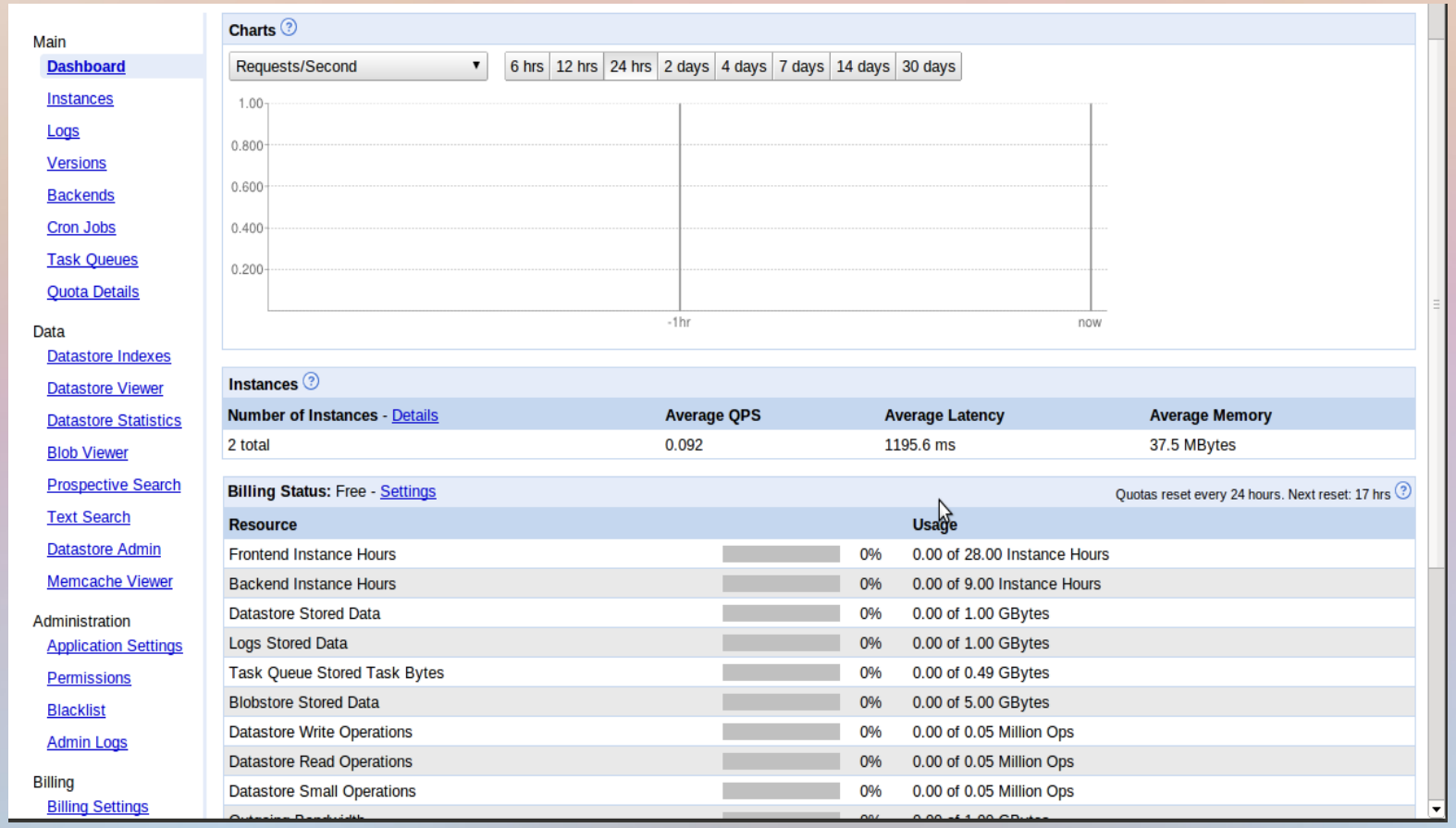
```
uis = PyperDB.all()

for p in uis:
    p.delete()
```

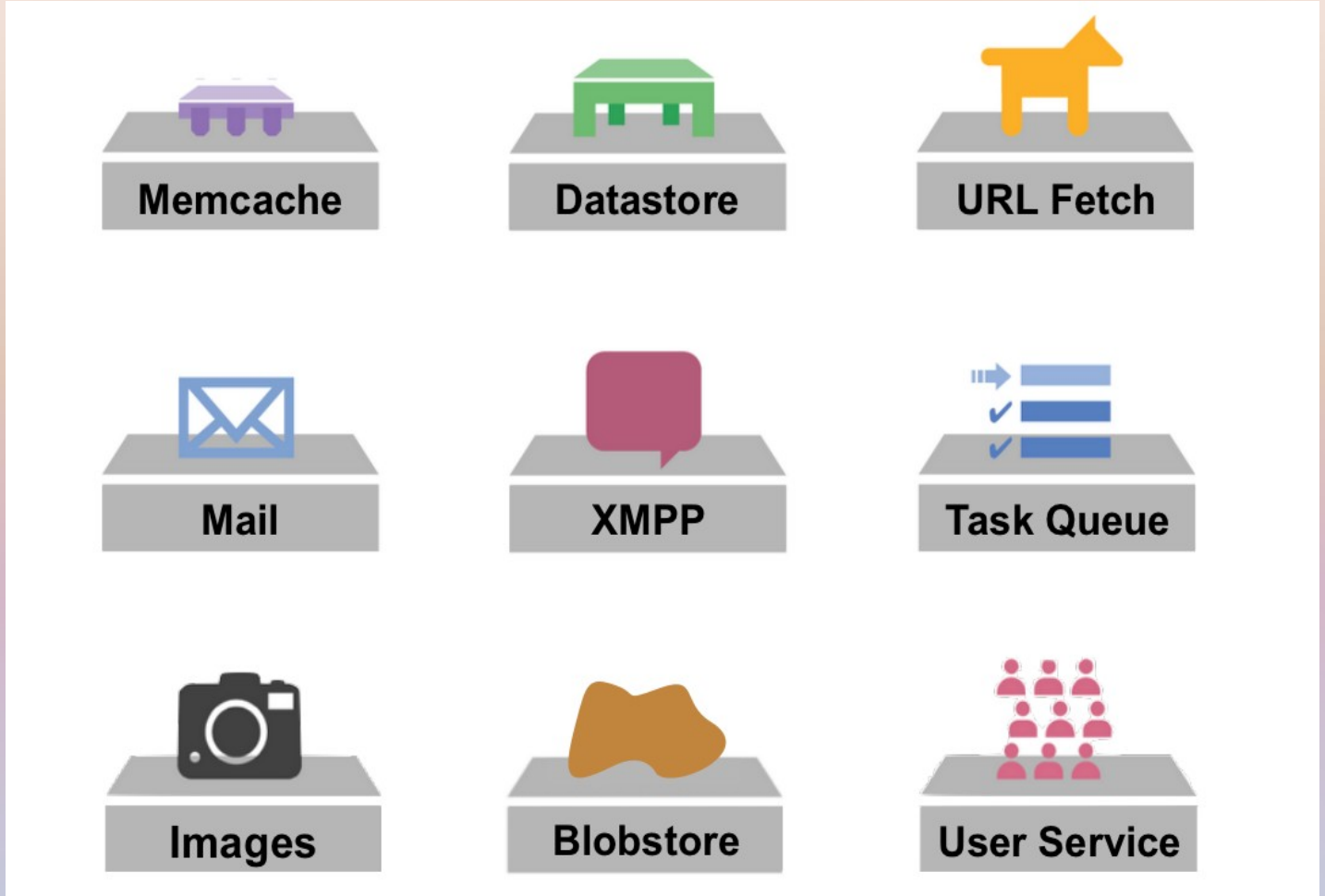
https://developers.google.com/appengine/docs/python/datastore/gqlreference

Admin & DashBoard

- Local Admin console running at http://localhost:9999/_ah/admin
- GAE Dashboard, logs, DS viewer etc. are used to manage your app



GAE Features



https://developers.google.com/appengine/docs/python/apis

FAQ

- Can I run existing django apps on App Engine ?
- Can I move my App Engine application to my hosting environment ?
  - Yes, Look at django-rel (django for non-relational db
    http://www.allbuttonspressed.com/projects/django-nonrel
    http://www.allbuttonspressed.com/projects/djangoappengine
- Can I run any Python code on App Engine ?
  - With certain limitations →
    - cannot write to FS but you can read (from flat file you uploaded)
    - cannot open a socket, can't make system call,
    - can't respond slowly to requests (60 secs)

Many Thanks !

deepakgarg.iitg@gmail.com

@donji

github.com/gargdeepak