Google App Engine

Outline

Cloud Computing (Brief)
What is Google App Engine (GAE)?
GAE Services
Storing Data and Data Processing
Real World Examples

Cloud Computing

“Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction.” -- NIST, The NIST Definition of Cloud Computing, September 2011.

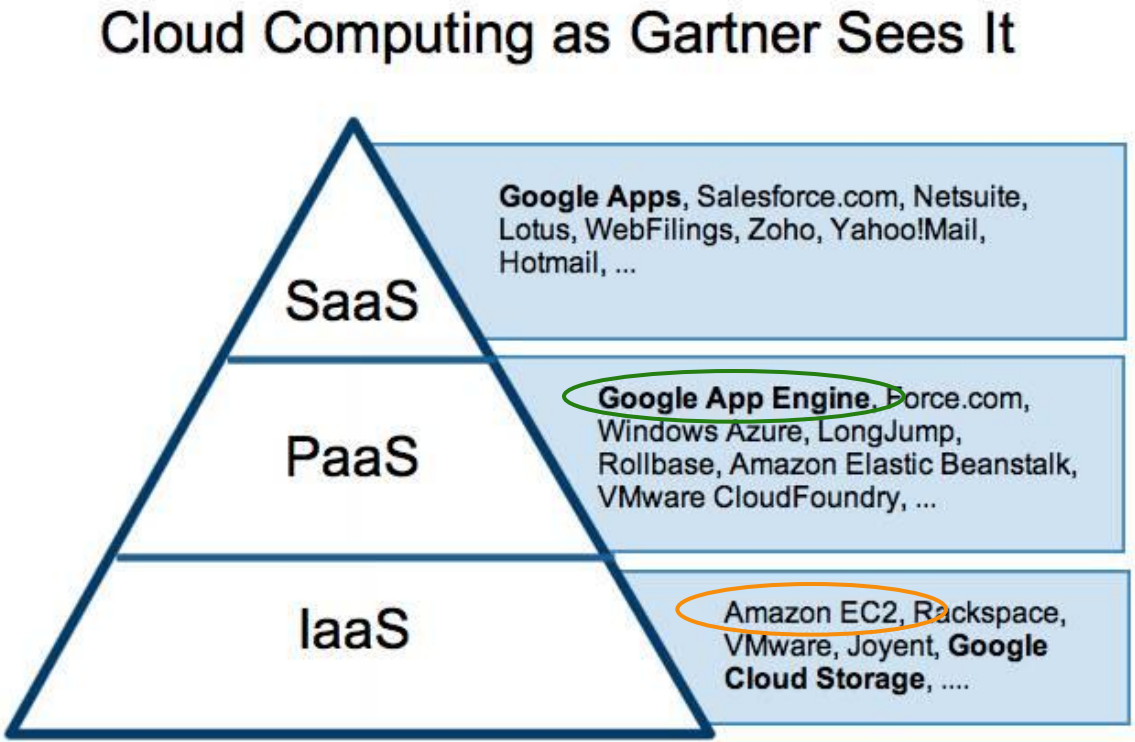
Cloud Computing

Key ideas:

- On-demand access
- Shared pool of resources
- Minimal service provider management/interaction

But, there’s a spectrum of cloud products & services...

Cloud Computing



What Is Google App Engine (GAE)?

Is it a web server?
Is it an application development environment?
Is it a gateway to other Google products & services?

YES

What Is Google App Engine (GAE)?

It's a web server

- Runs on Google's infrastructure using WSGI
- Can use a variety of Python development frameworks (Django, Jinja, etc.)
- Scales automatically

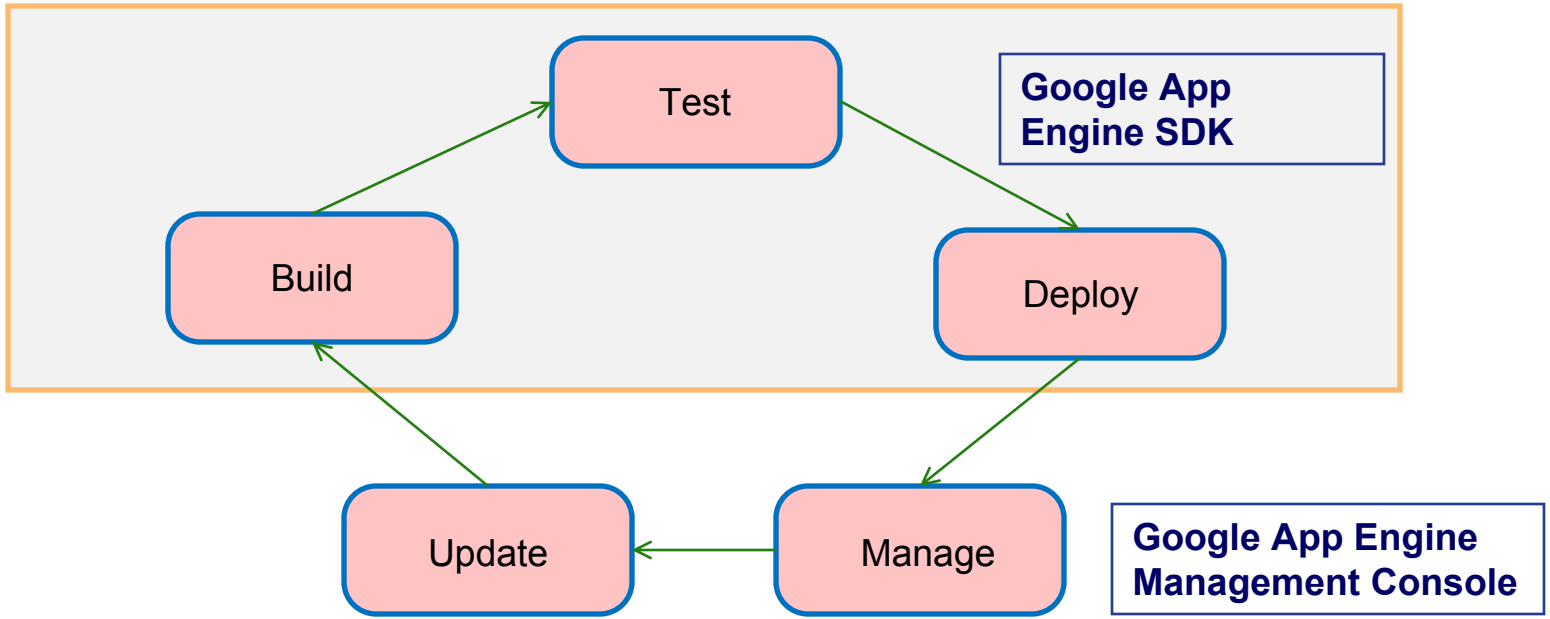
What Is Google App Engine (GAE)?

It's an application development environment

- Has its own SDK
- Build & test locally, deploy globally
- SDKs for Java, Python, Go & PHP
- Can develop on PCs, Mac OS or Linux
- Right price, too (FREE!)

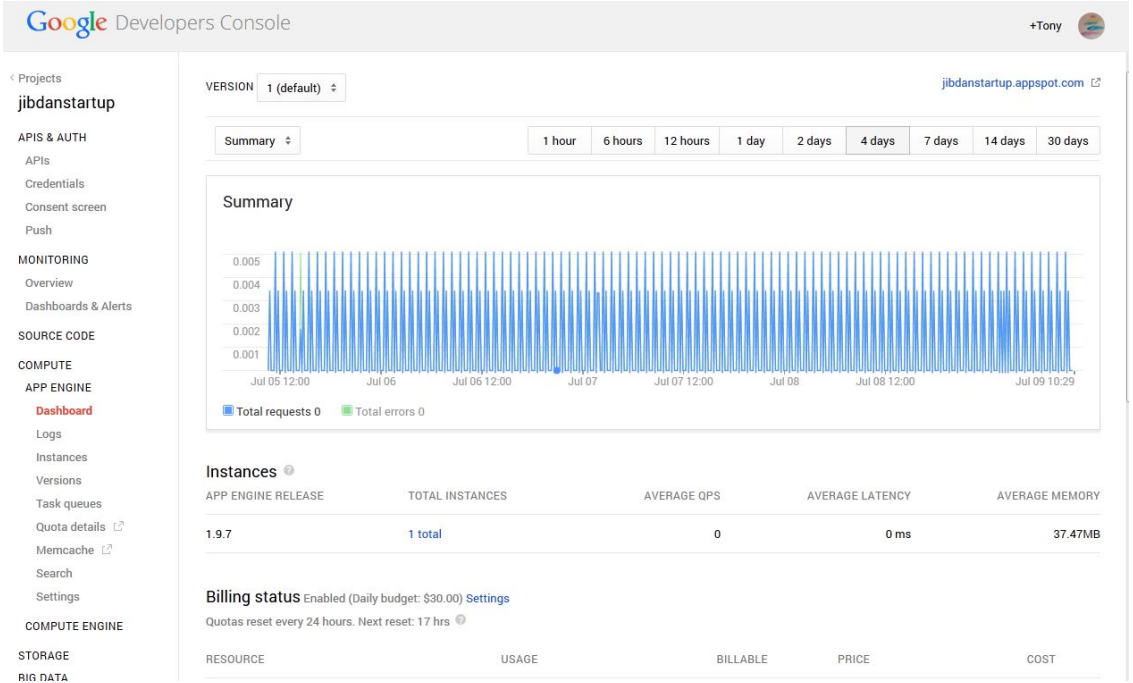
What Is Google App Engine (GAE)?

It's an application development environment



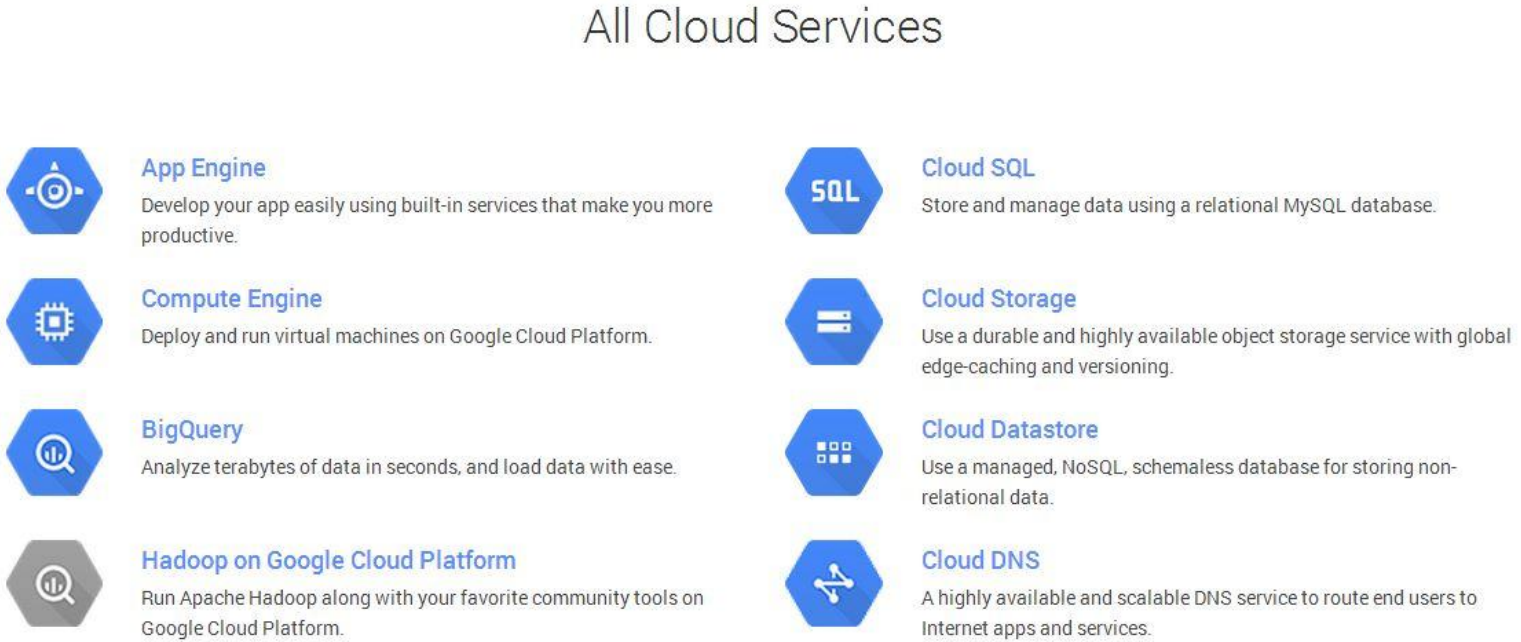
What Is Google App Engine (GAE)?

It’s an application development environment



What Is Google App Engine (GAE)?

It’s a gateway to other Google products



GAE Services

APIs galore:

- Memcache
- URLFetch
- Datastore
- SMS & Voice
- Mail
- OAuth
- Search

And many others including other Google products

GAE Services

URLFetch

Similar to urllib but optimized to run in Google's environment

urllib2

```
import urllib2

url = "http://www.google.com/"
try:
  result = urllib2.urlopen(url)
  doSomethingWithResult(result)
except urllib2.URLError, e:
  handleError(e)
```

urlfetch

```
from google.appengine.api import urlfetch

url = "http://www.google.com/"
result = urlfetch.fetch(url)
if result.status_code == 200:
  doSomethingWithResult(result.content)
```

GAE Services

Datastore

No-SQL database built-in to the framework:

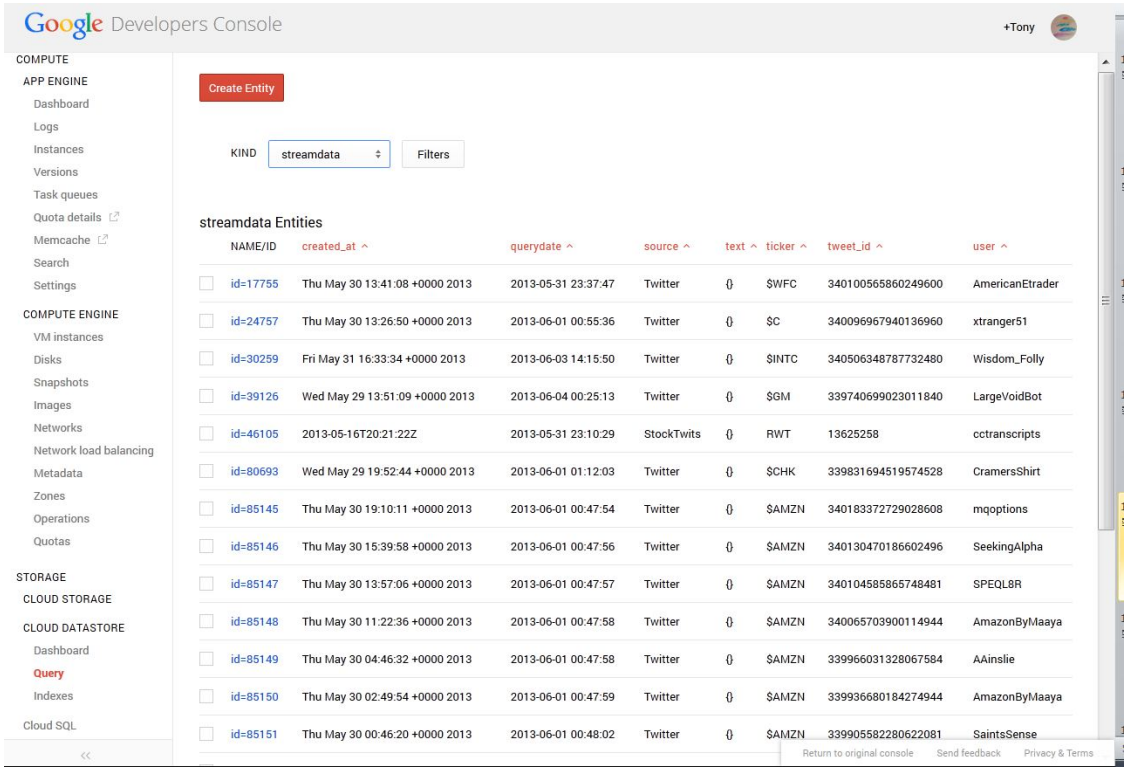
```
from google.appengine.ext import ndb

class Account(ndb.Model):
  username = ndb.StringProperty()
  userid = ndb.IntegerProperty()
  email = ndb.StringProperty()
```

Query:

```
qry = Account.query(Account.userid == 42)
```

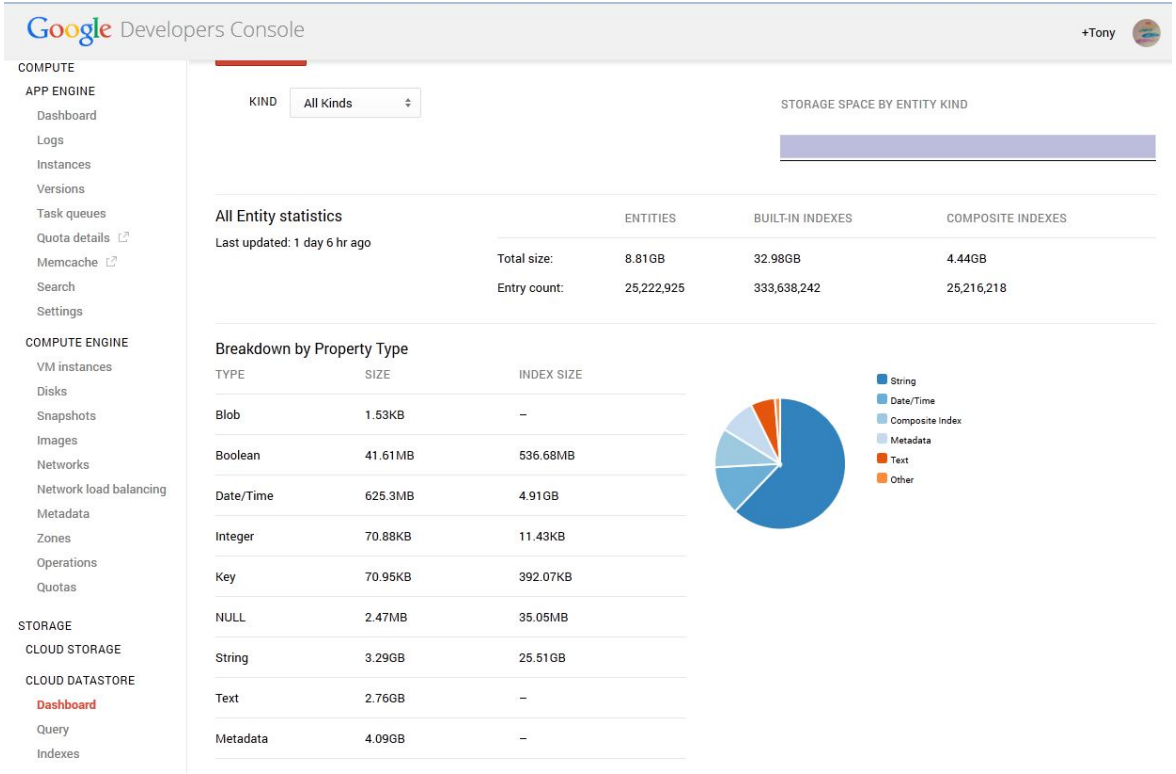
GAE Services

Datastore



GAE Services

Datastore



GAE Services

Cron jobs

- Great for repetitive tasks
  - “Every 10 minutes, run this Python module.”
- Managed via the cron.yaml file:

```
cron:
- description: daily summary job
  url: /tasks/summary
  schedule: every 24 hours
- description: monday morning mailout
  url: /mail/weekly
  schedule: every monday 09:00
  timezone: Australia/NSW
- description: new daily summary job
  url: /tasks/summary
  schedule: every 24 hours
  target: beta
```

Storing Data and Data Processing

- Beyond the Datastore
- Several Options:
  - Google Cloud Storage
  - Google Cloud SQL
  - Google BigQuery
  - Hadoop on Google Cloud (Requires Google Compute Engine)

Storing Data and Data Processing

Google Cloud Storage

- Store massive amounts of data offline, cheaply
  - Ideal for log files, old app data that doesn't need to be retrieved by app users.
- Example: 106GB ~$3/month on Google Cloud Storage. 45GB ~$8/month on GAE

Storing Data and Data Processing

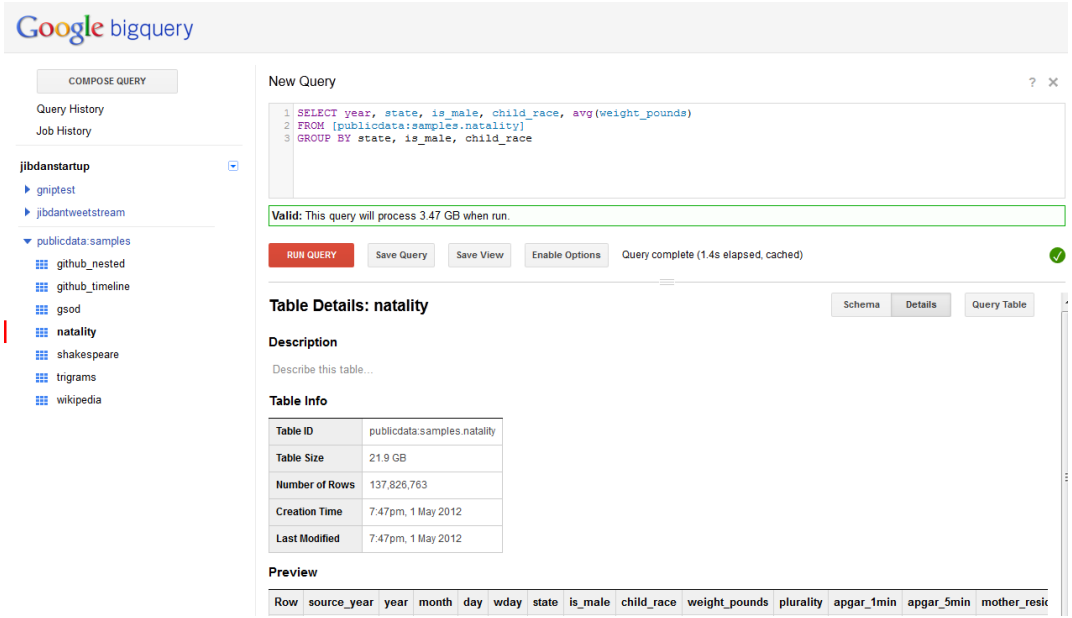
Google Cloud SQL

- Run your own MySQL instances
- No server configuration, encryption, replication, patch management or backups to set up. Google does it all.

Storing Data and Data Processing

Google BigQuery

- Tool to analyze large data sets



Storing Data and Data Processing

Hadoop on Google Cloud

- Processing data vs. analyzing data
- Usually involves deploying a cluster of VMs, hence used on Google Compute Engine
- Simpler App Engine solutions: MapReduce Python library, Managed VMs

Real World Examples