

Universitatea “Alexandru Ioan Cuza”
Facultatea de Informatică

Conf. Dr. Lenuța Alboaie
adria@info.uaic.ro

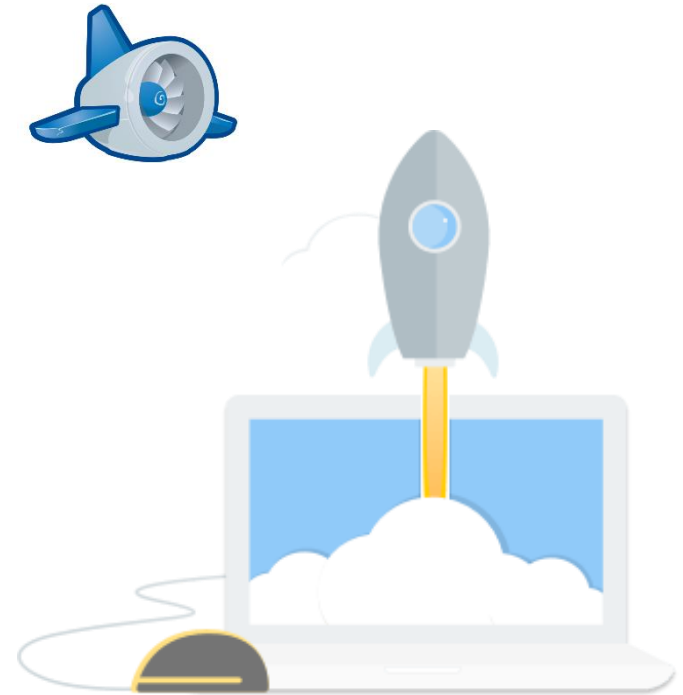
A graphic featuring a globe, clouds, and binary code, with the words "Cloud Computing" written in a cursive font across it.

Google in Cloud - Aplicatii -

Cuprins

Google App Engine

- Standard Environment
- Flexible Environment



Google App Engine

Google App Engine

- 2017 | Mediul de dezvoltare App Engine ofera

Standard Environment

About the Standard Environment

Python 2.7

Java 7

PHP 5.5

Go 1.6

Flexible Environment

About the Flexible Environment

Python 2.7, 3.5

Java 8

Node.js

Go 1.8

Ruby

PHP 5.6, 7

.NET

Custom Runtimes

Known Issues

[<https://cloud.google.com/appengine/docs/>]

Google App Engine

Google App Engine

- Mediul de dezvoltare App Engine ofera:
 - **App Engine Standard Environment - *Sandbox Environment*** –
cod + server Web+ *language runtime (modificat)* a.i. se respecta restrictiile *sandbox* => aplicatiile ruleaza intr-un mediu securizat, izolat de nivelul hardware, sistemul de operare si localizarea fizica a serverului
 - Aceasta limitare permite distribuirea cererilor web la mai multe servere web (pornirea/oprirea lor) in functie de cresterea/scaderea cererilor
 - **App Engine Flexible Environment** – permite rularea de aplicatii App Engine folosind **Google Compute Engine**
 - “*VM hosting environment offers more flexibility and provides more CPU and memory options*”.
 - Nu mai exista restrictiile din *sandbox runtimes*
- [<https://cloud.google.com/appengine/docs/>]

Feature	Flexible environment	Standard environment
Instance startup time	Minutes	Milliseconds
Maximum request timeout	60 minutes	60 seconds
Background threads	Yes	Yes, with restrictions
Background processes	Yes	No
SSH debugging	Yes	No
Scaling	Manual, Automatic	Manual, Basic, Automatic
Writing to local disk	Yes, ephemeral (disk initialized on each VM startup)	No
Modifying the runtime	Yes (through Dockerfile)	No
Automatic in-place security patches	Yes (excludes container image runtime)	Yes
Network access	Yes	Only via App Engine services (includes outbound sockets)
Supports installing third-party binaries	Yes	No
Location	North America or Asia Pacific. The flexible environment is not supported in Europe.	North America, Asia Pacific, or Europe
Pricing	Based on usage of vCPU , memory , and persistent disks	Based on instance hours

[<https://cloud.google.com/appengine/docs/the-appengine-environments>] 5

Google App Engine

App Engine Flexible Environment

- Folosind Serviciile Google Compute Engine, App Engine asigura automat:
 - scalarea si *load balancing*
 - verificarea starii instantelor si co-locarea optima alaturi de alte servicii din proiect
 - se ofera acces de root la instantele Compute Engine VM (accesul ssh este dezactivat in mod implicit)
 - suport pentru update-uri
 - se permite modificarea *environment-ului* si a sistemului de operare prin utilizarea *Dockerfiles*
- Sunt suportate nativ: *microservices, authorization, SQL and NoSQL databases, traffic splitting, logging, versioning, security scanning, si content delivery networks*

Google App Engine

App Engine Flexible Environment

– Runtimes

Google-supplied Dockerfile => standard runtime

- *ofera suport nativ pentru: Java 8 / Servlet 3.1 / Jetty 9, Python 2.7 and Python 3.5, Node.js, Ruby, PHP, .NET core, and Go*
- *Nu exista restrictii sandbox*
- *Se pot face interogari privind starea VM, se ofera suport pentru servicii ca: Datastore, Memcache, Task Queues, Logging, Users*

Docker image sau Dockerfile modificat (open source) => custom runtime

- *Se foloseste daca se doreste scrierea de cod in alte limbaje*
- *Se pot configura componente ca: interpretoare sau servere de aplicatii*

– Performanta

- *O gama larga de configuratii ale CPU si memoriei care odata configurate vor fi furnizate de AppEngine in mod automat*

Google App Engine

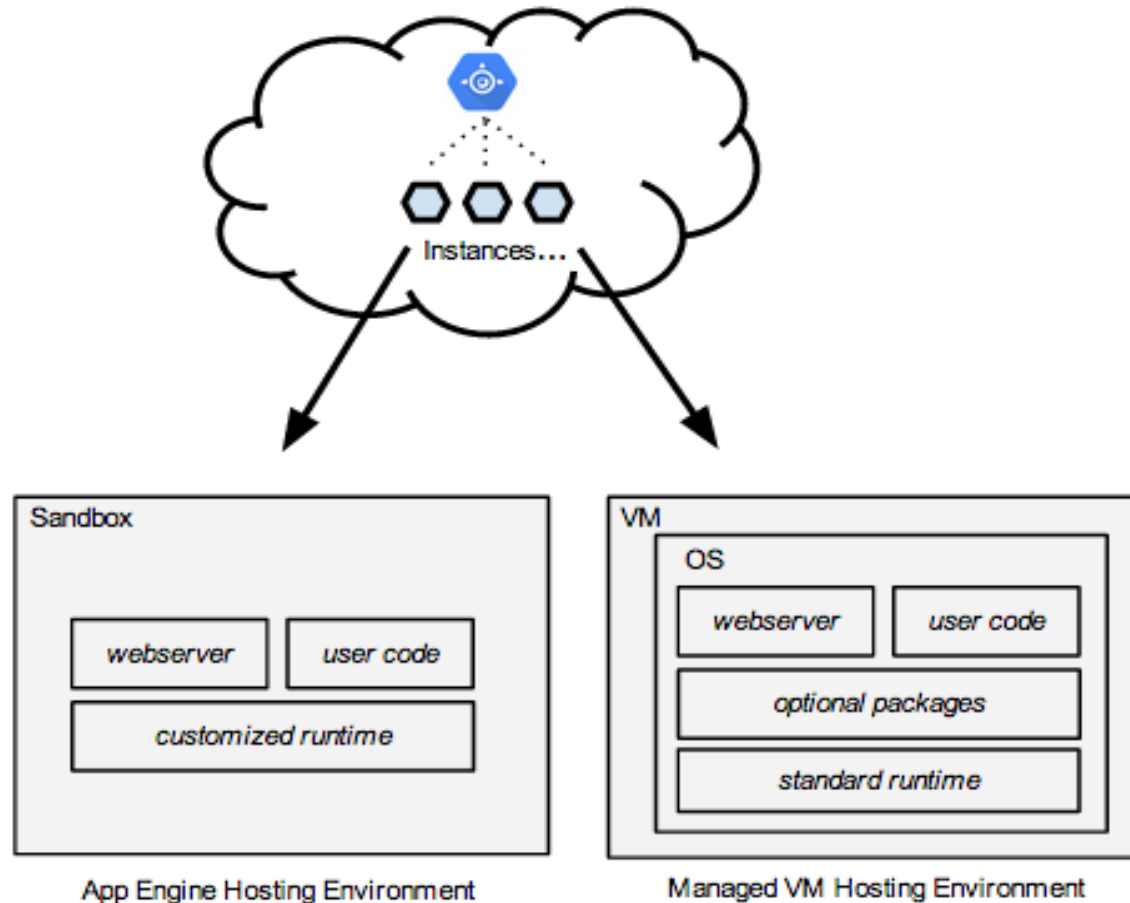
- App Engine Flexible Environment or Standard Environment

- Source code that is written in a version of any of the supported programming languages:
Python, Java, Node.js, Go, Ruby, PHP, or .NET
- Runs in a Docker container that includes a custom runtime or source code written in **other programming languages**.
- **Depends on other software, including operating system packages** such as imagemagick, ffmpeg, libgit2, or others through apt-get.
- Uses or depends on frameworks that include **native code**.
- Accesses the resources or services of your Cloud Platform project that reside in the **Compute Engine network**.
- Source code is written in **specific versions of the supported programming languages**:
 - Python 2.7
 - Java 7
 - PHP 5.5
 - Go 1.6
- Intended to **run for free or at very low cost**, where you pay only for what you need and when you need it. For example, your application can scale to 0 instances when there is no traffic.
- Experiences **sudden and extreme spikes of traffic** which require immediate scaling.

[<https://cloud.google.com/appengine/docs/the-appengine-environments>]

Google App Engine

- O aplicatie poate contine module care ruleaza in medii de gazduire diferite e.g se poate folosi pentru frontend *sandbox* si pentru procesare sa se foloseasca ***App Engine Flexible Environment***



Google App Engine

- App Engine Flexible Environment

<https://cloud.google.com/appengine/docs/flexible/>

GO

JAVA 8

PHP 5 / 7

PYTHON 2.7 / 3.5

.NET

NODE.JS

RUBY

CUSTOM RUNTIMES

Google App Engine

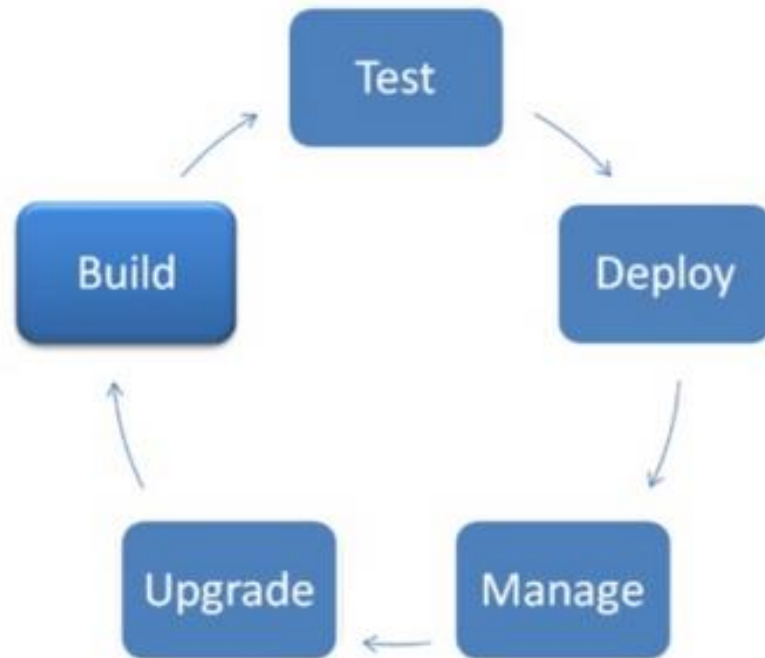


Figura. Pasii dezvoltarii unei aplicatii
(eng. *Application lifecycle*)

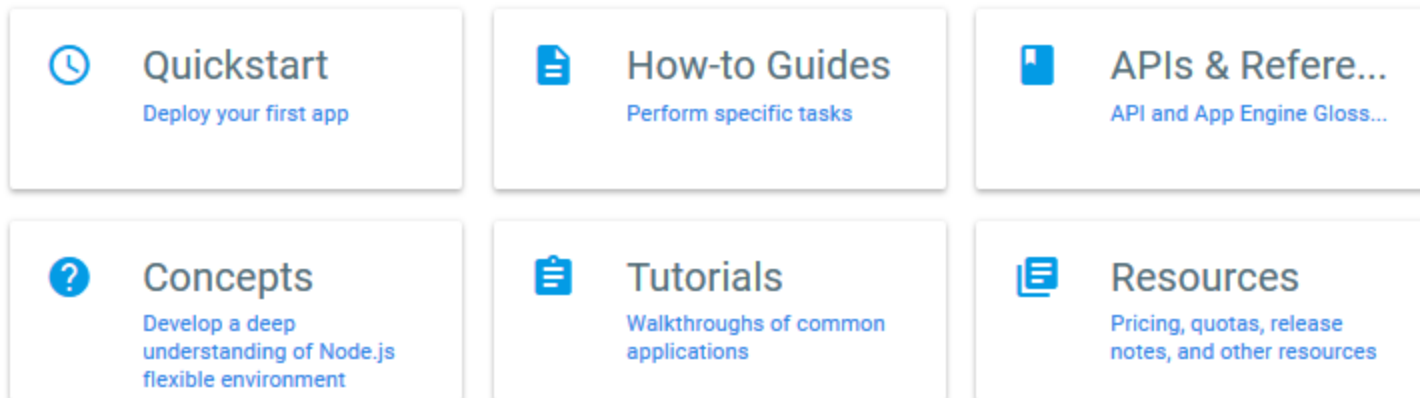
[<http://www.slideshare.net/rajdeep/introduction-to-google-app-engine-presentation>]

Google App Engine

- App Engine Flexible Environment - NodeJS

<https://cloud.google.com/appengine/docs/flexible/nodejs/>

The App Engine flexible environment is based on Google Compute Engine and automatically scales your app up and down while balancing the load.



Google App Engine | Flexible Environment



Google Cloud Platform helloworld2017

AM & Admin Settings DELETE

Project name helloworld2017

Project ID helloworld2017-16

Project number 418119710190

Save

New Project

Project name

HelloNode

Your project ID will be hellonode-163216 Edit

CANCEL CREATE

[<https://cloud.google.com/appengine/docs/flexible/nodejs/quickstart>]

Google App Engine | Flexible Environment



.....

Before

1. U

a

v

e

2. I

App Engine is *regional*, which means the infrastructure that runs your apps is located in a specific region and is managed by Google to be redundantly available across [all the zones within that region](#).

Meeting your latency, availability, or durability requirements are primary factors for selecting the region where your app's are run. You can generally select the region nearest to your app's users but you should consider the location of the [other Cloud Platform products and services](#) that are utilized by your app. Using services across multiple locations can affect your app's latency as well as [pricing](#).

App Engine is available in the following regions:

- us-central
- us-east1
- europe-west
- asia-northeast1

Google App Engine | Flexible Environment

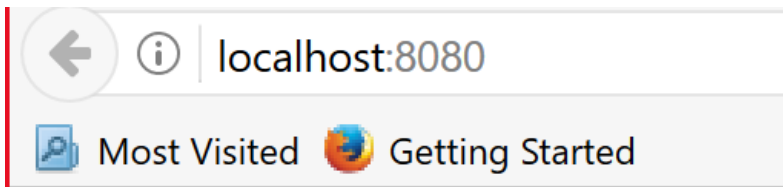


- Download -> Google Cloud SDK and then initialize the gcloud tool
- npm install – in directorul cu fisierele sursa

```
Google Cloud SDK Shell
D:\gae\FlexibleEnv\hello-world>npm install
[ ..... ] - fetchMetadata: sill mapToRegistry uri https://registry.npmjs.org/util
appengine-hello-world@0.0.1 D:\gae\FlexibleEnv\hello-world
```

- npm start
 - => pornirea locala a serverului

```
D:\gae\FlexibleEnv\hello-world>npm start
> appengine-hello-world@0.0.1 start D:\gae\FlexibleEnv\hello-world
> node app.js
App listening on port 8080
Press Ctrl+C to quit.
```



Hello, world!

[<https://cloud.google.com/appengine/docs/flexible/nodejs/quickstart>]

Google App Engine | Flexible Environment

```
D:\gae\FlexibleEnv\hello-world>gcloud init
Welcome! This command will take you through the configuration of gcloud.

Settings from your current configuration [default] are:
[core]
account = lalboaie@gmail.com
disable_usage_reporting = False
project = helloworld2017-163019

Your active configuration is: [default]

Updates are available for some Cloud SDK components. To install them,
please run:
  $ gcloud components update

Pick configuration to use:
[1] Re-initialize this configuration [default] with new settings
[2] Create a new configuration
Please enter your numeric choice: 1
```


Google App Engine | Flexible Environment

```
(https://cloud.google.com/compute) settings (Y/n)? y
Which Google Compute Engine zone would you like to use as project
default?
If you do not specify a zone via a command line flag while working
with Compute Engine resources, the default is assumed.
[1] asia-east1-b
[2] asia-east1-a
[3] asia-east1-c
[4] asia-northeast1-a
[5] asia-northeast1-c
[6] asia-northeast1-b
[7] europe-west1-b
[8] europe-west1-c
[9] europe-west1-d
[10] us-central1-b
[11] us-central1-f
[12] us-central1-a
[13] us-central1-c
[14] us-east1-b
[15] us-east1-c
[16] us-east1-d
[17] us-west1-a
[18] us-west1-b
[19] Do not set default zone
Please enter numeric choice or text value (must exactly match list
item): 12
```

Google Cloud SDK Shell - gcloud app deploy

```
D:\gae\FlexibleEnv\hello-world>gcloud app deploy
```

Google App Engine | Flexible Environment

Google Cloud SDK Shell - gcloud app deploy

```
D:\gae\FlexibleEnv\hello-world>gcloud app deploy
You are creating an app for project [hellonode2017].
WARNING: Creating an App Engine application for a project is irreversible and the region
cannot be changed. More information about regions is at
https://cloud.google.com/appengine/docs/locations.
```

```
Please choose the region where you want your App Engine application
located:
```

```
[1] us-central    (supports standard and flexible)
[2] us-east1      (supports standard and flexible)
[3] europe-west   (supports standard and flexible)
[4] asia-northeast1 (supports standard and flexible)
[5] cancel
Please enter your numeric choice: 1
```

```
Creating App Engine application in project [hellonode2017] and region [us-central]....done.
You are about to deploy the following services:
```

```
- hellonode2017/default/20170331t202241 (from [D:\gae\FlexibleEnv\hello-world\app.yaml])
  Deploying to URL: [https://hellonode2017.appspot.com]
```

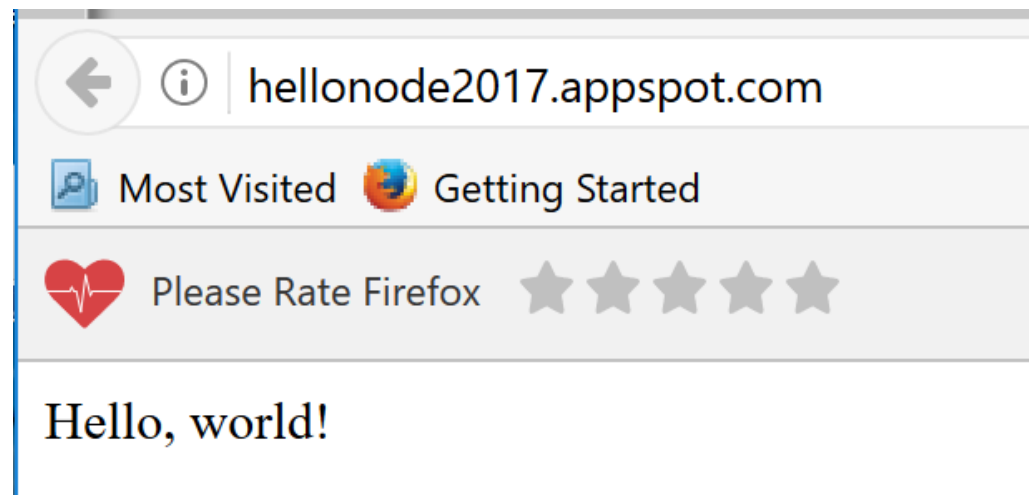
```
Do you want to continue (Y/n)? y
```

```
If this is your first deployment, this may take a while...|
```

thani

Google App Engine | Flexible Environment

```
Updating service [default]...done.  
Deployed service [default] to [https://hellonode2017.appspot.com]  
  
You can stream logs from the command line by running:  
$ gcloud app logs tail -s default  
  
To view your application in the web browser run:  
$ gcloud app browse  
  
D:\gae\FlexibleEnv\hello-world>
```



Google App Engine | Flexible Environment

DASHBOARD

ACTIVITY

Service/API

compute.googleapis.com

Now

Limit

1

Region

global

For more information, visit your [quota page](#).

Today

8:31 PM



Completed: Delete VM

aef-default-20170331t202241-q66l was deleted

8:29 PM



Completed: Update App ...

Lenuta Alboaie updated default

8:29 PM



Completed: Update App ...

Lenuta Alboaie updated default

Google App Engine | Flexible Environment

Enabled APIs

Some APIs are enabled automatically

Activity for the last hour

1 hour

6h

12h

1 day

2d

4d

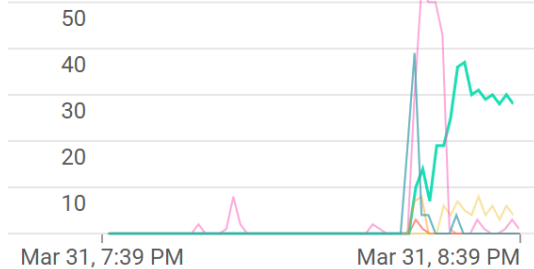
7d

14d

30d

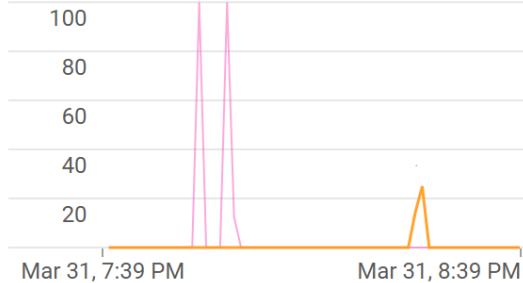
Traffic

Requests/sec



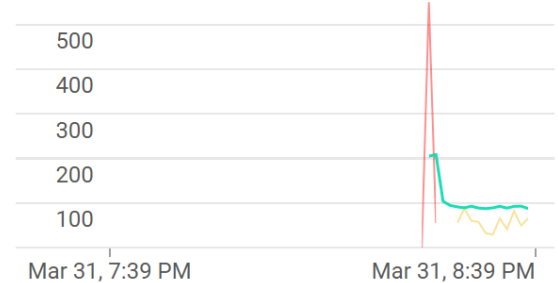
Errors

Percent of requests



Median latency

Milliseconds



Google App Engine

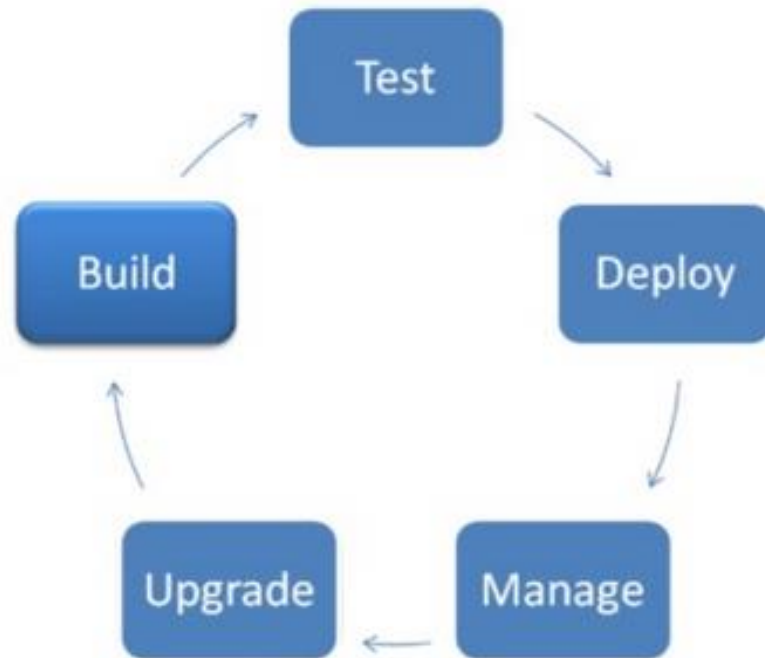


Figura. Pasii dezvoltarii unei aplicatii
(eng. *Application lifecycle*)

[<http://www.slideshare.net/rajdeep/introduction-to-google-app-engine-presentation>]

Google App Engine | Standard Environment



Instrumente pentru dezvoltatori

- Java – se poate downloada Java SDK in format plugin pentru Eclipse
- Python – Python SDK sub forma unei aplicatii GUI
 - Ambele SDK-uri sunt disponibile si ca arhive zip
- Fiecare SDK – contine un server Web de dezvoltare care permite rularea de aplicatii in mod local, si care simuleaza *runtime environment*, *datastore* si serviciile.
- Obs. Se poate utiliza Eclipse pentru dezvoltarea aplicatiilor Python folosind PYDev (extensie Eclipse care include un debugger Python interactiv)

Python 2.7

Java 7

PHP 5.5

Go 1.6

[<https://cloud.google.com/appengine/docs>]

Google App Engine | Standard Environment



Crearea unei aplicatii folosind Python SDK

- Descarcarea unui App Engine SDK in functie de limbajul dorit
- <http://code.google.com/appengine/>
- Fiecare SDK contine un server web de test
 - Se simuleaza conditiile din GAE (sandbox, serviciile existente)
- Deoarece SDK pentru Python a fost dezvoltat inainte de cel de Java, anumite facilitati se gasesc doar in primul
- <http://www.python.org/download/>
 - *Obs. The Python SDK is not compatible with Python 3*

>> dev_appserver.py --help

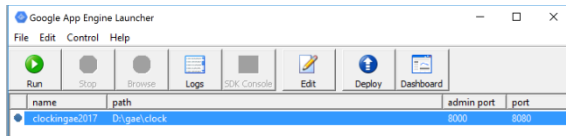
```
D:\gae\helloworld2017>dev_appserver.py --help
usage: dev_appserver.py [-h] [-A APP_ID] [--host HOST] [--port PORT]
                        [--admin_host ADMIN_HOST] [--admin_port ADMIN_PORT]
                        [--auth_domain AUTH_DOMAIN] [--storage_path PATH]
                        [--log_level {debug,info,warning,critical,error}]
                        [--max_module_instances MAX_MODULE_INSTANCES]
                        [--use_mtime_file_watcher [USE_MTIME_FILE_WATCHER]]
                        [--threadsafe_override THREADSAFE_OVERRIDE]
                        [--php_executable_path PATH]
                        [--php_remote_debugging [PHP_REMOTE_DEBUGGING]]
```


Google App Engine

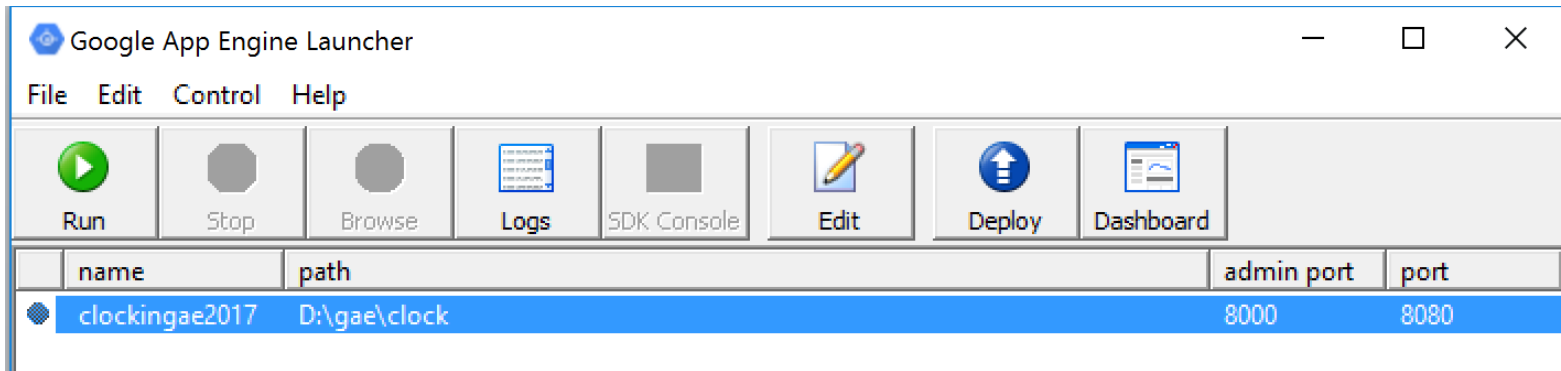


Crearea unei aplicatii in Python | Preliminarii

- Pentru Windows si MacOS, Python SDK include o aplicatie *Google App Engine Launcher* – permite managementul proiectelor Python folosind o interfata grafica



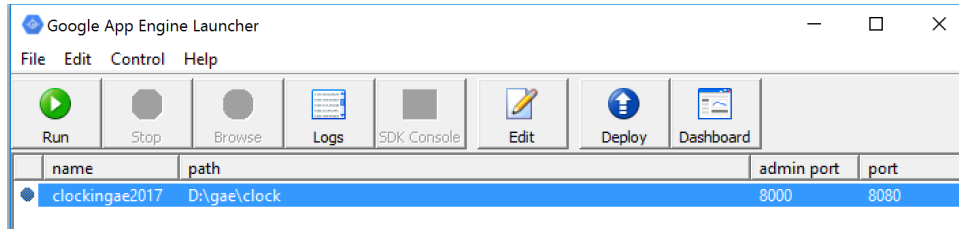
-Run - pornirea serverului Web de dezvoltare



Google App Engine



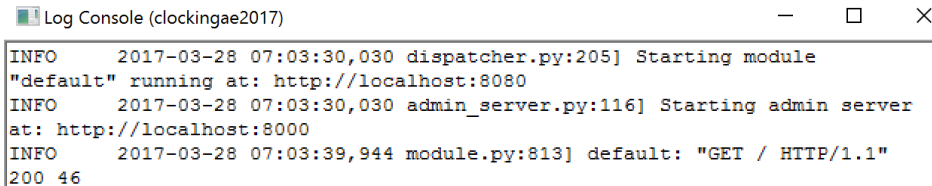
Crearea unei aplicatii in Python | Preliminarii



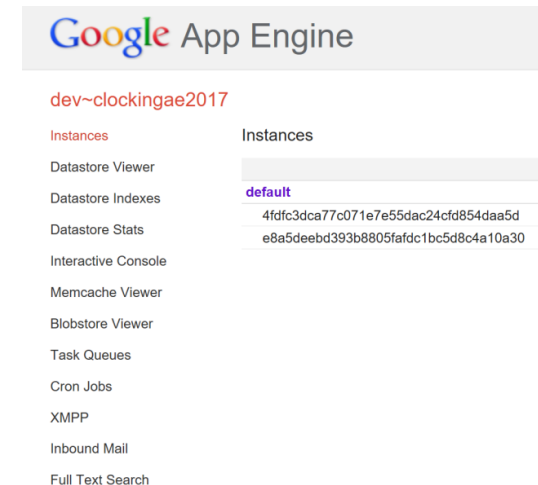
The time is: 2017-03-28 04:03:39.945000

- **Browse** – vizualizarea rezultatelor aplicatiei

- **Logs**



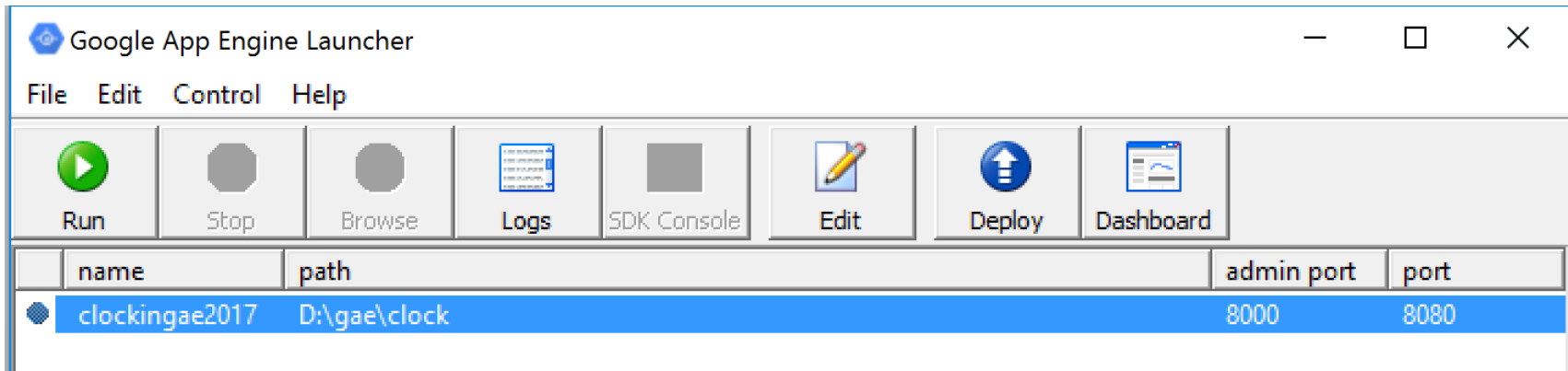
- **SDK Console** – interfata Web pentru serverul de dezvoltare (inspectarea aplicatiilor care ruleaza, instrumente pentru inspectarea *datastore*-ului, Consola interactiva pentru executarea de instructiuni Python)



Google App Engine



Crearea unei aplicatii in Python | Preliminarii

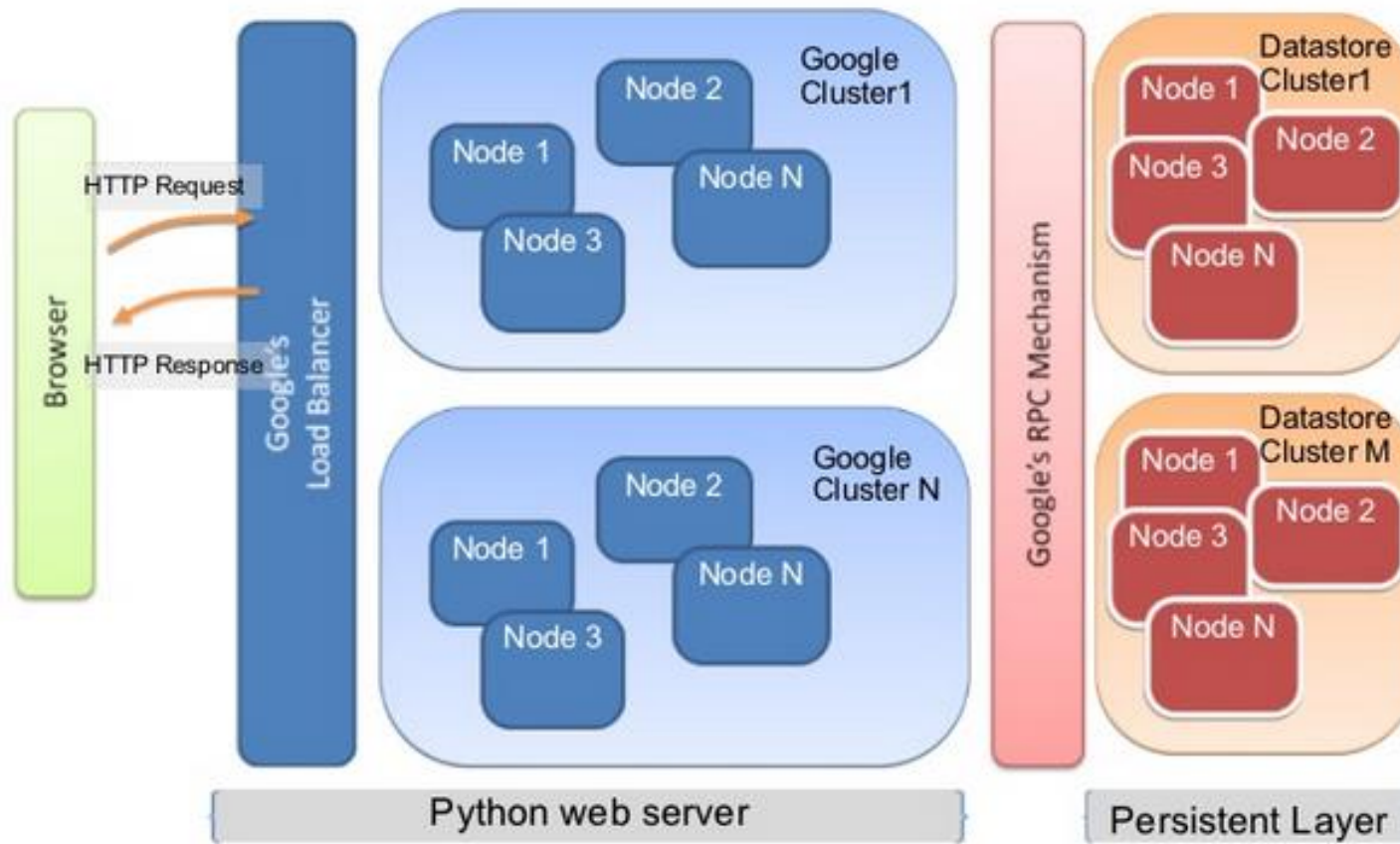


- **Dashboard** – deschide o consola de administrare App Engine pentru aplicatia rulata
- **Deploy** – incarcarea proiectului in GAE
 - Trebuie inregistrat un *application ID* cu AppEngine si modificat fisierul de configurare a aplicatiei



Google App Engine

App Engine Physical Deployment Diagram



[<http://www.slideshare.net/rajdeep/introduction-to-google-app-engine-presentation>]

Google App Engine



Crearea unei aplicatii in Python | Preliminarii

- O aplicatie App Engine raspunde la cereri Web cu ajutorul *request handlers*
 - Rutine care accepta parametrii de cerere si returneaza raspunsul
 - App Engine hotaraste ce *request handler* sa foloseasca pentru o anumita cerere de la un URL cerut
 - Exista un fisier de configurare care face maparea *handler* – URL
 - Obs. Pentru resursele statice (vezi curs) , se poate specifica URL-ul corespunzator pentru aceste fisiere
 - Exista in fisierul de configurare metadata asociate aplicatiei: versiune, ID
 - Atunci cand se face instalarea in App Engine (*deploy*) , fisierele aplicatiei (cod, fisiere de configurare, fisierele statice) au asociate aceste metadata
 - O aplicatie poate avea si fisiere de configurare specifice pentru servicii (indexarea din *datastore*, *task queues*, *scheduled tasks*) – acestea sunt asociate aplicatiei in general, si nu sunt specifice unei versiuni

Google App Engine



Crearea unei aplicatii in Python

- Aplicatia afiseaza timpul curent

Name	Ext	Size	Dat
[.]		<DIR>	03/
favicon	ico	1,150	02/
clockcgi	py	120	03/
app	yaml	192	03/

app.yaml - fisier de configurare

Obs. Directorul continind **app.yaml** este directorul root al aplicatiei

```
app.yaml * | main.py |
application: clockingae2017
version: 1
runtime: python
api_version: 1

handlers:
- url: /favicon\.ico
  static_files: favicon.ico
  upload: favicon\.ico

- url: .*
  script: main.py
```

Orice cerere pentru aceasta aplicatie (orice URL care se potriveste expresiei regulate /.*) va fi tratata de **main.py** (contine codul pentru *request Handler*)

Google App Engine



Crearea unei aplicatii in Python

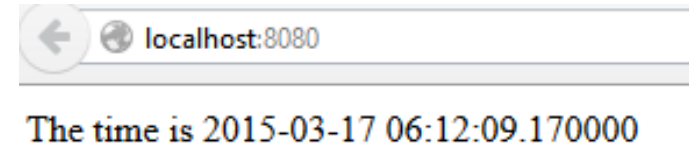
- Cream un fisier main.py

program.xml | app.yaml | clockcgi.py |

```
import datetime
print 'Content-Type: text/html'
print ' '
print '<p> The time is %s</p>' % str(datetime.datetime.now())
```

-Importam modulul *datetime* din bibliotecile Python

>>dev_appserver.py app.yaml



- Comunicarea cu App Engine se face prin protocolul CGI
 - Cand App Engine primeste o cerere pentru aplicatie, App Engine stabileste un runtime environment, cu datele cerute in variabilele de mediu
 - Identifica ce script *handler* va rula (foloseste fisierul de configurare)
 - Scriptul va genera un raspuns valid, afisand raspunsul

Obs. In exemplul dat “simulam” o implementare a protocolului CGI, rar intalnita in dezvoltarea de aplicatii

Google App Engine

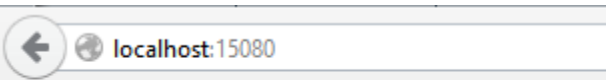


Crearea unei aplicatii in Python

- Framework-uri Python de lucru cu App Engine
 - Django, web2py, and Pylons
 - Implicit *webapp/ webapp2*

Generand
proiectul folosind
Launcher si
facand modificari
ca in figura, vom obtine
o noua versiune de
main.py

```
C:\gae\clock\main.py - Sublime Text 1.4 (UNREGISTERED)
File Edit Selection Find View Tools Project Preferences Help
app.yaml x main.py x mainCGI.py x
1 from google.appengine.ext import webapp
2 from google.appengine.ext.webapp.util import run_wsgi_app
3 import datetime
4
5 class MainPage(webapp.RequestHandler):
6     def get(self):
7         time = datetime.datetime.now()
8
9         self.response.headers['Content-Type'] = 'text/html'
10        self.response.out.write('<p>The time is: %s</p>' % str(
11            time))
12
13 application = webapp.WSGIApplication([('/', MainPage)],
14                                     debug=True)
15
16 def main():
17     run_wsgi_app(application)
18
19 if __name__ == '__main__':
20     main()
```



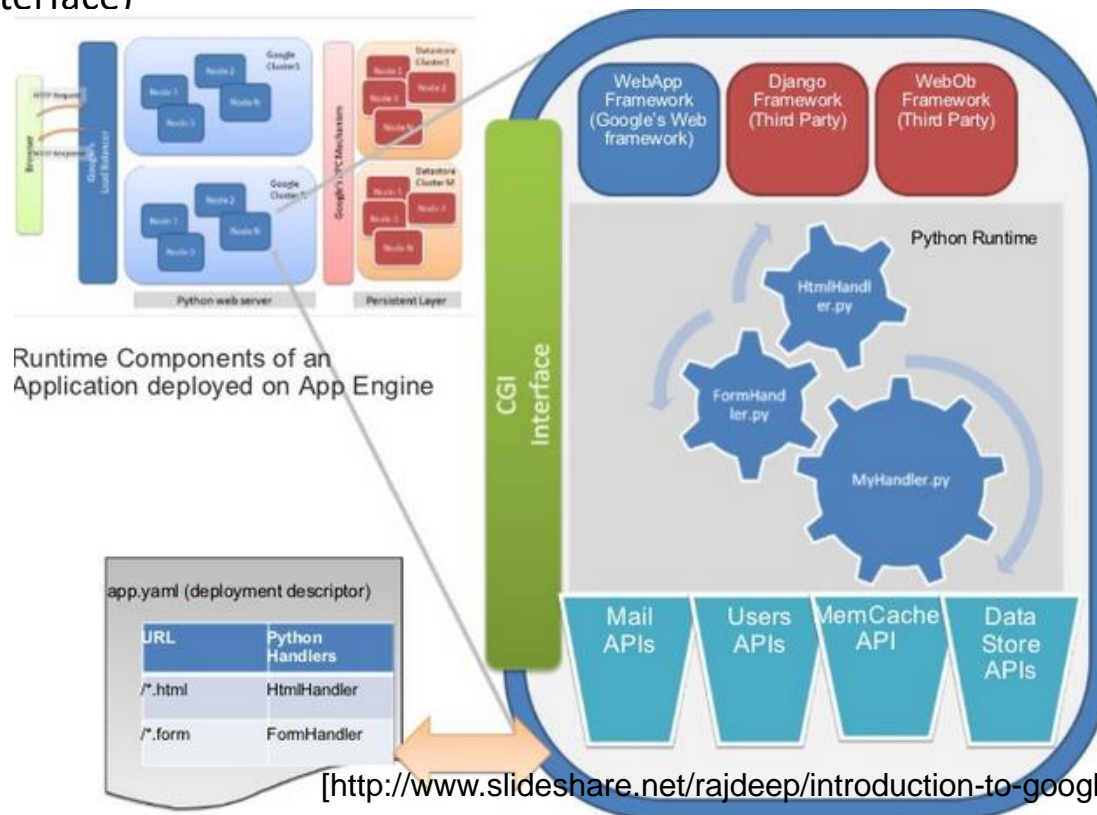
The time is: 2015-03-17 06:47:27.200000

Google App Engine



Crearea unei aplicatii in Python

- Cereri concurente si WSGI
 - In Python 2.7 apare suportul pentru aplicatii multithreading care pot face managementul cererilor concurente
 - => Se face trecerea de la un mediu CGI (Common Gateway Interface) la WSGI (Web Server Gateway Interface)



Google App Engine



Crearea unei aplicatii in Python

- In python 2.7 se permite rularea direct a aplicatiei (fara a se mai utiliza adapterul run_wsgi_app care ruleaza programul ca un script CGI)

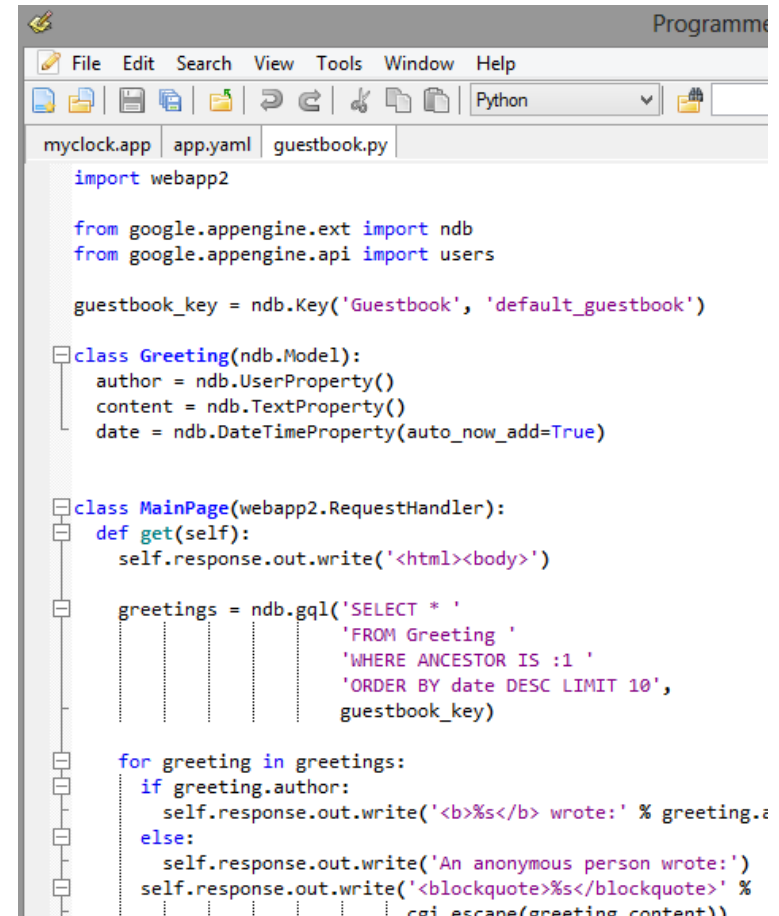
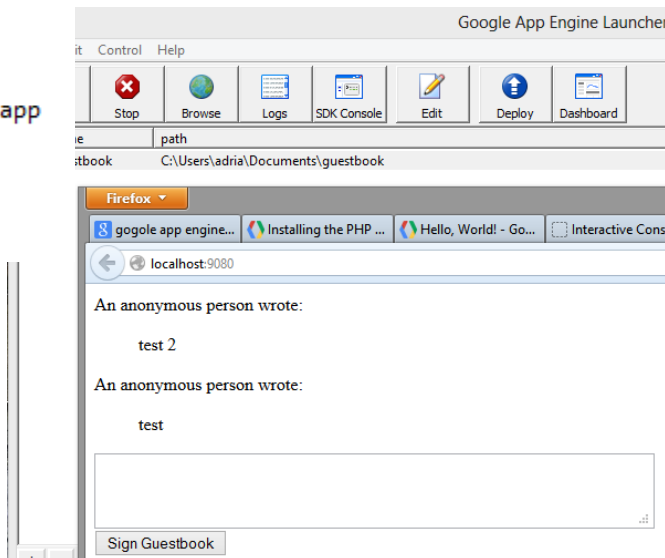
Exemplu de aplicatie folosind webapp2 (vezi arhiva cursului)

guestbook.py app.yaml

```
application: guestbook
version: 1
runtime: python27
api_version: 1
threadsafe: yes
```

```
handlers:
- url: .*
  script: guestbook.app
```

```
libraries:
- name: webapp2
  version: "2.5.2"
```



Google App Engine



Crearea unei aplicatii in Python

- Serverul Web Python si Java ofera o consola Web pentru inspectarea si debugging, in timp ce se face test pe masina locala

http://localhost:8080/_ah/admin

Google App Engine

Development SDK 1.9.18

dev~clockingae

Instances

Instances

Datastore Viewer

Datastore Indexes

Datastore Stats

Interactive Console

Memcache Viewer

Blobstore Viewer

Task Queues

Cron Jobs

XMPP

Inbound Mail

Full Text Search

	Latency (ms)	QPS	Total Requests	Logs	Runtime
default					python
500dfccf29b4ceb9ff1d8ee1b57ad368dd1c	0.0	0.00	0		

Google App Engine



Crearea unei aplicatii in Python

- **Consola Interactiva** permite crearea de cod Python si vizualizarea rezultatelor

The screenshot shows the Google App Engine console for the project 'dev~clockingae'. On the left is a sidebar with navigation links: Instances, Datastore Viewer, Datastore Indexes, Datastore Stats, Interactive Console (highlighted in red), Memcache Viewer, Blobstore Viewer, Task Queues, Cron Jobs, XMPP, Inbound Mail, and Full Text Search. The main area is titled 'Interactive Console' and contains a 'Restart Instance' button. Below the button is a text area with Python code:

```
import os
import pprint

from google.appengine.api import memcache
from google.appengine.api import mail
from google.appengine.api import urlfetch
from google.appengine.ext import db

pprint.pprint(os.environ.copy())
```

 Below the code is an 'Execute' button. At the bottom, the output of the execution is shown as a JSON object:

```
{'APPENGINE_RUNTIME': 'python27',
 'APPLICATION_ID': 'dev~clockingae',
 'AUTH_DOMAIN': 'gmail.com',
 'CURRENT_MODULE_ID': 'default',
 'CURRENT_VERSION_ID': '1.912452092156135148',
 'DATACENTER': 'us1',
 'DEFAULT_VERSION_HOSTNAME': 'localhost:15080',
 'INSTANCE_ID': '1561'}
```

Google App Engine



Crearea unei aplicatii in Python

- Inregistrarea Aplicatiei

- Crearea unui cont de dezvoltator
- Cloud Platform Console + Google Cloud SDK

App Engine Administration Console: <https://console.cloud.google.com>

Application ID – clockingae=> app.yaml

- Aplicatia

```
runtime: python27
api_version: 1
threadsafe: true
```

handlers:

```
- url: /*
  script: main.app
```

```
import webapp2
```

```
class MainPage(webapp2.RequestHandler):
```

```
    def get(self):
```

```
        self.response.headers['Content-Type'] = 'text/plain'
        self.response.write('Hello, World in 2017!!!')
```

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

Google App Engine



Crearea unei aplicatii in Python

- Testarea locala a Aplicatiei

`dev_appserver.py app.yaml`

The screenshot shows a Google Cloud SDK Shell terminal window with the command `dev_appserver.py app.yaml` executed. The terminal output includes several log messages: skipping the SDK update check, a warning about missing search indexes, and the starting of the API server, dispatcher, and admin server. It also shows two GET requests being handled by the default module. Below the terminal, a web browser window is open to `localhost:8080`, displaying the message "Hello, World in 2017!!!". The browser tabs include "localhost:8080", "Home - helloworld2017", and "Redirecting".

```
Google Cloud SDK Shell - dev_appserver.py app.yaml
D:\gae\helloworld2017>dev_appserver.py app.yaml
INFO     2017-03-31 23:59:29,315 devappserver2.py:764] Skipping SDK update check.
WARNING  2017-03-31 23:59:29,805 simple_search_stub.py:1146] Could not read search indexes f
rom c:\users\adria\appdata\local\temp\appengine.None\search_indexes
INFO     2017-03-31 23:59:29,809 api_server.py:268] Starting API server at: http://localhost
:58790
INFO     2017-03-31 23:59:29,813 dispatcher.py:199] Starting module "default" running at: ht
tp://localhost:8080
INFO     2017-03-31 23:59:29,819 admin_server.py:116] Starting admin server at: http://local
host:8000
INFO     2017-04-01 00:00:02,017 module.py:806] default: "GET / HTTP/1.1" 200 23
INFO     2017-04-01 00:00:02,046 module.py:806] default: "GET /favicon.ico HTTP/1.1" 404 154

localhost:8080 x Home - helloworld2017 x Redirecting x
Hello, World in 2017!!!
```

Google App Engine



Crearea unei aplicatii in Python

- Incarcarea Aplicatiei

gcloud app deploy sau gcloud init

```
D:\gae\helloworld2017>gcloud app deploy
You are about to deploy the following services:
- hellonode2017/default/20170401t000254 (from [D:\gae\helloworld2017\app.yaml])
  Deploying to URL: [https://hellonode2017.appspot.com]

Do you want to continue (Y/n)? n

ERROR: (gcloud.app.deploy) Aborted by user.

D:\gae\helloworld2017>gcloud init
Welcome! This command will take you through the configuration of gcloud.

Settings from your current configuration [default] are:
[compute]
region = us-central1
zone = us-central1-a
[core]
account = 1alboaie@gmail.com
disable_usage_reporting = False
project = hellonode2017

Your active configuration is: [default]

Pick configuration to use:
[1] Re-initialize this configuration [default] with new settings
[2] Create a new configuration
Please enter your numeric choice: 1
```

Google App Engine



Crearea unei aplicatii in Python

- Incarcarea Aplicatiei

gcloud app deploy sau **gcloud init**

```
D:\gae\helloworld2017>gcloud app deploy
You are about to deploy the following services:
- helloworld2017-163019/default/20170401t000829 (from [D:\gae\helloworld2017\app.yaml])
  Deploying to URL: [https://helloworld2017-163019.appspot.com]

Do you want to continue (Y/n)? y
```

```
Beginning deployment of service [default]...
Some files were skipped. Pass `--verbosity=info` to see which ones.
You may also view the gcloud log file, found at
[C:\Users\adria\AppData\Roaming\gcloud\logs\2017.04.01\00.08.27.676000.log].
```

```
= uploading 0 files to Google Cloud Storage
```

```
File upload done.
```


Google App Engine

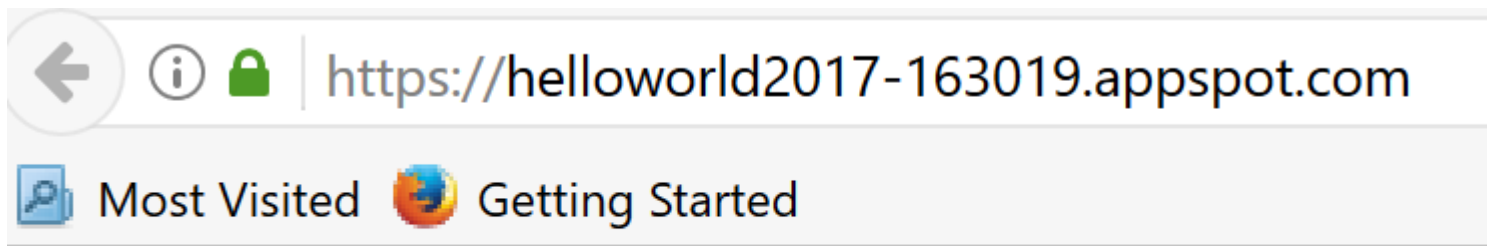


Crearea unei aplicatii in Python

- Incarcarea Aplicatiei

gcloud app deploy sau **gcloud init**

```
Updating service [default]...done.  
Deployed service [default] to [https://helloworld2017-163019.appspot.com]  
  
You can stream logs from the command line by running:  
$ gcloud app logs tail -s default  
  
To view your application in the web browser run:  
$ gcloud app browse  
  
D:\gae\helloworld2017>
```



Hello, World in 2017!!!

Google App Engine



Crearea unei aplicatii in Python | Google Launcher

- Inregistrarea Aplicatiei

- Crearea unui cont de dezvoltator
- Cloud Platform Console + Google Cloud SDK
- App Engine Administration Console: <https://console.cloud.google.com>



*Application ID – clockingae2017 => **app.yaml***

```
application: clockingae2017
version: 1
runtime: python27
api_version: 1
threadsafe: true
handlers:
- url: /favicon\.ico
  static_files: favicon.ico
  upload: favicon\.ico


- url: .*
  script: main.app
```

Google App Engine




Crearea unei aplicatii in Python | Google Launcher

- Codul Aplicatiei



```
application: clockingae2017
version: 1
runtime: python27
api_version: 1
threadsafe: true
handlers:
- url: /favicon\.ico
  static_files: favicon.ico
  upload: favicon\.ico
- url: .*
  script: main.app
```



```
import webapp2
import datetime

class MainHandler(webapp2.RequestHandler):
    def get(self):
        self.response.headers['Content-Type'] = 'text/plain'
        self.response.write(datetime.datetime.now())

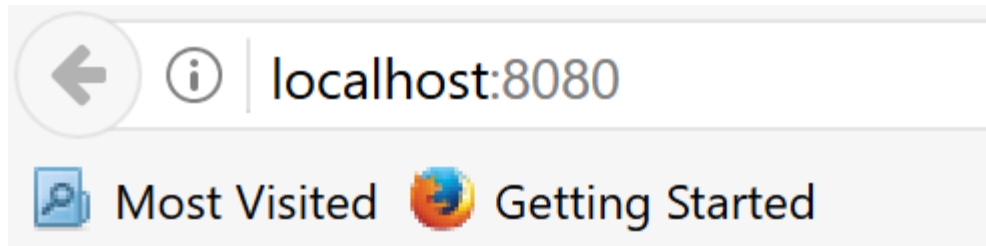
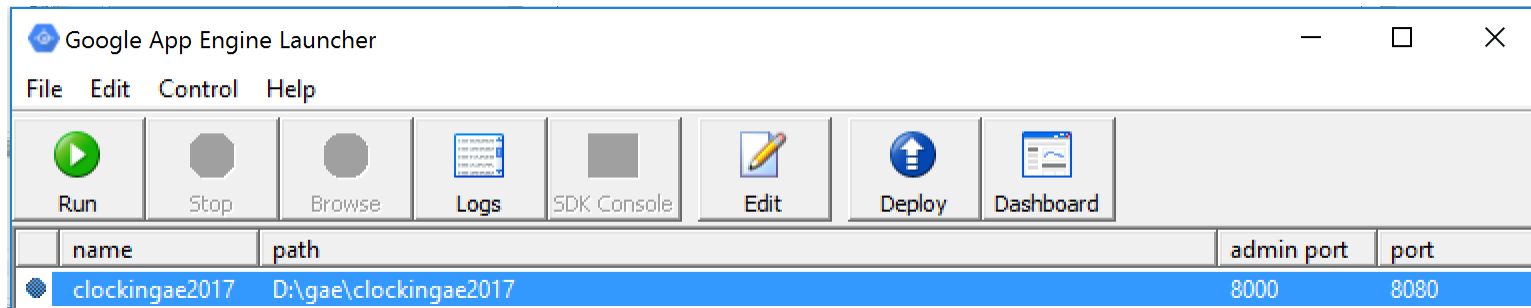
app = webapp2.WSGIApplication([
    ('/', MainHandler)
], debug=True)
```

Google App Engine



Crearea unei aplicatii in Python | Google Launcher

- Testarea locala a Aplicatiei



2017-04-01 04:29:02.519000

Google App Engine

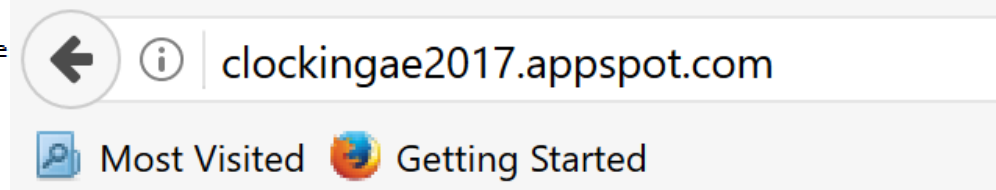


Crearea unei aplicatii in Python | Google Launcher

- Incarcarea Aplicatiei

```
2017-04-01 07:24:54 Running command: '['C:\\Python27\\pythonw.exe', '-u', 'C:\\Program
Files (x86)\\Google\\google_appengine\\appcfg.py', '--oauth2_credential_file=C:\\Users
\\adria\\.appcfg_oauth2_tokens', 'update', u'D:\\gae\\clockingae2017']"
07:24 AM Application: clockingae2017; version: 1
07:24 AM Host: appengine.google.com
07:24 AM Starting update of app: clockingae2017, version: 1
07:24 AM Getting current resource limits.
07:24 AM Scanning files on local disk.
07:25 AM Cloning 1 static file.
07:25 AM Cloning 4 application files.
07:25 AM Uploading 2 files and blobs.
07:25 AM Uploaded 2 files and blobs.
07:25 AM Compilation starting.
07:25 AM Compilation completed.
07:25 AM Starting deployment.
07:25 AM Checking if deployment succeeded.
07:25 AM Deployment successful.
07:25 AM Checking if updated app version is serving.
07:25 AM Completed update of app: clockingae2017, version: 1
07:25 AM Uploading index definitions.
2017-04-01 07:25:22 (Process exited with code
```

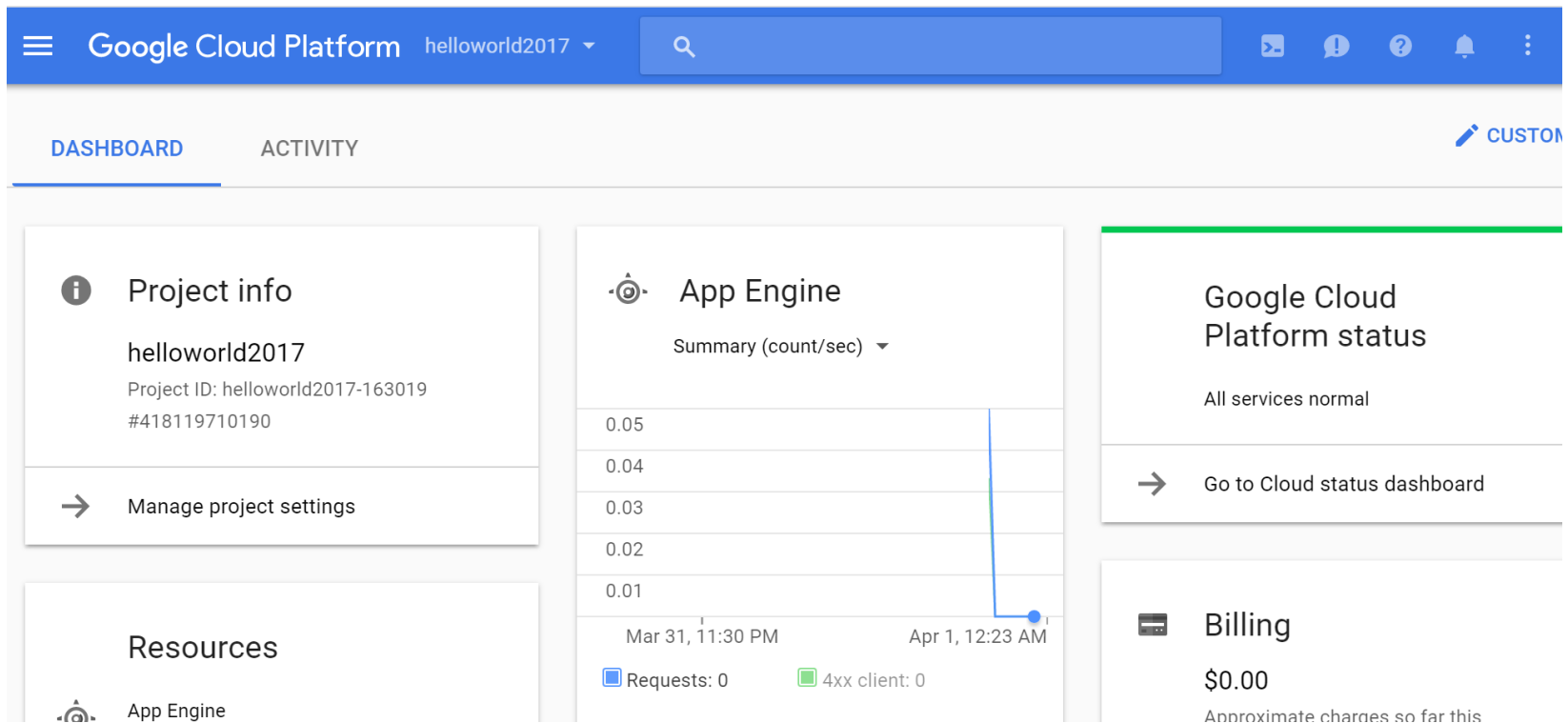
You can close this window now.



2017-04-01 04:25:47.801630

Google App Engine

Consola de administrare | **Google Developers Console**



Google App Engine

Consola de administrare | Admin Console

The screenshot shows the Google Cloud Platform Admin Console interface. The top navigation bar is blue and contains the Google Cloud Platform logo, the account name 'helloworld2017', a search bar, and icons for notifications, help, and user profile. The left sidebar is light gray and contains a menu with the following items: 'Stackdriver Trace' (selected), 'Overview', 'Trace List' (highlighted in blue), and 'Analysis Reports'. The main content area is white and displays the 'Trace List' page. It features a header with 'Trace List', a '+', 'UNDO ZOOM', 'AUTO RELOAD' button, and time range filters: '1 hour', '4 hours', '12 hours', '1 day' (selected), '3 days', and '1 week'. Below the header, there are filters for 'Request URI' (set to '+/'), 'HTTP method' (set to 'GET'), 'Service' (set to 'All services'), and 'Version' (set to 'All'). A vertical list of HTTP status codes is visible on the right side of the main content area, including 400, 401, 403, 404, 410, 5XX, and 500. The message 'There are no matched traces' is displayed in the center of the main content area.

Google Cloud Platform helloworld2017

Stackdriver Trace

Overview

Trace List

Analysis Reports

Trace List + UNDO ZOOM AUTO RELOAD 1 hour 4 hours 12 hours 1 day 3 days 1 week

Request URI HTTP method

+/ GET

Service Version

All services All

400 401 403 404 410 5XX 500

There are no matched traces

Google App Engine

Consola de administrare | Admin Console

Stackdriver
Trace

Overview

Trace List

Analysis Reports

Analysis Reports

New report request

Request URI

☐ Only analyze requests that result in remote procedure calls ?

HTTP method HTTP status

Report name (optional)

Service	Version	Time range	Start	End
<input type="radio"/> <input type="text" value="All services"/>	<input type="text" value="All versions"/>	<input type="text" value="Custom"/>	<input type="text" value="Mar 31, 2017, 11:22:01 PM EEST"/>	<input type="text" value="Apr 1, 2017, 12:22:01 AM EEST"/>

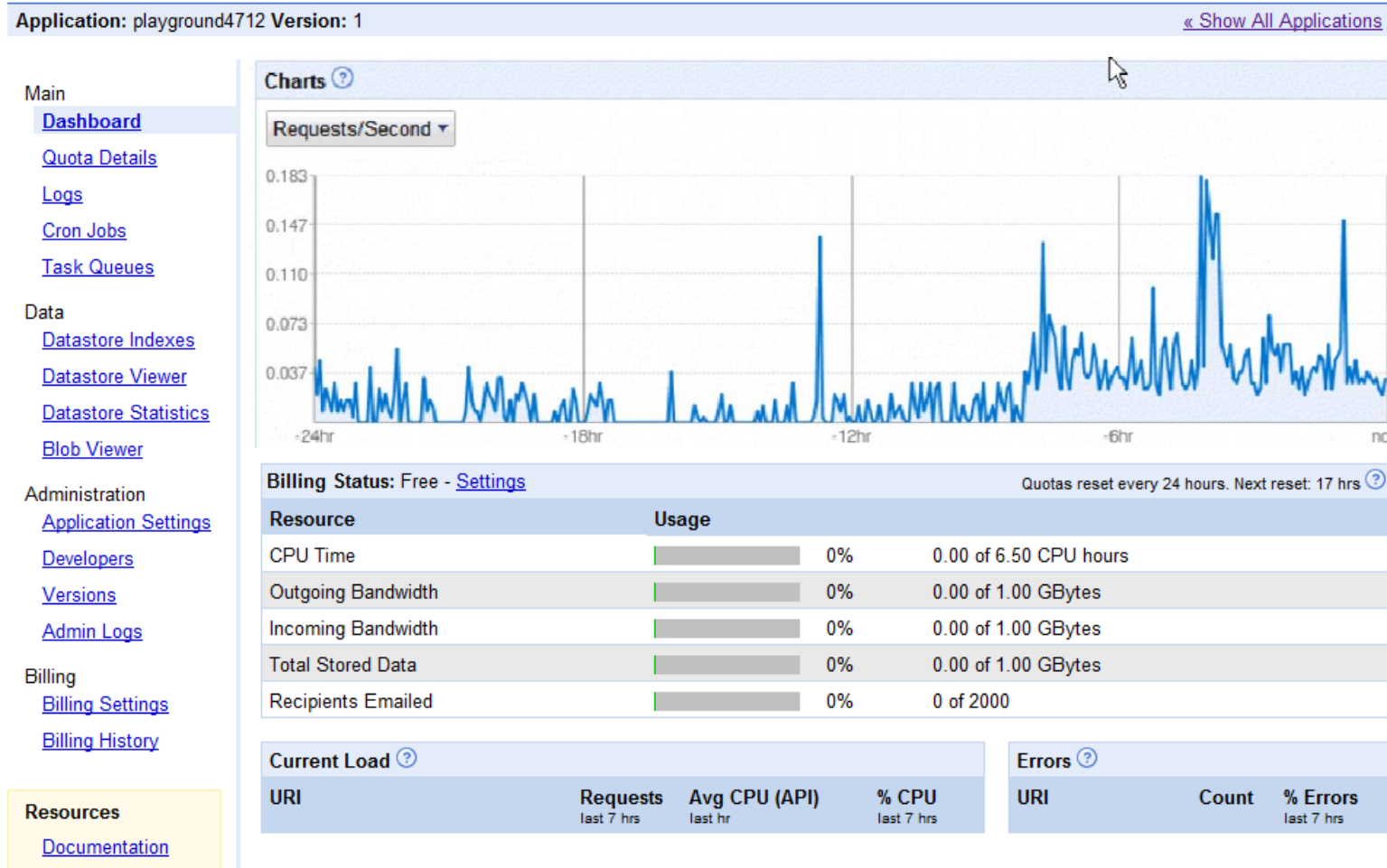
☒ Create comparison

Service	Version	Time range	Start	End
<input type="radio"/> <input type="text" value="All services"/>	<input type="text" value="All versions"/>	<input type="text" value="Custom"/>	<input type="text" value="Mar 24, 2017, 11:22:01 PM EET"/>	<input type="text" value="Mar 25, 2017, 12:22:01 AM EET"/>

Google App Engine



Consola de administrare ☺



Google App Engine



Crearea unei aplicatii in Php

- <http://helloworldphpcc.appspot.com/>

app.yaml | helloworld.php

```
application: helloworldphpcc
version: 1
runtime: php
api_version: 1

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

File Edit Options Encoding

```
k?php
echo 'hello world:~';
?>
```

Deployment To Google (helloworldphpcc)

```
Starting update of app: helloworldphpcc, version: 1
11:04 AM Getting current resource limits.
Password for lalboaie@gmail.com: 11:04 AM Scanning files on local disk.
11:04 AM Cloning 2 application files.
11:04 AM Uploading 2 files and blobs.
11:04 AM Uploaded 2 files and blobs
11:04 AM Compilation starting.
11:04 AM Compilation completed.
11:04 AM Starting deployment.
11:04 AM Checking if deployment succeeded.
11:04 AM Deployment successful.
11:04 AM Checking if updated app version is serving.
11:04 AM Completed update of app: helloworldphpcc, version: 1
2014-03-18 11:04:30 (Process exited with code 0)

You can close this window now.
```

Edit Deploy Dashboard

File Edit Control Help

Run Stop Browse Logs SDK Console

name	path
guestbook	C:\Users\adria\Documents\guestbook
helloworldphpcc	C:\gaephp\helloworld

Firefox

< r... The Python ... Migrating t... Interactive ...

hello world:~)

Bibliografie

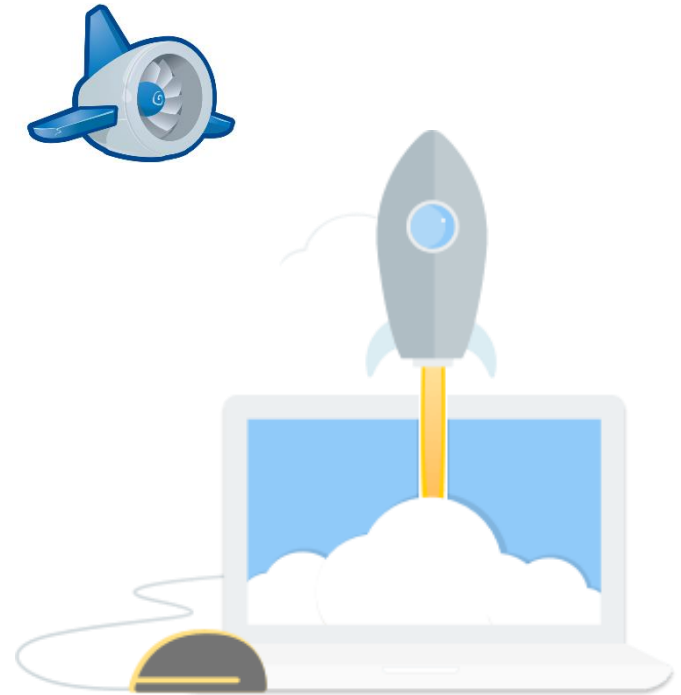


- <https://cloud.google.com/>
- <https://cloud.google.com/appengine/docs/flexible/nodejs/using-nodejs-libraries>
- <http://www.python.org/download/>
- <https://www.martinfowler.com/articles/microservices.html>
- ... 😊😊😊

Sumar

Google App Engine

- Standard Environment
- Flexible Environment



Universitatea “Alexandru Ioan Cuza”
Facultatea de Informatică

Întrebări?

