

DOCUMENTACIÓN

PROYECTO GO ANALYTICS

14 de junio de 2016

Ivan Leonardo Galeano Saavedra
Networks Lab.
Bogotá, Colombia
ilgaleanos@gmail.com

Índice general

1. Motivaciones, lenguajes y otros.	5
1.1. Lenguajes.	6
1.2. Arquitectura.	6

Capítulo 1

Motivaciones, lenguajes y otros.

Introducción

El presente texto es un intento de exponer de la manera mas clara, sencilla y concreta el código referente a Go Analytics, también llamado talenter o gogo para el equipo de desarrollo. El desarrollo esta realizado casi en su totalidad en código pyhton en su versión 2.7.X y django 1.8.12+; algunas partes como veremos están escritas en código C++ compilado para python. De la misma manera de la parte del usuario se uso javascript con jquery y librerías estándar para diversas funcionalidades.

Esta es la segunda y actual versión del código ya que inicialmente se tenia uno más sencillo, pero funcional, el cual por innumerables cambios de requerimientos se hizo inmanejable. Tratando de mantener una filosofía mas modular se elaboró pensando en un gogo sin versión 360. Esta versión 360 se agregó como módulos adicionales tratando de mantener el código que se había escrito sin modificaciones en miras a la estabilidad.

Inicialmente se expondrá la arquitectura actual, versiones, requerimientos y demás paquetería necesaria. Se dedicará un capítulo para cada carpeta de código (app de django) en la cual se expondrá la necesidad, funcionalidad y arquitectura de cada app.

Nota personal: Espero que el desarrollador que trate con el código sea muy comprensivo a la hora de examinarlo ya que el fue desarrollado sin unos requerimientos detallados, sino con cada nueva versión, se solicitaban nuevas modificaciones sobre lo ya elaborado lo que ha conllevado en una no clara consecución de un código completamente estable.

1.1. Lenguajes.

El uso de este manual requiere conocimiento en Django 1.8, Python 2.7, C++11 El código ha sido escrito en lenguaje Python y C++

```
Python 2.7.6 (default, Jun 22 2015, 17:58:13)
[GCC 4.8.2] on linux2
```

para el framework Django

Django 1.8.13

Javascript con JQuery 2.x.

1.2. Arquitectura.

En AWS poseemos 3 servicios:

1. EC2: 2 servidores, servidor nano para un servicio prestado a Slam Security (no hablaremos de ese proyecto en el manual) y servidor micro T2 con sistema operativo Ubuntu 14.04. 30 gb de almacenamiento y 1 Gb de memoria RAM.
2. RDS: 1 Servidor T2 nano de base de datos en Postgresql de 5 Gb de almacenamiento y 1 Gb de memoria RAM, este servidor se inició por las previsiones de ventas del área comercial.
3. SES: 1 conexión smtp de alto envío de correos.

Las conexiones las realicé de la siguiente manera. La aplicación corre en un servidor apache

```
Server version: Apache/2.4.7 (Ubuntu)
Server built:   Jan 14 2016 17:45:23
```

para el cual los certificados ssl los generé por medio de Let's Encrypt <https://letsencrypt.org/getting-started/>. estas claves ssl no estoy muy seguro de trasladarlas de servidor por el modo de generación, es por ello que no están incluidas y es altamente recomendable leer la documentación de este método de obtención y mantenimiento de claves ssl.

La aplicación se conecta localmente a una instancia de pgbouncer (*esto con la finalidad de mantener siempre una conexión abierta y evitar el overhead de estar abriendo conexiones*)

```
pgbouncer version 1.5.4
(compiled by <buldd@roseapple> at 2013-06-18 17:16:50)
```

La clave de la conexión local con pgbouncer es

```
'USER': 'usuariodb_gogo',
'PASSWORD': 'W#y2d@uV4+eSPuwrEc$UTrE4eCruTHas',
```

la clave de conexión de pgbouncer a la base de datos es

```
user=usuariodb_gogo  
password=W#y2d@uV4+eSPuwrEc$UTrE4eCruTHas
```

El archivo de configuración apache2.conf, los sites-availables, gogo pgbouncer.ini al igual que el userlist.txt se encuentran en ./gogo/gogo/configs/

La base de datos contiene 3 bases de datos una llamada gogo perteneciente a esta aplicación, otra llamada talenter que no es mas que una copia incompleta de la primera y una tercera perteneciente a slam. Se cuentan con tres Group Roles y tres Login Roles uno para cada aplicación y un rol maestro (no superusuario, RDS no lo proporciona) :

```
user=suidi  
password=Networks123*
```

Las claves de administración de dominios están bajo Juan Sebastián Henao P. y Ricardo Montoya Meneses.