GOOGLE APP ENGINE(GAE)

App Engine es una plataforma sin servidor totalmente administrada para desarrollar y alojar aplicaciones web a escala. Puedes elegir entre varios lenguajes, bibliotecas y marcos populares para desarrollar tus aplicaciones y luego dejar que App Engine se encargue de aprovisionar servidores y escalar las instancias de tu aplicación según la demanda.

* Por la complejidad:
* Balanceadores webs. \* Sistemas de archivos. \* Tiempos

GAE:

* Fácil de usar, sin conocimientos especializados en para montar la infraestructura.
* Enfocarse en la aplicación.
* Escalable
* Se paga por lo que se usa.  
  \*Aplicaciones corriendo en la misma infraestructura

GAE Standard:

* Funciona bajo el esquema de instancias con runtimes específicos. Python, PHP, y Go
* Auto escalamiento: No hay necesidades de balanceadores de carga.  
  \*Cuota gratuita
* Precio basado en uso

Restricciones:

No puedes haces escrituras sobre filesystem  
No puede tardar más de 60s en peticiones

Usarlo:  
¨\*Cuando se va a desarrollar desde 0  
\*Mayor economía posible  
\*Picos de trafico

GAE Flexible:

* Runtime personalizados
* $ basado en instancias
* Acceso a SSH (Protocolos de accesos remotas a servidores)  
  \*Escritura en disco  
  \*uso de librerías de terceros  
  \*despliegue sobre Docker (Open source para automatización de despliegue de código)  
  \*Se deben instalar librerías de App engine

¿Cuáles son los servicios que ofrece GAE?

1. Servicios, arquitectura orientada a microservicios.
2. Versiones, varias versiones de tu aplicación que puedes usar como quieras.
3. Máquinas virtuales.
4. Memcache, base de datos volátil de tipo llave, valor. Puede acceder mediante APIs.
5. Mail
6. Blobstore, para almacenamiento de elementos de gran tamaño.
7. El cloud shell es una línea de comandos basada en la web. Una máquina virtual que se prende para nuestro uso y que existe en tiempo de ejecución.
8. Trabajaremos principalmente con SDK porque vamos a trabajar de manera local en nuestro computador.
9. Vamos a hacer nuestro primer hola mundo, para esto es muy importante contar con una cuenta de GCP. Para empezar a usar los recursos de la nube de Google necesitamos activar el Billings, es decir una forma de pago, pero ten en cuenta que Google te da tus primeros 300usd o un año de uso de la plataforma.
10. Puedes entrar con tu cuenta de Gmail.
11. Este es el listado de comandos usados en este módulo:
12. gcloud auth login --> *Realizar login* en tu proyecto de GCP desde una línea d\_e comandos\_  
    gcloud config set project "projectID" --> *Configurar tu sesión a un proyecto en específico*  
    git clone https://github.com/xertica-cloud/helloworld\_in\_appengine.git *Clonar el repositorio del “Hola Mundo en App Engine”*  
    cd helloworld\_in\_appengine --> *Entrar al directorio del proyecto*  
    gcloud app deploy --> *Realizar el despliegue del hola mundo al proyecto previamente configurado*

Realizar el hola mundo en sus equipos, haciendo uso del Cloud SDK

Para esta clase vamos a estar trabajando con Python 2.7, webapp2 y AppEngineStandar.

Este es el listado de comandos usados en este modulo:

gcloud auth login --> *Realizar login* en tu proyecto de GCP desde una línea d\_e comandos\_  
gcloud config set project "projectID" --> *Configurar tu sesión a un proyecto en específico*  
git clone https://github.com/xertica-cloud/helloworld\_in\_appengine.git *Clonar el repositorio del “Hola Mundo en App Engine”*  
cd helloworld\_in\_appengine --> *Entrar al directorio del proyecto*  
gcloud app deploy --> *Realizar el despliegue del hola mundo al proyecto previamente configurado*

Dominios personalizados

. Reemplazar el dominio incluye por default AppEngine(.appspot.com) por uno propio(midominio.com)

Dominios. APP

.Mas seguridad

.Obtener un dominio más memorable

.Una manera mas rápida de que nuestro sitio sea encontrado

**Llave privada**

*sudo openssl rsa -inform pem -in /etc/lrtsencrypt/live/nombre.app/privkey.pem -outform pem | less*

**Llave pública**

*sudo less /etc/letsencrypt/live/nombre.app/fullchain.pem*

**Google Data Store**, es una base de datos NoSQL desarrollada por Google.

Diseñada para Backend. Nos da acceso mediante APIs. NoSQL. Escalable y confiable, orientada a la redundancia y soporta transacciones ACID. Orientada a la redundancia.

, Entidad- se asocia a un objeto, (un registro en base de datos)

. Kind – se asocia con tabla- categoría de un objeto.

. Propiedad- Un campo- Un dato individual para un objeto

. Key- ID único

NDB es una librería que nos da Google para trabajar en Python

GOOGLE CLOUD STORAGE

* Es un nuevo esquema de almacenamiento en la nube, robusto, escalable y económico. Si has trabajado con Google Drive.
* Es un sistema de almacenamiento de archivos en la nube. Cuando requieres manejar el tema de almacenamiento de archivos, backups.
* Puedes elegir tu esquema de pagos.

Base de datos no estructurado- pueden manejar versiones, permisos-

Alto desempeño. Acceso desde API, no es un sistema de archivos

**TIPOS DE INSTANCIAS**

**Tipo residente:** Corren todo el tiempo y no se apagan para garantizar el rendimiento de la  
aplicación.

Tendrás siempre el mejor rendimiento, pero los costos son más altos.

**Tipo dinámico:** Mas elástica, se prenden y apagan máquinas de acuerdo con la necesidad  
del negocio.

El prender y apagar las maquinas te ayuda a reducir los costos, pero puede afectar el  
performance del sistema.

**Escalamiento de Instancias**

* Escalamiento Manual.
* Escalamiento básico: usa instancias dinámicas.
* Escalamiento automático, usa ambos tipos de instancias, le puedes configurar n instancias  
  de tipo residente, pero si tu aplicación requiere más demanda puede crecer de manera  
  dinámica. Estas máquinas se van a apagar hasta el número mínimo n que definiste.

**Tipos de colas de tareas:**

Push --> Tareas que trabajan para recibir inputs. Tienen un límite de 10 min para la. Garantizan la ejecución de la tarea.

Pull --> Reaccionan cuando otra actividad se ejecuta

TAREAS PRGRAMADAS (CRON)

Es un servicio que nos permite ejecutar tareas cada cierto tiempo. Son tareas que se pueden ejecutar automáticamente.

Se puede agendar la ejecución en una fecha o intervalos específicos.

. Tareas lanzadas automáticamente por GAE

. Se puede agendar la ejecución en una fecha específica o en intervalos regulares.

. Son invocadas mediante una petición GET HTTP

. Tienen un tiempo límite de 10 min para su ejecución.

Cómo escribir un archivo cron:

. Archivo que contiene el agendamiento.

. Debe estar en la raíz del proyecto (como el app.yaml)

. La URLS de cada tarea deben ser direcciones válidas dentro de la aplicación.

FORMATO AGENDAMIENTO

. Se especifican mediante el uso de un formato basado en ingles simple

. Se pueden definir un intervalo específico de tiempo.

Google Cloud Endpoints

. Herramientas de backend para genera APIS RESTfull

.Generación dinámica de cliente en JavaScript, IOS o Android

. Autenticación, serialización de datos. API Explorer.

. Altamente escalable.

CÓMO FUNCIONA?

* 1. Escribes código del backend.
  2. Colocas anotaciones o decoradores.
  3. Generas librerías de clientes.
  4. Escribes el código del cliente.

**App Engine** nos ofrece varios servicios para **garantizar la seguridad** en tus aplicaciones.

1. Google Cloud Security Scanner. Realiza un rastreo a lo largo de toda la aplicación,  
   identificando posibles vulnerabilidades. Ejecuta todas las posibles entradas que pueda  
   tener un usuario en tu aplicación.

* XSS Cross site scripting. Inserción de código JS.
* Flash injection.
* Contenido mixto (HTTP o HTTPS).
* Borrar contraseñas de texto
* Uso de librerías de JS que sean inseguras u obsoletas.

\*\*¿Realizar un escaneo seguro? \*\*

1. Corre el escaneo en un entorno de pruebas.
2. Usa una cuenta de pruebas
3. Bloquea elementos individuales de UI.
4. Realiza respaldo de tus datos
5. Excluye URLs

Reglas de Firewall  
● Permite o restringe tráfico únicamente de redes específicas  
Permitir únicamente el tráfico de servicios específicos  
● Bloquear direcciones IP abusivas

Identy Access Proxy: se utiliza en otros productos de Google,

.Trabaja a nivel de aplicación.

. Utiliza el servicio de autenticación, servicios de API

. Administra el acceso a tus aplicaciones y recursos.

. Utiliza un control de acceso a nivel de apliación en lugar de definir reglas de firewall

CLOUDE SOURCE REPOSITORIES

Almacenamiento confiable.

Interfaz amigable.

Integración con github y big bucket.

STackdriver Debugger

Debug en linea con la herramienta Stackdriver debuger.

**¿Qué es un bug?** Lo asociamos con errores o interrupciones en nuestras aplicaciones. Siempre hay una probabilidad de que las aplicaciones tengan bugs.

Hacer la detección de lo que esta ocurriendo y el punto en el que pasa se puede hacer aprovechando diferentes herramientas y configuraciones de los entornos de trabajo. Por ejemplo podemos tener ambientes de pruebas y producción similares.

Google desarrolló StackDriver Debugger es una serie de herramientas que nos permiten hacer la revisión y analisis de código en la nube, en producción.

-Permite verificar estado de la aplicación.

-Puedes tomar capturas

-Se integra con tu IDE.

-Es de configuración sencilla.

-Se integra con Google Cloud Source repositories, GIT HUB o BITbucket

ERROR REPORTING

* Servicio que nos permite analizar la cantidad de errores que tenemos en tiempo real. Hace agrupamiento de errores.
* Permite filtrar de manera sencilla.
* Muestra detalles específicos. Facilita el seguimiento a errores.

LOGGIN

Permite almacenar y analizar los datos y eventos de los registros de GCP y AWS.

Genera alertas.

Permite hacer búsquedas detalladas.

Almacena registros por hasta 90 días.

Se integra con los servicios de Google Cloud Platform.

TRACE

Es un sistema de rastreo que te permite revisar la latencia de tus aplicaciones. Puedes hacer seguimiento de como se están propagando las solicitudes. Permite detectar cuellos de botella. Detecta problemas. Hace evaluación de rendimiento