

# Google Compute Engine – IaaS

---

Antonio Manuel Fernández Cantos

# ¿Qué es IaaS?

---

- Infraestructura como servicio.
- Acceso a recursos informáticos en un entorno virtualizado (la nube), a través de la red.
- El proveedor se encarga del mantenimiento físico.

# Ventajas

---

- Escalabilidad: recursos disponibles en el momento que el cliente los solicita. Desaparecen los tiempos de espera a la hora de ampliar la capacidad.
- No necesitamos preocuparnos del hardware.
- Pago por los recursos que se utilizan.
- Podemos acceder al servicio desde cualquier lugar, siempre que tengamos internet.

# Despliegue

# Herramientas

---

- **Vagrant:** Permite generar entornos reproducibles de forma sencilla.
- **Vagrant-google:** es un plugin que permite desplegar la máquina virtual en el IaaS.
- **Pip:** Necesario para instalar Fabric y Ansible.
- **Ansible:** automatiza tareas dentro de la máquina virtual.
- **Fabric:** Permite ejecutar tareas dentro del IaaS invocadas desde fuera de ella.
- **Clave ssh:** necesario para que la instancia nos pueda identificar.

# Credenciales

---

1. Introducir la clave pública en el panel de administración de nuestra máquina (desde la página web).
2. Obtener el **ID** del proyecto.
3. Client email (acabado en gserviceaccount.com).
4. La clave de la cuenta del servicio en formato JSON.

# Bot de Telegram

---

Uso de un bot de Telegram para comprobar que el IaaS recibe peticiones y responde a los usuarios que utilicen el bot.

Pasos para obtener el bot:

1. Buscamos BotFather en Telegram.
2. /start para iniciar conversación.
3. /newbot para obtener un nuevo bot.
4. Introducimos el nombre del bot **PaísesYCapitales**.
5. Nombre de usuario del bot: **paisesycapitalesbot**.
6. Guardamos el token que nos da BotFather para poder recibir y responder a las peticiones.



Antonio

/start

✓✓ 16:53



BotFather

I can help you create and manage Telegram bots. If you're new to the Bot API, please [see the manual](#).

16:53

You can control me by sending these commands:

[/newbot](#) - create a new bot

[/mybots](#) - edit your bots [**beta**]

[/mygames](#) - edit your [games](#) [**beta**]

#### Edit Bots

[/setname](#) - change a bot's name

[/setdescription](#) - change bot description

[/setabouttext](#) - change bot about info

[/setuserpic](#) - change bot profile photo

[/setcommands](#) - change the list of commands

[/deletebot](#) - delete a bot

#### Bot Settings

[/token](#) - generate authorization token

[/revoke](#) - revoke bot access token

[/setinline](#) - toggle [inline mode](#)

[/setinlinegeo](#) - toggle inline [location requests](#)

[/setinlinefeedback](#) - change [inline feedback](#) settings

[/setjoininggroups](#) - can your bot be added to groups?

[/setprivacy](#) - toggle [privacy mode](#) in groups

#### Games

[/newgame](#) - create a new [game](#)

[/listgames](#) - get a list of your games

[/editgame](#) - edit a game

[/deletegame](#) - delete an existing game



Antonio

[/newbot](#)

✓✓ 16:54



BotFather

Alright, a new bot. How are we going to call it? Please choose a name for your bot.

16:54



Antonio

PaísesYCapitales

✓✓ 16:55



BotFather

Good. Now let's choose a username for your bot. It must end in ``bot``. Like this, for example: TetrisBot or tetris\_bot.

16:55



Antonio

paísesycapitalesbot

✓✓ 16:56



BotFather

Done! Congratulations on your new bot. You will find it at [t.me/paisesycapitalesbot](https://t.me/paisesycapitalesbot). You can now add a description, about section and profile picture for your bot, see [/help](#) for a list of commands. By the way, when you've finished creating your cool bot, ping our Bot Support if you want a better username for it. Just make sure the bot is fully operational before you do this.

16:56

Use this token to access the HTTP API:



# Exportar variables

---

Necesitamos almacenar las credenciales obtenidas anteriormente en variables de entorno. Es importante que no estén presentes en ningún archivo por la seguridad de nuestro IaaS.

- `export PROJECT_ID=Nuestra_id_del_proyecto`
- `export CLIENT_EMAIL=Email_obtenido`
- `export KEY_LOCATION=Ruta_archivo_json`
- `export TOKENBOT=Token_bot`

# Vagrant

---

Creamos un archivo Vagrantfile para el despliegue de la máquina en el IaaS.

En el indicamos:

- Las variables de entorno para la identificación de nuestra cuenta.
- Donde se encuentra nuestra clave privada ssh para poder acceder a la instancia.
- El sistema operativo, la zona del IaaS (europa-west1) y el tamaño de la máquina.

# Ansible y Fabric

---

Ansible: Sirve para provisionar la máquina sin necesidad de acceder a ella explícitamente. Simula el `sudo apt install -y ...`. Para el despliegue he indicado que instale:

- Python
- Pip y python-telegram-bot
- Git
- Supervisor

Fabric: Permite ejecutar órdenes de forma remota dentro de la instancia.

# Makefile

---

```
desplegar:
    vagrant up --provider=google

provisionar:
    vagrant provision

clonar:
    fab -H Antkk10@130.211.83.45 -i /Users/antoniomfc90/.ssh/key_gce Clonar

ejecutar:
    fab -H Antkk10@130.211.83.45 -i /Users/antoniomfc90/.ssh/key_gce EjecutarApp

stop:
    fab -H Antkk10@130.211.83.45 -i /Users/antoniomfc90/.ssh/key_gce StopApp

destruir:
    vagrant destroy -f

borrar:
    fab -H Antkk10@130.211.83.45 -i /Users/antoniomfc90/.ssh/key_gce borrar
```

# Despliegue

---

Orden:

make desplegar

make provisionar

make clonar

make ejecutar

Equivale:

vagrant up --provider=google

vagrant provision

ab -H Antkk10@130.211.83.45 -i  
/Users/antoniomfc90/.ssh/key\_gce Clonar

fab -H Antkk10@130.211.83.45 -i  
/Users/antoniomfc90/.ssh/key\_gce EjecutarApp

Gracias por vuestra atención.  
¿Preguntas?