

Despliegue de múltiples dispositivos con BalenaCloud

Ernesto Díaz Almansa
15 de Julio de 2021

Indice

1. Objetivo.....	3
2. Preparación.....	3
2.1 Que es Docker??.....	4
2.1.1 Docker vs Máquinas Virtuales.....	4
2.1.2 Imágenes y Contenedores.....	5
3. Software a utilizar.....	5
3.1 git.....	5
3.1.1 Entendiendo cómo funciona GIT.....	6
3.2 balena-etcher.....	6
3.3 Claves ssh.....	6
4. Preparación de la aplicación de la nube.....	7
4.1 balena-cloud.....	7
4.2 Aplicación.....	8
5. Preparación de raspberrypi 3B+.....	10
5.1 Balena-etcher.....	10
5.2 Poner datos de ip estática en SO balenaOS.....	11
5.3 Puesta en marcha de balenaOS.....	11
6. Comunicación con balenacloud.....	12
6.1 Preparación de circuito.....	12
6.2 Envío de archivos.....	14
6.3 Añadir segundo dispositivo a una misma aplicación.....	17

1. Objetivo

El objetivo del proyecto que mediante una aplicación en la nube se pueda controlar y configurar el despliegue de varios dispositivos para IoT que esten o no esten en la misma red y podamos lanzar servicios o micro servicios mediante contenedores con docker y repositorios git.

La opción escogida ha sido balena-cloud, una plataforma que nos va a permitir esto en el modo free que són hasta 10 dispositivos.

En el proyecto sólo se hará una aplicación, pero se puede uno imaginar el despliegue de varias aplicaciones y poder hacer varias cosas y interactuar entre las aplicaciones.

2. Preparación

Proyecto basado en la memoria de 0. Vazquez del 19 de junio 2019. He cambiado algunos aspectos de la memoria.

En la memoria se ha trabajado con W10, pero viendo todo el software adicional a instalar he optado por la opción de trabajar con una distribución linux.

Hemos trabajado sobre un sistema operativo debian 10 (debian buster)

```
uname -a
```

```
1 SMP Debian 4.19.194-2 (2021-06-21) x86_64 GNU/Linux
```

También con dos raspberrypi3+ con los sistemas operativos balena. Se descargan directamente cuando se añade un dispositivo a una aplicación. Son sistemas operativos preparados con bastante seguridad, son SO de solo lectura. Con la seguridad que conlleva eso. Hay maneras de acceder a el mediante ssh tanto como al host como a main (el contenedor docker que hemos lanzado).

Al host hace puente a main

```
ssh -p 22222:4321 root@ip
```

Si solo se quiere al host

```
ssh -p 22222 root@ip
```

Unas de las maneras que se podria modificar el host seria accediendo físicamente a la tarjeta, así con cualquier software de gestión de partición una vez desmontado se podría poner de escritura la partición cambiar ro por rw /dev/root, pero lo dicho hay que tener acceso físico

2.1 Que es Docker??

Docker es una herramienta open-source que nos permite realizar una 'virtualización ligera', con la que poder empaquetar entornos y aplicaciones que posteriormente podremos desplegar en cualquier sistema que disponga de esta tecnología.

Para ello Docker extiende LXC (Linux Containers), que es un sistema de virtualización que permite crear múltiples sistemas totalmente aislados entre sí, sobre la misma máquina o sistema anfitrión.

2.1.1 Docker vs Máquinas Virtuales

La gran diferencia es que una máquina virtual necesita contener todo el sistema operativo mientras que un contenedor Docker aprovecha el sistema operativo sobre el cual se ejecuta, comparte el kernel del sistema operativo anfitrión e incluso parte de sus bibliotecas.

En la siguiente imagen podemos ver esta diferencia entre el enfoque de las máquinas virtuales y el utilizado por Docker:

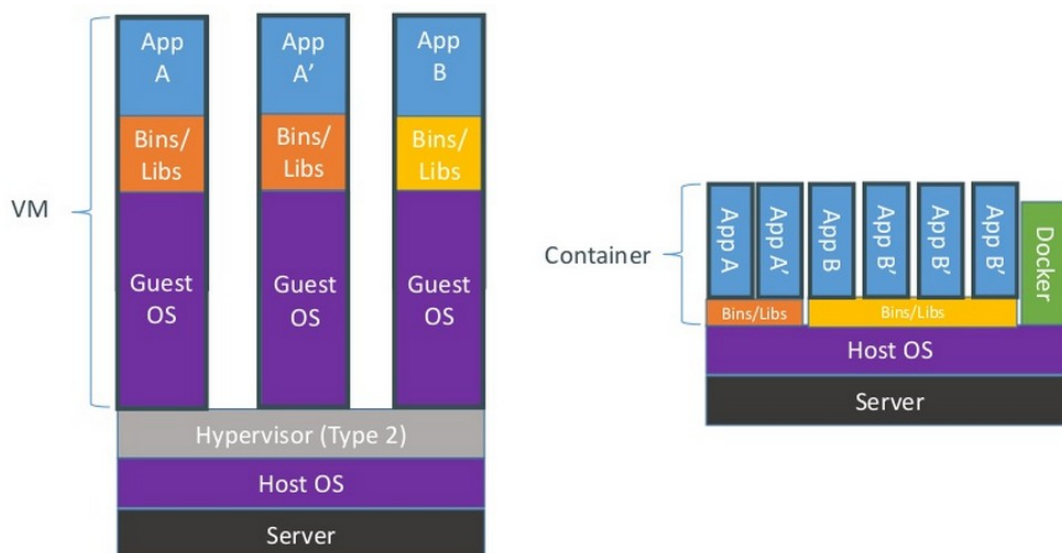


Diagram source: Docker Inc.

Respecto al almacenamiento en disco, una máquina virtual puede ocupar varios gigas ya que tiene que contener el sistema operativo completo, sin embargo los contenedores Docker sólo contienen aquello que las diferencia del sistema operativo en las que se ejecutan, por ejemplo un Ubuntu con Apache ocuparía unos 180 Mb. Desde el punto de vista del consumo de procesador y de memoria RAM, los contenedores Docker hacen un uso mucho más eficiente del sistema anfitrión, pues comparten con él, el núcleo del sistema operativo y parte de sus bibliotecas, con lo que únicamente usarán la memoria y la capacidad de cómputo que estrictamente necesiten. Otras ventajas destacables del uso de Docker serían:

- Las instancias se arrancan en pocos segundos.
- Es fácil de automatizar e implantar en entornos de integración continua.

- Existen multitud de imágenes que pueden descargarse y modificarse libremente.

Como inconvenientes podemos destacar:

- Sólo puede usarse de forma nativa en entornos Unix aunque se puede virtualizar gracias a boot2docker tanto en OSX como en Windows.

- Las imágenes sólo pueden estar basadas en versiones de Linux modernas (kernel 3.8 mínimo).

- Como es relativamente nuevo, puede haber errores de código entre versiones.

2.1.2 Imágenes y Contenedores

Los términos que hay que manejar con Docker son principalmente 2, las imágenes y contenedores.

Las imágenes en Docker se podrían ver como un componente estático, pues no son más que un sistema operativo base, con un conjunto de aplicaciones empaquetadas, mientras que un contenedor es la instanciación o ejecución de una imagen, pudiendo ejecutar varios contenedores a partir de una misma imagen.

Haciendo una analogía con la POO una imagen es una clase y un contenedor es la instanciación de una clase, es decir un objeto.

<https://www.adictosaltrabajo.com/2015/07/29/docker-for-dummies/>

3. Software a utilizar

3.1 git

Cambiamos los repositorios oficiales por los de non-free para poder instalar git. Añadimos, los repositorios anteriores los main se comentan o borran, como buena opción siempre hacer una copia de seguridad del fichero de configuración antes de cambiarlo.

La instalación de git es para poder comunicarnos con la nube de balena que tiene su repositorio git y poder hacer push sobre el repositorio. Requisito para no tener cuenta en git hub

```
vim etc/apt/source.list
```

```
deb http://deb.debian.org/debian/ buster main contrib non-free
deb-src http://deb.debian.org/debian/ buster main contrib non-free
```

```
deb http://security.debian.org/debian-security buster/updates main
contrib non-free
deb-src http://security.debian.org/debian-security buster/updates
main contrib non-free
```

```
# buster-updates, previously known as 'volatile'
```

```
deb http://deb.debian.org/debian/ buster-updates main contrib non-free
```

```
deb-src http://deb.debian.org/debian/ buster-updates main contrib  
non-free
```

```
apt update && apt install git
```

3.1.1 Entendiendo cómo funciona GIT

GIT es el SCV (sistema de control de versiones) de código abierto más utilizado que te permite rastrear los cambios realizados en los archivos. Las empresas y los programadores suelen utilizar el GIT para colaborar en el desarrollo de software y aplicaciones. Un proyecto GIT consta de tres secciones principales: el directorio de trabajo, el área de preparación y el directorio git. El directorio de trabajo es donde se agregan, borran y editan los archivos. Luego, los cambios son preparados (indexados) en el área de preparación. Después de que confirmes tus cambios, la instantánea de los cambios se guardará en el directorio git. Todo el mundo puede usar GIT ya que está disponible para Linux, Windows, Mac y Solaris. El software puede tener una fuerte curva de aprendizaje, pero hay muchos tutoriales disponibles para ayudarte.

<https://www.hostinger.es/tutoriales/comandos-de-git>

3.2 balena-etcher

Instalamos balena-etcher para ello hay que añadir repositorio

```
curl -1sLf \  
  'https://dl.cloudsmith.io/public/balena/etcher/setup.deb.sh' \  
  | sudo -E bash
```

```
apt update
```

```
apt install balena-etcher-electron
```

Error encontrado, no arranca el software, por falta de un enlace simbólico.

```
ln -sf '/opt/balenaEtcher/balena-etcher-electron'  
'/usr/bin/balena-etcher-electron'
```

```
chmod 4755 '/opt/balenaEtcher/chrome-sandbox' || true
```

```
update-mime-database /usr/share/mime || true
```

```
update-desktop-database /usr/share/applications || true
```

3.3 Claves ssh

Para poder conectarse vía ssh entre tu debian y balena hay que

crear unas clave.

Nos vamos al directorio home, en el directorio home sino existe la carpeta .ssh se crea entramos en el directorio.

```
ssh-keygen -t rsa
```

Copiamos id_rsa.pub en authorized_keys

```
cp id_rsa.pub authorized_keys
```

4. Preparación de la aplicación de la nube

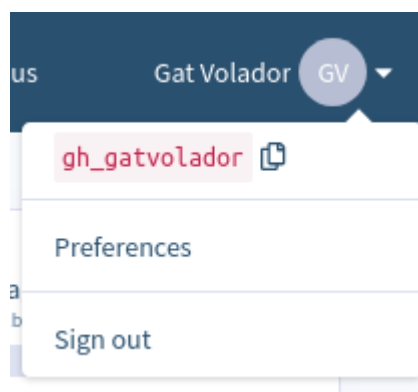
4.1 balena-cloud

balena.io te permite usar herramientas conocidas como Docker y Git para enviar actualizaciones de imágenes de contenedores. Como crear varias aplicaciones, y tener una flota de dispositivos de IOT.

Creamos cuenta en <https://www.balena.io/>.

En la cuenta creada nos creará a la vez la organización.

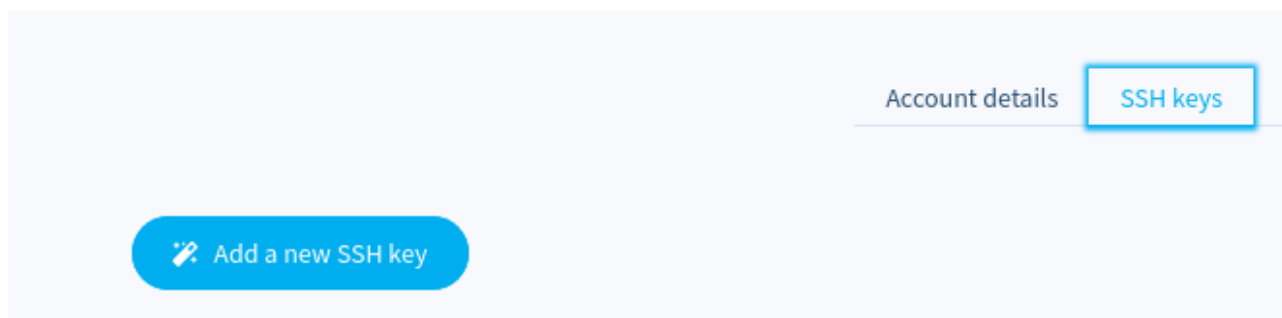
A usuario buscamos preferencias



SSH keys, añadimos nueva ssh key. Le ponemos un nombre. La clave que hemos creado anteriormente en nuestro sistema operativo.

Copiamos el contenido de la clave publica (id_rsa.pub o authorized_keys) pegamos y creamos.

Con esto tengamos comunicación mediante ssh con balena-cloud para



poder hacer las actualizaciones.

Add a new SSH key					Delete keys
<input type="checkbox"/>	Title	Date added	Public key	Fingerprint	
<input type="checkbox"/>	GitHub Key 1	Jul 5th 2021, 12:24 pm	ssh-rsa AAAAB3NzaC1yc2EAAAADAQ...	ad:51:a4:e9:11:ef:85:bc:86:e7:36:d5:20:5d:38:a6	

4.2 Aplicación

Ahora crearemos una aplicación en la cual tendremos que elegir el nombre de la aplicación, indicando que tipo de dispositivo que vamos a utilizar. No siendo de pago en la ultima casilla solo te deja acceder aun método de los tres.

Los tres tipos de aplicaciones disponibles para nuevas aplicaciones son: Inicio Las aplicaciones de inicio, disponibles de forma gratuita para todos los usuarios, tienen todas las funciones, lo que le permite ejecutar varios contenedores y configurar una URL pública para cada uno de sus dispositivos. Tiene un límite de diez dispositivos en todas las aplicaciones de inicio.

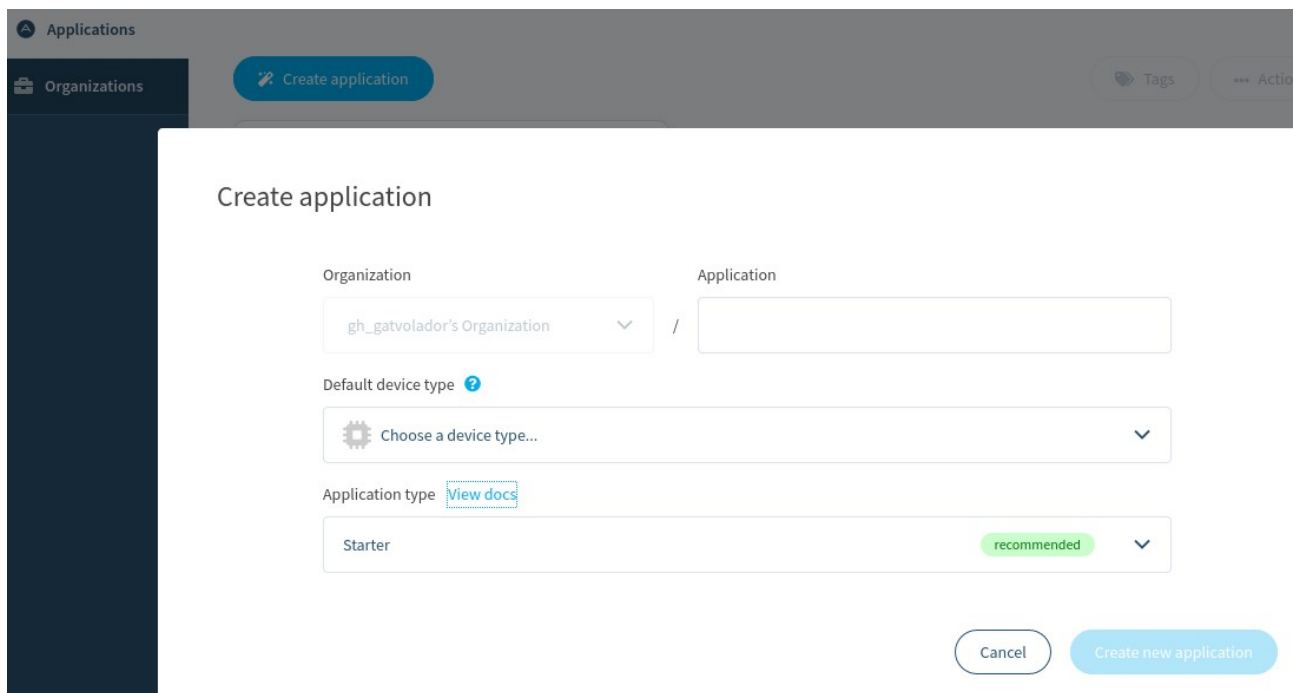
Nota: Los dispositivos inactivos cuentan para el límite de dispositivos de la aplicación de inicio.

Micro-servicios Las aplicaciones de microservicios, como las aplicaciones de inicio, tienen todas las funciones, lo que le permite ejecutar varios contenedores y configurar una URL pública, y están disponibles para todos los planes pagos.

Esenciales Las aplicaciones de Essentials le permiten ejecutar un único contenedor y no permiten URL de dispositivo públicas. Al igual que las aplicaciones de micro-servicios, solo están disponibles para planes pagos.

Nota: Todos los dispositivos en aplicaciones de inicio, micro-servicios y esenciales deben ejecutar balenaOS v2.12.0 o superior.

Si no tiene la opción de elegir uno de estos tipos de aplicaciones, el tipo de dispositivo seleccionado aún no tiene disponible una versión de balenaOS suficientemente alta. Se asignará un tipo de aplicación clásica de forma predeterminada.



The screenshot shows the 'Create application' modal in the Balena Cloud interface. The modal has a dark sidebar on the left with 'Organizations' selected. The main area is titled 'Create application'. It contains the following fields:

- Organization:** A dropdown menu showing 'gh_gatvolador's Organization'.
- Application:** An empty text input field.
- Default device type:** A dropdown menu with a gear icon and the text 'Choose a device type...'. A help icon (?) is next to the label.
- Application type:** A dropdown menu showing 'Starter'. A 'View docs' link is next to the label. A green badge with the word 'recommended' is next to the dropdown.

At the bottom right, there are two buttons: 'Cancel' and 'Create new application'.

Ahora llega la hora de incluir la primer dispositivo en la aplicación. Entramos en la aplicación recién creada y añadimos nuevo dispositivo.

En la primera casilla se elige el tipo de dispositivo en nuestro caso raspberrypi 3B+, en sistemas operativos otra vez como no es una opción de pago solo puedes elegir un SO el balenaOS, en pago te dan otra opción de S.O

Proceso de versión de soporte extendido (ESR)

Nota: esta función solo está disponible en los planes Producción y Empresa. El proceso de Versión de soporte extendido (ESR) de balenaOS permite a los propietarios de flotas actualizar a una nueva versión de ESR en sus dispositivos como máximo dos veces al año para asegurarse de que tienen una versión compatible. Las nuevas versiones de ESR se lanzan con un cronograma de tres meses y, como tal, los propietarios de flotas saben de antemano cuándo estarán disponibles las versiones y cuándo deben actualizarse. Cada versión de ESR se admite durante nueve meses a partir del momento de la publicación. Una versión de ESR compatible implica

que las vulnerabilidades de seguridad de alto riesgo y las correcciones de errores críticos se exportarán a esa versión, y se garantiza que no romperán la interfaz. Los backports de cualquier mejora funcional no están dentro del alcance.

La siguiente opción tenemos indicarle si el dispositivo va a ser de desarrollo o producción. Para primero usuarios siempre recomiendan desarrollo, para nuestro proyecto como recomendación siempre desarrollo.

Tenemos que elegir que tipo red se va conectar el dispositivo mediante cable ethernet o wifi. En nuestro caso es wifi. En este caso hay que indicarle la SSID, passphrase o password.

La imagen utiliza DHCP. En este paso no hay que preocuparse de datos de red ip estática, etc. Se hará más adelante después de quema de la imagen, al modificar un fichero de configuración.

En avanzado solo mueve el tiempo que tarda en buscar actualizaciones, por defecto esta en 10 minutos.

Por última dan la advertencia de que el modelo que dispositivo que hemos escogido no trabaja en banda de 5GHz, solo trabaja en banda de 2,4 GHz.

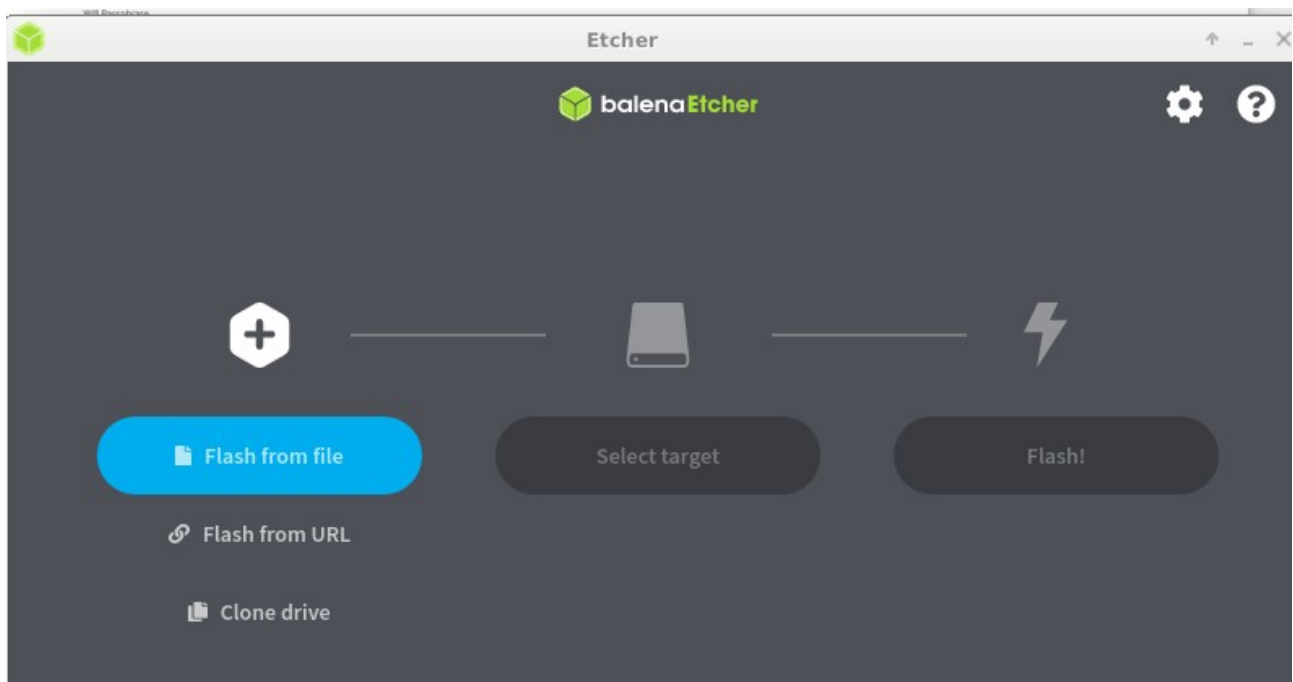
A la derecha de la pantalla una pequeña explicación los pasos a seguir hasta que sale por el panel de balena-cloud y referencia al manual.

Con todo relleno descargamos imagen, normalmente son imágenes pequeñas de unos cientos de megas.

5. Preparación de raspberrypi 3B+

5.1 Balena-etcher

Software preparado para la quema de imágenes en raspberry. Intuitivo de fácil manejo. Seleccionamos imagen descargada, elegimos tarjeta y flasheamos.



5.2 Poner datos de ip estática en SO balenaOS

En el caso de olvido de configuración de la red en este fichero se podría cambiar todo prácticamente, en nuestro caso solo vamos a modificar la parte de la ip estática que no hemos podido directamente por la gui de balena-cloud.


<https://www.balena.io/docs/reference/OS/network/2.x/>

```
[ipv4]  
address1=192.168.1.127/24,192.168.1.1  
dns=8.8.8.8;8.8.4.4;  
dns-search=  
method>manual
```

5.3 Puesta en marcha de balenaOS

Se instala tarjeta en raspberrypi3B+, dejamos que inicie el SO. Si todo va bien tarda unos minutos hasta aparecer por primera vez el dispositivo en nuestro panel de control de balena cloud.

segona



Architecture
aarch64

Created
Jul 7th 2021, 8:13 pm

Starter

Devices

Online
Config

Add device

<input type="checkbox"/>	Name	Status	Device type	Last seen	Created on	UUID
<input type="checkbox"/>	hidden-wave	Offline	Raspberry Pi 3 (using 64bit OS)	2 days ago	Jul 9th 2021, 11:18 am	3450

Una vez iniciado el sistema se quiso hacer una prueba, para comprobar que no se podía iniciar sesión en el S0. Mediante cable HDMI conectar raspberrypi3B+ y monitor. Efectivamente solo se ve una pantalla con el logo de la empresa sin ninguna oportunidad de poder interactuar con el sistema para poder iniciar.

Una vez añadida nuestro dispositivo a balena-cloud. Ya podemos conectarnos mediante ssh:

Al host hace puente a main

```
ssh -p 22222:4321 root@ip
```

Si solo se quiere al host

```
ssh -p 22222 root@ip
```

También acceder a los repositorios git de balena y poder trabajar a ellos.

6. Comunicación con balenacloud

6.1 Preparación de circuito

Vamos hacer un pequeño circuito con un diodo led y una resistencia de 100 ohms. Con su correspondiente fichero de python y tambien preparación del fichero dockerfile que serian los dos ficheros que necesitamos enviar al repositorio git de balena.

Creamos el fichero led.py:

```
#Importamos librerias necesarias:  
import RPi.GPIO as GPIO  
import time  
  
#Pin que vamos utilizar para led BCM  
led = 4  
  
#Configuramos pins  
GPIO.setmode (GPIO.BCM)  
GPIO.setwarnings(False)  
GPIO.setup(led,GPIO.OUT) #led  
  
#Bucle para encienda y apague led  
while 1:  
    GPIO.output(led,1)  
    time.sleep(1)  
    GPIO.output(led,0)  
    time.sleep(1)
```

Creamos fichero Dockerfile:

```
FROM balenalib/raspberrypi3-debian:latest  
ENV INITSYSTEM on  
RUN mkdir -p /usr/src/projectebalena1  
COPY ./led.py /usr/src/projectebalena1  
RUN apt-get update && apt-get install\  
    python3 python3-rpi.gpio -y  
CMD ["python3", "/usr/src/projectebalena1/led.py"]
```

A la hora de instalación y posterior funcionamiento del software instalado, hubo problemas por culpa de la imagen que descarguemos de la librería de balena raspberrypi3-python:3 que es en la que se basada la memoria, por su mal funcionamiento sin poder llegar a poder repararlo. Al final se cambio por raspberrypi3-debian:latest.

El fichero dockerfile tiene poco más que explicar:

Creamos una carpeta para tener todo ordenado, siempre se procurara utilizar la ruta entera, y copiamos el fichero.

```
RUN mkdir -p /usr/src/projectebalena1  
COPY ./led.py /usr/src/projectebalena1
```

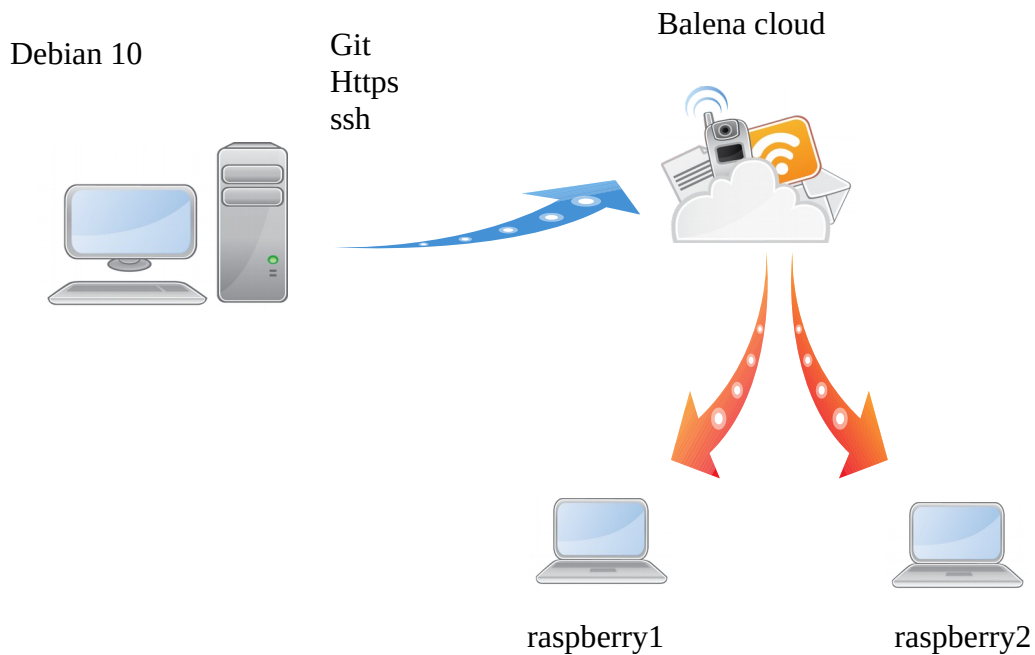
apt update descarga las listas de paquetes de los repositorios y las "actualiza" para obtener información sobre las versiones más recientes de paquetes y sus dependencias. Instalamos los paquetes necesarios para que funcione nuestro programa.

```
RUN apt-get update && apt-get install\  
    python3 python3-rpi.gpio -y
```

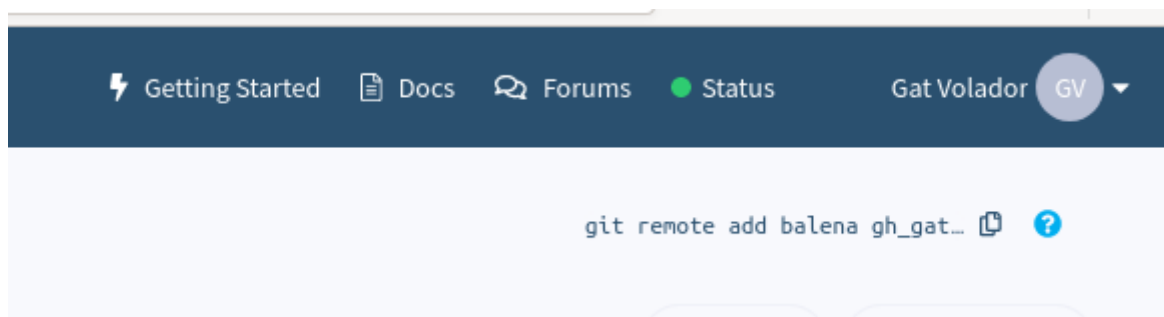
Por último ejecutaremos en un terminal python3 y ejecutará el fichero ubicado en la ruta.

CMD `["python3", "/usr/src/projectebalena1/led.py"]`

6.2 Envío de archivos



Solo nos faltaria conseguir la dirección de nuestro repositorio en balena de nuestra aplicación para poder conectarnos. En el panel de balena en el panel de la izquierda, clicamos a la aplicación que hemos creado anteriormente. Cuando se despliegue el submenu buscamos devices clicamos. En la parte superior derecha de la pantalla hay una ventana donde aparecerá la dirección. Copiamos



En nuestro caso seria:

git remote add balena [gh_gatvolador@git.balena-](https://github.com:gh_gatvolador@git.balena-)

[cloud.com:gh_gatvolador/segona.git](#)

Creamos un directorio en nuestro debian para tener una zona de montaje del proyecto

mkdir proyecto

Copiamos ficheros dockerfile y led.py a directorio y comprobamos:

cp led.py Dockerfile proyecto/ && cd proyecto && ls

Con esta información lo primero sería es clonar el repositorio que tenemos en balena cloud localmente. Con el comando copiado anteriormente modificamos el git remote add por git clone

git clone [gh_gatvolador@git.balena-cloud.com:gh_gatvolador/segona.git](#)

La siguiente acción sería iniciar el repositorio local

git init

Añadimos los ficheros del directorio actual

git add .

Se añade dirección donde se va mandar el repositorio.

git remote add balena [gh_gatvolador@git.balena-cloud.com:gh_gatvolador/segona.git](#)

El comando git commit guardará todos los cambios hechos en la zona de montaje o área de preparación (staging area), junto con una breve descripción del usuario, en un "commit" al repositorio local.

git commit

Y por último se le manda con un push:

git push balena master

Al final de la subida del repositorio

```
[Info] Redeploying release for commit 5ad6b78ef1ada

o git.balena-cloud.com:gh_gatvolador/segona.git
07b6717..5ad6b78 master -> master
```

```
git clone gh\_gatvolador@git.balena-cloud.com:gh\_gatvolador/segona.git
git init
git add .
git remote add balena gh\_gatvolador@git.balena-cloud.com:gh\_gatvolador/segona.git
git commit
git push balena master
```

Hay que esperar totalmente que en el panel pase el update, installing y el reboot que hace para que funcione la actualización. Dicho de otra manera si ya tenemos funcionando otra imagen seguirá funcionando esta imagen hasta que no haga el reboot despues de la instalación de la descarga.

STATUS
✓ Online

UUID
345059f

TYPE
Raspberry Pi 3 (using 64bit OS)

ONLINE FOR
2 minutes

HOST OS VERSION
balenaOS 2.80.3+rev1
development

SUPERVISOR VERSION
12.7.0

CURRENT RELEASE
5ad6b78 ✓

TARGET RELEASE
5ad6b78

LOCAL IP ADDRESS
10.199.160.157

PUBLIC IP ADDRESS
83.247.137.2

MAC ADDRESS
B8:27:EB:A7:2D:0C
B8:27:EB:F2:78:59

TAGS (0)
No tags configured yet

PUBLIC DEVICE URL
☐

NOTES
Add device notes...

SERVICES

Service	Status	Release
main	✓ Running	5ad6b78

A la hora de actualizar el repositorio y estamos trabajando en la misma zona de preparación hay que seguir otra serie de comandos porque daría resultado de errores de envío, repositorios duplicados.
En un segundo envío los comandos se quedaría de la siguiente manera.

```
git add .
git commit
git push balena master
```

Siempre hay que recordar, que hasta que no haga el reboot despues de la instalación, seguirá funcionando la imagen anterior.

6.3 Añadir segundo dispositivo a una misma aplicación

El añadir un segundo dispositivo a la misma aplicación es más rápido, al ya tener todo actualizado es simplemente añadir el dispositivo y esperar que actualice y arranque el servicio directamente.

Se hace la misma operación que la primera, nos vamos a aplicación balena-cloud damos en nuestra aplicación. En sumario añadimos dispositivo, seguimos los mismos pasos que seguimos en el

primer dispositivo. Después de flashear imagen y cambiar configuración del fichero de red para poder colocar ip. Insertamos tarjeta y esperamos a que salga el dispositivo en panel de balena-cloud.

