Documentación Técnica: Instalación y uso de Coolify con Ngrok

Autor: Pedro José Meixús Belsol

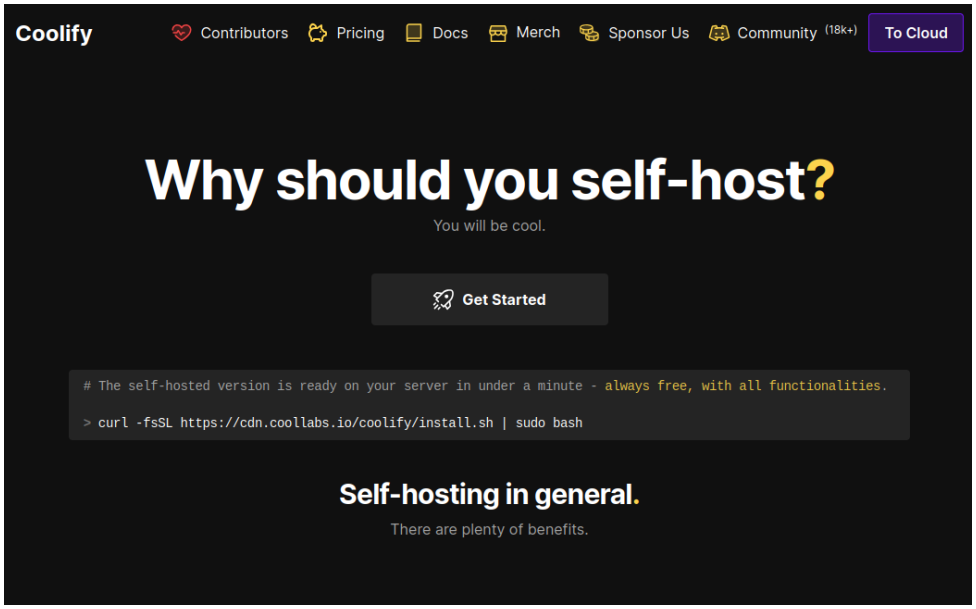
23 de enero de 2026

Índice

1. Instalación y preparación

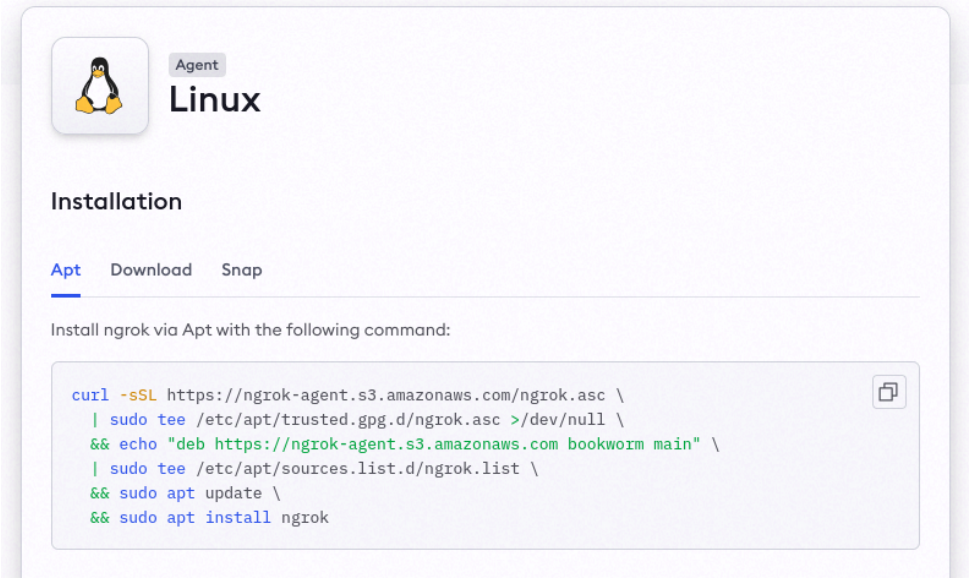
Para empezar, instalaremos una máquina virtual Linux (Ubuntu).

Ya con ella instalada, vamos a descargar Coolify, vamos a su página y utilizamos el comando que nos indican para la instalación:



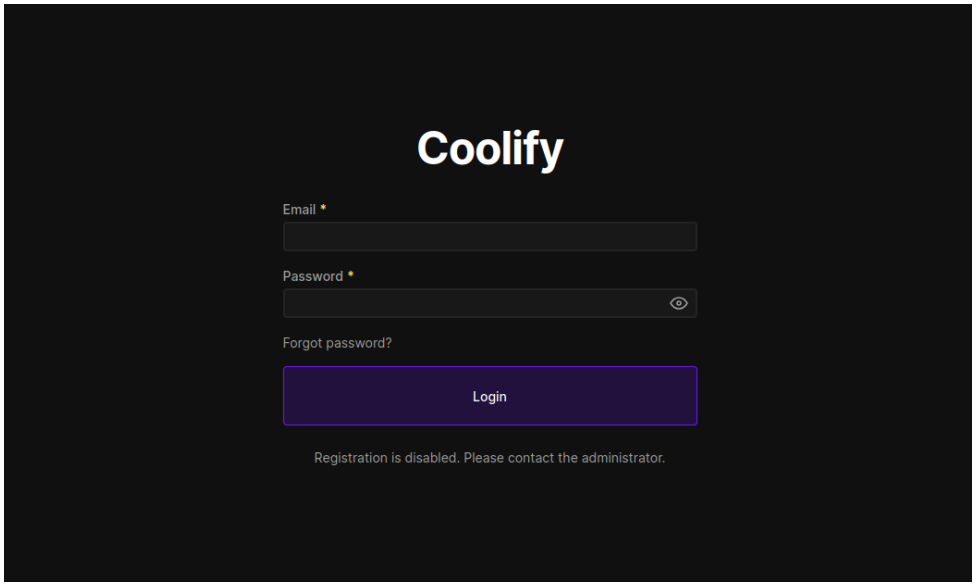
Se instalará y una vez termine accederemos a la IP que nos indica y crearemos una cuenta de Coolify.

Hecho esto vamos a instalar Ngrok, igual que con Coolify, accederemos a su página y usaremos el comando que nos indican:

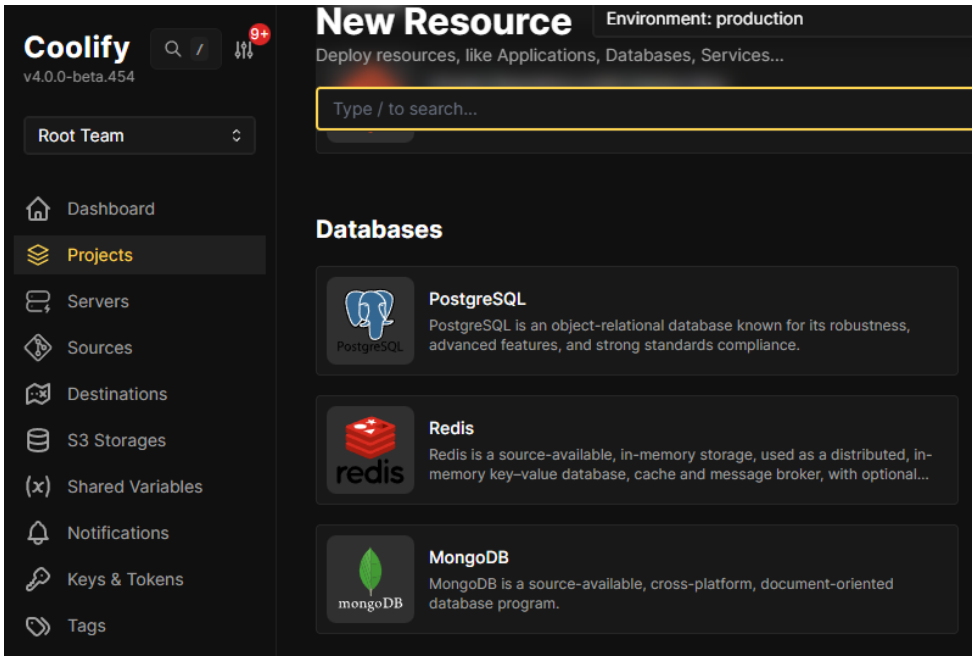


2. Configuración Coolify

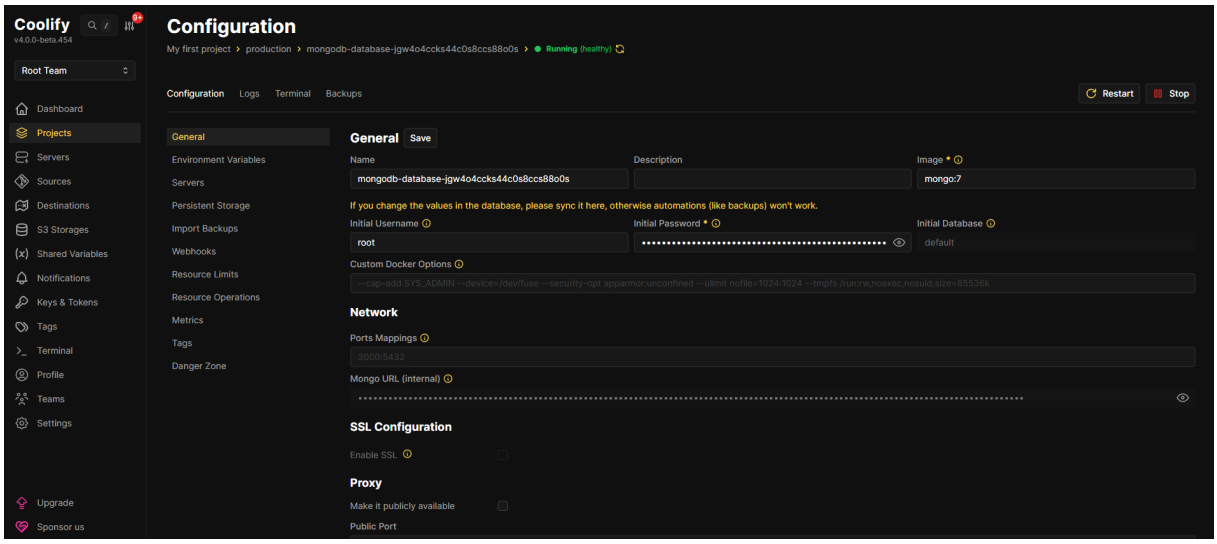
Una vez hecho esto, accederemos al 'localhost:8000' desde nuestro host, donde deberá aparecer la interfaz de Coolify, en esta podremos logearnos.



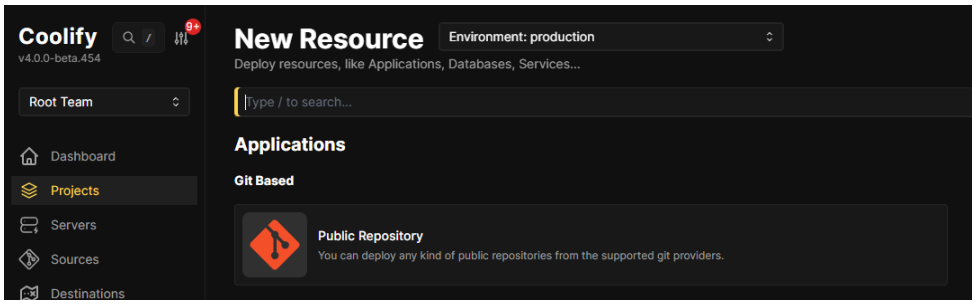
Una vez logeados crearemos nuestro propio proyecto. En el crearemos una nueva base de datos (en nuestro caso será Mongo):



Podremos acceder a su configuración y veremos que nos proporciona una URI. Le daremos a 'Deploy' y si arranca correctamente mostrará 'Running'



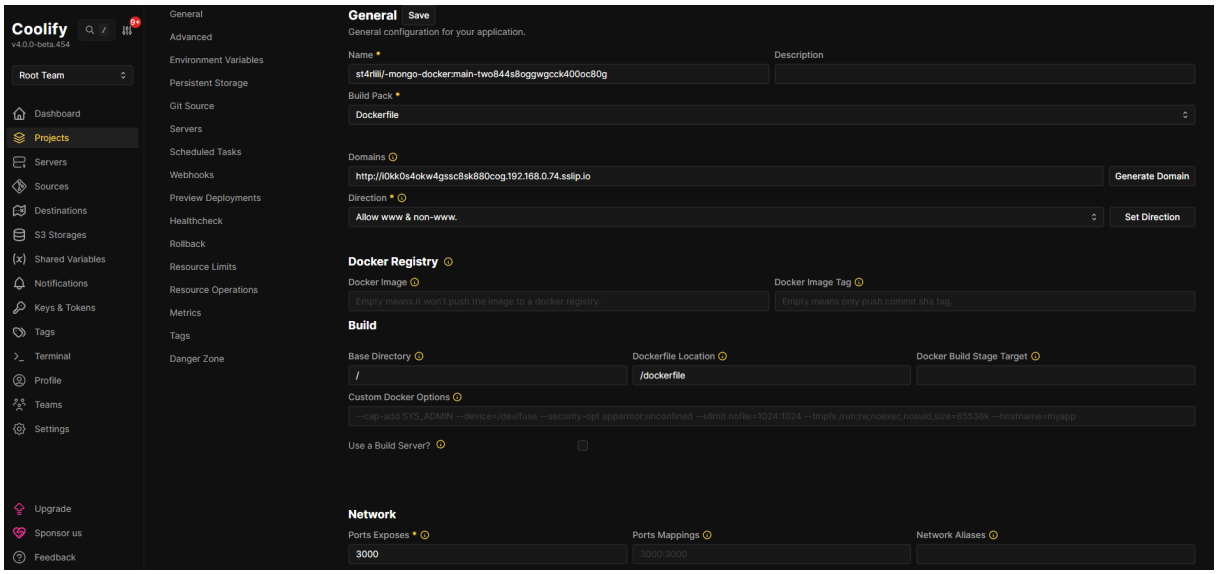
Ahora con la BBDD preparada, creamos la API, que obtiene desde nuestro repositorio de github:



Para una futura comprobación le añadire algo para mostrar, por ejemplo la palabra 'contenido'

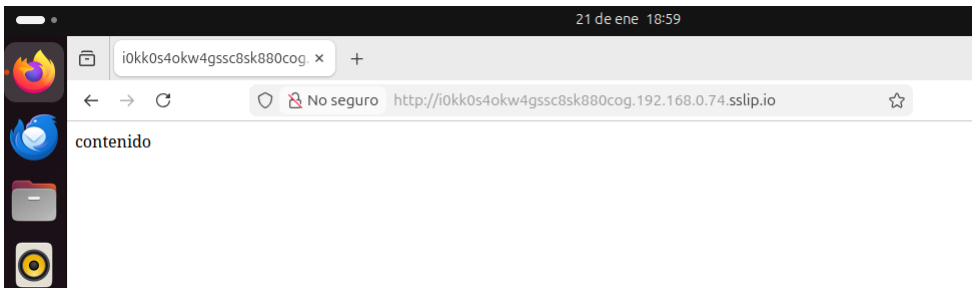
```
const app = express();
const PORT = 3000;

app.get('/', (req, res) => {
  res.send('contenido')
})
```

Una vez indicado el repositorio y aceptado, se desplegará y cuando termine nos dejará en una ventana como esta, donde debemos indicar el dockerfile, puerto, etc.

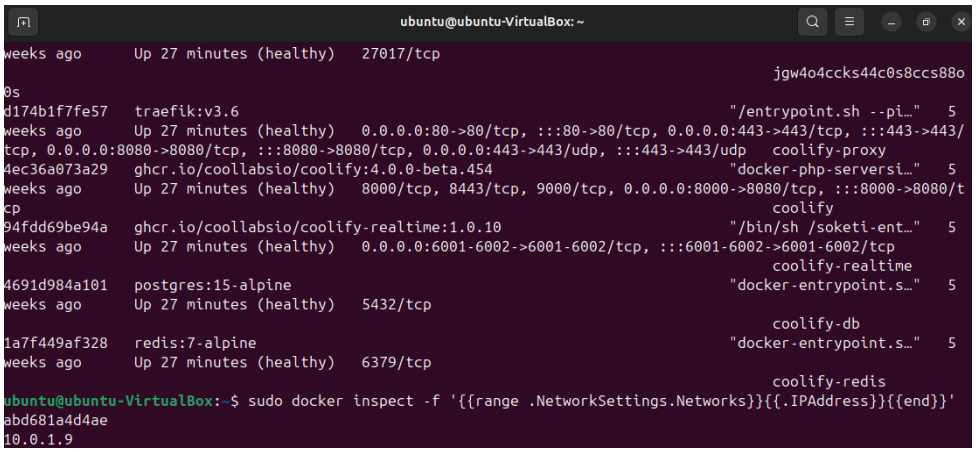

Coolify
v4.0.0-beta.454
Root Team
Dashboard
Projects
Servers
Sources
Destinations
S3 Storages
Shared Variables
Notifications
Keys & Tokens
Tags
Terminal
Profile
Teams
Settings
Upgrade
Sponsor us
Feedback
General
Advanced
Environment Variables
Persistent Storage
Git Source
Servers
Scheduled Tasks
Webhooks
Preview Deployments
Healthcheck
Rollback
Resource Limits
Resource Operations
Metrics
Tags
Danger Zone
General
Save
General configuration for your application.
Name *
st4rlll/-mongo-docker:main-two844s8oggwgcck400oc80g
Description
Build Pack *
Dockerfile
Domains
http://l0kk0s4okw4gssc8sk880cog.192.168.0.74.sslip.io
Generate Domain
Direction *
Allow www & non-www.
Set Direction
Docker Registry
Docker Image
Empty means it won't push the image to a docker registry.
Docker Image Tag
Empty means only push commit sha tag.
Build
Base Directory
/
Dockerfile Location
/dockerfile
Docker Build Stage Target
Custom Docker Options
--cap-add SYS_ADMIN --device=/dev/fuse --security-opt apparmor:unconfined --ulimit nofile=1024:1024 --tmpfs /run:rw,noexec,nosuid,size=65536k --hostname=myapp
Use a Build Server?
Network
Ports Exposes *
3000
Ports Mappings
3000:3000
Network Aliases

3. Pruebas de conexión

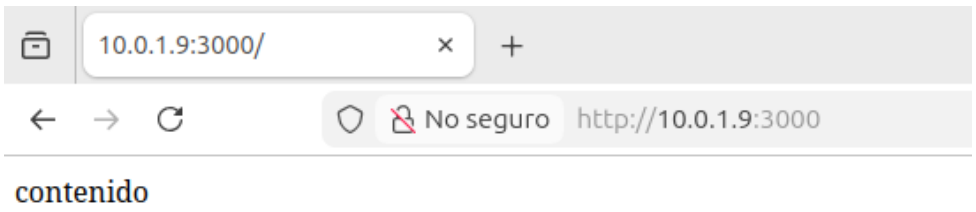
Con la IP que nos genera en el apartado 'Domains' podemos cambiar la dirección IP que sale ahí por la de nuestra máquina y redesplegar la API, así podremos acceder desde nuestra máquina virtual.



Ahora que tenemos la conexión funcional, vamos a buscar la IP del contenedor Docker donde esté nuestra API, para eso usaremos el comando 'inspect' junto con los parámetros necesarios y el ID del contenedor.



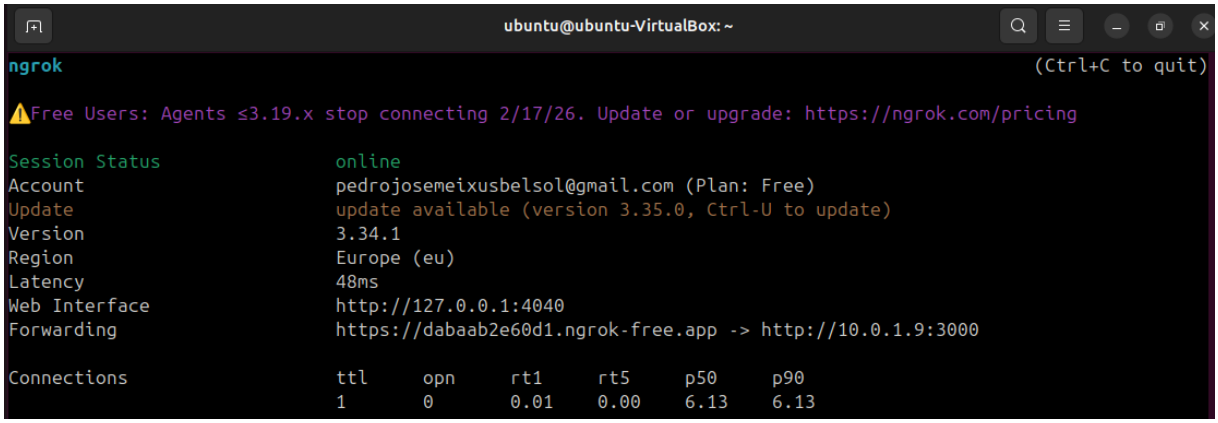
Si introducimos esa IP que nos ha dado junto con el puerto usado (3000) nos deberá llevar al mismo lugar.



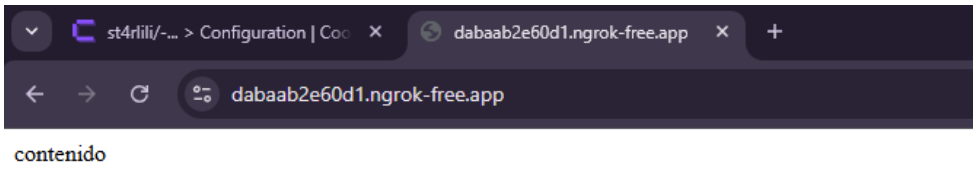
Si lo anterior ha funcionado, entonces podemos utilizar ngrok para exponer nuestra API, con el siguiente comando:

```
ubuntu@ubuntu-VirtualBox:~$ sudo docker inspect -f '{{range .NetworkSettings.Networks}}{{.IPAddress}}{{end}}'
abd681a4d4ae
10.0.1.9
ubuntu@ubuntu-VirtualBox:~$ ngrok http 10.0.1.9:3000
```

Nos llevará a esta ventana (la cual dejaremos abierta), que es el túnel que nos da la URL (la del apartado 'Forwarding') con la que podremos acceder desde nuestra máquina al contenido de la API.

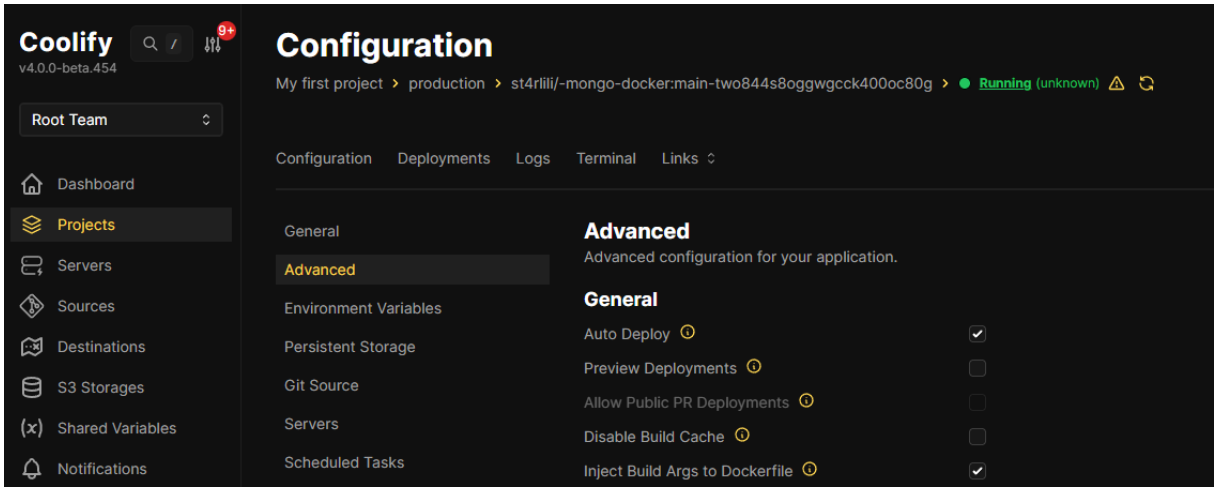


Accediendo a esa URL nos deberá aparecer el contenido correspondiente, al igual que en las anteriores pruebas.

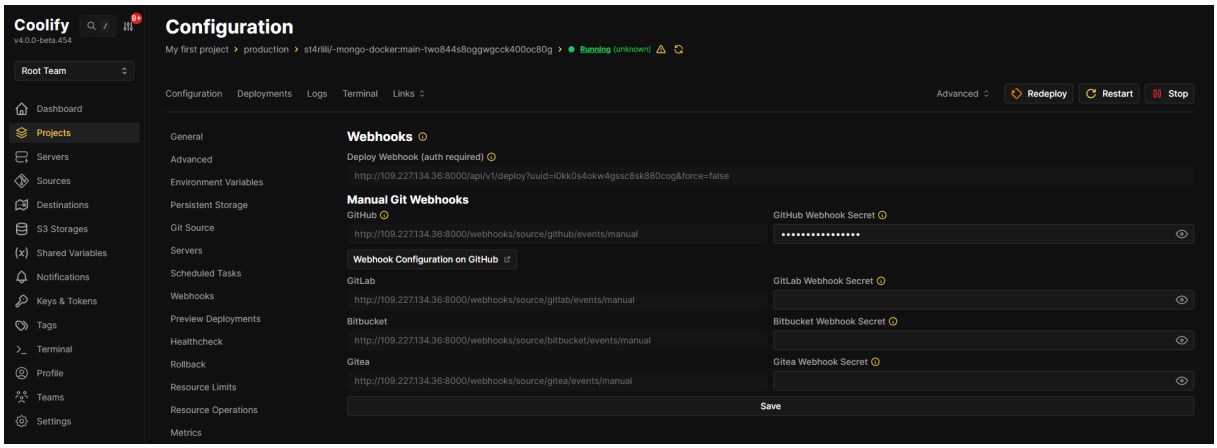


4. Webhook

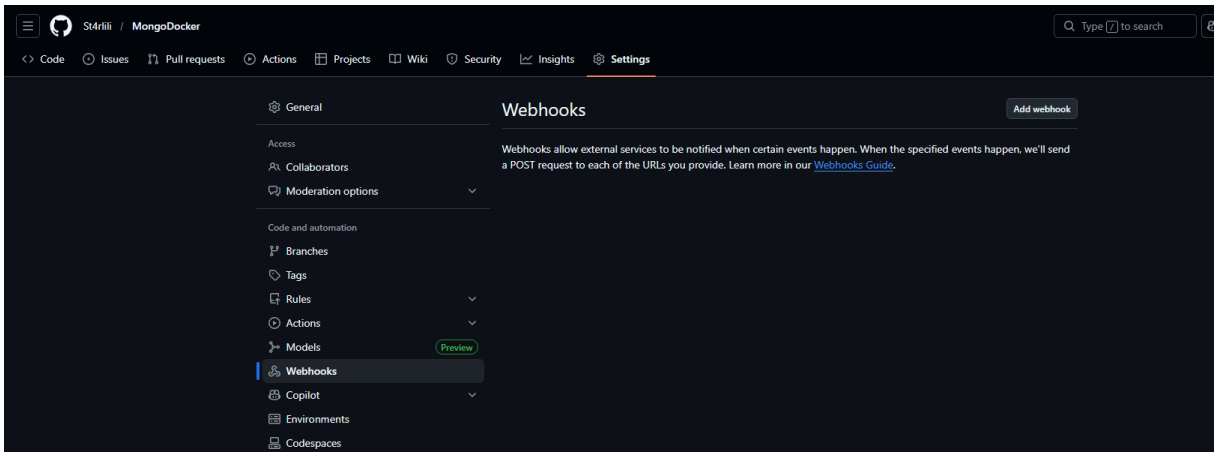
El último paso será configurar el webhook. Primero tendremos que ir a la configuración de nuestra API y activar el 'Auto Deploy'.



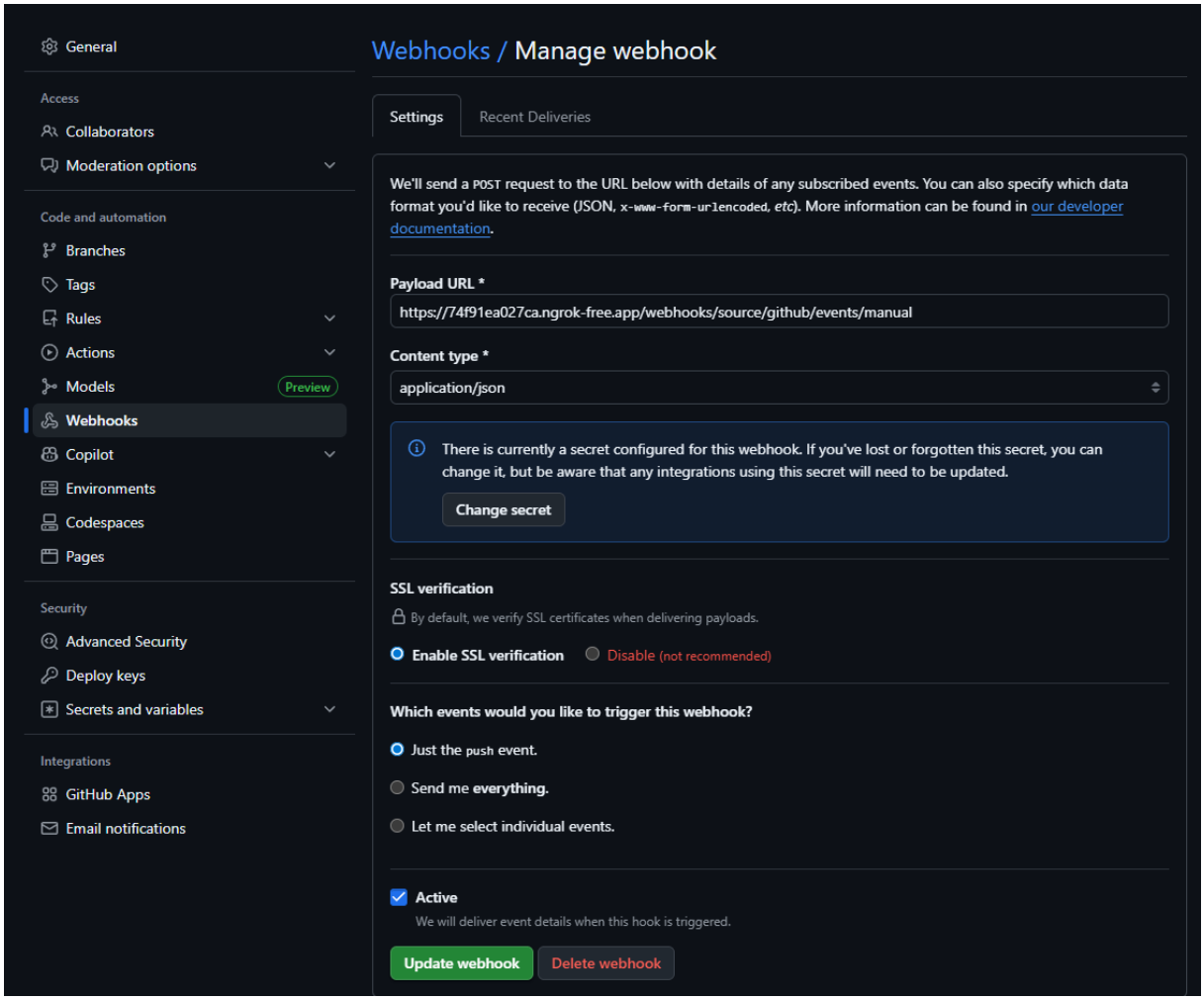
En el apartado de Webhook de la configuración de la API, añadiremos un 'secret' que será la 'contraseña' de nuestro webhook.



Ahora iremos al repositorio de github, al apartado 'settings' y en 'Webhooks' añadiremos uno nuevo.



Aquí solo deberemos introducir el 'Payload URL' que será la URL que nos da ngrok junto con la ruta de los webhooks de coolify, además de el secret que indicamos en coolify. El resto quedará predeterminado.



Con eso hecho el webhook estaría hecho y debería funcionar correctamente. En mi

caso tuve un error y coolify no recibía el mensaje del webhook, estuve probando por si podia ser el puerto, la ruta de ngrok o algo similar, pero no hago que funcione.