

IES Puerto de la Cruz Telesforo Bravo



Gobierno de Canarias

Consejería de Educación,
Universidades, Cultura y Deportes

38003999 IES Puerto de la Cruz - Telesforo Bravo

BUDDY (CI/CD)

DESPLIEGUE DE APLICACIONES WEB

Autor:

Gersán Cabo Del Pino

Indice

1. ¿Que es?	3
2. Utilización de pipelines	3
3. Comenzando con Buddy	4
3.1 Creamos una cuenta	4
3.2 Enlazamos con un proyecto existente	5
3.3 Pipelines mediante la GUI	7
3.4 Añadimos acciones a la pipeline	9
3.5 Arrancamos el pipeline	12
3.6 Pipelines mediante YML	13
4. Planes de pago	15
5. Buddy vs Circle CI	16
6. Buddy vs GitLab CI	16
7. Buddy vs Jenkins	17
8. Conclusión	18

1. ¿Que es?

Buddy es una herramienta web de integración y despliegue continuo, esta se integra con las principales plataformas usadas el desarrollo de software (GitHub, AWS, Docker, etc) y con múltiples lenguajes (JavaScript, PHP, Python, Ruby, Java, Go, etc).

Su principal elemento diferenciador frente a los competidores es que promete permitirnos hacer las pipelines desde una interfaz web sencilla y fácil de utilizar, evitandonos así tener que realizar ficheros .yml, aunque también se nos permite utilizarlos.

2. Utilización de pipelines

Buddy ofrece la posibilidad de diseñar el despliegue continuo mediante pipelines personalizadas que confirmen que todo ha ido correctamente:



Estos pipelines nos ayudarán a cosas como:

- Testear que la aplicación funciona correctamente antes de subir los cambios a la rama correspondiente.
- Testear que los envíos a un servidor reciben una respuesta correcta.
- Notificarnos en caso de fallos o incidencias.

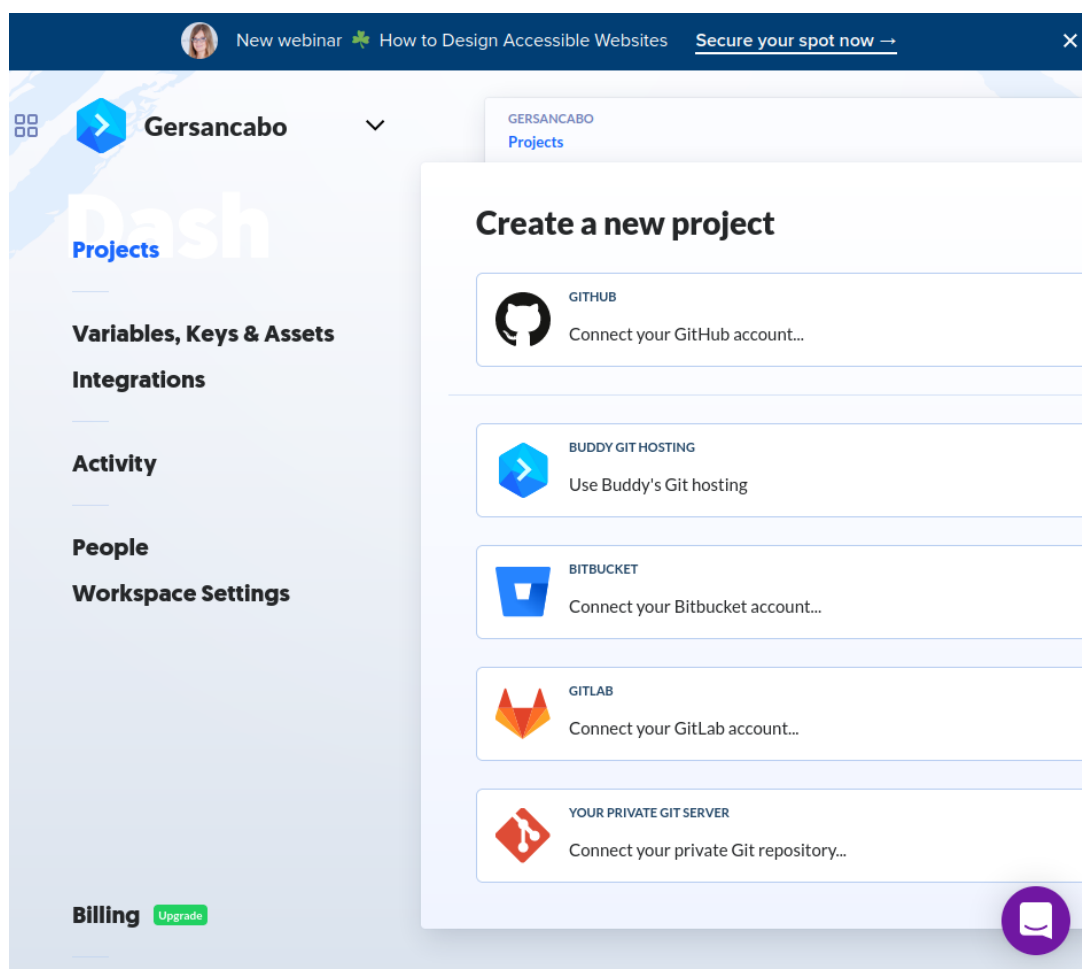
3. Comenzando con Buddy

3.1 Creamos una cuenta

Primero debemos crearnos una cuenta desde su [página principal](#), tienen distintos planes de pago (de los que hablaremos más adelante), nosotros nos decantamos por el gratuito para este tutorial.

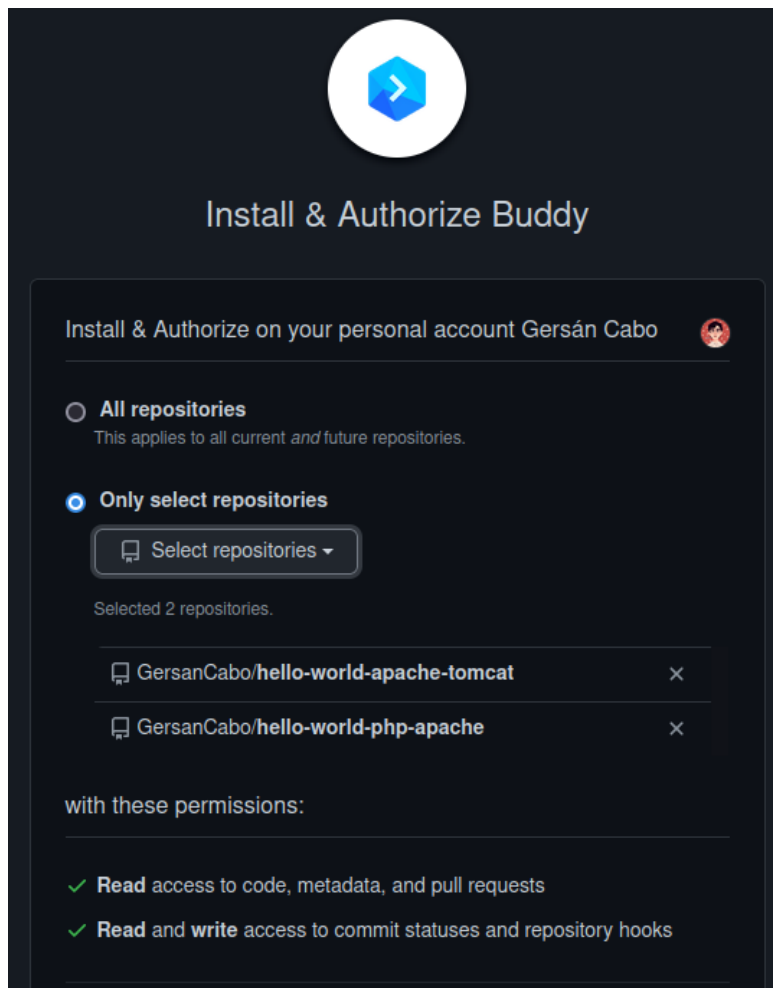
La cuenta la podemos crear con un email de empresa o iniciando sesión en GitHub, GitLab o BitBucket.

Esta es la página de inicio que nos recibirá:



3.2 Enlazamos con un proyecto existente


Para ello conectaremos con nuestra cuenta de Github y elegiremos el proyecto que nos interese testear.



Podemos también dar permiso a Buddy para acceder a todos nuestros repositorios, esto ya depende de nuestro parecer y podremos modificarlo cuando deseemos desde nuestra configuración de GitHub.

Después de darle permisos, nos aparecerá esta pantalla:

Create a new project



GIT PROVIDER

GersanCabo

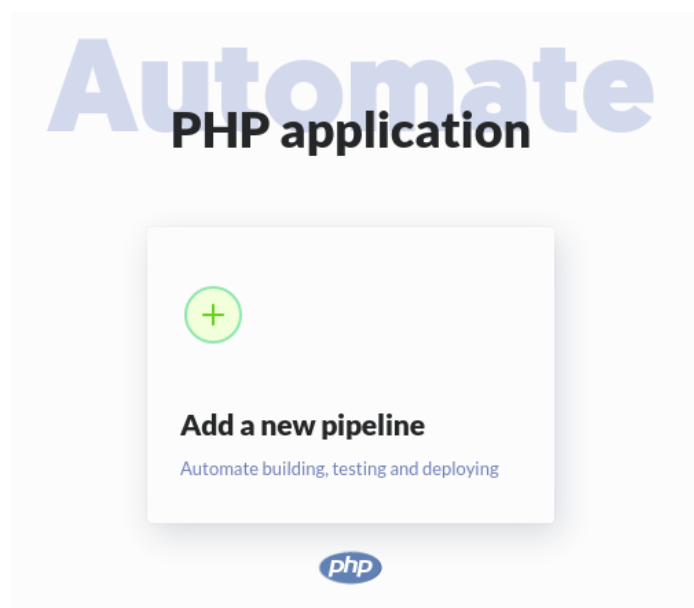
ORGANIZATION

GersanCabo

CAN'T SEE YOUR REPOSITORIES?

NAME	LAST UPDATE ^
Filter repository...	
hello-world-apache-tomcat	7 days
hello-world-php-apache	7 days


Una vez seleccionado ya podremos empezar a añadir pipelines:




3.3 Pipelines mediante la GUI

Esta será la página de configuración de la pipeline:

Add a new pipeline

New

Duplicate

PURPOSE

CI/CD: Build, test & deployAutomation: Monitor, backup & other




Pipeline runs at the provided Git revision, which is fetched to the filesystem.

NAME

Example: Deploy to production

TRIGGER

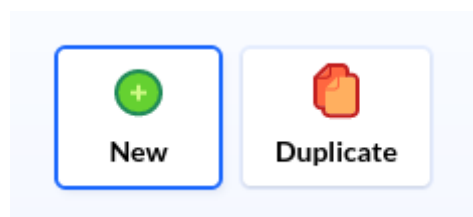
On eventsManuallyOn schedule

TRIGGER EVENT	SETTINGS	
Git push	 master	
		

Add pipeline

Ahora empezare a definir cada uno de los apartados que podemos configurar:

- Nueva o duplicada



Podemos crear la pipeline desde cero o duplicar ya una existente en otro proyecto

- Proposito

PURPOSE	
CI/CD: Build, test & deploy	Automation: Monitor, backup & other
Pipeline runs at the provided Git revision, which is fetched to the filesystem.	

Nuestra pipeline puede tener dos propósitos:

- Para integración y despliegue continuo.
- Para automatizar diversos procesos ajenos a lo primero.

- Nombre

NAME
Example: Deploy to production

Este apartado no tiene mucho misterio, será el nombre de nuestra pipeline y debe definir lo que hace la misma

- Disparador

Los triggers que activen la pipeline pueden ser los tres siguientes

- Eventos: Estos los podemos definir, uno utilizado frecuentemente cuando nuestro proyecto utiliza un repositorio de GitHub es el 'git push' de un usuario hacia una rama.
- Manual: El pipeline debe ser activado manualmente.
- Intervalos de tiempo: El pipeline se realizará cada cierto tiempo, podremos definir la franja que nos interese.

A partir de aquí cada trigger tiene su propia configuración, la que nosotros utilizaremos y, quizás la más interesante, es el trigger mediante evento.

3.4 Añadimos acciones a la pipeline

Ahora tendremos que decidir las acciones que queremos realizar entre todas las que nos ofrece Buddy.

Estas se dividen en categorías y Buddy nos sugiere algunas según los archivos que detecte en nuestro proyecto.

SUGGESTIONS

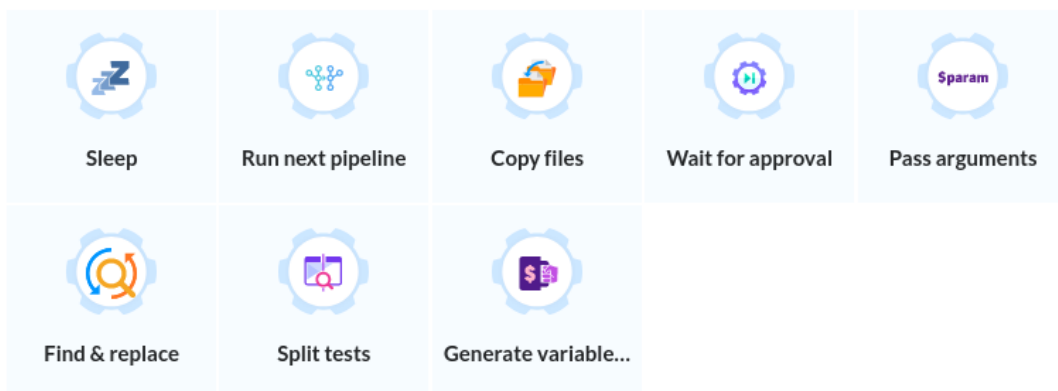


Algunas categorías interesantes son:

- Setup

Aquí encontraremos diversas acciones que afectarán al propio uso de la pipeline y al propio proceso de CI/CD que nos ofrece Buddy. Podríamos decir que son acciones que afectan más al trabajo de las pipelines que al propio proyecto.

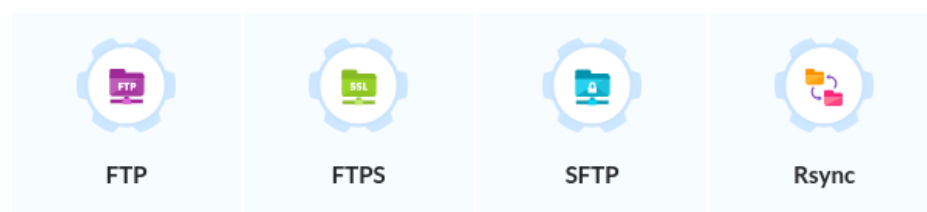
SETUP



- Transfer

Nos puede interesar transferir el propio proyecto a un servidor tras finalizar los pipelines de comprobación, transferir archivos hacia otra máquina, etc. Para ello están las siguientes acciones que hacen uso de la tecnología FTP y derivados.

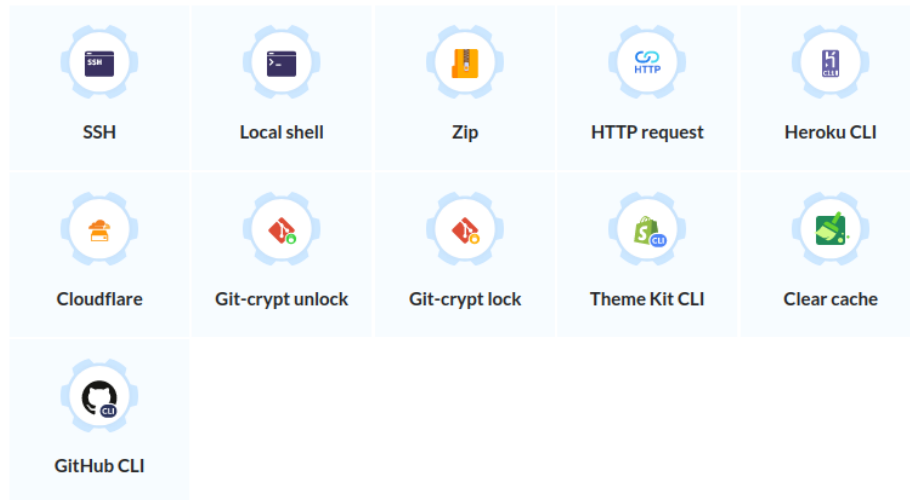
TRANSFER



- DevOps

En el ciclo de vida del desarrollo de software encontramos muchas tareas repetitivas que ralentizan nuestro trabajo, mediante estas acciones podremos automatizarlas, liberarnos de esa carga y agilizar el tiempo.

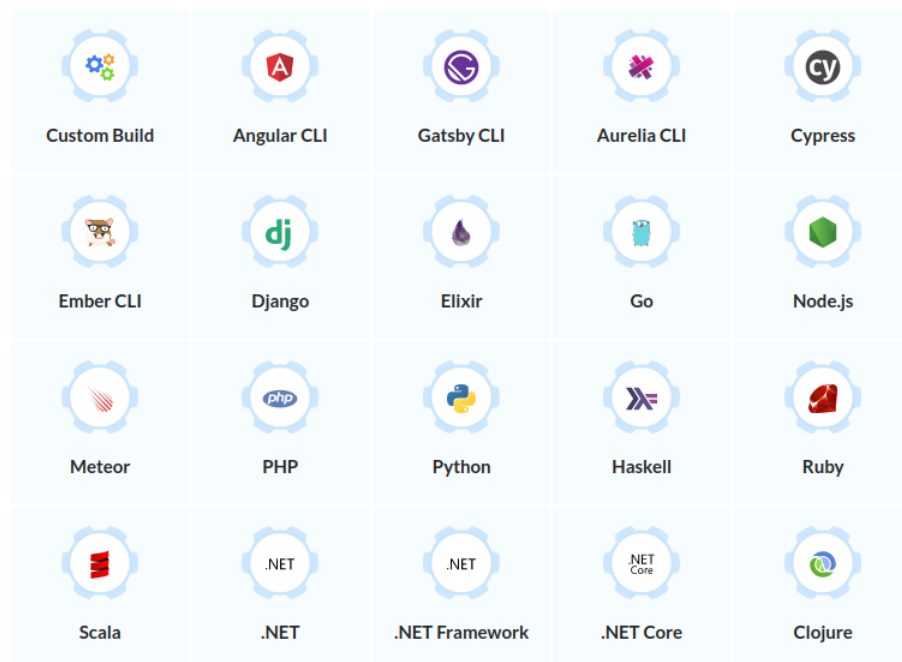
DEVOPS



- Run commands in dedicated environment

Esta es la categoría que cuenta con más acciones, ya que tenemos para decidir entre múltiples entornos (imágenes de Docker) con una serie de herramientas preinstaladas, además podremos utilizar otra imagen de Docker ajena a las que nos recomienda Buddy

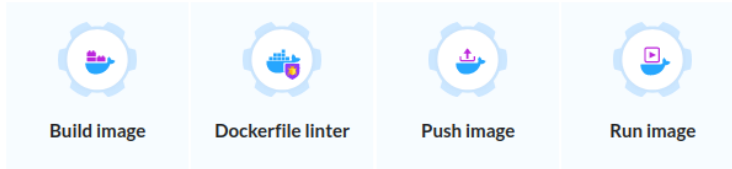
RUN COMMANDS IN DEDICATED ENVIRONMENT



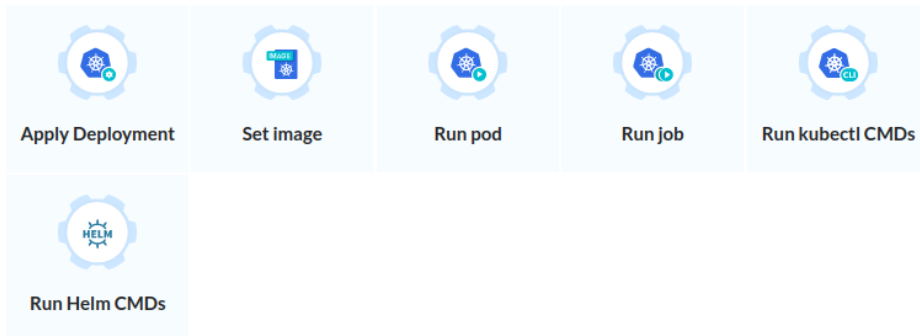
- Docker y Kubernetes

Utilizar Docker en nuestros proyectos es algo del día a día, con lo que no podían faltar esta serie de acciones para trabajar con contenedores e imágenes.

DOCKER



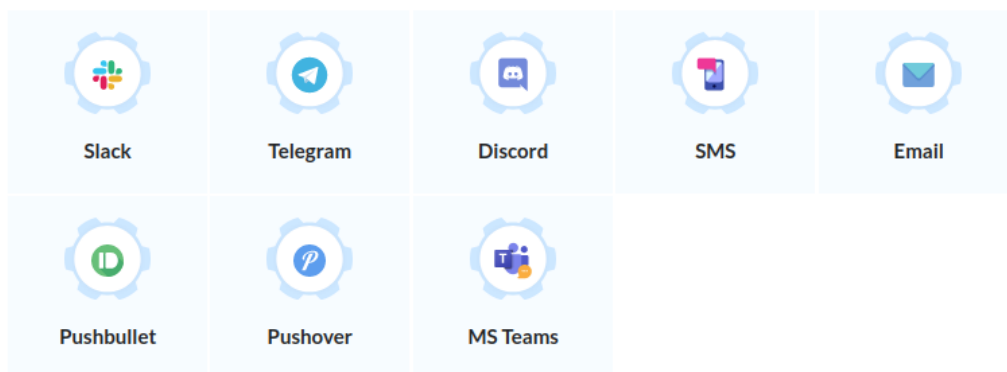
KUBERNETES



- Notifications

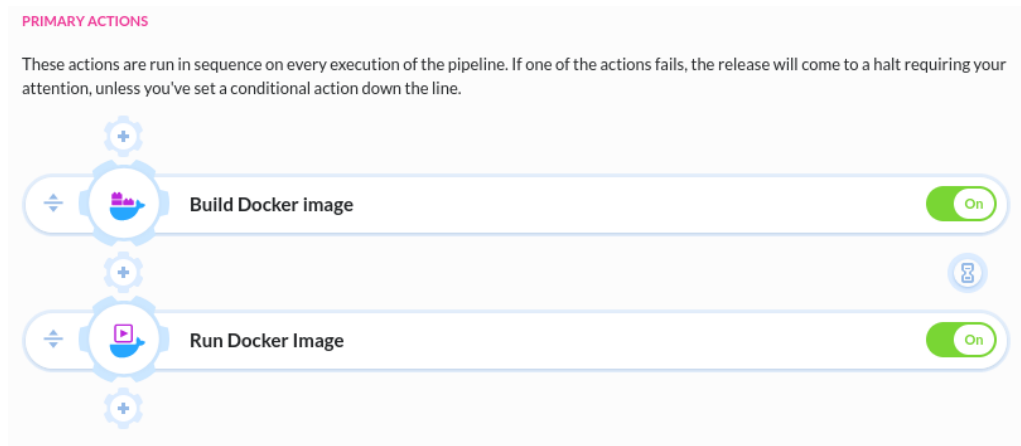
Conocer el resultado de nuestras pipelines o estar al tanto del monitoreo de nuestra aplicación resulta indispensable, para esto contamos con esta serie de acciones que utilizan herramientas de mensajería que nos son conocidas.

NOTIFICATIONS



Cada acción tiene sus propias opciones y configuración, además podremos crear las nuestras propias o editar los comandos que realizan cada una, con lo que enseñarlas alargaría este documento más de lo esperado, así que invito al lector a verlas por sí mismo y toquetear un poco la herramienta.

En nuestro caso, haremos un pipeline sencillo que consistiría en construir la imagen de Docker y arrancarla.



3.5 Arrancamos el pipeline

Le damos al botón Run y nos aparecerá lo siguiente:

Run Ejemplo Pipeline pipeline

REVISION

7

#28401b0: volviendo a any

Change commit...

COMMENT

Probando la pipeline por primera vez

[+ Pass variables](#)




Priority

Normal

Ponemos un comentario descriptivo como el de la captura y arrancamos.

A medida que se van realizando las acciones se nos mostrará el resultado de las ya terminadas, además tendremos acceso sencillo a los logs de las mismas.

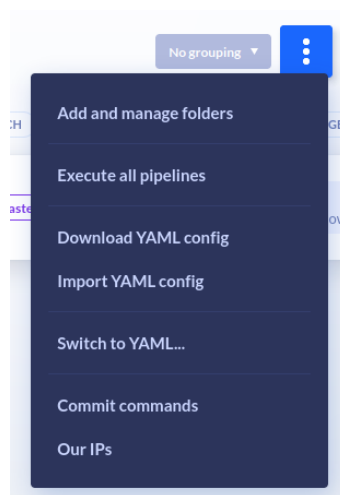
ACTIONS PERFORMED IN THIS EXECUTION

EXECUTED ACTION	STATUS
 Prepare Environment	00:00 Logs
 Build Docker image	00:26 Logs
 Run Docker Image	WORKING Logs

Dependiendo de nuestro plan de pago las acciones tardarán más o menos y tendremos cola si el tráfico de Buddy en ese momento es muy alto.

3.6 Pipelines mediante YAML

Para cambiar de modo GUI a modo YAML uno de nuestros proyectos debemos tocar la siguiente configuración:



Al tocar en “Switch to YAML...” nos aparecerá un menú de configuración, en el cual debemos seleccionar lo siguiente:

PIPELINE CONFIGURATION

Web UI **YAML**

INSTRUCTIONS Download buddy.yml or copy to clipboard and push it to the repository

Y nuestro proyecto pasará a modo YAML y buscara pipelines declaradas en el archivo buddy.yml de nuestro repositorio, un ejemplo de uno seria este:

```
- pipeline: "Java example"
  on: "EVENT"
  events:
  - type: "PUSH"
    refs:
    - "refs/heads/master"
  priority: "NORMAL"
  fail_on_prepare_env_warning: true
  actions:
  - action: "Execute: mvn clean install"
    type: "BUILD"
    working_directory: "/buddy/presentacion-buddy-java"
    docker_image_name: "library/maven"
    docker_image_tag: "3.3.3"
    execute_commands:
    - "mvn clean install"
    - "mvn clean test"
    cached_dirs:
    - "/root/.m2/repository"
    volume_mappings:
    - ":/buddy/presentacion-buddy-java"
    cache_base_image: true
    shell: "BASH"
```

Algunas etiquetas relevantes son:

- 'on' y 'events': Donde declararemos el evento, su disparador y la rama que hara activar este disparador (en caso de que nuestro disparador se active con un cambio en el repositorio).
- 'priority': Con los distintos planes de pago de Buddy podremos tener en proceso 1 o 2 pipelines a la vez, por lo que con esta opción podremos elegir las pipelines a las que damos más importancia y prefiramos que se ejecuten primero.
- 'actions': En esta etiqueta se detalla la acción a realizar en sí, declaramos el nombre de esta etapa de acciones ("Build", "Develop", "Test", etc), los directorios donde trabajaremos, la imagen de docker que usaremos, los comandos a ejecutar, la shell que utilizara Buddy, etc.

4. Planes de pago

Contamos con 4 planes de pago:

- Free (0\$/mes):
 - Hasta 5 proyectos
 - Hasta 120 pipelines
 - 512Mb de cache
 - 1Gb de RAM
 - 2 nucleos de CPU
 - 1 pipeline a la vez
 - 1 acción paralela a la vez
- Pro (75\$/mes):
 - Hasta 20 proyectos
 - Pipelines ilimitadas
 - 10Gb de cache
 - 3Gb de RAM
 - 2 nucleos de CPU
 - 1 pipeline a la vez
 - 1 acción paralela a la vez
- Hyper (200\$/mes):
 - Proyectos ilimitados
 - Pipelines ilimitadas
 - 20Gb de cache
 - 3Gb de RAM
 - 2 nucleos de CPU
 - 2 pipeline a la vez
 - 1 acción paralela a la vez
- On-premises (35\$/mes por usuario):

Todas las características del plan Hyper más:

 - Recursos customizados
 - Soporte empresarial
 - LDAP
 - Active Directory
 - Integración con GitHub EE y GitLab autogestionada

5. Buddy vs Circle CI

FEATURES	Buddy	CircleCI
Configuración del GUI	Sí	No
Configuración del YAML	Sí	Sí
Cloud	Sí	Sí
Soporte técnico	Sí	De pago
Compatibilidad con Docker	Sí	De pago
Entorno macOS	Sí	De pago
Sistemas de pago	Free: 0\$ Pro: 75\$ Hyper: 200\$ On-premises: 35\$/user	Free: 0\$ Performance: 15\$ Scale: 2000\$ Custom: ¿\$?

6. Buddy vs GitLab CI

FEATURES	Buddy	GitLabCI
Configuración del GUI	Sí	No
Configuración del YAML	Sí	Sí
Cloud	Sí	Sí
Soporte técnico	Sí	De pago
Compatibilidad con Docker	Sí	Sí
Entorno macOS	Sí	Sí
Sistemas de pago	Free: 0\$ Pro: 75\$ Hyper: 200\$ On-premises: 35\$/user	Free: 0\$ Premium: 19\$ Ultimate: 99\$ Add-ons: 10\$-60\$

7. Buddy vs Jenkins

FEATURES	Buddy	Jenkins
Configuración del GUI	Sí	Sí
Configuración del YAML	Sí	Plugin
Cloud	Sí	No
Soporte técnico	Sí	No
Compatibilidad con Docker	Sí	Sí
Selección de entorno	Sí	No
Sistemas de pago	Planes de pago	Totalmente gratuito

8. Conclusión

Buddy es una herramienta interesante con una curva de aprendizaje bastante ligera y que nos permite hacer pipelines de una manera muy cómoda, a su vez, la poca importancia que le dan en la documentación a la declaración de una pipeline de forma sintáctica y el énfasis que hacen en la utilización de la GUI, puede llevar a que nos resulte problemático trabajar sin ella o encontrar información sobre un posible fallo en las pipelines creada mediante .yaml.

Por otro lado, los tests se realizan en los servidores de Buddy, esto nos evita tener que consumir recursos propios y cargar de trabajo nuestros servidores pero, por otro lado, depender de la infraestructura de Buddy para estos test puede resultar problemático para algunos proyectos y empresas.

Para finalizar, el precio de su versión Pro y las características que nos ofrecen quizás resultan poco competitivas respecto a otros competidores, aunque su facilidad, el soporte técnico y la gran variedad en las plantillas de acciones que ofrecen son un gran factor diferenciador a tener en cuenta.