

Manual de Actualización

Docker + GitLab

Facturador PRO v1 & v2

Descripción

Como cliente posee soporte de actualizaciones durante una año una vez adquirida la aplicación, instalada o agregado al repositorio, si ha formado parte de este conjunto de clientes desde Enero quizá ya conozca los procedimientos de actualización mediante Docker, si en cambio es un cliente más antiguo las actualizaciones deben realizarse únicamente utilizando git ya que su instalación debió realizarse con Forge, sin embargo este documento está diseñado para que pueda actualizar su repositorio de manera independiente una vez culminado el tiempo asignado de soporte personalizado, será un conjunto de comandos muy prácticos de implementar.

Requisitos previos

1. Tener acceso a su servidor, vps, máquina virtual o local via SSH, en las instalaciones que realizamos para AWS o Google Cloud, hacemos entrega del usuario, la IP del servidor y la clave ssh que puede ser un archivo .ppk o .pem.
2. Tener instalado una versión de ssh en su máquina para conectarse de manera remota, puede utilizar putty, filezilla o una consola terminal.
3. Tener los accesos y credenciales de GitLab y de los entornos creados con Docker, usuario mysql, root entre otros que también son entregados una vez realizada la instalación, aunque una vez dentro de la instancia puede encontrarlos en los archivos .env y docker-compose.yml.

Pasos

1. Acceder a su instancia vía SSH (esta parte ya la deben conocer).
2. Loguearse como super usuario
ejecute **sudo su**
3. Ejecutar **docker ps** y verificar que estén al menos las siguientes líneas que equivalen a los contenedores de cada servicio iniciado por docker.

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
92f43f53f8e4	stenfrank/php:1.0	"php-fpm7.2"	18 hours ago	Up 18 hours	9000/tcp	multifactoralo_fpm2_1
914a35c72512	redis:alpine	"docker-entrypoint.s..."	18 hours ago	Up 18 hours	0.0.0.0:6381->6379/tcp	multifactoralo_redis2_1
de282cbbbc15	mariadb	"docker-entrypoint.s..."	18 hours ago	Up 18 hours	0.0.0.0:3307->3306/tcp	multifactoralo_mariadb2_1
eb049c8dba9c	stenfrank/nginx:1.0	"nginx"	18 hours ago	Up 18 hours	80/tcp	multifactoralo_nginx2_1

En la lista deberá fijarse en los **NAMES** que contengan **fpm** y **mariadb** ya que en ellos deberá acceder.

- **multifactoralo_fpm2_1** equivale al nombre del proyecto donde una vez accedido se encontrará justo en la ruta donde se ha alojado el repositorio y podrá ejecutar comandos **artisan**, **git** o **composer**. Este nombre puede variar un poco, puede llamarse **multifactoralnew_fpm1_1**.
- **multifactoralo_mariadb2_1** es el contenedor donde podrá acceder a mysql y utilizar comandos para ello, (desde aquí no se ejecutan comandos php, artisan, git, entre otros).

4. Una vez ubicado el contenedor a acceder ejecute el siguiente comando:

```
# docker exec -ti multifactoralnew_fpm1_1 /bin/bash
```

De esta manera se conectará al contenedor, observará un cambio en el inicio de la terminal.

5. Si ha accedido a **fpm** entonces se debe encontrar en la ruta `/var/www/html/` donde podrá ejecutar comandos de la siguiente lista (cada comando se ejecuta para situaciones distintas, mayormente no se necesitan ejecutar más de dos comandos en una actualización)

```
# git pull origin master
# git log -5
# php artisan migrate
# php artisan storage:link
# php artisan config:clear
# php artisan config:cache
# php artisan cache:clear
# php artisan tenancy:migrate
# composer update
# composer install
# composer dumpautoload
```

Si ha utilizado **git pull** debe ingresar las credenciales de GitLab, su correo y contraseña de acceso (la contraseña no se muestra en pantalla al ingresarla)

Debe hacer uso de los comandos según sea el caso, por ejemplo, posterior a un git pull que le muestre cambios en migraciones, deberá ejecutar `#php artisan migrate` y `#php artisan tenancy:migrate`, de esta manera actualizará los campos de las bases de datos, básicamente cada comando cumple una función específica y no todos son usados cada vez que se realiza una actualización.

6. Si ha accedido a mariadb podrá hacer uso de comandos mysql únicamente, al acceder deberá utilizar

```
# mysql -u root -p
```

le solicitará la contraseña la cual se encuentra en el archivo `docker-compose.yml` o `.env` dentro del proyecto, también le fue entregada al momento de instalar el proyecto en su servidor, si ha utilizado los canales de **Slack**, estos están mayormente en las chinchetas del canal.

Comandos extras

Para detener y restaurar los contenedores (servicios) dirijase a la ruta de su proyecto y ejecute los siguientes comandos

```
# docker-compose down
# docker-compose up -d
```

Otra manera de detener los contenedores puede ser individualmente con los siguientes comandos

```
# docker stop multifacturalo_fpm2_1
# docker start multifacturalo_fpm2_1
# docker restart multifacturalo_fpm2_1

docker restart multifacturalonew1_fpm1_1
```

Si desea respaldar o montar un respaldo de una bd puede ejecutar los siguientes comandos

```
# Backup
docker exec CONTAINER /usr/bin/mysqldump -u root --password=root DATABASE > backup.sql

# Restore
cat backup.sql | docker exec -i CONTAINER /usr/bin/mysql -u root --password=root DATABASE
```

Donde CONTAINER se refiere al nombre del contenedor multifacturalo_fpm2_1

POSIBLES ERRORES

Luego de una actualización donde utilice el comando “composer install” debe darle permisos a la carpeta en vendor del paquete de mpdf

```
# chmod -R 777 vendor/mpdf/mpdf/tmp
```