Water Data Project para Groots

Development - Repos & Dockerization

[**Alcance**](#_7s6m9f9z6bi2) **1**

[**Usuario y Organización**](#_wtxua8uoptop) **1**

[Organización](#_r1m7pbc6dgsp) 1

[**Repositorios DockerHub**](#_5l9yjgspgpbl) **1**

[**Repositorios GitHub**](#_9s30jcg5to6x) **2**

[**Dockerfiles**](#_pdl8qddoaxrf) **2**

[**Docker-Compose**](#_nzb0uukxq8ww) **2**

[**GitHub Actions**](#_45353p7hkkm4) **2**

# Alcance

Provee información acerca de la configuración de DockerHub y los elementos a considerar al momento de desarrollar las imagenes de docker subidas a DockerHub.

# Usuario y Organización

Se ha creado una cuenta en DockerHub.

* Usuario: nexiondev
* Email: [nexion-dev@nexion.com.bo](mailto:nexion-dev@nexion.com.bo)

### Organización

Se ha creado la organización:

* proagenda2030

# Repositorios DockerHub

General

<https://hub.docker.com/orgs/proagenda2030/repositories>

KPI:

<https://hub.docker.com/repository/docker/proagenda2030/kpi>

Support API:

<https://hub.docker.com/repository/docker/proagenda2030/support_api>

Support Bootstrap:

<https://hub.docker.com/repository/docker/proagenda2030/support_bootstrap>

Support Dashboards

<https://hub.docker.com/repository/docker/proagenda2030/dashboards>

# Repositorios GitHub

<https://github.com/Kobo-ProAgenda2030/kpi-custom-modules>

<https://github.com/Kobo-ProAgenda2030/kpi>

<https://github.com/Kobo-ProAgenda2030/kobo-docker>

<https://github.com/Kobo-ProAgenda2030/kobo-install>

<https://github.com/Kobo-ProAgenda2030/support-api>

<https://github.com/Kobo-ProAgenda2030/support-install>

<https://github.com/Kobo-ProAgenda2030/dashboards-docker>

https://github.com/Kobo-ProAgenda2030/dashboards-shiny

# Dockerfiles

<https://github.com/Kobo-ProAgenda2030/support-api/blob/dev/Dockerfile>

<https://github.com/Kobo-ProAgenda2030/support-api/blob/dev/Support.Bootstrap/bootstrap.Dockerfile>

https://github.com/Kobo-ProAgenda2030/dashboards-docker/blob/main/Dockerfile

# Docker-Compose

<https://github.com/Kobo-ProAgenda2030/kobo-docker/blob/2.020.45-proagenda2030/docker-compose.frontend.yml#L47>

<https://github.com/Kobo-ProAgenda2030/support-install/blob/main/deploy/templates/support-api-env/docker-compose.db.yml.tpl>

<https://github.com/Kobo-ProAgenda2030/support-install/blob/main/deploy/templates/support-api-env/docker-compose.frontend.yml.tpl>

<https://github.com/Kobo-ProAgenda2030/support-install/blob/main/deploy/templates/support-api-env/docker-compose.frontend.override.yml.tpl>

https://github.com/Kobo-ProAgenda2030/support-install/blob/main/deploy/templates/support-api-env/docker-compose.shiny.yml.tpl

# GitHub Actions

<https://github.com/Kobo-ProAgenda2030/kpi-custom-modules/blob/master/.github/workflows/release-package.yml>

<https://github.com/Kobo-ProAgenda2030/kpi/blob/proagenda2030/master/.github/workflows/push-to-dockerhub.yml>

<https://github.com/Kobo-ProAgenda2030/support-api/blob/dev/.github/workflows/build-push.yml>

<https://github.com/Kobo-ProAgenda2030/support-api/blob/dev/.github/workflows/build-push-bootstrap.yml>

<https://github.com/Kobo-ProAgenda2030/dashboards-docker/blob/main/.github/workflows/build-push.yml>

# Cómo actualizar los repositorios Fork

Los repositorios que se hicieron Fork del repositorio KoBoToolbox son los siguientes:

* <https://github.com/Kobo-ProAgenda2030/kpi>
* <https://github.com/Kobo-ProAgenda2030/kobo-docker>
* <https://github.com/Kobo-ProAgenda2030/kobo-install>

Es posible que pueda existir la necesidad de realizar la actualización de estos repositorios con el repositorio original. Por ejemplo, puede existir un bugfix que no está en el fork, ese bugfix se hizo en el repo de KoBoToolbox. Entonces necesitaremos realizar el merge correspondiente. Existen varias formas de hacer esto. A continuación se describe una manera que fue utilizada por el equipo de Nexion al hacer la actualización de la versión 2.020.45 a la versión de KoBoToolbox Master que vino del tag 2.021.12b.

Todas las operaciones son muy similares para los tres repositorios. Y se usará solamente línea de comandos. El procedimiento en general se describe de la siguiente manera:

* Se cambia el origen remoto a KoBoToolbox
* Se realiza el pull del repo
* Asumiendo que Master en KoBoToolbox es el que tiene el cambio que nos interesa, se hace el checkout de Master
* Pull de Master
* Se hace checkout del branch proagenda2030/master. El mismo se encuentra en la máquina local
* Se crea un nuevo branch a partir de éste. Usualmente se denomina feature/upgrade\_fork
* Checkout de feature/upgrade\_fork
* Merge de Master en el branch
  + Posiblemente se darán conflictos a resolver
* Abrir un editor de Git, por ejemplo Visual Studio Code y se resuelven todos los conflictos
* Se hace commit. NO SE HACE PUSH aún, igual dará un error si se intenta porque hay que recordar que estamos contra el remote origin de KoBoToolbox y no nuestro remote (Kobo-ProAgenda2030)
* Se cambia nuevamente el origin a nuestro remote (Kobo-ProAgenda2030)
* Se hace el push, ahora si ya es posible
* Vamos a Github y creamos un PR hacia el branch proagenda2030/master
* Se realizan las correcciones si aplican
* Merge y close del PR y eliminación del branch

## KPI

* Navegar a la carpeta del repositorio

|  |
| --- |
| git remote set-url origin git@github.com:kobotoolbox/kpi.git  git pull  git checkout master  git pull  git checkout proagenda2030/master  git branch feature/upgrade\_fork  git checkout feature/upgrade\_fork  git merge master |

* Realice todos los Fixes
* En Visual Studio Code realice los merges si hay conflictos
* En Visual Studio Code haga el commit

|  |
| --- |
| git remote set-url origin git@github.com:Kobo-ProAgenda2030/kpi.git  git push --set-upstream origin feature/upgrade\_fork |

* Crear el PR en github hacia proagenda2030/master
* Conseguir el approval
* Realizar el merge y eliminar el branch feature/upgrade\_fork

## kobo-docker

* Navegar a la carpeta del repositorio

|  |
| --- |
| git remote set-url origin git@github.com:kobotoolbox/kobo-docker.git  git pull  git checkout master  git pull  git checkout proagenda2030/master  git branch feature/upgrade\_fork  git checkout feature/upgrade\_fork  git merge master |

* Realice todos los Fixes
* En Visual Studio Code realice los merges si hay conflictos
* En Visual Studio Code haga el commit

|  |
| --- |
| git remote set-url origin git@github.com:Kobo-ProAgenda2030/kobo-docker.git  git push --set-upstream origin feature/upgrade\_fork |

* Crear el PR en github hacia proagenda2030/master
* Conseguir el approval
* Realizar el merge y eliminar el branch feature/upgrade\_fork

## kobo-install

* Navegar a la carpeta del repositorio

|  |
| --- |
| git remote set-url origin git@github.com:kobotoolbox/kobo-install.git  git pull  git checkout master  git pull  git checkout proagenda2030/master  git branch feature/upgrade\_fork  git checkout feature/upgrade\_fork  git merge master |

* Realice todos los Fixes
* En Visual Studio Code realice los merges si hay conflictos
* En Visual Studio Code haga el commit

|  |
| --- |
| git remote set-url origin git@github.com:Kobo-ProAgenda2030/kobo-install.git  git push --set-upstream origin feature/upgrade\_fork |

* Crear el PR en github hacia proagenda2030/master
* Conseguir el approval
* Realizar el merge y eliminar el branch feature/upgrade\_fork