

CREACIÓN AUTOMÁTICA DE IMÁGENES EN DOCKER HUB

JOSÉ DOMINGO MUÑOZ



IES GONZALO NAZARENO

FEBRERO 2021



CREACIÓN DE IMÁGGNES EN DOCKER HUB

La página Docker Hub nos ofrece un sistema de **integración continua** para la creación automática de imágenes. Lo que vamos a hacer es conectar nuestro repositorio de DockerHub con un repositorio de GitHub, donde tengamos el contexto para la construcción de la imagen (**Dcokerfile** y ficheros necesarios). Los pasos serán:



CONECTAMOS A LA CUENTA DE GITHUB (I)

Conectamos nuestra cuenta de Docker Hub con nuestra cuenta de GitHub. Para ello en la sección Account Settings - Linked Accounts:

The screenshot shows the 'Account Settings' page for user 'josedom24'. The 'Linked Accounts For Automated Builds' tab is selected. On the left, a sidebar lists settings: General, Linked Accounts (active), Security, Default Privacy, Notifications, Convert Account, and Deactivate Account. The main content area has a warning message at the top. Below it, the 'Linked Accounts' section explains that these links are for Automated Builds and includes a note that only one account can be connected at a time. A suggestion for using a service user is provided. At the bottom, a table lists providers and their connection status.

Provider	Account	Connect
GitHub	No account linked	Connect
Bitbucket	No account linked	Connect

Figura 1: Enlazamos a GitHub



CONECTAMOS A LA CUENTA DE GITHUB (II)

Pulsamos sobre Connect de GitHub, y autorizamos a DockerHub que acceda a nuestros repositorios:

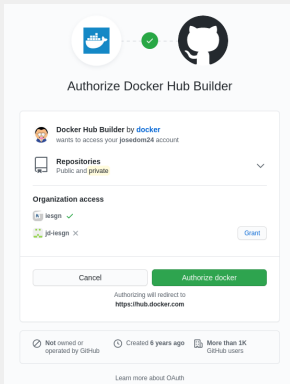


Figura 2: Damos permisos



CREAR UN REPOSITORIO DOCKER HUB

Crear un nuevo repositorio en Docker Hub: Ponemos un nombre y conectamos a un repositorio GitHub con el contexto.

The screenshot shows the Docker Hub interface for a repository named 'josedom24/docker_apache'. The page has a top navigation bar with tabs for 'General', 'Tags', 'Builds', 'Collaborators', 'Webhooks', and 'Settings'. The 'General' tab is active. The repository name 'josedom24/docker_apache' is displayed at the top, along with a 'Public View' button. Below the name, it says 'This repository does not have a description' and 'Last pushed: 7 minutes ago'. To the right, under 'Docker commands', there is a button with the command 'docker push josedom24/docker_apache:tagname'. The main content area is divided into two sections: 'Tags and Scans' and 'Recent builds'. The 'Tags and Scans' section shows a table with one tag, 'latest', which was pushed 7 minutes ago. The 'Recent builds' section shows three builds, all labeled 'Build in "master"', with the first one having a unique hash. At the bottom, there is a 'Readme' section with the text 'docker_apache2'.

TAG	OS	PULLED	PUSHED
latest		7 minutes ago	7 minutes ago

Build	Status
Build in "master" (44e5f56)	Success
Build in "master" (1bb86331)	Success
Build in "master" (1bb86331)	Success

Figura 3: Creamos un repositorio Docker Hub



En la sección Build del repositorio vemos las distintas build que vamos generando.

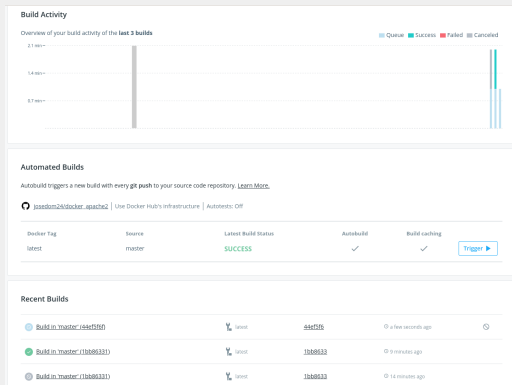


Figura 4: Sección build



BUILD AUTOMÁTICO

Cada vez que hagamos un push en el repositorio GitHub se creará un nuevo build y se creará una nueva imagen.

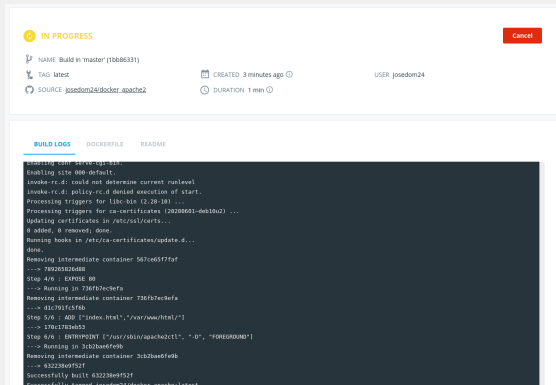


Figura 5: Build automático

