

IES JACARANDÁ



**Despliegue de
Aplicaciones Web**

**Guía para crear un
entorno de desarrollo
con Ruby on Rails**

Por: Chisela Marien Colás Gil

2 DAW

En esta práctica vamos a montar un entorno de desarrollo para el framework Rails, que usa el lenguaje de programación Ruby. “Ruby on Rails” es un framework que nos permite construir aplicaciones web con acceso a bases de datos.

Primero vamos a crearnos un directorio de trabajo (donde estará nuestra aplicación web) y cambiarnos a él con los siguientes comandos:

```
estudiante@DAW1:~$ mkdir rubyrails
estudiante@DAW1:~$ cd rubyrails
estudiante@DAW1:~/rubyrails$
```

En este directorio vamos a crear el contenedor de desarrollo de Bitnami con el archivo docker-compose.yml oficial, que obtenemos con el siguiente comando:

`$ curl -LO`

`https://raw.githubusercontent.com/bitnami/containers/main/bitnami/rails/docker-compose.yml`

```
estudiante@DAW1:~/rubyrails$ curl -LO https://raw.githubusercontent.com/bitnami/containers/main/bitnami/rails/docker-compose.yml
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           %             %         Dload  Upload  Total   Spent    Left   Speed
100  431  100  431    0     0  3063      0  --:--:-- --:--:-- --:--:--  3078
```

Este fichero .yml guarda la configuración necesaria para crear un contenedor con una base de datos mariadb y el framework rails. En este caso queremos que el puerto a utilizar sea el 8001 así que tendremos que cambiarlo en este fichero. En la siguiente imagen se puede ver el resultado final explicando qué hace cada línea.

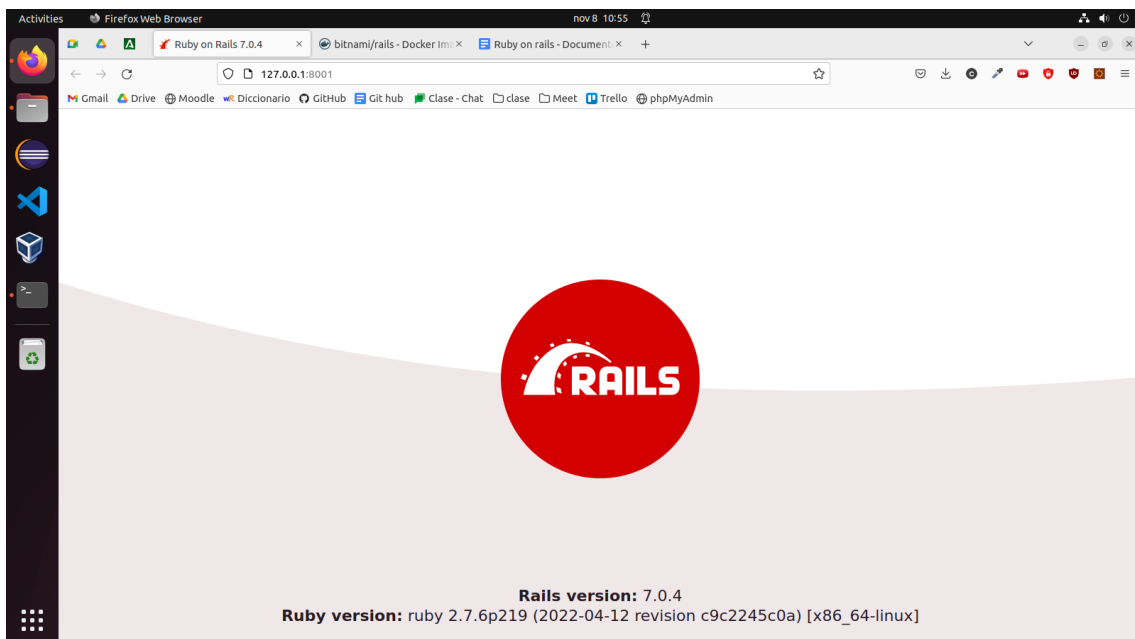


```
1 #indica la versión
2 version: '2'
3
4 #lista los servicios que se van a emplear
5 services:
6   #el primer servicio es una base de datos mariadb
7   mariadb:
8     #la imagen necesaria para crear el contenedor de mariadb
9     image: docker.io/bitnami/mariadb:10.6
10    #lista las variables de entorno para este contenedor
11    environment:
12      #variable que permite usar una contraseña vacía, solo recomendable para desarrollo
13      - ALLOW_EMPTY_PASSWORD=yes
14  #el segundo servicio será nuestra aplicación, el framework rails
15  myapp:
16    #la imagen necesaria para crear el contenedor de rails
17    image: docker.io/bitnami/rails:7
18    #los puertos que se van a mapear, en nuestro caso aquí se cambia el 3000 por el 8001
19    ports:
20      - '8001:3000'
21    #lista las variables de entorno para este contenedor
22    environment:
23      #host de la base de datos, en nuestro caso mariadb
24      - DATABASE_HOST=mariadb
25      #nombre de la base de datos
26      - DATABASE_NAME=my_app_development
27    #lista de volúmenes del contenedor
28    volumes:
29      #el directorio donde se encuentra el proyecto y la aplicación de ruby en rails
30      - './my-project:/app'
31    #este servicio depende de mariadb, por lo que se debe iniciar después de mariadb
32    depends_on:
33      - mariadb
```

Una vez obtenido modificado y entendido este archivo, solo nos queda levantar el contenedor con *docker-compose up*.

```
Activities Terminal nov 8 10:41
estudiante@DAW1: ~/rubyrails
estudiante@DAW1:~/rubyrails$ docker-compose up
Creating network "rubyrails_default" with the default driver
Pulling mariadb (docker.io/bitnami/mariadb:10.6)...
10.6: Pulling from bitnami/mariadb
c1a0533acd82: Pull complete
7430f56cc01c: Pull complete
Digest: sha256:09fe34a1ebc0bfbf6a6f2b48eb2ae8d5db4447749706c0df018a8bfdda7d8
Status: Downloaded newer image for bitnami/mariadb:10.6
Pulling nyapp (docker.io/bitnami/rails:7)...
7: Pulling from bitnami/rails
c1a0533acd82: Already exists
a60181e229e2: Pull complete
Digest: sha256:e79a1a4aacc486e170dac081092b2e6ca524d3d2f38d0af648832ea0e9fd3eb7
Status: Downloaded newer image for bitnami/rails:7
Creating rubyrails_mariadb_1 ... done
Creating rubyrails_myapp_1 ... done
Attaching to rubyrails_mariadb_1, rubyrails_myapp_1
mariadb_1 | mariadb 09:39:15.12
nyapp_1 | rails 09:39:15.64
mariadb_1 | mariadb 09:39:15.13 Welcome to the Bitnami mariadb container
nyapp_1 | rails 09:39:15.64 Welcome to the Bitnami rails container
mariadb_1 | mariadb 09:39:15.13 Subscribe to project updates by watching https://github.com/bitnami/containers
nyapp_1 | rails 09:39:15.13 Submit issues and feature requests at https://github.com/bitnami/containers/issues
mariadb_1 | mariadb 09:39:15.13 INFO ==> ** Starting MariaDB setup **
nyapp_1 | rails 09:39:15.17 INFO ==> Validating settings in MYSQL_* env vars
mariadb_1 | mariadb 09:39:15.17 INFO ==> Validating settings in MYSQL_* env vars
nyapp_1 | rails 09:39:15.17 WARN ==> You set the environment variable ALLOW_EMPTY_PASSWORD=yes. For safety reasons, do not use this flag in a production environne
mariadb_1 | mariadb 09:39:15.21 INFO ==> Initializing mariadb database
nyapp_1 | rails 09:39:15.22 INFO ==> Updating 'my.cnf' with custom configuration
mariadb_1 | mariadb 09:39:15.22 INFO ==> Updating 'my.cnf' with custom configuration
nyapp_1 | rails 09:39:15.65 Submit issues and feature requests at https://github.com/bitnami/containers/issues
mariadb_1 | mariadb 09:39:15.23 INFO ==> Setting slow_query_log option
nyapp_1 | rails 09:39:15.23 INFO ==> Setting slow_query_log option
mariadb_1 | mariadb 09:39:15.30 INFO ==> Setting long_query_time option
nyapp_1 | rails 09:39:15.31 INFO ==> Installing database
mariadb_1 | mariadb 09:39:15.65 INFO ==> ** Running Rails setup **
nyapp_1 | rails 09:39:15.68 INFO ==> Creating new Rails project
nyapp_1 | exist
nyapp_1 | create README.md
nyapp_1 | create Rakefile
nyapp_1 | create .ruby-version
nyapp_1 | create config.ru
nyapp_1 | create .gitignore
nyapp_1 | create gitattributes
nyapp_1 | create Gemfile
```

Y de esta manera abrimos nuestro navegador y ya tenemos nuestra aplicación activa a través de <http://127.0.0.1:8001>.



Página oficial de la imagen bitnami para rails:

<https://hub.docker.com/r/bitnami/rails/>