

variables de ambiente: application.yml

Docker

Ejemplo:  
msvc-security

```
server:
  port: ${PORT:9100}
msvc:
  tracks:
    url: ${TRACKS_URL:http://localhost:8082}
  albums:
    url: ${ALBUMS_URL:http://localhost:8081}
spring:
  application:
    name: msvc-security
  datasource:
    url: jdbc:mysql://${DB_HOST:localhost:3306}/${DB_DATABASE:db_users_springboot}
    username: ${DB_USERNAME:root}
    password: ${DB_PASSWORD:sasa}
    driver-class-name: com.mysql.cj.jdbc.Driver
  jpa:
    database-platform: org.hibernate.dialect.MySQLDialect
    show-sql: true
    hibernate:
      ddl-auto: update
  servlet:
    multipart:
      max-file-size: 10MB
  logging:
    level:
      org:
        hibernate:
          SQL: debug
eureka:
  instance:
```

## variables de ambiente: clientFeingRest

Docker

Ejemplo:  
msvc-security

```
usersapp > usersapp > clients > AlbumClientRest
AlbumClientRest.java
package com.theinsideshine.backend.usersapp.usersapp.clients;

import ...

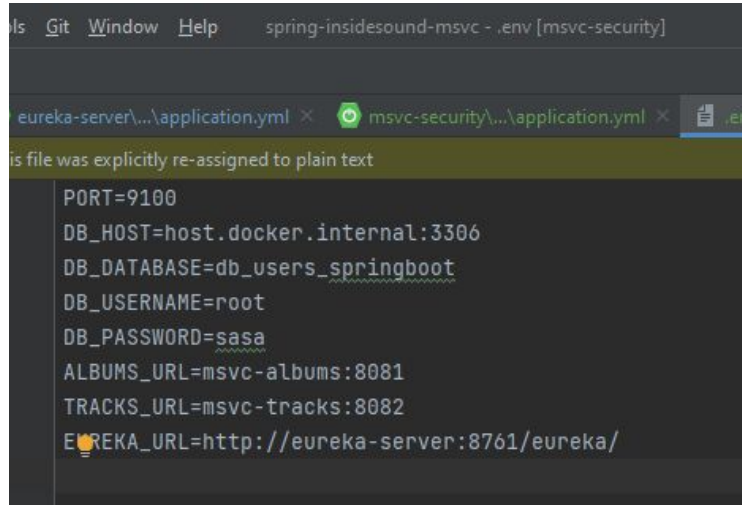
2 usages  theinsideshine *
@FeignClient(name="msvc-albums",url="${msvc.albums.url}")
public interface AlbumClientRest {

    1 usage  theinsideshine
    @DeleteMapping("username/{username}")
    ResponseEntity<?> removeAlbumsByUsername(@PathVariable String username);
}
```

```
TrackClientRest.java
1 package com.theinsideshine.backend.usersapp.usersapp.clients;
2
3 import ...
4
5
6
7
8
9 2 usages  theinsideshine *
@FeignClient(name = "msvc-tracks", url = "${msvc.tracks.url}")
10 public interface TrackClientRest {
11
12     1 usage  theinsideshine
13     @DeleteMapping("username/{username}")
14     ResponseEntity<?> removeTracksByUsername(@PathVariable String username);
15 }
```

archivo .env para construir imagen

Docker

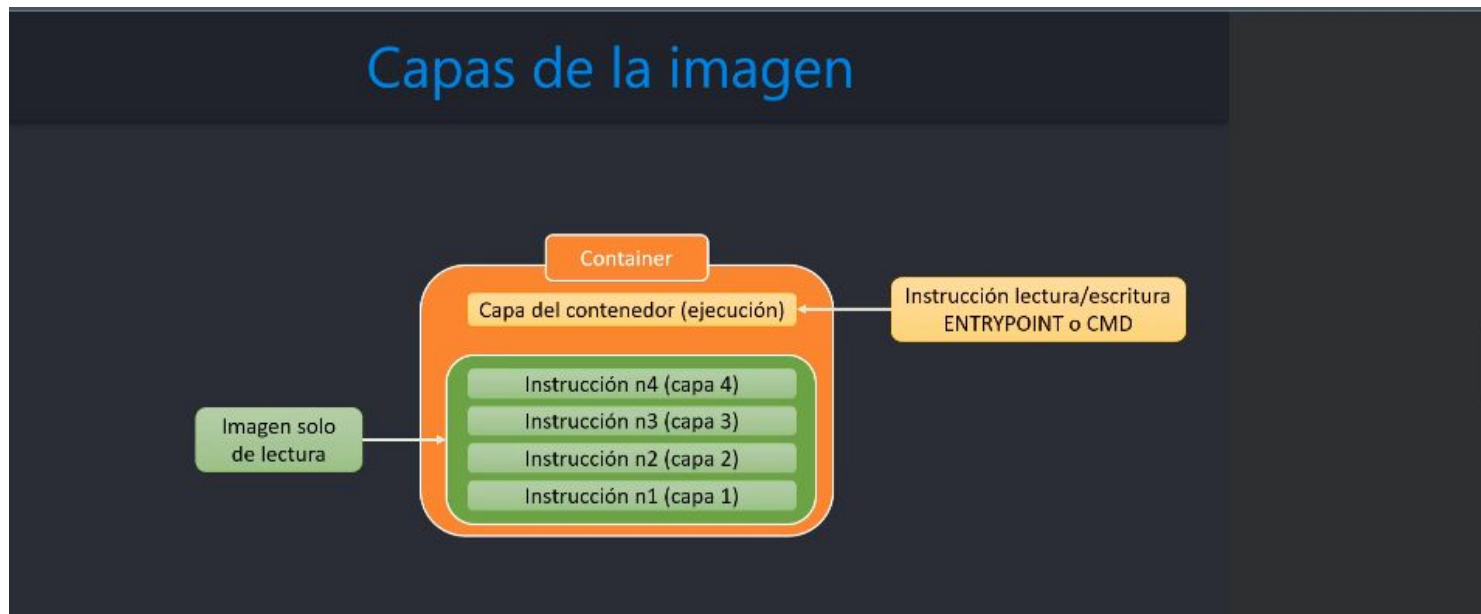


The screenshot shows a code editor window with the title bar "spring-insidesound-msvc - .env [msvc-security]". The editor has three tabs: "eureka-server\...\application.yml", "msvc-security\...\application.yml", and ".env". The ".env" tab is active, showing the following configuration:

```
PORT=9100
DB_HOST=host.docker.internal:3306
DB_DATABASE=db_users_springboot
DB_USERNAME=root
DB_PASSWORD=sasa
ALBUMS_URL=msvc-albums:8081
TRACKS_URL=msvc-tracks:8082
EUREKA_URL=http://eureka-server:8761/eureka/
```

A message at the top of the editor states: "this file was explicitly re-assigned to plain text".

Ejemplo:  
msvc-security

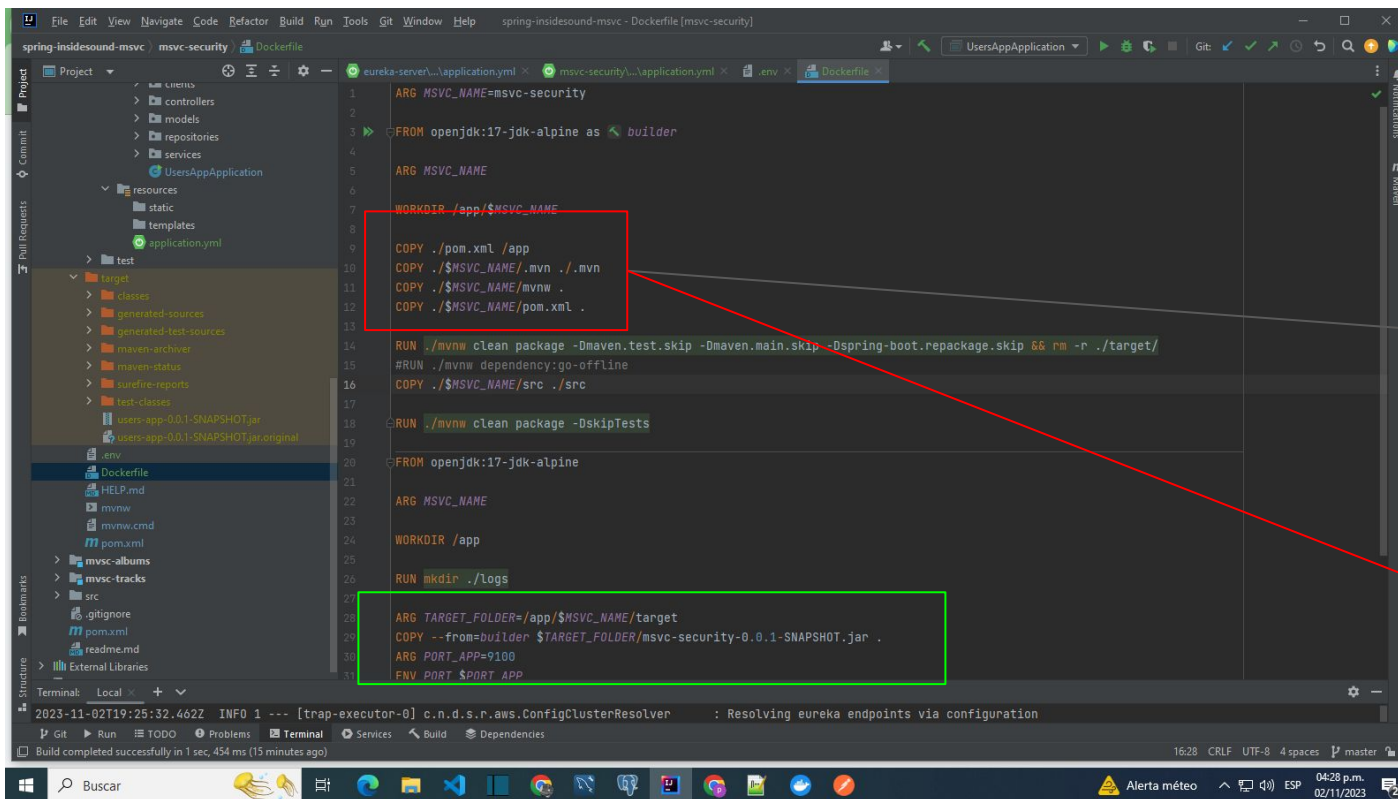


las capas se almacenan en el cache de docker, si cambia una capa se vuelve a ejecutar desde es capa hacia adelante

# Creando la imagen:Dockerfile

Docker

Ejemplo:  
msvc-security



```
1 ARG MSVC_NAME=msvc-security
2
3 FROM openjdk:17-jdk-alpine as builder
4
5 ARG MSVC_NAME
6
7 WORKDIR /app/${MSVC_NAME}
8
9 COPY ./pom.xml /app
10 COPY ./${MSVC_NAME}/.mvn ./
11 COPY ./${MSVC_NAME}/mvnw .
12 COPY ./${MSVC_NAME}/pom.xml .
13
14 RUN ./mvnw clean package -Dmaven.test.skip -Dmaven.main.skip -Dspring-boot.repackage.skip && rm -r ./target/
15 #RUN ./mvnw dependency:go-offline
16 COPY ./${MSVC_NAME}/src ./src
17
18 RUN ./mvnw clean package -DskipTests
19
20 FROM openjdk:17-jdk-alpine
21
22 ARG MSVC_NAME
23
24 WORKDIR /app
25
26 RUN mkdir ./logs
27
28 ARG TARGET_FOLDER=/app/${MSVC_NAME}/target
29 COPY --from=builder $TARGET_FOLDER/msvc-security-0.0.1-SNAPSHOT.jar .
30 ARG PORT_APP=9100
31 ENV PORT $PORT_APP
```

Annotations in the image:

- A red box highlights lines 9-12 (COPY commands).
- A red arrow points from the text "mvn wrapper:wrapper" to the red box.
- A green box highlights lines 28-31 (COPY and ENV commands).
- A red arrow points from the text "descarga las dependencia una sola vez si no hay cambios" to the green box.

mvn wrapper:wrapper

descarga las  
dependencia una sola  
vez si no hay  
cambios

Eureka server: dependiendo donde se levante la red es diferente

eureka-server

```
application.yml
eureka-server\...\application.yml
msvc-security\...\application.yml

server:
  port: ${PORT:8761}

spring:
  application:
    name: eureka-server

eureka:
  instance:
    prefer-ip-address: true
    hostname: localhost

  client:
    registerWithEureka: false
    fetchRegistry: false
```

Ejemplo:  
msvc-security

```
eureka-server\...\application.yml
msvc-security\...\application.yml
.env

hibernate:
  ddl-auto: update
servlet:
  multipart:
    max-file-size: 10MB
logging:
  level:
    org:
      hibernate:
        SQL: debug
eureka:
  client:
    serviceUrl:
      defaultZone: ${EUREKA_URL:http://localhost:8761/eureka/}
```

se levanta  
localmente

```
Git Window Help spring-insidesound-msvc - .env [msvc-security]
eureka-server\...\application.yml
msvc-security\...\application.yml
.env
This file was explicitly re-assigned to plain text
PORT=9100
DB_HOST=host.docker.internal:3306
DB_DATABASE=db_users_springboot
DB_USERNAME=root
DB_PASSWORD=sasa
ALBUMS_URL=msvc-albums:8081
TRACKS_URL=msvc-tracks:8082
EUREKA_URL=http://eureka-server:8761/eureka/
```

Se levanta con  
docker usando  
.env

Docker

docker run -p 8761:8761 --env-file .\eureka-server\.env -d --rm --name eureka-server --network spring eureka

Creando la imagen: desde  
D:\proyectos\insideSound\spring-insidesound-msvc>

Docker

### Eureka

```
docker build -t eureka . -f .\eureka-server\Dockerfile
docker network create spring
docker run -p 8761:8761 --env-file .\eureka-server\.env -d --rm --name eureka-server --network spring eureka
```

### Security

```
docker build -t security . -f .\msvc-security\Dockerfile
docker run -p 9100:9100 --env-file .\msvc-security\.env --rm --name msvc-security --network spring security
```

### Albums

```
docker build -t albums . -f .\msvc-albums\Dockerfile
docker run -p 8081:8081 --env-file .\msvc-albums\.env --rm --name msvc-albums --network spring albums
```

### Tracks

```
docker build -t tracks . -f .\msvc-tracks\Dockerfile
docker run -p 8082:8082 --env-file .\msvc-tracks\.env --rm --name msvc-tracks --network spring tracks
```

Creando la imagen: desde

D:\proyectos\insideSound\spring-insidesound-msvc>

Docker

Gateway

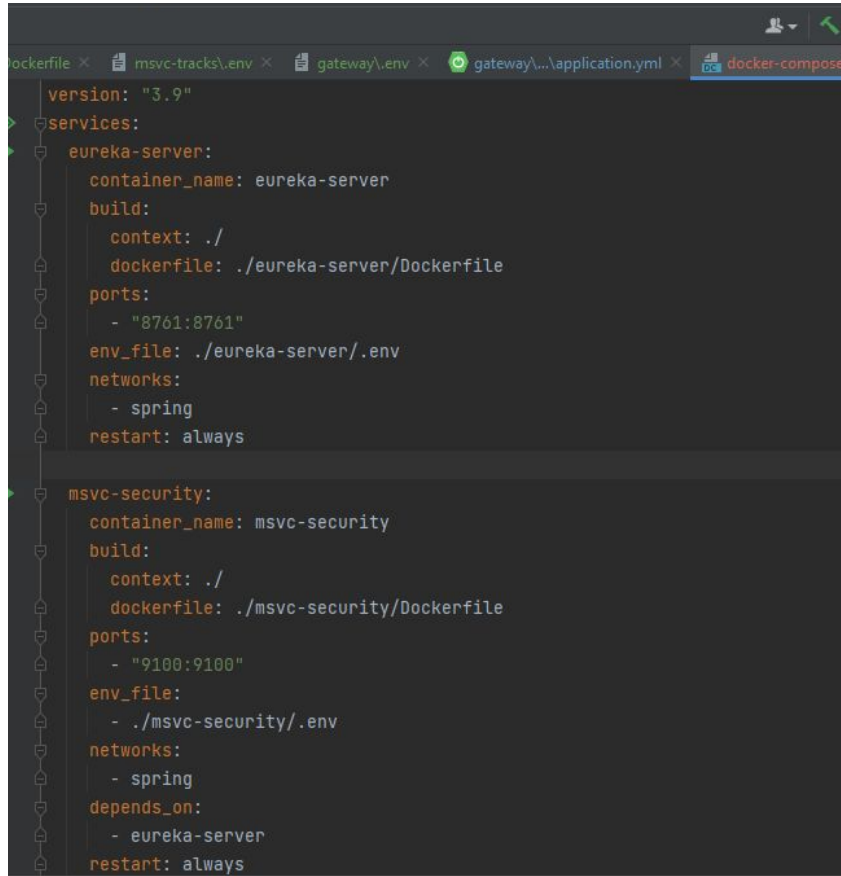
```
docker build -t gateway . -f .\gateway\Dockerfile
```

```
docker run -p 8090:8090 --env-file .\gateway\.env --rm --name gateway --network spring gateway
```



# Docker-compose

Docker

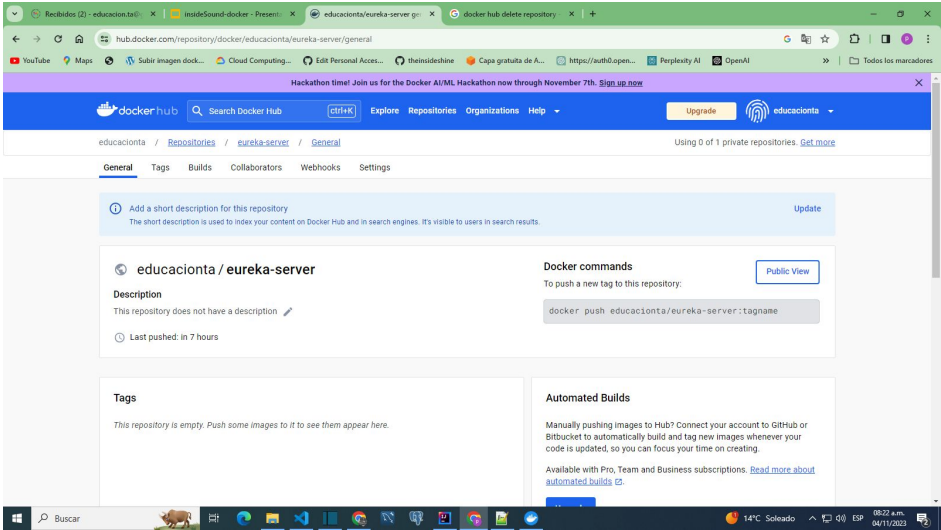


```
version: "3.9"
services:
  eureka-server:
    container_name: eureka-server
    build:
      context: ./
      dockerfile: ./eureka-server/Dockerfile
    ports:
      - "8761:8761"
    env_file: ./eureka-server/.env
    networks:
      - spring
    restart: always

  msvc-security:
    container_name: msvc-security
    build:
      context: ./
      dockerfile: ./msvc-security/Dockerfile
    ports:
      - "9100:9100"
    env_file:
      - ./msvc-security/.env
    networks:
      - spring
    depends_on:
      - eureka-server
    restart: always
```

# Dockerhub

## Docker



Copyright (C) Microsoft Corporation. Todos los derechos reservados.

Prueba la nueva tecnología PowerShell multiplataforma <https://aka.ms/pscore6>

```
PS D:\proyectos\insideSound\spring-insidesound-msvc> docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
spring-insidesound-msvc-security   latest             896934a1e8a4       Less than a second ago   398MB
spring-insidesound-msvc-albums     latest             7609ed0c490a       Less than a second ago   393MB
spring-insidesound-msvc-eureka-server latest             184742e5e199       Less than a second ago   378MB
spring-insidesound-msvc-gateway    latest             8534f74f5772       16 hours ago           374MB
spring-insidesound-msvc-msvc-tracks latest             112bb480548f       16 hours ago           393MB
PS D:\proyectos\insideSound\spring-insidesound-msvc> docker tag spring-insidesound-msvc-eureka-server educacionta/eureka-server
PS D:\proyectos\insideSound\spring-insidesound-msvc>
```

docker tag spring-insidesound-msvc-eureka-server educacionta/eureka-server  
docker login  
docker push educacionta/eureka-server

The screenshot shows a web browser window with multiple tabs. The active tab is 'educacionta/eureka-server general' on Docker Hub. The page displays the repository details for 'educacionta / eureka-server'. It includes a description section stating 'This repository does not have a description' and 'Last pushed: in 8 hours'. There is a 'Tags' section showing one tag, 'latest', which is an 'Image' type, pushed 'in 8 hours'. A 'Docker commands' section shows the command 'docker push educacionta/eureka-server:tagname'. An 'Automated Builds' section explains how to connect GitHub or Bitbucket for automatic builds. At the bottom, there is a 'Repository overview' section. The browser's address bar shows 'hub.docker.com/repository/docker/educacionta/eureka-server/general'. The Windows taskbar at the bottom shows the time as 08:25 a.m. on 04/11/2023, with a temperature of 14°C and weather 'Soleado'.

Recibidos (2) - educacion.ta@... | insideSound-docker - Presenta... | educacionta/eureka-server general | docker hub delete repository | +

hub.docker.com/repository/docker/educacionta/eureka-server/general

YouTube | Maps | Subir imagen dock... | Cloud Computing... | Edit Personal Acces... | theinsideshine | Capa gratuita de A... | https://auth0.open... | Perplexity AI | OpenAI | Todos los marcadores

**Add a short description for this repository** [Update](#)

The short description is used to index your content on Docker Hub and in search engines. It's visible to users in search results.

**educacionta / eureka-server**

**Description**

This repository does not have a description [✎](#)

Last pushed: in 8 hours

**Docker commands** [Public View](#)

To push a new tag to this repository:

```
docker push educacionta/eureka-server:tagname
```

**Tags**

This repository contains 1 tag(s).

Tag	OS	Type	Pulled	Pushed
latest		Image	---	in 8 hours

[See all](#) [Go to Advanced Image Management](#)

**Automated Builds**

Manually pushing images to Hub? Connect your account to GitHub or Bitbucket to automatically build and tag new images whenever your code is updated, so you can focus your time on creating.

Available with Pro, Team and Business subscriptions. [Read more about automated builds](#)

[Upgrade](#)

**Repository overview**

An overview describes what your image does and how to run it. It displays in [the public view of your repository](#).

```
Dockerfile x .env x .env x application.yml x docker-compose.yml
1   version: "3.9"
2   services:
3     eureka-server:
4       container_name: eureka-server
5       image: educacionta/eureka-server:latest
6     # build:
7     #   context: ./
8     #   dockerfile: ./eureka-server/Dockerfile
9     ports:
10      - "8761:8761"
11     env_file: ./eureka-server/.env
12     networks:
13       - spring
14     restart: always
15
```