

Informazio Sistemen Segurtasuna Kudeatzeko Sistemak 2025/2026

0 laborategia – Docker

Laborategi hau ez da azterketan sartuko baina beharrezkoa da proiektua burutzeko.

Helburuak:

- 1. Docker osatzen duten elementuen oinarrizko jakintza lortzea: irudiak, edukiontziak, biltegiak, bolumenak, ...
- 2. Zerbitzu talde baten oinarrizko hedapena docker-compose bidez.

Baliabideak:

- Ubuntu (Erabiltzailea/pasahitza: Isi/Isi).
- Egelan dauden Docker elementuak.
- · GitHub-en dagoen docker-compose oinarrizko proiektua.

1.- Sarrera

Docker GNU/Linux-erako "birtualizazio-azpiegitura" da, kontainerretan oinarritua¹. Oso tresna ezaguna da, jatorrizko ingurunearekiko berdinak diren zerbitzuak hedatzeko erabiltzen dena, horrela "Nire ordenagailuan badabil" ospetsua saihestuz (Baina zure bezeroarenean ez).

Docker-en elementu nagusia irudia da: zerbitzuak funtzionatzeko behar duen guztia (sistema, bitarrak, liburutegiak, fitxategiak...) jasotzen duen artxibo konprimitu aldaezina. Irudi batetik abiatuta kontainerrak sor daitezke, hau da, zerbitzuaren exekuzio isolatuak, iragankorrak eta aldakorrak. Dockerrek irudi-biltegiak sortzeko aukera ere ematen du: biltegi ofizialak zerbitzu espezifikoak eraikitzeko berrerabil daitezkeen irudi ofizialak ditu.

2.- Instalazioa eta konfigurazioa

Docker instalatzeko: \$ apt install docker.io

Docker-ek root sarbidea behar du. Sudo ez erabiltzeko:

- docker taldea sortu: \$ sudo groupadd docker
- Oraingo erabiltzailea docker taldeari gehitu: \$ sudo usermod -aG docker \$USER
- Sistema berabiarazi, berriro sartu, eta docker exekutatu: \$ docker run hello-world

¹ VirtualBox bezalako birtualizazio-tresnek ez bezala, Dockerren motorrak kontainerren eta sistema eragilearen arteko tarteko geruza sortzen du, eta, beraz, ez da sistema eragile "osoa" birtualizatu behar, kontainerrak askoz arinagoak eginez.



Informazio Sistemen Segurtasuna Kudeatzeko Sistemak 2025/2026

```
[mikel@Durruti ~]$ docker run hello-world

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
1. The Docker client contacted the Docker daemon.
2. The Docker daemon pulled the "hello-world" image from the Docker Hub. (amd64)
3. The Docker daemon created a new container from that image which runs the executable that produces the output you are currently reading.
4. The Docker daemon streamed that output to the Docker client, which sent it to your terminal.

To try something more ambitious, you can run an Ubuntu container with: $ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID: https://hub.docker.com/
For more examples and ideas, visit: https://docs.docker.com/get-started/
[mikel@Durruti ~]$ ■
```

3.- Irudiak kudeatu

Docker-ek biltegi lokala du, gure ordenagailuan erabiliko ditugun irudiekin.

> Zure ordenagailuko biltegian dauden irudiak ikusteko: \$ docker images

Urruneko biltegirik famatuena Docker Hub² da eta lehenetsia dago Docker³ instalatzean.

Docker Hub-en dauden irudiak begiratu.

Urruneko biltegitik biltegi lokalera irudi bat jeitsiko dugu:

- hello-world docker irudiaren esteka sakatu.
- Orrialdean adierazten duenez, jeitsi irudia biltegi lokalera: \$ docker pull hello-world

Zein komando erabili behar da irudiak gure biltegi lokaletik Docker Hub-era igotzeko?

4.- Kontainerrak exekutatu

Edukiontziak irudi batetik exekutatzen dira.

hello-world irudia erabiliz edukiontzi bat exekutatu: \$ docker run hello-world

Zein emaitza ematen digu?

Exekutatu \$ docker run -it ubuntu bash

Nondik atera da ubuntu irudia?

² http://hub.docker.com

³ Enpresa askok beraien Docker biltegiak dituzte, bezeroetan hedapena errazte aldera.



Informazio Sistemen Segurtasuna Kudeatzeko Sistemak 2025/2026

Zein ezberdintasun dago docker run eta docker run -it artean?

Zergatik aldatu da terminala?

Is bidez zerrenda bat egiten badugu, zein makinari dagokizkio fitxategiak?

Zein emaitza ematen digu docker ps -a komandoak?

Edukiontziak eteteko, izena edo id-a behar dugu:

Exekutatu \$ docker kill izena|id (Gero exekutatu \$ docker ps -a)

Nahiz eta edukiontziak ez ibili, ezabatu behar dira:

> Exekutatu \$ docker rm izena|id (Gero exekutatu \$ docker ps -a eta \$ docker images)

Ze ezberdintasun dago edukiontzia eten eta ezabatu artean?

Zein eragin dauka edukiontziak edukiontzia sortu duen irudiarengan?

Zelan ezabatzen dira irudiak?

Berriro ere exekutatu edukiontzi bat **ubuntu** iruditik abiatuz:

- \$ docker run -it ubuntu bash
- Beste terminal batean, exekutatu \$ docker exec izena_edukiontzi ls

Zein ezberdintasun dago run eta exec artean?

5.- Irudiak eraiki

Docker irudi bat eraikitzeko **Dockerfile** bat behar dugu. Dockerfile-ren bidez instrukzioak ematen dizkiogu Docker-ri irudia eraikitzeko. Adibidez:

- FROM: erabiliko den oinarrizko irudia (Bere "gainean" eraikitzeko).
- ADD: irudiari artxibo lokalak gehitzen dizkio.
- RUN: komandoak exekutatu.
- FROM <u>ubuntu</u>
 ADD msg /opt/
 RUN apt-get update && apt-get install less
 RUN date >> /opt/msg
 CMD ["cat"."/opt/msg"]

CMD: irudiak definitzen duen edukiontzia ibiltzen hasten denean exekutatuko den komandoa.

Irudi honen araberako edukiontzia exekutatzen dugunean, zein emaitza lortuko dugu? Zergatik?

Egela-n dagoen Dockerfile-tik hasita irudia eraikiko dugu:

- Jaitsi Dockerfile eGela-tik, msg artxiboarekin (Mezu bat du).
- > exekutatu \$ docker build -t="izena" . direktorio berean (Izena edozein izan daiteke).
- Irudia eraikia izan dela eta biltegira gehitua izan dela konprobatu: \$ docker images
- Eraiki berri dugun irudia jatorri duen edukiontzia exekutatu: \$ docker run izena

Edukiontzia exekutatzean zein emaitza lortzen da?



Informazio Sistemen Segurtasuna Kudeatzeko Sistemak 2025/2026

Nola aldatuko zenuke lortzen den mezua?

Docker-ekin host eta edukiontziaren artean direktorioak elkarbanatu ahal ditugu, eta horrela biek irakurri eta idatzi dezakete elkarbanatutako direktoriotan:

- dir-msg izeneko direktorio bat sortu, msg2 izeneko artxibo bat duena, "iep" katearekin: \$ mkdir dir-msg && echo "iep" > dir-msg/msg2
- Exekutatu edukiontzi bat ubuntu iruditik, interaktiboki, dir-msg direktorioa edukiontzi barruan /app direktorioan muntatuz: \$ docker run -it -v "\$(pwd)"/dir-msg:/app ubuntu bash
- Edukiontzi barruan zaudela, ondo montatu dela berrikusi: \$ cat app/msg2
- ➤ Beste terminal batean, dir-msg/msg2-ren edukia aldatu eta berriro exekutatu \$ cat app/msg2 edukiontzi barnean.

Artxiboaren edukia aldatu al da?

Zergatik?

Edukiontziak, definizioz, iragankorrak eta bizitza laburrekoak dira. Beraz, bolumenek Docker-en berebiziko garrantzia dute, datuak gordetzeko aukera ematen baitigute, nahiz eta edukiontzia akatu.

Docker-en bidez hedatuko diren aplikazioak garatzerakoan ohiko prozesua horrelakoa izaten da:

- 1. Edukiontzian garapen direktorioa eta datu direktorioa muntatu.
- 2. Garatu eta testak egin.
- 3. Aplikazioa prest dagoenean Dockerfile-ra gehitu.

6.- Zerbitzuak exekutatu

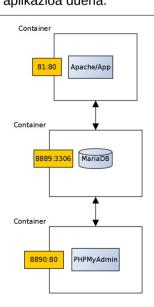
Docker-compose erabiliz zerbitzu taldeak definitzea daukagu, era koordinatuan exekutatu daitezen, zerbitzu bakoitza Docker irudi batean oinarritzen delarik. Ezaugarri hau oso garrantzitsua da, kontutan harturik gaur egun aplikazio asko zerbitzu taldeen bidez inplementatzen direla.

Docker-compose-ri buruz ikasteko 1 lanerako oinarri gixa erabili dezakezuen proiektua begiratuko dugu. Proiektuak hiru zerbitzu dauzka, Web sistema oso sinplea osatuz: Web zerbitzaria PHP aplikazioarekin, Maria Datu Basea, eta Maria DB kudeatzeko beste web zerbitzari bat PHPMyAdmin aplikazioa duena.

Proiektua daukan GitHub biltegia klonatu (Edo jaitsi): \$ git clone https://github.com/mikel-egana-aranguren/dockerlamp.git

Docker-lamp/docker-compose.yml artxiboak zerbitzuen definizioak dauzka. Kasu honetan hiru zerbitzu dauzkagu (Zerbitzuaren izena eta oinarri duen Docker irudiarena ezberdinak izan daitezke):

 web: zerbitzu honek "web" iruditik edukiontzia exekutatzen du. Web irudia Dockerfile-ren bidez eraikita dago, Apache Web zerbitzaria eta PHP aplikazioa /app direktorioan definituak dituena (Irudi hau PHP irudi





Informazio Sistemen Segurtasuna Kudeatzeko Sistemak 2025/2026

ofizialaren moldaketa besterik ez da⁴). **db** zerbitzuari lotzen zaio eta trafikoa 81 portutik (host) 80 portura bidaltzen du (Non Apache exekutatzen den).

- **db**: kasu honetan Maria DB irudi ofiziala erabiltzen da, moldatu gabe. Zerbitzua datuak ./mysql bolumenetik lortuz exekutatzen da (Alegia datuak gordetzen dira). "Environment" atalean konfigurazioa definitzen da, eta 8889 (host) portuaren trafikoa edukiontziaren 3306 portura bidaltzen du.
- phpmyadmin: kasu honetan PHPMyAdmin, irudi ofiziala erabiltzen da, moldatu gabe. db zerbitzura konektatzen da eta datu basea kudeatzeko erabiltzen da

Proiektua abian jartzeko:

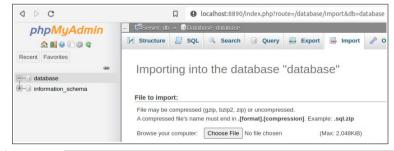
- > Terminala /docker-lamp direktorioan ipini.
- > web irudia eraiki: \$ docker build -t="web" .
- \$ docker-compose up bidez zerbitzuak abian jarri⁵.
- http://localhost:81 begiratu.
- Beharrezko datuak sartzeko, begiratu http://localhost:8890/



(docker-compose.yml-en definitu dugunez, "admin" erabiltzailea, "test" pasahitza). "database" sakatu eta gero "import", hor docker-lamp/database.sql artxiboa aukeratu.







Begiratu http://localhost:81, informazio gehiago eduki beharko luke.

Zerbitzuak eteteko, beste terminal batean, \$ docker-compose down Yeah, it works!

1 mikel 2 aitor

⁴ Docker-en ezaugarri garrantzitsuenetakoa Dockerfile-an oinarrizko irudia extensio bidez eraldatzea da.

⁵ Behar izatekotan, docker-compose \$ sudo apt install docker-compose bidez instalatu.