

# Introducció a Docker per a desenvolupadors

Unitat 3. Sistemes de control de versions

Entorns de desenvolupament



## Continguts

<b>1. Què és docker?</b>	<b>3</b>
Contenedor i imatges . . . . .	4
1.1. Preparació de l'entorn . . . . .	4
1.2. Instal·lació de Docker-CE . . . . .	6
Docker i serveis . . . . .	7
1.3. Docker i Busybox . . . . .	7
1.4. L'ordre «docker ps» . . . . .	9
1.5. Execució interactiva . . . . .	10
1.6. Netejant espai . . . . .	10
<b>2. Servidor de subversion amb Docker</b>	<b>11</b>
2.1. Descàrrega i execució de la imatge . . . . .	12
2.2. Configurant el servidor Subversion Edge . . . . .	13
2.3. Configuració del servidor . . . . .	17

## 1. Què és docker?



**Figura 1:** Logo de Docker

Segons la Wikipèdia:

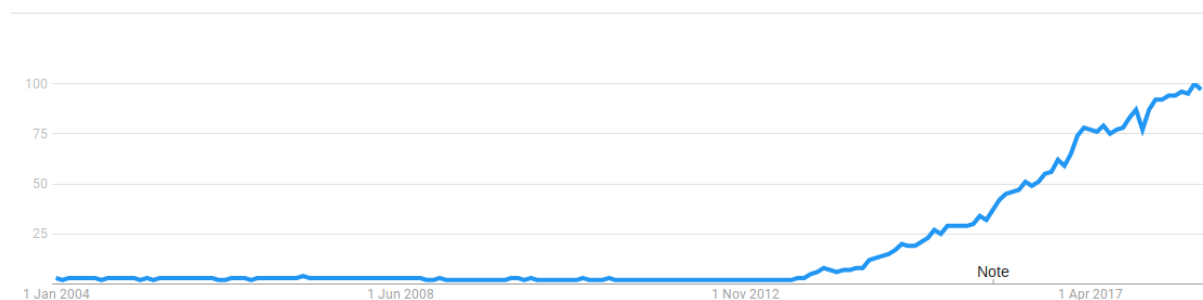
Docker és un projecte de codi obert que automatitza el desplegament d'aplicacions dins de contenidors de programari, i proporciona una capa addicional d'abstracció i automatització de virtualització d'aplicacions en múltiples sistemes operatius. Docker fa ús de les característiques d'aïllament de recursos del nucli de Linux, tals com els cgroups i els namespaces que permeten que contenidors independents s'executen dins una mateixa instància de Linux, evitant la sobrecàrrega d'iniciar i mantenir màquines virtuals.

Una definició més pràctica ens la dona la firma analista *451 Research*:

Docker és una eina que pot empaquetar una aplicació i les seues dependències en un contenidor virtual que es pot executar en qualsevol servidor Linux. Açò aporta flexibilitat i portabilitat a les aplicacions, ja que aquestes es poden executar en instal·lacions físiques, el núvol públic, núvols privats, etc.

Bàsicament, el que aconseguix Docker és oferir contenidors d'aplicacions que aprofiten les capacitats de virtualització del kernel de Linux per poder executar processos i serveis de forma aïllada. Es tracta d'un concepte paregut al de màquina virtual, però que no requereix d'un sistema operatiu, sinò que aprofita el kernel del propi Linux i les capacitats que té d'aïllar recursos, tals com la CPU, la memòria, la xarxa o l'entrada/eixida. Així doncs, podem tindre diversos contenidors compartint el mateix kernel de Linux, però cadascun amb restriccions d'accés a determinats recursos.

En els últims anys, ha estat una tecnologia cada vegada més utilitzada per la versatilitat que dona. Vegem un gràfic de les cerques de «Docker» des del 2004 (Google Trends).



**Figura 2:** Logo de Docker

Els principals avantatges de l'ús de contenidors són:

- La flexibilitat, ja que fins i tot les aplicacions més complexes poden incloure's als contenidors,
- La poca càrrega que suposen per al sistema, al compartir el mateix kernel que l'amfitrió,
- La possibilitat de desplegar actualitzacions *en calent*,
- La portabilitat, ja que es poden desenvolupar localment, desplegar al núvol i llançar-los en qualsevol lloc,
- L'escalabilitat, ja que permet incrementar automàticament rèpliques dels contenidors,
- Els serveis en contenidors poden apilar-se *on the fly*

## Contenedor i imatges

Al llarg del document parlarem sobre imatges i contenidors, pel que convé aclarir aquests conceptes.

Una **imatge** és un paquet executable que inclou tot allò necessari per executar una aplicació: el codi, l'entorn d'execució, llibreríes, variables d'entorn i fitxers de configuració.

Un **contenedor**, per la seua banda és una instància d'una imatge en execució: allò que es crea quan posem en marxa una imatge. Podríem dir que un contenedor és a una imatge el que un procés a un programa (procés=programa en execució -> contenedor=imatge en execució).

### 1.1. Preparació de l'entorn

**Font:** <https://docs.docker.com/install/linux/docker-ce/ubuntu/>

El projecte Docker disposa dels seus propis repositoris de programari. En aquest apartat anem a veure com descarregar-nos la versió Community de Docker, orientada a desenvolupadors i equips menuts que comencen amb Docker. L'alternativa empresarial seria Docker Enterprise Edition (EE).

En aquest apartat anem a instal·lar les eines necessàries per poder descarregar i instal·lar la Docker CE al nostre equip (o màquina virtual).

En primer lloc, actualitzem la caché de paquets de l'ordinador:

```
1 $ sudo apt-get update
```

I instal·lem els paquets següents:

```
1 $ sudo apt-get install \
2   apt-transport-https \
3   ca-certificates \
4   curl \
5   software-properties-common
```

Amb açò, descarreguem la clau GPG del lloc de Docker (amb l'ordre `curl`) i la incorporem al sistema (amb `apt-key add -`), per tal que el nostre sistema confien en el lloc per a la descàrrega de programari (fixeu-se que hi ha una canonada `|`):

```
1 $ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-
   key add -
2 OK
```

Per tal de comprovar que la clau s'ha instal·lat correctament:

```
1 $ sudo apt-key fingerprint 0EBFCD88
2 pub  rsa4096 2017-02-22 [SCEA]
3      9DC8 5822 9FC7 DD38 854A E2D8 8D81 803C 0EBF CD88
4 uid          [ unknown] Docker Release (CE deb) <docker@docker.com>
5 sub  rsa4096 2017-02-22 [S]
```

Si tot és correcte i han aparegut bé les claus de dalt, podem continuar afegint el repositori als orígens de programari del nostre sistema (el que tenim en `/etc/apt/sources.list*`):

```
1 $ sudo add-apt-repository \
2   "deb [arch=amd64] https://download.docker.com/linux/ubuntu \
3   $(lsb_release -cs) \
4   stable"
```

Fet açò, en versions anteriors a Bionic (Ubuntu 18.04) caldria fer un `apt-get update`, per refrescar la llista de paquets de programari disponibles al nostre sistema. Com que Ubuntu 18.04 ja refresca la llista de paquets cada vegada que afegim un dipòsit nou, no és necessari fer-ho en aquesta distribució.

I ara, ja instal·lem el paquet `docker-ce`.

## 1.2. Instal·lació de Docker-CE

Amb els repositoris ja configurats, només hem de realitzar la instal·lació amb `apt-get`:

```
1 $ sudo apt-get install docker-ce
```

Per tal de comprovar que tot ha funcionat bé, anem a llançar un contenidor amb la imatge «*hello-world*»:

```
1 $ sudo docker run hello-world
2 Unable to find image 'hello-world:latest' locally
3 latest: Pulling from library/hello-world
4 d1725b59e92d: Pull complete
5 Digest: sha256:0
   add3ace90ecb4adbf7777e9aacf18357296e799f81cab9fde470971e499788
6 Status: Downloaded newer image for hello-world:latest
7
8 Hello from Docker!
9 This message shows that your installation appears to be working
   correctly.
10
11 To generate this message, Docker took the following steps:
12  1. The Docker client contacted the Docker daemon.
13  2. The Docker daemon pulled the "hello-world" image from the Docker
   Hub.
14     (amd64)
15  3. The Docker daemon created a new container from that image which
   runs the
16     executable that produces the output you are currently reading.
17  4. The Docker daemon streamed that output to the Docker client, which
   sent it
18     to your terminal.
19
20 To try something more ambitious, you can run an Ubuntu container with:
21 $ docker run -it ubuntu bash
22
23 Share images, automate workflows, and more with a free Docker ID:
24 https://hub.docker.com/
25
26 For more examples and ideas, visit:
27 https://docs.docker.com/get-started/
```

Analitzem alguns detalls:

```
1 $ sudo docker run hello-world
```

Com veiem, l'execució de docker s'ha de realitzar com a sudo. Si volem utilitzar docker per a altres usuaris, només haurem de crear un grup docker i afegir ahi els usuaris que el puguin utilitzar.

Per altra banda, veiem com utilitzar docker per tal de llançar una imatge: `docker run`. En aquest cas,

una imatge de prova anomenat *hello-world*.

```
1 Unable to find image 'hello-world:latest' locally
2 latest: Pulling from library/hello-world
3 d1725b59e92d: Pull complete
4 Digest: sha256:0
   add3ace90ecb4adbf7777e9aacf18357296e799f81cab9fde470971e499788
5 Status: Downloaded newer image for hello-world:latest
6
7 Hello from Docker!
8 This message shows that your installation appears to be working
   correctly.
```

Com veiem, ens indica que no troba la imatge «hello-world:latest», pel que la descarrega de la llibreria (pull). Una vegada descarregada ja ens mostra el missatge que comença amb «*Hello from Docker*», corresponent a aquesta imatge.

Si tornem a llançar l'ordre, comprovarem com ja no realitza la descàrrega, sinò que executa directament la imatge Hello World.

## Docker i serveis

El dimoni (daemon) de Docker és el servei que gestiona la creació, execució i distribució de contenidors. L'ordre `docker` que hem vist anteriorment, és el client de docker, que permet a l'usuari interactuar amb el sistema, encara que també hi ha altres aplicacions clints.

Si volem que el servei de Docker s'iniciï en arrancar el sistema (systemd), farem:

```
1 sudo systemctl enable docker
```

I si volem llevar-lo de l'inici:

```
1 sudo systemctl disable docker
```

Podem trobar més informació sobre tot el que podem fer després d'haver instal·lat Docker en: <https://docs.docker.com/install/linux/linux-postinstall/#configure-docker-to-start-on-boot>, així com documentació per a la configuració del servei de Docker per a que escolte determinats ports mitjançant el fitxer `/etc/docker/daemon.json`: <https://docs.docker.com/install/linux/linux-postinstall/#configure-where-the-docker-daemon-listens-for-connections>

### 1.3. Docker i Busybox

A mode d'exemple, anem a veure com descarregar i llançar una imatge per a Docker de Busybox. Busybox no és més que una utilitat que combina moltes eines estàndards d'Unix i ordres de Linux en

un sol fitxer.

Per descarregar el busybox fem ús de l'ordre de Docker `docker pull` (fixeu-se que som l'usuari administrador):

```
1 # docker pull busybox
2 Using default tag: latest
3 latest: Pulling from library/busybox
4 8c5a7da1afbc: Pull complete
5 Digest: sha256:
   cb63aa0641a885f54de20f61d152187419e8f6b159ed11a251a09d115fdff9bd
6 Status: Downloaded newer image for busybox:latest
```

Amb açò hem descarregar la imatge de busybox pe a Docker des del Docker Hub (<https://hub.docker.com/explore/>).

Per tal de vore les imatges de Docker que tenim instal·lades, podem fer:

1	# docker images				
2	REPOSITORY	TAG	IMAGE ID	CREATED	
3	hello-world	latest	4ab4c602aa5e	3 weeks ago	
		1.84kB			
4	busybox	latest	e1ddd7948a1c	2 months ago	
		1.16MB			

Com podem comprovar, tenim la imatge de busybox i hello-world.

Anem ara a llançar el Busybox. Per a això farem:

```
1 # docker run busybox
```

Amb açò, si no tenim descarregada la imatge, el primer que farà és descarregar-la. Si ja la tenim descarregada, ometrà aquest pas. Amb la imatge de busybox al sistema, Docker la busca, la carrega en un contenidor i executa al busybox les ordres que li passem. Com que en aquest cas no li hem passat cap ordre, aparentment, no farà res.

Així doncs, per executar alguna cosa dins el docker li haurem de passar com a paràmetre:

```
1 # docker run busybox echo "hola"
2 hola
3
4 # docker run busybox ls
5 bin
6 dev
7 etc
8 home
9 proc
10 root
11 sys
12 tmp
```



```

13  usr
14  var
15
16  # docker run busybox cat /etc/passwd
17  root:x:0:0:root:/root:/bin/sh
18  daemon:x:1:1:daemon:/usr/sbin:/bin/false
19  bin:x:2:2:bin:/bin:/bin/false
20  sys:x:3:3:sys:/dev:/bin/false
21  sync:x:4:100:sync:/bin:/bin/sync
22  mail:x:8:8:mail:/var/spool/mail:/bin/false
23  www-data:x:33:33:www-data:/var/www:/bin/false
24  operator:x:37:37:Operator:/var:/bin/false
25  nobody:x:65534:65534:nobody:/home:/bin/false

```

Com podem veure, hem llançat tres ordres diferents sobre el Busybox: hem escrit hola, hem llistat el sistema de fitxers, i hem consultat el fitxer `/etc/passwd`. Fixeu-se que ni el sistema de fitxers que hem mostrat ni el fitxer `passwd` es corresponen amb l'estructura de fitxers del sistema o el fitxer `passwd` del nostre sistema. Estem accedint al sistema de fitxers i el fitxer `passwd` del propi Busybox. De fet, si fem un `ps aux`, veurem que no hi ha cap altre procés en el sistema:

```

1  # docker run busybox ps aux
2  PID    USER      TIME  COMMAND
3       1   root        0:00  ps aux

```

#### 1.4. L'ordre «docker ps»

L'ordre `docker ps` serveix per veure els contenidors que s'estan executant en un moment donat. Si fem en un terminal:

```
1 $ docker run busybox sleep 10
```

I des d'altre:

```

1 $ sudo docker ps
2  CONTAINER ID        IMAGE               COMMAND             CREATED
3  d4018e11cc64        busybox            "sleep 10"         5 seconds
4  ago                 Up 4 seconds       elegant_keldysh
4  joamuran@toki:~$

```

Veiem que tenim l'ordre `sleep 10` funcionant sobre la imatge de `busybox` en el contenidor `d4018e11cc64`.

## 1.5. Execució interactiva

Si volem llançar més d'una ordre per contenidor, podem fer ús del paràmetre -it (flag interactive):

```

1 root@toki:/home/joamuran# docker run -it busybox sh
2 / # ls
3 bin    dev    etc    home  proc  root  sys    tmp    usr    var
4 / # users
5
6 / # cat /etc/passwd
7 root:x:0:0:root:/root:/bin/sh
8 daemon:x:1:1:daemon:/usr/sbin:/bin/false
9 bin:x:2:2:bin:/bin:/bin/false
10 sys:x:3:3:sys:/dev:/bin/false
11 sync:x:4:100:sync:/bin:/bin/sync
12 mail:x:8:8:mail:/var/spool/mail:/bin/false
13 www-data:x:33:33:www-data:/var/www:/bin/false
14 operator:x:37:37:Operator:/var:/bin/false
15 nobody:x:65534:65534:nobody:/home:/bin/false
16 / # exit

```

## 1.6. Netejant espai

Amb l'opció -a de docker ps podem obtenir tots els contenidors que s'han creat en la sessió actual:

```

1 # docker ps -a
2 CONTAINER ID        IMAGE               COMMAND             CREATED             NAMES
3 5dde99182cec        busybox            "sh"               2 minutes ago      festive_ride
4 d4018e11cc64        busybox            "sleep 10"         16 minutes ago     elegant_keldysh
5 2a2b08d458c5        busybox            "ps aux"           20 minutes ago     zen_bassi
6 b17136bc9389        busybox            "cat /etc/passwd"  22 minutes ago     gifted_archimedes
7 a475a7587143        busybox            "ls"               23 minutes ago     jovial_leavitt
8 1febe609a9a9        busybox            "echo hola"        23 minutes ago     compassionate_spence
9 dc64b0307514        busybox            "sh"               25 minutes ago     clever_lalande

```

10	2a4808f37627	hello-world	"/hello"	About an hour ago	Exited (0) About an hour ago
11	31545996cff7	hello-world	"/hello"	About an hour ago	Exited (0) About an hour ago

Com veiem, la columna *status* indica que els contenidors han acabat. So volem eliminar-los, podríem fer:

```
1 # docker rm 5dde99182cec d4018e11cc64...
```

Aquest mecanisme és un poc tediós, pel que anem a fer-ho més senzill.

Amb la següent ordre, podem obtenir els ids dels contenidors que ja han acabat:

```
1 toki:/home/joamuran# sudo docker ps -a -q -f status=exited
2 5dde99182cec
3 d4018e11cc64
4 2a2b08d458c5
5 b17136bc9389
6 a475a7587143
7 1febe609a9a9
8 dc64b0307514
9 2a4808f37627
10 31545996cff7
```

Pel que si combinem aquesta ordre amb docker rm:

```
1 sudo docker rm $(docker ps -a -q -f status=exited)
```

## 2. Servidor de subversion amb Docker

Com hem comentat anteriorment, el Docker Hub (<https://hub.docker.com/explore/>) oferix una gran quantitat d'imatges. Aquest lloc pot entendre's com una espècie de Github per a imatges de Docker.

Les imatges es poden classificar de diverses formes atenent a diferents criteris.

Atenent al procés de creació de la imatge, podem distingir:

- **Imatges de base:** Aquelles que han estat creades de zero, generalment a partir de sistemes operatius com Ubuntu, Busybox o Debian, o bé
- **Imatges filles:** Aquelles que estan construïdes sobre una imatge base, amb funcionalitat addicional.

Per altra banda, atenent a qui ha creat les imatges, distingim:

- **Imatges oficials:** Són aquelles mantinguts per Docker, i que generalment són imatges de base. Generalment el nom és una cadena, com `busybox` i `hello-world`.
- **Imatges d'usuari:** Són aquelles creades i compartides pels usuaris. Generalment, són imatges basades en les imatges de base amb funcionalitat addicional. Normalment, s'anomenen amb `usuari/imatge`.

Per tal de disposar d'un servidor de Subversion, farem ús de la imatge `mamohr/subversion-edge` (<https://hub.docker.com/r/mamohr/subversion-edge>), basat en el servidor de subversion Subversion Edge de CollabNet, que disposa d'interfície web per a la seua gestió.

## 2.1. Descàrrega i execució de la imatge

Per tal de descarregar la imatge del Subversion Edge des de Docker Hub, farem:

```
1 $ docker run -d mamohr/subversion-edge
2 Unable to find image 'mamohr/subversion-edge:latest' locally
3 latest: Pulling from mamohr/subversion-edge
4 469cfcc7a4b3: Pull complete
5 7e814dc876c6: Pull complete
6 144c1c97a750: Pull complete
7 17eb6ecf9d0a: Pull complete
8 12492ca50197: Pull complete
9 32dc87336990: Pull complete
10 Digest: sha256:547356
    c2d4c2b2a7eb6e8f7b84de52118ca7089b98d2400f7c723e2280022ab6
11 Status: Downloaded newer image for mamohr/subversion-edge:latest
```

Com veiem, ens indica que no troba la imatge localment, pel que la descarregarà d'Internet. Si ens fixem, hem utilitzat l'opció `-d` (*detach*) de `docker run`, per llançar l'execució de forma independent, i no veure tots els missatges de log que puga generar l'execució per la terminal.

Amb l'ordre anterior, tindrem el servidor executant-se en un contenidor. De moment, no tindrem encara aquest servidor accessible, ja que per a això, cal fer que el contenidor siga *visible* des de la nostra màquina.

De moment, veiem els contenidors que tenim en execució:

```
1 :~$ docker ps
2 CONTAINER ID      IMAGE               COMMAND
3 3a44963e4e29      mamohr/subversion-edge  "/config/bootstrap.sh"
   26 seconds ago   Up 25 seconds      3343/tcp, 4434/tcp, 18080/
   tcp            nifty_ramanujan
```

I anem a aturar-lo i eliminar-lo:

```
1 $ docker stop 3a44963e4e29
2 3a44963e4e29
3 $ docker rm 3a44963e4e29
4 3a44963e4e29
```

Ara tornarem a llençar el contenidor de la següent forma:

```
1 $ docker run -d -p 3343:3343 -p 4434:4434 -p 18080:18080 --name svn-
  server mamohr/subversion-edge
2 bcebfd61510e4fb5c456503a81c528f535d2407a90df1a8a031b1300edc8335a
```

L'opció `-p` ens serveix per *exposar els ports* del contenidor al host local. Amb `-p PortLocal:PortContenidor` redireccionem el port pel que treballa el contenidor a un port en la màquina local, de manera que ja tenim accessibles estes adreces.

Per altra banda, també hem afegit l'opció `--name svn-server`, amb què li donem un nom al contenidor. D'aquesta manera ens serà més fàcil referir-nos a ell per aturar-lo o eliminar-lo.

## 2.2. Configurant el servidor Subversion Edge

Si ara accedim a través del navegador a l'adreça <http://127.0.0.1:3343/>, accedirem a la interfície web de gestió del servidor. Per accedir-hi, podem fer-ho amb l'usuari `admin` i contrassenya `admin`, per veure'n l'aspecte, però de moment, els canvis que fem al contenidor no tindran efecte sobre la màquina, ja que en aturar el contenidor, aquests es perdran. Per a això hem d'utilitzar el que es coneix com a *volums*, que no són més que carpetes del nostre equip que es *sincronitzaran* amb certes carpetes del contenidor.

Per fer tot açò, haurem de llançar l'ordre:

```
1 $ docker run -d -p 3343:3343 -p 4434:4434 -p 18080:18080 -v /srv/svn-
  data:/opt/csvn/data --name svn-server mamohr/subversion-edge
2 8a25bec4c67cf8fda2100655a79bb2269aabff7b4944dc8798991523bd1fd2cd
```

Amb el que indiquem que la carpeta `/opt/csvn/data` del contenidor, que és on el CSVN guarda tota la informació que genera, la sincronitzarem al *volum* (directori) `/srv/svn-data`.

Amb açò ja podem accedir al servidor via web, generar usuaris, i generar repositoris guardant els canvis. Veiem-ho pas per pas.

Una vegada accedim al sistema amb l'usuari «admin» i password «admin» (que en entorns de producció caldrà modificar!), veurem la següent finestra:

The screenshot shows the SubversionEdge web interface. The top navigation bar includes links for Repositories, Users, Administration, and Extensions. The user is logged in as 'Super Administrator (admin)'. A yellow banner at the top indicates that there are new updates available for download. The left sidebar contains a 'MAINTENANCE' section with links to Status, Server Logs, Software Updates, Statistics, and Jobs, as well as a 'CONFIGURATION' section with links to Server Settings, Authentication, Proxy Server, Mail Server, Logging, and Server Monitoring. The main content area is titled 'Status' and displays the following information:

- Subversion status:** Up (indicated by a green circle)
- Hostname:** fb8835815f46
- Repository parent:** <http://fb8835815f46:18080/svn/>
- Browse repositories:** <http://fb8835815f46:18080/viewvc/>

Below the status information is a table titled 'Information' with the following data:

Software version	5.2.0-4404.48
Subversion version	1.8.17-4404.48
Running since	10/23/2019 11:41:54 UTC
Repo health	Total repositories: 3 - Go to the Repositories tab to create or discover repositories.
Throughput on primary interface	78.25 B/s IN (over about 5 minutes); 797.07 B/s OUT (over about 5 minutes)
Disk Usage	as of 10/23/2019 11:43:55 UTC
Used space on root volume	22.7 GB
Used space by repositories	87.26 KB
Free space on repository volume	37.71 GB

A tip box on the left sidebar states: 'Tip: Are you supporting a large geographically distributed team across slow WAN links? Connect Subversion Edge to TeamForge and you can use their combined support for Subversion replicas and proxies. Drop a Subversion Edge proxy in the LAN of your remote team and they will get fast read access to their'.

**Figura 3:** Finestra principal de Subversion Edge

Fem clic en **Usuaris**, per gestionar els usuaris del sistema de control de versions:

The screenshot shows the 'Users' management page in the SubversionEdge interface. The top navigation bar includes links for es, Users, Administration, and Extensions. The user is logged in as 'Super Administrator (admin)'. A 'Click here' button is visible. The main content area is titled 'Users' and displays a table of users. The table has columns for Username, Full Name, and Description. The following table represents the data shown:

Username	Full Name	Description
admin	Super Administrator	admin user

Below the table, it says 'Showing 1 to 1 of 1 entries'. There are navigation buttons for 'Previous', '1', and 'Next'. A 'Create' button is also visible at the bottom right.

**Figura 4:** Gestió d'usuaris

I afegim un nou usuari:

es Users Administration Extensions Super Administrator (admin) Logout Help

Click here

### Users

10 records per page Filter:

Username	Full Name	Description
admin	Super Administrator	admin user

Showing 1 to 1 of 1 entries

← Previous 1 Next →

Create

**Figura 5:** Creant un nou usuari

Ara eixim del sistema, i entrem amb l'usuari que hem creat, per veure que podem accedir, i anem a crear un repositori des de la web. Per a això fem clic a la pestanya de repositoris:

SubversionEdge by CollabNet Repositories Extensions Jose A. (joamuran) Logout Help

REPOSITORIES

- Repository List
- Access Rules
- Backup Schedule
- Manage Templates

### Repositories

10 records per page Filter:

Name	Checkout command	Status
There are no repositories yet. You may create a new repository using the 'Create' button.		

Showing 0 entries

← Previous Next →

Create Discover Info Dump Verify Load Delete

**Tip:** Consider connecting your Subversion Edge server to TeamForge for enhanced permissions management functionality. TeamForge provides a Role-Based Access Control model that allows you to delegate permissions management to project teams.

**Figura 6:** Repositoris

I li donem a **Create**:

Repositories Extensions jose A. (joamuran) Logout Help ▾

## Create Repository

Name:

Initialize:

- ☒ Template
- ☐ Empty repository
- ☒ Create standard trunk/branches/tags structure
- ☐ Backup

[Create](#)

**Figura 7:** Creació d'un repositori

Repository: Projecte1

Repository: Projecte1

Backup Files [Scheduled Jobs](#) [Size Report](#)

10 records per page Filter:

File Name	Date	Size
The repository has no dump files.		

Showing 0 entries

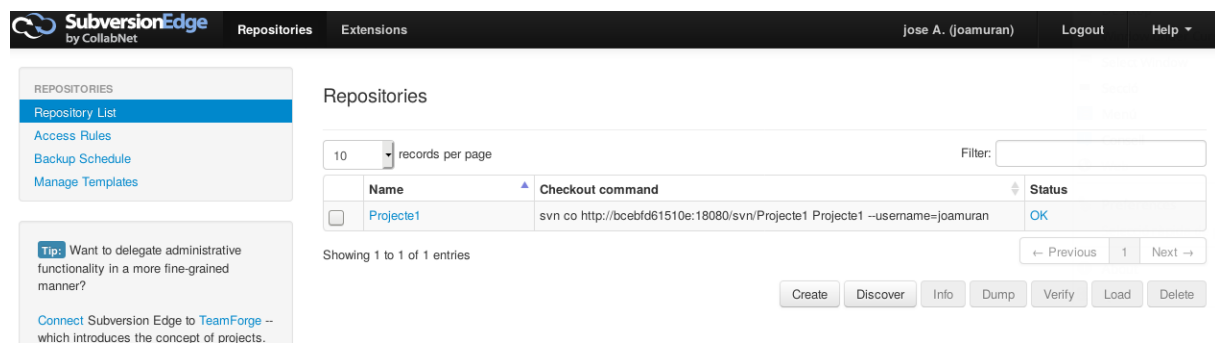
← Previous Next →

[Download](#) [Delete](#)

**Figura 8:** Creació d'un repositori

Una vegada creat, des de la finestra dels repositoris, podrem veure la URL en principi per accedir a ell:





**Figura 9:** Repositoris

Com veiem, apareix el nom del contenidor com al host al que ens hem de connectar. **Com que no tenim els DNS configurats per accedir a aquesta adreça, ho farem a través de l'adreça del host local.**

Així doncs, per accedir al repositori a través de la línia d'ordres, obrirem un terminal, i ens ubicarem en la carpeta del nostre equip des de la que volgam treballar amb Subversion, i farem ús de l'ordre `svn co` per fer un *checkout* del repositori a aquesta carpeta:

```
1 $ svn co http://127.0.0.1:18080/svn/test test
2 Authentication realm: <http://127.0.0.1:18080> CollabNet Subversion
  Repository
3 Password for 'joamuran': *****
```

Si desitgem accedir al repositori d'un altre usuari, deixarem en blanc la contrassenya, i ens demanarà l'usuari amb el que desitgem identificar-nos i la contrassenya corresponent.

L'eixida de l'ordre anterior serà semblant a la següent:

```
1 A    test/branches
2 A    test/tags
3 A    test/trunk
4 Checked out revision 1.
```

## 2.3. Configuració del servidor

Si accedim al servidor com a administradors (admin/admin), des de la pestanya *Administration*, podrem consultar i modificar la configuració del nostre sistema:

MAINTENANCE

Status

Server Logs

Software Updates

Statistics

Jobs

CONFIGURATION

Server Settings

Authentication

Proxy Server

Mail Server

Logging

Server Monitoring

EXTENSIONS

TeamForge

**Tip:** One of the best features Subversion Edge provides is automatic log rotation. Be sure to [configure the number of log files to keep](#) so that your server disk does not fill up with log files.

## Status

Subversion status:

Up

Hostname:

fb8835815f46

Repository parent:

<http://fb8835815f46:18080/svn/>

Browse repositories:

<http://fb8835815f46:18080/viewvc/>

Stop

## Information

Software version	5.2.0-4404.48
Subversion version	1.8.17-4404.48
Running since	10/23/2019 14:47:43 UTC
Repo health	Total repositories: 3 - Go to the Repositories tab to create or discover repositories.
Throughput on primary interface	96.06 B/s IN (over about 5 minutes); 1.72 KB/s OUT (over about 5 minutes)
Disk Usage	as of 10/24/2019 14:19:45 UTC
Used space on root volume	22.53 GB
Used space by repositories	261.79 KB
Free space on repository volume	37.71 GB

**Figura 10:** Administració del servidor

Com veiem, al menú de l'esquerra tenim opcions per configurar des dels paràmetres del servidor, com l'autenticació, el pas per un proxy, servidor de correu, gestió de logs o monitorització del sistema.

En primer lloc, anem a modificar el *hostname* del servidor, per a que en lloc d'aparèixer l'*id* del contenidor, ens aparega simplement *localhost*. Per a això anem a l'opció *Server Settings* del menú, i modifiquem el *hostname* a *\*localhost*. Fixeu-se que des d'ací mateix, podeu configurar el port pel que ens connectem al servidor (18080 per defecte), així com els directoris on guardem els repositoris:

## Server Settings

[Advanced Settings](#)

---

Hostname:	<input type="text" value="localhost"/>
The fully qualified hostname.	
Apache Encryption:	<input type="checkbox"/> Subversion Server should serve via https.
Port:	<input type="text" value="18080"/>
Standard ports may require additional setup.	
Repository Directory:	<input type="text" value="/opt/csvn/data/repositories"/>
Parent directory that includes all repositories.	
Backup Directory:	<input type="text" value="/opt/csvn/data/dumps"/>
Repository dump files will be stored here, under their respective repository name.	
Administrator:	<input type="text" value="Nobody"/>
Administrator Email:	<input type="text" value="devnull@collab.net"/>
Administrator Alternative Contact:	<input type="text"/>
Console Encryption:	<input type="checkbox"/> Subversion Edge Management Console should require https.

---

**Figura 11:** Modificació del hostname

**IMPORTANT! Tingueu en compte que al estar exposant els ports, és possible que en aplicar els canvis ens indique que el port està ocupat, i no es pot iniciar el servei. En aquest cas, caldrà aturar i eliminar el contenidor i llençar-lo de nou**