



La Paz, 20 de agosto de 2018

Nicolas Laguna Quiroga

Director General Ejecutivo

AGENCIA DE GOBIERNO ELECTRONICO Y TECNOLOGIAS DE INFORMACION Y COMUNICACION

Ref. Remisión manual de instalación del Sistema de Almacenes AGETIC
y cambio de responsable de Institución Publica

Presente:

Con anterioridad se me capacitó para poder instalar el sistema de Almacenes y Activos con el que cuenta su institución para ser aplicado a la Autoridad Plurinacional de la Madre Tierra APMT. Mi persona realizó la documentación de instalación mediante un manual detallado. Agradeciendo por la capacitación realizada y como aporte al sistema, me permito remitir adjunto a la presente nota en formato digital la documentación elaborada para poder ser mejorada y utilizada por su institución.

Del mismo modo me permito informar que mi persona dejó de prestar servicios desde el 31 de julio de 2018 en la APMT y por tanto ya no cumplo las funciones de coordinador para ninguno de los planes que debe conllevar la APMT con la AGETIC (Plan de Gobierno Electrónico, Plan Institucional de la Seguridad de La Información, Plan de migración a Software Libre, y otros asignados por la institución en su momento).

Agradeciendo por su tiempo y sin otro particular me despido atentamente,


Jorge Antonio Maydana Maydana
INGENIERO DE SISTEMAS
R.N.I. 16.479



Estado Plurinacional de Bolivia

MANUAL DE INSTALACION DE SISTEMAS DE ALMACENES Y ACTIVOS

ING JORGE ANTONIO MAYDANA MAYDANA

MANUAL DE INSTALACION DE SISTEMA DE ALMACENES Y ACTIVOS APMT

1. Introducción

La Autoridad Plurinacional de la Madre Tierra APMT, es una entidad autárquica, que se encarga de promover la gestión territorial, integral y sustentable de la mitigación y/o adaptación y de manera conjunta a los efectos del cambio climático para incrementar la resiliencia de los sistemas de vida y contribuir al desarrollo integral en armonía con la Madre Tierra a través de programas y/o proyectos y acciones en el marco de las metas y resultados definidos en el PDES y la Articulación al Plan Sectorial. Como toda institución publica cuenta con el área de activos y almacenes para el uso de los bienes del estado por parte de los funcionarios.

El presente Manual, tiene la finalidad de proporcionar a la Autoridad Plurinacional de la Madre Tierra - APMT, los pasos necesarios para realizar la instalación del Sistema de Almacenes y Activos proporcionado por la Agencia de Gobierno Electrónico AGETIC para la Autoridad Plurinacional de la Madre Tierra APMT.

El sistema mencionado, permite la administración de los suministros de almacenes y activos y en tal sentido, integra los lineamientos que enmarcan la actuación de las diferentes Unidades, en relación al uso y control de los materiales brindados y activos asignados, así como define los objetivos y funciones del almacenamiento y designación.

El Manual contiene conceptos e instrucciones para la instalación del sistema mencionado en tal sentido que un profesional en el área de sistemas pueda replicar la reinstalación o actualización del mismo..

2. Conceptos Básicos

2.1 Contenedores de Software

Los contenedores de software son paquetes de elementos que permiten ejecutar una aplicación determinada en cualquier sistema operativo.

Uno de los términos que vienen sonando dentro del mundo de la tecnología y el desarrollo de aplicaciones en los últimos años es el de los contenedores de software. Se utilizan para garantizar que una determinada aplicación se ejecute correctamente cuando cambie su entorno, sin dar fallos de ningún tipo. En cierto modo se asemeja a la tecnología de virtualización, aunque se puede decir que funcionan en un plano menor.

El uso de contenedores de software ha proliferado en los últimos años porque resultan útiles y ágiles para migrar cualquier desarrollo de una plataforma a otra. Si se desarrolla software y se desea pasarlo de un servidor instalado en un centro de datos a una máquina virtual que funciona en una nube pública, tal vez el código no termine de funcionar del todo bien en su nuevo entorno. Lo mismo ocurre al migrar una aplicación del sistema operativo Debian a producción, en el sistema Red Hat.

Si ese software desarrollado se coloca en un contenedor, es posible llevarlo al sistema que más convenga. Las diferencias que tienen los sistemas operativos y que hacen que la aplicación no funcione del todo bien al cambiar de entorno desaparecen. El contenedor actúa como una funda para el software que lo habilita para funcionar dentro del nuevo entorno. Solo hay que empaquetar el código y las herramientas que lo acompañen dentro de un contenedor.

2.2 Contenedor Docker

Docker es un proyecto de código abierto que automatiza el despliegue de aplicaciones dentro de contenedores de software, proporcionando una capa adicional de abstracción y automatización de

Virtualización a nivel de sistema operativo en Linux. Docker utiliza características de aislamiento de recursos del kernel Linux, tales como cgroups y espacios de nombres (namespaces) para permitir que "contenedores" independientes se ejecuten dentro de una sola instancia de Linux, evitando la sobrecarga de iniciar y mantener máquinas virtuales.

El soporte del kernel Linux para los espacios de nombres, aísla la vista que tiene una aplicación de su entorno operativo, incluyendo árboles de proceso, red, ID de usuario y sistemas de archivos montados, mientras que los cgroups del kernel proporcionan aislamiento de recursos, incluyendo la CPU, la memoria, el bloque de E / S y de la red. Desde la versión 0.9, Docker incluye la biblioteca libcontainer como su propia manera de utilizar directamente las facilidades de virtualización que ofrece el kernel Linux, además de utilizar las interfaces abstraídas de virtualización mediante libvirt, LXC (Linux Containers) y systemd-nspawn.

De acuerdo con la firma analista de la industria 451 Research, "Docker es una herramienta que puede empaquetar una aplicación y sus dependencias en un contenedor virtual que se puede ejecutar en cualquier servidor Linux. Esto ayuda a permitir la flexibilidad y portabilidad en donde la aplicación se puede ejecutar, ya sea en las instalaciones físicas, la nube pública, nube privada, etc."

3. Procedimiento de Instalación

Una vez terminada la instalación del Sistemas de Activos y Almacenes, debe ingresar a la misma mediante la clave de usuario administrador y posteriormente cambiar la misma con el fin de resguardar la seguridad del sistema:



Usuario : admin
Pass : demo123

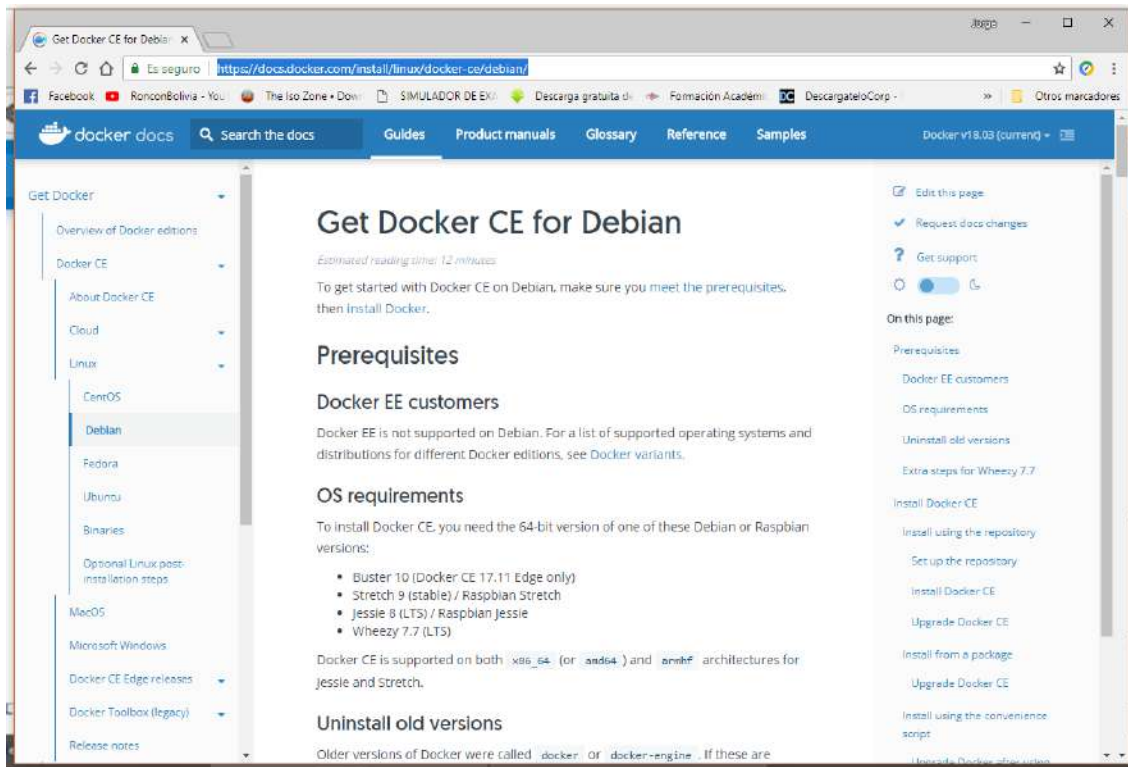
El sistema trabaja con contenedores y a continuación se explica paso a paso la instalando todas las herramientas requeridas.

3.1. Instalación de DOCKER CE

Para empezar instalaremos el docker como corresponde por tanto debemos entrar a:

<https://docs.docker.com/install/linux/docker-ce/debian/>

Al ingresar nos dará los pasos para instalar Docker CE para DEBIAN. El autor le coloca el término "CE" al docker puesto que en debian ya existe un programa llamado Docker que cumple funciones de docs de estado de las aplicaciones, es un paquete propio de debian. Por lo tanto, para estandarizar en todas las versiones de linux se le denomina "Docker CE".



Para la instalación se debe ejecutar los siguientes comandos:

- Actualizamos la lista de repositorios de DEBIAN

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

- Instalamos paquetes que permitan usar repositorios sobre HTTPS (Certificado HTTP Seguro) pues son necesarios para descargar Docker

```
$ sudo apt-get install \  
  apt-transport-https \  
  ca-certificates \  
  curl \  
  gnupg2 \  
  software-properties-common
```

Al escribir un comando y colocar “\” (BackSlash) al final del mismo, indica que existe un salto de línea en el comando por tanto no ejecuta el comando hasta que la última línea no tenga “\”, directamente enter.

- Ahora descargamos la Llave GPG oficial del portal web de Docker y lo adicionamos mediante apt-key (Se quita el “sudo” si se encuentra logueado como root)

```
$ curl -fsSL https://download.docker.com/linux/debian/gpg | sudo apt-key add -
```

- Opcionalmente podemos verificar si la llave GPG de docker fue insertada correctamente mediante el comando fingerprint

```
$ sudo apt-key fingerprint 0EBFCD88
```

The screenshot shows a terminal window titled 'apmt@ALMACEN: ~'. It displays the output of system updates, including the installation of 'unattended-upgrades' and various packages like 'gnupg-agent', 'curl', and 'gnupg'. The terminal then shows the command 'apt-key add -' being executed, followed by 'OK'. Below this, the command 'apt-key fingerprint 0EBFCD88' is run, resulting in the following output:

```
pub   rsa4096 2017-02-22 [SCEA]
      9DC8 5822 9FC7 DD38 854A  E2D8 8D81 803C 0EBF CD88
uid   [ unknown] Docker Release (CE deb) <docker@docker.com>
sub   rsa4096 2017-02-22 [S]
```

The output is highlighted with a red box. The terminal window also shows a taskbar at the bottom with icons for a menu, file manager, terminal, and a Docker download button. The system clock shows 16:36 on 13/6/2018.

- Adicionamos el repositorio de docker a la liste de DEBIAN

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

- En vista que ya tenemos el repositorio adherido al sistema, ahora procedemos a actualizar los mismos mediante apt-get update.

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

- Instalamos DOCKER CE

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

- Una vez terminado el proceso de instalacion verificamos que este funcionando

```
$ docker --version
```

Este comando devuelve la versión instalada y por tanto demuestra que esta funcionando el docker



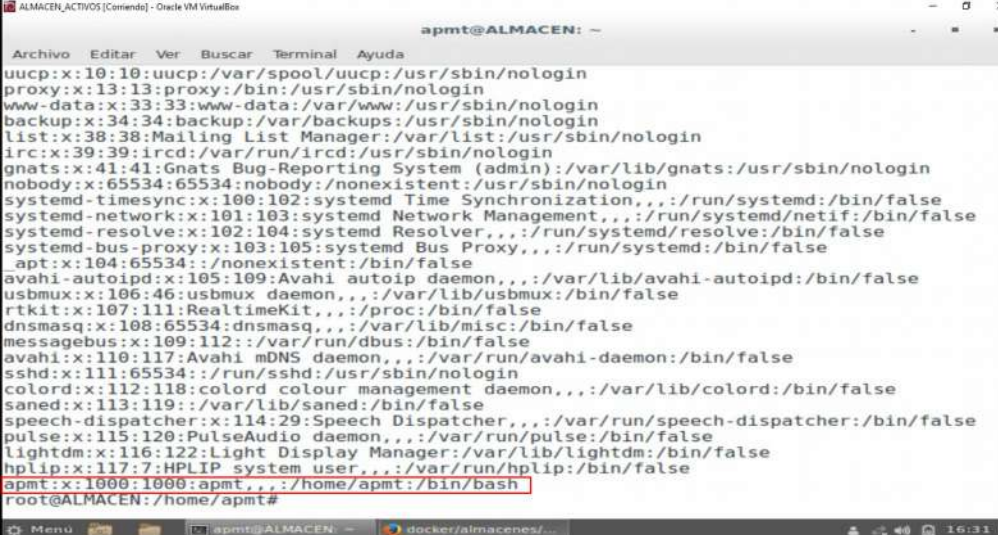
```
apmt@ALMACEN: ~  
Archivo Editar Ver Buscar Terminal Ayuda  
update-alternatives: utilizando /usr/bin/fakeroot-sysv para proveer /usr/bin/fakeroot (fakero  
Configurando gcc-6 (6.3.0-18+deb9u1) ...  
Configurando gcc (4:6.3.0-4) ...  
Configurando linux-headers-4.9.0-6-amd64 (4.9.88-1+deb9u1) ...  
Configurando dkms (2.3-2) ...  
Configurando linux-headers-4.9.0-6-amd64 (4.9.88-1+deb9u1) ...  
Configurando aufs-dkms (4.9+20161219-1) ...  
Loading new aufs-4.9+20161219 DKMS files...  
Building for 4.9.0-6-amd64  
Building initial module for 4.9.0-6-amd64  
Done.  
  
aufs:  
Running module version sanity check.  
- Original module  
- No original module exists within this kernel  
- Installation  
- Installing to /lib/modules/4.9.0-6-amd64/updates/dkms/  
  
depmod.....  
  
DKMS: install completed.  
Configurando linux-headers-amd64 (4.9+80+deb9u4) ...  
Procesando disparadores para libc-bin (2.24-11+deb9u3) ...  
root@ALMACEN:/home/apmt# docker --version  
Docker version 18.03.1-ce, build 9ee9f40  
root@ALMACEN:/home/apmt#
```

3.2. Permisos de usuario para docker

Es recomendable utilizar un usuario distinto de root pues dicho usuario cuenta con todos los privilegios del equipo y por tanto constituye un riesgo de seguridad. Para el ejemplo daremos permiso a un usuario “apmt” que se encuentra registrado como usuario normal en el equipo.

- Verificamos que usuarios están en nuestro equipos. Para ver los usuarios que se tiene se usa el comando:

```
$ cat /etc/passwd
```

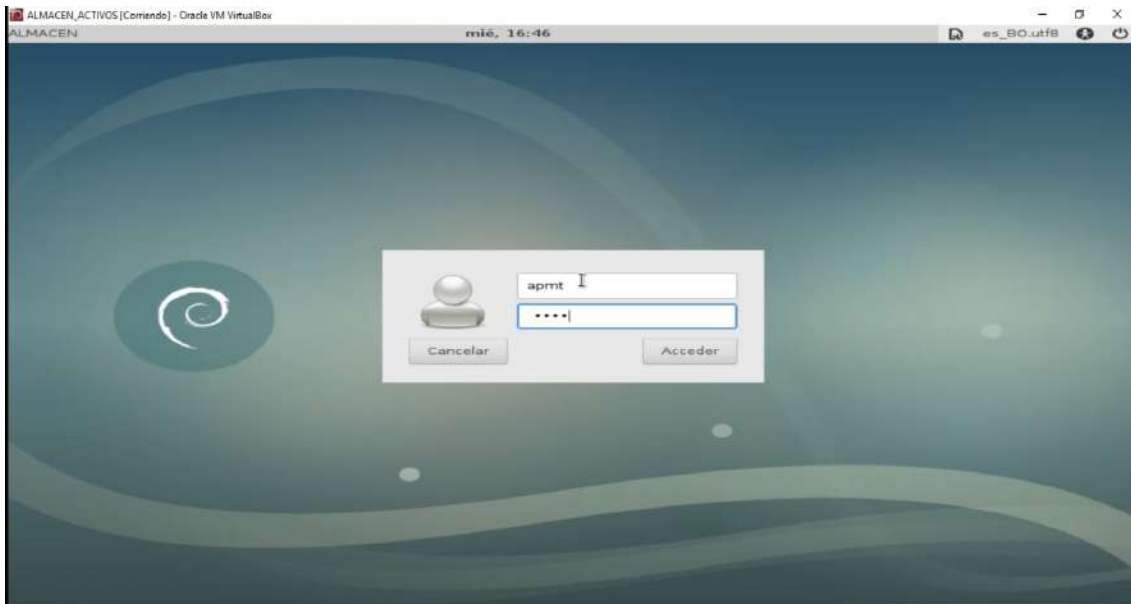


```
ALMACEN_ACTIVOS [Corriendo] - Oracle VM VirtualBox
apmt@ALMACEN: ~
Archivo Editar Ver Buscar Terminal Ayuda
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin)/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-timesync:x:100:102:systemd Time Synchronization,,,:/run/systemd:/bin/false
systemd-network:x:101:103:systemd Network Management,,,:/run/systemd/netif:/bin/false
systemd-resolve:x:102:104:systemd Resolver,,,:/run/systemd/resolve:/bin/false
systemd-bus-proxy:x:103:105:systemd Bus Proxy,,,:/run/systemd:/bin/false
apt:x:104:65534:/:nonexistent:/bin/false
avahi-autoipd:x:105:109:Avahi autoip daemon,,,:/var/lib/avahi-autoipd:/bin/false
usbmux:x:106:46:usbmux daemon,,,:/var/lib/usbmux:/bin/false
rtkit:x:107:111:RealtimeKit,,,:/proc:/bin/false
dnsmasq:x:108:65534:dnsmasq,,,:/var/lib/misc:/bin/false
messagebus:x:109:112:/:/var/run/dbus:/bin/false
avahi:x:110:117:Avahi mDNS daemon,,,:/var/run/avahi-daemon:/bin/false
sshd:x:111:65534:/:run/sshd:/usr/sbin/nologin
colord:x:112:118:colord colour management daemon,,,:/var/lib/colord:/bin/false
saned:x:113:119:/:/var/lib/saned:/bin/false
speech-dispatcher:x:114:29:Speech Dispatcher,,,:/var/run/speech-dispatcher:/bin/false
pulse:x:115:120:PulseAudio daemon,,,:/var/run/pulse:/bin/false
lightdm:x:116:122:Light Display Manager:/var/lib/lightdm:/bin/false
hplip:x:117:7:HPLIP system user,,,:/var/run/hplip:/bin/false
apmt:x:1000:1000:apmt,,,:/home/apmt:/bin/bash
root@ALMACEN:/home/apmt#
```

- Podemos observar en la gráfica que existe el usuario apmt que tiene permisos moderados. Usaremos ese usuario para levantar el docker. Por lo tanto, es necesario darle los permisos para que pueda ejecutar comandos docker.

```
$ usermod -aG docker apmt
```

- Para que se activen los cambios realizados debemos cerrar sesión e ingresar nuevamente. Si estamos con interfaz gráfica, debemos cerrar sesión de linux e ingresar nuevamente por la pantalla de logueo. En caso de modo consola ejecutamos el comando exit hasta que nos pida en la línea de comandos el logueo nuevamente.



- Ingresamos nuevamente a una terminal y escribimos el siguiente comando.

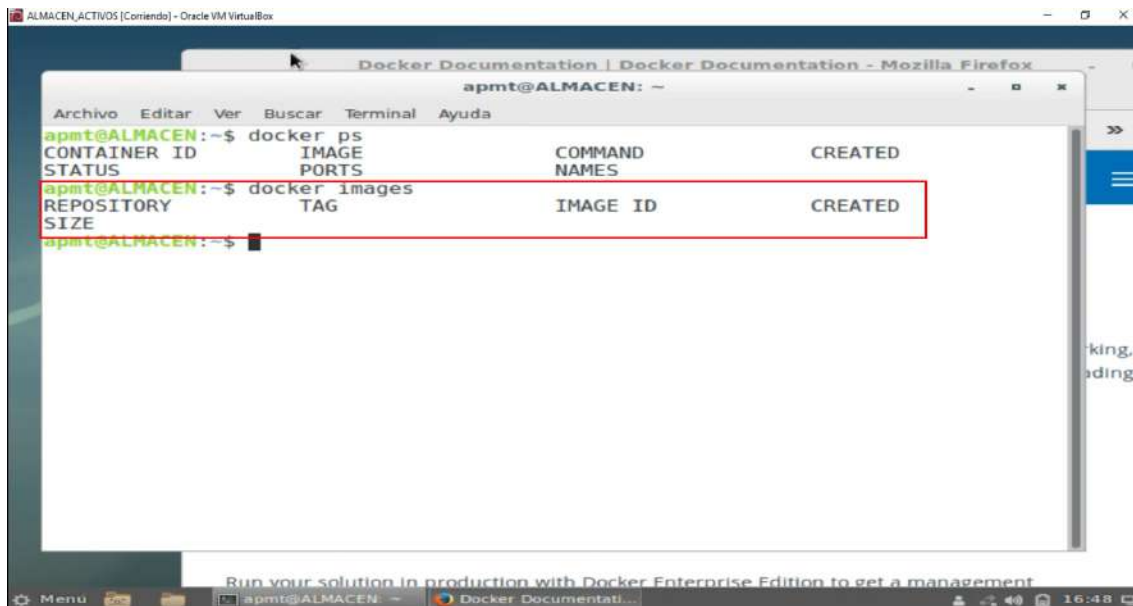
```
$ docker ps
```

El comando permite ver el estado de los contenedores levantados por docker. Como podemos ver en la siguiente grafica no existe aun ningún contendor levantado por que el equipo esta nuevo.



- El comando docker images nos permitirá ver las imágenes cargadas de contenedores en el sistema. Todo sistema se empaqueta en contenedores y estos a la vez cargan imágenes en el sistema.

\$ docker images



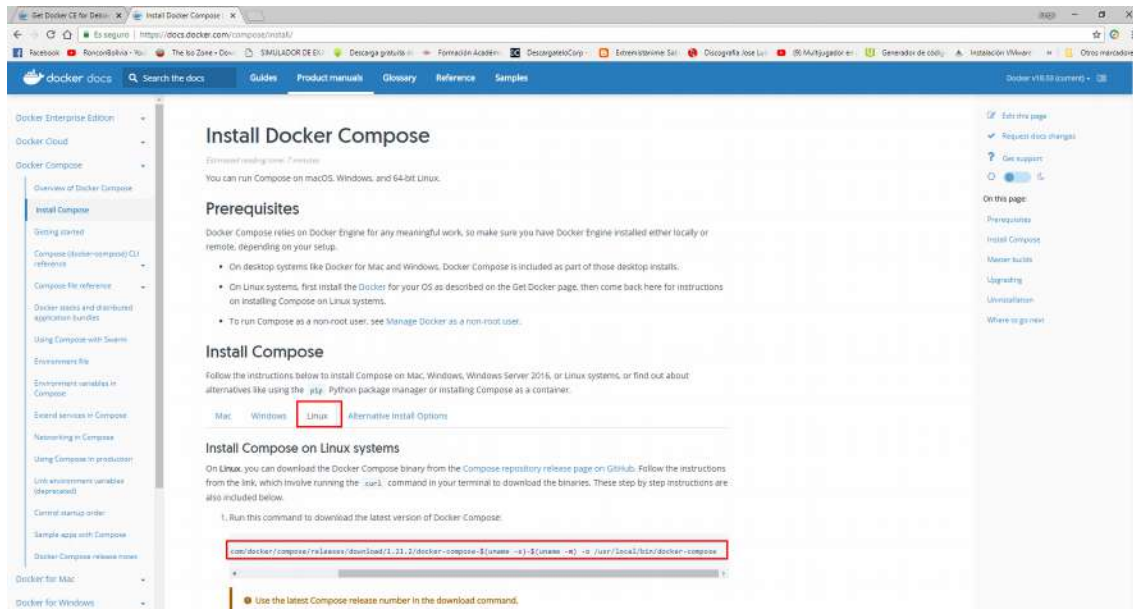
3.3. Instalación de Docker Compose

El Docker Compose, permite levantar contenedores sobre el Docker CE instalado en nuestra maquina. Se podría decir que es un gestor de contenedores.

- Ingresamos al link de instalación de docker compose

<https://docs.docker.com/compose/install/>

Elegimos el tab Linux y copiamos toda la sentencia del recuadro para ejecutarla en el modo consola



- Ingresamos en la consola como usuario root

\$ su

(Colocar la clave de root)

- Luego ejecutamos la sentencia

```
$ sudo curl -L https://github.com/docker/compose/releases/download/1.21.2/docker-compose-$(uname -s)-$(uname -m) -o /usr/local/bin/docker-compose
```

Esta sentencia permitirá mediante el comando curl descarga la versión más actual de docker compose y la ubicará en /usr/local/bin/docker-compose. Desde ese directorio serán ejecutados los comandos para gestionar contenedores.

```
apmt@ALMACEN:~$ docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED
STATUS            PORTS              NAMES
apmt@ALMACEN:~$ docker images
REPOSITORY          TAG                 IMAGE ID            CREATED
SIZE
apmt@ALMACEN:~$ sudo curl -L https://github.com/docker/compose/releases/download/1.21.2/docker-compose-$(uname -s)-$(uname -m) -o /usr/local/bin/docker-compose
[sudo] password for apmt:
apmt is not in the sudoers file. This incident will be reported.
apmt@ALMACEN:~$ su
Contraseña:
root@ALMACEN:/home/apmt# curl -L https://github.com/docker/compose/releases/download/1.21.2/docker-compose-$(uname -s)-$(uname -m) -o /usr/local/bin/docker-comp
ose
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left   Speed
100  617      0  617    0     0    648      0 --:--:-- --:--:-- --:--:--    648
100 10.3M  100 10.3M    0     0  1459k      0  0:00:07  0:00:07 --:--:--  2190k
root@ALMACEN:/home/apmt#
```

- Ahora le damos permisos para que el comando sea ejecutable. Si no se coloca dicha sentencia no se podrá ejecutar el docker compose

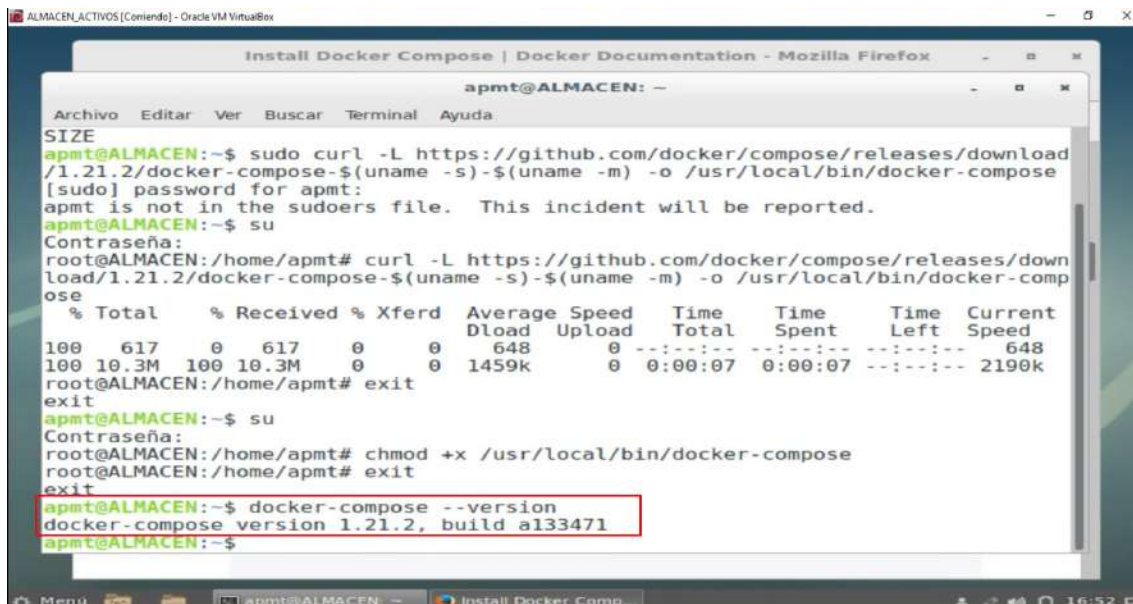
```
$ sudo chmod +x /usr/local/bin/docker-compose
```

- Salimos del usuario root con la sentencia:

```
$ exit
```

- Como usuario apmt, verificamos que este funcionando el docker compose.

```
$ docker-compose -version
```



```
apmt@ALMACEN: ~  
Archivo  Editor  Ver  Buscar  Terminal  Ayuda  
SIZE  
apmt@ALMACEN:~$ sudo curl -L https://github.com/docker/compose/releases/download/1.21.2/docker-compose-$(uname -s)-$(uname -m) -o /usr/local/bin/docker-compose  
[sudo] password for apmt:  
apmt is not in the sudoers file. This incident will be reported.  
apmt@ALMACEN:~$ su  
Contraseña:  
root@ALMACEN:/home/apmt# curl -L https://github.com/docker/compose/releases/download/1.21.2/docker-compose-$(uname -s)-$(uname -m) -o /usr/local/bin/docker-comp  
ose  
% Total % Received % Xferd Average Speed Time Time Time Current  
Dload Upload Total Spent Left Speed  
100 617 0 617 0 0 648 0 --:--:-- --:--:-- --:--:-- 648  
100 10.3M 100 10.3M 0 0 1459k 0 0:00:07 0:00:07 --:--:-- 2190k  
root@ALMACEN:/home/apmt# exit  
exit  
apmt@ALMACEN:~$ su  
Contraseña:  
root@ALMACEN:/home/apmt# chmod +x /usr/local/bin/docker-compose  
root@ALMACEN:/home/apmt# exit  
exit  
apmt@ALMACEN:~$ docker-compose --version  
docker-compose version 1.21.2, build a133471  
apmt@ALMACEN:~$
```

3.4. Instalación del Sistemas de Almacenes y Activos nsiaf de la AGETIC

Una vez concluidos los pasos realizados en los puntos 1,2,3 del presente manual de instalación, nuestro equipo está listo para poder utilizar contenedores de forma segura y con los permisos correspondientes de usuario. Por lo tanto, procedemos a instalar el sistema de almacenes y activos.

- Nos dirigimos al repositorio de la AGETIC ubicado en el siguiente link

<https://gitlab.geo.gob.bo/adsib/nsiaf>

En este link está el repositorio y la explicación de instalación de todo el proceso

- Clonamos el repositorio al home de nuestro usuario apmt mediante el comando git

```
$ git -c http.sslVerify=false clone https://gitlab.geo.gob.bo/adsib/nsiaf.git
```



```
ALMACEN_ACTIVOS [Corriendo] - Oracle VM VirtualBox
docker/almacenes/README.md · agetic-mysql · adsib / nsiaf · GitLab - Mozilla Fi...
apmt@ALMACEN: ~
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
100    617    0    617    0    0    648    0    --:--:--  --:--:--  --:--:--  648
100 10.3M 100 10.3M 0    0    1459k    0    0:00:07  0:00:07  --:--:--  2190k
root@ALMACEN:/home/apmt# exit
exit
apmt@ALMACEN:~$ su
Contraseña:
root@ALMACEN:/home/apmt# chmod +x /usr/local/bin/docker-compose
root@ALMACEN:/home/apmt# exit
exit
apmt@ALMACEN:~$ docker-compose --version
docker-compose version 1.21.2, build a133471
apmt@ALMACEN:~$ ls
apmt@192.168.100.1  Documentos  historia.odt  Música  Público
Descargas          Escritorio  Imágenes     Plantillas  Videos
apmt@ALMACEN:~$ git -c http.sslVerify=false clone https://gitlab.geo.gob.bo/adsib/nsiaf.git
Cloning into 'nsiaf'...
remote: Counting objects: 15792, done.
remote: Compressing objects: 100% (29/29), done.
remote: Total 15792 (delta 15), reused 0 (delta 0)
Receiving objects: 100% (15792/15792), 8.23 MiB | 8.82 MiB/s, done.
Resolving deltas: 100% (11783/11783), done.
apmt@ALMACEN:~$ cd
cp -env:ejemplo -env
```

- Entramos al directorio clonado

```
$ cd nsiaf/
```

- Verificamos el estado de la clonación

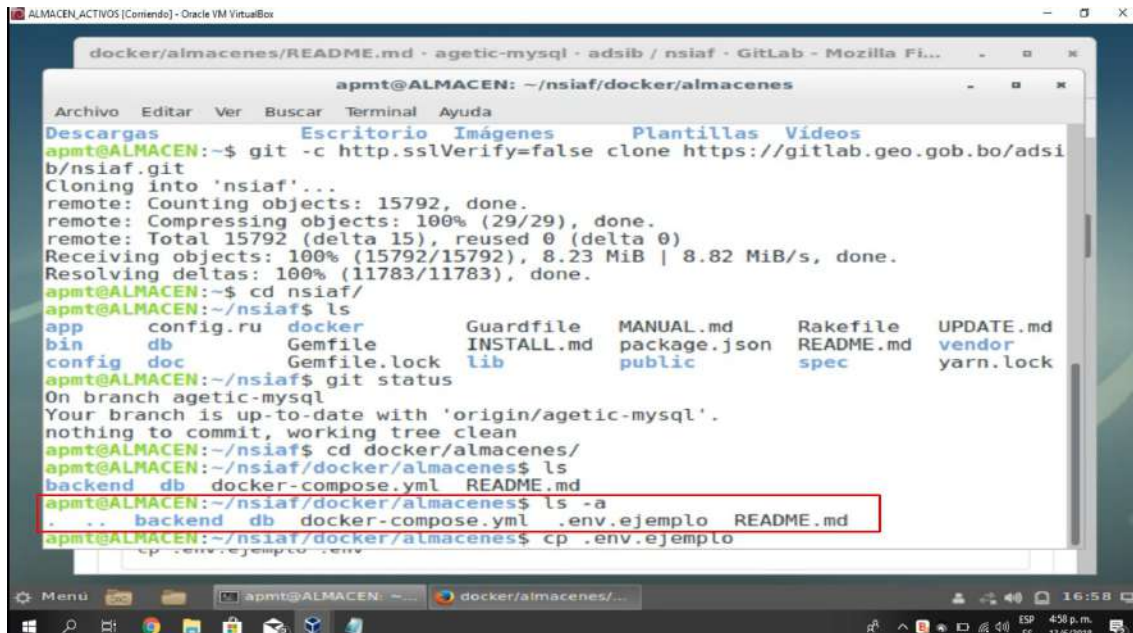
```
$ git status
```

```
ALMACEN_ACTIVOS [Corriendo] - Oracle VM VirtualBox
docker/almacenes/README.md · agetic-mysql · adsib / nsiaf · GitLab - Mozilla Fi...
apmt@ALMACEN: ~/nsiaf
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
exit
apmt@ALMACEN:~$ docker-compose --version
docker-compose version 1.21.2, build a133471
apmt@ALMACEN:~$ ls
apmt@192.168.100.1  Documentos  historia.odt  Música  Público
Descargas          Escritorio  Imágenes     Plantillas  Videos
apmt@ALMACEN:~$ git -c http.sslVerify=false clone https://gitlab.geo.gob.bo/adsib/nsiaf.git
Cloning into 'nsiaf'...
remote: Counting objects: 15792, done.
remote: Compressing objects: 100% (29/29), done.
remote: Total 15792 (delta 15), reused 0 (delta 0)
Receiving objects: 100% (15792/15792), 8.23 MiB | 8.82 MiB/s, done.
Resolving deltas: 100% (11783/11783), done.
apmt@ALMACEN:~$ cd nsiaf/
apmt@ALMACEN:~/nsiaf$ ls
app      config.ru  docker      Guardfile  MANUAL.md  Rakefile  UPDATE.md
bin      db         Gemfile     INSTALL.md  package.json  README.md  vendor
config  doc       Gemfile.lock  lib        public      spec      yarn.lock
apmt@ALMACEN:~/nsiaf$ git status
On branch agetic-mysql
Your branch is up-to-date with 'origin/agetic-mysql'.
nothing to commit, working tree clean
apmt@ALMACEN:~/nsiaf$
```

- Entramos a docker y almacenes

```
/nsiaf$ cd docker/almacenes/
```

```
/nsiaf$ ls -a
```



```

apmt@ALMACEN: ~/nsiaf/docker/almacenes
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
Descargas  Escritorio  Imágenes  Plantillas  Videos
apmt@ALMACEN:~$ git -c http.sslVerify=false clone https://gitlab.geo.gob.bo/adsib/nsiaf.git
Cloning into 'nsiaf'...
remote: Counting objects: 15792, done.
remote: Compressing objects: 100% (29/29), done.
remote: Total 15792 (delta 15), reused 0 (delta 0)
Receiving objects: 100% (15792/15792), 8.23 MiB | 8.82 MiB/s, done.
Resolving deltas: 100% (11783/11783), done.
apmt@ALMACEN:~$ cd nsiaf/
apmt@ALMACEN:~/nsiaf$ ls
app      config.ru  docker      Guardfile  MANUAL.md  Rakefile  UPDATE.md
bin      db         Gemfile     INSTALL.md package.json  README.md  vendor
config  doc        Gemfile.lock  lib         public      spec       yarn.lock
apmt@ALMACEN:~/nsiaf$ git status
On branch agetic-mysql
Your branch is up-to-date with 'origin/agetic-mysql'.
nothing to commit, working tree clean
apmt@ALMACEN:~/nsiaf$ cd docker/almacenes/
apmt@ALMACEN:~/nsiaf/docker/almacenes$ ls
backend  db  docker-compose.yml  README.md
apmt@ALMACEN:~/nsiaf/docker/almacenes$ ls -a
.  ..  backend  db  docker-compose.yml  .env.ejemplo  README.md
apmt@ALMACEN:~/nsiaf/docker/almacenes$ cp .env.ejemplo

```

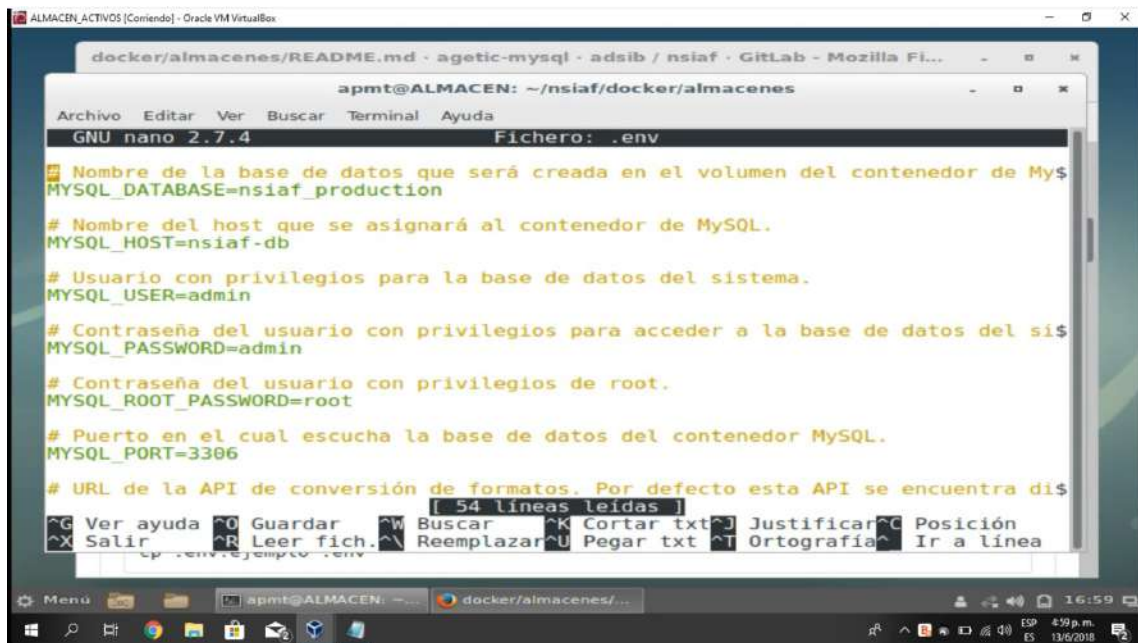
Acá podemos observar que hay 5 archivos

- El backend es el contenedor de la interfaz grafica
- El db es el contenedor de la base de datos mysql
- El archivo docker-compose.yml es el archivo de composición que da parámetros para que pueda levantar los contenedores
- El archivo .env.ejemplo tiene la información requerida para conexiones y configuraciones de los contenedores

- Copiamos el archivo .env.ejemplo con otro nombre y lo editamos

```
$ cp .env.ejemplo .enf
```

```
$ nano .env
```



```
apmt@ALMACEN: ~/nsiaf/docker/almacenes
GNU nano 2.7.4          Fichero: .env

# Nombre de la base de datos que será creada en el volumen del contenedor de MySQL.
MYSQL_DATABASE=nsiaf_production

# Nombre del host que se asignará al contenedor de MySQL.
MYSQL_HOST=nsiaf-db

# Usuario con privilegios para la base de datos del sistema.
MYSQL_USER=admin

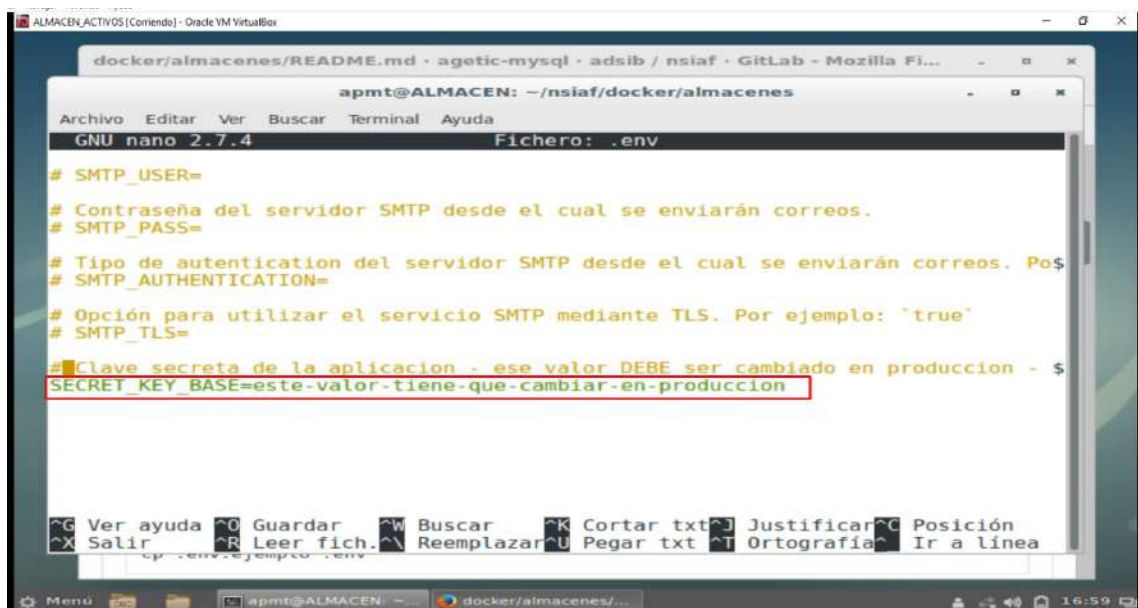
# Contraseña del usuario con privilegios para acceder a la base de datos del sistema.
MYSQL_PASSWORD=admin

# Contraseña del usuario con privilegios de root.
MYSQL_ROOT_PASSWORD=root

# Puerto en el cual escucha la base de datos del contenedor MySQL.
MYSQL_PORT=3306

# URL de la API de conversión de formatos. Por defecto esta API se encuentra dis
[ 54 líneas leídas ]
^G Ver ayuda  ^O Guardar  ^W Buscar    ^K Cortar txt^J Justificar^C Posición
^X Salir      ^R Leer fich.^_ Reemplazar^U Pegar txt ^T Ortografía^_ Ir a línea
```

En esta pantalla podemos ver las configuraciones para mysql que se aplicaran dentro del docker de base de datos. En la parte final del archivo pide SECRET_KEY_BASE que debemos generar pues es una llave única.



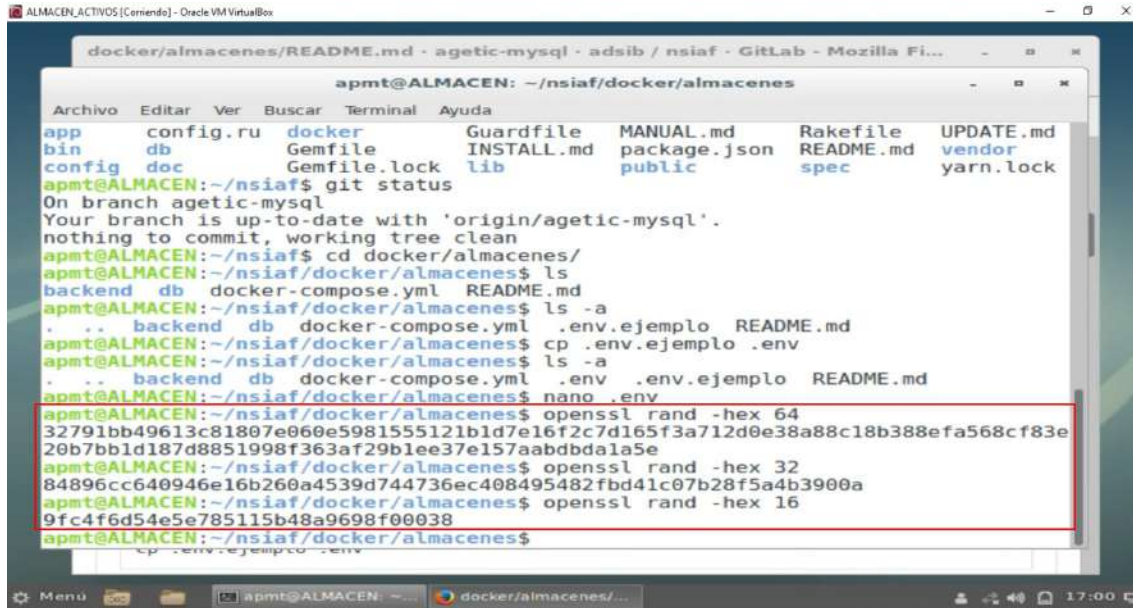
```
apmt@ALMACEN: ~/nsiaf/docker/almacenes
GNU nano 2.7.4          Fichero: .env

# SMTP_USER=
# Contraseña del servidor SMTP desde el cual se enviarán correos.
# SMTP_PASS=
# Tipo de authentication del servidor SMTP desde el cual se enviarán correos. Po$
# SMTP_AUTHENTICATION=
# Opción para utilizar el servicio SMTP mediante TLS. Por ejemplo: 'true'
# SMTP_TLS=
# Clave secreta de la aplicación - ese valor DEBE ser cambiado en produccion - $
SECRET_KEY_BASE=este-valor-tiene-que-cambiar-en-produccion
```

- Generamos la clave encriptada para SECRET_KEY_BASE mediante la sentencia

```
$ openssl rand -hex 64
```

Puede utilizarse otra cantidad de bits (16,32,64,etc)



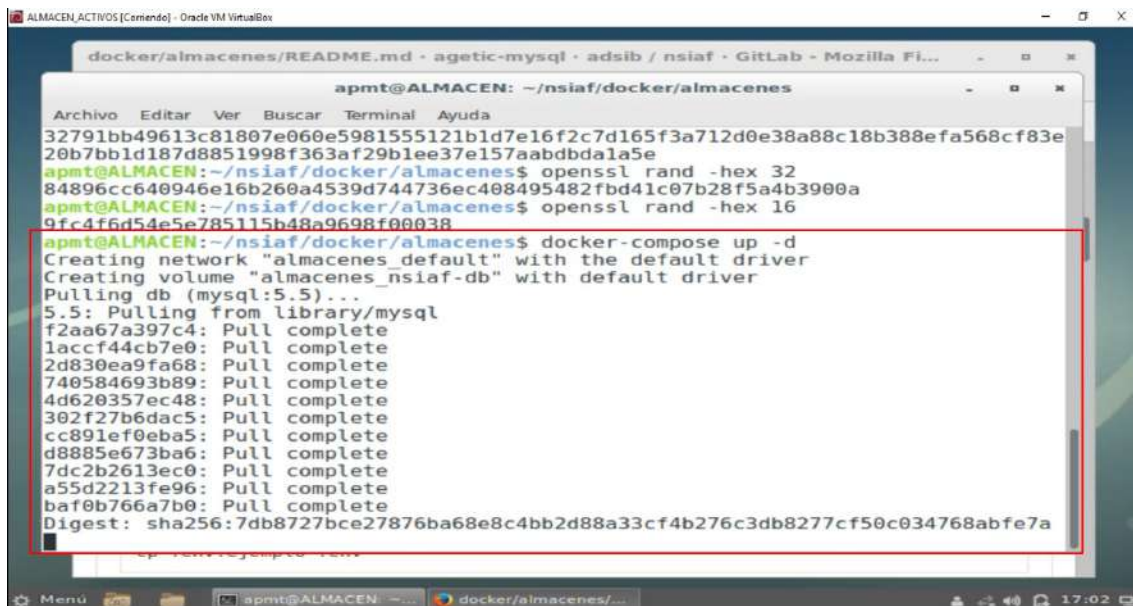
The screenshot shows a terminal window titled 'apmt@ALMACEN: ~/nsiaf/docker/almacenes'. The terminal displays the following commands and output:

```
apmt@ALMACEN:~/nsiaf/docker/almacenes$ git status
On branch agetic-mysql
Your branch is up-to-date with 'origin/agetic-mysql'.
nothing to commit, working tree clean
apmt@ALMACEN:~/nsiaf/docker/almacenes$ cd docker/almacenes/
apmt@ALMACEN:~/nsiaf/docker/almacenes$ ls
backend  db  docker-compose.yml  README.md
apmt@ALMACEN:~/nsiaf/docker/almacenes$ ls -a
.  ..  backend  db  docker-compose.yml  .env.ejemplo  README.md
apmt@ALMACEN:~/nsiaf/docker/almacenes$ cp .env.ejemplo .env
apmt@ALMACEN:~/nsiaf/docker/almacenes$ ls -a
.  ..  backend  db  docker-compose.yml  .env  .env.ejemplo  README.md
apmt@ALMACEN:~/nsiaf/docker/almacenes$ nano .env
apmt@ALMACEN:~/nsiaf/docker/almacenes$ openssl rand -hex 64
32791bb49613c81807e060e5981555121b1d7e16f2c7d165f3a712d0e38a88c18b388efa568cf83e
20b7bb1d187d8851998f363af29b1ee37e157aabdbdala5e
apmt@ALMACEN:~/nsiaf/docker/almacenes$ openssl rand -hex 32
84896ccc640946e16b260a4539d744736ec408495482fbd41c07b28f5a4b3900a
apmt@ALMACEN:~/nsiaf/docker/almacenes$ openssl rand -hex 16
9fc4f6d54e5e785115b48a9698f00038
apmt@ALMACEN:~/nsiaf/docker/almacenes$ cp .env.ejemplo .env
```

The last three lines of the terminal output, which show the generation of random hex strings for the .env file, are highlighted with a red box.

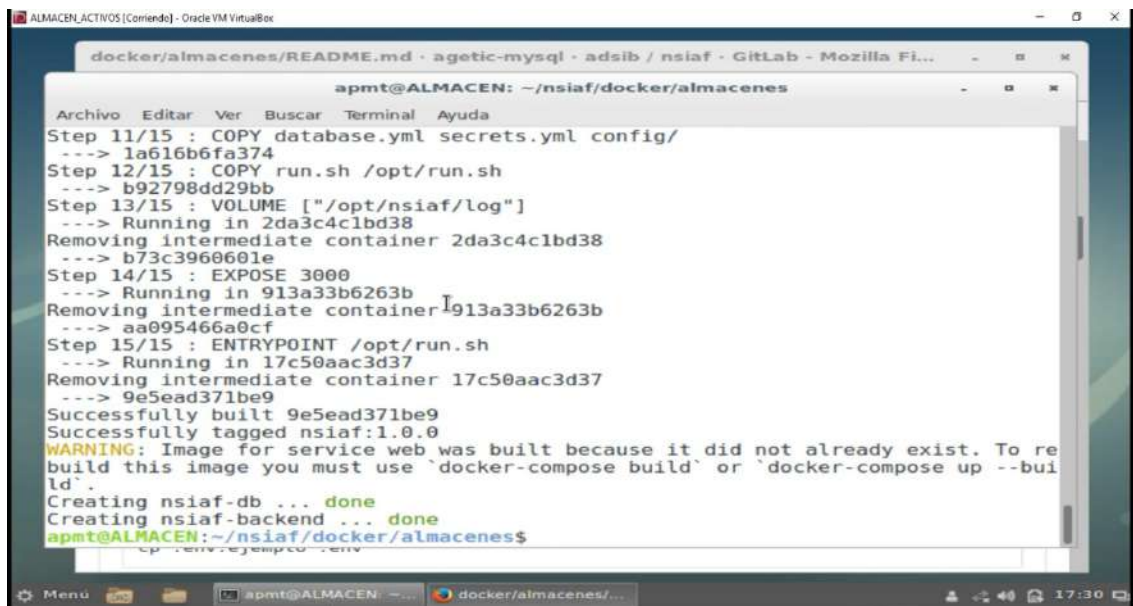
- Marcamos y colocamos la clave encriptada en el archivo .env
- Ahora probamos la instalación (Nota. Estamos en el directorio `/nsiaf/docker/almacenes` \$ desde ahí debe ejecutarse la siguiente sentencia)

`/nsiaf/docker/almacenes $ docker-compose up -d`



```
apmt@ALMACEN: ~/nsiaf/docker/almacenenes
Archivo Editar Ver Buscar Terminal Ayuda
32791bb49613c81807e060e5981555121b1d7e16f2c7d165f3a712d0e38a88c18b388efa568cf83e
20b7bb1d187d8851998f363af29b1ee37e157aabdbdala5e
apmt@ALMACEN:~/nsiaf/docker/almacenenes$ openssl rand -hex 32
84896ccc640946e16b260a4539d744736ec408495482fbd41c07b28f5a4b3900a
apmt@ALMACEN:~/nsiaf/docker/almacenenes$ openssl rand -hex 16
9fc4f6d54e5e785115b48a9698f00038
apmt@ALMACEN:~/nsiaf/docker/almacenenes$ docker-compose up -d
Creating network "almacenenes_default" with the default driver
Creating volume "almacenenes_nsiaf-db" with default driver
Pulling db (mysql:5.5)...
5.5: Pulling from library/mysql
f2aa67a397c4: Pull complete
1accf44cb7e0: Pull complete
2d830ea9fa68: Pull complete
740584693b89: Pull complete
4d620357ec48: Pull complete
302f27b6dac5: Pull complete
cc891ef0eba5: Pull complete
d8885e673ba6: Pull complete
7dc2b2613ec0: Pull complete
a55d2213fe96: Pull complete
baf0b766a7b0: Pull complete
Digest: sha256:7db8727bce27876ba68e8c4bb2d88a33cf4b276c3db8277cf50c034768abfe7a
```

Podemos ver que terminara la descarga y habilitación de los contenedores cuando devuelva el prompt.



```
apmt@ALMACEN: ~/nsiaf/docker/almacenenes
Archivo Editar Ver Buscar Terminal Ayuda
Step 11/15 : COPY database.yml secrets.yml config/
--> 1a616b6fa374
Step 12/15 : COPY run.sh /opt/run.sh
--> b92798dd29bb
Step 13/15 : VOLUME ["/opt/nsiaf/log"]
--> Running in 2da3c4c1bd38
Removing intermediate container 2da3c4c1bd38
--> b73c3960601e
Step 14/15 : EXPOSE 3000
--> Running in 913a33b6263b
Removing intermediate container 913a33b6263b
--> aa095466a0cf
Step 15/15 : ENTRYPOINT /opt/run.sh
--> Running in 17c50aac3d37
Removing intermediate container 17c50aac3d37
--> 9e5ead371be9
Successfully built 9e5ead371be9
Successfully tagged nsiaf:1.0.0
WARNING: Image for service web was built because it did not already exist. To re
build this image you must use `docker-compose build` or `docker-compose up --buil
d`.
Creating nsiaf-db ... done
Creating nsiaf-backend ... done
apmt@ALMACEN:~/nsiaf/docker/almacenenes$
```

- Ahora podemos ver si las imágenes de los contenedores fueron montadas en nuestro docker y si los servicios están activos.

\$ docker ps

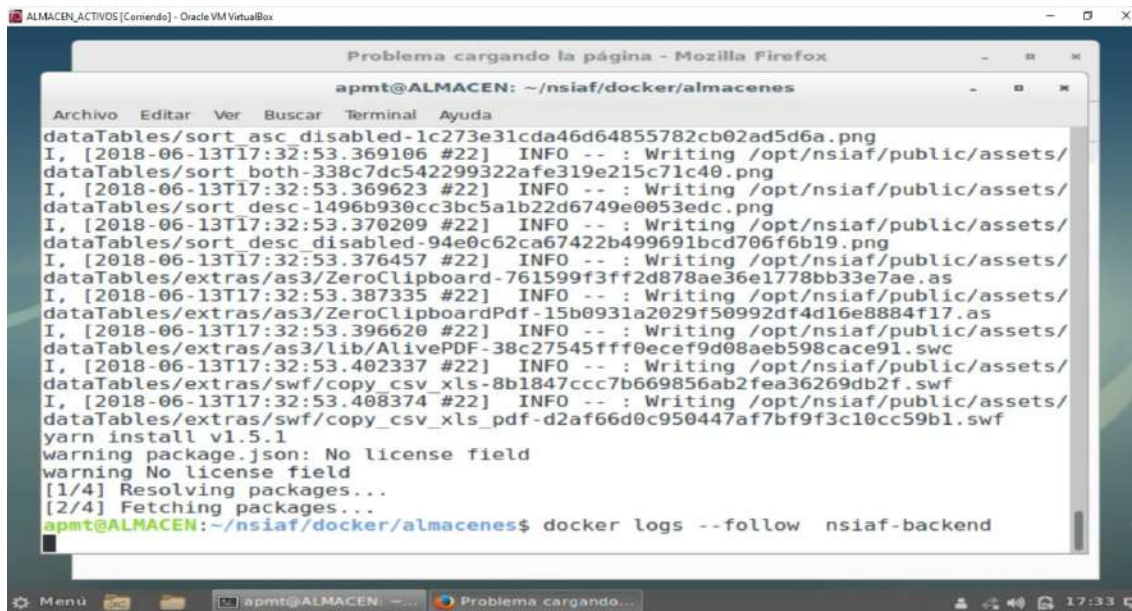

```
apmt@ALMACEN: ~/nsiaf/docker/almacenenes
CONTAINER ID   IMAGE          PORTS          NAMES          CREATED
d07e41586410   nsiaf:1.0.0    0.0.0.0:8888->3000/tcp  nsiaf-backend  About a minute
16dacc5039e5   mysql:5.5      3306/tcp      nsiaf-db        About a minute
```

En la gráfica vemos que el backend (interfaz del usuario) esta utilizable bajo el puerto 8888 que hace puente dentro del contenedor con el puerto 3000. Es decir que si colocamos en el navegador localhost:8888 nuestro sistema redirigirá al contenedor al puerto 3000 donde se encuentra levantado el servicio.

Nota: pese a que devuelve el prompt, es necesario esperar un momento más porque seguirá haciendo configuraciones en segundo plano, una vez terminadas las mismas se levantarán los servicios.

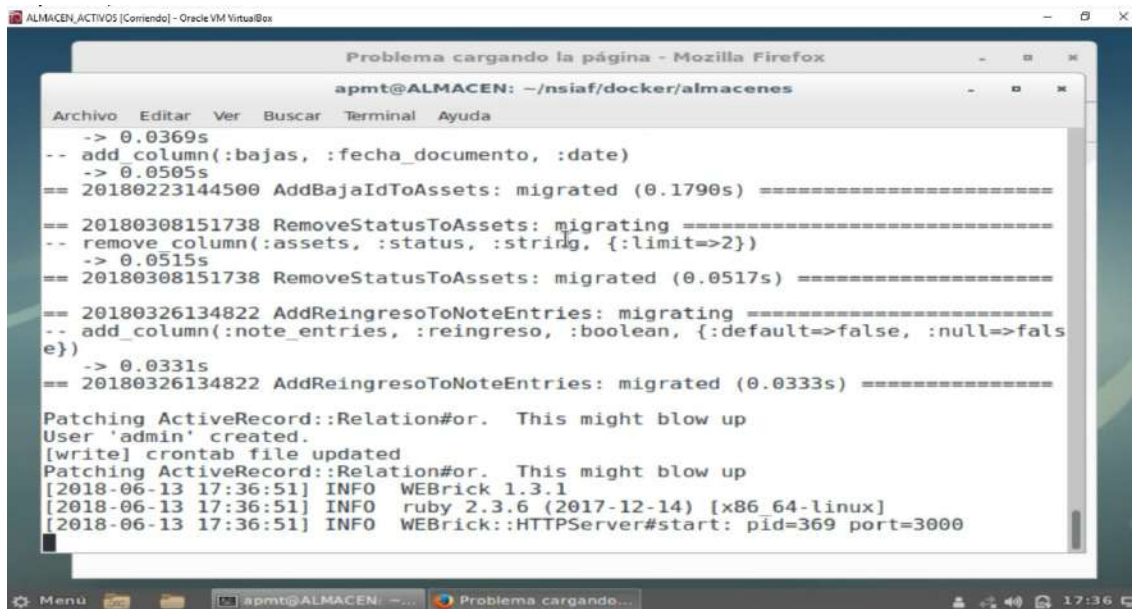
- Verificamos el estado de los contenedores mediante los logs

```
$ docker logs --follow nsiaf-backend
```



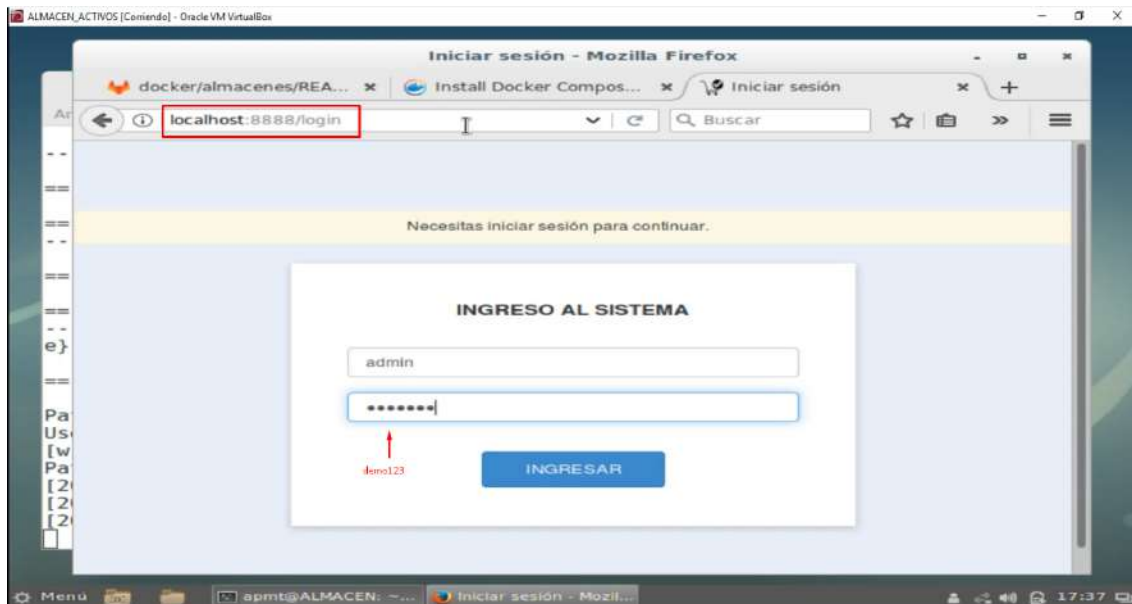
```
ALMACEN_ACTIVOS [Corriendo] - Oracle VM VirtualBox
Problema cargando la página - Mozilla Firefox
apmt@ALMACEN: ~/nsiaf/docker/almacenes
Archivo Editar Ver Buscar Terminal Ayuda
dataTables/sort asc disabled-1c273e31cda46d64855782cb02ad5d6a.png
I, [2018-06-13T17:32:53.369106 #22] INFO -- : Writing /opt/nsiaf/public/assets/
dataTables/sort both-338c7dc542299322afe319e215c71c40.png
I, [2018-06-13T17:32:53.369623 #22] INFO -- : Writing /opt/nsiaf/public/assets/
dataTables/sort desc-1496b930cc3bc5a1b22d6749e0053edc.png
I, [2018-06-13T17:32:53.370209 #22] INFO -- : Writing /opt/nsiaf/public/assets/
dataTables/sort desc disabled-94e0c62ca67422b499691bcd706f6b19.png
I, [2018-06-13T17:32:53.376457 #22] INFO -- : Writing /opt/nsiaf/public/assets/
dataTables/extras/as3/ZeroClipboard-761599f3ff2d878ae36e1778bb33e7ae.as
I, [2018-06-13T17:32:53.387335 #22] INFO -- : Writing /opt/nsiaf/public/assets/
dataTables/extras/as3/ZeroClipboardPdf-15b0931a2029f50992df4d16e8884f17.as
I, [2018-06-13T17:32:53.396620 #22] INFO -- : Writing /opt/nsiaf/public/assets/
dataTables/extras/as3/lib/AlivePDF-38c27545fff0ecef9d08aeb598cace91.swc
I, [2018-06-13T17:32:53.402337 #22] INFO -- : Writing /opt/nsiaf/public/assets/
dataTables/extras/swf/copy_csv_xls-8b1847ccc7b669856ab2fea36269db2f.swf
I, [2018-06-13T17:32:53.408374 #22] INFO -- : Writing /opt/nsiaf/public/assets/
dataTables/extras/swf/copy_csv_xls_pdf-d2af66d0c950447af7bf9f3c10cc59b1.swf
yarn install v1.5.1
warning package.json: No license field
warning No license field
[1/4] Resolving packages...
[2/4] Fetching packages...
apmt@ALMACEN:~/nsiaf/docker/almacenes$ docker logs --follow nsiaf-backend
```

Nota. El comando follow permite ver los logs constantemente sin devolver la línea del prompt. Para salir debemos precionar ctrl+c



```
ALMACEN_ACTIVOS [Corriendo] - Oracle VM VirtualBox
Problema cargando la página - Mozilla Firefox
apmt@ALMACEN: ~/nsiaf/docker/almacenes
Archivo Editar Ver Buscar Terminal Ayuda
-> 0.0369s
-- add_column(:bajas, :fecha_documento, :date)
-> 0.0505s
== 20180223144500 AddBajaIdToAssets: migrated (0.1790s) =====
== 20180308151738 RemoveStatusToAssets: migrating =====
-- remove_column(:assets, :status, :string, {:limit=>2})
-> 0.0515s
== 20180308151738 RemoveStatusToAssets: migrated (0.0517s) =====
== 20180326134822 AddReingresoToNoteEntries: migrating =====
-- add_column(:note_entries, :reingreso, :boolean, {:default=>false, :null=>false})
-> 0.0331s
== 20180326134822 AddReingresoToNoteEntries: migrated (0.0333s) =====
Patching ActiveRecord::Relation#or. This might blow up
User 'admin' created.
[write] crontab file updated
Patching ActiveRecord::Relation#or. This might blow up
[2018-06-13 17:36:51] INFO WEBrick 1.3.1
[2018-06-13 17:36:51] INFO ruby 2.3.6 (2017-12-14) [x86_64-linux]
[2018-06-13 17:36:51] INFO WEBrick::HTTPServer#start: pid=369 port=3000
```

- Ahora que se levantó el HTTPServer en el puerto 3000 procedemos a probar por un navegador web.

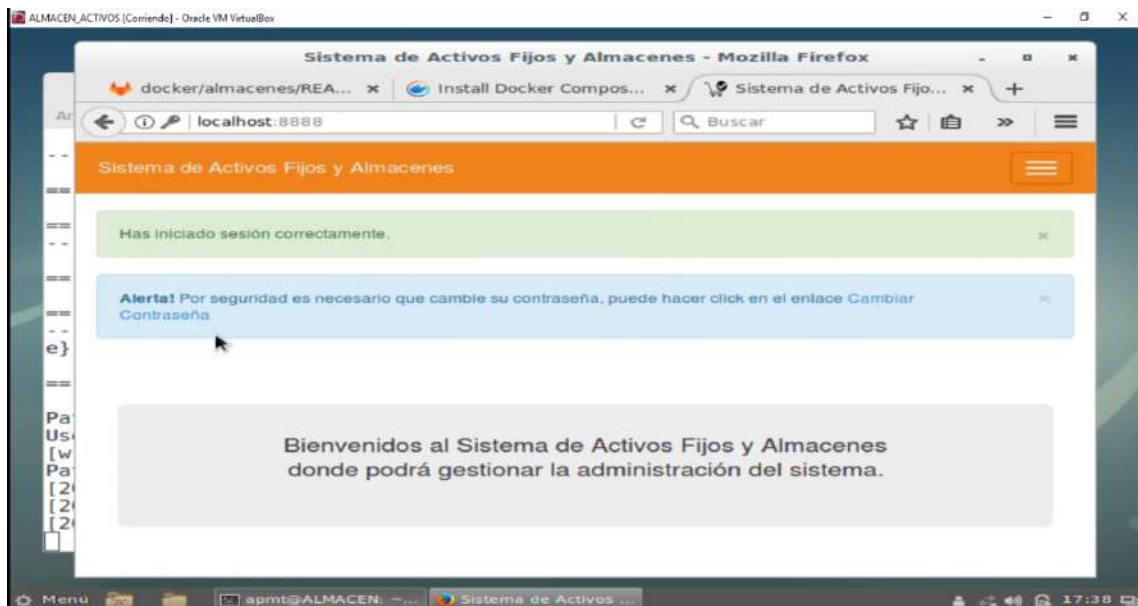


En la parte que dice localhost podemos usar también el IP de la maquina donde se encuentra instalado el contenedor o hacer DNAT con una IP publica de la institución mediante el firewall.

Como se muestra en la gráfica pedirá usuario estos son:

- **Usuario** : admin
- **Contraseña** : demo123

Nota. Se recomienda cambiar la contraseña de administrador por seguridad



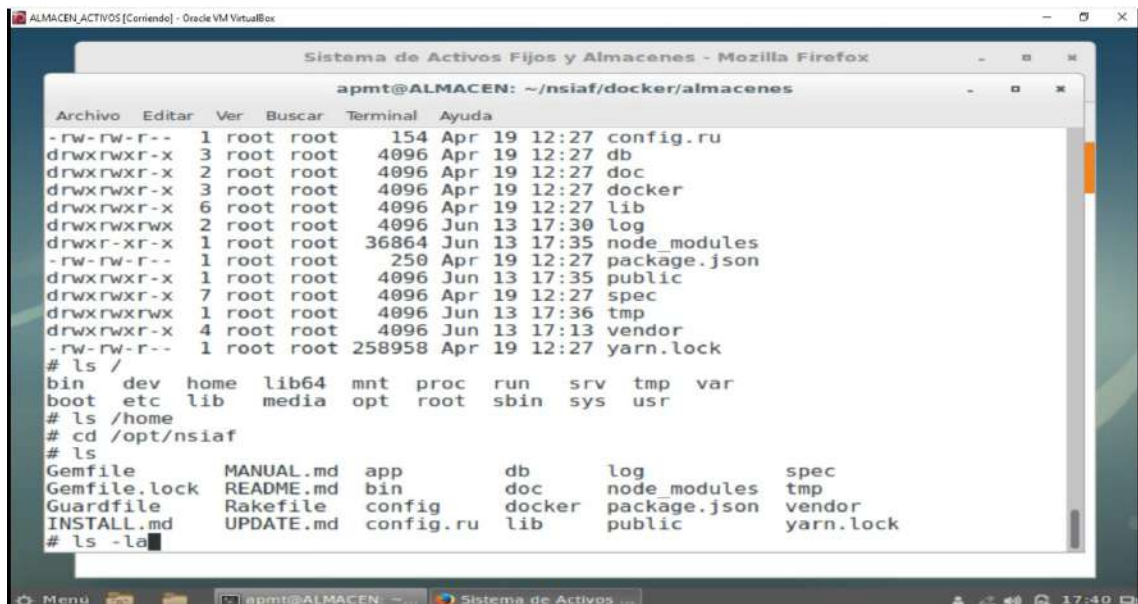
Finalmente, el sistema se encuentra instalado y listo para su uso.

4. Comando útil para manipulación de docker

- Podemos abrir una consola ssh dentro del contenedor

```
$ docker exec -it nsiaf-backend sh
```

Usando el anterior comando ya podemos ver todos los archivos que están en el contenedor nsiaf-backend. También podemos ver el código fuente del mismo

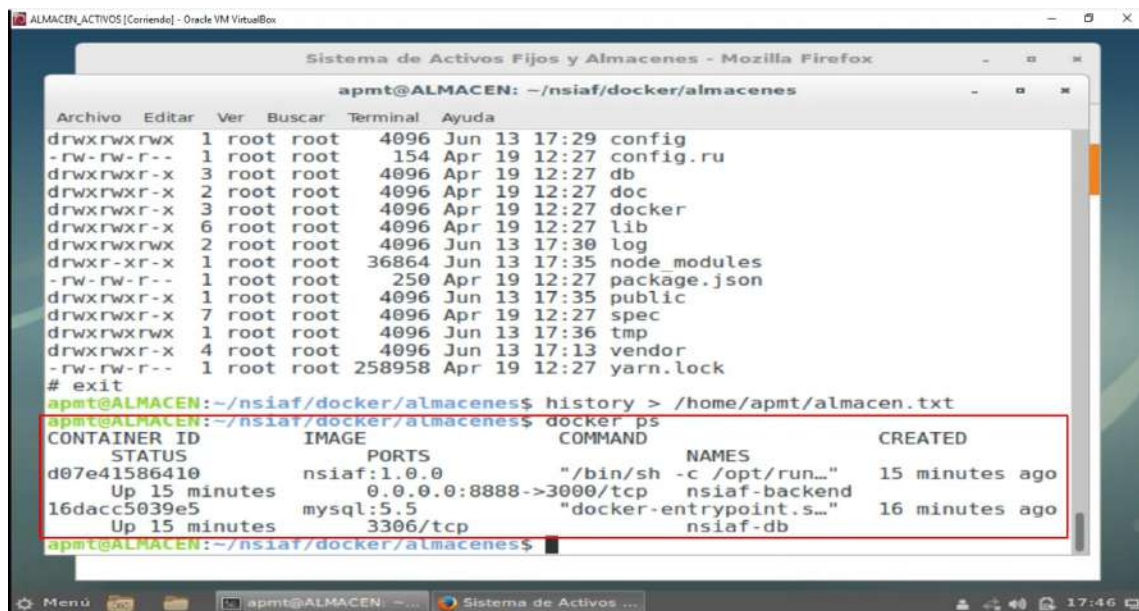


Como se muestra en la grafica, dentro de la carpeta /opt/nsiaf/ se encuentra el codigo fuente del sistema de activos y almacen echo en ruby

Para salir de la consola se usa el comando exit.

- Podemos ver los contenedores levantados y porque puerto se conectan.

\$ docker ps

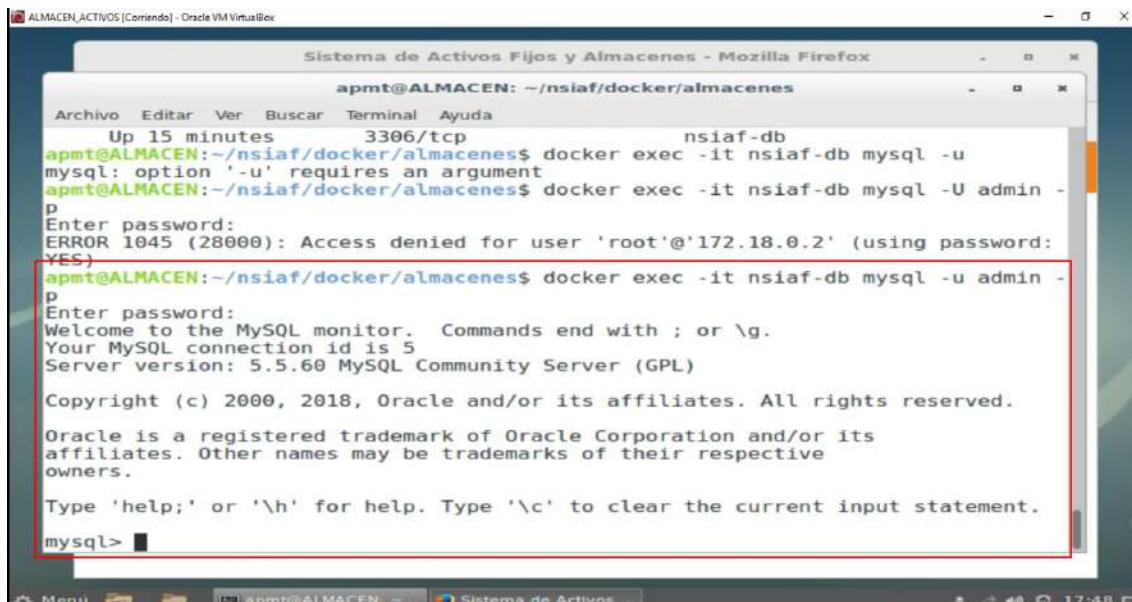


```
apmt@ALMACEN: ~/nsiaf/docker/almacenes
drwxrwxrwx 1 root root 4096 Jun 13 17:29 config
-rw-rw-r-- 1 root root 154 Apr 19 12:27 config.ru
drwxrwxr-x 3 root root 4096 Apr 19 12:27 db
drwxrwxr-x 2 root root 4096 Apr 19 12:27 doc
drwxrwxr-x 3 root root 4096 Apr 19 12:27 docker
drwxrwxr-x 6 root root 4096 Apr 19 12:27 lib
drwxrwxrwx 2 root root 4096 Jun 13 17:30 log
drwxr-xr-x 1 root root 36864 Jun 13 17:35 node_modules
-rw-rw-r-- 1 root root 250 Apr 19 12:27 package.json
drwxrwxr-x 1 root root 4096 Jun 13 17:35 public
drwxrwxr-x 7 root root 4096 Apr 19 12:27 spec
drwxrwxrwx 1 root root 4096 Jun 13 17:36 tmp
drwxrwxr-x 4 root root 4096 Jun 13 17:13 vendor
-rw-rw-r-- 1 root root 258958 Apr 19 12:27 yarn.lock
# exit
apmt@ALMACEN:~/nsiaf/docker/almacenes$ history > /home/apmt/almacen.txt
apmt@ALMACEN:~/nsiaf/docker/almacenes$ docker ps
CONTAINER ID        IMAGE               COMMAND             NAMES                  CREATED
d07e41586410       nsiaf:1.0.0        "/bin/sh -c /opt/run..." nsiaf-backend         15 minutes ago
16dacc5039e5       mysql:5.5           "docker-entrypoint.s..." nsiaf-db               16 minutes ago
apmt@ALMACEN:~/nsiaf/docker/almacenes$
```

En la grafica tenemos levantado 2 contenedores el nsiaf-backend que tiene el código fuente web y se accede por el puerto 8888 sin embargo dentro del contenedor el servicio web se encuentra en el puerto 3000.

- Podemos entrar en modo consola mediante docker en el contenedor del mysql, directamente a la base de datos

\$ docker exec -it nsiaf-db mysql -u admin -p



```
apmt@ALMACEN: ~/nsiaf/docker/almacenenes
Up 15 minutes 3306/tcp nsiaf-db
apmt@ALMACEN:~/nsiaf/docker/almacenenes$ docker exec -it nsiaf-db mysql -u
mysql: option '-u' requires an argument
apmt@ALMACEN:~/nsiaf/docker/almacenenes$ docker exec -it nsiaf-db mysql -U admin -
p
Enter password:
ERROR 1045 (28000): Access denied for user 'root'@'172.18.0.2' (using password:
YES)
apmt@ALMACEN:~/nsiaf/docker/almacenenes$ docker exec -it nsiaf-db mysql -u admin -
p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 5
Server version: 5.5.60 MySQL Community Server (GPL)

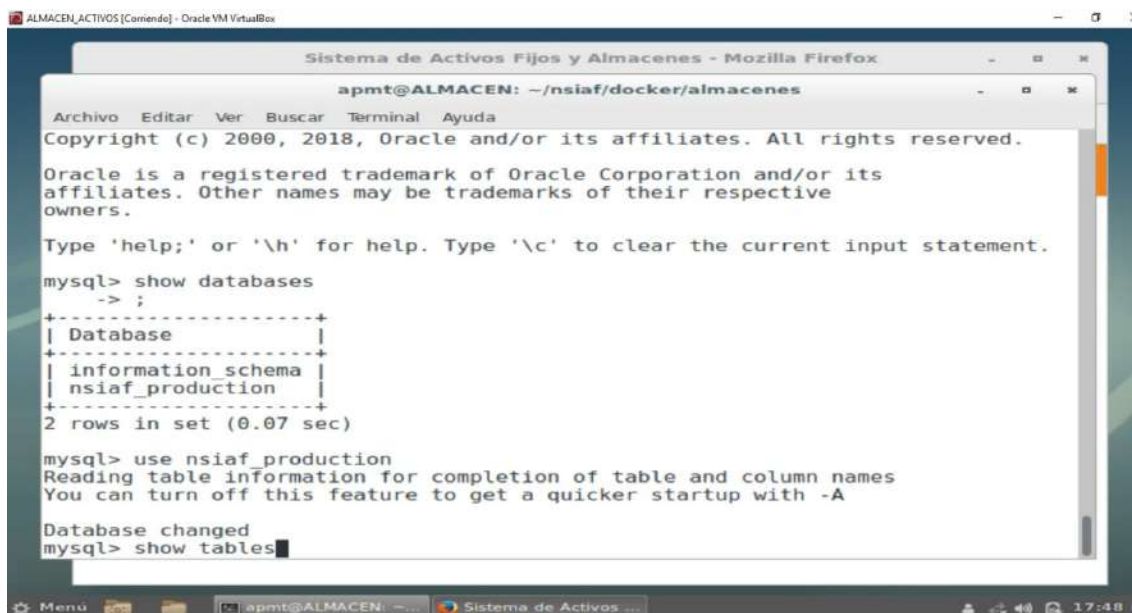
Copyright (c) 2000, 2018, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

Como vemos en la gráfica nos pide la contraseña para el caso del ejemplo también es admin (Esta contraseña dependerá de lo que se cambie en el archivo docker-compose.yml antes de levantar los contenedores). Ahora podemos ver o crear bases de datos, tablas, etc.



```
Copyright (c) 2000, 2018, Oracle and/or its affiliates. All rights reserved.

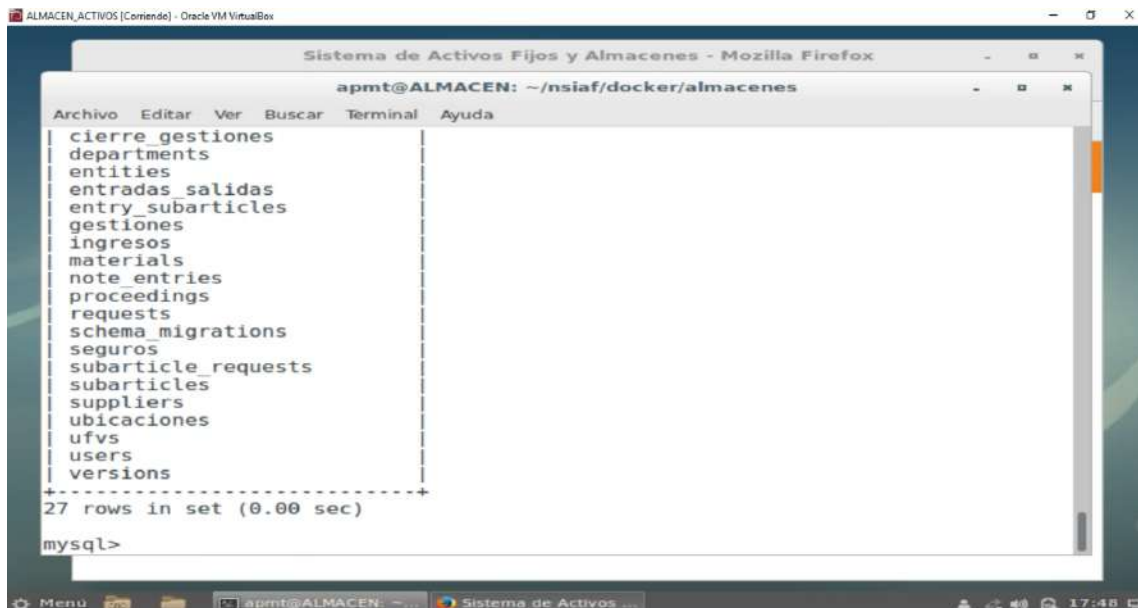
Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show databases
->
+-----+
| Database |
+-----+
| information schema |
| nsiaf_production |
+-----+
2 rows in set (0.07 sec)

mysql> use nsiaf_production
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

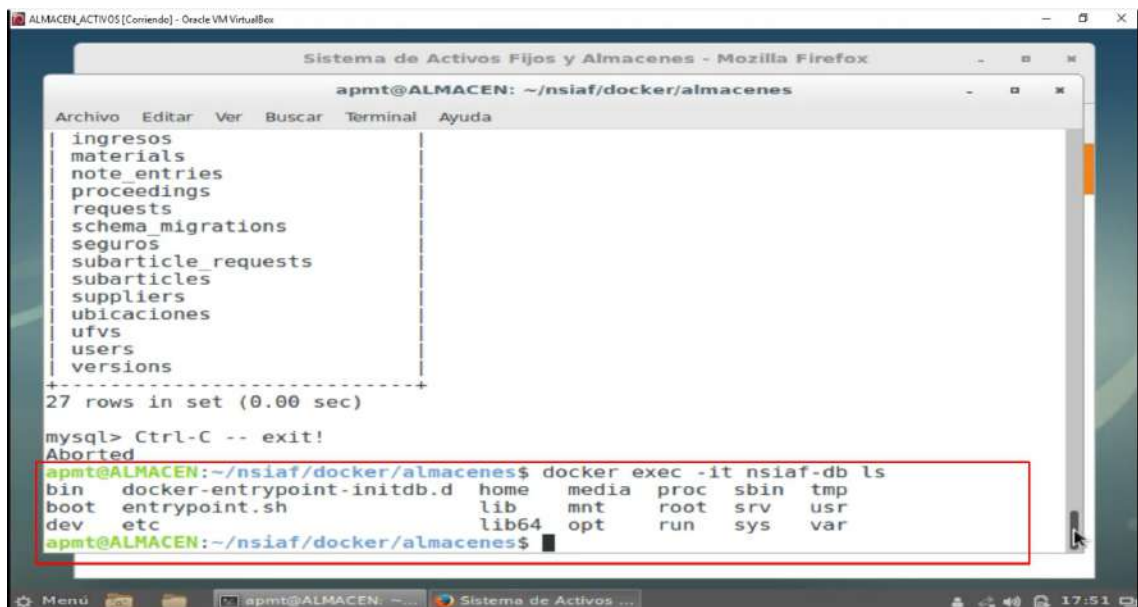
Database changed
mysql> show tables
```



Se puede salir de la base de datos con ctrl+c

- Para ver los archivos y directorios contenidos en el contenedor podemos usar el comando ls.

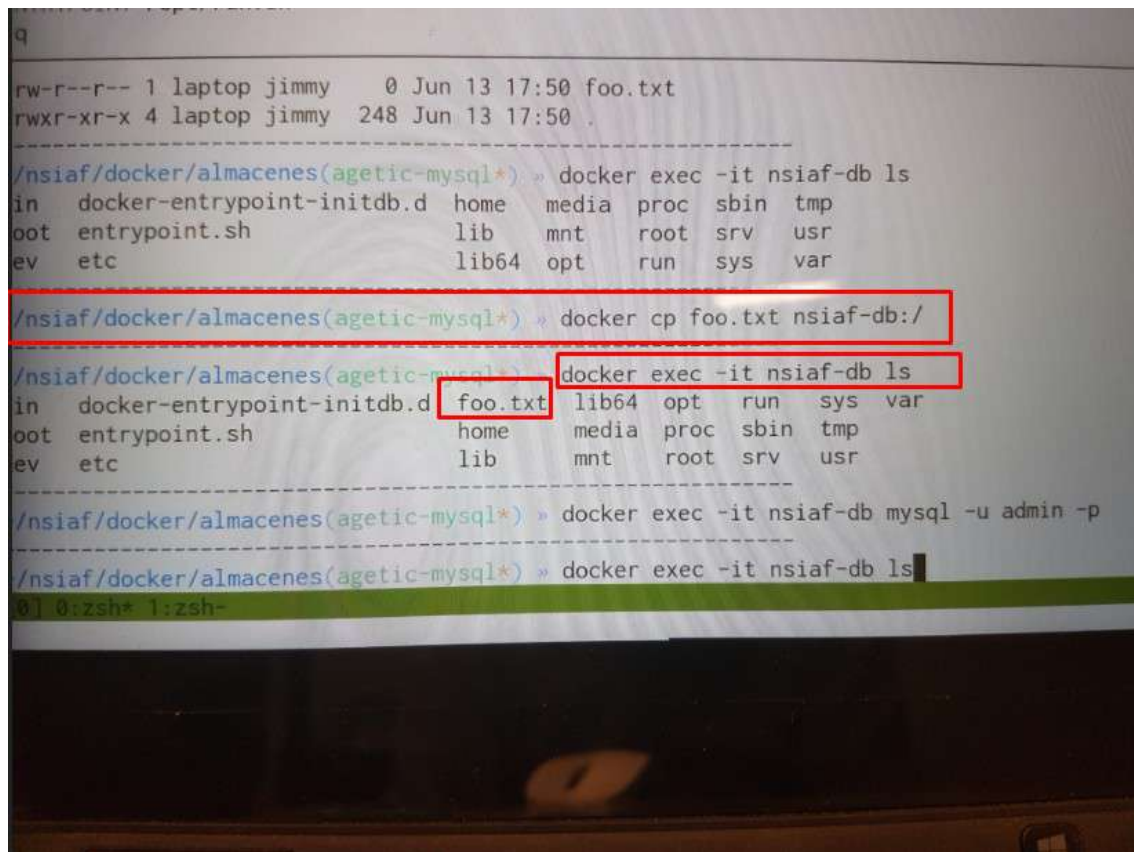
\$ docker exec -it nsiaf-db ls



- Podemos copiar un archivo SQL dentro del contenedor para importarlo a la base de datos

```
$ docker cp foo.txt nsiaf-db:/
```

En la línea de comando anterior indicamos al docker que copie el archivo foo.txt al contenedor nsiaf-db (El archivo foo.txt podemos cambiarlo por un archivo sql que ya tengamos listo para migrar)



```
q
rw-r--r-- 1 laptop jimmy  0 Jun 13 17:50 foo.txt
rwxr-xr-x 4 laptop jimmy 248 Jun 13 17:50 .
-----
/nsiaf/docker/almacenes(agetica-mysql*) » docker exec -it nsiaf-db ls
in  docker-entrypoint-initdb.d  home  media  proc  sbin  tmp
oot  entrypoint.sh              lib   mnt    root  srv   usr
ev  etc                        lib64 opt    run   sys   var
-----
/nsiaf/docker/almacenes(agetica-mysql*) » docker cp foo.txt nsiaf-db:/
/nsiaf/docker/almacenes(agetica-mysql*) » docker exec -it nsiaf-db ls
in  docker-entrypoint-initdb.d  foo.txt  lib64  opt    run   sys   var
oot  entrypoint.sh              home     media  proc  sbin  tmp
ev  etc                        lib      mnt    root  srv   usr
-----
/nsiaf/docker/almacenes(agetica-mysql*) » docker exec -it nsiaf-db mysql -u admin -p
/nsiaf/docker/almacenes(agetica-mysql*) » docker exec -it nsiaf-db ls
0] 0:zsh* 1:zsh-
```

Para ver que a sido copiado podemos ejecutar la sentencia ls como se muestra en el gráfico.

- En la siguiente grafica se puede ver que el nsiaf-backend puede conectarse desde cualquier equipo por el puerto 8888 que enmascara al puerto 3000 que se encuentra dentro del contenedor. En el caso del nsiaf-db no tiene conexión de salida, es decir, que no se puede conectar desde cualquier equipo sino que el nsiaf-backend es el único (Mediante su interfaz gráfica) que puede conectarse a la base de datos.

```
apmt@ALMACEN: ~/nsiaf/docker/almacenes
subarticle_requests
subarticles
suppliers
ubicaciones
ufvs
users
versions
+-----+
27 rows in set (0.00 sec)

mysql> Ctrl-C -- exit!
Aborted
apmt@ALMACEN:~/nsiaf/docker/almacenes$ docker exec -it nsiaf-db ls
bin  docker-entrypoint-initdb.d  home  media  proc  sbin  tmp
boot  entrypoint.sh                lib   mnt    root  srv   usr
dev   etc                          lib64 opt    run    sys   var
apmt@ALMACEN:~/nsiaf/docker/almacenes$ docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED
STATUS
d07e41586410       nsiaf:1.0.0        "/bin/sh -c /opt/run..." 27 minutes ago
Up 27 minutes      0.0.0.0:8888->3000/tcp  nsiaf-backend
16dacc5039e5       mysql:5.5          "docker-entrypoint.s..." 27 minutes ago
Up 27 minutes      3306/tcp           nsiaf-db
apmt@ALMACEN:~/nsiaf/docker/almacenes$
```

Para solucionar esto editamos el archivo docker-compose.yml y adicionamos el puerto en el ejemplo 4444 con el fin que enmascare al puerto 3306 dentro del contenedor nsiaf-db (Contenedor de la base de datos).

```
apmt@ALMACEN: ~/nsiaf/docker/almacenes
GNU nano 2.7.4          Fichero: docker-compose.yml

Version: '3'
services:
  web:
    build:
      context: ./backend
      dockerfile: ./Dockerfile
    image: nsiaf:1.0.0
    container_name: nsiaf-backend
    ports:
      - "8888:3000"
    depends_on:
      - db
    env_file: .env
    restart: always
  db:
    image: mysql:5.5
    container_name: nsiaf-db
    ports:
      - "4444:3306"
    volumes:
      - nsiaf-db:/var/lib/mysql
    env_file: .env
    restart: always
```

Nuevamente ejecutamos la sentencia Docker-compose con el fin que se actualicen los cambios. (La sentencia debe ejecutarse dentro del directorio /nsiaf/Docker/almacenes)

```
$ docker-compose up -d
```



```
apmt@ALMACEN: ~/nsiaf/docker/almacenes
tcp        0      0 0.0.0.0:80->0.0.0.0:80  LISTEN  -
tcp6       0      0 :::80->:::80          LISTEN  -
tcp6       0      0 :::22->:::22          LISTEN  -
tcp6       0      0 :::1:631->:::1:631    LISTEN  -
tcp6       0      0 :::8888->:::8888      LISTEN  -

apmt@ALMACEN:~/nsiaf/docker/almacenes$ docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED            STATUS
d07e41586410       nsiaf:1.0.0        "/bin/sh -c /opt/run..." 3 weeks ago        Up
30 minutes         0.0.0.0:8888->3000/tcp nsiaf-backend
16dacc5039e5       mysql:5.5           "docker-entrypoint.s..." 3 weeks ago        Up
30 minutes         3306/tcp            nsiaf-db

apmt@ALMACEN:~/nsiaf/docker/almacenes$ docker-compose up -d
Recreating nsiaf-db ... done
Recreating nsiaf-backend ... done

apmt@ALMACEN:~/nsiaf/docker/almacenes$ docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED            STATUS
13618da636ed       nsiaf:1.0.0        "/bin/sh -c /opt/run..." 51 seconds ago     Up
p 49 seconds       0.0.0.0:8888->3000/tcp nsiaf-backend
3258acf6dc82       mysql:5.5           "docker-entrypoint.s..." About a minute ago  Up
p About a minute   0.0.0.0:4444->3306/tcp nsiaf-db

apmt@ALMACEN:~/nsiaf/docker/almacenes$
```

Finalmente verificamos si ya tenemos acceso mediante el puerto 4444 con el comando docker ps.

\$ docker ps

```
apmt@ALMACEN: ~/nsiaf/docker/almacenes
tcp        0      0 0.0.0.0:80->0.0.0.0:80  LISTEN  -
tcp6       0      0 :::80->:::80          LISTEN  -
tcp6       0      0 :::22->:::22          LISTEN  -
tcp6       0      0 :::1:631->:::1:631    LISTEN  -
tcp6       0      0 :::8888->:::8888      LISTEN  -

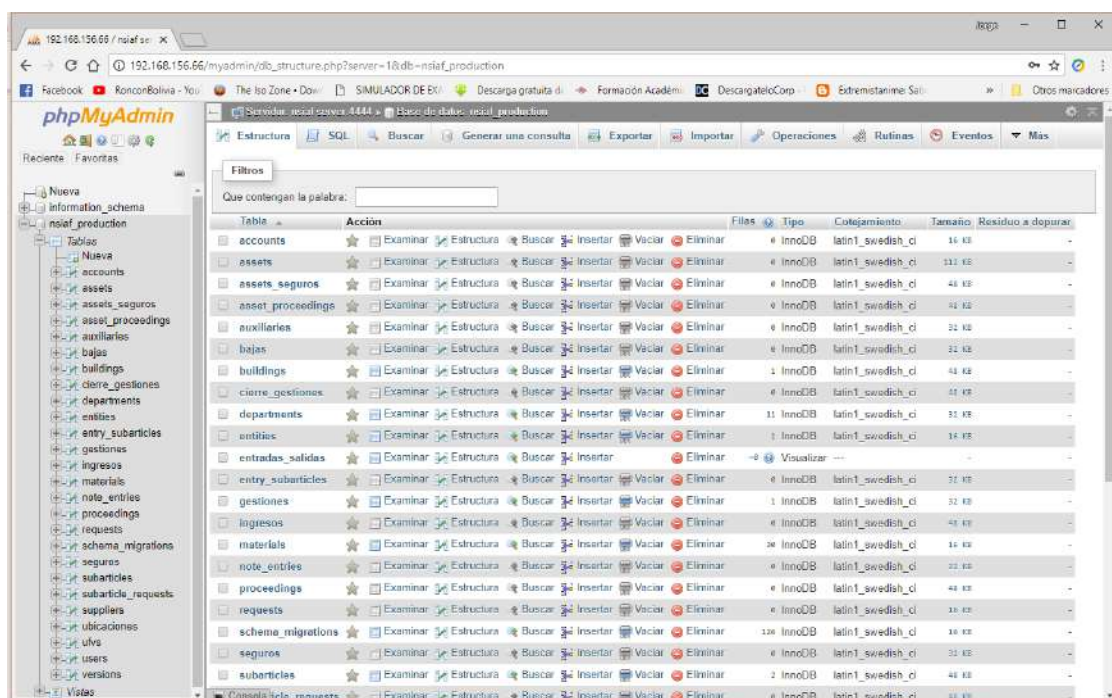
apmt@ALMACEN:~/nsiaf/docker/almacenes$ docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED            STATUS
d07e41586410       nsiaf:1.0.0        "/bin/sh -c /opt/run..." 3 weeks ago        Up
30 minutes         0.0.0.0:8888->3000/tcp nsiaf-backend
16dacc5039e5       mysql:5.5           "docker-entrypoint.s..." 3 weeks ago        Up
30 minutes         3306/tcp            nsiaf-db

apmt@ALMACEN:~/nsiaf/docker/almacenes$ docker-compose up -d
Recreating nsiaf-db ... done
Recreating nsiaf-backend ... done

apmt@ALMACEN:~/nsiaf/docker/almacenes$ docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED            STATUS
13618da636ed       nsiaf:1.0.0        "/bin/sh -c /opt/run..." 51 seconds ago     Up
p 49 seconds       0.0.0.0:8888->3000/tcp nsiaf-backend
3258acf6dc82       mysql:5.5           "docker-entrypoint.s..." About a minute ago  Up
p About a minute   0.0.0.0:4444->3306/tcp nsiaf-db

apmt@ALMACEN:~/nsiaf/docker/almacenes$
```

Finalmente en un administrador visual de base de datos (Ej: phpmyadmin) configuramos la ip del equipo cliente y su respectivo puerto (en el ejemplo 4444) para poder acceder a la base de datos.



5. Conclusiones

Con el sistema de Activos Fijos y Almacenes proporcionado por la AGETIC e instalado en los servidores de la APMT, se podrá administrar y gestionar de forma eficaz y eficiente tanto los materiales contenidos en almacenes como los activos asignados dentro de la APMT