

Trabajo Final

Laboratorio de Sistemas Operativos y Redes



Integrantes:

- Soares, Nicole
- Rodriguez Fontana, Abril

Cursada:

- 2do cuatrimestre de 2024

Profesor:

- José Luis Di Biase

Introducción	3
¿Cual es el objetivo del proyecto?	3
¿Qué es Mayan EDMS?	3
¿Qué es Docker?	3
Instalación	3
Ejecución	4
Ejemplos de uso	5
Conclusión	7

Introducción

¿Cual es el objetivo del proyecto?

El objetivo de este proyecto es realizar efectivamente la instalación, configuración y puesta en marcha del sistema de gestión Mayan EDMS como servicio de red utilizando las prácticas aprendidas dentro de la materia “Laboratorio de Sistemas Operativos y Redes” y usando Linux como sistema operativo, en este caso la distribución “Ubuntu”.

¿Qué es Mayan EDMS?

Mayan EDMS es un sistema de gestión de documentos de código abierto, el cual es usado por las organizaciones para almacenar, organizar y gestionar documentos digitales de manera eficiente.

¿Qué es Docker?

Docker es un proyecto de código abierto que permite crear, ejecutar y gestionar aplicaciones dentro de contenedores. Estos contenedores incluyen todo lo necesario para que una aplicación funcione, asegurando que se ejecute de la misma forma en cualquier lugar.

Instalación

La instalación la realizaremos dentro de la terminal escribiendo el código línea por línea.

En primera instancia vamos a realizar una actualización del sistema. Descargamos las versiones más recientes de los paquetes y dependencias necesarias con este comando:

```
sudo apt update  
sudo apt upgrade
```

Una vez esté todo actualizado vamos a crear una carpeta para empezar a descargar nuestros archivos ahí.

Luego de eso vamos a realizar la instalación de Docker, necesario para la posterior instalación de Mayan EDMS:

```
wget -qO- https://get.docker.com/ | sh
```

Después una vez finalizada la instalación, vamos a levantar los contenedores con el comando:

```
systemctl start docker
```

Puede ser que salte el siguiente error de que los permisos fueron denegados, lo resolvemos poniendo sudo delante de los comandos para poder ejecutar los comandos como administradores del sistema.

```
root@root:~# systemctl start docker
permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Get "http://%2Fvar%2Frun%2Fdocker.sock/v1.47/containers/json": dial unix /var/run/docker.sock: connect: permission denied
```

El siguiente paso es automatizar el inicio de docker para no tener la necesidad de escribir el comando anterior manualmente cada vez que queramos usar docker:

```
systemctl enable docker
```

Como último para chequear que se levantaron los contenedores y está funcionando podemos usar el siguiente comando:

```
docker ps
```

Los contenedores que están corriendo actualmente se verán de esta manera:

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS
AMES					
d327602f1434	redis:7.2.5-alpine3.20	"docker-entrypoint.s..."	4 days ago	Up 6 hours	6379/tcp
ayan-redis-1					
8e08cc5632a0	postgres:14.13-alpine3.20	"docker-entrypoint.s..."	4 days ago	Up 6 hours (healthy)	5432/tcp
ayan-postgresql-1					
7957ef85a5e0	rabbitmq:3.13.7-management-alpine	"docker-entrypoint.s..."	4 days ago	Up 6 hours	4369/tcp, 5671-5672/tcp, 15671-15672/tcp, 15691-15692/tcp
ayan-rabbitmq-1					

Luego vamos a realizar la descarga del archivos de configuración “docker-compose.yml”, el cual contiene toda la configuración necesaria para ejecutar la aplicación Mayan EDMS en Docker, y las variables de entorno necesarias para configurar Mayan EDMS correctamente, esto lo descargamos en la carpeta que creamos al principio:

```
curl https://gitlab.com/mayan-edms/mayan-edms/-/raw/master/docker/docker-compose.yml -O
curl https://gitlab.com/mayan-edms/mayan-edms/-/raw/master/docker/.env -O
```

Seguimos con el siguiente comando para ya poder arrancar el contenedor con Mayan EDMS. (por defecto corre en el puerto 80):

```
sudo docker compose up --detach
```

Si el puerto 80 está ocupado con otra cosa va a saltar un error de que ya se está usando ese puerto:

```
Error response from daemon: driver failed programming external connectivity on endpoint mayan-app-1 (8a909da2c1a3c11fb00173d63e4994447cce390c7ce954ea85e1dee8550b5ec0): failed to bind port 0.0.0.0:80/tcp: Error starting userland proxy: listen tcp4 0.0.0.0:80: bind: address already in use
nicole@nicole-IdeaPad-1-15ALC7:~/Proyectos/mayan$
```

Tenemos dos opciones para solucionarlo:

1) Cambiar el puerto en el archivo docker-compose.yml.

Buscamos donde dice ports y donde dice 80, lo cambiamos por el que nos sirva. (el que dice 8000 no debemos modificarlo porque es el puerto por donde está escuchando Mayan EDMS).

```
5 ports:
6   - "${MAYAN_FRONTEND_HTTP_PORT:-80}:8000"
7   # []
8
```

2) Liberar, de lo que habíamos utilizado previamente, el puerto a usar.

En este caso teníamos apache, por ende, con el siguiente comando podremos desocupar el puerto:

```
sudo /etc/init.d/apache2 stop
```

Después de haber elegido otro puerto o liberado el puerto 80, volvemos a correr el comando anterior y nos va a aparecer los contenedores que se levantaron para Mayan.

```
nicole@nicole-IdeaPad-1-15ALC7:~/Proyectos/mayan$ sudo docker compose up --detach
[+] Running 4/4
 ✓ Container mayan-redis-1      Running
 ✓ Container mayan-app-1        Started
 ✓ Container mayan-rabbitmq-1    Running
 ✓ Container mayan-postgresql-1 Running
nicole@nicole-IdeaPad-1-15ALC7:~/Proyectos/mayan$
```

redis y rabbit son cola de procesamientos, postgresql la base de datos y mayan en el programa.

Ejecución

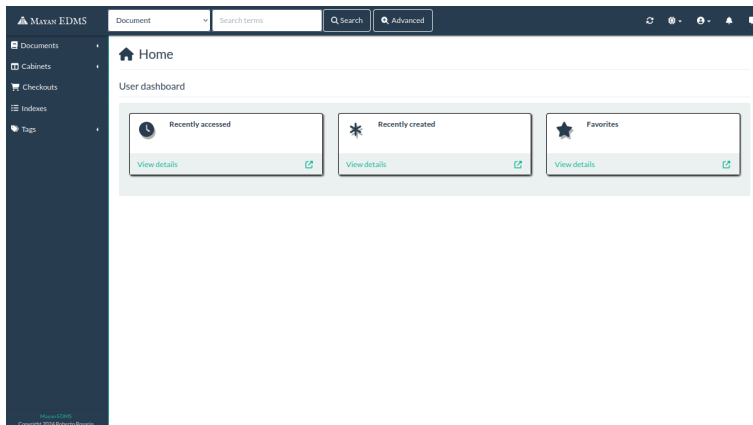
Ya realizada la instalación podemos proceder al navegador web. Escribiremos en la URL “http://localhost” y nos aparecerá el inicio de sesión de Mayan EDMS.



El localhost por defecto estará ubicado en el puerto 80, pero si se configura en otro puerto, deberíamos poner “http://localhost:” y luego de los dos puntos el número de puerto al que queremos acceder.

Al ser la primera vez que iniciamos en Mayan EDMS, nos dan un usuario y contraseña predeterminado el cual vamos a utilizar para entrar. Este usuario y contraseña podremos cambiarlo luego de iniciar sesión.

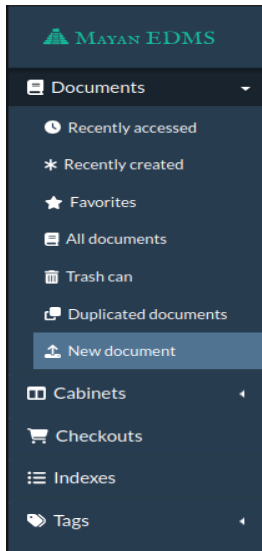
Una vez iniciemos sesión nos aparecerá esta página y Mayan EDMS estará listo para utilizarse:



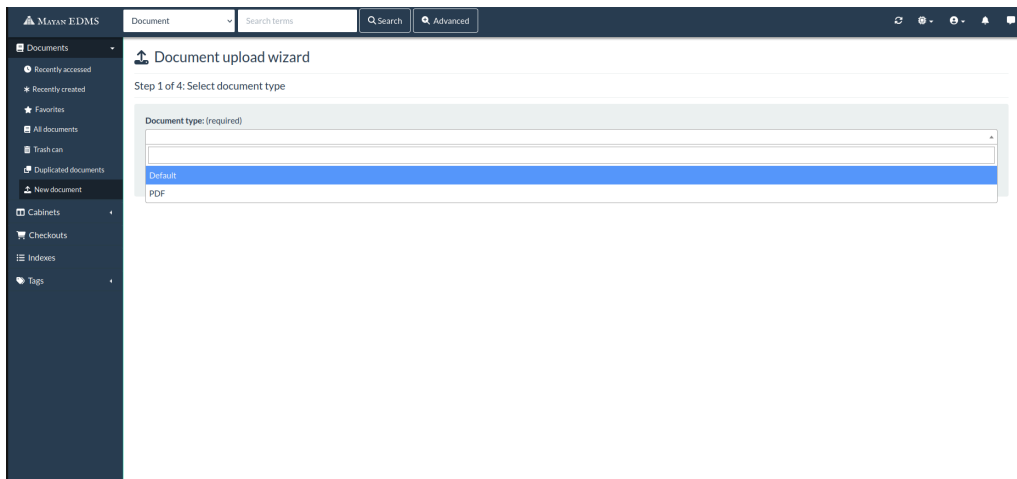
Ejemplo de uso

Desde la misma Página Web:

Nos ubicamos en el apartado New Document

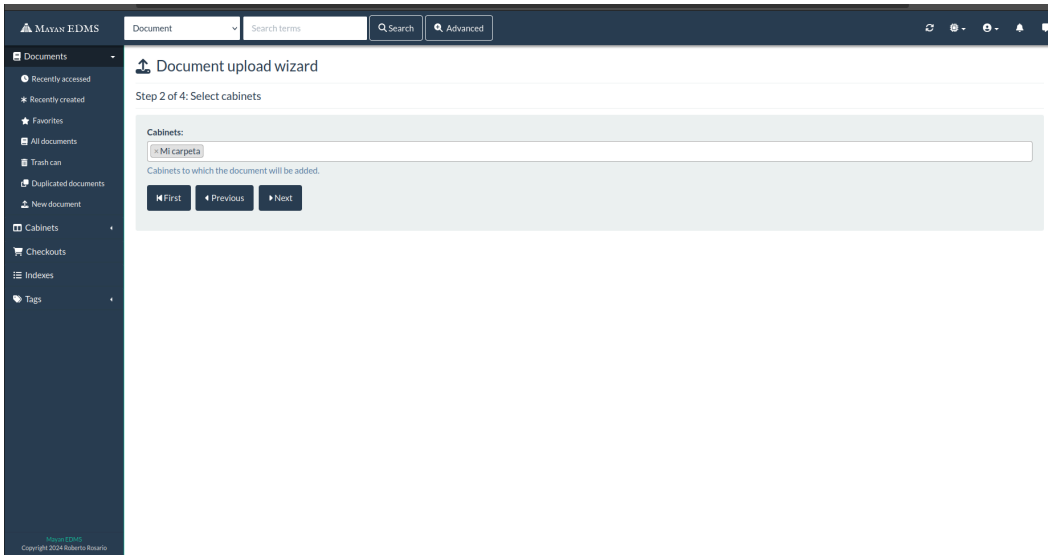


Nos va a pedir que completamos algunos datos del documento, en primera instancia, su tipo.



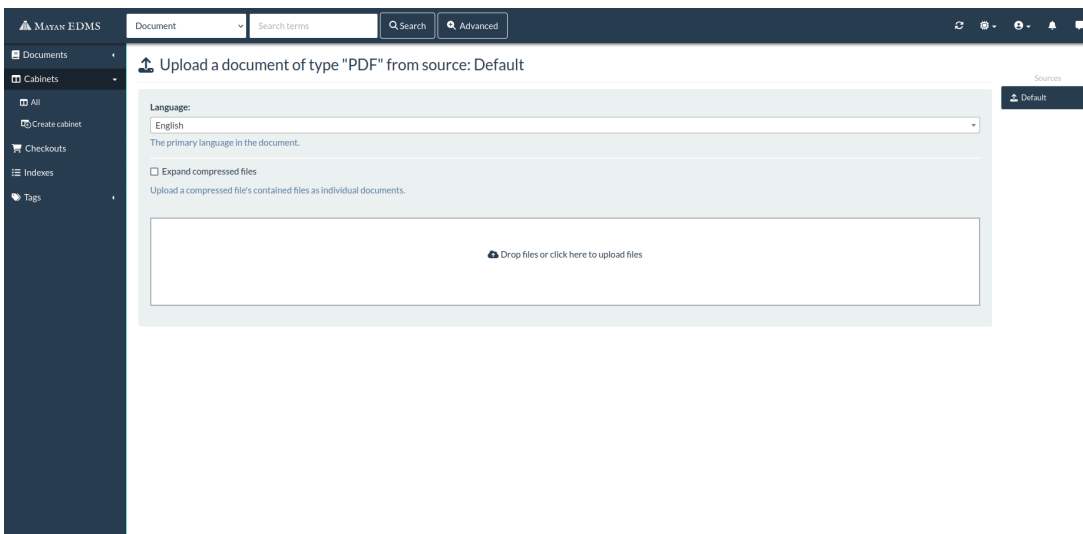
Siempre vamos a tener el Default y, si quisiéramos otro tipo, lo podemos crear, apretando el símbolo que está a la izquierda del símbolo del usuario -> Setup -> Document types -> Actions -> Create Document type.

Luego, nos va a pedir seleccionar en qué carpeta ubicarlo.



Para crear una carpeta vamos al apartado de cabinets a la izquierda y, al apretar, nos va a aparecer Create Cabinet.

Como último paso, podremos elegir el lenguaje y subir el documento.

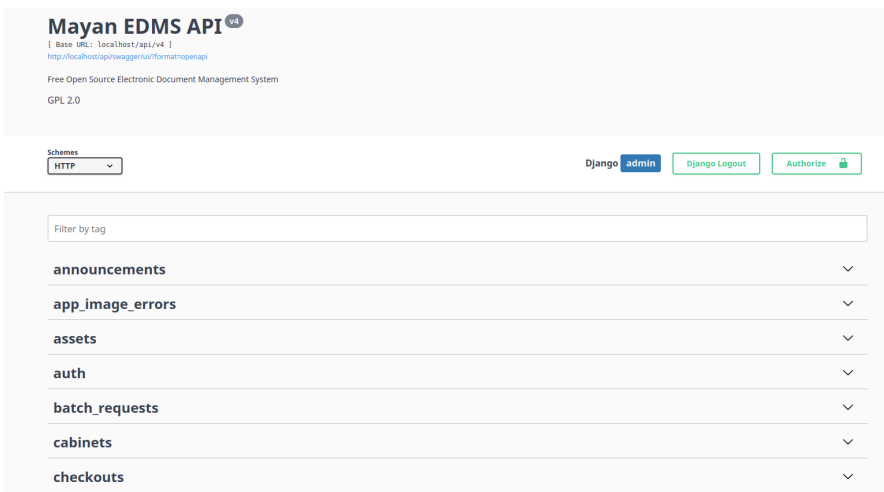


A través de las apis:

Utilizando la URL [“http://localhost/api/”](http://localhost/api/) accederemos a la página la cual contiene las opciones para utilizar las apis.


```
Api Root Filters OPTIONS GET
GET /api/
HTTP 200 OK
Allow: GET, HEAD, OPTIONS
Content-Type: application/json
Vary: Accept
{
  "count": 5,
  "next": null,
  "previous": null,
  "results": [
    {
      "Label": "API version root",
      "url": "http://localhost/api/v4/"
    },
    {
      "Label": "Swagger UI",
      "url": "http://localhost/api/swagger/ui/"
    },
    {
      "Label": "ReDoc UI",
      "url": "http://localhost/api/redoc/ui/"
    },
    {
      "Label": "API schema (JSON)",
      "url": "http://localhost/api/swagger.json"
    },
    {
      "Label": "API schema (YAML)",
      "url": "http://localhost/api/swagger.yaml"
    }
  ]
}
```

Decidimos utilizar Swagger, un conjunto de herramientas de código abierto que permite documentar, crear, definir, y consumir APIs para poder realizar el ejemplo de uso. De todas formas, podría utilizarse cualquiera de las opciones.



Mayan EDMS API ^{v4}
[Base URL: localhost/api/v4]
<http://localhost/api/swaggerui/format-ropenapi>
Free Open Source Electronic Document Management System
GPL 2.0

Schemes: HTTP

Django admin Django Logout Authorize

Filter by tag

- announcements
- app_image_errors
- assets
- auth
- batch_requests
- cabinets
- checkouts

La URL de base es: localhost/api/v4.

Vamos a usar las apis listadas del apartado "Documents":

documents		
GET	/documents/	documents_list
POST	/documents/	documents_create
GET	/documents/accessible/	documents_accessible_list
GET	/documents/created/	documents_created_list
GET	/documents/duplicated/	documents_duplicated_list
GET	/documents/favorites/	documents_favorites_list
POST	/documents/favorites/	documents_favorites_create
GET	/documents/favorites/{favorite_document_id}/	documents_favorites_read
DELETE	/documents/favorites/{favorite_document_id}/	documents_favorites_delete
POST	/documents/upload/	documents_upload_create
GET	/documents/{document_id}/	documents_read
PUT	/documents/{document_id}/	documents_update
PATCH	/documents/{document_id}/	documents_partial_update

Vamos a utilizar POST/documents/upload/ desde Visual Studio Code, por lo que primero vamos a realizar su descarga desde el link: <https://code.visualstudio.com/>

Una vez instalado, tendremos que instalar también Python desde el link: <https://www.python.org/downloads>

Una vez descargado lo anterior, crearemos un archivo en Visual Studio Code terminado en .py (en este caso lo llamaremos mayan, por ende, mayan.py) con las siguientes líneas de código:

```
mayan > mayan.py
1 import requests
2
3 # Configuración
4 url = "http://localhost/api/v4/documents/upload/" # endpoint de la API de Mayan para subir archivos
5 auth = ("admin", "laboratorio2024") # Autenticación básica (usuario y contraseña)
6
7 archivo = "tp01ej03pepe.st"
8
9 # Datos del formulario y archivo
10 data = {
11     "document_type_id": 1, # ID del tipo de documento
12     "name": archivo # Nombre del documento
13 }
14 files = {
15     "file": (archivo, open(archivo, "rb"), "text") # Archivo a cargar
16 }
17
18 # Solicitud POST
19 response = requests.post(url, auth=auth, data=data, files=files)
20
21
```

También subimos el archivo que queremos mandar por POST, el cual debe estar en la misma carpeta que el .py.

Los datos a completarse en auth, data y files, son los propios del usuario y archivo a subir.

Para finalizar, ejecutaremos con el siguiente comando:

`python3 mayan.py`

y ya tendríamos nuestro documento subido en Mayan EDMS.

Conclusión

La realización de este trabajo práctico profundizó el entendimiento de la materia “Laboratorio de Redes y Sistemas Operativos”, logrando profundizar los temas que presenta y, a su vez, ser una motivación para pensar distintas soluciones a los problemas que surgieron al realizar la instalación y configuración de un servicio de red que, posteriormente, lograría ponerse en marcha.

Agradecemos al profesor por darnos la posibilidad de realizar este proyecto y por tomarse el tiempo de realizar la corrección de los trabajos.