

MANUAL DE DESPLIEGUE

Para poner en marcha el ecosistema que se ha diseñado durante este proyecto, haremos uso de las herramientas que implementa el motor de Docker y Docker Compose desplegando, como se ha indicado, cada uno de los servicios en diferentes contenedores agrupados por stacks. Cada uno de esos stacks se define en un archivo docker-compose.yml cuyo despliegue montará los servicios que en él se definen y que serán funcionales por sí mismos.

Redes

Docker maneja las redes de manera perfectamente eficiente por defecto, pero se pueden configurar fácilmente para adaptarlas al despliegue que se vaya a realizar. En este caso, solo necesitamos definir una red a la que llamaremos `matrix_network` en la que estarán todos los servicios que necesiten comunicarse entre sí. De esta manera, la red queda desplegada y funcional y los contenedores se conectan a ella.

Para crear la red `matrix_network` ejecutamos el siguiente comando en una terminal:

```
docker network create --driver=bridge --subnet=172.19.0.0/16 --gateway=172.19.0.1  
matrix_network
```

--driver indica el tipo de red a definir. Admite los parámetros `bridge`, `host` y `none`.

--subnet define la red y máscara que deseemos

--gateway define la puerta de enlace de la red

API Google

Como ya se ha dicho, Synapse permite la autenticación por varios métodos. Uno de ellos es usando la API de google mediante la configuración de [OAuth 2.0](#) siguiendo las instrucciones que proporciona google:

En la primera pantalla se asigna un nombre al proyecto (`matrix`).

En la pantalla de consentimiento se rellena con la siguiente información:

Google Cloud matrix Buscar (/) recursos, documentos, productos y más

APIs y servicios Editar el registro de la app

APIs y servicios habilitados Biblioteca Credenciales **Pantalla de consentimiento d...** Acuerdos de uso de páginas

Dominio de la app

Para protegerlos a ti y a tus usuarios, Google solo permite que las apps que usan OAuth puedan emplear los dominios autorizados. Se mostrará la siguiente información a los usuarios en la pantalla de consentimiento.

Página principal de la aplicación
https://matrix.iesgrancapitan.org
Proporciona a los usuarios un vínculo a tu página principal

Vínculo a la Política de Privacidad de la aplicación
https://policies.google.com/privacy?hl=ES&hl=es
Proporciona a los usuarios un vínculo a tu página pública de Política de Privacidad

Vínculo a las Condiciones del Servicio de la aplicación
https://policies.google.com/terms?hl=ES&hl=es
Proporciona a los usuarios un vínculo a tu página pública de Condiciones del Servicio

Dominios autorizados

Cuando un dominio se usa en la pantalla de consentimiento o en la configuración del cliente de OAuth, debe contar con un registro previo aquí. Si debes verificar la app, ve [Google Search Console](#) para comprobar si tus dominios están autorizados. [Más información](#)

Dominio autorizado 1 *
iesgrancapitan.org

Dominio autorizado 2 *
google.com

[+ AGREGAR UN DOMINIO](#)

Información de contacto del desarrollador

Direcciones de correo electrónico *

Y en la de credenciales:

APIs y servicios ID de cliente para Aplicación web BORRAR

APIs y servicios habilitados Biblioteca Credenciales **Pantalla de consentimiento d...** Acuerdos de uso de páginas

Nombre *
matrix

El nombre de tu cliente de OAuth 2.0. Este nombre solo se usa para identificar al cliente en la consola y no se mostrará a los usuarios finales.

Los dominios de los URI que agregues a continuación se incorporarán automáticamente a tu pantalla de consentimiento de OAuth como dominios autorizados.

Orígenes autorizados de JavaScript

Para usar con solicitudes de un navegador

[+ AGREGAR URI](#)

URI de redireccionamiento autorizados

Para usar con solicitudes de un servidor web

URI 1 *
https://matrix.iesgrancapitan.org/_synapse/client/oidc/callback

[+ AGREGAR URI](#)

Nota: La configuración puede tardar entre 5 minutos y algunas horas en aplicarse

[GUARDAR](#) [CANCELAR](#)

Additional information

ID de cliente 9c
Fecha de creación 2h

Secretos del cliente

Si estás en proceso de cambiar la información

Secreto del cliente G
Fecha de creación 2h
Estado

[+ ADD SECRET](#)

Al finalizar el procedimiento, se genera un client_id y un client_secret que habrá que añadir en el archivo de configuración de Synapse detallado en el siguiente punto.

Una vez definida la red, se copia el contenido del [ANEXO II](#) en la ruta que se define en el docker-compose para el volumen. Se trata del archivo de configuración de Synapse homeserver.yml. El él se definirán las opciones que se deseen para el servidor. Para este proyecto se editan del original:

- El bloque “[modules](#)” para definir un módulo customizado que permitiese la implementación de LDAP.
- El bloque “[server](#)”. Añadir el [nombre del servidor](#) y la [URL pública](#) que usarán los clientes.
- Aun dentro del bloque “[server](#)”, comentar la [línea 344](#) y [añadir](#) “openid” como recurso que nos hará falta para la autenticación con google.
- En el bloque “[database](#)”, descomentar todo lo referente a la configuración de Postgres y modificarlo con los [parámetros de conexión a la base de datos](#). Comentar la [conexión por defecto](#) de SQLite.
- En el bloque “[login](#)” se indica la ruta donde se [almacenará el archivo de configuración](#) de login.
- En el bloque “[media store](#)” se indica la ruta donde se almacenarán los archivos que se compartan en la [línea 999](#). Y la [1697](#) se descomenta.
- Dentro del mismo bloque, en la línea [2051](#) se define el oidc para la autenticación con google para el que necesitamos el [client_id](#) y el [client_secret](#) que se proporcionan cuando se da de alta en la API de google.

Para tener un cliente web personalizado, se [descarga y descomprime](#) en la ubicación definida en el volumen la aplicación web como se indicó en el despliegue detallado. Se copia el config.sample.json a config.json y se modifica:

- En la sección “default_server_config” el base_url con la url del servidor matrix.iesgrancapitan.org, el nombre del servidor: matrix.iesgrancapitan.org y de nuevo la base_url de “identity_server”
- Como se tendrá un jitsi propio, casi al final del archivo se establece como “preferred_domain” el dominio del servidor de jitsi: jitsi.iesgrancapitan.org

Ahora podemos desplegar el docker-compose.yml de este stack que está definido en el [ANEXO III](#).

```
docker compose up -d
```

Una vez desplegado el contenedor, se puede manejar la base de datos con el SGBD pgAdmin4 introduciendo en un navegador la URL *localhost:5050*. La primera vez que se acceda, habrá que establecer la conexión con el servidor de la base de datos.

- En la pestaña general, se define un nombre para el servidor

The screenshot shows the 'matrix' application window with the 'General' tab selected. The window has a title bar with the 'matrix' logo and standard window controls. The tabs are 'General', 'Connection', 'Parameters', 'SSH Tunnel', and 'Advanced'. The 'General' tab contains the following fields:

- Name: A text input field containing 'matrix'.
- Server group: A dropdown menu showing 'Servers'.
- Background: A button with an 'X' icon.
- Foreground: A button with an 'X' icon.
- Shared?: A toggle switch that is currently turned off.
- Comments: A large text area.

At the bottom of the window, there are three buttons: 'Close' (with an 'X' icon), 'Reset' (with a circular arrow icon), and 'Save' (with a floppy disk icon). There are also information and help icons on the left.

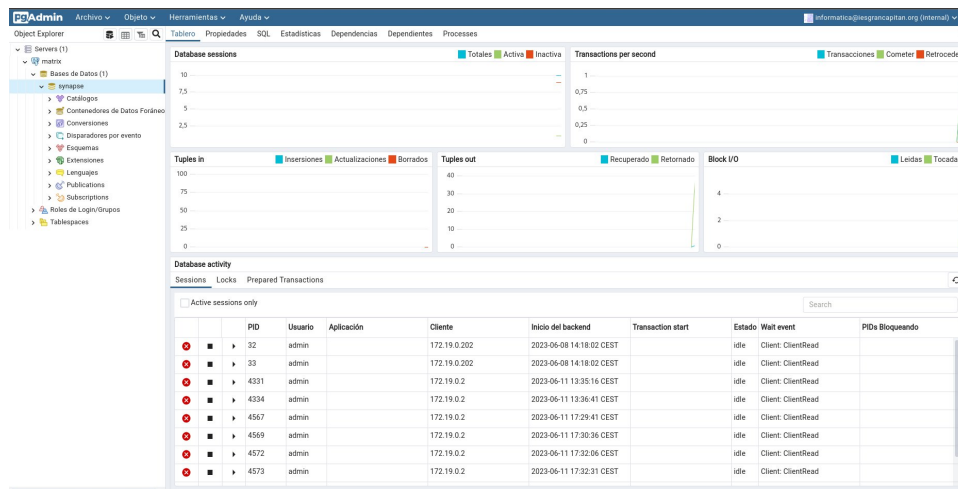
- En la pestaña conexión, se definen los parámetros de conexión con el servidor que se especifican en el docker-compose

The screenshot shows the 'matrix' application window with the 'Connection' tab selected. The window has the same title bar and tabs as the previous screenshot. The 'Connection' tab contains the following fields:

- Host name/address: A text input field containing 'db'.
- Port: A text input field containing '5432'.
- Maintenance database: A text input field containing 'synapse'.
- Username: A text input field containing 'admin'.
- Kerberos authentication?: A toggle switch that is currently turned off.
- Role: A text input field.
- Service: A text input field.

At the bottom of the window, there are three buttons: 'Close' (with an 'X' icon), 'Reset' (with a circular arrow icon), and 'Save' (with a floppy disk icon). There are also information and help icons on the left.

- Cuando los cambios se hayan guardado, aparecen las métricas de uso de la base de datos y el resto de herramientas para la administración de la base de datos



Stack LAM

Para el despliegue de este stack, lo hacemos desde el [ANEXO IV](#). Una vez desplegado,

```
docker compose up -d
```

se introduce en el navegador la dirección `localhost:8083` para acceder a LDAP Account Manager. La primera vez que se accede, se debe configurar las opciones del servidor.

- En los ajustes generales, introducimos la dirección en la que se encuentra el servidor. Se puede usar la IP, pero docker resuelve el nombre del contenedor a través de su DNS interno:

Preferencias del servidor

Dirección del servidor ^{*} [?]

Activar TLS [?]

Límite de búsqueda LDAP [?]

Parte del DN a ocultar [?]

Opciones Avanzadas

Nombre a mostrar [?]

Seguir referrals ☐ [?]

Resultados paginados ☐ [?]

Overlay de integridad referencial ☐ [?]

Ocultar la solicitud de contraseña para las contraseñas expiradas ☐ [?]

- Se configura el idioma

A Configuración del idioma

Idioma por defecto	Español (España) ?
Zona horaria	Europe/Madrid ?

- En los ajustes de herramientas, se define el visor del árbol con los DC de la organización

Ajustes de herramientas

Herramientas ocultas

Explorador de esquemas	<input type="checkbox"/>	Importación/exportación LDAP	<input type="checkbox"/>	Dispositivos WebAuthn	<input type="checkbox"/>
Edición múltiple	<input type="checkbox"/>	Editor de OU	<input type="checkbox"/>	Comprobar	<input type="checkbox"/>
Información del servidor	<input type="checkbox"/>	Editor de PDF	<input type="checkbox"/>	Enviar archivos	<input type="checkbox"/>
Editor de perfiles	<input type="checkbox"/>	Visor del árbol	<input type="checkbox"/>		

Visor del árbol

Sufijo del árbol ?

- Y en las preferencias de seguridad, se añade el usuario administrador a la lista de usuarios válidos

Preferencias de seguridad

Método del inicio de sesión ?

Lista de usuarios válidos *

 ?

- En la pestaña “Tipos de cuentas” se definen los sufijos LDAP para usuarios y grupos

Tipos de cuentas activos

Usuarios

Cuentas de usuario (p.ej. UNIX,Samba y Kolab) ↓ ✕

Sufijo LDAP * ?

Atributos del listado ?

Etiqueta personalizada ?

Filtro LDAP adicional ?

Oculto ☐ ?

Grupos

Cuentas del grupo (p.ej. Unix y Samba) ↑ ✕

Sufijo LDAP * ?

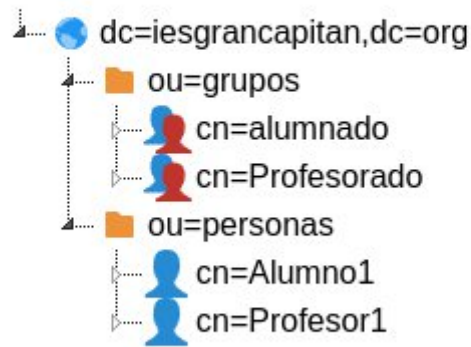
Atributos del listado ?

Etiqueta personalizada ?

Filtro LDAP adicional ?

Oculto ☐ ?

Se guardan los cambios y ya se puede acceder al servidor LDAP que muestra el árbol de ejemplo creado:



Stack Jitsi

Para desplegar un servicio Jitsi propio, se descarga **(no clonar)** el [repositorio](#) de github.

```
wget https://github.com/jitsi/docker-jitsi-meet/archive/refs/tags/stable-8719.zip
```

Una vez extraído, se copia el archivo env.example a .env ([ANEXO V](#)). Se ejecuta el script

```
./gen-passwords.sh
```

para establecer contraseñas fuertes en el archivo de variables de entorno que se acaba de crear.

Creamos los directorios que serán necesarios:

```
mkdir -p ~/.jitsi-meet-cfg/{web,transcripts,prosody/config,prosody/prosody-plugins-custom,jicofo,jvb,jigasi,jibri}
```

Ya se podría desplegar el contenedor y tendríamos un jitsi funcional.

```
docker compose up -d
```

Vamos a personalizarlo. Para ello, necesitamos modificar el archivo main-es.json que mapearemos como un volumen en el docker compose.

```
docker cp a175229d25e3:/usr/share/jitsi-meet/lang/main-es.json
```

Modificamos las líneas que deseemos editar. En este caso la 1145 y desplegamos el contenedor del [ANEXO VI](#).

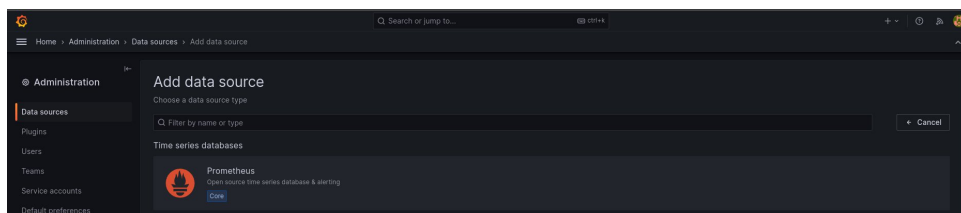
Stack Portainer

Este stack contiene un solo contenedor. Una vez se despliegue desde el [ANEXO VII](#), navegando hasta la URL `localhost:9443` y aparece un asistente para la configuración inicial del environment. Seleccionamos Docker Standalone y conexión por socket. Se crea un environment por defecto llamado local y ya se pueden ver todos los contenedores, stacks y redes que hay desplegados.

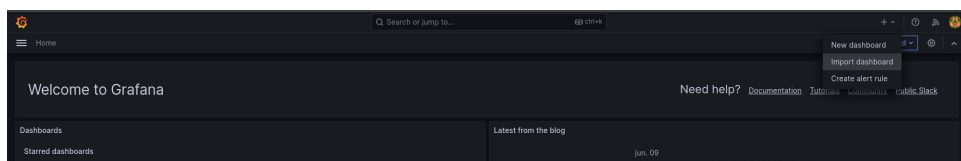
Stack Grafana

Para desplegar el servicio de Grafana, se necesitan dos archivos: el `docker-compose.yml` ([ANEXO VIII](#)) y el `prometheus.yml` ([ANEXO IX](#)). El primero, como cualquier compose, despliega los contenedores. El segundo archivo define los “jobs” que Prometheus va a monitorizar.

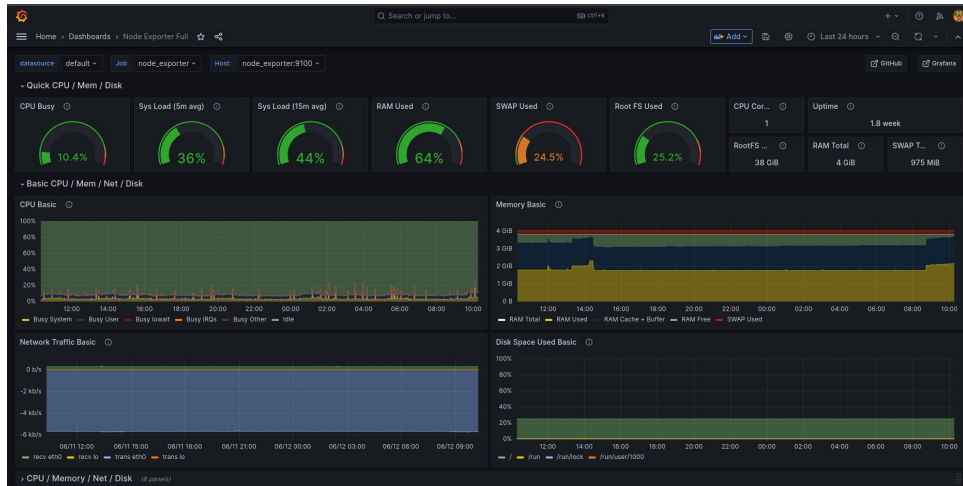
Después de desplegar el stack, se navega hasta la URL `localhost:3000` donde se podrán configurar los dashboards para la visualización de las métricas. Primero, configuramos el origen de los datos, en este caso, prometheus



Y ya se puede importar los dashboards de la fuente descrita en el diseño detallado, usando los códigos del elegido.



Cuando se carga el diseño que se ha decidido, se elige como fuente de datos prometheus del desplegable. Al cargar, aparecen las métricas.



Stack Heimdall

Con el fin de facilitar el uso y la navegación por las distintas aplicaciones web que se han implementado a lo largo de todo el despliegue, se añade un último contenedor desde el [ANEXO X](#). Para su configuración, se navega hasta *localhost:8091* y se empiezan a añadir cada una de las URLs que hemos estado utilizando en los stacks anteriores siguiendo el asistente.

A screenshot of the Grafana application configuration interface, titled 'Preview'. It shows a form for adding a new application. The 'Application Type' is set to 'Grafana' and the 'Website' is 'https://localhost:3000'. The 'Add application' section contains fields for 'Application name' (Grafana), 'Colour' (#161b1f), and 'URL' (https://localhost:3000). There is also a 'Tags (Optional)' field with the value 'Home dashboard' and an 'Upload an icon' button. The 'Description' field contains the text 'Data visualization & Monitoring with support for Graphite, InfluxDB, Prometheus, Elasticsearch and many more databases.' The interface has a dark theme and a sidebar on the right with navigation icons.