Jellyfin es una plataforma de medios de comunicación de código abierto que permite a los usuarios acceder a sus colecciones de películas, música y programas de televisión en cualquier dispositivo compatible con Jellyfin. Puede funcionar en cualquier sistema operativo, incluyendo Windows, macOS, Linux, Android, iOS y muchos más. Pero en este caso, vamos a hablar de cómo he utilizado Jellyfin con Docker y Docker-Compose para sincronizar todas las subidas en Transmission.

En primer lugar, Docker es una plataforma de contenedores que permite a los desarrolladores crear, ejecutar y compartir aplicaciones en cualquier lugar. Con Docker, podemos empaquetar todas las dependencias de una aplicación en un contenedor y luego ejecutar ese contenedor en cualquier sistema que tenga Docker instalado. Esto significa que podemos garantizar que nuestra aplicación se ejecute exactamente de la misma manera en cualquier lugar, independientemente del sistema operativo, la configuración o cualquier otra variable.

Jellyfin, por otro lado, es un servidor de medios de comunicación que nos permite transmitir nuestra música, películas y programas de televisión a cualquier dispositivo conectado a nuestra red local o remota. Al utilizar Jellyfin con Docker, podemos crear un contenedor que incluye todas las dependencias necesarias para ejecutar Jellyfin y luego ejecutar ese contenedor en cualquier sistema que tenga Docker instalado. De esta manera, podemos garantizar que nuestro servidor de medios se ejecute exactamente de la misma manera en cualquier lugar.

En segundo lugar, Docker-Compose es una herramienta que nos permite definir y ejecutar aplicaciones Docker de varias piezas. Podemos definir todas las dependencias de una aplicación en un archivo de configuración de Docker-Compose y luego ejecutar ese archivo para crear y ejecutar todos los contenedores necesarios para nuestra aplicación. Con Docker-Compose, podemos definir y gestionar todas las dependencias de nuestra aplicación en un solo lugar, lo que hace que sea mucho más fácil de mantener y actualizar.

En mi caso, he utilizado un archivo de configuración de Docker-Compose para definir dos contenedores: uno para Transmission y otro para Jellyfin. El contenedor de Transmission se encarga de descargar los torrents y almacenarlos en una carpeta compartida, mientras que el contenedor de Jellyfin se encarga de indexar y servir los archivos multimedia.

Además de la gestión centralizada de los contenedores, otra ventaja de utilizar Docker-Compose es la facilidad para escalar el sistema. Si necesitas aumentar la capacidad de Jellyfin, simplemente puedes agregar más contenedores utilizando el mismo archivo de configuración y Docker-Compose se encargará de coordinar todos los nuevos contenedores.

Este archivo docker-compose contiene la configuración para levantar dos contenedores: uno para Transmission (un cliente BitTorrent) y otro para Jellyfin (un servidor de medios).

**DOCKER-COMPOSE**

Este archivo docker-compose contiene la configuración para levantar dos contenedores: uno para Transmission (un cliente BitTorrent) y otro para Jellyfin (un servidor de medios).

En cuanto a la configuración de Transmission, se especifica que el contenedor se llamará "transmission", se usará la imagen "linuxserver/transmission" y siempre se reiniciará en caso de fallo.

También se especifica que el contenedor estará en la red "jellyfin" y se montarán dos volúmenes en el host: uno para los archivos descargados y otro para la configuración de Transmission. Además, se establecen las variables de entorno PUID, PGID y TZ para el usuario y grupo del contenedor y la zona horaria.

En cuanto a la configuración de Jellyfin, se especifica que el contenedor se llamará "jellyfin", se usará la imagen "jellyfin/jellyfin" y siempre se reiniciará en caso de fallo. También se especifica que el contenedor estará en la red "jellyfin" y se montarán varios volúmenes en el host para la configuración de Jellyfin, transcodificación de archivos, datos y películas. Además, se establece la variable de entorno TZ para la zona horaria.

Finalmente, se define una red llamada "jellyfin" que será utilizada por ambos contenedores y se especifica que su driver será "bridge".

Además, se exponen los puertos 9091, 51413 (TCP y UDP) para Transmission y los puertos 8096 y 8920 para Jellyfin en el host, lo que permitirá el acceso a las interfaces web de ambos servicios desde fuera del contenedor.

En resumen, Jellyfin es una excelente solución para la gestión de medios de comunicación en un entorno empresarial, y la utilización de Docker-Compose simplifica enormemente la gestión y escalabilidad del sistema. Con la ayuda de estas herramientas, he sido capaz de implementar una solución de gestión de medios de comunicación en mi lugar de trabajo que es altamente eficiente y fácil de mantener.