

## Docker container guide with Jellyfin media server v1

A current (2020-09) 10 step guide which step by step makes you run the mediaserver Jellyfin <https://jellyfin.org/> or <https://github.com/jellyfin> in a Docker container.

You can run any application in Docker as long as it can be installed and executed unattended, and the base operating system supports the app. Windows Server Core runs in Docker which means you can run pretty much any server or console application in Docker.

This guide will first help you install needed software, a pre-built container, configure it and get it running. Then in step two (part 9 → ) download a simple tutorial app and containerize it. Lastly download the advanced Jellyfin source code and create your own Jellyfin container from it.

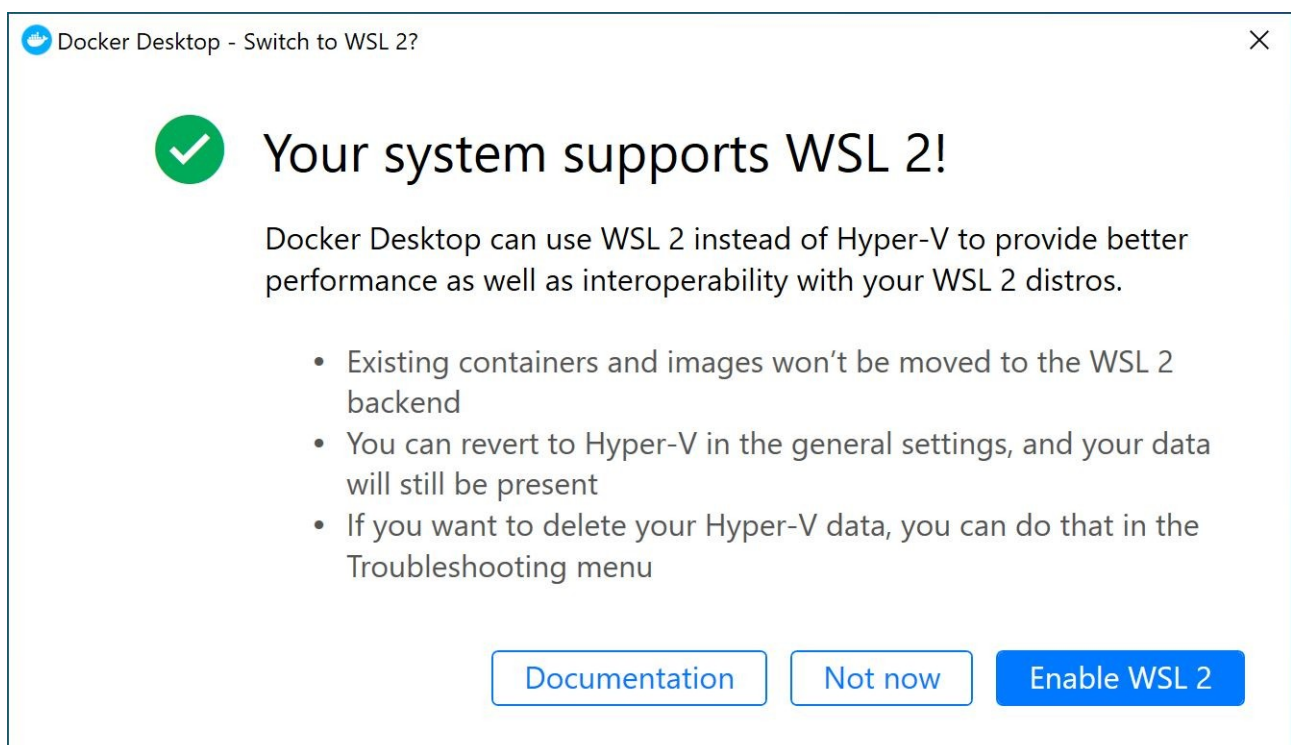
First you may want to read a little background overview...

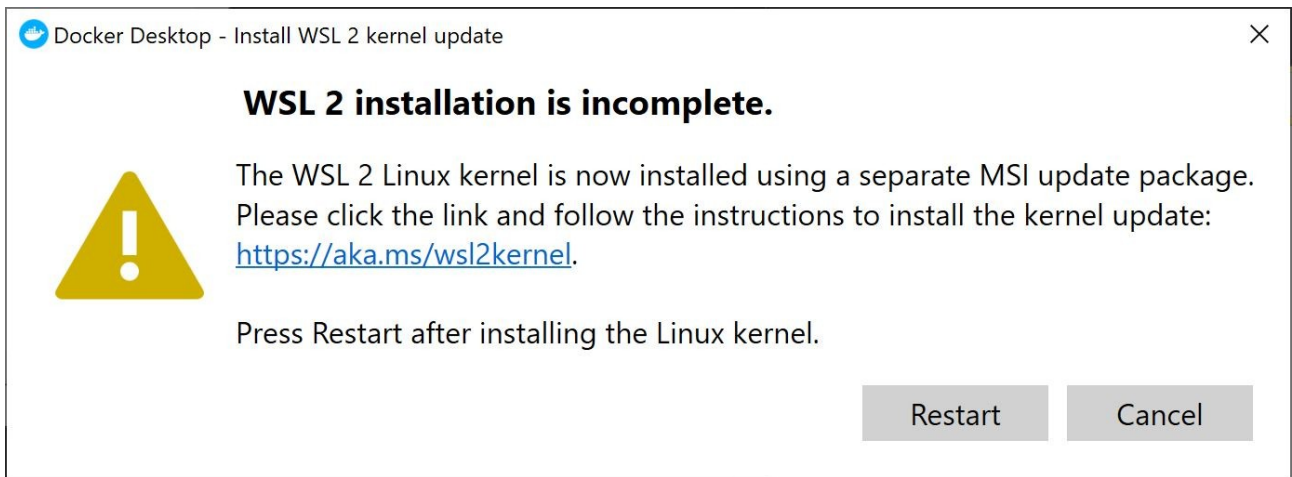
- <https://www.docker.com/resources/what-container>
- <https://docs.microsoft.com/en-us/virtualization/windowscontainers/>

1. Read Jellyfin on Docker for Dummies to get going quick.

- [https://www.reddit.com/r/jellyfin/comments/ercwn/jellyfin\\_on\\_docker\\_for\\_dummies/](https://www.reddit.com/r/jellyfin/comments/ercwn/jellyfin_on_docker_for_dummies/)

- Install Docker Desktop etc.





2. Read Docker ♥ WSL 2 – The Future of Docker Desktop for Windows. Install Windows Subsystem for Linux.

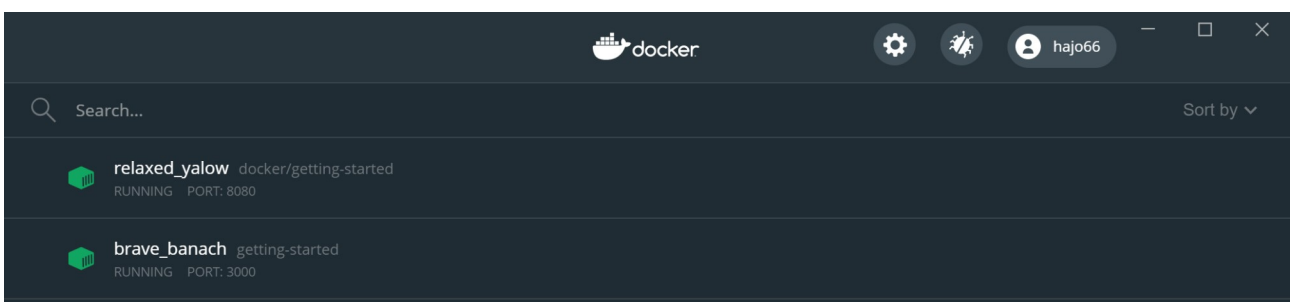
- <https://www.docker.com/blog/docker-hearts-wsl-2/>
- <https://docs.docker.com/docker-for-windows/wsl/>
- <https://docs.microsoft.com/en-us/windows/wsl/install-win10>
- <https://docs.microsoft.com/en-us/windows/wsl/wsl2-kernel>
- [https://www.reddit.com/r/docker/comments/euigvw/wsl2\\_docker\\_image\\_location/](https://www.reddit.com/r/docker/comments/euigvw/wsl2_docker_image_location/)

3. When Docker Desktop is installed try out the Getting Started guide/tutorial container with the command below (8080 if port 80 is taken already in your machine). The getting started guide contains info for step 9 in this guide as well so perform this step.

```
docker run -d -p 8080:80 docker/getting-started
```

4. Get familiar with Docker on YouTube.

- <https://www.youtube.com/playlist?list=PLVx1govxj-amqyqHceAhkcsopzi4PFcKc>
- Start the Docker Desktop Dashboard to get a UI for managing your containers. Docker Desktop > Dashboard. The image shows two running containers (getting started in step 3 and app tutorial in step 9) which you can drill down into to view settings, log output etc.



5. Read the Docker release notes especially about WSL2 mounts.

- <https://docs.docker.com/docker-for-windows/release-notes/>

Docker Desktop now prompts users to share directories for paths like

\\\\c/Users/foo, rather than only paths like C:\\Users\\foo and C:/Users/foo.

6. A valid **docker-compose.yml** file example for a single official Jellyfin container. Ensure personal firewall ports are open on the Docker host machine for LAN connections and port forwarding is enabled in NAT-router if remote WAN connections are used.

To start the container issue the command `docker-compose up -d` or `docker-compose up` in the same folder as the `.yml` file is located.

```
#####
# Jellyfin
# https://hub.docker.com/r/jellyfin/jellyfin
# https://www.reddit.com/r/jellyfin/comments/fqojrn/install_via_dockercompose_how_to_define_multiple/
#####
# https://www.reddit.com/r/jellyfin/comments/erccwn/jellyfin_on_docker_for_dummies/
# docker-compose up -d
# docker-compose stop
# docker-compose start
# https://hub.docker.com/r/linuxserver/jellyfin
# WSL2
# https://nickjanetakis.com/blog/setting-up-docker-for-windows-and-wsl-to-work-flawlessly
#####
version: '3.7'
services:
  jellyfin:
    image: 'jellyfin/jellyfin:latest'
    container_name: jellyfin
    environment:
      - PUID=1000
      - PGID=1000
      - TZ=Europe/Stockholm
    volumes:
      - ///c/vm/jellyfin/config:/config
      - ///c/vm/jellyfin/cache:/cache
      - ///z/TV Shows:/media/tvshows:ro
```

```
- ///z/Movies:/media/movies:ro
- ///c/bittorrent/Les visiteurs. Collection (1993-2016):/media/test:ro
```

```
#network_mode: "host"
```

```
networks:
```

```
- jellyfin_default
```

```
ports:
```

```
# If Jellyfin were using the standard 8096 inside the container but you wanted your
```

```
# clients to connect on 8181 the parameter would be -p 8181:8096
```

```
- '8096:8096'
```

```
- '8920:8920' #optional https
```

```
networks:
```

```
jellyfin_default:
```

```
external: true
```

7. Get familiar with docker-compose (only some options / commands shown) which use the above **docker-compose.yml** file.

```
PS C:\vm\docker Desktop> docker-compose.exe
```

Define and run multi-container applications with Docker.

Usage:

```
docker-compose [-f <arg>...] [options] [COMMAND] [ARGS...]
```

```
docker-compose -h|--help
```

Options:

```
-f, --file FILE          Specify an alternate compose file
                          (default: docker-compose.yml)
```

```
...
```

Commands:

```
...
```

```
ps                      List containers
```

```
start                   Start services
```

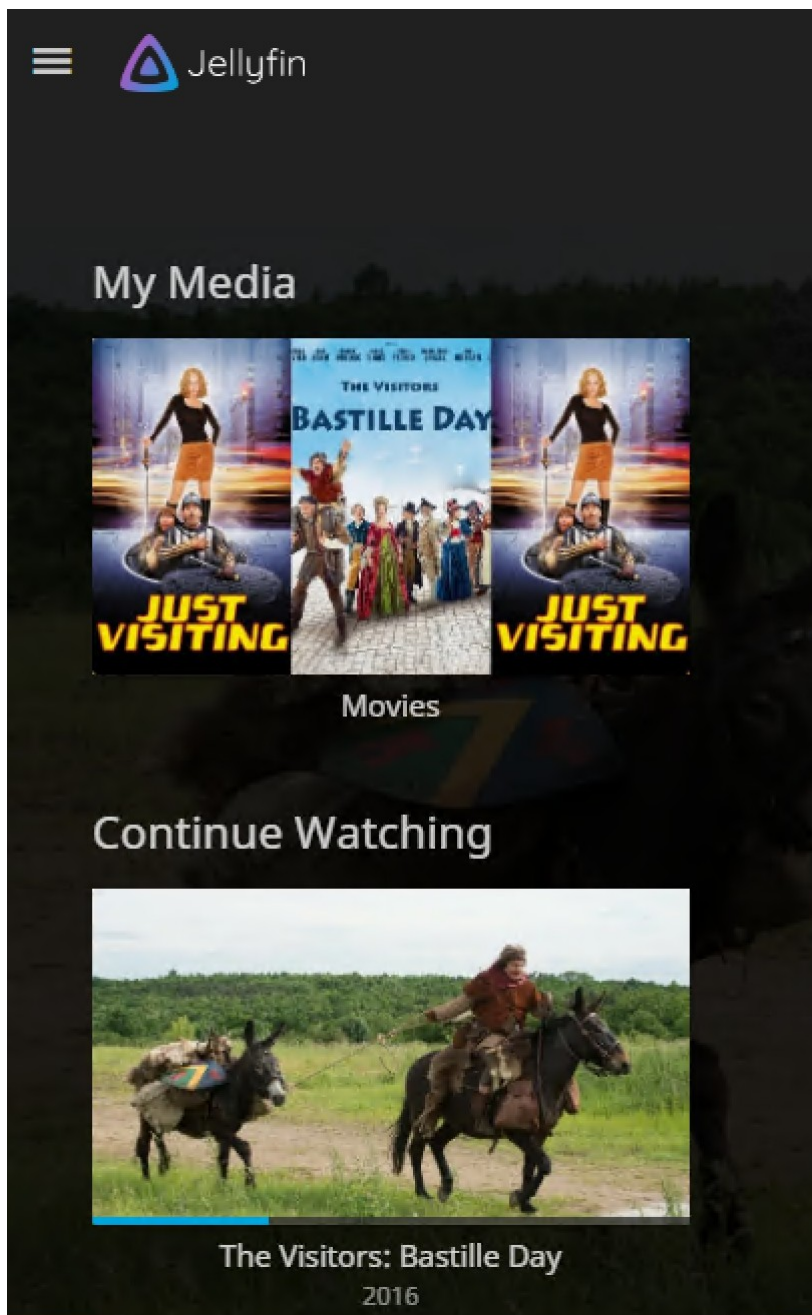
```
stop                    Stop services
```

```
up                      Create and start containers
```

```
...
```

Try for example: `docker-compose help up`

8. Set up Jellyfin so one can browse and play media. This is a **VERY** limited screen dump!



9. Perform the "Our Application" tutorial in the Getting Started guide. Ie. continue from point 3.  
#docker\>docker run -d -p 8080:80 docker/getting-started

Download and extract the app zip. Create the Dockerfile and study its contents.

#docker\app> docker build -t getting-started .

#docker\>docker run -dp 3000:3000 getting-started

10. Download the Jellyfin source code and create your own Jellyfin container with the instructions at: <https://jellyfin.org/docs/general/administration/building.html>

Issue the commands and run the application with Docker Desktop.

For me the jellyfin-web component may not have installed correctly even if the config seems to be given correct (line 3 – line 9) in the Dockerfile file. I got a page not found (HTTP ERROR 404) message using only <http://localhost:8096/>.

However using the full URL: <http://localhost:8096/web/index.html> the Jellyfin login page loads correct. But it is not possible to login/create an initial user so an error is still present.

Container log output:

```
jellyfin | [19:53:51] [ERR] [5] Jellyfin.Server.Implementations.Users.UserManager:
Error authenticating with provider Default
jellyfin | MediaBrowser.Controller.Authentication.AuthenticationException: Specified
user does not exist.
jellyfin | at
Jellyfin.Server.Implementations.Users.DefaultAuthenticationProvider.Authenticate(String
username, String password, User resolvedUser)
jellyfin | at
Jellyfin.Server.Implementations.Users.UserManager.AuthenticateWithProvider(IAuthenticat
ionProvider provider, String username, String password, User resolvedUser)
jellyfin | [19:53:51] [ERR] [5] Jellyfin.Server.Implementations.Users.UserManager:
Error authenticating with provider InvalidOrMissingAuthenticationProvider
jellyfin | MediaBrowser.Controller.Authentication.AuthenticationException: User
Account cannot login with this provider. The Normal provider for this user cannot be
found
jellyfin | at
Jellyfin.Server.Implementations.Users.InvalidAuthProvider.Authenticate(String username,
String password)
jellyfin | at
Jellyfin.Server.Implementations.Users.UserManager.AuthenticateWithProvider(IAuthenticat
ionProvider provider, String username, String password, User resolvedUser)
jellyfin | [19:53:51] [INF] [5] Jellyfin.Server.Implementations.Users.UserManager:
Authentication request for hans has been denied (IP: ::ffff:172.18.0.1).
jellyfin | [19:53:51] [ERR] [5] Jellyfin.Server.Middleware.ExceptionMiddleware:
Error processing request: Invalid username or password entered. URL POST
/Users/authenticatebyname.
```

Read about server development at: <https://github.com/jellyfin/jellyfin> when and if you want to compile the source (with Visual Studio etc.) to managed execution (non-virtual/containerized) code.

### Overview and go further with Docker containers

- To read more and understand the most basic important stuff: <https://docs.linuxserver.io/>
- Repository of official container images: <https://hub.docker.com/>
- How to Dockerize Windows Applications: The 5 Steps: <https://blog.sixeyed.com/how-to-dockerize-windows-applications>