Docker Compose Lab

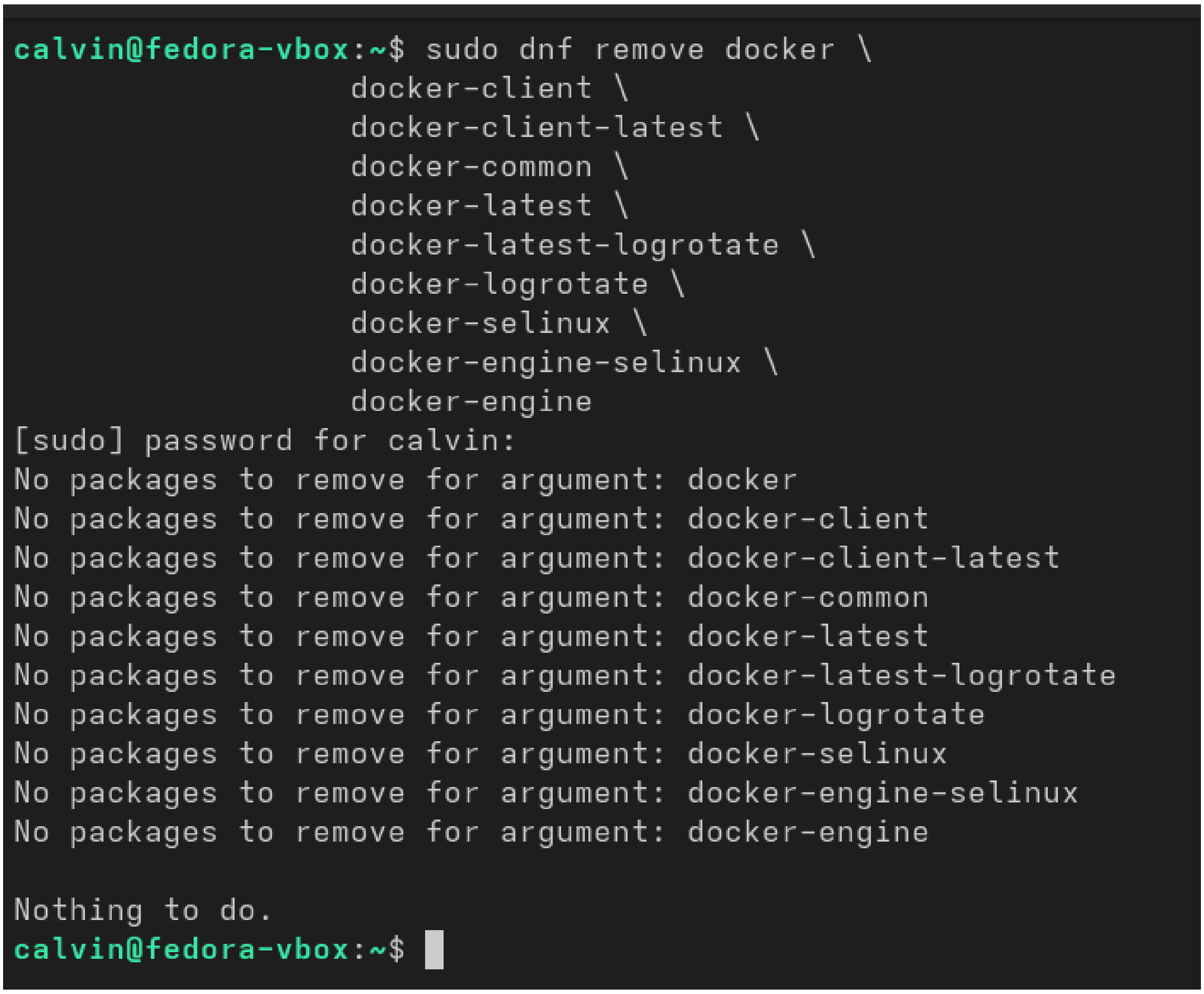
Calvin Wasilevich
2025.11.16

Installing Docker

I opted to use my existing fedora virtual machine to install docker. I used the instructions provided by docker to install the engine onto fedora systems. These instructions will be summarized here.

1. Remove any previous docker installations that may have come with the distribution. These are often out-of-date, and it is best to fetch the official docker application from the docker website. The possible other commands are removed using the command:

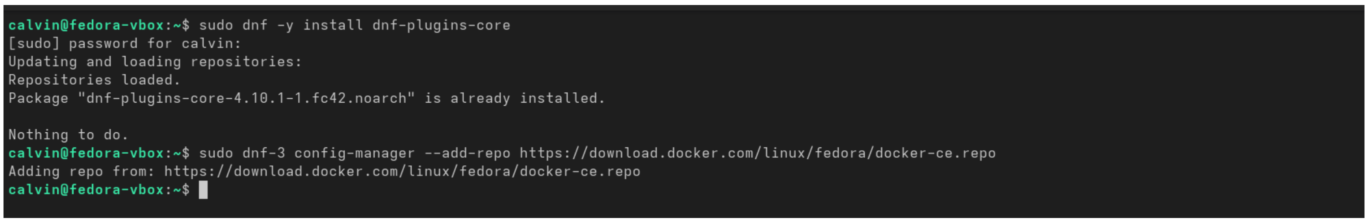
```
sudo dnf remove docker \
                  docker-client \
                  docker-client-latest \
                  docker-common \
                  docker-latest \
                  docker-latest-logrotate \
                  docker-logrotate \
                  docker-selinux \
                  docker-engine-selinux \
                  docker-engine
```



```
calvin@fedora-vbox:~$ sudo dnf remove docker \
                  docker-client \
                  docker-client-latest \
                  docker-common \
                  docker-latest \
                  docker-latest-logrotate \
                  docker-logrotate \
                  docker-selinux \
                  docker-engine-selinux \
                  docker-engine
[sudo] password for calvin:
No packages to remove for argument: docker
No packages to remove for argument: docker-client
No packages to remove for argument: docker-client-latest
No packages to remove for argument: docker-common
No packages to remove for argument: docker-latest
No packages to remove for argument: docker-latest-logrotate
No packages to remove for argument: docker-logrotate
No packages to remove for argument: docker-selinux
No packages to remove for argument: docker-engine-selinux
No packages to remove for argument: docker-engine

Nothing to do.
calvin@fedora-vbox:~$
```

2. The docker rpm repository holds the official docker installation files. It must be added to dnf in order for dnf to look to the docker repository to download and update packages. Add the docker rpm repository with the commands:

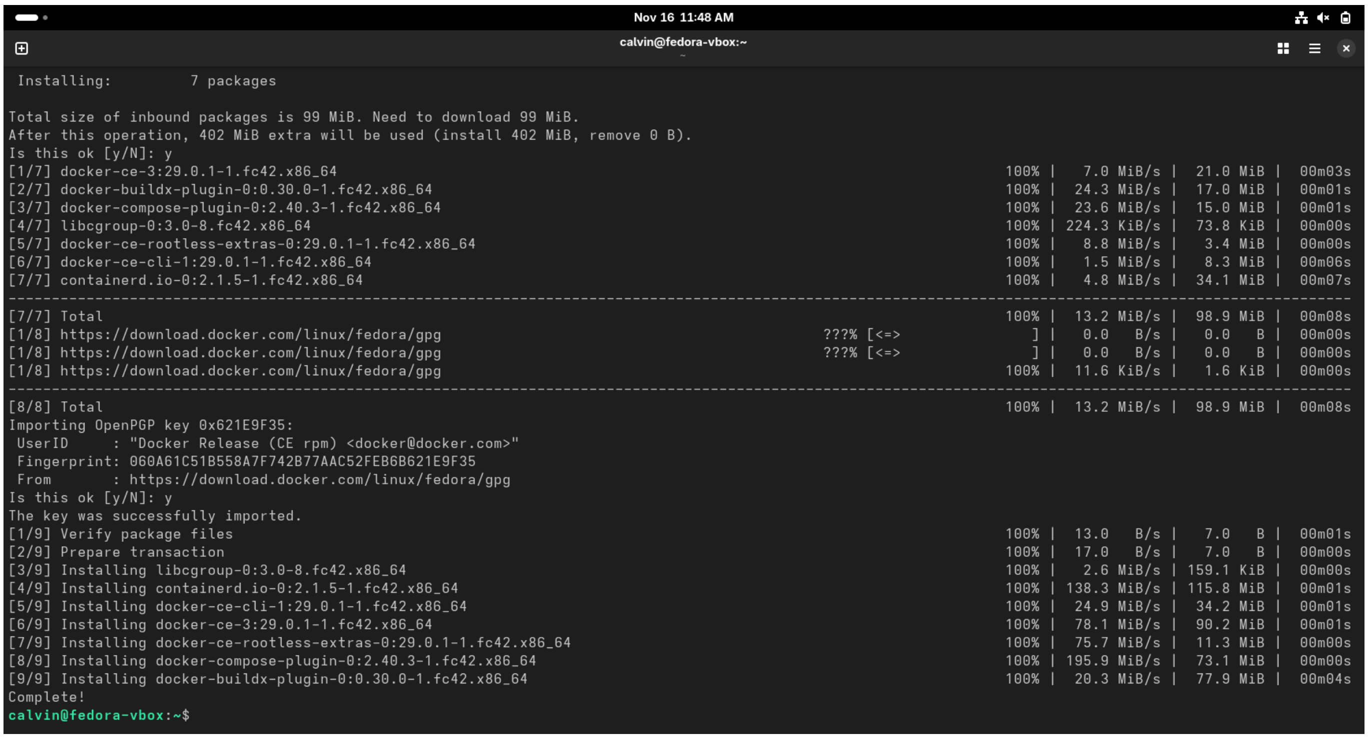
```
sudo dnf -y install dnf-plugins-core
sudo dnf-3 config-manager --add-repo
https://download.docker.com/linux/fedora/docker-ce.repo
```



```
calvin@fedora-vbox:~$ sudo dnf -y install dnf-plugins-core
[sudo] password for calvin:
Updating and loading repositories:
Repositories loaded.
Package "dnf-plugins-core-4.10.1-1.fc42.noarch" is already installed.

Nothing to do.
calvin@fedora-vbox:~$ sudo dnf-3 config-manager --add-repo https://download.docker.com/linux/fedora/docker-ce.repo
Adding repo from: https://download.docker.com/linux/fedora/docker-ce.repo
calvin@fedora-vbox:~$
```

3. Install docker (and supporting packages) from the new repo with:

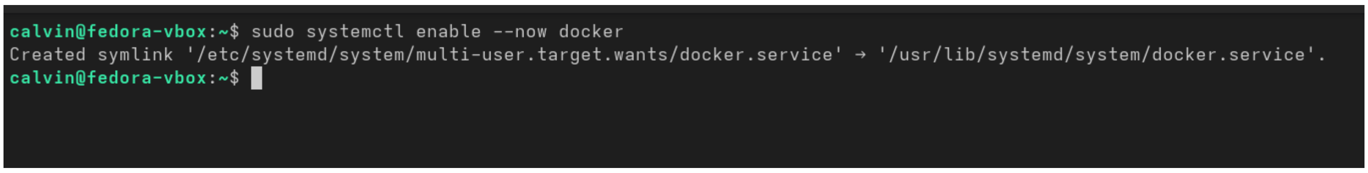
```
sudo dnf install docker-ce docker-ce-cli containerd.io docker-buildx-plugin
docker-compose-plugin
```



```
 Installing:         7 packages

Total size of inbound packages is 99 MiB. Need to download 99 MiB.
After this operation, 402 MiB extra will be used (install 402 MiB, remove 0 B).
Is this ok [y/N]: y
[1/7] docker-ce-3:29.0.1-1.fc42.x86_64                                                                             100% |   7.0 MiB/s |  21.0 MiB |  00m03s
[2/7] docker-buildx-plugin-0:0.30.0-1.fc42.x86_64                                                                  100% |  24.3 MiB/s |  17.0 MiB |  00m01s
[3/7] docker-compose-plugin-0:2.40.3-1.fc42.x86_64                                                                 100% |  23.6 MiB/s |  15.0 MiB |  00m01s
[4/7] libcgroup-0:3.0-8.fc42.x86_64                                                                                100% | 224.3 KiB/s |  73.8 KiB |  00m00s
[5/7] docker-ce-rootless-extras-0:29.0.1-1.fc42.x86_64                                                             100% |   8.8 MiB/s |   3.4 MiB |  00m00s
[6/7] docker-ce-cli-1:29.0.1-1.fc42.x86_64                                                                         100% |   1.5 MiB/s |   8.3 MiB |  00m06s
[7/7] containerd.io-0:2.1.5-1.fc42.x86_64                                                                          100% |   4.8 MiB/s |  34.1 MiB |  00m07s
-----------------------------------------------------------------------------------------------------------------------------------------------------------
[7/7] Total                                                                                                        100% |  13.2 MiB/s |  98.9 MiB |  00m08s
[1/8] https://download.docker.com/linux/fedora/gpg                                            ???% [<=>               ] |   0.0   B/s |   0.0   B |  00m00s
[1/8] https://download.docker.com/linux/fedora/gpg                                            ???% [<=>               ] |   0.0   B/s |   0.0   B |  00m00s
[1/8] https://download.docker.com/linux/fedora/gpg                                                                 100% |  11.6 KiB/s |   1.6 KiB |  00m00s
-----------------------------------------------------------------------------------------------------------------------------------------------------------
[8/8] Total                                                                                                        100% |  13.2 MiB/s |  98.9 MiB |  00m08s
Importing OpenPGP key 0x621E9F35:
 UserID     : "Docker Release (CE rpm) <docker@docker.com>"
 Fingerprint: 060A61C51B558A7F742B77AAC52FEB6B621E9F35
 From       : https://download.docker.com/linux/fedora/gpg
Is this ok [y/N]: y
The key was successfully imported.
[1/9] Verify package files                                                                                         100% |  13.0   B/s |   7.0   B |  00m01s
[2/9] Prepare transaction                                                                                          100% |  17.0   B/s |   7.0   B |  00m00s
[3/9] Installing libcgroup-0:3.0-8.fc42.x86_64                                                                     100% |   2.6 MiB/s | 159.1 KiB |  00m00s
[4/9] Installing containerd.io-0:2.1.5-1.fc42.x86_64                                                               100% | 138.3 MiB/s | 115.8 MiB |  00m01s
[5/9] Installing docker-ce-cli-1:29.0.1-1.fc42.x86_64                                                              100% |  24.9 MiB/s |  34.2 MiB |  00m01s
[6/9] Installing docker-ce-3:29.0.1-1.fc42.x86_64                                                                  100% |  78.1 MiB/s |  90.2 MiB |  00m01s
[7/9] Installing docker-ce-rootless-extras-0:29.0.1-1.fc42.x86_64                                                  100% |  75.7 MiB/s |  11.3 MiB |  00m00s
[8/9] Installing docker-compose-plugin-0:2.40.3-1.fc42.x86_64                                                      100% | 195.9 MiB/s |  73.1 MiB |  00m00s
[9/9] Installing docker-buildx-plugin-0:0.30.0-1.fc42.x86_64                                                       100% |  20.3 MiB/s |  77.9 MiB |  00m04s
Complete!
calvin@fedora-vbox:~$
```

4. Docker engine must be enabled to constantly run and run on startup. This is completed using systemctl with the command:

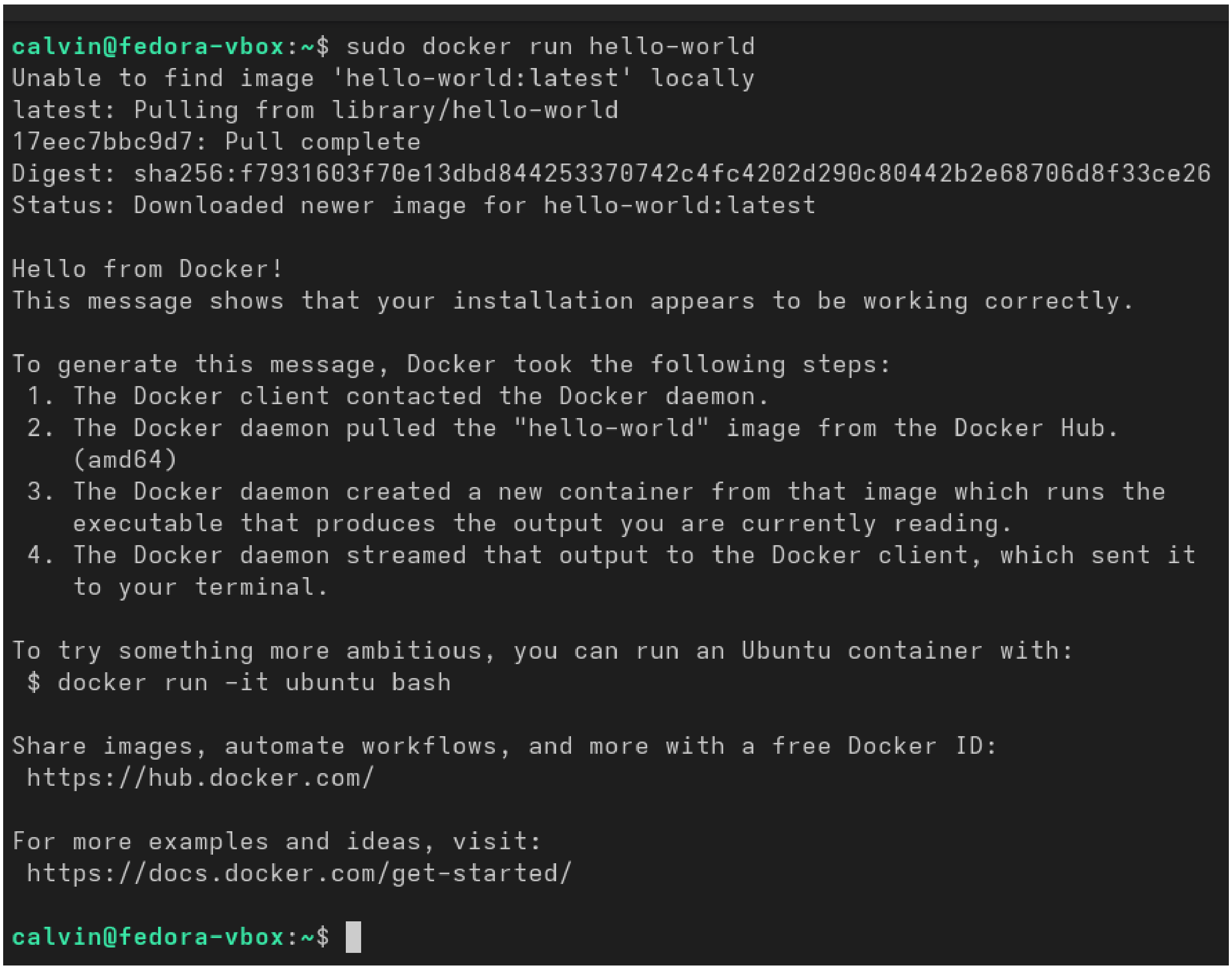
```
sudo systemctl enable --now docker
```



```
calvin@fedora-vbox:~$ sudo systemctl enable --now docker
Created symlink '/etc/systemd/system/multi-user.target.wants/docker.service' → '/usr/lib/systemd/system/docker.service'.
calvin@fedora-vbox:~$ 
```

5. Verify the docker installation with:

```
sudo docker run hello-world
```



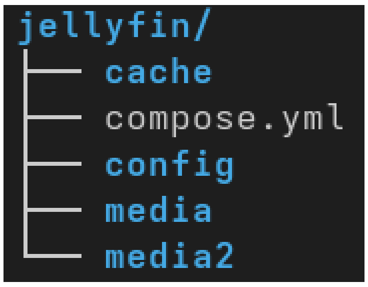
```
calvin@fedora-vbox:~$ sudo docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
17eec7bbc9d7: Pull complete
Digest: sha256:f7931603f70e13dbd844253370742c4fc4202d290c80442b2e68706d8f33ce26
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
 $ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
 https://hub.docker.com/

For more examples and ideas, visit:
 https://docs.docker.com/get-started/

calvin@fedora-vbox:~$
```

Create A Container

For this project, I opted to install Jellyfin, an open-source project for local video and audio streaming. The steps for installation are:

1. Locate the provided docker compose file
    - The docker compose file is found in the Jellyfin Documentation
    - This file is copied into the jellyfin directory

```
services:
  jellyfin:
    image: jellyfin/jellyfin
    container_name: jellyfin
    user: uid:gid
    ports:
      - 8096:8096/tcp
      - 7359:7359/udp
    volumes:
      - /path/to/config:/config
      - /path/to/cache:/cache
      - type: bind
        source: /path/to/media
        target: /media
      - type: bind
        source: /path/to/media2
        target: /media2
        read_only: true
      # Optional - extra fonts to be used during transcoding with subtitle burn-in
      - type: bind
        source: /path/to/fonts
        target: /usr/local/share/fonts/custom
        read_only: true
    restart: 'unless-stopped'
    # Optional - alternative address used for autodiscovery
    environment:
      - JELLYFIN_PublishedServerUrl=http://example.com
    # Optional - may be necessary for docker healthcheck to pass if running in
host network mode
    extra_hosts:
      - 'host.docker.internal:host-gateway'
```

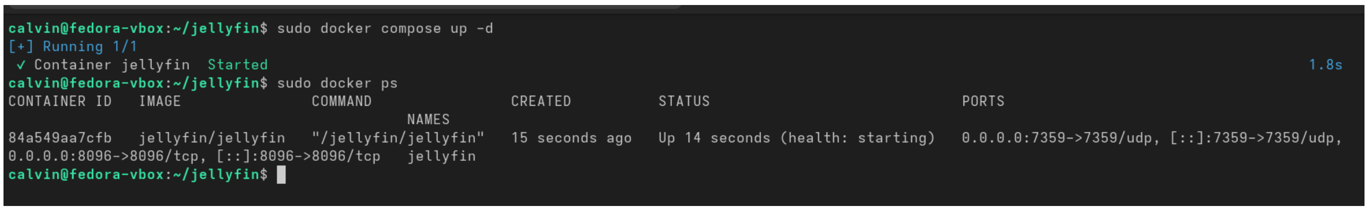
2. Modify and update docker compose file
   - All optional lines were removed, as this proof-of-concept does not need them
     - `extra_hosts`, `environment`, and the extra fonts volume
   - Subdirectories of the main jellyfin directory were made for config and media, and their paths were added to the docker compose file. The file structure was: 

```
jellyfin/
├── cache
├── compose.yml
├── config
├── media
└── media2
```

   - The user line was set to 1000:1000 indicating the "calvin" user and the "calvin" group (the default user on this virtual machine)

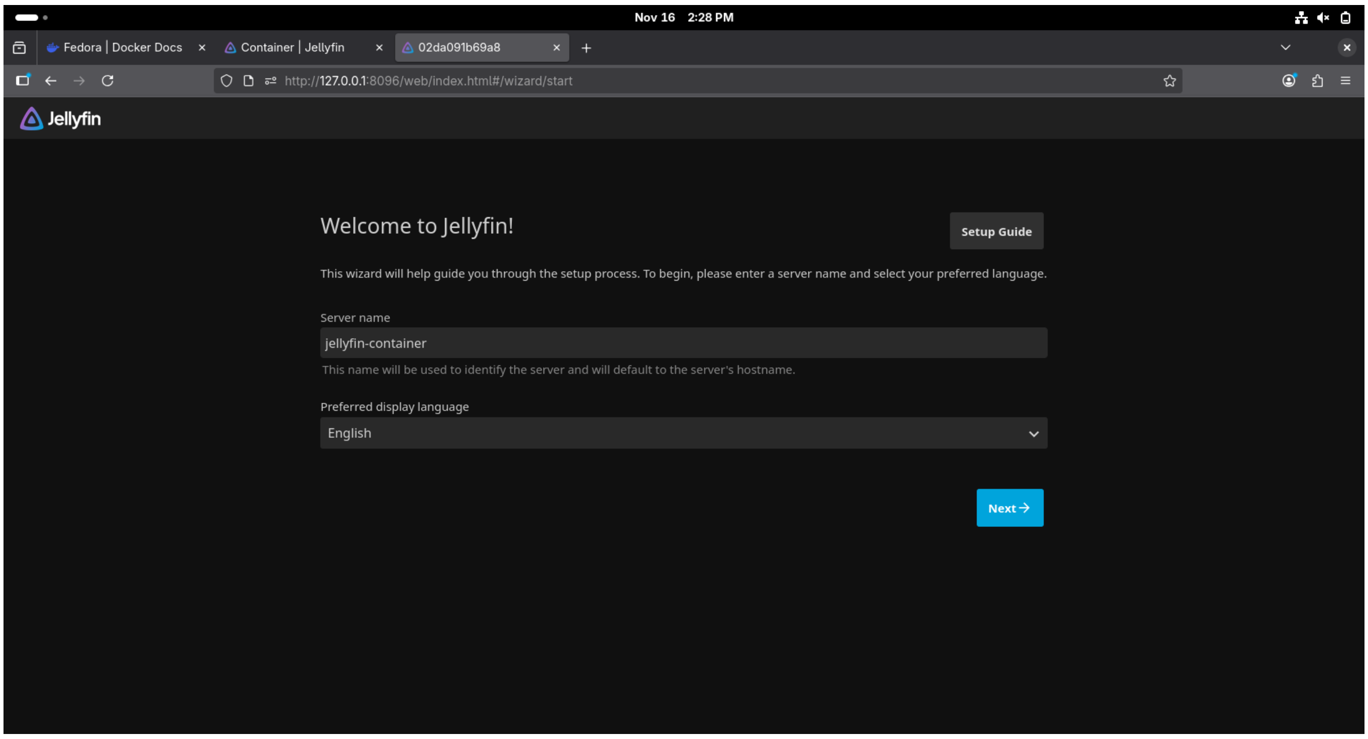
```
services:
  jellyfin:
    image: jellyfin/jellyfin
    container_name: jellyfin
    user: 1000:1000
    ports:
      - 8096:8096/tcp
      - 7359:7359/udp
    volumes:
      - /home/calvin/jellyfin/config:/config
      - /home/calvin/jellyfin/cache:/cache
      - type: bind
        source: /home/calvin/jellyfin/media
        target: /media
      - type: bind
        source: /home/calvin/jellyfin/media2
        target: /media2
        read_only: true
    restart: 'unless-stopped'
```

3. Make the docker container with `sudo docker compose up -d`
    - `-d` allows the container to run in detached mode



```
calvin@fedora-vbox:~/jellyfin$ sudo docker compose up -d
[+] Running 1/1
 ✓ Container jellyfin  Started                                                                                                      1.8s
calvin@fedora-vbox:~/jellyfin$ sudo docker ps
CONTAINER ID   IMAGE               COMMAND                 CREATED          STATUS                            PORTS
                                              NAMES
84a549aa7cfb   jellyfin/jellyfin   "/jellyfin/jellyfin"   15 seconds ago   Up 14 seconds (health: starting)   0.0.0.0:7359->7359/udp, [::]:7359->7359/udp,
0.0.0.0:8096->8096/tcp, [::]:8096->8096/tcp   jellyfin
calvin@fedora-vbox:~/jellyfin$ 
```

4. Login to jellyfin at the web address `http://localhost:8096` with any web browser (shown here is Firefox)
    - Complete setup and ensure jellyfin is working properly



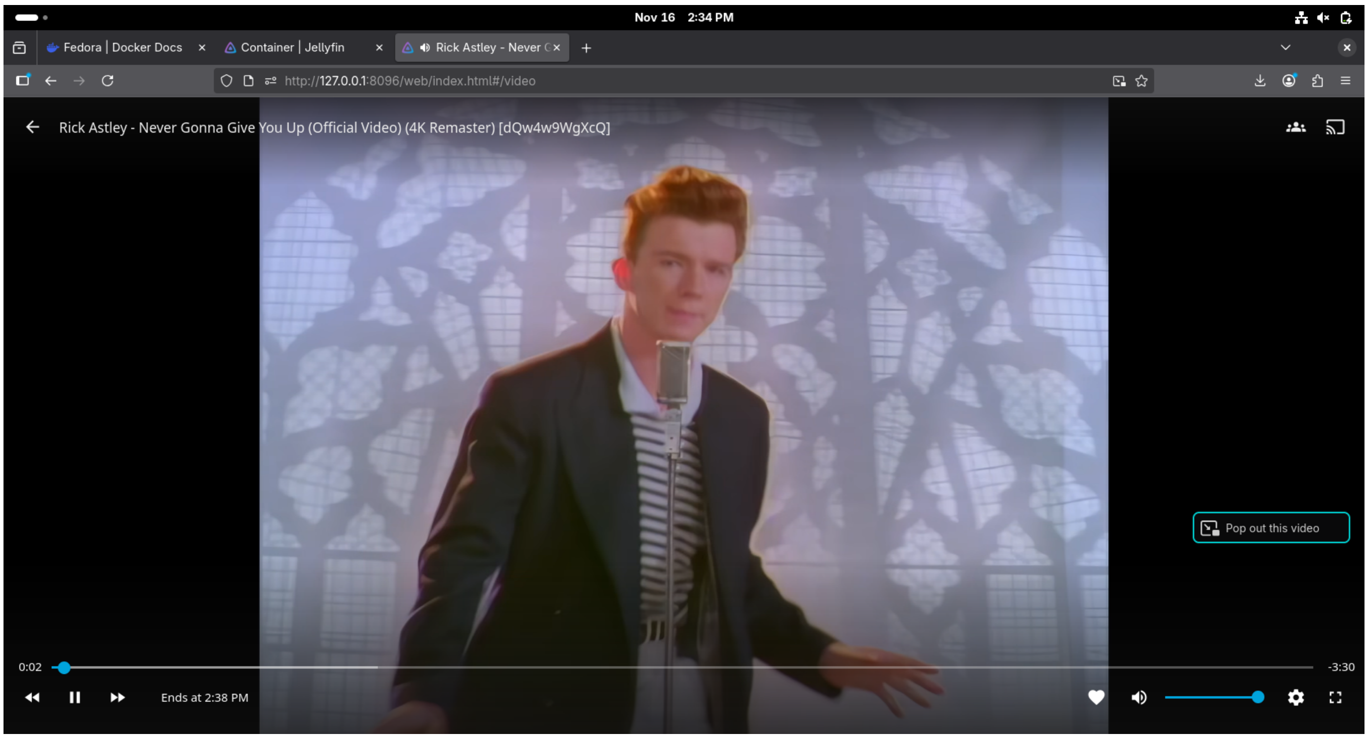
Welcome to Jellyfin!

This wizard will help guide you through the setup process. To begin, please enter a server name and select your preferred language.

Server name: jellyfin-container

This name will be used to identify the server and will default to the server's hostname.

Preferred display language: English


Nov 16 2:34 PM
Fedora | Docker Docs
Container | Jellyfin
Rick Astley - Never G
http://127.0.0.1:8096/web/index.html#/video
Rick Astley - Never Gonna Give You Up (Official Video) (4K Remaster) [dQw4w9WgXcQ]
Pop out this video
0:02
-3:30
Ends at 2:38 PM