

YT-DLP

Run YT-DLP inside the Docker container with the VPN support.

How to Use

Create an .env file (optional but recommended) in the same directory as your docker-compose.yml:

```
# Download Directory Path on Host
DOWNLOAD_DIR=/mnt/DATA/Downloads

# YT-DLP Options
AUTO_UPDATE=true
SERVICE_MODE=false
CHECK_INTERVAL=3600

# Timezone
TZ=Europe/Moscow

# OpenVPN Settings
VPN_OVPN_PATH=/absolute/path/to/your/config.ovpn # e.g.,
/home/username/vpn/config.ovpn

# Make sure that you have this in it:
# auth-user-pass /vpn/vpn.auth <-----
# ----- !!!
# -----
# -----

VPN_USERNAME=your_vpn_username
VPN_PASSWORD=your_vpn_password

# Use VPN toggle
USE_VPN=true
```

Replace the placeholders with your actual values. This file should be kept secure and not shared.

Then, run the container:

```
./start
```

Quick commands

- Download single video:

```
docker exec yt-dlp yt-dlp 'https://www.youtube.com/watch?v=VIDEO_ID'
```

- Download from batch file:

```
docker exec yt-dlp yt-dlp --batch-file /config/urls.txt
```

- Check container logs:

```
docker-compose logs -f yt-dlp
```

- Misc:

```
# Start yt-dlp with its own VPN
docker-compose --profile vpn up -d

# Check both VPN connections
echo "JDownloader VPN:"
docker exec openvpn wget -q0- ifconfig.me
echo -e "\nyt-dlp VPN:"
docker exec openvpn-yt-dlp wget -q0- ifconfig.me

# Download with yt-dlp
docker exec yt-dlp yt-dlp "https://www.youtube.com/watch?v=VIDEO_ID"

# View logs for yt-dlp VPN
docker logs -f openvpn-yt-dlp

# Stop only yt-dlp (leaves JDownloader running)
./cleanup.sh ytdlp

# More:

# Direct download
docker exec yt-dlp yt-dlp "https://www.youtube.com/watch?v=VIDEO_ID"

# Batch download
echo "https://www.youtube.com/watch?v=VIDEO1" >> ./yt-
dlp/config/urls.txt
docker exec yt-dlp /scripts/process-urls.sh

# Process subscribed channels
docker exec yt-dlp /scripts/process-channels.sh

# Enable service mode (automatic processing)
# Set SERVICE_MODE=true in .env and restart
```

Download script

```
./download.sh 'https://www.youtube.com/watch?v=VIDEO_ID'
```

with flags provided:

```
./download.sh --batch-file /config/urls.txt  
./download.sh --channels /config/channels.txt
```

Port Summary

With this setup, we will have (with [jDownloader](#) running in parallel):

- Port **3129**: JDownloader's VPN (original)
- Port **3130**: yt-dlp's VPN (new)
- Port **5800**: JDownloader Web UI
- Port **5900**: JDownloader VNC
- Port **8080**: Reserved for future yt-dlp Web UI
- Port **8081**: Reserved for future yt-dlp API

This configuration allows both services to run simultaneously with their own VPN connections without any conflicts.