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:

```
USE_VPN=true
VPN_USERNAME=your_vpn_username
VPN_PASSWORD=your_vpn_password
VPN_OVPN_PATH=/home/user/credentials.ovpn
DOWNLOAD_DIR=/mnt/remote/YT-DLP/

# ==========================================
# OPTIONAL (You can add if you want)
# ==========================================
TZ=Europe/Moscow
```

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 -qO- ifconfig.me
echo -e "\nyt-dlp VPN:"
docker exec openvpn-yt-dlp wget -qO- 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 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 'https://www.youtube.com/watch?v=VIDEO_ID'
```

with flags provided:

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

Port Summary

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

- Port `8086` → YoutubeDL-Material Web Interface (yt-dlp)
- Port `8081` → YoutubeDL-Material API
- Port `3130` → yt-dlp VPN
- Port `8085` → qBittorrent (already in use)
- Port `5800` → JDownloader Web UI
- Port `5900` → JDownloader VNC
- Port `3129` → JDownloader VPN

Access URLs

- yt-dlp Web Interface: http://amber.local:8086
- JDownloader: http://amber.local:5800
- qBittorrent: http://amber.local:8085

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

Note: The `amber.local` address represents imaginary machine in the network, update it according to your network configuration.