

Invidious

[Link](#)

1. Create [t-invidious-db.sh](#)

Create `/srv/path/Files/Invidious/docker/`

```
cd /srv/path/Files/Invidious/docker/
```

Create / download [init-invidious-db.sh](#)

```
#!/bin/bash
set -eou pipefail

psql --username "$POSTGRES_USER" --dbname "$POSTGRES_DB" < config/sql/channels.sql
psql --username "$POSTGRES_USER" --dbname "$POSTGRES_DB" < config/sql/videos.sql
psql --username "$POSTGRES_USER" --dbname "$POSTGRES_DB" < config/sql/channel_videos.sql
psql --username "$POSTGRES_USER" --dbname "$POSTGRES_DB" < config/sql/users.sql
psql --username "$POSTGRES_USER" --dbname "$POSTGRES_DB" < config/sql/session_ids.sql
psql --username "$POSTGRES_USER" --dbname "$POSTGRES_DB" < config/sql/nonces.sql
psql --username "$POSTGRES_USER" --dbname "$POSTGRES_DB" < config/sql/annotations.sql
psql --username "$POSTGRES_USER" --dbname "$POSTGRES_DB" < config/sql/playlists.sql
psql --username "$POSTGRES_USER" --dbname "$POSTGRES_DB" < config/sql/playlist_videos.sql
```

2. Generate po_token and visitor_data identities:

Generate po_token and visitor_data identities for passing all verification checks on YouTube side:

```
docker run quay.io/invidious/youtube-trusted-session-generator
```

You have to run this command on the same public IP address as the one blocked by YouTube. Not necessarily the same machine, just the same public IP address.

You will need to copy these two parameters in the third step.

Subsequent usage of this same token will work on the same IP range or even the same ASN. The point is to generate this token on a blocked IP as "unblocked" IP addresses seems to not generate a token valid for passing the checks on a blocked IP.

3. Stack

```
version: "3"
services:

  invidious:
    image: quay.io/invidious/invidious:latest
    # image: quay.io/invidious/invidious:latest-arm64 # ARM64/AArch64 devices
    restart: unless-stopped
    ports:
      - "3000:3000"
    environment:
      # Please read the following file for a comprehensive list of all available
      # configuration options and their associated syntax:
      # https://github.com/iv-org/invidious/blob/master/config/config.example.yml
      INVIDIOUS_CONFIG: |
        db:
          dbname: invidious
          user: kemal
          password: kemal
          host: invidious-db
          port: 5432
        check_tables: true
        signature_server: inv_sig_helper:12999
        visitor_data: "CHANGE_ME!!"
        po_token: CHANGE_ME!!"
        external_port: 443
        domain: your.domain.com # Don't put "https://"
        https_only: true
        statistics_enabled: true
        hmac_key: "CHANGE_ME!!" #generate using command "pwgen 20 1" or "openssl rand -hex 20"
or just 20 random characters
        admins: ["admin"]
    healthcheck:
      test: wget -nv --tries=1 --spider http://127.0.0.1:3000/api/v1/trending || exit 1
      interval: 30s
      timeout: 5s
      retries: 2
    logging:
      options:
        max-size: "1G"
        max-file: "4"
```

```
depends_on:
  - invidious-db

inv_sig_helper:
  image: quay.io/invidious/inv-sig-helper:latest
  init: true
  command: ["--tcp", "0.0.0.0:12999"]
  environment:
    - RUST_LOG=info
  restart: unless-stopped
  cap_drop:
    - ALL
  read_only: true
  security_opt:
    - no-new-privileges:true

invidious-db:
  image: docker.io/library/postgres:14
  restart: unless-stopped
  volumes:
    - /srv/Files/Invidious/postgresdata:/var/lib/postgresql/data
    - /srv/Files/Invidious/config/sql:/config/sql
    - /srv/Files/Invidious/docker/init-invidious-db.sh:/docker-entrypoint-initdb.d/init-
invidious-db.sh
  environment:
    POSTGRES_DB: invidious
    POSTGRES_USER: kema1
    POSTGRES_PASSWORD: kema1
  healthcheck:
    test: ["CMD-SHELL", "pg_isready -U $$POSTGRES_USER -d $$POSTGRES_DB"]

volumes:
  postgresdata:
```

The environment variable `POSTGRES_USER` cannot be changed. The SQL config files that run the initial database migrations are hard-coded with the username `kema1`.

Same goes with environment variable `port`. Stick to `3000:3000` and change other service port if needed.

Detailed post-install configuration available in the [configuration guide](#).

If you use a reverse proxy, you **must** configure invidious to properly serve request through it:

`https_only: true` : if your are serving your instance via https, set it to true

`domain: domain.ext` : if you are serving your instance via a domain name, set it here (no "https://")

`external_port: 443` : if your are serving your instance via https, set it to 443

Env variables must be put after `INVIDIOUS_CONFIG: |` in the compose file

[Open youtube links in Invidious](#) | [Link 2](#)

Revision #16

Created 23 January 2022 16:11:25 by Admin

Updated 18 April 2025 12:23:53 by Admin