IP Geo blocking

Qool restricts access to the Netherlands using NGINX GeoIP2.

1. Install the NGINX GeoIP2 module

```
sudo apt update
sudo apt install libnginx-mod-http-geoip2
```

Ubuntu should automatically create:

```
/etc/nginx/modules-enabled/50-mod-http-geoip2.conf
```

With the contents:

```
load_module modules/ngx_http_geoip2_module.so;
```

2. Download the country database

DB-IP publishes its free country database monthly in MMDB format. Select the current release from the official DB-IP download page. The free database requires attribution under CC BY 4.0.

For the November 2026 release:

```
cd /tmp

curl -fL \
  https://download.db-ip.com/free/dbip-country-lite-2026-11.mmdb.gz \
  -o dbip-country-lite.mmdb.gz

gzip -dc dbip-country-lite.mmdb.gz \
  > dbip-country-lite.mmdb

sudo install -d -o root -g root -m 0755 /usr/share/GeoIP

sudo install -o root -g root -m 0644 \
  dbip-country-lite.mmdb \
```

```
  /usr/share/GeoIP/dbip-country-lite.mmdb
```

DB-IP documents that Lite databases are updated monthly, so the `YYYY-MM` value must periodically be updated. DB-IP documentation

3. Define the country policy

Create `/etc/nginx/conf.d/nl-country-map.conf`:

```
geoip2 /usr/share/GeoIP/dbip-country-lite.mmdb {
    auto_reload 1h;
    $visitor_country_code default=ZZ country iso_code;
}

map $remote_addr $trusted_ip {
    default 0;

    # Replace this with your own trusted public IP.
    203.0.113.10 1;
}

map "$visitor_country_code:$trusted_ip" $country_allowed {
    default 0;

    # Allow clients classified as Dutch.
    ~^NL: 1;

    # Allow explicitly trusted source addresses.
    ~:1$ 1;
}
```

Alternative (more generic)

```
geoip2 /usr/share/GeoIP/dbip-country-lite.mmdb {
    auto_reload 1h;
    $visitor_country_code default=ZZ country iso_code;
}

# Allowed countries
map $visitor_country_code $allowed_country {
```

```
    default 0;
    NL 1;
    # BE 1;
    # DE 1;
}

# Trusted IP addresses
map $remote_addr $trusted_ip {
    default 0;
    109.199.113.197 1;
}

# Googlebot IP ranges
geo $googlebot_ip {
    default 0;
    include /etc/nginx/googlebot-ips.conf;
}

# Allow if ANY condition is true
map "$allowed_country$trusted_ip$googlebot_ip" $country_allowed {
    default 0;
    ~1 1;
}
```

4. Create the enforcement snippet

Create `/etc/nginx/snippets/some-countries-only.conf`:

```
# Reusable Netherlands-only access policy.
# Requires $nl_country_allowed in the HTTP context.
if ($country_allowed = 0) {
    return 403;
}
```

Using `if` with a simple `return` is appropriate here.

5. Enable it for a website

Inside the applicable `server` block:

```
server {
    listen 443 ssl;
    server_name example.com;

    include snippets/some-countries-only.conf;

    # Remaining website configuration...
}
```

6. Test before reloading

```
sudo nginx -t
```

Only when that succeeds:

```
sudo systemctl reload nginx
```

Test from:

- A Dutch connection, which should receive the normal website.
- A non-Dutch VPN endpoint, which should receive HTTP `403`.
- Your trusted IP, which should work regardless of its GeoIP classification.

You can inspect NGINX’s effective configuration with:

```
sudo nginx -T
```

Important considerations

- If NGINX is behind Cloudflare, another reverse proxy, or a load balancer, $remote_addr may be the proxy address. Configure NGINX’s real-IP module correctly before applying this restriction. Do not blindly trust client-supplied forwarding headers.
- A server-level restriction can also block Let’s Encrypt HTTP-01 validation. If Certbot uses HTTP-01, provide a separate unrestricted ACME challenge virtual host/location or use DNS-01 validation.
- GeoIP is not an absolute security boundary. VPNs, proxies, database inaccuracies, and stale records can bypass or incorrectly trigger it.

- The qool server currently has no detected automatic database updater. A monthly replacement of the MMDB file should therefore be scheduled if you want current classifications.
- SSH access is unaffected; this configuration only restricts websites served through the applicable NGINX virtual host.

Update Database

Save this as `/usr/local/sbin/update-dbip-country-lite`:

```
#!/usr/bin/env bash

set -euo pipefail

DESTINATION="/usr/share/GeoIP/dbip-country-lite.mmdb"
DOWNLOAD_BASE="https://download.db-ip.com/free"
TEMP_DIR="$(mktemp -d)"

cleanup() {
    rm -rf "${TEMP_DIR}"
}
trap cleanup EXIT

if [[ "${EUID}" -ne 0 ]]; then
    echo "Run this script as root: sudo $0" >&2
    exit 1
fi

for RELEASE in \
    "$(date -u +%Y-%m)" \
    "$(date -u -d '1 month ago' +%Y-%m)"
do
    URL="${DOWNLOAD_BASE}/dbip-country-lite-${RELEASE}.mmdb.gz"
    ARCHIVE="${TEMP_DIR}/database.mmdb.gz"
    DATABASE="${TEMP_DIR}/database.mmdb"

    echo "Trying DB-IP release ${RELEASE}..."

    if curl --fail --location --silent --show-error \
```

```
        --retry 3 \
        --output "${ARCHIVE}" \
        "${URL}"
    then
        gzip --test "${ARCHIVE}"
        gzip --decompress --stdout "${ARCHIVE}" > "${DATABASE}"

        if [[ ! -s "${DATABASE}" ]]; then
            echo "Downloaded database is empty." >&2
            exit 1
        fi

        install -d -o root -g root -m 0755 \
            "$(dirname "${DESTINATION}")"

        # Install under a temporary name and then replace atomically.
        install -o root -g root -m 0644 \
            "${DATABASE}" \
            "${DESTINATION}.new"

        mv -f "${DESTINATION}.new" "${DESTINATION}"

        echo "Installed DB-IP Country Lite release ${RELEASE}:"
        stat "${DESTINATION}"
        exit 0
    fi
done

echo "Could not download the current or previous monthly DB-IP release." >&2
exit 1
```

Maker it executable

```
sudo chmod 0755 /usr/local/sbin/update-dbip-country-lite
```

Run

```
sudo /usr/local/sbin/update-dbip-country-lite
```

The script:

- Tries the current month first.
- Falls back to the previous month if the current release is not yet available.
- Validates the gzip archive.
- Refuses to install an empty database.
- Uses a temporary download directory.
- Replaces the live database atomically.
- Preserves `root:root` ownership and mode `0644`.

Because your NGINX `geoip2` configuration uses:

```
auto_reload 1h;
```

NGINX should detect the replacement within an hour; a reload is unnecessary.

cron

To run it automatically every month, add `/etc/cron.d/update-dbip-country-lite`:

```
# Update DB-IP Country Lite at 04:17 on the third day of each month.
17 4 3 * * root /usr/local/sbin/update-dbip-country-lite >> /var/log/update-dbip-country-lite.log 2>&1
```

--

Revision #6
Created 2026-11-10 16:16:52 UTC by Max
Updated 2026-11-10 20:44:55 UTC by Max