GeoIP discontinuation; Upgrade to GeoIP2 with nginx on CentOS

Karl Johnson Follow
Feb 5 · 3 min read

GeoLite Legacy databases are discontinued as of January 2, 2019, they are not updated nor any longer available for download. Every user should move to GeoLite2 databases, a more contemporary versions of the GeoLite Legacy geolocation databases which are still available in a free version updated every month.

I've recently added the GeoIP2 module to nginx-more package for CentOS 6 and 7 to easily migrate from GeoIP to GeoIP2. By using this package, only a couple of lines in nginx configuration file are needed to configure the module. However, this module needs libmaxminddb which can be found in EPEL. To avoid repository dependency, I've rebased the package libmaxminddb on its latest version and made it available in the same yum repository.

GeoIP2 module installation

If nginx-more is already installed on your system, a simple yum update will make this module available, else let's install nginx-more on a new system:

```
CentOS 6 #> yum -y install
https://repo.aerisnetwork.com/pub/aeris-release-6.rpm

CentOS 7 #> yum -y install
https://repo.aerisnetwork.com/pub/aeris-release-7.rpm

#> yum install nginx-more
```

GeoIP2 configuration

First, we need latest MaxMind GeoLite2 country/city free databases:

```
mkdir /usr/share/GeoIP2
wget -P /usr/share/GeoIP2
https://geolite.maxmind.com/download/geoip/database/GeoLite2
-City.mmdb.gz
wget -P /usr/share/GeoIP2
https://geolite.maxmind.com/download/geoip/database/GeoLite2
-Country.mmdb.gz
gunzip /usr/share/GeoIP2/*.mmdb.gz
```

Then we need to load it in nginx.conf, preferably in the http section:

```
geoip2 /usr/share/GeoIP2/GeoLite2-Country.mmdb {
    auto_reload 60m;
    $geoip2_metadata_country_build metadata build_epoch;
    $geoip2_data_country_code country iso_code;
    $geoip2_data_country_name country names en;
}
```

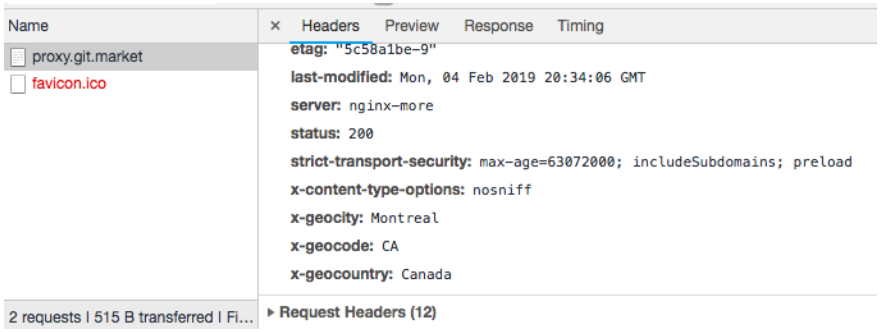
```
geoip2 /usr/share/GeoIP2/GeoLite2-City.mmdb {
    auto_reload 60m;
    $geoip2_metadata_city_build metadata build_epoch;
    $geoip2_data_city_name city names en;
}
```

Testing GeoIP2 country and city fetching

The easiest way to confirm that nginx is fetching all geo information is using a temporary header on any vhosts:

```
add_header X-GeoCountry $geoip2_data_country_name;
add_header X-GeoCode $geoip2_data_country_code;
add_header X-GeoCity $geoip2_data_city_name;
```

Your browser inspector should display your country name, country code and city name:



nginx GeoIP2 module headers

Block countries from accessing your website

Now that GeoIP2 module is confirmed to work, we can restrict a vhost to specific countries. Let's restrict public directory of this vhost to only visitors from Canada and USA:

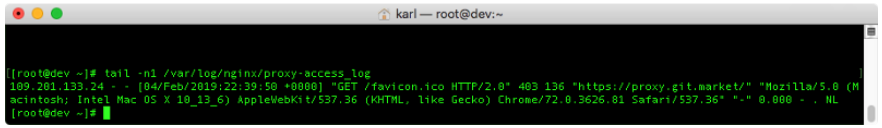
```
map $geoip2_data_country_code $allowed_country {
 default no;
 CA yes;
 US yes;
}


location / {
 if ($allowed_country = no) {
  return 403;
 }
}
```

Display country ID in nginx access_log

It can be useful for debugging or monitoring to see country code of each visitor in nginx access_log. Simply add $geoip2_data_country_code in your nginx log_format:

```
log_format main
 '$remote_addr - $remote_user [$time_local] "$request" '
 '$status $body_bytes_sent "$http_referer" '
 '"$http_user_agent" "$http_x_forwarded_for" '
```

```
    '$request_time $upstream_response_time $pipe '
    '$geoip2_data_country_code';
```



GeoIP2 nginx access_log country code exemple

About nginx-more

Nginx-more is a free CentOS RPM of Nginx with additional modules such as HTTP2, PageSpeed, Brotli, More Headers, Cache Purge, VTS, GeoIP2. It's compiled using recent GCC version and latest OpenSSL sources. It also includes some built-in configurations such as WordPress and Laravel php-fpm setup. The package is supported since 2014 and it will be ported to CentOS 8 when it will be released!