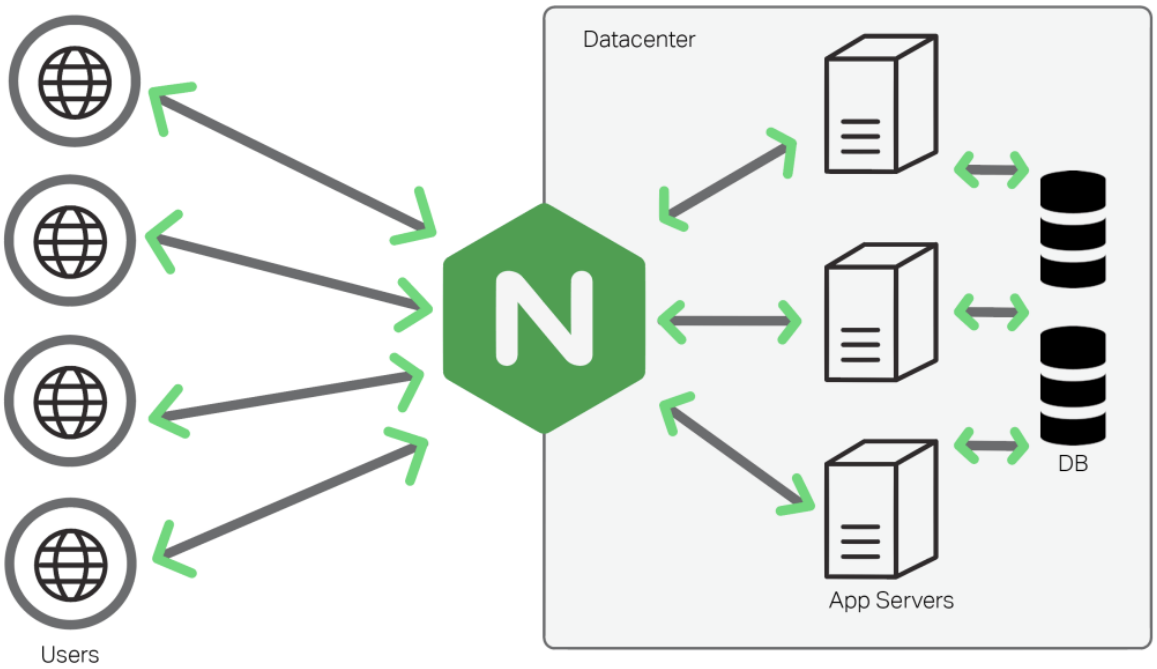
A Byte of Nginx

Sheng Yuan

June 23, 2018

Contents

Introduce



Nginx is a web server which can also be used as a reverse proxy, load balancer, mail proxy and HTTP cache.

Basics

```
$ nginx
$ nginx -s reload
$ nginx -s stop
$ nginx -s quit
$ nginx -s reopen
$ nginx -f /conf/nginx.conf
```

```
$ service nginx start
$ service nginx stop
$ service nginx restart
```

Config Grammar

- **simple directive** the name and parameters separated by spaces and ends with a semicolon (;)
- **block directive** the same structure as a simple directive, but instead of the semicolon it ends with a set of additional instructions surrounded by braces ({ and })

```
http {
  server {
    location / {
      root /data/www;
    }
    location /images/ {
      root /data;
    }
  }
}
```

Web Server

Simple Web Server

```
http {
  server {
    location / {
      root /data/www;
    }
    location /images/ {
      root /data;
    }
  }
}
```

The static file locations:
/data/www
/data/images

Web Server with alias

```
http {
  server {
    location /static {
      alias /opt/alexa/public;
    }
  }
}
```

The static file locations:
/opt/alexa/public

Proxy Server

Solution

```
server {
  location /debug {
    proxy_pass http://127.0.0.1:8000/debug;
  }

  location /static {
    root /data/public;
  }

  location /api/v2 {
    proxy_http_version 1.1;
    add_header Access-Control-Allow-Origin *;
    proxy_pass http://127.0.0.1:8080/api/v2;
  }
}
```

HTTP Load Balance

Problem

You need to distribute load between two or more HTTP servers.

Solution

```
upstream backend {
  server 10.10.12.45:80 weight=1;
  server app.example.com:80 weight=2;

}
server {
  location / {
    proxy_pass http://backend;
  }
}
```

TCP Load Balance

Problem

You need to distribute load between two or more TCP servers.

Solution

```
stream {
  upstream mysql_read {
    server read1.example.com:3306 weight=5;
    server read1.example.com:3306;
    server 10.10.12.34:3306 backup;
  }
  server {
    listen 3306;
    proxy_pass mysql_read;
  }
}
```

Load Balancing methods

- Round robin
- Least connections
- Generic hash
- Least time
- IP hash

Connection Limiting

Massively Scalable Content Caching

Problem

You need to cache content and need to define where the cache is stored.

Solution

```
proxy_cache_path /var/nginx/cache

  keys_zone=CACHE:60m levels=1:2
  inactive=3h
  max_size=20g;

proxy_cache CACHE;
```

Controlling Access

Solution

```
location /admin/ {
  deny 10.0.0.1;
  allow 10.0.0.0/20;
  allow 2001:0db8::/32;
  deny all;
}
```

Force Https

```
server {
  listen 80; return 301 https://testai.tclking.com$request_uri;
}
```

```
add_header Strict-Transport-Security max-age=31536000;
```

Configuring Logs

```
http {
  log_format geoproxy
    '[$time_local] $remote_addr '
    '$realip_remote_addr $remote_user '
    '$request_method $server_protocol '
    '$scheme $server_name $uri $status '
    '$request_time $body_bytes_sent '
    '$geoip_city_country_code3 $geoip_region '
    '$geoip_city" $http_x_forwarded_for '
    '$upstream_status $upstream_response_time '
    '"$http_referer" "$http_user_agent"';

  access_log /var/log/nginx/access.log geoproxy buffer=32k
  flush=1m;

  error_log /var/log/nginx/error.log main buffer=32k
  flush=1m;
}
```

The buffer parameter of the access_log directive denotes the size of a memory buffer that can be filled with log data before being written to disk. The flush parameter of the access_log directive sets the longest amount of time a log can remain in a buffer before being written to disk.

Performance Tuning

Keeping Connections Open to Clients

```
http {
  keepalive_requests 320;
  keepalive_timeout 300s;
}
```

Keeping Connections Open Upstream

```
proxy_http_version 1.1;
proxy_set_header Connection "";
upstream backend {
  server 10.0.0.42;
  server 10.0.2.56;
  keepalive 32;
}
```

OS Tuning

- Raising the number of open file descriptors is a more common need.
- Check the kernel setting for net.core.somaxconn, which is the maximum number of connections that can be queued by the kernel for NGINX to process.
- Enable more ephemeral ports.

Other Tools

Tools

mongo	mongoexport	mongooplog	mongos
mongod	mongofiles	mongoperf	mongostat
mongodump	mongoimport	mongorestore	mongotop

mongooplog Pulls oplog entries from another mongod instance.

mongos MongoDB shard process.

mongoperf Check disk I/O performance.

mongostat Returns counters of database operation.

mongotop Tracks/reports MongoDB read/write activities.

References

Chodorow, K. (2013).

MongoDB: The definitive guide.

O'Reilly Media, Inc.

Plugge, E., Hows, D., Membrey, P., and Hawkins, T. (2015).

The Definitive Guide to MongoDB: A complete guide to dealing with Big Data using MongoDB.

Apress.