



NGINX HTTP SERVER

Mithun Technologies, devopstrainingblr@gmail.com, +91-99809 23226

A large, faint watermark of the Mithun Technologies logo is centered on the page. The logo consists of a stylized blue 'T' shape with a bird-like figure at the top, and the text 'Mithun Technologies' in a blue sans-serif font below it.

Mithun
Technologies

Mithun
Technologies

MITHUN SOFTWARE SOLUTIONS
Bangalore

Nginx – HTTP Server

Install Nginx in Redhat

```
dnf install https://dl.fedoraproject.org/pub/epel/epel-release-latest-8.noarch.rpm
```

```
sudo yum install nginx
```

#Enable the nginx service

```
sudo systemctl enable nginx
```

#Start the nginx service

```
sudo systemctl start nginx
```

#Check the nginx service status

```
sudo systemctl status nginx
```

Rename default nginx conf file

```
sudo mv /etc/nginx/nginx.conf /etc/nginx/nginx.conf.bkp
```

Create nginx conf file in default path and define proxy(reverse proxy) rules.

```
sudo vi /etc/nginx/nginx.conf
```

Add the below lines

```
events{
    worker_connections 1024;
}
http {
    keepalive_timeout 5;
    upstream tomcatServers {
        keepalive 50;
        server 52.221.224.100:9980;
        server 18.141.224.37:8080;
    }
    server {
        listen 80;
```

Mithun Technologies +91-9980923226	Nginx HTTP server devopstrainingblr@gmail.com	Author Web site	Mithun Technologies http://mithuntechnologies.com
---------------------------------------	--	--------------------	--

```

location / {
    proxy_set_header    X-Real-IP $remote_addr;
    proxy_set_header    X-Forwarded-For
$proxy_add_x_forwarded_for;
    proxy_set_header    X-Forwarded-Proto $scheme;
    proxy_set_header    Host $host;
    proxy_pass http://tomcatServers;
}
}
}

```

Validate the nginx.conf by using below command

sudo nginx -t

Reload configurations

sudo nginx -s reload

Restart nginx Service to Reflect the proxy rules.

sudo systemctl restart nginx

Note: If you get 502 Bad GateWay

Security-Enhanced Linux (SELinux) is a Linux kernel security module for Linux systems that allows administrators to have more control over who can access the system. If SELinux enabled, which blocked Nginx from making outbound connections.

If SELinux is on and you're experiencing this, you might try setting httpd_can_network_connect to true, and then restarting nginx:

```
# setsebool -P httpd_can_network_connect true
```

Choosing a Load-Balancing Method

Round Robin – Requests are distributed evenly across the servers, with server weights taken into consideration. This method is used by default.

Mithun Technologies +91-9980923226	Nginx HTTP server devopstrainingblr@gmail.com	Author	Mithun Technologies
		Web site	http://mithuntechnologies.com

```
upstream backend {
    # no load balancing method is specified for Round Robin
    server mithuntechnologies.app1.com;
    server mithuntechnologies.app2.com;
}
```

Least Connections – A request is sent to the server with the least number of active connections, again with server weights taken into consideration:

```
upstream backend {
    least_conn;
    server mithuntechnologies.app1.com;
    server mithuntechnologies.app2.com;
}
```

IP Hash – The server to which a request is sent is determined from the client IP address. In this case, either the first three octets of the IPv4 address or the whole IPv6 address are used to calculate the hash value. The method guarantees that requests from the same address get to the same server unless it is not available.

```
upstream backend {
    ip_hash;
    server mithuntechnologies.app1.com;
    server mithuntechnologies.app2.com;
}
```

If one of the servers needs to be temporarily removed from the load-balancing rotation, it can be marked with the down parameter in order to preserve the current hashing of client IP addresses. Requests that were to be processed by this server are automatically sent to the next server in the group:

```
upstream backend {
    server mithuntechnologies.app1.com;
    server mithuntechnologies.app2.com;
    server mithuntechnologies.app3.com down;
}
```

Mithun Technologies +91-9980923226	Nginx HTTP server devopstrainingblr@gmail.com	Author Web site	Mithun Technologies http://mithuntechnologies.com
---------------------------------------	--	--------------------	--

Server Weights:

By default, NGINX distributes requests among the servers in the group according to their weights using the Round Robin method. The weight parameter to the server directive sets the weight of a server; the default is 1:

```
upstream backend {
    server mithuntechnologies.app1.com weight=5;
    server mithuntechnologies.app2.com;
    server mithuntechnologies.app3.com backup;
}
```

In the example, mithuntechnologies.app1.com has weight 5; the other two servers have the default weight (1), but the one with domain name mithuntechnologies.app3.com is marked as a backup server and does not receive requests unless both of the other servers are unavailable. With this configuration of weights, out of every 6 requests, 5 are sent to mithuntechnologies.app1.com and 1 to mithuntechnologies.app2.com.

Tuning Your NGINX Configuration

The following are some NGINX directives that can impact performance. As stated above, we only discuss directives that are safe for you to adjust on your own. We recommend that you not change the settings of other directives without direction from the NGINX team.

Worker Processes

NGINX can run multiple worker processes, each capable of processing a large number of simultaneous connections. You can control the number of worker processes and how they handle connections with the following directives:

worker_processes – The number of NGINX worker processes (the default is 1). In most cases, running one worker process per CPU core works well, and we recommend setting this directive to auto to achieve that. There are times when you may want to increase this number, such as when the worker processes have to do a lot of disk I/O.

Mithun Technologies +91-9980923226	Nginx HTTP server devopstrainingblr@gmail.com	Author	Mithun Technologies
		Web site	http://mithuntechnologies.com

worker_connections – The maximum number of connections that each worker process can handle simultaneously. The default is 512, but most systems have enough resources to support a larger number. The appropriate setting depends on the size of the server and the nature of the traffic, and can be discovered through testing.

Keepalive Connections

Keepalive connections can have a major impact on performance by reducing the CPU and network overhead needed to open and close connections. NGINX terminates all client connections and creates separate and independent connections to the upstream servers. NGINX supports keepalives for both clients and upstream servers. The following directives relate to client keepalives:

keepalive_requests – The number of requests a client can make over a single keepalive connection. The default is 100, but a much higher value can be especially useful for testing with a load-generation tool, which generally sends a large number of requests from a single client.

keepalive_timeout – How long an idle keepalive connection remains open. The following directive relates to upstream keepalives:

keepalive – The number of idle keepalive connections to an upstream server that remain open for each worker process. There is no default value.