# \*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Nginx \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

## Nginx Configuration Code:

upstream backend\_app {   
 ip\_hash;   
 server 127.0.0.1:3001;  
 server 127.0.0.1:8000;  
}  
  
server {  
 listen 443 ssl;  
 server\_name localhost;  
  
 ssl\_certificate /etc/ssl/certs/nginx-selfsigned.crt;  
 ssl\_certificate\_key /etc/ssl/private/nginx-selfsigned.key;  
  
 location / {  
 proxy\_pass http://backend\_app;  
 proxy\_set\_header Host $host;  
 proxy\_set\_header X-Real-IP $remote\_addr;  
 }  
}  
  
# Optional: Redirect HTTP to HTTPS  
server {  
 listen 80;  
 server\_name localhost;  
 return 301 https://$host$request\_uri;  
}

## Summary Table:

|  |  |
| --- | --- |
| Section | Purpose |
| upstream backend\_app | Defines two backend servers (3001 & 8000) for load balancing |
| server { listen 443 ssl; } | Handles HTTPS requests, proxies them to backend servers |
| ssl\_certificate & ssl\_certificate\_key | Enable SSL encryption |
| proxy\_pass backend\_app | Sends traffic to the upstream backend group |
| proxy\_set\_header | Preserves original client headers (for logging or app logic) |
| server { listen 80; } | Redirects HTTP → HTTPS |

## 1. Upstream Block

This block defines a load balancing group named 'backend\_app'.  
  
• upstream backend\_app { ... }  
Defines a group of backend servers that Nginx can send traffic to (load balancing pool).  
  
• ip\_hash;  
Ensures that requests from the same client IP always go to the same backend server. Useful for sticky sessions.

| Algorithm | Behavior |
| --- | --- |
| round-robin | Default — rotates through all backends equally |
| least\_conn | Sends traffic to the backend with the fewest active connections |
| ip\_hash | Uses client IP to consistently route requests to the same backend |

• server 127.0.0.1:3001;  
First backend application server — runs locally on port 3001.  
  
• server 127.0.0.1:8000;  
Second backend server — runs locally on port 8000.  
  
Result: When a client connects, Nginx will forward the request to either port 3001 or 8000 based on the client’s IP.

## 2. HTTPS Server Block

This block handles HTTPS traffic (port 443) and forwards it to the backend servers.  
  
• listen 443 ssl;  
Makes Nginx listen on port 443 (HTTPS) and enables SSL/TLS.  
  
• server\_name localhost;  
The domain name this server responds to. Here it’s localhost.  
  
• ssl\_certificate / ssl\_certificate\_key  
Specifies the paths to SSL/TLS certificate and private key.  
  
• location / { ... }  
Handles requests for all paths. The proxy\_pass sends traffic to the upstream backend\_app group.  
  
• proxy\_set\_header Host $host;  
Passes the original Host header to the backend.  
  
• proxy\_set\_header X-Real-IP $remote\_addr;  
Passes the client’s real IP address to the backend application.

## 3. HTTP → HTTPS Redirect

Redirects all HTTP requests (port 80) to HTTPS (port 443).  
  
• listen 80;  
Listens for unencrypted HTTP traffic.  
  
• server\_name localhost;  
Matches HTTP requests to localhost.  
  
• return 301 https://$host$request\_uri;  
Issues a 301 Permanent Redirect to the HTTPS version of the requested URL.  
Example:  
http://localhost/test → https://localhost/test