# NGINX AS A LOAD BALANCER

This project demonstrates using **Nginx** to distribute incoming traffic across multiple backend servers, a process known as **load balancing**. This improves performance and reliability by preventing any single server from becoming overwhelmed. The project repository is

<https://github.com/jaiswaladi246/nginx-loadbalancer.git>.

#### Steps

1. **Define Backend Servers**: In your Nginx configuration file, for example, /etc/nginx/sites-available/nginx-loadbalancer , create an

upstream block named backend\_apis. Inside this block, you will list the addresses of your backend servers. The

least\_conn directive is used to distribute requests to the server with the fewest active connections.

1. **Configure Proxy Pass**: In the main server block, the location /api/ directive will be configured to use the proxy\_pass to the http://backend\_apis group you defined earlier.
2. **Start Backend Services**: Start each backend service in a separate terminal. The document suggests running

node index.js in both backend1 and backend2 directories.

1. **Enable and Reload**: As with the other projects, create a symbolic link to enable the site and then reload Nginx.
2. **Simulate Traffic**: To test the load balancing, you can use a tool like ab (Apache Bench). The command

ab -n 100 -c 10 http://skjptpp.in/api/ will send 100 requests with 10 concurrent connections to the API endpoint to see how Nginx distributes the load.