Load balancing node applications:

**Clusters:** Cluster module lets you spin up multiple processes and connects the socket to your master process.(acts like big multithreaded server).

**PM2:** management tool for multi process nodejs setup, uses cluster under the hood.

**Testing load balancing using cluster:**

We are interested in how cpu is utilized between the processes to make sure that setup is actually balancing the load.

**Vagarant file**: Declarative file defines all software requirements, packages, os configurations, users and more.

**Chef:** Chef automates process of managing configurations , ensuring that every system you are responsible for configured correctly and consistently.

**Wrk:** load testing tool;  
**wrk -d30s -t12 -c5000** [**http://localhost:8080**](http://localhost:8080)terminology:

T12 :12 threads

-d30s 30 seconds

-c5000 concurrent requests 5000

It was seen nodejs is unable to distribute load evenly over all the processes/threads.

**Alternative:**

1. **NGINX**
2. **HAPROXY**
3. **VARNISH**

**Nginx:**

http server

built in load balancing

you can weight process so that you can assign more requests to one cluster.

Sticky sessions: users always hit the same instance after returning.

Timeouts

Horizontgal scaling

Ssl termination

**HAProxy:**

High availability proxy

Load balance lots of machines

Weighted processes, failover, horizontal scaling.

Default configuration of haproxy is to use single threaded architecture.

But you can set option to distribute load evenly across different nodes evenly:

Nbproc 2: spins up two haproxy processes.

**Varnish:**

Caching proxy means it handles all of the requests when it can and when it cant it’ll communicates with node server and forward the request and cahes the response for subsequest requests.