

Architecture!

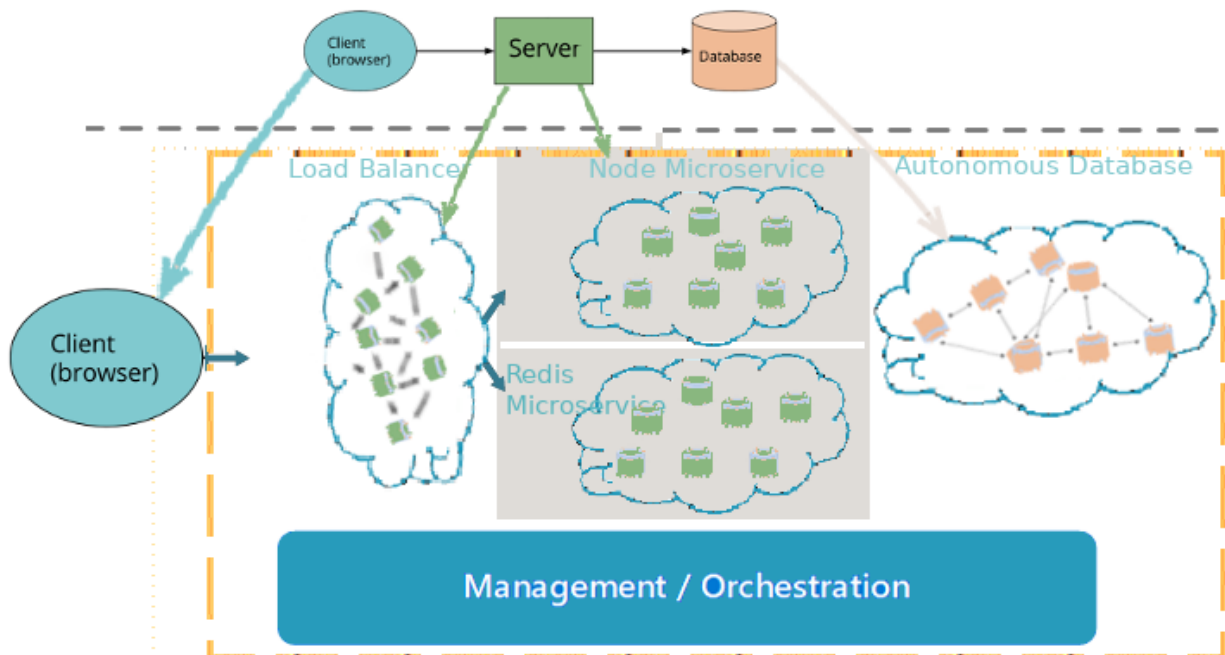
Table of contents

1. Deploy microservices to a Swarm cluster	1
1.1. Scalability	2
1.2. Availability	2
1.3. Security	2
1.4. Configuration	2

Architecture

We will be trying to create a swarm implementation that will allow communication between all of the members/nodes.

1. Deploy microservices to a Swarm cluster



- a microservice performs a simple task
 - communicates with clients or
 - other microservices
 - communication mechanisms such as REST API requests or Websocket etc
- Microservices can include any programming language you like
 - and with the orchestration tools they are easy to deploy and maintain



This architecture uses NodeJS and Redis microservices deployed as Docker containers

1.1. Scalability

You can scale your application by updating the number of replica nodes in the swarm cluster

```
...
  deploy:
    replicas: 15
    placement:
      max_replicas_per_node: 1
      constraints:
        - node.labels.region==regio1
...

```



Segmentation

Dynamic infrastructure: services can scale up and down without waiting for each other.

1.2. Availability

No single point of failure.

1.3. Security

1.4. Configuration