

Highly-scalable, data-intensive
and real-time apps

Udemy

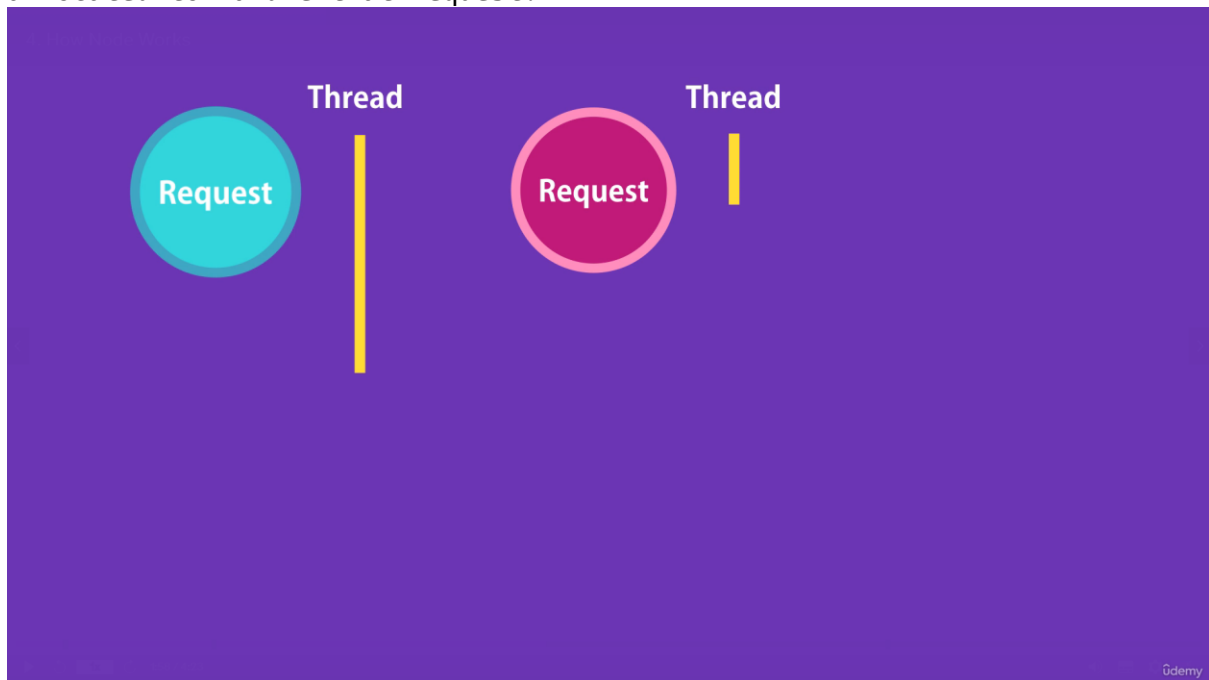
Non-blocking

ASYNCHRONOUS

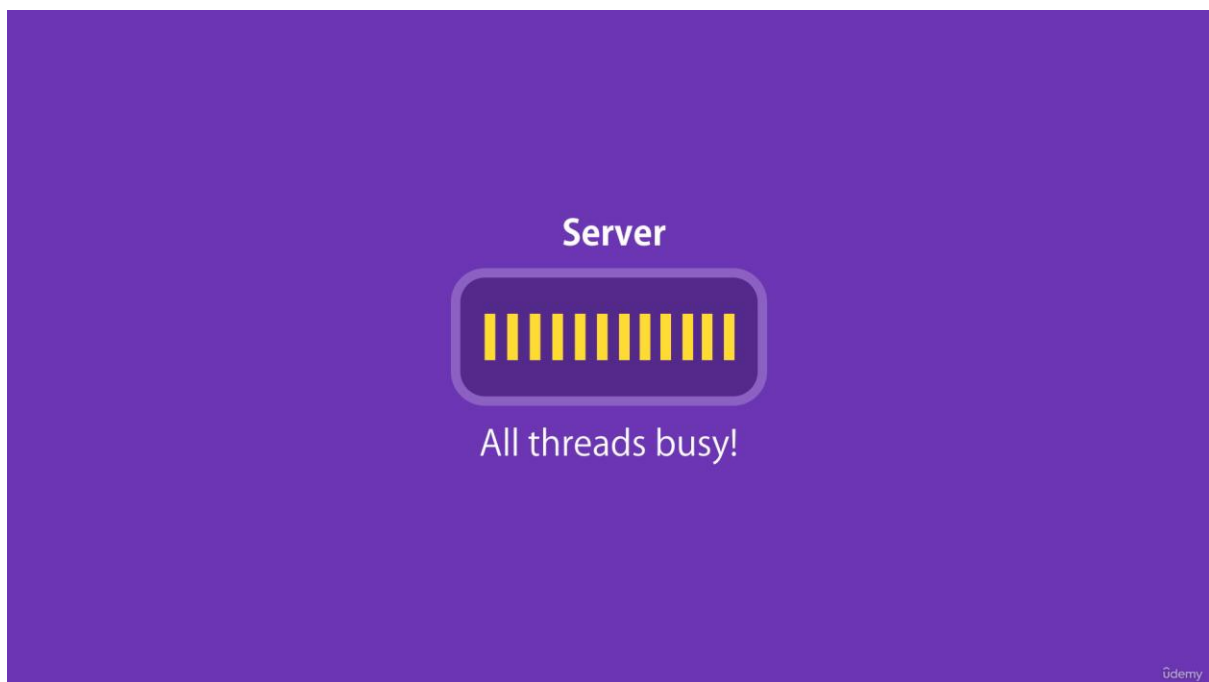
Udemy

Node is non - blocking asynchronous in nature.

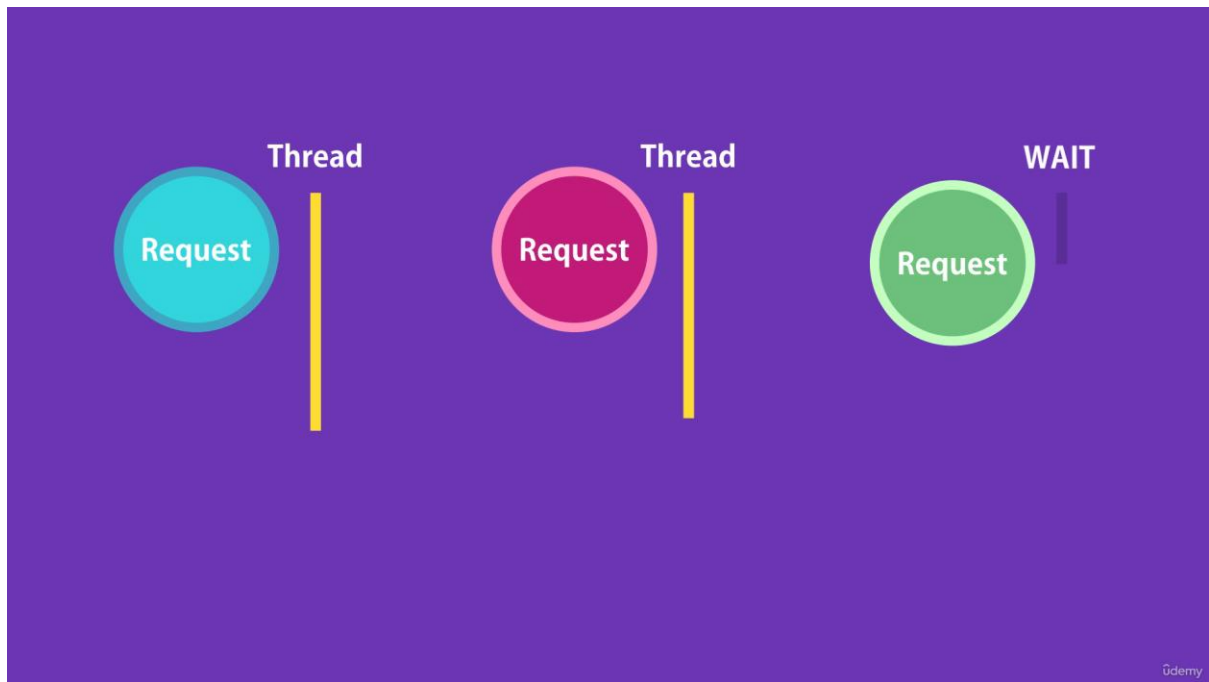
In the node environment, server is the thread which is allocated to handle the request. When we receive a request on the server a thread is allocated to handle that request.



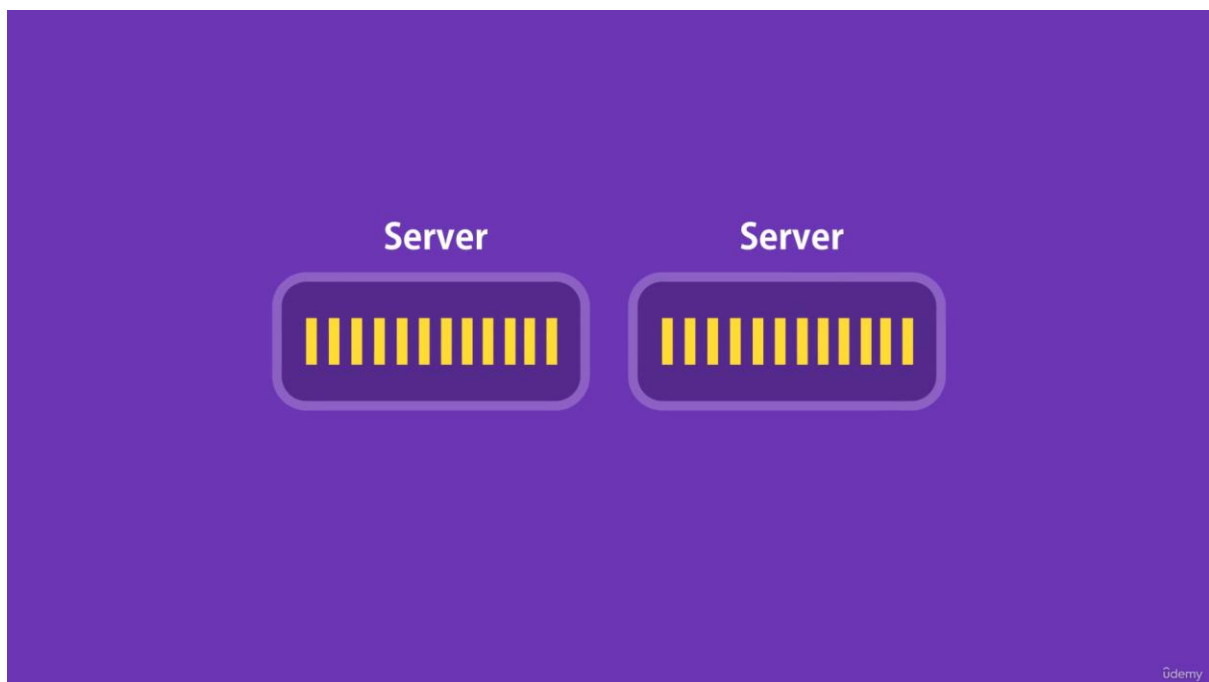
In synchronous nature a thread is allocated each time server receives the request.



Imagine we have large number of concurrent clients. At some point, we will run out of threads to serve these clients.



New clients have to wait until free threads are available.

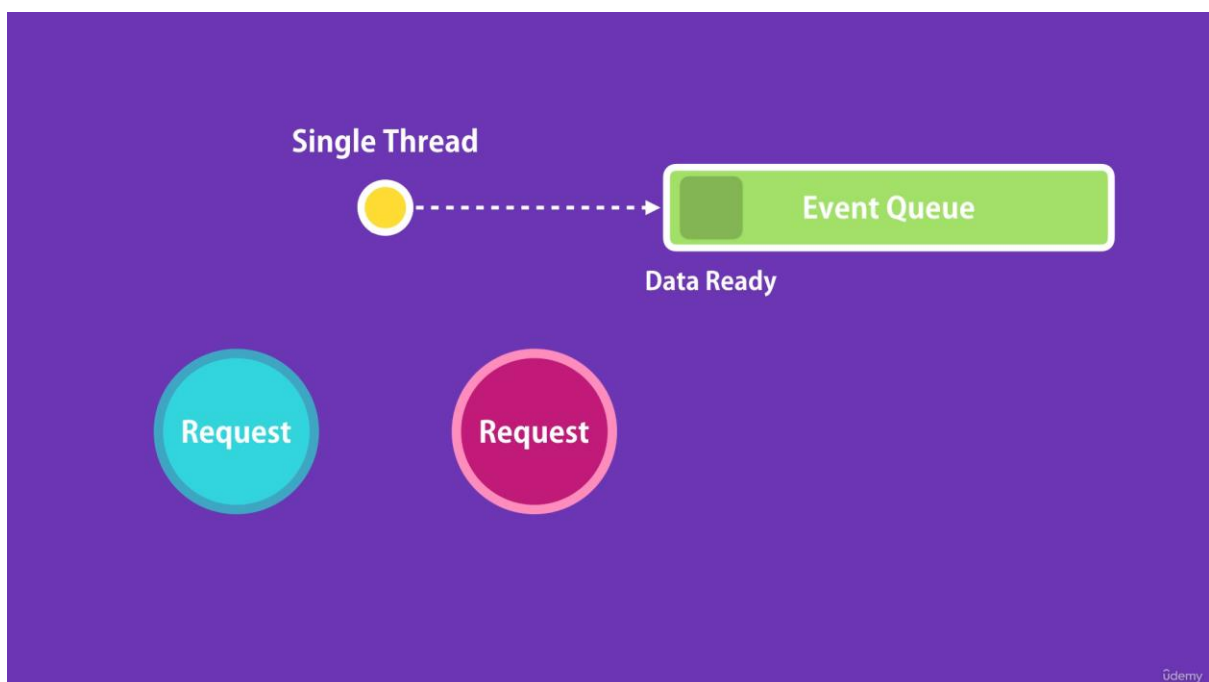


If we don't want these clients to wait, we need more hardware.

So, with synchronous behaviour, we are not utilizing the resources efficiently.

Node applications are **asynchronous** by default

Udemy



In node we have single thread to handle all requests. When a request arrives, that single thread is used to handle that request. If it needs to query a database, our thread does not need to wait for the database to return the data. While the database is executing the query, that thread will be used to serve another request. When the database is ready with the results, it puts the message in event queue. Node is continuously monitoring this queue in the background. When node finds an event in the event queue, it takes it out and process it.

Node is ideal for
I/O-intensive apps