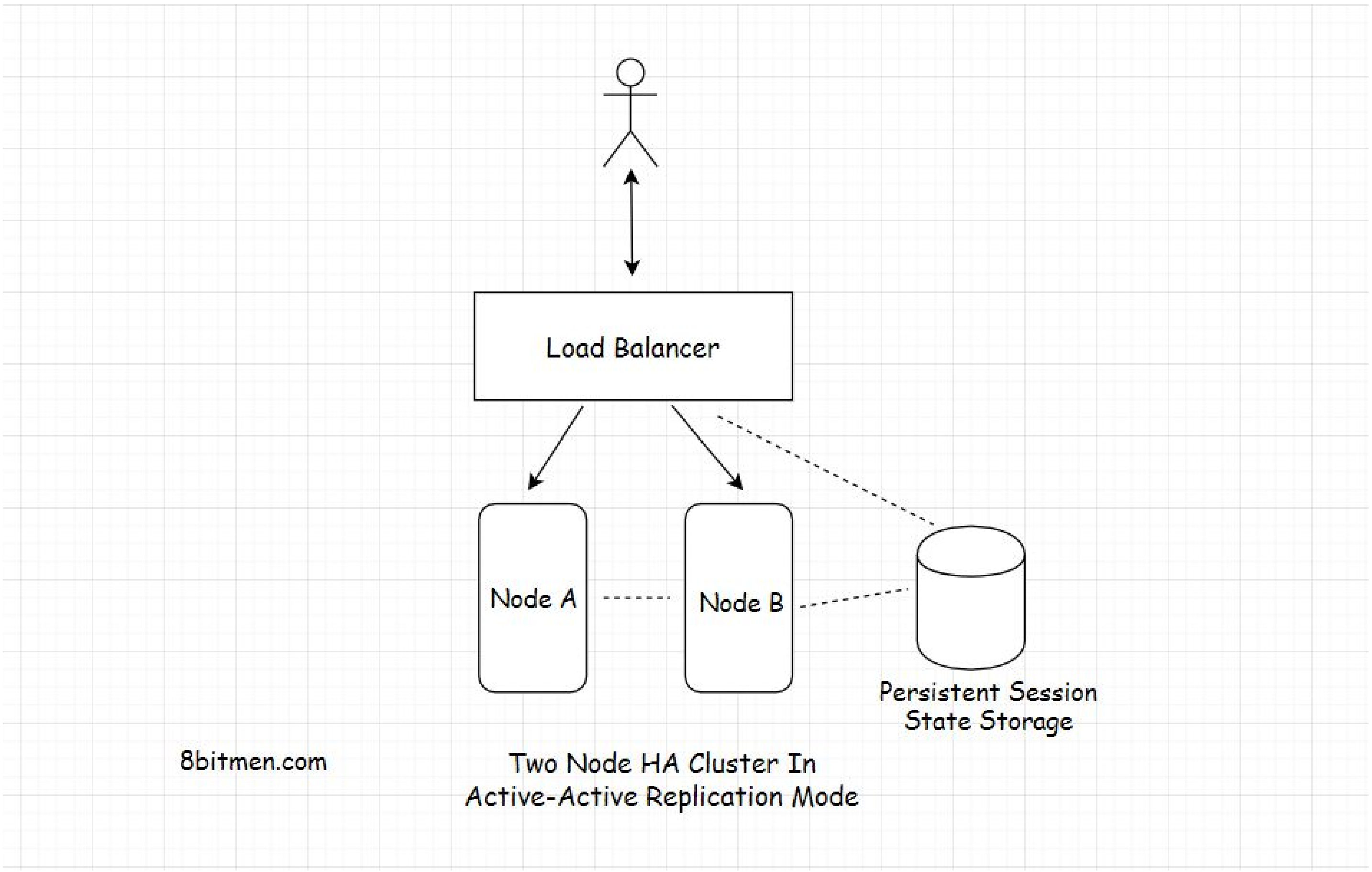
How Nodes Work Together in a Cluster? - Part 2

This lesson continues the conversation about communication between nodes in a cluster.

We'll cover the following



Load Balancer
Node A
Node B
Persistent Session State Storage
8bitmen.com
Two Node HA Cluster In Active-Active Replication Mode

Session replication

The *session state* is replicated across the nodes by the cluster when a request is processed by a certain node. An important thing to remember here is that despite multiple nodes running in the cluster, it still has to act as a single system.

All the nodes should have a common state at any point in time. The end-user interacting with the service doesn't care if the service is run by one server or a thousand servers. The cluster just has to maintain a consistent session state across all the nodes by replicating the sessions.

Session replication modes

Session replication can happen either *synchronously* or *asynchronously*. In the synchronous replication mode, the session state is replicated across all the nodes in the cluster before the response is returned to the user.

In the asynchronous replication mode, the session replication happens in a separate thread. It's decoupled from the response. The response is returned to the user even if the session replication fails.

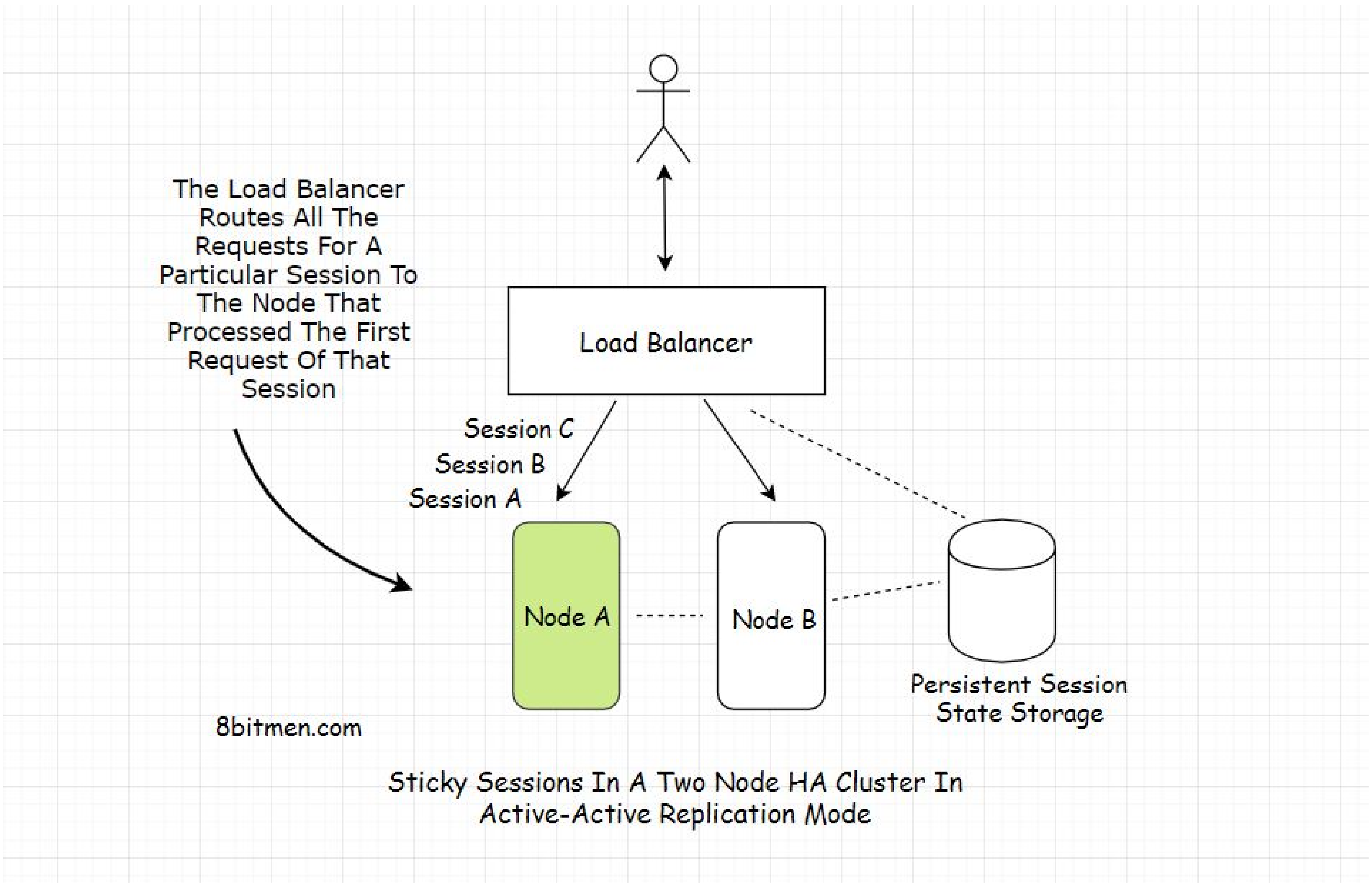
Choosing between *sync* and *async* session replication mode is a trade-off. *Synchronous replication* has a performance cost since the response has to wait for the session to be replicated across the nodes in the cluster. This adds latency to the response, but it also ensures a consistent session state across the nodes.

On the contrary, *async session replication* mode has no additional latency; the response is returned immediately. However, the consistent session state across the cluster is not guaranteed.

There is one more concept involved in this, the concept of *sticky sessions*.

Sticky sessions

Sticky sessions mean that the load balancer will route all the requests for a particular session to the server that processed the first request of that session.


The Load Balancer Routes All The Requests For A Particular Session To The Node That Processed The First Request Of That Session
Load Balancer
Session C
Session B
Session A
Node A
Node B
Persistent Session State Storage
8bitmen.com
Sticky Sessions In A Two Node HA Cluster In Active-Active Replication Mode

Synchronous session replication works with or without sticky sessions, but they are important when we pick the *asynchronous session replication* approach. Since the consistent session state across the cluster is not guaranteed when using async session replication, another request for a certain session landing on a different node can cause inconsistency issues. This is why all the requests for a particular session are routed to the original server.

However what if the original node processing the requests for a certain session goes down when using the async replication strategy?

In this case, the load balancer has to route the request to a different node. The new node has to contain the latest session state to avert any kind of inconsistency issues. This is an important thing to keep in mind when implementing the asynchronous session replication approach.

One downside of using sticky sessions is that it can add unwanted load to a particular server. The load balancer will keep on routing requests to it even if it is already piled up with requests.

Back to the node coordination scenario

Let’s jump back to the scenario we were discussing. *Node A* goes down; *Node B* receives the request; it reads *Node A’s* session info from the persistent session storage, processes the request, and returns the response.

Now, when *Node A* comes back online, it gets in touch with the cluster, invalidates the session* US_1* it created, and gets ready to receive the new requests.

This is how nodes communicate with each other and process requests in a cluster.

I’ve tried to present a very simple example of this so that you can get a bird’s eye view of how the nodes work in conjunction and so on. In reality, things are much more complicated, especially with cloud computing/distributed systems.

In the next lesson, let’s talk about how nodes reach a consensus in a cluster.