**Load Balancer**

* **It is used to distribute incoming request across multiple servers.**
* **If server is down that server is removed from loadbalancer. It means loadbalancer will check that server is up or down internally.**
* **Add and remove server in the network as per demand.**
* **Apache HTTP server is behave like load balancer.**

**How to Connect Apache HTTP Server with Tomcat**

1. First we have to download apache http server and check the port 80 into C:\Apache24\conf\httpd.conf file.
2. Run the httpd.exe file from C:\Apache24\bin.then apache http server will start.
3. Configure the multiple node(tomcat server).
4. Now to communicate between apache http server and tomcat we have to use mod\_jk communicator.
5. We have download mod\_jk.so file and we have to keep into C:\Apache24\modules folder. Then we have to load mod\_jk.so we have to create one configuration file mod\_jk.conf file and include into C:\Apache24\conf\httpd.conf file.
6. Create one conf/workers.properties file where we have to write all tomcat node information.
7. Also we have to change server.xml file into tomacat that is connector port and engine.
8. <http://www.tutorialbyexample.com/2015/02>

**Multiple way to setup the Load Balancer**

1. **Round Robin Fashion**

* **Where each server get the new request everytime. If we have three server then 1st request will go to 1st server , 2nd request will go to second server and third request will go to third server.**

1. **Least Connections**

* **Request sent to least used server. This is costlier due to some computing to get that which server have least no of resources.**

1. **IP Hash**

* **Request sent to the sever based on client IP.**

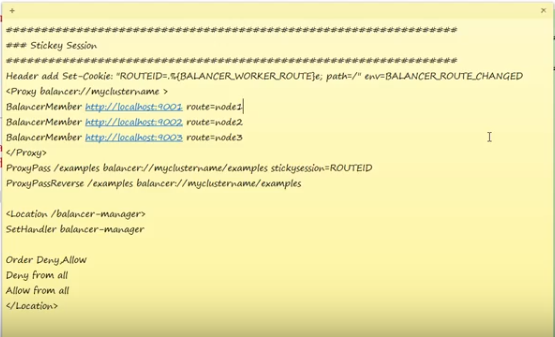
**Non Sticky Session**

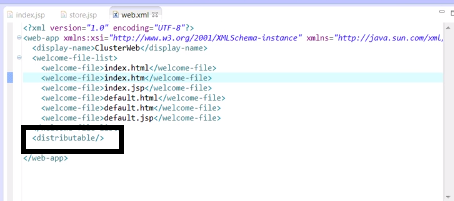
* **Seesion where we can not sure that user request will go to which node.**
* **For that we have to configure into httpd.conf file.**

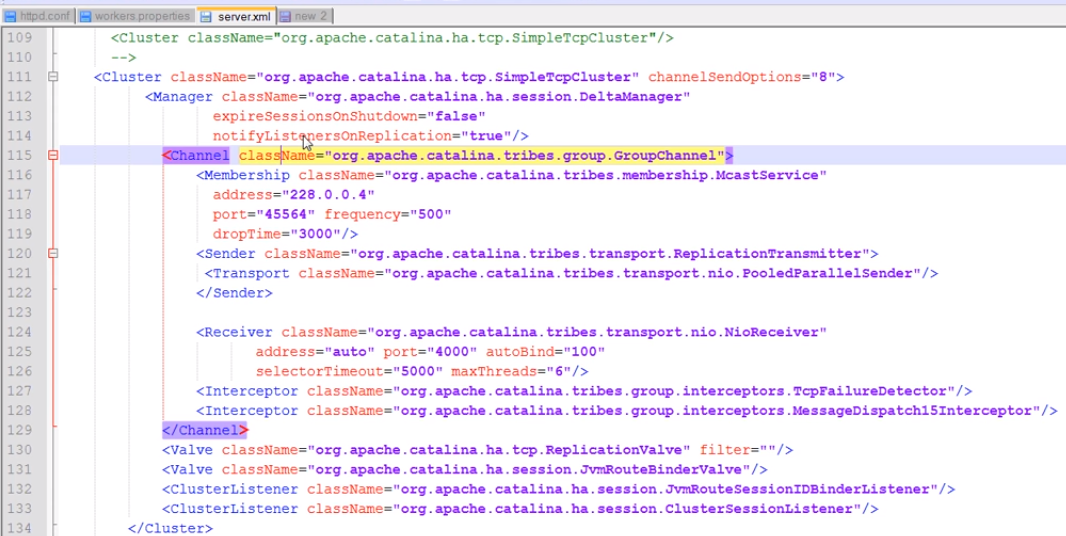
****

**Sticky Session with Replication**

* **Session where we can sure that request will go to one node when request is coming from same client and create some sessionid. If that node is down then remaining request will go to another node.**
* **This is called session persistent.**
* **Node that receives the remaining requests has a copy of the user session then user keeps on his session. So user continue to browse his web app without being disconnected.**

****

****

****

Data Replication in Distributed System

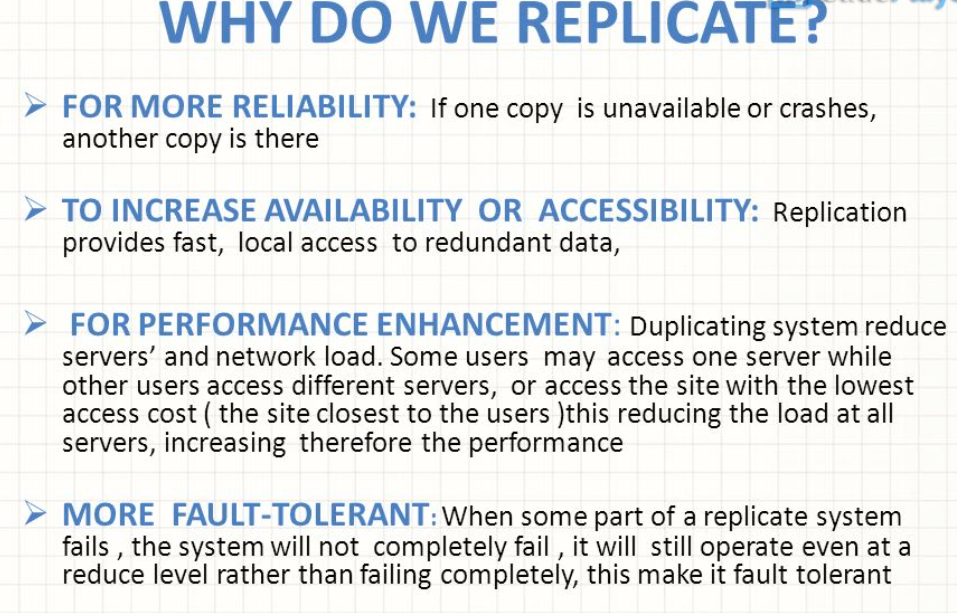
**1) Increased Availability**

**2) Faster Query Evaluation**

**3) Updates are challenging**

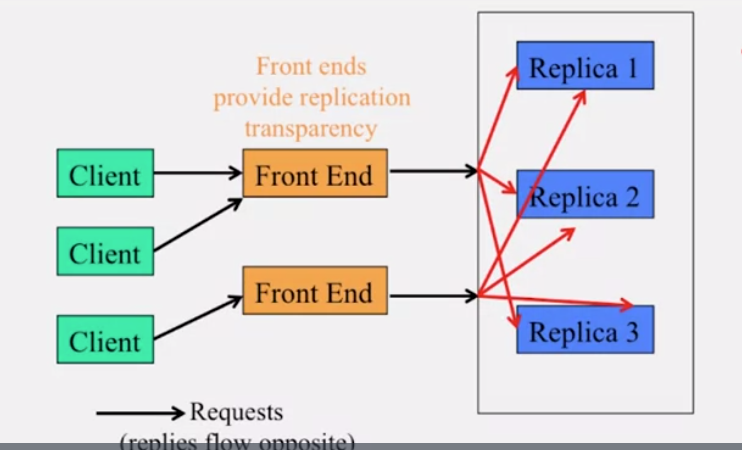
**->This is good for read only query.**

**->May cause problem into update queries, transaction and concurrency.**

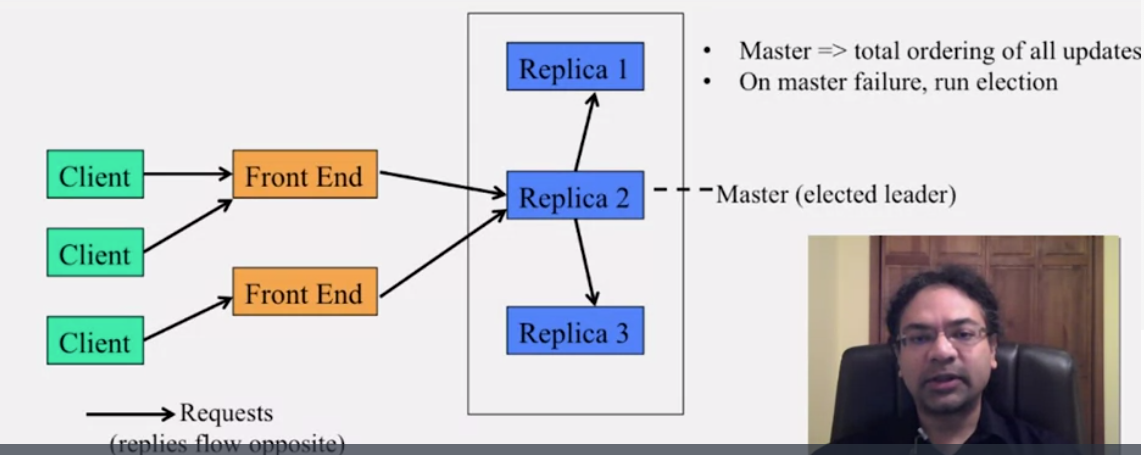
****

**Two ways of Replicas**

**1)Active : Treats all replicas identically. Here read/write will happen at entire replicas.**

****

**2)Passive : Uses a primary replicas(master). Here all writes will happen on master but read will happen at any replicas.If master failure then will run leader run protocol and elect master replica.**

****