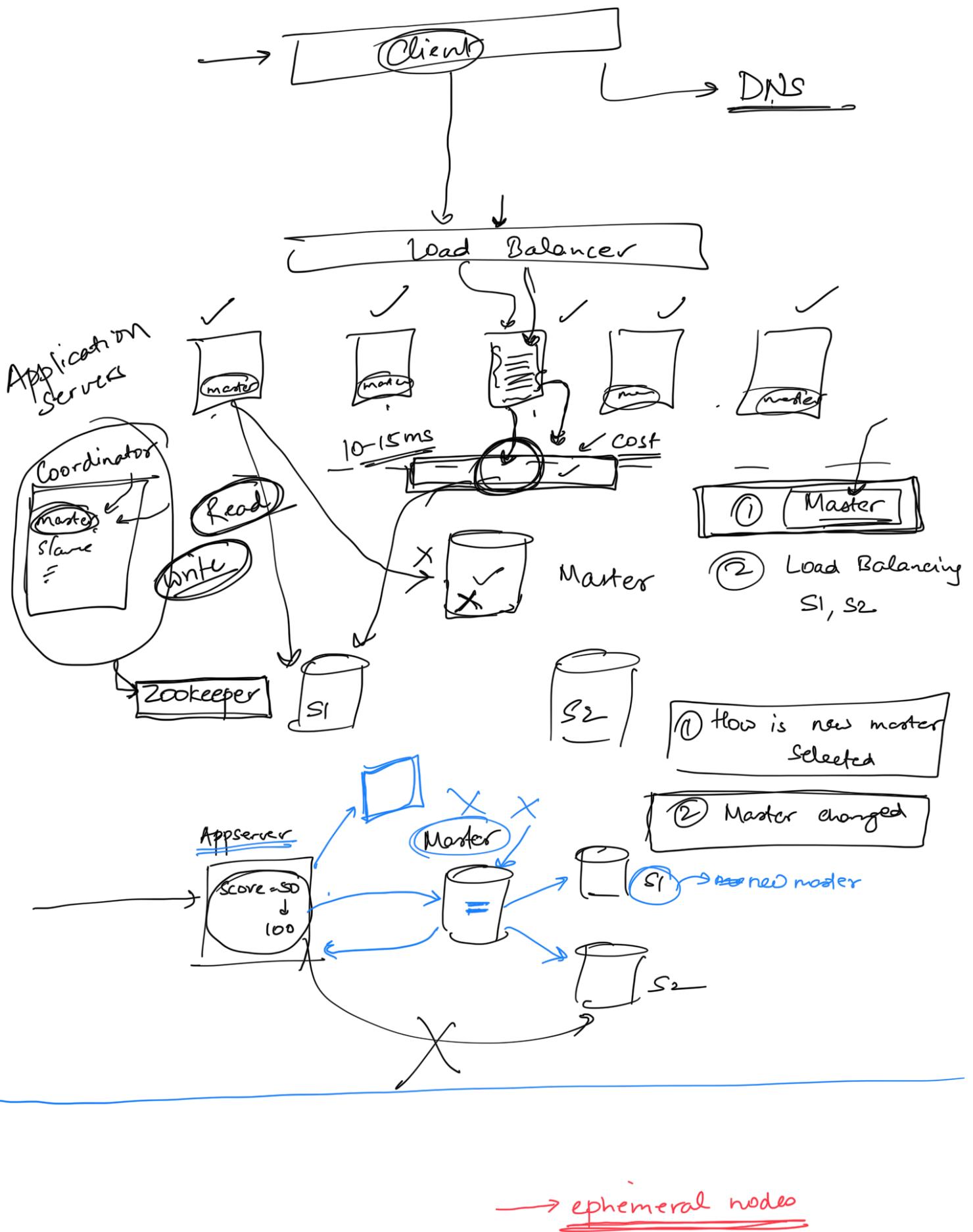


ZOOKEEPER AND KAFKA

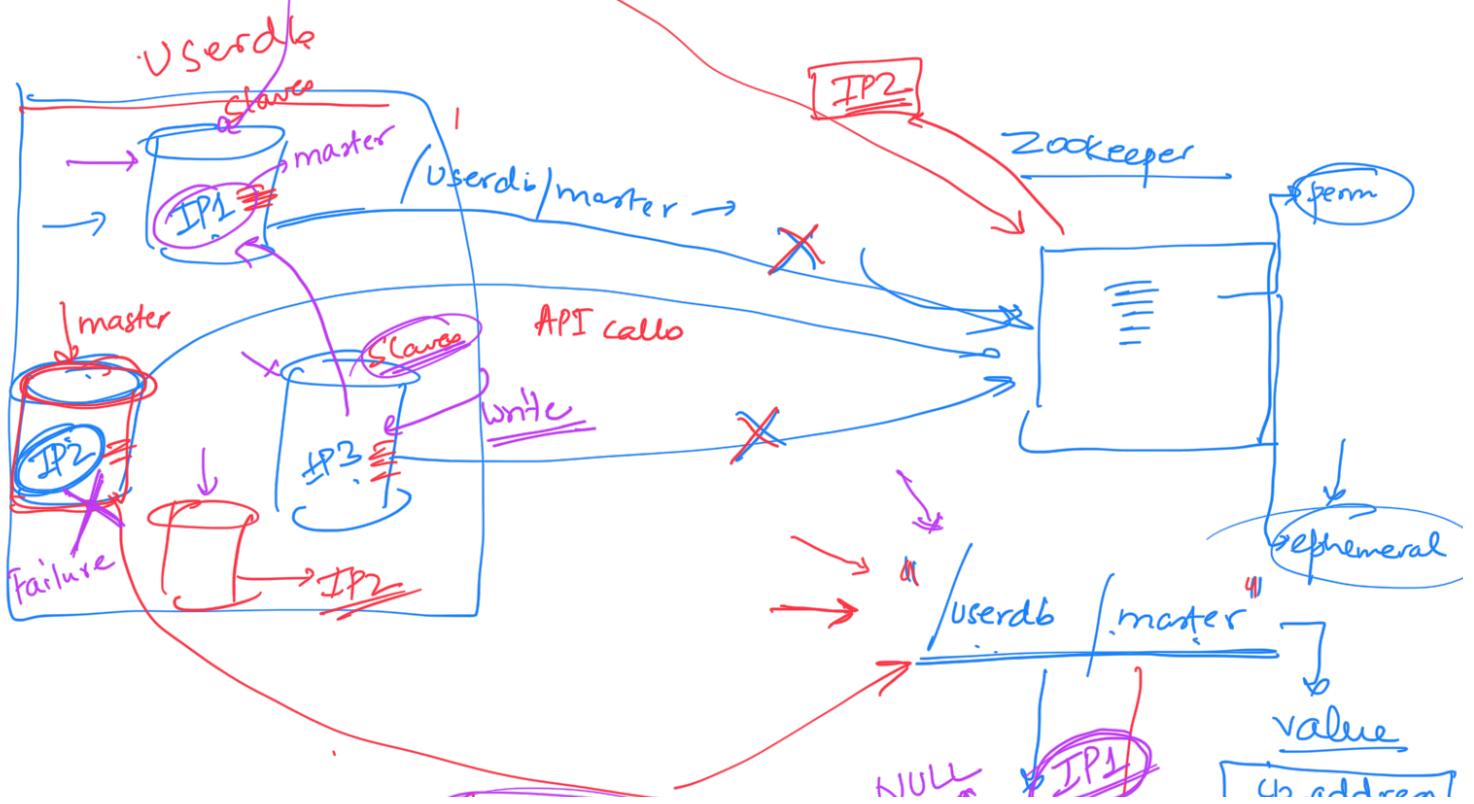
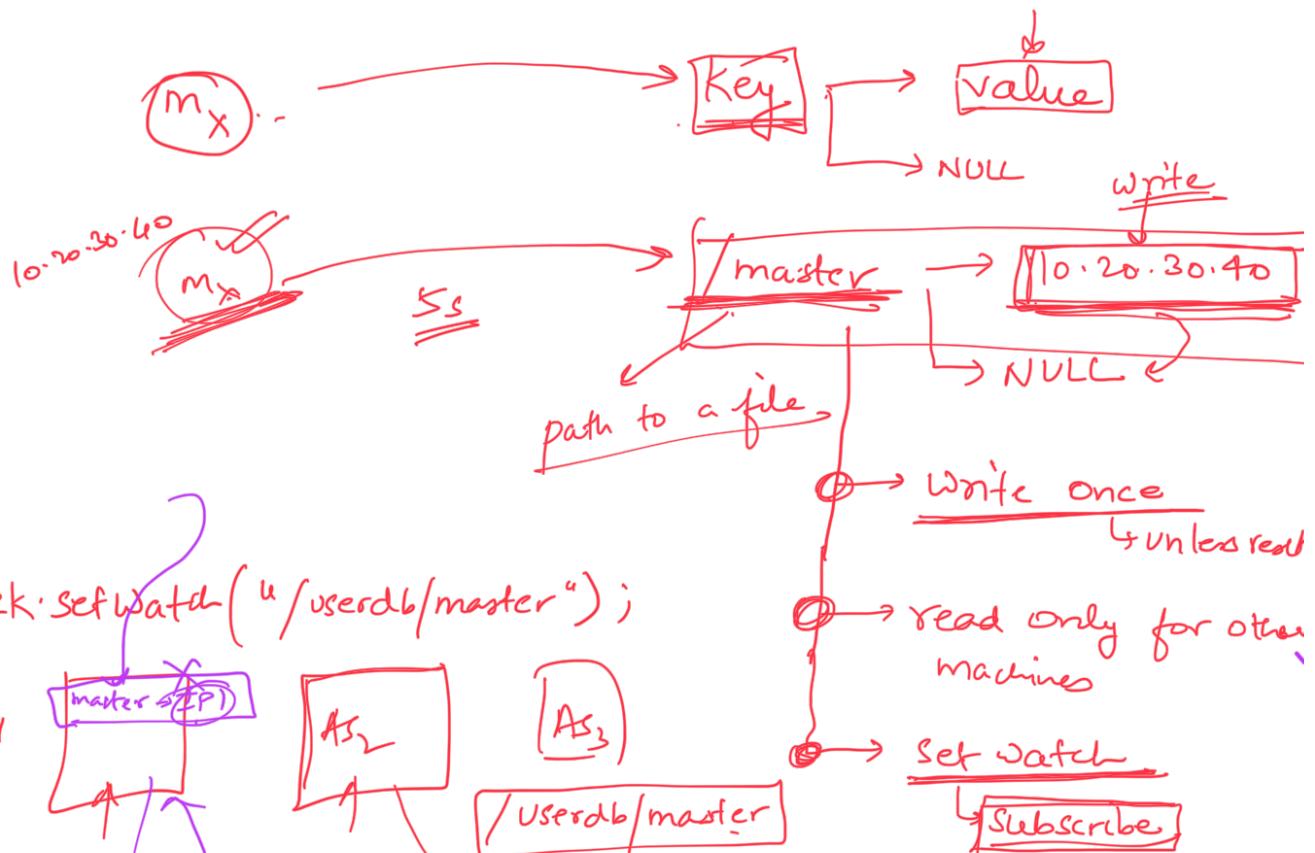


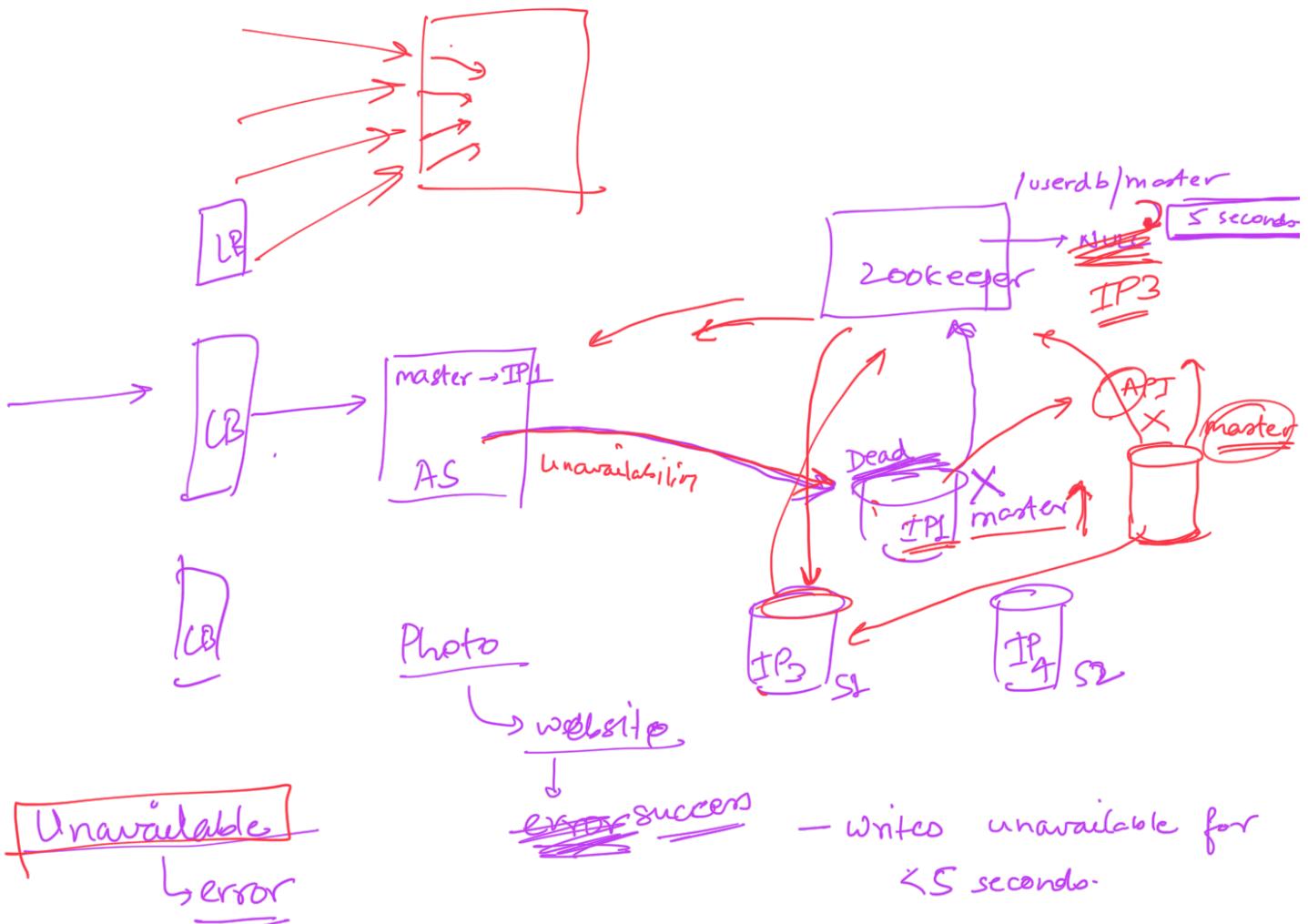
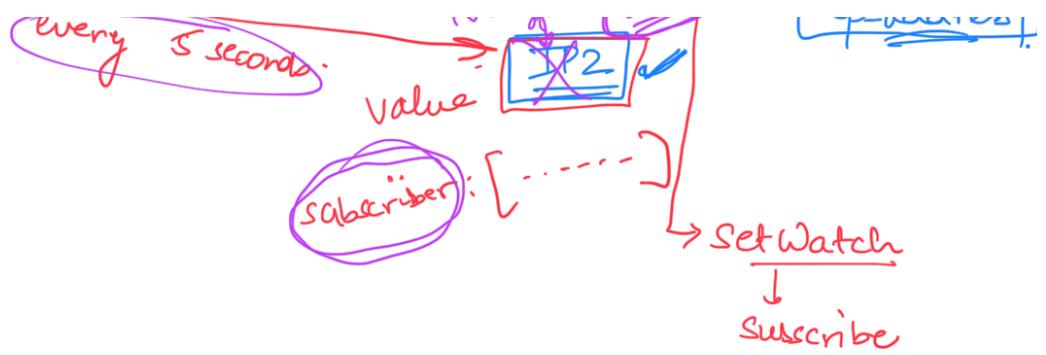
home/User/.../file1.txt

→ permanent

~~file~~ / machine / network = key

↑
key
↑
value





Data loss

Success

no data

① zk → collection of machines

→ ② zookeeper usecases

master

slave

write

When you start ...

Toy becoming master

1 Slavey

↳ who is master

before taking
requests

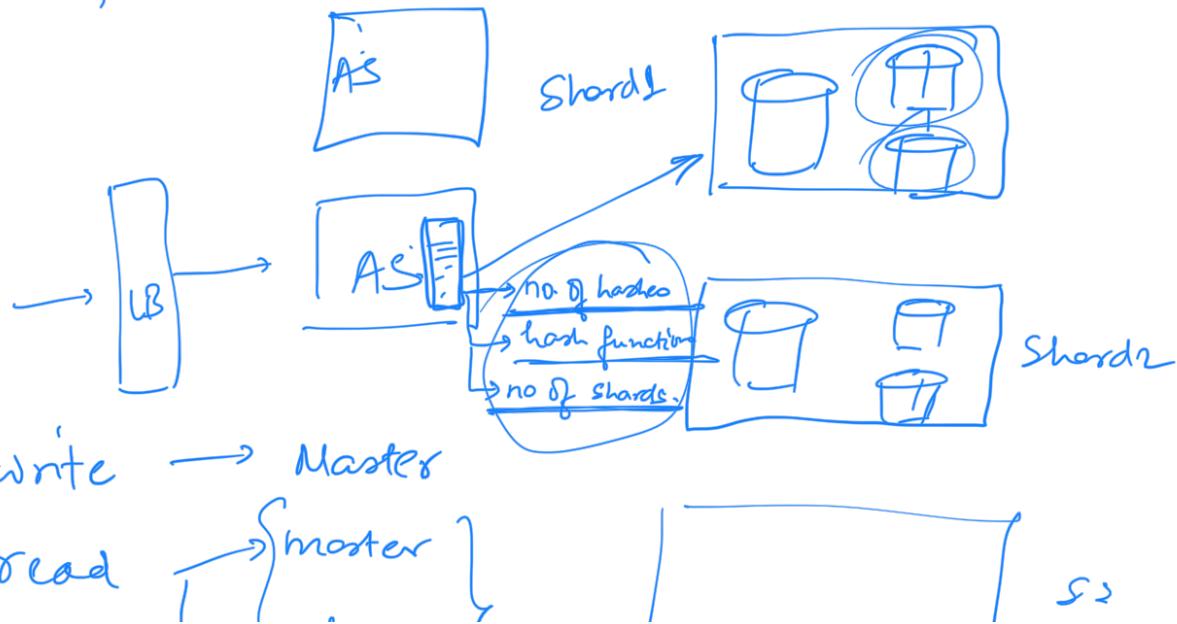
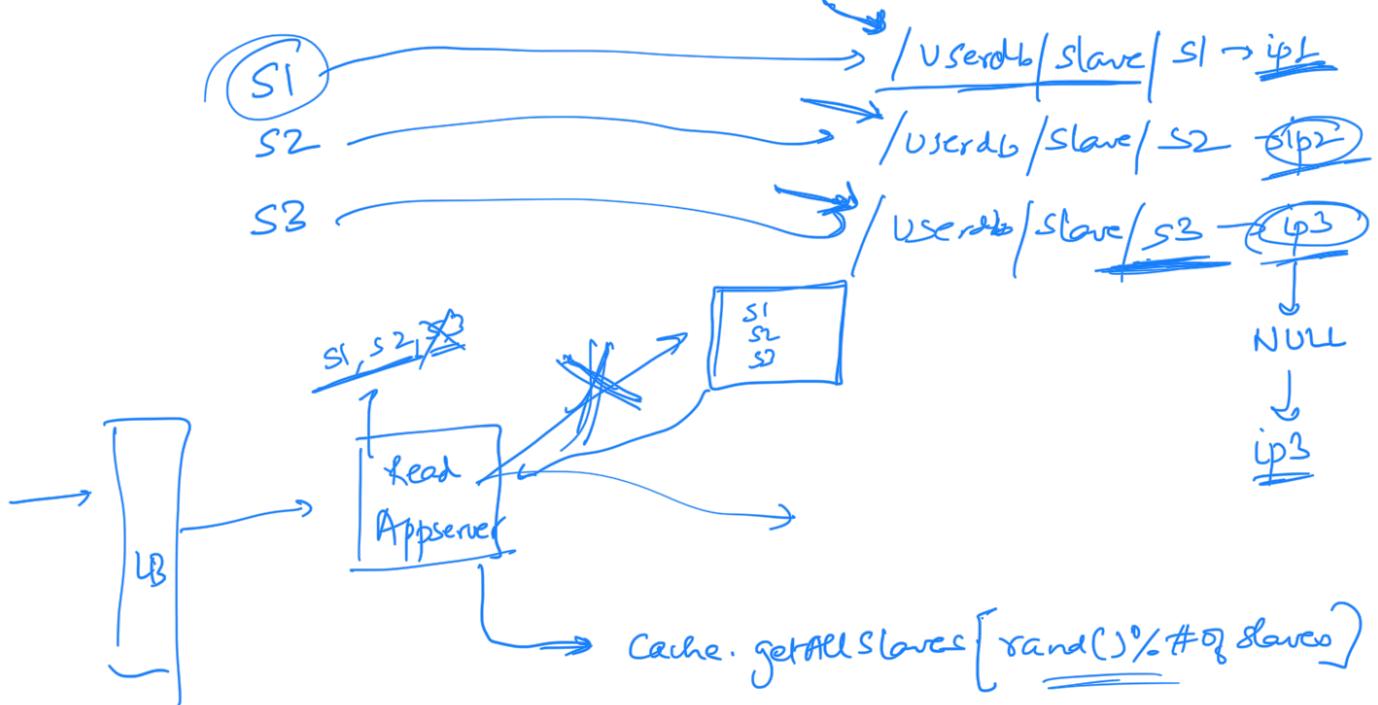
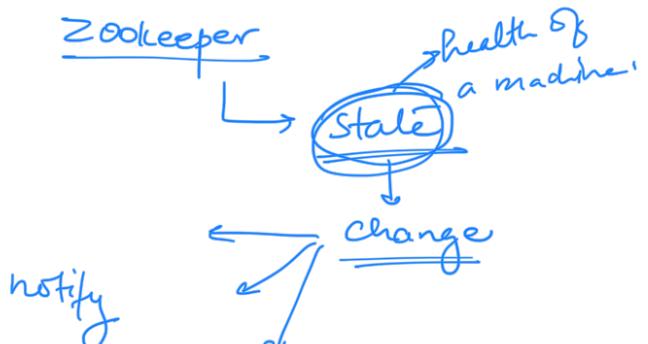
→ Read from ~~2~~
Value of master
→ Set Watch

Usecases

- ① master selection +
master value

②

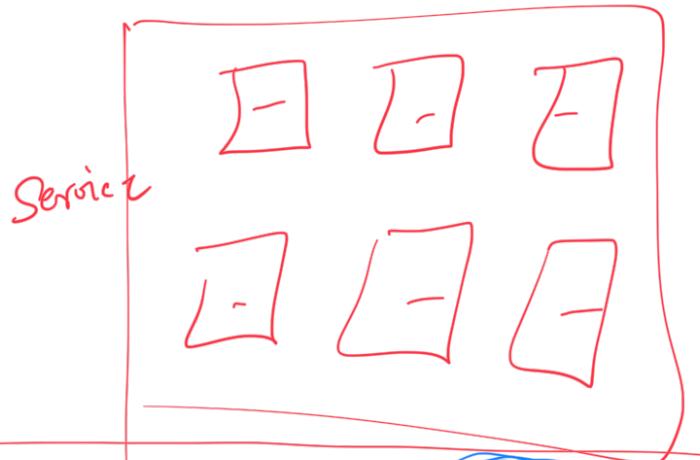
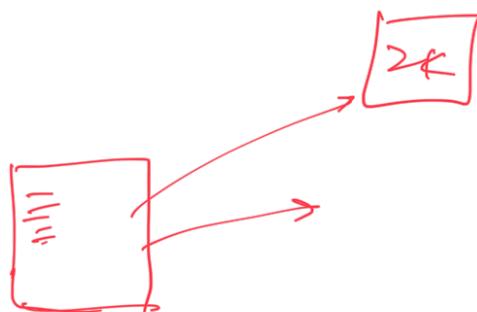
Zookeeper



\uparrow slaves \downarrow

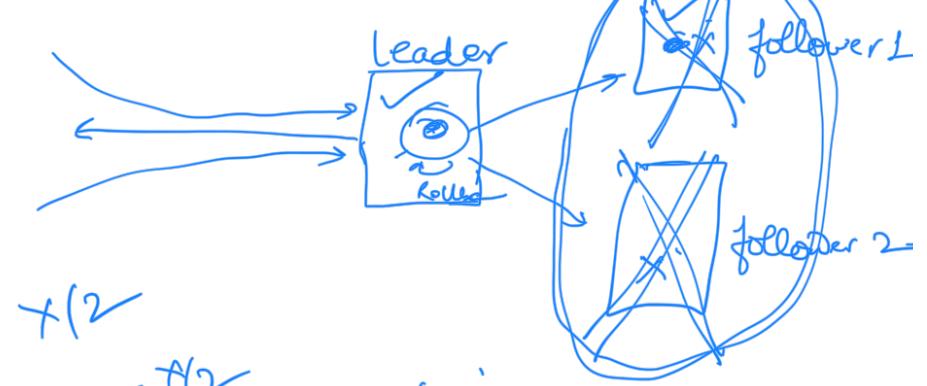
import HBaseClient

HBaseClient::init(zk_url, cluster_name, ...)

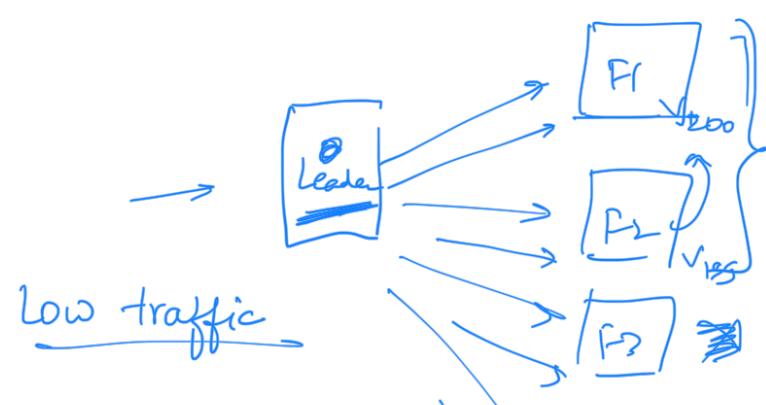


3

$w \geq \lceil \frac{2}{3} \rceil$



Read \rightarrow leader
majority of machines

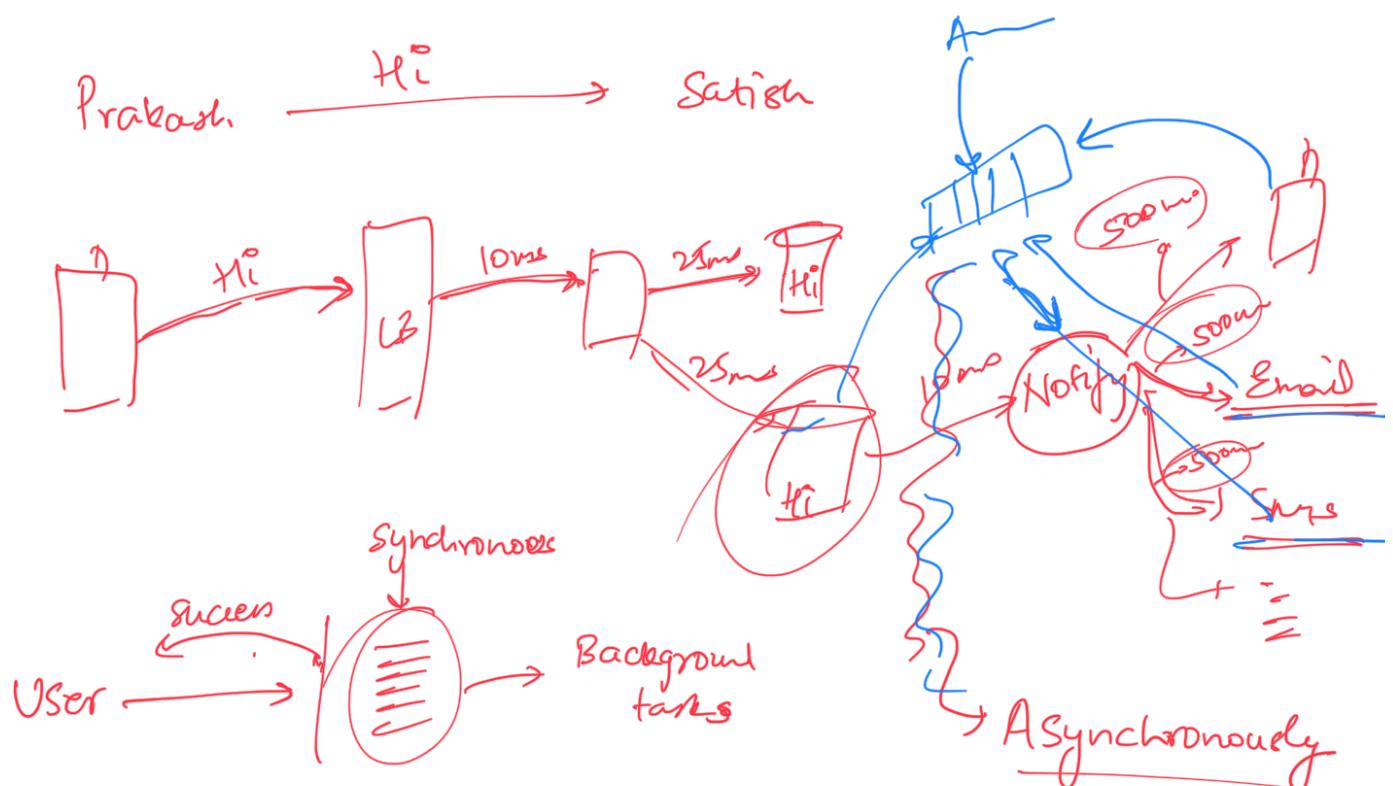


write <100 in a day ↗ RF

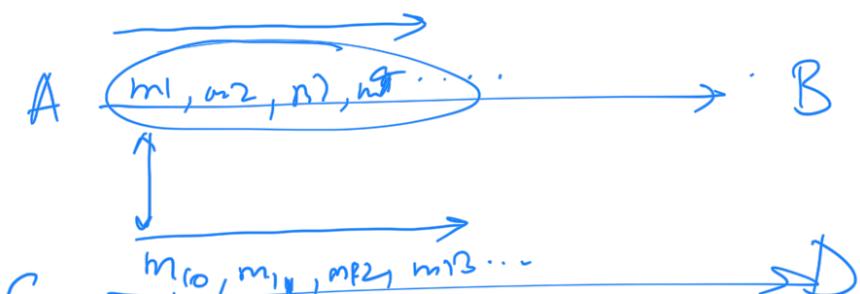
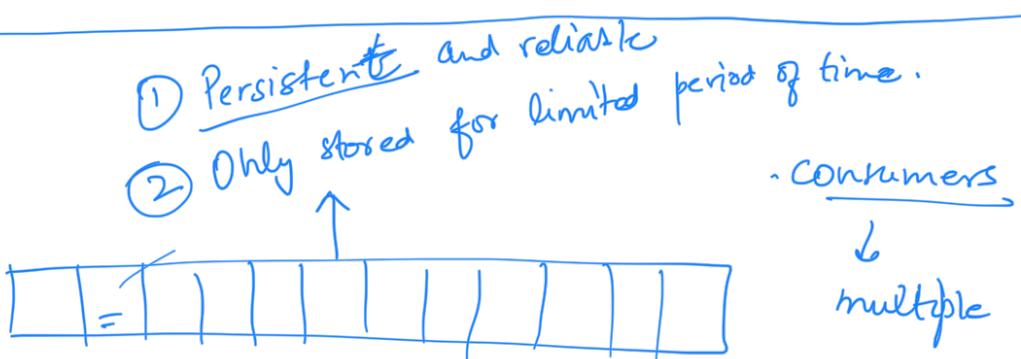
Read

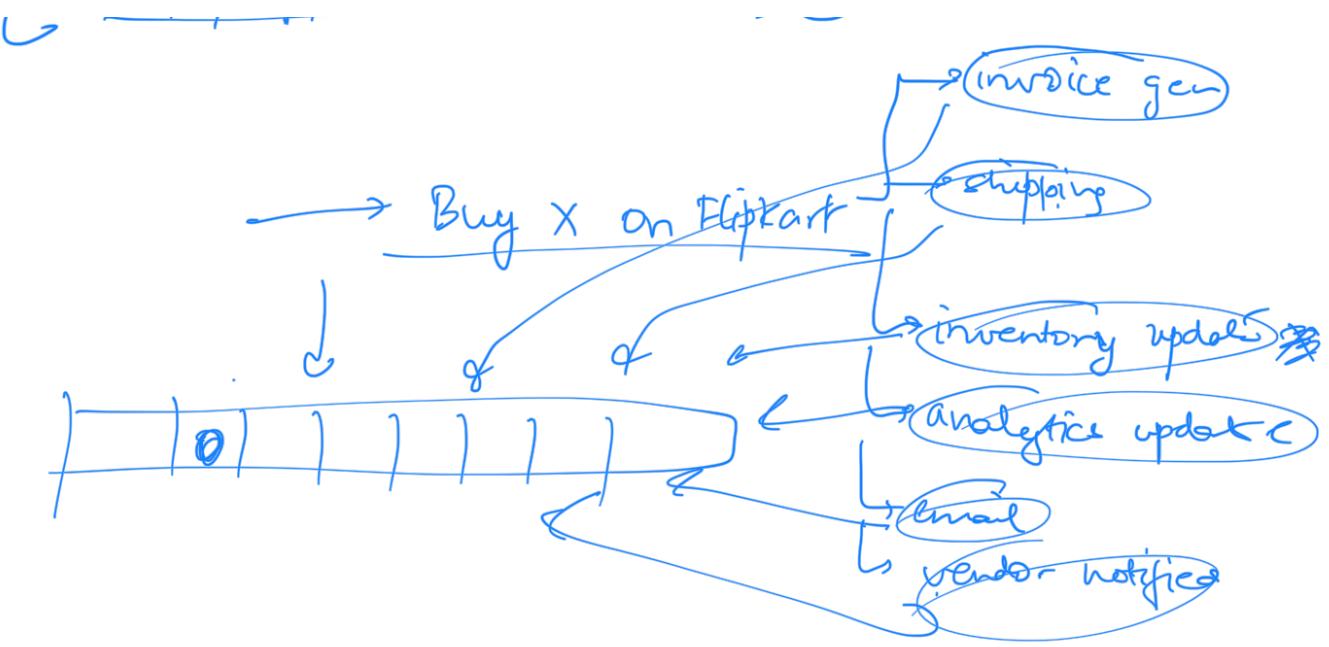
1000

KAFKA / MESSAGING QUEUE



Producers
multiple





Design messaging queues: Kafka

- ① Durable
- ② Easily scalable / throughput
- ③ consumers → parallel/multiple msg.
- ~~④ low latency~~
- ⑤ Retention of msgs.
Consumer → msg → index