Understanding Distributed Consensus

Laura Frank

Director of Engineering, CloudBees

CODESHIP

cloudbees.


CODESHIP


cloudbees

Agenda

- What is distributed consensus?
- What is Raft and where is it used?
  - Understanding quorum
  - Leader election
  - Log replication
- Failures and Recovery

Disclaimer: Production best practices are not necessarily my goal here

What is Distributed Consensus?

Distributed consensus is a fancy way to express the idea of **getting more than one party to agree** on something

This used to be easier

- We use tools like Kubernetes because we want to build a fault tolerant system, which requires redundancy and multiple resources
- Taking action on the system means that every managing entity in the system has to agree about the action

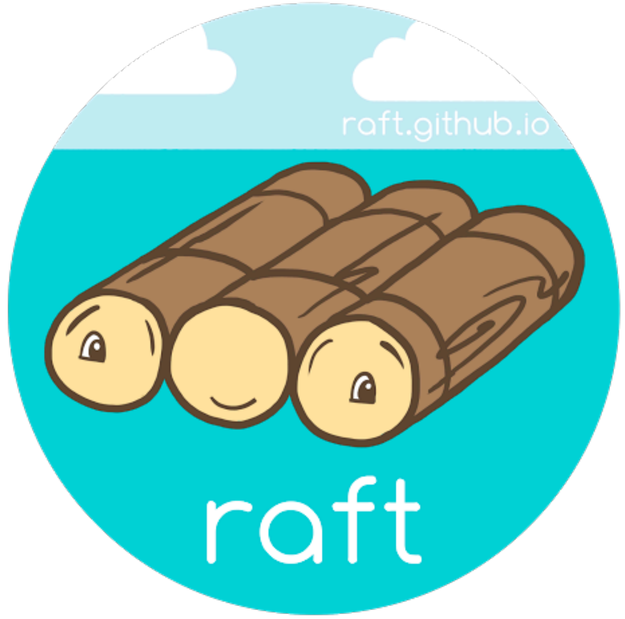
In a distributed model, that log entry can only “become truth” once the managers agree on the new value

THE PROBLEM WITH CLUSTERS

having N machines agree on a single state is hard

What is Raft?

- Raft is an algorithm to manage consensus-based systems (like container orchestrators)
- It is designed to be easy to understand
- It also has a cute logo



"I think I'll just write my own distributed consensus algorithm."

– NO SENSIBLE PERSON (EXCEPT THE AUTHORS OF RAFT)

Raft is used in a lot of places

Orchestration systems typically use a key/value store backed by a consensus algorithm

- Kubernetes —> etcd, and by extension, every system that uses etcd
- Docker (swarm mode)
- Nomad

- Zookeeper uses Zookeeper Atomic Broadcast (ZAB), which is similar to Raft

Raft is responsible for...

Log replication

Leader election

Safety (won't talk about this much today)

Being easier to understand

Quorum

Quorum

The minimum number of votes needed to perform an operation.

Without quorum, your system can't do work.

Quorum

This doesn't just mean that the nodes have t be online, but that they are in agreement!

It can be that all nodes are online but not in agreement. Raft will have to do reconciliation in the case of split brain, etc.

CODESHIP

cloudbees.

Quorum is satisfied with **over** 50% of agreement

$$(N/2) + 1$$

Managers	Quorum
1	1
2	2
3	2
4	3
5	3
6	4
7	4

Quorum is satisfied with **over** 50% of agreement

$$(N/2) + 1$$

Managers	Quorum	Fault Tolerance
1	1	0
2	2	0
3	2	1
4	3	1
5	3	2
6	4	2
7	4	3

Having two managers instead of one actually **doubles** your chances of losing quorum.

Quorum in Multiple AZs

Pay attention to datacenter topology when placing managers.

Manager Nodes	Distribution
3	1-1-1
5	1-2-2
7	3-2-2
9	3-3-3

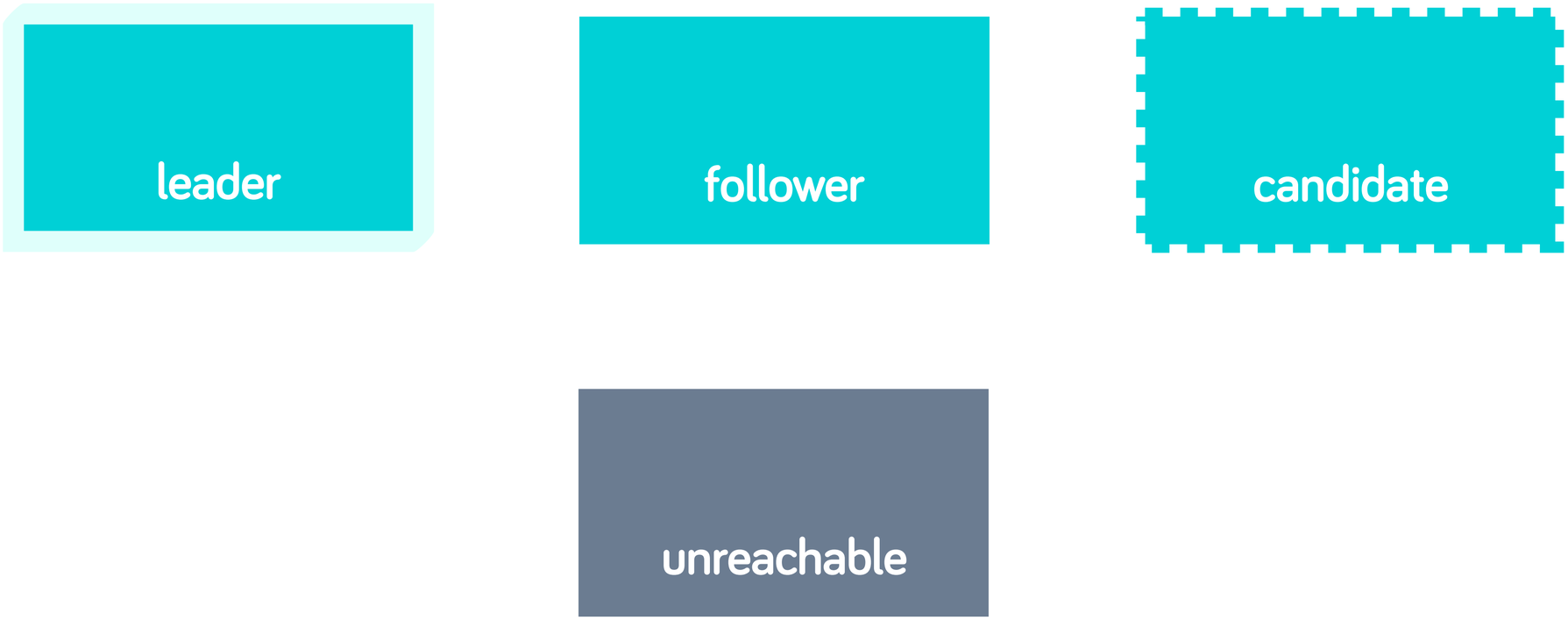
magically works with many turnkey cloud tools

CODESHIP

cloudbees

Leader Election

Raft consensus group



demo.consensus.group

Log Replication

The log is the source of truth for your application.

In the context of distributed computing (and this talk),
a log is an append-only, time-based record of data.

This log is for computers, not humans.

demo.consensus.group

?

In distributed computing, it's essential
that you understand log replication.

bit.ly/logging-post

Failure Recovery

Failure Recovery

Hard failure: datacenter on fire?

Restore from a backup

You have a backup, right?

ProTip: Some tools manage this for you

Generally speaking:

- The log snapshots live in `/snap`
- The WAL lives in `/wal`
- In etcd, there is also a `--wal-dir` flag for you to define a disk/dir of your choice

Restore from a backup

```
etcd \
        --data-dir=%backup_data_dir% \
        [--wal-dir=%backup_wal_dir%] \
        --force-new-cluster \
        ...
```

Restore from a backup

On a healthy manager (could be new) stop Docker

```
sudo rm -rf /var/lib/docker/swarm
```

Copy backup to `/var/lib/docker/swarm`

Start Docker

```
docker swarm init --force-new-cluster
```

Node IP addresses need updating

- Restoring from a backup == old IP addresses of member nodes
- etcd -> easy to update a member IP
- swarm -> not easy

Failure Recovery

Soft failure: losing quorum

Regain quorum

- Losing quorum is only about management
- Your app will continue to run
- No new actions or changes to state can be performed, including recovery, node promotion or demotion, or adding new tasks

What does this mean for your app?

Regain quorum

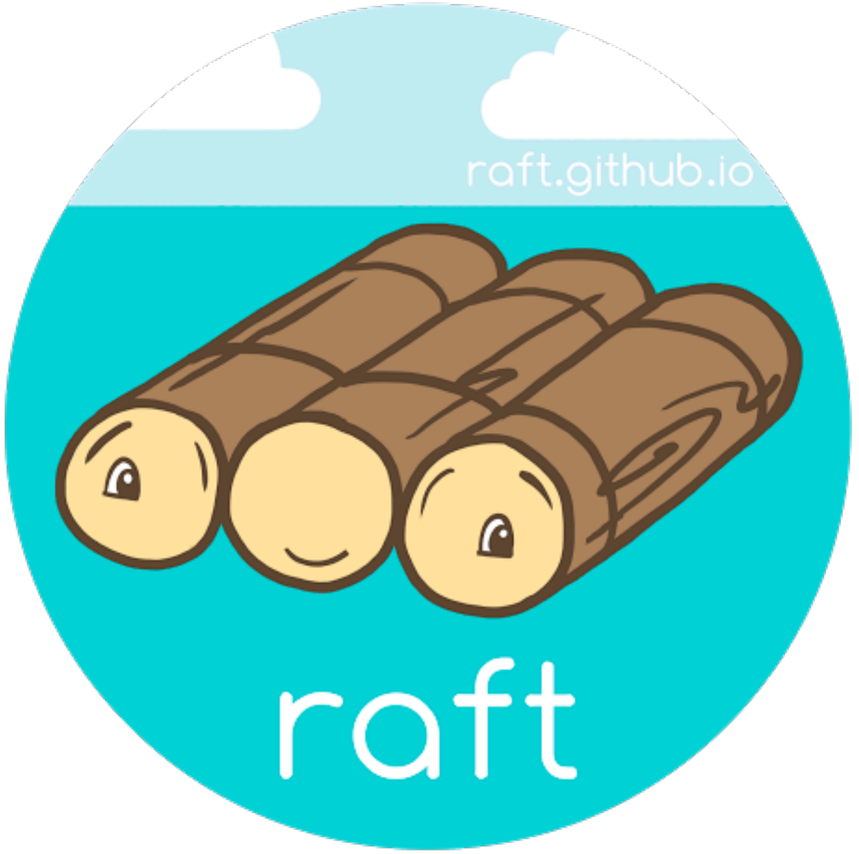
- Bring the downed nodes back online (derp)
- Or on a healthy node, initiate a new cluster via `--force-new-cluster`
  (swarm and etcd use the same flag)

  This will create a new cluster with one healthy manager
  - You need to promote new managers

Regain quorum

- You can’t just add a new healthy manager to the cluster

  Why? That would require consensus!


raft.github.io
raft

Thanks!

Laura Frank

Director of Engineering, CloudBees

CODESHIP

cloudbees