What You Should Know About etcd v3
What is etcd?
etcd.js

A modern, fast, React-like JS framework
etcd.js

A modern, fast, React-like JS framework

JK
ETCD KEY VALUE STORE

Fully Replicated, Highly Available, Consistent
How does etcd work?

- **Raft** consensus algorithm
  - Using a replicated log to model a state machine
  - "In Search of an Understandable Consensus Algorithm" (Ongaro, 2014)

- Three key concepts
  - *Leaders*
  - *Elections*
  - *Terms*
How does etcd work?

- The cluster **elects a leader** for every given **term**
- All log appends (--- state machine changes) are decided by that leader and propagated to followers
- Much much more at [raft.github.io/](https://raft.github.io/)
THE HEART OF CLOUD NATIVE

Kubernetes, Cloud Foundry Diego, Canal, many others
14,500 Stars on Github
380 contributors
CoreOS, Google, Red Hat, EMC, Cisco, Huawei, Baidu, Alibaba...
History of etcd

- **2013.8** Alpha release (v0.x)
- **2015.2** Stable release (v2.0+)
  - stable replication engine (new Raft implementation)
  - stable v2 API
- **2016.6** (v3.0+)
  - efficient, powerful API
  - highly scalable backend
Why talk about it now?
Self-hosted, Scale, and Federation with Kubernetes v1.4 and Beyond by Brandon Philips, CoreOS, Inc.
etcd v3.0 - "Scaling etcd to thousands of nodes"

- Efficient transport via gRPC and HTTP/2
- New powerful API based on k8s use-case
- Disk-backed and memory efficient storage
- Incremental snapshot for consistent performance
- Fix re-list issues with longer and memory-efficient key key history
Tuesday, March 28, 2017

Kubernetes 1.6: Multi-user, Multi-workloads at Scale

Today we’re announcing the release of Kubernetes 1.6.

- The default backend storage for the API server has been upgraded to use etcd v3 by default for new clusters. If you are upgrading from a 1.5 cluster, care should be taken to ensure continuity by planning a data migration window.
upgrading a ETCD2 cluster to the new (v3) etcd-member service on @coreos Container Linux is the worst documented upgrade procedure ever...
The Five Stages of Grief

1. Denial
2. Anger
3. Bargaining
4. Depression
5. Acceptance
The Five Stages of Grief

1. Denial
2. Anger
3. Bargaining
4. Depression
5. Acceptance
Common Mistakes
Where’d I leave my keys?

Demo
HTTP & gRPC have different namespaces
Wait, did that API change?

Demo
HTTP & gRPC have different namespaces
The world is flat

Demo
Directories are gone (sorry)

The `ls` convenience method is gone
New and Improved!
Querying Based on Prefixes
Ranges and Prefixes

They’re super cool, use them
Can I track changes?

demo
In v3? --rev is great

Track changes over time, woo
Read the Docs for More

“Interacting with etcd”
Quorum is Key
<table>
<thead>
<tr>
<th>CLUSTER SIZE</th>
<th>MAJORITY</th>
<th>FAILURE TOLERANCE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>6</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>7</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>8</td>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td>9</td>
<td>5</td>
<td>4</td>
</tr>
</tbody>
</table>
Does etcd do Byzantine Fault Tolerance?

(Nope)
RAFT doesn’t handle BFT

So, explicitly, neither does etcd
What happens if my 3 node cluster loses 2 nodes?

rip ur ded
(also, Demo)
What else is important?

“Migrate applications from using API v2 to API v3”
Ok, How do I upgrade to v3?

Follow “Upgrade etcd from 2.3 to 3.0”
Level up with Fire Drills
Etcd-operator is a guide for Fire Drills

Operator = software defined recovery

github.com/coreos/etcd-operator
Etcd-operator in action

We use the Operator in Tectonic, let’s see it in action
Let us know how you’re using etcd

Try using zetcd in lieu of ZooKeeper

@ThatMightBePaul
@ElsiePhilly