

#### **Getting started with Riak**

NoSQL Live · Boston, MA · March 2010



Rusty Klophaus (@rklophaus)
Sr. Engineer | Basho Technologies
<a href="http://www.basho.com">http://www.basho.com</a>
<a href="mailto:rusty@basho.com">rusty@basho.com</a>

#### Introductions



#### **Rusty Klophaus**

- Senior Engineer
- Joined in August '09



- Engineering Manager
- Joined in February '08





**Andy Gross** 

- VP of Engineering
- Joined in December '07



- Community Manager
- Joined in April '08







# Follow the team: <a href="mailto:obasho/team">obasho/team</a>





# There are 47 different NoSQL projects...

Where does Riak fit in?







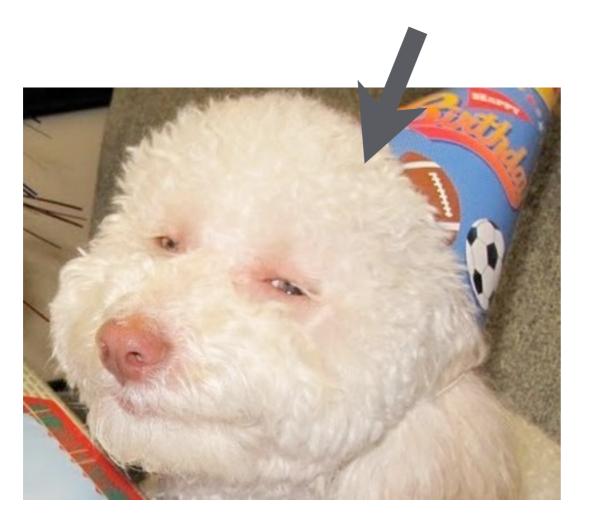
# Riak is a Dynamo-inspired key/value datastore built to scale predictably and easily.





# Riak is a Dynamo-inspired key/value datastore built to scale predictably and easily.

Your ops guy: calm, relaxed, and wearing a party hat.







# What characteristics of Riak become important at different cluster sizes?





# Single Box Riak



- NoSQL Key/Value, Flexible Schema
- Clients in Ruby, Python, Javascript, Java, PHP, Erlang
- Development Interface === Production Interface
- Configurable Buckets "A Profile is not an .mp3"
- Links Lightweight Data Relations





### Small Riak Cluster (~3 boxes)



- Distributed Queries for Performance / Capacity
- Javascript-based Map/Reduce (mini-Hadoop)
- Well-Behaved HTTP
  - nginx proxy config <a href="http://gist.github.com/323048">http://gist.github.com/323048</a>





# Large Riak Cluster (10+ boxes)



- Homogenous No special nodes
- Laugh in the face of machine failure
- Scale by adding machines
- Self-Contained Installation





# Enterprise (\$\$\$ / ~10's of boxes)



- On-Call Support 24x7x365
- Management Tools
- SNMP Monitoring
- Multi-site Replication





# Enterprise (\$\$\$ / ~10's of boxes)



- On-Call Support 24x7x365
- Management Tools
- SNMP Monitoring
- Multi-site Replication





#### **Tutorial**

**Get Riak** 

Connect with a Client

**Store Data** 

Store an Object with Links

Linkwalking

Map/Reduce





### **Get Riak**

#### **Download**

http://downloads.basho.com/riak

#### Start Riak

```
cd riak
bin/riak start
```





# Connect with Python

```
# Code is in ./riak/client_lib/python
import riak

# Connect
client = riak.RiakClient('127.0.0.1', 8098)
```





#### Store Data

```
mybucket = client.bucket('mybucket')
# Create an object...
obj = mybucket.new('myobject')
obj.set data({ 'foo' : 1, 'bar' : 2 })
obj.store()
# Read the object...
obj = mybucket.get('myobject')
print obj.get data()
# Or, open a web browser...
http://127.0.0.1:8098/riak/mybucket/myobject
```





## Store an Object with Links

```
bands = client.bucket('bands')
albums = client.bucket('albums')
members = client.bucket('members')
# Store a band, link to album and members...
obj = bands.new('Winger') \
    .add link(albums.new('Pull', 1275922).store()) \
    .add link(albums.new('IV', 542731).store()) \
    .add link(albums.new('Karma', 200170).store()) \
    .add_link(members.new('Kip Winger').store()) \
    .add link(members.new('Reb Beach').store()) \
    .add_link(members.new('John Roth').store()) \
    .add link(members.new('Rod M.').store()) \
    .store()
```





## Linkwalking





# Map/Reduce

```
# Count the number of sales...
result = obj \
    .link('albums') \
    .map("function(v) { return [v.values[0].data]; }") \
    .reduce("Riak.reduceSum") \
    .run()
```





# Thanks! / Questions?

#### **Questions?**

- riak-users@lists.basho.com (Public Mailing List)
- riak@basho.com (Core Riak Team)
- Follow @basho/team (Basho Twitter List)

#### Riak Resources

- http://riak.basho.com
- http://downloads.basho.com
- ./riak/client\_lib <--- Python, JS, Java, PHP libraries</li>



