



RIAK CLOUD STORAGE

Under Prof. Dr. Christian Baun

Presenters –
Sneha Sahu
Balasubramaniam Elayaperumal

Overview

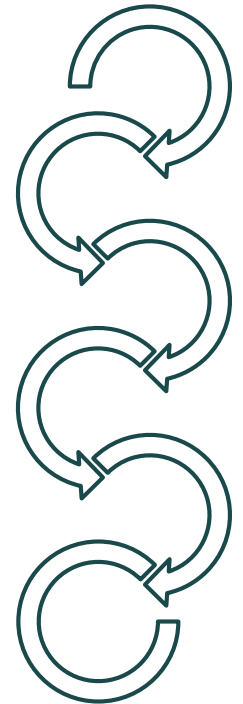
COMMON TERMS

1. FEATURES
2. ARCHITECTURE
3. PRE-REQUISITE
4. DEPLOYMENT
5. COMPARISON
6. CONCLUSION

DEMO

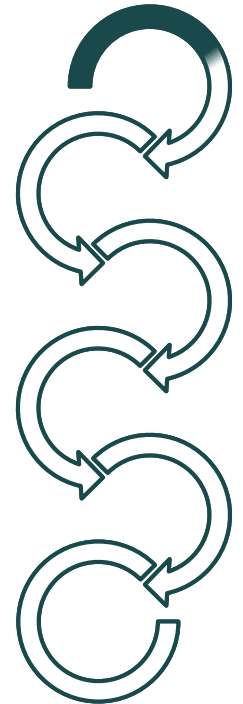
Common Terms

Term	Meaning
RIAK (RIAK KV)	Distributed noSQL database with key-value data structure. Multiple nodes
Riak Node	Not exactly same as a server, except in a production environment. A developer may run multiple nodes on a single laptop.
RIAK CS	Riak Cloud Storage
API	Set of functions and procedures
Reporting APIs	A reporting functionality that supports use cases like accounting, subscription, chargebacks, plugins with billing systems, efficient multi-department utilization, and much more
Stanchion	An application used by Riak CS to manage the serialization of requests, which enables Riak CS to manage globally unique entities like users and bucket names. Unlike Riak and Riak CS, which both run on multiple nodes in the cluster, there should be only one running Stanchion instance in the Riak CS cluster



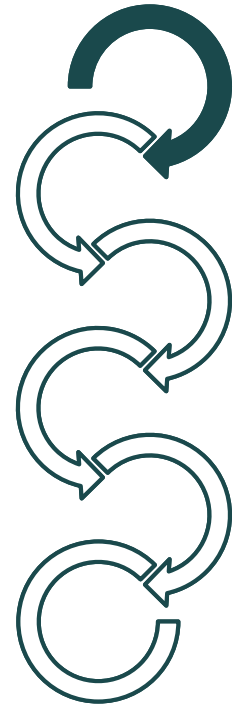
Features

- ▶ Open source cloud storage software that can be used to build private or public clouds
- ▶ Distributed Storage
 - Riak CS is built on top of RIAK (Riak KV).
 - multiple Riak Nodes form a Riak Cluster.
- ▶ Master-less cluster nodes
 - unlike any master-slave model, each node in a Riak cluster is equivalent and contains a complete, independent copy of the whole Riak package.
 - requests are not held hostage to a specific server in the cluster that may or may not be available.
- ▶ Amazon S3- Compatible API



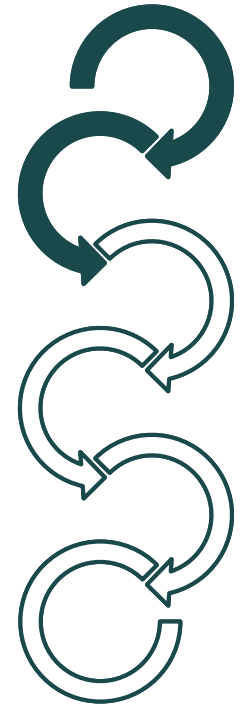
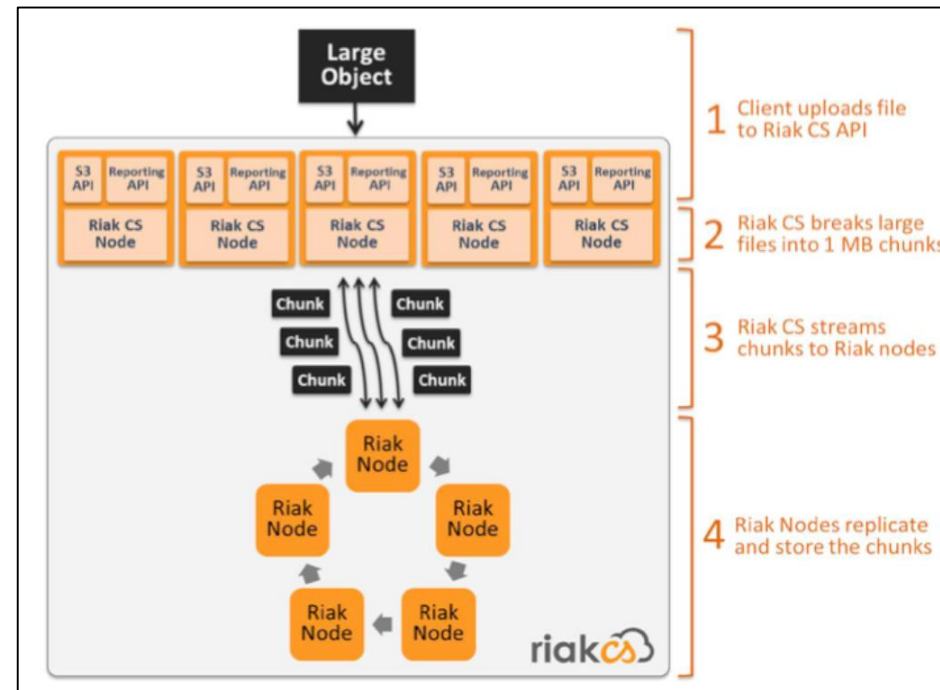
Features

- ▶ Multi-tenancy and per-tenant reporting
 - to access usage data and statistics over network I/O of each user.
- ▶ Larger Object support – i.e. enables storage of any type of conceivable data
 - images, video, documents, database backups, software binaries, etc.
- ▶ Multipart file upload
 - upload very large files to Riak CS as a series of parts. Parts can be between 5MB and 5GB.
- ▶ Multi-Datacenter Replication for Enterprise Editions only
 - greater reliability: active backups, disaster recovery, and data locality.



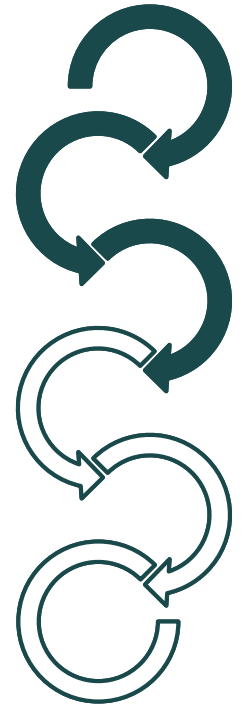
Architecture

- ▶ Consists of 3 software components –
 - **Riak KV** – stores all data and metadata.
 - **Riak CS nodes** (aka S3 API to client)– serves HTTP(S) service by sending requests to Riak KV.
 - **Stanchion** – man in the middle to serialize update requests to keys that needs strong consistency (CS buckets) in Riak KV.
- ▶ Link for all S3 supported API functionalities [~PUT,GET,DELETE, etc.]
 - <http://docs.basho.com/riak/cs/2.1.1/references/apis/storage/s3/>
- ▶ Few administrative APIs that are not in S3 –
 - to know incoming and outgoing traffic via CS nodes
 - to query storage statistics
 - to manage users such as create, disable/enable and update meta information



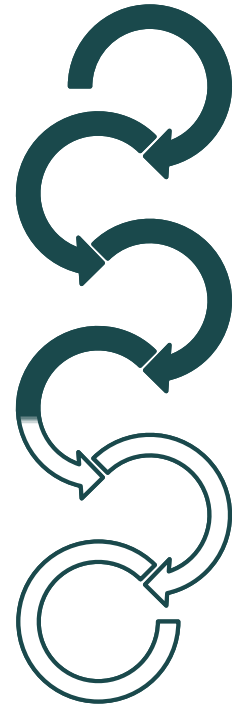
Pre-requisite

- ▶ for Riak CS
 - Riak KV
 - Stanchion
- ▶ for Riak KV
 - Erlang
 - GCC
- ▶ supporting OS
 - Ubuntu, CentOS, Fedora, Solaris, SmartOS, FreeBSD, and OS X.
 - Microsoft Windows
- ▶ DHC Rest Client (Extension for Chrome)



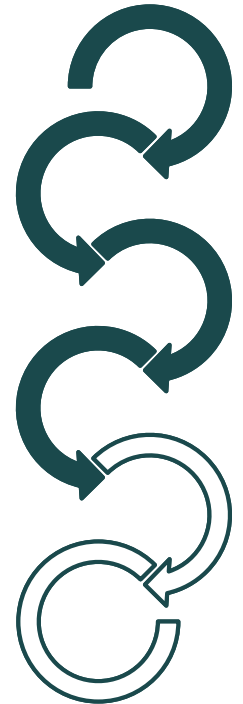
Deployment

- ▶ Install latest version of Erlang and GCC.
- ▶ Install and configure Riak KV on each node (each server in case of production)
 - as part of an OS-specific package, OR
 - from source (in this case, Erlang must be installed first)
- ▶ Install and configure Riak CS on each node
- ▶ Install and configure Stanchion on any one node
- ▶ start Riak KV, stanchion and Riak CS
- ▶ Upload/view using DHC Client



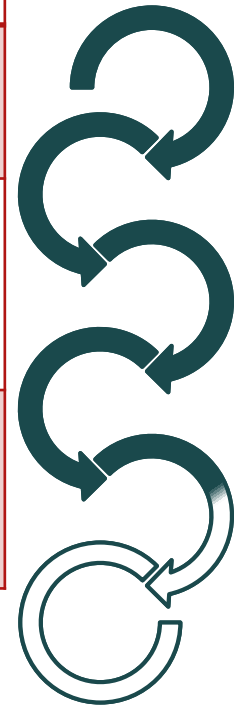
Deployment Links

- ▶ Erlang: <http://docs.basho.com/riak/kv/2.1.4/setup/installing/source/erlang>
- ▶ Riak KV installation:
 - <https://packagecloud.io/basho/riak/install>
 - <http://docs.basho.com/riak/kv/2.1.4/setup/installing/source/>
- ▶ Riak KV configuration for Riak CS: <http://docs.basho.com/riak/cs/2.1.1/cookbooks/configuration/riak-for-cs/>
- ▶ Riak CS installation:
 - <https://packagecloud.io/basho/riak-cs/install>
 - <http://docs.basho.com/riak/cs/2.1.1/cookbooks/installing/#installing-riak-cs-on-debian-or-ubuntu>
- ▶ Riak CS configuration: <http://docs.basho.com/riak/cs/2.1.1/cookbooks/configuration/riak-cs/>
- ▶ Stanchion installation: <http://docs.basho.com/riak/cs/2.1.1/cookbooks/installing/#installing-stanchion-on-a-node>
- ▶ Stanchion configuration: <http://docs.basho.com/riak/cs/2.1.1/cookbooks/configuration/stanchion/>
- ▶ Running Riak CS: <http://docs.basho.com/riak/cs/2.1.1/cookbooks/command-line-tools/>



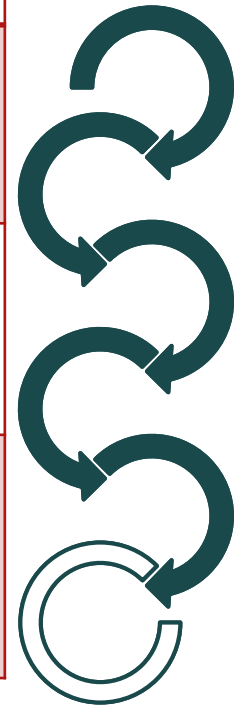
Comparison with SWIFT

feature	RIAK CS	SWIFT
Language	built on Erlang, (a language and platform engineered for extremely high availability)	written in Python
Installation	easy to install with relatively smaller number of components to manage. (~3 components; editing <10 config lines)	requires the installation and ongoing operational supervision of various components, each having deep dependency trees of their own.
Operations	a single administrative command on a newly provisioned host tells the system to <i>automatically integrate</i> the new device.	requires a high degree of manual management.



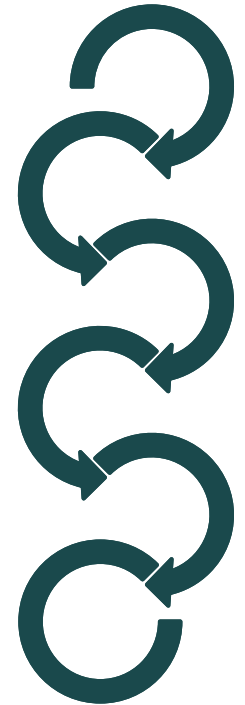
Comparison with SWIFT

feature	RIAK CS	SWIFT
Full-Stack Integration	stands alone as a storage service, having no specific related services for compute, VM image management, etc.	though it can run on its own, but is a part of the OpenStack project (a highly regarded and well-defined "stack" of services)
Write-time communication	always writes to the full number of desired hosts, thus immediately reaching full redundancy	Writes only a majority of replicas before declaring success, later replicate to full content in case of node failure
Support For Amazon S3 API	directly and natively supports the widely adopted S3 API	has its own custom (non-S3) API with its own strengths. Also available - externally developed middleware that emulates the S3 API on top of Swift.



Conclusion

- ▶ (-) not suitable for small database
 - A minimum of 5 data servers in a cluster is recommended ^[6]
- ▶ (+) Fault-tolerant
 - Reads and writes non-stop, regardless of outages or network partitions
- ▶ (+) High availability
 - Riak writes to and reads from multiple servers – Riak cluster nodes
 - cluster nodes can scale dynamically without any downtime
- ▶ (+) Cost effective
- ▶ (+) Easy to set up and maintain



DEMO

LETS SEE HOW IT WORKS...