Hands-on Cloud Storage

@jclouds

jclouds

multi-cloud library

First Things First

Thanks!

@jclouds

jclouds

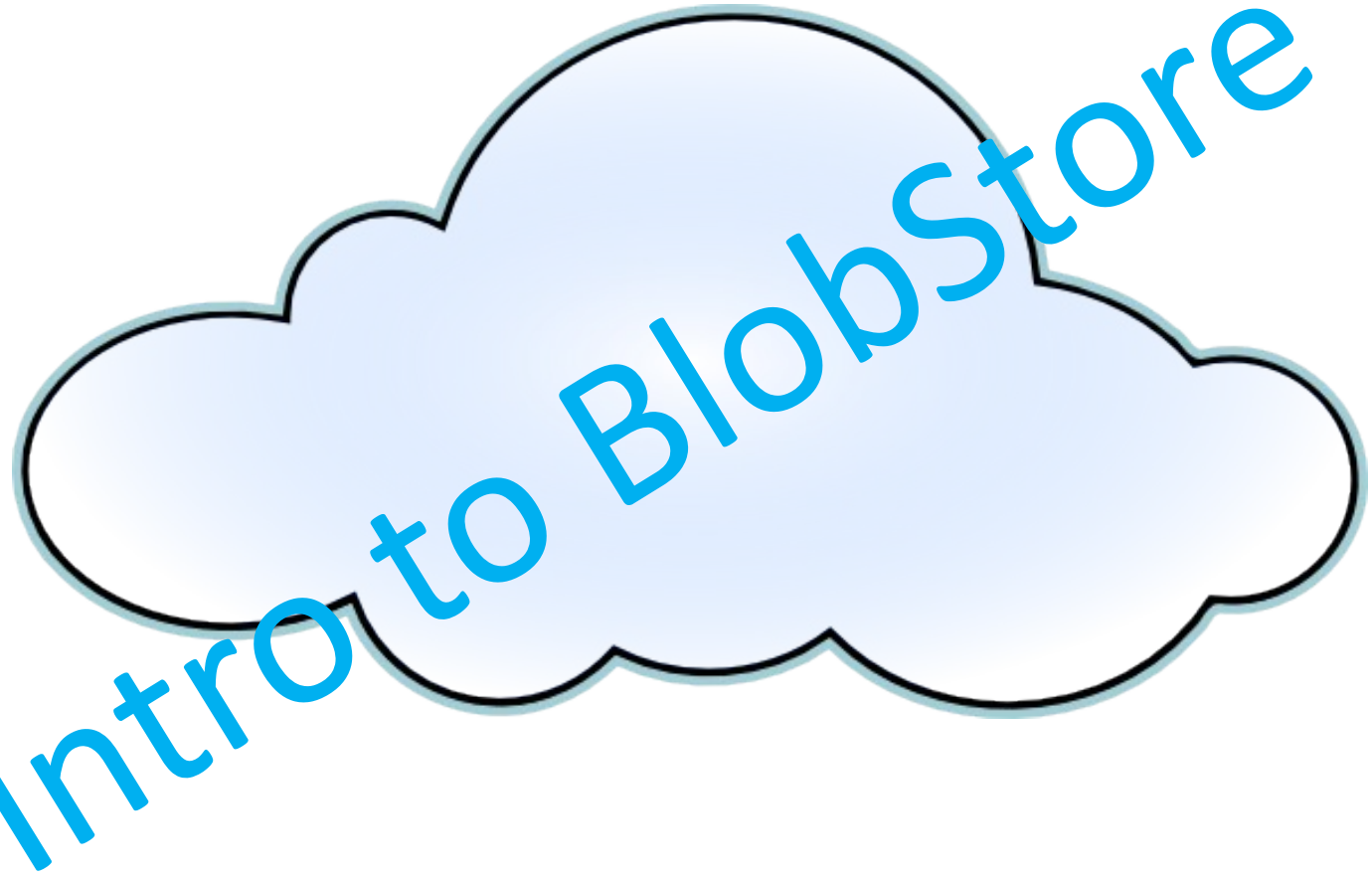
Ego Slide

Adrian Cole (@jclouds)

- Founded jclouds March 2009
- Chief Evangelist at cloudsoft

What's in Store (tee hee)

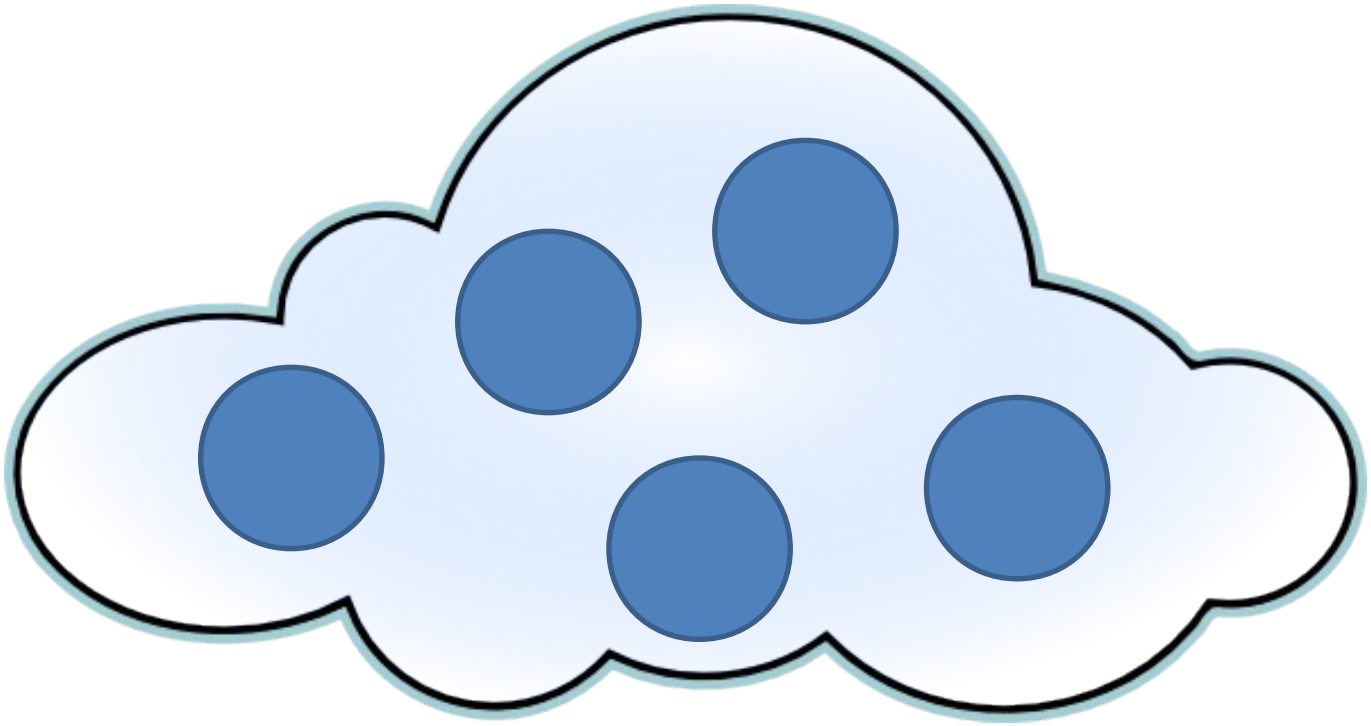
- BlobStore intro
- Setting up
- Let's get coding!
- Review
- Real-world showcase
- From here…



@jclouds

jclouds

Geddit? ;-)



An OSSM Persistence Store

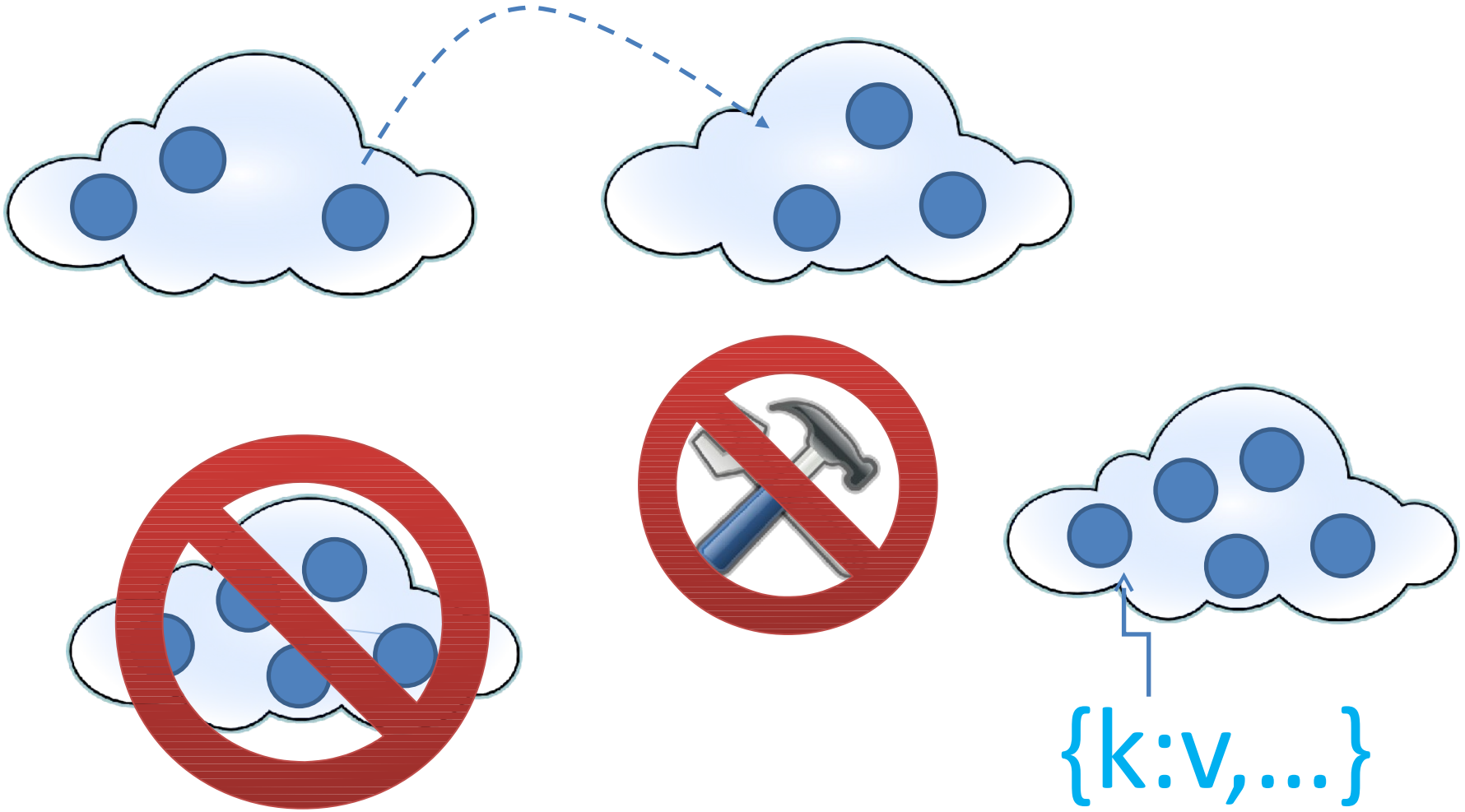
- **O**n-demand
- **S**elf-service
- **S**calable
- **M**easurable

™ Dave Nielsen, CloudCamp

@jclouds

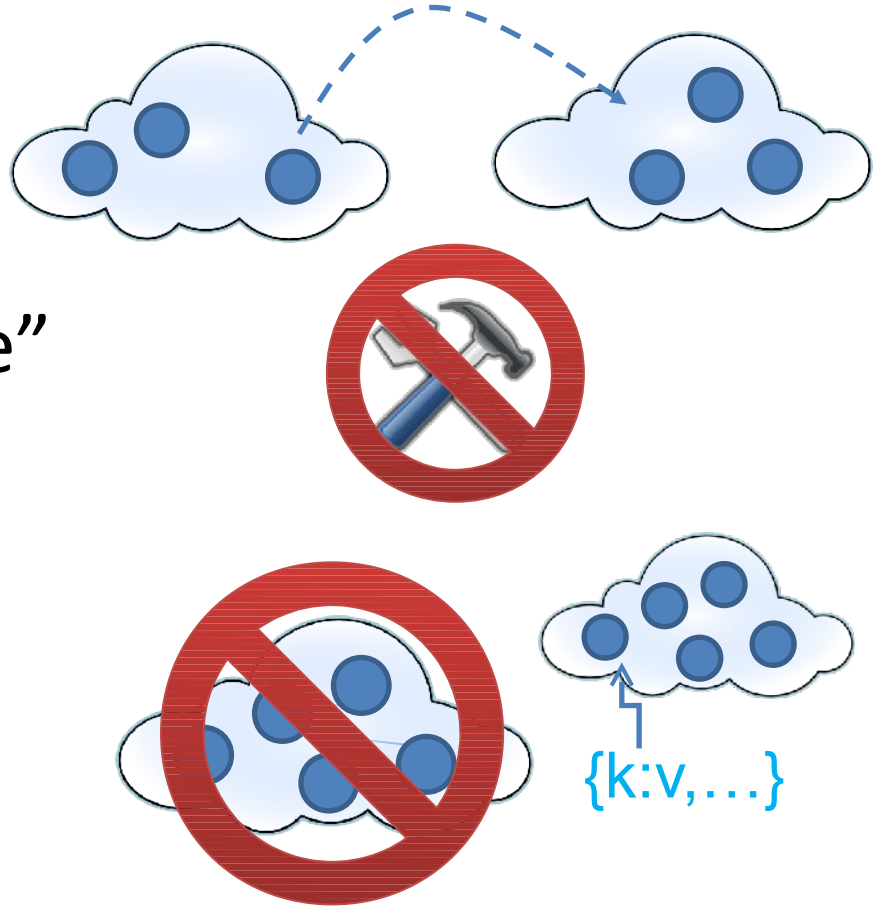
jclouds

Why?



Why?

- Low/No maintenance
- Portability
- No “accidental structure”
- Simple interface
- Choice
- …



In Good Company

@jclouds

jclouds

Why jclouds?

- Data Portability
  - APIs are not as compatible as they might appear
- Code Portability
  - Currently 33 cloud providers
- Enterprise-grad
  - Move petabytes of data
- Parallel operations without threading concerns
  - Outperforms many native SDKs
  - GAE compatible
  - Many tuning options

Why jclouds?

- OSGi compatible
- Clojure binding
- “Invented” many standard SDK features
  - e.g. sync/async APIs
- Tested!
  - “official” TCK for a number of cloud providers
  - also supports offline/local testing

Why jclouds?

- Location metadata
  - Don’t get locked in to a provider’s deployment policy
- Does the hard work so you don’t have to
  - Multi-part in native SDKs vs. `.multipart()` in jclouds
- Strong & active community
  - ~65 contributors, commercial support

No Free Lunches

- Watch for conflicting deps
  - Especially if living at the bleeding edge of fast-changing libs

Show Me The ~~Money~~Code

```
// init
context = new BlobStoreContextFactory()
          .createContext("s3", accesskeyid, secret);
blobStore = context.getBlobStore();
// create container
blobStore.createContainerInLocation(null,
    "adriansmovies");
// add blob
blob = blobStore.blobBuilder("sushi.avi").payload(file)
       .build();
blobStore.putBlob("adriansmovies", blob);
```

Show Me The ~~Money~~Code

```
(use 'org.jclouds.blobstore2)

(def *blobstore* (blobstore “azureblob” account key))

(create-container *blobstore* “movies”)

(put-blob *blobstore* “movies” (blob “tron.mp4“ :payload
tron-file))
```

What's The Big Deal Then?

```
PUT /sushi.avi HTTP/1.1 Host:
adriansmovies.s3.amazonaws.comContent-Length: 734859264Date:
Wed, 01 Mar  2006 12:00:00 GMTAuthorization: signature x-amz-
meta-Chef: Kawasaki
```


```
PUT /adriansmovies/sushi.avi HTTP/1.1 Host:
<account>.blob.core.windows.netContent-Length: 734859264
Date: Wed, 01 Mar  2006 12:00:00 GMTAuthorization:
SharedKey <app>:signature
x-ms-meta-Chef: Kawasaki
```


```
PUT /<api version>/<account>/adriansmovies/sushi.avi HTTP/1.1
Host: storage.clouddrive.com Transfer-Encoding: chunkedX-Auth-
Token: session-token X-Object-Meta-Chef: Kawasaki
```

@jclouds

What's The Big Deal Then?

```
POST /namespace/adriansmovies/sushi.avi HTTP/1.1 Content-
Length: 734859264Date: Wed, 01 Mar  2006 12:00:00 GMT
x-emc-uid: <uid> x-emc-signature: signature x-emc-meta:
Chef=Kawasaki
```


```
POST /<api version>/containers/id_of_ adriansmovies/contents
HTTP/1.1 Content-Length: 734859382
Content-Type=multipart/form-data; boundary=--jclouds--
Authorization=Basic GpjbG9=
----jclouds--
Content-Disposition: form-data; name="sushi.avi";
filename="sushi.avi"
Content-Type: application/octetstring
...

PUT /<api version>/files/from_above/metadata/Chef HTTP/1.1
Content-Length: 8
Content-Type: text/plain
Authorization: Basic GpjbG9=
Kawasaki
```

SOFTLAYER™

@jclouds

jclouds

In The Real World

- Distribution point
  - Splitter
  - Side-loading
- “Sysadmin bridge”
  - PipeHttpResponseToTarxpzfIntoDirectory
  - PipeHttpResponseToBash
- Security filter
  - Allow requests without exposing credentials

In The Real World

- MVC pattern
  - Especially for non-JVM client-side code
- Storage adapter
  - e.g. HDFS
- *More later!*


Setting Up

Let's Get Ready…

- Get a flash drive
- Copy skeleton project
- Run `mvn package` in *exercise1* to check dependencies
- Import *exercise1* into a Maven-supporting IDE
- Phew!


Let's Get Coding!

Anatomy of a BlobStore Project

1. `Create context`
2. `Get BlobStore API`
3. `Do stuff`
4. `Close context`

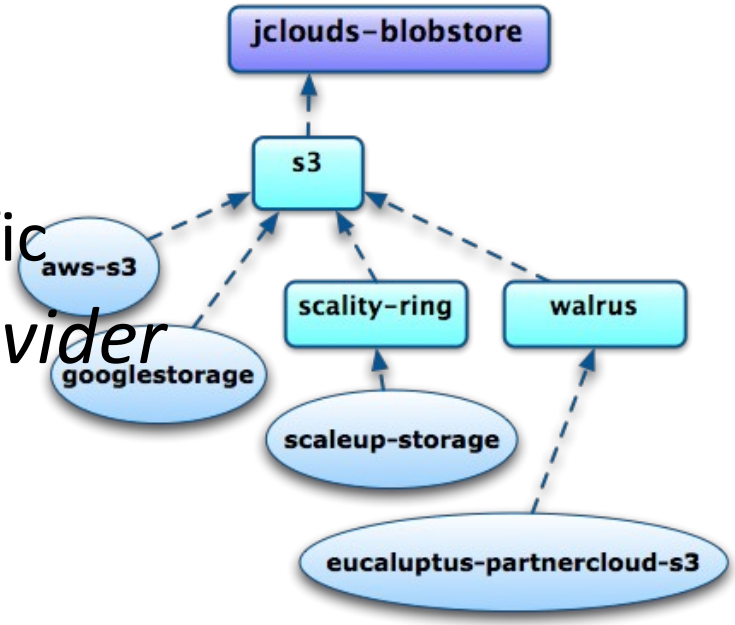
Creating a jclouds Context

- Provider
- Credentials
- Modules
- Additional options

A Theoretical Aside

jclouds modularity

- **APIs** *org.jclouds.api*
  - Describe services
  - "Abstract" and more specific
- **Providers** *org.jclouds.provider*
  - Implement APIs
  - Have location and defaults
- **Drivers** *org.jclouds.driver*
  - Add features to a jclouds context
  - e.g. log4j, Google App Engine support etc.



Tips & Tricks

- Use filesystem provider for offline/free testing
- Useful loggers: wire & headers
- Check tests for sample code
- Don’t reuse containers for testing
- Bear throttling in mind

Exercise 1

It’s All About Content!

TODO

1. Create blob from *cloud.jpg*
2. Upload to container, make readable publically
3. Get public URI
4. Paste URI into browser (erm, )
5. Clean up container

It’s All About Content!

- Content types matter
- Use *Content-Type* and *Content-Disposition* as necessary

Exercise 2

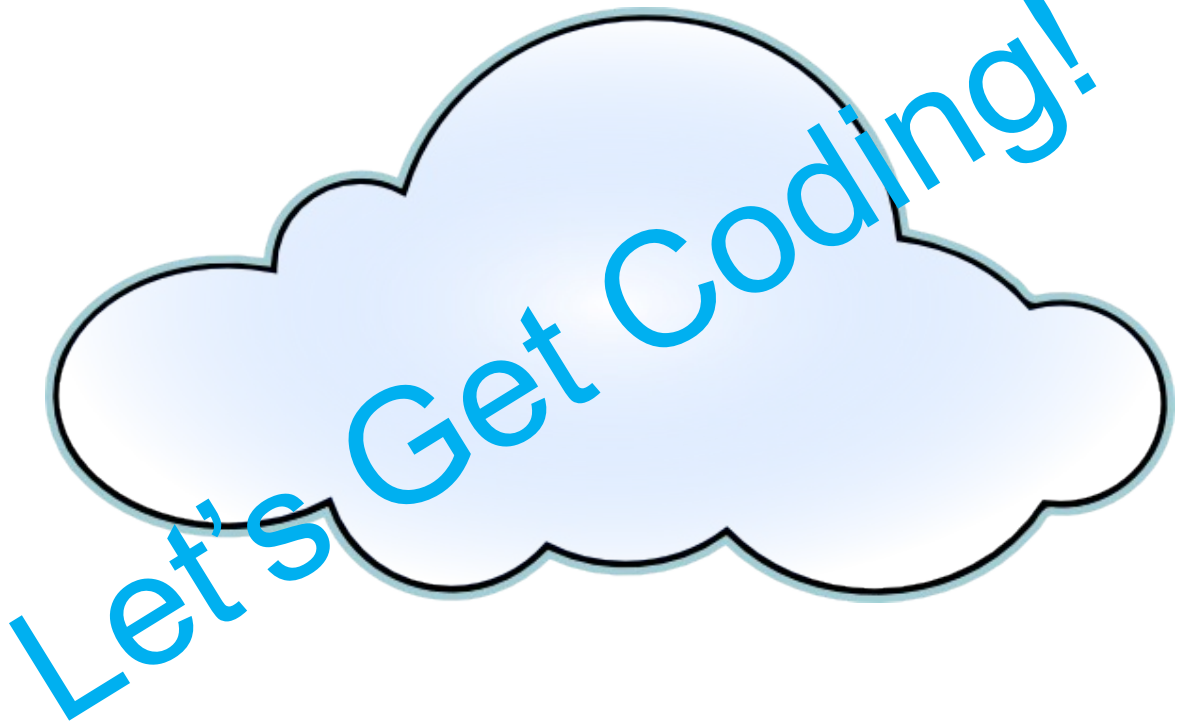
Scattergun

TODO

1. Start timer
2. Upload all five S3 documentation files
3. Measure time
4. Switch to async uploads
5. Report when all uploads have completed

Scattergun

- Can often do multiple tasks asynchronously
- Use listeners to wait for completion
- Useful abstration: async BlobMap



Exercise 3

Now You See It…

TODO

1. Start async upload of S3 doc
2. Wait until blob “exists”
3. Download blob and compare
4. Repeat, using more accurate metadata to determine availability

Now You See It…

- Be aware of your providers' consistency models
- Use appropriate metadata (user metadata, if necessary) to determine consistency
- Can read metadata without getting the whole blob

Exercise 4

Sort It!

TODO

1. Upload S3 docs to one folder, other docs to a different folder in your container
2. List the contents of both containers

Sort It!

- Can have folder structures in blobstores
- Supported natively be some providers, simulated in others
- Useful for organising files
- Public/private access configured at *container* level!


Review


From Here

From Here

- Demos, startup guides and examples
- Compute & LoadBalancer APIs too
  - incl. support for EC2, CloudStack, OpenStack etc.

More Info

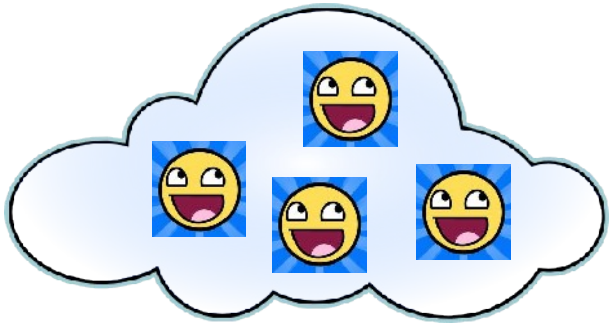
- http://www.jclouds.org
- http://github.com/jclouds/jclouds
- https://github.com/jclouds/jcloudsexamples
- http://www.cloudsoftcorp.com/news/cloudsoft-launches-professional-opensource-support-for-jclouds/
- http://www.cloudsoftcorp.com/wp-content/uploads/jclouds-Datasheet-Web.pdf

@jclouds

cloudsoft

jclouds

Come Hang Out!



- jclouds@googlegroups.com
- @jclouds
- IRC #jclouds on freenode
- work for cloudsoft


jclouds
multi-cloud library