
Taming the Cloud Database
with
Apache jclouds

<http://rack.to/ace14db>



Before we begin: Setup

- Virtual Machine pre-setup provided
 - <http://rack.to/ace14vm>
 - Avoid downloading during this presentation

Before we begin: Setup

- Alternatively you need
 - Oracle Java 7
 - <http://www.webupd8.org/2012/01/install-oracle-java-jdk-7-in-ubuntu-via.html>
 - maven
 - <http://maven.apache.org/download.cgi>
 - `git config --global core.autocrlf false`

Before we begin: Setup

- Testing on your machine (no subscriptions)
 - Devstack
 - <http://devstack.org/>
 - <http://blog.phymata.com/2014/04/18/devstack-icehouse-on-the-rackspace-cloud/>
- Providers: Rackspace, HP, others
 - <http://www.openstack.org/marketplace/public-clouds/>
 - <http://rack.to/ace14>

Before we begin: Setup

```
git clone https://github.com/zack-shoylev/jclouds-developer-examples.git
```

Multiple branches

```
git clone https://github.com/rackerlabs/jclouds-examples.git
```

Branch trove-example-only

Introductions

- Zack Shoylev
 - irc: zacksh twitter: @zackshoylev

freenode #jclouds

Also email

@rackspace.com



Developer Experience

<https://developer.rackspace.com/support/>

Goals

- jclouds
- Create a database in the cloud
- Abstractions and how to contribute
- How to add support for an API

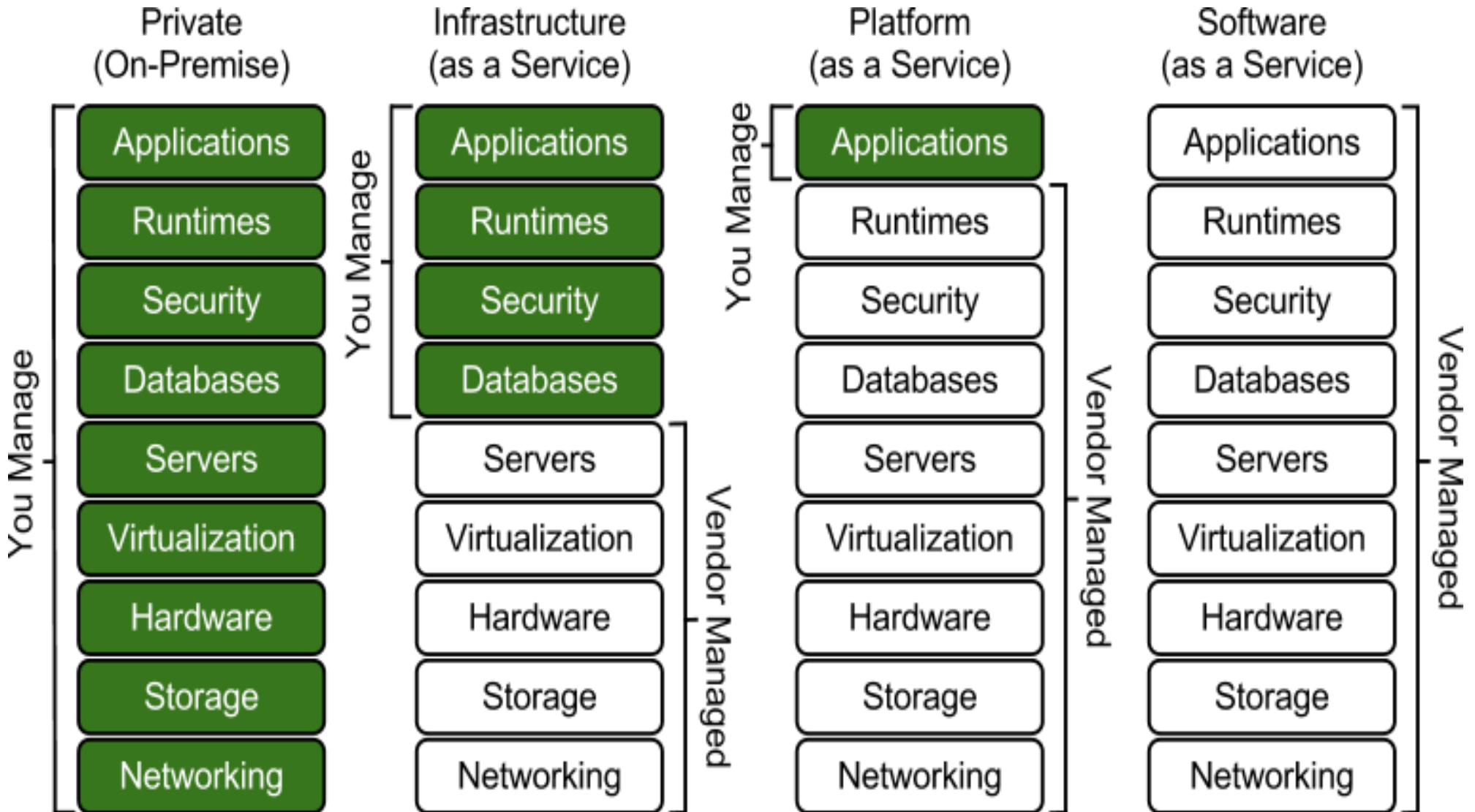
- Networked and distributed computing
- Variety of services
 - Compute (servers)
 - Storage (files)
 - Databases
 - Email
 - ...

The Cloud

- Can't someone else do it?

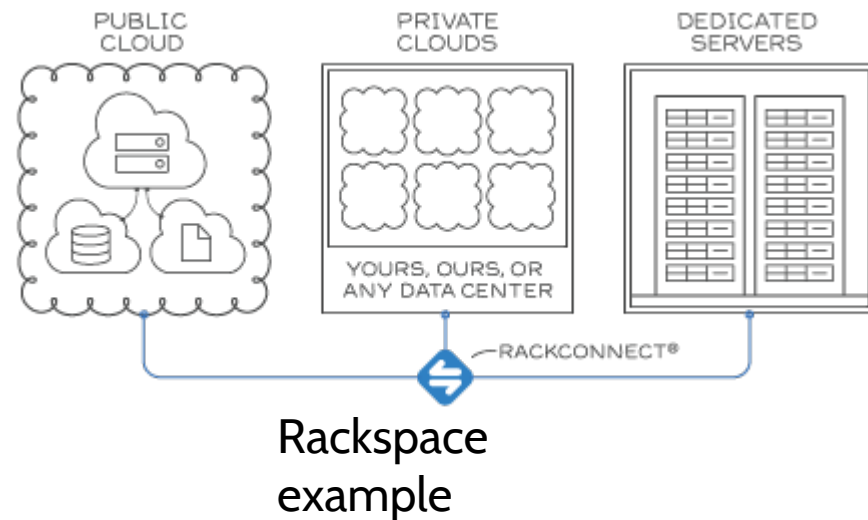


The Cloud

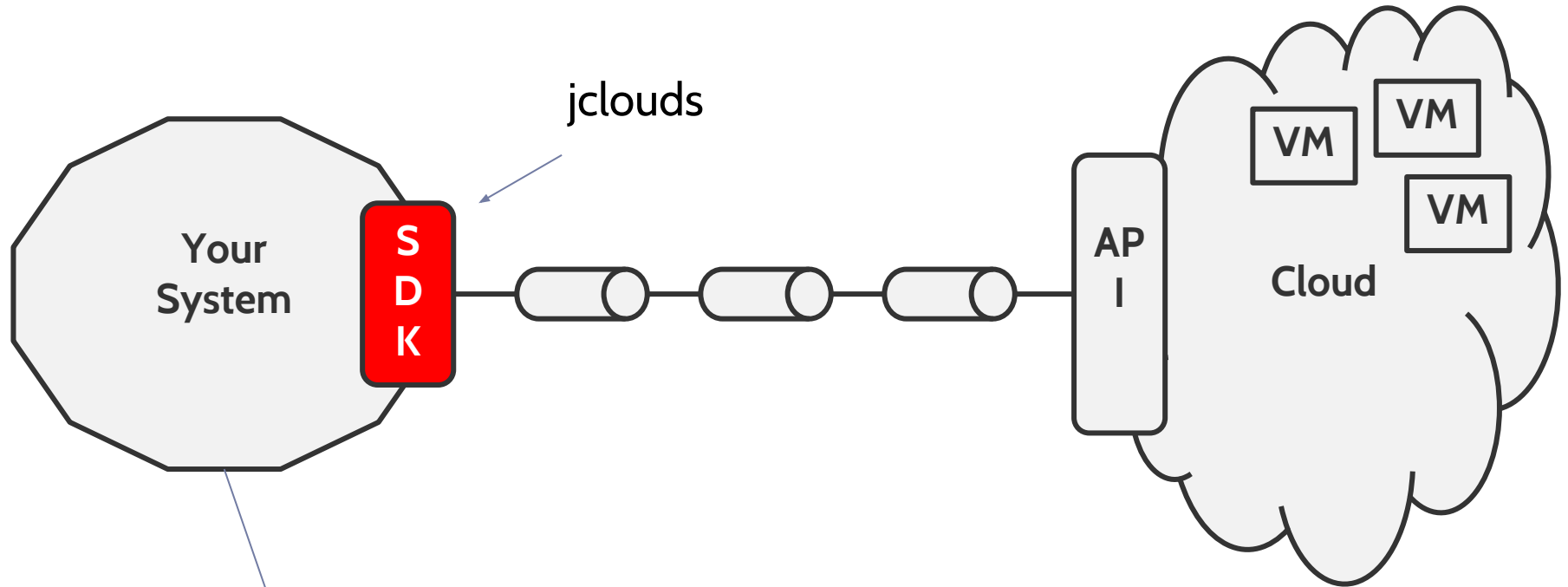


The Cloud

- Public cloud (external provider)
- Private cloud (internal deployment)
- Hybrid cloud
 - Public + Private



The Cloud



- This can be an application server
- Or your home machine
- Or belong to one of your end users
- Or a cloud VM
- Or a smartphone

- **Advantages**
 - Metered pricing (pay as you go)
 - Project scalability (unlimited)
 - Safer (offsite/redundant)
 - Economies of scale
 - Expertise
 - Support

➤ **Disadvantages**

➤ **Less hardware control**

- Provider-controlled downtime
- Virtualization (efficiency)

➤ **Provider lock-in**

- Provider-specific apis/sdks/features
- Expensive to switch clouds or deploy locally
- jclouds minimizes this disadvantage

- Database specific advantages
 - Optimized by provider
 - Settings
 - Container virtualization
 - Automated software updates
 - Redundant data storage
 - Backup
 - Migration
 - Choice
 - Clustering (upcoming)
 - Support

- Cloud OS
- Open-sourced by Rackspace and NASA (2010)
- Free
- Supported
 - AT&T, HP, IBM, Red Hat, Rackspace, Dell, Cisco, Intel, VMware, ...

➤ <http://stackalytics.com/>

- Cloud SDK
- Apache project
- Java (and Clojure)
- Easy
- Portable
- Cloud-agnostic
- Community
- Open source



jclouds

➤ <https://jclouds.apache.org/>




- HTTP requests, responses, retries
- Authentication and re-authentication
- Pagination
- Polling
- Rate limits
- Retries
- Abstractions
- Logging
- Less code


jclouds



➤ **Services**

- Storage
- Compute
- VM Images
- Load Balancers
- DNS
-  Databases
- ...

➤ **Providers**

- Openstack
-  ➤ Rackspace
- HP
- Amazon
- Azure
- ...

Compute Providers

jclouds

AWS	aws-ec2	US-VA,US-CA,IE,SG
Bluelock	bluelock-vcloud-zone01	US-IN
CloudSigma US	cloudsigma-lvs	US-NV
CloudSigma CH	cloudsigma-zrh	CH-ZH
CloudSigma DC	cloudsigma-wdc	US-DC
DigitalOcean	digitalocean	
ElasticHosts GB	elastichosts-lon-b	GB-LND
ElasticHosts GB	elastichosts-lon-p	GB-LND
ElasticHosts US	elastichosts-sat-p	US-TX
Go2Cloud	go2cloud-jhb1	ZA-GP
GoGrid	gogrid	US-CA,US-VA
Green House Data	greenhousedata-element-vcloud	US-WY
HP	hpcloud-compute	US-NV
Ninefold	ninefold-compute	AU-NSW
OpenHosting	openhosting-east1	US-VA
Rackspace UK (First Gen)	cloudservers-uk	GB-SLG
Rackspace US (First Gen)	cloudservers-us	US-IL,US-TX
Rackspace UK (Next Gen)	rackspace-cloudservers-uk	GB-SLG
Rackspace US (Next Gen)	rackspace-cloudservers-us	US-IL,US-TX
SeverLove	serverlove-z1-man	GB-MAN
SkaliCloud	skalicloud-sdg-my	MY-10
SoftLayer	softlayer	

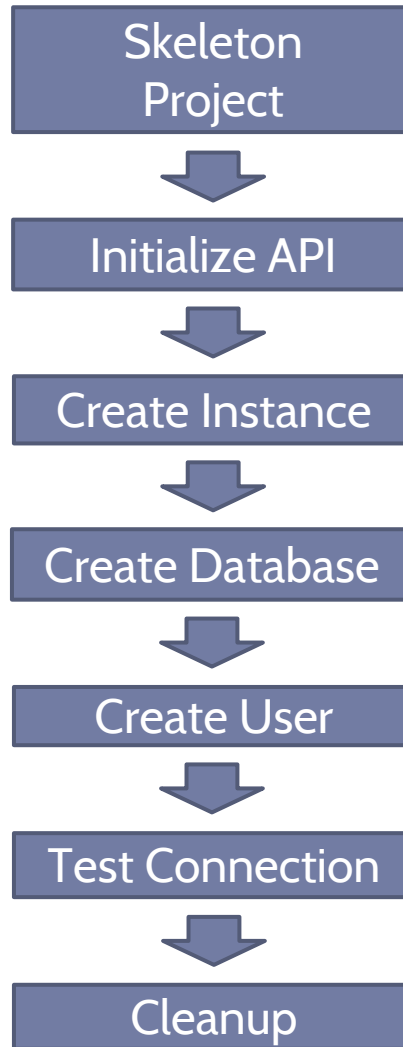


- Showcase database code
- Best practices
- Compare with compute code
- Gotchas and workarounds
- Abstractions

Requirements

- Maven 3
- Java 7+
- jclouds
- Windows or Linux [etc..] (thanks Java!)

Java Project



- **Maven Dependencies**
 - Selective subset
 - Versioning
 - jclouds-labs

POM

<https://github.com/jclouds/jclouds-examples/blob/master/rackspace/pom.xml>

```
<dependency>
```

```
  <groupId>org.apache.jclouds.provider</groupId>
```

```
  <artifactId>rackspace-clouddatabases-us</artifactId>
```

```
  <version>${jclouds.version}</version>
```

```
</dependency>
```

```
<dependency>
```

```
  <groupId>mysql</groupId>
```

```
  <artifactId>mysql-connector-java</artifactId>
```

```
  <version>5.1.25</version>
```

```
</dependency>
```

POM

```
mvn dependency:copy-dependencies "-DoutputDirectory=./lib"
```

Logging

```
// This module is responsible for enabling logging
Iterable<Module> modules = ImmutableSet.<Module> of(new SLF4JLoggingModule());

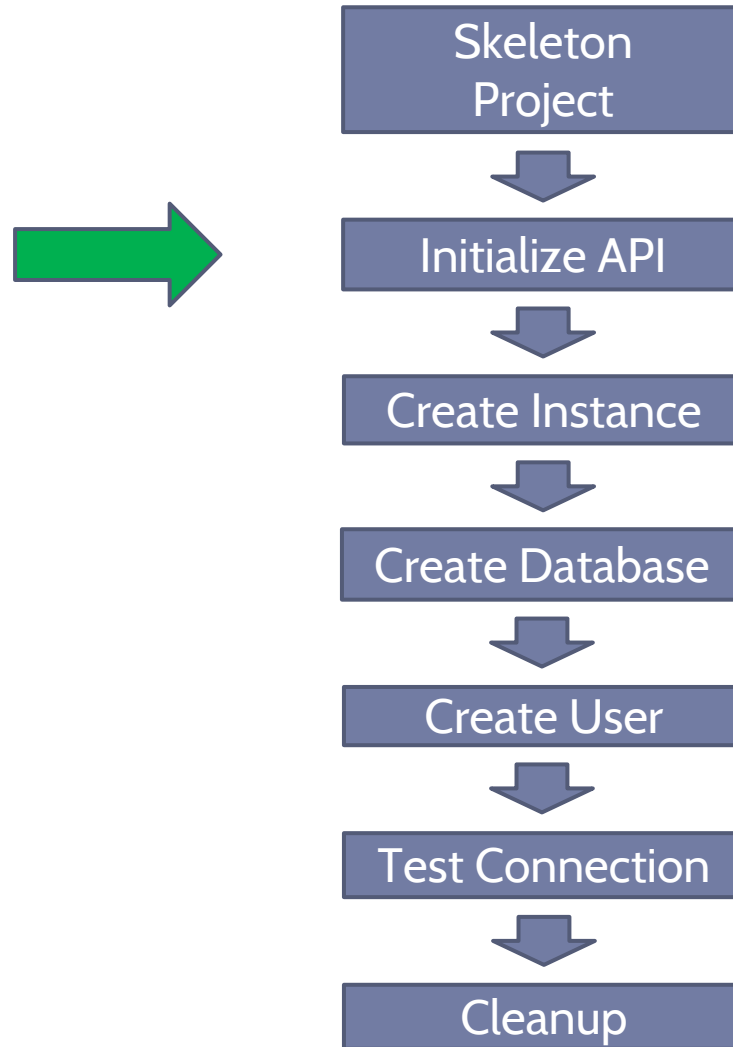
ComputeServiceContext context = ContextBuilder.newBuilder(provider)
    .credentials(username, apiKey)
    .modules(modules) // don't forget to add the modules to your context!
    .buildView(ComputeServiceContext.class);
```

logback.xml

```
<configuration scan="false">
  ...
  <appender name="WIREFILE" class="ch.qos.logback.core.FileAppender">
    <file>target/test-data/jclouds-wire.log</file>

    <encoder>
      <Pattern>%d %-5p [%c] [%thread] %m%n</Pattern>
    </encoder>
  </appender>
```

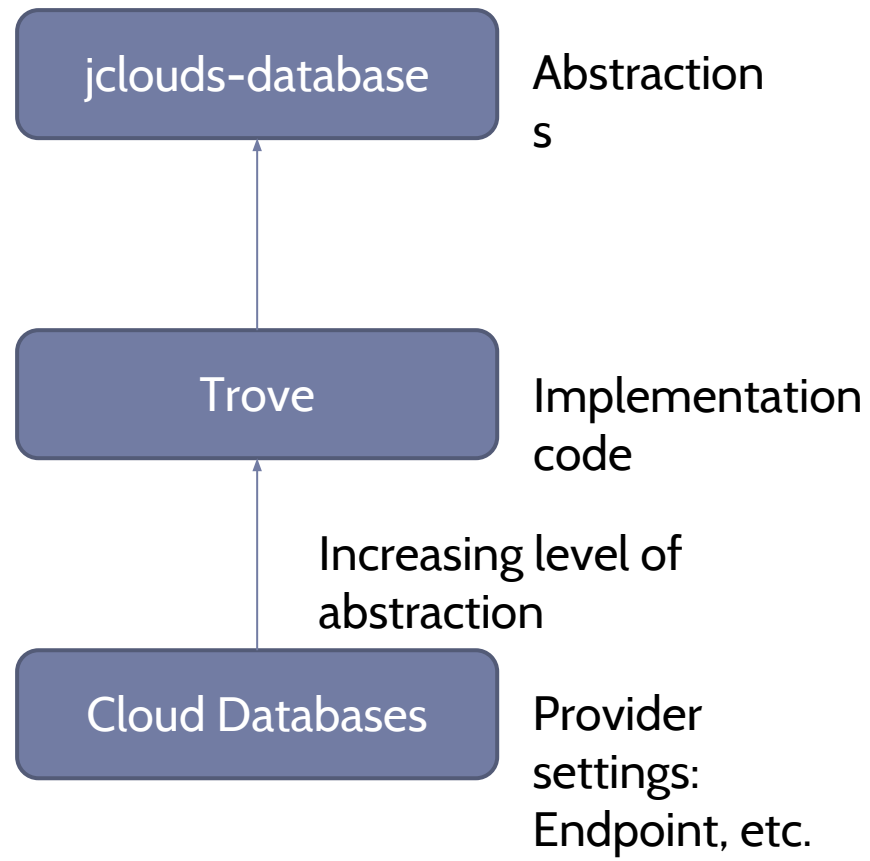
Java Project



Initialize

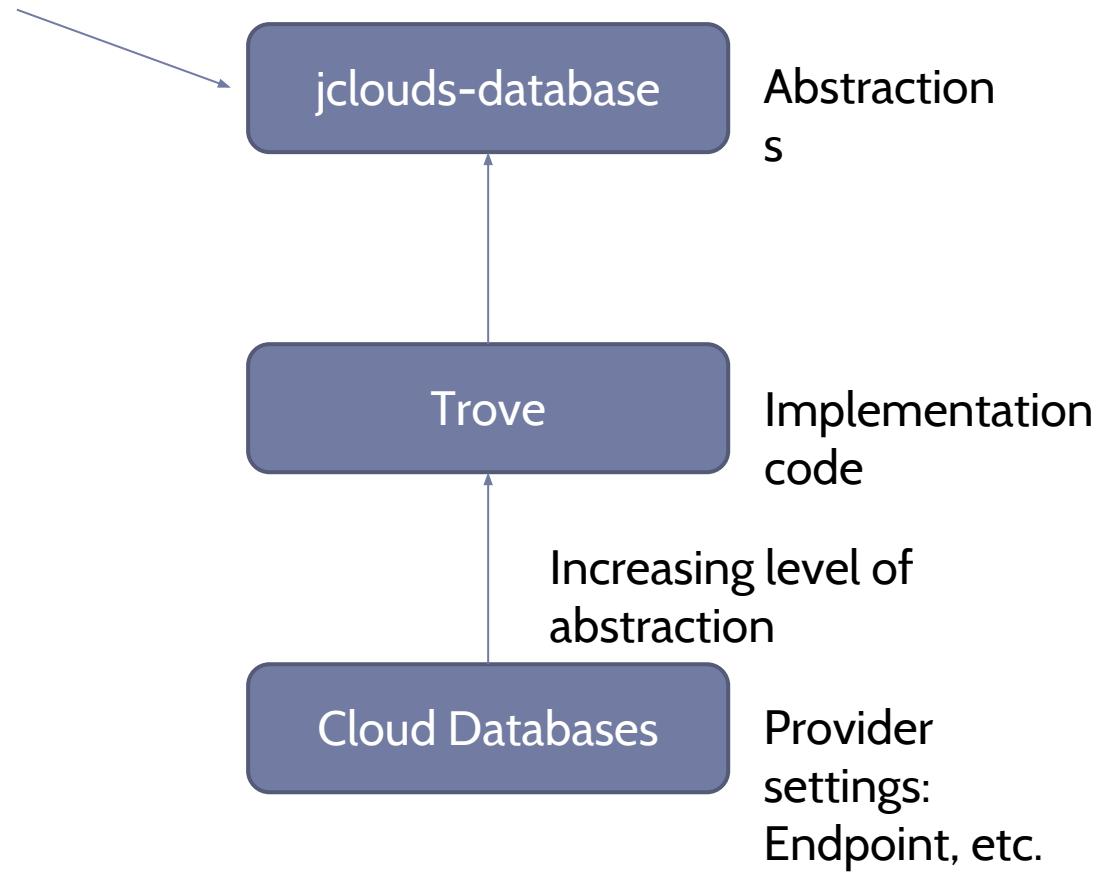
```
private final TroveApi troveApi;  
troveApi = ContextBuilder  
    .newBuilder("rackspace-clouddatabases-us")  
    .credentials(username, apiKey)  
    .buildApi(TroveApi.class);
```

Architecture



Architecture

Not implemented



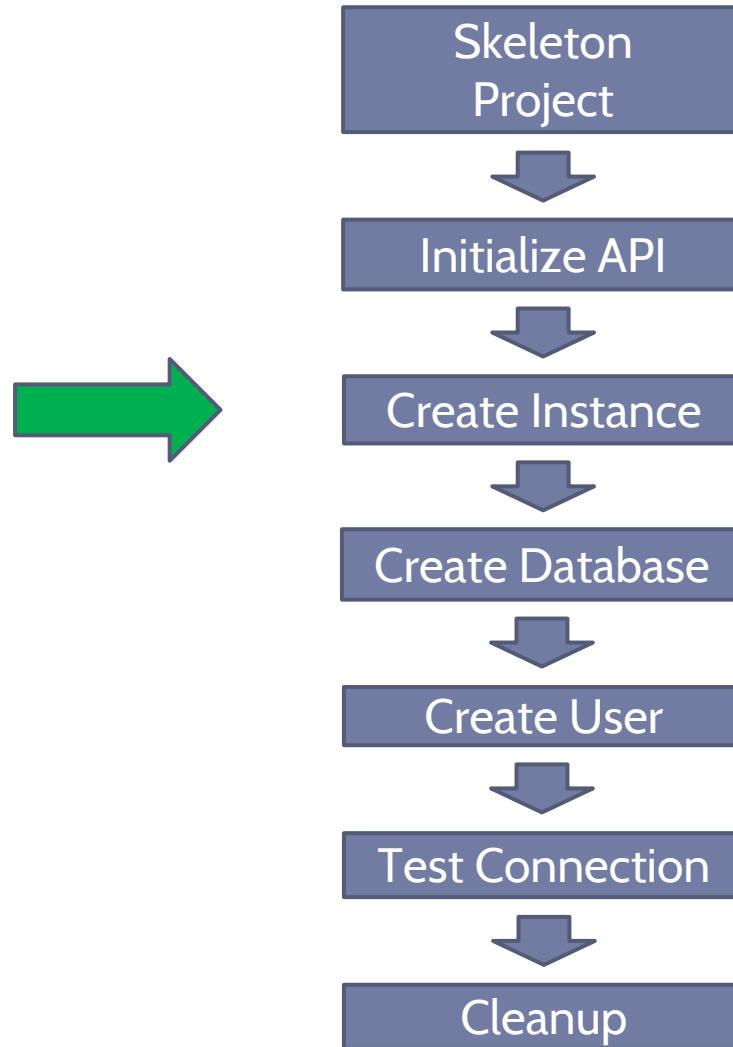
Initialize

```
private final FlavorApi flavorApi;  
flavorApi = troveApi.getFlavorApiForZone(ZONE);
```

APIs

- TroveApi
 - FlavorApi
 - InstanceApi
 - DatabaseApi
 - UserApi
- Utils

Java Project



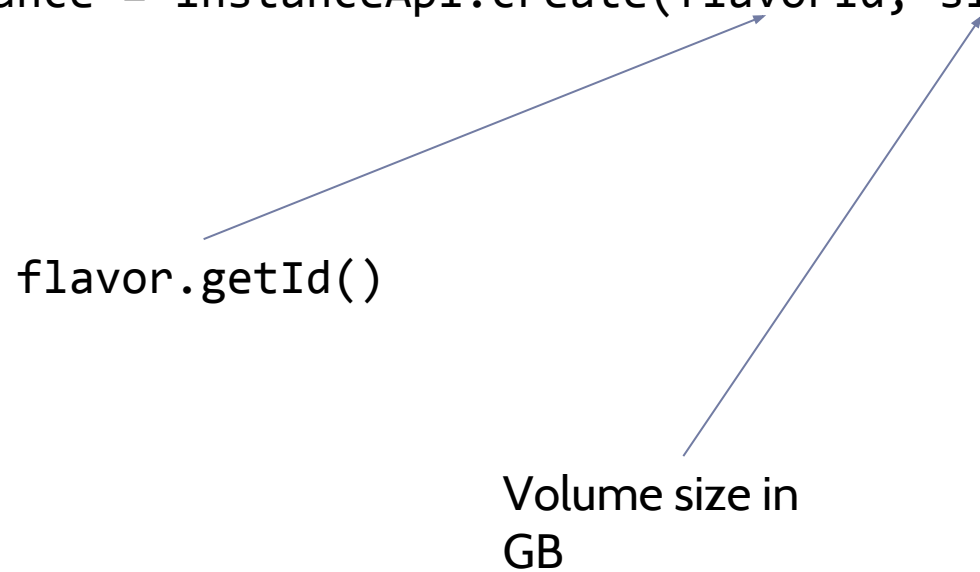
Create Instance

```
Flavor flavor = Iterables.getFirst(flavorApi.List(), null);
```

```
...
```

```
Instance instance = instanceApi.create(flavorId, size, name);
```

flavor.getId()



Volume size in
GB

Create Instance

```
Instance updatedInstance = awaitAvailable(instance, instanceApi);
```


Create Instance

```
Instance updatedInstance = awaitAvailable(instance, instanceApi);
```

Polys status, waits until ready. Will not retry by itself!



Actual Code

```
InstanceApi instanceApi = api.getInstanceApiForZone(zone);

for (int retries = 0; retries < 10; retries++) {

    Instance instance = null;

    try {

        instance = instanceApi.create(flavorId, size, name);

    } catch (Exception e) {

        Uninterruptibles.sleepUninterruptibly(15, TimeUnit.SECONDS);

        logger.error(Arrays.toString(e.getStackTrace()));

        continue;

    }

    Instance updatedInstance = awaitAvailable(instance, instanceApi);

    if (updatedInstance != null) {

        return updatedInstance;

    }

    instanceApi.delete(instance.getId());

    InstancePredicates.awaitDeleted(instanceApi).apply(instance);

}

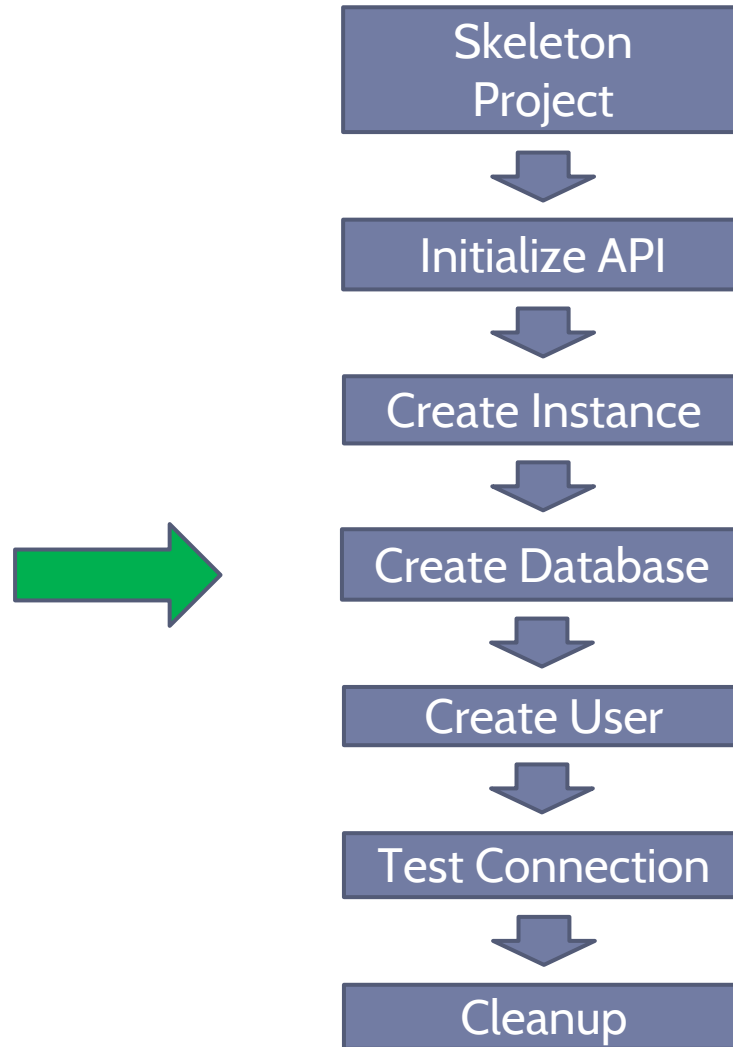
return null;
```

Actual Code

Utils.

```
getWorkingInstance(  
    String zone,  
    String name,  
    String flavorId,  
    int size)
```

Java Project



Create Database

```
troveApi = ContextBuilder.newBuilder(PROVIDER)  
    .credentials(username, apiKey)  
    .buildApi(TroveApi.class);
```

```
instanceApi = troveApi.getInstanceApiForZone(ZONE);
```

```
databaseApi = troveApi
```

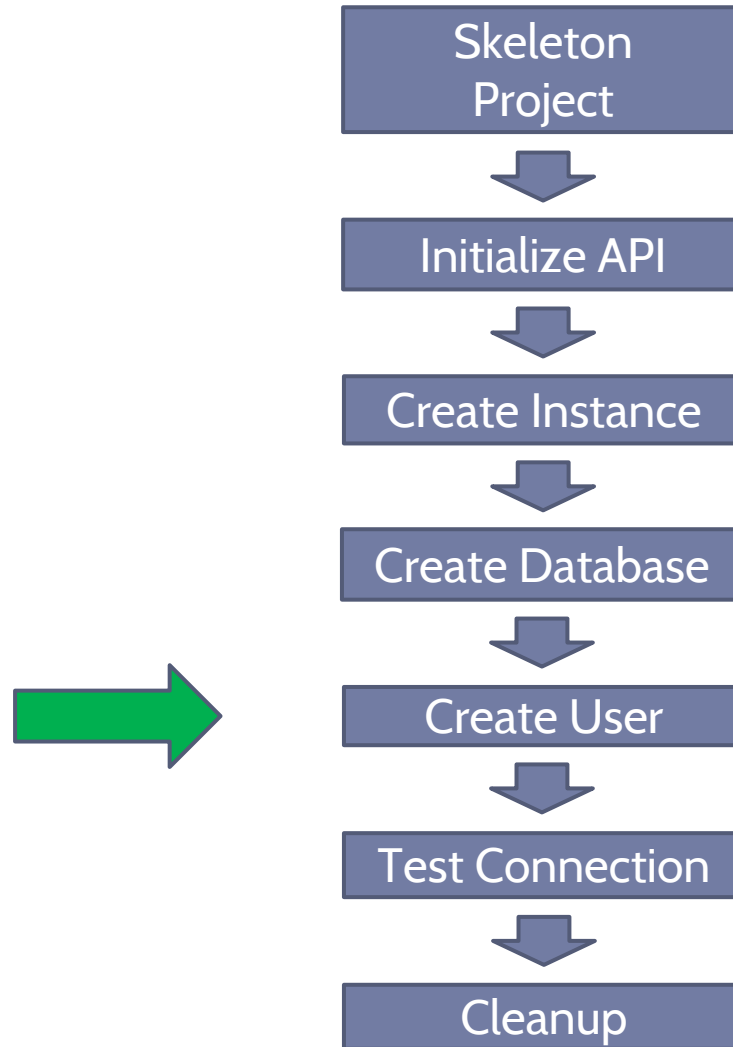
```
.getDatabaseApiForZoneAndInstance(ZONE, getInstance().getId());
```

getSomeApiForXandYandZ – fairly common in jclouds

Create Database

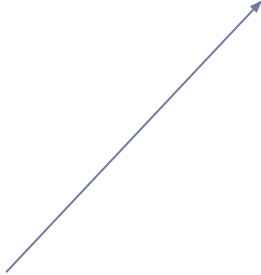
```
boolean result = databaseApi.create(NAME);
```

Java Project



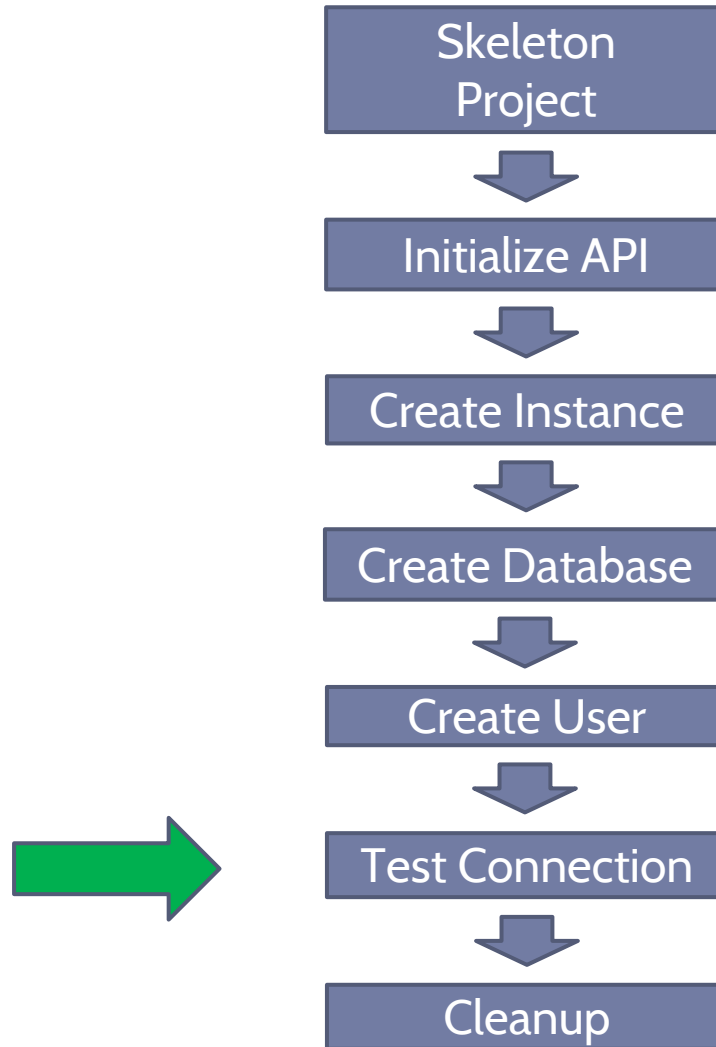
Create User

```
userApi = troveApi  
    .getUserApiForZoneAndInstance(ZONE, instance.getId());  
  
boolean result = userApi.create(UNAME, PASSWORD, DBNAME);
```



User name
User password
Database
name

Java Project



Load balancer needed!

```
CreateLoadBalancer createLB = CreateLoadBalancer.  
builder()
```

```
    .name(NAME)  
    .protocol("MYSQL")  
    .port(3306)  
    .algorithm(LoadBalancer.Algorithm.RANDOM)  
    .nodes(addNodes)  
    .virtualIPType(VirtualIP.Type.PUBLIC)  
    .build();
```

Test Connection

```
loadBalancer = lbApi.create(createLB);
```



Test Connection

```
StringBuilder connString = new StringBuilder();
connString.append("jdbc:mysql://");
connString
.append(getVirtualIPv4(getLb().getVirtualIPs()));
connString.append("/");
connString.append(DBNAME); connString.append("?
user=");
connString.append(UNAME);
connString.append("&password=");
connString.append(PASSWORD);
```

Test Connection

```
Connection conn =  
DriverManager.getConnection(connString.toString());
```



Test Connection

```
Statement stmt = conn.createStatement();  
ResultSet rs = stmt.executeQuery("SELECT 3+5");  
rs.first();  
  
System.out.format(" 3+5 is %s%n", rs.getInt(1));
```

Going forward

- **jclouds examples**

- <https://github.com/jclouds/jclouds-examples>

- **More docs**

- <http://jclouds.incubator.apache.org/documentation/>

- <http://javadocs.jclouds.cloudbees.net/>

- **Maven alternatives?**

- <http://jclouds.incubator.apache.org/documentation/userguide/installation-guide/>

- **Contribute!**

- <https://wiki.apache.org/jclouds/How%20to%20Contribute>

➤ Linux

➤ Compile

- `javac -classpath "lib/*:src/main/java/:src/main/resources/" src/main/java/org/jclouds/examples/rackspace/*.java`

➤ Run

- `java -classpath "lib/*:src/main/java/:src/main/resources/" org.jclouds.examples.rackspace.cloudatabases.CreateInstance username apikey`

➤ Windows

➤ Compile

- `javac -classpath "lib/*;src/main/java/src/main/resources/" src/main/java/org/jclouds/examples/rackspace/*.java`

➤ Run

- `java -classpath "lib/*;src/main/java/src/main/resources/" org.jclouds.examples.rackspace.cloudatabases.CreateInstance username apikey`

Going forward

- **TroveApi**
 - Backup Extension
 - Settings Extension
 - Clustering
- **Abstraction layer**

Going forward

- TroveApi

 - Backup Extension

 - Settings Extension

 - Clustering

- Abstraction layer

Rackspac
e

Microsof
t

Amazo
n

Salesforc
e

CouchDB

```
graph TD; Rackspac --> Abstraction; Microsoft --> Abstraction; Amazon --> Abstraction; Salesforce --> Abstraction; CouchDB --> Clustering;
```

Going forward

- TroveApi

- Backup Extension

- Settings Extension

- Clustering

- Abstraction layer

➤ Rackspac
e

Microsof
t

➤ Amazo
n

Salesforc
e

CouchDB

Going forward

Collaborate!

<http://jclouds.apache.org/>

<http://jclouds.apache.org/community/>



Fork <link to my fork>

Alternatively, fork `jclouds-labs-openstack` (for openstack).

Implementing an API

On to the IDE

Thank you!

<http://developer.rackspace.com>

sdk-support@rackspace.com

Zack Shoylev
Software Developer
zacksh #jclouds
@zackshoylev

