jclouds

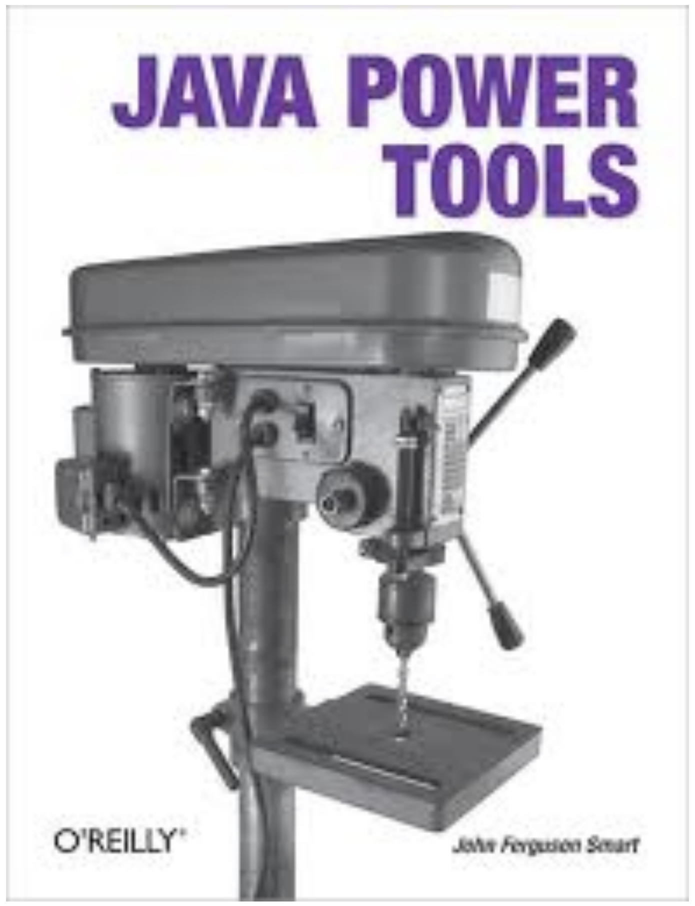
multi - cloud library

Java Power Tools: the cloud edition

adrian@jclouds.org
@jclouds
github jclouds/jclouds

BUY THIS BOOK!

even though I didn't write it!



agenda

intro

tools

questions

intro

Adrian Cole (@jclouds)

- founded jclouds
    - cloud consultant

disclosure

you don't have to be a cloudie or a java gear-head to use these tools.

compute cloud

infrastructure as a service

soft & hardware catalog

on demand machines

priced per hour

What jclouds provides

Portable APIs

Compute	*LoadBalancer*
BlobStore	*Table*

Provider APIs

Driver-Architecture

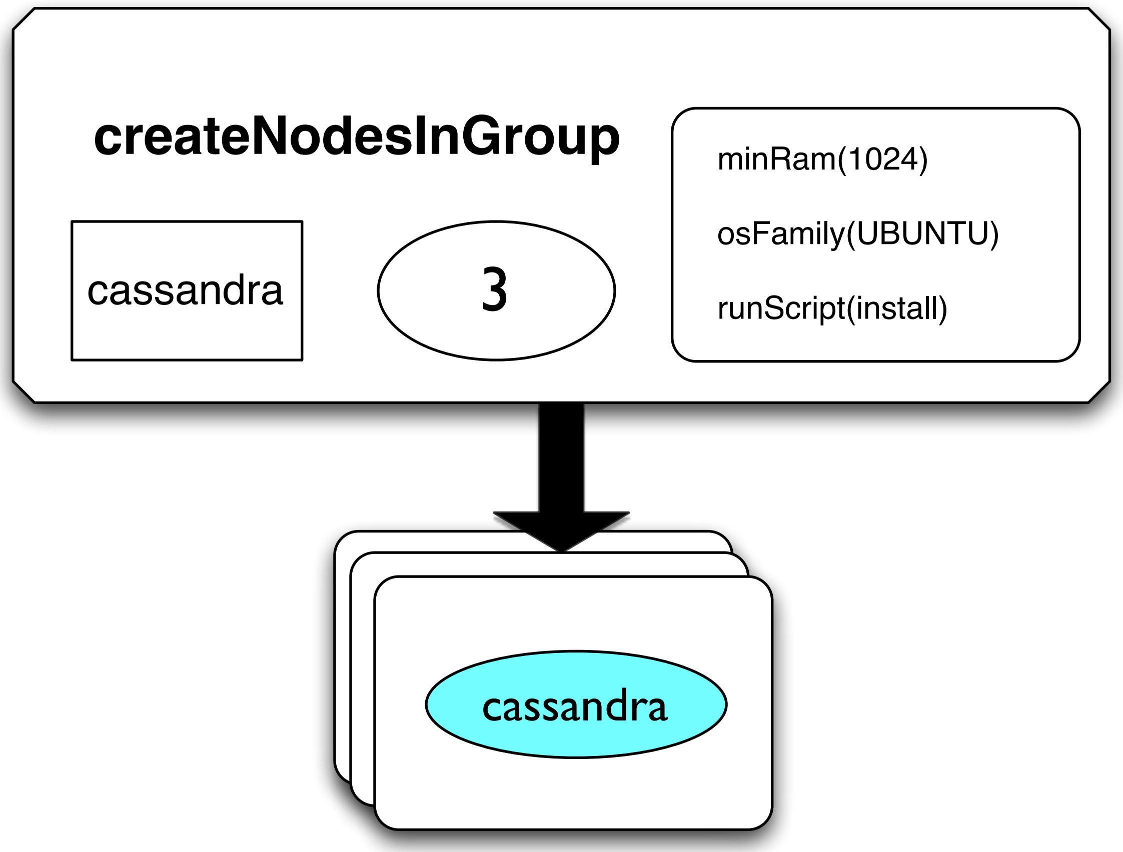
25 Tested Providers!

jcloud concepts

Templates abstractly describe nodes:
What OS to use, what versions...
What kind of hardware, what features are needed...

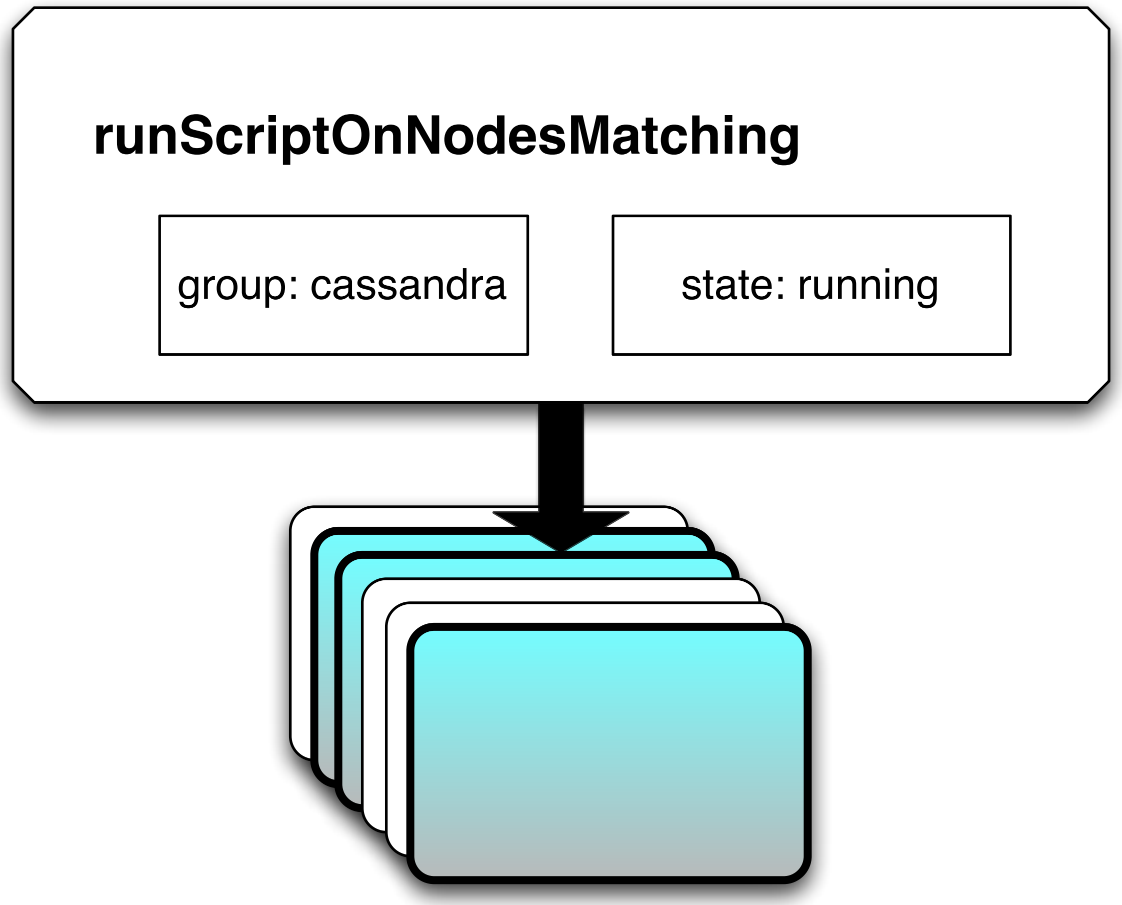
Groups organize identically configured nodes

jclouds bootstrap



jclouds operations

Operate based on predicates



java overview

github jclouds/jclouds

```
context = new ComputeServiceContextFactory().
            createContext("gogrid", apikey, secret,
              singleton(new JschSshClientModule()));

compute = context.getComputeService();

nodes = compute.createNodesInGroup(“cassandra”, 3,
                          runScript(extractAndRun).
                       inboundPorts(22, 8080, 9160));

for (NodeMetadata node : nodes) {
  node.getPublicAddresses();
  node.getPrivateAddresses();
}
```

clojure overview

github jclouds/jclouds

```
(with-compute-service [“aws-ec2” access secret :ssh]
  (def nodes
    (run-nodes “cassandra” 3
      (build-template compute
        :inbound-ports [22,8080,9160]
        :run-script install)))
  (pprint nodes))
```

jclouds power tools

pallet

arquillian

CARGO

ant

your build script uses ssh

create a node on demand

your are now in the cloud

ant compute task

github jclouds/jclouds

```
<compute actions="destroy,create" provider="${url}">
  <nodes group="${group}" os="UBUNTU" hardware="SMALLEST"
            runscript="runscript.sh"
            openports="22,${listenport}"
            publickeyfile="${publickeyfile}"
            hostproperty="host"
            usernameproperty="username" />
</compute>
```

cargo

deploy your application

choose your appserver

cloud is your container

cargo ssh integration

codehaus/cargo
github jclouds/jclouds-examples

```
<cargo containerId="tomcat6x" output="build/output.log"
                        log="build/cargo.log" action="start" timeout="600000">
  <zipurlinstaller installurl="${container.zip}" />
  <configuration home="build/cargo" type="standalone">
    <deployable type="war" file="${warfile}"/>
    <property name="cargo.logging" value="high"/>
    <property name="cargo.java.home" value="/usr/lib/jvm/java-6-openjdk"/>
    <property name="cargo.hostname" value="${host}"/>
    <property name="cargo.servlet.port" value="${listenport}"/>
    <property name="cargo.ssh.host" value="${host}"/>
    <property name="cargo.ssh.username" value="${username}"/>
    <property name="cargo.ssh.keyfile" value="${privatekeyfile}"/>
    <property name="cargo.ssh.remotebase" value="/var/cargo"/>
  </configuration>
</cargo>
```

arquillian

unit test your deployment

skip the build

cloud is an option

arquillian cloud container

jboss/arquillian

```
@Inject
private Cache<String, Integer> cache;

@Test @DeploymentTarget("active-1-dep")
public void callActive1() throws Exception
{
   int count = incrementCache(cache);
   Assert.assertEquals(1, count);
}

@Test @DeploymentTarget("active-2-dep")
public void callActive2() throws Exception
{
   int count = incrementCache(cache);
   Assert.assertEquals(2, count);
}
```

```
<cloud:container>
  <cloud:provider>gogrid</
  <cloud:identity>apikey<
  <cloud:credential>secret</
</cloud:container>
```

hudson plugin

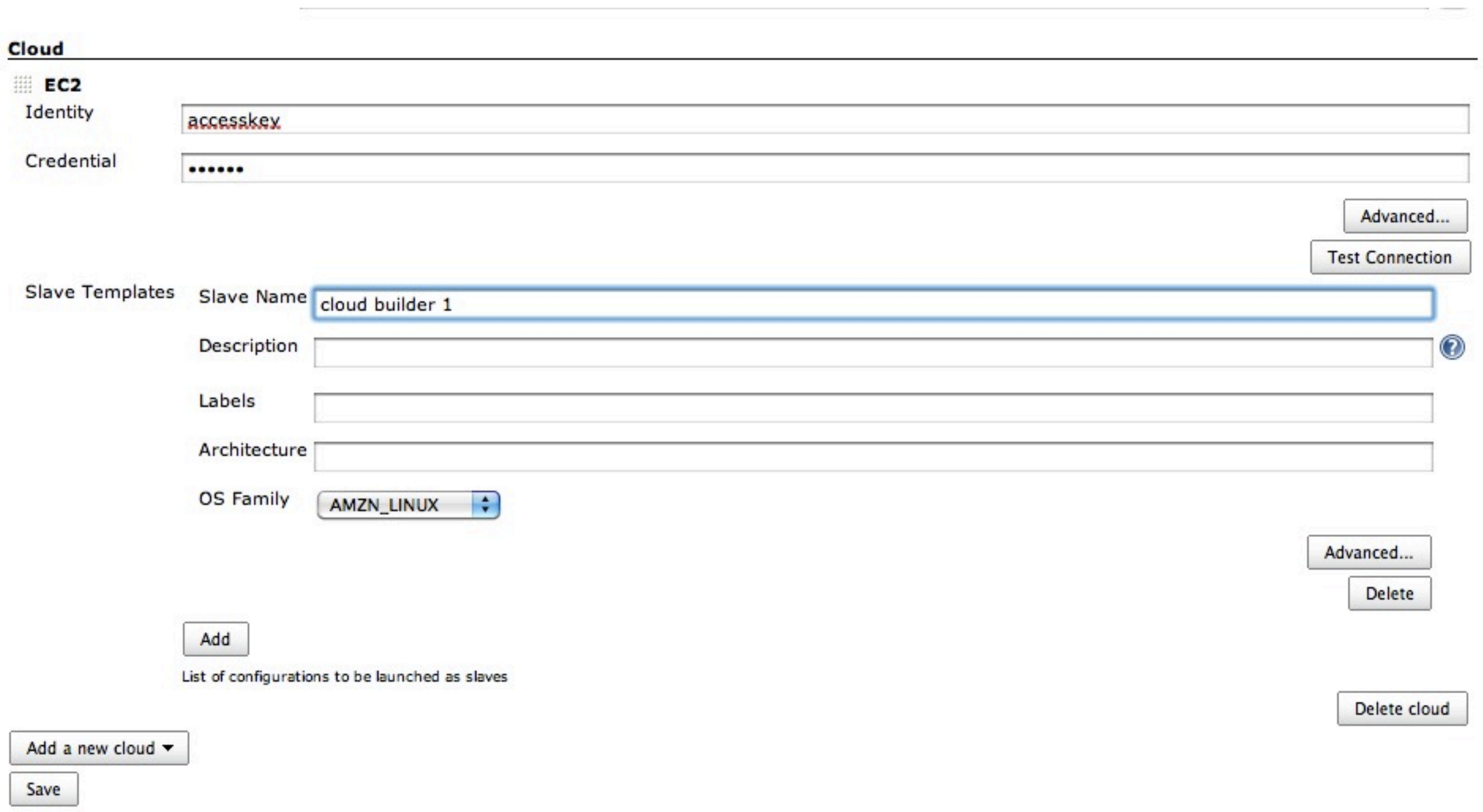
under construction

build on alternate platforms

offload your laptop

the cloud is your slave

hudson jclouds plugin java.net/hudson


Cloud

EC2

Identity	accesskey
Credential	••••••

Slave Templates

Slave Name	cloud builder 1
Description	
Labels	
Architecture	
OS Family	AMZN_LINUX

List of configurations to be launched as slaves

whirr

create and control services

like hadoop

cloud is a cluster

whirr

apache incubator

```
spec = new ClusterSpec();
spec.setProvider(“gogrid”);
spec.setIdentity(apikey);
spec.setCredential(secret);
spec.setClusterName(“cassandra”);
spec.setInstanceTemplates(ImmutableList.of(
     new InstanceTemplate(3,”cassandra”)));

cluster = new Service().launchCluster(spec);
Set<Cassandra.Client> clients = clients(cluster);
```

```
$ whirr launch-cluster \
  --cluster-name=cassandra \
  --instance-templates='3 cassandra'
```

pallet

build your environment

manage at runtime

cloud is clojure

pallet

github pallet/pallet

```
(defnode webserver
  "Basic  web  app, served by tomcat"
  {:os-family :ubuntu
   :os-version-matches "10.04"
   :inbound-ports [8080 22]}
   :bootstrap (phase (public-dns-if-no-nameserver)
                     (automated-admin-user))
   :configure (phase (tomcat))
   :deploy(phase (tomcat-deploy "webapp.war")))

(converge {webserver 1} :compute service)
```

github jclouds/jclouds

github jclouds/jclouds-examples
github jclouds/jclouds
codehaus cargo
jboss arquillian
java.net hudson
apache whirr
github pallet/pallet
github pallet/pallet-examples

Questions?

adrian@jclouds.org
@jclouds
http://www.meetup.com/jclouds