# How to Write a Whirr Service

Tom White, Cloudera, @tom_e_white

jclouds Meetup, GoGrid

10 March 2011

# What is Apache Whirr?

- An Apache Incubator project for running services in the cloud
- A community for sharing code and configuration for running clusters
  - People: 9 committers (6 orgs), more contributors and users
  - Projects: Hadoop, HBase, ZooKeeper, Cassandra
    - More coming: Voldemort, Mesos, Oozie
- Whirr is used for
  - Testing
  - Evaluation and proof of concept
  - Production (future)

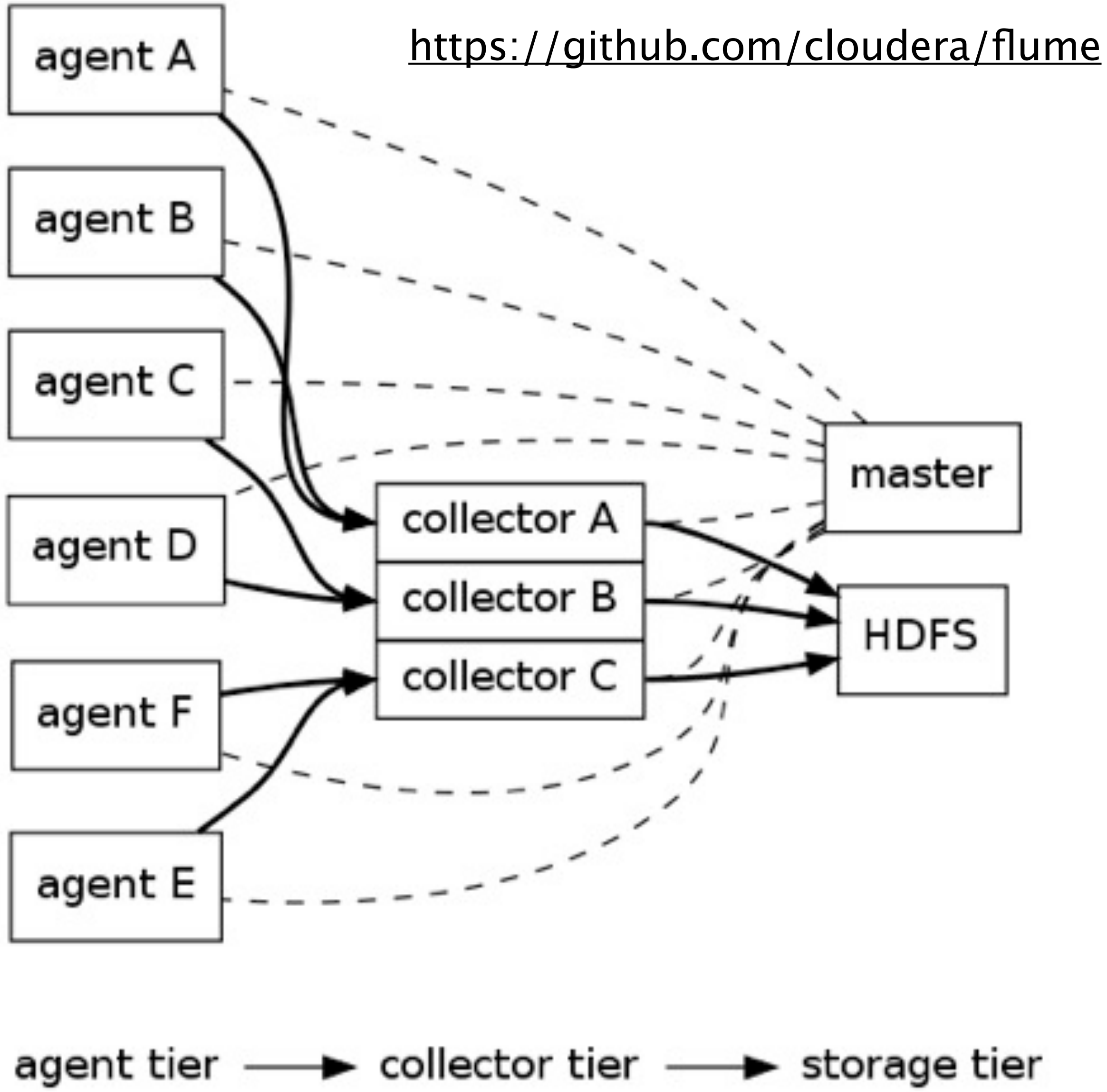
# Steps in writing a Whirr service

- 1. Identify service roles
- 2. Write a ClusterActionHandler for each role
- 3. Write scripts that run on cloud nodes
- 4. Package and install
- 5. Run

# 1. Identify service roles

- Flume, a service for collecting and moving large amounts of data
- Flume Master
  - The head node, for coordination
  - Whirr role name: `flumedemo-master`
- Flume Node
  - Runs agents (generate logs) or collectors (aggregate logs)
  - Whirr role name: `flumedemo-node`

https://github.com/cloudera/flume



# 2. Write a ClusterActionHandler for each role

```
public class FlumeNodeHandler extends ClusterActionHandlerSupport {

  public static final String ROLE = "flumedemo-node";

  @Override public String getRole() { return ROLE; }

  @Override
  protected void beforeBootstrap(ClusterActionEvent event) throws IOException,
      InterruptedException {
    addStatement(event, call("install_java"));
    addStatement(event, call("install_flumedemo"));
  }

  // more ...
}
```

# Handlers can interact...

```java
public class FlumeNodeHandler extends ClusterActionHandlerSupport {

  // continued ...

  @Override
  protected void beforeConfigure(ClusterActionEvent event) throws IOException,
      InterruptedException {
    // firewall ingress authorization omitted

    Instance master = cluster.getInstanceMatching(role(FlumeMasterHandler.ROLE));
    String masterAddress = master.getPrivateAddress().getHostAddress();
    addStatement(event, call("configure_flumedemo_node", masterAddress));
  }
}
```

# 3. Write scripts that run on cloud nodes

- install_java is built in
- Other functions are specified in individual files

```
function install_flumedemo() {
  curl -O http://cloud.github.com/downloads/cloudera/flume/flume-0.9.3.tar.gz
  tar -C /usr/local/ -zxf flume-0.9.3.tar.gz
  echo "export FLUME_CONF_DIR=/usr/local/flume-0.9.3/conf" >> /etc/profile
}
```

# You can run as many scripts as you want

- This script takes an argument to specify the master

```
function configure_flumedemo_node() {
  MASTER_HOST=$1
  cat > /usr/local/flume-0.9.3/conf/flume-site.xml <<EOF
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl"  href="configuration.xsl"?>
<configuration>
  <property>
    <name>flume.master.servers</name>
    <value>$MASTER_HOST</value>
  </property>
</configuration>
EOF
  FLUME_CONF_DIR=/usr/local/flume-0.9.3/conf \
    nohup /usr/local/flume-0.9.3/bin/flume master > /var/log/flume.log 2>&1 &
}
```

# 4. Package and install

- Each service is a self-contained JAR:

```
functions/configure_flumedemo_master.sh
functions/configure_flumedemo_node.sh
functions/install_flumedemo.sh
META-INF/services/org.apache.whirr.service.ClusterActionHandler
org/apache/whirr/service/example/FlumeMasterHandler.class
org/apache/whirr/service/example/FlumeNodeHandler.class
```

- Discovered using java.util.ServiceLoader facility
  - META-INF/services/org.apache.whirr.service.ClusterActionHandler:

```
org.apache.whirr.service.example.FlumeMasterHandler
org.apache.whirr.service.example.FlumeNodeHandler
```

- Place JAR in Whirr’s lib directory

# 5. Run

- Create a cluster spec file

```
whirr.cluster-name=flumedemo
whirr.instance-templates=1 flumedemo-master,1 flumedemo-node
whirr.provider=aws-ec2
whirr.identity=${env:AWS_ACCESS_KEY_ID}
whirr.credential=${env:AWS_SECRET_ACCESS_KEY}
```

- Then launch from the CLI

```
% whirr launch-cluster --config flumedemo.properties
```

- or Java

```
Configuration conf = new PropertiesConfiguration("flumedemo.properties");
ClusterSpec spec = new ClusterSpec(conf);
Service s = new Service();
Cluster cluster = s.launchCluster(spec);
// interact with cluster
s.destroyCluster(spec);
```

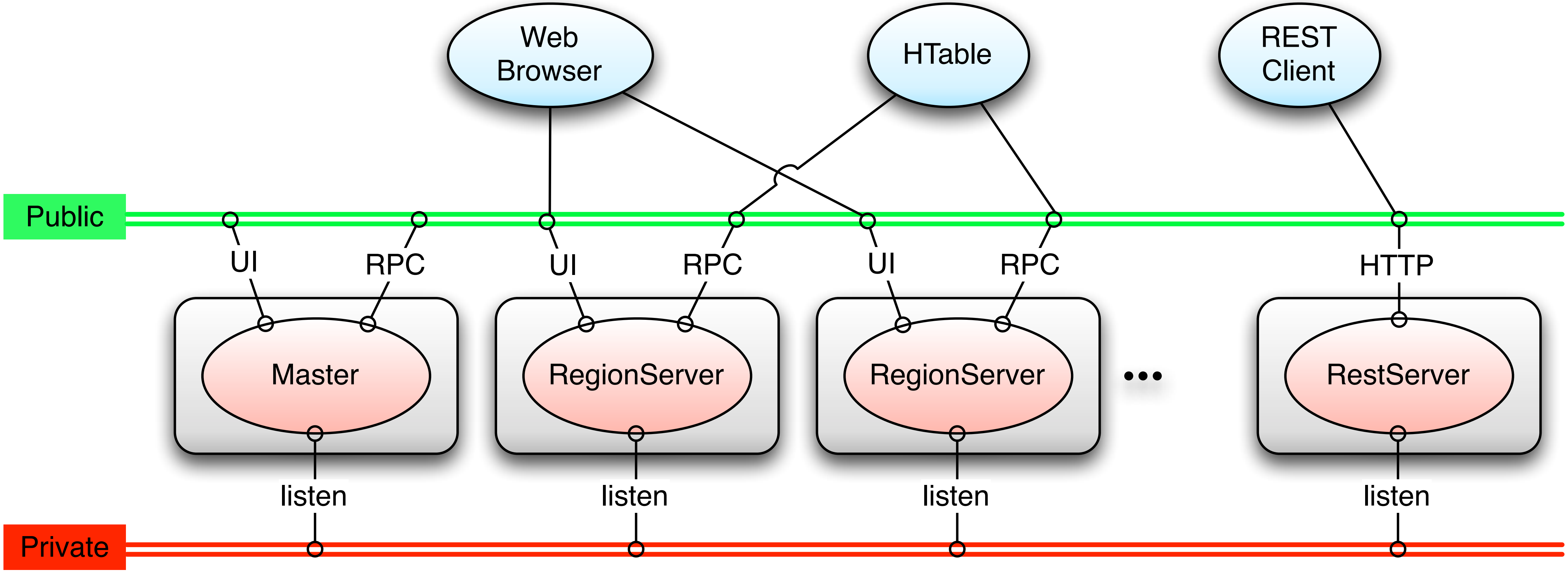
# Orchestration

- Instance templates are acted on independently in parallel
- Bootstrap phase
    - start 1 instance for the `flumedemo-master` role and run its bootstrap script
    - start 1 instance for the `flumedemo-node` role and run its bootstrap script
- Configure phase
    - run the configure script on the `flumedemo-master` instance
    - run the configure script on the `flumedemo-node` instance
- Note there is a barrier between the two phases, so nodes can get the master address in the configure phase

# Going further: Hadoop

- Service configuration
  - Flume example was very simple
  - In practice, you want to be able to override any service property
  - For Hadoop, Whirr generates the service config file and copies to cluster
  - E.g. `hadoop-common.fs.trash.interval=1440`
    - Sets `fs.trash.interval` in the cluster configuration
- Service version
  - Tarball is parameterized by `whirr.hadoop.tarball.url`

# Going further: HBase



# Going further: HBase

- Service composition

```
whirr.instance-templates=1 zookeeper+hadoop-namenode+hadoop-jobtracker+hbase-master,
5 hadoop-datanode+hadoop-tasktracker+hbase-regionserver
```

- HBase handlers will do the following in beforeConfigure():
  - open ports
  - pass in ZooKeeper quorum from `zookeeper` role
  - for non-master nodes: pass in master address
- Notice that the Hadoop cluster is overlaid on the HBase cluster

# Challenges

- Complexity
- Degrees of freedom
  - #clouds × #OS × #hardware × #locations × #services × #configs = a big number!
  - Known good configurations, recipes
  - Regular automated testing
- Debugging
  - What to do when the service hasn’t come up?
  - Logs
  - Recoverability

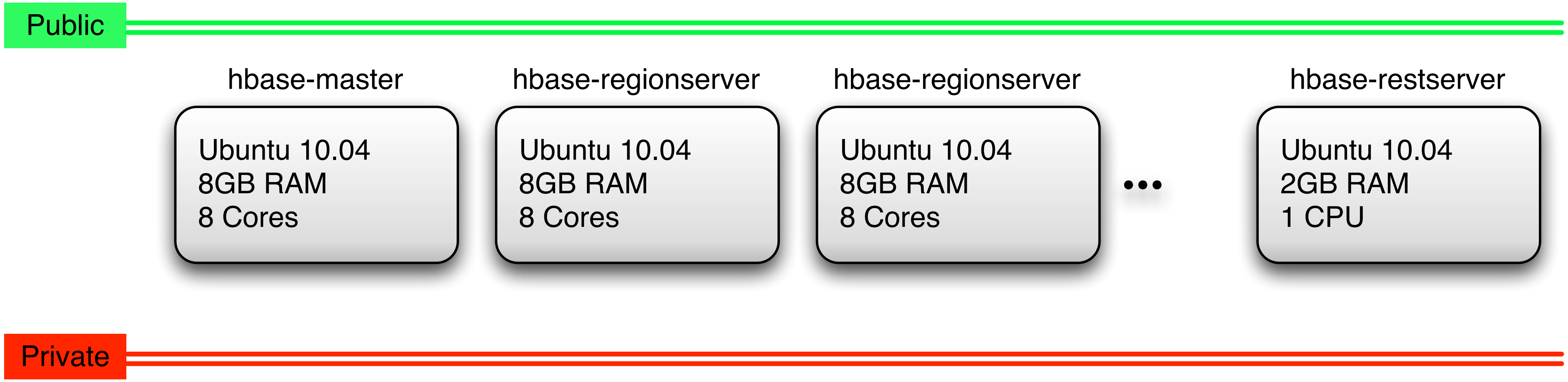
# What’s next?

- Release 0.4.0 coming soon
  - Evict failing nodes (WHIRR-167)
  - Locally-supplied scripts (WHIRR-225)
- More services
- More (tested) cloud providers
  - GoGrid!
- Pool provider/BYON
- Cluster resizing
- https://cwiki.apache.org/confluence/display/WHIRR/RoadMap

# Questions

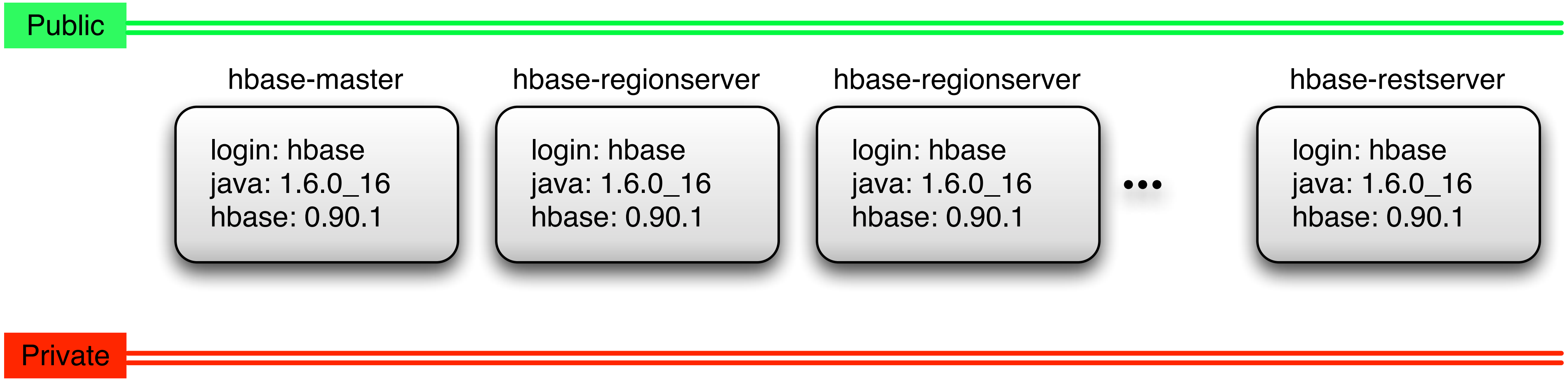
- Find out more at
  - http://incubator.apache.org/whirr
  - https://github.com/tomwhite/whirr-service-example
- Twitter: @tom_e_white

# provision

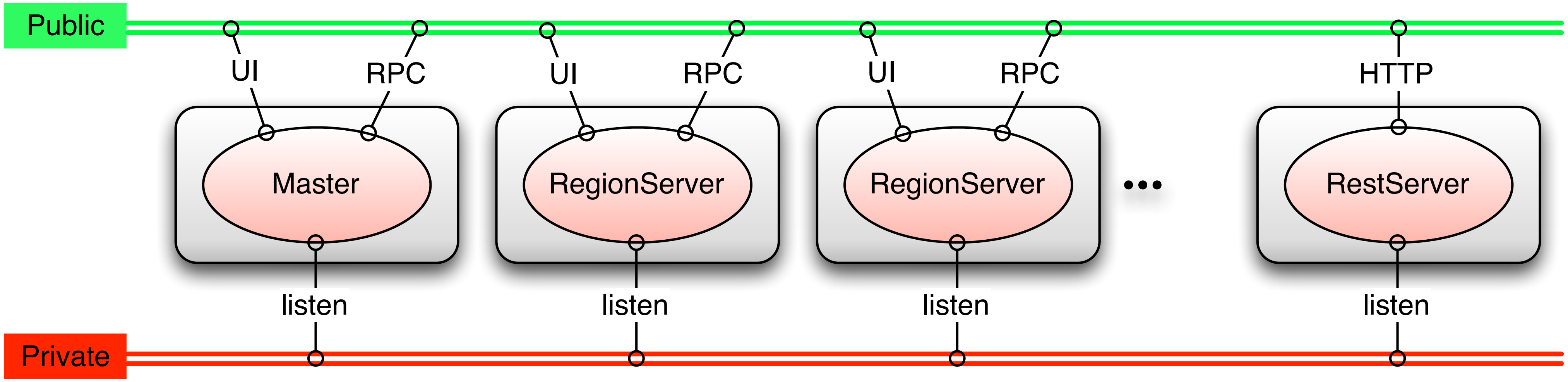


Public
hbase-master
Ubuntu 10.04
8GB RAM
8 Cores
hbase-regionserver
Ubuntu 10.04
8GB RAM
8 Cores
hbase-regionserver
Ubuntu 10.04
8GB RAM
8 Cores
...
hbase-restserver
Ubuntu 10.04
2GB RAM
1 CPU
Private

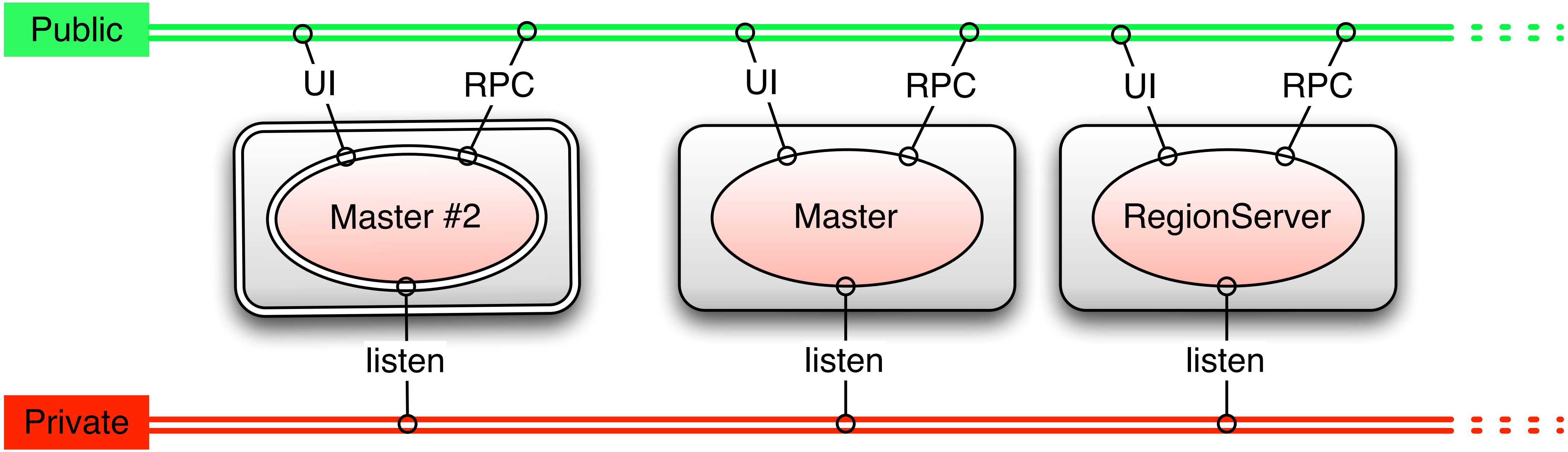
# install


Public
hbase-master
login: hbase
java: 1.6.0_16
hbase: 0.90.1
hbase-regionserver
login: hbase
java: 1.6.0_16
hbase: 0.90.1
hbase-regionserver
login: hbase
java: 1.6.0_16
hbase: 0.90.1
•••
hbase-restserver
login: hbase
java: 1.6.0_16
hbase: 0.90.1
Private

# configure


Public
UI
RPC
UI
RPC
UI
RPC
HTTP
Master
RegionServer
RegionServer
•••
RestServer
listen
listen
listen
listen
Private

# manage


Public
UI
RPC
UI
RPC
UI
RPC
Master #2
Master
RegionServer
listen
listen
listen
Private