

CometCloud-lite Installation Guide

CometCloud-lite includes basic CometCloud infrastructure and master/worker programming model. This tutorial explains how to download and setup CometCloud- lite and to run the given sample application.

1. System requirements

JDK 1.6 or above

Ports to open public for autonomic cloudbursts: 12340~12350 (not required for local run)

2. Using prepared EC2 image

1) Start an ec2 instance

- Image name: cometcloud-lite
- find image id starting with ami-
\$ ec2dim |grep cometcloud-lite
- start m1.small instance
\$ ec2run *imageID*
- get the instance IP address
\$ ec2din

2) Connect to the instance using putty

- Start putty.exe and write the IP address in Host Name (or IP address)
- Click SSH -> Auth -> Private key file for authentication -> Load cometcloud.ppk
- Userid/password to login the image: cometcloud/cometcloud

3) Working directory

- Directory tutorial includes Sample and lib directories.
- Sample includes all property files and scripts to run the sample master/worker application provided in CometCloud-lite master/worker programming model. Directory lib includes CometCloud executables and required libraries.

Note: if you use the given EC2 image, you can skip the following Download CometCloud-lite and Setup CometCloud-lite sections, and go to the directory tutorial/Sample/secure.

3. Download CometCloud-lite

```
$ svn checkout --username cometcloud-guest svn://cac.rutgers.edu/CometCloud-lite  
Password is cometcloud.
```

```
$ cp CometCloud-lite/Automate/application/trunk/conf/* your_working_directory
```

4. Setup CometCloud-lite

1) java CLASSPATH

Include all jar files from lib directory into java CLASSPATH

2) nodeFile

nodeAddress:number_of_processes

The process will be a master or a worker based on the role description in exceptionFile.

Example)

localhost:2

3) portFile

List of port numbers that masters or workers will use. This should have larger port numbers than the number of processes running on a single node.

Example)

5555

5556

5557

5558

4) exceptionFile

nodeAddress:port_number

comet.NodeType=process_role

If you don't describe roles here, it will become a worker as a default. Process roles are MASTER|WORKER|REQUEST_HANDLER. The request handler is to pick up tasks from the Comet space for isolated workers. Further details are in section 6.

Example)

localhost:5555

comet.NodeType=MASTER

This setup will run a master on localhost port 5555 and workers for others. In the above example, localhost port 5556 will become a worker.

5. Run the sample application

1) Start overlay control server. The overlay control server will receive messages to run masters or workers.

```
$ ./startOverlayServer.sh
```

or

```
$ java -cp $CLASSPATH tassl.automate.overlay.OverlayControlServer 4444
```

2) Start the application starter. This will start comet starter so as to run local and remote peers based on their roles.

```
$ ./startSample.sh
```

or

```
$ java -cp $CLASSPATH tassl.automate.programmodel.masterworker.sample.AppStarter  
-nodeFile nodeFile -portFile portFile -exceptionFile exceptionFile  
-propertyFile chord.properties -propertyFile squid.properties -propertyFile comet.properties  
-propertyFile sample.properties
```

This simple application in the package, `tassl.automate.programmodel.masterworker.simple`, is to sum a range of numbers. To change the number of tasks, set `numTask` in `sample.properties`, otherwise, 1000 will be the default number of tasks.

Note: when you run a large number of tasks and see the lack of JVM memory, increase java memory using `-Xms` or `-Xmx`.

6. Run the sample application using isolated (unsecured) workers

If you use `cometcloud-lite` EC2 image, go to the directory `tutorial/Sample/isolated`.

1) Setup one or more request handlers in `exceptionFile`

```
localhost:5555
```

```
comet.NodeType=MASTER
```

```
localhost:5556
```

```
comet.NodeType=REQUEST_HANDLER
```

Now a master will run on localhost port 5555 and a request handler on localhost port 5556.

Note that no secure worker runs.

2) Edit `RequestHandlerList`

This list should include request handler's hostname or IP address and the proxy will refer this list to forward workers' requests to.

Example)

```
localhost
```

3) Edit `comet.properties`

Set `IsolatedProxy` to your proxy hostname or IP address.

4) Start proxy

```
$ ./startProxy.sh
```

or

```
$ java -cp $CLASSPATH tassl.automate.application.node.isolate.RequestHandlerProxy
```

5) Start overlay control server.

```
$ ./startOverlayServer.sh
```

or

```
$ java -cp $CLASSPATH tassl.automate.overlay.OverlayControlServer 4444
```

6) Start the application starter

```
$ ./startSample.sh
```

or

```
$ java -cp $CLASSPATH tassl.automate.programmodel.masterworker.sample.AppStarter  
-nodeFile nodeFile -portFile portFile -exceptionFile exceptionFile  
-propertyFile chord.properties -propertyFile squid.properties -propertyFile comet.properties  
-propertyFile sample.properties
```

7) Start isolated workers

```
$ ./startIsolatedWorker.sh
```

or

```
$ java -cp $CLASSPATH tassl.automate.application.node.isolate.CloudBurstStarter  
-propertyFile comet.properties
```