

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

2) Connect to the instance

Userid is ubuntu and no password is required.

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 CometCloud-lite.

## 3. Download CometCloud-lite

### 3.1. Using svn tools

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

Password is cometcloud.

### 3.2 Using Eclipse

- 1) Select menu Help -> Install New Software..
- 2) Type [http://subclipse.tigris.org/update\\_1.6.x/](http://subclipse.tigris.org/update_1.6.x/) in Work with:
- 3) Click Select All button, then click Next
- 4) In your new java project, select File -> Import, select SVN and click Next
- 5) Select Use existing repository location, then click Next
- 6) Type svn://cac.rutgers.edu/CometCloud-lite in Url, username is cometcloud-guest and password is cometcloud.

7) Select src directory then finish.

### **3.3 Using Netbeans**

- 1) Select menu Team -> Subversion and follow the directions to install Subversion.
- 2) After installing Subversion, restart Netbeans.
- 3) Select menu Team -> Subversion -> checkout
- 4) Type `svn://cac.rutgers.edu/CometCloud-lite` in Repository Url
- 5) Type `cometcloud-guest` in user and `cometcloud` in password.
- 6) In your new java project, set src in the downloaded CometCloud-lite to your source directory.

## **4. Setup CometCloud-lite**

Example configuration files are located in CometCloud-lite/simple\_run.

```
$ cp CometCloud-lite/simple_run/* your_working_directory
```

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 matrix multiplication using secure workers in Master/Worker

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.

```
$ ./startMatrix.sh  
or  
$ java -cp $CLASSPATH tassl.automate.application.MatrixMultiplication.MatrixStarter -  
nodeFile nodeFile -portFile portFile -exceptionFile exceptionFile -propertyFile  
chord.properties -propertyFile squid.properties -propertyFile comet.properties -propertyFile  
matrix_multiplication.properties
```

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 matrix multiplication using isolated (unsecured) workers

1) Setup one or more request handlers in exceptionFile. Please refer exceptionFile.isolate.

```
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. Please refer comet.properties.isolate.

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

```
$ ./startMatrix.sh
```

or

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

7) Start isolated workers

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

or

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