

Seafile 4.1.2 instructions: (Ubuntu 12.04)

Note: All seafile commands must be run as root.

TO INSTALL SEAFILE:

On server machine:

```
clone duet GitHub  
sudo duet/SeaFileServer/SF-server-build-install.sh  
ls ~/SeaFileServer/seafile-data/storage/commits/
```

there should be one folder with a long name.

Log in to seahub by using

```
elinks http://<IP-ADDRESS>:8001
```

and enter your email/password. Then do

```
ls ~/SeaFileServer/seafile-data/storage/commits/
```

again and you should see two folders. The name of the new folder is the library ID, record this ID as you'll need it for the client configuration.

On client machine:

```
clone duet GitHub  
sudo duet/SeaFileClient/SF-client-build-install.sh  
sudo duet/SeaFileClient/SF-client-setup.sh
```

TO RECOMPILE SEAFILE AFTER MAKING CODE CHANGES:

On server machine:

```
sudo duet/SeaFileServer/SF-server-recompile-install.sh
```

When asked for folders for seafile setup, leave it blank and it will use the existing configuration in the default location.

Do not run `SF-server-nginx-setup.sh` again.

On client machine:

```
sudo duet/SeaFileClient/SF-client-recompile-install.sh
```

Do not run `SF-client-setup.sh` again.

Using Seafile with scripts

OVERVIEW OF HOW A SYNC TEST WORKS:

1. Start the client daemon, sync an empty folder on the client to the server then stop the client daemon. This time is not counted, to reduce the amount of the time taken during a test that is not related to inotify.
2. Generate a set of files in the folder that we previously synced, clear the cache so they are not still in memory when the sync runs. Start timing, start the client daemon and start recording performance data. When the sync is complete (seaf-cli status reports “synchronized”), stop timing and stop recording performance data.
3. Delete all of the files in the sync folder, wait for the sync to complete to ensure this repository on the server is empty. Stop the client daemon, restart the server processes and perform garbage collection on the server (otherwise seaf file server keeps a history of all the deleted files).

TO RUN REGULAR SYNC TEST ACROSS MANY SETS OF FILES:

On client machine run:

```
SF-server-remote-restart.exp  
automated-test.sh
```

NOTE: There will be an error about “511 Unknown Service” that will repeat multiple times then be replaced by “Checking...”, this is normal and is a byproduct of making the previously daemonized seaf-daemon and ccnet processes not run as daemons.

TO PRODUCE LOGS AND GPROF OUTPUT FROM SYNCING ONE SET OF FILES:

On client machine run:

```
SF-server-remote-restart.exp  
log-and-gprof.sh 1 <number-of-files> <size-of-files>
```

NOTE: There will be an error about “511 Unknown Service” that will repeat multiple times then be replaced by “Checking...”, this is normal and is a byproduct of making the previously daemonized seaf-daemon and ccnet processes not run as daemons.

This will generate a set of files into SeaFileLibraries using maketree.py and then sync them with the server, producing a full set of debug logs from seaf file in `~/ccnet/logs/<TIMESTAMP>` and gprof output in `SEAFILEDIR` (which is `~/` by default)

TO TEST TIME TAKEN TO CONFIRM A DIRECTORY HAS ALREADY BEEN SYNCED:

On client machine run:

```
SF-server-remote-restart.exp
```

```
automated-initialization-test.sh
```

NOTE: There will be an error about “511 Unknown Service” that will repeat multiple times then be replaced by “Checking...”, this is normal and is a byproduct of making the previously daemonized seaf-daemon and ccnet processes not run as daemons.

Using Seaf file manually

SERVER:

To start server:

```
cd ~/SeaFileServer/seaf file-server-4.1.2/  
sudo service nginx start  
./seaf file.sh start  
./seahub.sh start 8001
```

To stop server:

```
./seahub.sh stop  
./seaf file.sh stop  
sudo service nginx stop
```

seaf file.sh start starts the following processes:

```
~/SeaFileServer/seaf file-server-4.1.2/seaf file/bin/seaf file-controller  
~/SeaFileServer/seaf file-server-4.1.2/seaf file/bin/ccnet-server  
~/SeaFileServer/seaf file-server-4.1.2/seaf file/bin/seaf-server
```

seahub.sh start 8001 starts four instances of

```
python2.7 ~/SeaFileServer/seaf file-server-4.1.2/seahub/manage.py
```

CLIENT:

To use seaf-cli:

```
seaf-cli start  
seaf-cli sync -l <LIBRARY-ID> -s http://<IP ADDRESS>:<PORT> -d  
    <LOCAL LIBRARY TARGET> -u <USERNAME OR EMAIL> -p <PASSWORD>  
seaf-cli status  
seaf-cli stop
```

seaf-cli documentation: <https://seacloud.cc/group/3/wiki/seaf file-cli-manual/>

seaf-cli start starts the following processes:

/usr/bin/seaf-daemon

/usr/bin/ccnet

Troubleshooting:

A MESSAGE SAYING “CONNECTION REFUSED” usually means that either seafiler or seahub is not running on c157.

IF `SEAF-CLI STATUS` HANGS ON “CONNECTING SERVER”

Change the ccnet port in ccnet.conf on both machines to one that hasn't been used before.

On server, located at /my/install/directory/ccnet/ccnet.conf

On client, located at ~/.ccnet/ccnet.conf

IF `SEAF-CLI STATUS` HANGS ON STATUS “UNKNOWN”

Remove seaf-cli data in ~/.ccnet and ~/SeaFileClient/seafiler-cli/seafiler-client/ then run seaf-cli init again

IF NUMBER OF INOTIFY WATCHES AVAILABLE BECOMES AN ISSUE

Set the system max user watches higher, e.g.

```
sudo sh -c 'echo 100000 > /proc/sys/fs/inotify/max_user_watches'
```

for a temporary change or

```
sudo sh -c 'echo "fs.inotify.max_user_watches=30000" >>
/etc/sysctl.conf'
```

for a permanent change, then remove seaf-cli config files and re-run seaf-cli init. See

<https://github.com/haiwen/seafiler/issues/169>

The sync will slow down over time, this is caused by seafiler server archiving deleted files. Run ~/SeaFileServer/seafiler-server-4.1.2/seaf-gc.sh on c157 to remove these files and fix this problem. SF-server-remote-restart.exp will run this from c158, as well as restarting the server using restart-seafiler-server.sh. **Note that the existing test scripts already do this.**

For issues not specifically mentioned here, first make sure to check for any seafiler-related processes that haven't been closed by the normal scripts and kill them. This can happen when seaf-cli start is done more than once in a row without seaf-cli stop in between.

If none of this solves the issue, the support forum is found at <https://forum.seafiler-server.org/>