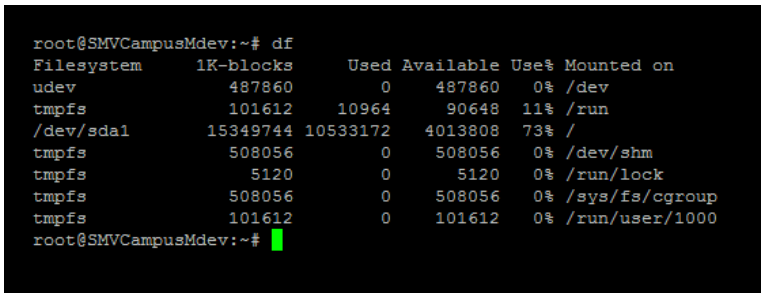
Connect Layer - Managing Hard Disk Space

- **Product:** campusM
- **Operating system:** iOS, Android

Typically Connect Layer servers are dedicated to campusM; we don't store any data on these servers. There are exceptions (e.g. servers which also host databases for integrations, or other apps for the University) but this document covers the usual case of the campusM app being the sole use of a server.

Diagnosis

The command `df` (stands for "disk filesystem") shows how much space remains on each partition on a linux server. It's a simple command to use and is worth doing as a precaution whenever you visit a server, particularly if there are any issues reported.



A typical output from `df`. Note that `df -h` will list the storage amounts in human readable format instead of bytes.

Of particular interest are the two columns on the right. "`Use%`" shows the percentage of the partition is used. "`Mounted on`" shows the area of the file system that is mounted on a particular partition (as opposed to the physical location, which is show under "`Filesystem`", but not particularly relevant to our clearing disk space).

The exact partition used to house the campusM files varies a great deal. Often it's `/var` or `/` but there are no hard and fast rules, and different files may be spread across more than one mounted drive.

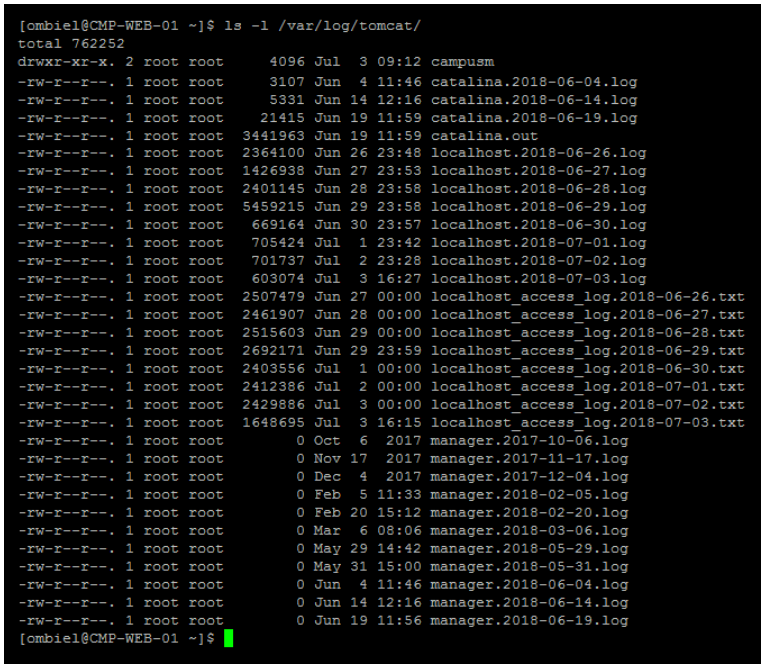
If a partition shows high usage, it may need some space clearing. If it is showing 100% use, it absolutely will need space clearing; and may be the cause of any issues reported since it would prevent campusM from handling any traffic at all with the inability to write log file activity.

Clearing Space

Log Files

Log files are the most likely cause of disk space issues. If logs are not cleared regularly, they *will* eventually fill up the disk.

The exact content of Tomcat's log directory will vary between servers but will have a familiar structure.



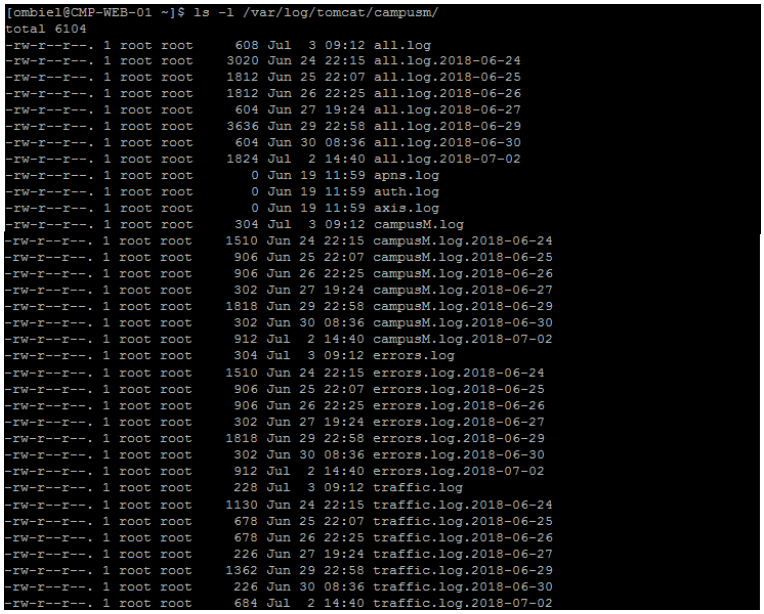
A Typical Tomcat logs directory.

The Tomcat logs directory usually contains catalina.out, localhost, localhost_access and manager, with several date-stamped archive files of each. These contain log messages written to stdout and stderr (in Java terms, outputs such as `System.out.println` and `e.printStackTrace`).

While developers should be using log4j, in practice these files can still gather a lot of logs, and will grow over time.

We do not need to retain many archived log files. When an error is reported, it will usually be present in the current, or previous day's logs. It's very rare for us to need to look at anything older. With this in mind, it is considered safe to delete any log files more than a week old (although the standard log rotate recommendation in the typical setup is to rotate anything older than two weeks.

There will also be one or more subdirectories within the base logs directory. Sometimes a subdirectory will have the same name as the app's WAR file; either"campusM" (older) or "campusm" (newer) depending on the generation of Connect Layer deployment. Because Linux directory names are case-sensitive "campusm" and "campusM" may exist in the same logs directory.



```
[ombiel@CMP-WEB-01 ~]$ ls -l /var/log/tomcat/campusm/
total 6104
-rw-r--r--. 1 root root     608 Jul  3 09:12 all.log
-rw-r--r--. 1 root root    3020 Jun 24 22:15 all.log.2018-06-24
-rw-r--r--. 1 root root    1812 Jun 25 22:07 all.log.2018-06-25
-rw-r--r--. 1 root root    1812 Jun 26 22:25 all.log.2018-06-26
-rw-r--r--. 1 root root     604 Jun 27 19:24 all.log.2018-06-27
-rw-r--r--. 1 root root    3636 Jun 29 22:58 all.log.2018-06-29
-rw-r--r--. 1 root root     604 Jun 30 08:36 all.log.2018-06-30
-rw-r--r--. 1 root root    1824 Jul  2 14:40 all.log.2018-07-02
-rw-r--r--. 1 root root       0 Jun 19 11:59 apns.log
-rw-r--r--. 1 root root       0 Jun 19 11:59 auth.log
-rw-r--r--. 1 root root       0 Jun 19 11:59 axis.log
-rw-r--r--. 1 root root     304 Jul  3 09:12 campusM.log
-rw-r--r--. 1 root root    1510 Jun 24 22:15 campusM.log.2018-06-24
-rw-r--r--. 1 root root     906 Jun 25 22:07 campusM.log.2018-06-25
-rw-r--r--. 1 root root     906 Jun 26 22:25 campusM.log.2018-06-26
-rw-r--r--. 1 root root     302 Jun 27 19:24 campusM.log.2018-06-27
-rw-r--r--. 1 root root    1818 Jun 29 22:58 campusM.log.2018-06-29
-rw-r--r--. 1 root root     302 Jun 30 08:36 campusM.log.2018-06-30
-rw-r--r--. 1 root root     912 Jul  2 14:40 campusM.log.2018-07-02
-rw-r--r--. 1 root root     304 Jul  3 09:12 errors.log
-rw-r--r--. 1 root root    1510 Jun 24 22:15 errors.log.2018-06-24
-rw-r--r--. 1 root root     906 Jun 25 22:07 errors.log.2018-06-25
-rw-r--r--. 1 root root     906 Jun 26 22:25 errors.log.2018-06-26
-rw-r--r--. 1 root root     302 Jun 27 19:24 errors.log.2018-06-27
-rw-r--r--. 1 root root    1818 Jun 29 22:58 errors.log.2018-06-29
-rw-r--r--. 1 root root     302 Jun 30 08:36 errors.log.2018-06-30
-rw-r--r--. 1 root root     912 Jul  2 14:40 errors.log.2018-07-02
-rw-r--r--. 1 root root     228 Jul  3 09:12 traffic.log
-rw-r--r--. 1 root root    1130 Jun 24 22:15 traffic.log.2018-06-24
-rw-r--r--. 1 root root     678 Jun 25 22:07 traffic.log.2018-06-25
-rw-r--r--. 1 root root     678 Jun 26 22:25 traffic.log.2018-06-26
-rw-r--r--. 1 root root     226 Jun 27 19:24 traffic.log.2018-06-27
-rw-r--r--. 1 root root    1362 Jun 29 22:58 traffic.log.2018-06-29
-rw-r--r--. 1 root root     226 Jun 30 08:36 traffic.log.2018-06-30
-rw-r--r--. 1 root root     684 Jul  2 14:40 traffic.log.2018-07-02
```

The content of the subdirectories also follows a similar pattern, with a number of types of log, and a number of date-stamped archives of each. In order to manage disk space effectively we need to manage these subdirectories as well as the base logs directory.

WAR Files

It's not uncommon to leave a few backup WAR files in the home directory on the dev server, typically with fewer on a live server. These files are in the region of 30-50Mb. Unless there's a very good reason to keep them, we don't usually need to keep any more than the active WAR files (and none of the backup WAR files from past deployments). These are good candidates for deletion.

WAR files in the webapps directory should be kept to an absolute minimum. In addition to needlessly taking up disk space, extra WAR files deployed as webapps will consume memory allocated to Tomcat. Unless there is a very good reason there should be no more than 1-2 campusM WAR file/s here.

Managing Disk Space

Immediate Resolution

Faced with an overflowing log directory, the first solution should be to delete some old logs. Begin with any from previous years (running, e.g. `rm -f /var/log/tomcat/logs/*2017*` will quickly clear all logs from 2017). Repeat this, deleting any log files older than, say, a week (unless you ohave a compelling reason to keep anything older, in which cases it would be recommended to archive those older logs and offload them to a different disk).

Repeat in any subdirectories that are present.

Log Rotation

Prevention is better than cure, and we can prevent log files consuming all disk space with log rotation. This means old files are deleted regularly, so only the latest logs are retained, and disk space is preserved as recommended in our Connect layer setup guide log rotate recommendation and additional details below.

Logging Properties Files

Part of defining log rotation is through the log4j.properties file (in the Tomcat conf directory or deployed in the WAR file).

The logging behaviour of the subdirectories are defined in this file.

```
# LOGFILE is set to be a File appender using a PatternLayout.
log4j.appender.LOGFILE=org.apache.log4j.DailyRollingFileAppender
log4j.appender.LOGFILE.File=${catalina.base}/logs/campusM/all.log
log4j.appender.LOGFILE.DatePattern='.'yyyy-MM-dd
log4j.appender.LOGFILE.layout=org.apache.log4j.PatternLayout
log4j.appender.LOGFILE.layout.ConversionPattern=%d [%t] %-5p %c %x - %m%n
log4j.appender.LOGFILE.MaxBackupIndex=7
```

The last line: `log4j.appender.LOGFILE.MaxBackupIndex=7` means only seven days' log files will be retained. Each log file defined in log4j.properties must have a similar line.

Logrotate

Logrotate is a linux system utility that manages the automatic rotation and compression of log files. It is used to manage the logs in the Tomcat base logs directory.

Logrotate is installed by your package manager in the usual way:

`aptitude update && aptitude install logrotate` (Debian and Ubuntu)

`yum update && yum install logrotate` (CentOS, Red Hat, etc.)

(The client is responsible for installing software.)

The main config for logrotate is usually found in `/etc/logrotate.d`, with specific config for individual applications (in our case Tomcat) within `/etc/logrotate.d`. In this directory edit, or create a file called tomcat.

In there you will need to create a config entry for each log directory that campusM creates. This example manages the all.log file. Similar entries will be required for campusM.log and others:

```
/var/log/tomcat/campusm/all.log {
    missingok
    notifempty
    delaycompress
    sharedscripts
    create 0644 tomcat:tomcat
    endscript
}
```

There is a good deal more information in the logrotate man file: `man logrotate`

Logging Levels

campusM will always produce log files; supporting the app would be near-impossible without them. We can manage the size of logs produced, though. Levels are defined in logging.properties and log4j.properties(both in the Tomcat conf directory). The levels are as follows (in order from the most to least verbose):

- ALL
- DEBUG
- INFO
- WARN
- ERROR
- FATAL
- OFF

It is recommended to user INFO or WARN on the Production Connect Layer and DEBUG on the Sandbox Connect Layer.

- **Article last edited:** DD-Mmm-YYYY