**MongoDB Linux File Permissions**

When installing MongoDB on a Linux system you may have problems in starting the mongod process. It may start using sudo mongod but not mongod. You may be getting file permission errors.

When you log into a Linux system there is a root user and other user level accounts. The following assumes the user is called ec2-user. This is the default user for Amazon instances.

Never log in as root and install s/w under root. This complicates the recovery of the system for other users.

In the examples below MongoDB was installed from a repository using the command:

sudo yum install mongo-10gen mongo-10gen-server

this should follow exactly the instructions given by mongo here: <http://docs.mongodb.org/manual/tutorial/install-mongodb-on-redhat-centos-or-fedora-linux/>

Yum will not only download and install the binary distributions of Mongo but will install the log and db directories in the correct places.

There are documents (<http://docs.fedoraproject.org/en-US/Fedora_Draft_Documentation/0.1/html/RPM_Guide/> ) which describe the Linux convention of where to install additional configuration files and directories for database storage. These specifications define what a RPM package does(same for yum). When software is converted from a tar file into an RPM package it must adhere to an ***extra set of deployment and installation conventions*** which you don’t get with a tar or tar.gz or zip file.

Universally, conf files go under /etc/COMPONENT\_NAME and log files under /var/log and db files under /var .

All of the NOSQL databases including Hadoop, HBase, etc.. follow this convention.

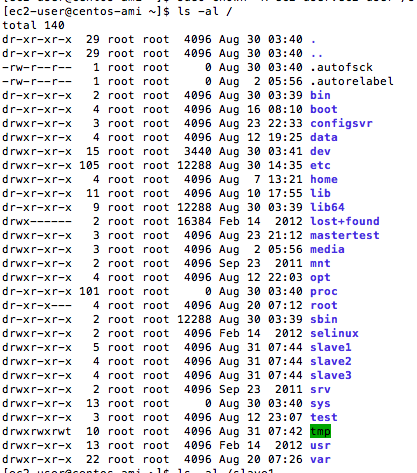
If you download the mongo binary distribution yourself and store it under /home/ec2-user and create separate directories for configuration under /home/ec2-user/mongo you won’t be following convention.

There is tremendous emphasis on using configuration conventions to keep people from getting confused during system upgrades and updates.

MongoDB requires a directory to store the contents of the database and a file to store the logs.

For this example we choose to put the database in /slave1/db and the logs in /slave1/log.

The command sudo mkdir –p /slave1/db creates the directory but the owner is root. All the folders at the root directory / are owned by root.



We need to change both the owner and group id of the file to the current user who is going to run mongodb; in this case ec2-user.

We need to fix this by:

**>sudo chown –R ec2\_user:ec2\_user /slave1/db**

This changes the owner and group to ec2-user for the directory /slave1/db.

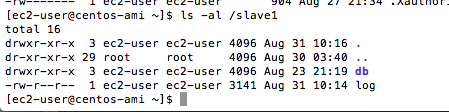
If we want to change /slave1 also:

**>sudo chown –R ec2\_user:ec2\_user /slave1**

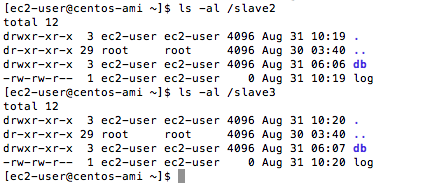
Create the log file using:

**>touch /slave1/log**

**>sudo chown ec2-user:ec2\_user /slave1/log**

****

**Verify all the slave directories are the same owner and group.**

****

Verify same for config server directories!!

