***Finesse & Jenkins Installation/Migration on Linux***

***Fitnesse Installation:***

**1. Create a user and group for finesse (optional)**

I didn't do this because I wanted tomcat, Jenkins and finesse all running as the same user. Call it laziness to avoid any permissions classing but it doesn't change the process that you need to create or choose what user you're going to make it run as. Don't make it run as your user or root!

**2. Download the jar file and place it in /usr/share/finesse**

Make the folder too of course. It can belong to root as long as the finesse user has read access

**3. Create the folder to run in at /var/lib/finesse**

Fitnesse user needs write permissions here so may as well make it the owner. This is where FitNesseRoot is going to end up and also where you're going to want any libraries and classes it can include. I found finesse isn't very good at using absolute paths when doing includes and also doesn't handle spaces. Something to look out for!

**4. Create the folder to store finesse logs /var/log/finesse**

Make sure the finesse user can write

**5. Create the below file as /etc/init/fitnesse.conf**

description "finesse: Fitnesse acceptance testing framework"

author "geoff@warmage.com"

start on (local-filesystems and net-device-up IFACE!=lo)

stop on runlevel [!2345]

env USER="finesse"

env GROUP="finesse"

env FITNESSE\_LOG="/var/log/finesse"

env FITNESSE\_ROOT="/var/lib/finesse"

env HTTP\_PORT=8081

env JAVA\_OPTS=""

env JAVA\_HOME="/usr/lib/jvm/default-java"

#limit nofile 8192 8192

pre-start script

test -f $FITNESSE\_ROOT/fitnesse.jar || { stop ; exit 0; }

end script

script

FITNESSE\_ARGS="-p $HTTP\_PORT -l $FITNESSE\_LOG"

exec daemon --name=finesse --inherit --chdir=$FITNESSE\_ROOT \

--output=$FITNESSE\_LOG/fitnesse-output.log --user=$USER \

-- $JAVA\_HOME/bin/java $JAVA\_OPTS -jar fitnesse.jar $FITNESSE\_ARGS

end script

6. Link from /etc/init.d/finesse to /lib/init/upstart-job

sudo ln -s /lib/init/upstart-job /etc/init.d/finesse

Make sure it's executable

sudo chmod +x /etc/init.d/finesse

nearly there now

**7. Setup run levels for finesse to run with**

I used sudo sysv-rc-conf -P to see the run levels and turned on 2, 3, 4 & 5.

References:

<http://iflifewereagame.blogspot.com/2012/05/setting-up-fitnesse-on-ubuntu-in-7.html>

**Jenkins Migration**

1. The migration step by step:
2. Create the new Jenkins host machine. Install Jenkins and make sure it runs. In my case I also made sure the /etc/default/Jenkins file is copied as it contained some specific configuration. We are using chef so I made sure this file was managed via chef. Also I made sure the apache virtual host configuration file for Jenkins was managed.
3. Stop Jenkins service on the new host.
4. Run Rsync as described above while the old Jenkins is still running
5. Start Jenkins service on the new host but put it in “quiteDown” mode to prevent and Jobs from being processed. As the old Jenkins is still active it could be problematic if the same Jenkins jobs are being triggered in parallel. To put Jenkins into “quiteDown” mode just call the following url: https://<your\_jenkins>/quietDown. This step is to verify that Jenkins correctly starts on the new host. See if it can connect to all the slaves (if you have any).
6. When done you can stop Jenkins on the new host again
7. Stop Jenkins on the old host (Now the downtime begins)
8. Run Rsync for a second time. This should be quite fast. In my case it took around a minute.
9. When Rsync is finished start Jenkins on the new host (End of downtime)
10. Adjust DNS Record so that it points to the new Jenkins host

***References:***

<https://staxmanade.com/2015/01/how-to-migrate-a-jenkins-job-to-new-jenkins-server/>