How to set up git /conf collect in linux from junos

1. Make sure your Linux client Is connected to the internet.
   (You can cheat by going into the settings in VMware and set the interface to NAT.)
   However I would recommend you to make this work beforehand.

2. Open your linux client, and run the following command, to make sure you're able to ssh between the client and the router.
   sudo apt-get install openssh-server

```
bjorn@ubuntu:~$ sudo apt-get install openssh-server
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  ncurses-term openssh-sftp-server ssh-import-id
Suggested packages:
  ssh-askpass rssh molly-guard monkeysphere
The following NEW packages will be installed:
  ncurses-term openssh-server openssh-sftp-server ssh-import-id
0 upgraded, 4 newly installed, 0 to remove and 167 not upgraded.
Need to get 636 kB of archives.
After this operation, 5,145 kB of additional disk space will be used.
Do you want to continue? [Y/n]
```

   Press enter to continue – then the download and installation will execute.

3. Next!
   Connect to your router.
   This can be done, by using the application we just downloaded "SSH"
   To find out what to connect to, write the following in the command prompt:
   ifconfig
   And then it should look something like this:

```
bjorn@ubuntu:~$ ifconfig
ens33     Link encap:Ethernet  HWaddr 00:0c:29:06:75:0b
          inet addr:192.168.94.10  Bcast:192.168.94.255  Mask:255.255.255.0
          inet6 addr: fe80::bd3b:8ac4:44d1:110/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:7405 errors:0 dropped:0 overruns:0 frame:0
          TX packets:4936 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:6407089 (6.4 MB)  TX bytes:530621 (530.6 KB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:3054 errors:0 dropped:0 overruns:0 frame:0
          TX packets:3054 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:223144 (223.1 KB)  TX bytes:223144 (223.1 KB)
```

   From inet – you will have the ipv4 address of the subnet you're on, typically the interface/router is set to .1 – so in my case it would be 192.168.94.1 that I will be connecting to.

Now in order to connect, simply write the following:
ssh root@192.168.94.1

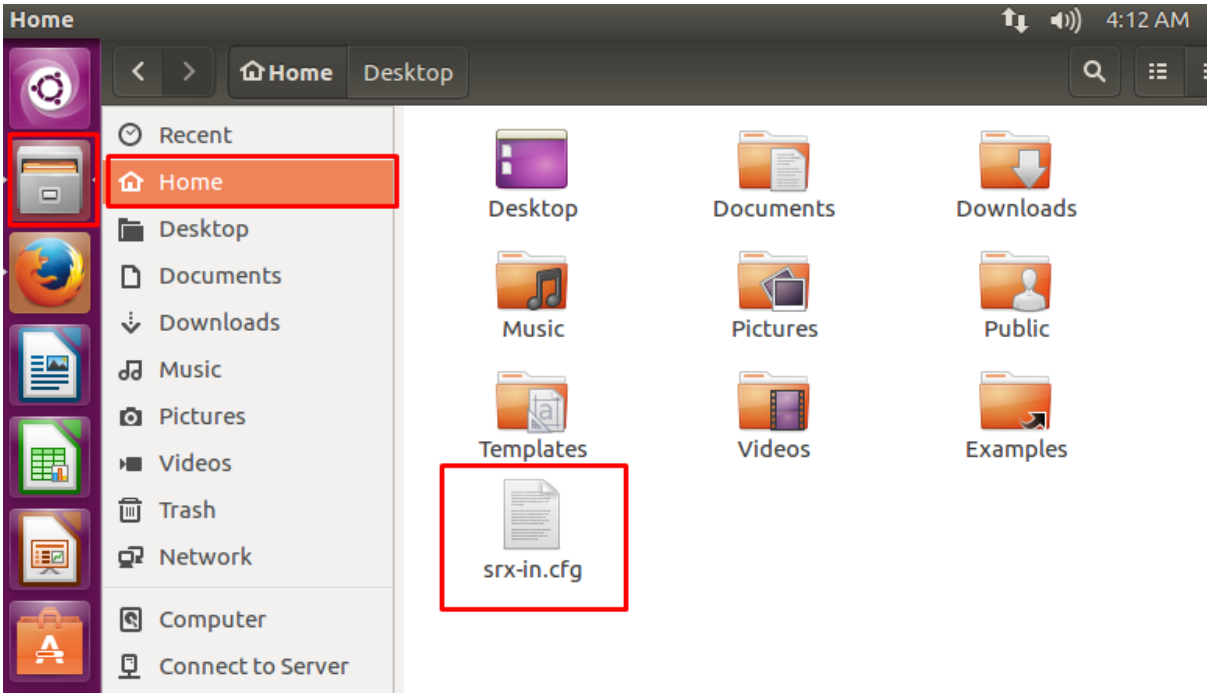
```
bjorn@ubuntu:~$ ssh root@192.168.94.1
Password:
--- JUNOS 12.1X47-D15.4 built 2014-11-12 02:13:59 UTC
root@SRX-IN% cli
root@SRX-IN> Edit
```

Go to “edit” within the router as shown above.

4. Time to get the configuration file of the router!
   Write the following line, inside the router: (of course insert your own computers name and ip address, as im assuming it’s not called bjorn)
   save scp://bjorn@192.168.94.10:.//home/bjorn/srx-in.cfg
   Without my own setup:
   save scp://PCUSER@PCIPADDRESS:.//home/PCUSER/srx-in.cfg

```
[edit]
root@SRX-IN# save scp://bjorn@192.168.94.10:.//home/bjorn/srx-in.cfg
The authenticity of host '192.168.94.10 (192.168.94.10)' can't be established.
ECDSA key fingerprint is 21:59:ad:65:30:04:b7:17:3d:53:83:ec:0e:11:dd:7c.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.94.10' (ECDSA) to the list of known hosts.
bjorn@192.168.94.10's password:
tempfile                                      100% 4634     4.5KB/s   00:00
Wrote 186 lines of configuration to 'scp://bjorn@192.168.94.10:.//home/bjorn/srx
-in.cfg'

[edit]
```

It should then place itself under home in your linux machine as seen here:



5. INSTALL GIT ON YOUR LINUX MACHINE
   write the following command:
   sudo apt-get install git git-gui

```
bjorn@ubuntu:~$ sudo apt-get install git git-gui
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  git-man gitk liberror-perl
Suggested packages:
  git-daemon-run | git-daemon-sysvinit git-doc git-el git-email git
  git-arch git-cvs git-mediawiki git-svn
The following NEW packages will be installed:
  git git-gui git-man gitk liberror-perl
0 upgraded, 5 newly installed, 0 to remove and 167 not upgraded.
Need to get 4,099 kB of archives.
After this operation, 29.0 MB of additional disk space will be used
Do you want to continue? [Y/n]
```

   To Continue press enter.

6. Now that git has been installed we can run some git commands.

   Next just write the following commands to create, and check if there is anything in a folder.
   To make a folder = “Make directory name: configs”:
   mkdir configs
   To go to the folder (change directory)
   cd configs
   Now that we are in the folder we run the following command to make it into a Git repository.
   git init
   To check if there is anything in the folder(Nothing should be shown) write:
   ls

7. Now go back to the router, go to edit and write the following:
   save scp://bjorn@192.168.94.10://home/bjorn/configs/srx-in.cfg

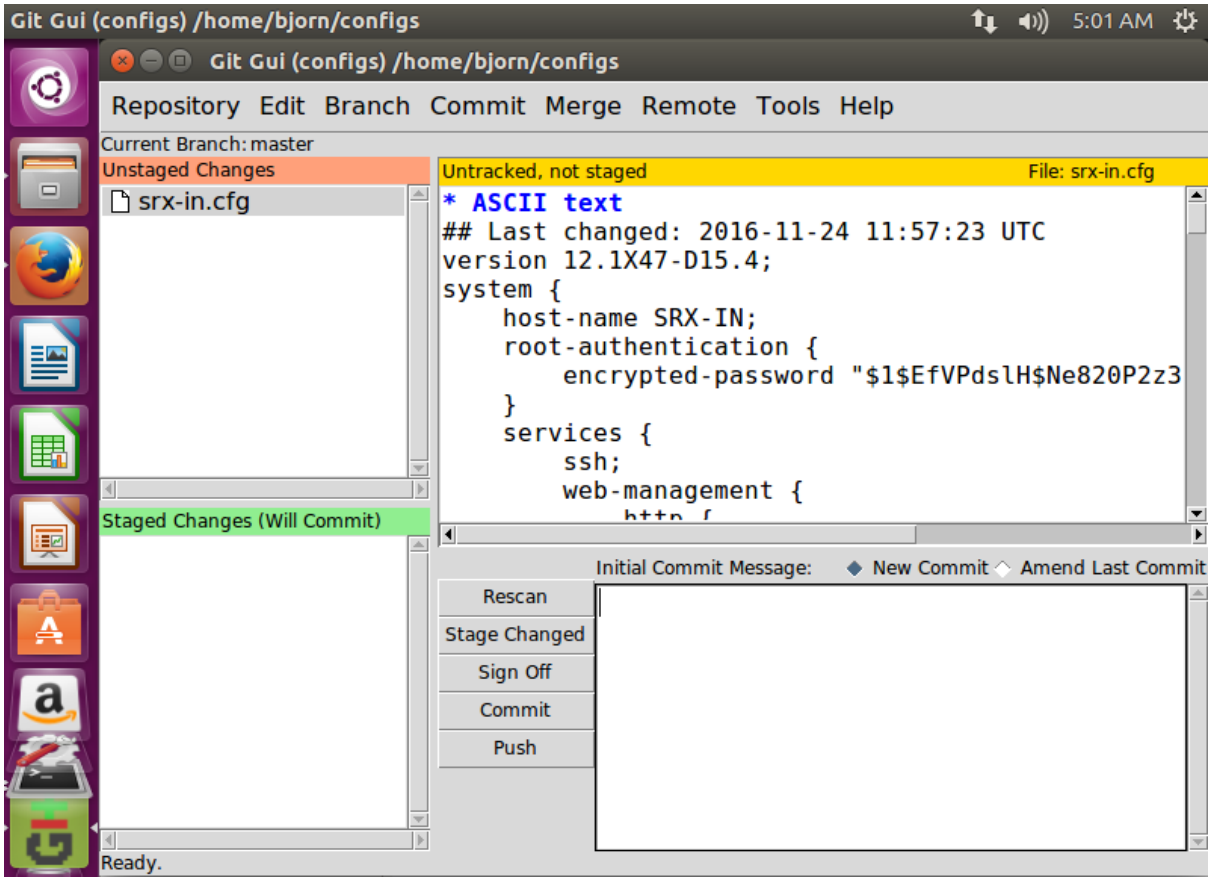
```
[edit]
root@SRX-IN# ...8.94.10://home/bjorn/configs/srx-in.cfg
bjorn@192.168.94.10's password:
tempfile                                      100% 4634     4.5KB/s   00:00
Wrote 186 lines of configuration to 'scp://bjorn@192.168.94.10://home/bjorn/conf
igs/srx-in.cfg'

[edit]
root@SRX-IN#
```

   This will save the file to the new folder (git repository) we just created.

8. In order to see your repository, in a flashy graphical interface write the following command in linux:

   git gui

Then this should appear:



To get an overview over your files in git write the following command:
git status

9. To add the file to git, write:
git add srx-in.cfg
Simply the name of the file..
Then you can see the that a new file has been added by simply writing:
git status
again.

```
bjorn@ubuntu:~/configs$ git add srx-in.cfg
bjorn@ubuntu:~/configs$ git status
On branch master

Initial commit

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)

        new file:   srx-in.cfg

bjorn@ubuntu:~/configs$
```

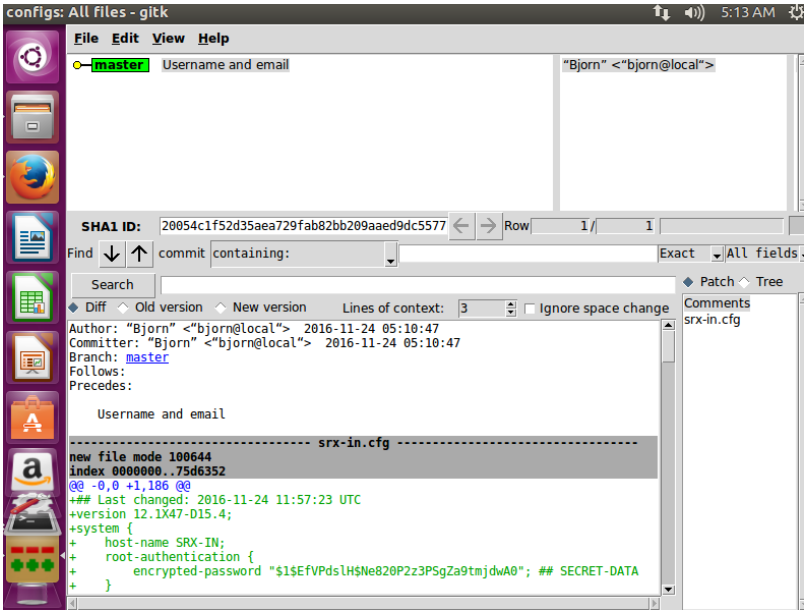
In order to commit all your changes in git, write:
git commit
You should then get the following screen:

```
bjorn@ubuntu:~/configs$ git commit

*** Please tell me who you are.

Run

  git config --global user.email "you@example.com"
  git config --global user.name "Your Name"

to set your account's default identity.
Omit --global to set the identity only in this repository.

fatal: unable to auto-detect email address (got 'bjorn@ubuntu.(none)')
bjorn@ubuntu:~/configs$
```

Then we add the things it asks for, in order to validate our identity for the program.
git config –global user.email “bjorn@local”
git config –global user.name “Bjorn”
git commit
When committing it will ask you to write a reason for the changes, simply write a comment, press control X then yes and then hit enter.

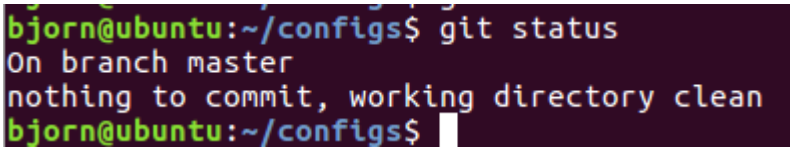
```
bjorn@ubuntu:~/configs$ git config --global user.email “bjorn@local“
bjorn@ubuntu:~/configs$ git config --global user.name “Bjorn”
bjorn@ubuntu:~/configs$ git commit
[master (root-commit) 20054c1] Username and email
 1 file changed, 186 insertions(+)
 create mode 100644 srx-in.cfg
bjorn@ubuntu:~/configs$
```

10. You can then write:
    gitk
    in order to see an overview of your commit changes, and what you have changed.



Just press “x” in the corner to close it again.
we can then yet again check for the git status:

git status
to see if we are missing to commit anything:



In my case everything has been commit, or nothing has been changed without commit.

Write:
gitk
to see what changes are missing to be commited, if you're missing anything, it gives you a graphical overview.