***Steps to Set Up a Two-Way Synchronization (Mirror) for Git Repositories***

Prepare Linux A **192.168.0.110** and Linux B **192.168.0.108** Repositories

**Install GIT on Both A and B machines:**

Command: **sudo apt-get install git -y**

**Create Global Identity on Both A and B machines**:

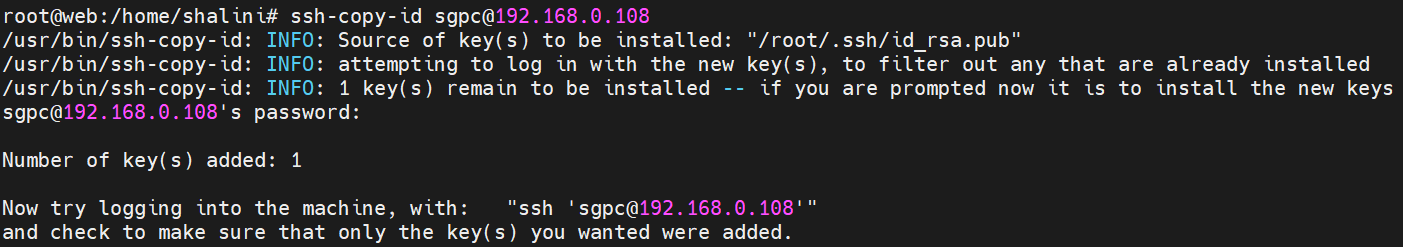
git config --global user.email "shalini@omni-techsolutions.com"

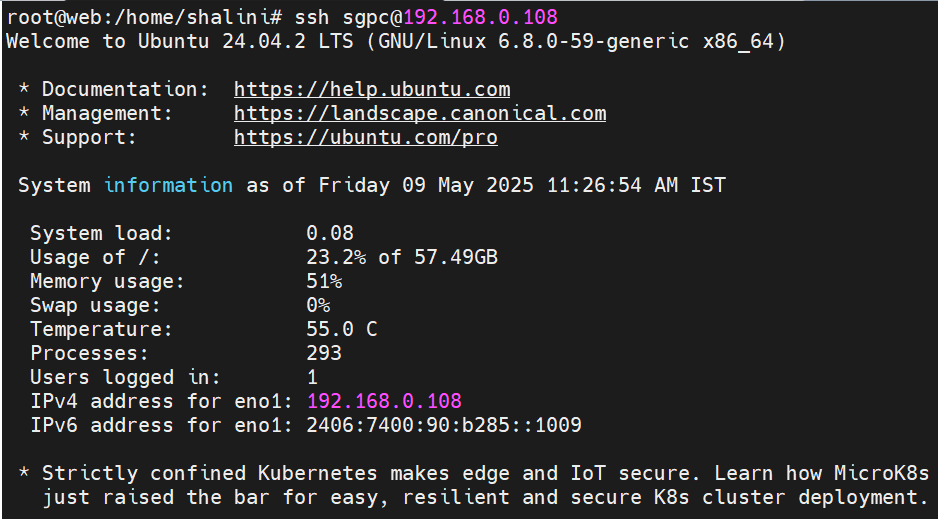
git config --global user.name "shalini"

Steps to Set Up Password-less **SSH Login** and Use **rsync** on both A & B machines:

**Generate an SSH Key Pair on Linux A**: ***192.168.0.110***

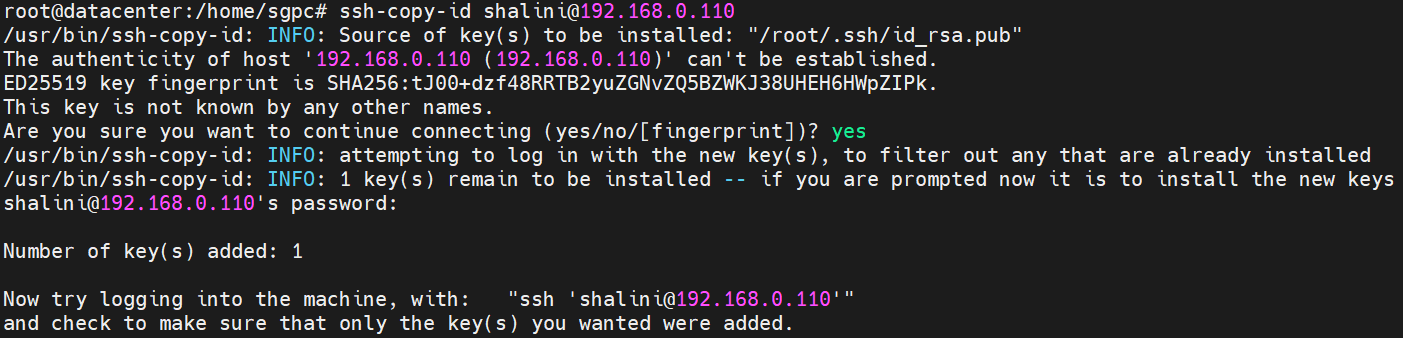
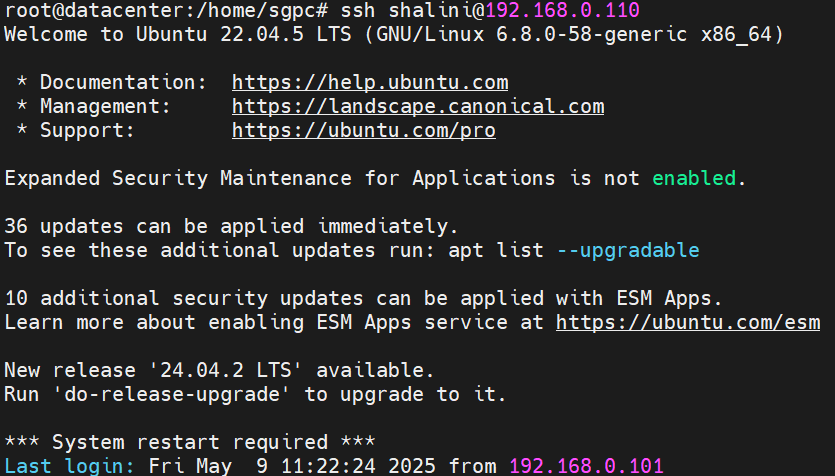
* Command: **ssh-keygen -t rsa -b 2048**
* Copy public key to Linux B: **ssh-copy-id sgpc@192.168.0.108 (**key for Linux B)
* Pass: **sgpcdata123**
* Test Password-less SSH Login**: ssh sgpc@192.168.0.108**





**Generate an SSH Key Pair on Linux B**: ***192.168.0.108***

* Command: **ssh-keygen -t rsa -b 2048**
* Copy public key to Linux B: **ssh-copy-id shalini@192.168.0.110 (**key for Linux A**)**
* Test Password-less SSH Login**: ssh shalini@192.168.0.110**

**On Linux A:** 192.168.0.110

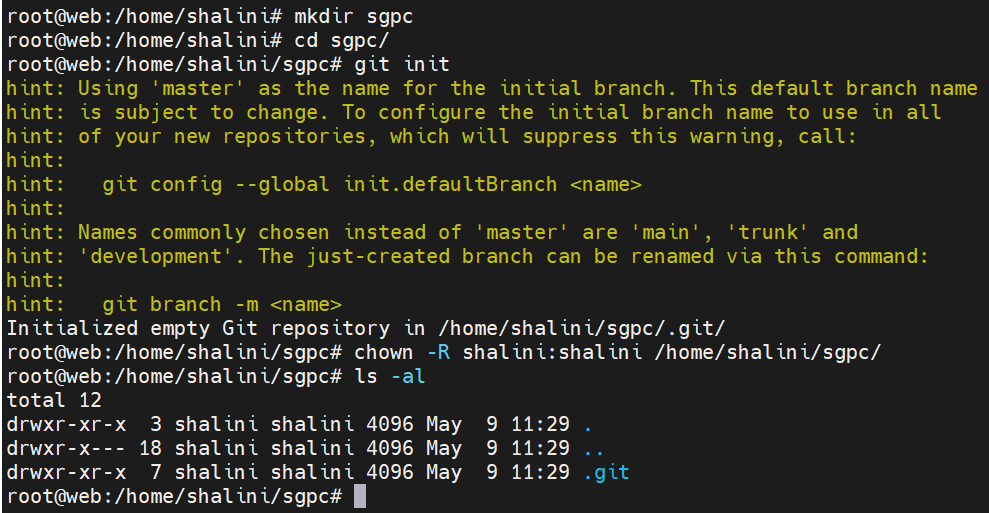
**User:** shalini

**Pass:** shalini123

**Step1:** Create the directory and initialize the repository

**Commands:**

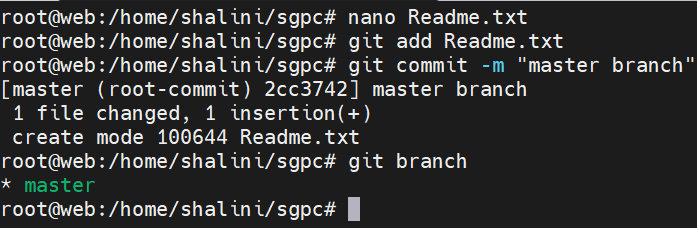
* + - cd /home/shalini
    - mkdir sgpc
    - cd sgpc
    - git init
    - chown -R shalini:shalini /home/shalini/sgpc/



**Step2:** Create a “master” branch

**Commands:**

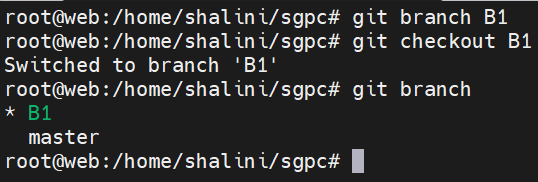
* nano Readme.txt
* git add Readme.txt --- (track/stagging the created file)
* git commit -m “master branch” – (commit into master)
* git branch -- (listing all branches)



**Step3:** Creating a new branch

**Commands:**

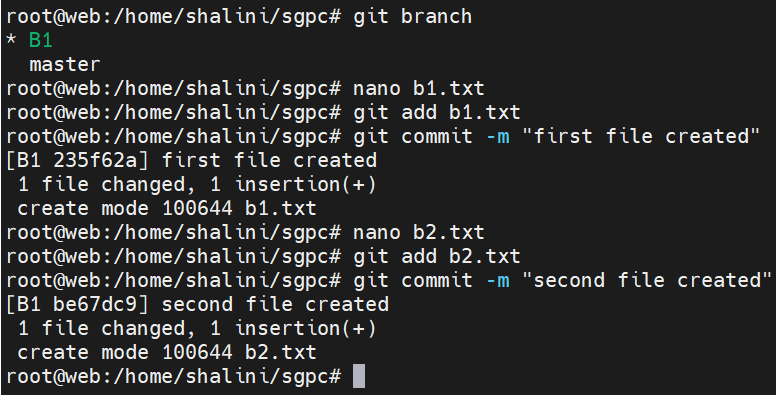
* + git branch B1
  + git checkout B1 -- (Navigate into new branch)
  + git branch



**Step4:** Creating, copying files into new branch

**Commands:**

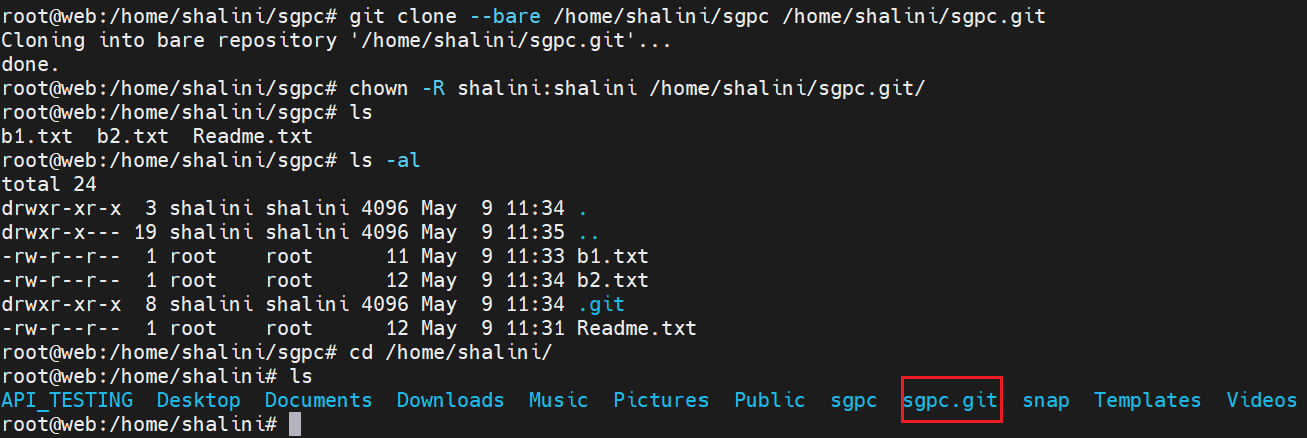
* + nano b1.txt
  + git add b1.txt
  + git commit -m "first file created"
  + nano b2.txt
  + git add b2.txt
  + git commit -m "second file created"



**Step5:** Create a Bare Git Repository for Remote Access

**Command:**

* + cd /home/shalini/sgpc
  + git clone --bare /home/shalini/sgpc /home/shalini/sgpc.git
  + chown -R shalini:shalini /home/shalini/sgpc.git/



**Step6:** Please follow the steps on the Linux B machine and then return to Linux A.

**On Linux B:** 192.168.0.108

**User:** sgpc

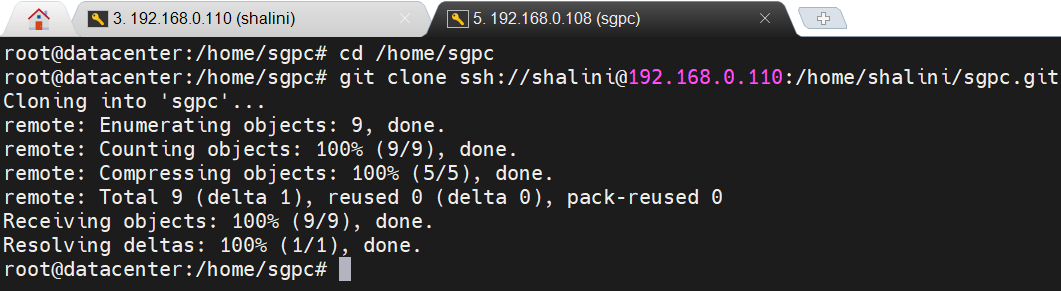
**Pass:** sgpcdata123

**Step7:** Cloning the Repository to Linux B

cd /home/sgpc

**Command:** git clone ssh://shalini@192.168.0.110:/home/shalini/sgpc.git

Pass: **shalini123** (Password of Linux A machine)



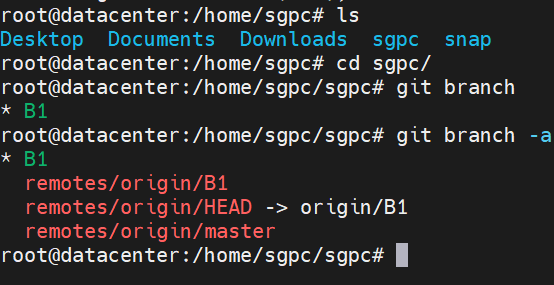
**Step8**: Navigate to the cloned repository on Linux B

Command:

cd sgpc/

git branch (It shows only the current switch branch on the Linux A machine.)

git branch -a (Listing all the branches)

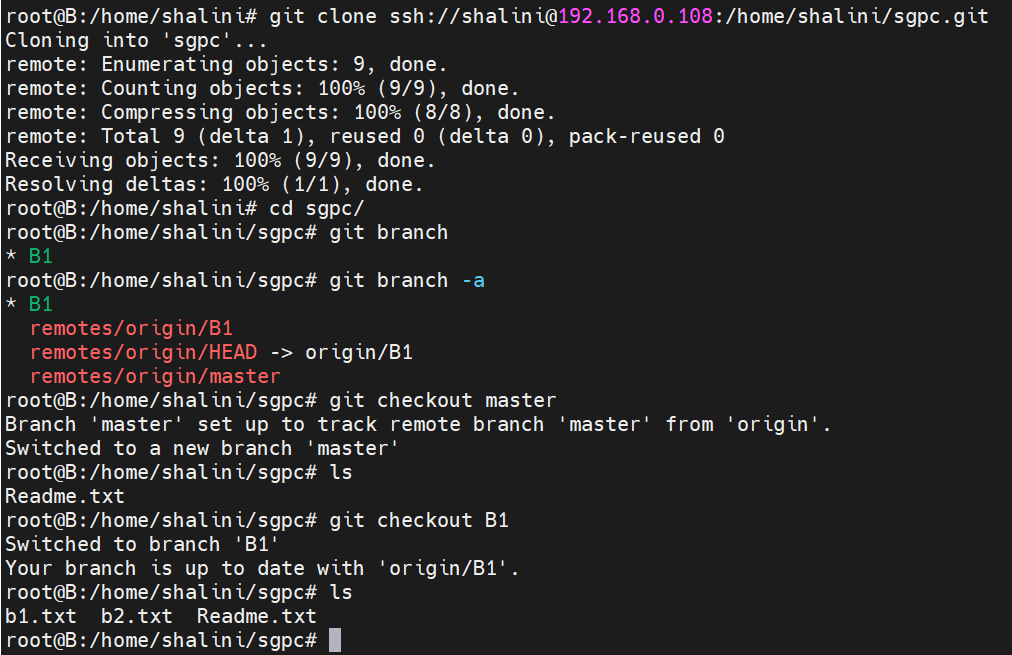


**Step9:** Check all the commit changes

**Command:**

git checkout master

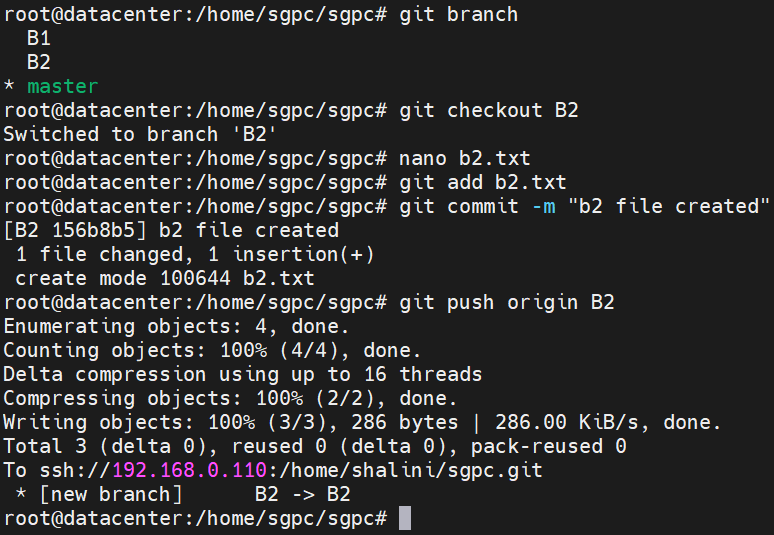
git status -- (Checking the file commit



**Step10:** Create a new branch and push the new branch to the Linux A machine.

**Commands:**

* git checkout master
* git branch B2
* git checkout B2
* nano b2.txt
* git add b2.txt
* git commit -m "b2 file created"
* git push origin B2 -- (**Pushing the new branch to the Linux A machine**.)

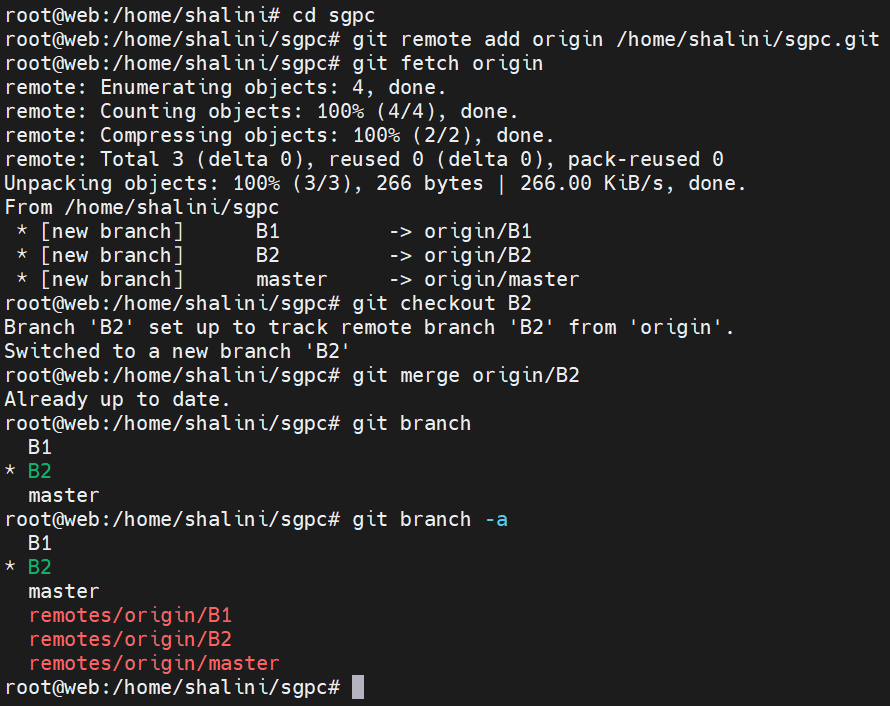


**Step11:** Please follow the steps on the Linux A machine.

**On Linux A machine: 192.168.0.110**

**Commands:**

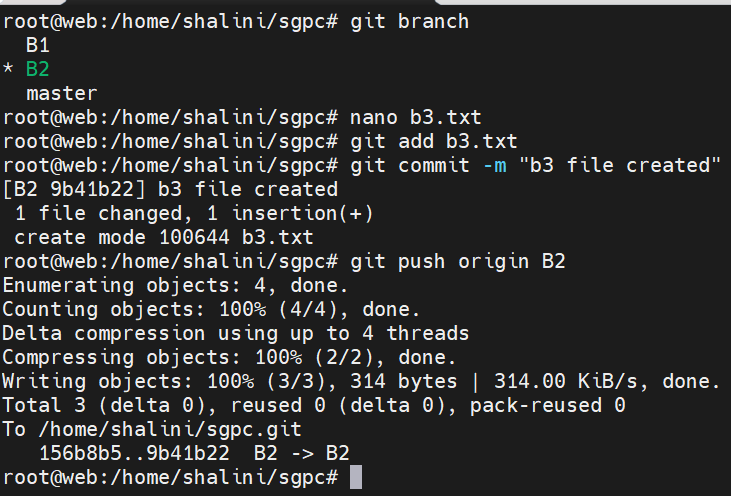
* + git remote add origin /home/shalini/sgpc.git
  + git fetch origin
  + git checkout B2
  + git merge origin/B2



**Step12**: creating a new file on B2 branch and pushing to Linux B

**Commands:**

* + nano b3.txt
  + git add b3.txt
  + git commit -m “b3 file created”
  + git push origin B2

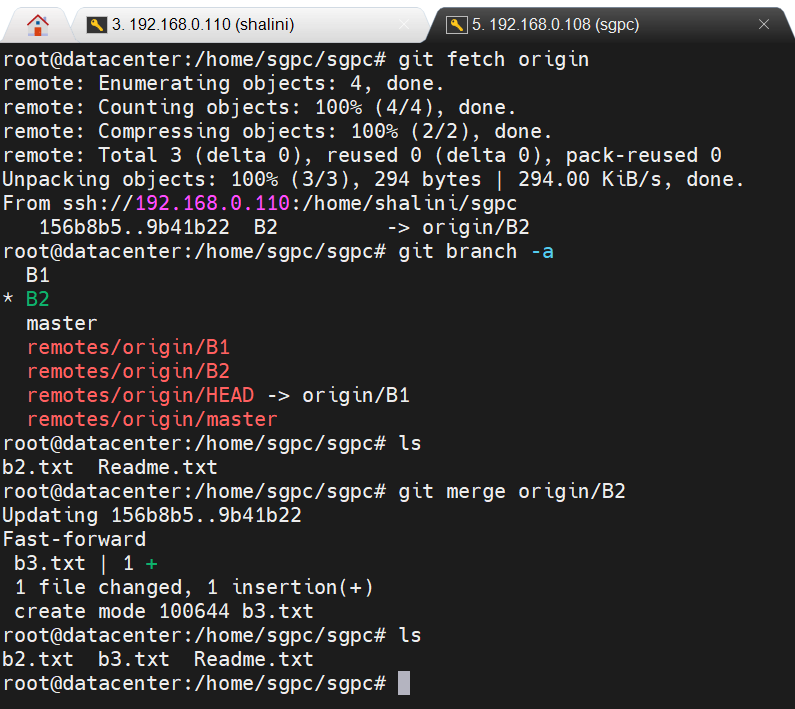


**Step13:** Please follow the steps on the Linux B machine.

**On Linux B machine: 192.168.0.108**

**Command:**

* + git fetch origin
  + git branch -a
  + git checkout B2
  + git merge origin/B2



**Push origin**

**Bare to clone Repo**

**LINUX B**

**192.168.0.108**

**CLONE REPO: sgpc**

**LINUX A**

**192.168.0.110**

**REPO: sgpc**

**BARE: sgpc.git**