# Environment:

At least 2 machines (possibly virtual) with installed on both:

* **Unix** –like OS.
* **Python 2.7+** (must contain ‘unittest’ module)
* Installed **Nfs** utility. (will require ‘nfs-kernel-server’ as well for server and ‘nfs-common’ for client machines).. See links.
* **Ssh** application. See links.

3 users with different permission access on client machine (root, sudoer, common user)

# Preparations.

User creation.

Client:

We will need 1 user with sudo permission level on client machine,

<**sudo\_user**> – with sudoer permission, add him in **visudo** as **(<user> ALL=(ALL) NOPASSWD: ALL**)

Server:

All the work on server machine executed by root user.

Connection between machines.

Passwordless ssh connection . Set up for all users for remote access from server to client machine. link

Nfs.  
To install ‘Nfs’ and run its services follow the steps that described in ‘Nfs installation link’.

During the installation steps, mount shared folders according next paths :

Server**: server\_dir\_path = ’/mnt/public\_nfs’**

Client**: client\_dir\_path = ‘/mnt/nfs’**

**Test script declarations:**

Has to be executed by python interpreter 2.7+

Has to be started with single file, example ’test\_runner.py’, on server machine.

It has to transfer the folder with all its scripts to a remote client pc\*.

‘\*’ following the word mean that this world linked to a specific value which you can find in attachments at the end of the file.

# Test Module.

## Module preparation:

1. Turn on both pcs and server machines.
2. If you are not root switch to root. (example, **‘sudo su -**’)
3. Go to server directory. (example, ‘**cd server\_dir\_path’**)
4. Set up Nfs configure file to next value:

**“server\_dir\_path \*(rw, sync, no\_root\_squash, no\_subtree\_smthing)”**

example,

**“ echo “server\_dir\_path \*(rw, sync, no\_root\_squash, no\_subtree\_check)” >> /etc/exports“ )**

, for server\_dir\_path see attachments.

1. Start Nfs services for both machines if they was turned off. (example ,  **‘/etc/init.d/nfs start’**)
2. Check internet connection for each pc.
3. Turn of firewall for port used by Nsf.
4. Mount server and client Nfs directories if they weren’t.
5. Place ‘**Testing\_script’** in any folder on server machine.

## Class preparations:

1. Create 7 files and directories in mounted directory with different permissions {**777,770,700,444,440,400,000**}, names will be ‘f.{}’ for file and ‘d.{}’ for directory , where {} – matches the permission. Store all the files in ‘files\_list’ and directories in ‘directory\_list’.

‘files\_list’ = [ **f**.**777, f.770, f.700, f.444, f.440, f.400, f.000]**

1. All script starting work will be done from server side.

## Test Suite. Test Sudoer Permission Access.

*Test Suite preparations:*

1. Set up data for expected results, according to expected results for each test case.
2. Execute TestFilePermissionsAsRoot by using ssh connection request with **Sudo privilegue**.
3. The script will switch its working directory to mounted for client machine.

### Test case. Files exist.

Check every file for existence.

1. Loop over ‘file\_ list’, check if each file is exists
2. Compare it with expected result in expected result list.
3. Each comparison should produce a message into log file and console output.

Expected result:

exist files with permissions: 777, 770 , 700, 444, 440, 400, 000

not exist files with permissions : none

### Test case. Files are readable.

Check if every file is readable according permission is writable.

1. Loop over ‘file\_ list’, check a file is readable for current user
2. Compare the outcome with expected result in expected result list.
3. Each comparison should produce a message into log file and console output.

Expected result:

can read from files with permissions: 777, 770 , 700, 444, 440, 400, 000

can’t read from files with permissions : none

### Test case. Files are writable.

Check if file with according permission is writable.

1. Loop over ‘file\_ list’, check a file is writable for current user
2. Compare the outcome with expected result in expected result list.
3. Each comparison should produce a message into log file and console output.

Expected result:

can write into file with permissions: 777, 770 , 700

can’t write into file with permissions: 444, 440, 400, 000

## Tear Dawn.

1.Remove all the data inside of server mounted folder.

## Test Suite. Test Common User Permission Access.

*Test Suite preparations:*

1. Set up data for expected results, according to expected results for each test case.
2. Execute TestFilePermissionsAsRoot by using ssh connection request with **common user privilegue**.
3. The script will switch its working directory to mounted for client machine.

### Test case. File exist.

Check every for existence.

1. Loop over ‘file\_ list’, check if each file is exists
2. Compare it with expected result in expected result list.
3. Each comparison should produce a message into log file and console output.

Expected result:

files with permissions exist : 777, 770 , 700, 444, 440, 400, 000

files with permissions not exist : none

### Test case. Files are readable.

Check if every file is readable according permission is writable.

1. Loop over ‘file\_ list’, check a file is readable for current user
2. Compare the outcome with expected result in expected result list.
3. Each comparison should produce a message into log file and console output.

Expected result:

can read from files with permissions: 777, 444

can’t read from files with permissions : 700, 770, 440, 400, 000

### Test case. Files are writable.

Check if file with according permission is writable.

1. Loop over ‘file\_ list’, check a file is writable for current user
2. Compare the outcome with expected result in expected result list.
3. Each comparison should produce a message into log file and console output.

Expected result:

can write into file with permissions: 777

can’t write into file with permissions: 770, 700, 444, 440, 400, 000

### Tear Down.

1.Remove all the data inside of server mounted folder.

## Class preparations:

1. At server, create 7 directories ( and a file in each of them), in mounted directory with different permissions {**777,770,700,444,440,400,000**}, names will be ‘f.{}’ for file and ‘d.{}’ for directory , where {} – matches the permission. Store all the files in ‘files\_list’ and directories in ‘directory\_list’.

‘files\_list’ = [ **f**.**777, f.770, f.700, f.444, f.440, f.400, f.000]**

1. All script starting work will be done from server side.

## Test Suite. Test Sudoer Permission Access.

*Test Suite preparations:*

1. Set up data for expected results, according to expected results for each test case.
2. Execute TestFilePermissionsAsRoot by using ssh connection request with **Sudo privilegue**.
3. The script will switch its working directory to mounted for client machine.

### Test case. Directory exist.

Check every directory for existence.

1. Loop over ‘directory \_ list’, check if each exists
2. Compare it with expected result in expected result list.
3. Each comparison should produce a message into log file and console output.

Expected result:

exist directory with permissions: 777, 770 , 700, 444, 440, 400, 000

not exist directory with permissions : none

### Test case. Directory are readable.

Check if every directory is readable according permission is writable.

1. Loop over ‘directory \_ list’, check a file in it is readable for current user
2. Compare the outcome with expected result in expected result list.
3. Each comparison should produce a message into log file and console output.

Expected result:

can read from files inside of directory with permissions: 777, 770 , 700, 444, 440, 400, 000

can’t read from files inside of directory with permissions : none

### Test case. Directories are writable.

Check if directory with according permission is writable.

1. Loop over ‘directory \_ list’, check a file is writable for current user
2. Compare the outcome with expected result in expected result list.
3. Each comparison should produce a message into log file and console output.

Expected result:

can write into file inside of directory with permissions: 777, 770 , 700

can’t write into file inside of directory with permissions: 444, 440, 400, 000

## Tear Dawn.

1.Remove all the data inside of server mounted folder.

## Test Suite. Test Common User Permission Access.

*Test Suite preparations:*

1. Set up data for expected results, according to expected results for each test case.
2. Execute TestFilePermissionsAsRoot by using ssh connection request with **common user privilegue**.
3. The script will switch its working directory to mounted for client machine.

### Test case. Directory exist.

Check every directory for existence.

1. Loop over ‘directory \_ list’, check if each exists
2. Compare it with expected result in expected result list.
3. Each comparison should produce a message into log file and console output.

Expected result:

directory with permissions exist : 777, 770 , 700, 444, 440, 400, 000

directory with permissions not exist : none

### Test case. Directory are readable.

Check if every directory is readable according permission is writable.

1. Loop over ‘directory \_ list’, check a file in it is readable for current user
2. Compare the outcome with expected result in expected result list.
3. Each comparison should produce a message into log file and console output.

Expected result:

can read from files inside of directory with permissions: 777, 444

can’t read from files inside of directory with permissions : 700, 770, 440, 400, 000

### Test case. Directories are writable.

Check if directory with according permission is writable.

1. Loop over ‘directory \_ list’, check a file is writable for current user
2. Compare the outcome with expected result in expected result list.
3. Each comparison should produce a message into log file and console output.

Expected result:

can write into file inside of directory with permissions: 777

can’t write into file inside of directory with permissions: 770, 700, 444, 440, 400, 000

### Tear Down.

1.Remove all the data inside of server mounted folder.

# Tear Dawn Module.

Call out from server pc a final script, that will:

1. Check for existence several linux system log file in /var/log folder, such as message,syslog,kernl-log
2. Transfer it’s script log file together with linux logs to server pc.(server\_pc\_log\_directory\*)
3. If logs transfer was successful – delete all script data on clients end.
4. If there was an errors during logfiles transfer action, leave workspace as it is.

# Test data.

## Script name to execute.

To execute this script run python file - **test\_runner.py**

## Logging:

* Log file name for script : ‘nfs\_test\_logs.log’
* Log file directory for client pc: **'**/tmp/nfs\_testing' -(scripts directory)
* Log file directory for server pc: **'**/tmp/nfs\_testing'

## Mounted directory:

* server = '/mnt/public\_nfs'
* client = '/mnt/nfs'

## Ip addresses for ssh connection:

* server = '192.168.182.2'
* client = '192.168.182.5'

## User names:

* sudoer(client) = **'nikita'**
* root (server )= **'root'**

# Links:

## NFS.

* Wiki - <https://en.wikipedia.org/wiki/Network_File_System>
* Redhat - <http://www.tecmint.com/how-to-setup-nfs-server-in-linux/>
* Ubuntu - <https://www.digitalocean.com/community/tutorials/how-to-set-up-an-nfs-mount-on-ubuntu-14-04>

## SSH.

* Wiki - <https://en.wikipedia.org/wiki/Secure_Shell>
* Passwordless connection set up - <https://www.digitalocean.com/community/tutorials/how-to-set-up-ssh-keys--2>