

Super User is a question and answer site for computer enthusiasts and power users. It's 100% free, no registration required.

[Tell me more](#) x

Give write permissions to multiple users on a folder in Ubuntu

CAREERS 2.0
by stackoverflow



>



Easily apply for your dream job
No formatting needed!

There is a folder that is owned by user tomcat6:

```
drwxr-xr-x 2 tomcat6 tomcat6 69632 2011-05-06 03:43 document
```

I want to allow another user (ruser) write permissions on document folder. The two users (tomcat6 and ruser) does not belong to same group. I have tried using `setfacl` :

```
sudo setfacl -m u:ruser:rwX document
```

but this gives me `setfacl: document: Operation not supported error`. Kindly help me.

[linux](#) [security](#) [file-permissions](#) [acl](#)

asked May 9 '11 at 8:44



[TheVillageIdiot](#)
272 1 3 14

1 Answer

There are two ways to do this: set the directory to "world" writable or create a new group for the two users and make the directory writeable to that group.

Obviously making it world writeable is a Bad Thing, so the second option is preferable.

Users in Linux can belong to more than one group. In this case you want to create a brand new group, let's call it `tomandruser` :

```
sudo groupadd tomandruser
```

Now that the group exists, add the two users to it:

```
sudo usermod -a -G tomandruser tomcat6
sudo usermod -a -G tomandruser ruser
```

Now all that's left is to set the permissions on the directory:

```
sudo chgrp -R tomandruser /path/to/the/directory
sudo chmod -R 770 /path/to/the/directory
```

Now only members of the `tomandruser` group can read, write, or execute anything within the directory. Note the `-R` argument to the `chmod` and `chgrp` commands: this tells them to recurse into every sub directory of the target directory and modify every file and directory it finds.

You may also want to change 770 to something like 774 if you want others to be able to read the files, 775 if you want others to read and execute the files, etc. Group assignment changes won't take effect until the users log out and back in.

edited May 9 '11 at 10:47

answered May 9 '11 at 10:06



[Amazed](#)
4,397 1 13 25

works like a charm – [swapnilsarwe](#) Aug 21 '12 at 19:36

You probably want to also set the set-group-ID flag for directories, to make new files and sub-directories automatically owned by the right group: `sudo find /path/to/the/directory -type d -exec chmod 2770 '{}' \;` – [Marcello Nuccio](#) Dec 13 '12 at 14:28

- I'd avoid using `chmod 770`, `775` or whatever. That messes with the permissions of *all* files. Instead use something like `chmod -R g+w` to add write permissions without mucking up everything else. – [Christian Varga](#) Jan 3 at 15:11

If a user creates a new file there (say, mysql by `SELECT INTO outfile`), it sets permissions to its primary group (`mysql` in this case), and the file is not accessible by another user anyway. How to workaround this? – [Olexa](#)
May 15 at 11:53

Found an answer for my question here: superuser.com/a/19333/171762 – [Olexa](#) May 15 at 12:37

show 2 more comments

Not the answer you're looking for? Browse other questions tagged [linux](#) [security](#)

[file-permissions](#) [acl](#) or [ask your own question](#).