Granting regular system user with permissions to restart system services (/etc/init.d/ daemons) on Debian Linux via sudo

Author : admin

If you're a system administrator, you have certainly have faced the task to allow a regular system user without system root permissions to be able to restart some of the system services, every now and then. This case is very often encountered, if you have to work with a group of programmers who need to restart certain daemon services either that directly are related to some of your programmers.

An example case would be a java programmer/s, who runs code on a tomcat or resin server. Another possible scenario which I just encountered is a php programmer, who needs to experiment with nginx rewrite rules.

Another scenario, where you have to grant access to **stop, start or restart** functions of some init scripts like let's say **/etc/init.d/mysql or /etc/init.d/apache** would be if you have to work in a team with other *system administrators* or some *unix tech support guy (second system admin) etc.*

The cases in which you might need to enpower an otherwise regular user BSD/Linux system account with extra administrator (root) permissions are multiple, however what is important is how you can do this when suddenly you need to complete this task.

There are quite a lot of things written on the topic and I remember that the first time I had to add a regular Linux system accounts to have an extra permissions only tosome system binaries was a real pain in the ass.
It took me a lot of searching online reading forums and asking in irc.freenode.net until I got the solution, but thanksfully finally I got it.
Therefore I found it might be interesting to my readers to find out how this trivial but very helpful task can be achieved on Linux and BSD and possibly other UNIX server or desktop installs.

The key to the whole set task resides in the is in the handy Linux tool **Sudo** that I believe every system administrator out there is aware of.
Even the *sudo tool* debian package description indicates that it's the exact tool one needs to allow certain user to execute commands as root. Here is sudo's description:

Sudo (superuser do) allows a system
administrator to give certain users (or
groups of users) the ability to run some
(or all) commands as root or another user
while logging the commands and arguments.

Using sudo to complete the set task, on a first glimpse looks really complex, however it appears to achieve allowing a regular user to execute as root is pretty simple.

To give you an idea how you can achieve the task I'll give a simple scenario.
Let's assume **you need to add the Linux system users with login names: *ivan* and testuser to be able to restart your Apache,MySQL and Nginx servers via the services respective init scripts.**

Here is how to achieve it:

1. First you need to have the sudo package installed

debian:~# apt-get install sudo

If you already have the package then simply skip step 1 and proceed further.

2. Edit /etc/sudoers with your favourite text editor and find the text

root ALL=(ALL) ALL

3. After the text insert the code:

Cmnd_Alias APACHE = /etc/init.d/apache2
User_Alias APACHEUSERS = ivan, testuser
APACHEUSERS ALL = NOPASSWD : APACHE

Cmnd_Alias MYSQL = /etc/init.d/mysql
User_Alias NGINXUSERS = ivan, testuser
MYSQLUSERS ALL = NOPASSWD : NGINX

Cmnd_Alias NGINX = /etc/init.d/nginx
User_Alias NGINXUSERS = soccerfame, hipo
NGINXUSERS ALL = NOPASSWD : NGINX

That's all now the normal system users with non-root permissions **ivan** and **testuser** will be allowed to restart your Debian Linux Apache, MySQL and Nginx services.

4. Now let's test your if your users can properly restart the 3 services with the *testuser*

debian:~# su testuser
testuser@debian:~$ id
uid=1001(testuser) gid=1001(testuser) groups=1001(testuser)

testuser@debian:~$ sudo /etc/init.d/nginx restart
Restarting nginx: nginx.
testuser@debian:~$ sudo /etc/init.d/apache2 restart
Restarting web server: apache2 ... waiting .

```
testuser@debian:~$ sudo /etc/init.d/mysql restart
Stopping MySQL database server: mysqld.
Starting MySQL database server: mysqld.
Checking for corrupt, not cleanly closed and upgrade needing tables..
```

As you can see in the above commands execution output now my **testuser** which is a non-root user is able to use some of my services which require administrator permissions.

Now if you need to add some more services which require root permissions, simply place some more code in the */etc/sudoers* let's assume you need to put the example service which binary is to be found in **/usr/sbin/example** , you will have to place in your */etc/sudoers*

```
Cmnd_Alias EXAMPLE = /usr/sbin/example
User_Alias EXAMPLEXUSERS = ivan, testuser
MYSQLUSERS ALL = NOPASSWD : EXAMPLE
```