

Create user and password on Linux non interactive and add it to sudo a tiny Dev Ops script

Author : admin



A common task for SysAdmins who managed a multitude of servers remotely via Secure Shell was to add a user and assign password by using a script, this was sometimes necessary to set-up some system users and create access for university users on 10 / 20 testing Linux servers.

Nowadays this **task of adding user to a list of remote servers and granting the new user superuser permissions** through `/etc/sudoers` is practiced heavily by the so called **Dev Ops (Just another Business Word for Senior System Administrators with good scripting skills and a little bit of development experience - same game different name.**

The Dev Ops System Integration Engineers use this useful **add non-interactive user via SSH in Cloud environments** in order to prepare superuser (root permissioned through `/etc/sudoers`) user, that is later be used for lets say deployment on a few hundred of servers of lets say **LAMP (Linux + Apache + MySQL + PHP)** or **LEMP (Linux NGINX MySQL PHP)** or **Software Load Balancer** HAProxy balacing for MySQL clusters / Nginx Application servers / JIRAs etc, through a Playbook script with some deployment automation tool such as [Ansible](#).

Well enough talk here is the few lines of code which does create a user locally:

```
linux:~# apt-get install --yes sudo
linux:~# useradd devops --home /home/devops -s /bin/bash
linux:~# mkdir /home/devops
linux:~# chown -R devops:devops /home/devops
linux:~# echo 'username:testpass' | chpasswd
```

Though this lines could be invoked easily by passing it as arguments via **ssh** it is often unhandy to run them on remote host, because some of the remote hosts against executed, might have already the user existent with granted permissions for **sudo**

Thus a much better way to do things is use below script and first upload it to remote servers by running the scp command in a loop:

```
while read line; do
scp root@$i:/root/
ssh "create user noninteractive and add to sudoers.sh"
done
```

Where servers_list.txt contains a list of remote IPs:

```
#!/bin/bash
# Create new user/group and add nopasswd login to sudoers
# Author: Georgi Georgiev
# has to be run sa root - sudo devops
# hipo@pc-freak.net
```

```
u_id='devops';
g_id='devops';
pass='testpass';
sudoers_f='/etc/sudoers';
```

```
check_install_sudo () {
if [ $(dpkg --get-selections | cut -f1|grep -E '^sudo') ]; then
apt-get install --yes sudo
else
printf "Nothing to do sudo installed";
fi
}
```

```
check_install_user () {
```

```
if [ "$(sed -n "/$u_id/p" /etc/passwd/wc -l)" -eq 0 ]; then
apt-get install --yes sudo
apt-get install --yes sudo
useradd $u_id --home /home/$u_id
mkdir /home/$u_id
chown -R $u_id:$g_id /home/$u_id
echo "$u_id:$pass" | chpasswd
cp -rpf /etc/bash.bashrc /home/$u_id
if [ "$(sed -n "/$u_id/p" $sudoers_f/wc -l)" -eq "0" ]; then
echo "$u_id ALL=(ALL) NOPASSWD: ALL" >> $sudoers_f
else
echo "$u_id existing. Exiting ..";
exit 1;
fi

else
echo "Will do nothing because $u_id exists";
fi

}

check_install_sudo;
check_install_user;
```

By the way this task was the simplest task given by a Company where I applied for a Dev Ops System Engineer, so I hope this will help someone else too.

P.S. If you prefer Shell scripts (even though much harder, time consuming etc.) as a mean of automation as an alternative to **Ansible** / **Chef** I suggest you check out and perhaps try to do the task with <http://fuckingshellscripts.org> :)