Linux Shell Scripting - Create User Accounts

In this tutorial, I will be creating a shell script that can be used to create Linux user accounts.

Prerequisites

- access to a running Linux distribution

To complete this tutorial, you will need access to a running Linux distribution. If you do not already have access to a Linux system, I have a number of VirtualBox tutorials where I demonstrate the installation of CentOS 7 and Ubuntu 22 as virtual machines, accessible **here**.

I also have a few tutorials where I demonstrate the creation of AWS compute instances, RHEL 8 and Ubuntu 20, accessible **here**. Keep in mind that you will need to create an AWS account to be able to create a compute instance. If you do not have an AWS account, my tutorial **Create AWS Free Tier Account** is accessible **here.**

For this tutorial, I will be using my **CentOS 7 VM** for testing, but the script will work on any running Linux system.

This script will create a user account on the local system. The person executing this script must have superuser privileges. Furthermore, when executing the script, a username argument must be supplied. Optionally, subsequent arguments will be used as the account description (comment). It will auto-generate the account password, and, finally, it will display the account's username, password and host name where the script was executed.

Steps to complete tutorial:

Create Script

To begin, in my home directory, I created a new **`scripts`** directory and, in it, I placed a newly created empty script file named **`add_local_user.sh`**

```
/home/liam/scripts/add_local_user.sh
```

I will begin by adding the full path of the command interpreter (in this case **/bin/bash**), as well as, a brief description of what the script does:

```
#!/bin/bash
#
# This script creates a user account on the local system.
```

Ensure Superuser Privileges

Then, I want to ensure that the script is being executed by a user with superuser privileges (either **root** or using **sudo**). I will check the environment variable **UID** which is a unique identifier assigned to each user. If the **$UID == 0**, then either the **root** user, or a user with **sudo** privileges, has executed the script.

I will verify the **UID** values assigned to the root user & my non-root user on my running CentOS 7 VM:

```
$ id -u root
```

```
$ id -u liam
```

```
[liam@centos7-vm ~]$
[liam@centos7-vm ~]$ id -u root
0
[liam@centos7-vm ~]$ id -u liam
1000
[liam@centos7-vm ~]$
```

We can see from the above command output that the root user has a UID of 0 and my non-root user has a UID of 1000.

Using this code snippet, I am checking if the **root** user, or a non-root user using **sudo** privileges, is executing the script. If not, provide a helpful usage message and exit the script.

```
# Make sure the script is being executed with superuser privileges.
if [[ "${UID}" -ne 0 ]]
then
        echo "You must use superuser privileges to run this script."
        exit 1
fi
```

I will now test this out, for success & failure, by setting the execute permission on the script and executing it:

```
$ chmod +x add_local_user.sh
$ sudo ./add_local_user.sh
$ ./add_local_user.sh
```

```
[liam@centos7-vm scripts]$
[liam@centos7-vm scripts]$ chmod +x add_local_user.sh
[liam@centos7-vm scripts]$
[liam@centos7-vm scripts]$ ls -l add_local_user.sh
-rwxrwx---. 1 root vboxsf 244 Sep 20 07:59 add_local_user.sh
[liam@centos7-vm scripts]$
[liam@centos7-vm scripts]$ sudo ./add_local_user.sh
[liam@centos7-vm scripts]$
[liam@centos7-vm scripts]$ ./add_local_user.sh
You must use superuser privileges to run this script.
[liam@centos7-vm scripts]$
```

You will notice that executing the script without superuser privileges generated the helpful usage message and exited the script.

Script Arguments

Next, I want to ensure that a username argument is passed to the script. If not, I provide usage information via standard error (**STDERR**). Optionally, additional arguments will be used as the account description (comment).

```
# Provide usage information to person executing the script.
if [[ "${#}" -lt 1 ]]
then
        echo "Usage: ${0} USER_NAME [COMMENT]..." >&2
        echo "This script creates an account on the local system with the name of USER_NAME and
a comments field of COMMENT." >&2
        exit 1
fi
```

I will now assign the first provided argument to a variable. Then, I will use the bash built-in **`shift`** command to shift the positional parameters to the left ensuring that all of the remaining script arguments, **`${@}`**, can be assigned to another variable.

```
# first argument provided on the command line as the account username
USER_NAME="${1}"

shift

# remaining arguments on the command line will be treated as the comment for the account
COMMENT="${@}"

# used for testing
echo ${USER_NAME}
echo ${COMMENT}
```

I will now test this out, for success & failure, by executing the script again:

```
$ sudo ./add_local_user.sh fred Fred Smiley Finney
$ sudo ./add_local_user.sh earl Earl Pearl
$ sudo ./add_local_user.sh jack
$ sudo ./add_local_user.sh
$ ./add_local_user.sh
```

```
[liam@centos7-vm scripts]$
[liam@centos7-vm scripts]$ sudo ./add_local_user.sh fred Fred Smiley Finney
fred
Fred Smiley Finney
[liam@centos7-vm scripts]$ sudo ./add_local_user.sh earl Earl Pearl
earl
Earl Pearl
[liam@centos7-vm scripts]$ sudo ./add_local_user.sh jack
jack

[liam@centos7-vm scripts]$ sudo ./add_local_user.sh
Usage: ./add_local_user.sh USER_NAME [COMMENT]...
This script creates an account on the local system with the name of USER_NAME an
d a comments field of COMMENT.
[liam@centos7-vm scripts]$ ./add_local_user.sh
You must use superuser privileges to run this script.
[liam@centos7-vm scripts]$
```

You will notice that executing the script without at least one argument generated the helpful usage messages and exited the script.

Create user

I will add the following to the script to create a user account, using the arguments provided, and discard any output generated.

```
# create new user account
useradd -c "${COMMENT}" -m "${USER_NAME}" &> /dev/null
```

If a user account was not created, I will display a message and exit the script.

```
if [[ "${?}" -ne 0 ]]
then
      echo "User account was NOT created." >&2
      exit 1
fi
```

Generate a password

I will now demonstrate the steps used to generate the user account password.

First, I will use the **RANDOM** built-in shell variable.

```
[liam@centos7-vm ~]$
[liam@centos7-vm ~]$ man bash | grep RANDOM
      RANDOM Each time this parameter is referenced, a random integer between 0 and 32767 is
             value to RANDOM.  If RANDOM is unset, it loses its special properties, even  if
             any  of COMP_WORDBREAKS, RANDOM, SECONDS, LINENO, HISTCMD, FUNCNAME, GROUPS, or
[liam@centos7-vm ~]$
```

```
$ PASSWORD="${RANDOM}"
$ echo "${PASSWORD}"
```

```
[liam@centos7-vm scripts]$
[liam@centos7-vm scripts]$ PASSWORD="${RANDOM}"
[liam@centos7-vm scripts]$ echo "${PASSWORD}"
26040
[liam@centos7-vm scripts]$ PASSWORD="${RANDOM}"
[liam@centos7-vm scripts]$ echo "${PASSWORD}"
10353
[liam@centos7-vm scripts]$ PASSWORD="${RANDOM}"
[liam@centos7-vm scripts]$ echo "${PASSWORD}"
20220
[liam@centos7-vm scripts]$
```

Now I will use the current date/time as the basis for the password.

```
[liam@centos7-vm scripts]$
[liam@centos7-vm scripts]$ man date | grep -e "%[s]"
       %s     seconds since 1970-01-01 00:00:00 UTC
[liam@centos7-vm scripts]$
```

```
$ PASSWORD=$(date +%s)
$ echo "${PASSWORD}"
```

```
[liam@centos7-vm scripts]$
[liam@centos7-vm scripts]$ PASSWORD=$(date +%s)
[liam@centos7-vm scripts]$ echo "${PASSWORD}"
1663681873
[liam@centos7-vm scripts]$ PASSWORD=$(date +%s)
[liam@centos7-vm scripts]$ echo "${PASSWORD}"
1663681883
[liam@centos7-vm scripts]$ PASSWORD=$(date +%s)
[liam@centos7-vm scripts]$ echo "${PASSWORD}"
1663681892
[liam@centos7-vm scripts]$
```

I will also use nanoseconds to add randomization.

```
[liam@centos7-vm scripts]$
[liam@centos7-vm scripts]$ man date | grep -e "%[N]"
       %N     nanoseconds (000000000..999999999)
[liam@centos7-vm scripts]$
```

```
$ PASSWORD=$(date +%s%N)
$ echo ${PASSWORD}
```

```
[liam@centos7-vm scripts]$
[liam@centos7-vm scripts]$ PASSWORD=$(date +%s%N)
[liam@centos7-vm scripts]$ echo ${PASSWORD}
1663682217438097528
[liam@centos7-vm scripts]$ PASSWORD=$(date +%s%N)
[liam@centos7-vm scripts]$ echo ${PASSWORD}
1663682223852887174
[liam@centos7-vm scripts]$ PASSWORD=$(date +%s%N)
[liam@centos7-vm scripts]$ echo ${PASSWORD}
1663682230307583571
[liam@centos7-vm scripts]$
```

I will now include **sha256sum** to generate a more complex password.

```
[liam@centos7-vm ~]$
[liam@centos7-vm ~]$ man sha256sum | head -n 14
SHA256SUM(1)                    User Commands                    SHA256SUM(1)



NAME
       sha256sum - compute and check SHA256 message digest

SYNOPSIS
       sha256sum [OPTION]... [FILE]...

DESCRIPTION
       Print  or check SHA256 (256-bit) checksums.  With no FILE, or when FILE
       is -, read standard input.
```

```
$ PASSWORD=$(date +%s%N | sha256sum | head -c32)
$ echo "${PASSWORD}"
```

```
[liam@centos7-vm scripts]$
[liam@centos7-vm scripts]$ PASSWORD=$(date +%s%N | sha256sum | head -c32)
[liam@centos7-vm scripts]$ echo "${PASSWORD}"
be591dcd7e8991fbc64e11c7541fe7ba
[liam@centos7-vm scripts]$
[liam@centos7-vm scripts]$ PASSWORD=$(date +%s%N | sha256sum | head -c32)
[liam@centos7-vm scripts]$ echo "${PASSWORD}"
2921fbabd07a86fb0b779eecaa423a18
[liam@centos7-vm scripts]$
[liam@centos7-vm scripts]$ PASSWORD=$(date +%s%N | sha256sum | head -c32)
[liam@centos7-vm scripts]$ echo "${PASSWORD}"
20e0285d2c2a0758e8a7a4e300c1fc0c
[liam@centos7-vm scripts]$
```

I will now combine the different steps to improve password complexity.

```
$ PASSWORD=$(date +%s%N${RANDOM}${RANDOM} | sha256sum | head -c48)
$ echo "${PASSWORD}"
```

```
[liam@centos7-vm scripts]$
[liam@centos7-vm scripts]$ PASSWORD=$(date +%s%N${RANDOM}${RANDOM} | sha256sum | head -c48)
[liam@centos7-vm scripts]$ echo "${PASSWORD}"
00d32c70950cc4f94cf7e08f240be99d9f68c8168202851c
[liam@centos7-vm scripts]$
[liam@centos7-vm scripts]$ PASSWORD=$(date +%s%N${RANDOM}${RANDOM} | sha256sum | head -c48)
[liam@centos7-vm scripts]$ echo "${PASSWORD}"
f2c6e9bba4995bdd7a3d41d029b3a81104b8b0ed5a0d747c
[liam@centos7-vm scripts]$
[liam@centos7-vm scripts]$ PASSWORD=$(date +%s%N${RANDOM}${RANDOM} | sha256sum | head -c48)
[liam@centos7-vm scripts]$ echo "${PASSWORD}"
cc81151cfe6371e66f068a05ce9983779576fb3f0bc0e4a0
[liam@centos7-vm scripts]$
```

To make it even more complex, I will add a special character at the end.

To do this, I will, first, demonstrate the use of the **fold** command using the **-w** option to specify the width (number of characters) of the fold.

```
FOLD(1)                          User Commands                          FOLD(1)

NAME
       fold - wrap each input line to fit in specified width

SYNOPSIS
       fold [OPTION]... [FILE]...

DESCRIPTION
       Wrap  input  lines in each FILE (standard input by default), writing to
       standard output.
```

```
[liam@centos7-vm ~]$
[liam@centos7-vm ~]$ man fold | grep -e "-[w]"
       -w, --width=WIDTH
[liam@centos7-vm ~]$
```

```
$ SPECIAL='!@#$%^&*()_-+='
$ echo ${SPECIAL} | fold -w  1
```

```
[liam@centos7-vm scripts]$
[liam@centos7-vm scripts]$ SPECIAL='!@#$%^&*()_-+='
[liam@centos7-vm scripts]$
[liam@centos7-vm scripts]$ echo ${SPECIAL} | fold -w  1
!
@
#
$
%
^
&
*
(
)
_
-
+
=
```

Now I will add the **shuf** commad to shuffle the results each time it's run.

```
[liam@centos7-vm ~]$
[liam@centos7-vm ~]$ man shuf | head -n 7
SHUF(1)                          User Commands                          SHUF(1)



NAME
       shuf - generate random permutations
```

```
$ echo '!@#$%^&*()_-+=' | fold -w1 | shuf | head -c 1
```

```
[liam@centos7-vm scripts]$
[liam@centos7-vm scripts]$ echo '!@#$%^&*()_-+=' | fold -w1 | shuf | head -c 1
*[liam@centos7-vm scripts]$
[liam@centos7-vm scripts]$ echo '!@#$%^&*()_-+=' | fold -w1 | shuf | head -c 1
![liam@centos7-vm scripts]$
[liam@centos7-vm scripts]$ echo '!@#$%^&*()_-+=' | fold -w1 | shuf | head -c 1
@[liam@centos7-vm scripts]$
[liam@centos7-vm scripts]$ echo '!@#$%^&*()_-+=' | fold -w1 | shuf | head -c 1
@[liam@centos7-vm scripts]$
[liam@centos7-vm scripts]$ echo '!@#$%^&*()_-+=' | fold -w1 | shuf | head -c 1
^[liam@centos7-vm scripts]$
[liam@centos7-vm scripts]$ echo '!@#$%^&*()_-+=' | fold -w1 | shuf | head -c 1
#[liam@centos7-vm scripts]$
[liam@centos7-vm scripts]$
```

I will now add the following to the script:

```
# generate a password for the new account.
SPECIAL=$(echo '!@#$%^&*()_-+=' | fold -w1 | shuf | head -c1)
PASSWORD=$(date +%s%N${RANDOM}${RANDOM} | sha256sum | head -c48)
PASSWORD_FINAL="${PASSWORD}${SPECIAL}"
```

Set user password

I will now assign the generated password to the new user account by using the **`--stdin`** option of the **`passwd`** command.

```
[liam@centos7-vm ~]$
[liam@centos7-vm ~]$ man passwd | grep -A 2 -e " --stdin"
       --stdin
              This  option is used to indicate that passwd should read the new
              password from standard input, which can be a pipe.
[liam@centos7-vm ~]$
```

```
echo "${PASSWORD_FINAL}" | passwd --stdin "${USER_NAME}" &> /dev/null
```

If the user account password was not set, I will display a message and exit the script.

```
if [[ "${?}" -ne 0 ]]
then
      echo "User account password NOT set!" >&2
      exit 1
fi
```

Force password change on first login

I will add the following to the script to force password reset and ensure the user must create a new password at first login.

```
passwd -e "${USER_NAME}" &> /dev/null
```

Display Script Output

Finally, I will display the new username, password and host system information on screen. This information can be sent to the new user.

```
# display new username, password and host system information
echo
echo -e "username:\n${USER_NAME}\n"
echo -e "password:\n${PASSWORD_FINAL}\n"
echo -e "host:\n${HOSTNAME}"
exit 0
```

Review Script

```
#!/bin/bash

# This script creates a user account on the local system.
# The person executing this script must have superuser privileges.
# When executing the script, a username argument must be supplied.
# Optionally, subsequent arguments can be supplied as the account
# description (comment).
# It will auto-generate the account password.
# Finally, it will display the account's username, password and host.

# Ensure user has permission to execute script.
if [[ "${UID}" -ne 0 ]]
then
        echo "You must have superuser privileges to execute this script." >&2
        exit 1
fi

# Provide usage information to person executing the script.
if [[ "${#}" -lt 1 ]]
then
        echo "Usage: ${0} USER_NAME [COMMENT]..." >&2
        echo "This script creates an account on the local system with the name of USER_NAME and
a comments field of COMMENT." >&2
        exit 1
fi

# first argument provided on the command line as the account username
USER_NAME="${1}"

shift

# remaining arguments on the command line will be treated as the comment for the account
COMMENT="${@}"
```

```
# create new user account
useradd -c "${COMMENT}" -m "${USER_NAME}" &> /dev/null

if [[ "${?}" -ne 0 ]]
then
        echo "User account was NOT created." >&2
        exit 1
fi

# generate a password for the new account.
SPECIAL=$(echo '!@#$%^&*()_-+=' | fold -w1 | shuf | head -c1)
PASSWORD=$(date +%s%N${RANDOM}${RANDOM} | sha256sum | head -c48)
PASSWORD_FINAL="${PASSWORD}${SPECIAL}"

# assign the password to the newly created user account.
echo "${PASSWORD_FINAL}" | passwd --stdin "${USER_NAME}" &> /dev/null

# Check to see if the passwd command succeeded.
if [[ "${?}" -ne 0 ]]
then
        echo "User account password NOT set!" >&2
        exit 1
fi

# expire new user account password to force password reset at 1st login
passwd -e "${USER_NAME}" &> /dev/null

# display new username, password and host system information
echo
echo -e "username:\n${USER_NAME}\n"
echo -e "password:\n${PASSWORD_FINAL}\n"
echo -e "host:\n${HOSTNAME}"

exit 0
```

Create user accounts

It is now time to create a few accounts and ensure that when logging in to each account a new password must be set.

I will now test this out, for success & failure, by executing the script again:

```
$ sudo ./add_local_user.sh fred Fred Smiley Finney
```

```
[liam@centos7-vm scripts]$
[liam@centos7-vm scripts]$ sudo ./add_local_user.sh fred Fred Smiley Finney

username:
fred

password:
6c1c8fc415ef605383bb7ef5199bdf39dd03e38918820d4f=

host:
centos7-vm
[liam@centos7-vm scripts]$
```

I will note the password generated and use it when logging into the newly created account.

6c1c8fc415ef605383bb7ef5199bdf39dd03e38918820d4f=

$ su - fred

```
[liam@centos7-vm scripts]$
[liam@centos7-vm scripts]$ su - fred
Password:
You are required to change your password immediately (root enforced)
Changing password for fred.
(current) UNIX password:
New password:
Retype new password:
[fred@centos7-vm ~]$
[fred@centos7-vm ~]$
[fred@centos7-vm ~]$ pwd
/home/fred
[fred@centos7-vm ~]$
```

$ sudo ./add_local_user.sh earl Earl Pearl

```
[liam@centos7-vm scripts]$
[liam@centos7-vm scripts]$ sudo ./add_local_user.sh earl Earl Pearl

username:
earl

password:
f8812e6c7ba8f855595609da0e0640e337247c5b04f6efa5(

host:
centos7-vm
[liam@centos7-vm scripts]$
```

f8812e6c7ba8f855595609da0e0640e337247c5b04f6efa5(

$ su - earl

```
[liam@centos7-vm scripts]$
[liam@centos7-vm scripts]$ su - earl
Password:
You are required to change your password immediately (root enforced)
Changing password for earl.
(current) UNIX password:
New password:
Retype new password:
[earl@centos7-vm ~]$
[earl@centos7-vm ~]$ pwd
/home/earl
[earl@centos7-vm ~]$
```

$ sudo ./add_local_user.sh jack

```
[liam@centos7-vm scripts]$
[liam@centos7-vm scripts]$ sudo ./add_local_user.sh jack

username:
jack

password:
495313435d07ed6b476540327fc35684578ac13526930226@

host:
centos7-vm
[liam@centos7-vm scripts]$
```

495313435d07ed6b476540327fc35684578ac13526930226@

$ su - jack

```
[liam@centos7-vm scripts]$
[liam@centos7-vm scripts]$ su - jack
Password:
You are required to change your password immediately (root enforced)
Changing password for jack.
(current) UNIX password:
New password:
Retype new password:
[jack@centos7-vm ~]$
[jack@centos7-vm ~]$ pwd
/home/jack
[jack@centos7-vm ~]$
```

I will perform my last verification with the following:
$ ls -l /home
$ tail -n 3 /etc/passwd

```
[liam@centos7-vm scripts]$
[liam@centos7-vm scripts]$ ls -l /home
total 0
drwx------. 2 earl earl  83 Sep 21 06:52 earl
drwx------. 2 fred fred  83 Sep 21 06:47 fred
drwx------. 2 jack jack  83 Sep 21 07:06 jack
drwx------. 3 liam liam 126 Sep 20 09:26 liam
[liam@centos7-vm scripts]$
[liam@centos7-vm scripts]$ tail -n 3 /etc/passwd
fred:x:1001:1001:Fred Smiley Finney:/home/fred:/bin/bash
earl:x:1002:1002:Earl Pearl:/home/earl:/bin/bash
jack:x:1003:1003::/home/jack:/bin/bash
[liam@centos7-vm scripts]$
```

The user accounts were successfully created and password expiration worked as expected.

Finally, I will test for failure with the following:

```
$ sudo ./add_local_user.sh
$ ./add_local_user.sh
```

```
[liam@centos7-vm scripts]$
[liam@centos7-vm scripts]$ sudo ./add_local_user.sh
Usage: ./add_local_user.sh USER_NAME [COMMENT]...
This script creates an account on the local system with the name of USER_NAME and
 a comments field of COMMENT.
[liam@centos7-vm scripts]$
[liam@centos7-vm scripts]$ ./add_local_user.sh
You must use superuser privileges to run this script.
[liam@centos7-vm scripts]$
```

I hope you have enjoyed completing this tutorial and found it helpful.

If you would like to know how to create a shell script that can disable, delete and archive user accounts, my tutorial **Delete User Accounts** is accessible **here**. My Linux tutorials page is **here**, while my main tutorials page is accessible **here**.

Back to Top