18/09/2024

User Account Management Commands

1.ADDUSER

Add user or create user

```
sudo adduser <username>
```

Displays the entire information about user present in passwd file

```
getent passwd
```

/etc/passwd file contain a list of all users(info)

```
cat adduser /etc/passwd
```

List the usernames only

```
cut -d: -f1 /etc/passwd
compgen -u
```

Assigns a specific user ID (UID) to the new user.

```
sudo adduser --uid 1050 username
```

Assigns a specific group ID (GID) to the new user.

```
sudo adduser --gid 1001 username
```

Adds the new user to a specific existing group.

```
sudo adduser --ingroup <groupname> <username>
```

Creates a user without setting a password

```
sudo adduser --disabled-password username
```

Disables login for the new user

```
sudo adduser --disabled-login username
```

Create a user with home directory

```
->sudo useradd -m alice
```

Create user with specific shell /bin/zsh

```
->sudo useradd -s /bin/zsh bob
```

2.ADDGROUP

Create the group

```
sudo addgroup <groupname>
sudo groupadd <groupname>
```

Adding Users to a Group

```
sudo usermod -aG <groupname> <username>
```

List all groups:

```
getent group
```

Check the specific group:

```
getent group groupname
```

View the /etc/group file, which contains group information (This will show all groups, their GIDs, and the members associated with each group.**)**

```
cat /etc/group
```

To list only the group names

```
cut -d: -f1 /etc/group
```

3.USEDEL

To delete the specific user

```
sudo userdel <username>
```

Note:

-> Ensure that the user is not logged in or running processes, as userdel might fail in such cases.
->You can use the pkill command to terminate processes for the user before deletion.

Terminate the processes for the specific user you want to delete

```
sudo pkill -u username
```

(Now, we can delete the user using userdel)

4. USERMOD

Used to modify user account information

Syntax: sudo usermod [options] username

i. Change the User’s Login Name:

```
sudo usermod -l <newusername> <oldusername>
```

ii. Add User to a Group (if the user is already created, otherwise we need to create user first then we can add in group)

```
sudo usermod -aG <groupname> <username>
```

iii. Remove User from a Group:

```
sudo gpasswd -d username groupname
```

iv. Change the User’s Home Directory to new directory

```
sudo usermod -d /new/home/dir username
```

v. Change the User’s Home Directory and Move Content:

```
sudo usermod -d /new/home/dir -m username
```

List the users under specific group

```
getent group <groupname>
```

5. PASSWD

passwd command is used to change user passwords.

Changing Another User's Password

```
sudo passwd <username>
```

Force Password Change on Next Login

forces the user to change their password upon next login by expiring the current password.

```
sudo passwd -e <username>
```

Lock the passwod (disable the password)

```
sudo passwd -l <username>
```

Unlock the account (re-enable login)

Unlocks a user account that was previously locked using -l

```
sudo passwd -u username
```

Remove a user’s password (delete password**)**

```
sudo passwd -d username
```

Set password inactivity period

```
sudo passwd -i days username
```

Set the minimum number of days between password changes

```
sudo passwd -n days username
```

Set the maximum number of days a password is valid

(Sets the maximum number of days a password is valid. After this period, the user is forced to change their password.)

```
sudo passwd -x days username
```

Set a password warning period:

```
sudo passwd -w days username
```

(Sets the number of days before the password expires that the user will start receiving warnings to change it)

Display password status for a user:

```
sudo passwd -S username
```

(Displays the password status, including whether it’s locked, expired, or active)

SU (substitute user)

->used to switch to another user account

Switch to Another User

```
su <username>
```

Switch to Root User

```
su -i
```

Note:

We can directly switch to root user using **sudo -i** command

->In root user account we can run the above commands without using **sudo**

To logout from root user **exit**

Scenario based questions

To create a user without a home directory

```
sudo useradd -M sysuser
```

Creating a Group with a Specific Group ID (GID)

```
sudo groupadd -g 1005 managers
```

To delete a user and their home directory

```
sudo userdel -r <username>
```

Create a user john who should use /bin/bash as the default login shell.

```
useradd -s /bin/bash jane
```

Create a user without setting a password at the time of creation

```
sudo adduser --disabled-password username
```

To assign a user to multiple groups during creation

```
sudo adduser username group1,group2
```

Create a user with a home directory located at a custom path instead of the default /home/username

```
sudo adduser --home /custom/path/username username
```

Create a user with a specific comment or description in their account information

```
->sudo adduser --gecos "BridgeLabz , DevOps , 11111111 ," <username>
```

Change the ownership of directory '/etc/passwd' to the user 'ram'

```
->  sudo chown ram:webdev /etc/passwd
```

(Here webdev is group , other users of webdev group can also access the /etc/passwd directory but owner of that directory is ram)

Change the permission of the above directory '/etc/passwd' to 755

```
-> sudo chmod 755 /etc/passwd
```

We can check the ownership of the above directory '/etc/passwd' using ls command

```
-> ls -lh /etc/passwd
```

Customize userid of ram to 1111

```
->sudo usermod -u 1111 ram
```

Customize groupid of developer to 2222

```
->sudo groupmod -g 2222 developer
```

Remove the user ‘ram’ from ‘developer’ group

```
->sudo gpasswd -d ram developer
```

Create a user as ‘sham’ and add to the group ‘developer’ in single command

```
->sudo adduser -m -G developer sham
```

Create a user as ‘sham’ and add to the multiple groups ‘developer’ & ‘tester’ in single command

```
-> sudo adduser -m -G developer, tester sham
```

Create a user as john with a specific home directory instead of (/home/john)

```
-> sudo useradd -m -d /opt/john john
```

Force a user to change their password upon next login(-e expires the user’s password)

```
->sudo passwd -e alice
```

Change the username of an existing user

```
-> sudo usermod -l johnsmith john_doe
```

Create a new user as ram with no password

```
-> sudo adduser --disabled-password ram
```

Set the password for user ram

```
->sudo passwd ram
```

Find out which users are currently logged into the system and see their login information

```
-> who
```

Details of an existing user ram, including their group memberships

```
-> id ram
```

After creating the user ram, verify that user was successfully added to the system

```
->cat /etc/passwd | grep ram
```

Create a user lisa with a home directory /data/lisa and set their shell to /bin/bash.

```
-> sudo useradd -d /data/lisa -s /bin/bash -m lisa
```

Create the group projectteam, even if a group with the same name already exists.
->sudo groupadd -f projectteam

After creating a group projectteam, verify that the group was successfully created?
->cat /etc/group | grep projectteam

Sticky Bit:

It is a special permission that can be set on directories to control file deletion within that directory.

When set on a directory, the sticky bit ensures that only the file's owner, the directory's owner, or the superuser (root) can delete or rename files within that directory.

Setting sticky bit

->sudo chmod 1777 /common

The 1 at the beginning of the permission (e.g., 1777) indicates the sticky bit is set.

Verify Sticky Bit

->ls -ld /common

Output: drwxrwxrwt 2 root root 4096 Sep 19 15:00 /common

t at the end of the permissions (drwxrwxrwt) indicates the sticky bit is set.

Assignment 19/09/2024

1. List out 5 files in your system which consuming most of the disk space

   ```
   -> find -type f -exec du -h {} \;|sort -rh | head -n 5
   ```

2. Create one common folder in such a way that anyone can create files inside that independently and should not be able to delete other users’ files from that common folder.

   ```
   -> sudo mkdir /pushpa
   ->sudo chmod 1777 /common (not be able to delete other users’ files) Sticky bits
   ```

3. Create user name "shubham" and add that user in the group "adm"

   ```
   ->sudo useradd Shubham
   ->sudo groupadd adm
   ->sudo usermod -aG adm Shubham
   ```

   a) Create folder /data, change owner and group as "root:adm

   ```
   ->sudo mkdir /data
   ->sudo chown root:adm /data
   ```

   b) Change /data permission such a way that user can able to write data in this folder and ownership of files or folder which you creates in this folder should be same as parent folder i.e /data folder permission (root:adm)

   ```
   ->sudo chmod 277
   ```

Scenario 1: You are tasked with managing temporary users and roles on a Linux server for a short-term project.

1. Create Temporary User Accounts:

* A consultant, Michael Scott, is joining for a 2-month project. Create a user account with the username michaels.

```
->sudo useradd michaels
```

* Set an account expiration date 60 days from today.

```
-> sudo chage -E $(date -d "60 days" +%y-%m-%d) michaels
```

2. Create a Temporary Project Group:

* Create a group called temp_project to manage access for all temporary users.

```
->sudo groupadd temp_project
```

3. Assign Group Membership:

* Assign michaels to the temp_project group.

```
->sudo usermod –aG temp_project michaels
```

* Add another existing user janedoe to the temp_project group for collaborative work.

```
-> sudo usermod –aG temp_project janedoe
```

4. Modify Temporary User:

* Midway through the project, Michael needs access to additional resources. Add him to the developers group (which already exists).

```
-> sudo usermod –aG developers michaels
```

5. Early Account Termination:

* The project is completed earlier than expected, and Michael's account should be disabled immediately. Disable the account without deleting it, so it can be re-enabled later if needed.

```
->sudo usermod –L michaels
```

To unlock the Michael's account(Before unlocking we have to set the new password)

```
-> sudo usermod –U michaels
```

6. Post-Project Cleanup:

* After project completion, delete the temp_project group and remove all its members from the group. Ensure that michaels’ account is permanently removed along with his home directory and associated files.

Removes the members from group temp_project

```
->sudo gpasswd –d michaels temp_project
-> sudo gpasswd –d janedoe temp_project
```

Delete the group temp_project

```
->sudo groupdel  temp_project
```

Remove Michaels’ account is permanently

```
->sudo userdel –r michaels
```

Scenario 2: Due to a security incident, you need to take immediate action to lock down certain user accounts and enforce stricter password policies.

Lock User Accounts:

Lock the accounts of users: Adam (adam), Eve (eve), and Jack (jack) to prevent them from logging in during the investigation.

```
->sudo passwd –l adam
-> sudo passwd –l eve
->sudo passwd –l jack
```

Enforce Strong Password Policies:

Set a minimum password length of 12 characters for all users. Require all users to change their passwords immediately.

Account Auditing: Generate a list of all user accounts and their password status.

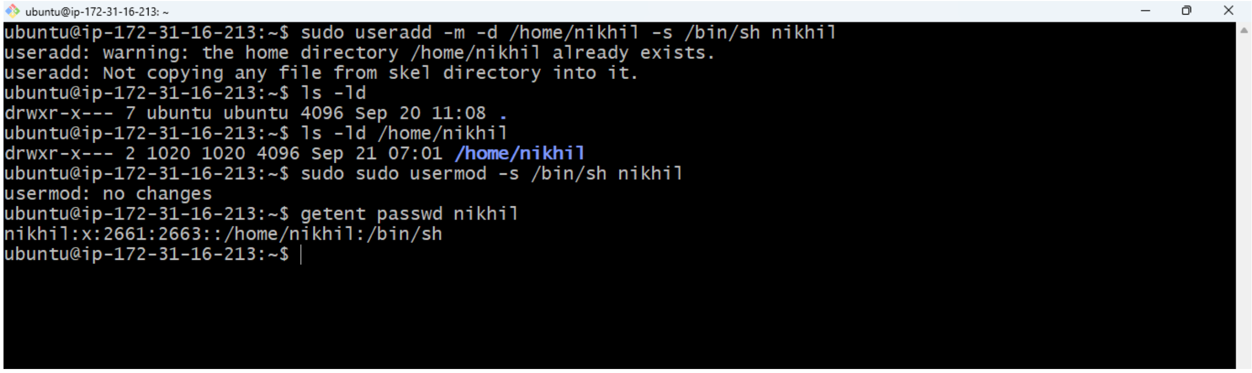
```
->sudo passwd –S adam
->sudo passwd –S eve
->sudo passwd –S jack
```

Assignment 2 - User Account Management

1) Create user "nikhil" with home directory set as "/home/nikhil"

a) nikhil user should have "/bin/sh" shell for his environment

->sudo adduser -d -m /home/nikhil -s /bin/sh

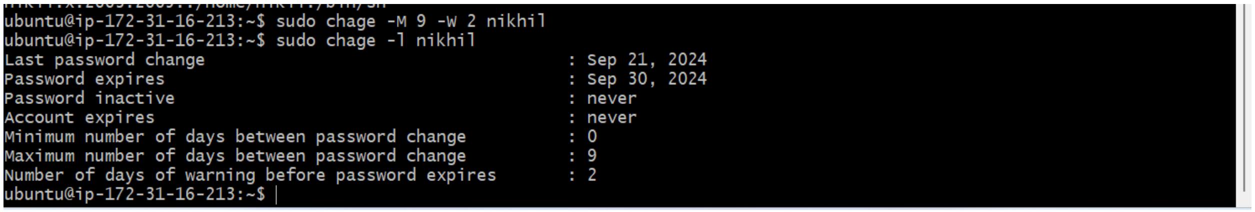


```
ubuntu@ip-172-31-16-213:~$ sudo useradd -m -d /home/nikhil -s /bin/sh nikhil
useradd: warning: the home directory /home/nikhil already exists.
useradd: Not copying any file from skel directory into it.
ubuntu@ip-172-31-16-213:~$ ls -ld
drwxr-x--- 7 ubuntu ubuntu 4096 Sep 20 11:08 .
ubuntu@ip-172-31-16-213:~$ ls -ld /home/nikhil
drwxr-x--- 2 1020 1020 4096 Sep 21 07:01 /home/nikhil
ubuntu@ip-172-31-16-213:~$ sudo sudo usermod -s /bin/sh nikhil
usermod: no changes
ubuntu@ip-172-31-16-213:~$ getent passwd nikhil
nikhil:x:2661:2663::/home/nikhil:/bin/sh
ubuntu@ip-172-31-16-213:~$ |
```

b) His password should expire in 9 days and 2 days before password expiry, he should get warning. User account must expire in 1 month from creation date

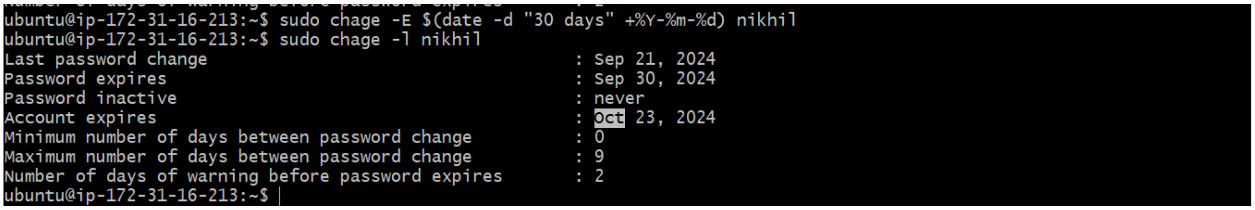
->sudo chage –M 9 –W 2 nikhil

->sudo chage –l nikhil   (to view the password information of nikhil)



```
nikhil:x:2663:2663::/home/nikhil:/bin/sh
ubuntu@ip-172-31-16-213:~$ sudo chage -M 9 -W 2 nikhil
ubuntu@ip-172-31-16-213:~$ sudo chage -l nikhil
Last password change                                    : Sep 21, 2024
Password expires                                        : Sep 30, 2024
Password inactive                                       : never
Account expires                                         : never
Minimum number of days between password change          : 0
Maximum number of days between password change          : 9
Number of days of warning before password expires       : 2
ubuntu@ip-172-31-16-213:~$ |
```

->sudo chage –E $(date –d “30 days” +%Y-%m-%d) nikhil



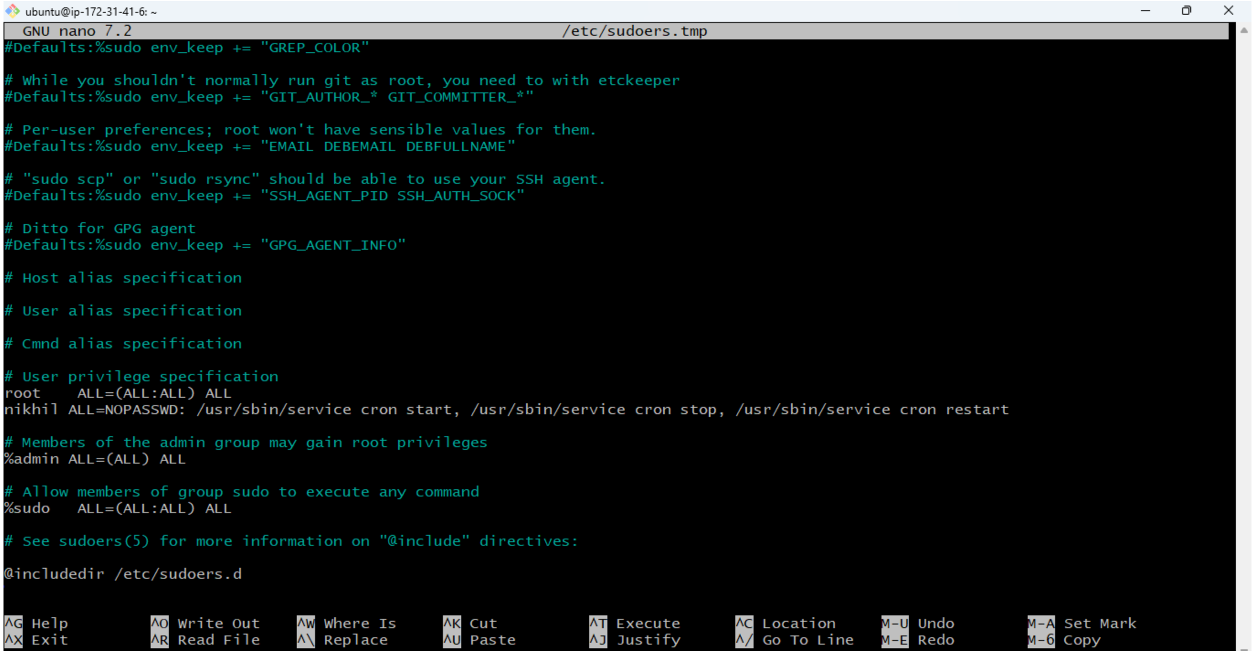
```
Number of days of warning before password expires       : 2
ubuntu@ip-172-31-16-213:~$ sudo chage -E $(date -d "30 days" +%Y-%m-%d) nikhil
ubuntu@ip-172-31-16-213:~$ sudo chage -l nikhil
Last password change                                    : Sep 21, 2024
Password expires                                        : Sep 30, 2024
Password inactive                                       : never
Account expires                                         : Oct 23, 2024
Minimum number of days between password change          : 0
Maximum number of days between password change          : 9
Number of days of warning before password expires       : 2
ubuntu@ip-172-31-16-213:~$ |
```

c) Give him root privileges to start/stop cron daemon.

->sudo visudo

Adding the line to User privilege specification

->nikhil ALL=NOPASSWD: /usr/sbin/service cron start, /usr/sbin/service cron stop, /usr/sbin/service cron restart

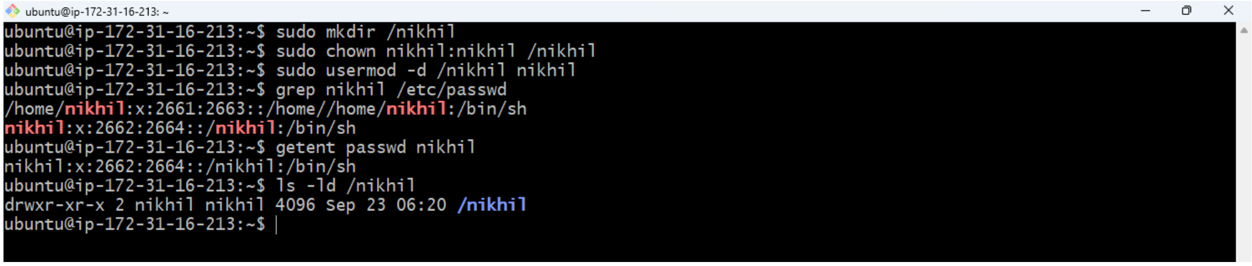


```
  GNU nano 7.2                                            /etc/sudoers.tmp
#Defaults:%sudo env_keep += "GREP_COLOR"

# While you shouldn't normally run git as root, you need to with etckeeper
#Defaults:%sudo env_keep += "GIT_AUTHOR_* GIT_COMMITTER_*"

# Per-user preferences; root won't have sensible values for them.
#Defaults:%sudo env_keep += "EMAIL DEBEMAIL DEBFULLNAME"

# "sudo scp" or "sudo rsync" should be able to use your SSH agent.
#Defaults:%sudo env_keep += "SSH_AGENT_PID SSH_AUTH_SOCK"

# Ditto for GPG agent
#Defaults:%sudo env_keep += "GPG_AGENT_INFO"

# Host alias specification

# User alias specification

# Cmnd alias specification

# User privilege specification
root    ALL=(ALL:ALL) ALL
nikhil ALL=NOPASSWD: /usr/sbin/service cron start, /usr/sbin/service cron stop, /usr/sbin/service cron restart

# Members of the admin group may gain root privileges
%admin ALL=(ALL) ALL

# Allow members of group sudo to execute any command
%sudo   ALL=(ALL:ALL) ALL

# See sudoers(5) for more information on "@include" directives:

@includedir /etc/sudoers.d

^G Help         ^O Write Out    ^W Where Is     ^K Cut          ^T Execute      ^C Location     M-U Undo        M-A Set Mark
^X Exit         ^R Read File    ^\ Replace      ^U Paste        ^J Justify      ^/ Go To Line   M-E Redo        M-6 Copy
```

2) Inside folder "/" , create new home directory as "nikhil" (/nikhil) and setup this folder as a home directory for user "Nikhil")

->sudo mkdir /nikhil

->sudo chmod –d /nikhil nikhil



```
ubuntu@ip-172-31-16-213:~$ sudo mkdir /nikhil
ubuntu@ip-172-31-16-213:~$ sudo chown nikhil:nikhil /nikhil
ubuntu@ip-172-31-16-213:~$ sudo usermod -d /nikhil nikhil
ubuntu@ip-172-31-16-213:~$ grep nikhil /etc/passwd
/home/nikhil:x:2661:2663::/home//home/nikhil:/bin/sh
nikhil:x:2662:2664::/nikhil:/bin/sh
ubuntu@ip-172-31-16-213:~$ getent passwd nikhil
nikhil:x:2662:2664::/nikhil:/bin/sh
ubuntu@ip-172-31-16-213:~$ ls -ld /nikhil
drwxr-xr-x 2 nikhil nikhil 4096 Sep 23 06:20 /nikhil
ubuntu@ip-172-31-16-213:~$ |
```