

User & Permissions in Ubuntu

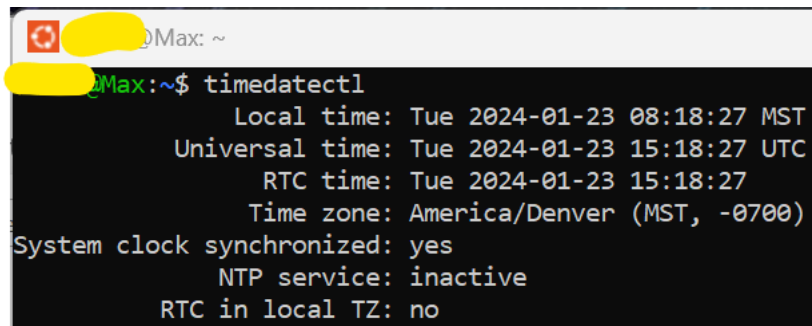
- **Note:** This project is provided by **Codecademy** (<https://www.codecademy.com>)
- **Note:** I did this project in the **Ubuntu** terminal.

Project description

I am the system administrator of a small start-up. I need to manage access levels on a Linux computer. The following three users need to work with their specific files on the Linux computer.

- **James** is a member of the Marketing team who needs full access to all the Marketing files.
- **Destiny** is a member of the Sales team who needs full access to all the Sales files.
- **Carolyn** is an administrator who needs read access to all files of all departments to monitor security and compliance.

Start Project Timestamp

A terminal window with a dark background. The prompt is 'Max: ~'. The command 'timedatectl' has been entered and executed. The output shows the local time as Tue 2024-01-23 08:18:27 MST, universal time as Tue 2024-01-23 15:18:27 UTC, RTC time as Tue 2024-01-23 15:18:27, and time zone as America/Denver (MST, -0700). It also indicates the system clock is synchronized, NTP service is inactive, and RTC is in local TZ.

```
Max: ~  
Max:~$ timedatectl  
    Local time: Tue 2024-01-23 08:18:27 MST  
    Universal time: Tue 2024-01-23 15:18:27 UTC  
        RTC time: Tue 2024-01-23 15:18:27  
    Time zone: America/Denver (MST, -0700)  
System clock synchronized: yes  
        NTP service: inactive  
    RTC in local TZ: no
```

Environment Setup

1. I first escalated my access level to the root using the **sudo -i** command and entering my password.

```
root@Max: ~
Max:~$ sudo -i
[sudo] password for Max:
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 5.15.133.1-microsoft-standard-WSL2 x86_64)

* Documentation:  https://help.ubuntu.com
* Management:    https://landscape.canonical.com
* Support:        https://ubuntu.com/advantage

* Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
  just raised the bar for easy, resilient and secure K8s cluster deployment.

  https://ubuntu.com/engage/secure-kubernetes-at-the-edge

This message is shown once a day. To disable it please create the
/root/.hushlogin file.
root@Max:~#
```

2. I created the groups for the marketing team, sales team, and the information technology team with the **groupadd** command. Then I verified the creation of the group with the **cat /etc/group** command (results highlighted in yellow).

```
root@Max:~# groupadd marketing-team -f
root@Max:~# groupadd sales-team -f
root@Max:~# groupadd it-team -f
root@Max:~# cat /etc/group
root:x:0:
daemon:x:1:
bin:x:2:
sys:x:3:
adm:x:4:syslog,msant
tty:x:5:
disk:x:6:
lp:x:7:
mail:x:8:
```

```
render:x:110:
syslog:x:111:
uudd:x:112:
tcpdump:x:113:
_ssh:x:114:
admin:x:115:
netdev:x:116:msant
msant:x:1000:
plocate:x:117:
marketing-team:x:1001:
sales-team:x:1002:
it-team:x:1003:
root@Max:~#
```

3. I then created and added the users to the group names with the **useradd** command. I also added **Carolyn** to the **sudo** group to have full environmental access with the **adduser** command.
- **Note:** The users are promptly added to their proper groups using the **-g** option. A new **home** directory is created for each new user since the **-m** flag is enabled.

```
root@Max: ~
root@Max:~# useradd james -g marketing-team -m
root@Max:~# useradd destiny -g sales-team -m
root@Max:~# useradd carolyn -g it-team -m
root@Max:~# adduser carolyn sudo
Adding user 'carolyn' to group 'sudo' ...
Adding user carolyn to group sudo
Done.
root@Max:~#
```

4. The newly created **home** directories for the users can be observed using the **ls /home/** command.

```
root@Max:~# ls /home/
carolyn  destiny  james  msant
```

- a. Verify that the users are added to their proper group by listing the groups for each user using the **groups [username]** command. Example shown below:

```
carolyn destiny james
root@Max:~# groups james
james : marketing-team
root@Max:~#
```

5. To create separate directories for each department, I download Codecademy's available archive of pre-created directories using the **wget** command (highlighted in yellow). Alternatively, the archive can be downloaded using the **curl** command. The archive website to download is:

<https://static-assets.codecademy.com/Courses/learn-linux/linux-shell-utilities/project-data.tar.gz>

```
root@Max:~# wget https://static-assets.codecademy.com/Courses/learn-linux/linux-shell-utilities/project-data.tar.gz
--2024-01-23 08:32:58-- https://static-assets.codecademy.com/Courses/learn-linux/linux-shell-utilities/project-data.tar
.gz
Resolving static-assets.codecademy.com (static-assets.codecademy.com)... 104.17.212.81, 104.18.199.63, 2606:4700::6811:d
451, ...
Connecting to static-assets.codecademy.com (static-assets.codecademy.com)|104.17.212.81|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 9081 (8.9K) [application/x-gzip]
Saving to: 'project-data.tar.gz'

project-data.tar.gz      100%[=====>]  8.87K  --.-KB/s   in 0s
2024-01-23 08:32:58 (92.5 MB/s) - 'project-data.tar.gz' saved [9081/9081]
```

6. I extracted the archive to create the data directory and the sub-directories for the department with the following command:

```
root@Max:~# tar -xzvf project-data.tar.gz -C /tmp/
data/
data/IT/
data/IT/all_users.txt
data/IT/inventory_template.txt
data/Marketing/
data/Marketing/marketing_plan.txt
data/Marketing/social_media_strategy.txt
data/Sales/
data/Sales/contract_template.txt
data/Sales/sales_records.csv
```

Note: This command extracts the tar ball and creates the directory structure in the **/tmp** directory, which is a temporary directory in Linux where contents are automatically wiped out every 10 days.

7. The creation of files and directories can then be verified by visualizing the directory structure using the following command that relies on **sed**, a stream editor for filtering and transforming text:

```
find /tmp/data | sed -e "s/[^\-][^\-]*\// |/" -e "s/|\\([^\- ]\\)/|-\1/"
```

or using the **tree utility**. Which can be installed if not done so:

```
tree -d /tmp/data
```

The file structure looks like the following:

```
root@Max:~# find /tmp/data | sed -e "s/[^-][^\\/]*/|/g" -e "s/|\\([^- ]\\)/|-\\1/"
|-data
| |-Sales
| | |-sales_records.csv
| | |-contract_template.txt
| |-Marketing
| | |-marketing_plan.txt
| | |-social_media_strategy.txt
| |-IT
| | |-inventory_template.txt
| | |-all_users.txt
```

```
root@Max:~# tree -d /tmp/data
/tmp/data
├── IT
├── Marketing
└── Sales
3 directories
```

9. I modified the file permissions using the **chown** command to reflect the requirements for each user and group. For example, I reflected **marketing-team** as the owner of the **Marketing** group (**/tmp/data/Marketing**). The same was done for the **sales-team** and **it-team**. Which is shown below

```
root@Max:~# chown :marketing-team /tmp/data/Marketing
root@Max:~# chown :sales-team /tmp/data/Sales
root@Max:~# chown :it-team /tmp/data/IT
```

10. I then adjusted the permission levels for each directory with the **chmod** command. Read, write, and execute for the **user** and **group**. Read only for **others**. Which is shown below.

```
root@Max:~# chmod -R 774 /tmp/data/Marketing
root@Max:~# chmod -R 774 /tmp/data/Sales
root@Max:~# chmod -R 774 /tmp/data/IT
```

11. The owner group for each directory can be verified with the **ls** command if you list all directories in **/tmp/data**. Which is highlighted in yellow.

```
root@Max:~# ls /tmp/data -l
total 12
drwxrwxr-- 2 root it-team      4096 Mar  3  2022 IT
drwxrwxr-- 2 root marketing-team 4096 Mar  3  2022 Marketing
drwxrwxr-- 2 root sales-team    4096 Mar  3  2022 Sales
```

Access Verification

1. I then impersonated a user to verify their level of access. I started with **James**.

```
root@Max:~# su james
$
```

2. I had access to the Marketing directory as **James**'.

```
$ ls /tmp/data/Marketing
marketing_plan.txt  social_media_strategy.txt
```

3. The data of one of the file can be read with the **cat** command. Example shown below:

```
$ cat /tmp/data/Marketing/marketing_plan.txt
Marketing Plan

The file outlines the 2022 plan of the marketing team. The plan includes four phases as described below.

Phase 1. Identify prospects in North America.
Phase 2. Identify the competition service the prospects use.
Phase 3. Identify a point of contact for each prospect.
Phase 4. Run a targeted campaign against the points of contact.$
$
```

4. When trying to read into the Sales directory, which **James** shouldn't have access to, this is the result:

```
$ ls /tmp/data/Sales/contract_template.txt
ls: cannot access '/tmp/data/Sales/contract_template.txt': Permission denied
```

5. I logged into **Destiny**, and this is the result:

```
$ exit
root@Max:~# su destiny
$ ls /tmp/data/Sales/contract_template.txt
/tmp/data/Sales/contract_template.txt
```

End Project Timestamp

```
$ timedatectl
Local time: Tue 2024-01-23 10:18:26 MST
Universal time: Tue 2024-01-23 17:18:26 UTC
RTC time: Tue 2024-01-23 17:32:47
Time zone: America/Denver (MST, -0700)
System clock synchronized: no
NTP service: inactive
RTC in local TZ: no
```