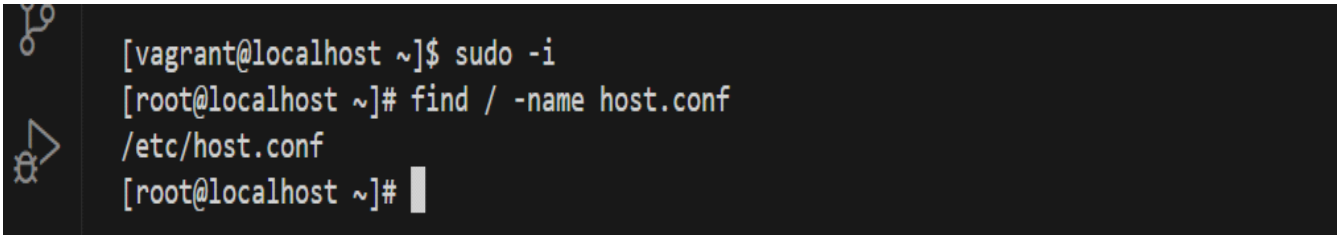
ITYOURWAY – PROJECT ONE

During a deployment to production, an error occurred: "/tmp/deployment/uat/host.conf": No such file or directory, Deployment failed! It seems the deployment script is unable to access the "host.conf" file.

TASK 1

1. Verify if the path "/opt/deployment/uat/host.conf" is correct.



2. If incorrect, provide the correct path to the developers.

3. Create a file named ityourway_project.txt inside your home directory and add the sentence: “This is group nine (9) project presentation and we are staying focused till the end of the February session.”

NOTE: Change Group Nine (9) to your group number.

```
[root@localhost ~]# echo "This is group two (2) project presentation and we are staying focused till the end of the February session"
 > ityourway_project.txt
[root@localhost ~]# cat ityourway_project.txt
This is group two (2) project presentation and we are staying focused till the end of the February session
[root@localhost ~]#
```

4. Determine the group ownership of the “ityourway_project.txt” file.

```
[root@localhost ~]# ll
total 20
-rw-------. 1 root root 5570 Apr 30  2020 anaconda-ks.cfg
-rw-r--r--. 1 root root  107 Mar 27 10:26 ityourway_project.txt
-rw-------. 1 root root 5300 Apr 30  2020 original-ks.cfg
[root@localhost ~]#
```

5. Check the permissions of the “ityourway_project.txt” file for OTHERS (non-owner, non-group members).

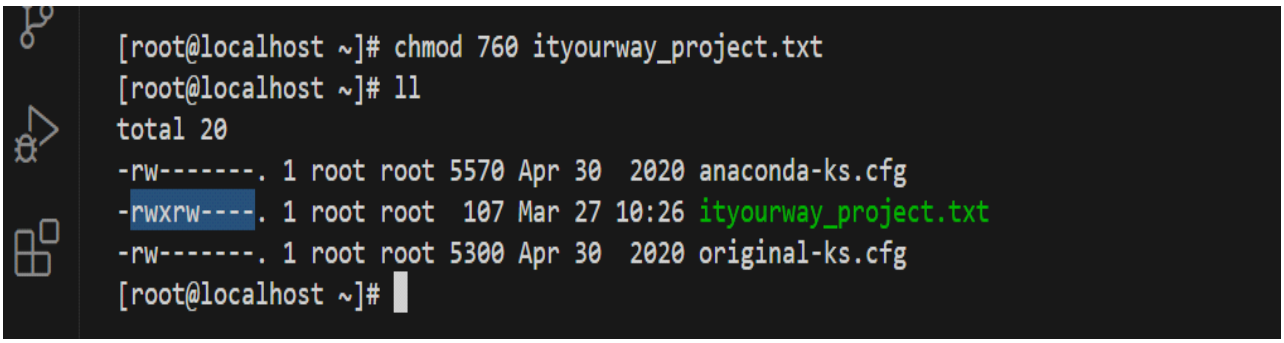
```
[root@localhost ~]# ll
total 20
-rw-------. 1 root root 5570 Apr 30  2020 anaconda-ks.cfg
-rw-r--r--. 1 root root  107 Mar 27 10:26 ityourway_project.txt
-rw-------. 1 root root 5300 Apr 30  2020 original-ks.cfg
[root@localhost ~]# Others have read only permission
```

6. Change the permissions on the ityourway_project.txt file as follows:

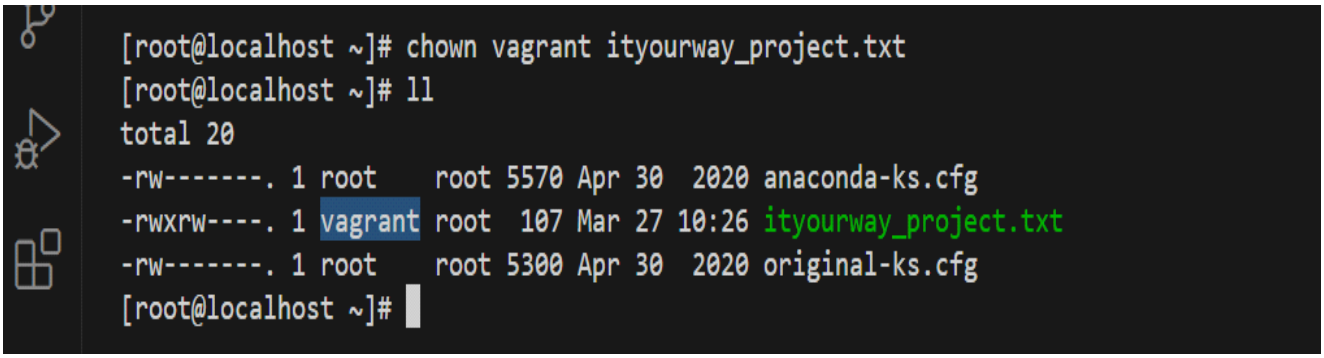
Owner: Full permissions

Group: Read and write permissions

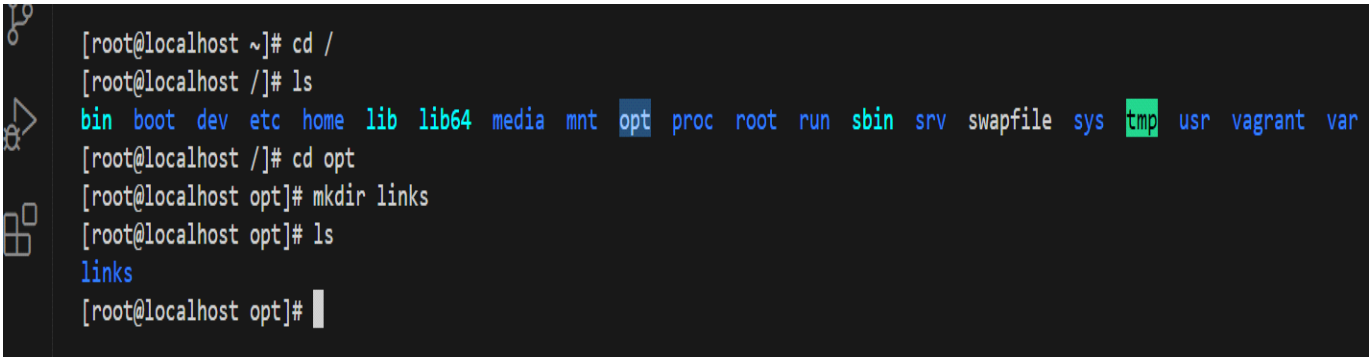
Others: No permissions



7. Change the owner of the ityourway_project.txt file to the user vagrant".



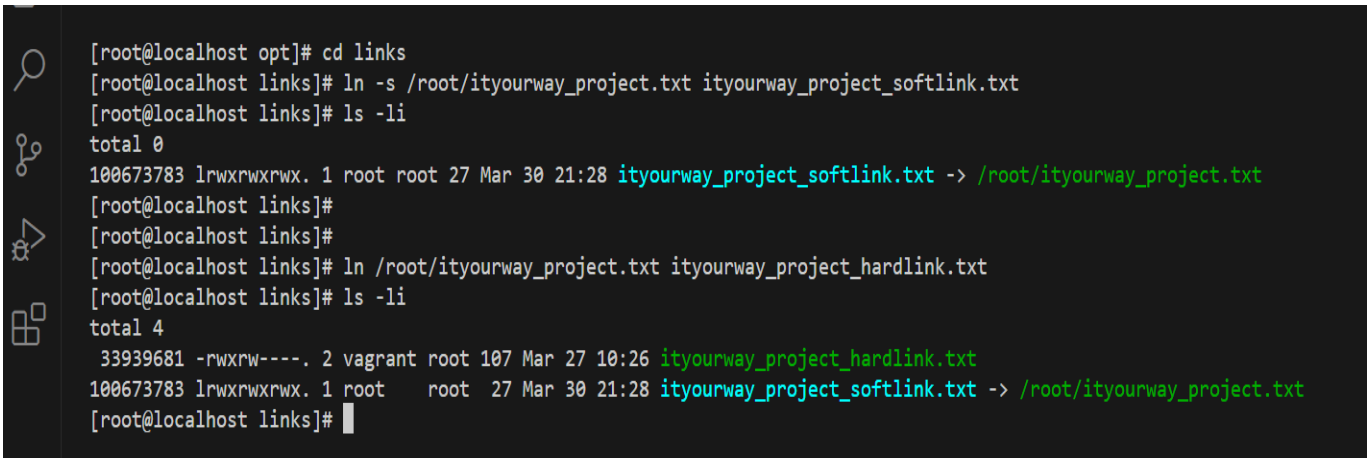
8. There is a directory in the root directory called “opt”. In the opt directory, create another directory called “links”



9. In the “links” directory:

– create a softlink for ityourway_project.txt and place it here in the “links” directory. You can call this new softlink file as “ityourway_project_softlink.txt “

– create a hardlink for ityourway_project.txt and place it here in the “links” directory. You can call this new hardlink file as ityourway_project_hardlink.txt



10. What can you say about the inode numbers of the original file (ityourway_project.txt) compared to the inode numbers of the link files (ityour_project_softlink.txt and ityour_project_hardlink.txt).

```
[root@localhost links]# ls -li /root/ityourway_project.txt
33939681 -rwxrw----. 2 vagrant root 107 Mar 27 10:26 /root/ityourway_project.txt
[root@localhost links]# ls -li
total 4
 33939681 -rwxrw----. 2 vagrant root 107 Mar 27 10:26 ityourway_project_hardlink.txt
100673783 lrwxrwxrwx. 1 root    root  27 Mar 30 21:28 ityourway_project_softlink.txt -> /root/ityourway_project.txt
[root@localhost links]# The hardlink and the original file have the same inode numbers which is “33939681” while the inode number of
 the softlink, “100673783”, is different from that of the original file.
```

11. In the “passwd“ file, use the awk and grep command to print out ONLY the username of the root user on the server

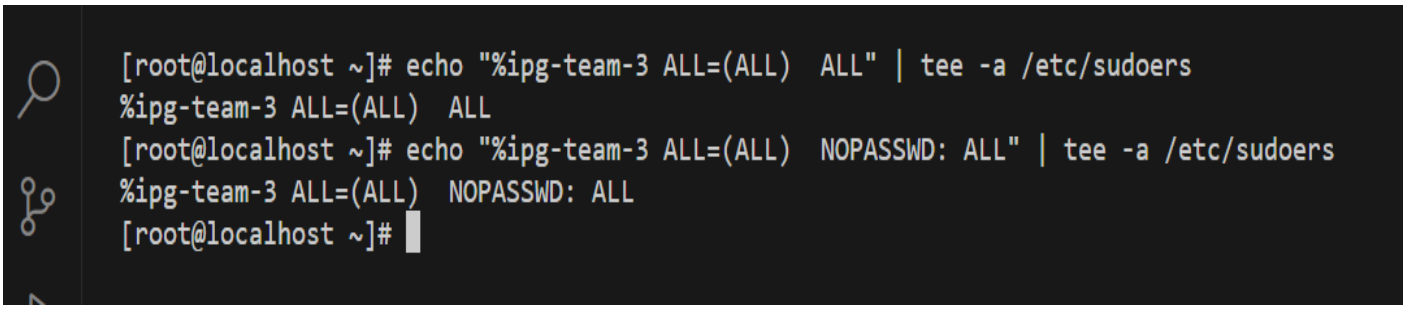
```
[root@localhost links]# awk -F ":" '{print $1}' /etc/passwd | grep root
root
[root@localhost links]#
```

12. Assuming there is a group on the server called “ipg-team-3”, use the “echo” command or the “sed” command to either ADD two (2) lines or INSERT two (2) lines into the “sudoers” file.

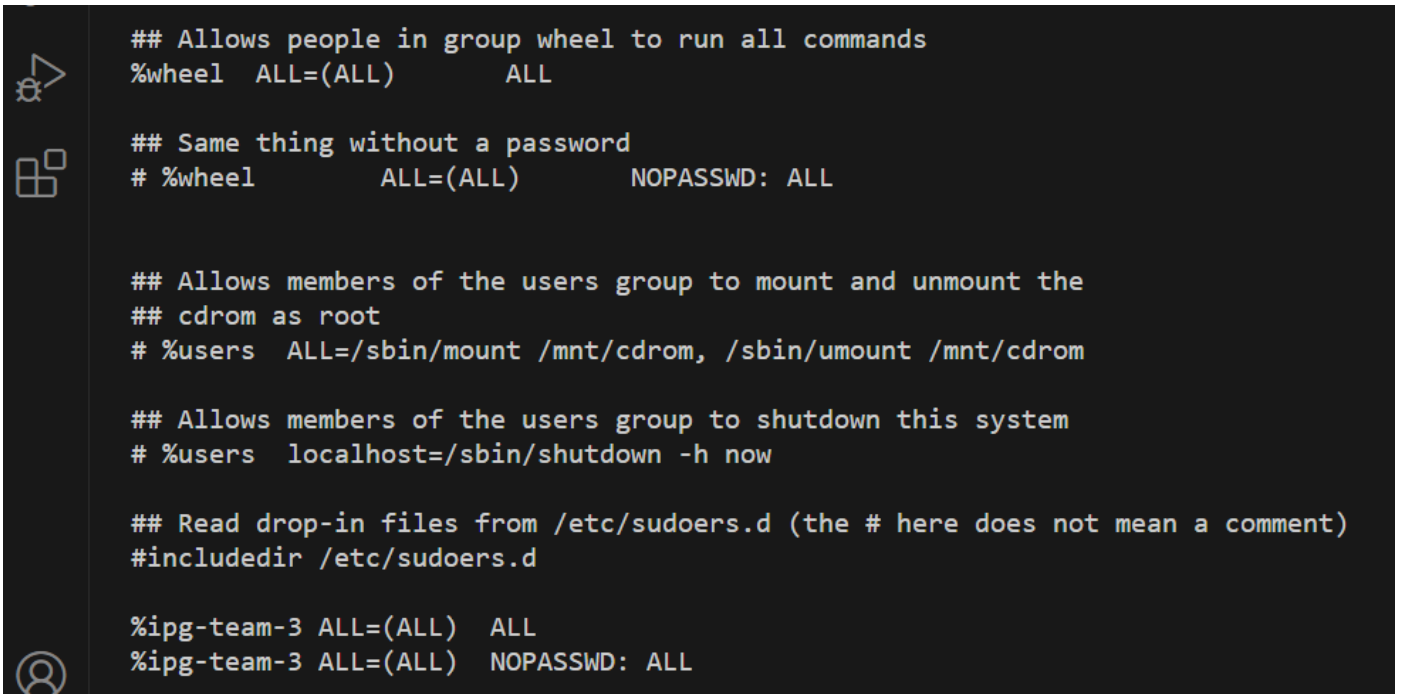
– One line must give the group: ipg-team-3 permissions to run sudo commands

– The other line must give the group: ipg-team-3 permissions to run sudo commands with No Passwords.

(NOTE: EXAMPLES OF BOTH LINES ARE FOUND IN THE SUDOERS FILES)



```
[root@localhost ~]# echo "%ipg-team-3 ALL=(ALL)  ALL" | tee -a /etc/sudoers
%ipg-team-3 ALL=(ALL)  ALL
[root@localhost ~]# echo "%ipg-team-3 ALL=(ALL)  NOPASSWD: ALL" | tee -a /etc/sudoers
%ipg-team-3 ALL=(ALL)  NOPASSWD: ALL
[root@localhost ~]#
```



```
## Allows people in group wheel to run all commands
%wheel  ALL=(ALL)       ALL

## Same thing without a password
# %wheel        ALL=(ALL)       NOPASSWD: ALL


## Allows members of the users group to mount and unmount the
## cdrom as root
# %users  ALL=/sbin/mount /mnt/cdrom, /sbin/umount /mnt/cdrom

## Allows members of the users group to shutdown this system
# %users  localhost=/sbin/shutdown -h now

## Read drop-in files from /etc/sudoers.d (the # here does not mean a comment)
#includedir /etc/sudoers.d

%ipg-team-3 ALL=(ALL)  ALL
%ipg-team-3 ALL=(ALL)  NOPASSWD: ALL
```

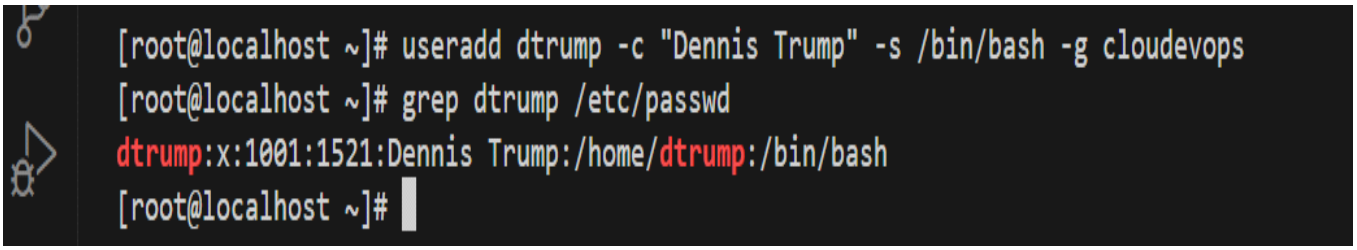
TASK 2

Your Company has just employed a new DevOps/Cloud Engineer.

As part of your task as a Linux Systems Administrator, you have been tasked to get the engineer set up in the company by following the below procedures:

1. The user will be added to a group called “cloudevops“, check on the system if this group exists. If it does not, please create this group with a group id of “1521“.

2. Create the new user to have a username = “dtrump”, full name = “Dennis Trump”, shell = /bin/bash and add the user to the newly created “cloudevops” group.



3. Since the “cloudevops” group was newly created, all members in this group may not be able to run sudo commands, Give sudo privileges to all members of the “cloudevops” group.

(NOTE: ONLY GIVE THIS SUDO PRIVILEGE TO THE GROUP (cloudevops) and NOT INDIVIDUALLY TO THE MEMBERS OF THE GROUP (IN THIS WAY WHEN A NEW MEMBER GETS ADDED TO THE GROUP, they will automatically INHERIT THE SUDO PRIVILEGE.)

```
[root@localhost ~]# echo "%cloudevops ALL=(ALL)  ALL" | tee -a /etc/sudoers
%cloudevops ALL=(ALL)  ALL
[root@localhost ~]#
```

```
## Allows people in group wheel to run all commands
%wheel  ALL=(ALL)       ALL

## Same thing without a password
# %wheel        ALL=(ALL)       NOPASSWD: ALL

## Allows members of the users group to mount and unmount the
## cdrom as root
# %users  ALL=/sbin/mount /mnt/cdrom, /sbin/umount /mnt/cdrom

## Allows members of the users group to shutdown this system
# %users  localhost=/sbin/shutdown -h now

## Read drop-in files from /etc/sudoers.d (the # here does not mean a comment)
#includedir /etc/sudoers.d

%ipg-team-3 ALL=(ALL)  ALL
%ipg-team-3 ALL=(ALL)  NOPASSWD: ALL
%cloudevops ALL=(ALL)  ALL
```

4. Also Allow members of the newly created group to be able to run the sudo command without password – NO PASSWORD

```
[root@localhost ~]# echo "%cloudevops ALL=(ALL)  NOPASSWD: ALL" | tee -a /etc/sudoers
%cloudevops ALL=(ALL)  NOPASSWD: ALL
[root@localhost ~]#
```

```
## Same thing without a password
# %wheel        ALL=(ALL)       NOPASSWD: ALL

## Allows members of the users group to mount and unmount the
## cdrom as root
# %users  ALL=/sbin/mount /mnt/cdrom, /sbin/umount /mnt/cdrom

## Allows members of the users group to shutdown this system
# %users  localhost=/sbin/shutdown -h now

## Read drop-in files from /etc/sudoers.d (the # here does not mean a comment)
#includedir /etc/sudoers.d

%ipg-team-3 ALL=(ALL)  ALL
%ipg-team-3 ALL=(ALL)  NOPASSWD: ALL
%cloudevops ALL=(ALL)  ALL
%cloudevops ALL=(ALL)  NOPASSWD: ALL
```

5. Once these privileges have been given to the “cloudevops” group, SWITCH to the new user and run the below commands:

command 1 = whoami (To show the username)

```
[root@localhost ~]# su dtrump
[dtrump@localhost root]$ whoami
dtrump
[dtrump@localhost root]$
```

command 2 = id (This will display the id of the user and give us a bit more information about the user.)

```
[dtrump@localhost root]$ id
uid=1001(dtrump) gid=1521(cloudevops) groups=1521(cloudevops) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[dtrump@localhost root]$
```

command 3 = sudo -l (This command will display the allowed (and forbidden) commands for the user)

NOTE: With the “sudo -l” the -l here is a lowercase of L and NOT an “i”.

```
[dtrump@localhost root]$ sudo -l
Matching Defaults entries for dtrump on localhost:
    !visiblepw, always_set_home, match_group_by_gid, always_query_group_plugin, env_reset, env_keep="COLORS DISPLAY HOSTNAME
    HISTSIZE KDEDIR LS_COLORS", env_keep+="MAIL PS1 PS2 QTDIR USERNAME LANG LC_ADDRESS LC_CTYPE", env_keep+="LC_COLLATE
    LC_IDENTIFICATION LC_MEASUREMENT LC_MESSAGES", env_keep+="LC_MONETARY LC_NAME LC_NUMERIC LC_PAPER LC_TELEPHONE",
    env_keep+="LC_TIME LC_ALL LANGUAGE LINGUAS _XKB_CHARSET XAUTHORITY", secure_path=/sbin\:/bin\:/usr/sbin\:/usr/bin

User dtrump may run the following commands on localhost:
    (ALL) ALL
    (ALL) NOPASSWD: ALL
[dtrump@localhost root]$
```

TASK 3: AUTOMATING THE MANUAL SET UP OF THE USERS (NEW EMPLOYEES)

As a Linux Systems Administrator, you've frequently onboarded new DevOps/Cloud engineers into the company manually (As seen in TASK 2). To streamline this process and ensure consistency, you have decided to automate the setup process using a shell script. Your goal is to create a script that can handle user and group creation, assign sudo privileges, and perform necessary configurations for new employees.

Here's what the script should accomplish and some clues to follow:

1. Check if the "group_name" group exists. If not, create it with a group ID of "group_id".

```
[root@localhost ~]# touch group2.sh
[root@localhost ~]# vi group2.sh
```

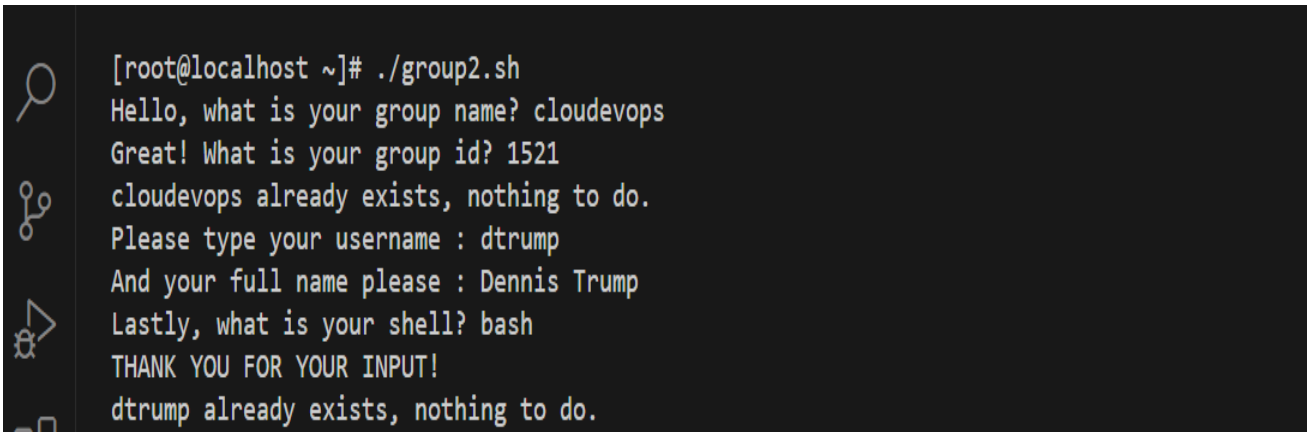
```
# Ask user for group name and id
read -p "Hello, what is your group name? " group_name
sleep 2
read -p "Great! What is your group id? " group_id
sleep 2

# Command to check if group name exists assigned to a variable
group=`grep $group_name /etc/group`

# Check if group exists or not
if [ $group ]; then
    echo "$group_name already exists, nothing to do."
else
    groupadd "$group_name" -g "$group_id"
    echo "Group created."
fi
```

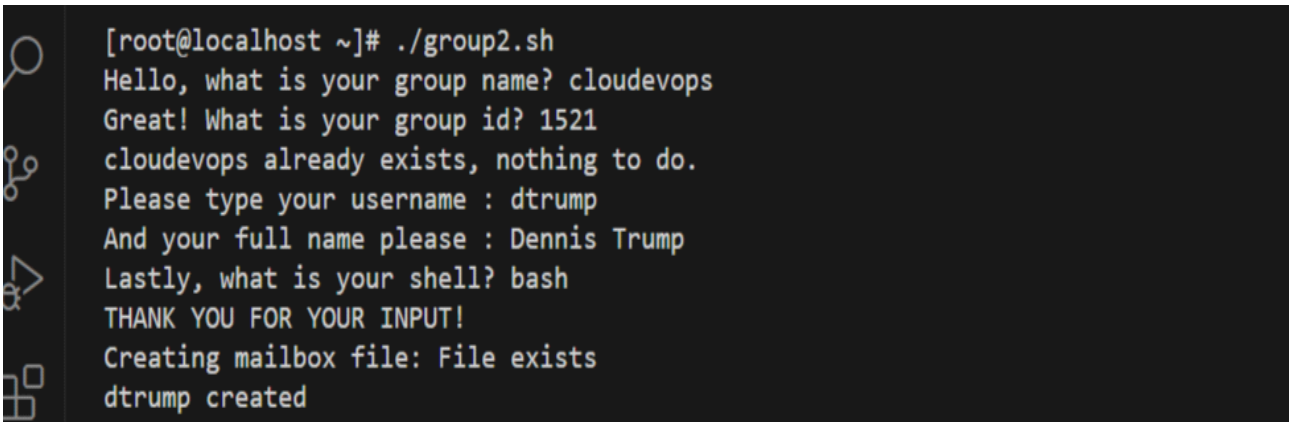
```
[root@localhost ~]# ./group2.sh
Hello, what is your group name? cloudevops
Great! What is your group id? 1521
cloudevops already exists, nothing to do.
Please type your username :
```

```
[root@localhost ~]# ./group2.sh
Hello, what is your group name? cloudevops
Great! What is your group id? 1521
Group created.
Please type your username :
```

2. Create a new user named “username” with full name, assigned to the “group_name” group, and with a shell as “shell”.

```
# Ask user for user for input
read -p "Please type your username : " username
sleep 2
read -p "And your full name please : " full_name
sleep 2
read -p "Lastly, what is your shell? " shell
sleep 2

echo "THANK YOU FOR YOUR INPUT!"

# Command to check if username exists assigned to a variable
user=`grep $username /etc/passwd`

# Check if user exists
if [[ $user ]]; then
    echo "$username already exists, nothing to do."
else
    useradd "$username" -c "$full_name" -g "$group_name" -s /bin/"$shell"
    echo "$username created"
fi
```


[root@localhost ~]# ./group2.sh
Hello, what is your group name? cloudevops
Great! What is your group id? 1521
cloudevops already exists, nothing to do.
Please type your username : dtrump
And your full name please : Dennis Trump
Lastly, what is your shell? bash
THANK YOU FOR YOUR INPUT!
dtrump already exists, nothing to do.


[root@localhost ~]# ./group2.sh
Hello, what is your group name? cloudevops
Great! What is your group id? 1521
cloudevops already exists, nothing to do.
Please type your username : dtrump
And your full name please : Dennis Trump
Lastly, what is your shell? bash
THANK YOU FOR YOUR INPUT!
Creating mailbox file: File exists
dtrump created

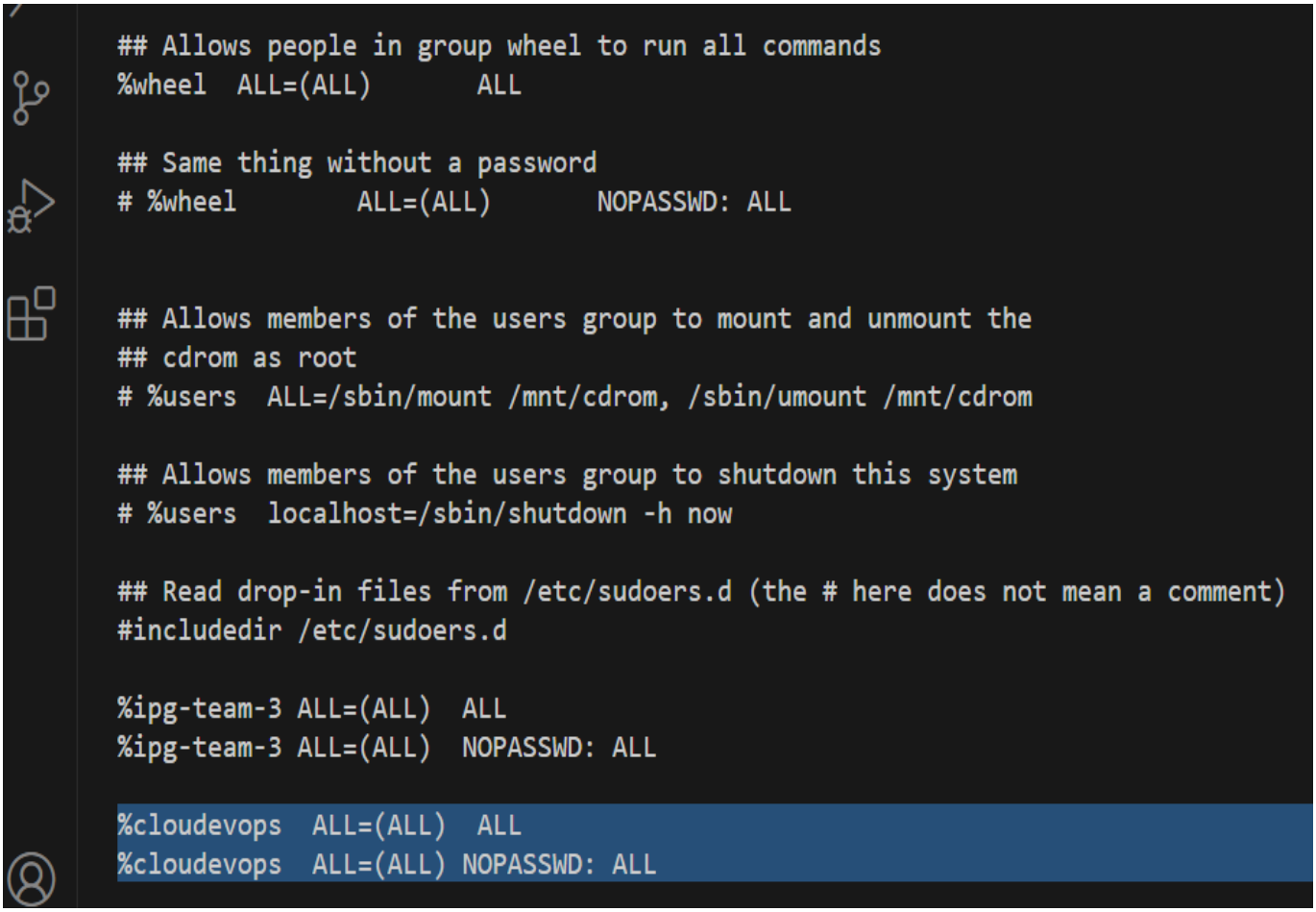
3. Grant sudo privileges to the “group_name” group by configuring the sudoers
file

```
# Give sudo privileges to "cloudevops" as a group

  # Command to find sudo privilege assigned to a variable
sudo_privilege=`grep $group_name /etc/sudoers`

  # Check if sudo privilege is granted or not
if [[ $sudo_privilege ]]; then
    echo "Sudo privilege already granted to $group_name"
else
   echo "%$group_name  ALL=(ALL)  ALL" >> /etc/sudoers
   echo "%$group_name  ALL=(ALL) NOPASSWD: ALL" >> /etc/sudoers
   echo "Sudo privilege now granted to $group_name."
fi

sleep 4
```

```
Lastly, what is your shell? bash
THANK YOU FOR YOUR INPUT!
dtrump already exists, nothing to do.
Sudo privilege already granted to cloudevops
```

```
[root@localhost ~]# ./group2.sh
Hello, what is your group name? cloudevops
Great! What is your group id? 1521
cloudevops already exists, nothing to do.
Please type your username : dtrump
And your full name please : Dennis Trump
Lastly, what is your shell? bash
THANK YOU FOR YOUR INPUT!
dtrump already exists, nothing to do.
Sudo privilege now granted to cloudevops.
```

4. Allow members of the “group_name” group to run sudo commands without a password prompt. (Check sudoers file for how this is done)



```
## Allows people in group wheel to run all commands
%wheel  ALL=(ALL)       ALL

## Same thing without a password
# %wheel        ALL=(ALL)       NOPASSWD: ALL


## Allows members of the users group to mount and unmount the
## cdrom as root
# %users  ALL=/sbin/mount /mnt/cdrom, /sbin/umount /mnt/cdrom

## Allows members of the users group to shutdown this system
# %users  localhost=/sbin/shutdown -h now

## Read drop-in files from /etc/sudoers.d (the # here does not mean a comment)
#includedir /etc/sudoers.d

%ipg-team-3 ALL=(ALL)  ALL
%ipg-team-3 ALL=(ALL)  NOPASSWD: ALL

%cloudevops  ALL=(ALL)  ALL
%cloudevops  ALL=(ALL) NOPASSWD: ALL
```

5. Switch to the new user “username” and perform the following commands:

– whoami: (This will display the username).

– id: (This will display additional user information).

– sudo -l: (This will display the allowed (and forbidden) commands for the user).

NOTE : Do a little research to perform number 5 within your script using a “HereDoc in bash scripting”

```
echo "Please find below the username, id of the user and a list of allowed and forbidden commands for the user:"

sleep 4

# Using the HereDoc to execute the following commands
sudo -s << COMMAND
su ${username}
whoami
id
sudo -l
COMMAND


# END
```

```
Please find below the username, id of the user and a list of allowed and forbidden commands for the user:
dtrump
uid=1001(dtrump) gid=1521(cloudevops) groups=1521(cloudevops) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
Matching Defaults entries for dtrump on localhost:
    !visiblepw, always_set_home, match_group_by_gid, always_query_group_plugin, env_reset, env_keep="COLORS DISPLAY HOSTNAME HISTSIZE
    KDEDIR LS_COLORS", env_keep+="MAIL PS1 PS2 QTDIR USERNAME LANG LC_ADDRESS LC_CTYPE", env_keep+="LC_COLLATE LC_IDENTIFICATION
    LC_MEASUREMENT LC_MESSAGES", env_keep+="LC_MONETARY LC_NAME LC_NUMERIC LC_PAPER LC_TELEPHONE", env_keep+="LC_TIME LC_ALL LANGUAGE
    LINGUAS _XKB_CHARSET XAUTHORITY", secure_path=/sbin\:/bin\:/usr/sbin\:/usr/bin

User dtrump may run the following commands on localhost:
    (ALL) ALL
    (ALL) NOPASSWD: ALL
[root@localhost ~]#
```