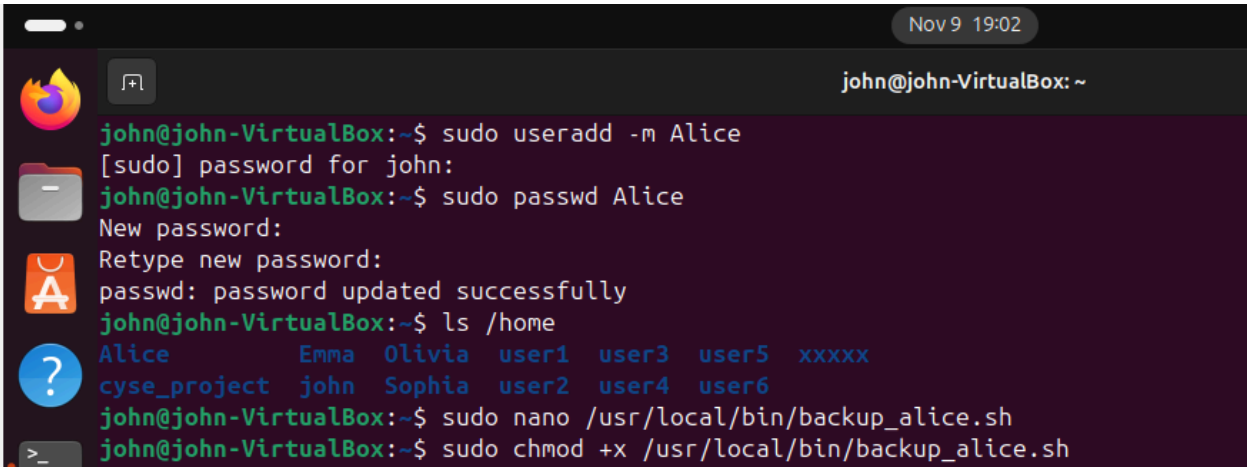
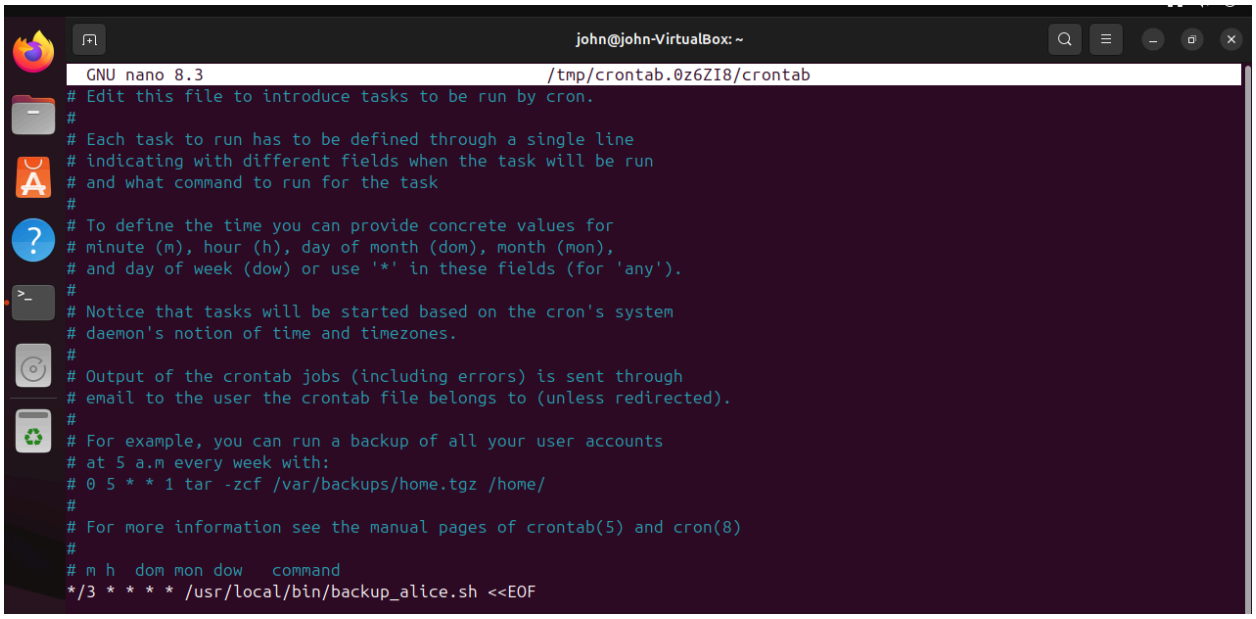
Steps 1-2:



Step 3:



Step 4:

```
john@john-VirtualBox:~$ sudo ls -lh /var/backups
total 2.2M
-rw-r--r-- 1 root root  60K Oct 12 16:53 alternatives.tar.0
-rw-r--r-- 1 root root 2.7K Sep  2 18:31 alternatives.tar.1.gz
-rw-r--r-- 1 root root  95K Oct  5 20:10 apt.extended_states.0
-rw-r--r-- 1 root root  11K Aug 31 18:10 apt.extended_states.1.gz
-rw-r--r-- 1 root root    0 Oct 12 16:53 dpkg.arch.0
-rw-r--r-- 1 root root   32 Sep  2 18:31 dpkg.arch.1.gz
-rw-r--r-- 1 root root 1.5K Aug 31 18:10 dpkg.diversions.0
-rw-r--r-- 1 root root  323 Aug 31 18:10 dpkg.diversions.1.gz
-rw-r--r-- 1 root root  200 Apr 15  2025 dpkg.statoverride.0
-rw-r--r-- 1 root root  168 Apr 15  2025 dpkg.statoverride.1.gz
-rw-r--r-- 1 root root 1.6M Oct  5 20:10 dpkg.status.0
-rw-r--r-- 1 root root 370K Aug 31 18:11 dpkg.status.1.gz
-rw-r--r-- 1 root root 2.3K Nov  9 19:21 John-2025.11.09-19.21.01.tar.gz
john@john-VirtualBox:~$
```

Step 5:

```
john@john-VirtualBox:~$ sudo crontab -l
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h  dom mon dow   command


john@john-VirtualBox:~$
```

Extra credit:


Nov 9 19:33
john@john-VirtualBox: ~
GNU nano 8.3 /usr/local/bin/cleanup_backups.sh
backup_dir="/var/backups"
threshold=3
count=$(ls -l ${backup_dir}/*.tar.gz 2>/dev/null | wc -l)
if ["$count" -gt "$threshold"]; then
echo "Found $count backups. Keeping newest $threshold..."
ls -1t ${backup_dir}/*.tar.gz | tail -n +$((threshold + 1)) | xargs sudo rm -f
echo "Old backups deleted."
else
echo "Backup count ($count) is under threshold. Nothing to delete."
fi