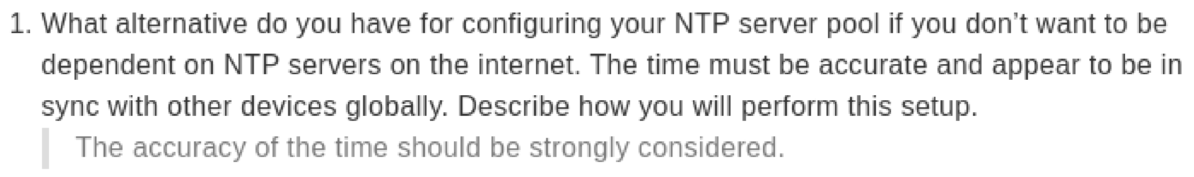
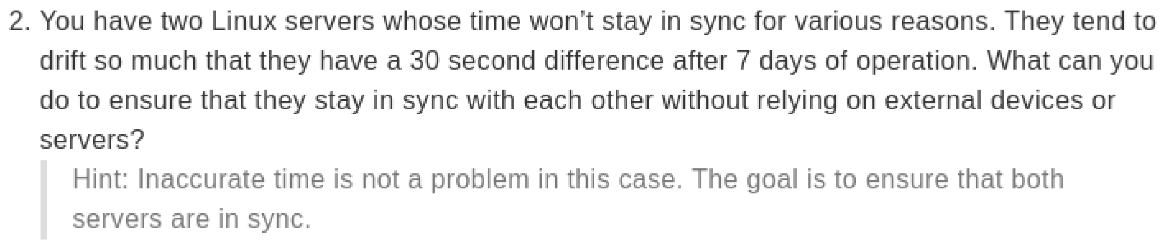
Name: Mosab Fathy Ramadan Mohamed
Group: B20-SD-01
Lab 11: System time and Package managers



1. What alternative do you have for configuring your NTP server pool if you don't want to be dependent on NTP servers on the internet. The time must be accurate and appear to be in sync with other devices globally. Describe how you will perform this setup.

   > The accuracy of the time should be strongly considered.

Answer:
Global Positioning System (GPS) relies on satellites instead of servers.
It is one of the global navigation satellite systems (GNSS) that provides geolocation and time information to a GPS receiver anywhere on or near the Earth where there is an unobstructed line of sight to four or more GPS satellites. It does not require the user to transmit any data, and operates independently of any telephonic or Internet reception.
GPS time is theoretically accurate to about 14 nanoseconds, due to the clock drift relative to International Atomic Time that the atomic clocks in GPS transmitters experience Most receivers lose some accuracy in their interpretation of the signals and are only accurate to about 100 nanoseconds
For more information about timekeeping in GPS please refer to this link



2. You have two Linux servers whose time won't stay in sync for various reasons. They tend to drift so much that they have a 30 second difference after 7 days of operation. What can you do to ensure that they stay in sync with each other without relying on external devices or servers?

   > Hint: Inaccurate time is not a problem in this case. The goal is to ensure that both servers are in sync.

Answer:
Since accuracy is not the problem and we only want them both to be in sync but at the same time we don't want to rely on any external stuff we can make one server rely on the other.
For this we can use the client-server NTP approach where one server will be the NTP server and the other will be the NTP client.
This will result in the second server(NTP client) syncing with the first server(NTP server) every two minutes

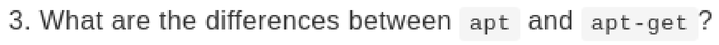
The configuration for the Server and Client could be done as explained in the lab:
Server:

```
$ apt install -y ntp
$ apt install -y ntpstat
$ vi /etc/ntp.conf
$ systemctl restart ntp
$ systemctl enable ntp
$ sudo ufw allow ntp
```

Client:

```
$ vi /etc/systemd/timesyncd.conf
add the following: NTP=<ntp-server-address>
$ systemctl restart systemd-timesyncd
```

Note: I skipped explaining the commands because the commands weren’t really required in the question and it was also explained in the lab itself, so i hope that’s fine.



3. What are the differences between `apt` and `apt-get`?

Answer:
To put it simply, apt is the command meant for the Linux user, and apt-get is the command meant for system use. In technical terms, this means that apt provides a high level interface for package management and apt-get provides a low level interface.
The two commands basically perform the same functions, but apt is easier to use and has some user-friendly features, like a status bar to show the progress as packages are installed. Since apt is designed for a Linux user, it doesn’t cause much fuss when developers update it or introduce their own implementations of it. Only the user is affected. On the other hand, apt-get receives fewer updates and needs to remain backward compatible. System functions and scripts rely on the predictability of apt-get.
apt consists of the most used features in apt-get and apt-cache and it can also manage apt.conf files.
So apt was made to have a better structured and intuitive usage to help the user, instead of jumping back and forth between different commands.

4. Why should System Administrators prefer `apt upgrade` over `apt full-upgrade`?

Answer:
apt upgrade is the command used to download and apply any available updates to your packages in a safe manner by not removing packages that are previously installed in a given Linux system, while “apt full-upgrade” command is used to do the same thing except if needed previously installed packages are removed to make the upgrade happen.

And, the removal of previously installed packages is unsafe because they might be used somewhere else for something which will result in that something failing due to the package removal.
So, apt full-upgrade might lead to some issues due to how unsafe it is.
That is why System Administrators should use apt upgrade instead of apt full-upgrade whenever they can.

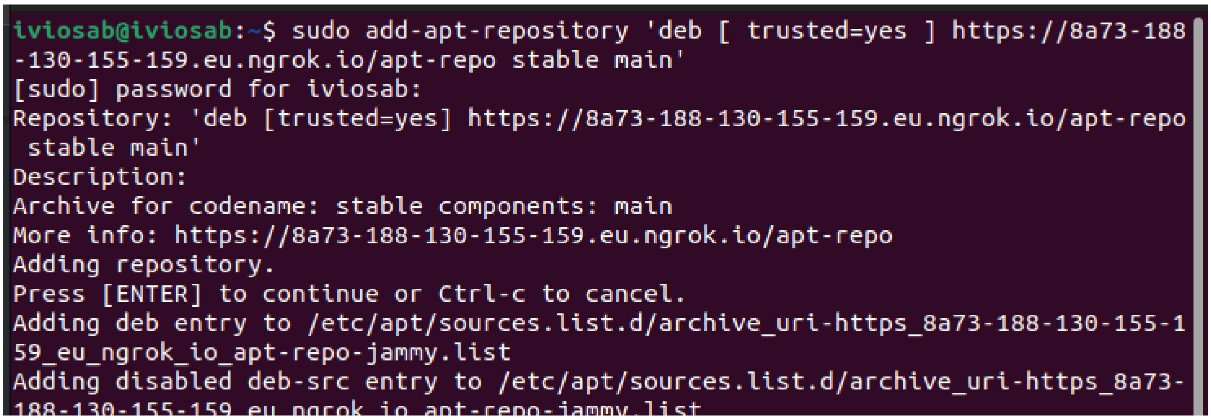
5. Show how you will install a package `lab11assign` from this repository https://8a73-188-130-155-159.eu.ngrok.io/apt-repo/.

   > You can use the following if you are on Innopolis network: http://10.90.137.100/apt-repo/

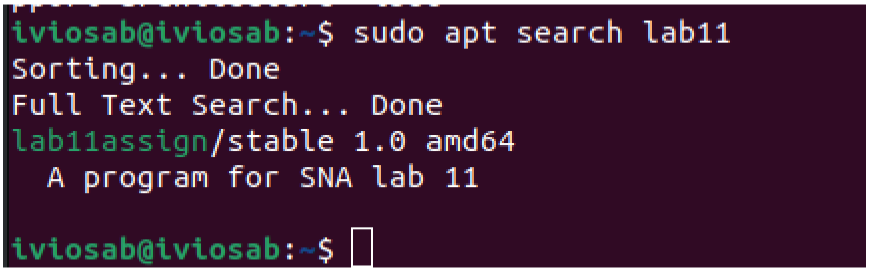
   - You need to add this repository to your package index.
   - The repository is not signed, so you need to add it as a trusted repository in the options.
     > Hint: [trusted=yes]
   - After adding the repository, show the output when you run `$ apt search lab11`

   > You are not allowed to manually download the debian package and install it.

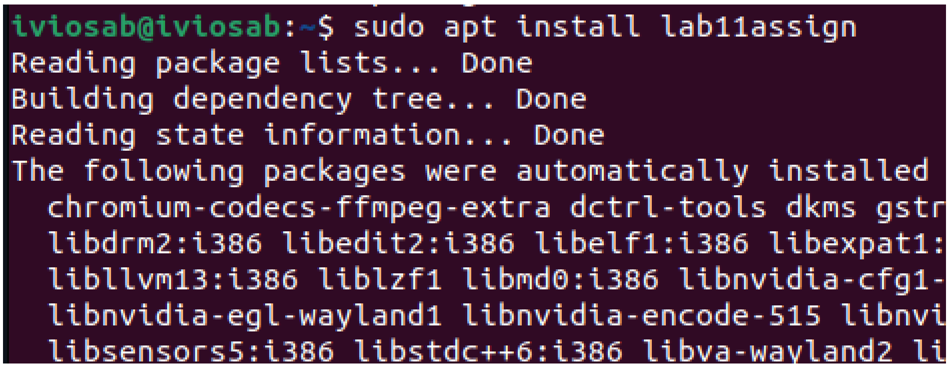
Answer:



```
iviosab@iviosab:~$ sudo add-apt-repository 'deb [ trusted=yes ] https://8a73-188
-130-155-159.eu.ngrok.io/apt-repo stable main'
[sudo] password for iviosab:
Repository: 'deb [trusted=yes] https://8a73-188-130-155-159.eu.ngrok.io/apt-repo
 stable main'
Description:
Archive for codename: stable components: main
More info: https://8a73-188-130-155-159.eu.ngrok.io/apt-repo
Adding repository.
Press [ENTER] to continue or Ctrl-c to cancel.
Adding deb entry to /etc/apt/sources.list.d/archive_uri-https_8a73-188-130-155-1
59_eu_ngrok_io_apt-repo-jammy.list
Adding disabled deb-src entry to /etc/apt/sources.list.d/archive_uri-https_8a73-
188-130-155-159_eu_ngrok_io_apt-repo-jammy.list
```



```
iviosab@iviosab:~$ sudo apt search lab11
Sorting... Done
Full Text Search... Done
lab11assign/stable 1.0 amd64
  A program for SNA lab 11

iviosab@iviosab:~$ 
```



```
iviosab@iviosab:~$ sudo apt install lab11assign
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed
  chromium-codecs-ffmpeg-extra dctrl-tools dkms gstr
  libdrm2:i386 libedit2:i386 libelf1:i386 libexpat1:
  libllvm13:i386 liblzf1 libmd0:i386 libnvidia-cfg1-
  libnvidia-egl-wayland1 libnvidia-encode-515 libnvi
  libsensors5:i386 libstdc++6:i386 libva-wayland2 li
```

6. Create an Ubuntu package that meets the following requirements:
    - The package creates the directory `/var/helloworld/` on the target system.
    - The package contains the python script `/var/helloworld/helloworld.py`. The python script is simple:

      ```
      #!/usr/bin/env python3
      print("Hello, World!")
      ```

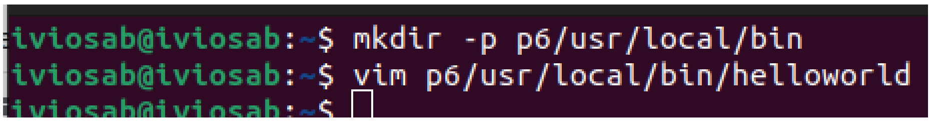
    - The package should deploy a bash script `helloworld` that executes `/var/helloworld/hello.py` on the target system.

Take the following steps after building the package

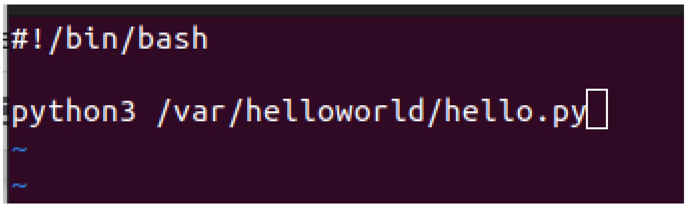
- List the content of the package with the command `$ dpkg -c <package-name>.deb`.
- Install the package and show all artifacts added to your system by the package.

> After a user installs your package, he should be able to run `$ helloworld` from the terminal without additinonal steps.
> The expected flow of execution is `helloworld (bash script) -> helloworld.py -> Output (Hello, World!)`

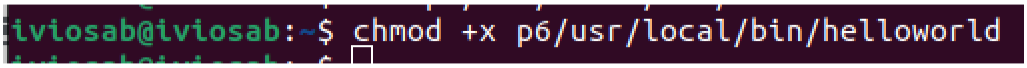
Answer:



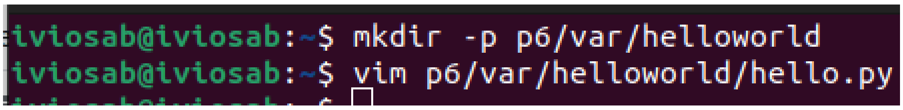
```
iviosab@iviosab:~$ mkdir -p p6/usr/local/bin
iviosab@iviosab:~$ vim p6/usr/local/bin/helloworld
iviosab@iviosab:~$ 
```



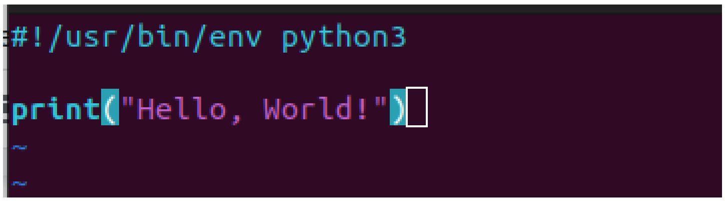
```
#!/bin/bash

python3 /var/helloworld/hello.py
~
~
```



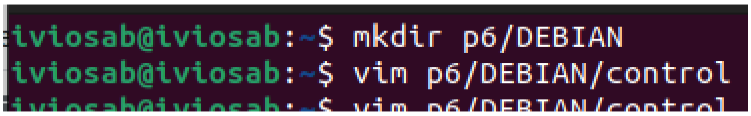
```
iviosab@iviosab:~$ chmod +x p6/usr/local/bin/helloworld
```



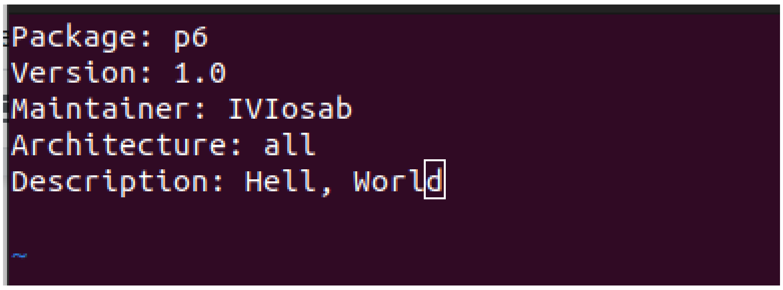
```
iviosab@iviosab:~$ mkdir -p p6/var/helloworld
iviosab@iviosab:~$ vim p6/var/helloworld/hello.py
```



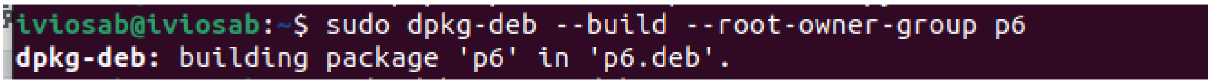
```
#!/usr/bin/env python3

print("Hello, World!")
~
~
```



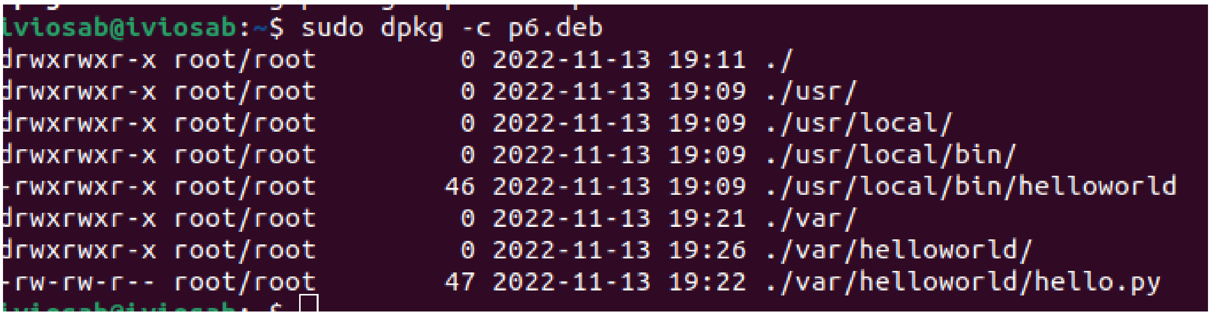
```
iviosab@iviosab:~$ mkdir p6/DEBIAN
iviosab@iviosab:~$ vim p6/DEBIAN/control
iviosab@iviosab:~$ vim p6/DEBIAN/control
```



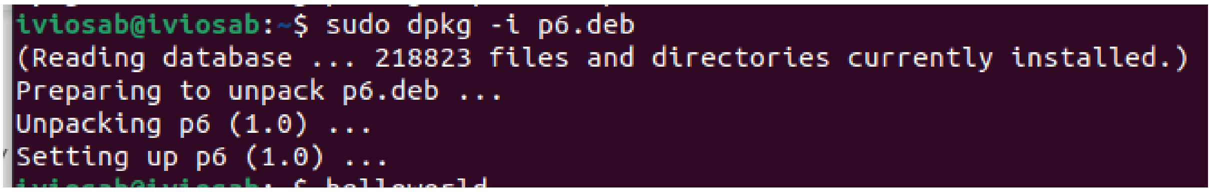
```
Package: p6
Version: 1.0
Maintainer: IVIosab
Architecture: all
Description: Hell, World

~
```



```
iviosab@iviosab:~$ sudo dpkg-deb --build --root-owner-group p6
dpkg-deb: building package 'p6' in 'p6.deb'.
```



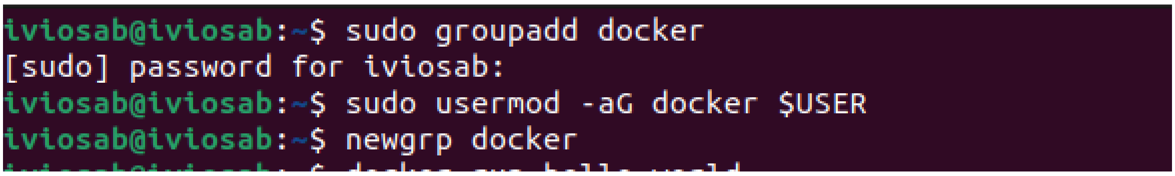
```
iviosab@iviosab:~$ sudo dpkg -c p6.deb
drwxrwxr-x root/root         0 2022-11-13 19:11 ./
drwxrwxr-x root/root         0 2022-11-13 19:09 ./usr/
drwxrwxr-x root/root         0 2022-11-13 19:09 ./usr/local/
drwxrwxr-x root/root         0 2022-11-13 19:09 ./usr/local/bin/
-rwxrwxr-x root/root        46 2022-11-13 19:09 ./usr/local/bin/helloworld
drwxrwxr-x root/root         0 2022-11-13 19:21 ./var/
drwxrwxr-x root/root         0 2022-11-13 19:26 ./var/helloworld/
-rw-rw-r-- root/root        47 2022-11-13 19:22 ./var/helloworld/hello.py
iviosab@iviosab:~$
```



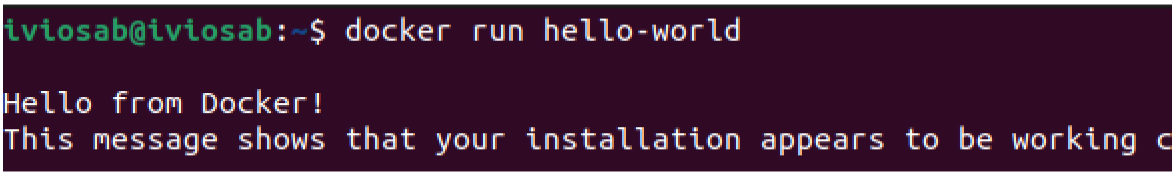
```
iviosab@iviosab:~$ sudo dpkg -i p6.deb
(Reading database ... 218823 files and directories currently installed.)
Preparing to unpack p6.deb ...
Unpacking p6 (1.0) ...
Setting up p6 (1.0) ...
iviosab@iviosab:~$ helloworld
```

7. Find and add new source repository to be used for yum.
    - Install a package from it (for example MongoDB).
    - Check with the RPM package manager to verify that the package was installed, and provide details such as dependencies needed.
    - Find logs related to all actions from the previous steps.

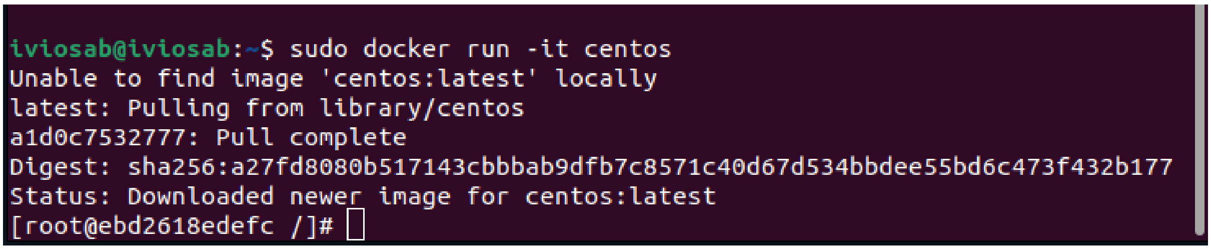
Answer:



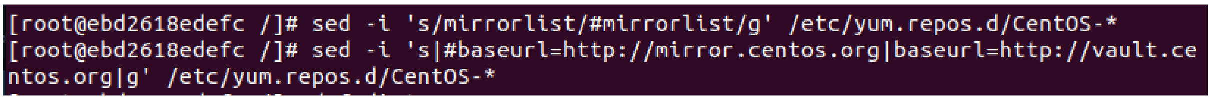
```
iviosab@iviosab:~$ sudo groupadd docker
[sudo] password for iviosab:
iviosab@iviosab:~$ sudo usermod -aG docker $USER
iviosab@iviosab:~$ newgrp docker
iviosab@iviosab:~$ docker run hello-world
```



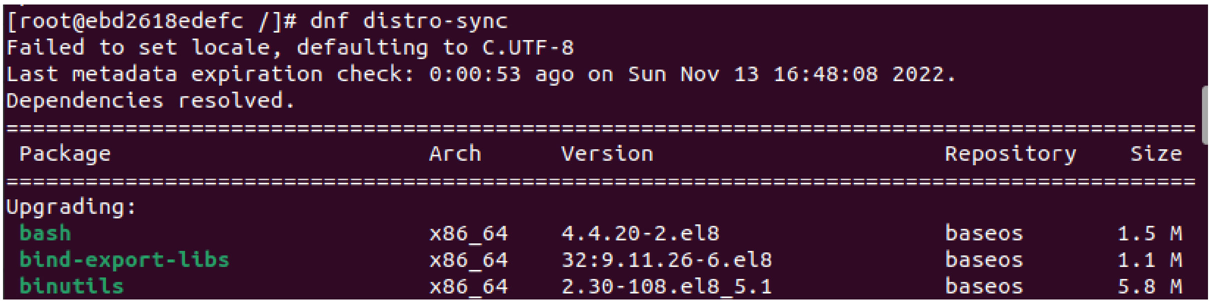
```
iviosab@iviosab:~$ docker run hello-world

Hello from Docker!
This message shows that your installation appears to be working c
```



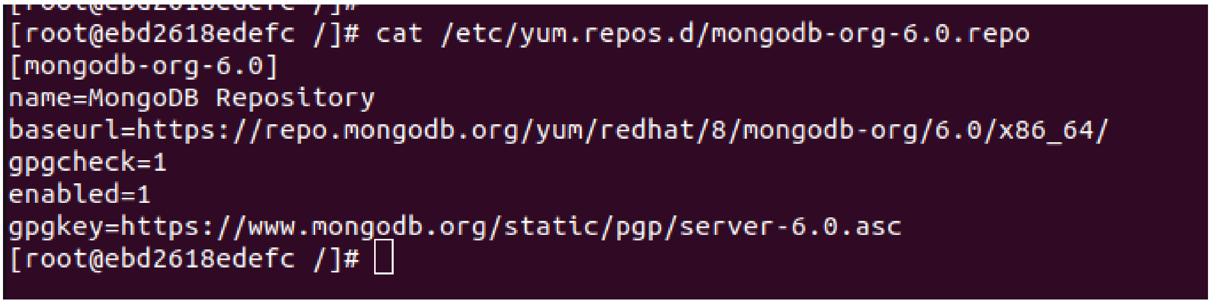
```
iviosab@iviosab:~$ sudo docker run -it centos
Unable to find image 'centos:latest' locally
latest: Pulling from library/centos
a1d0c7532777: Pull complete
Digest: sha256:a27fd8080b517143cbbbab9dfb7c8571c40d67d534bbdee55bd6c473f432b177
Status: Downloaded newer image for centos:latest
[root@ebd2618edefc /]# 
```



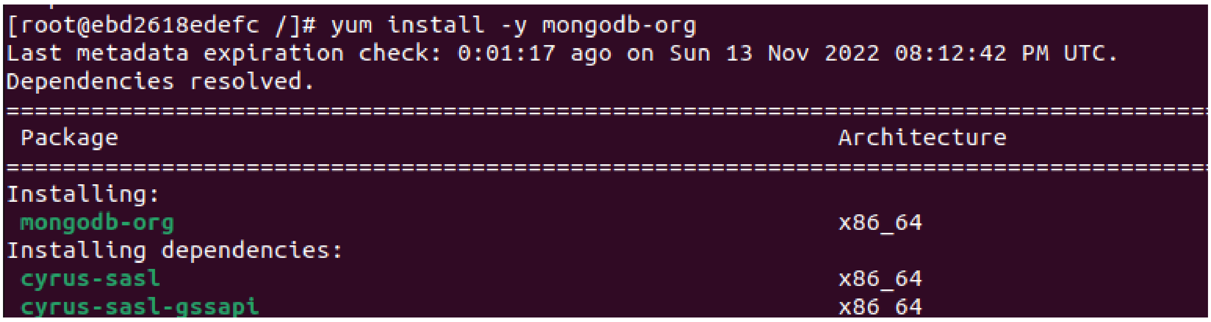
```
[root@ebd2618edefc /]# sed -i 's/mirrorlist/#mirrorlist/g' /etc/yum.repos.d/CentOS-*
[root@ebd2618edefc /]# sed -i 's|#baseurl=http://mirror.centos.org|baseurl=http://vault.ce
ntos.org|g' /etc/yum.repos.d/CentOS-*
```



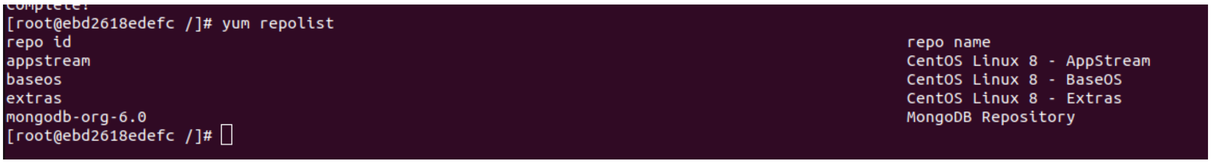
```
[root@ebd2618edefc /]# dnf distro-sync
Failed to set locale, defaulting to C.UTF-8
Last metadata expiration check: 0:00:53 ago on Sun Nov 13 16:48:08 2022.
Dependencies resolved.
==========================================================================================
 Package                         Arch      Version                      Repository    Size
==========================================================================================
Upgrading:
 bash                            x86_64    4.4.20-2.el8                 baseos       1.5 M
 bind-export-libs                x86_64    32:9.11.26-6.el8             baseos       1.1 M
 binutils                        x86_64    2.30-108.el8_5.1             baseos       5.8 M
```



```
[root@ebd2618edefc /]# cat /etc/yum.repos.d/mongodb-org-6.0.repo
[mongodb-org-6.0]
name=MongoDB Repository
baseurl=https://repo.mongodb.org/yum/redhat/8/mongodb-org/6.0/x86_64/
gpgcheck=1
enabled=1
gpgkey=https://www.mongodb.org/static/pgp/server-6.0.asc
[root@ebd2618edefc /]# 
```



```
[root@ebd2618edefc /]# yum install -y mongodb-org
Last metadata expiration check: 0:01:17 ago on Sun 13 Nov 2022 08:12:42 PM UTC.
Dependencies resolved.
============================================================================
 Package                                                Architecture
============================================================================
Installing:
 mongodb-org                                            x86_64
Installing dependencies:
 cyrus-sasl                                             x86_64
 cyrus-sasl-gssapi                                      x86_64
```



```
[root@ebd2618edefc /]# yum repolist
repo id                                                                  repo name
appstream                                                                CentOS Linux 8 - AppStream
baseos                                                                   CentOS Linux 8 - BaseOS
extras                                                                   CentOS Linux 8 - Extras
mongodb-org-6.0                                                          MongoDB Repository
[root@ebd2618edefc /]# 
```