Curso introducción y manejo de nf-core/ampliseq

Manual de instalación de nextflow y docker

Julio 2022

Versiones utilizadas en este manual:

UBUNTU

- Ubuntu 20.04.1

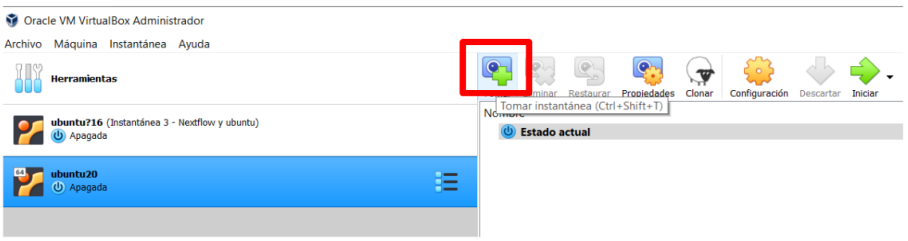
Elaborado por:

Versión 0.1. Julio 2022
José Eduardo García López

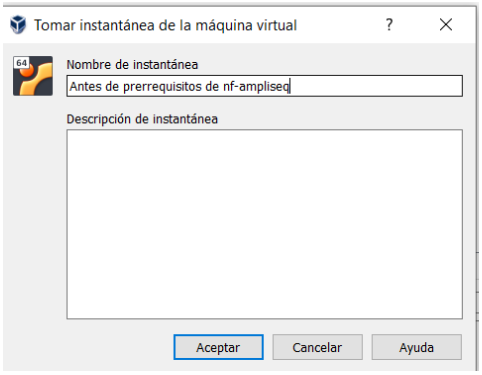
INSTANTÁNEA ANTES DE LAS INSTALACIONES

Antes de empezar, es importante respaldar el estado actual de nuestra máquina virtual en caso de que queramos regresar a como estaba antes de la instalación de los prerrequisitos de nf-core/ampliseq.

1. Abre tu VirtualBox de Ubuntu 20.04
2. En las opciones de VirtualBox da clic en tomar instantánea



3. Guarda la instantánea como “Antes de prerrequisitos denf-core/ampliseq”



INSTALACIÓN DE DOCKER

1. Inicia tu máquina virtual y abre una terminal
2. En la terminal escribe `sudo apt-get update` , pon la contraseña de maquina virtual y espera a que se actualicen los paquetes.

```
edu@edu-VirtualBox:~$ sudo apt-get update
[sudo] password for edu:
Get:1 http://security.ubuntu.com/ubuntu focal-security InRelease [114 kB]
Hit:2 http://mx.archive.ubuntu.com/ubuntu focal InRelease
Get:3 http://mx.archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]
Get:4 http://security.ubuntu.com/ubuntu focal-security/main amd64 DEP-11 Metadata [40.7 kB]
Get:5 http://mx.archive.ubuntu.com/ubuntu focal-backports InRelease [108 kB]
Get:6 http://security.ubuntu.com/ubuntu focal-security/universe amd64 DEP-11 Metadata [66.9 kB]
Get:7 http://security.ubuntu.com/ubuntu focal-security/multiverse amd64 DEP-11 Metadata [2 464 B]
Get:8 http://mx.archive.ubuntu.com/ubuntu focal-updates/main amd64 DEP-11 Metadata [278 kB]
Get:9 http://mx.archive.ubuntu.com/ubuntu focal-updates/universe amd64 DEP-11 Metadata [390 kB]
Get:10 http://mx.archive.ubuntu.com/ubuntu focal-updates/multiverse amd64 DEP-11 Metadata [944 B]
Get:11 http://mx.archive.ubuntu.com/ubuntu focal-backports/main amd64 DEP-11 Metadata [7 988 B]
Get:12 http://mx.archive.ubuntu.com/ubuntu focal-backports/universe amd64 DEP-11 Metadata [30.5 kB]
Fetched 1 154 kB in 3s (447 kB/s)
Reading package lists... Done
edu@edu-VirtualBox:~$
```

3. Ahora vamos a instalar algunos comandos y paquetes necesarios para docker, copia y pega el siguiente comando:

   `sudo apt-get install ca-certificates curl gnupg lsb-release`

   Una vez que le des enter al comando, Ubuntu te preguntara si quieres instalar los paquetes, escribe *“y”* y da enter


```
edu@edu-VirtualBox:~$ sudo apt-get install ca-certificates curl gnupg lsb-release
Reading package lists... Done
Building dependency tree
Reading state information... Done
lsb-release is already the newest version (11.1.0ubuntu2).
lsb-release set to manually installed.
ca-certificates is already the newest version (20211016~20.04.1).
ca-certificates set to manually installed.
gnupg is already the newest version (2.2.19-3ubuntu2.2).
gnupg set to manually installed.
The following package was automatically installed and is no longer required:
  libfwupdplugin1
Use 'sudo apt autoremove' to remove it.
The following additional packages will be installed:
  libcurl4
The following NEW packages will be installed:
  curl libcurl4
0 upgraded, 2 newly installed, 0 to remove and 0 not upgraded.
Need to get 396 kB of archives.
After this operation, 1 121 kB of additional disk space will be used.
Do you want to continue? [Y/n]
```

4. Ahora vamos a agregar las llaves a los repositorios de docker, crea una carpeta con el siguiente comando:

   `sudo mkdir -p /etc/apt/keyrings`

   Posteriormente, copia y pega el próximo comando para agregar la llave GPG de docker :

   ```
   curl -fsSL https://download.docker.com/linux/ubuntu/gpg \
   | sudo gpg --dearmor -o /etc/apt/keyrings/docker.gpg
   ```

   O en una sola línea:

   ```
   curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /etc/apt/keyrings/docker.gpg
   ```

```
edu@edu-VirtualBox:~$  sudo mkdir -p /etc/apt/keyrings
edu@edu-VirtualBox:~$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /etc/apt/keyrings/docker.gpg
edu@edu-VirtualBox:~$
```

5. Usa el siguiente comando para agregar el repositorio de docker:

   ```
   echo "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.gpg] https://download.docker.com/linux/ubuntu \
     $(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
   ```

```
edu@edu-VirtualBox:~$ echo \
>   "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.gpg] https://download.docker.com/linux/ubuntu \
>   $(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
edu@edu-VirtualBox:~$
```

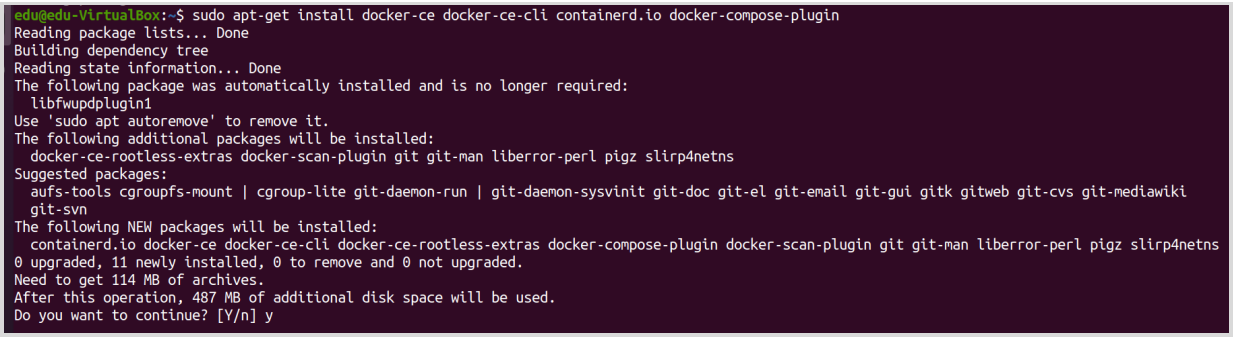
6. Antes de instalar Docker vuelve a actualizar los paquetes:

   `sudo apt-get update`

```
edu@edu-VirtualBox:~$ sudo apt-get update
Get:1 http://security.ubuntu.com/ubuntu focal-security InRelease [114 kB]
Get:2 https://download.docker.com/linux/ubuntu focal InRelease [57.7 kB]
Hit:3 http://mx.archive.ubuntu.com/ubuntu focal InRelease
Get:4 https://download.docker.com/linux/ubuntu focal/stable amd64 Packages [17.6 kB]
Get:5 http://mx.archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]
Get:6 http://mx.archive.ubuntu.com/ubuntu focal-backports InRelease [108 kB]
Fetched 411 kB in 2s (265 kB/s)
Reading package lists... Done
edu@edu-VirtualBox:~$
```

7. Instala docker con el siguiente comando, cuando te pregunte si quieres instalar nuevos paquetes escribe “**y**” y pulsa enter:

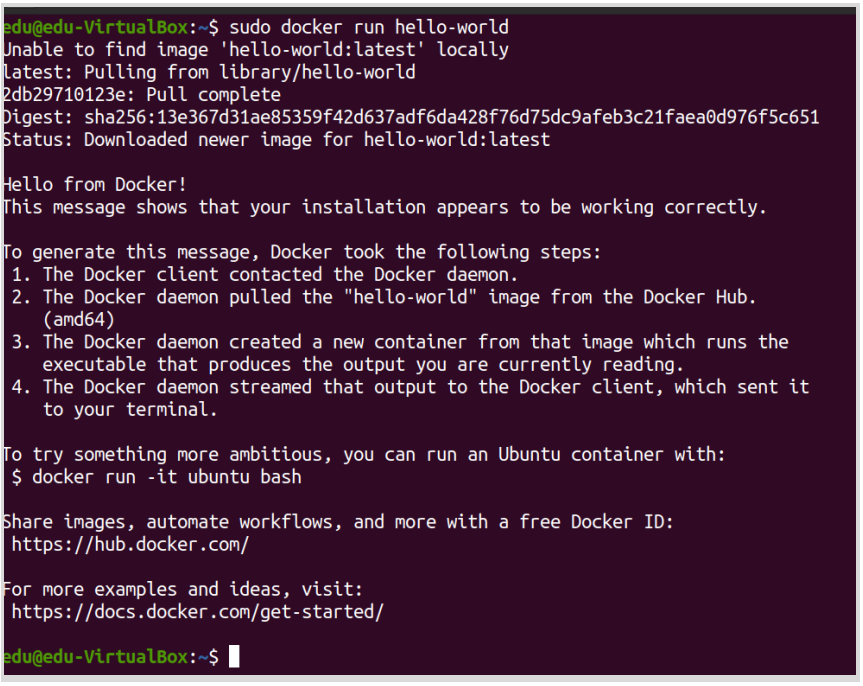
```
sudo apt-get install docker-ce docker-ce-cli containerd.io docker-compose-plugin
```



Comprueba la instalación de Docker:

```
sudo docker run hello-world
```

Si la instalación fue correcta obtendrás el siguiente mensaje:



Ahora vamos a hacer que docker se pueda correr sin la necesidad de *sudo*:

7. Agrega tu **usuario** al grupo Docker:

```
sudo usermod -aG  docker TuUsuario
```

```
edu@edu-VirtualBox:~$ sudo usermod -aG docker edu
```

8. Actualiza los grupos con el siguiente comando:

`newgrp docker`

```
edu@edu-VirtualBox:~$  newgrp docker
edu@edu-VirtualBox:~$
```

9. Confirma que puedes correr Docker sin necesidad de **sudo**:

`docker run hello-world`

```
edu@edu-VirtualBox:~$  docker run hello-world

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
 $ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
 https://hub.docker.com/

For more examples and ideas, visit:
 https://docs.docker.com/get-started/
```

Puedes obtener más información en el manual de Docker

INSTALACIÓN DE NEXTFLOW

1. Instala Java 11 con el siguiente comando:

   `sudo apt install openjdk-11-jre-headless`

   Cuando ubuntu te pregunte si quieres instalar nuevos paquetes, escribe "**y**" y da enter:

```
edu@edu-VirtualBox:~$ sudo apt install openjdk-11-jre-headless
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following package was automatically installed and is no longer required:
  libfwupdplugin1
Use 'sudo apt autoremove' to remove it.
The following additional packages will be installed:
  ca-certificates-java java-common
Suggested packages:
  default-jre fonts-dejavu-extra fonts-ipafont-gothic fonts-ipafont-mincho fonts-wqy-microhei
  | fonts-wqy-zenhei
The following NEW packages will be installed:
  ca-certificates-java java-common openjdk-11-jre-headless
0 upgraded, 3 newly installed, 0 to remove and 0 not upgraded.
Need to get 37.4 MB of archives.
After this operation, 171 MB of additional disk space will be used.
Do you want to continue? [Y/n]
```

   La instalación tardará unos minutos, cuando termine verás el siguiente mensaje:

```
Updating certificates in /etc/ssl/certs...
0 added, 0 removed; done.
Running hooks in /etc/ca-certificates/update.d...

done.
done.
edu@edu-VirtualBox:~$
```

2. Ahora en tu *home* vamos a crear una carpeta llamada "bin" donde vamos a instalar el archivo ejecutable de nextflow.

   Escribimos el comando **`cd`** en la terminal para dirigirnos al *home,* una vez ahi crearemos una carpeta llamada "bin" con el siguiente comando:

   `mkdir bin`

```
edu@edu-VirtualBox:~$ cd
edu@edu-VirtualBox:~$ mkdir bin
edu@edu-VirtualBox:~$ ls
bin  Desktop  Documents  Downloads  Music  Pictures  Public  Templates  Videos
edu@edu-VirtualBox:~$
```

   Una vez creada la carpeta, usa el comando cd para ingresar a la carpeta "**bin**":

   `cd bin`

   Dentro de la carpeta vamos a descargar el ejecutable de nextflow:

   `curl -s https://get.nextflow.io | bash`

Después de descargar nextflow verás el siguiente mensaje:

```
      N E X T F L O W
      version 22.04.4 build 5706
      created 19-06-2022 20:51 UTC (15:51 CDT)
      cite doi:10.1038/nbt.3820
      http://nextflow.io


Nextflow installation completed. Please note:
- the executable file `nextflow` has been created in the folder: /home/edu/bin
- you may complete the installation by moving it to a directory in your $PATH
```

Pese a que ya instalamos Nextflow, aún no podemos usar el comando afuera de la carpeta bin, donde está el ejecutable. Tenemos que crear un enlace simbólico en otra carpeta para poder ejecutar Nextflow desde cualquier lugar.

3. Con el comando cd nos vamos a dirigir a la carpeta /bin/ donde se encuentran los comandos de Ubuntu:

`cd /bin/`

Dentro de la carpeta /bin/, hay que crear un enlace simbólico que nos dirija al ejecutable de Nextflow, usa el siguiente comando. Donde **usuario** es el nombre de que de usuario que escogiste en tu máquina virtual:

`sudo ln -s /home/usuario/bin/nextflow nextflow`

```
edu@edu-VirtualBox:~$ cd /bin/
edu@edu-VirtualBox:/bin$ sudo ln -s /home/edu/bin/nextflow nextflow
[sudo] password for edu:
edu@edu-VirtualBox:/bin$
```

9. Dirigete con el comando **cd** a tu home y confirma que puedes usar el comando **nextflow** desde cualquier carpeta:

```
edu@edu-VirtualBox:/bin$ cd
edu@edu-VirtualBox:~$ nextflow
Usage: nextflow [options] COMMAND [arg...]

Options:
  -C
     Use the specified configuration file(s) overriding any defaults
  -D
     Set JVM properties
  -bg
     Execute nextflow in background
  -c, -config
     Add the specified file to configuration set
  -d, -dockerize
     Launch nextflow via Docker (experimental)
  -h
     Print this help
```

10. Confirma que nextflow funciona con el siguiente comando:

`nextflow run hello`

Si la instalación fue exitosa, obtendrás este mensaje:


```
edu@edu-VirtualBox:~$ nextflow run hello
N E X T F L O W  ~  version 22.04.4
Pulling nextflow-io/hello ...
 downloaded from https://github.com/nextflow-io/hello.git
Launching `https://github.com/nextflow-io/hello` [magical_archimedes] DSL2 - revision: 4eab81bd42 [m
ster]
executor >  local (4)
[d2/ac693f] process > sayHello (3) [100%] 4 of 4 ✔
Ciao world!

Hola world!

Bonjour world!

Hello world!
```

Puedes obtener más información en el manual de Nextflow

! Felicidades, has instalado Nextflow y Docker correctamente¡

INSTALACIÓN DE GIT

---

1. Desde el escritorio abre una nueva terminal, dando clic derecho y eligiendo la opción “Open in terminal”.
2. En la terminal escribe el comando para instalar git:

   `sudo apt install git`

```
edu@edu-VirtualBox:~/Desktop$ sudo apt install git
[sudo] password for edu:
Reading package lists... Done
Building dependency tree
Reading state information... Done
git is already the newest version (1:2.25.1-1ubuntu3.4).
git set to manually installed.
The following package was automatically installed and is no longer required:
  libfwupdplugin1
Use 'sudo apt autoremove' to remove it.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
```

3. Una vez instalado, escribe **`git`** en la terminal para confirmar que el comando funciona. Obtendrás un resultado como el siguiente:

```
edu@edu-VirtualBox:~/Desktop$ git
usage: git [--version] [--help] [-C <path>] [-c <name>=<value>]
           [--exec-path[=<path>]] [--html-path] [--man-path] [--info-path]
           [-p | --paginate | -P | --no-pager] [--no-replace-objects] [--bare]
           [--git-dir=<path>] [--work-tree=<path>] [--namespace=<name>]
           <command> [<args>]

These are common Git commands used in various situations:

start a working area (see also: git help tutorial)
   clone             Clone a repository into a new directory
   init              Create an empty Git repository or reinitialize an existing one
```

1. El pipeline de nf-core/ampliseq, es distribuido por la comunidad de nf-core, por lo tanto, se puede instalar y empezar a correr con un solo comando. No obstante, para probar el pipeline con datos y recursos mínimos vamos a usar un script construido por Biofreelancer.

   Abre una terminal y dirígete al home con el comando `cd` , en el home descarga el dataset de prueba con el siguiente comando:

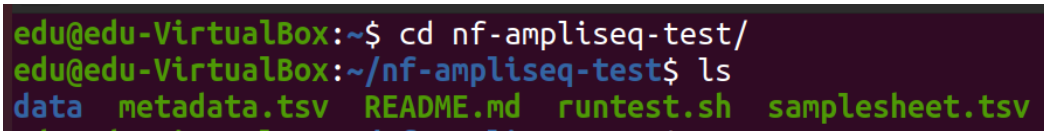
   ```
   git clone https://github.com/biofreelancer/nf-ampliseq-test.git
   ```

   Una vez descargado, con el comando `ls` podrás ver que se creó la carpeta "***nf-ampliseq-test".***

2. Entra a la carpeta "***nf-ampliseq-test" :***

   ```
   cd nf-ampliseq-test
   ```

3. Dentro de la carpeta podrás encontrar todo lo necesario para correr nf-cpre con datos de prueba:



4. Corre el test de nf-core/ampliseq con el siguiente comando:

   ```
   bash runtest.sh
   ```

   Al correr el script empezará la ejecución de nf-core/ampliseq


```
edu@edu-VirtualBox:~/nf-ampliseq-test$ bash runtest.sh
N E X T F L O W  ~  version 22.04.4
WARN: It appears you have never run this project before -- Option `-resume` is ignored
Launching `https://github.com/nf-core/ampliseq` [friendly_bohr] DSL2 - revision: aed35e5b4b [master]

------------------------------------------------------
                                          ,--./,-.
                                         /,-._.--~'
  |\ | |__  __ /  ` /  \ |__) |__         } {
  | \| |       \__, \__/ |  \ |___     \`-._,-`-,
                                          `._,._,'
  nf-core/ampliseq v2.3.2
------------------------------------------------------
Core Nextflow options
  revision                : master
  runName                 : friendly_bohr
  containerEngine         : docker
  launchDir               : /home/edu/nf-ampliseq-test
  workDir                 : /home/edu/nf-ampliseq-test/work
  projectDir              : /home/edu/.nextflow/assets/nf-core/ampliseq
  userName                : edu
  profile                 : docker
  configFiles             : /home/edu/.nextflow/assets/nf-core/ampliseq/nextflow.config
```

Conforme se completan los diferentes módulos del pipeline, verás como cambian a 100% y se marcan con una palomilla.

```
[fa/50d511] process > NFCORE_AMPLISEQ:AMPLISEQ:BARRNAP (ASV_seqs.fasta)                       [100%] 1 of 1 ✔
[b3/bdc1f0] process > NFCORE_AMPLISEQ:AMPLISEQ:FORMAT_TAXONOMY                                [100%] 1 of 1 ✔
[0d/5b71d4] process > NFCORE_AMPLISEQ:AMPLISEQ:DADA2_TAXONOMY (ASV_seqs.fasta,assignTaxonom... [100%] 1 of 1 ✔
[d3/ae0eff] process > NFCORE_AMPLISEQ:AMPLISEQ:DADA2_ADDSPECIES (ASV_tax.rds,addSpecies.fna... [  0%] 0 of 1
[5f/020bf0] process > NFCORE_AMPLISEQ:AMPLISEQ:QIIME2_TAXONOMY:QIIME2_INSEQ (ASV_seqs.fasta)  [100%] 1 of 1 ✔
[-        ] process > NFCORE_AMPLISEQ:AMPLISEQ:QIIME2_TAXONOMY:QIIME2_CLASSIFY               -
[2c/d38371] process > NFCORE_AMPLISEQ:AMPLISEQ:QIIME2_INASV (ASV_table.tsv)                   [100%] 1 of 1 ✔
[01/2264c9] process > NFCORE_AMPLISEQ:AMPLISEQ:QIIME2_INSEQ (ASV_seqs.fasta)                  [100%] 1 of 1 ✔
[-        ] process > NFCORE_AMPLISEQ:AMPLISEQ:QIIME2_INTAX                                  -
[-        ] process > NFCORE_AMPLISEQ:AMPLISEQ:QIIME2_FILTERTAXA                             -
[-        ] process > NFCORE_AMPLISEQ:AMPLISEQ:FILTER_STATS                                  -
[-        ] process > NFCORE_AMPLISEQ:AMPLISEQ:MERGE_STATS_FILTERTAXA                        -
[-        ] process > NFCORE_AMPLISEQ:AMPLISEQ:QIIME2_EXPORT:QIIME2_EXPORT_ABSOLUTE          -
[-        ] process > NFCORE_AMPLISEQ:AMPLISEQ:QIIME2_EXPORT:QIIME2_EXPORT_RELASV            -
[-        ] process > NFCORE_AMPLISEQ:AMPLISEQ:QIIME2_EXPORT:QIIME2_EXPORT_RELTAX            -
[-        ] process > NFCORE_AMPLISEQ:AMPLISEQ:QIIME2_EXPORT:COMBINE_TABLE_QIIME2            -
[-        ] process > NFCORE_AMPLISEQ:AMPLISEQ:QIIME2_EXPORT:COMBINE_TABLE_DADA2             -
[-        ] process > NFCORE_AMPLISEQ:AMPLISEQ:QIIME2_BARPLOT                                -
[11/0915bd] process > NFCORE_AMPLISEQ:AMPLISEQ:METADATA_ALL (metadata.tsv)                    [100%] 1 of 1 ✔
[0c/564b0d] process > NFCORE_AMPLISEQ:AMPLISEQ:METADATA_PAIRWISE (metadata.tsv)               [100%] 1 of 1 ✔
[-        ] process > NFCORE_AMPLISEQ:AMPLISEQ:QIIME2_DIVERSITY:QIIME2_TREE                  -
[-        ] process > NFCORE_AMPLISEQ:AMPLISEQ:QIIME2_DIVERSITY:QIIME2_DIVERSITY_CORE       -
[-        ] process > NFCORE_AMPLISEQ:AMPLISEQ:QIIME2_DIVERSITY:QIIME2_DIVERSITY_ALPHA      -
[-        ] process > NFCORE_AMPLISEQ:AMPLISEQ:QIIME2_DIVERSITY:QIIME2_DIVERSITY_BETA       -
[-        ] process > NFCORE_AMPLISEQ:AMPLISEQ:QIIME2_DIVERSITY:QIIME2_DIVERSITY_BETAORD    -
[-        ] process > NFCORE_AMPLISEQ:AMPLISEQ:CUSTOM_DUMPSOFTWAREVERSIONS                  -
[-        ] process > NFCORE_AMPLISEQ:AMPLISEQ:MULTIQC                                      -
```

Después de aprox. 10 minutos obtendrás el siguiente mensaje:


```
[b4/d33beb] process > NFCORE_AMPLISEQ:AMPLISEQ:MULTIQC
-[nf-core/ampliseq] Pipeline completed successfully-

Completed at: 11-Jul-2022 17:14:37
Duration    : 11m 29s
CPU hours   : 0.3
Succeeded   : 51


======

              nf-ampliseq TEST SUCCESSFUL

======
edu@edu-VirtualBox:~/nf-ampliseq-test$
```

! Felicidades, has instalado nf-core/ampliseq correctamente¡