

Administracio de Sistemes Gestors de Bases de Dades - Sebastian Ciscar

None

None

Table of contents

1. 1 Amazon Web Services (AWS)	3
1.1 1.1 Què és AWS Academy?	3
1.2 1.2 Com accedisc a AWS Academy?	3
1.3 1.3 Què és AWS Academy Learner Lab?	3
1.4 1.4 Com puc utilitzar AWS Academy Learner Lab?	3
1.5 1.5 Serveis d'Amazon Web Services (AWS)	3
1.6 1.6 Com crear instàncies EC2?	4
1.7 1.7 Com connectar-nos a una instància EC2 per SSH des de Linux	16
1.8 1.7.3 Pas 3. Descarreguem la clau privada .PEM des del Learner Lab	18
1.9 1.7.4 Pas 4. Connectar-nos per SSH des de Linux	19
2. Instalación MySql Server	20
2.1 Instalación utilizando binarios	20
2.2 Crear un usuario y un grupo de mysql	21
2.3 Obtenga y desempaquete la aplicación	21
2.4 Instalación utilizando el gestor de paquetes APT de linux.	21
2.5 Actualizar la versión de MySql Server	22
2.6 Paquetes disponibles del repositorio APT de Mysql	24
2.7 Configuración y pruebas posteriores a la instalación	25
2.8 Actualización de mysql	28
2.9 Uso de imagen Docker para MySql	30
3. Instalación MongoDB	32
3.1 Introducción	32
3.2 Instalación utilizando binarios	32
3.3 Instalación utilizando el administrador de paquetes	32
3.4 Iniciar el servicio de MongoDB	32
3.5 Detener el servicio de MongoDB	33
3.6 Conexión a MongoDB	33
3.7 Crear usuario administrador	33
4. Imagen de MongoDB en docker	33
4.1 Persistencia de datos	33
4.2 Conexión a MongoDB en Docker	33
5. Probar la conexión a MongoDB	34
6. Asegurar MongoDB	34

1. 1 Amazon Web Services (AWS)

Amazon Web Services (AWS) és una col·lecció de serveis de computació en el núvol pública que, en conjunt, formen una plataforma de computació en el núvol, oferida per Amazon a través d'Internet.

1.1 1.1 Què és AWS Academy?

AWS Academy és un programa d'AWS que ofereix gratuïtament a les institucions d'educació superior un pla d'estudis de computació en el núvol, preparant els estudiants per obtenir certificacions reconegudes en la indústria.

Referències: Preguntes freqüents sobre AWS Academy.

1.2 1.2 Com accedisc a AWS Academy?

L'URL per accedir a AWS Academy és la següent:

https://www.awsacademy.com/LMS_Login

1.3 1.3 Què és AWS Academy Learner Lab?

AWS Academy Learner Lab és un laboratori que permet als estudiants treballar amb alguns dels serveis d'AWS durant un període de temps. Cada estudiant disposa d'un crèdit de 100 dòlars per utilitzar en serveis de la plataforma AWS.

1.4 1.4 Com puc utilitzar AWS Academy Learner Lab?

En aquesta guia trobaràs tots els passos necessaris per accedir i utilitzar la plataforma AWS Academy Learner Lab.

Guia de l'estudiant d'AWS Academy Learner Lab. (pdf)

1.5 1.5 Serveis d'Amazon Web Services (AWS)

A continuació es mostren alguns dels serveis d'AWS que utilitzarem durant el curs.

- **Amazon EC2 (Elastic Compute Cloud):** Servei web que proporciona capacitat de càlcul escalable en el núvol.
- **Amazon S3 (Simple Storage Service):** Servei d'emmagatzematge d'objectes.
- **Amazon EBS (Elastic Block Store):** Servei d'emmagatzematge de blocs.
- **Amazon EFS (Elastic File System):** Servei d'emmagatzematge de fitxers en xarxa, compatible amb NFS.
- **Amazon RDS (Relational Database Service):** Servei de bases de dades gestionat per AWS.
- **Amazon VPC (Virtual Private Cloud):** Servei per crear xarxes virtuals en AWS.
- **AWS IAM (Identity and Access Management):** Servei per administrar usuaris i permisos d'accés als recursos d'AWS.
- **AWS KMS (Key Management Service):** Servei per crear i administrar claus de xifratge.
- **Amazon Lambda:** Servei basat en esdeveniments per executar codi sense gestionar servidors.
- **Amazon Elastic Beanstalk:** Servei per desplegar i escalar aplicacions web sense servidors.
- **Amazon ECS (Elastic Container Service):** Servei d'orquestració de contenidors.
- **Amazon EKS (Elastic Kubernetes Service):** Servei per executar Kubernetes en AWS.
- **AWS Fargate:** Servei per executar contenidors sense gestionar servidors.
- **Amazon Lightsail:** Solució VPS senzilla per a desenvolupadors i petites empreses.

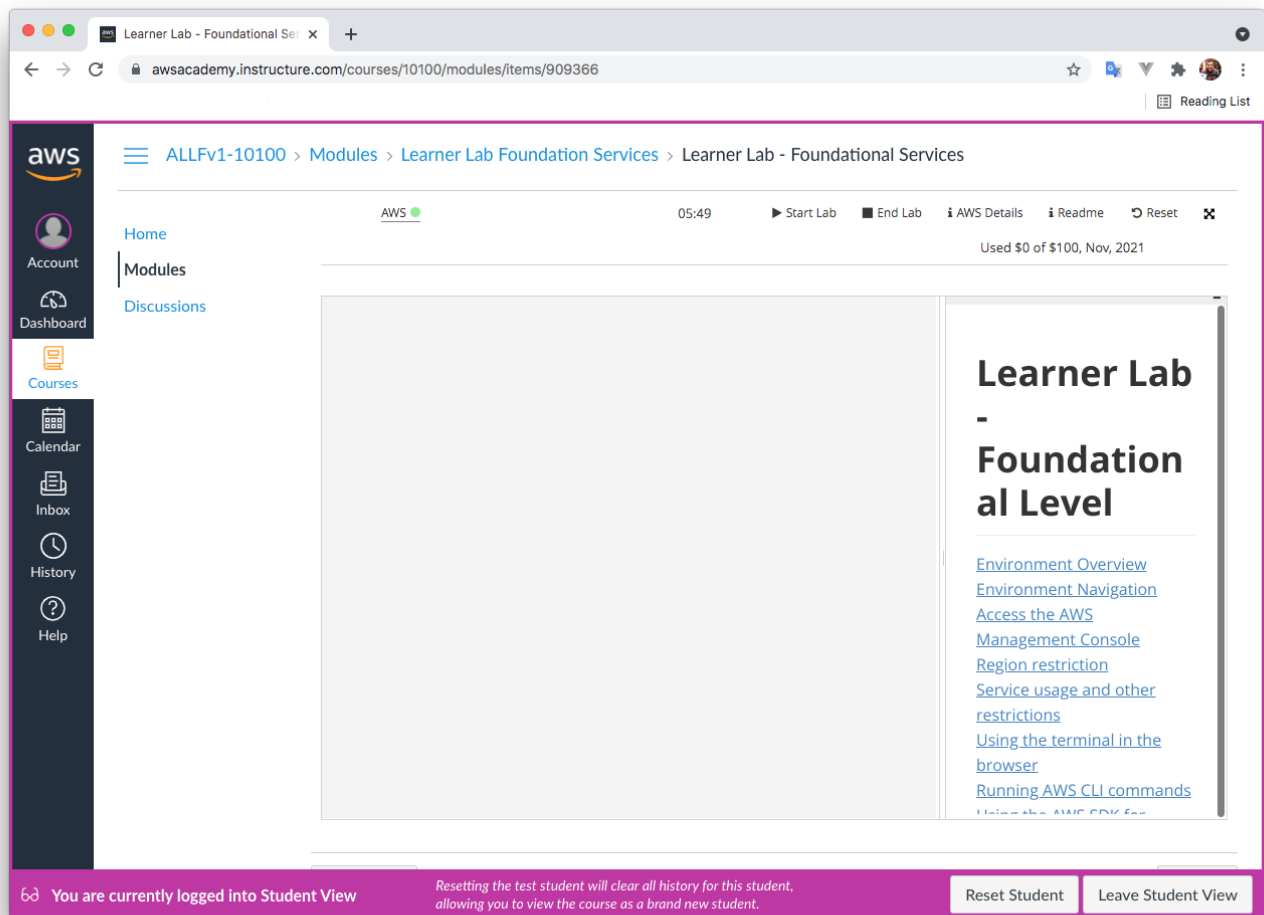
1.6 1.6 Com crear instàncies EC2?

En aquesta guia trobaràs els passos per crear instàncies EC2 en AWS.

Tutorial bàsic per crear instàncies EC2 en Amazon Web Services (AWS). (pdf)

1.6.1 1.6.1 Pas 1. Iniciem el laboratori

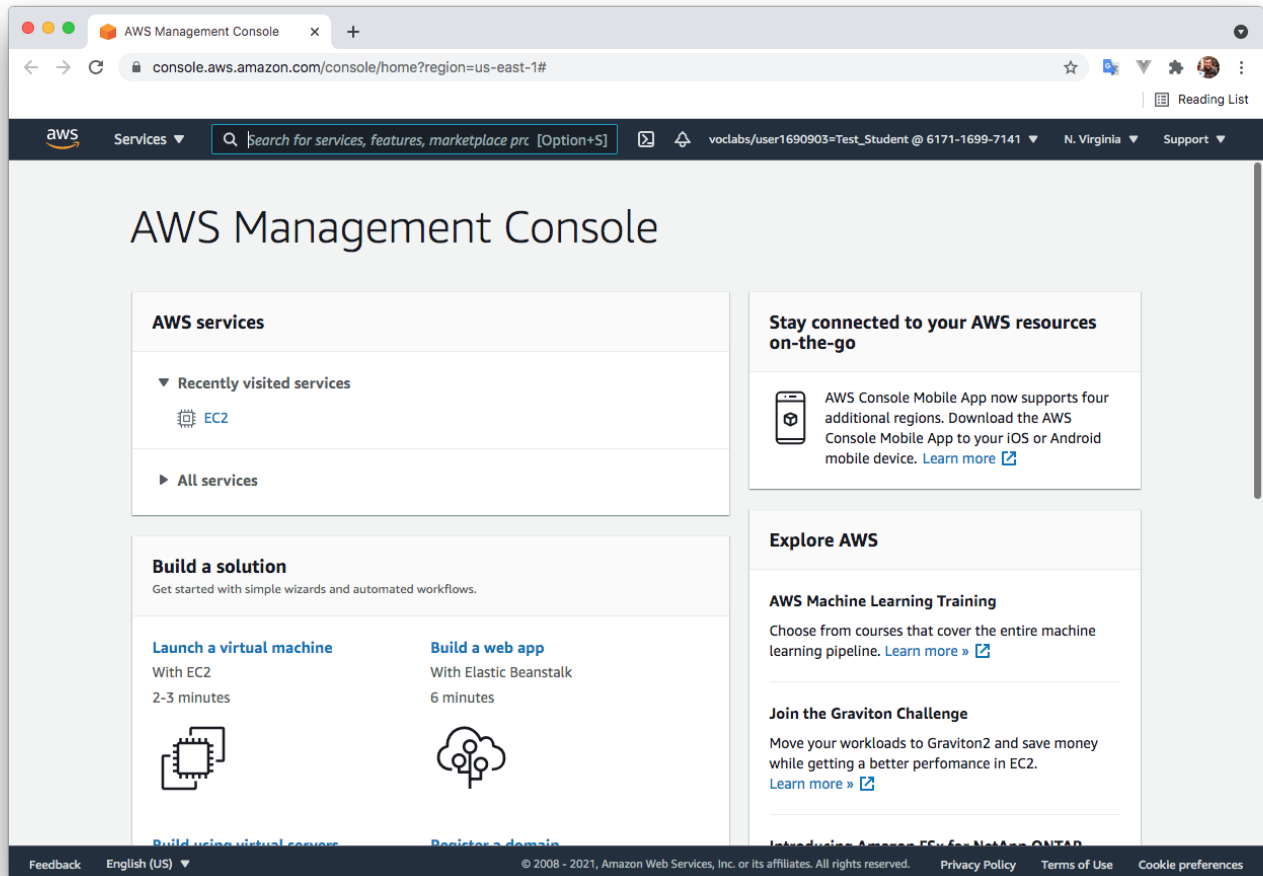
Des del curs d'AWS Academy iniciem el laboratori prement *Start Lab* i esperem que l'icona al costat del text d'AWS es torne verd.



1.6.2 1.6.2 Pas 2. Accedim a la consola de gestió d'AWS

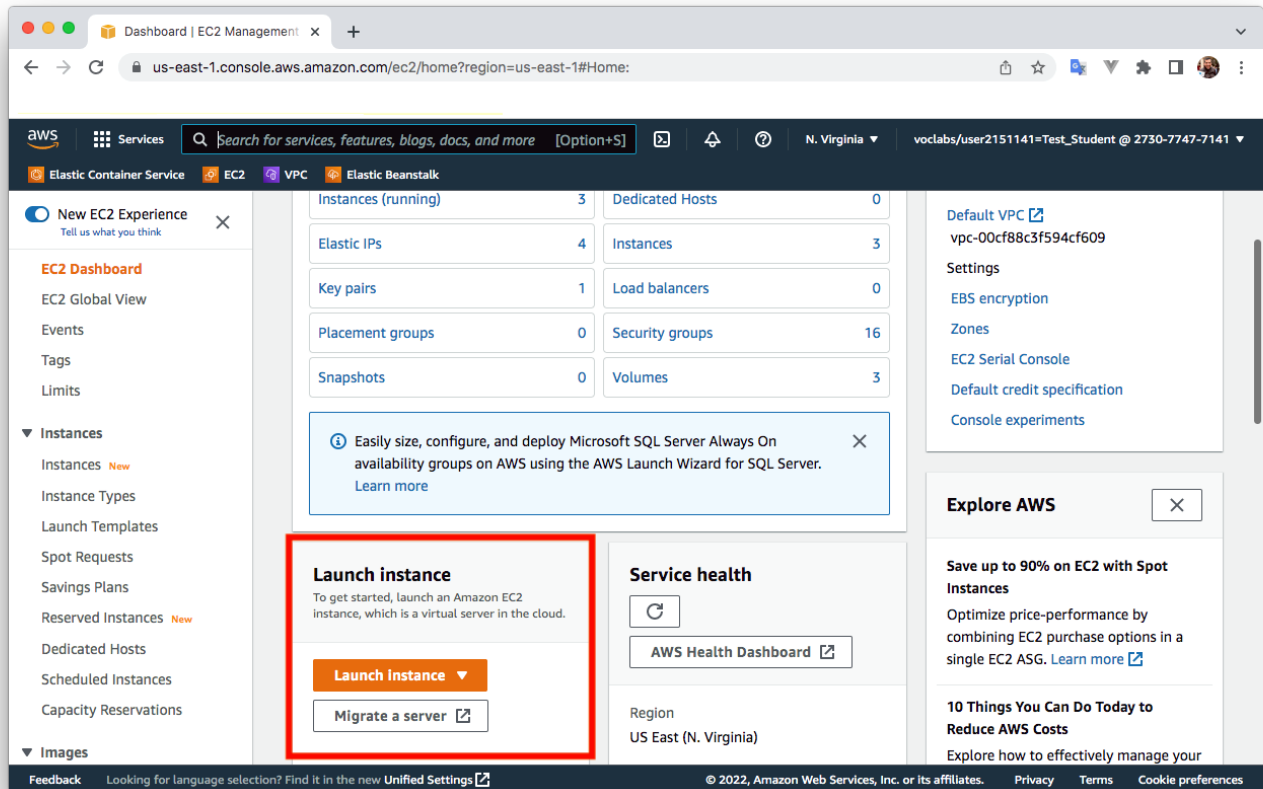
Premem sobre el text d'AWS, i s'obrirà una nova pestanya amb la consola de gestió d'AWS.

Seleccionem el servei EC2.



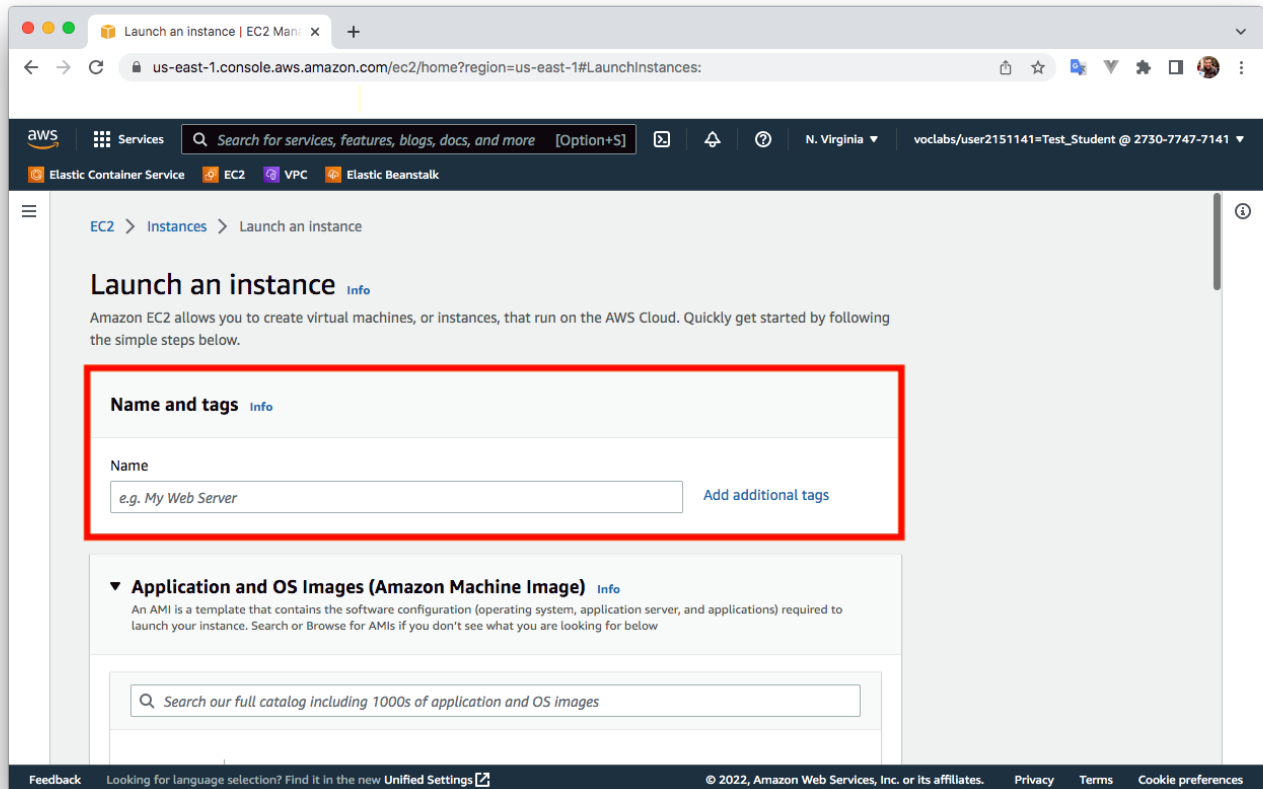
1.6.3 1.6.3 Pas 3. Cerquem el botó "Launch Instance" per crear una instància EC2

La manera més ràpida de crear una instància EC2 és a través del botó *Launch Instance* que apareix en el tauler del servei EC2.



1.6.4 1.6.4 Pas 4. Assignem un nom a la instància EC2 que anem a crear

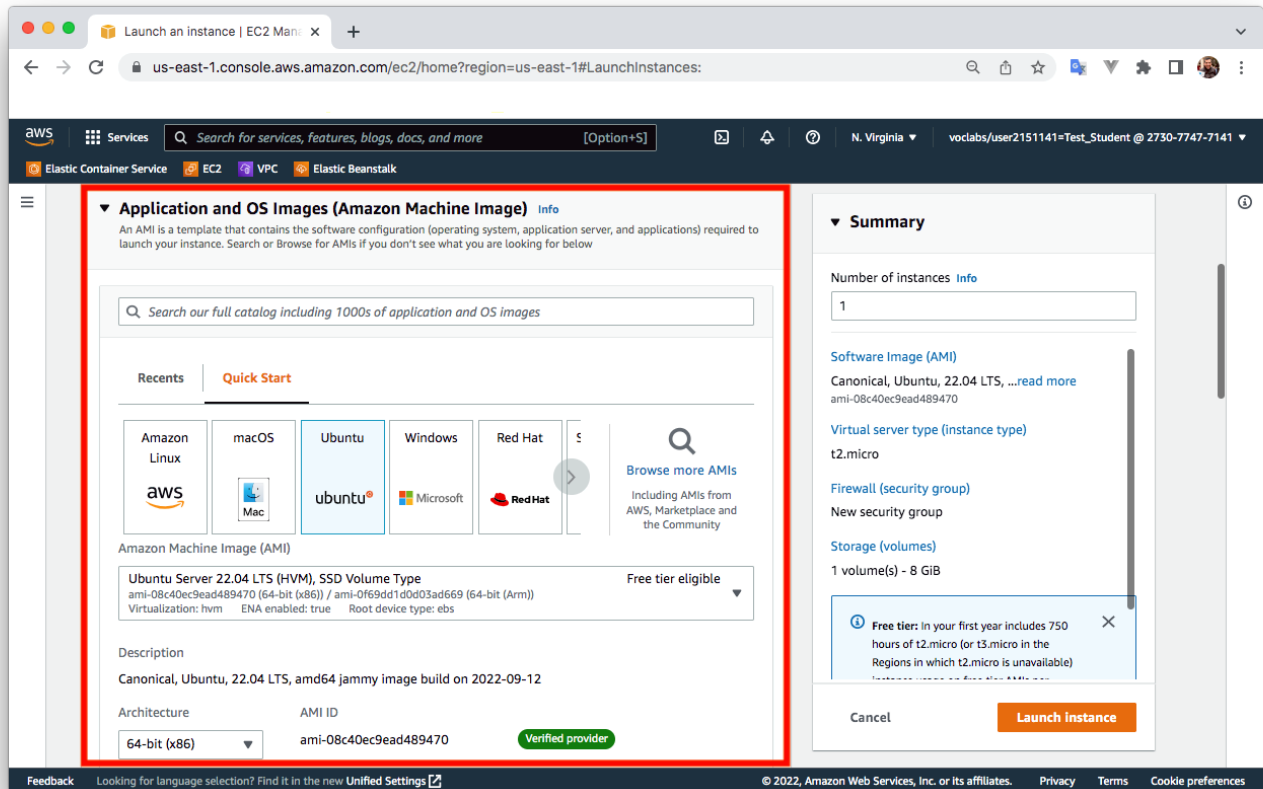
En aquest pas podem assignar-li un nom a la instància EC2 i afegir etiquetes amb metadades (parelles clau-valor). Les etiquetes permeten organitzar o filtrar les instàncies.



1.6.5 1.6.5 Pas 5. Seleccióem la AMI i l'arquitectura de la instància EC2

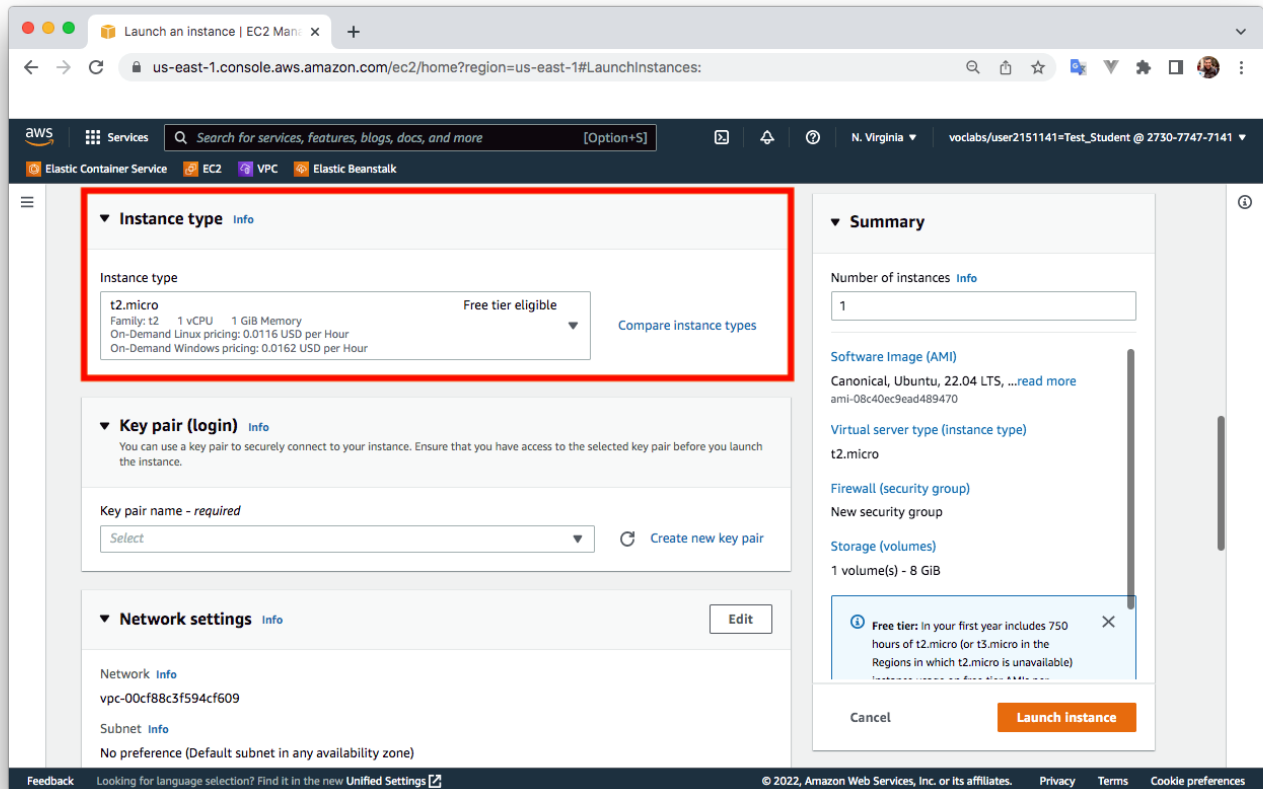
Seleccióem la AMI (Amazon Machine Image) i l'arquitectura (x86 o Arm).

En aquest exemple, seleccióem la imatge *Ubuntu Server 22.04 LTS (HVM), SSD Volume Type, 64 bits (x86)*.



1.6.6 1.6.6 Pas 6. Seleccionem el tipus d'instància EC2

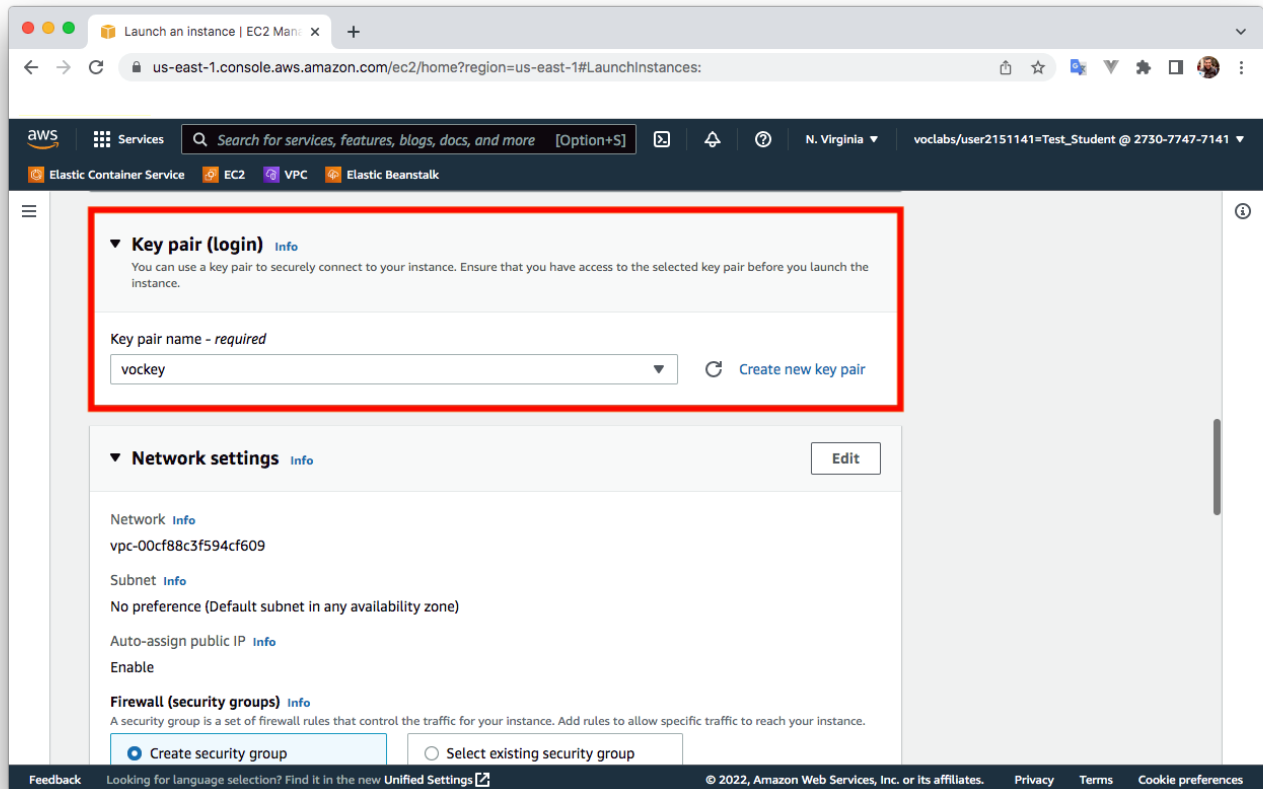
Seleccionem crear una instància de tipus *t2.micro*, amb 1 vCPU i 1 GiB de RAM.



1.6.7 1.6.7 Pas 7. Seleccionem la clau pública SSH que injectarem a la instància EC2

Seleccionem la clau pública SSH que injectarem a la instància. Podem generar un parell de claus SSH o utilitzar les claus associades al nostre compte en AWS Learner Lab.

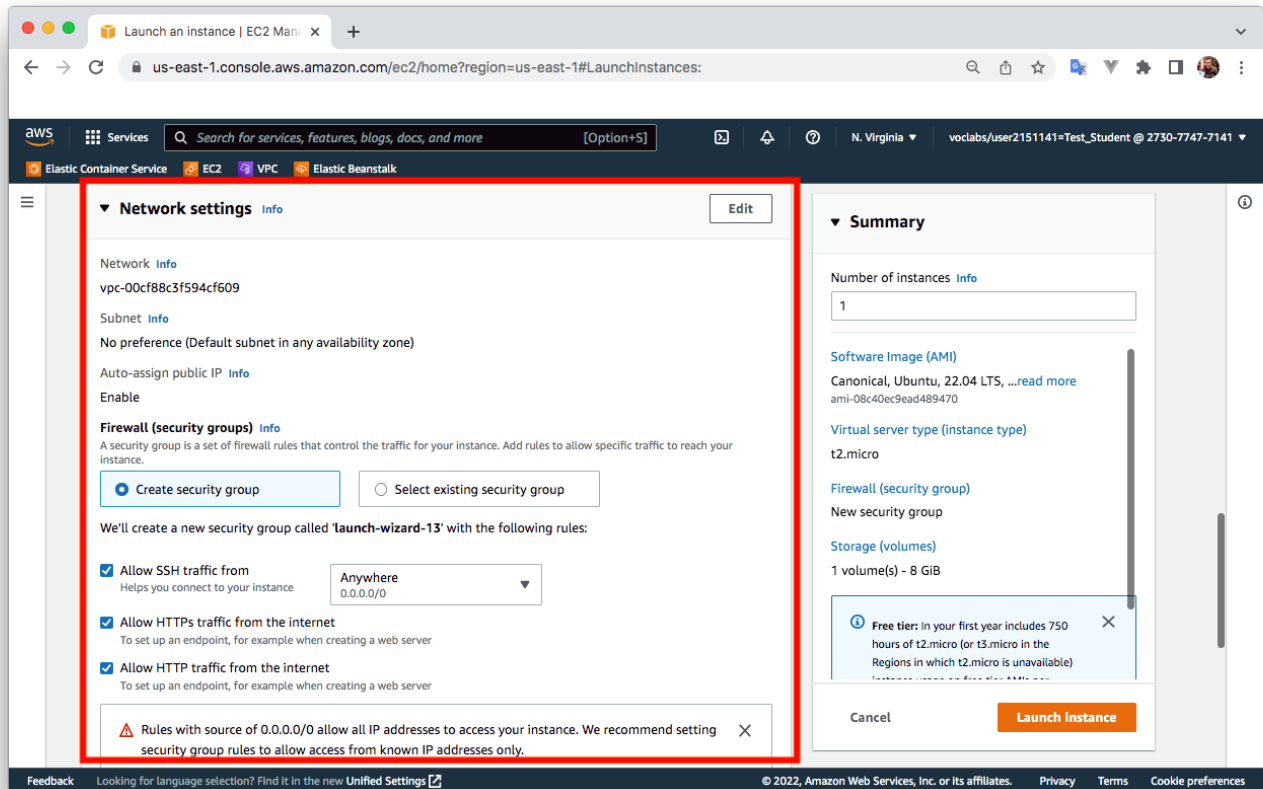
En aquest exemple, utilitzem la clau pública *vockey*.



1.6.8 1.6.8 Pas 8. Configurem la xarxa i els grups de seguretat

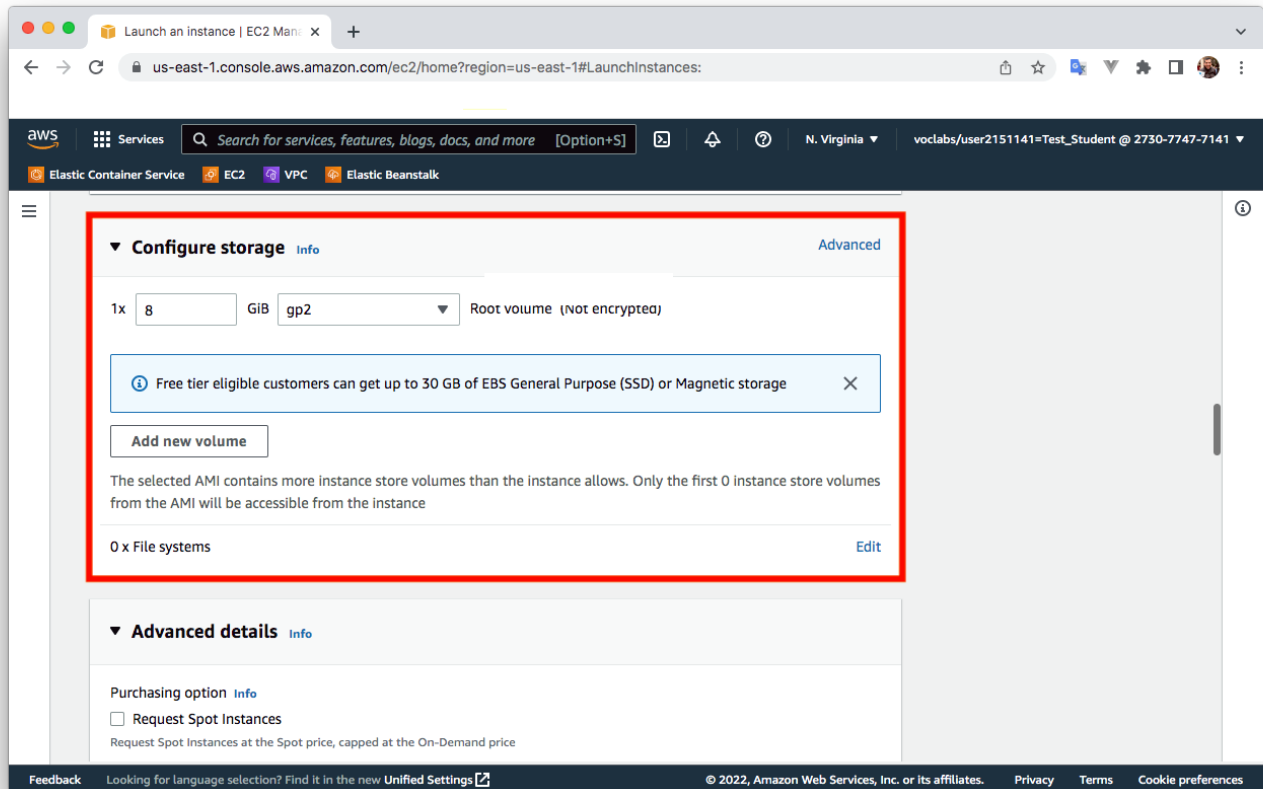
Podem configurar la xarxa VPC i les regles del grup de seguretat. En aquest exemple, creem un nou grup de seguretat amb les següents regles:

- **SSH.** Protocol: TCP. Port: 22. Origen: 0.0.0.0/0
- **HTTP.** Protocol: TCP. Port: 80. Origen: 0.0.0.0/0
- **HTTPS.** Protocol: TCP. Port: 443. Origen: 0.0.0.0/0



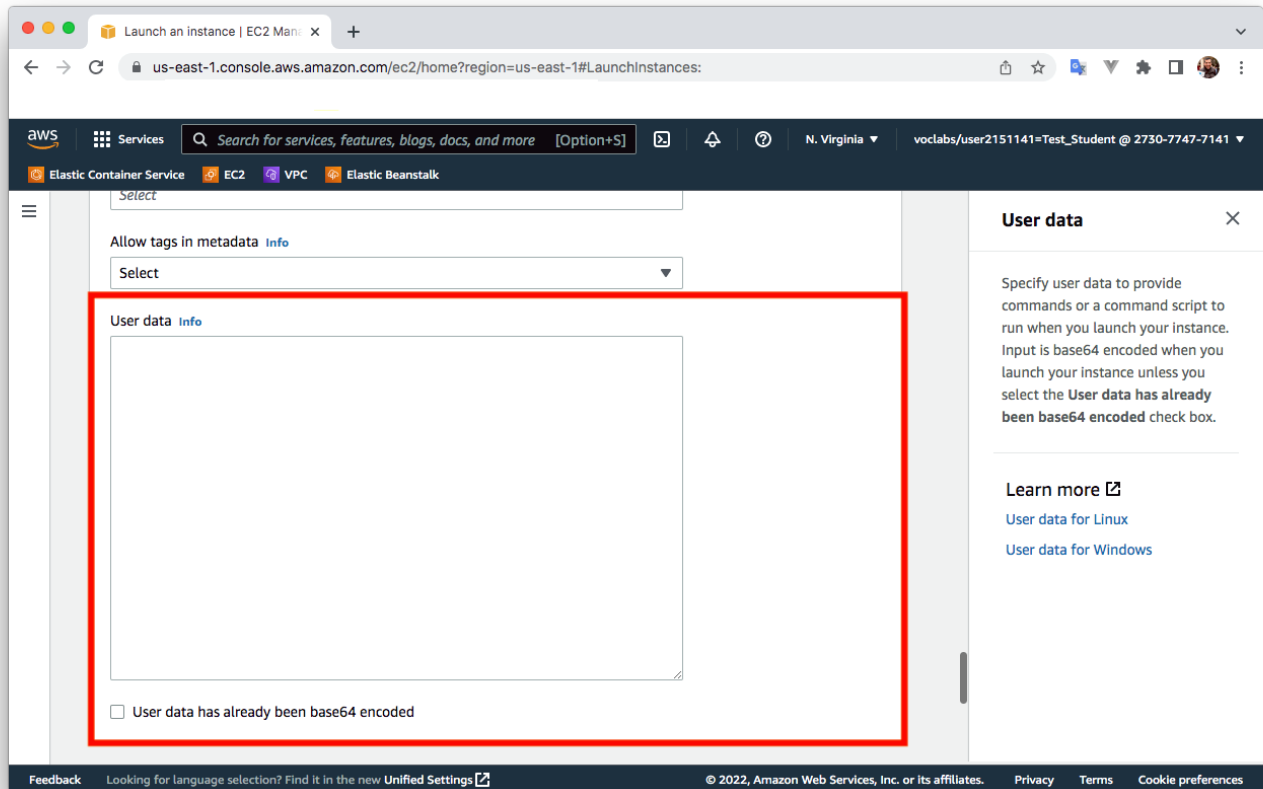
1.6.9 1.6.9 Pas 9. Afegim l'emmagatzematge que utilitzarà la instància

Configurem l'emmagatzematge de la instància EC2. En aquest exemple, utilitzem un disc SSD de 8 GB per defecte.



1.6.10 1.6.10 Pas 10. Detalls avançats

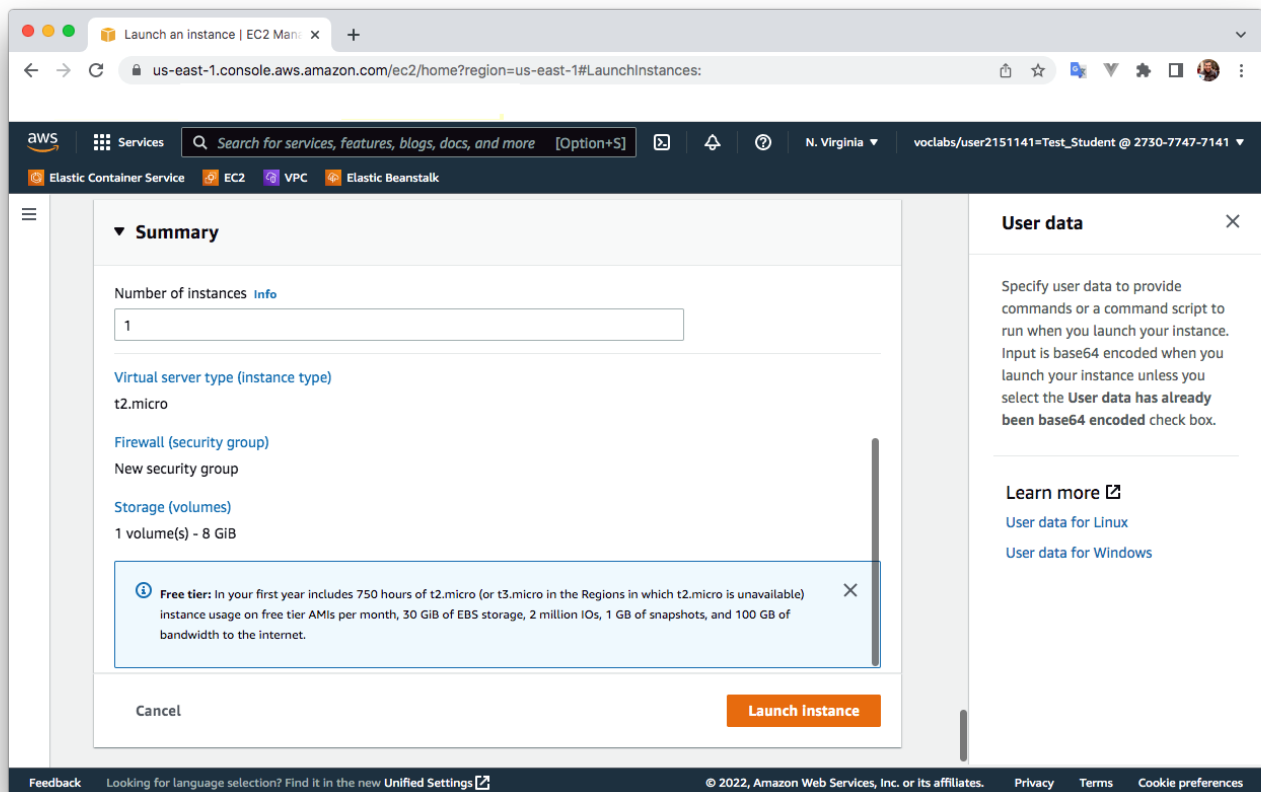
En aquest pas podem configurar alguns detalls avançats de la instància. Per exemple, en el camp *User data* podem indicar comandos o un script que s'executarà quan es reinici la instància. Aquesta opció permet preparar la instància perquè s'inici automàticament amb l'estat desitjat.



1.6.11 1.6.11 Pas 11. Revisem la configuració seleccionada

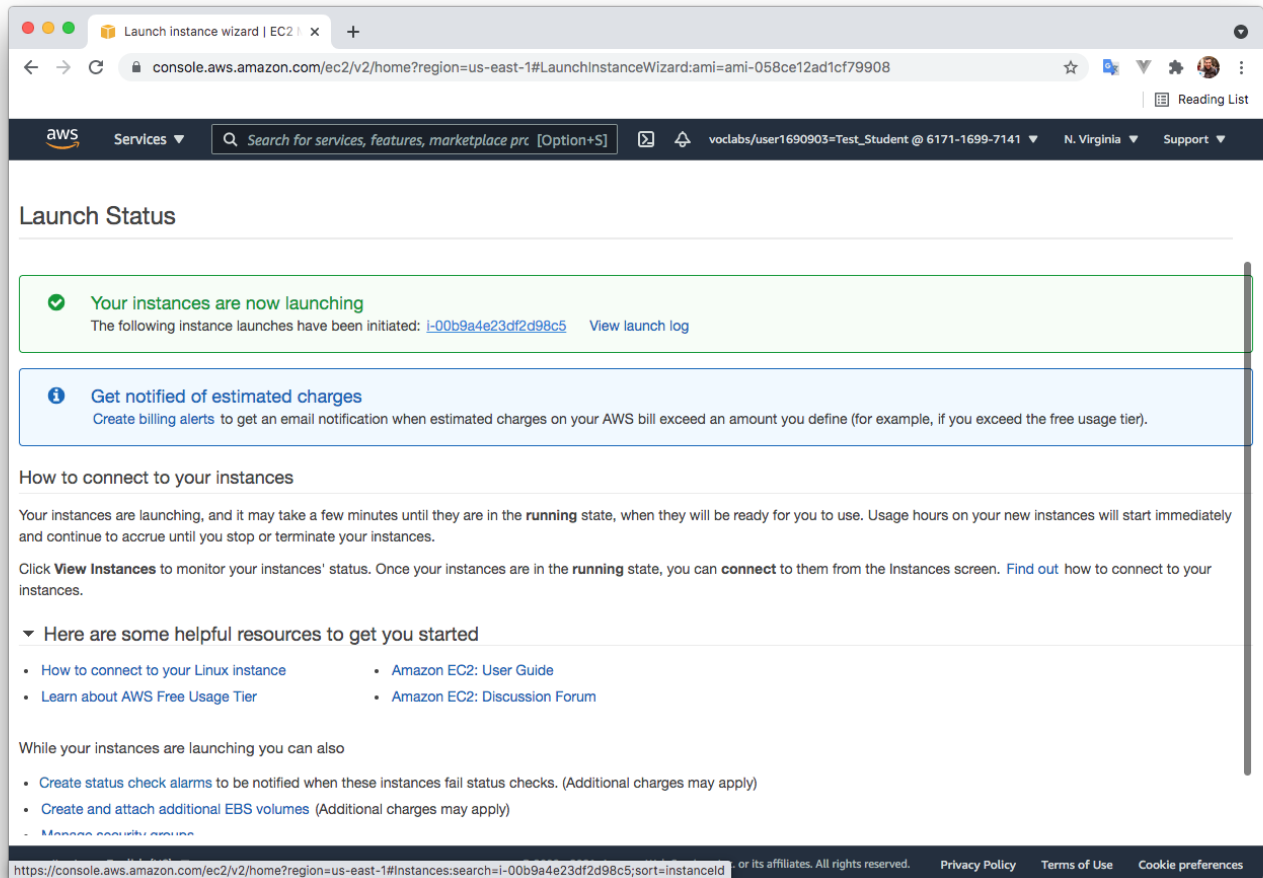
En aquest pas podem revisar la configuració seleccionada i triar el nombre d'instàncies que volem crear.

Per crear la instància, feu clic sobre el botó *Launch Instance*.



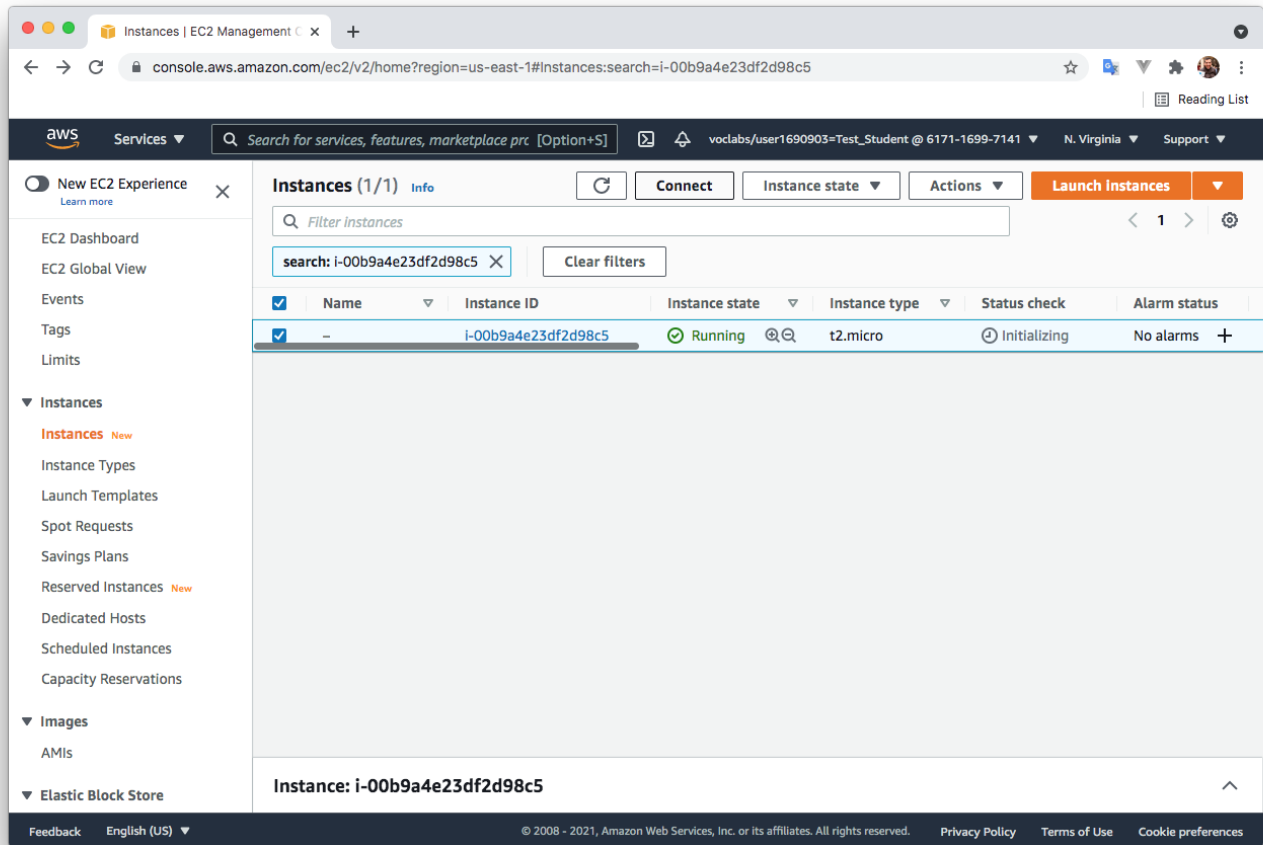
1.6.12 1.6.12 Pas 12. Estat de la instància

En aquest pas apareixerà un missatge indicant que la instància s'està creant. Farem clic sobre l'identificador de la instància que apareix en el quadre verd.



1.6.13 1.6.13 Pas 13. Consultem l'estat de la instància des de la consola de gestió

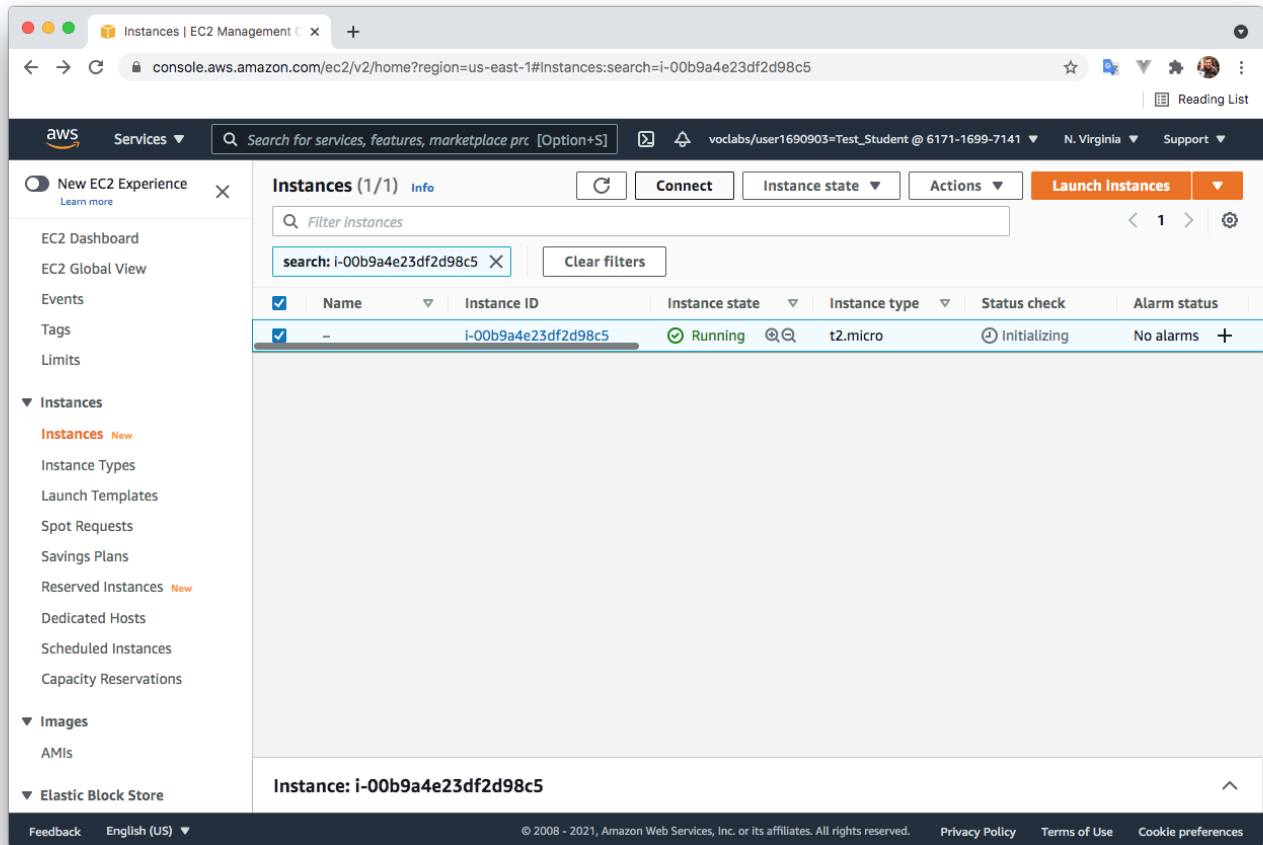
Després de pocs segons, la instància que hem creat apareixerà a la consola. Quan la instància mostre l'estat *Running*, estarà llesta per connectar-nos.



1.7 1.7 Com connectar-nos a una instància EC2 per SSH des de Linux

1.7.1 1.7.1 Pas 1. Seleccionem la instància i fem clic sobre Connect

Quan la instància mostre l'estat *Running*, marcarem el checkbox que apareix al principi de la línia i després farem clic sobre *Connect*.



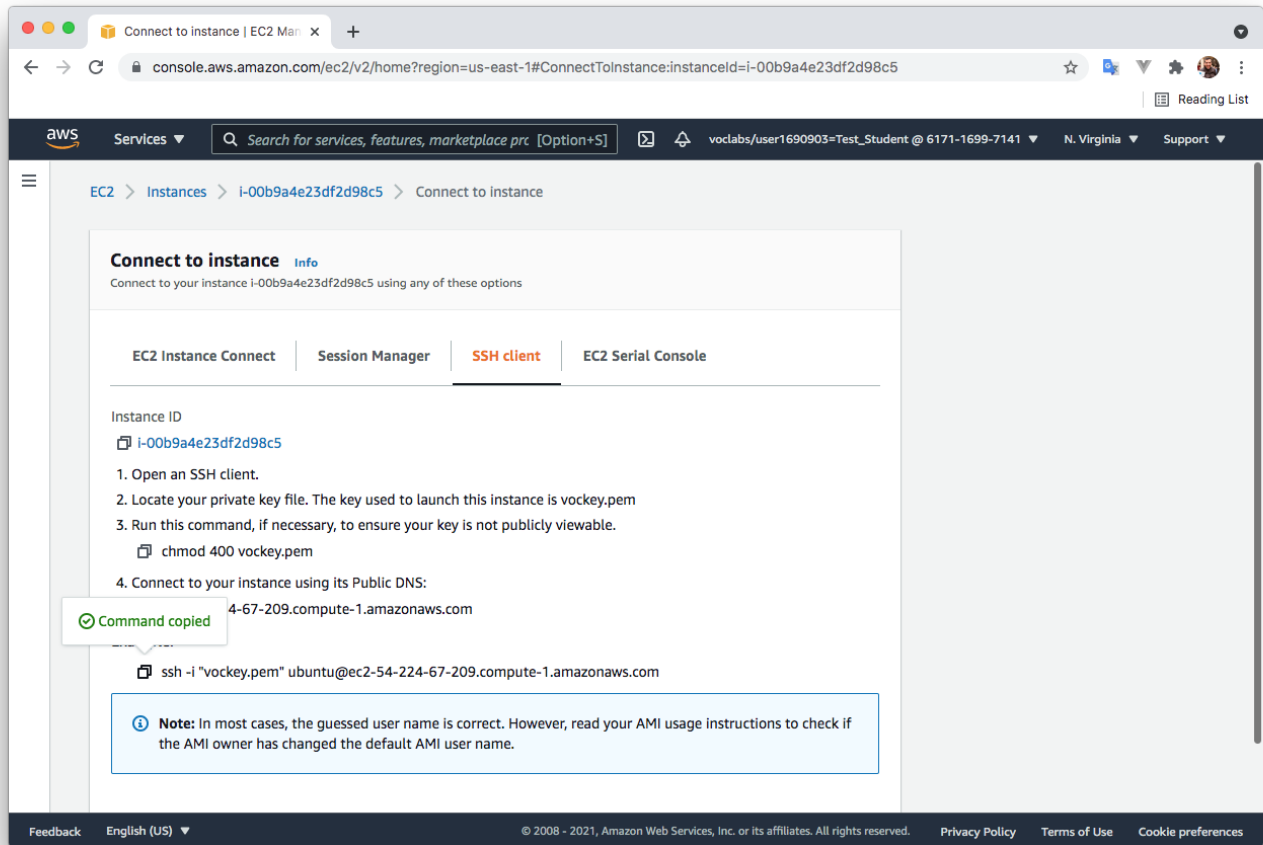
1.7.2 1.7.2 Pas 2. Obtenim el comando per connectar-nos des d'un client SSH en Linux

En aquest pas apareixerà com podem connectar-nos a la instància que acabem de crear.

Seleccionarem l'opció *SSH client* i llegirem els passos per connectar-nos a la instància. Podem copiar el comando que apareix en aquest pas, que conté el nom de l'arxiu .pem de la clau privada, el nom d'usuari i el DNS públic de la instància.

Exemple:

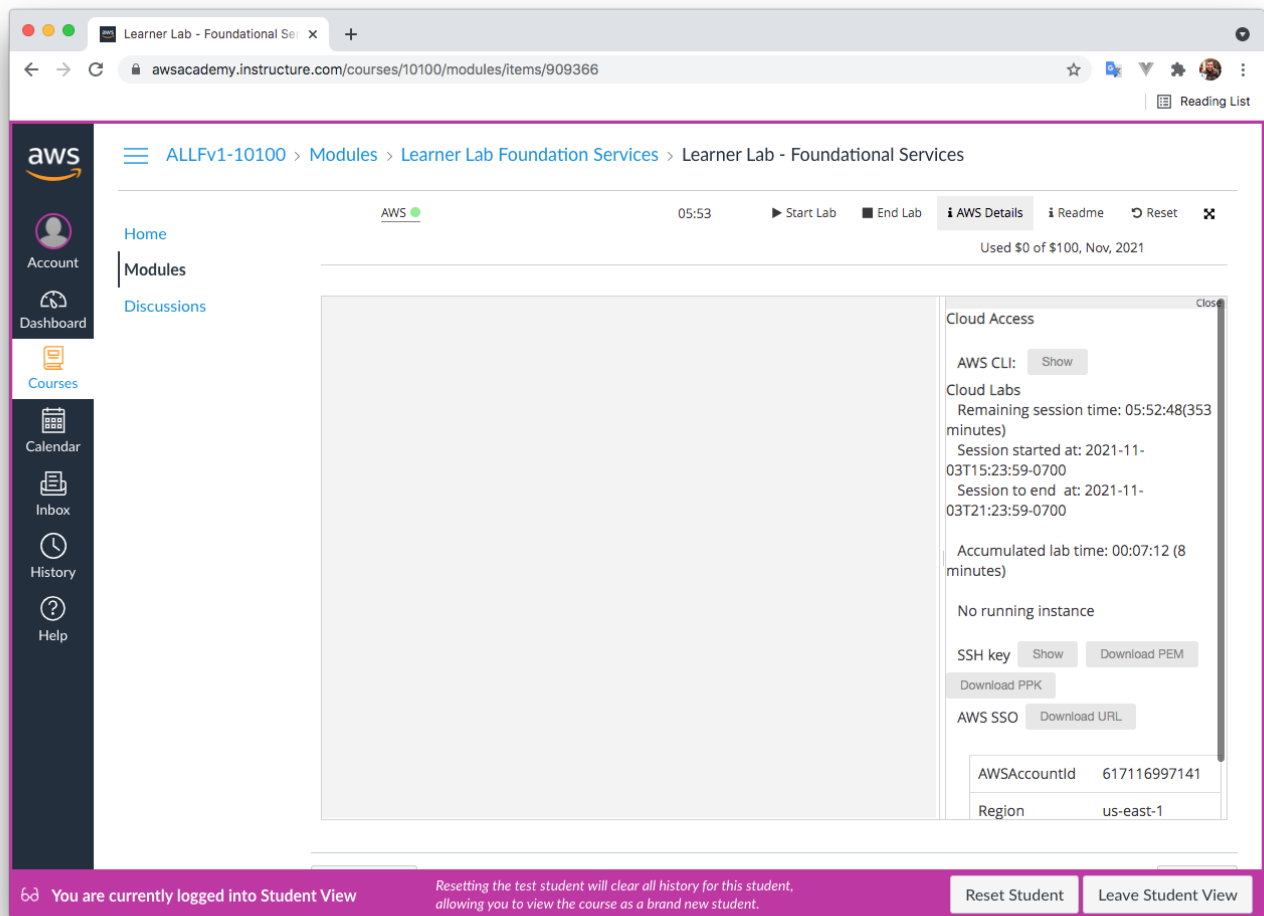
```
1 $ ssh -i "vockey.pem" ubuntu@ec2-54-224-67-209.compute-1.amazonaws.com
```



1.8 1.7.3 Pas 3. Descarreguem la clau privada .PEM des del Learner Lab

Per descarregar la clau, fem clic sobre *AWS Details*, i apareixerà la possibilitat de descarregar la clau com un arxiu .PEM o .PPK.

En Linux utilitzarem la clau .PEM, i en Windows, si usem PuTTY, utilitzarem la clau .PPK. Quan descarregueu la clau, es descarregarà com `labuser.pem` o `labuser.ppk`.



1.9 1.7.4 Pas 4. Connectar-nos per SSH des de Linux

Quan descarregueu la clau privada `labuser.pem`, guardeu-la en un directori i, des d'allí, realitzeu els següents passos.

Renombreu l'arxiu `labuser.pem` com `vockey.pem`:

```
1 $ mv labuser.pem vockey.pem
```

IMPORTANT: canviem els permisos de l'arxiu perquè només el propietari tinga permisos de lectura:

```
1 $ chmod 400 vockey.pem
```

Executem el comando copiat en el pas 2 per connectar-nos per SSH a la instància EC2 des d'un terminal de Linux. El comando serà semblant a aquest, però amb un DNS públic diferent:

```
1 $ ssh -i "vockey.pem" ubuntu@ec2-54-224-67-209.compute-1.amazonaws.com
```

Esta pàgina forma part del curs Implantació d'Aplicacions Web de José Juan Sánchez Hernández i el seu contingut es distribueix sota una llicència [Creative Commons Reconeixement-NoComercial-CompartirIgual 4.0 Internacional](#)

2. Instalación MySql Server

2.1 Instalación utilizando binarios

Oracle proporciona un conjunto de distribuciones binarias de MySQL. Estas incluyen distribuciones genéricas binarias en forma de archivos **tar**, archivos con una tar.xz como extensión. Existen diferentes binarios específicos para cada plataforma. En esta sección vamos a ver la instalación de MySQL desde una distribución binaria de archivo **tar** comprimido para la plataforma **Unix/Linux**.

Ejemplo para descargar el binario:

```
1 terminal> wget https://dev.mysql.com/get/Downloads/MySQL-8.0/mysql-8.0.26-linux-glibc2.12-x86_64.tar.xz
```

Advertencias: * Si ha instalado previamente una versión de MySQL utilizando YUM/APT (Gestores de paquetes) puede experimentar problemas. Asegurarse de eliminar las aplicaciones y ficheros de configuración completamente del sistema * MySQL depende de la biblioteca **libaio**, si es necesario instalela utilizando el administrador de paquetes adecuado.

```
1 terminal> apt-cache search libaio # search for info
2 terminal> apt-get install libaio1 # install library for APT-based systems
```

Para instalar una distribución binaria de archivo tar comprimido, descomprímalo en la ubicación de instalación que elija (normalmente /usr/local/mysql). Esto crea los directorios que se muestran en la siguiente tabla.

Directorio	Contenido
bin	programas de utilidad, cliente y servidor mysqld
docs	Manual de MySQL en formato Info
man	Páginas de manual de Unix
include	Páginas de manual de Unix
lib	Bibliotecas
share	Mensajes de error, diccionario y SQL para la instalación de la base de datos
support-files	Archivos de soporte varios

La secuencia de comandos para instalar MySQL es la siguiente:

```
1 terminal> groupadd mysql
2 terminal> useradd -r -g mysql -s /bin/false mysql
3 terminal> cd /usr/local
4 terminal> tar xvf /path/to/mysql-VERSION-OS.tar.xz
5 terminal> ln -s full-path-to-mysql-VERSION-OS mysql
6 terminal> export PATH=$PATH:/usr/local/mysql/bin
7 terminal> cd mysql
8 terminal> mkdir mysql-files
9 terminal> chown mysql:mysql mysql-files
10 terminal> chmod 750 mysql-files
11 terminal> bin/mysqld --initialize --user=mysql
12 terminal> bin/mysql_ssl_rsa_setup
13 terminal> bin/mysqld_safe --user=mysql & # Next command is optional
14 terminal> cp support-files/mysql.server /etc/init.d/mysql.server
```

Este proceso asume que tenemos acceso **root** al sistema. Alternativamente podemos prefijar cada comando usando el comando **sudo**

2.2 Crear un usuario y un grupo de mysql

Si su sistema aún no tiene un usuario y un grupo para usar para ejecutar **mysqld**, es posible que deba crearlos. La sintaxis de **useradd** y **groupadd** puede diferir ligeramente en diferentes versiones de **Unix/Linux**, o pueden tener diferentes nombres como **adduser** y **addgroup**.

```
1 terminal> groupadd mysql
2 terminal> useradd -r -g mysql -s /bin/false mysql
```

Debido a que el usuario solo es necesario para fines de propiedad, no de inicio de sesión, el comando **useradd** utiliza las opciones **-ry -s /bin/false** para crear un usuario que no tenga permisos de inicio de sesión en el servidor.

2.3 Obtenga y desempaquete la aplicación

Elija el directorio en el que desea descomprimir la distribución y cambie la ubicación en él. El ejemplo aquí descomprime la distribución en **/usr/local**. Por lo tanto, las instrucciones asumen que tiene permiso para crear archivos y directorios en **/usr/local**. Si ese directorio está protegido, debe realizar la instalación como **root**. Obtenga un archivo de distribución siguiendo las instrucciones de la [“Cómo obtener MySQL”](#) de la documentación de MySQL. Para una versión determinada, las distribuciones binarias para todas las plataformas se crean a partir de la misma distribución fuente de MySQL. Desempaquete la distribución, que crea el directorio de instalación. **tar** puede descomprimir y descomprimir la distribución si tiene la opción de soporte:

```
1 terminal> cd /usr/local
2 terminal> tar xvf /path/to/mysql-VERSION-OS.tar.xz
```

El comando **tar** crea un directorio llamado **.mysql-VERSION-OS**

Si su **tar** no es compatible con **xz** utilizar el siguiente comando:

```
1 terminal> xz -dc /path/to/mysql-VERSION-OS.tar.xz | tar x
```

Por último crear el enlace simbólico a la carpeta de instalación y agregar el directorio al **path** para que sea más sencillo acceder a los programas.

```
1 terminal> ln -s full-path-to-mysql-VERSION-OS mysql
2 terminal> export PATH=$PATH:/usr/local/mysql/bin
```

Por último saltar a: Configuración y pruebas posteriores a la instalación.

2.4 Instalación utilizando el gestor de paquetes APT de linux.

Al igual que en el punto anterior, asumimos que en el sistema no se ha instalado previamente ninguna versión de MySQL Server o ha sido eliminada completamente.

2.4.1 Pasos para una nueva instalación de MySQL

Primero agregar el paquete **.deb** del repositorio APT de MySQL. Podemos descargarlo del [repositorio](#). Instalar el paquete con el siguiente comando (reemplazar **x-y-z** con la versión utilizada)

```
1 terminal> sudo dpkg -i mysql-apt-config_w.x.y-z_all.deb
```

Durante la instalación del paquete, se pedirá que elija las versiones del servidor MySQL y otros componentes (por ejemplo, MySQL Workbench) que quiere instalar. Si no estás seguro de qué versión elegir, no cambies las opciones predeterminadas seleccionadas. Actualizar la información de los repositorios **paso obligado**

```
1 terminal> sudo apt-get update
```

2.4.2 Instalación del paquete con APT

Instalar con el comando:

```
1 terminal> sudo apt-get install mysql-server
```

Esto instala el paquete para el servidor MySQL, así como los paquetes para el cliente y para los archivos comunes de la base de datos. Durante la instalación, se le pedirá que proporcione una **contraseña** para el usuario **root** para su instalación de MySQL.

2.4.3 Iniciar y detener el servidor

El servidor MySQL se inicia automáticamente después de la instalación. Puede verificar el estado del servidor MySQL con el siguiente comando:

```
1 terminal> systemctl status mysql
```

Si el sistema operativo está habilitado para systemd, los comandos estándar systemctl (o alternativamente, servicio con los argumentos invertidos) como detener, iniciar, estado y reiniciar deben usarse para administrar el servicio del servidor MySQL. El servicio mysql está habilitado de forma predeterminada y se inicia al reiniciar el sistema. Consulte Administrar MySQL Server con systemd para obtener información adicional.

Opciones de **systemctl**.

Opcion	Significado	Comando
stop	Detener el servicio	systemctl stop mysql
start	Iniciar el servicio	systemctl start mysql
restart	Reiniciar el servicio (star/stop)	systemctl restart mysql
reload	Recargar la configuración	systemctl reload mysql

2.4.4 Instalar componentes adicionales de MySql

Puedes usar APT para instalar componentes individuales de MySQL desde el repositorio APT de MySQL. Suponiendo que ya tiene el repositorio APT de MySQL en la lista de repositorios de su sistema. Primero, use el siguiente comando para obtener la información más reciente del paquete del repositorio APT de MySQL:

```
1 terminal> sudo apt-get update
```

Ejemplos:

```
1 terminal> sudo apt-get install mysql-workbench-community #cliente workbench
2 terminal> sudo apt-get install libmysqlclient21 #bibliotecas cliente compartidas
```

2.5 Actualizar la versión de MySql Server

Advertencias: * Tener una copia de seguridad previa de la información antes de actualizar * Si es posible probar primero en un servidor de preproducción * Los comandos que veremos a continuación no sirven si la instalación previa no es de Oracle... Si es MariaDB o Percona

Primero asegurarse que tenemos el repositorio MySQL apt en nuestra lista de repositorios Actualizar el sistema

```
1 terminal> sudo apt-get update
```

Como regla general, para actualizar de una serie de lanzamientos a otra, vaya a la siguiente serie en lugar de omitir una serie. Por ejemplo, si actualmente estás ejecutando MySQL 5.5 y deseas actualizar a una serie más nueva, actualiza primero a MySQL 5.6 antes de actualizar a 5.7, y así sucesivamente. Actualizar con el siguiente comando:

```
1 terminal> sudo apt-get install mysql-server
```

Si hay que actualizar otros paquetes utilizar el mismo comando:

```
1 terminal> sudo apt-get install package-name
```

Para saber el nombre de los paquetes instalados desde el repositorio APT de MySQL, use el siguiente comando:

```
1 terminal> dpkg -l | grep mysql | grep ii
```

Advertencia: Si realizas una actualización de todo el sistema mediante ***apt-get upgrade***, solo la biblioteca MySQL y los paquetes de desarrollo se actualizan con versiones más recientes (si están disponibles). Para actualizar otros componentes, incluido el servidor, el cliente, la suite de pruebas, etc., use el comando `apt-get install`. El servidor MySQL siempre se reinicia después de una actualización de APT. Antes de MySQL 8.0.16, ejecute ***mysql_upgrade*** después de que el servidor se reinicia para verificar y posiblemente resolver cualquier incompatibilidad entre los datos antiguos y el software actualizado. ***mysql_upgrade*** también realiza otras funciones; consulte [mysql_upgrade](#). A partir de MySQL 8.0.16, este paso no es necesario, ya que el servidor realiza todas las tareas previamente manejadas por ***mysql_upgrade***.

2.5.1 Eliminando MySql con APT

Para desinstalar el servidor MySQL y los componentes relacionados que se han instalado usando el repositorio APT de MySQL, elimine el servidor MySQL y cualquier otro software que se haya instalado automáticamente usando los siguientes comandos:

```
1 terminal> sudo apt-get remove mysql-server
2 terminal> sudo apt-get autoremove
```

Para desinstalar otros paquetes instalados manualmente utilizar:

```
1 terminal> sudo apt-get remove package-name
```

Para ver una lista de los paquetes que han instalado desde el repositorio APT de MySQL, use el siguiente comando:

```
1 terminal> dpkg -l | grep mysql | grep ii
```

2.6 Paquetes disponibles del repositorio APT de Mysql

Package Name	Description
mysql-server	Metapackage for installing the MySQL server
mysql-community-server	MySQL server
mysql-client	Metapackage for installing the MySQL client
mysql-cluster-community-auto-installer *	Auto installer for NDB Cluster
mysql-cluster-community-client	MySQL client for NDB Cluster
mysql-cluster-community-data-node	NDB Cluster data node
mysql-cluster-community-java	NDB Cluster Java drivers
mysql-cluster-community-management-server	NDB Cluster management node
mysql-cluster-community-memcached	NDB Cluster memcached server
mysql-cluster-community-nodejs	NDB Cluster Node.js adapters
mysql-cluster-community-server	MySQL server for NDB Cluster
mysql-cluster-community-source	Source package for NDB Cluster
mysql-cluster-community-test	NDB Cluster testsuite
mysql-community-client	MySQL client
mysql-common	MySQL database common files
libmysqlclient21	MySQL database client library
libmysqlclient-dev	MySQL database development files
libmysqld-dev	MySQL embedded database development files
mysql-testsuite	Metapackage for installing the MySQL test suite
mysql-community-test	MySQL test suite
mysql-community-bench	MySQL benchmark suite
mysql-community-source	MySQL source code
mysql-workbench-community	MySQL Workbench (not available for Debian platforms)
mysql-connector-python-py3	MySQL Connector/Python for supported Ubuntu versions with Python 3.2 or later
mysql-connector-python	MySQL Connector/Python for supported Debian versions with Python 2.6.3 or later, and for supported Ubuntu versions with Python 2.6.3 to 3.1
mysql-router	MySQL Router
ndbclient	NDB Cluster client
ndbclient-dev	NDB Cluster client development library

2.7 Configuración y pruebas posteriores a la instalación

Después de instalar MySQL, se debe inicializar el directorio de datos, incluidas las tablas en el esquema `mysql` del sistema: * Para algunos métodos de instalación de MySQL, la inicialización del directorio de datos es automática * Para otros métodos de instalación, se debe inicializar el directorio de datos manualmente. Estos incluyen la instalación desde distribuciones genéricas binarias y de origen en sistemas Unix y similares a Unix, y la instalación desde un paquete ZIP en Windows. En general, cuando instalamos desde un instalador (exe o msi en Windows o apt/yum en linux) la inicialización del directorio es automática. Cuando la instalación es mediante la copia de binarios o compilación hay que iniciar estos directorios manualmente.

2.7.1 Inicialización del directorio de datos

Descripción general de la inicialización

Esta sección describe cómo inicializar el directorio de datos manualmente para los métodos de instalación de MySQL para los cuales la inicialización del directorio de datos no es automática. Advertencia. En MySQL 8.0, el complemento de autenticación predeterminado ha cambiado de `mysql_native_password` a `caching_sha2_password`, y la cuenta '`root`'@'`localhost`' administrativa utiliza `caching_sha2_password` de forma predeterminada. Si prefiere que la cuenta **root** utilice el complemento de autenticación predeterminado anterior (`mysql_native_password`), consulte [caching_sha2_password](#) y [la cuenta administrativa raíz](#). Los pasos son los siguientes: * Cambiar la ubicación al nivel superior de instalación de mysql * Crear un directorio * Otorgar la propiedad de la carpeta al usuario y grupo mysql * Utilizar el servidor para inicializar el directorio de datos

```
1 terminal> cd /usr/local/mysql
2 terminal> mkdir mysql-files
3 terminal> chown mysql:mysql mysql-files
4 terminal> chmod 750 mysql-files
5 terminal> bin/mysqld --initialize --user=mysql
```

Si queremos implementar el soporte automático para conexiones seguras, hay que utilizar la tilidad **mysql_ssl_rsa_setup** para crear los archivos SSL y RSA respectivamente

```
1 terminal> bin/mysql_ssl_rsa_setup
```

Procedimiento de inicialización del directorio de datos

Cambiar la ubicación al nivel superior de instalación de mysql.

```
1 terminal> cd /usr/local/mysql
```

Para inicializar el directorio de datos, invoque `mysqld` con la opción `--initialize` o `--initialize-insecure`, dependiendo de si desea que el servidor genere una contraseña inicial aleatoria para la cuenta '`root`'@'`localhost`', o cree esa cuenta sin contraseña:

- Use `--initialize` para la instalación "segura por defecto" (es decir, incluida la generación de una rootcontraseña inicial aleatoria). En este caso, la contraseña se marca como caducada y debe elegir una nueva.
- Con `--initialize-insecure`, no se genera ninguna contraseña para root. **Esto es inseguro**; se supone que tiene la intención de asignar una contraseña a la cuenta de manera oportuna antes de poner el servidor en uso de producción.

En sistemas Unix y similares a Unix, es importante que los directorios y archivos de la base de datos sean propiedad de la cuenta `mysql` de inicio de sesión para que el servidor tenga acceso de lectura y escritura a ellos cuando lo ejecute más tarde. Para garantizar esto inicia `mysqld` desde la cuenta `root` del sistema e incluye la opción `--user` como se muestra aquí:

```
1 terminal> bin/mysqld --initialize --user=mysql
2 terminal> bin/mysqld --initialize-insecure --user=mysql
```

Podría ser necesario especificar otras opciones, como `--basedir` o `--datadir` si `mysqld` no puede identificar las ubicaciones correctas para el directorio de instalación o el directorio de datos. Por ejemplo (ingrese el comando en una sola línea):

```
1 terminal> bin/mysqld --initialize --user=mysql
2 --basedir=/opt/mysql/mysql
3 --datadir=/opt/mysql/mysql/data
```

Alternativamente, coloca la configuración de opciones relevante en un archivo de opciones y pase el nombre de ese archivo a mysqld. Para sistemas Unix y similares a Unix, supón que el nombre del archivo de opciones es `/opt/mysql/mysql/etc/my.cnf`. Pon estas líneas en el archivo:

```
1 [mysqld]
2 basedir=/opt/mysql/mysql
3 datadir=/opt/mysql/mysql/data
```

Acciones en el servidor durante la inicialización del directorio de datos

Cuando se invoca con la opción `--initialize` o `--initialize-insecure`, mysqld realiza las siguientes acciones durante la secuencia de inicialización del directorio de datos: * El servidor verifica la existencia del directorio de datos (si no existe lo crea) * Si el directorio de datos existe pero no está vacío, el servidor se cierra después de producir un mensaje de error:

```
1 [ERROR] --initialize specified but the data directory exists. Aborting.
```

En este caso, elimine o cambie el nombre del directorio de datos y vuelva a intentarlo. Se permite que un directorio de datos existente no esté vacío si cada entrada tiene un nombre que comienza con un punto (.). * Dentro del directorio de datos el servidor crea el esquema de mysql y sus tablas, incluidas las tablas del diccionario de datos, las tablas de concesión, las tablas de zona horaria y las tablas de ayuda del lado del servidor. * El servidor inicializa el espacio de la tabla del sistema y las estructuras de datos relacionadas necesarias para administrar las tablas InnoDB * El servidor crea una cuenta 'root'@'localhost' de superusuario y otras cuentas reservadas + Con `--initialize` pero sin `--initialize-insecure`, el servidor genera una contraseña aleatoria, la marca como caducada y escribe un mensaje que muestra la contraseña.

```
1 * Con "--initialize-insecure", (ya sea con o sin --initialize porque --initialize-insecure implica --initialize), el servidor no genera una contraseña ni la marca caducada, y escribe un mensaje de advertencia:
```

```
1 #initialice
2 [Warning] A temporary password is generated for root@localhost:
3 iTag"AfrH5ej
```

```
1 #initialize-insecure
2 [Warning] root@localhost is created with an empty password ! Please
3 consider switching off the --initialize-insecure option.
```

* El servidor completa las tablas de ayuda del lado del servidor que se utilizan para la declaración HELP * Si se proporcionó la variable `init_file` del sistema para nombrar un archivo de sentencias SQL, el servidor ejecuta las sentencias en el archivo. Esta opción le permite realizar secuencias de arranque personalizadas. * Por último se cierra el servidor

Asignación de la contraseña root posterior a la inicialización

Estos son los pasos a seguir: * Iniciar el servidor * Conectarse al servidor

```
1 mysql -u root -p
```

* Luego, cuando se le solicite la contraseña, ingrese la contraseña aleatoria que generó el servidor durante la secuencia de inicialización:

```
1 Enter password: (enter the random root password here)
```

* Si inició con `--initialize-insecure` conectarse sin contraseña. Después de conectarse use **ALTER USER** para generar una nueva contraseña.

```
1 mysql -u root --skip-password
2 ...
3 mysql> ALTER USER 'root'@'localhost' IDENTIFIED BY 'root-password';
```

2.7.2 Iniciar el servidor

Para iniciar el servidor:

```
1 terminal> systemctl start mysqld
```

ó si la instalación incluye *mysqld_safe*

```
1 terminal> bin/mysqld_safe --user=mysql &
```

Es importante que el servidor MySQL se ejecute con una cuenta root de inicio de sesión sin privilegios (sin). Para garantizar esto, ejecute *mysqld_safe* como root e incluya la opción *--user* como se muestra. De lo contrario, debe ejecutar el programa mientras está conectado como mysql, en cuyo caso puede omitir la opción *--user* del comando.

2.7.3 Probar el servidor

Después de inicializar el directorio de datos y de haber iniciado el servidor, realizar algunas pruebas sencillas para asegurarse de que funciona satisfactoriamente. Esta sección asume que su ubicación actual es el directorio de instalación de MySQL y que tiene un subdirectorio bin que contiene los programas de MySQL usados aquí. Si eso no es cierto, ajuste los nombres de la ruta de comando en consecuencia.

Alternativamente, agregue el directorio bin a su configuración de variable de entorno PATH. Eso permite que su terminal (intérprete de comandos) encuentre programas MySQL correctamente, de modo que pueda ejecutar un programa escribiendo solo su nombre, no su nombre de ruta.

Utilice **mysqladmin** para verificar que el servidor se esté ejecutando. Los siguientes comandos proporcionan pruebas simples para verificar si el servidor está funcionando y respondiendo a las conexiones:

```
1 terminal> bin/mysqladmin version
2 terminal> bin/mysqladmin variables
```

Si no puedes conectarte al servidor, especifica una opción *-u root* para conectarte. Si ya ha asignado una contraseña para la root cuenta, también deberá especificar *-p*. Por ejemplo:

```
1 terminal> bin/mysqladmin -u root -p version
2 Enter password: (enter root password here)
```

La salida debería ser similar a:

```
1 terminal> bin/mysqladmin version
2 mysqladmin Ver 14.12 Distrib 8.0.26, for pc-linux-gnu on i686
3 ...
4
5 Server version      8.0.26
6 Protocol version    10
7 Connection          Localhost via UNIX socket
8 UNIX socket         /var/lib/mysql/mysql.sock
9 Uptime:              14 days 5 hours 5 min 21 sec
10
11 Threads: 1 Questions: 366 Slow queries: 0
12 Opens: 0 Flush tables: 1 Open tables: 19
13 Queries per second avg: 0.000
```

Verifique que puede apagar el servidor (incluya una opción *-p* si la cuenta root ya tiene una contraseña):

```
1 terminal> bin/mysqladmin -u root shutdown
```

Verifique que pueda iniciar el servidor nuevamente. Haga esto usando *mysqld_safe* o invocando *mysqld* directamente. Por ejemplo:

```
1 terminal> bin/mysqld_safe --user=mysql &
```

Utilice *mysqlshow* para ver qué bases de datos existen:

```

1 terminal> bin/mysqlshow
2 +-----+
3 | Databases |
4 +-----+
5 | information_schema |
6 | mysql |
7 | performance_schema |
8 | sys |
9 +-----+
```

2.7.4 Asegurar la cuenta inicial de MySQL

Esta sección describe cómo asignar una contraseña a la cuenta root inicial creada durante el procedimiento de instalación de MySQL, si aún no lo has hecho.

Advertencia: * En todas las plataformas, la distribución de MySQL incluye *mysql_secure_installation*, una utilidad de línea de comandos que automatiza gran parte del proceso de asegurar una instalación de MySQL.

Es posible que ya se haya asignado una contraseña a la cuenta inicial en las siguientes circunstancias: * Las instalaciones que utilizan paquetes Debian le dan la opción de asignar una contraseña. * Para la inicialización del directorio de datos realizada manualmente usando *mysqld --initialize*, *mysqld* genera una contraseña aleatoria inicial, la marca como vencida y la escribe en el registro de errores del servidor.

La tabla *mysql.user* de concesión define la cuenta de usuario inicial de MySQL y sus privilegios de acceso. La instalación de MySQL crea solo una cuenta '*root*'@'*localhost*' de **superusuario** que tiene todos los privilegios y puede hacer cualquier cosa. Si la cuenta root tiene una contraseña vacía, la instalación de MySQL está desprotegida: cualquiera puede conectarse al servidor MySQL root sin una contraseña y recibir todos los privilegios.

Para asignar una contraseña para la cuenta root MySQL inicial, utilice el siguiente procedimiento. Reemplace **root-password** en los ejemplos con la contraseña que desea usar.

Si la cuenta root existe con una contraseña aleatoria inicial que ha caducado, conéctate al servidor usando root esa contraseña, luego elige una nueva contraseña.

```

1 terminal> mysql -u root -p
2 Enter password: (enter the random root password here)
3 Elija una nueva contraseña para reemplazar la contraseña aleatoria:
4 ...
5 ...
6 mysql> ALTER USER 'root'@'localhost' IDENTIFIED BY 'root-password';
```

Si la cuenta root existe pero no tiene contraseña, conéctate al servidor root sin usar contraseña, luego asigna una contraseña. Este es el caso si inicializó el directorio de datos usando **mysqld --initialize-insecure**.

Conéctate al servidor root sin usar contraseña:

```

1 terminal> mysql -u root --skip-password
2 Asignar una contraseña:
3 ...
4 mysql> ALTER USER 'root'@'localhost' IDENTIFIED BY 'root-password';
```

2.8 Actualización de mysql

2.8.1 Rutas de actualización

- Se admite la actualización de MySQL 5.7 a 8.0. Sin embargo, la actualización solo se admite entre las versiones de disponibilidad general (GA). Para MySQL 8.0, es necesario que actualice desde una versión de MySQL 5.7 GA (5.7.9 o superior). No se admiten actualizaciones de versiones no GA de MySQL 5.7.
- Se recomienda actualizar a la última versión antes de actualizar a la siguiente versión. Por ejemplo, actualice a la última versión de MySQL 5.7 antes de actualizar a MySQL 8.0.
- No se admite la actualización que omite versiones. Por ejemplo, no se admite la actualización directa de MySQL 5.6 a 8.0.

- Una vez que una serie de lanzamientos alcanza el estado de Disponibilidad general (GA), se admite la actualización dentro de la serie de lanzamientos (de una versión GA a otra versión GA). Por ejemplo, actualizar desde MySQL 8.0.x a 8.0.y está apoyado. (No se admite la actualización que involucre versiones de estado de desarrollo que no sean de GA). También se admite la omisión de una versión. Por ejemplo, actualizar desde MySQL 8.0.x a 8.0.z está apoyado. MySQL 8.0.11 es la primera versión de estado GA dentro de la serie de versiones MySQL 8.0.

2.8.2 Qué actualiza el proceso de actualización de MySQL

La instalación de una nueva versión de MySQL puede requerir la actualización de estas partes de la instalación existente: * Paso 1: Actualización del diccionario de datos * Paso 2: Actualización del servidor

La siguiente lista muestra los comandos de actualización anteriores a MySQL 8.0.16 y los comandos equivalentes para MySQL 8.0.16 y versiones posteriores:

- Realice una actualización normal (pasos 1 y 2 según sea necesario):
- Antes de MySQL 8.0.16: [**mysqld**] seguido de [**mysql_upgrade**]
- A partir de MySQL 8.0.16: [**mysqld**]
- Realice solo el paso 1 según sea necesario:
- Antes de MySQL 8.0.16: no es posible realizar todas las tareas de actualización descritas en el paso 1 excluyendo las descritas en el paso 2. Sin embargo, puede evitar actualizar los esquemas de usuario y el esquema `sys` usando [**mysqld**] seguido de [**mysql_upgrade**] con las opciones [`--upgrade-system-tables`] y [`--skip-sys-schema`].
- A partir de MySQL 8.0.16: [**mysqld --upgrade = MINIMAL**]
- Realice el paso 1 según sea necesario y fuerce el paso 2:
- Antes de MySQL 8.0.16: [**mysqld**] seguido de [**mysql_upgrade --force**]
- A partir de MySQL 8.0.16: [**mysqld --upgrade = FORCE**]

Actualización con el repositorio APT de MySQL

- Antes de realizar cualquier actualización a MySQL, siga cuidadosamente las instrucciones en [Actualización de MySQL]. Entre otras instrucciones discutidas allí, *es especialmente importante hacer una copia de seguridad de su base de datos antes de la actualización*.
- Las siguientes instrucciones asumen que MySQL se ha instalado en su sistema usando el repositorio APT de MySQL; si ese no es el caso, siga las instrucciones que se dan en [Reemplazo de una distribución nativa de MySQL usando el repositorio APT de MySQL] o [Reemplazo de un servidor MySQL instalado por una descarga directa del paquete deb]. También tenga en cuenta que no puede usar el repositorio APT de MySQL para actualizar una distribución de MySQL que haya instalado desde un repositorio de software no nativo (por ejemplo, de MariaDB o Percona).
- Asegúrese de que tiene el repositorio APT de MySQL en la lista de repositorios de su sistema
- Asegurarte de que tienes la información del paquete más actualizado

```
1 terminal> sudo apt-get update
```

- Utilice el comando:

```
1 terminal> apt-get install mysql-server
```

4. Para acutlizar cualquier otro paquete:

```
1 terminal> apt-get install package-name
```

5. Para ver los nombres de los paquetes intalados

```
1 terminal> dpkg -l | grep mysql | grep ii
```

6. El servidor se reiniciará después de una actualización. Antes de MySQL 8.0.16, ejecute `mysql_upgrade` después de que el servidor se reinicie para verificar y posiblemente resolver cualquier incompatibilidad entre los datos antiguos y el software actualizado. A partir de MySQL 8.0.16, este paso no es necesario, ya que el servidor realiza todas las tareas previamente manejadas por `mysql_upgrade`.

2.9 Uso de imagen Docker para MySql

2.9.1 Descargar la imagen de Docker

Para descargar la imagen de Docker de MySQL, ejecute el siguiente comando:

```
1 terminal> docker pull mysql/mysql-server:tag
```

Donde **tag** es la versión de la imagen de MySQL que desea descargar. Por ejemplo, para descargar la imagen de MySQL 8.0, ejecute el siguiente comando:

```
1 terminal> docker pull mysql/mysql-server:8.0
```

2.9.2 Iniciar un contenedor de Docker con MySQL

Para iniciar un contenedor de Docker con MySQL, ejecute el siguiente comando:

```
1 terminal> docker run --name=mysql-container -e MYSQL_ROOT_PASSWORD=root-password -d mysql/mysql-server:tag
```

Donde **mysql-container** es el nombre del contenedor, **root-password** es la contraseña de root de MySQL y **tag** es la versión de la imagen de MySQL que desea utilizar. Por ejemplo, para iniciar un contenedor de Docker con MySQL 8.0 y la contraseña de root **root**, ejecute el siguiente comando:

```
1 terminal> docker run --name=mysql-container -e MYSQL_ROOT_PASSWORD=root -d mysql/mysql-server:8.0
```

2.9.3 Conectarse al contenedor de Docker con MySQL

Para conectarse al contenedor de Docker con MySQL, ejecute el siguiente comando:

```
1 terminal> docker exec -it mysql-container mysql -uroot -p
```

Donde **mysql-container** es el nombre del contenedor. Se le pedirá que ingrese la contraseña de root de MySQL. Una vez que haya ingresado la contraseña, se conectará al servidor MySQL en el contenedor de Docker.

2.9.4 Detener y eliminar el contenedor de Docker con MySQL

Para detener el contenedor de Docker con MySQL, ejecute el siguiente comando:

```
1 terminal> docker stop mysql-container
```

Para eliminar el contenedor de Docker con MySQL, ejecute el siguiente comando:

```
1 terminal> docker rm mysql-container
```

2.9.5 Persistencia de datos en contenedores de Docker

Para persistir los datos en un contenedor de Docker con MySQL, puede montar un volumen de Docker en el contenedor. Para hacer esto, agregue la opción **-v** al comando **docker run**. Por ejemplo, para montar un volumen de Docker en el contenedor de MySQL, ejecute el siguiente comando:

```
1 terminal> docker run --name=mysql-container -e MYSQL_ROOT_PASSWORD=root -v /path/to/mysql-data:/var/lib/mysql -d mysql/mysql-server:8.0
```

Donde **/path/to/mysql-data** es la ruta en su sistema host donde desea almacenar los datos de MySQL. Esto montará el volumen en el directorio **/var/lib/mysql** en el contenedor de Docker, lo que permitirá que los datos de MySQL persistan incluso si el contenedor se detiene o se elimina.

3. Instalación MongoDB

3.1 Introducción

MongoDB es una base de datos NoSQL de código abierto, muy popular y ampliamente utilizada en aplicaciones modernas para almacenar datos en documentos JSON. MongoDB es una base de datos NoSQL de alto rendimiento y escalable que se puede utilizar para almacenar datos de cualquier tipo sin tener que definir la estructura de los datos de antemano.

3.2 Instalación utilizando binarios

Para instalar MongoDB en un sistema Linux, primero debemos descargar el archivo binario de MongoDB desde el sitio web oficial de MongoDB. A continuación, descomprimos el archivo binario y movemos el directorio de MongoDB a la ubicación deseada.

```
1 $ wget https://fastdl.mongodb.org/linux/mongodb-linux-x86_64-4.4.6.tgz
2 $ tar -xvzf mongodb-linux-x86_64-4.4.6.tgz
3 $ mv mongodb-linux-x86_64-4.4.6 /usr/local/mongodb
```

3.3 Instalación utilizando el administrador de paquetes

MongoDB también se puede instalar utilizando el administrador de paquetes de la distribución de Linux. A continuación se muestra cómo instalar MongoDB en Ubuntu utilizando el administrador de paquetes apt.

```
1 $ sudo apt-get update
2 $ sudo apt-get install mongodb
```

3.3.1 directorios de MongoDB

MongoDB utiliza varios directorios para almacenar datos, registros y archivos de configuración. A continuación se muestra una lista de los directorios utilizados por MongoDB:

Directorio	Descripción
/data/db	Directorio predeterminado para los datos de MongoDB
/var/log/mongodb	Directorio de registros de MongoDB
/etc/mongod.conf	Archivo de configuración de MongoDB
/usr/local/mongodb	Directorio de instalación de MongoDB

3.4 Iniciar el servicio de MongoDB

Para iniciar el servicio de MongoDB, primero debemos crear el directorio de datos de MongoDB y luego iniciar el servicio de MongoDB utilizando el comando mongod.

```
1 $ mkdir -p /data/db
2 $ /usr/local/mongodb/bin/mongod
```

También podemos iniciar el servicio de MongoDB utilizando el administrador de servicios systemctl.

```
1 $ sudo systemctl start mongod
```


3.5 Detener el servicio de MongoDB

Podemos detener el servicio de MongoDB utilizando el comando `mongod --shutdown`.

```
1 $ /usr/local/mongodb/bin/mongod --shutdown
```

Mediante `systemctl`

```
1 $ sudo systemctl stop mongod
```

3.6 Conexión a MongoDB

Para conectarse a MongoDB, primero debemos iniciar el shell de MongoDB utilizando el comando `mongo`. A continuación, podemos conectarnos a la base de datos de MongoDB utilizando el comando `use`.

```
1 $ /usr/local/mongodb/bin/mongo
2 > use testdb
```

3.7 Crear usuario administrador

Para crear un usuario administrador en MongoDB, primero debemos conectarnos a la base de datos de administración y luego crear un usuario administrador utilizando el comando `db.createUser()`.

```
1 $ /usr/local/mongodb/bin/mongo
2 > use admin
3 > db.createUser
4 (
5   {
6     user: "admin",
7     pwd: "admin123",
8     roles: [ { role: "userAdminAnyDatabase", db: "admin" } ]
9   }
10 )
```

4. Imagen de MongoDB en docker

Para ejecutar una instancia de MongoDB en un contenedor de Docker, primero debemos descargar la imagen oficial de MongoDB desde Docker Hub y luego ejecutar el contenedor de MongoDB utilizando el comando `docker run`.

```
1 $ docker pull mongo
2 $ docker run -d -p 27017:27017 --name mongod mongo
```

4.1 Persistencia de datos

Para persistir los datos de MongoDB en un contenedor de Docker, podemos montar un volumen de Docker en el directorio de datos de MongoDB utilizando la opción `-v`.

```
1 $ docker run -d -p 27017:27017 -v /data/db:/data/db --name mongod mongo
```

4.2 Conexión a MongoDB en Docker

Para conectarse a la instancia de MongoDB en un contenedor de Docker, primero debemos iniciar el shell de MongoDB utilizando el comando `docker exec` y luego conectarnos a la base de datos de MongoDB utilizando el comando `mongo`.

```
1 $ docker exec -it mongod mongo
2 > use testdb
```

5. Probar la conexión a MongoDB

Para probar la conexión a MongoDB, podemos crear una base de datos y una colección en MongoDB y luego insertar algunos documentos en la colección.

```
1 $ /usr/local/mongodb/bin/mongo
2 > use testdb
3 > db.createCollection("users")
4 > db.users.insert({name: "Alice", age: 30})
5 > db.users.insert({name: "Bob", age: 25})
6 > db.users.find()
```

6. Asegurar MongoDB

Para asegurar MongoDB, podemos habilitar la autenticación de MongoDB y crear usuarios con roles específicos en MongoDB.

```
1 $ /usr/local/mongodb/bin/mongo
2 > use admin
3 > db.createUser
4 (
5   {
6     user
7   : "admin",
8     pwd: "admin123",
9     roles: [ { role: "userAdminAnyDatabase", db: "admin" } ]
10  }
11 )
12 > db.auth("
13 admin", "admin123")
```