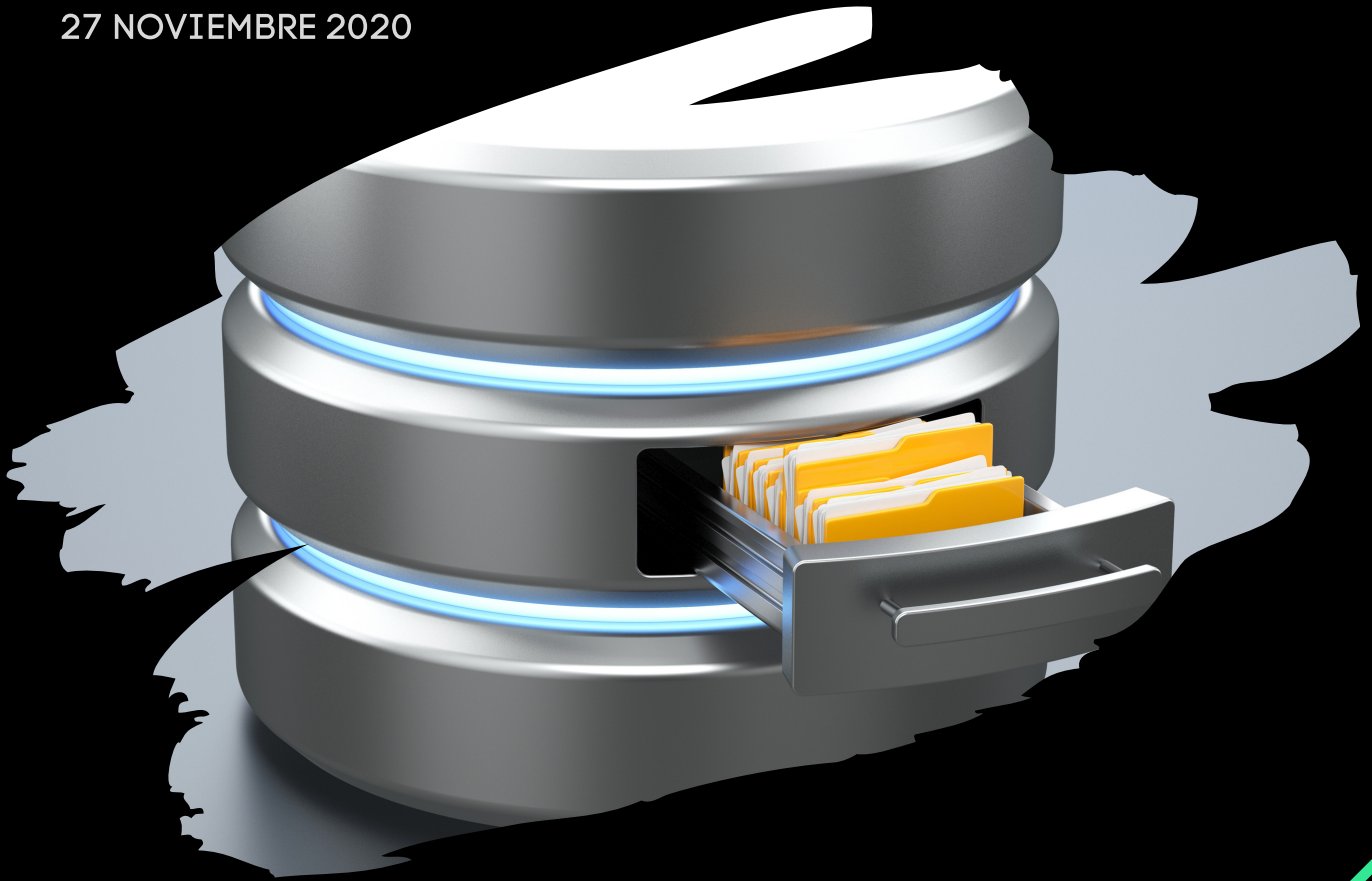


27 NOVIEMBRE 2020



AWS- UBUNTU

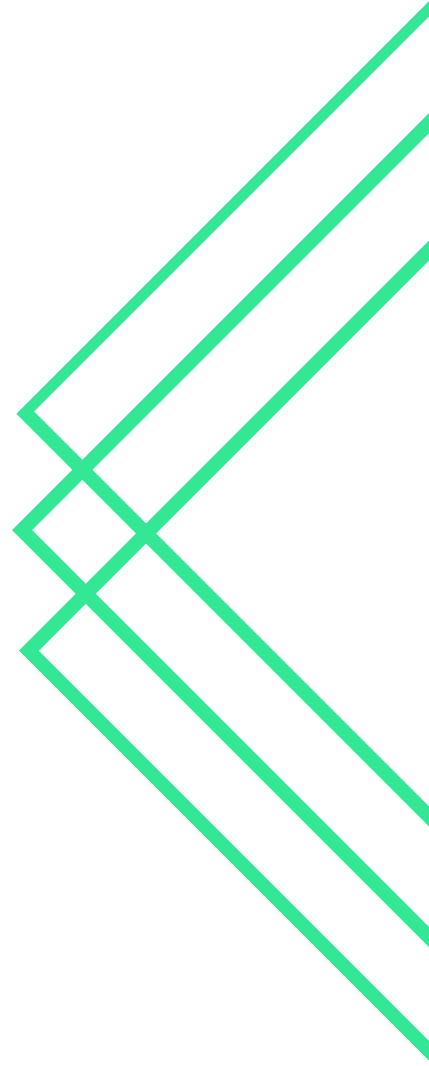
Hanna Siddhartha Lizarraga Ceballos

Objetivo

Se tiene como objetivo aprender a usar las instancias en amazon LightSail utilizando OS de Ubuntu y que en este por medio de docker podamos crear o importar base de datos.

Alcance

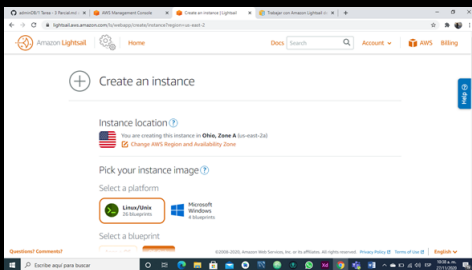
En este caso al instalar Ubuntu e integrar docker a este, se hará un clon de repositorio de git hub para acceder a nuestra base de datos y crear tablas y contenido en esta.



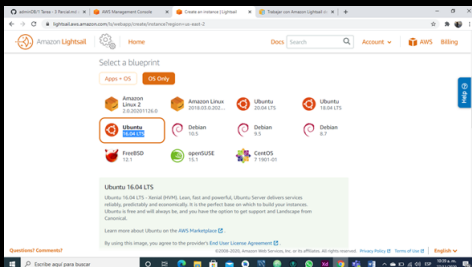
Como crear una instancia en Amazon LightSail

Primero ingresamos a la pagina de aws.amazon.com, seguido de esto iniciamos sesión para consola y seleccionamos Usuario raíz.

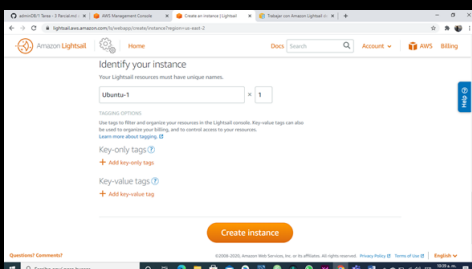
Una vez hecho esto damos click en Crear Instancia.



Seleccionamos Linux

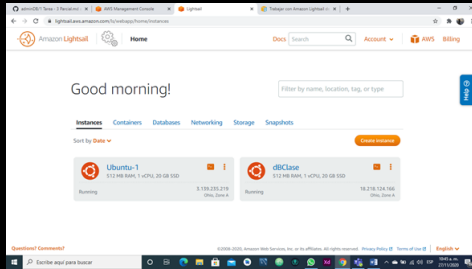


Seleccionamos la opción de OS y escogemos Ubuntu

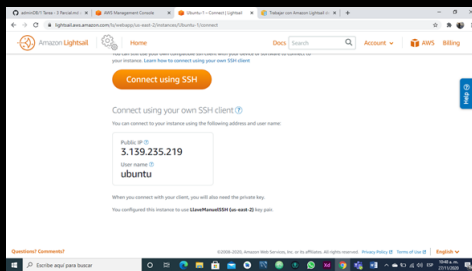


Creamos la instancia

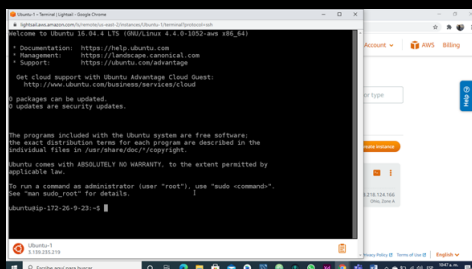
Como crear una instancia en Amazon LightSail



Al crear la instancia se vera así

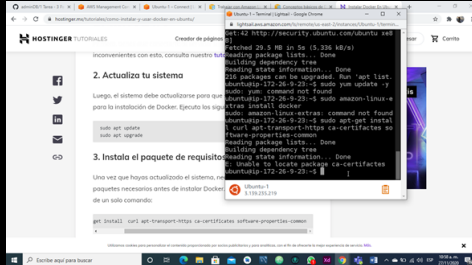


Al ingresar podemos observar nuestro IP el cual utilizaremos mas adelante.



Nos conectamos por SSH y actualizamos el sistema.

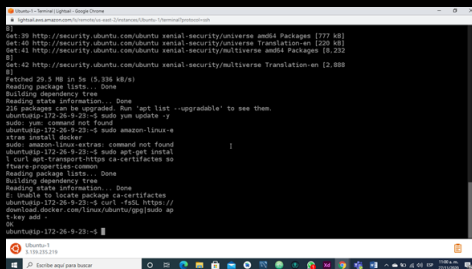
Instalando Docker en Ubuntu



Instalamos el paquete de requisitos previos

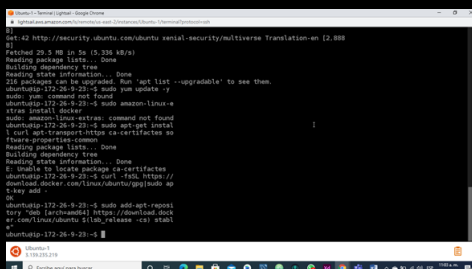
`sudo apt-get install curl apt-transport-https ca-certificates software-properties-common`

- `apt-transport-https`: permite que el administrador de paquetes transfiera datos a través de https
- `ca-certificates`: permite que el navegador web y el sistema verifiquen los certificados de seguridad
- `curl`: transfiere datos
- `software-properties-common`: agrega scripts para administrar el software



Agregamos los repositorios de Docker

`curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -`



Agregamos el repositorio

`sudo add-apt-repository \"deb [arch=amd64] https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable\"`

Instalando Docker en Ubuntu

```

Last login: Fri Nov 27 10:45:42 2020 from 52.55.24.90
To run a command as administrator (user "root"), use "sudo -i" command.
See "man sudo_root" for details.

ubuntu@172.26.9.29:~$ sudo apt update
Hit:1 http://us-east-2.ec2.archive.ubuntu.com/ubuntu xenial InRelease
Get:2 http://us-east-2.ec2.archive.ubuntu.com/ubuntu xenial-updates InRelease [109 kB]
Get:3 http://us-east-2.ec2.archive.ubuntu.com/ubuntu xenial-backports InRelease [107 kB]
Get:4 http://us-east-2.ec2.archive.ubuntu.com/ubuntu xenial/main Sources [108 kB]
Get:5 http://us-east-2.ec2.archive.ubuntu.com/ubuntu xenial/restricted Sources [4,096 B]
Get:6 http://us-east-2.ec2.archive.ubuntu.com/ubuntu xenial/universe Sources [7,726 kB]
Get:7 http://us-east-2.ec2.archive.ubuntu.com/ubuntu xenial/security InRelease [109 kB]
Get:8 http://us-east-2.ec2.archive.ubuntu.com/ubuntu xenial/multiverse Sources [179 kB]
Get:9 http://us-east-2.ec2.archive.ubuntu.com/ubuntu xenial/universe amd64 Packages [7,532 kB]
Get:10 http://us-east-2.ec2.archive.ubuntu.com/ubuntu xenial/multiverse Translation-en [4,354 kB]
Get:11 http://us-east-2.ec2.archive.ubuntu.com/ubuntu xenial/multiverse amd64 Packages [148 kB]
Get:12 http://us-east-2.ec2.archive.ubuntu.com/ubuntu xenial/multiverse Translation-en [108 B]
Get:13 http://us-east-2.ec2.archive.ubuntu.com/ubuntu xenial-updates/main Sources [529 kB]
Get:14 http://us-east-2.ec2.archive.ubuntu.com/ubuntu xenial-updates/restricted Sources [15,108 B]
Get:15 http://us-east-2.ec2.archive.ubuntu.com/ubuntu xenial-updates/universe Sources [436 kB]
Get:16 http://us-east-2.ec2.archive.ubuntu.com/ubuntu xenial-updates/multiverse Sources [11 kB]

```

Actualizamos la información del repositorio

```
sudo apt update
```

```

ubuntu@172.26.9.29:~$ apt-cache policy docker-ce
docker-ce:
  Installed: none
  Candidate: 19.03.13-3-ubuntu-xenial
  Version table:
   19.03.13-3-ubuntu-xenial 500
     500 https://download.docker.com/linux/ubuntu/xenial/stable amd64 Packages
     519.03.12-3-ubuntu-xenial 500
     500 https://download.docker.com/linux/ubuntu/xenial/stable amd64 Packages
     519.03.11-3-ubuntu-xenial 500
     500 https://download.docker.com/linux/ubuntu/xenial/stable amd64 Packages
     519.03.10-3-ubuntu-xenial 500
     500 https://download.docker.com/linux/ubuntu/xenial/stable amd64 Packages
     519.03.9-3-ubuntu-xenial 500
     500 https://download.docker.com/linux/ubuntu/xenial/stable amd64 Packages
     519.03.8-3-ubuntu-xenial 500
     500 https://download.docker.com/linux/ubuntu/xenial/stable amd64 Packages

```

Para saber si estamos instalando desde el repositorio de Docker en lugar del repositorio predeterminado de Ubuntu.

```
apt-cache policy docker-ce
```

```

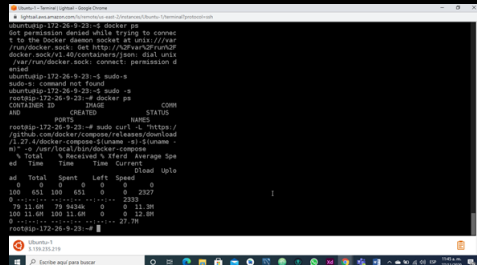
ubuntu@172.26.9.29:~$ sudo apt install docker-ce
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  aufs-tools cgroupfs-mount containerd.io docker-ce-cli libltdl7 libncursesw5
  libpam0g libseccomp2 libsystemd0 libudev1
Suggested packages:
  docker-ce-cli libltdl7 libncursesw5
The following NEW packages will be installed:
  aufs-tools cgroupfs-mount containerd.io
  docker-ce docker-ce-cli libltdl7 pigz
The following package will be upgraded:
  libseccomp2
1 upgraded, 7 newly installed, 0 to remove and 0 not installed.
Need to get 91.2 MB of archives.
After this operation, 343 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://us-east-2.ec2.archive.ubuntu.com/ubuntu xenial-updates/main amd64 containerd.io amd64 1.2.7-1 [28.6 kB]
Get:2 http://us-east-2.ec2.archive.ubuntu.com/ubuntu xenial-updates/main amd64 aufs-tools amd64 3.10-1 [12.7 kB]
Get:3 http://us-east-2.ec2.archive.ubuntu.com/ubuntu xenial-updates/main amd64 cgroupfs-mount amd64 1.0 [10.5 kB]
Get:4 http://us-east-2.ec2.archive.ubuntu.com/ubuntu xenial-updates/main amd64 docker-ce amd64 19.03.13-3 [10.8 kB]
Get:5 http://us-east-2.ec2.archive.ubuntu.com/ubuntu xenial-updates/main amd64 docker-ce-cli amd64 19.03.13-3 [10.8 kB]
Get:6 http://us-east-2.ec2.archive.ubuntu.com/ubuntu xenial-updates/main amd64 libltdl7 amd64 2.2.14-1 [12.7 kB]
Get:7 http://us-east-2.ec2.archive.ubuntu.com/ubuntu xenial-updates/main amd64 libseccomp2 amd64 2.2.0-4 [28.6 kB]
Fetched 91.2 MB in 10s (9,120 kB/s)

```

Instalando Docker en Ubuntu 18.04

```
sudo apt install docker-ce
```


Instalar Compose en sistemas Linux



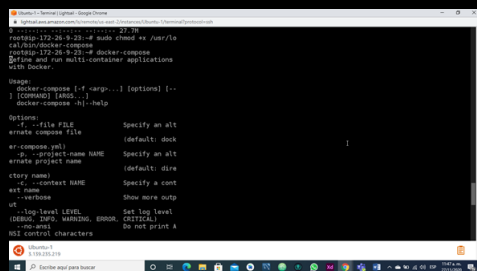
```

root@172-26-9-23:~# docker ps
set permission denied while trying to connect
to the Docker daemon socket at unix:///var/
/run/docker.sock: Get http://localhost/v1.27:
dial unix
/var/run/docker.sock: connect: permission d
enied
root@172-26-9-23:~# sudo -s
sudo:sudo not found
root@172-26-9-23:~# sudo -s
root@172-26-9-23:~# docker ps
CONTAINER ID        IMAGE               COMMAND
AND                CREATED            STATUS             COM
root@172-26-9-23:~# sudo curl -L "https://
github.com/docker/compose/releases/download/1.27.4/docker-compose.$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose
% Total    % Received % Flowed Average Spe
ad Time   Time   Time  Current
0 0 0 0 0 0 0
100 651 100 651 0 0 2397
8 11.6M 100 11.6M 0 0 11.3M
79 11.6M 79 94348 0 0 11.3M
100 11.6M 100 11.6M 0 0 12.2M
0 0 0 0 0 0 0
root@172-26-9-23:~#

```

Descargar la versión estable actual de Docker Compose

```
sudo curl -L "https://github.com/docker/compose/releases/download/1.27.4/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose
```



```

root@172-26-9-23:~# sudo -s
root@172-26-9-23:~# docker-compose
Get time and run multi-container applications with Docker.

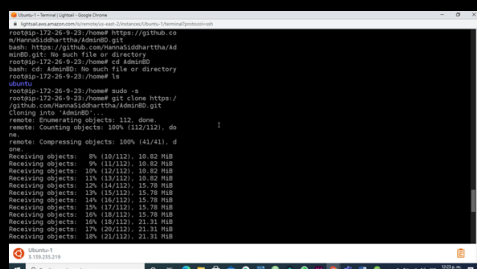
Usage:
  docker-compose [-f ...] [optional] [-]
  docker-compose -h|--help

Options:
  -f, --file FILE             Specify an alt
                             ername compose file
                             (default: dock
er-compose.yml)
  -p, --project-name NAME     Specify an alt
                             ername project name
                             (default: dire
ctory name)
  -c, --context NAME         Specify a cont
                             ext name
  --verbose                   Show more outp
ut
  --log-level LEVEL          Set log level
                             (debug, info, warning, error,
                             critical)
  --no-ansi                  Do not print A
NSI control characters

```

Aplicar permisos ejecutables al binario.

```
sudo chmod +x /usr/local/bin/docker-compose
```



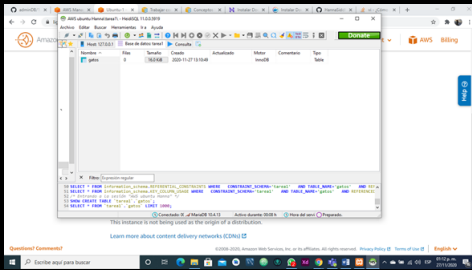
```

root@172-26-9-23:~# cd /root
root@172-26-9-23:~# git clone https://github.com/docker/compose.git
Cloning into 'compose'...
remote: Enumerating objects: 112, done.
remote: Counting objects: 100% (112/112), 4
KB
remote: Compressing objects: 100% (41/41), 4
KB
Receiving objects: 0% (0/112), 10.82 MiB
Receiving objects: 9% (10/112), 10.82 MiB
Receiving objects: 10% (12/112), 10.82 MiB
Receiving objects: 11% (13/112), 10.82 MiB
Receiving objects: 12% (14/112), 10.78 MiB
Receiving objects: 13% (15/112), 10.78 MiB
Receiving objects: 14% (16/112), 10.78 MiB
Receiving objects: 15% (17/112), 10.78 MiB
Receiving objects: 16% (18/112), 10.78 MiB
Receiving objects: 17% (19/112), 10.78 MiB
Receiving objects: 18% (20/112), 10.78 MiB
Receiving objects: 19% (21/112), 10.78 MiB

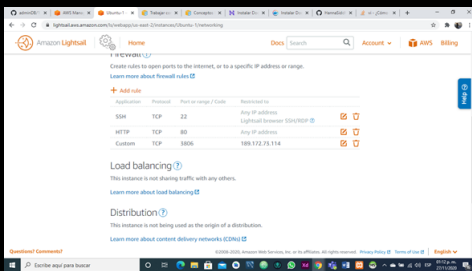
```

Clonamos nuestro repo.

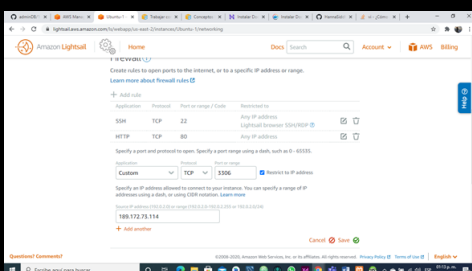
Contenido en nuestra base de datos y reglas y restricciones de firewall.



Al entrar a nuestra base de datos, creamos una tabla.



Después regresamos a nuestro panel y en Networking damos click y creamos una regla en firewall. También podemos agregar o eliminar por medio de puerto o IP el acceso a nuestra instancia.



Para crear una restricción como bloqueo de acceso podemos utilizar nuestra IP para que solo nosotros podamos ingresar.

Conclusión

Es una práctica muy fácil la cual es indispensable saber ya que facilita mucho el trabajo de los desarrolladores una de las muchas ventajas de Docker es que hacen que la implementación de software sea mucho más eficiente que antes. Gracias a esto, los desarrolladores no tendrán problemas para saber cómo se ejecutará su aplicación fuera del entorno de prueba.