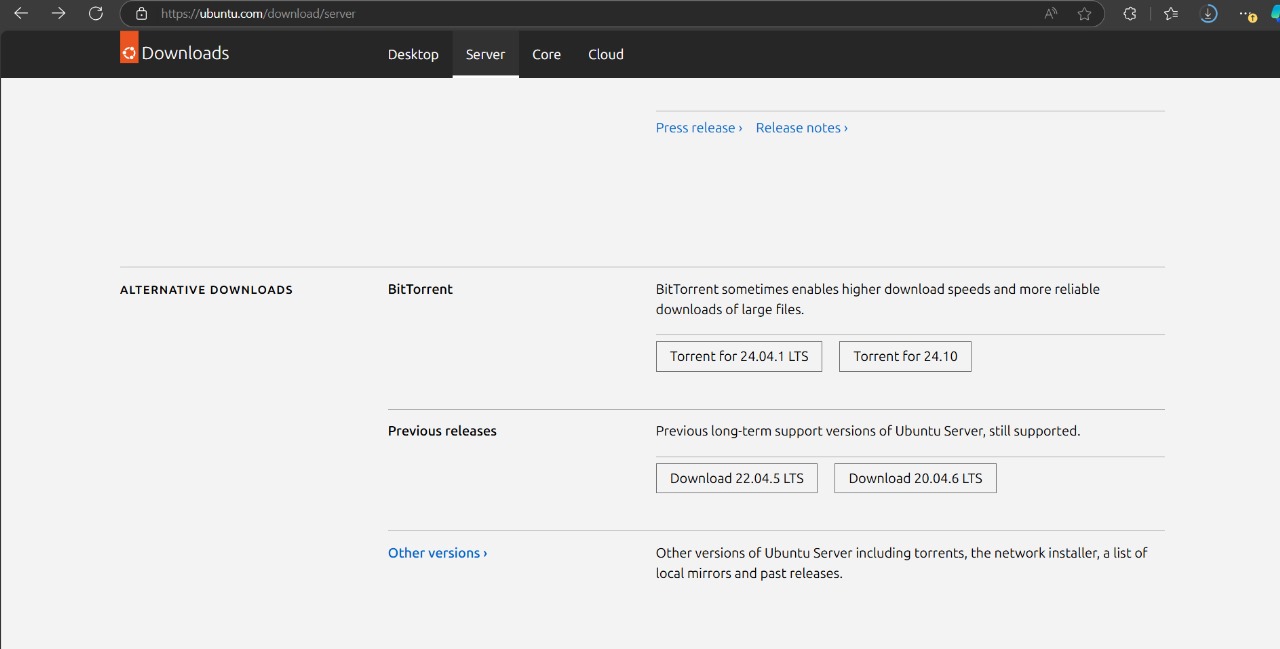
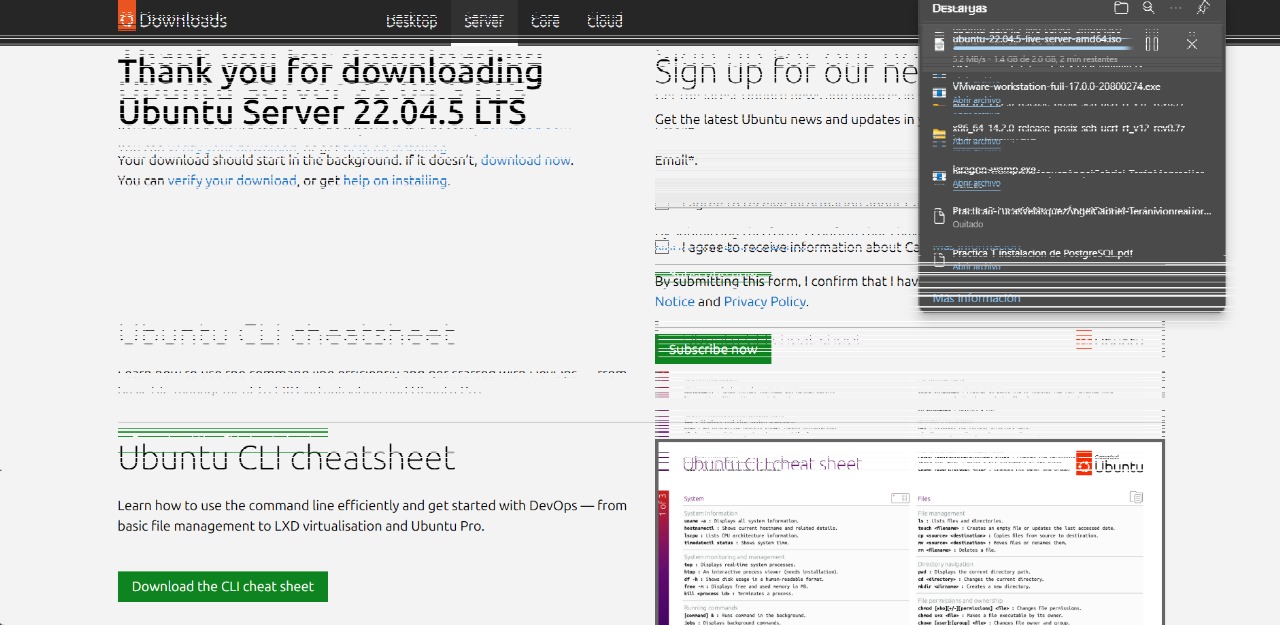
A jueves 24 de octubre de 2024.

Se inició la descarga de VMware Workstation, haciendo una cuenta en BROADCOM se descargó la versión 17.0.0.

Se inició la instalación con la prueba gratuita, de manera no licenciada, para poder trabajar con ello un determinado lapso de tiempo.

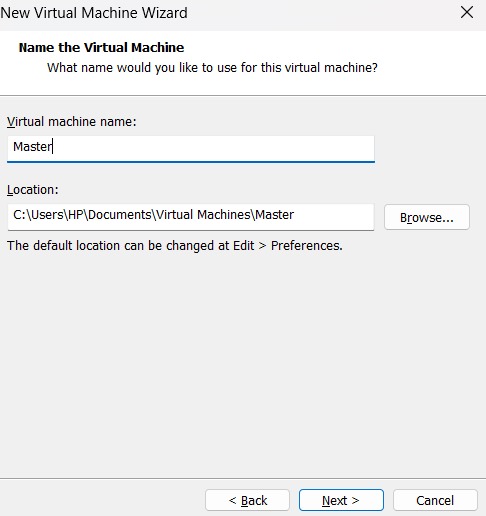
Se descargó el iso o también conocido como disco virtual del sistema operativo de Ubuntu versión 24.04.5.

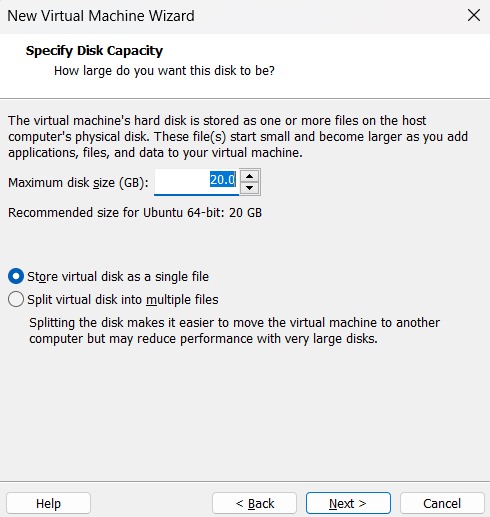


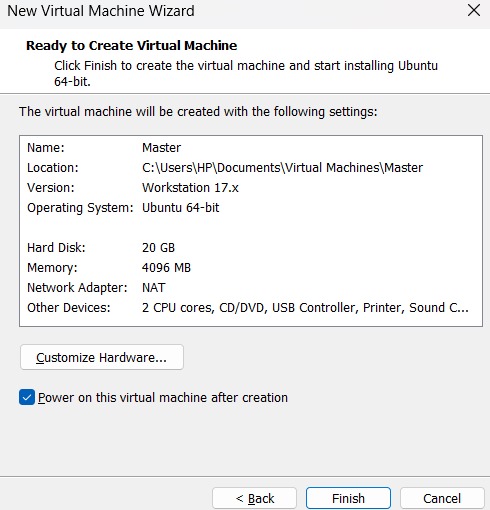


Se abre la ventana para crear una nueva maquina virtual y se insertó el iso que se acaba de descargar.

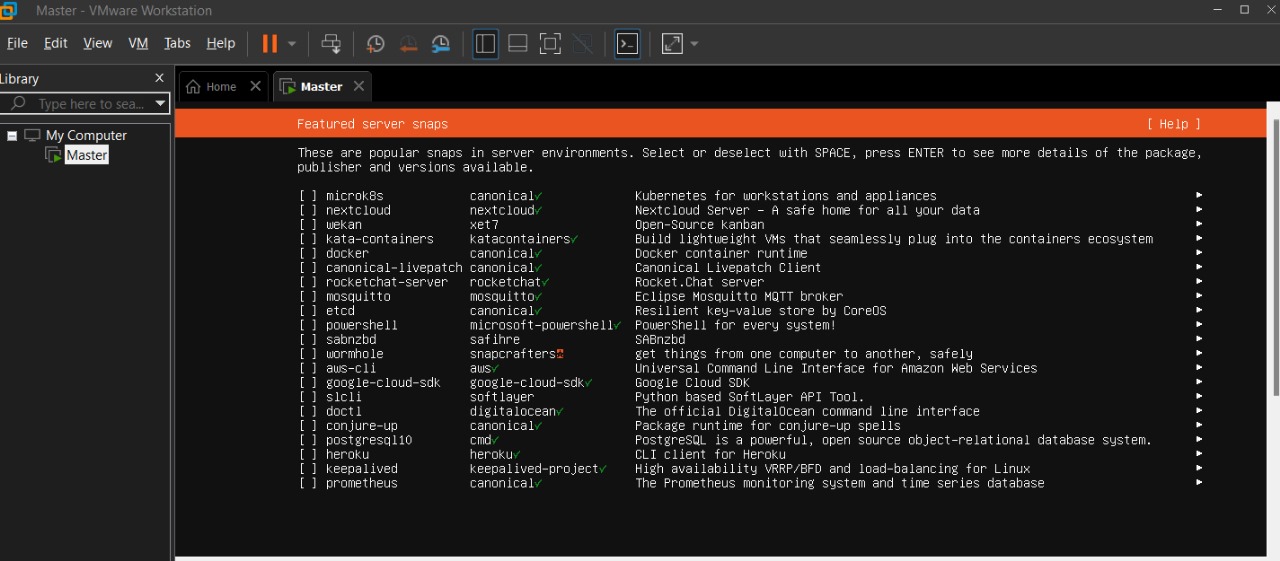
El nombre asignado para la máquina virtual fue Master, se le colocaron 20 GB de almacenamiento, 4 GB de RAM, y se dejaron las particiones por defecto.

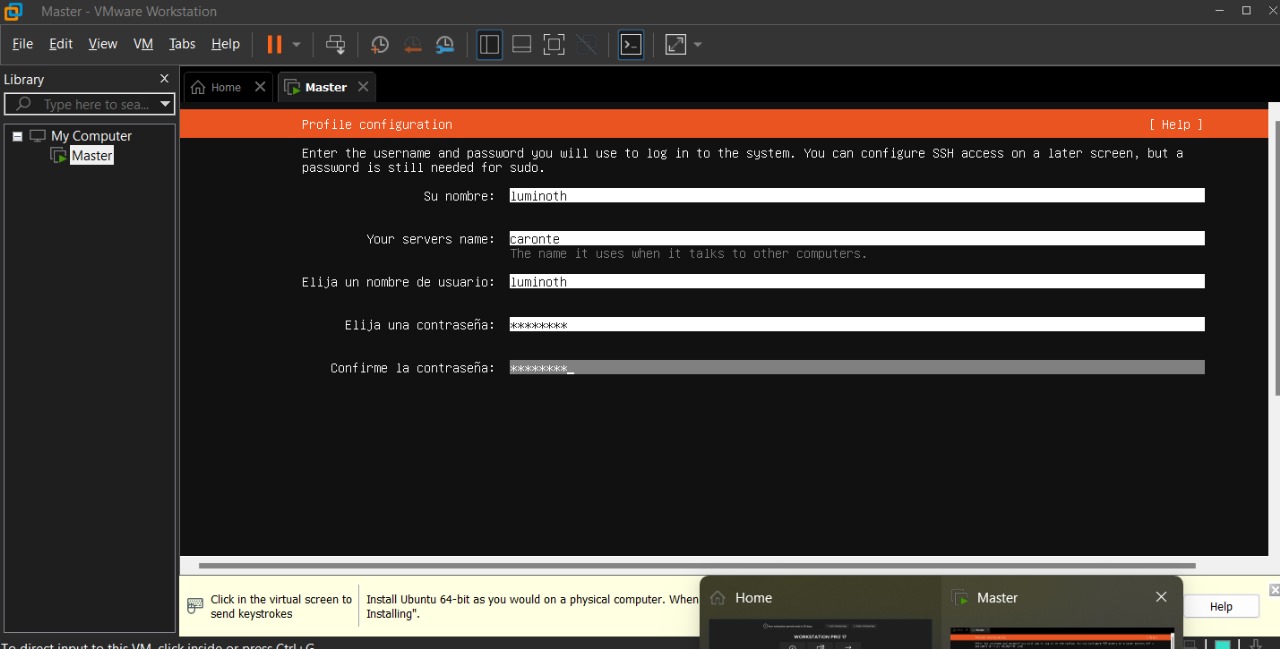




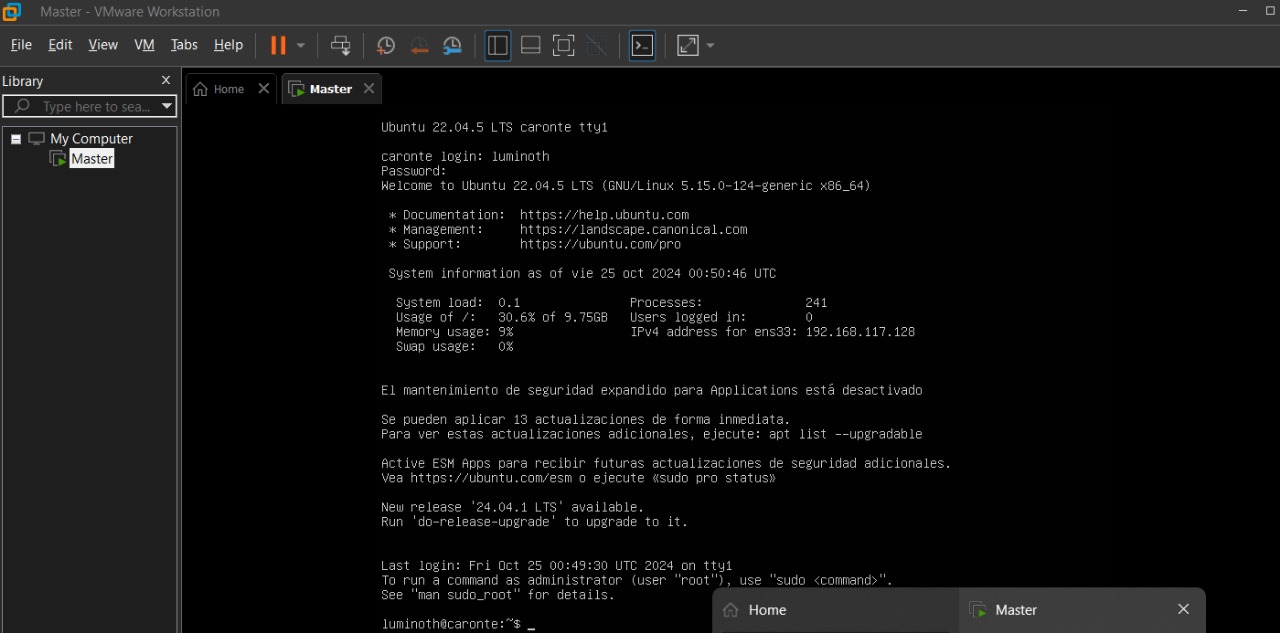


El nombre del usuario principal fue “luminoth”, tal y como se indico en el proyecto, y por contraseña se consideró al nombre del usuario; no se instaló ningún snap; se completa la instalación y se reinicia la máquina.

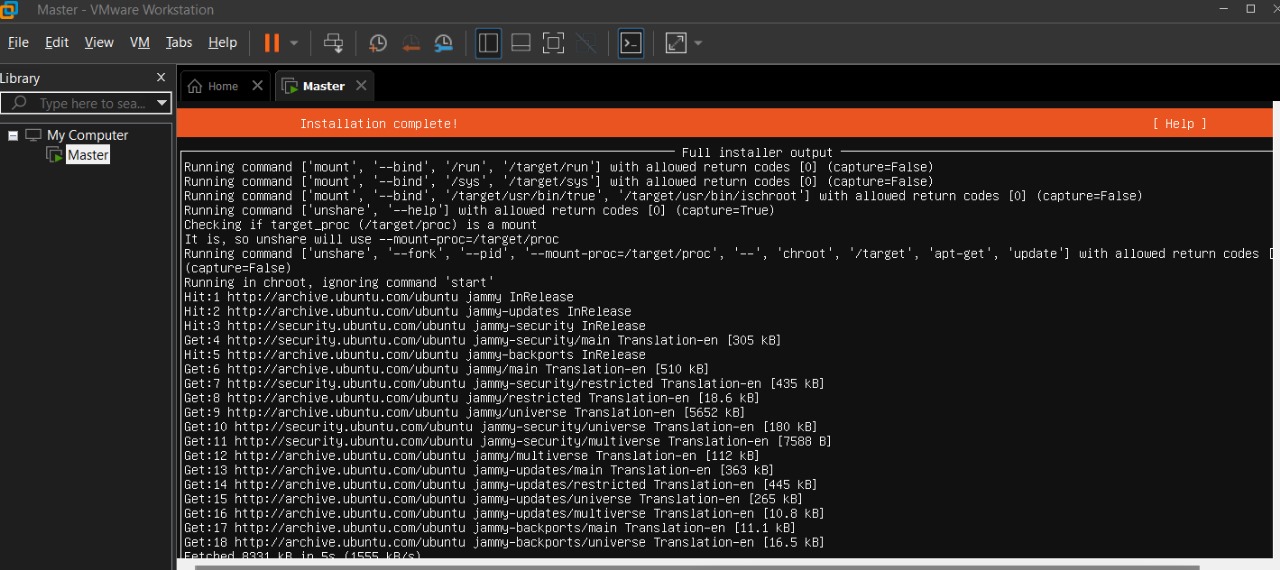




Se accedió al sistema como luminoth y colocamos la contraseña del usuario actual, y seleccionamos una contraseña para el super usuario; se instaló net-tolos para el manejo de ip.

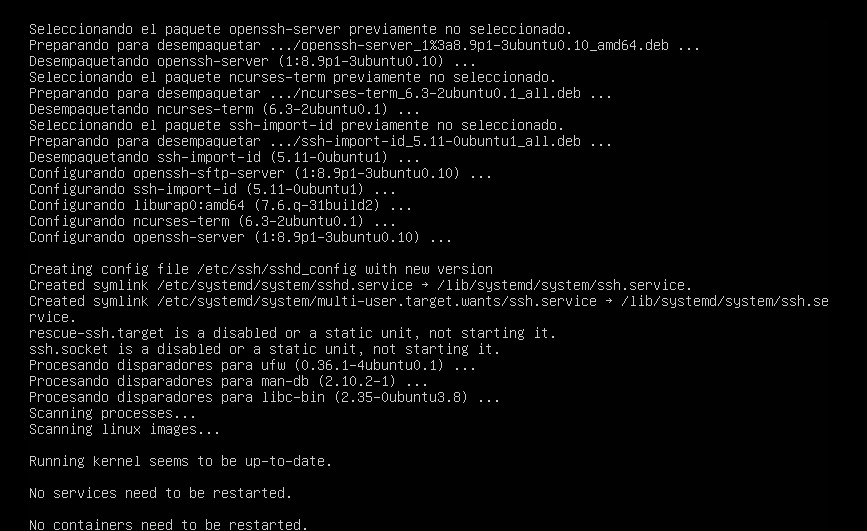


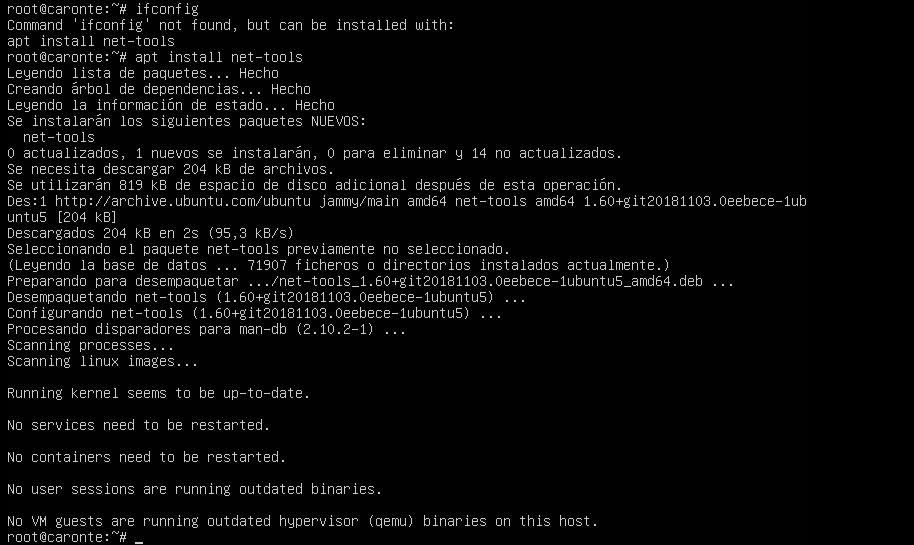
Se instaló ssh en la máquina con el comando ‘apt install ssh-server’





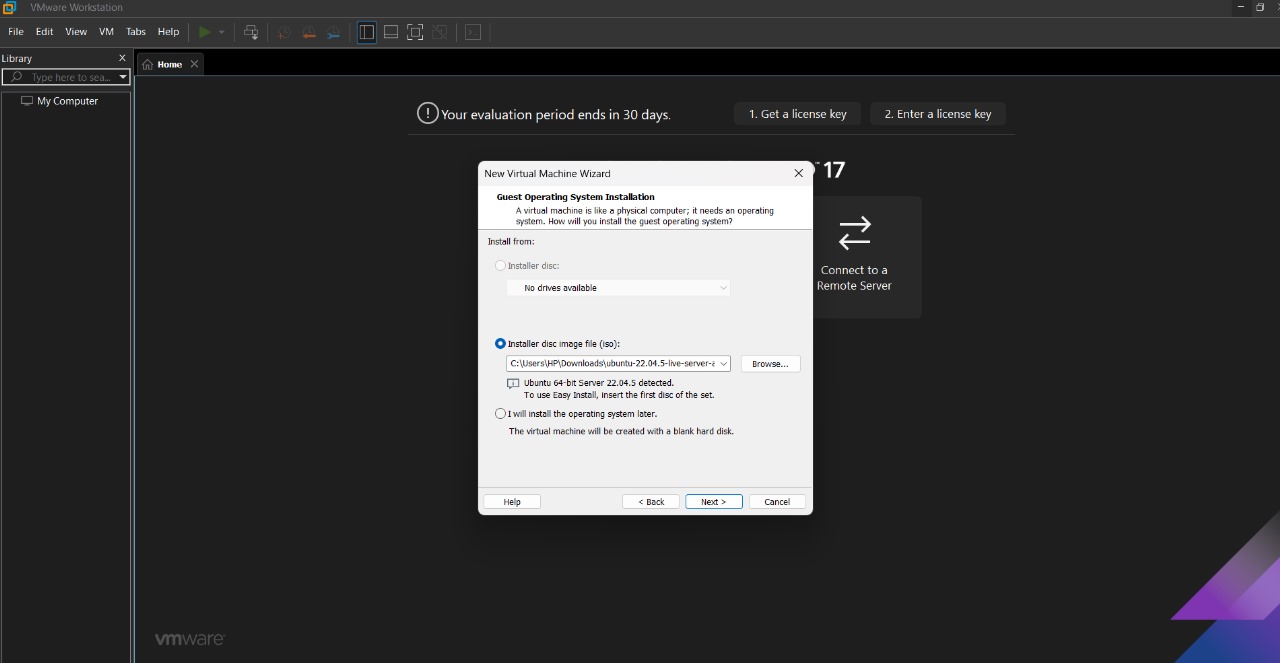
Seguido de ello se descargó un Bitvise SSH Client, en la sección de login, en el host se colocó la ip de la máquina virtual Ubuntu creada en VMware y en este caso la ip fue 192.168.117.128.





Viernes 25 de octubre de 2024

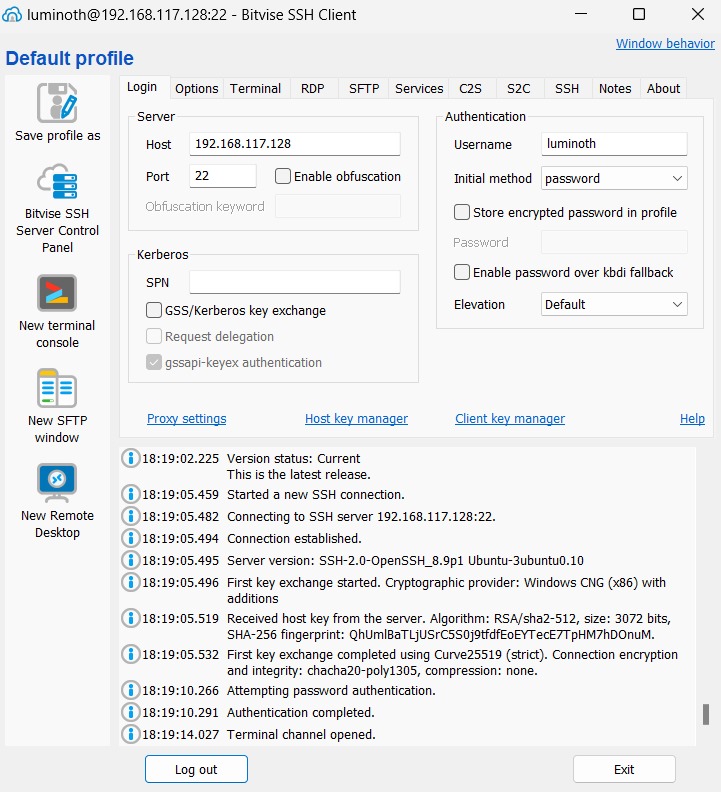
Se creó una nueva máquina virtual y se le colocó el iso que se acababa de descargar.



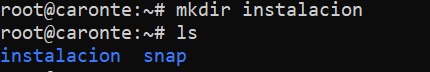
Modificamos el firewall para permitir conexiones ssh y se reviso que escuchara el puerto 22.



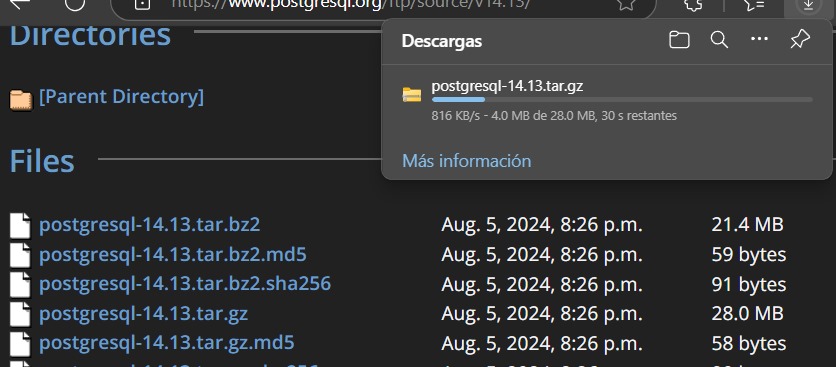
Para lograr la conexión con la máquina virtual se tuvo que dar de ata con luminoth, un usuario mortal, ya que Ubuntu no permite una conexión directa al super usuario desde el cliente de ssh.



Se creó el directorio “instalacion”

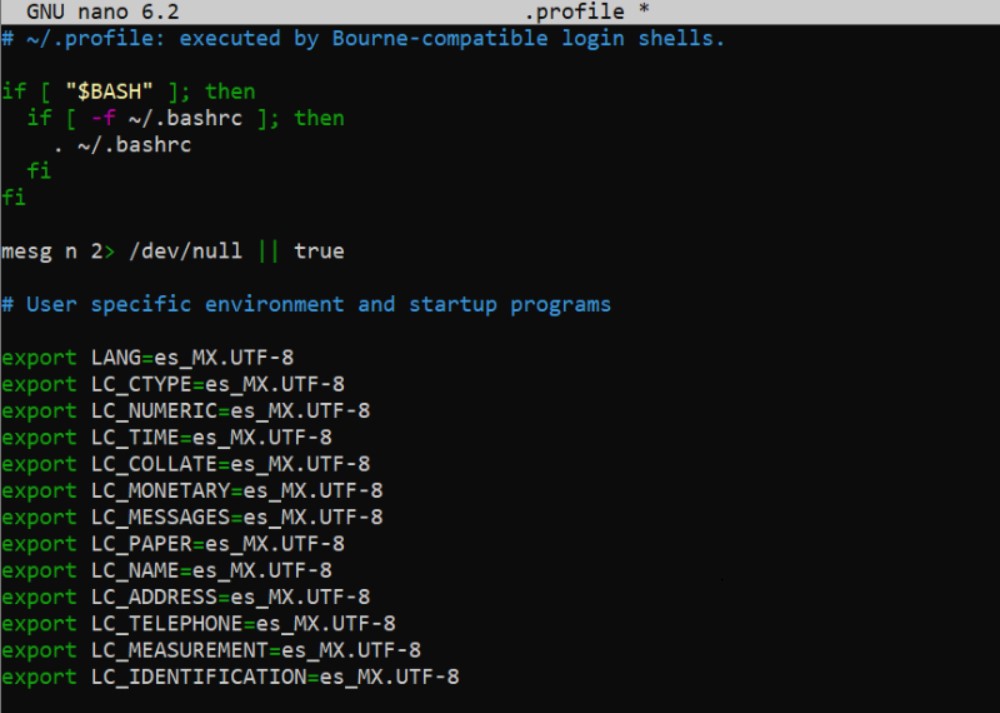


Se descargó el código fuente de postgres versión 14.13 empaquetado





Se modificaron las variables de entorno para colocar el idioma y caracteres mexicanos. No se modifica el LC\_ALL, es necesario modificar cada variable por separado.



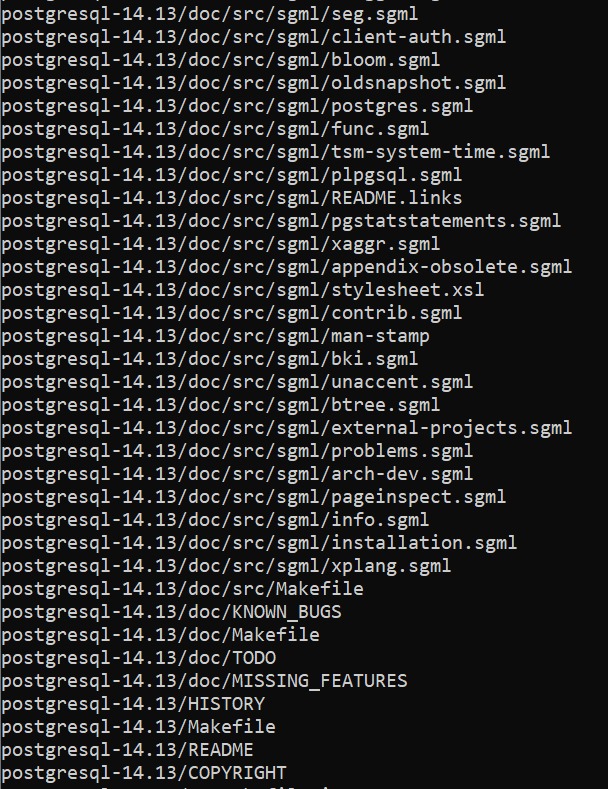
Se guardan los cambios de manera permanente



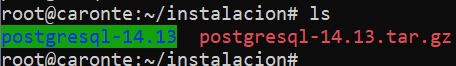
Se movió el paquete del código fuente de postgres al directorio “instalacion” que se acababa de crear.



Se hace el destareo y desempaquetamiento del postgres



El desempaquetado creó una carpeta llamada PostgreSQL-14.13, el cual es el de color verde.



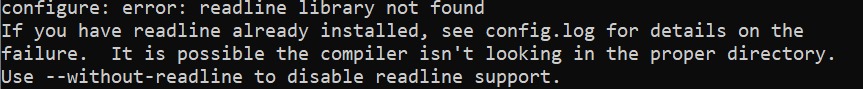
Se descargaron algunas herramientas de configuración, se recibieron algunas advertencias porque se cambió la variable de entorno a español UTF-8



Ahora se configura el entorno de compilación



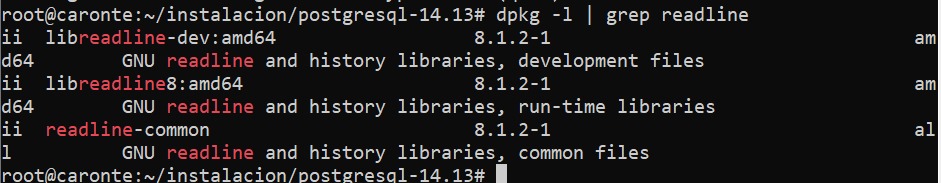
Se observa que no se tiene instalada la librería readline

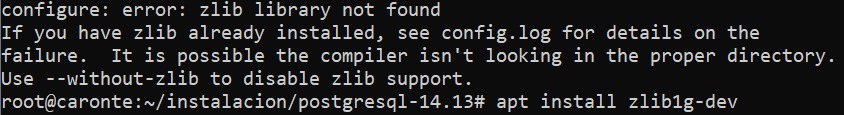


Se instala dicha librería



Verificamos que se haya instalado

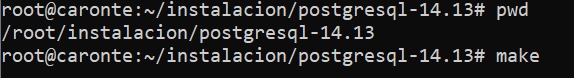




El desempaquetado creo una carpeta, la cual es la que se encuentra de color verde



Como parte de verificar la instalación utilizamos el comando pwd para asegurarnos de encontrarnos en dicho directorio.

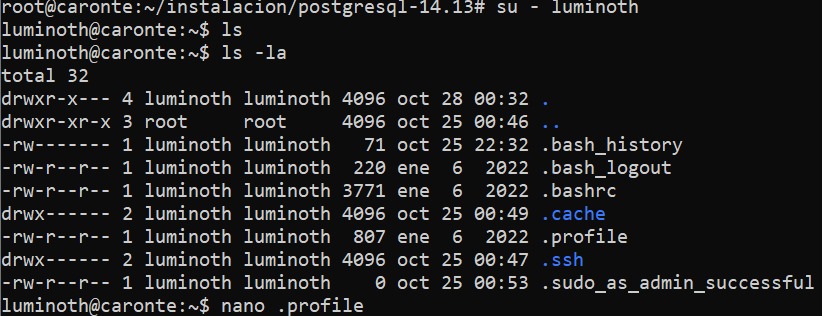


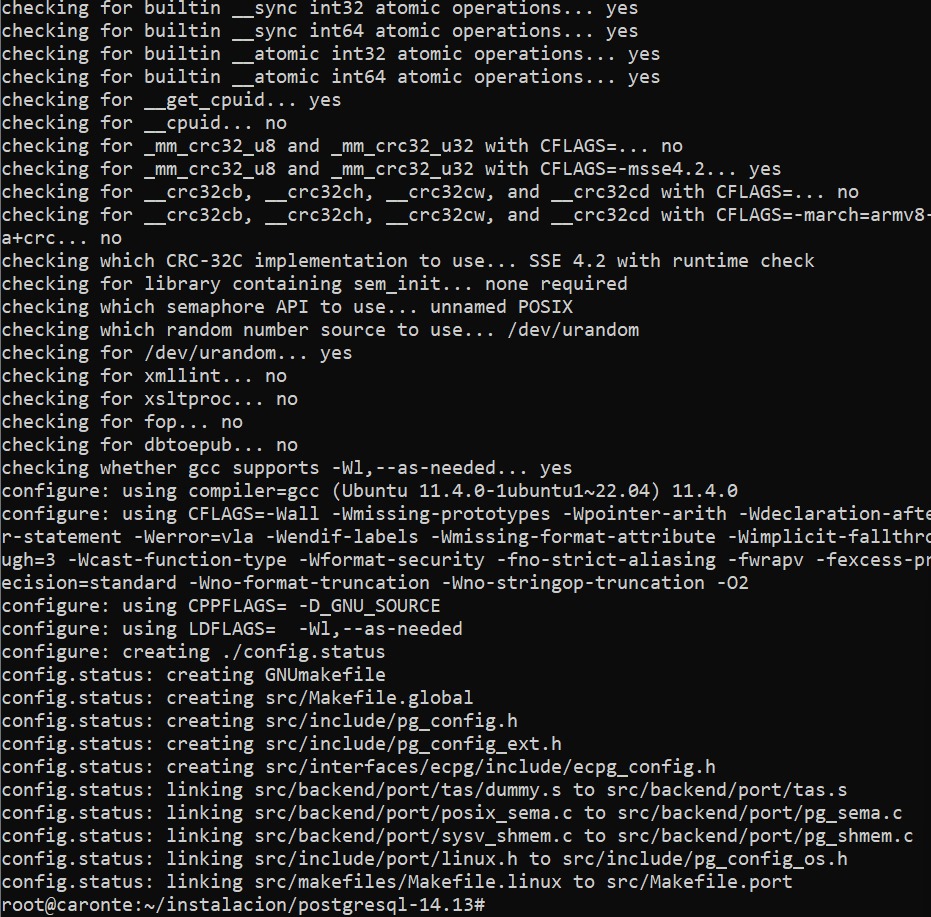
Creamos el directorio data



Después le dimos la propiedad de esqueleto a luminoth y revisamos que hayan sido cambiado correctamente de propietario. 

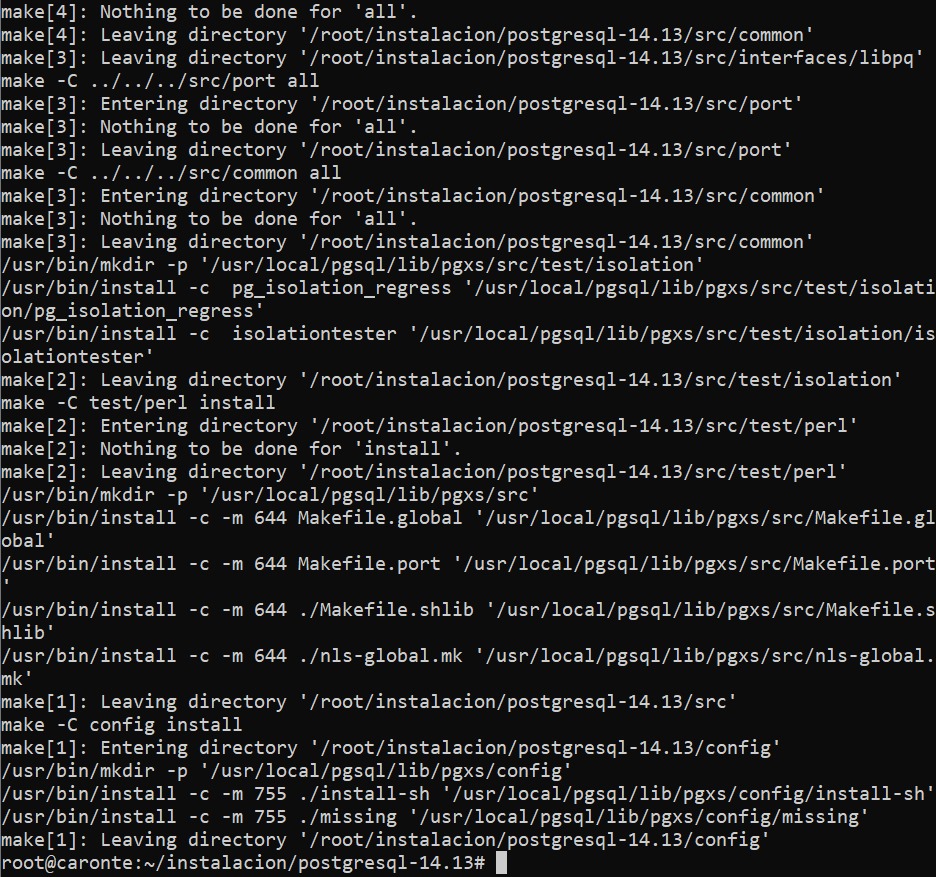
Y verificamos que el esqueleto le pertenezca a luminoth



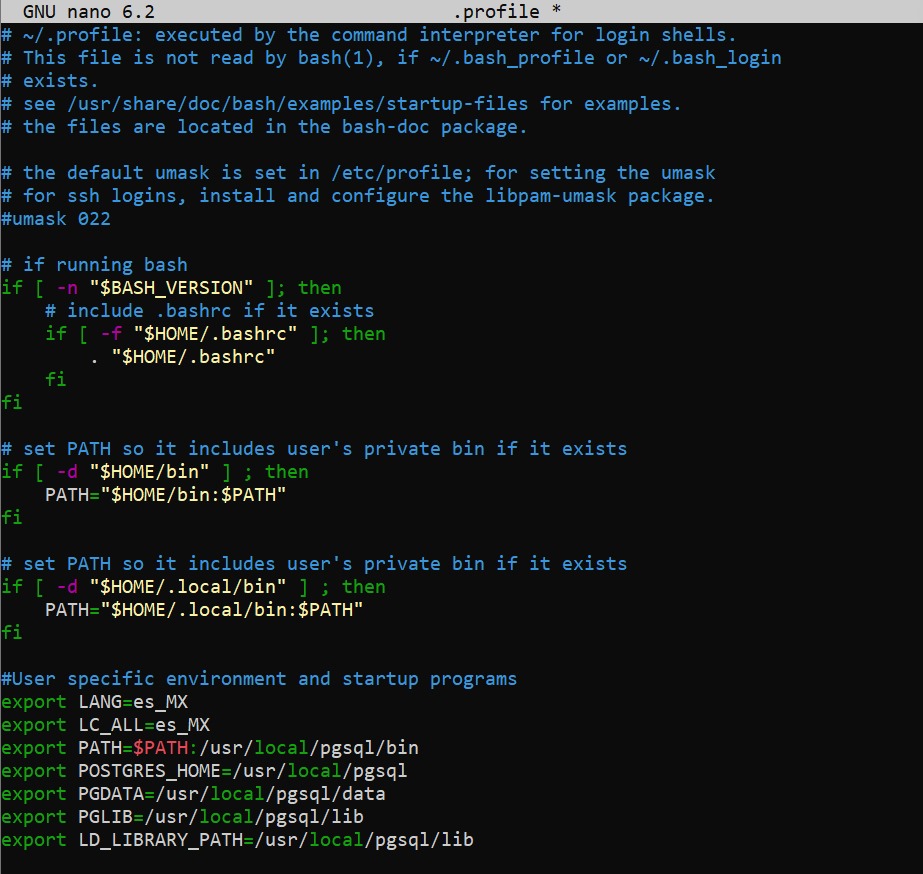
Volvemos a ejecutar el ./configure y comprobamos que ahora ya no nos aparece ningún error

Ahora se compila el código fuente de postgres

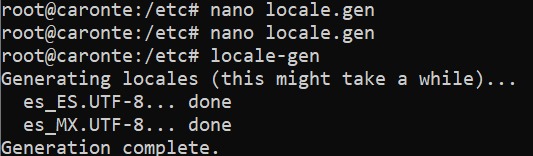




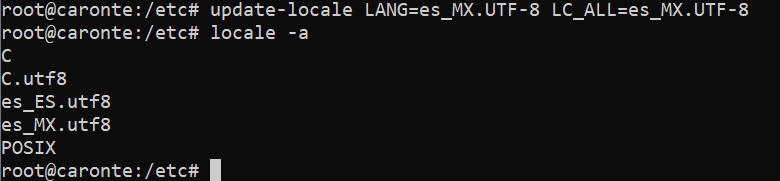
Ahora modificamos las variables de entorno



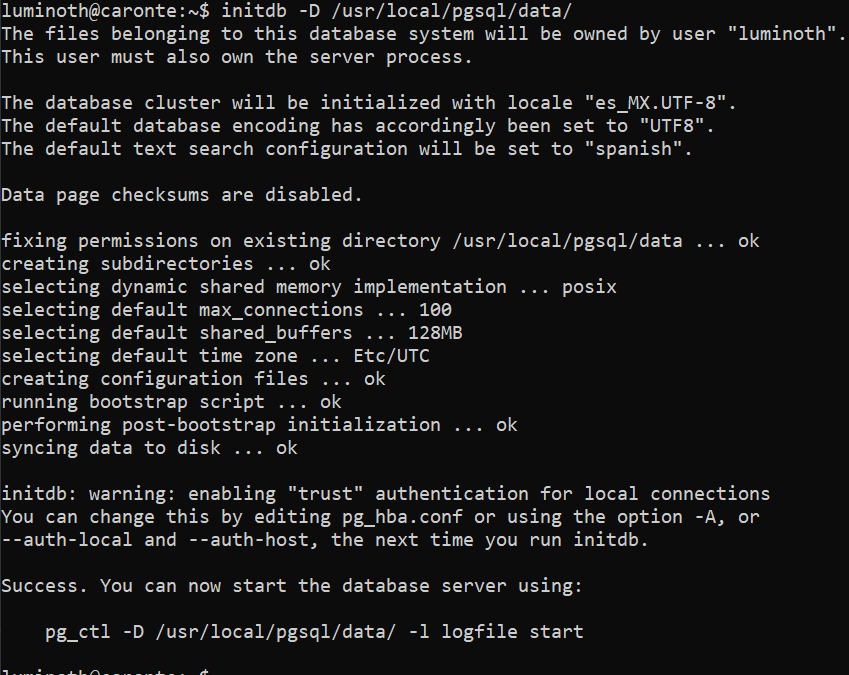
Modificamos un error que se detecto en el horario en el archivo /etc/locale.gen



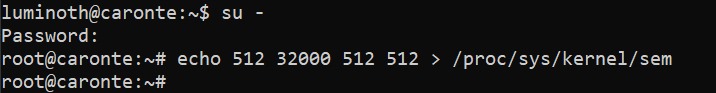
Y lo actualizamos



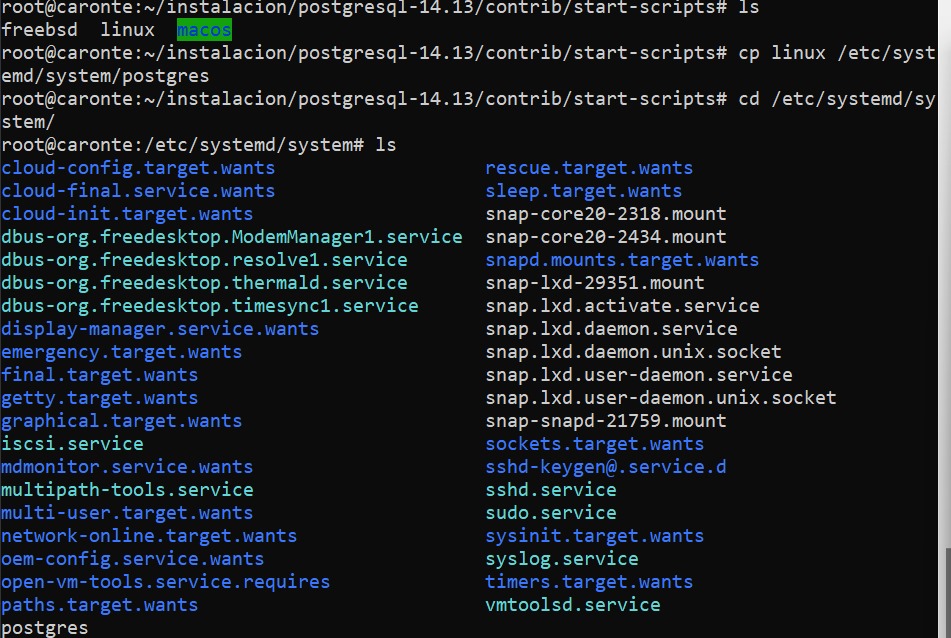
Inicializamos el cluster



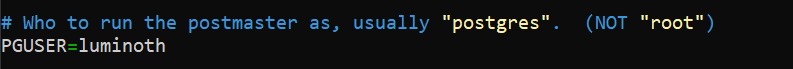
Ajustamos los límites de semáforos del kernel de Linux desde root



Nos movemos a la siguiente ruta



Y copiamos el archivo de Linux a la siguiente ruta, modificamos el archivo de postgres con nano.



Se agrega el -i después del PGDATA



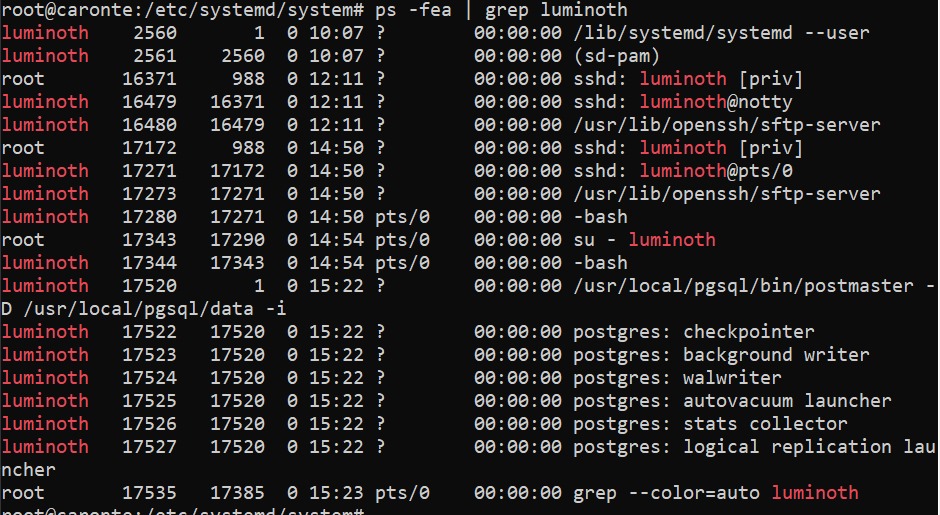
Damos Permisos de ejecución



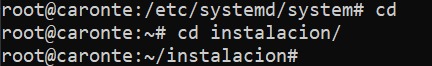
Iniciamos la operación de postgres



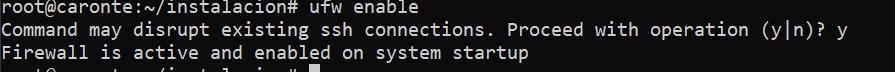
Verificamos que ya este arriba



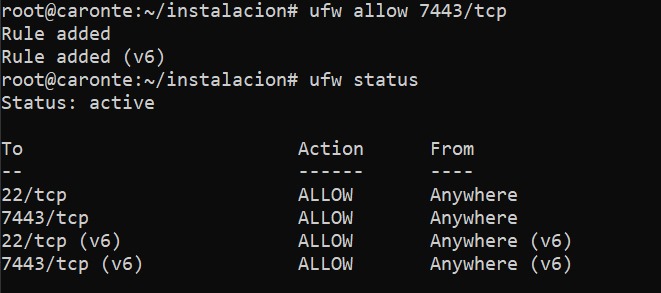
Nos movemos a la carpeta de instalación



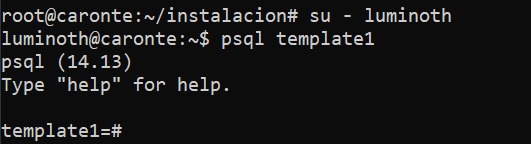
Habilitamos el firewall ufw



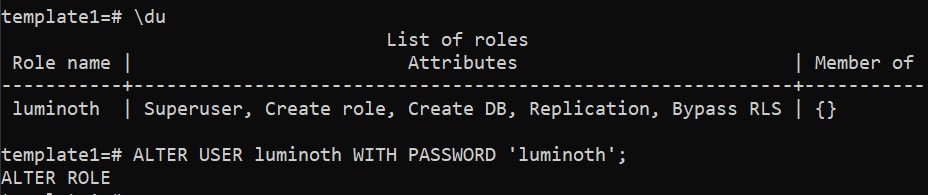
Abrimos el puerto 7343 con ufw y verificamos que este abierto



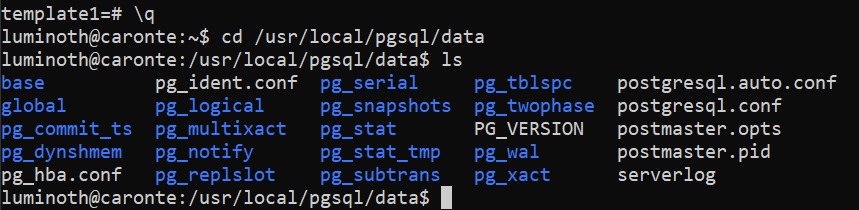
Asignamos una contraseña a luminoth para que pueda manejar las bases de datos



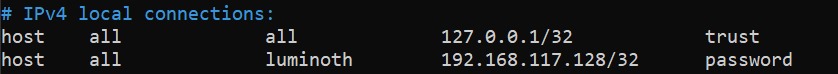
Observamos que ya esta y le vamos a poner contraseña



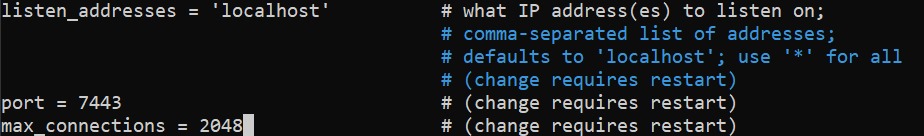
Salimos del manejador y nos dirigimos a la siguiente ruta



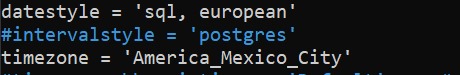
Con nano modificamos el archivo pg\_hba.conf y agregamos al usuario luminoth junto con el ip de la maquina virtual de Ubuntu

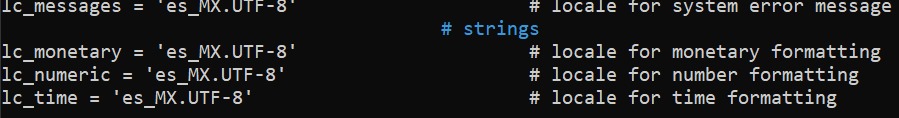
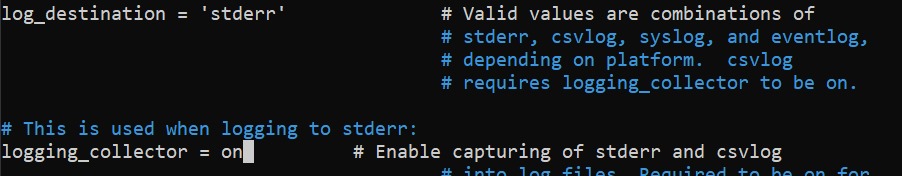


Ahora se modifica el PostgreSQL.conf

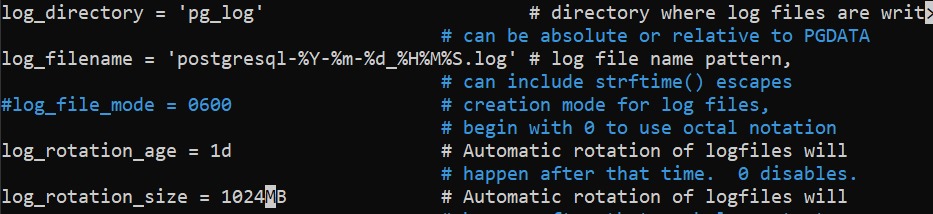


Se modifica datestyle y timezone





Se modifica log a log

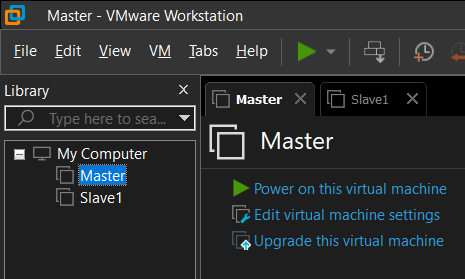


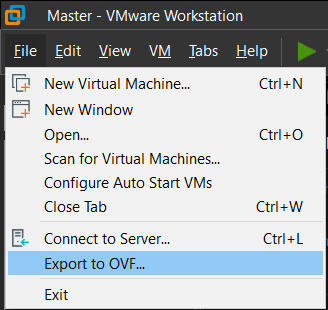
Y se prenden los tres

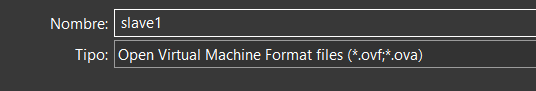


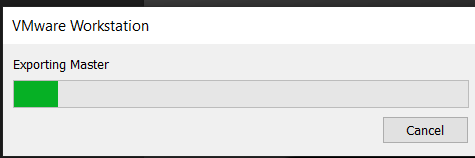
Replicación de base de Datos

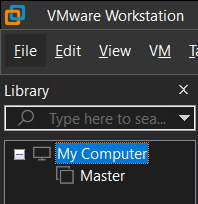
CLONACION DE LA MÁQUINA VIRTUAL:

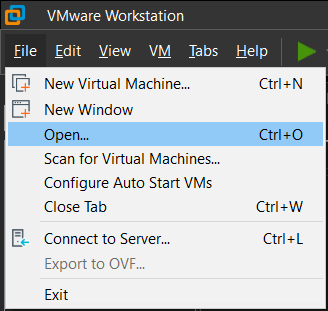
 Teniendo la máquina Master

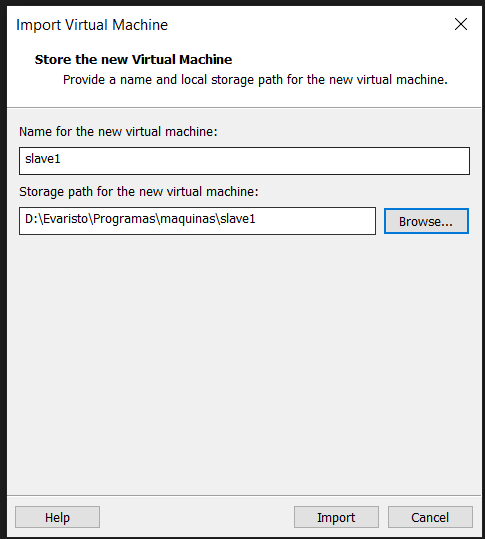
 Exportar a OVF

Guardar el archivo

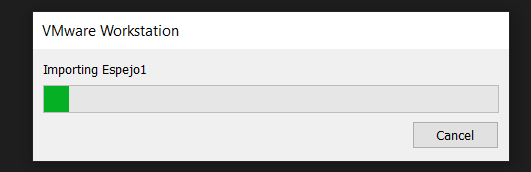
Exportando

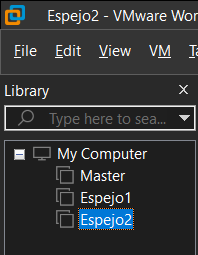
En my computer

File 🡪 open



Importar la máquina virtual

Importando la máquina virtual: Nota -> el nombre de la máquina se cambió a espejo1, por problemas con las carpetas

Ya se tienen las dos máquinas virtuales

Ahora, las configuraciones son exactamente las mismas en ambas máquinas. El proceso sólo se hará para la primera: mercurioe1

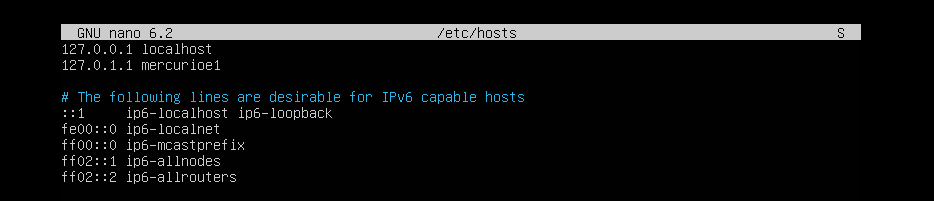
Cambiando el hostname

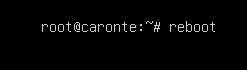
 modificar el archivo hostname



Y modificar

/etc/hosts



Reiniciar la máquina

REPLICACION

Otorgar permisos de replicación al usuario:

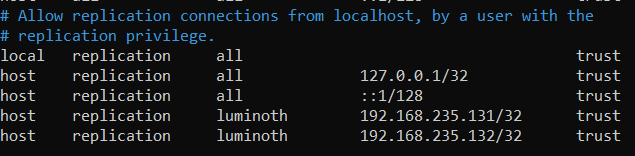


….

IP de mercurioe1

IP de andromedae2

En pg\_hba.conf



Se agregaron las dos últimas líneas

Configuración de los parámetros de replicación en:



Se hicieron estas modificaciones











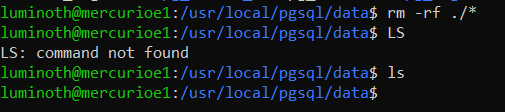




EN mercurioe1:

Nos pocicionamos en la carpeta 

Y borramos todo el contenido con:



Dentro de la carpeta data vacía, creamos la copia del servidor maestro Caronte.





Configuraciones en el archivo postgresql.conf en el servidor mercurioe1

En el archivo: 

Cambios:

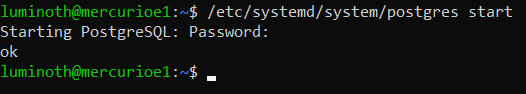




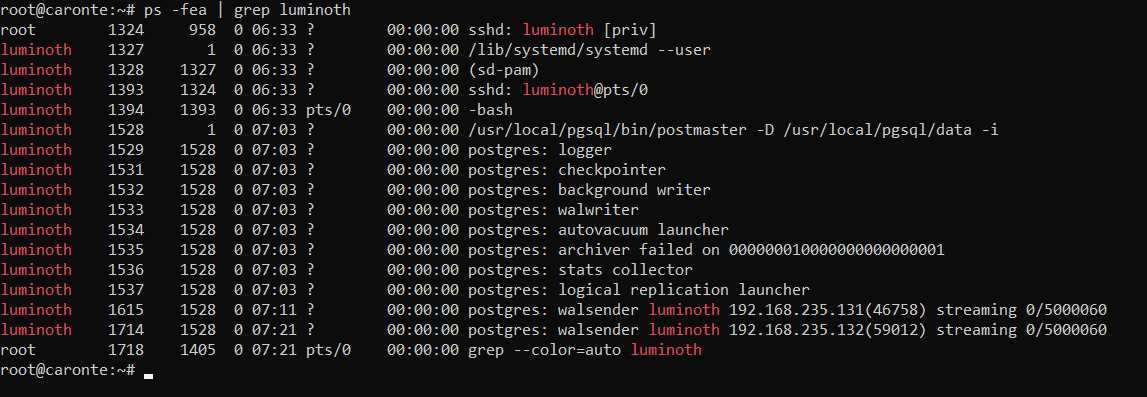




Activar el servicio de PostgreSQL en mercurioe1



Comprobando que la replicación de los dos espejos se haya implementado correctamente:



Comprobando replicación con pgadmin

