|  |
| --- |
|  |
| Donkey Guide |
|  |
| Consejos útiles para tus primeros días en Alstom |

Índice

[I. Configuración inicial del PC con Autopilot 2](#_Toc176505300)

[II. Instalar software en el PC 2](#_Toc176505301)

[III. Solicitar permisos de Administrador 3](#_Toc176505302)

[IV. Configurar Python en VSCode 3](#_Toc176505303)

[V. Configurar « autoformatter » en VSCode 5](#_Toc176505304)

[VI. Cuenta en SAP Concur 6](#_Toc176505305)

[VII. GitLab 7](#_Toc176505306)

[VIII. Deepbox 9](#_Toc176505307)

[IX. Conectarse a PostgreSQL 9](#_Toc176505308)

[X. Solicitar permisos de administrador 11](#_Toc176505309)

[XI. Copiar ficheros de forma remota entre dispositivos en la misma red 11](#_Toc176505310)

[XII. Configuración NAT router Juniper 12](#_Toc176505311)

[XIII. Servicio FTP/SFTP 13](#_Toc176505312)

[XIV. Creación de usuarios 15](#_Toc176505313)

[XV. Servicio RDP 15](#_Toc176505314)

[XVI. Crear y activar un entorno virtual en VSCode 19](#_Toc176505315)

[XVII. Restaurar resolución de pantalla (drivers de NVidia no operativos) 19](#_Toc176505316)

[XVIII. Instalar git-sim 20](#_Toc176505317)

[XIX. No se pudo obtener el bloqueo /var/lib/dpkg/lock 20](#_Toc176505318)

[XX. Como desplegar una app web/API de Python en un servidor WSGI 21](#_Toc176505319)

|  |
| --- |
| DONkey guide |

Configuración inicial del PC con Autopilot

* Deberías de seguir los pasos indicados en el manual que se te proporciona con el PC (“Manual de Autopilot. Configuración Automática de Equipos”).
* En caso de que algo no vaya según lo previsto, contacta directamente con IT a través del teléfono 918351031.
* Si en algún momento crees que necesitas un correo y una contraseña, es posible que a tu manager le hayan enviado previamente a tu incorporación un correo con tal información. Pídeselo si es que no te lo ha mandado.
* Se supone que tienes que dejar 2.5h el PC conectado a la corriente mientras se realiza la instalación automática de todas las aplicaciones corporativas. NO sale ninguna barra de estado o de progreso, así que no intentes saber cuánto queda de proceso. Cuando tengas nuevos iconos en tu Escritorio, es posible que todo se haya completado con éxito.

Instalar software en el PC

Hay dos formas de instalar SW en el PC, dependiendo si eres o no Admin.

Si no tienes el privilegio de ser Administrador, olvídate de descargar nada de Internet. Ve directamente al “Software Center” (icono de Windows AppStore, que deberías tener en tu Escritorio tras la configuración inicial), y descarga lo que puedas de ahí.

Por el contrario, si eres de esos pocos con privilegios, puedes descargar e instalar SW no incluido en el Software Center simplemente introduciendo tu usuario y pwd de Admin cuando lo solicite el cuadro de diálogo:

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Es posible que tú también quieras estar en ese selecto grupo. Échale un ojo al punto 3.

Solicitar permisos de Administrador

Es posible que en un primer momento sea tu manager el que tenga que solicitar tales permisos. Esto lo hará a través del Servicio de requerimientos de IT (<https://alstom.service-now.com/asp>). Guárdate el link, te será útil en un futuro próximo.

Una vez te gestionen la solicitud de tu manager, te harán llegar un User y un Password a tu correo, los cuales debes usar cuando te salga la ventana de “Control de cuentas de usuario” (como la del punto 2).

Es muy probable que esto no te funcione a la primera, así que aquí tienes varias opciones que puedes intentar:

* Reiniciar el PC para que los cambios surtan efecto.
* Introducir tu correo de Alstom y la contraseña del mismo en la ventana de “Control de cuentas de usuario”.
* Introducir el comando “gpupdate /force” en el CMD, para actualizar las directivas de usuario.

En caso de que nada de esto sirva, sin duda tendrás que pedir ayuda, así que tienes varias opciones:

* Volver a poner una incidencia en el Servicio de requerimientos de IT
* Usar el Chat en Directo para hablar directamente con un operador que te solucionará el problema (funcionan realmente bien).

Después de esto, deberían solucionártelo sin problema.

Configurar Python en VSCode

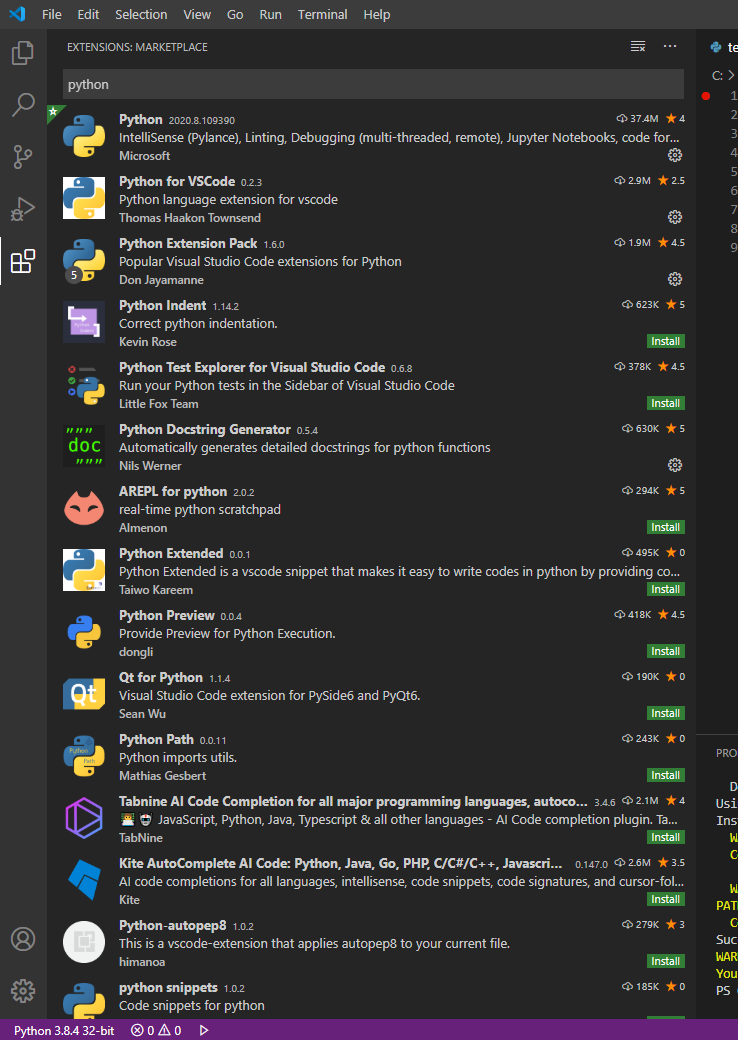
Es posible que sea tu primera vez usando VSCode así que tranquilo, configurarlo es un poco tedioso.

En este caso queremos usarlo para desarrollo en Python, por lo que habrá adaptarlo para tal fin.

* + Extensiones

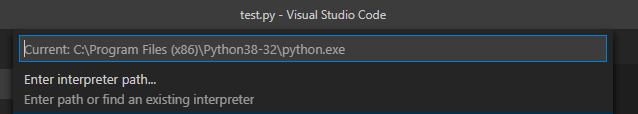
Si hacemos click en la sección de “Extensiones” de la parte izquierda nos saldrá un buscador en la parte superior. Escribimos en él “Python”. Nos aparecerán todas las extensiones instalables en VSCode para personalizar nuestro IDE. Solamente tendremos que darle a “Install” para que se realice el proceso (ver imagen).

Sin embargo, una parte fundamental (y sin la cual nada de esto funcionaría) es tener el intérprete de Python descargado en nuestro PC. Para ello, iríamos al “Software Center” del punto 2 y descargaríamos la última versión disponible de Python.



* + Intérprete

Una vez instalado, tenemos que decirle a VSCode qué intérprete queremos. Para ello, hacemos click en la esquina inferior izquierda de la ventana de VSCode (en naranja en la imagen), y se nos desplegará una lista en la parte superior central. Haremos click en “Enter interpreter path…” y, seguidamente, introduciremos la ruta en la que está nuestro intérprete.



Pero, ¿dónde está? Pues muy probablemente se te haya instalado en la siguiente ruta: “C:\Program Files (x86)\Python38-32”, o en una similar de tu PC.

También es posible que VSCode haya detectado varios intérpretes en tu PC, por lo que directamente puedes seleccionar uno de ellos desde la lista que te facilita en lugar de introducir la ruta manualmente:

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

* + Instalar librerías
  + Debería ser tan sencillo como introducir este texto en CMD o en el Terminal de VSCode:
  + pip install nombre\_de\_la\_libreria

Sin embargo, la vida no es de color de rosa. Es posible que no te funcione. Aquí van varias opciones para intentar solucionarlo:

* Si te sale un error del tipo “ERROR: Could not install packages due to an EnvironmentError”, es porque estás conectado por cable a la red de Alstom (no te va a dejar descargar nada). Tendrás que usar, por tanto, un Stick USB de Internet.
* Es posible que tengas que abrir el CMD como administrador.
* Puede que tengas que meter la ruta de Python en las variables de Sistema:

Windows+E>”Este equipo” (botón derecho)>Propiedades>Configuración avanzada del sistema (necesitarás permisos de Admin)>Opciones avanzadas>Variables de entorno>Path (editar) -> pegamos la ruta del intérprete de Python.

Configurar « autoformatter » en VSCode

¿Cuántas veces nos hemos preguntado por qué no programamos todos de la misma manera? ¿Cuántas veces hemos deseado que nuestro compañero use el mismo formato, estructura o sintaxis que nosotros? Aquí está el autoformateo para ayudarte.

Existen varias extensiones que permiten formatear un script con un determinado formato de manera automática. Algunas son *Autopep8, Black o Yapf.* En este caso nos centraremos en la manera de configurar *Black*.

Para ello, seguiremos estos sencillos pasos (VE AL PUNTO IX PARA PERMISOS):

### Lo primero que haremos es ejecutar la siguiente sentencia en el terminal:

pip install black

o si no funciona, usa este otro:

py -m pip install black

o

python -m pip install black

### Después, presionaremos « Ctrl + p » y, en el prompt superior pegamos la siguiente línea y presionamos  « Enter ». Dejamos que se instale la extensión.

ext install ms-python.python

### Seguidamente, presionamos « Ctrl + , » o vamos al icono del engranaje (parte inferior izquierda), y clicamos en « Settings ». En el buscador superior pegamos la siguiente sentencia :

format on save

Y clicamos en el cuadro de « Editor : Format on Save »

### En el buscador pegamos la siguiente sentencia :

python formatting provider

Y seleccionamos el que deseemos. En este caso, *Black*.

### ¡Y ya está ! Simplemente, cada vez que guardemos un script, VSCode se encargará de corregir el formato de lo que hayas escrito. Disfruta !

Cuenta en SAP Concur

¿Eres una persona de negocios y necesitas cargar gastos a la empresa? Esto te interesa.

Si se desea imputar cualquier gasto derivado de la actividad que realizamos es necesario hacerlo a través de la aplicación SAP Concur (checa este link [Iniciar sesión en Concur | Concur Solutions](https://www.concursolutions.com/nui/signin)).

El usuario que deberás incluir será de la forma:

[número\_de\_empleado\_de\_ALPS@alstom.com](mailto:número_de_empleado_de_ALPS@alstom.com)

La primera vez que intentes acceder no tendrás contraseña asignada, por lo que habrá que crear una nueva cuenta. Para ello tienes que seguir los siguientes pasos:

### Mandarle un correo a Ángel Castaños ([angel.castanos@alstomgroup.com](mailto:angel.castanos@alstomgroup.com)), solicitando la creación de un nuevo usuario.

### Te responderá con un correo para los pasos que hay que seguir (ver mail adjunto). Para que se pueda crear una cuenta de Concur, previamente has de tener actualizada tu cuenta bancaria en ALPS y tener creado un perfil en GSI (ver paso 4).

### 

### RRHH debería de aceptar la actualización de tus datos bancarios en ALPS.

### Para tener un perfil en GSI, debes de ponerte en contacto con el Key User de GSI (Jerome Bompart, [jerome.bompart@alstomgroup.com](mailto:jerome.bompart@alstomgroup.com)). Esta persona te pedirá, a su vez, un justificante bancario de titularidad donde aparezcan los datos de tu cuenta y tu nombre, para poder hacer tal cuenta. Se te avisará cuando este perfil de GSI se haya completado.

### Una vez llegados hasta aquí, hay que ponerse de nuevo en contacto con Ángel Castaños para que, por fin, te genere el perfil de Concur.

Notas

También es muy útil la app móvil de SAP Concur (así que descárgala en tu móvil inteligente). Esta app se vincula automáticamente a otras apps de viajes (como Uber o Cabify), de manera que todos los tickets o facturas que se hagan con ellas se cargarán directamente en nuestra cuenta de Concur (poniéndonos las cosas fáciles).

La app de Concur es útil para la creación de nuevos gastos. Sin embargo, para hacer la imputación y solicitar la aprobación de tu manager, deberá hacerse desde la web.

GitLab

Si te ha tocado empezar a lidiar con Git significa que te estás convirtiendo en todo un programador, ¡enhorabuena!

Los inicios en estos sistemas de control de versiones suelen ser un poco duros, pero aquí tienes unos pequeños apuntes para hacértelo más sencillo.

* + Empezamos:

Lo primero de todo es ir a la URL del sitio: <https://gitlab.com/>

Aquí deberás registrarte si aún no eres usuario o meter tus credenciales (vamos, as usual). El departamento tiene un repositorio con el siguiente user y password:

* User: monitorizacioncn
* Pwd: Alstom123\_
  + Crear un nuevo proyecto o unirte a uno ya existente:

Si quieres empezar tu propia aventura en Git, puedes crear un nuevo proyecto simplemente haciendo click en *New Project*.

Ahí tendrás varias opciones (crear un proyecto desde cero, importarlo, etc.). Por lo general querremos empezar uno nuevo, así que iremos a *Create blank Project*.

En esta nueva sección simplemente habrá que introducir el nombre de proyecto y el tag que queremos darle al proyecto para que aparezca en la URL (por ejemplo, *nuevo\_proyecto*). Se podrá añadir una descripción del proyecto (si eres de dar muchas explicaciones) y seleccionar si queremos que sea *Public* o *Private*, dependiendo de si queremos que sea explícitamente visible desde dentro de nuestro grupo de trabajo o por todos los usuarios de GitLab.

Una vez se haya creado el proyecto en el repositorio de GitLab, habrá que vincularlo a alguna carpeta en nuestro PC. Veamos el paso siguiente.

* + Vincular un Proyecto en GitLab a nuestro PC:

Este es el paso clave. Crearemos una carpeta en nuestro PC directamente vinculada con la URL del proyecto en el repositorio de GitLab.

Para ello, dentro de la página del proyecto en GitLab nos aparecerá la opción *Clone*. En ella, copiaremos la ruta que pone *Clone with HTTPS* (por ejemplo, <https://gitlab.com/monitorizacioncn/nuevo_proyecto.git>).

Seguidamente, deberemos tener instalado en nuestro ordenador GitBash (un terminal para Windows que ofrece utilidades basadas en Unix para gestión de comandos en línea de Git).

Para hacer el control de versiones con GitBash, lo más cómodo es situar nuestro terminal en el directorio en el que queremos trabajar. Para ello podemos hacerlo de dos maneras:

* Abrir GitBash y navegar por las carpetas hasta llegar a la ruta que nos interesa (con comandos cd y cd.. de Unix).
* Abrir GitBash directamente en la carpeta que queremos a través del explorador de archivos, haciendo click en el botón derecho>*GitBash here*.

Una vez en la ruta deseada, simplemente escribiremos: *git clone [ruta del repo]*. Con el ejemplo de antes:

*git clone* [*https://gitlab.com/monitorizacioncn/nuevo\_proyecto.git*](https://gitlab.com/monitorizacioncn/nuevo_proyecto.git)

Se nos creará una carpeta con el mismo nombre del proyecto (*nuevo\_proyecto*) dentro del directorio en el que nos encontremos. ¡Y ya está!

* + Comandos para trabajar con Git:

Una vez que hayamos vinculado el repo a tu PC, hay varias acciones interesantes que conviene conocer. Todos los comandos a continuación se ejecutan sobre GitBash:

* Estado del repo: para detectar posibles cambios, tanto en el repo y que nosotros no tengamos en local, como cambios en local que aún no han sido enviados al repo, podemos introducir:
  + git status
* Añadir algo al repo: en caso de que hayamos incluido algo en nuestra carpeta de proyecto que aún no está en el repo, introduciremos:

git add .

Esto indica a Git qué queremos subir. Si se escribe el punto (‘.’) significa que queremos añadir todo lo nuevo.

Tras esto, tenemos que hacer un commit. Esto une los cambios locales a la copia local del repo. La sentencia es tal que:

git commit -m ‘comentario’

El *comentario*, que tiene que ir entre comillas(‘ ‘), es obligatorio.

Por último, para que esa nueva copia local con todos los cambios se suba al repositorio online, es necesario ejecutar este comando:

git push

Con todo esto ya habríamos compartido nuestros cambios con el resto de usuarios del repositorio. En caso de que el comando *status* nos indique cambios en el repo que aún no están en nuestra copia local, para actualizarnos tendremos que ejecutar:

git pull

En este caso, lo que hace Git es descargar en local la última versión del repo y, en caso de conflicto con nuestros propios ficheros, hacer un *fetch/merge*.

Y esto es todo lo básico para empezar en el apasionante mundo del control de versiones. ¡Bienvenido!

Deepbox

Deepbox es un servidor propio del departamento ubicado en la planta baja de las instalaciones de Alstom en Pinto.

Su usuario es *deepbox* y su contraseña *Alstom123\_* . Si te quieres conectar a él de forma remota, échale un ojo a *Conectarse a Deepbox.*

Conectarse a PostgreSQL

En caso de que necesites acceder a ingentes cantidades de datos, es posible que tengas que conectarte a alguna base de datos. Una opción interesante (y gratuita) es usar PostgreSQL.

PostgreSQL se compone de dos partes: por un lado el servidor de bases de datos, que estará ubicado en cualquier ordenador y es el que almacena toda la información, y por otro lado el visualizador (administrador) que nos sirve para interactuar con ese servidor.

* + Instalación de PostgreSQL:

Por ello, primero deberemos instalar y crear un servidor de bases de datos en algún lugar. Para ello nos descargamos PostgreSQL. Una vez instalado, necesitamos cambiar unos valores de sus ficheros de configuración. Por defecto, PostgreSQL solo permite el acceso a su servidor desde el localhost. Sin embargo, como queremos crear un servidor accesible desde cualquier punto, hemos de cambiar las siguientes líneas:

* En el archivo *postgresql.conf,* descomentar la línea de ‘listen\_addresses’ y cambiar su valor de 127.0.0.1:

Imagen que contiene Forma

Descripción generada automáticamente

* Reiniciar el servidor de PostgreSQL con este comando: *systemctl restart postgresql.service*
* Cambiar la configuración de autenticación añadiendo las siguientes líneas al final del archivo *pg\_hba.conf*:



Con esto ya tendríamos listo el servidor para poder acceder a él desde cualquier IP.

* + Instalación de pgAdmin:

Este SW es el visualizador del servidor y nos servirá para interactuar con las bases de datos de forma más intuitiva.

Una vez te lo descargues e instales, tendrás que definir un usuario y contraseña.

* + Conexión a PostgreSQL:

Para ello simplemente habrá que crear un servidor nuevo (botón derecho en *Servers*>*Create>Server*). En la ficha habrá que indicar el nombre del servidor en la pestaña *General*, y la IP, puerto (ver NOTA en *Conectarse a Deepbox*), nombre de la base de datos de mantenimiento, usuario y contraseña en la pestaña *Connection*.

* + Conectarse a Deepbox:

Deepbox es el servidor propio del departamento, ubicado en la planta baja de las instalaciones de Pinto. Este ordenador está conectado a una red con IP fija, lo cual nos permite acceder a él desde cualquier lugar (fuera de la red de Alstom).

Tendremos dos maneras de conectarnos y a dos puertos diferentes:

* Conexión desde la red BTING2018\_5Ghz:

Este es el router que provee la IP fija al servidor Deepbox. Si nos conectamos por wifi con nuestro portátil a este router, estaremos dentro de la misma subred que el servidor, por lo que necesitaremos saber simplemente la IP del concentrador y el puerto del servidor:

IP concentrador: **192.168.0.2**

Puerto Deepbox: **88**

**NOTA 1**: si quisiéramos conectarnos a este equipo por SSH, estos datos serán los que haya que introducir para acceder al servidor de Deepbox.

Si en lugar de al propio servidor nos quisiéramos conectar directamente al servidor de PostgreSQL, éste tiene un puerto propio dentro de la misma IP:

IP concentrador: **192.168.0.2**

Puerto PostgreSQL en Deepbox: **5432**

**NOTA 2**: Estos últimos datos serán los que haya que introducir en el formulario de creación de servidor de PostgreSQL para conectarnos al servidor de Deepbox.

* Conexión desde una red externa:

Gracias a la IP fija que provee el router de Pinto, podremos tener acceso a nuestro Deepbox desde cualquier punto (allá vamos, teletrabajo!). Para ello, simplemente tendremos que apuntar a la IP pública del concentrador, manteniendo los mismos puertos:

IP pública: 95.63.34.230

Puerto Deepbox: **88**

Puerto PostgreSQL en Deepbox: **5432**

Solicitar permisos de administrador

Si por circunstancias es necesario que realices alguna acción especial en el PC, tendrás que otorgar a dicha acción PERMISOS DE ADMINISTRADOR. Para ello, tendrás que solicitarlos en [Service Now](https://alstom.service-now.com/asp/).

En el apartado *Request something* encontrarás el bloque *PC Administration Rights Add/Remove*. Solo tendrás que clicar en él y rellenar el formulario.

Copiar ficheros de forma remota entre dispositivos en la misma red

Si dos dispositivos están conectados a la misma red local es posible compartir ficheros entre ellos. Para enviar objetos del equipo A al equipo B, esto se puede realizar bien desde el propio dispositivo A, o bien desde un equipo C conectado a A a través de una conexión remota.

El comando para copiar es tan simple como esto:

scp -v userA@IPsourceA:/pathOfTheFileToCopy userB@IPdestinationB:/pathWhereToCopy

Sin embargo, la aplicacion *scp* utiliza un túnel SSH para hacer una copia segura entre los equipos, por lo que habrá que asegurarse de que ambos equipos tienen permisos para realizar una conexión SSH entre ellos. Para comprobar/configurar esto, los pasos son los siguientes:

* Comprobar la conexión SSH:

ssh -v userB@IPdestinationB

Si la conexión SSH se permite, es que ya está todo configurado. Si se nos muestra algún mensaje de “*Permission denied”* será necesario hacer algunos pasos más.

* Crear una clave SSH pública en el equipo A.

ssh-keygen -t rsa

* Compartir la clave pública generada con el equipo B.

ssh-copy-id -i ~/.ssh/id\_rsa.pub userB@IPdestinationB

Si esto funciona, se habrá generado un fichero *authorized\_keys* en el equipoB, en la carpeta */home/userB/.ssh*.

Probamos de nuevo la conexión SSH. Si aún así siguen existiendo problemas de permisos, se pueden hacer algunas acciones adicionales en ambos equipos:

Nota: los cambios en los ficheros de configuración de SSH pueden reducir la seguridad del equipo.

* Borrar los ficheros *known\_hosts* de la carpeta */home/user/.ssh* en ambos equipos.
* Configurar los siguientes parámetros en el fichero /etc/ssh/ssh\_config:
  + - PasswordAuthentication no
    - StrictHostKeyChecking no
* Configurar los siguientes parámetros en el fichero /etc/ssh/sshd\_config:
  + - PermitRootLogin no
    - PubkeyAuthentication yes
* En ambos casos, relanzar el servicio:
  + - sudo systemctl restart ssh
    - sudo systemctl restart sshd

Si esto sigue dando errores, probar a acceder desde y hacia todos los usuarios (del root A al root B, del root A a otro usuario en B, etc.)

Configuración NAT router Juniper

La configuración NAT permite redireccionar el tráfico que circula a través de un router. En este caso, el router Juniper es al que se conectan la mayoría de dispositivos del laboratorio de Pinto.

Para acceder a la configuración del dispositivo se introduce la IP del router (192.168.0.1). El usuario y la contraseña son administrador/bombEMS14.

Una vez dentro, la edición de NAT se puede hacer a través la pestaña NAT o de la pestaña CLI. En general, es más sencillo hacerlo a través de la GUI de la pestaña NAT. Sin embargo, si existen muchas reglas creadas es posible que la interfaz no pueda cargar tanta información, así que la opción es crear las reglas a mano, a través de su fichero de configuración.

El modo de operación es acceder a la pestaña CLI, copiar el fichero actual directamente desde la ventana del navegador, guardarlo en un fichero de texto en el PC (para tener un fichero de backup), editarlo de forma local (añadiendo las reglas y los pool de funciones correspondientes), y pegar de nuevo la información editada en la ventana de CLI del navegador. Seguidamente, hacer *commit*.

Tabla de enrutado actual:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Regla | Pool | IP |  | Puerto | Servicio | Puerto servicio |
| Rule\_HHSSH\_88 | HHSSH\_88 | 192.168.2.55 | 192.168.0.2 | 88 | ssh | 22 |
| Rule\_HHSFTP\_8822 | HHSFTP\_8822 | 192.168.2.55 | 192.168.0.2 | 8822 | sftp | 22 |
| Rule\_HHFTP\_8821 | HHFTP\_8821 | 192.168.2.55 | 192.168.0.2 | 8821 | ftp | 21 |
| Rule\_HHPOSTGRESQL\_5432 | HHPOSTGRESQL\_5432 | 192.168.2.55 | 192.168.0.2 | 5432 | postgres | 5432 |
| Rule\_HHSHINYPROXY\_5431 | HHSHINYPROXY\_5431 | 192.168.2.55 | 192.168.0.2 | 5431 | shiny | 5431 |
| Rule\_RDP\_8889 | HHRDP\_8889 | 192.168.2.55 | 192.168.0.2 | 8889 | RDP | 3389 |

Servicio FTP/SFTP

Para permitir la transferencia de ficheros entre distintos equipos de manera remota se debe habilitar un servicio de FTP en el dispositivo remoto. Los pasos para un SO Linux son los siguientes:

* Actualizar los paquetes del SO:
  + sudo apt update
* Instalar la aplicación FTP:

sudo apt install vsftpd

* Iniciar y habilitar el servicio FTP:

sudo systemctl start vsftpd

sudo systemctl enable vsftpd

* Es recomendable hacer una copia del fichero de configuración original:

sudo cp /etc/vsftpd.conf /etc/vsftpd.conf\_default

* Crear el usuario FTP y su contraseña:

sudo useradd -m *user\_name*

sudo passwd *user\_name*

* Configurar el firewall para que el tráfico FTP vaya por el puerto elegido:

sudo ufw allow 20/tcp

sudo ufw allow 21/tcp

* Contectar el PC con servidor FTP

Sudo ftp name\_of\_the\_system

Para conseguir el *name\_of\_the\_system* hay que introducir *hostname* en el terminal.

* Es recomendable también cambiar el directorio por defecto del servicio FTP:

sudo mkdir /srv/ftp/*new\_location*

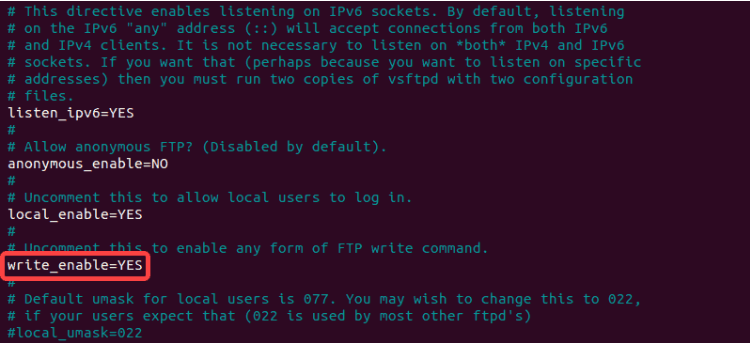
sudo usermod -d /srv/ftp/*new\_location* ftp

* Para que los cambios surtan efecto, se resetea el servicio FTP:

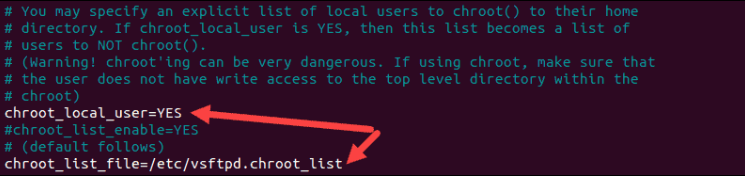
Sudo systemctl restart vsftpd.service

* Autenticar los usuarios FTP:

sudo nano /etc/vsftpd.conf



* + Resetear el servicio una vez hechos los cambios.
* Para limitar el uso del servicio FTP por usuarios hay que añadir dos comandos en el fichero de configuración.



* + En caso de que el archivo *chroot\_list* no esté en el directorio */etc* será necesario crearlo.

**NOTA respecto FTP:**

El servicio dejó de estar operativo una temporada y no se lograba dar con lo que causaba la falla. Se propusieron utilizar puertos pasivos. Así, se incluyó al final del *vsftpd.conf*:

pasv\_enable=YES

pasv\_min\_port=8819

pasv\_max\_port=8820

También se tienen que habilitar esos dos puertos en el firewall del Deepbox:

sudo ufw allow 8819

sudo ufw allow 8820

Además, hay que crear las correspondientes *rule* y *pool* en el Juniper para hacer el NAT entre los puertos 8819 y 8820 del Juniper al 8819 y 8820 del Deepbox, respectivamente.

**NOTA respecto SFTP:**

El ChrootDir de SFTP tiene que ser propiedad de root:

sudo chown root:root /var/www/RESTRICTED\_DIR

sudo chmod 755 /var/www/RESTRICTED\_DIR

Posteriormente, los contenidos de dicha carpeta sí tienen que ser solo propiedad del usuario *sftp*:

sudo usermod -a -G www-data MY\_USER

sudo chown MY\_USER:www-data /var/www/RESTRICTED\_DIR/\*

sudo chmod 775 -R /var/www/RESTRICTED\_DIR/\*

Creación de usuarios

Para crear nuevos usuarios en Linux hay que seguir los siguientes pasos (revisar USERMOD):

* Creación de usuario:

sudo useradd *nombre\_usuario*

* Creación de contraseña:

sudo passwd *nombre\_usuario*

* Asignación de una carpeta de inicio (por defecto o personalizada):

sudo useradd -m *nombre\_usuario*

sudo useradd -m -d */path/de/la/carpeta nombre\_usuario*

* Asignación de un *Shell*:

sudo useradd -s */ruta/del/Shell nombre\_usuario*

Servicio RDP

Para habilitar el servicio de Escritorio Remoto en Ubuntu habrá que seguir los siguientes pasos. Desde un terminal del equipo:

* Se instala la aplicación de Escritorio Remoto XRDP:
  + - sudo apt-get install xrdp
* Se instala el entorno de escritorio XFCE:
  + - sudo apt-get install xfce4
* Se puede instalar como opción, MATE:
  + - sudo apt-get install mate-core mate-desktop-environment mate-notification-daemon
* Puede que nos pida elegir entre alguno de los “gestores de aplicaciones”, si es que hay más de uno instalado en el equipo (e.g. gdm3, kdm o lightdm).
* Se ejecuta el siguiente comando para añadir unas líneas al fichero de configuración de XRDP:
  + - sudo sed -i.bak '/fi/a #xrdp multiple users configuration \n xfce-session \n' /etc/xrdp/startwm.sh
* En caso de haber instalado MATE y no XRDP:
  + - sudo sed -i.bak '/fi/a #xrdp multiple users configuration \n mate-session \n' /etc/xrdp/startwm.sh
* Se añade el usuario que queramos al certificado de ssl generado por XRDP:
  + - sudo adduser [usuario] ssl-cert
* Habilitamos el puerto 3389 en el Firewall de Ubuntu:
  + - sudo ufw allow 3389/tcp
* Si se usa IPTABLES en lugar de UFW, se pueden usar estos comandos:
  + - sudo iptables -A INPUT -p tcp --dport 3389 -j ACCEPT
    - sudo netfilter-persistent save
    - sudo netfilter-persistent reload
* Reiniciamos el servicio de XRDP (o MATE, en caso de haberlo instalado):
  + - sudo /etc/init.d/xrdp restart
* Se puede comprobar que el proceso se ha completado con éxito si somos capaces de establecer conexión con el puerto 3389 (podemos probar en localhost).
  + - telnet localhost 3389
  + Una vez que se da de alta el servicio, se puede acceder por escritorio remoto a nuestra máquina Ubuntu:

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Es posible que en un primer momento se nos quede la pantalla en negro. Esto se debe a que hay una sesión iniciada en la máquina y no nos dejaría generar una nueva. Para solucionar esto, dos opciones:

### Usar XFCE4

Entramos en el terminal de la máquina e introducimos estos comandos:

# echo "startxfce4" > ~/.Xsession

# chmod +x ~/.Xsession

# sudo systemctl restart xrdp.service

NOTA: Hay que hacer esta acción en el fichero .Xsession de cada usuario (sin sudo).

Con esto se nos cargará el entorno XFCE4, que tiene esta apariencia:

Interfaz de usuario gráfica, Patrón de fondo

Descripción generada automáticamente

### Permitir la creación de múltiples sesiones

Editar el siguiente fichero:

nano /etc/xrdp/startwm.sh

Introducir estas líneas al principio del fichero:

unset DBUS\_SESSION\_BUS\_ADDRESS

unset XDG\_RUNTIME\_DIR

Resetear el servicio

sudo systemctl restart xrdp

Con esto se cargará el mismo entorno que la máquina original (pero la latencia es mayor):

Patrón de fondo

Descripción generada automáticamente

Crear y activar un entorno virtual en VSCode

Un entorno virtual nos ayuda a “encapsular y normalizar” los recursos requeridos por un proyecto de VSCode. Esto nos permite usar, para un desarrollo de código particular, un intérprete de código o unas librerías determinadas que únicamente queramos usar en este caso. De esta manera, el desarrollo de otras soluciones que necesiten de otra configuración no impactarán sobre el funcionamiento de esta.

**CREACIÓN**

Para crear un entorno virtual existen dos grandes opciones: *venv* y *conda*. El uso de uno u otro solo dependerá de los estándares del departamento o la preferencia personal. En este caso usaremos *venv*.

* Opción 1: en VSCode, *Ctrl+Shift+P,* para entrar en la *Command Palette* y buscar “*Python: Create Environment”*. Te ofrecerá primero elegir el gestor de entornos virtuales y luego el intérprete de Python.
* Opción 2: en el terminal de VSCode, crear el entorno virtual introduciendo esta sentencia:

python -m venv .<nombre\_del\_entorno\_virtual>

Seguidamente, un pop-up te pedirá seleccionar el intérprete para ese entorno.

Tras esto, se habrá creado en tu directorio local una carpeta con el nombre que le hayas dado al entorno virtual. Ahora, para activarlo y comenzar a trabajar con él, tendrás que ejecutar el script *activate.* Debes tener cuidado con el terminal que estés ejecutando:

* Si es un CMD:

.\.<nombre\_del\_entorno\_virtual>\Scripts\activate.bat

* Si es en PowerShell:

.\.<nombre\_del\_entorno\_virtual>\Scripts\Activate.ps1

Si esto se ha realizado con éxito, el prompt del terminal mostrará, al inicio y entre paréntesis, el nombre del entorno virtual que ha sido activado.

Puedes encontrar más información [aquí](https://code.visualstudio.com/docs/python/environments).

Restaurar resolución de pantalla (drivers de NVidia no operativos)

Si ocurriese que la resolución de la pantalla del PC no puede ser cambiada desde la ventana de *Configuración de pantalla*, esto puede deberse a varias causas, como por ejemplo: los drivers oficiales no están instalados, los drivers no están adecuadamente configurados, o existe algún bloqueo por parte del SO hacia estos drivers.

En este caso, se explican los pasos para solucionar esta última casuística (se particulariza para el caso concreto de que la tarjeta gráfica es NVidia).

### Aseguar que nvidia-prime está instalado

Ejecutar en terminal el siguiente comando:

sudo apt install nvidia-prime

### Ejecutar los siguientes comandos tras su confirmación o instalación

sudo prime-select nvidia

sudo rm /lib/modprobe.d/blacklist-nvidia.conf (\*)

sudo update-initramfs -u (\*\*)

\*Es posible que no exista este fichero. En cambio, deberá haber algún otro fichero en cuyo título aparezca “nvidia” y dentro contenga una *blacklist* con datos en el formato: “blacklisk <item>”. Se borra ese fichero.

\*\*Cuando se hace este comando, se muestra por pantalla a qué versión del kernel de Linux se refiere. Es posible que haya un fichero de *blacklist* con referencia a ese kernel en su nombre. Se borra ese fichero en caso de duda y se ejecuta de nuevo este comando.

### Reiniciar el equipo

sudo reboot

Tras el reinicio, si todo ha ido según lo esperado, la resolución de la pantalla debería haberse reestablecido a un valor adecuado. En cualquier caso, en la ventana *Configuración>Pantalla*, ya se habrá identificado el monitor conectado y permitirá cambiar ciertos de sus parámetros.

Instalar git-sim

git-sim es una herramienta estupenda si necesitas quedar como un pro de Git delante de algún compañero. Para instalar esta herramienta tan útil, solamente tienes que descargarte este repo y seguir los pasos de su *Quickstart*:

<https://github.com/initialcommit-com/git-sim>

Si sigues todos los pasos adecuadamente, tendrás esta fantástica herramienta lista para ser usada. Sin embargo, posiblemente te salga un error al intentar lanzarlo. Si te falta la librería *ffmpeg* solo tienes que seguir los pasos de esta guía para solucionarlo:

<https://phoenixnap.com/kb/ffmpeg-windows>

Tras esto, todo debería funcionar correctamente y podrás sorprender a cualquiera.

No se pudo obtener el bloqueo /var/lib/dpkg/lock

Este error suele ocurrir en distribuciones que usan el gestor de paquetes *dpkg*. Al intentar instalar o desinstalar alguna aplicación, *dpkg* encuentra que sus recursos están bloqueados por otro proceso. En este punto hay dos opciones: dejar que termine el proceso (si es algo conocido), o cerrarlo.

Con el siguiente comando se listan los procesos activos:

sudo fuser -v /var/lib/dpkg/lock

Si hay algo corriendo, se puede matar con:

sudo fuser -vki /var/lib/dpkg/lock

Si no hubiera nada dentro del fichero, se elimina el propio fichero:

sudo rm /var/lib/dpkg/lock

Como desplegar una app web/API de Python en un servidor WSGI

Referencias:

<https://blog.devgenius.io/serve-python-app-on-nginx-6bc57ceaed4c>

<https://a4u.medium.com/deploy-django-with-nginx-gunicorn-and-ssl-certificate-ce7d037c7507>

<https://medium.com/@to.adarshsingh/deploying-a-python-web-service-api-server-on-ubuntu-20-04-fc45d0f23c79>

El esquema básico de esta arquitectura es el siguiente:

Diagrama

Descripción generada automáticamente

Figura 1. Esquema básico de servidor WSGI

Para que una app sea accesible desde Internet, es necesario que pase por un servidor web que, a su vez, hace de proxy reverso para redirigir las peticiones a la máquina servidora a cada una de las aplicaciones que están corriendo por debajo.

En este caso, el ejemplo es sobre una aplicación en Python, pero el esquema y pasos a seguir son similares en cualquier lenguaje. Los pasos a continuación son un resumen. Si se desean detalles técnicos se pueden mirar las referencias indicadas arriba.

### Creación de la app web en Python:

Para crear una app web en Python se usa el framework de Flask. Esto arranca un socket o puerto de nuestra máquina para que la app pueda recibir peticiones HTTP a través de él.

Una vez creada la app, comprobar que funciona de forma local.

### Middleware: Gunicorn

Gunicorn es un paquete de Python que hace de interfaz entre las peticiones que lleguen al servidor web y la aplicación que corre sobre un framework de Flask.

Para ello, se instala gunicorn en nuestra máquina. Después, se lanza gunicorn indicando que dirección y puerto queremos escuchar (será en el que corre la app de Python) y el módulo y la instancia Flask:

gunicorn –bind 0.0.0.0:5000 <code>:<app>

Sobre esto, una nota importante. La sintaxis <code>:<app> significa :

<code> : es el nombre del modulo de Python (sin la extensión, .py), que contiene la app web (la inicialización de la instancia Flask, los endpoints, las lógicas, etc).

<app>: es la propia instancia Flask dentro del módulo <code>, o bien, una función dentro de <code> cuyo *return* es la propia instancia Flask.

Además, en este caso, el módulo <code> no necesita ejecutar Flask(\_\_name\_\_).run() para lanzar la instancia de Flask ya que, este paso, lo realiza internamente Gunicorn. Es decir, si tienes esa línea en tu módulo, coméntala.

De alguna manera, Gunicorn solo necesita saber la dirección a la que apuntar, qué módulo contiene la app, y cuál es la instancia creada de Flask, para ser él quien la sirva.

### Configurar app como servicio

De forma que nuestra app de gunicorn se ejecute en segundo plano y siempre que la máquina esté encendida, tendremos que declarar un servicio.

Para ello, creamos un nuevo servicio en /etc/systemd/system, en el que indicamos el directorio de trabajo y la ejecución que queremos realizar (que será la línea de comandos en el punto anterior).

Tras esto, habilitar y lanzar el servicio, normalmente.

### Configurar Nginx como servidor web

Una vez que Gunicorn está listo, debemos configurar el servidor web que hará de interfaz entre nuestra máquina e Internet.

Para ello, lo primero es instalar (si no se tiene ya), Nginx.

Seguidamente, hay que crear el fichero de configuración de nuestra app web dentro de /etc/nginx/sites-available, con el nombre que se quiera.

Dentro de él, se definen los bloques *server*. Por lo general, se define uno para HTTP y otro para HTTPS. Un ejemplo es el siguiente:

server{

listen 5000 ssl;

server\_name <ip\_pública\_del\_servidor o dominio>

ssl\_certificate /etc/nginx/ssl/selfsigned.crt ;

ssl\_certificate\_key /etc/nginx/ssl/selfsigned.key;

ssl\_dhparam /etc/nginx/ssl/dhparam.pem;

location / {

proxy\_pass [http://unix:/<dirección\_del\_código\_de\_la\_app>/<code>.sock](http://unix:/%3cdirección_del_código_de_la_app%3e/%3ccode%3e.sock);

include /etc/nginx/proxy\_params;

}

}

En el código anterior se ha declarado que Nginx escuche las peticiones en el puerto 5000, estableciendo una conexión segura a través de HTTPS con certificado ssl (autogenerado), y que reenvíe las peticiones al *socket* de Unix donde está corriendo la app de Python sobre Gunicorn.

En el include *proxy\_params*, se indican ciertos atributos del cliente de la petición HTTPS que se envían a la propia app. Este fichero puede editarse y se encuentra dentro de /etc/nginx.

Para generar el certificado autofirmado de SSL (al que se hace referencia en el código anterior):

* Crear la carpeta ssl dentro de nginx:

sudo mkdir -p /etc/nginx/ssl

* Crear un nuevo certificado y llave:

sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/nginx/ssl/selfsigned.key -out /etc/nginx/ssl/selfsigned.crt

Durante este proceso se pedirán ciertos datos de identificación, como empresa, email o localización.

* Crear un fichero *dhparam* para mayor seguridad:

sudo openssl dhparam -out /etc/nginx/ssl/dhparam.pem 2048

Tras todo esto, se hace un link simbólico del fichero en *sites-available* a *sites-enabled*:

sudo ln -s /etc/nginx/sites-available/<app\_web\_config> /etc/nginx/sites-enabled

Tras esto, se chequea que toda la sintaxis dentro de Nginx es correcta, se reinicia el servicio de Nginx, se habilita en el Firewall el puerto en el que Nginx está escuchando, y con todo ello la app debería de ser accesible a través de la IP pública y el puerto configurado a través de una conexión https:

sudo nginx -t

sudo ufw allow 5000

sudo ufw allow ‘Nginx Full’

sudo systemctl restart nginx.service