# cComo subir una aplicación flask a un servidor Ubuntu

La aplicación probablemente tiene múltiples dependencias(por lo menos tiene flask como dependencia) es necesario tener un .txt con la lista de las dependencias para luego instalarlas en el servidor. Si se esta usando un entorno virtual en Windows

(en caso de no estar usando un entorno virtual saltar hasta pip freeze)

**Paso I Escribir las dependencias en un txt**

**Paso opcional entorno virtual**

-Ejecutar cmd como administrador

-ir a la carpeta de la aplicación

-una vez en la misma carpeta que el entorno virtual usar el comando

“venv\Scripts\activate”

Ej:



Ahora aparecerá entre paréntesis el nombre del entorno virtual

Ej:



Una vez en el entorno usar <------------ (Saltar aquí si no hay entorno en el desarrollo)

“pip freeze > requirements.txt”

Para escribir todas las dependencias de pip a un archivo de text con el nombre requirements localizado en la carpeta actual

**Paso II Subir la aplicación al servidor**

Suponiendo que se tiene un servidor Ubuntu(probablemente podría ser Debian sin cambiar ningún paso) se puede acceder al servidor desde la línea de comandos son el comando ssh(Secure Shell). Se ingresa de la siguiente manera

“ssh [usuario]@[ip o dominio]”

El usuario es el usuario del servidor.

Para encontrar la ip se puede encontrar al usar el comando ip a. Imprimira información en la consola, entiendo que esta depende de la tarjeta que tenga para el internet. Ej

1: (…)

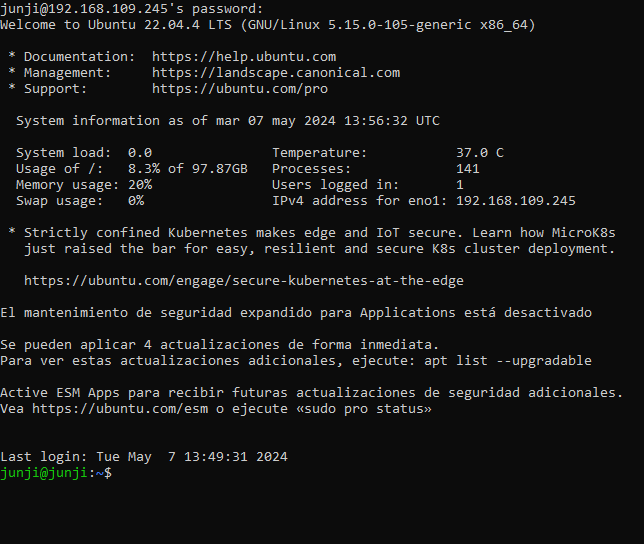
Inet 127.0.0.1/8 //Ignorar esto. Es la ip de localhost

2 eno1 (…)

Inet 192.168.109.245/24 //Esta es la ip quitando el /24. Esta es la mascara y no se como funciona

Tener en cuenta que esta es la ip para la red local, por lo que solo se permite hacer ssh desde la misma red (wifi o ethernet)

Una vez ya se ingreso al servidor a través de ssh se debería estar en el directorio (usuario):~/



Para salir del servidor se ingresa el comando *logout*.

Ahora después de salir del servidor vamos a usar el comando scp para mover archivos desde un computador hasta el servidor a través de ssh

Use el siguiente comando

“scp -r [Origen] [Destino]”

Donde -r significa que es recursivo es decir que si selecciono una carpeta como origen se enviaran todos los archivos y carpetas en esa carpeta. El origen debería ser la carpeta que contenga la aplicación a mover y destino debería ser la cuenta de usuario. Ej

“scp -r cumpleannos-main [junji@192.168.109.245:~](mailto:junji@192.168.109.245:~)”

Importante el destino tiene que ser [usuario]@[ip o dominio]:~

La :~ indica que es un directorio en el servidor, de lo contrario se creara una carpeta llamada [usuario]@[ip o dominio].

Texto

Descripción generada automáticamente

//dir muestra una lista de archivos y carpetas en Windows

**Paso III Usar terminal de Linux**

**Comandos útiles para navegar la terminal**

**“sudo” //al ponerlo detrás de cualquier comando este se ejecuta como administrador**

**“cat file” //muestra el contenido de file en la terminal**

**“cd dir” //cambia el directorio actual a dir**

**“ls” //muestra una lista de todos los archivos y directorios de la carpeta actual**

**“vim file” //abre el archivo en el editor de texto vim para poder editarlo(para poder editarlo puede ser necesario permisos de administrador**

**“apt-get install” para instalar programas de linux (necesita sudo)**

**“apt-get remove” para desinstalar (necesita sudo)**

**Paso IV Instalar la aplicación en el servidor**

Para instalar la aplicación primero es necesario instalar todas las dependencias, para esto es recomendable crear un entorno virtual. Para realizar un entorno virtual en un sistema operativo Linux:

pip install virtualenv (es posible que te pida instalarlo a través de aptget)

se crea el entorno con:

python3 -m venv [nombre del entorno]

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

Para abrir el entorno se usa:

“source [nombre del entorno]/bin/actívate”

Texto

Descripción generada automáticamente

Ahora ir al directorio donde este la aplicación(debería estar en :~)

usar pip freeze < requirements.txt

Mas información sobre venv:

<https://aaronlelevier.github.io/virtualenv-cheatsheet/>

**Paso V Detalles que faltan para que funcione(en caso de que de estos problemas)**

Es posible que algunos paquetes tengan problemas para instalarse especialmente mysql. Probablemente sea necesario instalar dependencias manualmente.

Otro problema es que en Linux no es posible ingresar credenciales de root directamente. Por lo que será necesario editar tu archivo de configuración de la base de datos Ej:

“db.py” aquí será necesario remplazar root por tu nombre de usuario y la contraseña por la contraseña de tu cuenta de usuario. Luego de esto será necesario instalar mysql(apt-get) y crear una cuenta con tus mismas credenciales que tu cuenta de usuario.

Primero iniciar mysql escribiendo sudo mysql en la terminal. Solo se tiene acceso desde root en este momento por lo que solo se puede acceder con sudo

Luego usar:

CREATE USER [usuario]@'localhost' IDENTIFIED BY [contraseña];

Donde usuario y contraseña son el usuario y contraseña de tu usuario en el servidor

Texto

Descripción generada automáticamente

Luego darle privilegios con:

GRANT ALL PRIVILEGES ON \* . \* TO [usuario]@localhost

Y refrescar los privilegios con

FLUSH PRIVILEGES;

Texto

Descripción generada automáticamente

Para ver los usuarios usa:

SELECT User, Host FROM mysql.user;

Texto

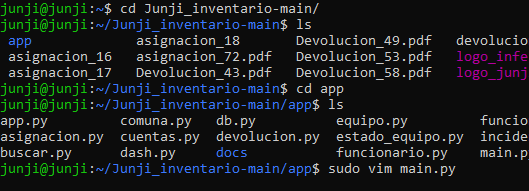
Descripción generada automáticamente

Para salir de mysql usar ctrl + z.

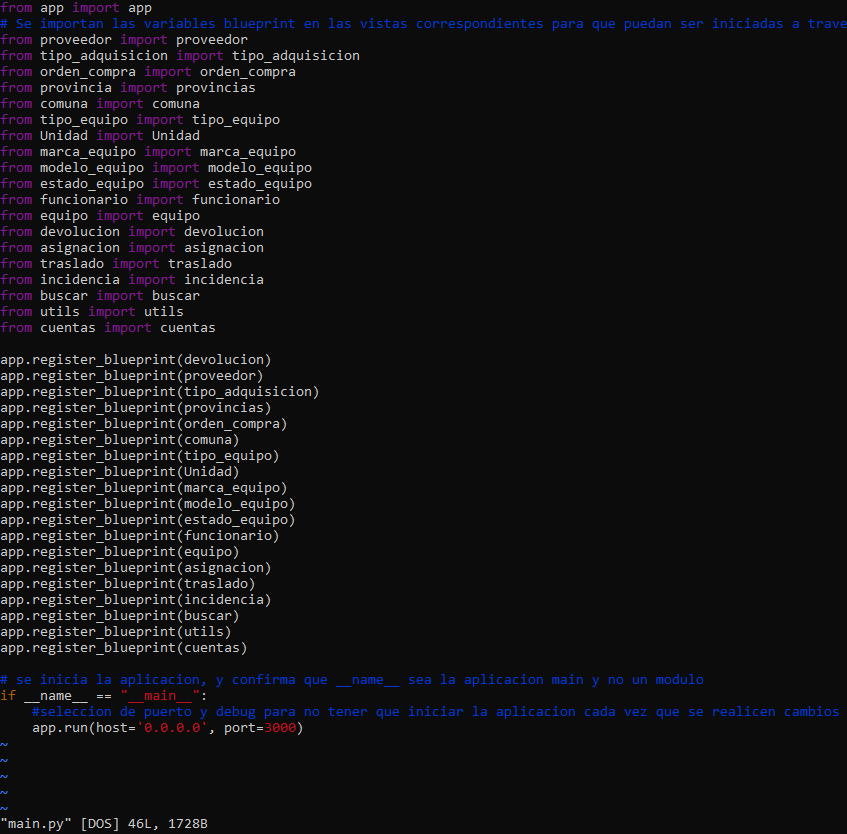
Por ultimo es necesario cambiar el contenido del archivo main.py para que corra la aplicación como servidor.

Ir a la carpeta donde este el código de la aplicación y editar el archivo de entrada de la aplicación con:

“sudo vim [nombre\_archivo].py”



Dentro de vim se usa ‘j’,’k’ para moverse verticalmente y ‘h’,’l’ para moverse horizontalmente. El editor de texto funciona a base de modos por defecto esta en modo normal que se usa para navegar tampien se pueden borrar líneas con ‘dd’, copiar con ‘yy’ y pegar con ‘p’. Para escribir se pasa a modo insert con ‘i’ o ‘a’. Para volver al modo normal se puede usar [Esc]. Para borrar un carácter desde el modo normal se usa ‘x’. Para salir de vim se usa desde el modo normal ‘shift + zz’ o lo que es lo mismo ‘ZZ’. En caso de emergencia se puede forzar la salida presionando desde el modo normal ‘:’ para abrir la línea de comandos integrada y ‘q!’ para forzar la salida.

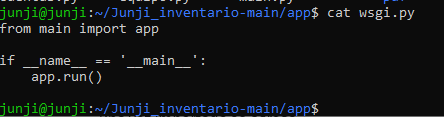


Ir a la linea que selecciona el puerto (app.run(port=3000, debug=True)cambiarlo a app.run(host=0.0.0.0, port=3000)

En este punto deberías poder correr la aplicación con python y acceder desde el equipo en desde el cual accediste con ssh colocando la [ip]:[puerto] en el navegador.

**Paso VI Gunicorn y WSGI**

Primero hay que agregar un archivo wsgi(Web Server Gateway Interface) este será el punto de entrada de la aplicación para el servidor. Luego instalamos Gunicorn que será el punto medio entre la aplicación y nginx. Este debe estar en el directorio de



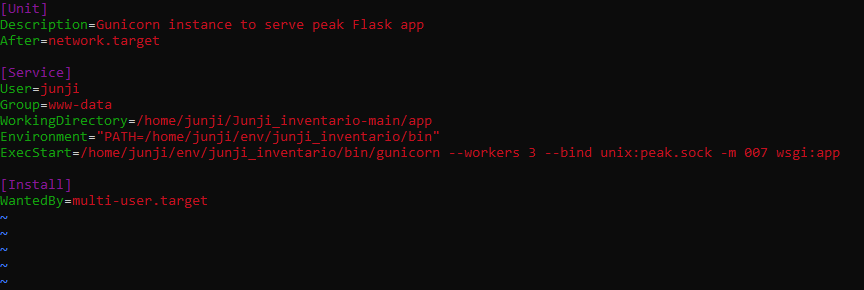
instalar gunicorn(en apt)

usar el siguiente comando gunicorn –bind 0.0.0.0:5000 wsgi:app (desde el directorio donde creaste wsgi.py)

abrir el siguiente archivo

sudo vim /etc/systemd/system/peak.service

copiar lo siguiente al archivo



Activar el archivo

Sudo systemctl start peak (peak es el nombre de del archivo que se creo arriba y también es el nombre del puerto que se crea en ese archivo)

Si no funciona

Sudo systemctl daemon reload

Repetir el start

sudo systemctl enable peak

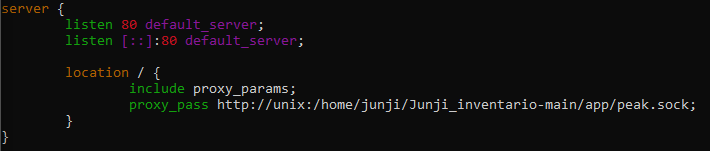
sudo systemctl status( muestra si esta funcionando bien)

Instalar nginx

Sudo apt install nginx

Crear archivo de configuración para el servidor nginx

“sudo vim /etc/nginx/sites-available/peak.sock”



Crear un link de sites avalilable a sites enabled(solo sites enabled importa realmente)

“sudo ln -s /etc/nginx/sites-available/peak.conf /etc/nginx/sites-enabled/”

Esto crea una copia del archive peak.sock a sites enabled y cuando uno de los dos se edite el otro también se edita.

Para comprobar que la configuración funciona

“sudo nginx -t”

Si funciona se puede iniciar el servicio con

sudo systemctl restart nginx

Revisar la firewall del servidor con sudo ufw status y verificar que los puertos estén funcionando. Si no esta funcionando correctamente agregar

“sudo ufw allow “Nginx Full””

Ahora debería funcionar la aplicación en el navegador ingresando la ip. Incluso cuando se reinicie el servidor

**Parte VII Cambiar la ip a una ip fija**

Editar el archivo /etc/netplan/00-installer-config.yaml

Texto

Descripción generada automáticamente

Donde la nueva id es la que sale debajo de addresses:

La que sale en gateway4: 192.168.109.1 es la ip del router.

“sudo neetplan apply”

Ahora deberia funcionar correctamente.

crear archivo wsgi.py en el directorio de la aplicacion. En este se importa app desde main.py

(o cualquiera sea la entrada de la aplicacion)

una vez importada se agrega

from main import app

if \_\_name\_\_ = '\_\_main\_\_':

app.run()

bind:

ingresar siguiente comando

gunicorn --bind 0.0.0.0:5000 wsgi:app.

Esto asocia el localhost:5000(cambiar puerto si la aplicacion usa otro puerto ej:3000) a

crear servicio en /etc/systemd/system/nombreservicio.service

escribir el entorno asociado a al entorno de python y el execStart a la ruta de wsgi

Instalar Nginx

nginx config

Una vez instalado nginx. ir a la carpeta /etc/nginx

aqui deberia entre varias cosas haber dos carpetas llamadas

sites-available y sites-enabled

en sites enabled es donde nginx busca para usar los programas

aqui habra que crear archivos [nombre del sitio].conf ej: para subir una pagina se crea un .conf

server {

#estas dos lineas son para que se ingrese con la ip y no con un dominio.

listen 80 default\_server;

listen [::]:80 default\_server;

location / {

include proxy\_params

#esta es la ruta

proxy\_pass http://unix:/home/junji/Junji\_inventario-main/app/peak.sock;

}

}

#mysql:

#usuario: junji

#contraseña: 1234

al acceder a mysql desde el codigo en linux no se puede acceder directamente a root. Es nesesario crear una cuenta de mysql con

mismo nombre y contraseña que el usuario de linux que sea usado por el usuario. Se accede a el usuario desde el codigo con

las credenciales de la cuenta de escritorio.

tablas, columnas y base de datos son sensitivas a las mayusculas por lo que hay que refactorizar

fuente: https://www.youtube.com/watch?v=KWIIPKbdxD0