«Cómo instalar Python 3 y configurar un entorno de programación en un servidor Ubuntu 18.04» por Lisa Tagliaferri

- This work is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License.
- **English language**: This article is a translation from English into Spanish, published under license «Attribution-NonCommercial-ShareAlike 4.0 International (CC BY-NC-SA 4.0) », written by Lisa Tagliaferri, published on line by the company for leasing virtual machines DigitalOcean. The tittle is «***How To Install Python 3 and Set Up a Programming Environment on an Ubuntu 18.04 Server***» and we created a copy at Wayback Machine for prevent in future a broken link. This work is licensed under the mencioned license but, of course, in castilian language (AKA *spanish*): «Atribución-NoComercial-CompartirIgual 4.0 Internacional (CC BY-NC-SA 4.0) ».
-

- Esta obra está bajo una Licencia Creative Commons Atribución-NoComercial-CompartirIgual 4.0 Internacional.
- **En castellano:** Este artículo es una traducción del inglés al castellano, publicado bajo licencia (en idioma inglés) «Attribution-NonCommercial-ShareAlike 4.0 International (CC BY-NC-SA 4.0) » escrito por Lisa Tagliaferri, publicado en línea por la empresa de alojamiento de máquinas virtuales DigitalOcean. El título original en idioma inglés es «***Cómo instalar Python 3 y configurar un entorno de programación en un servidor Ubuntu 18.04***» y hemos creado una copia en Wayback Machine para prevenir un posible enlace roto a futuro.

Introducción

Python es un lenguaje de programación flexible y versátil que puede ser aprovechado para muchos caso, especialmente en guiones, automatización, análisis de datos, aprendizaje automático y desarrollo en infraestructuras que soportan servicios. Fue publicado por primera vez en 1991 con un nombre inspirado por el grupo de comedia *Monty Python*, ya que el equipo de desarrollo quería hacer de Python un lenguaje de programación que fuera divertido de usar. Rápido para ser configurado y escrito en un estilo relativamente sencillo con retorno inmediato de errores, Python es una gran opción tanto para desarrolladores principiantes y avanzados. Python 3 es la versión más reciente de este lenguaje y es considerado su futura opción.

Este tutorial le permitirá tener configurado un servidor Ubuntu 18.04 con un entorno de programación para Python 3. El programar sobre un servidor tiene muchas ventajas y ofrece apoyo en la colaboración a través de varios proyectos en desarrollo. Los principios generales expuestos en este tutorial son válidos para ser hechos bajo cualquier distribución Linux basada en Debian.

Prerrequisitos

Para completar este tutorial, usted debe tener un usuario que no sea *usuario administrador o usuario raíz* pero con derechos de usuario raíz (derecho para usar comando **sudo**) en un servidor Ubuntu 18.04. Para aprender cómo lograr este acometido, por favor siga los pasos expuestos en nuestra guía de configuración de inicio manual o ejecute nuestro guion automatizado.

Si usted no está familiarizado con un entorno de línea de comandos, usted puede encontrar de utilidad nuestro artículo «*Una introducción a la Terminal en Linux*» (en idioma inglés) para lograr el

estar mejor orientado en el uso de la terminal.

Paso 1 — Configurando Python 3

Ubuntu 18.04 y otras versiones de Linux basados en Debian vienen con ambas versiones preinstaladas de Python, la 2 y 3. Para asegurarse que todas las versiones se encuentren al día, vamos a actualizar y mejorar el sistema con el comando **apt** , es decir, el ***A**dvanced **P**acking **T**ool* (Herramienta Avanzada de Empaquetados) de Ubuntu:

```
$ sudo apt update
$ sudo apt -y upgrade
```

Una vez el proceso esté completo, podemos verificar el número exacto de la versión Python 3 que tengamos instalado en el sistema por medio de la introducción de:

```
$ python3 -V
```

La letra "V" debe ir en mayúscula, recalco eso. Usted obtendrá una salida por pantalla, en la ventana terminal, que le permitirá conocer la versión exacta. Este número puede cambiar a lo largo del tiempo, pero en todo caso usted podrá ver algo similar a esto:

Salida

```
Python 3.6.7
```

Para administrar los paquetes para Python, vamos a instalar **pip**, una herramienta que permitirá instalar y manejar los paquetes de programación que deseamos usar en nuestros proyectos a desarrollar. Usted puede aprender más acerca de módulos y paquetes por medio de **pip** leyendo «*Cómo importar módulos en Python 3*» (en inglés).

```
$ sudo apt install -y python3-pip
```

Dichos paquetes para Python pueden ser instalados luego con:

$ pip3 install nombre_de_paquete

Aquí nombre_de_paquete puede referirse a cualquier paquete o librería de Python, tal como **Django** para el desarrollo web o **NumPy** para cómputo científico. De tal manera que, si usted quiere instalar N**umPy**, puede hacerlo con el comando **pip3 install numpy** .

Existen unos cuantos paquetes y herramientas de desarrollo más a instalar para asegurarse de que tengamos una robusta configuración para nuestro entorno de desarrollo:

```
$ sudo apt install build-essential libssl-dev libffi-dev python3-dev
```

Una vez Python está puesto a punto con **pip** y las demás herramientas instaladas, podremos configurar un Entorno Virtual de Desarrollo para desarrollar nuestros proyectos.

Paso 2 — Configurando un Entorno Virtual de Desarrollo

Con los Entornos Virtuales de Desarrollo habilitados usted tiene un espacio aislado en su servidor para sus proyectos en Python, asegurándose de que cada uno tenga su propio conjunto de dependencias que no irrumpirán con algún otro de sus proyectos.

Configurar un entorno de programación nos proporciona un mayor control sobre nuestro trabajo con Python y sobre las muy diferentes versiones de paquetes que maneja. Esto es especialmente útil cuando cuando se trabaja con paquetes de terceros.

Usted puede configurar tantos entornos de programación como usted quiera. Cada ambiente es básicamente un directorio o carpeta en su servidor que tiene unos cuantos guiones que actúan y conforman un ambiente virtual.

Mientras hay unas cuantas maneras de lograr un entorno de programación en Python, nosotros acá usaremos el módulo **venv**, el cual es parte de la librería estándar de Python 3. Vamos a instalar **venv** tecleando lo siguiente:

```
$ sudo apt install -y python3-venv
```

Cuando esté instalado, estaremos ya listos para crear entornos. Vamos a elegir en cuál directorio queremos colocar nuestro ambiente de programación, o crear una nueva carpeta con **mkdir** de la siguiente manera:

```
$ mkdir entornos
```

Una vez estemos en el directorio donde alojaremos nuestros entornos, usted puede crear un entorno ejecutando el siguiente comando:[*(Ver nota del traductor al final de este artículo)*]

```
$ python3.6 -m venv mi_entorno
```

Esencialmente, **pyvenv** configura un nuevo directorio que contiene unos cuantos renglones que podemos visualizar con el comando **ls**:

```
# ls mi_entorno
```

Salida:

```
bin include lib lib64 pyvenv.cfg share
```

Trabajando juntos todos esos ficheros aseguran que vuestros proyectos están aislados, no van más allá de los límites de vuestro ordenador local, de tal manera que los archivos de sistema y los archivos del proyecto no se mezclan.. Esta es una buena práctica para el control de versiones y para asegurarse que cada de vuestros proyectos tenga acceso a los paquetes particulares que ellos necesitan. **Python Wheels**, un formato de construcción de paquetes que puede acelerar vuestro software en producción por medio de reducir la cantidad de veces necesarias para compilar, estará en el directorio para compartir, denominado **share.**

Para usar este entorno, usted necesita activarlo primero, lo cual puede lograr por medio del tecleo del siguiente comando, el cual llama un guion denominado **activate**:

```
# source mi_entorno/bin/activate
```

Vuestro indicador de sistema tendrá ahora como prefijo el nombre de vuestro entorno, en este caso lo hemos llamado mi_entorno. Dependiendo de cuál versión de Linux Debian usted esté ejecutando, su prefijo puede aparecer algo ligeramente diferente, pero el nombre del entorno en paréntesis debería ser la primera cosa a ver en vuestra línea:

```
(my_env) sammy@ubuntu:~/environments$
```

Este prefijo nos permite saber que el entorno denominado mi_entorno está actualmente activo, lo cual significa que cuando creamos los programas acá ellos usarán solamente este ambiente en particular con sus ajustes y paquetes correspondientes.

> **Nota:** dentro de este entorno virtual usted puede simplemente utilizar los comandos **python** y **pip** en vez de **python3** y **pip3**, si lo prefiere. Si usted usa Python 3 en su máquina y fuera de un entorno virtual, necesitará usar exclusivamente los comandos **python3** y **pip3**.

Después de todos estos pasos debidamente realizados, su entorno virtual está listo para su uso.

Paso 3 — Creando un programa «Hola, mundo»

Ahora que tenemos nuestro entorno virtual completamente configurado, vamos a crear un tradicional programa tipo «Hola, mundo». Esto permitirá comprobar nuestro ambiente y nos brinda la excelente oportunidad de tomar familiaridad con Python si es que aún no estamos listos para ello.

Para realizar esto, abriremos en una ventana de comandos un editor de texto tal como **nano** y crearemos un fichero nuevo:

```
(mi_entorno) sammy@ubuntu:~/entornos$ nano hola.py
```

Al desplegarse el editor de texto en la ventana, teclearemos lo siguiente a nuestro programa:

```
print("¡Hola, mundo!")
```

Salga de **nano** pulsando CTRL + X, respondiendo que sí Y y luego Intro para guardar el fichero.

Una vez haya usted abandonado **nano** y de regreso a la línea de comandos de su sistema operativo, procedemos a correr el programa:

```
(mi_entorno) sammy@ubuntu:~/entornos$ python hola.py
```

```
Salida:
¡Hola, mundo!
```

Para salir del entorno virtual, simplemente teclee el comando **deactivate** y regresará así al directorio original.

Conclusión

¡Felicitaciones! En este momento usted ya tiene configurado un entorno de programación para Python 3 en vuestro servidor Linux Debian ¡y puede, desde ya, comenzar a codificar un proyecto!

Si usted está utilizando una máquina local en vez de un servidor, vaya al siguiente enlace «Cómo instalar y configurar un Entorno de Programacaión para Python 3» (en inglés), el cual contiene una serie de enlaces y ubique el sistema operativo que usa su máquina local.

Con vuestro servidor listo para el desarrollo de software, puede continuar el aprendizaje de programación bajo Python con nuestro libro electrónico «Cómo Programar en Python 3» (en inglés), sin costo alguno o también puede visitar nuestros proyectos (también en inglés) dedicados a ello.

- This work is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License.
- **English language**: This article is a translation from English into Spanish, published under license «Attribution-NonCommercial-ShareAlike 4.0 International (CC BY-NC-SA 4.0) », written by Lisa Tagliaferri, published on line by the company for leasing virtual machines DigitalOcean. The tittle is «***How To Install Python 3 and Set Up a Programming Environment on an Ubuntu 18.04 Server***» and we created a copy at Wayback Machine for prevent in future a broken link. This work is licensed under the mencioned license but, of course, in castilian language (AKA *spanish*): «Atribución-NoComercial-CompartirIgual 4.0 Internacional (CC BY-NC-SA 4.0) ».
-

- Esta obra está bajo una Licencia Creative Commons Atribución-NoComercial-CompartirIgual 4.0 Internacional.
- **En castellano:** Este artículo es una traducción del inglés al castellano, publicado bajo licencia (en idioma inglés) «Attribution-NonCommercial-ShareAlike 4.0 International (CC BY-NC-SA 4.0) » escrito por Lisa Tagliaferri, publicado en línea por la empresa de alojamiento de máquinas virtuales DigitalOcean. El título original en idioma inglés es «***Cómo instalar Python 3 y configurar un entorno de programación en un servidor Ubuntu 18.04***» y hemos creado una copia en Wayback Machine para prevenir un posible enlace roto a futuro.

Notas

- En el ejemplo utilizamos «python3.6 -m venv mi_entorno» para usar la versión 3.6 de Python pero tal vez necesitemos una versión diferente, por ejemplo 3.7 : para ello debemos instalar «sudo apt install python3.7-minimal» y así sucesivamente.