

Instalar y usar virtualenv con Python 3

Generalidades

Virtualenv es una herramienta utilizada para crear un entorno Python aislado. Este entorno tiene sus propios directorios de instalación que no comparten bibliotecas con otros entornos virtualenv o las bibliotecas instaladas globalmente en el servidor.

Virtualenv es la manera más fácil recomendada para configurar un ambiente personalizado Python.



Para correr los siguientes comandos, debes iniciar sesión en el servidor vía SSH con tu usuario Shell. Ve los siguientes artículos para más información.

- [Generalidades de SSH](#)
- [Cómo crear un usuario con acceso Shell \(SSH\)](#)

Diferencia entre virtualenv y venv

venv es un paquete que viene con Python 3. Python 2 no contiene venv.

virtualenv es una biblioteca que ofrece más funcionalidades que venv. Ve el siguiente enlace para obtener una lista de características que venv no ofrece en comparación con virtualenv.

- <https://virtualenv.pypa.io/en/stable/>

Aunque puedes crear un entorno virtual usando venv con Python3, se recomienda que instales y use virtualenv en su lugar.

Instalar Virtualenv usando pip3

Virtualenv es instalado por defecto en todos los servidores DreamHost para las versiones de Python 2. Si estás trabajando con Python 3, debes instalar virtualenv usando `pip3`.

Quizá quieras actualizar primero a `pip3`.

```
[server]$ python3 -m pip install --upgrade pip
```

Estas instrucciones asumen que tu ya has instalado una [versión personalizada de Python 3](#). Después de la instalación y si tu Shell está usando esta versión, corre `pip3` para instalar virtualenv:

```
[server]$ pip3 install virtualenv
Collecting virtualenv
  Downloading virtualenv-15.1.0-py2.py3-none-any.whl (1.8MB)
    100% |████████████████████████████████████████| 1.8MB 367kB/s
Installing collected packages: virtualenv
Successfully installed virtualenv-15.1.0
```

Necesitarás el comando completo para el virtualenv de la versión de Python, corre entonces lo siguiente para verlo.

```
[server]$ which virtualenv
/home/username/opt/python-3.6.2/bin/virtualenv
```

Crear un ambiente virtual usando una versión personalizada de Python

Cuando trabajas con un ambiente virtual en Python, es común usar una [versión personalizada de Python 3](#) en vez de la versión del servidor. Para crear un nuevo ambiente virtual usando tu versión personalizada instalada, sigue los siguientes pasos:



Los siguientes pasos usando las versión 3.6.2 de Python. Asegúrate de usar la versión que tienes instalada.

1. Toma nota del camino que toma el archivo para la versión personalizada de Python ya instalada. Si has seguido las instrucciones del [artículo de instalación](#), el camino es:

```
[server]$ which python3
/home/username/opt/python-3.6.2/bin/python
```

2. Navega al [directorio de tu sitio](#), donde crearás el nuevo ambiente virtual:

```
[server]$ cd ~/example.com
```

3. Corre el siguiente comando para actualizar tu archivo **.bash_profile**:

```
[server]$ . ~/.bash_profile
```

4. Crea un ambiente virtual mientras especificas la versión de Python que deseas usar. El siguiente comando crea un virtualenv llamado 'venv' y usa una bandera `-p` para especificar el camino a la versión de Python 3 que acabas de instalar.



Puedes llamar el virtualenv como tu quieras.

```
[server]$ virtualenv -p /home/username/opt/python-3.6.2/bin/python3 venv

Running virtualenv with interpreter /home/username/opt/python-3.6.2/bin/py
Using base prefix '/home/username/opt/python-3.6.2'
New python executable in /home/username/example.com/env/bin/python3
Also creating executable in /home/username/example.com/env/bin/python
Installing setuptools, pip, wheel...done.
```

- Este comando crea una copia local en tu ambiente específico para este sitio web. Mientras trabajas en tu sitio web, deberás activar el ambiente local para asegurarte que estás trabajando con la versión correcta de tus herramientas y paquetes.



Es posible que veas el siguiente error al instalar.

```
setuptools pip failed with error code 1` error
```

Si es así, corre lo siguiente:

```
[user@localhost]$ pip3 install --upgrade setuptools
```

Intenta nuevamente y deberías poder instalar sin un error.

5. Para activar el nuevo ambiente virtual, corre lo siguiente:

```
[server]$ source venv/bin/activate
```

El nombre del actual ambiente virtual aparece a la izquierda del intento. Por ejemplo:

```
(venv) [server]$
```

6. Para verificar la versión correcta de Python, corre lo siguiente:

```
[server]$ python -V
Python 3.6.2
```

Cualquier paquete que instalas usando `pip` está ahora ubicado en la carpeta de proyectos de ambientes virtuales, aislado de la instalación global de Python.

Desactivar tu virtualenv

Cuando finalices tu trabajo en tu ambiente virtual, puedes desactivarlo corriendo lo siguiente:

```
[server]$ deactivate
```

- Esto te pone nuevamente en las configuraciones por defecto de tu usuario de Shell.

Eliminar tu ambiente virtual

Para eliminar tu ambiente virtual, simplemente elimina la carpeta del proyecto. Usando el ejemplo anterior, corre el siguiente comando:

```
[server]$ rm -rf venv
```

Instalar módulos personalizados

Ve el siguiente artículo para ver información de como usar `pip` para instalar módulos de Python.

- [Usar pip para instalar módulos de Python 3](#)

Solución de Problemas

Errores creando un virtualenv

Puedes ver los siguientes errores cuando trates de crear un virtualenv usando Python 3.7.

AttributeError: module 'importlib._bootstrap' has no attribute 'SourceFileLoader'

OSError: Command /home/username/venv/bin/python3 -c "import sys, pip; sys...d\" + sys.argv[1:]))" setuptools pip failed with error code 1

Agregar la siguientes líneas cuando [instales una versión personalizada de OpenSSL](#) a tu **.bash_profile** para que resuelva eso.

```
export LC_ALL="en_US.UTF-8"
```

Utiliza la ruta completa a tu virtualenv personalizado

También es posible que cuando ejecutes el comando virtualenv, esté usando una versión fuera de tu instalación personalizada. Intenta correr esto en su lugar para confirmar la ruta completa a su virtualenv de Python3.

```
[server]$ which virtualenv
/home/username/opt/python-3.8.0/bin/virtualenv
```

Esto debería responder con la versión en tu directorio personalizado de Python3. Luego puedes crear un virtualenv usando la ruta completa como esta:

```
[server]$ /home/username/opt/python-3.8.0/bin/virtualenv -p /home/username/opt
```