



Despliegue de Django en AWS con Nginx

Christian Escalante - CEO & Founder

Paso 1: AMAZON WEB SERVICES

Paso 1 - Amazon Web Services

Amazon Web Service

a. Crearemos la instancia en Amazon Web Services.

b. Posteriormente instalaremos las dependencias necesarias para nuestro proyecto.

Proyecto en Django

```
$ sudo apt-get install -y nano postgresql-9.5  
postgresql-contrib-9.5 postgresql-server-dev-9.5 nginx git  
circus make build-essential libssl-dev zlib1g-dev  
libbz2-dev libreadline-dev libsqlite3-dev wget curl llvm  
libncurses5-dev libncursesw5-dev xz-utils python-setuptools
```

Configurando Nginx

Paso 1 - Amazon Web Services

Amazon Web Service

c. Crearemos un usuario para el proyecto.

```
$ sudo adduser <app_name>
```

Proyecto en Django

d. Bloqueamos el password del usuario.

```
$ sudo passwd -l <app_name>
```

Configurando Nginx

d. Añadimos www-data al usuario para Nginx.

```
$ sudo adduser www-data <app_name>
```

Paso 1 - Amazon Web Services

Amazon Web Service

e. Ingresamos en la cuenta del usuario creado.

```
$ sudo -u <app_name> -i
```

Proyecto en Django

f. Instalamos pyenv.

```
$ git clone https://github.com/yyuu/pyenv.git ~/.pyenv
```

Configurando Nginx

g. Editamos .bash_profile para que funcione pyenv

```
$ nano ~/.bash_profile
```

Paso 1 - Amazon Web Services

Amazon Web Service

Proyecto en Django

Configurando Nginx

h. Guardamos las siguientes líneas.

```
export PYENV_ROOT="$HOME/.pyenv"  
export PATH="$PYENV_ROOT/bin:$PATH"  
eval "$(pyenv init -)"
```

i. Salimos y volvemos a entrar.

```
$ exit
```

```
$ sudo -u <app_name> -i
```

Paso 1 - Amazon Web Services

Amazon Web Service

j. Ejecutamos pyenv para ver que funciona, luego instalamos Python.

```
$ pyenv  
$ pyenv install 3.5.2
```

Proyecto en Django

i. Verificamos si tenemos la versión instalada y si está seleccionada.

```
$ pyenv versions  
$ pyenv local 3.5.2
```

Configurando Nginx



<http://www.indestim.es>

info@indestim.es



@indestim



/indestim