

# Deploy Django

---

Ubuntu 21.04 - Apache - Django - MySQL

Learn Full Stack Development  
[www.projectfullstack.com](http://www.projectfullstack.com)

# Step 1: Setup Server

- Using Digital Ocean as a VPS provider:
  - Use this link (also in description) for \$100 in Digital Ocean credits
    - <https://m.do.co/c/e30460a6fcfc>
  - Register / Login
  - Create a new droplet
    - Distribution: 21.04
    - Size: \$5/month
      - 1 GB RAM
      - 1 CPU
      - 25GB SSD Disk
    - Datacenter: The closest one to you
    - Hostname: An easily identifiable name













# Step 1: Setup Server

- Distribution

## Create Droplets

Choose an image ?

Distributions Container distributions Marketplace Backups Custom images

 Ubuntu 21.04 x64 	 FreeBSD Select version 	 Fedora Select version 	 Debian Select version 	 CentOS Select version 	 Rocky Linux Select version 
---	--	--	--	--	---

# Step 1: Setup Server

- Choose A Plan

Choose a plan

[Help me choose](#)

SHARED CPU	DEDICATED CPU			
Basic	General Purpose	CPU-Optimized	Memory-Optimized	Storage-Optimized <b>NEW</b>

Basic virtual machines with a mix of memory and compute resources. Best for small projects that can handle variable levels of CPU performance, like blogs, web apps and dev/test environments.

CPU options: ☒ Regular Intel with SSD

☐ Premium Intel with NVMe SSD **NEW**

☐ Premium AMD with NVMe SSD **NEW**

<b>\$5</b> /mo \$0.007/hour	<b>\$10</b> /mo \$0.015/hour	<b>\$15</b> /mo \$0.022/hour	<b>\$20</b> /mo \$0.030/hour	<b>\$40</b> /mo \$0.060/hour	<b>\$80</b> /mo \$0.119/hour
1 GB / 1 CPU 25 GB SSD Disk 1000 GB transfer	2 GB / 1 CPU 50 GB SSD Disk 2 TB transfer	2 GB / 2 CPUs 60 GB SSD Disk 3 TB transfer	4 GB / 2 CPUs 80 GB SSD Disk 4 TB transfer	8 GB / 4 CPUs 160 GB SSD Disk 5 TB transfer	16 GB / 8 CPUs 320 GB SSD Disk 6 TB transfer

Learn Full Stack Development


[www.projectfullstack.com](http://www.projectfullstack.com)

[www.patreon.com/projectfullstack](http://www.patreon.com/projectfullstack)

# Step 1: Setup Server

- Datacenter

Choose a datacenter region

<div><div> New York</div><div><div>1</div><div>2</div><div>3</div></div></div>	<div><div> San Francisco</div><div><div>1</div><div>2</div><div>3</div></div></div>	<div><div> Amsterdam</div><div><div>2</div><div>3</div></div></div>	<div><div> Singapore</div><div><div>1</div></div></div>	<div><div> London</div><div><div>1</div></div></div>	<div><div> Frankfurt</div><div><div>1</div></div></div>
<div><div> Toronto</div><div><div>1</div></div></div>	<div><div> Bangalore</div><div><div>1</div></div></div>				

# Step 1: Setup Server

- Authentication - NOTE: I will use a root password for this tutorial. I HIGHLY recommend you use SSH keys for your production level deployments

## Authentication ?



### SSH keys

A more secure authentication method



### Password

Create a root password to access Droplet (less secure)

### Create root password \*

Type your password...

#### PASSWORD REQUIREMENTS

- Must be at least 8 characters long
- Must contain 1 uppercase letter (cannot be first or last character)
- Must contain 1 number
- Cannot end in a number or special character

 Please store your password securely. You will not be sent an email containing the Droplet's details or password.



# Step 1: Setup Server

- Finalize and Create- Make sure to use a descriptive host name. This will make it easier in the future if you have several droplets

## Finalize and create

### How many Droplets?

Deploy multiple Droplets with the same [configuration](#).

—

1 Droplet

+

### Choose a hostname

Give your Droplets an identifying name you will remember them by. Your Droplet name can only contain alphanumeric characters, dashes, and periods.

deploying-django-on-ubuntu21-04-apache-mysql



### Add tags

Use tags to organize and relate resources. Tags may contain letters, numbers, colons, dashes, and underscores.

Type tags here

### Select Project

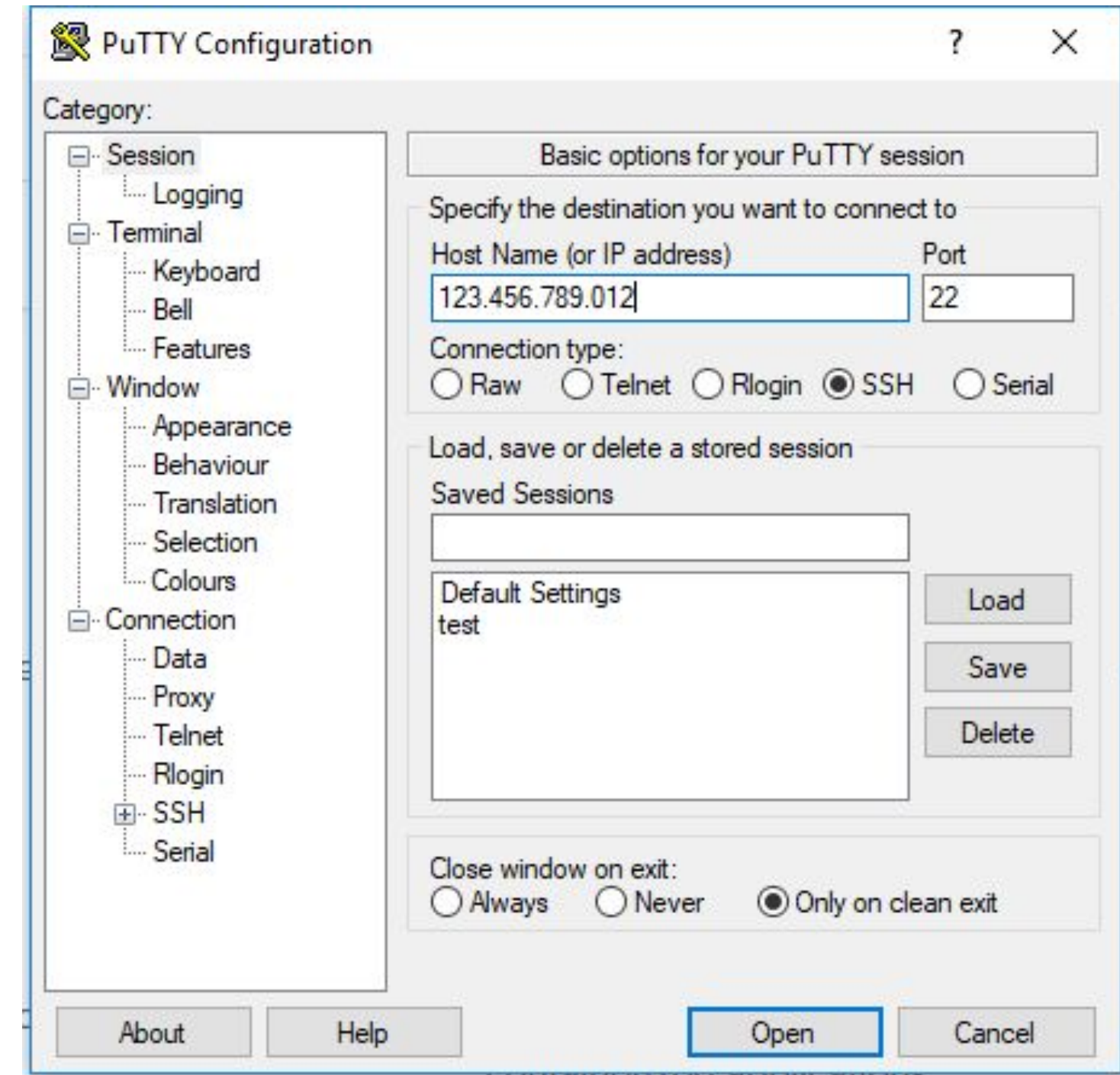
Assign Droplets to a project

 Tutorials

▼

# Step 2: Connect To Server

- Using Linux / Mac open a terminal:
  - SSH
    - `$ ssh root@<your_ip>`
    - Example: `ssh root@123.456.789.012`
    - Select yes to add the server to your known hosts
    - Enter password
- Using Windows:
  - PuTTY
    - Select Session from the left sidebar
    - Enter your ip address in the Host Name input
    - Click the “Open” button
    - Select “Accept” at the popup
    - Enter “root” in the login input
    - Enter password





# Step 2: Connect To Server

- Using Digital Ocean dashboard
  - In digital ocean, select your droplet and go to the “access” tab, then select Launch Droplet Console

**ubuntu-s-1vcpu-1gb-nyc1-01** ON  
in [Tutorials](#) / 1 GB Memory / 25 GB Disk / NYC1 - Ubuntu 21.04 x64

ipv4: 142.93.116.158    ipv6: [Enable now](#)    Private IP: 10.136.0.3    Floating IP: [Enable now](#)    Console: [📄](#) [?](#)

Graphs  
**Access**  
Power  
Volumes  
Resize  
Networking  
Backups  
Snapshots  
Kernel  
History  
Destroy  
Tags  
Recovery

### Droplet Console

Use the Droplet Console for native-like terminal access to your Droplet from your browser. Here is [the list of supported OSes](#) for the new console.

Log in as...  
root

**Launch Droplet Console**

### Recovery Console

Use the Recovery Console if you need to use the recovery ISO or you can't connect to your Droplet with the Droplet Console. To use the recovery console, you must enable password authentication. If necessary, you can reset your root password below.

**Launch Recovery Console**

# Step 3: Apply Software Updates

- Update the system to make sure we have the most up to date software
- \$ apt update && apt upgrade
- \$ sudo reboot ← reboots the system, optional but recommended

```
root@deploying-django-on-ubuntu-21-04-apache-mysql:~# apt update && apt upgrade
```

# Step 4: Create directory structure

- CD to the root of your system
  - \$ cd /
- Create the following directory structure
  - Project-name ( I will use “django-project” for this)
    - src/
    - site/
      - logs/
      - public/
        - media/
        - static/

```
root@ubuntu-s-1vcpu-1gb-nyc1-01:/# tree django-project/  
django-project/  
├── site  
│   ├── logs  
│   ├── public  
│   │   ├── media  
│   │   └── static  
└── src
```

Learn Full Stack Development

[www.projectfullstack.com](http://www.projectfullstack.com)

[www.patreon.com/projectfullstack](http://www.patreon.com/projectfullstack)



# Step 5: Install PIP & Setup VirtualEnv

- PIP is a package manager we use to install python packages
  - Django is a python package!
- Virtualenv is used to manage Python packages for different projects.
  - Using virtualenv allows you to avoid installing Python packages globally which could break system tools or other projects. You can install virtualenv using pip.
- `$ sudo apt install python3-pip`
- `$ sudo pip3 install virtualenv`
- Create a virtualenv
  - Make sure you are in `/project-name/` directory
  - `$ virtualenv venv -p python3`
  - `$ source venv/bin/activate`
  - `$ pip install django==3.2.8`

# Step 6: Create Django Project

- Make sure your venv is activated and django is installed
  - \$ pip freeze
- \$ cd /project-name/src
- \$ django-admin startproject some-project-name . ← don't forget the period at the end!
- Add your server's IP address to settings.py Allowed Host constant
- Run the django development server
  - python manage.py runserver 0.0.0.0:8000
- Open a web browser, goto <your-ip-address>:8000
  - Example: 123.456.789.010:8000
- You should see the default django welcome page



# Step 7: Install MySQL

- \$ sudo apt install mysql-server
- \$ sudo mysql\_secure\_installation
  - Yes to everything!
- Creating a user and a database
  - \$ mysql
  - mysql> CREATE USER 'django user'@'localhost' IDENTIFIED BY 'XLq4zhNFPjfwyHgX';
  - mysql> CREATE DATABASE 'django project';
  - mysql> GRANT ALL PRIVILEGES ON 'django project'.\* to 'django user'@'localhost';
  - mysql> FLUSH PRIVILEGES;

# Step 8: Connect MySQL and Django

- Install “mysqlclient” python package

- \$ sudo apt install python3-dev
- \$ sudo apt install libmysqlclient-dev
- \$ pip install mysqlclient

- Add the following to settings.py

```
DATABASES = {  
    'default': {  
        'ENGINE': 'django.db.backends.mysql',  
        'NAME': 'tutorial',  
        'USER': 'django',  
        'PASSWORD': 'your-password',  
        'HOST': 'localhost, # default is localhost, so this can be omitted  
        'PORT': '3306' # default is 3306, so this can be omitted  
    }  
}
```

Learn Full Stack Development

[www.projectfullstack.com](http://www.projectfullstack.com)

[www.patreon.com/projectfullstack](http://www.patreon.com/projectfullstack)

# Step 8: Connect MySQL and Django

- Check django, create superuser, make migrations, runserver
  - \$ python manage.py check
  - \$ python manage.py migrate
  - \$ python manage.py createsuperuser
  - \$ python manage.py runserver 0.0.0.0:8000

In a browser go to your-ip-address:8000/admin and log in with the superuser credentials

# Step 9: Install & Configure Apache2

- \$ sudo apt install apache2 libapache2-mod-wsgi-py3
- Check apache installation
  - In a browser go to your-ip-address
  - You should see the Apache2 default page
- CD to /etc/apache2/sites-available/
  - Change 000-default.conf to the following:
    - See next slide

# Step 9: Install & Configure Apache2

- CD to /etc/apache2/sites-available/
  - Change 000-default.conf to the following:

```
<VirtualHost *:80>
```

```
ErrorLog /django-project/site/logs/error.log
CustomLog /django-project/site/access.log combine
```

```
alias /static /django-project/site/public/static
<Directory /django-project/site/public/static>
    Require all granted
</Directory>
```

```
<Directory /django-project/src/tutorial>
    <Files wsgi.py>
        Require all granted
    </Files>
</Directory>
```

```
WSGIDaemonProcess tutorial python-home=/django-project/venv python-path=django-project/src/
WSGIProcessGroup tutorial
WSGIScriptAlias / /django-project/src/tutorial/wsgi.py
```

```
</VirtualHost>
```



# Step 9: Install & Configure Apache2

- Make sure your syntax is correct for the 000-default.conf
  - `$ sudo apachectl configtest`
- Restart Apache2 for changes to take effect
  - `$ sudo service apache2 restart`
- In a browser visit your ip address

# Step 10: Configuring Static Files

- In a browser, visit your-ip-address/admin
  - Notice the static files (css/javascript/etc) are not loading
- First we need a place for static files to live on our server
  - `$ mkdir /project_name/site/public/static`
- Add the following to settings.py:
  - `STATIC_ROOT = 'project_name/site/public/static'`
- Now that django knows where to put static files, we have to tell it to do so:
  - `python manage.py collectstatic`
- Finally we have to tell Apache to look in this directory for static files
  - Add the following to 000-default.conf
    - `alias /static /project_name/site/public/static`
    - `<Directory /project_name/site/public/static>`
    - `Require all granted`
    - `</Directory>`
- In a browser, visit your-ip-address/admin. You should now have static files being served