**How to Point Domain and Host Django Project on Apache Web Server VPS Hosting**

* Get Access to Remote Server via SSH

Syntax:- ssh -p PORT USERNAME@HOSTIP

Example:- ssh -p 22 raj@216.32.44.12

#### **Note:- Run Below Commands on Your Remote Server Linux Machine or VPS, Not on Your Local Windows Machine**

* Verify that all required softwares are installed

apache2 -v

python --version

apache2ctl -M

pip --version

- SQLite is Included with Python

python -c "import sqlite3; print(sqlite3.sqlite\_version)"

* If Required Softwares and Modules are not Installed then Install them:

sudo apt install apache2

sudo apt install python

sudo apt install libapache2-mod-wsgi-py3

sudo apt install python3-pip

* Install virtualenv

pip list

sudo pip install virtualenv

* Verify Apache2 is Active and Running

sudo service apache2 status

* Verify Web Server Ports are Open and Allowed through Firewall

sudo ufw status verbose

* Go to Your Project Directory

Syntax:- cd /var/www/project\_folder\_name

Example:- cd /var/www/DoinDenimz\_Backend

* Create Virtual env

Syntax:- virtualenv env\_name

Example:- virtualenv env

* Activate Virtual env

Syntax:- source virtualenv\_name/bin/activate

Example:- source env/bin/activate

* Install Dependencies

pip install -r requirements.txt

## How to Install PostgreSQL On Ubuntu

**Follow the steps in the sections below to install PostgreSQL from the PostgreSQL repository.**

**Step 1: Add PostgreSQL Repository**

Run the following command to add the PostgreSQL repository to your system:

sudo sh -c 'echo "deb https://apt.postgresql.org/pub/repos/apt $(lsb\_release -cs)-pgdg main" > /etc/apt/sources.list.d/pgdg.list'

**Step 2: Add the Repository Signing Key**

The next step is to fetch the repository's GPG key and add it to APT's trusted keyring. This allows APT to verify the authenticity of packages downloaded from the PostgreSQL repository.

Run the command below:

wget --quiet -O - https://www.postgresql.org/media/keys/ACCC4CF8.asc | sudo apt-key add –

**Step 3: Update the Package List**

After adding the official PostgreSQL repository, update the package list to ensure you install the latest PostgreSQL package.

sudo apt update

**Step 4: Install PostgreSQL**

To install **PostgreSQL** run the following command:

sudo apt install postgresql

## **Verify PostgreSQL Installation**

Verify that PostgreSQL has been installed by checking the PostgreSQL service status. Run the command below:

sudo systemctl status postgresql

## **Connect to PostgreSQL**

To establish a connection with the database, log into the **postgres** account with:

sudo -i -u postgres

Next, open a **postgres** prompt using the command:

psql

Use the following syntax to create a postgres password:

\password postgres

You will be prompted to type a new password. Repeat this for the owner and postgres user, giving each a strong, unique password.

To exit psql, type

\q.

Use the following syntax to create a database:

Syntax: CREATE DATABASE [dbname];

Example: CREATE DATABASE doindenimz\_database;

## Check Connection Information

If you are connected to PostgreSQL and want to see details of the connection, use the command:

\conninfo

Use the following syntax to create a user and password:

Syntax: CREATE USER <username> with ENCRYPTED PASSWORD ‘<password>';

Example: CREATE USER doindenimz with ENCRYPTED PASSWORD ‘postgres@123';

Use the following syntax to create a user and password:

Syntax: ALTER DATABASE <dbname> OWNER TO <username>;

ALTER DATABASE doindenimz\_database OWNER TO doindenimz;

To exit psql, type

\q.

Allow PostgreSQL TCP port 5432 in the Firewall

PostgreSQL default HTTP port is 5432, you’ll need to allow access to this port on the firewall.

If your firewall is UFW type the following commands:

sudo ufw allow 5432/tcp

## **Add PostgreSQL Server to pgAdmin**

**Check PostgreSQL Configuration**

1. Verify postgresql.conf:

Locate the postgresql.conf file. Its location can vary, but it's often in

cd /etc/postgresql/16/main/

sudo nano postgresql.conf

listen\_addresses = '\*'

1. Verify pg\_hba.conf:

Locate the pg\_hba.conf file. Its location can vary, but it's often in

cd /etc/postgresql/16/main/

sudo nano pg\_hba.conf

#IPv4 Addresses

host all all 0.0.0.0/0 md5

#IPv6 Addresses

host all all ::0/0 md5

1. Restart PostgreSQL configuration to apply changes:

sudo service restart postgresql

1. Reload PostgreSQL configuration to apply changes:

sudo service reload postgresql

## **Add PostgreSQL Server to pgAdmin**

In pgAdmin, click on the “Servers” menu on the left sidebar, then right-click and choose “Register” then select “Server” Enter a name for your server and switch to the “Connection” tab. Fill in the following details:

**Host name/address:** <ip address>  
**Port:** 5432  
**Maintenance database:** postgres  
**Username:** postgres  
**Password:** (Enter the password you set for the PostgreSQL user)

Click “Save” to add the server.

Congratulations! You’ve successfully installed and configured PostgreSQL and pgAdmin on your Ubuntu system. You can now use pgAdmin to manage your databases, tables, and perform various administrative tasks through its intuitive interface.

* Serve Static Files

cd /var/www/doindenimz

python manage.py collectstatic

* Create Database Tables

python manage.py makemigrations

python manage.py migrate

* Create Superuser

python manage.py createsuperuser

* If Database File throws error Permission Denied then Set below permissions
* Make Webserver as owner for database file. Our Webserver is running as www-data and group is also www-data.

Syntax:-

sudo chown -R www-data:www-data database\_folder

sudo chmod 775 database\_folder

sudo chmod 664 database\_folder/database\_file

Example:-

sudo chown -R www-data:www-data mbdb

sudo chmod 775 mbdb

sudo chmod 664 mbdb/db.sqlite3

* If Media Files (User Uploaded Files) throws error Permission Denied then Set below permissions

sudo chown -R www-data:www-data media

* If needed Deactivate Virtual env

deactivate

* Create Virtual Host File

sudo nano /etc/apache2/sites-available/your\_domain.conf

sudo nano /etc/apache2/sites-available/tailor.conf

* Add Following Code in Virtual Host File

<VirtualHost \*:80>

    ServerName tailorapi.broaderai.com

    ServerAdmin webmaster@localhost

    Alias /static /var/www/DoinDenimz\_Backend/static

    <Directory /var/www/DoinDenimz\_Backend/static>

        Require all granted

    </Directory>

    Alias /media /var/www/DoinDenimz\_Backend/media

    <Directory /var/www/DoinDenimz\_Backend/media>

        Require all granted

    </Directory>

    <Directory /var/www/DoinDenimz\_Backend/tailorecommerce>

        <Files wsgi.py>

            Require all granted

        </Files>

    </Directory>

    WSGIPassAuthorization On

    WSGIDaemonProcess tailor python-path=/var/www/DoinDenimz\_Backend python-home=/var/www/DoinDenimz\_Backend/env

    WSGIProcessGroup tailor

    WSGIScriptAlias / /var/www/DoinDenimz\_Backend/tailorecommerce/wsgi.py

    ErrorLog ${APACHE\_LOG\_DIR}/error.log

    CustomLog ${APACHE\_LOG\_DIR}/access.log combined

RewriteEngine on

RewriteCond %{SERVER\_NAME} =tailorapi.broaderai.com

RewriteRule ^ https://%{SERVER\_NAME}%{REQUEST\_URI} [END,NE,R=permanent]

</VirtualHost>

* Check Configuration is correct or not

sudo apache2ctl configtest

* Enable Virtual Host

cd /etc/apache2/sites-available/

sudo a2ensite tailor.conf

* Restart Apache2

sudo service apache2 restart

* Reload Apache2

sudo service apache2 reload

* If get Error mod\_wsgi (pid=1234): Failed to proxy response from daemon then follow below instructions:
* Open apache2.conf

cd /etc/apache2

sudo nano apache2.conf

* Write below code in the bottom of apache2.conf file

WSGIApplicationGroup %{GLOBAL}

* To Know more about %{GLOBAL} follow this link: <https://modwsgi.readthedocs.io/en/develop/configuration-directives/WSGIApplicationGroup.html>
* Restart Apache2

sudo service apache2 restart

### How to Enable HTTPS in Your Domain Hosted on Linux Remote Server or VPS

#### **Let's Encrypt is a non-profit certificate authority run by Internet Security Research Group that provides X.509 certificates for Transport Layer Security encryption at no charge.**

* To Access Remote Server via SSH

Syntax:- ssh -p PORT USERNAME@HOSTIP

Example:- ssh -p 22 root@216.32.44.12

* Install Certbot and python3-certbot-apache

sudo apt install certbot python3-certbot-apache

- Certbot is a free, open source software tool for automatically using Let’s Encrypt certificates on manually-administrated websites to enable HTTPS.

- python3-certbot-apache is a Apache plugin for Certbot.

* Verify Web Server Ports are Open and Allowed through Firewall

ufw status verbose

* Obtain an SSL certificate

sudo certbot --apache

* Check Status of Certbot

sudo systemctl status certbot.timer

* Dry Run SSL Renewal

sudo certbot renew --dry-run

#### **Special Tip: If you face error "Name duplicates previous WSGI daemon definition" while installing SSL Certificate for your domain then comment below code then try to install SSL Certificate again and after successful installation un-comment it**

* Open tailor.conf

cd /etc/apache2/sites-available/

sudo a2ensite tailor.conf

#WSGIDaemonProcess any\_name python-home=/var/www/project\_folder\_name/myprojectenv python-path=/var/www/project\_folder\_name

#WSGIProcessGroup any\_name

#WSGIScriptAlias / /var/www/project\_folder\_name/inner\_project\_folder\_name/wsgi.py

#### **Then try to install SSL Certificate again and after successful installation below code un-comment it**

sudo certbot install --cert-name tailorapi.broaderai.com

WSGIDaemonProcess any\_name python-home=/var/www/project\_folder\_name/myprojectenv python-path=/var/www/project\_folder\_name

WSGIProcessGroup any\_name

WSGIScriptAlias / /var/www/project\_folder\_name/inner\_project\_folder\_name/wsgi.py

* Restart Apache2

sudo service apache2 restart

* Reload Apache2

sudo service apache2 restart

* Open Browser and write this url on it and hit enter:

tailorapi.broaderai.com