**Debian Server Set Up for Django Instruction**

In this guide we will set up clean Debian server for Python and Django projects. We will configure secure SSH connection, install from Debian repositories and from sources all needed packages.

**Create user, setup SSH**

Connect through SSH to remote Debian server and update repositories and install some initial needed packages:

sudo apt-get update ; \

sudo apt-get install -y vim htop git curl wget unzip zip gcc build-essential make

Configure SSH:

sudo vim /etc/ssh/sshd\_config

AllowUsers www

PermitRootLogin no

PasswordAuthentication no

Restart SSH server, change www user password:

sudo service ssh restart

**Change locale**

sudo localedef ru\_RU.UTF-8 -i ru\_RU -fUTF-8 ; \

export LANGUAGE=ru\_RU.UTF-8 ; \

export LANG=ru\_RU.UTF-8 ; \

export LC\_ALL=ru\_RU.UTF-8 ; \

sudo locale-gen ru\_RU.UTF-8 ; \

sudo dpkg-reconfigure locales

**Init — must-have packages & ZSH**

sudo apt-get install -y zsh tree redis-server nginx zlib1g-dev libbz2-dev libreadline-dev llvm libncurses5-dev libncursesw5-dev xz-utils tk-dev liblzma-dev python3-dev python-pil python3-lxml libxslt-dev python-libxml2 python-libxslt1 libffi-dev libssl-dev python-dev gnumeric libsqlite3-dev libpq-dev libxml2-dev libxslt1-dev libjpeg-dev libfreetype6-dev libcurl4-openssl-dev supervisor

**Install python 3.8**

Build from source python 3.8, install with prefix to ~/.python folder:

wget https://www.python.org/ftp/python/3.8.4/Python-3.8.4.tgz ; \

tar xvf Python-3.8.\* ; \

cd Python-3.8.4 ; \

mkdir ~/.python ; \

./configure --enable-optimizations --prefix=/home/www/.python ; \

make -j8 ; \

sudo make altinstall

Now python3.8 in /home/www/.python/bin/python3.8.

Add Python 3.8 to BASH

vim ~/.bashrc

export PATH=$PATH:/home/www/.python/bin

Ok, now we can pull our project from Git repository (or create own), create and activate Python virtual environment:

mkdir ~/code

cd code

mkdir Happy-Family

cd Happy-Family

git clone https://github.com/blackw00d/HappyFamily.git

cd ..

python3.8 -m venv env

. ./env/bin/activate

Update pip:

pip install -U pip

pip install –r requirements.txt

**Configure Project**

create .env:

vim .env

Add export parameters to .env file and save.

export SECRET\_KEY="dfsfsdfksjdkfls324242"  
export DEBUG="0"  
export DJANGO\_ALLOWED\_HOSTS="PUBLIC\_ADDRESS 127.0.0.1"  
export SQL\_ENGINE="django.db.backends.postgresql"  
export SQL\_DATABASE="hf\_db"  
export SQL\_USER="admin"  
export SQL\_PASSWORD="some\_password"  
export SQL\_HOST="localhost"  
export SQL\_PORT="5432"  
export AUTH\_USER\_MODEL="HFhtml.Users"  
export EMAIL\_HOST="smtp.yandex.ru"  
export EMAIL\_HOST\_USER="user@yandex.ru"  
export EMAIL\_HOST\_PASSWORD="pass"  
export EMAIL\_PORT="587"  
export EMAIL\_USE\_TLS="1"

Add .env to PATH:

. .env

**Install(if not in requirements) and configure Gunicorn**

vim gunicorn\_config.py

command = '/home/www/code/Happy-Family/env/bin/gunicorn'

pythonpath = '/home/www/code/Happy-Family/HappyFamily'

bind = '127.0.0.1:8001'

workers = 5

user = 'www'

limit\_request\_fields = 32000

limit\_request\_field\_size = 0

raw\_env = 'DJANGO\_SETTINGS\_MODULE=HappyFamily.settings'

mkdir bin

vim bin/start\_gunicorn.sh

#!bin/bash

. /home/www/code/Happy-Family/HappyFamily/.env

. /home/www/code/Happy-Family/env/bin/activate

exec gunicorn -c "/home/www/code/Happy-Family/HappyFamily/gunicorn\_config.py" HappyFamily.wsgi:application

chmod +x bin/start\_gunicorn.sh

. ./bin/start\_gunicorn.sh

**Install and configure PostgreSQL**

Install PostgreSQL 11 and configure locales.

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

RELEASE=$(lsb\_release -cs) ; \

echo "deb http://apt.postgresql.org/pub/repos/apt/ ${RELEASE}"-pgdg main | sudo tee /etc/apt/sources.list.d/pgdg.list ; \

sudo apt update ; \

sudo apt -y install postgresql-11 ; \

Add locales to /etc/profile:

sudo vim /etc/profile

export LANGUAGE=ru\_RU.UTF-8

export LANG=ru\_RU.UTF-8

export LC\_ALL=ru\_RU.UTF-8

Change postges password, create clear database named hf\_db:

sudo passwd postgres

su - postgres

export PATH=$PATH:/usr/lib/postgresql/11/bin

createdb --encoding UNICODE hf\_db --username postgres

exit

Create dbms db user and grand privileges to him:

sudo -u postgres psql

postgres=# ...

create user admin with password 'some\_password';

ALTER USER admin CREATEDB;

grant all privileges on database hf\_db to admin;

\c hf\_db

GRANT ALL ON ALL TABLES IN SCHEMA public to admin;

GRANT ALL ON ALL SEQUENCES IN SCHEMA public to admin;

GRANT ALL ON ALL FUNCTIONS IN SCHEMA public to admin;

CREATE EXTENSION pg\_trgm;

ALTER EXTENSION pg\_trgm SET SCHEMA public;

UPDATE pg\_opclass SET opcdefault = true WHERE opcname='gin\_trgm\_ops';

\q

exit

Now we can test connection. Create ~/.pgpass with login and password to db for fast connect:

vim ~/.pgpass

localhost:5432:hf\_db:admin:some\_password

chmod 600 ~/.pgpass

psql -h localhost -U admin hf\_db

Run SQL dump, if you have:

psql -h localhost hf\_db admin < dump.sql

**Configure NGINX**

sudo vim /etc/nginx/sites-enabled

default@

Delete all and write:

server {

listen 80 default\_server;

listen [::]:80 default\_server;

root /var/www/html;

index index.html index.htm index.nginx-debian.html;

server\_name \_;

location /static {

alias /home/www/code/Happy-Family/HappyFamily/static;

try\_files $uri $uri =404;

}

location /media {

alias /home/www/code/Happy-Family/HappyFamily/media;

try\_files $uri $uri =404;

}

location / {

proxy\_pass http://127.0.0.1:8001;

proxy\_set\_header X-Forwarded-Host $server\_name;

proxy\_set\_header X-Real-IP $remote\_addr;

add\_header P3P 'CP="ALL DSP COR PSAa PSDa OUR NOR ONL UNI COM NAV"';

add\_header Access-Control-Allow-Origin \*;

}

}

Restart NGINX

sudo service nginx restart

**Options for static**

Don’t forget create folder **static** in root directory and then

python manage.py collectstatic

**Install and configure supervisor**

Now recommended way is using Systemd instead of supervisor.

If you need supervisor — welcome:

sudo vim /etc/supervisor/conf.d/HappyFamily.conf

[program:www\_gunicorn]

command=/home/www/code/Happy-Family/bin/start\_gunicorn.sh

user=www

process\_name=%(program\_name)s

numprocs=1

autostart=true

autorestart=true

redirect\_stderr=true

**Configure Gmail**

<https://myaccount.google.com/security>

Untrusted apps that have access to your account

**ON**