Django

The OpenLiteSpeed Django One-Click app automatically installs performance web server openlitespeed, and Django. This image tends to be up to **2.5** times faster than Django on Nginx with Gunicorn! Developers can update files directly without using the built-in development server. OpenLiteSpeed features easy setup for SSL and RewriteRules. OLS is flexible and also supports Node.js and Ruby apps, as well as CMSs like WordPress.

Quick Start

Step 1.

Use the "OpenLiteSpeed NodeJS Marketplace Image" to create a ECS with any plan you want.

Step 2.

An interactive script that runs will first prompt you for your domain or subdomain.

You can press CTRL+C and continue to SSH. The prompt will open again the next time you log in, and will continue to do so until you finish the whole setup.

```
Please input a valid domain:
Please verify it is correct. [y/N]
```

Tip

Enter the root domain only, then the system will add both the root domain and the www domain for you..

You can also automatically apply Let's Encrypt SSL if your domain is pointed to this server already. Enter `y` and your email address to finish the process.

```
Do you wish to issue a Let's encrypt certificate for this domain? [y/N]
Please enter your E-mail:
Please verify it is correct: [y/N]
```

Once finished, you should see **Certificate has been successfully installed...**

```
Do you wish to force HTTPS rewrite rule for this domain? [y/N]
```

Once finished, force HTTPS rules will be applied

```
Do you wish to update the system which include the web server? [Y/n]
```

This script will automatically go away after your domain has been added.

Step 3.

Visit the default script by entering `http://Server_IP/` on your browser and you should see `Hello, world!`.

You can also enter `http://Server_IP/admin/` to get the Django administration page

Start editing the Django project under `/usr/local/lsws/Example/html/`

Please access into virtual environment first

```
source /usr/local/lsws/Example/html/bin/activate
```

Django demo site structure:

```
demo
|-demo
   |-settings.py
   |-urls.py
   |-wsgi.py
   |-__init__.py
|-manage.py
|-public
   |-static
|-db.sqlite3
|-app
|-stderr.log
```

Optional but recommended:

Update system software.

```
sudo apt-get update && sudo apt-get upgrade -y
```

Your system is installed and ready to use!

Components

The OpenLiteSpeed Django 1-Click Droplet installs several packages and performs other actions on your system.

Package Installation

Component	Version
Linux	Ubuntu 18.04.1
OpenLiteSpeed	Latest from LiteSpeedtech Repo
Django	Latest from APT
Certbot	Latest from Certbot’s PPA

Other Actions

Enable Firewall

This droplet enables the UFW firewall to allow only SSH (port 22), HTTP (port 80) and HTTPS (port 443) access.

Set Up Demo Django Project

The droplet executes the following steps in order to set up a demo Django project called `demo`, and a Django app called `app`. These steps are taken automatically. You don't need to do anything.

- Create a demo project called `demo` and Django app called `app`:

  ```
  django-admin startproject demo
  cd demo
  python3 manage.py startapp app
  ```

- Edit `settings.py` to allow hosts:

```
ALLOWED_HOSTS = ['*']
```

- Edit `settings.py` to look for static files in a directory called `static` in our root directory:

```
STATIC_URL = '/python/static/'
STATIC_ROOT = '/usr/local/lsws/Example/html/demo/public/static'
```

- Collect static files:

```
mkdir -p public/static
python3 manage.py collectstatic
```

- Create a view which returns `Hello, world!` as an HTTP response:

```
def index(request):
    return HttpResponse("Hello, world!")
```

- Set a default url:

```
urlpatterns = [
    path('', views.index, name='index'),
    path('admin/', admin.site.urls),
]
```

- Change owner:

```
chown -R nobody:nogroup /usr/local/lsws/Example/html/demo
```

- Set up a context via **Web Admin > Virtual Hosts > Context > Add**:
    - Type = `App Server`
    - URI = `/`
    - Location = `/usr/local/lsws/Example/html/demo/`
    - Binary Path = `/usr/local/lsws/fcgi-bin/lswsgi`
    - Application Type = `WSGI`
    - Startup File = `demo/wsgi.py`
    - Environment = `PYTHONHOME=/usr/local/lsws/Example/html`

Benchmark Comparison

Use the following command to test from an x-CPU plan server(ab) to an x-CPU plan server(DOMAIN)

```
ab -n 100000 -k -H "Accept-Encoding: gzip,deflate" -c 100 http://DOMAIN/
```

Requests per Second (The larger the number, the better)

Nginx+Gunicorn	Openlitespeed
700	1750

How to Access the Installed Software

From a terminal on your local computer, connect to the server as root, like so:

```
ssh root@use_your_server_ip
```

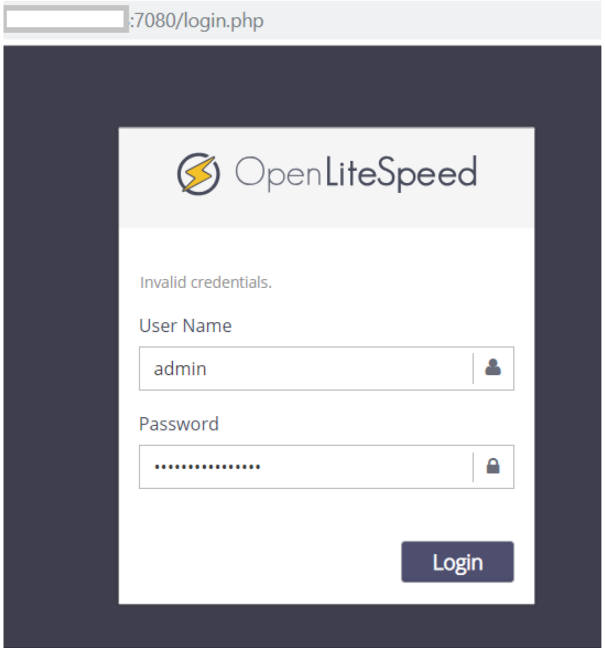
Be sure to substitute the server's public IP address for `use_your_server_ip`.

Web Server Control Panel Access

Get the WebAdmin admin password:

```
cat .litespeed_password
```

Visit `https://use_your_server_ip:7080` to access WebAdmin in a browser.



[redacted]:7080/login.php

Invalid credentials.

User Name: admin

Password: ••••••••••••••••

By default, WebAdmin uses port 7080. To allow access to 7080 from your IP(e.g. 1.2.3.4):

```
ufw allow from 1.2.3.4 to any port 7080
```

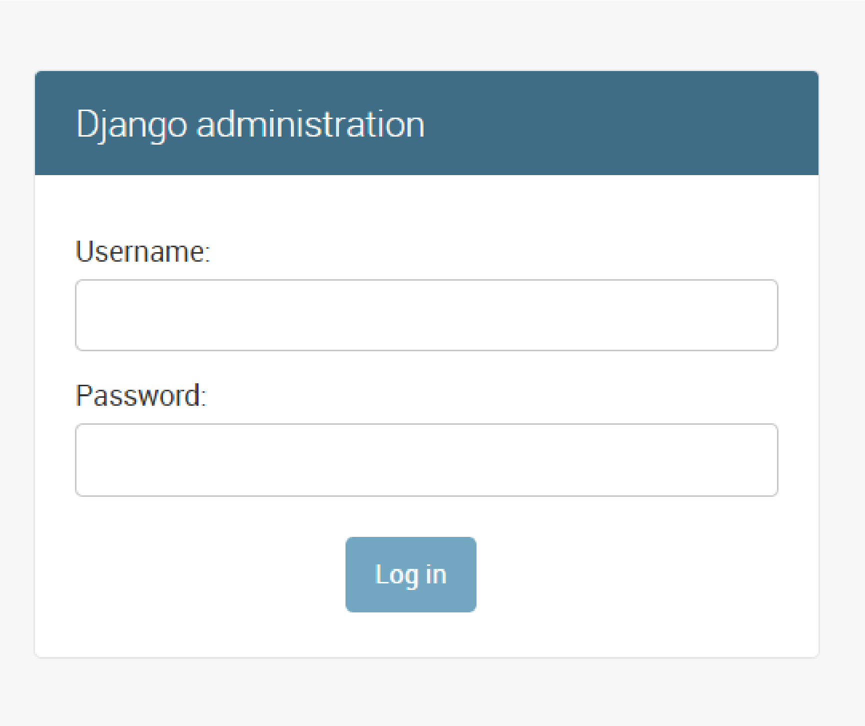
You can also allow all IPs access to port 7080:

```
ufw allow 7080
```

We suggest turning this port off once you've finished setup:

```
ufw delete allow 7080
```

Django Administration Page



Run the following command in the project directory to setup an account and password:

```
python3 manage.py createsuperuser
```

Optional Setup

Enable HTTPS

Setting up an SSL certificate enables HTTPS on the web server, which secures the traffic between the server and the clients connecting to it. Certbot is a free and automated way to set up SSL certificates on a server.

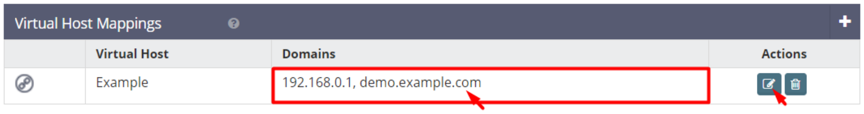
Step 1. Register Domain

To use Certbot, you'll need a registered domain name and DNS records:

- An A record from the domain (e.g., `example.com`) to the server's IP address
- An A record from the domain prefaced with `www` (e.g., `www.example.com`) to the server's IP address.

Step 2. Add Domain to Listener

Navigate to **OpenLiteSpeed Web Server WebAdmin > Listeners**, and add Your Domain to HTTP/HTTPS.



	Virtual Host	Domains
	Example	192.168.0.1, demo.example.com

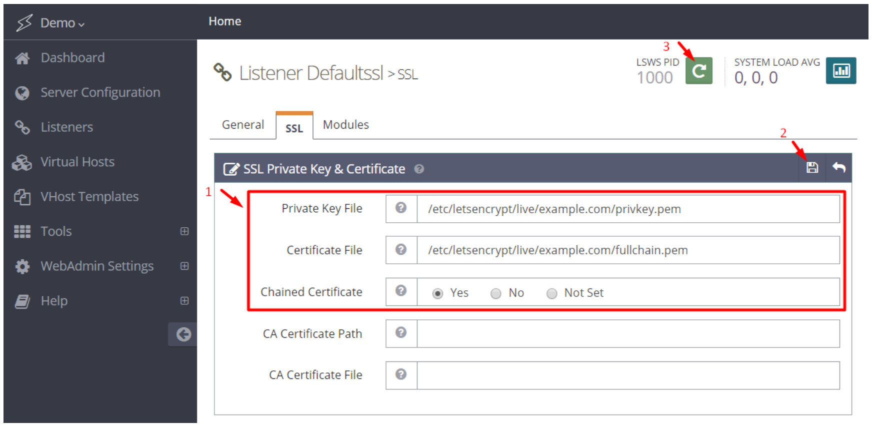
Step 3. Certbot

Once the DNS records are set up, you can generate the SSL certificate. Be sure to substitute the correct domain name in the following command:

```
certbot certonly --webroot -w /var/www/html/ -d example.com -d www.examp
```

If certificate verification is a success, you should find your certificate files stored in `/etc/letsencrypt/`

Step 4. Set SSL for HTTPS



Private Key File: /etc/letsencrypt/live/example.com/privkey.pem
Certificate File: /etc/letsencrypt/live/example.com/fullchain.pem
Chained Certificate: ● Yes ○ No ○ Not Set
CA Certificate Path:
CA Certificate File:

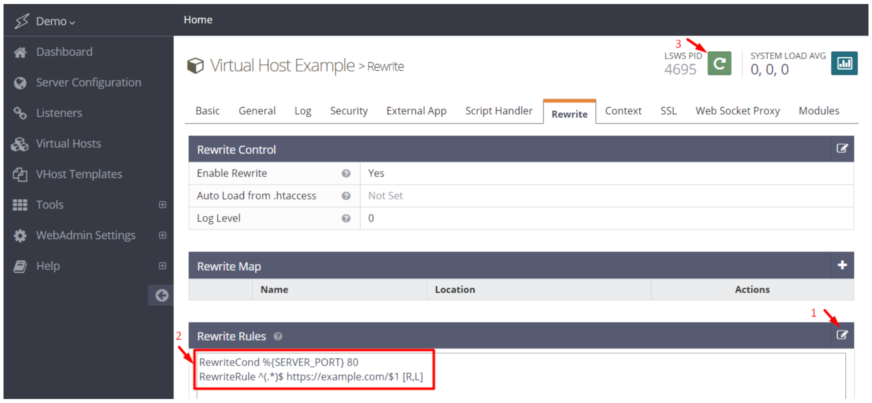
Navigate to **OpenLiteSpeed Web Server WebAdmin > Listeners > SSL**, and edit the following three items:

- **Private Key File** = `/etc/letsencrypt/live/example.com/privkey.pem`
- **Certificate File** = `/etc/letsencrypt/live/example.com/fullchain.pem`
- **Chained Certificate** = `Yes`

Save and perform a Graceful Restart.

Now your server should support TLS1.1, TLS 1.2, and TLS 1.3.

Step 5. Redirect HTTP to HTTPS



LSWS PID 4695

SYSTEM LOAD AVG 0, 0, 0

Rewrite Control

Enable Rewrite	Yes
Auto Load from .htaccess	Not Set
Log Level	0

Rewrite Map

Name	Location	Actions

Rewrite Rules

```
RewriteCond %{SERVER_PORT} 80
RewriteRule ^(.*)$ https://example.com/$1 [R,L]
```

HTTPS traffic on port 443 is already allowed through the firewall. After you set up HTTPS, you can optionally rewrite all HTTP traffic to HTTPS.

Add the following rules to **OpenLiteSpeed Web Server WebAdmin > Virtual Hosts > Rewrite > Rewrite Rules**

```
RewriteCond %{SERVER_PORT} 80
RewriteRule ^(.*)$ https://example.com/$1 [R,L]
```

Method for Uploading Files

You can serve files from the web server by adding them to the web root using SFTP or other tools.

Frequently Asked Questions

How do I Reset my Web Server WebAdmin Password?

If you forget your password, you may run the following command to reset it:

```
/usr/local/lsws/admin/misc/admpass.sh
```

It will ask for the WebAdmin username, which should be `admin`. Then, enter your new password.

How do I Create Additional Virtual Hosts?

▼ **Manual Setup**

By default, OpenLiteSpeed has an example virtual host already created. You can create more virtual hosts if you like. See Create Virtual Hosts on OpenLiteSpeed.

How do I Create Additional Apps by Context?

How do I Change the Django Startup File?

If you want to change the default startup file name from `wsgi.py` to `demo.js`, just update the Context and set **Startup File** = `demo.py`.

How do I Create Django with a Virtual Environment?

How do I Fix Permission Issues?

```
chown -R nobody:nogroup /usr/local/lsws/Example/html/demo
```

Change `/usr/local/lsws/Example/demo` to your actual project path

How do I Switch SQLite to PostgreSQL?

If you would prefer to use PostgreSQL instead of the default SQLite 3 database, try the following steps:

1. Install PostgreSQL:

   ```
   apt install postgresql postgresql-contrib -y
   ```

2. Create the database, user, and password, and apply privileges:

   ```
   sudo -u postgres psql
   CREATE DATABASE demo;
   CREATE USER myprojectuser WITH PASSWORD 'password';
   GRANT ALL PRIVILEGES ON DATABASE demo TO myprojectuser;
   \q
   ```

3. Access the virtual environment by following step 3:

   ```
   source /usr/local/lsws/Example/html/bin/activate
   ```

4. Install PostgreSQL drivers for Python 3:

   ```
   pip3 install psycopg2-binary
   ```

5. Update the Django project database. Go to the project, and edit the `settings.py` file:

```
cd /usr/local/lsws/Example/html/demo
vi demo/settings.py
```

Update database section, replacing the following:

```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.sqlite3',
        'NAME': os.path.join(BASE_DIR, 'db.sqlite3'),
    }
}
```

with:

```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.postgresql_psycopg2',
        'NAME': 'demo',
        'USER': 'myprojectuser',
        'PASSWORD': 'password',
        'HOST': 'localhost',
        'PORT': '',
    }
}
```

6. Update/migrate the database:

```
python3 manage.py makemigrations
python3 manage.py migrate
```

Verify the database connection with the following command: `python3 manage.py shell`

```
>>> from django import db
>>> print(db.connections.databases)
```

```
Output:
{'default': {'ENGINE': 'django.db.backends.postgresql_psycopg2', 'NAME':
```

If you have existing data that must be migrated, take a look at this StackOverflow topic for tips.

Note: Feel free to use the database name, username and password of your choice.

How do I Set Up a Wagtail Project with an Existing Config?

Move default "Hello World" demo site from server:

```
mv /usr/local/lsws/Example/html/demo /tmp
```

Install Wagtail python package:

```
pip3 install wagtail
```

Start a new demo site:

```
wagtail start demo
```

Access the project directory:

```
cd demo
```

Install the required package:

```
pip3 install -r requirements.txt
python3 manage.py migrate
```

Create a static file folder:

```
mkdir -p /usr/local/lsws/Example/html/demo/demo/public/static
```

Add these two lines to the end of `/usr/local/lsws/Example/html/demo/demo/settings/base.py`:

```
STATIC_URL = '/python/static/'
STATIC_ROOT = '/usr/local/lsws/Example/html/demo/public/static'
```

Collect static files:

```
python3 manage.py collectstatic
```

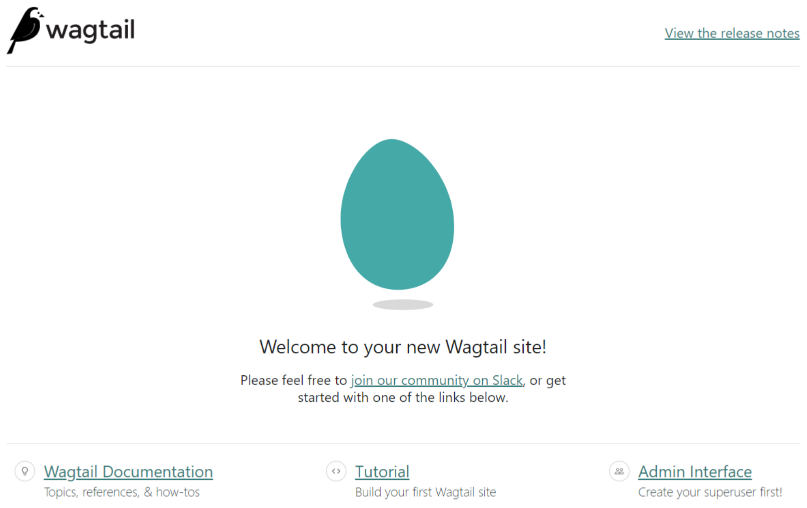
Create superuser for admin page access:

```
python3 manage.py createsuperuser
```

Change owner to default LiteSpeed permissions:

```
chown -R nobody:nogroup /usr/local/lsws/Example/html/demo/
```

Front page:



wagtail

View the release notes

Welcome to your new Wagtail site!

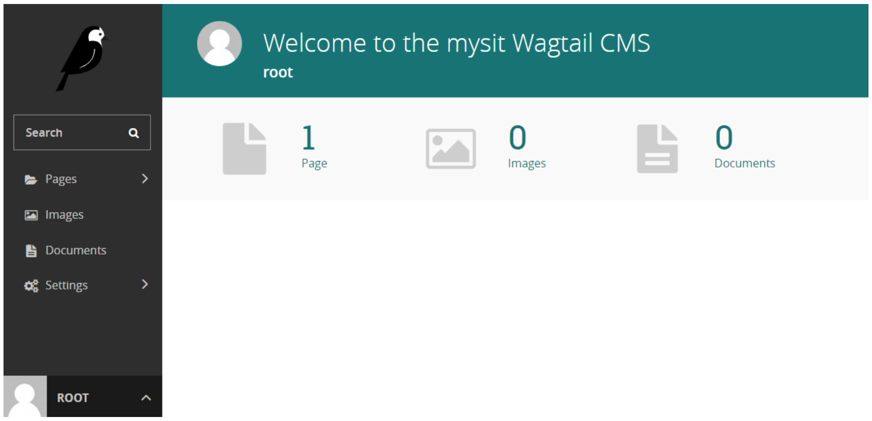
Please feel free to join our community on Slack, or get started with one of the links below.

Wagtail Documentation
Topics, references, & how-tos

Tutorial
Build your first Wagtail site

Admin Interface
Create your superuser first!

Admin page:



Welcome to the mysit Wagtail CMS

root

How do I Import my Django Project From Git?

Let's use pyconbalkan as an example. To import the project, run the following command:

```
git clone https://github.com/PythonBalkan/pyconbalkan
```

Install your project's requirements:

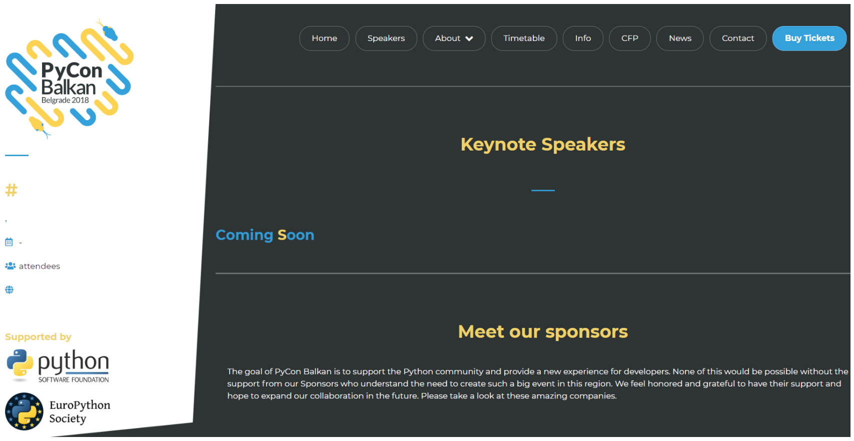
```
pip3 install -r requirements.txt
```

Create a `.env` file, and the content required by the project. Run migrate:

```
python3 manage.py migrate
```

Follow the **Set Up Demo Django Project** instructions to set up `settings.py`, collect static files, and change owner and context. More info

You should see the following screen:



Keynote Speakers

Coming Soon

Meet our sponsors

The goal of PyCon Balkan is to support the Python community and provide a new experience for developers. None of this would be possible without the support from our Sponsors who understand the need to create such a big event in this region. We feel honored and grateful to have their support and hope to expand our collaboration in the future. Please take a look at these amazing companies.