

uWSGI

uWSGI is a deployment option on servers like [nginx](#), [lighttpd](#), and [cherokee](#); see [FastCGI](#) and [Standalone WSGI Containers](#) for other options. To use your WSGI application with uWSGI protocol you will need a uWSGI server first. uWSGI is both a protocol and an application server; the application server can serve uWSGI, FastCGI, and HTTP protocols.

The most popular uWSGI server is [uwsgi](#), which we will use for this guide. Make sure to have it installed to follow along.

Watch Out:

Please make sure in advance that any `app.run()` calls you might have in your application file are inside an `if __name__ == '__main__':` block or moved to a separate file. Just make sure it's not called because this will always start a local WSGI server which we do not want if we deploy that application to uWSGI.

Starting your app with uwsgi


uwsgi is designed to operate on WSGI callables found in python modules.

Given a flask application in `myapp.py`, use the following command:

```
$ uwsgi -s /tmp/yourapplication.sock --manage-script-name --mount /your
```

The `--manage-script-name` will move the handling of `SCRIPT_NAME` to uwsgi, since it is smarter about that. It is used together with the `--mount` directive which will make requests to `/yourapplication` be directed to `myapp:app`. If your application is accessible at root level, you can use a single `/` instead of `/yourapplication`. `myapp` refers to the name of the file of your flask application (without extension) or the module which provides `app`. `app` is the callable inside of your application (usually the line reads `app = Flask(__name__)`).

If you want to deploy your flask application inside of a virtual environment, you need to also add `--virtualenv /path/to/virtual/environment`. You might also need to add `--plugin python` or `--plugin python3` depending on which python version you use for your project.

 v: 1.1.x ▼

Configuring nginx

A basic flask nginx configuration looks like this:

```
location = /yourapplication { rewrite ^ /yourapplication/; }
location /yourapplication { try_files $uri @yourapplication; }
location @yourapplication {
    include uwsgi_params;
    uwsgi_pass unix:/tmp/yourapplication.sock;
}
```

This configuration binds the application to `/yourapplication`. If you want to have it in the URL root its a bit simpler:

```
location / { try_files $uri @yourapplication; }
location @yourapplication {
    include uwsgi_params;
    uwsgi_pass unix:/tmp/yourapplication.sock;
}
```