How to install nginx, gunicorn and Flask

Published: February 24, 2011

I like to work on my personal project and for the moment, I wanted to use nginx, gunicorn with my Flask application. To explain the architecture of my installation, nginx is the web server with a cache for the static content (html, css, javascript and images). the Flask application will be served by the gunicorn webserver; When there is a request, the nginx server (front-end) will forward to the gunicorn server for the dynamic content.

Flask

Here is a small project I like to follow, this one has been developped by Armin Ronacher The documentation is certainly the best documentation for a project, pragmatic and concise. The source code is well documented and the developer is the maintainer of Werkzeug, Pygments and is in the Pocoo Team .

Installation

```
pip install flask
```

Example

```
#!/usr/bin/env python
# -*- encoding: utf-8 -*-
"""
    deploy.py
    ~~~~~~~~~

    :author: (c) 2011 - Stephane Wirtel <stephane@wirtel.be>
    :license: BSD
"""
from flask import Flask
from flask import render_template_string

# load the middleware from werkzeug
# This middleware can be applied to add HTTP proxy support to an application
# that was not designed with HTTP proxies in mind.
# It sets `REMOTE_ADDR`, `HTTP_POST` from `X-Forwarded` headers.
from werkzeug.contrib.fixers import ProxyFix


app = Flask(__name__)
@app.route('/')
def index():
    return render_template_string('<h1>Hello World</h1>')

app.wsgi_app = ProxyFix(app.wsgi_app)
```

Gunicorn

I discovered gunicorn during the PyCon FR 2010 at Paris, I found this project very cool, because the community is very active, and Benoit C is a good evangelist of gunicorn. He wants to create the best WSGI server, the fastest and the simplier to configure.

gunicorn has written in C, it's a very minimalist web server, but its works is well done.

Installation

```
pip install gunicorn
```

Configuration

The configuration is easy, you just define a configuration file. Here is a small example where we specify the number of workers.

```
workers = 2
bind = '127.0.0.1:18000'
proc_name = 'www.example.com'
pidfile = '/tmp/www.example.com.pid'
```

```
gunicorn deploy:app
```

Nginx

Installation

```
apt-get install nginx
```

Configuration

The configuration of nginx (on Debian Like) is in the /etc/nginx/sites-enabled directory

```
upstream frontends {
    # We define the binding of the gunicorn web server
    server 127.0.0.1:18000;
}

server {
    listen 80;

    server_name localhost;

    access_log /var/log/nginx/localhost-access.log;
    error_log /var/log/nginx/localhost-error.log debug;

    location / {
        proxy_pass_header Server;
        proxy_set_header Host $http_host;
        proxy_set_header X-Forward-For $proxy_add_x_forwarded_for;
        proxy_redirect off;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Scheme $scheme;
        if (-f $request_filename) {
            proxy_pass http://frontends;
            break;
        }
    }
}
```

- Benchmark of Python WSGI Servers
- Nginx – HTTP server
- Flask – Web site of the Flask microframework
- Gunicorn – WSGI Server
- How to deploy Flask
- Gunicorn Deployment

Like

Add New Comment

Login

Type your comment here.

Showing 1 comment

Sort by popular now

Svejk

Hi. You have an error in the code you posted, in the config file for nginx. Where you have

if (-f $request_filename)   it should be
if (!-f $request_filename).I was using your tutorial and it took me a bit to figure that out. Hope to spare someone else the effort.
Thank you.

6 months ago   Like   Reply

Subscribe by email   RSS

Trackback URL http://disqus.com/forums/stephane-