Why do I need Nginx and something like Gunicorn?

Im looking for a overly simplified answer for the following question. Im trying to build a foundational understanding of how Nginx works along side something like GUnicorn.

Do I need both Nginx and something like Gunicorn to deploy django apps on Nginx?

If so, what actually handles the HTTP requests?

Ps. I dont want to use Apache and mod_wsgi!

nginx django

edited **Nov 15 '11 at 23:48**
Bart De Vos
10.3k 3 19 43

asked **Nov 15 '11 at 21:16**
a.m.
63 5

feedback

1 Answer

Overly simplified: You need something that executes Python but Python isn't the best at handling all types of requests.

[disclaimer: I'm a Gunicorn developer]

Less simplified: Regardless of what app server you use (Gunicorn, mod_wsgi, mod_uwsgi, cherrypy) any sort of non-trivial deployment will have something upstream that will handle the requests that your Django app should not be handling. Trivial examples of such requests are serving static assets (images/css/js).

This results in two first tiers of the classic "three tier architecture". Ie, the webserver (Nginx in your case) will handle many requests for images and static resources. Requests that need to be dynamically generated will then be passed on to the application server (Gunicorn in your example). (As an aside, the third of the three tiers is the database)

Historically speaking, each of these tiers would be hosted on separate machines (and there would most likely be multiple machines in the first two tiers, ie: 5 web servers dispatch requests to two app servers which in turn query a single database).

In the modern era we now have applications of all shapes and sizes. Not every weekend project or small business site actually needs the horsepower of multiple machines and will run quite happily on a single box. This has spawned new entries into the array of hosting solutions. Some solutions will marry the app server to the web server (Apache httpd + mod_wsgi, Nginx + mod_uwsgi, etc). And its not at all uncommon to host the database on the same machine as one of these web/app server combinations.

Now in the case of Gunicorn, we made a specific decision (copying from Ruby's Unicorn) to keep things separate from Nginx while relying on Nginx's proxying behavior. Specifically, if we can assume that Gunicorn will never read connections directly from the internet, then we don't have to worry about clients that are slow. This means that the processing model for Gunicorn is embarrassingly simple.

The separation also allows Gunicorn to be written in pure Python which minimizes the cost of development while not significantly impacting performance. It also allows users the ability to use other proxies (assuming they buffer correctly).

As to your second question about what actually handles the HTTP request, the simple answer is Gunicorn. The complete answer is both Nginx and Gunicorn handle the request. Basically, Nginx will receive the request and if it's a dynamic request (generally based on URL patterns) then it will give that request to Gunicorn, which will process it, and then return a response to Nginx which then forwards the response back to the original client.

So in closing, yes. You need both Nginx and Gunicorn (or something similar) for a proper Django deployment. If you're specifically looking to host Django with Nginx, then I would investigate Gunicorn, mod_uwsgi, and maybe CherryPy as candidates for the Django side of things.

answered **Nov 15 '11 at 21:49**

Paul J. Davis
406 4 2

Thanks for taking the time to write such a detailed answer! Any recommended reading on the this "3 tier architecture"? – a.m. Nov 16 '11 at 1:23

Was this post useful to you? Yes No

Not the answer you're looking for? Browse other questions tagged nginx django or ask your own question.