



← [So how exactly do I delete a design?](#)

Unicorn + Bluepill + Nginx

May 16, 2011 [View Comments](#) by James

If you found this article and are expecting a post about the Matrix Blue/Red pill analogy combined with a fairytale animal then I'm afraid you're on the wrong page. If however, you're interested in a fast, kick-ass Ruby on rails environment then you *ARE* in the right place.

Getting a stable, Ruby on Rails app running in a production environment can often be tricky. Sometimes there's different Ruby versions installed, different users might have different gem paths, environment variables need to be set, RVM needs to be considered, unicorn, nginx, apache-*sometimes it's a complete and utter mess.*

That's why my advice is simple. Keep it simple. With [Halftone](#), the only Rails application on the server is Halftone. We don't share resources with any other application. Since I started from scratch on a clean server, our rails app has the bare minimum needed to run. Only one Ruby version, one repository of gems, one HTTP server and one process monitoring software.

I cannot stress this enough, keep your app's system requirements simple and isolated.

Below I'm going to run through the software I use to keep things dandy at Halftoneapp.

Bluepill

Bluepill is a process monitoring engine written in Ruby.

We love Ruby here at Halftone so it was hard to pass up working with configuration files which accept Ruby code. Bluepill monitors our Unicorn process, if it runs amuck and starts chewing up valuable memory or CPU usage – we send a QUIT signal which starts a graceful shutdown, allowing Unicorn to finish. Unicorn will then spawn a new worker process to make up for the dead unicorn, as sad as that sounds.

We did start out using GOD, however recently switched to Bluepill due to its ability to monitor child processes such as those spawned by Unicorn.

How did we install it?

```
gem install bluepill
sudo mkdir /var/bluepill (Bluepill requires this folder as per it's docs)
sudo chown appuser.appuser /var/bluepill
```

How do we run it?

First you'll need a configuration file. We've uploaded a version of the Bluepill config we use, though modified slightly with extended comments. You can find it [here](#).

DON'T FREAK OUT. Yes, the configuration file linked above is lengthy but don't be scared – take some time to read it all and become familiar with the settings.

We run Bluepill as a local user, the same user that runs our Rails application.

To start Bluepill run

```
1 | bluepill load /path/to/unicorn.pill --no-privileged
```

This starts Bluepill and should start Unicorn. Run `ps -aux` to see if all looks swell. Keep an eye on the Bluepill log to debug any problems you may encounter.

Visit [Bluepill](#) for more details.

Note! Because we've started Bluepill with the no-privileged mode, you'll need to specify this everytime you interact with it.

For example;

```
1 | bluepill stop unicorn --no-privileged
```

UNICORN

Straight from the homepage; “Unicorn is an HTTP server for Rack applications designed to only serve fast clients on low-latency, high-bandwidth connections and take advantage of features in Unix/Unix-like kernels”.

Or in simple terms, Unicorn gets in, serves the request and then gets the heck outta there. Unicorn does not implement persistent connections w can often leave client connections idle; holding up valuable worker processes. All our intensive, lengthy processes are handled by Resque; a background job processor (we'll talk about that in a separate post).

“Unicorns don't care if you believe in them any more than you care if they believe in you”

How did we install it?

```
gem install unicorn
```

Visit the [Unicorn website](#) for more details.

What about our config file?

There's not much to Unicorn. Simply point it to your Rails app, set the number of worker processes you'd like and away you go. See here for a [sample unicorn config](#).

How do you run it?

If you're using Bluepill to manage Unicorn, you'll see how we start Unicorn in the config file. Otherwise you can launch Unicorn manually using the following.

```
1 | /path/to/unicorn_rails -Dc unicorn.conf.rb -E production
```

Nginx

We use Nginx to reverse proxy HTTP requests to Unicorn.

A request comes in for halftoneapp.com/blah and Nginx sends this to Unicorn via a unix socket.

How did we install it?

On Debian – apt-get install nginx

We added the following to the Nginx configuration file.

```
1 | server {
2 |     server_name yourdomain.com;
3 |     root /home/yourappuser/app/current/public;
4 |     try_files /system/maintenance.html $uri $uri/index.html @app;
5 |     location @app {
6 |         proxy_pass http://unix:/home/yourappuser/app/shared/sockets/yourappuser.sock;
7 |     }
8 |     error_page 404 400 500 502 /error.html;
9 |     location = /error.html {
10 |         root /var/www;
11 |     }
12 | }
```

The important part is lining up the proxy_pass option to the same socket you've specified in the Listen option in the Unicorn configuration file.

Parting lines

I hope this helps you in your quest for the ultimate Ruby on Rails environment. If you have any questions or if something I've said is incorrect (we are human after all), feel free to leave a comment below.

Categories: [Tutorials](#)

Like



and 1 other liked this.

Add New Comment



Showing 13 comments

Sort by oldest first



Georges Duplessy

Nice write-up, I'll try it out as soon as I can in my development environment - yeah that's my desktop haha.

[8 months ago](#)

[Like](#) [Reply](#)



jamsi

Thanks mate. Let us know how you go :)

[8 months ago](#) [in reply to Georges Duplessy](#)

[Like](#) [Reply](#)



Georges Duplessy

I definitely will :)

[8 months ago](#) [in reply to jamsi](#)

[Like](#) [Reply](#)



Bunnios

The real WTF in this post is "... in a production environment...". What you describe sounds more like a dev environment

[8 months ago](#)

[Like](#) [Reply](#)



jamsi

How would you describe a production environment?

[8 months ago](#) [in reply to Bunnios](#)

[Like](#) [Reply](#)



TerenceTrentDarby

They got the description bang on. You wouldn't do anything like this in dev... you'd just restart unicorn. Running BluePill in dev is silly, unless you are a huge ass distributed corporation. If you are a one man shop, articles like this are a god send, so jamsi, thank you man.

[7 months ago](#) [in reply to Bunnios](#) [3 Likes](#)

[Like](#) [Reply](#)



gabriele renzi

care to share why you preferred this setup to, say, nginx+passenger?

[7 months ago](#)

[Like](#) [Reply](#)



Stanley Shilov

Thanks for the post. How do you ensure that bluebill starts up on an unexpected reboot?

[7 months ago](#)

[Like](#) [Reply](#)



jamsi



James Thullbery

Do you use RVM on your server? You mention it above and you're not using sudo when you install your gems.

[7 months ago](#)

[Like](#) [F](#)



jamsi

Yes we do. All our gems are installed in the app users home directory.

[7 months ago](#) [in reply to James Thullbery](#)

[Like](#) [F](#)



billybobza

Thanks for the article. I come from a php background and am new to hosting Rails apps. Could you explain why you recommend only one app per server? I've been wondering how I can have multiple apps on my stage server, and then host multiple apps on the live server. What sort of problems are encountered?

Thanks in advance

Billy

[6 months ago](#)

[Like](#) [F](#)



Tom

This is great. The only part missing is how you set bluepill to automatically start after a system restart.

[2 weeks ago](#)

[Like](#) [F](#)

[✉ Subscribe by email](#) [📡 RSS](#)

• About

Getting feedback about your designs over email sucks. Things get lost, attachments never work, and comments are unorganized. [Halftone](#) is a controlled, polished and discrete place to show, speak and improve.

• Recent Posts

- [Unicorn + Bluepill + Nginx](#)
- [So how exactly do I delete a design?](#)
- [CSS Guide: Annotation Comments](#)
- [Annotations v.3](#)
- [A day in the life of a Halftoneapp Developer](#)

