Best practices for new Rails deployments on Linux?

Asked 16 years, 1 month ago Modified 15 years, 6 months ago

Viewed 4k times

I've used straight Mongrel, I've used Mongrel clusters behind Apache, I've looked at Thin, and I'm becoming very intrigued by Passenger. I've looked at Nginx, too. I've looked at MRI, Ruby Enterprise Edition, Rubinius, and JRuby. There are a lot of options, each claiming to be the new holy grail.

What is the best option out there for a brand new, fully up-to-date deployment? The only assumptions are this:

- The app is Rails 2.2 based. (I know 2.2 isn't fully released yet, but neither is this deployment.)
- The server is Linux-based. Probably Ubuntu Hardy, but really, whatever works best in this case.
- Rails will need to be fully functional and probably talk to a MySQL database.
- Everything else is negotiable.

Given these especially broad constraints, which combination of software will yield the best result, in terms of concurrency and low overhead?

I'm leaning toward Apache with the "worker" mpm and Passenger + Ruby Enterprise Edition, simply because it offers immediate stability and simplicity of setup and maintenance.

Am I likely to be particularly better off with another option?

ruby-on-rails ruby linux deployment release-management

Share

Improve this question

Follow

edited Jun 1, 2009 at 13:23

Ludwig Weinzierl

16.5k • 10 • 46 • 49

asked Nov 11, 2008 at 6:24

Jim Puls

81k • 10 • 75 • 78

9 Answers

Sorted by: Highest score (default)

16

I switched from Mongrel Cluster to Passenger two weeks ago (Debian Linux Server). I didn't look back for a second. Passenger is probably the easiest way to get your new server up and running. Performance and reliability are reasonable too.

Personally, I like to spend my time working on exciting new Rails projects rather than dealing with deployment

issues - Passenger enables me to do exactly that. However, Mongrel or something else may still be preferable if you have some kind special requirements (doesn't apply for most products).

Share Improve this answer

Follow

answered Nov 11, 2008 at 13:16

Christoph Schiessl

6,858 ● 4 ● 35 ● 45

1 +1 for Passenger -- super easy deployment as just as fast (sometimes faster for my app) as Mongrel used to be. Wonderful! – PJ. Nov 12, 2008 at 13:59

another +1 for passenger. I struggled for a week trying to set up a pack of mongrels on debian, but finished a deployment setup in an afternoon with passenger. – Alan Nov 15, 2008 at 22:22

+1 for Passenger. Deployment took me about 10 minutes, from start to finish. – mlambie Nov 27, 2008 at 5:53

10

This morning, DHH talks about this very topic on his own blog:

> But somehow the message of Passenger has been a little slow to sink in. There's already a ton of big sites running off it. Including Shopify, MTV, Geni, Yammer, and we'll be moving over first Ta-da List shortly, then hopefully the rest of the 37signals suite quickly thereafter.

> So while there are still reasons to run your own custom multi-tier setup of manually configured pieces, just like there are people shying away from mod_php for their particulars, I think we've finally settled on a default answer. Something that doesn't require you to really think about the first deployment of your Rails application. Something that just works out of the box. Even if that box is a shared host!

http://www.loudthinking.com/posts/30-myth-1-rails-is-hard-to-deploy

Tobias Lütke on the topic of switching Shopify (million requests/day) to Passenger:

> All this means that the total amount of memory that is used by Shopify during normal operations went from average of 9GB to an average of 5GB. We evenly distributed the savings amongst more Shopify processes and more memcached space which moved our average response time from 210ms to 130ms while traffic grew 30% in the last few months.
>
> In conclusion: I cannot see any reason to choose a different deployment strategy at this point. Its simple, complete, fast and well documented.

http://blog.leetsoft.com/2008/11/15/passenger

Share Improve this answer

Follow

edited Jun 20, 2020 at 9:12

Community Bot

1 • 1

answered Nov 13, 2008 at 16:08

Jim Puls

81k • 10 • 75 • 78

4

We've been using the old standard nginx -> mongrel stack for the last 18 months, and although it was not trivial to set up the first time around, it's proven flexible, and has dealt with some very high traffic sites for us. Nginx in particular has been absolutely rock solid and fast, and if you can get your app page-caching you can deal with a lot of requests.

Stuck mongrels have been an issue, so we use monit to kill them when they misbehave. Again, it was not totally trivial to set up, but we've used the same process on many many sites at this point.

We haven't played with passenger yet, so perhaps it's easier and more stable, I'll defer to the other responders on that one, all I can say is that there's no reason at all you can't build a solid stack with nginx and mongrel.

Share Improve this answer

Follow

answered Nov 11, 2008 at 14:14

Cameron Price

1,185 • 8 • 14

We have switched fron NginX+Mongrel to Passenger.

2

I fully believe that Passenger is going to be the new standard for rails, despite NginX and Mongrel cluster being endorsed by some very smart people. Recent advances in Passenger have really propelled it forward.

Our current configuration is something like this:

Web servers

- Ubuntu 8.04 LTS
- Phusion Passenger on Apache2
- MRI Ruby 1.8.6 and friends (form apt)
- Ruby Gems 1.3.0 (Installed from source)

Database servers

- Centos 5
- MySQL Cluster (we just switched to this, but it is promising)

Having standardized on the exact linux distro we've been able to write Capitrano recipes to help deployment (slight variations in configuration have been the source of MANY service outages) and otherwise simplify our lives.

Share Improve this answer

Follow

answered Nov 15, 2008 at 22:04

csexton

24.7k ● 15 ● 57 ● 57

1

Have a look at Litespeed. You can get a free version that runs on 1 cpu or pay to get multi cpu. It is a bit expensive but is rock solid and handles rails brilliantly (i.e. uses less memory and is less of an overhead to monitor and setup). I run a massive amount of apps on it and it doesn't miss a beat.

Share Improve this answer

Follow

answered Nov 11, 2008 at 10:31

user36344

11 ● 2

1

We also switched from Mongrel to mod_passenger and found stability greatly improved with this effort required to setup and maintain. Good choice.

Share Improve this answer

Follow

answered Nov 11, 2008 at 13:34

Simon

1

Another bit of gold:

Josh Peek's Slicehost gem is full of Capistrano recipes that are much simpler and much more organized than Deprec. Nothing in there is especially Slicehost-specific, either.

Share Improve this answer

Follow

answered Mar 16, 2009 at 19:52

Jim Puls
81k • 10 • 75 • 78

0

I'm hosting my new apps with Apache2 and Passenger on Ubuntu Hardy. Seems like the easiest and best option for most scenarios. I have just joined Slicehost.com for that purpose. They seem to get good reviews and have the most competitive prices of first class hosts.

I can't really endorse them yet because I'm a new client but the set of guides and range of support options is impressive.

What you are not mentioning is how large and popular your app is/will be. This criteria could affect the decision process.

Share Improve this answer

Follow

answered Nov 11, 2008 at 9:00

allesklar
9,570 • 6 • 38 • 53

0

Capistrano + Deprec for actually setting up my stack on Ubuntu and physically managing the deployment.

Nginx proxying to Mongrel clusers for the server architecture. It isn't the newest, bleeding edge technique but it works well, it is getting well-documented, and it is very, very high performance even when working on small VPSes. Assuming you haven't borked the application you

can Slashdot a 128 MB Slicehost VPS and it just keeps coming back for more.

Having said that: there were a *lot* of gotchas the first time around, until I figured out how Nginx actually worked. After that its amazing -- like a little Apachelet with a slight Russian accent.

Share Improve this answer

Follow

edited Mar 16, 2009 at 19:53

Jim Puls

81k ● 10 ● 75 ● 78

answered Nov 12, 2008 at 2:38

Patrick McKenzie

4,076 ● 26 ● 23