AD113: Speed Up Your Applications w/ Nginx and PageSpeed

Eric McCormick

Thanks for coming!

Agenda

- Who's Talking?

- Who is this for?
- What are you selling?
- Where can I use this thingamajig?
- When can I adopt it?
- Why does this benefit me?
- How should I get on this bus to crazy town?

- Demos

Who's Talking?

Eric McCormick

Twitter: @edm00se

Blog: edm00se.io

XPages developer since '11

Java developer on and off since '02

Experimented with WebSphere and DB2 in college

MWLUG 2015
Transforming Collaboration Through Innovation
Ritz-Carlton - Downtown Atlanta
August 19-21, 2015

What Is My Goal?

1. A harmonious development environment for myself and any development staff at my company
2. A unified ~~theory~~ practice of development, for consistent cross-platform and cross-app development behavior
3. Making all application logic and data consistently accessible across separate, but organic, systems
4. Making contracted development work more easily performed
5. Automating everything I can

For my users: applications that "work well" (respond quickly, easy to use, don't hang around for lunch).

I Want It All

Who's This For?

Developers!

Who's This For?... pt.2

Specifically, web developers.

… specifically web developers that care about / have access to / can bribe an admin for access to the full application stack

… aka web application developers

.... web app devs that want blazing fast application speed for their users, peace and harmony, and for dogs and cats to live together.

Who's This For?... pt.3

What you should know already:

- fundamentals of web application and XPages "stack" development
- differences of between the runtime generated markup of HTML (created by a our JEE environment) and static HTML assets (and CSS, JS, images)

What you should get out of this:

- an understanding of how a couple excellent freely available tools can make your applications more performant for your users
- hand off TLS/SSL cert. handling from your app server (Domino)
- provide finer grain (and automated) control over static web resources and their:
  - caching (what, how long, no-cache paths)
  - rewrites (concatenation)
  - compression

What you won't get out of this:

- how to grow a beard
- many other things that go without saying

What Are You Selling?

The same as usual...

Better Web Applications!

What Are You Selling?... pt.2

What is it all about?

- Segregated application design/development is good for my users, my company, and myself (and the other developers that we interact with)
- I blogged about development with HttpServlets in XPages (NSF-local ones, without OSGi plugins)
- I even did a Notes in 9 video on the subject (a second one bridging the gap between back-end and front-end development should be out soon)

What does it all mean?

- I have more (than “average” XPages application) static web assets
  - not just CSS and JS files, but even HTML files (who knew!)
- Just as we set an HTTP API (e.g.- RESTful API endpoint) to Cache-Control: No Cache, etc., we can cache our static assets in the user’s browser for faster loading times
  - for more complex setups you can even host the static assets on a dedicated server separate from a Domino/XPages server, for faster loading times yet (and map in an Domino/XPages API assets as an api route, e.g.- /xsp, /api, etc.)
  - this is not covered here (watch my blog)

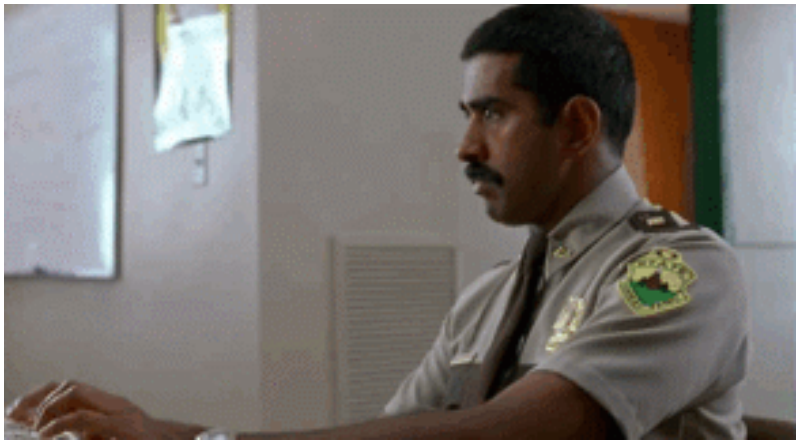
What Are You Selling?... pt.3

How can I achieve all this developer + user bliss?

3 Pieces:

1. Segregated Application Design (aka- have static assets to cache)
2. Reverse Proxy (to forward our API requests without cache, respond with cached static assets)
3. Tweak, fiddle, and “enhance”

Google PageSpeed!



Wait, What?

Google PageSpeed Tools:
https://developers.google.com/speed/pagespeed/

“The PageSpeed tools analyze and optimize your site following web best practices.”

Flavors:

- Nginx or Apache webserver modules (or IIS or Apache Traffic Server ports)
  - relies on the PageSpeed Optimisation Library (PSOL)
- PageSpeed Insights (online, freely available, assesses both mobile and desktop versions w/ score)
  - demo volunteer?
- PageSpeed Chrome Extension (great for non-public use cases, lots of the same info as Insights)
  - demo local
- ~~PageSpeed Service (commercially provided by Google, all PageSpeed modules, filters, infrastructure)~~
  PageSpeed Service deactivated 03-August-2015

DEMO: **PageSpeed Insights**
(online, freely available, assesses both mobile and desktop versions w/ score)

demo volunteer?

DEMO: **PageSpeed Chrome Extension**
(great for non-public use cases, lots of the same info as Insights; also PSI tool, installable via npm)

Where Can I Use This Thingamajig?

In front of your application server
(it's a reverse proxy, that's where they live)

preferably on the same "box" (it doesn't *have* to be, but strongly advised).

NGINX is pronounced *engine X.*

When Can I Adopt It?

As soon as you (or a sufficiently bribed admin) can run the install and do the necessary config (which I outline and demo in the “how”).

Why Does This Benefit Me?

In your well structured app (you amazing developer), you now have static assets which are fully cacheable; or not, it's all configurable, like a choose your own adventure app server.

Your user's application performance will only increase due to the faster responses.

You can now manage, off of your application server (Domino):

- cache control
- Gzip compression
- HTTPS/SSL/TLS certificates
- authentication (see Jesse Gallagher's blog posts on "arbitrary authentication", etc. with Nginx)

Side benefit: if you're addicted to fast apps, you've now opened the door to forward the API calls mapped to the Domino/app server and host the purely static assets off of Nginx (or Apache) for even faster applications!

One Ticket to Crazy Town Please

How do we do this? You guessed it, with Nginx + PageSpeed module!

PageSpeed module is available for:

- Apache -binary or source builds for Debian/Ubuntu and CentOS
- Nginx, *nix only, no Windows :'-(-source builds only for Debian/Ubuntu and CentOS
- IISpeed (port), https://www.iispeed.com/

With Nginx, you must build Nginx from source specifically configured with PageSpeed support (aka- we're hooking in deep).

Crazy Train… pt.2

Follow the Google PageSpeed install and build from source instructions for Nginx on *nix (I'm using an Ubuntu 64-bit 14.04.2 LTS distro).

What do the instructions say? The short, short version is:

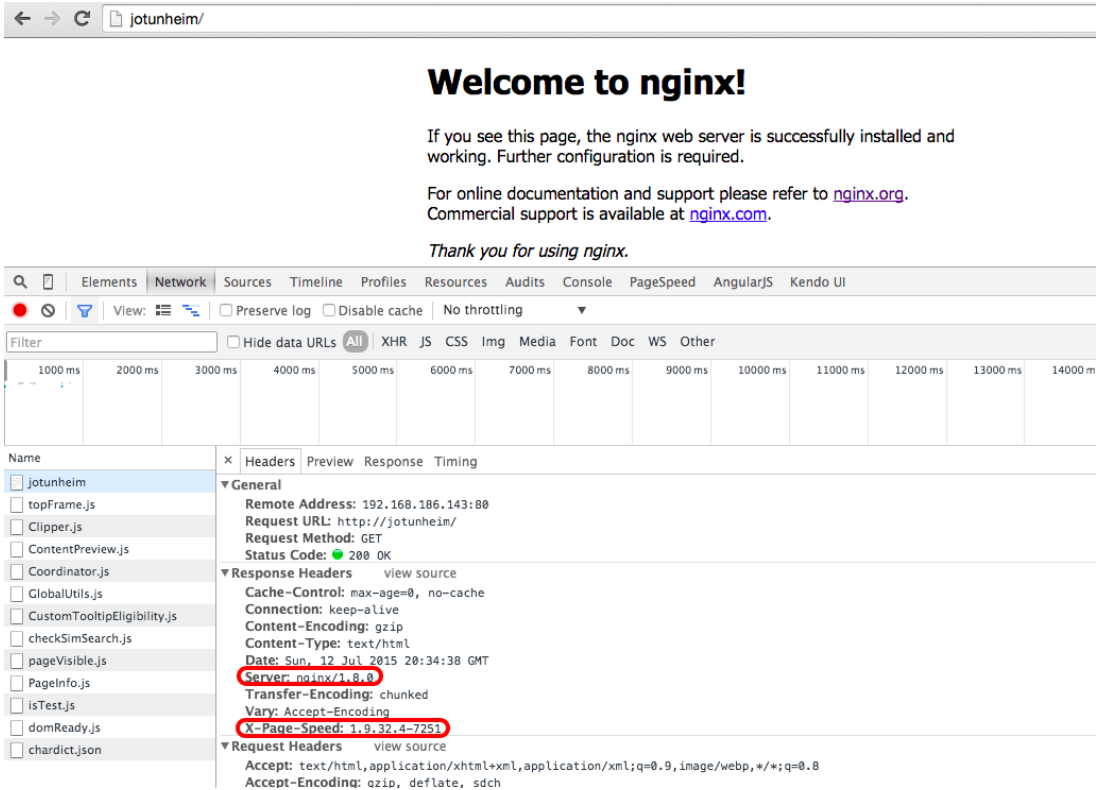
- install a couple dependencies
- download the PageSpeed (and PSOL) sources
- download the Nginx source
- build PSOL, then the PageSpeed module
- configure Nginx w/ PageSpeed and build

```
thor@jotunheim:~/nginx/nginx-1.8.0$ ./configure --add-module=$HOME/ngx_pagespeed
/ngx_pagespeed-release-${NPS_VERSION}-beta
```

My install put things into */usr/local/nginx/*.

Crazy Train… pt.3

Once that's all done… it's alive!



Config Anatomy

Simple (simple) HTTP server config

/usr/local/nginx/conf/nginx.conf

```
worker_processes  1;
events {
    worker_connections  1024;
}
http {
    include       mime.types;
    default_type  application/octet-stream;
    sendfile        on;
    keepalive_timeout  65;
    server {
        listen       80;
        server_name  localhost;
        location / {
            root   html;
            index  index.html index.htm;
        }
        error_page   500 502 503 504  /50x.html;
        location = /50x.html {
            root   html;
        }
    }
}
```

Demo 1:
a Domi-faux server behind Nginx w/ reverse proxy

psst Eric, you want the nginx.conf_simpleProxy file you saved earlier

Demo 2:
a reverse proxied server w/ PageSpeed doing all the PageSpeed things

psst Eric, you want the nginx.conf_PageSpeed-ified file you saved earlier

Demo 3:
a server w/ Nginx handling the HTTPS/SSL cert

psst Eric, you want the nginx.conf_withSSL file you saved earlier

Note on HTTPS Certificates

Demo 3 has a self-signed TLS certificate, generated via openssl.

Want to go that route, try this tutorial from Digital Ocean, or this one for Apache.

Alternatives:

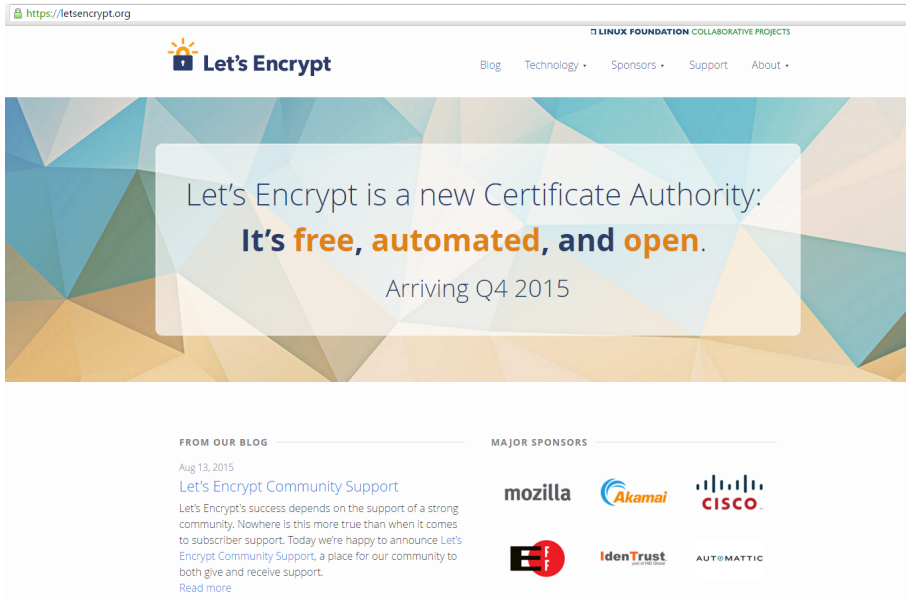
- paid for certificate (e.g.- from GoDaddy, etc.)
- no HTTPS
- letsencrypt.org

Let's Encrypt:

- sponsored by EFF, Mozilla, and others
- free, automated, open source
- coming soon

How Soon?

- First certificate: Week of September 7, 2015
- General availability: Week of November 16, 2015



Note on HTTPS Certificates, pt.2

Using Let's Encrypt.org:

- can automatically configure Apache 2.x
- support for Nginx 0.8.48+ mostly working (for auto config)
- can always just generate files into local directory (regardless of auto config)
- works through Let's Encrypt's own root and intermediate CAs
- configurable RSA key length
- can generate quickly
  - from a *nix (Ubuntu, Debian, Fedora, Centos 7, Mac OS X) environment
  - using Docker
  - instructions on starting use
- you can revoke a cert (one you've created for that server)

Once created, either drop it in place of your self-signed or (for the fearless, until general release) let it auto configure for you.

Not a magic solution, still essentially an intermediate "self-signed" certificate, but better than having your users click through the scary "untrusted" screens.

Some FAQs

- Why is PageSpeed not rewriting (combining) my CSS, JS, etc. files?
  link because it's not supposed to… if filters like *collapse_whitespace* are working, then...
  "Your resources (images, css, javascript) aren't cacheable. If PageSpeed sees `cache-control` headers such as `nocache` or `private` it will not rewrite the resources."

  We tell it to forget the Cache-Control and Expires headers with *proxy_hide_headers*.

- Why is Nginx failing to start as soon as I enable mod_pagespeed?
  you most likely have forgotten to build either PageSpeed or PSOL from source, failed to configure Nginx correctly, or malformed your conf syntax (use semi-colons at line ends!)

- What if I want the same config on a dev box, but want to see changes?
  do as this SO answer suggests and use a .htaccess file with *ModPageSpeed off*

- But what's it all cost?
  not a thing, it's all free (except for the infrastructure to run an Nginx instance; which isn't much)

- Do I really need PageSpeed for this?
  No! You can achieve most, if not all, that the ngx_pagespeed modules does without PageSpeed. It just makes it easier.

DEMO: CLOUDFLARE©

(time permitting)

A taste of a freely available, developer friendly way of doing a lot of this, without a DIY server.

Questions?

Write better, faster, stronger apps! We have the ability.