# 

# Managing Continuous Delivery at Enterprise scale

Jason Whittington

Equifax Workforce Solutions

Nginx.conf 2017

September 5, 2017

Portland, Oregon

## DRAFT

This document is a preview draft and is not complete.

## Introduction: Our Team Mission

Delivering a rapidly growing and evolving portfolio of applications and partner integrations places challenges on any enterprise environment. To help manage that growth Equifax Workforce Solutions has spent the last five years transforming the organization to be more nimble. The “Business” side of the house now works closely with the “Technology” side via Agile and Kanban processes. Teams have adopted sophisticated build and test infrastructure that allow them to rapidly react to business desires and ship code.

Our team was tasked with improving the delivery process, which can loosely be defined as “what happens after the code is built”. To deliver the value the business wants as fast as they want we would need to provide

* **Infrastructure Provisioning**
* **Fast and repeatable code deployment**
* **Rock-solid uptime and no-downtime deployments**
* **Rapid feedback**

Taking our lead from books like Gene Kim’s *The Phoenix Project* and Jez Humble’s *Continuous Delivery* we set out to implement a Continuous Delivery environment where application code could “flow like water” between environments. We want to make frequent small changes instead of infrequent large changes as the risk is smaller and doing something frequently means you get better at it than doing something infrequently.

* ***Frequent Small Changes are better than fewer large changes***

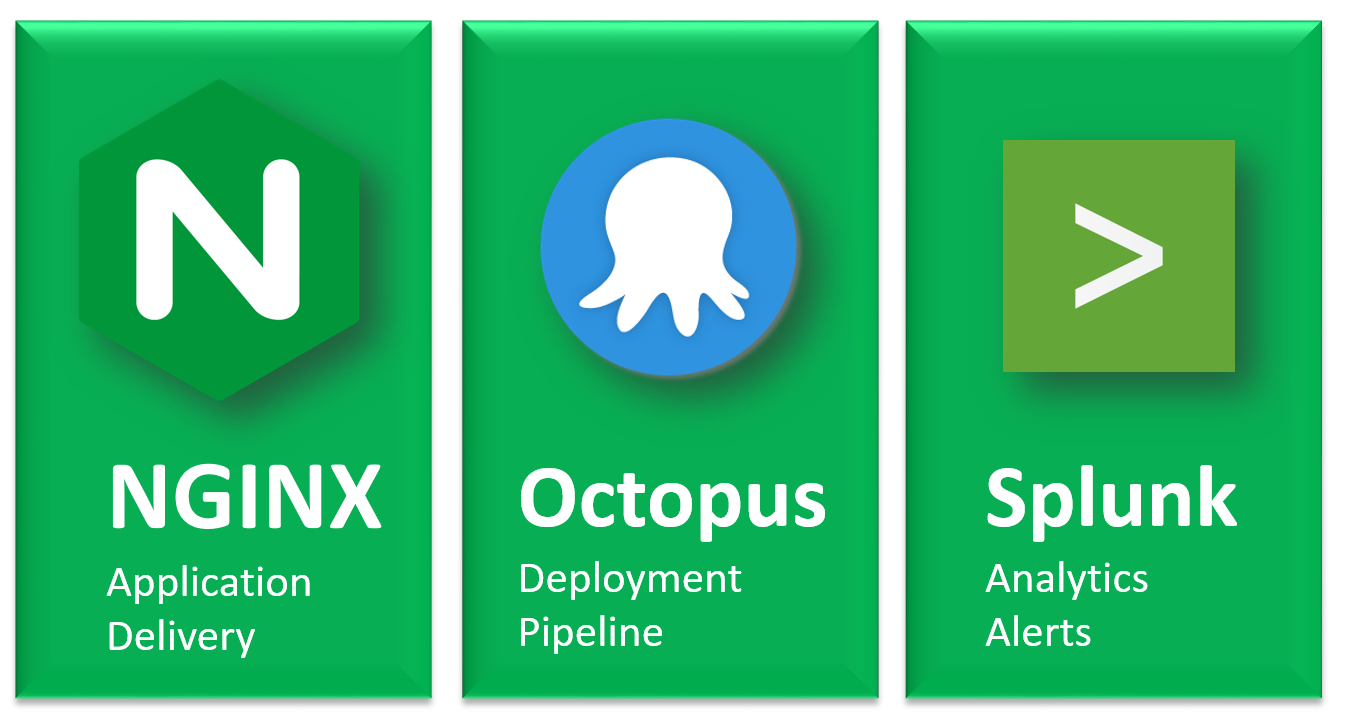
## Computing at Equifax

Equifax has a global presence with computing power running on multiple data centers spread across five continents. Many environments, many networks and lots of firewalls. Different technologies reign in different regions and groups, so the environment as a whole is extremely heterogenous. Equifax and Workforce Solutions manage sensitive financial data, much of which is falls under access and audit controls like Sarbanes-Oxley (SOX), PCI DSS, and FISMA. Access to the external Internet is tightly controlled, proxied, and audited. These controls add some extra wrinkles to the way we manage applications – we’re not as free as a small startup to just “put it in the cloud”.

### Equifax Workforce Solutions

Equifax Workforce Solutions has a large investment in the Microsoft.NET platform as well as a smattering of other technology stacks. Applications are commonly deployed to Windows servers and use a variety of back-end databases. As a result our tools tend to be .NET-centric, but as .NET moves onto Linux our tools are following along with it.

## Our platform: NOS



Our team surveyed many tools that claim to manage various parts of the Continuous Delivery process. We found that there is a sweet spot of “easy to get started” and “powerful and extensible”. Many tools were one or the other but we found that three tools in particular were strong in both dimensions and worked very well together: NGINX Plus, Octopus Deploy, and Splunk.

### NGINX Plus

NGINX Plus proved early on to be easy and flexible. We have been able to deprecate other routing solutions in favor of NGINX. While open-source nginx is a great product we chose to license Nginx Plus due to features like live health checks, dashboards, and the upstream\_conf dynamic configuration API. We rely on those features to implement blue-green deployment and similar strategies to deploy code to “inactive” environments, test and then “flip”.

### Octopus Deploy

We began using Octopus Deploy to manage Windows applications. We found it easy to use and extremely powerful – as of this writing we have deployed code changes something like 25,000 times in the last year and that number is increasing quickly. Octopus Deploy recently added support for CentOS 7 which means we can now deploy NGINX configurations the same way we deploy application logic and many annoying management tasks have become much simpler.

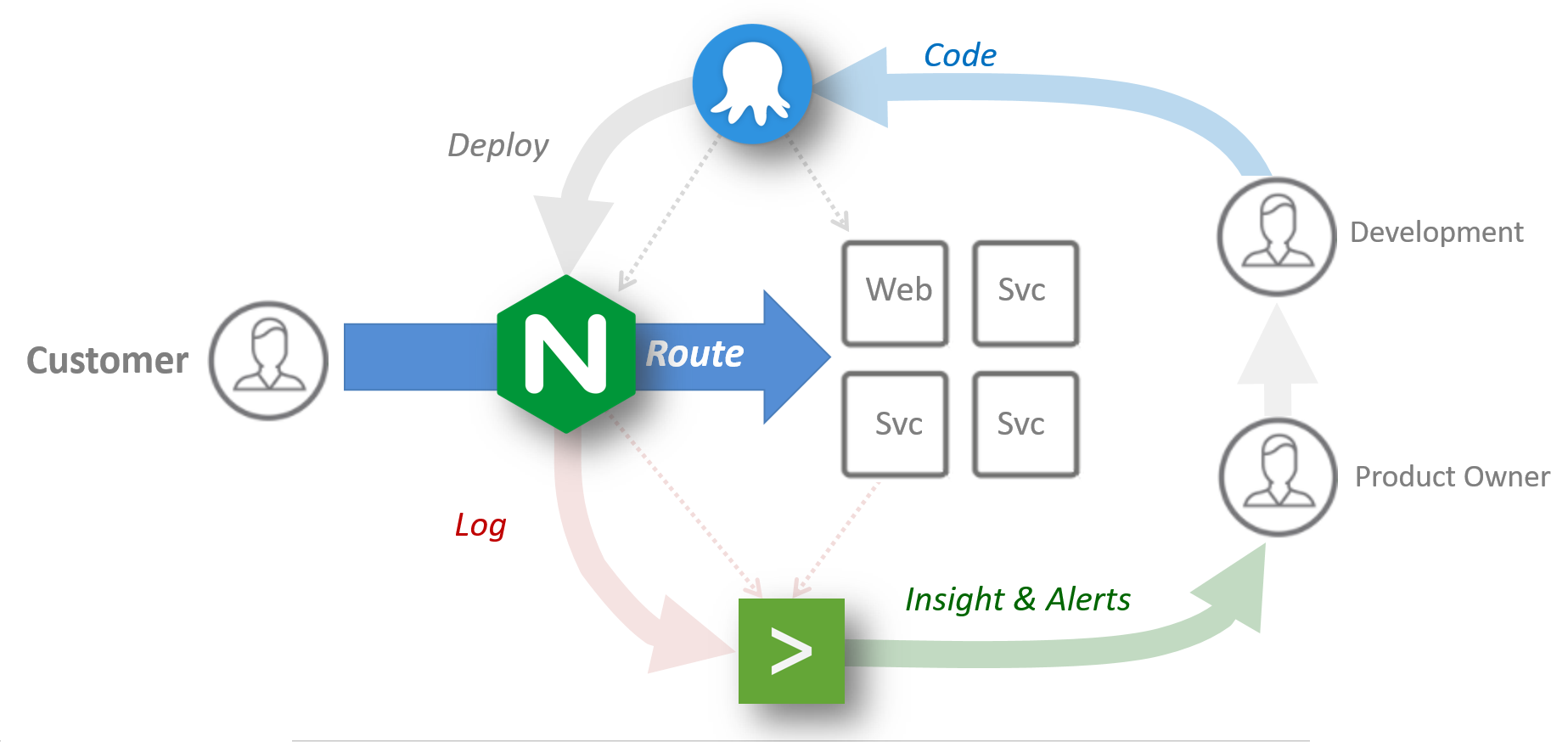
The most compelling features for us are:

* Easy integration with build tools via standard “nuget” packaging
* Easy to define environments that contain machines, then tag machines with roles
* Well-defined “Release” and “Lifecycle” models
* Easy variable replacement
* Easy and extremely flexible “Process”
* Powershell / bash extensibility
* The product itself is extremely API driven, so we can integrate it with other tools (like nginx)
* Integrated deployment of application and routing changes

### Splunk

Splunk is a well-known log management tool and it’s a natural partner to nginx and Octopus Deploy. Sending NGINX logs to Splunk lets us get insight on how properties are being accessed but also lets us get near-real-time alerts when something “funny” seems to being going on.

## The NOS Workflow

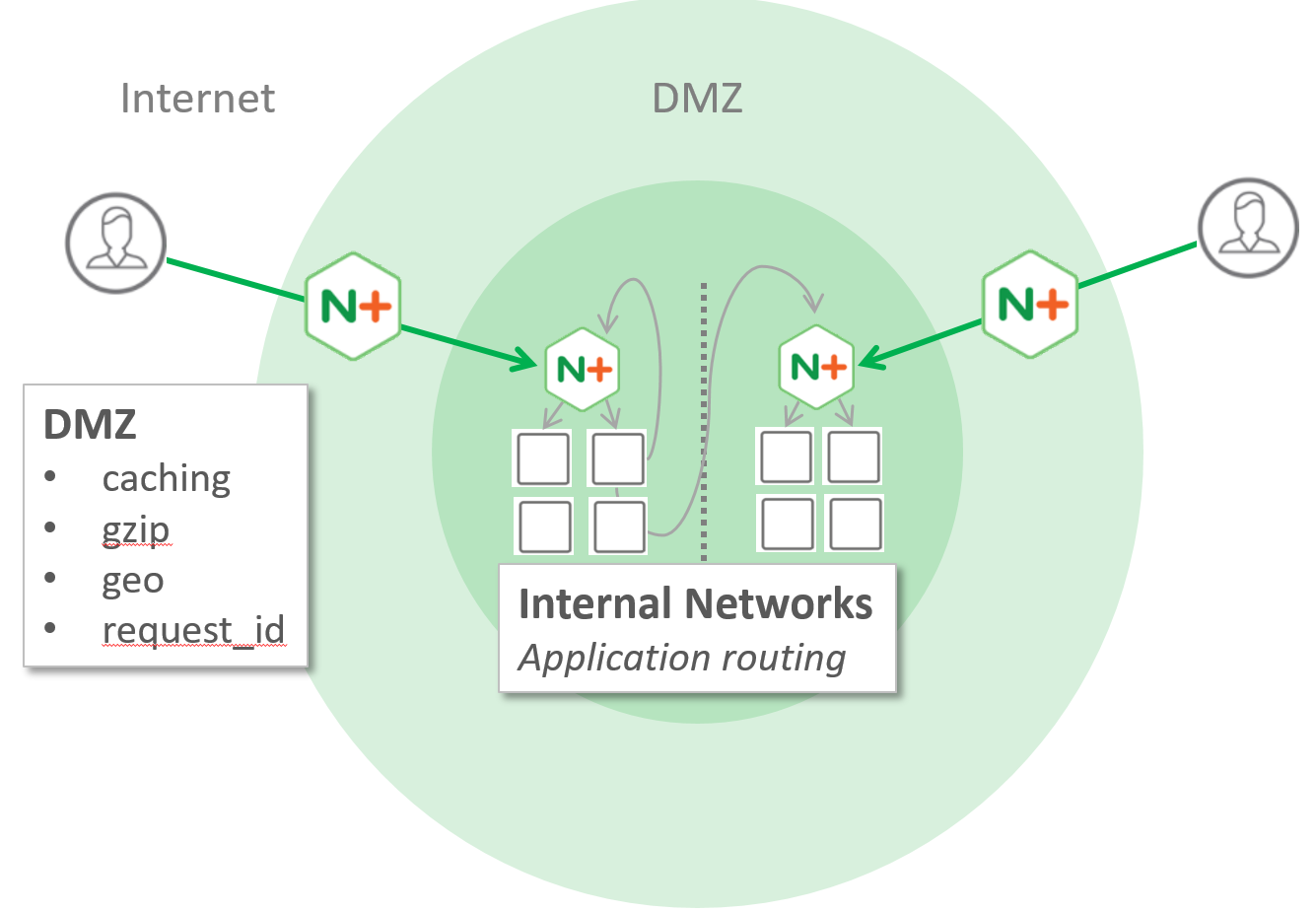
Octopus, NGINX and Splunk enable the workflow shown below. NGINX routes traffic mostly in a reverse-proxy role. Octopus Deploy is used to deploy application changes as well as configuration changes on corresponding NGINX instances. Octopus Deploy’s support for environments, variables and custom processes makes it pretty easy to define nginx virtual servers and upstreams and then select an environment and deliver the configuration (potentially to multiple nodes in multiple roles at the same time). Applications, Nginx, and Octopus Deploy can all send their logs to Splunk which provides visualizations, insights and alerts that feed back into the development cycle.

## Adapting MRA patterns

The ***Microservice Reference Architecture*** patterns published by NGINX were useful starting points. The Router Mesh pattern was closest to the pattern we found most useful. When applied to an Enterprise space a two-tier model emerges naturally with NGINX in “DMZ” and “application routing” roles. The roles are managed differently – “DMZ” role servers exist to

* Filter raw external traffic into something useful
* Improve performance via gzip and caching
* Provide services for applications via additional http headers (e.g. geo, request\_id)

Application servers provide blue-green and general internal traffic routing. This is illustrated below. Notice than calls between services on the Internal network do not traverse the DMZ; internal traffic is confined to internal NGNIX nodes.



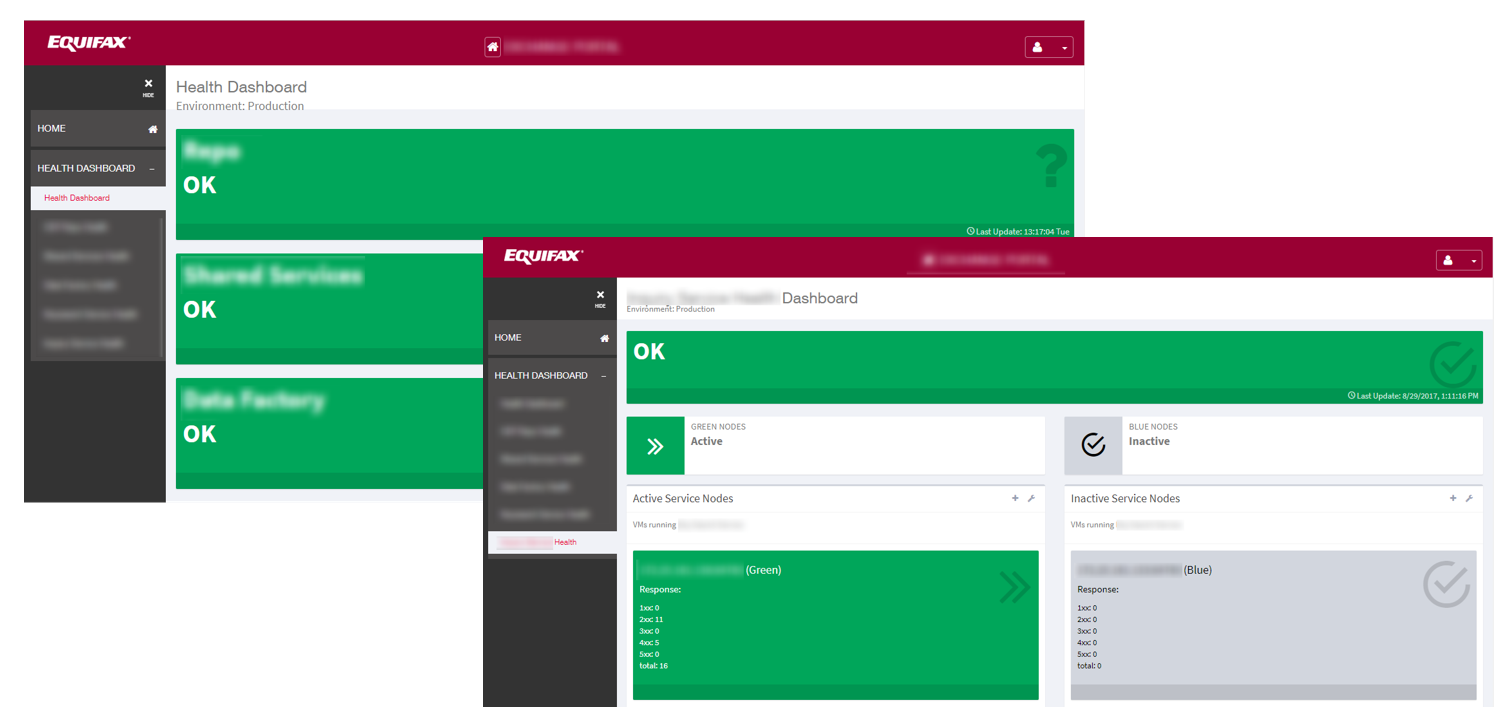
See the whitepaper *Blue-Green Deployment with NGINX* for more detail.

## Lessons learned: Visibility is crucial

When we began our journey we had some issues that were exacerbated by the fact that

* It wasn’t obvious what the configuration was supposed to be
* It wasn’t obvious what the configuration *actually was*

We learned that we really needed to build custom dashboards for important applications that query health and display integrated viewers. The nginx+ /status endpoint made it easy to integrate nginx information into these dashboards. The screenshots below show some of the dashboards we created. Once they were up and running we saw some dramatic improvements in the the speed at which we could develop new features and respond to incidents.



## Visibility: Splunk

Splunk has proven to be an extremely useful partner to NGINX+. We use it to build dashboards to see near real-time usage reports and to fire alerts when something goes amiss. Splunk dashboards don’t provide a full solution (They won’t show you desired and current configuration) but they are good for general analytics. Alerts have proven invaluable – we have alerts set up to watch for a variety of conditions and alert us when things go amiss. These alerts have saved us from several incidents since we adopted them.

## General challenges

NGINX Plus has proven to be an excellent product but there have been a few things that we have found

* **httponly/secure cookie management**  
  There’s no good way to indicate that nginx should mark all cookies with the httponly and/or secure attributes.
* **State management**Managing the state of NGINX instances leaves a bit to be desired. The Dashboard isn’t quite adequate and managing “state” files is a pain. These difficulties are multiplied when deploying to a cluster where cluster coherency is a problem. NGINX could stand to have better cluster awareness, particularly in “active/active” scenarios where multiple clusters are active.
* **Access Control**The NGINX API is extremely useful but securing it is challenging in Enterprise environment. Certificate-based solutions work fairly well but access control is coarse-grained, Lack of LDAP integration makes it hard to integrate NGINX access management with Enterprise entitlement management systems.

## Future Efforts

* **Higher-order integrations**We will be working on extending this model to integrate public cloud with internal networks and to implement higher-order traffic management. One of the problems we still are interested to hear about is implementing a “Single source of truth” for managing configuration. We are evaluating tools like Consul, etcd and Spring Config.

## About the author

Jason Whittington is a 25-year veteran of the Software industry and is currently employed as a Continuous Delivery Architect at Equifax Workforce Solutions in St. Louis, Mo.

## Equifax and Equifax Workforce Solutions

Equifax is an Atlanta Fortune 500 (3.4+ billion on revenue, 10k+ employees). Equifax made its name as a credit-reporting institution but its business is rapid expanding to all manner of financial insights. Equifax recently adopted the slogan *Powering the World with Knowledge*™ to reflect this expanded mindset. Equifax Workforce Solutions (based in St. Louis, Mo) is a great example of this expanding scope. Workforce Solution has a rapidly diversifying list of products and services and is seeing a rise in partner integrations that allow it to deliver tailored customer solutions far beyond traditional credit reports.