

Miguel Pilar's Resume

I am Miguel Pilar, a Web Developing code monkey turned "DevOps/Inframagical Architect". This is my resume written in ReStructuredText and formatted using sphinx, and not even using a custom sphinx theme.

To see it compiled please click [here for single-page HTML](#) or [here for multi-page HTML](#) or [here for PDF](#). Thanks for looking.

Note

This is a preliminary experiment of writing a resume with sphinx, as such it is in flux and under development until this note disappears. *It will probably never disappear.*

The format of this resume is inspired by [rdegges'](#) and [wraithan's](#) but using sphinx for a resume seems serendipitous.

Contents

About Me

I am a developer/hacker that has very many interests, both in life and technology. I love playing around with technology I've had experience with system administration, networking and know enough to shoot myself in the foot in very creative ways.

As far as languages ~~go~~ are concerned, I have tried many, have worked professionally with a range of languages, including C#, Python, go and some ruby and have had to debug in few other languages. My favorite language is by far python.

For better or worse I am also a typical PC gamer, playing all sorts of game, sucking at a lot of them too but most importantly enjoying myself.

Personality

By nature I try to learn many things even if it is to know just the basics, to that end I've explored things ranging from blacksmithing to basic home repair to welding. I feel that the more breadth of knowledge one has the easier it is to see problems and solutions in different angles.

I have had a few hobbies through the years, the more recent ones are:

- 3D priting
- Sous Vide Cooking

Irrational Thoughts

If I have a constant recurring irrational thought is that I am uncomfortable with PHP, unlike other languages I have seen I have been unable to find beauty in a PHP program. Perhaps I have not been introduced to good PHP, but even the language design itself seems inelegant to me. This is not to mean I don't dive into PHP programs to diagnose them or see how they tick, but that it is well outside my comfort zone.

Technical Skills

This is a list of my stronger technical skills. Through my years I've made many different small projects with varied purposes. Among them:

- 6+ years experience with Python and C#
- Have played with many others like Go, Ruby and even AutoIT

C#

When not programming in Python for work I've had to *maintain* C# applications be them web, console or desktop. I've had some experience working with ASP.NET MVC.

Python

I've spent the last few years programming in Python professionally and otherwise. Most of my experience with Python is centered around writing web applications and management scripts.

requests

I will be frank, I don't actually remember how to use urllib for actual web requests (still use it to parse urls, etc) and would have to spend a few minutes figuring things out again, requests is just so perfect.

Django

For most web projects that interact with databases I enjoy using django. I feel The django app ecosystem is powerful and make it very easy to create a complex project quickly and efficiently.

Flask

For smaller web projects and prototypes I turn to Flask, nimble enough to create small projects quickly but powerful enough to expand them into larger projects. Specially true of projects that don't need an ORM to be tightly integrated.

Falcon

A great framework for middleware and/or pure APIs, it's speedy and very useful for APIs and generally "glue" projects.

Celery

When doing tasks asynchronously celery provides a quick and very well designed platform to do them with.

Redis

I've used Redis as a very powerful hash and key/value store. I am always impressed with it's speed and stability.

PostgreSQL

This is my preferred relational database, but I have only played around with it's more advanced and powerful features like writeable CTEs.

jQuery

I don't always have to work with javascript, but when I do, I enjoy using jQuery for it. At least where browser compatibility is concerned.

Git

My preferred source control mechanism, but not the only one I have experience with. I am by no means an expert but know the commands used on most workflows.

AWS

I am familiar with most AWS services, that is to say I've used most of them, while I am not an expert at all of them I understand where they are best used for certain jobs. I have scripted some aspects of our operations using python and boto.

munin/monit

I have used munin and monit to monitor and script some self-healing into hosts.

StatsD

While it has been replaced with Cloudwatch, StatsD has been a go-to tool for metrics collection.

Grafana

A great way to create a dashboard for metrics.

Varnish

Varnish is for my go-to tool for caching full web pages or even fragments through ESI. I've spent time tweaking configurations and adapting Varnish to cache pages in a legacy application that had no concept of browser caching.

Traffic Server

When a caching layer does not need to have as many customizations of the request/response as can be done with Varnish I have used Apache Traffic Server which is an extremely powerful caching server with a tiered architecture.

haproxy

While there are other load balancing options, I find that haproxy has the most flexible balancing algorithms.

spread/wackamole

This project is outdated but it's still useful to run N master clusters of load balancers when necessary (it allows haproxy to scale horizontally by sharing IPs among a cluster).

Projects

Admittedly I do not have a huge portfolio of projects to offer, and only one public project per-se, and I think that's one of my largest shortcomings: I seldom do my own projects, instead I do small hacks and fixes to other projects and submit pull requests.

Public

gigya-server-lib

I have only one public project, a port of [Gigya's](#) SDK to python called [gigya-server-lib](#). The library is not entirely pythonic to maintain it closer in usage to the original. It has been abandoned since Gigya developed their own.

As an aside, while working on this library I discovered a security bug that was disclosed to Gigya.

Assorted pull requests

I've made small contributions to the following projects:

- **Mongo C# Driver:**
 - [CSHARP-204](#): found, diagnosed and fixed a concurrency issue in the mongodb driver for C#
 - [CSHARP-200](#): created a patch in response to a feature request for the C# Driver
- [AL-Redis](#): [a quick patch](#) in response to having seen a TODO on the code. (I never got the chance to use AL-Redis but found the TODO while browsing the code.)

- [mumble](#): No link but some client functionality for saving last connected channel and some other settings c.2005, patch submission was via email/IRC back then.

Private

- Architected new version of our Newspaper CMS for availability and scalability using .NET MVC 4.0 and .NET Web API 2.0, implementing an MQ based job system for better backend user satisfaction. The FrontEnd was architected to use the subset of ESI available in varnish for a better user experience
- Implemented a read-only API translation middleware using .NET Web API 2.0 and AutoMapper to provide a proxy API so that legacy API responses could be cleaned up and made more consistent:
 - Corrected usage of mixed case, mixed language keys in json response
 - Coerced output into consistent output models (previously two representations of a model presented two different ways would have inconsistent keys)
 - Corrected HTTP status responses (legacy API always responded with 200 even for empty responses)
- **Implemented JIRA webhook processor in Falcon that:**
 - Triggered and synchronized PagerDuty alerts for multiple projects/teams
 - Managed internal subproject issue dependencies to create a global project that gave visibility to project managers into many teams and projects
- Led switch to AWS for various products, worked closely with developers to help make the transition smoother and diagnose any problems
- Created guidelines with networking and datacenter operations teams to influence their architectural choices
- Wrote a real time log aggregator and analytics service in python to monitor the most popular news articles on [elnuevodia.com](#) and [primerahora.com](#)
- Configured varnish to do things nobody should ever do on varnish to make up for a third party application's lack of caching
- Managed the cloud infrastructure initiative for GFR Media using AWS and Rackspace Cloud.
- Architected solutions for web deployments using various AWS services

Work Experience

GFR Media

DevOps/Senior Web Developer (2008-present)

- Architected and implemented several critical infrastructure changes necessary to scale operations to new, quickly changing, traffic requirements
- Moved company away from vendor lock-in to a more flexible and heterogeneous environment that could rapidly adapt to requirement changes
- Implemented shift into cloud environments for hosting some parts of the web infrastructure
- Designed auto-provisioning caching layers for largest local news sites and their supporting APIs
- Led shift from monolithic applications towards more stateless API-based applications
- Responsible for final architectural and performance review for applications

Insight Communications

IT/Web Developer (2007-2008)

- Developed information service (411) database application using django and solr
- Part of team that maintained 150+ workstations and servers for incoming and outgoing telemarketing campaigns

NeXellence

Junior Web Developer (2006-2007)

- Wrote screen scraping software for a legacy application in order to adapt it for web use.

Vcom3D

QA/System Administration Intern (2003-2006)

- Maintained web and local servers (3 total), including web, file storage and version control servers.
- Maintained fleet of testing computers and cloning environment
- Performed final QA on 3D client application and installer
- Configured and maintained VPS based (Linode) web and email server
 - Immediately regretted trying to set up a spam resistant email server on 64MB of RAM