TERMINAL LABS

Joseph D.M. Nix
US Citizen

Last updated – June 16, 2019

Summary

I am a Python developer and entrepreneur who has grown a Python consultancy firm of highly skilled personnel. I have a decade of experience in web development and DevOps. I have successfully architected and implemented entire stacks for clients by myself, and I have integrated well with larger teams. I am also active in the Open Source world, maintaining and contributing to several projects. I have a background in physics and mathematics, and have a deep love for science and technology. I am a very pragmatic person, and enjoy finding creative and effective solutions to problems, and seeing them come alive in the world.

SKILLS

Computer Science/ Programming:

Full-stack Development, Team Management, Python Development, DevOps, Python Optimization, Linux Application Development, Database Development and Administration, Server Administration, Algorithms, Pair Programming, Computational Complexity Theory, Data Compression, Encryption, TNO (Trust No One) data storage design, and Data Security.

Physics:

Astrophysics, Cosmology, Stellar Structure, Stellar Evolution, High Energy Processes, Plasma Dynamics, General Relativity, Condensed Matter Physics, Quantum Mechanics

Mathematics:

Calculus, Ordinary and Partial Differential Equations, Statistics and Probability, Number Theory, Linear and Modern Algebra, Real and Complex Analysis, Tensor Analysis

Propulsion Systems:

Modeling of variable specific impulse thrusters for the purpose of mission planning and exploration of high energy processes for use in future high energy density propulsion systems.

Modeling of Various Electric Propulsion Systems:

Ion thrusters, Resistojets, parallel and coaxial Pulsed Plasma thrusters, Electromagnetic thrusters, Pulsed Electromagnetic thrusters, Magnetohydrodynamic thrusters

--Modeling done using Mathematica, FORTRAN

Miscellaneous:

Networking, Finance, Patent Law, Securities Law, Contract Law, software licensing, Latin, Spanish - limited working proficiency, French and Italian - elementary proficiency

Programming Languages:

Python	9 years
Bash	7 years
HTML5/CSS3/JS	7 years
SQL	6 years
Wolfram Language (Mathematica)	3 years
Fortran 95	3 years
Ruby	2 years
C	2 years
Lisp	1 year
C++	1 year
NSIS	1 year

Computer Hardware:

- I have assembled, repaired, and upgraded many computers. I thoroughly research my builds before purchasing the component to avoid problems, from faulty cell processors to a CPU lacking instruction sets needed to allow virtualizations.

- I have troubleshooted many hardware problems to find and replace dead components, and know how to use a voltmeter and soldering iron.
- I have installed network infrastructure and equipment for clients. This includes: installing wireless Internet systems, configuring routers, running / terminating cable, setting up multiple monitor workstations, building racks and rack-mountable servers, configuring multi-boot systems.

Operating Systems:

Linux, Debian, Linux Mint, Ubuntu, Redhat, CentOS, ClearOS, DSL, OS X, OpenBSD, FreeBSD, SUSE, Mandriva, Arch

I began using Linux in 2007 by experimenting with many distributions in rapid succession until landing firmly on Ubuntu 7.04. I used Ubuntu steadily through 12.10, then used Linux Mint for several years. Currently I prefer Ubuntu with Cinnamon as my workstation setup, with Ubuntu, Debian, and CentOS for various server needs.

Databases:

PostgreSQL, MySQL, SQLite, Memcached, Elasticsearch, Solr

Web Servers:

Nginx, Lighttpd, Gunicorn

System Administration / DevOps:

SSH, Coreutils, OpenVPN, OpenSSL, AWS (EC2, S3, Cloudfront, Route53), DigitalOcean, Rackspace, VirtualBox, Vagrant, SaltStack, Chef, Fabric, Libcloud, OpenStack, nmap, Wireshark / Ethereal, Bind9, Gearman, Supervisor

Python Web Frameworks:

Django, Flask, Lektor, Tornado, Mezzanine

Other Notable Python Packages:

Lektor, NetworkX, PyInstaller, PiCloud, BeautifulSoup, django-cumulus, Lazythumbs, PIL, NumPy, SciPy, ctypes

Web Languages:

HTML5, CSS3, Less, JavaScript, jQuery, CoffeeScript, Twitter Bootstrap

Version Control:

Git, Mercurial, Subversion, Github, Bitbucket

Open Source Software:

Maintainer of: Lektor, Rambo
Contributor to: Lazythumbs, QRFactory

WORK EXPERIENCE

- Python Consultant -- 2008 – present

  Worked with a range of clients, led remote (across the globe) and local teams, development projects, designed and installed server/networking gear, disaster recovery, interacted with UI/UX designers/design firms, developed projects "from scratch", inherited projects midway through development, used modern DevOps tools to manage infrastructure, spent a large amount of time researching emerging technologies, and written a lot of code.

- Founder, CEO of Terminal Labs -- 2007 – present

  Bootstrapped a software consultancy firm that has grown every year.

- UAH Physics Senior Capstone Project ------------------------------------ 2007 – 2008

  Created a theoretical model for the super nuclear density neutron star core. Neutron stars are predicted to have a core that is denser than the nucleus of an atom. There are no generally accepted models for physical interactions at these densities. I created a model for the neutron star core that takes this into account by assuming that the core of the neutron star core is a gas of microscopic black holes. It was shown that there is a mass and density of microscopic black holes for which Hawking radiation allows for there to be an equilibrium between the radiation pressure and the gravitational potential of the black holes in the gas and the outer mass of the neutron star.

PROFESSIONAL PROJECTS

PYTHON CONSULTATION PROJECT

Client (name withheld) -- July 2018 – present

Client, a young company with a web platform with an active, niche community. This company had a PHP/MySQL stack on a single server when I started working for them. I was brought on to extend the feature set of the website, and begin the process of migrating its stack to something more scalable, stable, and Flask based.

Technologies used:

Python, Flask, SQLAlchemy, sqlacodegen, AWS (EC2, Cloudfront, Lambda, S3), Zappa, Asyncio, BeautifulSoup, New Relic, MySQL, SaltStack, PHP, Apache, Supervisor, uWSGI.

My involvement included:

- Used sqlacodegen to generate an approximate models file for an existing database, and then introspected the database by hand to correct the generated models file. This was then used as a starting point for use of SQLAlchemy within the Flask app.
- Scraped various web data with BeautifulSoup. Processed this data into updated database records and structured S3 data, for easy consumption.
- Used Asyncio to both fetch web resources and post data to S3.
- Created Flask application to serve an API, providing data from the database and scraped resources.
- Helped debug PHP, Apache, and server issues.
- Used Zappa to host Flask apps in AWS Lambda instances that served an API.
- Used Zappa to run scheduled tasks in AWS Lambda instances.
- Used SaltStack to orchestrate EC2 instances and entirely control them. These instances became production and staging Flask web servers, database servers, task servers, and PHP servers. Previously there was a single server only, with Apache and MySQL running on it. The PHP servers are being slowly phased out.
- Introduced staging and development environments, and testing practices.

PYTHON CONSULTATION PROJECT

Cox Media Group -- June 2014 – July 2016

Cox Media Group, a subsidiary of Atlanta-based Cox Enterprises, is an integrated broadcasting, publishing and digital media company. The company's operations include 15 broadcast television stations and one local cable channel, 86 radio stations, four metro newspapers, more than a dozen non-daily publications and more than 100 digital services. At this time, the company operated one of the largest production deployments of Django in the world.

Technologies used:

Python, Django, Django-South, jQuery, Vagrant, VirtualBox, Ubuntu, Postgres, HTML5/CSS3, Less, Git, Virtualenv, Grunt, Hudson/Jenkins, sitespeed.io, Review Board, JIRA, Lazythumbs, tox, New Relic, WriteCapture, Postscribe.

My involvement included:

- Developed for the Medley CMS used by the Austin American Statesman (Austin, TX), the Atlanta Journal and Constitution (Atlanta, GA), the Dayton Daily News (Dayton, OH), and the Palm Beach Post (Palm Beach, FL). The Medley CMS is used by hundreds of content producers and Medley sites are viewed by millions of users across the country.
- Helping successfully achieve launch goals of the JP2 in the Fall of 2014
- Worked as a Full Stack developer for the Newspaper Teams working on the Janus Redesign Project - JP2; a redesign of Medley that was deployed in late 2014, first for the redesign Team 1, then for the Performance team.
- Working in a large team of dozens of other Python and Web Developers, QA Analysts, and Business Representatives.
- Helped the Quality Assurance (QA) analysts by picking up some of their workload when they were backlogged.
- Converted PNG/PDF web page design mock-ups files into fully functional interactive HTML5/CSS3/Less/JS based web pages and UI features.
- Provided general code maintenance and refactoring.
- Created new Django HTML templates which were rendered into new customer facing web pages.
- Worked with clients automatic ad serving systems to rearrange how and where certain ads were included in customer facing web pages.

- Worked with HTML5/CSS3/Less/JS to make many small improvements to the look and feel of customer facing web pages.
- Worked with Virtualbox and Vagrant to set up multiple VM guests for testing/development and other various uses.
- Fixed many bugs. These fixes ranged from small visual problems to complex backend issues.
- Wrote documentation for internal code and systems such as story pages.
- Created a new staff member page for content producers
- Extended the functionality and uses of the Taxonomy system in Medley, a sophisticated mapping system of various objects within the Medley CMS, aimed to make the administration of the websites more powerful and flexible for the content producers.
- I did the research, design, and planning, as well as initial coding for a set of modifications to the Taggit app in Medley, so that it can auto generate links to custom objects whenever matching text is found in story text.
- Wrote a Bash script to repeatedly test the download speed of a webpage and output statistics on the results such as number of tests, average, and standard deviation.
- Improved a private fork of the OSS repository called Lazythumbs (render-on-request thumbnailing for django) to add an additional action that resized images while preserving aspect ratio and never cropping.
- Submitted a pull request (which has been merged) of a large set of CMG updates of Lazythumbs back to it's open source repository on GitHub.
- Given the task of auditing and revising the use of many Javascript scripts in order to speed up the rendering of JP2 sites. This involved carefully weeding through all JS scripts rendered by every page by default, and deciding if they can be moved down in load order, moved to different templates to be selectively loaded, or removed entirely. I deprecated, updated, replaced, and consolidated scripts and libraries. I had autonomy in this project, and set my own priorities based on initial and successive investigations.
- Refactored delayed, asynchronous JavaScript execution by replacing custom implementation of WriteCapture with Postscribe.
- Worked with the performance team to decrease page load times, page load time perception, and otherwise optimize user experience from a performance perspective. New Relic was used to generate granular metrics and identify specific pain points. We saw a 600% decrease in page load time.

PYTHON CONSULTATION PROJECT

Modavanti -- March 2013 – August 2014

Modavanti is a New York based E-commerce company that sells sustainable fashion from around the world. They bring an element of environmental and social awareness to the runway. In 2018 Modavanti was acquired by DoneGood.

Technologies used:

Python, Django, Django-South, jQuery, Rackspace, elasticsearch, Ubuntu, Chef, Ruby, HTML5/CSS3, Less, Git, Mercurial, Virtualenv, Gearman, SimpleJSON, authorize.net, Fabric, Memcached, curl, supervisor, BeautifulSoup.

My involvement included:

- Lead Developer - including being the primary technical advisor to management, the go-between for non-technical management and developers, managing up to 7 other developers in multiple locations around the planet at a time, and developing myself.
- On-call website maintenance
- Managing technical priorities in Bitbucket's issue tracker
- Reviewed, approved / denied, and collaborated on, and merged pull requests from other team members
- Reworked how product images are downloaded from the Rackspace CDN.
- Worked with Elasticsearch to enhance how customers search for products and JavaScript/HTML and views to display them in a stylish and technically challenging way
- Worked to make the stack more stable, and to plan and implement perpetually better ways to simulate and test the stack to minimize risk to the stack. Upgraded stack components and added new packages.
- Revised the project documentation to better describe how to perform a full-stack restart/cloud infrastructure transfer.
- Supported the client's cloud infrastructure both the production VMs and the staging VM. This included restarting various services, upgrades, whole OS restarts, reconfiguring with Chef.
- Worked with Rackspace's API to their CDN in order to batch upload image files.

- Implemented an algorithm to track the popularity of items and reorder them on the shop page based on a Maxwell-Boltzman distribution to class items in 10 tiers of popularity, and reorder them to show the most popular items first. This also had a manual override for items the management wanted to actively push to a certain tier.
- Worked with the Python Gearman client to create worker scripts that were deployed to the background worker server for various tasks such as tracking item popularity and sending emails.
- Maintained the Supervisord and Chef configuration that controlled the Gearman background worker processes.
- Worked with Virtualbox to set up multiple VM guests for testing/development and other various uses.
- Worked with autherize.net Python API to create Modavanti virtual currency
- Worked with UPS Python API
- Used Chef to deploy and manage the whole set of servers, including DNS, Memcached, workers, Elasticsearch, load balancers, Staging, Frontend, Chef-backup
- Implemented a promo code system to allow for discounts to promotional items on the website
- Implemented a Modavanti 'credit' system to allow for users to have credits that can be spent as bulk discounts to any purchase they make.
- Taught management how to properly interact with Django Admin to modify their website

PYTHON CONSULTATION PROJECT

mobiusmed.com --- June 2013 – December 2013

Mobius Medical Systems offers innovative software for modern radiation oncology. Specifically, they offer products that help Medical Physicists analyze radiation treatment plans for accuracy. Intensity modulated radiation therapy allows for radiation to more selectively target cancer cells. However, plan error tends to increase with modulation. To fully leverage modulation, plan error must be controlled as much as possible. In 2018 Mobius was acquired by Varian Medical Systems.

Technologies used:

Python, CouchDB, PyCUDA, NumPy, Flask, Genshi, Foundation, jQuery, Coffee-Script, couchdb-python, HTML5/CSS3, Selenium, PhantomJS, Git, couchable, Virtualenv.

My involvement included:

- Ported an algorithm originally written in NumPy to C, to be called in Python via ctypes to aid in hitting performance benchmarks.
- Refactored code to reduce memory usage redundancy in NumPy array objects.
- Worked with PhantomJS to reconfigure the DPI resolution of exported PDF files.
- Built a dedicated GPU server

PYTHON CONSULTATION PROJECT

agyield.com -- January 2012 – March 2013

AgYield offers a very useful tool for risk management for farmers that allows them to understand the costs and risks related to their crops in order to keep track of and project profit scenarios.

Technologies used:

Python, Django, AWS, MySQL, Ubuntu, Mercurial.

My involvement included:

- Worked with the CEO and CTO of the company that offers the agyield.com service to understand and revise the development roadmap.
- Performed a broad ranging review of the client’s technical operations.
- Worked with AWS to help maintain the client’s production servers.
- Studied commodities trading and crop insurance.
- Worked with the client’s ETL systems that regularly downloaded and processed data purchased from a third party.
- Interpreted various properties of features, options, and crop insurance types to correct algorithms that calculated ROI.
- Optimized/Refactored crop insurance calculation functions in Python.
- Ran a security audit on their stack.
- Made their website cross-browser compatible.

- Modified the account section of the website to allow users to change their passwords and other account credentials - this was previously set by agyield personnel.

PRO BONO PROJECT

Disaster Recovery for client -- February 2013

Client was a small company that had unrecoverably lost it's source code, and only had a single copy on their production server running their webapp. They didn't know how to access this copy safely, or if it was a complete repository or just a single state of the code. I safely logged in to their running server (leaving it unchanged and running), found the source code which was thankfully a full git repository, and backed this repository up onto GitHub.

SIDE PROJECTS

OPEN SOURCE PROJECT

Lektor -- 2017 - Present

Lektor is a flexible and powerful static content management system for building complex and beautiful websites out of flat files — for people who do not want to make a compromise between a CMS and a static blog engine. I have been a Lektor user for a while (see the section on our own website). I began contributing to Lektor and in June 2017 I became it’s maintainer. Lektor is a popular open source project, and my involvement as its maintainer has a very broad scope, including contributing, creating releases, and managing issues, pull requests, and the general direction of the project.

I also own/manage several Lektor plugins and a theme. I try to stay active within the Lektor community, particularly on our Gitterand & IRC channels, and on GitHub, helping people with Lektor and adjacent tools like Jinja and sometimes submitting issues or pull requests on adjacent projects, such as the PyPI Warehouse, setuptools, and Pallets projects.

Technologies used:

Lektor, Click, Flask, Jinja2, React, Webpack, virtualenv, Conda, Black, setuptools / python plugins, PyPI Warehouse, AWS S3 / Cloudfront, GitHub API & Pages, Markdown, rST, Imagemagick, Travis CI, Appveyor, PyTest.

OPEN SOURCE PROJECT

Inflation -- 2017 - 2018

Inflation was an experimental tool that allowed you to simply orchestrate and provision a cluster. It is built on top of **Rambo** (listed below). With Inflation you can with a couple commands you can create a variable number of virtual machines. This cluster can either be local on a powerful enough computer or created with cloud based virtual machines such as DigitalOcean Droplets or AWS EC2 instances. The cluster is easily configured to have subsets of VMs created and provisioned differently for specialized purposes, so subsets may have varying hardware and configured with different software, all networked together for powerful parallel computing.

The work is open source and can be found here:
https://github.com/terminal-labs/inflation

Technologies used:

Python 2 & 3, Bash, DigitalOcean, Vagrant, Rambo, VirtualBox, SaltStack, Hadoop, HDFS, Ambari, Dask, Pandas, Jupyter, Distributed, Ubuntu, Yarn.

My involvement included:

- Configured inflation to install Hadoop with Hortonworks Ambari.
- Configured inflation to install Hadoop HDFS.
- Wrote Jupyter notebooks that ran on a Hadoop Edge Node to distribute data storage and computation, using Anaconda Dask and Distributed, Pandas, and Hadoop HDFS.
- Used Jupyter's Magic to use HDFS to import data and in situ configuration of Edge Nodes.
- Imported ~100 GB of data from a Digital Ocean Volume onto HDFS and processed it with Python.
- Cycled clusters repeatedly.
- Created a 64 node Hadoop cluster, and a 120 node general cluster.
- Automated setting of FQDNs, hostnames, and SSH keys.

OPEN SOURCE PROJECT

Rambo -- 2015 - Present

Rambo is a tool that allows you to simply provision new virtual machines on any provider and have them be nearly identical. To accomplish this Rambo makes heavy use of Vagrant and it's various plugins for different providers. As a design philosophy, we really want to have development environments that are as similar as possible to production environments. This streamlines development, catches bugs, and helps smooth production releases. So to achieve this, we wanted to automate the provisioning of these environments. For this we used SaltStack. Among other things, SaltStack is great for provisioning machines. With it, we can provision a local or remote instance pretty easily.

The work is open source and can be found here:
https://github.com/terminal-labs/rambo

Technologies used:

Python, Bash, Ruby, AWS EC2, DigitalOcean, Vagrant, VirtualBox, LXC, SaltStack, Jira, Bitbucket, Git, Mercurial, Postgres, Debian, Bitbucket Pipelines.

My involvement included:

- Worked with numerous Vagrant plugins each from different developers and organizations with different design philosophies.
- Created virtual machines on AWS EC2, DigitalOcean, VirtualBox, & LXC
- Worked with complex SSH key vendor and authentication systems.
- Maintaining multiple code repositories in Mercurial and Git.
- Smoothed out many inconsistencies between several different cloud providers.
- Created SaltStack configurations with over 50 salt states and custom grains.
- Factored out all potentially custom data into grains to make using the salt states easily customizable for unique projects.
- Wrote some custom Vagrant code in Ruby.
- Worked with LXC on Ubuntu 16.04/17.04.
- Worked with Bitbucket Pipelines.
- Created custom Virtualbox images for use with Rambo.
- Wrote documentation and tested against its instructions.

OPEN SOURCE PROJECT

terminallabs.com --- 2011 - Present

terminallabs.com is our own company website. Through it we have taken the opportunity to use several technologies. Though we have a fairly simple website, the body of work that supports our website continually evolves. As much as our time allows, we habitually use this to experiment with new technologies and methodologies, while striving to maintain ease of maintainability. Currently, our website is using the Lektor as a CMS, Framework, and Static Site Generator.

The work is open source and can be found here:
https://github.com/terminal-labs/tl_web

Technologies used:

Python, Django, Mezzanine, Flask, Lektor, AWS (EC2, S3, Cloudfront, Route53), DigitalOcean, Vagrant, Salt Stack, Jira, Bitbucket, Git, Mercurial, Postgres, Debian, Buildbot, Jinja, Less, jQuery

My involvement included:

- Iteratively developing the frontend codebase working with HTML5/JS/LessCSS and Django or Jinja templates
- Iteratively developing the Python server backend with Django / Mezzanine / Flask. This included creating models changes, views, and modifying the CMS for altering content in admin.
- Using Lektor as a web-framework and static site generator and hosting this static site over AWS S3 & Cloudfront with SSL.
- Maintaining multiple code repositories in Mercurial and Git.
- Creating a repository that heavily uses Vagrant and Salt Stack to create and manage both development and production environments on either AWS or DigitalOcean. With a simple `vagrant up` with a couple optional arguments you can spin up local or cloud-based dev instances or cloud-based production instances.

OPEN SOURCE PROJECT

QRFactory -- March 2017

QRFactory is a simple utility I made because I could not find a free QR code generator that can create a stylized QR code of the unmodified data I input, overlay an image, and maintain high resolutions. At best, some websites could do some of those things, but I got tired of looking so I wrote my own. I created an open source Python project that is a utility that takes any input string and creates a stylized QR code for it, and then takes an image SVG, and places this image in the center of the QR code. The image manipulations are all done on SVGs, so they do not lose their scalable quality. bitbucket.org/terminal_labs/qrfactory

Technologies used:

Ubuntu, PIP, VirtualEnvWrapper, QR Code spec, Segno, svgutils, svgwrite, Bitbucket, Git.

PYTHON DEMONSTRATION PROJECT

PiCloud/iPhone pi distribution statistics app -------------------- April 2013 – March 2013

This nice iOS app is a simple demonstration of calling the PiCloud cluster via an iPhone app. From the app, the user can search for an inputted string in several billion digits of the decimal expansion of Pi. In 2013 PiCloud was acquired by Dropbox.

Technologies used:

Ubuntu Server, Apache Cordova, PiCloud, OS X, Xcode, Python Tornado, Nginx, HTML5/CSS3, jQuery, Mercurial.

My involvement included:

- Developed a functional backend server in Python Tornado for receiving/processing AJAX data sent from the iPhone app.
- Studied and worked with PiCloud and their API in order to become proficient with their environment.
- Configured a Nginx server to support the “mockup viewer”.
- For this project we needed to test on physical iOS devices, so I used multiple developer tools from the Apple ecosystem (iOS provisioning portal, app signing, physical device deployment)
- Wrote backend code to compute statistics on substring distribution.

EDUCATION

- 2005 - 2010

University of Alabama in Huntsville (UAH): B.S. in Physics & Mathematics Courses: General Relativity, Quantum Mechanics I-II, Astrophysics I-II, Thermodynamics, Mechanics, Electric Propulsion, Complex Analysis, Number Theory, Algebraic Structures, Foundations in Mathematics, Ordinary and Partial Differential Equations, Modern Physics, Probability, Linear Algebra, Mathematical Methods in Physics. My Senior research project was a theoretical model for neutron star cores that attempts to account for their super nuclear density.

PATENTS ISSUED

- High Specific Impulse Superfluid and Nanotube Propulsion Device, System and Propulsion Method.
  - Patent 8,991,150. 31 Mar. 2015.
- Web Insulation System, Valve for a Web Insulation System, and a Storage Container Using the Web Insulation System.
  - Patent 8,991,636. 31 Mar. 2015.
- Apparatus and Method for Anonymously Presenting Targeted Advertisements and Desirable Media Content in Association with a Virtual Currency.
  - Patent 9,001,979. 7 Apr. 2015.
- Web insulation system, valve for a web insulation system, and a storage container using the web insulation system (continuation).
  - Patent 9,279,540. 8 Mar. 2016.

AWARDS AND ACCOMPLISHMENTS

- Member of the American Institute of Aeronautics and Astronautics Nuclear & Future Flight Propulsion Technical Committee.
- Participant in UAH honors program

VOLUNTEER AND SOCIAL

- Volunteer teacher and tutor
- Blood donor
- Member of American Institute of Astronautics and Aeronautics
- Regular attendee and occasional presenter at local Python and DevOps other tech meetups

Some Programs & Tools That I Have Experience With

Software Programs:

- Emacs
- Mathematica
- GIMP

Misc Tools:

- Gensh
- Jinja
- Git
- Mercurial
- Subversion
- 7-Zip
- Unix Utils (ssh, rsync, awk, grep, ...)
- Lastpass
- TrueCrypt
- PhantomJS
- Virtualenv
- Less
- Grunt
- SpinRite
- GParted
- Virtualenv
- jQuery

Web Apps:

- Jira
- Redmine
- Bitbucket
- Github
- Pivotal Tracker
- Review Board
- Google Drive / Docs
- Dropbox
- VersionOne

Python Frameworks and Python Tools

- Lektor
- Django-CMS
- Django
- Python Tornado
- Flask
- PyInstaller
- PySide
- PiCloud
- PIL
- Lazythumbs
- Mezzanine
- BeautifulSoup

DevOps:

- Docker
- VirtualBox
- Vagrant
- SaltStack
- VMWare
- OpenVPN
- Gearman
- Fabric

Operating Systems:

- Linux Mint
- Ubuntu
- Fedora
- Debian
- Damn Small Linux
- Apple OS X
- FreeBSD
- CentOS
- ClearOS

Database Systems:

- PostgreSQL
- CouchDB
- MySQL
- SQLite
- Elasticsearch
- Memcached

Servers:

- Nginx
- Lighttpd
- Green Unicorn

Compilers:

- GCC
- Dev C++
- g95
- F95
- Shed Skin

Amazon Web Services (AWS):

- S3
- EC2
- AWS CLI
- Route 53
- CloudFront

Platforms and Analytics:

- Adobe Analytics
- Adobe DTM
- Amazon Web Services (AWS)
- New Relic
- Omniture
- Rackspace