

Matthew Olenik

olenikm@gmail.com
<https://github.com/mattolenik>
<https://olenik.io>

B.S. Software Engineering
Rochester Institute of Technology, 2011

Summary

- Polyglot developer of 9+ years with a focus on backend and infrastructure engineering, with side interests in front-end and machine learning
- Focused on building developer tools and improving the software lifecycle, from code to deployment and support
- Passion for correctness in software, advocate for full end-to-end tests and test-driven development
- Consistently identifies engineering problems and solutions, owning projects from spec/design phase, build, test, document, deployment and support.

Experience at Microsoft (Xbox Live), VMware, Curalate, eSentire

eSentire Inc

DevOps/Infrastructure Team

Software Engineer

April 2018 — Present

Developer Tools

- Designed, implemented, and tested escli, a swiss army knife for engineering tasks
 - Git repo composition that works as a replacement for submodules, simpler for devs to understand and use, with a focus on CLI usability
 - Enforces deprecation of repos to encourage keeping dependencies up to date
 - Templating system for creating new projects using Go's text/template engine
 - Custom, simple package manager for internal tools, cross platform, powered by Artifactory
 - Used by all of engineering for composing repos of any complexity
 - Led the effort to open source it under the name git-require (public release TBD)
- Created standard project template for writing new Go CLIs with e2e tests

Terraform and Automation

- Created solution for multi-account VPC networking CIDR block management with NetBox
 - Wrote Terraform provider for NetBox, focus on e2e tests against real NetBox
 - Made upstream contributions to NetBox for race conditions
 - Automated deployment and update of NetBox into AWS with Terraform and Jenkins
- Owned design and implementation of infrastructure for esINSIDER (<https://www.esentire.com/resources/library/esinsider>)
- Automation for generating Terraform code from existing prod resources then importing the state
- Created Terraform state management CLI tool to work around state bugs in Terraform 0.11
- Jenkins scripted pipeline work, such as refactoring old multi-stage Jenkins jobs into scripted pipelines
- Provided support for the rest engineering in their day-to-day use of our development pipeline

Open Source

- Spearheaded public engineering blog, coordinated with other engineers to create content
- Pushed for open sourcing of escli under the name git-require (TBD public date)
- Made upstream contributions to NetBox (<https://github.com/netbox-community/netbox>)

Curalate, Core Services

DevOps Engineer

Platform team owning core services, dev/CLI tools, infrastructure

April 2017 — February 2018

- Set standard for AWS Lambdas with scaffolding for writing new, well-tested functions in Python
- Improved dev tools for working with Terraform, built Docker mixin system to reuse code between QA/PROD (Scala)
- Overhauled ECS cluster management with blue/green cluster deployments and autoscaling
- Automated ops tasks (e.g. backups, syncs, migrations) with Terraform and AWS Lambda
- On-call experience with DataDog, Loggly, CloudWatch

VMware, Photon Controller

Software Engineer, MTS III

Open source multi-tenant IaaS (<https://vmware.github.io/photon-controller>) for on-premise management of VMs and containers

January 2014 — May 2016

Revamped Build Pipeline & Developer Workflow

- Simplified Docker deployment pattern for dev and test machines, eliminating manual testing
- Reduced the 'vagrant up' time from 20min to 30sec, dramatically improving developer productivity
- Reduced the size of the product deliverables (a VM image) from 9gb to 1gb
- Streamlined build process of VM deliverables using Jenkins and Packer

Built Integration Tools

- Designed the API for and implemented the Go client SDK (<https://github.com/vmware/photon-controller-go-sdk>), used in the PC CLI (<https://github.com/vmware/photon-controller-cli>)
- Wrote the plugin (<https://github.com/vmware/bosh-photon-cpi>) for integration with the Cloud Foundry PaaS bosh (<https://bosh.cloudfoundry.org>)

vCenter Core Team (2014 — 2015)

- Wrote the interface between UI and vSphere API for new HTML5-based installer in v6.0
- Fixed bugs and implemented features in a large C++ codebase, worked closely with ESXi

Microsoft, Xbox Live Commercial Services

Software Development Engineer

June 2011 — January 2014

- Refactored legacy Xbox 360 services to run on modern CI/CD with Autopilot (<http://research.microsoft.com/apps/pubs/default.aspx?id=64604>), enabling integration with new services for the 2013 holiday release
- Implemented, tested, and deployed features in services for: user accounts, billing and offering, in-game DLC, TV-on-demand, and others
- Owned user email preference services, consisting of a REST API, and background jobs for exporting and migrating bulk data
- Implemented revenue reporting using COSMOS (like Hadoop but with a SQL-like language)

Selected Projects

- hclq (<https://github.com/mattolenik/hclq>) — a tool for querying and modifying Terraform HCL files, similar to jq
- unity-web-neural-network (<https://github.com/mattolenik/unity-web-neural-network>), simple C# neural network library optimized to run in the browser as JS under Unity 3D

Selected Skills

- Best with C#, Go, Java, C++, Bash, adept with Scala, Python, Ruby — prefer functional languages
- Backend service development (mostly REST), CLI and dev tools, occasional frontend in JS
- Managing complex build pipelines involving Jenkins, Packer, Docker, Terraform, and other tools
- Building tested infrastructure — writing infra code with tests, as production code
- On-call engineering, debugging live applications, working production issues