

Ben Dean

Software Engineer specializing in automation and tools to streamline the build and deployment pipelines.

[Email](#) / [LinkedIn](#) / [GitHub](#) / [Keybase](#)

TL;DR

Just the highlights in case you don't want to read the whole thing.

- Over 14 years of experience on various DevOps, Build, Tools teams.
- Extensive experience with various DevOps tools and frameworks, IaC, CI/CD, etc.
- **Automate Everything** - I do not like manual processes. Automation reduces the opportunity for human error.
- Ability to learn whatever is needed. Gained from years of reading other people's code to help them figure out what's going wrong and solving complex problems. Also from picking up new programming languages and technologies.

Technical Experience

Lead Software Engineer @ [Finvi](#) (formerly known as Ontario Systems) (2008 - Present)

Jan 2012 - Present - Various "DevOps" teams and projects

Starting around the beginning of 2012 I joined a team whose goal was to automate the build, deployment, and infrastructure of new software products. This started with deploying systems to private hosted environments (VMWare) and transitioned over time to deploy system to public cloud services (AWS). There are a number of technologies and tools used some dating back to that initial work in 2012.

- [Jenkins](#) - build server, deployment orchestration
- [Chef](#) - configuration management and automated software provisioning
- [Terraform](#) - infrastructure automation
- [Packer](#) - building images to deploy in AWS
- [Docker](#) - building containerized applications
- [Kubernetes](#) - cluster for running containerized microservices
- [Helm](#) - deploying software releases to Kubernetes

In addition to the "devops" work, our team was often tasked with various side projects. Some notable projects are:

- Web app to organize product install and upgrade testing across multiple combinations of versions and supported platforms.
- Simple artifact repository with additional APIs to support various needs of our software.
- Web app to track delivery of software products to premise customers.
- CLI for launching containerized versions of [InterSystems](#) Caché/Ensemble/IRIS products in Docker.
- Web app and auto-updating framework for downloading and updating various in-house tools.

Jan 2009 - Dec 2011 - Large scale .NET-based enterprise web application built on top of Microsoft Dynamics CRM and hosted in IIS 7.

- As a member of tools/build team I was responsible for understanding the system architecture and providing utility programs, libraries, and installers that enable the rest of the team to be productive and deliver a cohesive, working product.
- The technologies used in this product include .NET 3.5, Microsoft Dynamics CRM, SharePoint, IIS 7, C#, WCF, WF, Silverlight, SQL Server, WiX, T4 Code Generation, MSTest, and Rhino.Mocks.
- Microsoft Team Foundation Server was used as a source control repository and task tracking system.

Sept 2008 - Dec 2008 - Large scale Java-based enterprise web application using the Spring Framework and hosted on the JBoss Application Server.

- On this product I worked on a team responsible for providing the framework code that connected the entire application. We used Spring's Inversion of Control framework to wire together the different parts of the larger web application.
- The technologies used in this product include Java 6, Spring, JBoss, Hibernate, JMS, CXF, JUnit, and EasyMock.
- Subversion was used for version control and Maven was used to manage software artifacts.

Software Developer / Instructor / Graduate Assistant @ [Ball State University](#) (2004 - 2008)

- Teaching Assistant for several computer science courses - running labs, grading assignments, writing automation.
- Research Assistant on several projects with published research papers.
- Teaching internship team-teaching interdisciplinary course on computer game design.
- Instructor for intro to computers summer courses.
- Contract software developer with Mathematics Department.

Skills

I am familiar with many programming languages and technologies. Some of these are listed earlier in the work experience section, but I'll highlight some of the more recent skills here.

Programming languages

- [Ruby](#)

Used in our [Chef](#) cookbooks, various automation tools, and some simple web applications.

- [Go](#)

Used in microservices we deploy to [Kubernetes](#) and in various automation tools. I'm familiar with many popular open source go libraries and SDKs.

- [HCL](#)

The configuration language of Hashicorp's [Terraform](#) and [Packer](#) tools. Also recently used in [Docker bake](#) for building images.

- [Make](#)

Most of our builds use have a `Makefile` that defines various build targets. These are designed to work the same on the build server as in local environments so developers can build their software using the same tools as the build server.

- [Bash](#)

Used to script various repetitive tasks and as part of our build and deployment automation.

- [Groovy](#)

Used in [Jenkins Pipeline](#) scripts and libraries of shared build pipeline functionality.

- [ObjectScript](#)

The programming language used by the [InterSystems](#) platforms that many [Finvi](#) products are built upon.

- I often have to help product teams figure out what's going wrong in their projects which involves reading and sometimes writing code in various languages. These include Python, C#, C++, Java, Perl, etc.

Technologies, frameworks, and platforms

- [AWS](#)

The cloud platform used for [Finvi](#)'s products. Services used include (but are not limited to) EC2, ELB, VPC, Route 53, EKS, ECR, IAM, S3, Aurora/RDS, Lambda, KMS, and SSM.

- [OCI](#) - Oracle Cloud Infrastructure

We also deploy products to Oracle's Cloud, OCI. Similar cloud offerings to AWS are used in OCI.

- [Kubernetes](#)

We use Kubernetes, in particular AWS's Elastic Kubernetes Service ([EKS](#)) and Oracle's OCI Kubernetes Engine ([OKE](#)) to deploy our container-based applications and microservices.

- [Helm](#)

Helm is the tool we use to release code to our kube clusters. All of our microservices and web applications that run in kube have helm charts in their repos. These charts also make use of some library charts we make available using a private [ChartMuseum](#). We also make use of a number open source charts to deploy things like [Jenkins](#), [NGINX Ingress Controller](#), [ExternalDNS](#), and [cert-manager](#).

- [Jenkins](#)

Jenkins is our build server, used both to build the [Finvi](#) products and deploy them to various stacks in AWS using all the aforementioned technologies. Our Jenkins server is configured using [Configuration as Code](#) and uses build nodes both on EC2 instances and kube pods.

- [Github Actions](#)

We use github actions for building and publishing some of our open sources projects. I also used them to build and publish [this cv](#).

- [Chef](#)

This is used to install and configure [Finvi](#) software on EC2 instances in AWS. I have written and maintain over 30 cookbooks as well as utilizing and contributing to open source cookbooks from the [Chef Supermarket](#).

- [Terraform](#)

I use Terraform to automate the creation of infrastructure in AWS. Our terraform repos are configured to deploy into different AWS accounts based on configuration variables. Some common functionality is modularized to be reused by multiple repos, and we also make use of some open source community modules from the [Terraform Registry](#).

- [Docker](#)

Several [Finvi](#) products are based on microservices, which are built and delivered as Docker images. Most are either built on base images from [Docker Hub](#) (running Alpine, Debian, or Ubuntu Linux operating systems) or from scratch images that run binaries we build.

- [NGINX](#) and [Apache httpd](#)

Various [Finvi](#) products use Apache and/or NGINX to server their web applications or proxy to some backend webserver.

Other skills

There are a number of other skills that are important for software development but don't fit into the categories of programming languages or technologies.

- I use a mixture of Windows and Linux for my development environment. I'm currently running Ubuntu on WSL2. I also know Amazon Linux, CentOS, Red Hat, Oracle Linux, Debian, and a little Alpine Linux for our EC2 and Docker container-based products.
- I have used a source control systems and am most familiar with [Git](#) and [Mercurial](#).
- I encourage teams to use [Semantic Versioning](#) to help with dependency management software maintainability.
- I have worked with a number of branching strategies, in particular [Gitflow](#) and [feature branch](#) workflows. Open source projects I've contributed to typically use some form of [GitHub flow](#) or trunk-based workflows.
- I have contributed to a number of open source projects and am skilled at reading other people's code.
- I have used [Jira](#) and [FogBugz](#) for task management.
- I have worked on teams that use Scrum and Kanban-based processes. I've also read the [Manifesto for Agile Software Development](#).
- I have used [Test-driven Development](#) on a number of projects.
- My text editor of choice is [Vim](#), but I've also used JetBrains IntelliJ and RubyMine, Visual Studio, VSCode, Notepad++, and others. I used to use Emacs in grad school and don't hold anything against people who prefer it over vi.

Education

Master of Science in Computer Science. GPA 3.97/4.00
[Ball State University](#) - Muncie, IN (2004 - 2006)

Bachelor of Science in Computer Science Systems. GPA 2.75/4.00
[Taylor University](#) - Upland, IN (1998 - 2002)