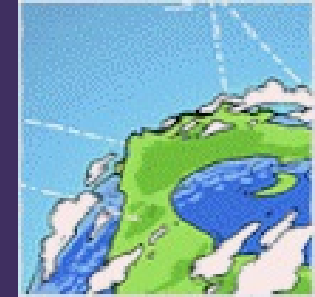


Scott Lowe
Author, Blogger, Geek
<https://blog.scottlowe.org>
Twitter: @scott_lowe



Infrastructure as Code with Pulumi

Defining your infrastructure using programming languages

A quick bit about me

- Husband, father, Jeoper, all-around geek
- Blogger (19 years at <https://blog.scottlowe.org>)
- Author (8 books so far)
- Former podcaster (The Full Stack Journey podcast)
- Have worked at Nicira/VMware, Heptio/VMware, Kong, and Pulumi

Why infrastructure as code (IaC)?

- Enables easily repeatable, consistent deployments of infrastructure
- Used with version control, makes it possible to easily see **who** changed **what** in your infrastructure
- A form of documentation (doesn't replace good documentation, though!)
- Simplifies team collaboration (through code reviews, shared repositories, etc.)

Why programming languages for IaC?

- Programming language knowledge is a reusable and broadly applicable skill
- There is a powerful ecosystem of supporting tools (IDE support, syntax highlighting, code completion, linting, code analysis)
- Native, intuitive support for logic/branching and looping

Why Pulumi for IaC?

- Broad ecosystem of supported providers and platforms
- Broad range of supported languages (JavaScript, TypeScript, C#/F#/VB.NET, Go, Python, even YAML)
- No intermediate transpilation into another format; communicates with provider APIs directly
- Tools available to convert from other formats
- Pulumi AI can help with code generation

Some Pulumi terminology

- **Project:** A directory containing a `Pulumi.yaml` file
- **Program:** Code using Pulumi's SDKs to define infrastructure
- **Stack:** An independent instance (with its own state and its own configuration) of the infrastructure defined in the program
- **Resource:** The “things” that make up your cloud infrastructure
- **Input** or **Output:** An Input is a parameter to a resource; an Output is a property of a resource (asynchronous)

Demo time!

Pulumi Commands

- `pulumi login`: Logs you into a state backend
- `pulumi preview`: Runs the program and generates a preview of the changes
- `pulumi up`: Runs the program and declaratively makes the changes
- `pulumi refresh`: Updates Pulumi's state from the infrastructure
- `pulumi destroy`: Exactly what you think it does!

Pulumi Backends

- **Pulumi Cloud:** Pulumi's SaaS backend that offers state and secrets storage, “free forever” individual tier, paid tiers (free trials available) with features like Deployments (hosted CI/CD), RBAC, etc.
- **Cloud object storage:** Services like Amazon S3, Azure Blob Storage, or Google Cloud Storage
- **Local:** Fine for PoC but not recommended for production use

Pulumi Cloud

×

+

Private browsing

←

→

↻

🏠

🛡️

🔒

https://app.pulumi.com/signin?reason=401

☆

🌐

📄

⏏

☰

Pulumi

Star

18,704

Slack

Docs

Registry

Pulumi AI

Sign in

Sign In

New to Pulumi? [Create an account](#)

Email or Username

Password

Sign in

[Forgot Password?](#)


OR

GitHub

GitLab

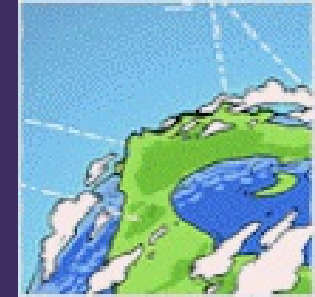
Atlassian

[Sign in via SSO](#)



[Back to the demo](#)

Scott Lowe
Author, Blogger, Geek
<https://blog.scottlowe.org>
Twitter: @scott_lowe



Thank you!

Sample code is available at <https://github.com/scottlowe/2024-90dod-iac-pulumi>