

# Functional DevOps

Geoff Purdy

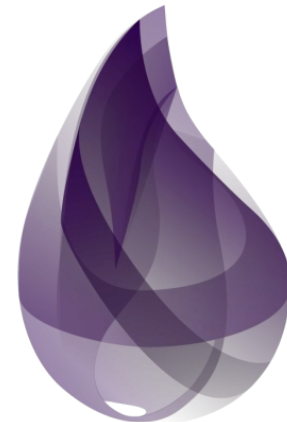
Triangle Clojure Users Group Co-Organizer

DevOps Engineer

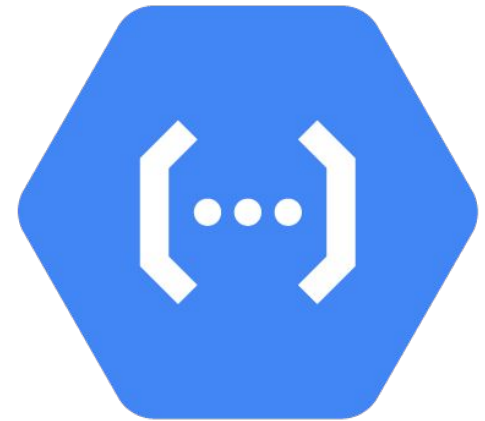
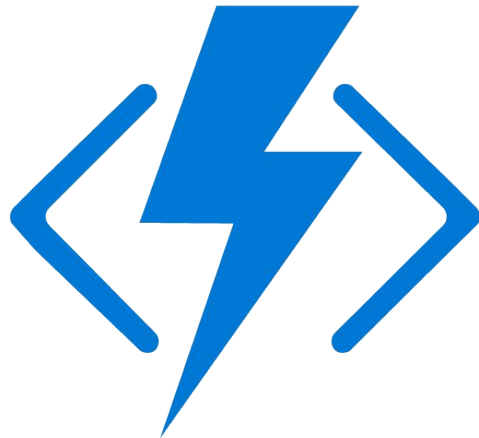
 @geoff\_purdy

# Functional programming

"Functional programming (FP) is beginning to hit the mainstage" --*The New Dawn Of Functional Programming*



# Not about FaaS



# Not non-dysfunctional DevOps



# DevOps Toolchain



# DevOps Toolchain



This article appears to **contain a large number of buzzwords.**

# Computations as mathematical functions

- First Class Functions
- Favor recursion over control structures
- No side effects

# Immutable state

- Don't modify existing
- Create new
- Concurrency
- Testability



# Declarative syntax

- What
- Not how



# ~~Programming Language~~ ~~Theory~~

*Von Neumann Languages*



*Lambda Calculus*



# Riemann

- Riemann monitors distributed systems
- Aggregate events
- Stream processing



# Riemann

- Riemann written in Clojure
- Configuration is a Clojure program

```
(where (or (service #"^api")
           (service #"^app"))
      (where (tagged "exception")
              (rollup 5 3600
                    (email "dev@foo.com")))
      (else
        (changed-state
          (email "ops@foo.com")))))
```

# NixOS

- The Purely Functional Linux Distribution
- Based on Nix - The Purely Functional Package Manager



# NixOps

- The NixOS Cloud Deployment Tool
- Tool for deploying sets of NixOS Linux machines

```
{  
  webserver =  
    { deployment.targetEnv = "virtualbox";  
      services.httpd.enable = true;  
      services.httpd.documentRoot = "/data";  
      fileSystems."/data" =  
        { fsType = "nfs4";  
          device = "fileserver:/"; };  
    };  
  
  fileserver =  
    { deployment.targetEnv = "virtualbox";  
      services.nfs.server.enable = true;  
      services.nfs.server.exports = "...";  
    };  
}
```

# DSL Template Generation

- Infrastructure as Code
- Multi-cloud
- Output JSON, YAML, etc



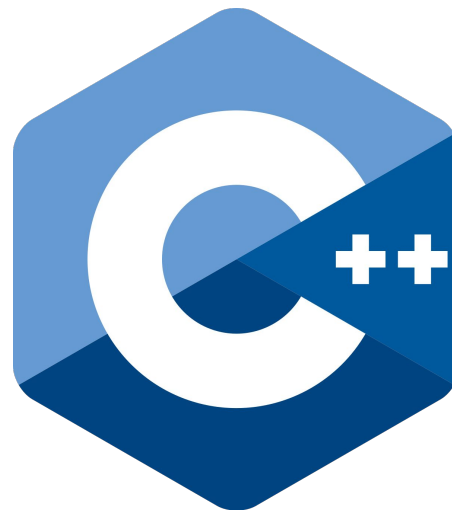
# Fugue/Ludwig

- Domain-Specific Language designed for coding cloud infrastructure
- Functional language
- Data
- Functions

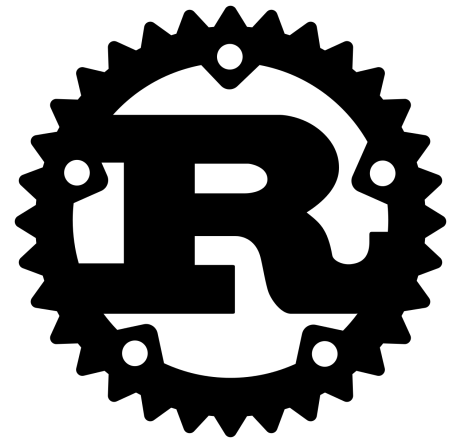
```
exampleNetwork: Network.new {  
  name: "Example VPC",  
  region: exampleRegion,  
  cidr: "10.0.0.0/16",  
  publicSubnets: [  
    (examplePrimaryAz, "10.0.1.0/24"),  
    (exampleSecondaryAz, "10.0.2.0/24")  
  ],  
  privateSubnets: []  
}
```



# Imperative: Functional Style



# Multi-paradigm



# Learning Resources

- <http://riemann.io/clojure.html>
- <https://nixos.org/nixops/>
- <https://pragprog.com/book/fugue/scalable-cloud-ops-with-fugue>
- MIT Structure and Interpretation of Computer Programs

# References

- <https://clojure.org/about/rationale>
- [https://clojure.org/about/functional\\_programming](https://clojure.org/about/functional_programming)
- [https://en.wikipedia.org/wiki/DevOps\\_toolchain](https://en.wikipedia.org/wiki/DevOps_toolchain)
- <https://www.forrester.com/report/The+New+Dawn+Of+Functional+Programming/-/E-RES135953>
- [https://www.thocp.net/biographies/papers/backus\\_turingaward\\_lecture.pdf](https://www.thocp.net/biographies/papers/backus_turingaward_lecture.pdf)