Patterns In Infrastructure as Code

Takeshi Kondo / @chaspy

Infra Study Meetup #1

Passed 🎉

HashiCorp Certified: Terraform Associate

The Terraform Associate certification is for Cloud Engineers specializing in operations, IT, or development who know the basic concepts and skills associated with open source HashiCorp Terraform. Candidates will be best prepared for this exam if they have professional experience using Terraform in production, but performing the exam objectives in a personal demo environment may also be sufficient. This person understands which enterprise features exist and what can and cannot be done using the open source offering. Visit our exam partner to schedule and take the exam.

HashiCorp

HashiCorp Product

Certification Exam Name	HashiCorp Certified Terraform Associate
Exam Outcome	Pass
Percentage score	70%
Total score	40
Maximum Possible Score	57

Circuit Breaking
@chaspy_

はてなブログに投稿しました #はてなブログ （たぶん）日本人最速取得です HashiCorp Certified: Terraform Associate を取った - ツナワタリマイライフ
blog.chaspy.me/entry/2020/04/...

Agenda

- Infrastructure as Code in Quipper
- Patterns in Infrastructure as Code
  1. Environment Pattern
  2. Scaffold Pattern
  3. Backup Pattern

Agenda

- Infrastructure as Code in Quipper
- Patterns in Infrastructure as Code
  1. Environment Pattern
  2. Scaffold Pattern
  3. Backup Pattern

- What
- Why
- Point
- Example

Agenda

- Infrastructure as Code in Quipper
- Patterns in Infrastructure as Code
  1. Environment Pattern
  2. Scaffold Pattern
  3. Backup Pattern

Infrastructure as Code in Quipper (1)

- Infrastructure / Cloud (AWS)
  - Terraform
  - Codenize.tools (Miam, Roadworker)
- Application Platform (Kubernetes)
  - Kube-aws
- Server Provisioning
  - Ansible

Infrastructure as Code in Quipper (2)

- SaaS Configuration
  - Datadog
    - Dashboard
    - SLO
    - Monitor
  - Pingdom
  - Deadman’s snitch
  - CircleCI
    - Environment Variables

Infrastructure as Code in Quipper(2)

- SaaS Configuration
  - Datadog
    - Dashboard
    - SLO
    - Monitor
  - Pingdom
  - Deadman's snitch
  - CircleCI
    - Environment Variables

Terraform

Codenize.tools (Barkdog)

circleci-env (Internal tool)

Infrastructure as Code in Quipper(2)

- SaaS Configuration
  - Datadog
    - Dashboard
    - SLO
    - Monitor
  - Pingdom
  - Deadman's snitch
  - CircleCI
    - Environment Variables

Terraform

Codenize.tools (Barkdog)

circleci-env (Internal tool)

Infrastructure as Code in Quipper(2)

- SaaS Configuration
  - Datadog
    - Dashboard
    - SLO
    - Monitor
  - Pingdom
  - Deadman's snitch
  - CircleCI
    - Environment Variables

Terraform

Codenize.tools (Barkdog)

circleci-env (Internal tool)

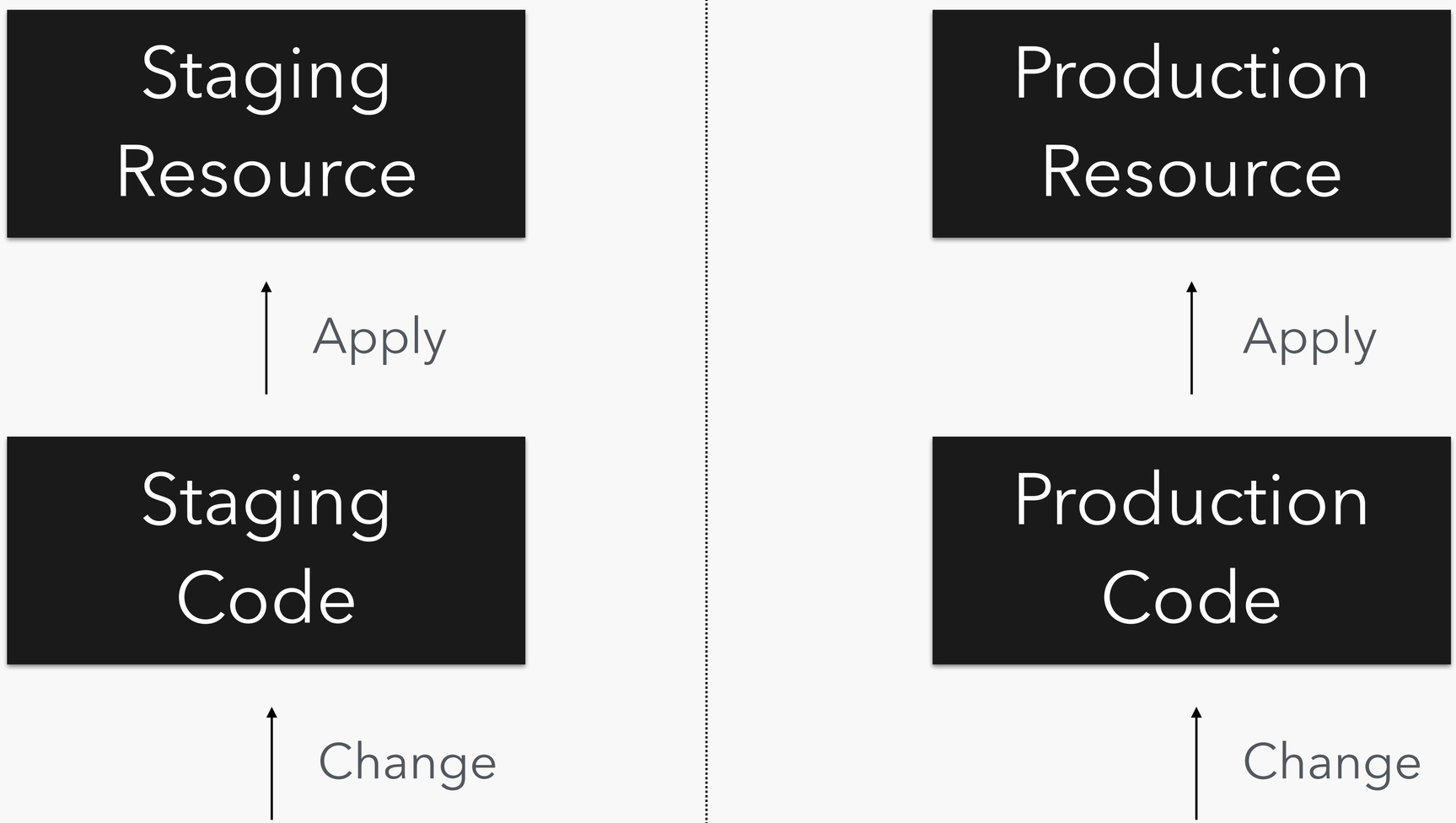
Agenda

- Infrastructure as Code in Quipper
- Patterns in Infrastructure as Code
  1. Environment Pattern
  2. Scaffold Pattern
  3. Backup Pattern

- What
- Why
- Point
- Example

Environment Pattern

Environment Pattern



Environment Pattern: Why

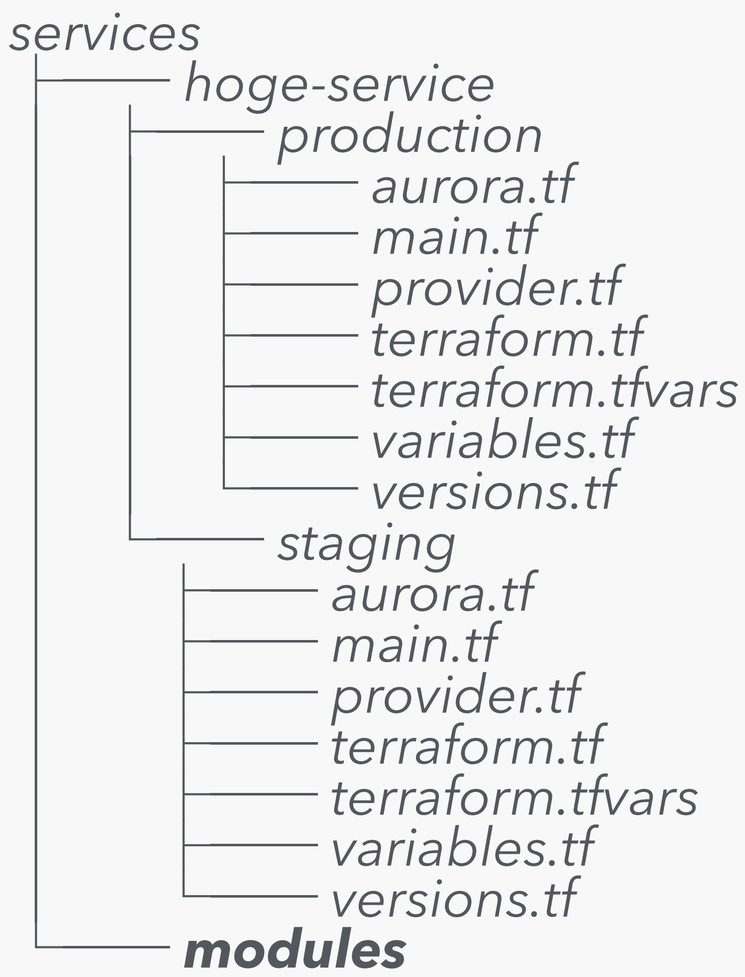
- To minimize blast radius
- To test before releasing to production

Environment Pattern: Point

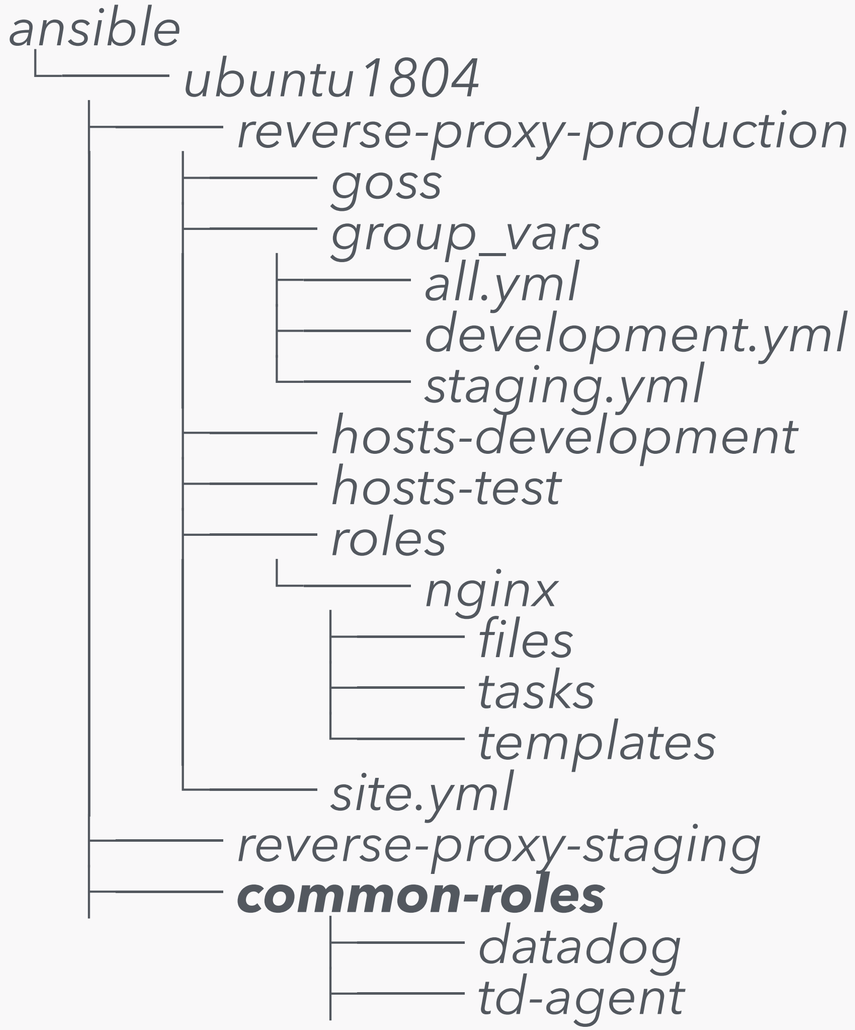
- ~~Don't~~ **Repeat Yourself**
  - Minimize commonality
  - Keep it simple stupid
- Enforcing, not communize

Environment Pattern: Example

Terraform

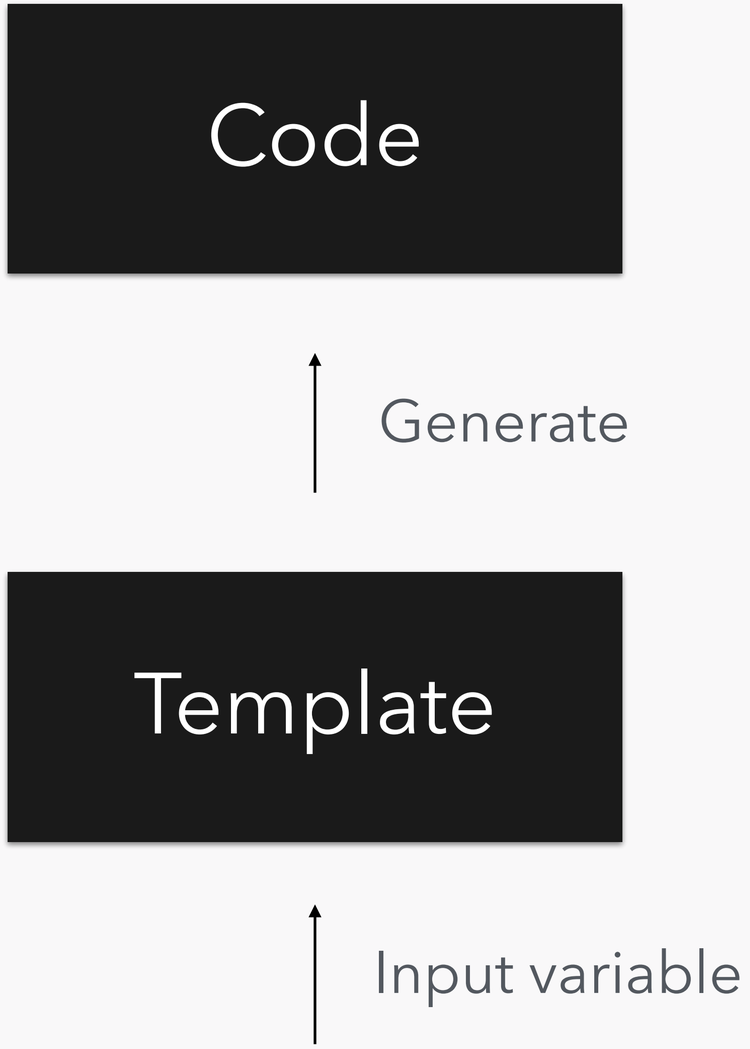


Ansible



Scaffold Pattern

Scaffold Pattern



Scaffold Pattern: Why

- To make self-service easier
  - Reduce cognitive load

Scaffold Pattern: Point

- Write a document
- Make each program do one thing well

https://en.wikipedia.org/wiki/Unix_philosophy

Scaffold Pattern: Example

- Create new application with Kubernetes manifest
- Create new resource with Terraform HCL
- Create AWS Aurora resource definition
- Create Datadog SLO Definition

Create AWS Aurora resource definition

Steps with Terraform

If you are using Terraform for the first time in your service, don't forget "Scaffold a new service".

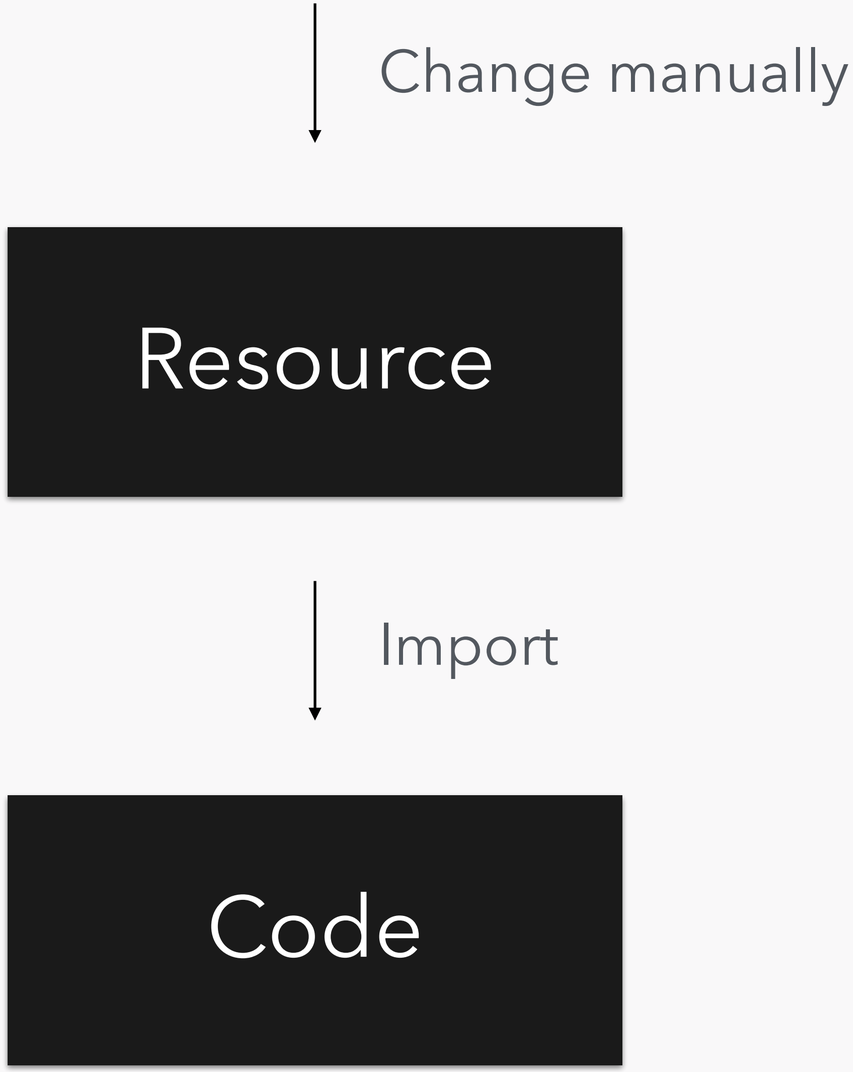
ref:

- https://github.com/quipper/[redacted]#scaffold-a-new-service
- https://github.com/quipper[redacted]#scaffold-a-new-service

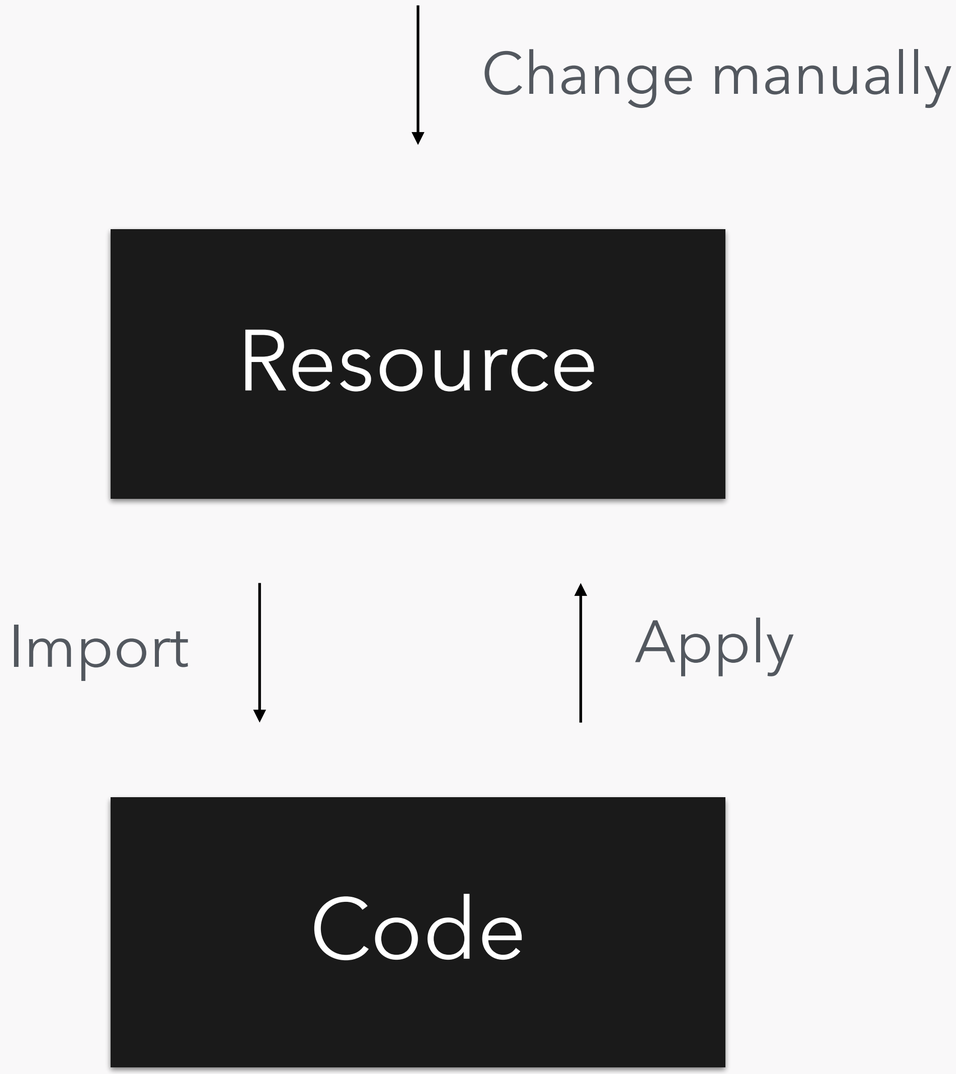
1. Go to the Repository and checkout your feature branch
2. Exec `./script/generate_tf_for_rds.sh -e <staging|production> -d <mysql|postgres> <service-name>`
    - Note for
        - Only allows **lowercase alphanumeric characters** and **hyphens**
        - Must be specified with the same name as the monorepo directory
3. Add Circle CI job to check the result of terraform plan
4. Commit and Push and create PullRequest to master branch
5. Check the result of terraform plan
6. Ask @your-team to review

Backup Pattern

Backup Pattern



Backup Pattern



Backup Pattern: Why

- Easy to change from GUI (especially SaaS Configuration)
- For backup
- For greppability
- For bulk replacement

Backup Pattern: Point

- Sync Interval
  - If long, it may overwrite the actual configuration

Backup Pattern: Example

- Datadog Dashboard
- Datadog Monitor
- Jenkins Configuration (SCM Sync Plugin)

Datadog Dashboard

Sync from current actual dashboards #56

Open **quippo** wants to merge 9 commits into `master` from `sync-data`

Conversation 9 | Commits 9 | Checks 0 | Files changed 2

quippo commented 4 days ago

@quipper/sre

Merge if CI passed.

Script:
https://github.com/quipper/[redacted]/blob/master/import.sh

Sync from current actual dashboards by import.sh ✕ a51d180

quippo commented 4 days ago Author

From CircleCI

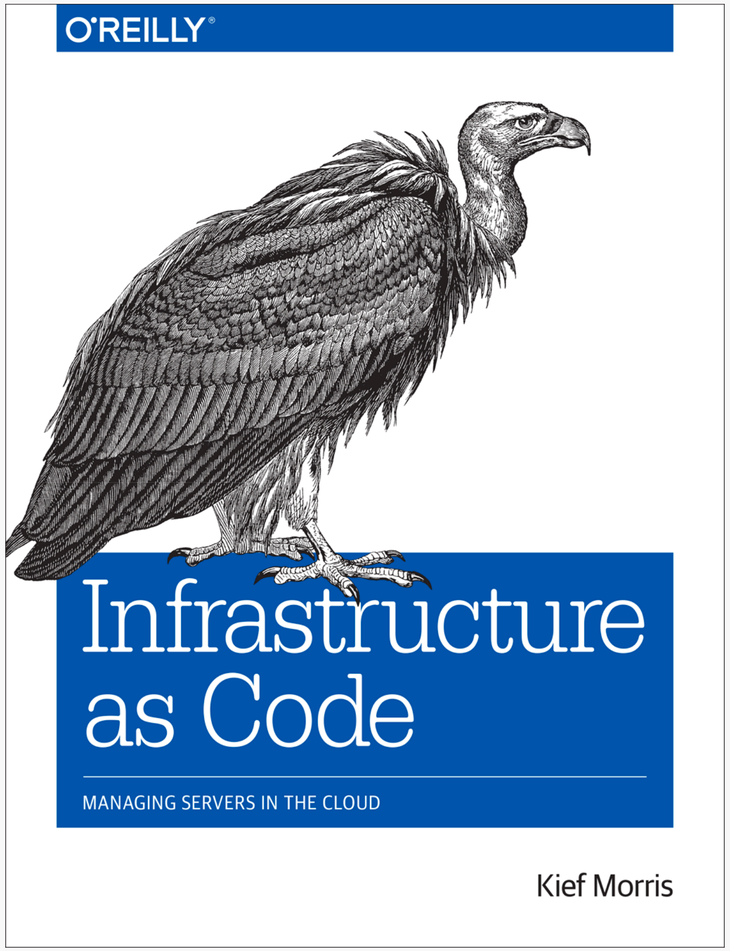
workflow job

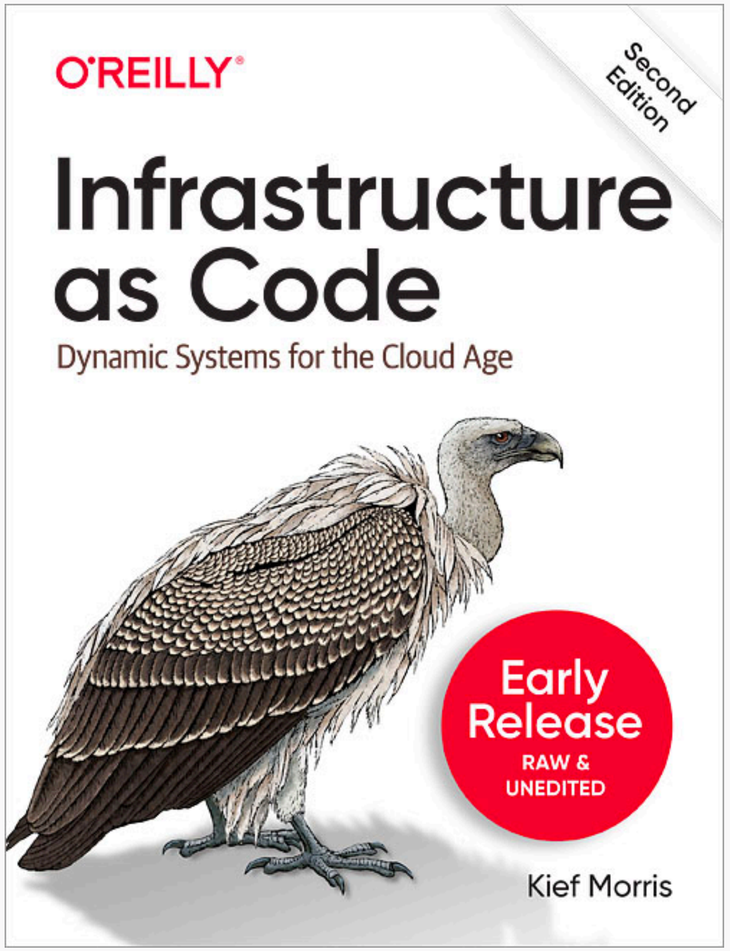
✅ Command exited with code 0

```
terraform plan
```

▶ Command output

Let's Read 😋





http://shop.oreilly.com/product/0636920294382.do

Special Thanks 🙏

- @suzuki-shunsuke 
  - To lead splitting Terraform state
  - To improve CI/CD pipeline
- SRE graduates
  - To introduce of Infrastructure as Code
  - To build workflow with maintainability

Happy (Infrastructure as) Coding 😋

Thank You!

Takeshi Kondo

Site Reliability Engineer
at Quipper

Terraform-jp

chaspy

chaspy_