



Get Started with GitOps

Operations by Pull Request enable Terraform for Teams

Cloud Posse

[<hello@cloudposse.com>](mailto:hello@cloudposse.com)

<https://cloudposse.com/>

@cloudposse



WHAT AM I GETTING MYSELF INTO

What to Expect

Feelings of Euphoria

Aha! Moments

Reduced Anxiety

AND...

What is GitOps? (not rocket science)

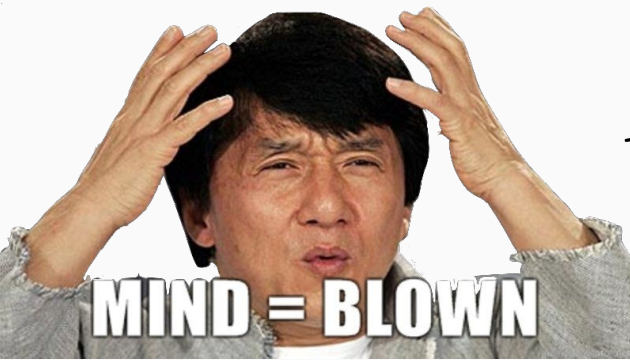
Why it's awesome (and you'll agree)

How to get started... (our way)

LIVE DEMO

Q&A

+ HashiConf
News!



Who is this dude?



Founder of **Cloud Posse** a DevOps Professional Services Company

We've pioneered **SWEETOPS**



Collaborative DevOps for Companies

(100% OPEN SOURCE)

ME
(ERIK OSTERMAN)



(cloudposse.com)

We got problems.

We Maintain **100+ Terraform Modules** (the largest!)

Dozens of Helm Charts

Pain in the *ss to **test everything**

Multi-stage **rollouts get complicated**

Lots of tools like Helm, Kops, Terraform and Cloud Formation

Thousands of users (*hey, some problems are good to have!*)

**WE'VE
GOT
ISSUES**



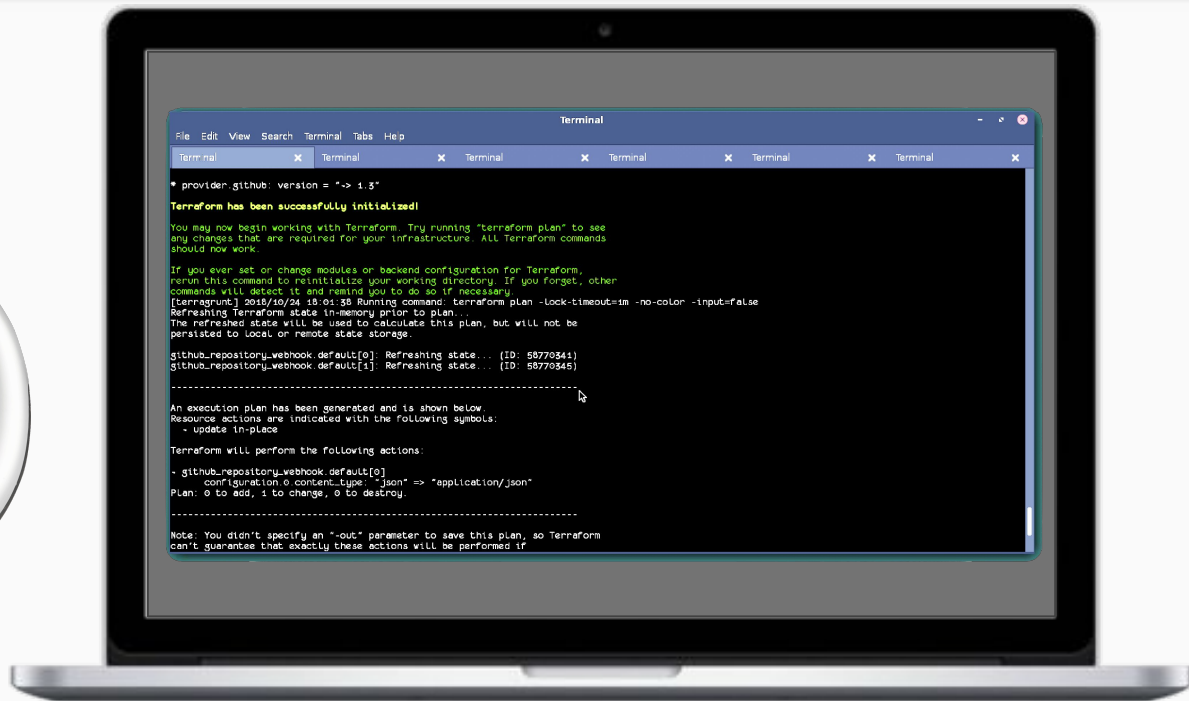
Goal:

Effortlessly Deploy Infrastructure

(e.g. spin up RDS Database with Terraform, or deploy Helm Chart)

One Approach...

Make changes in the
privacy of your personal laptop.
(sometimes after a few beers)



SWEAR

“I ^ it worked on my
machine.”

Then comes... **LAUNCH DAY**

```
aws_instance.salt_master_a (remote-exec): consul start/running, process 3431
aws_instance.salt_master_a: Creation complete
aws_eip.salt_master_a: Creating...
  allocation_id: "" => "<computed>"
  association_id: "" => "<computed>"
  domain: "" => "<computed>"
  instance: "" => "i-2a1a9afc"
  private_ip: "" => "<computed>"
  public_ip: "" => "<computed>"
  vpc: "" => "v1"
aws_eip.salt_master_a: Error: 1 error(s) occurred:

+ Failure associating EIP: InvalidAllocationID.NotFound: The allocation ID 'eipalloc-9b0b7cfe' does not exist
aws_route53_record.dns_b: Creation complete
Error applying plan:

1 error(s) occurred:

+ 1 error(s) occurred:

+ 1 error(s) occurred:

+ Failure associating EIP: InvalidAllocationID.NotFound: The allocation ID 'eipalloc-9b0b7cfe' does not exist

Terraform does not automatically rollback in the face of errors.
Instead, your Terraform state file has been partially updated with
any resources that successfully completed. Please address the error
above and apply again to incrementally change your infrastructure.
```

PRODUCTION



Other Problems...

No Audit Trails (huge risk)

Complicated **Manual Rollouts**

Not clear what's been deployed (configuration drift)

Failed Deployments on Merge ***(NOW WHAT?!)***

Insufficient **Code Reviews**

No one knows how to make changes

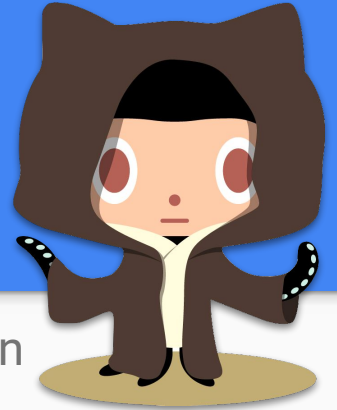


So....

Let's fix

this.

Let's Practice GitOps.



Use Git as a System of Record for the desired state of configuration

Do **Operations by Pull Request** for Infrastructure as Code

Then use **Continuous Delivery** to apply changes to infrastructure

(BASICALLY IT'S A CI/CD FOR DEVOPS)

Issue **commands using comments** to trigger actions (a.k.a "ChatOps")

(E.g. "@bot give me a plan", "@bot deploy these changes")



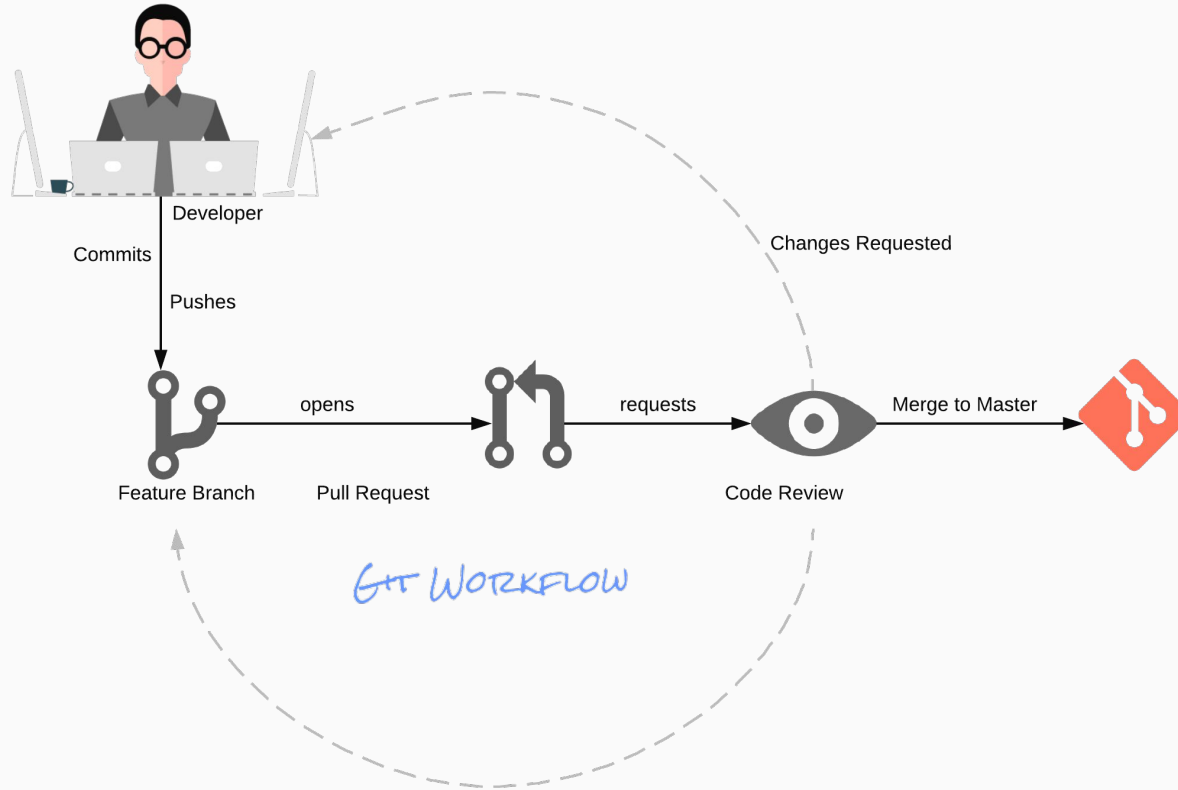
Run **PLAN**

See what should change

Run **APPLY**

See what actually happened

The "Git Workflow"



Why do you care?
Teamwork.

GitOps Objectives

Repeatable - Apply changes the same way every time
(even your entire stack all at once!)

Predictable - Know what's going to happen
(e.g. before you merge)

Auditable - See what was done
(e.g. when things were applied. see if there were errors)

Accessible - Anyone who can open a PR can contribute



The Solution



Atlantis

<https://runatlantis.io>



Now an official HashiCorp sponsored project

HashiCorp

Built for Terraform

(but **will run anything**)

About Atlantis

Purpose-built for Terraform (understands `init`, `plan`, `apply`)

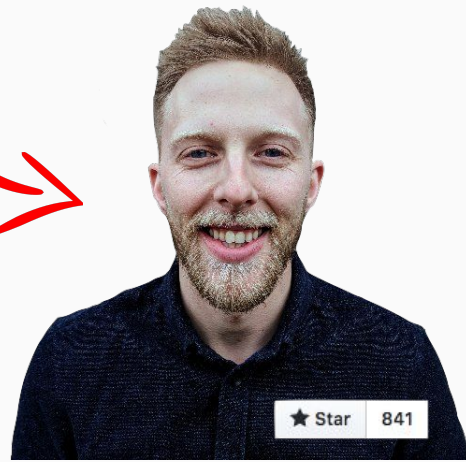
Project started at  **hootsuite**

**CURRENT MAINTAINER IS
LUKE KYSON**

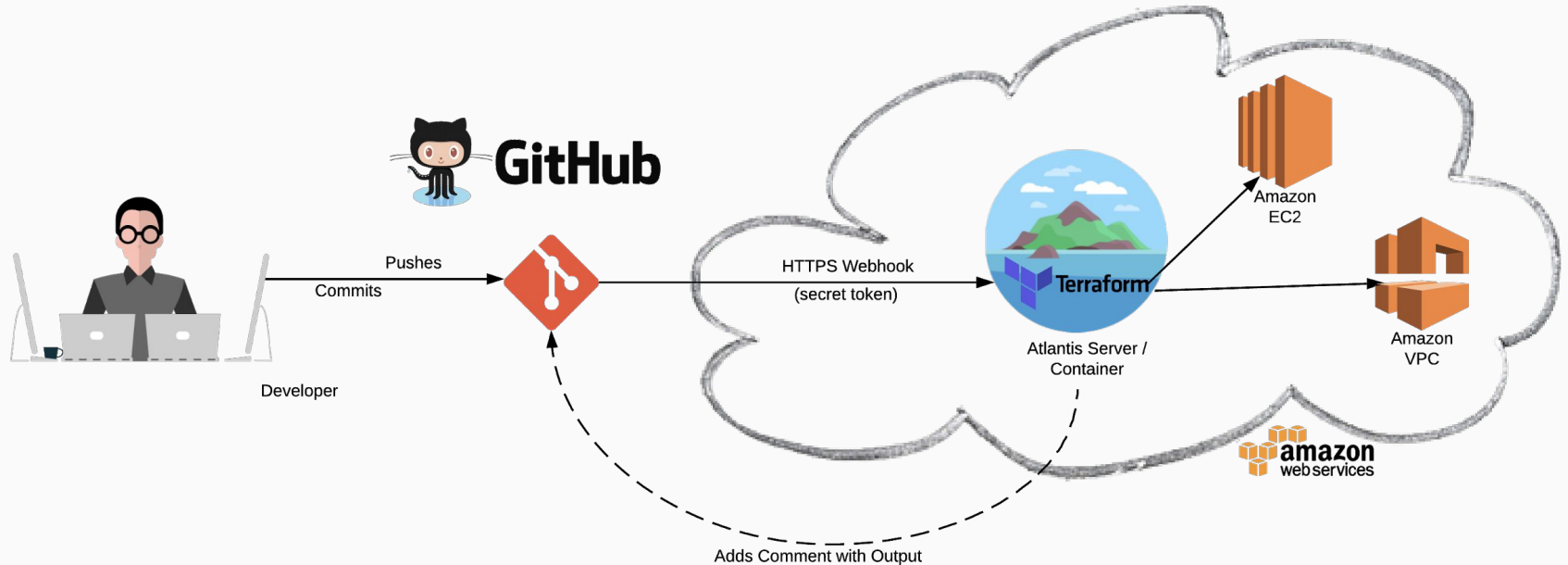
Officially forked into
<https://github.com/runatlantis/atlantis>

Open Source APACHE2

100% Golang with good test coverage



Basic Flow Diagram





BECAUSE WE CAN
RUN ANY COMMAND

TERRAFORM

CLOUD FORMATION

HELM

HELMFILE



But will it work with...

TERRAGRUNT? YES

GITLAB? YES

BITBUCKET? YES

DOCKER? YES

BUT WAIT!
THERE'S MORE!



“Interactive”

Pull Requests

ready?
set.
go!

Step One: Open Pull Request

Add user aknysh #25

Open osterman wants to merge 1 commit into master from add-aknysh

Conversation 1 Commits 1 Checks 0

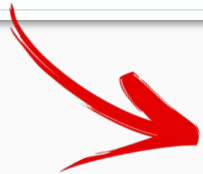
osterman commented 7 minutes ago

what

- Add user aknysh for @aknysh

why

- AWS Admin access



Add user aknysh #25

Open osterman wants to merge 2 commits into master from add-aknysh

Conversation 4 Commits 2 Checks 0 Files changed 1

Changes from all commits Jump to... +11 -0

Diff settings Review changes

```
11 conf/users/aknysh.tf
... @@ -0,0 +1,11 @@
1 + module "aknysh" {
2 +   source = "git::https://github.com/cloudposse/terraform-aws-iam-user.git?ref=tags/0.1.0"
3 +   name   = "aknysh"
4 +   pgp_key = "keybase:aknysh"
5 +   groups = "${local.admin_groups}"
6 +   force_destroy = "true"
7 + }
8 +
9 + output "aknysh_decrypt_command" {
10 +   value = "${module.osterman.keybase_password_decrypt_command}"
11 + }
```



Step Two: Review "Auto Plan"

What you get



cloudpossebot commented 8 minutes ago

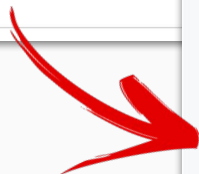
Member + 😊 ...

Ran Plan in dir: `conf/users` workspace: `default`

► Show Output

- ▶ To apply all unapplied plans from this pull request, comment with:
 - `atlantis/root apply -p users`

**FAILING
TO PLAN IS
PLANNING
TO FAIL**



```
+ module.aknysh.aws_iam_user.default
  id: <computed>
  arn: <computed>
  force_destroy: "true"
  name: "aknysh"
  path: "/"
  unique_id: <computed>

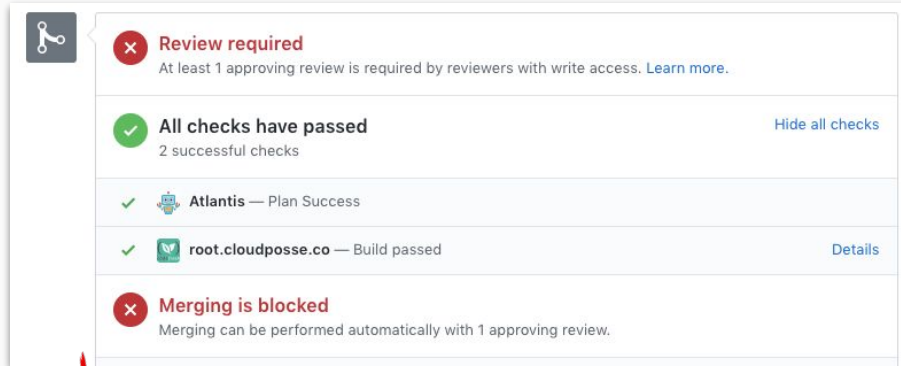
+ module.aknysh.aws_iam_user_group_membership.default
  id: <computed>
  groups.#: "1"
  groups.1306695553: "cpc0-root-admin"
  user: "aknysh"


+ module.aknysh.aws_iam_user_login_profile.default
  id: <computed>
  encrypted_password: <computed>
  key_fingerprint: <computed>
  password_length: "24"
  password_reset_required: "true"
  pgp_key: "keybase:osterman"
  user: "aknysh"


Plan: 3 to add, 0 to change, 0 to destroy.
```







Step Three: Seek Approval




 **Review required**
At least 1 approving review is required by reviewers with write access. [Learn more.](#)

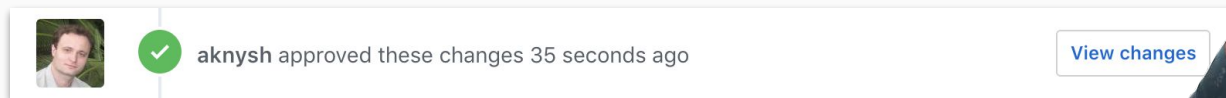
 **All checks have passed** [Hide all checks](#)
2 successful checks



  Atlantis — Plan Success

  root.cloudposse.co — Build passed [Details](#)

 **Merging is blocked**
Merging can be performed automatically with 1 approving review.

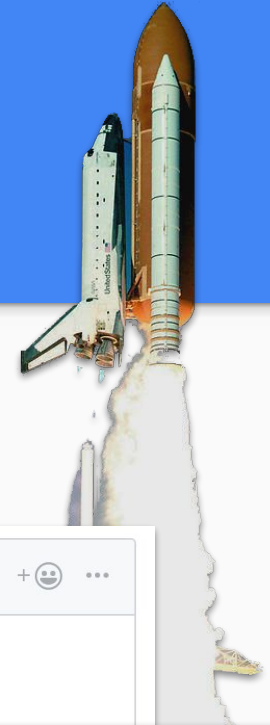
CODE REVIEW



  aknysh approved these changes 35 seconds ago [View changes](#)



Step Four: Deploy Changes



osterman commented 5 minutes ago

Member



atlantis/root apply



cloudpossebot commented 2 minutes ago

Member



Ran Apply in dir: `conf/users` workspace: `default`

► Show Output

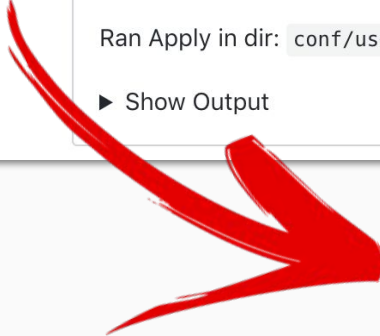
Apply complete! Resources: 3 added, 0 changed, 0 destroyed.
Releasing state lock. This may take a few moments...

Outputs:

account_alias = cpco-root-account

aknysh_decrypt_command = echo "wcFMAYl0j/2+yeBZARAAWYED+UcMl+QBBnY+l3kvnnGEHS7n0Y0fBUyZie7

osterman_decrypt_command = echo "wcFMAYl0j/2+yeBZARAAWYED+UcMl+QBBnY+l3kvnnGEHS7n0Y0fBUyZi



Step Five: Merge Pull Request



Pull request successfully merged and closed

You're all set—the `add-aknysh` branch can be safely deleted.

Delete branch

Nailed It!

That was

easy.



Atlantis Users?



hootsuite™

lyft



Blinkist

pagerduty



shopify



CBS SPORTS

(...and soon most of our customers)



Cognite



What others are saying...



nat

@nuttaay



We prevent #terraform changes to our infrastructure that haven't been approved by a reviewer with the "require-approval:true" flag on the #atlantis config, this makes our PR process 👍🔒 @runatlantis



natalysheinin commented just now

```
atlantis apply -d terraform/clusters/shopify-gke-bugbounty
```



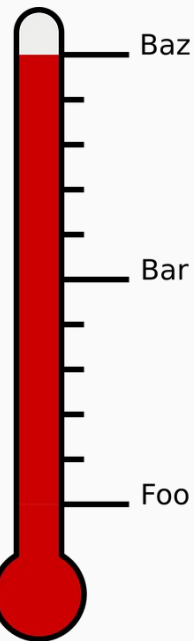
sa-atlantis commented just now

Apply Failed: Pull request must be approved before running apply.

2:22 AM - 20 Jul 2018



Kelsey Hightower says... *Extra Dope*



Kelsey Hightower ✓

@kelseyhightower

The live demo of Atlantis was extra dope.
[@hootsuite](#) should be proud to have sent one
of their best.

[HTTPS://RUNATLANTIS.IO](https://runatlantis.io)



How to get started

1. Deploy Atlantis (e.g. ECS, Kubernetes+Helm)
2. Add `atlantis.yaml` to each repo
3. Get back to work (sorry).



OR JUST ASK US FOR HELP =>



Deploy Atlantis on ECS Fargate



```
fargate certificate create
fargate certificate validate
fargate lb create
fargate lb alias
fargate service create
```

1. CREATE TLS CERTIFICATE
2. ACTIVATE IT
3. CREATE LOAD BALANCER
4. ASSIGN DNS
5. DEPLOY CONTAINER

<https://github.com/cloudposse/geodesic-aws-atlantis>

Example `atlantis.yaml`

```
version: 2
projects:
- name: "alpinist"
  dir: "terraform"
  workspace: "default"
  terraform_version: "v0.11.7"
  autoplan:
    when_modified:
      - "*.tf"
    enabled: true
  apply_requirements:
    - "approved"
  workflow: "default"
```

```
# define list of chart repositories
# list of projects in this repo
# friendly name for this project
# directory with the tf code
# workspace to use with this project
# terraform version to use
# automatically run terraform plan
#   when...
#     any .tf file changes
#     and enabled
# then run terraform apply
#   only when approved
#     run this workflow
```

STEPS CAN BE ENTIRELY CUSTOMIZED.

Example `atlantis.yaml` (Continued)

```
workflows:
  default:
    plan:
      steps:
        - run: "init-terraform"
        - run: |-
            terraform plan -no-color \
              -var-file atlantis.tfvars -out $PLANFILE
    apply:
      steps:
        - run: |-
            terraform apply -no-color \
              -var-file atlantis.tfvars $PLANFILE
# define list of workflows
# friendly name for this workflow
# to do a plan
#   perform some steps
#     run a command to initialize tf state
# use fancy YAML conventions
# run a terraform plan use with -var-file
# save the plan to a file for later
# to do a plan...
# run these steps..
#   with some fancy YAML
#     run apply using previous plan
#       $PLANFILE ensures WYSIWYG
```

STEPS CAN BE ENTIRELY CUSTOMIZED.

Live Demo

- 1. ADD USER**
- 2. OPEN PR**
- 3. RUN PLAN**
- 4. SEEK APPROVAL (OR NOT)**
- 5. APPLY**
- 6. MERGE**



DEMO GODS



PLEASE LET THIS DEMO WORK

Fabulous

Our Best Practices

Use **one Atlantis Server per account** (prod, dev, staging, identity, security, etc)

Use **IAM Service Account** for credentials (not hardcoded credentials)

Use GitHub **CODEOWNERS**

Use **-var-files** for non-secrets

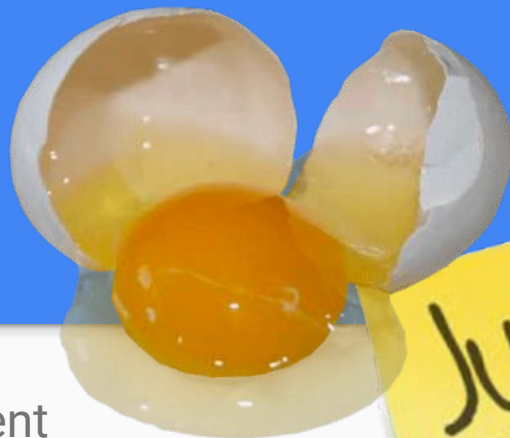
Use **chamber** by segmentio for secrets (SSM+KMS)

Disable for forks



Gotchas

THE
GOOD
THE
BAD
& THE
UGLY



Atlantis is under active development

We've forked it to support *what we needed*

1. Restricted Users

2. Git Submodules

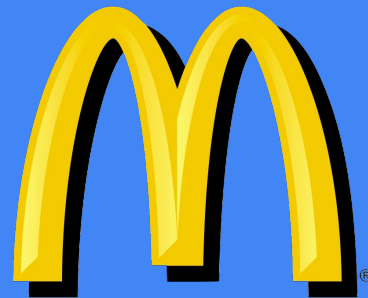
3. Multiple Pipelines (e.g. `atlantis/prod.yaml`, `atlantis/staging.yaml`)

4. Destroy action

5. Custom wake words (e.g. "echo, shut up")

 <https://github.com/cloudposse/atlantis>

GitOps



i'm lovin' it™

Stop living dangerously.

Start using GitOps.

- Practice total transparency in operations
- Increase **Productivity**, Simplify **Maintenance**, Ensure **Repeatability**
- Reduce the barrier to entry
- Scalable strategy to manage lots of infrastructure

[HTTPS://GITHUB.COM/RZUNATLANTIS/ATLANTIS](https://github.com/rzunatlantis/atlantis)

HashiConf 2018 Announcements



HashiCorp Atlantis



HashiCorp

Terraform

0.12 (alpha 1) released

+ "Terraform State as a Service"



HashiCorp

Vault

Automatic Unsealing -> Open Source



New provider! Manage charts with terraform



Links



SWEETOPS

Our Fork

<https://github.com/cloudposse/atlantis>

Our Slack Community

JOIN OUR COMMUNITY!

<https://slack.cloudposse.com/>

Our Demo

<https://github.com/cloudposse/root.cloudposse.co>

Totally Bodacious



Geodesic (container+env pattern for Infrastructure as Code)

github.com/cloudposse/geodesic

Packages (our complete toolchain + alpine packages)

github.com/cloudposse/packages

Build Harness (Makefiles on Steroids; build anything)

github.com/cloudposse/build-harness

Reference Architectures

github.com/cloudposse?q=cloudposse.co

Documentation

docs.cloudposse.com



Cloud Posse

A Totally Sweet DevOps Professional Services Company

Hire us. =)

100+ Free Terraform Modules

Active Community

Awesome Documentation

github.com/cloudposse/

slack.cloudposse.com

docs.cloudposse.com



(free consultation)