

A hands-off approach for your Terraform

Jeff Knurek



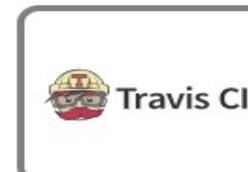
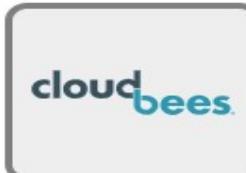
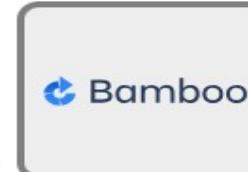
Continuous Delivery starts with
Continuous Infrastructure

Kris Buytaert

UB4.136
2020-02-02

<https://www.youtube.com/watch?v=LnyBo6m4ods>

https://archive.fosdem.org/2020/schedule/event/continuous_delivery_starts_with_continuous_infrastructure



In case of fire

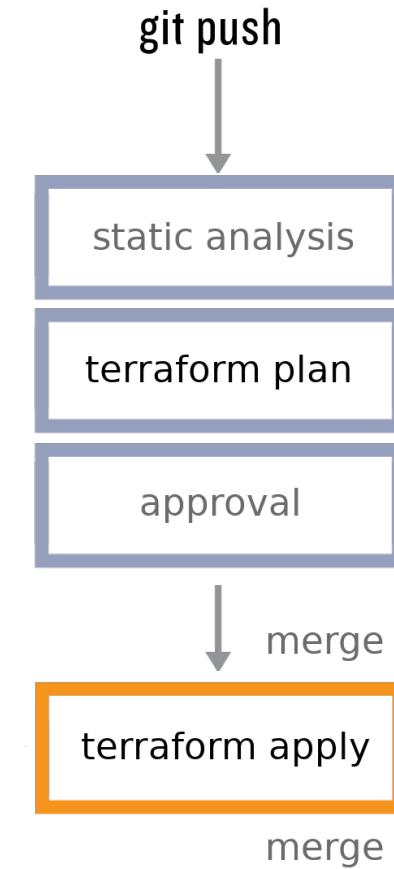


- 1. git commit
- 2. git push
- 3. leave building

In case of fire



- ⦿ 1. git commit
- ⬆ 2. git push
- 🏃 3. leave building



git push

```
graph TD; A[git push] --> B[static analysis]; B --> C[terraform plan]; C --> D[approval]; D --> E[terraform apply]; E -- merge --> F[merge]; E -- merge --> G[merge]
```

static analysis

terraform plan

approval

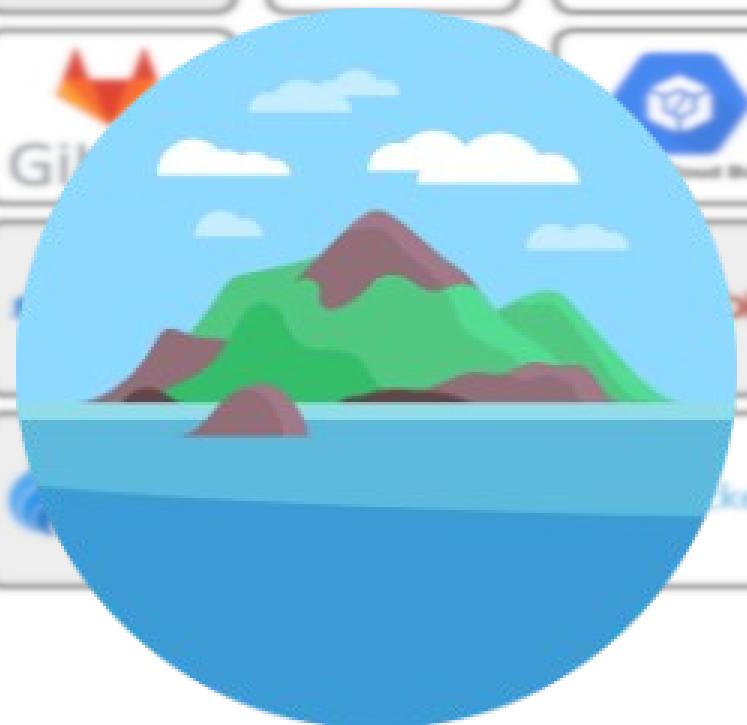
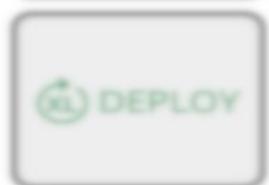
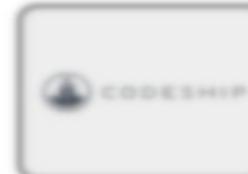
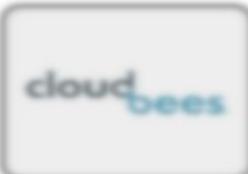
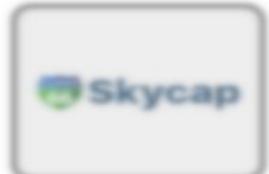
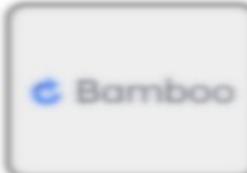
terraform apply

merge

merge

Use what works





terraform apply --target



State management



Variable management



Lock file



Static analysis



TFSEC

checkov
by bridgecrew

Terrascan

Static analysis

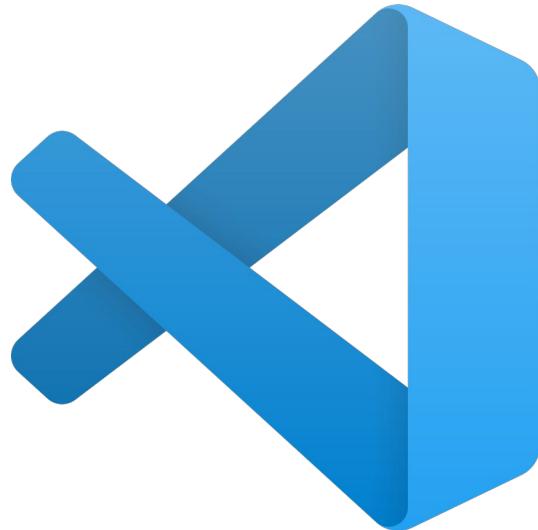
```
$ terraform validate
```

```
$ terraform fmt
```

TFLint

⌚ <https://github.com/terraform-linters/tflint>

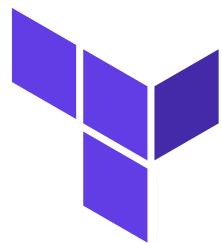
IDE



🔗 <https://github.com/hashicorp/vscode-terraform>

```
"[terraform)": {  
    "editor.formatOnSave": true  
}  
"terraform-ls.experimentalFeatures": {  
    "validateOnSave": true  
}
```

Terraform Cloud



HashiCorp

Terraform Cloud

Atlantis

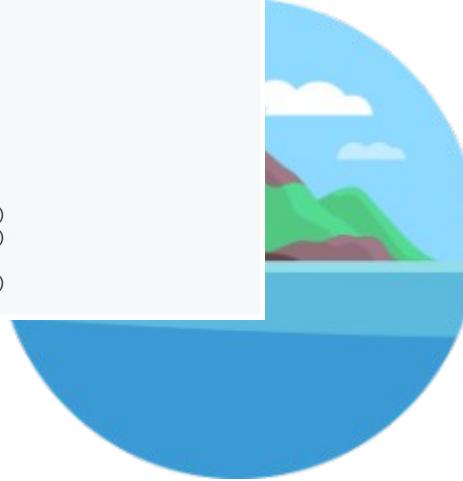


⌚ <https://github.com/runatlantis/atlantis>

Atlantis

```
# module.apps.kubernetes_deployment.deployment is tainted, so must be replaced
-/+ resource "kubernetes_deployment" "deployment" {
  ~ id              = "some-app" -> (known after apply)
  wait_for_rollout = true

  ~ spec {
    - active_deadline_seconds      = 0 -> null
    - automount_service_account_token = false -> null
    dns_policy                      = "ClusterFirst"
    host_ipc                         = false
    host_network                     = false
    host_pid                          = false
    + hostname                       = (known after apply)
    + node_name                      = (known after apply)
    restart_policy                   = "Always"
    + service_account_name          = (known after apply)
```



⌚ <https://github.com/runatlantis/atlantis>

Atlantis

```
# module.apps.kubernetes_deployment.deployment is tainted, so must be replaced
-/+ resource "kubernetes_deployment" "deployment" {
  ~ id              = "some-app" -> (known after apply)
  wait_for_rollout = true

  ~ spec {
    - active_deadline_seconds      = 0 -> null
    - automount_service_account_token = false -> null
    dns_policy                      = "ClusterFirst"
    host_ipc                        = false
    host_network                     = false
    host_pid                         = false
    + hostname                       = (known after apply)
    + node_name                      = (known after apply)
    restart_policy                  = "Always"
    + service_account_name          = (known after apply)

# module.apps.kubernetes_deployment.deployment is tainted, so must be replaced
-/+ resource "kubernetes_deployment" "deployment" {
  ~ id              = "some-app" -> (known after apply)
  wait_for_rollout = true

  ~ spec {
    - active_deadline_seconds      = 0 -> null
    - automount_service_account_token = false -> null
    dns_policy                      = "ClusterFirst"
    host_ipc                        = false
    host_network                     = false
    host_pid                         = false
    + hostname                       = (known after apply)
    + node_name                      = (known after apply)
    restart_policy                  = "Always"
    + service_account_name          = (known after apply)
```



/runatlantis/atlantis

tfarbe

```
# module.apps.kubernetes_deployment.deployment is tainted, so must be replaced
-/+ resource "kubernetes_deployment" "deployment" {
  ~ id                  = "some-app" -> (known after apply)
  wait_for_rollout = true

  ~ spec {
    - active_deadline_seconds      = 0 -> null
    - automount_service_account_token = false -> null
    dns_policy                      = "ClusterFirst"
    host_ipc                         = false
    host_network                     = false
    host_pid                          = false
    + hostname                        = (known after apply)
    + node_name                       = (known after apply)
    restart_policy                   = "Always"
    + service_account_name           = (known after apply)

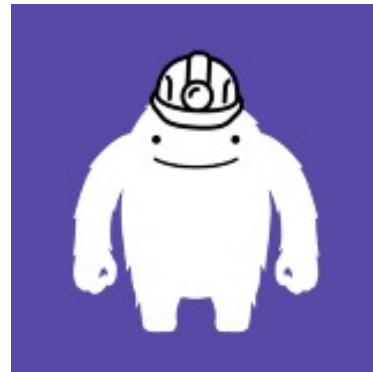
# module.apps.kubernetes_deployment.deployment is tainted, so must be replaced
-/+ resource "kubernetes_deployment" "deployment" {
  ~ id                  = "some-app" -> (known after apply)
  wait_for_rollout = true

  ~ spec {
    - active_deadline_seconds      = 0 -> null
    - automount_service_account_token = false -> null
    dns_policy                      = "ClusterFirst"
    host_ipc                         = false
    host_network                     = false
    host_pid                          = false
    + hostname                        = (known after apply)
    + node_name                       = (known after apply)
    restart_policy                   = "Always"
    + service_account_name           = (known after apply)
```

```
# module.apps.kubernetes_service.service will be updated in-place
~ resource "kubernetes_service" "service" {
  ~ id                  = "some-app"          30
  load_balancer_ingress = []
  ~ metadata {          31
    annotations = {}  32
    generation   = 0   33
  ~ labels {           34
    "app.kubernetes.io/managed-by" = "terraform" 35
    "app.kubernetes.io/version"   = "latest"     36
  }                   37
  ~ becomes:          38
  ~ id                = "some-app"          39
  ~ load_balancer_ingress = []  40
  ~ metadata {          41
    annotations = {}  42
    generation   = 0   43
  ~ labels {           44
    "app.kubernetes.io/managed-by" = "terraform" 45
    "app.kubernetes.io/version"   = "latest"     46
  }                   47
  ~ becomes:          48
# module.apps.kubernetes_deployment.deployment is tainted, so must be replaced
-/+ resource "kubernetes_deployment" "deployment" {
  ~ id                  = "some-app"          49
  wait_for_rollout = true

  ~ spec {
    - active_deadline_seconds      = 0 -> null
    - automount_service_account_token = false -> null
    dns_policy                      = "ClusterFirst"
    host_ipc                         = false
    host_network                     = false
    host_pid                          = false
    + hostname                        = (known after apply)
    + node_name                       = (known after apply)
    restart_policy                   = "Always"
    + service_account_name           = (known after apply)
```

terragrunt



⌚ <https://github.com/gruntwork-io/terragrunt>

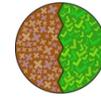
Tests

end-2-end



Terratest

⌚ <https://github.com/gruntwork-io/terratest>



Kitchen-Terraform

⌚ <https://github.com/newcontext-oss/kitchen-terraform>

unit



Conftest

⌚ <https://github.com/open-policy-agent/conftest>



terrafrom-compliance

⌚ <https://github.com/terraform-compliance/cli>

Mock Infrastructure



⌚ <https://github.com/localstack/localstack>

Versions

tfenv



<https://github.com/tfutils/tfenv>

<https://wiki.nikitavoloboev.xyz/devops/terraform>

Example/Demos

Github action

```
name: "Terraform"
...
- name: Setup Terraform
  uses: hashicorp/setup-terraform@v1
  with:
    cli_config_credentials_token: ${{ secrets.TF_API_TOKEN }}
- name: Terraform Format
  id: fmt
  run: terraform fmt -check
- name: Terraform Init
  id: init
  run: terraform init
- name: Terraform Plan
  id: plan
  if: github.event_name == 'pull_request'
  run: terraform plan -no-color
  continue-on-error: true
- name: Terraform Plan Status
```

CircleCI

```
version: '2.1'
orbs:
  terraform: 'circleci/terraform@dev:alpha'
workflows:
  deploy_infrastructure:
    jobs:
      - terraform/fmt:
          checkout: true
          context: terraform
      - terraform/validate:
          checkout: true
          context: terraform
          requires:
            - terraform/fmt
      - terraform/plan:
          checkout: true
          context: terraform
          persist-workspace: true
          requires:
            - terraform/validate
      - terraform/apply:
```

Jenkins

```
stages {
    stage('Init') {
        steps {
            sh 'terraform init'
        }
    }
    stage('Validate') {
        steps {
            sh 'terraform fmt'
            sh 'terraform validate'
        }
    }
    stage('Plan') {
        steps {
            sh 'terraform plan'
        }
    }
    stage('Apply') {
        steps {
            sh 'terraform apply -input=false -auto-approve'
        }
    }
    when {
        expression { env.BRANCH_NAME == "master" }
```

Visibility



Jeff Knurek

clevertech



@knurekit



jeff-knurek

photos found on Unsplash