A hands-off approach for your Terraform

Jeff Knurek
Continuous Delivery starts with Continuous Infrastructure

Kris Buytaert

UB4.136
2020-02-02

https://archive.fosdem.org/2020/schedule/event/continuous_delivery_starts_with_continuous_infrastructure
https://www.youtube.com/watch?v=LnyBo6m4ods
CI/CD works for devs.

- So many tools
- So much effort to force it

But ops then just does things manually.
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

Diagram:
- git push
  - static analysis
  - terraform plan
  - approval
  - merge
  - terraform apply
  - merge
Use what works
ci/cd works for devs

- So many tools
- So much effort to force it

But ops then just does things manually
terraform apply --target
State management
Variable management
Lock file
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
}
Atlantis

https://github.com/runatlantis/atlantis
# module.apps.kubernetes_deployment.deployment is tainted, so must be replaced

```bash
-/+ resource "kubernetes_deployment" "deployment" {
  ~ id                   = "some-app" -> (known after apply)
  wait_for_rollout = true

  ~ spec {
    - activeDeadlineSeconds = 0 -> null
    - automountServiceAccountToken = false -> null
    - dnsPolicy = "ClusterFirst"
    - hostIpc = false
    - hostNetwork = false
    - hostPid = false
    + hostname = (known after apply)
    + nodeName = (known after apply)
    + restartPolicy = "Always"
    + serviceAccountName = (known after apply)
  }
}
```

[GitHub](https://github.com/runatlantis/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_ip = false
      - host_network = false
      - host_pid = false
      + hostname = (known after apply)
      + node_name = (known after apply)
      + restart_policy = "Always"
  }
# module.apps.kubernetes_deployment.deployment is tainted, so must be replaced

```python
+- 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

```python
+- resource "kubernetes_service" "service" {
  id = "some-app"
  loadBalancerIngress = []
  
  metadata {
    annotations = {}
    generation = 0
    labels = {
      "app.kubernetes.io/managed-by" = "terraform"
      "app.kubernetes.io/version" = "1.0.1"
    }
  }
  
  spec {
    externalTrafficPolicy = TERMINAL
    loadBalancerIP = "true"
    loadBalancerSourceIP = "true"
    selector {
      node_name = (known after apply)
      service_account_name = (known after apply)
      x齒" = (known after apply)
    }
    + port (name: "http", port: 80, targetPort: 80)
    + port (name: "https", port: 443, targetPort: 443)
  }
}
```

https://github.com/jeff-knurek/tfarbe
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

terraform-compliance
- https://github.com/terraform-compliance/cli
Mock Infrastructure

https://github.com/localstack/localstack
tfenv

https://github.com/tfutilis/tfenv

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

https://learn.hashicorp.com/tutorials/terraform/github-actions
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:
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"}
        }
    }
}

https://wbassler23.medium.com/securing-secrets-for-your-iac-using-jenkins-terraform-and-ansible-vault-7009e0a7eb32
Visibility
Jeff Knurek

@knurekit
jeff-knurek

clevertech

photos found on Unsplash