

# Automate Your Infrastructure with GitOps, Terraform and Drone



Image from Europeana on [Unsplash](#)

# Hello, I'm Jim Sheldon!

## Nice to meet you :)

- Infrastructure Engineer @ Meltwater for 5 years
- I live in New Hampshire, USA with my wife and our three children



@bitberk



jimsheldon



“

**“Deliver the leading integrated media and social intelligence solution for PR, Comms and Marketing departments”**

<https://meltwater.com>

The screenshot shows the Meltwater website homepage. At the top, there's a navigation bar with links for Solutions, Products, Customer Stories, Resources, Blog, About, Support, Login, and a language selector (En). A prominent teal button labeled "Request demo" is located in the top right corner. The main content area features a large, light blue background with white text on the left side. The text reads: "We help you monitor, understand and influence the world around you." Below this text are two buttons: a teal one labeled "Request a demo" and a white one with a play icon labeled "Watch video". To the right of the text is a photograph of a woman with blonde hair, wearing glasses and a denim jacket, smiling while holding a smartphone to her ear. She is seated at a desk in an office environment, surrounded by papers, a keyboard, and a computer monitor. In the bottom right corner of the page, there's a small teal circular icon with a white speech bubble symbol.

We help you  
monitor,  
understand and  
influence the world  
around you.

Request a demo ▶ Watch video

meltwater.com

# Meltwater Products

Media Monitoring

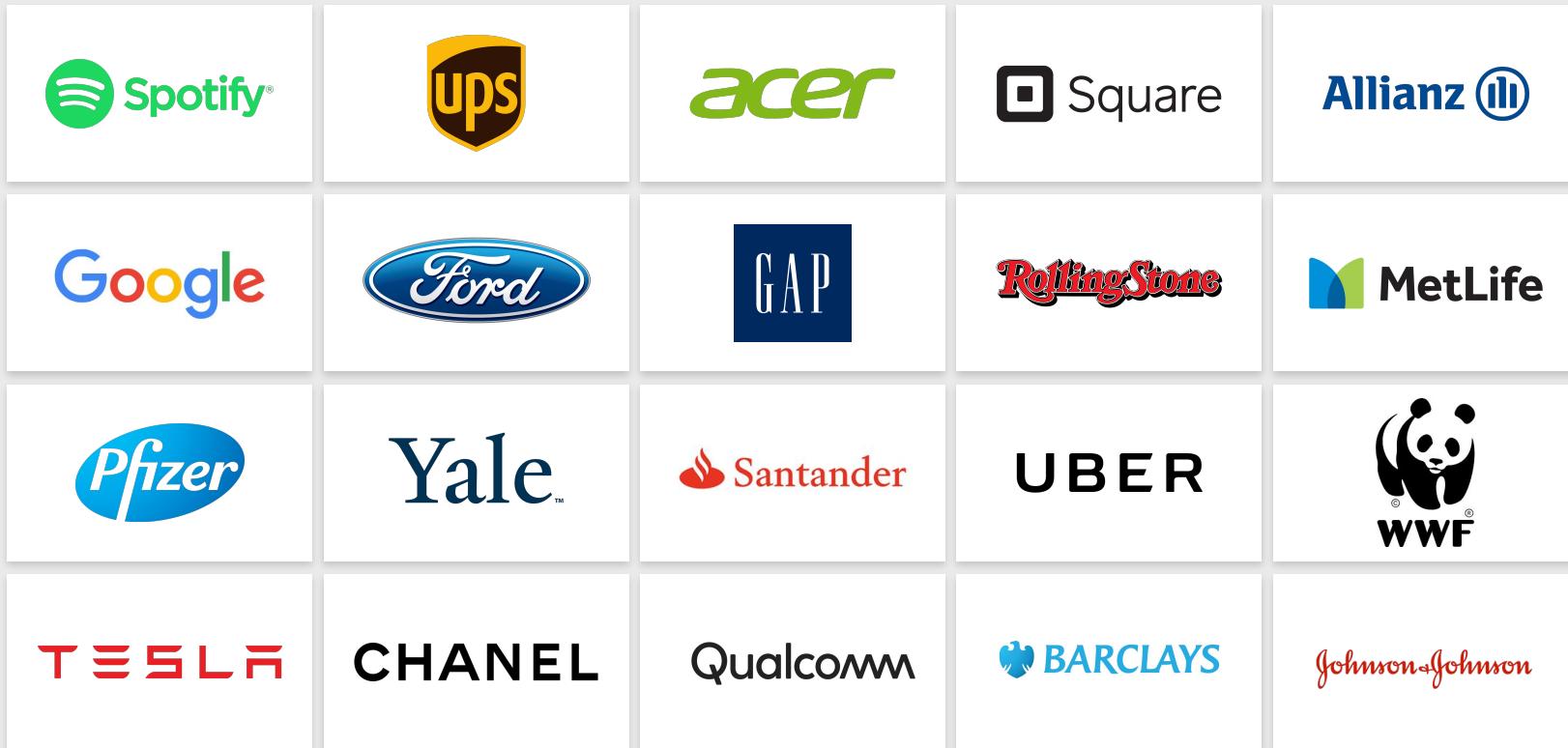
Media Outreach

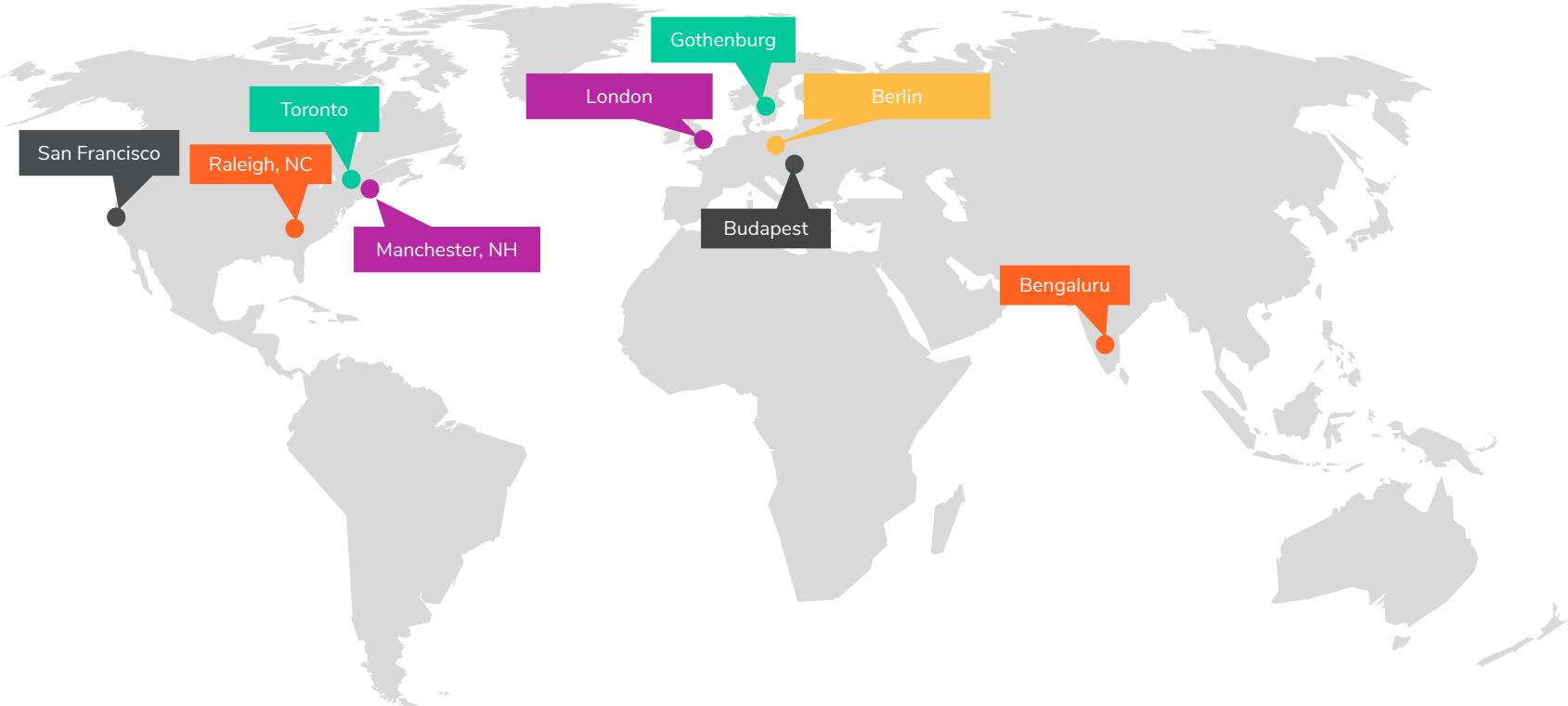
Social Listening

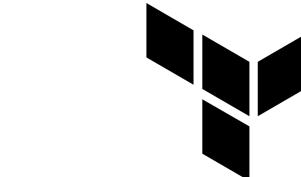
Social Publishing & Engagement

Social Influencer Management

Some of our 30K+ clients







# FOUNDATION



“

“We enable teams to **accelerate** without thinking too much about the infrastructure, allowing a rapid path from **ideation** to **prototype** to providing **business value**”

# What are we talking about?



## GitOps

Git is the “source of truth” for all running infrastructure



## Terraform

Infrastructure-as-code of choice



## Drone

CI/CD solution

## Why GitOps?

- Follow proven software development processes,  
Infrastructure-as-code *is code!*
- Easily see differences over time
- Pull requests checks let us know what is changing, and  
give us an audit trail to refer back to
- Easily roll back changes by reverting commits
- Our GitHub organization is a searchable resource

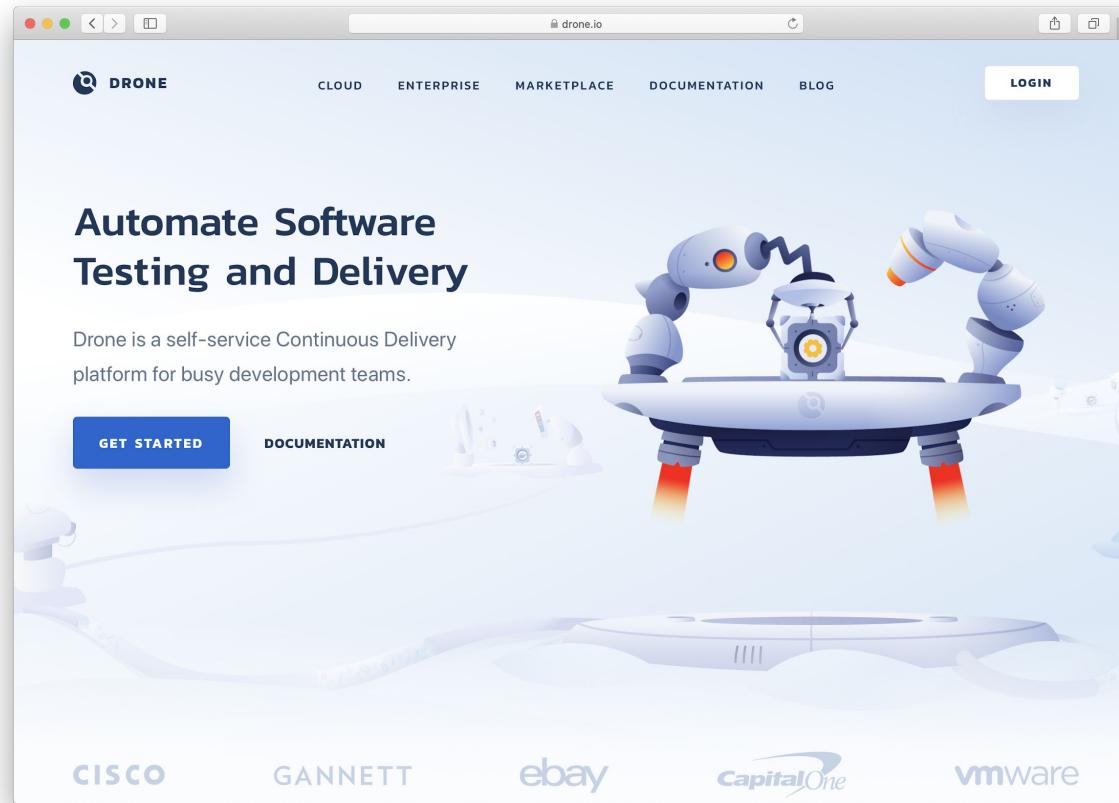
## Why Terraform?

- Cloud agnostic
- Dozens of providers (AWS, Datadog, Kubernetes, etc)
- Idempotent
- Excellent documentation
- Fairly low learning curve

## Why Drone?

- Container-native
- Unopinionated
- Large ecosystem of plugins
- GitHub for authentication
- Ease of administration
- Works for just about everything (Terraform, Kubernetes...)

<https://drone.io>



The screenshot shows the Drone.io homepage. At the top, there's a navigation bar with links for CLOUD, ENTERPRISE, MARKETPLACE, DOCUMENTATION, BLOG, and LOGIN. Below the navigation, a large banner features the text "Automate Software Testing and Delivery" in bold blue letters. To the right of the text is a 3D illustration of a futuristic drone with two robotic arms holding a small device. The drone has a circular base with a propeller at the bottom. Below the banner, there are two buttons: "GET STARTED" in white on a blue background and "DOCUMENTATION" in white on a grey background. At the bottom of the page, there are logos for Cisco, Gannett, eBay, CapitalOne, and VMware.

<https://cloud.drone.io>

The screenshot shows the homepage of the Drone Cloud website. At the top, there's a navigation bar with links for Cloud, Enterprise, Plugins, Support, and a prominent green LOGIN button. To the left of the main content area is a dark sidebar featuring a logo of a stylized 'D' inside a circle, followed by the text "Continuous Integration, Free for the Open Source Community". Below this, a brief description states: "Drone Cloud is a free Continuous Integration service for the Open Source community, powered by blazing fast bare-metal servers." Two buttons, "LOGIN" and "READ THE DOCS", are located at the bottom of this sidebar. To the right of the sidebar is a large graphic of several server racks connected by a network of glowing blue lines, symbolizing a cloud infrastructure. The main content area below has a white background and features the heading "Accelerating Open Source Development". It contains three callout boxes: "Multiple Architectures" (describing upstreaming support for diverse Arm ecosystems), "Blazing Fast, Bare Metal Servers" (describing the use of Packet's infrastructure), and "100% free for Open Source" (mentioning sponsorships). Each callout box includes a small icon related to its content.

Cloud Enterprise Plugins Support **LOGIN**

Continuous Integration,  
Free for the Open Source Community

Drone Cloud is a free Continuous Integration service for the Open Source community, powered by blazing fast bare-metal servers.

**LOGIN** **READ THE DOCS**

Accelerating Open Source Development

**Multiple Architectures**  
Our goal is to upstream all the things! In order to do that with the diverse Arm ecosystem, we're providing gobs of CI/CD infrastructure.

**Blazing Fast, Bare Metal Servers**  
Drone Cloud runs your Continuous Integration workloads on blazing fast, bare metal infrastructure owned by Packet.

**100% free for Open Source**  
Drone Cloud would not be possible without our generous sponsors. If you are interested in becoming a sponsor please [contact us](#).

## Our Internal Drone Service

**1,382 repos**



**227 users**



**1,000+ / day**





# 28

*Meltwater Terraform repositories with Drone pipelines run in the past 30 days*



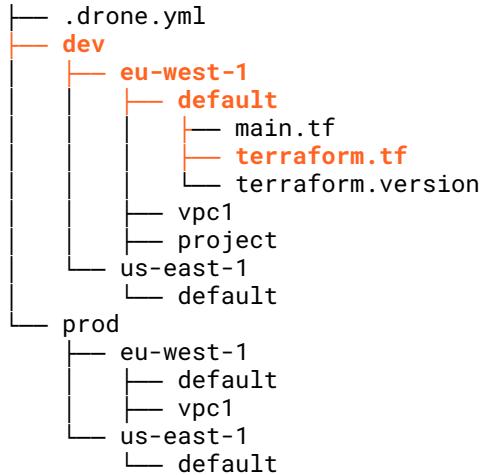
# 448

*Individual Meltwater Terraform pipelines run in the past 30 days*

# Repository directory structure

```
├── .drone.yml
└── dev
    ├── eu-west-1
    │   ├── default
    │   └── vpc1
    └── us-east-1
        └── default
└── prod
    ├── eu-west-1
    │   ├── default
    │   └── vpc1
    └── us-east-1
        └── default
```

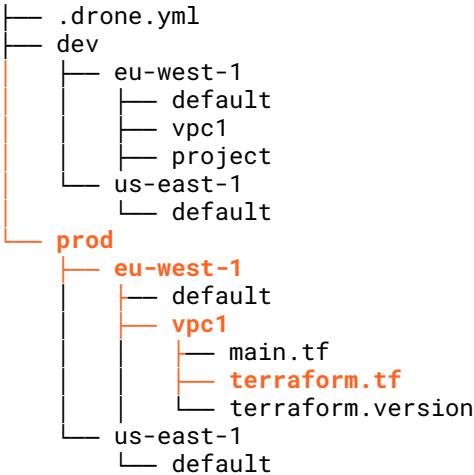
# State files



**terraform.tf**

```
terraform {
  backend "s3" {
    bucket = "tf-state-mw-myteam-dev"
    key    = "eu-west-1/default.tfstate"
    region = "eu-west-1"
  }
}
```

# State files



**terraform.tf**

```
terraform {  
  backend "s3" {  
    bucket = "tf-state-mw-myteam-prod"  
    key    = "eu-west-1/vpc1.tfstate"  
    region = "eu-west-1"  
  }  
}
```

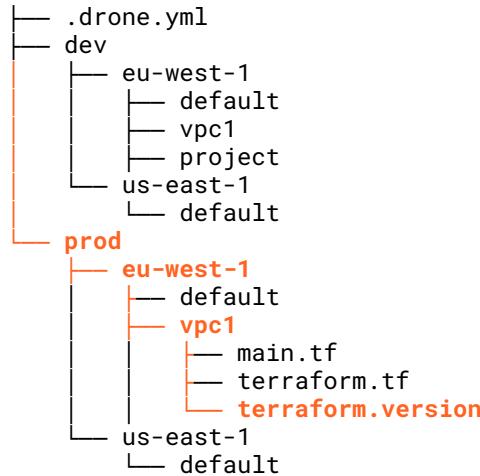
# Terraform version files

```
└── .drone.yml
    ├── dev
    │   ├── eu-west-1
    │   │   ├── default
    │   │   │   ├── main.tf
    │   │   │   └── terraform.version
    │   │   └── terraform.version
    │   ├── vpc1
    │   ├── project
    │   └── us-east-1
    │       └── default
    └── prod
        ├── eu-west-1
        │   ├── default
        │   ├── vpc1
        └── us-east-1
            └── default
```

`terraform.version`

0.12.14

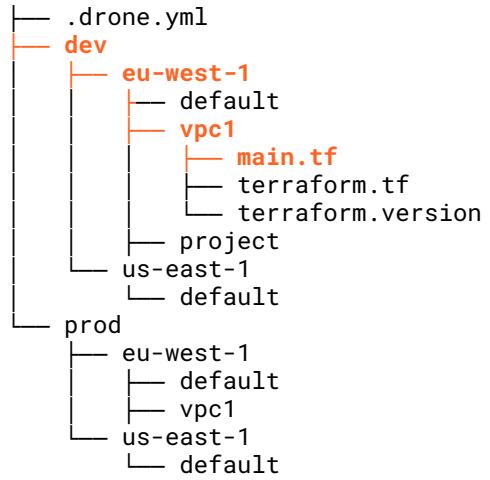
# Terraform version files



`terraform.version`

0.11.14

# Modules

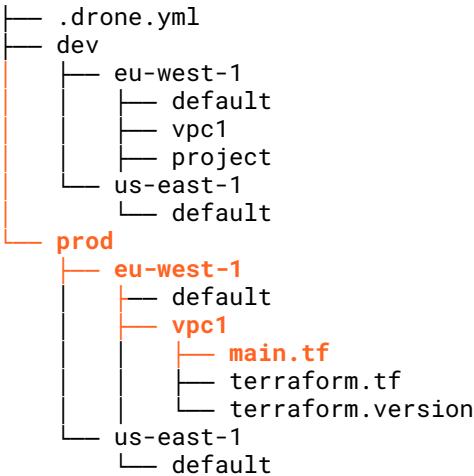


## main.tf

```
module "example" {
  source =
  "git::git@github.com:meltwater/terraform-aws-
  modules.git//example?ref=example_v2"

  foo = "bar"
}
```

# Modules



**main.tf**

```
module "example" {
  source =
  "git::git@github.com:meltwater/terraform-aws-
  modules.git//example?ref=example_v1"

  foo = "bar"
}
```

.drone.yml

pipeline



Image from Mike Benna on [Unsplash](#)

## .drone.yml

```
kind: pipeline
type: docker
name: default

concurrency:
  limit: 1

trigger:
  branch:
    - master

volumes:
- name: dot-aws
  temp: {}
```

# .drone.yml

## steps

```
- name: files_changed
  image: alpine:3
  commands:
    - apk add git
    - |-
      for DIR in $(git --no-pager diff --name-only
$DRONE_COMMIT_BEFORE..$DRONE_COMMIT_AFTER | sed
"s/^\\(\\.\\.\\/.*)\\2/" | sed "s/^\\(.*)\\1/" |
sort | uniq ); do
      for VERSION in $(find . -name terraform.version | sed
"s/^\\(\\.\\.\\/.*)\\2/" | sed "s/^\\(.*)\\1/" |
sort | uniq); do
        case "$DIR" in
          "$VERSION")
            echo $VERSION >> terraform_dirs
            echo "INFO Adding $VERSION to terraform_dirs";;
          esac
        done
      done
```

# .drone.yml

## steps

```
- name: aws_credentials
  image: amazon/aws-cli:2.0.31
  volumes:
  - name: dot-aws
    path: /root/.aws
  environment:
    ASSUME_ROLE_EXTERNAL_ID:
      from_secret: assume_role_external_id
    AWS_ACCOUNT_ID:
      from_secret: demo_account_id
  commands:
    - aws configure set role_arn
"arn:aws:iam::$AWS_ACCOUNT_ID:role/DroneTerraformContainerDays"
    - aws configure set region eu-west-1
    - aws configure set credential_source Ec2InstanceMetadata
    - aws configure set external_id $ASSUME_ROLE_EXTERNAL_ID
    - aws sts get-caller-identity
```

# .drone.yml

## steps

```
- name: terraform_${DRONE_BUILD_EVENT}
  image: alpine:3
  volumes:
    - name: dot-aws
      path: /root/.aws
  environment:
    AWS_METADATA_URL: http://localhost/not/existent/url
    AWS_SDK_LOAD_CONFIG: 1
  commands:
    - apk add bash curl git
    - git clone --branch v1.0.2 https://github.com/tfutils/tfenv.git /opt/tfenv
    - ln -s /opt/tfenv/bin/* /usr/local/bin
    - |-
        ROOT_DIR=$(pwd)
        for DIR in $(cat terraform_dirs); do
          echo "INFO  Changing to $DIR"
          cd $ROOT_DIR/$DIR
          tfenv install $(cat terraform.version)
          tfenv use $(cat terraform.version)
          terraform init
          if [ "$DRONE_BUILD_EVENT" == "pull_request" ]; then
            terraform plan
          elif [ "$DRONE_BUILD_EVENT" == "push" ]; then
            terraform apply
          fi
        done
```

<https://github.com/meltwater/drone-convert-pathschanged>

The screenshot shows a GitHub repository page for the project `drone-convert-pathschanged`. The repository has 10 stars, 27 forks, and 5 open issues. The code tab is selected, showing a list of 34 commits from the master branch. The commits include initial commits of alpha code, adding parse pipelines, moving tests, fixing documentation, and adding Prometheus metrics. The README.md file describes the project as a Drone conversion extension. The repository also includes a Readme, Apache-2.0 License, four releases (the latest being 0.1.0), two contributors (jimsheldon and apoorva-marisomaradhy), and a Go language profile.

Branch: master

34 commits 1 branch 4 tags

Initial commit of alpha code  
Add parse pipelines include test (#24)  
initial commit of alpha code  
move test to its own step (#23)  
initial commit of alpha code  
LICENSE is not markdown  
[SKIP CI] fix token documentation (#21)  
Add parse pipelines include test (#24)  
Add tests (#22)  
Prometheus metrics

README.md

build success

A Drone conversion extension to include/exclude pipelines and steps based on paths changed.

Please note this project requires Drone server version 1.4 or higher.

About

Drone extension to include/exclude pipelines and pipeline steps based on paths changed

Readme Apache-2.0 License

Releases 4

0.1.0 (Latest) on Feb 4 + 3 releases

Contributors 2

jimsheldon apoorva-marisomaradhy

Languages

Go 99.1% Shell 0.9%

## conversion

## extension

```
kind: pipeline
type: docker
name: default

steps:
- name: readme
  image: busybox
  commands:
  - echo "README.md was changed"
when:
  paths:
    include:
    - README.md
```

# anchors

```
aws_credentials: &aws_credentials
  image: amazon/aws-cli:2.0.31
  volumes:
    - name: dot-aws
      path: /root/.aws
  commands:
    - aws configure --profile $TF_ENV-$TF_REGION set role_arn
      "arn:aws:iam::$AWS_ACCOUNT_ID:role/DroneTerraformContainerDays"
    - aws configure --profile $TF_ENV-$TF_REGION set region $TF_REGION
    - aws configure --profile $TF_ENV-$TF_REGION set credential_source Ec2InstanceMetadata
    - aws configure --profile $TF_ENV-$TF_REGION set external_id $ASSUME_ROLE_EXTERNAL_ID

- <<: *aws_credentials
  name: dev_eu-west-1_aws_credentials
  environment:
    ASSUME_ROLE_EXTERNAL_ID:
      from_secret: assume_role_external_id
    AWS_ACCOUNT_ID:
      from_secret: demo_account_id
    TF_ENV: dev
    TF_REGION: eu-west-1
  when:
    paths:
      include:
        - dev/eu-west-1/**
```

# anchors

```
terraform: &terraform
  image: alpine:3
  volumes:
    - name: dot-aws
      path: /root/.aws
  commands:
    - apk add bash curl git
    - git clone --branch v1.0.2 https://github.com/tfutils/tfenv.git /opt/tfenv
    - ln -s /opt/tfenv/bin/* /usr/local/bin
    - |-
      ROOT_DIR=$(pwd)
      for DIR in $(cat terraform_dirs | grep ^$TF_ENV/$TF_REGION); do
        echo "INFO  Changing to $DIR"
        cd $ROOT_DIR/$DIR
        tfenv install $(cat terraform.version)
        tfenv use $(cat terraform.version)
        terraform init
        if [ "$DRONE_BUILD_EVENT" == "pull_request" ]; then
          terraform plan
        elif [ "$DRONE_BUILD_EVENT" == "push" ]; then
          terraform apply
        fi
      done

- <<: *terraform
  name: dev_eu-west-1_terraform_${DRONE_BUILD_EVENT}
  environment:
    AWS_METADATA_URL: http://localhost/not/existent/url
    AWS_SDK_LOAD_CONFIG: 1
    AWS_PROFILE: dev-eu-west-1
    TF_ENV: dev
    TF_REGION: eu-west-1
  when:
    paths:
      include:
        - dev/eu-west-1/**
```



# DEMO!

<https://github.com/jimsheldon/terraform-containerdays>

The screenshot shows a GitHub repository page for 'jimsheldon / terraform-containerdays'. The repository is private, has 1 branch, and 0 tags. It contains 26 commits from 'jimsheldon' revertting changes related to VPC configuration. The README.md file describes the repository as a demo for a talk on "Automate Your Infrastructure With GitOps, Terraform and Drone".

**Code** | **Pull requests** | **Actions** | **Security** | **Insights**

**Code** | **1 branch** | **0 tags** | **26 commits**

**About**  
Configuration for talk given at ContainerDays

**Languages**  
HCL 100.0%

**README.md**

## terraform-containerdays

Demo repository for talk on "Automate Your Infrastructure With GitOps, Terraform and Drone"

© 2020 GitHub, Inc. Terms Privacy Security Status Help Contact GitHub Pricing API Training Blog About

# IAM role

(your account)

```
resource "aws_iam_role" "role_drone" {
  name          = "DroneTerraformContainerDays"
  assume_role_policy = data.aws_iam_policy_document.assume_policy_drone.json
}

data "aws_iam_policy_document" "assume_policy_drone" {
  statement {
    effect = "Allow"
    principals {
      identifiers = [
        data.aws_kms_secrets.drone.plaintext["trusted_account_arn"],
      ]
      type = "AWS"
    }
    actions = ["sts:AssumeRole"]
    condition {
      test      = "StringEquals"
      values    = [data.aws_kms_secrets.drone.plaintext["external_id"]]
      variable = "sts:ExternalId"
    }
  }
}
```

# IAM role (drone account)

```
data "aws_iam_policy_document" "foundation" {
  statement {
    effect      = "Allow"
    actions     = ["sts:AssumeRole"]
    resources   = [
      "arn:aws:iam::${var.foundation_account_id}:role/DroneTerraformContainerDays",
    ]
  }
}

resource "aws_iam_role_policy" "foundation" {
  name      = "allow_foundation_assume"
  role      = "${data.aws_iam_role.drone_agent.id}"
  policy    = "${data.aws_iam_policy_document.foundation.json}"
}
```



# kubernetes

# K8S directory structure

```
.drone.yml
dev
  └── prometheus
  └── thanos
  └── grafana
  └── blackbox
prod
  └── prometheus
  └── grafana
  └── blackbox
```

<https://github.com/drone-runners/drone-runner-kube>

The screenshot shows the GitHub repository page for `drone-runners/drone-runner-kube`. The page includes a navigation bar with links to Why GitHub?, Team, Enterprise, Explore, Marketplace, and Pricing. The search bar is empty. There are buttons for Sign in and Sign up. The repository name is displayed in the header, along with a star count of 58 and a fork count of 26. Below the header, there are tabs for Code, Pull requests (2), Actions, Projects, Security, and Insights. The Code tab is selected. A dropdown menu shows the current branch is master. To the right of the code area, there is an "About" section describing the project as a Drone runner that executes a pipeline using native Kubernetes resources. It includes links to Readme and View license. The "Releases" section shows 4 tags. The "Contributors" section lists 14 contributors with their profile icons. The "Languages" section shows Go at 99.5% and Other at 0.5%. The main content area displays a list of commits from bradrydzewski, showing changes to .github, command, docker, engine, internal, licenses, samples, scripts, .drone.yml, .gitignore, BUILDING.md, CHANGELOG.md, LICENSE.md, README.md, and go.mod files.

Branch: master

Go to file Code

bradrydzewski committed 21e352d 14 days ago

remove platform from runner filter 8 months ago

enable environment extensions 14 days ago

minor fixes after review 3 months ago

enable environment extensions 14 days ago

matching image and registry hostname should exclude proto 6 months ago

stub project 9 months ago

fix typo to snake case 3 months ago

use buildx to publish placeholder image 9 months ago

remove unused fields 9 months ago

improve formatting when dumping manifest 9 months ago

Minor fix in BUILDING.md 6 months ago

docs: update CHANGELOG.md 3 months ago

stub project 9 months ago

update readme [ci skip] 9 months ago

enable environment extensions 14 days ago

About

Drone runner that executes a pipeline using native Kubernetes resources

Readme

View license

Releases

4 tags

Contributors 14

+ 3 contributors

Languages

Go 99.5% Other 0.5%

.drone.yml

k8s runner

```
kind: pipeline
type: kubernetes
name: default

steps:
- name: greeting
  image: alpine
  commands:
    - echo "hello world"
```

<https://www.terraform.io/docs/providers/kubernetes/index.html>

The screenshot shows a web browser displaying the Terraform documentation for the Kubernetes provider. The page has a purple header with the Terraform logo and navigation links for Intro, Tutorials, Docs, Community, Enterprise, Download, GitHub, Sign In, and Create Account. The main content area has a white background with a title 'Kubernetes Provider' and a 'View on Terraform Registry' link. On the left, there's a sidebar with a tree view of available providers and data sources. The 'Kubernetes Provider' node is expanded, showing its sub-resources: all\_namespaces, config\_map, ingress, namespace, secret, service\_account, service, and storage\_class. Below the sidebar, a section titled 'Example Usage' contains a code snippet for creating a Kubernetes namespace:

```
provider "kubernetes" {
  config_context_auth_info = "ops"
  config_context_cluster    = "mycluster"
}

resource "kubernetes_namespace" "example" {
  metadata {
    name = "my-first-namespace"
  }
}
```

At the bottom of the page, there's a section titled 'Kubernetes versions'.

<https://katacoda.com/mw-fou>

The screenshot shows the Katacoda web interface. At the top, there's a navigation bar with links for Learn, Create, Embed, For Vendors, For Teams, For Enterprises, Search, Log In, and Sign Up. On the left, the Katacoda logo is displayed next to the text "Meltwater Foundation" and the handle "@mw-fou". Below this, there's a yellow circular icon containing a white silhouette of a head. To the right, a section titled "Share Your Success" features "Share" buttons for LinkedIn and Twitter.

The main content area displays five course cards:

- Drone CD** (2 SCENARIOS): CI/CD with Drone.io will ensure students are comfortable with both the local exec and web experience of v1.x of the drone toolkit. [Start Course](#)
- ElastAlert** (1 SCENARIOS): How to write ElastAlert rules to alert on data in Elasticsearch. [Start Course](#)
- Prometheus** (2 SCENARIOS): How to use Prometheus.io to monitor your applications and infrastructure. [Start Course](#)
- StatsD** (1 SCENARIOS): Basics of StatsD and custom push-based metrics. [Start Course](#)
- Terraform** (1 SCENARIO): How to use Terraform. [Start Course](#)

<https://underthehood.meltwater.com>

The screenshot shows a web browser displaying the [under the hood](https://underthehood.meltwater.com) blog. The header features the blog's logo, the title "under the hood", and a subtitle "The official meltwater engineering blog". The navigation menu includes links for BLOG (which is active), ABOUT, JOBS, OPEN SOURCE, and ARCHIVES. The main content area displays two blog posts:

- Improving Record Linking for our Knowledge Graph (Part 2)**  
July 10, 2020 | 0 Comments  
Meltwater recently released a new product feature called Signals, which helps our customers to identify business-critical events.  
In a previous post we introduced the concept of record linking, and presented our first approach for merging information from multiple sources for our Knowledge Graph.  
In this second post, we share how we improved those initial models based on user feedback and analysis of learning features, and also present formal evaluation metrics.  
[Read on >](#)
- The Record Linking Pipeline for our Knowledge Graph (Part 1)**  
June 29, 2020 | 0 Comments  
Meltwater recently released a new product feature called Signals, which helps our customers to identify business-critical events. One of the technical systems

On the right side of the page, there are three sidebar boxes:

- About Us**: We are the engineers of [Meltwater](#). Find our open source projects at [GitHub](#). Here we write about the things we do.
- Recent Posts**:
  - [Improving Record Linking for our Knowledge Graph \(Part 2\)](#)
  - [The Record Linking Pipeline for our Knowledge Graph \(Part 1\)](#)
  - [Let's Talk about Feelings!](#)
  - [Tech Talk: Scalability Testing of a Production Kubernetes Cluster](#)
  - [The Journey To Front-End Performance – Assessing Current Performance](#)
- GitHub Repos**:
  - [addict](#): User management lib for Phoenix Framework
  - [AGDISTIS](#): AGDISTIS - Agnostic Named Entity Disambiguation

*Thank you!*

Questions?

