DevSecOps, IaC Drift & Driftctl

Conf42, Apr 29th 2021 (London, UK)

MIND THE GAP

BETWEEN THE CODE AND THE PLATFORM

Agenda

- Learning from IaC users (causes, consequences, solutions)
- Stories & Live Demos
- Q&A

Story

- ❤️ GitOps
- 🥰 Users
- 😱 Drift

TL;DR

Almost everyone has experienced infrastructure drift recently.

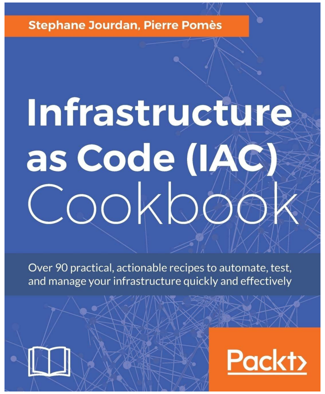
There's security implications to not knowing about drifts.

We built driftctl to help.

whoami

Stephane Jourdan:

- @sjourdan (Twitter, GitHub, GitLab,...)
- 20 years (Dev)Ops
- Co-founded 3 tech companies (🇨🇦 | 🇪🇺) and 1 sound studio.
- “Infrastructure-as-Code Cookbook” author
- Driftctl tool co-founder! cloudskiff/driftctl



Stephane Jourdan, Pierre Pomès
Infrastructure as Code (IAC) Cookbook
Over 90 practical, actionable recipes to automate, test, and manage your infrastructure quickly and effectively
Packt>

Definition

Infrastructure Drift / ˈɪn frəˌstrʌk tʃər drɪft /

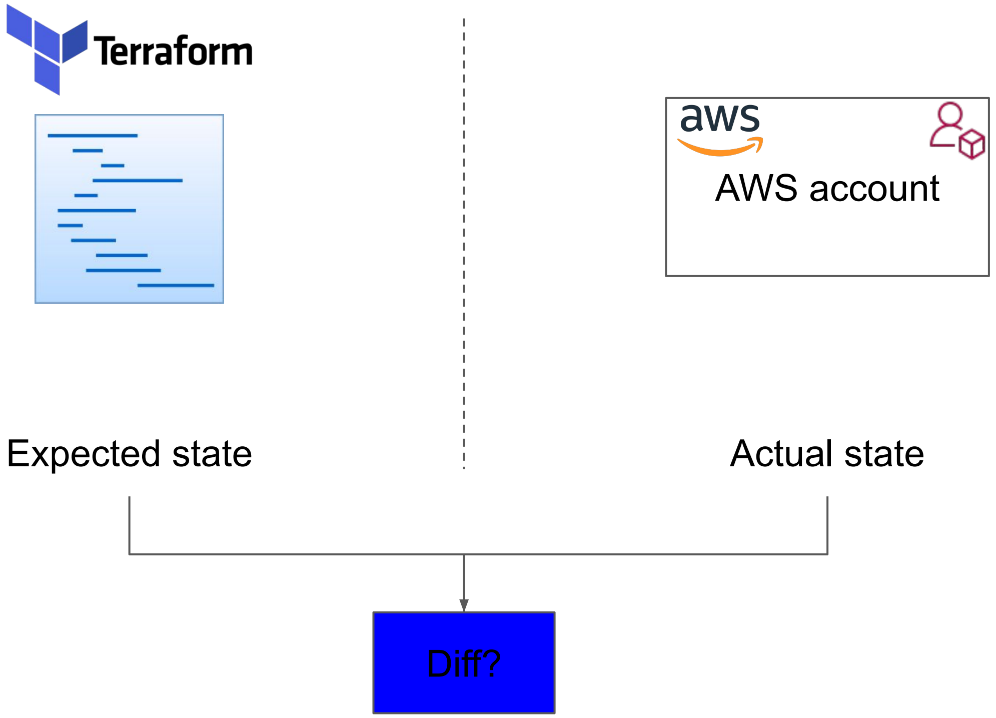
Noun

1. happens when the reality and the expectations don’t match.

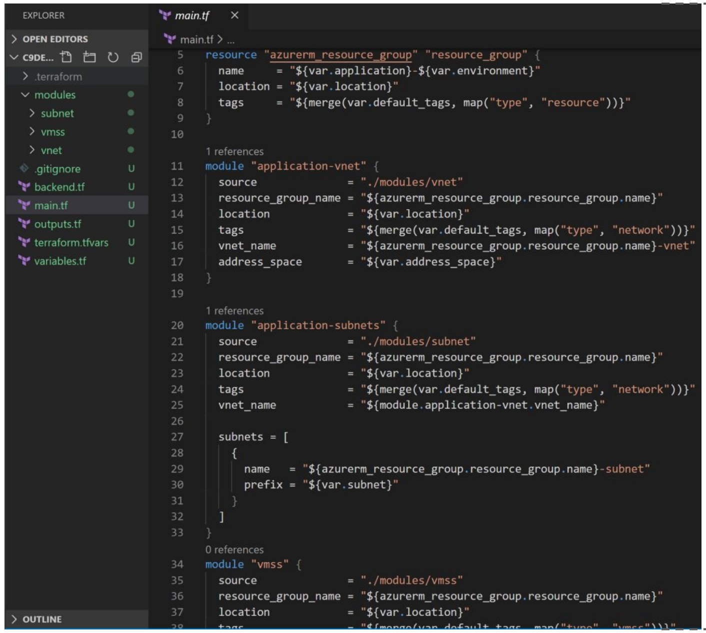
Synonyms for Infrastructure Drift

1. omg

drift?

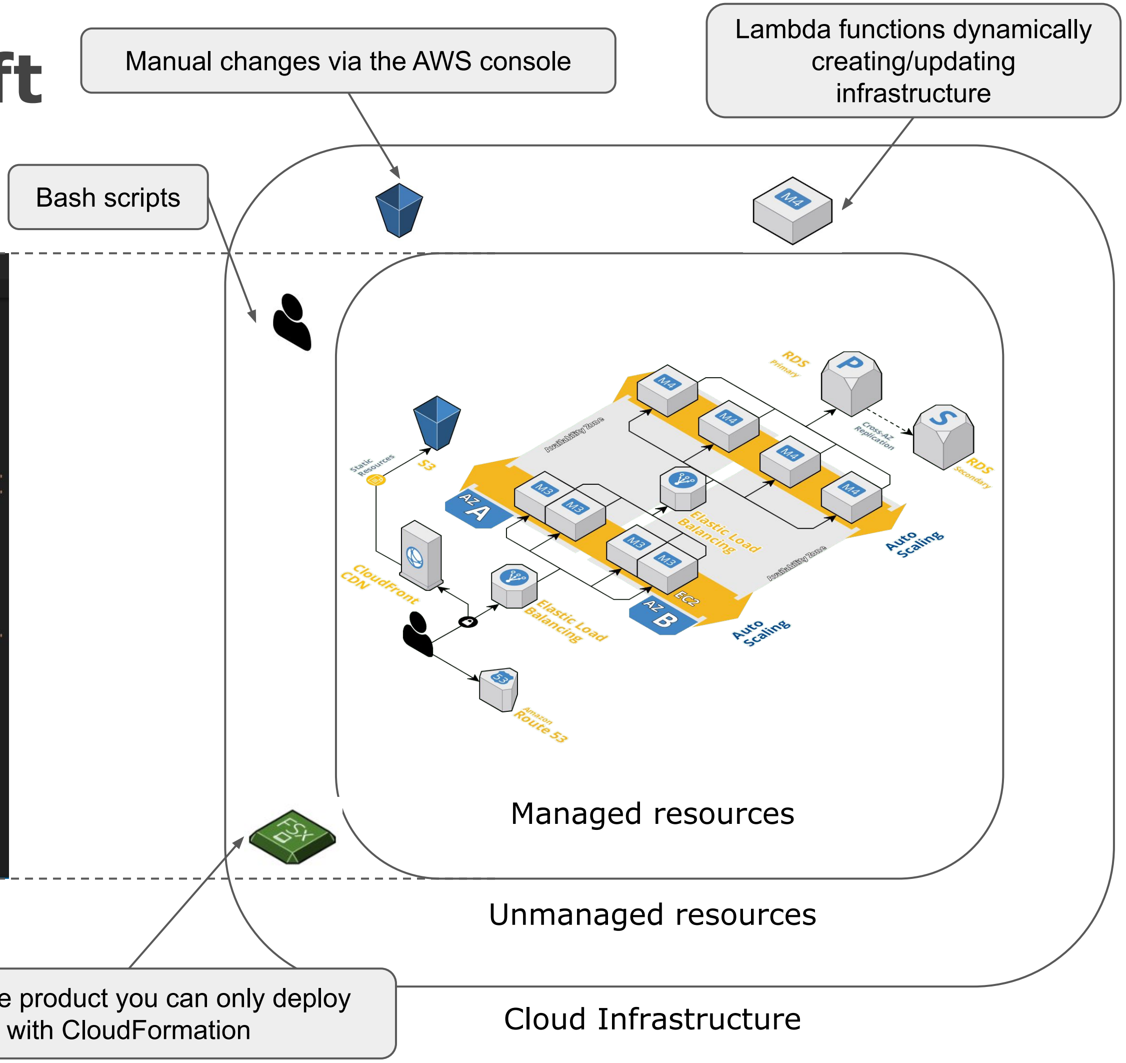

Terraform
aws
AWS account
Expected state
Actual state
Diff?

Understanding drift



```
resource "azurerm_resource_group" "resource_group" {
  name     = "${var.application}-${var.environment}"
  location = "${var.location}"
  tags     = "${merge(var.default_tags, map("type", "resource"))}"
}

1 references
module "application-vnet" {
  source              = "./modules/vnet"
  resource_group_name = "${azurerm_resource_group.resource_group.name}"
  location            = "${var.location}"
  tags                = "${merge(var.default_tags, map("type", "network"))}"
  vnet_name           = "${azurerm_resource_group.resource_group.name}-vnet"
  address_space       = "${var.address_space}"
}

1 references
module "application-subnets" {
  source              = "./modules/subnet"
  resource_group_name = "${azurerm_resource_group.resource_group.name}"
  location            = "${var.location}"
  tags                = "${merge(var.default_tags, map("type", "network"))}"
  vnet_name           = "${module.application-vnet.vnet_name}"

  subnets = [
    {
      name   = "${azurerm_resource_group.resource_group.name}-subnet"
      prefix = "${var.subnet}"
    }
  ]
}
0 references
module "vmss" {
  source              = "./modules/vmss"
  resource_group_name = "${azurerm_resource_group.resource_group.name}"
  location            = "${var.location}"
  tags                = "${merge(var.default_tags, map("type", "vmss"))}"
```

Infrastructure As Code



Why is it so important?

- ➔ Companies are massively adopting infrastructure automation
- ➔ Drift is a blind spot for DevSecOps
- ➔ Drift introduces compliance and security issues
- ➔ Parity between code and cloud is a prerequisite to shifting security to the left (moving security sooner in the development process)

National Security Agency | Cybersecurity Information

Mitigating Cloud Vulnerabilities

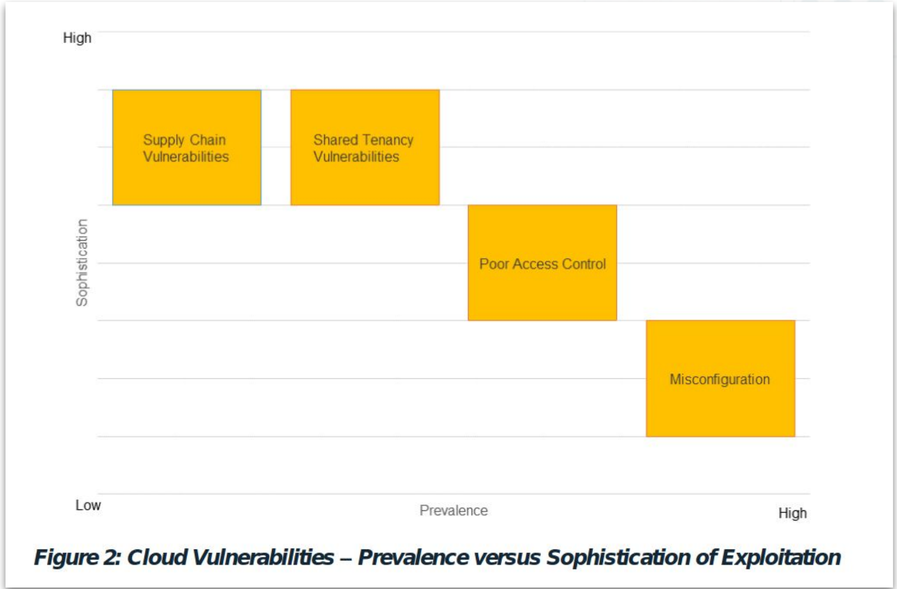


Figure 2: Cloud Vulnerabilities – Prevalence versus Sophistication of Exploitation

Blind Spots

How can you know about misconfigurations

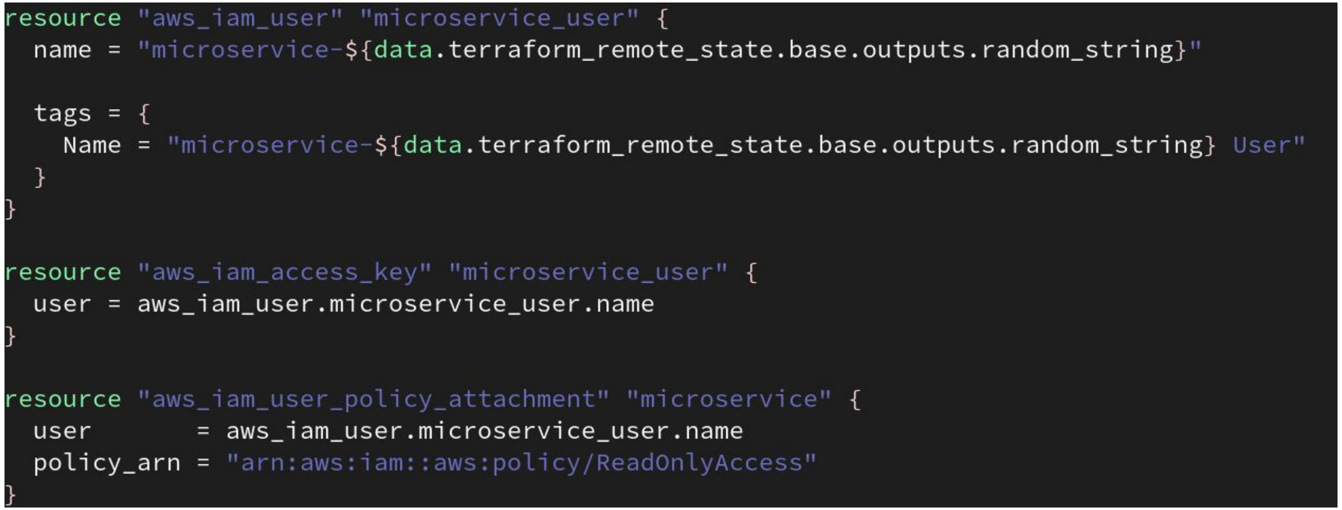
If you don't even know about existing resources?

Stories!

Quick AWS IAM Story

How a simple lambda with read-only access ended up with rogue Administrative access and keys

- *without anyone noticing*

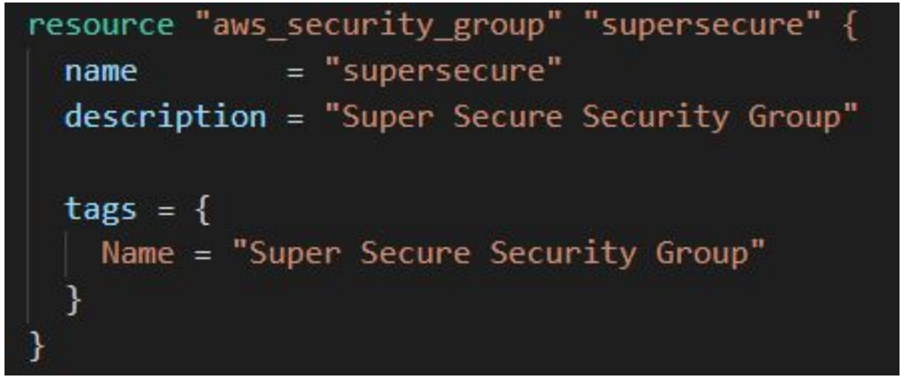


```
resource "aws_iam_user" "microservice_user" {
  name = "microservice-${data.terraform_remote_state.base.outputs.random_string}"

  tags = {
    Name = "microservice-${data.terraform_remote_state.base.outputs.random_string} User"
  }
}

resource "aws_iam_access_key" "microservice_user" {
  user = aws_iam_user.microservice_user.name
}

resource "aws_iam_user_policy_attachment" "microservice" {
  user       = aws_iam_user.microservice_user.name
  policy_arn = "arn:aws:iam::aws:policy/ReadOnlyAccess"
}
```

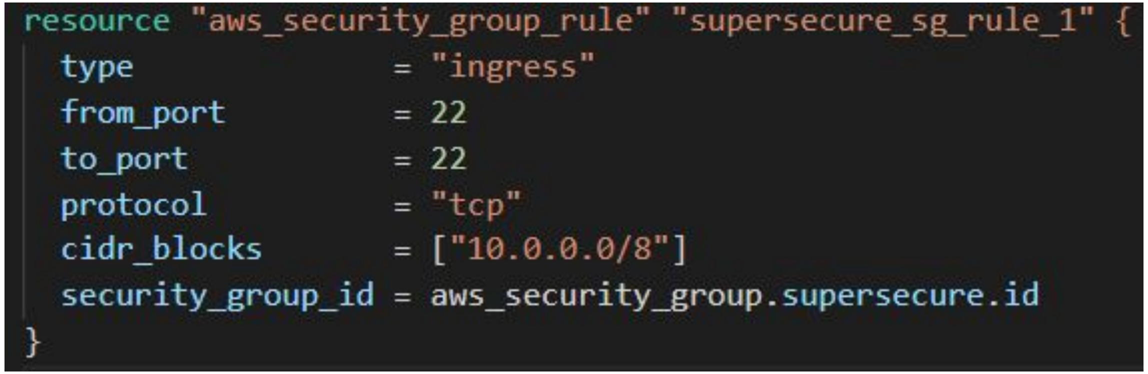
Quick AWS Security Group Story

How someone opened up everything to anyone on IPv4 & IPv6

- *without anyone noticing*



```
resource "aws_security_group" "supersecure" {
  name        = "supersecure"
  description = "Super Secure Security Group"

  tags = {
    Name = "Super Secure Security Group"
  }
}
```

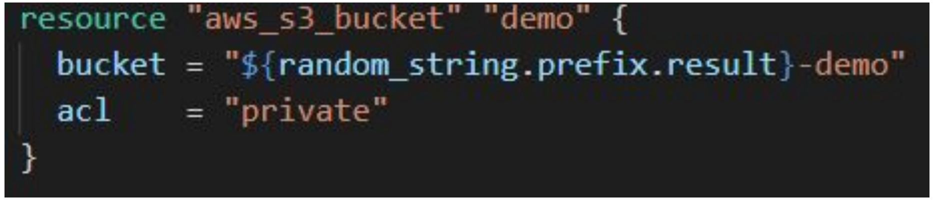


```
resource "aws_security_group_rule" "supersecure_sg_rule_1" {
  type              = "ingress"
  from_port         = 22
  to_port           = 22
  protocol          = "tcp"
  cidr_blocks       = ["10.0.0.0/8"]
  security_group_id = aws_security_group.supersecure.id
}
```

Quick AWS S3 Story

How a scripting issue created a billing nightmare

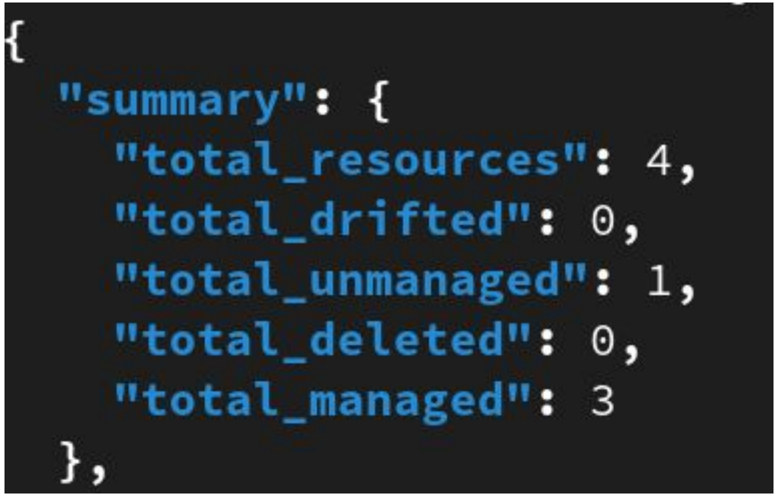
- *With only billing noticing*



```
resource "aws_s3_bucket" "demo" {
  bucket = "${random_string.prefix.result}-demo"
  acl    = "private"
}
```

JSON Output

```
$ driftctl scan --output json://./output.json
```



```
{
  "summary": {
    "total_resources": 4,
    "total_drifted": 0,
    "total_unmanaged": 1,
    "total_deleted": 0,
    "total_managed": 3
  },
```

```
$ jq '.coverage' < output.json

75
```

Driftctl Filters

```
$ driftctl scan --filter "Type=='aws_security_group_rule'"

Scanning resources: ⣷ (51)

Found unmanaged resources:

  aws_security_group_rule:

    - Type: ingress, SecurityGroup: sg-00c1621e81e5b17c1, Protocol: All,
Ports: All, Source: 0.0.0.0/0
```

JMESPATH

JMESPath is a query language for JSON.

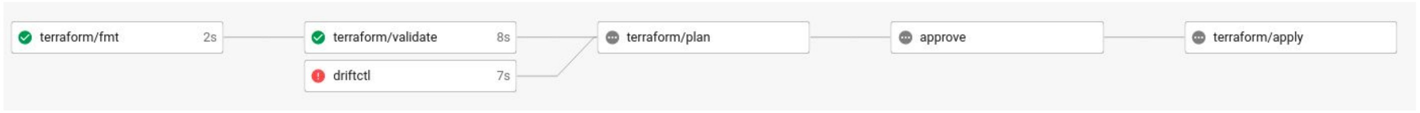
Driftctl Ignore

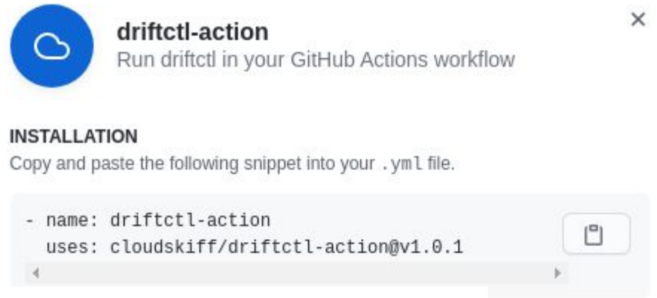
```
$ head .driftignore

aws_iam_user.terraform

[...]
```

CI Integration

Show GitOps ❤️

- Circle CI Orb
- GitHub Action
- Gitlab CI
- ...

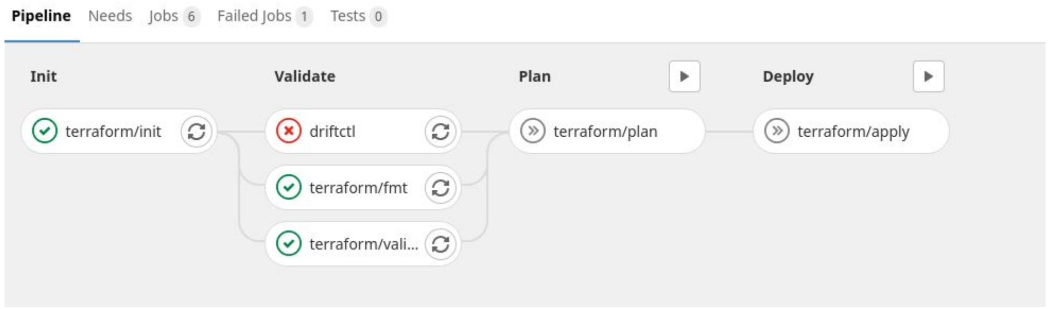




driftctl-action
Run driftctl in your GitHub Actions workflow

INSTALLATION
Copy and paste the following snippet into your `.yml` file.

```
- name: driftctl-action
  uses: cloudskiff/driftctl-action@v1.0.1
```



driftctl

Our own open-source solution for drift management

- AWS & GitHub Support (more to come)
- Terraform State support (local/S3/HTTP)
- Filtering & Ignore support
- Written in Go
- Apache 2.0 License

cloudskiff/driftctl   driftctl.com/d

driftctl

TL;DR (Closing)

driftctl

Almost everyone has experienced infrastructure drift recently.

We built driftctl to help.

MIND THE GAP

BETWEEN THE CODE AND THE PLATFORM

driftctl

Our own open-source solution for drift management

Why

- Even the best teams didn’t automate everything
- Scripts / Lambdas / Microservices are authenticated
- Customers and bosses do exist (*with admin credentials*)

driftctl