Beauty and the Build: Charlotte Tilbury's move from CloudFormation to Terraform

HashiDays London

Annem Sabah Shah

Introductions

SCAN ME

- Annem Shah
- Cloud Platform Engineering @ Charlotte Tilbury Beauty
- Connect with me on LinkedIn!

Quiz Time !!!

What is idempotency in the context of infrastructure as code?

(Prize at stake!)

GUIs GUIs!

- Always been able to click a few buttons in a GUI, and spin up a compute instance, add some networking rules.

- You have now created a hosted virtual machine, ready to use.

CloudFormation to Terraform

- Undertaking transformation work to migrate our entire CloudFormation codebase to Terraform.

- Terraform is Hashicorp’s infrastructure-as-code offering, works across multiple clouds.

- CloudFormation is AWS’s offering, only working with AWS resources.

A scary thought ...

At any given time, there would be services with something declared in the CloudFormation stack that does not match reality!

Idempotency and declarative code

CloudFormation

```
Resources:
  CloudTrail:
    DeletionPolicy: "Retain"
    UpdateReplacePolicy: "Retain”
    DependsOn: "S3BucketPolicy"
    Type: "AWS::CloudTrail::Trail"
      Properties:
```

Terraform

```
resource "aws_cloudtrail" "cloudtrail" {
  name                      = var.cloudtrail_name
  cloud_watch_logs_role_arn = aws_iam_role.cloudtrail_cloudwatch_role.arn
  s3_bucket_name            = aws_s3_bucket.cloudtrail_bucket.bucket
```

Modular, flexible Terraform

- With the migration, we wanted to ensure the Terraform codebase was reusable, templated and flexible.

- Each CloudFormation stack became a Terraform module.

HCP Terraform

- CLI interactions and formatting issues.
- Custom build steps if running Terraform in generic CI/CD tool.
- Steep learning curve for engineers new to Terraform.

Cultivating a high-velocity platform team!

The migration

- Matching up CloudFormation resources to Terraform.
- Referring to Hashicorp documentation when needed.
- Testing plans.

Terraform Import block

- Command released last year (TF v1.5.0) to successfully import Terraform resources in-code.

- Alternative to CLI import command.

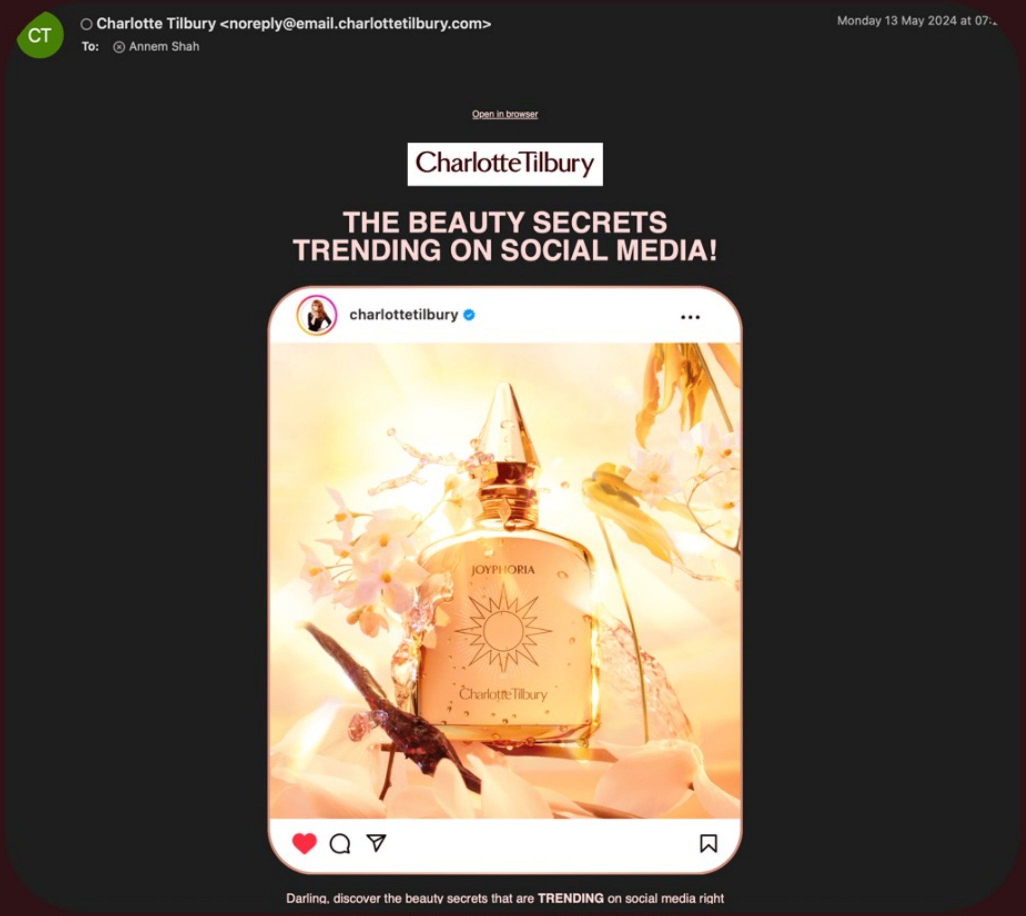
- Resources that already exist – we do not want to delete them, or their metadata or stored items.

Case Study 101: Assets S3 bucket

- Small, very impactful repository.

- Defined S3 bucket, and CORS configuration.

- Pulled fonts, images and thumbnails into customer emails.



CT ○ Charlotte Tilbury <noreply@email.charlottetilbury.com>

To: Annem Shah

Monday 13 May 2024 at 07:

Open in browser

CharlotteTilbury

THE BEAUTY SECRETS
TRENDING ON SOCIAL MEDIA!

charlottetilbury

Darling, discover the beauty secrets that are **TRENDING** on social media right

Case Study 101: Assets S3 bucket

- Defined the S3 bucket in a Terraform module.
- Added environment-specific variables in HCP Terraform.
- Called the module and imports in the main.tf configuration.
- Could see what we are planning on importing before adding new resources – CORS config.

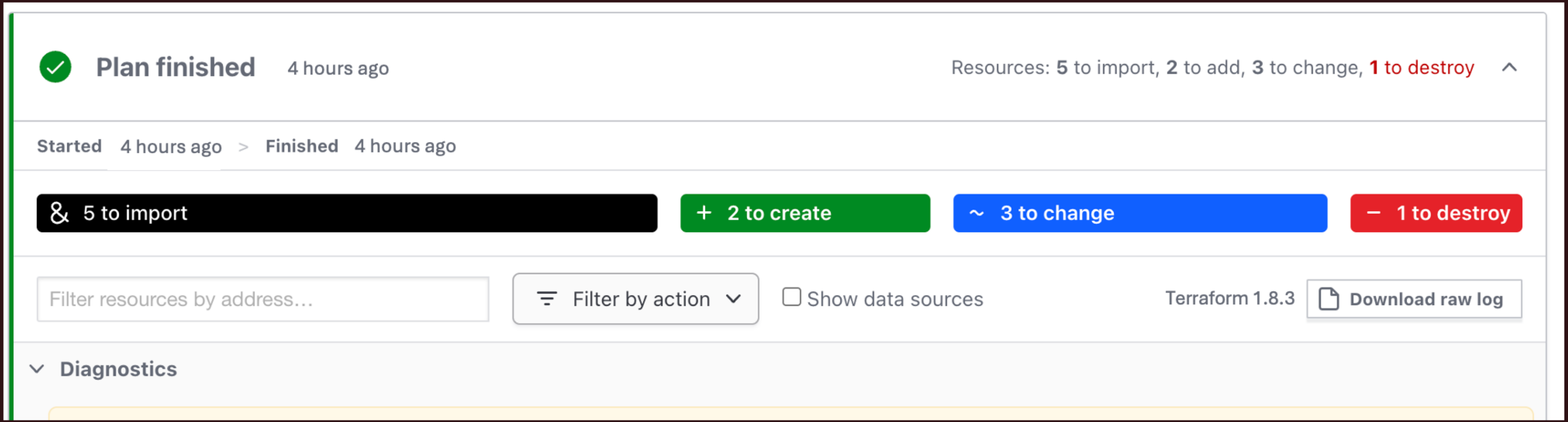
Case Study 101: Assets S3 bucket

```
module "shared-assets" {
  source.           = "./modules/shared-assets"
  account_id        = var.account_id
  cloudfront_oai_id = var.cloudfront_oai_id
  hosted_zone_id.   = var.hosted_zone_id
  environment       = var.environment_name
}

data "aws_s3_bucket" "shared_assets_bucket" {
  bucket = ”prod-shared-assets"
}

import {
  to = module.shared_assets.aws_s3_bucket.shared_assets_bucket
  id = data.aws_s3_bucket.shared_assets_bucket.id
}
```

Success! Importing



✓ **Plan finished** 4 hours ago

Resources: **5** to import, **2** to add, **3** to change, **1** to destroy

Started 4 hours ago > **Finished** 4 hours ago

Tying up any loops

- Decommission the CloudFormation stack.
- Check over all the resources thoroughly, including policies.
- Easier to rollback changes now rather than later.

Pitfalls

- IDs for import block vary by resource.
- Ensure like-for-like in the import plan.
- Remember to add deletion policies when decommissioning CloudFormation!

Key takeaways

- AWS CloudFormation generally works well enough in single-CSP environments, but it has flaws.

- The importance of separating infrastructure and application code deployment meant migration was the next natural step for us.

- HCP Terraform helps simplify this in a large team with varied levels of expertise, allowing us to focus on scaling our platform.

Wrapping up

- Find me at the networking reception.
- Prize winner.
- Let’s connect!

SCAN ME

Charlotte Tilbury