Improving the Developer Experience of Infrastructure as Code and GitOps

FOSDEM Feb 7, 2021
Agenda

1. DX Gap
2. Demo
3. Code
About me

Philipp Strube

Twitter @pst418
K8s Slack @pst in #kubestack
DX Gap
DX Gap of IaC

1. Upfront effort
2. Maintenance effort
3. Slow feedback loops
## Use-case specific frameworks

<table>
<thead>
<tr>
<th>Frameworks</th>
</tr>
</thead>
<tbody>
<tr>
<td>django</td>
</tr>
<tr>
<td>RAILS</td>
</tr>
<tr>
<td>spring</td>
</tr>
<tr>
<td>Gatsby</td>
</tr>
<tr>
<td>KUBESTACK</td>
</tr>
<tr>
<td>Python</td>
</tr>
<tr>
<td>Java</td>
</tr>
<tr>
<td>JS</td>
</tr>
</tbody>
</table>
The Paradigm-Shift that enables the **Kubestack framework**

- **Application Layer**
  - App
  - App
  - App

- **Platform Layer**
  - weak abstraction
  - OS
  - OS
  - OS
  - Infrastructure

- **Platform Layer**
  - powerful abstraction
  - Containers & Kubernetes
  - OS
  - Infrastructure
Environments
Workspaces as Environments

```hcl
resource "google_container_cluster" "example1" {
  name = var.name

  # ...
}

resource "google_container_cluster" "example2" {
  name = format("%s-%s", var.name, terraform.workspace)

  # ...
}
```
Metadata module

```
module "cluster_metadata" {
  source = "../../common/metadata"

  name_prefix = local.name_prefix
  base_domain = local.base_domain

  provider_name = "aws"
  provider_region = data.aws_region.current.name
}
```
Configuration
Environment Configuration

```hcl
config = {
  workspace1 = {
    attr1 = "value1"
    attr2 = "value1"
  }
  workspace2 = {
    attr1 = "value1"
    attr2 = "value2"
  }
}

module "example" {
  attr1 = var.config[terraform.workspace].attr1
  attr2 = var.config[terraform.workspace].attr2
}
```
Inheritance module

gke_zero = {
  apps = {
    # ...
    cluster_min_node_count = 1
    cluster_max_node_count = 10
    region = "europe-west1"
    cluster_node_locations = "europe-west1-b,europe-west1-c, europe-west1-d"
  };
  ops = {
    cluster_max_node_count = 1
    cluster_node_locations = "europe-west1-b,europe-west1-c"
  }
}
GitOps Flow
GitOps Flow

1. Change
2. Review
3. Validate
4. Promote
Pipeline runtime & resiliency

● Docker images
  ○ Multi-cloud: `kubestack/framework:$version`
  ○ AKS only: `kubestack/framework:$versions-aks`
  ○ EKS only: `kubestack/framework:$versions-eks`
  ○ GKE only: `kubestack/framework:$versions-gke`
  ○ KIND only: `kubestack/framework:$versions-kind`

● Extend the Dockerfile
  ○ `FROM kubestack/framework:v0.12.1-beta.0-aks`

● Vendoring Terraform providers

● Exception: Terraform modules are downloaded from Github
Scope
Committed, desired, and current state
Cluster Services
Cluster Services

1. Validated in ops, before promoted to apps

2. Unify application environments in multi-cluster or multi-cloud use-case
● Terraform GitOps Framework
  ○ The best DX for IaC
  ○ And rock-solid automation
Thank you for your attention!

1. Tutorial on: www.kubestack.com
2. Code at: github.com/kbst/terraform-kubestack
3. Join the conversation: #kubestack on the K8s Slack