Hands-on Gitops

Weaveworks – [https://weave.works](https://weave.works) – @weaveworks
Kubecon Seattle – December 2018

Brice Fernandes – brice@weave.works – @fractallambda
Hi

I’m Brice

I work for Weaveworks as a customer success engineer

You can find Weaveworks at https://www.weave.works or @weaveworks

The team at Weaveworks is behind the GitOps model

You can find me online at @fractallambda and email me at brice@weave.works
Agenda today

1. Gitops principles (Presentation)
2. Hands on gitops (Exercises)
3. Gitops on Weave Cloud (Demo)
Everything available at

tinyurl.com/gitops-tutorial
What is Gitops?
GitOps is...

An operation model

Derived from CS and operation knowledge

Technology agnostic (name notwithstanding)

A set of principles (Why instead of How)

A way to speed up your team
1. The entire system is described declaratively.

2. The canonical desired system state is versioned (with Git)

3. Approved changes to the desired state are automatically applied to the system

4. Software agents ensure correctness and alert on divergence
The entire system is described declaratively.
The entire system is described declaratively.

Beyond code, data ⇒

- Implementation independent
- Easy to abstract in simple ways
- Easy to validate for correctness
- Easy to generate & manipulate from code
The entire system is described declaratively.

Beyond code, data ⇒

Implementation independent

Easy to abstract in simple ways

Easy to validate for correctness

Easy to generate & manipulate from code
How is that different from Infrastructure as code?
How is that different from Infrastructure as code?

It’s about consistency in the failure case.
It’s about consistency in the failure case.

When imperative systems fail, the system ends up in an unknown, inconsistent state.
When imperative systems fail, the system ends up in an unknown, inconsistent state.

Declarative changes let you think of changes as transactions.
Declarative changes let you think of changes as transactions.

This is a very good thing.
The canonical desired system state is versioned (with Git)
The canonical desired system state is versioned (with Git)

Canonical Source of Truth (DRY)

With declarative definition, trivialises rollbacks

Excellent security guarantees for auditing

Sophisticated approval processes (& existing workflows)

Great Software ↔ Human collaboration point
Changes to the desired state are automatically applied to the system.
Approved changes to the desired state are automatically applied to the system

**Significant velocity gains**

Privileged operators don’t cross security boundaries

Separates **What** and **How**.
Software agents ensure correctness and alert on divergence
Software agents ensure correctness and alert on divergence

Continuously checking that desired state is met

System can self heal

Recovers from errors without intervention (PEBKAC)

It’s the control loop for your operations
The entire system is described declaratively.

The canonical desired system state is versioned (with Git)

Approved changes to the desired state are automatically applied to the system

Software agents ensure correctness and alert on divergence
Typical CICD pipeline

Shares credentials cross several logical security boundaries.

- Git creds
- CI creds
- Container Registry (CR) creds
- CR creds
- API creds

Dev RW → Code Repo RW → CI RO RW → Container Registry RO → Cluster API

CR creds^1

Continuous Integration Boundary Continuous Delivery/Deployment
GitOps pipeline

Credentials are never shared across a logical security boundary.

Canonical desired state store
GitOps pipeline

Credentials are never shared across a logical security boundary.
GitOps pipeline

Credentials are never shared across a logical security boundary.

Process & constraints enforcement
GitOps pipeline

Credentials are never shared across a logical security boundary.

Exceptional auditing and attribution
Gitops is Functional Reactive Programming…

…for your infrastructure.

Like React, but for servers and applications.
What should be GitOps’ed?
What should be GitOps’ed?

I’m so very sorry
# Rules for alerting on stuff related to corporate IT.

#

# All of these alerts must have 'team = "corp"' in their labels. This routes
# them to the #corporate-it channel and away from the production on-call.

# Alert when our terradiff for corp doesn't match the environment.

ALERT CorpTerraDiff
IF max(terradiff_plan_exit_code) == 2
FOR 1h
LABELS {
  severity = "warning",
  team = "corp",
}

ANNOTATIONS {
  summary = "weaveworks/corp terraform config differs from reality",
  impact = "Someone has lost access to something, or someone has access without approval",
  dashboardURL = "${base_url}/admin/corp-terradiff",
  containerName = "terradiff",
}
resource "github_membership" "bricef" {
  username = "bricef"
  role = "admin"
}
AlertManager

CorpTerradiff - weaveworks/corp terraform config differs from reality
Someone has lost access to something, or someone has access without approval
Dashboard
Gitops Hands-on
Everything available at tinyurl.com/gitops-tutorial
Installing and using OSS Flux
<table>
<thead>
<tr>
<th>Workload</th>
<th>Image</th>
<th>Current Tag</th>
<th>Behind</th>
<th>Latest Tag</th>
<th>Update</th>
</tr>
</thead>
<tbody>
<tr>
<td>loadtest:deployment/load-test</td>
<td>weaveworks demos/load-test</td>
<td>master-57a...</td>
<td>5mo</td>
<td></td>
<td>UP TO DATE</td>
</tr>
<tr>
<td>sock-shop:deployment/carts</td>
<td>weaveworks demos/carts</td>
<td>master-1cf...</td>
<td>5mo</td>
<td></td>
<td>UP TO DATE</td>
</tr>
<tr>
<td>sock-shop:deployment/carts-db</td>
<td>mongo</td>
<td>No tag specified</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>sock-shop:deployment/catalogue</td>
<td>weaveworks demos/catalogue</td>
<td>master-bf5...</td>
<td>5mo</td>
<td></td>
<td>UP TO DATE</td>
</tr>
<tr>
<td>sock-shop:deployment/catalogue-db</td>
<td>weaveworks demos/catalogue-db</td>
<td>master-bf5...</td>
<td>5mo</td>
<td></td>
<td>UP TO DATE</td>
</tr>
<tr>
<td>sock-shop:deployment/front-end</td>
<td>weaveworks demos/front-end</td>
<td>master-a34...</td>
<td>6h</td>
<td>Diverged</td>
<td></td>
</tr>
<tr>
<td>sock-shop:deployment/orders</td>
<td>weaveworks demos/orders</td>
<td>master-f89...</td>
<td>5mo</td>
<td></td>
<td></td>
</tr>
<tr>
<td>sock-shop:deployment/orders-db</td>
<td>mongo</td>
<td>No tag specified</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>sock-shop:deployment/payment</td>
<td>weaveworks demos/payment</td>
<td>master-b64...</td>
<td>4mo</td>
<td></td>
<td></td>
</tr>
<tr>
<td>sock-shop:deployment/queue-master</td>
<td>weaveworks demos/queue-master</td>
<td>master-b89...</td>
<td>5mo</td>
<td></td>
<td></td>
</tr>
<tr>
<td>sock-shop:deployment/rabbitmq</td>
<td>rabbitmq</td>
<td>3.7-rc-man...</td>
<td>2mo</td>
<td>23 versions</td>
<td></td>
</tr>
<tr>
<td>sock-shop:deployment/session-db</td>
<td>redis</td>
<td>32bit</td>
<td>1mo</td>
<td>4 versions</td>
<td></td>
</tr>
<tr>
<td>sock-shop:deployment/shipping</td>
<td>weaveworks demos/shipping</td>
<td>master-ba1...</td>
<td>5mo</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
History

- Manual deploy: weaveworks/demos/front-end:master-af3b817e to sock-shop:deployment/front-end
  23 minutes ago

- Sync to cluster: sock-shop:deployment/front-end (20c187340be)
  23 minutes ago

- Commit: sock-shop:deployment/front-end (20c187340be)
  by Matthias <matthias@weave.works>
  23 minutes ago

- Sync to cluster: sock-shop:deployment/catalogue (f68ba08a507)
  4 days ago

- Unlock: sock-shop:deployment/catalogue (f68ba08a507)
  Demo
  by Brice <brice@weave.works>
  4 days ago

- Sync to cluster: sock-shop:deployment/catalogue (2fd23dd6bd93)
  4 days ago

- Lock: sock-shop:deployment/catalogue (2fd23dd6bd93)
  Demo
  by Brice <brice@weave.works>
  4 days ago

  4 days ago
KEEP CALM IT'S DEMO TIME

Using Weave Cloud
Questions?

Weaveworks
https://weave.works
@weaveworks
@fractallambda
brice@weave.works

go.weave.works/extended

45 day trial

tinyurl.com/gitops-tutorial