Managing EKS at Scale with GitOps

Alexis Richardson, CEO Weaveworks
@monadic
alexis@weave.works
http://weave.works

AWS Container Days - November 2019
About Weaveworks

- Founded in 2014, backed by Google Ventures & Accel Partners
- Team of industry leaders from multiple projects
- Mission: accelerate cloud native platform adoption in the enterprise, through our Weave Kubernetes Platform and open source projects.
Genesis of GitOps

“GitOps” is our name for how we use developer tooling to drive operations. Git is a part of every developer’s toolkit. Using the practices outlined in this post, our developers operate Kubernetes via Git. We manage and monitor all of our applications and the whole ‘cloud native stack’ using GitOps.

Excerpt from original blog post – GitOps Operations by Pull Request, August 2017
GitOps is born

The scene: spring 2016, a peaceful morning in London.
The sun is shining. Birds tweet.
“I’m about to make a change that will probably wipe out all our systems”
“Tom, are you sure we want to do that?”
<click>
“Oops - I’ve just deleted all our Kubernetes clusters on AWS”

CHAOS ENSUES
We use declarative infrastructure ie. Kubernetes, Docker, Terraform, & more

Our entire system including code, config, monitoring rules, dashboards, is described in GitHub with full audit trail

We can roll out major or minor changes as atomic pull requests, and automatically converge then check for diffs if system diverges from the “source of truth” in Git

How Weaveworks rebuilt systems in 45 mins?

Now 5 mins or less
Flux joins CNCF

“[you] made the single source of truth possible in git, and it was so much more clear”

Kyle Rockman, Under Armour
Argo & Flux merging in CNCF

- What does this merger mean?
- Ability to standardize on GitOps
- First joint project - Argo Flux GitOps Engine
Future

- Extending GitOps to progressive delivery
- Canary, Blue/Green and other controlled deployments
- Experiments under policy
**GitOps @ Weaveworks Timeline**

- 4 part blog post “GitOps - Operations by Pull Request” published
- Flux is developed and released as first GitOps Operator
- InfoQ and Techcrunch publish GitOps articles
- Flux is accepted as a project in CNCF
- Flagger is released for GitOps and Progressive delivery
- Profiles and JKcfg for GitOps platforms and templates

Reproducible Automated Deployment — Policy & Compliance — Progressive Delivery — Alert & Fix Failures Automatically
Can GitOps be a “complete solution”?

Git as the single source of truth of a system’s desired state, enabling reproducible automated deployment, management and drift alerts

GitOps Diffs compare desired state with observed state continuously

… What about CLUSTERS
GitOps gives us a developer workflow – what can we do with it
Give me the stack I want

Booting Kubernetes clusters is easy. Managing a complete application platform is hard.

“Give me an ML stack”

● Make it safe, consistent, auditable, upgradeable

● Can you answer these three questions about your Stack?
  ○ Do you know if it is in the correct state?
  ○ Can you reproduce your entire app + cluster stack at will?
  ○ Can you cleanly upgrade the whole stack?
The Cloud-Native App Delivery Problem

Customers need to deliver **MANY APPs** for **MANY USE CASES**. Machine Learning is just one of them.

- Figure out what an ‘app’ means
- Figure out what ‘platform add-ons’ you need for your use case, e.g. machine learning
- **Deploy apps and add-ons correctly** and properly wired up to other services, e.g. RDS, Lambda
- **Verify** cluster, test add-ons and apps are configured correctly for dev, staging to prod…
- **Operate**, shut down, reproduce stack (think “cattle” not “pets”)
- **Manage**, maintain, upgrade, patch any part of this according to POLICY
- **Scale** to fleets of clusters, across many use cases and cloud options

*We want easily reproducible cattle and not pets or (worse) snowflakes*
Reproducible clusters

• Consistent Kubernetes App platforms with Argo Flux CD

• A single git repository can contain:
  • Cluster definition (eg. K8s version, IP range for pods, CRI, CNI, CSI)
  • List of nodes
  • Core apps used by your team eg. Machine Learning needs MLOps
  • Additional custom apps specific to your team (eg. Authentication, CICD, Monitoring, etc.)

• Cluster creation can be driven completely via GitOps with pull requests
Build any stack based on a MODEL

GitOps Managed

Multi-cloud deployments

Platform Model

Applications

git commit –m “app” && git push

Model Profiles

Web
MLOps
Buildpack

EKSctl
WKSctl

kubernetes

Amazon EKS

Firekube
Reproducible platforms on EKS

Cluster repo → Flux - GitOps Operator → API Server → etcd

Platform Configuration

sync → create/delete → store

watch

EKS Controller

Kind: EKS Cluster

Manages (ssh)

Amazon EKS
Reproducible platforms and OSS K8s
Declarative Clusters

- Example: Advanced cluster architecture with two AWS auto-scaling groups (nodegroups)

```yaml
apiVersion: eksctl.io/v1alpha5
kind: ClusterConfig
metadata:
  name: jpmc-demo-cluster-1
  region: eu-west-1
  version: "1.13"
nodeGroups:
  - name: applications
    labels: {applications-only: "true"}
    minSize: 5
    maxSize: 15
    instanceType: m5.xlarge
  - name: management
    labels: {management-only: "true"}
    taint: {management-only: "true:NoSchedule"}
    iam:
      attachPolicyARNs:
        - arn:aws:iam::123:policy/ng-management-1
      withAddonPolicies:
        albIngress: true
        externalDNS: true
    desiredCapacity: 2
    instanceType: m5.large
    securityGroups:
      attachIDs: [sg-0b44c48bcba5b7362]
```
Consistent toolchains across environments

- EKSctl profiles for consistent and complete clusters
- Takes a cluster, adds Flux to the cluster and links it to a git repository
- Adds the “app-dev” profile config to your repository
- Flux watches the repository and deploys the workloads
eksctl enable profile

Amazon EKS

corner

--git-url

profile manifests

gitops repo

<profile-url or name>

profile

cluster

flux

flux

HELM

HELM
Build any stack based on a MODEL

Multi-cloud deployments

GitOps Managed

Platform Model

Applications

git commit –m “app” && git push

Model Profiles

Web
MLOps
Buildpack

Amazon EKS

Firekube

EKSctl

WKSctl

kubernetes

Buildpack

MLOps

Web
Weave: Automated Reproducible Clusters

Automated Reproducible GitOps Experience Dev to Prod

Weaveworks GitOps Manager
- Progressive App Delivery
  - Flagger
- GitOps Policy & Compliance
  - OPA, JK
- GitOps Dashboard
  - Weave Cloud

Weaveworks Kubernetes Platform
- Application Quickstarts
  - (ML, Web App, Mobile, Big Data, …)
- GitOps Continuous Delivery
  - Flux
- GitOps APIs for K8s Provisioning & Config Mgmt
  - wksctl, eksctl, aksctl, other ctls

GitOps Policy & Compliance
- OPA, JK
- Progressive App Delivery
- Weave Cloud

Public Cloud, vSphere, Firecracker, On-Metal, Air-Gapped, Edge
Cluster Dashboard
# Cluster Profile Definitions

<table>
<thead>
<tr>
<th>Name</th>
<th>K8s v.</th>
<th>Node groups</th>
<th>Components &amp; apps</th>
<th>Last change</th>
<th>Last modified by</th>
<th>Last modified by email</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Production cluster</strong></td>
<td>1.15.0</td>
<td>10x m3.large</td>
<td>Prometheus, metrics server, fluentd, Elasticsearch, Istio, Jaeger, Helm, Tiller, wkp-external-dns, Vault</td>
<td>12:45 9/10/2019</td>
<td><a href="mailto:john.smith@acmecorp.com">john.smith@acmecorp.com</a></td>
<td><a href="mailto:john.smith@acmecorp.com">john.smith@acmecorp.com</a></td>
</tr>
<tr>
<td><strong>Machine learning cluster</strong></td>
<td>1.14.0</td>
<td>5x m3.large</td>
<td>Kubeflow, Knative, Prometheus, metrics server, fluentd, Elasticsearch, Istio, Jaeger, Helm, Tiller, wkp-external-dns</td>
<td>12:45 9/10/2019</td>
<td><a href="mailto:john.smith@acmecorp.com">john.smith@acmecorp.com</a></td>
<td><a href="mailto:john.smith@acmecorp.com">john.smith@acmecorp.com</a></td>
</tr>
<tr>
<td><strong>Search components</strong></td>
<td>1.15.0</td>
<td>5x m3.large</td>
<td>Elasticsearch-cluster-search, search-ui, search-api, redis</td>
<td>12:45 9/10/2019</td>
<td><a href="mailto:john.smith@acmecorp.com">john.smith@acmecorp.com</a></td>
<td><a href="mailto:john.smith@acmecorp.com">john.smith@acmecorp.com</a></td>
</tr>
</tbody>
</table>
Alerting and Events Dashboard across clusters

**Firing alerts**

<table>
<thead>
<tr>
<th>Monitor Alert</th>
<th>Production Cluster</th>
</tr>
</thead>
<tbody>
<tr>
<td>2019-11-06T18:48:834Z</td>
<td>(NodeHighCPUUsage - WARNING FIRING) CPU usage has been over 10% for 5m</td>
</tr>
<tr>
<td>Node: gke-sock-shop-default-pool-9652982b-1l7z</td>
<td></td>
</tr>
<tr>
<td>Node: gke-sock-shop-default-pool-9652982b-7p5s</td>
<td></td>
</tr>
<tr>
<td>Node: gke-sock-shop-default-pool-9652982b-rvffn</td>
<td></td>
</tr>
<tr>
<td>containerName: prom-node-exporter</td>
<td></td>
</tr>
</tbody>
</table>

**Events**

**Drift detected**

<table>
<thead>
<tr>
<th>Machine Learning Cluster</th>
<th>Production Cluster</th>
</tr>
</thead>
<tbody>
<tr>
<td>2019-11-06T00:06:234Z</td>
<td>The definition is 6 nodes but berlin-ai is running with 5.</td>
</tr>
</tbody>
</table>

**Policy Check Failed**

<table>
<thead>
<tr>
<th>Production Cluster</th>
</tr>
</thead>
<tbody>
<tr>
<td>2019-11-06T09:34:23.584Z</td>
</tr>
<tr>
<td>commit ca57f6f</td>
</tr>
<tr>
<td>hello-deployment.yaml -- Critical production deployment: 'hello-world my not have fewer than two replicas</td>
</tr>
</tbody>
</table>
We can do Progressive Delivery this way too

Weave & Argo & AWS Enable Automated Deployments & Progressive Delivery
Progressive Delivery Toolset

Applications

Flagger

App Mesh Prometheus

App Mesh Controller

Flux GitOps Operator

Amazon EKS

App Mesh Grafana

App Mesh Injector

HELM

Flux Helm Operator

> git push

> eksctl enable profile appmesh

> eksctl enable repo

> eksctl create cluster
Visit our booth #S51 to meet the team, grab some swag and learn more about **automating Kubernetes with GitOps**!

Try our GitOps Hands-On for a prize.

www.weave.works