

Why GitOps Matters for your business success



Steve George
COO, Weaveworks
@futurile • steveg@weave.works

Thesis

1. GitOps supports the DevOps approach in a highly effective way
2. GitOps brings DevOps to Cloud Native software operations
3. It can significantly improve the way we operate software



What is GitOps?

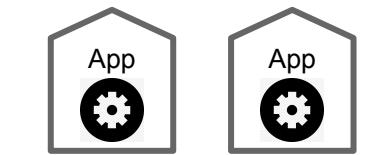
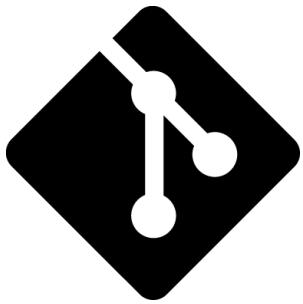




Created by Oksana Lolykina
From Near Project



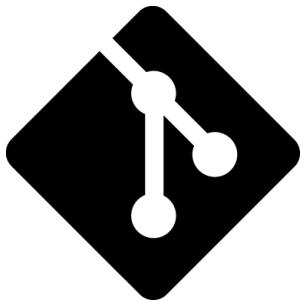
Created by Oksana Lolykina
From Near Project



Runtime Env



Desired State
Store all code and configuration



Automation



Created by SBTG
from Neur Project

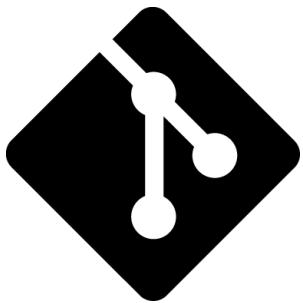
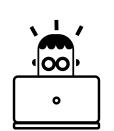
Actual State
Runtime environment



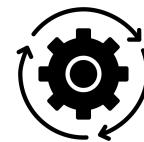
Runtime Env



Desired State
Store all code and configuration



Automation



Actual State
Runtime environment



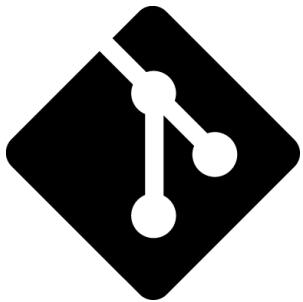
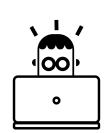
Runtime Env



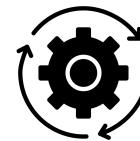
Declarative
Convergent loop



Desired State
Store all code and configuration



Automation



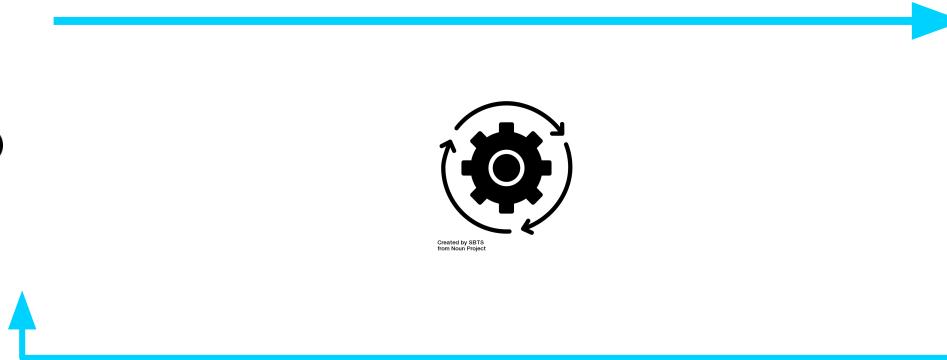
Actual State
Runtime environment



Runtime Env



Single interface
to operations



CNCF - Principles of GitOps



The entire system
is described
declaratively

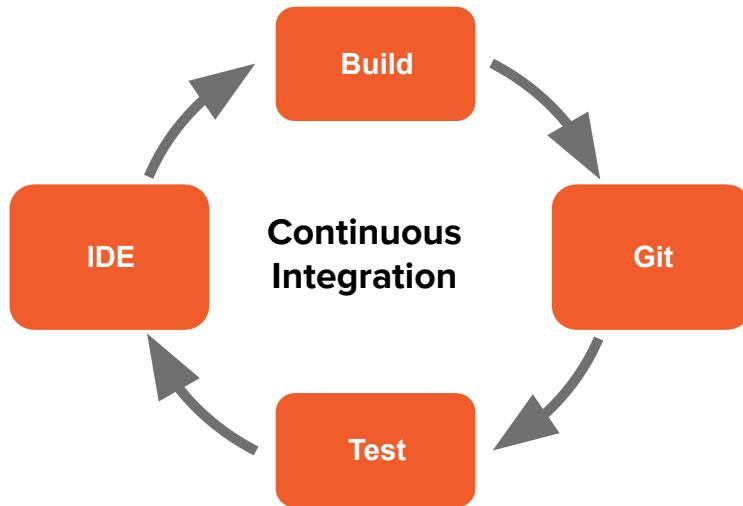
The canonical
desired system
state is **versioned**
in git

Approved
changes can be
**automatically
applied**
to the system

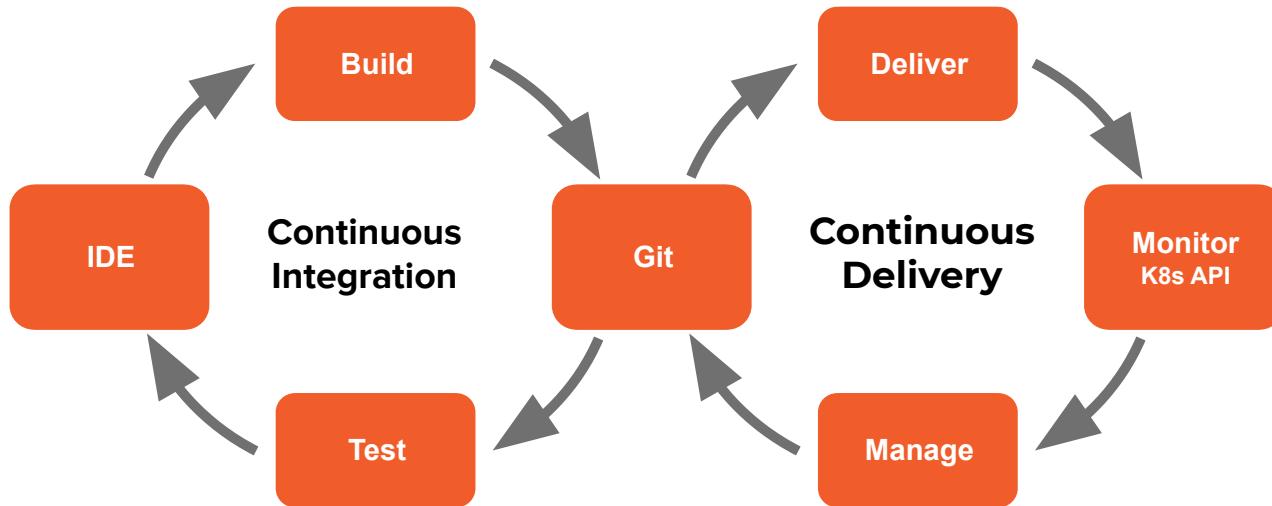
Software agents
ensure
correctness and
alert (diffs &
actions)

Actions are performed
when the run time
state, diverges from
the state in git,
creating a
closed loop system

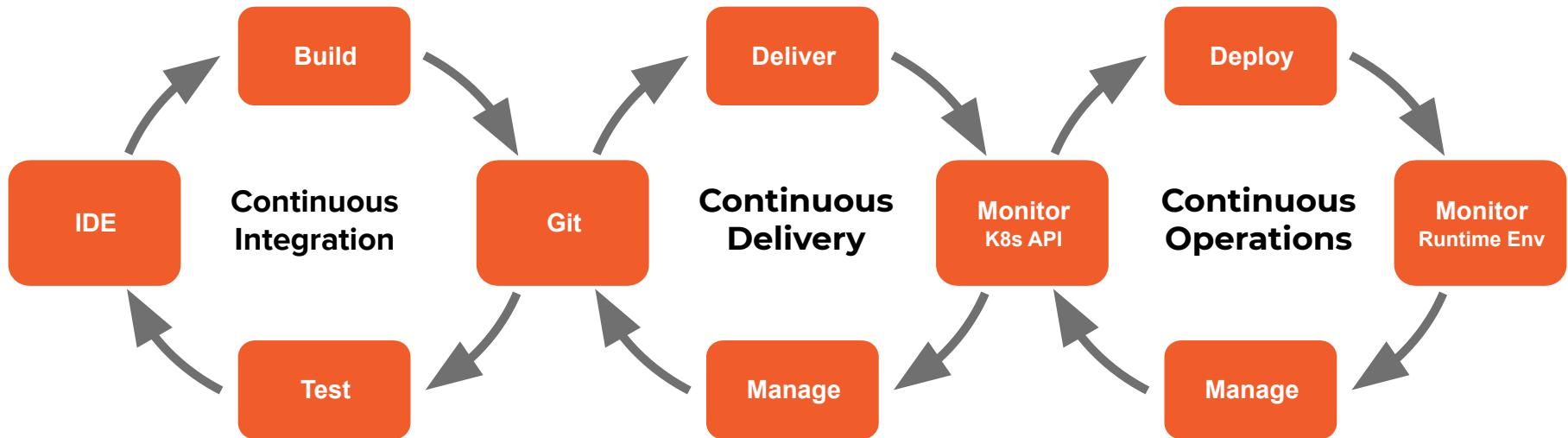
GitOps - Extending how we operate



GitOps - Extending how we operate



GitOps - Extending how we operate



Get better at doing software



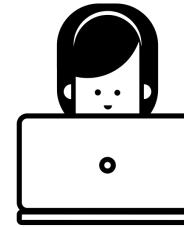
Aspect of Software Delivery Performance*	Elite	High	Medium	Low
Deployment frequency For the primary application or service you work on, how often does your organization deploy code to production or release it to end users?	On-demand (multiple deploys per day)	Between once per day and once per week	Between once per week and once per month	Between once per month and once every six months
Lead time for changes For the primary application or service you work on, what is your lead time for changes (i.e., how long does it take to go from code committed to code successfully running in production)?	Less than one day	Between one day and one week	Between one week and one month	Between one month and six months
Time to restore service For the primary application or service you work on, how long does it generally take to restore service when a service incident or a defect that impacts users occurs (e.g., unplanned outage or service impairment)?	Less than one hour	Less than one day ^a	Less than one day ^a	Between one week and one month
Change failure rate For the primary application or service you work on, what percentage of changes to production or released to users result in degraded service (e.g., lead to service impairment or service outage) and subsequently require remediation (e.g., require a hotfix, rollback, fix forward, patch)?	0-15% ^{b,c}	0-15% ^{b,d}	0-15% ^{c,d}	46-60%

<https://cloud.google.com/blog/products/devops-sre/the-2019-accelerate-state-of-devops-elite-performance-productivity-and-scaling>



- Enable application (DevOps) teams to release more frequently, reduce lead time & operate cloud native applications more effectively

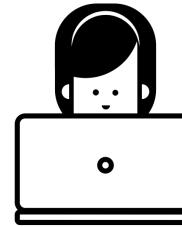
Familiar Tooling Enables Self-Service



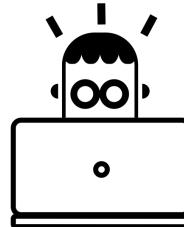
Created by Oksana Latysheva
from Noun Project

- Enable application (DevOps) teams to release more frequently, reduce lead time & operate cloud native applications more effectively

Familiar Tooling Enables Self-Service



Created by Oksana Latysheva
from Noun Project



Created by Oksana Latysheva
from Noun Project

- Enable Platform teams to maintain reliability, security, compliance and cost management

Enables Resilience Security and compliance built-in

Aspect of Software Delivery Performance*	Elite	High	Medium	Low
Deployment frequency For the primary application or service you work on, how often does your organization deploy code to production or release it to end users?	On-demand (multiple deploys per day)	Between once per day and once per week	Between once per week and once per month	Between once per month and once every six months
Lead time for changes For the primary application or service you work on, what is your lead time for changes (i.e., how long does it take to go from code committed to code successfully running in production)?	Less than one day	Between one day and one week	Between one week and one month	Between one month and six months
Time to restore service For the primary application or service you work on, how long does it generally take to restore service when a service incident or a defect that impacts users occurs (e.g., unplanned outage or service impairment)?	Less than one hour	Less than one day ^a	Less than one day ^a	Between one week and one month
Change failure rate For the primary application or service you work on, what percentage of changes to production or released to users result in degraded service (e.g., lead to service impairment or service outage) and subsequently require remediation (e.g., require a hotfix, rollback, fix forward, patch)?	0-15% ^{b,c}	0-15% ^{b,d}	0-15% ^{c,d}	46-60%

<https://cloud.google.com/blog/products/devops-sre/the-2019-accelerate-state-of-devops-elite-performance-productivity-and-scaling>



Familiar Tools

Fix permissions in kube-addon-manager #91261

Merged k8s-ci-bot merged 1 commit into kubernetes:master from tosi3k:an-patch-1 20 hours ago

Conversation 4 Commits 1 Checks 0 Files changed 2 +5 -2

tosi3k commented yesterday

What type of PR is this?
/kind bug

What this PR does / why we need it:
Last built `kube-addon-manager` image (v9.1.0) had wrong permissions set for both `kubectl` and `kube-addons.sh` files preventing updating the manifests in #91240 because AddonManager container wasn't able to execute them.

I tested the newly built image (v9.1.1) using the presubmit tests and it works.

Which issue(s) this PR fixes:
No issue is opened for that.

Special notes for your reviewer:
/sig scalability
/assign wojtek-t

Does this PR introduce a user-facing change?:
NONE

Additional documentation e.g., KEPs (Kubernetes Enhancement Proposals), usage docs, etc.:
`Fix permissions in kube-addon-manager` ✓ 3c5585b

k8s-ci-bot assigned wojtek-t yesterday

k8s-ci-bot added `release-note-none` `kind/bug` `area/provider/gcp` `sig/scalability` `cncf-cla: yes` `needs-priority` labels yesterday

k8s-ci-bot requested review from cheftako and MrHohn yesterday

jtek-t commented yesterday

im
iprove

k8s-ci-bot added the `lgtm` label yesterday

Reviewers: cheftako, MrHohn

Assignees: wojtek-t

Labels: approved, area/provider/gcp, cncf-cla: yes, kind/bug, lgtm, needs-priority, release-note-none, sig/cloud-provider, sig/scalability, sig/vx

Projects: None yet

Milestone: v1.19

Linked issues: Successfully merging this pull request may close these issues.
None yet

Notifications: Customize
Subscribe
You're not receiving notifications from this thread.

Participants: 4 participants

1 similar comment

k8s-ci-bot merged commit `cf13f8d` into kubernetes:master 20 hours ago
18 checks passed

k8s-ci-bot added this to the v1.19 milestone 20 hours ago

tosi3k mentioned this pull request 18 hours ago
Update kube-addon-manager to v9.1.1 #91240



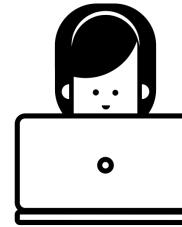
Self Service Ops

(not self-service infra)

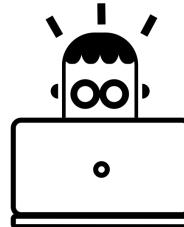


- Enable application (DevOps) teams to release more frequently, reduce lead time & operate cloud native applications more effectively

Familiar Tooling
Enables Self-Service



Created by Oksana Latysheva
from Noun Project



Created by Oksana Latysheva
from Noun Project

- Enable Platform teams to maintain reliability, security, compliance and cost management

Enables Resilience
Security and compliance built-in

Resilience through Git Semantics

versioned

```
less
commit 119c3905b3c0a975e313556f7b2049492a8252ac (HEAD -> master, origin/master, origin/HEAD)
Merge: 5433c28d 7377aae0
Author: Hidde Beydals <hiddeco@users.noreply.github.com>
Date: Mon May 18 21:00:34 2020 +0200

Merge pull request #3060 from circa10a/docs/helm-operator-integration

commit 7377aae0c1e3e01021c818a1d4dd8b8487e51b5e
Author: Caleb Lemoine <caleblemoine@gmail.com>
Date: Fri May 15 19:48:40 2020 -0500

docs: update helm operator integration glob patterns

commit 5433c28dd96585ee9db160f9ea4068ee4ed7b713
Merge: f6b18042 83c084e3
Author: Hidde Beydals <hiddeco@users.noreply.github.com>
Date: Wed May 13 12:42:36 2020 +0200

Merge pull request #3053 from maruina/secret-annotations

commit 83c084e3e5ddb99db6b81e1e932ebcf15379f150b
Author: Matteo Ruina <matteo.ruina@skyscanner.net>
Date: Mon May 11 16:38:33 2020 +0200

Support annotations in secret

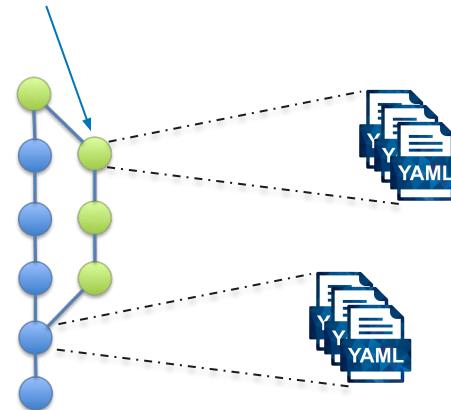
Do not bump chart

Fix value used

commit f6b1804221699e38af85ef6ff5da1474d4317c83
Merge: 0ad6a83e e969cd71
Author: Hidde Beydals <hiddeco@users.noreply.github.com>
Date: Tue May 12 17:50:37 2020 +0200

Merge pull request #3008 from fluxcd/non-preferred-resources
```

each version is a complete representation



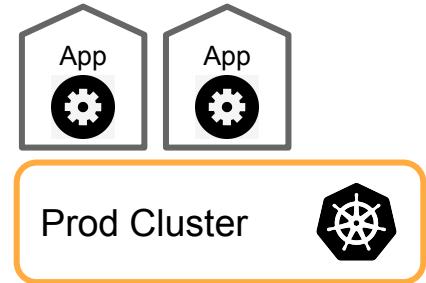
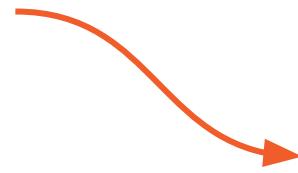
immutable





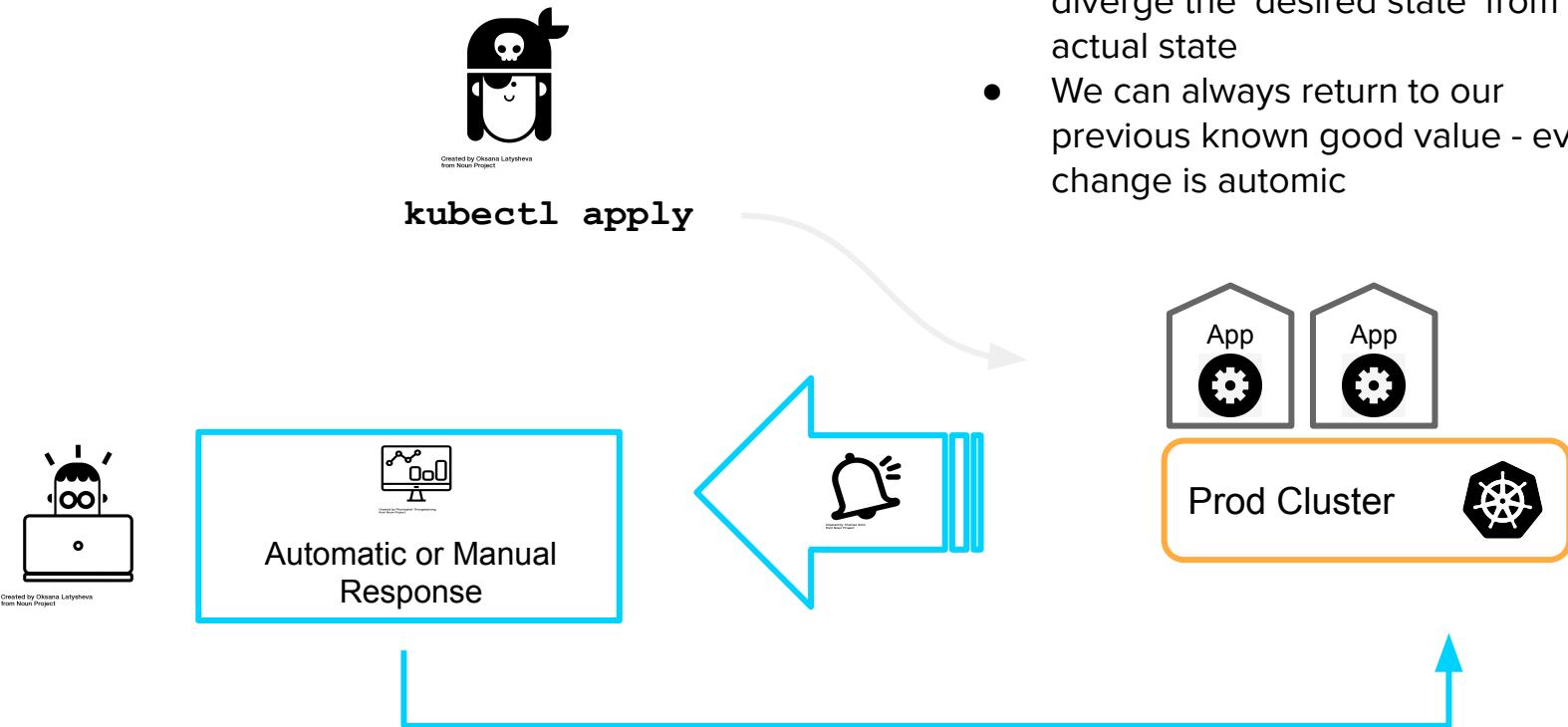
Created by Oksana Latyshova
from Noun Project

kubectl apply



Revert Any change

- Any change to the system will diverge the ‘desired state’ from the actual state
 - We can always return to our previous known good value - every change is automic



Compliance ❤️ GitOps

```
less
commit 119c3905b3c0a975e313556f7b2049492a8252ac (HEAD -> master, origin/master, origin/HEAD)
Merge: 5433c28d 7377aae0
Author: Hidde Beydals <hiddeco@users.noreply.github.com>
Date: Mon May 18 21:00:34 2020 +0200

Merge pull request #3060 from circa10a/docs/helm-operator-integration

commit 7377aae0c1e3e01021c818a1d4dd8b8487e51b5e
Author: Caleb Lemoine <caleblemoine@gmail.com>
Date: Fri May 15 19:48:40 2020 -0500

docs: update helm operator integration glob patterns

commit 5433c28dd96585ee9db160f9ea4068ee4ed7b713
Merge: f6b18042 83c084e3
Author: Hidde Beydals <hiddeco@users.noreply.github.com>
Date: Wed May 13 12:42:36 2020 +0200

Merge pull request #3053 from maruina/secret-annotations

commit 83c084e3e3ddb99d6b81e1e932ebcf15379f150b
Author: Matteo Ruina <matteo.ruina@skyscanner.net>
Date: Mon May 11 16:38:33 2020 +0200

Support annotations in secret

Do not bump chart

Fix value used

commit f6b1804221699e38af85ef6ff5da1474d4317c83
Merge: 0ad6a83e e969cd71
Author: Hidde Beydals <hiddeco@users.noreply.github.com>
Date: Tue May 12 17:50:37 2020 +0200

Merge pull request #3008 from fluxcd/non-preferred-resources
```

Fix permissions in kube-addon-manager #91261

↳ Merged k8s-ci-bot merged 1 commit into kubernetes:master from tos13k:an-patch-1 20 hours ago

Conversation 4 Commits 1 Checks 0 Files changed 2 +5 -2

tos13k commented yesterday

What type of PR is this?
/kind bug

What this PR does / why we need it:
Last built `kube-addon-manager` image (v9.1.0) had wrong permissions set for both `kubectl` and `kube-addons.sh` files preventing updating the manifests in #91240 because AddonManager container wasn't able to execute them.

I tested the newly built image (v9.1.1) using the presubmit tests and it works.

Which issue(s) this PR fixes:
No issue is opened for that.

Special notes for your reviewer:
/sig scalability
/assign wojtek-t

Does this PR introduce a user-facing change?:
NONE

Additional documentation e.g., KEPs (Kubernetes Enhancement Proposals), usage docs, etc.:

Fix permissions in kube-addon-manager ✓ 3c5585b

k8s-ci-bot assigned wojtek-t yesterday

k8s-ci-bot added release-note-none kind/bug travis sig/scalability cncf-cla:yes needs-priority area/provider/gcp sig/cloud-provider labels yesterday

k8s-ci-bot requested review from cheftako and MrHohn yesterday

jtek-t commented yesterday

im iprove

Open

k8s-ci-bot added the lgtm label yesterday

View details

Merge 3c5585b

Assignees wojtek-t

Labels approved area/provider/gcp cncf-cla: yes kind/bug lgtm needs-priority release-note-none sig/cloud-provider sig/scalability sig/x5 Projects None yet Milestone v1.19 Linked issues Successfully merging this pull request may close these issues. None yet Notifications Customize Subscribe You're not receiving notifications from this thread. 4 participants

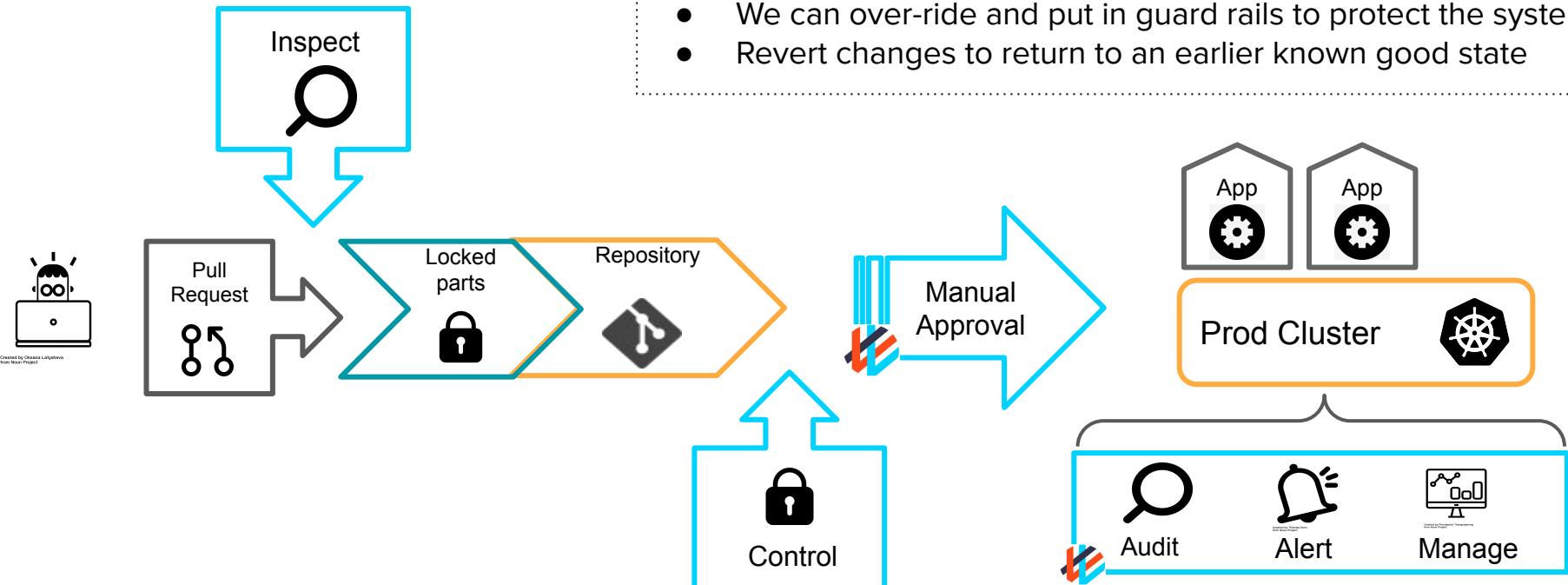
Aspect of Software Delivery Performance*	Elite	High	Medium	Low
Deployment frequency For the primary application or service you work on, how often does your organization deploy code to production or release it to end users?	On-demand (multiple deploys per day)	Between once per day and once per week	Between once per week and once per month	Between once per month and once every six months
Lead time for changes For the primary application or service you work on, what is your lead time for changes (i.e., how long does it take to go from code committed to code successfully running in production)?	Less than one day	Between one day and one week	Between one week and one month	Between one month and six months
Time to restore service For the primary application or service you work on, how long does it generally take to restore service when a service incident or a defect that impacts users occurs (e.g., unplanned outage or service impairment)?	Less than one hour	Less than one day ^a	Less than one day ^a	Between one week and one month
Change failure rate For the primary application or service you work on, what percentage of changes to production or released to users result in degraded service (e.g., lead to service impairment or service outage) and subsequently require remediation (e.g., require a hotfix, rollback, fix forward, patch)?	0-15% ^{b,c}	0-15% ^{b,d}	0-15% ^{c,d}	46-60%

<https://cloud.google.com/blog/products/devops-sre/the-2019-accelerate-state-of-devops-elite-performance-productivity-and-scaling>



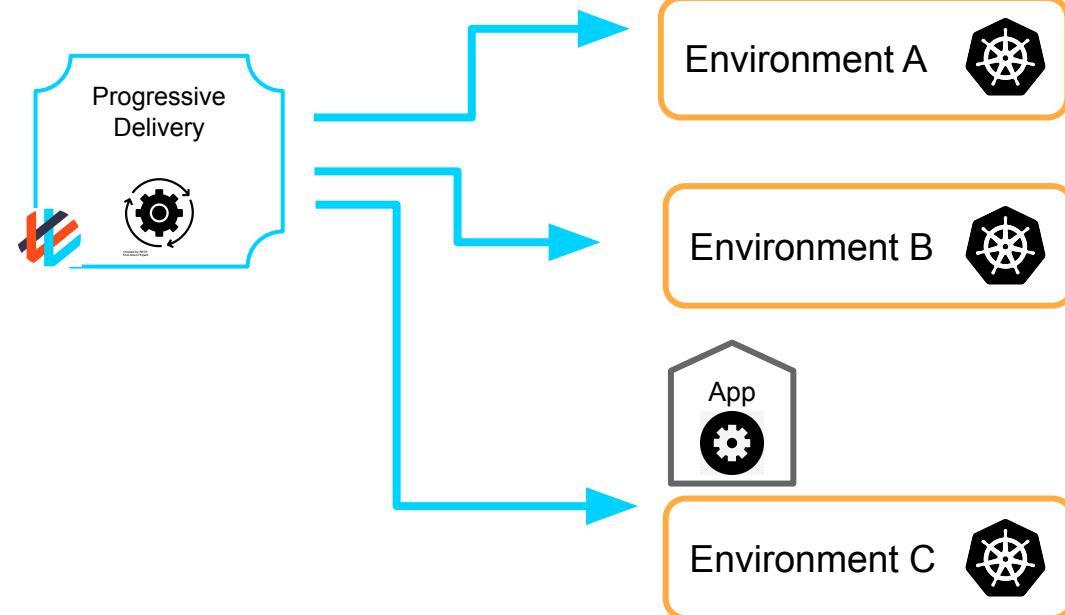
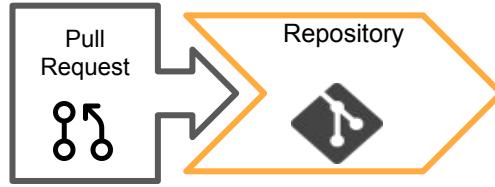
Move fast and don't break things

- Represents the entire state of the system - platform, services and applications
- We can easily inspect changes ensuring they meet standards
- We can over-ride and put in guard rails to protect the system
- Revert changes to return to an earlier known good state





Created by Olesya Litvinova
Weave Mesh Project



Progressive Delivery

- GitOps enables Progressive Delivery as each change is atomic and complete
- Easy to revert a change
- Can be for complete environments or the applications running in those environments

Aspect of Software Delivery Performance*	Elite	High	Medium	Low
Deployment frequency For the primary application or service you work on, how often does your organization deploy code to production or release it to end users?	On-demand (multiple deploys per day)	Between once per day and once per week	Between once per week and once per month	Between once per month and six months
Lead time for changes For the primary application or service you work on, what is your lead time for changes (i.e., how long does it take to go from code committed to code successfully running in production)?	Less than one day	Between two days and one week	Between two weeks and one month	Between one month and six months
Time to restore service For the primary application or service you work on, how long does it generally take to restore service when a service incident or a defect that impacts users occurs (e.g., unplanned outage or service impairment)?	Less than one hour	<ul style="list-style-type: none"> ● Resilience <ul style="list-style-type: none"> ○ Versioned, immutable store ○ Complete representation ○ Drift detection and remediation 		
Change failure rate For the primary application or service you work on, what percentage of changes to production or released to users result in degraded service (e.g., lead to service impairment or service outage) and subsequently require remediation (e.g., require a hotfix, rollback, fix forward, patch)?	0-15% ^{b,c}	0-15% ^{b,d}	0-15% ^{c,d}	46-60%
<ul style="list-style-type: none"> ● Familiar Tools - git for collaboration ● Self Service ● Pull-based (security) ● Implicit audit log (compliance) ● Resilience <ul style="list-style-type: none"> ○ Versioned, immutable store ○ Complete representation ○ Drift detection and remediation ● Config reviews in Git ● Progressive Delivery 				

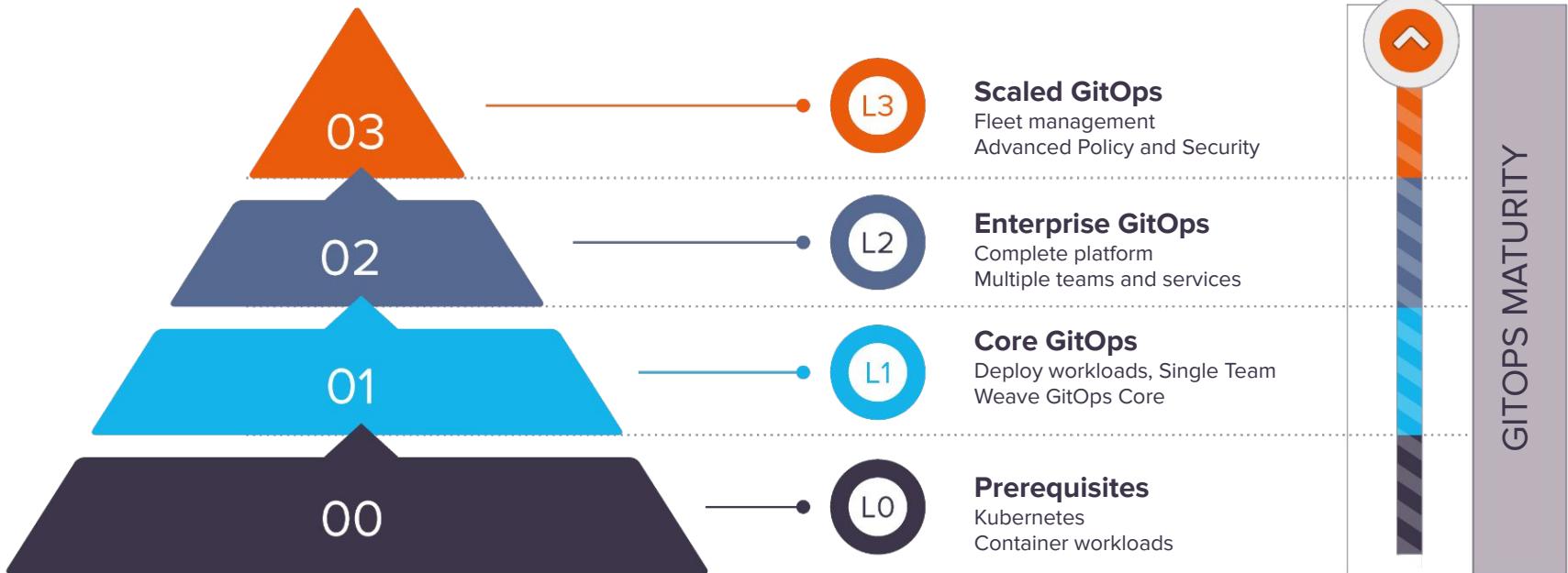
<https://cloud.google.com/blog/products/devops-sre/the-2019-accelerate-state-of-devops-elite-performance-productivity-and-scaling>



Adopting GitOps



GitOps Maturity Model - Summary





谢谢

Thank You

Danke

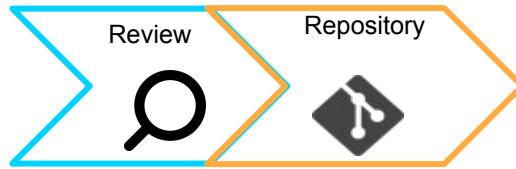
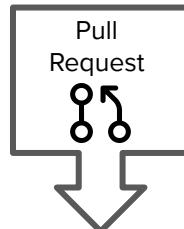
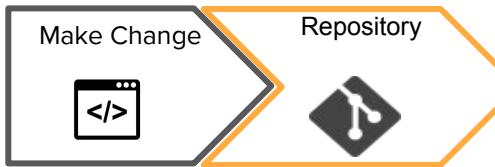
Obrigado

Спасибо!

Merci

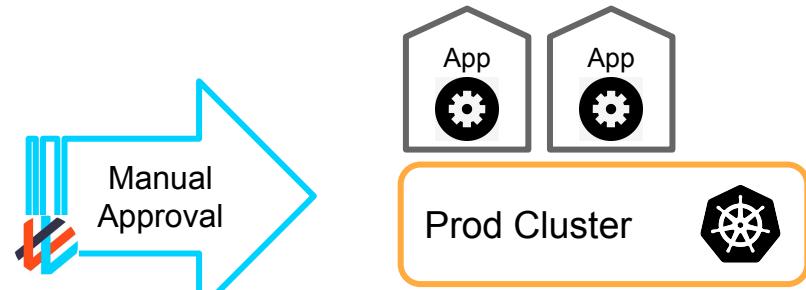
شكرا

Steve George
COO, Weaveworks
@futurile • steveg@weave.works

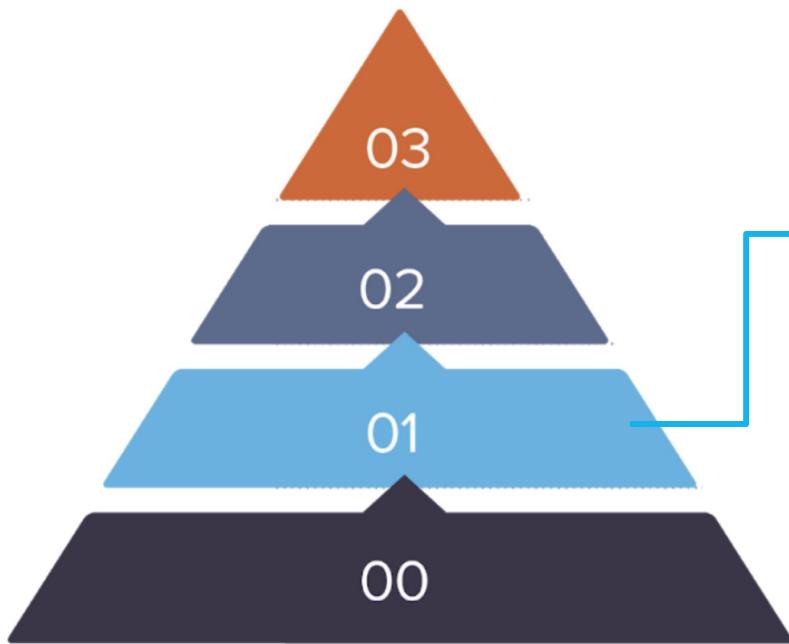


A point of collaboration

- Operations by Pull Request - a natural way for teams to work together
- Teams can choose who deploys where, when and how
- Configuration is distributed



Level 1 - Core GitOps



Capabilities

- Start with simple applications and services - few dependencies
- GitOps for the Workload / Application layer
 - Automated reconciliation / drift detection
- Declarative infrastructure as code but without automatic reconciliation
- Environment & Secrets Management

Open Community - GitOps Working Group



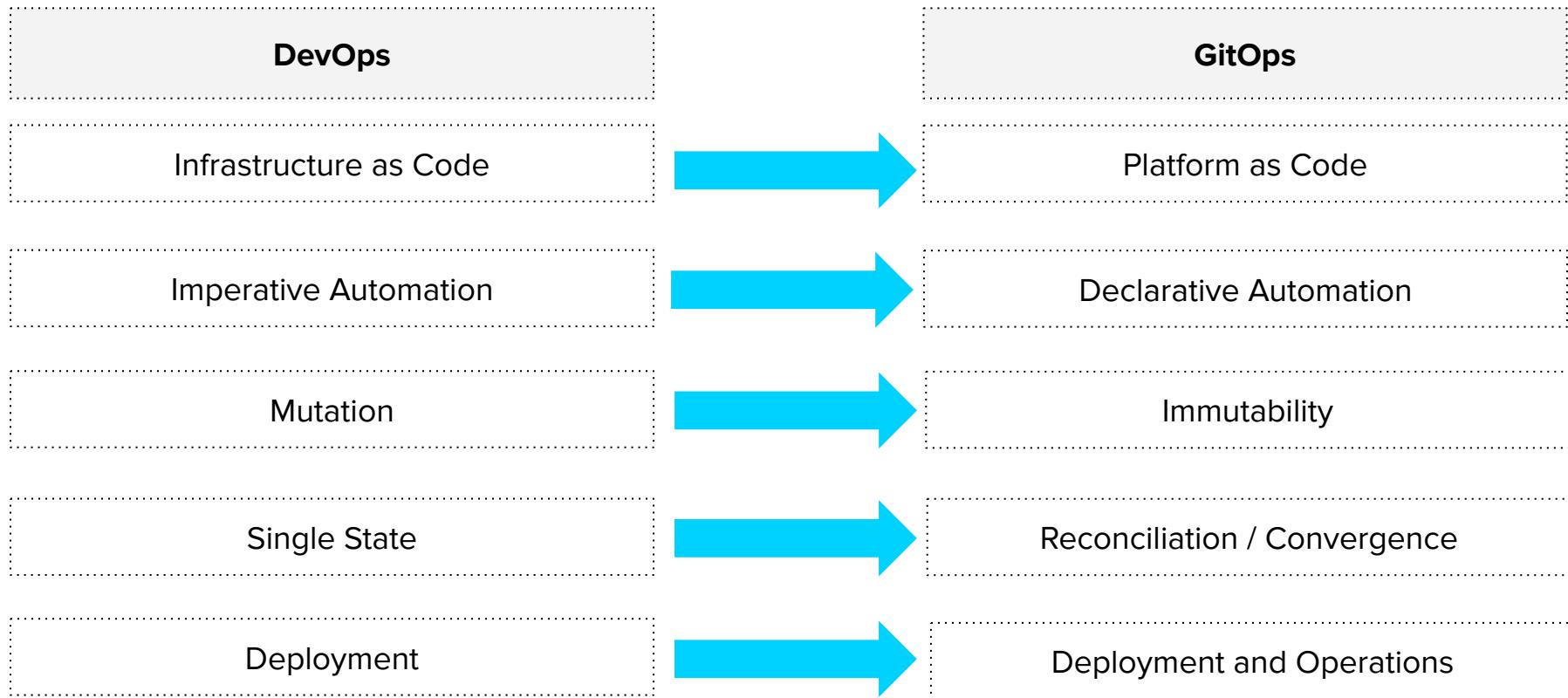
- Defining a vendor neutral, principle-led meaning for GitOps
 - Interoperability and clarity
 - Exploring and developing the use-cases
- Industry wide collaboration
 - Amazon, Codefresh, GitHub, Microsoft and Weaveworks are founding members
 - User community
- GitOps Principles
 - Declarative Configuration
 - Version controlled, immutable storage
 - Automated delivery
 - Software agents reconcile
 - Closed loop

<https://github.com/gitops-working-group/gitops-working-group>

Thesis:

GitOps supports the DevOps agenda in a particularly effective manner. (where Git is the interface for operational actions.)

Evolution of DevOps to GitOps



Principles of GitOps



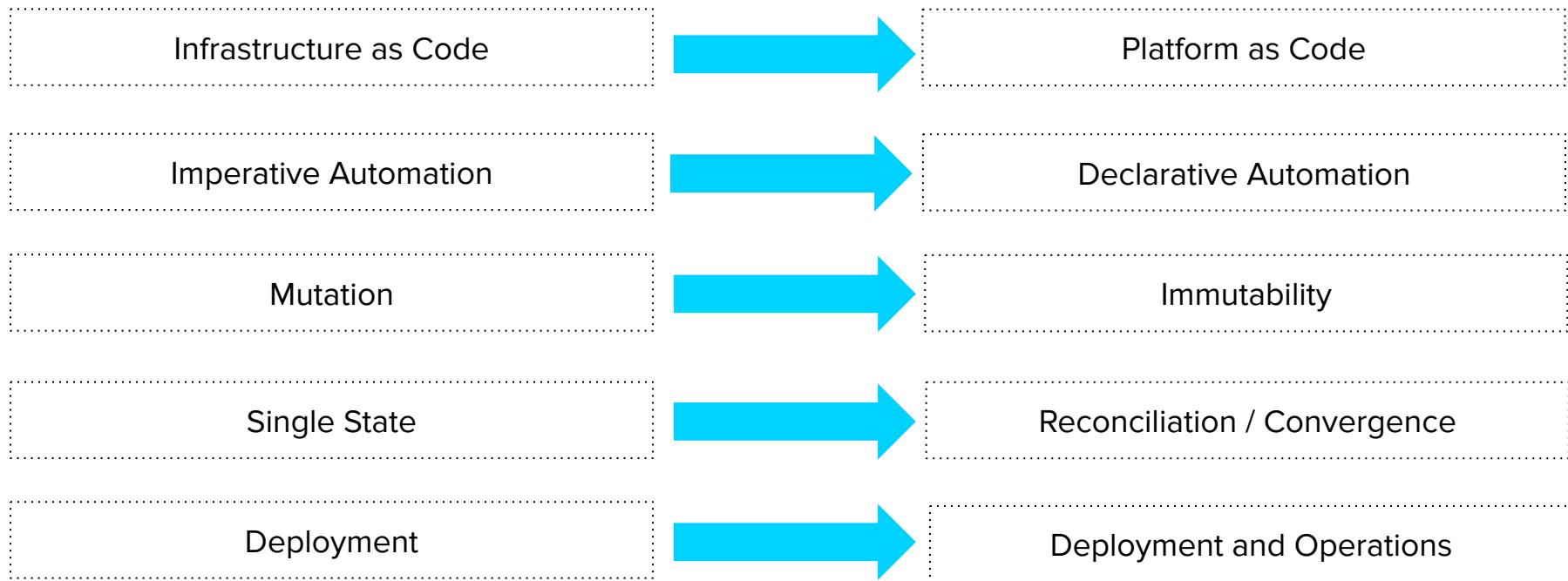
The entire system is described **declaratively**

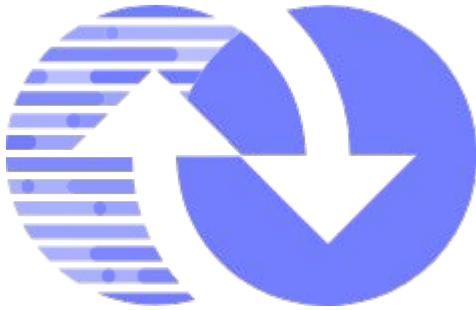
The canonical desired system state is **versioned** in git

Approved changes can be **automatically applied** to the system

Software agents ensure correctness and alert (diffs & actions)

GitOps Elements





<https://github.com/gitops-working-group/gitops-working-group>

1. **Declarative Configuration:** All resources managed through a GitOps process must be completely expressed declaratively.
2. **Version controlled, immutable storage:** Declarative descriptions are stored in a repository that supports immutability, versioning and version history. For example, git.
3. **Automated delivery:** Delivery of the declarative descriptions, from the repository to runtime environment, is fully automated.
4. **Software Agents:** Reconcilers maintain system state and apply the resources described in the declarative configuration.
5. **Closed loop:** Actions are performed on divergence between the version controlled declarative configuration and the actual state of the target system.

The Principles of GitOps



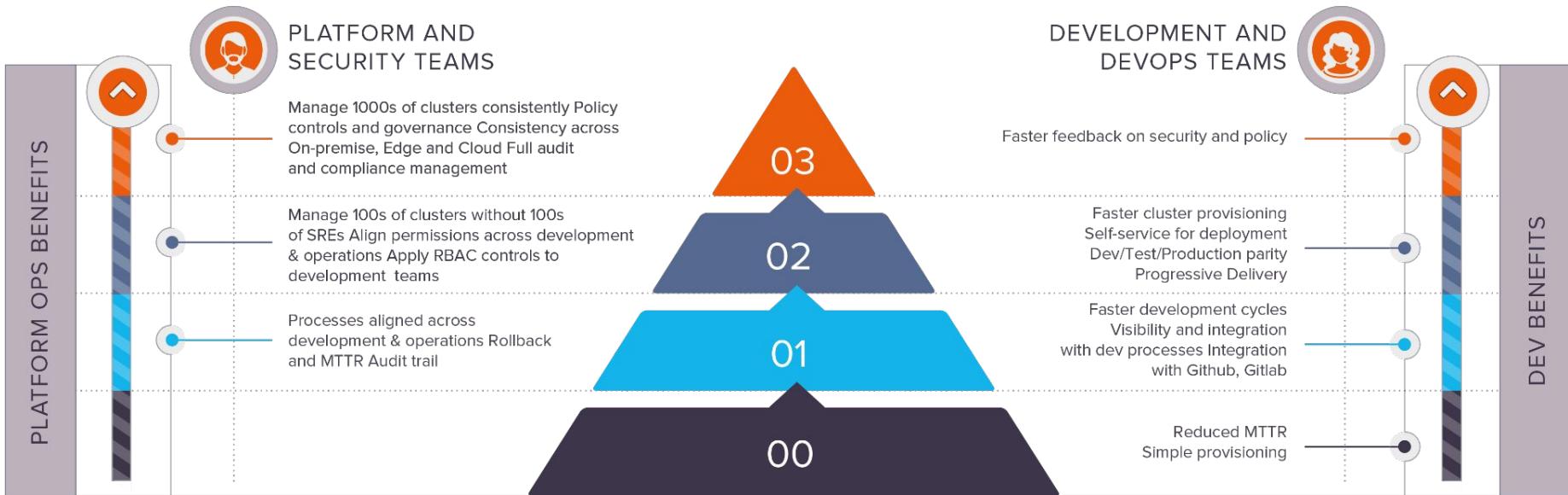
The entire system is described **declaratively**

The canonical desired system state is versioned in **git**

Approved changes can be **automatically applied** to the system

Software agents ensure correctness and perform actions on divergence in a closed loop

GitOps Maturity Model - Benefits for different teams



GitOps Maturity Model - Details

		Capabilities	Business Benefit
INCREASING MATURITY	Level 3 Scaled GitOps	Fleet management and fleet reconciliation Mass customization Advanced Policy Advanced Cross-Enterprise Governance Multi-Cloud and Hybrid	Enables Support for 5G/IoT/Edge deployment Significant cost reduction for fleet management Supports massive scale Enables Enterprise Governance and Compliance Empowers cross-vendor choice, visibility and control
	Level 2 Enterprise GitOps	GitOps for the whole environment <ul style="list-style-type: none">- Infra, Cluster, Config, Workloads Separation of Cluster from Workloads Security / RBAC / Audit / Governance / Policy Progressive Delivery Package / Template Customization & Management	Improves agility across teams Further reductions in cost as you scale Enables Governance & Compliance validation & audit Improves security Enables Corporate Policy enforcement
	Level 1 Core GitOps	Declarative IaC without reconciliation GitOps for the Workload / App <ul style="list-style-type: none">- Automated reconciliation / drift detection Environment & Secrets Management Pull-based approach Package Management (helm or kustomize)	Increases frequency of deployment Reduces cost of operations Reduces MTTR Increases visibility and audit of operations Introduces observability Introduces basic Governance
	Level 0 Prerequisites for GitOps	Declarative Infrastructure as Code (IaC) Declarative Workloads Version control, Immutable images Suitable infrastructure (e.g. Kubernetes)	“Cloud-native” Faster provisioning of infrastructures Repeatability Immutability

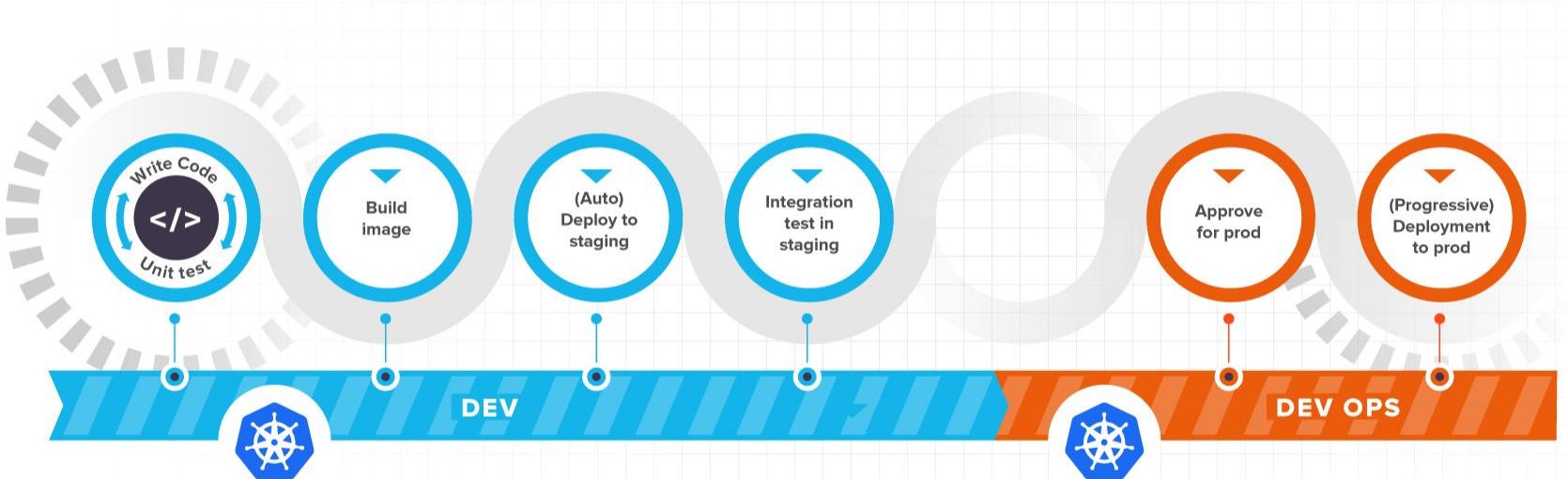


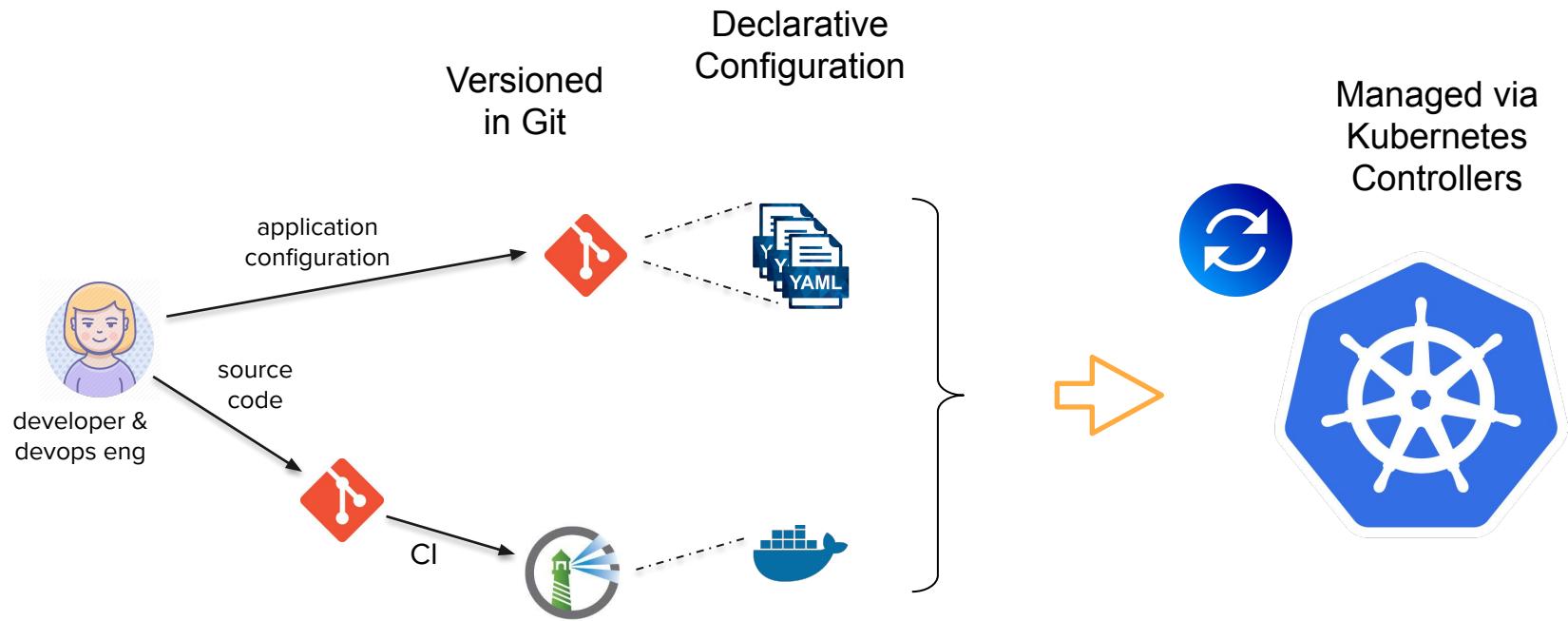
GitOps – An Operating Model for Cloud Native

Unifying Deployment, Monitoring and Management.

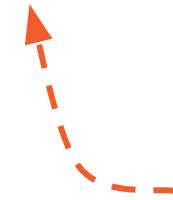
Git as the single source of truth of a system's desired state

- **ALL** intended operations are committed by pull request
- **ALL** diffs between intended and observed state with automatic convergence
- **ALL** changes are observable, verifiable and auditable

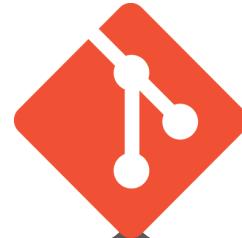




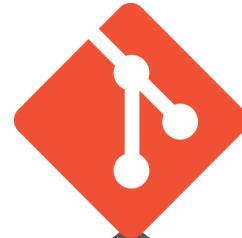
What should be GitOps'ed?



*I'm so very
sorry*



?



Kubernetes Manifests

Dashboards

Application checklists

Application configuration

Alerts

Monitoring Data
Pipeline

Provisioning config

Operator
Runbooks

Sealed Secrets

Use Cases

Different use-cases for Weave GitOps

1. Application Delivery and Operations

Enabling application development teams to do DevOps at scale.
Deploying applications through a dev/staging/prod pipeline.

2. Platform as code

Deploying the same ‘standard platform’ using a platform-as-code approach. Whether on-premise or in the cloud. Deploying and managing multiple different Kubernetes clusters.

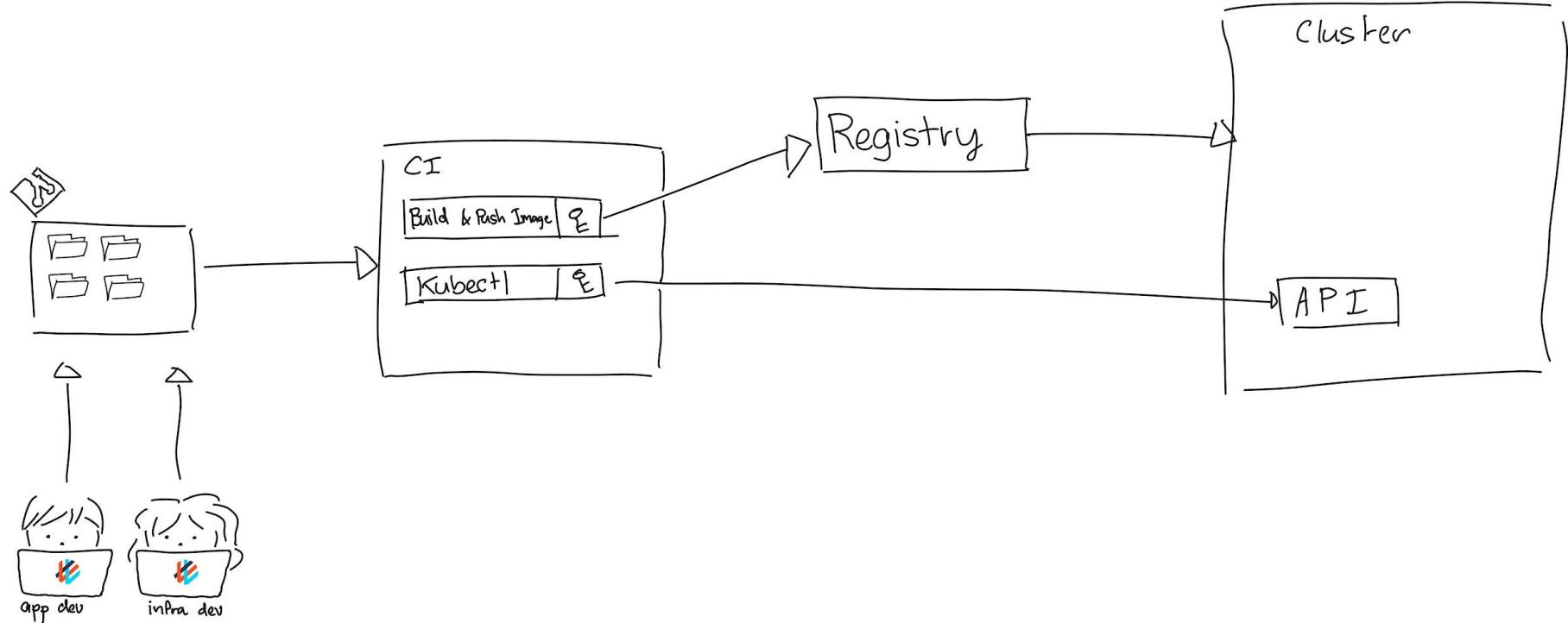
3. Security and controls

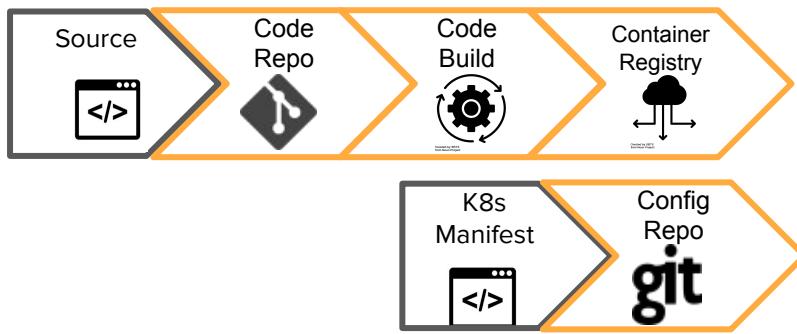
Using GitOps-based security controls to secure the deployment for both Operators and DevOps teams.

The screenshot displays the Weaveworks Kubernetes Platform interface. On the left, a sidebar titled 'Workspaces' lists three entries: 'devs-workspace', 'team-bananas-workspace', and 'team-victor-workspace'. Each entry includes details such as the namespace (e.g., 'devs-ns', 'team-bananas-ns', 'team-victor-ns'), the GitHub repository URL (e.g., 'ssh://git@github.com/foot-org/devs-workspace'), and the commit hash (e.g., '2020-11-25T11:52:09Z'). To the right of the workspace list is a GitHub repository page for 'team-bananas-workspace'. The repository shows 0 stars, 0 forks, and 4 pull requests. A file named 'nginx.yaml' is shown with a commit message 'foot Update nginx.yaml' made 13 days ago. Below the GitHub interface is a 'README.md' file containing instructions for deploying a workload using GitOps.

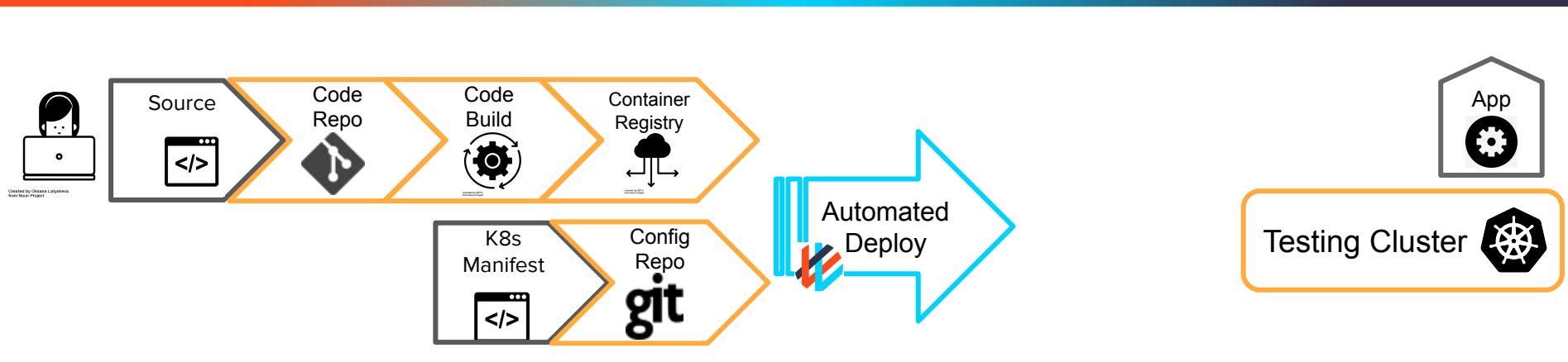


Traditional CI/CD

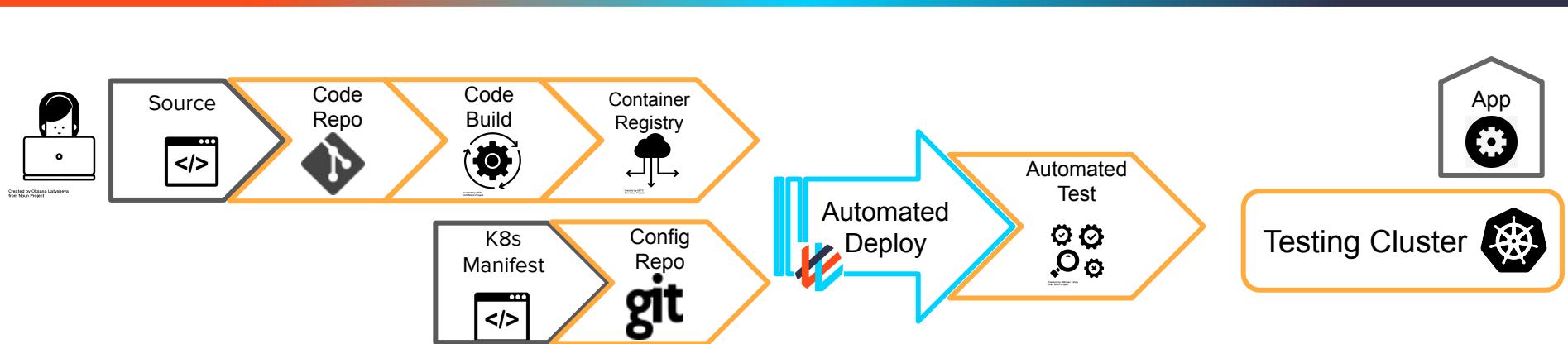




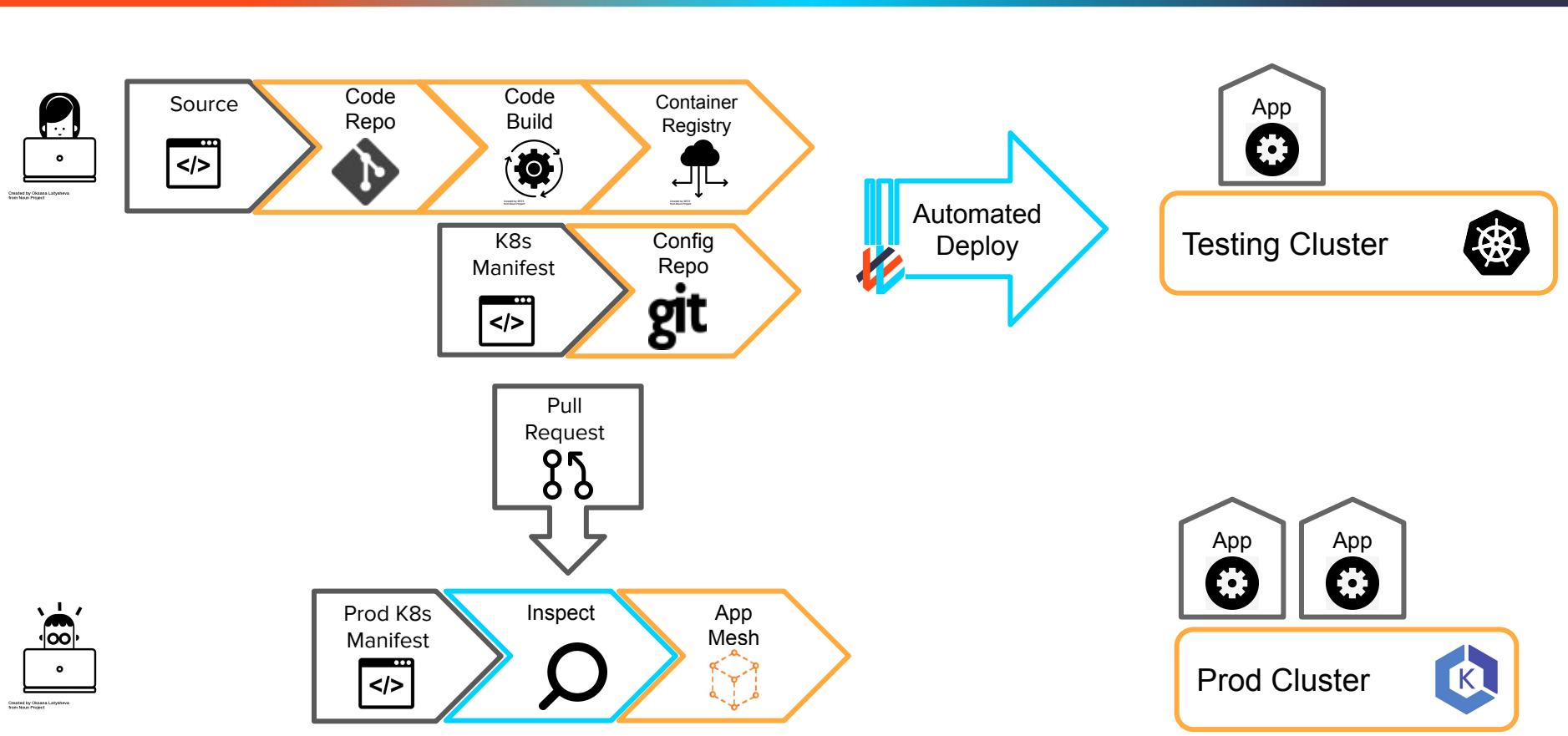
Testing Cluster

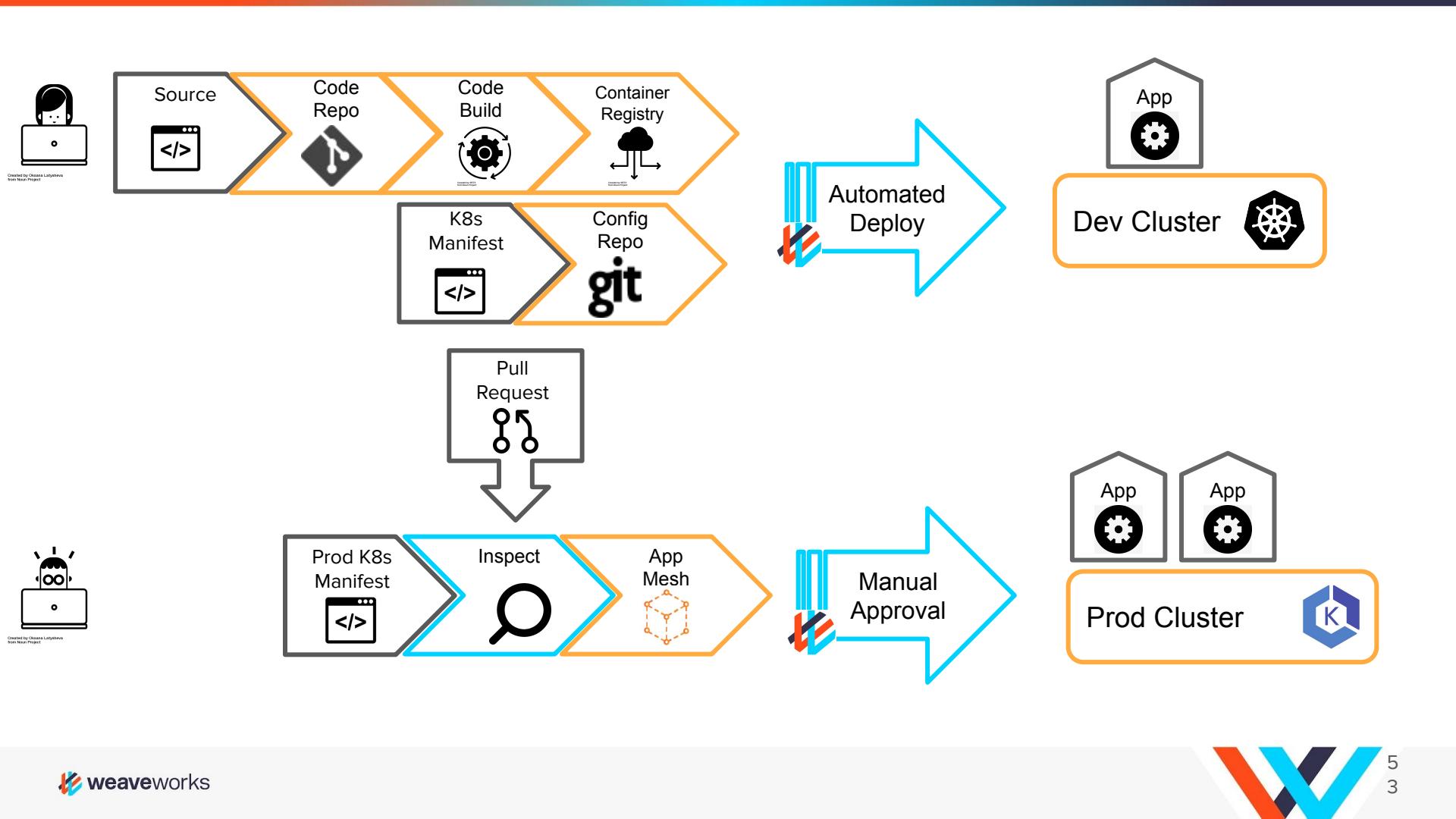


Automated deployment



Automated Testing as part of
the pipeline





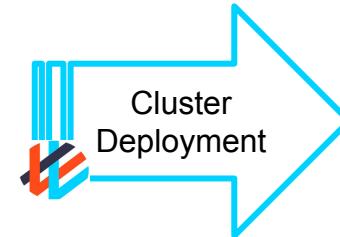
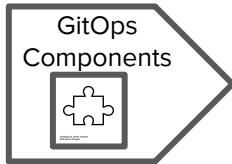
Use Case: Cluster Operations

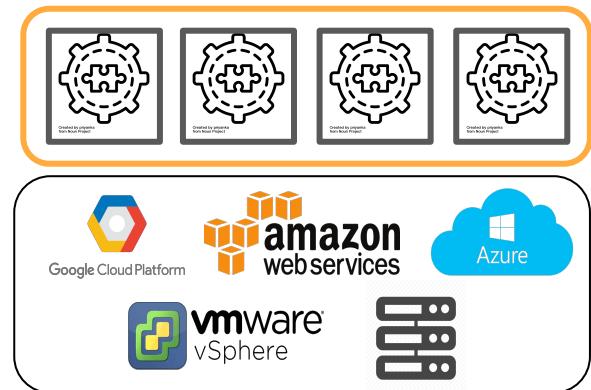
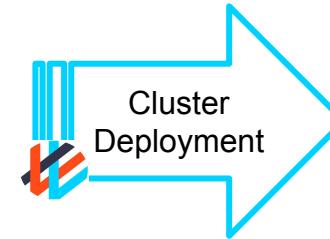
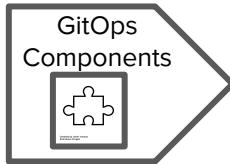
Weave GitOps Enterprise - Cluster Operations

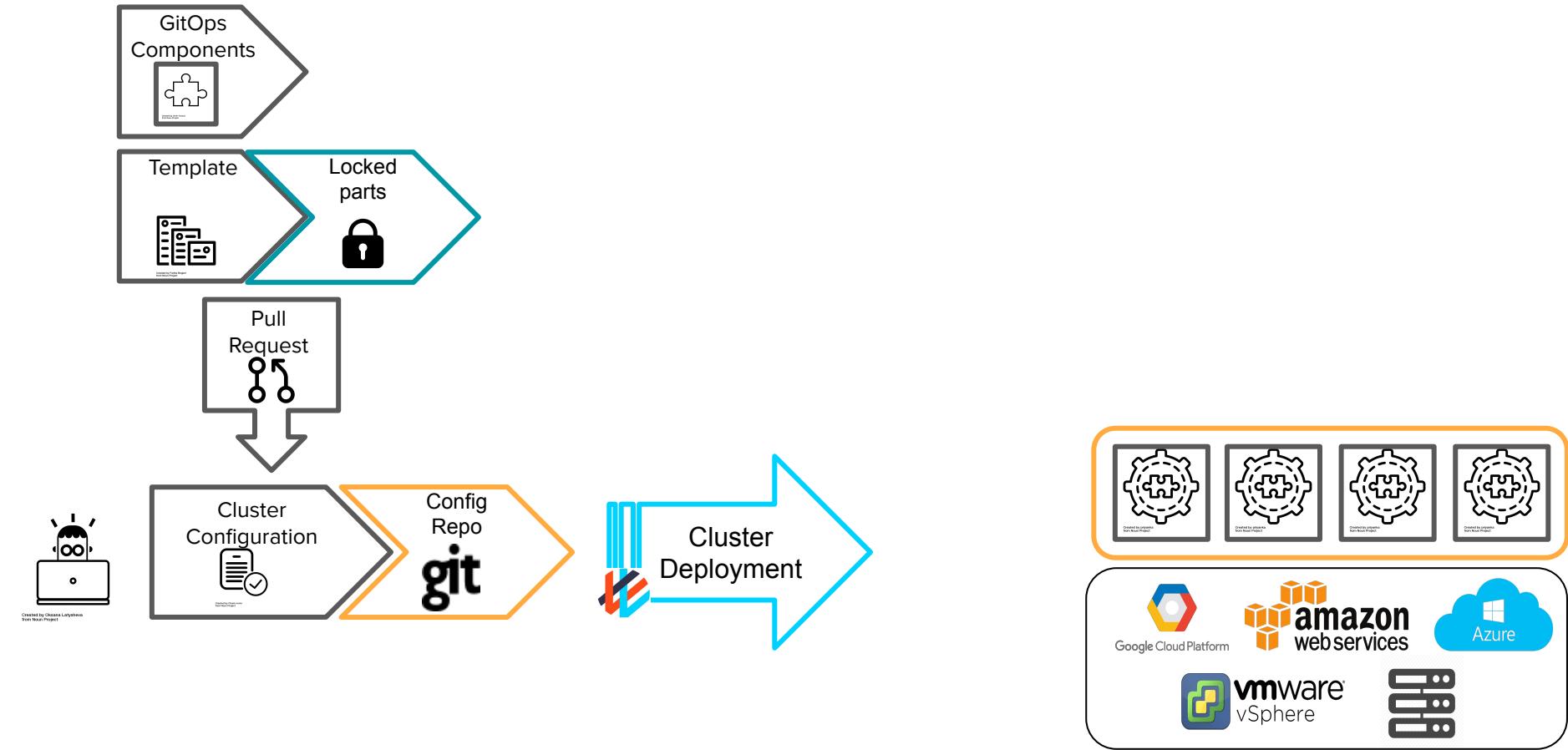
- Standard Cluster components to deploy a platform-as-code
- Multiple different K8s backends
- Templating to provide configuration and multi-cluster management

The screenshot displays the Weaveworks Kubernetes Platform interface. On the left, a sidebar lists three workspaces: 'devs-ns', 'team-bananas-ns', and 'team-victor-ns'. Each workspace entry includes a 'Namespace' column (e.g., 'foot-org / devs-namespace'), a 'Repository' column (e.g., '2020-11-25T11:52:09Z ssh://github.com/foot-org/devs-workspace'), and a 'Status' column. Below the workspace list is a note: 'Need help? Contact us at support@weave.works'. To the right, the 'team-bananas-ns' workspace is selected, showing its GitHub repository details. The GitHub interface shows 0 stars, 0 forks, and 4 pull requests. A commit titled 'foot Update nginx.yaml' is visible, dated 13 days ago. The commit message is 'View code'. At the bottom of the GitHub view, there's a 'README.md' section and a terminal-like area with the command 'git add deployment.yaml'.









Use Case: Security and Controls

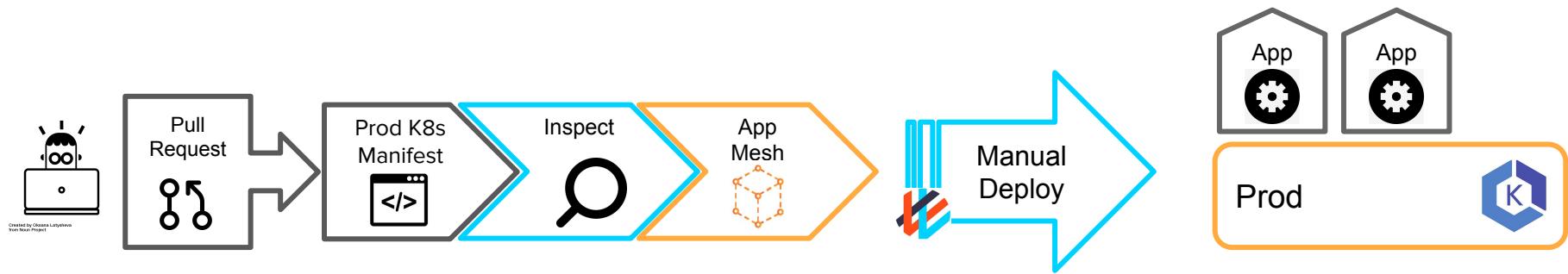
Security and controls

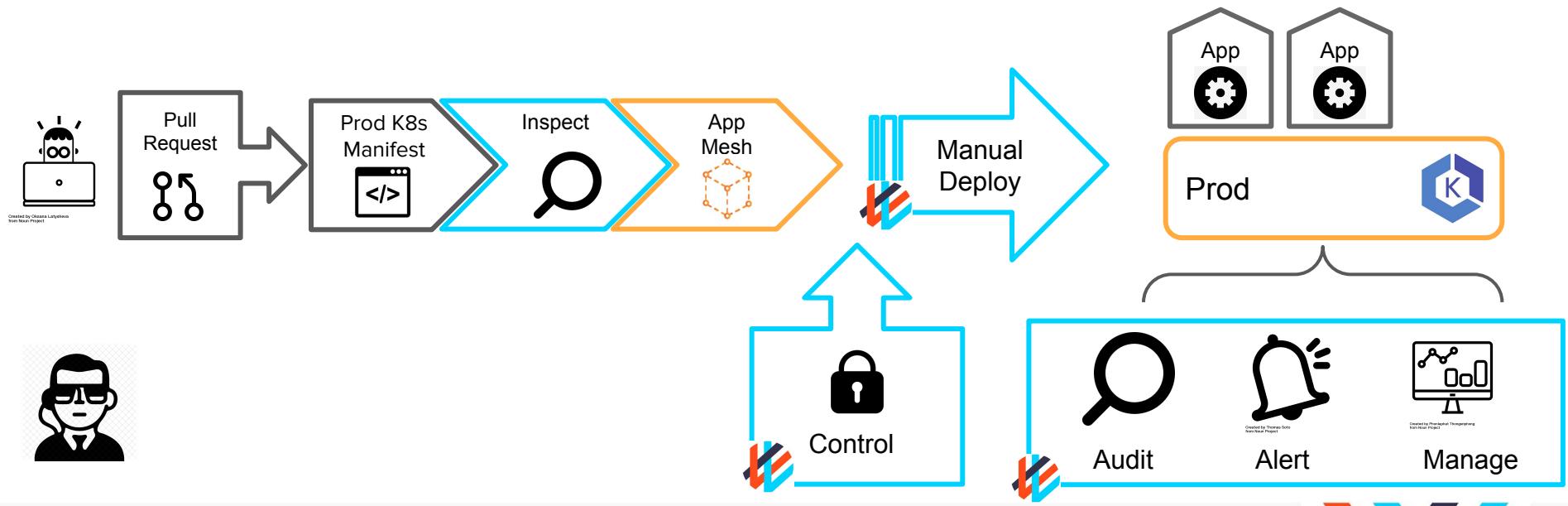
Using GitOps-based security controls to secure the deployment for both Operators and DevOps teams.

Adding additional guard rails to prevent bad deployments.

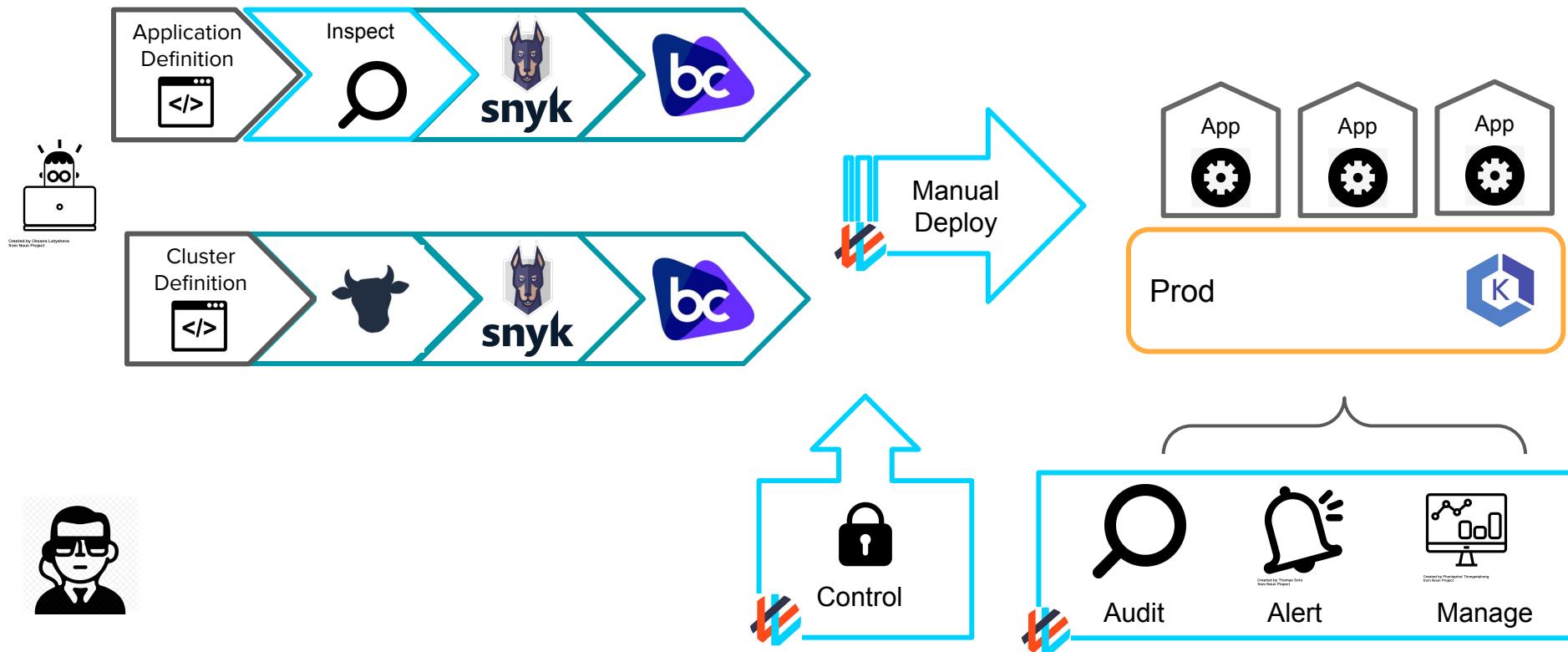
Operating and inspecting applications in production.

The screenshot displays the Weaveworks Kubernetes Platform interface. On the left, a sidebar lists three workspaces: 'devs-workspace', 'team-bananas-workspace', and 'team-victor-workspace'. Each workspace entry includes the namespace, repository URL (e.g., 'foot-org / devs-workspace'), and a 'View in GitHub' button. To the right of the sidebar, a detailed view of the 'team-bananas-workspace' is shown. This view includes a GitHub repository card for 'team-bananas-workspace' (private, 0 stars, 0 forks), a code editor with a commit titled 'foot Update nginx.yaml' (13 days ago), and a 'README.md' file. Below the code editor, there's a section titled 'team-bananas-workspace' with instructions for deploying a workload using GitOps.

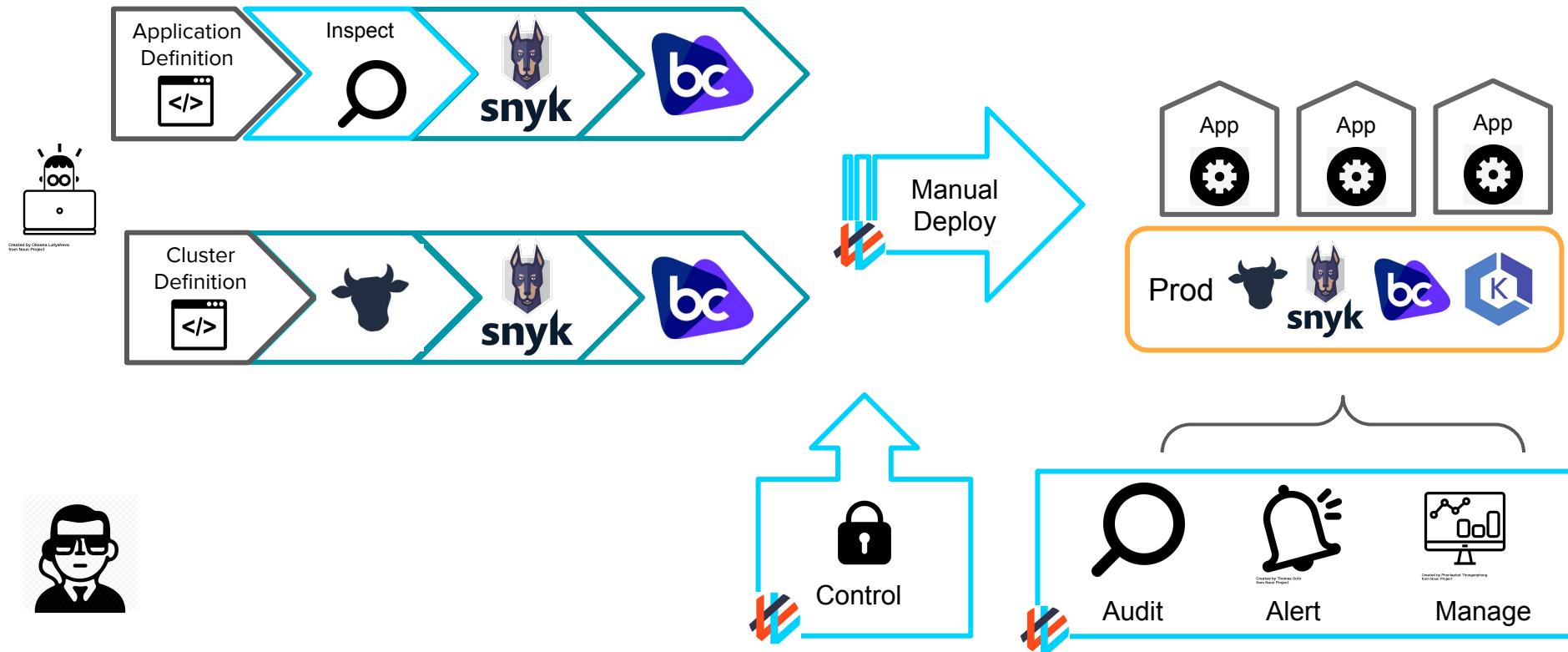




Cluster and Application Control



Cluster and Application Control



GitOps Maturity Model - Summary

