GitLab

Red Hat

GitOps

Agenda

- Why GitOps? What problem is it solving?
- RedHat & GitOps
- GitLab & GitOps
- Use Cases

GitLab

Why GitOps?

About me

Daniel Marquard

Sr. Defense Solutions Architect

@danielmarquard

gitlab.com/danielmarquard

linkedin.com/in/danielmarquard

What is GitOps?

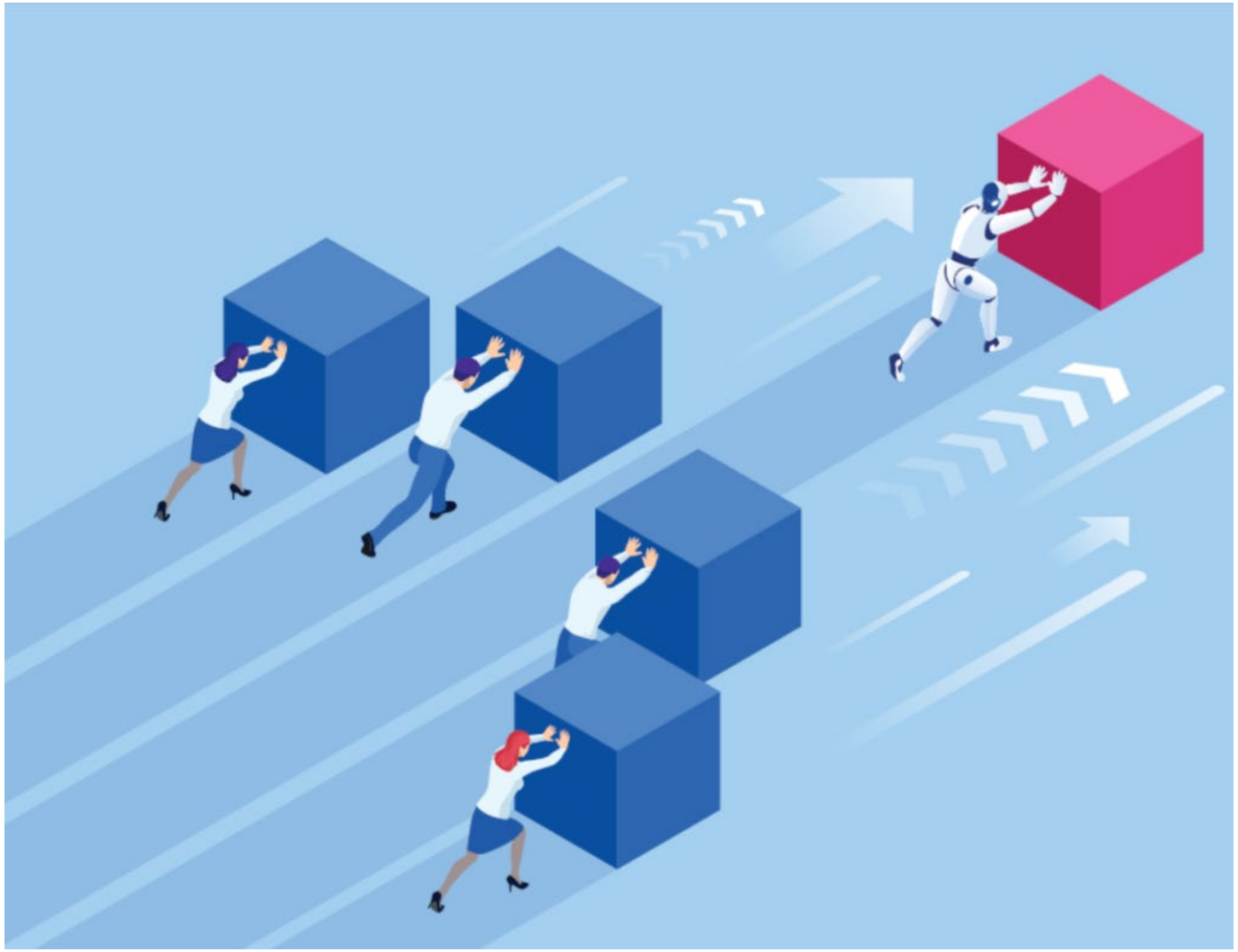
> GitOps is an operational framework
> that takes **DevOps best practices**
> used for application development such as version control,
> collaboration, compliance, and CI/CD,
> and applies them to **infrastructure automation** .

IaC...why bother?

Repeatability

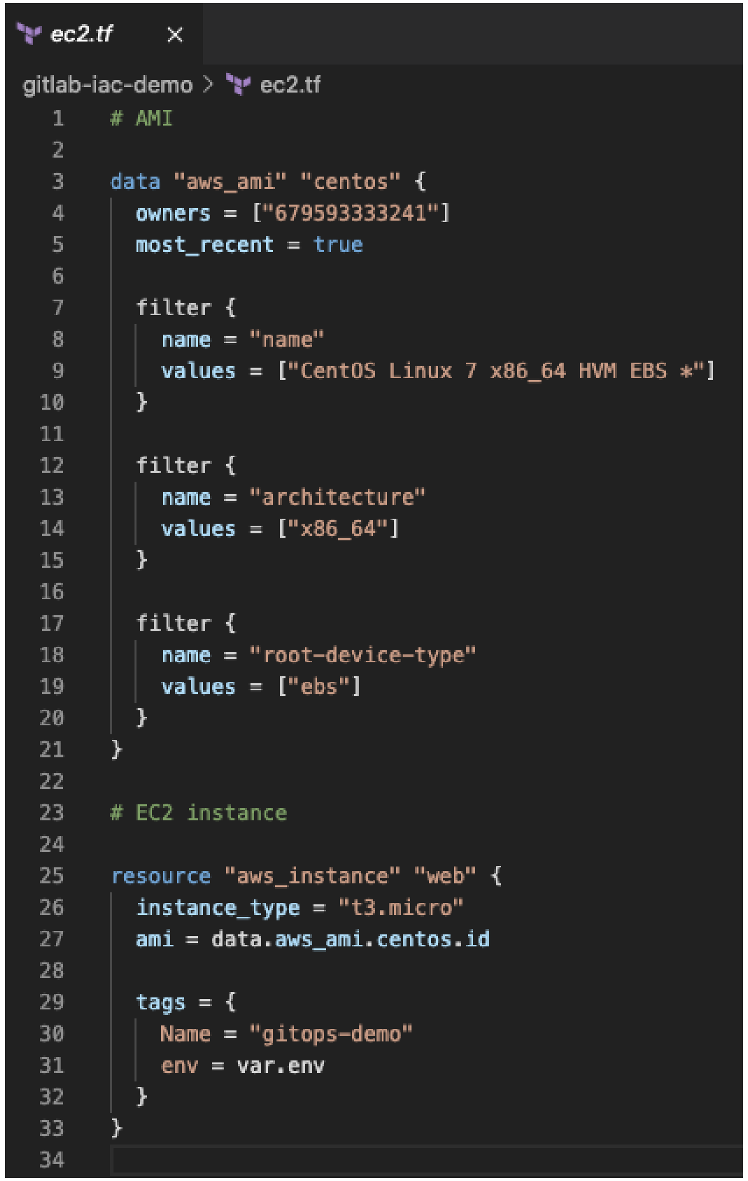
- Automation
- Deploy to 1 environment

  or 1,000
- Consistency



IaC is self-documenting

- Manually-configured resources are black boxes
- No need for infrastructure documentation beyond diagrams
- Through git and change control policies, infrastructure changes are tracked over time



```
# AMI

data "aws_ami" "centos" {
  owners = ["679593333241"]
  most_recent = true

  filter {
    name = "name"
    values = ["CentOS Linux 7 x86_64 HVM EBS *"]
  }

  filter {
    name = "architecture"
    values = ["x86_64"]
  }

  filter {
    name = "root-device-type"
    values = ["ebs"]
  }
}

# EC2 instance

resource "aws_instance" "web" {
  instance_type = "t3.micro"
  ami = data.aws_ami.centos.id

  tags = {
    Name = "gitops-demo"
    env = var.env
  }
}

```

IaC enables idempotence

Idempotence *(/ˌaɪdəmˈpoʊtəns/)*

The property of certain operations in mathematics and computer science whereby they can be applied multiple times without changing the result beyond the initial application.

- Provides control, regulation, and predictability
- Consistent behavior across all deployments
- Underlying defects impact all of none
- Bug fixes apply to all

GitOps = IaC + MRs + CI/CD

GitOps vs IaC

GitOps

- Code is stored in git repository
- Change is enacted via Merge Request
- Code is scanned for security and best practices
- Infrastructure updates are automated

IaC

- Code may/may not be version controlled
- Code changes may or may not go through a review/approval process.
- Changes can be applied many ways (FTP or SSH to the server, command line manual runs, etc.) They may or may not be automated.

Components of GitOps

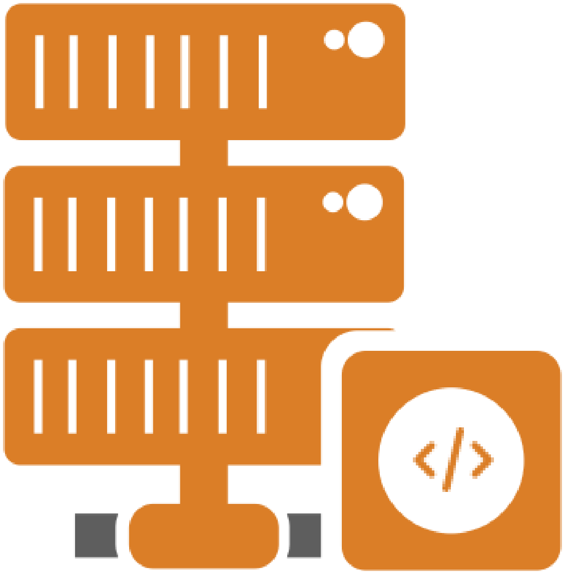
IaC

MRs

CI/CD

GitOps: Environments stored as code in Git

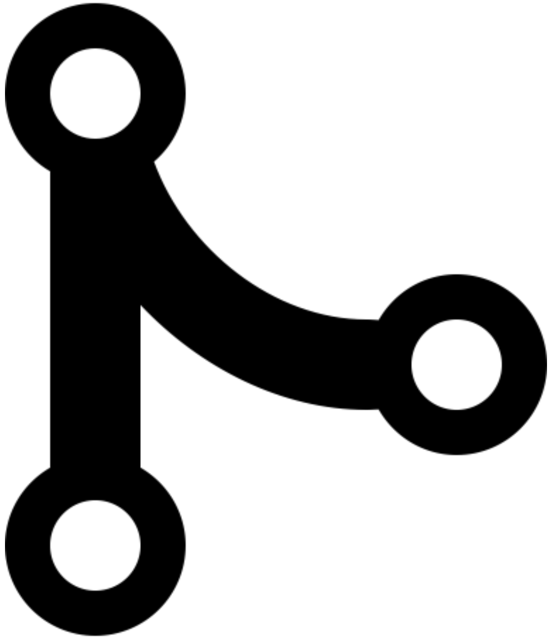
IaC



- Infrastructure as Code (IaC) or X-as-Code (XaC-infra, config, policy, etc.)
- Declarative code describes the desired state
- Stored in a Git version control
- Git tooling as the UI

GitOps: MRs as the agent of change

MRs



- Merge Requests (MRs) or Pull Requests (PRs) are the “gate”

- Main branch == product branch (Default, Trunk, “main”, etc.)

- Code review, collaboration, and approvals

GitOps: Automated Reconciliation

CI/CD

- Continuous Integration and Continuous Delivery (CI/CD) as a “reconciliation loop”
- CI/CD runs as a reconciler loop. Can be agent (pull) or agentless (push)
- When the infrastructure state is out of sync with the definition CI/CD updates the infra to match the definition in Git
- Changes are implemented automatically (no manual updates)

GitOps challenges to adoption

- Operations engineers need to adopt a developer-centric view of their work
- Need a sophisticated level of deployment automation
- Not all operations belongs in git
  - Observability tools
  - Feature flags
  - Incident management

GitLab

GitLab and GitOps

GitLab is the most popular solution for the Enterprise

COMPANY

- Incorporated in 2014
- 1300+ employees across 65 countries
- GitLab Federal entity est. in 2018

BROAD ADOPTION

- 100,000+ organizations
- Millions of users
- 70% share of selfmanaged DevOps repository market

STRONG COMMUNITY

- Open source model
- 2,500+ code contributors
- 10,000+ total contributors

GENERAL DYNAMICS

NORTHROP GRUMMAN

Raytheon

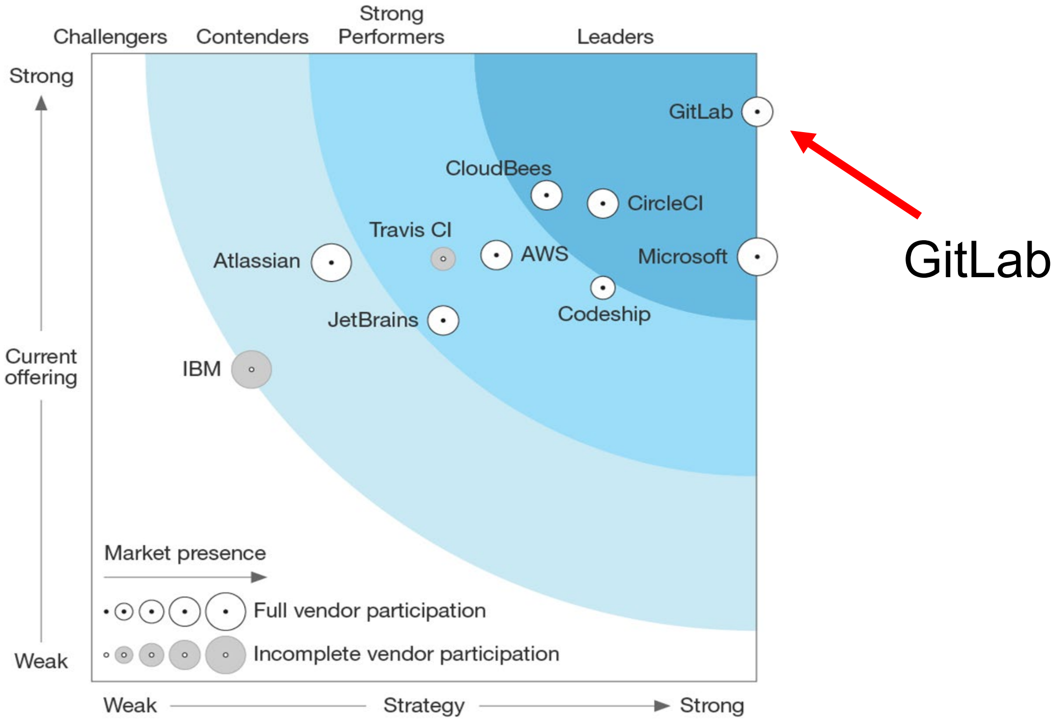
LOCKHEED MARTIN

BAE SYSTEMS
INSPIRED WORK

GitLab recognized as a Leader

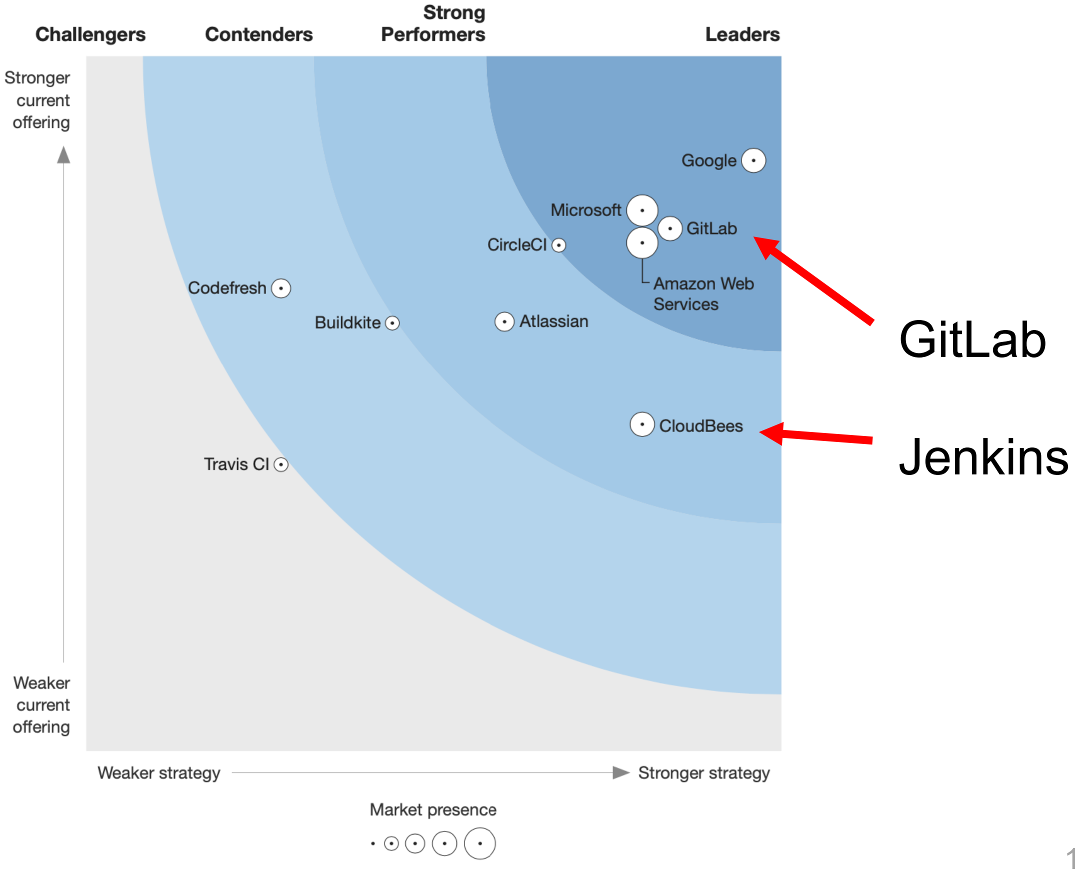
Forrester: CI

“GitLab’s vision is to serve enterprise -scale, integrated software development teams - The Forrester WaveTM: Continuous Integration Tools, Q3 2017 report



Forrester: Cloud Native CI

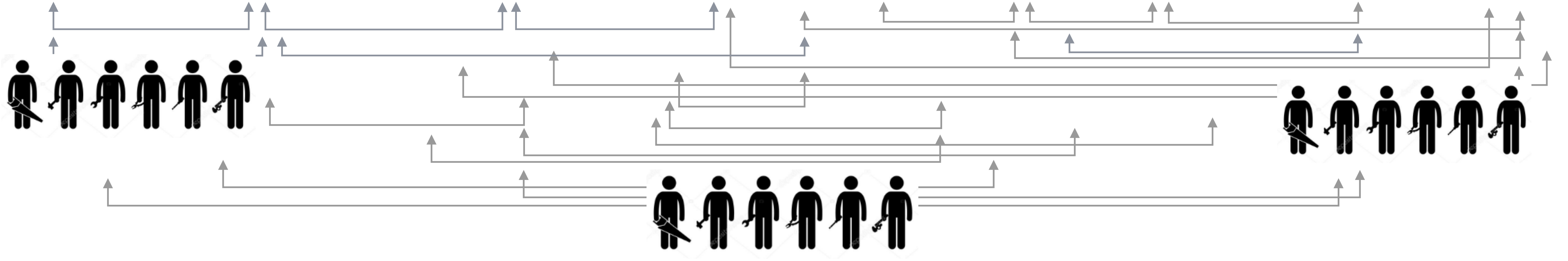
The only non cloud provider to be a leader



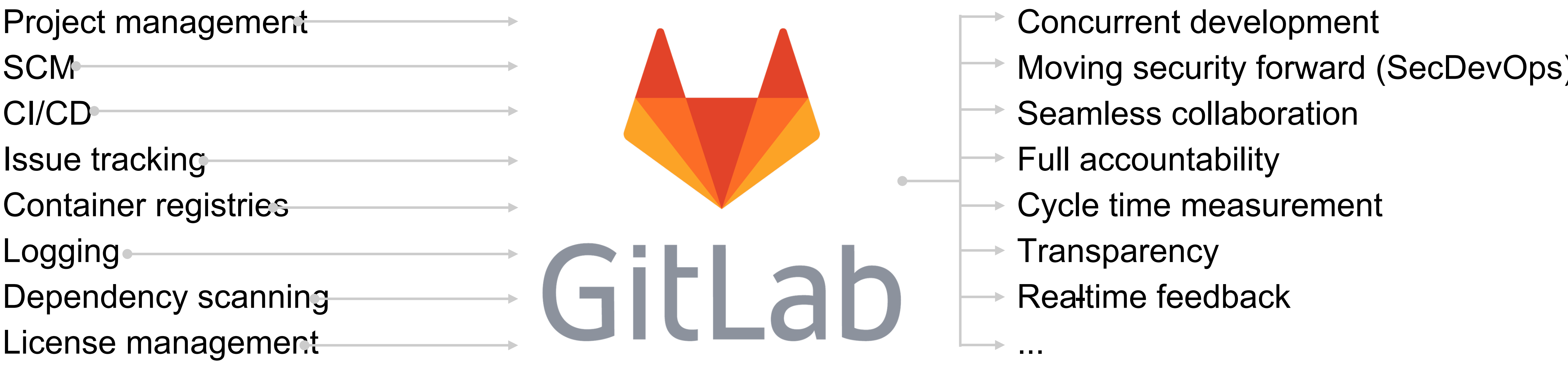
The Forrester Wave™ is copyrighted by Forrester Research, Inc. Forrester and Forrester Wave™ are trademarks of Forrester Research, Inc. The Forrester Wave™ is a graphical representation of Forrester's call on a market and is plotted using a detailed spreadsheet with exposed scores, weightings, and comments. Forrester does not endorse any vendor, product, or service depicted in the Forrester Wave. Information is based on best available resources. Opinions reflect judgment at the time and are subject to change.

Today's Mission Challenge- Complex Toolchains Reinforce Silos

Manage	Plan	Create	Verify	Package	Secure	Release	Configure	Monitor	Defend
COLLABNET VERSIONONE	JIRA	GitHub	Jenkins	JFrog Artifactory	FORTIFY		p	New Relic.	Symantec.
VERSIONONE	REDMINE flexible project management	Gerrit	Travis CI	Sonatype Nexus	sonarqube	XL DEPLOY	CHEF	Nagios	splunk>



Say hello to GitLab- an entirely new thing



GITLAB IS REPLACING POINT TOOLS…

- Project management
- SCM
- CI/CD
- Issue tracking
- Container registries
- Logging
- Dependency scanning
- License management

AND ENABLING ENTIRELY NEW CAPABILITI

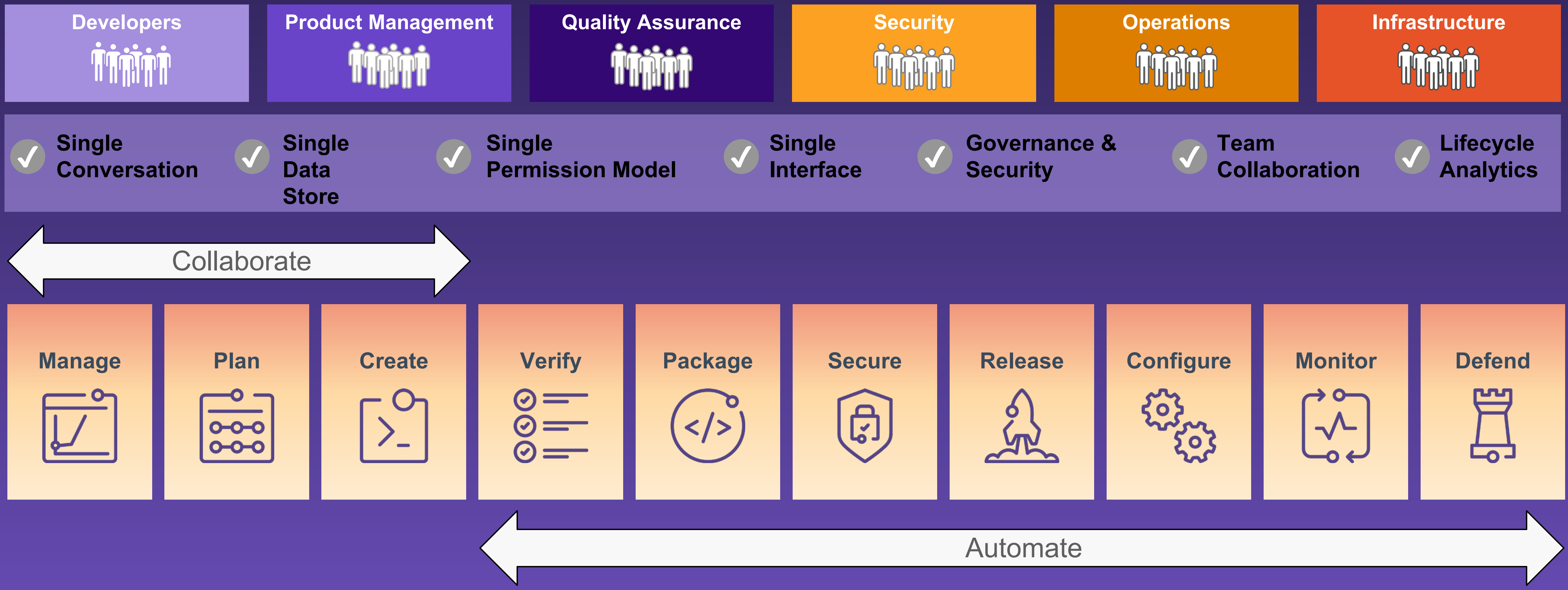
- Concurrent development
- Moving security forward (SecDevOps
- Seamless collaboration
- Full accountability
- Cycle time measurement
- Transparency
- Real-time feedback
- ...

Asingle, intuitive user experience, data model and integrations

GitLab

Built from the ground up as a single application

200% faster DevSecOps lifecycle



Developers

Product Management

Quality Assurance

Security

Operations

Infrastructure

- ✓ Single Conversation
- ✓ Single Data Store
- ✓ Single Permission Model
- ✓ Single Interface
- ✓ Governance & Security
- ✓ Team Collaboration
- ✓ Lifecycle Analytics

Collaborate

Manage

Plan

Create

Verify

Package

Secure

Release

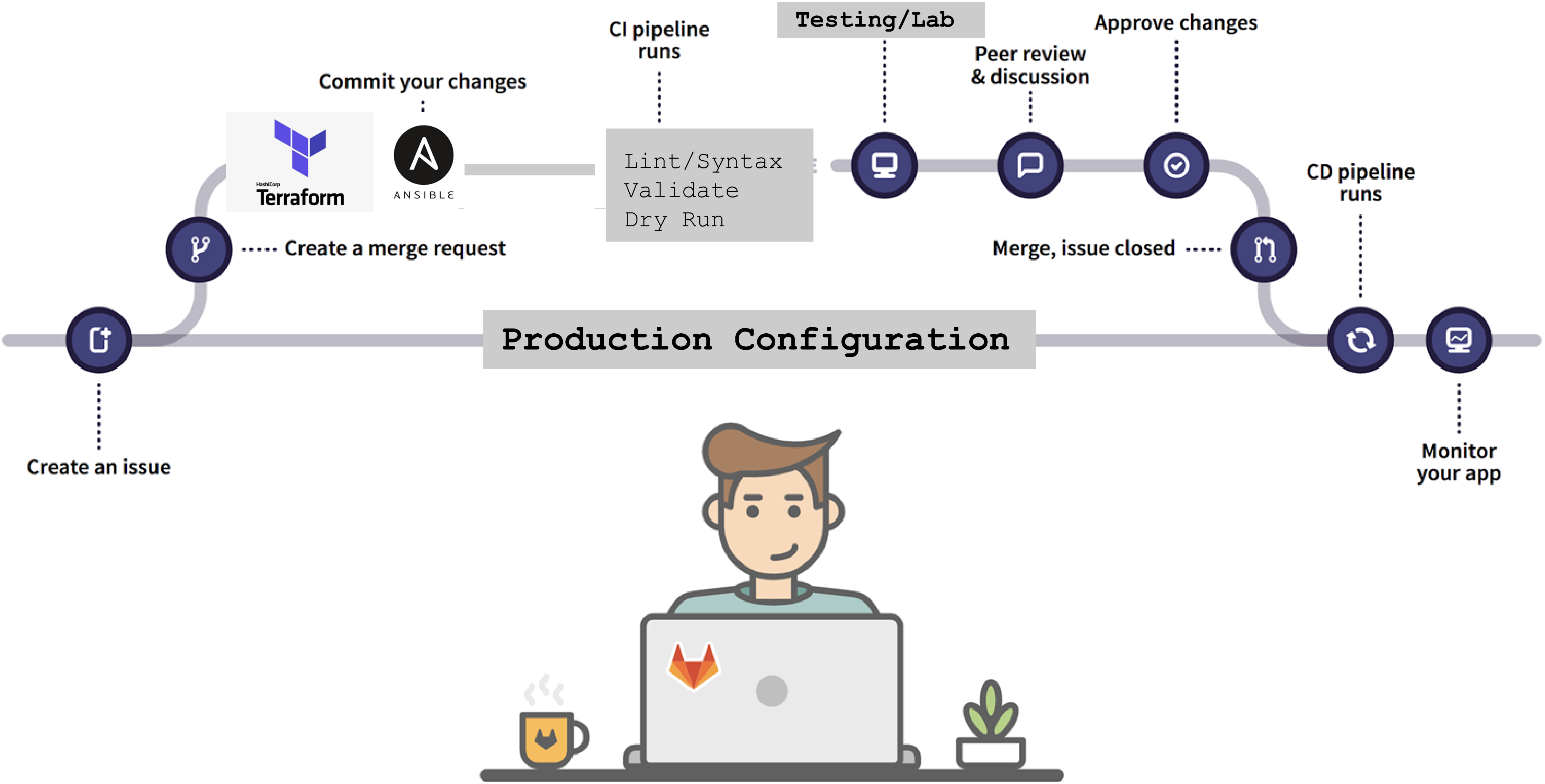
Configure

Monitor

Defend

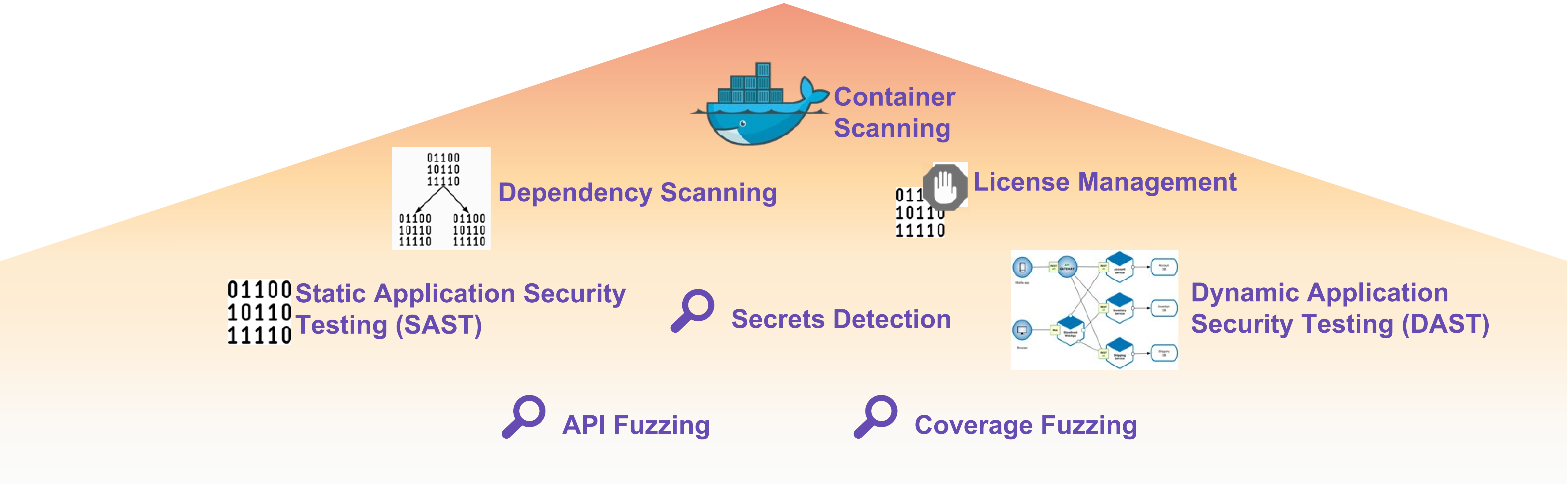
Automate

How GitLab does GitOps



GitLab Secure Capabilities–Continuous Application Security

Manage | Plan | Create | Verify | Package | **Secure** | Release | Configure | Monitor | Defend



Container Scanning

Dependency Scanning

License Management

Static Application Security Testing (SAST)

Secrets Detection

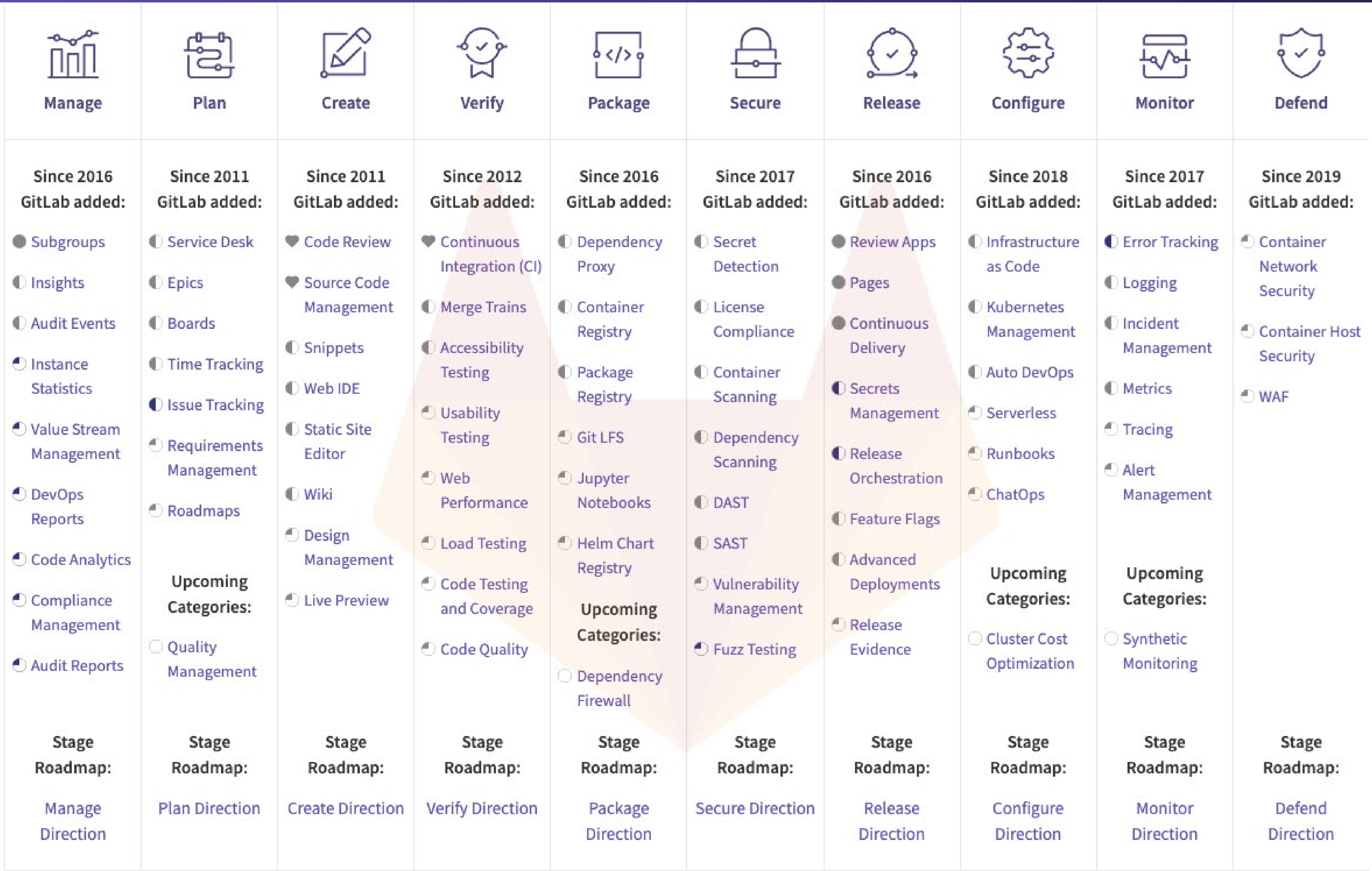
Dynamic Application Security Testing (DAST)

API Fuzzing

Coverage Fuzzing

Security Testing Built into Merge Requests

GitLab is the first single application for the entire DevOps lifecycle

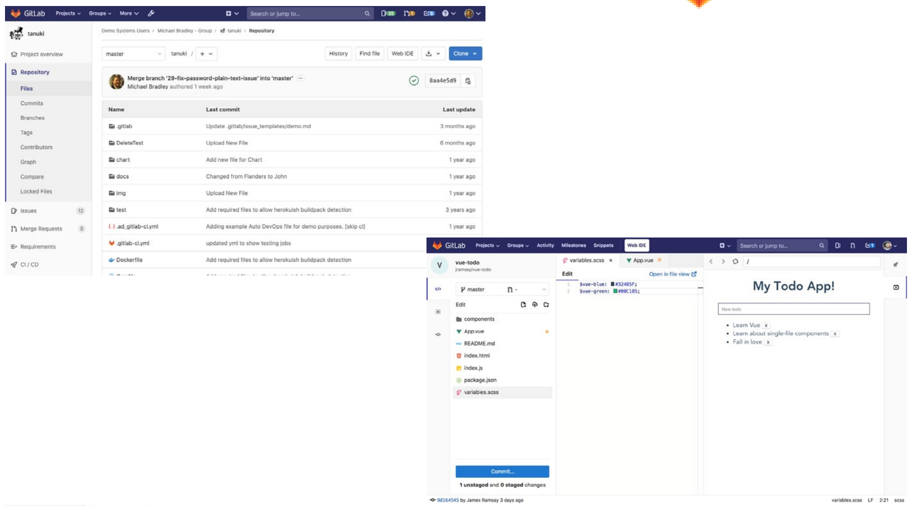


Manage	Plan	Create	Verify	Package	Secure	Release	Configure	Monitor	Defend
Since 2016 GitLab added:	Since 2011 GitLab added:	Since 2011 GitLab added:	Since 2012 GitLab added:	Since 2016 GitLab added:	Since 2017 GitLab added:	Since 2016 GitLab added:	Since 2018 GitLab added:	Since 2017 GitLab added:	Since 2019 GitLab added:
● Subgroups ◐ Insights ◐ Audit Events ◔ Instance Statistics ◔ Value Stream Management ◔ DevOps Reports ◔ Code Analytics ◔ Compliance Management ◔ Audit Reports	◐ Service Desk ◐ Epics ◐ Boards ◐ Time Tracking ◐ Issue Tracking ◔ Requirements Management ◔ Roadmaps	♥ Code Review ♥ Source Code Management ◐ Snippets ◐ Web IDE ◐ Static Site Editor ◐ Wiki ◔ Design Management ◔ Live Preview	♥ Continuous Integration (CI) ◐ Merge Trains ◐ Accessibility Testing ◔ Usability Testing ◔ Web Performance ◔ Load Testing ◔ Code Testing and Coverage ◔ Code Quality	◐ Dependency Proxy ◐ Container Registry ◐ Package Registry ◔ Git LFS ◔ Jupyter Notebooks ◔ Helm Chart Registry	◐ Secret Detection ◐ License Compliance ◐ Container Scanning ◐ Dependency Scanning ◐ DAST ◐ SAST ◔ Vulnerability Management ◔ Fuzz Testing	● Review Apps ● Pages ● Continuous Delivery ◐ Secrets Management ◐ Release Orchestration ◐ Feature Flags ◐ Advanced Deployments ◔ Release Evidence	◐ Infrastructure as Code ◐ Kubernetes Management ◐ Auto DevOps ◔ Serverless ◔ Runbooks ◔ ChatOps	◐ Error Tracking ◐ Logging ◐ Incident Management ◐ Metrics ◔ Tracing ◔ Alert Management	◔ Container Network Security ◔ Container Host Security ◔ WAF
	Upcoming Categories: ○ Quality Management			Upcoming Categories: ○ Dependency Firewall			Upcoming Categories: ○ Cluster Cost Optimization	Upcoming Categories: ○ Synthetic Monitoring	
Stage Roadmap: Manage Direction	Stage Roadmap: Plan Direction	Stage Roadmap: Create Direction	Stage Roadmap: Verify Direction	Stage Roadmap: Package Direction	Stage Roadmap: Secure Direction	Stage Roadmap: Release Direction	Stage Roadmap: Configure Direction	Stage Roadmap: Monitor Direction	Stage Roadmap: Defend Direction

Code Repository

Source code management is where development team sharing and collaboration begins. GitLab is a Git-based fully integrated platform for software development.

- **Collaborate**
  - Review, comment, and improve each other’s code
  - Share code, enable re-use and ‘innersourcing’.
  - File locking prevents conflicts
  - Included WebIDE enables development on any platform
- **Accelerate**
  - Git base repository, enabling developers to work from their local copy of the code
  - Branch code, make changes and then quickly merge code
- **Compliant & Secure**
  - Review, track and approve code changes with powerful merge requests
  - Automatically scan for code quality and security with every commit.
  - Simplify auditing and compliance with granular access controls and reporting



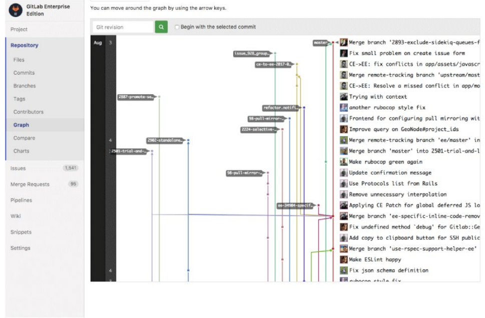
Merge branch '29-fix-password-plain-text-issue' into 'master'
Michael Bradley authored 1 week ago 8aa4e5d9

Name	Last commit	Last update
.gitlab	Update .gitlab/issue_templates/demo.md	3 months ago
DeleteTest	Upload New File	6 months ago
chart	Add new file for Chart	1 year ago
docs	Changed from Flanders to John	1 year ago
img	Upload New File	1 year ago
test	Add required files to allow herokuish buildpack detection	3 years ago
.ad_gitlab-ci.yml	Adding example Auto DevOps file for demo purposes. [skip ci]	1 year ago
.gitlab-ci.yml	updated yml to show testing jobs	1 year ago
Dockerfile	Add required files to allow herokuish buildpack detection	

```
$vue-blue: #32485F;
$vue-green: #00C185;
```

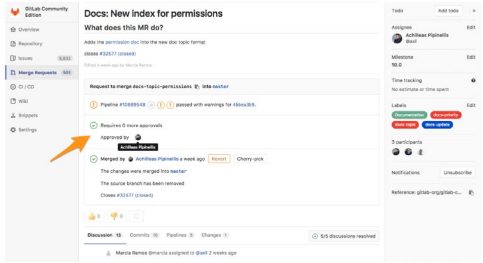
1 unstaged and 0 staged changes

9d164545 by James Ramsay 3 days ago



You can move around the graph by using the arrow keys.

```
Merge branch '2893-exclude-sidekiq-queues-f
Fix small problem on create issue form
CE->EE: fix conflicts in app/assets/javascr
Merge remote-tracking branch 'upstream/mast
CE->EE: Resolve a missed conflict in app/mo
Trying with context
another rubocop style fix
Frontend for configuring pull mirroring wit
Improve query on GeoNode#project_ids
Merge remote-tracking branch 'ee/master' in
Merge branch 'master' into 2501-trial-and-l
Make rubocop green again
Update confirmation message
Use Protocols list from Rails
Remove unnecessary interpolation
Applying CE Patch for global deferred JS lo
Merge branch 'ee-specific-inline-code-remov
Fix undefined method `debug' for Gitlab::Ge
Add copy to clipboard button for SSH public
Merge branch 'use-rspec-support-helper-ee'
Make ESLint happy
Fix json schema definition
```



Docs: New index for permissions

What does this MR do?

Adds the permission doc into the new doc topic format

closes #32577 (closed)

Edited a week ago by Marcia Ramos

Request to merge docs-topic-permissions into master

Pipeline #10889548 passed with warnings for 4bbea3b9.

Requires 0 more approvals

Approved by Achilleas Pipinellis

Merged by Achilleas Pipinellis a week ago

The changes were merged into master

The source branch has been removed

Closes #32577 (closed)

5/5 discussions resolved

Marcia Ramos @marcia assigned to @axil 2 weeks ago

Assignee: Achilleas Pipinellis @axil

Milestone: 10.0

Time tracking: No estimate or time spent

Labels: Documentation, docs-priority, docs-topic, docs-update

3 participants

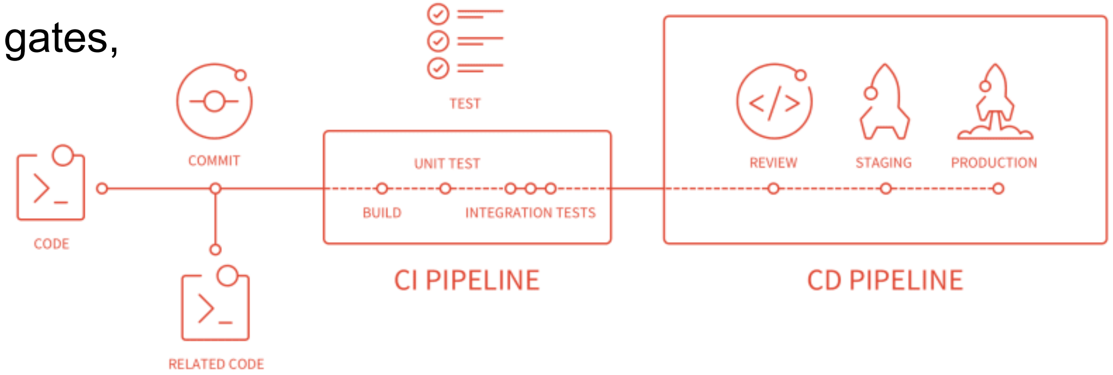
Notifications: Unsubscribe

Reference: gitlab-org/gitlab-c...

CI/CD

GitLab Continuous Integration (CI) & Continuous Delivery / Deployment (CD).

- **Advantages of GitLab CI/CD**
  - **Integrated**: CI/CD is part of GitLab, enabling a single conversation from planning to deployment.
  - **Open source**: CI/CD is a part of both the open source GitLab Community Edition and the proprietary GitLab Enterprise Edition.
  - **Easy to learn**: Well **documented** with **examples** and our **Quick Start** guide.
  - **Seamless**: Single great UI/UX for your team.
  - **Scalable**: Test run distributed on separate machines of which you can add as many as you want.
  - **Faster results**: Each build can be split into multiple jobs that run in parallel or multiple machines.
  - **Optimized for delivery**: Multiple stages, manual deploy gates, **environments**, and **variables**.

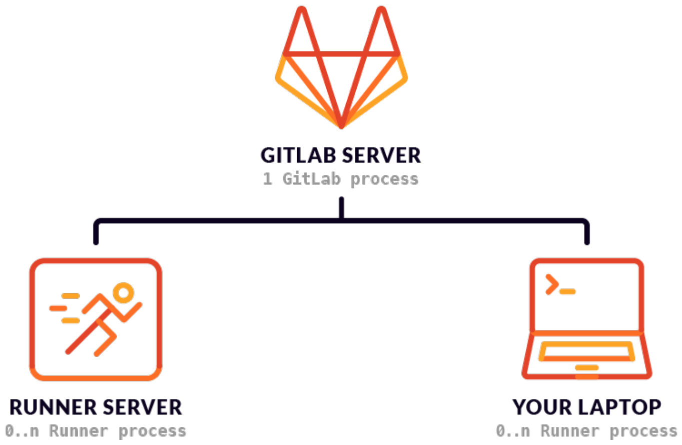


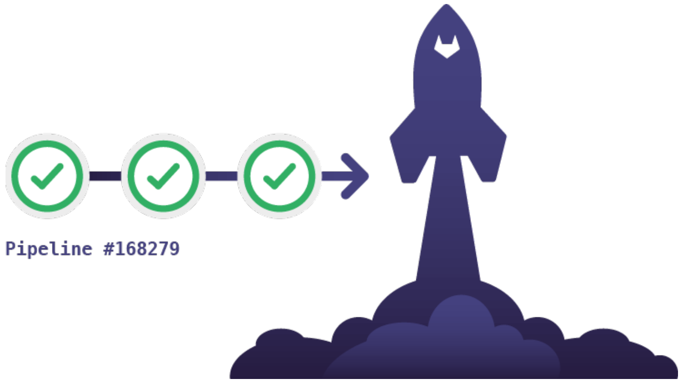
CI/CD

GitLab Continuous Integration (CI) & Continuous Delivery (CD).

- One Application for the entire DevSecOps lifecycle
  - Built your app using GitLab Runners
    - **GitLab Runner** is an application which processes builds. It can be deployed separately and works with GitLab CI/CD through an API.
      - Works on any platform the can build Go binaries including Linux, macOS, Windows, FreeBSD and Docker.
      - Test programming languages .Net, Java, Python, C, PHP, and others.
      - Feature rich with Autoscaling, great Docker support, run multiple jobs concurrently.
  - Run unit and integration tests to check that code is valid
  - Live preview of development branches with Review Apps before merging into stable.
  - Deploy to multiple environments like staging and production, and support advanced features such as a **canary deployments**.
  - Monitor performances and status of your application
  - Built for Cloud Native with Kubernetes integration
  - See the status of each build within the Merge Request

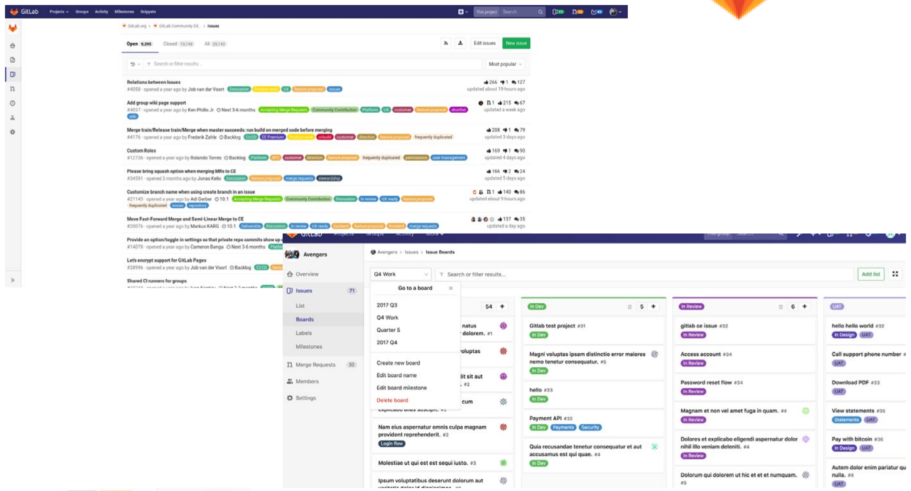
CREATE
PLAN
DEV
VERIFY
PACKAGE
RELEASE
CONFIGURE
OPS
MONITOR



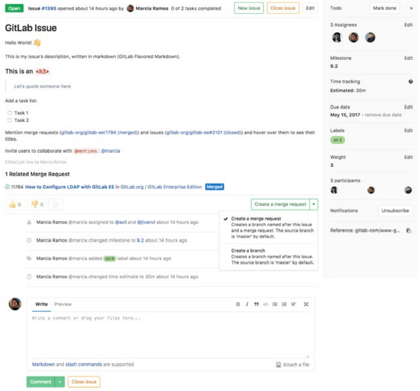


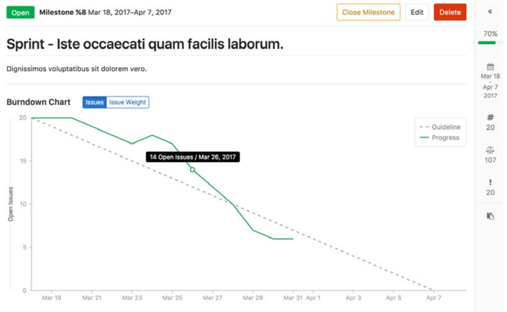
Issue Management

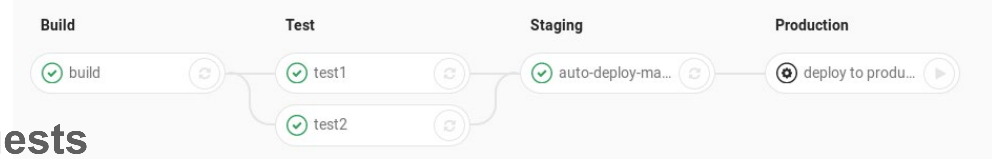
Plan and manage projects. GitLab enables lean and agile project management from basic issue tracking to Scrum and Kanban style project management that scales from small teams to large complex organizations.



- Track & manage issues
  - Collaborate and define specific business needs with **Issues / User Stories**.
  - Track ownership, effort, size, complexity and priority of resolution.
  - Eliminate silos and enable cross-functional engagement.
- Agile project management
  - Manage sprints with **Milestones** and **Burndown Charts**.
  - Track your backlog with issue lists and **Labels** for filtering prioritization.
- Visualize work with **Issue Boards**
  - Visualize the status of work across the lifecycle.
  - Manage, assign and track the flow of work.
  - Enable Kanban and Scrum styles of agile delivery
- DevOps pipeline traceability
  - Link issues with actual code changes in **merge requests**
  - Visualize and track the status of builds, testing, security scans, and delivery



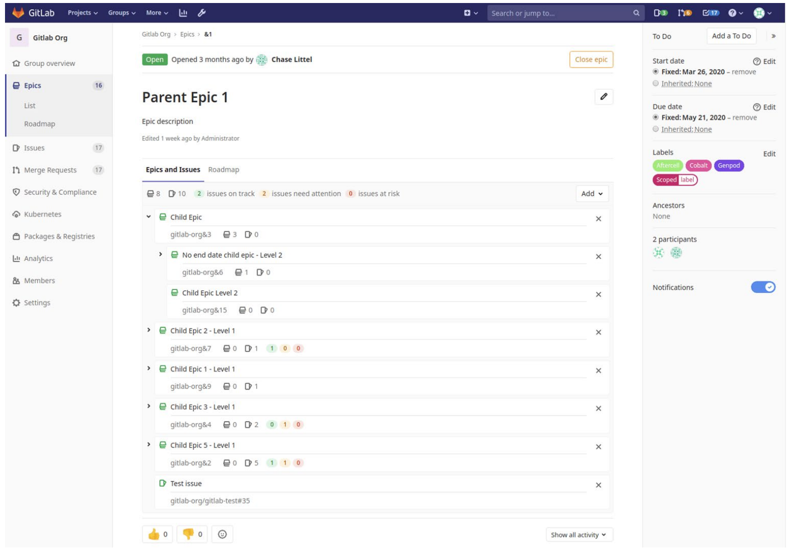




Issue Management

Agile Portfolio Management. GitLab helps you manage and govern portfolios of agile projects.

- Plan future work with Epics
  - Organize new initiatives and efforts into **Epics**
  - Plan sub epics and issues into sprints and milestone
- Roadmaps visualize value delivery
  - Prioritize and visualize sequence of delivery with **Roadmaps**.
  - Communicate plans, timing, and strategic flow
  - Maintain visibility from strategic plans to execution
- GitLab uses GitLab to build and deliver GitLab
  - See how we do it with our public **GitLab.org project**.



Gitlab Org › Epics › &1

Open Opened 3 months ago by Chase Littel

Parent Epic 1

Epic description

Edited 1 week ago by Administrator

8 10 2 issues on track 2 issues need attention 0 issues at risk

- Child Epic — gitlab-org&3 3 0
  - No end date child epic - Level 2 — gitlab-org&6 1 0
  - Child Epic Level 2 — gitlab-org&15 0 0
- Child Epic 2 - Level 1 — gitlab-org&7 0 1 1 0 0
- Child Epic 1 - Level 1 — gitlab-org&9 0 1
- Child Epic 3 - Level 1 — gitlab-org&4 0 2 0 1 0
- Child Epic 5 - Level 1 — gitlab-org&2 0 5 1 1 0
- Test issue — gitlab-org/gitlab-test#35

👍 0 👎 0

Start date
● Fixed: Mar 26, 2020 – remove
○ Inherited: None

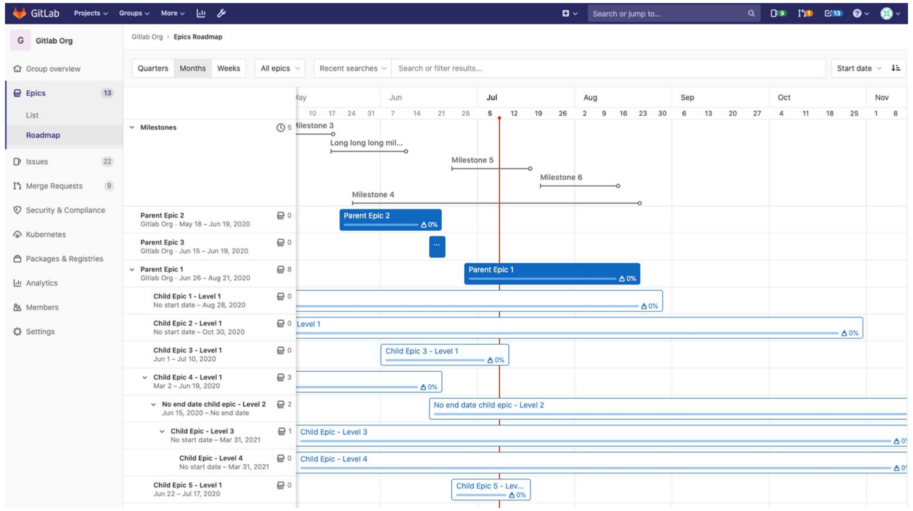
Due date
● Fixed: May 21, 2020 – remove
○ Inherited: None

Labels: Aftercell Cobalt Genpod Scoped label

Ancestors: None

2 participants

Notifications



Gitlab Org › Epics Roadmap

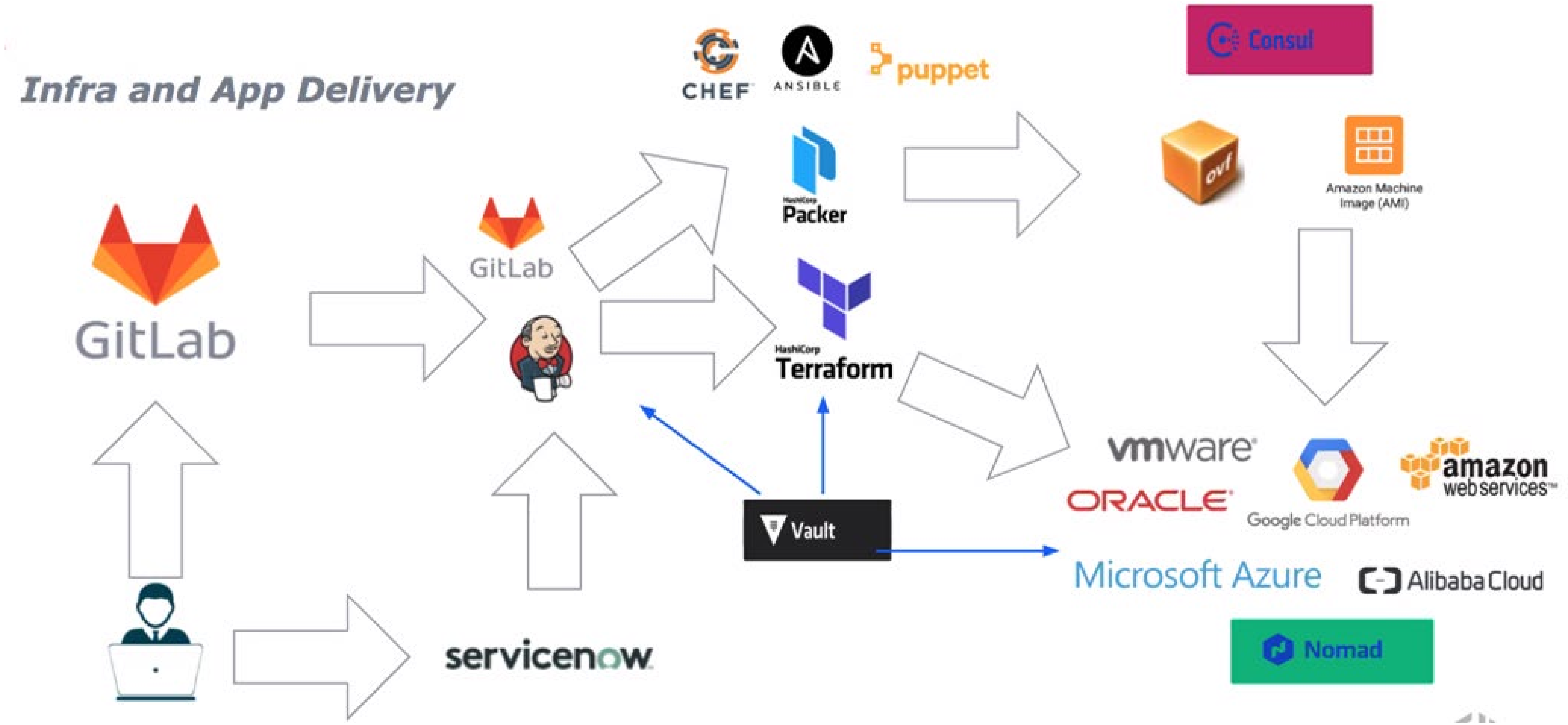
Milestones 5

Parent Epic 2 Gitlab Org · May 18 – Jun 19, 2020	0
Parent Epic 3 Gitlab Org · Jun 15 – Jun 19, 2020	0
Parent Epic 1 Gitlab Org · Jun 26 – Aug 21, 2020	8
Child Epic 1 - Level 1 No start date – Aug 28, 2020	0
Child Epic 2 - Level 1 No start date – Oct 30, 2020	0
Child Epic 3 - Level 1 Jun 1 – Jul 10, 2020	0
Child Epic 4 - Level 1 Mar 2 – Jun 19, 2020	3
No end date child epic - Level 2 Jun 15, 2020 – No end date	2
Child Epic - Level 3 No start date – Mar 31, 2021	1
Child Epic - Level 4 No start date – Mar 31, 2021	0
Child Epic 5 - Level 1 Jun 22 – Jul 17, 2020	0

GitLab

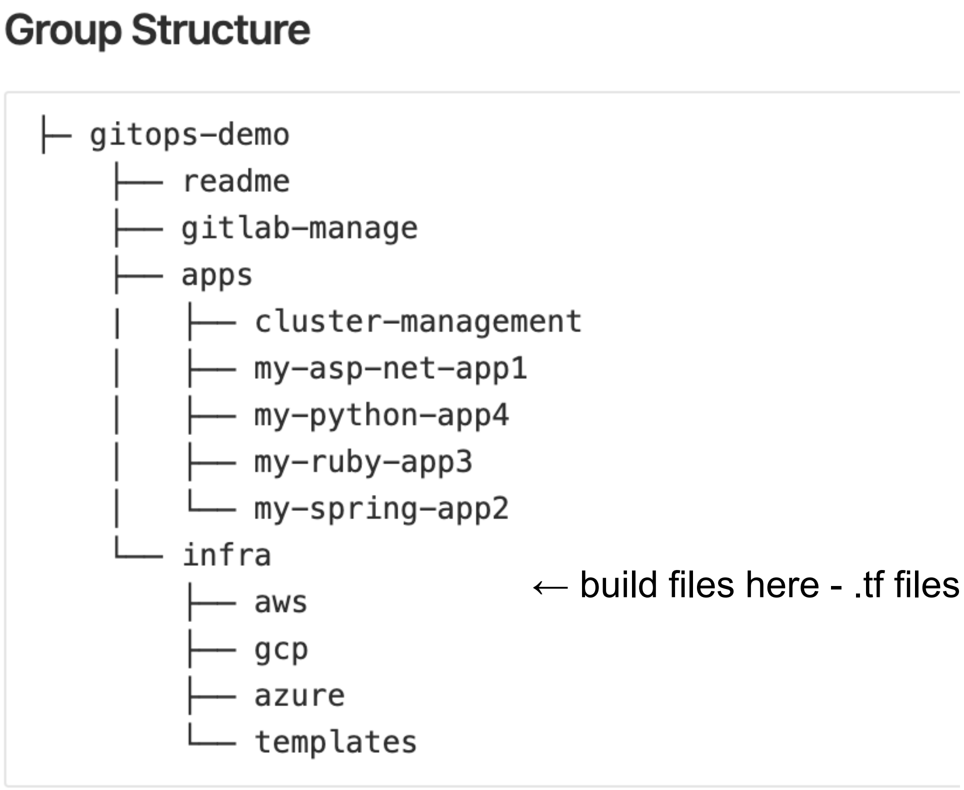
Use Cases

Cloud Operating Model - Automated at Scale



GitLab Architecture Sample

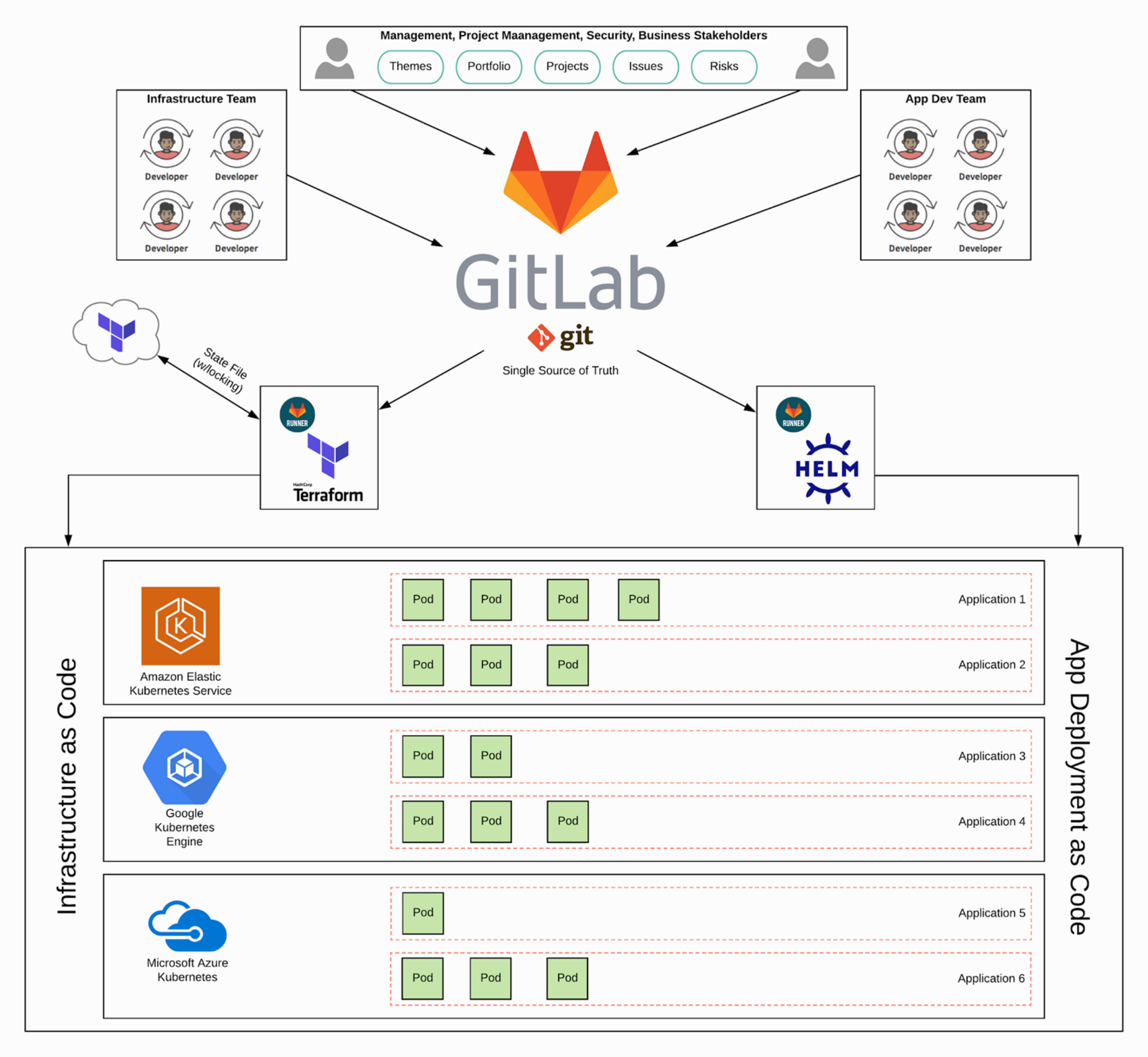
1. Terraform setups K8s clusters in three clouds and registers to GitLab Group
2. New managed-apps install via CI kicks off
3. AutoDevOps deploys apps via CI/CD
4. App “migration” by updating App Environment



Group Structure

```
├ gitops-demo
    ├── readme
    ├── gitlab-manage
    ├── apps
    |    ├── cluster-management
    |    ├── my-asp-net-app1
    |    ├── my-python-app4
    |    ├── my-ruby-app3
    |    └── my-spring-app2
    └── infra
         ├── aws
         ├── gcp
         ├── azure
         └── templates
```

← build files here - .tf files



GitLab

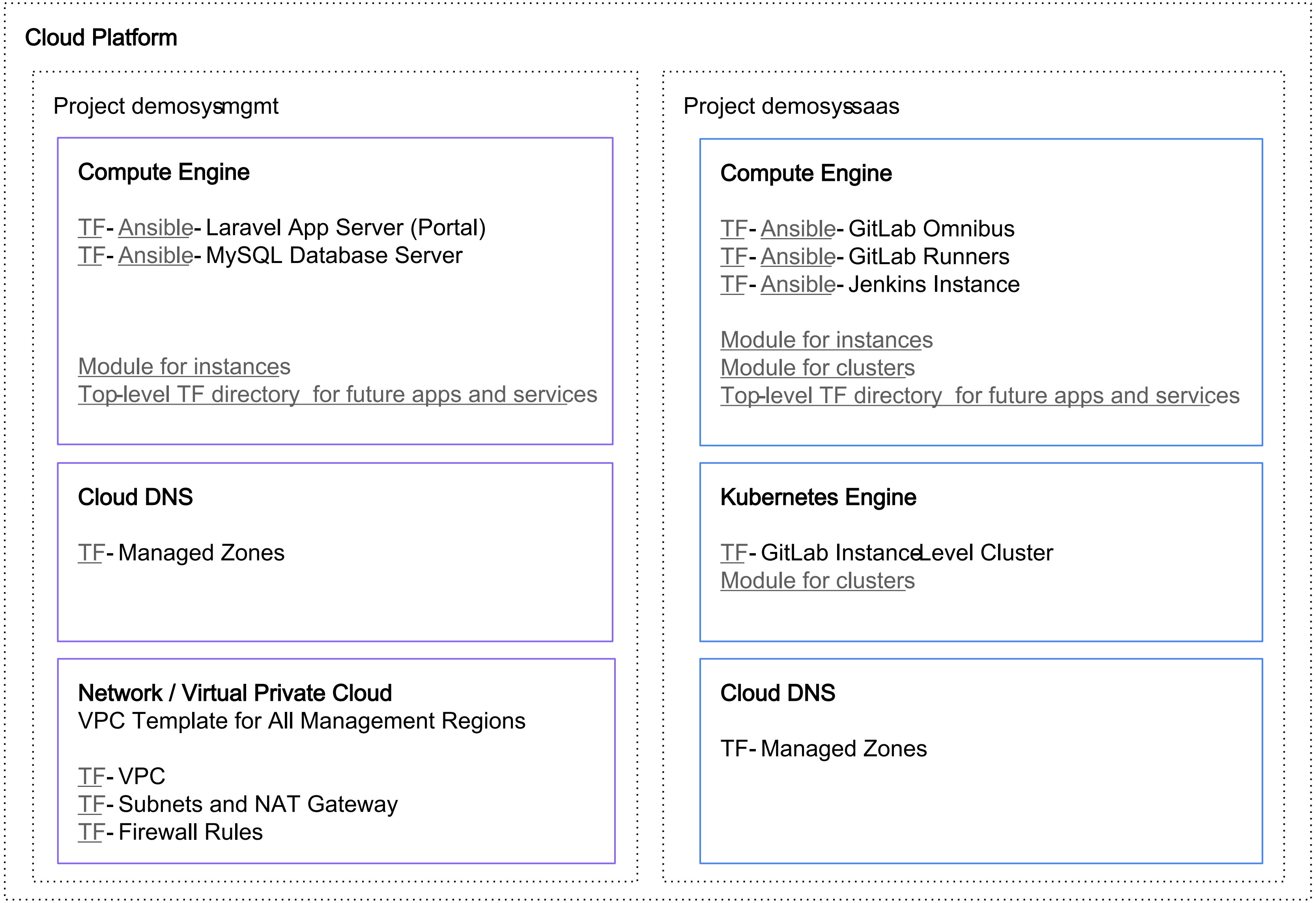
GitLab Demo Systems

GitLab Demo Systems

GitLab repository for playbooks / configurations and CI pipeline. Issues and Merge requests to manage changes / iterations.

Terraform is used to create infrastructure

Ansible is used to install and configure software



Cloud Platform

Project demosysmgmt

Compute Engine

TF - Ansible - Laravel App Server (Portal)
TF - Ansible - MySQL Database Server

Module for instances
Top-level TF directory for future apps and services

Cloud DNS

TF - Managed Zones

Network / Virtual Private Cloud
VPC Template for All Management Regions

TF - VPC
TF - Subnets and NAT Gateway
TF - Firewall Rules

Project demosyssaas

Compute Engine

TF - Ansible - GitLab Omnibus
TF - Ansible - GitLab Runners
TF - Ansible - Jenkins Instance

Module for instances
Module for clusters
Top-level TF directory for future apps and services

Kubernetes Engine

TF - GitLab InstanceLevel Cluster
Module for clusters

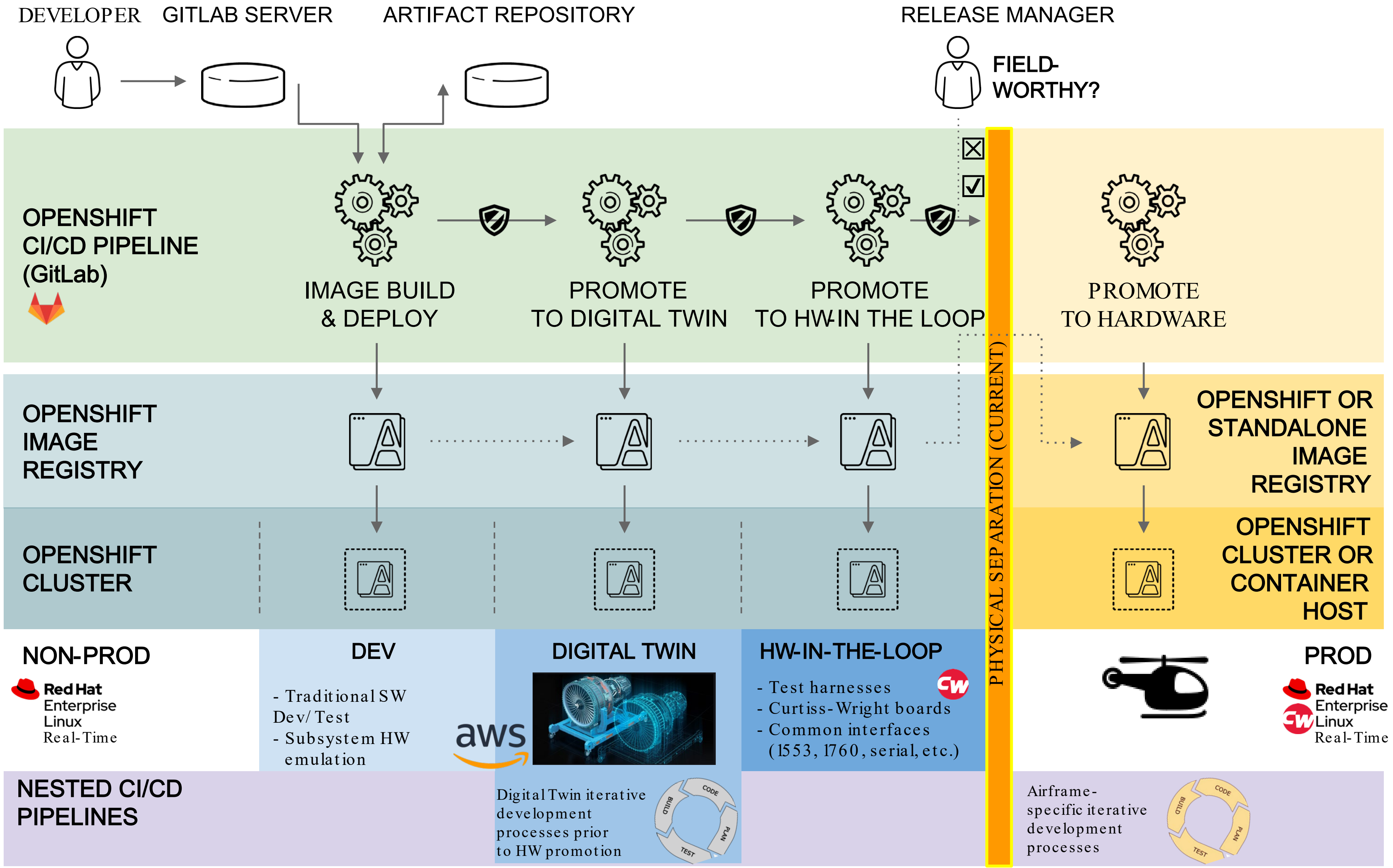
Cloud DNS

TF - Managed Zones

GitLab

Embedded Systems

Embedded DevSecOps Big Picture



DEVELOPER

GITLAB SERVER

ARTIFACT REPOSITORY

RELEASE MANAGER

FIELD-WORTHY?

☒

☑

OPENSHIFT CI/CD PIPELINE (GitLab)

IMAGE BUILD & DEPLOY

PROMOTE TO DIGITAL TWIN

PROMOTE TO HW-IN THE LOOP

PROMOTE TO HARDWARE

OPENSHIFT IMAGE REGISTRY

OPENSHIFT OR STANDALONE IMAGE REGISTRY

OPENSHIFT CLUSTER

OPENSHIFT CLUSTER OR CONTAINER HOST

PHYSICAL SEPARATION (CURRENT)

NON-PROD

Red Hat Enterprise Linux Real-Time

DEV

- Traditional SW Dev/Test
- Subsystem HW emulation

aws

DIGITAL TWIN

HW-IN-THE-LOOP

- Test harnesses
- Curtiss-Wright boards
- Common interfaces (1553, 1760, serial, etc.)

PROD

Red Hat Enterprise Linux Real-Time

NESTED CI/CD PIPELINES

Digital Twin iterative development processes prior to HW promotion

CODE

PLAN

TEST

BUILD

Airframe-specific iterative development processes

CODE

PLAN

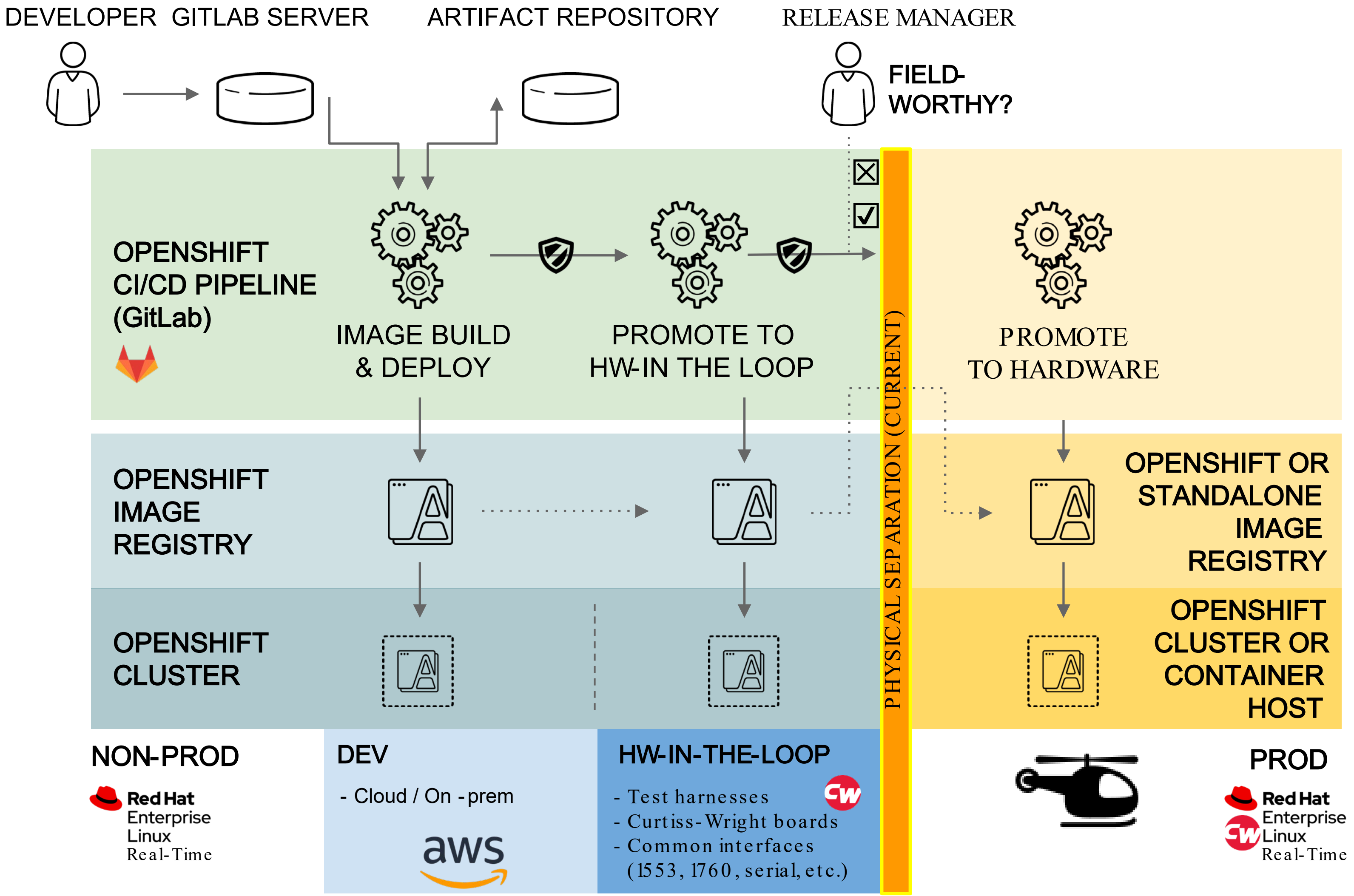
TEST

BUILD

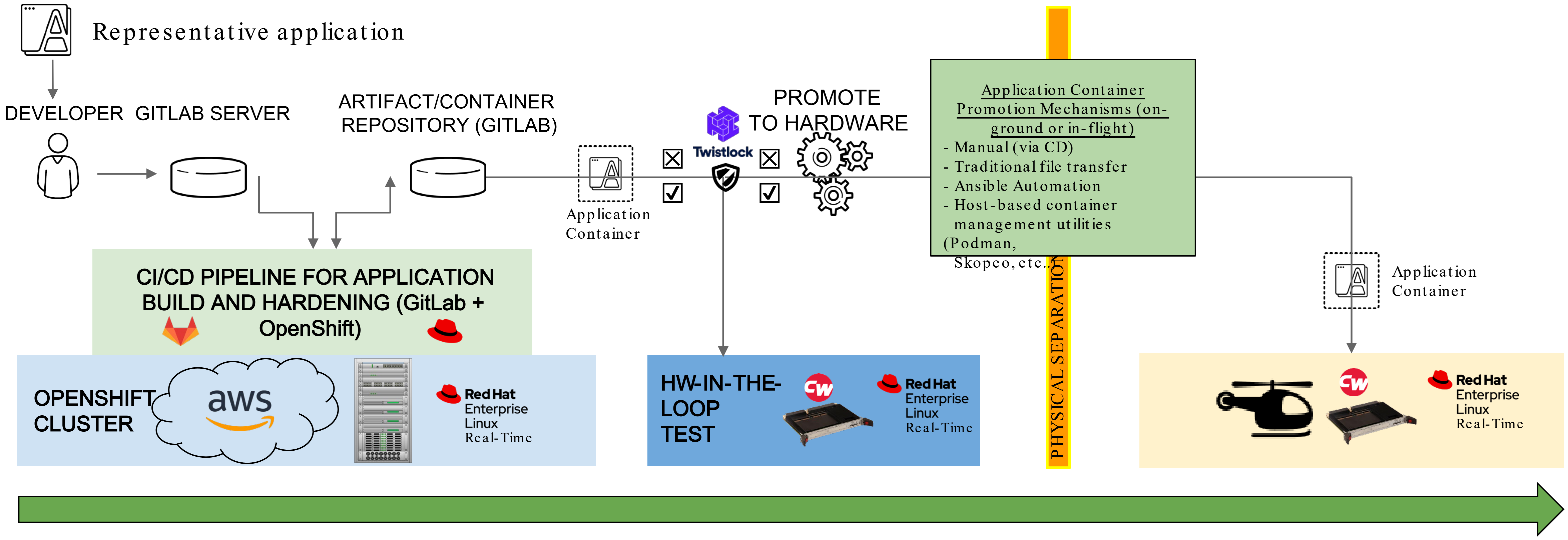
Red Hat

Embedded DevSecOps POC Scope

- Scope is shifted towards the hardware, beginning with an existing application.
- Well-established and proven DEV pipeline stage ensures the quality and security of the application container prior to deployment per DoD DSOP guidance.


DEVELOPER
GITLAB SERVER
ARTIFACT REPOSITORY
RELEASE MANAGER
FIELD-WORTHY?
☒
☑
OPENSHIFT CI/CD PIPELINE (GitLab)
IMAGE BUILD & DEPLOY
PROMOTE TO HW-IN THE LOOP
PROMOTE TO HARDWARE
OPENSHIFT IMAGE REGISTRY
OPENSHIFT CLUSTER
PHYSICAL SEPARATION (CURRENT)
OPENSHIFT OR STANDALONE IMAGE REGISTRY
OPENSHIFT CLUSTER OR CONTAINER HOST
NON-PROD
Red Hat Enterprise Linux Real-Time
DEV
- Cloud / On -prem
aws
HW-IN-THE-LOOP
- Test harnesses
- Curtiss-Wright boards
- Common interfaces (1553, 1760, serial, etc.)
PROD
Red Hat Enterprise Linux Real-Time

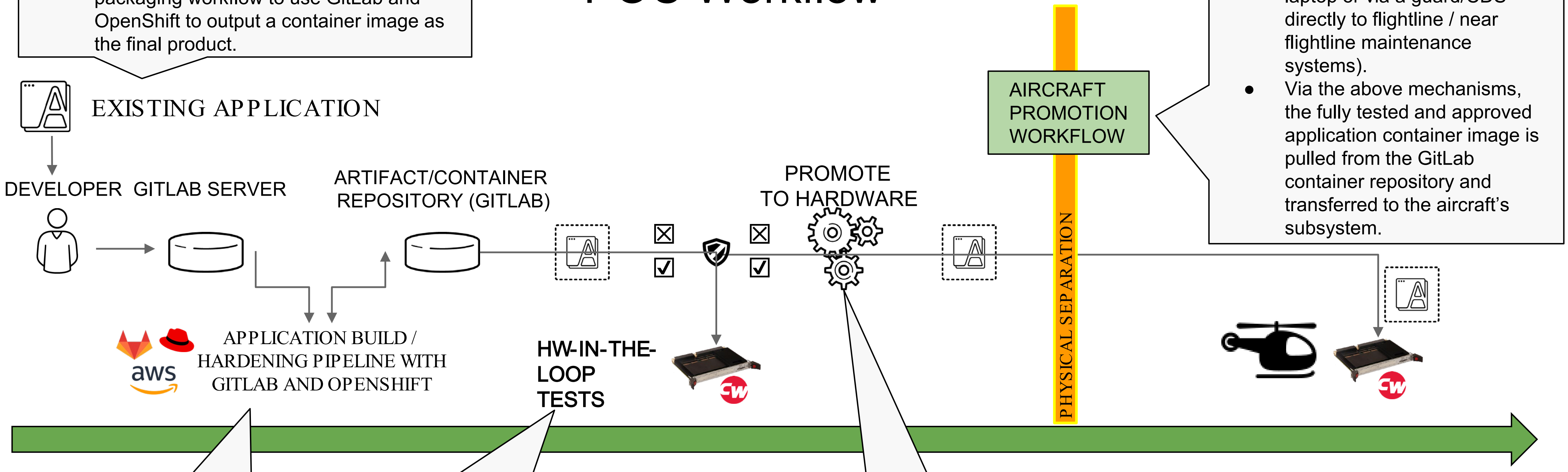
Embedded DevSecOps - POC Components


Representative application
DEVELOPER
GITLAB SERVER
ARTIFACT/CONTAINER REPOSITORY (GITLAB)
Application Container
Twistlock
☒ ☑
☒ ☑
PROMOTE TO HARDWARE
Application Container Promotion Mechanisms (on-ground or in-flight)
- Manual (via CD)
- Traditional file transfer
- Ansible Automation
- Host-based container management utilities (Podman, Skopeo, etc..)
PHYSICAL SEPARATION
Application Container
CI/CD PIPELINE FOR APPLICATION BUILD AND HARDENING (GitLab + OpenShift)
OPENSHIFT CLUSTER
aws
Red Hat Enterprise Linux Real-Time
HW-IN-THE-LOOP TEST
CW
Red Hat Enterprise Linux Real-Time
CW
Red Hat Enterprise Linux Real-Time

Embedded DevSecOps POC Workflow

1

- Choose an existing application.
- Minimally adapt the existing development / packaging workflow to use GitLab and OpenShift to output a container image as the final product.


EXISTING APPLICATION
DEVELOPER
GITLAB SERVER
ARTIFACT/CONTAINER REPOSITORY (GITLAB)
aws
APPLICATION BUILD / HARDENING PIPELINE WITH GITLAB AND OPENSHIFT
☒
☑
HW-IN-THE-LOOP TESTS
☒
☑
PROMOTE TO HARDWARE
AIRCRAFT PROMOTION WORKFLOW
PHYSICAL SEPARATION

2

- Developers use GitLab and OpenShift running in AWS to create a new version of the application by modifying code, and executing a build / hardening / testing pipeline
- This results in a container image of that application deposited in the GitLab container repository.

3

- Once the application container is built and passes virtual/emulation testing, the HW-in-the-loop environment is triggered to pull the application container image from the GitLab container repository and run it (via Ansible, Podman, Skopeo, etc..).
- Test suites are run against the application container from a GitLab pipeline to confirm functionality on the target hardware.

4

If the HW tests pass and the designated approver deems the application ready for production, an “aircraft promotion” workflow is initiated.

5

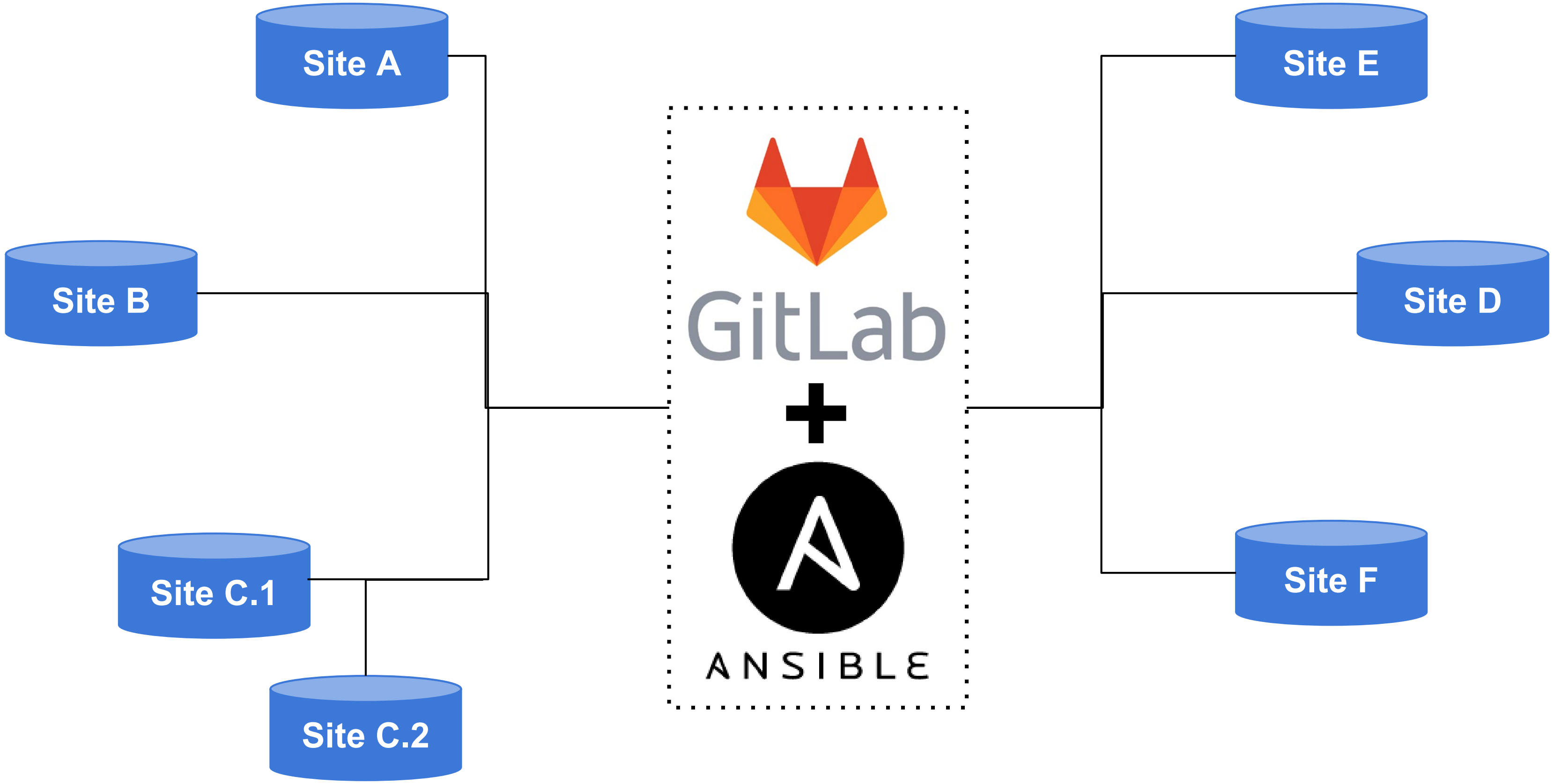
- Aircraft promotion workflow of the application container can be as manual or as automated as is feasible (container image can be written to a CD, secure laptop or via a guard/CDS directly to flightline / near flightline maintenance systems).
- Via the above mechanisms, the fully tested and approved application container image is pulled from the GitLab container repository and transferred to the aircraft’s subsystem.

Red Hat

GitLab

GitLab & Ansible
Manage Routers & Switches

GitLab & Ansible - Configuration Management



Q&A