Red Hat Open Tour 2022

Gain robust repeatability as self service, by automating the automation

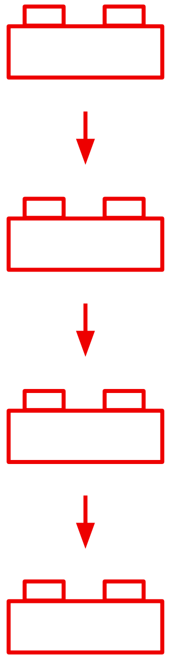
Speaker:

Solution Architect

In this presentation i am going to talk about

- Standardisation
- Automation
- Collaboration

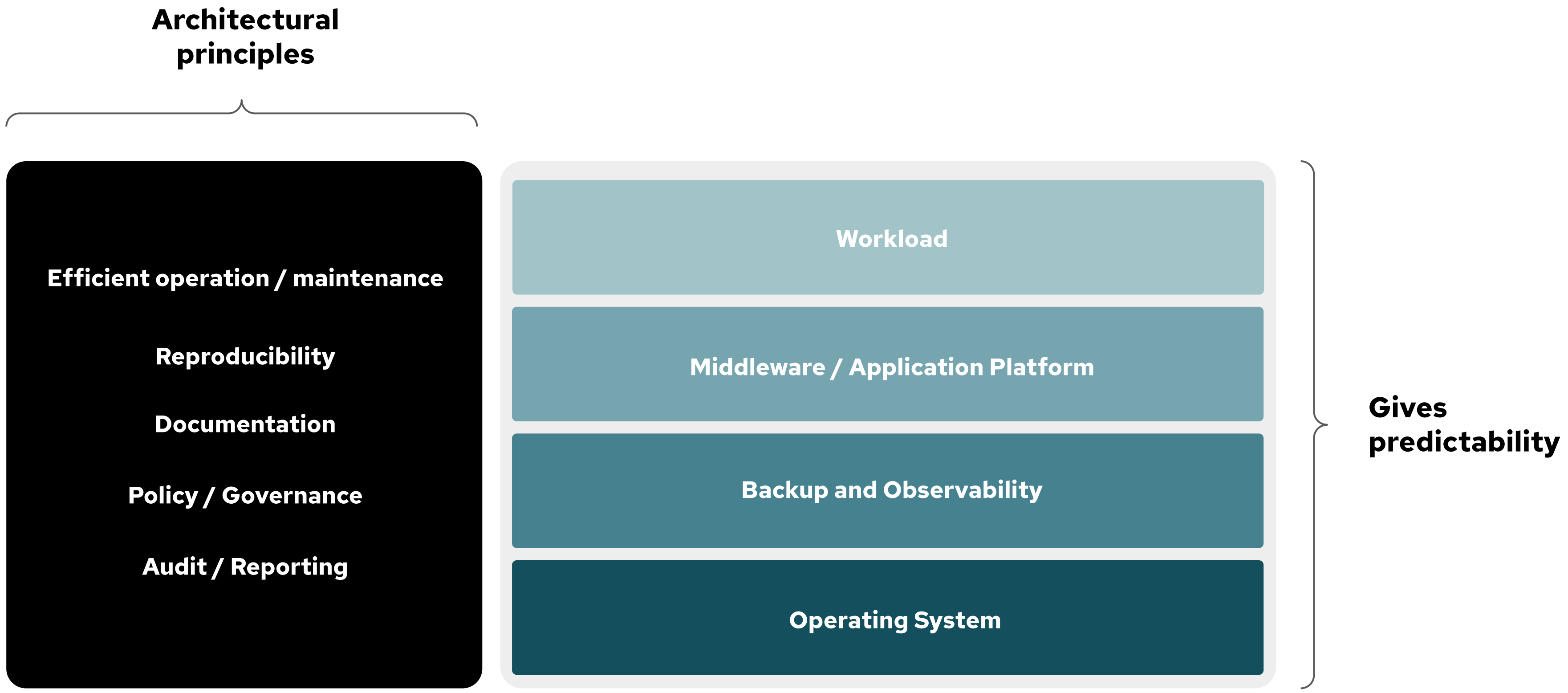
It all starts with standardisation



Picture credit to Association for Advancing Automation (A3)

Standardisation in IT

Strategic Viewpoint



Standard Operating Environment (SOE)

Definition:

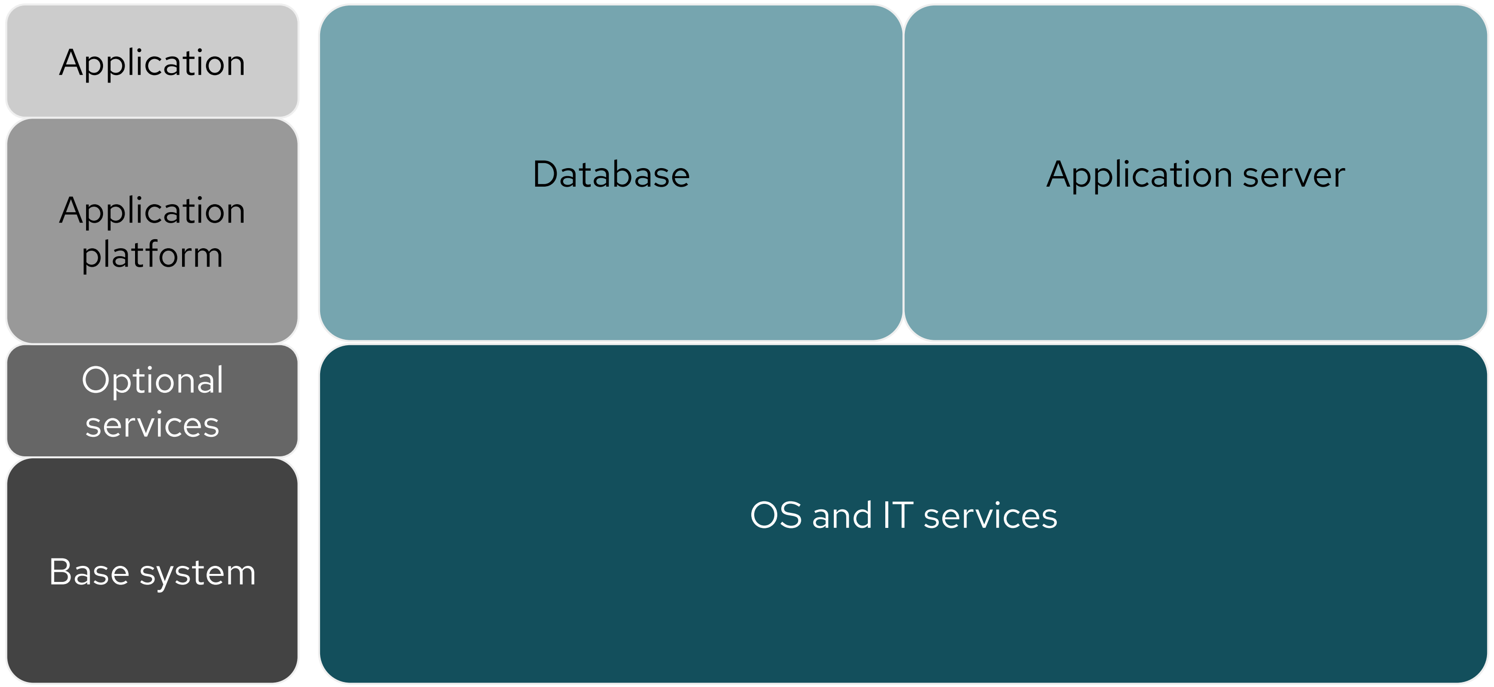
“Provides tools, standards and best practices to manage the lifecycle of an entire, deployed stack – from operating system and infrastructure services through to middleware and applications.”

What areas does it focus on?

- Automation
- Standardisation
- Lifecycle management
- Reporting

Standard Operating Environment (SOE)

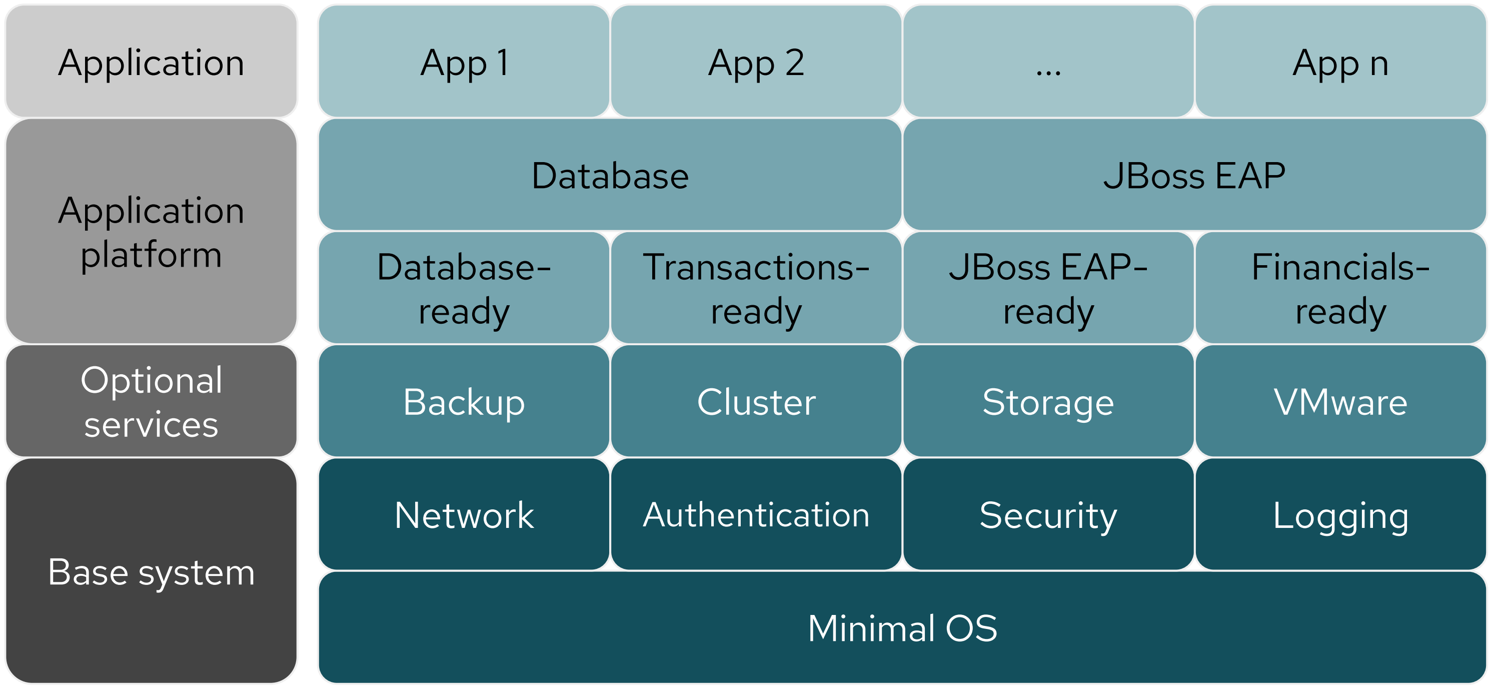
One-size-fits-most, generic servers with functional application blocks



Basic approach

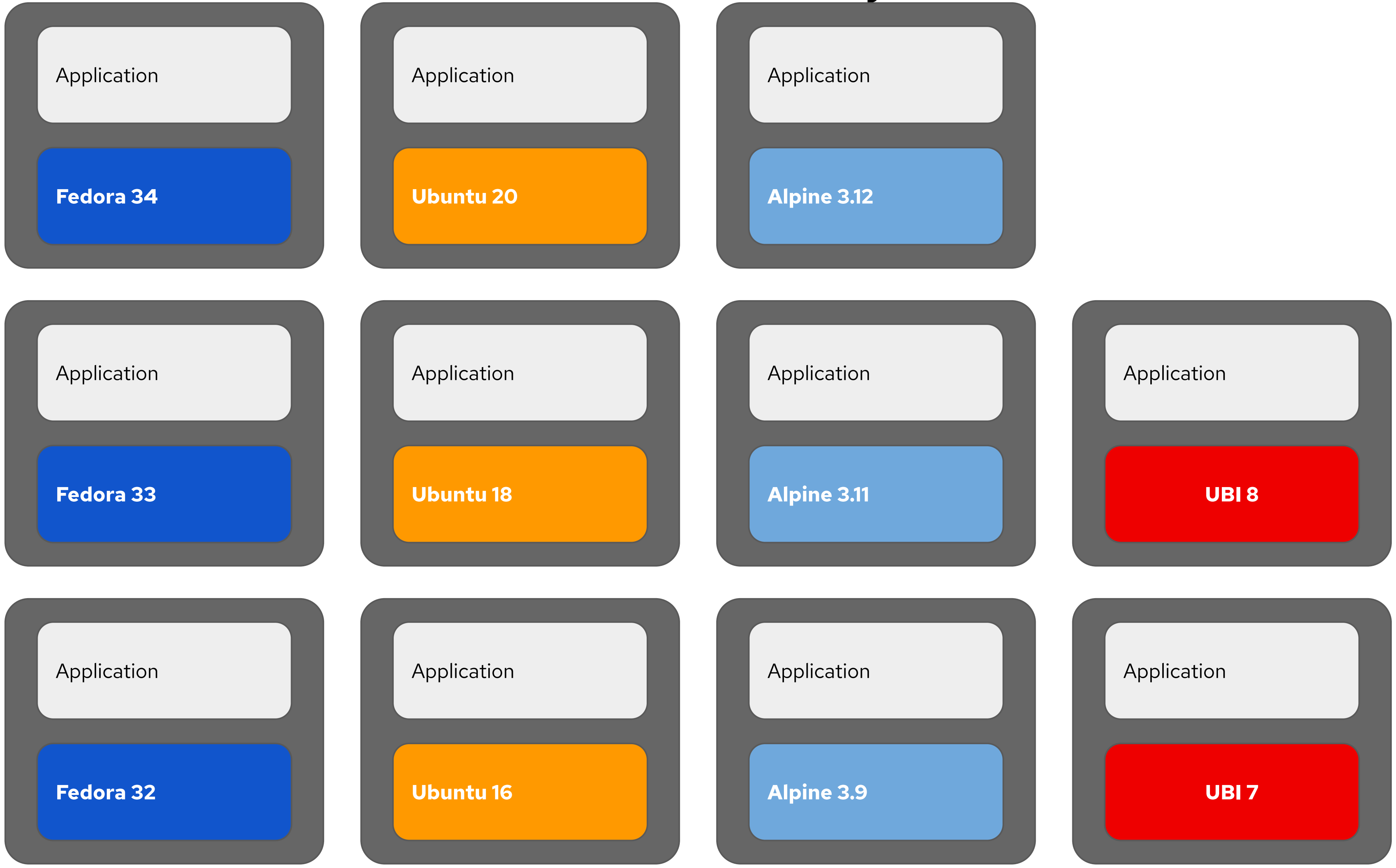
Standard Operating Environment (SOE)

Concept: Independent yet compatible and interchangeable components



Adaptive SOE approach

What about containers? they need SOE's too

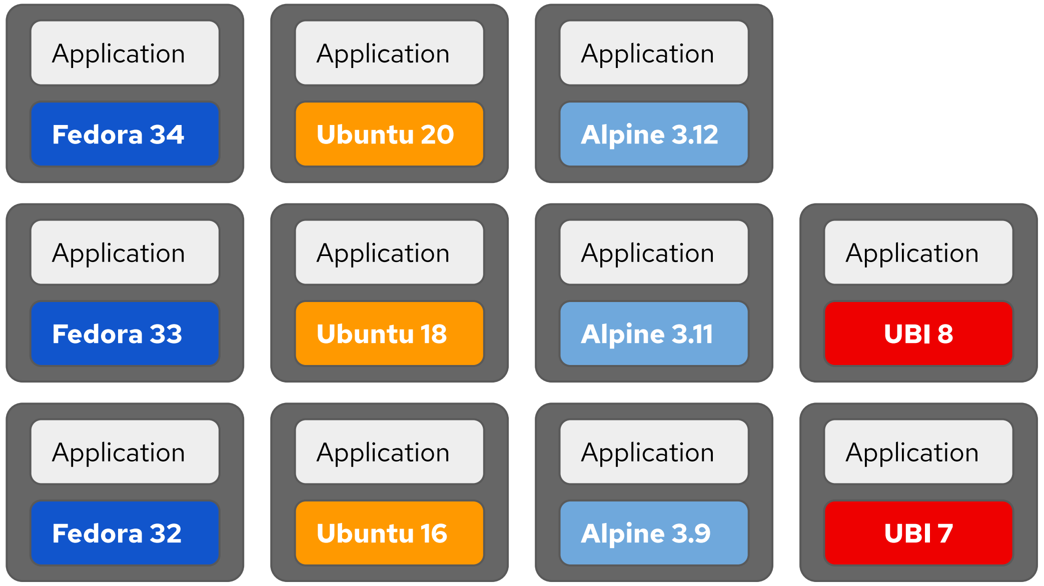


- 8 different versions of glibc
- 3 different versions of muslc
- 11 different versions of OpenSSL

With no standard

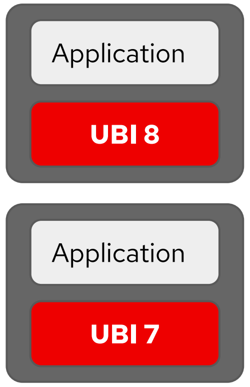
Everyone will make their own choice

No Standard Operating Environment



- 8 different versions of glibc
- 3 different versions of muslc
- 11 different versions of OpenSSL

Standard based on UBI 7 & UBI 8



- 2 different versions of glibc
- 2 different versions of OpenSSL

Efficiency Through Automation

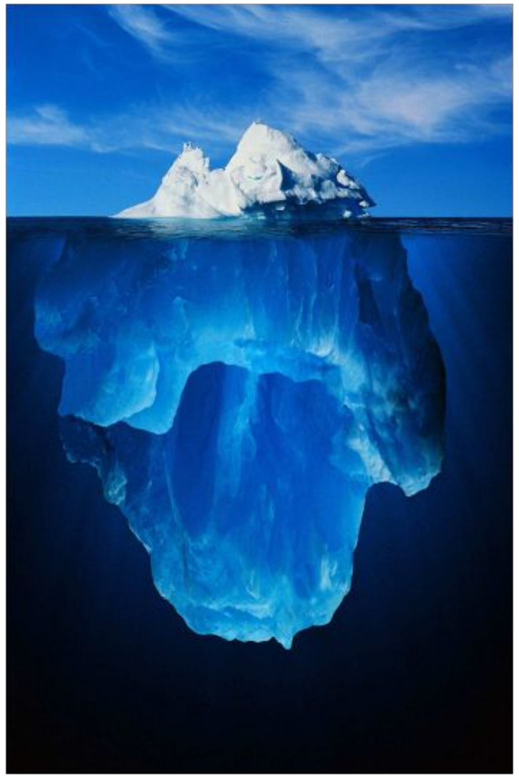
Ok, standards are great, but:

- only define point-in-time snapshots of the environment
- take time to maintain in a complex environment
- By automating the process of implementing standards we achieve:
- higher flexibility to accommodate change ⇒ higher agility
- higher operational efficiency
- eases lifecycle management

What kinds of automation?

IT Automation

your stack



Business value

Lots of tech

use cases

FULLY AUTOMATED PROVISIONING

SECURITY/ COMPLIANCE

CONFIG MANAGEMENT

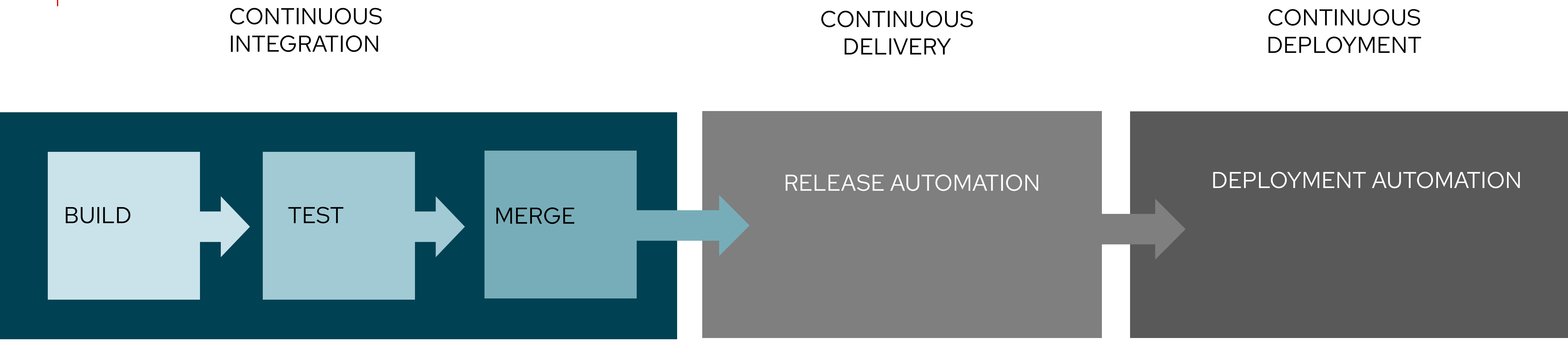
CONTINUOUS DELIVERY

ORCHESTRATION

APP DEPLOYMENT

Red Hat

Application Development and Deployment



Red Hat OpenShift

Red Hat Ansible Automation Platform

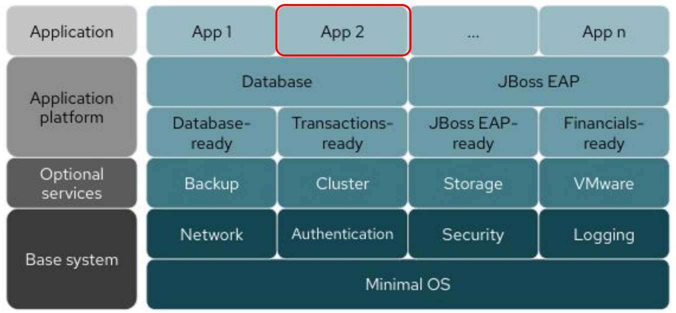
Git* & Ansible Automation Platform Application upgrade via CI/CD

In this demo you will see

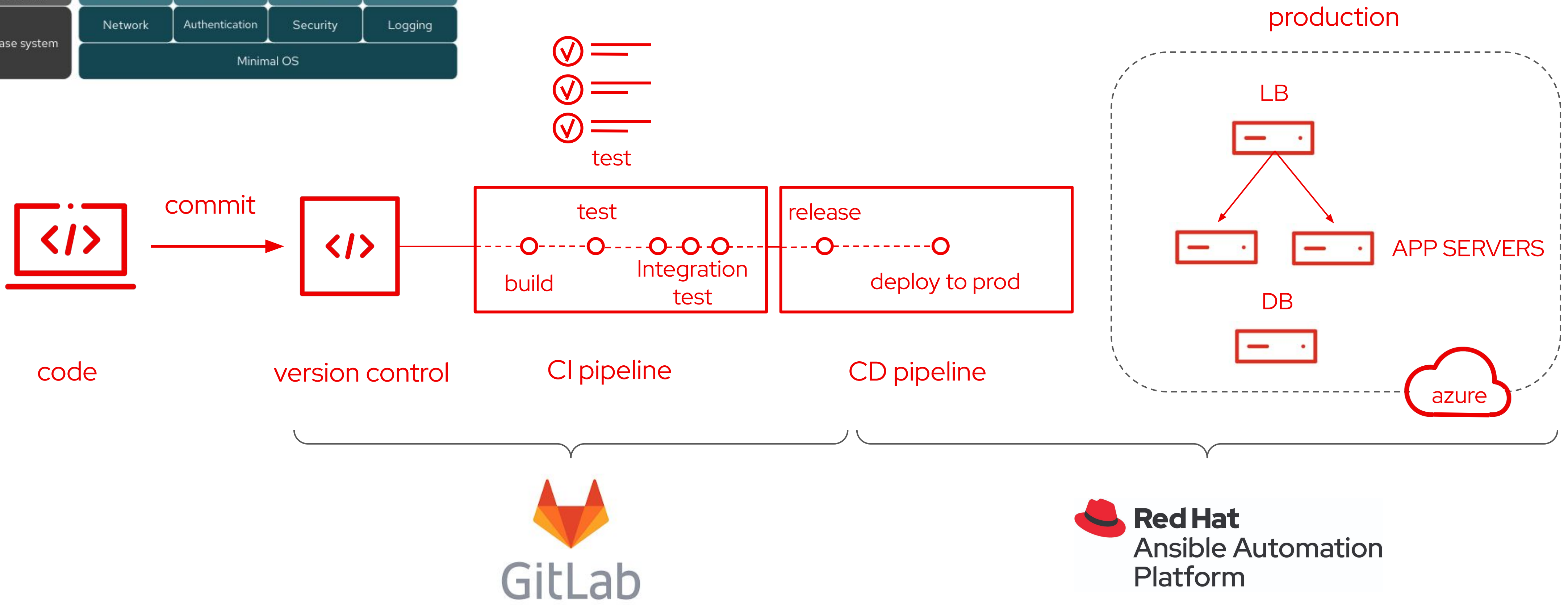
- Commit application change to git
- Triggers pipeline run with tests
- Deploy change to production
- Remove 1st appserver from load balancer
- Update 1st appserver and enable in load balancer
- Remove 2nd appserver from load balancer
- Update 2nd appserver and enable in load balancer.

Seamless upgrade of application

Red Hat



Application	App 1	App 2	...	App n
Application platform	Database		JBoss EAP	
	Database-ready	Transactions-ready	JBoss EAP-ready	Financials-ready
Optional services	Backup	Cluster	Storage	VMware
Base system	Network	Authentication	Security	Logging
	Minimal OS			



DEMO

Seamless upgrade of application

- ✓ No manual steps
- ✓ No human errors
- ✓ Predictable outcomes
- ✓ Higher efficiency
- ✓ Faster time to market
- ✓ Less stress

Continuous Integration

Simplified collaborative application development

- Self service application platform
- Build and test your application automatically
- Standardised native tools
- Everything as code
  - Application
  - Deployment
  - Build and test
- Collaborative workflow

OpenShift Pipelines

Open source, standardised, cloud-native

based on TEKTON

Why OpenShift Pipelines?

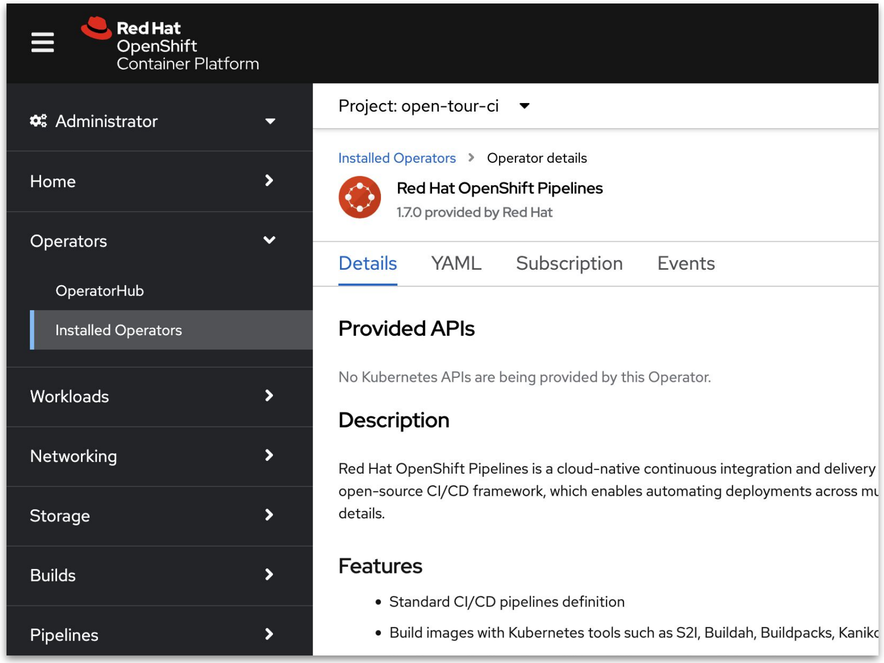
Built for Cloud-native

Scale on-demand

Secure pipeline execution

Flexible and powerful

Pipelines as a service



Red Hat OpenShift Pipelines

1.7.0 provided by Red Hat

Provided APIs

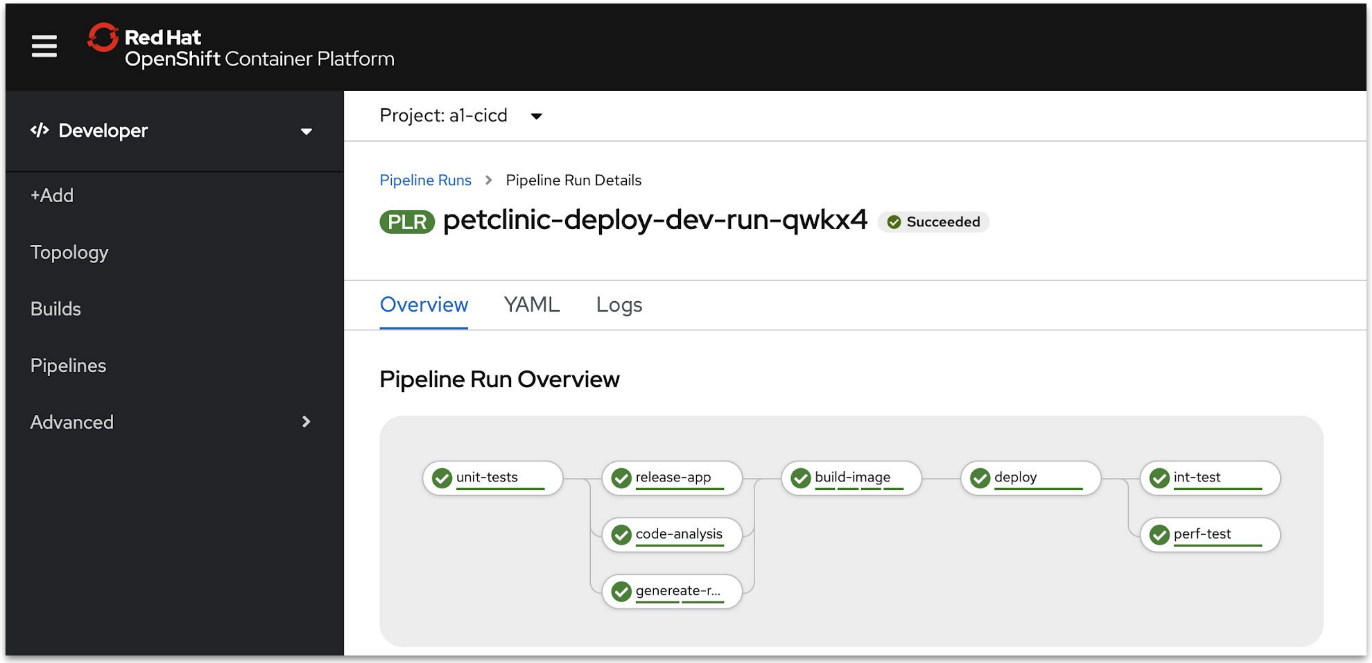
No Kubernetes APIs are being provided by this Operator.

Description

Red Hat OpenShift Pipelines is a cloud-native continuous integration and delivery
open-source CI/CD framework, which enables automating deployments across mu
details.

Features

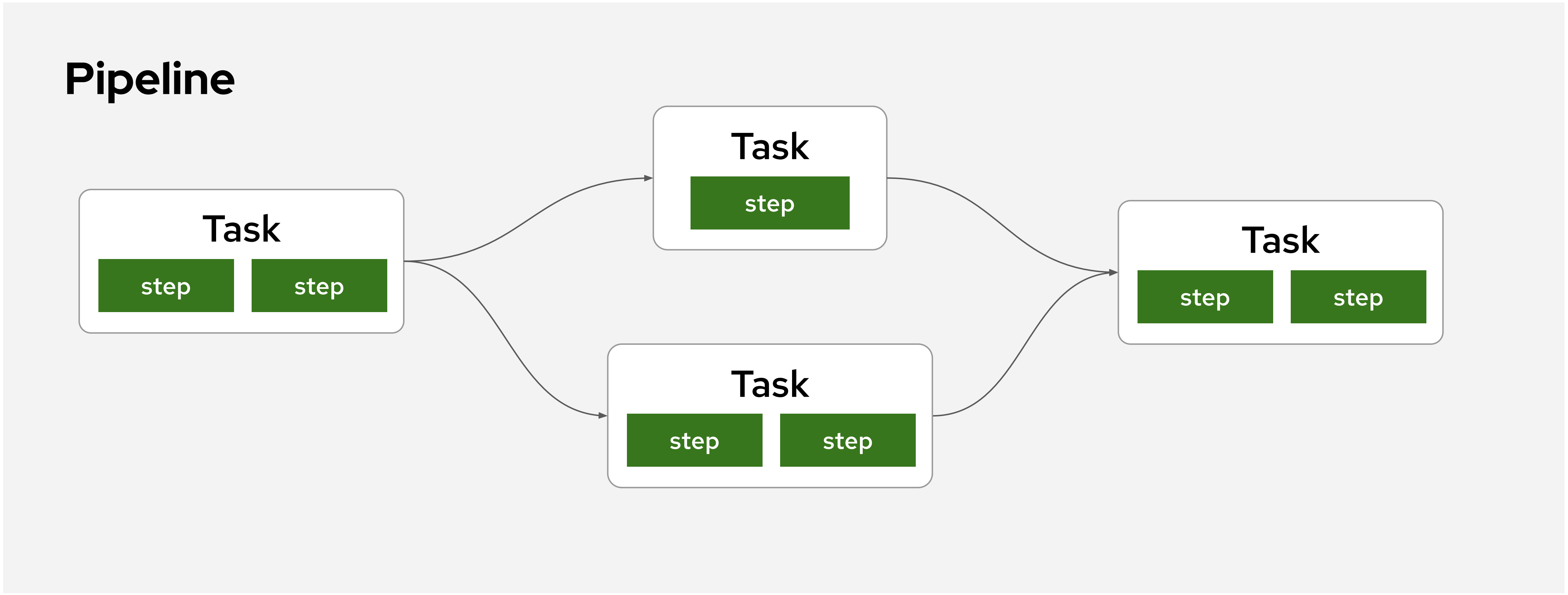
- Standard CI/CD pipelines definition
- Build images with Kubernetes tools such as S2I, Buildah, Buildpacks, Kaniko



PLR petclinic-deploy-dev-run-qwkx4 Succeeded

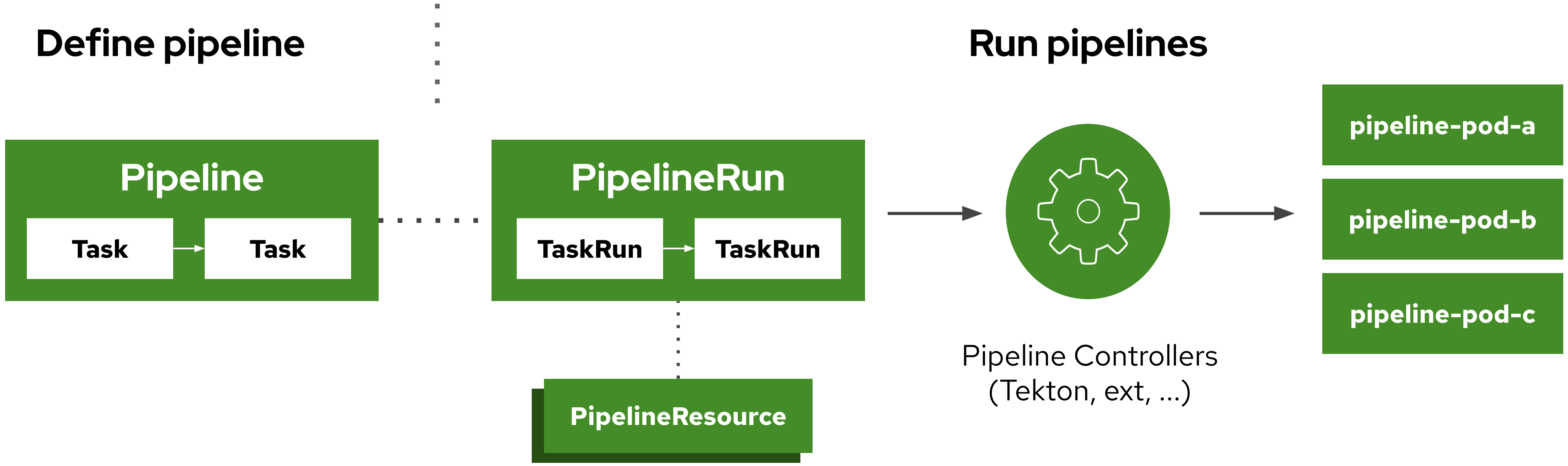
Pipeline Run Overview

OpenShift Pipelines - Tekton concepts

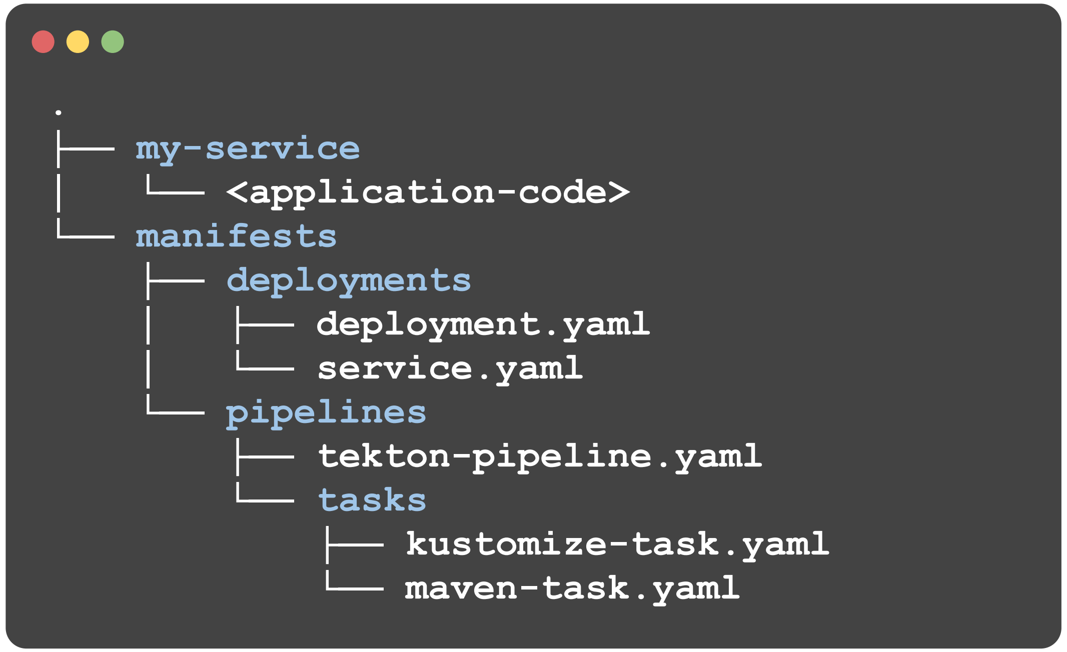


Red Hat

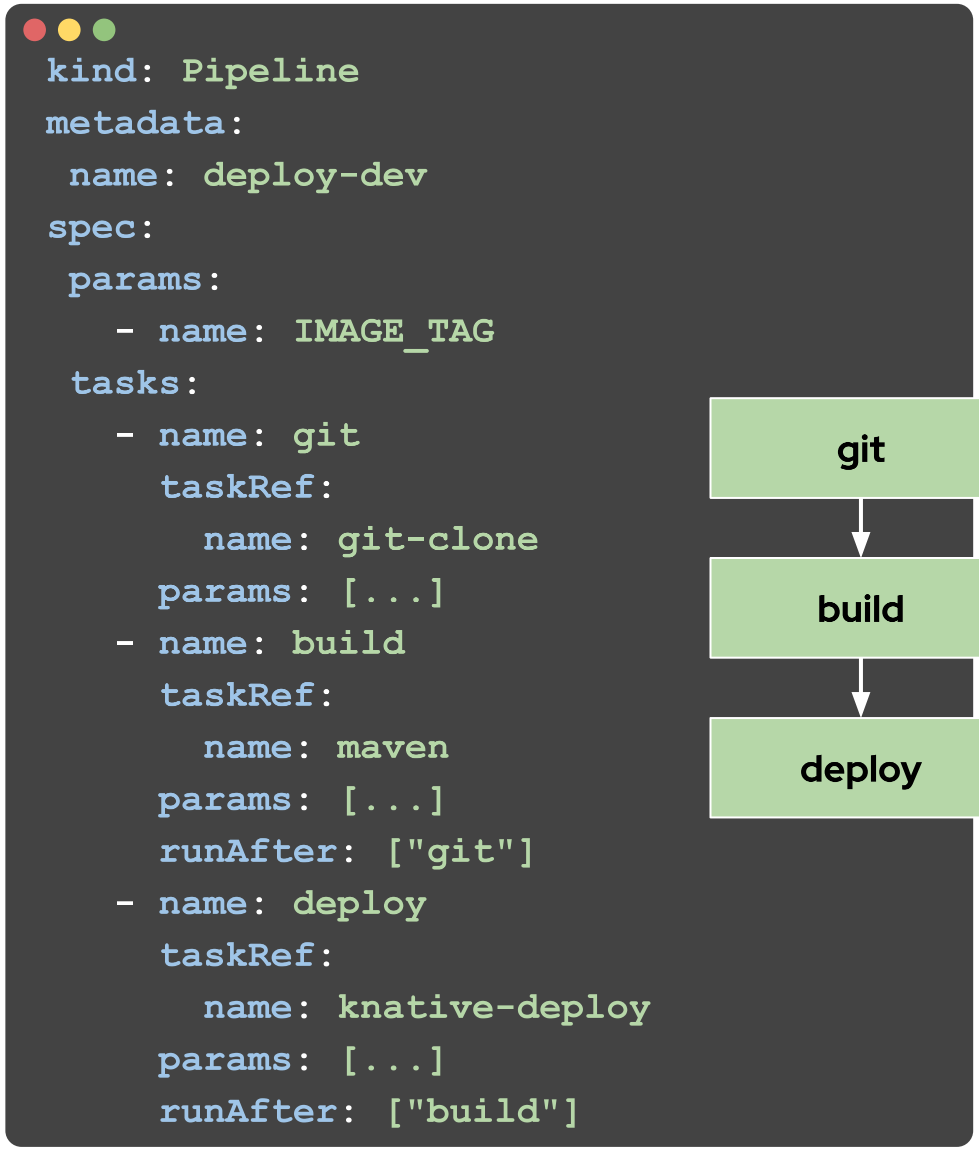
OpenShift Pipelines - Architecture



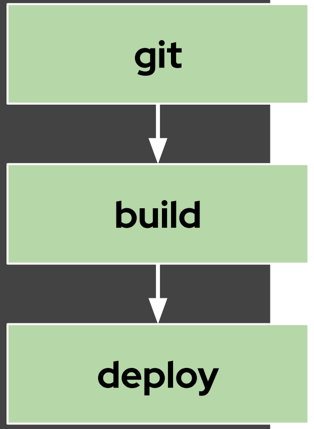
Everything as code



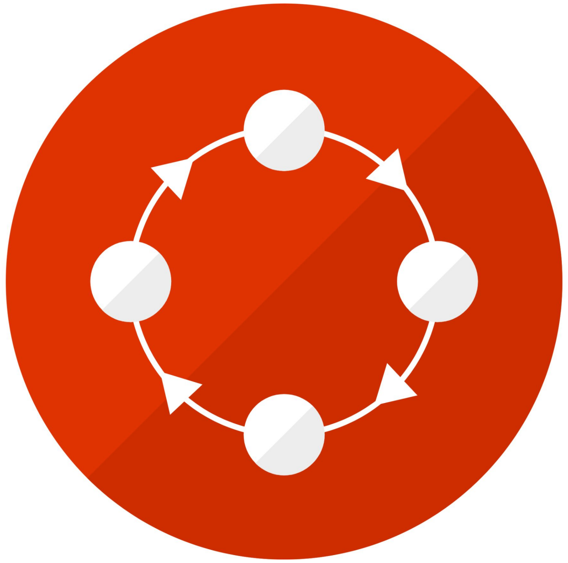
```
.
├── my-service
│   └── <application-code>
└── manifests
    ├── deployments
    │   ├── deployment.yaml
    │   └── service.yaml
    └── pipelines
        ├── tekton-pipeline.yaml
        └── tasks
            ├── kustomize-task.yaml
            └── maven-task.yaml
```



```
kind: Pipeline
metadata:
 name: deploy-dev
spec:
 params:
   - name: IMAGE_TAG
 tasks:
   - name: git
     taskRef:
       name: git-clone
     params: [...]
   - name: build
     taskRef:
       name: maven
     params: [...]
     runAfter: ["git"]
   - name: deploy
     taskRef:
       name: knative-deploy
     params: [...]
     runAfter: ["build"]
```



Pipelines as code



- GitOps enabled - git-centric workflow
- Integrated with Git provider
  - Events, actions
- Pipelines run in cluster
  - No pre-configured infrastructure

Automate the automation

Standardise your CI - cloud-native style

1. **Pipelines as a (cluster) service**

2. **CI resources - cloud-native**

3. **Git-centric workflow**

DEMO

- ✓ No manual steps
- ✓ No human errors
- ✓ Predictable outcomes
- ✓ Higher efficiency
- ✓ Faster time to market
- ✓ Less stress

Simplified collaborative application development

Red Hat

Continuous Delivery

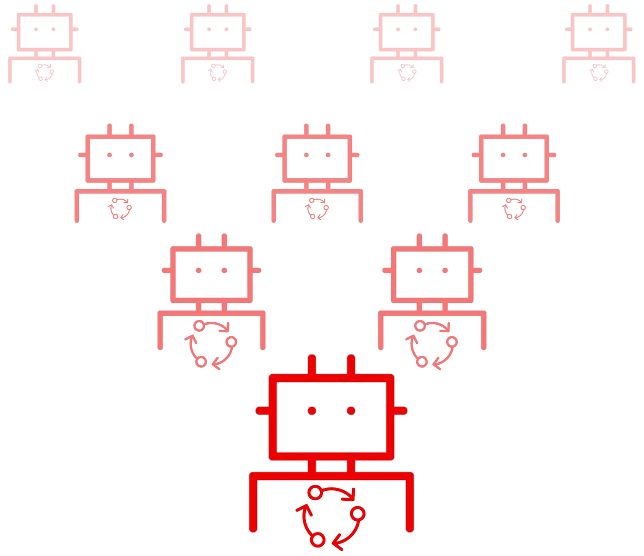
Hybrid cloud pattern: Multicloud GitOps

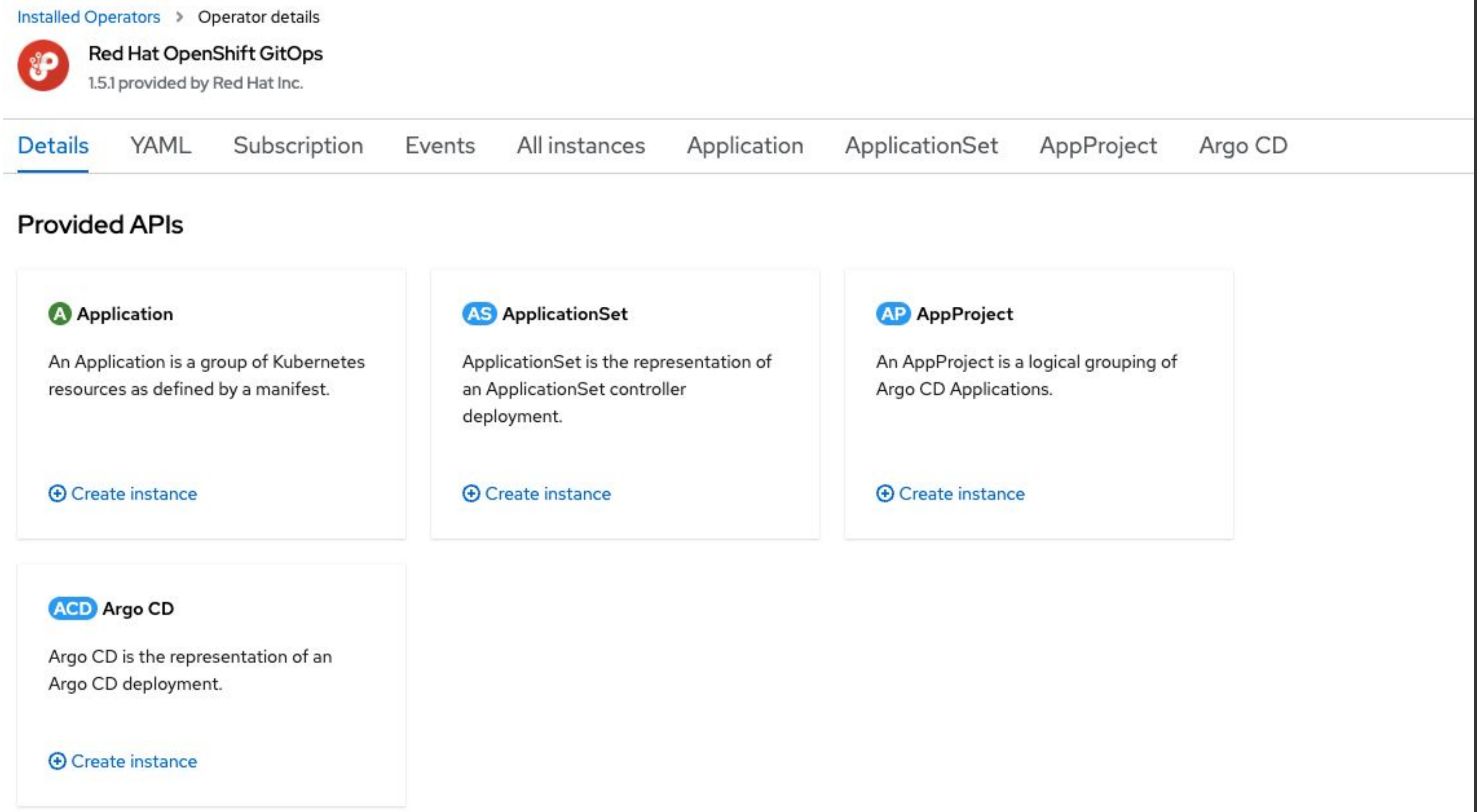
Automated business continuity

- Keep delivering no matter of location
- Automate introduction of new features
- Manage risks by replication and scaling out environments
- Everything automated

We want everything as code.

Applications, configurations and secrets delivered to autonomous environments.
Visible change history.
Comes with self healing.




Installed Operators > Operator details

Red Hat OpenShift GitOps

1.5.1 provided by Red Hat Inc.

Details | YAML | Subscription | Events | All instances | Application | ApplicationSet | AppProject | Argo CD

Provided APIs

Application

An Application is a group of Kubernetes resources as defined by a manifest.

⊕ Create instance

ApplicationSet

ApplicationSet is the representation of an ApplicationSet controller deployment.

⊕ Create instance

AppProject

An AppProject is a logical grouping of Argo CD Applications.

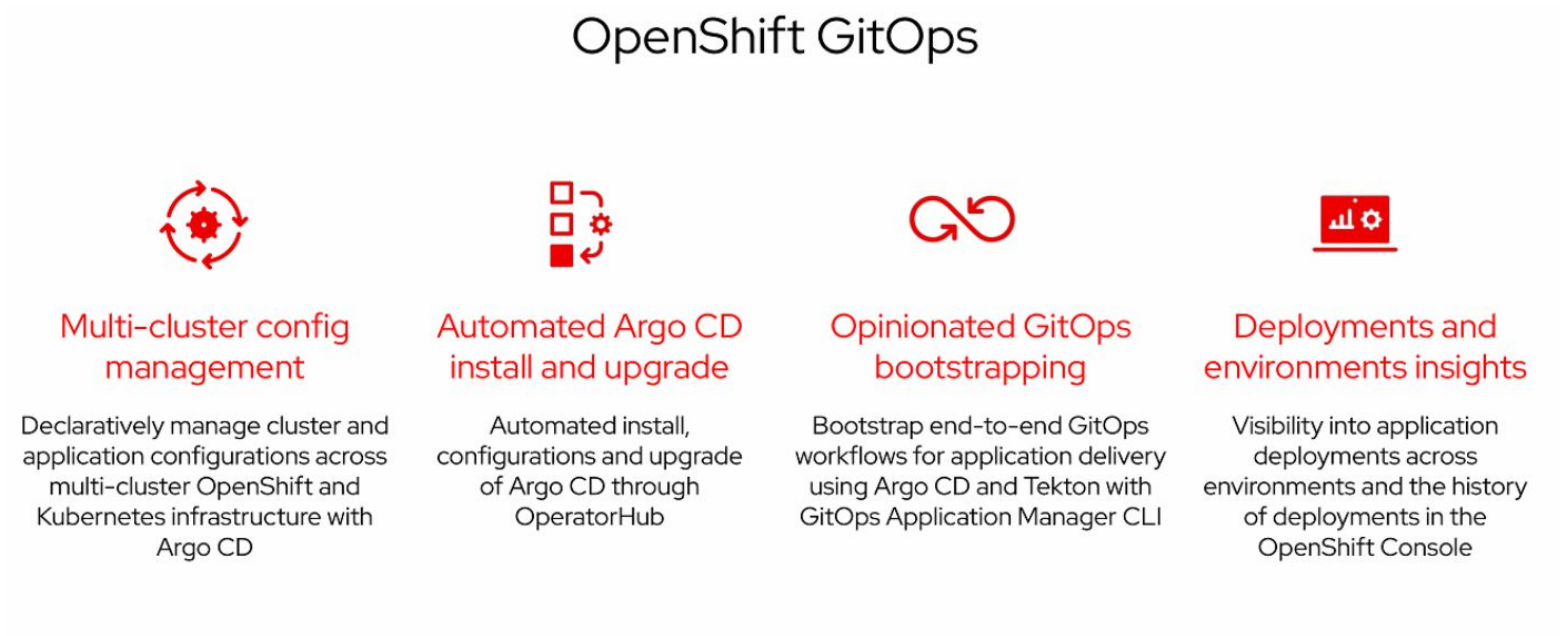
⊕ Create instance

Argo CD

Argo CD is the representation of an Argo CD deployment.

⊕ Create instance

Operator based



OpenShift GitOps

Multi-cluster config management

Declaratively manage cluster and application configurations across multi-cluster OpenShift and Kubernetes infrastructure with Argo CD

Automated Argo CD install and upgrade

Automated install, configurations and upgrade of Argo CD through OperatorHub

Opinionated GitOps bootstrapping

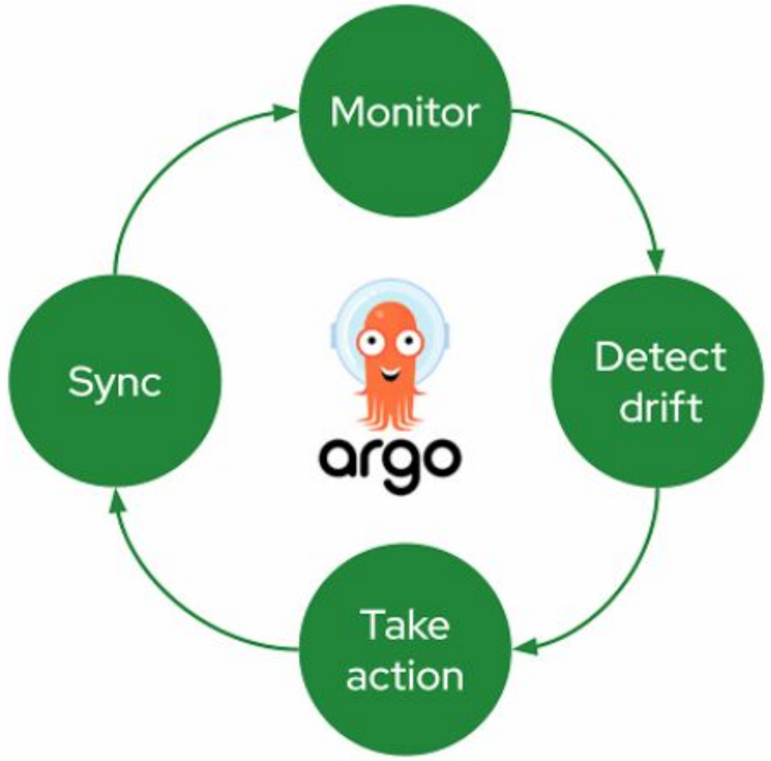
Bootstrap end-to-end GitOps workflows for application delivery using Argo CD and Tekton with GitOps Application Manager CLI

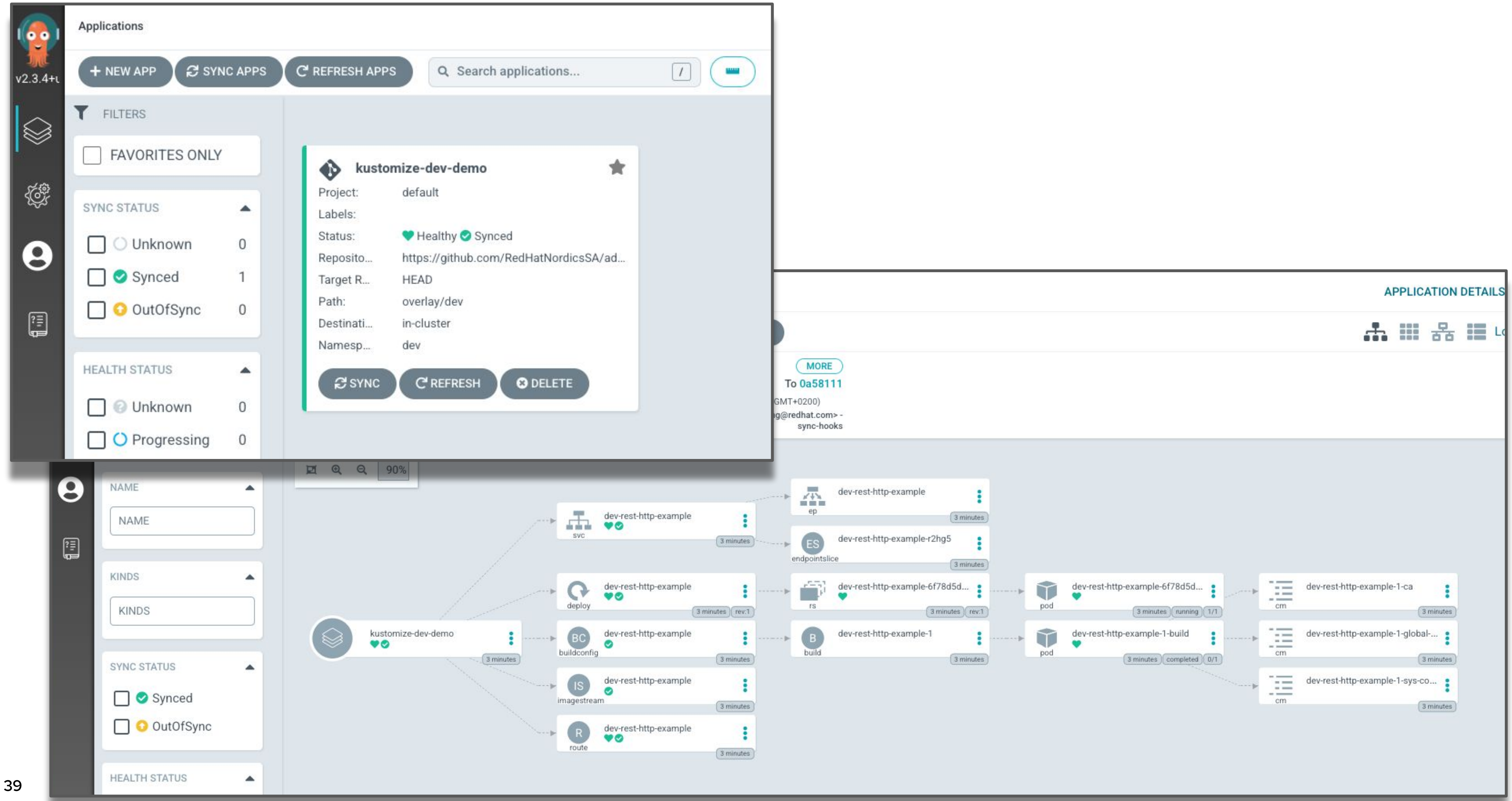
Deployments and environments insights

Visibility into application deployments across environments and the history of deployments in the OpenShift Console

Argo CD

- Cluster and application configuration versioned in Git
- Automatically syncs configuration from Git to clusters
- Drift detection, visualization and correction
- Granular control over sync order for complex rollouts
- Rollback and rollforward to any Git commit
- Manifest templating support (Helm, Kustomize, etc)
- Visual insight into sync status and history





☐	Unknown	0
☐	Synced	1
☐	OutOfSync	0

☐	Unknown	0
☐	Progressing	0

kustomize-dev-demo

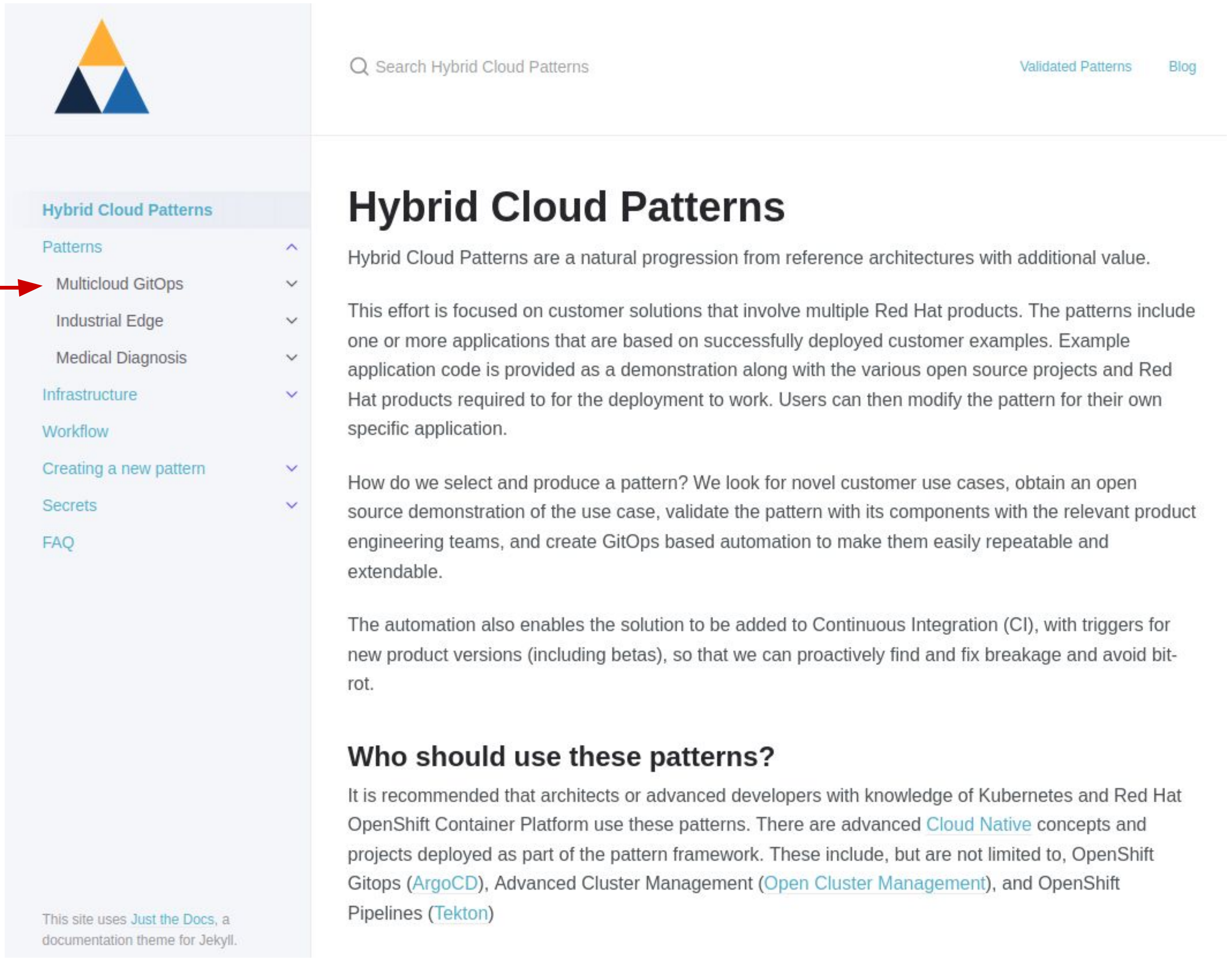
Project:	default
Labels:	
Status:	Healthy Synced
Reposito...	https://github.com/RedHatNordicsSA/ad...
Target R...	HEAD
Path:	overlay/dev
Destinati...	in-cluster
Namesp...	dev

To 0a58111
GMT+0200)
g@redhat.com> -
sync-hooks

SYNC STATUS
☐ Synced
☐ OutOfSync

kustomize-dev-demo — 3 minutes
dev-rest-http-example (svc) — 3 minutes
dev-rest-http-example (deploy) — 3 minutes, rev:1
dev-rest-http-example (buildconfig) — 3 minutes
dev-rest-http-example (imagestream) — 3 minutes
dev-rest-http-example (route) — 3 minutes
dev-rest-http-example (ep) — 3 minutes
dev-rest-http-example-r2hg5 (endpointslice) — 3 minutes
dev-rest-http-example-6f78d5d... (rs) — 3 minutes, rev:1
dev-rest-http-example-1 (build) — 3 minutes
dev-rest-http-example-6f78d5d... (pod) — 3 minutes, running, 1/1
dev-rest-http-example-1-build (pod) — 3 minutes, completed, 0/1
dev-rest-http-example-1-ca (cm) — 3 minutes
dev-rest-http-example-1-global-... (cm) — 3 minutes
dev-rest-http-example-1-sys-co... (cm) — 3 minutes

TAKE IDEAS FROM HYBRID CLOUD PATTERNS



This site uses Just the Docs, a documentation theme for Jekyll.

Hybrid Cloud Patterns

Hybrid Cloud Patterns are a natural progression from reference architectures with additional value.

This effort is focused on customer solutions that involve multiple Red Hat products. The patterns include one or more applications that are based on successfully deployed customer examples. Example application code is provided as a demonstration along with the various open source projects and Red Hat products required to for the deployment to work. Users can then modify the pattern for their own specific application.

How do we select and produce a pattern? We look for novel customer use cases, obtain an open source demonstration of the use case, validate the pattern with its components with the relevant product engineering teams, and create GitOps based automation to make them easily repeatable and extendable.

The automation also enables the solution to be added to Continuous Integration (CI), with triggers for new product versions (including betas), so that we can proactively find and fix breakage and avoid bit-rot.

Who should use these patterns?

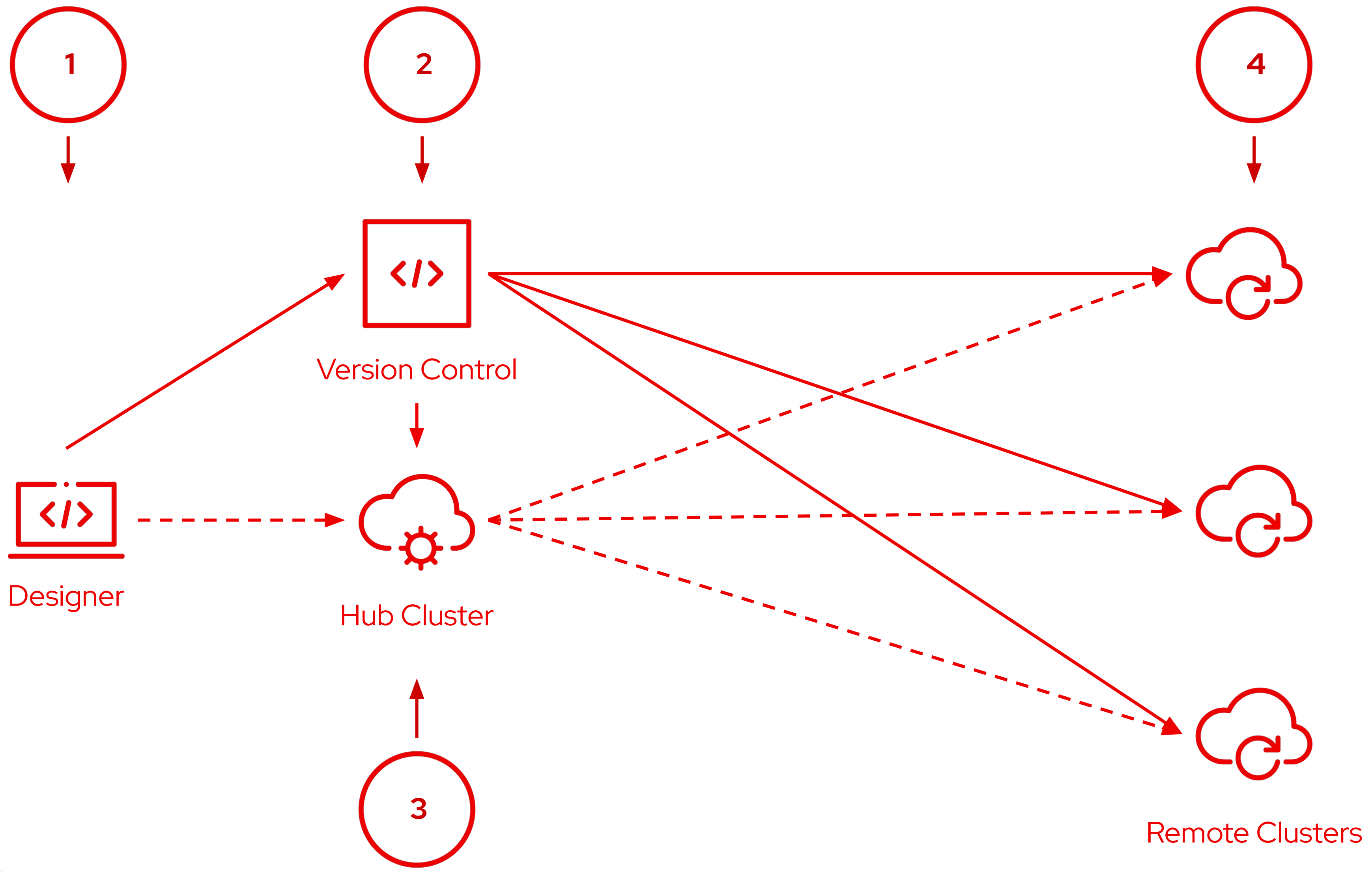
It is recommended that architects or advanced developers with knowledge of Kubernetes and Red Hat OpenShift Container Platform use these patterns. There are advanced Cloud Native concepts and projects deployed as part of the pattern framework. These include, but are not limited to, OpenShift Gitops (ArgoCD), Advanced Cluster Management (Open Cluster Management), and OpenShift Pipelines (Tekton)

Let’s look at the multicloud gitops pattern today

https://redhat-gitops-patterns.io/
https://hybrid-cloud-patterns.io/multicloud-gitops/

1. Create or Enhance
2. Version Control
3. Automate
4. Continuously Deliver

Red Hat

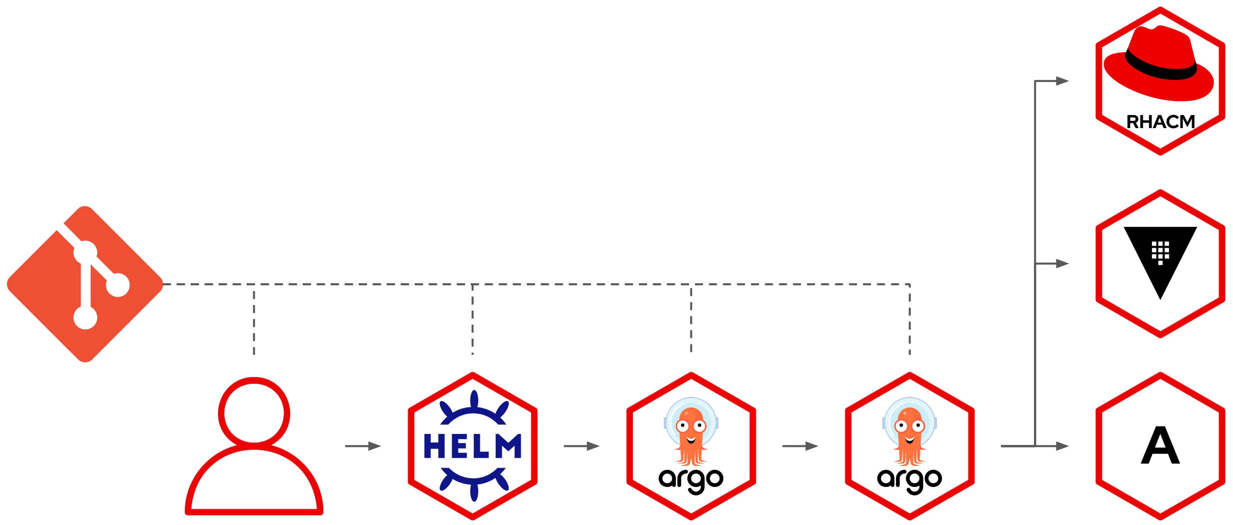


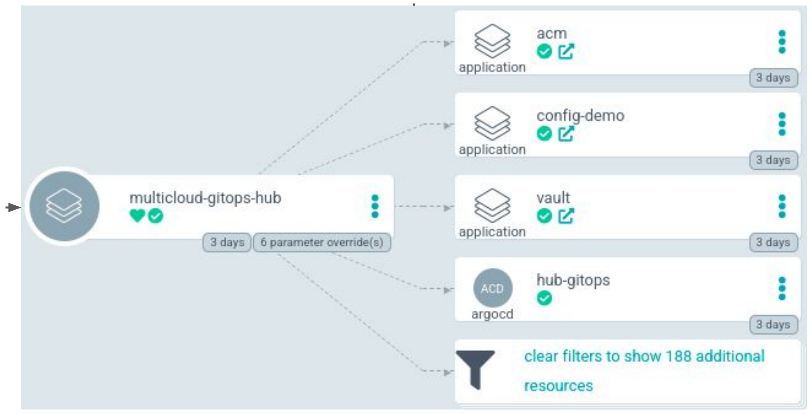
One repository to control delivery versions

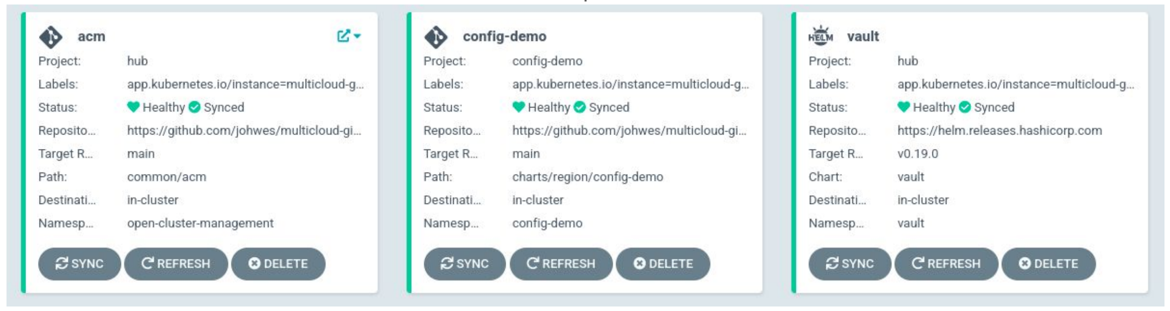
Several environments in hybrid clouds to automatically adapt to configuration or application changes.

1. Create
2. Commit
3. Automate - - - -
4. Keep Delivering ———

MANAGEMENT HUB SETUP



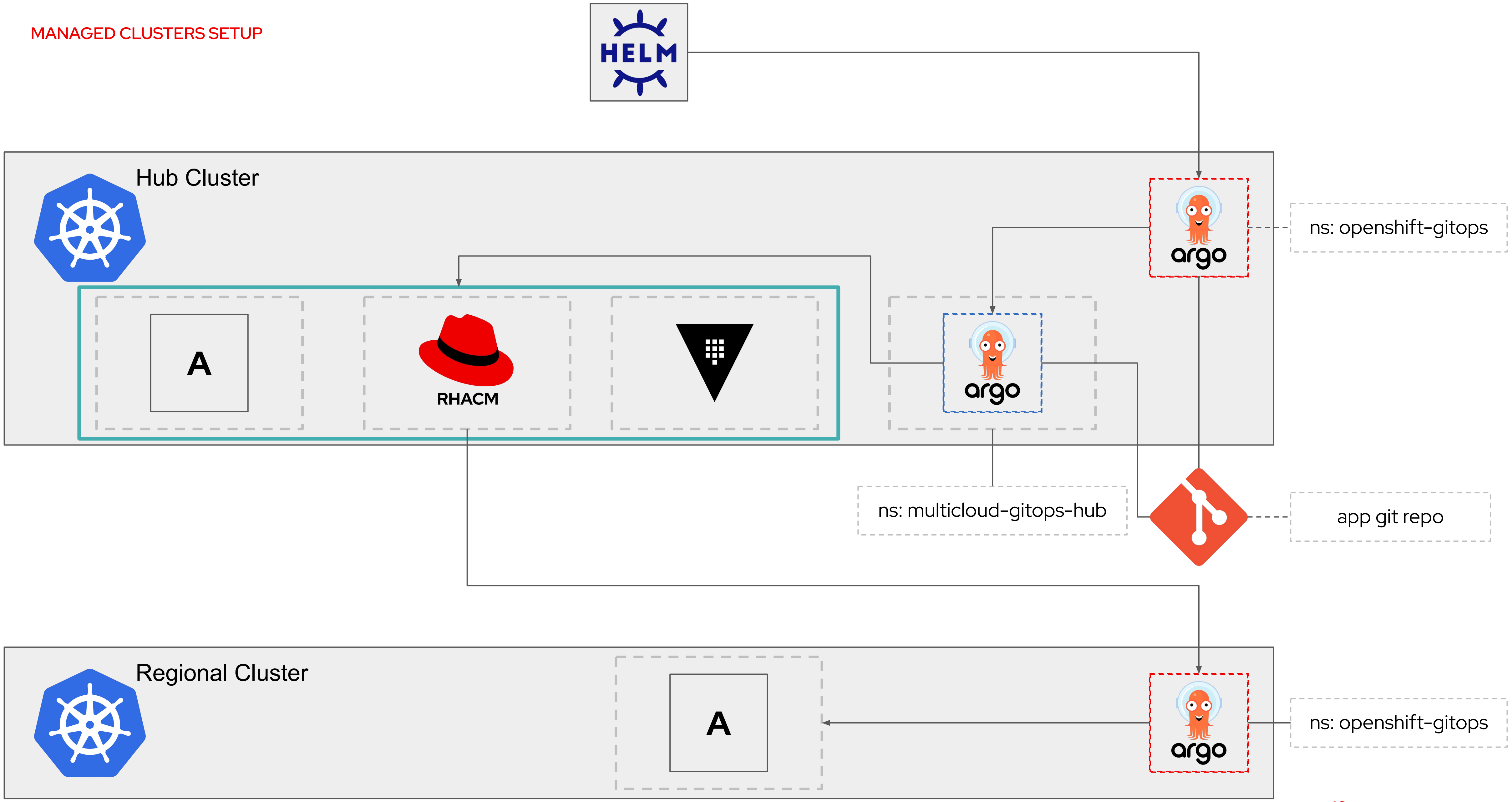




	acm	config-demo	vault
Project:	hub	config-demo	hub
Labels:	app.kubernetes.io/instance=multicloud-g...	app.kubernetes.io/instance=multicloud-g...	app.kubernetes.io/instance=multicloud-g...
Status:	Healthy Synced	Healthy Synced	Healthy Synced
Reposito...	https://github.com/johwes/multicloud-gi...	https://github.com/johwes/multicloud-gi...	https://helm.releases.hashicorp.com
Target R...	main	main	v0.19.0
Path: / Chart:	common/acm	charts/region/config-demo	vault
Destinati...	in-cluster	in-cluster	in-cluster
Namesp...	open-cluster-management	config-demo	vault

Red Hat

MANAGED CLUSTERS SETUP



DEMO

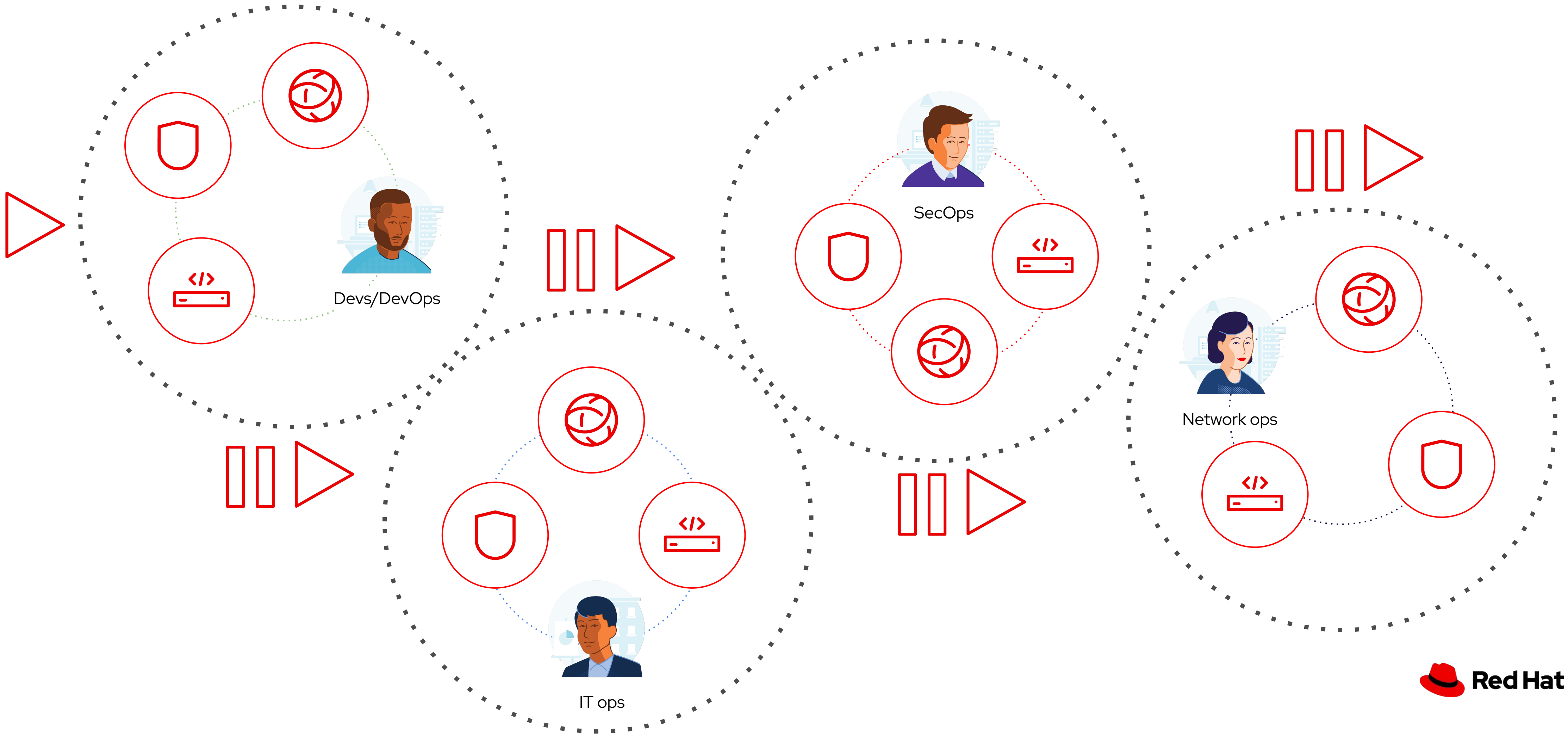
Automated business continuity

- ✓ No manual steps
- ✓ No human errors
- ✓ Predictable outcomes
- ✓ Higher efficiency
- ✓ Faster time to market
- ✓ Less stress

Provide business value through collaboration

But many organizations have a common problem...

Too many unintegrated, domain-specific tools, limited collaboration and scale



Red Hat

In this presentation you learned about

- Standardisation
- Automation
- Collaboration

To gain robust repeatability as self service, by automating the automation

Red Hat

Red Hat Services get you going!

Red Hat