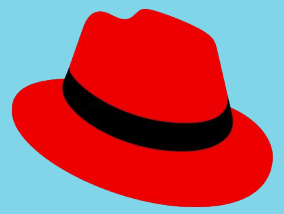
Connect

Red Hat



Red Hat

OpenShift GitOps & Advanced Cluster Management

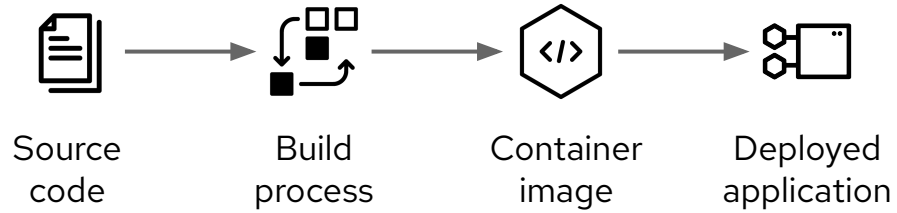
Continuous Delivery *at scale* on OpenShift

Mark Roberts
Principal Solution Architect
Red Hat

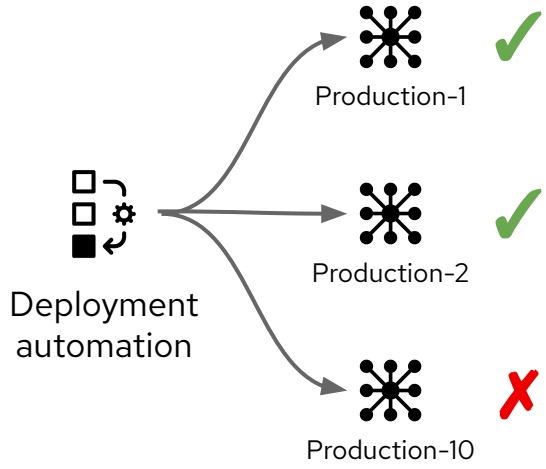
Red Hat

The challenge

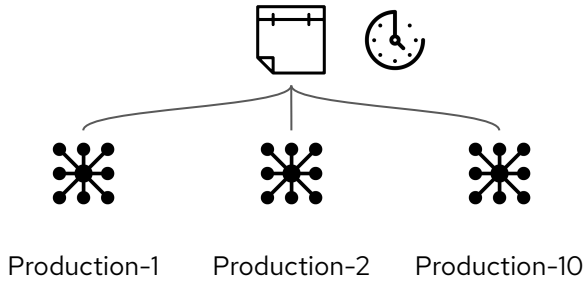
Secure Software Pipelines



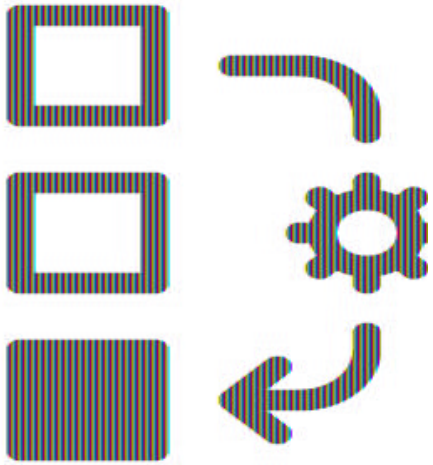
Deployment to multiple clusters



Ephemeral clusters



Red Hat



Objective

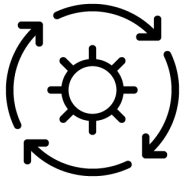
To create an automated build and deploy process

Red Hat

GitOps Principles

The system is described declaratively

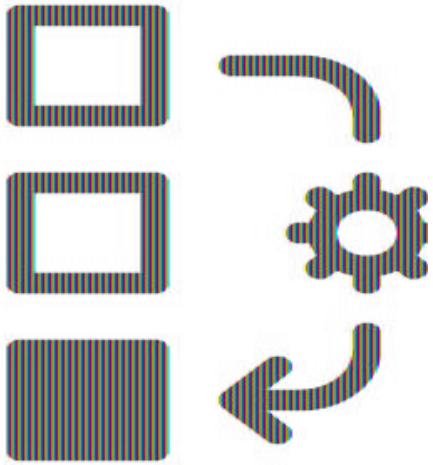
The desired state is versioned in Git



Approved changes can be applied automatically

A controller exists to detect and act on drift

Objective



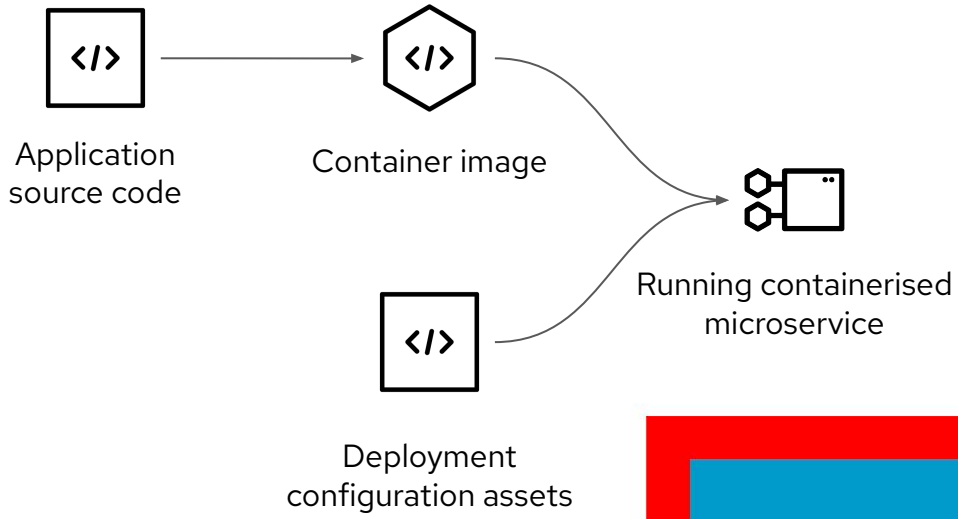
To create an automated build and deploy process in which the assets are stored in and managed from a Git repository

At scale in an ephemeral environment

Red Hat

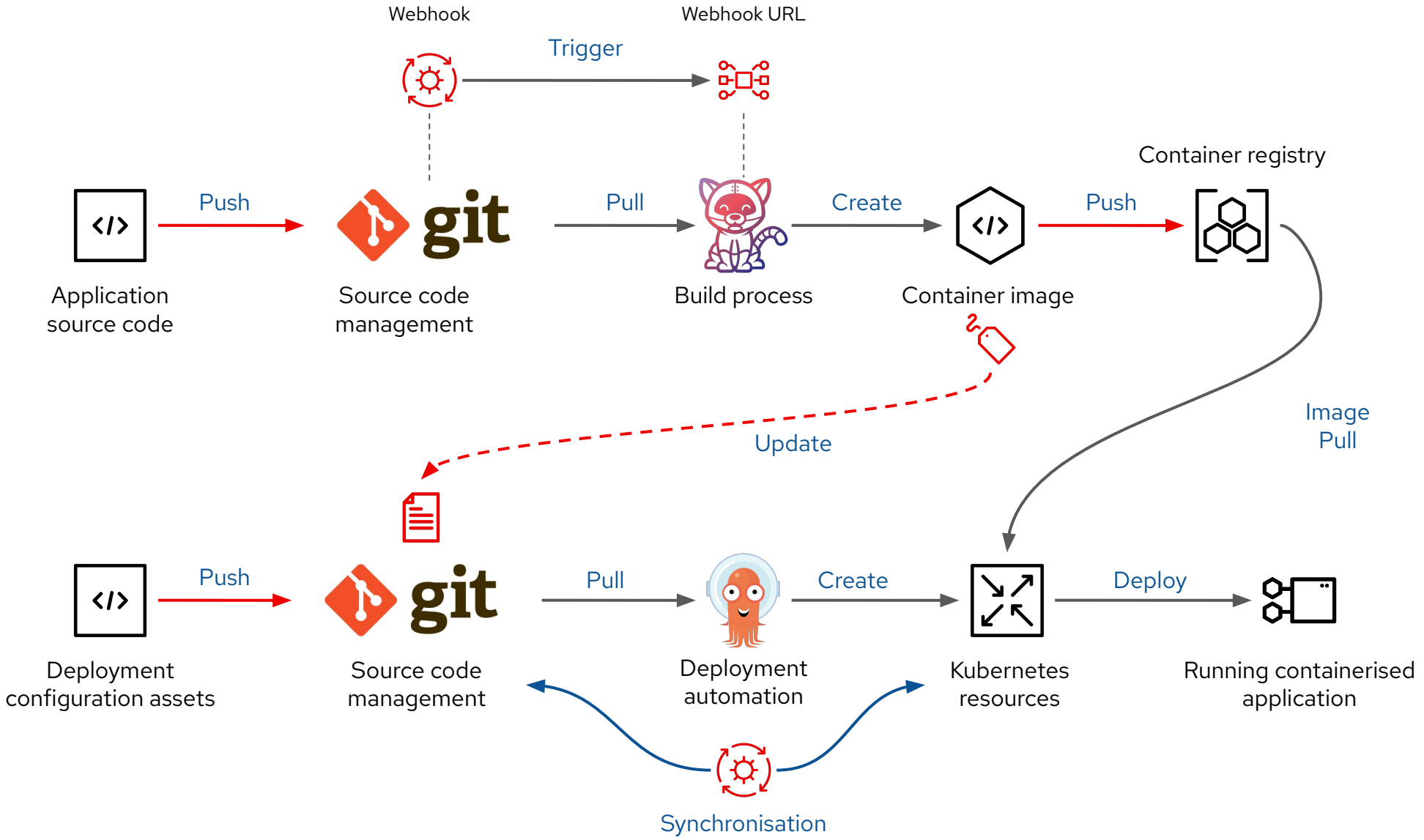
Source content and objectives

- From source content to running microservice
  - Source code of application
  - Container images (both a target and a source asset)
  - Kubernetes resources - services, configuration maps, secrets etc.
- Two major phases
  - Build - Continuous integration
  - Deploy - Continuous delivery



Red Hat

CI / CD Integrated delivery process



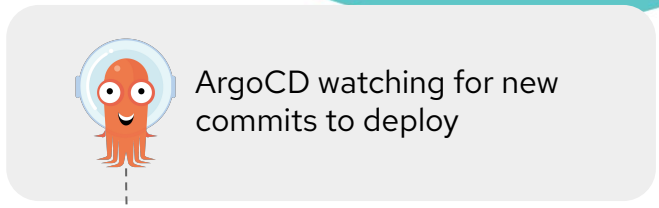
OpenShift Pipelines

OpenShift GitOps

Red Hat

OpenShift Pipelines (ArgoCD)
Application

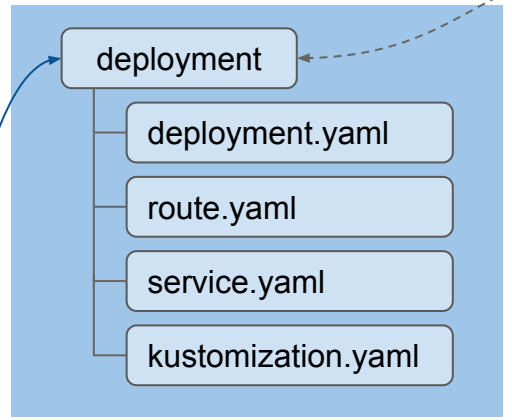
- A definition of a set of content to be deployed together which deliver value
- A simple source ⇒ target relationship
- A series of applications are grouped together in the context of a project

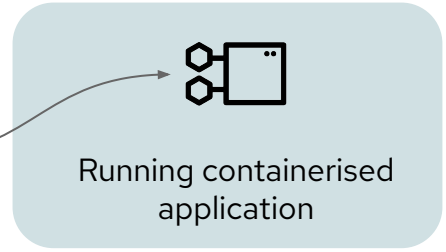



```yaml
apiVersion: argoproj.io/v1alpha1
kind: Application
metadata:
  name: my-app
  namespace: openshift-gitops
spec:
  project: simple-apps
```

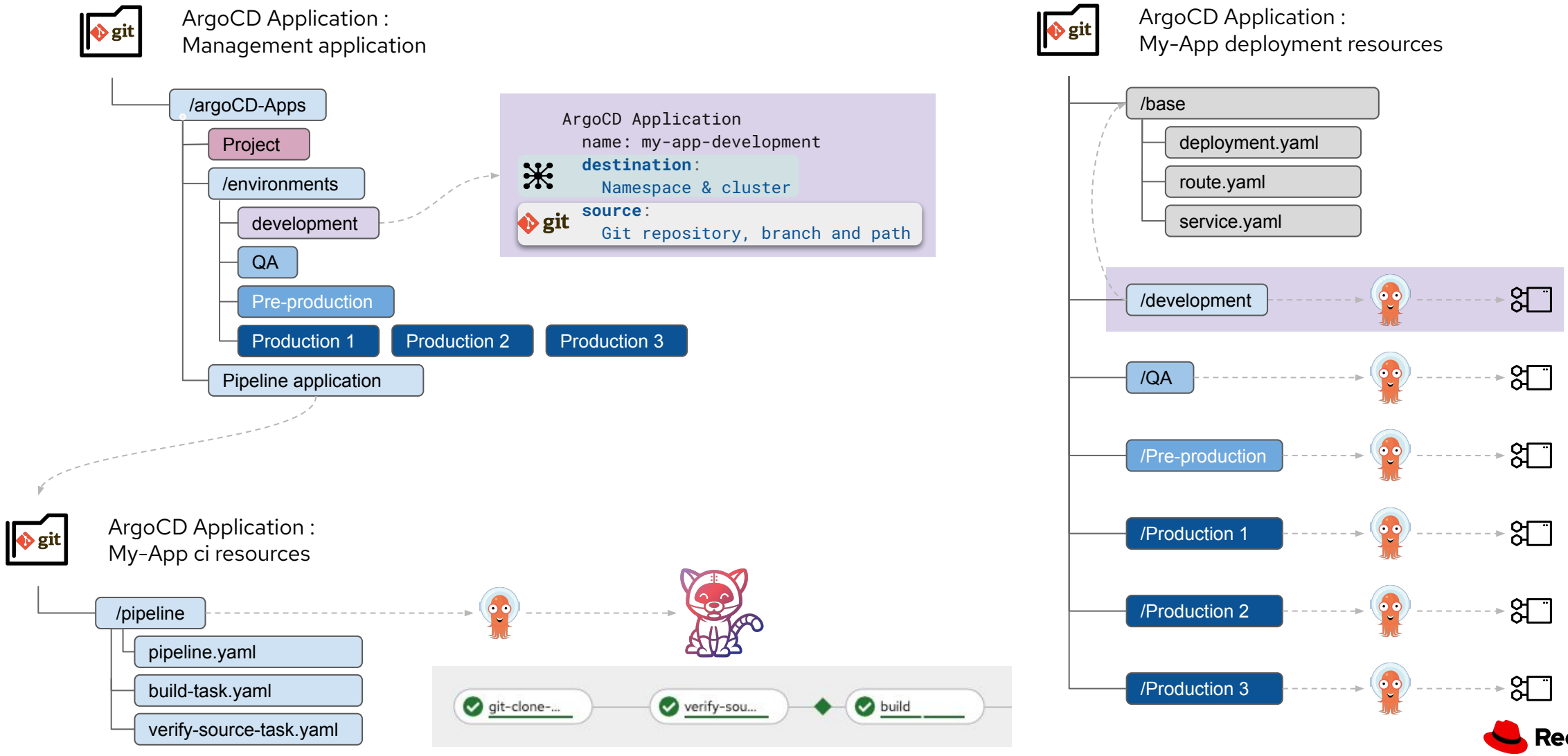
```yaml
destination:
  namespace: simple-pipeline
  server: https://kubernetes.default.svc
```

```yaml
source:
  path: pipelines/simplePipeline/deployment
  targetRevision: main
  repoURL: https://github.com/marrober/simple-apps.git
```





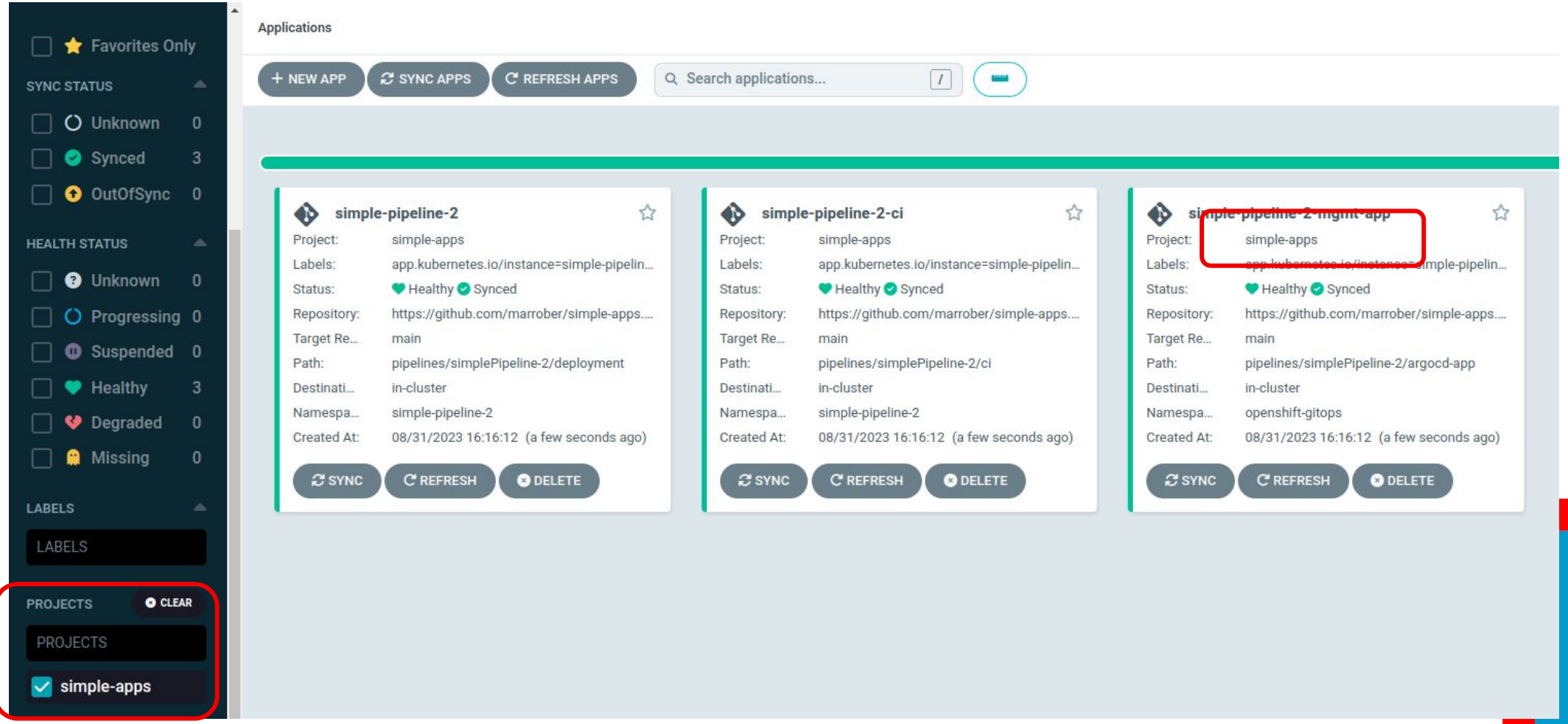
ArgoCD Project and applications



Red Hat

ArgoCD Project and applications

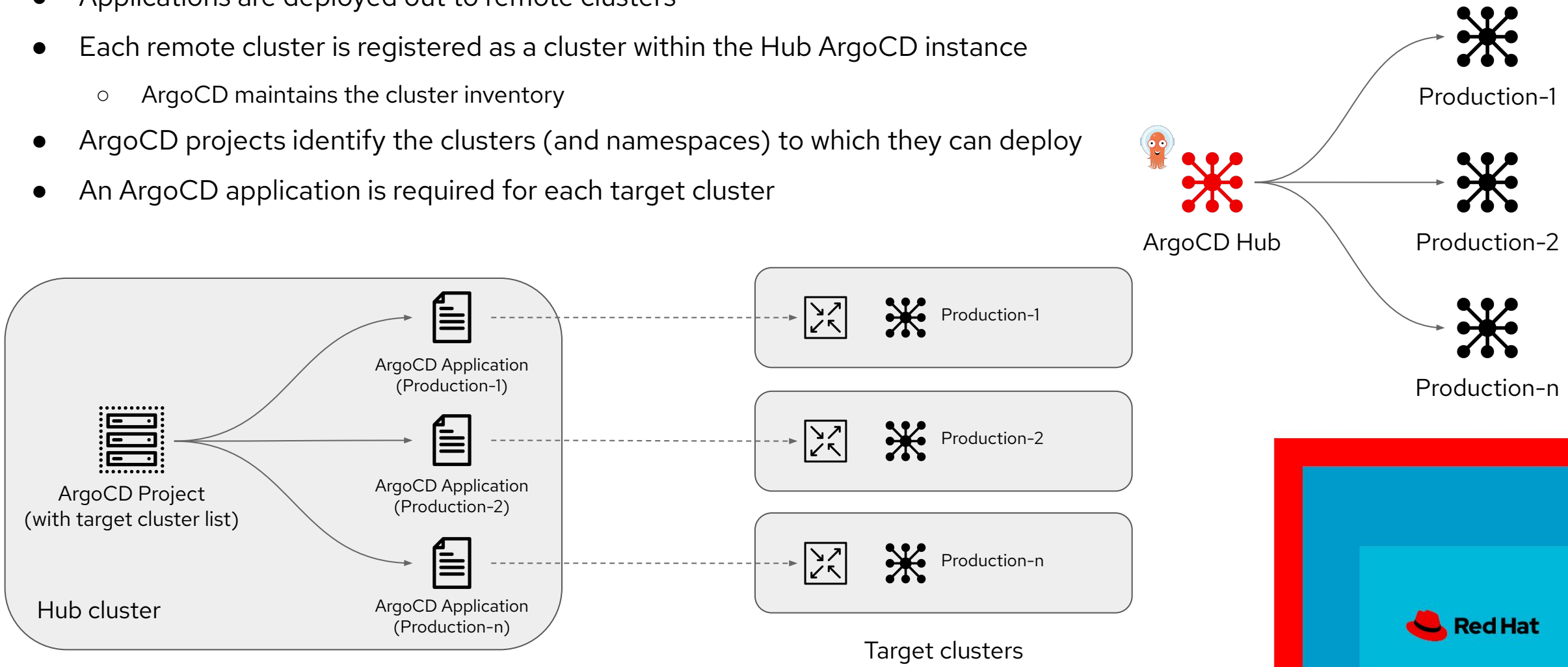
- Project based filtering in the ArgoCD web user interface helps focus on the applications that you need



Red Hat

ArgoCD : Push model

- ArgoCD hub targets ArgoCD instance in the receiving clusters
- Applications are deployed out to remote clusters
- Each remote cluster is registered as a cluster within the Hub ArgoCD instance
  - ArgoCD maintains the cluster inventory
- ArgoCD projects identify the clusters (and namespaces) to which they can deploy
- An ArgoCD application is required for each target cluster



ArgoCD Push model

Positives and negatives

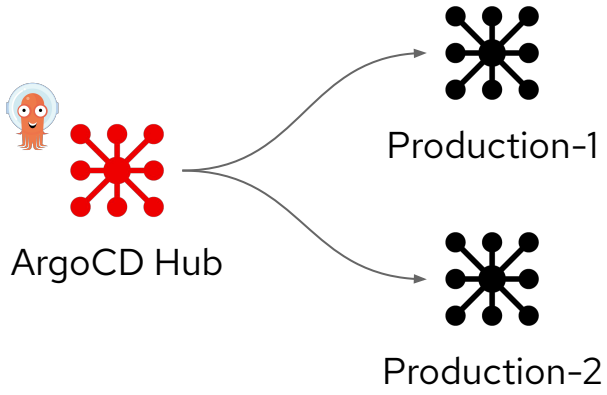
- Positives
  - Simple process to create the assets
  - Kustomize can be used to inject variance into a ‘base’ ArgoCD application to set the destination for each cluster
  - Centralised approach in which deployment automation runs on the hub cluster
- Negatives
  - Large numbers of target clusters can create a proliferation of ArgoCD Applications to manage
  - Could be problematic to maintain the cluster inventory over time
  - Requires direct connectivity from ArgoCD hub to target clusters
  - Hub cluster may be viewed as a single point of failure

ArgoCD deployment to multiple clusters –

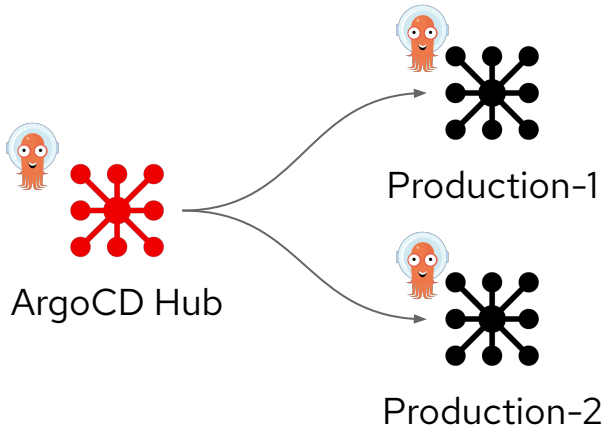
Pull model

- Enhanced by Red Hat Advanced Cluster Management for Kubernetes (ACM)
- The ArgoCD application definition is created on the hub and distributed to the remote cluster
- Status is reported back to the Hub cluster for overall reporting
- Benefits
  - Each remote cluster runs a separate ArgoCD instance and application lifecycle operating on independent schedules
  - Satisfies requirements of security teams : Remote (production) clusters connect to the hub

Push model



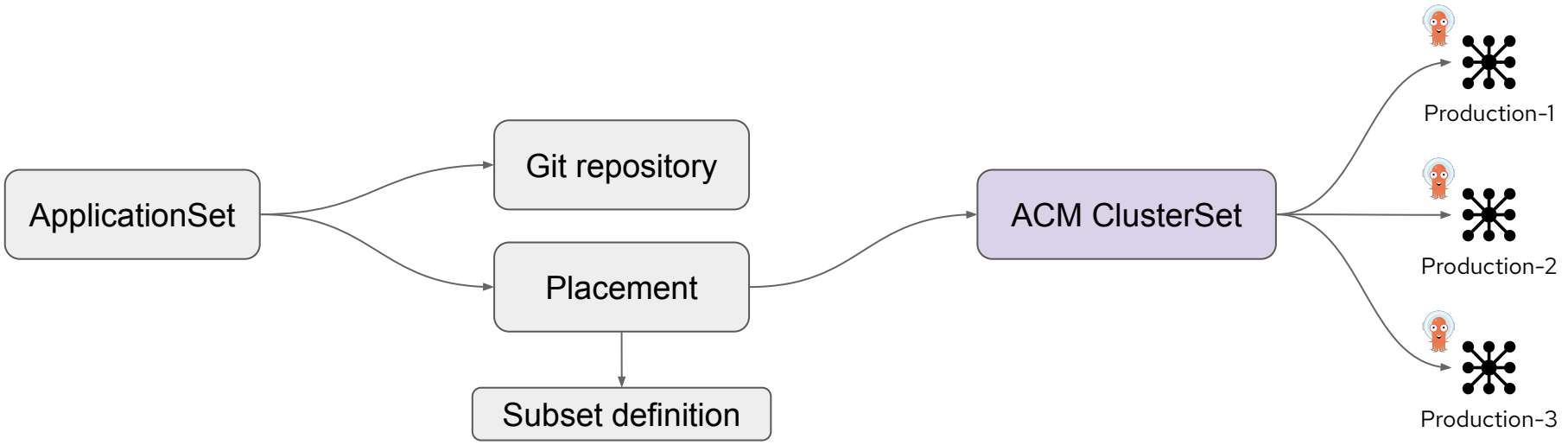
Pull model



Red Hat

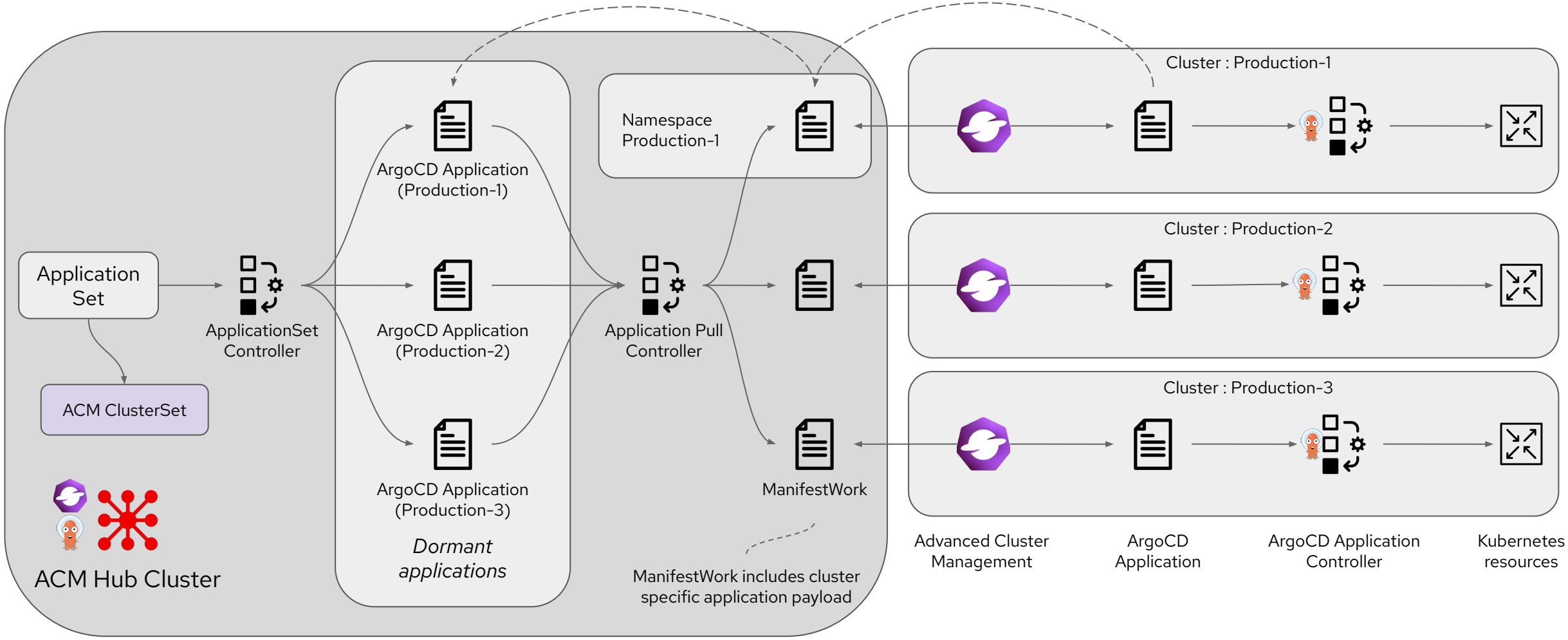
ArgoCD pull model resources

- ApplicationSet object is used to manage an application that is deployed to multiple clusters
  - References the source Git repository of deployable content
  - References a placement policy to select the target clusters
- Placement object identifies a ClusterSet and can select a subset of such clusters if required
- ACM ClusterSet
  - A set of ACM managed clusters that together deliver a function
  - Examples :
    - All production clusters
    - All production clusters in European region etc.



Red Hat

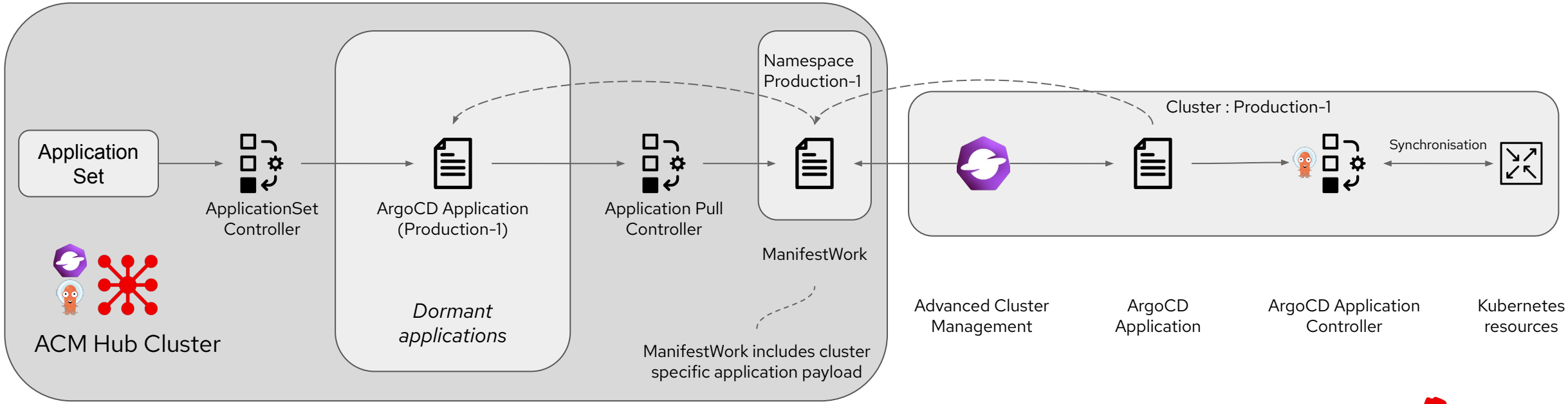
ApplicationSet Controller



Red Hat

ApplicationSet Controller - details

- *ApplicationSet Controller* creates (dormant) individual ArgoCD applications for each identified target cluster
- *Application Pull Controller* creates manifestWork objects for each target cluster
- manifestWork objects in cluster specific namespace are monitored (in a pull mode)
- Application object is extracted from manifestWork and created on the remote cluster
- Local ArgoCD instance on remote cluster processes the application object to create Kubernetes resources



Red Hat

OpenShift GitOps Summary

Red Hat
Advanced Cluster Management
for Kubernetes

OpenShift GitOps

- OpenShift GitOps delivers an automated deployment process
  - … in which the assets are stored in and managed from a Git repository
- Reconciliation of manual changes made to Kubernetes resource so that the Git content remains the single source of truth
- Visualisation of application status through the web user interface
- Management of complex multi-component applications through ArgoCD projects
- Control of environmental variance through the use of Kustomize

ArgoCD Push Model

- Independent of Red Hat Advanced Cluster Management for Kubernetes
- Centralised management of application lifecycle
- Supports multiple clusters via ArgoCD cluster registration
- Simpler process than the pull model - less moving parts
- Less content to manage on the remote cluster
- Ideal for more transient remote clusters

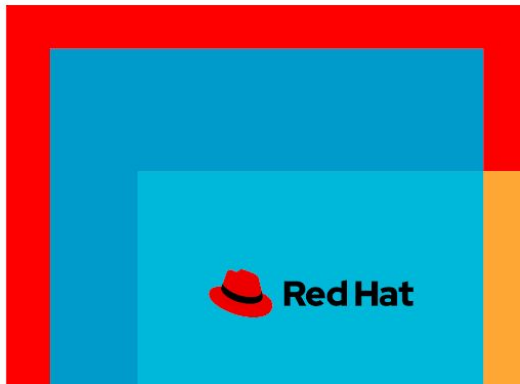
ArgoCD Pull Model

- Requires Red Hat Advanced Cluster Management for Kubernetes
- Distribution of ArgoCD applications to remote clusters
- Remote clusters operate the application lifecycle independent of the hub
- Status is reported back to the Hub cluster for overall reporting
- Supports secure connections - remote connect to Hub via ACM
- Less Kubernetes resources to manage

Red Hat

What’s next

- Have a chat today
- Get in touch - mroberts@redhat.com
- Schedule some time - web meeting or face to face
- Search the Red Hat blog site : https://www.redhat.com/en/blog



Red Hat Summit

Thank you


linkedin.com/company/red-hat


facebook.com/redhatinc


youtube.com/user/RedHatVideos


twitter.com/RedHat

Red Hat

References

- Blogs -
  - An introduction to GitOps - link
  - Introducing the Argo CD Application Pull Controller for Open Cluster Management - link
  - An Introduction to ApplicationSets in OpenShift GitOps - link
  - GitOps Deployment and Image Management - link
  - Your Guide to Continuous Delivery with OpenShift GitOps and Kustomize - link
  - A guide to using ArgoCD and GitOps with RBAC - link
  - A Guide to Secrets Management with GitOps and Kubernetes - link
  - A Guide to GitOps and Secret Management with ArgoCD Operator and SOPS - link
  - How to Use HashiCorp Vault and Argo CD for GitOps on OpenShift - link
  - Learn about Progressive Application Delivery with GitOps - link
  - A Guide to Going From Zero to OpenShift Cluster with GitOps - link
  - Understanding GitOps with Red Hat Advanced Cluster Management - link

Red Hat