# Beyond AIOps

## OpenShift Commons

Marcel Hild

Manager, AI CoE, Office of the CTO, Red Hat



Red Hat

AI Ops platforms are software systems that

combine big data and **AI** or machine learning functionality to

enhance and partially **replace** a broad range of

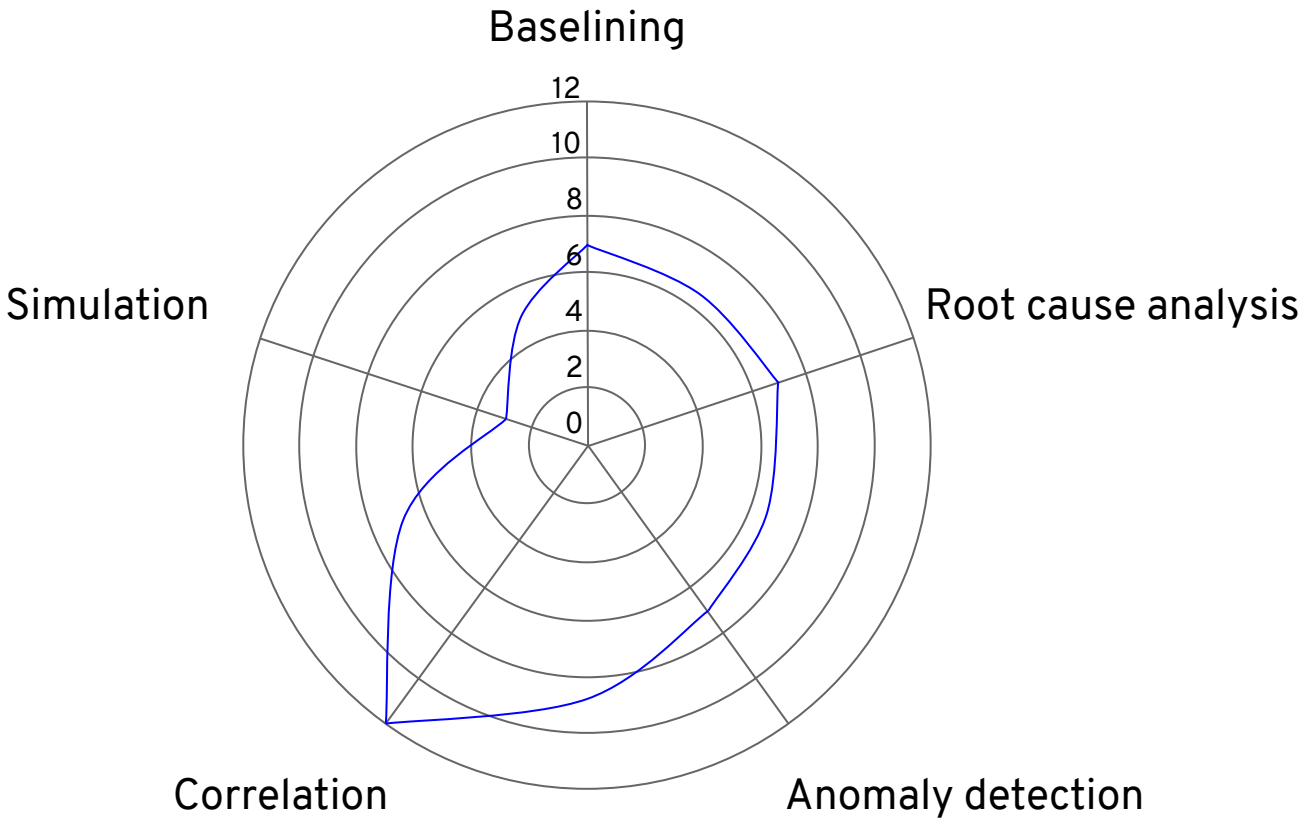
**IT operations** processes and tasks, including

availability and performance monitoring, event correlation and analysis,

IT service management, and automation.

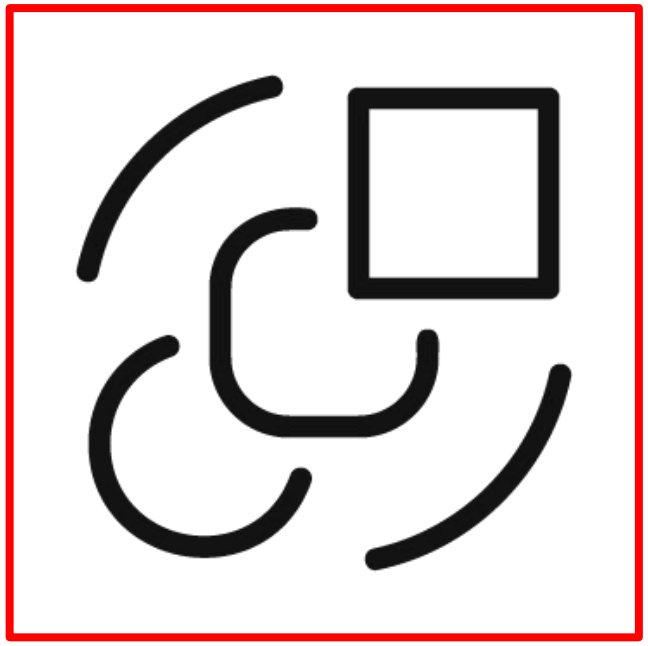
Source: Gartner Market Guide for AIOps Platforms Published: 03 August 2017 ID: G00322184

# Recurring AIOps Features

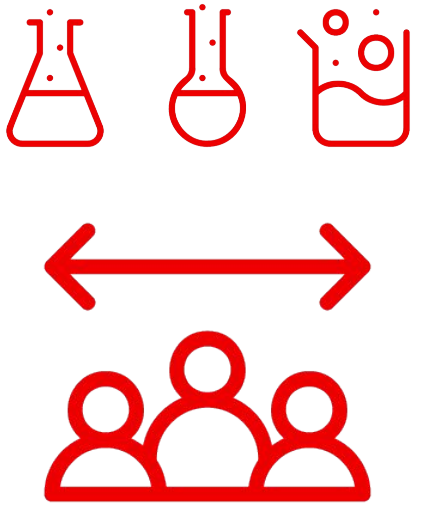


# AIOps in a Box

## It's not a Product



Boxed Product



Experiments → Capabilities

# AIOps is Culture

## Artificial Intelligence for IT Operations

The cultural change we saw brought about by

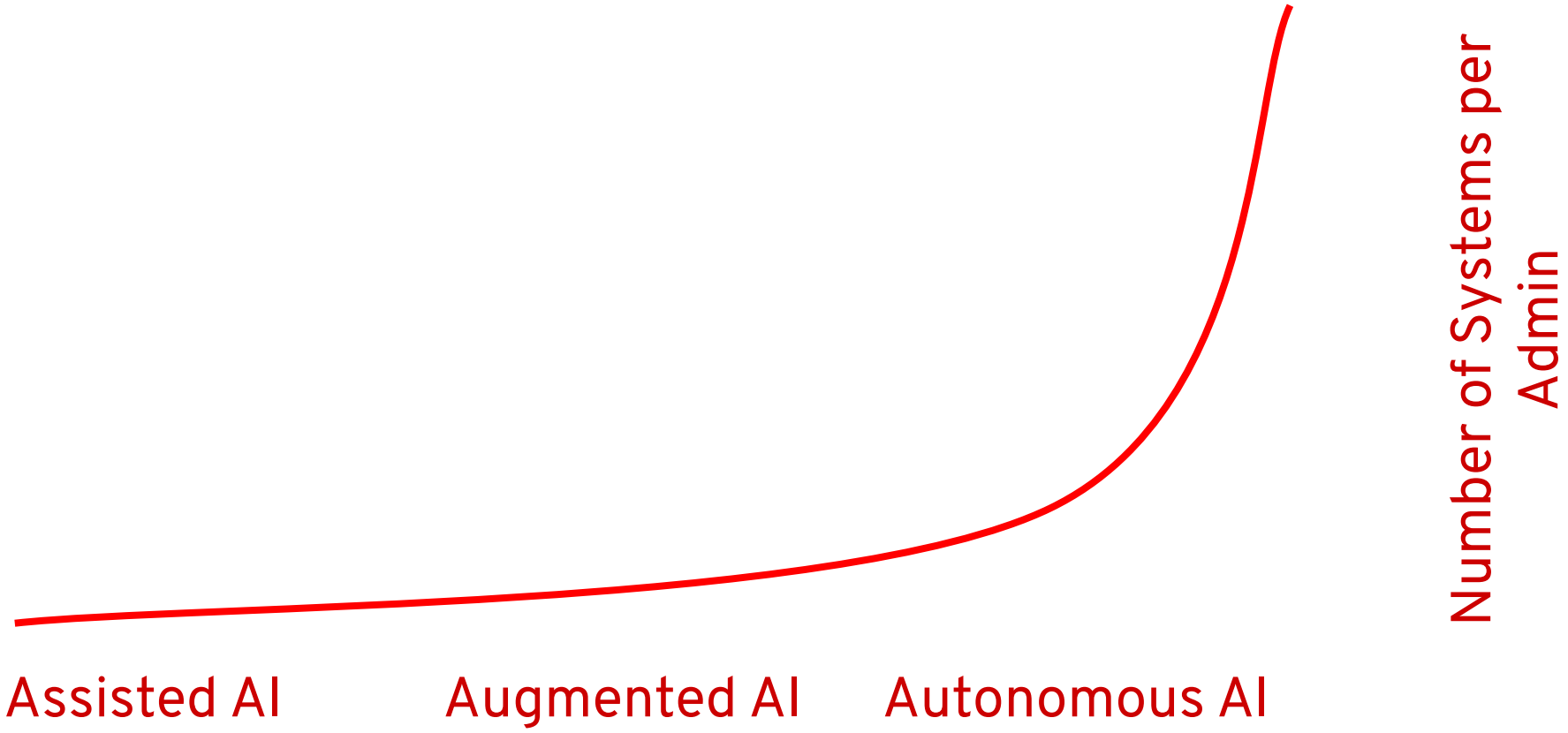
**Dev & Ops == DevOps**

Is being repeated in the combination of

**AI + DevOps == AIOps**

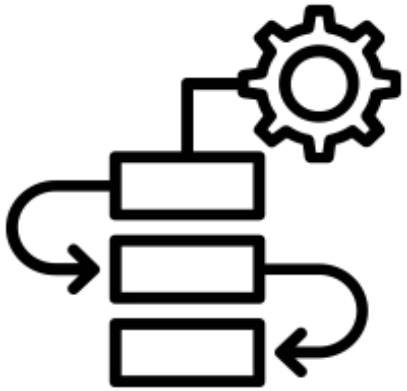
# Increase Efficiency with AI

## The path to the self-driving cluster



# Using Data To Enable The Critical Personas

Automate the open hybrid cloud at scale to democratize state of the art operations.



**OPERATIONS**

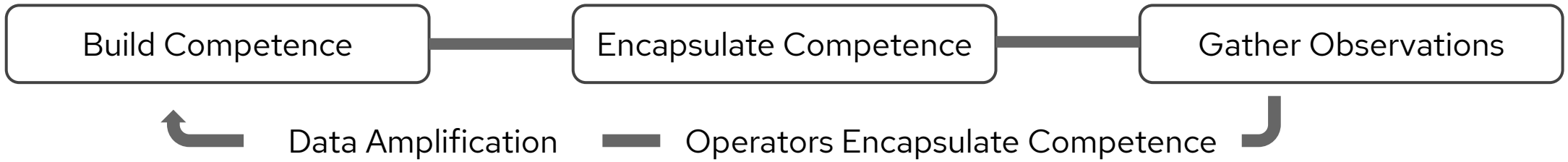
**Objective:** Encapsulated Operational Excellence

Reducing operational cost:

**Analysts indicate 100x cost reduction for operating infrastructure**

- Highly automated
- Using engineering principles to manage infrastructure

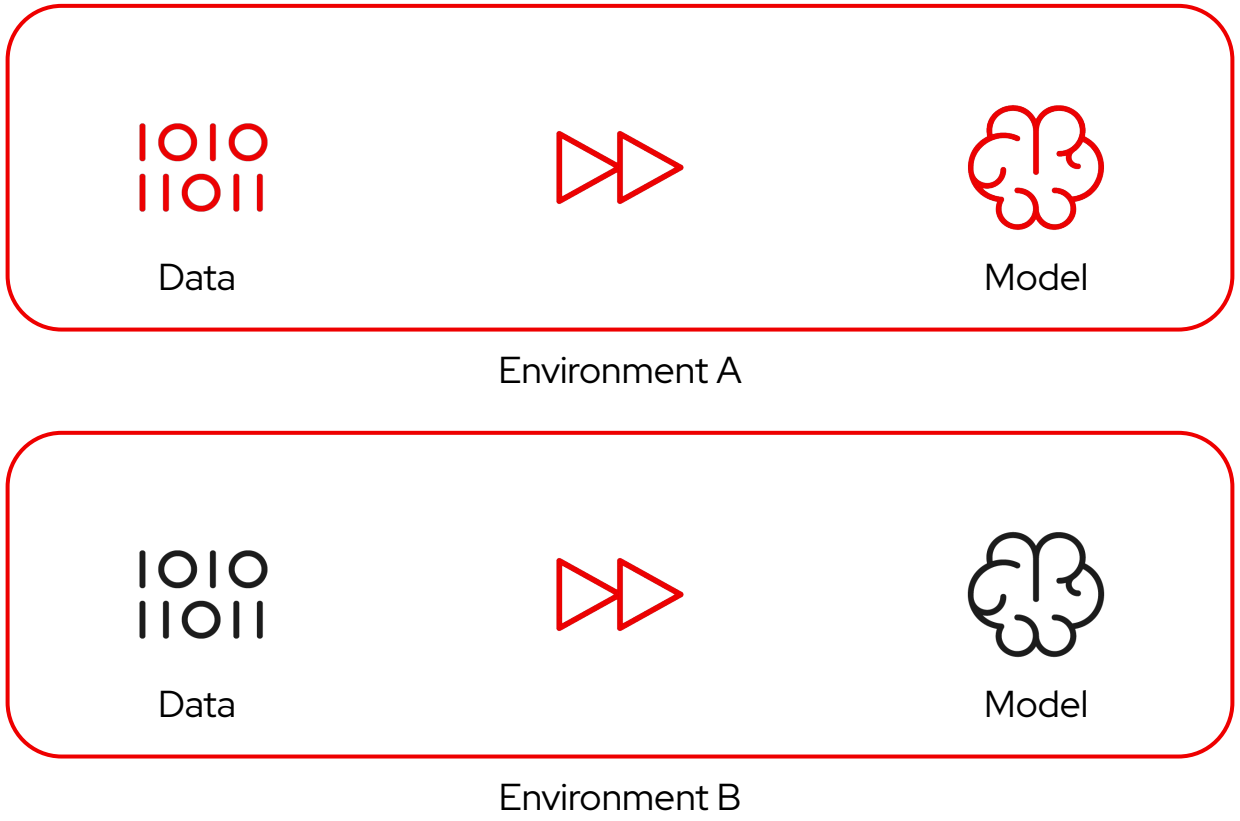
Build cost-efficiency for our customers; encapsulating operation experience in code is the path to get there.






Red Hat

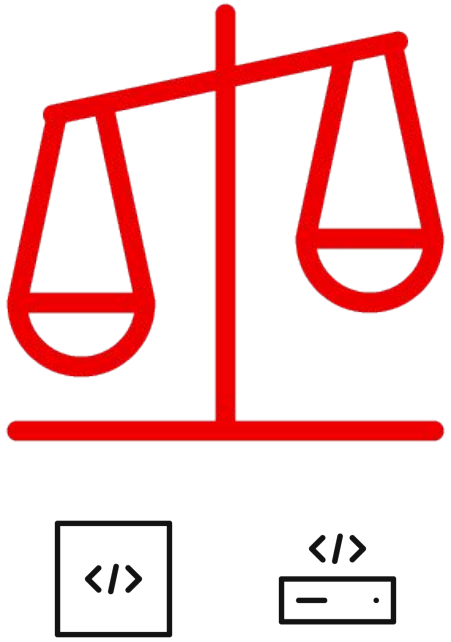
# Traditional Learning



# Once upon a time...

# Before Open Source, Code Was Value

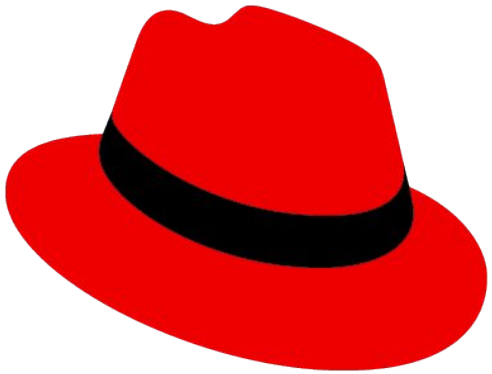
Code



Ops

But operating the code, even at scale, was left to the folks in the basement

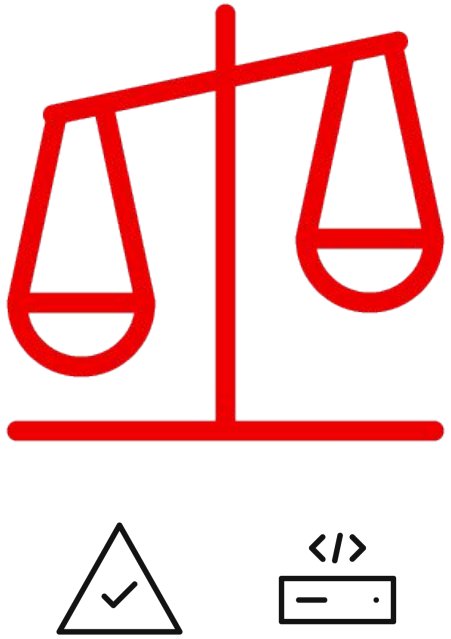
# Then open source happened...



"RALEIGH, N.C. – March 26, 2002 – Red Hat, Inc. (Nasdaq: RHAT) today announced Red Hat Linux Advanced Server, the first enterprise-class Linux operating system."

# … And The Value Moved From The Code To The Product

Product



Ops

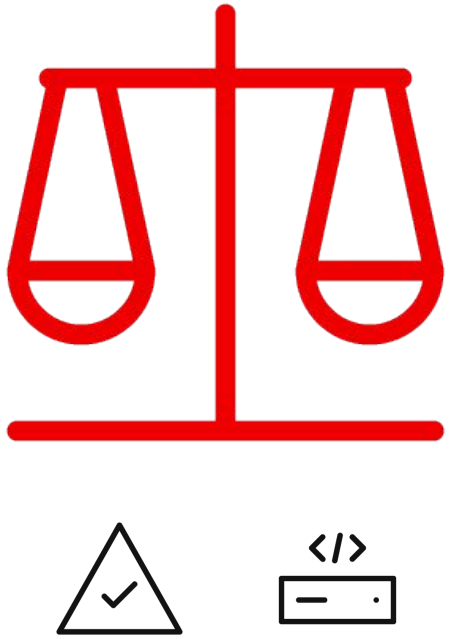
But we still didn't put much value on what those folks in the basement were doing

# Then everything grew like crazy and scale got really really important


"Amazon Web Services (AWS) had $17.46 billion in annual revenue in 2017. By end of 2018, the number had grown to $25.65 billion. AWS reported 37% growth in 2019. In 2019, AWS alone accounted for 12% of Amazon's profits (up from 11% in 2018)."

# Suddenly The Folks In The Basement Are Valuable!

Product

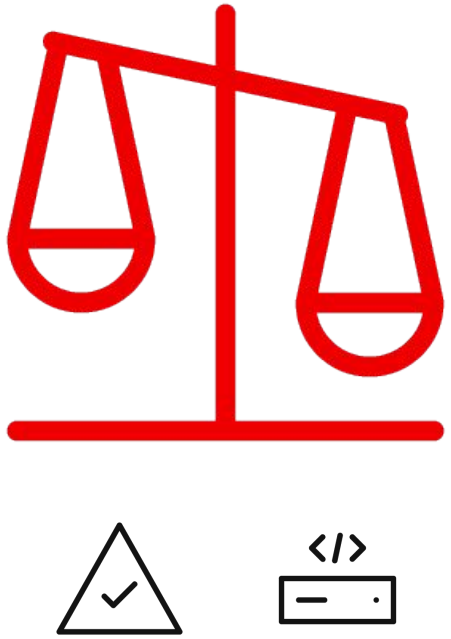


Ops

... And they're not in the basement any more. But like code in the days before open source, the tools and techniques and knowledge of operation at scale are proprietary.

# Suddenly The Folks In The Basement Are MORE Valuable!

Open Source Code



Ops

... And they're not in the basement any more. But like code in the days before open source, the tools and techniques and knowledge of operation at scale are proprietary.

If the value in IT is in ops, and ops are proprietary, then open source has a problem.

Cloud and open source executive with Amazon Web Services (AWS)


Everything == Code

Everything != Ops Platform

In other words, the software was important but not where the compelling value was. If a customer can't use the software, it has no value. The value is in operationalizing that software so the customer can be productive with it.



https://www.infoworld.com/article/3601681/what-happens-when-you-open-source-everything.html
https://twitter.com/mjasay/status/1341805991482966016

"Operate First is an initiative to operate software in a production-grade environment - bringing users, developers and operators closer together.

Ideally Operate First becomes a partner to Upstream First as a basic tenet of our workflow."

# Operate First Is The Solution

With the Mass Open Cloud and Open Infra Labs, Red Hat is launching an effort to open source cloud operations at scale.

## Upstream Projects Operate First

We will open the MOC to upstream communities who need a place to operate their services in order to develop them.

## Red Hat Products Operate First

We will begin operating our own products on the MOC in the open, before we ship them to our customers.

## Open Telemetry, Open Tracing, Open Ops

We will work with the community around the MOC to develop new tools in the open that will be the key to open, autonomous operations at scale.

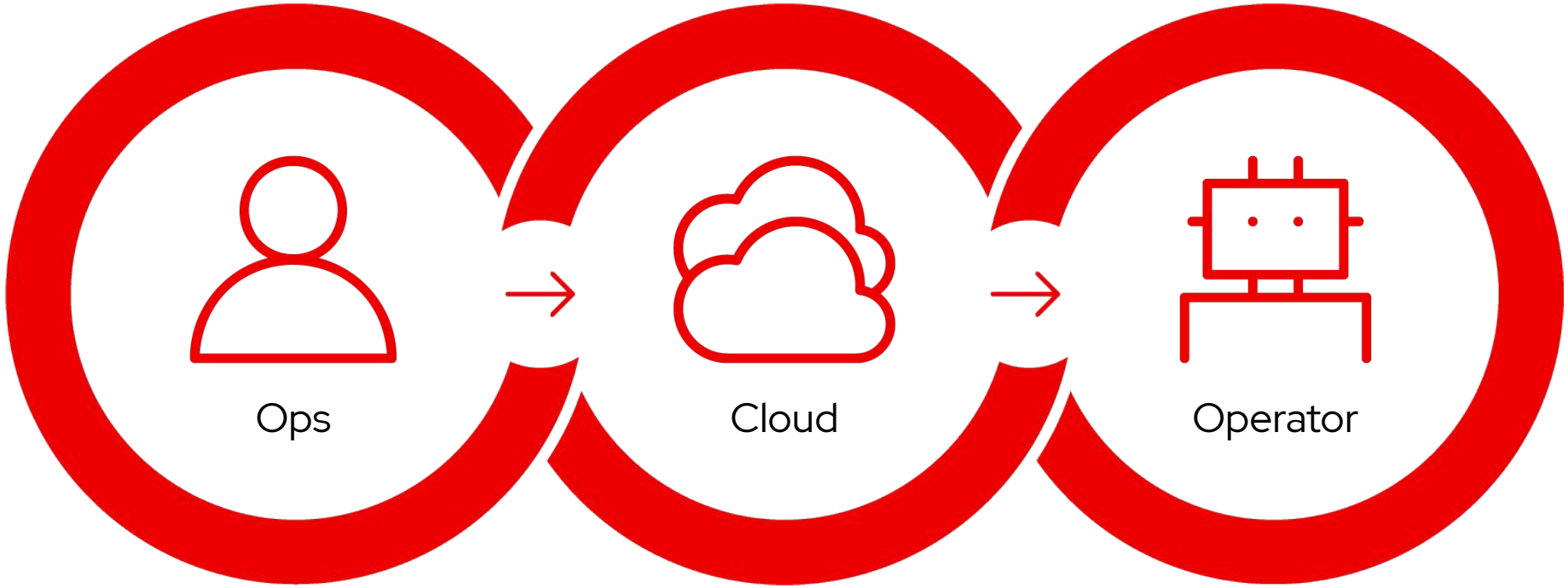


Red Hat


# CLOUD with full visibility into the operations center

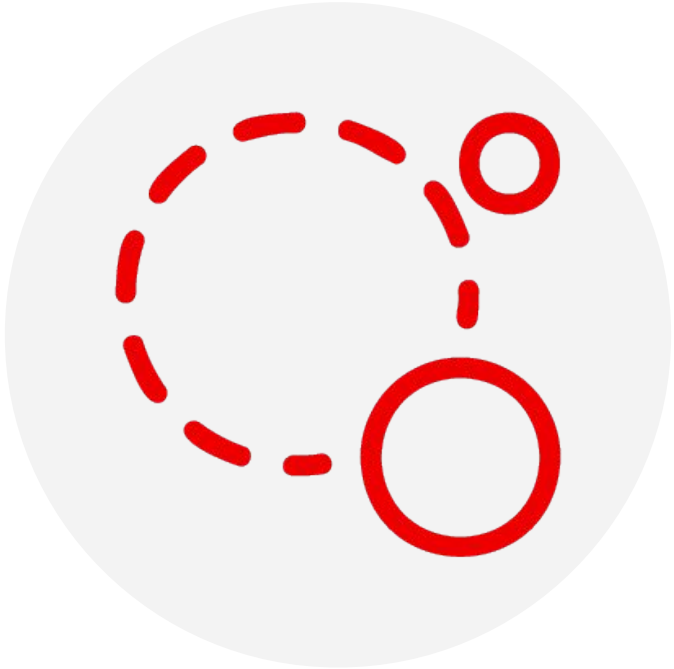
# From Ops to Operators

## Operators are codified operational knowledge



# The Power of Open Source

## Turn Users into Contributors



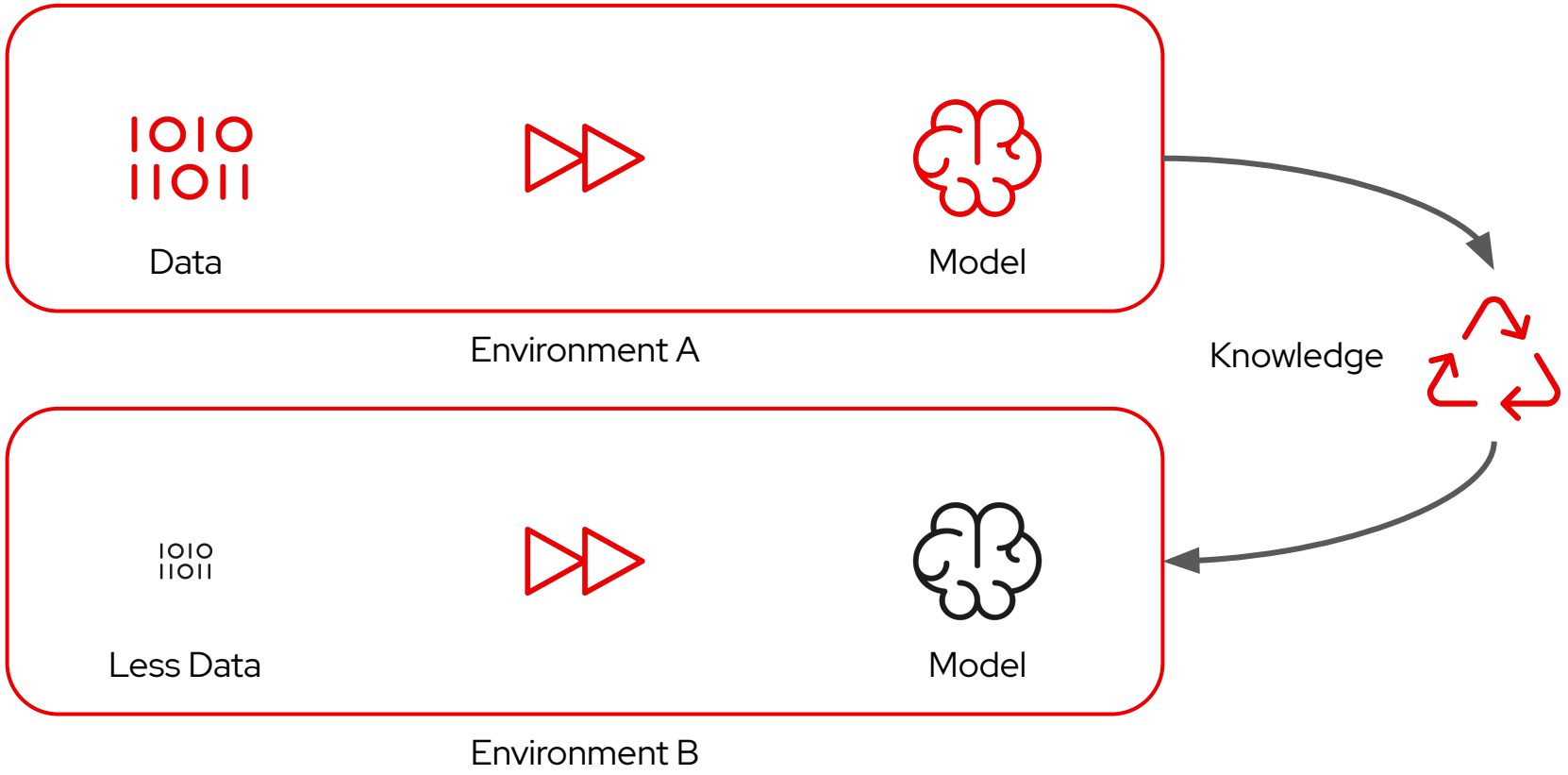
Read Only Access to **all** the Data

Easy onboarding

Read → Report → Resolve

# Beyond AIOps

# Transfer Learning



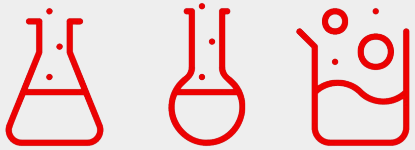
# AIOps in a Community

Discussion
Collaboration
Standards

Grow collectively
Codify operational experience
Democratize operations

data as a competitive differentiator

Leverage the power of Many



# OpenDataHub

## Operate First

Common history with Operate First Team

Young project

Users, Users, Users

e.g. ACM/OCP/M4D/IBM/CNV are other workloads

# Call to Action

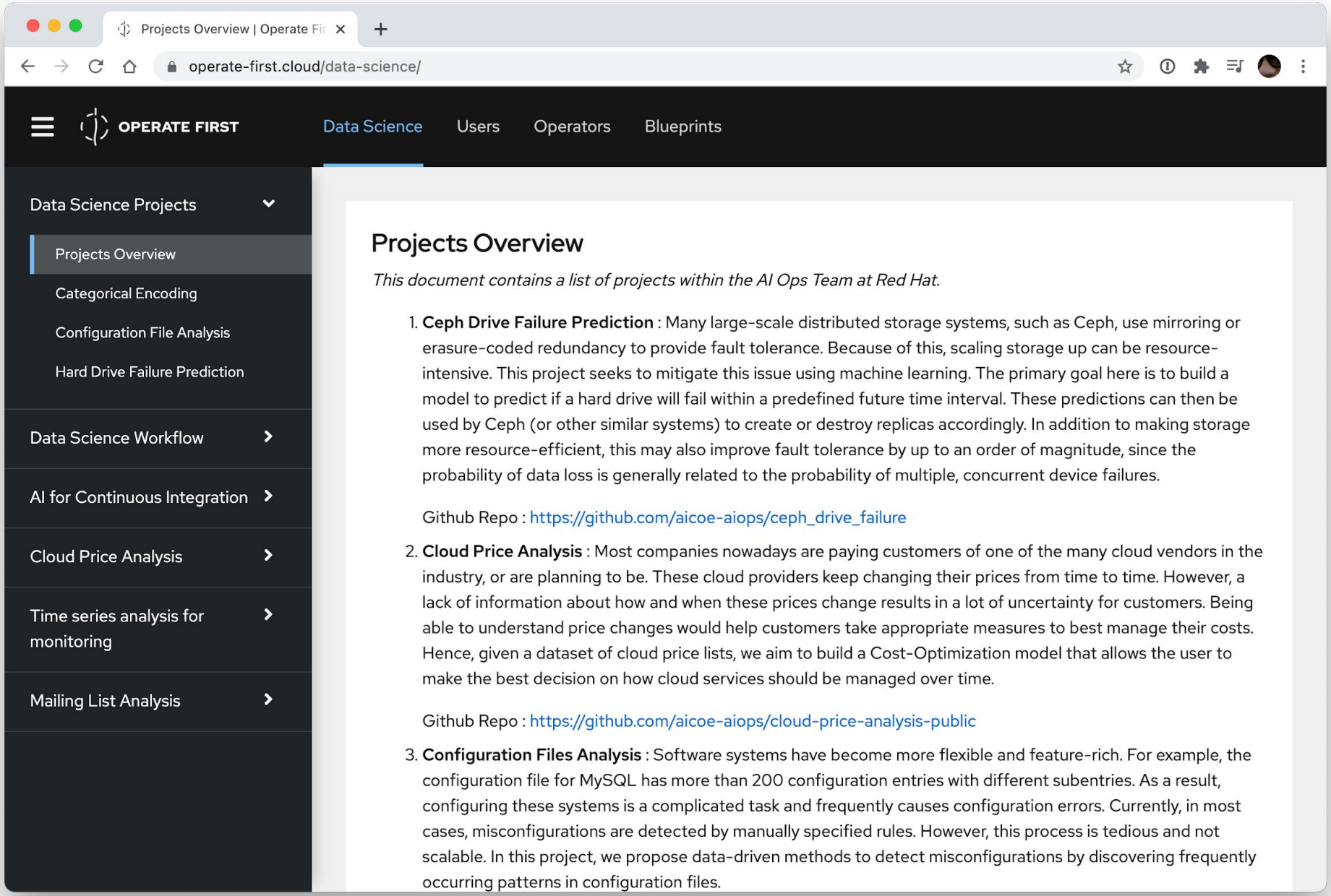
Community means collaboration

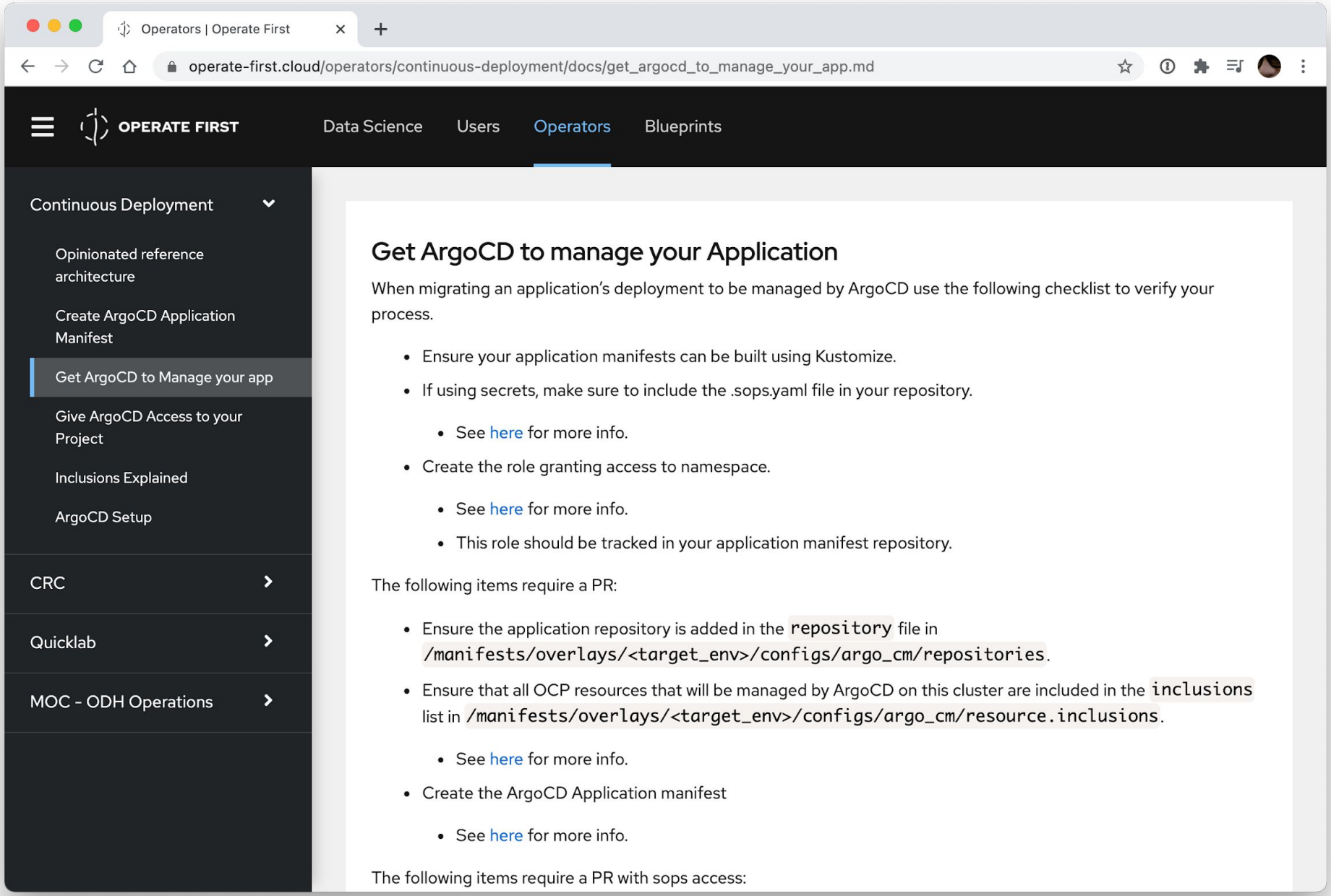
Access All Areas

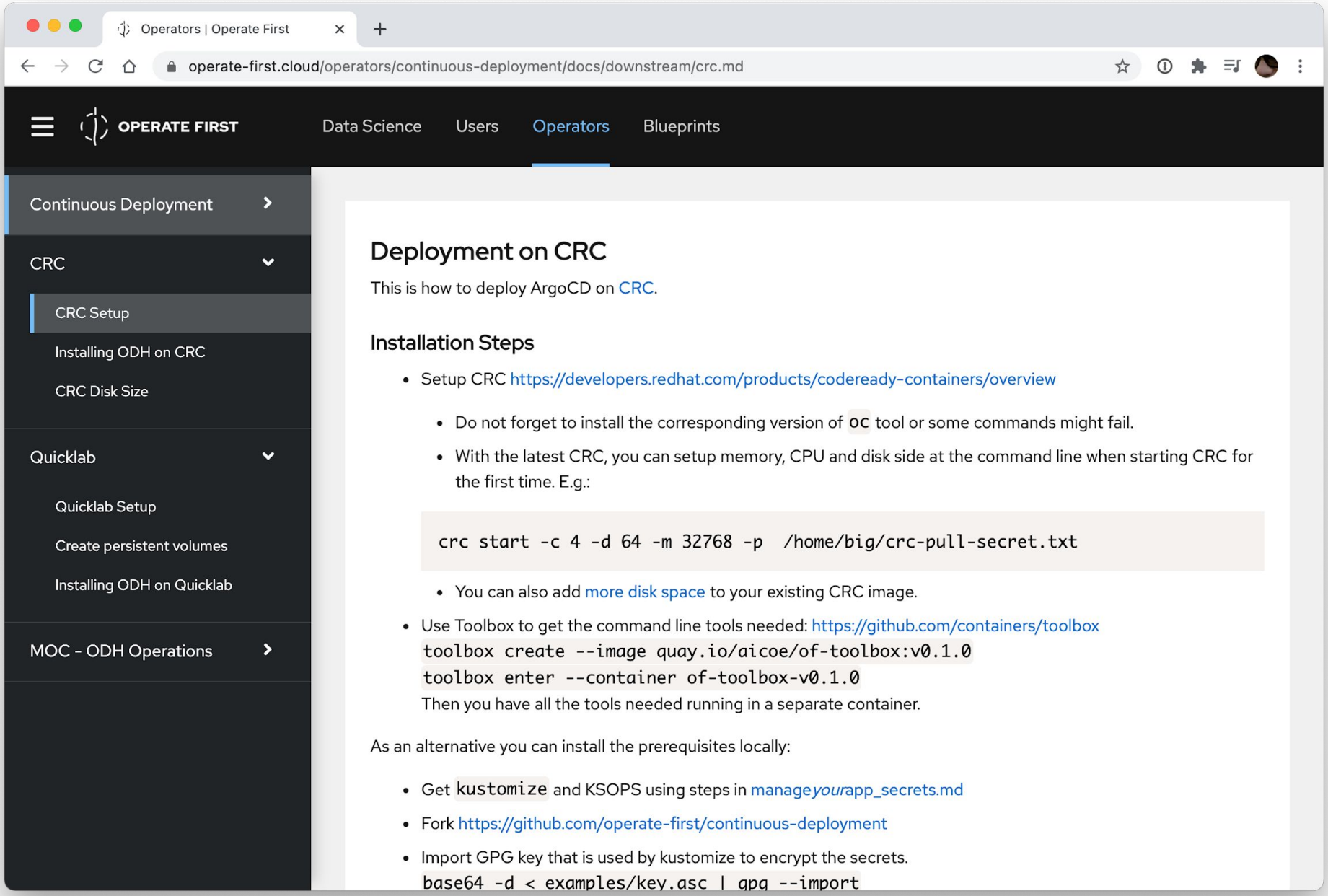
Deploy Demos

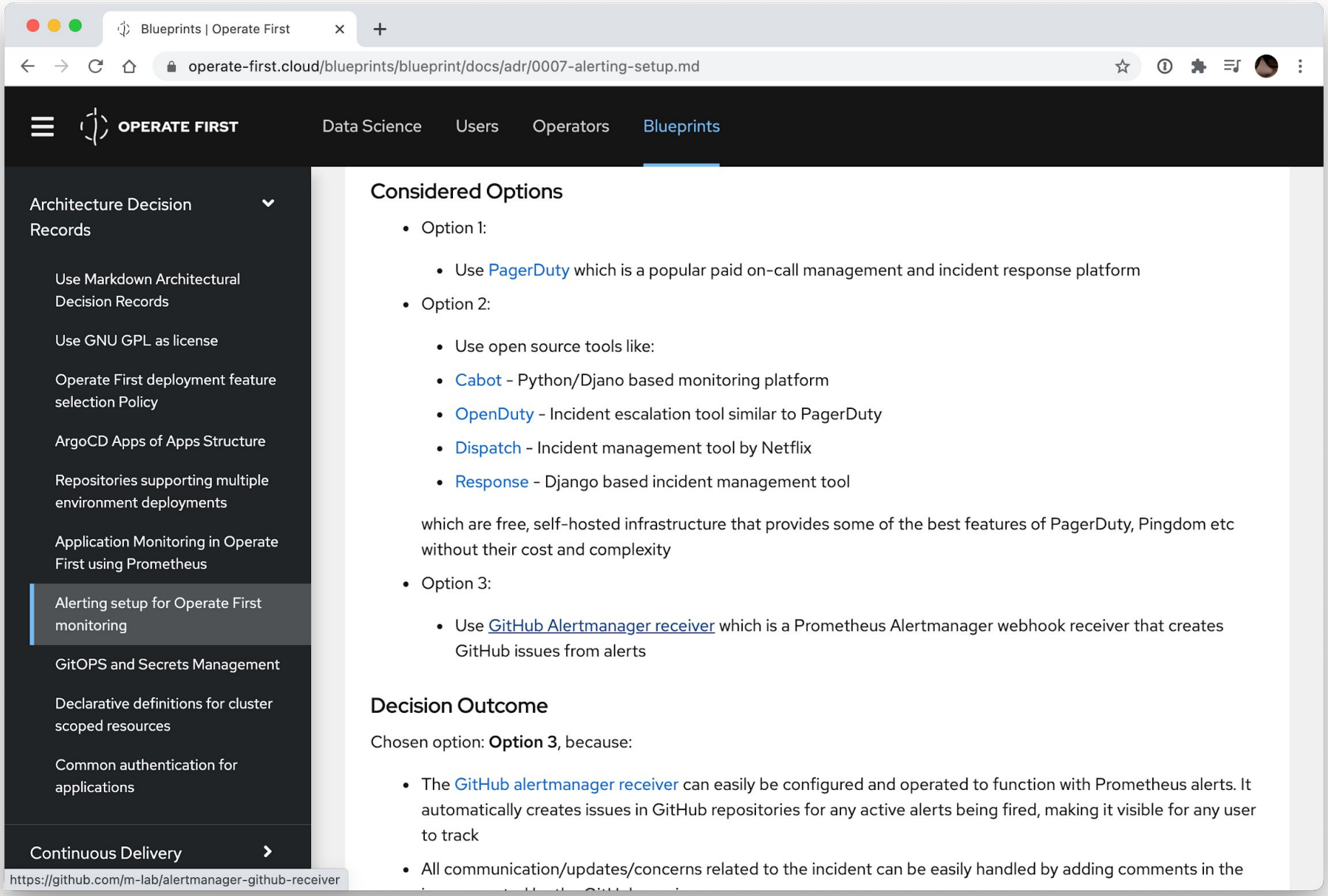
Solve Issues

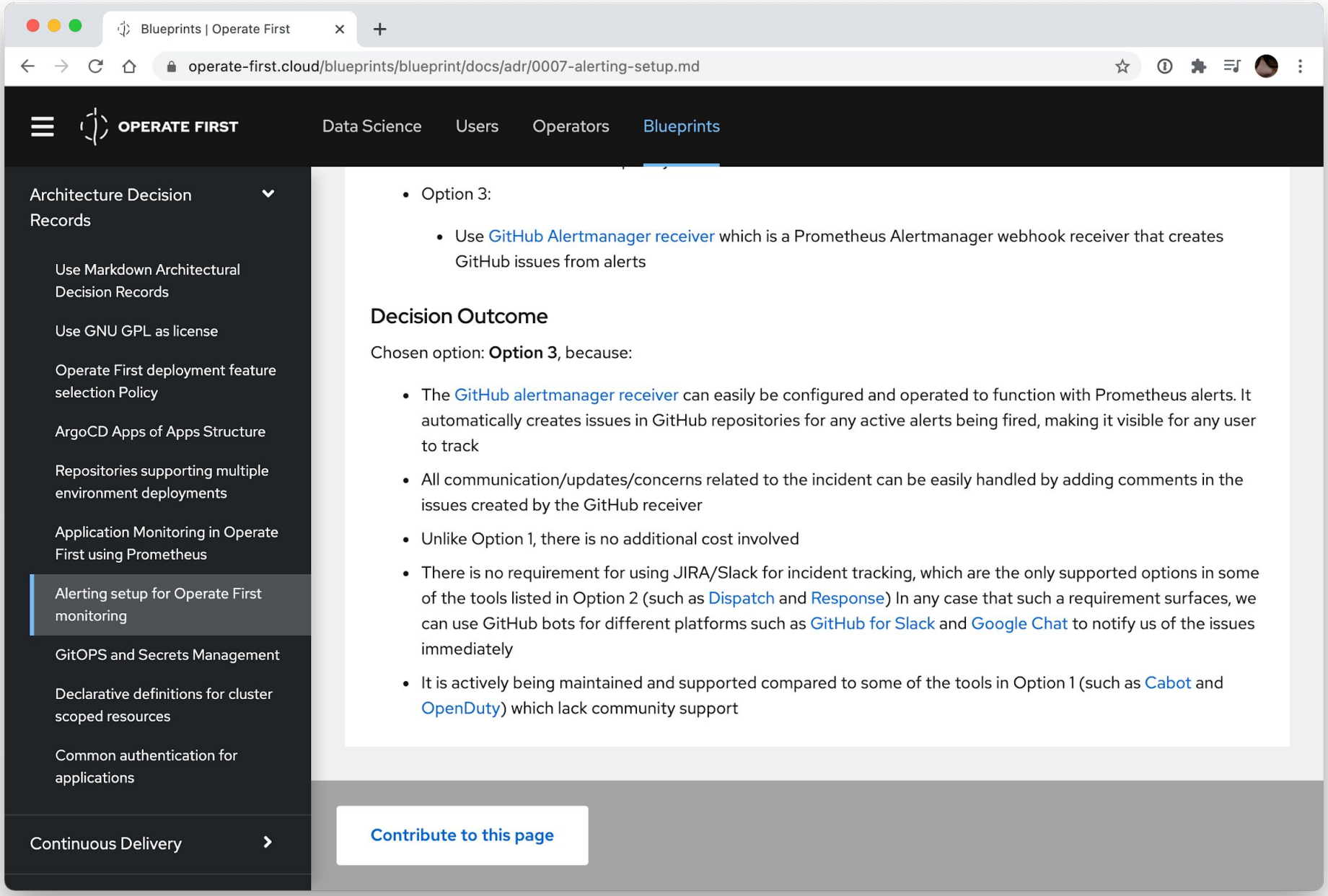
operate-first.cloud


Projects Overview | Operate Fir
operate-first.cloud/data-science/
OPERATE FIRST
Data Science
Users
Operators
Blueprints
Data Science Projects
Projects Overview
Categorical Encoding
Configuration File Analysis
Hard Drive Failure Prediction
Data Science Workflow
AI for Continuous Integration
Cloud Price Analysis
Time series analysis for monitoring
Mailing List Analysis
Projects Overview
This document contains a list of projects within the AI Ops Team at Red Hat.
1. Ceph Drive Failure Prediction : Many large-scale distributed storage systems, such as Ceph, use mirroring or erasure-coded redundancy to provide fault tolerance. Because of this, scaling storage up can be resource-intensive. This project seeks to mitigate this issue using machine learning. The primary goal here is to build a model to predict if a hard drive will fail within a predefined future time interval. These predictions can then be used by Ceph (or other similar systems) to create or destroy replicas accordingly. In addition to making storage more resource-efficient, this may also improve fault tolerance by up to an order of magnitude, since the probability of data loss is generally related to the probability of multiple, concurrent device failures.
Github Repo : https://github.com/aicoe-aiops/ceph_drive_failure
2. Cloud Price Analysis : Most companies nowadays are paying customers of one of the many cloud vendors in the industry, or are planning to be. These cloud providers keep changing their prices from time to time. However, a lack of information about how and when these prices change results in a lot of uncertainty for customers. Being able to understand price changes would help customers take appropriate measures to best manage their costs. Hence, given a dataset of cloud price lists, we aim to build a Cost-Optimization model that allows the user to make the best decision on how cloud services should be managed over time.
Github Repo : https://github.com/aicoe-aiops/cloud-price-analysis-public
3. Configuration Files Analysis : Software systems have become more flexible and feature-rich. For example, the configuration file for MySQL has more than 200 configuration entries with different subentries. As a result, configuring these systems is a complicated task and frequently causes configuration errors. Currently, in most cases, misconfigurations are detected by manually specified rules. However, this process is tedious and not scalable. In this project, we propose data-driven methods to detect misconfigurations by discovering frequently occurring patterns in configuration files.

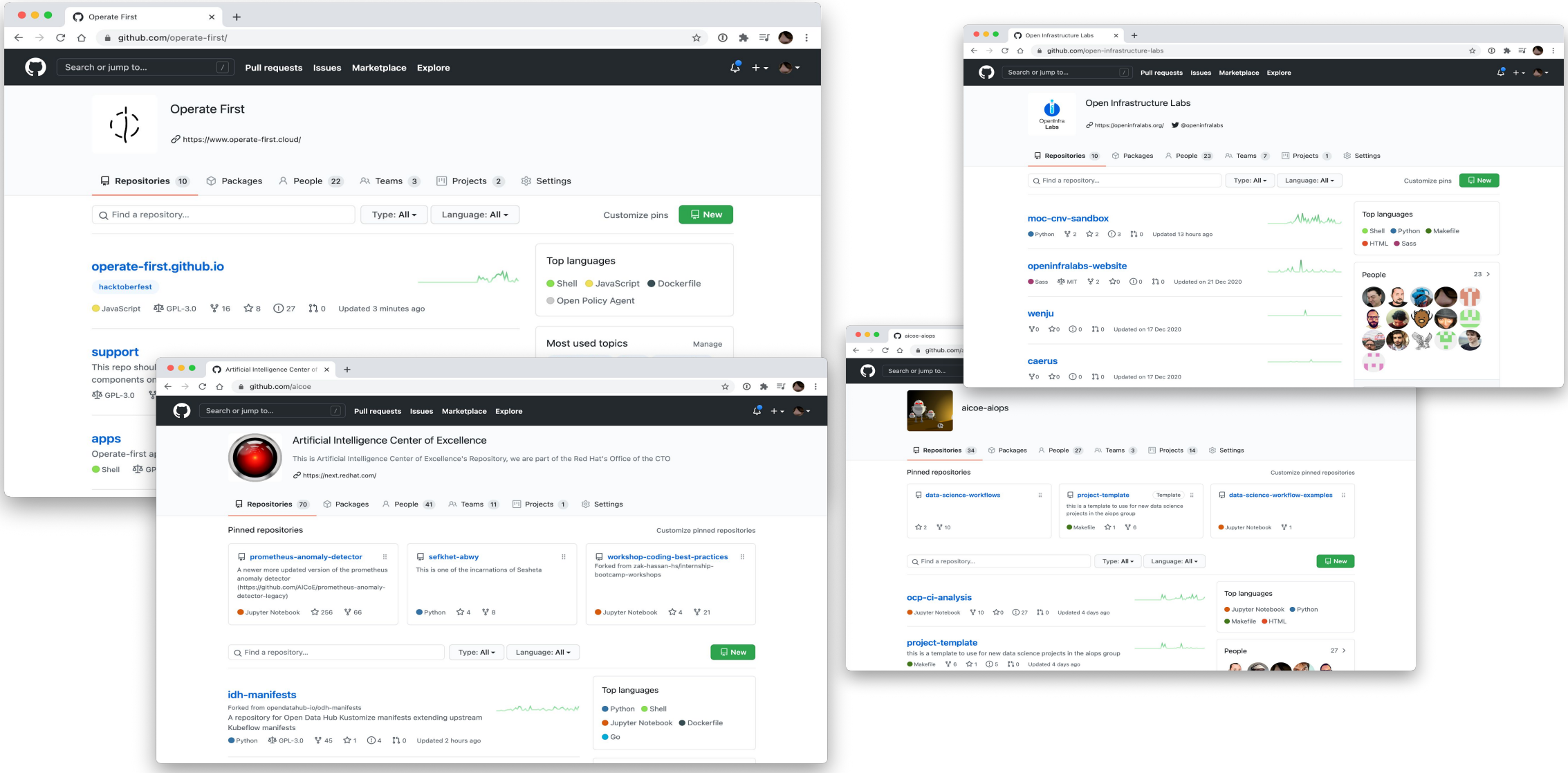

Operators | Operate First
operate-first.cloud/operators/continuous-deployment/docs/get_argocd_to_manage_your_app.md
OPERATE FIRST
Data Science
Users
Operators
Blueprints
Continuous Deployment
Opinionated reference architecture
Create ArgoCD Application Manifest
Get ArgoCD to Manage your app
Give ArgoCD Access to your Project
Inclusions Explained
ArgoCD Setup
CRC
Quicklab
MOC - ODH Operations
Get ArgoCD to manage your Application
When migrating an application's deployment to be managed by ArgoCD use the following checklist to verify your process.
Ensure your application manifests can be built using Kustomize.
If using secrets, make sure to include the .sops.yaml file in your repository.
See here for more info.
Create the role granting access to namespace.
See here for more info.
This role should be tracked in your application manifest repository.
The following items require a PR:
Ensure the application repository is added in the repository file in /manifests/overlays/<target_env>/configs/argo_cm/repositories.
Ensure that all OCP resources that will be managed by ArgoCD on this cluster are included in the inclusions list in /manifests/overlays/<target_env>/configs/argo_cm/resource.inclusions.
See here for more info.
Create the ArgoCD Application manifest
See here for more info.
The following items require a PR with sops access:


Operators | Operate First
operate-first.cloud/operators/continuous-deployment/docs/downstream/crc.md
OPERATE FIRST
Data Science
Users
Operators
Blueprints
Continuous Deployment
CRC
CRC Setup
Installing ODH on CRC
CRC Disk Size
Quicklab
Quicklab Setup
Create persistent volumes
Installing ODH on Quicklab
MOC - ODH Operations
Deployment on CRC
This is how to deploy ArgoCD on CRC.
Installation Steps
Setup CRC https://developers.redhat.com/products/codeready-containers/overview
Do not forget to install the corresponding version of oc tool or some commands might fail.
With the latest CRC, you can setup memory, CPU and disk side at the command line when starting CRC for the first time. E.g.:
crc start -c 4 -d 64 -m 32768 -p /home/big/crc-pull-secret.txt
You can also add more disk space to your existing CRC image.
Use Toolbox to get the command line tools needed: https://github.com/containers/toolbox
toolbox create --image quay.io/aicoe/of-toolbox:v0.1.0
toolbox enter --container of-toolbox-v0.1.0
Then you have all the tools needed running in a separate container.
As an alternative you can install the prerequisites locally:
Get kustomize and KSOPS using steps in manageyourapp_secrets.md
Fork https://github.com/operate-first/continuous-deployment
Import GPG key that is used by kustomize to encrypt the secrets.
base64 -d < examples/key.asc | gpg --import


Blueprints | Operate First
operate-first.cloud/blueprints/blueprint/docs/adr/0007-alerting-setup.md
OPERATE FIRST
Data Science
Users
Operators
Blueprints
Architecture Decision Records
Use Markdown Architectural Decision Records
Use GNU GPL as license
Operate First deployment feature selection Policy
ArgoCD Apps of Apps Structure
Repositories supporting multiple environment deployments
Application Monitoring in Operate First using Prometheus
Alerting setup for Operate First monitoring
GitOPS and Secrets Management
Declarative definitions for cluster scoped resources
Common authentication for applications
Continuous Delivery
https://github.com/m-lab/alertmanager-github-receiver
Considered Options
Option 1:
Use PagerDuty which is a popular paid on-call management and incident response platform
Option 2:
Use open source tools like:
Cabot - Python/Djano based monitoring platform
OpenDuty - Incident escalation tool similar to PagerDuty
Dispatch - Incident management tool by Netflix
Response - Django based incident management tool
which are free, self-hosted infrastructure that provides some of the best features of PagerDuty, Pingdom etc without their cost and complexity
Option 3:
Use GitHub Alertmanager receiver which is a Prometheus Alertmanager webhook receiver that creates GitHub issues from alerts
Decision Outcome
Chosen option: Option 3, because:
The GitHub alertmanager receiver can easily be configured and operated to function with Prometheus alerts. It automatically creates issues in GitHub repositories for any active alerts being fired, making it visible for any user to track
All communication/updates/concerns related to the incident can be easily handled by adding comments in the


Blueprints | Operate First
operate-first.cloud/blueprints/blueprint/docs/adr/0007-alerting-setup.md
OPERATE FIRST
Data Science
Users
Operators
Blueprints
Architecture Decision Records
Use Markdown Architectural Decision Records
Use GNU GPL as license
Operate First deployment feature selection Policy
ArgoCD Apps of Apps Structure
Repositories supporting multiple environment deployments
Application Monitoring in Operate First using Prometheus
Alerting setup for Operate First monitoring
GitOPS and Secrets Management
Declarative definitions for cluster scoped resources
Common authentication for applications
Continuous Delivery
Option 3:
Use GitHub Alertmanager receiver which is a Prometheus Alertmanager webhook receiver that creates GitHub issues from alerts
Decision Outcome
Chosen option: Option 3, because:
The GitHub alertmanager receiver can easily be configured and operated to function with Prometheus alerts. It automatically creates issues in GitHub repositories for any active alerts being fired, making it visible for any user to track
All communication/updates/concerns related to the incident can be easily handled by adding comments in the issues created by the GitHub receiver
Unlike Option 1, there is no additional cost involved
There is no requirement for using JIRA/Slack for incident tracking, which are the only supported options in some of the tools listed in Option 2 (such as Dispatch and Response) In any case that such a requirement surfaces, we can use GitHub bots for different platforms such as GitHub for Slack and Google Chat to notify us of the issues immediately
It is actively being maintained and supported compared to some of the tools in Option 1 (such as Cabot and OpenDuty) which lack community support
Contribute to this page

# Operate First






Red Hat

# Thank you

linkedin.com/in/marcelhild

twitter.com/durandom

http://operate-first.cloud



Red Hat