Transcending UI Limitations...

Custom UI Solutions architected within a Console Framework

Quiana Berry
OCP Product Manager Intern

Ali Mobrem
Principal Product Manager

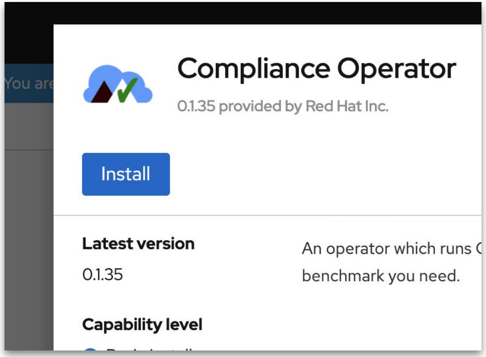
AGENDA

- ❖ Why Should You Care?
  - ➢ Use Case
- ❖ Ways you can Customize on OCP
- ❖ Problem
  - ➢ History of Static Plugins
  - ➢ Solution
- ❖ Who is the Solution for?
  - ➢ Internal/External Stakeholders
- ❖ Value Add ?
  - ➢ Internal
  - ➢ External
- ❖ Benefits of the Console Framework
- ❖ Operators
- ❖ What are Plugins
- ❖ Plugin Architecture
  - ➢ Anatomy of Plugin Framework
  - ➢ Technical Breakdown
  - ➢ Console/ Red Hat Plugins
- ❖ How to Enable Plugins
  - ➢ Mock Ups
  - ➢ Demo
- ❖ Roadmap projection
- ❖ Feedback

Why Should You Care?

Compliance Officer

Install



You are

Compliance Operator

0.1.35 provided by Red Hat Inc.

Install

Latest version

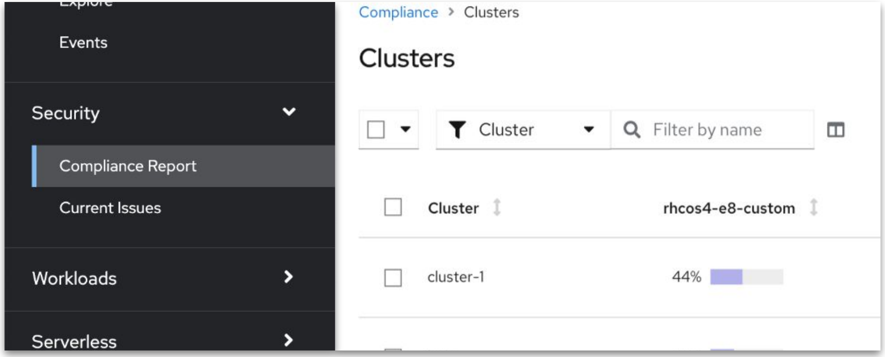
0.1.35

An operator which runs C

benchmark you need.

Capability level

Security Nav Auto - Generated



Explore

Events

Security

Compliance Report

Current Issues

Workloads

Serverless

Compliance › Clusters

Clusters

Cluster

Filter by name

☐	Cluster	rhcos4-e8-custom
☐	cluster-1	44%

Quiana Berry: PM Intern

The Console as your Canvas....

- Add navigation items
- Add YAML templates
- Add alert notification

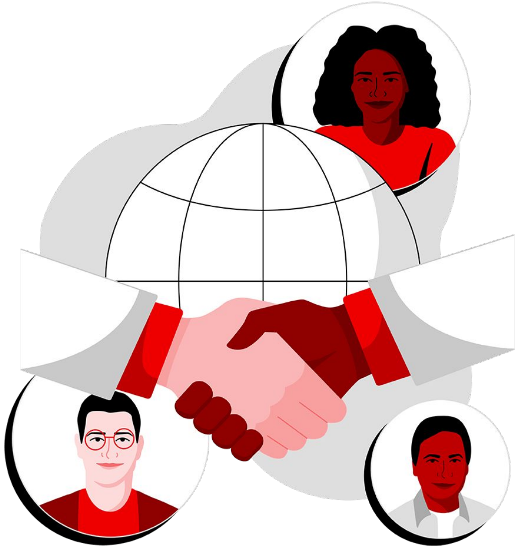
- Add topology nodes and edges
- Add developer catalog items

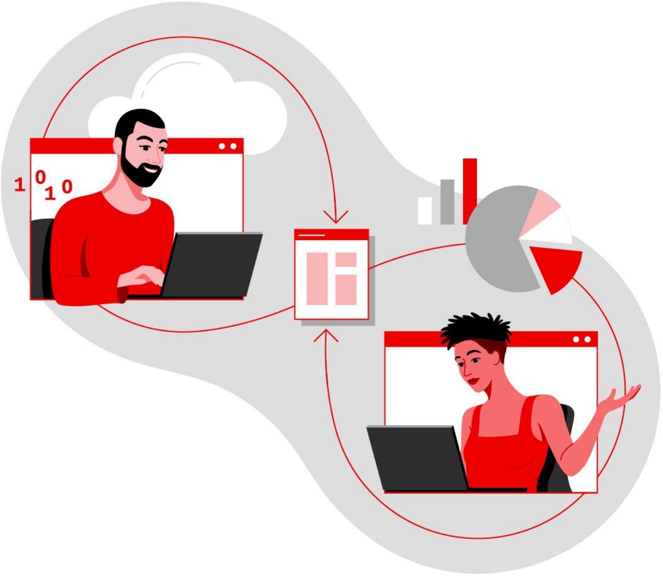
- Add custom pages and routes
- Override default resource pages

- Capture telemetry
- Add dashboards
- Add resource actions

- ... and more!

How Would You Customize Your Experience on the Console?





Problems

Difficulty with layering products seamlessly & reducing redundancy

Slower time to market and less control on UI experience for OCP users

Monorepos are hard to scale with growing number of teams

→

Solution:

Console Framework

UI extensions within a single-pane of glass that enables reproducible UIs

Enable customers to customize UIs on their own release cadence

Dynamic Plugins enable quicker scaling of internal & external teams

Quiana Berry: PM Intern

Problems

Half-dozen teams contribute to the console on a monorepo

Some teams have integration gaps

Monorepos are hard to scale with growing number of teams

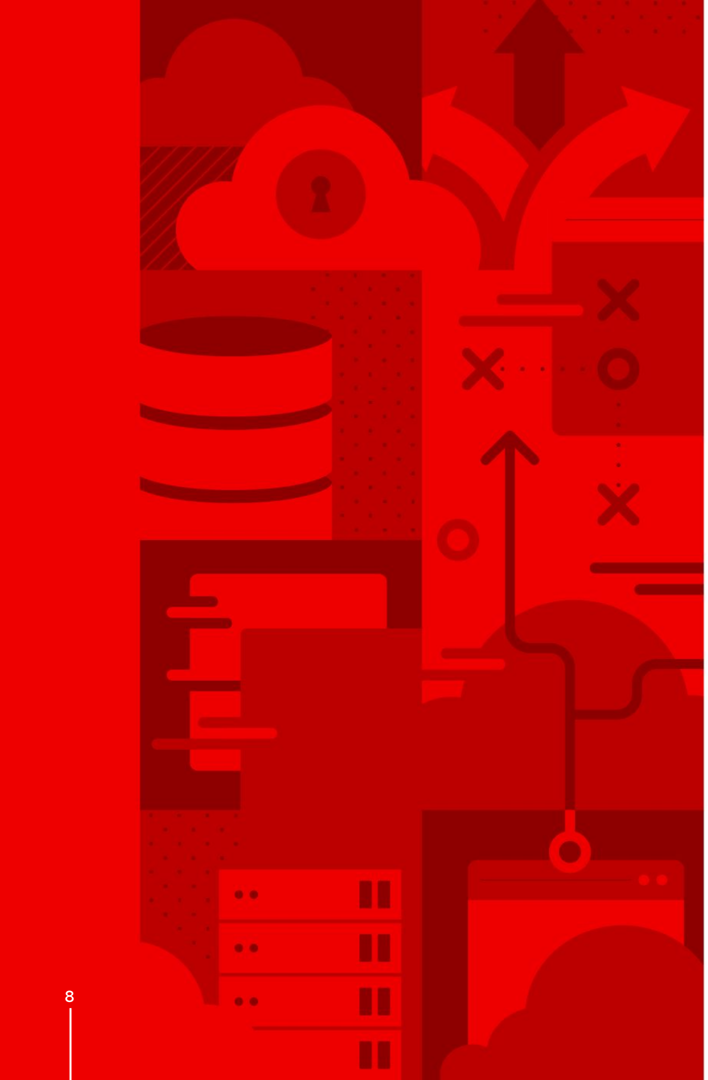
Console Framework Solution

Enable teams to release on their own release cadence

Console Framework bridges integration gaps by extending UI within a single-pane of glass

Dynamic Plugins enable scaling of internal teams

Quiana Berry: PM Intern



What are Dynamic Plugins?

Dynamic Plugins are a combination of metadata, code, and extensions delivered on the console through an operator

Quiana Berry: PM Intern

Who is this Solution for?

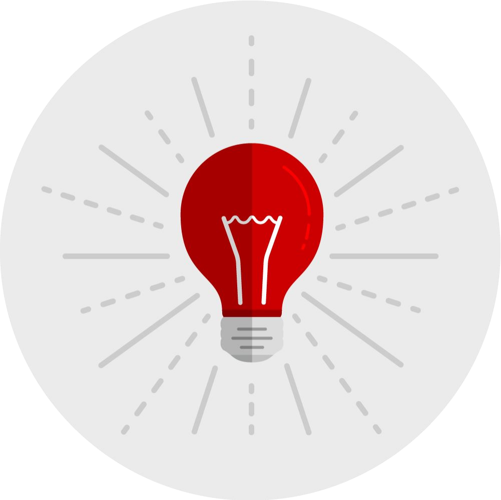
Internal Stakeholder

- OCP Core Plugin Team
- Red Hat Plugin Team

External Stakeholder

- Partner Plugin Team
- Customers Plugin Team

Win-Win For All Stakeholders



Internal Value Add - PM

Improved integration of layered products -> quicker build time with Console Framework -> quicker delivery -> enhanced customer experiences on OCP

Internal Value Add - Engineers/Devs

Shift responsibility from internal engineer teams to external teams so they can scale quicker

External Value Add - Customers

Freedom to build and customize experiences according to their individual needs

Quiana Berry: PM Intern

Benefits of the Console Framework Solution

Contribute UI to the console separate from the console release schedule

Resource based access control checks to provide users with personalized UI

Consistent unified tooling within a single console UI

Plugin enablement controlled by admin

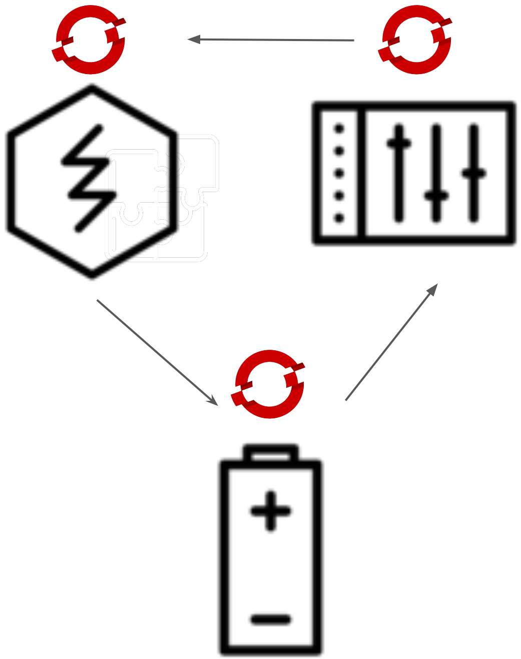
SDK for common components and utilities

Feature flagged UI for progressive app

Direct access to Kubernetes API

Extend UI contributed by other plugins

Operators: Power the Console Framework

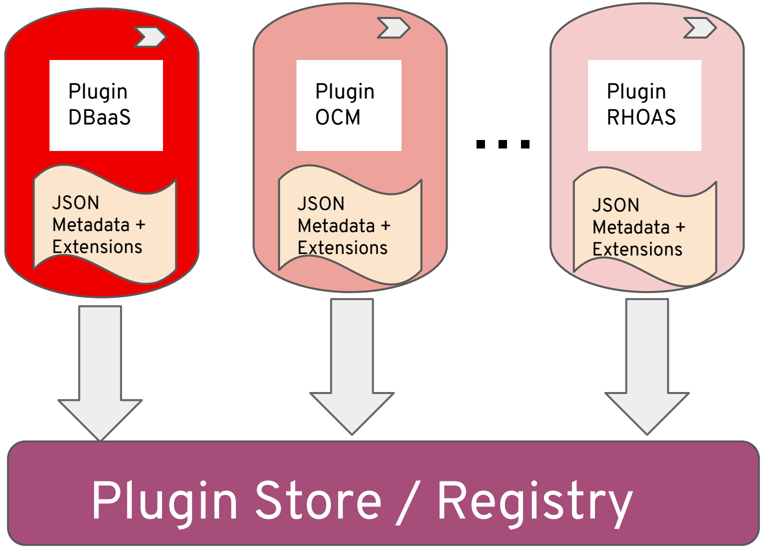


01 Enable UI Extensions

02 Enable “Single Pane of Glass”

03 Enable Layering Products

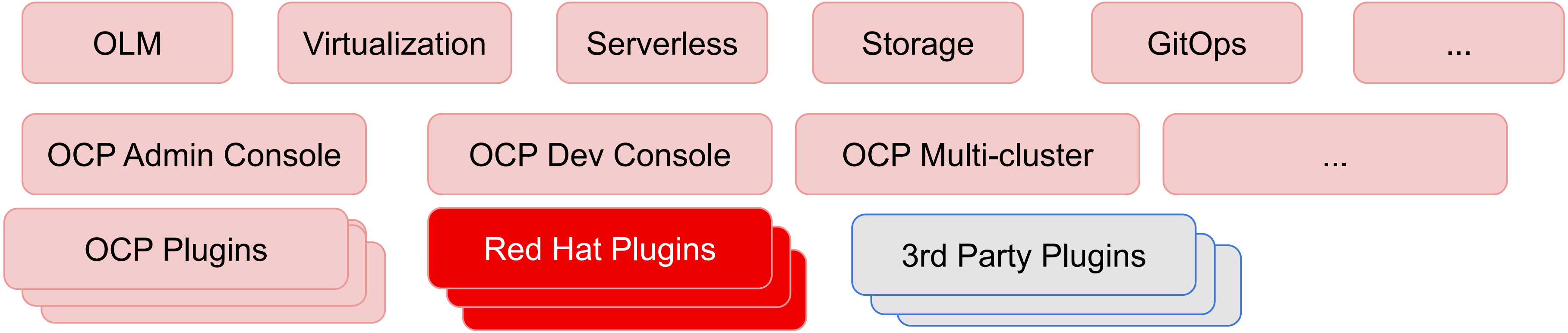
What Are Plugins?



Plugins are pieces of software that are integrated with metadata, to the platform to "extend" the capabilities of OCP

Quiana Berry: PM Intern

Console Core Plugins



OLM

Virtualization

Serverless

Storage

GitOps

...

OCP Admin Console

OCP Dev Console

OCP Multi-cluster

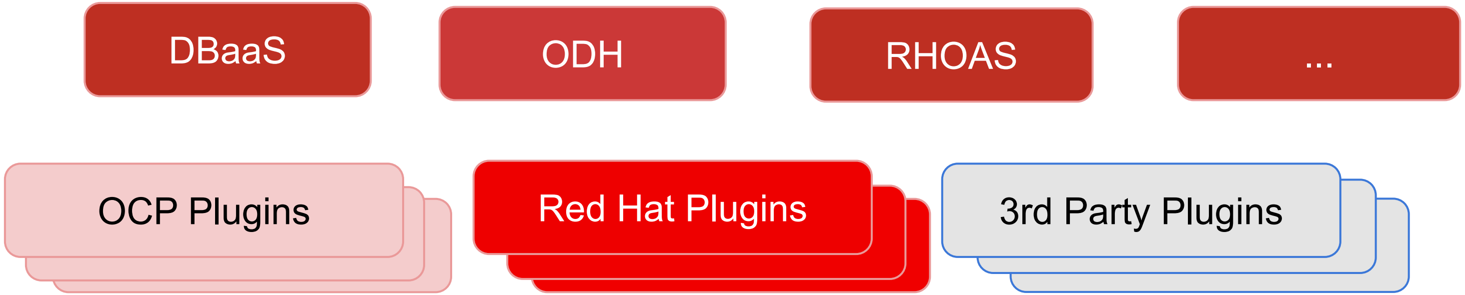
...

OCP Plugins

Red Hat Plugins

3rd Party Plugins

Red Hat Plugins



DBaaS

ODH

RHOAS

...

OCP Plugins

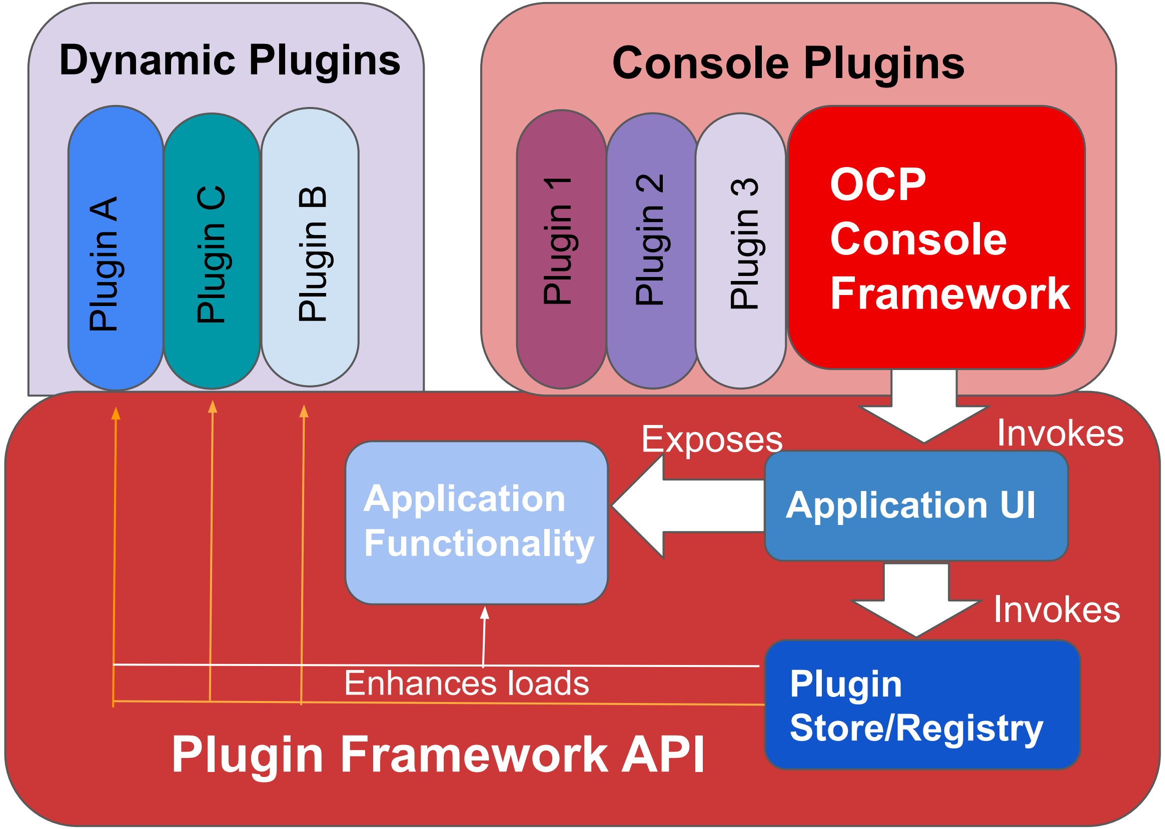
Red Hat Plugins

3rd Party Plugins

Plugin Architecture

A Console Framework
Architected for your Success...

Anatomy of the Plugin Framework



Dynamic Plugins

Plugin A

Plugin C

Plugin B

Console Plugins

Plugin 1

Plugin 2

Plugin 3

OCP Console Framework

Invokes

Exposes

Application Functionality

Application UI

Invokes

Plugin Store/Registry

Enhances loads

Plugin Framework API

Quiana Berry: PM Intern

2 Use Cases - 1 Framework

Hybrid

Use-Case

Enhance the existing OCP

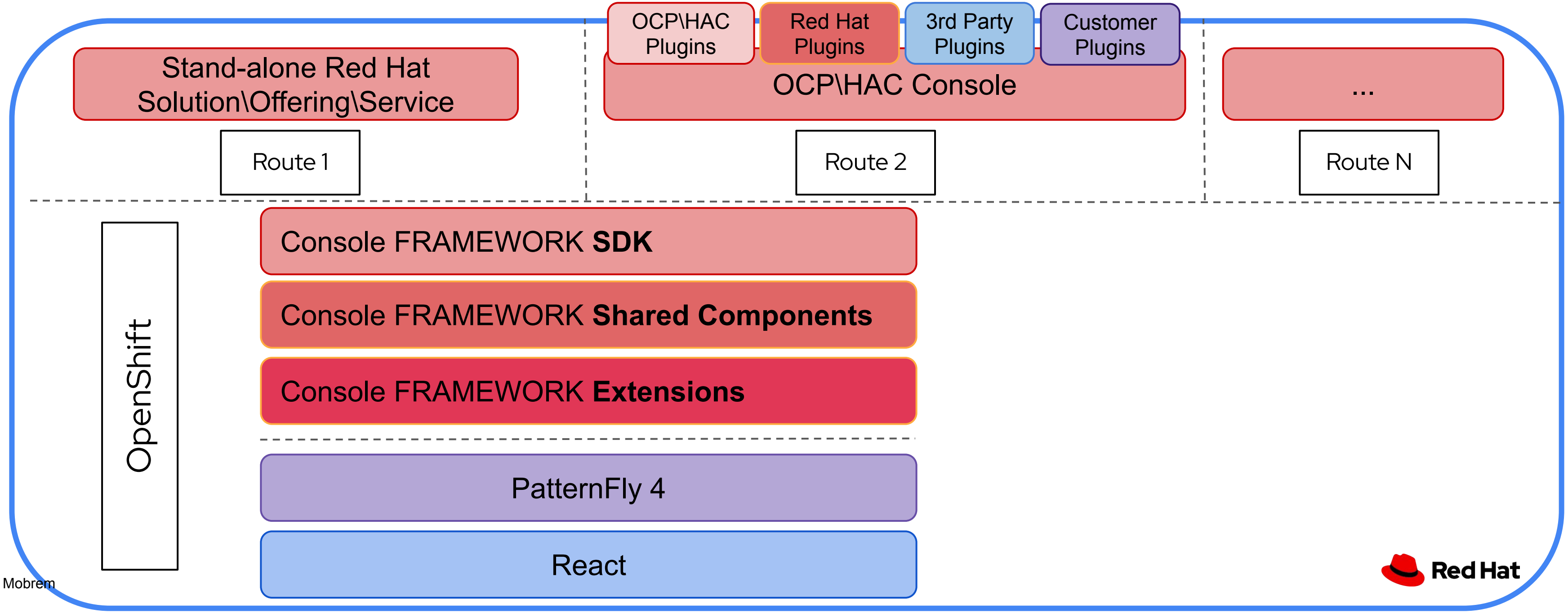
Stand Alone

Use-Case

Build Stand-alone UI on Kubernetes

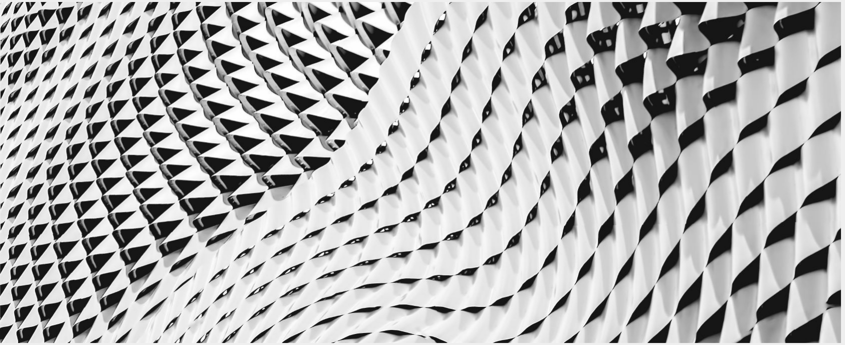
Architectural Union of Use Cases

Component breakdown

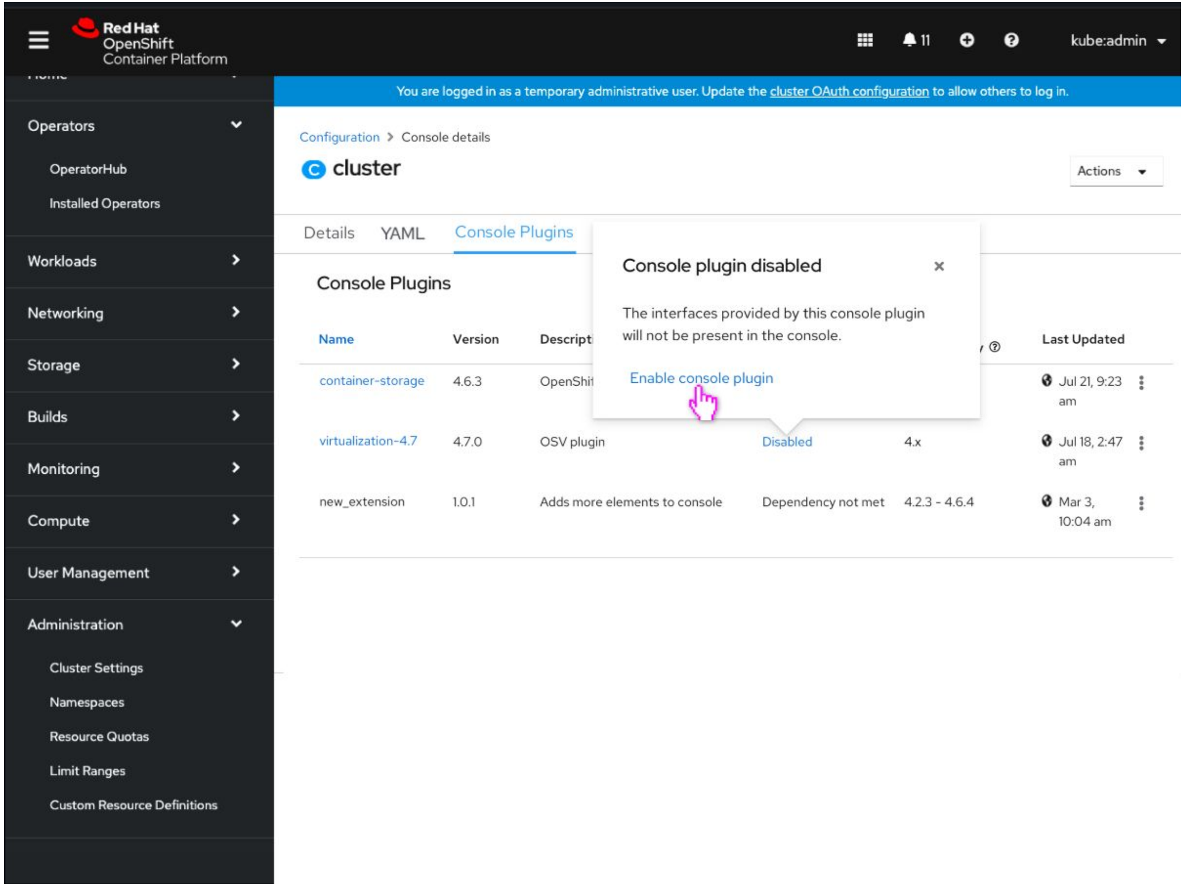

OCP\HAC Plugins
Red Hat Plugins
3rd Party Plugins
Customer Plugins
Stand-alone Red Hat Solution\Offering\Service
OCP\HAC Console
...
Route 1
Route 2
Route N
OpenShift
Console FRAMEWORK **SDK**
Console FRAMEWORK **Shared Components**
Console FRAMEWORK **Extensions**
PatternFly 4
React

PM: Ali Mobrem

How to Enable Plugins on the Console...



How You Can Enable Dynamic Plugins



You are logged in as a temporary administrative user. Update the cluster OAuth configuration to allow others to log in.

Console Plugins

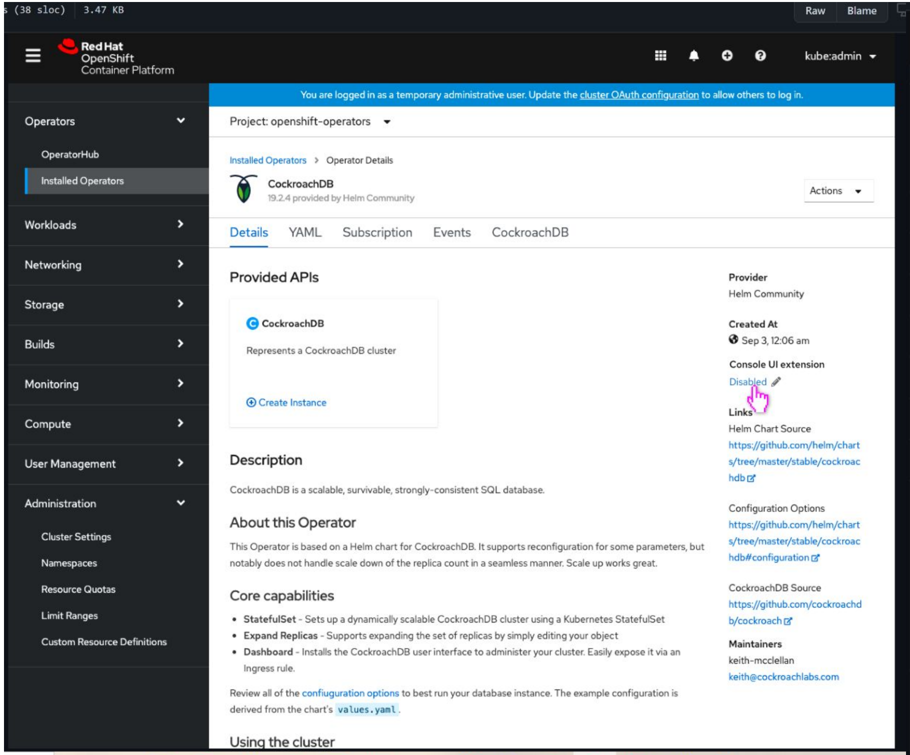
Name	Version	Descripti		ⓘ	Last Updated
container-storage	4.6.3	OpenShi			Jul 21, 9:23 am
virtualization-4.7	4.7.0	OSV plugin	Disabled	4.x	Jul 18, 2:47 am
new_extension	1.0.1	Adds more elements to console	Dependency not met	4.2.3 - 4.6.4	Mar 3, 10:04 am

Console plugin disabled ×

The interfaces provided by this console plugin will not be present in the console.

Enable console plugin

How You Can Enable Dynamic Plugins (cont...)



You are logged in as a temporary administrative user. Update the cluster OAuth configuration to allow others to log in.

Project: openshift-operators

Installed Operators > Operator Details

CockroachDB
19.2.4 provided by Helm Community

Provided APIs

CockroachDB
Represents a CockroachDB cluster

Description

CockroachDB is a scalable, survivable, strongly-consistent SQL database.

About this Operator

This Operator is based on a Helm chart for CockroachDB. It supports reconfiguration for some parameters, but notably does not handle scale down of the replica count in a seamless manner. Scale up works great.

Core capabilities

- **StatefulSet** - Sets up a dynamically scalable CockroachDB cluster using a Kubernetes StatefulSet
- **Expand Replicas** - Supports expanding the set of replicas by simply editing your object
- **Dashboard** - Installs the CockroachDB user interface to administer your cluster. Easily expose it via an Ingress rule.

Review all of the confiuguration options to best run your database instance. The example configuration is derived from the chart's `values.yaml`.

Using the cluster

Provider
Helm Community

Created At
Sep 3, 12:06 am

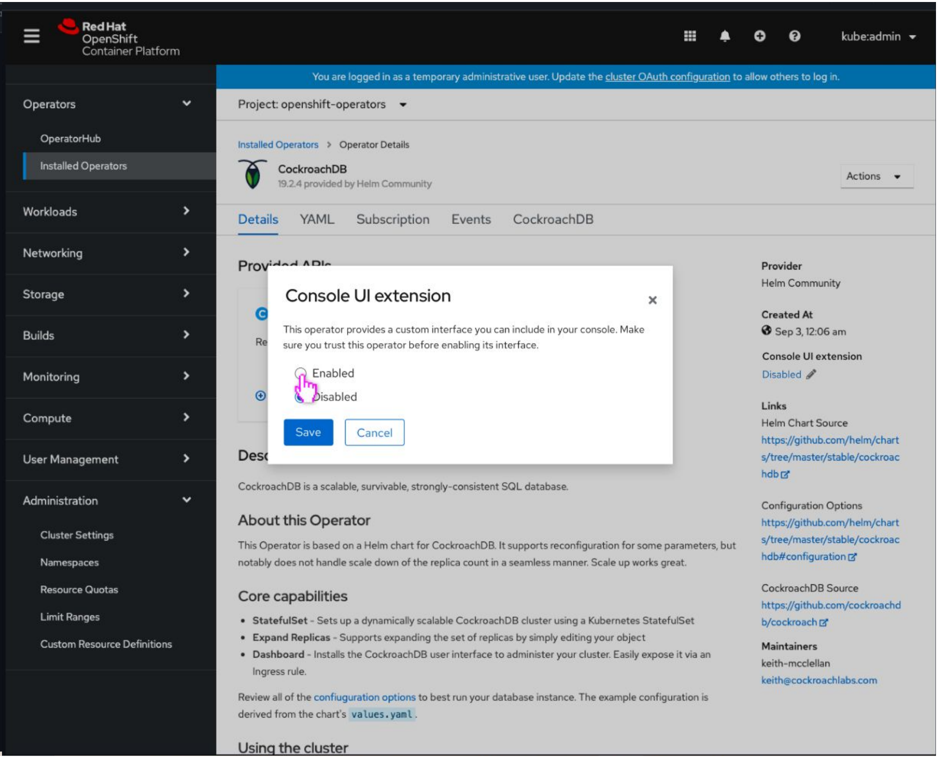
Console UI extension
Disabled

Links
Helm Chart Source
https://github.com/helm/charts/tree/master/stable/cockroachdb

Configuration Options
https://github.com/helm/charts/tree/master/stable/cockroachdb#configuration

CockroachDB Source
https://github.com/cockroachdb/cockroach

Maintainers
keith-mcclellan
keith@cockroachlabs.com



You are logged in as a temporary administrative user. Update the cluster OAuth configuration to allow others to log in.

Project: openshift-operators

Installed Operators > Operator Details

CockroachDB
19.2.4 provided by Helm Community

Console UI extension

This operator provides a custom interface you can include in your console. Make sure you trust this operator before enabling its interface.

○ Enabled
○ Disabled

Description

CockroachDB is a scalable, survivable, strongly-consistent SQL database.

About this Operator

This Operator is based on a Helm chart for CockroachDB. It supports reconfiguration for some parameters, but notably does not handle scale down of the replica count in a seamless manner. Scale up works great.

Core capabilities

- **StatefulSet** - Sets up a dynamically scalable CockroachDB cluster using a Kubernetes StatefulSet
- **Expand Replicas** - Supports expanding the set of replicas by simply editing your object
- **Dashboard** - Installs the CockroachDB user interface to administer your cluster. Easily expose it via an Ingress rule.

Review all of the confiuguration options to best run your database instance. The example configuration is derived from the chart's `values.yaml`.

Using the cluster

Provider
Helm Community

Created At
Sep 3, 12:06 am

Console UI extension
Disabled

Links
Helm Chart Source
https://github.com/helm/charts/tree/master/stable/cockroachdb

Configuration Options
https://github.com/helm/charts/tree/master/stable/cockroachdb#configuration

CockroachDB Source
https://github.com/cockroachdb/cockroach

Maintainers
keith-mcclellan
keith@cockroachlabs.com

Alternative: Enable via Cluster Settings Demo

Console Framework Road Map:

A Look at the Past, Present, Future

Recap: Dynamic Plugins-
The Road to an Extensible K8S UI

Near Term

Dynamic Plugin Framework

Remove dependency of OCP releases and enable **Operators** to deliver new UI

Mid term

Transition to Dynamic Plugins

Move new and existing internal teams to use Dynamic Plugins

Long Term

Making Dynamic Plugins Extensible

Dynamic plugins made available to public, starting with select partners then general public

Flexible & Frictionless

- Loosely couples OCP add-ons giving Operators flexibility to introduce new **UI pages**, **components** and **layouts** to the Console at **any given time**.
- Encapsulates UI code with new Operators versions
- Operators:
    - OpenShift Virtualization
    - OpenShift Serverless
    - OpenShift Pipelines
    - OpenShift Container Storage
    - Container Security Operator
    - etc..

Road to Dynamic Console Framework

Near Term

(3-6 months)

- Create foundation for SDK testing for Dynamic Plugins
- Move as many static plugins and transform them to Dynamic Plugins
- Separate plugins from mono repo to their own repos
- Resolve experimental technical gaps & unblock internal teams
- Address how to effectively share code and reuse code from the same repo exiting
- Provide enablement docs, materials and Demo plugins

Mid Term

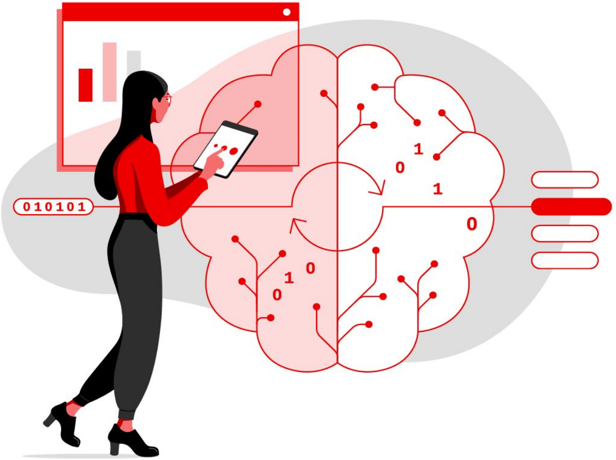
(6-9 months)

- Continue migration from Static plugins to Dynamic
- Start conversion of the Admin and Dev perspectives into dynamic plugins
- Enable managed services from cloud.redhat.com to take advantage of console Framework
- Create a multi-cluster plugin
- Dynamic plugin feature stable, mature and ready external partners

Long Term

(9+ months)

- Enable framework for 3rd Party ISV and Customer Plugins
- Console Framework positioned as the foundation of Cloud.redhat.com
- Console Framework becomes go-to framework for building custom UI's at Red Hat
- *Auto-recommended UI templates per use case based on telemetry and tracking usage data on OCP*



“You never change things by fighting the existing reality. To change something, build a new model that makes the existing model obsolete.”

Buckminster Fuller

Feedback: Be Apart of the Solution

- What are you most excited about building on OCP?

- What are your team's biggest needs with ?
  - Reqs/ Integration

Creating a new experience with the Console Framework?

- Connect with appropriate PM regarding roadmap/requirements/use cases
- Connect with appropriate UX Lead regarding design
- Work with appropriate PM & Arch to Identify extension points, verify what's needed is available, and if not get it on the backlog
- **C.RH.C use cases?**
  - PM: Ali Mobrem??, UX: Mary Clarke??, Arch: Jessica Forrester
- **Admin use cases?**
  - PM: Ali Mobrem, UX: Colleen Hart, Arch: Sam Padgett
- **Developer/DevOps use cases?**
  - PM: Serena Nichols, UX: Beau Morley, Arch: Christian Vogt

Thank you

Red Hat is the world's leading provider of enterprise open source software solutions. Award-winning support, training, and consulting services make Red Hat a trusted adviser to the Fortune 500.

linkedin.com/company/red-hat

youtube.com/user/RedHatVideos

facebook.com/redhatinc

twitter.com/RedHat

Competitors

https://clutch.sh/docs/about/comparison/

Jira - Why convo

Christian Heidenreich12:45 PM
Is there any concern around too many "siloed" experience instead of unifying and thinking about E2E journeys?

Gaurav Singh12:49 PM
we are also looking into Krew to unify install

Jessica Forrester12:50 PM
all teams contributing to the experience need to be working closely with the UXD team

Ben Parees12:50 PM
ah, jessica is here. I was gonna point to her :)

(and sam)

Christen McLemore12:51 PM
next question from Ben P

Jesus Rodriguez12:51 PM
In the past, RH would have UX guidelines for pluggable UIs. We've done this with RHCI and other places.

Michael Hrivnak12:52 PM
https://cockpit-project.org/

Jesus Rodriguez12:54 PM
The key thing is to setup UX guidelines. That will keep the UIs uniform as possible.

You12:55 PM
that's the goal Jesus :)

Peter Lauterbach12:57 PM
is there an upstream for these plugins?

Jesus Rodriguez12:57 PM
@quiana yes, this is how we've solved this problem in the past. Looking forward to this.

Ben Parees1:01 PM
is the idea here that this UI is agnostic to how the fleet of clusters are managed/created (e.g. ACM vs hypershift, hive or not, etc)?

Ben Parees1:03 PM
(we have lots of ways for a customer to end up with multiple clusters and some of those have their own/unique UX flows)

Christian Heidenreich1:05 PM
Do we already have some discussions how this will work w/ the dynamic plugin stuff if the component exposing the UX is deployed in a spoke?

Jessica Forrester1:06 PM
@christian That is one of the things architectural discussions in our list :)

Christian Heidenreich1:06 PM
fantastic! :)

What extensions are currently built on OCP (Static Plugins)

CRDs - static customization of pre-identified areas of the console

Dynamic Forms - Pre fill data-driven and responsive to user inputs

OLM Descriptors - metadata driven UI generation for OLM components

RH Native Plugins - deep integration with RH components by internal teams

competot

Different Flavors of Extensions

Static Plugins

- Code Embedded in the console
- Delivered as part of the OpenShift release
- Immediately effective upon application startup

Dynamic Plugins

- Code not tied to the console
- Delivered on the Cluster
- Operator Released

Future of Dynamic Plugs

- New products/ offerings can be generated with Dynam PLugins unlocking new doors of possibilities- Dynamic Plugin store
- Q4 New teams will be integrated for faster adoption ?