Starting Your AI journey with Openshift on Hybrid Cloud

Practical Life Cycle Tips for Intelligent Applications

Yury Titov

Sr. EMEA Black Belt for Managed Cloud Services

Bucharest, 15th of April 2025

Introduction

Yury Titov

- former senior EMEA Architect
- present: senior BlackBelt for Managed Cloud Services
- always: open source dude

Agenda

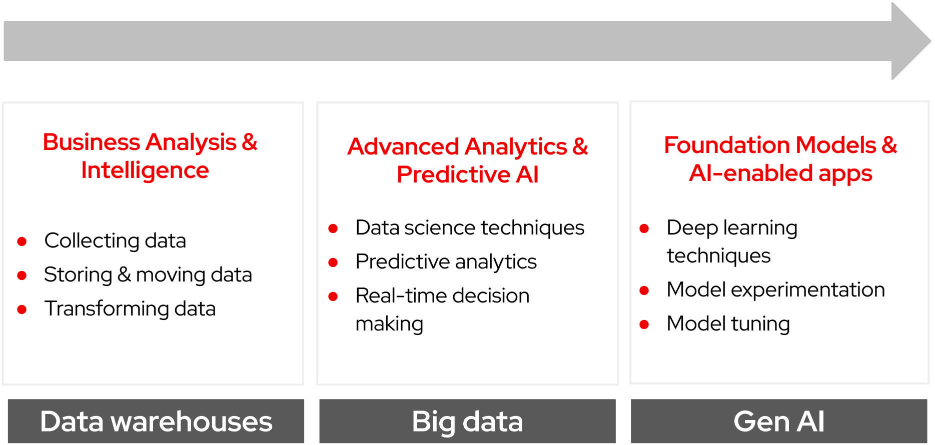
- How the AI/ML landscape is evolving: market opportunities & challenges
- AI Application Examples vs intelligent Application?
- Challenges of Operationalizing AI ?
- Team topologies and operationalizing models
- Red Hat OpenShift AI - key features and walkthrough
- Demo
- Why application platforms? Logging, Monitoring, usw. Tekton, GitOps architecture, operators, self-service
- Where to start?
- Bonus and Community: InstructLab und Neural Magic
- Conclusion and Workshop proposal

How the AI/ML landscape is evolving

AI has undergone significant evolution

The evolution of AI: from Business Intelligence to Generative AI

- Predictive AI runs businesses today
- Foundation models provide a shortcut for realizing the value of AI



Business Analysis & Intelligence

- Collecting data
- Storing & moving data
- Transforming data

Data warehouses

Advanced Analytics & Predictive AI

- Data science techniques
- Predictive analytics
- Real-time decision making

Big data

Foundation Models & AI-enabled apps

- Deep learning techniques
- Model experimentation
- Model tuning

Gen AI

Intelligent Applications?

Examples of intelligent applications

- **Recommendation engines**
  Netflix, Amazon, etc.
- **Virtual assistant**
  Siri, Alexa, etc.
- **Detecting fraudulent activity**
  Money laundering, spam, hacking, insurance
- **Quantifying risks and making smart decisions**
  Insurance, loans
- **Pattern detection**
  Images, videos: how many cars, humans, etc.
- **Analyze specialized data**
  Seismic data for oil and gas
- **Teach AI to play video games**
  AI opponents
- **Text analysis**
  Summarization, accuracy, offensive, plagiarism detection
- **Medical**
  Tumour detection
- **Customer retention**
  Predict who's about to leave

Generative AI Application Examples

- **Text Generation**
  Content creation, chatbots, etc.
- **Code Generation**
  Automate and supplement code development
- **Image Creation**
  Create new images for art, design, games, etc.
- **Music development**
  Create original music based on existing styles
- **Medical applications**
  Suggest new molecules for drug development

- **Data augmentation (synthetic data)**
  Create additional training data for model development
- **Anomaly detection**
  Detect outliers in new data
- **Content personalization**
  Personalize content like product recommendations
- **Language translation and summarization**
  Translate text or summarize long passages
- **Compliance**
  Analyze contracts or other documents for compliance

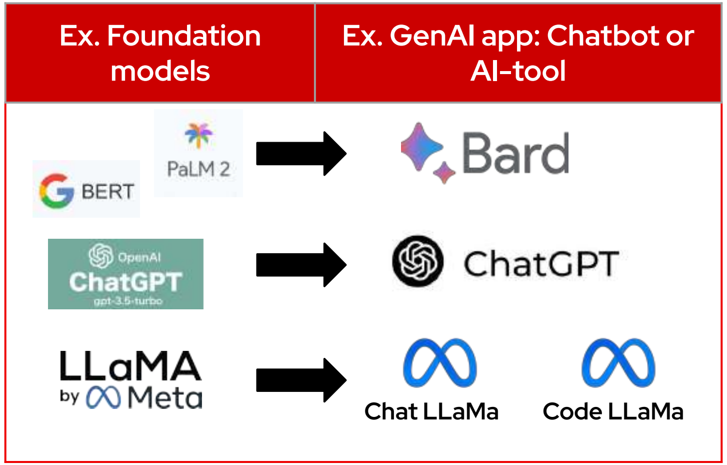
Operationalize AI with Red Hat OpenShift AI

(Generative) AI applications are powered by foundation models

Foundation models allow developing specialized AI-enabled applications

Benefits of foundation models:

- **Time to value** – alleviates the cost of compute and people
- **Accuracy** – increases with the amount of data use during training
- **Accessibility** – makes advanced AI capabilities available to non-experts
- **Versatility** – offers support for a wide range of tasks and applications



Ex. Foundation models

Ex. GenAI app: Chatbot or AI-tool

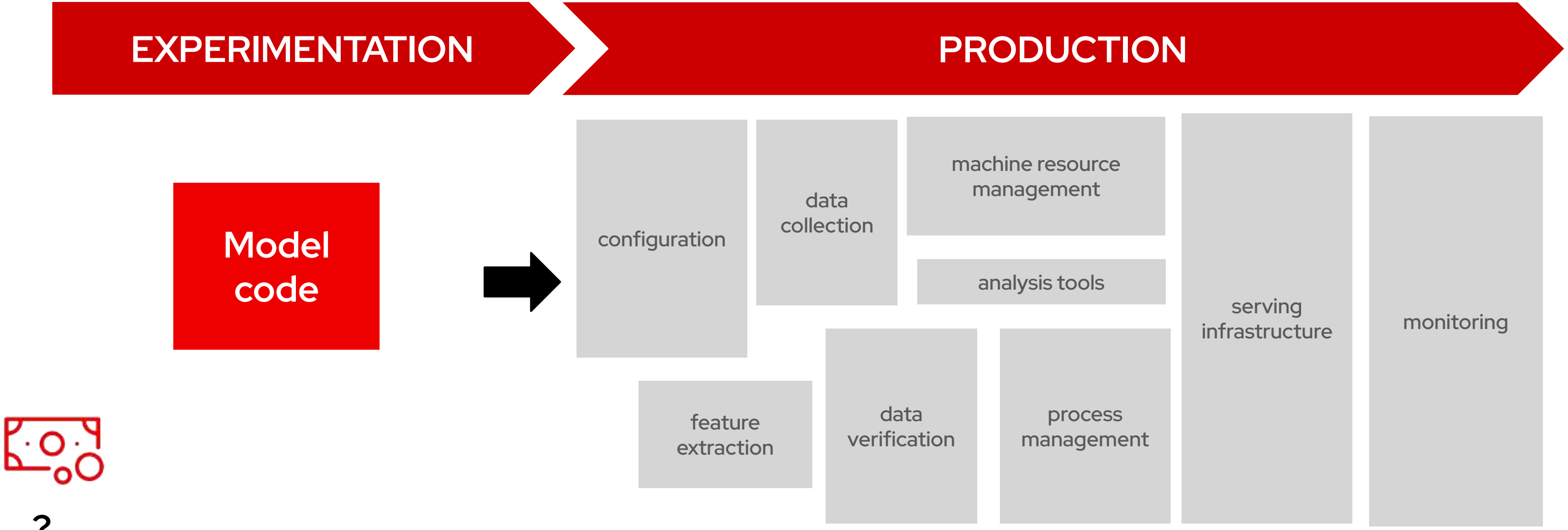
It’s not magic.
It’s math.

All of the amazing things that AI and Generative AI can do all comes down to mathematical computation.

- Compute intensive
- Storage intensive
- There are no small workloads
- Quota attainment

Poorly designed systems lead to failed ML projects

Lack of focus on end-to-end system builds technical debt



EXPERIMENTATION

PRODUCTION

Model code

configuration

data collection

machine resource management

analysis tools

serving infrastructure

monitoring

feature extraction

data verification

process management

?

Technical debt is a barrier to production

Real Life View of Technical Teams on AI*

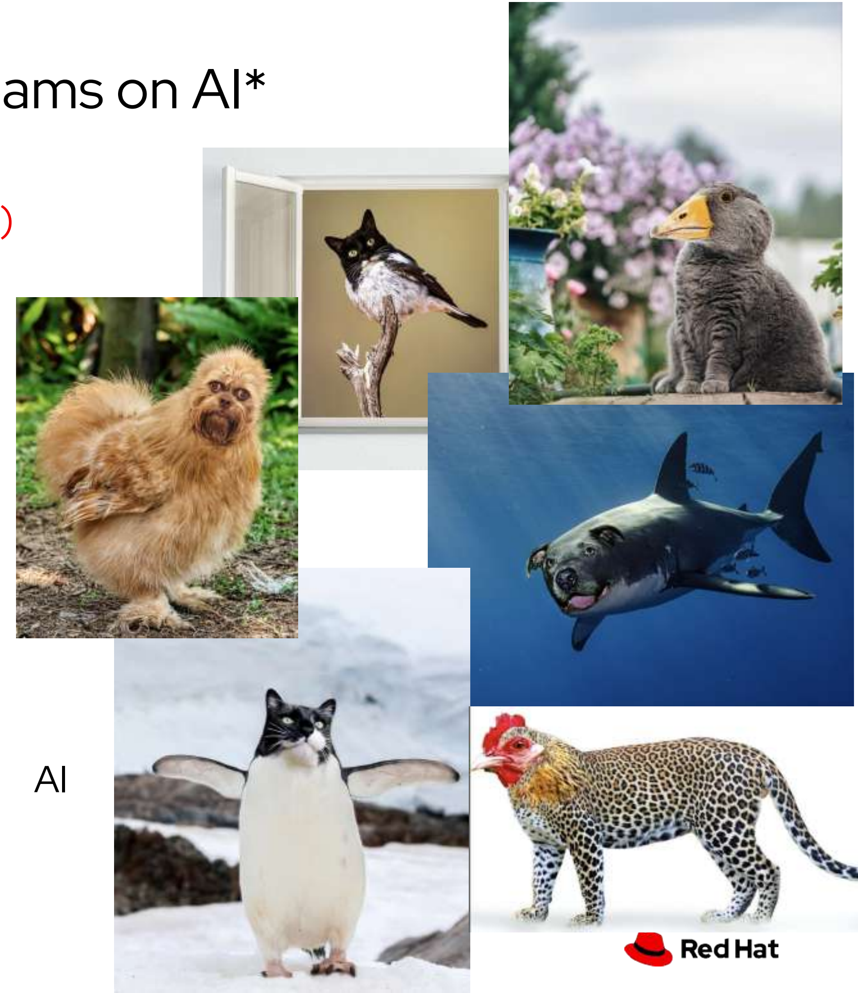
*gathered from real life experience in EMEA ;)

Legacy
Monolith

shutterstock.com · 2293790255

Modern
Microservices

AI



Operationalizing AI/ML requires collaboration

Every member of your team plays a critical role in a complex process

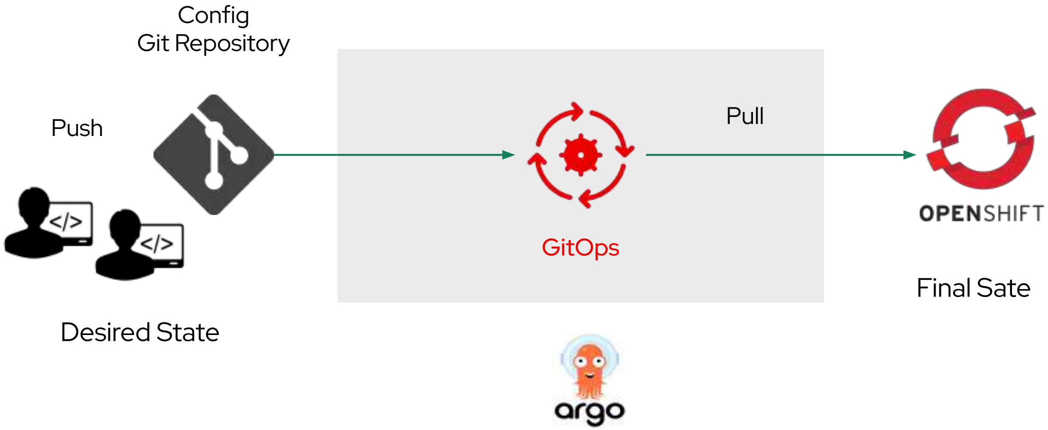
	Set goals	Gather and prepare data	Develop model	Integrate models in app dev	Model monitoring and management
Business leadership	━				
Data engineer		━			
Data scientist			━		━
AI engineer			━	━	━
App developer				━	━
IT operations/Platform Engineering		━	━	━	━

https://cloud.google.com/architecture/mlops-continuous-delivery-and-automation-pipelines-in-machine-learning

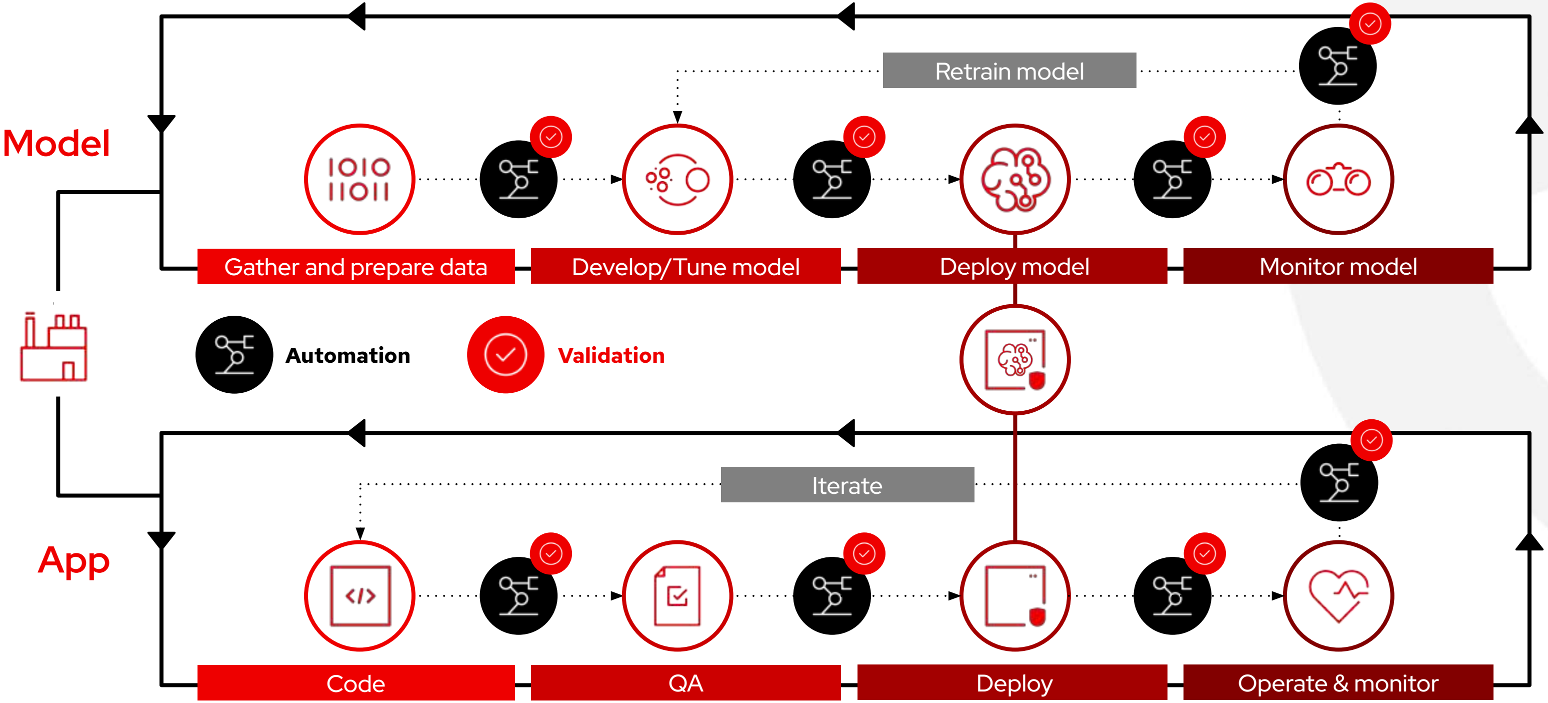
ArgoCD

Git-/MLOps: Different clusters for each stage of Application Lifecycle

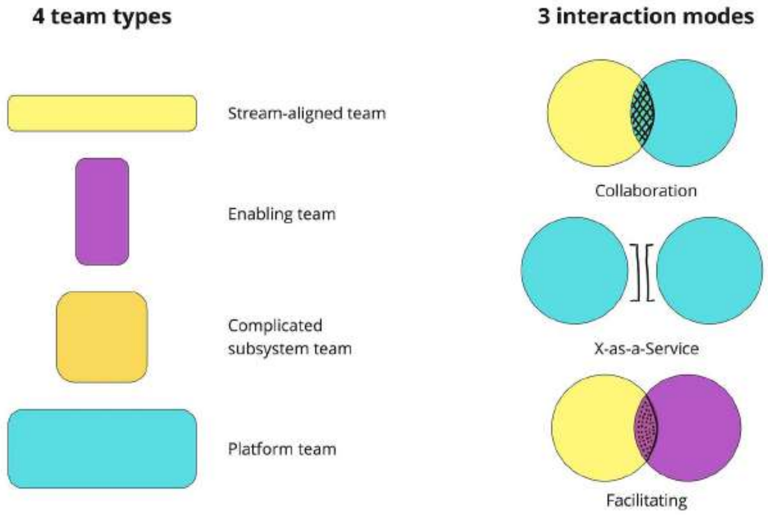
Manage AI Apps and Infra



Lifecycle for operationalizing (containerised) models



Teams



1. **Stream-aligned teams**

   aligned to a single delivery stream, such as a product or service (what others might call a “product team” or a “feature team”).

2. **Enabling teams**

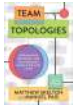
   specialists in a particular domain that guide stream-aligned teams

3. **Complicated-subsystem teams**

   maintain a particularly complex subsystem, such as an ML model

4. **Platform teams**

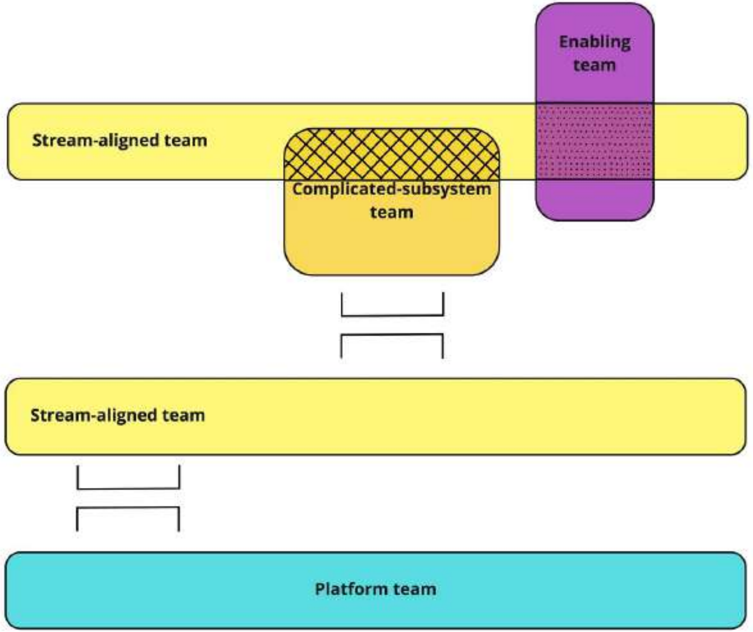
   provide internal services like deployment platforms or data services

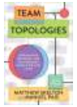


Team Topologies: Organizing Business and Technology Teams for Fast Flow, Pias & Skelton
ISBN: 9781942788812

Red Hat’s approached are informed by, and align with, Team Topologies

Typical interactions between different teams



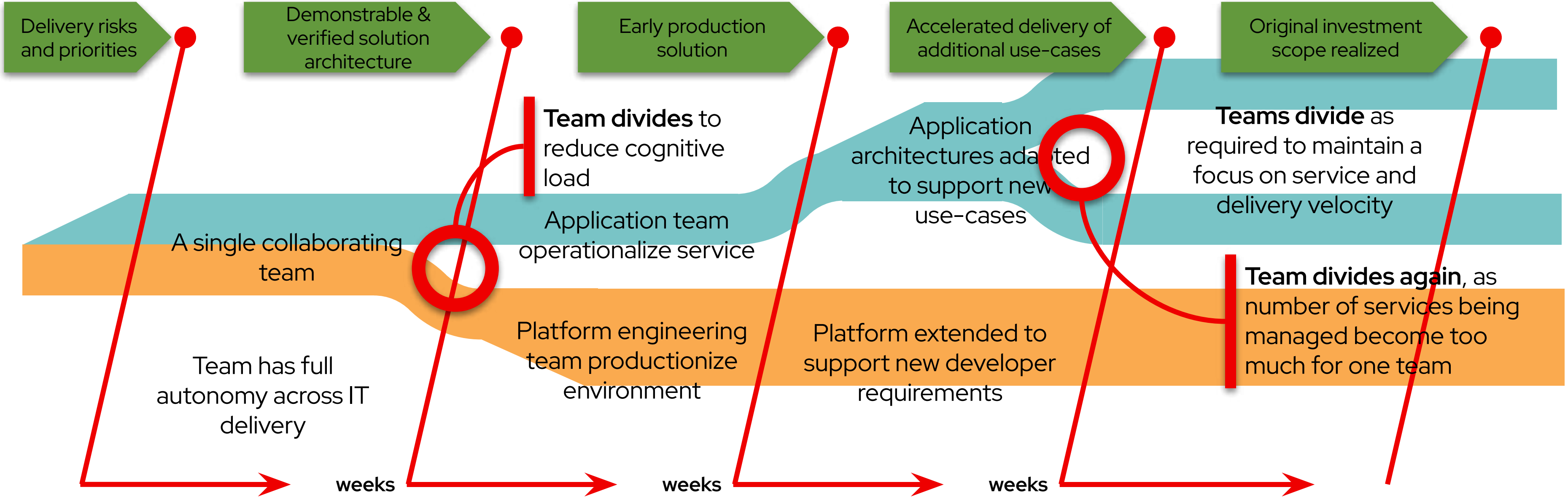


Team Topologies: Organizing Business and Technology Teams for Fast Flow, Pias & Skelton
ISBN: 9781942788812

Red Hat's approached are informed by, and align with, Team Topologies

Red Hat recommends an evolutionary approach to organisational change

Organisational change is seeded through delivery of specific services, and designed to scale as required



Delivery risks and priorities

Demonstrable & verified solution architecture

Early production solution

Accelerated delivery of additional use-cases

Original investment scope realized

A single collaborating team

Team has full autonomy across IT delivery

Team divides to reduce cognitive load

Application team operationalize service

Platform engineering team productionize environment

Application architectures adapted to support new use-cases

Platform extended to support new developer requirements

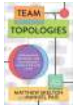
Teams divide as required to maintain a focus on service and delivery velocity

Team divides again, as number of services being managed become too much for one team

weeks

weeks

weeks



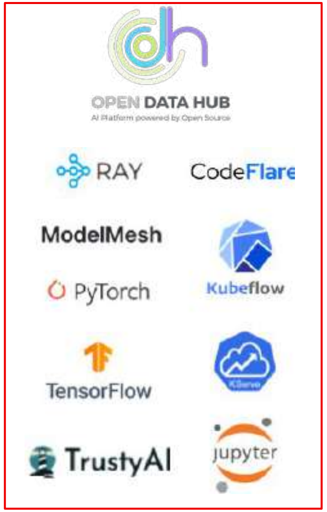
Team Topologies: Organizing Business and Technology Teams for Fast Flow, Pias & Skelton

ISBN: 9781942788812

Red Hat's approached are informed by, and align with, Team Topologies

Simplify AI adoption

Designed to increase AI adoption and enhance trust in AI initiatives



Red Hat
OpenShift AI

Flexible

A composable platform for rapid dev and delivery of AI-enabled apps

Expand

A certified AI partner ecosystem for delivering an E2E AI/ML experience

Red Hat AI - Key features

Model development

Interactive, collaborative UI for exploratory data science, and model training, tuning and serving

Model serving

Model serving routing for deploying models to production environments

Model monitoring

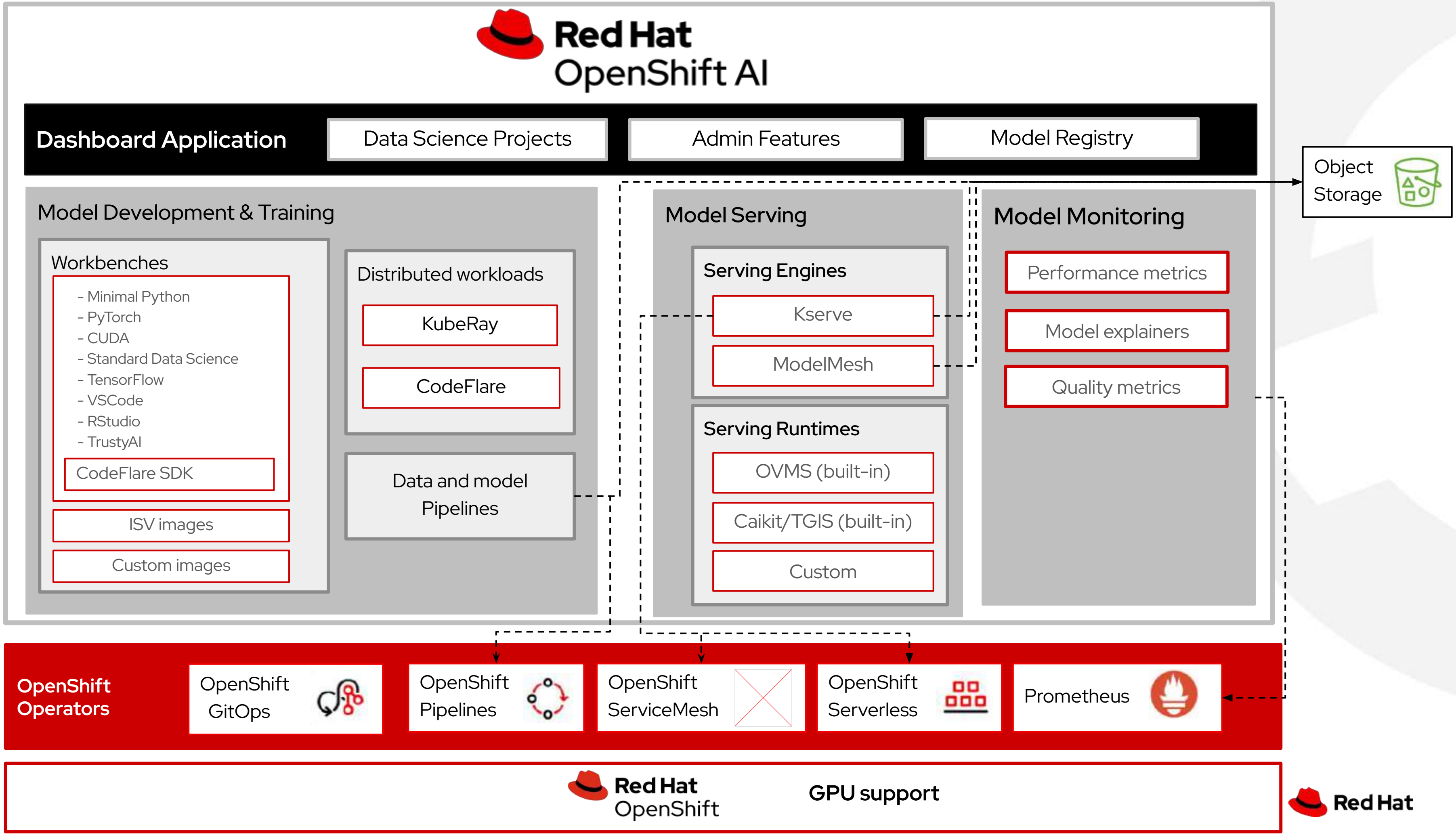
Centralized monitoring for tracking models performance and accuracy

Data & model pipelines

Visual editor for creating and automating data science pipelines

Distributed workloads

Seamless experience for efficient data processing, model training, tuning and serving



Red Hat OpenShift AI

Dashboard Application

- Data Science Projects
- Admin Features
- Model Registry

Object Storage

Model Development & Training

Workbenches

- Minimal Python
- PyTorch
- CUDA
- Standard Data Science
- TensorFlow
- VSCode
- RStudio
- TrustyAI

CodeFlare SDK

ISV images

Custom images

Distributed workloads

- KubeRay
- CodeFlare

Data and model Pipelines

Model Serving

Serving Engines

- Kserve
- ModelMesh

Serving Runtimes

- OVMS (built-in)
- Caikit/TGIS (built-in)
- Custom

Model Monitoring

- Performance metrics
- Model explainers
- Quality metrics

OpenShift Operators

- OpenShift GitOps
- OpenShift Pipelines
- OpenShift ServiceMesh
- OpenShift Serverless
- Prometheus

Red Hat OpenShift

GPU support

Build an AI platform for E2E AI lifecycle management

Red Hat
OpenShift

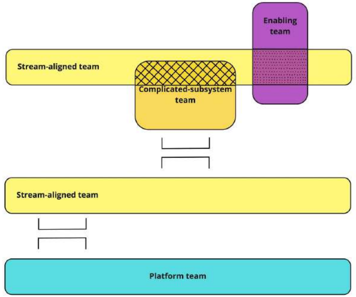
Trusted, comprehensive and consistent hybrid application platform for managing the entire application lifecvcle

Red Hat
OpenShift AI

Open hybrid AI/ML platform, built on top of OpenShift, to create and deliver AI-enabled apps securely at scale across hybrid-clouds

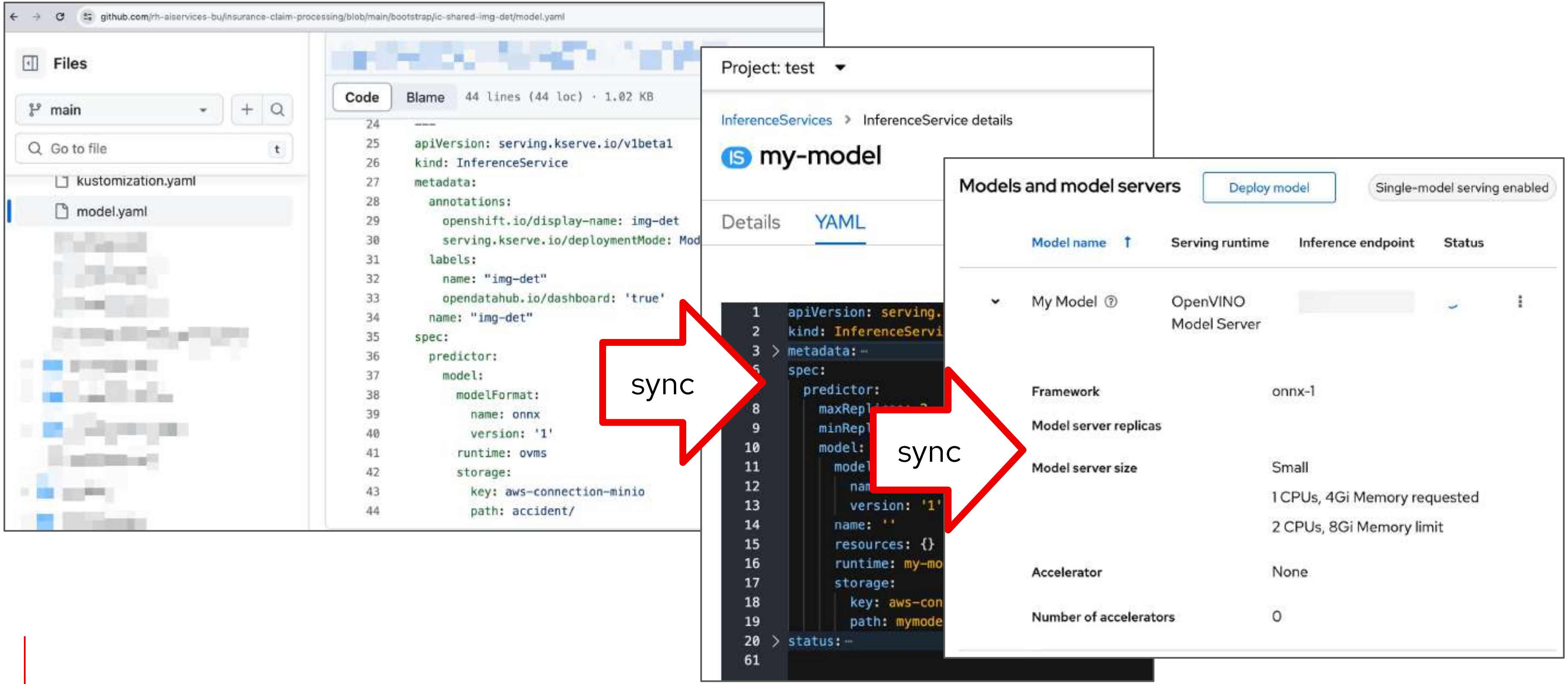
Red Hat's
AI Partner Ecosystem

Best-of-breed AI technologies from a certified partner ecosystem to complement or extend Red Hat's AI capabilities



UI to Yaml

GitOps (MLOps): Everything in RHAI has a YAML representation


github.com/rh-aiservices-bu/insurance-claim-processing/blob/main/bootstrap/ic-shared-img-det/model.yaml
Files
main
Go to file
t
kustomization.yaml
model.yaml
Code
Blame
44 lines (44 loc) · 1.02 KB
sync
sync
Project: test
InferenceServices > InferenceService details
my-model
Details
YAML
Models and model servers
Deploy model
Single-model serving enabled

```yaml
---
apiVersion: serving.kserve.io/v1beta1
kind: InferenceService
metadata:
  annotations:
    openshift.io/display-name: img-det
    serving.kserve.io/deploymentMode: Mod
  labels:
    name: "img-det"
    opendatahub.io/dashboard: 'true'
  name: "img-det"
spec:
  predictor:
    model:
      modelFormat:
        name: onnx
        version: '1'
      runtime: ovms
      storage:
        key: aws-connection-minio
        path: accident/
```

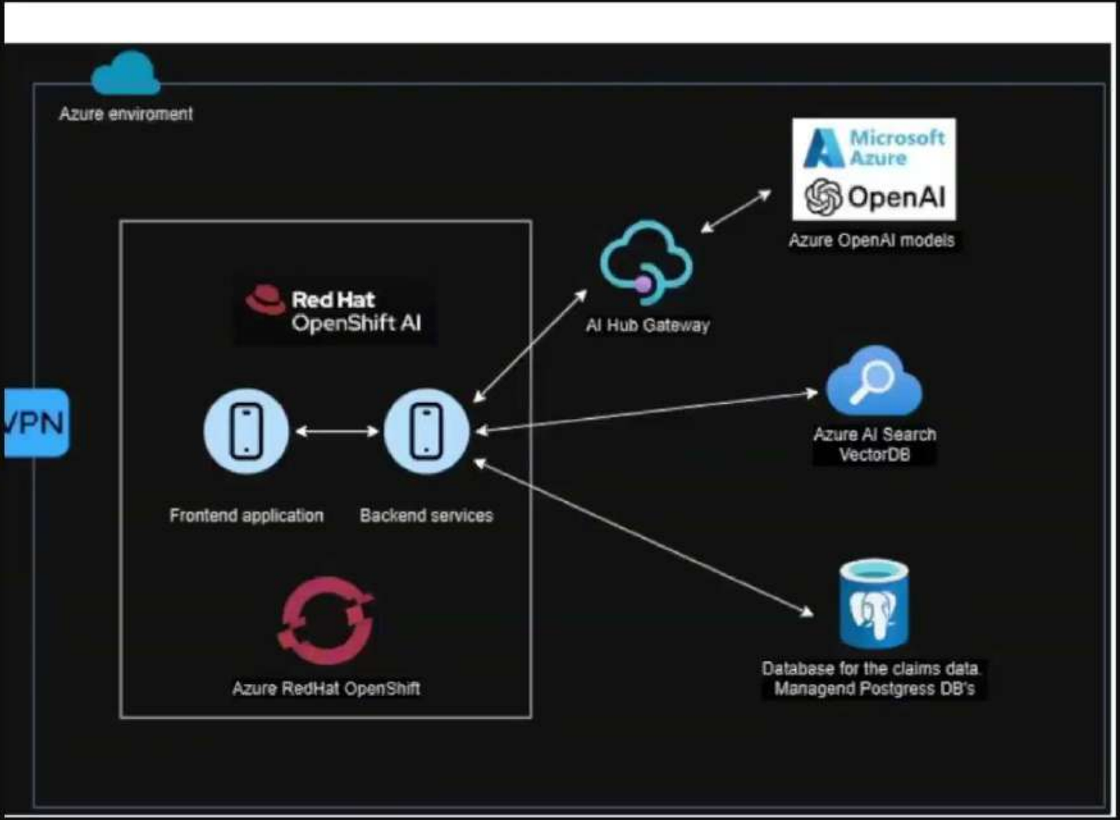
```yaml
apiVersion: serving.
kind: InferenceServi
metadata: …
spec:
  predictor:
    maxRepl
    minRepl
    model:
      model
        nam
        version: '1'
      name: ''
      resources: {}
      runtime: my-mo
      storage:
        key: aws-con
        path: mymode
status: …
```

Model name	Serving runtime	Inference endpoint	Status
My Model	OpenVINO Model Server		

Framework	onnx-1
Model server replicas	
Model server size	Small 1 CPUs, 4Gi Memory requested 2 CPUs, 8Gi Memory limit
Accelerator	None
Number of accelerators	0

What is Red Hat OpenShift AI (RHOAI) solving

- **MLOps**
  - RHOAI helps you build out an enterprise grade AI and MLOps platform to create and deliver GenAI and predictive models by providing supported AI tooling on top of OpenShift.
  - It's based on OpenShift, a container based application platform that efficiently scales to handle workload demands of AI operations and models.
  - You can run your AI workloads across the hybrid cloud, including edge and disconnected environments.
- **Unified app platform**
  - OpenShift supports the end-to-end application lifecycle. RHOAI extends OpenShift to AI models, getting them into to AI models and getting them into production with OpenShift best practices.
  - Seamless collaboration across multiple personas including IT Ops, Data scientists and application developers by providing a unified platform.
- **Extensibility**
  - RHOAI is built to be modular, allowing for a customizable AI/ML stack where you can plug in partners or open source software and technologies where needed to build out an MLOps platform that fits your organization.
- **No vendor lock-in**
  - Thanks to being modular and able to run across the hybrid cloud, you have the freedom to migrate and extend as needed, allowing you to keep up with the speed of AI innovation.


Azure enviroment
Microsoft Azure
OpenAI
Azure OpenAI models
Red Hat OpenShift AI
AI Hub Gateway
VPN
Frontend application
Backend services
Azure AI Search VectorDB
Azure RedHat OpenShift
Database for the claims data.
Managend Postgress DB's

A consistent platform no matter how or where you run

Red Hat OpenShift cloud services—Fully managed, start quickly

aws

Red Hat OpenShift Service on AWS

Microsoft Azure

Azure Red Hat OpenShift

IBM Cloud

Red Hat OpenShift on IBM Cloud

Google Cloud

Red Hat OpenShift Dedicated

Self-Managed Red Hat OpenShift—Customer managed, for control and flexibility

Red Hat OpenShift

Hybrid cloud: on **public cloud**, on-premises on **physical** or **virtual** infrastructure, and at the **edge**

Build and run a platform **OR** using Azure Red Hat OpenShift (ARO)

Use your tool of choice with integrated Azure components.

Azure Monitor

Azure Firewall

Azure Resource Manager

Azure Arc

Azure Log Analytics

Azure AD

And more.

The Parts

Monitoring | Service Mesh | Dev Tools

Registry | Metrics | Logging | CI/CD

Kubernetes Cluster Services
Basic Networking :: Ingress

Kubernetes

Custom OS

DIY/xKS

The Assembled Car

Monitoring | Service Mesh | Dev Tools

Registry | Metrics | Logging | CI/CD

OpenShift Cluster Services
Networking :: Router :: OLM

Kubernetes

Red Hat Core OS

- Application Platform -
Self-managed Red Hat OpenShift

The Car & Pit Crew

SRE and Customer Success

Monitoring | Service Mesh | Dev Tools

Registry | Metrics | Logging | CI/CD

OpenShift Cluster Services
Networking :: Router :: OLM

Kubernetes

Red Hat Core OS

Microsoft Azure

- Turnkey Application Platform -
Azure Red Hat OpenShift (ARO)

Red Hat

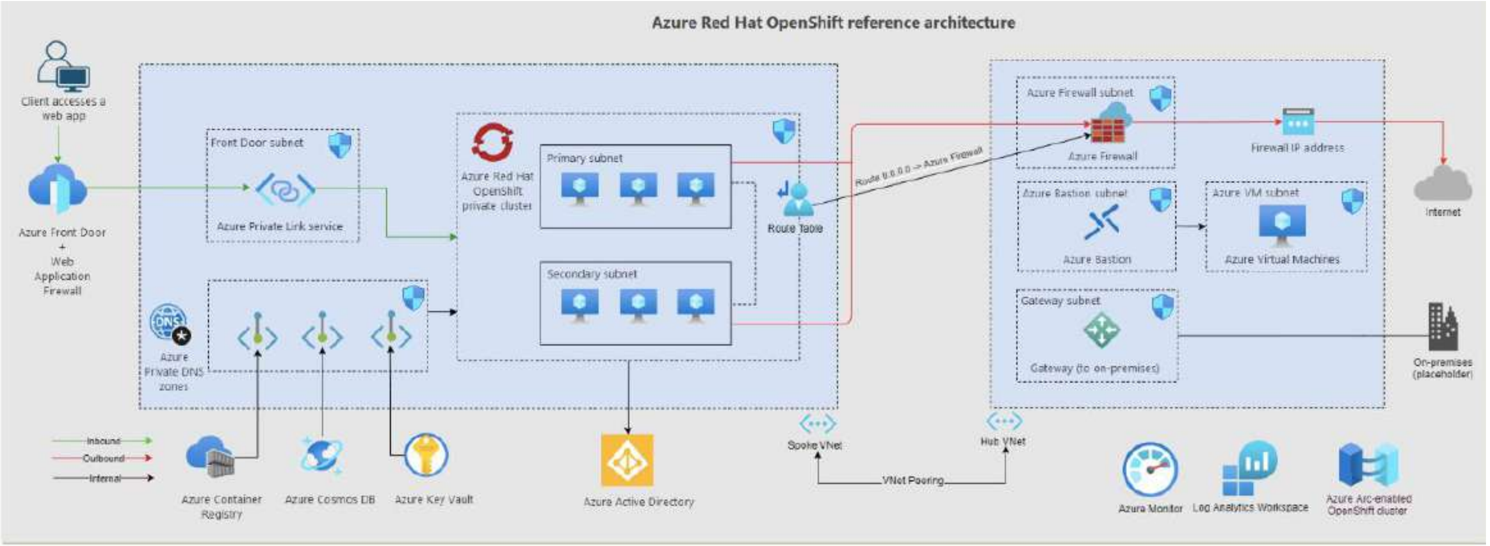
Azure Red Hat OpenShift integrates with OpenShift and Azure Developer and Management Tools

- OpenShift developer console
- Code Ready Workspaces
- OpenShift Operators
- OpenShift AI
- OpenShift Pipelines
- Red Hat Runtimes
- OpenShift API Management
- OpenShift Serverless
- OpenShift Service Mesh
- OpenShift developer sandbox
- OpenShift GitOps

- Log analytics workspace
- Azure Arc-enabled OpensShift cluster
- Azure Resource Manager
- Azure Arc
- Azure AD
- Azure Monitor
- Azure Firewall
- Azure OpenAI
- Azure Visual Studio
- Azure Load Testing
- Azure Log Analytics

Additional tools: Azure Developer Tools and Management products

Accelerate your deployments with guidance from the ARO landing zone accelerator



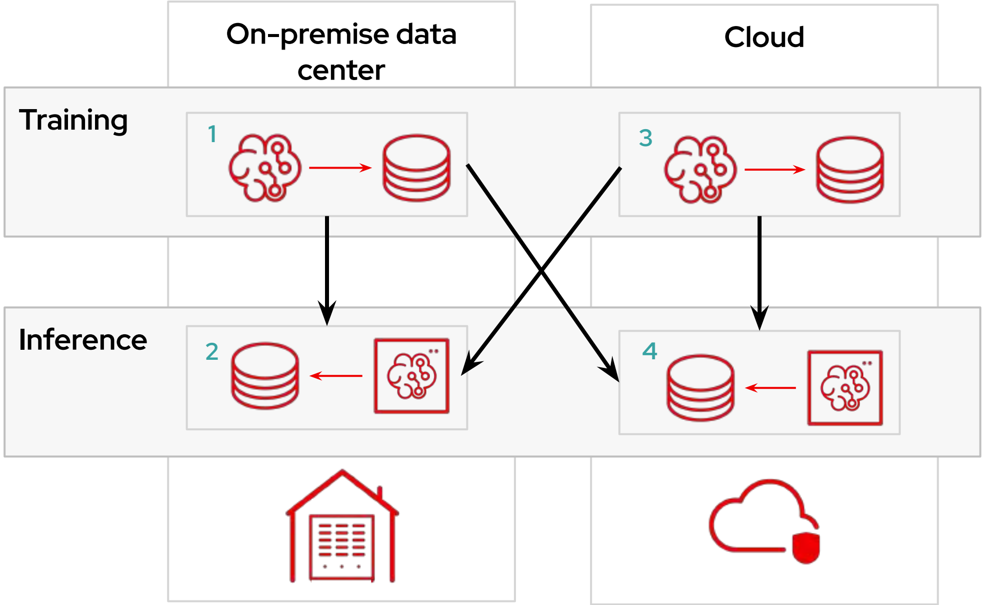
https://learn.microsoft.com/en-us/azure/cloud-adoption-framework/scenarios/app-platform/azure-red-hat-openshift/landing-zone-accelerator

Gain hybrid cloud flexibility

Train and deploy models and AI-enabled apps on-premises, cloud or edge

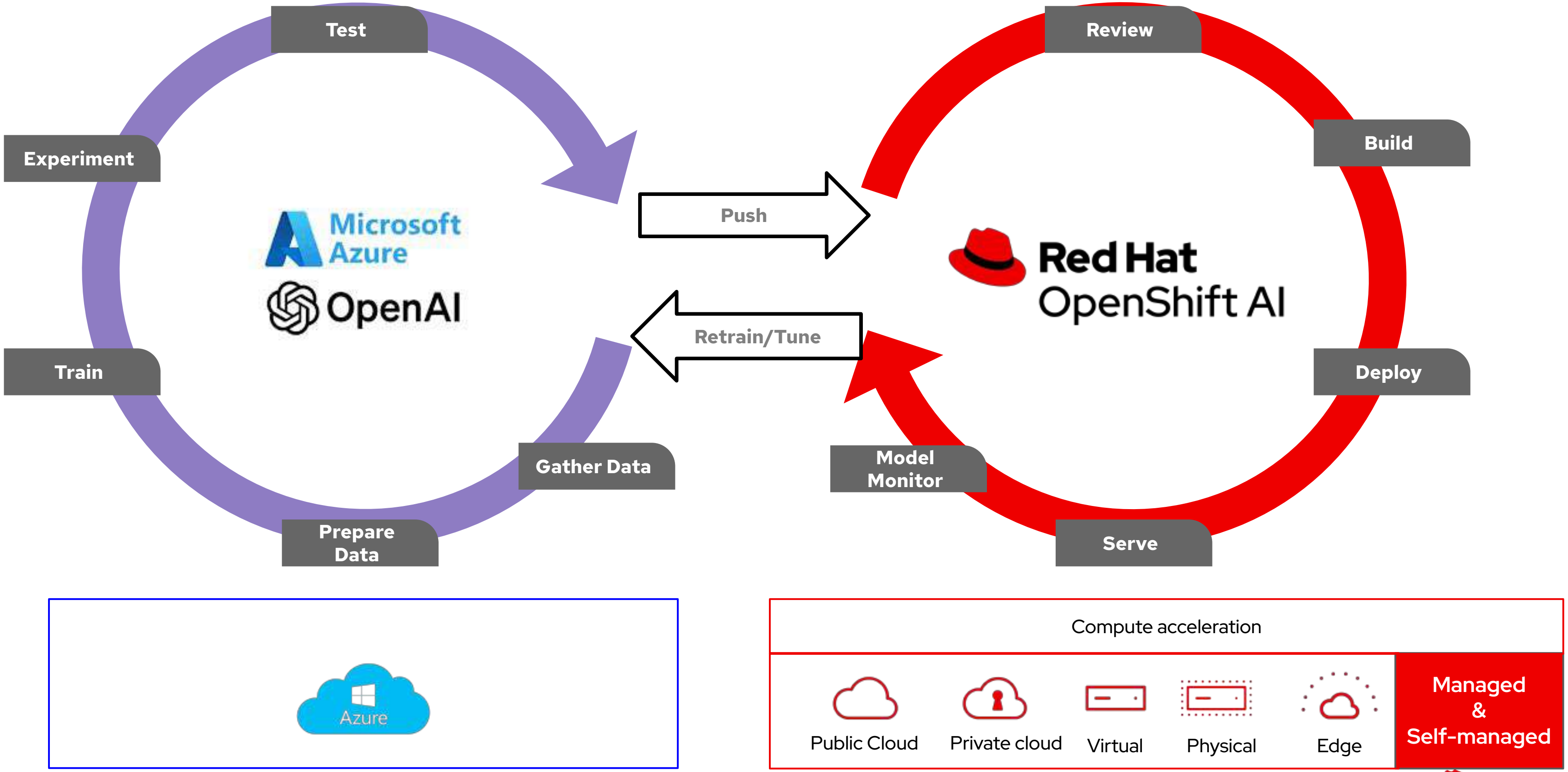
What you do should not dictate **where** you do it

1. Data on-prem = Train on-prem
2. Data on-prem = Inference on-prem
3. Data in the cloud = Train on cloud
4. Data in the cloud = Inference on cloud



Inner loop (dev)

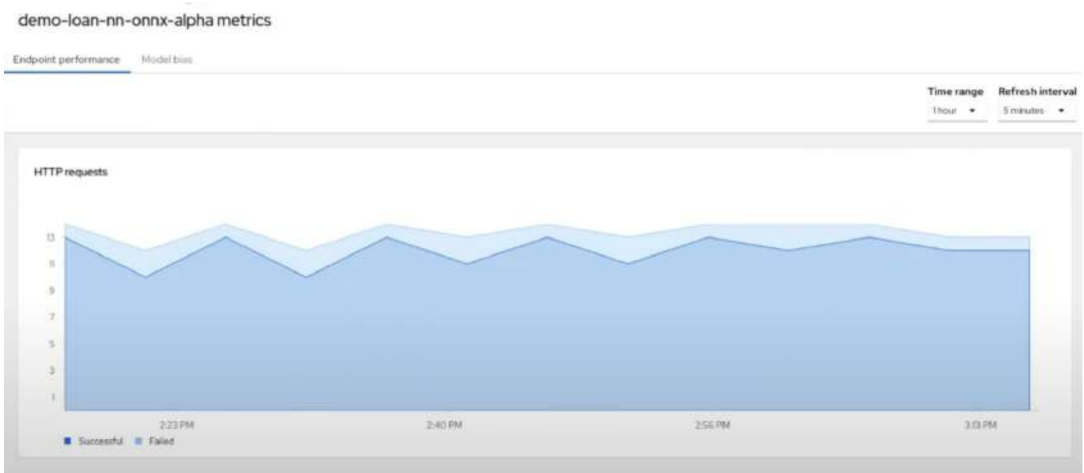
Outer loop (prod)



Monitoring

Monitoring in RHAI

You can get to monitoring by clicking on a served model, either in Data Science Project or in the Model Serving page.



demo-loan-nn-onnx-alpha metrics

Time range: 1 hour

Refresh interval: 5 minutes

Monitoring Model Performance

In Red Hat AI, you can monitor the following metrics for all the models that are deployed on a model server:

- **HTTP requests**
  - The number of HTTP requests that have failed or succeeded for all models on the server.
- **Average response time (ms)**
  - For all models on the server, the average time it takes the model server to respond to requests.
- **CPU utilization (%)**
  - The percentage of the CPU's capacity that is currently being used by all models on the server.
- **Memory utilization (%)**
  - The percentage of the system's memory that is currently being used by all models on the server.

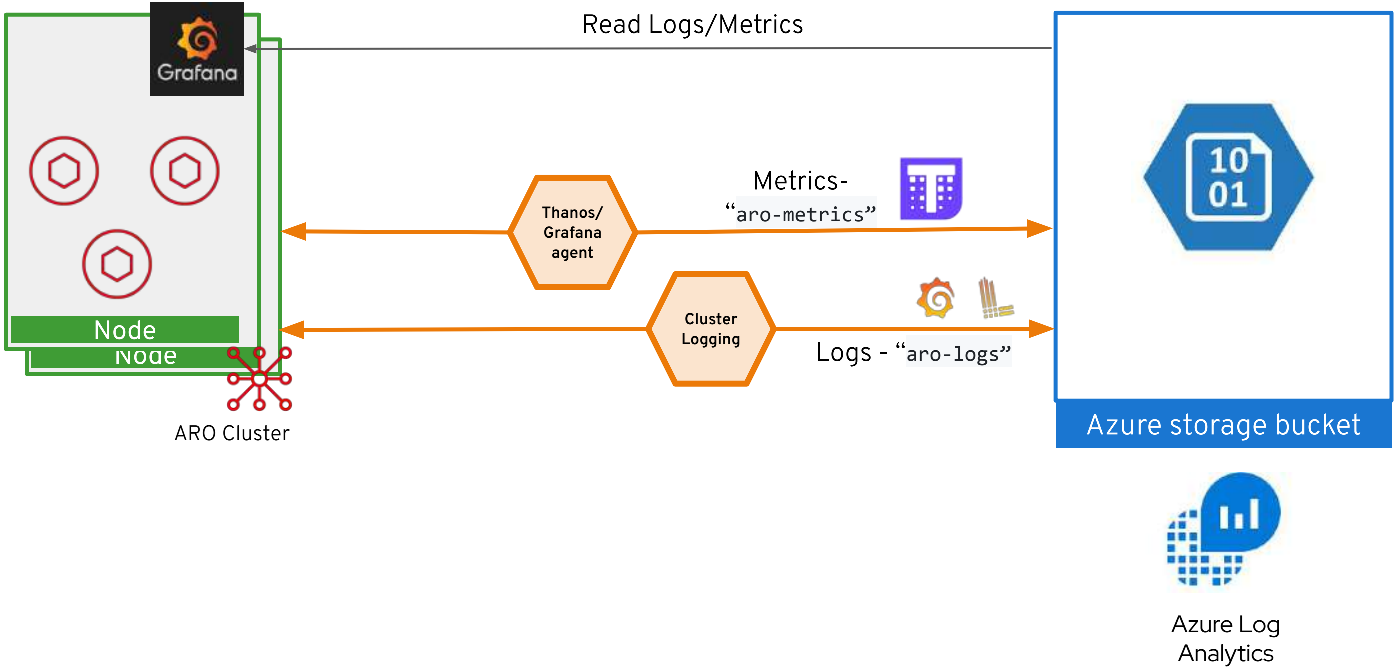
Logging

Forwarding Metrics and Logs to Azure Files

Shipping logs to an enterprise-wide log management system.

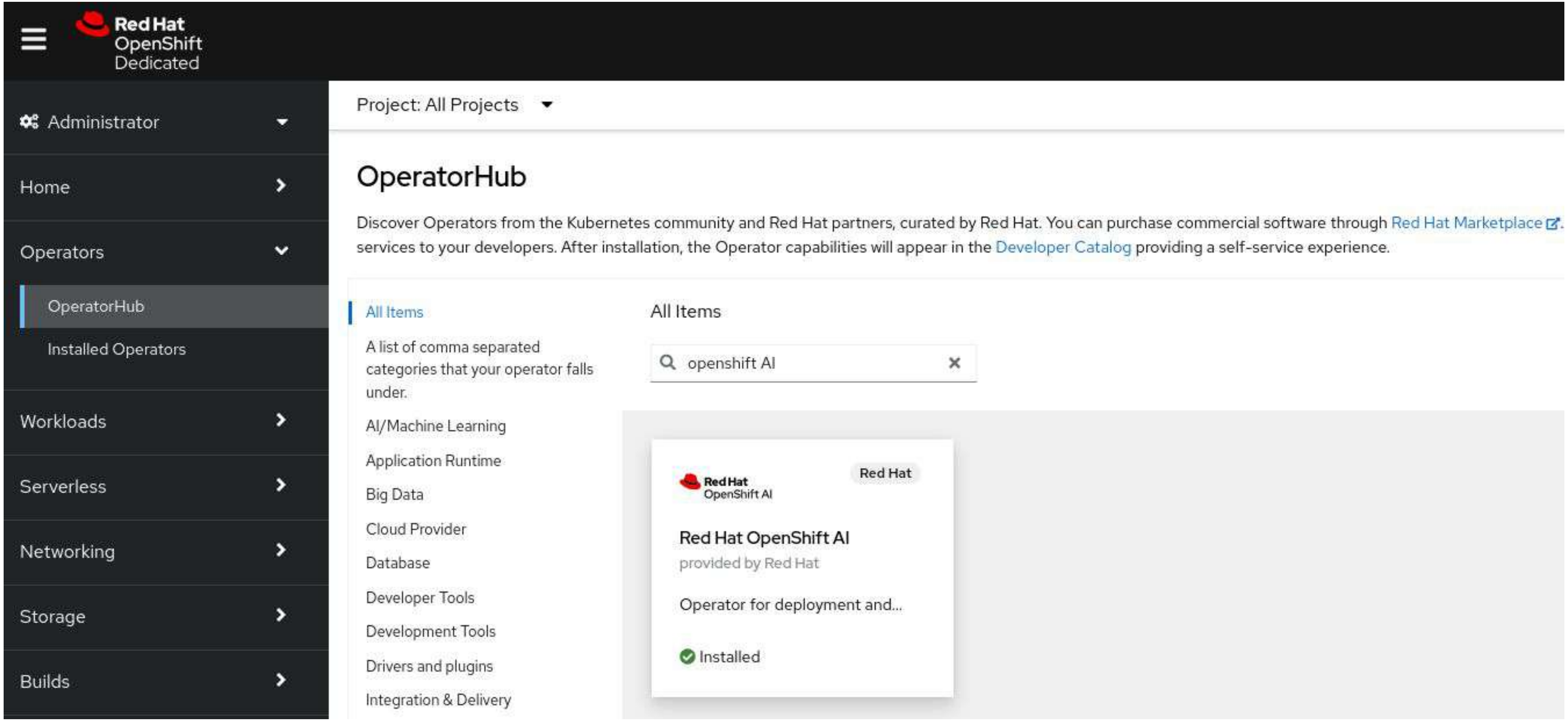
- **OpenShift Cluster logs** are **stored in cluster** by **default**.
- **Cluster logs** can be **shipped** to a variety of log management systems such as **FluentD, ElasticSearch, Syslog, Loki, Kafka, and Splunk.**
- Shipping logs to Azure Files allows them to be viewed in Grafana and other visualization tools.

Observability Forwarding to Azure Files



Where do we start?

Install RHOAI



OperatorHub

Discover Operators from the Kubernetes community and Red Hat partners, curated by Red Hat. You can purchase commercial software through Red Hat Marketplace. services to your developers. After installation, the Operator capabilities will appear in the Developer Catalog providing a self-service experience.

openshift AI

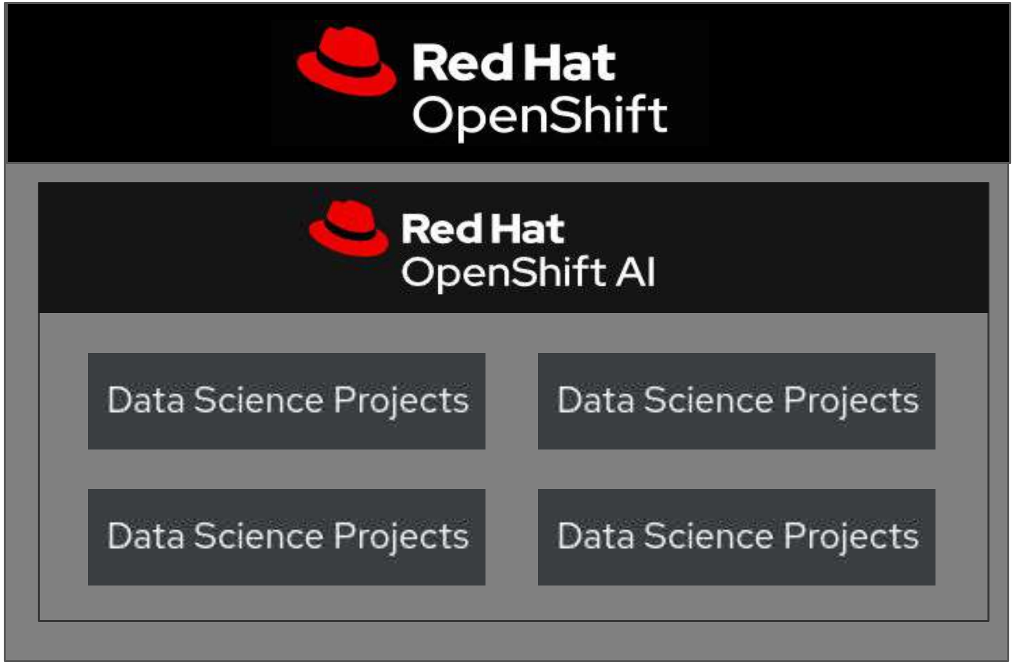
Red Hat OpenShift AI
Red Hat

Red Hat OpenShift AI
provided by Red Hat

Operator for deployment and...

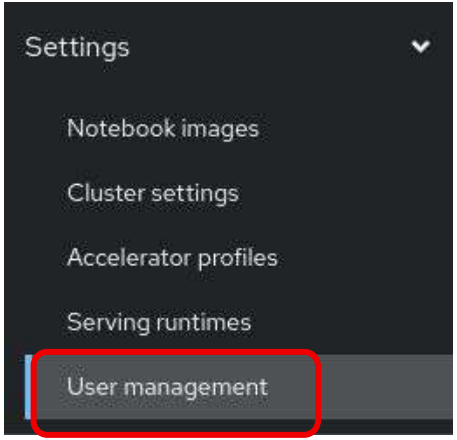
✓ Installed

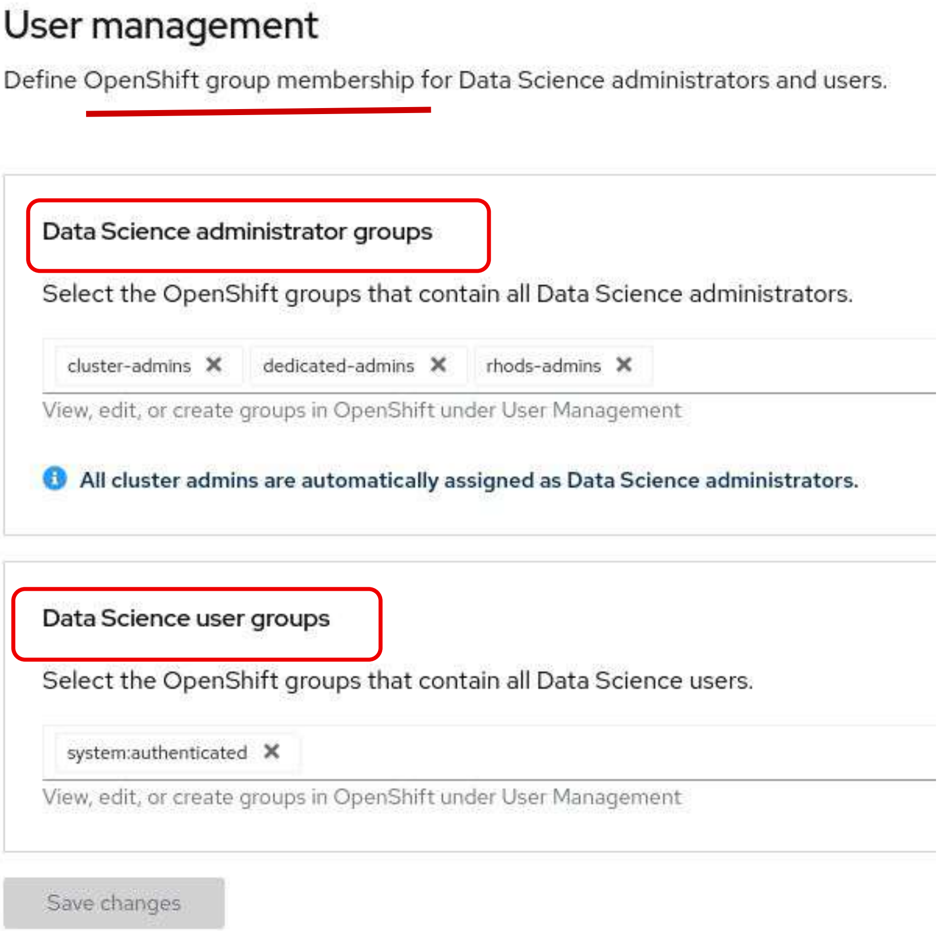
Data Science Projects



- Multiple data science projects.
- Isolation from other projects
- Created by admins or users
- User/Group access privileges

User Management





User management

Define OpenShift group membership for Data Science administrators and users.

Data Science administrator groups

Select the OpenShift groups that contain all Data Science administrators.

cluster-admins ✕ dedicated-admins ✕ rhods-admins ✕

View, edit, or create groups in OpenShift under User Management

ℹ All cluster admins are automatically assigned as Data Science administrators.

Data Science user groups

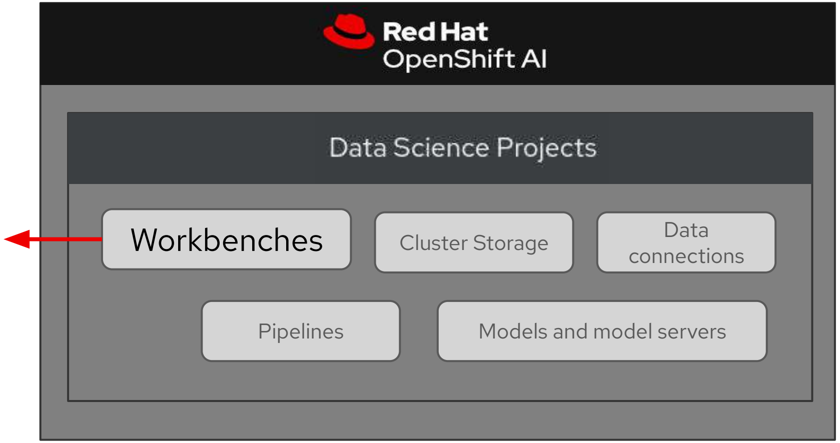
Select the OpenShift groups that contain all Data Science users.

system:authenticated ✕

View, edit, or create groups in OpenShift under User Management

Workbenches

- **Notebook Image**
  - **Development environment** in the form of a container image
    - combination of IDE like Jupyter Notebook, VSCode, etc., and choice of AI/ML framework like Tensorflow, PyTorch etc.,
  - **Custom notebook images**.
- **Deployment size**
  - Container size → **# CPUs** & **Memory** size
  - Accelerator → Choice of **Accelerators**/**GPUs**
- **Environment variables**
  - Config Map
  - Secret
- **Cluster Storage**
  - **PVC** connected to the development environment to store code & related artifacts.
- **Data connections**
  - **Object store** for hosting models as well as storing pipeline artifacts.



Red Hat
OpenShift AI

Data Science Projects

Workbenches

Cluster Storage

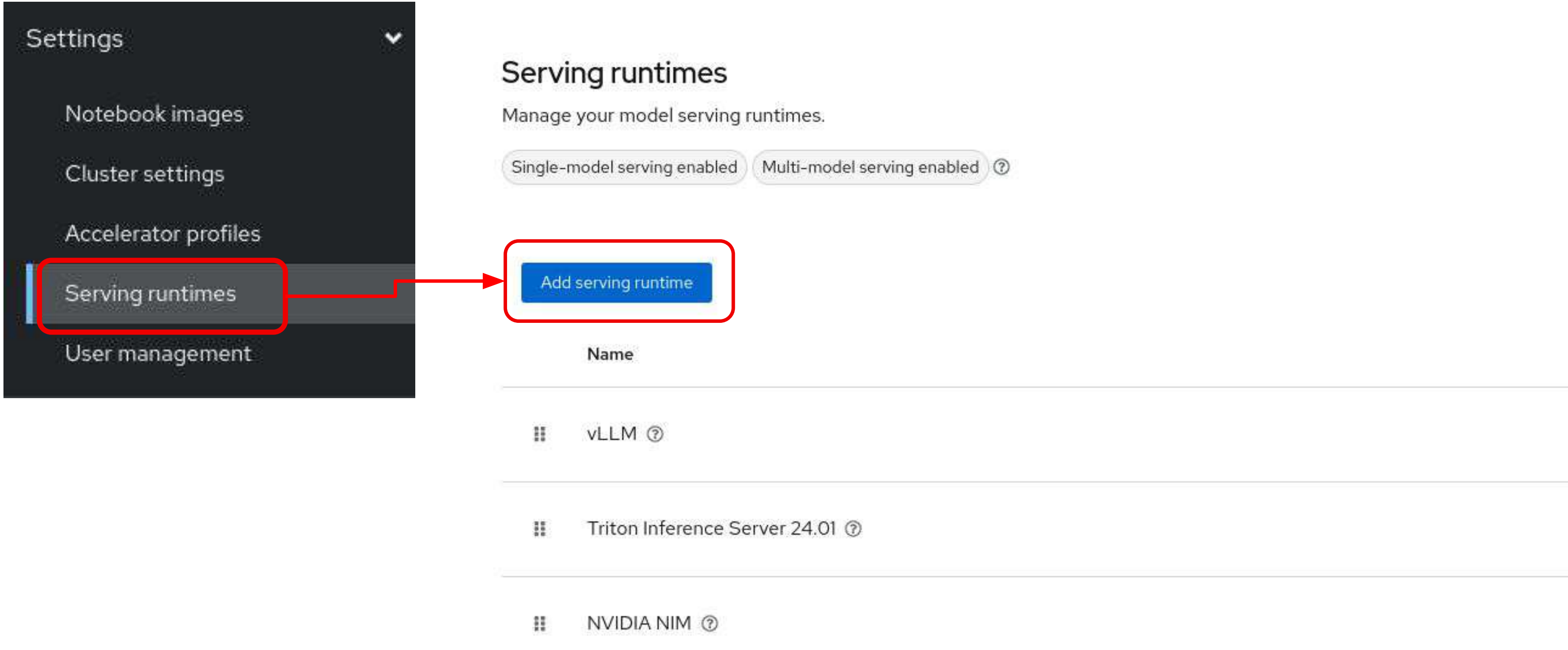
Data connections

Pipelines

Models and model servers

Custom Serving Runtime

Add serving runtime

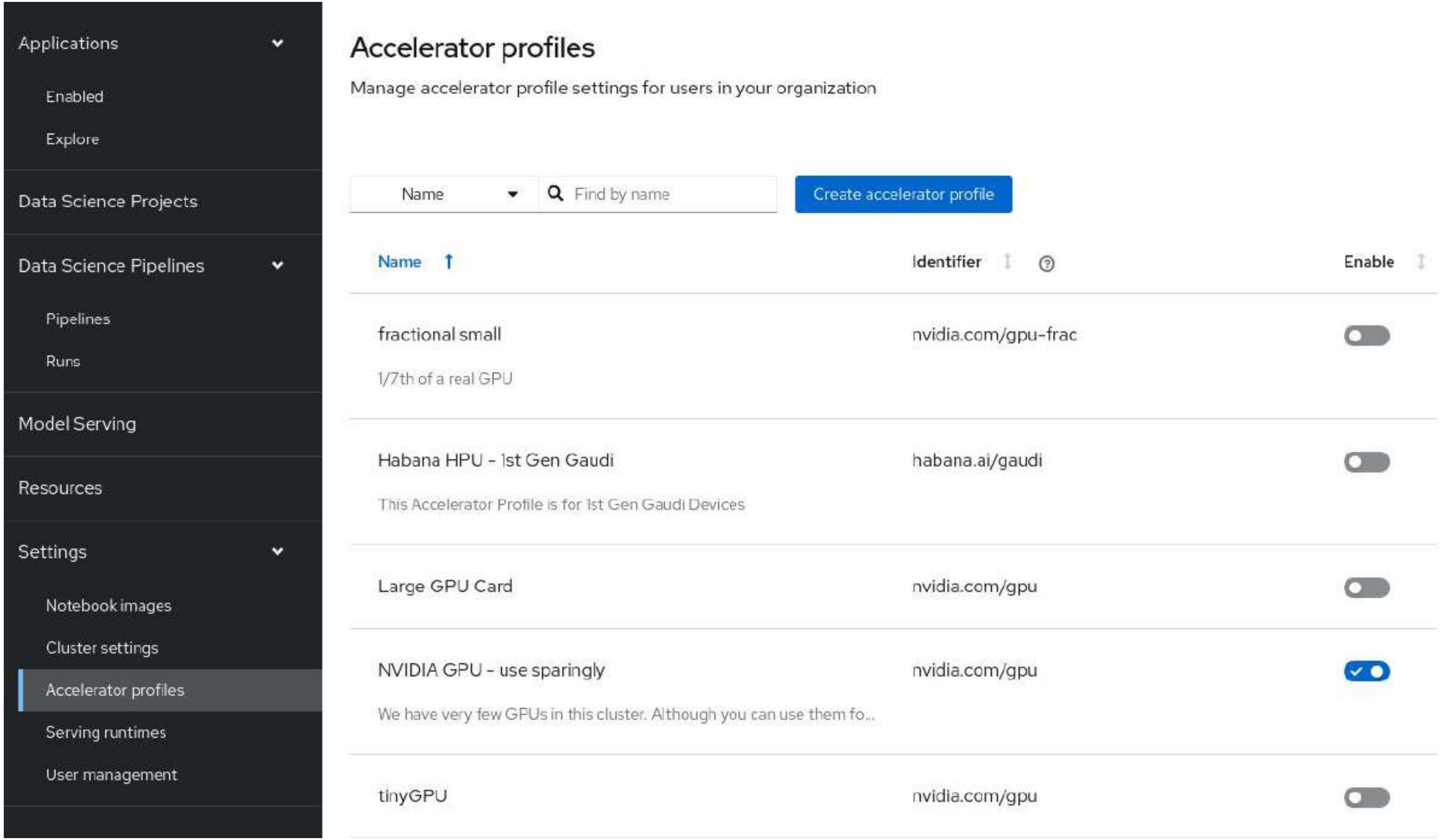


Serving runtimes

Manage your model serving runtimes.

Name
vLLM
Triton Inference Server 24.01
NVIDIA NIM

Accelerator Profile

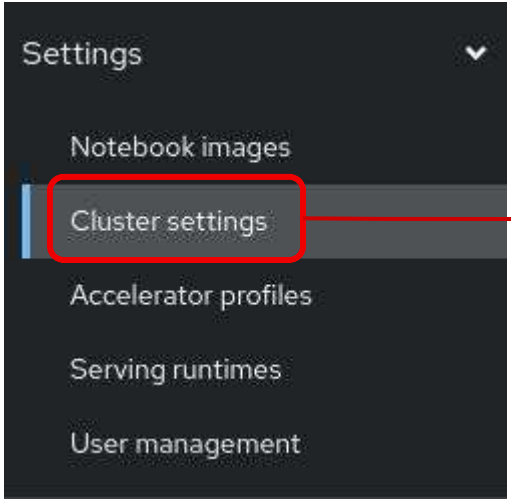


Accelerator profiles

Manage accelerator profile settings for users in your organization

Name	Identifier	Enable
fractional small 1/7th of a real GPU	nvidia.com/gpu-frac	○
Habana HPU - 1st Gen Gaudi This Accelerator Profile is for 1st Gen Gaudi Devices	habana.ai/gaudi	○
Large GPU Card	nvidia.com/gpu	○
NVIDIA GPU - use sparingly We have very few GPUs in this cluster. Although you can use them fo...	nvidia.com/gpu	✓
tinyGPU	nvidia.com/gpu	○

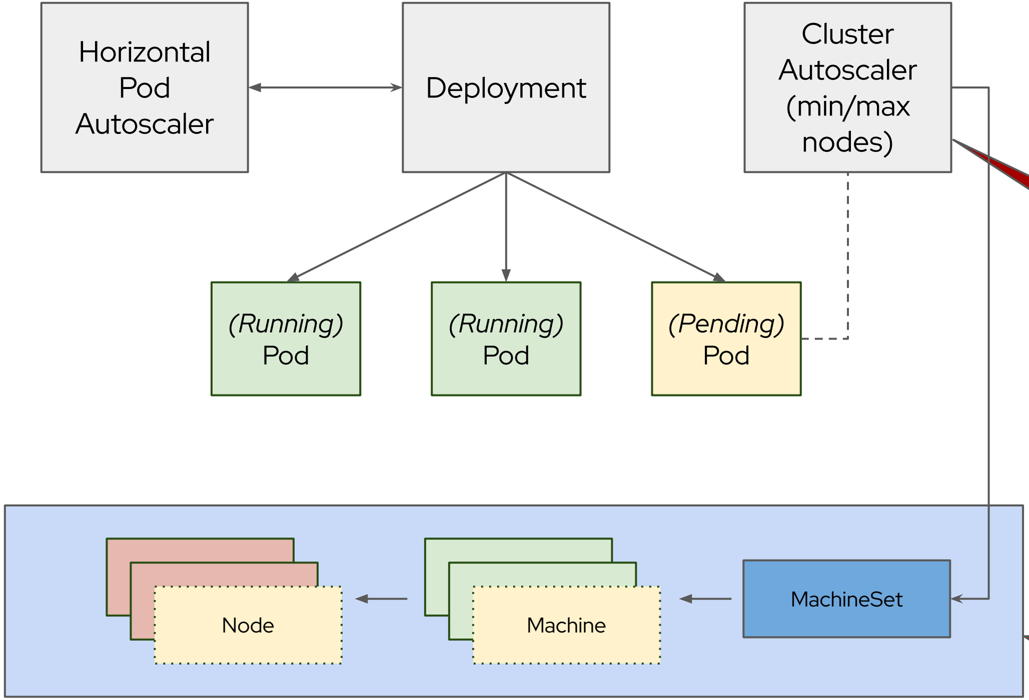
Cluster Settings



1. Model serving platforms
2. PVC size
3. Stop idle notebooks
4. Usage data collection
5. Notebook pod tolerations

Cluster Autoscaling

Automatically responding to cluster demand provisioning new nodes (incl. GPUs)

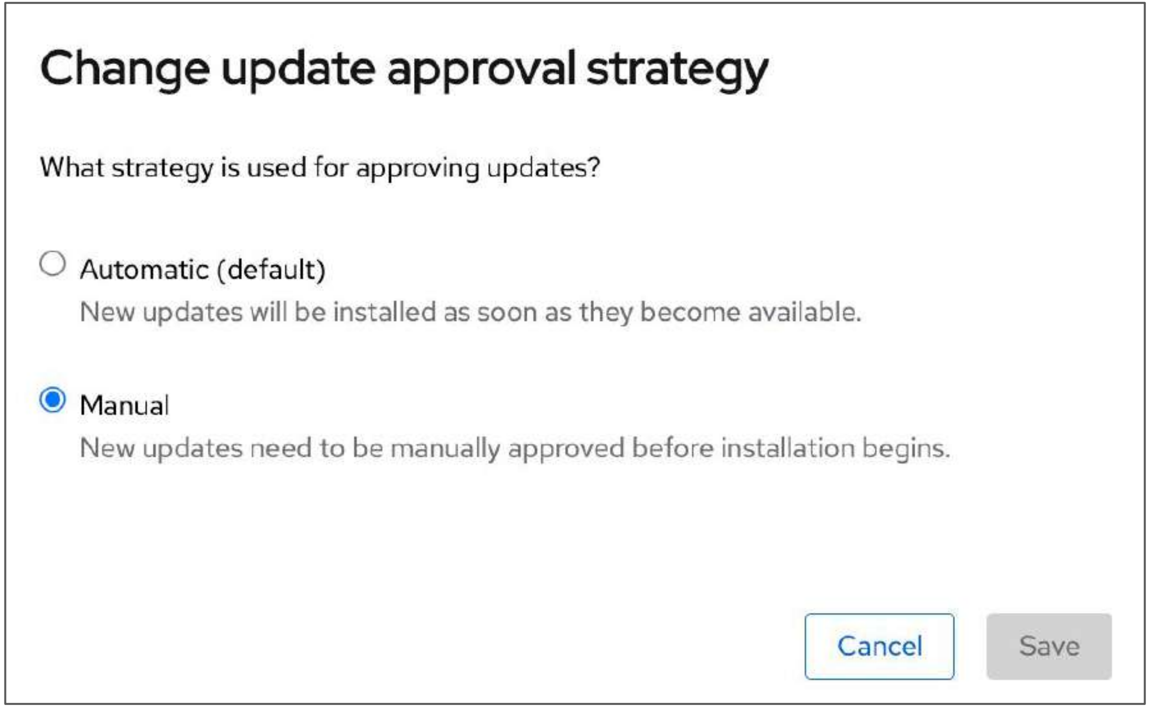


The cluster autoscaler scales the MachineSet up to accommodate the pending pod.

Once the MachineSet scales up, it creates a new machine, which creates a new node.

Update RHAI

Automatic vs Manual



Change update approval strategy

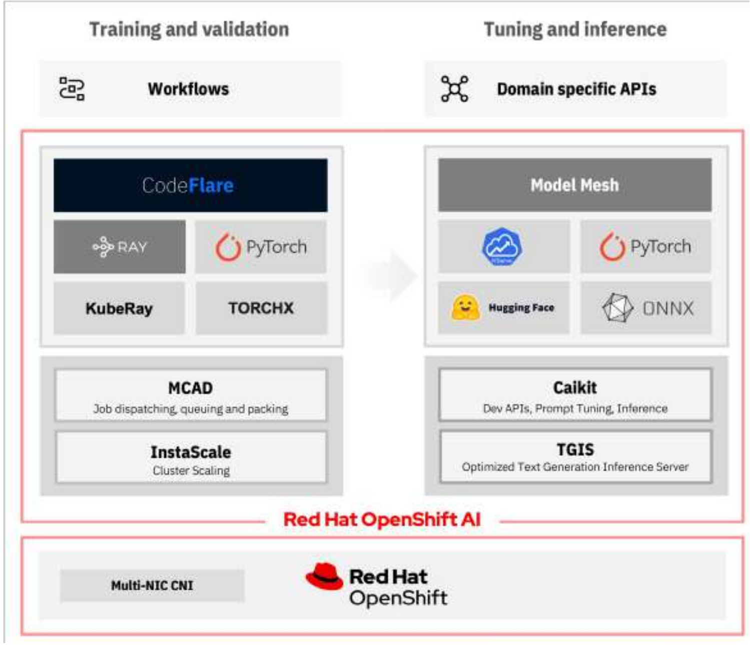
What strategy is used for approving updates?

○ Automatic (default)
New updates will be installed as soon as they become available.

● Manual
New updates need to be manually approved before installation begins.

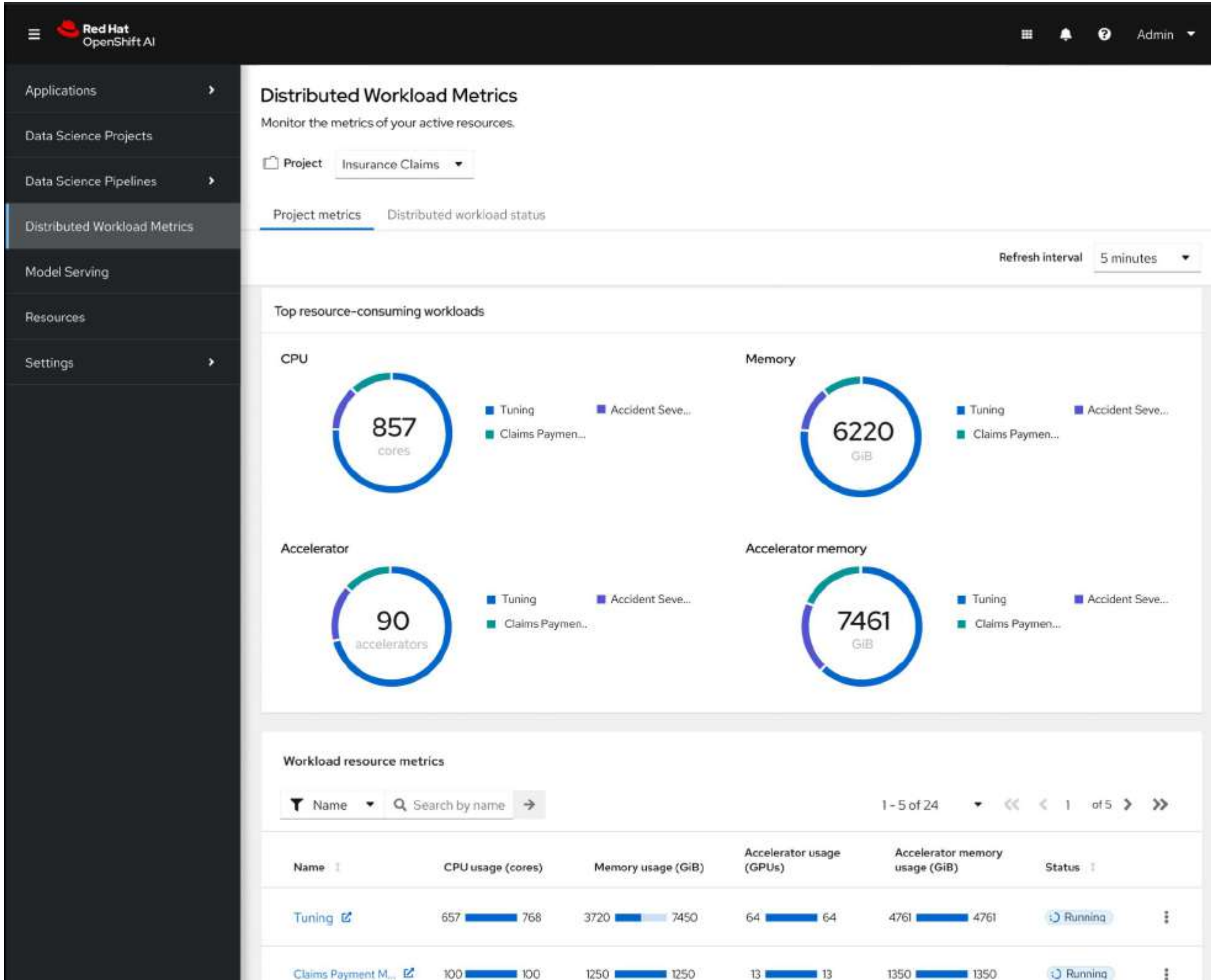
Distributed workloads

Distributed Workloads Overview



- Distributed training is used to distribute a **larger job across multiple nodes**, for example fine-tuning an LLM when a single node does not have enough GPUs.
- With the **CodeFlare** component in RHOAI, you can **spin up Ray clusters** inside your OpenShift cluster.
- You can then **submit jobs** to these **Ray clusters**, where the **jobs will be distributed** across a selected amount of nodes you have available.
- This also gives you **access to the Ray dashboard**, helping you **keep track** of the jobs and their logs.

Distributed Workloads



Distributed Workload Metrics

Monitor the metrics of your active resources.

Project: Insurance Claims

Refresh interval: 5 minutes

Top resource-consuming workloads

CPU: 857 cores — ■ Tuning ■ Accident Seve... ■ Claims Paymen...

Memory: 6220 GiB — ■ Tuning ■ Accident Seve... ■ Claims Paymen...

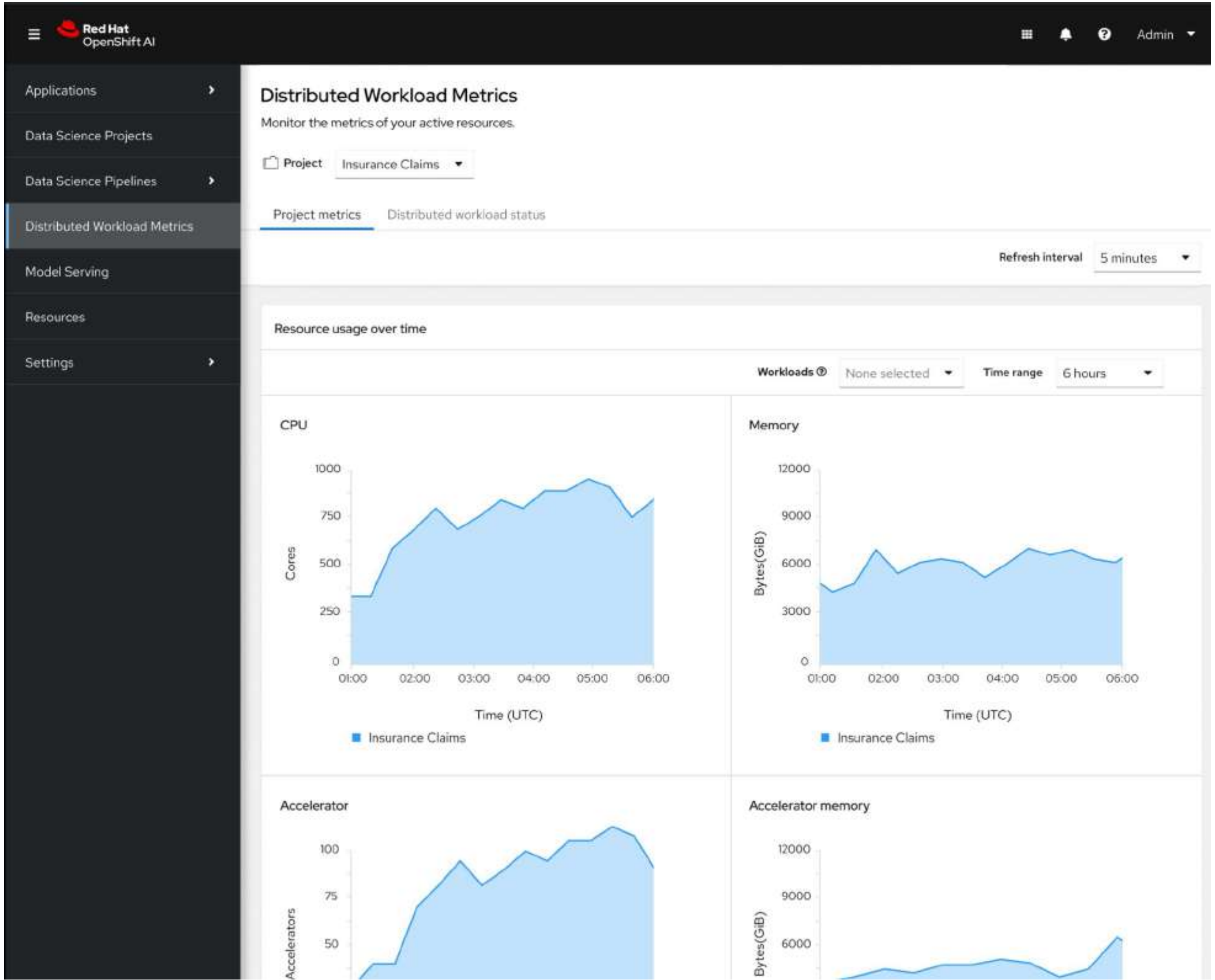
Accelerator: 90 accelerators — ■ Tuning ■ Accident Seve... ■ Claims Paymen..

Accelerator memory: 7461 GiB — ■ Tuning ■ Accident Seve... ■ Claims Paymen...

Workload resource metrics

Name	CPU usage (cores)	Memory usage (GiB)	Accelerator usage (GPUs)	Accelerator memory usage (GiB)	Status
Tuning	657 768	3720 7450	64 64	4761 4761	Running
Claims Payment M...	100 100	1250 1250	13 13	1350 1350	Running

Distributed Workloads



Distributed Workload Metrics

Monitor the metrics of your active resources.

Project: Insurance Claims

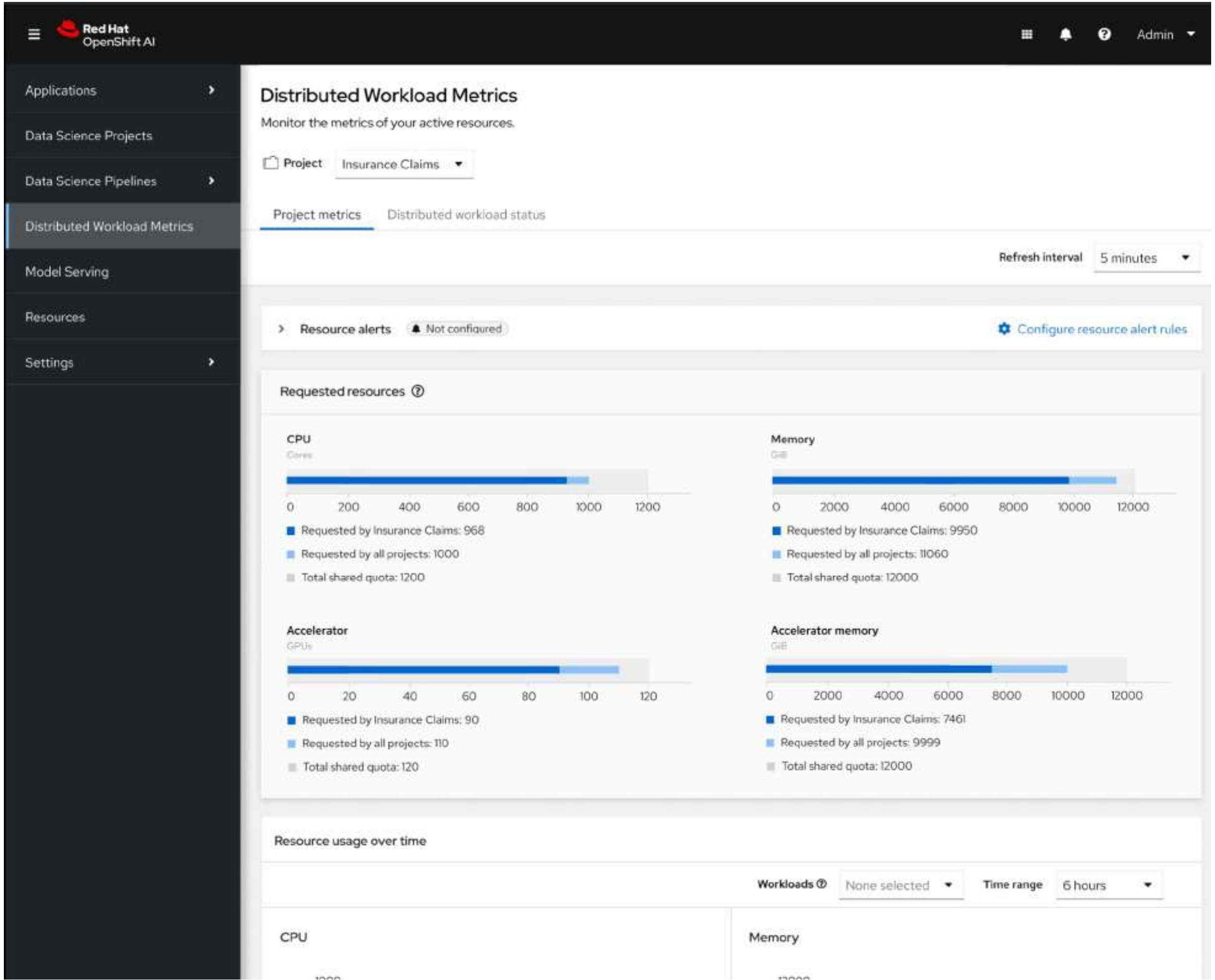
Refresh interval: 5 minutes

Resource usage over time

Workloads: None selected

Time range: 6 hours

Distributed Workloads



Distributed Workload Metrics

Monitor the metrics of your active resources.

Project: Insurance Claims

Refresh interval: 5 minutes

Resource alerts — Not configured

Requested resources

CPU (Cores)

- Requested by Insurance Claims: 968
- Requested by all projects: 1000
- Total shared quota: 1200

Memory (GiB)

- Requested by Insurance Claims: 9950
- Requested by all projects: 11060
- Total shared quota: 12000

Accelerator (GPUs)

- Requested by Insurance Claims: 90
- Requested by all projects: 110
- Total shared quota: 120

Accelerator memory (GiB)

- Requested by Insurance Claims: 7461
- Requested by all projects: 9999
- Total shared quota: 12000

Resource usage over time

Workloads: None selected

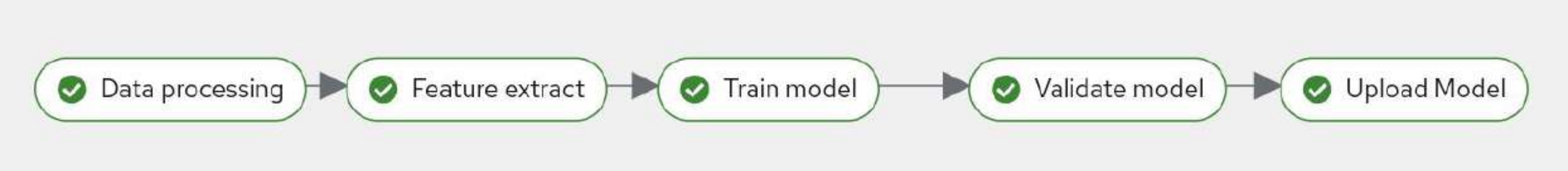
Time range: 6 hours

DS Pipelines

Data Science Pipelines

Data Science Pipelines

- Portable ML workflows to automate end-to-end ML tasks.
- Enables continuous integration and deployment of machine learning operations in staging and production.
- **Based on Kubeflow pipelines**. This internally **leverages Argo Workflows** to run the ML workflows.
- Example:
  - Here is a sample workflow that automates the ML tasks of processing data, extracting features from the data, train the ml model, validate it and upload the model to s3 object store.



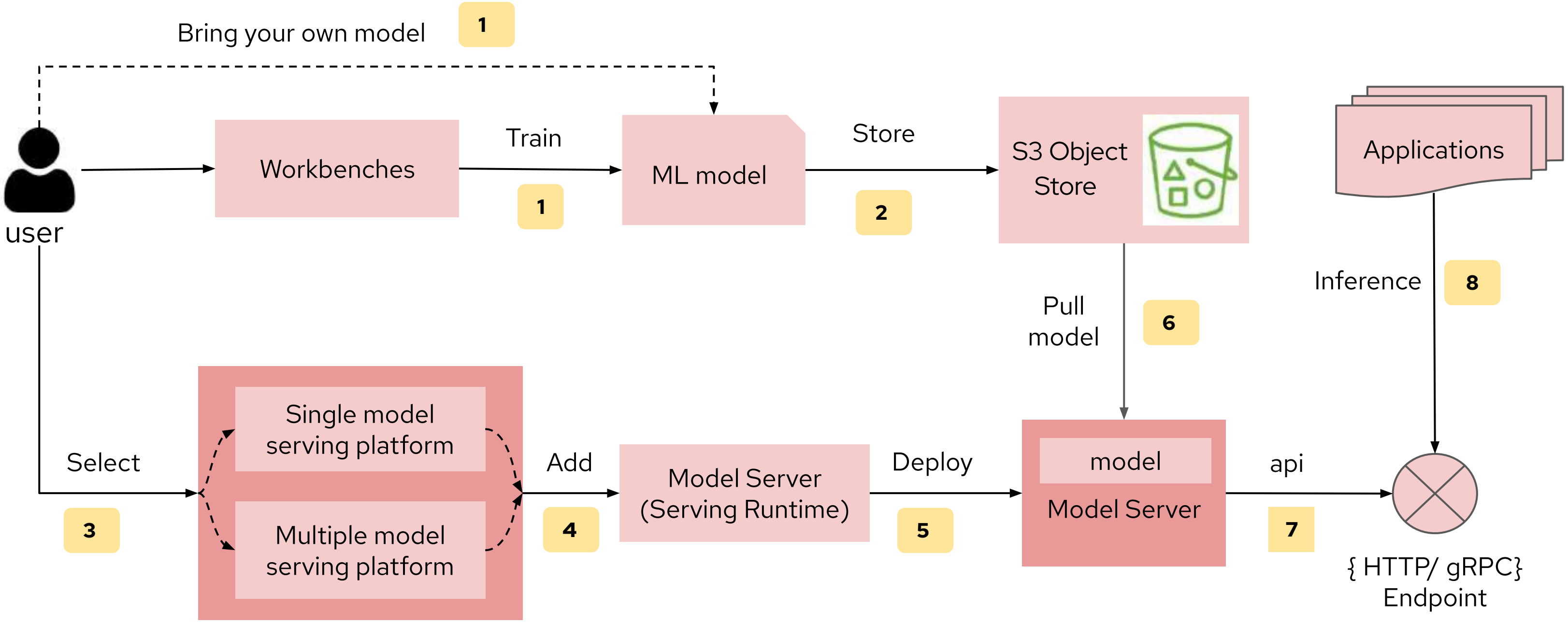
Data Science Pipelines

- Users can have **one pipeline server per project** and execute multiple pipelines.
- Pipelines uses a **Object Storage** to
    - store artifacts such as logs, data passed between steps, dependency files, and results.
- **Share data** between steps through:
    - Through parameters (small data)
    - Through volumes (large data)
    - Object storage (large data)
- **Experiment tracking**
    - Pipeline runs can be used as experiments, and the run view can be used to track those experiments.

Model Serving

Model Serving Workflow

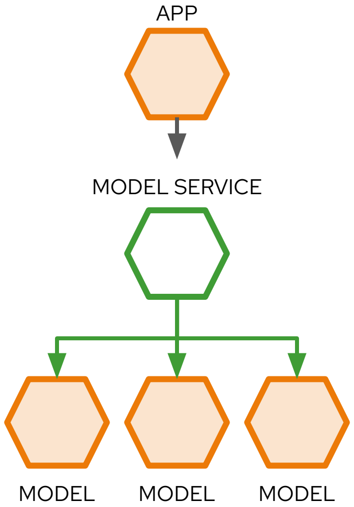
Model serving allows exposing the predictive or generative function of machine learning models in the form of an api.



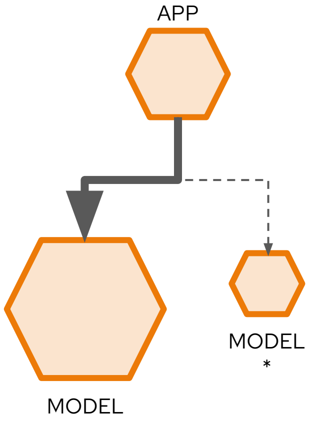
Model Serving

Models as stateless microservices

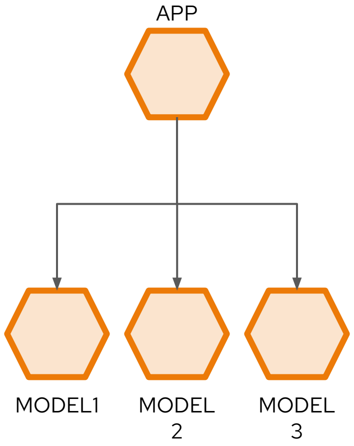
SCALE HORIZONTALLY



PHASED ROLLOUTS



MULTIPLE TRIALS



Conclusion

- Challenges of Operationalizing AI ?
- Team topologies and operationalizing models
- MLOps and Infra-as-Code
- Why application platforms?
- Where to start?
- RHAI walkthrough

Red Hat

Thank you!

Yury Titov

ytitov@redhat.com

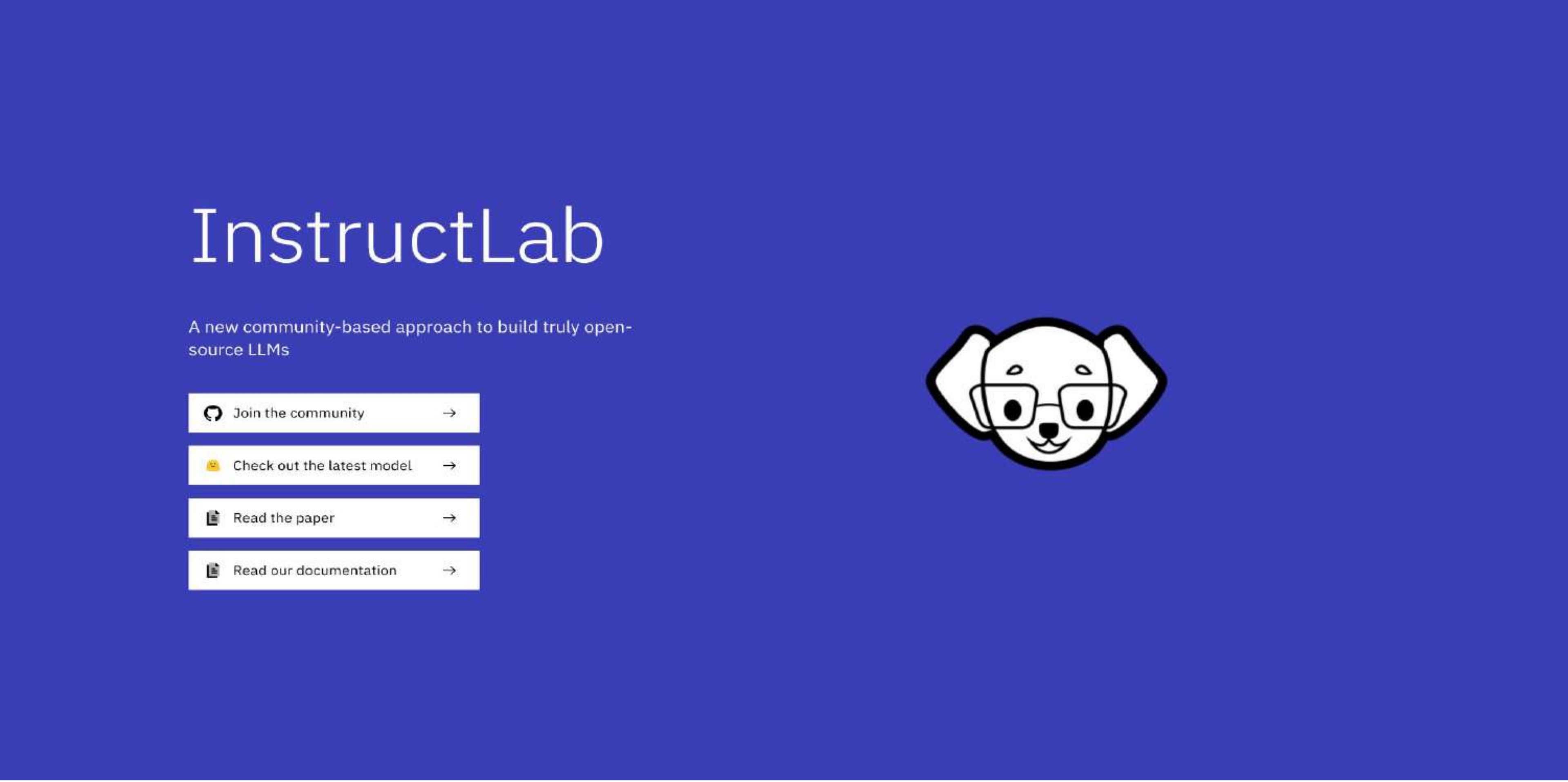
in linkedin.com/company/red-hat

youtube.com/user/RedHatVideos

facebook.com/redhatinc

twitter.com/RedHat

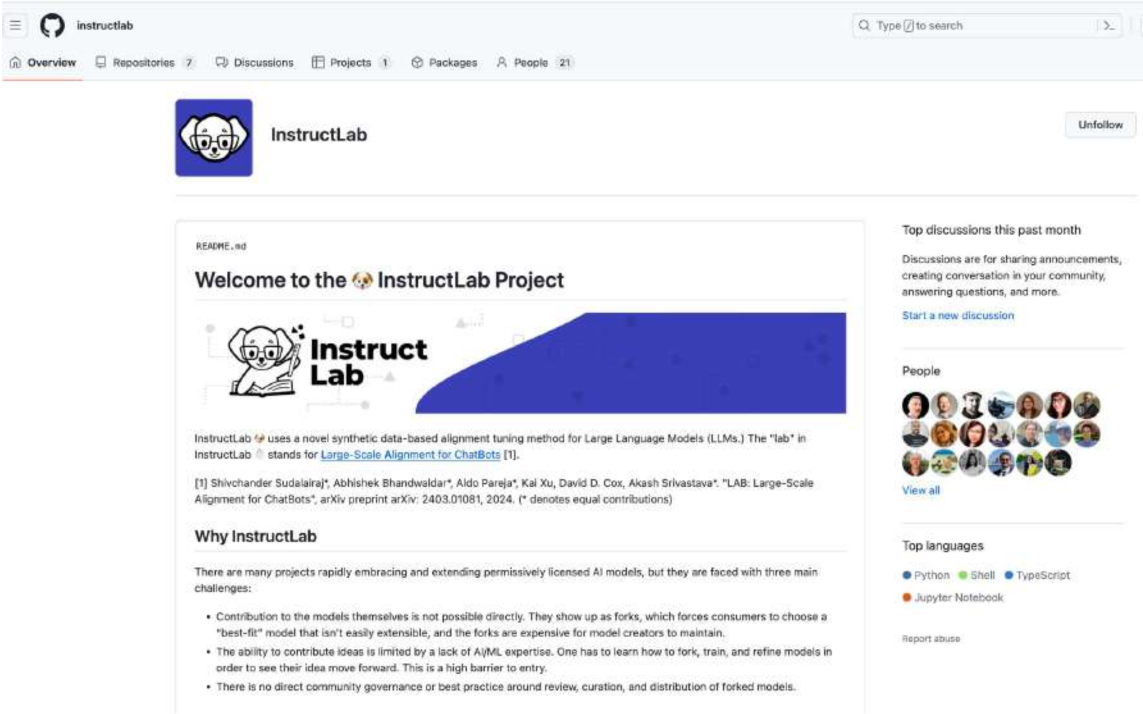
Community



InstructLab

A new community-based approach to build truly open-source LLMs

InstructLab: Open source community for Gen AI model development



Welcome to the 🐶 InstructLab Project

InstructLab 🐶 uses a novel synthetic data-based alignment tuning method for Large Language Models (LLMs.) The "lab" in InstructLab 🐶 stands for Large-Scale Alignment for ChatBots [1].

[1] Shivchander Sudalairaj*, Abhishek Bhandwaldar*, Aldo Pareja*, Kai Xu, David D. Cox, Akash Srivastava*. "LAB: Large-Scale Alignment for ChatBots", arXiv preprint arXiv: 2403.01081, 2024. (* denotes equal contributions)

Why InstructLab

There are many projects rapidly embracing and extending permissively licensed AI models, but they are faced with three main challenges:

- Contribution to the models themselves is not possible directly. They show up as forks, which forces consumers to choose a "best-fit" model that isn't easily extensible, and the forks are expensive for model creators to maintain.
- The ability to contribute ideas is limited by a lack of AI/ML expertise. One has to learn how to fork, train, and refine models in order to see their idea move forward. This is a high barrier to entry.
- There is no direct community governance or best practice around review, curation, and distribution of forked models.

Top discussions this past month

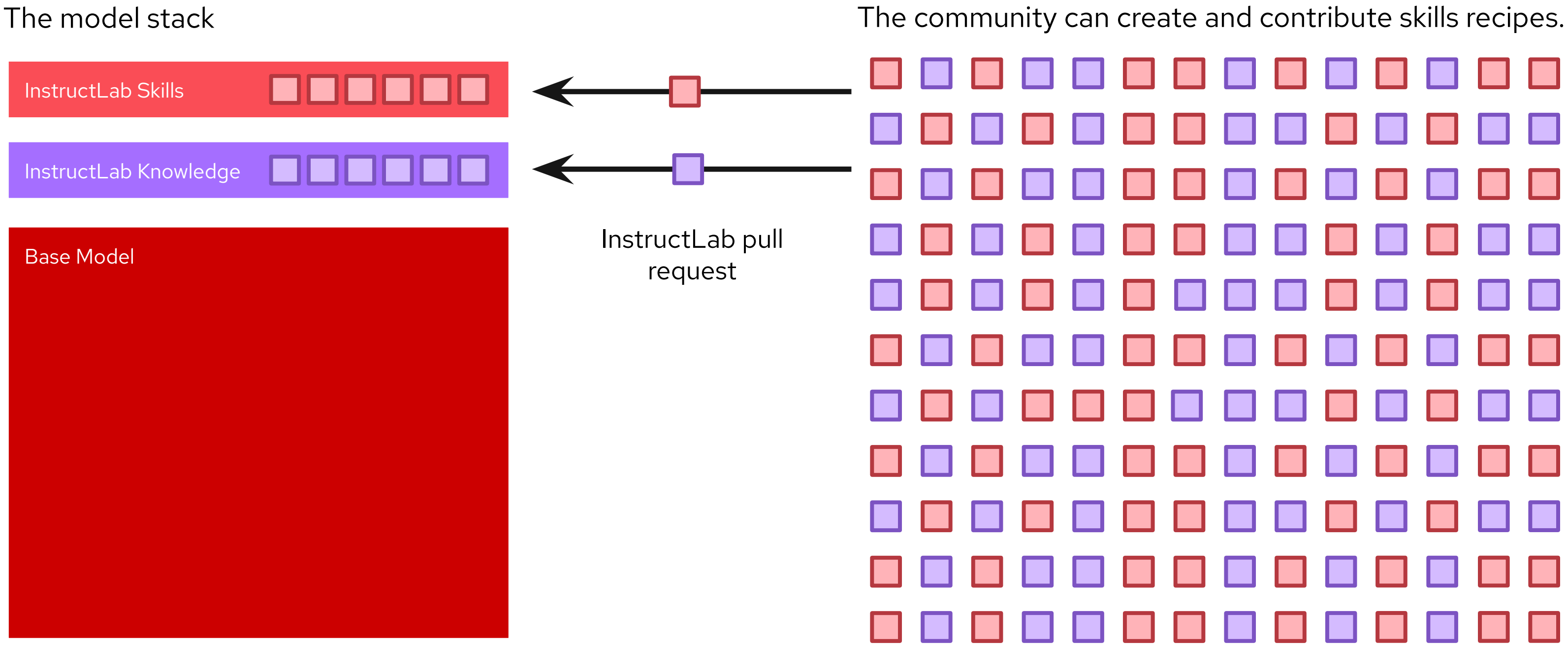
Discussions are for sharing announcements, creating conversation in your community, answering questions, and more.

Start a new discussion

Top languages

● Python ● Shell ● TypeScript
● Jupyter Notebook

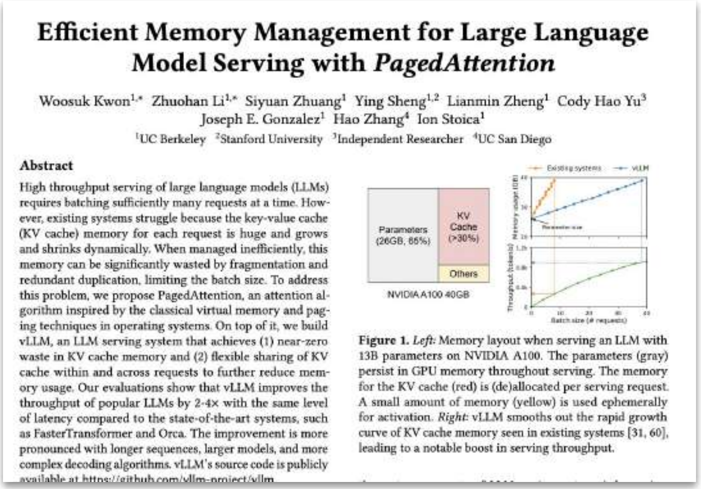
InstructLab enables **community-driven** development and evolution of models



vLLM: Neural Magic

vLLM: A 2 Year Journey of Performance

vLLM has rapidly evolved from a research project to the open source default.



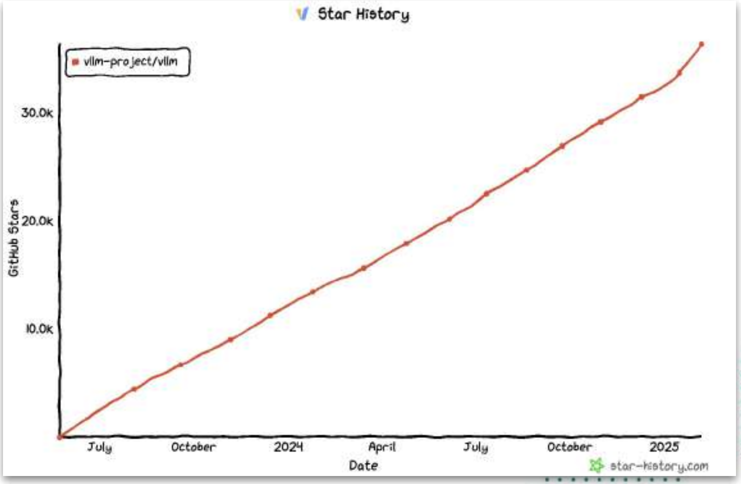
Efficient Memory Management for Large Language Model Serving with *PagedAttention*

Woosuk Kwon[1,*] Zhuohan Li[1,*] Siyuan Zhuang[1] Ying Sheng[1,2] Lianmin Zheng[1] Cody Hao Yu[3]
Joseph E. Gonzalez[1] Hao Zhang[4] Ion Stoica[1]
[1]UC Berkeley [2]Stanford University [3]Independent Researcher [4]UC San Diego

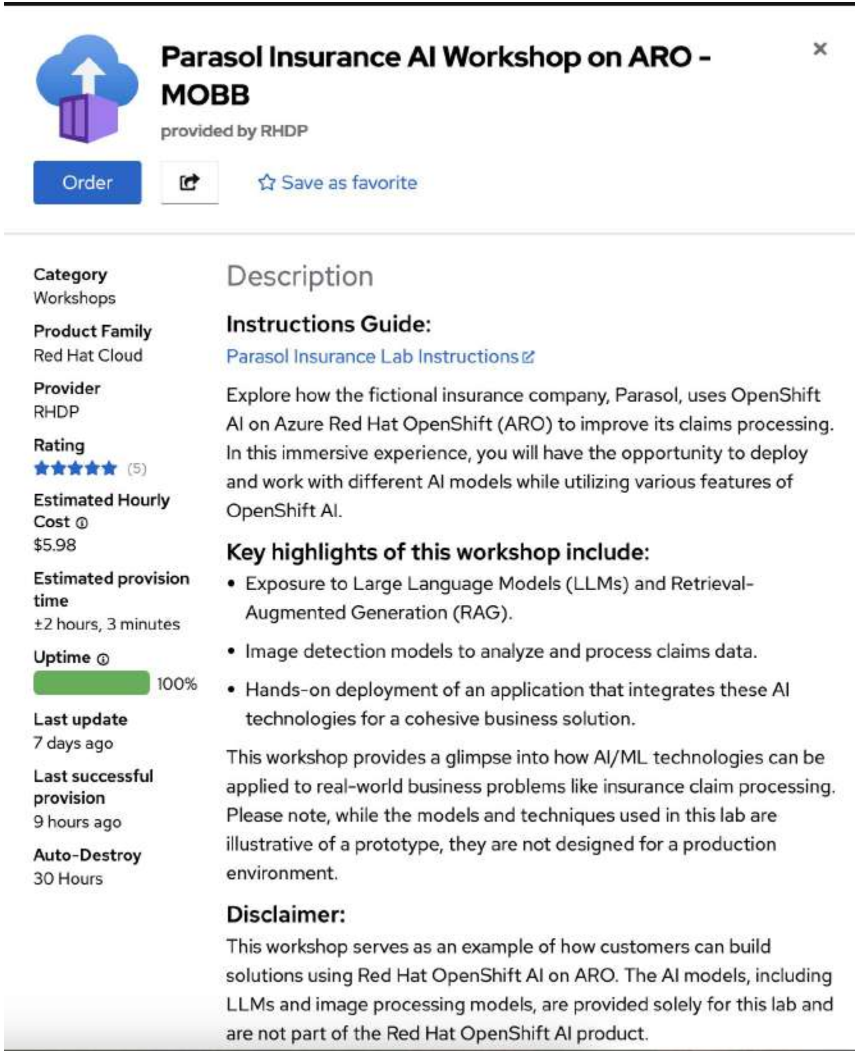
Abstract

High throughput serving of large language models (LLMs) requires batching sufficiently many requests at a time. However, existing systems struggle because the key-value cache (KV cache) memory for each request is huge and grows and shrinks dynamically. When managed inefficiently, this memory can be significantly wasted by fragmentation and redundant duplication, limiting the batch size. To address this problem, we propose PagedAttention, an attention algorithm inspired by the classical virtual memory and paging techniques in operating systems. On top of it, we build vLLM, an LLM serving system that achieves (1) near-zero waste in KV cache memory and (2) flexible sharing of KV cache within and across requests to further reduce memory usage. Our evaluations show that vLLM improves the throughput of popular LLMs by 2-4× with the same level of latency compared to the state-of-the-art systems, such as FasterTransformer and Orca. The improvement is more pronounced with longer sequences, larger models, and more complex decoding algorithms. vLLM's source code is publicly available at https://github.com/vllm-project/vllm

Figure 1. *Left:* Memory layout when serving an LLM with 13B parameters on NVIDIA A100. The parameters (gray) persist in GPU memory throughout serving. The memory for the KV cache (red) is (de)allocated per serving request. A small amount of memory (yellow) is used ephemerally for activation. *Right:* vLLM smooths out the rapid growth curve of KV cache memory seen in existing systems [31, 60], leading to a notable boost in serving throughput.



Pervasive → 100k daily installs in Jan 2025
Explosive Growth → 10x usage increase in 2024



Parasol Insurance AI Workshop on ARO - MOBB

provided by RHDP

Category
Workshops

Product Family
Red Hat Cloud

Provider
RHDP

Rating
★★★★★ (5)

Estimated Hourly Cost ⓘ
$5.98

Estimated provision time
±2 hours, 3 minutes

Uptime ⓘ
🟩 100%

Last update
7 days ago

Last successful provision
9 hours ago

Auto-Destroy
30 Hours

Description

Instructions Guide:

Parasol Insurance Lab Instructions

Explore how the fictional insurance company, Parasol, uses OpenShift AI on Azure Red Hat OpenShift (ARO) to improve its claims processing. In this immersive experience, you will have the opportunity to deploy and work with different AI models while utilizing various features of OpenShift AI.

Key highlights of this workshop include:

- Exposure to Large Language Models (LLMs) and Retrieval-Augmented Generation (RAG).
- Image detection models to analyze and process claims data.
- Hands-on deployment of an application that integrates these AI technologies for a cohesive business solution.

This workshop provides a glimpse into how AI/ML technologies can be applied to real-world business problems like insurance claim processing. Please note, while the models and techniques used in this lab are illustrative of a prototype, they are not designed for a production environment.

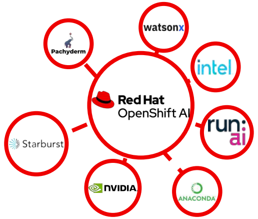
Disclaimer:

This workshop serves as an example of how customers can build solutions using Red Hat OpenShift AI on ARO. The AI models, including LLMs and image processing models, are provided solely for this lab and are not part of the Red Hat OpenShift AI product.

Operationalize AI with Red Hat OpenShift AI’s ecosystem

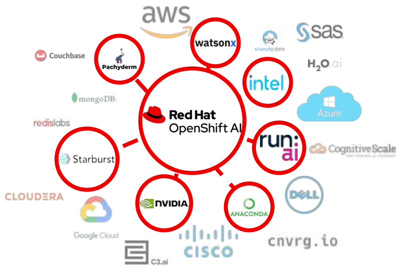
Red Hat’s partner ecosystem on AI/ML

Empowers choice with the **best-of-breed AI technologies** from a certified partner ecosystem that solves for customers use case, capabilities and deployment options



Red Hat OpenShift AI integrated technology partners

Technology has been integrated into Red Hat OpenShift AI to complement the platform and extend capabilities.



Red Hat OpenShift certified partner ecosystem

Certified AI/ML vendors that provide a native integration to OpenShift and provide complementary or extended capabilities to Red Hat OpenShift AI

Red Hat OpenShift AI

Gather and prepare data

Solutions for data access, preparation and storage

watsonx	Open data lakehouse architecture to store, organize, and access data
Pachyderm	Brings "git for data" for data versioning and governance
Starburst	Analytics engine accessing data where it lives

Develop or tune model

Support for experimentation and model tuning

watsonx	Expands the reach of AI to business users & democratizes AI
run:ai	Increase hardware utilization using fractional GPUs and node scaling
NVIDIA	Accelerate model training & tuning
intel AI TOOLS	Maximize training performance on Intel architecture
ANACONDA	Provides open source packages & libraries and data science distribution

Integrate models in app dev

Infrastructure for model deployment

watsonx	Easily deploy generative AI and ML models to production
run:ai	Optimize compute resources to significantly cut costs
NVIDIA	High performance model inferencing
intel OpenVINO	Fully integrated model dev environment and optimizes your model for inference on Intel hardware

Model monitoring and management

Monitor and manage for responsibility and transparency

watsonx	Toolkit to help manage and monitor the risk

Important partner: Starburst

Data Services for Modern AI/ML Use Cases

Performance

From petabytes to exabytes – query data from disparate sources using SQL – with high concurrency

Control your price/performance with the latest cost-based optimizer

Caching available for frequently accessed data

Connectivity

40+ supported enterprise connectors

High performance parallel connectors for Oracle, Teradata, Snowflake and more

Ceph

Red Hat AMQ

Microsoft SQL Server

teradata.

ORACLE

EDB

Couchbase

Security

Kerberos, LDAP & SSO Integration

Global Security for fine-grained access control

Data Encryption/Masking

Higher security posture than vanilla K8's

Apache Ranger

KEYCLOAK

Management

Configuration

Autoscaling & High Availability

Query/Cluster Monitoring

Deploy Anywhere

Multi-Cluster Management

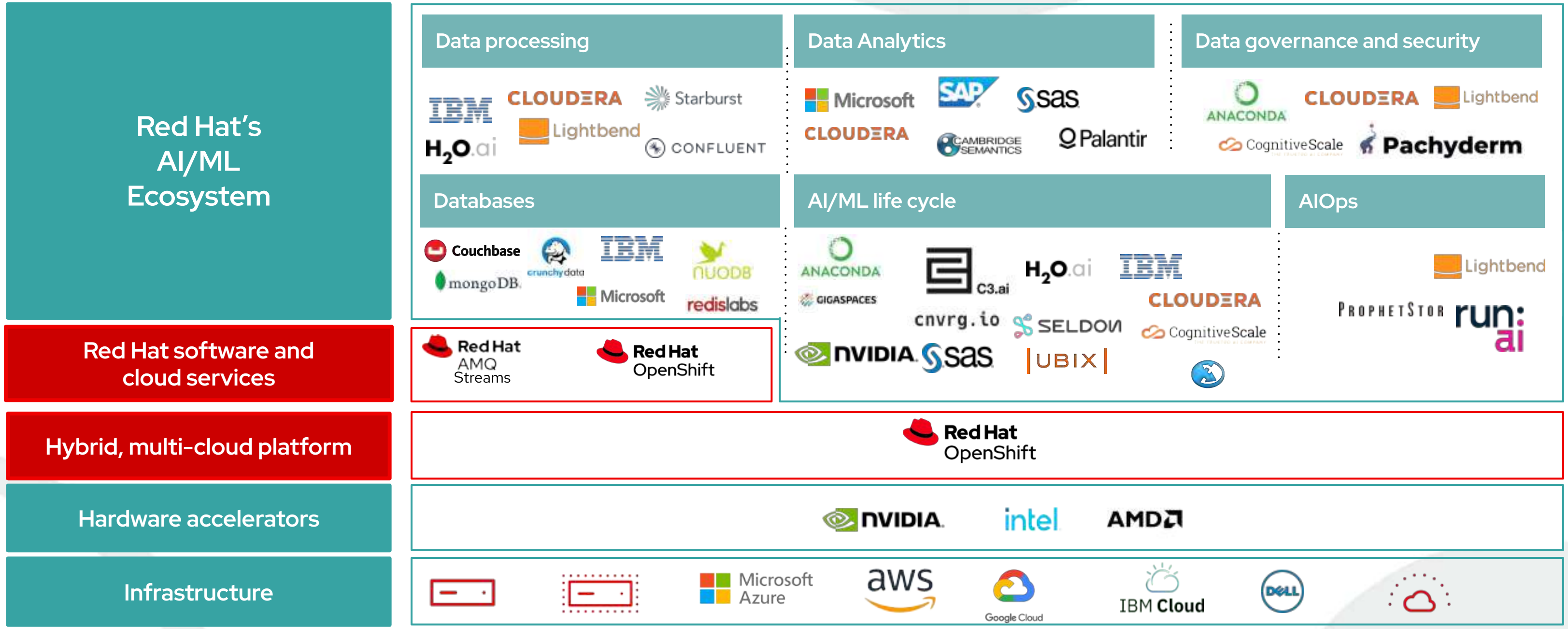
Red Hat OpenShift

aws

Making AI accessible to all users

	Features	Use case	Persona
watsonx.ai™	Foundation models	Ready-to-use models	AI Builders
	AI studio	UI for training, prompt-tuning and experimentation	Citizen data scientists and Data scientists
Red Hat OpenShift AI	AI/ML platform	Distributed training and serving mechanisms	Data scientists and MLOps
Red Hat OpenShift	App platform	Fleet management and apps life cycle management	App Developers and DevOps
GPUs	Hardware accelerators	Resource management	DevOps
	Deploy anywhere	Hybrid and multicloud deployments	Platform engineers

Strategic partnerships + Red Hat AI/ML offerings



Red Hat's AI/ML Ecosystem

Data processing

Data Analytics

Data governance and security

Databases

AI/ML life cycle

AIOps

Red Hat software and cloud services

Hybrid, multi-cloud platform

Hardware accelerators

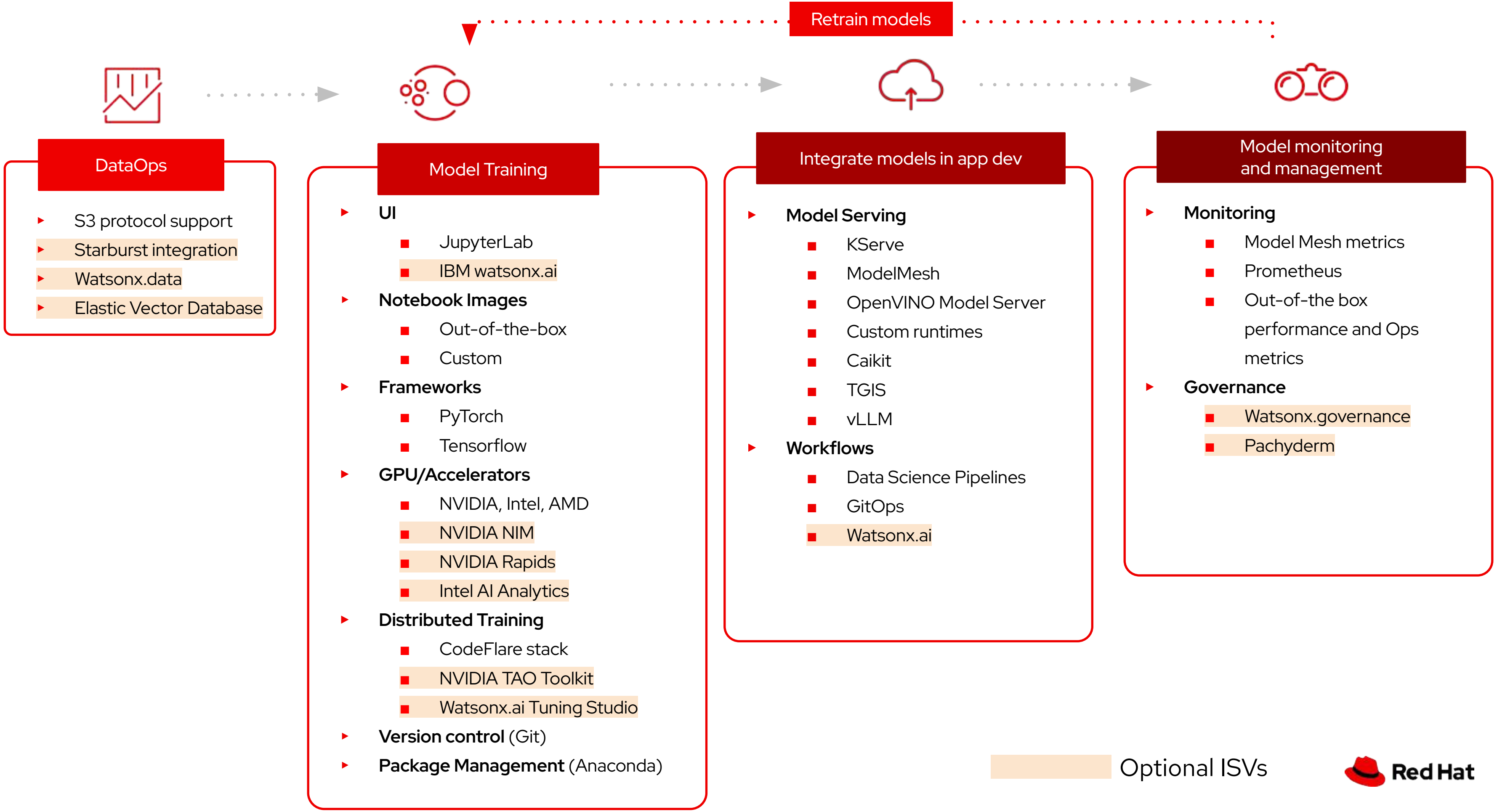
Infrastructure

Customer experiences

Platform Users

Product overview

Product Overview



DataOps

- S3 protocol support
- Starburst integration
- Watsonx.data
- Elastic Vector Database

Model Training

- **UI**
  - JupyterLab
  - IBM watsonx.ai
- **Notebook Images**
  - Out-of-the-box
  - Custom
- **Frameworks**
  - PyTorch
  - Tensorflow
- **GPU/Accelerators**
  - NVIDIA, Intel, AMD
  - NVIDIA NIM
  - NVIDIA Rapids
  - Intel AI Analytics
- **Distributed Training**
  - CodeFlare stack
  - NVIDIA TAO Toolkit
  - Watsonx.ai Tuning Studio
- **Version control** (Git)
- **Package Management** (Anaconda)

Integrate models in app dev

- **Model Serving**
  - KServe
  - ModelMesh
  - OpenVINO Model Server
  - Custom runtimes
  - Caikit
  - TGIS
  - vLLM
- **Workflows**
  - Data Science Pipelines
  - GitOps
  - Watsonx.ai

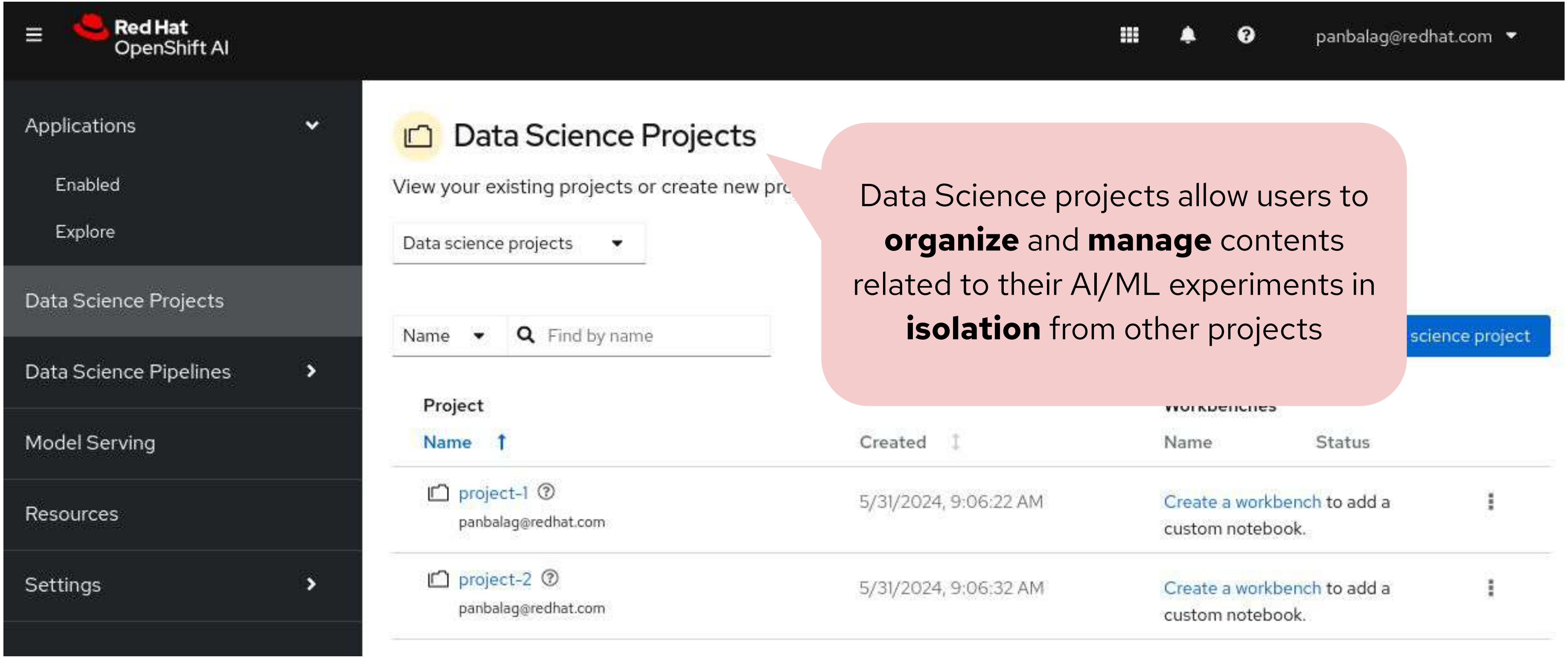
Model monitoring and management

- **Monitoring**
  - Model Mesh metrics
  - Prometheus
  - Out-of-the box performance and Ops metrics
- **Governance**
  - Watsonx.governance
  - Pachyderm

▨ Optional ISVs

Data Science Projects

Data Science Projects



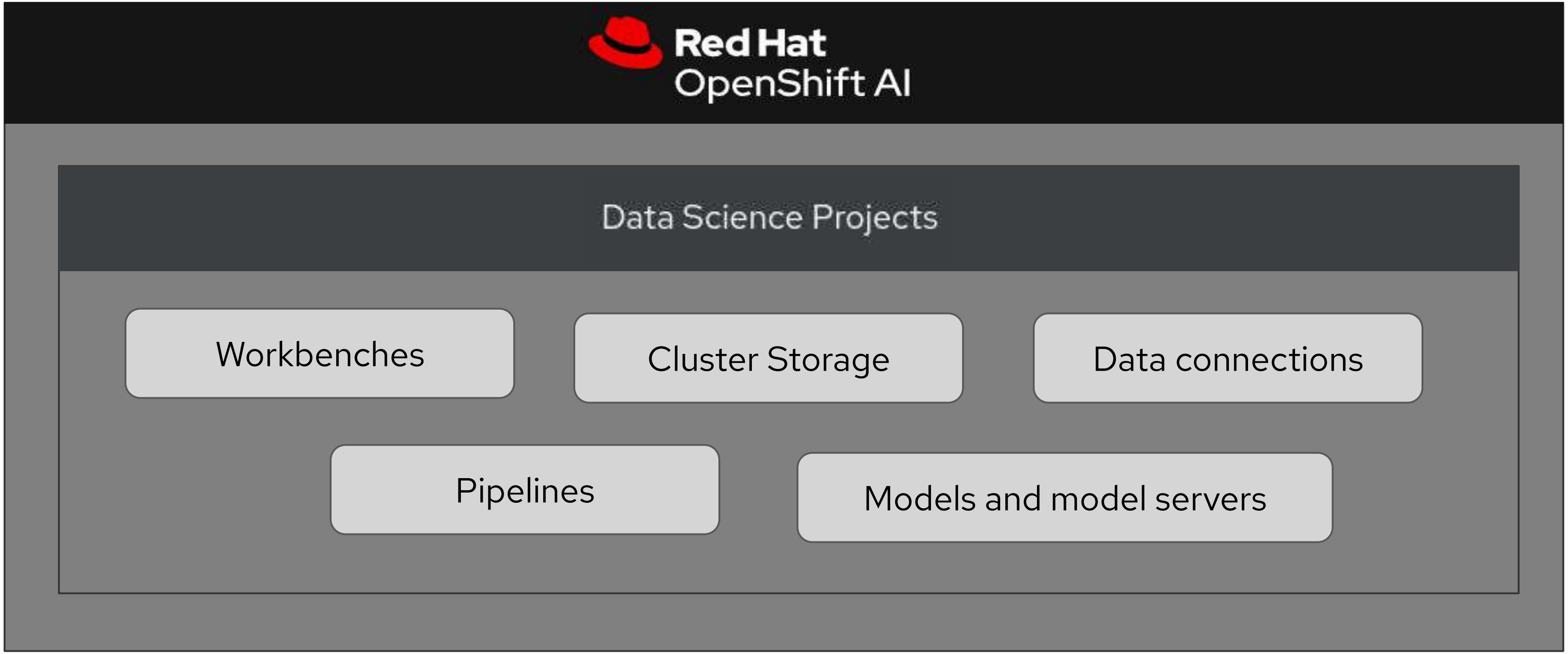
Data Science Projects

View your existing projects or create new pro

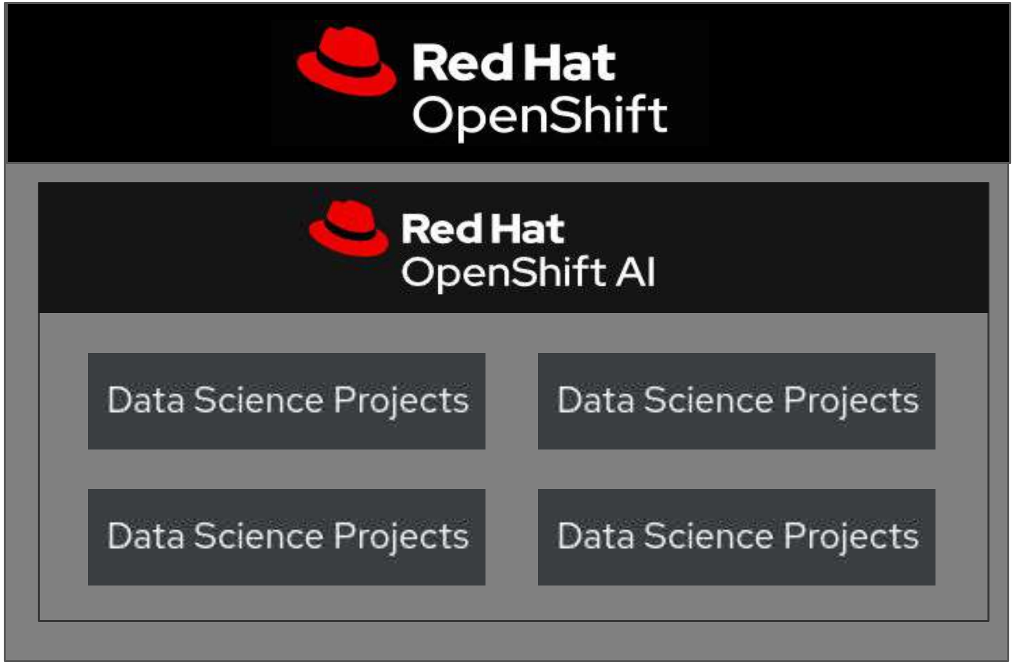
Project Name	Created	Workbenches Name	Status
project-1 panbalag@redhat.com	5/31/2024, 9:06:22 AM	Create a workbench to add a custom notebook.	
project-2 panbalag@redhat.com	5/31/2024, 9:06:32 AM	Create a workbench to add a custom notebook.	

Data Science projects allow users to **organize** and **manage** contents related to their AI/ML experiments in **isolation** from other projects

Data Science Projects

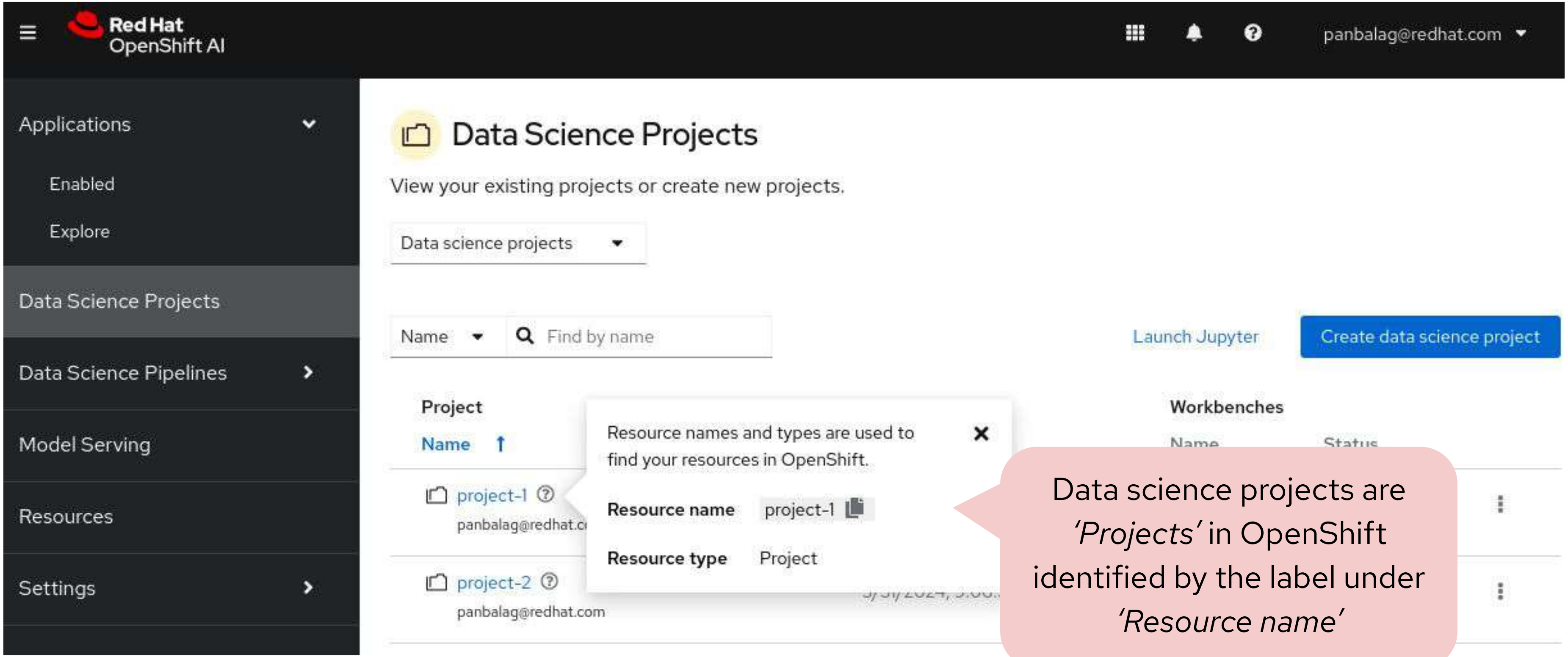


Data Science Projects



- Multiple data science projects.
- Isolation from other projects
- Created by admins or users
- User/Group access privileges

Data Science Projects



Data Science Projects

View your existing projects or create new projects.

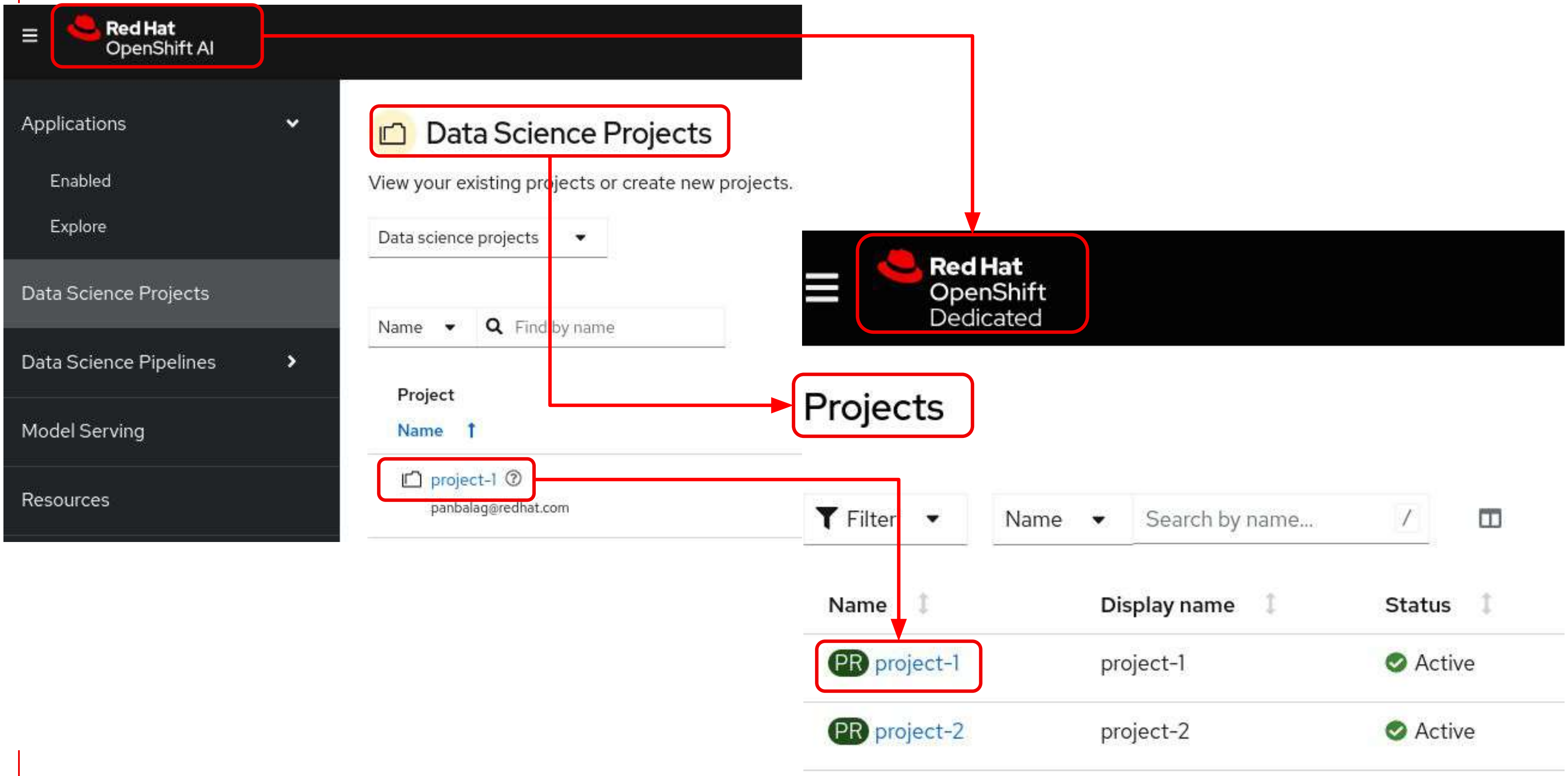
Resource names and types are used to find your resources in OpenShift.

Resource name project-1

Resource type Project

Data science projects are *'Projects'* in OpenShift identified by the label under *'Resource name'*

Data Science Projects



View your existing projects or create new projects.

Project Name
project-1 panbalag@redhat.com

Name	Display name	Status
PR project-1	project-1	✓ Active
PR project-2	project-2	✓ Active

Data Science Projects

Collaborate within a project

- Users that create a data science project
  - become an admin of that project
  - can give access to a project to any user or group
- Users with access permissions can access all resources in the project, modify them, and create new ones.
- Limiting user level access to data science projects needs to be handled at an OpenShift level at the moment

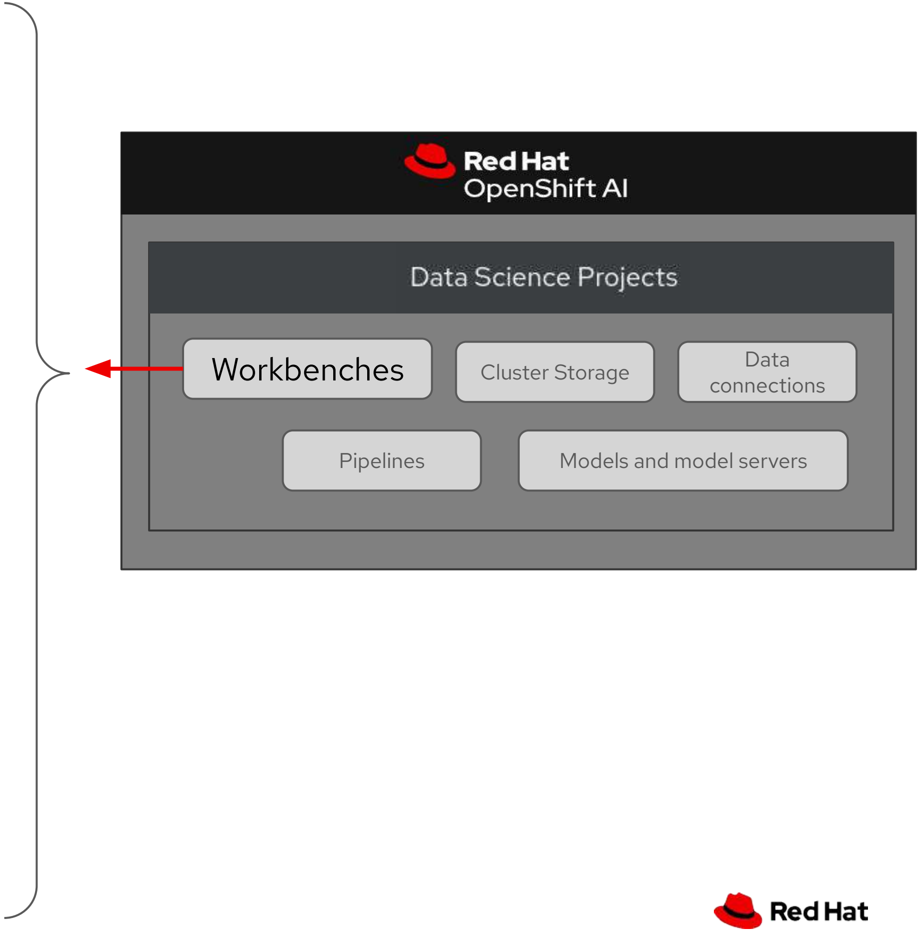
Collaborate between projects

- Due to isolation of data science projects, resources need to be explicitly exposed in order to be shared between projects.
- A good way to do this is to have an external resource which the projects have access to.
  - Examples:
    - A git repository with shared code
    - An object storage with shared artifacts
    - A structured database with shared data

Workbenches

Workbenches

- **Notebook Image**
  - **Development environment** in the form of a container image
    - combination of IDE like Jupyter Notebook, VSCode, etc., and choice of AI/ML framework like Tensorflow, PyTorch etc.,
  - **Custom notebook images**.
- **Deployment size**
  - Container size → **# CPUs** & **Memory** size
  - Accelerator → Choice of **Accelerators**/**GPUs**
- **Environment variables**
  - Config Map
  - Secret
- **Cluster Storage**
  - **PVC** connected to the development environment to store code & related artifacts.
- **Data connections**
  - **Object store** for hosting models as well as storing pipeline artifacts.



Red Hat

Workbenches

Default Notebook Images

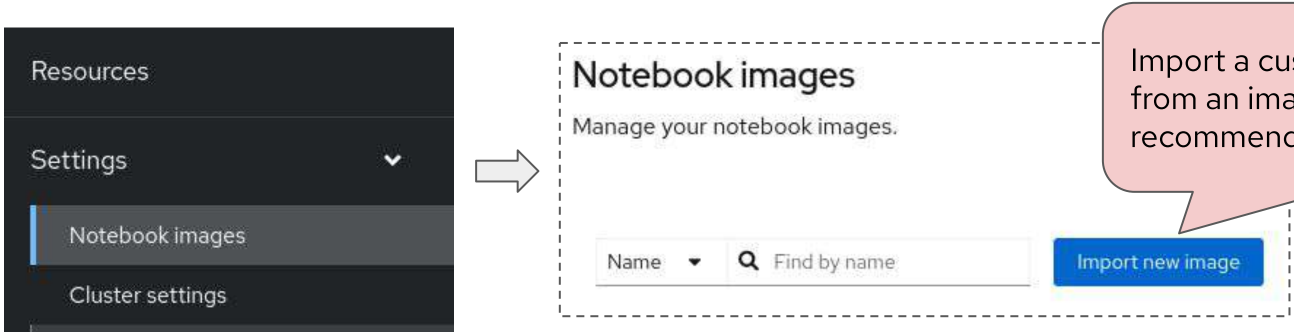
Image	Description
CUDA	For compute-intensive data science models that require GPU support, the Compute Unified Device Architecture (CUDA) notebook image provides **access to the NVIDIA CUDA Toolkit** with GPU-accelerated libraries and optimization tools.
Standard Data Science	Contains **commonly used libraries** to assist you in developing your machine learning models.
TensorFlow	**TensorFlow**, a popular open source machine learning platform. TensorFlow provides advanced libraries, data visualization features that allows users to build, monitor and track models.
PyTorch	**PyTorch** is another open source machine learning library optimized for deep learning like computer vision or natural language processing models.
Minimal Python	A **minimal environment with JupyterLab** for basic exploration.
Trusty AI	For AI/ML work with **model explainability, tracing, and accountability**, & runtime monitoring
Habana AI	For high-performance optimization of deep learning training workloads and maximize training throughput and efficiency with **Habana Gaudi devices**.
code-server (Technology Preview)	Provides you with a **VSCode** environment, allowing you to customize the environment through **extensions**.

Default notebook images

Workbenches

Customizing Workbenches

- To customize the workbench you can either:
  - Install dependencies on top of a workbench
  - Use a custom notebook image
- You can use package managers such as pip to add/remove dependencies in an existing workbench
  - Dependencies installed within the workbench are by default not saved to the persistent storage, this is by choice as restarting the workbench is an easy way to reset the environment if something caused an issue with the dependencies
- You can create and use custom notebook images to completely customize the environment

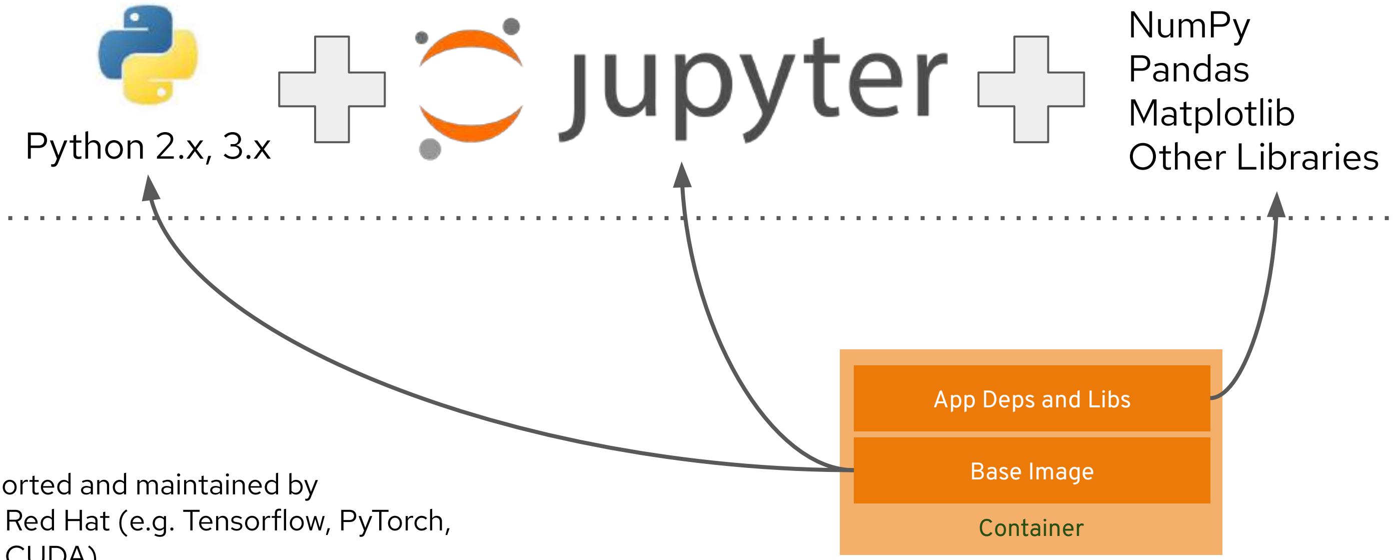


Import a custom notebook image from an image location along with recommended accelerators.

Customizing Workbenches

Base Notebook Images

Reproducible and shareable environments for building, training and serving



Python 2.x, 3.x

jupyter

NumPy
Pandas
Matplotlib
Other Libraries

App Deps and Libs

Base Image

Container

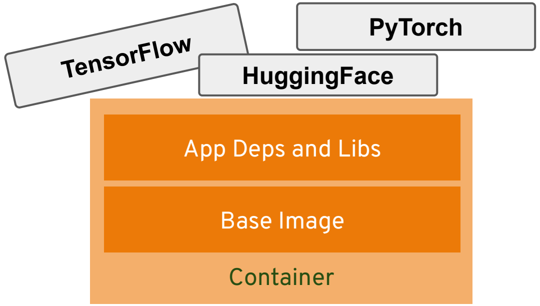
Supported and maintained by

- Red Hat (e.g. Tensorflow, PyTorch, CUDA)
- partner (Anaconda, Intel)
- you (custom notebooks)

Customizing Workbenches

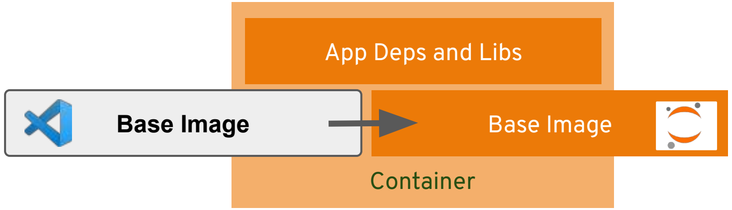
Customizing the workbench

Adding packages on top of a good image



Just remember that they are removed when restarting the workbench*

Creating your own custom image with all dependencies you need



You can now version and maintain it according to your preferences

* This is on purpose so that you can un-mess-up your environment easily if you get into dependency issues.

Model Registry Preview (Coming later in 2024)

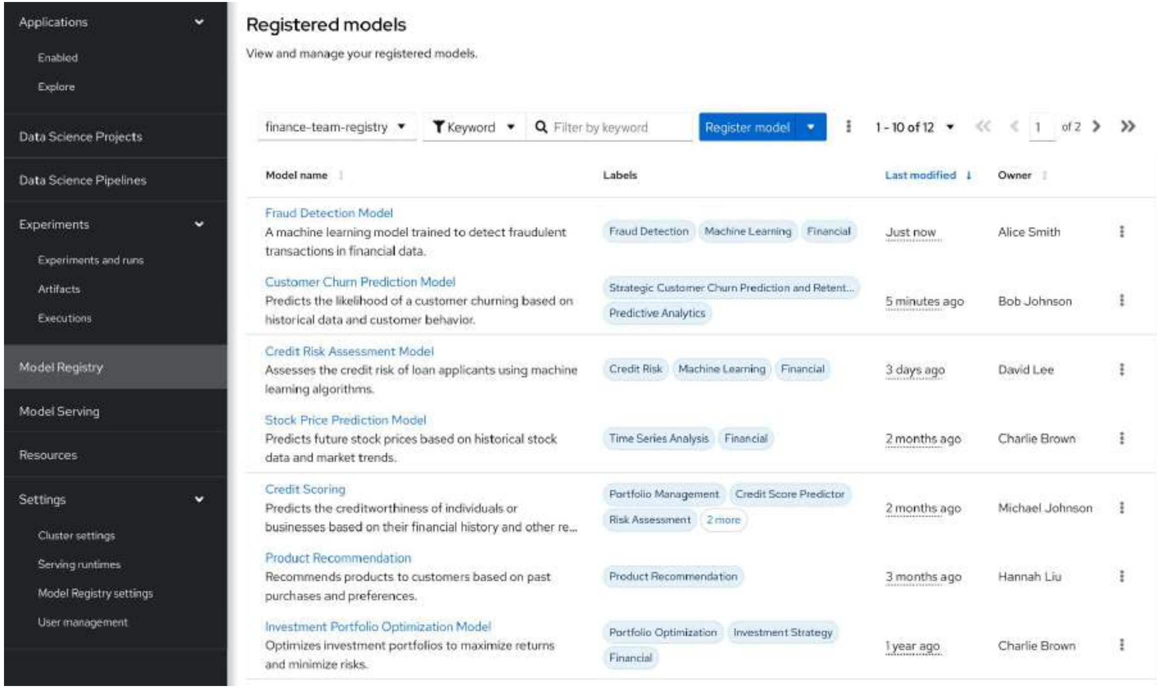
Model Registry Preview

How will it work?

- Can register a model along with properties such as name, tags, description, model type, dataset etc.
- Can edit the details of the model.
- Uses S3 as a default backend but can link to models in other storages as well, for example separate S3 or PVC.
- Can store artifacts such as generated files, sample data, text files, etc.

Model Registry Preview

List models



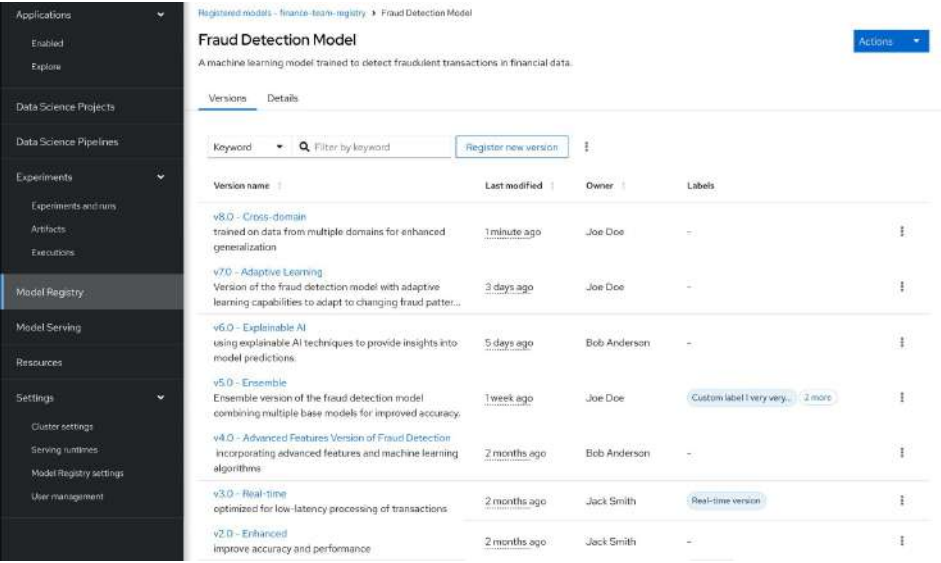
Registered models

View and manage your registered models.

Model name	Labels	Last modified	Owner
Fraud Detection Model A machine learning model trained to detect fraudulent transactions in financial data.	Fraud Detection Machine Learning Financial	Just now	Alice Smith
Customer Churn Prediction Model Predicts the likelihood of a customer churning based on historical data and customer behavior.	Strategic Customer Churn Prediction and Retent... Predictive Analytics	5 minutes ago	Bob Johnson
Credit Risk Assessment Model Assesses the credit risk of loan applicants using machine learning algorithms.	Credit Risk Machine Learning Financial	3 days ago	David Lee
Stock Price Prediction Model Predicts future stock prices based on historical stock data and market trends.	Time Series Analysis Financial	2 months ago	Charlie Brown
Credit Scoring Predicts the creditworthiness of individuals or businesses based on their financial history and other re...	Portfolio Management Credit Score Predictor Risk Assessment 2 more	2 months ago	Michael Johnson
Product Recommendation Recommends products to customers based on past purchases and preferences.	Product Recommendation	3 months ago	Hannah Liu
Investment Portfolio Optimization Model Optimizes investment portfolios to maximize returns and minimize risks.	Portfolio Optimization Investment Strategy Financial	1 year ago	Charlie Brown

Model Registry Preview

Model details and versions

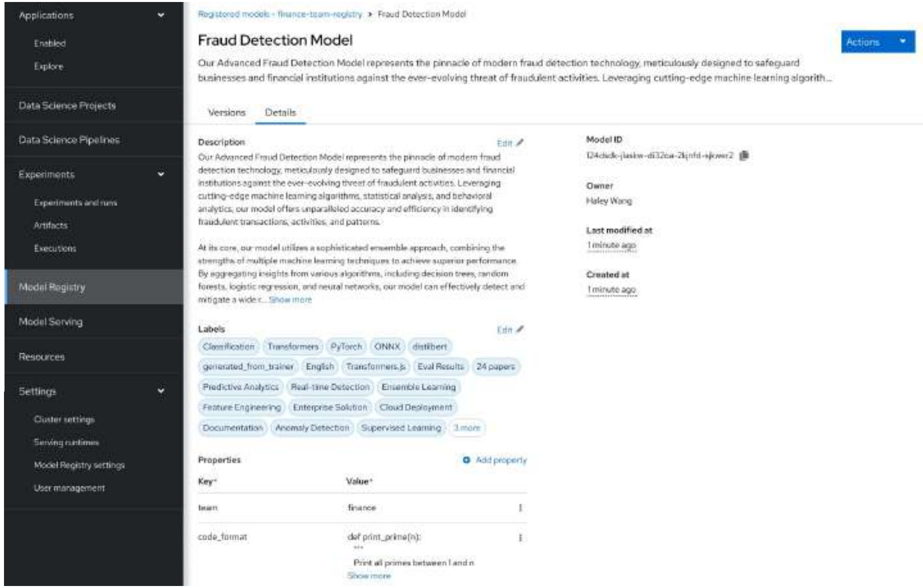


Registered models - finance-team-registry › Fraud Detection Model

Fraud Detection Model

A machine learning model trained to detect fraudulent transactions in financial data.

Version name	Last modified	Owner	Labels
v8.0 - Cross-domain trained on data from multiple domains for enhanced generalization	1 minute ago	Joe Doe	-
v7.0 - Adaptive Learning Version of the fraud detection model with adaptive learning capabilities to adapt to changing fraud patter...	3 days ago	Joe Doe	-
v6.0 - Explainable AI using explainable AI techniques to provide insights into model predictions	5 days ago	Bob Anderson	-
v5.0 - Ensemble Ensemble version of the fraud detection model combining multiple base models for improved accuracy.	1 week ago	Joe Doe	Custom label I very very... 2 more
v4.0 - Advanced Features Version of Fraud Detection incorporating advanced features and machine learning algorithms	2 months ago	Bob Anderson	-
v3.0 - Real-time optimized for low-latency processing of transactions	2 months ago	Jack Smith	Real-time version
v2.0 - Enhanced improve accuracy and performance	2 months ago	Jack Smith	-



Registered models - finance-team-registry › Fraud Detection Model

Fraud Detection Model

Our Advanced Fraud Detection Model represents the pinnacle of modern fraud detection technology, meticulously designed to safeguard businesses and financial institutions against the ever-evolving threat of fraudulent activities. Leveraging cutting-edge machine learning algorith...

Description

Our Advanced Fraud Detection Model represents the pinnacle of modern fraud detection technology, meticulously designed to safeguard businesses and financial institutions against the ever-evolving threat of fraudulent activities. Leveraging cutting-edge machine learning algorithms, statistical analysis, and behavioral analytics, our model offers unparalleled accuracy and efficiency in identifying fraudulent transactions, activities, and patterns.

At its core, our model utilizes a sophisticated ensemble approach, combining the strengths of multiple machine learning techniques to achieve superior performance. By aggregating insights from various algorithms, including decision trees, random forests, logistic regression, and neural networks, our model can effectively detect and mitigate a wide r...

Labels

Classification · Transformers · PyTorch · ONNX · distilbert · generated_from_trainer · English · Transformers.js · Eval Results · 24 papers · Predictive Analytics · Real-time Detection · Ensemble Learning · Feature Engineering · Enterprise Solution · Cloud Deployment · Documentation · Anomaly Detection · Supervised Learning · 3 more

Properties

Key*	Value*
team	finance
code_format	def print_prime(n): """ Print all primes between 1 and n

Model ID

124dsdk-jlaskw-di32ca-2kjnfd-sjkwer2

Owner

Haley Wang

Last modified at

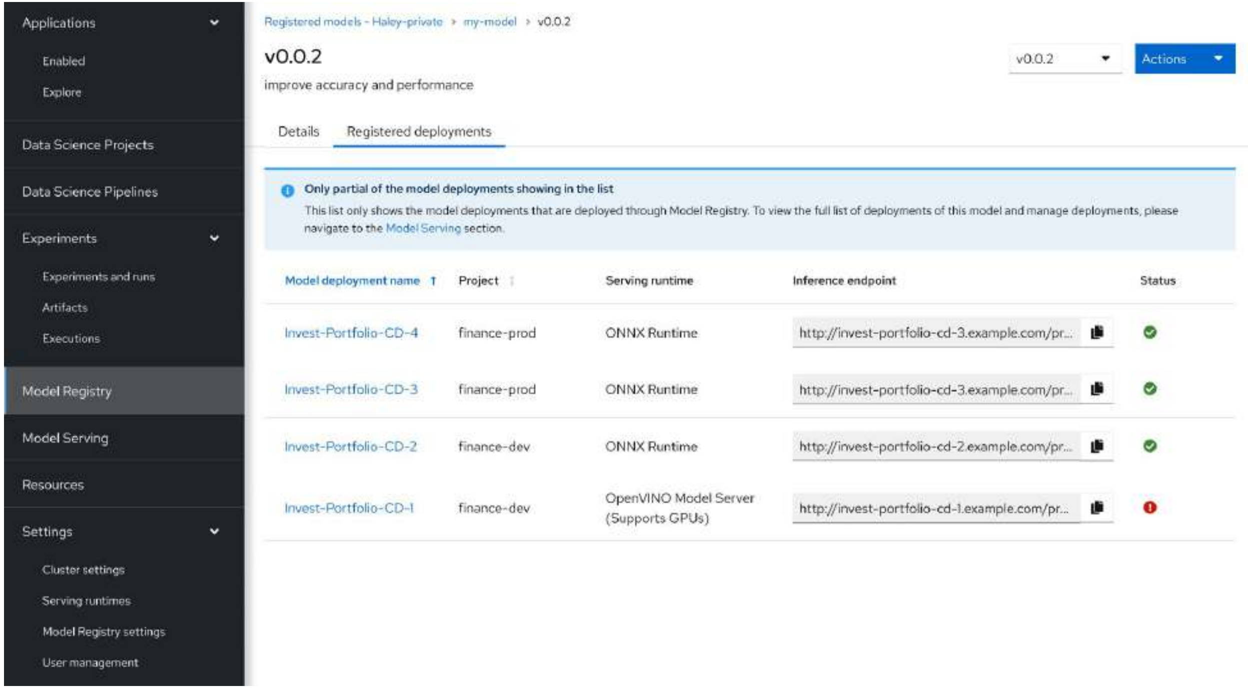
1 minute ago

Created at

1 minute ago

Model Registry Preview

Deploy and keep track



v0.0.2

improve accuracy and performance

Only partial of the model deployments showing in the list

This list only shows the model deployments that are deployed through Model Registry. To view the full list of deployments of this model and manage deployments, please navigate to the Model Serving section.

Model deployment name	Project	Serving runtime	Inference endpoint	Status
Invest-Portfolio-CD-4	finance-prod	ONNX Runtime	http://invest-portfolio-cd-3.example.com/pr...	✓
Invest-Portfolio-CD-3	finance-prod	ONNX Runtime	http://invest-portfolio-cd-3.example.com/pr...	✓
Invest-Portfolio-CD-2	finance-dev	ONNX Runtime	http://invest-portfolio-cd-2.example.com/pr...	✓
Invest-Portfolio-CD-1	finance-dev	OpenVINO Model Server (Supports GPUs)	http://invest-portfolio-cd-1.example.com/pr...	❗

Model Serving

Model Serving Runtimes

Single model serving	
Serving Runtime	Model frameworks supported
OpenVino Model Server	ONNX OpenVino IR TensorFlow
Caikit	Caikit
Text Generation Inference Server (TGIS)	PyTorch
vLLM ServingRuntime for KServe	vLLM

Multi-model serving	
Serving Runtime	Model frameworks supported
OpenVino Model Server	ONNX OpenVino Intermediate Representation (IR) TensorFlow

Users can also create **custom runtimes**

Monitoring

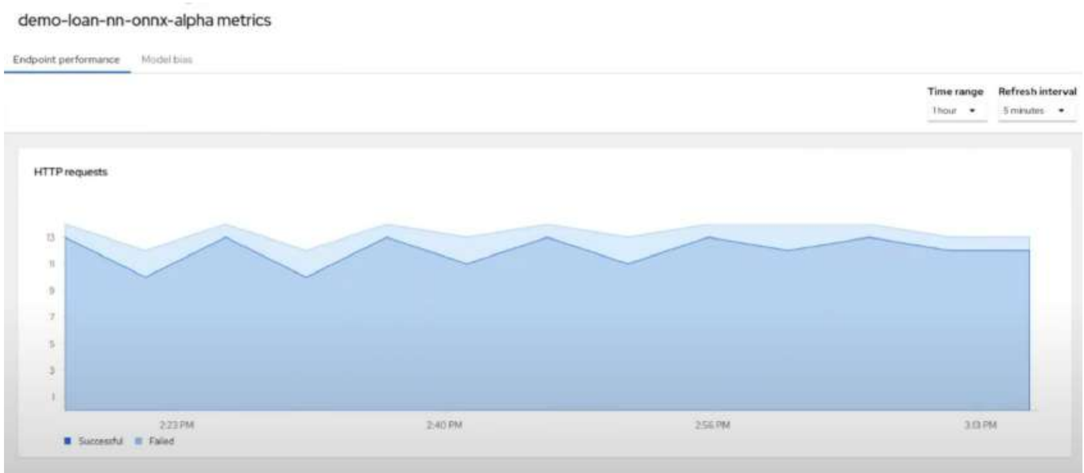
Monitoring Model Performance

In OpenShift AI, you can monitor the following metrics for all the models that are deployed on a model server:

- **HTTP requests**
  - The number of HTTP requests that have failed or succeeded for all models on the server.
- **Average response time (ms)**
  - For all models on the server, the average time it takes the model server to respond to requests.
- **CPU utilization (%)**
  - The percentage of the CPU's capacity that is currently being used by all models on the server.
- **Memory utilization (%)**
  - The percentage of the system's memory that is currently being used by all models on the server.

Monitoring in RHOAI

You can get to monitoring by clicking on a served model, either in Data Science Project or in the Model Serving page.



demo-loan-nn-onnx-alpha metrics

Endpoint performance | Model bias

Time range: 1 hour

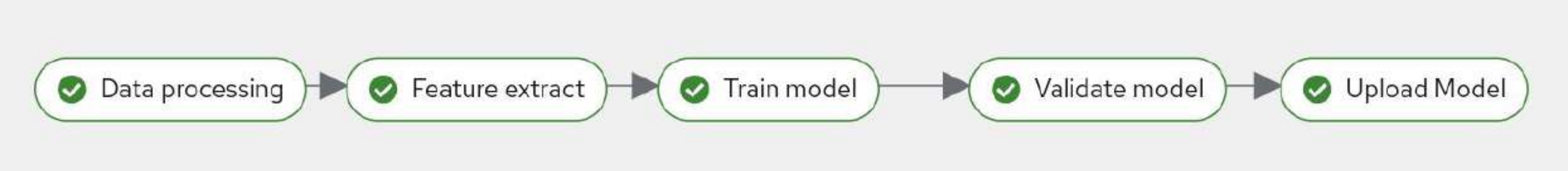
Refresh interval: 5 minutes

DS Pipelines

Data Science Pipelines

Data Science Pipelines

- Portable ML workflows to automate end-to-end ML tasks.
- Enables continuous integration and deployment of machine learning operations in staging and production.
- Based on Kubeflow pipelines. This internally leverages Argo Workflows to run the ML workflows.
- Example:
  - Here is a sample workflow that automates the ML tasks of processing data, extracting features from the data, train the ml model, validate it and upload the model to s3 object store.



Data Science Pipelines

Data Science Pipelines

- Users can have **one pipeline server per project** and execute multiple pipelines.
- Pipelines uses a **Object Storage** to
  - store artifacts such as logs, data passed between steps, dependency files, and results.
- **Share data** between steps through:
  - Through parameters (small data)
  - Through volumes (large data)
  - Object storage (large data)
- **Experiment tracking**
  - Pipeline runs can be used as experiments, and the run view can be used to track those experiments.

Data Science Pipelines

Components

- **Pipeline Server**
  - A server that is attached to your data science project and hosts your data science pipeline.
  - Requires S3-compatible data connection to store your pipeline artifacts.
- **Pipeline**
  - A pipeline defines the configuration of your machine learning workflow and the relationship between each component in the workflow.
    - Pipeline code: A definition of your pipeline in a Tekton-formatted YAML file.
    - Pipeline graph (using Elyra GUI): A graphical illustration of the steps executed in a pipeline run and the relationship between them.
- **Pipeline run**: An execution of your pipeline.
  - Triggered run: A previously executed pipeline run.
  - Scheduled run: A pipeline run scheduled to execute at least once.

Data Science Pipelines

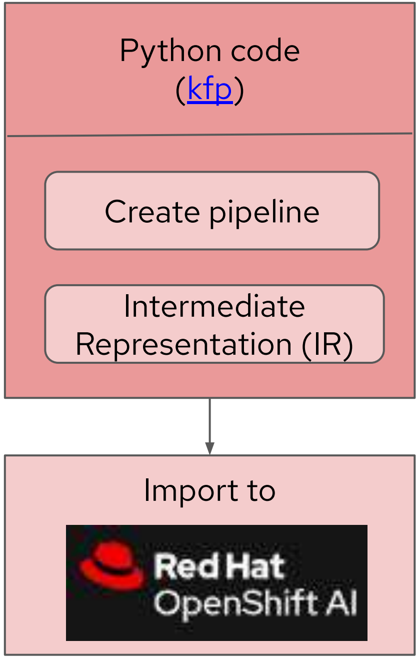
Defining a Pipeline

1. Using Kubeflow Pipelines SDK

ml_train_upload.py

```
import kfp
from kfp.components import create_component_from_func

get_data_component = create_component_from_func(
                              get_data,
                              base_image=...",
                              packages_to_install=[ ]
)
@kfp.dsl.pipeline(name="train_upload_stock_kfp")
def sdk_pipeline():
          get_data_task = get_data_component()
          ...
from kfp_tekton.compiler import TektonCompiler
...
TektonCompiler().compile(sdk_pipeline, __file__.replace(".py", ".yaml"))
```



Python code
(kfp)

Create pipeline

Intermediate Representation (IR)

Import to

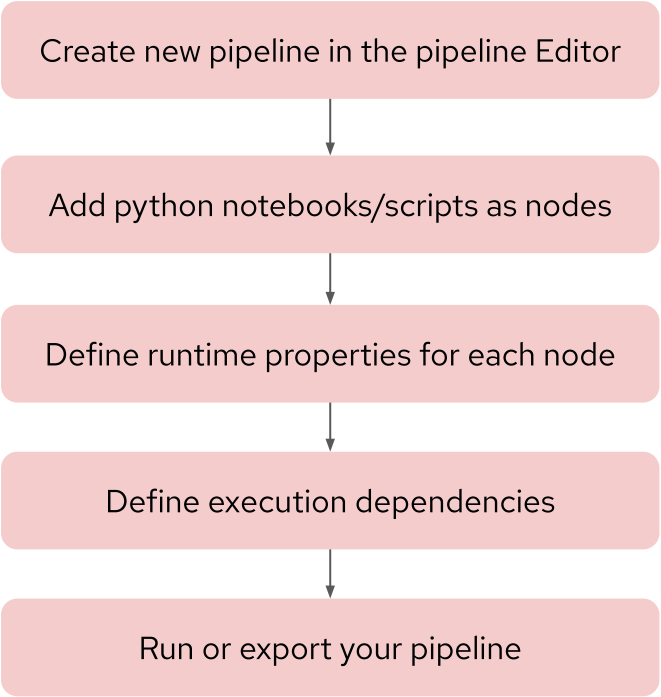
Red Hat OpenShift AI

Defining a Pipeline

2. Using Elyra JupyterLab Extension

jupyter

Pipeline Editor



Create new pipeline in the pipeline Editor

Add python notebooks/scripts as nodes

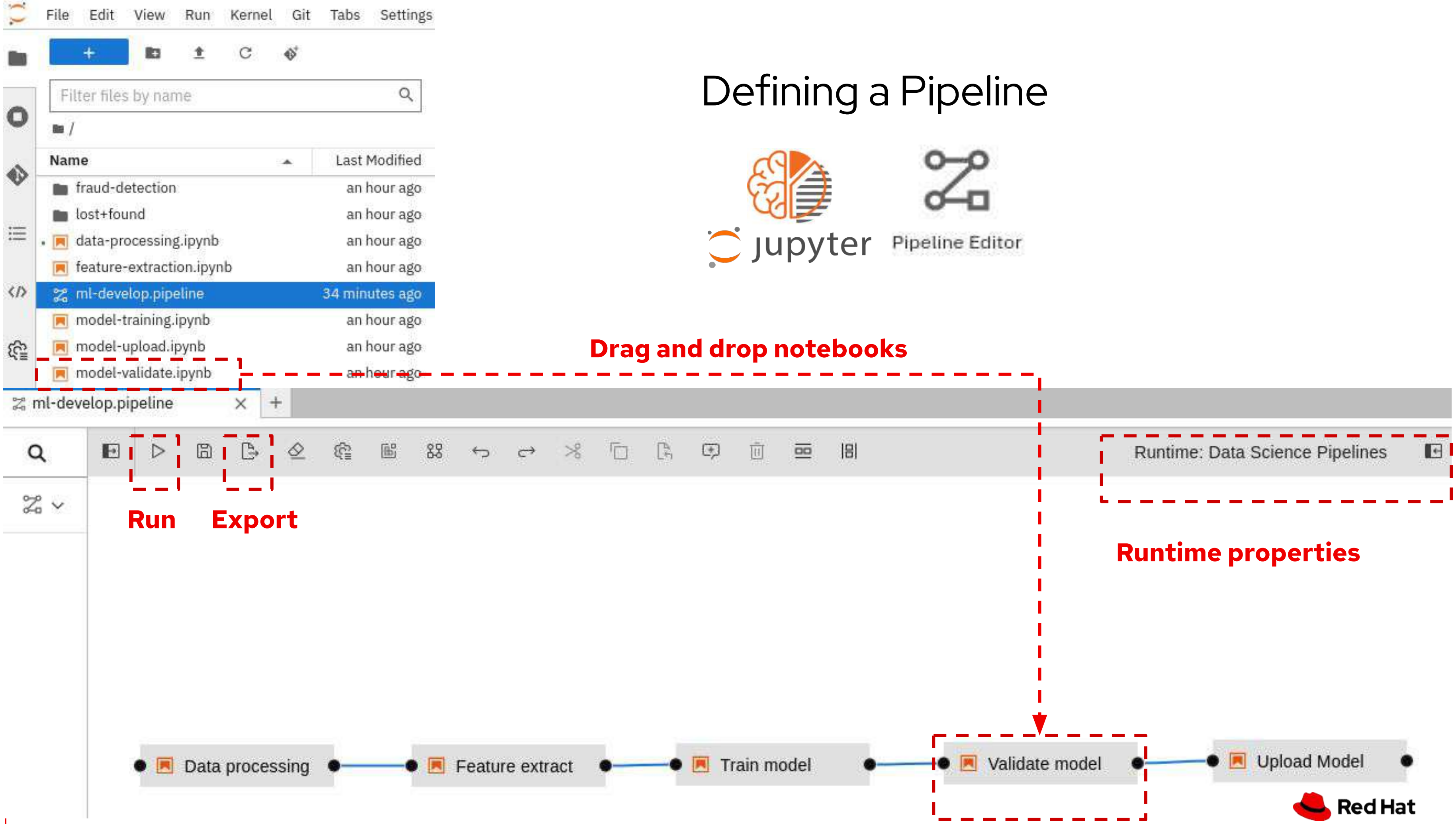
Define runtime properties for each node

Define execution dependencies

Run or export your pipeline

Defining a Pipeline

Pipeline Editor



Name	Last Modified
fraud-detection	an hour ago
lost+found	an hour ago
data-processing.ipynb	an hour ago
feature-extraction.ipynb	an hour ago
ml-develop.pipeline	34 minutes ago
model-training.ipynb	an hour ago
model-upload.ipynb	an hour ago
model-validate.ipynb	an hour ago

Drag and drop notebooks

Run

Export

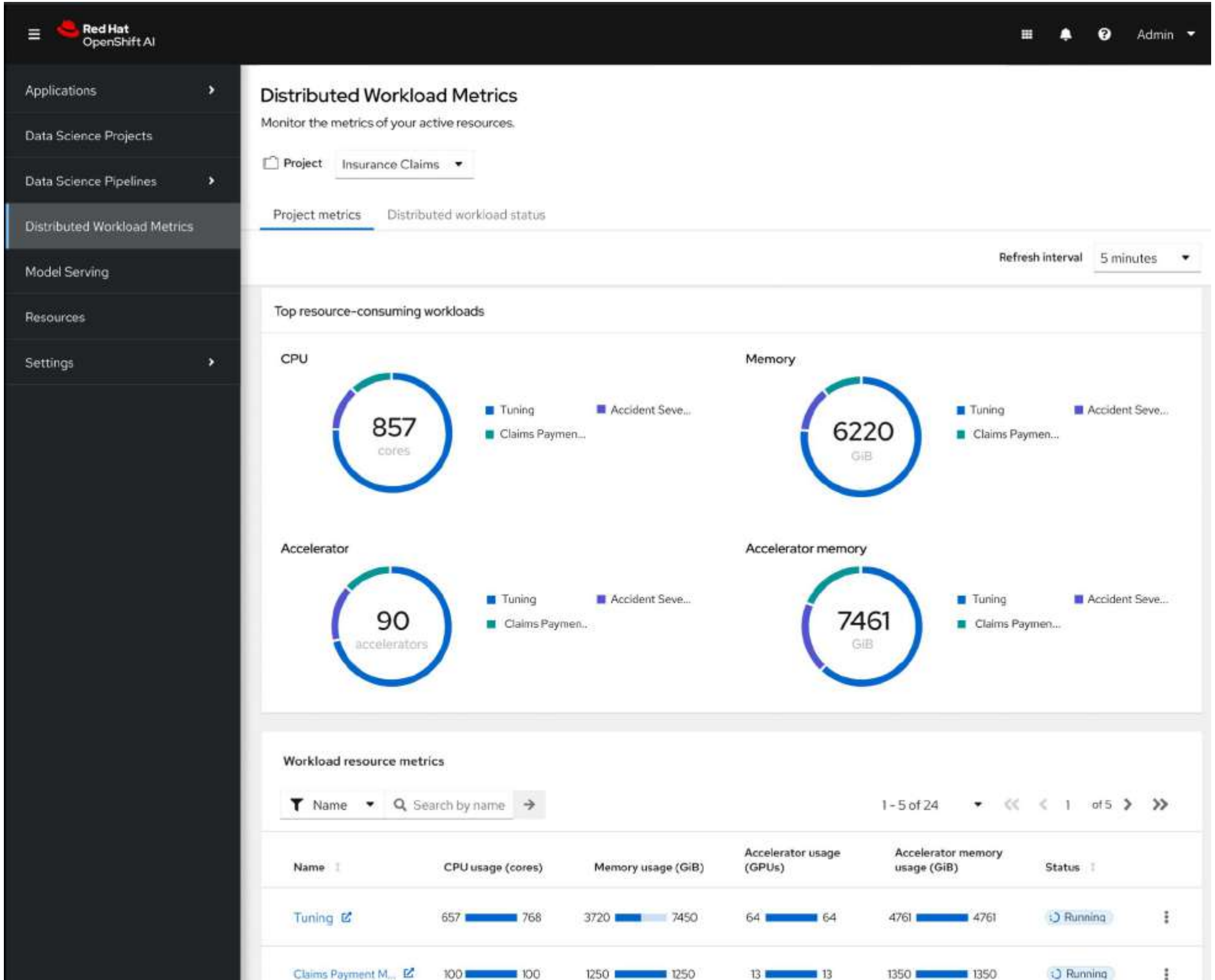
Runtime properties

Distributed workloads

Distributed training overview

- Distributed training is used to distribute a larger job across multiple nodes, for example fine-tuning an LLM when a single node does not have enough GPUs.
- With the CodeFlare component in RHOAI, you can spin up Ray clusters inside your OpenShift cluster.
- You can then submit jobs to these Ray clusters, where the jobs will be distributed across a selected amount of nodes you have available.
- This also gives you access to the Ray dashboard, helping you keep track of the jobs and their logs.

Distributed Workloads



Distributed Workload Metrics

Monitor the metrics of your active resources.

Project: Insurance Claims

Project metrics | Distributed workload status

Refresh interval: 5 minutes

Top resource-consuming workloads

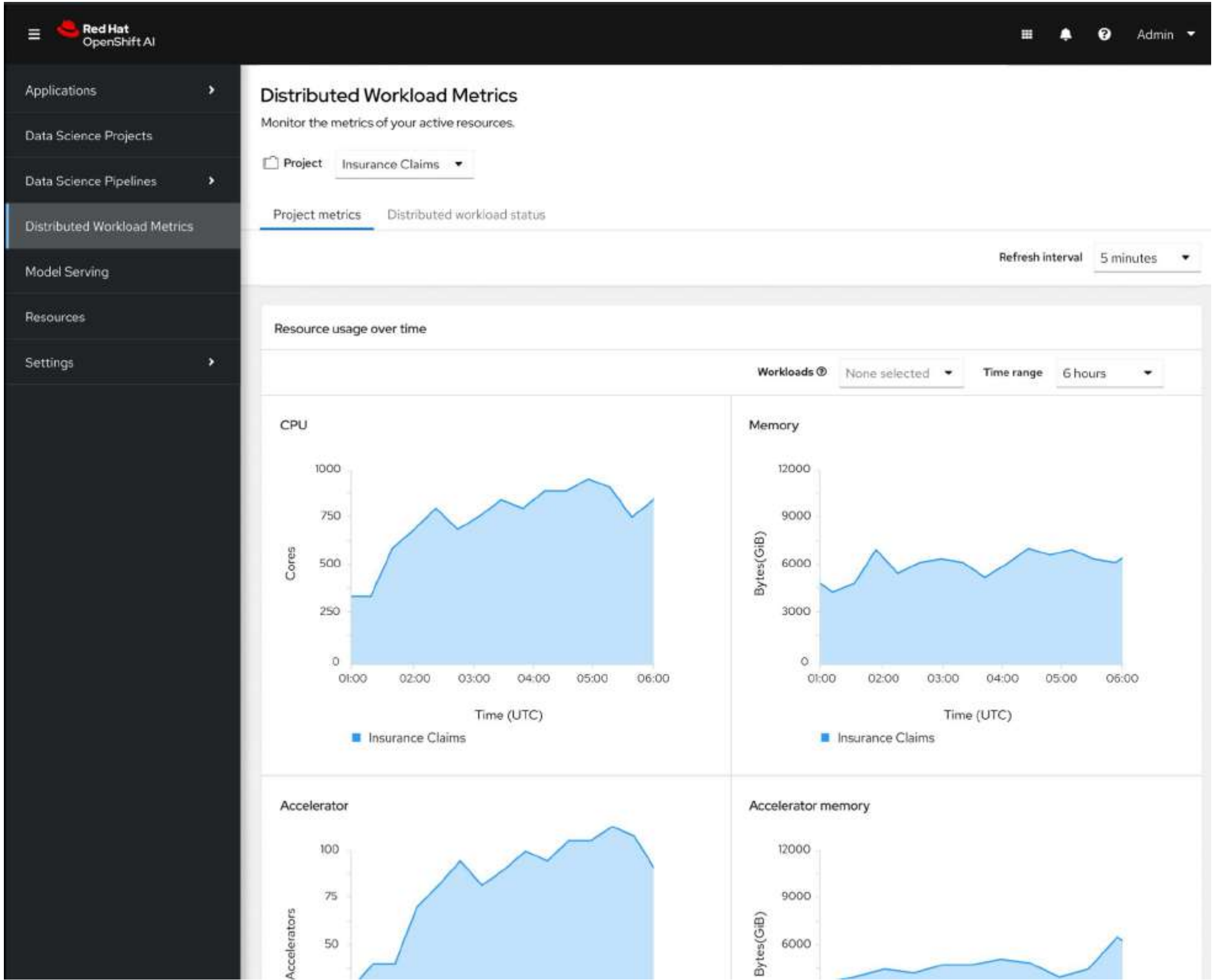
Workload resource metrics

Name | Search by name

1 - 5 of 24 | 1 of 5

Name	CPU usage (cores)	Memory usage (GiB)	Accelerator usage (GPUs)	Accelerator memory usage (GiB)	Status
Tuning	657 768	3720 7450	64 64	4761 4761	Running
Claims Payment M...	100 100	1250 1250	13 13	1350 1350	Running

Distributed Workloads



Distributed Workload Metrics

Monitor the metrics of your active resources.

Project Insurance Claims

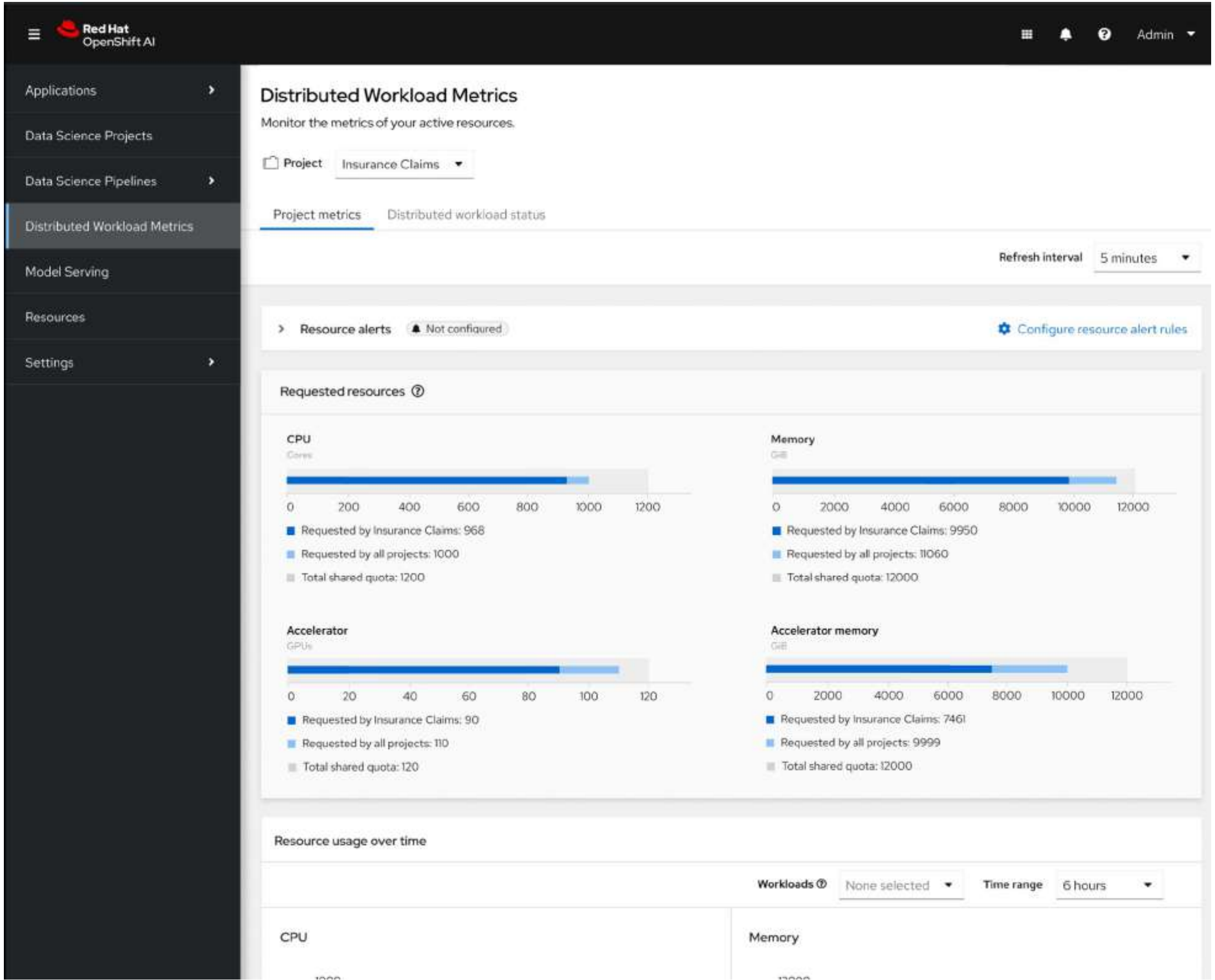
Refresh interval 5 minutes

Resource usage over time

Workloads None selected

Time range 6 hours

Distributed Workloads



Distributed Workload Metrics

Monitor the metrics of your active resources.

Project: Insurance Claims

Refresh interval: 5 minutes

Resource alerts: Not configured

Requested resources

CPU

Cores

- Requested by Insurance Claims: 968
- Requested by all projects: 1000
- Total shared quota: 1200

Memory

GiB

- Requested by Insurance Claims: 9950
- Requested by all projects: 11060
- Total shared quota: 12000

Accelerator

GPUs

- Requested by Insurance Claims: 90
- Requested by all projects: 110
- Total shared quota: 120

Accelerator memory

GiB

- Requested by Insurance Claims: 7461
- Requested by all projects: 9999
- Total shared quota: 12000

Resource usage over time

Workloads: None selected

Time range: 6 hours

CPU

Memory

Platform Admins

Flavors of RHOAI

Flavors of RHOAI

Supported deployment options		
Options available	**Self-managed RHOAI**	**Cloud Service RHOAI**
Bare metal	✓	
Virtual	✓	
Private cloud	✓	
Red Hat OpenShift on AWS (ROSA)	✓	✓
Azure Red Hat OpenShift (ARO)	✓	(future)
IBM Cloud	✓	
OSD-GCP/OSD-AWS	✓	✓
Edge	(future)	

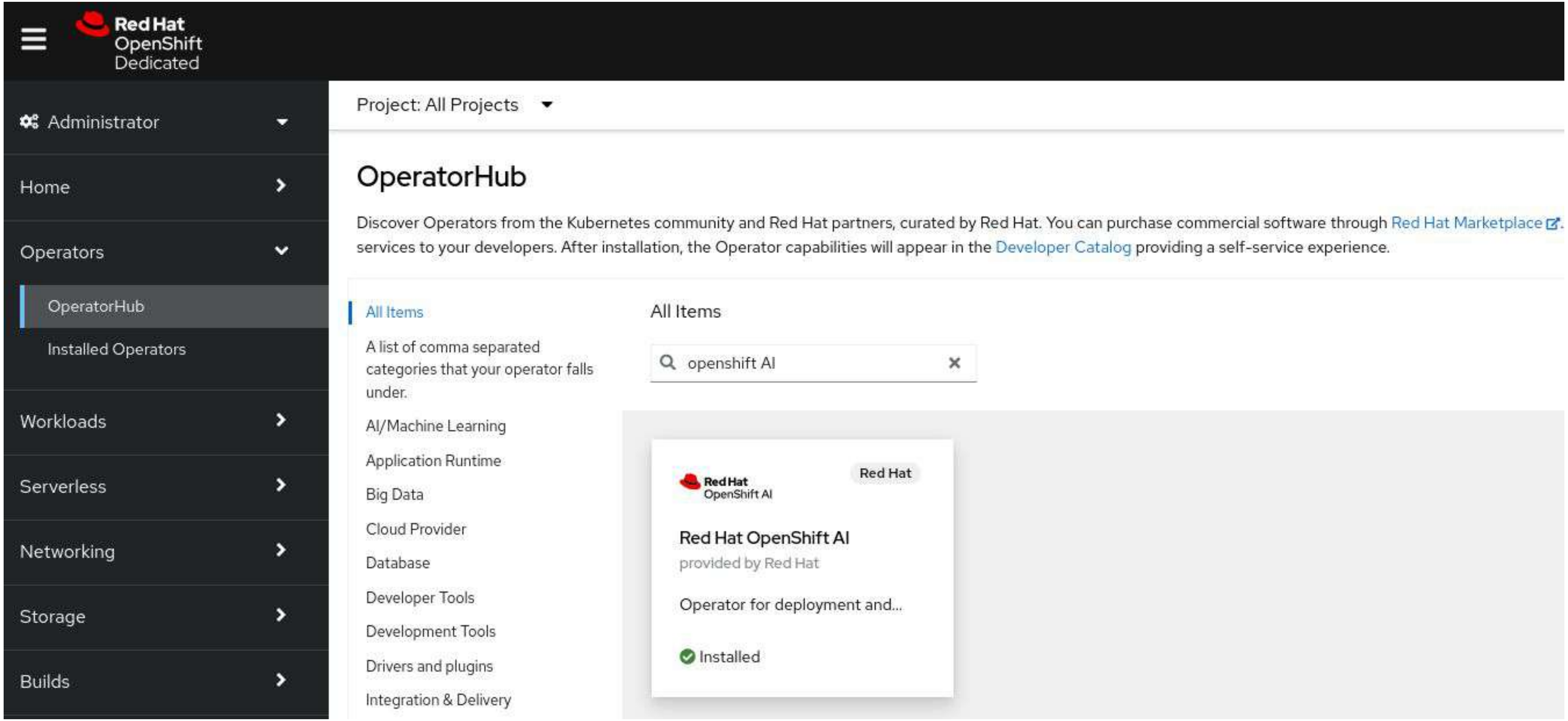
Flavors of RHOAI

Disconnected

- RHOAI can be installed on disconnected clusters.
- When installed disconnected, everything you need to run RHOAI and its default components are installed with it.
- For everything outside the default components, such as custom runtimes, notebook images, or Python dependencies, you will need to manually bring it into the cluster for it to work.
- For more details on how to install disconnected, refer to the documentation.

Install/Upgrades/Support

Install RHOAI



OperatorHub

Discover Operators from the Kubernetes community and Red Hat partners, curated by Red Hat. You can purchase commercial software through Red Hat Marketplace. services to your developers. After installation, the Operator capabilities will appear in the Developer Catalog providing a self-service experience.

openshift AI

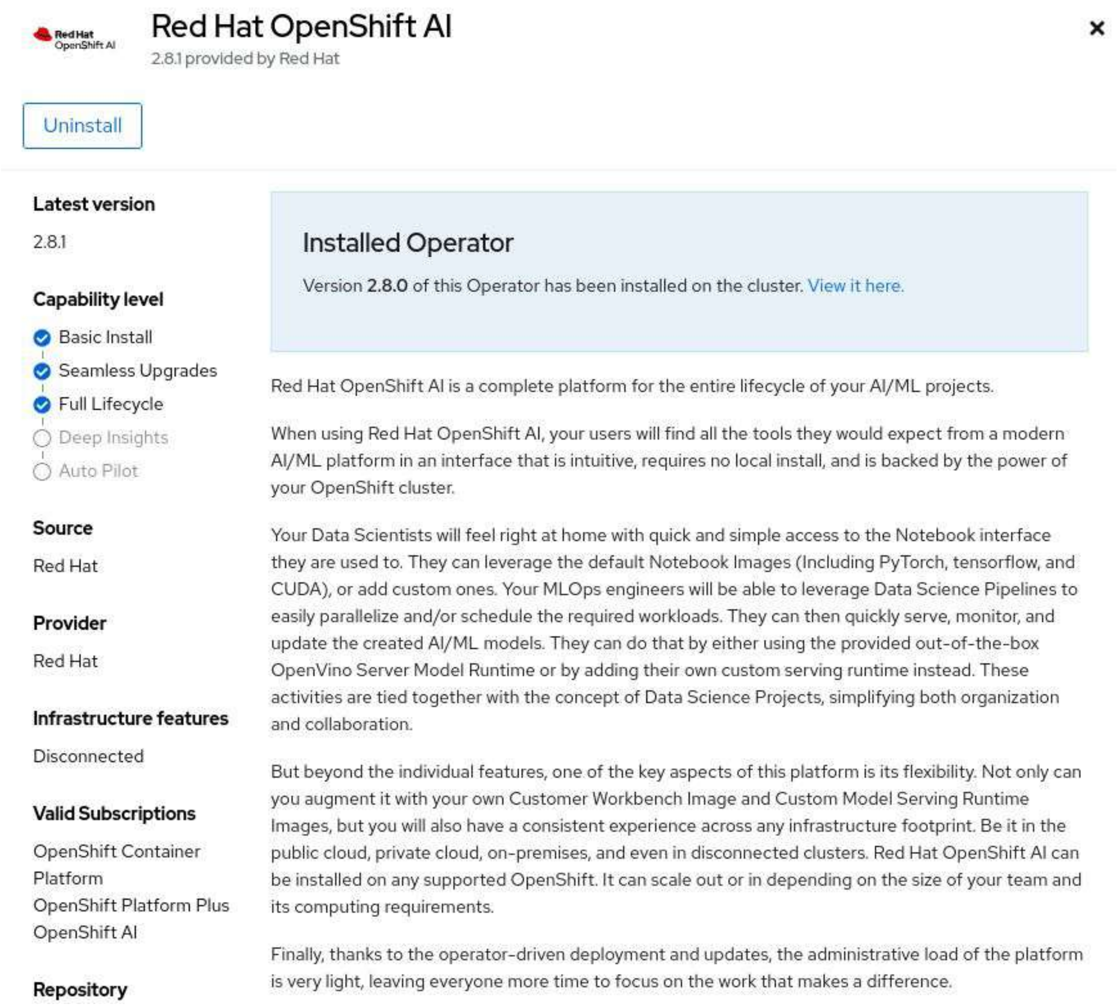
Red Hat OpenShift AI
Red Hat

Red Hat OpenShift AI
provided by Red Hat

Operator for deployment and...

✓ Installed

Install RHOAI



Red Hat OpenShift AI

2.8.1 provided by Red Hat

Uninstall

Latest version

2.8.1

Capability level

- ✓ Basic Install
- ✓ Seamless Upgrades
- ✓ Full Lifecycle
- ○ Deep Insights
- ○ Auto Pilot

Source

Red Hat

Provider

Red Hat

Infrastructure features

Disconnected

Valid Subscriptions

OpenShift Container Platform
OpenShift Platform Plus
OpenShift AI

Repository

Installed Operator

Version **2.8.0** of this Operator has been installed on the cluster. View it here.

Red Hat OpenShift AI is a complete platform for the entire lifecycle of your AI/ML projects.

When using Red Hat OpenShift AI, your users will find all the tools they would expect from a modern AI/ML platform in an interface that is intuitive, requires no local install, and is backed by the power of your OpenShift cluster.

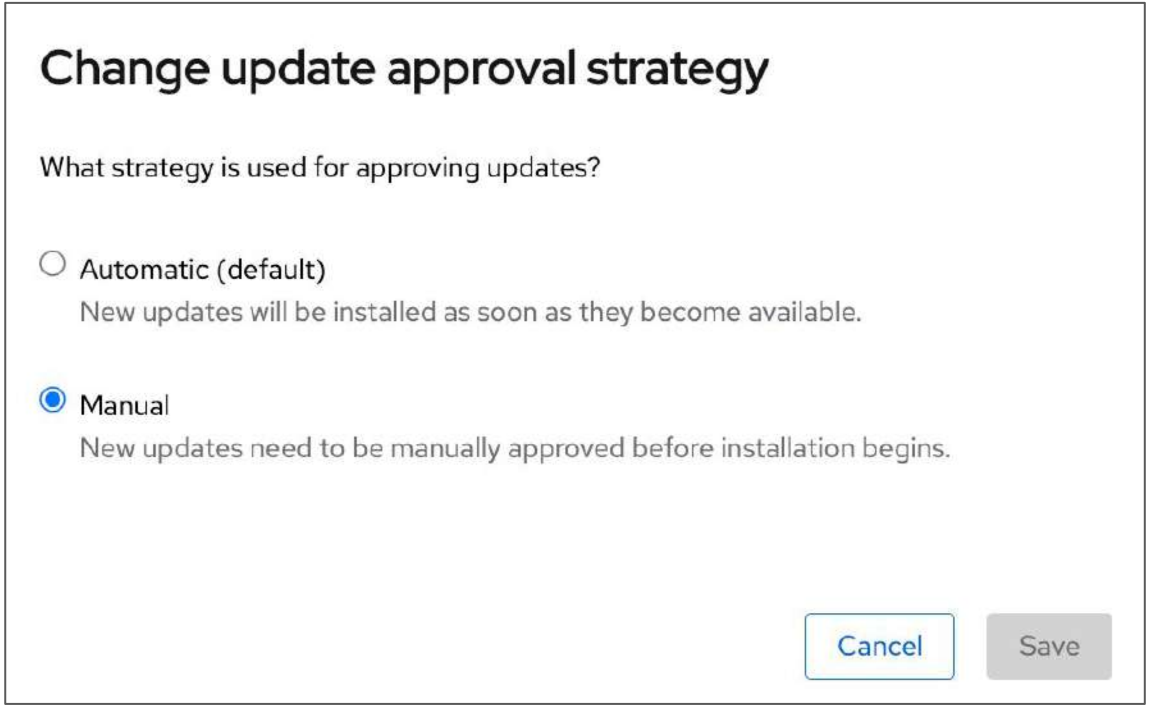
Your Data Scientists will feel right at home with quick and simple access to the Notebook interface they are used to. They can leverage the default Notebook Images (Including PyTorch, tensorflow, and CUDA), or add custom ones. Your MLOps engineers will be able to leverage Data Science Pipelines to easily parallelize and/or schedule the required workloads. They can then quickly serve, monitor, and update the created AI/ML models. They can do that by either using the provided out-of-the-box OpenVino Server Model Runtime or by adding their own custom serving runtime instead. These activities are tied together with the concept of Data Science Projects, simplifying both organization and collaboration.

But beyond the individual features, one of the key aspects of this platform is its flexibility. Not only can you augment it with your own Customer Workbench Image and Custom Model Serving Runtime Images, but you will also have a consistent experience across any infrastructure footprint. Be it in the public cloud, private cloud, on-premises, and even in disconnected clusters. Red Hat OpenShift AI can be installed on any supported OpenShift. It can scale out or in depending on the size of your team and its computing requirements.

Finally, thanks to the operator-driven deployment and updates, the administrative load of the platform is very light, leaving everyone more time to focus on the work that makes a difference.

Update RHOAI

Automatic vs Manual



Change update approval strategy

What strategy is used for approving updates?

○ Automatic (default)
New updates will be installed as soon as they become available.

● Manual
New updates need to be manually approved before installation begins.

Support

There are three release types:

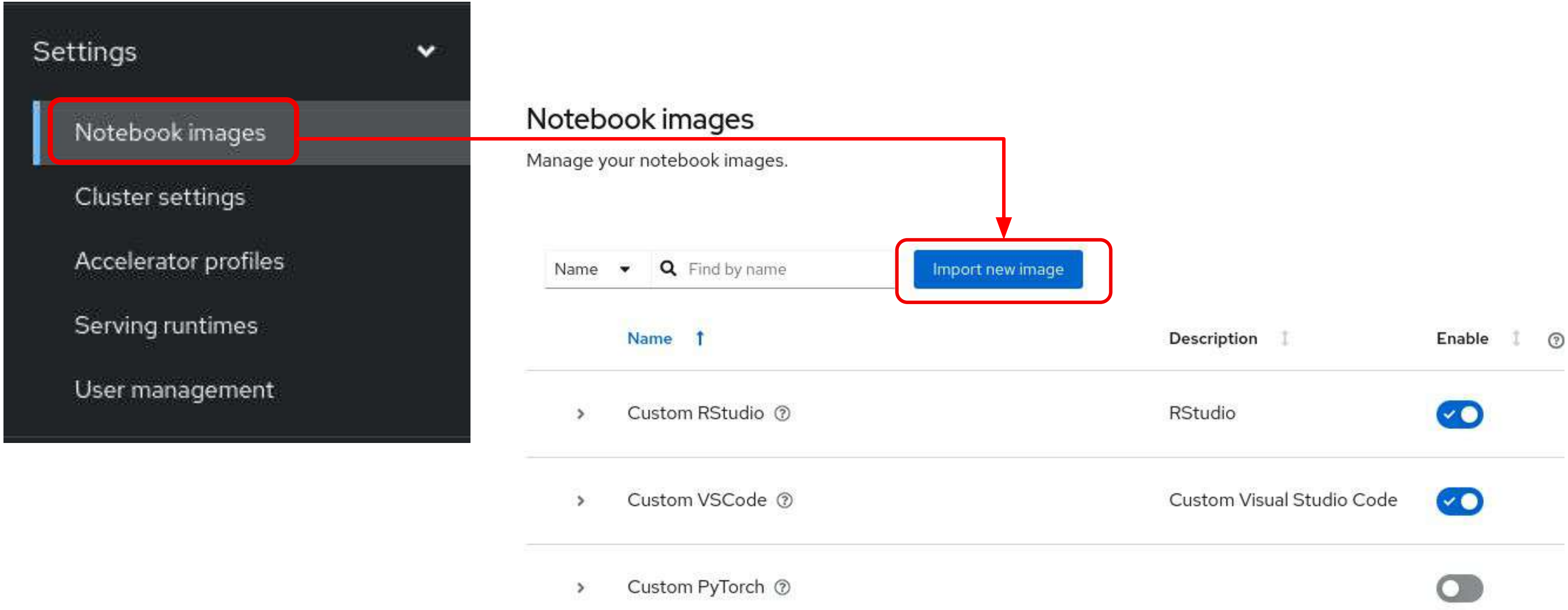
- **Fast** - Includes full support for a month, or until the next fast release is available. This is for customers who want the latest and greatest features, just beware that the fast update rate may not always be desirable.
- **Stable** - Includes full support for seven months. One stable release is released every 3rd fast release.
  This is for customers who want stability and to update according to their own schedule while still being supported.
- **Extended Update Support (EUS)** - Includes full support for seven months followed by Extended Update Support for eleven months. Red Hat issues a EUS release every nine minor releases.

You can see the versions and more details in this documentation.

Customize Workbenches and Serving Runtimes

Custom Notebook Image

Import new image



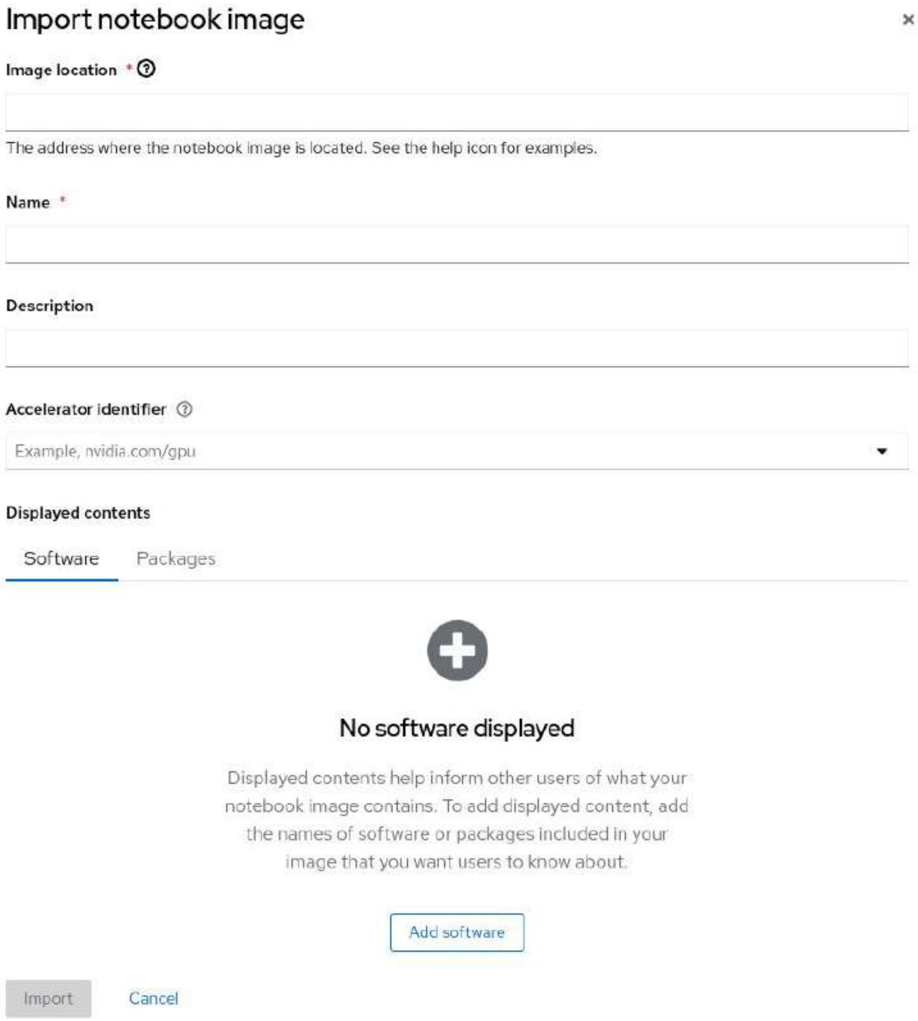
Notebook images

Manage your notebook images.

	Name	Description	Enable
›	Custom RStudio	RStudio	✓
›	Custom VSCode	Custom Visual Studio Code	✓
›	Custom PyTorch		○

Custom Notebook Image

Add a custom notebook image to run custom workbenches by simply providing the image location.



Import notebook image

Image location *

The address where the notebook image is located. See the help icon for examples.

Name *

Description

Accelerator identifier

Example, nvidia.com/gpu

Displayed contents

No software displayed

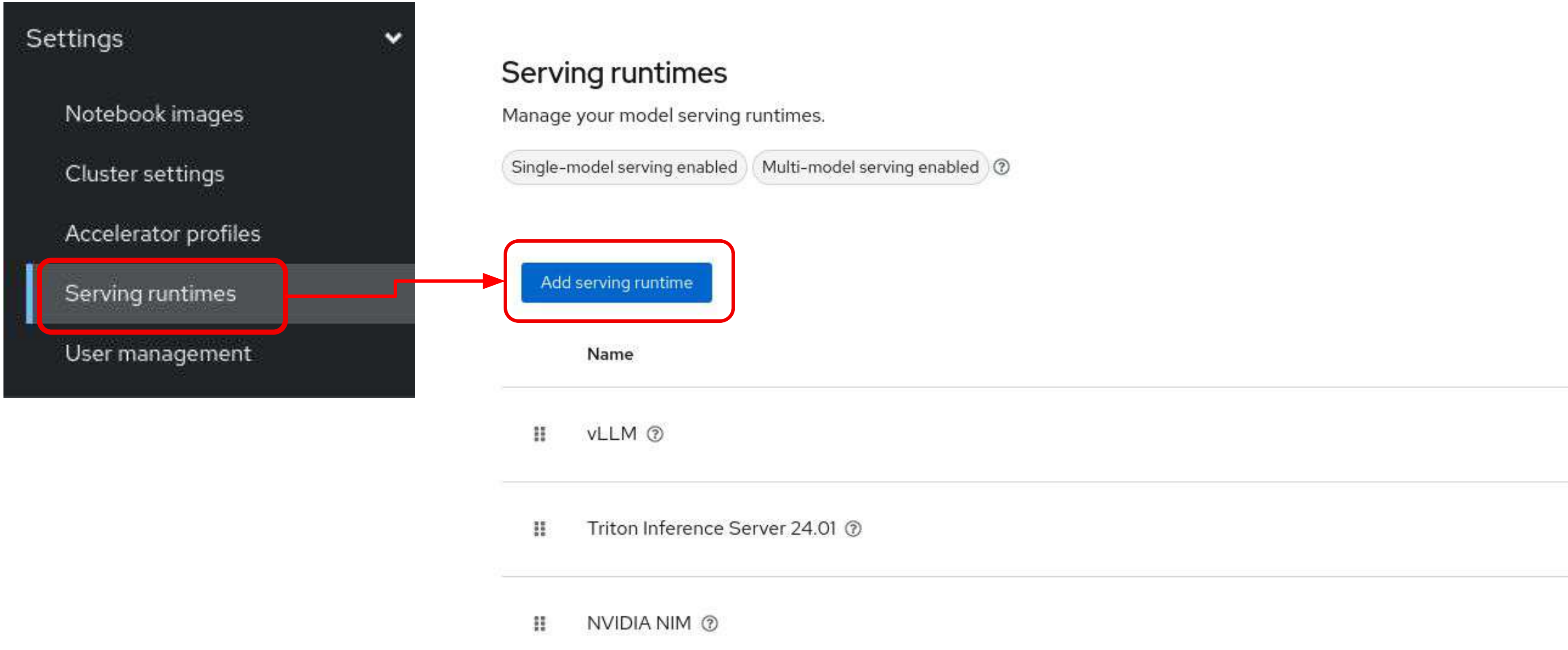
Displayed contents help inform other users of what your notebook image contains. To add displayed content, add the names of software or packages included in your image that you want users to know about.

Example images here: https://github.com/opendatahub-io-contrib/workbench-images

Red Hat

Custom Serving Runtime

Add serving runtime



Serving runtimes

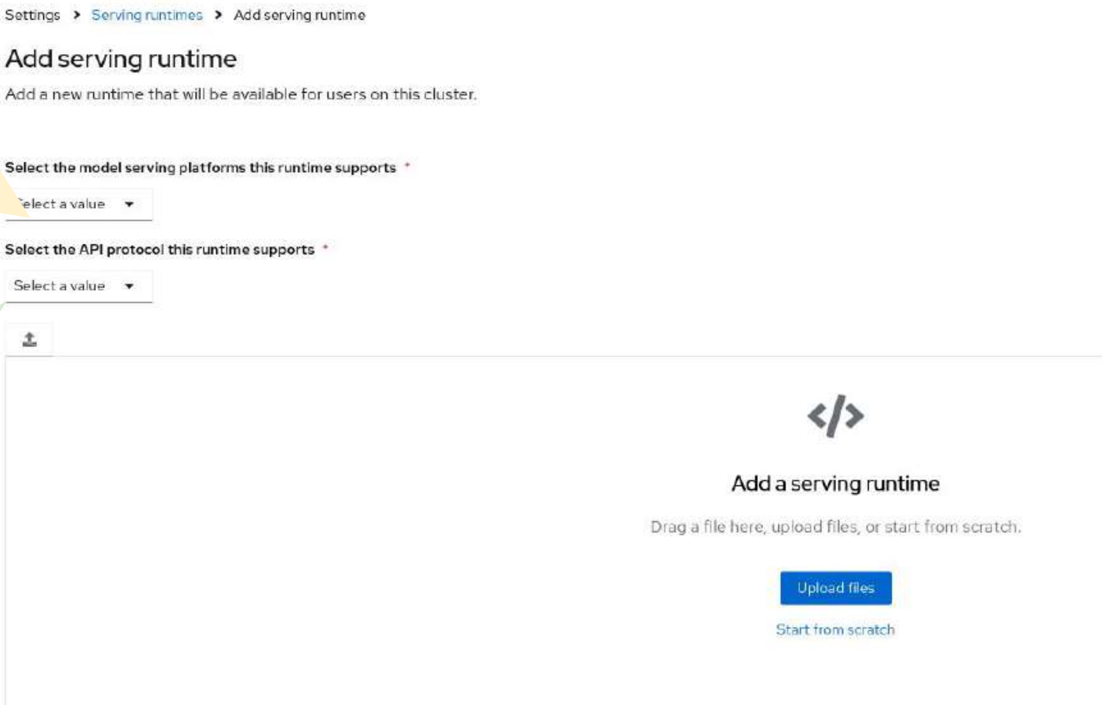
Manage your model serving runtimes.

Name
vLLM
Triton Inference Server 24.01
NVIDIA NIM

Custom Serving Runtime

Single-model serving platform
(or)
Multi-model serving platform

REST
(or)
gRPC



Settings > Serving runtimes > Add serving runtime

Add serving runtime

Add a new runtime that will be available for users on this cluster.

Select the model serving platforms this runtime supports *

Select a value ▾

Select the API protocol this runtime supports *

Select a value ▾

Add a serving runtime

Drag a file here, upload files, or start from scratch.

Documentation and examples:

- Multi model serving
- Single model serving
- Example (custom-runtime-triton)
- Example (VLLM)

Customize RHOAI cluster

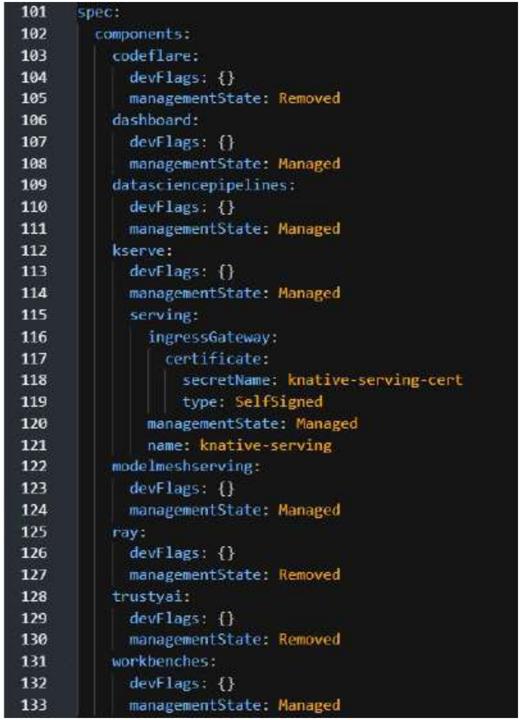
Customize RHOAI Cluster

Enable or disable components

You can enable or disable RHOAI components inside of your DataScienceCluster yaml.

These are the components you can enable/disable:

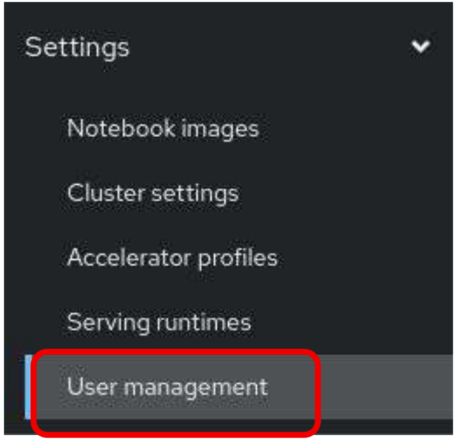
- CodeFlare (for distributed training)
- Dashboard
- Data Science Pipelines
- Kserve (the component for single-model serving)
- Modelmesh serving (the component for multi-model serving)
- Ray (for distributed training)
- TrustyAI
- Workbenches

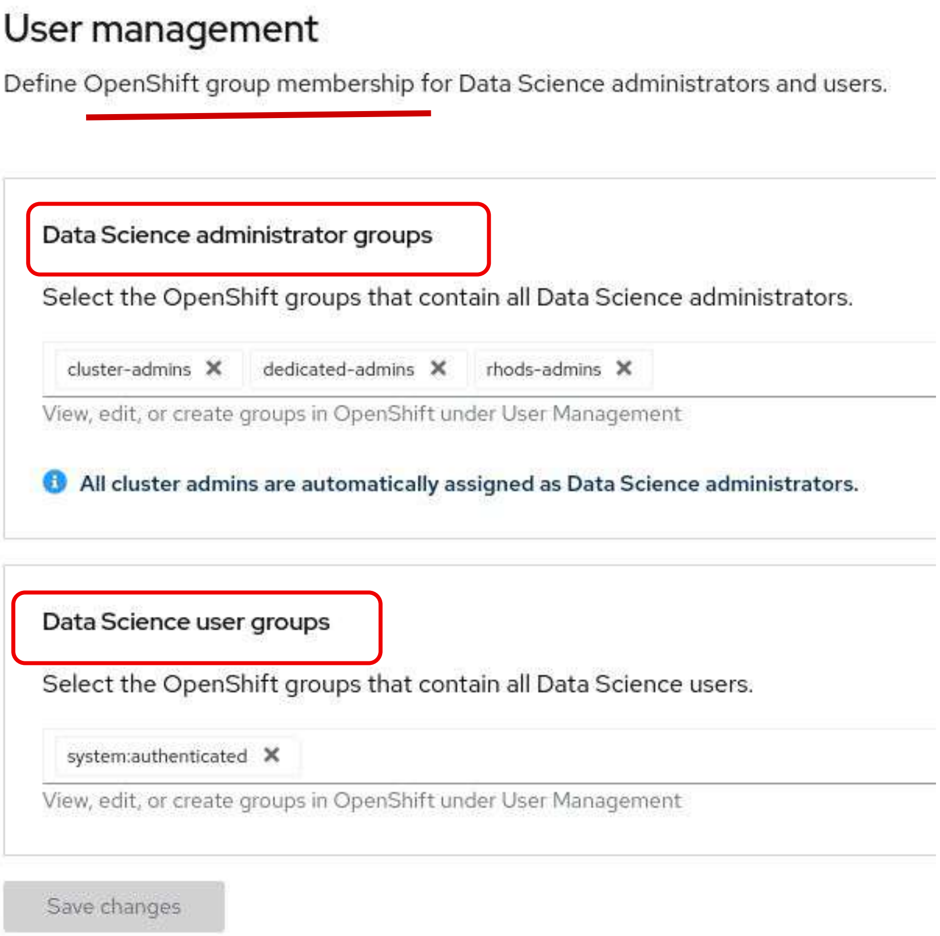


```yaml
spec:
  components:
    codeflare:
      devFlags: {}
      managementState: Removed
    dashboard:
      devFlags: {}
      managementState: Managed
    datasciencepipelines:
      devFlags: {}
      managementState: Managed
    kserve:
      devFlags: {}
      managementState: Managed
      serving:
        ingressGateway:
          certificate:
            secretName: knative-serving-cert
            type: SelfSigned
        managementState: Managed
        name: knative-serving
    modelmeshserving:
      devFlags: {}
      managementState: Managed
    ray:
      devFlags: {}
      managementState: Removed
    trustyai:
      devFlags: {}
      managementState: Removed
    workbenches:
      devFlags: {}
      managementState: Managed
```

User Management

User Management





User management

Define OpenShift group membership for Data Science administrators and users.

Data Science administrator groups

Select the OpenShift groups that contain all Data Science administrators.

cluster-admins ✕ dedicated-admins ✕ rhods-admins ✕

View, edit, or create groups in OpenShift under User Management

ℹ All cluster admins are automatically assigned as Data Science administrators.

Data Science user groups

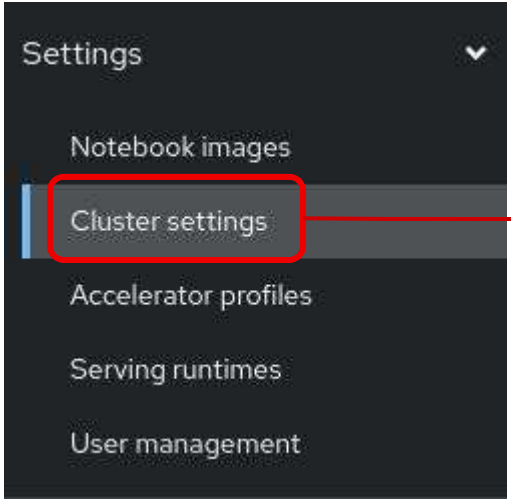
Select the OpenShift groups that contain all Data Science users.

system:authenticated ✕

View, edit, or create groups in OpenShift under User Management

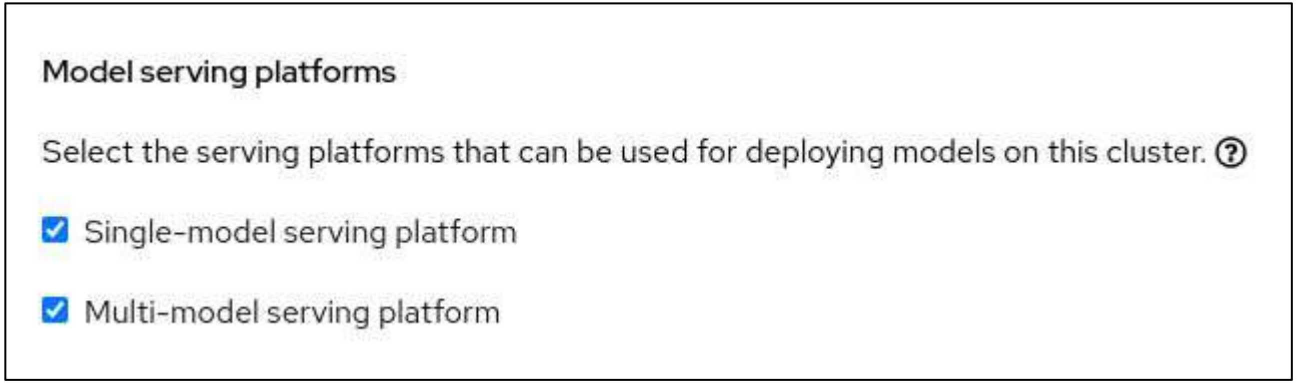
Resource Management

Cluster Settings



1. Model serving platforms
2. PVC size
3. Stop idle notebooks
4. Usage data collection
5. Notebook pod tolerations

Cluster Settings

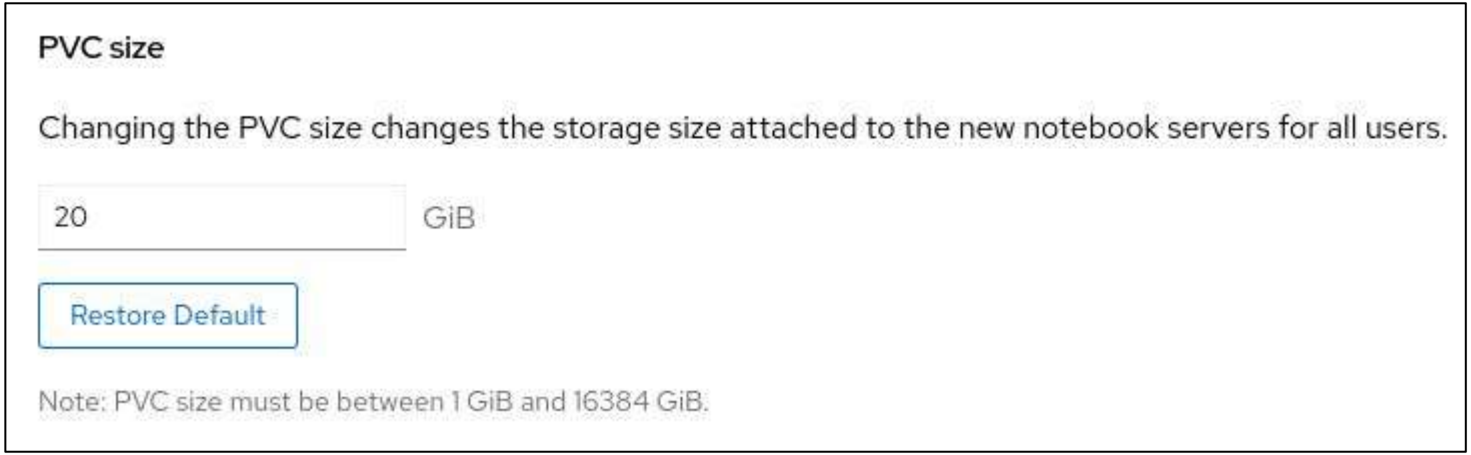


Model serving platforms

Select the serving platforms that can be used for deploying models on this cluster. ?

☑ Single-model serving platform

☑ Multi-model serving platform



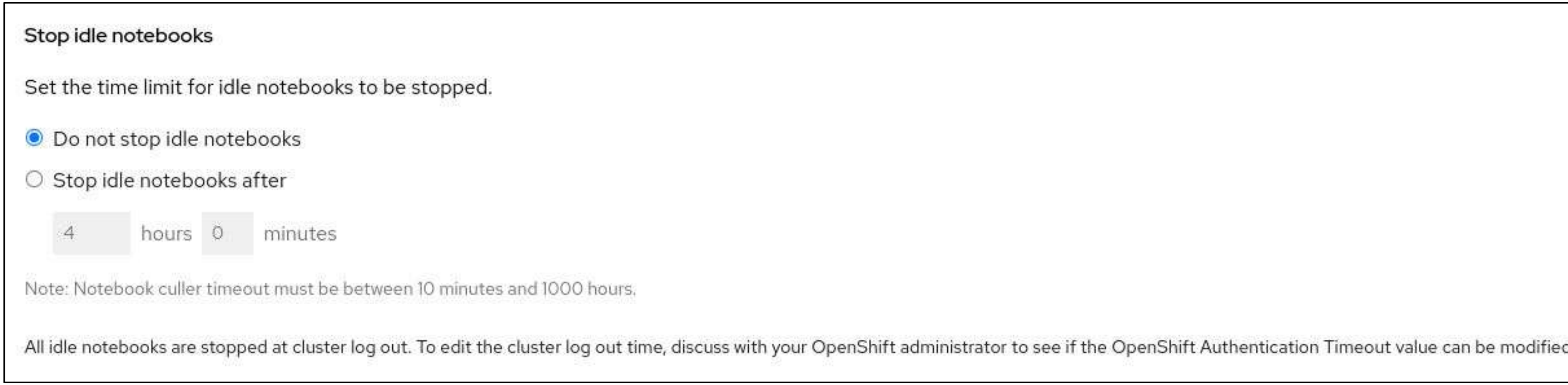
PVC size

Changing the PVC size changes the storage size attached to the new notebook servers for all users.

20 GiB

Note: PVC size must be between 1 GiB and 16384 GiB.

Cluster Settings



Stop idle notebooks

Set the time limit for idle notebooks to be stopped.

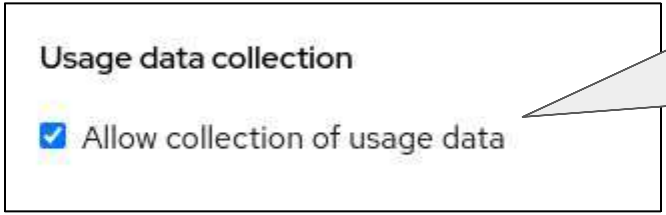
● Do not stop idle notebooks

○ Stop idle notebooks after

4 hours 0 minutes

Note: Notebook culler timeout must be between 10 minutes and 1000 hours.

All idle notebooks are stopped at cluster log out. To edit the cluster log out time, discuss with your OpenShift administrator to see if the OpenShift Authentication Timeout value can be modifie

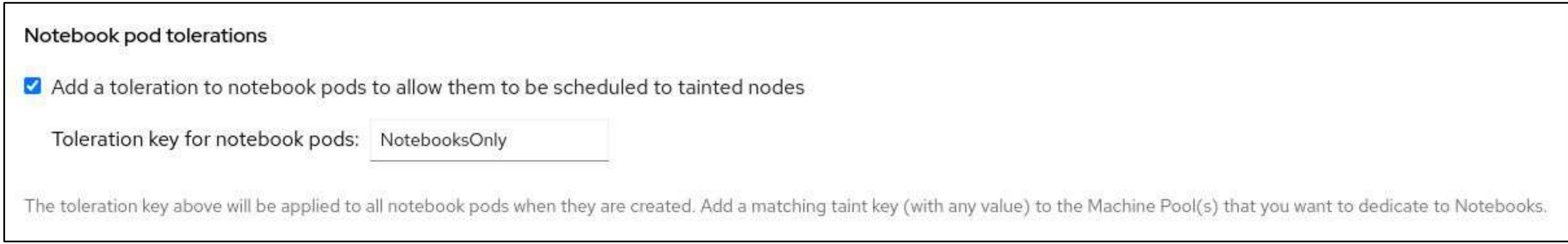


Usage data collection

☑ Allow collection of usage data

- applications enabled in the product dashboard.
- deployment sizes used (CPU/memory resources allocated).
- documentation resources accessed from the product dashboard.
- name of the notebook images
- user identification - unique random identifier per user
- usage information about components, features, and extensions.

Cluster Settings



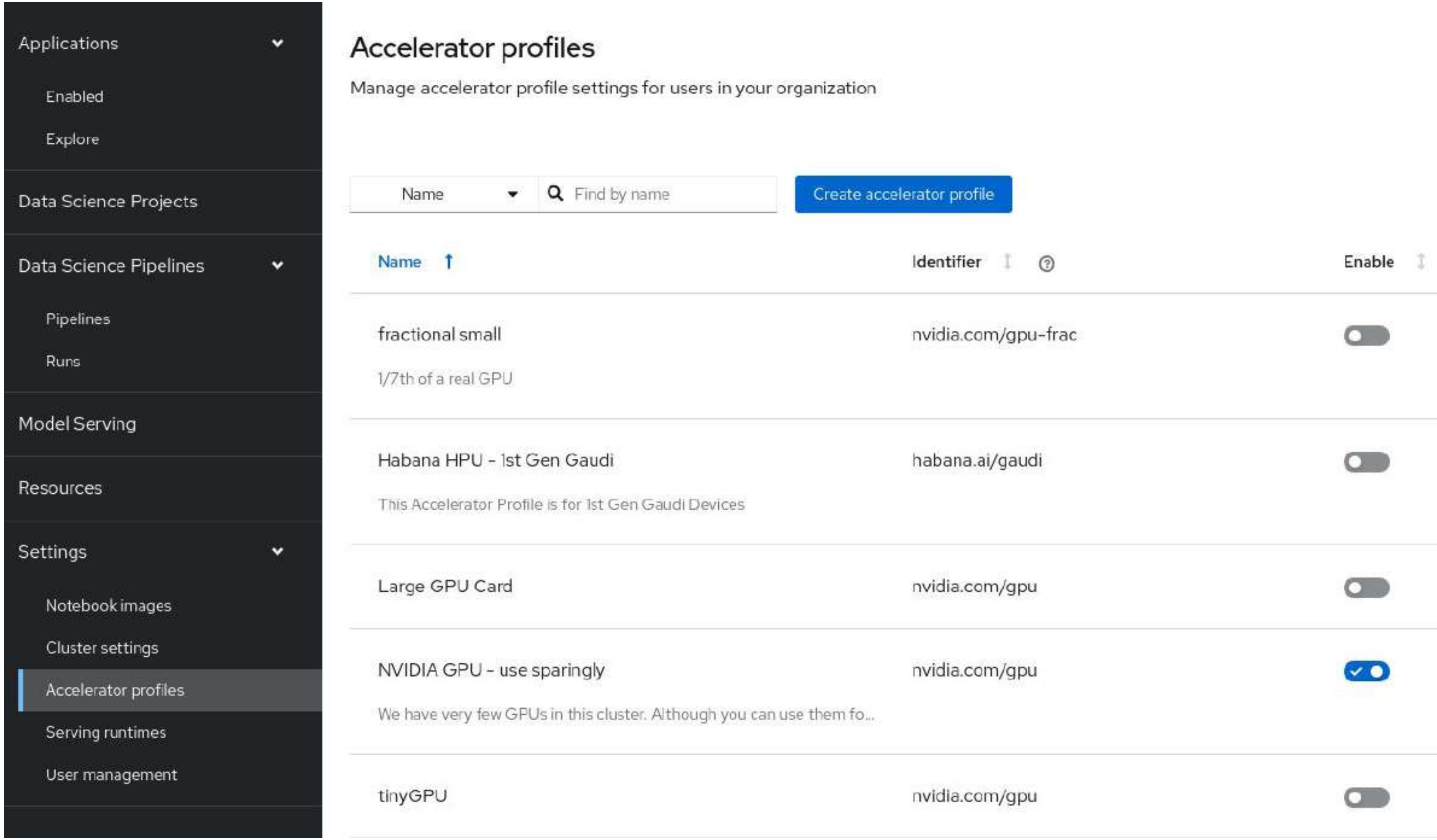
Notebook pod tolerations

☑ Add a toleration to notebook pods to allow them to be scheduled to tainted nodes

Toleration key for notebook pods: NotebooksOnly

The toleration key above will be applied to all notebook pods when they are created. Add a matching taint key (with any value) to the Machine Pool(s) that you want to dedicate to Notebooks.

Accelerator Profile

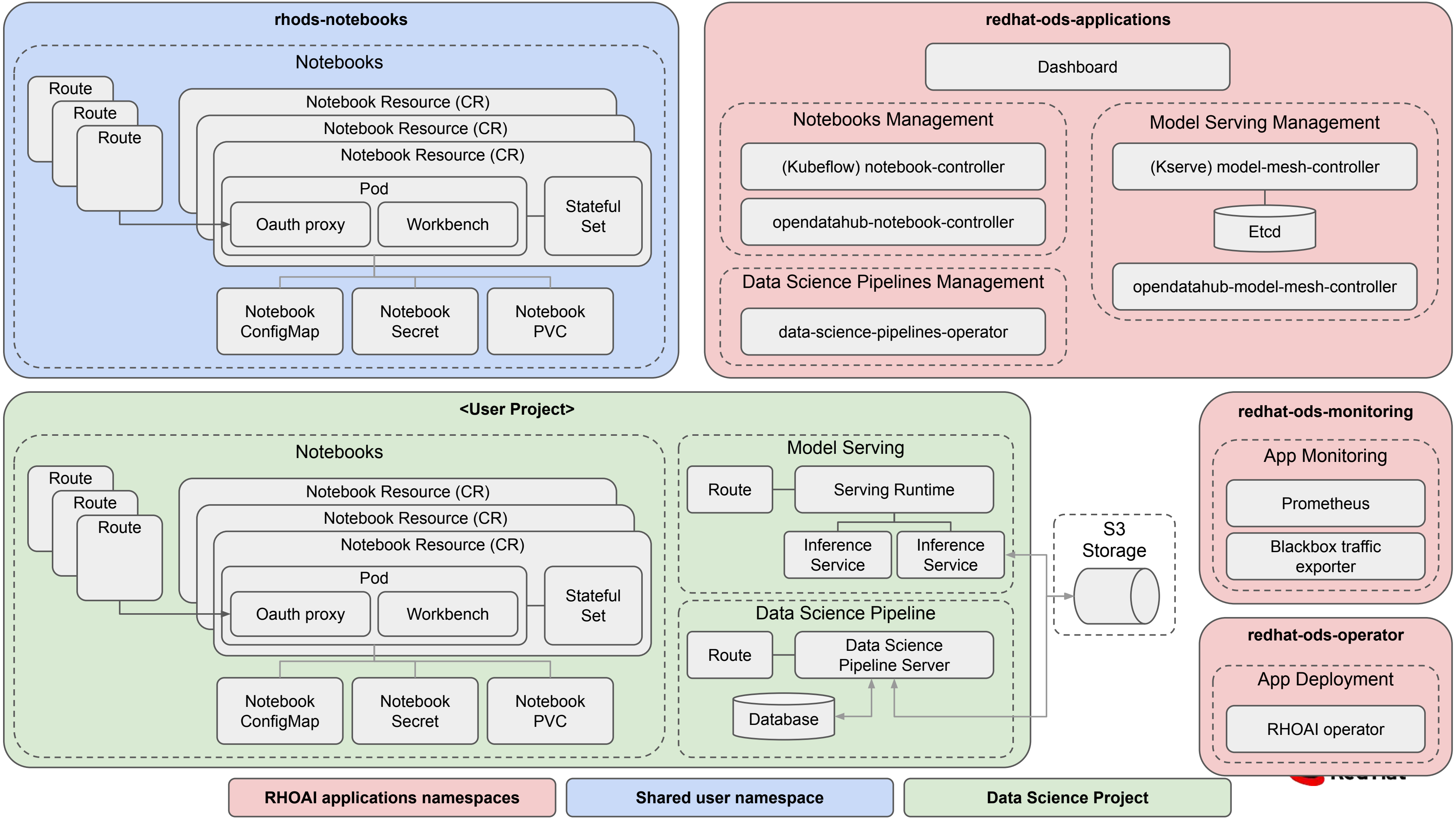


Accelerator profiles

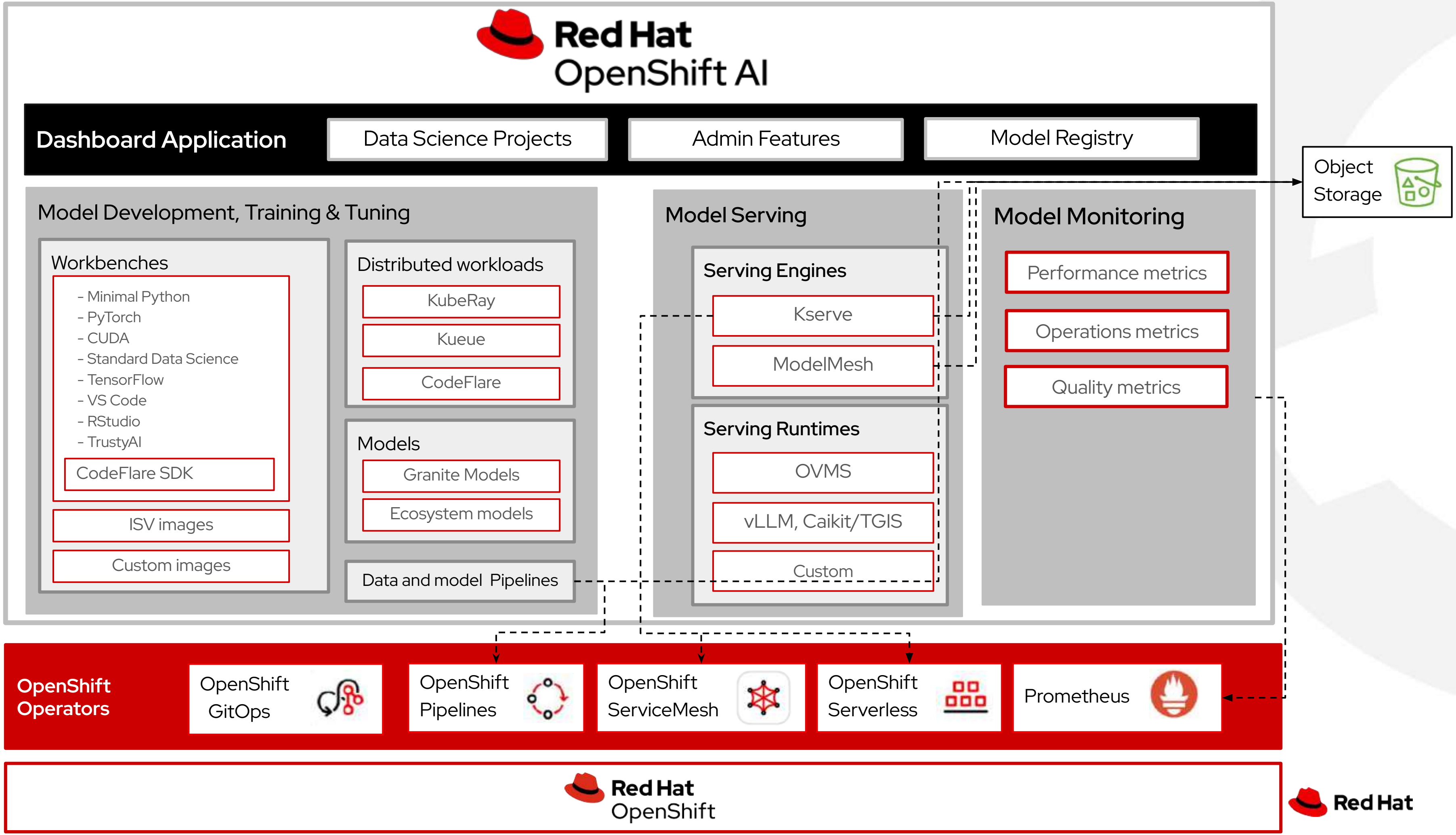
Manage accelerator profile settings for users in your organization

Name	Identifier	Enable
fractional small 1/7th of a real GPU	nvidia.com/gpu-frac	○
Habana HPU - 1st Gen Gaudi This Accelerator Profile is for 1st Gen Gaudi Devices	habana.ai/gaudi	○
Large GPU Card	nvidia.com/gpu	○
NVIDIA GPU - use sparingly We have very few GPUs in this cluster. Although you can use them fo...	nvidia.com/gpu	✓
tinyGPU	nvidia.com/gpu	○

Product Architecture


rhods-notebooks
Notebooks
Route
Route
Route
Notebook Resource (CR)
Notebook Resource (CR)
Notebook Resource (CR)
Pod
Oauth proxy
Workbench
Stateful Set
Notebook ConfigMap
Notebook Secret
Notebook PVC
redhat-ods-applications
Dashboard
Notebooks Management
(Kubeflow) notebook-controller
opendatahub-notebook-controller
Data Science Pipelines Management
data-science-pipelines-operator
Model Serving Management
(Kserve) model-mesh-controller
Etcd
opendatahub-model-mesh-controller
<User Project>
Notebooks
Route
Route
Route
Notebook Resource (CR)
Notebook Resource (CR)
Notebook Resource (CR)
Pod
Oauth proxy
Workbench
Stateful Set
Notebook ConfigMap
Notebook Secret
Notebook PVC
Model Serving
Route
Serving Runtime
Inference Service
Inference Service
Data Science Pipeline
Route
Data Science Pipeline Server
Database
S3 Storage
redhat-ods-monitoring
App Monitoring
Prometheus
Blackbox traffic exporter
redhat-ods-operator
App Deployment
RHOAI operator
RHOAI applications namespaces
Shared user namespace
Data Science Project

Product Components



Red Hat OpenShift AI

Dashboard Application
- Data Science Projects
- Admin Features
- Model Registry

Object Storage

Model Development, Training & Tuning

Workbenches
- Minimal Python
- PyTorch
- CUDA
- Standard Data Science
- TensorFlow
- VS Code
- RStudio
- TrustyAI

CodeFlare SDK

ISV images

Custom images

Distributed workloads
- KubeRay
- Kueue
- CodeFlare

Models
- Granite Models
- Ecosystem models

Data and model Pipelines

Model Serving

Serving Engines
- Kserve
- ModelMesh

Serving Runtimes
- OVMS
- vLLM, Caikit/TGIS
- Custom

Model Monitoring
- Performance metrics
- Operations metrics
- Quality metrics

OpenShift Operators
- OpenShift GitOps
- OpenShift Pipelines
- OpenShift ServiceMesh
- OpenShift Serverless
- Prometheus

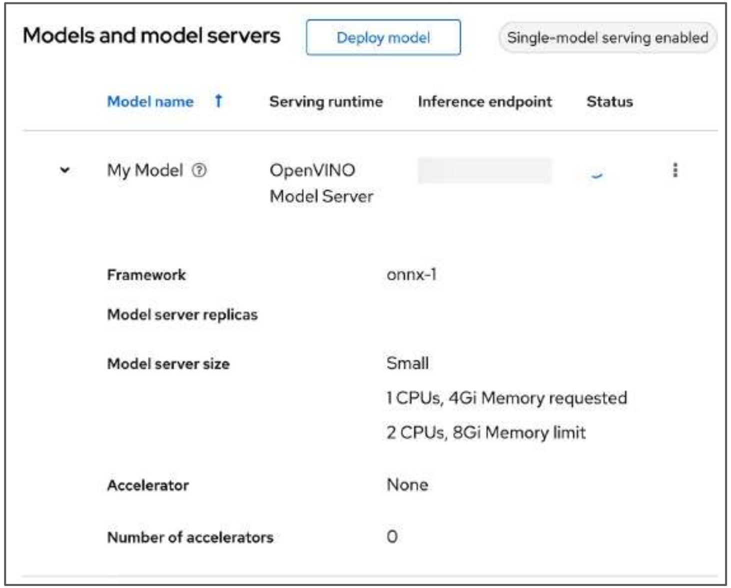
Red Hat OpenShift

Go into production

UI to Yaml

UI to Yaml

Everything in RHOAI has an OpenShift representation

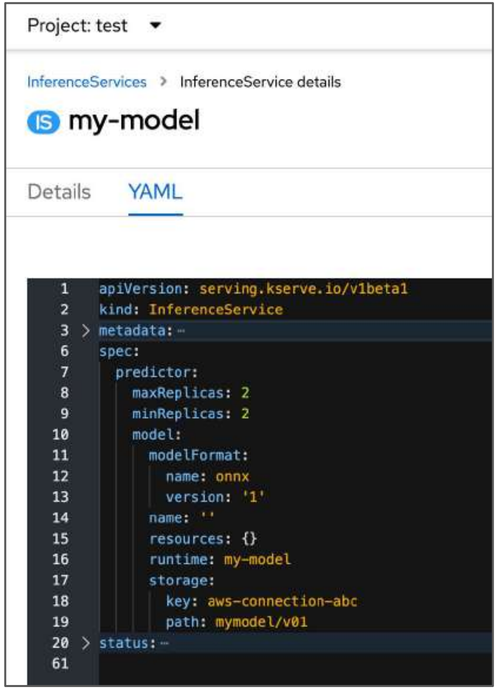


Models and model servers

Model name	Serving runtime	Inference endpoint	Status
My Model	OpenVINO Model Server		

Framework	onnx-1
Model server replicas	
Model server size	Small 1 CPUs, 4Gi Memory requested 2 CPUs, 8Gi Memory limit
Accelerator	None
Number of accelerators	0

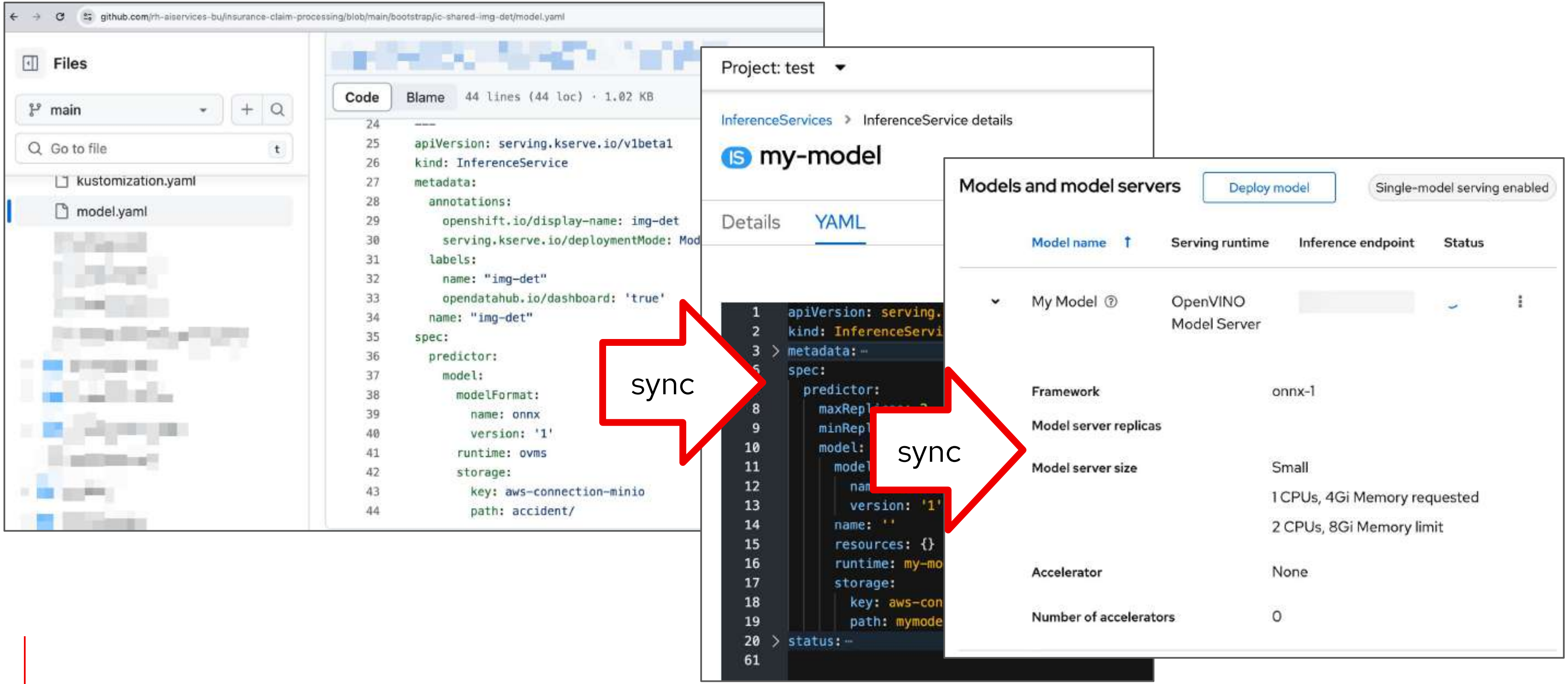
↔



```yaml
apiVersion: serving.kserve.io/v1beta1
kind: InferenceService
metadata: …
spec:
  predictor:
    maxReplicas: 2
    minReplicas: 2
    model:
      modelFormat:
        name: onnx
        version: '1'
      name: ''
      resources: {}
      runtime: my-model
      storage:
        key: aws-connection-abc
        path: mymodel/v01
status: …
```

UI to Yaml

GitOps



github.com/rh-aiservices-bu/insurance-claim-processing/blob/main/bootstrap/ic-shared-img-det/model.yaml

```
---
apiVersion: serving.kserve.io/v1beta1
kind: InferenceService
metadata:
  annotations:
    openshift.io/display-name: img-det
    serving.kserve.io/deploymentMode: Mod
  labels:
    name: "img-det"
    opendatahub.io/dashboard: 'true'
  name: "img-det"
spec:
  predictor:
    model:
      modelFormat:
        name: onnx
        version: '1'
      runtime: ovms
      storage:
        key: aws-connection-minio
        path: accident/
```

```
apiVersion: serving.
kind: InferenceServi
metadata: ⋯
spec:
  predictor:
    maxRepl
    minRepl
    model:
      model
        nam
        version: '1'
      name: ''
      resources: {}
      runtime: my-mo
      storage:
        key: aws-con
        path: mymode
status: ⋯
```

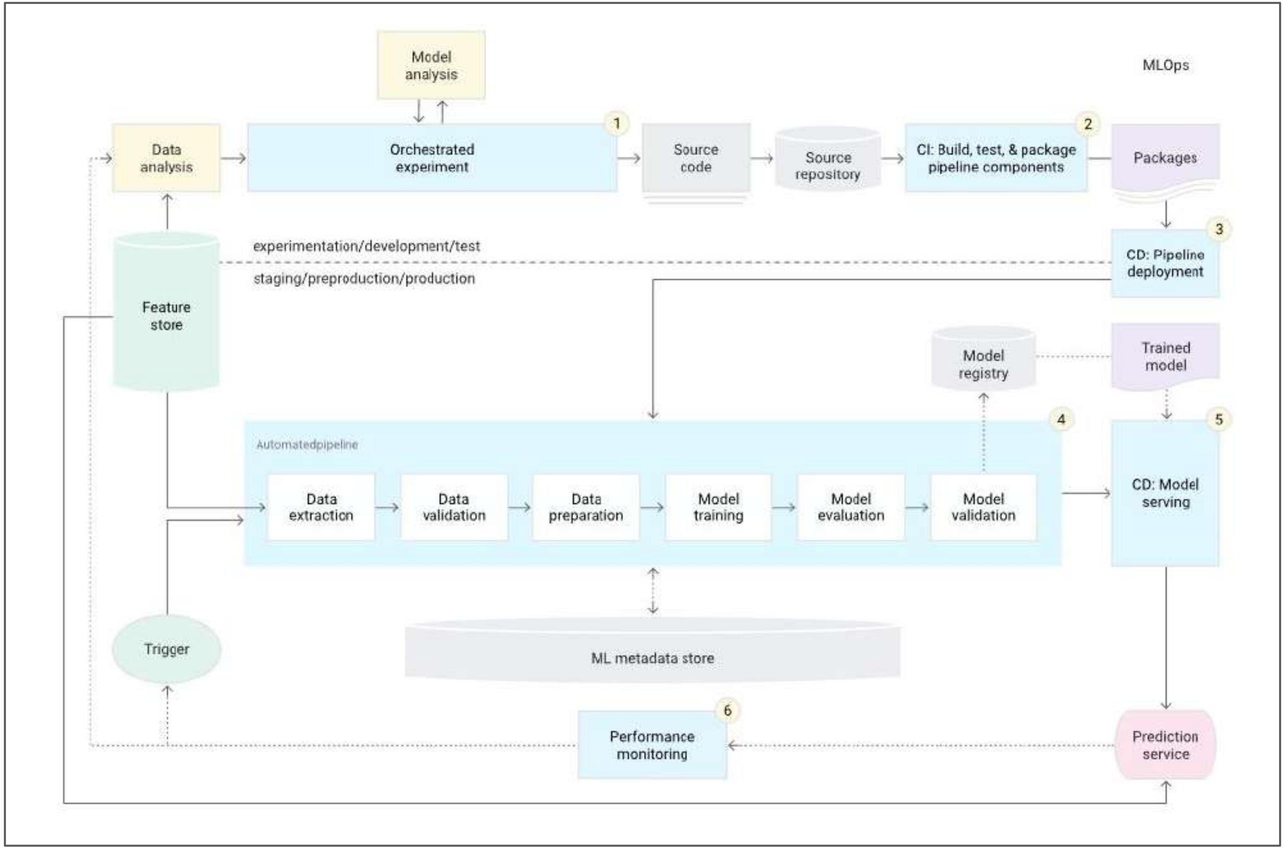
Model name	Serving runtime	Inference endpoint	Status
My Model	OpenVINO Model Server		

Framework	onnx-1
Model server replicas	
Model server size	Small 1 CPUs, 4Gi Memory requested 2 CPUs, 8Gi Memory limit
Accelerator	None
Number of accelerators	0

MLOps Automation

MLOps Automation

Mature MLOps Flow



https://cloud.google.com/architecture/mlops-continuous-delivery-and-automation-pipelines-in-machine-learning

Red Hat OpenShift

Extend with ISVs

NVIDIA

NVIDIA & Red Hat OpenShift AI

A certified solution to deploy and manage AI workloads in containers with optimized software

Ease of deployment and scale

Run AI workloads in the most optimal, scalable & secure infrastructure with a consistent platform for deployments

AI/ML Maximum performance

Ensure Machine Learning modeling and inference are executed with accelerated compute-intensive capabilities

Enable collaboration across teams

Provide self-service access to AI/ML tools and infrastructure, and streamline delivery of intelligent applications

Optional section marker or title

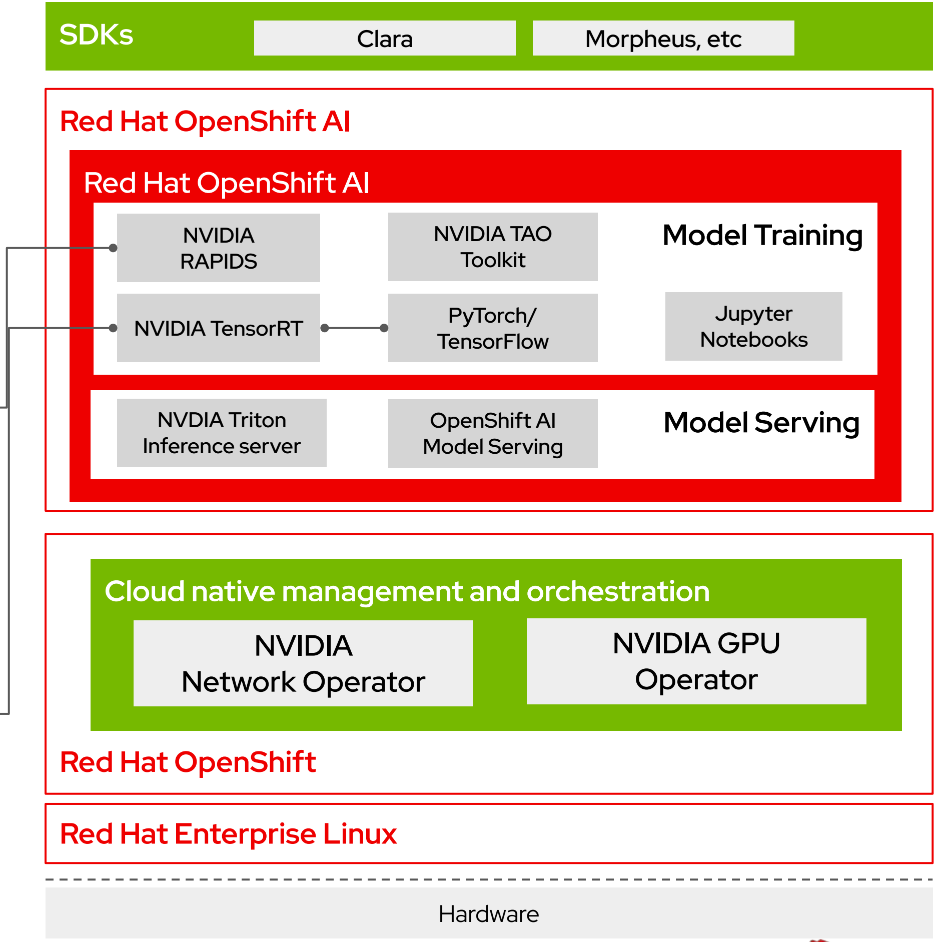
Train models faster and calibrate for higher precision

NVIDIA RAPIDs + OpenShift AI

Accelerate model training time by accessing data science libraries (numpy, pandas, scikit-learn, etc.) through Red Hat OpenShift AI Notebooks.

TensorFlow, PyTorch & NVIDIA TensorRT + OpenShift AI

Leverage GPU optimized deep learning and standard frameworks directly from Red Hat OpenShift AI Notebooks.



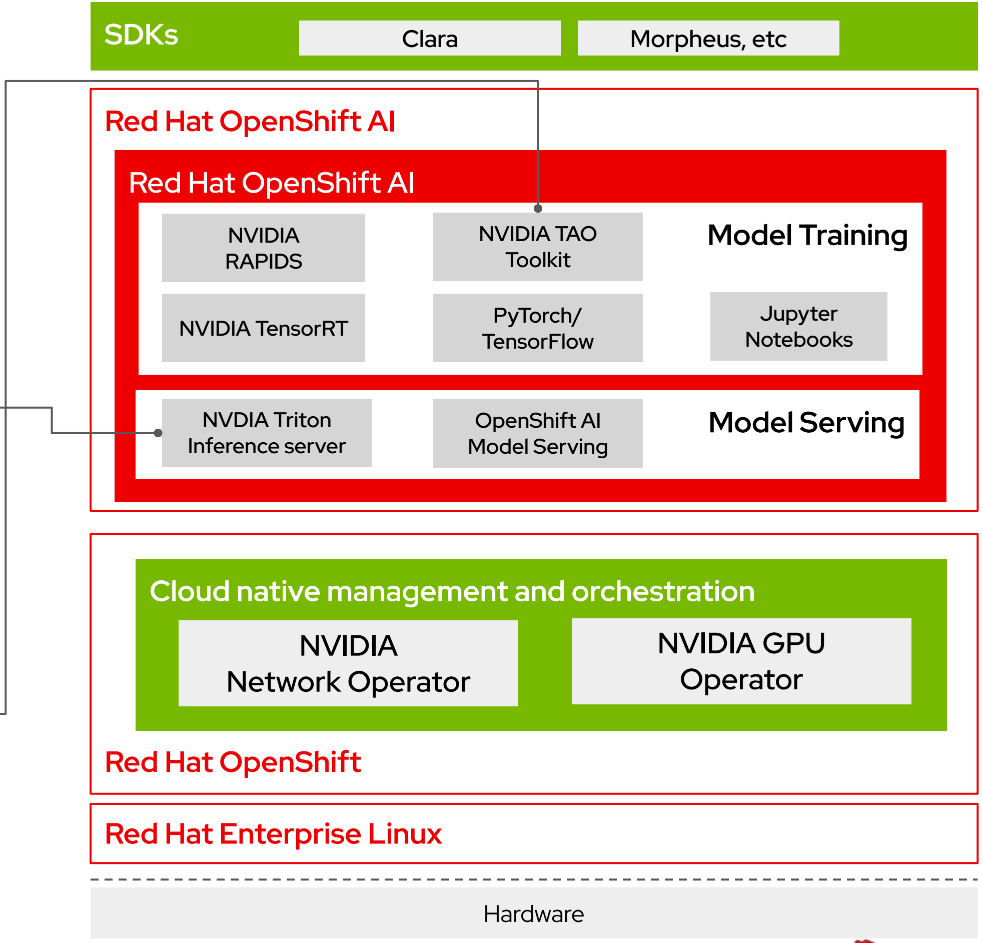
Train models faster and calibrate for higher precision

NVIDIA Triton Inference Server + OpenShift AI

Red Hat OpenShift AI ML Ops capabilities supports model execution in production for inferencing leveraging the GPU acceleration.

NVIDIA TAO toolkit + OpenShift AI

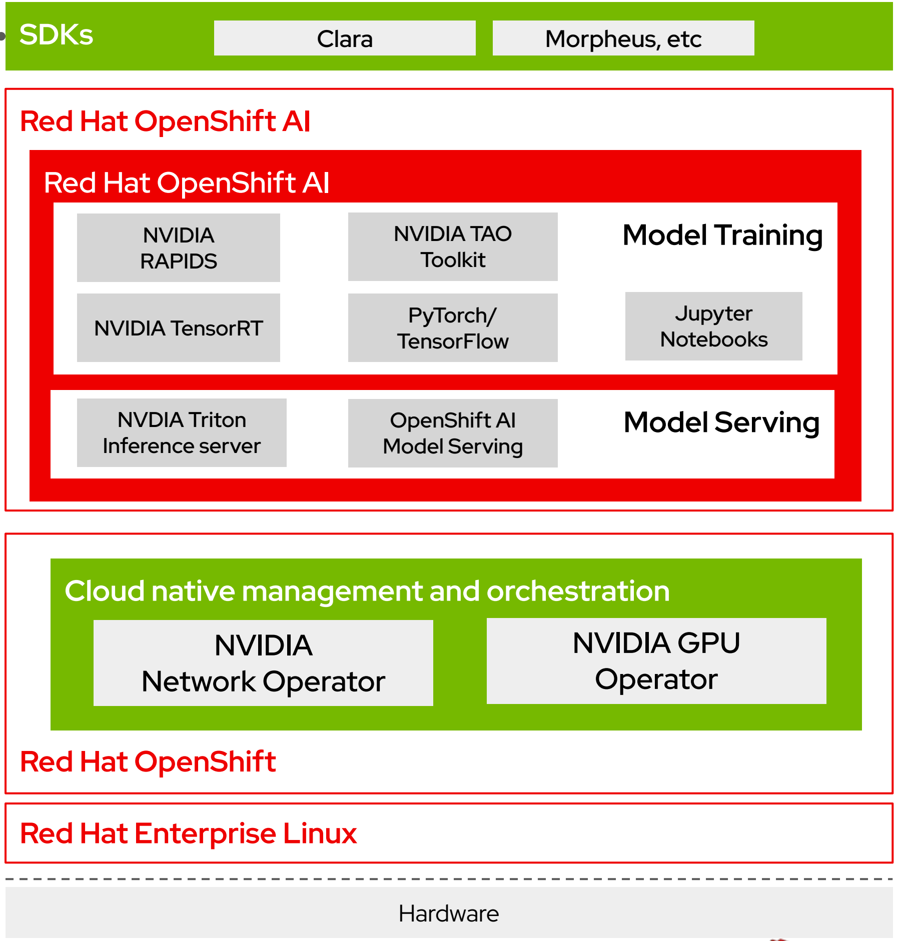
Train new models through transfer learning and monitor the model using OpenShift AI ML Ops capabilities.



Train models faster and calibrate for higher precision

NVIDIA NGC & SDKs

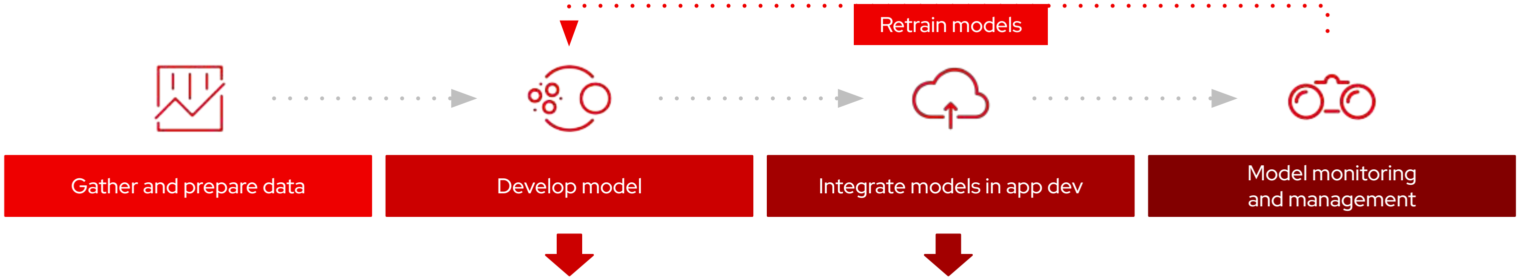
Users can combine models trained using Red Hat OpenShift AI with NVIDIA SDKs to develop AI enabled applications.



INTEL

Intel & Red Hat OpenShift AI

Accelerate data science using Intel hardware



Retrain models

Gather and prepare data

Develop model

Integrate models in app dev

Model monitoring and management

Accelerate model training

Out-of-the-box speed with AI Analytics Toolkit

intel AI TOOLS

AI Tools from Intel Benefits with RHOAI

- **Drop-in acceleration** with minimal code changes directly in notebooks
- Use low-level **optimizations** with popular Python AI frameworks
  - Tensorflow, PyTorch, NumPy & more on heterogeneous architectures
  - Speed up CPU intensive packages: Pandas, Scikit-Learn, & XGBoost
- **High Performance Intel Python distribution offers optimized and distributed compute**. Scale Pandas and Scikit-learn CPU and GPU workloads to multiple cores and nodes with minimal code changes.
- Increased model **accuracy** and **performance** using optimized algorithms within scikit-learn and XGBoost
- **Quantization** capabilities with the Intel Neural Compressor
- **Automated retraining and transfer learning**

Accelerate model inference

High performance inference using Intel CPUs

OpenVINO

Intel OpenVINO Benefits with RHOAI

- **High performance model inference** from edge to cloud
  - Support for multiple Deep Learning frameworks including TensorFlow, Caffe, PyTorch, MXNet, Keras, ONNX
  - Applicable to Machine & Deep Learning tasks: computer vision, speech recognition, natural language processing, and more
- **Easy Deployment of Model Server** at Scale in Kubernetes and OpenShift
- **Support multiple storage options** (S3, Azure Blob, GSC, local)
- **Configurable Resource Restrictions and Security Context** with OpenShift resource requirements
- **Quantization**
- **Configurable Service Options** based on infrastructure requirements

References:

- AI Analytics Toolkit

Starburst

Starburst & Red Hat OpenShift AI

Data Services for Modern AI/ML Use Cases

Performance

From petabytes to exabytes – query data from disparate sources using SQL – with high concurrency

Enhance your query performance with the latest cost-based optimizer

Caching available for frequently accessed data

Connectivity

40+ supported enterprise connectors

High performance parallel connectors for Oracle, Teradata, Snowflake and more

Ceph

Red Hat AMQ

teradata.

Microsoft SQL Server

ORACLE

Security

Kerberos, LDAP & SSO Integration

Global Security for fine-grained access control

Data Encryption/Masking

Higher security posture than vanilla K8's

Apache Ranger

KEYCLOAK

Management

Configuration

Autoscaling & High Availability

Query/Cluster Monitoring

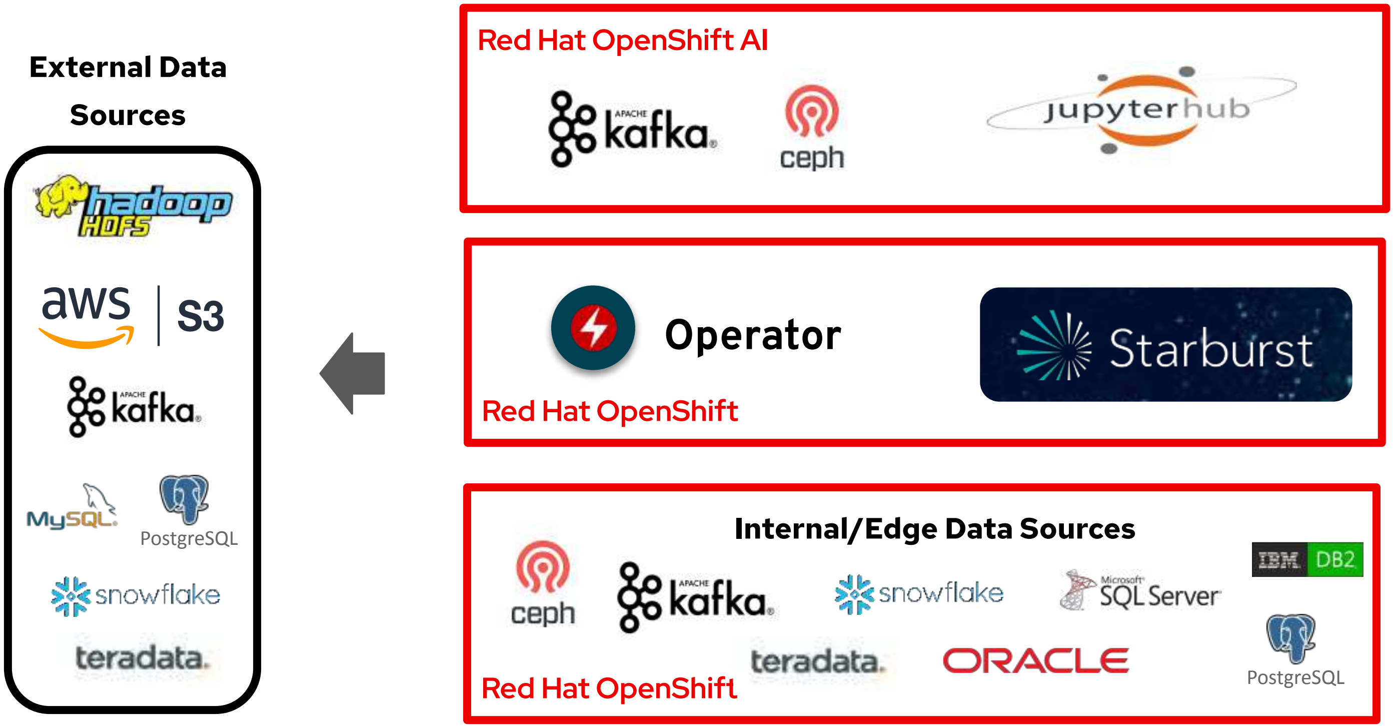
Deploy Anywhere

Multi-Cluster Management

Red Hat OpenShift

aws

Data Acquisition and Preparation



External Data Sources

Red Hat OpenShift AI

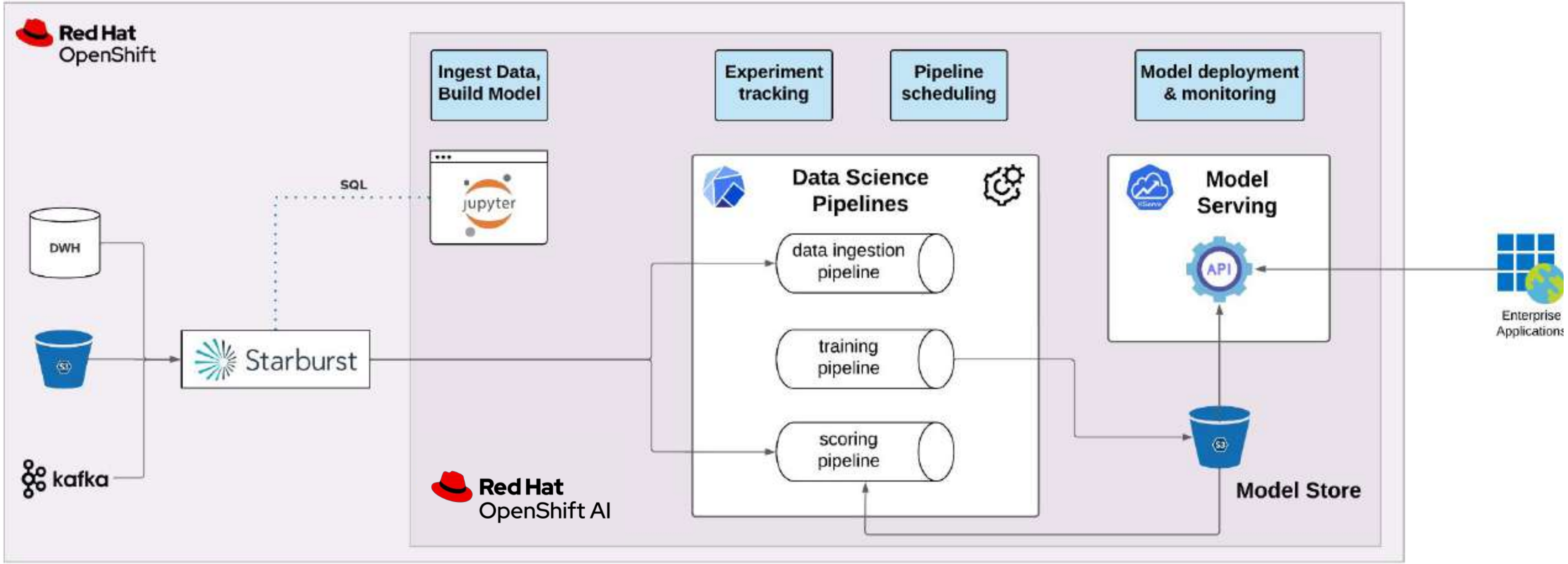
Operator

Red Hat OpenShift

Internal/Edge Data Sources

Red Hat OpenShift

ML workflow with RHOAI and Starburst



watsonx

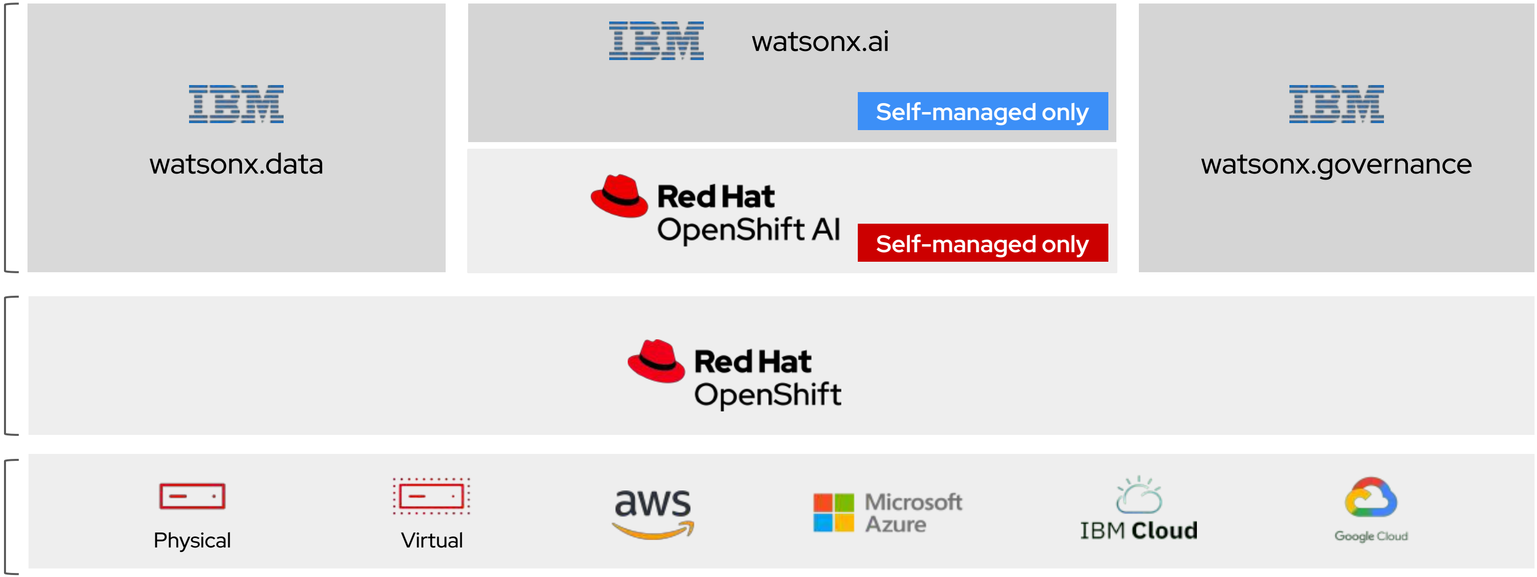
OpenShift AI + watsonx.ai

- Extend to include data processing, storage and governance along with visual foundation model tuning in an integrated offering with Watsonx.ai
- Accelerate **Generative AI** adoption
  - Using **IBM's Granite models** or **IBM's suite of curated foundation models** (through IBM's partnership with Hugging Face), **'Bring your own' foundation models** and open source foundation models.
  - Using **Prompt Lab** to customize foundation models with advanced prompt engineering capabilities.
- Advanced **MLOps capabilities** enabled visually or with code through a unified data+AI collaborative studio.
  - **AutoAI** automates end to end stages in AI/ML Lifecycle.
  - **Automated pipelines** with advanced features such as automated machine learning, model management and model monitoring pipelines.

Component Stack w/ all watsonx components

Red Hat OpenShift AI and IBM watsonx

High-performing, cloud-native AI open source stack runs on Red Hat OpenShift AI



Model development, serving and monitoring

IBM watsonx.data

IBM watsonx.ai

Self-managed only

Red Hat OpenShift AI

Self-managed only

IBM watsonx.governance

Orchestration, compute resource and fleet management

Red Hat OpenShift

Deploy anywhere

Red Hat Ansible Lightspeed
with IBM watsonx Code Assistant

The developer interface
Deployed natively in Visual Studio Code via the Ansible VS Code extension

The integrated service
Integration of AI services into Ansible Automation Platform via the Ansible VS Code extension

The generative AI
IBM watsonx Code Assistant powered by the Ansible-specific watsonx.ai foundation model

RHEL AI & InstructLab

RHEL AI

Red Hat
Enterprise Linux AI

Foundation Model Platform

Seamlessly develop, test, and run Granite family large language models (LLMs) for enterprise applications.

Granite family models

Open source-licensed LLMs, distributed under the Apache-2.0 license, with complete transparency on training datasets.

InstructLab model alignment tools

Scalable, cost-effective solution for enhancing LLM capabilities and making AI model development open and accessible to all users.

Optimized bootable model runtime instances

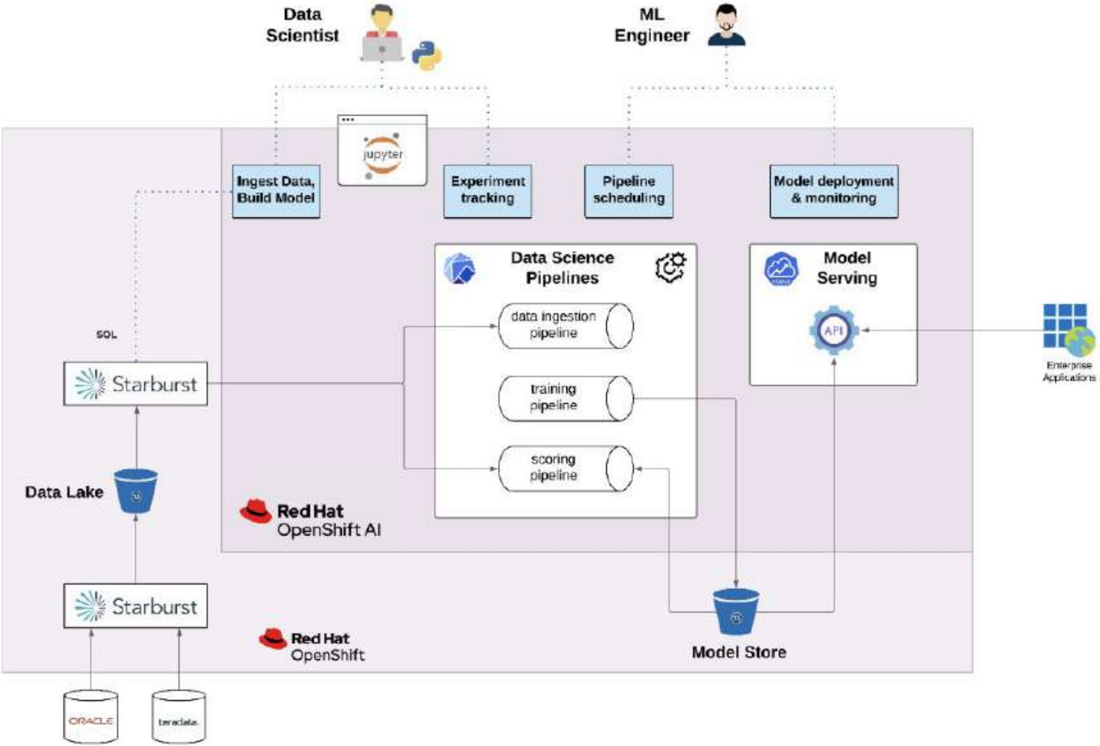
Granite models & InstructLab tooling packaged as a bootable RHEL image, including Pytorch/runtime libraries and hardware optimization (NVIDIA, Intel and AMD).

Enterprise support, lifecycle & indemnification

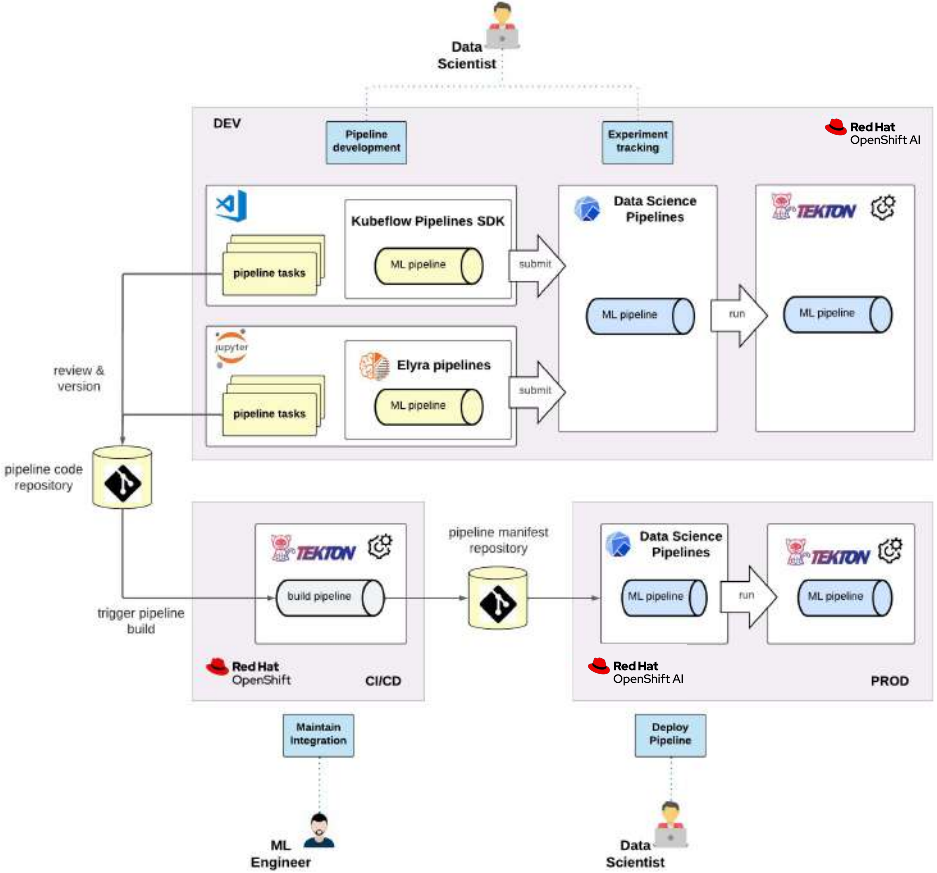
Trusted enterprise platform, 24x7 production support, extended model lifecycle and model IP indemnification by Red Hat.

Use Cases

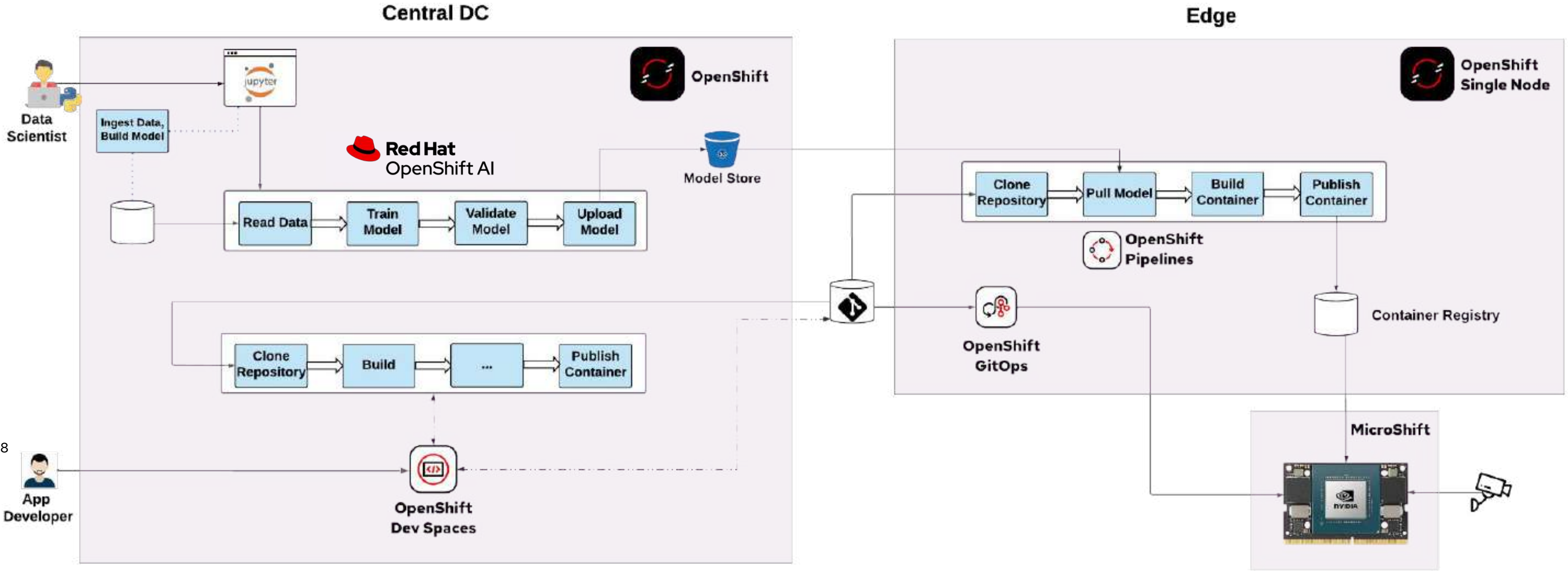
ML Platform at Airline Company



Using pipelines in production



Edge AI Delivery Workflow



Red Hat

Appendix