



Connect

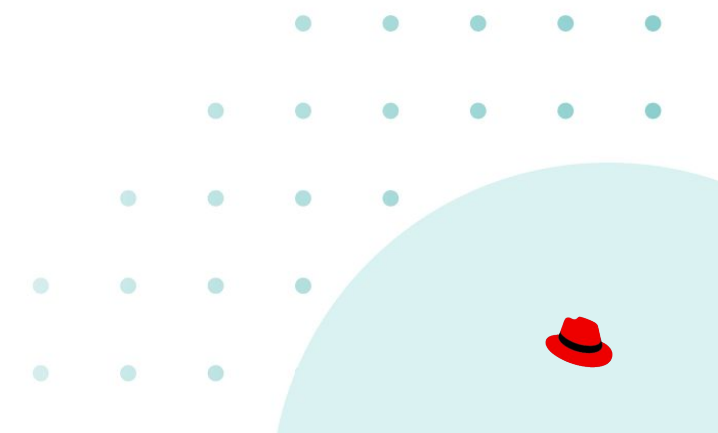
How OpenShift AI Streamlines ML Workflows for Faster, More Efficient Delivery?



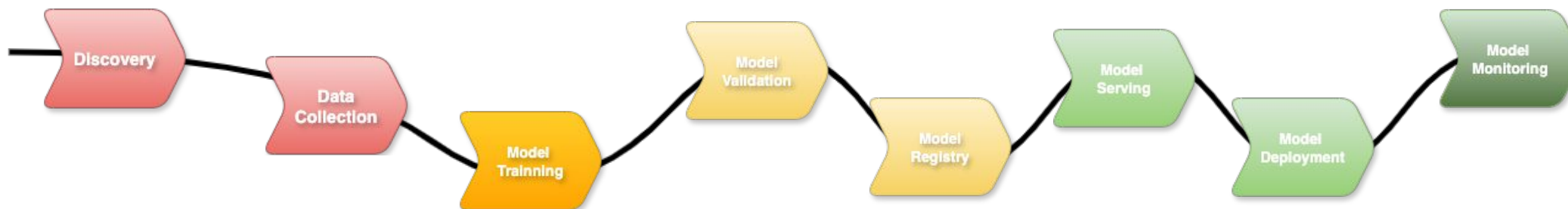


Paulo Menon

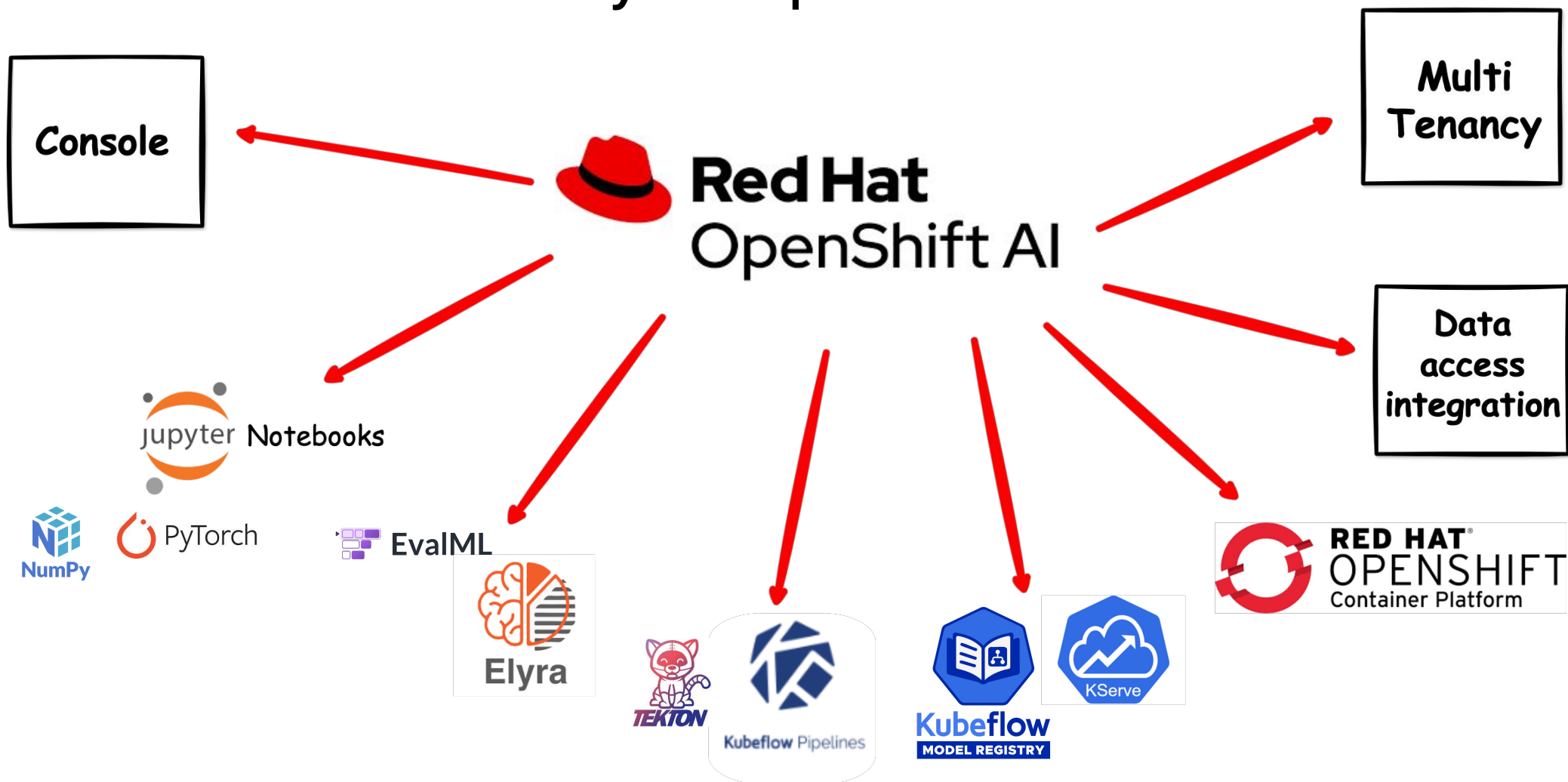
Senior Architect
Red Hat



What is a ML/AI Workflow?

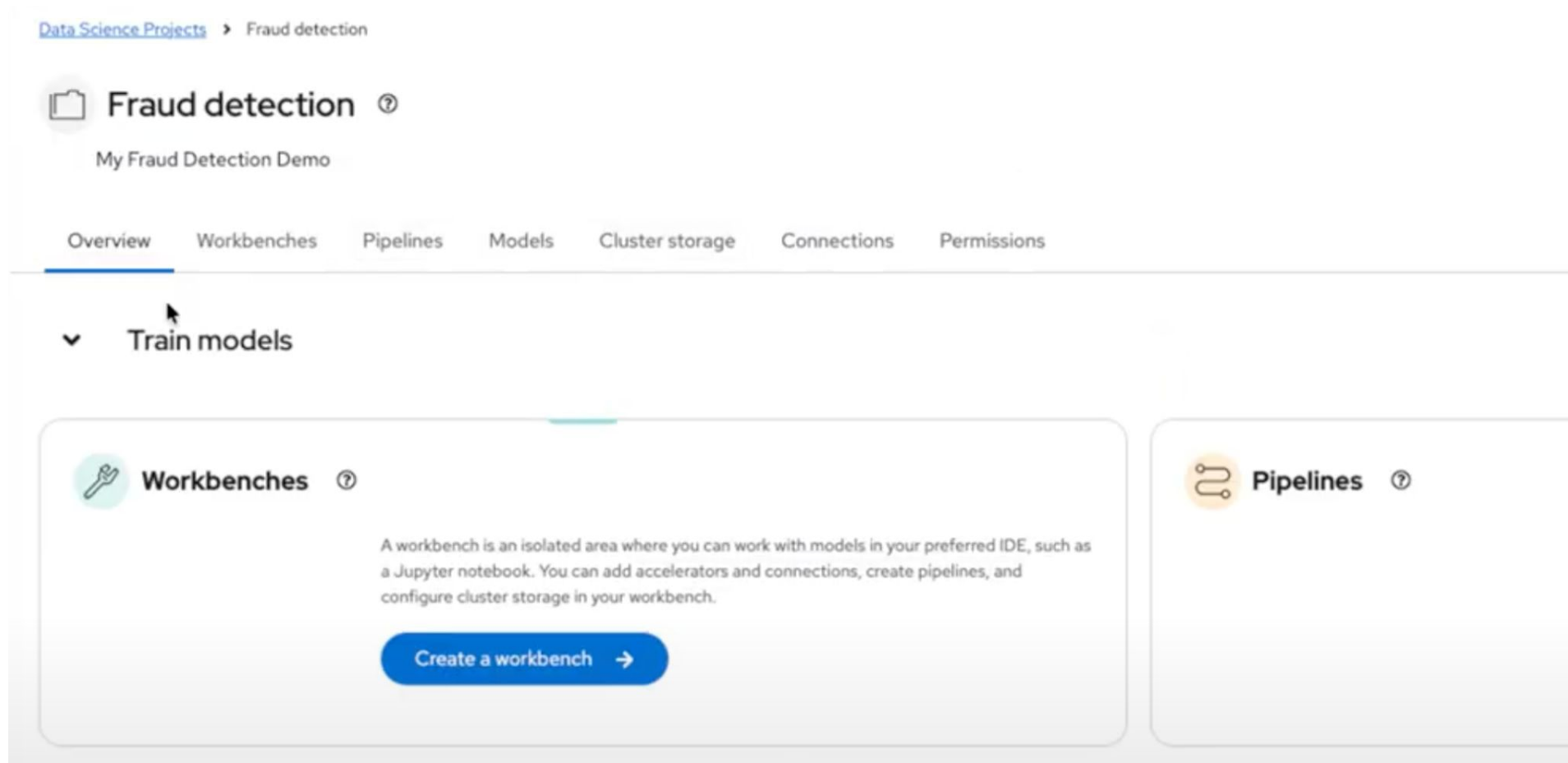


Why use Openshift AI?



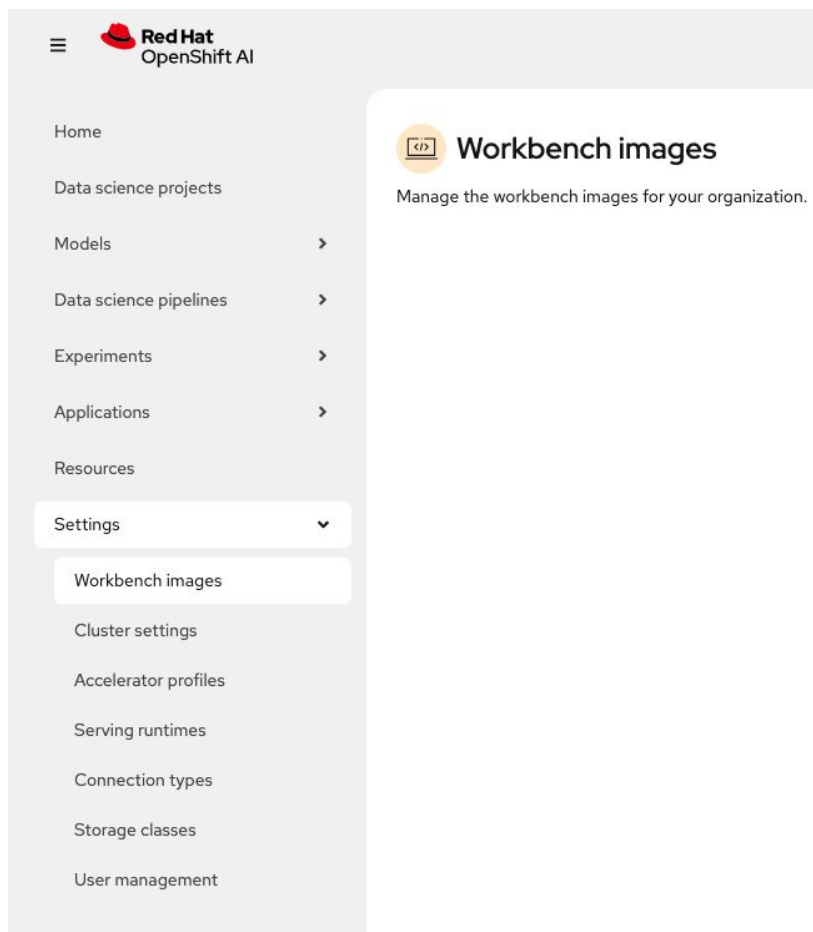
Configuring OpenShift AI workbenches

Use Case:



Configuring OpenShift AI workbenches

Configuration set up and workbench for this demo



Configuring Openshift AI workbenches

Create a Project and Workbench

Red Hat
OpenShift AI

Home

Data science projects

Models

Data science pipelines

Experiments

Applications

Resources

Settings

Data Science Projects > Fraud detection

Fraud detection ⓘ
My Fraud Detection Project

OverviewWorkbenchesPipelinesModelsCluster storageConnectionsPermissions

Workbenches ⓘ

Create workbench

| Name | Workbench image | Container size | Status |
|---|--|----------------|--|
| Fraud Detection ⓘ ⓘ My Fraud Detection workbench | TensorFlow 2025.1 (1d7de4c) Latest | Small | Starting Pulling workbench image |

Cluster storage

fraud-detection-storage
Max 20GiB
Mount path: /opt/app-root/src/ ⓘ

Packages

JupyterLab v4.2
TensorFlow v2.18
Tensorboard v2.18
Nvidia-CUDA-CUI2-Bundle v12.5
Boto3 v1.37
Kafka-Python-ng v2.2
Kfp v2.12
Matplotlib v3.10
Numpy v1.26
Pandas v2.2
Scikit-learn v1.6
Scipy v1.15
Odh-Elyra v4.2
PyMongo v4.11
Pyodbc v5.2
Codeflare-SDK v0.29
Sklearn-onnx v1.18
Pycopg v3.2

Limits

2 CPU, 8GiB Memory listed

Requests

1 CPU, 8GiB Memory requested



Implementing Openshift AI to streamline ML Workflow

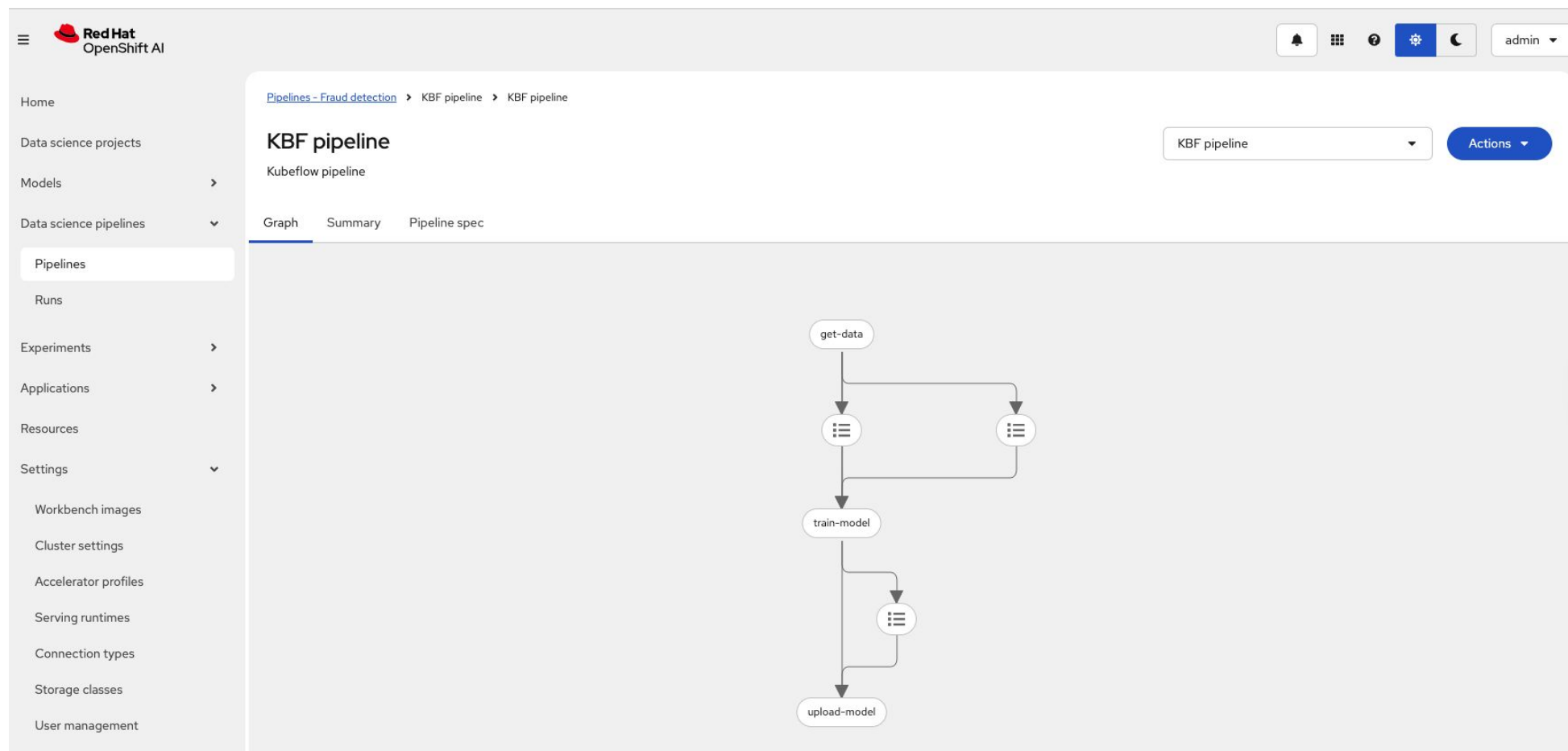
Inner loop

The screenshot displays the OpenShift AI Launcher interface. On the left, a file explorer shows the directory structure of a project named 'fraud-detection'. The files listed include 'assets', 'data', 'pipeline', 'ray-scripts', 'setup', 'utils', 'workshop', and several Jupyter notebooks (e.g., '0_sandbox.ipynb', '1_experiment_tr...', '2_save_model.ip...', etc.). The file '6 Train Save.pipe...' is selected. The main area shows a pipeline workflow with two steps: '1_experiment_...' and '2_save_model...'. The first step is highlighted with a blue border. The interface includes a menu bar (File, Edit, View, Run, Kernel, Git, Tabs, Settings, Help) and a toolbar with various icons for file operations and pipeline management. The top right corner indicates the runtime is 'Data Science Pipelines'.



Implementing Openshift AI to streamline ML Workflow

Outer loop



Implementing OpenShift AI to streamline ML Workflow

Model Register

Create model registry

Name *

Model Registry

The resource name will be model-registry.

[Edit resource name](#) ⓘ

Description

My model registry

Connect to external MySQL database

This external database is where model data is stored.

Host *

minio-service.showcase.svc.cluster.local

Port *

9000

Username *

minio

Password *

.....

🔒

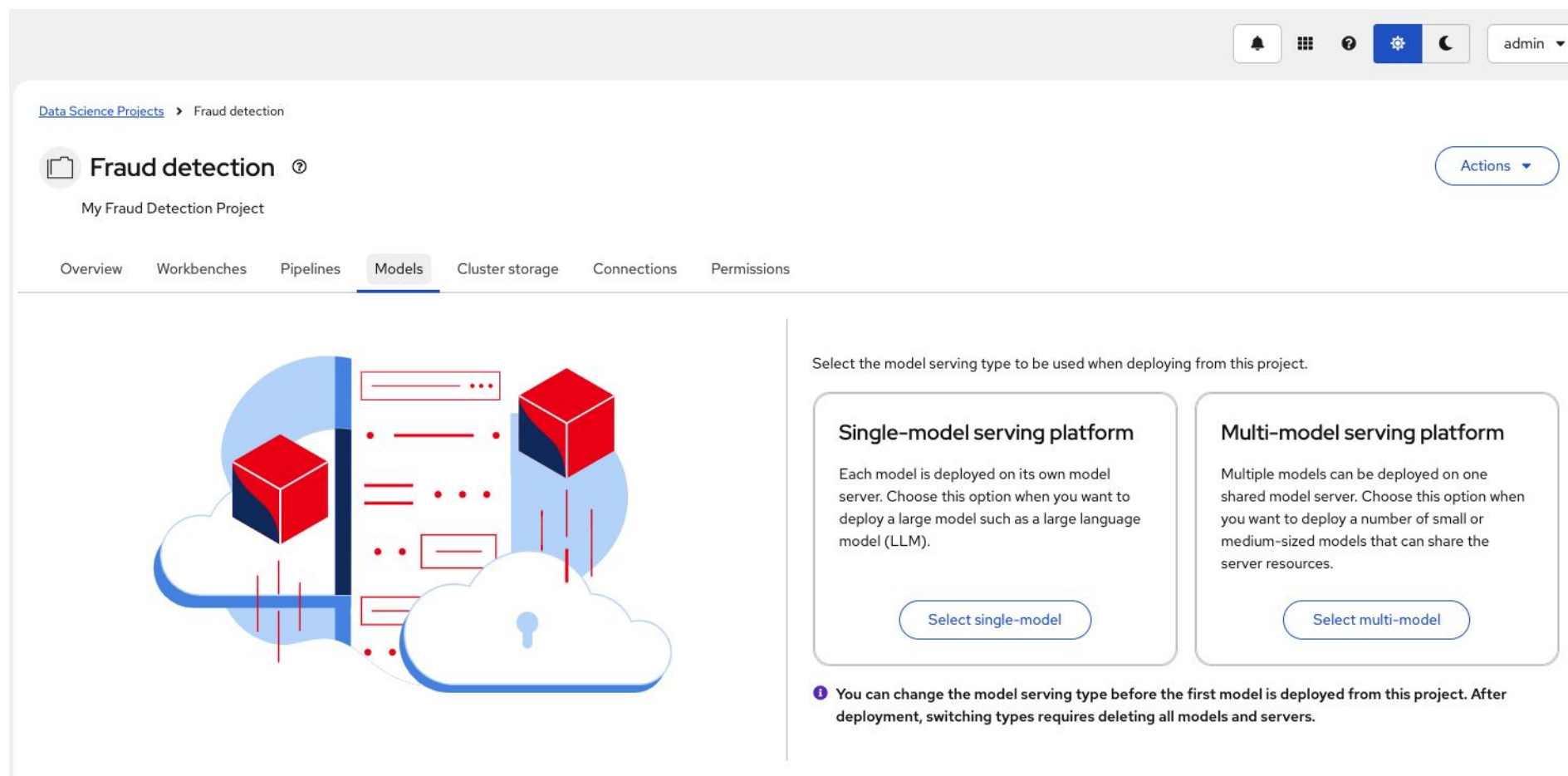
Create

Cancel



Implementing OpenShift AI to streamline ML Workflow

Model Serving

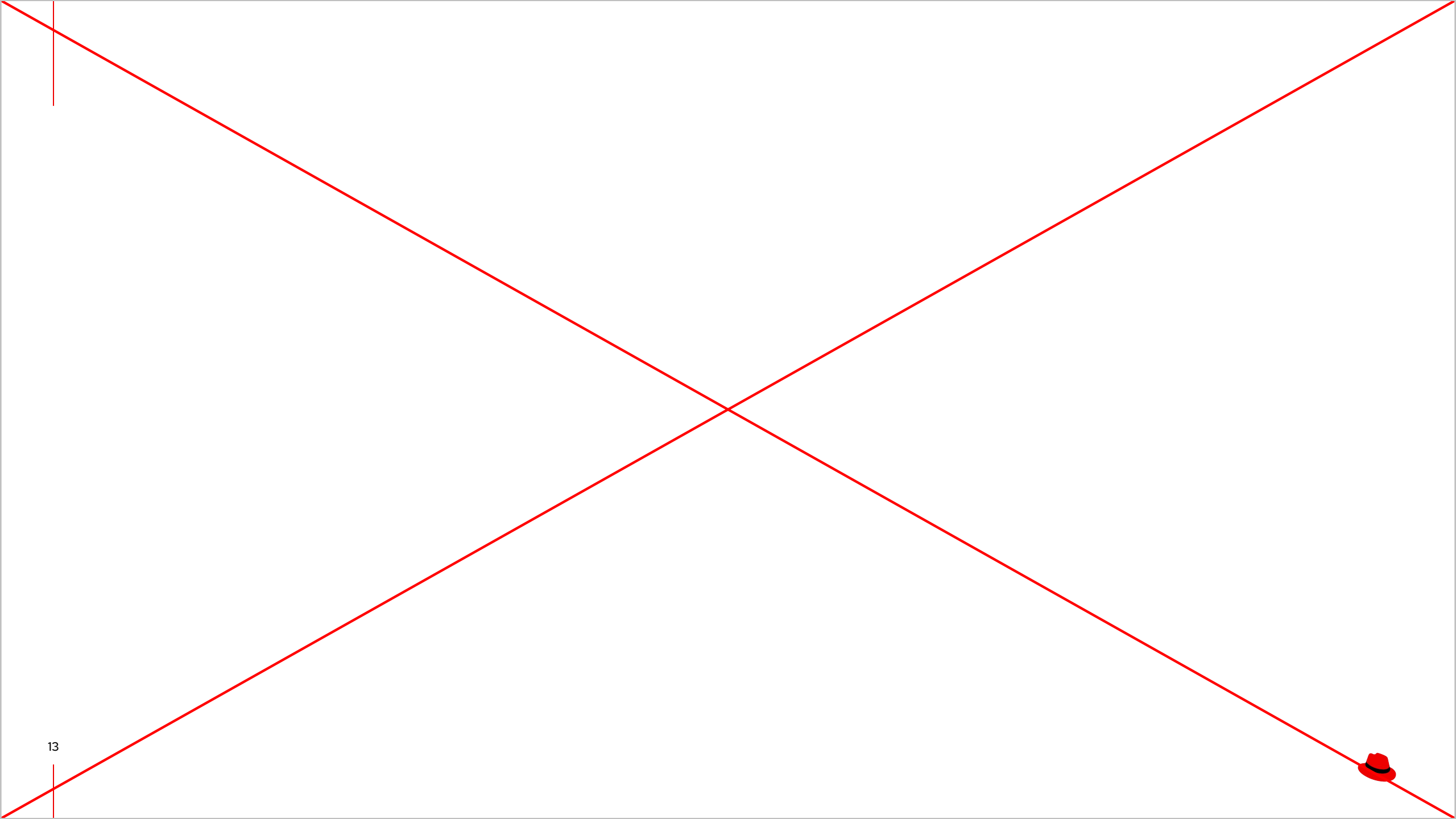


The screenshot displays the OpenShift AI console interface for a project named 'Fraud detection'. The top navigation bar includes a user profile 'admin' and various system icons. The breadcrumb trail shows 'Data Science Projects > Fraud detection'. The project name 'Fraud detection' is prominently displayed with a folder icon and a help icon. Below the project name, it says 'My Fraud Detection Project'. A horizontal tab bar contains 'Overview', 'Workbenches', 'Pipelines', 'Models' (which is the active tab), 'Cluster storage', 'Connections', and 'Permissions'. The main content area is divided into two sections. On the left, there is a large illustration featuring a blue cloud, a red cube, and a white cloud with a blue keyhole. On the right, there is a text prompt: 'Select the model serving type to be used when deploying from this project.' Below this prompt are two selectable options, each in a rounded rectangle. The first option is 'Single-model serving platform', which includes the text: 'Each model is deployed on its own model server. Choose this option when you want to deploy a large model such as a large language model (LLM).' and a button labeled 'Select single-model'. The second option is 'Multi-model serving platform', which includes the text: 'Multiple models can be deployed on one shared model server. Choose this option when you want to deploy a number of small or medium-sized models that can share the server resources.' and a button labeled 'Select multi-model'. At the bottom of the right section, there is an information icon followed by the text: 'You can change the model serving type before the first model is deployed from this project. After deployment, switching types requires deleting all models and servers.'



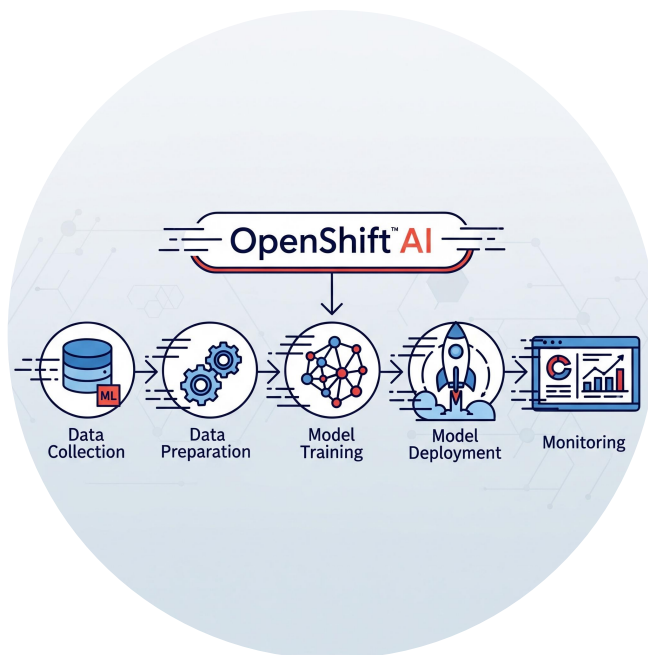
Demo time!





13

Concluding our session



- ▶ Key takeaways from our discussion
- ▶ Faster, more efficient ML delivery
- ▶ OpenShift AI streamlines ML workflows
- ▶ Questions and further discussion





Connect

Thank you



linkedin.com/company/red-hat



facebook.com/redhatinc



youtube.com/user/RedHatVideos



twitter.com/RedHat

