Set up Elyra with Data-Science Pipeline

Note: As Elyra inclusion in the notebook image is still a **work in progress**
And ODH-Dashboard support is yet to be worked on, a lot of the setup below would be automated, and users wouldn't go over the same steps.
This is major just to showcase a connection of notebooks with the data-science pipeline.

Pre-requisite:
- Openshift cluster with openshift-pipeline installed

Steps:

1. Setup data-science-pipeline in the cluster.
    a. (Only if DSPO is not installed via RHODS already)
    As DS pipeline operator is yet to be part of RHODS.
    One can install the DSPO with the help of kfdef:

```
apiVersion: kfdef.apps.kubeflow.org/v1
kind: KfDef
metadata:
 name: odh-core
spec:
 applications:
 - kustomizeConfig:
    repoRef:
      name: manifests
      path: odh-common
   name: odh-common
 - kustomizeConfig:
    parameters:
      - name: namespace
        value: openshift-operators
    repoRef:
      name: manifests
      path: openshift-pipelines/cluster
   name: openshift-pipelines
 - kustomizeConfig:
    repoRef:
      name: app
```

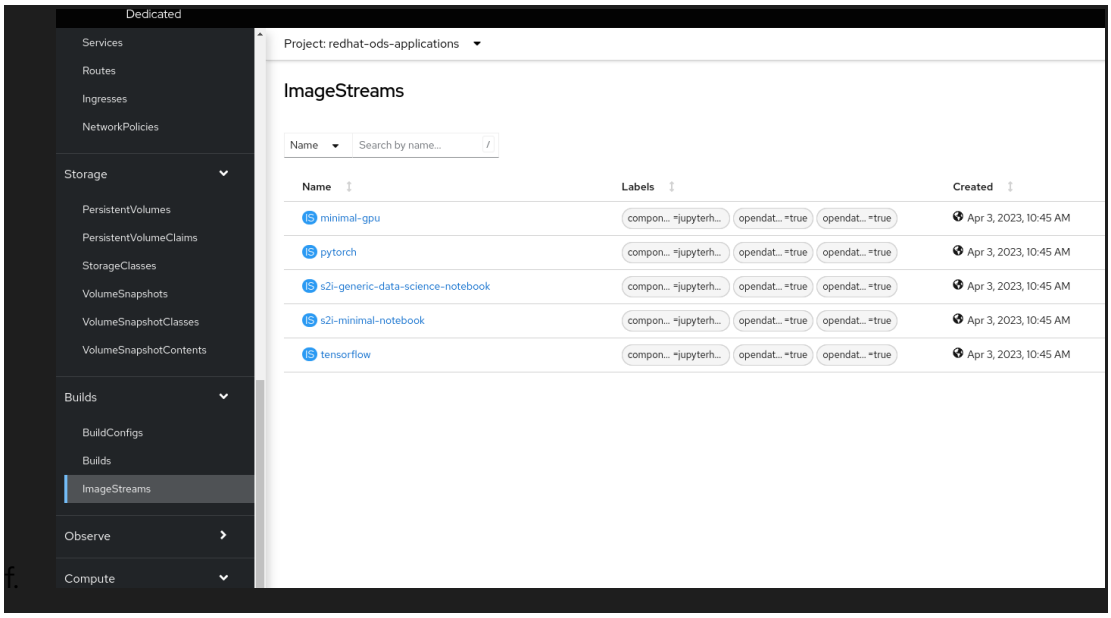
```
    path: config
  name: data-science-pipelines-operator
repos:
- name: manifests
  uri: https://github.com/opendatahub-io/odh-manifests/tarball/master
- name: app
  uri: https://github.com/opendatahub-io/data-science-pipelines-operator/tarball/main
version: master
```

b. Once data-science pipeline operator pods are up. An instance of a data-science pipeline can be set up.(all this would be later done through odh-dashboard in future)
   For now:
   Create a new namespace: `oc new-project dspa`
   Apply the CR:

```
apiVersion: datasciencepipelinesapplications.opendatahub.io/v1alpha1
kind: DataSciencePipelinesApplication
metadata:
 name: sample
spec:
 objectStorage:
  minio:
   image:
'quay.io/opendatahub/minio:RELEASE.2019-08-14T20-37-41Z-license-compliance'
 mlpipelineUI:
  image: 'quay.io/opendatahub/odh-ml-pipelines-frontend-container:beta-ui'
```

c. Once everything is set up, we require following information for elyra pipelines
   i. DS pipeline serve api : check for route on dspa namespace
   ii. Minio storage details: check for secret named **ml-pipeline-minio-artifact**
d. Now that ds-pipeline is setup, time to setup elyra notebook
e. On the redhat-ods-application namespace,
   Edit the imagestream `s2i-generic-data-science-notebook`

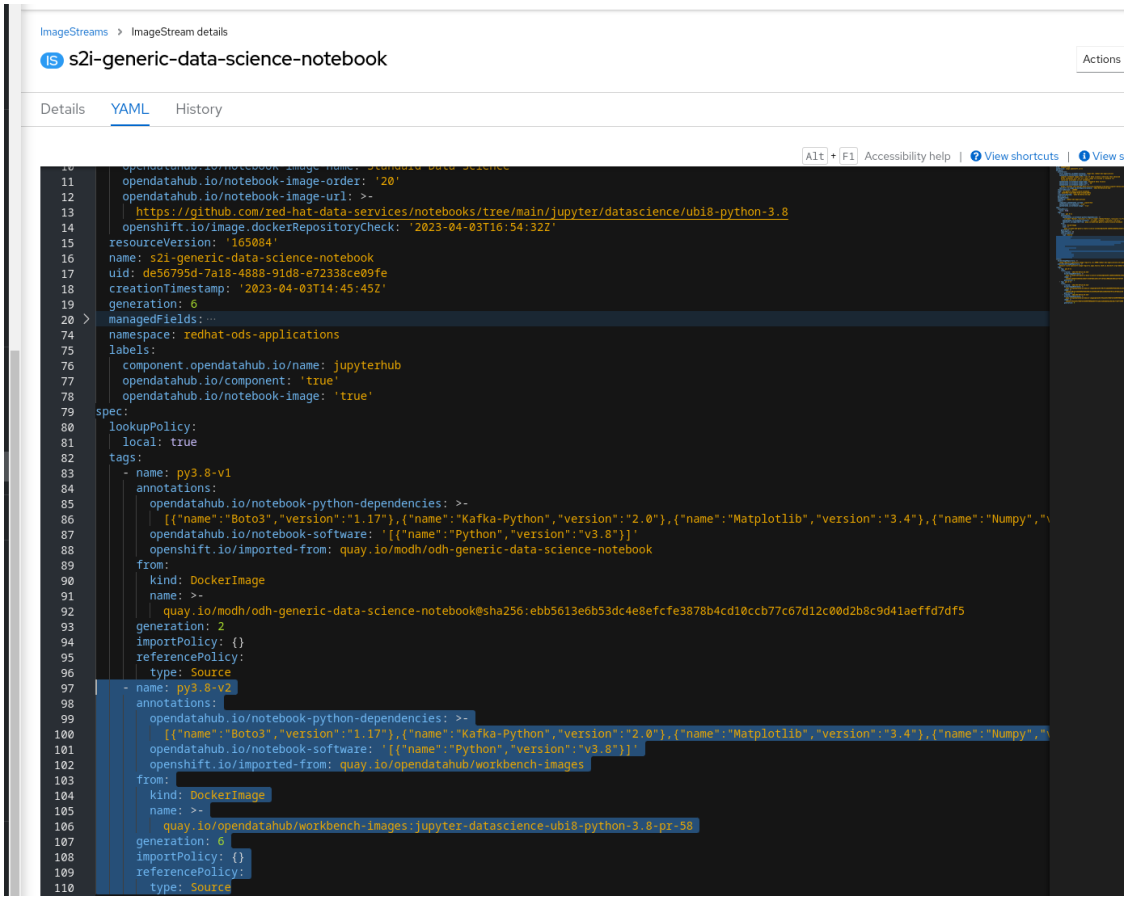
f.



Add additional tag:

```
- name: py3.8-v2
   annotations:
    opendatahub.io/notebook-python-dependencies: >-

[{"name":"Boto3","version":"1.17"},{"name":"Kafka-Python","version":"2.0"},{"name":"Matplotlib","version":"3.4"},{
"name":"Numpy","version":"1.19"},{"name":"Pandas","version":"1.2"},{"name":"Scikit-learn","version":"0.24"},{"nam
e":"Scipy","version":"1.6"}]
    opendatahub.io/notebook-software: '[{"name":"Python","version":"v3.8"}]'
    openshift.io/imported-from: quay.io/opendatahub/workbench-images
   from:
    kind: DockerImage
    name: >-
     quay.io/opendatahub/workbench-images:jupyter-datascience-ubi8-python-3.8-20230407-e95157b
   generation: 8
   importPolicy: {}
   referencePolicy:
     type: Source
```

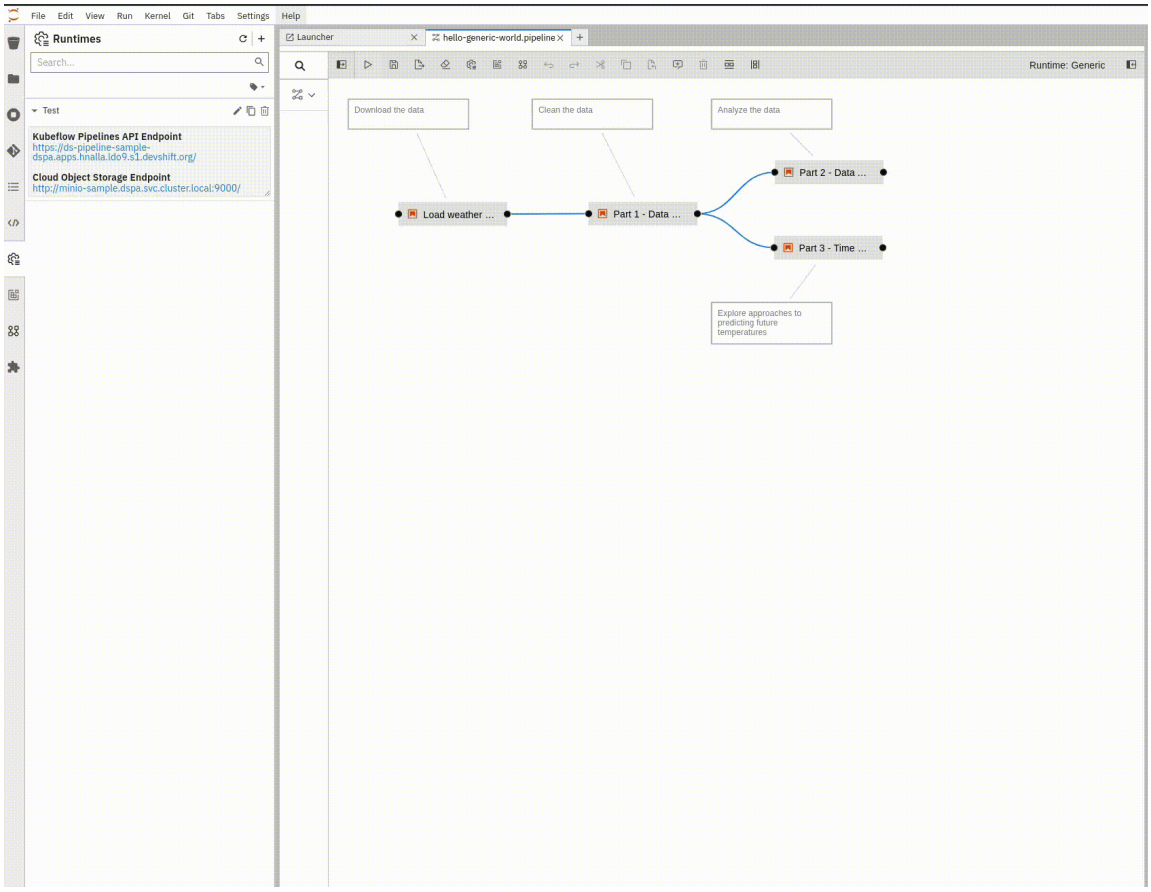


Once the changes are saved, Launch the notebook server selecting “Standard Data Science py3.8-v2”

- Once the server is started, we need to configure data -science-pipeline with notebook manually(Note: this would be automated by odh-dashboard)
  Head to runtimes configure:
  Fill the runtime configuration details:
  1. kubeflow pipeline API endpoint: ds pipeline server api route
  2. Authentication type: Existing_bearer_token
  3. Kubeflow Pipelines API Endpoint Password Or Token: user bearer token of openshift

  Leave all the other options black
  1. For cloud object storage:
     a. Cloud Object Storage Endpoint: http://minio-sample.dspa.svc.cluster.local:9000
     b. Cloud Object Storage Bucket Name: test
     c. Cloud Object Storage Credentials Secret: <access-token>

d. Cloud Object Storage Password: <secret-access-token>



Now we are set up.

Elyra has examples to run through the setup.

Follow the tutorial:

https://github.com/elyra-ai/examples/tree/main/pipelines/run-generic-pipelines-on-kubeflow-pipelines