

Kubeflow Pipelines



- Containerized implementations of ML Tasks
 - Pre-built components: Just provide params or code snippets (e.g. training code)
 - Create your own components from code or libraries
 - Use any runtime, framework, data types
 - Attach k8s objects - volumes, secrets

- Specification of the sequence of steps
 - Specified via Python DSL
 - Inferred from data dependencies on input/output

- Input Parameters
 - A “Run” = Pipeline invoked w/ specific parameters
 - Can be cloned with different parameters

- Schedules
 - Invoke a single run or create a recurring scheduled pipeline

The screenshot displays the Kubeflow Pipelines interface. At the top, there are navigation buttons: '+ Create run' (highlighted with a red circle), 'Upload version', '+ Create experiment', and 'Delete'. Below this is a breadcrumb trail: '[Demo] TFX - Taxi Tip Prediction Model Trainer ([Demo] TFX - T...'. The main area shows a pipeline graph with nodes: 'csvexamplegen', 'statisticsgen', 'schemagen', 'examplevalidator', 'transform', 'trainer', 'evaluator', 'modelvalidator', and 'pusher'. The graph shows data dependencies between these components.

At the bottom, a table lists various pipeline samples:

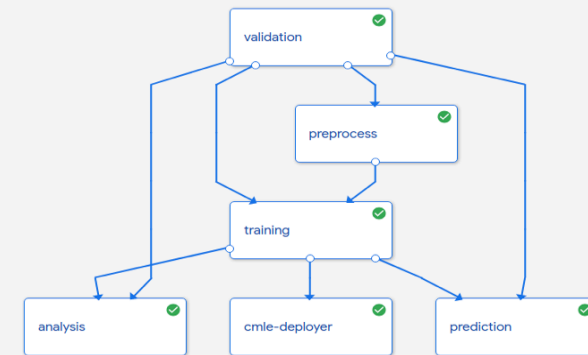
<input type="checkbox"/>	Pipeline name	Description	Uploaded on ↓
<input type="checkbox"/>	[Sample] Basic - Condition	A pipeline shows how to use dsl.Condition. For source code, refer to https://github.com/ku...	02/01/2019, 11:24:37
<input type="checkbox"/>	[Sample] Basic - Exit Handler	A pipeline that downloads a message and print it out. Exit Handler will run at the end. For s...	02/01/2019, 11:24:36
<input type="checkbox"/>	[Sample] Basic - Immediate ...	A pipeline with parameter values hard coded. For source code, refer to https://github.com/ku...	02/01/2019, 11:24:34
<input type="checkbox"/>	[Sample] Basic - Parallel Join	A pipeline that downloads two messages in parallel and print the concatenated result. For ...	02/01/2019, 11:24:33
<input type="checkbox"/>	[Sample] Basic - Sequential	A pipeline with two sequential steps. For source code, refer to https://github.com/kubeflo...	02/01/2019, 11:24:32
<input type="checkbox"/>	[Sample] ML - TFX - Taxi Tip ...	Example pipeline that does classification with model analysis based on a public tax cab BI...	02/01/2019, 11:24:30
<input type="checkbox"/>	[Sample] ML - XGBoost - Trai...	A trainer that does end-to-end distributed training for XGBoost models. For source code, re...	02/01/2019, 11:24:29

Define Pipeline with Python SDK

```

@dsl.pipeline(name='Taxi Cab Classification Pipeline Example')
def taxi_cab_classification(
    output_dir,
    project,
    Train_data      = 'gs://bucket/train.csv',
    Evaluation_data  = 'gs://bucket/eval.csv',
    Target          = 'tips',
    Learning_rate    = 0.1, hidden_layer_size = '100,50', steps=3000):

    tfdv              = TfdvOp(train_data, evaluation_data, project, output_dir)
    preprocess        = PreprocessOp(train_data, evaluation_data, tfdv.output["schema"], project, output_dir)
    training          = DnnTrainerOp(preprocess.output, tfdv.schema, learning_rate, hidden_layer_size, steps,
                                     target, output_dir)
    tfma              = TfmaOp(training.output, evaluation_data, tfdv.schema, project, output_dir)
    deploy            = TfServingDeployerOp(training.output)
  
```



Compile and Submit Pipeline Run

```

dsl.compile(taxi_cab_classification, 'tfx.tar.gz')
run = client.run_pipeline(
    'tfx_run', 'tfx.tar.gz', params={'output': 'gs://dpa22', 'project': 'my-project-33'})
  
```



Visualize the state of various components

The screenshot displays the Kubeflow dashboard interface. On the left is a navigation sidebar with options: Pipelines, Experiments, Artifacts, Executions, Archive, Documentation, Github Repo, and AI Hub Samples. The main area shows a pipeline execution graph with nodes: csvexampleger, statisticsgen, schemagen, examplevalidator, resolvernode-lates..., evaluator, train, and pusher. The 'evaluator' node is highlighted with a green checkmark. A modal window titled 'csvexampleger' is open, showing tabs for Artifacts, Input/Output, Volumes, Manifest, and Logs. The 'Artifacts' tab is active, displaying a 'Static HTML' artifact. This artifact contains a table of feature statistics.

Cluster name: cluster-4
Build commit: 743746b
[Report an Issue](#)

Runtime execution graph. Only steps that are currently running or have a

Numeric Features (15)				
	count	missing	mean	std dev
dropoff_census_tract	3,618	28.93%	17.0B	333k
dropoff_community_area	4,905	3.65%	21.2	17.85
dropoff_latitude	4,915	3.46%	41.9	0.04
dropoff_longitude	4,915	3.46%	-87.65	0.06



Pipelines versioning

Pipelines

[+ Upload pipeline](#) [Refresh](#) [Delete](#)

Filter pipelines

<input type="checkbox"/>	Pipeline name	Description	Uploaded on ↓
<input type="checkbox"/>	▶ [Tutorial] DSL - Control structures	source code Shows how to use conditional execution and exit handlers. This pipeline will randomly fail to demonstra...	2/20/2020, 3:28:12 PM
<input type="checkbox"/>	▶ [Tutorial] Data passing in python com...	source code Shows how to pass data between python components.	2/20/2020, 3:28:11 PM
<input type="checkbox"/>	▼ [Demo] TFX - Taxi Tip Prediction Mod...	source code GCP Permission requirements . Example pipeline that does classification with model analysis based on ...	2/20/2020, 3:28:10 PM
<input type="checkbox"/>	Version name		Uploaded on ↓
<input type="checkbox"/>	TFX - Taxi Tip Prediction Model Trainer_version_at_2020-03-03T15:44:30.197Z		3/3/2020, 7:55:03 AM
<input type="checkbox"/>	[Demo] TFX - Taxi Tip Prediction Model Trainer		2/20/2020, 3:28:10 PM

Rows per page: 10 < >

<input type="checkbox"/>	▶ [Demo] XGBoost - Training with Confu...	source code GCP Permission requirements . A trainer that does end-to-end distributed training for XGBoost models.	2/20/2020, 3:28:09 PM
--------------------------	---	---	-----------------------

Rows per page: 10 < >

Pipelines lets you group and manage multiple versions of a pipeline.



Artifact Tracking

Getting Started Pipelines Experiments **Artifacts** Executions Archive Documentation Github Repo AI Hub Samples

Artifacts

Filter

Pipeline/Workspace ↑	Name	ID	Type	URI	Created at
		1	ExternalArtifact	gs://ml-pipeline-playground/tfx_t...	
taxi_pipeline_with_parameters	examples	2	Examples	gs://aju-pipelines/tfx_taxi_simpl...	2/20/2020, 5:1...
	statistics	3	ExampleStatistics	gs://aju-pipelines/tfx_taxi_simpl...	2/20/2020, 5:1...
	schema	4	Schema	gs://aju-pipelines/tfx_taxi_simpl...	2/20/2020, 5:1...
	anomalies	5	ExampleAnomalies	gs://aju-pipelines/tfx_taxi_simpl...	2/20/2020, 5:1...
	transform_graph	6	TransformGraph	gs://aju-pipelines/tfx_taxi_simpl...	2/20/2020, 5:1...
	transformed_e...	7	Examples	gs://aju-pipelines/tfx_taxi_simpl...	2/20/2020, 5:1...
	model	8	Model	gs://aju-pipelines/tfx_taxi_simpl...	2/20/2020, 5:2...
	evaluation	9	ModelEvaluation	gs://aju-pipelines/tfx_taxi_simpl...	2/20/2020, 5:2...
	blessing	10	ModelBlessing	gs://aju-pipelines/tfx_taxi_simpl...	2/20/2020, 5:2...
	pushed_model	11	PushedModel	gs://aju-pipelines/tfx_taxi_simpl...	2/20/2020, 5:2...
	evaluation	12	ModelEvaluation	gs://aju-pipelines/tfx_taxi_simpl...	2/20/2020, 5:4...

Artifacts for a run of the “TFX Taxi Trip” example pipeline. For each artifact, you can view details and get the artifact URL—in this case, for the model.

Artifacts

← model

Overview Lineage Explorer

Type: Model

URI
[gs://aju-pipelines/tfx_taxi_simple/85265540-6a06-4969-a49e-1f65741878be/Trainer/model/7](#)

Properties

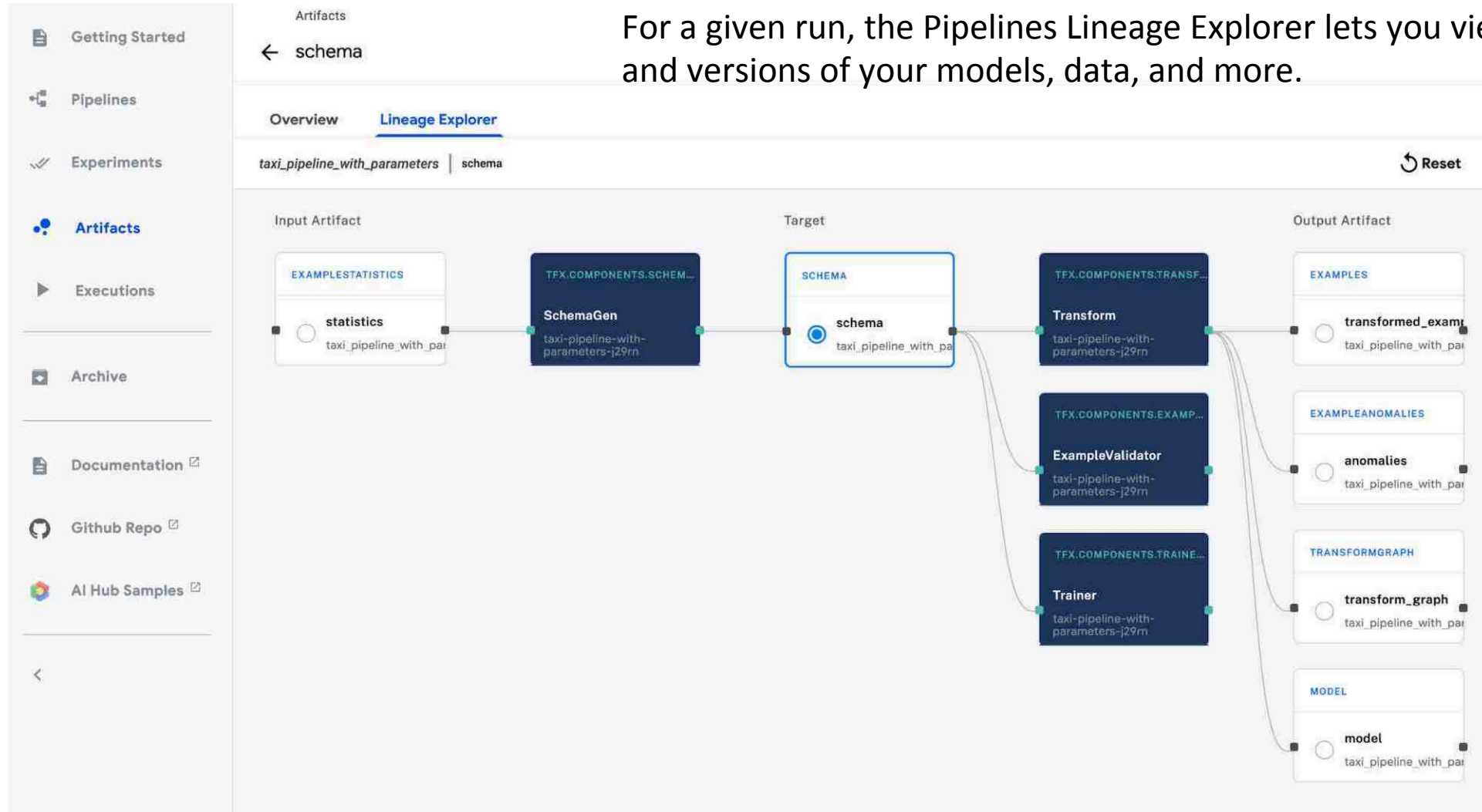
Custom Properties

name	pipeline_name	producer_component	state
model	taxi_pipeline_with_parameters	Trainer	published

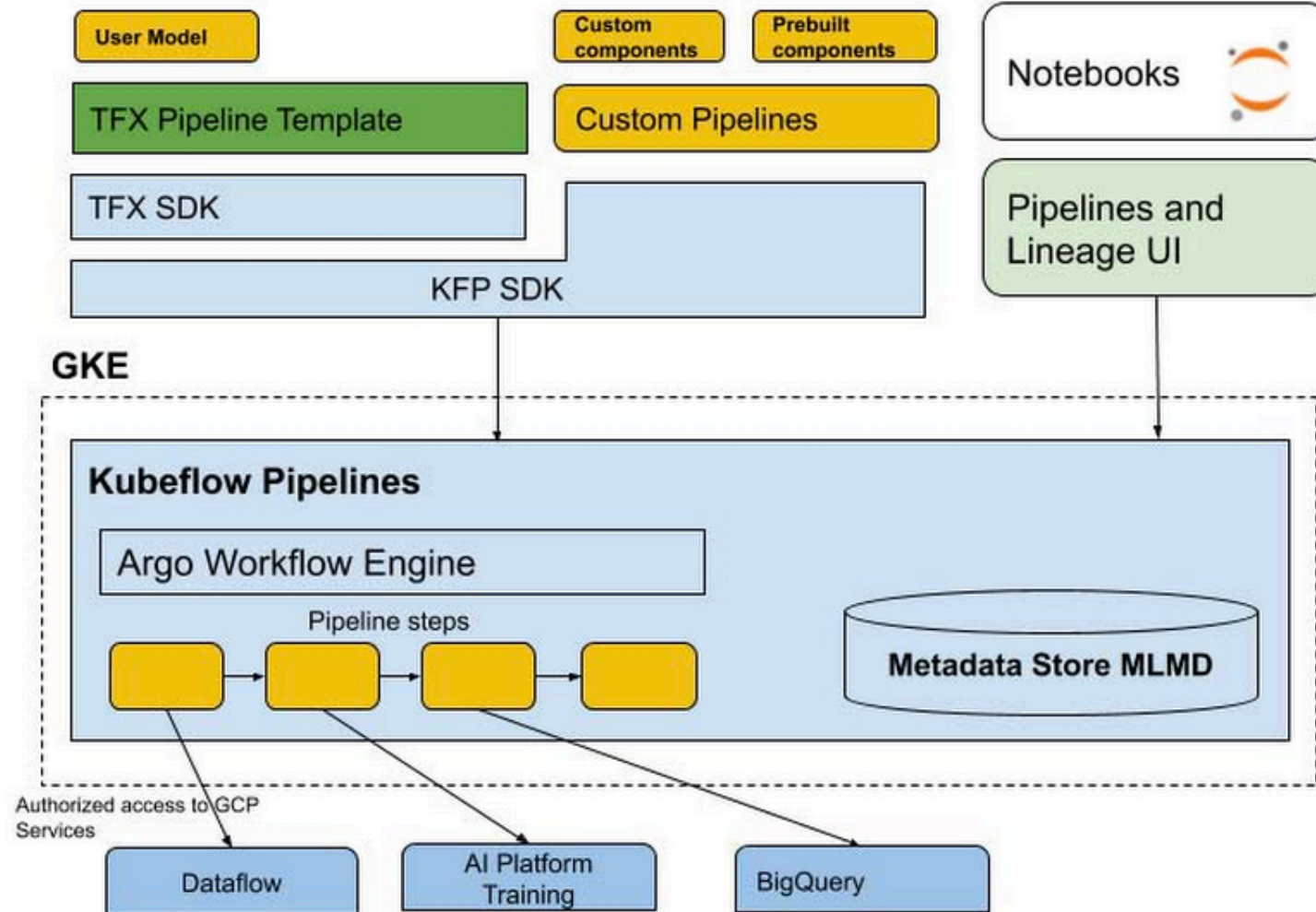


Lineage Tracking

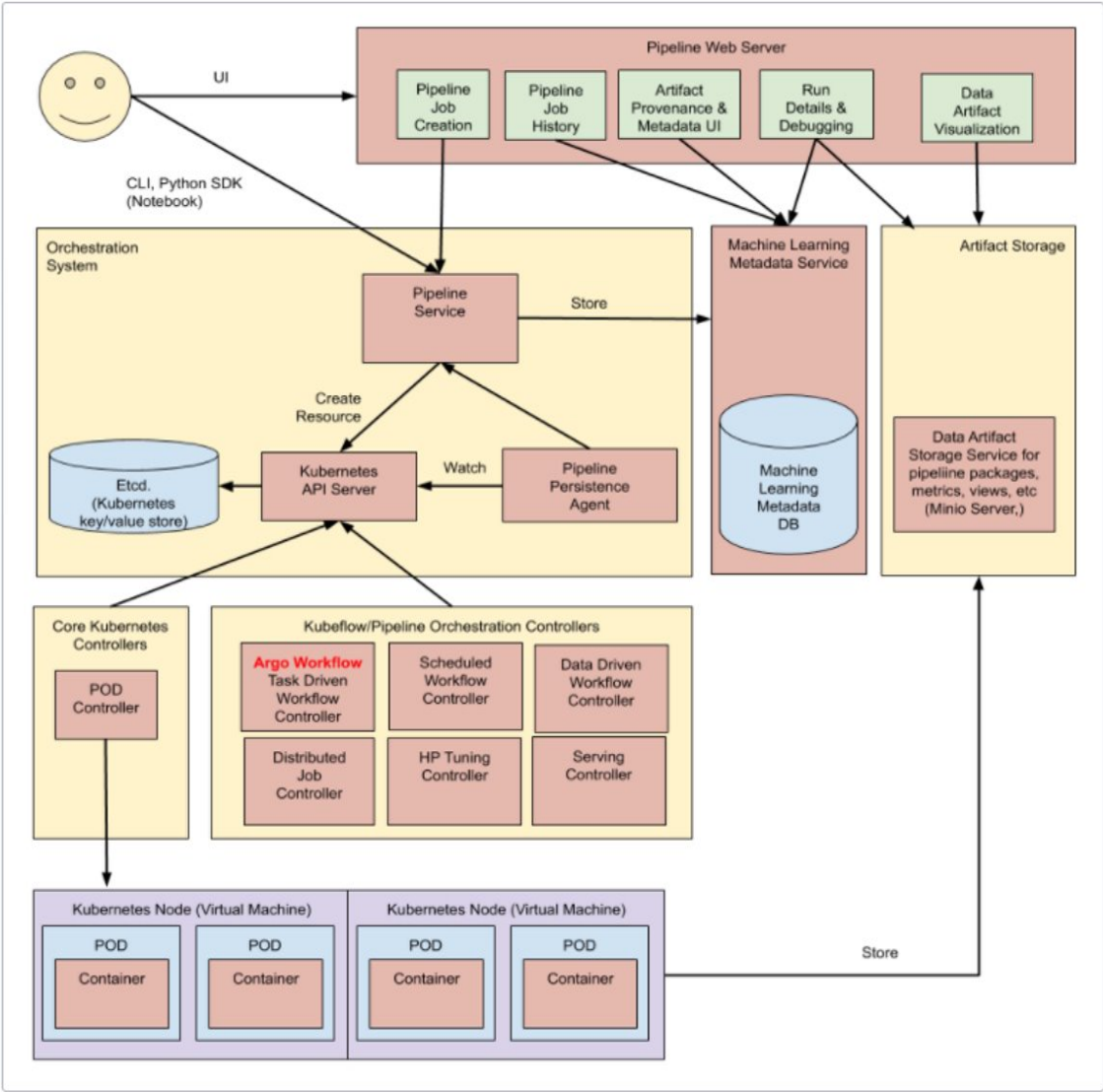
For a given run, the Pipelines Lineage Explorer lets you view the history and versions of your models, data, and more.



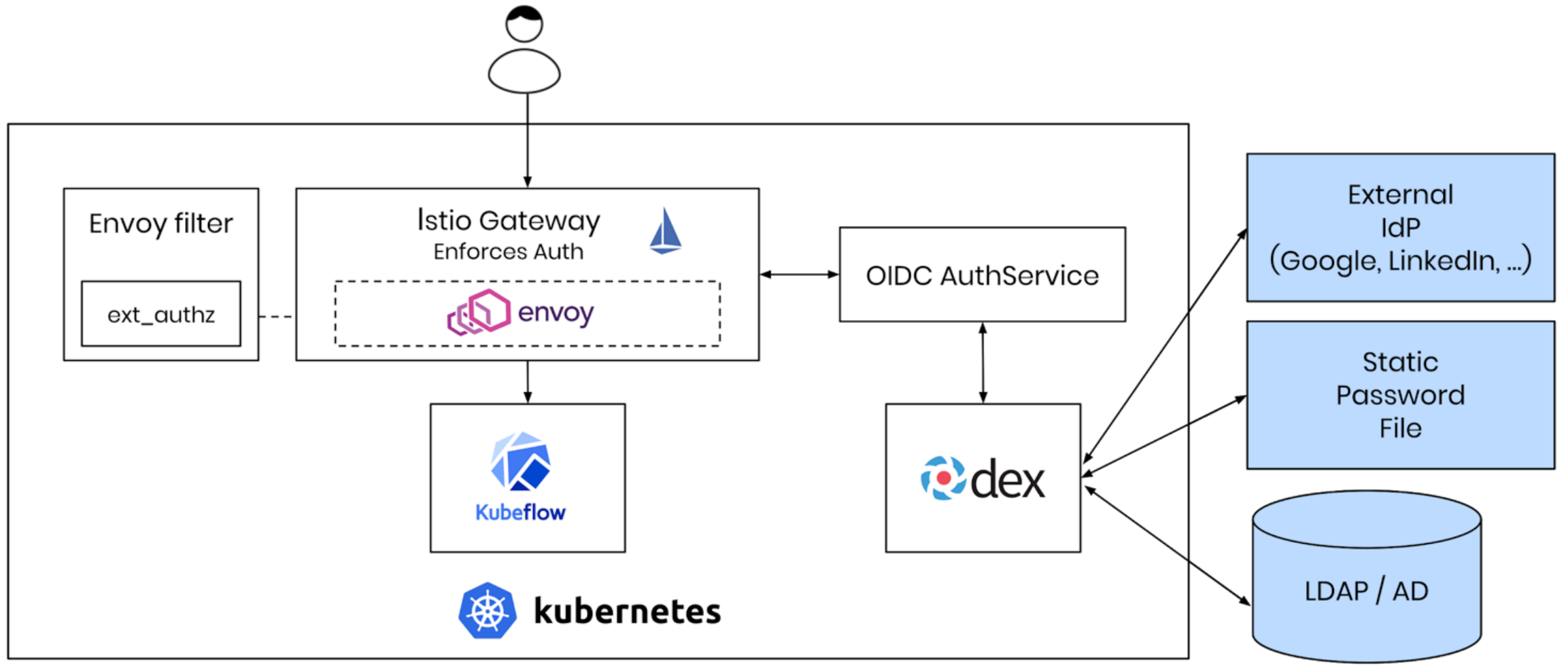
Kubeflow Pipeline Architecture



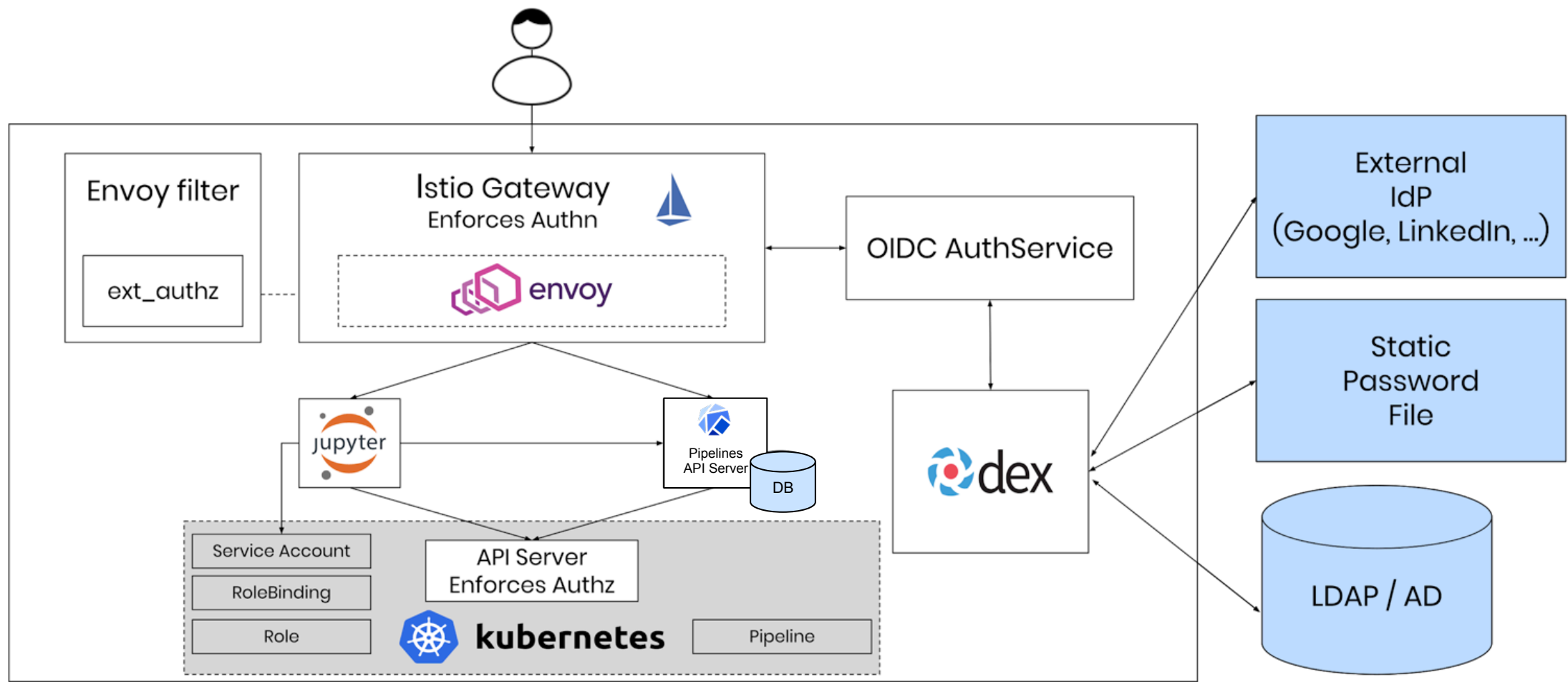
Kubeflow Pipeline Architecture



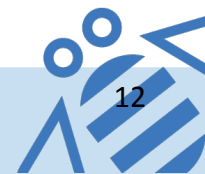
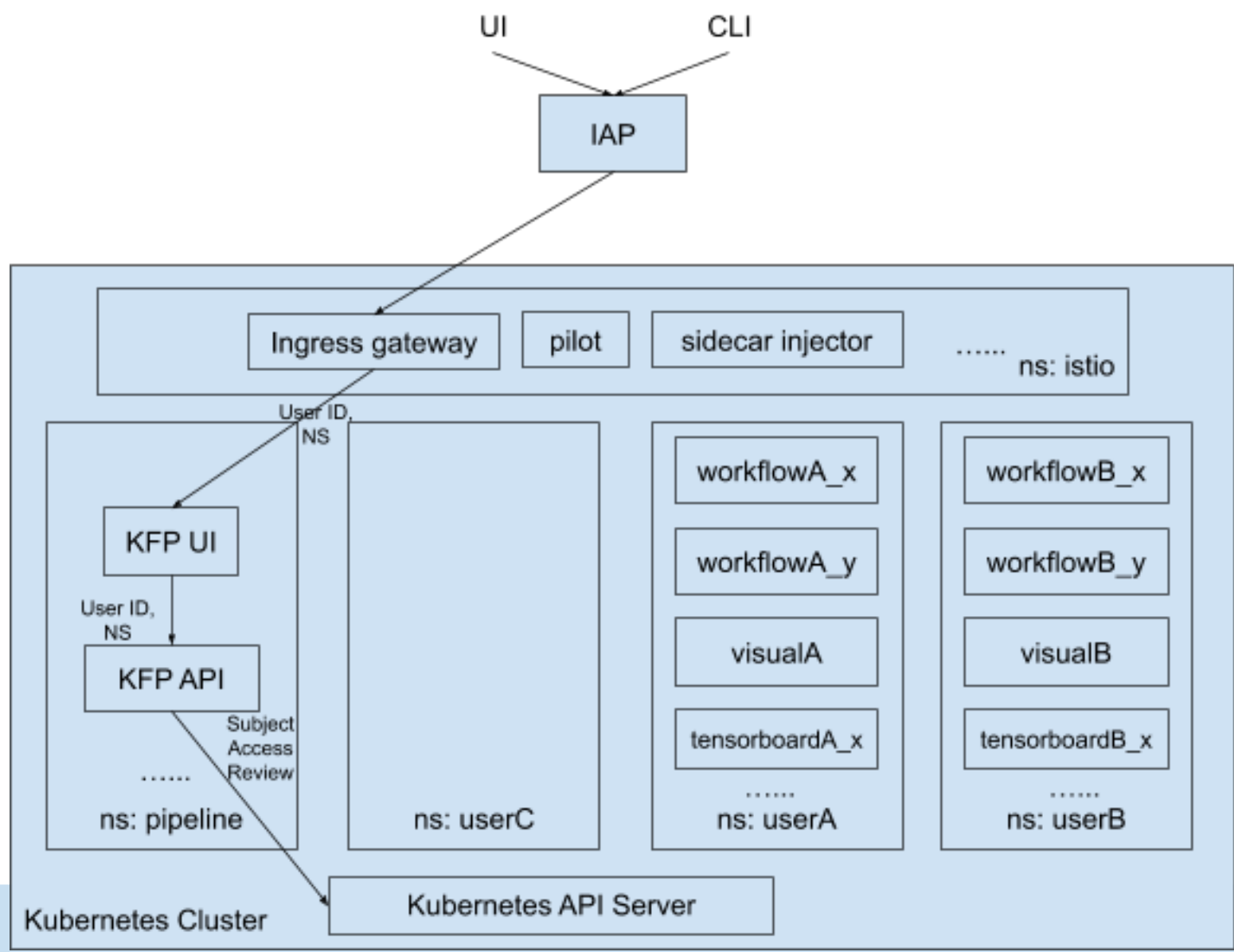
Multi User Kubeflow - Authentication: Istio + Dex



Multi User Pipelines: Overview



Multi User Pipelines: Resource Isolation



IBM Elyra-Notebook based Kubeflow Pipelines

<https://github.com/elyra-ai/elyra>



Kubeflow

File Edit View Run Kernel Git Tabs Settings Help

Launcher x untitled.pipeline

demo-pipelines /

Name	Last Modified
generated-pipelines	3 months ago
outputs	3 months ago
demo.pipeline	4 months ago
generate-contributions.i...	3 months ago
generate-stats.ipynb	3 months ago
kfp-operator.ipynb	6 days ago
overview_with_kfp.ipynb	3 months ago
overview_with_papermill...	3 months ago
overview_with_run.ipynb	3 months ago
overview.ipynb	3 months ago
untitled.pipeline	2 minutes ago

```
graph LR; A[generate-contributions.i...] --> B[overview]
```

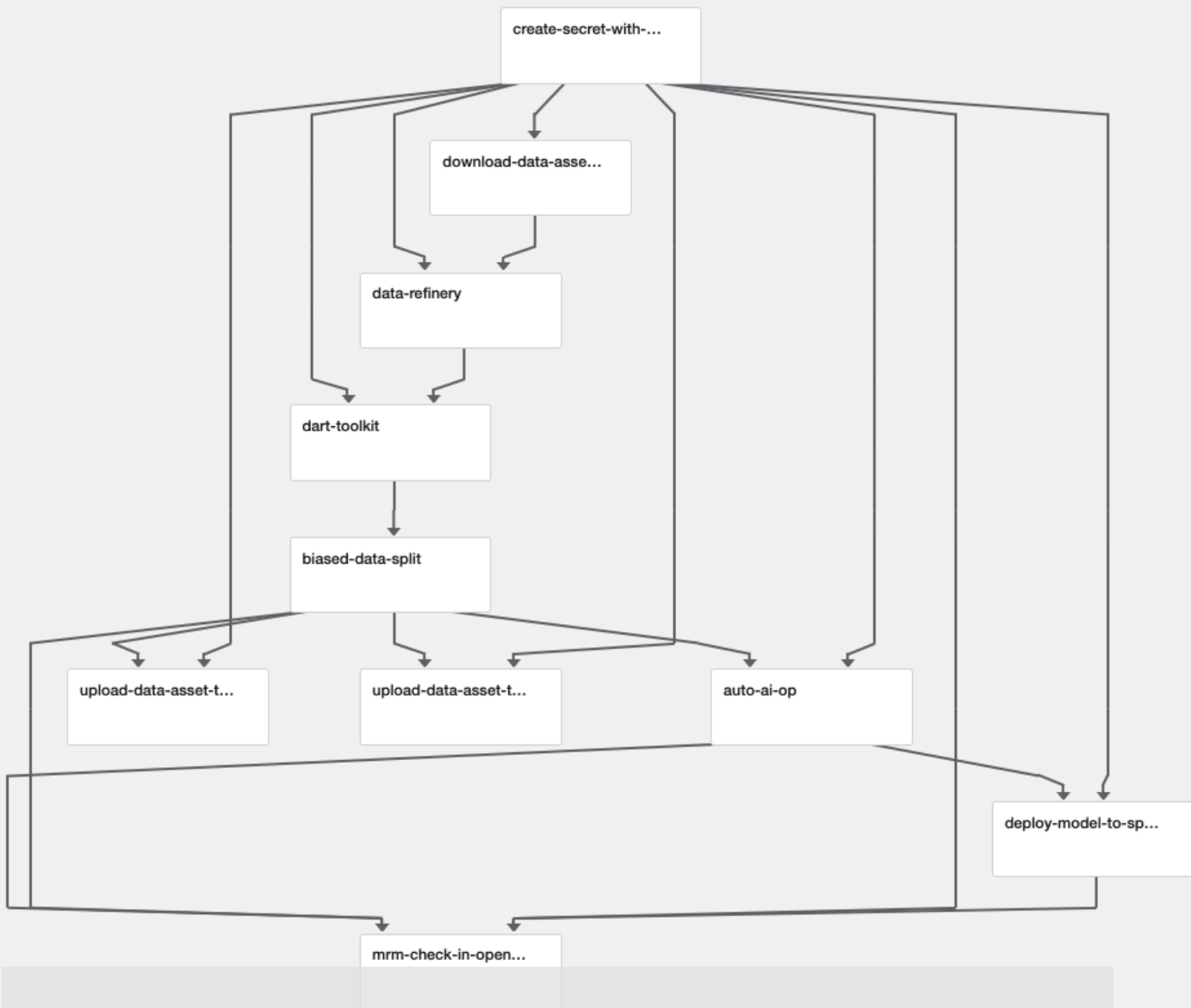
The image shows a screenshot of the Elyra Notebook interface. On the left, a file explorer displays a directory named 'demo-pipelines' containing various files and folders. The file 'generate-stats.ipynb' is highlighted. The main workspace shows a Kubeflow Pipeline diagram with two nodes: 'generate-contributions.i...' and 'overview', connected by a flow arrow. The interface includes a menu bar at the top with options like File, Edit, View, Run, Kernel, Git, Tabs, Settings, and Help. A toolbar below the menu bar contains icons for running, saving, deleting, and other actions. The pipeline diagram is displayed in a central area with a toolbar above it.

Watson AI Pipelines: Goals

- Demonstrate that Watson can be used for end-end AI lifecycle data prep/model training/model risk validation/model deployment/monitoring/updating models
- Demonstrate that the full lifecycle can be operated programmatically, and have **Tekton** as a backend instead of Argo

```
19 Requirement already satisfied: numpy in /opt/app-root/lib/ovh/conda/6/kite-packages (from line=>0.1.1.33) (1.16.0)
20 Requirement already satisfied: joblib>=0.11 in /opt/app-root/lib/ovh/conda/6/kite-packages (from line=>0.1.1.33) (0.13.2)
21 Requirement already satisfied: python-dateutil in /opt/app-root/lib/ovh/conda/6/kite-packages (from line=>0.1.1.33) (2.8.1)
22 Requirement already satisfied: pyyaml>=2.0 in /opt/app-root/lib/ovh/conda/6/kite-packages (from line=>0.1.1.33) (5.4.1)
23 Requirement already satisfied: six>=1.10 in /opt/app-root/lib/ovh/conda/6/kite-packages (from line=>0.1.1.33) (1.16.0)
24 Requirement already satisfied: pytz in /opt/app-root/lib/ovh/conda/6/kite-packages (from line=>0.1.1.33) (2019.3)
25 Requirement already satisfied: cyclotron>=0.10 in /opt/app-root/lib/ovh/conda/6/kite-packages (from line=>0.1.1.33) (0.10.0)
26 Requirement already satisfied: PyWavelets>=0.4 in /opt/app-root/lib/ovh/conda/6/kite-packages (from line=>0.1.1.33) (1.1.1)
27 Requirement already satisfied: networkx>=2.0 in /opt/app-root/lib/ovh/conda/6/kite-packages (from line=>0.1.1.33) (2.6.3)
28 Requirement already satisfied: pillow>=3.0 in /opt/app-root/lib/ovh/conda/6/kite-packages (from line=>0.1.1.33) (8.1.0)
29 Requirement already satisfied: imageio>=2.3.0 in /opt/app-root/lib/ovh/conda/6/kite-packages (from line=>0.1.1.33) (2.10.3)
30 Requirement already satisfied: decorator>=4.3.0 in /opt/app-root/lib/ovh/conda/6/kite-packages (from line=>0.1.1.33) (4.4.2)
31 - drift...
32 enablement finished.
33 running nrm...
34 [{"entity": {"data_mart_id": "00000000-0000-0000-0000-000000000000"}, "name": "MM monitor", "instance": "4b451ccc-5dfb-42af-9926-8"}, {"entity": {"data_mart_id": "00000000-0000-0000-0000-000000000000"}, "name": "MM monitor", "instance": "4b451ccc-5dfb-42af-9926-8"}]
35 MM monitor instance: 4b451ccc-5dfb-42af-9926-8
36 Triggering MM computation with https://rcc.cod
37 Done triggering MM computation with nrm.monitor
38 running upload and evaluate for validation_test
39 10:40:05 upload_in_progress
40 10:40:12 upload_in_progress
41 10:40:31 upload_in_progress
42 10:40:31 upload_in_progress
43 10:40:31 upload_in_progress
44 10:40:31 upload_in_progress
45 10:40:31 upload_in_progress
46 running upload and evaluate for validation_test
47 10:40:56 upload_in_progress
48 10:41:12 running
49 10:41:21 running
50 10:41:43 running
51 10:41:55 finished
52 running upload and evaluate for validation_test
53 10:42:01 upload_in_progress
54 10:42:14 upload_in_progress
55 10:42:28 running
56 10:42:44 running
57 10:42:58 running
58 10:43:11 finished
59 running upload and evaluate for validation_test
60 10:43:17 upload_in_progress
61 10:43:29 upload_in_progress
62 10:43:44 running
63 10:43:59 running
64 10:44:11 running
65 10:44:25 running
66 10:44:37 finished
67 running upload and evaluate for validation_test
68 10:44:43 upload_in_progress
69 10:44:55 upload_in_progress
70 10:45:08 running
71 10:45:21 running
72 10:45:35 running
73 10:45:48 finished
74
```

Rank	Name	Algorithm	Accuracy (Optimized)	Enhancements	Build time
1	Pipeline 4	Gradient Boosting Classifier	0.807	HPO-1 FE HPO-2	00:01:48
2	Pipeline 3	Gradient Boosting Classifier	0.804	HPO-1 FE	00:04:19
3	Pipeline 2	Gradient Boosting Classifier	0.804	HPO-1	00:00:38
4	Pipeline 1	Gradient Boosting Classifier	0.802	None	00:00:07



Run details

Pipeline *

Train the model and monitor with OpenScale Choose

Pipeline Version *

Train the model and monitor with OpenScale Choose

Run name *

Run of Train the model and monitor with OpenScale (a28a6)

Description (optional)

This run will be associated with the following experiment

Experiment *

GCR-AutoAI-Experiment-1 Choose

Run Type

One-off Recurring

Run parameters

Specify parameters required by the pipeline

github_token

6fd86cff0394892e772cd84d43a9e2d7546b1576

ai_config_url

https://raw.githubusercontent.com/Al-Lifecycle-Poland/kubeflow-pipelines-credentials/master/config_cpd

catalog_name

DataCatalog

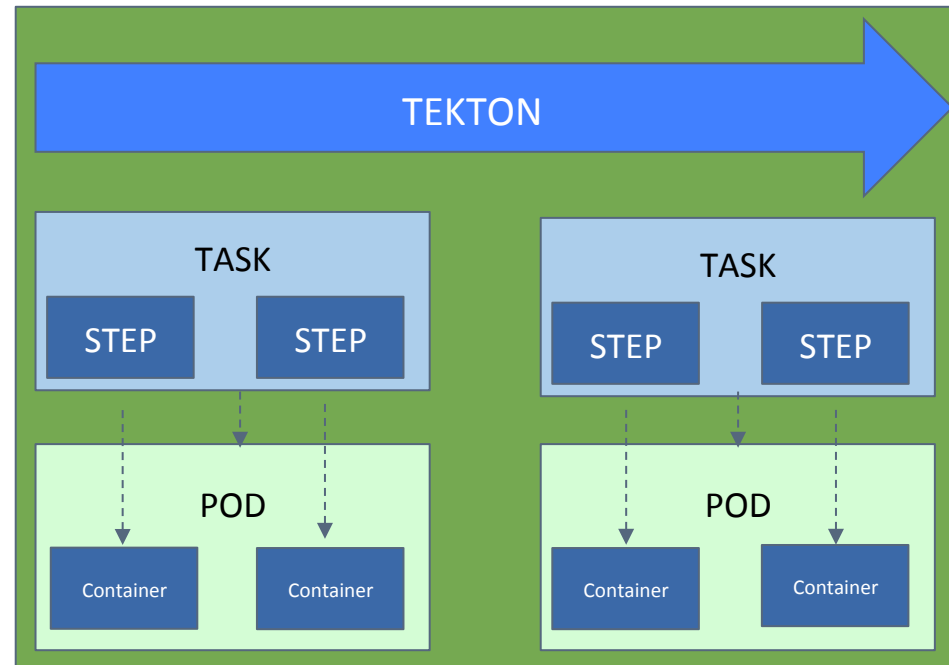
asset_id

2737bafc-3f78-4e2d-850a-e7f352b3d6b8

pre_production_space_uid

1dd2aaec-781a-4712-a7ff-ae1862cf7a84

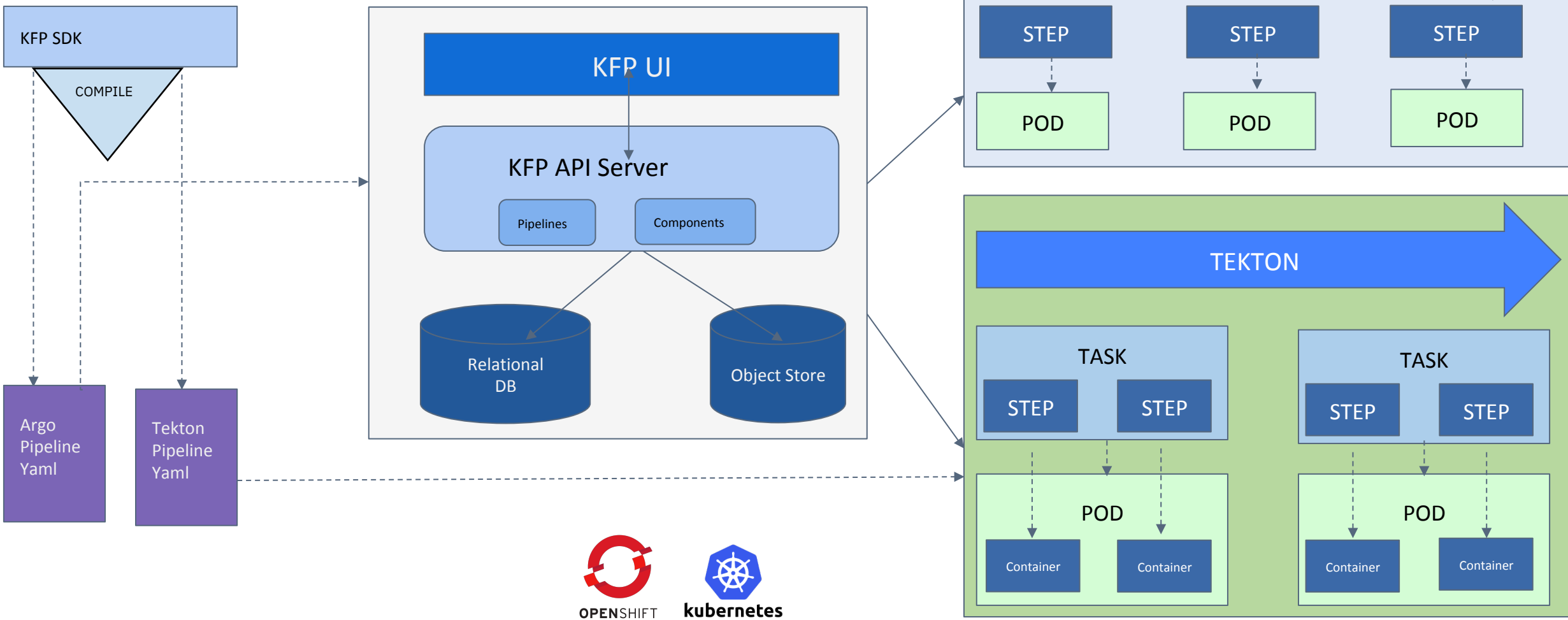
- ❑ The Tekton Pipelines project provides Kubernetes-style resources for declaring CI/CD-style pipelines.
- ❑ Tekton introduces several new CRDs including Task, Pipeline, TaskRun, and PipelineRun.
- ❑ A PipelineRun represents a single running instance of a Pipeline and is responsible for creating a Pod for each of its Tasks and as many containers within each Pod as it has Steps.



- ❑ A **PipelineResource** defines an object that is an input (such as a git repository) or an output (such as a docker image) of the pipeline.
- ❑ A **PipelineRun** defines an execution of a pipeline. It references the Pipeline to run and the PipelineResources to use as inputs and outputs.
- ❑ A **Pipeline** defines the set of Tasks that compose a pipeline.
- ❑ A **Task** defines a set of build Steps such as compiling code, running tests, and building and deploying images.



KFP – Tekton Phase One

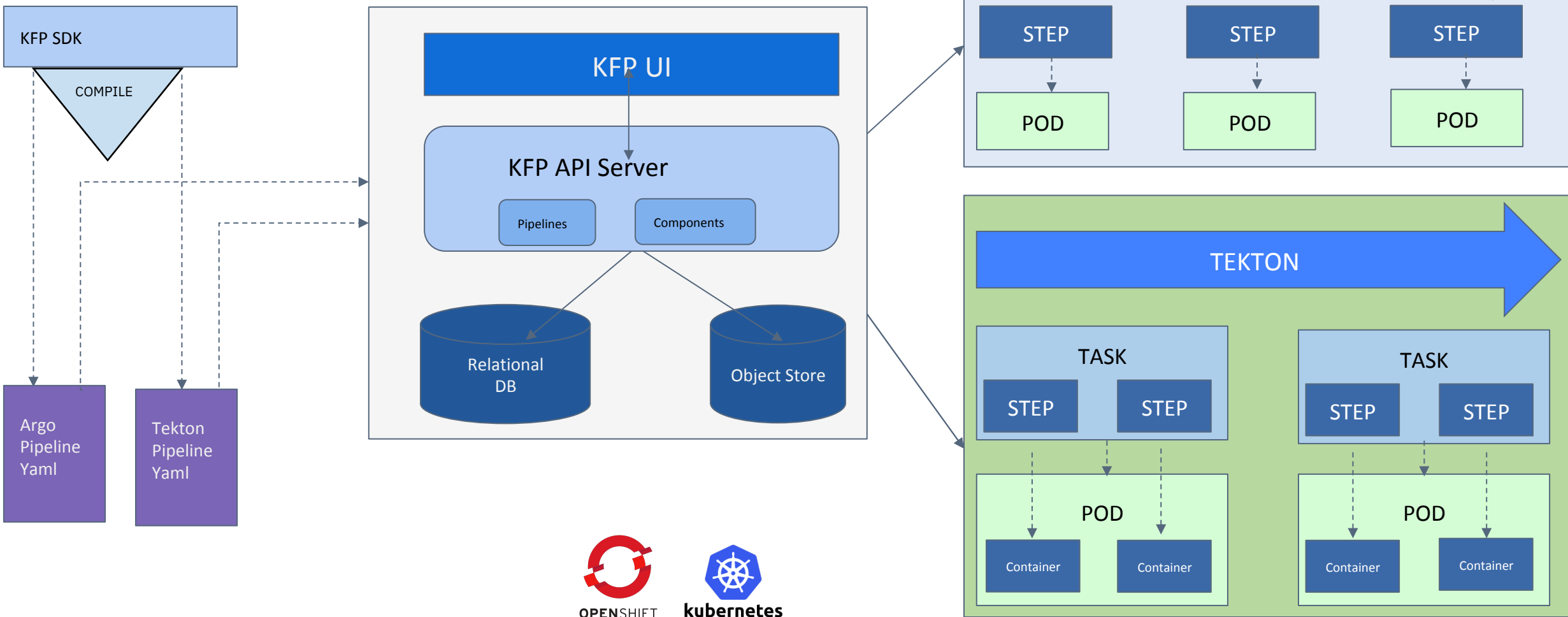


Pluggable Components

- Spark
- Watson Studio
- WML
- Open Scale
- Kubeflow Training
- Seldon
- AIF360
- ART
- KATIB
- KFSERVING
- ...
- ...



KFP – Tekton Phase Two



OPENSIFT



kubernetes

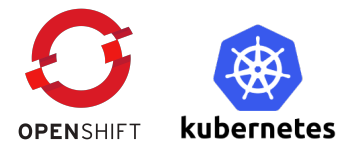
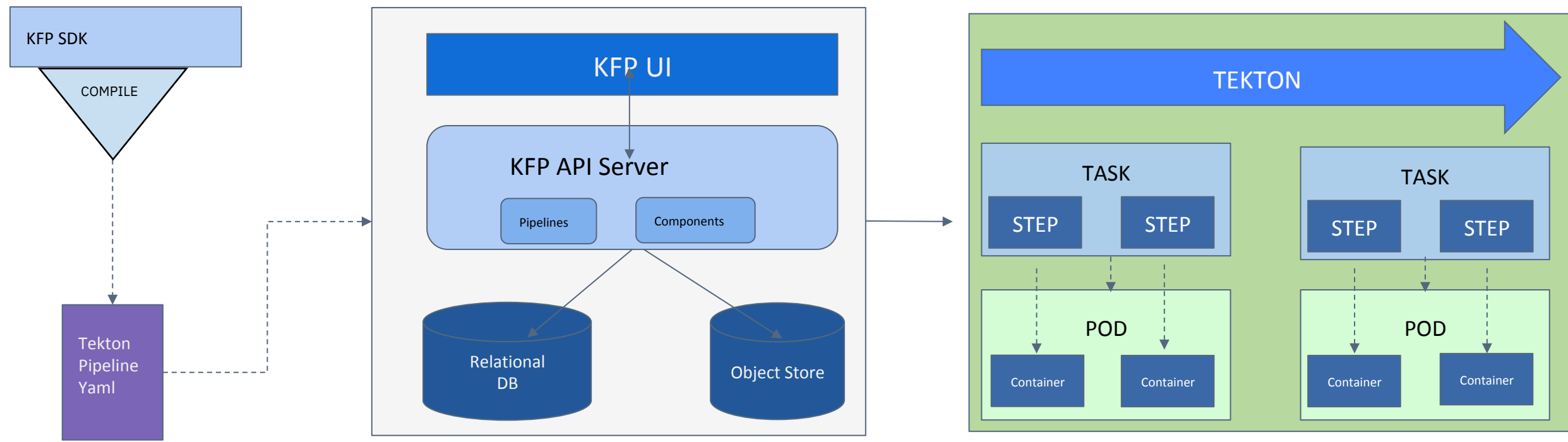


Pluggable Components

- Spark
- Watson Studio
- WML
- Open Scale
- Kubeflow Training
- Seldon
- AIF360
- ART
- KATIB
- KFSERVING
- ...
- ...



Kubeflow Pipelines with Tekton: Delivered



Pluggable Components

- Spark
- Watson Studio
- WML
- Open Scale
- Kubeflow Training
- Seldon
- AIF360
- ART
- KATIB
- KFSERVING
- ...
- ...



DSL features implemented



- Pipeline DSL features with native Tekton implementation
 - pod_annotations and pod_labels
 - Retries
 - Volumes
 - Timeout for Tasks and Pipelines
 - RunAfter
 - Input Parameters
 - ContainerOp
 - Affinity, Node Selector, and Tolerations
- Pipeline DSL features with custom Tekton implementation
 - Features with same behavior as Argo
 - InitContainers
 - Conditions
 - ResourceOp, VolumeOp, and VolumeSnapshotOp
 - Output Parameters
 - Input Artifacts
 - Output Artifacts
 - Features with limitations
 - ParallelFor - Tracking issue
 - Variable Substitutions - Tracking issue
 - ImagePullSecrets - Tracking issue
 - Features with different behavior than Argo
 - Sidecars - Tracking issue
- Pipeline features that are unavailable on Tekton
 - Exit Handler - Tracking PR



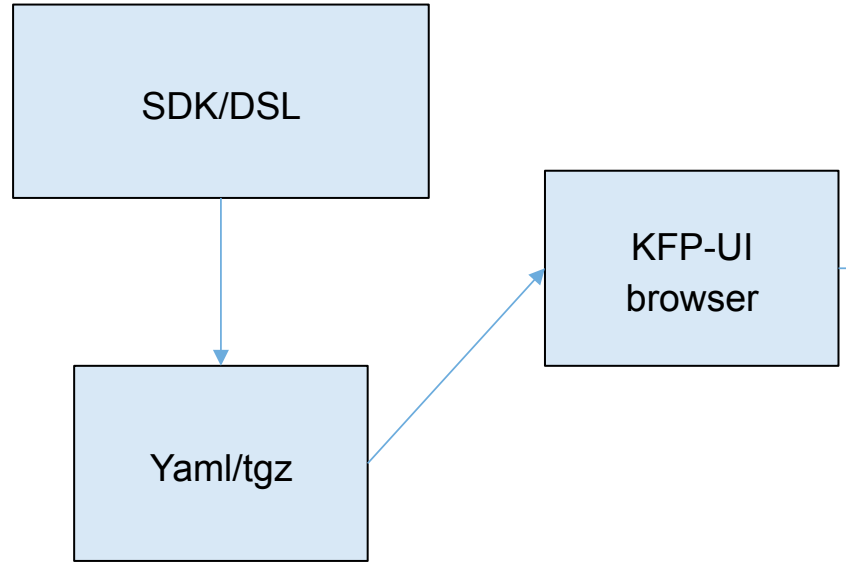
- MNIST End to End example with Kubeflow components
- Hyperparameter tuning using Katib
- Trusted AI Pipeline with AI Fairness 360 and Adversarial Robustness 360 components
- Training and Serving Models with Watson Machine Learning
- Lightweight python components example
- The flip-coin pipeline
- Nested pipeline example

<https://github.com/kubeflow/kfp-tekton/blob/master/samples/README.md>

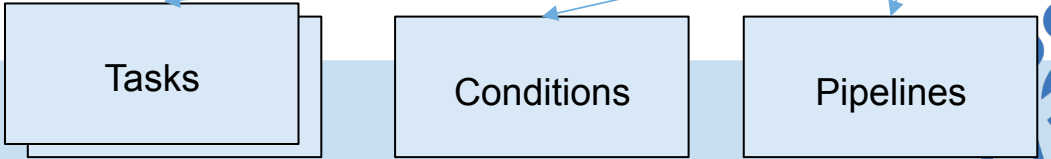
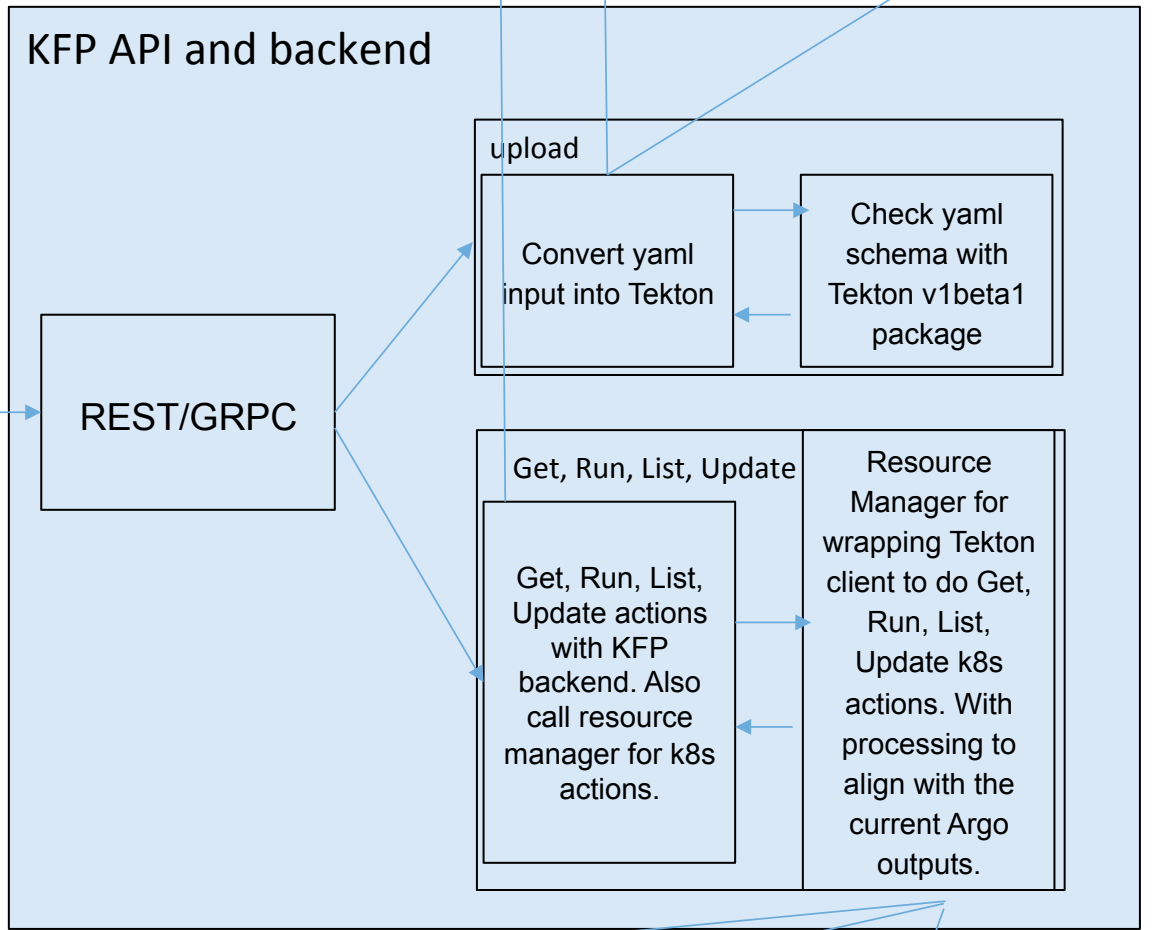
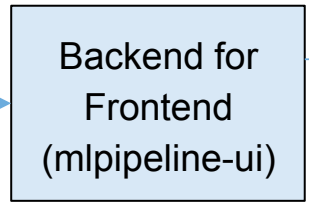




Local Machine



Server Machine





```
In [12]: kfp.Client(host="169.62.93.163").run_pipeline(experiment_id="74f7f363-96f8-487e-8632-4980b0971c7a",
job_name="sample-job",
pipeline_id="e684bc9e-cb30-4a3e-88f7-5c768202e6b7")
```

Run link [here](#)

```
Out[12]: {'created_at': datetime.datetime(2020, 5, 22, 0, 7, 46, tzinfo=tzutc()),
'description': None,
'error': None,
'finished_at': datetime.datetime(1970, 1, 1, 0, 0, tzinfo=tzutc()),
'id': '752ed34b-4ade-4654-b7d7-829618edd530',
'metrics': None,
'name': 'sample-job',
'pipeline_spec': {'parameters': None,
'pipeline_id': 'e684bc9e-cb30-4a3e-88f7-5c768202e6b7',
'pipeline_manifest': None,
'pipeline_name': 'tekton-parameters',
'workflow_manifest': '{"kind": "PipelineRun", "apiVersion": "tekton.dev/v1beta1", "metadata": {"name": "pipelinerun-with-taskspec-to-echo-message", "creationTimestamp": null}, "spec": {"pipelineSpec": {"tasks": [{"name": "echo-message", "taskSpec": {"params": [{"name": "MESSAGE", "type": "string", "default": "Hello World!"}], "steps": [{"name": "echo", "image": "ubuntu", "resources": {}}, "scrip
```

[+ Create run](#)
[+ Create experiment](#)
[Compare runs](#)
[Clone run](#)
[Archive](#)
[Refresh](#)

Experiments

[All experiments](#) [All runs](#)

Filter experiments

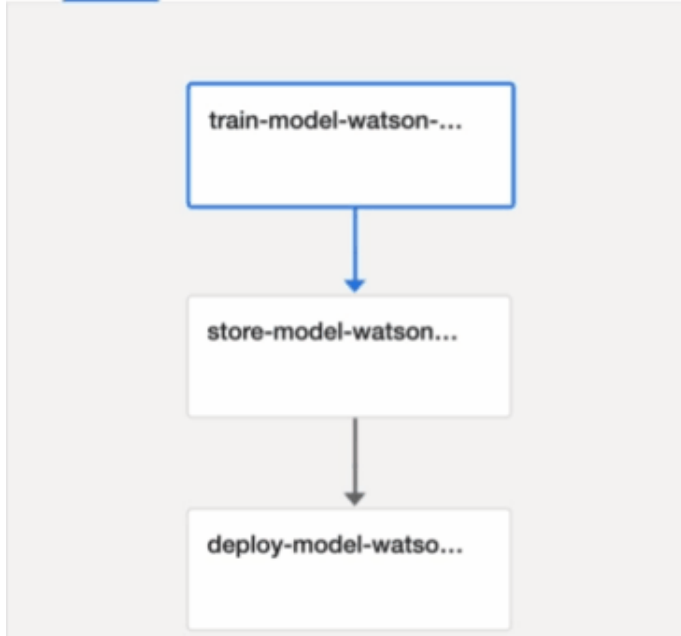
Experiment name	Description	Last 5 runs				
▼ Default	All runs created without specifying an experiment will be grouped here.					
<input type="checkbox"/> Run name	Status	Duration	Pipeline Version	Recurring Run	Start time	
<input type="checkbox"/> sample-job	?	-	tekton-parameters	-	5/21/2020, 5:07:46 PM	

Same KFP Experience: DAG, backed by Tekton YAML

Pipelines + Create run + Upload version + Create experiment Delete

← default-watson-ml (default-watson-ml)

Graph YAML



✕ train-model-watson-machine-learning

Input parameters

compute_name	
compute_nodes	
execution_command	
framework	
framework_version	
run_definition	
run_name	
runtime	
runtime_version	
train_code	

Output parameters

run-uid	/tmp/outputs/run_uid/data
training-uid	/tmp/outputs/training_uid/data

Arguments

Show summary Static pipeline graph

Same KFP Exp: Logs, Lineage Tracking and Artifact Tracking

Experiments > tekton-experiments

Retry Clone run Terminate Archive

← Run of watson-ml-pipeline-with-artifacts (d6bd5)

Graph Run output Config

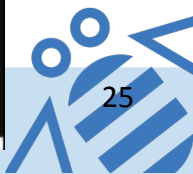
```
graph TD; A[create-secret-ku...] --> B[train-model-wats...]; B --> C[store-model-wats...]; C --> D[deploy-model-wa...]
```

Runtime execution graph. Only steps that are currently running are highlighted.

kfp-on-wml-training-run-1dd60-train-model-watson-machine--xt4gc

Input/Output Visualizations ML Metadata Volumes **Logs** Pod Events

```
9 #####
10
11
12
13
14 -----
15 Log monitor done.
16 -----
17
18
19
20
21 #####
22
23 Metric monitor started for training run: af80b10e-12f3-4053-a71c-31ff4ea8df56
24
25 #####
26
27
28
29
30 -----
31 Metric monitor done.
32 -----
33
34
35 status: {'state': 'pending'}
36 {'completed_at': '2020-07-06T21:15:15.208Z', 'message': {'text': 'Training job af80b10e-12f3-4053-a71c-31ff4ea
37 training_details {'metadata': {'created_at': '2020-07-06T21:11:38.049Z', 'guid': 'af80b10e-12f3-4053-a71c-31ff
38
```



End to end Kubeflow Components : With KFP-Tekton

Recurring run configs
0 active
[Manage](#)

Experiment description

Runs

+ Create run

+ Create recurring run

Compare runs

Clone run

Archive

Filter runs



<input type="checkbox"/>	Run name	Status	Duration	Pipeline Version	Recurring Run...	Start time ↓
<input type="checkbox"/>	Run of mnist-e2e-pipeline (7d2c8)	✓	-	mnist-e2e-pipeline	-	7/7/2020, 12:28:38 AM
<input type="checkbox"/>	Run of mnist-model-cleanup (91455)	✓	-	mnist-model-cleanup	-	7/6/2020, 5:27:54 PM
<input type="checkbox"/>	mnist-e2e-pipeline-animesh (bf69b)	✓	-	mnist-e2e-pipeline	-	7/6/2020, 4:48:15 PM
<input type="checkbox"/>	Run of watson-ml-pipeline-with-artifacts (d...	✓	-	watson-ml-pipeline-with-arti...	-	7/6/2020, 2:11:07 PM
<input type="checkbox"/>	Run of watson-ml-pipeline-with-artifacts (d...	✓	-	watson-ml-pipeline-with-arti...	-	6/22/2020, 6:21:28 PM
<input type="checkbox"/>	Watson-ml-pipeline-with-artifacts	✓	-	watson-ml-pipeline-with-arti...	-	6/14/2020, 7:15:30 PM
<input type="checkbox"/>	▲ Run of watson-ml-pipeline (f5876)	✓	-	-	-	6/11/2020, 4:23:45 PM
<input type="checkbox"/>		✓	-	-	-	6/2/2020, 5:19:25 PM



Compiled Pipelines on Tekton

Tekton

Tekton resources ^

- Pipelines
- PipelineRuns**
- PipelineResources
- Tasks
- ClusterTasks
- TaskRuns

Namespace

All Namespaces × ▾

About

Import Tekton resources

Secrets

ServiceAccounts

PipelineRuns

🔍 Input a label filter of the format labelKey:labelValue

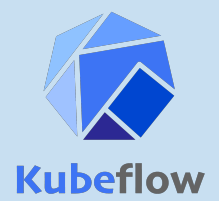
Create +

Status	Name	Pipeline	Namespace	Created	Duration	
✓	kfp-on-wml-training-run-...	kfp-on-wml-training	default	20 hours ago	6 minutes 23 seconds	⋮
✓	launch-trusted-ai-pipelin...	launch-trusted-ai-pipeline	anonymous	2 days ago	9 minutes 3 seconds	⋮
✓	conditional-execution-pip...	conditional-execution-pip...	default	2 days ago	52 seconds	⋮
✓	end-to-end-pipeline-run	end-to-end-pipeline	anonymous	2 days ago	14 minutes 41 seconds	⋮





Running Pipelines on Tekton



- Tekton
- Tekton resources
 - Pipelines
 - PipelineRuns**
 - PipelineResources
 - Tasks
 - ClusterTasks
 - TaskRuns
- Namespace
 - default

kfp-on-wml-training-run-p7n6f 20 hours ago

Rerun

Succeeded Tasks Completed: 4, Skipped: 0

- create-secret-kubernete...
- train-model-watson-mac...
- train-mode... **Completed**
- store-model-watson-ma...
- deploy-model-watson-m...

train-model-watson-machine-learning **Completed**

Logs Status Details

```

training_id {'metadata': {'created_at': '2020-05-07T23:57:46.868Z', 'guid': 'b200eef4-3dde-4b4e-a521-fe751735932c'},
get status {'state': 'running'}}

#####

Log monitor started for training run: b200eef4-3dde-4b4e-a521-fe751735932c

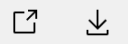
#####

-----
Log monitor done.
-----

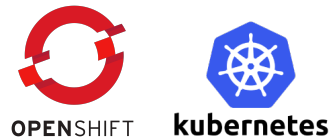
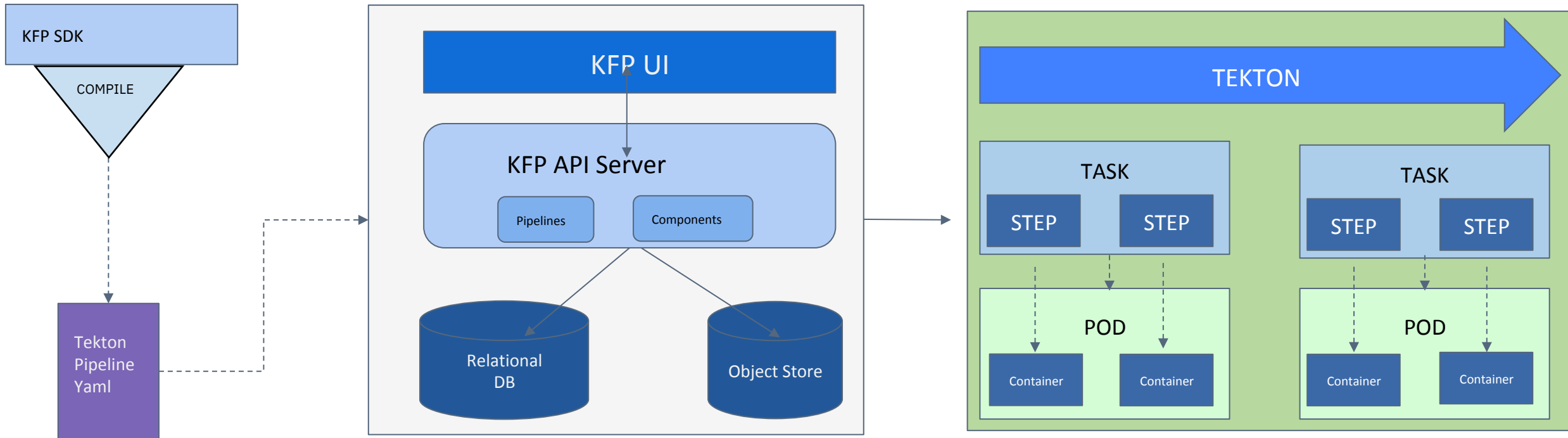
#####

Metric monitor started for training run: b200eef4-3dde-4b4e-a521-fe751735932c

```



Kubeflow Pipelines with Tekton: Delivered



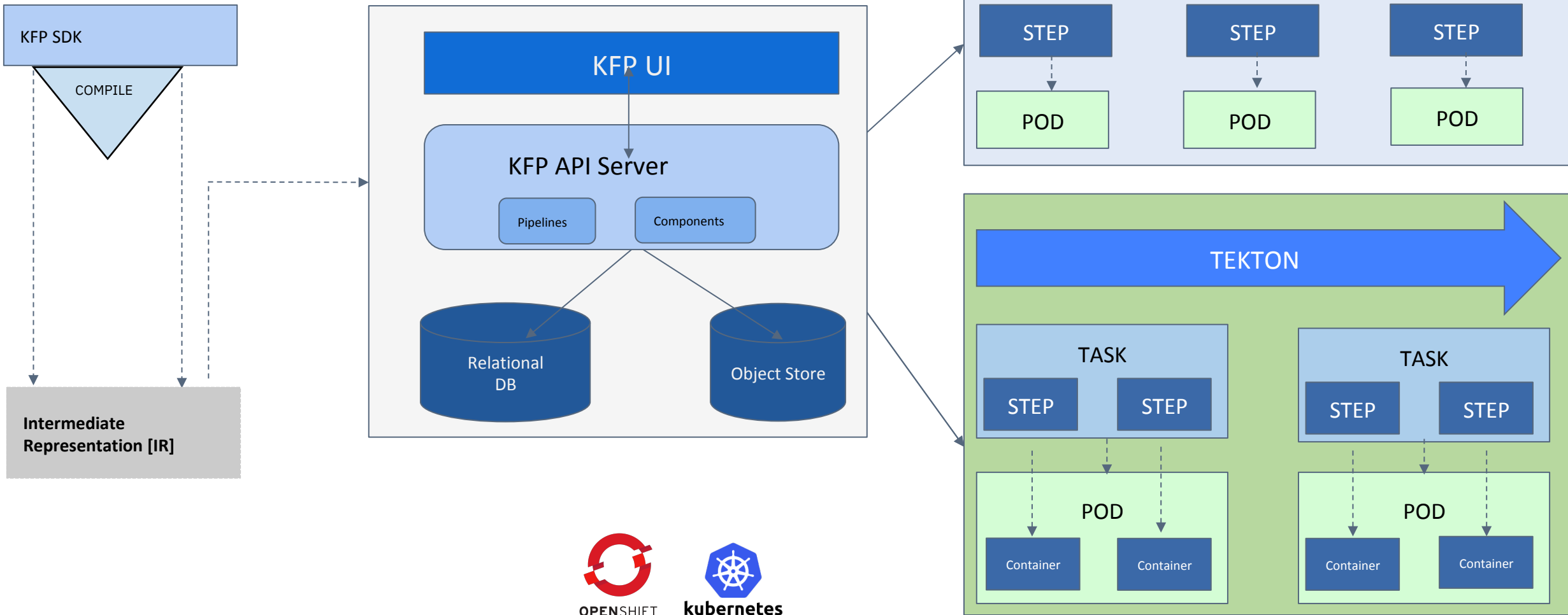
Pluggable Components

- Spark
- Watson Studio
- WML
- Open Scale
- Kubeflow Training
- Seldon
- AIF360
- ART
- KATIB
- KFSERVING
- ...
- ...





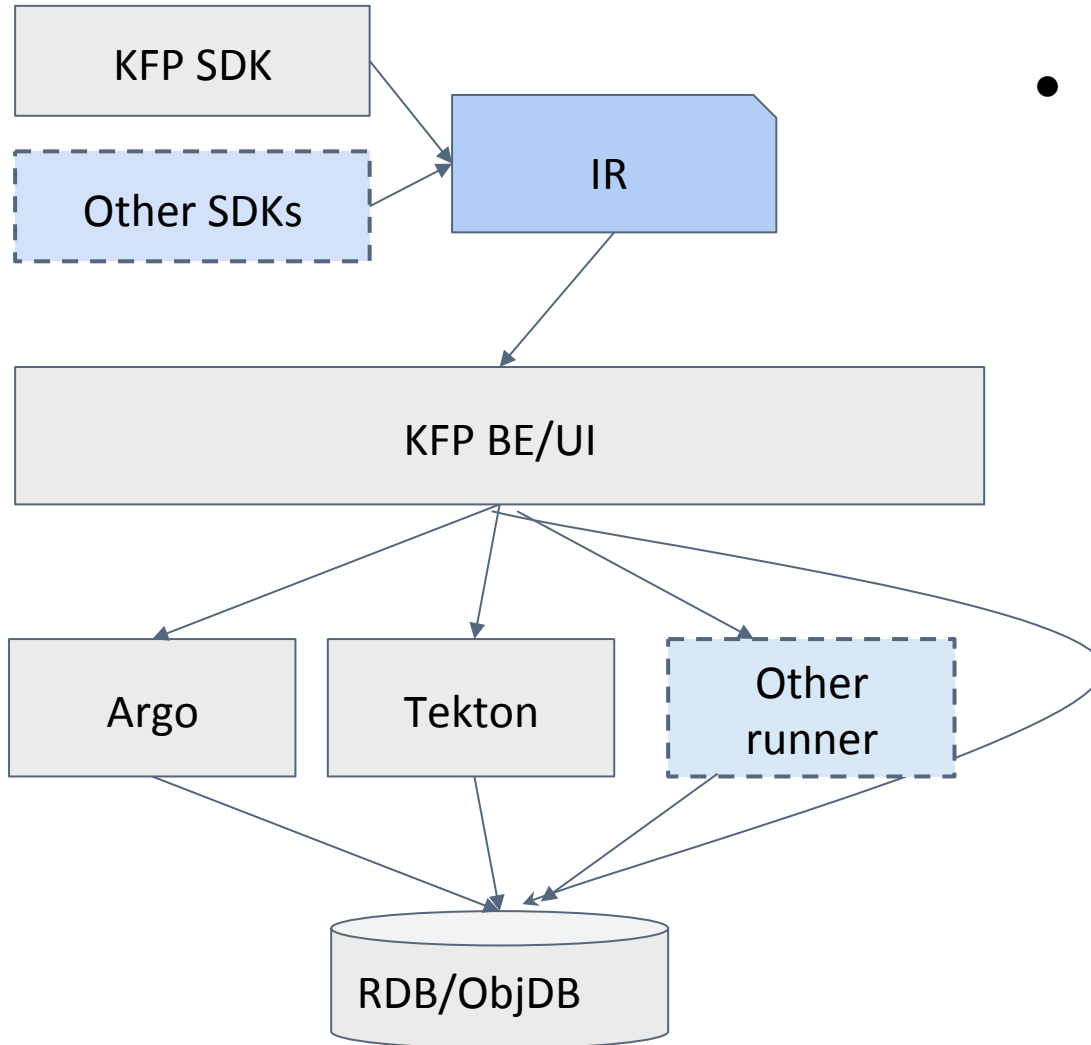
Future: KFP – Tekton Phase Three



Pluggable Components

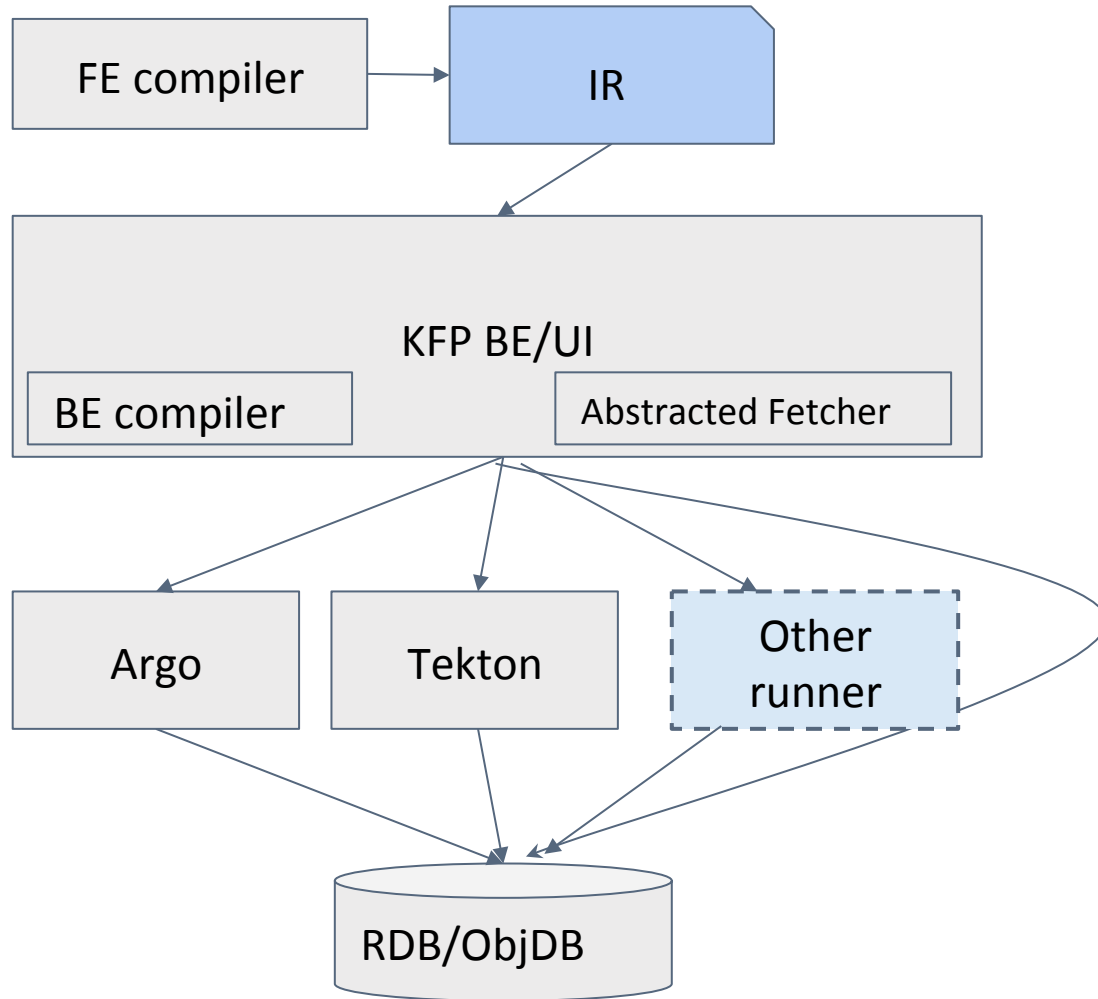
- Spark
- Watson Studio
- WML
- Open Scale
- Kubeflow Training
- Seldon
- AIF360
- ART
- KATIB
- KFSERVING
- ...
- ...





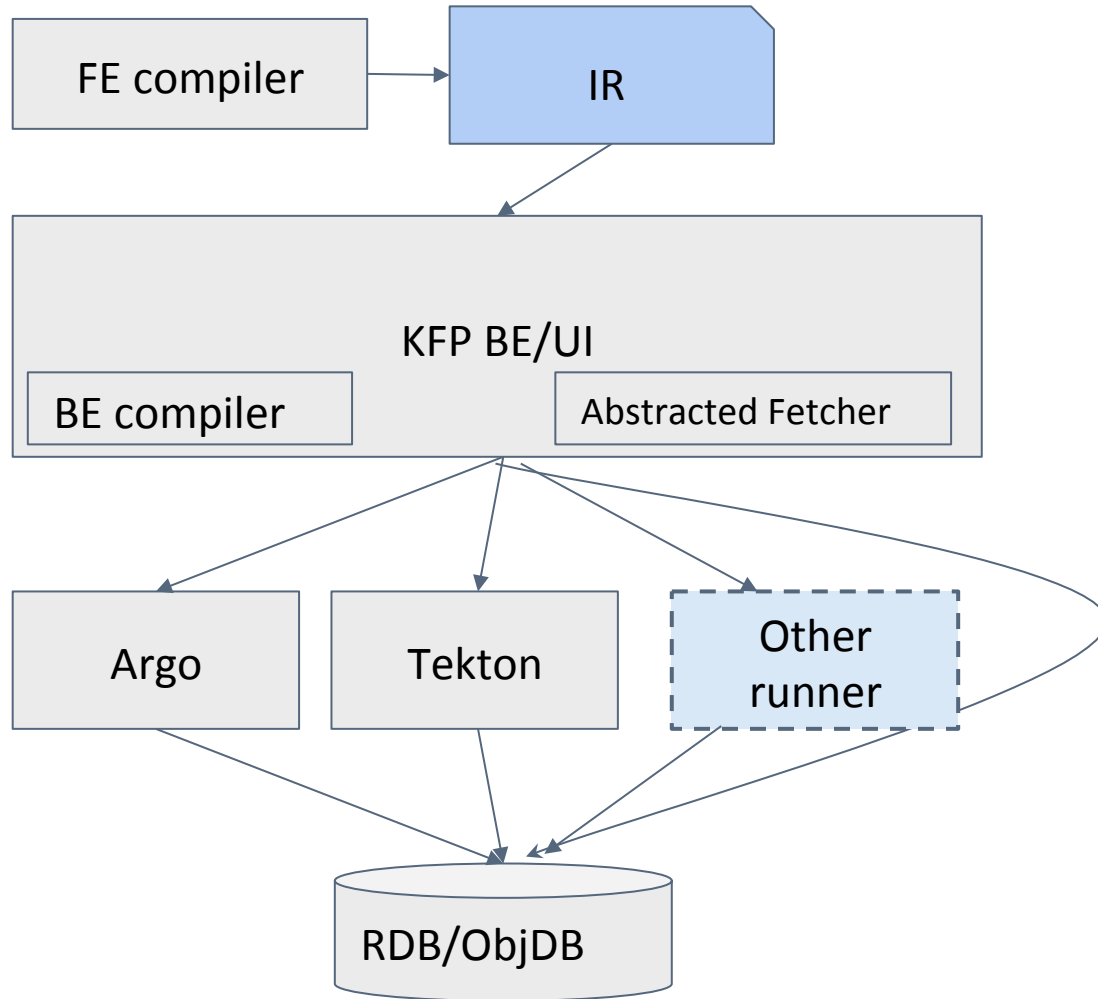
- IR
 - Core part
 - platform-agnostic
 - semantic representation
 - Extension part
 - Specific runner configs
 - K8s, Docker, PythonClass
 - Special
 - Lives separately from KFP repo
 - KFP: the k8s impl supports this IR





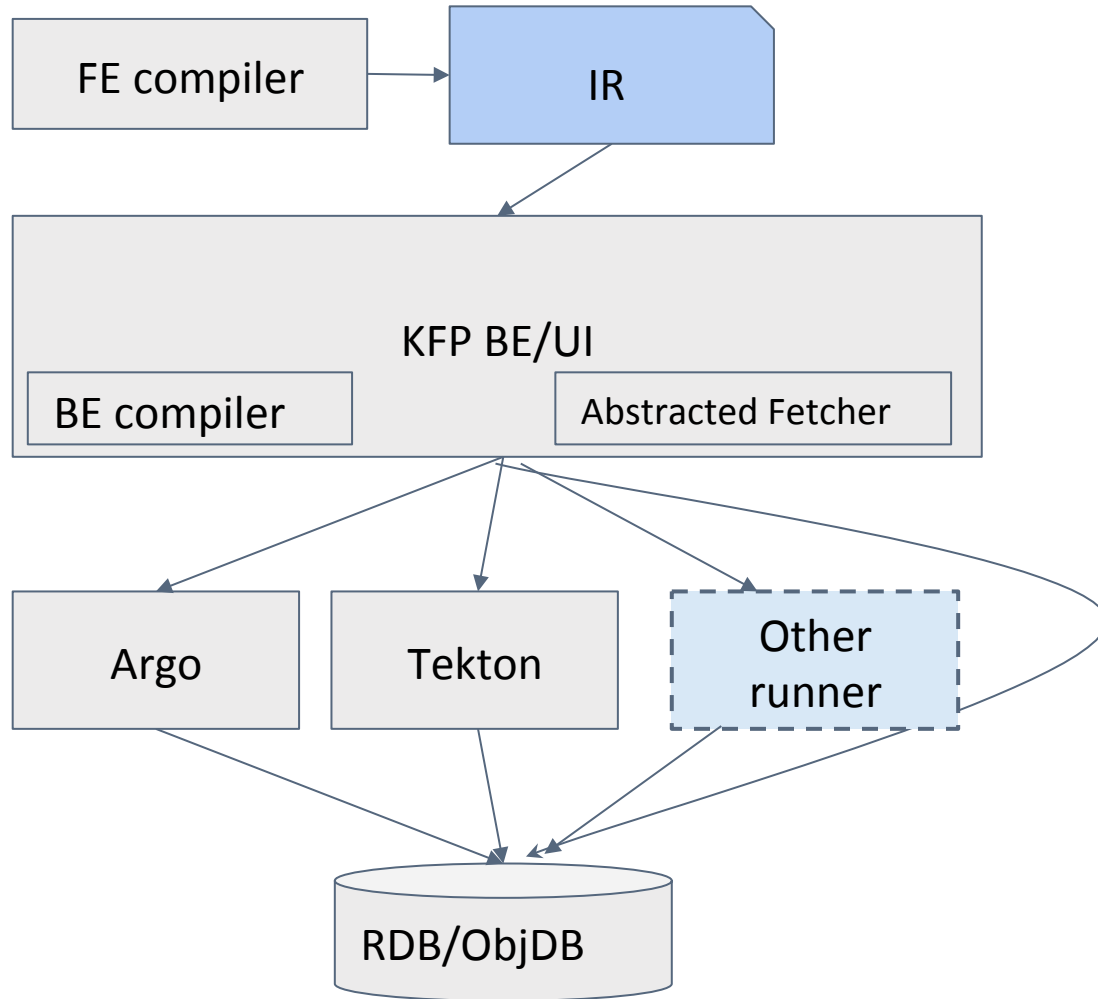
- SDK / Compiler
 - FE compiler
 - codes => IR
 - BE compiler
 - IR => runner-spec
 - Possible other runner
- BE/FE
 - Common data fetching
 - RDB/ObjDB
 - Specific data fetching
 - BE interface for extension
 - Decouple FE with runner





- IR Conformance
 - Pass IR Conformance Test / Validation
- KFP Conformance
 - BE compiler for specific runner
 - Data fetcher impl for specific runner
 - E2e test cases



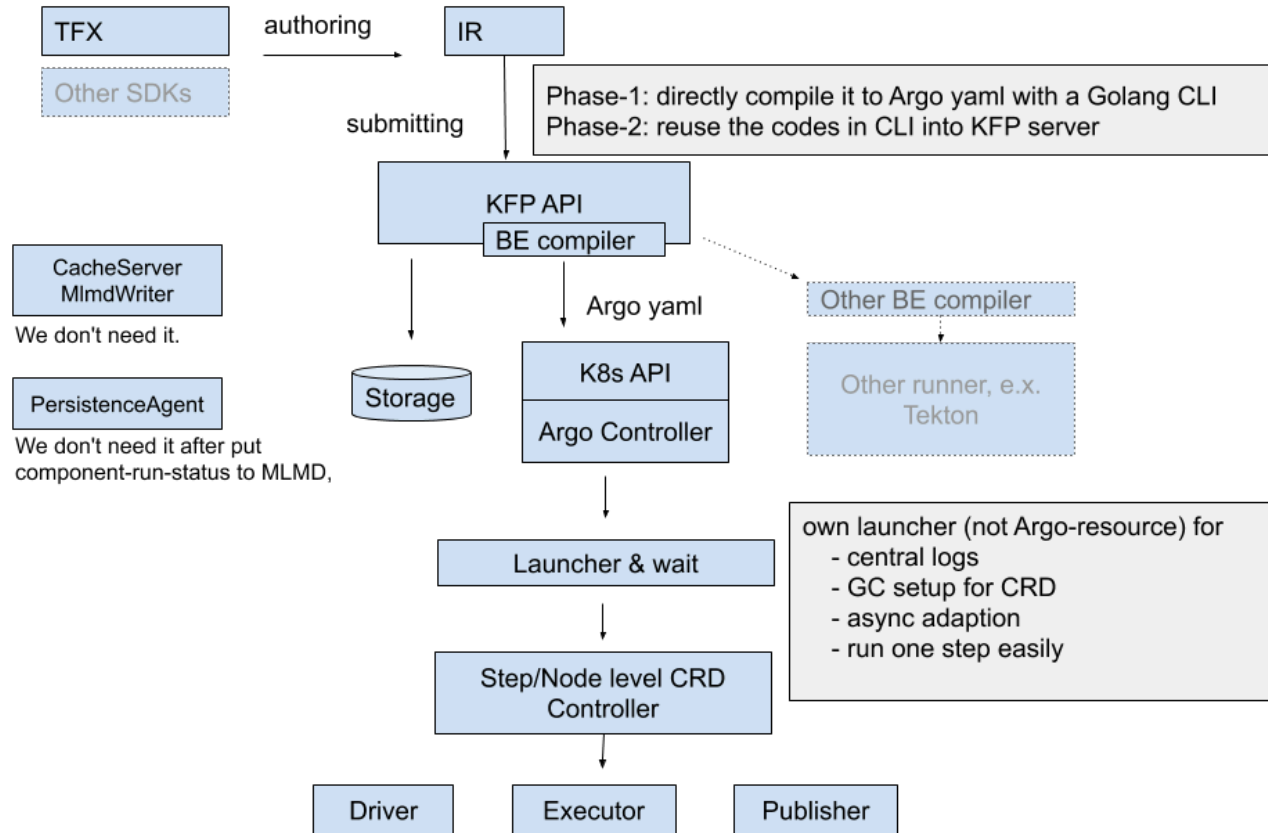


- IR schema public v1 - **Q3**
- IR based KFP - **Q3**
 - TFX SDK as frontend compiler
 - KFP API which accepts IR (design)
 - BE/UI refactoring (design)
 - UI changes & Data fetcher
 - IR storage schema / data access
 - BE compiler placeholder
 - One step/node runner supports
 - Native MLMD
 - Custom Container
- KFP e2e runnable with IR (KFP) - **Q4**
 - Finish refactoring
 - FE compiler ready with IR
 - Backend compiler in KFP BE ready with Argo
 - BE/UI can support IR



1. [TFX SDK](#) is the long term option
 2. [ml-metadata](#) is the long term option
- Support native MLMD
 - Bring TFX Driver, Executor & Publisher concepts to KFP server
 - MLMD is used to track inputs/outputs etc. and for visualization decoupled with data producer
 - Support custom container
 - Container/K8s is 1st-class concept in IR
 - Current
 - KFP SDK support it well
 - TFX-KubeflowDagRunner only can run TFX based image





(1) supports native [ml-metadata](#)

(2) supports arbitrary container images
args is placeholder and will be replaced in runtime
guided by [ml-metadata](#)





OpenAIHub

Pipelines

Components

Models

Notebooks

Operators

Workspace

Made with Gifox

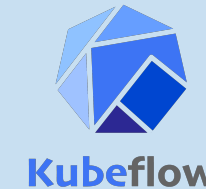
Stay Tuned!

Models

Machine learning models that can be used in your pipelines.

[VIEW ALL MODELS](#)

[UPLOAD A MODEL](#)



OpenAIHub

Pipelines

Components

Models

Notebooks



OpenAIHub

Pipelines

Components

Models

Notebooks

Operators

Workspace

Made with Gifox

Notebooks

Notebooks for your data science tasks.

[VIEW ALL NOTEBOOKS](#)

[UPLOAD A NOTEBOOK](#)

[PIPELINE CREATOR](#)

Pipelines

Pipelines for your machine learning workloads.

[VIEW ALL PIPELINES](#)

[UPLOAD A PIPELINE](#)



OpenAIHub

Pipelines

Components

Models

Notebooks

Components

Components that can be used to build your pipelines.

[VIEW ALL COMPONENTS](#)

[UPLOAD A COMPONENT](#)

[PIPELINE CREATOR](#)

Deploy Model - Knative

Deploy AI models using Knative serving.

OpenSource

Deploy Model - Kubernetes

Deploy AI models using Kubernetes deployment.

OpenSource

Deploy Model - Watson Machine Learning

Deploy stored model on Watson Machine Learning as a web service.

IBM Watson Machine Learning

Jupyter

Runs a jupyter notebook and saves to object storage.

OpenSource

Model Fairness Check - PyTorch

Perform a fairness check on a certain attribute using AIF360 to make sure the model is fair and ethical.

OpenSource

Model Robustness Check - PyTorch

Perform a robustness check using fast gradient attack with ART to make sure the model is robust against simple gradient changes.

OpenSource

Serve PyTorch Model - Seldon Core

Serve PyTorch Models remotely as web service using Seldon Core.

OpenSource

Store model - Watson Machine Learning

Store and persistent trained model on Watson Machine Learning.

IBM Watson Machine Learning

Main Open Source Github Repository:

<https://github.com/kubeflow/kfp-tekton>

IBM internal Slack channels

#kfp-tekton

#kubeflow

The Kubeflow external Slack workspace is

kubeflow.slack.com

To join, click here

https://join.slack.com/t/kubeflow/shared_invite/zt-cpr020z4-PfcAue_2nw67~iIDy7maAQ

