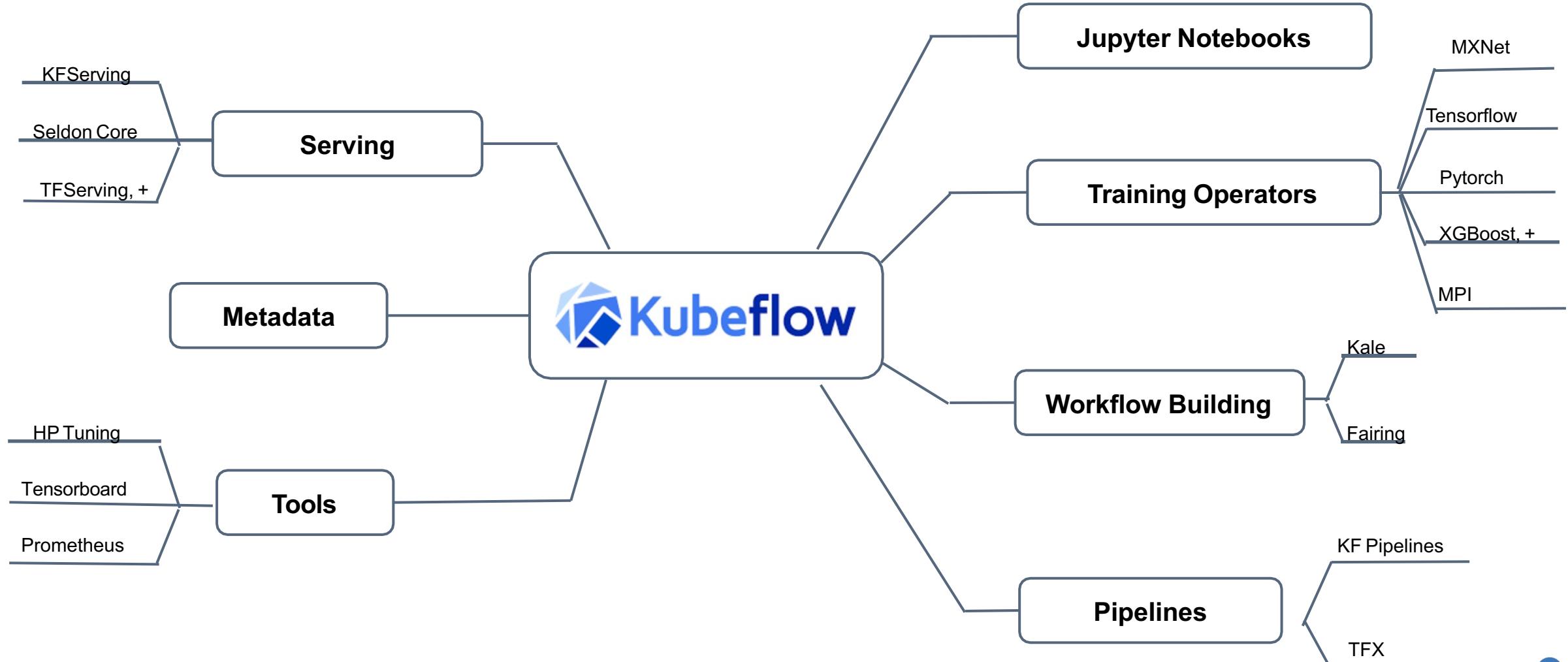
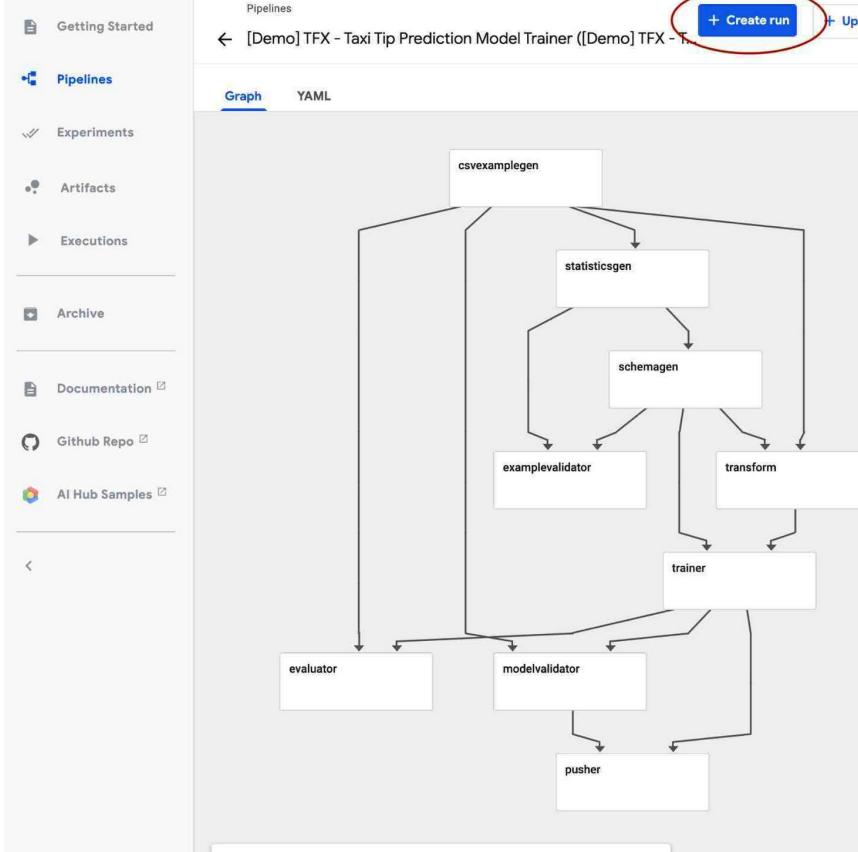


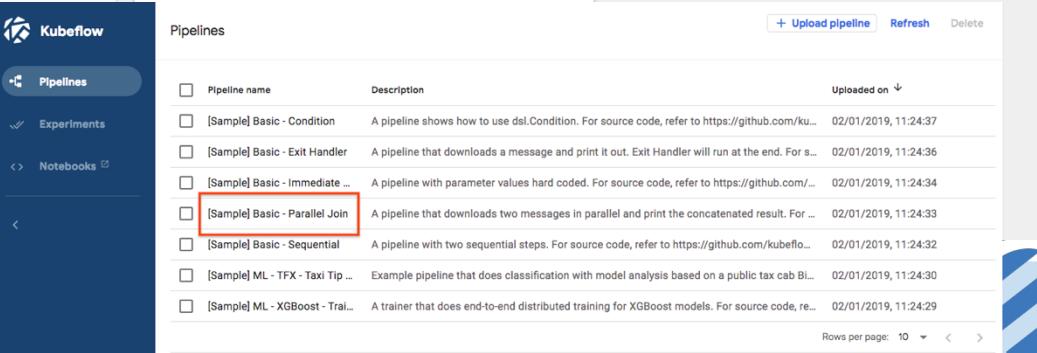
Tommy Li, Feng Li



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

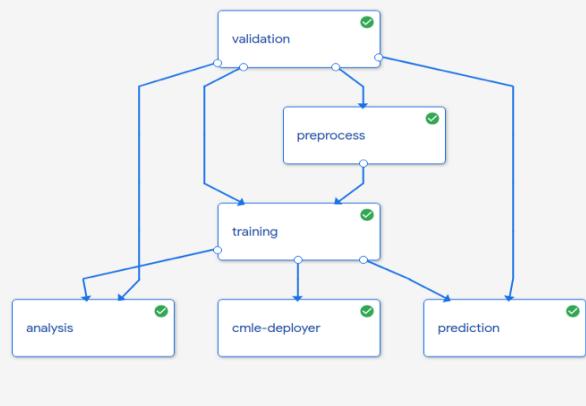




Define Pipeline with Python SDK

```
@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

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

Cluster name: cluster-4
Build commit: 743746b
Report an Issue

Graph Run output Config

csvexamplegen → statisticsgen → schemagen → examplevalidator → evaluator → pusher

resolvernode-lates... → evaluator

train → evaluator

Static HTML

Sort by Feature ▾ Reverse order Feature search (...)

Features: int(8) float(7) string(2)
 unknown(1)

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

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

Pipelines versioning

Pipelines

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

Filter pipelines



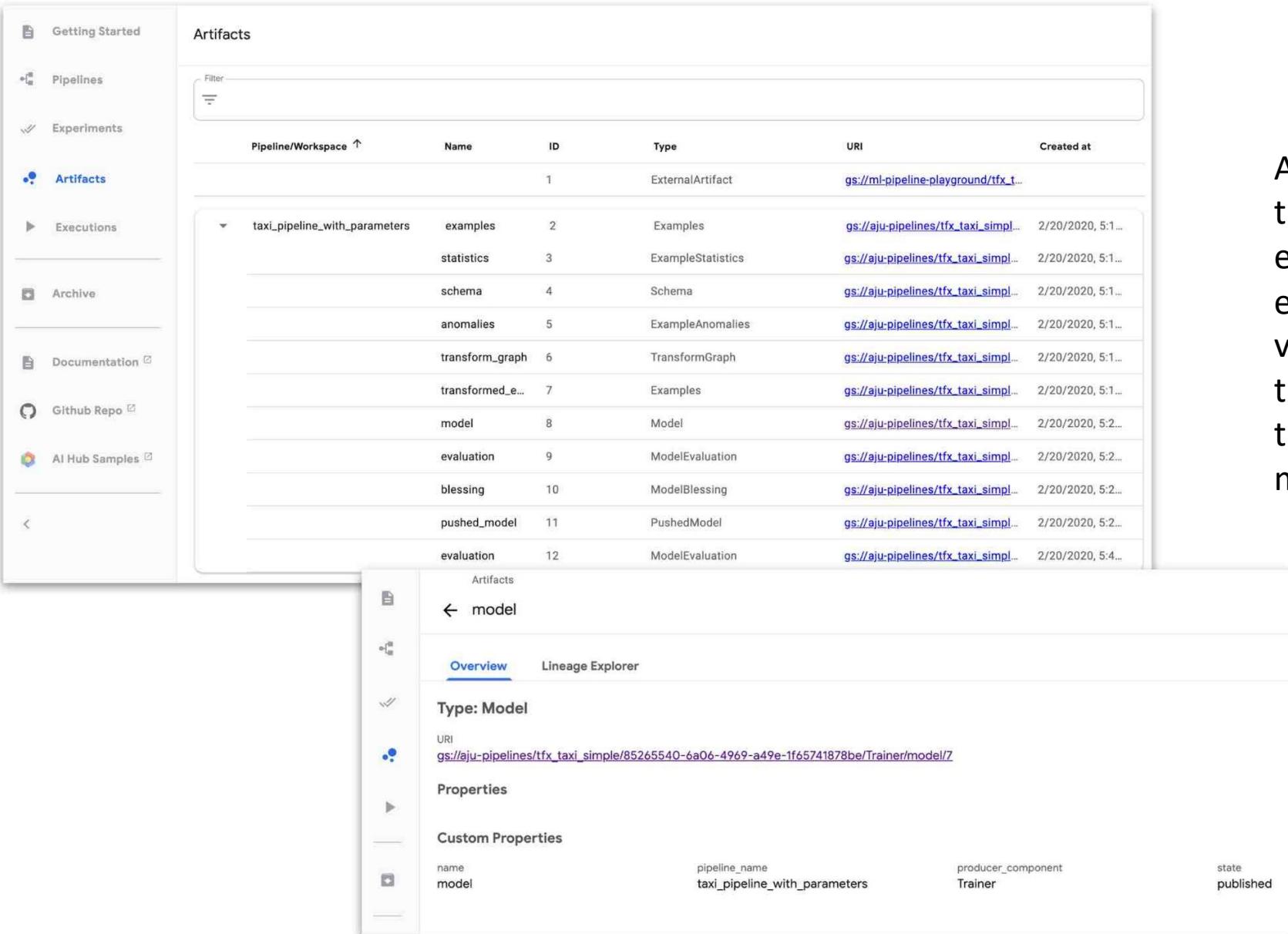
<input type="checkbox"/>	Pipeline name	Description	Uploaded on
<input type="checkbox"/>	[Tutorial] DSL - Control structures	source code Shows how to use conditional execution and exit handlers. This pipeline will randomly fail to demonstra...	2/20/2020, 3:28:12 PM
<input type="checkbox"/>	[Tutorial] 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



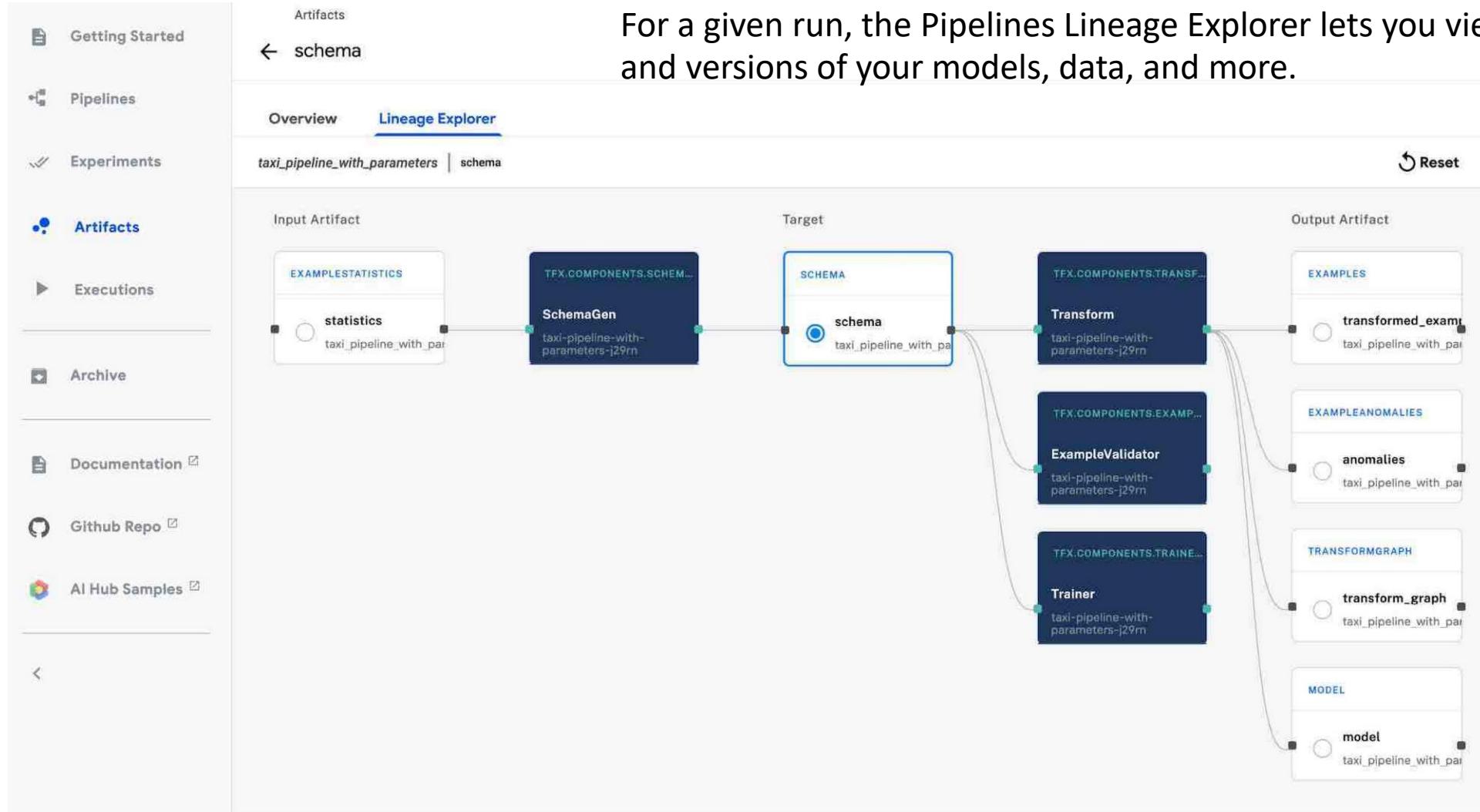
The screenshot shows the Kubeflow UI for artifact tracking. On the left, a sidebar navigation includes: Getting Started, Pipelines, Experiments, **Artifacts** (selected), Executions, Archive, Documentation, Github Repo, and AI Hub Samples. The main area is titled "Artifacts" and lists artifacts from a pipeline named "taxi_pipeline_with_parameters". The table columns are: Pipeline/Workspace, Name, ID, Type, URI, and Created at. The table data is as follows:

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...

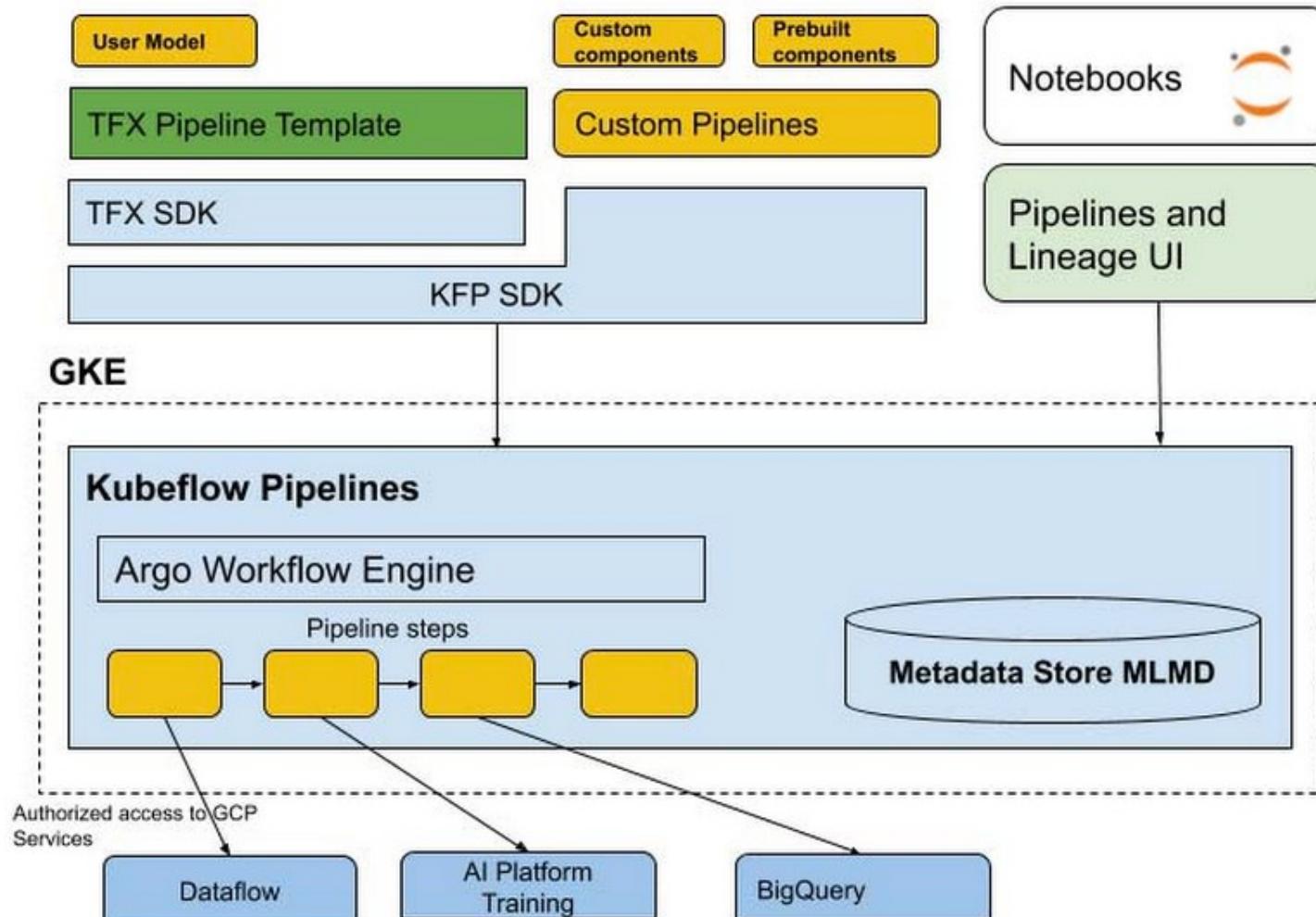
Below the artifact list, a detailed view is shown for the "model" artifact. The view includes tabs for Overview (selected) and Lineage Explorer. The "Type: Model" section shows the URI: [gs://aju-pipelines/tfx_taxi_simple/85265540-6a06-4969-a49e-1f65741878be/Trainer/model/7](#). The "Properties" section lists: name (model), pipeline_name (taxi_pipeline_with_parameters), producer_component (Trainer), and state (published). The "Custom Properties" section is currently empty.

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.

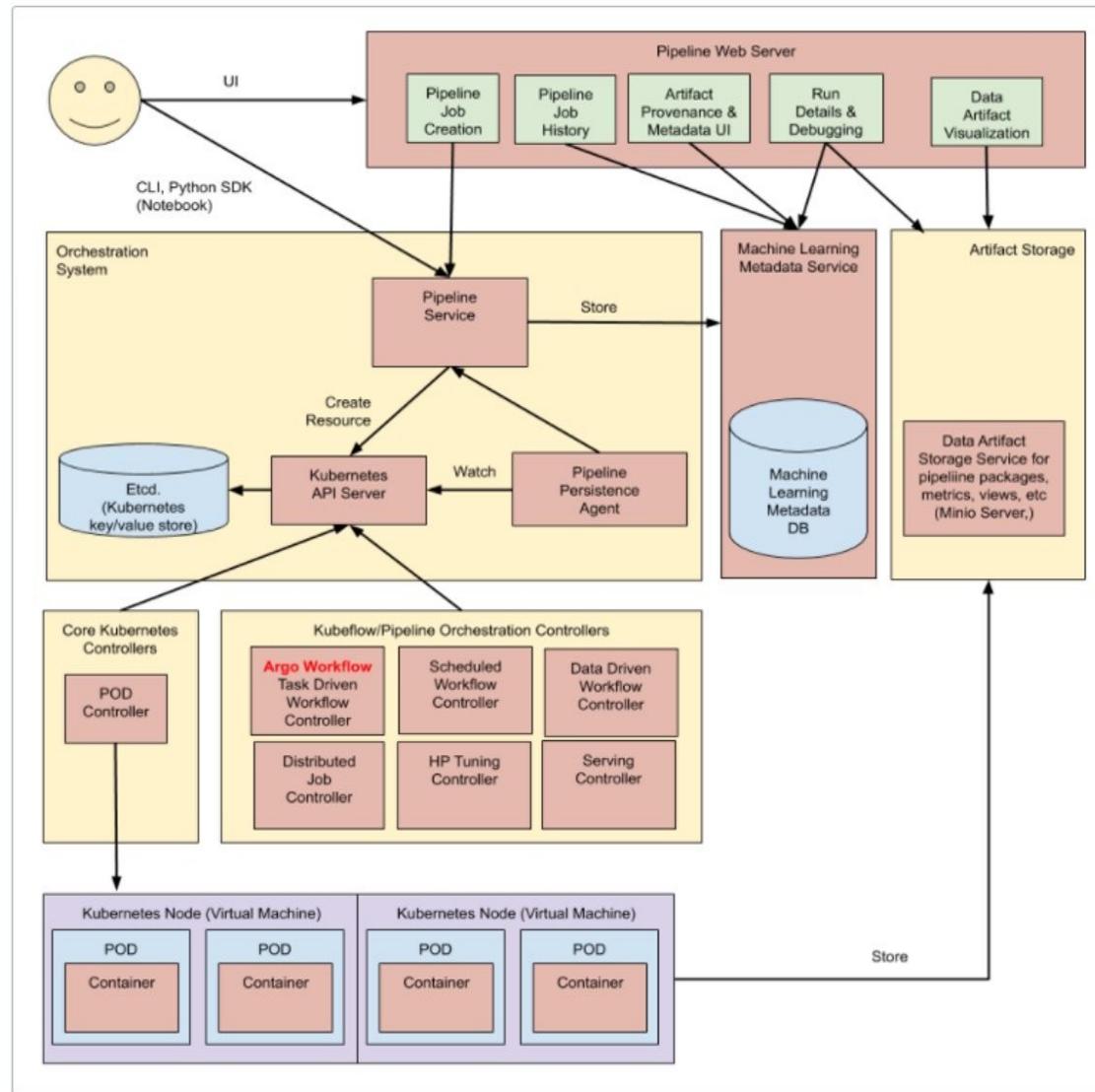
Lineage Tracking



Kubeflow Pipeline Architecture

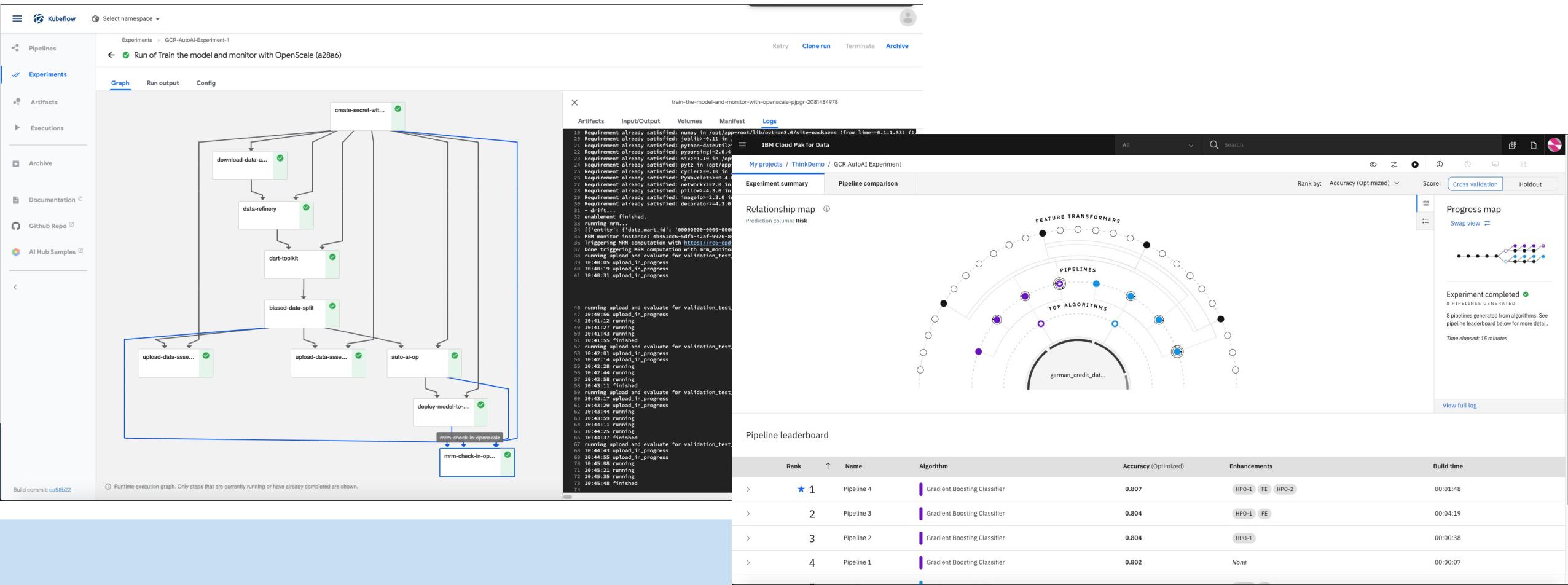


Kubeflow Pipeline Architecture



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



The screenshot displays the Kubeflow UI interface, specifically the Pipelines section. On the left, a sidebar lists various sections: Pipelines, Experiments (selected), Artifacts, Executions, Archive, Documentation, Github Repo, and AI Hub Samples. The main area shows a pipeline named "GCR-AutoAI-Experiment-1" for a task titled "Run of Train the model and monitor with OpenScale (a28a6)".

The pipeline graph (Graph tab) illustrates the workflow:

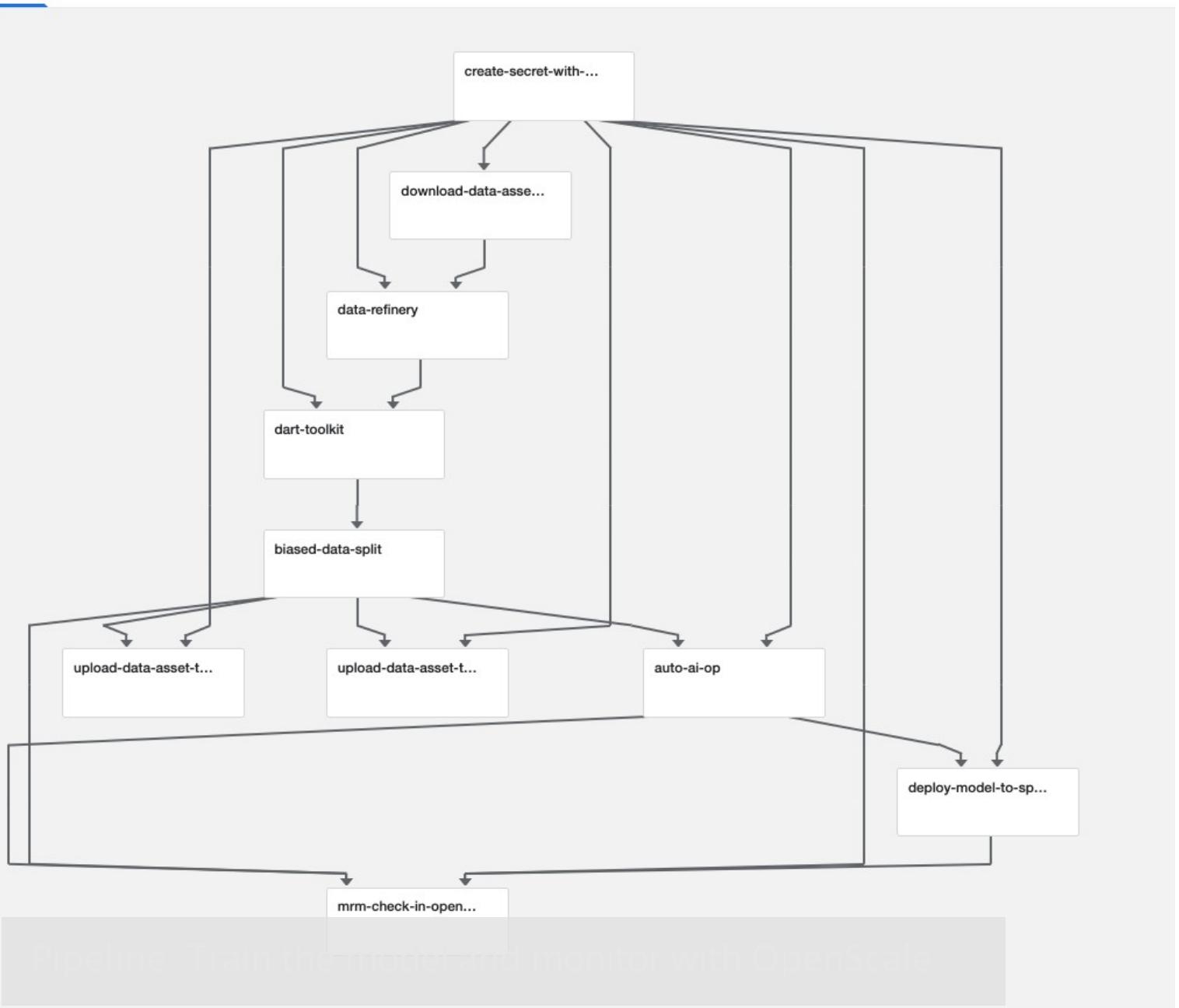
```
graph TD; A[create-secret-with...]; A --> B[download-data-a...]; B --> C[data-refinery]; C --> D[dart-toolkit]; D --> E[biased-data-split]; E --> F[auto-ai-op]; F --> G[deploy-model-to...]; G --> H[mrm-check-in-openscale]; H --> I[mrm-check-in-openscale]
```

The "Logs" tab shows the execution history:

```
train-the-model-and-monitor-with-openscale-pjpr-2081484978
...
46 running upload and evaluate for validation_test
47 10:48:56 upload_in_progress
48 10:48:56 running
49 10:48:56 finished
50 running upload and evaluate for validation_test
51 10:48:56 upload_in_progress
52 10:48:56 running
53 10:48:56 finished
54 running upload and evaluate for validation_test
55 10:48:56 upload_in_progress
56 10:48:56 running
57 10:48:56 finished
58 running upload and evaluate for validation_test
59 10:48:56 upload_in_progress
60 10:48:56 running
61 10:48:56 finished
62 running upload and evaluate for validation_test
63 10:48:56 upload_in_progress
64 10:48:56 running
65 10:48:56 finished
66 running upload and evaluate for validation_test
67 10:48:56 upload_in_progress
68 10:48:56 running
69 10:48:56 finished
70 running upload and evaluate for validation_test
71 10:48:56 upload_in_progress
72 10:48:56 running
73 10:48:56 finished
74 10:48:56 finished
```

The right side of the screen features a "Relationship map" showing connections between Feature Transformers, Pipelines, Top Algorithms, and datasets like "german_credit_dat...". Below it is a "Pipeline leaderboard" table:

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/IBM-AI-Lifecycle-Poland/kubeflow-pipelines-credentials/master/config_cpd

catalog_name

DataCatalog

asset_id

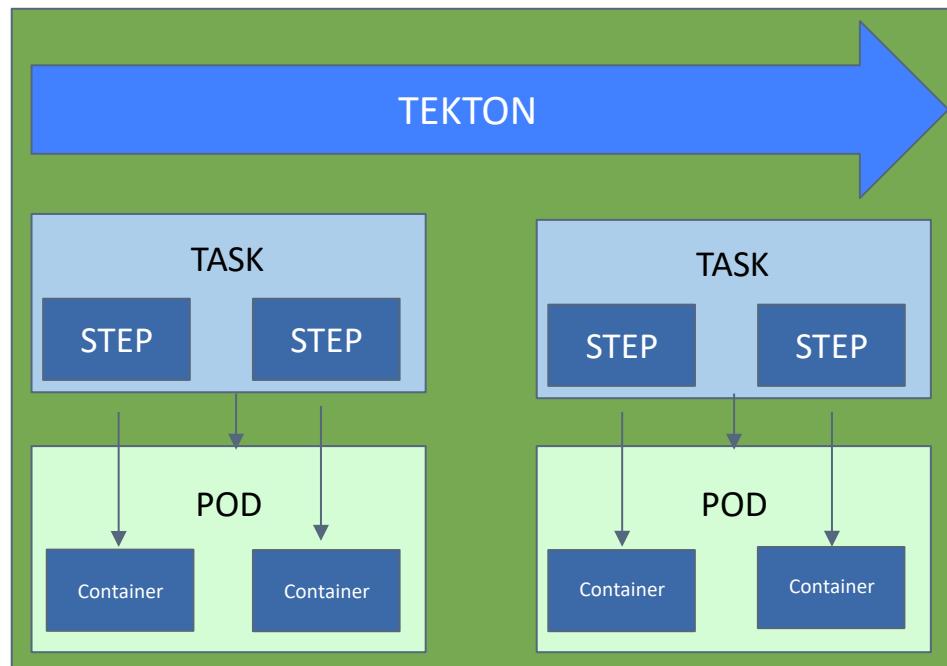
2737bafc-3f78-4e2d-850a-e7f352b3d6b8

pre_production_space_uid

1dd2aaec-781a-4712-a7ff-ae1862cf7a84

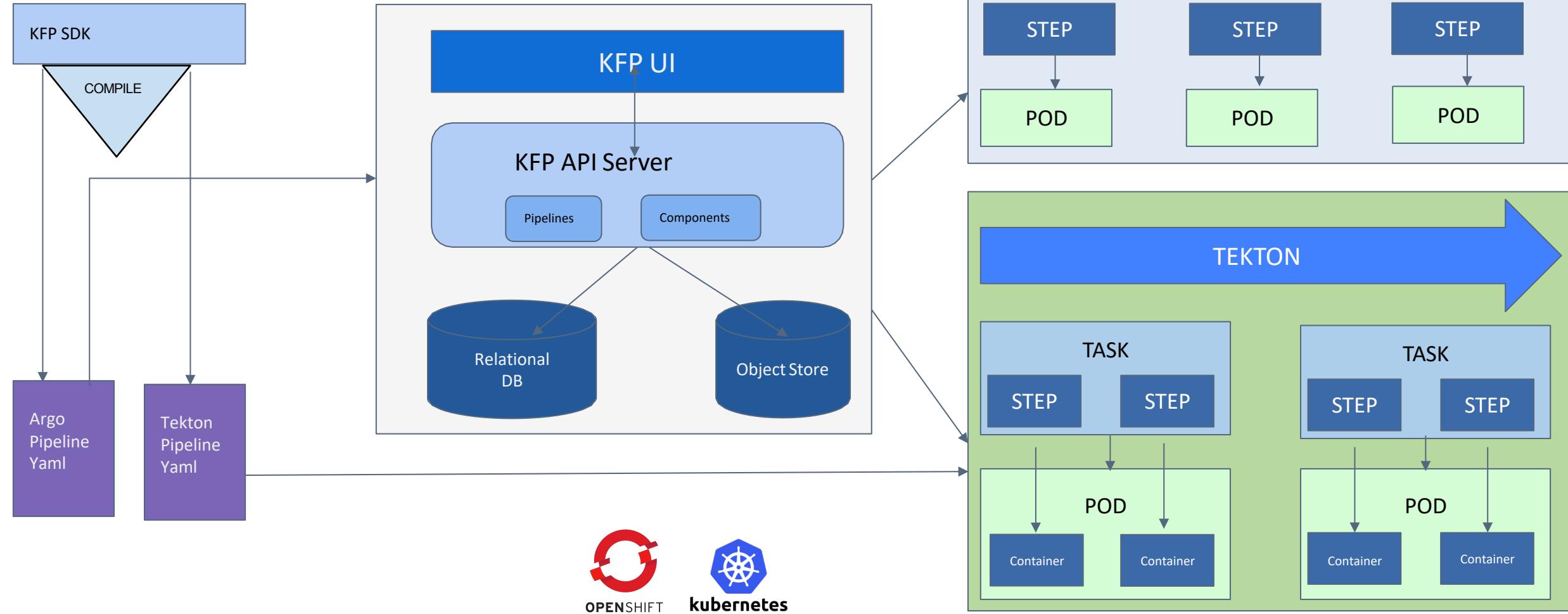
Tekton

- ❑ 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.

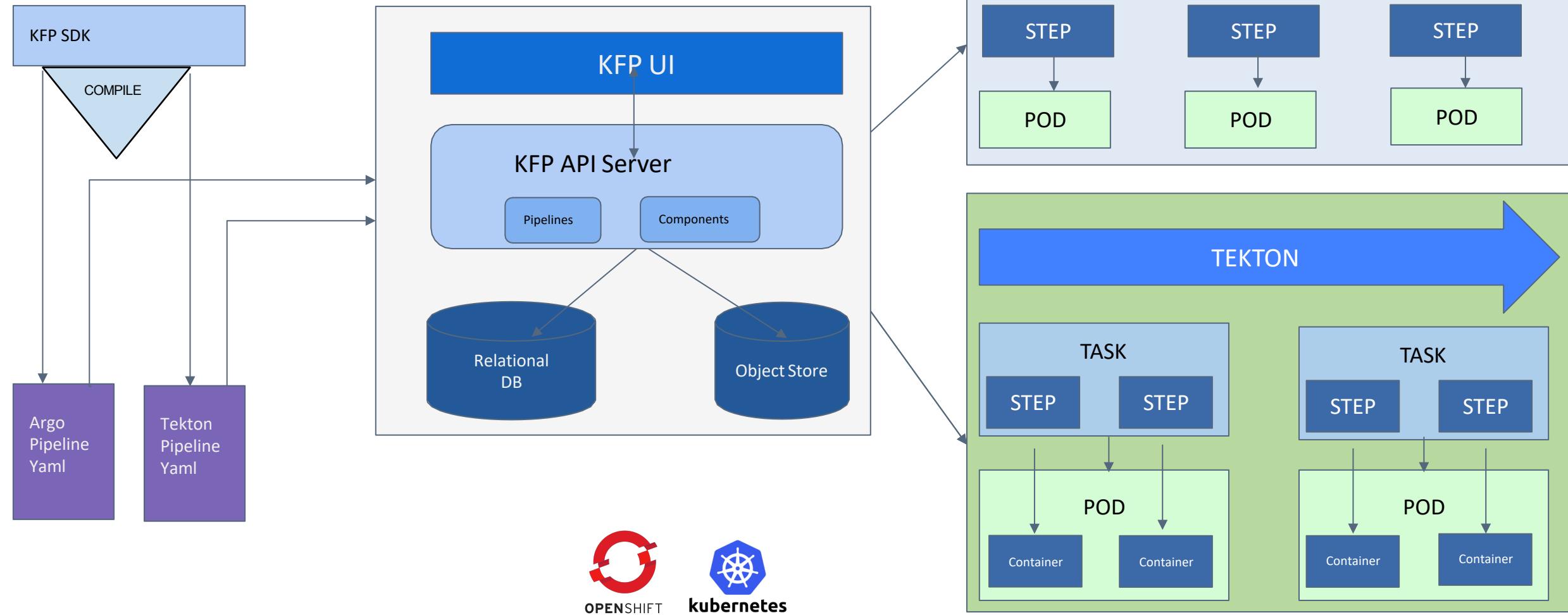
KFP – Tekton Phase One



Pluggable Components



KFP – Tekton Phase Two



Pluggable Components

Spark

Watson Studio

WML

Open Scale

Kubeflow Training

Seldon

AIF360

ART

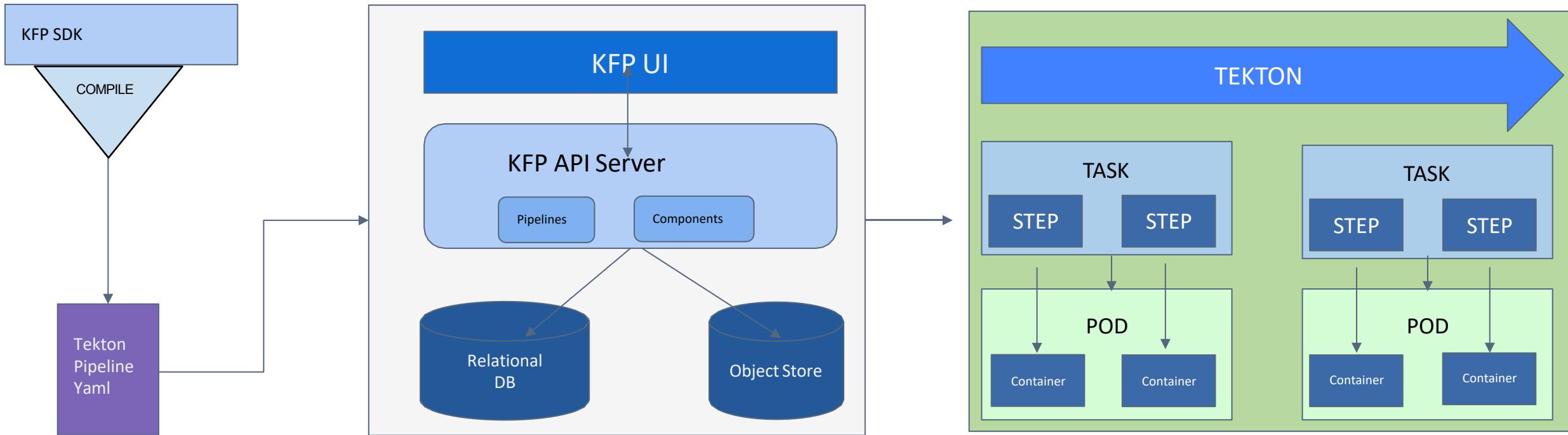
KATIB

KFSERVING

...



Kubeflow Pipelines with Tekton: Delivered



OPENSHIFT



kubernetes



Pluggable Components

Spark

Watson Studio

WML

Open Scale

Kubeflow Training

Seldon

AIF360

ART

KATIB

KFSERVING

...



DSL features implemented

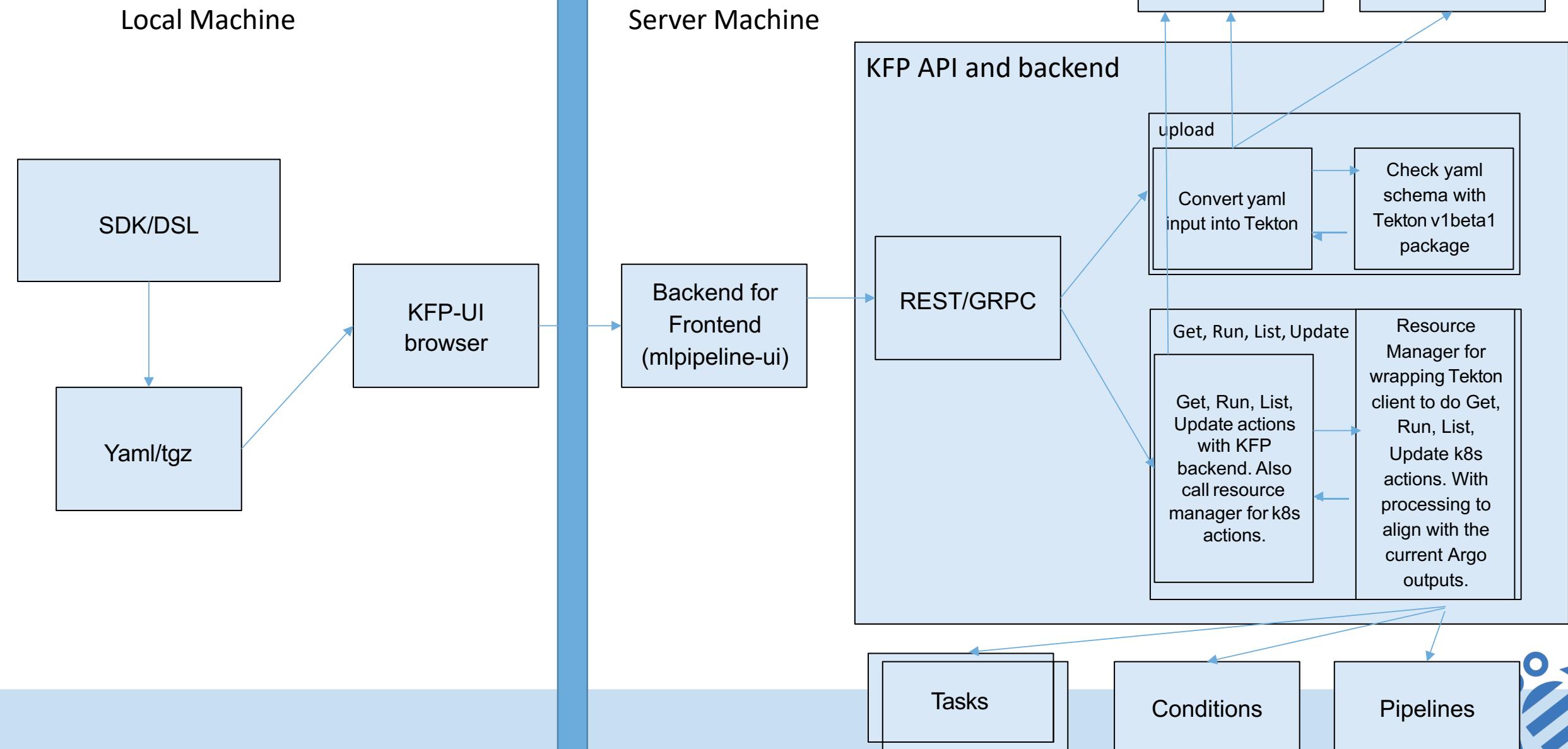
- Pipeline DSL features with native Tekton implementation
 - pod_annotations and pod_labels
 - Retries
 - Volumes
 - Timeout for Tasks and Pipelines
 - RunAfter
 - Input Parameters
 - ContainerOp
 - Affinity, Node Selector, and Tolerations
- Pipeline DSL features with custom Tekton implementation
 - 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

Pipeline samples we are running

- 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](https://github.com/kubeflow/kfp-tekton/blob/master/samples/README.md)







```
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": {}, "script": "echo $MESSAGE"}]}}}'}}
```

Experiments

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

Filter experiments

**Experiment name****Description****Last 5 runs****Default**

All runs created without specifying an experiment will be grouped here.

 Run name**Status****Duration****Pipeline Version****Recurring Run****Start time** sample-job

-

tekton-parameters

-

5/21/2020, 5:07:46 PM

Same KFP Experience: DAG, backed by Tekton YAML

Pipelines

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

+ Create run + Upload version + Create experiment Delete

[Graph](#) [YAML](#)

train-model-watson-... → store-model-watson... → deploy-model-watso...

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
0s | 1194 x 660 | x

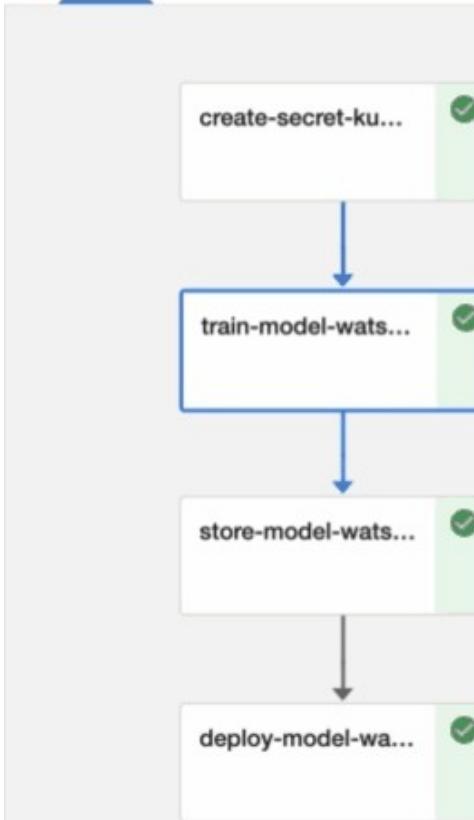
Same KFP Exp: Logs, Lineage Tracking and Artifact Tracking

Experiments > tekton-experiments

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

Retry  Clone run Terminate Archive

Graph Run output Config



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 Metric monitor started for training run: af80b10e-12f3-4053-a71c-31ff4ea8df56
23 #####
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-31ff4ea8df56 started.'}}
37 training_details {'metadata': {'created_at': '2020-07-06T21:11:38.049Z', 'guid': 'af80b10e-12f3-4053-a71c-31ff4ea8df56'}}
38
```

① Runtime execution graph. Only steps that are currently running are shown.

0s | 1214 x 669 | 

Recurring run configs Experiment description 

0 active Manage

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"/>	0s  1195 x 632  (b2541)		-	-	-	6/2/2020, 5:19:25 PM

Compiled Pipelines on Tekton

Tekton

Tekton resources ^

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

Namespace

All Namespaces X ▾

About

Import Tekton resources

Secrets

ServiceAccounts

PipelineRuns

Input a label filter of the format labelKey:labelValue

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-pipel...	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	⋮

Create +

Running Pipelines on Tekton

Tekton

Tekton resources ^

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

Namespace

default x v

About

Import Tekton resources

Secrets

ServiceAccounts

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

Succeeded Tasks Completed: 4, Skipped: 0 □

create-secret-kubernetes...
train-model-watson-machine-learning Completed
train-mode... Completed
store-model-watson-ma...
deploy-model-watson-m...

Logs Status Details

```
training_id {'metadata': {'created_at': '2020-05-07T23:57:46.868Z', 'guid': 'b200eef4-3dde-4b4e-a521-fe751735932c', 'name': 'kfp-on-wml-training-run-p7n6f'}, 'spec': {'image': 'ibm/kubeflow-training:0.1.0', 'resources': {'limits': {'cpu': '1', 'memory': '1Gi'}, 'requests': {'cpu': '1', 'memory': '1Gi'}}, 'script': 'train-model-watson-machine-learning.sh', 'timeout': '10m'}, 'status': {'state': 'running'}}

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

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

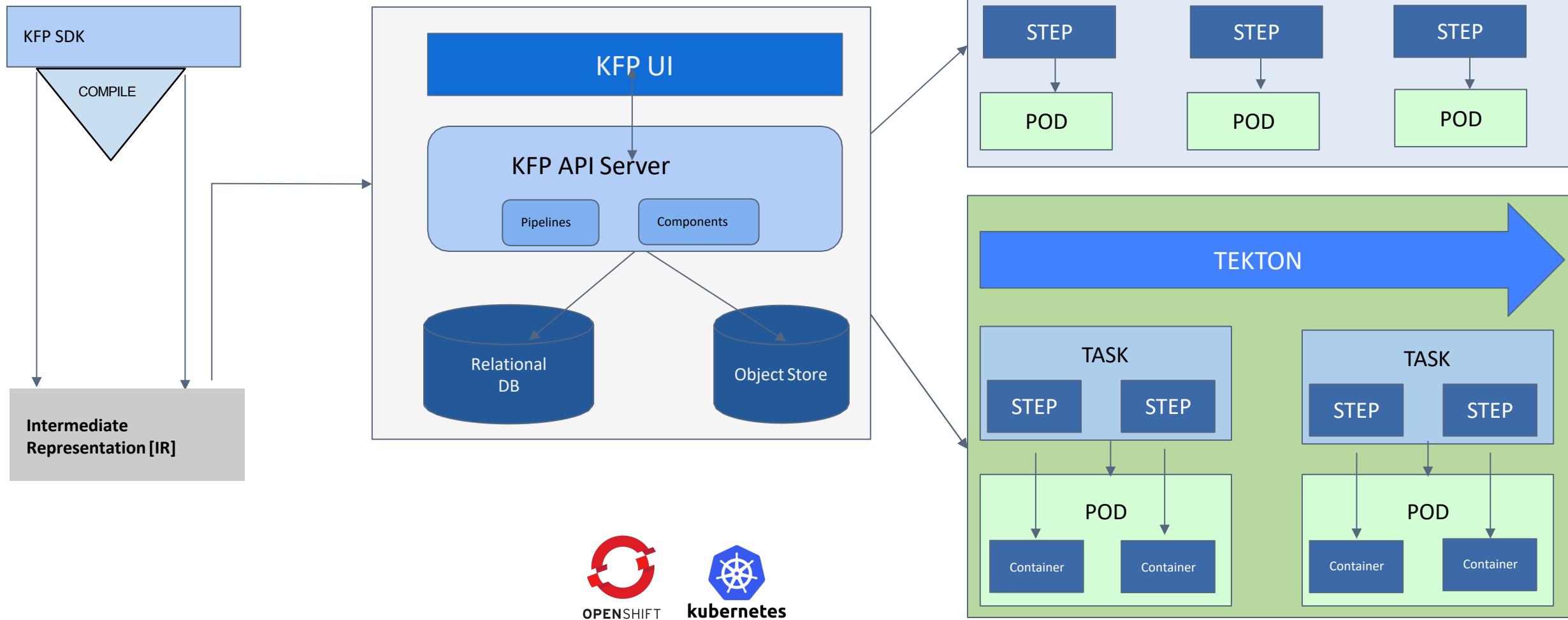
#####
Metric monitor started for training run: b200eef4-3dde-4b4e-a521-fe751735932c
#####

Rerun 

DEMO



Future: KFP – Tekton Phase Three



Pluggable Components



Useful Links

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

