Vertex AI for Easier ML Deployments

GFSA 2021 India

Sayak Paul (@RisingSayak)

$whoami

- I call `model.fit()` @ Carted
- I contribute to TensorFlow Hub, Keras Examples
- Netflix Nerd 👀
- My coordinates are here - sayak.dev

Acknowledgements

Sara Robinson (@SRobTweets)
Karl Weinmeister (@kweinmeister)
Chansung Park (@algo_diver)

Agenda

01 Challenges in MLOps

02 Tooling for MLOps

03 Vertex AI for building MLOps stack

04 Q&A

Some questions ...

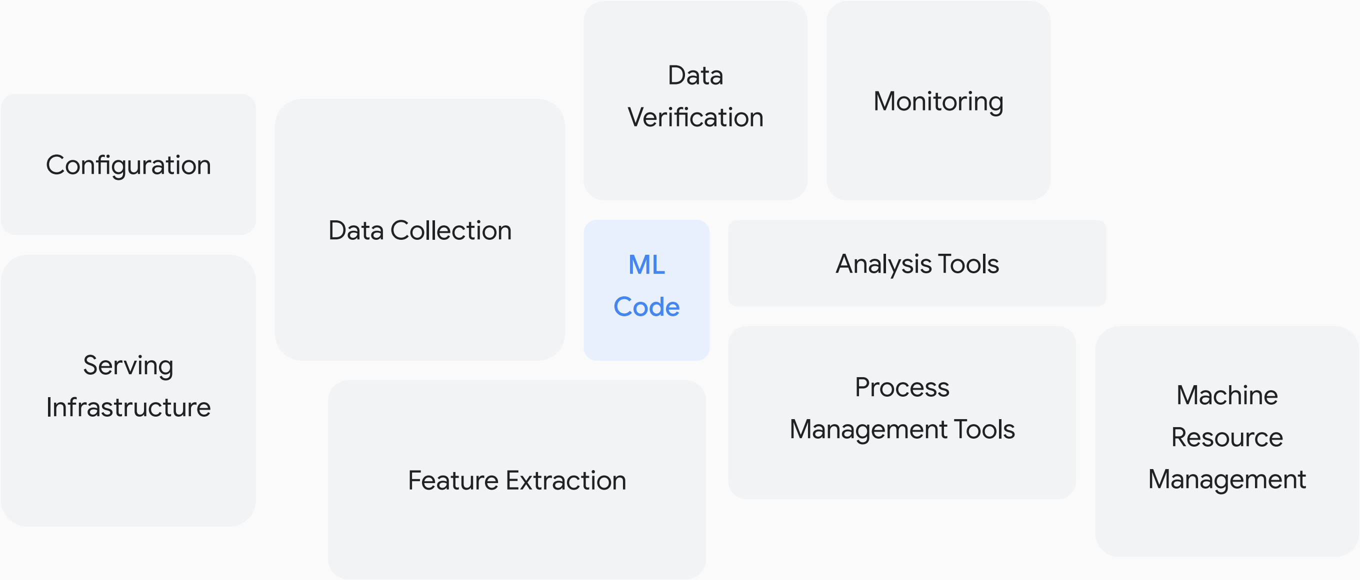
- Do train ML models continuously on new data?
- Do you deploy ML models continuously?
- Do you deploy multiple kinds of models?
- Are you training model training and deployment as a CI/CD system?
- Do you perform model monitoring at frequent intervals?
- Are your models equipped with auto-scaling?
- Are you managing the artifacts of all the steps of your ML workflow?
- ...

What is MLOps, though?

An ML engineering culture and practice that aims at **unifying** ML system development (Dev) and ML system operation (Ops).

- Google Cloud

It’s hard stuff ...



Courtesy: Google Cloud

Hidden Technical Debt in Machine Learning Systems, NeurIPS 2015

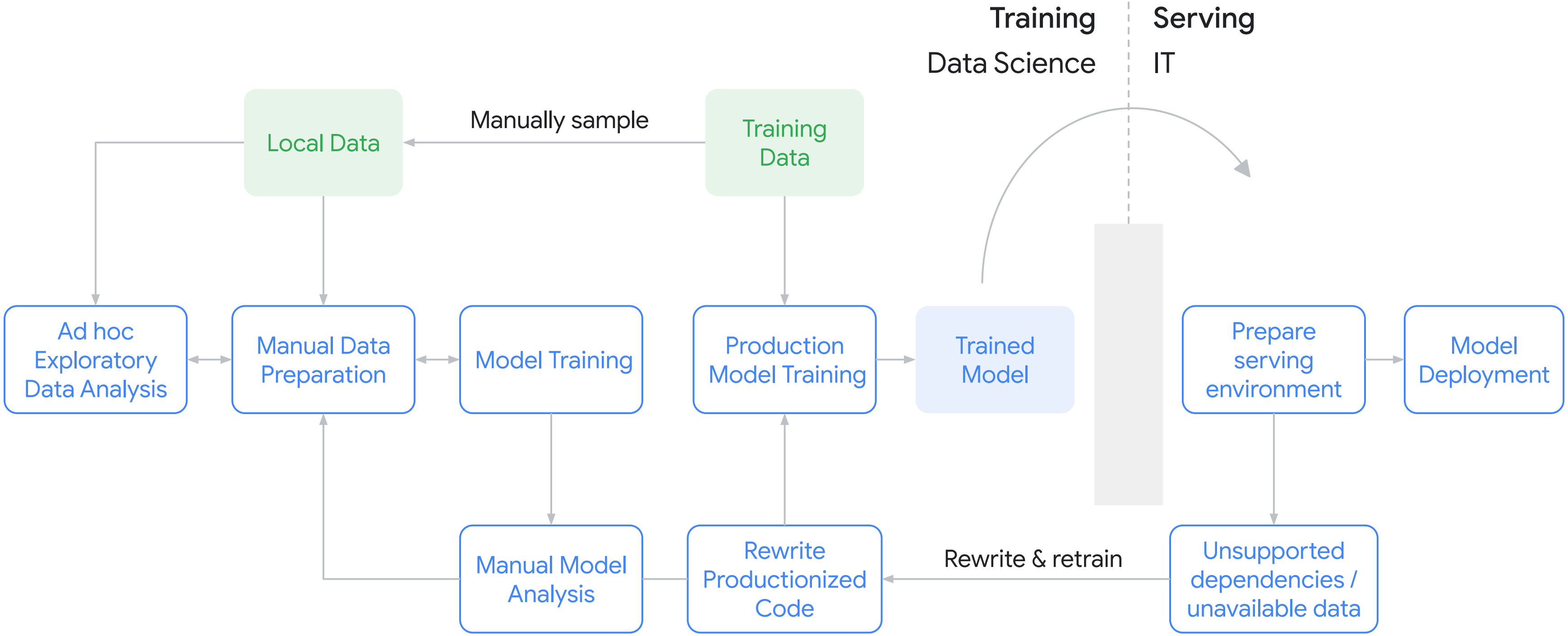
But it’s important stuff too

MLOps can help you productionize and standardize

ML systems rapidly and reliably.

- Google Cloud

What's happening today: Data Science and IT (Ops) are isolated



The Challenges on Productionizing ML

ML challenges

- Governance of data, features, models, pipelines and experiments
- Continuous training and deployment
- Training-serving skew
- Data validation
- Model analysis
- Fairness and Explainability

Production system challenges

- Scalability
- Availability
- Portability
- Reproducibility
- Modularity
- Monitoring and Alerting
- Security
- Hosted or Serverless

Enter Vertex AI!

Vertex AI is a managed ML platform for every practitioner to speed up the rate of experimentation and accelerate deployment of ML models.

Talk is cheap, show me code

```python
job = aiplatform.CustomTrainingJob(
    display_name=JOB_NAME,
    script_path="task.py",
    container_uri=TRAIN_IMAGE,
    requirements=["tensorflow_datasets==1.3.0"],
    model_serving_container_image_uri=DEPLOY_IMAGE,
)

MODEL_DISPLAY_NAME = "cifar10-" + TIMESTAMP

# Start the training
if TRAIN_GPU:
    model = job.run(
        model_display_name=MODEL_DISPLAY_NAME,
        args=CMDARGS,
        replica_count=1,
        machine_type=TRAIN_COMPUTE,
        accelerator_type=TRAIN_GPU.name,
        accelerator_count=TRAIN_NGPU,
    )
```

Asynchronous Training

```python
DEPLOYED_NAME = "cifar10_deployed-" + TIMESTAMP

TRAFFIC_SPLIT = {"0": 100}

MIN_NODES = 1
MAX_NODES = 1

if DEPLOY_GPU:
    endpoint = model.deploy(
        deployed_model_display_name=DEPLOYED_NAME,
        traffic_split=TRAFFIC_SPLIT,
        machine_type=DEPLOY_COMPUTE,
        accelerator_type=DEPLOY_GPU.name,
        accelerator_count=DEPLOY_NGPU,
        min_replica_count=MIN_NODES,
        max_replica_count=MAX_NODES,
    )
```

Deployment for Online Predictions

All from a Colab Notebook: bit.ly/vertex-online

Talk is cheap, show me code

```python
job = aiplatform.CustomTrainingJob(
    display_name=JOB_NAME,
    script_path="task.py",
    container_uri=TRAIN_IMAGE,
    requirements=["tensorflow_datasets==1.3.0"],
    model_serving_container_image_uri=DEPLOY_IMAGE,
)

MODEL_DISPLAY_NAME = "cifar10-" + TIMESTAMP

# Start the training
if TRAIN_GPU:
    model = job.run(
        model_display_name=MODEL_DISPLAY_NAME,
        args=CMDARGS,
        replica_count=1,
        machine_type=TRAIN_COMPUTE,
        accelerator_type=TRAIN_GPU.name,
        accelerator_count=TRAIN_NGPU,
    )
```

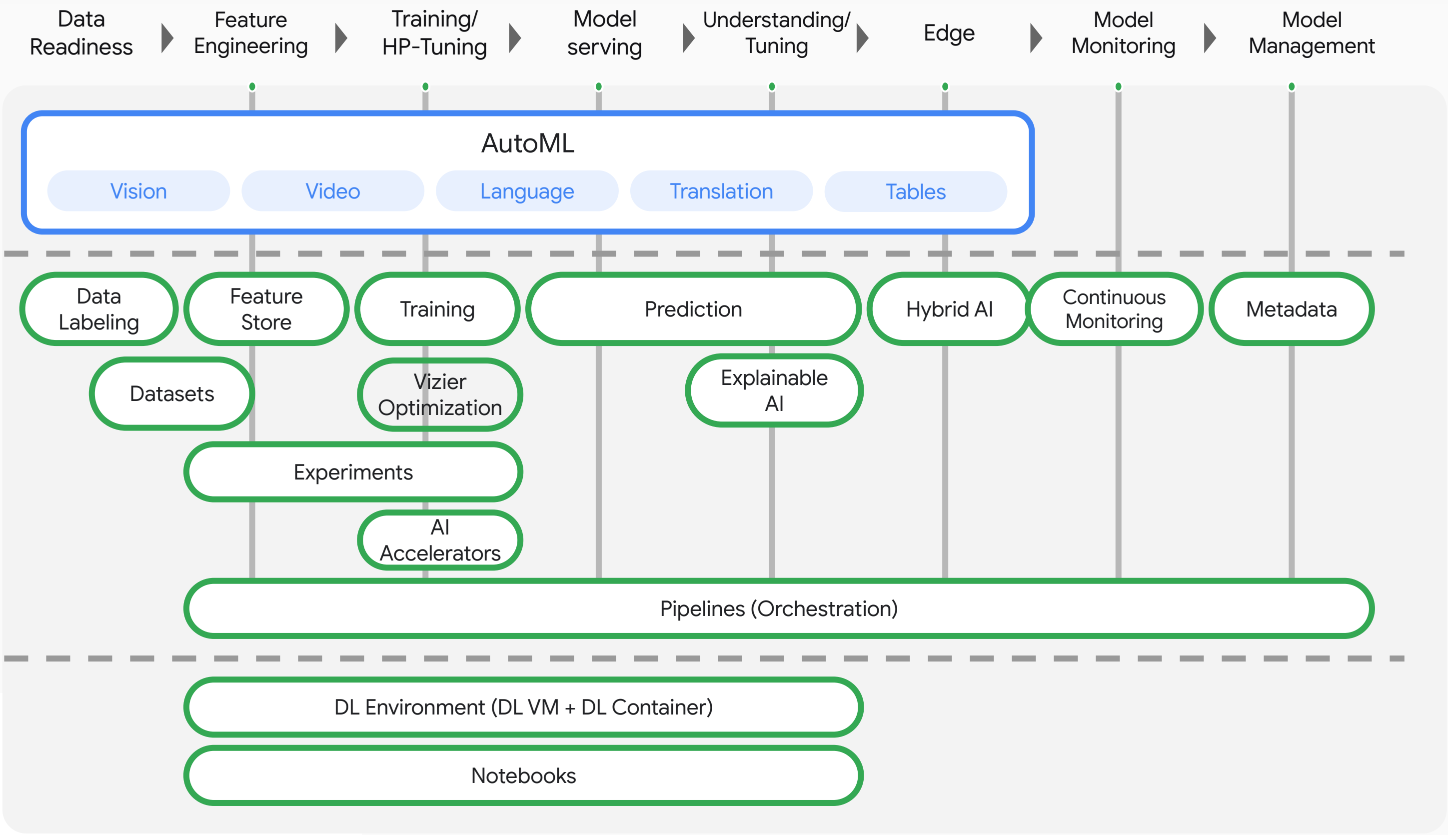
Asynchronous Training

```python
# Make SDK batch_predict method call
batch_prediction_job = model.batch_predict(
    instances_format="jsonl",
    predictions_format="jsonl",
    job_display_name=BATCH_PREDICTION_JOB_NAME,
    gcs_source=BATCH_PREDICTION_GCS_SOURCE,
    gcs_destination_prefix=BATCH_PREDICTION_GCS_DEST_PREFIX,
    model_parameters=None,
    machine_type=DEPLOY_COMPUTE,
    accelerator_type=DEPLOY_GPU,
    accelerator_count=DEPLOY_NGPU,
    starting_replica_count=MIN_NODES,
    max_replica_count=MAX_NODES,
    sync=True,
)
```

Running ***Batch*** Predictions

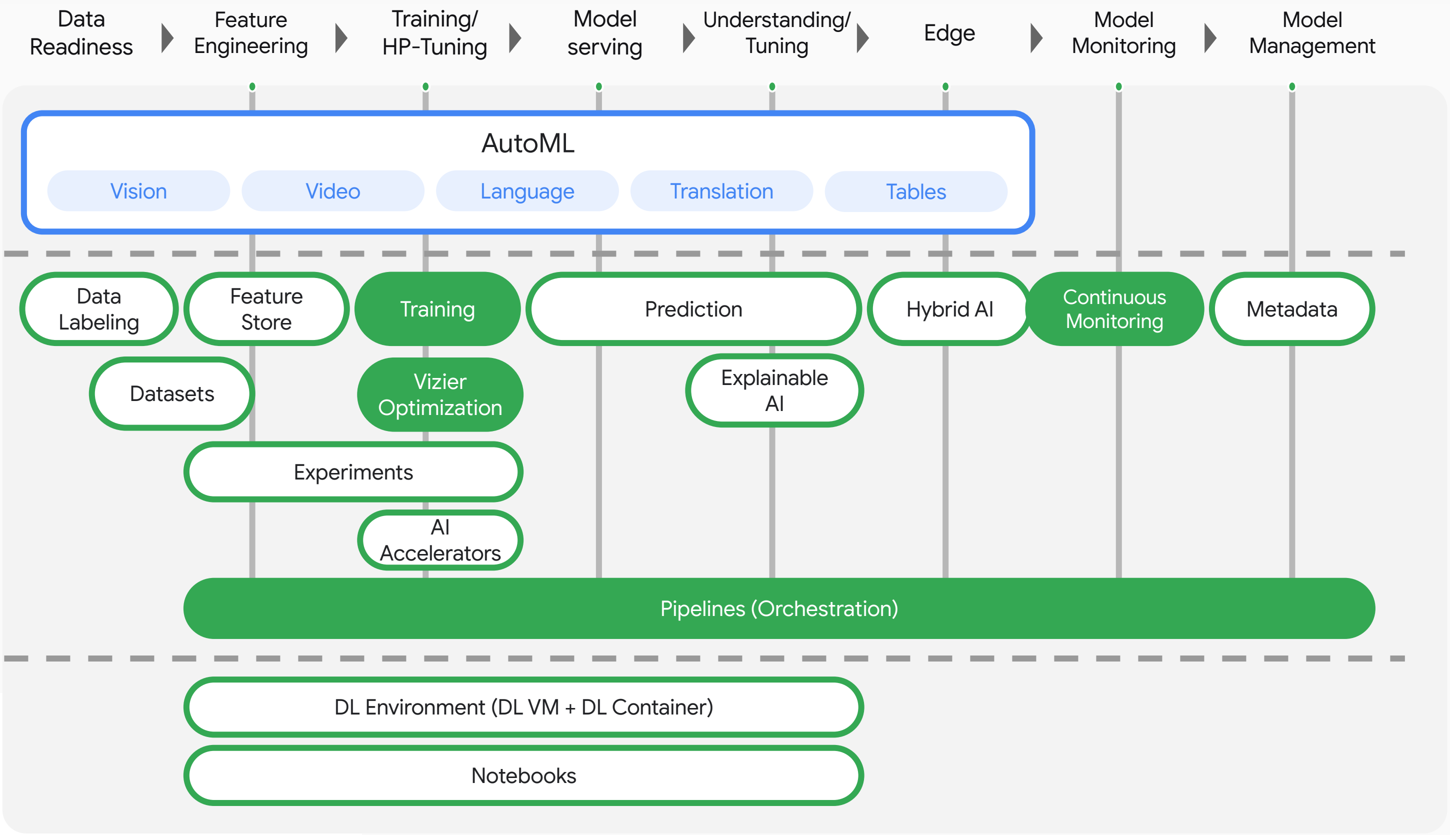
All from a Colab Notebook: bit.ly/vertex-batch

What's included in Vertex AI?



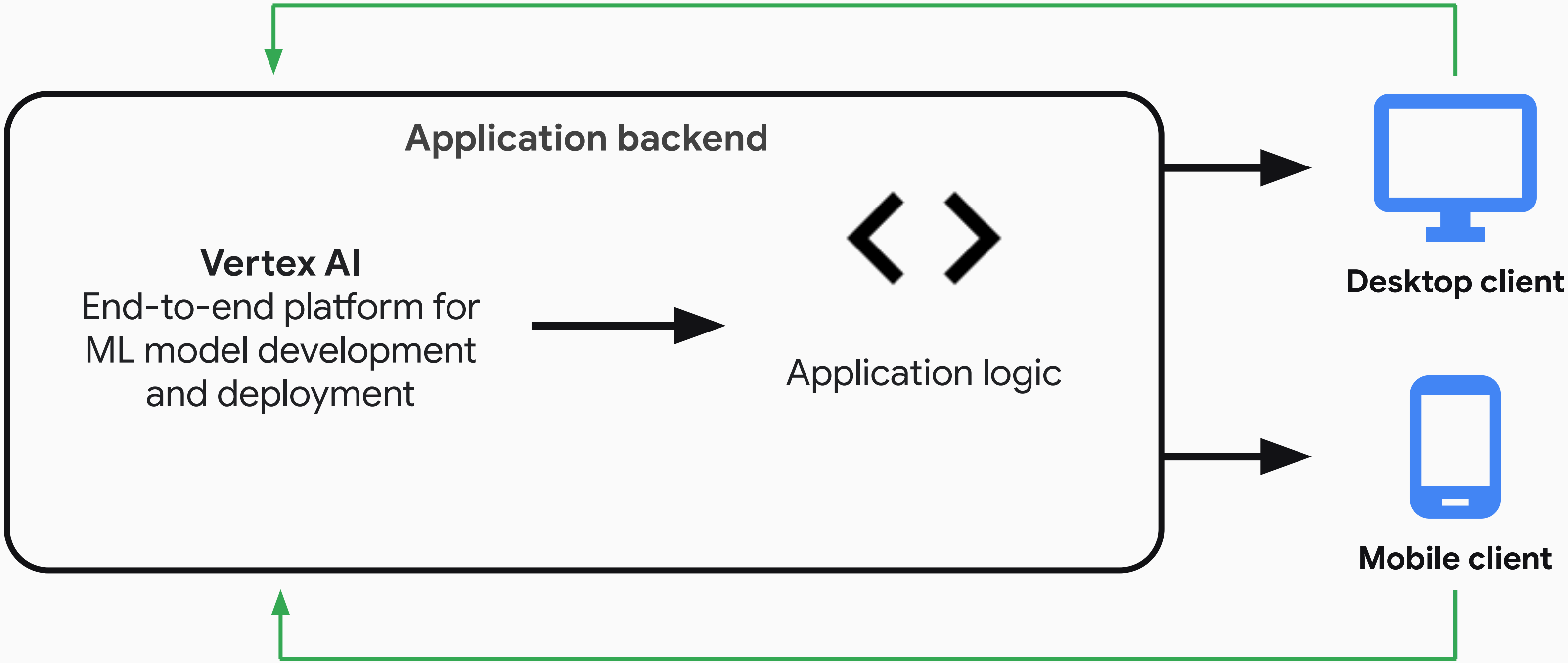
Courtesy: Google Cloud

What's included in Vertex AI?



Courtesy: Google Cloud

How does MLOps fit into app development?



Courtesy: Google Cloud

My model deployment is done at scale. Let’s now go on a trip 🚀 🛫

My model deployment is done at scale. Let’s now go on a trip 🚀 🛫

But ...

A deployed model is only the beginning

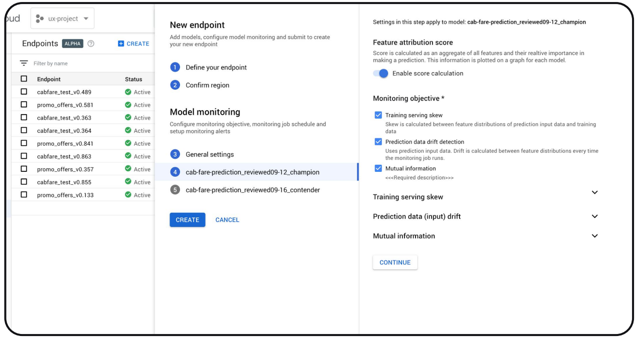
To avoid **concept** and **model drift**, ML models often need to be continuously monitored, retrained, and updated.

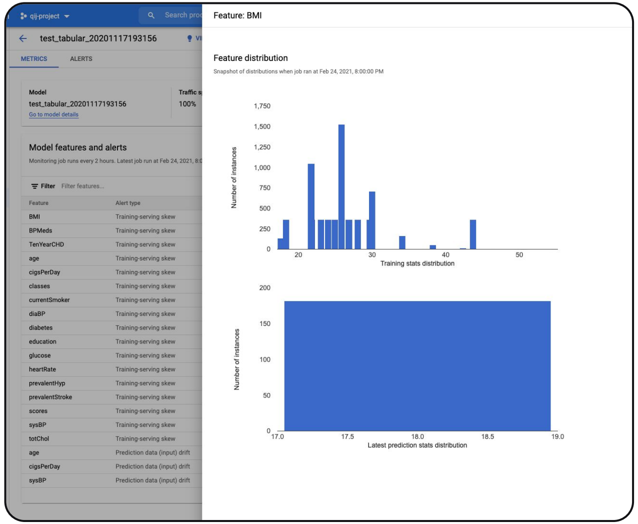
Pipelines can help **automate this workflow**.

Courtesy: Google Cloud

Model Monitoring

- **Automatically alert** your data scientists and ML engineers when model performance changes
- Detect **drift** and **training-serving skew**
- Provides confidence in model **reliability**

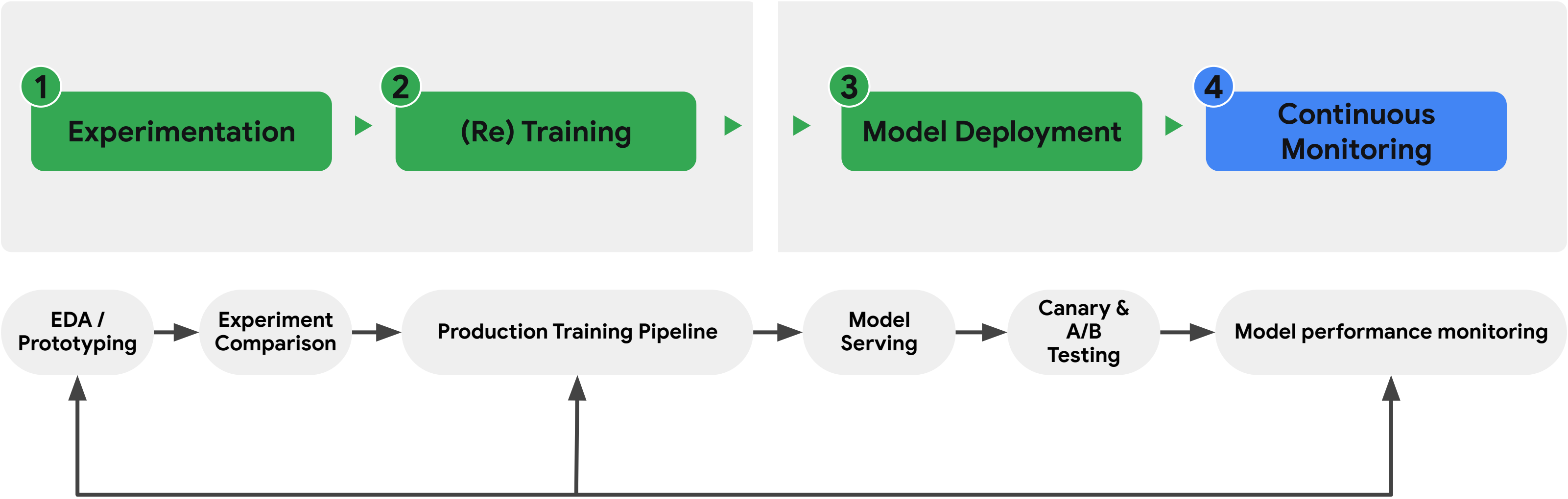




All from a Colab Notebook: bit.ly/vertex-monitor

Courtesy: Google Cloud

But how do we do this continuously?

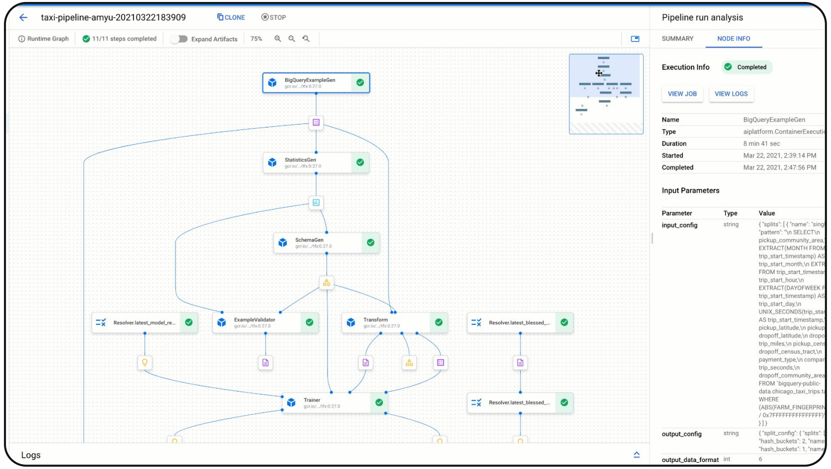


Courtesy: Google Cloud

Managed Pipelines

- Build pipelines with familiar open-source Python SDKs like TFX and Kubeflow Pipelines

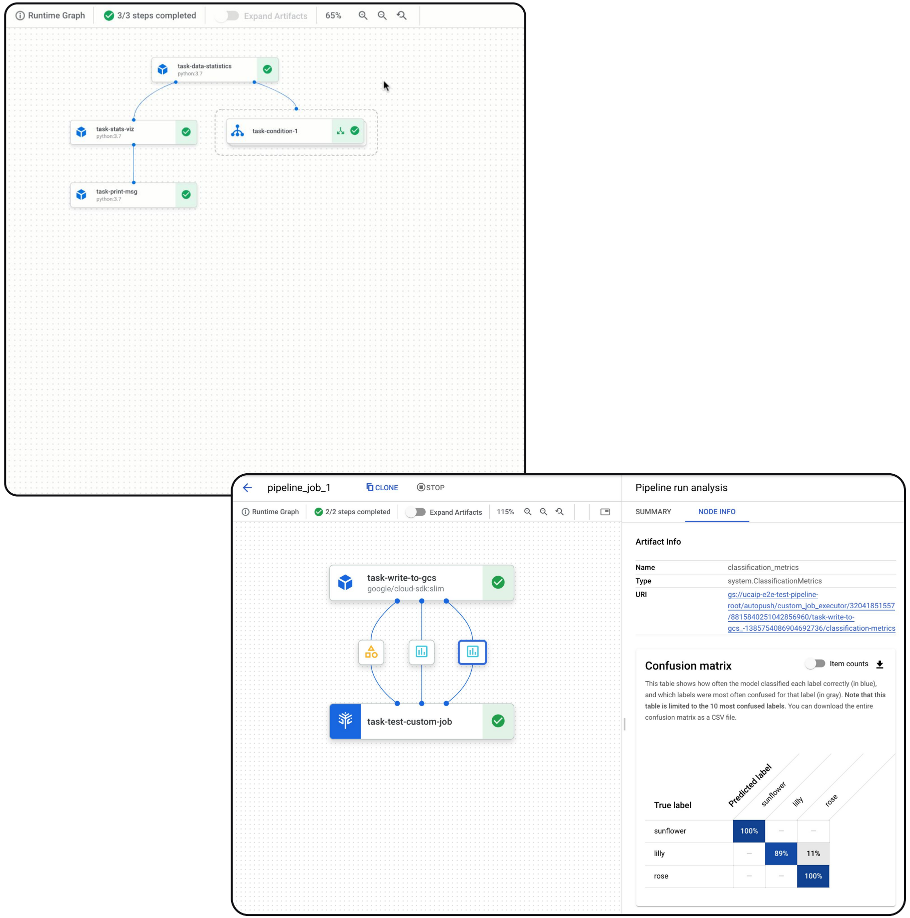
- Automated, scalable, serverless, cost effective: pay only for what you use



Courtesy: Google Cloud

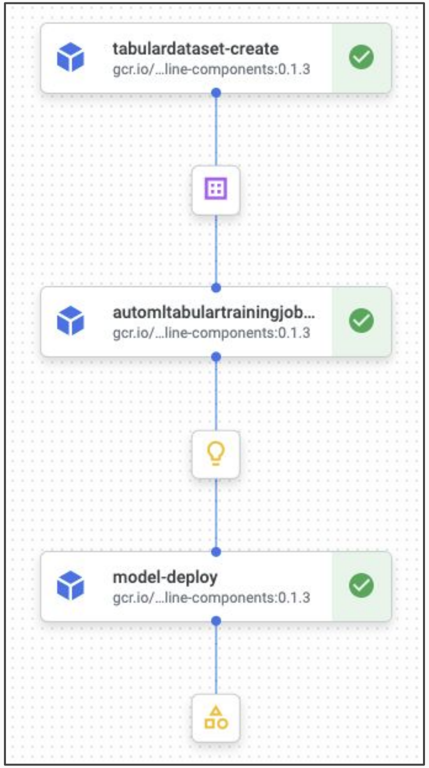
Managed Pipelines

- Add conditional logic and branches to your pipeline
- Store metadata for every artifact produced by the pipeline
- Track artifacts, lineage, metrics, and execution across your ML workflow
- Support for Cloud IAM, VPC-SC, and CMEK



Courtesy: Google Cloud

What does this look in code?

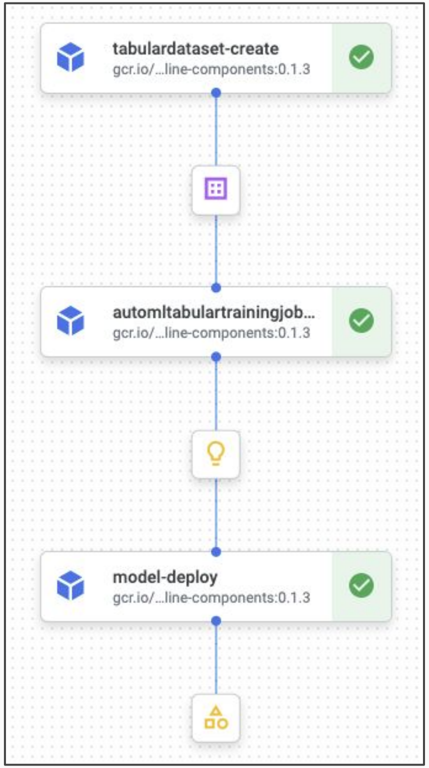


A common ML workflow

```python
@kfp.dsl.pipeline(name="automl-tab-training-v2")
def pipeline(project: str = PROJECT_ID):

    dataset_create_op = gcc_aip.TabularDatasetCreateOp(
        project=project, display_name="housing", gcs_source=gcs_csv_path
    )

    training_op = gcc_aip.AutoMLTabularTrainingJobRunOp(
        project=project,
        display_name="train-housing-automl_1",
        optimization_prediction_type="regression",
        optimization_objective="minimize-rmse",
        column_transformations=[
            {"numeric": {"column_name": "longitude"}},
            {"numeric": {"column_name": "latitude"}},
            {"numeric": {"column_name": "housing_median_age"}},
            {"numeric": {"column_name": "total_rooms"}},
            {"numeric": {"column_name": "total_bedrooms"}},
            {"numeric": {"column_name": "population"}},
            {"numeric": {"column_name": "households"}},
            {"numeric": {"column_name": "median_income"}},
            {"numeric": {"column_name": "median_house_value"}},
        ],
        dataset=dataset_create_op.outputs["dataset"],
        target_column="median_house_value",
    )

    deploy_op = gcc_aip.ModelDeployOp(  # noqa: F841
        model=training_op.outputs["model"],
        project=project,
        machine_type="n1-standard-4",
    )
```

1. [Write the pipeline]()

What does this look in code?



A common ML workflow

```python
from kfp.v2 import compiler  # noqa: F811

compiler.Compiler().compile(
    pipeline_func=pipeline, package_path="tab_regression_pipeline.json"
)
```

The pipeline compilation generates the `tab_regression_pipeline.json` job spec file.

Next, instantiate an API client object:

```python
from kfp.v2.google.client import AIPlatformClient  # noqa: F811

api_client = AIPlatformClient(project_id=PROJECT_ID, region=REGION)
```

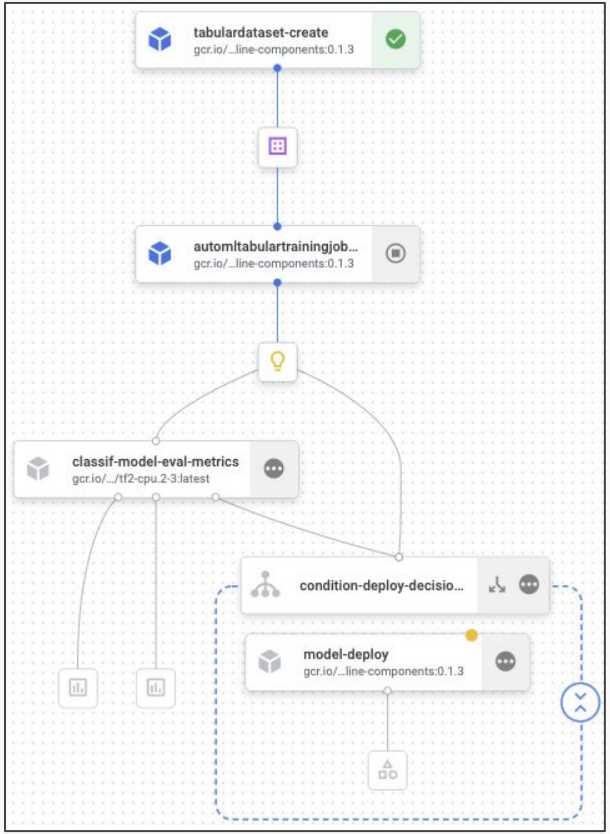
Then, you run the defined pipeline like this:

```python
response = api_client.create_run_from_job_spec(
    "tab_regression_pipeline.json",
    pipeline_root=PIPELINE_ROOT,
    parameter_values={"project": PROJECT_ID},
)
```

2. Compile and Run

All from a Colab Notebook: bit.ly/vertex-automl

What about more complicated ones?



More code but certainly doable from a Colab Notebook :D

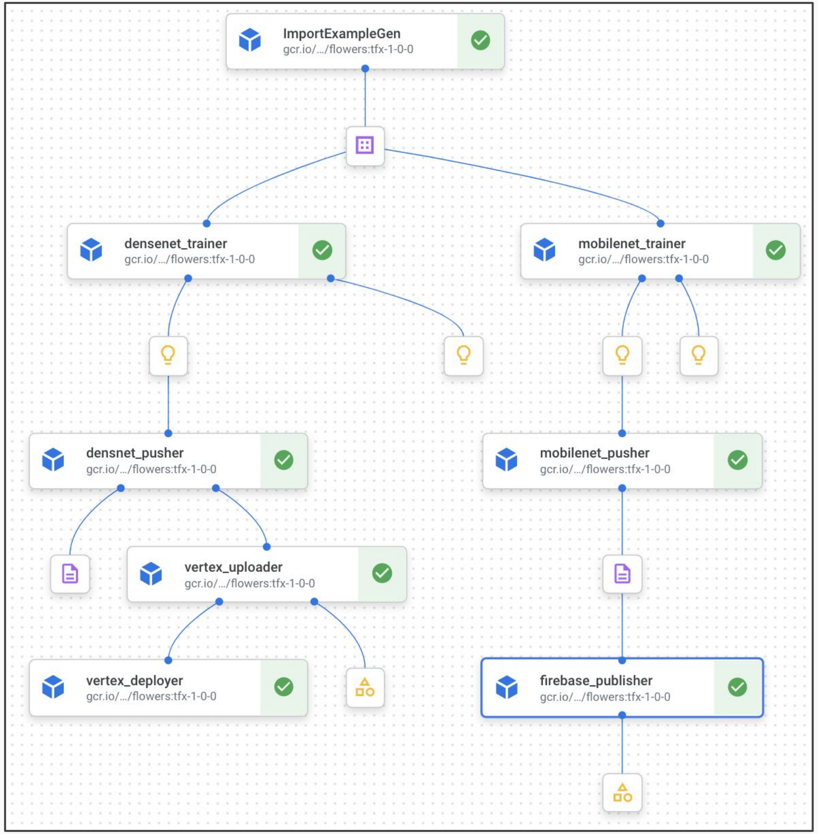
bit.ly/vertex-beans

Standard KFP components

- `AutoMLImageTrainingJobRunOp(...)`
- `AutoMLTabularTrainingJobRunOp(...)`
- `AutoMLTextTrainingJobRunOp(...)`
- `AutoMLVideoTrainingJobRunOp(...)`
- `ModelBatchPredictOp(...)`
- `ModelDeployOp(...)`

Full list is available here: [bit.ly/kfp-gcp](bit.ly/kfp-gcp)

What if I wanted to allow two different models?

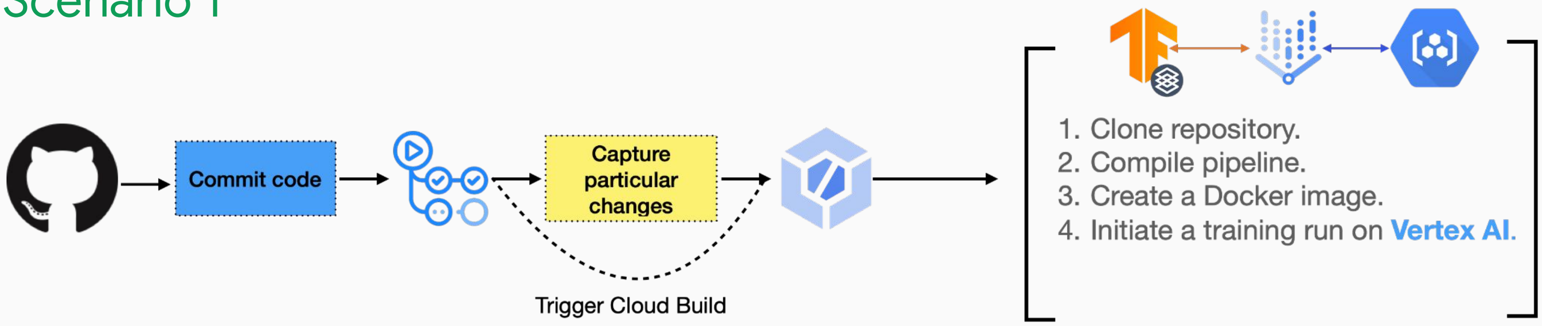


Sky's the limit!

bit.ly/dual-deployments

Can we incorporate ***CI/CD?***

Scenario 1



bit.ly/ci-cd-vertex-2

Cloud Build

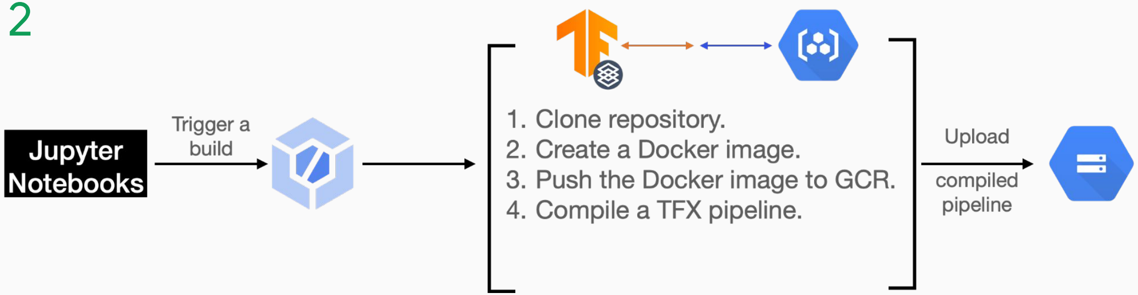
Vertex AI

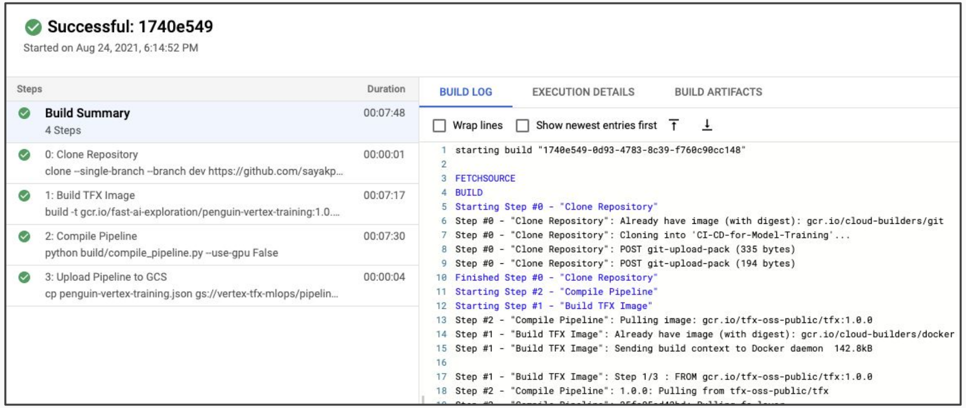
Google Container Registry

TensorFlow Extended

Can we incorporate *CI/CD*?

Scenario 2

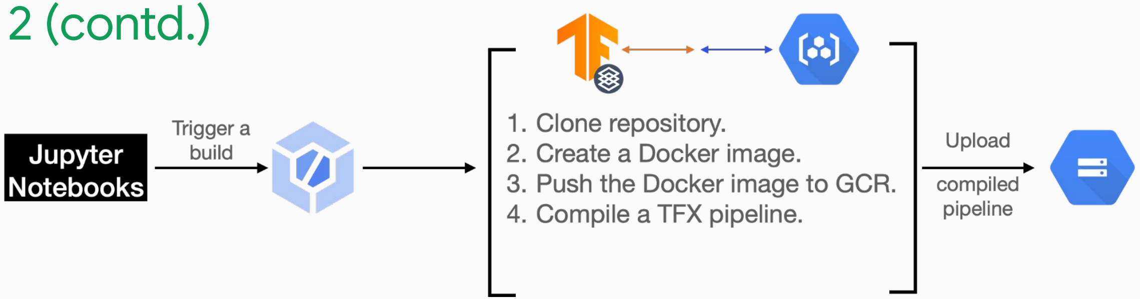




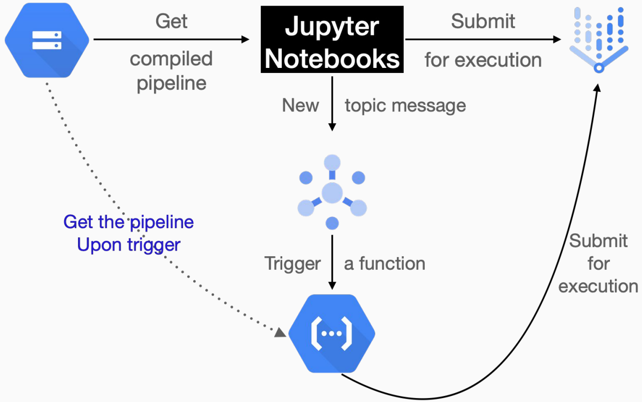
How the build looks like?

Can we incorporate ***CI/CD?***

Scenario 2 (contd.)



bit.ly/ci-cd-vertex

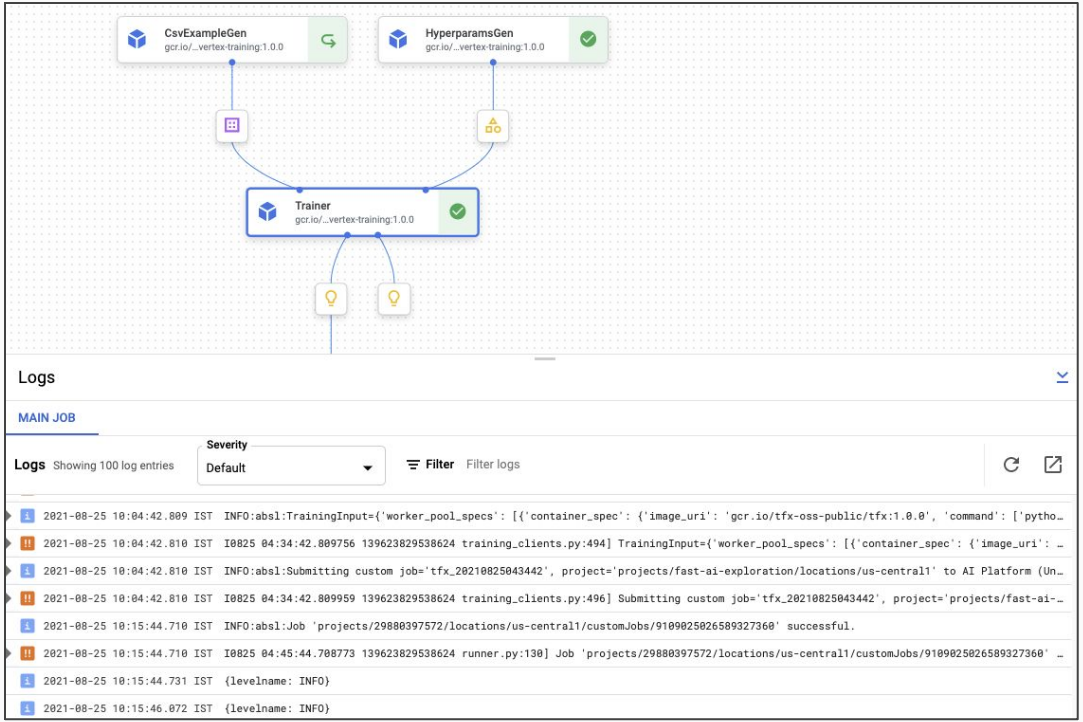


Operate effortlessly

☐	penguin-vertex-pipelines-20210723050040	Succeeded	penguin-vertex-pipelines	27 min 48 sec	Jul 23, 2021, 10:30:41 AM	Jul 23, 2021, 10:58:30 AM
☐	automl-tab-training-v2-20210723034427	Succeeded	automl-tab-training-v2	1 hr 28 min	Jul 23, 2021, 9:14:28 AM	Jul 23, 2021, 10:42:49 AM
☐	automl-tab-training-v2-20210723033914	Failed	automl-tab-training-v2			
☐	custom-cifar10-training-20210723022432	Failed	custom-cifar10-training	2 min 47 sec	Jul 23, 2021, 7:54:33 AM	Jul 23, 2021, 7:57:21 AM
☐	custom-cifar10-training-20210723022324	Failed	custom-cifar10-training			
☐	hello-world-v2-20210722115629	Succeeded	hello-world-v2	5 min 2 sec	Jul 22, 2021, 5:26:30 PM	Jul 22, 2021, 5:31:32 PM
☐	hello-world-v2-20210722115441	Failed	hello-world-v2			
☐	automl-image-training-v2-20210722093207	Canceled	automl-image-training-v2	5 min 49 sec	Jul 22, 2021, 3:02:08 PM	Jul 22, 2021, 3:07:57 PM
☐	automl-tab-beans-training-v2-20210722090230	Canceled	automl-tab-beans-training-v2	26 min 13 sec	Jul 22, 2021, 2:32:31 PM	Jul 22, 2021, 2:58:44 PM
☐	pipeline-with-loops-and-conditions-spsayakpaul-20210722082831	Succeeded	pipeline-with-loops-and-conditions-spsayakpaul	8 min 6 sec	Jul 22, 2021, 1:58:32 PM	Jul 22, 2021, 2:06:38 PM
☐	pipeline-with-loops-and-conditions-spsayakpaul-20210722082539	Failed	pipeline-with-loops-and-conditions-spsayakpaul			
☐	pipeline-with-loops-and-conditions-spsayakpaul-20210722081841	Failed	pipeline-with-loops-and-conditions-spsayakpaul			

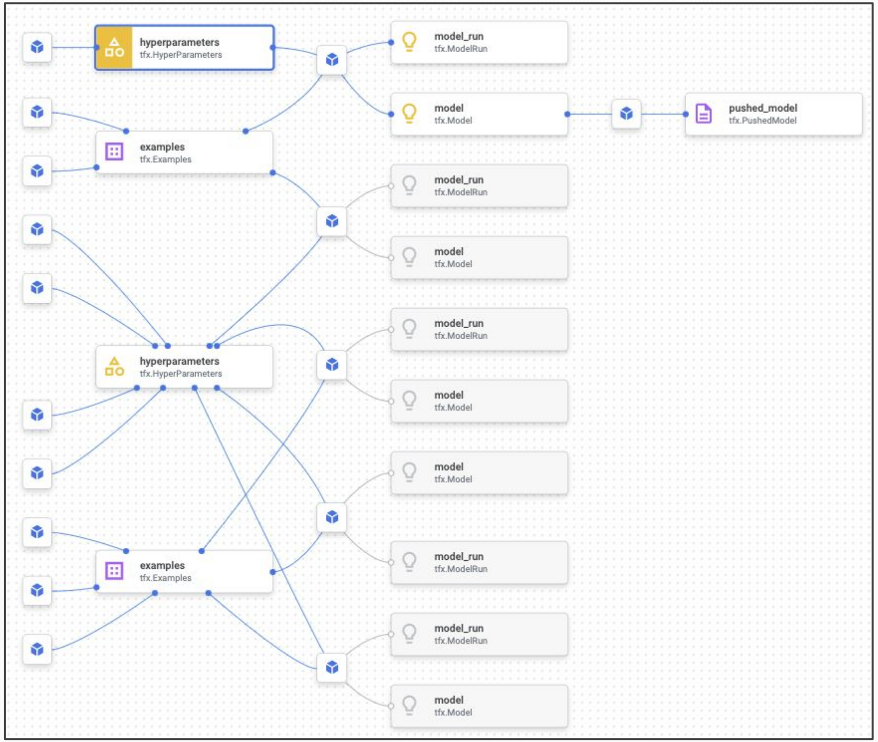
A central dashboard for all your pipelines

Operate effortlessly



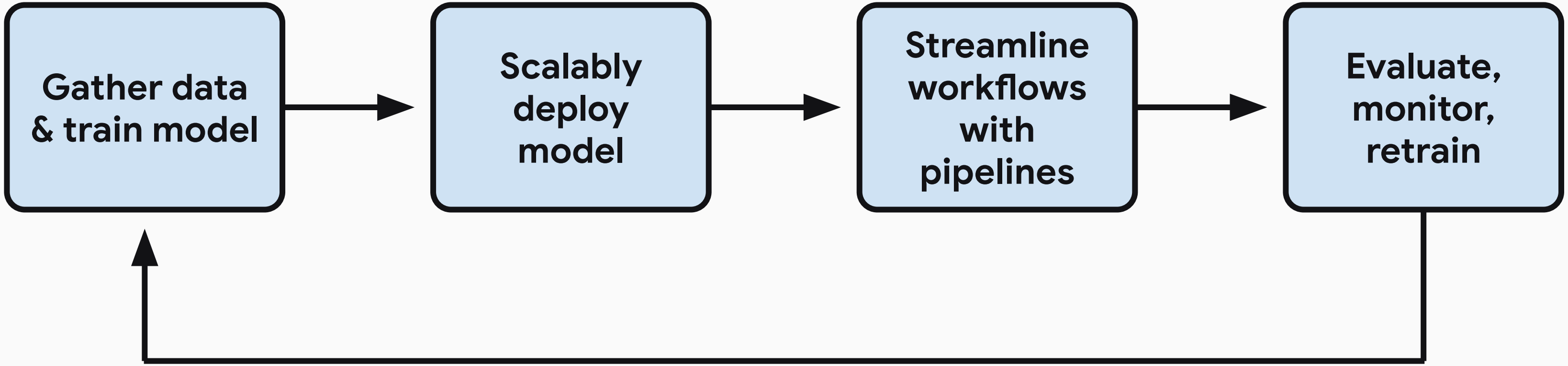
Automated logging at each step

Operate effortlessly



Track lineage of all the artifacts

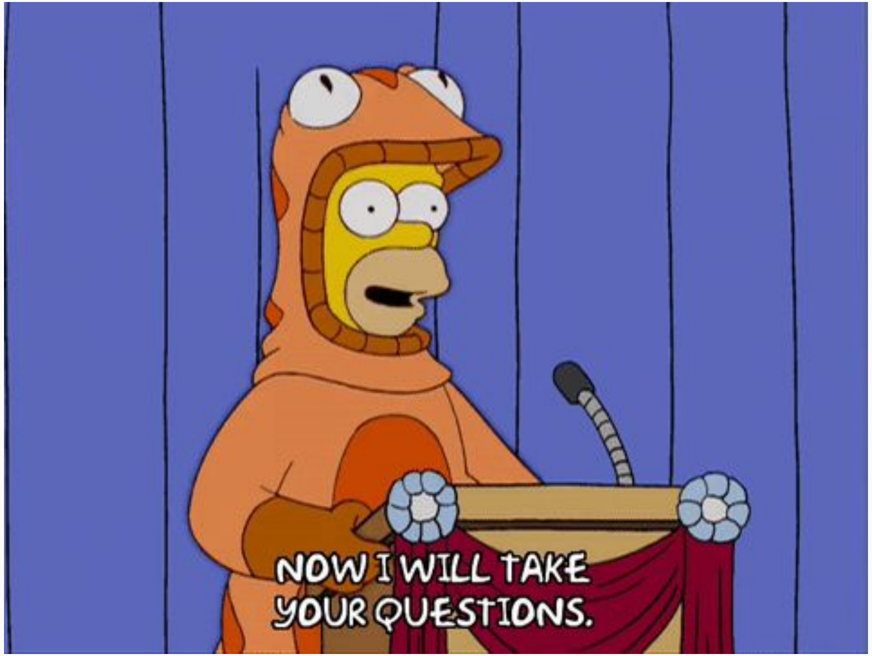
Streamlining ML workflows with Vertex AI



Courtesy: Google Cloud

Learning more

- Vertex AI documentation
- Goldmine of MLOps code
- Machine Learning Design Patterns [Book]
- Applied ML Summit from GCP
- Machine Learning Engineering for Production (MLOps) Specialization
- Made With ML: Home
- Full Stack Deep Learning


NOW I WILL TAKE
YOUR QUESTIONS.

@RisingSayak