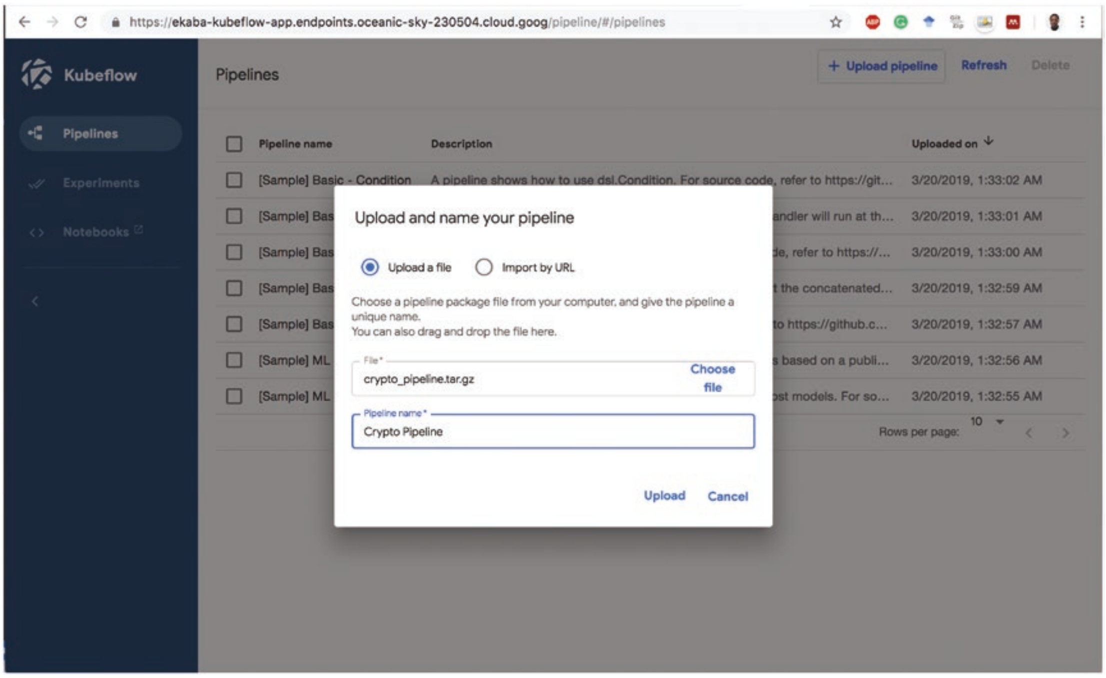
Upload and Execute the Pipeline to Kubeflow Pipelines

The following steps upload and execute the compiled pipeline on Kubeflow Pipelines:

1. Upload the pipeline to Kubeflow Pipelines (Figure 47-1).



Pipeline name	Description	Uploaded on
[Sample] Basic - Condition	A pipeline shows how to use dsl.Condition. For source code, refer to https://git...	3/20/2019, 1:33:02 AM
[Sample] Bas	andler will run at th...	3/20/2019, 1:33:01 AM
[Sample] Bas	de, refer to https://...	3/20/2019, 1:33:00 AM
[Sample] Bas	t the concatenated...	3/20/2019, 1:32:59 AM
[Sample] Bas	to https://github.c...	3/20/2019, 1:32:57 AM
[Sample] ML	s based on a publi...	3/20/2019, 1:32:56 AM
[Sample] ML	ost models. For so...	3/20/2019, 1:32:55 AM

Upload and name your pipeline

[✓ Upload a file] Import by URL

Choose a pipeline package file from your computer, and give the pipeline a unique name.
You can also drag and drop the file here.

File*: crypto_pipeline.tar.gz   Choose file

Pipeline name*: Crypto Pipeline

Upload   Cancel

Figure 47-1. *Upload the compiled pipeline to Kubeflow Pipelines*

2. Click the pipeline to see the static graph of the flow (Figure 47-2).

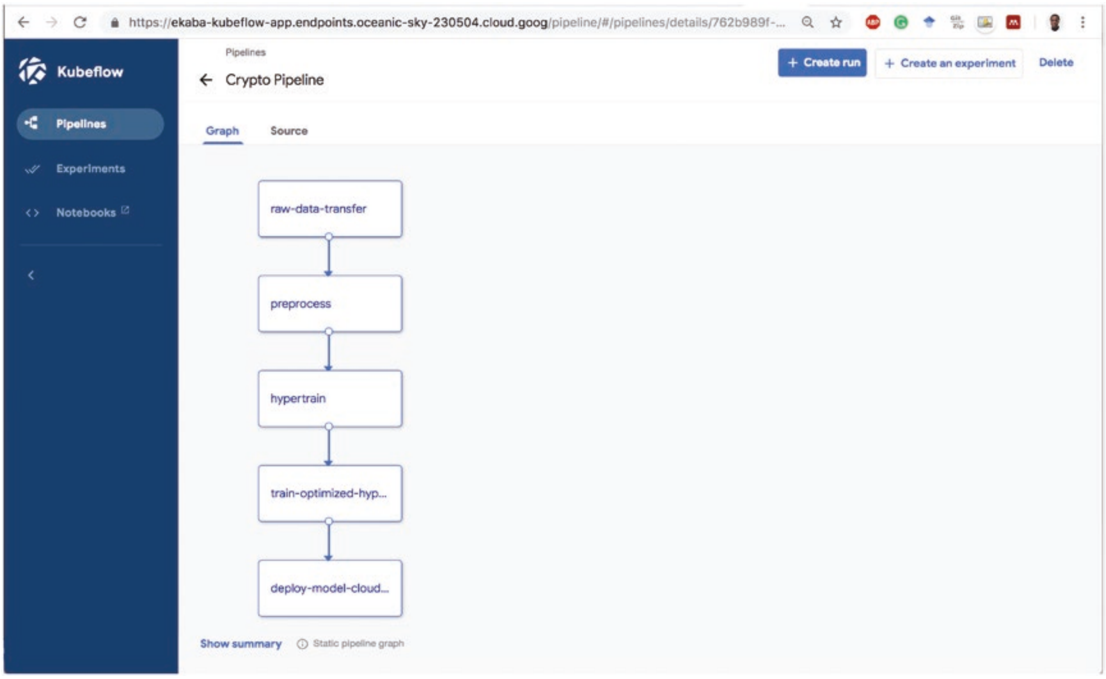
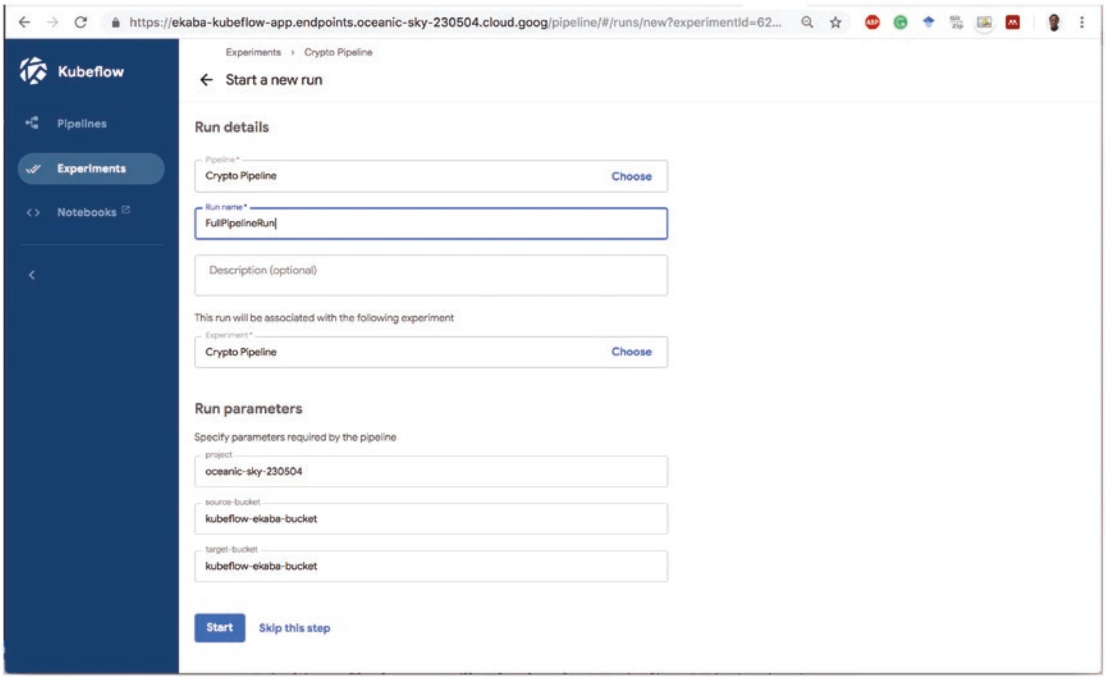


Figure 47-2. *Pipeline summary graph*

3. Create an Experiment and run to execute the pipeline (Figure 47-3).



https://ekaba-kubeflow-app.endpoints.oceanic-sky-230504.cloud.goog/pipeline/#/runs/new?experimentId=62...
Kubeflow
Pipelines
Experiments
Notebooks
Experiments › Crypto Pipeline
Start a new run
Choose
Start
Skip this step

Run details

Pipeline*: Crypto Pipeline

Run name*: FullPipelineRun

Description (optional)

This run will be associated with the following experiment

Experiment*: Crypto Pipeline

Run parameters

Specify parameters required by the pipeline

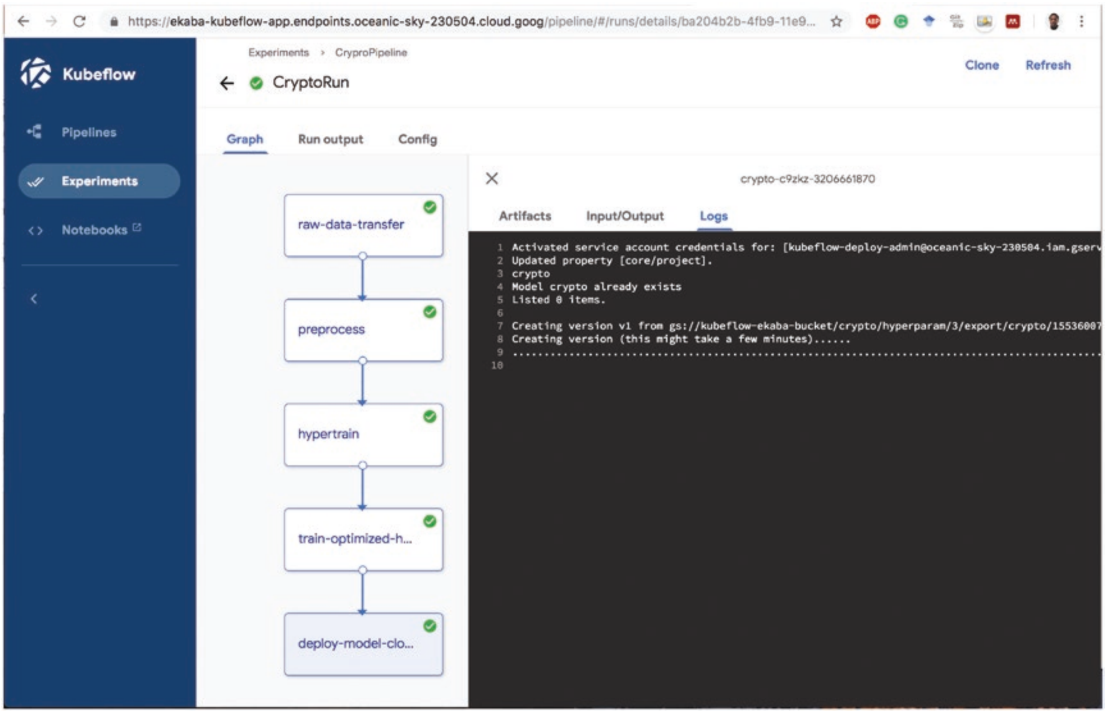
project: oceanic-sky-230504

source-bucket: kubeflow-ekaba-bucket

target-bucket: kubeflow-ekaba-bucket

Figure 47-3. *Create and run the Experiment*

4. Completed Pipeline run (Figure 47-4).

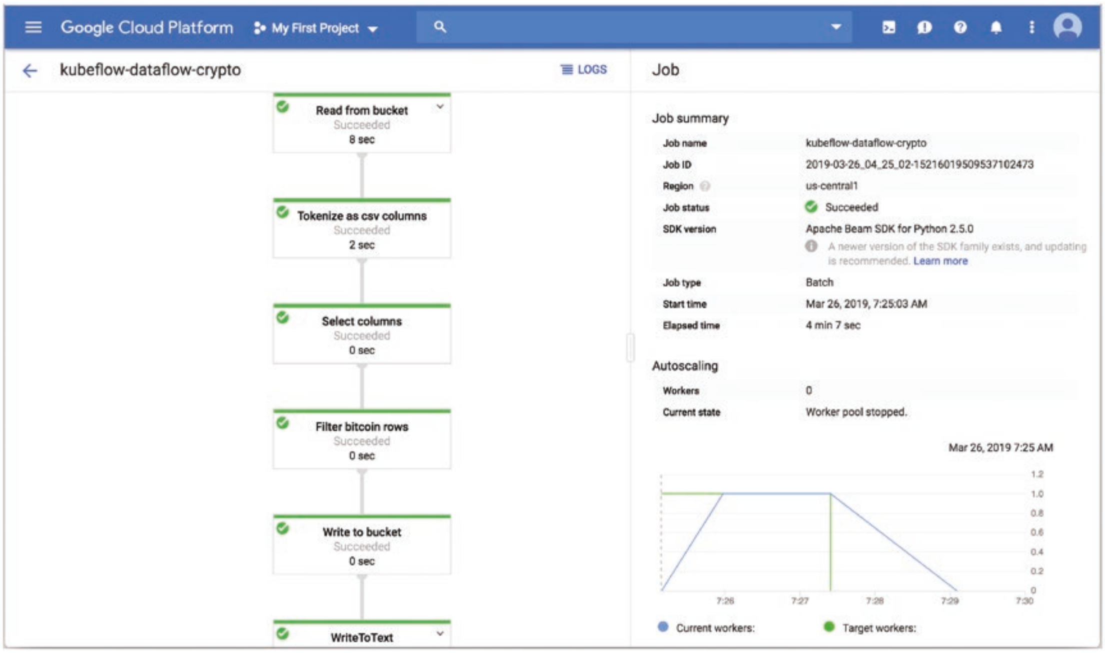


```
Activated service account credentials for: [kubeflow-deploy-admin@oceanic-sky-230504.iam.gserv
Updated property [core/project].
crypto
Model crypto already exists
Listed 0 items.

Creating version v1 from gs://kubeflow-ekaba-bucket/crypto/hyperparam/3/export/crypto/15536007
Creating version (this might take a few minutes)......
.......................................................................................
```

Figure 47-4. *Completed Pipeline run*

Completed Dataflow Pipeline: The completed run of the second component of the Pipeline, which is to transform the dataset with Cloud Dataflow, is illustrated in Figure 47-5.



kubeflow-dataflow-crypto

Read from bucket
Succeeded
8 sec

Tokenize as csv columns
Succeeded
2 sec

Select columns
Succeeded
0 sec

Filter bitcoin rows
Succeeded
0 sec

Write to bucket
Succeeded
0 sec

WriteToText

Job

Job summary

Job name	kubeflow-dataflow-crypto
Job ID	2019-03-26_04_25_02-15216019509537102473
Region	us-central1
Job status	Succeeded
SDK version	Apache Beam SDK for Python 2.5.0 A newer version of the SDK family exists, and updating is recommended. Learn more
Job type	Batch
Start time	Mar 26, 2019, 7:25:03 AM
Elapsed time	4 min 7 sec

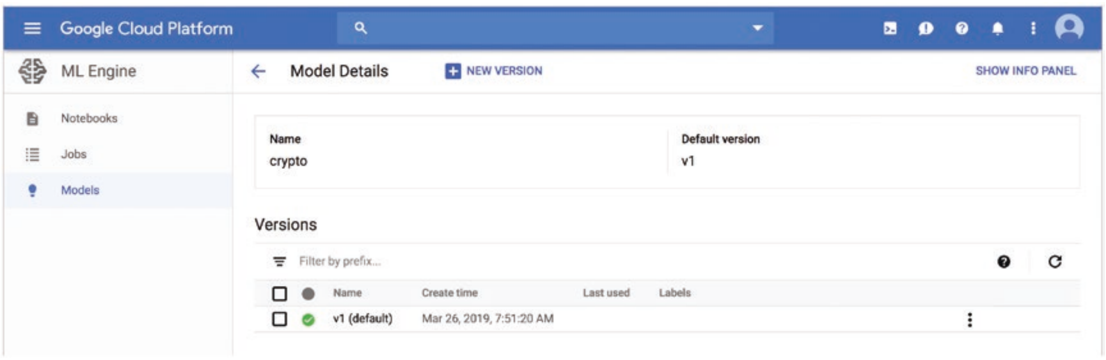
Autoscaling

Workers	0
Current state	Worker pool stopped.

Mar 26, 2019 7:25 AM

Figure 47-5. Completed Dataflow run

Deployed model on Cloud MLE: The deployed model on Cloud MLE, which is the fifth component of the Pipeline, is illustrated in Figure 47-6.



Model Details

Name	Default version
crypto	v1

Versions

	Name	Create time	Last used	Labels
●	v1 (default)	Mar 26, 2019, 7:51:20 AM		

Figure 47-6. Deployed model on Cloud MLE

Note Always remember to clean up cloud resources when they are no longer needed.

Delete Kubeflow: Run the script to delete the deployment.

```
# navigate to kubeflow app
cd ${KFAPP}

# run script to delete the deployment
${KUBEFLOW_SRC}/scripts/kfctl.sh delete all
```

Delete the Kubernetes cluster: Replace name with your own cluster name.

```
# delete the kubernetes cluster
gcloud container clusters delete ekaba-gke-cluster
```

This chapter covered building an end-to-end machine learning product as a containerized application on Kubernetes with Kubeflow and Kubeflow pipelines. Again, the code for this chapter may be accessed by cloning the book repository to the Cloud Shell.

This concludes this book.

Index