

Fairness and Explainable AI

- An End-to-end example on Tensorboard with Kubeflow

葉信和 / Hsin-Ho Yeh
Founder @ 信誠金融科技
hsinho.yeh@footprint-ai.com

About me

- 2020 - Present at 信誠金融科技
 - Shrimping: A data-sharing platform
 - <https://get-shrimping.footprint-ai.com>
 - Tintin: a machine learning platform for everyone
 - <https://get-tintin.footprint-ai.com>
- 2016 - 2020 at IglooInsure (16M+ in series A+ 2020)
 - Provide digital insurance for e-economic world
 - Funded in KUL, Headquartered in Singapore
 - First employee/ Engineering Lead / Regional Head/ Chief Engineer
- 2013 - 2016 at Studio Engineering @ hTC
 - Principal Engineer on Cloud Infrastructure Team
- 2009 - 2012 at IIS @ Academia Sinica
 - Computer vision, pattern recognition, and data mining
- CS@CCU, CS@NCKU alumni



課程綱要

- 課前知識
- 概念簡介
- 環境介紹
- 詞彙定義
- 範例練習
- 問與答

課前知識

- Be comfortable with UNIX command line
 - Navigating directories with ``cd`` or ``tree``
 - Editing files, like ``vim``, ``nano``
 - Bash scripting, like `env` or looping
- Be an expert with ``Google``
 - <https://letmegooglethat.com/?q=you+can+google+it>
- It is totally OK if you don't know what is Container and Kubernetes

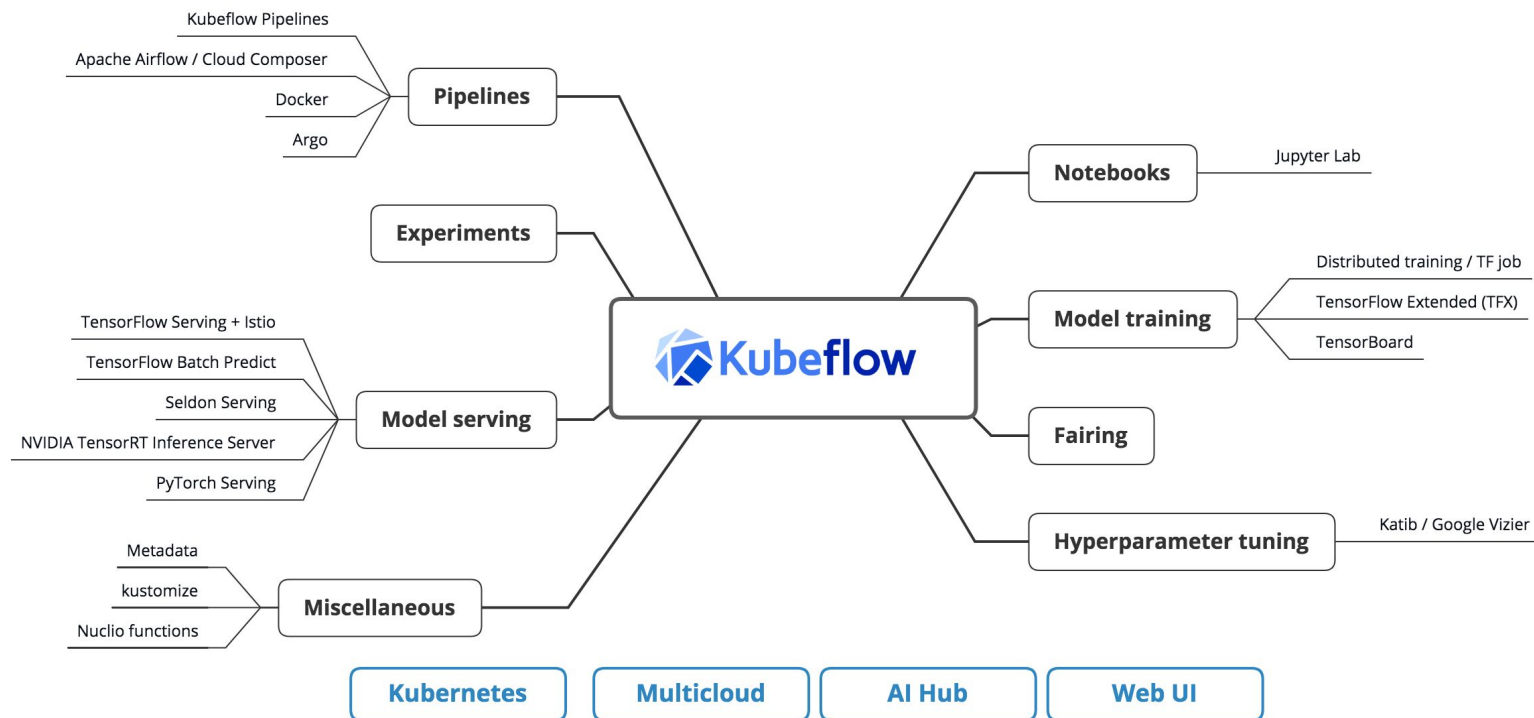
荀子《儒效篇》

「不聞不若聞之，聞之不若見之，見之不若知之，知之不若行之；學至于行之而止矣。」

```
git clone https://github.com/FootprintAI/kubeflow-workshop
```

Or [Click Me](#)

Kubeflow架構



Fairness and Explainable AI

Why explainable AI?

- 是「神之一手」還是「無法接受的錯誤」？

“It’s not a human move. I’ve never seen a human play this move.” (Fan Hui, 2016).

What is explainable AI?

1. Verification of the system

- A black box system is must not trust by default. For example, the use of models in health care should be interpreted and verified by medical experts in an absolute necessity.

2. Improvement of the system

- If the model is explainable, it is easy to perform weakness analysis for better model improvement.

3. Learning from the system

- Als are trained with millions of examples while experts are trained with a limited number of examples. With explainable AI, we are able to acquire new insights from the distilled knowledge.

4. Compliance to legislation

Art. 22 GDPR Automated individual decision-making, including profiling

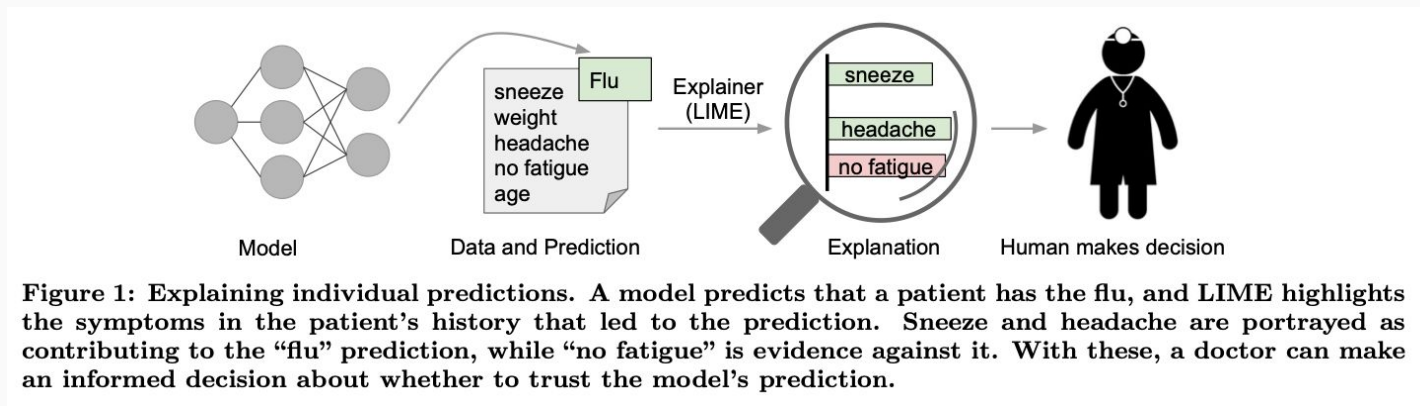
1. The data subject shall have the right not to be subject to a decision based solely on automated processing, including profiling, which produces legal effects concerning him or her or similarly significantly affects him or her.
2. Paragraph 1 shall not apply if the decision:
 - (a) is necessary for entering into, or performance of, a contract between the data subject and a data controller;
 - (b) is authorised by Union or Member State law to which the controller is subject and which also lays down suitable measures to safeguard the data subject's rights and freedoms and legitimate interests; or
 - (c) is based on the data subject's explicit consent.
3. In the cases referred to in points (a) and (c) of paragraph 2, the data controller shall implement suitable measures to safeguard the data subject's rights and freedoms and legitimate interests, at least the right to obtain human intervention on the part of the controller, to express his or her point of view and to contest the decision.
4. Decisions referred to in paragraph 2 shall not be based on special categories of personal data referred to in Article 9(1), unless point (a) or (g) of Article 9(2) applies and suitable measures to safeguard the data subject's rights and freedoms and legitimate interests are in place.

對個人有法律或重大影響的決定，不得基於 (Article 9 規定的) 個人種族、政治立場、宗教與哲學信仰、商業關係這些個人資料[2]

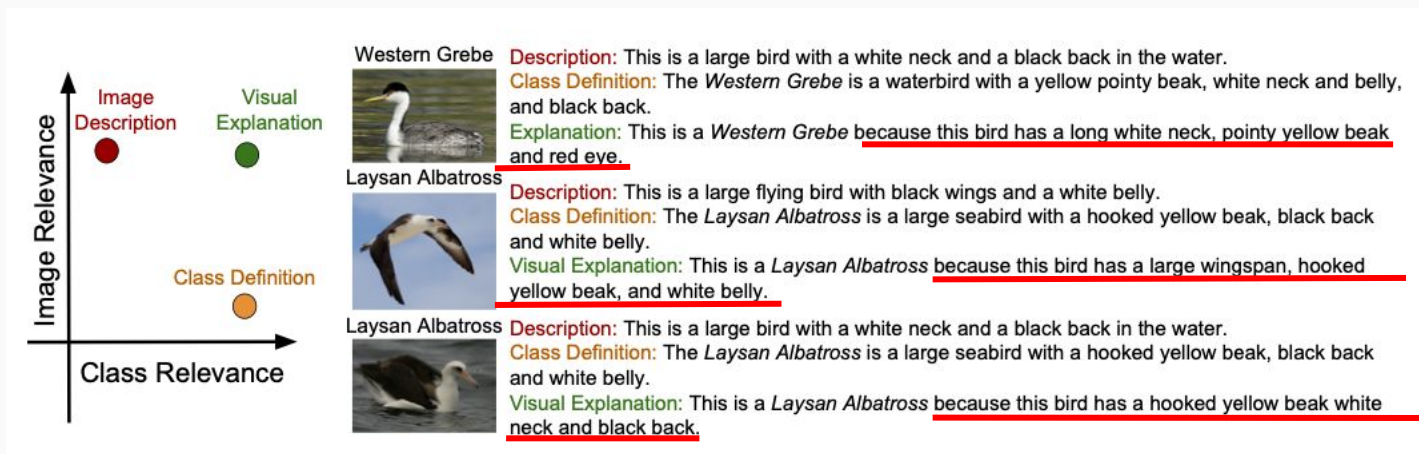
[1] <https://gdpr-info.eu/art-22-gdpr/>

[2] <https://medium.com/trustableai/歐盟-gdpr-生效對機器學習應用的影響d153de909e4f>

- Highlight the symptoms that led to the prediction.

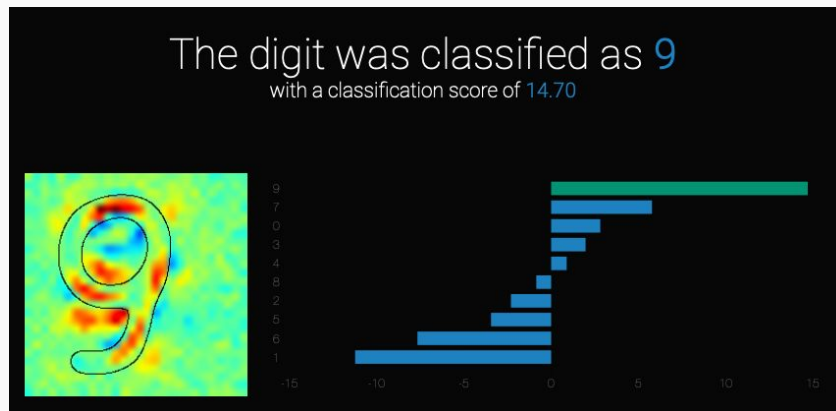
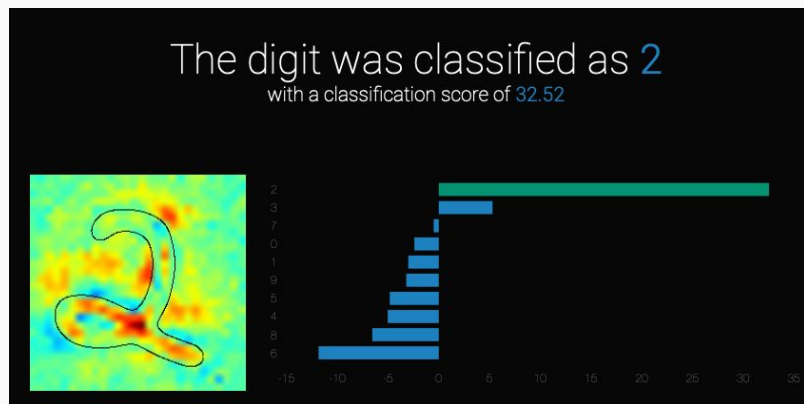


- Generating Visual Explanations



Explainable AI Examples (3/3)

- Heatmap to support the decision in digital recognition.



Ref: <http://heatmapping.org>

What is fairness AI?

1. Fairness of a system

- A particular problem is fit to be approximated or decided upon by an ML system

2. Fairness of a result

- do the outputs of the algorithm actually correspond to what would constitute a fair response by a human.

3. Fairness of an approach

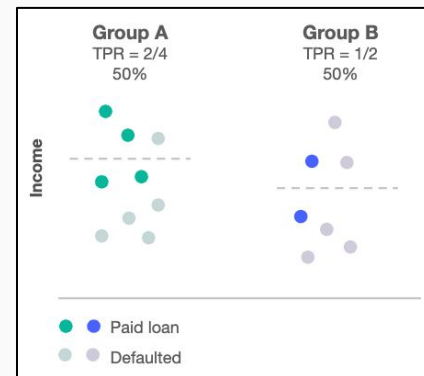
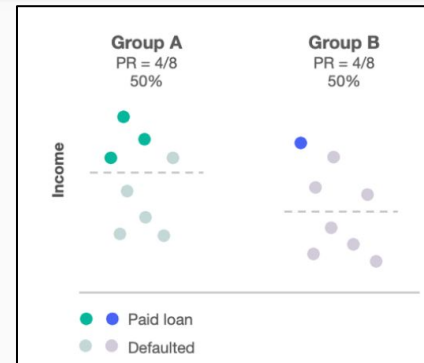
- White box models
 - i. Approaching fairness by developing training methods that aim to produce interpretable ML models.
 - ii. Limitations:
 1. Not feasible for complex models (e.g. deep learning models)
 2. Fairness is bound by the ability for a subject to meaningfully understand.
- Mathematical Fairness

What is fairness AI?

- Mathematical Fairness

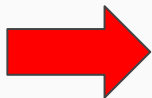
- Demographic parity (top)
 - The proportion of each segment of a protected class (e.g. gender) should receive the positive outcome at equal rates.
 - Best for university admission.
- Individual fairness
 - Similar individual are treated similarly
- Equal opportunity (bottom)
 - Each group should get the positive outcome at equal rates.
 - Best for fraud detection

- [Interactive website](#)

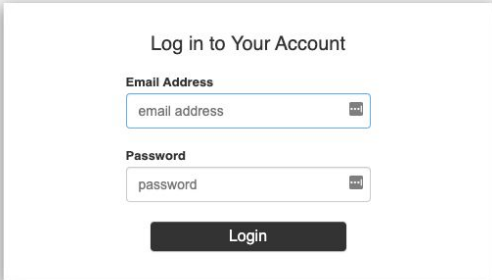
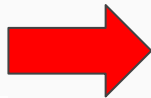


環境介紹

1. `kubectl port-forward
svc/istio-ingressgateway -n istio-system
8080:80 --address 0.0.0.0`



2. Open `http://localhost:8080`

A screenshot of a web login form titled "Log in to Your Account". It features two input fields: "Email Address" with a placeholder "email address" and a password icon, and "Password" with a placeholder "password" and a password icon. Below the fields is a black "Login" button.

Log in to Your Account

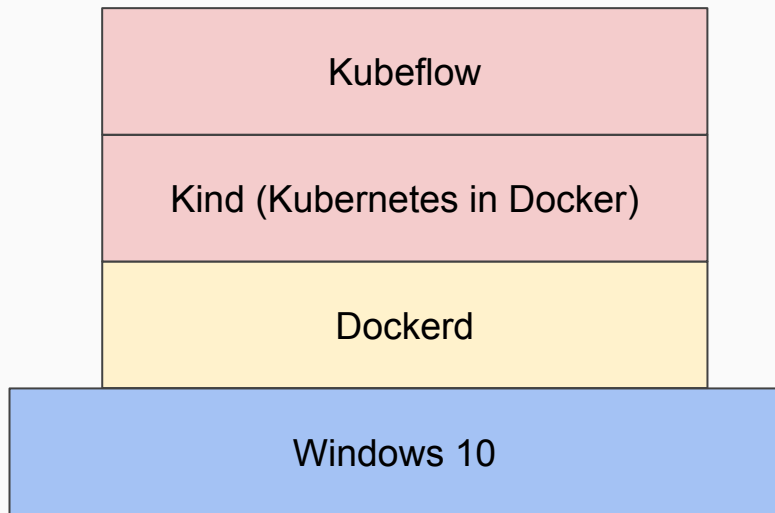
Email Address

email address

Password

password

Login



Wait! 所以我說那個帳號密碼呢?

Account: user@example.com

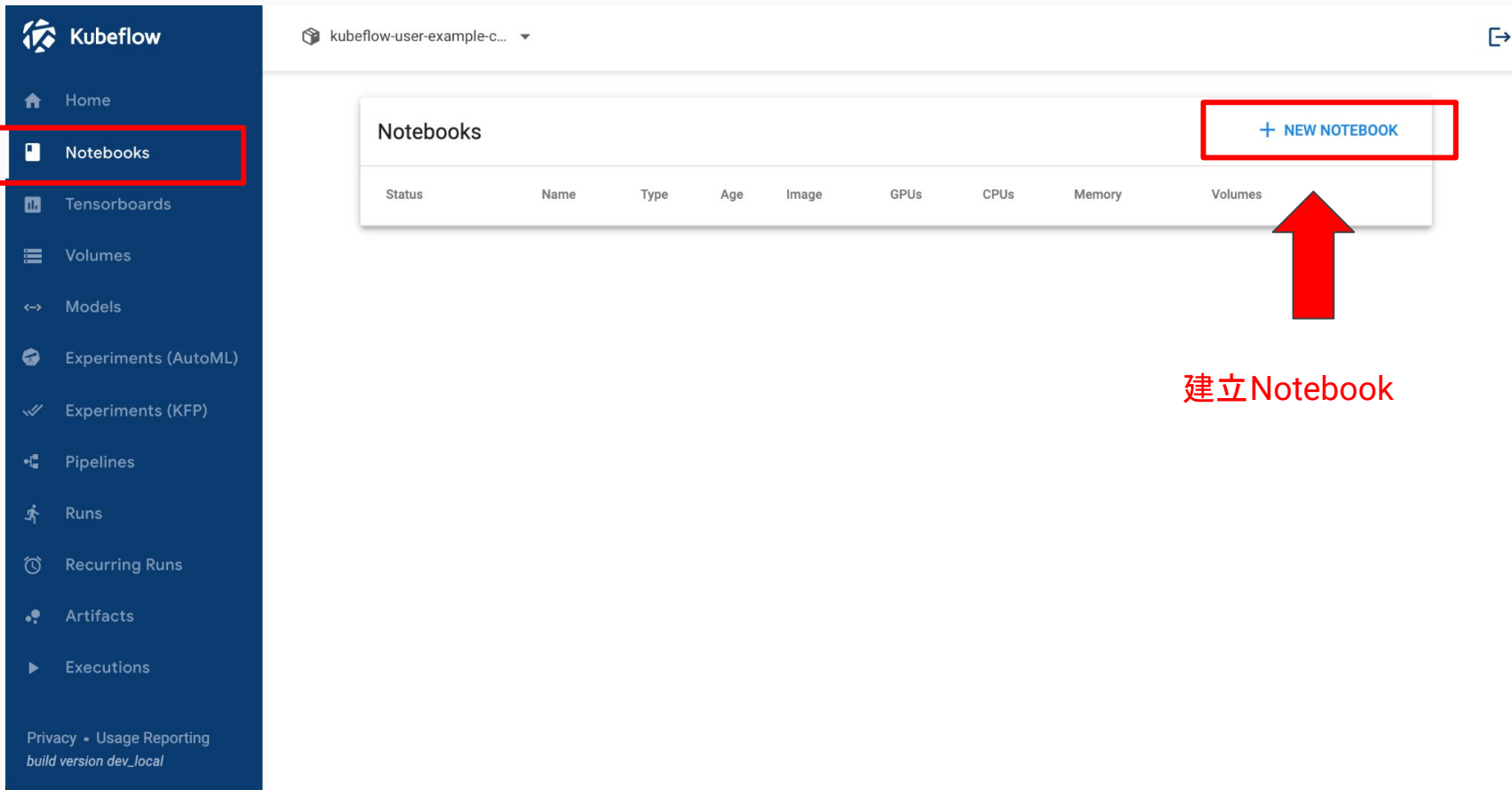
Password: 12341234

```
// 查看所有namespace  
Kubectl get namespaces
```

```
// 查看kubeflow中所有運行的Pod  
kubectl get pods -n kubeflow
```

```
// 查看目前使用者運行的Pod  
kubectl get pods -n kubeflow-user-example-com
```

Step1: 開啟Notebook作為開發環境 (1/3)



The screenshot displays the Kubeflow dashboard interface. On the left, a dark blue sidebar contains navigation links: Home, Notebooks, Tensorboards, Volumes, Models, Experiments (AutoML), Experiments (KFP), Pipelines, Runs, Recurring Runs, Artifacts, and Executions. The 'Notebooks' link is highlighted with a red rectangle. The main content area shows a 'Notebooks' section with a table header including Status, Name, Type, Age, Image, GPUs, CPUs, Memory, and Volumes. A '+ NEW NOTEBOOK' button is located in the top right corner of this section, also highlighted with a red rectangle. A large red arrow points from the text '建立Notebook' (Create Notebook) below the button to the button itself.

Kubeflow

kubeflow-user-example-c...

Notebooks


+ NEW NOTEBOOK

Status	Name	Type	Age	Image	GPUs	CPUs	Memory	Volumes
--------	------	------	-----	-------	------	------	--------	---------

建立Notebook

Privacy • Usage Reporting
build version dev_local

Step1: 開啟Notebook作為開發環境 (2/3)

 Kubeflow

Home

Notebooks

Tensorboards

Volumes

Models

Experiments (AutoML)

Experiments (KFP)

Pipelines

Runs

Recurring Runs

Artifacts

Executions

Manage Contributors

kubeflow-user-example-c...

Specify the name of the Notebook Server and the Namespace it will belong to.

Name

demo

Namespace

kubeflow-user-example-com

Image

A starter Jupyter Docker Image with a baseline deployment of TensorFlow and PyTorch ML packages

☐ Custom Image

jupyterlab

1

2

Image

j1r0q0g6/notebooks/notebook-servers/jupyter-tensorflow-full:v1.4

Advanced Options

CPU / RAM

Specify the total amount of CPU and RAM reserved by your Notebook Server. For CPU-intensive workloads, you can choose more than 1 CPU (e.g. 1.5).

Requested CPUs

0.5

Requested memory in Gi

1

給定名稱以及指定其 CPU/Memory

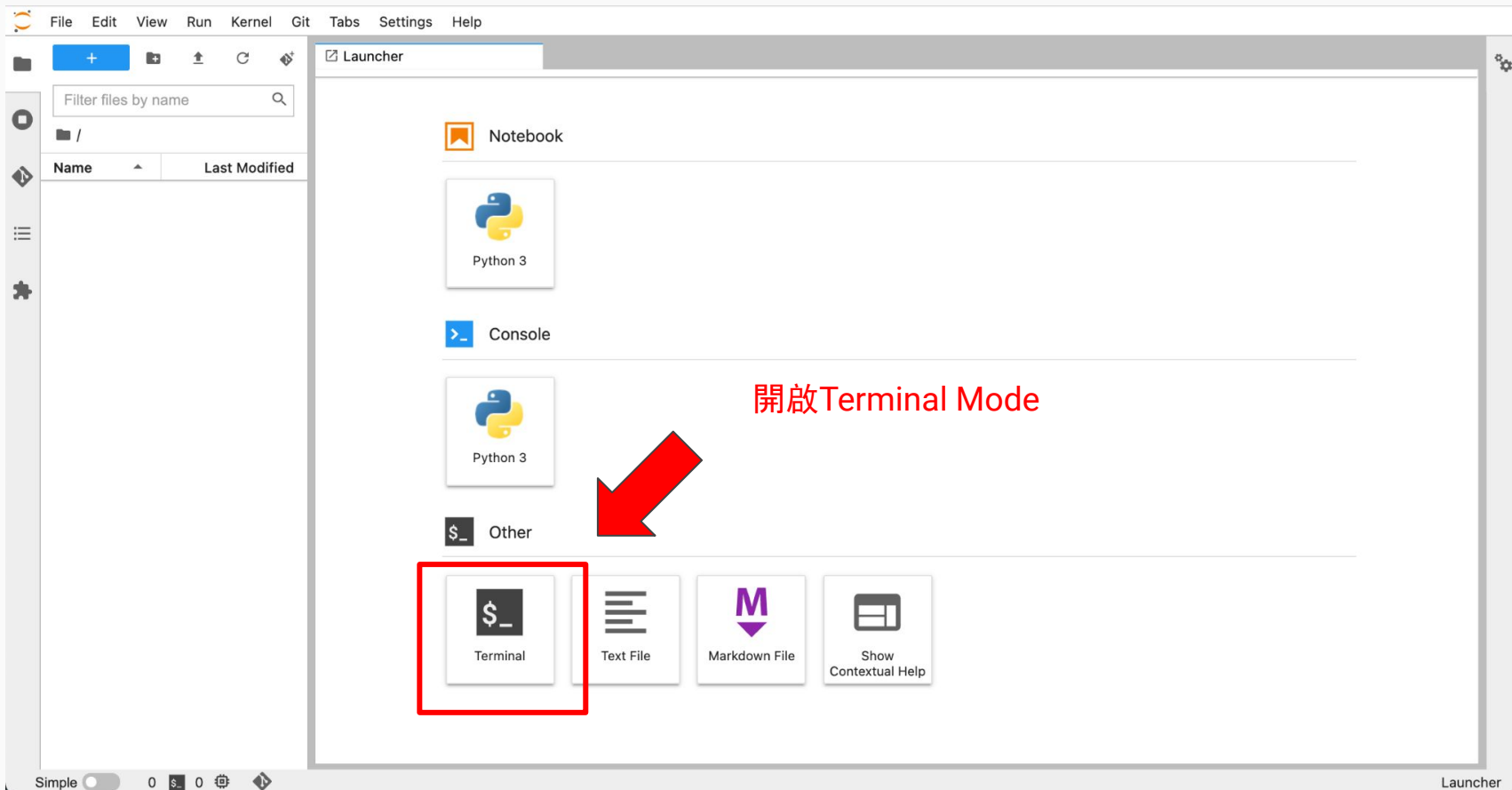
Step1: 開啟Notebook作為開發環境 (3/3)

The image shows the Kubeflow web interface. On the left is a dark blue sidebar with navigation links: Home, Notebooks (selected), Tensorboards, Volumes, Models, Experiments (AutoML), Experiments (KFP), Pipelines, Runs, Recurring Runs, and Artifacts. The main content area has a header with 'kubeflow-user-example-c...' and a '+ NEW NOTEBOOK' button. Below is a 'Notebooks' table with columns: Status, Name, Type, Age, Image, GPUs, CPUs, Memory, Volumes, and actions. Two notebooks are listed: 'demo1' (jupyter-scipy:v1.4, 20 hours ago) and 'demo2' (jupyter-tensorflow-full:v1.4, 2 hours ago). The 'demo2' row's 'CONNECT' button is highlighted with a red box, and a large red arrow points from the text '連線Notebook' below to this button.

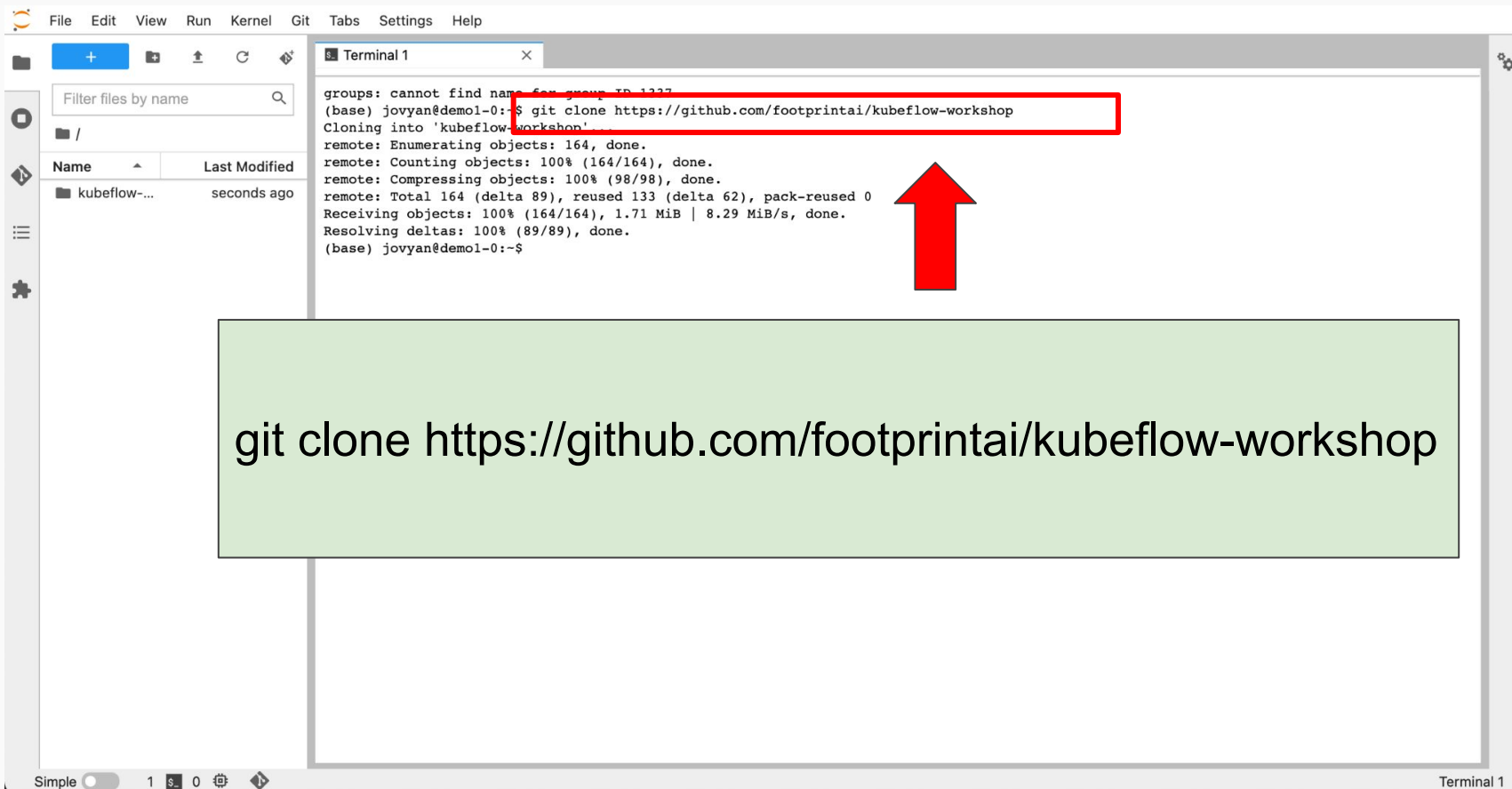
Status	Name	Type	Age	Image	GPUs	CPUs	Memory	Volumes	
✓	demo1		20 hours ago	jupyter-scipy:v1.4	0	0.5	1Gi	⋮	CONNECT
✓	demo2		2 hours ago	jupyter-tensorflow-full:v1.4	0	0.5	1Gi		CONNECT

連線Notebook

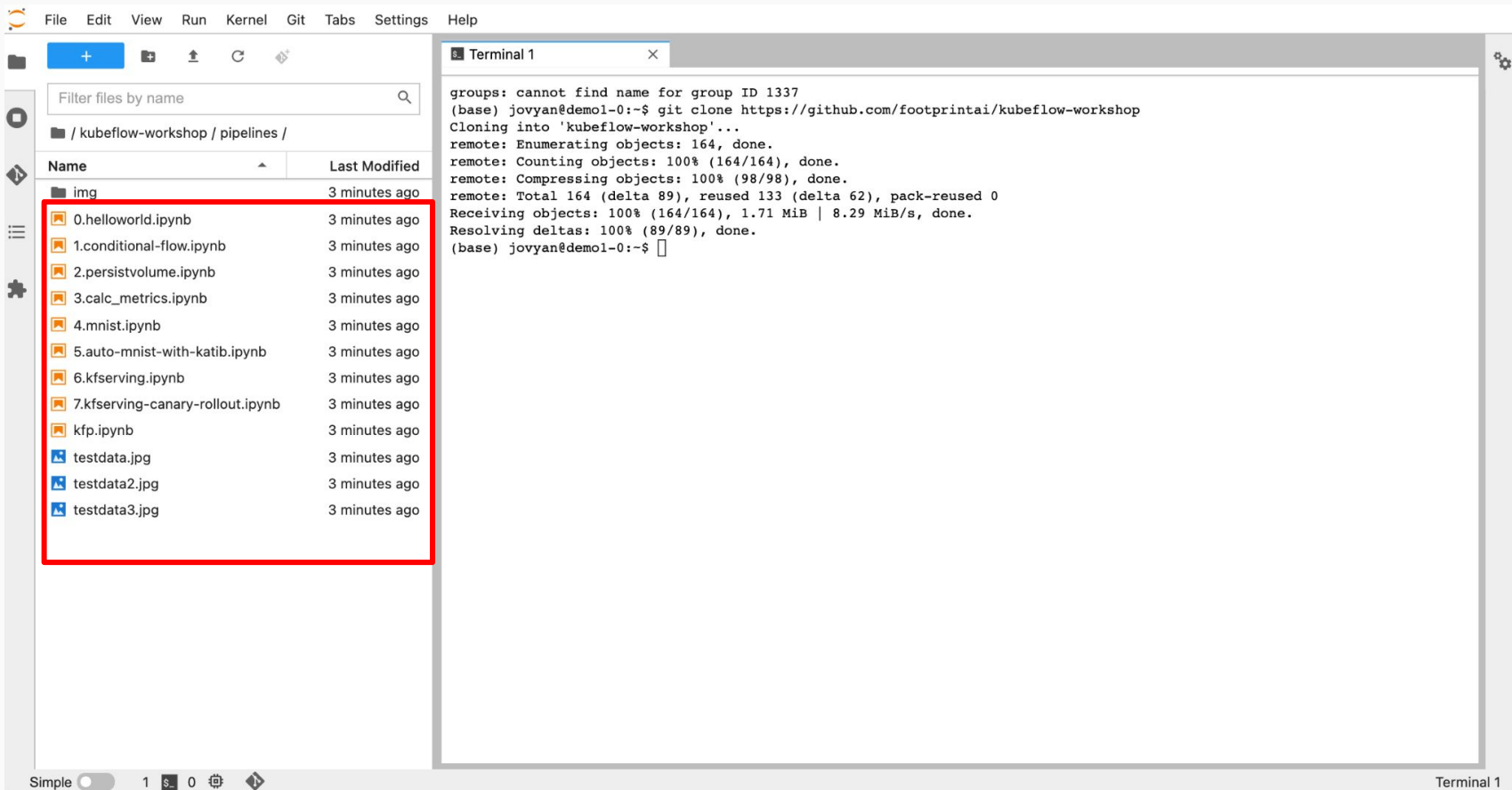
Step2: 開啟Terminal下載範例程式(1/3)



Step2: 開啟Terminal下載範例程式(2/3)



Step2: 開啟Terminal下載範例程式(3/3)



The screenshot shows a JupyterLab environment. On the left, the file browser displays the directory `/ kubeflow-workshop / pipelines /`. A red box highlights the following files:

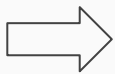
Name	Last Modified
img	3 minutes ago
0.helloworld.ipynb	3 minutes ago
1.conditional-flow.ipynb	3 minutes ago
2.persistvolume.ipynb	3 minutes ago
3.calc_metrics.ipynb	3 minutes ago
4.mnist.ipynb	3 minutes ago
5.auto-mnist-with-katib.ipynb	3 minutes ago
6.kfserving.ipynb	3 minutes ago
7.kfserving-canary-rollout.ipynb	3 minutes ago
kfp.ipynb	3 minutes ago
testdata.jpg	3 minutes ago
testdata2.jpg	3 minutes ago
testdata3.jpg	3 minutes ago

On the right, the terminal window (Terminal 1) shows the output of a `git clone` command:

```
(base) jovyan@demol-0:~$ git clone https://github.com/footprintai/kubeflow-workshop
Cloning into 'kubeflow-workshop'...
remote: Enumerating objects: 164, done.
remote: Counting objects: 100% (164/164), done.
remote: Compressing objects: 100% (98/98), done.
remote: Total 164 (delta 89), reused 133 (delta 62), pack-reused 0
Receiving objects: 100% (164/164), 1.71 MiB | 8.29 MiB/s, done.
Resolving deltas: 100% (89/89), done.
(base) jovyan@demol-0:~$
```

詞彙說明

詞彙說明

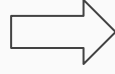
[illegible]

Pipeline Code

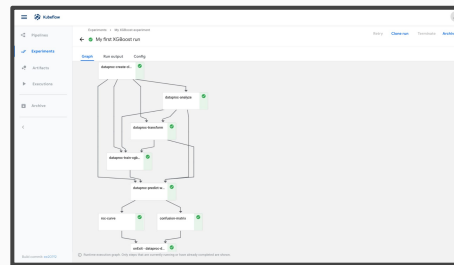


編譯後

Workflow Resource

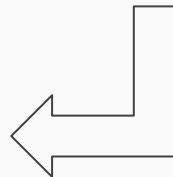
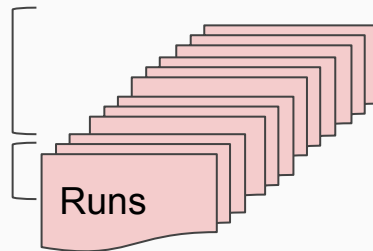


建立 Pipeline



實驗1 (Experiment1)

實驗2 (Experiment2)



Create Run

範例1 Hello World!

Step3: 編譯helloworld.ipynb (1/2)

The screenshot shows the JupyterLab interface. On the left, a sidebar displays a file explorer for the 'kubeflow-workshop / pipelines /' directory. The file '0.helloworld.ipynb' is selected. The main editor window shows the code from '0.helloworld.ipynb'. The code defines a pipeline with two tasks, 'echo1_op' and 'echo2_op', and a compiler to generate workflow artifacts. A red box highlights the final line of code: `kfp.compiler.Compiler().compile(execution_order_pipeline, 'helloworld.zip')`.

```
[ ]: with open("requirements.txt", "w") as f:
      f.write("kfp==1.8.9\n")

!pip install -r requirements.txt --upgrade --user

[ ]: import kfp
      from kfp import dsl

      def echo1_op(text1):
          return dsl.ContainerOp(
              name='echo1',
              image='library/bash:4.4.23',
              command=['sh', '-c'],
              arguments=['echo "$0"', text1])

      def echo2_op(text2):
          return dsl.ContainerOp(
              name='echo2',
              image='library/bash:4.4.23',
              command=['sh', '-c'],
              arguments=['echo "$0"', text2])

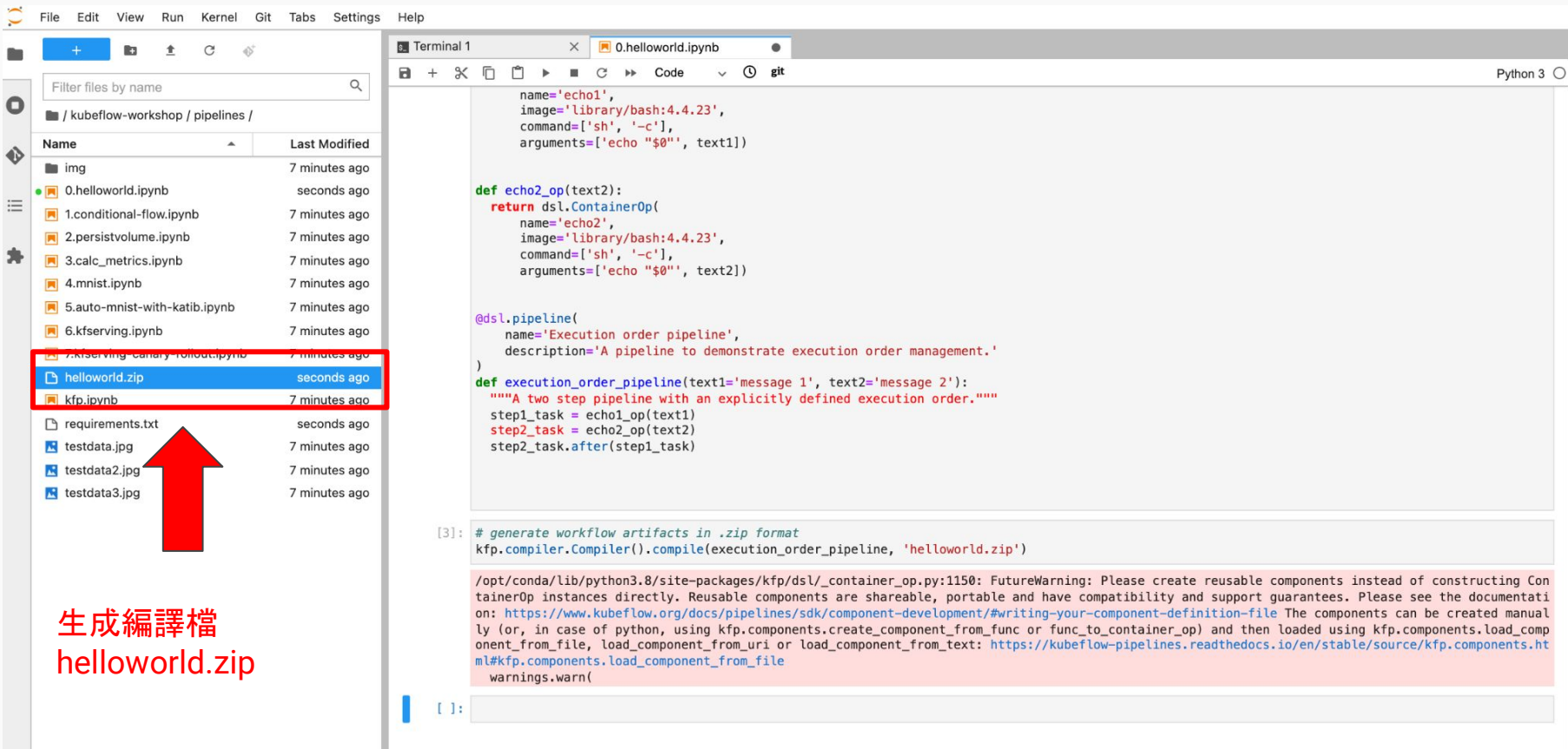
      @dsl.pipeline(
          name='Execution order pipeline',
          description='A pipeline to demonstrate execution order management.'
      )

      def execution_order_pipeline(text1='message 1', text2='message 2'):
          """A two step pipeline with an explicitly defined execution order."""
          step1_task = echo1_op(text1)
          step2_task = echo2_op(text2)
          step2_task.after(step1_task)

[ ]: # generate workflow artifacts in .zip format
      kfp.compiler.Compiler().compile(execution_order_pipeline, 'helloworld.zip')

[ ]:
```

Step3: 編譯helloworld.ipynb (2/2)



The screenshot shows the Kubeflow Pipelines interface. On the left, the file explorer displays a list of files in the '/ kubeflow-workshop / pipelines /' directory. The file 'helloworld.zip' is highlighted with a red box, and a red arrow points to it. The right pane shows the code editor for '0.helloworld.ipynb', which contains the following Python code:

```
name='echo1',
image='library/bash:4.4.23',
command=['sh', '-c'],
arguments=['echo "$0"', text1])

def echo2_op(text2):
    return dsl.ContainerOp(
        name='echo2',
        image='library/bash:4.4.23',
        command=['sh', '-c'],
        arguments=['echo "$0"', text2])

@dsl.pipeline(
    name='Execution order pipeline',
    description='A pipeline to demonstrate execution order management.'
)
def execution_order_pipeline(text1='message 1', text2='message 2'):
    """A two step pipeline with an explicitly defined execution order."""
    step1_task = echo1_op(text1)
    step2_task = echo2_op(text2)
    step2_task.after(step1_task)
```


Below the code editor, the terminal output shows the command to generate workflow artifacts in .zip format:

```
[3]: # generate workflow artifacts in .zip format
kfp.compiler.Compiler().compile(execution_order_pipeline, 'helloworld.zip')
```

The terminal output also displays a FutureWarning message from the kfp/dsl/_container_op.py file, advising to create reusable components instead of constructing ContainerOp instances directly.

生成編譯檔
helloworld.zip

Step4: 建立Pipeline (1/7)

 **Kubeflow**

Home

Notebooks

Tensorboards

Volumes

Models

Experiments (AutoML)

Experiments (KFP)

Pipelines

Runs

Recurring Runs

Artifacts

Executions

Manage Contributors

kubeflow-user-example-c...

Filter pipelines

Pipeline name

Description

Upload

[Tutorial] V2 lightweight Python com...

[source code](#) Shows different component input and output options for KFP v2 components.

11/30/2021, 1:02:25 PM

[Tutorial] DSL - Control structures

[source code](#) Shows how to use conditional execution and exit handlers. This pipeline will randomly fail to demonstr...

11/30/2021, 1:02:24 PM

[Tutorial] Data passing in python co...

[source code](#) Shows how to pass data between python components.

11/30/2021, 1:02:23 PM

[Demo] TFX - Taxi tip prediction mod...

[source code](#) [GCP Permission requirements](#). Example pipeline that does classification with model analysis based on...

11/30/2021, 1:02:22 PM

[Demo] XGBoost - Iterative model tra...

[source code](#) This sample demonstrates iterative training using a train-eval-check recursive loop. The main pipeline ...

11/30/2021, 1:02:21 PM

Rows per page: 10

+ Upload pipeline

Refresh

Delete

32

Step4: 建立Pipeline (2/7)

Kubeflow

kubeflow-user-example-c...

Pipeline Versions

← Upload Pipeline or Pipeline Version

☒ Create a new pipeline ☐ Create a new pipeline version under an existing pipeline

Upload pipeline with the specified package.

Pipeline Name
0.helloworld

Pipeline Description
0.helloworld

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

For expected file format, refer to [Compile Pipeline Documentation](#).

☒ Upload a file
File
helloworld (19).zip [Choose file](#)

☐ Import by url
Package Url

Code Source (optional)

[Create](#) [Cancel](#)

1.輸入Pipeline名稱

2.指定編譯完後 Zip位置

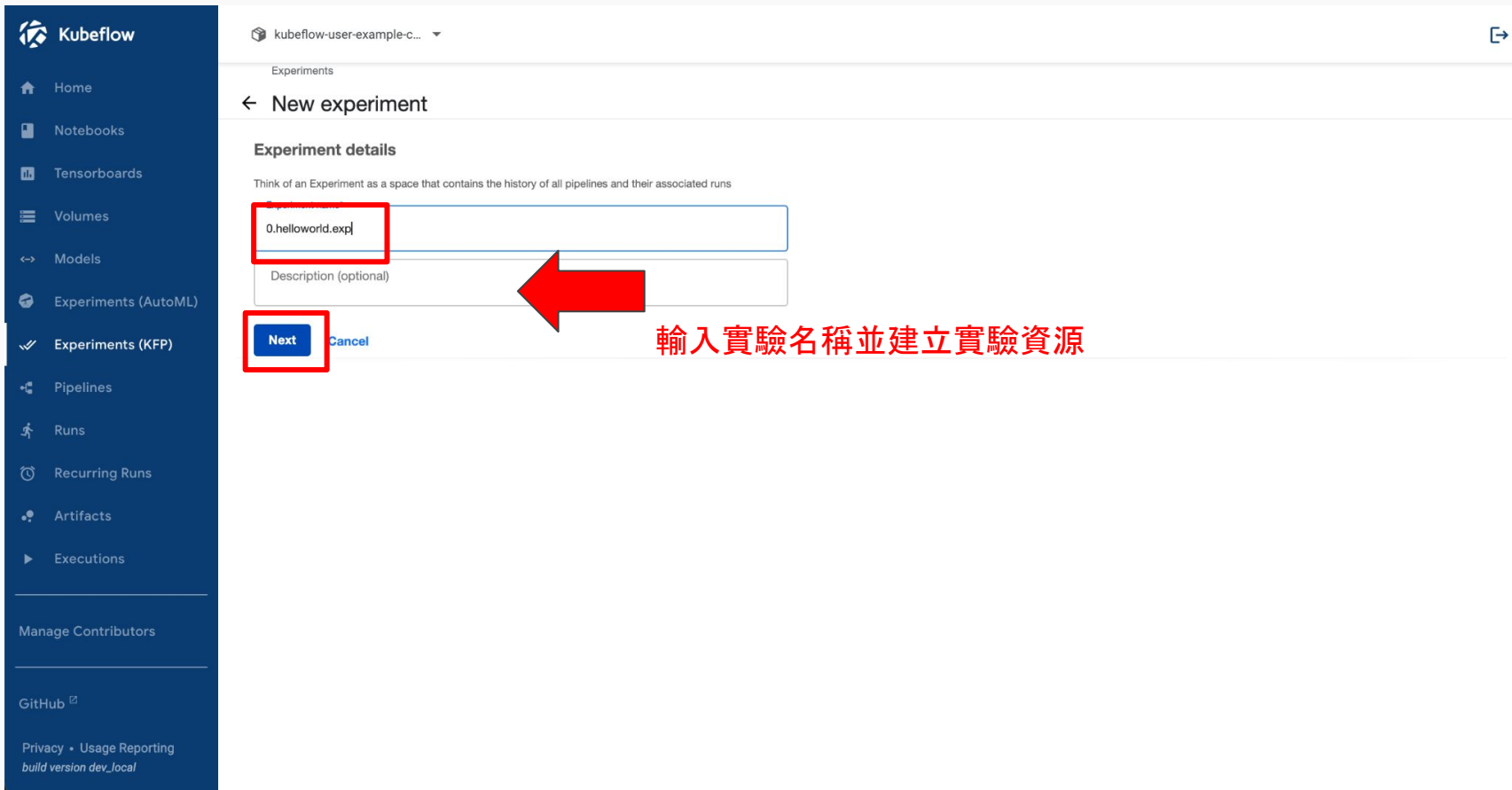
3.建立

Step4: 建立Pipeline (3/7)

The screenshot displays the Kubeflow Pipelines web interface. On the left is a dark blue sidebar with navigation links: Home, Notebooks, Tensorboards, Volumes, Models, Experiments (AutoML), Experiments (KFP), Pipelines, Runs, Recurring Runs, Artifacts, and Executions. The main content area shows the 'Pipelines' section for a user named 'kubeflow-user-example-c...'. The selected pipeline is '0.helloworld (0.helloworld)'. At the top right of the pipeline view are three buttons: '+ Create run', '+ Upload version', and '+ Create experiment' (which is highlighted with a red rectangular box). To the right of the '+ Create experiment' button is a 'Delete' link. Below the buttons, the pipeline graph is shown in 'Graph' view, featuring two steps: 'echo1' and 'echo2', connected by a downward arrow. A 'Simplify Graph' toggle is visible above the graph. At the bottom left, a 'Summary' panel is partially visible, showing the pipeline's ID, version, and source.

建立實驗以便管理 Run

Step4: 建立Pipeline (4/7)



Kubeflow

Home

Notebooks

Tensorboards

Volumes

Models

Experiments (AutoML)

Experiments (KFP)

Pipelines

Runs

Recurring Runs

Artifacts

Executions

Manage Contributors

GitHub

Privacy • Usage Reporting
build version dev_local

kubeflow-user-example-c...

Experiments

← New experiment

Experiment details

Think of an Experiment as a space that contains the history of all pipelines and their associated runs

0.helloworld.exp

Description (optional)

Next Cancel

輸入實驗名稱並建立實驗資源

Step4: 建立Pipeline (5/7)

Kubeflow

kubeflow-user-example-c...

Run details

Pipeline*
0.helloworld [Choose](#)

Pipeline Version*
0.helloworld [Choose](#)

Run name*
Run of 0.helloworld (1e261)

Description (optional)

This run will be associated with the following experiment

Experiment*
0.helloworld.exp [Choose](#)

This run will use the following Kubernetes service account. ?

Service Account (Optional)

Run Type

☒ One-off ☐ Recurring

Run parameters

Specify parameters required by the pipeline

text1
message 1

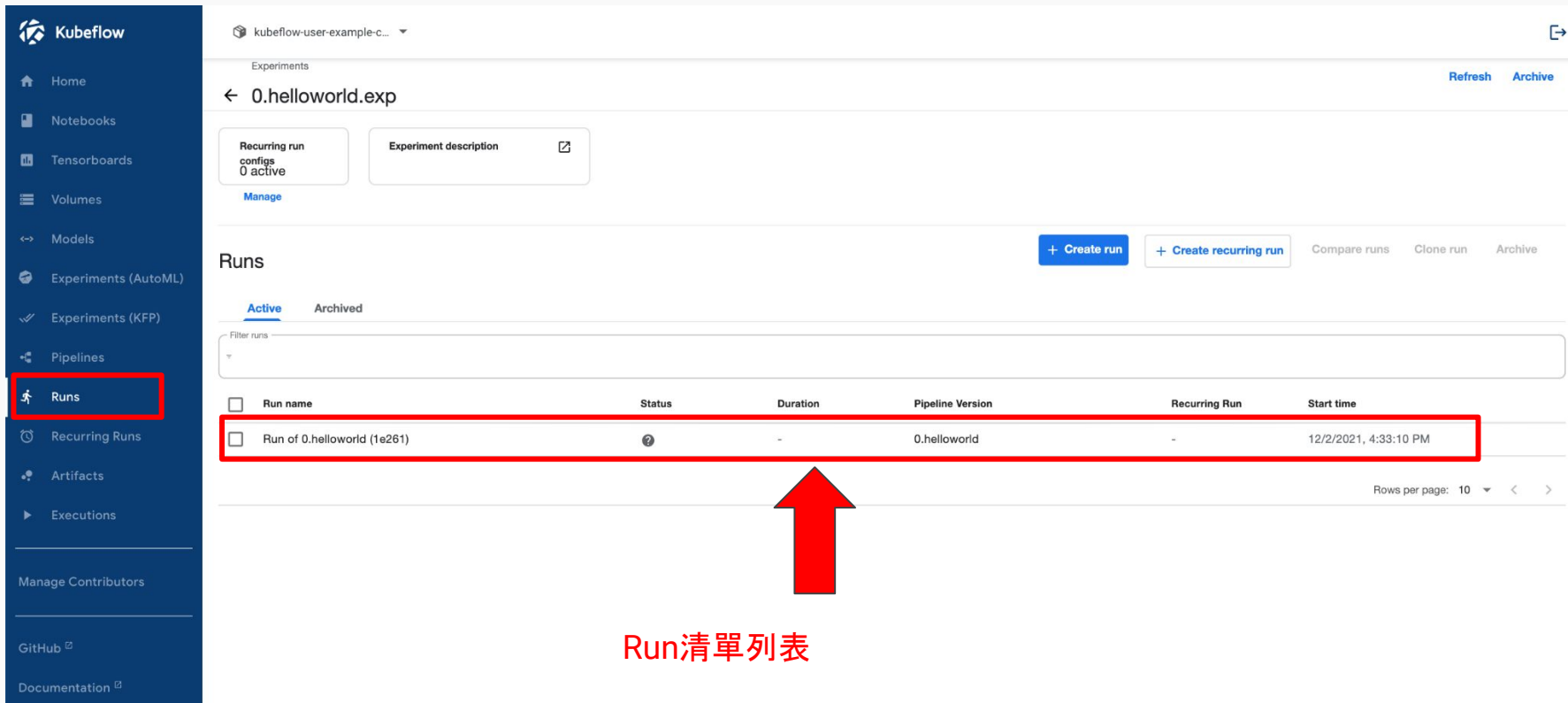
text2
message 2

[Start](#) [Skip this step](#)

1. 運行此Pipeline成為Run資源

2. 指定其歸屬的實驗資源

Step4: 建立Pipeline (6/7)



Kubeflow

Home

Notebooks

Tensorboards

Volumes

Models

Experiments (AutoML)

Experiments (KFP)

Pipelines

Runs

Recurring Runs

Artifacts

Executions

Manage Contributors

GitHub

Documentation

kubeflow-user-example-c...

Experiments

Refresh Archive

← 0.helloworld.exp

Recurring run
configs
0 active
Manage

Experiment description

Runs

+ Create run + Create recurring run Compare runs Clone run Archive

Active Archived

Filter runs

<input type="checkbox"/>	Run name	Status	Duration	Pipeline Version	Recurring Run	Start time
<input type="checkbox"/>	Run of 0.helloworld (1e261)	?	-	0.helloworld	-	12/2/2021, 4:33:10 PM

Rows per page: 10 < >

Run清單列表

Step4: 建立Pipeline (7/7)

The screenshot displays the Kubeflow console interface. On the left is a dark blue sidebar with navigation links: Home, Notebooks, Tensorboards, Volumes, Models, Experiments (AutoML), Experiments (KFP), Pipelines, Runs, Recurring Runs, Artifacts, Executions, Manage Contributors, and GitHub. The main content area shows the 'Run of 0.helloworld (1e261)' page. The 'Graph' tab is active, showing a pipeline with two steps: 'echo1' and 'echo2', both marked with green checkmarks. A red rectangle highlights this graph. To the right, a modal window titled 'execution-order-pipeline-fxxfn-4223123588' is open, showing the 'Input/Output' tab. This tab lists input parameters (text1: message 1), input artifacts, output parameters, and output artifacts. The 'main-logs' output artifact is highlighted with a red rectangle, showing a link to 'minio://mlpipeline/artifacts/execution-order-pipeline-qvlt5-2021/11/30/execution-order-pipeline-qvlt5-137025384/main.log' and the text 'message 1'. A large red arrow points from the text '運行結果輸出' (Output of execution results) to the highlighted output artifact section.

Kubeflow

Home
Notebooks
Tensorboards
Volumes
Models
Experiments (AutoML)
Experiments (KFP)
Pipelines
Runs
Recurring Runs
Artifacts
Executions
Manage Contributors
GitHub

kubeflow-user-example-c...

Experiments > 0.helloworld.exp

← Run of 0.helloworld (1e261)

Graph Run output Config

Simplify Graph

echo1

echo2

execution-order-pipeline-fxxfn-4223123588

Input/Output Visualizations Details Volumes Logs Pod Events ML Metadata

Input parameters

text1 message 1

Input artifacts

Output parameters

Output artifacts

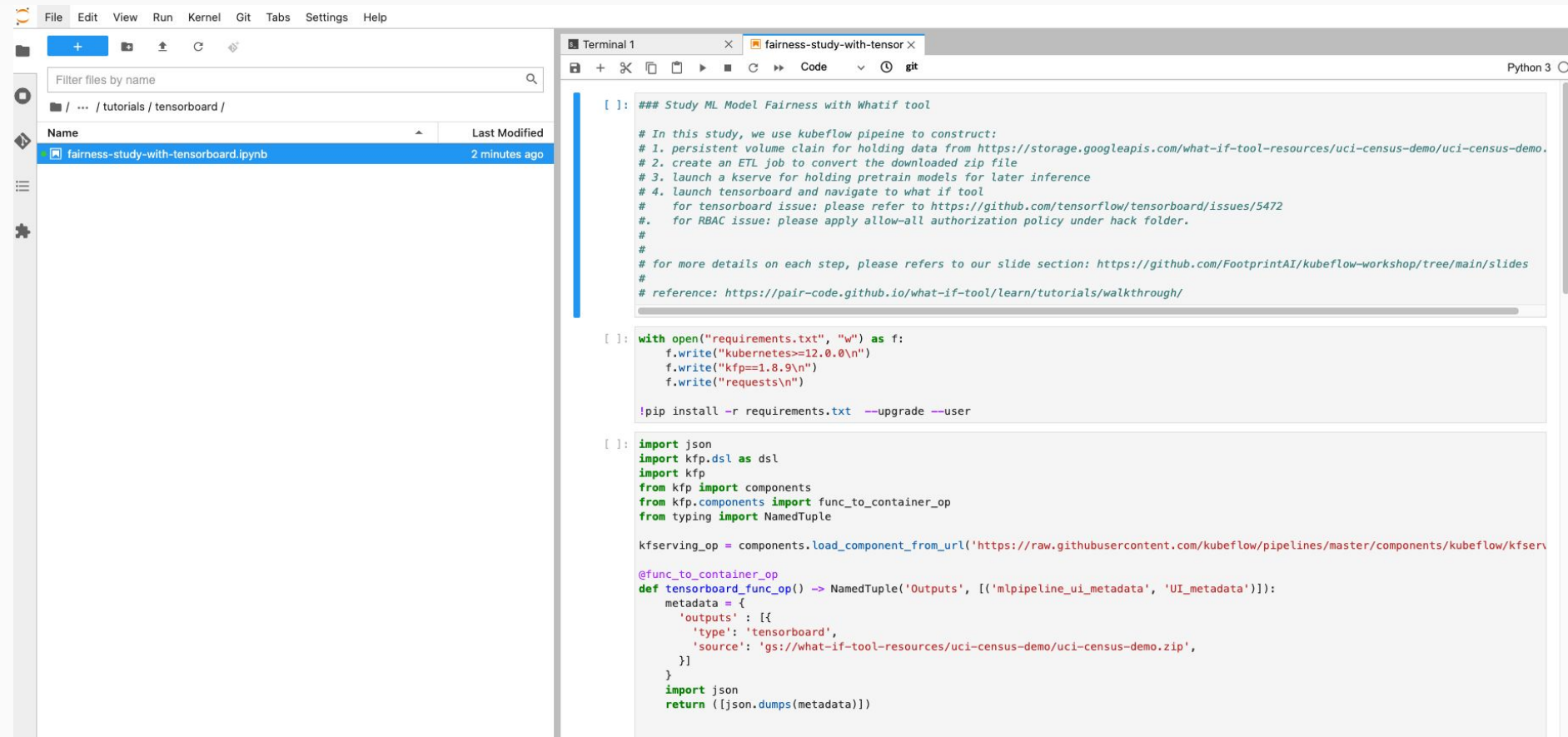
main-logs minio://mlpipeline/artifacts/execution-order-pipeline-qvlt5-2021/11/30/execution-order-pipeline-qvlt5-137025384/main.log View All

message 1

運行結果輸出

範例2 Fairness Study with What-If Tool and Tensorboard

Step5: 建立fairness pipeline (1/3)



The screenshot displays a JupyterLab environment. On the left, the file browser shows the directory structure: `/ ... / tutorials / tensorboard /`. A file named `fairness-study-with-tensorboard.ipynb` is listed, last modified 2 minutes ago. The main area on the right is a code editor for a Jupyter notebook cell. The cell has a title `### Study ML Model Fairness with Whatif tool` and contains the following content:

```
[ ]: ## In this study, we use kubeflow pipeline to construct:  
# 1. persistent volume claim for holding data from https://storage.googleapis.com/what-if-tool-resources/uci-census-demo/uci-census-demo.  
# 2. create an ETL job to convert the downloaded zip file  
# 3. launch a kserve for holding pretrain models for later inference  
# 4. launch tensorboard and navigate to what if tool  
# for tensorboard issue: please refer to https://github.com/tensorflow/tensorboard/issues/5472  
# for RBAC issue: please apply allow-all authorization policy under hack folder.  
#  
# for more details on each step, please refers to our slide section: https://github.com/FootprintAI/kubeflow-workshop/tree/main/slides  
#  
# reference: https://pair-code.github.io/what-if-tool/learn/tutorials/walkthrough/  
  
[ ]: with open("requirements.txt", "w") as f:  
    f.write("kubernetes>=12.0.0\n")  
    f.write("kfp==1.8.9\n")  
    f.write("requests\n")  
  
!pip install -r requirements.txt --upgrade --user  
  
[ ]: import json  
import kfp.dsl as dsl  
import kfp  
from kfp import components  
from kfp.components import func_to_container_op  
from typing import NamedTuple  
  
kfserving_op = components.load_component_from_url('https://raw.githubusercontent.com/kubeflow/pipelines/master/components/kubeflow/kfserv  
  
@func_to_container_op  
def tensorboard_func_op() -> NamedTuple('Outputs', [(('mlpipeline_ui_metadata', 'UI_metadata'))]):  
    metadata = {  
        'outputs': [{  
            'type': 'tensorboard',  
            'source': 'gs://what-if-tool-resources/uci-census-demo/uci-census-demo.zip',  
        }]  
    }  
    import json  
    return ([json.dumps(metadata)])
```


Step5: 建立fairness pipeline (2/3)

ubeflow

Home

Notebooks

Tensorboards

Volumes

Models

Experiments (AutoML)

Experiments (KFP)

Pipelines

Runs

Recurring Runs

Artifacts

Executions

Manage Contributors

kubeflow-user-example-c...

Pipelines

← tensorboard-whatiftool (tensorboard-whatiftool_version_at_2021-12-22T08:14:50.210Z)


+ Create run + Upload version + Create experime

Graph YAML

☐ Simplify Graph

```
graph TD; mypvc --> etl; mypvc --> KFServing[Kubeflow - Serve Model using KFServing]; mypvc --> TensorboardFuncOp; etl --> KFServing; etl --> TensorboardFuncOp; KFServing --> TensorboardFuncOp; KFServing --> TensorboardFuncOp;
```

Step5: 建立fairness pipeline (3/3)

 Kubeflow

Home

Notebooks

Tensorboards

Volumes

Models

Experiments (AutoML)

Experiments (KFP)

Pipelines

Runs

Recurring Runs

Artifacts


Executions

Manage Contributors

Privacy • Usage Reporting
build version dev_local

kubeflow-user-example-c...

Experiments > uci

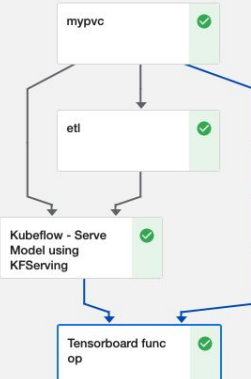
←  Run of tensorboard-whatiftool_version_at_2021-12-22T07:18:38.074Z (da9a8)

Graph

Run output

Config

☐ Simplify Graph



```
graph TD; mypvc[mypvc] --> etl[etl]; mypvc --> kfserving[Kubeflow - Serve Model using KFServing]; etl --> kfserving; kfserving --> tensorboard[Tensorboard func op];
```

✕

kfserving-pipeline-rc5qs-1149399848

Input/Output

Visualizations

Details

Volumes

Logs

Pod

Events

ML Metadata

Tensorboard

TF Image

TensorFlow 2.2.2

Start Tensorboard

Visualization Creator

create visualizations manually

Add visualizations to your own components following instructions in [Visualize Results in the Pipelines UI](#).

Step6: hack tensorboard (1/3)

(此步為進階功能, 若無法實作也無仿)

為了讓tensorboard可以查看本地端的資料, 我們得將其對應資源建立起:

1. 先尋找pvc的位置以及 viewer id
2. 修改至hack/tensorboard-use-local-volume.yaml底下
3. 刪掉舊有的viewer resource (hack/delete-all-viewer.sh)

```
root@instance-x:/home/hsinhoyeh# kubectl delete viewers -n kubeflow-user-example-com --all  
viewer.kubeflow.org "viewer-a4e50b4f89b636c7e2d12bebd3da533901b3ea8a" deleted
```

4. 然後kubectl apply -f hack/tensorboard-use-local-volume.yaml

```
root@instance-x:/home/hsinhoyeh# kubectl get viewer -n kubeflow-user-example-com
```

NAME	AGE
viewer-a4e50b4f89b636c7e2d12bebd3da533901b3ea8a	13s

```
root@instance-1:~# kubectl get pvc -n kubeflow-user-example-com
```

NAME	STATUS	VOLUME	CAPACITY	ACCESS MODES	STORAGECLASS	AGE
volumeop-sequential-mv8sz-newpvc	Bound	pvc-17431607-83bd-44f3-abe9-79b0108ae1cc	1Gi	RWO	standard	2d21h
workspace-demo1	Bound	pvc-cf2d97d1-0202-4e2e-bafe-f398562627e2	5Gi	RWO	standard	19h
workspace-demo2	Bound	pvc-f1489494-999e-461c-8fe0-96654b625104	5Gi	RWO	standard	58m
workspace-tester1	Bound	pvc-c5fa720d-3ae7-4a2e-95d7-7edc4f668d13	5Gi	RWO	standard	2d22h

Step6: hack tensorboard (2/3)

(此步為進階功能, 若無法實作也無仿)

修改tensorboard deployment launch script

1. 先找到deployment的名字 (`kubectl get deployment -n kubeflow-user-example-com`)

```
root@instance-x:/home/hsinhoyeh# kubectl get deployment -n kubeflow-user-example-com
NAME                                READY   UP-TO-DATE   AVAILABLE   AGE
ml-pipeline-ui-artifact             1/1     1             1           2d2h
ml-pipeline-visualizationserver     1/1     1             1           2d2h
uci-census-predictor-default-00001-deployment 1/1     1             1           66m
viewer-a4e50b4f89b636c7e2d12bebd3da533901b3ea8a-deployment 1/1     1             1           2m11s
```

2. 修改deployment (`kubectl edit deployment/<viewer-id> -n kubeflow-user-example-com`)

```
spec:
  containers:
  - args:
    - tensorboard
    - --logdir=/data
    - path_prefix /tensorboard/viewer-a4e50b4f89b636c7e2d12bebd3da533901b3ea8a/
    - --bind_all
```

3. 然後存檔

Step6: hack tensorboard (3/3)

(此步為進階功能, 若無法實作也無仿)

1. 修改authorization policy, 改成不檢查權限 (kubectl apply -f hack/allow-all-authorization-policy.yaml)
2. 連線到Tensorboard via port-forward

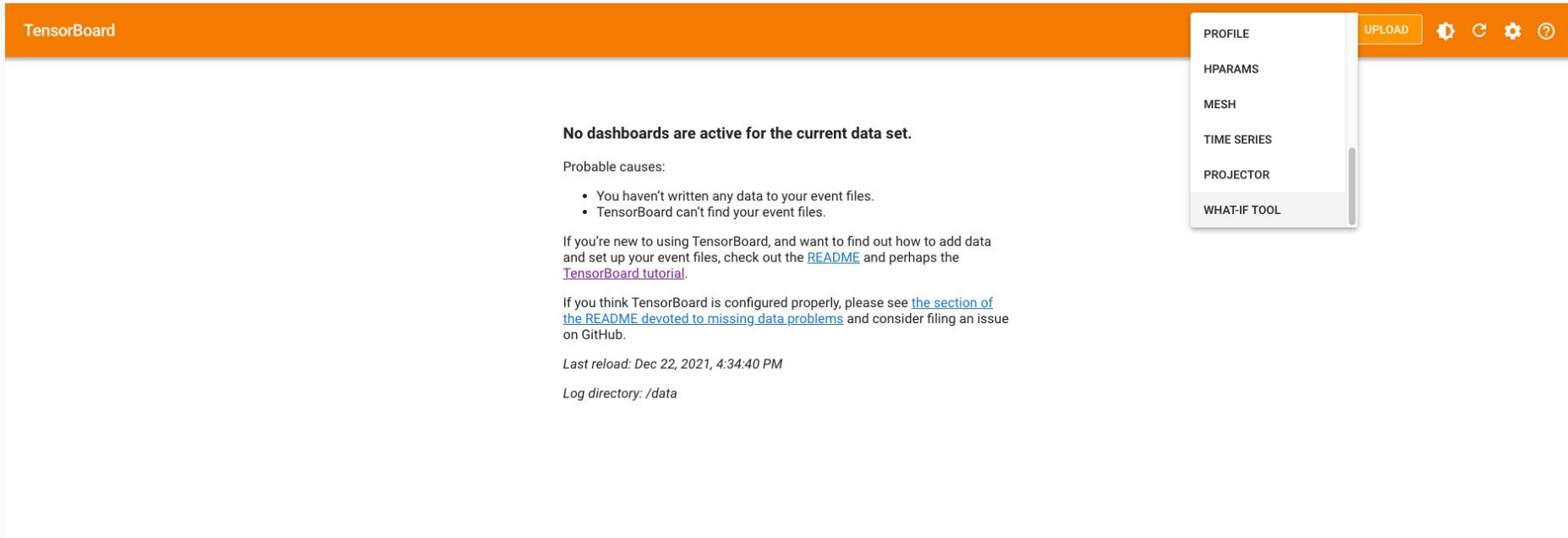
1. 取得svc名稱 (kubectl get svc -n kubeflow-user-example-com)

```
root@instance-x:/home/hsinhoyeh# kubectl get svc -n kubeflow-user-example-com
```

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
ml-pipeline-ui-artifact	ClusterIP	10.96.108.10	<none>	80/TCP	2d3h
ml-pipeline-visualizationserver	ClusterIP	10.96.42.220	<none>	8888/TCP	2d3h
test1	ClusterIP	10.96.224.201	<none>	80/TCP	29h
uci-census	ExternalName	<none>	knative-local-gateway.istio-system.svc.cluster.local	<none>	72m
uci-census-predictor-default	ExternalName	<none>	knative-local-gateway.istio-system.svc.cluster.local	80/TCP	72m
uci-census-predictor-default-00001	ClusterIP	10.96.171.220	<none>	81/TCP	72m
uci-census-predictor-default-00001-private	ClusterIP	10.96.8.198	<none>	80/TCP, 9090/TCP, 9091/TCP, 8022/TCP	72m
viewer-a4e50b4f89b636c7e2d12bebd3da533901b3ea8a-service	ClusterIP	10.96.29.148	<none>	80/TCP	7m50s

2. kubectl port-forward
svc/viewer-a4e50b4f89b636c7e2d12bebd3da533901b3ea8a-service 8091:80
--address=0.0.0.0 -n kubeflow-user-example-com
3. Open http://localhost:8091

Step7: configure what-if tool with tensorboard (1/3)



The screenshot shows the TensorBoard web interface. At the top is an orange header bar with the text "TensorBoard" on the left. On the right side of the header bar, there is a vertical menu with the following items: PROFILE, HPARAMS, MESH, TIME SERIES, PROJECTOR, and WHAT-IF TOOL. The "WHAT-IF TOOL" item is currently selected and highlighted. To the right of the menu is an "UPLOAD" button, followed by three icons: a gear (settings), a circular arrow (refresh), another gear (settings), and a question mark (help).

No dashboards are active for the current data set.

Probable causes:

- You haven't written any data to your event files.
- TensorBoard can't find your event files.

If you're new to using TensorBoard, and want to find out how to add data and set up your event files, check out the [README](#) and perhaps the [TensorBoard tutorial](#).

If you think TensorBoard is configured properly, please see [the section of the README devoted to missing data problems](#) and consider filing an issue on GitHub.

Last reload: Dec 22, 2021, 4:34:40 PM

Log directory: /data

Step7: configure what-if tool with tensorboard (2/3)

Set up your data and model

Inference address

uci-census-predictor-default.kubeflow-user-example-com.svc.cluster.local:80

Model name

uci-census

Model version (optional)

1

Model signature (optional)

ADD ANOTHER MODEL FOR COMPARISON

Model Type

☒ Classification ☐ Regression ☐ Uses Predict API

Path to examples

/data/uci_census/adult.tfrecord

☐ SequenceExamples

Maximum number of examples to load

1000

Sampling ratio (0.2 = sample ~20% of examples)

1

Path to label dictionary (optional)

☒ Maps predicted class indices to labels from text file

Max classes to display

5

☐ Multi-class classification model

Cancel

Accept

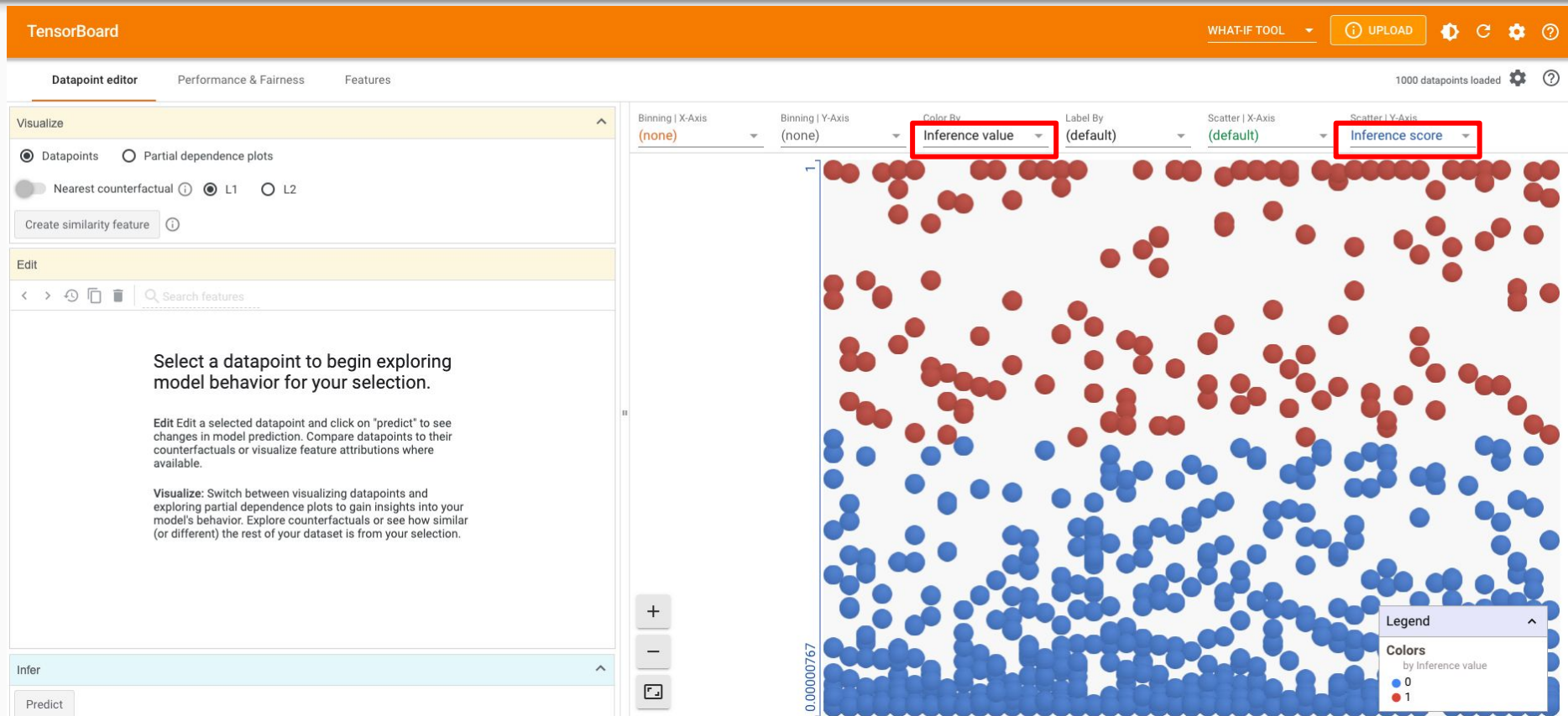
1. Our online model serving address:
uci-census-predictor-default.kubeflow-user-example-com.svc.cluster.local:80

2. Model name:
uci-census

3. Our test data used in this study:
/data/uci_census/adult.tfrecord

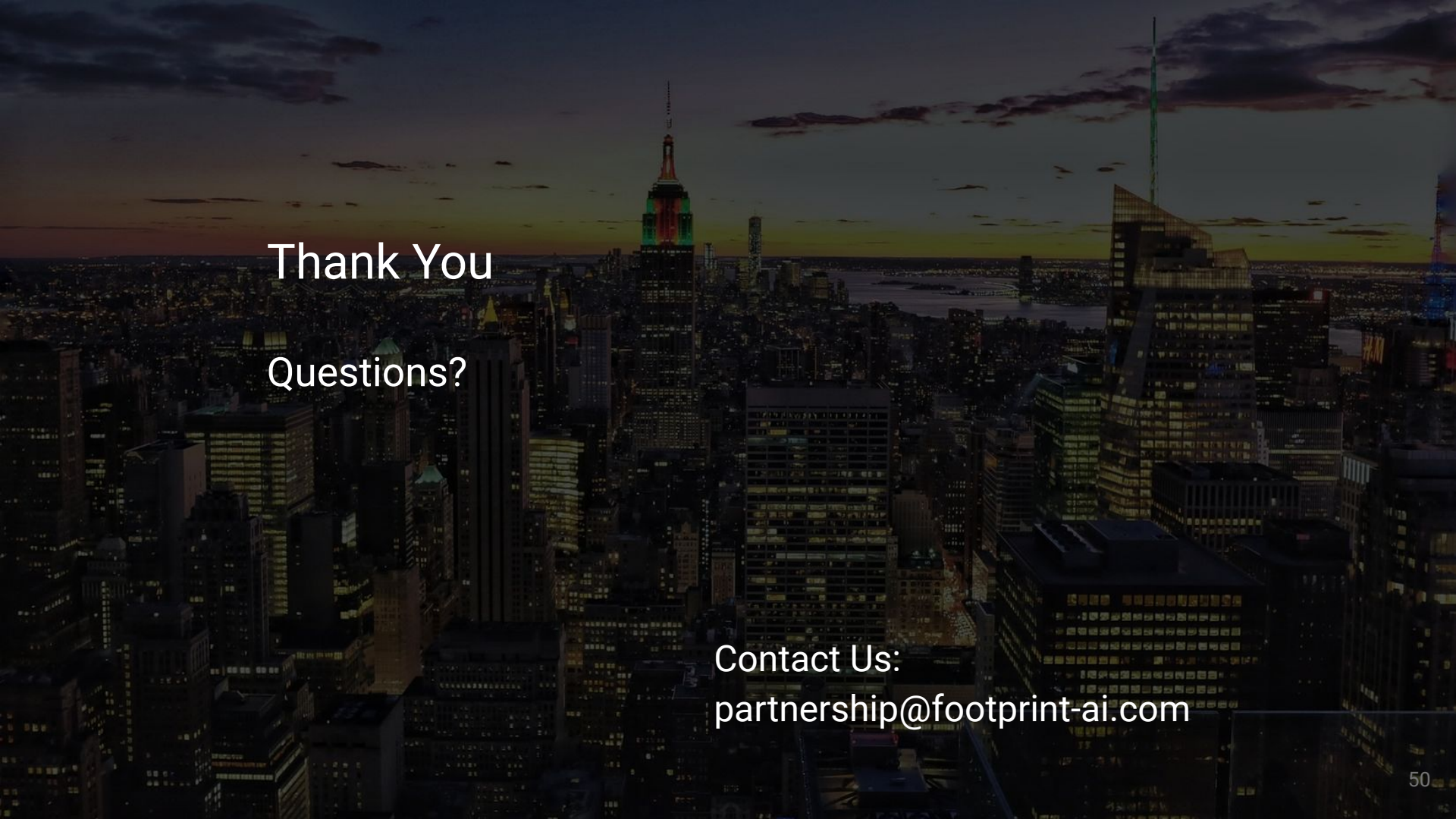
4. Number of samples used in this study: 1000

Step7: configure what-if tool with tensorboard (3/3)



What we have achieved...

- Understand fairness and explainable AI
- Build a kubeflow pipeline for
 - Kserve for serving a pre-build machine learning models
 - Tensorboard for analyzing fairness and explainable on the given data
- Interactive with What-if Tool

An aerial photograph of the New York City skyline at dusk. The sky is a mix of dark blue and orange, with scattered clouds. The city is densely packed with skyscrapers, many of which are illuminated with their interior lights. The Empire State Building is prominent in the center, with its top lit in red and green. The Hudson River is visible in the background, with the New York-New York Hotel & Casino's replica of the Empire State Building on the right side of the image.

Thank You
Questions?

Contact Us:
partnership@footprint-ai.com

- Documentations
 - <https://www.kubeflow.org/>
- Kubectl cheatsheet
 - <https://kubernetes.io/docs/reference/kubectl/cheatsheet/>