

Kubeflow Workshop

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



2023/04/20

Download Slides

<http://bit.ly/3mKUshp>



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>
 - **KaFeiDo**: machine learning platform for green economy
- 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



Kubernetes Case Study: Open AI

OpenAI adopted Kubernetes since 2016 for portability, cost saving, and improved efficiency[1,2].

Years	Nodes	Estimated Cost [3]
2018	2,500	= 3 * 2500 * 24 = US\$ 180,000 / day
2021	7,500	= 3 * 7500 * 24 = US\$ 540,000 / day
2023	?	

[1] <https://kubernetes.io/case-studies/openai/>
<https://blog.pichuang.com.tw/20230214-openai-scaling-kubernetes-to-7500-nodes.html>

[2] https://blog.pichuang.com.tw/20230214-openai-scaling-kubernetes-to-7500-nodes.html#_1

[3] AWS P3.2xlarge equips 8 vCores, 61 GB Memory, 1 Core of V100-16Gb GPU, charges US\$ 3 per hour.

What is Kubernetes?

History Of Kubernetes

- Borg: the predecessor to Kubernetes
 - Google revealed the first time of its detail in an academic research paper, describing a “cluster manager that runs hundreds of thousands of jobs, from many thousands of different applications, across a number of clusters each with up to tens of thousands of machines.”[1]
 - A in-house cluster manager system inside Google for running every google services including Gmail, Google Maps, Google Docs...[2]
 - In a scale with ‘over 2 billion containers per week’ [3]
- The very first version of Kubernetes was released in 2015
- The latest version is v1.26, released at 2023.

[1] <https://research.google/pubs/pub43438/>

[2] <https://www.wired.com/2016/04/want-build-empire-like-googles-os/>

[3] <https://cloud.redhat.com/blog/building-kubernetes-bringing-google-scale-container-orchestration-to-the-enterprise>



Kubernetes Distributions Evolution

To Cloud



To Edge



lightweight

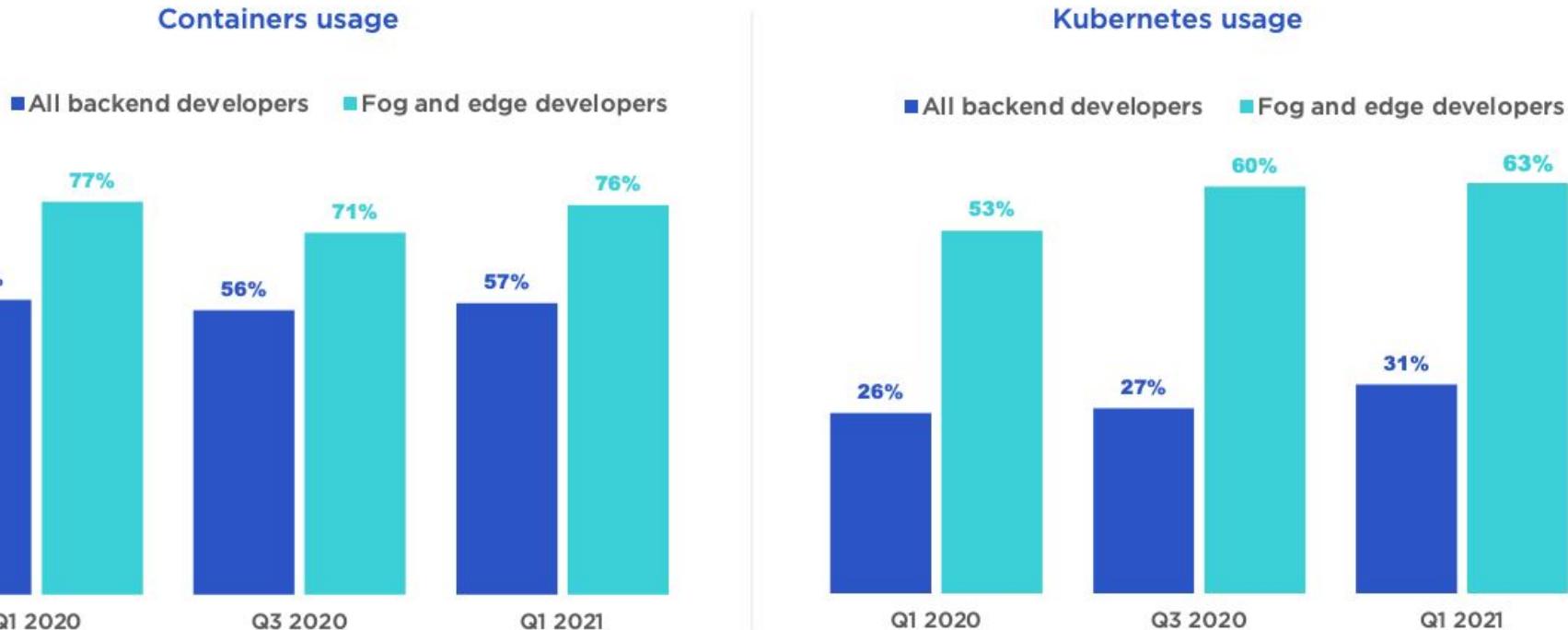


What is Kubernetes adoption rate so far?

Usage of cloud native technologies across regions



Container and Kubernetes adoption rate on edge computing



Relevant Jobs In Taiwan

Setting Popular Recent

 Senior Site Reliability Engineer
MaiCoin

Full time · Mid-Senior level
📍 110 台灣台北市信義區 ▲ 2
\$ 1.2M ~ 2.5M TWD/year

🕒 Updated 3 Months ago 500+ Unread

Save Follow

 Machine Learning Engineer (Ads) 🔥
Dcard 狀卡科技股份有限公司

Full time · Entry level
📍 台北 · 大安區
\$ 900K ~ 2.2M TWD/year

🕒 Updated 18 days ago 1,000+ Unread

Save Follow

 (Sr.) DevOps Engineer 運維開發工程師
OpenNet 開網有限公司

Full time · Mid-Senior level
📍 台灣台北 ▲ 2
\$ 800K ~ 1.8M TWD/year

🕒 Updated a month ago 300+ Unread

Save Follow

 【技術部】資深DevOps/SRE工程師
聖霏有限公司

Full time · Mid-Senior level
📍 大安區 · 台北
\$ 40K ~ 100K TWD/month

🕒 Updated 3 Months ago 50+ Unread

Save Follow

 LINE TV 數據工程師 (Data Engineer) 🔥
LINE TV_巧克技新媒體股份有限公司

Full time · Entry level
📍 台灣 · 台北
\$ 50K ~ 80K TWD/month

🕒 Updated 6 days ago 1,000+ Unread

Save Follow

 中階/資深後端工程師 Mid/Senior Backend Engineer (Python) 🔥
Linker Networks Inc. 美商實益凌科網路科技有限公司台灣分公司

Full time · Mid-Senior level
📍 Taipei · Kaohsiung ▲ 2
\$ 50K ~ 150K TWD/month

🕒 Updated 2 months ago 500+ Unread

Save Follow

 Junior DevOps/SRE
Splashtop Inc.

Full time · Entry level
📍 105 台灣台北市松山區 ▲ 2
\$ 650K ~ 1M TWD/year

🕒 Updated 4 days ago 100+ Unread

Save Follow

 Senior DevOps / SRE
Splashtop Inc.

Full time · Mid-Senior level
📍 105 台灣台北市松山區 ▲ 2
\$ 1M ~ 1.5M TWD/year

🕒 Updated 4 days ago 50+ Unread

Save Follow

 Site Reliability Engineer
OneDegree

Full time · Mid-Senior level
📍 台灣台北
\$ 840K ~ 1.5M TWD/year

🕒 Updated 4 days ago 100+ Unread

Save Follow

 Machine Learning Engineer 🔥
Dcard 狀卡科技股份有限公司

Full time · Mid-Senior level
📍 台北 · 大安區
\$ 900K ~ 2.2M TWD/year

🕒 Updated 18 days ago 1,000+ Unread

Save Follow

 Machine Learning Engineer (Search)
Dcard 狀卡科技股份有限公司

Full time · Mid-Senior level
📍 台灣台北市大安區
\$ 900K ~ 2.2M TWD/year

🕒 Updated a month ago 50+ Unread

Save Follow

 Mid-Level Software Engineer, Cymetrics (Backend)
OneDegree

Full time · Mid-Senior level
📍 台北 · 信義區
\$ 840K ~ 1.2M TWD/year

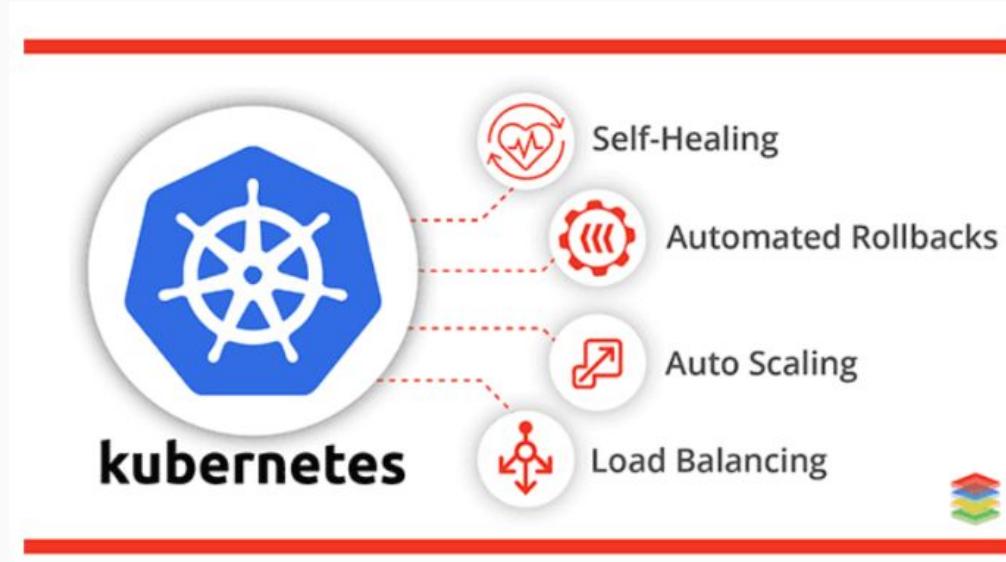
🕒 Updated 4 days ago 100+ Unread

Save Follow

 【資訊研發部】PHP 後端資深工程師 (可遠端)
聯合數位資通

Full time · Mid-Senior level
📍 台灣台北 ▲ 5
\$ 60K ~ 80K TWD/month

Kubernetes Feature Highlighted

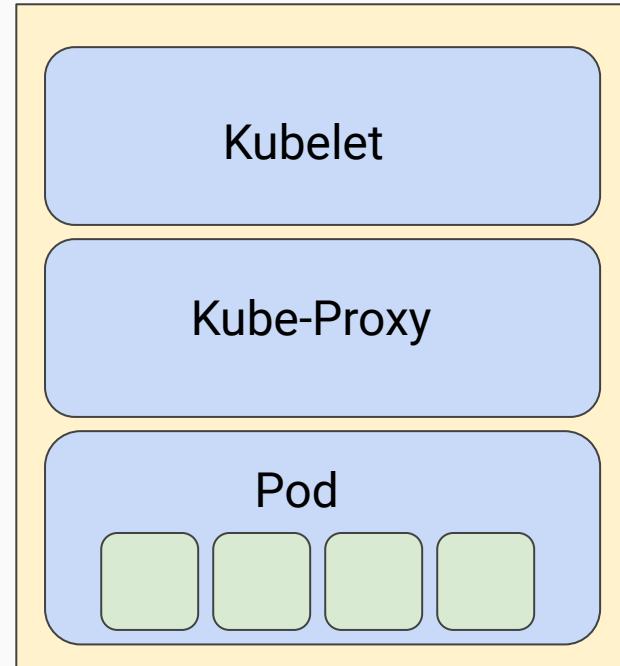


What is Kubernetes?

- High level concepts
 - **Node** are machine that run containerized applications.
 - **Pod** are unit for application workload.
 - **Scheduler** schedules pods to run on nodes.
 - **Deployment/Replica Set** ensures that a specified number of pod replicas are running at any one time.
 - **Service** is an abstract way to expose an application running on a set of Pods as a network service.

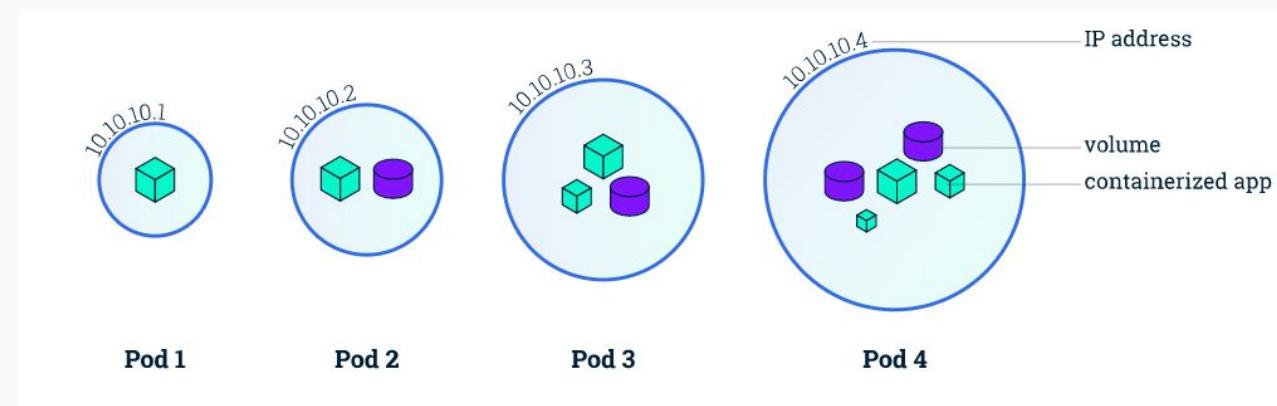
What is a Node?

- Container runtime
 - Docker / CRI-o
- Kubelet
 - Primary node agent running on each node
 - It register the node with api server and manage pods according to PodSpec.
- Kube-Proxy
 - Network proxy runs on each node. This reflects services as defined in the Kubernetes API on each node and can do simple TCP, UDP, and SCTP stream forwarding or round robin forwarding across a set of backends.



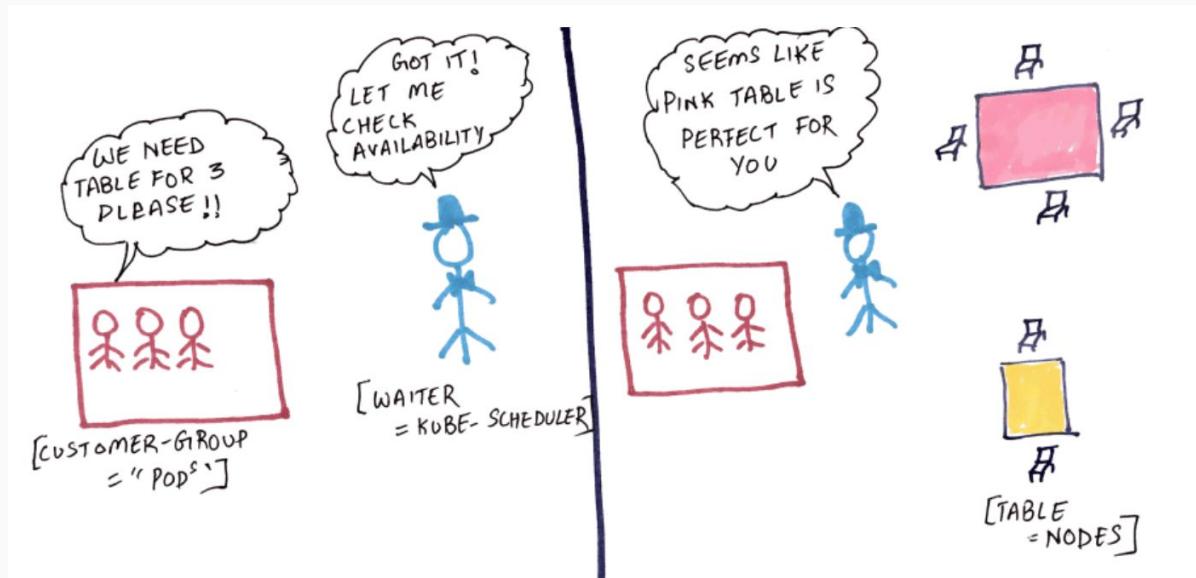
What is a Pod?

- A pod represents a logical application, it could contains a or multiple containers.
- A pod has unique IP address, persistent storage volume, and a configuration on how container should run
- Containers inside the same pod shares namespaces.
 - Containers inside the same pod can locate each other and communicate via localhost



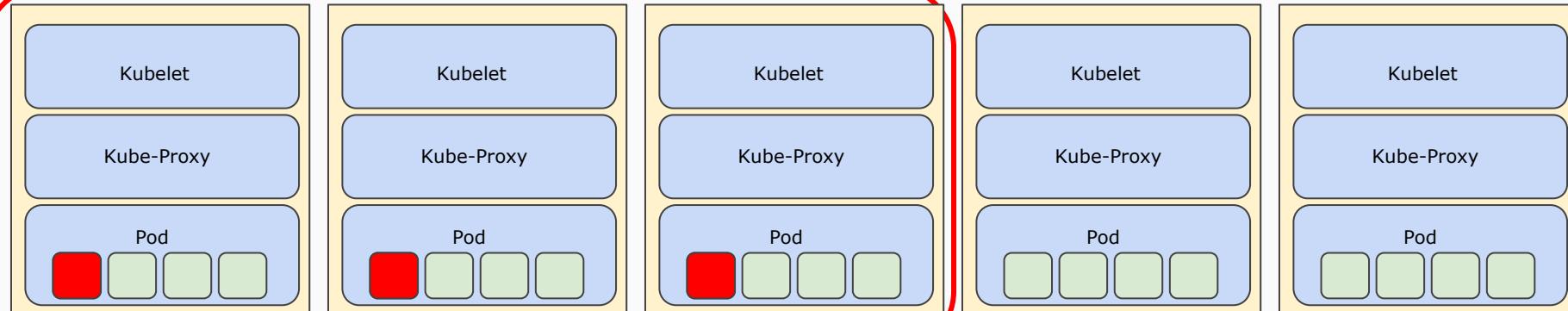
What is a Scheduler?

- The scheduler determines (filtering & scoring) which Nodes are valid placements for each Pod in the scheduling queue according to constraints and available resources.



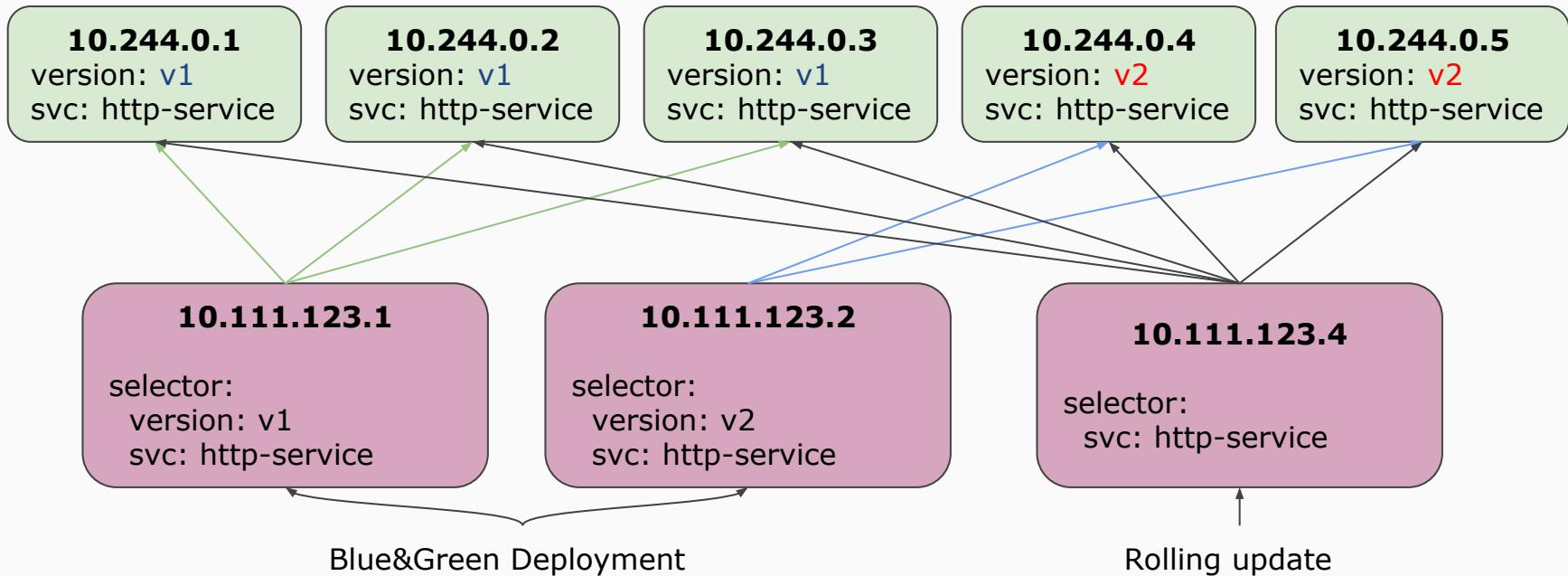
What is Replica Set?

- Manage a replicated set of pods
- Create pods from a template
- Ensure the desired number of pods running
- Online resizing and self-healing



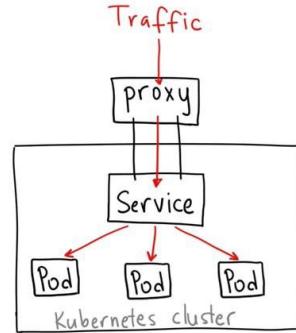
Replica =
3

What is Service?

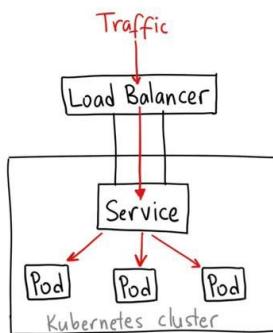


What is Service?

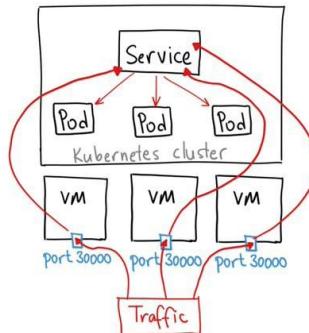
ClusterIP



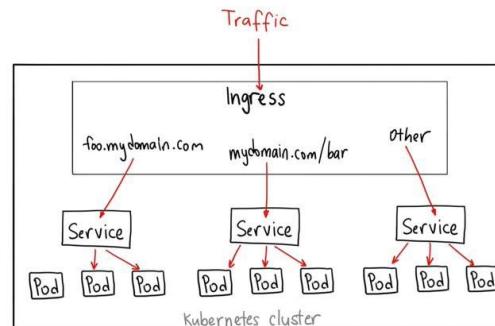
LoadBalancer



NodePort

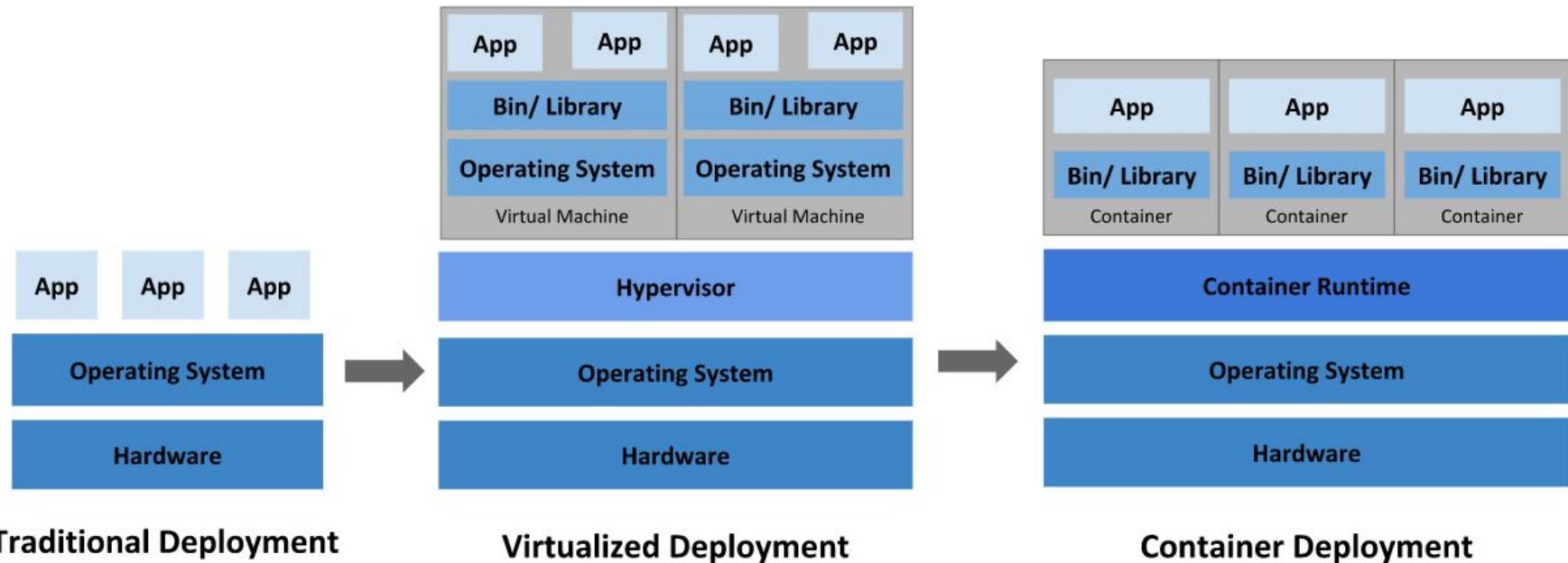


Ingress



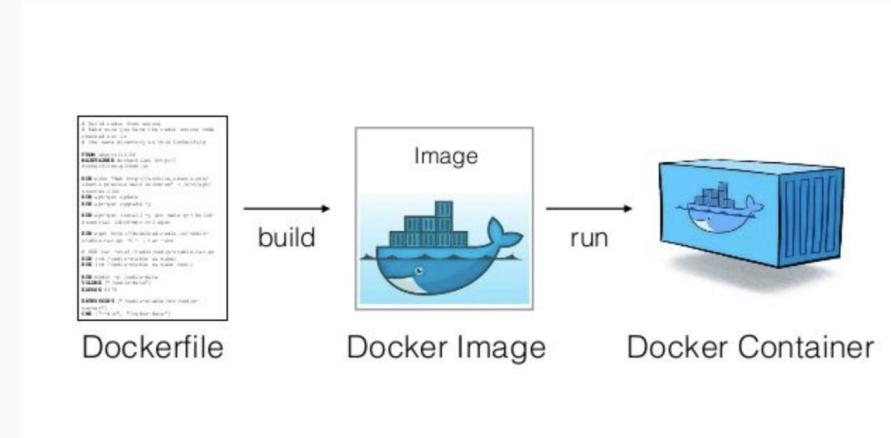
What is Container?

What is containerized deployment?



What is Container?

- Container
 - Container Image = Application code + dependencies
 - Runtime environment (cgroups, namespaces, env vars)
- Container Registry
 - Container repository



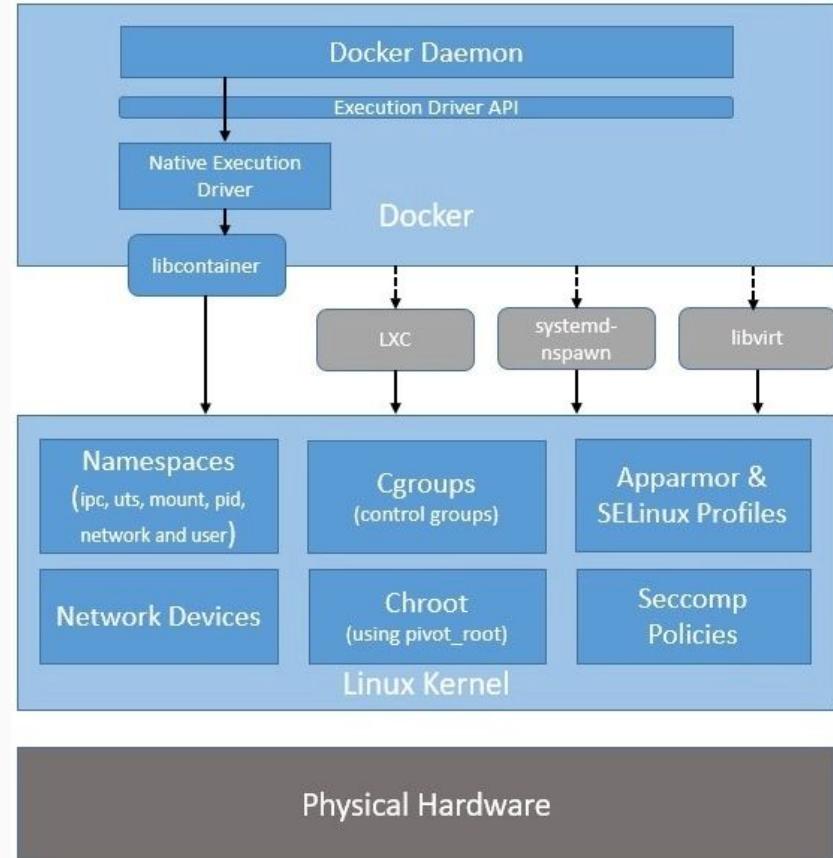
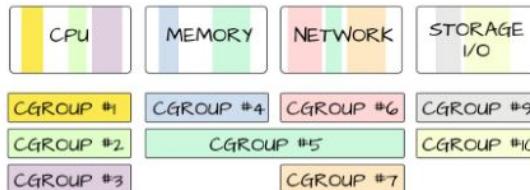
Ref: <https://medium.com/platformer-blog/practical-guide-on-writing-a-dockerfile-for-your-application-89376f88b3b5>

How container works?

- Namespace for isolation
- Cgroups for resource limiting

Cgroups : Isolation and accounting

- cpu
- memory
- block i/o
- devices
- network
- numa
- freezer



Ref: <https://www.baeldung.com/linux/docker-containers-evolution>
<https://medium.com/@BeNitinAgarwal/understanding-the-docker-internals-7ccb052ce9fe>

What is Dockerfile?

- A dockerfile contains instructions needed to build an container image

```
FROM ubuntu:18.04
```

```
RUN apt-get update && apt-get install -y build-essential
```

```
COPY . /app
```

```
RUN make /app
```

```
CMD python /app/app.py
```

How to build a Docker Image

```
FROM php:7.0-apache
```

registry

```
COPY index.php /var/www/html/index.php
```

repository

```
EXPOSE 80
```

tag

```
docker build -t footprintai/k8sworkshop:php-demo -f Dockerfile .
```

```
=> [internal] load metadata for docker.io/library/php:7.0-apache 4.6s
```

```
...
```

```
=> [2/2] COPY index.php /var/www/html/index.php
```

```
0.8s
```

```
=> exporting to image
```

```
0.2s
```

```
=> => exporting layers
```

```
0.1s
```

```
=> => writing image
```

```
sha256:e74d16d21b10069d0beba2cc6daf7cc011723d7e51523c3830e50b1bc5338e88 0.0s
```

```
=> => naming to docker.io/footprintai/k8sworkshop:php-demo 0.0s
```



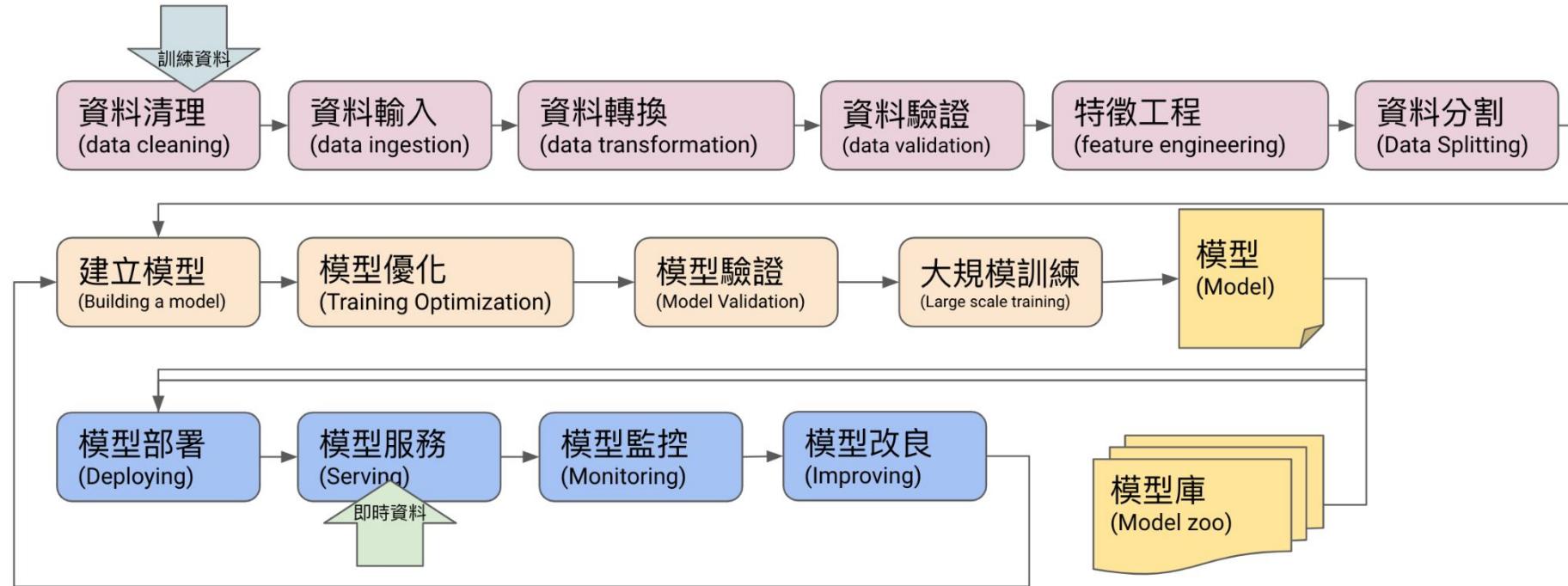
DevOps + ML
= MLOps

MLOps is the process of taking an experimental Machine Learning model into a production system by including continuous development practice of DevOps in the software field.

Ref: <https://en.wikipedia.org/wiki/MLOps>

Source: <https://www.kubeflow.org/>

Real-world Machine Learning Application - End-to-End ML LifeCycle



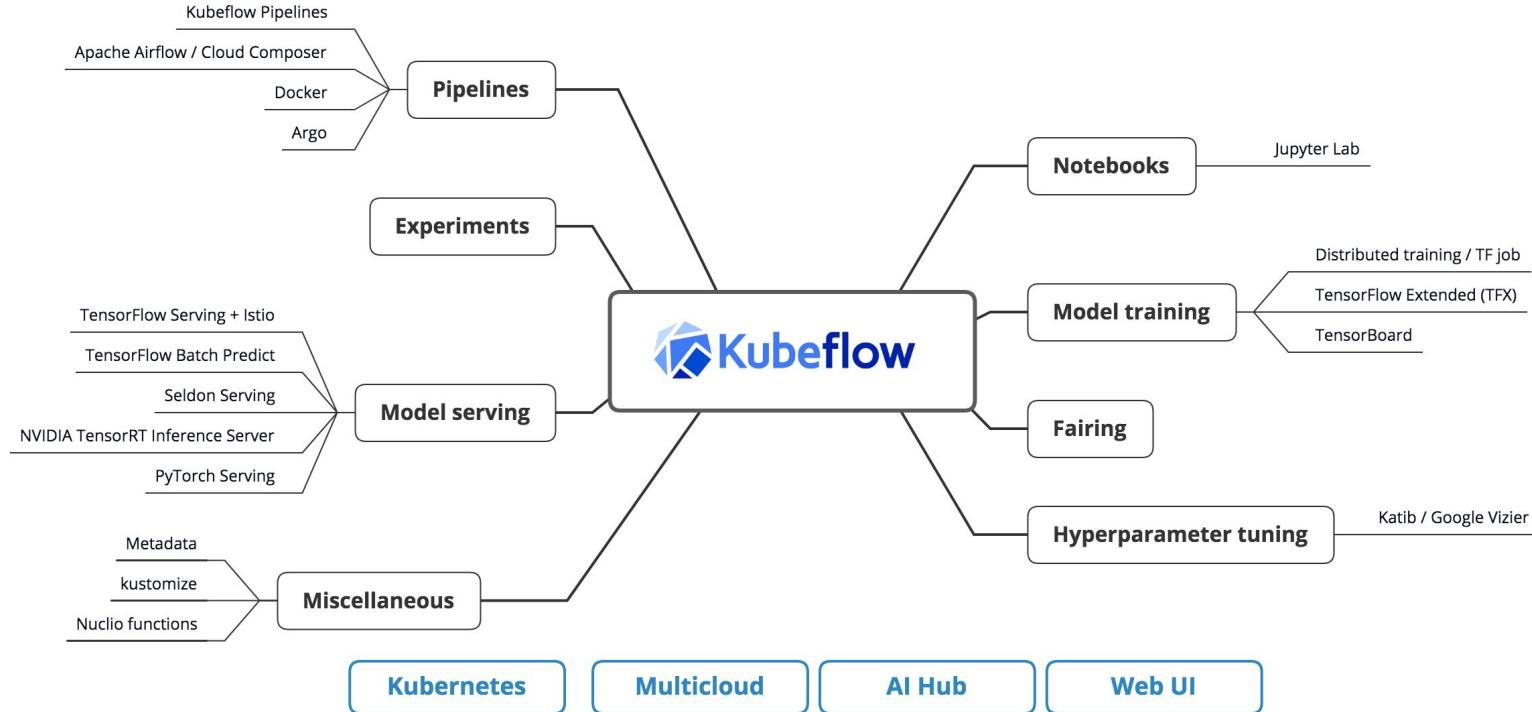
Oh, you want to use ML on K8s?

Before that, can you become an expert in:

- Containers
- Packaging
- Kubernetes service endpoints
- Persistent volumes
- Scaling
- Immutable deployments
- GPUs, Drivers & the GPL
- Cloud APIs
- DevOps
- ...



Architectures



Hands on

Pre-requirement

- 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 export with `Google`
 - <https://letmegooglet.com/?q=you+can+google+it>
- It is totally OK if you don't know what is Container and Kubernetes

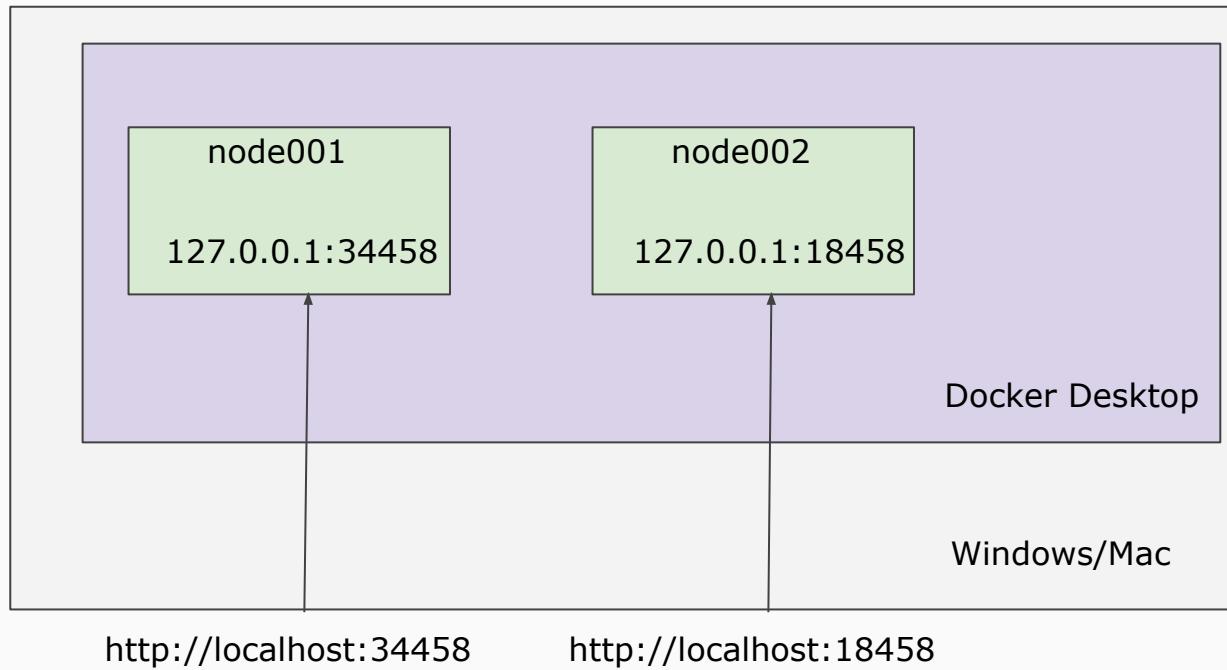
孩子，您多久沒唸中文了？

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。」

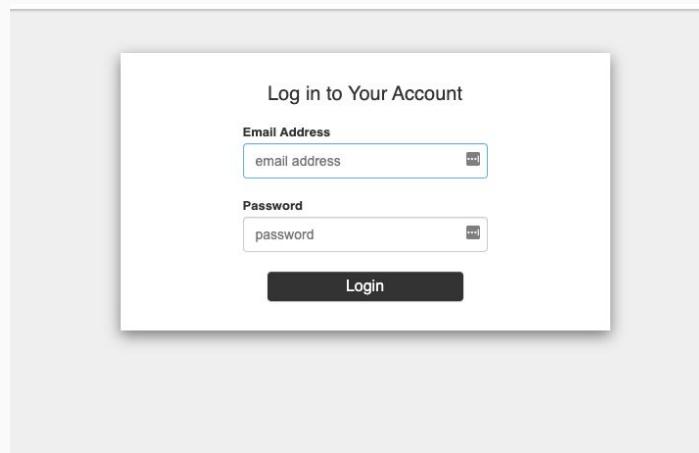
multikf: One-click Installation

- Multikf: <https://github.com/footprintai/multikf>



Wait! 所以我說那個帳號密碼呢?

Account: user@example.com
Password: 12341234



Step1: Use Notebook as Online IDE (1/3)

The screenshot shows the Kubeflow web interface. On the left, a sidebar menu is visible with the following items:

- Home
- Notebooks** (highlighted with a red box)
- Tensorboards
- Volumes
- Models
- Experiments (AutoML)
- Experiments (KFP)
- Pipelines
- Runs
- Recurring Runs
- Artifacts
- Executions

At the bottom of the sidebar, there is a link to "Privacy • Usage Reporting" and a note about the build version: "build version dev_local".

The main content area is titled "Notebooks" and displays a table with columns: Status, Name, Type, Age, Image, GPUs, CPUs, Memory, and Volumes. There is no data in the table.

In the top right corner of the main area, there is a button labeled "+ NEW NOTEBOOK" with a red box around it. A large red arrow points upwards towards this button from the bottom right.

Step1: Use Notebook as Online IDE (2/3)

The screenshot shows the Kubeflow interface for creating a new Notebook Server. On the left is a sidebar with various options: Home, Notebooks (highlighted), Tensorboards, Volumes, Models, Experiments (AutoML), Experiments (KFP), Pipelines, Runs, Recurring Runs, Artifacts, and Executions. Below these are Manage Contributors and Privacy & Help links.

The main area is titled "Specify the name of the Notebook Server and the Namespace it will belong to." It contains two input fields: "Name" (containing "demo") and "Namespace" (containing "kubeflow-user-example-com").

Below this is a section for the "Image". It includes a "Custom Image" checkbox (unchecked), a dropdown menu showing "jupyterlab" selected (with options 1 and 2), and a detailed view of the "jupyterlab" entry. This view shows the image URL "j1r0q0g6/notebooks/notebook-servers/jupyter-tensorflow-full:v1.4".

Under "Advanced Options", there is a "CPU / RAM" section. It asks to "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)". Two input fields are shown: "Requested CPU" (containing "0.5") and "Requested Memory (in GiB)" (containing "1").

A large red box highlights the "Name" field. Two red arrows point from the text "Specify a name and resources like CPU and Memory" to the "Requested CPU" and "Requested Memory" fields respectively.

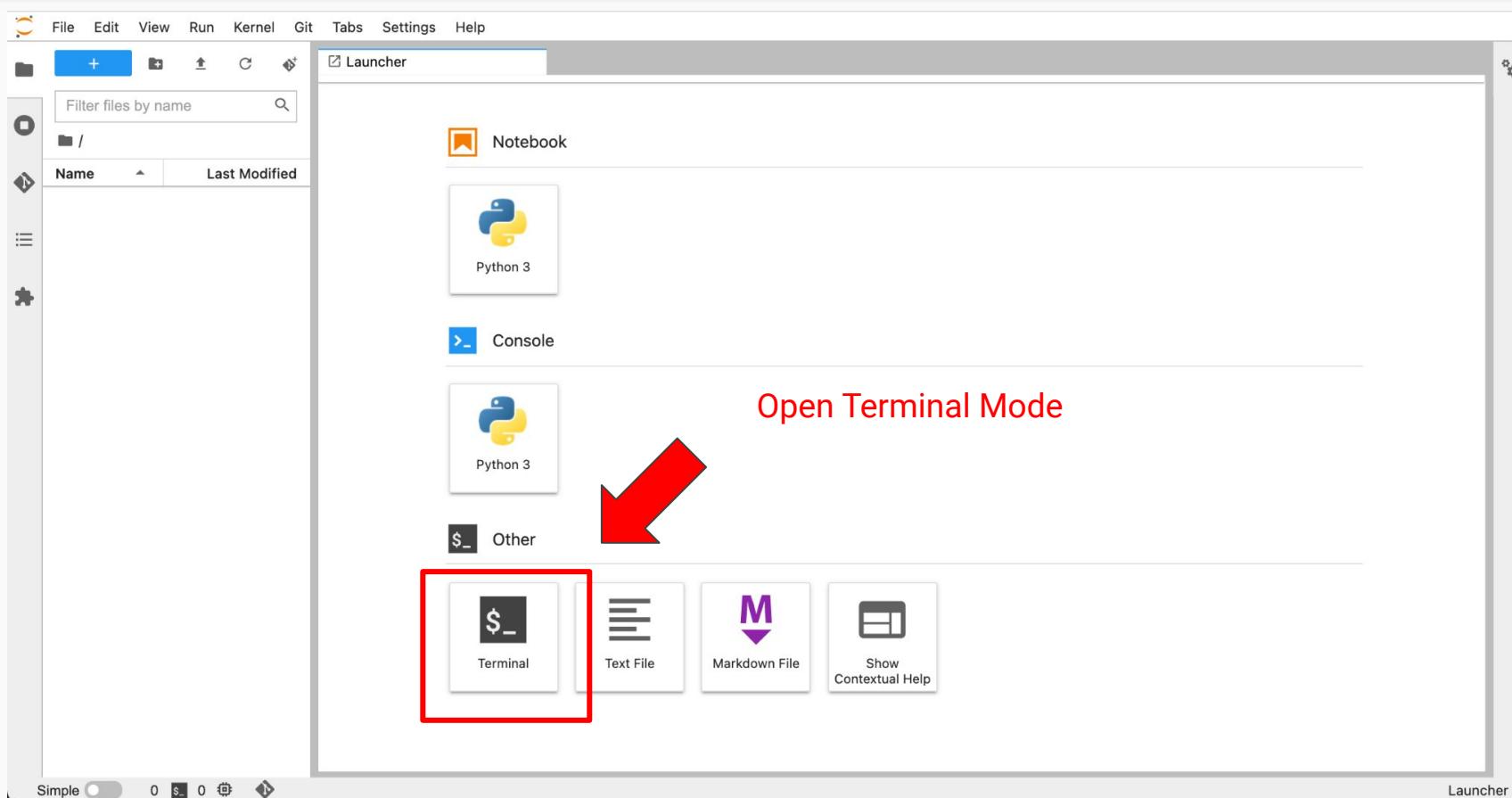
Specify a name and resources like CPU and Memory

Step1: Use Notebook as Online IDE (3/3)

The screenshot shows the Kubeflow interface with the 'Notebooks' tab selected in the sidebar. The main area displays a table of notebooks with columns for Status, Name, Type, Age, Image, GPUs, CPUs, Memory, and Volumes. Two notebooks are listed: 'demo1' and 'demo2'. The 'demo2' row is highlighted with a red box around its 'CONNECT' button, which is also highlighted with a red arrow pointing upwards.

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

Step2: Use terminal to download the materials (1/3)



Step2: Use terminal to download the materials (2/3)

The screenshot shows a Jupyter Notebook interface. On the left is a file browser sidebar. In the center is a terminal window titled "Terminal 1" containing a command-line session. A red box highlights the command "git clone https://github.com/footprintai/kubeflow-workshop". A large red arrow points upwards from a green box containing the same command towards the terminal window. The green box is positioned below the terminal window.

```
groups: cannot find name for group ID 1222
(base) jovyan@dem01-0:~$ 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@dem01-0:~$
```

git clone https://github.com/footprintai/kubeflow-workshop

Simple 1 0 Terminal 1 38

Step2: Use terminal to download the materials (3/3)

The screenshot shows a Jupyter Notebook interface. On the left is a file browser pane with a sidebar containing icons for file operations like new file, copy, move, and delete. A search bar at the top of the browser allows filtering by name. Below the search bar is a list of files under the path `/kubeflow-workshop / pipelines /`. The list includes:

- img (directory)
- 0.helloworld.ipynb
- 1.conditional-flow.ipynb
- 2.persistvolume.ipynb
- 3.calc_metrics.ipynb
- 4.mnist.ipynb
- 5.auto-mnist-with-katib.ipynb
- 6.kfserving.ipynb
- 7.kfserving-canary-rollout.ipynb
- kfp.ipynb
- testdata.jpg
- testdata2.jpg
- testdata3.jpg

A red box highlights the first seven items in the list. To the right of the file browser is a terminal window titled "Terminal 1". It displays the output of a `git clone` command:

```
groups: cannot find name for group ID 1337
(base) jovyan@demo1-0:~$ 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@demo1-0:~$
```

At the bottom of the terminal window, there is a status bar with the text "Terminal 1".

Kubeflow Terms

詞彙說明



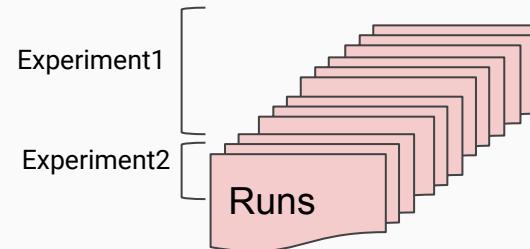
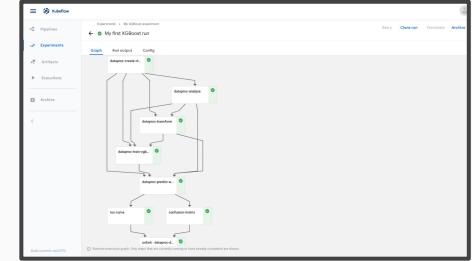
```
1.1 with open('requirements.txt', 'w') as f:  
    f.write("pyyaml<4.0.0  
    pip install --upgrade --user  
  
1.2 import sys  
1.3 import os  
1.4  
1.5 def echo(text):  
1.6     return os.GlobbingPattern()  
1.7     image_id="library/bash:4.4.23"  
1.8     command="sh -c \"echo %s\" %s" % (text, text)  
1.9     arguments=(text, "text")  
  
1.10 def echo_parallel(*args):  
1.11     image_id="library/bash:4.4.23"  
1.12     command="sh -c \"parallel bash -c \"echo {}\" {}\" -r" % (args[0], args[1])  
1.13     arguments=(args[0], "args[1]")  
  
1.14 def pipeline():  
1.15     """  
1.16         # pipeline code to demonstrate execution order management.  
1.17         # This pipeline will demonstrate how to explicitly define execution order.  
1.18         # step1_task < step2_task < step3_task < step4_task  
1.19         # step1_task < step2_task < step3_task  
1.20         # step1_task < step2_task < step4_task  
1.21     """  
1.22     step1_task = echo_parallel("step1", "1")  
1.23     step2_task = echo_parallel("step2", "2")  
1.24     step3_task = echo_parallel("step3", "3")  
1.25     step4_task = echo_parallel("step4", "4")  
  
1.26     # generate workflow artifacts in .zip format  
1.27     My_pipeline=compile_parallel_compile_parallel_order_pipeline("yellowworld.zip")
```

Pipeline Code

Compiled

Workflow Resource

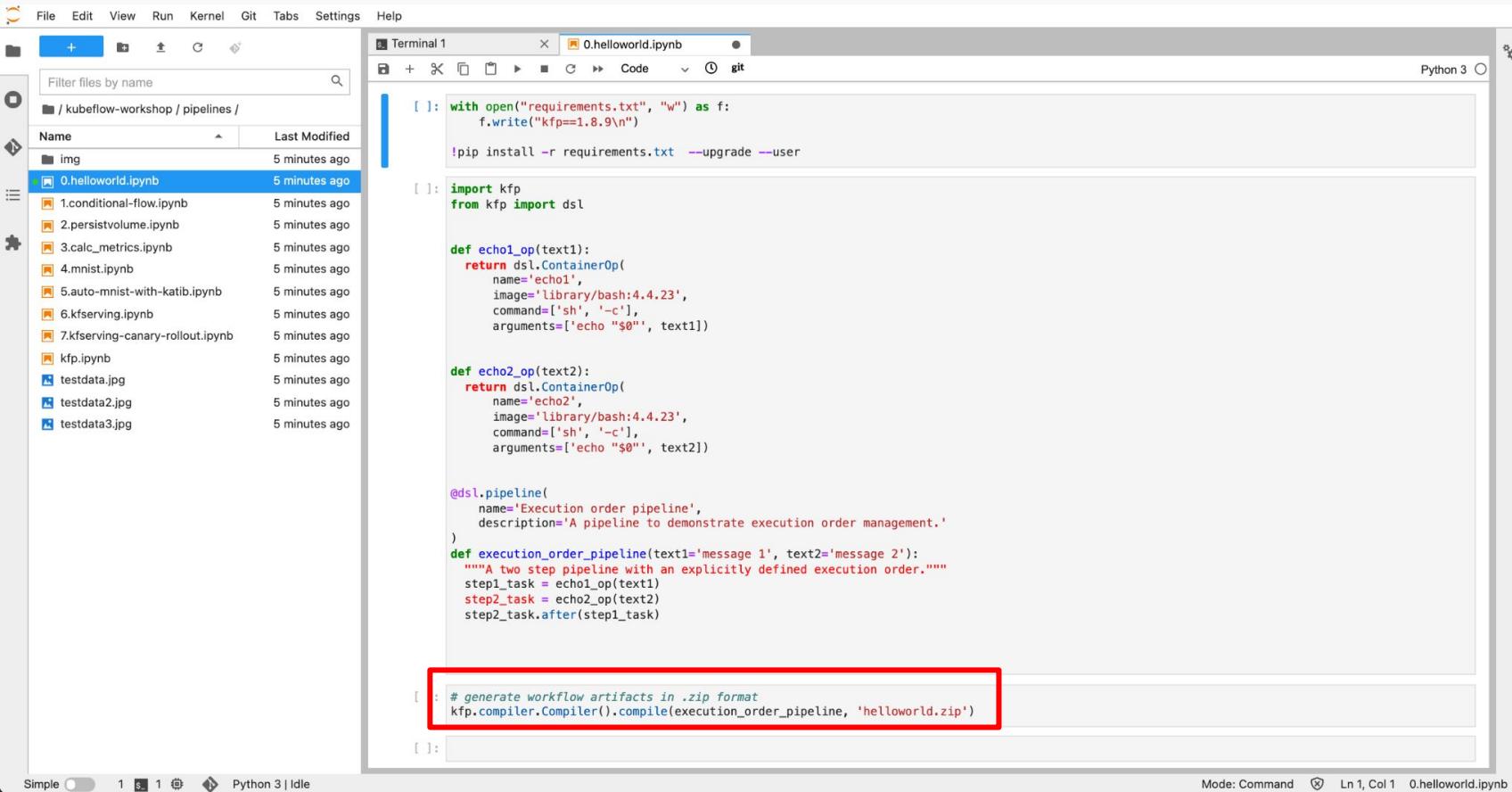
Create a Pipeline



Create Run

Hello World Example

Step4: Compile helloworld.ipynb (1/2)



The screenshot shows a Jupyter Notebook interface with the following details:

- File Bar:** File, Edit, View, Run, Kernel, Git, Tabs, Settings, Help.
- Left Sidebar:** Shows a file tree under the path `/kubeflow-workshop / pipelines /`. The current file selected is `0.helloworld.ipynb`.
- Terminal 1:** A terminal window titled `0.helloworld.ipynb` running Python 3. It displays the following code:

```
[ ]: 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')
```


```



The last line of code, kfp.compiler.Compiler().compile(execution_order_pipeline, 'helloworld.zip'), is highlighted with a red box.



Bottom Status Bar: Simple, Python 3 | Idle, Mode: Command, Ln 1, Col 1, 0.helloworld.ipynb, 43.


```

Step4: Compile helloworld.ipynb (2/2)



File Edit View Run Kernel Git Tabs Settings Help

Terminal 1 × 0.helloworld.ipynb Python 3

Filter files by name

/ kubeflow-workshop / pipelines /

Name	Last Modified
img	7 minutes ago
0.helloworld.ipynb	seconds ago
1.conditional-flow.ipynb	7 minutes ago
2.persistvolume.ipynb	7 minutes ago
3.calc_metrics.ipynb	7 minutes ago
4.mnist.ipynb	7 minutes ago
5.auto-mnist-with-katib.ipynb	7 minutes ago
6.kfserving.ipynb	7 minutes ago
7.kfserving_end-to-end.ipynb	7 minutes ago
helloworld.zip	seconds ago
kfp.ipynb	7 minutes ago
requirements.txt	seconds ago
testdata.jpg	7 minutes ago
testdata2.jpg	7 minutes ago
testdata3.jpg	7 minutes ago

```
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)

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

/opt/conda/lib/python3.8/site-packages/kfp/dsl/_container_op.py:1150: FutureWarning: Please create reusable components instead of constructing ContainerOp instances directly. Reusable components are shareable, portable and have compatibility and support guarantees. Please see the documentation on: https://www.kubeflow.org/docs/pipelines/sdk/component-development/#writing-your-component-definition-file The components can be created manually (or, in case of python, using kfp.components.create_component_from_func or func_to_container_op) and then loaded using kfp.components.load_component_from_file, load_component_from_uri or load_component_from_text: https://kubeflow-pipelines.readthedocs.io/en/stable/source/kfp.components.html#kfp.components.load\_component\_from\_file
warnings.warn(
```

generated helloworld.zip

Step4: Create a Pipeline (1/7)

The screenshot shows the Kubeflow Pipelines interface. On the left, a sidebar menu is visible with various options: Home, Notebooks, Tensorboards, Volumes, Models, Experiments (AutoML), Experiments (KFP), Pipelines (highlighted with a red box), Runs, Recurring Runs, Artifacts, and Executions. Below this is a 'Manage Contributors' section. The main area is titled 'Pipelines' and contains a table of existing pipelines. The table has columns for Pipeline name, Description, and Last modified. A red box highlights the 'Upload pipeline' button in the top right corner of the header. A large red arrow points upwards towards this button.

Pipeline name	Description	Last modified
[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

Step4: Create a Pipeline (2/7)

The screenshot shows the Kubeflow interface for creating a new pipeline. The left sidebar lists various options like Home, Notebooks, Tensorboards, Volumes, Models, Experiments (AutoML), Experiments (KFP), Pipelines, Runs, Recurring Runs, Artifacts, and Executions. The main area is titled 'Upload Pipeline or Pipeline Version' with a sub-section for 'Pipeline Versions'. It shows two radio button options: 'Create a new pipeline' (selected) and 'Create a new pipeline version under an existing pipeline'. A large red box highlights the 'Pipeline Name' field, which contains '0.helloworld'. Another red box highlights the 'File*' input field, which contains 'helloworld (19).zip'. A third red box highlights the 'Create' button at the bottom left. Red arrows point from the right side to each of these three highlighted areas, labeled '1.Pipeline Name', '2.specify zip file location', and '3.Create' respectively.

Kubeflow

kubeflow-user-example-c... ▾

← 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 helloworld (19).zip Choose file

Import by url Package Url

Code Source (optional)

Create Cancel

1.Pipeline Name

2.specify zip file location

3.Create

Step4: Create a Pipeline (3/7)

The screenshot shows the Kubeflow Pipelines interface. On the left, a sidebar menu includes Home, Notebooks, Tensorboards, Volumes, Models, Experiments (AutoML), Experiments (KFP), Pipelines, Runs, Recurring Runs, Artifacts, and Executions. Below this is a section for Manage Contributors.

The main area displays a pipeline named "0.helloworld (0.helloworld)". It shows a graph with two nodes: "echo1" at the top and "echo2" below it. An arrow points from "echo1" to "echo2". Above the graph, there are tabs for "Graph" (selected) and "YAML". To the right of the graph are buttons for "+ Create run", "+ Upload version", and "+ Create experiment". A large red box highlights the "+ Create experiment" button, and a red arrow points upwards towards it from the bottom right. A red callout text "Create an experiment" is positioned to the right of the arrow.

Summary [Hide](#)

ID	Version
6f25028f-01e3-4acd-9389-7ec2031fb04b	0.helloworld

Version source

Step4: Create a Pipeline (4/7)

The screenshot shows the Kubeflow interface for creating a new experiment. On the left is a dark sidebar with navigation links: Home, Notebooks, Tensorboards, Volumes, Models, Experiments (AutoML), Experiments (KFP) (which is expanded to show Pipelines, Runs, Recurring Runs, Artifacts, and Executions), Manage Contributors, and GitHub integration.

The main area is titled "Experiments" and "← New experiment". It has a sub-section titled "Experiment details" with a descriptive text: "Think of an Experiment as a space that contains the history of all pipelines and their associated runs".

A form is displayed for entering experiment details:

- Name:** A text input field containing "0.helloworld.exp" is highlighted with a red border.
- Description (optional):** An empty text input field.
- Buttons:** A row of buttons at the bottom left of the form, with "Next" highlighted with a red border and a large red arrow pointing to it from the right.

Text overlay: The text "Enter experiment name and press Next button." is overlaid in red in the center-right area of the form.

Step4: Create a Pipeline (5/7)

The screenshot shows the Kubeflow UI for creating a pipeline run. The left sidebar contains navigation links: Home, Notebooks, Tensorboards, Volumes, Models, Experiments (AutoML), Experiments (KFP) (selected), Pipelines, Runs, Recurring Runs, Artifacts, Executions, Manage Contributors, GitHub, Documentation, Privacy + Usage Reporting, and build version dev_local.

The main area is titled "Run details". It includes fields for "Pipeline", "Pipeline Version", "Run name", "Description (optional)", "Experiment", "Service Account (Optional)", "Run Type" (set to "One-off"), and "Run parameters". The "Run parameters" section contains fields for "text1" and "text2". At the bottom are "Start" and "skip this step" buttons.

Two red arrows point from the right towards the "Pipeline" and "Experiment" fields, which are highlighted with red boxes. Red text annotations provide instructions:

1. Run a pipeline and specify its version
2. Add its experiment name

Step4: Create a Pipeline (6/7)

Kubeflow

kubeflow-user-example-c... ▾

Experiments

← 0.helloworld.exp

Recurring run configs 0 active

Experiment description

Manage

Refresh Archive

Runs

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

Active Archived

Filter runs

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

Rows per page: 10 < >

Run List

Step4: Create a Pipeline (7/7)

The screenshot shows the Kubeflow interface for creating a pipeline. On the left, the sidebar includes options like Home, Notebooks, Tensorboards, Volumes, Models, Experiments (AutoML), Experiments (KFP), Pipelines, Runs, Recurring Runs, Artifacts, and Executions. The main area displays a pipeline graph titled "Run of 0.helloworld (1e261)". The graph consists of two nodes: "echo1" and "echo2", connected by a downward arrow. A red box highlights this graph. To the right, a modal window titled "execution-order-pipeline-fxxfn-4223123588" shows the "Input/Output" tab. It lists "Input parameters" (text1: message 1) and "Input artifacts". Under "Output parameters" and "Output artifacts", it shows "main-logs" with the value "minio://mipeline/artifacts/execution-order-pipeline-qvlt5/2021/11/30/execution-order-pipeline-qvlt5-137025724/main.log" and "message 1". A large red arrow points upwards from the "Output artifacts" section towards the text "運行結果輸出" (Execution Result Output). The top right of the screen has buttons for Retry, Clone run, Terminate, and Archive.

Graph

Run output

Config

Simplify Graph

echo1

echo2

execution-order-pipeline-fxxfn-4223123588

Input/Output

Input parameters

text1 message 1

Input artifacts

Output parameters

Output artifacts

main-logs

minio://mipeline/artifacts/execution-order-pipeline-qvlt5/2021/11/30/execution-order-pipeline-qvlt5-137025724/main.log

message 1

View All

運行結果輸出

Hyperparameter Example

Step5: Hyperparameter tuning with katib (1/4)

File Edit View Run Kernel Git Tabs Settings Help

5.auto-mnist-with-katib.ipynb

```
[1]: with open("requirements.txt", "w") as f:  
    f.write("kfp==1.8.9\n")  
    f.write("kubeflow-katib==0.12.0\n")  
  
!pip install -r requirements.txt --upgrade --user  
  
Requirement already satisfied: kfp==1.8.9 in /home/jovyan/.local/lib/python3.8/site-packages (from -r requirements.txt (line 1)) (1.8.9)  
Collecting kubeflow-katib==0.12.0  
    Downloading kubeflow_katib-0.12.0-py3-none-any.whl (89 kB)  
     ██████████ | 89 kB 5.1 MB/s eta 0:00:011  
Requirement already satisfied: typing-extensions<4,>=3.7.4 in /opt/conda/lib/python3.8/site-packages (from kfp==1.8.9->-r requirements.txt (line 1)) (3.7.4.3)  
Requirement already satisfied: tabulate<1,>=0.8.6 in /opt/conda/lib/python3.8/site-packages (from kfp==1.8.9->-r requirements.txt (line 1)) (0.8.9)  
Requirement already satisfied: protobuf<4,>=3.13.0 in /opt/conda/lib/python3.8/site-packages (from kfp==1.8.9->-r requirements.txt (line 1)) (3.17.3)  
Requirement already satisfied: requests-toolbelt<1,>=0.8.0 in /opt/conda/lib/python3.8/site-packages (from kfp==1.8.9->-r requirements.txt (line 1)) (0.9.1)  
Requirement already satisfied: PyYAML<6,>=5.3 in /opt/conda/lib/python3.8/site-packages (from kfp==1.8.9->-r requirements.txt (line 1)) (5.4.1)  
Requirement already satisfied: strip-hints<1,>=0.1.8 in /opt/conda/lib/python3.8/site-packages (from kfp==1.8.9->-r requirements.txt (line 1)) (0.1.10)  
Requirement already satisfied: docstring-parser<1,>=0.7.3 in /opt/conda/lib/python3.8/site-packages (from kfp==1.8.9->-r requirements.txt (line 1)) (0.11)  
Requirement already satisfied: cloudpickle<3,>=2.0.0 in /home/jovyan/.local/lib/python3.8/site-packages (from kfp==1.8.9->-r requirements.txt (line 1)) (2.0.0)  
Requirement already satisfied: jsonschema<4,>=3.0.1 in /opt/conda/lib/python3.8/site-packages (from kfp==1.8.9->-r requirements.txt (line 1)) (3.2.0)  
Requirement already satisfied: absl-py<=0.11,>=0.9 in /opt/conda/lib/python3.8/site-packages (from kfp==1.8.9->-r requirements.txt (line 1)) (0.11.0)  
Requirement already satisfied: google-auth<2,>=1.6.1 in /opt/conda/lib/python3.8/site-packages (from kfp==1.8.9->-r requirements.txt (line 1)) (1.31.0)
```

Step5: Hyperparameter tuning with katib (2/4)

The screenshot shows the Kubeflow interface for managing experiments. On the left, a sidebar lists various resources: Home, Notebooks, Tensorboards, Volumes, Models, Experiments (AutoML), Experiments (KFP), Pipelines, Runs, Recurring Runs, and Artifacts. The main area is titled 'Experiments' and shows the details for the experiment 'hello1'. It includes sections for 'Recurring run configs' (0 active) and 'Experiment description'. Below this is a 'Runs' section with tabs for 'Active' (selected) and 'Archived'. A search bar labeled 'Filter runs' is present. A table lists three runs, with the first run highlighted by a red box. The columns in the table are: Run name, Status, Duration, Pipeline Version, Recurri..., Start time, quotient, and remainder. The first run's status is 'Running' (indicated by a question mark icon), duration is '-' (likely pending), pipeline version is 'hello-world_vers...', start time is '5/17/2022, 9:42:...', and it has a quotient of '0.000' and a remainder of '6.000'.

Run name	Status	Duration	Pipeline Version	Recurri...	Start time	quotient	remainder
Run of hello-world_versio...	?	-	hello-world_vers...	-	5/17/2022, 9:42:...	0.000	6.000
Run of hello-world_versio...	✓	0:06:19	hello-world_vers...	-	5/17/2022, 9:30:...		
Run of hello-world_versio...	✓	0:01:24	hello-world_vers...	-	5/17/2022, 9:23:...		

Step5: Hyperparameter tuning with katib (3/4)

The screenshot shows the Kubeflow interface. On the left, a sidebar menu is visible with the following items:

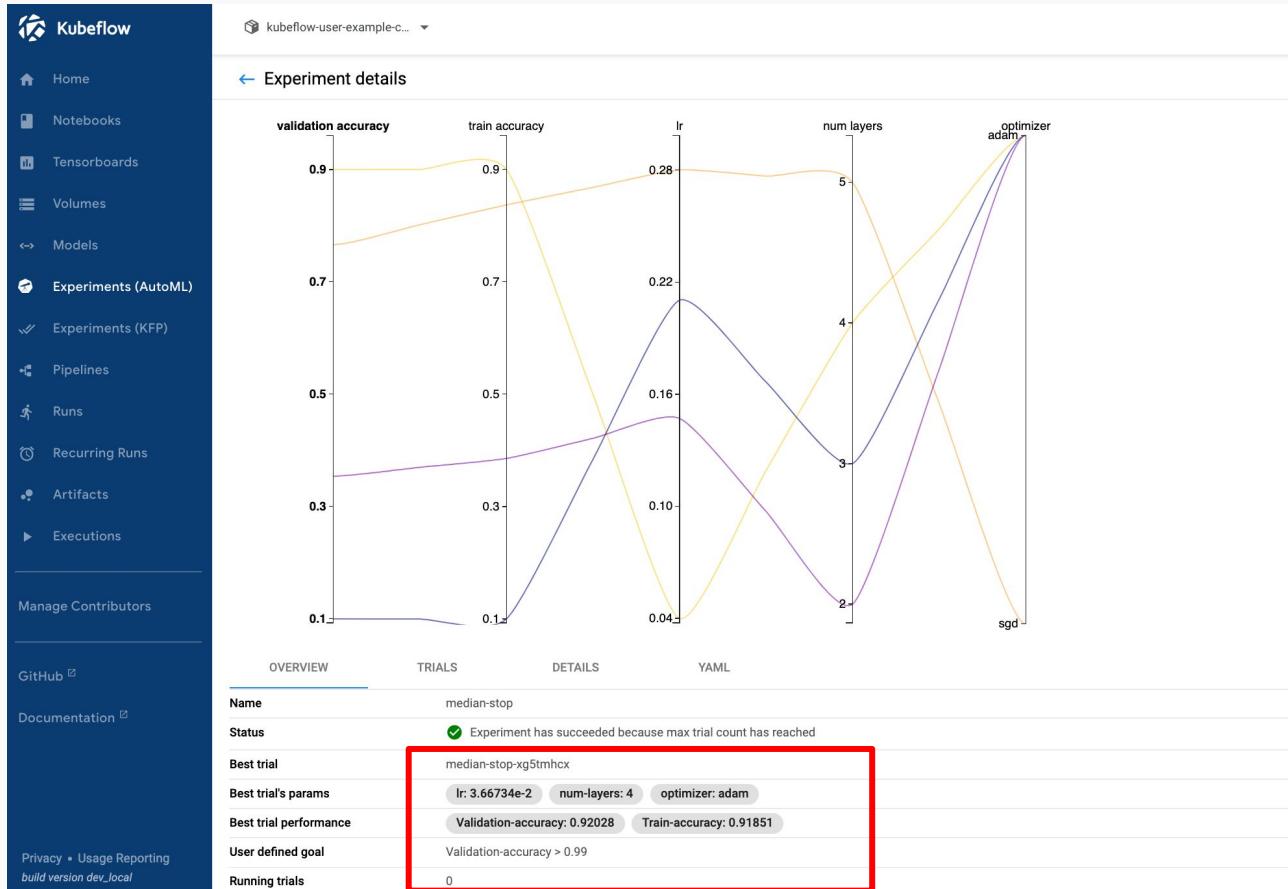
- Home
- Notebooks
- Tensorboards
- Volumes
- Models
 - Experiments (AutoML) **(highlighted by a red box)**
 - Experiments (KFP)
- Pipelines
- Runs
- Recurring Runs
- Artifacts

The main content area is titled "Experiments". It displays a table with the following columns: Status, Name, Age, Successful trials, Running trials, Failed trials, and Optimal trial. One experiment is listed:

Status	Name	Age	Successful trials	Running trials	Failed trials	Optimal trial
✓	median-stop	8 minutes ago	4	0	0	Validation accuracy: 0.92028

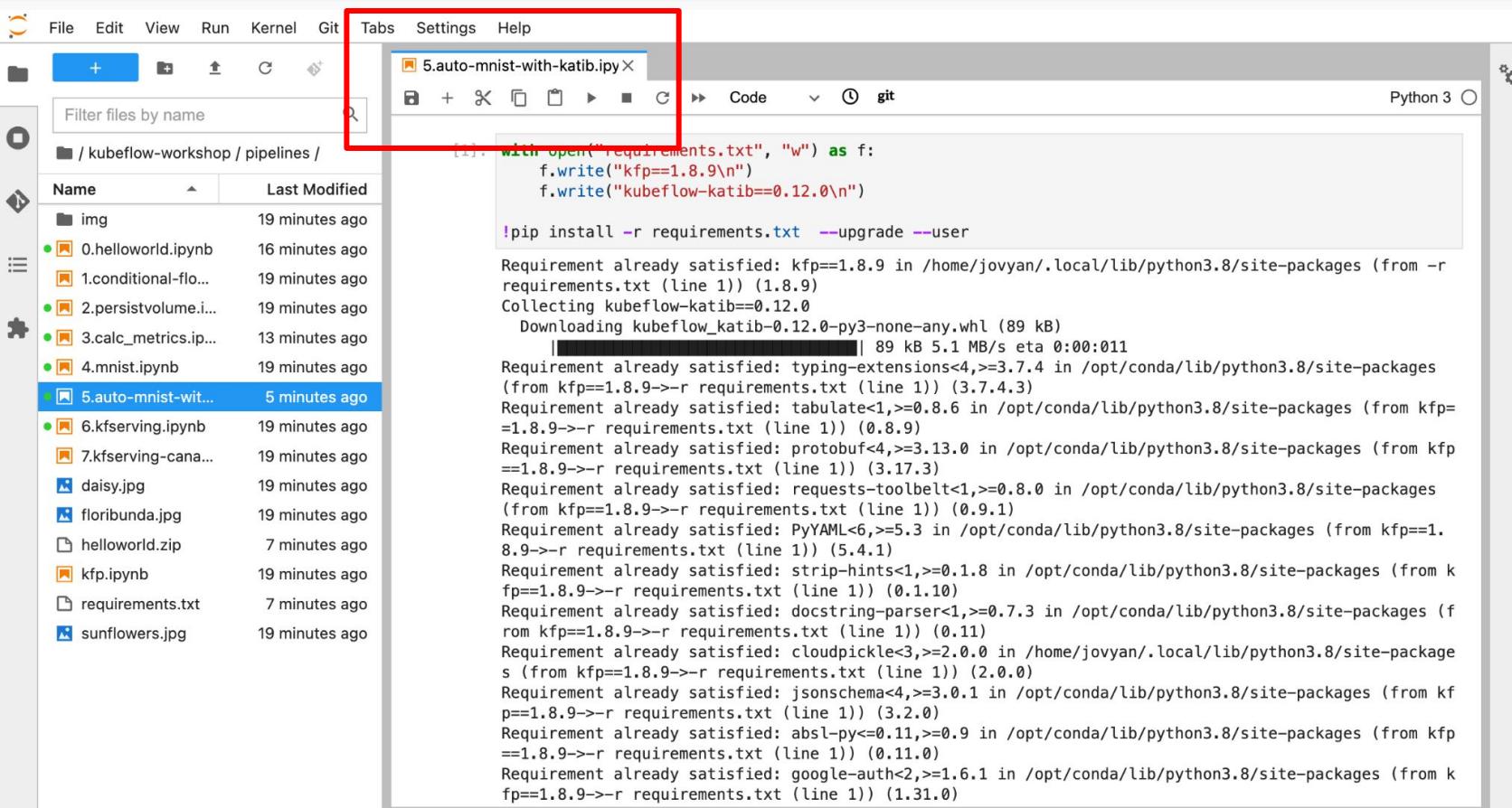
A blue "+ NEW EXPERIMENT" button is located in the top right corner of the experiments section.

Step5: Hyperparameter tuning with kubeflow (4/4)



Kserve Example

Step6: Serving model with Kserve (1/4)



File Edit View Run Kernel Git Tabs Settings Help

5.auto-mnist-with-katib.ipynb

```
[1]: with open("requirements.txt", "w") as f:  
    f.write("kfp==1.8.9\n")  
    f.write("kubeflow-katib==0.12.0\n")  
  
!pip install -r requirements.txt --upgrade --user  
  
Requirement already satisfied: kfp==1.8.9 in /home/jovyan/.local/lib/python3.8/site-packages (from -r requirements.txt (line 1)) (1.8.9)  
Collecting kubeflow-katib==0.12.0  
  Downloading kubeflow_katib-0.12.0-py3-none-any.whl (89 kB)  
 |██████████| 89 kB 5.1 MB/s eta 0:00:011  
Requirement already satisfied: typing-extensions<4,>=3.7.4 in /opt/conda/lib/python3.8/site-packages (from kfp==1.8.9->-r requirements.txt (line 1)) (3.7.4.3)  
Requirement already satisfied: tabulate<1,>=0.8.6 in /opt/conda/lib/python3.8/site-packages (from kfp==1.8.9->-r requirements.txt (line 1)) (0.8.9)  
Requirement already satisfied: protobuf<4,>=3.13.0 in /opt/conda/lib/python3.8/site-packages (from kfp==1.8.9->-r requirements.txt (line 1)) (3.17.3)  
Requirement already satisfied: requests-toolbelt<1,>=0.8.0 in /opt/conda/lib/python3.8/site-packages (from kfp==1.8.9->-r requirements.txt (line 1)) (0.9.1)  
Requirement already satisfied: PyYAML<6,>=5.3 in /opt/conda/lib/python3.8/site-packages (from kfp==1.8.9->-r requirements.txt (line 1)) (5.4.1)  
Requirement already satisfied: strip-hints<1,>=0.1.8 in /opt/conda/lib/python3.8/site-packages (from kfp==1.8.9->-r requirements.txt (line 1)) (0.1.10)  
Requirement already satisfied: docstring-parser<1,>=0.7.3 in /opt/conda/lib/python3.8/site-packages (from kfp==1.8.9->-r requirements.txt (line 1)) (0.11)  
Requirement already satisfied: cloudpickle<3,>=2.0.0 in /home/jovyan/.local/lib/python3.8/site-packages (from kfp==1.8.9->-r requirements.txt (line 1)) (2.0.0)  
Requirement already satisfied: jsonschema<4,>=3.0.1 in /opt/conda/lib/python3.8/site-packages (from kfp==1.8.9->-r requirements.txt (line 1)) (3.2.0)  
Requirement already satisfied: absl-py<=0.11,>=0.9 in /opt/conda/lib/python3.8/site-packages (from kfp==1.8.9->-r requirements.txt (line 1)) (0.11.0)  
Requirement already satisfied: google-auth<2,>=1.6.1 in /opt/conda/lib/python3.8/site-packages (from kfp==1.8.9->-r requirements.txt (line 1)) (1.31.0)
```

Filter files by name

- / kubeflow-workshop / pipelines /
- Name Last Modified
- img 19 minutes ago
- 0.helloworld.ipynb 16 minutes ago
- 1.conditional-flo... 19 minutes ago
- 2.persistvolume.i... 19 minutes ago
- 3.calc_metrics.ip... 13 minutes ago
- 4.mnist.ipynb 19 minutes ago
- 5.auto-mnist-wit... 5 minutes ago
- 6.kfserving.ipynb 19 minutes ago
- 7.kfserving-cana... 19 minutes ago
- daisy.jpg 19 minutes ago
- floribunda.jpg 19 minutes ago
- helloworld.zip 7 minutes ago
- kfp.ipynb 19 minutes ago
- requirements.txt 7 minutes ago
- sunflowers.jpg 19 minutes ago

Step6: Serving model with Kserve (2/4)

The screenshot shows the Kubeflow interface for managing machine learning experiments. On the left, a sidebar lists various resources: Home, Notebooks, Tensorboards, Volumes, Models, Experiments (AutoML), Experiments (KFP), Pipelines, Runs, Recurring Runs, and Artifacts. The 'Runs' section is currently selected.

The main area displays the 'Experiments' page for 'hello1'. It includes sections for 'Recurring run configs' (0 active) and 'Experiment description'. Below this is the 'Runs' section, which has tabs for 'Active' (selected) and 'Archived'. There are buttons for '+ Create run' and '+ Create recurring run'. The 'Runs' table lists three entries:

Run name	Status	Duration	Pipeline Version	Recurri...	Start time	quotient	remainder
Run of hello-world_version_1	?	-	hello-world_version_1	-	5/17/2022, 9:42:...		
Run of hello-world_version_1	✓	0:06:19	hello-world_version_1	-	5/17/2022, 9:30:...		
Run of hello-world_version_1	✓	0:01:24	hello-world_version_1	-	5/17/2022, 9:23:...	0.000	6.000

A red box highlights the first row in the table, indicating the successful deployment of the model version.

Step6: Serving model with Kserve (3/4)

The screenshot shows the Kubeflow interface. On the left, a dark sidebar lists various components: Home, Notebooks, Tensorboards, Volumes, Models (which is highlighted with a red box), Experiments (AutoML), Experiments (KFP), Pipelines, Runs, Recurring Runs, and Artifacts. The main area is titled "Model Servers" and contains a table with the following data:

Status	Name	Age	Predictor	Runtime	Protocol	Storage URI
✓	tensorflow-sample	2 minutes ago	Tensorflow	1.14.0		gs://kfserving-samples/models/tensorflow/flo...

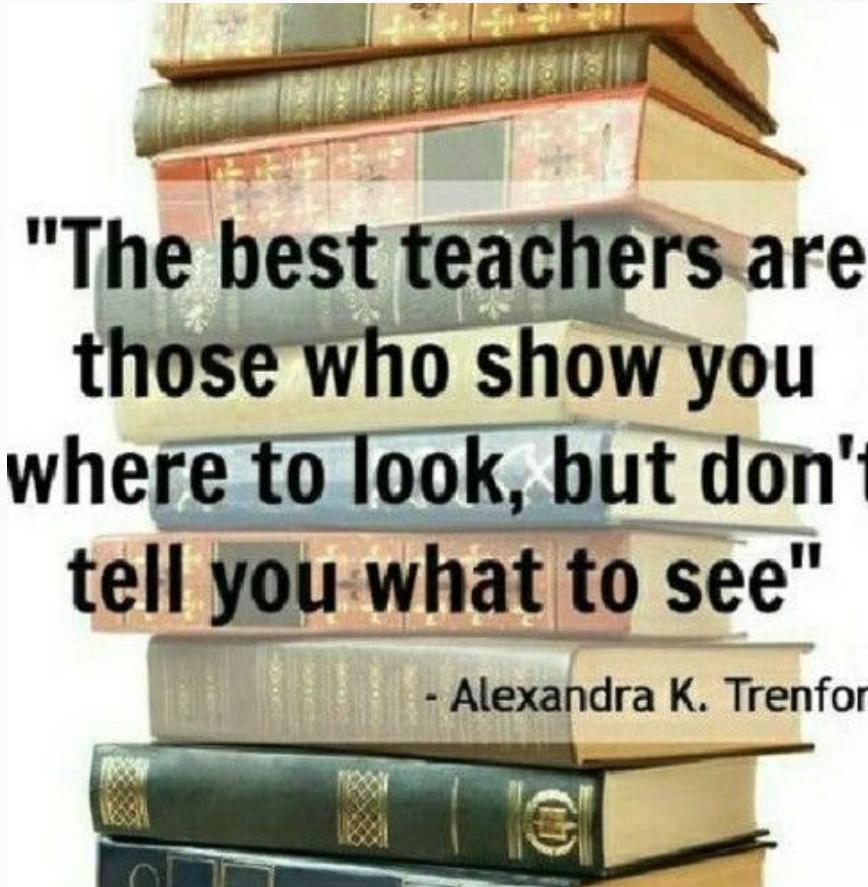
A red box highlights both the "Models" item in the sidebar and the "tensorflow-sample" row in the table. To the right of the table, there are "NEW MODEL SERVER" and "Edit" buttons.

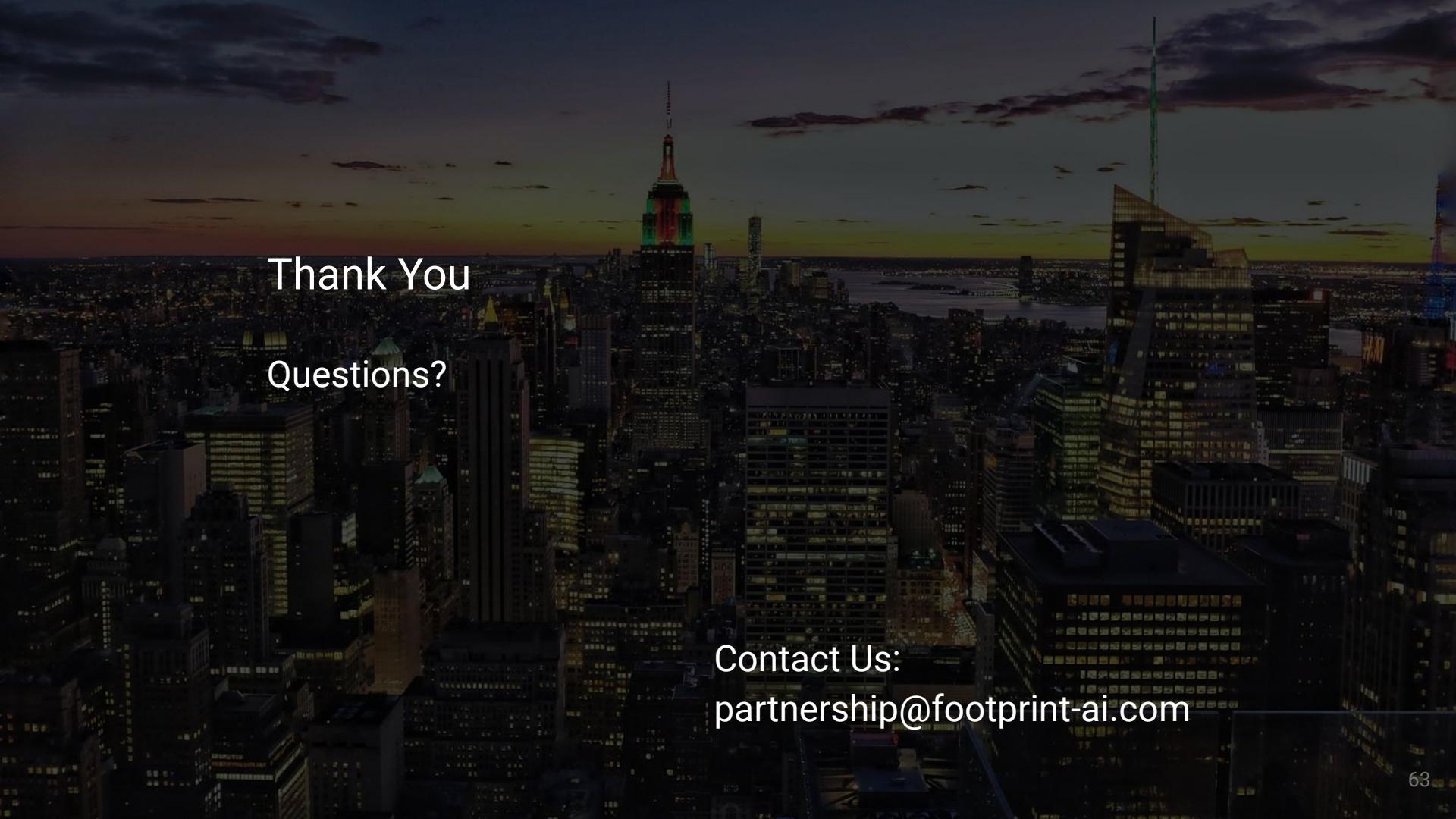
Step6: Serving model with Kserve (4/4)

The screenshot shows a Jupyter Notebook interface with the following details:

- File Bar:** File, Edit, View, Run, Kernel, Git, Tabs, Settings, Help.
- File Explorer:** Shows a directory structure under "/kubeflow-workshop / pipelines /".
- Code Cell:** The active cell (cell 4) contains Python code for sending a POST request to an Istio Ingress Gateway using the requests library. A red box highlights this code. The code reads a file named "floribunda.jpg", encodes its content to base64, and sends a POST request to "http://istio-ingressgateway.istio-system/v1/models/tensorflow-sample:predict" with the "authservice_session" cookie set to a specific value. The response is printed to the console.
- Output Cell:** The output of the highlighted code is shown in the bottom cell (cell 1), which uses curl to send a REST API request to the same endpoint.
- Network Tab:** A network tab in the browser's developer tools shows the "authservice_session" cookie being sent with the request.
- Python 3:** The kernel is set to Python 3.

Quote: The best teachers are those who show you where to look, but don't tell you what to see.



The background of the slide is a photograph of a city skyline at night, likely New York City, with the Empire State Building prominently visible. The sky is a mix of dark blues and warm orange and yellow hues from the setting sun.

Thank You

Questions?

Contact Us:
partnership@footprint-ai.com