



# Kubeflow

## NetApp Solutions

NetApp  
May 10, 2024

# 目录

Kubeflow .....	1
Kubeflow 部署 .....	1
Kubeflow 操作和任务示例 .....	2

# Kubeflow

## Kubeflow 部署

本节介绍在 Kubernetes 集群中部署 Kubeflow 必须完成的任务。

### 前提条件

在执行本节所述的部署练习之前，我们假定您已执行以下任务：

1. 您已有一个正在运行的Kubernetes集群、并且正在运行的Kubeflow版本支持此Kubernetes版本。有关支持的Kubernetes版本的列表、请参阅中您的Kubeflow版本的依赖关系 "[Kubeflow 官方文档](#)"。
2. 您已在Kubernetes集群中安装并配置NetApp Astra三端存储。有关Astra三项功能的更多详细信息、请参见 "[Astra Trident 文档](#)"。

### 设置默认 Kubernetes StorageClass

在部署Kubeflow之前、我们建议在Kubernetes集群中指定一个默认StorageClass。Kubeflow部署过程可能会尝试使用默认StorageClass配置新的永久性卷。如果未将任何StorageClass指定为默认StorageClass、则部署可能会失败。要在集群中指定默认 StorageClass ，请从部署跳转主机执行以下任务。如果已在集群中指定默认 StorageClass ，则可以跳过此步骤。

1. 将现有 StorageClasses 之一指定为默认 StorageClass 。以下示例命令显示了名为的StorageClass的命名 `ontap-ai-flexvols-retain` 作为默认StorageClass。



`ontap-nas-flexgroup` Trident 后端类型的最小 PVC 大小相当大。默认情况下， KubeFlow 会尝试配置大小只有少数几 GB 的 PVC 。因此，在部署 Kubeflow 时，不应将利用 `ontap-nas-flexgroup` 后端类型的 StorageClass 指定为默认 StorageClass 。

```
$ kubectl get sc
NAME                                PROVISIONER                AGE
ontap-ai-flexgroups-retain          csi.trident.netapp.io     25h
ontap-ai-flexgroups-retain-iface1   csi.trident.netapp.io     25h
ontap-ai-flexgroups-retain-iface2   csi.trident.netapp.io     25h
ontap-ai-flexvols-retain            csi.trident.netapp.io     3s
$ kubectl patch storageclass ontap-ai-flexvols-retain -p '{"metadata": {"annotations":{"storageclass.kubernetes.io/is-default-class":"true"}}}'
storageclass.storage.k8s.io/ontap-ai-flexvols-retain patched
$ kubectl get sc
NAME                                PROVISIONER                AGE
ontap-ai-flexgroups-retain          csi.trident.netapp.io     25h
ontap-ai-flexgroups-retain-iface1   csi.trident.netapp.io     25h
ontap-ai-flexgroups-retain-iface2   csi.trident.netapp.io     25h
ontap-ai-flexvols-retain (default)  csi.trident.netapp.io     54s
```

## Kubeflow部署选项

部署Kubeflow有许多不同的选项。请参见 "[Kubeflow 官方文档](#)" 有关部署选项的列表、请选择最适合您的需求的选项。

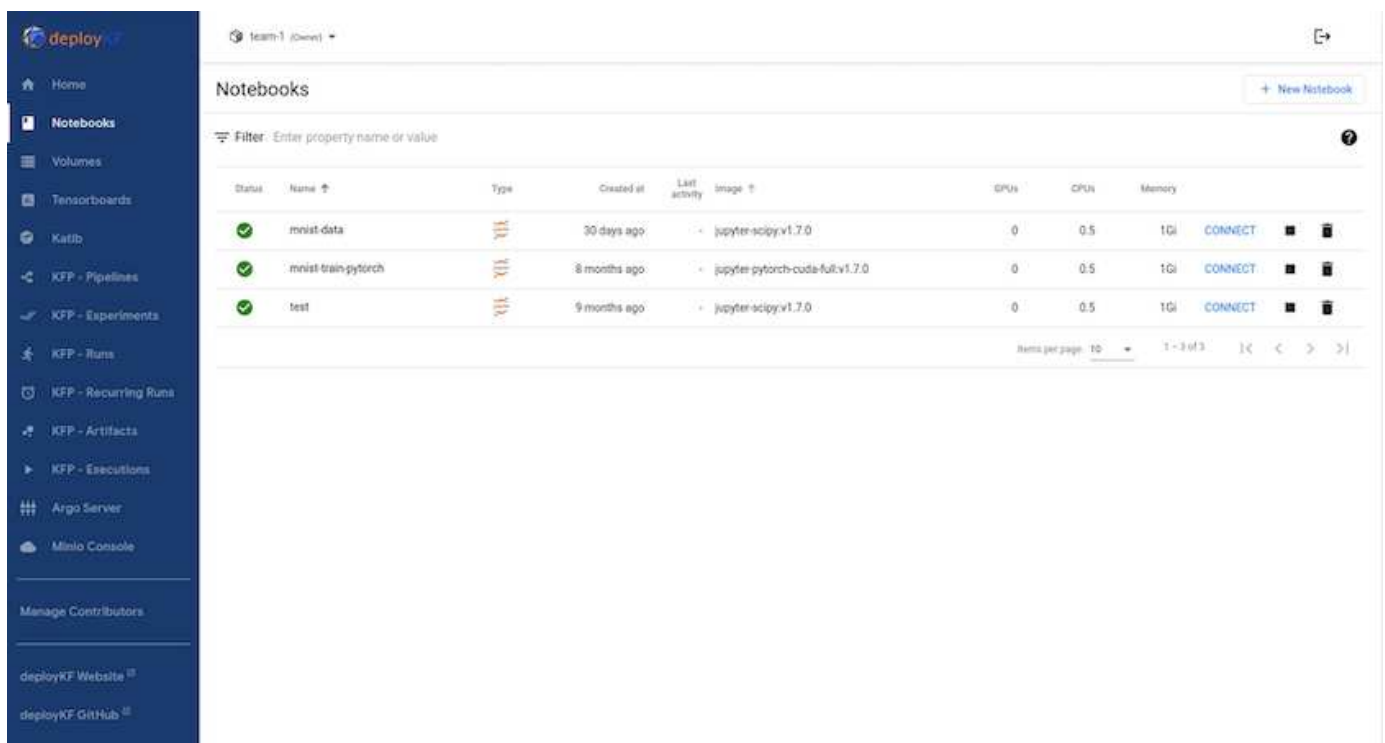


出于验证目的、我们使用部署了Kubeflow 1.7 "[部署KF](#)" 0.1.1.

## Kubeflow 操作和任务示例

为数据科学家或开发人员配置 **Jupyter** 笔记本电脑工作空间

Kubeflow 能够快速配置新的 Jupyter 笔记本电脑服务器，以充当数据科学家工作空间。有关 Kubeflow 上下文中 Jupyter 笔记本电脑的详细信息，请参见 "[Kubeflow 官方文档](#)"。



### 将NetApp数据操作工具包与Kubeflow结合使用

。 "[适用于 Kubernetes 的 NetApp 数据科学工具包](#)" 可与 Kubeflow 结合使用。将 NetApp 数据科学工具包与 Kubeflow 结合使用具有以下优势：

- 数据科学家可以直接从Jupyter笔记本中执行高级NetApp数据管理操作、例如创建快照和克隆。
- 使用Kubeflow管道框架、可以将高级NetApp数据管理操作(例如创建快照和克隆)整合到自动化工作流中。

请参见 "[Kubeflow 示例](#)" 有关将工具包与 Kubeflow 结合使用的详细信息，请参见 NetApp Data Science Toolkit GitHub 存储库中的一节。

## 示例工作流-使用Kubeflow和NetApp数据操作工具包训练图像识别模型

本节介绍使用Kubeflow和NetApp数据操作工具包培训和部署用于图像识别的神经网络所涉及的步骤。此示例用于展示整合了NetApp存储的培训作业。

### 前提条件

创建一个包含所需配置的文档、以用于Kubeflow管道中的训练和测试步骤。

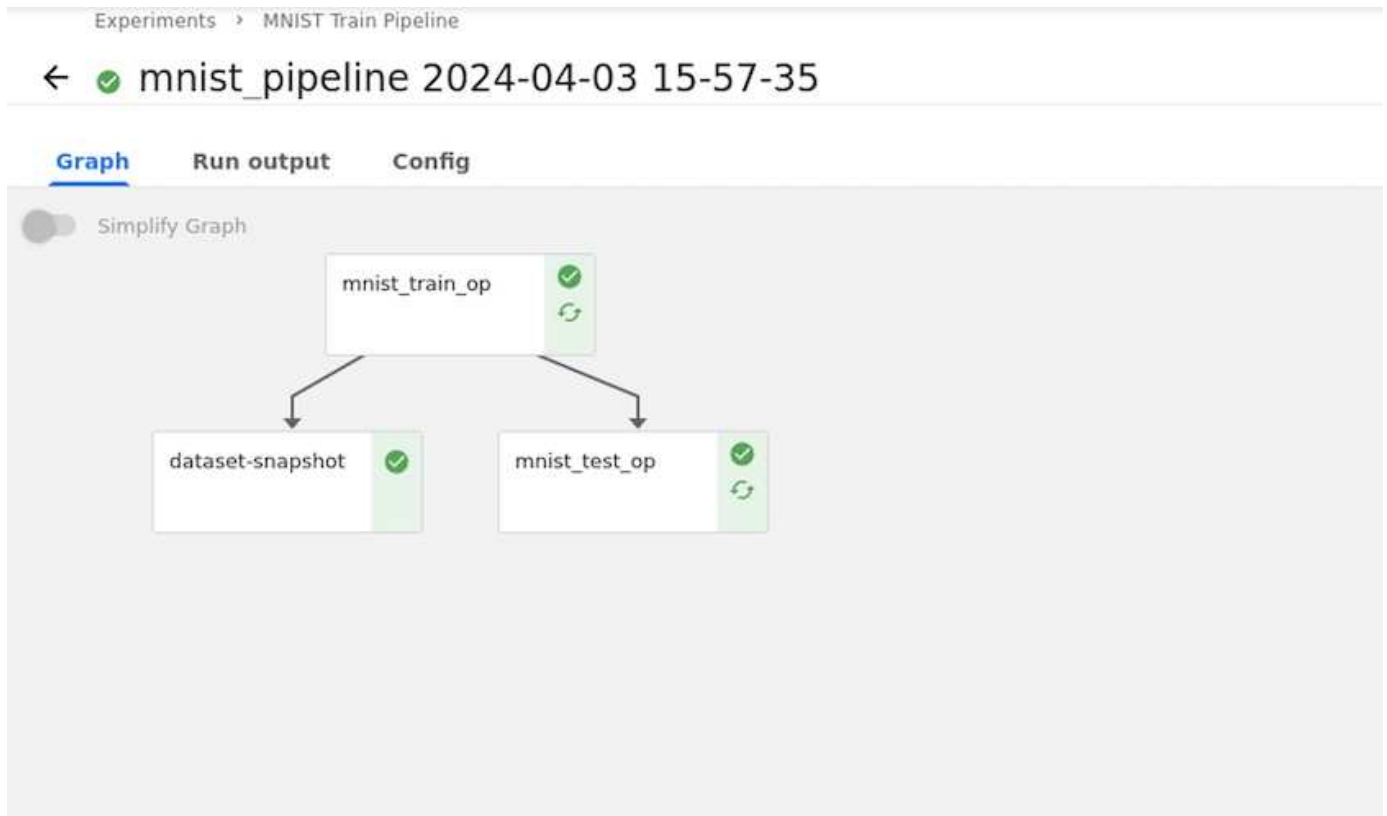
以下是一个多克文件示例-

```
FROM pytorch/pytorch:latest
RUN pip install torchvision numpy scikit-learn matplotlib tensorboard
WORKDIR /app
COPY . /app
COPY train_mnist.py /app/train_mnist.py
CMD ["python", "train_mnist.py"]
```

根据您的要求，安装运行该程序所需的所有必需库和软件包。在训练机器学习模型之前、我们假定您已部署有效的Kubeflow。

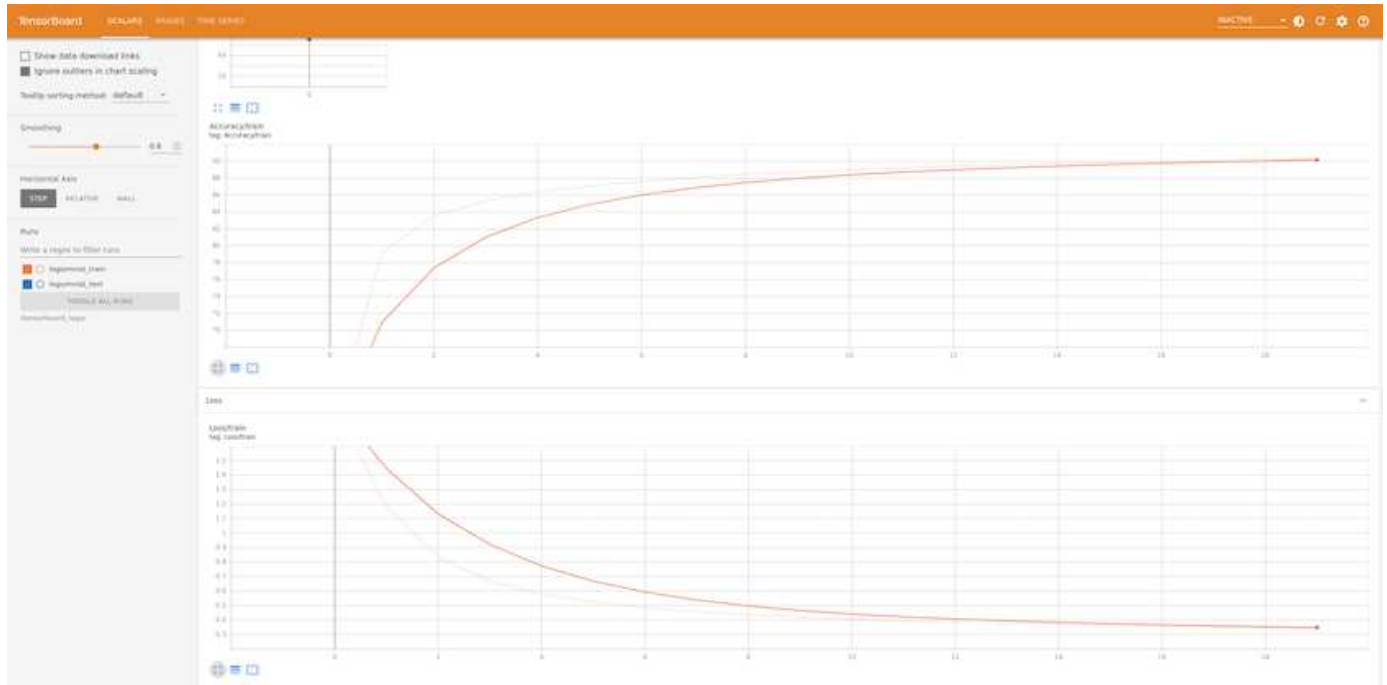
### 使用PyTorch和Kubeflow管道训练有关MNIST数据的小型NN

我们以一个小型神经网络为例、该网络是根据MNIST数据进行训练的。MNIST数据集由0到9位数字的字母图像组成。图像的大小为28x28像素。数据集分为60、000个训练影像和10、000个验证影像。用于此实验的神经网络是一个2层馈送网络。使用Kubeflow管道执行培训。请参见文档 ["此处"](#) 有关详细信息 ...我们的Kubeflow管道整合了"前提条件"部分中的Docker映像。



## 使用Tensorboard直观显示结果

模型训练完成后、我们可以使用Tensorboard直观显示结果。"Tensorboard" 作为Kubeflow信息板上的一项功能提供。您可以为您的作业创建自定义的tensorboard。以下示例显示了训练精度与的图解环比和训练损失的数量与时代的数量。



## 使用Katb试验超参数

"Katb." 是Kubeflow中的一个工具、可用于实验模型超参数。要创建实验、请先定义所需的指标/目标。这通常是测试准确性。定义指标后、选择要使用的超参数(优化器/learning\_rate /层数)。Katb使用用户定义的值执行超参数扫描、以找到满足所需度量的最佳参数组合。您可以在用户界面的每个部分中定义这些参数。或者,也可以使用必要的规范定义\*YAML\*文件。下面是一个Katis实验的示意图-

The screenshot shows the 'Experiment details' page in the deploy.k8s.io interface. The left sidebar contains navigation options like Home, Notebooks, Volumes, Tensorboards, Katib, and KFP - Pipelines. The main content area displays the following information:

- Objective:**
  - Name: Validation-accuracy
  - Type: maximize
  - Goal: 0.9
  - Additional metrics: Train-accuracy
- Trials:**
  - Max failed trials: 3
  - Max trials: 12
  - Parallel trials: 3
- Parameters:**
  - lr: Parameter type: double, Min: 0.01, Max: 0.03
  - num-layers: Parameter type: int, Min: 1, Max: 64
  - optimizer: Parameter type: categorical, sgd, adam, trl
- Algorithm:**
  - Name: grid
- Metrics collector:**
  - Collector type: File

The screenshot shows the 'Experiment details' page for an experiment named 'mnist-gytorch'. A message at the top states: "Couldn't find any successful Trial." Below this, there are tabs for OVERVIEW, TRIALS, DETAILS, and YAML. The OVERVIEW tab is active, showing the following details:

- Name: mnist-gytorch
- Status: Experiment is running
- Best trial: No optimal trial yet
- Best trial's params: No optimal trial yet
- Best trial performance: No optimal trial yet
- User defined goal: Validation-accuracy > 0.9
- Running trials: 3
- Failed trials: 0
- Succeeded trials: 0

Below the overview is an 'Experiment Conditions' section with a filter input field: "Filter: Enter property name or value".

## 使用NetApp快照保存数据以实现可跟踪性

在模型训练期间、我们可能希望保存训练数据集的快照、以便于跟踪。为此、我们可以向管道中添加Snapshot步骤、如下所示。要创建快照、可以使用 "适用于Kubernetes的NetApp DataOps工具包"。

```

@dsl.pipeline(
    name = 'MNIST Classification Pipeline',
    description = 'Train a simple NN for classification'
)
def mnist_pipeline():
    mnist_train_task = mnist_train_op()
    mnist_train_task.apply(
        kfp.onprem.mount_pvc('mnist-data', 'mnist-data-vol', '/mnt/data/')
    )

    mnist_test_task = mnist_test_op()
    mnist_test_task.apply(
        kfp.onprem.mount_pvc('mnist-data', 'mnist-data-vol', '/mnt/data/')
    )

    volume_snapshot_name = "mnist-pytorch-snapshot"
    dataset_snapshot = dsl.ContainerOp(
        name="dataset-snapshot",
        image="python:3.9",
        command=["/bin/bash", "-c"],
        arguments=["\
            python3 -m pip install netapp-dataops-k8s && \
            echo "" + volume_snapshot_name + "" > /volume_snapshot_name.txt && \
            netapp_dataops_k8s_cli.py create volume-snapshot --pvc-name=" + "mnist-data" + " --snapshot-name=" + str(volume_snapshot_name) + " --namespace={work[low.namespace]}",
            file_outputs={"volume_snapshot_name": "/volume_snapshot_name.txt"}
        ]
    )
    mnist_test_task.after(mnist_train_task)
    dataset_snapshot.after(mnist_train_task)

```

请参见 ["适用于Kubeflow的NetApp数据操作工具包示例"](#) 有关详细信息 ...



## 版权信息

版权所有 © 2024 NetApp, Inc.。保留所有权利。中国印刷。未经版权所有者事先书面许可，本档中受版权保护的任何部分不得以任何形式或通过任何手段（图片、电子或机械方式，包括影印、录音、录像或存储在电子检索系统中）进行复制。

从受版权保护的 NetApp 资料派生的软件受以下许可和免责声明的约束：

本软件由 NetApp 按“原样”提供，不含任何明示或暗示担保，包括但不限于适销性以及针对特定用途的适用性的隐含担保，特此声明不承担任何责任。在任何情况下，对于因使用本软件而以任何方式造成的任何直接性、间接性、偶然性、特殊性、惩罚性或后果性损失（包括但不限于购买替代商品或服务；使用、数据或利润方面的损失；或者业务中断），无论原因如何以及基于何种责任理论，无论出于合同、严格责任或侵权行为（包括疏忽或其他行为），NetApp 均不承担责任，即使已被告知存在上述损失的可能性。

NetApp 保留在不另行通知的情况下随时对本文档所述的任何产品进行更改的权利。除非 NetApp 以书面形式明确同意，否则 NetApp 不承担因使用本文档所述产品而产生的任何责任或义务。使用或购买本产品不表示获得 NetApp 的任何专利权、商标权或任何其他知识产权许可。

本手册中描述的产品可能受一项或多项美国专利、外国专利或正在申请的专利的保护。

有限权利说明：政府使用、复制或公开本文档受 DFARS 252.227-7013（2014 年 2 月）和 FAR 52.227-19（2007 年 12 月）中“技术数据权利 — 非商用”条款第 (b)(3) 条规定的限制条件的约束。

本文档中所含数据与商业产品和/或商业服务（定义见 FAR 2.101）相关，属于 NetApp, Inc. 的专有信息。根据本协议提供的所有 NetApp 技术数据和计算机软件具有商业性质，并完全由私人出资开发。美国政府对这些数据的使用权具有非排他性、全球性、受限且不可撤销的许可，该许可既不可转让，也不可再许可，但仅限在与交付数据所依据的美国政府合同有关且受合同支持的情况下使用。除本文档规定的情形外，未经 NetApp, Inc. 事先书面批准，不得使用、披露、复制、修改、操作或显示这些数据。美国政府对国防部的授权仅限于 DFARS 的第 252.227-7015(b)（2014 年 2 月）条款中明确的权利。

## 商标信息

NetApp、NetApp 标识和 <http://www.netapp.com/TM> 上所列的商标是 NetApp, Inc. 的商标。其他公司和产品名称可能是其各自所有者的商标。