Kubeflow

NetApp Solutions

NetApp
May 10, 2024

目录

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Kubenetes集群、并且正在运行的Kubeflow版本支持此Kubenetes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Kubbernetes集群中安装并配置NetApp Asta三端存储。有关Asta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Kubenetes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

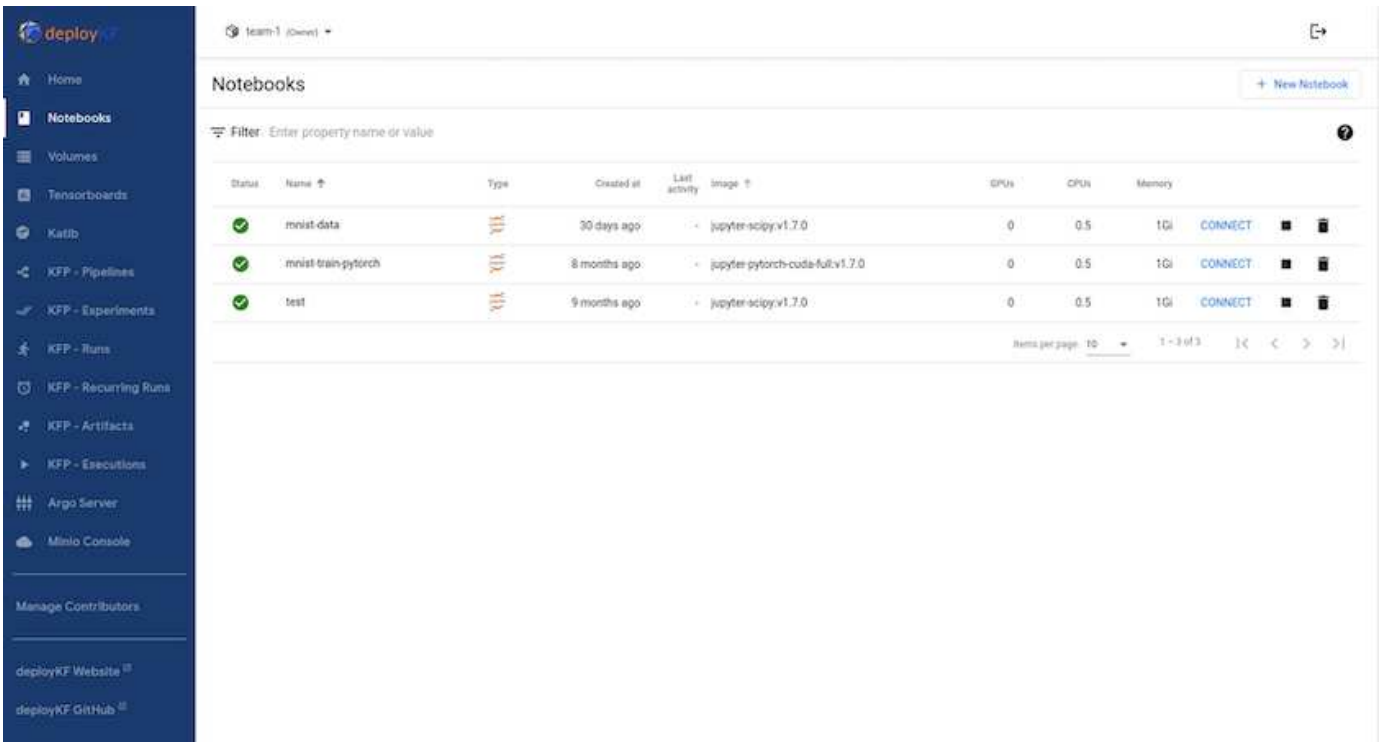
部署Kubeflow有许多不同的选项。请参见 "Kubeflow 官方文档" 有关部署选项的列表、请选择最适合您的需求的选项。

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

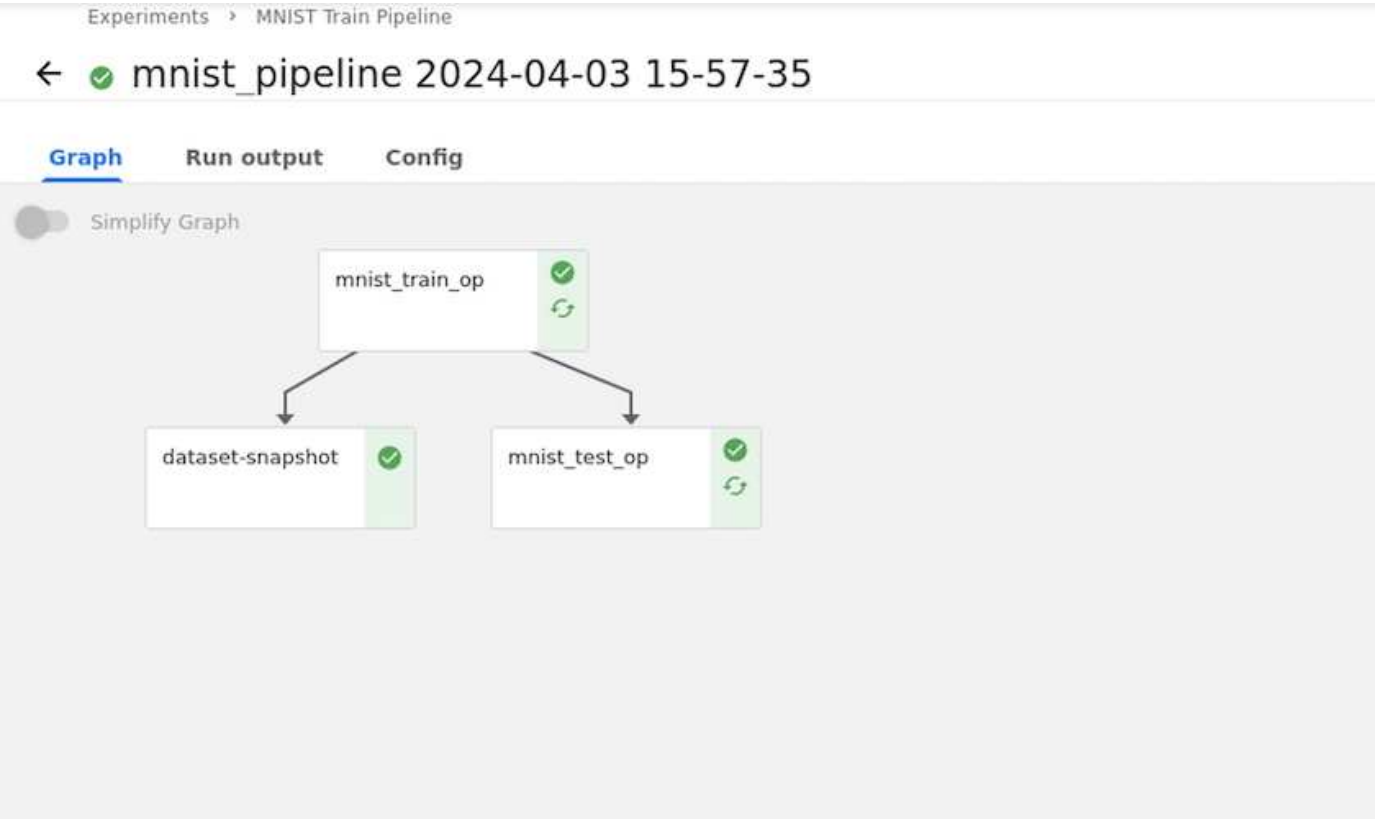
创建一个包含所需配置的文档、以用于Kubeflow管道中的训练和测试步骤。
以下是一个多克尔文件示例-

```
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。

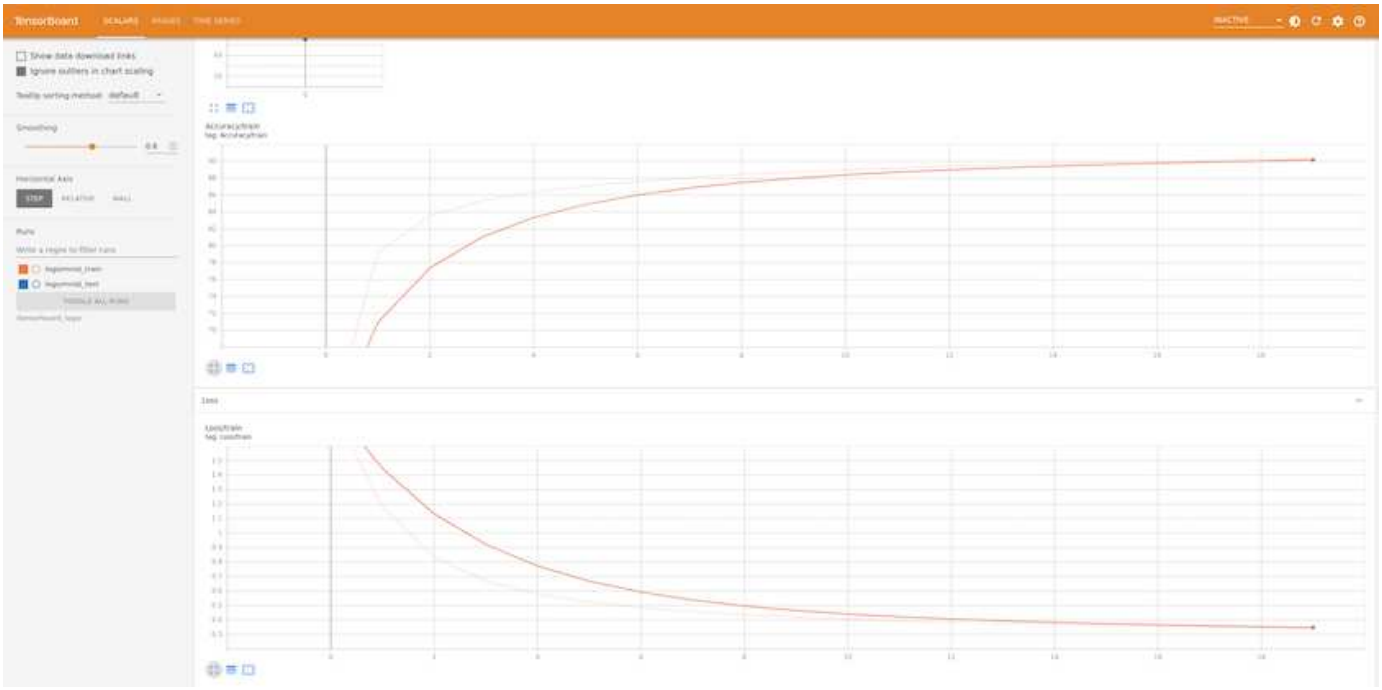
使用**PyTorch**和**Kubeflow**管道训练有关**MNIST**数据的小型**NN**

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。



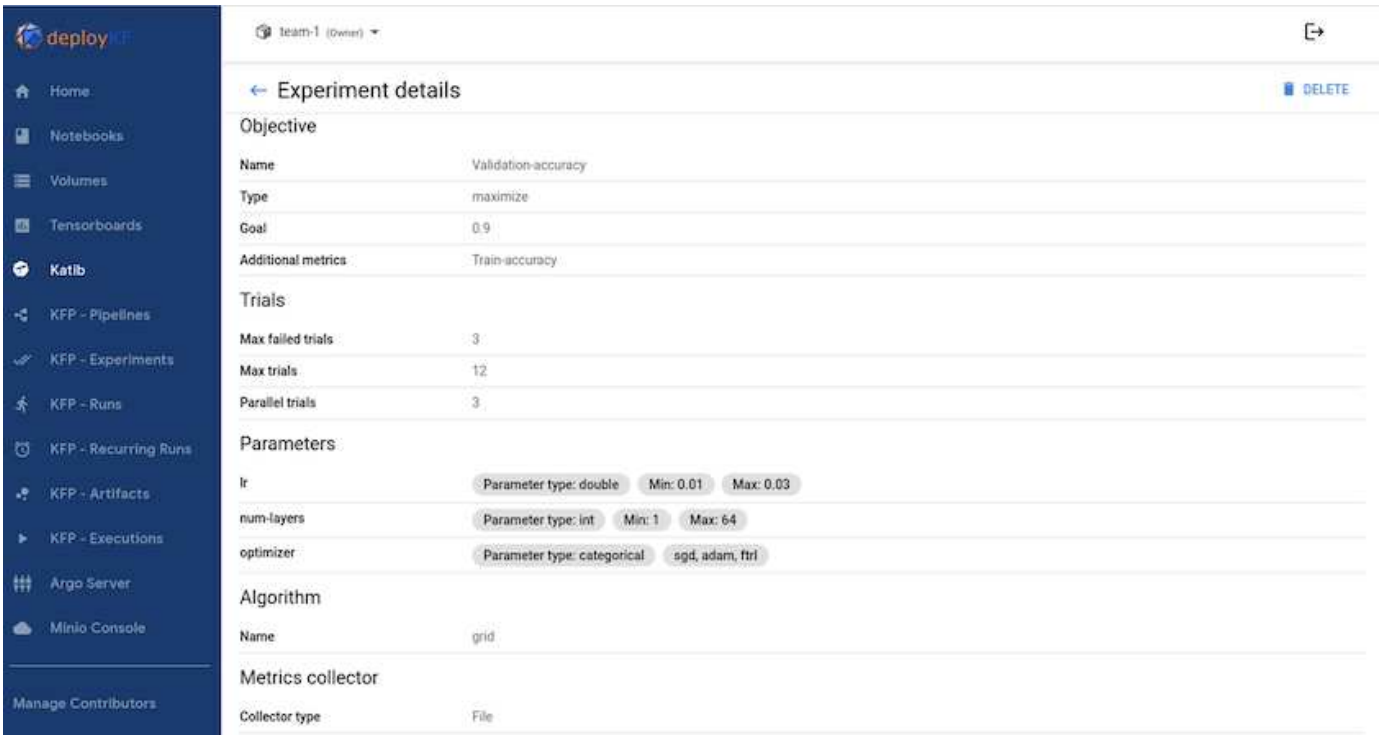
使用**Tensorboard**直观显示结果

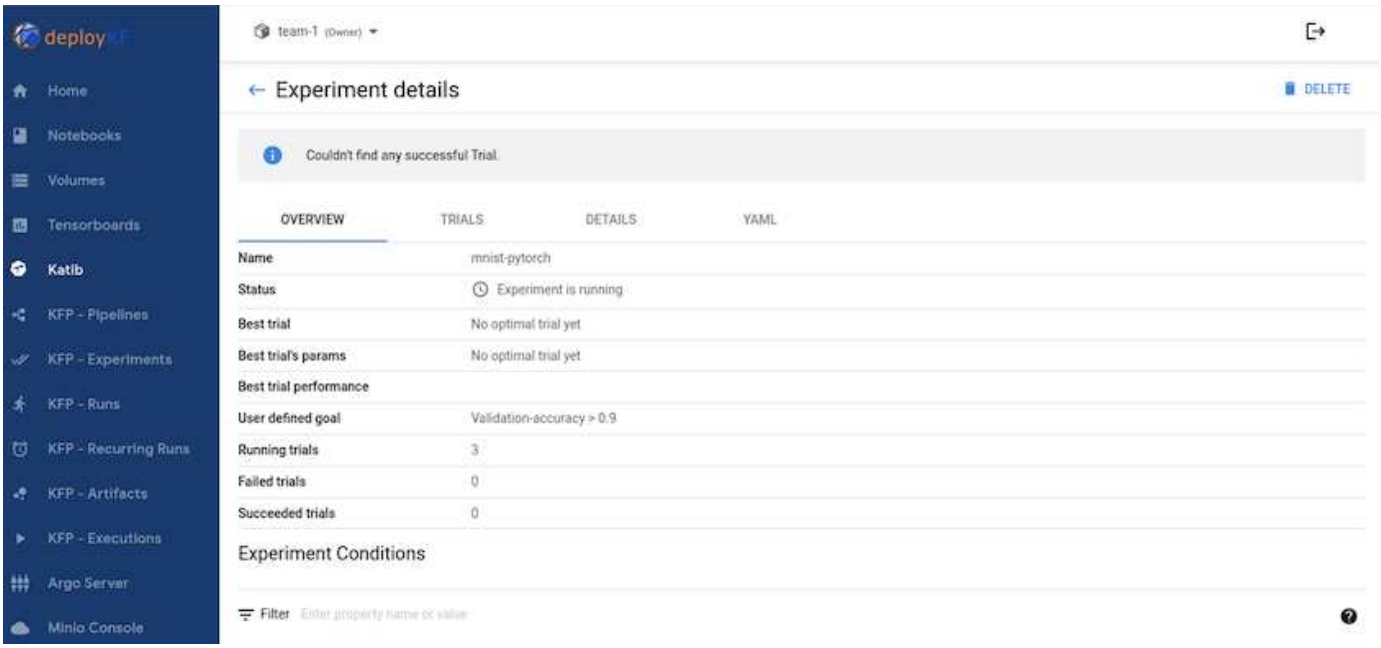
模型训练完成后、我们可以使用Tensorboard直观显示结果。 "Tensorboard" 作为Kubeflow信息板上的一项功能提供。您可以为您的作业创建自定义的tensorboard。以下示例显示了训练精度与的图解环比和训练损失的数量与时代的数量。



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/leining_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-





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={{workflow.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 …