응용 프로그램 배포

NetApp Solutions

NetApp
April 20, 2024

This PDF was generated from https://docs.netapp.com/ko-kr/netapp-solutions/ai/mlrun_get_code_from_github.html on April 20, 2024. Always check docs.netapp.com for the latest.

목차

응용 프로그램 배포 1
GitHub에서 코드를 가져옵니다 1
작업 환경을 구성합니다 1
Grafana 대시보드 배포 13

응용 프로그램 배포

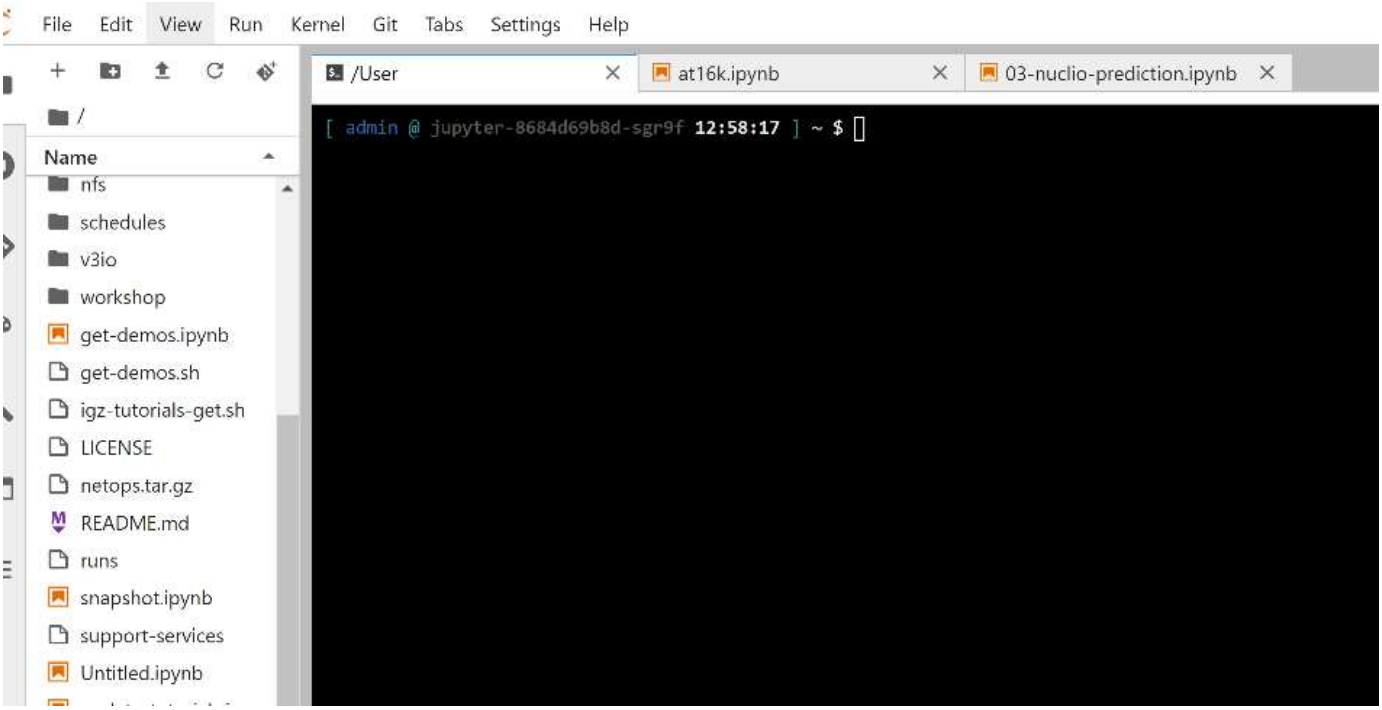
다음 섹션에서는 응용 프로그램을 설치하고 배포하는 방법에 대해 설명합니다.

GitHub에서 코드를 가져옵니다

이제 Iguazio 클러스터와 개발자 환경에서 NetApp Cloud Volume 또는 NetApp Trident 볼륨을 사용할 수 있으므로 애플리케이션 검토를 시작할 수 있습니다.

사용자는 고유한 작업 공간(디렉터리)을 가지고 있습니다. 모든 노트북에서 사용자 디렉토리 경로는 "/User"입니다. Iguazio 플랫폼은 디렉토리를 관리합니다. 위의 지침을 따르면 NetApp Cloud 볼륨은 '/NetApp' 디렉토리에 있습니다.

Jupyter 터미널을 사용하여 GitHub에서 코드를 가져옵니다.



Jupyter 터미널 프롬프트에서 프로젝트를 복제합니다.

```
cd /User
git clone .
```

이제 Jupyter 작업 공간의 파일 트리에 'netops-netapp' 폴더가 표시됩니다.

작업 환경을 구성합니다

'노트북'의 et_env-example.ipynb를 'et_env.ipynb'로 복사합니다. 'et_env.ipynb'를 열고 편집합니다. 이 노트북은 자격 증명, 파일 위치 및 실행 드라이버에 대한 변수를 설정합니다.

위의 지침을 따르면 다음 단계만 변경할 수 있습니다.

1. 이과지오 서비스 대시보드에서 이 값을 'docker_registry'로 얻습니다

   예: dddocker-registry.default-tenant.app.clusterq.iguaziodev.com:80`

2. 'admin'을 Iguazio 사용자 이름으로 변경합니다.

   "IGZ_container_path='/users/admin"

   다음은 ONTAP 시스템 연결 세부 정보입니다. Trident를 설치할 때 생성한 볼륨 이름을 포함합니다. 다음은 온-프레미스 ONTAP 클러스터에 대한 설정입니다.

   ```
   ontapClusterMgmtHostname = '0.0.0.0'
   ontapClusterAdminUsername = 'USER'
   ontapClusterAdminPassword = 'PASSWORD'
   sourceVolumeName = 'SOURCE VOLUME'
   ```

   Cloud Volumes ONTAP에 대한 설정은 다음과 같습니다.

```
MANAGER=ontapClusterMgmtHostname
svm='svm'
email='email'
password=ontapClusterAdminPassword
weid="weid"
volume=sourceVolumeName
```

기본 **Docker** 이미지를 생성합니다

ML 파이프라인을 구축하는 데 필요한 모든 것이 Iguazio 플랫폼에 포함되어 있습니다. 개발자는 파이프라인을 실행하고 Jupyter Notebook에서 이미지 생성을 실행하는 데 필요한 Docker 이미지의 사양을 정의할 수 있습니다. 노트북 'create-images.ipynb'를 열고 모든 셀을 실행합니다.

이 노트북은 파이프라인에서 사용하는 두 개의 이미지를 만듭니다.

- "iguazio/NetApp. ML 작업을 처리하는 데 사용됩니다.

  Create image for training pipeline

  ```
  [4]: fn.build_config(image=docker_registry+'/iguazio/netapp', commands=['pip install \
       v3io_frames fsspec>=0.3.3 PyYAML==5.1.2 pyarrow==0.15.1 pandas==0.25.3 matplotlib seaborn yellowb
       fn.deploy()
  ```

- 'NetApp/파이프라인' NetApp Snapshot 복사본을 처리하는 유틸리티를 포함합니다.

  Create image for Ontap utilitites

  ```
  [8]: fn.build_config(image=docker_registry + '/netapp/pipeline:latest',commands=['apt -y update','pip install v3io_frames netapp_ontap'
       fn.deploy()
  ```

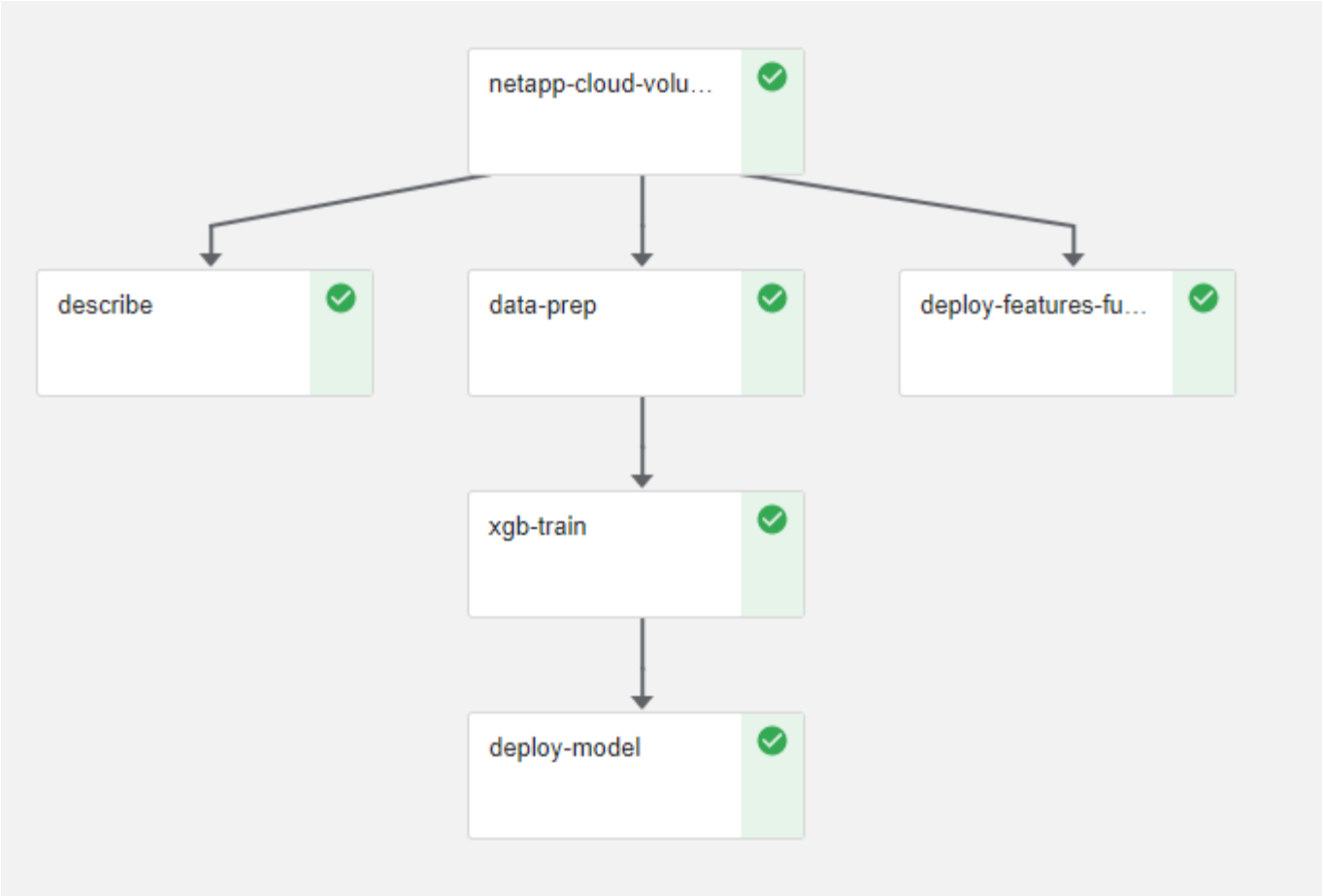
개별 **Jupyter Notebooks**를 검토합니다

다음 표에는 이 작업을 만드는 데 사용한 라이브러리와 프레임워크가 나와 있습니다. 이러한 모든 구성 요소는 Iguazio의 역할 기반 액세스 및 보안 제어와 완벽하게 통합되어 있습니다.

라이브러리/프레임워크	설명
MLRun	머신 러닝/AI 파이프라인을 조립, 실행 및 모니터링할 수 있도록 Iguazio에서 관리합니다.
뉴클레오	서버를 사용하지 않는 함수 프레임워크가 Iguazio와 통합되었습니다. 이과지오(Iguazio)에서 관리하는 오픈 소스 프로젝트로도 사용할 수 있습니다.
Kubeflow	파이프라인을 구축하는 Kubernetes 기반 프레임워크 이과지오가 기여하는 오픈 소스 프로젝트이기도 합니다. 이과지오와 통합되어 나머지 인프라와의 보안 및 통합을 강화합니다.
Docker 를 참조하십시오	Docker 레지스트리는 Iguazio 플랫폼에서 서비스로 실행됩니다. 이 설정을 변경하여 레지스트리에 연결할 수도 있습니다.
NetApp Cloud Volumes를 참조하십시오	AWS에서 실행되는 Cloud Volumes를 통해 대량의 데이터에 액세스하고 Snapshot 복사본을 교육에 사용되는 데이터 세트의 버전으로 만들 수 있습니다.
트라이던트	Trident는 NetApp에서 관리하는 오픈 소스 프로젝트입니다. Kubernetes에서 스토리지 및 컴퓨팅 리소스와 통합할 수 있도록 지원합니다.

우리는 여러 노트북을 사용하여 ML 파이프라인을 구성하였습니다. 각 노트북은 파이프라인에 함께 들어가기 전에 개별적으로 테스트할 수 있습니다. 이 데모 애플리케이션의 배포 흐름에 따라 각 노트북을 개별적으로 다룹니다.

원하는 결과는 데이터의 스냅샷 복사본을 기반으로 모델을 교육하고 추론을 위해 모델을 구축하는 파이프라인입니다. 완료된 MLRun 파이프라인에 대한 블록 다이어그램이 다음 이미지에 나와 있습니다.



데이터 생성 기능 구축

이 섹션에서는 Nuclio 서버리스 기능을 사용하여 네트워크 장치 데이터를 생성하는 방법에 대해 설명합니다. 이 활용 사례는 파이프라인을 구축한 이과지오 클라이언트에서 수정되었으며 이과지오 서비스를 사용하여 네트워크 디바이스 장애를 모니터링하고 예측합니다.

네트워크 장치에서 오는 데이터를 시뮬레이션했습니다. Jupyter Notebook data-generator.ipynb를 실행하면 10분마다 실행되는 서버리스 기능이 생성되고 새 데이터가 있는 Parquet 파일이 생성됩니다. 함수를 배포하려면 이 전자 필기장의 모든 셀을 실행합니다. 를 참조하십시오 "Nuclio 웹 사이트" 이 노트북에서 잘 모르는 구성 요소를 검토합니다.

함수를 생성할 때 다음 주석이 있는 셀은 무시됩니다. 노트북의 모든 셀은 기능의 일부로 간주됩니다. '%nuclio magic' 을(를) 활성화하려면 Nuclio 모듈을 가져옵니다.

```
# nuclio: ignore
import nuclio
```

기능에 대한 사양에서는 함수가 실행되는 환경, 함수가 트리거되는 방식 및 사용되는 리소스를 정의했습니다.

```
spec = nuclio.ConfigSpec(config={"spec.triggers.inference.kind":"cron",

"spec.triggers.inference.attributes.interval" :"10m",
                                "spec.readinessTimeoutSeconds" : 60,
                                "spec.minReplicas" : 1},……
```

INIT_CONTEXT 함수는 함수 초기화 시 Nuclio Framework에 의해 호출됩니다.

```
def init_context(context):
    ….
```

함수에 없는 코드는 함수가 초기화될 때 호출됩니다. 이 함수를 호출하면 처리기 함수가 실행됩니다. 처리기의 이름을 변경하고 함수 사양에서 지정할 수 있습니다.

```
def handler(context, event):
            …
```

배포 전에 노트북에서 기능을 테스트할 수 있습니다.

```
%%time
# nuclio: ignore
init_context(context)
event = nuclio.Event(body='')
output = handler(context, event)
output
```

이 기능은 노트북에서 배포하거나 CI/CD 파이프라인에서 배포할 수 있습니다(이 코드 조정).

```
addr = nuclio.deploy_file(name='generator',project='netops',spec=spec,
tag='v1.1')
```

파이프라인 노트북

이 노트북은 이 설정을 위해 개별적으로 실행할 수 없습니다. 이 내용은 각 전자 필기장에 대한 검토일 뿐입니다. 파이프라인을 구성하는 요소로 호출한 것입니다. 개별적으로 실행하려면 MLRun 설명서를 검토하여 Kubernetes 작업으로 실행합니다.

SNAP_CV.iynb

이 노트북은 파이프라인의 시작 부분에 있는 Cloud Volume Snapshot 복사본을 처리합니다. 볼륨의 이름을 파이프라인 컨텍스트로 전달합니다. 이 노트북은 스냅샷 복사본을 처리하기 위해 셸 스크립트를 호출합니다. 파이프라인에서 실행되는 동안 실행 컨텍스트에는 실행에 필요한 모든 파일을 찾는 데 도움이 되는 변수가 포함되어 있습니다. 이 코드를

작성하는 동안 개발자는 이 코드를 실행하는 컨테이너의 파일 위치에 대해 걱정할 필요가 없습니다. 나중에 설명했듯이 이 응용 프로그램은 모든 종속성을 포함하여 배포되며 실행 컨텍스트를 제공하는 파이프라인 매개 변수의 정의입니다.

```
command = os.path.join(context.get_param('APP_DIR'),"snap_cv.sh")
```

생성된 스냅샷 복사본 위치는 파이프라인의 단계에서 사용할 MLRun 컨텍스트에 배치됩니다.

```
context.log_result('snapVolumeDetails',snap_path)
```

다음 세 개의 노트북은 병렬로 실행됩니다.

데이터 준비 **.ipynb**

원시 메트릭을 기능으로 전환하여 모델 교육을 활성화해야 합니다. 이 노트북은 Snapshot 디렉토리에서 원시 메트릭을 읽고 모델 훈련을 위한 기능을 NetApp 볼륨에 씁니다.

파이프라인 컨텍스트에서 실행되는 경우 입력 DATA_DIR에 스냅샷 복사 위치가 포함됩니다.

```
metrics_table = os.path.join(str(mlruncontext.get_input('DATA_DIR',
os.getenv('DATA_DIR','/netpp'))),
                              mlruncontext.get_param('metrics_table',
os.getenv('metrics_table','netops_metrics_parquet')))
```

ipynb 설명

수신 메트릭을 시각화하기 위해 Kubeflow 및 MLRun UI를 통해 사용할 수 있는 플롯 및 그래프를 제공하는 파이프라인 단계를 배포합니다. 각 실행에는 이 시각화 도구의 고유 버전이 있습니다.

```
ax.set_title("features correlation")
plt.savefig(os.path.join(base_path, "plots/corr.png"))
context.log_artifact(PlotArtifact("correlation",  body=plt.gcf()),
local_path="plots/corr.html")
```

Deploy-feature-function.ipynb

NetApp은 이상 징후를 찾기 위한 메트릭을 지속적으로 모니터링합니다. 이 노트북은 들어오는 메트릭에 대한 예측을 실행하는 데 필요한 기능을 생성하는 서버리스 기능을 생성합니다. 이 노트북은 함수 생성을 호출합니다. 기능 코드는 노트북 data-prep.ipynb에 있다. 이러한 목적을 위해 파이프라인에서 한 단계씩 동일한 전자 필기장을 사용합니다.

훈련**.iynb**

피처를 작성한 후 모델 교육을 시작합니다. 이 단계의 출력은 추론을 위해 사용할 모델입니다. 또한 각 실행(실험)을 추적하기 위해 통계를 수집합니다.

예를 들어 다음 명령은 해당 실험의 컨텍스트에 정확도 점수를 입력합니다. 이 값은 Kubeflow 및 MLRun에서 볼 수

있습니다.

```
context.log_result('accuracy',score)
```

deploy-추론**-function.ipynb**입니다

파이프라인의 마지막 단계는 모델을 서버리스 기능으로 구축하여 연속 추론을 수행하는 것입니다. 이 노트북은 'nuclio-추론-function.ipynb'에 정의된 서버리스 기능의 생성을 호출합니다.

파이프라인 검토 및 구축

파이프라인에서 모든 노트북을 함께 실행할 경우 실험을 지속적으로 실행하여 새로운 측정 지표를 기준으로 모델의 정확성을 재평가할 수 있습니다. 먼저 파이프라인 iptynb 노트북을 엽니다. NetApp과 Iguazio가 이 ML 파이프라인 구축을 단순화하는 방법을 자세히 설명 드리겠습니다.

MLRun을 사용하여 컨텍스트를 제공하고 파이프라인의 각 단계에 대한 리소스 할당을 처리합니다. MLRun API 서비스는 Iguazio 플랫폼에서 실행되며 Kubernetes 리소스와 상호 작용하는 지점입니다. 각 개발자는 리소스를 직접 요청할 수 없습니다. API는 요청을 처리하고 액세스 제어를 활성화합니다.

```
# MLRun API connection definition
mlconf.dbpath = 'http://mlrun-api:8080'
```

파이프라인은 NetApp Cloud Volumes 및 온프레미스 볼륨과 함께 사용할 수 있습니다. Cloud Volumes를 사용하기 위해 이 데모를 구축했지만 코드에서 온프레미스 실행 옵션을 확인할 수 있습니다.

```
# Initialize the NetApp snap fucntion once for all functions in a notebook
if [ NETAPP_CLOUD_VOLUME ]:
    snapfn =
code_to_function('snap',project='NetApp',kind='job',filename="snap_cv.ipyn
b").apply(mount_v3io())
    snap_params = {
    "metrics_table" : metrics_table,
    "NETAPP_MOUNT_PATH" : NETAPP_MOUNT_PATH,
    'MANAGER' : MANAGER,
    'svm' : svm,
    'email': email,
    'password': password ,
    'weid': weid,
    'volume': volume,
    "APP_DIR" : APP_DIR
       }
else:
    snapfn =
code_to_function('snap',project='NetApp',kind='job',filename="snapshot.ipy
nb").apply(mount_v3io())
….
snapfn.spec.image = docker_registry + '/netapp/pipeline:latest'
snapfn.spec.volume_mounts =
[snapfn.spec.volume_mounts[0],netapp_volume_mounts]
      snapfn.spec.volumes = [ snapfn.spec.volumes[0],netapp_volumes]
```

Jupyter 노트북을 Kubeflow 단계로 전환하는 데 필요한 첫 번째 작업은 코드를 함수로 전환하는 것입니다. 기능에는 해당 노트북을 실행하는 데 필요한 모든 사양이 있습니다. 전자 필기장을 아래로 스크롤하면 파이프라인의 모든 단계에 대한 기능을 정의하는 것을 볼 수 있습니다.

노트북의 일부입니다	설명
code_to_function> (MLRun 모듈의 일부)	함수 이름: 프로젝트 이름. 모든 프로젝트 아티팩트를 구성하는 데 사용됩니다. 이것은 MLRun UI에서 볼 수 있습니다. 있습니다. 이 경우에는 Kubernetes 작업입니다. 이는 Dask, MPI, 스파크k8s 등이 될 수 있습니다. 자세한 내용은 MLRun 설명서를 참조하십시오. 파일. 전자 필기장의 이름입니다. Git(HTTP)의 위치일 수도 있습니다.
이미지	이 단계에서 사용 중인 Docker 이미지의 이름입니다. 앞에서 create-image.ipynb 전자 필기장으로 이 기능을 만들었습니다.
volume_mounts 및 volume	런타임에 NetApp Cloud Volume을 마운트하기 위한 세부 정보

단계에 대한 매개 변수도 정의합니다.

```
params={   "FEATURES_TABLE":FEATURES_TABLE,
           "SAVE_TO" : SAVE_TO,
           "metrics_table" : metrics_table,
           'FROM_TSDB': 0,
           'PREDICTIONS_TABLE': PREDICTIONS_TABLE,
           'TRAIN_ON_LAST': '1d',
           'TRAIN_SIZE':0.7,
           'NUMBER_OF_SHARDS' : 4,
           'MODEL_FILENAME' : 'netops.v3.model.pickle',
           'APP_DIR' : APP_DIR,
           'FUNCTION_NAME' : 'netops-inference',
           'PROJECT_NAME' : 'netops',
           'NETAPP_SIM' : NETAPP_SIM,
           'NETAPP_MOUNT_PATH': NETAPP_MOUNT_PATH,
           'NETAPP_PVC_CLAIM' : NETAPP_PVC_CLAIM,
           'IGZ_CONTAINER_PATH' : IGZ_CONTAINER_PATH,
           'IGZ_MOUNT_PATH' : IGZ_MOUNT_PATH
            }
```

모든 단계에 대한 함수 정의가 있으면 파이프라인을 구성할 수 있습니다. 우리는 이 정의를 만들기 위해 'kfp' 모듈을 사용합니다. MLRun을 사용하는 것과 자체적으로 구축하는 것의 차이점은 코딩의 단순화 및 단축입니다.

정의한 기능은 MLRun의 AS_STEP 기능을 이용하여 STEP 부품으로 변한다.

스냅샷 단계 정의

스냅샷 기능을 시작하고 v3io를 소스로 출력 및 마운트합니다.

```
snap = snapfn.as_step(NewTask(handler='handler',params=snap_params),
name='NetApp_Cloud_Volume_Snapshot',outputs=['snapVolumeDetails','training
_parquet_file']).apply(mount_v3io())
```

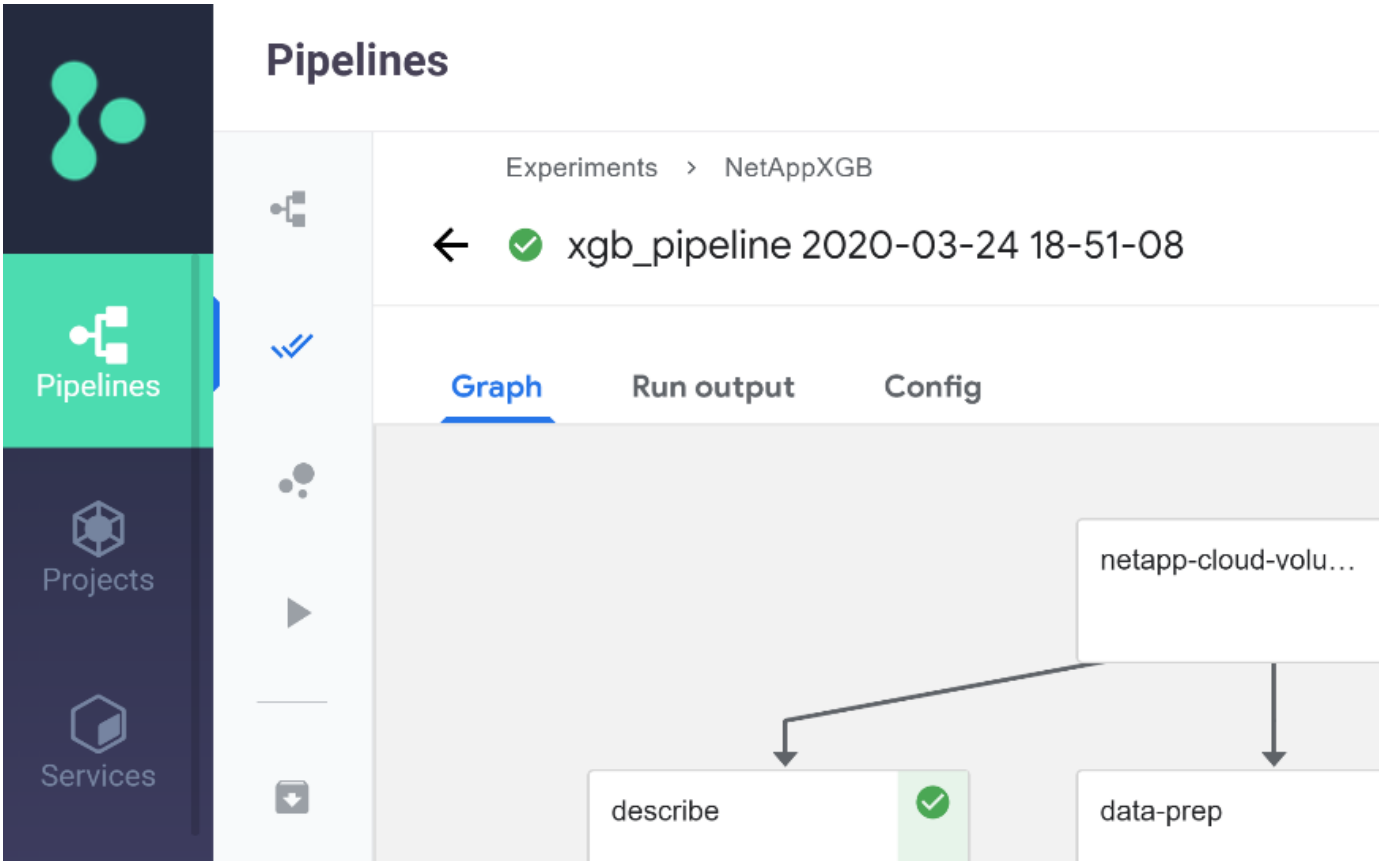
매개 변수	세부 정보
새 작업	NewTask 는 함수 실행의 정의입니다.
(MLRun 모듈)	핸들러. 호출할 Python 함수의 이름입니다. 전자 필기장에서 이름 처리기를 사용했지만 필수 사항은 아닙니다. 매개 변수 실행에 전달된 매개 변수. 코드 안에서 context.get_param('parameter')을 사용하여 값을 가져옵니다.
AS_STEP	이름. Kubeflow 파이프라인 단계의 이름입니다. 출력. 이 값은 완료 시 단계에서 사전에 추가하는 값입니다. SNAP_CV.iynb 노트북을 살펴보십시오. mount_v3io(). 이를 통해 파이프라인을 실행하는 사용자에 대해 /User를 마운트하는 단계를 구성합니다.

```
prep = data_prep.as_step(name='data-prep',
handler='handler',params=params,
                          inputs = {'DATA_DIR':
snap.outputs['snapVolumeDetails']} ,

out_path=artifacts_path).apply(mount_v3io()).after(snap)
```

매개 변수	세부 정보
입력	이전 단계의 출력을 단계별로 전달할 수 있습니다. 이 경우 snap.outputs ['sapVolumeDetails']는 스냅 단계에서 생성한 스냅샷 복사본의 이름입니다.
아웃_경로	MLRun 모듈 log_artifacts를 사용하여 생성하는 아티팩트를 배치할 위치입니다.

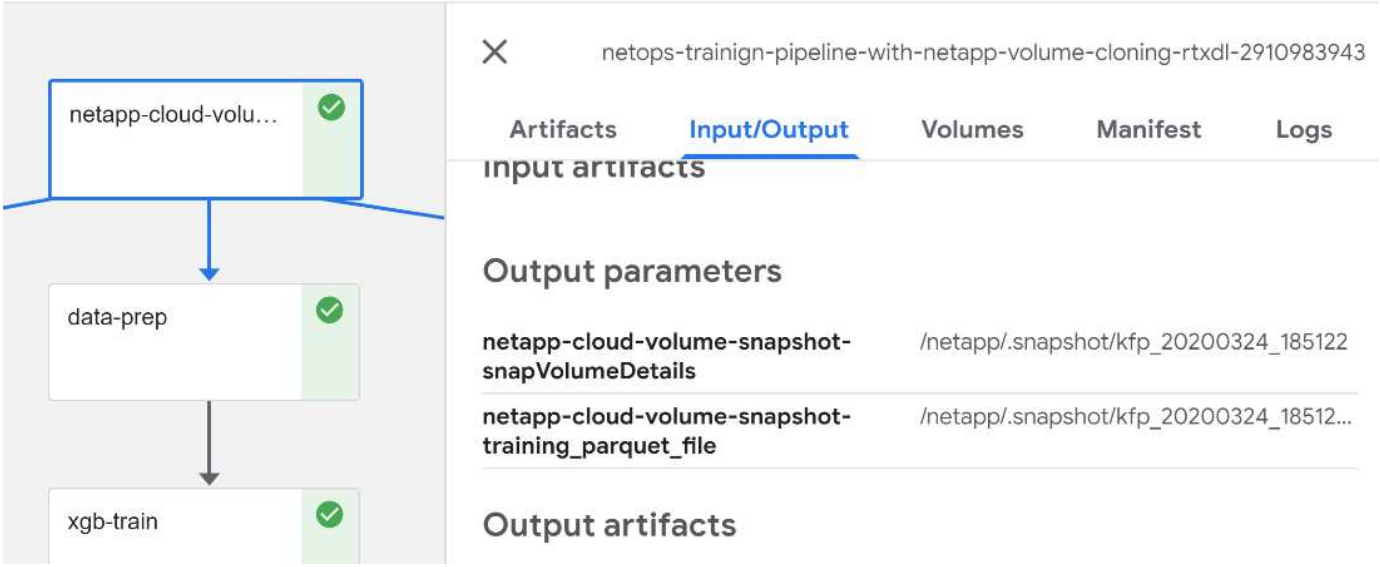
pipeline.ipynb는 위에서 아래로 실행할 수 있다. 그런 다음 Iguazio 대시보드에서 Pipelines 탭으로 이동하여 Iguazio 대시보드 파이프라인 탭에 표시된 진행 상황을 모니터링할 수 있습니다.



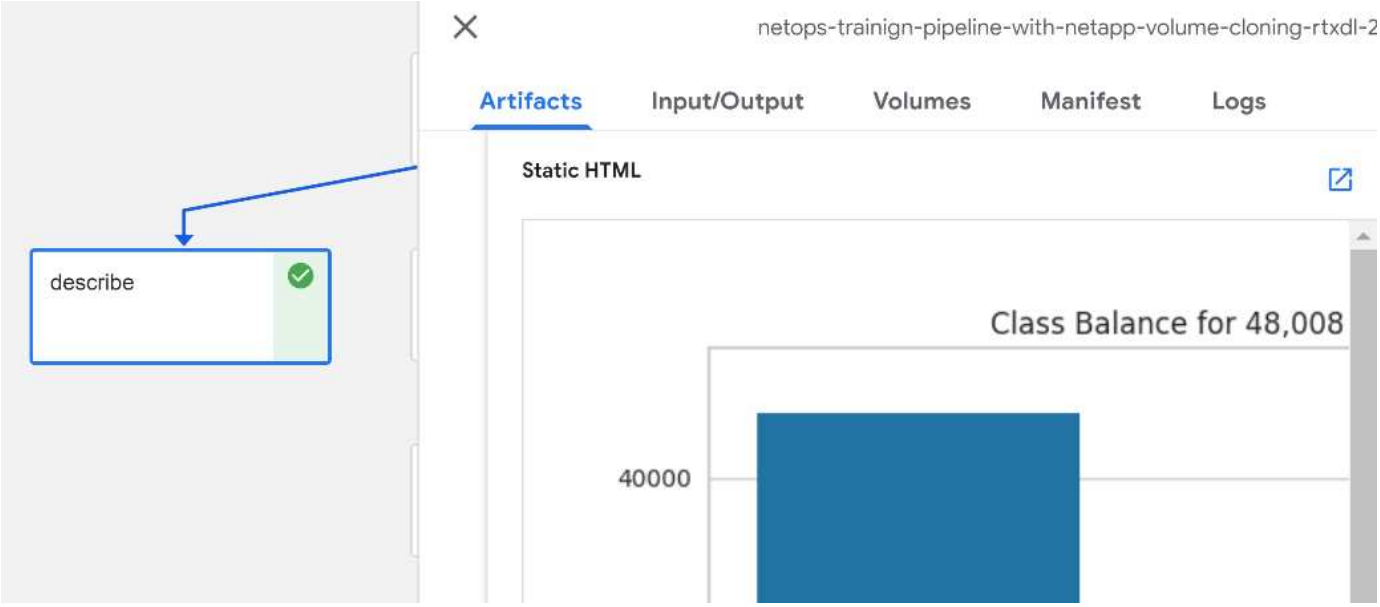
모든 러닝에서 훈련 단계의 정확성을 기록했기 때문에 훈련 정확도 기록에서도 볼 수 있듯이 각 실험마다 정확한 기록을 가지고 있습니다.

Run name	Status	Duration	Pipeline Version	Recurring ...	Start time	accuracy
xgb_pipeline 2020-03-24 18-51-...	✔	0:08:43	[View pipeline]	-	3/24/2020, 2:51:09 PM	0.985
xgb_pipeline 2020-03-19 13-31-...	✔	0:08:14	[View pipeline]	-	3/19/2020, 9:31:19 AM	0.980
xgb_pipeline 2020-03-18 12-56-...	✔	0:08:11	[View pipeline]	-	3/18/2020, 8:56:08 AM	0.990
xgb_pipeline 2020-03-17 19-49-...	✔	0:08:03	[View pipeline]	-	3/17/2020, 3:49:31 PM	0.985
xgb_pipeline 2020-03-17 18-34-...	✔	0:05:54	[View pipeline]	-	3/17/2020, 2:34:56 PM	0.980
xgb_pipeline 2020-03-17 17-34-...	✔	0:04:48	[View pipeline]	-	3/17/2020, 1:34:16 PM	0.982
xgb_pipeline 2020-03-17 17-01-...	✔	0:05:25	[View pipeline]	-	3/17/2020, 1:01:58 PM	0.987
xgb_pipeline 2020-03-16 16-47-...	✔	0:06:08	[View pipeline]	-	3/16/2020, 12:47:19 ...	0.983
xgb_pipeline 2020-03-16 13-57-...	✔	0:05:18	[View pipeline]	-	3/16/2020, 9:57:03 AM	0.980

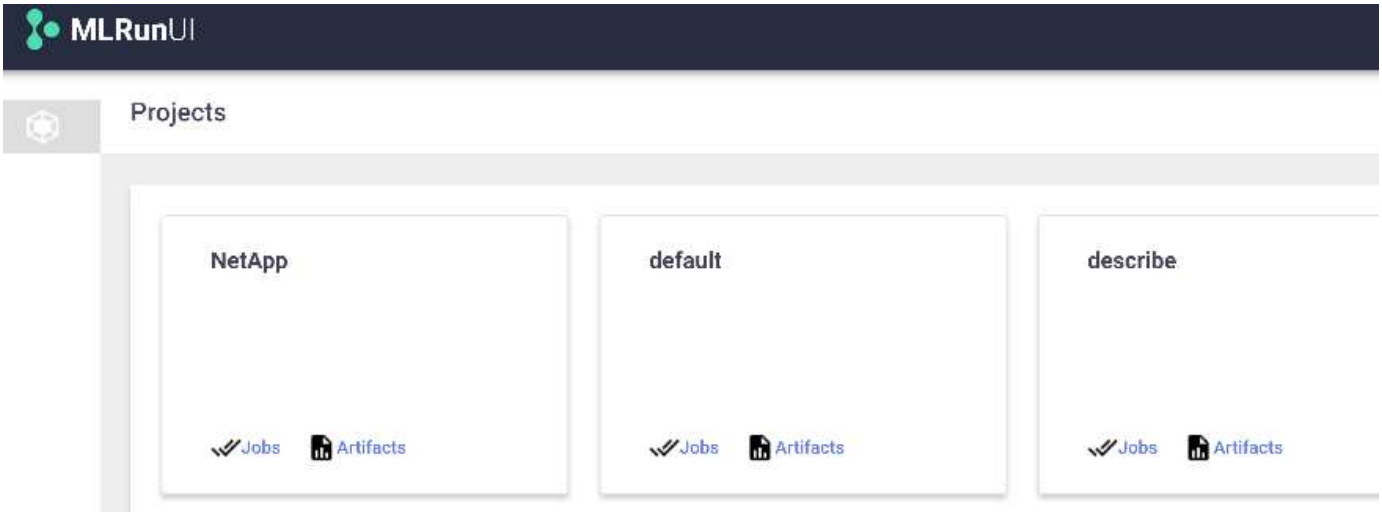
스냅샷 단계를 선택하면 이 실험을 실행하는 데 사용된 스냅샷 복사본의 이름을 볼 수 있습니다.



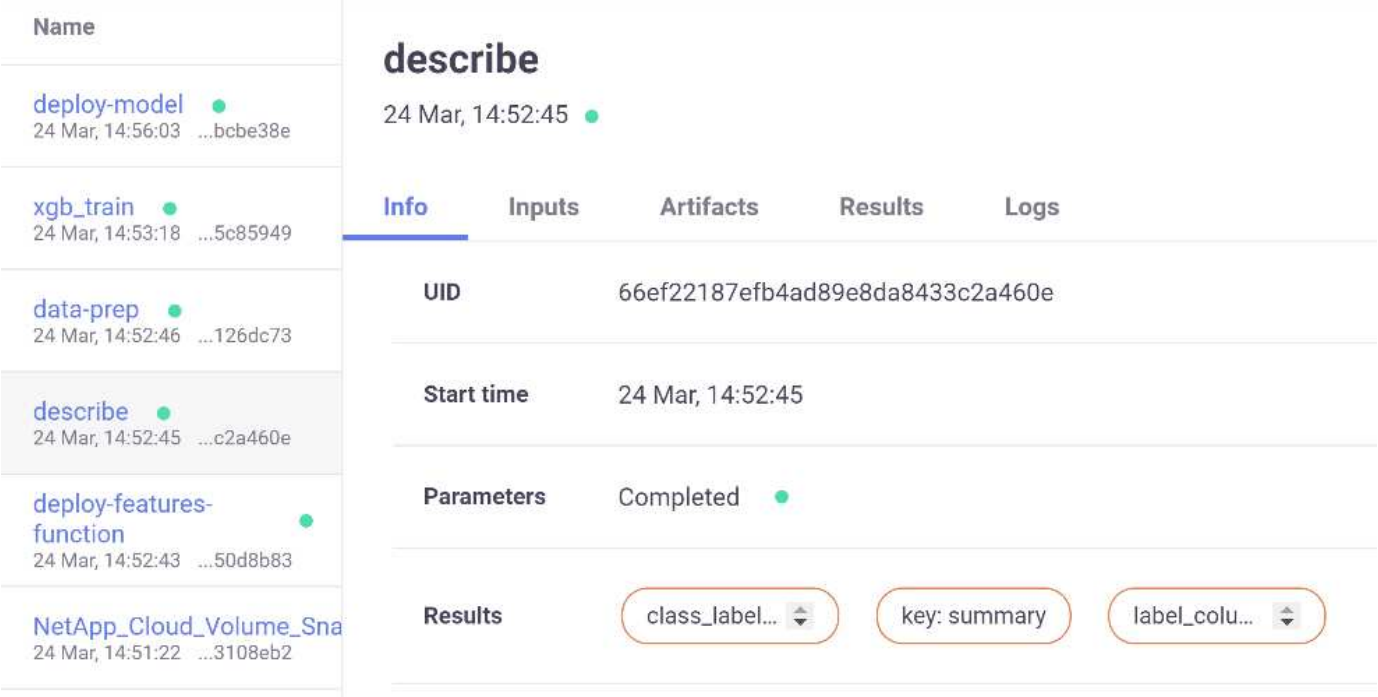
설명된 단계에는 우리가 사용한 지표를 탐색할 수 있는 시각적 인공물이 있습니다. 다음 이미지와 같이 전체 플롯을 보기 위해 확장할 수 있습니다.



또한 MLRun API 데이터베이스는 프로젝트별로 구성된 각 실행의 입력, 출력 및 아티팩트를 추적합니다. 각 시리즈의 입력, 출력 및 아티팩트의 예는 다음 영상에서 확인할 수 있습니다.



각 직무마다 추가 세부 정보를 저장합니다.



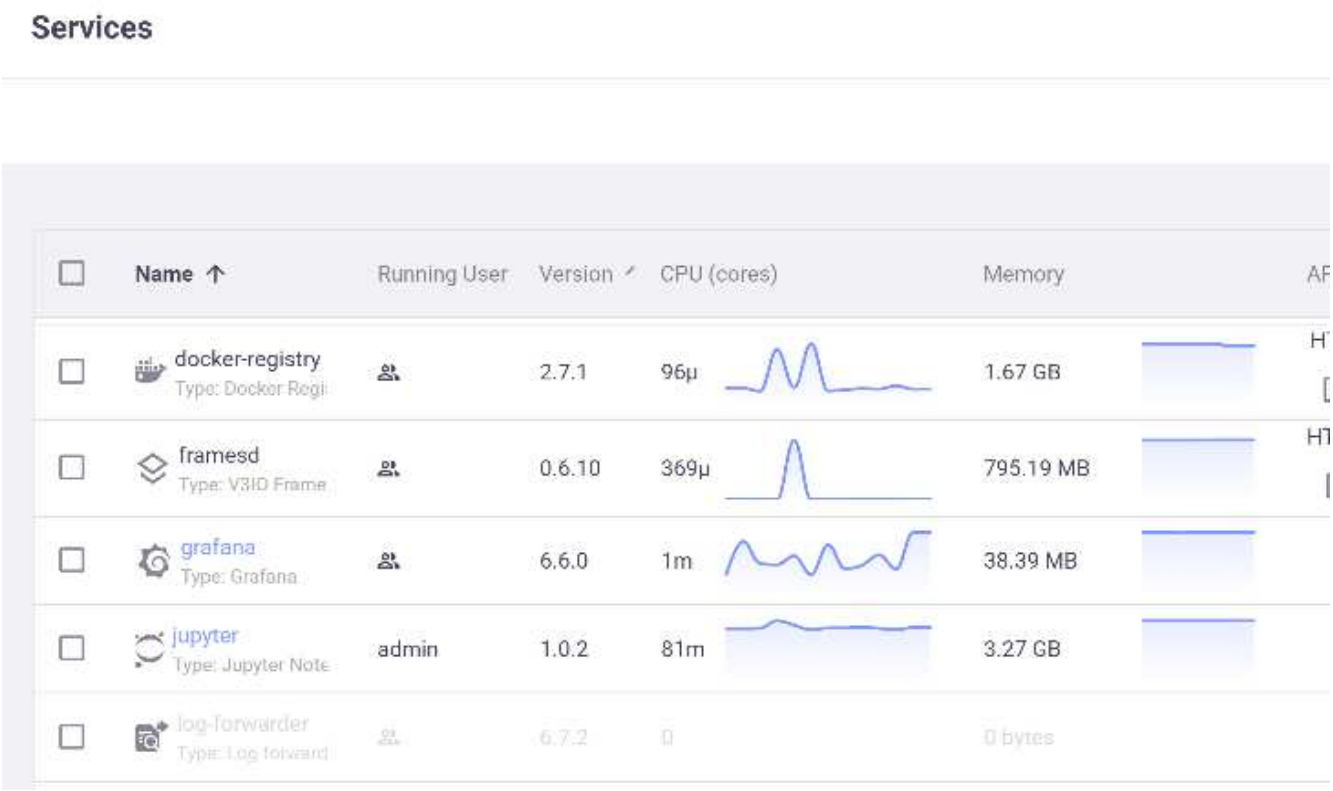
MLRun에 대한 자세한 내용은 이 문서에서 다룰 수 있는 것보다 많습니다. 단계와 함수의 정의를 비롯한 AI 아티팩트는 API 데이터베이스에 저장하고 버전을 지정한 후 개별 또는 전체 프로젝트로 호출할 수 있습니다. 프로젝트를 저장하고 나중에 사용할 수 있도록 Git에 푸시할 수도 있습니다. 자세한 내용은 에서 확인하시기 바랍니다 "MLRun GitHub 사이트".

Grafana 대시보드 배포

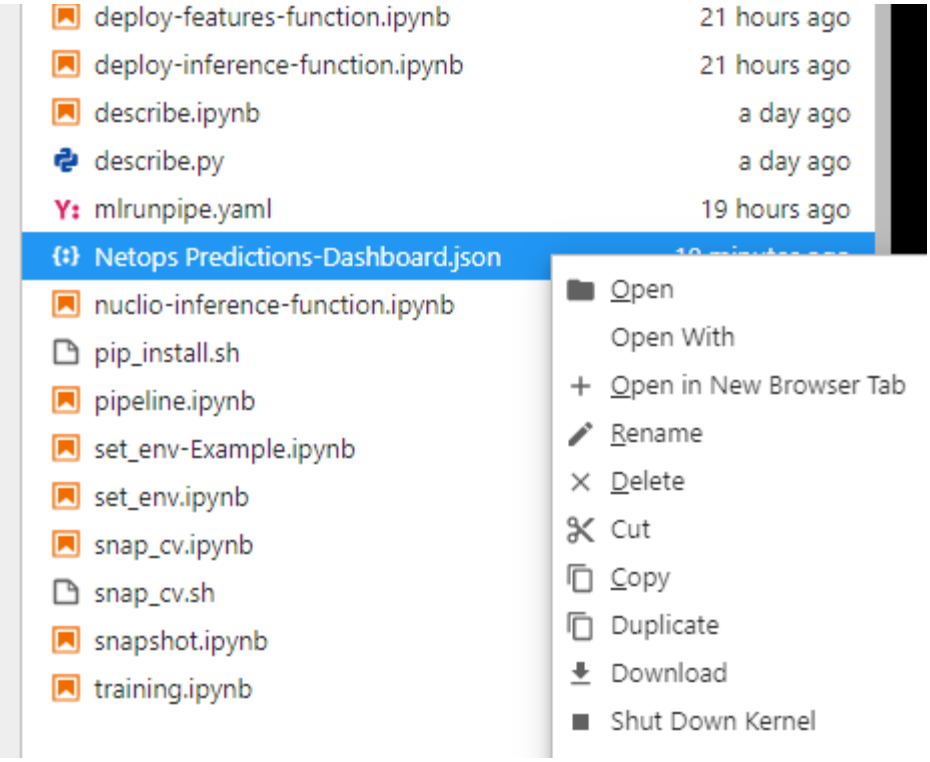
모든 것이 배포되면 새 데이터에 대한 추론을 실행합니다. 이 모델은 네트워크 장치 장비의 고장을 예측합니다. 예측 결과는 Iguazio 시간 계열 테이블에 저장됩니다. Iguazio의 보안 및 데이터 액세스 정책과 통합된 플랫폼을 통해 Grafana로 결과를 시각화할 수 있습니다.

제공된 JSON 파일을 클러스터의 Grafana 인터페이스로 가져와 대시보드를 구축할 수 있습니다.

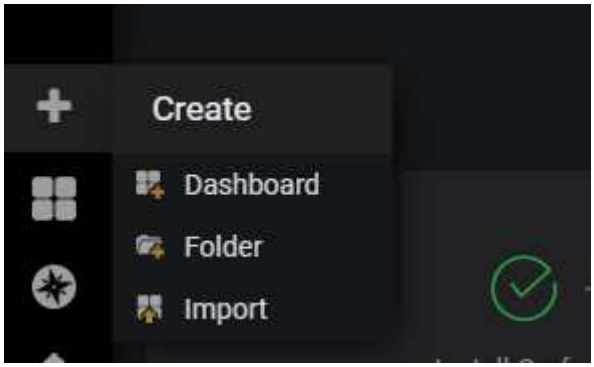
1. Grafana 서비스가 실행 중인지 확인하려면 서비스 아래를 살펴보십시오.



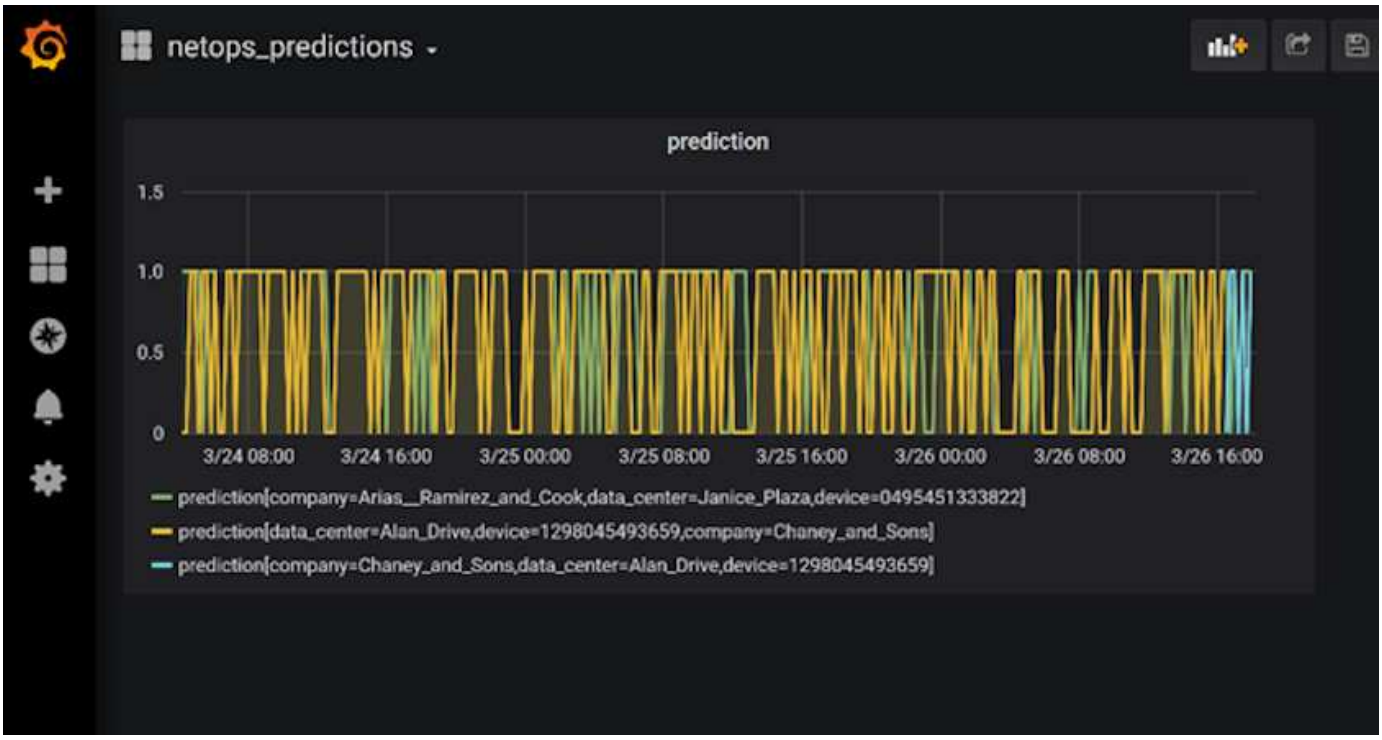
2. 이 인스턴스가 없으면 서비스 섹션에서 인스턴스를 배포합니다.
   a. 새 서비스를 클릭합니다.
   b. 목록에서 Grafana를 선택합니다.
   c. 기본값을 그대로 사용합니다.
   d. 다음 단계를 클릭합니다.
   e. 사용자 ID를 입력합니다.
   f. 서비스 저장 을 클릭합니다.
   g. 맨 위에 있는 변경 내용 적용을 클릭합니다.
3. 대시보드를 배포하려면 Jupyter 인터페이스를 통해 NetopsPredictions-Dashboard.json 파일을 다운로드합니다.



4. 서비스 섹션에서 Grafana를 열고 대시보드를 가져옵니다.



5. Upload(*.json) File을 클릭하고 앞서 다운로드한 파일('NetopsPredictions - Dashboard.json')을 선택한다. 업로드가 완료되면 대시보드가 표시됩니다.



정리 기능을 배포합니다

많은 데이터를 생성할 때 정리 및 정리하는 것이 중요합니다. 이를 위해 정리 기능을 정리 iynb 공책에 배치한다.

이점

NetApp과 Iguazio는 Docker, Kubernetes와 같은 오케스트레이션 툴과 함께 Kubeflow, Apache Spark 및 TensorFlow 등의 필수 프레임워크를 구축하여 AI 및 ML 애플리케이션의 구축 속도를 높이고 단순화합니다. NetApp과 Iguazio는 엔드 투 엔드 데이터 파이프라인을 통합함으로써 수많은 고급 컴퓨팅 워크로드에서 발생하는 지연 시간과 복잡성을 줄이고 개발 및 운영 간의 격차를 효과적으로 좁혀줍니다. 데이터 과학자는 대규모 데이터 세트에서 쿼리를 실행하고 교육 단계 동안 권한 있는 사용자와 데이터 및 알고리즘 모델을 안전하게 공유할 수 있습니다. 컨테이너식 모델이 운영 환경에 준비되면 개발 환경에서 운영 환경으로 쉽게 이동할 수 있습니다.

저작권 정보

Copyright © 2024 NetApp, Inc. All Rights Reserved. 미국에서 인쇄된 본 문서의 어떠한 부분도 저작권 소유자의 사전 서면 승인 없이는 어떠한 형식이나 수단(복사, 녹음, 녹화 또는 전자 검색 시스템에 저장하는 것을 비롯한 그래픽, 전자적 또는 기계적 방법)으로도 복제될 수 없습니다.

NetApp이 저작권을 가진 자료에 있는 소프트웨어에는 아래의 라이센스와 고지사항이 적용됩니다.

본 소프트웨어는 NetApp에 의해 '있는 그대로' 제공되며 상품성 및 특정 목적에의 적합성에 대한 명시적 또는 묵시적 보증을 포함하여(이에 제한되지 않음) 어떠한 보증도 하지 않습니다. NetApp은 대체품 또는 대체 서비스의 조달, 사용 불능, 데이터 손실, 이익 손실, 영업 중단을 포함하여(이에 국한되지 않음), 이 소프트웨어의 사용으로 인해 발생하는 모든 직접 및 간접 손해, 우발적 손해, 특별 손해, 징벌적 손해, 결과적 손해의 발생에 대하여 그 발생 이유, 책임론, 계약 여부, 엄격한 책임, 불법 행위(과실 또는 그렇지 않은 경우)와 관계없이 어떠한 책임도 지지 않으며, 이와 같은 손실의 발생 가능성이 통지되었다 하더라도 마찬가지입니다.

NetApp은 본 문서에 설명된 제품을 언제든지 예고 없이 변경할 권리를 보유합니다. NetApp은 NetApp의 명시적인 서면 동의를 받은 경우를 제외하고 본 문서에 설명된 제품을 사용하여 발생하는 어떠한 문제에도 책임을 지지 않습니다. 본 제품의 사용 또는 구매의 경우 NetApp에서는 어떠한 특허권, 상표권 또는 기타 지적 재산권이 적용되는 라이센스도 제공하지 않습니다.

본 설명서에 설명된 제품은 하나 이상의 미국 특허, 해외 특허 또는 출원 중인 특허로 보호됩니다.

제한적 권리 표시: 정부에 의한 사용, 복제 또는 공개에는 DFARS 252.227-7013(2014년 2월) 및 FAR 52.227-19(2007년 12월)의 기술 데이터-비상업적 품목에 대한 권리(Rights in Technical Data -Noncommercial Items) 조항의 하위 조항 (b)(3)에 설명된 제한사항이 적용됩니다.

여기에 포함된 데이터는 상업용 제품 및/또는 상업용 서비스(FAR 2.101에 정의)에 해당하며 NetApp, Inc.의 독점 자산입니다. 본 계약에 따라 제공되는 모든 NetApp 기술 데이터 및 컴퓨터 소프트웨어는 본질적으로 상업용이며 개인 비용만으로 개발되었습니다. 미국 정부는 데이터가 제공된 미국 계약과 관련하여 해당 계약을 지원하는 데에만 데이터에 대한 전 세계적으로 비독점적이고 양도할 수 없으며 재사용이 불가능하며 취소 불가능한 라이센스를 제한적으로 가집니다. 여기에 제공된 경우를 제외하고 NetApp, Inc.의 사전 서면 승인 없이는 이 데이터를 사용, 공개, 재생산, 수정, 수행 또는 표시할 수 없습니다. 미국 국방부에 대한 정부 라이센스는 DFARS 조항 252.227-7015(b)(2014년 2월)에 명시된 권한으로 제한됩니다.

상표 정보

NETAPP, NETAPP 로고 및 http://www.netapp.com/TM에 나열된 마크는 NetApp, Inc.의 상표입니다. 기타 회사 및 제품 이름은 해당 소유자의 상표일 수 있습니다.