



Implementazione dell'applicazione

NetApp Solutions

NetApp
April 26, 2024

This PDF was generated from https://docs.netapp.com/it-it/netapp-solutions/ai/mlrun_get_code_from_github.html on April 26, 2024. Always check docs.netapp.com for the latest.

Sommario

- Implementazione dell'applicazione. 1
 - Ottieni codice da GitHub 1
 - Configurare l'ambiente di lavoro 1
 - Implementare Grafana Dashboard. 13

Implementazione dell'applicazione

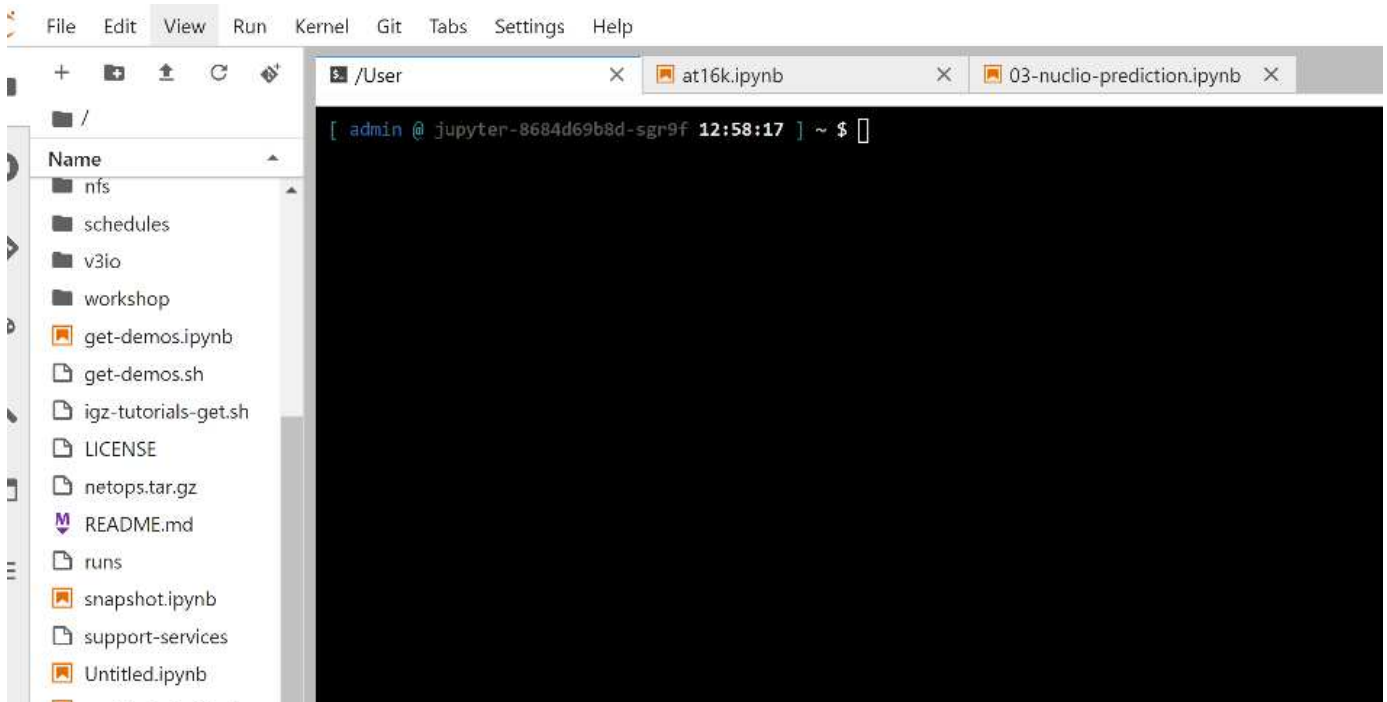
Le sezioni seguenti descrivono come installare e implementare l'applicazione.

Ottieni codice da GitHub

Ora che il volume NetApp Cloud o NetApp Trident è disponibile per il cluster Iguazio e l'ambiente di sviluppo, è possibile iniziare a rivedere l'applicazione.

Gli utenti dispongono di un proprio spazio di lavoro (directory). Su ogni notebook, il percorso alla directory dell'utente è /User. La piattaforma Iguazio gestisce la directory. Se si seguono le istruzioni riportate sopra, il volume NetApp Cloud è disponibile in /netapp directory.

Ottieni il codice da GitHub utilizzando un terminale Jupyter.



Al prompt del terminale Jupyter, clonare il progetto.

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

A questo punto, viene visualizzata la netops- netapp Nella struttura dei file nell'area di lavoro Jupyter.

Configurare l'ambiente di lavoro

Copiare il Notebook set_env-Example.ipynb come set_env.ipynb. Aprire e modificare set_env.ipynb. Questo notebook imposta le variabili per le credenziali, le posizioni dei file e i driver di esecuzione.

Se si seguono le istruzioni riportate in precedenza, le uniche modifiche da apportare sono le seguenti:

1. Ottieni questo valore dalla dashboard dei servizi Iguazio: `docker_registry`

Esempio: `docker-registry.default-tenant.app.clusterq.iguaziodev.com:80`

2. Cambiare `admin` Al tuo nome utente Iguazio:

```
IGZ_CONTAINER_PATH = '/users/admin'
```

Di seguito sono riportati i dettagli di connessione del sistema ONTAP. Includere il nome del volume generato al momento dell'installazione di Trident. La seguente impostazione si intende per un cluster ONTAP on-premise:

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

La seguente impostazione è per Cloud Volumes ONTAP:

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

Creare immagini base Docker

Tutto ciò di cui hai bisogno per creare una pipeline ML è incluso nella piattaforma Iguazio. Lo sviluppatore può definire le specifiche delle immagini Docker richieste per eseguire la pipeline ed eseguire la creazione dell'immagine da Jupyter notebook. Aprire il notebook `create_images.ipynb` Ed eseguire tutte le celle.

Questo notebook crea due immagini che utilizziamo in pipeline.

- `iguazio/netapp`. Utilizzato per gestire le attività ML.

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/pipeline`. Contiene utility per gestire le copie Snapshot di NetApp.

Create image for Ontap utilities

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

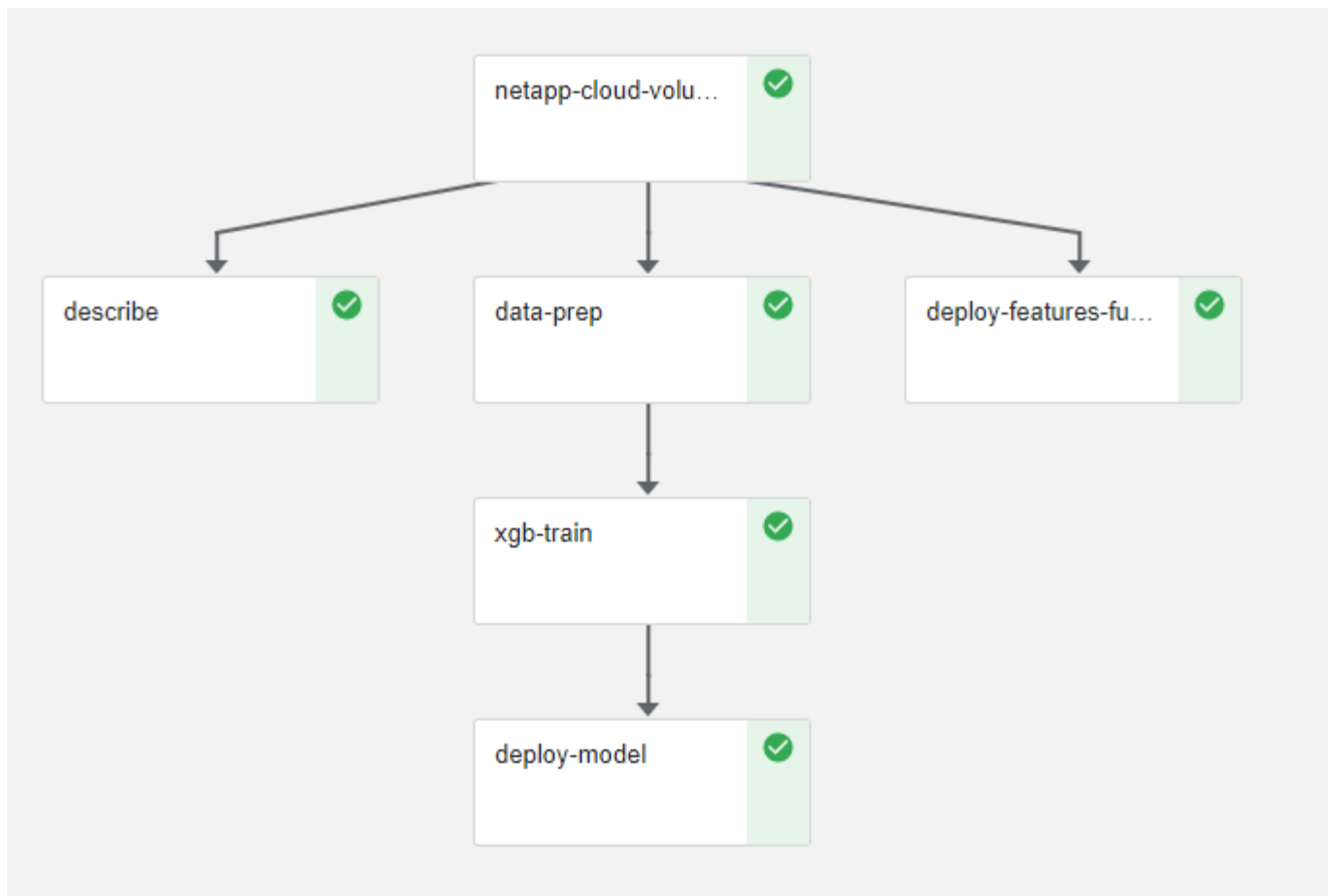
Esamina i singoli notebook Jupyter

La tabella seguente elenca le librerie e i framework utilizzati per creare questo task. Tutti questi componenti sono stati completamente integrati con i controlli di sicurezza e accesso basati sui ruoli di Iguazio.

Librerie/Framework	Descrizione
MLRun	Gestito da Iguazio per consentire l'assemblaggio, l'esecuzione e il monitoraggio di una pipeline ML/ai.
Nuclio	Un framework di funzioni senza server integrato con Iguazio. Disponibile anche come progetto open-source gestito da Iguazio.
Kubeflow	Un framework basato su Kubernetes per implementare la pipeline. Si tratta anche di un progetto open-source al quale Iguazio contribuisce. È integrato con Iguazio per una maggiore sicurezza e integrazione con il resto dell'infrastruttura.
Docker	Un registro Docker viene eseguito come servizio nella piattaforma Iguazio. È inoltre possibile modificare questa impostazione per connettersi al registro.
NetApp Cloud Volumes	I volumi cloud eseguiti su AWS ci offrono l'accesso a grandi quantità di dati e la possibilità di eseguire copie Snapshot per la versione dei set di dati utilizzati per il training.
Trident	Trident è un progetto open-source gestito da NetApp. Facilita l'integrazione con risorse di storage e calcolo in Kubernetes.

Abbiamo utilizzato diversi notebook per costruire LA pipeline ML. Ogni notebook può essere testato singolarmente prima di essere messo insieme nella pipeline. Ciascun notebook viene descritto singolarmente seguendo il flusso di implementazione di questa applicazione dimostrativa.

Il risultato desiderato è una pipeline che forma un modello basato su una copia Snapshot dei dati e implementa il modello per l'inferenza. Un diagramma a blocchi di una pipeline MLRun completa viene mostrato nell'immagine seguente.



Implementare la funzione di generazione dei dati

In questa sezione viene descritto come abbiamo utilizzato le funzioni senza server di Nuclio per generare i dati dei dispositivi di rete. Il caso di utilizzo viene adattato da un client Iguazio che ha implementato la pipeline e utilizzato i servizi Iguazio per monitorare e prevedere i guasti dei dispositivi di rete.

Abbiamo simulato i dati provenienti dai dispositivi di rete. Esecuzione del notebook Jupyter `data-generator.ipynb` Crea una funzione senza server che viene eseguita ogni 10 minuti e genera un file di parquet con nuovi dati. Per implementare la funzione, eseguire tutte le celle di questo notebook. Vedere "[Sito web di Nuclio](#)" per esaminare eventuali componenti non familiari presenti in questo notebook.

Una cella con il seguente commento viene ignorata durante la generazione della funzione. Si presume che ogni cella del notebook faccia parte della funzione. Importare il modulo Nuclio per attivarlo `%nuclio magic`.

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

Nella specifica della funzione, abbiamo definito l'ambiente in cui viene eseguita la funzione, il modo in cui viene attivata e le risorse che utilizza.

```
spec = nuclio.ConfigSpec(config={"spec.triggers.inference.kind":"cron",
                                "spec.triggers.inference.attributes.interval" : "10m",
                                "spec.readinessTimeoutSeconds" : 60,
                                "spec.minReplicas" : 1},.....
```

Il `init_context` La funzione viene richiamata dal framework Nuclio all'inizializzazione della funzione.

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

Quando la funzione viene inizializzata, viene richiamato qualsiasi codice non presente in una funzione. Quando lo si richiama, viene eseguita una funzione di handler. È possibile modificare il nome del gestore e specificarlo nella specifica della funzione.

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

È possibile verificare la funzione dal notebook prima dell'implementazione.

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

La funzione può essere implementata dal notebook o da una pipeline ci/CD (adattando questo codice).

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

Notebook Pipeline

Questi notebook non devono essere eseguiti singolarmente per questa configurazione. Questa è solo una recensione di ogni notebook. Li abbiamo invocati come parte della pipeline. Per eseguirli singolarmente, consultare la documentazione di MLRun per eseguirli come lavori Kubernetes.

snap_cv.ipynb

Questo notebook gestisce le copie Cloud Volume Snapshot all'inizio della pipeline. Passa il nome del volume al contesto della pipeline. Questo notebook richiama uno script shell per gestire la copia Snapshot. Durante l'esecuzione nella pipeline, il contesto di esecuzione contiene variabili che consentono di individuare tutti i file

necessari per l'esecuzione. Durante la scrittura di questo codice, lo sviluppatore non deve preoccuparsi della posizione del file nel contenitore che lo esegue. Come descritto in seguito, questa applicazione viene implementata con tutte le dipendenze, ed è la definizione dei parametri della pipeline che fornisce il contesto di esecuzione.

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

La posizione della copia Snapshot creata viene inserita nel contesto MLRun per essere utilizzata dalle fasi della pipeline.

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

I tre notebook successivi vengono eseguiti in parallelo.

data-prep.ipynb

Le metriche raw devono essere trasformate in funzionalità per consentire la formazione su modelli. Questo notebook legge le metriche raw dalla directory Snapshot e scrive le funzionalità per il training sui modelli nel volume NetApp.

Quando viene eseguito nel contesto della pipeline, l'input DATA_DIR Contiene la posizione della copia Snapshot.

```
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')))
```

descripse.ipynb

Per visualizzare le metriche in entrata, implementiamo una fase di pipeline che fornisce grafici e grafici disponibili attraverso le interfacce utente Kubeflow e MLRun. Ogni esecuzione dispone di una propria versione di questo tool di visualizzazione.

```
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

Monitoriamo continuamente le metriche alla ricerca di anomalie. Questo notebook crea una funzione senza server che genera le funzionalità necessarie per eseguire la previsione sulle metriche in entrata. Questo notebook richiama la creazione della funzione. Il codice funzione si trova nel notebook data- prep.ipynb. A questo scopo, utilizziamo lo stesso notebook come passaggio della pipeline.

training.ipynb

Dopo aver creato le funzionalità, avviamo il training sul modello. L'output di questa fase è il modello da utilizzare per l'deduzione. Raccogliamo inoltre statistiche per tenere traccia di ogni esecuzione (esperimento).

Ad esempio, il comando seguente inserisce il punteggio di precisione nel contesto dell'esperimento. Questo valore è visibile in Kubeflow e MLRun.

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

deploy-inference-function.ipynb

L'ultima fase della pipeline consiste nell'implementare il modello come funzione senza server per deduzione continua. Questo notebook richiama la creazione della funzione senza server definita in `nuclio-inference-function.ipynb`.

Esaminare e costruire la pipeline

La combinazione di eseguire tutti i notebook in una pipeline consente l'esecuzione continua di esperimenti per rivalutare l'accuratezza del modello rispetto alle nuove metriche. Aprire innanzitutto `pipeline.ipynb` notebook. Ti illustreremo i dettagli che mostrano come NetApp e Iguazio semplificano l'implementazione di questa pipeline ML.

Utilizziamo MLRun per fornire contesto e gestire l'allocazione delle risorse in ogni fase della pipeline. Il servizio API MLRun viene eseguito nella piattaforma Iguazio ed è il punto di interazione con le risorse Kubernetes. Ogni sviluppatore non può richiedere direttamente le risorse; l'API gestisce le richieste e abilita i controlli di accesso.

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

La pipeline può funzionare con volumi cloud NetApp e volumi on-premise. Questa dimostrazione è stata realizzata per utilizzare i volumi cloud, ma è possibile vedere nel codice l'opzione di esecuzione on-premise.

```

# Initialize the NetApp snap function 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.ipyn
b").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]

```

La prima azione necessaria per trasformare un notebook Jupyter in un passo Kubeflow è trasformare il codice in una funzione. Una funzione ha tutte le specifiche richieste per eseguire il notebook. Quando scorri il notebook, puoi vedere che definiamo una funzione per ogni fase della pipeline.

Parte del notebook	Descrizione
<code_to_function> (parte del modulo MLRun)	Nome della funzione: Nome del progetto. utilizzato per organizzare tutti gli artefatti del progetto. Questo è visibile nell'interfaccia utente di MLRun. Gentile. In questo caso, un lavoro Kubernetes. Questo potrebbe essere Dask, mpi, sparkk8s e molto altro ancora. Per ulteriori informazioni, consulta la documentazione di MLRun. File. Il nome del notebook. Questa può anche essere una posizione in Git (HTTP).
immagine	Il nome dell'immagine Docker che stiamo utilizzando per questo passaggio. Abbiamo creato questo documento in precedenza con il notebook create-image.ipynb.
montaggi_volumi e volumi	Dettagli per montare il NetApp Cloud Volume in fase di esecuzione.

Definiamo anche i parametri per le fasi.

```
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
}
```

Una volta definita la funzione per tutti i passaggi, è possibile costruire la pipeline. Utilizziamo il `kfp` per definire questa definizione. La differenza tra l'utilizzo di `MLRun` e la creazione di codice da soli è la semplificazione e la riduzione del codice.

Le funzioni che abbiamo definito vengono trasformate in componenti passo-passo utilizzando `as_step` Funzione di `MLRun`.

Definizione della fase Snapshot

Avviare una funzione Snapshot, eseguire l'output e montare `v3io` come origine:

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

Parametri	Dettagli
NewTask	NewTask è la definizione dell'esecuzione della funzione.
(Modulo MLRun)	Gestore. Nome della funzione Python da richiamare. Abbiamo utilizzato il gestore dei nomi nel notebook, ma non è necessario. parametri. I parametri passati all'esecuzione. All'interno del codice, utilizziamo <code>Context.get_param('PARAMETRO')</code> per ottenere i valori.

Parametri	Dettagli
as_step	Nome. Nome della fase della pipeline Kubeflow. output. Questi sono i valori che la procedura aggiunge al dizionario al completamento. Dai un'occhiata al notebook SNAP_cv.ipynb. mount_v3io(). In questo modo viene configurato il passo per montare /User per l'utente che esegue la pipeline.

```

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)

```

Parametri	Dettagli
input	È possibile passare a un passo gli output di un passo precedente. In questo caso, snap.outputs['snapVolumeDetails'] è il nome della copia Snapshot creata nel passo SNAP.
out_path	Una posizione in cui posizionare gli artefatti che generano utilizzando il modulo MLRun log_Artifacts.

Puoi correre pipeline.ipynb dall'alto verso il basso. È quindi possibile accedere alla scheda Pipeline dalla dashboard di Iguazio per monitorare l'avanzamento, come mostrato nella scheda Pipeline della dashboard di Iguazio.



Poiché abbiamo registrato la precisione delle fasi di training in ogni sessione, abbiamo una registrazione di accuratezza per ogni esperimento, come mostrato nella documentazione relativa alla precisione del training.

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

Se si seleziona la fase Snapshot, è possibile visualizzare il nome della copia Snapshot utilizzata per eseguire questo esperimento.

netops-trainign-pipeline-with-netapp-volume-cloning-rtxdl-2910983943

Artifacts **Input/Output** Volumes Manifest Logs

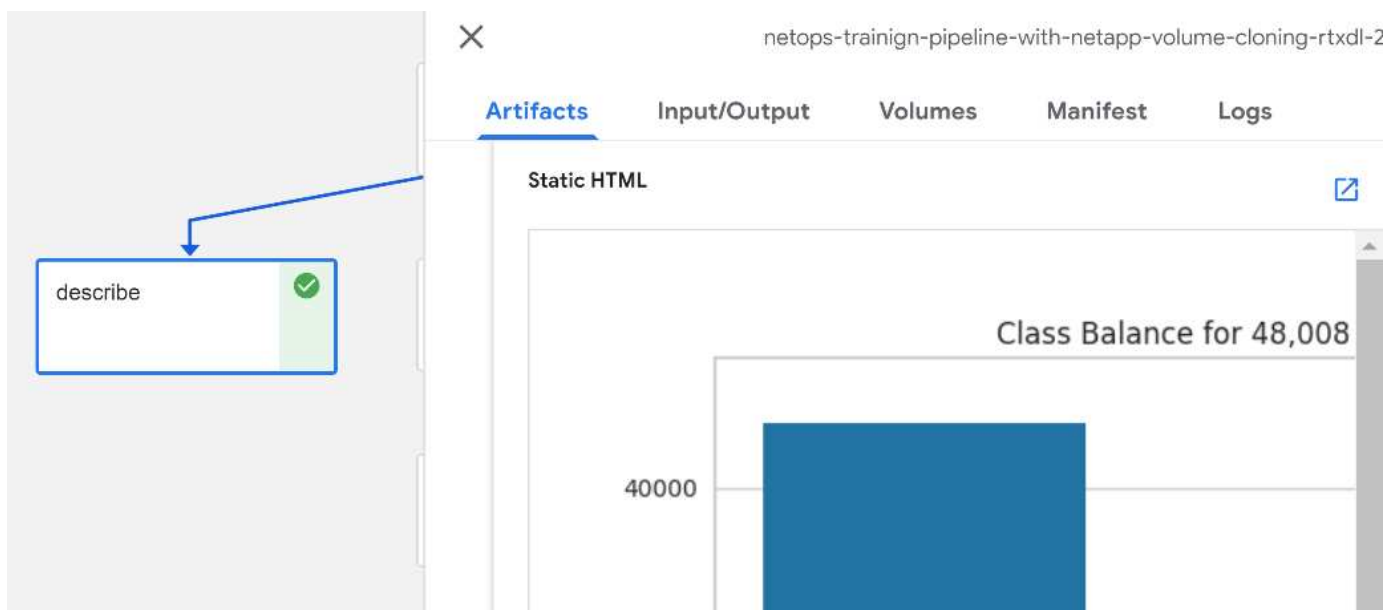
input artifacts

Output parameters

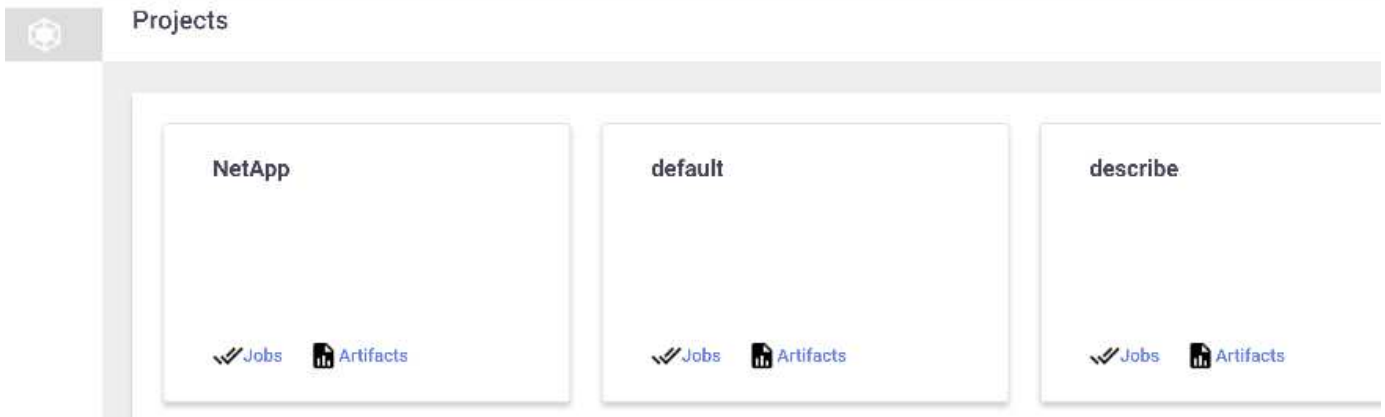
netapp-cloud-volume-snapshot-snapVolumeDetails	/netapp/.snapshot/kfp_20200324_185122
netapp-cloud-volume-snapshot-training_parquet_file	/netapp/.snapshot/kfp_20200324_18512...

Output artifacts

La fase descritta presenta artefatti visivi per esplorare le metriche utilizzate. È possibile espandere per visualizzare il grafico completo come mostrato nell'immagine seguente.



Il database API di MLRun tiene traccia anche di input, output e artefatti per ogni esecuzione organizzata per progetto. Un esempio di input, output e artefatti per ciascuna serigrafia può essere visualizzato nell'immagine seguente.



Per ogni lavoro, memorizziamo ulteriori dettagli.

Name	
deploy-model ● 24 Mar, 14:56:03 ...bcbe38e	
xgb_train ● 24 Mar, 14:53:18 ...5c85949	
data-prep ● 24 Mar, 14:52:46 ...126dc73	
describe ● 24 Mar, 14:52:45 ...c2a460e	describe 24 Mar, 14:52:45 ●
deploy-features-function ● 24 Mar, 14:52:43 ...50d8b83	Info Inputs Artifacts Results Logs
NetApp_Cloud_Volume_Sna 24 Mar, 14:51:22 ...3108eb2	UID 66ef22187efb4ad89e8da8433c2a460e
	Start time 24 Mar, 14:52:45
	Parameters Completed ●
	Results <div> class_label... ▾ key: summary label_colu... ▾ </div>

In questo documento sono disponibili ulteriori informazioni su MLRun. Gli artefatti di ai, inclusa la definizione delle fasi e delle funzioni, possono essere salvati nel database API, con versione e richiamati singolarmente o come progetto completo. I progetti possono anche essere salvati e inviati a Git per un utilizzo successivo. Ti invitiamo a scoprire di più su ["Sito MLRun GitHub"](#).













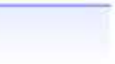




Implementare Grafana Dashboard

Una volta implementato tutto, eseguiamo le inferenze sui nuovi dati. I modelli prevedono guasti sulle apparecchiature di rete. I risultati della previsione vengono memorizzati in una tabella Iguazio TimeSeries. È possibile visualizzare i risultati con Grafana nella piattaforma integrata con la policy di sicurezza e accesso ai dati di Iguazio.

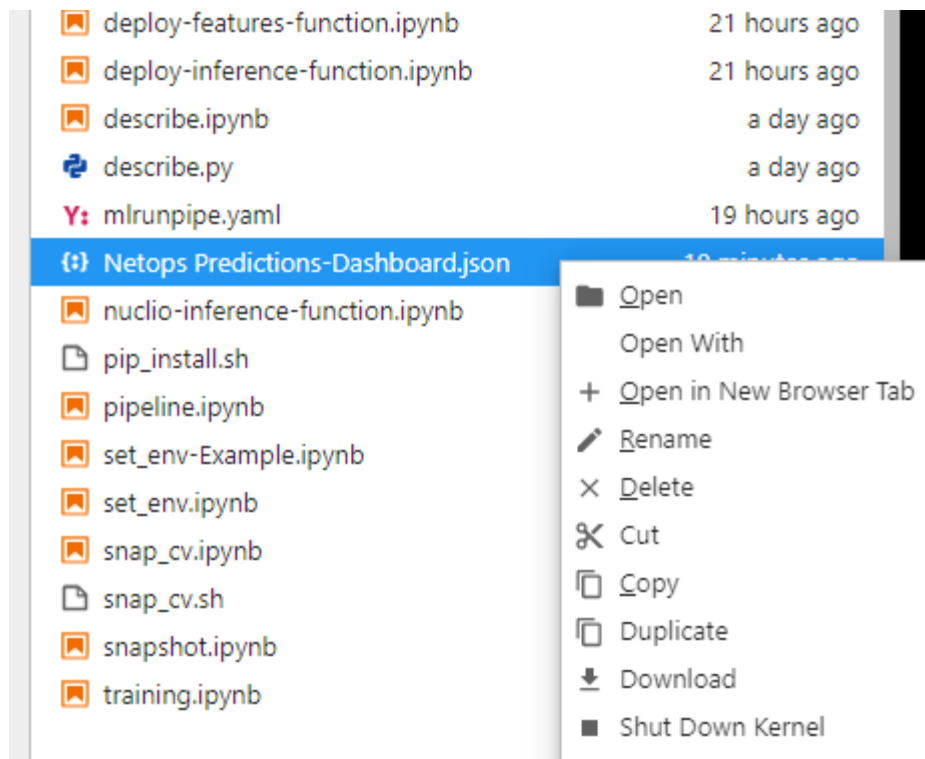
È possibile implementare la dashboard importando il file JSON fornito nelle interfacce Grafana del cluster.

1. Per verificare che il servizio Grafana sia in esecuzione, consultare la sezione servizi.

Services

<input type="checkbox"/>	Name ↑	Running User	Version ↕	CPU (cores)	Memory	AF
<input type="checkbox"/>	 docker-registry Type: Docker Regi		2.7.1	96μ 	1.67 GB 	H
<input type="checkbox"/>	 framesd Type: V3IO Frame		0.6.10	369μ 	795.19 MB 	H
<input type="checkbox"/>	 grafana Type: Grafana		6.6.0	1m 	38.39 MB 	
<input type="checkbox"/>	 jupyter Type: Jupyter Note	admin	1.0.2	81m 	3.27 GB 	
<input type="checkbox"/>	 log-forwarder Type: Log forward		6.7.2	0 	0 bytes 	

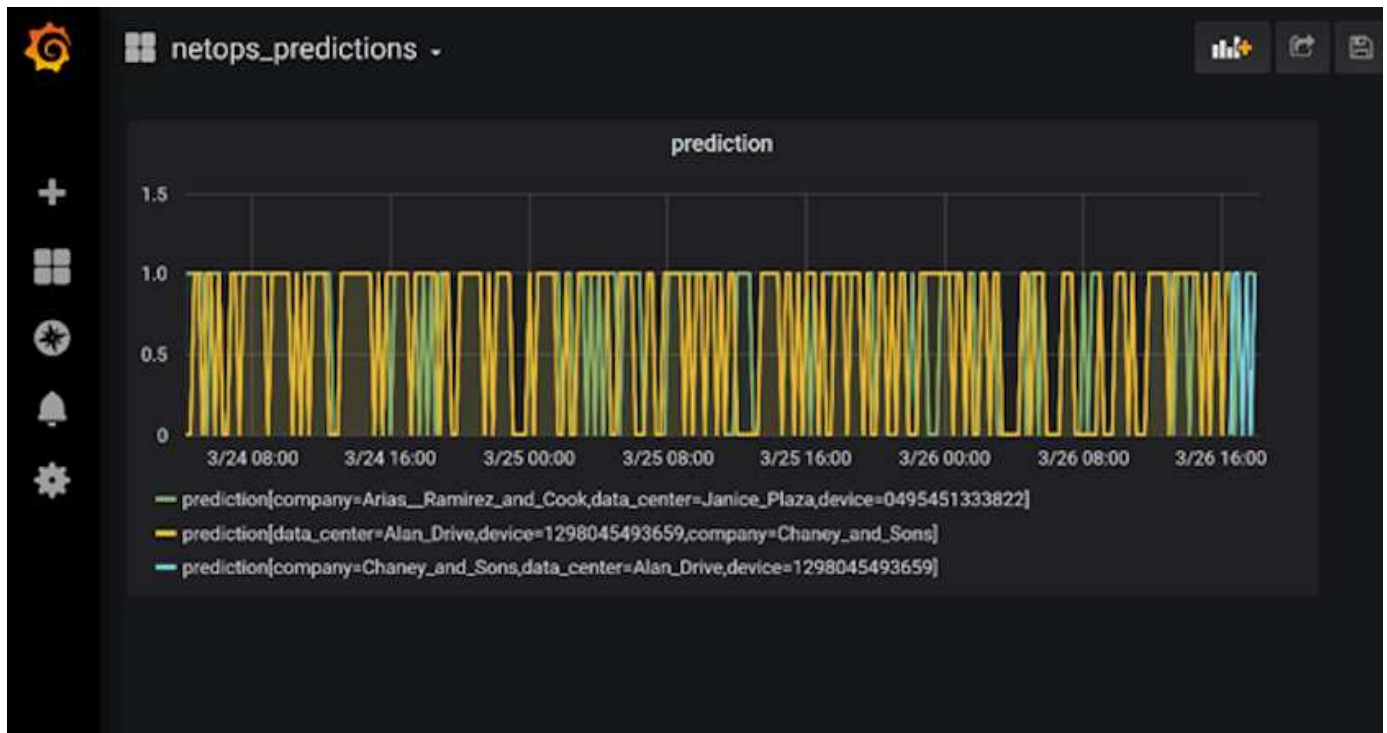
2. Se non è presente, distribuire un'istanza dalla sezione servizi:
 - a. Fare clic su nuovo servizio.
 - b. Selezionare Grafana dall'elenco.
 - c. Accettare le impostazioni predefinite.
 - d. Fare clic su Avanti.
 - e. Inserire l'ID utente.
 - f. Fare clic su Salva servizio.
 - g. Fare clic su Apply Changes (Applica modifiche) nella parte superiore.
3. Per implementare la dashboard, scaricare il file `NetopsPredictions-Dashboard.json` Tramite l'interfaccia Jupyter.



4. Aprire Grafana dalla sezione servizi e importare la dashboard.



5. Fare clic su carica *.json E selezionare il file scaricato in precedenza (NetopsPredictions-Dashboard.json). La dashboard viene visualizzata al termine del caricamento.



Implementare la funzione di pulizia

Quando si generano molti dati, è importante mantenere le cose pulite e organizzate. A tale scopo, implementare la funzione di pulizia con `cleanup.ipynb` notebook.

Benefici

NetApp e Iguazio accelerano e semplificano l'implementazione delle applicazioni ai e ML creando framework essenziali come Kubeflow, Apache Spark e TensorFlow, oltre a tool di orchestrazione come Docker e Kubernetes. Unificando la pipeline di dati end-to-end, NetApp e Iguazio riducono la latenza e la complessità inerenti a molti carichi di lavoro di calcolo avanzati, colmando efficacemente il divario tra sviluppo e operazioni. I data scientist possono eseguire query su set di dati di grandi dimensioni e condividere in modo sicuro dati e modelli algoritmici con utenti autorizzati durante la fase di training. Dopo che i modelli containerizzati sono pronti per la produzione, è possibile spostarli facilmente dagli ambienti di sviluppo agli ambienti operativi.

Informazioni sul copyright

Copyright © 2024 NetApp, Inc. Tutti i diritti riservati. Stampato negli Stati Uniti d'America. Nessuna porzione di questo documento soggetta a copyright può essere riprodotta in qualsiasi formato o mezzo (grafico, elettronico o meccanico, inclusi fotocopie, registrazione, nastri o storage in un sistema elettronico) senza previo consenso scritto da parte del detentore del copyright.

Il software derivato dal materiale sottoposto a copyright di NetApp è soggetto alla seguente licenza e dichiarazione di non responsabilità:

IL PRESENTE SOFTWARE VIENE FORNITO DA NETAPP "COSÌ COM'È" E SENZA QUALSIVOGLIA TIPO DI GARANZIA IMPLICITA O ESPRESSA FRA CUI, A TITOLO ESEMPLIFICATIVO E NON ESAUSTIVO, GARANZIE IMPLICITE DI COMMERCIALIZZABILITÀ E IDONEITÀ PER UNO SCOPO SPECIFICO, CHE VENGONO DECLINATE DAL PRESENTE DOCUMENTO. NETAPP NON VERRÀ CONSIDERATA RESPONSABILE IN ALCUN CASO PER QUALSIVOGLIA DANNO DIRETTO, INDIRETTO, ACCIDENTALE, SPECIALE, ESEMPLARE E CONSEGUENZIALE (COMPRESI, A TITOLO ESEMPLIFICATIVO E NON ESAUSTIVO, PROCUREMENT O SOSTITUZIONE DI MERCI O SERVIZI, IMPOSSIBILITÀ DI UTILIZZO O PERDITA DI DATI O PROFITTI OPPURE INTERRUZIONE DELL'ATTIVITÀ AZIENDALE) CAUSATO IN QUALSIVOGLIA MODO O IN RELAZIONE A QUALUNQUE TEORIA DI RESPONSABILITÀ, SIA ESSA CONTRATTUALE, RIGOROSA O DOVUTA A INSOLVENZA (COMPRESA LA NEGLIGENZA O ALTRO) INSORTA IN QUALSIASI MODO ATTRAVERSO L'UTILIZZO DEL PRESENTE SOFTWARE ANCHE IN PRESENZA DI UN PREAVVISO CIRCA L'EVENTUALITÀ DI QUESTO TIPO DI DANNI.

NetApp si riserva il diritto di modificare in qualsiasi momento qualunque prodotto descritto nel presente documento senza fornire alcun preavviso. NetApp non si assume alcuna responsabilità circa l'utilizzo dei prodotti o materiali descritti nel presente documento, con l'eccezione di quanto concordato espressamente e per iscritto da NetApp. L'utilizzo o l'acquisto del presente prodotto non comporta il rilascio di una licenza nell'ambito di un qualche diritto di brevetto, marchio commerciale o altro diritto di proprietà intellettuale di NetApp.

Il prodotto descritto in questa guida può essere protetto da uno o più brevetti degli Stati Uniti, esteri o in attesa di approvazione.

LEGENDA PER I DIRITTI SOTTOPOSTI A LIMITAZIONE: l'utilizzo, la duplicazione o la divulgazione da parte degli enti governativi sono soggetti alle limitazioni indicate nel sottoparagrafo (b)(3) della clausola Rights in Technical Data and Computer Software del DFARS 252.227-7013 (FEB 2014) e FAR 52.227-19 (DIC 2007).

I dati contenuti nel presente documento riguardano un articolo commerciale (secondo la definizione data in FAR 2.101) e sono di proprietà di NetApp, Inc. Tutti i dati tecnici e il software NetApp forniti secondo i termini del presente Contratto sono articoli aventi natura commerciale, sviluppati con finanziamenti esclusivamente privati. Il governo statunitense ha una licenza irrevocabile limitata, non esclusiva, non trasferibile, non cedibile, mondiale, per l'utilizzo dei Dati esclusivamente in connessione con e a supporto di un contratto governativo statunitense in base al quale i Dati sono distribuiti. Con la sola esclusione di quanto indicato nel presente documento, i Dati non possono essere utilizzati, divulgati, riprodotti, modificati, visualizzati o mostrati senza la previa approvazione scritta di NetApp, Inc. I diritti di licenza del governo degli Stati Uniti per il Dipartimento della Difesa sono limitati ai diritti identificati nella clausola DFARS 252.227-7015(b) (FEB 2014).

Informazioni sul marchio commerciale

NETAPP, il logo NETAPP e i marchi elencati alla pagina <http://www.netapp.com/TM> sono marchi di NetApp, Inc. Gli altri nomi di aziende e prodotti potrebbero essere marchi dei rispettivi proprietari.