



Implementación de la aplicación

NetApp Solutions

NetApp
April 25, 2024

This PDF was generated from https://docs.netapp.com/es-es/netapp-solutions/ai/mlrun_get_code_from_github.html on April 25, 2024. Always check docs.netapp.com for the latest.

Tabla de contenidos

- Implementación de la aplicación 1
 - Obtenga el código de GitHub 1
 - Configurar entorno de trabajo 1
 - Implemente el panel de Grafana 13

Implementación de la aplicación

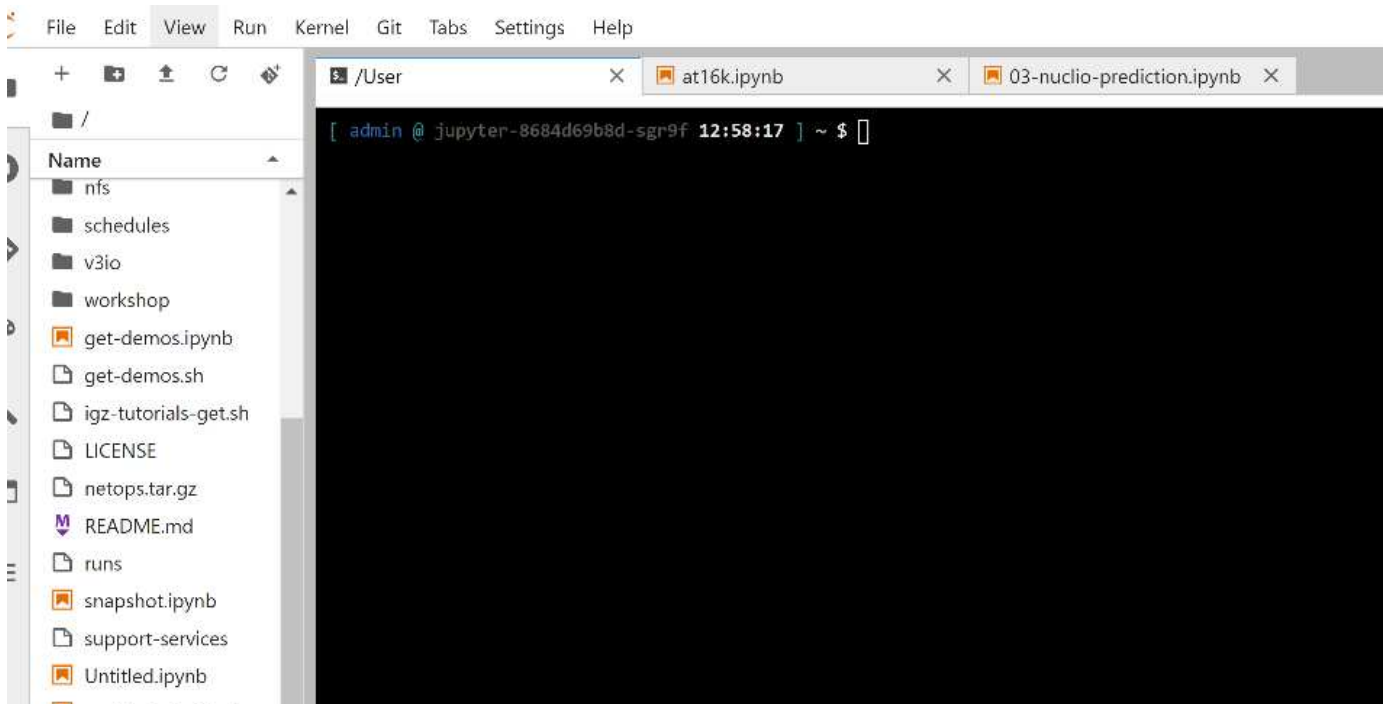
En las siguientes secciones se describe cómo instalar e implementar la aplicación.

Obtenga el código de GitHub

Ahora que el volumen de cloud de NetApp o el volumen de Trident de NetApp están disponibles para el clúster de Iguazio y el entorno de desarrolladores, puede comenzar a revisar la aplicación.

Los usuarios tienen su propio espacio de trabajo (directorio). En cada notebook, la ruta al directorio de usuario es `/User`. La plataforma Iguazio administra el directorio. Si sigue las instrucciones anteriores, el volumen de cloud de NetApp está disponible en `/netapp` directorio.

Obtenga el código de GitHub con una terminal Jupyter.



En el prompt de la terminal de Jupyter, clone el proyecto.

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

Ahora debería ver la `netops- netapp` Carpeta en el árbol de archivos del espacio de trabajo Jupyter.

Configurar entorno de trabajo

Copie el Notebook `set_env-Example.ipynb` como `set_env.ipynb`. Abrir y editar `set_env.ipynb`. Este cuaderno establece variables para las credenciales, ubicaciones

de archivos y controladores de ejecución.

Si sigue las instrucciones anteriores, los pasos siguientes son los únicos cambios que se deben realizar:

1. Obtenga este valor desde el panel de servicios de Iguazio: `docker_registry`

Ejemplo: `docker-registry.default-tenant.app.clusterq.iguazidev.com:80`

2. Cambiar `admin` A su nombre de usuario de Iguazio:

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

A continuación, se muestran los detalles de conexión del sistema ONTAP. Incluya el nombre del volumen que se generó al instalar Trident. La siguiente configuración es para un clúster ONTAP en las instalaciones:

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

La siguiente configuración es para Cloud Volumes ONTAP:

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

Cree imágenes Docker básicas

Todo lo que necesita para construir un ducto ML está incluido en la plataforma Iguazio. El desarrollador puede definir las especificaciones de las imágenes Docker necesarias para ejecutar la canalización y ejecutar la creación de imágenes desde el portátil Jupyter. Abra el portátil `create-images.ipynb` Y ejecute todas las celdas.

Este cuaderno crea dos imágenes que utilizamos en la canalización.

- `iguazio/netapp`. Se utiliza para manejar tareas 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 utilidades para gestionar las copias snapshot de 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())
```

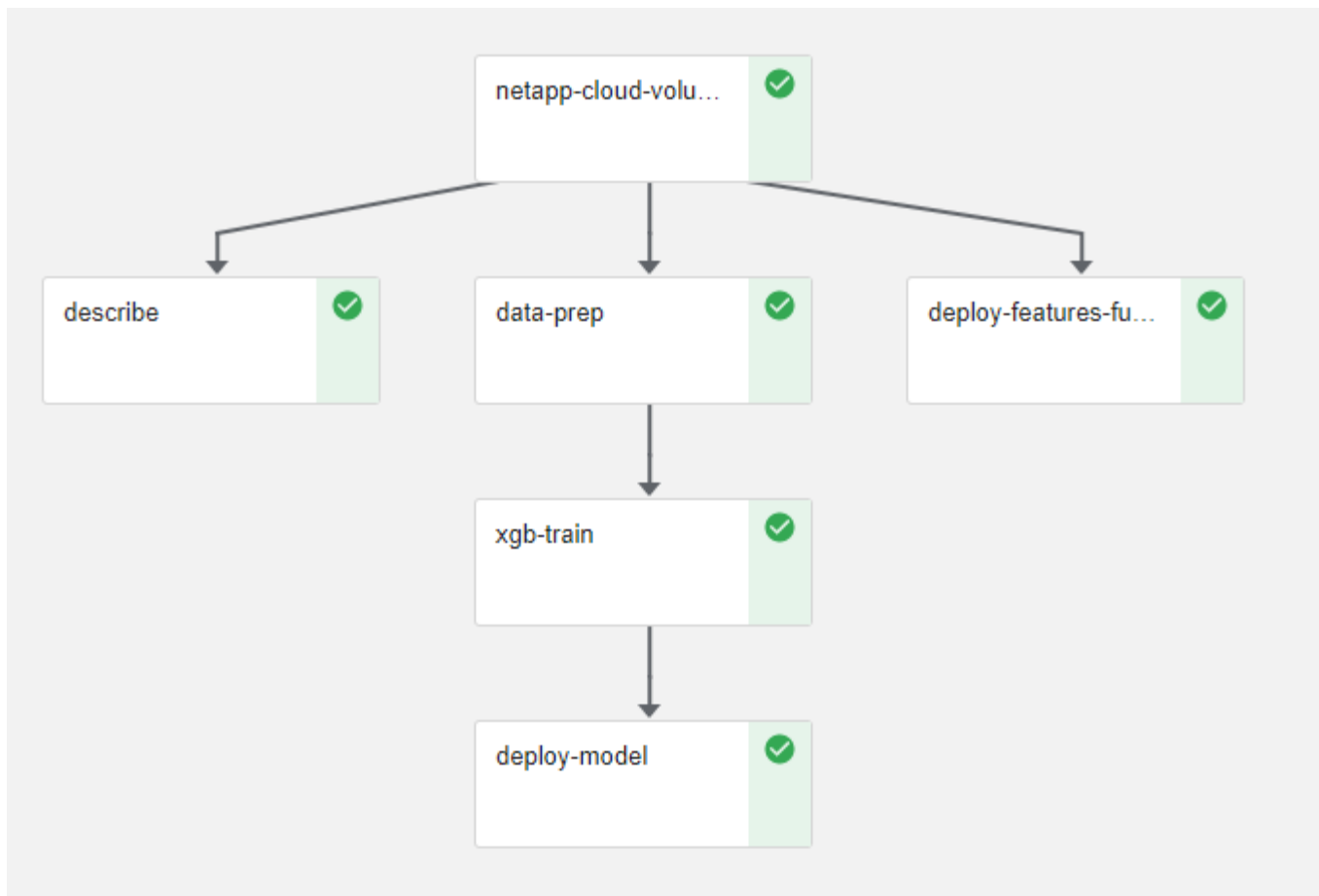
Revise los cuadernos Jupyter individuales

En la siguiente tabla se enumeran las bibliotecas y los marcos que utilizamos para crear esta tarea. Todos estos componentes se han integrado plenamente con los controles de seguridad y acceso basados en funciones de Iguazio.

Bibliotecas/Marco	Descripción
MLRun	Un gestionado por Iguazio para permitir el montaje, ejecución y supervisión de una canalización ML/IA.
Nuclio	Un marco de funciones sin servidor integrado con Iguazio. También disponible como proyecto de código abierto gestionado por Iguazio.
Kubeflow	Un marco basado en Kubernetes para poner en marcha la canalización. Este es también un proyecto de código abierto al que colabora Iguazio. Se integra con Iguazio para una mayor seguridad e integración con el resto de la infraestructura.
Docker	Un registro Docker se ejecuta como servicio en la plataforma Iguazio. También puede cambiar esta opción para conectarse al registro.
Cloud Volumes de NetApp	Cloud Volumes que se ejecuta en AWS nos proporciona acceso a grandes cantidades de datos y la capacidad de realizar copias de Snapshot para versiones de los conjuntos de datos utilizados para el entrenamiento.
Trident	Trident es un proyecto de código abierto gestionado por NetApp. Facilita la integración con los recursos de almacenamiento y computación en Kubernetes.

Hemos utilizado varios portátiles para construir la canalización DE ML. Cada portátil puede probarse individualmente antes de ser reunido en la tubería. Cubrimos cada portátil individualmente tras el flujo de despliegue de esta aplicación de demostración.

El resultado deseado es una canalización que entrena un modelo basado en una copia Snapshot de los datos y pone en marcha el modelo para la inferencia. En la siguiente imagen se muestra un diagrama de bloque de una canalización MLRun completada.



Ponga en marcha la función de generación de datos

En esta sección se describe cómo utilizamos las funciones de Nuclio serverless para generar datos de dispositivos de red. El caso de uso se adapta a partir de un cliente Iguazio que ha implementado la canalización y utiliza servicios Iguazio para supervisar y predecir los fallos del dispositivo de red.

Simulamos los datos procedentes de dispositivos de red. Ejecución del cuaderno Jupyter `data-generator.ipynb` Crea una función sin servidor que se ejecuta cada 10 minutos y genera un archivo Parquet con nuevos datos. Para implementar la función, ejecute todas las celdas de este portátil. Consulte ["Sitio Web de Nuclio"](#) para revisar cualquier componente desconocido en este cuaderno.

Al generar la función, se ignora una celda con el siguiente comentario. Se asume que todas las celdas del portátil forman parte de la función. Importe el módulo Nuclio para activarlo `%nuclio magic`.

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

En la especificación para la función, definimos el entorno en el que se ejecuta la función, cómo se activa y los recursos que consume.

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

La `init_context` La función es invocada por el marco Nuclio tras la inicialización de la función.

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

Cualquier código que no esté en una función se invoca cuando se inicializa la función. Cuando lo invoca, se ejecuta una función de controlador. Puede cambiar el nombre del controlador y especificarlo en la especificación de función.

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

Puede probar la función desde el portátil antes de la implementación.

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

La función puede desplegarse desde el portátil o puede desplegarse a partir de una canalización CI/CD (adaptando este código).

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

Cuadernos de pipeline

Estos cuadernos no están diseñados para ejecutarse individualmente para esta configuración. Esto es sólo una revisión de cada bloc de notas. Los invocamos como parte del proyecto. Para ejecutarlas individualmente, revise la documentación MLRun para ejecutarlas como trabajos de Kubernetes.

snap_cv.ipynb

Este portátil gestiona las copias snapshot de Cloud Volume al principio de la canalización. Pasa el nombre del volumen al contexto de la canalización. Este cuaderno invoca un script de shell para manejar la copia

Snapshot. Mientras se ejecuta en la canalización, el contexto de ejecución contiene variables que ayudan a localizar todos los archivos necesarios para su ejecución. Mientras escribe este código, el desarrollador no tiene que preocuparse por la ubicación del archivo en el contenedor que lo ejecuta. Como se describe más tarde, esta aplicación se implementa con todas sus dependencias y es la definición de los parámetros de canalización que proporciona el contexto de ejecución.

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

La ubicación de la copia Snapshot creada se coloca en el contexto de MLRun que consumirán los pasos de la canalización.

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

Los siguientes tres portátiles se ejecutan en paralelo.

data-prep.ipynb

Las métricas sin formato deben convertirse en funciones para permitir el entrenamiento de modelos. Este cuaderno lee las métricas sin formato del directorio Snapshot y escribe las funciones de entrenamiento de modelos en el volumen de NetApp.

Cuando se ejecuta en el contexto de la canalización, la entrada `DATA_DIR` Contiene la ubicación de la 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')))
```

describa.ipynb

Para visualizar las métricas entrantes, implementamos un paso de canalización que proporciona gráficos y gráficos disponibles a través de las interfaces de usuario de Kubeflow y MLRun. Cada ejecución tiene su propia versión de esta herramienta de visualización.

```
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

Supervisamos continuamente las métricas en busca de anomalías. Este bloc de notas crea una función sin servidor que genera las funciones que necesitan ejecutar la predicción en las métricas entrantes. Este cuaderno invoca la creación de la función. El código de función se encuentra en el portátil `data-prep.ipynb`. Observe que utilizamos el mismo bloc de notas como un paso en la tubería para este propósito.

training.ipynb

Una vez que creamos las funciones, activamos la formación del modelo. El resultado de este paso es el modelo que se va a utilizar para la inferencia. También recopilamos estadísticas para realizar un seguimiento de cada ejecución (experimento).

Por ejemplo, el siguiente comando introduce la puntuación de precisión en el contexto de ese experimento. Este valor es visible en Kubeflow y MLRun.

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

despliegue-inferencia-function.ipynb

El último paso de la canalización es poner en marcha el modelo como una función sin servidor para la inferencia continua. Este cuaderno invoca la creación de la función sin servidor definida en `nuclio-inference-function.ipynb`.

Revisar y crear el canalización

La combinación de ejecutar todos los portátiles en una tubería permite que la ejecución continua de experimentos reevalúe la precisión del modelo con las nuevas métricas. En primer lugar, abra la `pipeline.ipynb` portátil. Le repasamos por los detalles que muestran cómo NetApp y Iguazio simplifican la puesta en marcha de esta canalización DE ML.

Utilizamos MLRun para proporcionar contexto y manejar la asignación de recursos a cada paso de la canalización. El servicio de API MLRun se ejecuta en la plataforma Iguazio y es el punto de interacción con los recursos de Kubernetes. Cada desarrollador no puede solicitar recursos directamente; la API gestiona las solicitudes y habilita los controles de acceso.

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

La canalización puede funcionar con Cloud Volumes de NetApp y los volúmenes en las instalaciones. Hemos creado esta demostración para usar Cloud Volumes, pero puede ver en el código la opción para ejecutarse en las instalaciones.

```
# 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 primera acción necesaria para convertir un cuaderno Jupyter en un paso de Kubeflow es convertir el código en una función. Una función tiene todas las especificaciones necesarias para ejecutar ese portátil. A medida que se desplaza hacia abajo por el bloc de notas, puede ver que definimos una función para cada paso de la canalización.

Parte del portátil	Descripción
<code_to_function> (parte del módulo MLRun)	Nombre de la función: Nombre del proyecto. se utiliza para organizar todos los artefactos del proyecto. Esto es visible en la interfaz de usuario de MLRun. Amable. En este caso, un trabajo de Kubernetes. Esto podría ser DASK, mpi, sparkk8s, y más. Consulte la documentación de MLRun para obtener más detalles. Archivo. El nombre del portátil. También puede ser una ubicación en Git (HTTP).
imagen	El nombre de la imagen de Docker que estamos utilizando para este paso. Lo hemos creado anteriormente con el bloc de notas create-image.ipynb.
montajes_volúmenes y volúmenes	Detalles para montar el Cloud Volume de NetApp en tiempo de ejecución.

También definimos parámetros para los pasos.

```
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
}
```

Después de tener la definición de función para todos los pasos, puede construir la canalización. Utilizamos la `kfp` módulo para realizar esta definición. La diferencia entre el uso de `MLRun` y la construcción por su cuenta es la simplificación y el acortamiento de la codificación.

Las funciones definidas se convierten en componentes de pasos mediante el `as_step` Función de `MLRun`.

Definición de paso de instantánea

Inicie una función `Snapshot`, la salida y el montaje `v3io` como fuente:

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

Parámetros	Detalles
Nueva tarea	Newtask es la definición de la ejecución de la función.
(Módulo <code>MLRun</code>)	Manipulador. Nombre de la función Python que se va a invocar. Utilizamos el controlador de nombres en el portátil, pero no es necesario. parámetros. Los parámetros que pasamos a la ejecución. Dentro de nuestro código, utilizamos <code>context.get_param</code> («PARAMETER») para obtener los valores.

Parámetros	Detalles
paso_as	Nombre. Nombre del paso de la canalización de Kubeflow. salidas. Estos son los valores que el paso agrega al diccionario al terminar. Eche un vistazo al portátil SNAP_cv.ipynb. mount_v3io(). Esto configura el paso para montar /User para el usuario que ejecuta la canalización.

```

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)

```

Parámetros	Detalles
entradas	Puede pasar a un paso las salidas de un paso anterior. En este caso, snap.outputs[snapVolumeDetails] es el nombre de la copia Snapshot que creamos en el paso snap.
ruta de salida	Ubicación para colocar artefactos que generan utilizando el módulo MLRun log_Artifacts.

Puede ejecutar `pipeline.ipynb` de arriba a abajo. A continuación, puede ir a la pestaña tuberías desde el panel de control de Iguazio para supervisar el progreso tal y como se ve en la pestaña tuberías del panel de control de Iguazio.



Debido a que hemos registrado la precisión del paso de entrenamiento en cada carrera, tenemos un registro de precisión para cada experimento, como se ve en el registro de precisión de entrenamiento.

<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

Si selecciona el paso Snapshot, puede ver el nombre de la copia Snapshot que se utilizó para ejecutar este experimento.

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

El paso descrito tiene artefactos visuales para explorar las métricas que utilizamos. Puede expandir para ver el trazado completo como se ve en la siguiente imagen.

netops-trainign-pipeline-with-netapp-volume-cloning-rtxdl-2

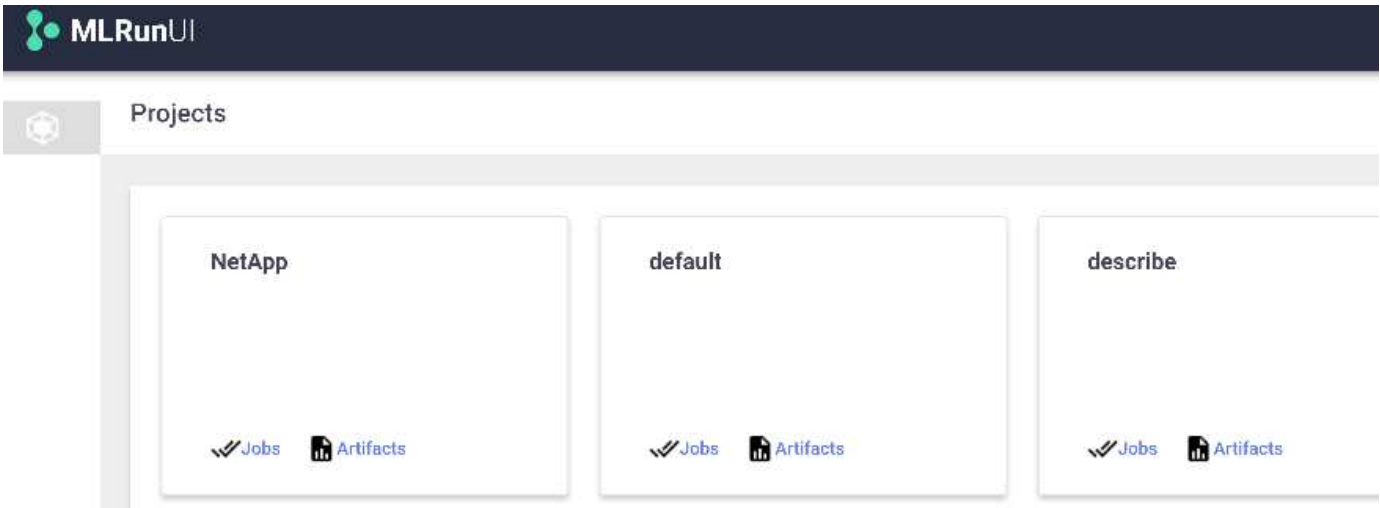
Artifacts **Input/Output** Volumes Manifest Logs

Static HTML

Class Balance for 48,008

40000

La base de datos de la API MLRun también realiza un seguimiento de las entradas, salidas y artefactos de cada ejecución organizada por el proyecto. En la siguiente imagen se puede ver un ejemplo de entradas, salidas y artefactos para cada secuencia.



Para cada trabajo, almacenamos detalles adicionales.

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

deploy-features-function ●
24 Mar, 14:52:43 ...50d8b83

NetApp_Cloud_Volume_Sna ●
24 Mar, 14:51:22 ...3108eb2

describe

24 Mar, 14:52:45 ●

Info

Inputs

Artifacts

Results

Logs

UID

66ef22187efb4ad89e8da8433c2a460e

Start time

24 Mar, 14:52:45

Parameters

Completed ●

Results

class_label... ▾

key: summary

label_colu... ▾

Hay más información sobre MLRun que podemos cubrir en este documento. Los artefactos de al, incluida la definición de los pasos y las funciones, se pueden guardar en la base de datos de API, con versiones e invocados individualmente o como un proyecto completo. Los proyectos también se pueden guardar e insertar en Git para su uso posterior. Le animamos a obtener más información en la ["Sitio de MLRun GitHub"](#).

Implemente el panel de Grafana















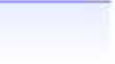




Una vez que todo se pone en marcha, ejecutamos inferencias sobre nuevos datos. Los modelos predicen fallos en el equipo de dispositivo de red. Los resultados de la predicción se almacenan en una tabla de timbres de Iguazio. Puede visualizar los resultados con Grafana en la plataforma integrada con la política de acceso a datos y seguridad de Iguazio.

Puede implementar la consola importando el archivo JSON proporcionado en las interfaces de Grafana en el

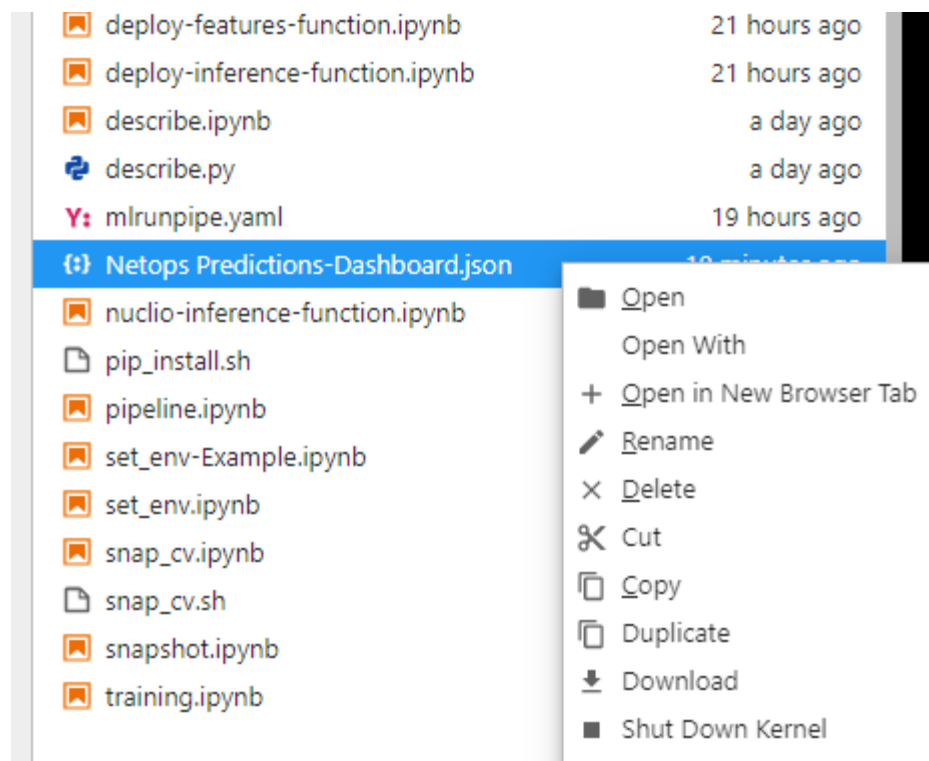
clúster.

1. Para verificar que el servicio Grafana se está ejecutando, busque Servicios.

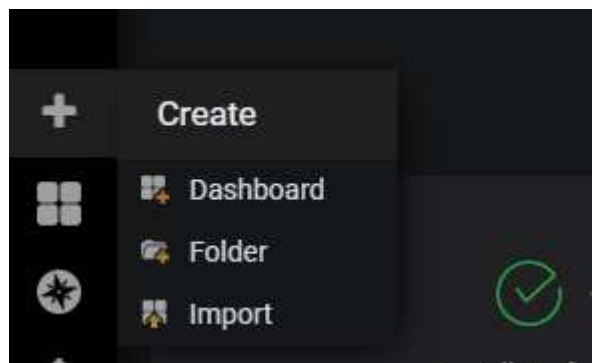
Services

<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 	

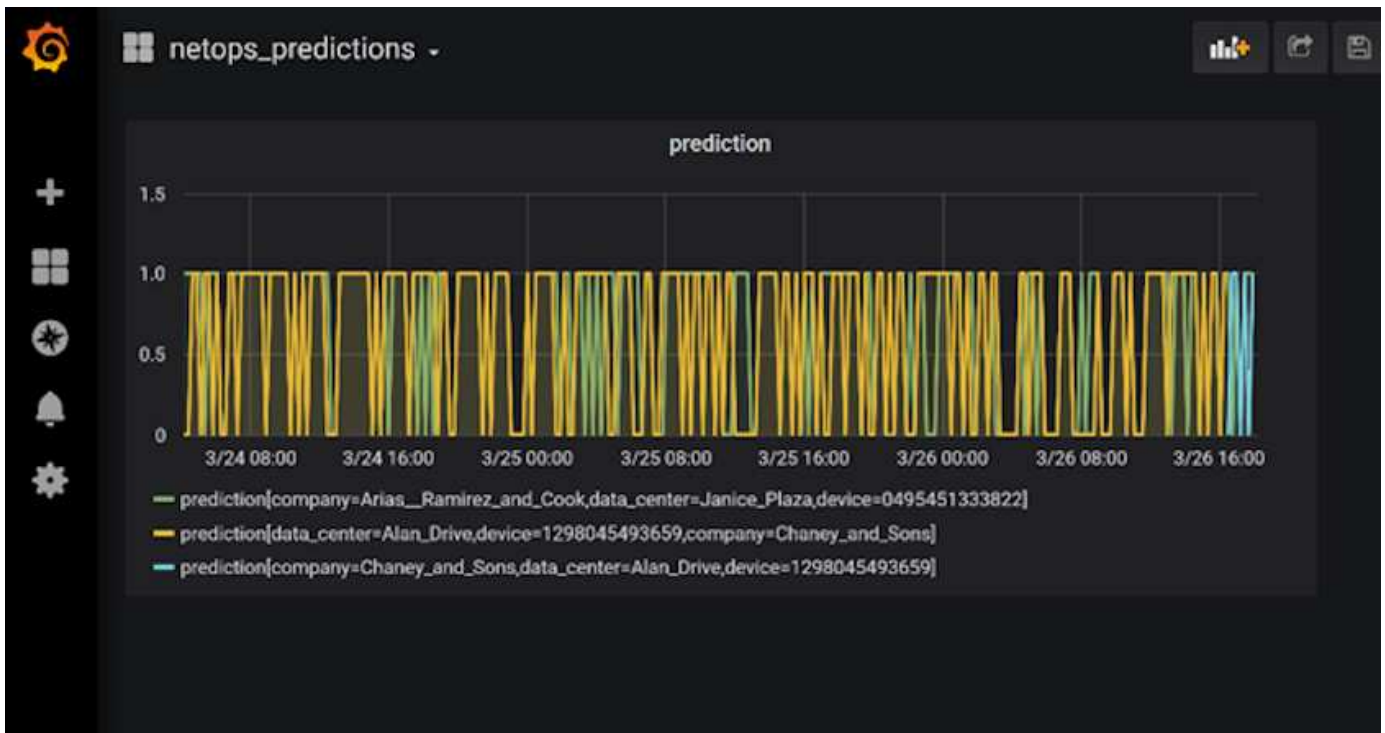
2. Si no está presente, implemente una instancia en la sección Servicios:
 - a. Haga clic en Nuevo servicio.
 - b. Seleccione Grafana de la lista.
 - c. Acepte los valores predeterminados.
 - d. Haga clic en Siguiente paso.
 - e. Introduzca su ID de usuario.
 - f. Haga clic en Guardar servicio.
 - g. Haga clic en aplicar cambios en la parte superior.
3. Para implementar el panel de control, descargue el archivo `NetopsPredictions-Dashboard.json` A través de la interfaz Jupyter.



4. Abra Grafana en la sección Servicios e importe el panel de control.



5. Haga clic en Upload *.json File (Archivo) y seleccione el archivo que descargó anteriormente (NetopsPredictions-Dashboard.json). El panel se muestra una vez finalizada la carga.



Despliegue la función de limpieza

Cuando genera una gran cantidad de datos, es importante mantener las cosas limpias y organizadas. Para ello, implemente la función de limpieza con `cleanup.ipynb` portátil.

Beneficios

NetApp y Iguazio aceleran y simplifican la puesta en marcha de aplicaciones de IA y ML mediante la creación de marcos esenciales como Kubeflow, Apache Spark y TensorFlow, junto con herramientas de orquestación como Docker y Kubernetes. Al unificar la canalización de datos completa, NetApp y Iguazio reducen la latencia y la complejidad inherentes a muchas cargas de trabajo informáticas avanzadas, y esta brecha entre el desarrollo y las operaciones. Los científicos de datos pueden ejecutar consultas en grandes conjuntos de datos y compartir de forma segura datos y modelos algorítmicos con usuarios autorizados durante la fase de entrenamiento. Después de que los modelos en contenedores están listos para la producción, puede moverlos fácilmente desde entornos de desarrollo a entornos operativos.

Información de copyright

Copyright © 2024 NetApp, Inc. Todos los derechos reservados. Imprimido en EE. UU. No se puede reproducir este documento protegido por copyright ni parte del mismo de ninguna forma ni por ningún medio (gráfico, electrónico o mecánico, incluidas fotocopias, grabaciones o almacenamiento en un sistema de recuperación electrónico) sin la autorización previa y por escrito del propietario del copyright.

El software derivado del material de NetApp con copyright está sujeto a la siguiente licencia y exención de responsabilidad:

ESTE SOFTWARE LO PROPORCIONA NETAPP «TAL CUAL» Y SIN NINGUNA GARANTÍA EXPRESA O IMPLÍCITA, INCLUYENDO, SIN LIMITAR, LAS GARANTÍAS IMPLÍCITAS DE COMERCIALIZACIÓN O IDONEIDAD PARA UN FIN CONCRETO, CUYA RESPONSABILIDAD QUEDA EXIMIDA POR EL PRESENTE DOCUMENTO. EN NINGÚN CASO NETAPP SERÁ RESPONSABLE DE NINGÚN DAÑO DIRECTO, INDIRECTO, ESPECIAL, EJEMPLAR O RESULTANTE (INCLUYENDO, ENTRE OTROS, LA OBTENCIÓN DE BIENES O SERVICIOS SUSTITUTIVOS, PÉRDIDA DE USO, DE DATOS O DE BENEFICIOS, O INTERRUPCIÓN DE LA ACTIVIDAD EMPRESARIAL) CUALQUIERA SEA EL MODO EN EL QUE SE PRODUJERON Y LA TEORÍA DE RESPONSABILIDAD QUE SE APLIQUE, YA SEA EN CONTRATO, RESPONSABILIDAD OBJETIVA O AGRAVIO (INCLUIDA LA NEGLIGENCIA U OTRO TIPO), QUE SURJAN DE ALGÚN MODO DEL USO DE ESTE SOFTWARE, INCLUSO SI HUBIEREN SIDO ADVERTIDOS DE LA POSIBILIDAD DE TALES DAÑOS.

NetApp se reserva el derecho de modificar cualquiera de los productos aquí descritos en cualquier momento y sin aviso previo. NetApp no asume ningún tipo de responsabilidad que surja del uso de los productos aquí descritos, excepto aquello expresamente acordado por escrito por parte de NetApp. El uso o adquisición de este producto no lleva implícita ninguna licencia con derechos de patente, de marcas comerciales o cualquier otro derecho de propiedad intelectual de NetApp.

Es posible que el producto que se describe en este manual esté protegido por una o más patentes de EE. UU., patentes extranjeras o solicitudes pendientes.

LEYENDA DE DERECHOS LIMITADOS: el uso, la copia o la divulgación por parte del gobierno están sujetos a las restricciones establecidas en el subpárrafo (b)(3) de los derechos de datos técnicos y productos no comerciales de DFARS 252.227-7013 (FEB de 2014) y FAR 52.227-19 (DIC de 2007).

Los datos aquí contenidos pertenecen a un producto comercial o servicio comercial (como se define en FAR 2.101) y son propiedad de NetApp, Inc. Todos los datos técnicos y el software informático de NetApp que se proporcionan en este Acuerdo tienen una naturaleza comercial y se han desarrollado exclusivamente con fondos privados. El Gobierno de EE. UU. tiene una licencia limitada, irrevocable, no exclusiva, no transferible, no sublicenciable y de alcance mundial para utilizar los Datos en relación con el contrato del Gobierno de los Estados Unidos bajo el cual se proporcionaron los Datos. Excepto que aquí se disponga lo contrario, los Datos no se pueden utilizar, desvelar, reproducir, modificar, interpretar o mostrar sin la previa aprobación por escrito de NetApp, Inc. Los derechos de licencia del Gobierno de los Estados Unidos de América y su Departamento de Defensa se limitan a los derechos identificados en la cláusula 252.227-7015(b) de la sección DFARS (FEB de 2014).

Información de la marca comercial

NETAPP, el logotipo de NETAPP y las marcas que constan en <http://www.netapp.com/TM> son marcas comerciales de NetApp, Inc. El resto de nombres de empresa y de producto pueden ser marcas comerciales de sus respectivos propietarios.