Ai responsabile e deduzione riservata - NetApp ai con Protopia Image Transformation

NetApp Solutions

NetApp
May 14, 2024

Sommario

Ai responsabile e deduzione riservata - NetApp ai con Protopia Image Transformation

TR-4928: Ai responsabile e deduzione riservata - NetApp ai con Protopia Image e Data Transformation

Sathish Thyagarajan, Michael Oglesby, NetApp Byung Hoon Ahn, Jennifer Cwagenberg, Protopia

Le interpretazioni visive sono diventate parte integrante della comunicazione con l'emergere dell'acquisizione e dell'elaborazione delle immagini. L'intelligenza artificiale (ai) nell'elaborazione di immagini digitali offre nuove opportunità di business, come nel campo medico per l'identificazione di tumori e altre malattie, nell'analisi visiva geospaziale per lo studio dei rischi ambientali, nel riconoscimento dei modelli, nell'elaborazione video per la lotta alla criminalità e così via. Tuttavia, questa opportunità comporta anche responsabilità straordinarie.

Più le organizzazioni prendono decisioni in mano all'ai, più accettano rischi legati alla privacy e alla sicurezza dei dati e a questioni legali, etiche e normative. L'intelligenza artificiale responsabile consente a aziende e organizzazioni governative di creare fiducia e governance che sono fondamentali per l'intelligenza artificiale su larga scala nelle grandi imprese. Questo documento descrive una soluzione di inferenza ai convalidata da NetApp in tre scenari diversi utilizzando le tecnologie di gestione dei dati NetApp con il software di offuscamento dei dati Protopia per privatizzare i dati sensibili e ridurre rischi e preoccupazioni etiche.

Milioni di immagini vengono generate ogni giorno con diversi dispositivi digitali sia da consumatori che da aziende. La conseguente massiccia esplosione dei dati e del carico di lavoro di calcolo fa sì che le aziende si rivolgono alle piattaforme di cloud computing per scalabilità ed efficienza. Nel frattempo, i problemi di privacy relativi alle informazioni sensibili contenute nei dati delle immagini sorgono con il trasferimento a un cloud pubblico. La mancanza di garanzie di sicurezza e privacy diventa la principale barriera all'implementazione dei sistemi ai di elaborazione delle immagini.

Inoltre, è disponibile la "diritto di cancellazione" Dal GDPR, il diritto di un individuo di richiedere che un'organizzazione cancella tutti i propri dati personali. C'è anche il "Legge sulla privacy", che stabilisce un codice di pratiche di informazione eque. Le immagini digitali come le fotografie possono costituire dati personali ai sensi del GDPR, che regola le modalità di raccolta, elaborazione e cancellazione dei dati. In caso contrario, la mancata conformità al GDPR potrebbe comportare multe elevate per la violazione delle conformità che possono danneggiare seriamente le organizzazioni. I principi di privacy sono tra le fondamenta dell'implementazione dell'ai responsabile che garantisce l'equità nelle previsioni del modello di apprendimento automatico (ML) e di apprendimento approfondito (DL) e riduce i rischi associati alla violazione della privacy o della conformità alle normative.

Questo documento descrive una soluzione di progettazione validata in tre scenari diversi, con e senza offuscamento delle immagini, per preservare la privacy e implementare una soluzione di ai responsabile:

- **Scenario 1.** deduzione on-demand nel notebook Jupyter.
- **Scenario 2.** deduzione batch su Kubernetes.
- **Scenario 3.** Server di inferenza NVIDIA Triton.

Per questa soluzione, utilizziamo Face Detection Data Set and Benchmark (FDDB), un set di dati delle regioni dei volti progettato per studiare il problema del rilevamento dei volti senza vincoli, in combinazione con il

framework di apprendimento automatico PyTorch per l'implementazione di FaceBoxes. Questo set di dati contiene le annotazioni per 5171 volti in un set di 2845 immagini di varie risoluzioni. Inoltre, questo report tecnico presenta alcune delle aree di soluzione e i casi di utilizzo rilevanti raccolti dai clienti NetApp e dai tecnici sul campo nelle situazioni in cui questa soluzione è applicabile.

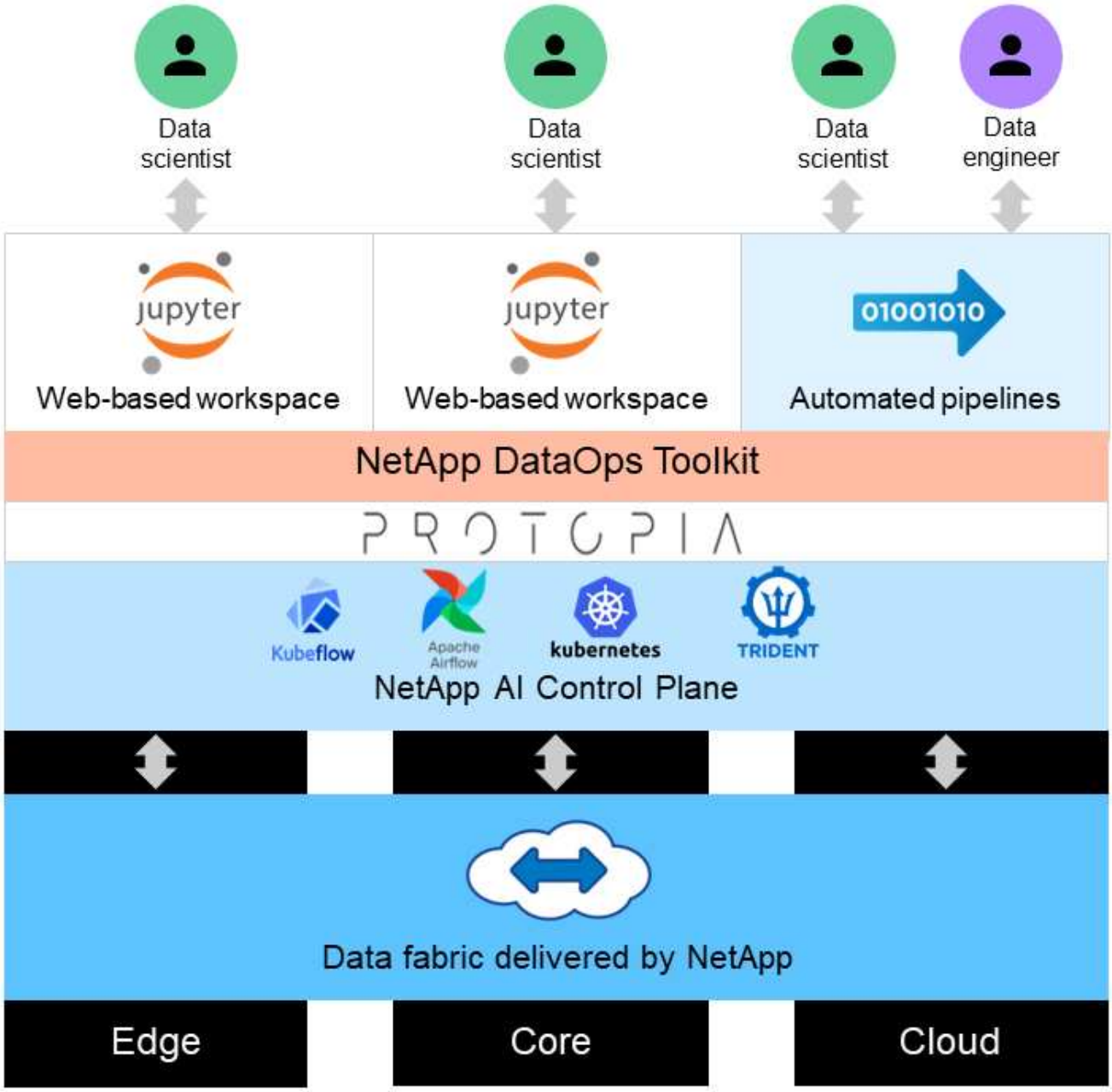
Pubblico di riferimento

Questo report tecnico è destinato ai seguenti destinatari:

- Business leader e architetti aziendali che desiderano progettare e implementare un'ai responsabile e affrontare i problemi di protezione dei dati e privacy relativi all'elaborazione delle immagini facciali negli spazi pubblici.
- Data scientist, data engineer, ricercatori ai/ machine learning (ML) e sviluppatori di sistemi ai/ML che mirano a proteggere e preservare la privacy.
- Architetti aziendali che progettano soluzioni di offuscamento dei dati per modelli e applicazioni ai/ML conformi agli standard normativi come GDPR, CCPA o il Privacy Act del Department of Defense (DoD) e organizzazioni governative.
- Data scientist e ingegneri ai alla ricerca di modi efficienti per implementare modelli di deduzione ai/ML/DL e deep learning (DL) in grado di proteggere le informazioni sensibili.
- Edge Device Manager e amministratori degli edge server responsabili dell'implementazione e della gestione dei modelli di inferenza edge.

Architettura della soluzione

Questa soluzione è progettata per gestire carichi di lavoro ai di deduzione in batch e in tempo reale su grandi set di dati utilizzando la potenza di elaborazione delle GPU insieme alle CPU tradizionali. Questa convalida dimostra l'inferenza di conservazione della privacy per ML e la gestione ottimale dei dati richiesta per le organizzazioni che cercano implementazioni ai responsabili. Questa soluzione offre un'architettura adatta per una piattaforma Kubernetes a nodo singolo o multiplo per il cloud computing e l'edge interconnesso con NetApp ONTAP ai al core on-premise, il toolkit NetApp DataOps e il software di offuscamento Protopia utilizzando le interfacce Jupyter Lab e CLI. La figura seguente mostra una panoramica dell'architettura logica del data fabric basato su NetApp con DataOps Toolkit e Protopia.



Il software di offuscamento Protopia viene eseguito perfettamente sul NetApp DataOps Toolkit e trasforma i dati prima di lasciare il server di storage.

Aree di soluzione

L'elaborazione delle immagini digitali offre numerosi vantaggi, consentendo a molte organizzazioni di sfruttare al meglio i dati associati alle rappresentazioni visive. Questa soluzione NetApp e Protopia offre un design di inferenza ai unico per proteggere e privatizzare i dati ai/ML nel ciclo di vita ML/DL. Consente ai clienti di mantenere la proprietà dei dati sensibili, utilizzare modelli di implementazione del cloud pubblico o ibrido per scalabilità ed efficienza, alleviando i problemi relativi alla privacy e implementando l'inferenza ai ai ai ai edge.

Intelligence ambientale

Esistono diversi modi in cui i settori possono sfruttare le analisi geospaziali nelle aree dei rischi ambientali. I governi e il dipartimento delle opere pubbliche possono trarre utili informazioni sulla salute pubblica e sulle condizioni meteorologiche per consigliare meglio il pubblico durante una pandemia o un disastro naturale come gli incendi. Ad esempio, è possibile identificare un paziente COVID-positivo in spazi pubblici, come aeroporti o ospedali, senza compromettere la privacy della persona interessata e avvisare le rispettive autorità e il pubblico nelle vicinanze per le misure di sicurezza necessarie.

Dispositivi indossabili edge

Nel settore militare e nei campi di battaglia, è possibile utilizzare l'inferenza ai all'edge come dispositivi indossabili per monitorare la salute dei soldati, monitorare il comportamento dei conducenti e avvisare le autorità sulla sicurezza e i rischi associati all'avvicinamento ai veicoli militari, preservando e proteggendo la privacy dei soldati. Il futuro dei militari sta diventando high-tech con Internet of Battlefield Things (IoBT) e Internet of Military Things (IoMT) per attrezzature da combattimento indossabili che aiutano i soldati a identificare i nemici e a migliorare le performance in battaglia utilizzando il calcolo rapido edge. La protezione e la conservazione dei dati visivi raccolti da dispositivi edge come droni e dispositivi indossabili è fondamentale per tenere a bada hacker e nemici.

Operazioni di evacuazione non combattente

Le operazioni di evacuazione dei non combattenti (NEO) sono condotte dal DOD per aiutare a evacuare cittadini e cittadini degli Stati Uniti, personale civile del DOD e persone designate (nazione ospitante (HN) e cittadini di paesi terzi (TCN)) la cui vita è in pericolo per un adeguato rifugio sicuro. I controlli amministrativi in uso utilizzano processi di screening per l'evacuazione in gran parte manuali. Tuttavia, l'accuratezza, la sicurezza e la velocità dell'identificazione degli evacuati, del monitoraggio degli evacuati e dello screening delle minacce potrebbero potenzialmente essere migliorate utilizzando strumenti ai/ML altamente automatizzati combinati con tecnologie di offuscamento video ai/ML.

Ricerca sanitaria e biomedica

L'elaborazione delle immagini viene utilizzata per diagnosticare patologie per la pianificazione chirurgica da immagini 3D ottenute dalla tomografia computerizzata (TC) o dall'imaging a risonanza magnetica (MRI). Le norme sulla privacy di HIPAA regolano le modalità di raccolta, elaborazione e cancellazione dei dati da parte delle organizzazioni per tutte le informazioni personali e le immagini digitali come le fotografie. Affinché i dati possano qualificarsi come condivisibili in base alle normative HIPAA Safe Harbor, è necessario rimuovere le immagini fotografiche a pieno facciale e le immagini comparabili. Tecniche automatizzate come la deidentificazione o il cranio-gli algoritmi di spelatura utilizzati per oscurare le caratteristiche facciali di un individuo dalle immagini strutturali TC/RM sono diventati una parte essenziale del processo di condivisione dei dati per gli istituti di ricerca biomedica.

Migrazione nel cloud degli analytics ai/ML

I clienti aziendali hanno tradizionalmente addestrato e implementato modelli ai/ML on-premise. Per motivi di efficienza e scalabilità, questi clienti si stanno espandendo per spostare le funzioni ai/ML in implementazioni di cloud pubblico, ibrido o multi-cloud. Tuttavia, sono vincolati da quali dati possono essere esposti ad altre infrastrutture. Le soluzioni NetApp affrontano una gamma completa di minacce alla cybersicurezza richieste per "protezione dei dati" E la valutazione della sicurezza e, se combinata con la trasformazione dei dati di Protopia, riducono al minimo i rischi associati alla migrazione dei carichi di lavoro ai/ML di elaborazione delle immagini nel cloud.

Per ulteriori casi di utilizzo per l'edge computing e l'inferenza ai in altri settori, vedere "TR-4886 ai Inferencing at the Edge" E il blog NetApp ai, "Intelligence e privacy".

Panoramica della tecnologia

Questa sezione fornisce una panoramica dei vari componenti tecnici necessari per completare questa soluzione.

Protopia

Protopia ai offre una soluzione software-only senza invadenze per l'inferenza riservata nel mercato odierno. La soluzione Protopia offre una protezione senza pari per i servizi di inferenza riducendo al minimo l'esposizione delle informazioni sensibili. L'intelligenza artificiale viene alimentata solo nelle informazioni contenute nel record di dati che sono veramente essenziali per eseguire l'attività in corso e niente di più. La maggior parte delle attività di inferenza non utilizza tutte le informazioni presenti in ogni record di dati. Indipendentemente dal fatto che l'ai stia utilizzando immagini, voce, video o persino dati tabulari strutturati, Protopia offre solo ciò di cui ha bisogno il servizio di inferenza. La tecnologia brevettata core utilizza il rumore matematicamente curato per trasformare in modo stocoso i dati e raccogliere le informazioni non necessarie per un determinato servizio ML. Questa soluzione non maschera i dati, ma modifica la rappresentazione dei dati utilizzando un rumore casuale a cura.

La soluzione Protopia formula il problema di modificare la rappresentazione come metodo di massimizzazione delle perturbazioni basato su gradiente che conserva ancora le informazioni pertinenti nello spazio delle funzionalità di input rispetto alla funzionalità del modello. Questo processo di rilevamento viene eseguito come un passo di messa a punto al termine della formazione sul modello ML. Dopo che il pass genera automaticamente un set di distribuzioni di probabilità, una trasformazione dei dati a basso overhead applica campioni di rumore da queste distribuzioni ai dati, offuscandoli prima di passarli al modello per l'inferenza.

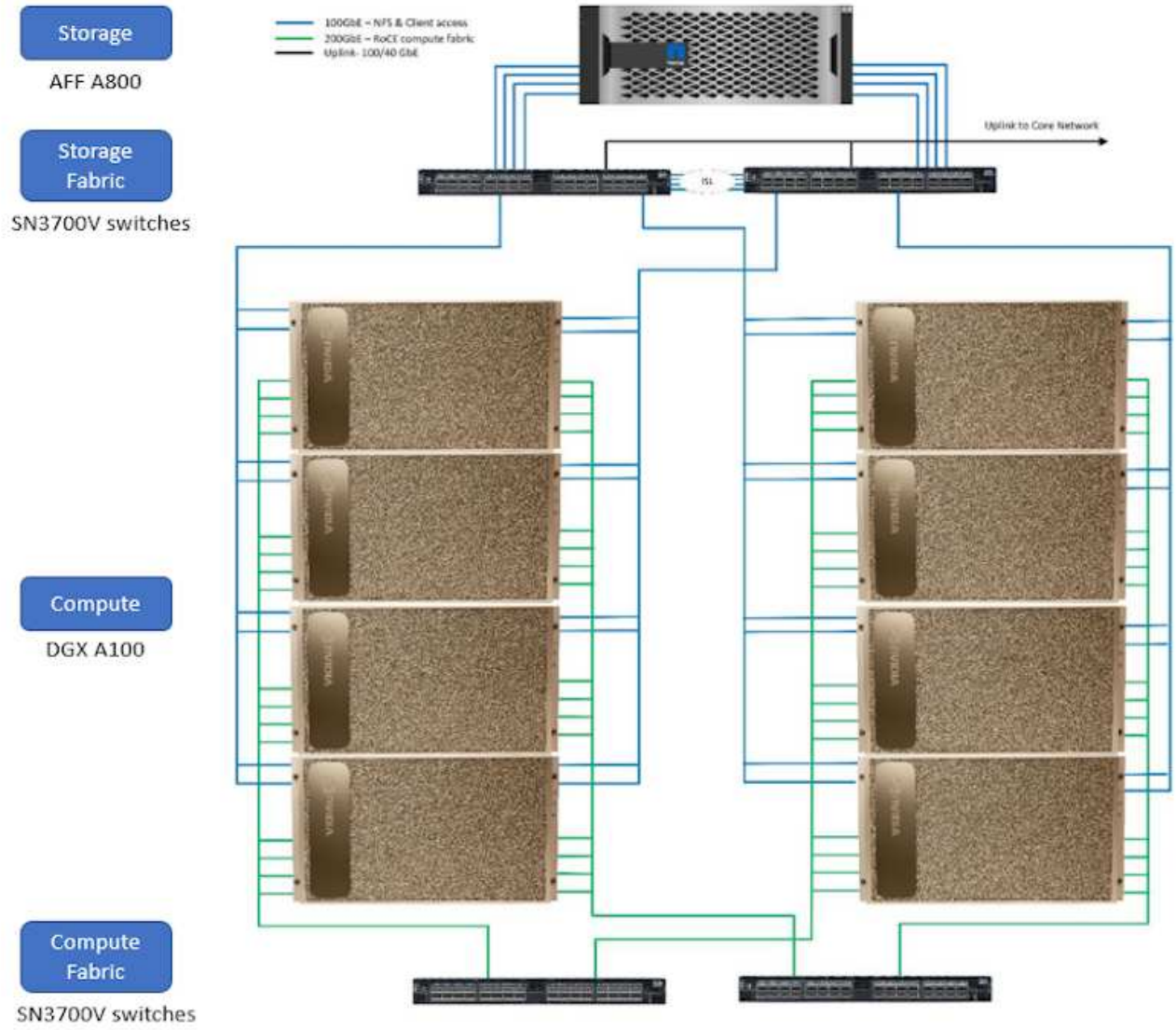
NetApp ONTAP ai

L'architettura di riferimento NetApp ONTAP ai, basata su sistemi DGX A100 e sistemi storage connessi al cloud NetApp, è stata sviluppata e verificata da NetApp e NVIDIA. Offre alle organizzazioni IT un'architettura che offre i seguenti vantaggi:

- Elimina le complessità di progettazione
- Consente una scalabilità indipendente di calcolo e storage
- Consente ai clienti di partire da piccoli e scalare perfettamente
- Offre una gamma di opzioni di storage per diverse performance e costi

ONTAP ai integra perfettamente i sistemi DGX A100 e i sistemi storage NetApp AFF A800 con reti all'avanguardia. ONTAP ai semplifica le implementazioni di ai eliminando la complessità e le congetture di progettazione. I clienti possono iniziare con poco e crescere senza interruzioni, gestendo in modo intelligente i dati dall'edge al core, fino al cloud e viceversa.

La figura seguente mostra diverse varianti della famiglia di soluzioni ai ONTAP con sistemi DGX A100. Le prestazioni del sistema AFF A800 sono verificate con un massimo di otto sistemi DGX A100. Aggiungendo coppie di controller storage al cluster ONTAP, l'architettura può scalare su più rack per supportare molti sistemi DGX A100 e petabyte di capacità storage con performance lineari. Questo approccio offre la flessibilità di modificare i rapporti calcolo-storage in modo indipendente in base alle dimensioni dei modelli DL utilizzati e alle metriche di performance richieste.



Per ulteriori informazioni su ONTAP ai, vedere "NVA-1153: NetApp ONTAP ai con sistemi NVIDIA DGX A100 e switch Ethernet Mellanox Spectrum."

NetApp ONTAP

ONTAP 9.11, l'ultima generazione di software per la gestione dello storage NetApp, consente alle aziende di modernizzare l'infrastruttura e passare a un data center predisposto per il cloud. Sfruttando le funzionalità di gestione dei dati leader del settore, ONTAP consente la gestione e la protezione dei dati con un singolo set di strumenti, indipendentemente dalla posizione dei dati. Puoi anche spostare liberamente i dati ovunque siano necessari: Edge, core o cloud. ONTAP 9.11 include numerose funzionalità che semplificano la gestione dei dati, accelerano e proteggono i dati critici e abilitano le funzionalità dell'infrastruttura di nuova generazione nelle architetture di cloud ibrido.

NetApp DataOps Toolkit

NetApp DataOps Toolkit è una libreria Python che consente a sviluppatori, data scientist, ingegneri DevOps e data engineer di eseguire facilmente varie attività di gestione dei dati, come il provisioning quasi istantaneo di un nuovo volume di dati o di un'area di lavoro JupyterLab, la clonazione quasi istantanea di un volume di dati o di un'area di lavoro JupyterLab, Snapshot quasi istantanee di un volume di dati o di uno spazio di lavoro

JupyterLab per la tracciabilità o il baselining. Questa libreria Python può funzionare come un'utility a riga di comando o una libreria di funzioni che è possibile importare in qualsiasi programma Python o notebook Jupyter.

Server di inferenza NVIDIA Triton

NVIDIA Triton Inference Server è un software open-source per l'inferenza che aiuta a standardizzare l'implementazione e l'esecuzione del modello per offrire ai in produzione rapida e scalabile. Triton Inference Server ottimizza l'inferenza ai consentendo ai team di implementare, eseguire e scalare modelli di ai addestrati da qualsiasi framework su qualsiasi infrastruttura basata su GPU o CPU. Triton Inference Server supporta tutti i framework principali, come TensorFlow, NVIDIA TensorRT, PyTorch, MXNet, OpenVINO e così via. Triton si integra con Kubernetes per l'orchestrazione e la scalabilità che puoi utilizzare in tutte le principali piattaforme ai e Kubernetes del cloud pubblico. È inoltre integrato con molte soluzioni software MLOPS.

PyTorch

"PyTorch" È un framework ML open-source. Si tratta di una libreria di tensore ottimizzata per il deep learning che utilizza GPU e CPU. Il pacchetto PyTorch contiene strutture di dati per i tensionatori multidimensionali che forniscono molte utility per la serializzazione efficiente dei tensionatori tra altre utili utility. Dispone inoltre di una controparte CUDA che consente di eseguire i calcoli del tensore su una GPU NVIDIA con funzionalità di calcolo. In questa convalida, utilizziamo la libreria OpenCV-Python (cv2) per validare il nostro modello, sfruttando al contempo i concetti di computer vision più intuitivi di Python.

Semplifica la gestione dei dati

La gestione dei dati è fondamentale per le operazioni IT aziendali e per i data scientist, in modo che le risorse appropriate vengano utilizzate per le applicazioni ai e per la formazione dei set di dati ai/ML. Le seguenti informazioni aggiuntive sulle tecnologie NetApp non rientrano nell'ambito di questa convalida, ma potrebbero essere rilevanti a seconda dell'implementazione.

Il software per la gestione dei dati ONTAP include le seguenti funzionalità per ottimizzare e semplificare le operazioni e ridurre il costo totale delle operazioni:

- Compaction dei dati inline e deduplica estesa. La compattazione dei dati riduce lo spazio sprecato all'interno dei blocchi di storage e la deduplica aumenta significativamente la capacità effettiva. Ciò vale per i dati memorizzati localmente e per i dati a più livelli nel cloud.
- Qualità del servizio (AQoS) minima, massima e adattativa. I controlli granulari della qualità del servizio (QoS) aiutano a mantenere i livelli di performance per le applicazioni critiche in ambienti altamente condivisi.
- NetApp FabricPool. Offre il tiering automatico dei dati cold per le opzioni di cloud storage pubblico e privato, tra cui Amazon Web Services (AWS), Azure e la soluzione di storage NetApp StorageGRID. Per ulteriori informazioni su FabricPool, vedere "TR-4598: Best practice FabricPool".

Accelera e proteggi i dati

ONTAP offre livelli superiori di performance e protezione dei dati ed estende queste funzionalità nei seguenti modi:

- Performance e latenza ridotta. ONTAP offre il throughput più elevato possibile con la latenza più bassa possibile.
- Protezione dei dati. ONTAP offre funzionalità di protezione dei dati integrate con gestione comune su tutte le piattaforme.

- NetApp Volume Encryption (NVE). ONTAP offre crittografia nativa a livello di volume con supporto per la gestione delle chiavi sia integrata che esterna.
- Multi-tenancy e autenticazione a più fattori. ONTAP consente la condivisione delle risorse dell'infrastruttura con i massimi livelli di sicurezza.

Infrastruttura a prova di futuro

ONTAP aiuta a soddisfare le esigenze di business esigenti e in continua evoluzione con le seguenti funzionalità:

- Scalabilità perfetta e operazioni senza interruzioni. ONTAP supporta l'aggiunta senza interruzioni di capacità ai controller esistenti e ai cluster scale-out. I clienti possono eseguire l'upgrade alle tecnologie più recenti, come NVMe e 32GB FC, senza costose migrazioni dei dati o interruzioni.
- Connessione al cloud. ONTAP è il software per la gestione dello storage più connesso al cloud, con opzioni per lo storage software-defined (ONTAP Select) e le istanze native del cloud (NetApp Cloud Volumes Service) in tutti i cloud pubblici.
- Integrazione con le applicazioni emergenti. ONTAP offre servizi dati di livello Enterprise per piattaforme e applicazioni di prossima generazione, come veicoli autonomi, città intelligenti e industria 4.0, utilizzando la stessa infrastruttura che supporta le applicazioni aziendali esistenti.

NetApp Astra Control

La famiglia di prodotti NetApp Astra offre servizi di storage e gestione dei dati applicativa per le applicazioni Kubernetes on-premise e nel cloud pubblico, basati sulle tecnologie di storage e gestione dei dati di NetApp. Consente di eseguire facilmente il backup delle applicazioni Kubernetes, migrare i dati in un cluster diverso e creare istantaneamente cloni applicativi funzionanti. Se è necessario gestire le applicazioni Kubernetes in esecuzione in un cloud pubblico, consultare la documentazione per "Servizio di controllo Astra". Astra Control Service è un servizio gestito da NetApp che fornisce la gestione dei dati applicativa dei cluster Kubernetes in Google Kubernetes Engine (GKE) e Azure Kubernetes Service (AKS).

NetApp Astra Trident

Astra "Trident" NetApp è uno storage dinamico open-source orchestrator per Docker e Kubernetes che semplifica la creazione, la gestione e il consumo dello storage persistente. Trident, un'applicazione nativa di Kubernetes, viene eseguita direttamente all'interno di un cluster Kubernetes. Trident consente ai clienti di implementare senza problemi le immagini dei container DL sullo storage NetApp e offre un'esperienza di livello Enterprise per le implementazioni dei container ai. Gli utenti di Kubernetes (sviluppatori ML, data scientist e così via) possono creare, gestire e automatizzare orchestrazione e cloning per sfruttare le funzionalità avanzate di gestione dei dati basate sulla tecnologia NetApp.

Copia e sincronizzazione di NetApp BlueXP

"Copia e sincronizzazione di BlueXP" È un servizio NetApp per una sincronizzazione dei dati rapida e sicura. Sia che tu debba trasferire file tra condivisioni di file SMB o NFS on-premise, NetApp StorageGRID, NetApp ONTAP S3, NetApp Cloud Volumes Service, Azure NetApp Files, Amazon Simple Storage Service (Amazon S3), Amazon Elastic file System (Amazon EFS), BLOB di Azure, Google Cloud Storage, o IBM Cloud Object Storage, BlueXP Copy and Sync sposta i file dove ne hai bisogno in modo rapido e sicuro. Una volta trasferiti, i dati sono completamente disponibili per l'utilizzo sia sull'origine che sulla destinazione. BlueXP Copy e Syncc sincronizzano costantemente i dati in base alla pianificazione predefinita, spostando solo i delta, in modo da ridurre al minimo tempo e denaro necessari per la replica dei dati. BlueXP Copy and Sync è un tool software-as-a-service (SaaS) estremamente semplice da configurare e utilizzare. I trasferimenti dei dati attivati da BlueXP Copy e Sync sono effettuati dai broker di dati. Puoi implementare i broker di dati BlueXP Copy e Sync

in AWS, Azure, Google Cloud Platform o on-premise.

Classificazione NetApp BlueXP

Basato su potenti algoritmi ai, "Classificazione NetApp BlueXP" offre controlli automatizzati e governance dei dati nell'intero data estate. Puoi individuare facilmente i risparmi sui costi, identificare i problemi di conformità e privacy e trovare opportunità di ottimizzazione. La dashboard di classificazione BlueXP ti fornisce le informazioni utili per identificare i dati duplicati per eliminare la ridondanza, mappare i dati personali, non personali e sensibili e attivare gli avvisi per i dati sensibili e le anomalie.

Piano di test e validazione

Per questa progettazione della soluzione, sono stati validati i seguenti tre scenari:

- Un'attività di inferenza, con e senza offuscamento di Protopia, all'interno di uno spazio di lavoro JupyterLab orchestrato utilizzando il NetApp DataOps Toolkit per Kubernetes.
- Un processo di deduzione in batch, con e senza offuscamento di Protopia, su Kubernetes con un volume di dati orchestrato utilizzando NetApp DataOps Toolkit per Kubernetes.
- Un'attività di deduzione che utilizza un'istanza di NVIDIA Triton Inference Server orchestrata utilizzando NetApp DataOps Toolkit per Kubernetes. Abbiamo applicato l'offuscamento protopico all'immagine prima di invocare l'API di inferenza Triton per simulare il requisito comune che prevede l'offuscamento dei dati trasmessi sulla rete. Questo flusso di lavoro è applicabile ai casi di utilizzo in cui i dati vengono raccolti all'interno di una zona attendibile ma devono essere trasferiti all'esterno di tale zona attendibile per l'deduzione. Senza l'offuscamento di Protopia, non è possibile implementare questo tipo di workflow senza che i dati sensibili abbandonino la zona attendibile.

Eseguire il test della configurazione

La seguente tabella descrive l'ambiente di convalida della progettazione della soluzione.

Componente	Versione
Kubernetes	1.21.6
Driver NetApp Astra Trident CSI	22.01.0
NetApp DataOps Toolkit per Kubernetes	2.3.0
Server di inferenza NVIDIA Triton	21.11-py3

Procedura di test

Questa sezione descrive le attività necessarie per completare la convalida.

Prerequisiti

Per eseguire le attività descritte in questa sezione, è necessario avere accesso a un host Linux o macOS con i seguenti strumenti installati e configurati:

- Kubernetl (configurato per l'accesso a un cluster Kubernetes esistente)
  - È possibile trovare le istruzioni di installazione e configurazione "qui".

- NetApp DataOps Toolkit per Kubernetes
  - È possibile trovare le istruzioni per l'installazione "qui".

Scenario 1 – deduzione on-demand in JupyterLab

1. Creare uno spazio dei nomi Kubernetes per i carichi di lavoro di inferenza ai/ML.

```
$ kubectl create namespace inference
namespace/inference created
```

2. Utilizza il NetApp DataOps Toolkit per eseguire il provisioning di un volume persistente per l'archiviazione dei dati su cui eseguire l'inferenza.

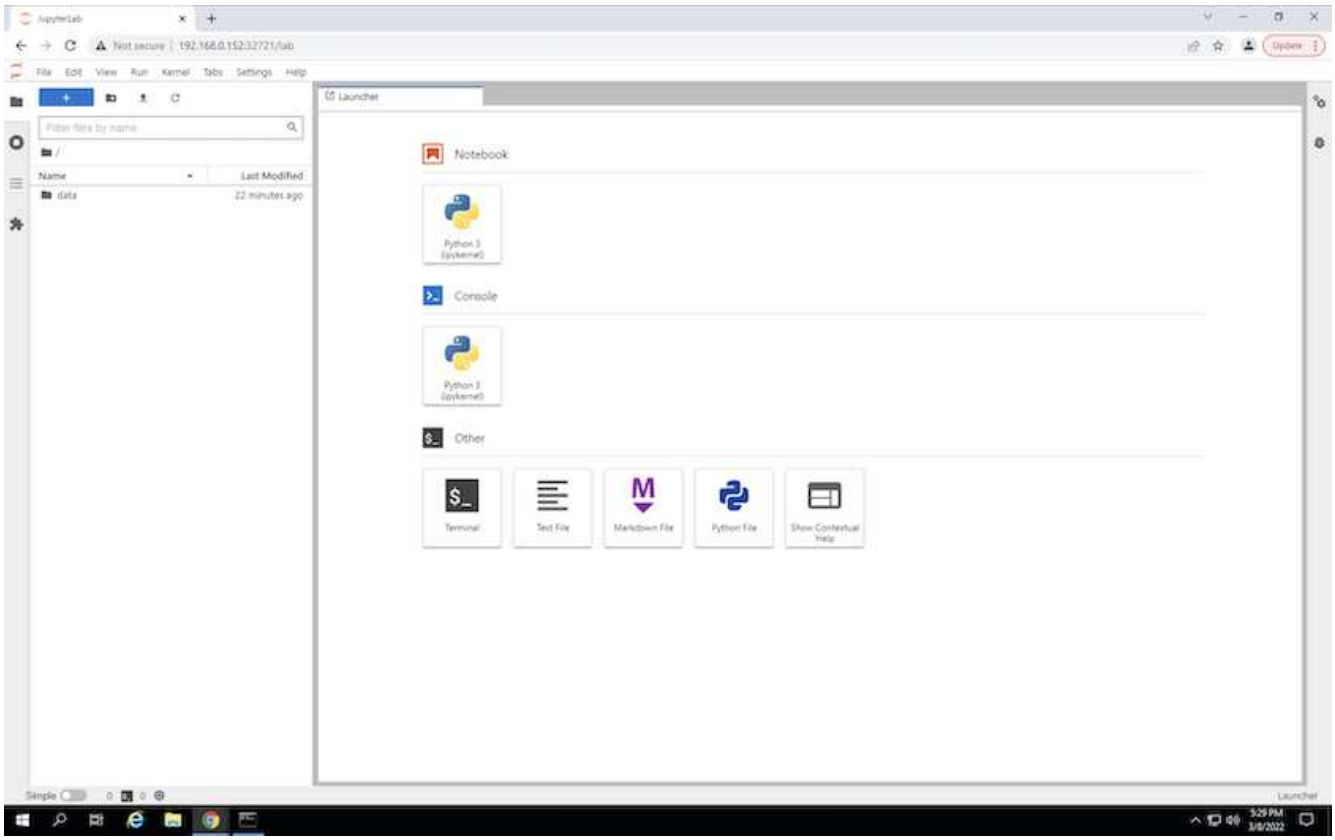
```
$ netapp_dataops_k8s_cli.py create volume --namespace=inference --pvc
-name=inference-data --size=50Gi
Creating PersistentVolumeClaim (PVC) 'inference-data' in namespace
'inference'.
PersistentVolumeClaim (PVC) 'inference-data' created. Waiting for
Kubernetes to bind volume to PVC.
Volume successfully created and bound to PersistentVolumeClaim (PVC)
'inference-data' in namespace 'inference'.
```

3. Utilizza il NetApp DataOps Toolkit per creare un nuovo spazio di lavoro JupyterLab. Montare il volume persistente creato nel passaggio precedente utilizzando `--mount- pvc` opzione. Allocare le GPU NVIDIA nell'area di lavoro secondo necessità utilizzando `-- nvidia-gpu` opzione.

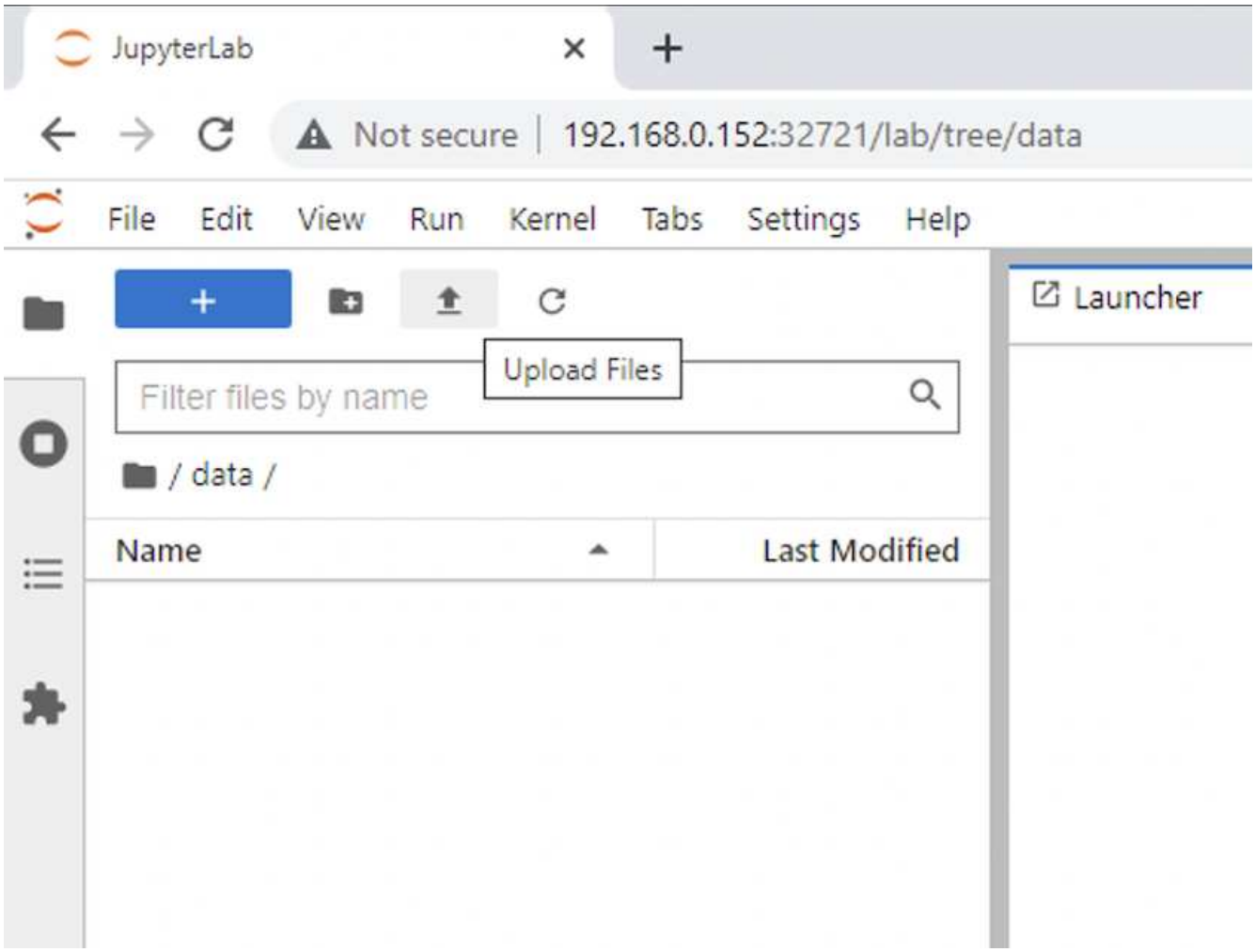
   Nell'esempio seguente, il volume persistente `inference-data` È montato sul container dello spazio di lavoro JupyterLab all'indirizzo `/home/jovyan/data`. Quando si utilizzano le immagini container ufficiali di Project Jupyter, `/home/jovyan` Viene presentato come la directory di primo livello all'interno dell'interfaccia web di JupyterLab.

```
$ netapp_dataops_k8s_cli.py create jupyterlab --namespace=inference
--workspace-name=live-inference --size=50Gi --nvidia-gpu=2 --mount
-pvc=inference-data:/home/jovyan/data
Set workspace password (this password will be required in order to
access the workspace):
Re-enter password:
Creating persistent volume for workspace...
Creating PersistentVolumeClaim (PVC) 'ntap-dsutil-jupyterlab-live-
inference' in namespace 'inference'.
PersistentVolumeClaim (PVC) 'ntap-dsutil-jupyterlab-live-inference'
created. Waiting for Kubernetes to bind volume to PVC.
Volume successfully created and bound to PersistentVolumeClaim (PVC)
'ntap-dsutil-jupyterlab-live-inference' in namespace 'inference'.
Creating Service 'ntap-dsutil-jupyterlab-live-inference' in namespace
'inference'.
Service successfully created.
Attaching Additional PVC: 'inference-data' at mount_path:
'/home/jovyan/data'.
Creating Deployment 'ntap-dsutil-jupyterlab-live-inference' in namespace
'inference'.
Deployment 'ntap-dsutil-jupyterlab-live-inference' created.
Waiting for Deployment 'ntap-dsutil-jupyterlab-live-inference' to reach
Ready state.
Deployment successfully created.
Workspace successfully created.
To access workspace, navigate to http://192.168.0.152:32721
```

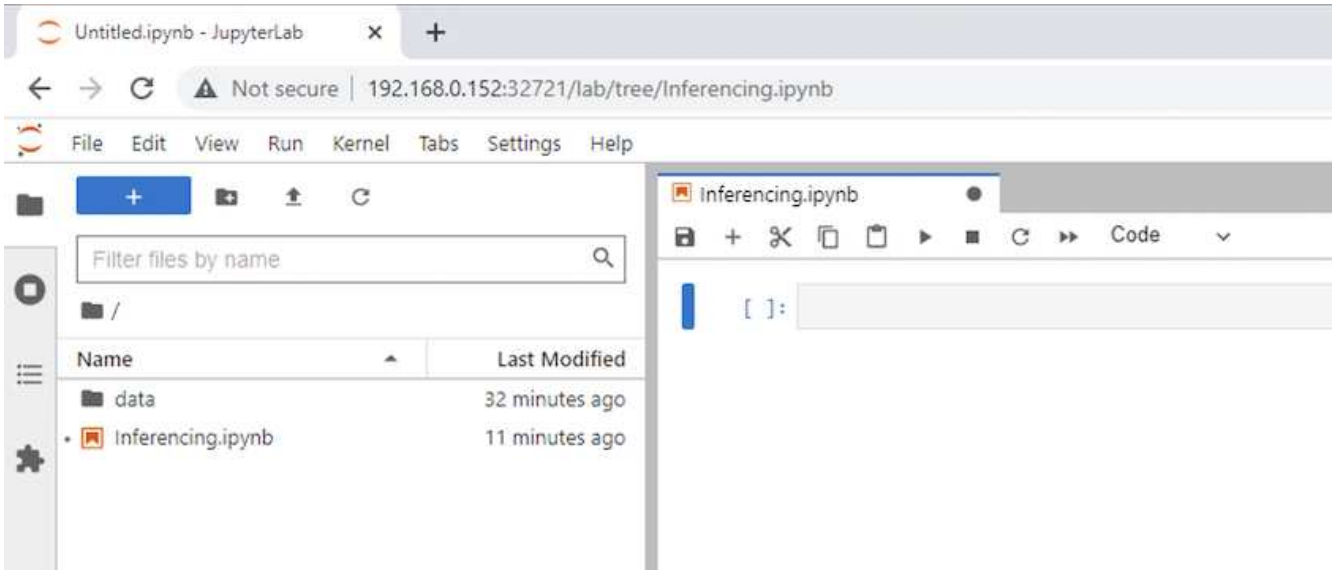
4. Accedere all'area di lavoro di JupyterLab utilizzando l'URL specificato nell'output di `create jupyterlab` comando. La directory dei dati rappresenta il volume persistente montato nell'area di lavoro.



5. Aprire `data` directory e caricare i file su cui eseguire la deduzione. Quando i file vengono caricati nella directory dei dati, vengono memorizzati automaticamente sul volume persistente montato nell'area di lavoro. Per caricare i file, fare clic sull'icona Upload Files (carica file), come mostrato nell'immagine seguente.



6. Tornare alla directory di livello superiore e creare un nuovo notebook.



7. Aggiungere il codice di deduzione al notebook. L’esempio seguente mostra il codice di deduzione per un caso d’uso di rilevamento dell’immagine.

Launcher | image-demo-pytorch.ipynb

Code | Python 3 (ipykernel)

STEP 3-1: Clean (Without obfuscation) detection

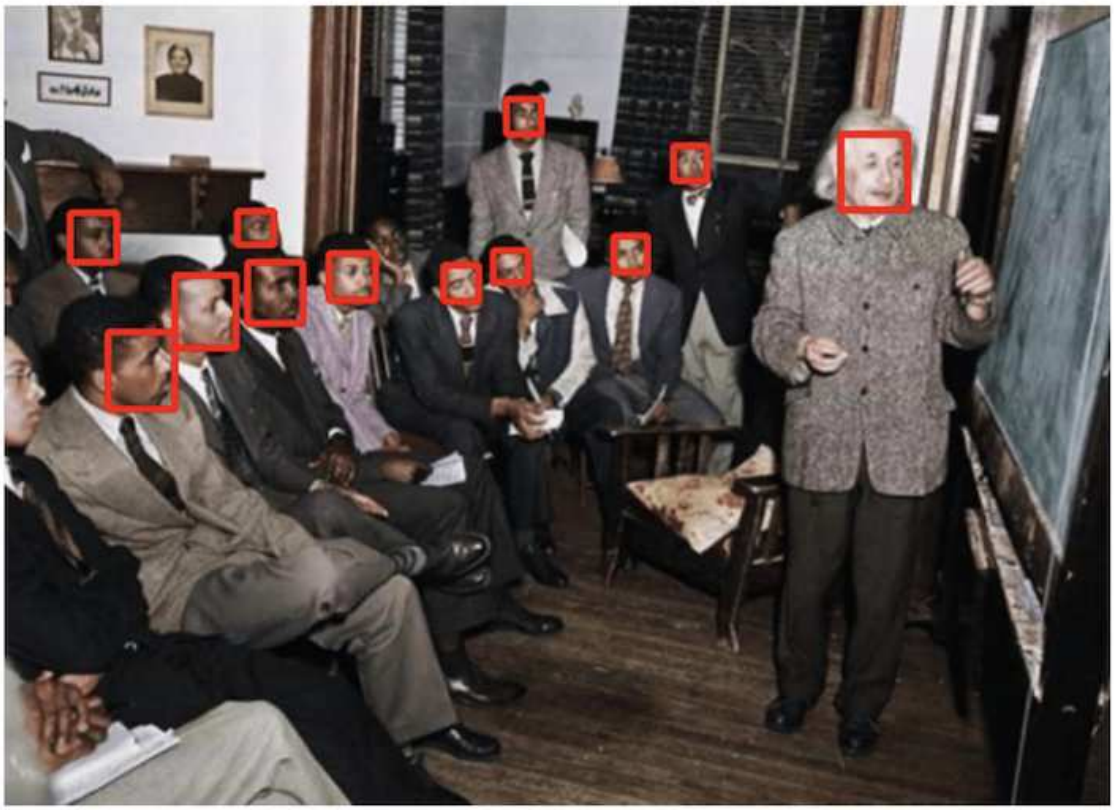
[9]:
```
# get current frame
frame = input_image

# preprocess input
preprocessed_input = preprocess_input(frame)
preprocessed_input = torch.Tensor(preprocessed_input).to(device)

# run forward pass
clean_activation = clean_model.forward_head(preprocessed_input)  # runs the first few layers
loc, pred = clean_model.forward_tail(clean_activation)  # runs rest of the layers

# postprocess output
clean_pred = (loc.detach().cpu().numpy(), pred.detach().cpu().numpy())
clean_outputs = postprocess_outputs(
    clean_pred, [[input_image_width, input_image_height]], priors, THRESHOLD
)

# draw rectangles
clean_frame = copy.deepcopy(frame)  # needs to be deep copy
for (x1, y1, x2, y2, s) in clean_outputs[0]:
    x1, y1 = int(x1), int(y1)
    x2, y2 = int(x2), int(y2)
    cv2.rectangle(clean_frame, (x1, y1), (x2, y2), (0, 0, 255), 4)
```

Launcher | image-demo-pytorch.ipynb

Code | Python 3 (ipykernel)

Visualize Clean (Without obfuscation) detection

[10]:
```
show_cv2_image(clean_frame, scale=2)
```



8. Aggiungi l’offuscamento di Protopia al tuo codice di deduzione. Protopia collabora direttamente con i clienti per fornire documentazione specifica per il caso d’utilizzo e non rientra nell’ambito di questo report tecnico. Nell’esempio seguente viene illustrato il codice di deduzione per un caso di utilizzo del rilevamento dell’immagine con l’aggiunta dell’offuscamento di Protopia.

Launcher | image-demo-pytorch.ipynb

Code | Python 3 (ipykernel)

STEP 3-2: Protopia AI (With obfuscation) detection

```
[11]: # get current frame
frame = input_image

# preprocess input
preprocessed_input = preprocess_input(frame)
preprocessed_input = torch.Tensor(preprocessed_input).to(device)

# run forward pass
not_noisy_activation = noisy_model.forward_head(preprocessed_input)  # runs the first few layers
#################################################################
#          SINGLE ADITIONAL LINE FOR PRIVATE INFERENCE          #
#################################################################
noisy_activation = noisy_model.forward_noise(not_noisy_activation)
#################################################################
loc, pred = noisy_model.forward_tail(noisy_activation)  # runs rest of the layers

# postprocess output
noisy_pred = (loc.detach().cpu().numpy(), pred.detach().cpu().numpy())
noisy_outputs = postprocess_outputs(
    noisy_pred, [[input_image_width, input_image_height]], priors, THRESHOLD * 0.5
)

# get reconstruction of the noisy activation
noisy_reconstruction = decoder_function(noisy_activation)
noisy_reconstruction = noisy_reconstruction.detach().cpu().numpy()[0]
noisy_reconstruction = unpreprocess_output(
    noisy_reconstruction, (input_image_width, input_image_height), True
).astype(np.uint8)

# draw rectangles
for (x1, y1, x2, y2, s) in noisy_outputs[0]:
    x1, y1 = int(x1), int(y1)
    x2, y2 = int(x2), int(y2)
    cv2.rectangle(noisy_reconstruction, (x1, y1), (x2, y2), (0, 0, 255), 4)
```

Visualize Protopia AI (With obfuscation) detection

Scenario 2 – deduzione in batch su Kubernetes

1. Creare uno spazio dei nomi Kubernetes per i carichi di lavoro di inferenza ai/ML.

```
$ kubectl create namespace inference
namespace/inference created
```

2. Utilizza il NetApp DataOps Toolkit per eseguire il provisioning di un volume persistente per l'archiviazione dei dati su cui eseguire l'inferenza.

```
$ netapp_dataops_k8s_cli.py create volume --namespace=inference --pvc
-name=inference-data --size=50Gi
Creating PersistentVolumeClaim (PVC) 'inference-data' in namespace
'inference'.
PersistentVolumeClaim (PVC) 'inference-data' created. Waiting for
Kubernetes to bind volume to PVC.
Volume successfully created and bound to PersistentVolumeClaim (PVC)
'inference-data' in namespace 'inference'.
```

3. Popolare il nuovo volume persistente con i dati su cui eseguire l'deduzione.

   Esistono diversi metodi per caricare i dati su un PVC. Se i tuoi dati sono attualmente memorizzati in una piattaforma di storage a oggetti compatibile con S3, come NetApp StorageGRID o Amazon S3, puoi utilizzare "NetApp DataOps Toolkit S3 Data Mover". Un altro metodo semplice consiste nel creare un'area di lavoro JupyterLab e quindi caricare i file attraverso l'interfaccia web di JupyterLab, come descritto nei passaggi da 3 a 5 della sezione "Scenario 1 – deduzione on-demand in JupyterLab."

4. Creare un lavoro Kubernetes per l'attività di deduzione in batch. Nell'esempio seguente viene illustrato un processo di deduzione in batch per un caso d'uso di rilevamento dell'immagine. Questo lavoro esegue la deduzione su ogni immagine in un set di immagini e scrive le metriche di precisione di deduzione su stdout.

```
$ vi inference-job-raw.yaml
apiVersion: batch/v1
kind: Job
metadata:
  name: netapp-inference-raw
  namespace: inference
spec:
  backoffLimit: 5
  template:
    spec:
      volumes:
      - name: data
        persistentVolumeClaim:
          claimName: inference-data
      - name: dshm
        emptyDir:
          medium: Memory
      containers:
      - name: inference
        image: netapp-protopia-inference:latest
        imagePullPolicy: IfNotPresent
        command: ["python3", "run-accuracy-measurement.py", "--dataset",
"/data/netapp-face-detection/FDDB"]
        resources:
          limits:
            nvidia.com/gpu: 2
        volumeMounts:
        - mountPath: /data
          name: data
        - mountPath: /dev/shm
          name: dshm
      restartPolicy: Never
$ kubectl create -f inference-job-raw.yaml
job.batch/netapp-inference-raw created
```

5. Verificare che il lavoro di deduzione sia stato completato correttamente.

```
$ kubectl -n inference logs netapp-inference-raw-255sp
100%|██████████| 89/89 [00:52<00:00,  1.68it/s]
Reading Predictions : 100%|██████████| 10/10 [00:01<00:00,  6.23it/s]
Predicting ... : 100%|██████████| 10/10 [00:16<00:00,  1.64s/it]
==================== Results ====================
FDDB-fold-1 Val AP: 0.9491256561145955
FDDB-fold-2 Val AP: 0.9205024466101926
FDDB-fold-3 Val AP: 0.9253013871078468
FDDB-fold-4 Val AP: 0.9399781485863011
FDDB-fold-5 Val AP: 0.9504280149478732
FDDB-fold-6 Val AP: 0.9416473519339292
FDDB-fold-7 Val AP: 0.9241631566241117
FDDB-fold-8 Val AP: 0.9072663297546659
FDDB-fold-9 Val AP: 0.9339648715035469
FDDB-fold-10 Val AP: 0.9447707905560152
FDDB Dataset Average AP: 0.9337148153739079
==================================================
mAP: 0.9337148153739079
```

6. Aggiungi l'offuscamento di Protopia al tuo lavoro di deduzione. È possibile trovare istruzioni specifiche per l'aggiunta di offuscamento Protopia direttamente da Protopia, che non rientra nell'ambito di questo report tecnico. Nell'esempio seguente viene illustrato un processo di deduzione in batch per un caso di utilizzo del rilevamento dei volti con offuscamento di Protopia aggiunto utilizzando un valore ALFA di 0.8. Questo lavoro applica l'offuscamento di Protopia prima di eseguire la deduzione per ogni immagine in un set di immagini e quindi scrive le metriche di precisione dell'inferenza su stdout.

   Abbiamo ripetuto questo passaggio per i valori ALFA 0.05, 0.1, 0.2, 0.4, 0.6, 0.8, 0.9 e 0.95. I risultati sono riportati in ""Confronto della precisione delle conferenze"."

```
$ vi inference-job-protopia-0.8.yaml
apiVersion: batch/v1
kind: Job
metadata:
  name: netapp-inference-protopia-0.8
  namespace: inference
spec:
  backoffLimit: 5
  template:
    spec:
      volumes:
      - name: data
        persistentVolumeClaim:
          claimName: inference-data
      - name: dshm
        emptyDir:
          medium: Memory
      containers:
      - name: inference
        image: netapp-protopia-inference:latest
        imagePullPolicy: IfNotPresent
        env:
        - name: ALPHA
          value: "0.8"
        command: ["python3", "run-accuracy-measurement.py", "--dataset",
"/data/netapp-face-detection/FDDB", "--alpha", "$(ALPHA)", "--noisy"]
        resources:
          limits:
            nvidia.com/gpu: 2
        volumeMounts:
        - mountPath: /data
          name: data
        - mountPath: /dev/shm
          name: dshm
      restartPolicy: Never
$ kubectl create -f inference-job-protopia-0.8.yaml
job.batch/netapp-inference-protopia-0.8 created
```

7. Verificare che il lavoro di deduzione sia stato completato correttamente.

```
$ kubectl -n inference logs netapp-inference-protopia-0.8-b4dkz
100%|██████████| 89/89 [01:05<00:00,  1.37it/s]
Reading Predictions : 100%|██████████| 10/10 [00:02<00:00,  3.67it/s]
Predicting ... : 100%|██████████| 10/10 [00:22<00:00,  2.24s/it]
==================== Results ====================
FDDB-fold-1 Val AP: 0.8953066115834589
FDDB-fold-2 Val AP: 0.8819580264029936
FDDB-fold-3 Val AP: 0.8781107458462862
FDDB-fold-4 Val AP: 0.9085731346308461
FDDB-fold-5 Val AP: 0.9166445508275378
FDDB-fold-6 Val AP: 0.9101178994188819
FDDB-fold-7 Val AP: 0.8383443678423771
FDDB-fold-8 Val AP: 0.8476311547659464
FDDB-fold-9 Val AP: 0.8739624502111121
FDDB-fold-10 Val AP: 0.8905468076424851
FDDB Dataset Average AP: 0.8841195749171925
==================================================
mAP: 0.8841195749171925
```

Scenario 3 – NVIDIA Triton Inference Server

1. Creare uno spazio dei nomi Kubernetes per i carichi di lavoro di inferenza ai/ML.

```
$ kubectl create namespace inference
namespace/inference created
```

2. Utilizza NetApp DataOps Toolkit per eseguire il provisioning di un volume persistente da utilizzare come repository di modelli per NVIDIA Triton Inference Server.

```
$ netapp_dataops_k8s_cli.py create volume --namespace=inference --pvc
-name=triton-model-repo --size=100Gi
Creating PersistentVolumeClaim (PVC) 'triton-model-repo' in namespace
'inference'.
PersistentVolumeClaim (PVC) 'triton-model-repo' created. Waiting for
Kubernetes to bind volume to PVC.
Volume successfully created and bound to PersistentVolumeClaim (PVC)
'triton-model-repo' in namespace 'inference'.
```

3. Memorizzare il modello sul nuovo volume persistente in un "formato" Riconosciuto da NVIDIA Triton Inference Server.

   Esistono diversi metodi per caricare i dati su un PVC. Un metodo semplice consiste nel creare un'area di lavoro JupyterLab e quindi caricare i file attraverso l'interfaccia web di JupyterLab, come descritto nei passaggi da 3 a 5 in "Scenario 1 – deduzione on-demand in JupyterLab. "

4. Utilizza NetApp DataOps Toolkit per implementare una nuova istanza di NVIDIA Triton Inference Server.

```
$ netapp_dataops_k8s_cli.py create triton-server --namespace=inference
--server-name=netapp-inference --model-repo-pvc-name=triton-model-repo
Creating Service 'ntap-dsutil-triton-netapp-inference' in namespace
'inference'.
Service successfully created.
Creating Deployment 'ntap-dsutil-triton-netapp-inference' in namespace
'inference'.
Deployment 'ntap-dsutil-triton-netapp-inference' created.
Waiting for Deployment 'ntap-dsutil-triton-netapp-inference' to reach
Ready state.
Deployment successfully created.
Server successfully created.
Server endpoints:
http: 192.168.0.152: 31208
grpc: 192.168.0.152: 32736
metrics: 192.168.0.152: 30009/metrics
```

5. Utilizzare un SDK del client Triton per eseguire un'attività di deduzione. Il seguente estratto di codice Python utilizza l'SDK del client Python di Triton per eseguire un'attività di deduzione per un caso di utilizzo del rilevamento dei volti. Questo esempio chiama l'API Triton e passa un'immagine per la deduzione. Il server di inferenza Triton riceve quindi la richiesta, richiama il modello e restituisce l'output di deduzione come parte dei risultati API.

```
# get current frame
frame = input_image
# preprocess input
preprocessed_input = preprocess_input(frame)
preprocessed_input = torch.Tensor(preprocessed_input).to(device)
# run forward pass
clean_activation = clean_model_head(preprocessed_input)  # runs the
first few layers
#######################################################################
##############
#          pass clean image to Triton Inference Server API for
inferencing           #
#######################################################################
##############
triton_client =
httpclient.InferenceServerClient(url="192.168.0.152:31208",
verbose=False)
model_name = "face_detection_base"
inputs = []
outputs = []
```

```
inputs.append(httpclient.InferInput("INPUT__0", [1, 128, 32, 32],
"FP32"))
inputs[0].set_data_from_numpy(clean_activation.detach().cpu().numpy(),
binary_data=False)
outputs.append(httpclient.InferRequestedOutput("OUTPUT__0",
binary_data=False))
outputs.append(httpclient.InferRequestedOutput("OUTPUT__1",
binary_data=False))
results = triton_client.infer(
    model_name,
    inputs,
    outputs=outputs,
    #query_params=query_params,
    headers=None,
    request_compression_algorithm=None,
    response_compression_algorithm=None)
#print(results.get_response())
statistics =
triton_client.get_inference_statistics(model_name=model_name,
headers=None)
print(statistics)
if len(statistics["model_stats"]) != 1:
    print("FAILED: Inference Statistics")
    sys.exit(1)

loc_numpy = results.as_numpy("OUTPUT__0")
pred_numpy = results.as_numpy("OUTPUT__1")
#######################################################################
##############
# postprocess output
clean_pred = (loc_numpy, pred_numpy)
clean_outputs = postprocess_outputs(
    clean_pred, [[input_image_width, input_image_height]], priors,
THRESHOLD
)
# draw rectangles
clean_frame = copy.deepcopy(frame)  # needs to be deep copy
for (x1, y1, x2, y2, s) in clean_outputs[0]:
    x1, y1 = int(x1), int(y1)
    x2, y2 = int(x2), int(y2)
    cv2.rectangle(clean_frame, (x1, y1), (x2, y2), (0, 0, 255), 4)
```

6. Aggiungi l’offuscamento di Protopia al tuo codice di deduzione. È possibile trovare istruzioni specifiche per il caso d’utilizzo per aggiungere l’offuscamento Protopia direttamente da Protopia; tuttavia, questo processo non rientra nell’ambito di questo report tecnico. Nell’esempio seguente viene illustrato lo stesso codice Python mostrato nel precedente passaggio 5, ma con l’aggiunta dell’offuscamento di Protopia.

Si noti che l'offuscamento Protopia viene applicato all'immagine prima che venga passata all'API Triton. Pertanto, l'immagine non offuscata non lascia mai la macchina locale. Solo l'immagine offuscata viene passata attraverso la rete. Questo flusso di lavoro è applicabile ai casi di utilizzo in cui i dati vengono raccolti all'interno di una zona attendibile, ma devono essere trasferiti all'esterno di tale zona attendibile per l'deduzione. Senza l'offuscamento di Protopia, non è possibile implementare questo tipo di workflow senza che i dati sensibili si allontanino dalla zona di fiducia.

```
# get current frame
frame = input_image
# preprocess input
preprocessed_input = preprocess_input(frame)
preprocessed_input = torch.Tensor(preprocessed_input).to(device)
# run forward pass
not_noisy_activation = noisy_model_head(preprocessed_input)  # runs the
first few layers
#################################################################
#          obfuscate image locally prior to inferencing          #
#          SINGLE ADITIONAL LINE FOR PRIVATE INFERENCE           #
#################################################################
noisy_activation = noisy_model_noise(not_noisy_activation)
#################################################################
#######################################################################
####################
#          pass obfuscated image to Triton Inference Server API for
inferencing           #
#######################################################################
####################
triton_client =
httpclient.InferenceServerClient(url="192.168.0.152:31208",
verbose=False)
model_name = "face_detection_noisy"
inputs = []
outputs = []
inputs.append(httpclient.InferInput("INPUT__0", [1, 128, 32, 32],
"FP32"))
inputs[0].set_data_from_numpy(noisy_activation.detach().cpu().numpy(),
binary_data=False)
outputs.append(httpclient.InferRequestedOutput("OUTPUT__0",
binary_data=False))
outputs.append(httpclient.InferRequestedOutput("OUTPUT__1",
binary_data=False))
results = triton_client.infer(
    model_name,
    inputs,
    outputs=outputs,
    #query_params=query_params,
    headers=None,
```

```
    request_compression_algorithm=None,
    response_compression_algorithm=None)
#print(results.get_response())
statistics =
triton_client.get_inference_statistics(model_name=model_name,
headers=None)
print(statistics)
if len(statistics["model_stats"]) != 1:
    print("FAILED: Inference Statistics")
    sys.exit(1)

loc_numpy = results.as_numpy("OUTPUT__0")
pred_numpy = results.as_numpy("OUTPUT__1")
#######################################################################
###################

# postprocess output
noisy_pred = (loc_numpy, pred_numpy)
noisy_outputs = postprocess_outputs(
    noisy_pred, [[input_image_width, input_image_height]], priors,
THRESHOLD * 0.5
)
# get reconstruction of the noisy activation
noisy_reconstruction = decoder_function(noisy_activation)
noisy_reconstruction = noisy_reconstruction.detach().cpu().numpy()[0]
noisy_reconstruction = unpreprocess_output(
    noisy_reconstruction, (input_image_width, input_image_height), True
).astype(np.uint8)
# draw rectangles
for (x1, y1, x2, y2, s) in noisy_outputs[0]:
    x1, y1 = int(x1), int(y1)
    x2, y2 = int(x2), int(y2)
    cv2.rectangle(noisy_reconstruction, (x1, y1), (x2, y2), (0, 0, 255),
4)
```

Confronto della precisione delle deduzione

Per questa convalida, abbiamo eseguito la deduzione per un caso d'utilizzo di rilevamento dell'immagine utilizzando un set di immagini raw. Quindi, abbiamo eseguito la stessa attività di deduzione sullo stesso set di immagini con l'aggiunta dell'offuscamento di Protopia prima dell'inferenza. Abbiamo ripetuto l'attività utilizzando diversi valori DI ALPHA per la componente di offuscamento Protopia. Nel contesto dell'offuscamento di Protopia, il valore ALFA rappresenta la quantità di offuscamento applicata, con un valore ALFA più alto che rappresenta un livello più elevato di offuscamento. Abbiamo quindi confrontato la precisione delle deduzione in queste

diverse esecuzioni.

Le due tabelle seguenti forniscono dettagli sul nostro caso di utilizzo e delineano i risultati.

Protopia collabora direttamente con i clienti per determinare il valore ALFA appropriato per un caso di utilizzo specifico.

Componente	Dettagli
Modello	FaceBoxes (PyTorch) -
Dataset	Dataset FDDB

Offuscamento di Protopia	ALFA	Precisione
No	N/A.	0.9337148153739079
Sì	0.05	0.9028766627325002
Sì	0.1	0.9024301009661478
Sì	0.2	0.9081836283186224
Sì	0.4	0.9073066107482036
Sì	0.6	0.8847816568680239
Sì	0.8	0.8841195749171925
Sì	0.9	0.8455427675252052
Sì	0.95	0.8455427675252052

Velocità di offuscamento

Per questa convalida, abbiamo applicato l'offuscamento di Protopia a un'immagine da 1920 x 1080 pixel cinque volte e misurato il tempo necessario per completare ogni volta la fase di offuscamento.

Abbiamo utilizzato PyTorch in esecuzione su una singola GPU NVIDIA V100 per applicare l'offuscamento e abbiamo cancellato la cache della GPU tra un'esecuzione e l'altra. La fase di offuscamento ha richiesto rispettivamente 5,47 ms, 5,27 ms, 4,54 ms, 5,24 ms e 4,84 ms per completare le cinque corse. La velocità media è stata di 5,072 ms.

Conclusione

I dati esistono in tre stati: A riposo, in transito e nel calcolo. Una parte importante di qualsiasi servizio di inferenza ai dovrebbe essere la protezione dei dati dalle minacce durante l'intero processo. La protezione dei dati durante le deduzione è fondamentale perché il processo può esporre informazioni private sui clienti esterni e sull'azienda che fornisce il servizio di deduzione. Protopia ai è una soluzione software-only non invadente per deduzione ai riservata nel mercato odierno. Con Protopia, l'ai viene alimentata solo le informazioni trasformate nei record di dati che sono essenziali per eseguire l'attività di ai/ML a portata di mano e niente di più. Questa trasformazione stocastica non è una

forma di mascheramento e si basa sul cambiamento matematico della rappresentazione dei dati utilizzando il rumore curato.

I sistemi storage NetApp con funzionalità ONTAP offrono le stesse performance o migliori dello storage SSD locale e, in combinazione con il toolkit NetApp DataOps, offrono i seguenti vantaggi a data scientist, data engineer, sviluppatori ai/ML e decision maker IT aziendali o aziendali:

- Condivisione semplice dei dati tra sistemi ai, analytics e altri sistemi aziendali critici. Questa condivisione dei dati riduce l'overhead dell'infrastruttura, migliora le performance e ottimizza la gestione dei dati in tutta l'azienda.
- Calcolo e storage scalabili in maniera indipendente per ridurre al minimo i costi e migliorare l'utilizzo delle risorse.
- Workflow di sviluppo e implementazione ottimizzati grazie a copie Snapshot e cloni integrati per spazi di lavoro degli utenti istantanei ed efficienti in termini di spazio, controllo integrato delle versioni e implementazione automatizzata.
- Protezione dei dati di livello Enterprise e governance dei dati per disaster recovery, business continuity e requisiti normativi.
- Invocazione semplificata delle operazioni di gestione dei dati; copie Snapshot delle aree di lavoro dei data scientist per il backup e la tracciabilità dal NetApp DataOps Toolkit nei notebook Jupyter.

La soluzione NetApp e Protopia offre un'architettura scalabile e flessibile, ideale per le implementazioni di inferenza ai di livello Enterprise. Consente la protezione dei dati e fornisce privacy per le informazioni sensibili, laddove i requisiti di inferenza ai confidenziali possono essere soddisfatti con pratiche ai responsabili sia nelle implementazioni on-premise che nel cloud ibrido.

Dove trovare ulteriori informazioni e riconoscimenti

Per ulteriori informazioni sulle informazioni descritte in questo documento, consultare i seguenti documenti e/o siti Web:

- Software per la gestione dei dati NetApp ONTAP: Libreria di informazioni ONTAP

  http://mysupport.netapp.com/documentation/productlibrary/index.html?productID=62286

- Storage persistente NetApp per container - NetApp Trident

  "https://netapp.io/persistent-storage-provisioner-for-kubernetes/"

- NetApp DataOps Toolkit

  "https://github.com/NetApp/netapp-dataops-toolkit"

- Storage persistente NetApp per container: NetApp Astra Trident

  "https://netapp.io/persistent-storage-provisioner-for-kubernetes/"

- Protopia ai: Inferenza riservata

  "https://protopia.ai/blog/protopia-ai-takes-on-the-missing-link-in-ai-privacy-confidential-inference/"

- Copia e sincronizzazione di NetApp BlueXP

"https://docs.netapp.com/us-en/occm/concept_cloud_sync.html#how-cloud-sync-works"

- Server di inferenza NVIDIA Triton

  "https://developer.nvidia.com/nvidia-triton-inference-server"

- Documentazione di NVIDIA Triton Inference Server

  "https://docs.nvidia.com/deeplearning/triton-inference-server/index.html"

- FaceBoxes in PyTorch

  "https://github.com/zisianw/FaceBoxes.PyTorch"

Ringraziamenti

- Mark Cates, Principal Product Manager, NetApp
- Sufian Ahmad, Technical Marketing Engineer, NetApp
- Hadi Esmaeilzadeh, Chief Technology Officer e Professor, Protopia ai