



Optimierung der KI-Pipeline über NetApp Speicherplattformen hinweg

NetApp artificial intelligence solutions

NetApp
September 25, 2026

Inhalt

- Optimierung der KI-Pipeline über NetApp Speicherplattformen hinweg 1
 - Optimierung der KI-Pipeline über NetApp Speicherplattformen hinweg (StorageGRID, ONTAP und E-Series) 1
 - 1. Zusammenfassung 1
 - 2. Lösungsübersicht, Geschäftswert und Kundenvorteile 4
 - 3. Architekturübersicht 6
 - 4. NetApp, Technologiekomponenten von Drittanbietern und Open Source 10
 - 5. Lösungsdesign und Details zur Storage Architektur 11
 - 6. Datenmobilität mit NetApp XCP 15
 - 7. Bereitstellungsarchitektur 18
 - 8. Konfigurationsreferenz 23
 - 9. Betriebshandbuch 40
 - 10. Sicherheitsaspekte, Validierung und Tests 42
 - 11. NetApp Differenzierung und Wettbewerbspositionierung, Anhänge 47

Optimierung der KI-Pipeline über NetApp Speicherplattformen hinweg

Optimierung der KI-Pipeline über NetApp Speicherplattformen hinweg (StorageGRID, ONTAP und E-Series)

Karthikeyan Nagalingam, NetApp

Diese validierte Architektur präsentiert eine gesteuerte, speicherebenenbewusste KI-Datenpipeline, die durch Apache Airflow orchestriert wird. Die Lösung erfasst tabellarische Rohdaten und Textdaten aus NetApp StorageGRID, erstellt deterministische Datensätze und Manifeste in einem ONTAP NAS Bucket und verwendet NetApp XCP, um aufbereitete Daten je nach Laufzeitkonfiguration entweder nach NetApp ONTAP S3 oder LustreFS zu verschieben. Anschließend trainiert und optimiert sie scikit-learn-Modelle inkrementell, generiert Vorhersagen, speichert erfolgreiche Trainingsergebnisse als Checkpoints und archiviert Modelle, Metriken, Manifeste und Vorhersagenachweise in StorageGRID. Die Architektur bietet reproduzierbare Ausführung, konfigurierbare Speicherplatzierung, Schutzmechanismen bei Fehlern, checkpointbasierte Wiederherstellung und nachvollziehbare Datenherkunft über den gesamten KI-Lebenszyklus hinweg. Sie ist für Architekten, Dateningenieure, ML-Ingenieure und Infrastrukturteams vorgesehen, die ein wiederholbares Bereitstellungs- und Validierungsmuster für KI-Pipelines über NetApp Speicherebenen hinweg benötigen.

Inhaltsverzeichnis

1. ["Zusammenfassung"](#)
2. ["Lösungsübersicht, Geschäftswert und Kundenvorteile"](#)
3. ["Architekturübersicht"](#)
4. ["NetApp, Drittanbieter- und Open Source-Technologiekomponenten"](#)
5. ["Details zu Lösungsdesign und Speicherarchitektur"](#)
6. ["Datenmobilität mit NetApp XCP"](#)
7. ["Bereitstellungsarchitektur"](#)
8. ["Konfigurationsreferenz"](#)
9. ["Betriebsanleitung"](#)
10. ["Sicherheitsaspekte, Validierung und Tests"](#)
11. ["NetApp Differenzierung und Wettbewerbspositionierung, Anhänge"](#)

1. Zusammenfassung

Karthikeyan Nagalingam, NetApp

[[1-1-purpose-of-this-document]]
== 1.1 Zweck dieses Dokuments

Diese NetApp Validated Architecture (NVA) dokumentiert eine produktionsreife KI-Datenpipeline, die Rohdaten erfasst, sie für maschinelles Lernen vorbereitet, sie mit hoher Geschwindigkeit mithilfe von NetApp XCP

zwischen Storage Tiers verschiebt, Modelle trainiert und inkrementell optimiert, Vorhersagen generiert und die Ergebnisse für Governance und Reproduzierbarkeit archiviert. Sie dient als Leitfaden für Architekten und Infrastrukturteams in Bezug auf Designbegründung, Bereitstellung, Konfiguration und Betrieb.

[[1-2-audience]]
== 1.2 Zielgruppe

Lösungsarchitekten, Speicher-/Infrastrukturingenieure, Datenplattformingenieure, ML-Ingenieure und IT-Entscheidungsträger, die NetApp Speicher- und Datenmobilitätstechnologie für KI/ML-Workloads evaluieren.

[[1-3-business-challenge-overview]]
== 1.3 Überblick über die geschäftlichen Herausforderungen

KI-Teams benötigen Daten, die im Objektspeicher landen, um zuverlässig transformiert und anschließend an die Rechen- und Speicherebene übertragen zu werden, die das Training am besten unterstützt, beispielsweise an elastisches S3 für allgemeine Workloads oder an leistungsstarke parallele Dateisysteme (LustreFS) für E/A-intensive Trainings. Ohne eine kontrollierte Datenmobilität und Orchestrierungsschicht sind Teams auf fehleranfällige, spezialisierte Skripte angewiesen, denen Wiederholungsversuche, Checkpointing, Speicherflexibilität und Audit-Trails fehlen.

KI-Programme in Unternehmen stagnieren, nicht aufgrund von Modell- oder Rechenleistungsbeschränkungen, sondern weil herkömmliche Speichersysteme nicht für KI ausgelegt sind. Das Verschieben massiver Datensätze, die kontinuierliche Auslastung der GPUs während des Trainings, die Verwaltung des Checkpoint-Overheads und die Kontrolle der Speicherkosten erfordern mehr Aufwand als die eigentliche KI-Arbeit. Einstufige Architekturen erzwingen einen Kompromiss zwischen Leistung und Kosten, der sich mit zunehmender Größe von Modellen und Datensätzen noch verstärkt.

[[1-4-netapp-solution-summary]]
== 1.4 NetApp Lösungsübersicht

Die Lösung orchestriert Datenerfassung, -aufbereitung, Datenmobilität, Training, Feinabstimmung, Inferenz und Archivierung als Apache Airflow DAG (Directed Acyclic Graph). StorageGRID bildet die Basis für die Rohdaten- und Langzeitarchivierungs-Objekt-Tiers. Die Datenaufbereitung schreibt mit einem Laufzeitstempel versehene Datensätze und Manifeste in einen ONTAP NAS Bucket, der über NFS als XCP Quelle bereitgestellt wird. NetApp XCP verschiebt die aufbereiteten Daten dann in ein zur Laufzeit auswählbares Trainingsziel, NetApp ONTAP S3 oder LustreFS, ohne dass zum Wechseln der Tiers Codeänderungen erforderlich sind.



In diesem Design wird der ONTAP NAS Bucket primär aus Gründen der XCP-Unterstützung und -Validierung verwendet, da er eine konsistente NFS-Quelle für den Mobilitäts-Workflow bereitstellt. In realen Produktionsumgebungen können dieselben aktiven Daten auch direkt über leistungsstarke RDMA-fähige Pfade bereitgestellt werden, sodass die ONTAP NAS Schicht eher als betrieblich komfortables und supportfähiges Staging-Muster als als einzige Zugriffsmethode zur Laufzeit zu betrachten ist.

Diese Pipeline ist eine konkrete Implementierung der umfassenderen **NetApp AI Storage Architecture**, eines speziell entwickelten, dreistufigen Speicherframeworks, das Leistung und Kosten auf jede Phase der KI-Workload abstimmt:

- **E-Series (LustreFS):** Paralleler Speicher mit extrem hohem Durchsatz für GPU-intensives Modelltraining und Checkpointing.
- **ONTAP (AFF/AFX):** Hochleistungsfähiger aktiver Speicher für Training, Feinabstimmung, Inferenz, Vektor-/RAG-Workloads und ausgewählte Anwendungsfälle der Datenaufbereitung mit FAS und NAS Bucket Unterstützung.

- **StorageGRID:** Skalierbarer, S3-kompatibler Objektspeicher für Datenerfassung, Governance und Langzeitaufbewahrung.

Die Datenverschiebung zwischen den Ebenen wird vollständig über Apache Airflow und NetApp XCP automatisiert, sodass manuelle Vorgänge vollständig aus dem kritischen Pfad entfallen.

Die NetApp Speicherarchitektur unterstützt Unternehmen bei der verteilten Datenaufbereitung, dem Modelltraining, der Feinabstimmung, der Inferenz und der Lebenszyklusverwaltung, ohne alle Daten in einem einzigen zentralen Data Lake zusammenführen zu müssen. Durch die Reduzierung unnötiger Datenbewegungen eignet sie sich ideal für hybride KI-Workflows.

[[1-5-why-netapp]]
 == 1.5 Warum NetApp

Dimension	Konventioneller Ansatz	NetApp AI Speicherarchitektur
GPU Produktivität	Storage-Engpässe führen zu teuren Leerlaufzeiten bei Rechenressourcen	GPUDirect Storage über RDMA trägt dazu bei, dass GPU Cluster voll ausgelastet bleiben
Datenmobilität	Manuelle Skripte verzögern Pipelines	Automatisierte Bewegung zwischen den Ebenen über Airflow + XCP
Kostenkontrolle	Alle Daten auf teurem Flash-Speicher	FabricPool stuft selten genutzte Daten automatisch in kostengünstigen Objektspeicher herab
Mandantenfähigkeit	Gemeinsame Namensräume bergen das Risiko von Datenlecks	Dedizierte ONTAP SVMs gewährleisten eine strikte Mandantentrennung
Workload-Breite	Separate Stacks für Training und Inferenz	Eine einzige einheitliche Architektur, die den gesamten KI-Lebenszyklus abdeckt

[[1-6-the-business-case]]
 == 1.6 Der Business Case

GPU-Rechenleistung ist typischerweise der größte Kostenfaktor im Budget einer KI-Infrastruktur. Jede Stunde, die ein Cluster im Leerlauf auf Daten wartet, bedeutet einen direkten, messbaren finanziellen Verlust. NetApp stellt sicher, dass die richtigen Daten zur richtigen Zeit auf der richtigen Ebene verfügbar sind, automatisch, und ermöglicht so einen höheren KI-Durchsatz, niedrigere Gesamtbetriebskosten und eine Architektur, die sich ohne Neugestaltung vom Proof of Concept bis zur produktiven Unternehmensumgebung skalieren lässt.

[[1-7-key-benefits-at-a-glance]]
 == 1.7 Die wichtigsten Vorteile auf einen Blick

Nutzen	Beschreibung
Einheitliche Orchestrierung	Eine einzelne DAG umfasst ingest → archive
Flexibilität der Storage-Ebene	S3 oder LustreFS wird pro Lauf über die Konfiguration ausgewählt
Reduzierte Kosten für die Neuschulung	Inkrementelle Feinabstimmung über <code>partial_fit</code>

Nutzen	Beschreibung
Schnellere Wiederherstellung	Fortsetzen des checkpointbasierten Trainings
Vollständige Abstammung	Manifeste, effective-config Protokolle, Archivdatensätze
Keine Speicherbindung	Multiprotokoll: NFS, S3, Lustre
GPU-Auslastung	Gestuftter Storage versorgt die Rechenleistung, sodass Kosten durch inaktive GPUs vermieden werden.
Kostenoptimierte Aufbewahrung	FabricPool stuft selten genutzte Daten automatisch in Objektspeicher aus

2. Lösungsübersicht, Geschäftswert und Kundenvorteile

Karthikeyan Nagalingam, NetApp

[[2-1-solution-description]]

== 2.1 Lösungsbeschreibung

Die Pipeline ist als Airflow DAG `example_ai_pipeline_sklearn` implementiert und setzt sich aus den Aufgaben `ingestion`, `data_prep`, `xcp_preflight_source`, `xcp_copy_prep_to_training`, `model_training`, `fine_tuning`, `inferencing`, `checkpoint_model_training`, `route_manual_archive_stage` und `manual_archive` zusammen.

[[2-2-use-case-and-problem-statement]]

== 2.2 Anwendungsfall und Problemstellung

Kunden müssen tabellarische Regressions- und Textklassifizierungsmodelle mit Rohdaten trainieren und regelmäßig aktualisieren, die aus StorageGRID stammen. Die Pipeline bereitet diese Daten in einem ONTAP NAS Bucket vor und verwendet dann NetApp XCP, um sie je nach Auswahl des Ziels pro Lauf entweder an ONTAP S3 oder an LustreFS für das Training zu übermitteln.

[[2-3-target-customer-profile-and-industry-applicability]]

== 2.3 Zielkundenprofil und Anwendbarkeit in der Branche

Finanzdienstleistungen (Risiko-/Betrugsmodelle), Fertigung (vorausschauende Wartung), Gesundheitswesen (Klassifizierung/Triage), Einzelhandel (Nachfrageprognose), jedes Unternehmen, das ein geregeltes, wiederholbares ML-Training auf NetApp Infrastruktur benötigt.

[[2-4-solution-scope-and-boundaries]]

== 2.4 Lösungsumfang und Grenzen

Im Leistungsumfang enthalten: Ingestion-Gating, Datenaufbereitung (Python/Spark), XCP Datenmobilität (S3/LustreFS), scikit-learn-Training und inkrementelles Feintuning, Inferenz, Checkpointing, manuelle Archivierung. **Nicht im Leistungsumfang enthalten:** Echtzeit-Modellbereitstellungs-APIs, Streaming-Datenaufnahme, GPU-basierte Deep-Learning-Frameworks.

[[2-5-assumptions-and-prerequisites]]

== 2.5 Annahmen und Voraussetzungen

- StorageGRID S3-kompatibler Endpunkt, der von Airflow aus für den Zugriff auf Rohdaten und die

Archivierung erreichbar ist.

- ONTAP NAS bucket ist von Airflow aus für Schreibvorgänge vorbereiteter Daten erreichbar und über NFS für den XCP Host verfügbar.
- NetApp ONTAP S3-kompatibler Endpunkt, der von Airflow und XCP erreichbar ist, wenn ONTAP S3 als Trainingsziel ausgewählt ist.
- NetApp XCP ist installiert und über SSH erreichbar (Airflow Verbindung `ssh_default`).
- Der LustreFS Client ist auf dem XCP Host eingebunden, wenn LustreFS ausgewählt ist.
- Airflow 2.x mit `boto3`, `scikit-learn`; optionalen PySpark + Delta/Iceberg-Paketen.

[[2-6-business-drivers]]

== 2.6 Geschäftstreiber

Verkürzung der Zeit bis zum Modell, Senkung der Infrastrukturkosten durch inkrementelles Training, Eliminierung des Risikos einer einmaligen Skripterstellung.

[[2-7-value-proposition-summary]]

== 2.7 Zusammenfassung des Wertversprechens

StorageGRID, ONTAP NAS, ONTAP S3, LustreFS und XCP in Kombination mit Airflow Orchestrierung bieten eine speicherebenenbewusste, beobachtbare und ausfallsichere KI Pipeline.

[[2-8-stakeholder-benefits]]

== 2.8 Vorteile für Stakeholder

Stakeholder	Nutzen
Data Engineering	Deklarative, parametergesteuerte Vorbereitung; automatische Dateierkennung; Unterstützung für Spark/Delta/Iceberg
Data Science/ML	Inkrementelle Feinabstimmung; konsistente Artefaktquelle über alle Phasen hinweg; reproduzierbare Aufteilungen
IT-Betrieb	Laufzeitspeicherauswahl; Schutz-/effective-config-Protokolle; checkpoint-basierte Wiederherstellung
Compliance/Governance	Manifeste und Archivdatensätze gewährleisten eine prüfbare Nachvollziehbarkeit; konfigurierbare Aufbewahrungsphase

[[2-9-tco-considerations]]

== 2.9 TCO Überlegungen

Inkrementelles Fine-Tuning und Checkpoint Resume reduzieren die wiederkehrenden Rechenkosten; die Flexibilität der Storage-Ebene vermeidet eine Überbereitstellung von Hochleistungsspeicher für alle Workloads.

[[2-10-roi-considerations]]

== 2.10 ROI Überlegungen

Schnellere Modellaktualisierungszyklen, reduzierter manueller Wartungsaufwand, verkürzte Wiederherstellungszeit pro Ausfallereignis.

3. Architekturübersicht

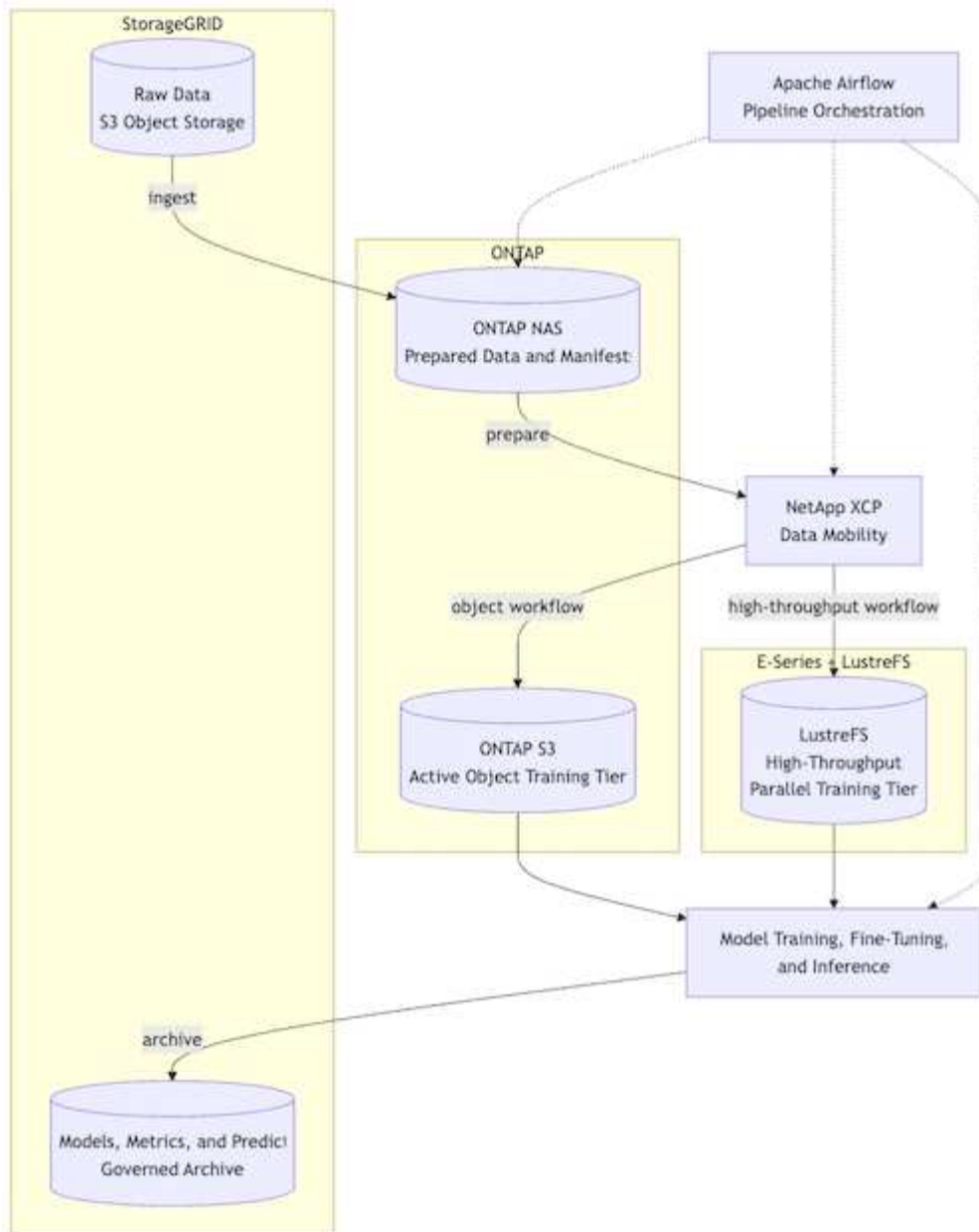
Karthikeyan Nagalingam, NetApp

Dieser Abschnitt beschreibt den logischen Workflow, die Interaktionen der Komponenten, die physische Bereitstellung und die Datenherkunft der validierten KI-Pipeline. Die Diagramme zeigen zusammen, wie Apache Airflow die Datenübertragung von StorageGRID über ONTAP NAS orchestriert, optional NetApp XCP verwendet, um Daten zur Modellausführung auf ONTAP S3 oder LustreFS zu platzieren, und die verwalteten Ergebnisse an den StorageGRID Archivspeicher zurückgibt.

[[3-1-high-level-three-tier-ai-storage-architecture]]

== 3.1 Hochrangige dreistufige KI-Speicherarchitektur

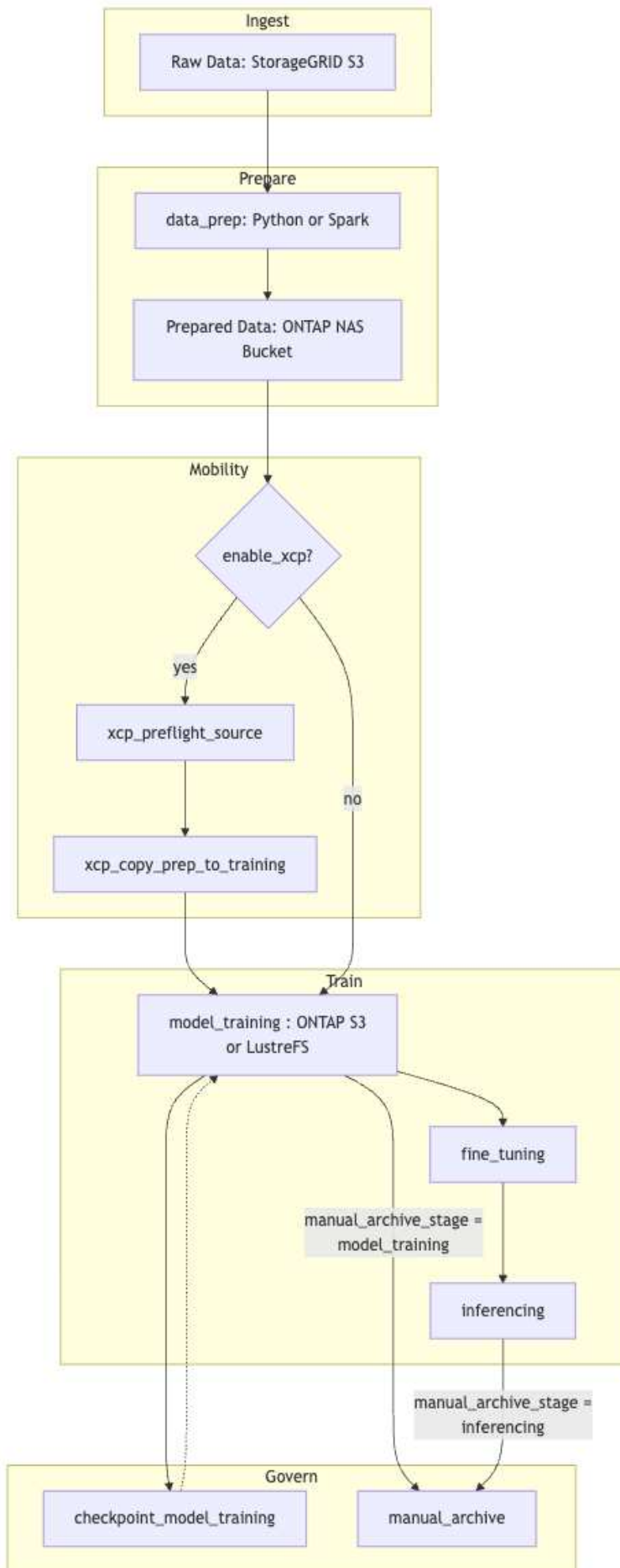
Die NetApp AI Speicherarchitektur platziert Daten auf der Speicherebene, die am besten zu der jeweiligen Lebenszyklusphase passt. StorageGRID wird für skalierbare, kostenoptimierte Rohdatenaufnahme und kontrollierte Archivierung verwendet. ONTAP NAS und ONTAP S3 werden für die aktive Datenaufbereitung, das Datenmobilität Staging und objektbasierte Trainingsworkflows verwendet. E-Series mit LustreFS wird verwendet, wenn ein hoher Durchsatz und parallele Training I/O erforderlich sind. Apache Airflow orchestriert den Lebenszyklus, während NetApp XCP aufbereitete Daten von der ONTAP NAS Ebene in die ausgewählte aktive Trainingsebene verschiebt.



[[3-2-high-level-solution-architecture]]

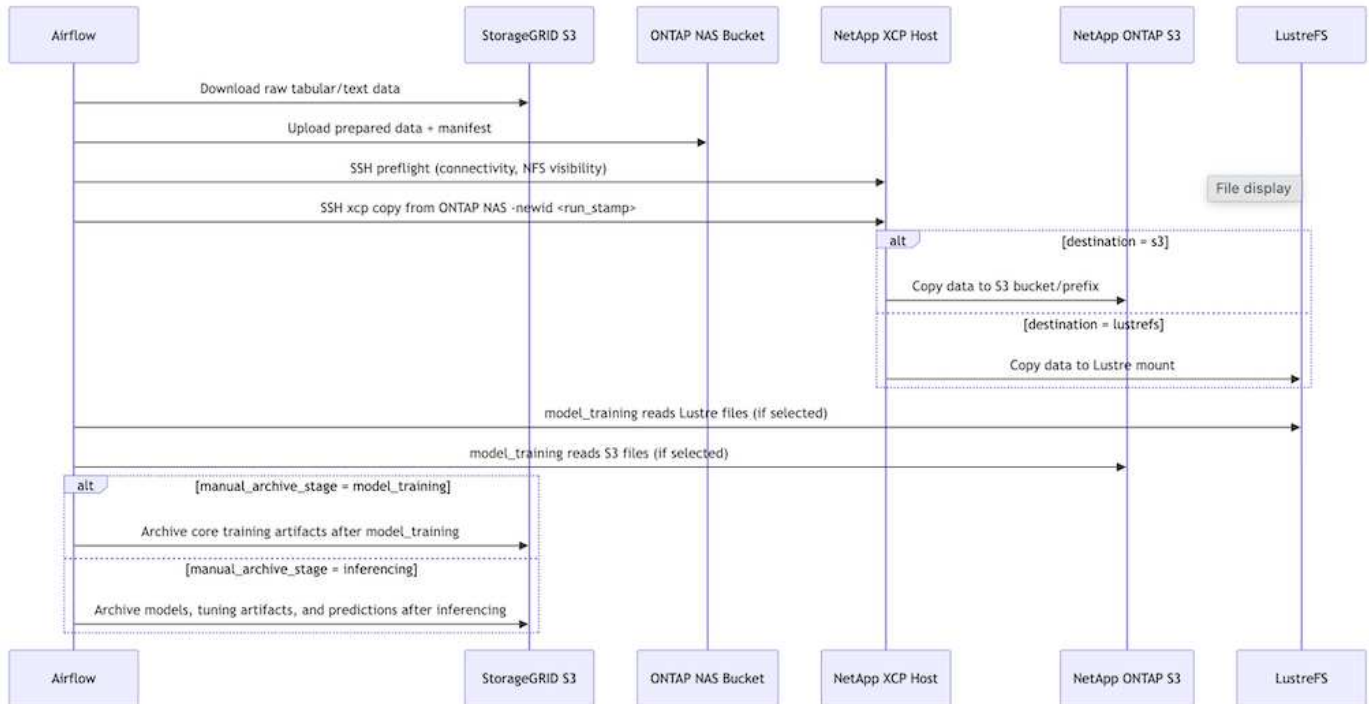
== 3.2 Übergeordnete Lösungsarchitektur

Dieses Workflow-Diagramm veranschaulicht den von Airflow gesteuerten Lebenszyklus von den StorageGRID Rohdaten über die Aufbereitung, optionale XCP-Datenmobilität, Modellausführung, Checkpointing und Archivierung. `enable\_xcp` Dabei wird festgelegt, ob aufbereitete Daten vor dem Training vom ONTAP NAS Bucket nach ONTAP S3 oder LustreFS kopiert werden. Die Archivierungszweige sind unabhängig: `manual\_archive\_stage=model\_training` Der eine archiviert die Baseline unmittelbar nach dem Training, während `manual\_archive\_stage=inferencing` der andere das vollständige Ergebnis nach der Generierung von Vorhersagen archiviert.



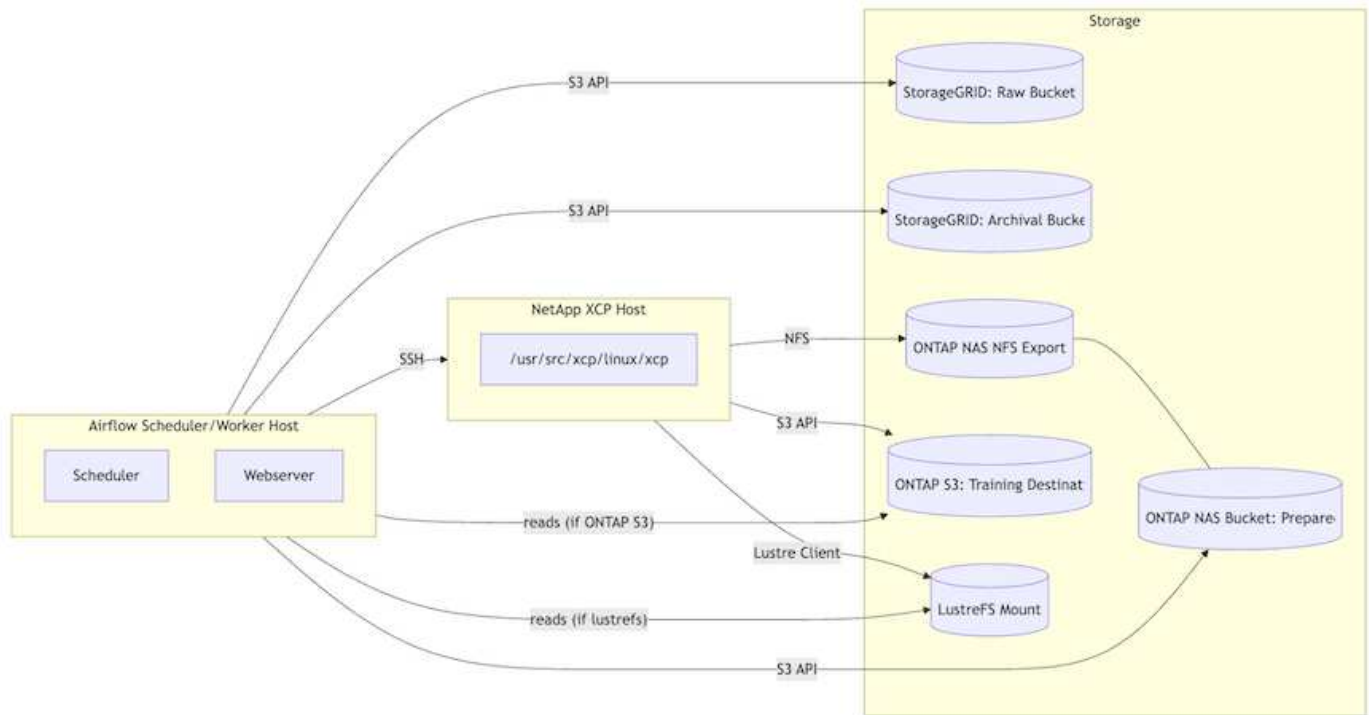
[[3-3-component-interaction-diagram]] == 3.3 Komponenteninteraktionsdiagramm

Dieses Sequenzdiagramm zeigt, welche Plattformkomponente welchen Datenaustausch durchführt. Airflow liest Rohdatenobjekte aus StorageGRID und schreibt aufbereitete Daten in den ONTAP NAS Bucket; anschließend ruft es XCP über SSH auf. XCP liest den ONTAP NAS NFS Export und schreibt die Daten in die konfigurierte Trainingsschicht. Der letzte Bedingungsblock zeigt, dass archivierte Inhalte in StorageGRID von der ausgewählten manuellen Archivierungsstufe abhängen.



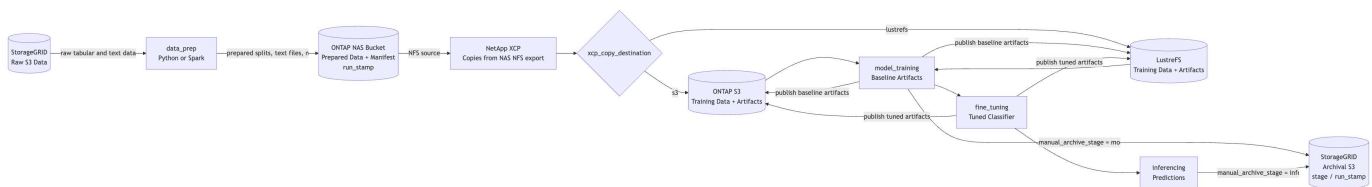
[[3-4-physical-deployment-architecture]] == 3.4 Physische / Bereitstellungsarchitektur

Dieses Bereitstellungsdiagramm zeigt die erforderlichen Hosts, Speicherendpunkte und Netzwerkschnittstellen. Der Airflow Host nutzt S3 APIs für StorageGRID, den ONTAP NAS Bucket und das ausgewählte ONTAP S3 Ziel und verwendet SSH, um den XCP Host zu steuern. Der XCP Host greift über NFS auf die vorbereitete Datenquelle zu und schreibt entweder in ONTAP S3 oder in den LustreFS Mount; wenn LustreFS ausgewählt ist, muss der Airflow Host diesen ebenfalls einbinden, um Trainingsdaten zu lesen und Artefakte zu materialisieren.



[[3-5-data-flow-overview]]
 == 3.5 Datenflussübersicht

Dieser kompakte Ablauf fasst die Herkunft von Daten und Artefakten zusammen. Jeder Durchlauf erhält eine eindeutige `run_stamp`ID`, die aufbereitete Daten, XCP-Ausgabe, Modellartefakte, Vorhersagen und archivierte Objekte derselben Ausführung zuordnet. Die Auswahl des XCP-Ziels steuert die Ebene für Trainingsdaten und Modellartefakte, während ``manual_archive_stage` festlegt, ob StorageGRID das Basistrainingspaket oder das vollständige Inferenzpaket empfängt.



4. NetApp, Technologiekomponenten von Drittanbietern und Open Source

Karthikeyan Nagalingam, NetApp

Komponente	Rolle in der Lösung
NetApp StorageGRID	S3-kompatible Ebenen für Rohdatenerfassung und Archivierung von Objektspeicher
NetApp ONTAP NAS Bucket / NFS	Ebene der vorbereiteten Daten; NFS-Quellexport wird vor der Datenübertragung durch XCP Preflight validiert

Komponente	Rolle in der Lösung
NetApp ONTAP S3	Optionales Ziel für XCP Trainingsdaten und Modellartefakte
NetApp XCP	Datenmobilitäts-Engine mit hohem Durchsatz; kopiert nach S3 oder LustreFS
NetApp FabricPool	Optionales Tiering für ONTAP NAS aufbereitete Daten und archivnahe Daten zur Kostenoptimierung
LustreFS (NetApp integriert)	Paralleles Hochleistungs-Dateisystem für die Trainingsschicht
NetApp Console (optional)	Erweiterte Überwachung/Beobachtbarkeit für ONTAP und XCP Ressourcen
ONTAP Snapshot/Backup	Zeitpunktbezogener Schutz für vorbereitete Daten in der ONTAP NAS-Ebene, unabhängig von der Pipeline

[[4-1-netapp-xcp-core-value]]
== 4.1 NetApp XCP - Kernwert

XCP ist die Mobilitätsschicht, die den Datenspeicherort vom Rechenstandort entkoppelt. Ein einziger Konfigurationsschalter (`xcp_copy_destination`) legt fest, ob aufbereitete Daten und Modellartefakte in NetApp ONTAP S3 oder in einer LustreFS Hochleistungsebene landen, wobei `model_training`, `fine_tuning`, und `inferencing` automatisch aus demselben Ziel lesen und in dasselbe Ziel schreiben.

[[4-2-third-party-and-open-source-technology-components]]
== 4.2 Technologiekomponenten von Drittanbietern und Open Source

Komponente	Zweck
Apache Airflow	DAG Orchestrierung, Wiederholungsversuche, XCom Task-Kommunikation, Terminplanung
Apache Spark	Optionale verteilte Transformations-Engine für <code>data_prep</code>
Delta Lake / Apache Iceberg	Optionale Lakehouse-Tabellenformate für vorbereitete tabellarische Daten
scikit-learn	<code>LinearRegression</code> (tabellarisch) und <code>SGDClassifier</code> (textlich, inkrementell)
Airbyte/Apache NiFi	Optionale Erfassungs-Connectors vor <code>data_prep</code>
boto3	AWS S3-kompatibles SDK für StorageGRID, ONTAP NAS Bucket und ONTAP S3 Interaktionen

5. Lösungsdesign und Details zur Storage Architektur

Karthikeyan Nagalingam, NetApp

[[5-1-design-principles]]
== 5.1 Gestaltungsprinzipien

Konfiguration statt Codeänderungen; explizite Phasengrenzen mit typisierten Verträgen; konsistentes Artefakt-Routing; Validierung mit frühzeitigem Abbruch bei umsetzbaren Fehlern; Neutralität der Storage-Ebene.

[[5-2-stage-design-summary]]

== 5.2 Zusammenfassung des Stufendesigns

Stufe	Designübersicht
Erfassung	<code>ingestion_tool</code> wählt <code>s3_direct</code> , <code>none</code> , <code>airbyte</code> oder <code>nifi</code> aus; gibt normalisierte Metadaten zurück
Datenaufbereitung	Erkennt/akzeptiert rohe tabellarische StorageGRID-Schlüssel, normalisiert Airbyte/einfache CSV-Dateien, erzeugt deterministische Trainings-/Validierungs-/Inferenz-Aufteilungen und schreibt ein mit einem Ausführungsstempel versehenes Präfix und Manifest der vorbereiteten Daten in den ONTAP NAS Bucket
Datenmobilität (XCP)	Kopiert vorbereitete Daten aus dem ONTAP NAS NFS-Export; <code>xcp_copy_destination</code> steuert den XCP Ziel- und Trainingslesepfad
Modelltraining	Training lokal oder vom XCP Ziel (S3/LustreFS); Erkennung in mehreren Formaten (CSV/Parquet/JSON)
Feinabstimmung	Materialisiert das Basismodell aus dem XCP-Ziel; <code>partial_fit</code> mit erweitertem Klassensatz; veröffentlicht das optimierte Modell erneut
Schlussfolgerung	Materialisiert angepasste Artefakte aus demselben XCP Ziel; konfigurierbarer Split-/Textbereich
Checkpoint/Fortsetzen	<code>write_stage_checkpoint</code> speichert den Abschlussdatensatz; <code>small_resume_guard</code> validiert/verwendet ihn bei nachfolgenden Ausführungen
Archivierung	<code>model_training</code> archiviert sofort die trainierten Kernmodelle; <code>inferencing</code> wartet auf Vorhersagen und archiviert den vollständigen Ergebnissatz; optionale Einbindung von XCP-Eingaben und Bereinigung des Rohordners

[[5-3-configuration-driven-philosophy]]

== 5.3 Konfigurationsgesteuerte Philosophie

Alle nicht geheimen Speicher-, Engine- und Verhaltensauswahlen sind `dag_run.conf` Parameter, der Wechsel zwischen S3 und LustreFS oder Python und Spark erfordert keine Codeänderung, während der von Airflow verwaltete Secrets-Speicher die erforderlichen Anmeldeinformationen auflöst.

[[5-4-storage-architecture-detail]]

== 5.4 Details zur Storage Architektur

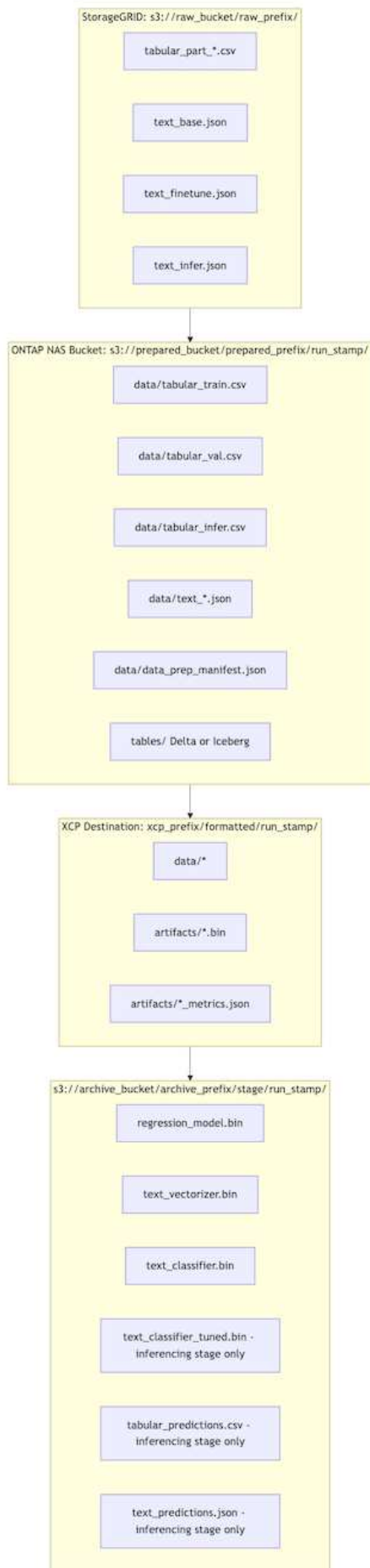
Die Speicherarchitektur unterteilt den KI-Lebenszyklus in speziell entwickelte Ebenen: StorageGRID speichert Rohdaten und Archivobjekte, der ONTAP NAS Bucket enthält aufbereitete und mit einem Ausführungsstempel

versehene Datensätze, und ONTAP S3 oder LustreFS stellt die ausgewählte aktive Trainingsebene bereit. Diese Trennung ermöglicht die unabhängige Verwaltung von Quelldaten, aufbereiteten Daten, Modellartefakten und archivierten Nachweisen, während die Herkunft durch die gemeinsame `run_stamp` erhalten bleibt.

[[5-4-1-storage-layout-diagrams]]
 === 5.4.1 Storage Layout-Diagramme

Stufe	Logischer Speicherpfad	Inhalte / Gespeicherte Artefakte
StorageGRID Raw-Tier	<code>s3://raw_bucket/raw_prefix/</code>	Tabellarische CSV-Teile (<code>tabular_part_*.csv</code>), Texteingaben (<code>text_base.json</code> , <code>text_finetune.json</code> , <code>text_infer.json</code>)
ONTAP NAS vorbereite	<code>s3://prepared_bucket/prepared_prefix/run_stamp/</code>	Formatierte Aufteilungen (<code>data/tabular_*.csv</code>), Textdateien, <code>data_prep_manifest.json</code> , Lakehouse Tabellen (<code>tables/Delta/Iceberg</code>)
Aktive Trainingsstufe (XCP Dest)	<code><xcp_prefix>/formatted/run_stamp/</code>	Kopierte Datenaufteilungen, trainierte Modellbinärdateien (<code>artifacts/*.bin</code>), Bewertungsmetriken (<code>artifacts/*_metrics.json</code>)
StorageGRID Archivierungsebene	<code>s3://archive_bucket/archive_prefix/stage/run_stamp/</code>	Phasenbezogene Basislinien- oder vollständige Modellartefakte, Vorhersagenachweise (<code>tabular_predictions.csv</code> , <code>text_predictions.json</code>)
Inter-Tier-Herkunft und -Ablauf	Rohdaten → Aufbereitet → Aktive Ebene → Archivierung	Durchgängige Datenübertragung und Artefaktverfolgung über alle Speicherebenen hinweg, verknüpft über die gemeinsame <code>run_stamp</code>

Dieses Diagramm bildet das logische Bucket- und Präfix-Layout für einen einzelnen Pipeline-Lauf über alle Storage Tiers hinweg ab:



[[5-4-2-storage-sizing-guidance]]

=== 5.4.2 Leitfaden zur Storage-Dimensionierung

Jede Speicherebene sollte entsprechend ihrer Rolle und Aufbewahrungsrichtlinie unabhängig dimensioniert werden. Das ONTAP NAS und das ausgewählte XCP Ziel müssen gleichzeitige aktive Ausführungen unterstützen, während die StorageGRID Kapazität in erster Linie durch Rohdaten und die Archivaufbewahrung bestimmt wird. Bei der Planung der maximalen Pipeline Parallelität sollten temporäre Arbeitskapazität und Wachstumsreserven berücksichtigt werden.

Stufe	Grundlage für die Größenbestimmung
StorageGRID Rohdaten-Bucket	Aufnahmevolumen × Aufbewahrungszeitraum
ONTAP NAS Bucket für vorbereitete Daten	Größe des vorbereiteten Datensatzes × Anzahl der beibehaltenen Ausführungen
XCP Ziel (S3 oder Lustre)	Maximale Größe des gleichzeitigen Trainingsdatensatzes + Modellartefakt-Overhead
Archiv-Bucket	Aufbewahrungsrichtlinie × Auswahl der Archivierungsphase (model_training = kleiner; inferencing = größer)

[[5-4-3-storage-performance-considerations]]

=== 5.4.3 Überlegungen zur Storage-Performance

Das XCP-Ziel wird pro Lauf ausgewählt `xcp_copy_destination`, sodass die Speicherleistung an die Arbeitslast angepasst wird, anstatt jede Arbeitslast auf eine einzige Ebene zu zwingen. LustreFS ist für Datensätze mit hoher Zeilenanzahl, viele gleichzeitige Leser oder E/A-intensives Training geeignet. ONTAP S3 ist für kostenbewusste Workloads mit elastischem Zugriff geeignet, deren Leistungsanforderungen kein paralleles Dateisystem erfordern. In beiden Fällen hält XCP Daten und Modellartefakte auf die ausgewählte Trainingsebene abgestimmt.

6. Datenmobilität mit NetApp XCP

Karthikeyan Nagalingam, NetApp

NetApp XCP stellt den kontrollierten Übertragungspfad zwischen der vorbereiteten Datenebene von ONTAP NAS und der aktiven Trainingsebene bereit. Airflow führt die Vorabprüfung durch und initiiert den Kopiervorgang über den XCP Host; `xcp_copy_destination` für jeden Durchlauf wird ONTAP S3 oder LustreFS ausgewählt. Dasselbe ausgewählte Ziel wird vom Modelltraining, der Feinabstimmung und der Inferenz verwendet, um Daten und Artefakte auf einer konsistenten Ebene zu halten.

[[6-1-why-netapp-xcp]]

== 6.1 Warum NetApp XCP

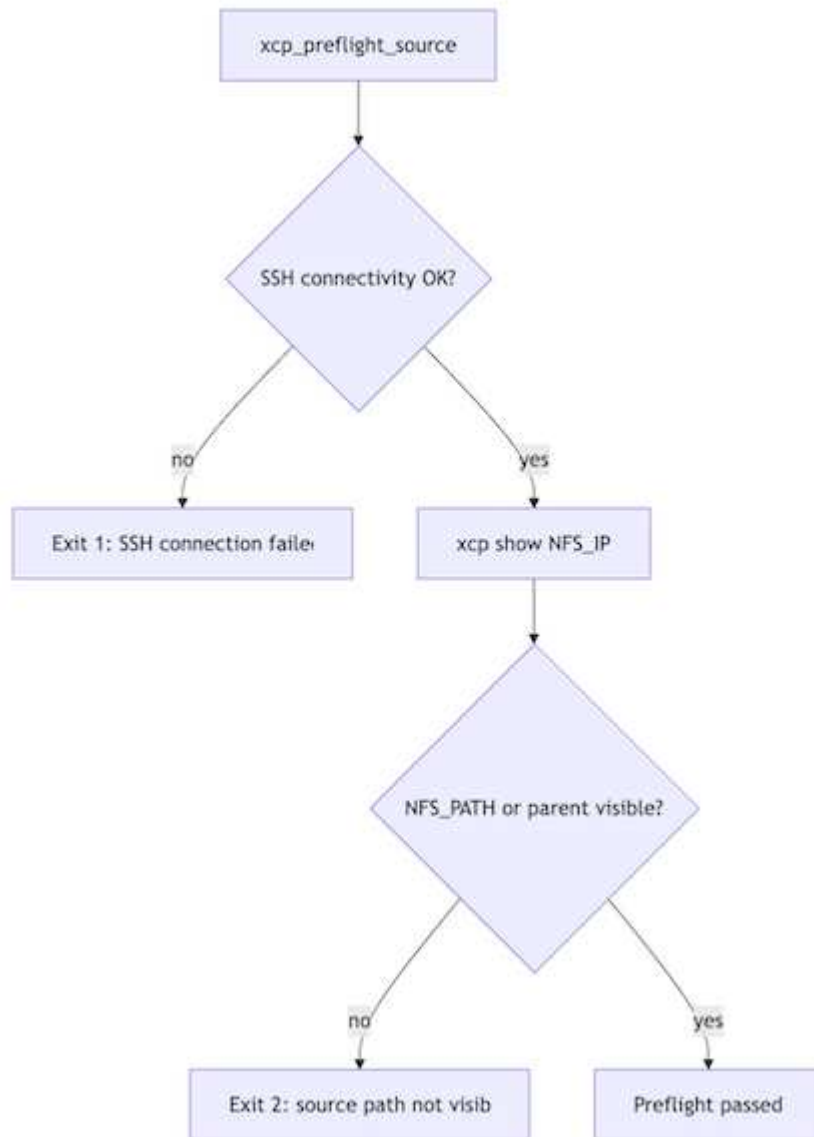
XCP ist eine parallele, Multithreading-fähige Kopier-Engine, die die von dieser Architektur verwendeten NFS-Quellen sowie ONTAP S3- und LustreFS-Ziele unterstützt. Die Option `-newid <run_stamp>` ordnet jede Übertragung der Pipeline-Ausführung zu und unterstützt so die Nachverfolgbarkeit über vorbereitete Daten, Trainingseingaben und Speicherorte von Modellartefakten hinweg.

[[6-2-xcp-preflight-validation-flow]]

== 6.2 XCP Preflight Validierungsablauf

Die Vorabprüfung wird vor einem XCP Kopiervorgang ausgeführt. Dabei wird überprüft, ob Airflow den XCP

Host über SSH erreichen kann und ob der konfigurierte ONTAP NAS NFS Quellpfad für XCP sichtbar ist. Die expliziten Abbruchbedingungen stoppen die Pipeline, bevor ein Kopiervorgang beginnt, wenn die Konnektivität oder die Sichtbarkeit der Quelle nicht korrekt ist.



[[6-3-xcp-copy-to-s3-command-template]]
== 6.3 XCP Kopieren nach S3 - Befehlsvorlage

Dieser Pfad ist zu verwenden, wenn `xcp_copy_destination=s3`. XCP liest die vorbereiteten, mit einem Laufstempel versehenen Daten aus dem ONTAP NAS NFS Export und schreibt sie in den konfigurierten ONTAP S3 Bucket und das zugehörige Präfix. Das benannte XCP S3 Profil stellt die Zielanmeldeinformationen auf dem XCP Host bereit; die Modellphasen erkennen dann sowohl Daten als auch Artefakte über `xcp_s3_training_prefix`.

```
xcp copy -newid <run_stamp> \  
-s3.profile <xcp_s3_profile> \  
-s3.endpoint <xcp_s3_endpoint> \  
<xcp_nfs_source_ip>:<xcp_nfs_source_path> \  

```

```
s3://<xcp_s3_bucket>/<xcp_s3_dest_prefix>
```

Beispiel: Der vorbereitete ONTAP NAS Datensatz wird in den ONTAP S3 Trainings-Bucket kopiert.

```
/usr/src/xcp/linux/xcp copy -newid 20260901_120000 \  
-s3.profile ontaps3 \  
-s3.endpoint http://10.63.150.161 \  
10.63.150.159:/prepnasbucket/example_ai_pipeline \  
s3://trainingbucket/example_ai_pipeline
```

Dieses Beispiel verwendet 20260901_120000 als die XCP Jobidentität. Die Trainingsaufgabe findet die kopierten Dateien unter dem konfigurierten xcp_s3_training_prefix, z. B. example_ai_pipeline/formatted/20260901_120000/data/.

[[6-4-xcp-copy-to-lustrefs-command-template]]
== 6.4 XCP Copy to LustreFS - Befehlsvorlage

Dieser Pfad wird verwendet, wenn xcp_copy_destination=lustrefs. XCP kopiert vorbereitete Daten von der eingebundenen ONTAP NAS Quelle zum eingebundenen LustreFS Ziel. Beide Einbindungen müssen auf dem XCP Host verfügbar sein, und LustreFS muss auch auf dem Airflow Worker eingebunden sein, der Modelltraining, Feinabstimmung und Inferenz durchführt.

```
findmnt -T <xcp_lustrefs_source_path> -o FSTYPE,TARGET -n | grep -E  
'^nfs[0-9]* ' # source mount type  
findmnt -T <xcp_lustrefs_dest_path> -o FSTYPE,TARGET -n | grep -E '^lustre  
' # destination mount type  
xcp copy -newid <run_stamp> \  
file://<xcp_lustrefs_source_path> \  
file://<xcp_lustrefs_dest_path>
```

Beispiel: Kopieren des vorbereiteten ONTAP NAS-Datensatzes auf das LustreFS-Trainingsmount

```
findmnt -T /prepnasbucket/example_ai_pipeline -o FSTYPE,TARGET -n | grep  
-E '^nfs[0-9]* '  
findmnt -T /mnt/lustre/client -o FSTYPE,TARGET -n | grep -E '^lustre '  
  
/usr/src/xcp/linux/xcp copy -newid 20260901_120000 \  
file:///prepnasbucket/example_ai_pipeline \  
file:///mnt/lustre/client
```

Die beiden findmnt-Befehle überprüfen, ob der Quellpfad über NFS und der Zielpfad über LustreFS auf dem XCP-Host eingebunden sind, bevor der Kopiervorgang beginnt. Mit `xcp\_lustrefs\_training\_subpath=example\_ai\_pipeline` liest training die mit einem Laufstempel versehenen Daten unter /mnt/lustre/client/example_ai_pipeline/formatted/20260901_120000/data/.

[[6-5-xcp-destination-decision-flow]]
 == 6.5 XCP Zielentscheidungsablauf

Dieses Entscheidungsdiagramm zeigt, wie ein Laufzeitparameter das Kopierziel und den Lesepfad für das Training bestimmt. Für das ONTAP S3 Objektziel wird `s3` ausgewählt und für das parallele Dateisystemziel `lustrefs`. Nachgelagerte Stufen verwenden automatisch die passende Quelle, sodass bei Änderungen der Speicherplatzierung keine Anpassungen am Modellcode erforderlich sind.



[[6-6-xcp-failure-handling-and-guardrails]]
 == 6.6 XCP Fehlerbehandlung und Schutzmaßnahmen

Diese Schutzmechanismen machen Konfigurations- und Infrastrukturfehler zum frühestmöglichen praktikablen Zeitpunkt erkennbar. Fehler bei der Vorabprüfung verhindern eine ungültige Übertragung, Dateisystemtypprüfungen validieren die ONTAP NAS und LustreFS Mounts, und die Validierung der S3-Anmeldeinformationen gilt nur, wenn ONTAP S3 als ausgewähltes XCP Ziel verwendet wird.

Zustand	Guard-Verhalten
Nicht unterstützter <code>xcp_copy_destination</code> Wert	Die Pipeline lehnt den Wert ab, bevor Daten verschoben werden.
LustreFS Mount nicht vorhanden	Die Validierung des Dateisystemtyps ist fehlgeschlagen; der Kopiervorgang wurde vor der Datenverschiebung angehalten.
SSH nicht erreichbar	Preflight wird mit Code 1 beendet, bevor ein Kopiervorgang erfolgt.
NFS Pfad nicht sichtbar	Preflight wird mit Code 2 beendet, bevor ein Kopiervorgang erfolgt.
Fehlende XCP S3 Anmeldeinformationen (nur S3 Modus)	<code>model_training/artifact sync</code> löst explizite Fehler aus <code>ValueError</code>

7. Bereitstellungsarchitektur

Karthikeyan Nagalingam, NetApp

Dieser Abschnitt definiert die Infrastrukturplatzierung, die Softwareabhängigkeiten und die Netzwerkkonnektivität, die für den Betrieb der Pipeline erforderlich sind. Die Bereitstellung trennt die Airflow Orchestrierung vom XCP Datenmobilitätshost und verbindet beide Komponenten mit StorageGRID, ONTAP NAS und dem ausgewählten ONTAP S3 oder LustreFS Trainingsziel.

[[7-1-reference-infrastructure-requirements]]
 == 7.1 Anforderungen an die Referenzinfrastruktur

Die folgenden Komponenten bilden den minimalen funktionalen Umfang der Bereitstellung. Der Airflow Host ist für die ausgewählte Vorbereitungs-Engine und die gleichzeitige Aufgabenlast zu dimensionieren; der XCP Host ist so zu platzieren, dass er ohne unnötige Netzwerk-Hops auf den ONTAP NAS NFS Export und das ausgewählte Ziel zugreifen kann. Der LustreFS Modus erfordert kompatible Mounts sowohl auf dem XCP Host als auch auf dem Airflow Worker.

Komponente	Anforderung
Airflow Host	CPU/Speicher auf die Transformationslast von Spark oder Python dimensioniert
XCP-Host	Netzwerkzugriff auf NFS/Lustre und NetApp ONTAP S3 Endpunkte
LustreFS Clients	Auf dem XCP-Host und dem Airflow-Host eingebunden, wenn diese als Ziel ausgewählt sind

[[7-1-1-reference-validation-environment]]

== 7.1.1 Referenzvalidierungsumgebung

Die folgende Umgebung diene als repräsentatives Validierungsprofil für einen einzelnen Knoten. Sie ist ein Ausgangspunkt für die Funktions- und Leistungsbewertung und keine Empfehlung zur Dimensionierung für den Produktiveinsatz; Kunden müssen Rechenleistung, Netzwerk, Speicherkapazität und Schutz entsprechend ihrem Datenvolumen sowie den Anforderungen an Parallelität, Aufbewahrung, Wiederherstellung und Servicelevel dimensionieren.

Schicht	Referenzhardware / Referenzsoftware
Compute und Netzwerk	Ein Server mit 256 GB RAM, 64 CPU-Kernen und 10 GbE Anbindung
ONTAP Active Storage	Ein NetApp A800 System mit 48 x 1,8 TB SSDs
Objektspeicher	Eine NetApp StorageGRID SG5864 Appliance
Hochdurchsatz-Trainingsspeicher	Ein NetApp E2812 System mit LustreFS
Plattformsoftware	Kubernetes, Apache Airflow, Apache Airbyte, Single-Node Apache Spark, NetApp XCP und LustreFS

[[7-2-software-version-matrix]]

== 7.2 Softwareversionsmatrix

Diese Matrix identifiziert die Laufzeitsoftware, die von der validierten Pipeline verwendet wird. Die Airflow- und Python-Umgebungen müssen mit den bereitgestellten Providern und Paketen kompatibel sein. Spark, Delta Lake und Iceberg sind optional und werden nur benötigt, wenn `data_prep` der Spark-Transformationspfad oder ein entsprechendes Tabellenformat verwendet wird.

Software	Version/Hinweise
Apache Airflow	2.x
Python	3,11
boto3	Neueste stabile Version
scikit-learn	Neueste stabile Version

Software	Version/Hinweise
PySpark (optional)	Kompatibel mit Delta 3.2.0 / Iceberg 1.5.2 Laufzeitpaketen
NetApp XCP	Auf einem entfernten Host installiert, z. B. /usr/src/xcp/linux/xcp

[[7-3-network-requirements]]

== 7.3 Netzwerkanforderungen

Diese Netzwerkflüsse müssen durch Routing-, Firewall- und Namensauflösungsrichtlinien zugelassen werden. HTTPS wird für jeden S3-kompatiblen Endpunkt in der Produktion empfohlen; dabei ist der konfigurierte benutzerdefinierte Port zu verwenden, wenn ein Endpunkt nicht den Standard HTTPS Port verwendet. Die Lustre Client-Konnektivität ist gemäß der bereitgestellten LustreFS Implementierung zu bestätigen.

Ablauf	Protokoll/Port
Airflow → ONTAP S3	HTTPS/HTTP (443/80 oder benutzerdefiniert)
Airflow → XCP Host	SSH (22)
XCP Host → NFS Quelle	NFS (2049)
XCP-Host → LustreFS	Lustre Client Ports
XCP Host → ONTAP S3 (S3 Modus)	HTTPS/HTTP

[[7-4-deployment-notes]]

== 7.4 Hinweise zur Bereitstellung

Abschnitt / Thema	Leitfaden / Betriebspraxis
Einrichten der SSH Verbindung	Die Airflow-Verbindung <code>ssh_default</code> zum XCP Host über Port 22 mit eingeschränkten SSH-Schlüsselberechtigungen konfigurieren (<code>600</code>).
Service-Lifecycle-Management	Dient <code>tools/airflow_ctl.sh</code> zur Verwaltung des lokalen Airflow-Schedulers und des Lebenszyklus des Webserver während der Bereitstellung und Wartung.
Aufbewahrung von Zugangsdaten und Geheimnissen	Anmeldeinformationen, API-Token und Zugriffsschlüssel sollten in Airflow Connections, Variablen oder einem Secrets-Backend statt inline in <code>dag_run.conf</code> gespeichert werden.
Beobachtbarkeit von Infrastruktur und Aufgaben	Die unabhängige Überwachung des Scheduler Heartbeat und der Aufgabenausführung ermöglicht die Unterscheidung von Problemen mit der Orchestrierungsverfügbarkeit und DAG-Fehlern.

[[7-5-installation-and-prerequisites]]

== 7.5 Installation und Voraussetzungen

Dieser Unterabschnitt beschreibt die grundlegenden Installationsschritte, die zum Erstellen, Konfigurieren und Ausführen der NetApp AI validierten Pipeline in einer neuen Umgebung erforderlich sind. Er ist für Operatoren,

Architekten und Ingenieure gedacht, die den Airflow-Arbeitsbereich einrichten, bevor DAGs ausgeführt werden.

[[7-5-1-prerequisites]]

=== 7.5.1 Voraussetzungen

Vor der Installation der Lösung ist zu bestätigen, dass der Zielhost die folgenden Mindestanforderungen erfüllt:

- Linux Betriebssystem, vorzugsweise Ubuntu 22.04/24.04 oder RHEL 8+.
- Python 3.11 mit `pip`, `venv` und verfügbarem Build-Tooling.
- Git ist auf dem Host für den Quellcodeabruf und die Versionsverwaltung installiert.
- Apache Airflow 2.x wird in einer dedizierten virtuellen Python-Umgebung eingesetzt.
- SSH-Zugriff vom Airflow Host auf den NetApp XCP Host.
- Netzwerkverbindung vom Airflow Host zu StorageGRID, ONTAP NAS und dem ausgewählten ONTAP S3 oder LustreFS Ziel.
- S3-kompatibler Endpunktzugriff für Rohdaten, aufbereitete Daten und Archivspeicher.
- Optional: Java-Laufzeitumgebung und Spark 3.x, falls der ``spark`` Vorbereitungspfad verwendet wird.
- Optional: Delta Lake und Iceberg Laufzeitpakete, wenn `table_format=delta` oder `table_format=iceberg` ausgewählt ist.
- Optional: Lustre Client Einbindung, falls `xcp_copy_destination=lustrefs` verwendet wird.

[[7-5-2-software-package-acquisition-and-repository-access]]

=== 7.5.2 Beschaffung von Softwarepaketen und Zugriff auf das Repository

Um das Softwarepaket, die Automatisierungs-DAGs, die Bereitstellungsskripte und die Validierungsartefakte für diese Lösung zu erhalten, kontaktieren Sie das NetApp AI & Data Mobility Engineering-Team unter ng-data-mobility-in-ai-pipeline@netapp.com.

Sobald der Zugriff gewährt wurde, wird der Projektquellcode aus GitHub in das Zielarbeitsverzeichnis heruntergeladen oder geklont:

```
# Example GitHub clone
mkdir -p /opt/netapp-ai
cd /opt/netapp-ai

git clone https://github.com/<your-org>/<your-repo>.git
cd <your-repo>

git checkout main
ls -la
```

Wenn Sie das Repository bereits unter `/opt/netapp-ai/<your-repo>` geklont haben, setzen Sie den Vorgang in diesem Arbeitsverzeichnis fort, anstatt eine weitere Kopie herunterzuladen.

[[7-5-3-create-the-python-environment]]

=== 7.5.3 Erstellen der Python-Umgebung

Es ist eine dedizierte virtuelle Umgebung zu erstellen und die Basisabhängigkeiten für Airflow und die Pipeline

sind zu installieren.

```
export REPO_ROOT=/opt/netapp-ai/<your-repo>
cd "$REPO_ROOT"
python3 -m venv "$REPO_ROOT/.venv"
source "$REPO_ROOT/.venv/bin/activate"

python -m pip install --upgrade pip setuptools wheel
pip install "apache-airflow>=2.8,<3.0" boto3 scikit-learn
```

Für die Spark-basierte Vorbereitung und die Lakehouse-Tabellenformate sind die optionalen Pakete nach Bedarf zu installieren:

```
pip install pyspark==3.5.*
pip install delta-spark==3.2.0
pip install apache-iceberg==1.5.2
```

Es sollten genau die Paketversionen verwendet werden, die von Ihrer Spark-Laufzeitumgebung und Clustertopologie unterstützt werden. Inkompatible Kombinationen aus Spark, Delta und Iceberg sollten nicht gemischt werden.

[[7-5-4-configure-airflow-and-the-repository]]
=== 7.5.4 Airflow und das Repository konfigurieren

Vom Stammverzeichnis des geklonten Repositories aus wird die Airflow Metadatenbank initialisiert, und es wird überprüft, ob die DAG-Dateien für Airflow sichtbar sind.

```
export AIRFLOW_HOME="$REPO_ROOT/.airflow"
export AIRFLOW__CORE__DAGS_FOLDER="$REPO_ROOT"
export AIRFLOW__CORE__LOAD_EXAMPLES=False

airflow db init
airflow dags list | grep -E
"example_ai_pipeline|example_ai_pipeline_sklearn"
```

Der Arbeitsbereich enthält die Airflow Konfigurationsdatei und die für die validierte Pipeline benötigten DAGs. Wenn das Repository ein Hilfsskript für die lokale Lebenszyklusverwaltung enthält, wird dieses zum Starten des Schedulers und des Webservers verwendet, anstatt Airflow manuell aufzurufen.

```
# Start the Airflow scheduler and webserver with the deployment's service
manager.
# or, if using the standard commands:
# airflow scheduler
# airflow webserver
```

[[7-5-5-required-connectivity-and-secret-configuration]]

=== 7.5.5 Erforderliche Konnektivität und Konfiguration von Secrets

Vor dem Ausführen der Pipeline ist zu bestätigen, dass die folgenden Ressourcen vom Airflow Host aus erreichbar sind:

- StorageGRID Rohdaten S3 Endpunkt.
- ONTAP NAS prepared-data bucket Endpunkt.
- ONTAP S3 Zielendpunkt, wenn `xcp_copy_destination=s3`.
- LustreFS einhängen, wenn `xcp_copy_destination=lustrefs`.
- XCP Host über SSH unter Verwendung von ``ssh_default`` oder der konfigurierten SSH Verbindung.

Anmeldeinformationen werden in Airflow Connections, Variables oder einem Secrets-Backend gespeichert, statt sie in den Shell-Verlauf oder den DAG-Code einzubetten. Die erforderlichen Zugriffsschlüssel, Endpunkte und Profilzuordnungen werden vor dem Auslösen eines Pipeline-Laufs festgelegt.

[[7-5-6-verify-the-installation]]

=== 7.5.6 Überprüfung der Installation

Eine erfolgreiche Installation ist bestätigt, wenn der Airflow Scheduler und die DAGs ausgeführt werden und die Beispielpipeline ohne Konfigurationsfehler aufgelistet und ausgelöst werden kann.

```
airflow dags list
airflow tasks list example_ai_pipeline_sklearn
```

Wenn die Installation korrekt ist, sollte der DAG in Airflow vorhanden und für einen `CONF_JSON`-gesteuerten Lauf mit dem Trigger-Skript im Stammverzeichnis des Repositorys bereit sein.

```
./trigger_and_wait_ai_pipeline_sklearn.sh
```

Zu diesem Zeitpunkt ist die Umgebung für die Konfigurationsbeispiele in Abschnitt 8 und die im Betriebshandbuch beschriebenen Bereitstellungsszenarien bereit.

8. Konfigurationsreferenz

Karthikeyan Nagalingam, NetApp

Die Konfigurationsreferenz beschreibt die Laufzeitparameter, die die Datenerfassung, Datenaufbereitung, XCP Datenmobilität, das Training, Checkpointing und die Archivierung steuern. Der ONTAP NAS Bucket wird in erster Linie als Staging-Schicht für aufbereitete Daten und XCP verwendet, um Supportfähigkeit und Portabilität zu gewährleisten. Im Echtzeit-Produktionsbetrieb können dieselben aufbereiteten Daten auch über direkte RDMA-fähige Zugriffspfade genutzt werden, wenn dies besser den Anforderungen an Workload und Latenz entspricht.

Das gesamte Laufzeitverhalten wird über `dag_run.conf` bereitgestellt, sodass derselbe DAG verschiedene Quellsysteme, Transformations-Engines, Speicherziele, Wiederherstellungsrichtlinien und

Archivierungsbereiche bedienen kann. Es sollten nur die für den ausgewählten Workflow erforderlichen Parametergruppen konfiguriert, Anmeldeinformationen in Airflow Connections oder einem Secrets-Backend gespeichert und die Beispiele als Vorlagen für Nicht-Produktionsumgebungen statt als Anmeldeinformationswerte verwendet werden.

[[8-1-ingestion-parameters]]

== 8.1 Aufnahmeparameter

Diese Einstellungen dienen zur Auswahl und Konfiguration des vorgelagerten Datenerfassungsmechanismus. `s3_direct` Dabei werden die in der Datenaufbereitungsgruppe konfigurierten StorageGRID Rohdatenobjekte verwendet; Airbyte und NiFi Einstellungen sind nur erforderlich, wenn die jeweiligen Erfassungstools ausgewählt sind.

Parameter	Standard	Erforderlich	Beschreibung	Beispiel
<code>ingestion_tool</code>	<code>s3_direct</code>	Nein	Auswahl des Aufnahmemodus	<code>"s3_direct"</code>
<code>airbyte_api_url</code>	Keine	Nur Airbyte	Airbyte API Endpunkt	<code>"http://airbyte:8000"</code>
<code>airbyte_connection_id</code>	Keine	Nur Airbyte	Airbyte Verbindungs-ID	<code>"connection-123"</code>
<code>airbyte_api_token</code>	Keine	Optional	Bearer Token	<code>"<TOKEN>"</code>
<code>airbyte_timeout_secs</code>	30	Nein	Airbyte Zeitüberschreitung	60
<code>nifi_api_url</code>	Keine	NiFi nur	NiFi API-Endpunkt	<code>"http://nifi:8080"</code>
<code>nifi_process_group_id</code>	Keine	NiFi nur	NiFi Prozessgruppe	<code>"abc123"</code>
<code>nifi_timeout_secs</code>	30	Nein	NiFi Zeitüberschreitung	60

[[8-2-data-preparation-parameters]]

== 8.2 Parameter zur Datenaufbereitung

Diese Einstellungen steuern den Zugriff auf Rohdaten, die Ausgabe aufbereiteter Daten, die Eingabeerkennung und das Transformationsverhalten. Die `s3_raw_*` Parameter beziehen sich auf StorageGRID, während die code-compatible `s3_formatted_*` Parameter den ONTAP NAS Bucket identifizieren, in den aufbereitete, mit einem Laufzeitstempel versehene Daten und Manifeste geschrieben werden.

Unformatierter StorageGRID / Prepared-Data ONTAP NAS-Bucket:

Separate Endpunkte und Anmeldeinformationen werden konfiguriert, wenn StorageGRID und der ONTAP NAS Bucket unterschiedliche S3-kompatible Dienste oder Zugriffsrichtlinien verwenden. Die generischen Fallback-Schlüssel gelten nur, wenn keine spezifischere Einstellung für Rohdaten oder aufbereitete Daten vorhanden ist.

Parameter	Standard	Beschreibung	Beispiel
s3_raw_bucket	ai-raw-data	StorageGRID Rohdaten-Eingabe-Bucket	"bucket1"
s3_raw_prefix	example_ai_pipeline/raw	StorageGRID Rohdatenpräfix	"example_ai_pipeline/raw"
s3_raw_endpoint_url	Fallback-Kette	StorageGRID S3 Endpunkt	"http://10.63.150.62:10444"
s3_raw_access_key_id	Fallback-Kette	Unformatierter Zugriffsschlüssel	"<KEY>"
s3_raw_secret_access_key	Fallback-Kette	Rohschlüssel	"<SECRET>"
s3_raw_region	aws_region	Rohregion	"us-east-1"
s3_formatted_bucket	ai-formatted-data	ONTAP NAS Ausgabebucket für vorbereitete Daten	"prepnasbucket"
s3_formatted_prefix	example_ai_pipeline/formatted	ONTAP NAS Prepared-Data-Präfix	"example_ai_pipeline/formatted"
s3_formatted_endpoint_url	Fallback-Kette	ONTAP NAS Bucket S3 Endpunkt	"http://10.63.150.159"
s3_formatted_access_key_id	Fallback-Kette	ONTAP NAS Bucket-Zugriffsschlüssel	"<KEY>"
s3_formatted_secret_access_key	Fallback-Kette	ONTAP NAS Bucket Geheimschlüssel	"<SECRET>"

Allgemeiner Ausweichweg: s3_endpoint_url, s3_access_key_id, s3_secret_access_key, s3_session_token, s3_verify, aws_region, aws_access_key_id, aws_secret_access_key, aws_session_token.

Eingabeauswahl:

Für die Verarbeitung bekannter CSV-Eingaben werden explizite Objektschlüsseleinstellungen verwendet, andernfalls ermittelt die Aufgabe geeignete tabellarische Objekte unter dem konfigurierten StorageGRID Rohpräfix.

Parameter	Standard	Beschreibung	Beispiel
s3_tabular_object_keys	Automatische Erkennung	Explizite Liste der CSV-Schlüssel	["raw/demo/a.csv"]
s3_tabular_object_key	Keine	Einzelner expliziter CSV-Schlüssel	"raw/demo/a.csv"

Transformation:

Wählen Sie den Python-Pfad für eine ressourcenschonende lokale Vorbereitung oder Spark für eine verteilte Vorbereitung. Sampling und der Seed gelten reproduzierbar für die generierten Trainings-, Validierungs- und

Inferenzaufteilungen.

Parameter	Standard	Beschreibung	Beispiel
transformation_engine	python	python oder spark	"spark"
spark_required	false	Ausfall statt Fallback	true
table_format	none	none/delta/iceberg	"delta"
table_format_required	false	Fehler, wenn das Format nicht verfügbar ist	true
sample_count	alle Zeilen	Zeilenlimit (≤ 0 = alle)	14400
seed	7	Reproduzierbarer Shuffle Seed	11

Spark:

Diese Einstellungen gelten nur für `transformation_engine=spark`. Sie steuern die Platzierung von Spark-Prozessen, die Ressourcenzuweisung, Zeitlimits und optionale Laufzeitabhängigkeiten von Delta Lake oder Iceberg.

Parameter	Standard	Beschreibung	Beispiel
spark_submit_bin	spark-submit	Spark Binärdatei	"/opt/spark/bin/spark-submit"
spark_master	local[*]	Spark Master	"local[*]"
spark_driver_memory	4g	Treiberspeicher	"8g"
spark_executor_memory	4g	Executor Speicher	"8g"
spark_timeout_secs	900/7200	Zeitüberschreitung (0 = deaktiviert)	0
spark_delta_packages	Delta 3.2.0	Delta Runtime-Paket	"io.delta:delta-spark_2.12:3.2.0"
spark_iceberg_packages	Iceberg 1.5.2	Iceberg Laufzeitpaket	"org.apache.iceberg:iceberg-spark-runtime-3.5_2.12:1.5.2"
spark_iceberg_catalog	local	Name des Iceberg Katalogs	"local"

[[8-3-xcp-and-training-destination-parameters]]
 == 8.3 XCP und Trainingszielparameter

Diese Gruppe dient, `enable\_xcp=true` wenn die ONTAP NAS NFS-Quelle validiert, der XCP Host aufgerufen und die aktive Trainingsebene ausgewählt werden sollen. `xcp\_copy\_destination` Sie ist der steuernde

Schalter: Er wählt entweder die ONTAP S3-spezifischen oder die LustreFS-spezifischen Einstellungen aus, und nachgelagerte Modellstufen verwenden dieselbe ausgewählte Ebene.

Parameter	Standard	Erforderlich	Beschreibung	Beispiel
enable_xcp	false	Ja für XCP	Aktiviert den XCP-Zweig	true
xcp_copy_destination	s3	Nein	s3 oder lustrefs	"lustrefs"
xcp_nfs_source_ip	10.63.150.159	XCP Vorabprüfung	NFS-Server-IP	"10.63.150.159"
xcp_nfs_source_path	/prepnasbucket/example_ai_pipeline	XCP Vorabprüfung	NFS Exportpfad	"/prepnasbucket/example_ai_pipeline"
ssh_user	root	Nein	XCP host SSH Benutzer	"root"
ssh_host	10.63.150.178	Nein	XCP Hostadresse	"10.63.150.178"

S3 Ziel (erforderlich, wenn xcp_copy_destination=s3):

Diese Einstellungen sind nur für den ONTAP S3 Trainingspfad erforderlich. Die Profil- oder direkten Anmeldeinformationen ermöglichen XCP und der Modellartefakt-Synchronisierungslogik den Zugriff auf den ausgewählten ONTAP S3 Bucket und das Präfix.

Parameter	Standard	Beschreibung	Beispiel
xcp_s3_endpoint	Standard	XCP S3-Endpunkt	"http://10.63.150.161"
xcp_s3_bucket	Keine	XCP Ziel Bucket	"trainingbucket"
xcp_s3_dest_prefix	example_ai_pipeline	Zielpräfix	"example_ai_pipeline"
xcp_s3_training_prefix	xcp_s3_dest_prefix	Präfix für die Trainingserkennung	"example_ai_pipeline"
xcp_s3_profile	ontaps3	Benanntes Anmeldeinformationsprofil	"ontaps3"
xcp_s3_profiles	Keine	Profilübersicht	{"ontaps3": {...}}
xcp_s3_access_key_id	Keine	Direkter Schlüsselfallback	"<KEY>"
xcp_s3_secret_access_key	Keine	Direkter geheimer Fallback	"<SECRET>"
xcp_s3_region	aws_region	XCP S3 Region	"us-east-1"

LustreFS Ziel (erforderlich, wenn xcp_copy_destination=lustrefs):

Diese Einstellungen gelten nur für den LustreFS Trainingspfad. Quelle und Ziel müssen auf dem XCP Host eingebunden und erreichbar sein; das Ziel muss außerdem für den Airflow Worker erreichbar sein, der die Modellphasen ausführt.

Parameter	Standard	Beschreibung	Beispiel
xcp_lustrefs_source_path	/prepnasbucket/example_ai_pipeline	XCP Kopierquelle	"/prepnasbucket/example_ai_pipeline"
xcp_lustrefs_dest_path	/mnt/lustre/client	LustreFS Mount-Ziel	"/mnt/lustre/client"
xcp_lustrefs_training_subpath	xcp_s3_dest_prefix	Unterverzeichnis „Training“ unter Mount	"example_ai_pipeline"
xcp_lustrefs_newid	data_prep Ausführungsstempel	XCP Jobkennung	"20260901_120000"

Verhalten der Textquelle:

Diese Optionen steuern, wie die drei Textdatensätze nach der XCP Mobilität gefunden werden. Explizite Schlüssel überschreiben die Erkennung; der lokale Fallback kann deaktiviert werden, damit die Texteingaben vom ausgewählten XCP-Ziel stammen.

Parameter	Standard	Beschreibung	Beispiel
xcp_text_allow_local_fallback	true	Lokalen Text-Fallback zulassen	false
xcp_text_base_key	Automatisch	Explizite Basistextposition	".../text_base.json"
xcp_text_finetune_key	Automatisch	Explizite Position des Feinabstimmungstextes	".../text_finetune.json"
xcp_text_infer_key	Automatisch	Explizite Schlussfolgerung-Textposition	".../text_infer.json"

[[8-4-model-training-parameters]]
 == 8.4 Parameter für das Modelltraining

Diese Einstellungen wählen die lokale oder über XCP bereitgestellte Quelle aus, begrenzen das Trainingsvolumen und bewahren eine reproduzierbare Datenreihenfolge. Wenn XCP aktiviert ist, liest das Training die ausgewählte ONTAP S3 oder LustreFS Ebene und veröffentlicht die Basisartefakte wieder an demselben Zielort.

Parameter	Standard	Beschreibung	Beispiel
enable_xcp	false	Auswahl von XCP oder lokaler Quelle	true
xcp_copy_destination	s3	Die Quell-Tier wird ausgewählt	"lustrefs"

Parameter	Standard	Beschreibung	Beispiel
sample_count	alle Zeilen	Limit für Trainingsstichproben	14400
seed	7	Reproduzierbares Mischen	11

Produziert: regression_model.bin, text_vectorizer.bin, text_classifier.bin, train_metrics.json.

[[8-5-fine-tuning-parameters]]
 == 8.5 Feinabstimmungsparameter

Die Feinabstimmung materialisiert die Basistextartefakte aus dem ausgewählten XCP-Ziel, wendet inkrementelles Lernen auf den Feinabstimmungsdatensatz an und veröffentlicht den optimierten Klassifikator und seine Metriken erneut. Die XCP-Einstellungen sollten mit dem Modelltraining konsistent bleiben, um die Herkunft der Artefakte zu gewährleisten.

Parameter	Standard	Beschreibung	Beispiel
enable_xcp	false	Ermöglicht das Materialisieren/Veröffentlichen von Artefakten	true
xcp_copy_destination	s3	Auswahl der Artefaktquelle/des Artefaktziels	"lustrefs"

Verbraucht: text_vectorizer.bin, text_classifier.bin, text_finetune.json. Produziert: text_classifier_tuned.bin, fine_tune_metrics.json.

[[8-6-inference-parameters]]
 == 8.6 Inferenzparameter

Diese Flags steuern den Umfang der Bewertung, nachdem das optimierte Modell aus der ausgewählten Trainingsstufe generiert wurde. Eine der Optionen ist zu aktivieren, wenn Validierungs- oder Geschäftsanforderungen Ausgaben erfordern, die über die standardmäßige Inferenzaufteilung und die Inferenztexteingabe hinausgehen.

Parameter	Standard	Beschreibung	Beispiel
infer_all_splits	false	Bewerten von Train/Val/Infer-Aufteilungen	true
infer_all_text	false	Base-/Feinabstimmungs-/Inferenztext bewerten	true

Verbraucht: regression_model.bin, text_vectorizer.bin, text_classifier_tuned.bin.
 Erzeugt: tabular_predictions.csv, text_predictions.json.

[[8-7-checkpoint-parameters]]
 == 8.7 Checkpoint-Parameter

Checkpointing ist auf `model_training` beschränkt. Diese Einstellungen legen fest, ob ein Datensatz zum Trainingsabschluss erstellt wird, wo dieser gespeichert wird, wie mit Upload-Fehlern umgegangen wird und ob ein späterer Durchlauf ein gültiges abgeschlossenes Trainingsergebnis überprüft oder wiederverwendet.

Parameter	Standard	Beschreibung	Beispiel
<code>checkpoint_enabled</code>	<code>true</code>	Ermöglicht die Erstellung von Prüfpunktdateien	<code>true</code>
<code>checkpoint_store</code>	<code>local</code>	<code>local</code> oder <code>formatted_s3</code>	<code>"formatted_s3"</code>
<code>checkpoint_upload_fail_mode</code>	<code>warn</code>	<code>warn</code> oder <code>fail</code>	<code>"fail"</code>
<code>training_checkpoint_reuse_mode</code>	<code>off</code>	<code>off/verify_only/resume_if_exists</code>	<code>"resume_if_exists"</code>
<code>checkpoint_reuse_mode</code>	<code>off</code>	Abwärtskompatibler Alias	<code>"verify_only"</code>

[[8-8-manual-archive-parameters]]
 == 8.8 Manuelle Archivparameter

Diese Gruppe dient zum Aktivieren der StorageGRID Archivierung, zum Angeben des Ziel-Bucket und der Zugangsdaten sowie zum Auswählen des Archivierungspunkts. Der ausgewählte `manual_archive_stage` steuert sowohl den Zeitpunkt der Archivierung als auch, ob das Archiv nur Trainingsartefakte der Baseline oder den vollständigen Ergebnissatz der Inferenz enthält.

Parameter	Standard	Erforderlich	Beschreibung	Beispiel
<code>manual_archive_enabled</code>	<code>false</code>	Nein	Ermöglicht die Archivierung	<code>true</code>
<code>manual_archive_stage</code>	<code>inferencing</code>	Nein	Archivpunkt und Artefaktsatz: <code>model_training</code> für Kerntrainingsartefakte oder <code>inferencing</code> für den vollständigen Ergebnissatz	<code>"model_training"</code>
<code>manual_archive_bucket</code>	<code>archivalbucket</code>	Ja, wenn aktiviert	Archive Bucket	<code>"archivalbucket"</code>
<code>manual_archive_prefix</code>	<code>ai-models</code>	Nein	Archivpräfix	<code>"ai-models"</code>
<code>manual_archive_endpoint</code>	<code>Standard</code>	Nur benutzerdefiniert	Archive S3 Endpunkt	<code>"http://10.63.150.62:10444"</code>

Parameter	Standard	Erforderlich	Beschreibung	Beispiel
manual_archive_profile	sgdlocal	Empfohlen	Anmeldeinformati onsprofil	"sgdlocal"
manual_archive_profiles	Keine	Optional	Profilübersicht	{"sgdlocal": {...}}
manual_archive_include_xcp_inputs	false	Nein	XCP-Eingaben in Archiv einbinden	true
manual_archive_cleanup_raw	false	Nein	Entfernen des lokalen Rohdatenordners nach dem Upload	true

Aliase: fabricpool_archive_* für alle manual_archive_* Parameter.

[[8-8-1-manual-archive-stage-decision]]

=== 8.8.1 Entscheidung über die manuelle Archivstufe

Die route_manual_archive_stage Branch Task liest manual_archive_stage einmal pro DAG Lauf. Der ausgewählte Wert bestimmt sowohl, wann die Archivierung ausgeführt wird, als auch, welche Artefakte in den StorageGRID Archivierungs Bucket hochgeladen werden. Archivobjekte werden unter s3://<manual_archive_bucket>/<manual_archive_prefix>/<stage>/<run_stamp>/ gespeichert.

`manual_archive_stage` Wert	Archivierungszeitpunkte	Artefakte archiviert	Empfohlene Verwendung
model_training	Unmittelbar danach model_training; es wird nicht auf Feinabstimmung oder Inferenz gewartet	regression_model.bi n, text_vectorizer.bin, text_classifier.bin, train_metrics.json	Ein reproduzierbares Basismodell beibehalten, das Archivvolumen minimieren oder einen Kontrollpunkt vor späteren Phasen aufbewahren
inferencing (Standard)	Nach fine_tuning und inferencing abgeschlossen	Alle model_training Artefakte plus text_classifier_tun ed.bin, fine_tune_metrics.j son, tabular_predictions .csv und text_predictions.js on	Vollständige Geschäftsergebnisse und Prognosenachweise für einen Modelllauf beibehalten

Für beide Auswahlmöglichkeiten wird manual_archive_include_xcp_inputs=true festgelegt, dass lokal materialisierte XCP-Trainingseingaben hinzugefügt werden, sofern vorhanden.

manual_archive_enabled=false wird so festgelegt, dass die Archivierungsaufgabe übersprungen wird, ohne den Trainings- und Inferenz-Workflow zu ändern.

[[8-9-complete-sample-configuration]]

== 8.9 Vollständige Beispielkonfiguration

Dieser Katalog enthält Startkonfigurationen für die wichtigsten Bereitstellungs- und Testpfade. Allen Beispielen liegt zugrunde, dass sich Rohdaten in StorageGRID befinden und aufbereitete Daten in den ONTAP NAS Bucket geschrieben werden. Bucket-Namen, Endpunktadressen, Dateisystempfade und Profilnamen sind durch Werte aus der Zielumgebung zu ersetzen. Produktionszugriffsschlüssel oder Geheimnisse dürfen nicht in CONF_JSON abgelegt werden; die Konfiguration erfolgt stattdessen über Airflow Connections, Variables oder ein Secrets Backend.

[[8-9-1-xcp-to-lustrefs-with-spark-and-delta-lake]]

=== 8.9.1 XCP zu LustreFS mit Spark und Delta Lake

Dieses Beispiel mit vollem Durchsatz wird für die verteilte Vorbereitung und die Hochleistungsmodell-E/A von LustreFS verwendet. Es verarbeitet alle geeigneten tabellarischen Dateien unter `s3_raw_prefix`, schreibt Delta-Tabellen während der Vorbereitung, speichert den Trainings-Checkpoint im ONTAP NAS prepared-data bucket und archiviert die vollständige Inferenz-Ergebnismenge.

```
{
  "ingestion_tool": "s3_direct",
  "s3_raw_bucket": "bucket1",
  "s3_raw_prefix": "example_ai_pipeline/raw",
  "s3_formatted_bucket": "prepnasbucket",
  "s3_formatted_prefix": "example_ai_pipeline/formatted",

  "enable_xcp": true,
  "xcp_copy_destination": "lustrefs",
  "xcp_nfs_source_ip": "10.63.150.159",
  "xcp_nfs_source_path": "/prepnasbucket/example_ai_pipeline",
  "xcp_lustrefs_source_path": "/prepnasbucket/example_ai_pipeline",
  "xcp_lustrefs_dest_path": "/mnt/lustre/client",
  "xcp_lustrefs_training_subpath": "example_ai_pipeline",
  "xcp_text_allow_local_fallback": false,

  "transformation_engine": "spark",
  "spark_required": true,
  "table_format": "delta",
  "table_format_required": true,
  "spark_driver_memory": "8g",
  "spark_executor_memory": "8g",
  "spark_timeout_secs": 0,
  "sample_count": 14400,
  "seed": 11,

  "infer_all_splits": true,
  "infer_all_text": true,

  "checkpoint_enabled": true,
  "checkpoint_store": "formatted_s3",
```

```

"training_checkpoint_reuse_mode": "off",

"manual_archive_enabled": true,
"manual_archive_stage": "inferencing",
"manual_archive_bucket": "archivalbucket",
"manual_archive_prefix": "ai-models",
"manual_archive_profile": "sgdlocal",
"manual_archive_cleanup_raw": true
}

```

[[8-9-2-no-xcp-with-python-preparation]]

=== 8.9.2 Keine XCP Vorbereitung mit Python

Diese Basiskonfiguration wird für einen ressourcenschonenden Funktionslauf verwendet. Die Vorbereitung nutzt die lokale Python Engine, und die nachfolgenden Phasen verwenden die lokalen Pfade der Pipeline für vorbereitete Daten und Artefakte; XCP, Spark, Tabellenformate, Checkpoint-Wiederverwendung und Archivierung sind deaktiviert.

```

{
  "ingestion_tool": "s3_direct",
  "s3_raw_bucket": "bucket1",
  "s3_raw_prefix": "example_ai_pipeline/raw",
  "s3_formatted_bucket": "prepnasbucket",
  "s3_formatted_prefix": "example_ai_pipeline/formatted",
  "enable_xcp": false,
  "transformation_engine": "python",
  "table_format": "none",
  "sample_count": 200,
  "seed": 11,
  "checkpoint_enabled": false,
  "manual_archive_enabled": false
}

```

[[8-9-3-python-preparation-with-two-explicit-input-files]]

=== 8.9.3 Python Vorbereitung mit zwei expliziten Eingabedateien

Dieser gezielte Test dient zur Validierung der Pipeline anhand von zwei bekannten StorageGRID CSV-Objekten. `s3_tabular_object_keys` Die automatische Erkennung tabellarischer Dateien wird deaktiviert; die Texteingaben verwenden weiterhin ihre erwarteten Speicherorte unter dem Rohdaten-Präfix.

```

{
  "ingestion_tool": "s3_direct",
  "s3_raw_bucket": "bucket1",
  "s3_raw_prefix": "example_ai_pipeline/raw",
  "s3_tabular_object_keys": [
    "example_ai_pipeline/raw/tabular_part_01.csv",

```

```

        "example_ai_pipeline/raw/tabular_part_02.csv"
    ],
    "s3_formatted_bucket": "prepnasbucket",
    "s3_formatted_prefix": "example_ai_pipeline/formatted",
    "enable_xcp": false,
    "transformation_engine": "python",
    "table_format": "none",
    "sample_count": 200,
    "seed": 11,
    "manual_archive_enabled": false
}

```

[[8-9-4-python-preparation-with-automatic-all-file-discovery]]

=== 8.9.4 Python Vorbereitung mit automatischer Erkennung aller Dateien

Wenn `s3_tabular_object_keys` und `s3_tabular_object_key` weggelassen werden, werden alle geeigneten tabellarischen Objekte unter dem StorageGRID Rohdaten-Präfix verarbeitet. Dieses Beispiel eignet sich für einen Funktions- oder Skalierungstest für alle Dateien mithilfe der Python Vorbereitung Engine.

```

{
    "s3_raw_bucket": "bucket1",
    "s3_raw_prefix": "example_ai_pipeline/raw",
    "s3_formatted_bucket": "prepnasbucket",
    "s3_formatted_prefix": "example_ai_pipeline/formatted",
    "enable_xcp": false,
    "transformation_engine": "python",
    "table_format": "none",
    "sample_count": 0,
    "seed": 11,
    "manual_archive_enabled": false
}

```

`sample_count: 0` bedeutet, dass bei der Datenaufbereitung alle verfügbaren Zeilen verwendet werden. Für einen begrenzten Testlauf ist ein positiver Wert festzulegen.

[[8-9-5-xcp-to-ontap-s3-with-python-preparation]]

=== 8.9.5 XCP zu ONTAP S3 mit Python Vorbereitung

Diese Konfiguration wird verwendet, wenn vorbereitete Daten vom ONTAP NAS NFS-Export in einen ONTAP S3 Training Bucket kopiert werden müssen. Modelltraining, Feinabstimmung und Inferenz erzeugen und veröffentlichen Artefakte über dasselbe ONTAP S3-Ziel.

```

{
    "s3_raw_bucket": "bucket1",
    "s3_raw_prefix": "example_ai_pipeline/raw",
    "s3_formatted_bucket": "prepnasbucket",

```

```

"s3_formatted_prefix": "example_ai_pipeline/formatted",
"enable_xcp": true,
"xcp_copy_destination": "s3",
"xcp_nfs_source_ip": "10.63.150.159",
"xcp_nfs_source_path": "/prepnasbucket/example_ai_pipeline",
"xcp_s3_endpoint": "https://ontap-s3.example.com",
"xcp_s3_bucket": "trainingbucket",
"xcp_s3_dest_prefix": "example_ai_pipeline",
"xcp_s3_training_prefix": "example_ai_pipeline",
"xcp_s3_profile": "ontaps3",
"transformation_engine": "python",
"table_format": "none",
"sample_count": 14400,
"seed": 11,
"checkpoint_enabled": true,
"checkpoint_store": "formatted_s3",
"manual_archive_enabled": true,
"manual_archive_stage": "model_training",
"manual_archive_bucket": "archivalbucket",
"manual_archive_profile": "storagegrid-archive"
}

```

[[8-9-6-xcp-to-ontap-s3-with-spark-and-iceberg]]

=== 8.9.6 XCP zu ONTAP S3 mit Spark und Iceberg

Diese Konfiguration wird für die verteilte Vorbereitung mit Iceberg-Tabellenausgabe und ONTAP S3 als aktivem Trainingsziel verwendet. Dabei werden `spark_required` und `table_format_required` auf `true` gesetzt, wenn ein Lauf fehlschlagen muss, anstatt auf eine Alternative zurückzugreifen, falls Spark oder Iceberg nicht verfügbar sind.

```

{
  "s3_raw_bucket": "bucket1",
  "s3_raw_prefix": "example_ai_pipeline/raw",
  "s3_formatted_bucket": "prepnasbucket",
  "s3_formatted_prefix": "example_ai_pipeline/formatted",
  "enable_xcp": true,
  "xcp_copy_destination": "s3",
  "xcp_nfs_source_ip": "10.63.150.159",
  "xcp_nfs_source_path": "/prepnasbucket/example_ai_pipeline",
  "xcp_s3_bucket": "trainingbucket",
  "xcp_s3_dest_prefix": "example_ai_pipeline",
  "xcp_s3_profile": "ontaps3",
  "transformation_engine": "spark",
  "spark_required": true,
  "table_format": "iceberg",
  "table_format_required": true,
}

```

```

"spark_driver_memory": "8g",
"spark_executor_memory": "8g",
"spark_timeout_secs": 0,
"sample_count": 0,
"seed": 11,
"manual_archive_enabled": true,
"manual_archive_stage": "inferencing",
"manual_archive_bucket": "archivalbucket",
"manual_archive_profile": "storagegrid-archive"
}

```

[[8-9-7-scenario-selection-summary]]

=== 8.9.7 Zusammenfassung der Szenarioauswahl

Szenario	XCP Ziel	Vorbereitungs modul	Tabellenformat	Eingabebereich	Archivierungsp hase
8.9.1	LustreFS	Funke	Delta	Alle gefundenen Dateien	inferencing
8.9.2	Keine	Python	Keine	Alle gefundenen Dateien	Deaktiviert
8.9.3	Keine	Python	Keine	Zwei explizite Dateien	Deaktiviert
8.9.4	Keine	Python	Keine	Alle gefundenen Dateien	Deaktiviert
8.9.5	ONTAP S3	Python	Keine	Alle gefundenen Dateien	model_traini ng
8.9.6	ONTAP S3	Funke	Iceberg	Alle gefundenen Dateien	inferencing

[[8-9-8-execution-walkthroughs]]

=== 8.9.8 Walkthroughs zur Ausführung

Die Beispiele werden aus dem Airflow workspace ausgeführt. Das trigger script generiert eine eindeutige `manual__<UTC timestamp> run ID`, wartet auf den Abschluss und gibt bei einem fehlgeschlagenen oder mit Zeitüberschreitung beendeten Lauf einen Exit-Code ungleich null zurück. Die folgenden vollständigen Konfigurationen verwenden dasselbe `CONF_JSON="$ (python3 - <<'PY' ...)"` Muster wie die Betriebsbeispiele. Die referenzierten Airflow-managed connections, XCP profiles und archive profiles werden vor der Ausführung in Ihrem secrets backend konfiguriert, und `CONF_JSON` dies bleibt auf Bucket-, Endpunkt-, Pfad- und Profilkennungen ohne Secrets beschränkt.

Der gültige Tabellenformatname lautet `iceberg`. `icerberg` ist kein unterstützter Wert.

Alle Dateien, Spark, Iceberg, XCP zu ONTAP S3, Checkpoint zu prepared-data S3, Archiv nach der Inferenz:

```
CONF_JSON="$ (python3 - <<'PY'
```

```

import json

print(json.dumps({
    "ingestion_tool": "s3_direct",
    "enable_xcp": True,
    "xcp_copy_destination": "s3",
    "xcp_text_allow_local_fallback": True,

    "s3_raw_bucket": "bucket1",
    "s3_raw_prefix": "example_ai_pipeline/raw",
    "s3_raw_endpoint_url": "https://storagegrid.example.com",
    "s3_raw_region": "us-east-1",

    "s3_formatted_bucket": "prepnasbucket",
    "s3_formatted_prefix": "example_ai_pipeline/formatted",
    "s3_formatted_endpoint_url": "https://ontap-nas.example.com",
    "s3_formatted_region": "us-east-1",

    "xcp_nfs_source_ip": "10.63.150.159",
    "xcp_nfs_source_path": "/prepnasbucket/example_ai_pipeline",
    "xcp_s3_endpoint": "https://ontap-s3.example.com",
    "xcp_s3_bucket": "trainingbucket",
    "xcp_s3_dest_prefix": "example_ai_pipeline",
    "xcp_s3_training_prefix": "example_ai_pipeline",
    "xcp_s3_profile": "ontaps3",

    "transformation_engine": "spark",
    "spark_required": True,
    "table_format": "iceberg",
    "table_format_required": True,
    "sample_count": 0,
    "seed": 11,
    "spark_driver_memory": "8g",
    "spark_executor_memory": "8g",
    "spark_timeout_secs": 0,
    "infer_all_splits": True,
    "infer_all_text": True,

    "checkpoint_enabled": True,
    "checkpoint_store": "formatted_s3",
    "manual_archive_enabled": True,
    "manual_archive_stage": "inferencing",
    "manual_archive_bucket": "archivalbucket",
    "manual_archive_prefix": "ai-models",
    "manual_archive_profile": "sgdlocal",
    "manual_archive_endpoint": "https://storagegrid.example.com",

```

```

        "manual_archive_cleanup_raw": True,
        "manual_archive_include_xcp_inputs": True,
    }, separators=(",", ":"))
PY
) "
CONF_JSON="$CONF_JSON" AIRFLOW_BIN=./.venv/bin/airflow INGESTION_TOOL
=s3_direct \
    ./trigger_and_wait_ai_pipeline_sklearn.sh

```

Alle Dateien, Spark, Iceberg, XCP nach LustreFS, Checkpoint in prepared-data S3, Archiv nach Inferenz:

```

CONF_JSON="$(python3 - <<'PY'
import json

print(json.dumps({
    "ingestion_tool": "s3_direct",
    "enable_xcp": True,
    "xcp_copy_destination": "lustrefs",
    "ssh_user": "root",
    "ssh_host": "10.63.150.178",
    "xcp_text_allow_local_fallback": True,

    "s3_raw_bucket": "bucket1",
    "s3_raw_prefix": "example_ai_pipeline/raw",
    "s3_raw_endpoint_url": "https://storagegrid.example.com",
    "s3_raw_region": "us-east-1",

    "s3_formatted_bucket": "prepnasbucket",
    "s3_formatted_prefix": "example_ai_pipeline/formatted",
    "s3_formatted_endpoint_url": "https://ontap-nas.example.com",
    "s3_formatted_region": "us-east-1",

    "xcp_nfs_source_ip": "10.63.150.159",
    "xcp_nfs_source_path": "/prepnasbucket/example_ai_pipeline",
    "xcp_lustrefs_source_path": "/prepnasbucket/example_ai_pipeline",
    "xcp_lustrefs_dest_path": "/mnt/lustre/client",
    "xcp_lustrefs_training_subpath": "example_ai_pipeline",

    "transformation_engine": "spark",
    "spark_required": True,
    "table_format": "iceberg",
    "table_format_required": True,
    "sample_count": 0,
    "seed": 11,
    "spark_driver_memory": "8g",

```

```

"spark_executor_memory": "8g",
"spark_timeout_secs": 0,
"infer_all_splits": True,
"infer_all_text": True,

"checkpoint_enabled": True,
"checkpoint_store": "formatted_s3",
"manual_archive_enabled": True,
"manual_archive_stage": "inferencing",
"manual_archive_bucket": "archivalbucket",
"manual_archive_prefix": "ai-models",
"manual_archive_profile": "sgdlocal",
"manual_archive_endpoint": "https://storagegrid.example.com",
"manual_archive_cleanup_raw": True,
"manual_archive_include_xcp_inputs": True,
}, separators=(",", ":"))
PY
) "
AIRFLOW_BIN=./.venv/bin/airflow INGESTION_TOOL=s3_direct \
./trigger_and_wait_ai_pipeline_sklearn.sh

```

Für den ONTAP S3 Modus wird nur der nicht geheime `xcp_s3_profile`Name`` in ``CONF_JSON`` übergeben, und die entsprechenden Anmeldeinformationen werden vor dem Lauf auf dem XCP Host oder über den von Airflow verwalteten Secret Storage definiert. Für den LustreFS Modus wird die `xcp_s3_profiles`Zuordnung`` absichtlich ausgelassen, da die Trainingsroute ONTAP S3 Anmeldeinformationen nicht auflöst. In beiden Beispielen werden die Anmeldeinformationen für StorageGRID Rohdaten, ONTAP NAS vorbereitete Daten und das Archiv über den von Airflow verwalteten Secret Storage und nicht als Inline JSON definiert. Das Weglassen von `s3_tabular_object_keys`` und das Festlegen von `sample_count`` auf 0 fordert die automatische Erkennung aller geeigneten Tabellendateien sowie die Verwendung aller verfügbaren Zeilen an.

Python, kein XCP, Test mit zwei Dateien:

```

CONF_JSON='{ "s3_raw_bucket": "bucket1", "s3_raw_prefix": "example_ai_pipeline
/raw", "s3_tabular_object_keys": ["example_ai_pipeline/raw/tabular_part_01.c
sv", "example_ai_pipeline/raw/tabular_part_02.csv"], "s3_formatted_bucket": "
prepnasbucket", "s3_formatted_prefix": "example_ai_pipeline/formatted", "enab
le_xcp": false, "transformation_engine": "python", "table_format": "none", "samp
le_count": 200, "manual_archive_enabled": false}' \
./trigger_and_wait_ai_pipeline_sklearn.sh

```

Spark, Delta Lake, XCP zu LustreFS:

```

CONF_JSON='{ "enable_xcp": true, "xcp_copy_destination": "lustrefs", "xcp_nfs_s
ource_ip": "10.63.150.159", "xcp_nfs_source_path": "/prepnasbucket/example_ai
_pipeline", "xcp_lustrefs_source_path": "/prepnasbucket/example_ai_pipeline"

```

```
, "xcp_lustrefs_dest_path": "/mnt/lustre/client", "xcp_lustrefs_training_subp  
ath": "example_ai_pipeline", "transformation_engine": "spark", "spark_required  
": true, "table_format": "delta", "table_format_required": true, "sample_count":  
14400, "manual_archive_enabled": true, "manual_archive_stage": "inferencing"}'  
\  
./trigger_and_wait_ai_pipeline_sklearn.sh
```

Spark, Iceberg, XCP zu ONTAP S3:

```
CONF_JSON='{ "enable_xcp": true, "xcp_copy_destination": "s3", "xcp_nfs_source_  
ip": "10.63.150.159", "xcp_nfs_source_path": "/prepnasbucket/example_ai_pipel  
ine", "xcp_s3_bucket": "trainingbucket", "xcp_s3_dest_prefix": "example_ai_pip  
eline", "xcp_s3_profile": "ontaps3", "transformation_engine": "spark", "spark_r  
equired": true, "table_format": "iceberg", "table_format_required": true, "sampl  
e_count": 0, "manual_archive_enabled": true, "manual_archive_stage": "inferenci  
ng"}' \  
./trigger_and_wait_ai_pipeline_sklearn.sh
```

Vor einem XCP Lauf ist zu prüfen, ob der konfigurierte ONTAP NAS NFS Pfad und die Erreichbarkeit des Ziels vom XCP Host aus gegeben sind. Nach einem erfolgreichen Lauf sind die Aufgabenprotokolle für [data_prep] effective_config, [xcp_copy] effective_config, [model_training] effective_config und [manual_archive] zu prüfen, um die ausgewählte Engine, den Eingabebereich, das Ziel und die Archivierungsphase zu bestätigen.

9. Betriebshandbuch

Karthikeyan Nagalingam, NetApp

Dieser Leitfaden behandelt die routinemäßige Ausführung, Überwachung, Störungstriage, Wiederherstellung des Trainings, Archivprüfung, Kapazitätsoptimierung und Datensicherung. Er ist zusammen mit den Konfigurationsszenarien in Abschnitt 8 zu verwenden, damit dieselbe Pipeline über die Trainingsmodi lokal, ONTAP S3 und LustreFS hinweg konsistent betrieben wird.

[[9-1-execution-walkthrough]]
== 9.1 Ablauf der Ausführung

Einen Lauf über `trigger_and_wait_ai_pipeline_sklearn.sh` mit einer `CONF_JSON` Nutzlast auslösen. Das Skript übermittelt einen eindeutig benannten manuellen Lauf, überwacht den Status bis zum Erreichen eines Endzustands und gibt, sofern verfügbar, lokale Protokollenden für fehlgeschlagene Aufgaben aus. Jede primäre Aufgabe erzeugt beim Start einen `effective_config` Datensatz, sodass Bediener die ausgewertete Konfiguration überprüfen können, anstatt sich nur auf die übermittelte Nutzlast zu verlassen.

[[9-2-monitoring-and-observability]]
== 9.2 Überwachung und Beobachtbarkeit

Der DAG-Status kann in der Airflow-Benutzeroberfläche oder -CLI überwacht und mit der Lauf-ID und `run_stamp` korreliert werden. Die Guard-Logs sind grep-freundlich: [data_prep] effective_config,

[model_training] effective_config, [xcp_copy] effective_config, und [checkpoint] resume_summary. Diese Einträge können überprüft werden, um die Transformations-Engine, das ausgewählte XCP-Ziel, die Trainingsquelle und die Entscheidung zur Checkpoint-Wiederverwendung für jeden Lauf zu validieren.

[[9-3-troubleshooting-quick-reference]] == 9.3 Kurzanleitung zur Fehlerbehebung

Anhand dieser Tabelle lässt sich häufigen Aufgabenfehlern die erste Korrekturmaßnahme zuordnen. Probleme mit der Quellverbindung und der Einbindung sollten vor einem erneuten Lauf behoben werden; ein erneuter Versuch ohne Korrektur einer ungültigen Konfiguration oder eines nicht verfügbaren Endpunkts führt zum gleichen Fehler.

Symptom	Wahrscheinliche Ursache	Auflösung
Could not resolve XCP S3 credentials	Fehlend/falsch xcp_s3_profile in xcp_s3_profiles	Profilname und Anmeldeinformationen zuordnen überprüfen
no readable Lustre source directory	LustreFS ist nicht eingebunden oder fehlerhaft xcp_lustrefs_dest_path	Überprüfen Sie den Dateisystemtyp mit findmnt -T /mnt/lustre/client -o FSTYPE,TARGET -n; korrekter Pfad
Missing required text dataset files	Text-JSONs fehlen im Rohdaten-S3-Präfix	Hochladen text_base.json, text_finetune.json, text_infer.json
Preflight Exit-Code 1	SSH nicht erreichbar	Verbindung und Host/Benutzer ssh_default überprüfen
Preflight Exit-Code 2	NFS Pfad nicht exportiert/sichtbar	Export prüfen mit xcp show <ip>

[[9-4-checkpoint-operations]] == 9.4 Checkpoint Vorgänge

Die Checkpoint-Funktion gilt nur für model_training. training_checkpoint_reuse_mode=resume_if_exists dient dazu, ein gültiges abgeschlossenes Trainingsergebnis wiederzuverwenden und redundantes Modelltraining zu überspringen, oder verify_only zur Überprüfung der Checkpoint-Eignung, ohne das Überspringen auszuführen. Halten Sie die Wiederverwendung während der ersten Tests oder immer dann deaktiviert (off, wenn sich Trainingseingaben, Konfiguration oder erwartete Artefakte geändert haben.

[[9-5-manual-archive-operations]] == 9.5 Manuelle Archivierungsvorgänge

Legen Sie manual_archive_enabled=true fest und wählen Sie manual_archive_stage bewusst aus. Verwenden Sie model_training, wenn die Governance die Baseline des Kernmodells unmittelbar nach erfolgreichem Training erfordert; verwenden Sie inferencing, wenn das Archiv die finalen Vorhersagen enthalten muss. Ein Lauf mit ausgewähltem model_training wird nach dem Archivierungszweig mit Feinabstimmung und Inferenz fortgesetzt; die späteren Ausgaben werden dabei lediglich nicht archiviert. Uploads unter dem mit dem Laufstempel versehenen StorageGRID Präfix sowie [manual_archive] Protokolle für die ausgewählte Phase, die gefundenen Dateien und die Anzahl der Uploads sollten überprüft werden.

[[9-6-scaling-guidance]]
== 9.6 Skalierungsleitfaden

Die Parameter `sample_count`, `spark_driver_memory`, `spark_executor_memory` und `spark_timeout_secs` werden an das Eingabevolumen und das Servicelevel-Ziel angepasst. Es wird mit begrenzten Stichproben begonnen, um Datenstruktur und Routing zu validieren; anschließend wird `sample_count=0` für Läufe mit allen Zeilen verwendet, nachdem StorageGRID, ONTAP NAS, XCP und die ausgewählte Trainingsstufe über ausreichende Kapazität und ausreichenden Durchsatz verfügen.

[[9-7-backup-and-recovery]]
== 9.7 Datensicherung und Wiederherstellung

Der ONTAP NAS-Datenspeicher für vorbereitete Daten wird mit ONTAP Snapshot und Backup geschützt. StorageGRID Schutz- und Aufbewahrungsrichtlinien werden auf Rohdaten- und Archivdatenspeicher angewendet; der Archivdatenspeicher behält den Verlauf unabhängig von einer erneuten Pipeline-Ausführung bei. Diese Speicherschutzmaßnahmen werden mit dem Trainings-Checkpoint-Datensatz kombiniert, um nach einer Unterbrechung auf Aufgaben-, Host- oder Standortebene die korrekten, laufbezogenen Daten und Baseline-Artefakte wiederherzustellen.

10. Sicherheitsaspekte, Validierung und Tests

Karthikeyan Nagalingam, NetApp

Dieser kombinierte Abschnitt behandelt die erforderlichen Kontrollmechanismen für den sicheren Betrieb der KI-Pipeline sowie die Validierungsaktivitäten, mit denen nachgewiesen wird, dass sich das Design in repräsentativen Umgebungen wie erwartet verhält. Dieselbe Trennung der Storage-Tiers, das Zugriffsmodell mit minimalen Berechtigungen, die Prüfpunktlogik und die laufbezogene Datenherkunft, die in der Pipeline verwendet werden, fließen auch in den Validierungsplan und die betrieblichen Leitplanken ein.

[[10-1-security-considerations]]
== 10.1 Sicherheitsüberlegungen

Sicherheitsmaßnahmen sollten jede Speicherebene und Pipeline-Funktion isolieren und gleichzeitig die für Audits erforderliche Datenherkunft bewahren. Für StorageGRID Rohdaten, ONTAP NAS aufbereitete Daten, das ausgewählte XCP-Ziel und die StorageGRID Archivierung sollten separate Anmeldeinformationen mit minimalen Berechtigungen verwendet werden. Der Netzwerkverkehr sollte verschlüsselt und alle Geheimnisse außerhalb der DAG-Definition, der Beispiele und des Shell-Verlaufs gespeichert werden.

Bereich	Empfehlung
Anmeldedatenverwaltung	Airflow Connections/Variables oder ein Secrets Manager sollten verwendet werden; Inline-Klartext in <code>dag_run.conf</code> ist zu vermeiden.
Daten während der Übertragung	TLS-fähige S3-Endpunkte und verschlüsselte NFS/Lustre Übertragung, sofern unterstützt
Zugangskontrolle	S3-Anmeldeinformationen pro Rolle (StorageGRID Rohdaten, ONTAP NAS aufbereitete Daten, XCP Ziel, StorageGRID Archivierung) auf die minimal erforderlichen Berechtigungen beschränken
Geheimnisse in Konfigurationen	Rotieren Sie alle während der Tests offengelegten Zugangsdaten; behandeln Sie Trigger-Skripte/Shell-Verlauf als vertraulich

Bereich	Empfehlung
Prüfprotokoll	Datenaufbereitungsmanifeste, Prüfpunktprotokolle und Archivmanifeste dienen als Nachweis für die Einhaltung von Vorschriften.

[[10-2-validation-and-testing]]

== 10.2 Validierung und Testen

Die Validierung bestätigt, dass das konfigurierbare Verhalten der Pipeline in Bezug auf Routing, Fehlerbehandlung, Datenmobilität und die Erkennung mehrerer Dateien zuverlässig funktioniert. Jeder Validierungstestbereich wurde in einer repräsentativen Umgebung mit realen Airflow DAG Triggern ausgeführt, wobei die Protokollausgaben und Speichermanifeste auf Korrektheit überprüft wurden.

[[10-2-1-functional-routing-validation]]

=== 10.2.1 Validierung des funktionalen Routings

Dieser Testbereich validiert das durchgängige Daten- und Artefaktrouting, wenn `enable_xcp=true`.

- **Ziel:** Sicherstellen, dass Datenaufbereitung, XCP-Mobilität, Modelltraining, Feinabstimmung und Inferenz konsistent das ausgewählte XCP-Ziel verwenden (`s3` oder `lustrefs`).
- **Testverfahren:** DAG-Läufe wurden mit `xcp_copy_destination="s3"` und `xcp_copy_destination="lustrefs"` ausgelöst. XCom-Ausgaben, `effective-config`-Protokolle und Speicherorte wurden überprüft.
- **Validierungsergebnis:**
 - In `data_prep` wurden formatierte tabellarische Aufteilungen und Textdateien auf der ONTAP NAS Prepared-Data-Ebene unter `<xcp_prefix>/formatted/<run_stamp>/data/` bereitgestellt.
 - In `xcp_copy` diesem Fall wurde der vorbereitete Datensatz an das gewählte XCP Ziel für die nachfolgenden Modellschritte übertragen.
 - In `model_training` wurden Modelleingaben aus dem XCP Ziel geladen, und Basisartefakte (`regression_model.bin`, `text_vectorizer.bin`, `text_classifier.bin`, `train_metrics.json`) wurden zurück in `<xcp_prefix>/formatted/<run_stamp>/artifacts/` veröffentlicht.
 - In `fine_tuning` wurden Basisvektorisierungs-/Klassifizierungsartefakte aus dem XCP Ziel materialisiert, inkrementell optimiert und als `text_classifier_tuned.bin` und `fine_tune_metrics.json` zurück veröffentlicht.
 - In `inferencing` wurden der optimierte Textklassifikator sowie das Basisregressionsmodell und der Vektorisierer aus dem XCP-Ziel materialisiert, um Vorhersagen zu generieren.

[[10-2-2-local-fallback-and-non-xcp-execution-validation]]

=== 10.2.2 Validierung der lokalen Fallback- und Nicht-XCP-Ausführung

Dieser Testbereich validiert das Pipeline-Verhalten, wenn die XCP Datenmobilität deaktiviert ist (`enable_xcp=false`).

- **Ziel:** Sicherstellen, dass die Pipeline reibungslos mit direkten lokalen Pfaden für vorbereitete Daten funktioniert, ohne dass eine SSH-Verbindung, Remote XCP Hosts oder XCP Anmeldeinformationsprofile erforderlich sind.
- **Testverfahren:** Ausgelöste Pipeline-Ausführungen mit `enable_xcp=false` und Standardpfaden für

S3/lokal.

- **Validierungsergebnis:**

- XCP Preflight und Kopieraufgaben wurden sicher umgangen.
- `model_training`, `fine_tuning` und `inferencing` lesen direkt aus lokalen Verzeichnissen mit aufbereiteten Daten und lokal veröffentlichten Artefakten.
- Es wurde überprüft, dass keine Fehler bei der Auflösung von XCP S3-Anmeldeinformationen oder SSH-Preflight-Fehler auftraten, als XCP deaktiviert war.

[[10-2-3-performance-and-tier-comparison-ontap-s3-vs-lustrefs]]

=== 10.2.3 Leistungs- und Tiervergleich (ONTAP S3 vs. LustreFS)

Dieser Testbereich evaluiert die E/A-Latenz, den Discovery-Overhead und den Trainingsdurchsatz zwischen ONTAP S3 Objektspeicher und LustreFS Parallelspeichersystem Ebenen.

- **Zielsetzung:** Quantifizierung der Leistungsmerkmale der E-Serie mit LustreFS im Vergleich zu ONTAP AFF/AFX mit aktiven ONTAP S3 Trainingsebenen.
- **Testverfahren:** Gemessen wurden die Dateierkennungszeit, die Latenz beim Laden des Trainingsdatensatzes und die Dauer der Veröffentlichung/Materialisierung von Artefakten bei identischen Datensatzgrößen (14,400+ Zeilen, Text-/Tabelleneingaben aus mehreren Dateien).
- **Validierungsergebnis:**
 - **LustreFS Tier:** Nachweislich geringerer Aufwand bei der Dateierkennung und schnellere Latenz bei Lesezugriffen mit wahlfreiem Zugriff während des Modelltrainings, wodurch es sich ideal für GPU-intensive parallele Trainingsworkloads mit hoher Dateianzahl eignet.
 - **ONTAP S3 Tier:** Bietet hohen Durchsatz bei einfacherer Verwaltung mehrerer Protokolle und ist somit optimal für kostenbewussten aktiven Speicher mit elastischem Zugriff geeignet, ohne dass Client-seitig Dateisystem-Mounts erforderlich sind.

[[10-2-4-failure-recovery-and-training-checkpoint-validation]]

=== 10.2.4 Validierung von Wiederherstellung nach Ausfällen und Schulungs-Checkpoints

Dieser Testbereich validiert das Checkpoint-Resume-System mit dem Geltungsbereich `model_training`.

- **Ziel:** Es ist sicherzustellen, dass die Pipeline-Wiederherstellung zuvor erfolgreiche Trainingsläufe korrekt erkennt und redundante Berechnungen überspringt, während die Integrität der Artefakte erhalten bleibt.
- **Testverfahren:**
 - a. Pipeline mit `checkpoint_enabled=true` und `checkpoint_store="formatted_s3"` ausgeführt.
 - b. Simulierter Aufgabenfehler oder erneute Ausführung mit `training_checkpoint_reuse_mode="resume_if_exists"`.
 - c. Getestete `verify_only` und `off` Wiederverwendungsmodi.
- **Validierungsergebnis:**
 - Wenn `resume_if_exists` ein gültiger Checkpoint JSON + alle Kernartefakte in S3/LustreFS vorhanden waren, wurde `model_training` sofort mit der Entscheidung `RESUMED_FROM_CHECKPOINT` beendet, unter Protokollierung `[checkpoint] resume_summary: decision=RESUMED_FROM_CHECKPOINT`.
 - Nachgelagert `fine_tuning` und `inferencing` wurden die verifizierten Checkpoint-Artefakte erfolgreich materialisiert.

- Wenn `verify_only` dies festgelegt war, validierte der Guard die Existenz des Artefakts, ohne das Training zu überspringen.

[[10-2-5-scalability-and-multi-file-dataset-validation]]

=== 10.2.5 Skalierbarkeit und Validierung von Datensätzen mit mehreren Dateien

Dieser Testbereich validiert die Skalierbarkeit über große tabellarische und Textdatensätze mit mehreren Dateien hinweg.

- **Ziel:** Bestätigung der automatischen Datensatzerkennung, der Ausführung der Spark-Transformations-Engine und der Unterstützung des Lakehouse-Tabellenformats (`delta` und `iceberg`).
- **Testverfahren:**
 - a. Die Aufnahme und Aufbereitung von Dutzenden von CSV/Parquet-Dateien unter dem StorageGRID Rohpräfix (`s3_raw_prefix` wurde durchgeführt).
 - b. Getestet wurde die explizite Dateiauswahl mit `s3_tabular_object_keys` im Vergleich zur automatischen Erkennung aller Dateien (`sample_count=0`).
 - c. Vorbereitung mit PySpark mit `table_format="delta"` und `table_format="iceberg"` ausgeführt.
- **Validierungsergebnis:**
 - Spark distributed preparation hat große, mehrteilige tabellarische Datensätze erfolgreich eingelesen, gefiltert, aufgeteilt und formatiert.
 - Die automatische Erkennung identifizierte präzise alle gültigen tabellarischen CSV/Parquet-Objekte und ignorierte dabei nicht tabellarische Artefakte.
 - Sowohl die Tabellenformate Delta Lake als auch Iceberg wurden korrekt erstellt und unter `tables/` registriert, wobei das nachgelagerte Modelltraining die formatierten Aufteilungen erkannte und las.

[[10-2-6-example-customer-evaluation-workflow-sizing-assumptions-and-validation-metrics]]

=== 10.2.6 Beispielhafter Workflow zur Kundenbewertung, Dimensionierungsannahmen und Validierungsmetriken

Dieser Workflow unterstützt Kunden bei der Bewertung, ob das Design ihren Anforderungen an die Reife ihrer KI-Pipeline, die Datenhoheit und die Leistungsziele entspricht. Es wird mit einem repräsentativen Datensatz und einem Pipeline-Lauf begonnen, anschließend werden Datenvolumen, Dateianzahl, Modellkomplexität und Anzahl paralleler Läufe erst erhöht, nachdem jedes Akzeptanzkriterium erfüllt ist. Die Ergebnisse werden `run_stamp` protokolliert, damit Vergleiche zwischen ONTAP S3 und LustreFS dieselben Eingabedaten und dieselbe Konfiguration verwenden.

Bewertungsschritt	Beispielhafter Arbeitsablauf	Dimensionierungsannahme / Entscheidung	Validierungsmetriken und Akzeptanznachweise
1. Eine Ausgangsbasis festlegen	Der Python-Vorbereitungspfad wird mit <code>enable_xcp=false</code> einer begrenzten Stichprobe ausgeführt.	Die Referenzvalidierungsumgebung aus Abschnitt 7.1.1 dient als anfängliche funktionale Basislinie.	Erfolgreicher DAG-Abschluss; Anzahl der Eingabezeilen und Ausgabe-Splits; Modellmetriken; Aufgabendauer; CPU-, Speicher- und lokale Festplattenauslastung.

Bewertungsschritt	Beispielhafter Arbeitsablauf	Dimensionierungsannahme / Entscheidung	Validierungsmetriken und Akzeptanznachweise
2. Datenhoheit validieren	Rohdaten, aufbereitete Daten, aktive Trainingsdaten und Archive werden in genehmigten Speicherendpunkten und Regionen aufbewahrt. Anmeldeinformationen und Aufbewahrungsrichtlinien gelten für jede Ebene.	Die zulässigen Speicherorte, Zugriffsrollen, Verschlüsselungsanforderungen und die Aufbewahrungsrichtlinie sollten vor dem Verschieben von Daten festgelegt werden.	Endpunkt, Bucket, Präfix und Region in der effektiven Konfiguration und in den Manifesten erfasst; Überprüfung des Zugriffs nach dem Least-Privilege-Prinzip; Nachweis der Archivierungs- und Löschrichtlinien.
3. Validierung der Datenmobilität	XCP wird aktiviert und derselbe vorbereitete Lauf nach ONTAP S3 kopiert, anschließend wird der Vorgang für LustreFS wiederholt.	10-GbE-Konnektivität und ausreichende Zielkapazität für den vorbereiteten Lauf, die Artefakte, den Arbeitsbereich und gleichzeitige Läufe müssen sichergestellt sein.	XCP Abschlussstatus; Anzahl kopierter Bytes und Dateien; Kopierdauer und Durchsatz; Anzahl der Quell-/Zielformate und Vergleich von Prüfsumme oder Manifest; erfolgreicher Vorabcheck.
4. Überprüfung der Eignung des Trainingsniveaus	Training, Feinabstimmung und Inferenz erfolgen für jedes ausgewählte Ziel mit denselben Datensätzen und Modelleinstellungen.	ONTAP S3 eignet sich für objektbasierte Workflows; E-Series mit LustreFS eignet sich, wenn paralleles Trainings-I/O oder eine hohe Anzahl von Dateien dies rechtfertigen.	Datensatzerkennung und Ladezeit; Dauer von Training, Feinabstimmung und Inferenz; Dauer der Veröffentlichung/Materialisierung von Artefakten; CPU-/GPU-Auslastung, sofern zutreffend; Konsistenz der Modellqualität.
5. Skalierung und Wiederherstellung validieren	Erhöhung <code>sample_count</code> für alle Zeilen sowie der Dateianzahl und der gleichzeitigen Ausführungen, anschließend Test der Wiederverwendung von Prüfpunkten und der Archivierung.	Kapazität für die Speicherung laufbezogener Datensätze und Artefakte sowie Wachstumsspielraum dimensionieren; CPU- und Speicherkapazität für Spark und gleichzeitige Airflow-Aufgaben dimensionieren.	End-to-End-Verstrichene Zeit; Erfolgsquote bei erfolgreichen Ausführungen; Warteschlangenzeit; Entscheidung zur Wiederaufnahme des Checkpoints und verstrichene Zeit; Vollständigkeit des Archivs; Speicherwachstum pro Durchlauf; Erreichen des Recovery Time Objective (RTO).

Vor Beginn der Evaluierung sollten Kunden quantitative Zielvorgaben für die End-to-End-Laufzeit, den XCP-Durchsatz, die Ladezeit der Trainingsdaten, die maximale Anzahl gleichzeitiger Ausführungen, die Wiederherstellungszeit und die Speicheraufbewahrung definieren. Die gemessene Basislinie und jeder

skalierte Test sollten mit diesen Zielvorgaben verglichen werden, um festzustellen, ob die Architektur die beabsichtigte Workload und die betrieblichen Anforderungen erfüllt.

11. NetApp Differenzierung und Wettbewerbspositionierung, Anhänge

Karthikeyan Nagalingam, NetApp

Dieser kombinierte Abschnitt stellt die Architektur gängigen Alternativen gegenüber und bietet Betreibern und Prüfern das gemeinsame Referenzmaterial. Der entscheidende Unterschied liegt nicht in einem einzelnen Speicherendpunkt, sondern in der richtliniengesteuerten Kombination aus StorageGRID, ONTAP NAS, ONTAP S3, LustreFS, XCP und Airflow, durch die jede Phase des KI-Lebenszyklus die jeweils passende Speicherebene nutzen kann, ohne den Workflow neu aufbauen zu müssen.

[[11-1-netapp-differentiation-and-competitive-positioning]]
== 11.1 NetApp Differenzierung und Wettbewerbspositionierung

Dieser Vergleich stellt die Architektur gängigen Alternativen gegenüber. Ihr Alleinstellungsmerkmal ist nicht ein einzelner Speicherendpunkt, sondern die richtliniengesteuerte Kombination aus StorageGRID, ONTAP NAS, ONTAP S3, LustreFS, XCP und Airflow, die es jeder Phase des KI-Lebenszyklus ermöglicht, eine geeignete Ebene zu nutzen, ohne den Workflow neu aufbauen zu müssen.

Alternativer Ansatz	Einschränkung	NetApp Solution Advantage
Pipelines, die ausschließlich nativ in der Cloud ausgeführt werden	An ein einziges Speicherparadigma gebunden	Austausch von S3/LustreFS über die Konfiguration
Selbst erstellte skriptbasierte Pipelines	Kein Wiederholungsversuch, kein Prüfpunkt, keine Herkunftsverfolgung	Airflow Orchestrierung + Checkpoint/Fortsetzen nativ
Einstufige Speicherarchitekturen	Ein Leistungs-/Kostenprofil für alle Workloads	Auswahl der Tiering-Ebene pro Durchlauf über <code>xcp_copy_destination</code>
Vollständiges Neutraining bei jedem Update	Hohe wiederkehrende Rechenkosten	Inkrementelle Feinabstimmung über <code>partial_fit</code>

[[11-2-appendices]]
== 11.2 Anhänge

Die Anhänge bieten gemeinsame Terminologie und Kontext zur Dokumentenkontrolle für Leser, die diese NVA betreiben, überprüfen oder anpassen. Die Revisionshistorie ist aktuell zu halten, wenn dokumentiertes Verhalten, Parameter, Speicherlayouts oder Validierungsnachweise geändert werden.

[[11-2-1-glossary]]
=== 11.2.1 Glossar

Diese Definitionen legen einheitliche Bedeutungen für die Speicher-, Orchestrierungs- und Herkunftsbegriffe der Architektur fest.

Begriff	Definition
XCP	NetApps Hochgeschwindigkeits-Tool für parallele Datenkopie/-migration
ONTAP S3	NetApps S3-kompatibler Objektspeicherdienst auf ONTAP
LustreFS	Hochleistungsfähiges paralleles verteiltes Dateisystem
DAG	Gerichteter azyklischer Graph, Workflow-Definition von Airflow
XCom	Airflows Mechanismus für die Übergabe von Daten zwischen Aufgaben
run_stamp	Eindeutiger Zeitstempel-Identifikator pro Ausführung zur Bestimmung der Herkunft
Manifest	JSON-Datensatz, der die Eingaben, Ausgaben und die Konfiguration einer Stufe beschreibt

[[11-2-2-revision-history]]

=== 11.2.2 Revisionshistorie

Jede veröffentlichte Änderung sollte protokolliert werden, um den Prüf- und Genehmigungsverlauf des Dokuments zu erhalten.

Version	Datum	Autor	Ändern
1,0	25. September 2026	Team für Lösungsarchitektur	Erste publikationsreife NVA mit vollständigen Parametertabellen und Diagrammen

[[11-2-3-contact-us-and-software-requests]]

=== 11.2.3 Kontakt und Softwareanfragen

Zur Anforderung des Softwarepakets, der Automatisierungs-DAGs, der Bereitstellungsskripte oder der Validierungsartefakte für die Bereitstellung und Validierung dieser Architektur sowie für Feedback und technische Anfragen können Sie das NetApp AI & Data Mobility Team kontaktieren:

- **E-Mail:** ng-data-mobility-in-ai-pipeline@netapp.com
- **Zweck:** Softwarepaketanfragen, Bereitstellungshinweise, Unterstützung bei der Architekturvalidierung und technisches Feedback.

Copyright-Informationen

Copyright © 2026 NetApp. Alle Rechte vorbehalten. Gedruckt in den USA. Dieses urheberrechtlich geschützte Dokument darf ohne die vorherige schriftliche Genehmigung des Urheberrechtsinhabers in keiner Form und durch keine Mittel – weder grafische noch elektronische oder mechanische, einschließlich Fotokopieren, Aufnehmen oder Speichern in einem elektronischen Abrufsystem – auch nicht in Teilen, vervielfältigt werden.

Software, die von urheberrechtlich geschütztem NetApp Material abgeleitet wird, unterliegt der folgenden Lizenz und dem folgenden Haftungsausschluss:

DIE VORLIEGENDE SOFTWARE WIRD IN DER VORLIEGENDEN FORM VON NETAPP ZUR VERFÜGUNG GESTELLT, D. H. OHNE JEGLICHE EXPLIZITE ODER IMPLIZITE GEWÄHRLEISTUNG, EINSCHLIESSLICH, JEDOCH NICHT BESCHRÄNKT AUF DIE STILLSCHWEIGENDE GEWÄHRLEISTUNG DER MARKTGÄNGIGKEIT UND EIGNUNG FÜR EINEN BESTIMMTEN ZWECK, DIE HIERMIT AUSGESCHLOSSEN WERDEN. NETAPP ÜBERNIMMT KEINERLEI HAFTUNG FÜR DIREKTE, INDIREKTE, ZUFÄLLIGE, BESONDERE, BEISPIELHAFTE SCHÄDEN ODER FOLGESCHÄDEN (EINSCHLIESSLICH, JEDOCH NICHT BESCHRÄNKT AUF DIE BESCHAFFUNG VON ERSATZWAREN ODER -DIENSTLEISTUNGEN, NUTZUNGS-, DATEN- ODER GEWINNVERLUSTE ODER UNTERBRECHUNG DES GESCHÄFTSBETRIEBS), UNABHÄNGIG DAVON, WIE SIE VERURSACHT WURDEN UND AUF WELCHER HAFTUNGSTHEORIE SIE BERUHEN, OB AUS VERTRAGLICH FESTGELEGTER HAFTUNG, VERSCHULDENSUNABHÄNGIGER HAFTUNG ODER DELIKTSHAFTUNG (EINSCHLIESSLICH FAHRLÄSSIGKEIT ODER AUF ANDEREM WEGE), DIE IN IRGEND EINER WEISE AUS DER NUTZUNG DIESER SOFTWARE RESULTIEREN, SELBST WENN AUF DIE MÖGLICHKEIT DERARTIGER SCHÄDEN HINGEWIESEN WURDE.

NetApp behält sich das Recht vor, die hierin beschriebenen Produkte jederzeit und ohne Vorankündigung zu ändern. NetApp übernimmt keine Verantwortung oder Haftung, die sich aus der Verwendung der hier beschriebenen Produkte ergibt, es sei denn, NetApp hat dem ausdrücklich in schriftlicher Form zugestimmt. Die Verwendung oder der Erwerb dieses Produkts stellt keine Lizenzierung im Rahmen eines Patentrechts, Markenrechts oder eines anderen Rechts an geistigem Eigentum von NetApp dar.

Das in diesem Dokument beschriebene Produkt kann durch ein oder mehrere US-amerikanische Patente, ausländische Patente oder anhängige Patentanmeldungen geschützt sein.

ERLÄUTERUNG ZU „RESTRICTED RIGHTS“: Nutzung, Vervielfältigung oder Offenlegung durch die US-Regierung unterliegt den Einschränkungen gemäß Unterabschnitt (b)(3) der Klausel „Rights in Technical Data – Noncommercial Items“ in DFARS 252.227-7013 (Februar 2014) und FAR 52.227-19 (Dezember 2007).

Die hierin enthaltenen Daten beziehen sich auf ein kommerzielles Produkt und/oder einen kommerziellen Service (wie in FAR 2.101 definiert) und sind Eigentum von NetApp, Inc. Alle technischen Daten und die Computersoftware von NetApp, die unter diesem Vertrag bereitgestellt werden, sind gewerblicher Natur und wurden ausschließlich unter Verwendung privater Mittel entwickelt. Die US-Regierung besitzt eine nicht ausschließliche, nicht übertragbare, nicht unterlizenzierbare, weltweite, limitierte unwiderrufliche Lizenz zur Nutzung der Daten nur in Verbindung mit und zur Unterstützung des Vertrags der US-Regierung, unter dem die Daten bereitgestellt wurden. Sofern in den vorliegenden Bedingungen nicht anders angegeben, dürfen die Daten ohne vorherige schriftliche Genehmigung von NetApp, Inc. nicht verwendet, offengelegt, vervielfältigt, geändert, aufgeführt oder angezeigt werden. Die Lizenzrechte der US-Regierung für das US-Verteidigungsministerium sind auf die in DFARS-Klausel 252.227-7015(b) (Februar 2014) genannten Rechte beschränkt.

Markeninformationen

NETAPP, das NETAPP Logo und die unter <http://www.netapp.com/TM> aufgeführten Marken sind Marken von NetApp, Inc. Andere Firmen und Produktnamen können Marken der jeweiligen Eigentümer sein.