Redbooks
ibm.com/redbooks

Making Data Smarter with IBM Spectrum Discover: Practical AI Solutions

Ivaylo B. Bozhinov
Isom Crawford Jr., PhD
Joseph Dain
Mathias Defiebre
Maxime Deloche
Kiran Ghag
Vasfi Gucer
Xin Liu
Abeer Selim
Gauthier Siri
Christopher Vollmar

Analytics

Storage

In partnership with
IBM **Academy of Technology**

IBM

Redbooks

IBM Redbooks

Making Data Smarter with IBM Spectrum Discover

October 2020

SG24-8488-00

Note: Before using this information and the product it supports, read the information in “Notices” on page vii.

First Edition (October 2020)

This edition applies to IBM Spectrum Discover Version 2.0.3.1.

Contents

Notices

This information was developed for products and services offered in the US. This material might be available from IBM in other languages. However, you may be required to own a copy of the product or product version in that language in order to access it.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:
IBM Director of Licensing, IBM Corporation, North Castle Drive, MD-NC119, Armonk, NY 10504-1785, US

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you provide in any way it believes appropriate without incurring any obligation to you.

The performance data and client examples cited are presented for illustrative purposes only. Actual performance results may vary depending on specific configurations and operating conditions.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to actual people or business enterprises is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at “Copyright and trademark information” at http://www.ibm.com/legal/copytrade.shtml

The following terms are trademarks or registered trademarks of International Business Machines Corporation, and might also be trademarks or registered trademarks in other countries.

AIX®
Enterprise Design Thinking™
Global Business Services®
IBM®
IBM Cloud®
IBM Cloud Pak®
IBM Elastic Storage®
IBM Garage™
IBM Security™
IBM Spectrum®
IBM Spectrum Storage™
IBM Watson®
IBM Z®
Insight®
Maximo®
Redbooks®
Redbooks (logo) ®
System Storage™
TotalStorage™
Watson™

The following terms are trademarks of other companies:

ITIL is a Registered Trade Mark of AXELOS Limited.

The registered trademark Linux® is used pursuant to a sublicense from the Linux Foundation, the exclusive licensee of Linus Torvalds, owner of the mark on a worldwide basis.

Linear Tape-Open, LTO, Ultrium, the LTO Logo and the Ultrium logo are trademarks of HP, IBM Corp. and Quantum in the U.S. and other countries.

Microsoft, Windows, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Ceph, OpenShift, Red Hat, are trademarks or registered trademarks of Red Hat, Inc. or its subsidiaries in the United States and other countries.

VMware, and the VMware logo are registered trademarks or trademarks of VMware, Inc. or its subsidiaries in the United States and/or other jurisdictions.

Other company, product, or service names may be trademarks or service marks of others.

Preface

More than 80% of all data that is collected by organizations is not in a standard relational database. Instead, it is trapped in unstructured documents, social media posts, machine logs, and so on. Many organizations face significant challenges to manage this deluge of unstructured data, such as the following examples:

- Pinpointing and activating relevant data for large-scale analytics
- Lacking the fine-grained visibility that is needed to map data to business priorities
- Removing redundant, obsolete, and trivial (ROT) data
- Identifying and classifying sensitive data

IBM® Spectrum Discover is a modern metadata management software that provides data insight for petabyte-scale file and Object Storage, storage on-premises, and in the cloud. This software enables organizations to make better business decisions and gain and maintain a competitive advantage.

IBM Spectrum® Discover provides a rich metadata layer that enables storage administrators, data stewards, and data scientists to efficiently manage, classify, and gain insights from massive amounts of unstructured data. It improves storage economics, helps mitigate risk, and accelerates large-scale analytics to create competitive advantage and speed critical research.

This IBM Redbooks® publication presents several use cases that are focused on artificial intelligence (AI) solutions with IBM Spectrum Discover. This book helps storage administrators and technical specialists plan and implement AI solutions by using IBM Spectrum Discover and several other IBM Storage products.

Authors

This book was produced by a team of specialists from around the world.

Ivaylo B. Bozhinov has been working at IBM Bulgaria for 4 years as a technical support professional. His main areas of expertise are IBM Power Systems products, IBM AIX®, IBM i, and Red Hat Enterprise Linux. He has a bachelor's degree in Information Technology from the State University of Librarian and Information Technology of Sofia, Bulgaria. He holds several IBM certifications, which include Hadoop Administration, Hadoop Foundation, Data Science Foundation, IBM Private Cloud, and IBM Blockchain Foundation. His areas of interest include AI, DL, ML, blockchain, and cloud.

Isom Crawford Jr., PhD is a Subject Matter Expert for Software Defined Infrastructure at IBM Washington Systems Center. He has over 20 years of experience in computer software product architecture and development. He holds a PhD in Mathematical Sciences from the University of Texas at Dallas and MS in Applied Mathematics from Oklahoma State University. He has developed and delivered multiple technical training courses, holds nine patents, and authored multiple publications, including *Software Optimization for High Performance Computers* (ISBN 0130170089).

Joe Dain is a Senior Technical Staff Member and Master Inventor in the IBM Systems Storage organization in Tucson, Arizona. He is currently on his 26th invention plateau and has over 100 patents issued and pending worldwide. Joseph joined IBM in 2003 with a BS in Electrical Engineering and is the Chief Architect for IBM Spectrum Discover.

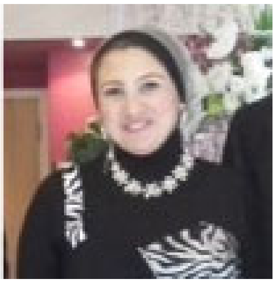
Mathias Defiebre is a leading IBM expert for Analytics, Object Storage, and Data Protection with over 20 years of storage experience. From IBM’s EMEA Storage Competence Centre (ESCC), he provides support to customers through the Advanced Technical Skills (pre-sales support) and Lab Services channels (Implementations, Migrations, Health checks, Proof of Concepts, and Workshops). He graduated from the University of Cooperative Education Mannheim with a German Diploma in Information Technology Management and a Bachelor of Science. Mathias also is a Master Certified IT Specialist and an IBM Certified Specialist for TotalStorage™ Networking and Virtualization Architectures. He is the author of several Storage Redbooks publications.

Maxime Deloche is a Deep Learning Engineer working at the Cognitive Systems Lab of the IBM Systems Center Montpellier, France. As a technical pre-sales specialist, he helps clients developing AI workloads around computer vision and natural language processing. He is also developing his expertise in cloud technologies and how to use them for AI applications. He joined IBM in 2018 after receiving an Engineering Degree from the ENSIMAG, Grenoble, and was part of the IBM team that won the French Radiology Society data challenge in 2019 with their lung nodules detection AI pipeline.

Kiran Ghag is an IBM Storage Solution Architect, working with IBM Systems Lab Services at IBM India. He received his bachelors degree in Computer Engineering from Mumbai University. His current interests include software-defined storage (SDS) and high-performance computing (HPC) using IBM Spectrum family. Kiran has over 18 years of experience in storage systems design, deployment, operations, and optimization.

Vasfi Gucer is an IBM Redbooks project leader with the IBM Garage™ for Systems organization. He writes extensively and teaches IBM classes worldwide about IBM products. His focus has been primarily on cloud computing, including hybrid cloud storage technologies for the last 6 years. Vasfi is also an IBM Certified Senior IT Specialist, Project Management Professional (PMP), IT Infrastructure Library (ITIL) V3 Expert.

Xin Liu is a Senior Technical Manager of Advanced Technical Skills team and a Level-II Certified IT Specialist in IBM Great China Group. He has 14 years of working experience in various IBM business units, covering solution design and delivery. He is an expert in building and optimizing data infrastructure for data intensive workloads. In recent years, he focused on helping customer better manage their unstructured data and improve the capabilities of data governance. Xin holds a master's degree in Electronic Engineering from Tsinghua University, Beijing.



Abeer Selim is a Certified IT Specialist Level 2 in IBM Global Business Services® and the Integration Practice Lead at the Client Innovation Center (CIC), IBM Egypt. She has over 11 years of experience in the IT industry. Abeer co-authored several scientific papers, such as *Machine Learning Methodologies in Brain-Computer Interface Systems*, *Machine learning methodologies in P300 speller Brain-Computer Interface systems*, and *Electrode Reduction Using ICA and PCA in P300 Visual Speller Brain-Computer Interface System.* Abeer holds a B.S. and M.S. in Biomedical and Systems Engineering from Cairo University in Egypt.

Gauthier Siri is a Storage IT Specialist from IBM France, based at the IBM Systems Center Montpellier. He started his IT career as a storage administrator and he is now part of a worldwide pre-sales team, whose goal is to help IBM and Business Partner Sales Teams to demonstrate IBM Storage added-value. With a strong block storage background, his natural curiosity led him to new technologies, from automation to containerization, and to support solutions like IBM Spectrum Discover or IBM Storage solution for OpenShift Containerization platform.

Christopher Vollmar is an IBM Certified Consulting IT Specialist (Level 3 Thought Leader) and Storage Architect based in Toronto, Ontario, Canada with the IBM Systems Group. Christopher is currently focused on helping customers build storage solutions using the IBM Spectrum Storage/Software Defined Storage family. He is an Enterprise Design Thinking™ Co-Creator and an IBM Redbooks Platinum Author focused on helping customers develop private and hybrid storage cloud solutions using the IBM Spectrum Storage™ family and Converged Infrastructure solutions. Christopher has worked for IBM for almost 20 years across several different areas of the IBM business, and has spent the past 10 years working with the IBM System Storage™. Christopher holds an honours degree in Political Science from York University.

Thanks to the following people for their contributions to this project:

David Wohlford
Larry Coyne
Pallavi V Galgali
Erica Wazewski
Bob Chesebrough
Khanh Ngo
IBM USA

Now you can become a published author, too!

Here’s an opportunity to spotlight your skills, grow your career, and become a published author—all at the same time! Join an IBM Redbooks residency project and help write a book in your area of expertise, while honing your experience using leading-edge technologies. Your efforts will help to increase product acceptance and customer satisfaction, as you expand your network of technical contacts and relationships. Residencies run from two to six weeks in length, and you can participate either in person or as a remote resident working from your home base.

Find out more about the residency program, browse the residency index, and apply online at:

ibm.com/redbooks/residencies.html

Comments welcome

Your comments are important to us!

We want our books to be as helpful as possible. Send us your comments about this book or other IBM Redbooks publications in one of the following ways:

- Use the online **Contact us** review Redbooks form found at:

  ibm.com/redbooks

- Send your comments in an email to:

  redbooks@us.ibm.com

- Mail your comments to:

  IBM Corporation, IBM Redbooks
  Dept. HYTD Mail Station P099
  2455 South Road
  Poughkeepsie, NY 12601-5400

Stay connected to IBM Redbooks

- Find us on Facebook:

  http://www.facebook.com/IBMRedbooks

- Follow us on Twitter:

  http://twitter.com/ibmredbooks

- Look for us on LinkedIn:

  http://www.linkedin.com/groups?home=&gid=2130806

- Explore new Redbooks publications, residencies, and workshops with the IBM Redbooks weekly newsletter:

  https://www.redbooks.ibm.com/Redbooks.nsf/subscribe?OpenForm

- Stay current on recent Redbooks publications with RSS Feeds:

  http://www.redbooks.ibm.com/rss.html

1

IBM Spectrum Discover overview

In this chapter, we provide a comprehensive overview of the IBM Spectrum Discover metadata management software platform. This overview helps storage administrators, data stewards, and data scientists understand the capabilities that are available to them with the addition of IBM Spectrum Discover.

This chapter includes the following topics:

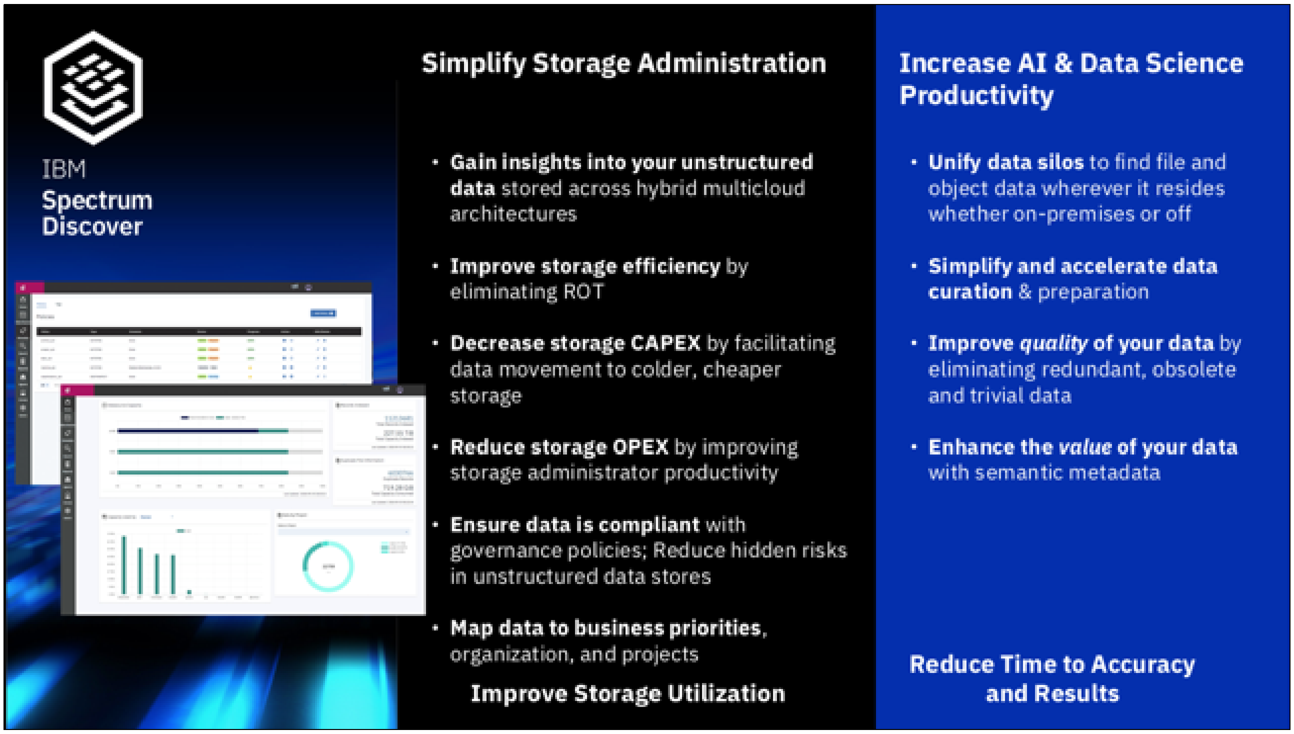
1.1 Introduction

More than 80% of all data that is collected by organizations is not in a standard relational database. Instead, it is trapped in unstructured documents, social media posts, machine logs, images, and other data. Many organizations face significant challenges to manage this deluge of unstructured data, such as the following examples:

- Pinpointing and activating relevant data for large-scale analytics.
- Lacking the fine-grained visibility that is needed to map data to business priorities.
- Removing redundant, obsolete, and trivial (ROT) data.
- Identifying and classifying sensitive data.

IBM Spectrum Discover is modern metadata management software that provides data insight for petabyte-scale file and Object Storage, storage on premises, and in the cloud. This software enables organizations to make better business decisions, and gain and maintain a competitive advantage.

The benefits of IBM Spectrum Discover are highlighted in Figure 1-1.



Simplify Storage Administration

- **Gain insights into your unstructured data** stored across hybrid multicloud architectures
- **Improve storage efficiency** by eliminating ROT
- **Decrease storage CAPEX** by facilitating data movement to colder, cheaper storage
- **Reduce storage OPEX** by improving storage administrator productivity
- **Ensure data is compliant** with governance policies; Reduce hidden risks in unstructured data stores
- **Map data to business priorities,** organization, and projects

Improve Storage Utilization

Increase AI & Data Science Productivity

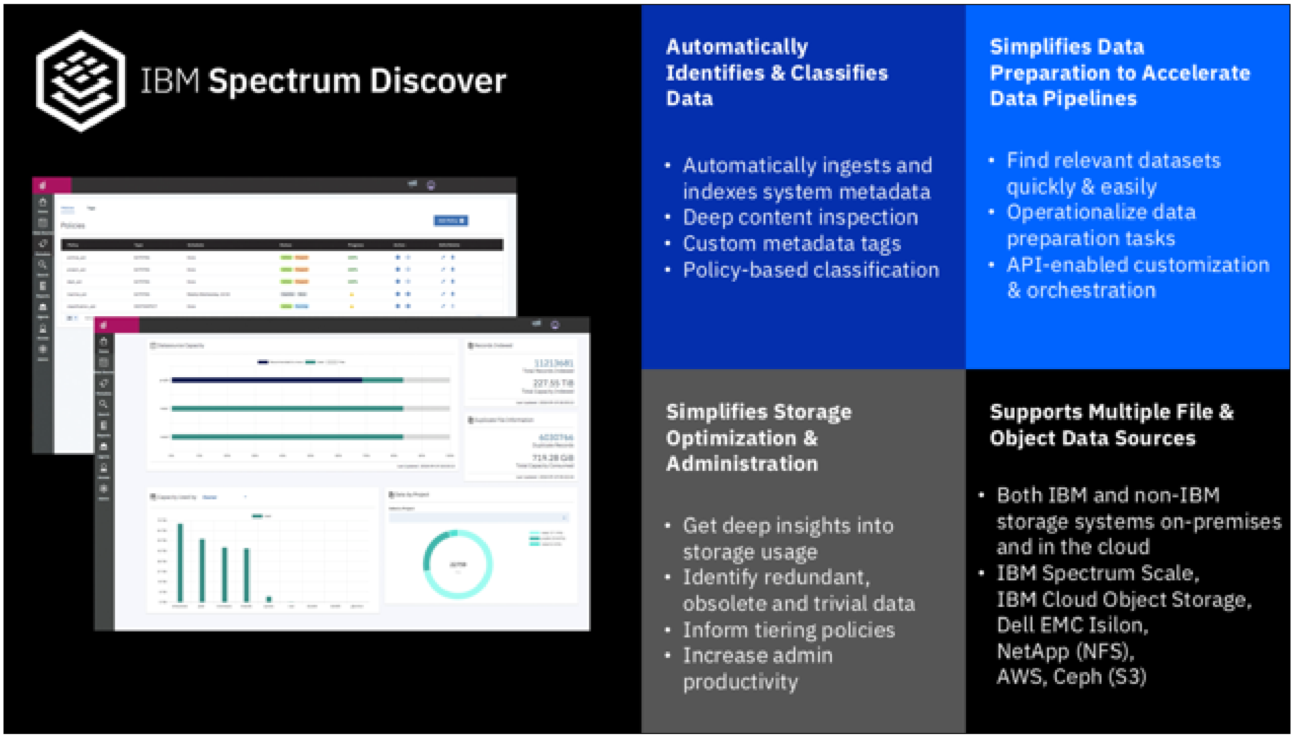
- **Unify data silos** to find file and object data wherever it resides whether on-premises or off
- **Simplify and accelerate data curation** & preparation
- **Improve *quality* of your data** by eliminating redundant, obsolete and trivial data
- **Enhance the *value* of your data** with semantic metadata

Reduce Time to Accuracy and Results

Figure 1-1 Benefits of IBM Spectrum Discover

IBM Spectrum Discover provides a rich metadata layer that enables storage administrators, data stewards, and data scientists to efficiently manage, classify, and gain insights from massive amounts of unstructured data. It also improves storage economics, helps mitigate risk, and accelerates large-scale analytics to create competitive advantage and speed-critical research.

The key capabilities of IBM Spectrum Discover are shown in Figure 1-2.



IBM Spectrum Discover

Automatically Identifies & Classifies Data

- Automatically ingests and indexes system metadata
- Deep content inspection
- Custom metadata tags
- Policy-based classification

Simplifies Data Preparation to Accelerate Data Pipelines

- Find relevant datasets quickly & easily
- Operationalize data preparation tasks
- API-enabled customization & orchestration

Simplifies Storage Optimization & Administration

- Get deep insights into storage usage
- Identify redundant, obsolete and trivial data
- Inform tiering policies
- Increase admin productivity

Supports Multiple File & Object Data Sources

- Both IBM and non-IBM storage systems on-premises and in the cloud
- IBM Spectrum Scale, IBM Cloud Object Storage, Dell EMC Isilon, NetApp (NFS), AWS, Ceph (S3)

Figure 1-2 Capabilities of IBM Spectrum Discover

1.2 Extensible platform for data oversight

IBM Spectrum Discover is an extensible platform that provides data insight for unstructured data by scanning and cataloging metadata from storage systems. The catalog can consist of metadata from numerous storage systems, on-premises or in the cloud, which enables a holistic view of data across the entire enterprise.

> **Note:** *Metadata* is data that describes data. Metadata captures the useful attributes of the associated source data to give the metadata context and meaning. For example, source data is a file or an object. The metadata is a set of attributes that are key-value pairs. The metadata records are associated with the file or object and are typically stored on the same system as the source data.
>
> System metadata is created and updated by the host system and not the application software. IBM Spectrum Discover enables the addition of tags that can capture non-system metadata-specific attributes.

After the initial scan of a data source and the population of the basic metadata information within the catalog is complete, the catalog can then be enriched. The enrichment comes from more information that is derived from the internal capabilities of IBM Spectrum Discover, purpose-built applications that use the extensible platform architecture, or custom tags that act as an extension of the system metadata that can contain organizational information beyond the view and limits of the source storage system.

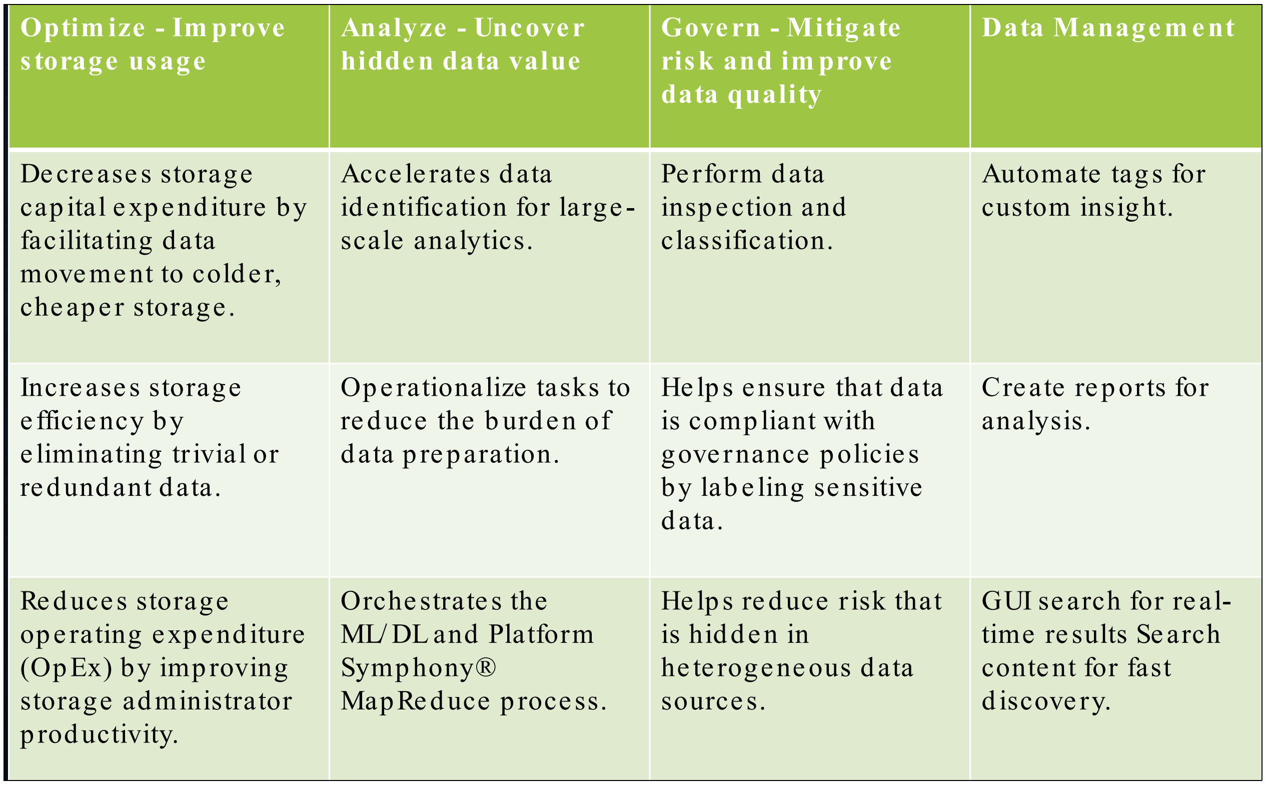
> **Note:** The source storage system metadata is not relocated to IBM Spectrum Discover, but remains in the original location. The IBM Spectrum Discover catalog is populated with a pointer to the original location.

These enrichments can all be carried out from the IBM Spectrum Discover GUI and by using IBM Spectrum Discover REST application programming interfaces (APIs). Using the enriched metadata that is provided by IBM Spectrum Discover enables storage administrators and users to generate various reports that are based on any of the information that is within the catalog. Also, the enriched metadata can be used with the extensible platform architecture to perform actions on selected data.

1.3 Benefits

IBM Spectrum Discover provides the following benefits (see Figure 1-3).

- Simplify data discovery and data heritage so organizations can much more easily identify, prepare, and optimize their data.
- Data IBM Insight® for analytics, governance, and optimization
- Help organizations derive greater business value from their unstructured data.
- Automates identification, classification, and tagging of unstructured data at scale.
- Provides comprehensive data insight by combining system and customer metadata to give data more context and meaning.

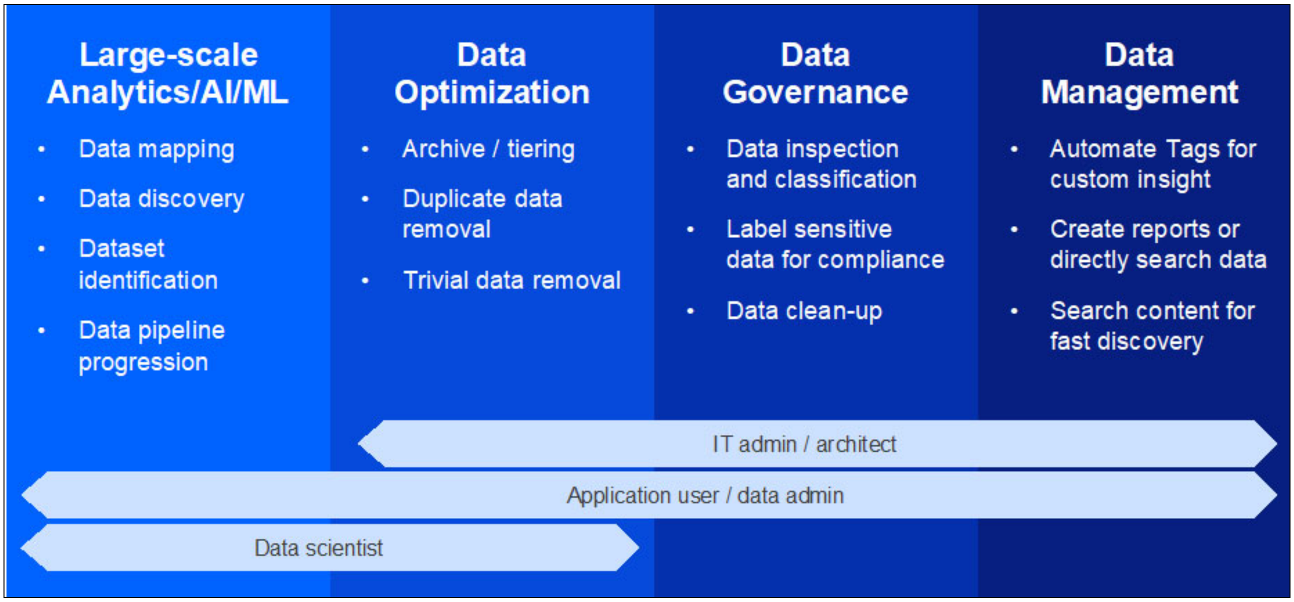


Optimize - Improve storage usage	Analyze - Uncover hidden data value	Govern - Mitigate risk and improve data quality	Data Management
Decreases storage capital expenditure by facilitating data movement to colder, cheaper storage.	Accelerates data identification for large-scale analytics.	Perform data inspection and classification.	Automate tags for custom insight.
Increases storage efficiency by eliminating trivial or redundant data.	Operationalize tasks to reduce the burden of data preparation.	Helps ensure that data is compliant with governance policies by labeling sensitive data.	Create reports for analysis.
Reduces storage operating expenditure (OpEx) by improving storage administrator productivity.	Orchestrates the ML/DL and Platform Symphony® MapReduce process.	Helps reduce risk that is hidden in heterogeneous data sources.	GUI search for real-time results Search content for fast discovery.

Figure 1-3 IBM Spectrum Discover benefits

1.4 IBM Spectrum Discover use cases

This section provides an overview of the key use cases for IBM Spectrum Discover, as shown in Figure 1-4.



Large-scale Analytics/AI/ML

- Data mapping
- Data discovery
- Dataset identification
- Data pipeline progression

Data Optimization

- Archive / tiering
- Duplicate data removal
- Trivial data removal

Data Governance

- Data inspection and classification
- Label sensitive data for compliance
- Data clean-up

Data Management

- Automate Tags for custom insight
- Create reports or directly search data
- Search content for fast discovery

IT admin / architect

Application user / data admin

Data scientist

Figure 1-4 Key use cases of IBM Spectrum Discover

1.4.1 Large-scale analytics/artificial intelligence/machine learning

This use case enables data scientists and researchers to uses the insights that are obtained by IBM Spectrum Discover to quickly and easily identify the relevant data for their experiment across billions of files and objects in the heterogeneous storage environment. Researchers and data scientists use the IBM Spectrum Discover GUI or REST APIs to perform data discovery that is based on the information in the IBM Spectrum Discover catalog.

The researchers or data scientists can identify, classify, and organize data by applying user-defined tags that are based on the system metadata (such as file path information). Researchers and data scientists can also set up policies to perform automated content-based keyword search extract key aspects from the data and index it into IBM Spectrum Discover.

After the data set is identified, it can be automatically registered as a data set in an upstream artificial intelligence (AI)/analytics application, such as IBM Cloud® Pak for Data and IBM Watson® Knowledge Catalog (WKC), and other items, such as IBM Watson Machine Learning Accelerator, IBM Maximo® Visual Inspection (formerly called IBM Visual Insights), or TensorFlow. The identified data set can also be automatically moved from a *warm* tier to a *hot* tier, such as Non-Volatile Memory Express (NVMe) backed storage that is attached to GPUs through InfiniBand before starting the data set registration.

Such a use case is described in Chapter 4., “Using artificial intelligence in medical imaging: JFR Challenge” on page 63.

Another key aspect of this use case is creating fully automated inferencing pipelines (ingest, curate, analyze, and infuse) that use the metadata event-driven architecture that is described in Chapter 3., “AI pipeline that uses IBM Spectrum Discover” on page 51, to automatically perform an inference operation on new data in the data lake and insert the resulting label information into the IBM Spectrum Discover catalog.

1.4.2 Data and storage optimization use case

This use case is understood and popular in the market because it pertains to the use of the system metadata that is collected across the heterogeneous storage environment to simplify storage administration and improve storage utilization by providing the following capabilities:

- Gain insights into your unstructured data that is stored across hybrid multi-cloud architectures.
- Improve storage efficiency by eliminating redundant, outdated, and trivial (ROT) data.
- Identify data by file type, size, and potentially duplicate data.
- Decrease storage CAPEX (capital expenditures) by facilitating data movement to colder, cheaper storage. Understand how your data is aging.
- Reduce storage OPEX (operating expenses) by improving storage administrator productivity.
- Ensure that data is compliant with governance policies. Reduce hidden risks in unstructured data stores.
- Map data to business priorities, organization, and projects.

These capabilities are accomplished by the built-in analytics capabilities of the IBM Spectrum Discover platform.

For more information about this use case, see 5.6.2, “Data Optimization use case: Data migration to tape for cost-efficient archiving” on page 124.

1.4.3 Data governance

IBM Spectrum Discover can classify data that is based on user-defined tags and the content of the data. The built-in Content Classification capability of IBM Spectrum Discover supports over 1000 different file and object types that use an open source tool called Apache Tika. Apache Tika is also embedded in some IBM Watson Natural Language Processing (NLP) products to perform a similar capability.

IBM Spectrum Discover also provides built-in support for Digital Imaging and Communications in Medicine (DICOM) medical images and genomics Variant Call Format (VCF) data.

The Content inspection capability can identify key fields, such as Social Security Number (SSN), phone numbers, account numbers, and many other fields to identify and tag content that contains Personally Identifiable Information (PII) and sensitive data by providing the following capabilities:

- Automate the identification and classification of documents that might potentially contain PII and sensitive data.
- Support for content-based data classification enables users to set up policies to automatically identify, classify, and categorize data, which can be used for specific business needs.

For the data steward and the chief information office (CIO), the ability to find and organize documents that are based on content greatly helps with their data administration efforts; for example, identifying data that might be subject to specific governance policies or compliance regulations.

Note: Enforcing data governance policies is *not* supported in IBM Spectrum Discover. Integration with the WKC component in IBM Cloud Pak® for Data (IBM CP4D) supports this capability.

For more information about this use case, see 5.6.1, “Data Governance use case: Data staging for high-performance processing” on page 116.

1.4.4 Data management

IBM Spectrum Discover offers enhanced unstructured data management capabilities that include automatically tagging data for custom insight, creating reports or directly searching data, searching content for fast discovery, and using these insights for data movement and data cleanup. By automatically adding tags to data that is based on institutional knowledge or derived information that is based on the metadata or content of the data, data stewards and data administrators can develop a much deeper understanding of the data that they need to manage.

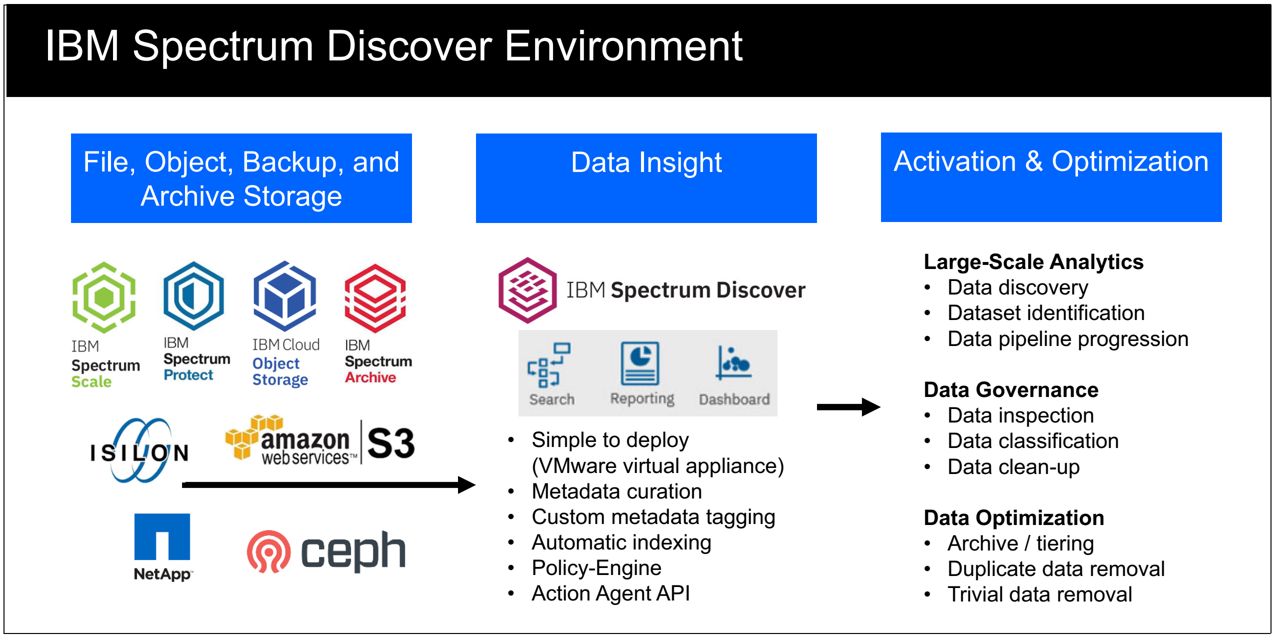
IBM Spectrum Discover can dynamically visualize or generate reports that are based on the metadata and associated tags of the files. So, users can link data sets that were not previously associated, or search those data sets for a deeper understanding of what that data is. It also can run searches in the data content for attributes that are relevant to the business.

IBM Spectrum Discover provides a search bar and a more advanced search pane to help users quickly find subsets of records that are indexed. Search results are displayed in a columnar table that contains information that is correlated to search criteria. What a user can see or not see is determined by using role-based access control (RBAC). All of these features enable a strong decision-making capability around the management of the data by using the metadata about the data or data from the content files in a simple searchable format.

For more information about use cases for categorization and searching of health and medical information, see Chapter 2., “Generic imagery use cases” on page 29.

1.5 Architecture

IBM Spectrum Discover is an extensible platform that provides exabyte-scale data ingestion, data visualization, data activation, and business-oriented data mapping from across the enterprise. It enables data management at a broader level, which enables a more precise view of the "who, what, where, when, and why" aspects of an organization's data rather than the myopic view of data from a single storage system. The IBM Spectrum Discover architecture is shown in Figure 1-5.


IBM Spectrum Discover Environment

File, Object, Backup, and Archive Storage

Data Insight

Activation & Optimization

- Simple to deploy (VMware virtual appliance)
- Metadata curation
- Custom metadata tagging
- Automatic indexing
- Policy-Engine
- Action Agent API

Large-Scale Analytics
- Data discovery
- Dataset identification
- Data pipeline progression

Data Governance
- Data inspection
- Data classification
- Data clean-up

Data Optimization
- Archive / tiering
- Duplicate data removal
- Trivial data removal

Figure 1-5 IBM Spectrum Discover architecture

IBM Spectrum Discover can scan or ingest billions of records in the course of a day. Ingesting data consists of reading metadata information from the source storage system and automatically cataloging the information into the IBM Spectrum Discover platform. This feature enables IBM Spectrum Discover to deliver results of complex queries or multi-faceted searches against the metadata information ultrafast, even when the catalog contains billions of entries. The search results are visualized by the GUI's drill-down dashboard nearly instantaneously.

IBM Spectrum Discover easily connects to the following data sources to rapidly ingest, consolidate, and index metadata for billions of files and objects:

- IBM Cloud Object Storage
- IBM Spectrum Scale and IBM Elastic Storage® Server
- IBM Spectrum Protect
- IBM Spectrum Archive
- Isilon Network File System (NFS) exports and Server Message Block (SMB) shares
- NetApp NFS exports and Server Message Block (SMB) shares
- Amazon Web Services (AWS) Simple Storage Service (S3)
- Red Hat Ceph S3

IBM Spectrum Discover provides a rich metadata layer that enables storage administrators, data stewards, and data scientists to efficiently manage, classify, and gain insights from massive amounts of unstructured data. It improves storage economics, helps mitigate risk, and accelerates large-scale analytics to create competitive advantage and speed critical research.

IBM Spectrum Discover is extensible, which provides a mechanism for communication with applications that can provide even greater insight into selected data by interrogating the contents of the full data, rather than just the metadata.

The IBM Spectrum Discover platform embeds an Apache Kafka instance, which enables a communication stream that can publish and subscribe to streams of records, similar to an enterprise messaging system. The streams of records are processed as they occur.

This feature enables IBM Spectrum Discover to enhance the contents of the catalog with the results of the inspection of the data when they become available. However, by using this same mechanism, real-time streaming data pipelines reliably get data between systems or applications that can enable even more capabilities, such as moving or deleting data. In addition to reliability, this extensible design provides an amazing amount of flexibility, so the possible use cases for IBM Spectrum Discover are nearly limitless.

To capitalize on the flexible, extensible architecture of IBM Spectrum Discover, the following APIs are provided:

- Policy management API
- Custom application API

The custom application API is used to establish the Kafka topic interfaces that are used for messaging and carrying out work to be done on selected data, hence the name *action agent*.

The policy management API is a RESTful web service that creates, lists, updates, and deletes policies. The policy management API also provides the means to start a policy immediately or schedule it to run on a schedule.

Business-oriented data mapping can be carried out by the policy management API. An example of business-oriented data mapping is adding a project name to the catalog that is based on the location (or path) to the data.

1.5.1 Role-based access control

IBM Spectrum Discover provides access to resources that are based on roles. Customers can restrict access to information based on roles. The role that is assigned to a user or group determines their privileges.

Users and groups can be associated with collections, which use policies that determine the metadata that is available to view. User and group access can be authenticated by IBM Spectrum Discover, an LDAP server, or the IBM COS System. The administrator can manage the user access functions.

Roles

Roles determine how users and groups access records or the IBM Spectrum Discover environment. If a user or group is assigned to multiple roles, the least restrictive role is applicable. For example, if you are assigned the role of Data User and you are also assigned the role of a Data Admin, you have the privileges of a Data Admin.

The following roles are available:

Admin This role can create users, groups, and collections, and manage LDAP connections. This role can use the Application Management APIs to install, upgrade, or delete IBM Spectrum Discover applications that use the IBM Spectrum Discover API service.

Data Admin This role can access all metadata that is collected by IBM Spectrum Discover, and is not restricted by policies or collections. This role can also define tags and policies, including policies that assign a collection value to a set of records.

Collection Admin Manages data within a list of collections and user and group access to those collections. The Collection Admin role is a bridge between the Data Admin role and the Data User role. Users with the Collection Admin role can create, update, and delete the policies for the collections that they administer and view, update, and delete policies of data users for the collections they administer.

Data User This role can access metadata that is collected by IBM Spectrum Discover, but metadata access can be restricted by policies in the collections that are assigned to users in this role. This role can also define tags and policies that is based on the collections to which the role is assigned.

Service User IBM service and support personnel.

1.5.2 Data source connections

A data source connection specifies the parameters for cataloging metadata from a source system to IBM Spectrum Discover.

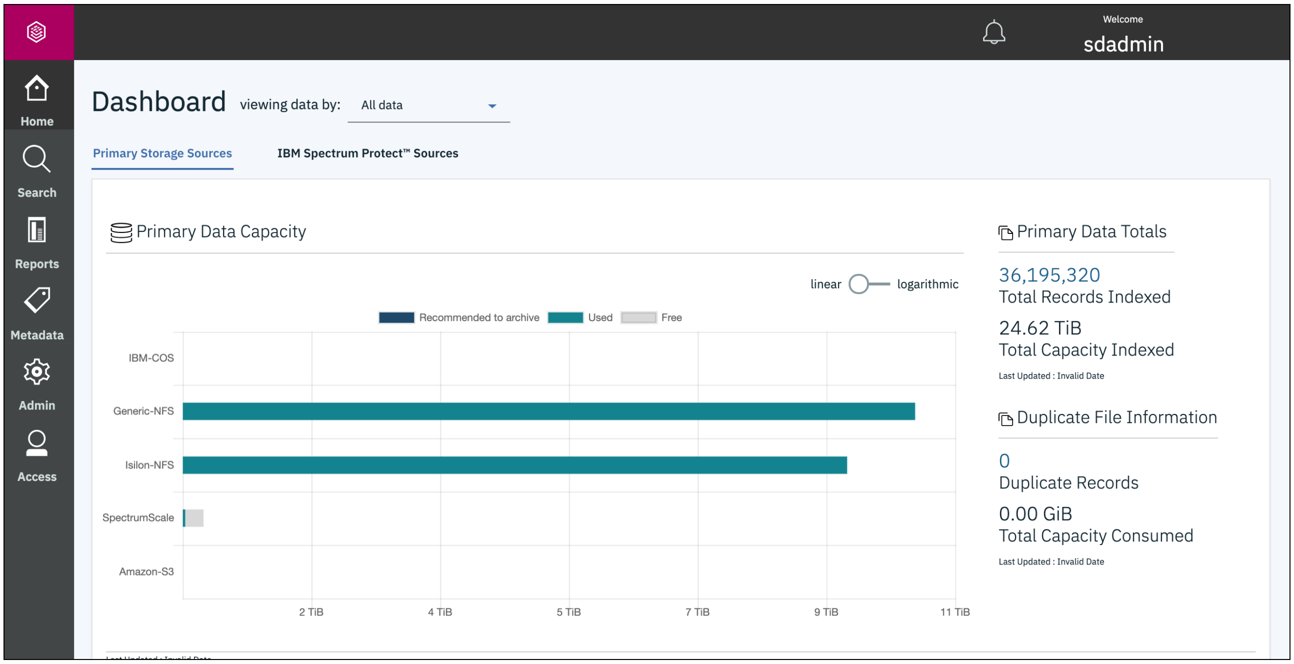
Without the proper connection information, ingesting metadata from a connected system fails. You can use the Data Source Connections page of the GUI to view connection information for the data sources that are connected to your environment.

1.5.3 GUI

The IBM Spectrum Discover GUI is a portal that is used for administration purposes and running data searches, generating reports, managing policies and tags, and managing user access.

Note: Based on a user’s role, they might not have access to all areas of the GUI.

The IBM Spectrum Discover GUI Dashboard is shown in Figure 1-6.



viewing data by: All data

36,195,320
Total Records Indexed

24.62 TiB
Total Capacity Indexed

Last Updated : Invalid Date

0
Duplicate Records

0.00 GiB
Total Capacity Consumed

Last Updated : Invalid Date

Figure 1-6 IBM Spectrum Discover GUI Dashboard

Understanding size and capacity differences

IBM Spectrum Discover collects size and capacity information.

Consider the following points:

- *Size* refers to the size of a file or object in bytes.
- *Capacity* refers to the amount of space the file or object uses on the source storage in bytes.

For objects, size and capacity values always match. However, for files, the size and capacity values can be different because of file system block overhead or sparsely populated files.

> **Note:** Storage protection overhead (such as RAID values or erasure coding) and replication overhead are not captured in the capacity values.

1.5.4 Reports

Reports for IBM Spectrum Discover are grouped or non-grouped:

- *Grouped* reports feature information for count and sum in columns.
- *Non-grouped* reports feature information in rows.

Data Curation Reports are a way for administrators to view the state of their storage environment in different ways. They can range from high-level grouped information to individual record-level information.

For example, you can sort a report by owner, project, and department, or you can generate a list of records that meet specific criteria.

For more information about generating reports, see IBM Knowledge Center.

1.6 A deeper look at metadata

Metadata is at the heart of IBM Spectrum Discover (collecting, creating, analyzing, reporting, and other activities). Metadata collection and creation involve the use of tags, policies, and agents. Management and exploration of metadata also involves tags and policies, and searching, visualization, and reporting.

IBM Spectrum Discover users can collect, define, explore, and report metadata, as described next.

1.6.1 Cataloging metadata

Cataloging metadata in IBM Spectrum Discover is the process of ingesting and indexing the system metadata records from a source. Cataloging metadata transforms the metadata records into data that the user can reference and use.

The system metadata collection depends upon what type of storage system is being scanned. File system scans result in a different set of system metadata than Object Storage systems. File system metadata include tags, such as size, owner, path, file name, access time (atime), and modification time (mtime). In contrast, Object Storage system metadata includes bucket name, object name, object length, content type, system UUID, and other data.

IBM Spectrum Discover features live event notifications for IBM COS data connections. These notifications are triggered by user actions on the source data and result in updating the system metadata on the IBM Spectrum Discover system. Example actions include reading, writing, moving, and deleting data, and changing permissions or ownership. The events generate a metadata record in real time that is stored in IBM Spectrum Discover.

1.6.2 Enriching metadata

IBM Spectrum Discover can enrich the metadata from supported platforms with more information by using custom tags, policies, and action agents.

Tags

A *tag* is a custom metadata field (or key-value pair) that is used to supplement storage system metadata with organization-specific information.

An organization might segment their storage by project or by chargeback department. Those facets do not show in the system metadata, and the storage systems do not provide management and reporting capabilities based on those organizational concepts. By using custom tags, you can store more information, and manage, report, and search for data by using that organizationally important information.

Types of tags

The following types of tags are available:

- Categorization tags: Contain values, such as project, department, and security classification. Categorization tags can be open or restricted. If it is open, listed selections can be used. If it is closed, selection is limited to true or false.
- Characteristic tags: Can contain any value that is needed to describe or classify the object. Also can contain long descriptive values and the size limit is 4 KB.

1.6.3 Policies and user-defined metadata

One of the most powerful aspects of IBM Spectrum Discover is its ability to accommodate metadata that is created by users of the associated data or the IBM Spectrum Discover system. You can collect metadata by logically combining metadata, by using header extraction tools (such as Apache Tika), or defining your own software agents to extract metadata unique to your business.

Policies are used to add information about the source data that is indexed in IBM Spectrum Discover. A policy determines the set of files to add tag values to or send to an action agent through filtering criteria. The policies give the user the ability to run actions once or on a set schedule.

Policies work in batches and can be paused, resumed, stopped, and restarted. The user can control the load on the IBM Spectrum Discover system and on the source storage system in the case of DEEPINSPECT policies.

A policy includes the following components:

Policy ID Name of the policy.

Filter Selects a set of documents to work.

Action ID, parameters, and schedule.

Three basic categories of policies support user-defined metadata collection policies:

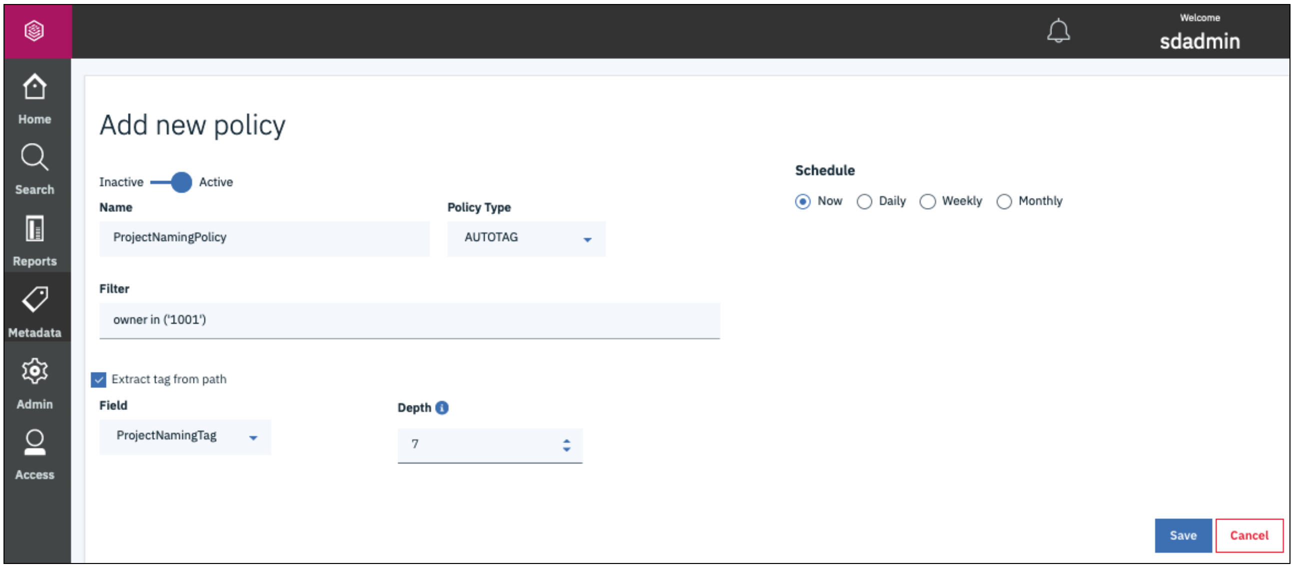
- AUTOTAG
- CONTENTSEARCH
- DEEPINSPECT

The following sections describe creating these policies. Tags are a prerequisite for policies because the purpose of a policy is to assign values to one or more tags.

Policies feature a few common attributes, such as the name of the policy, the filter that is associated with the policy, and one or more tags to which the policy assigns values. When defining a filter, you use the same syntax as the `WHERE` clause in a standard SQL query. Ultimately, the filter identifies a subset of objects to which the policy applies.

The first step for any policy is to create it.

A policy tags a set of records based on filter criteria with a predefined set of tags. The ease of creating an AUTOTAG policy by using the IBM Spectrum Discover GUI is shown in Figure 1-7.



Add new policy

Inactive ● Active

Name: ProjectNamingPolicy

Policy Type: AUTOTAG

Schedule: ● Now ○ Daily ○ Weekly ○ Monthly

Filter: owner in ('1001')

☑ Extract tag from path

Field: ProjectNamingTag

Depth: 7

Figure 1-7 Autotag policy

AUTOTAG

In an example, we define the restricted tag `oldVideo` to have only values of `TRUE` and `FALSE`. Select the **Metadata** icon in left pane of the IBM Spectrum Discover GUI and select **Policies tab** → **Add Policy**. In the resulting “Add new policy” window, complete the following steps:

1. Choose and enter the name for the policy.
2. Select **AUTOTAG** as the Policy Type.
3. Define the criteria for assigning the tag the wanted value; that is, define the Filter for the policy.
4. Select **Add tag.**
5. Select the wanted tag by selecting the **Tag** drop-down list.
6. Select the wanted value by selecting the **Values** drop-down list.

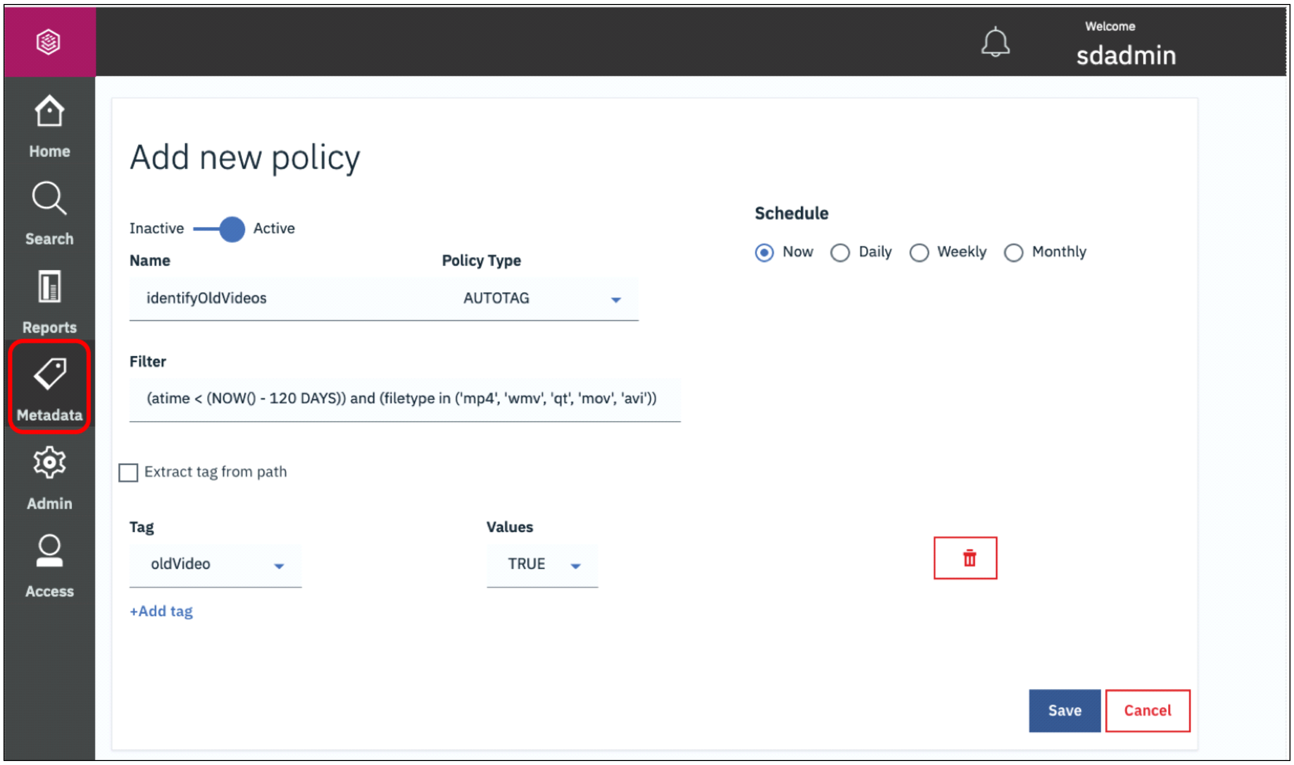
The AUTOTAG policy is basic in its operation. From a logical perspective, it can be represented as shown in the following example:

```
if (<filter>) then <tag> = <value>
```

For example, consider the defined policy that is shown in Figure 1-8. In this example, the policy `identifyOldVideos` is an AUTOTAG policy that assigns a value of `TRUE` to the tag **`oldVideo`** if the file satisfies the condition (filter):

```
(atime < (NOW() - 120 DAYS)) and (filetype in ('mp4', 'wmv', 'qt', 'mov', 'avi'))
```

For example, if the file was not accessed in 120 days and is a video file (that is, it has a file extension of `.mp4`, `.wmv`, `.qt`, `.mov`, or `.avi`) set the file's `oldVideo` tag to value of `TRUE`.



Add new policy

Inactive ● Active

Name: identifyOldVideos

Policy Type: AUTOTAG

Schedule: ● Now ○ Daily ○ Weekly ○ Monthly

Filter: (atime < (NOW() - 120 DAYS)) and (filetype in ('mp4', 'wmv', 'qt', 'mov', 'avi'))

□ Extract tag from path

Tag: oldVideo

Values: TRUE

Figure 1-8 Defining an AUTOTAG policy with IBM Spectrum Discover

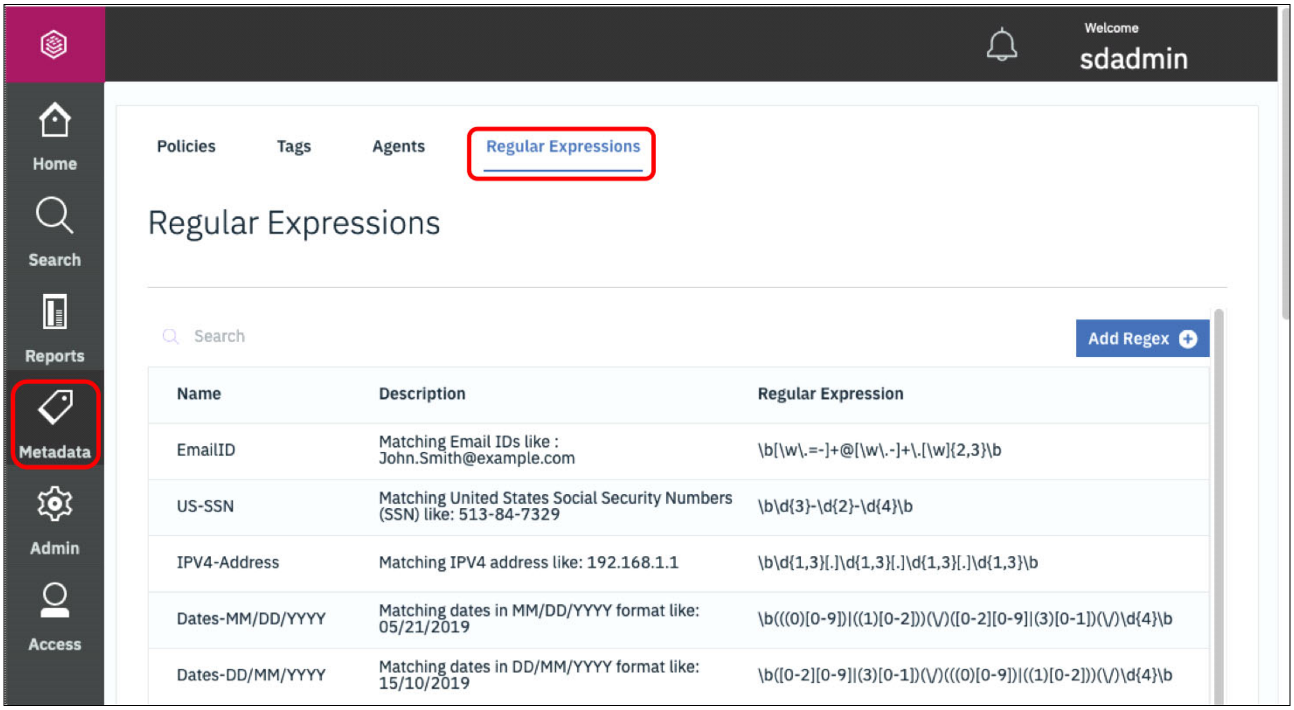
CONTENTSEARCH

Beginning with Version 2.0.1, IBM Spectrum Discover can enrich metadata through content inspection of source data by using the built-in CONTENTSEARCH agent. To use this function, you define regular expressions (regex) to search for and create policies that use these regex.

When the policy runs, the files or objects are retrieved from the source system by the CONTENTSEARCH agent, converted to text format if necessary, and searched by using the defined regex. The results of the search are returned to IBM Spectrum Discover and the metadata of the files that are updated according to the policy's definition.

Regular expressions,

Before defining your CONTENTSEARCH policy, identify the regex that you want to use with the policy. To do so, select the **Metadata** icon on the left pane of the IBM Spectrum Discover GUI and then, select the **Regular Expressions** tab on the Metadata page. The list of available regex as displayed by the GUI is shown in Figure 1-9 on page 16. Search through the list for expressions that apply to your scenario.



Regular Expressions

Search

Add Regex

Name	Description	Regular Expression
EmailID	Matching Email IDs like : John.Smith@example.com	\b[\w\.=-]+@[\w\.-]+\.[\w]{2,3}\b
US-SSN	Matching United States Social Security Numbers (SSN) like: 513-84-7329	\b\d{3}-\d{2}-\d{4}\b
IPV4-Address	Matching IPV4 address like: 192.168.1.1	\b\d{1,3}[.]\d{1,3}[.]\d{1,3}[.]\d{1,3}\b
Dates-MM/DD/YYYY	Matching dates in MM/DD/YYYY format like: 05/21/2019	\b(((0)[0-9])|((1)[0-2]))(\/)([0-2][0-9]|(3)[0-1])(\/)\d{4}\b
Dates-DD/MM/YYYY	Matching dates in DD/MM/YYYY format like: 15/10/2019	\b([0-2][0-9]|(3)[0-1])(\/)(((0)[0-9])|((1)[0-2]))(\/)\d{4}\b

Figure 1-9 IBM Spectrum Discover provides a predefined list of regular expressions

Table 1-1 lists the IBM Spectrum Discover V2.0.3 Regular Expressions that are included with the base installation.

Table 1-1 Regular Expressions that are included with the base installation

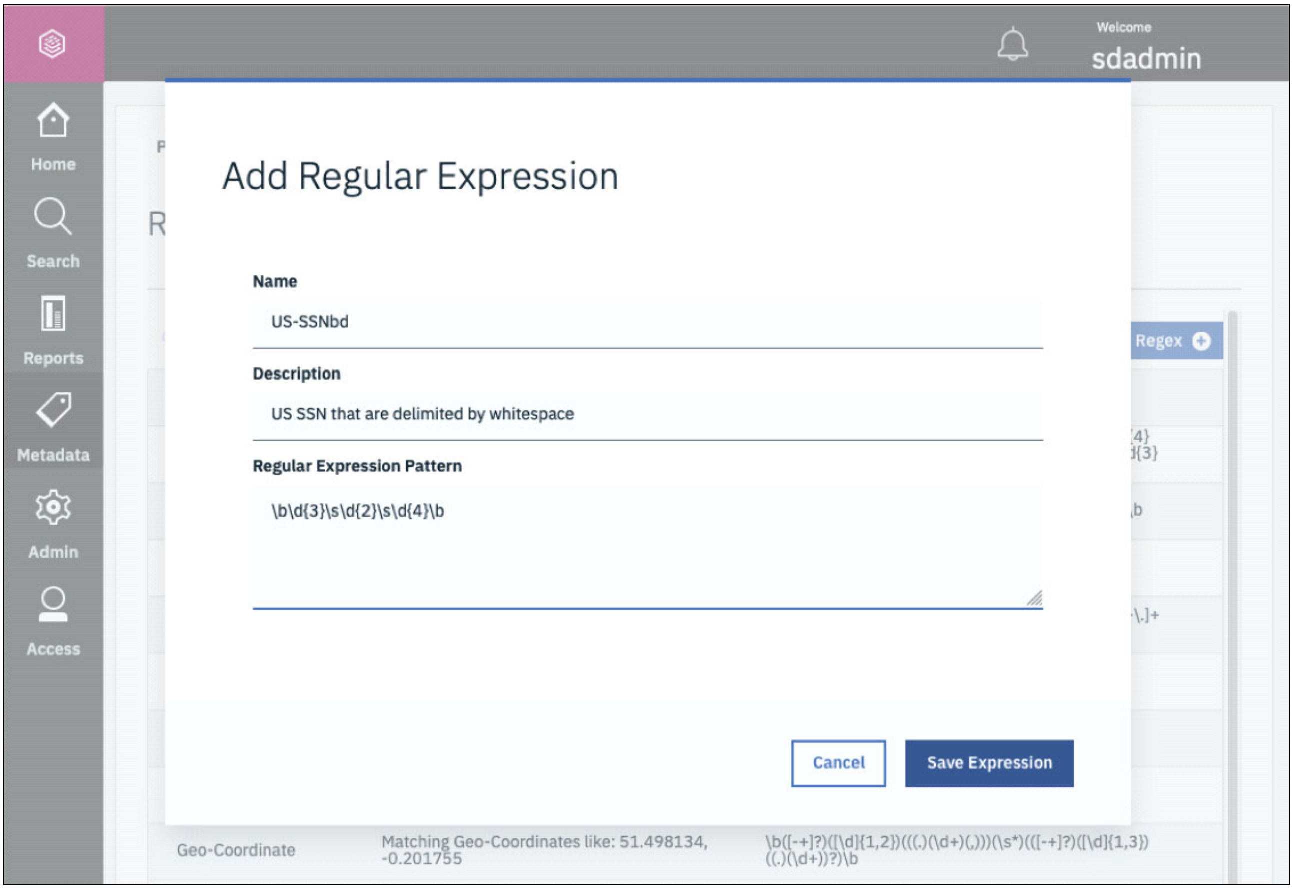
Name	Description	Regular Expression (Pattern)
VisaCard	Matching Visa Card numbers, such as: 4563-7568-5698-4587	\b([4]\d{3}[\s]\d{4}[\s]\d{4}[\s]\d{4}|[4]\d{3}[-]\d{4}[-]\d{4}[-]\d{4}|[4]\d{3}[.]\d{4}[.]\d{4}[.]\d{4}|[4]\d{3}\d{4}\d{4}\d{4})\b
AmexCard	Matching American Express Card numbers, such as: 340000000000009	\b3[47][0-9]{13}\b
MasterCard	Matching MasterCard numbers, such as: 5258704108753590	\b(?:5[1-5][0-9]{2}|222[1-9]|22[3-9][0-9]|2[3-6][0-9]{2}|27[01][0-9]|2720)[0-9]{12}\b
USZIPCode	Matching United States ZIP codes such as: 97580	\b((\d{5}-\d{4})|(\d{5})|([A-Z]\d[A-Z]\s\d[A-Z]\d))\b
URL	Matching URLs, such as: http://www.test.com/dir/filename.jpg?var1=foo#bar&	\b((http[s]?|ftp):\/)?\/?([^:\/\s]+)((\/\w+)*\/)([\w\-\.]+[^#?\s]+)(.*)?(#[\w\-]+)?\b
EmailID	Matching Email IDs, such as: John.Smith@example.com	\b[\w\.=-]+@[\w\.-]+\.[\w]{2,3}\b
US-SSN	Matching United States, such as: 513-84-7329	\b\d{3}-\d{2}-\d{4}\b
IPV4-Address	Matching IPV4 address, such as: 192.168.1.1	\b\d{1,3}[.]\d{1,3}[.]\d{1,3}[.]\d{1,3}\b
Dates-MM/DD/YYYY	Matching dates in MM/DD/YYYY format, such as: 05/21/2019	\b(((0)[0-9])|((1)[0-2]))(\/)([0-2][0-9]|(3)[0-1])(\/)\d{4}\b

Name	Description	Regular Expression (Pattern)
Dates-DD/MM/YYYY	Matching dates in DD/MM/YYYY format, such as: 15/10/2019	\b([0-2][0-9]|(3)[0-1])(\/)(((0)[0-9])|((1)[0-2]))(\/)\d{4}\b
Currency	Matching currency, such as: 123, 25.50	\b(\d+(\.\d{2})?)\b
CVV-Number	Matching Credit Card Verification value number, such as: 670, 0927	\b([0-9]{3,4})\b
CreditCardExpirationDate	Matching Credit Card Expiration Date, such as: 11/12	\b\d{2}\/\d{2}\b
CanadianSIN	Matching Canadian Social Insurance Number, such as: 123-456-789	\b(\d{3}[\s]\d{3}[\s]+\d{3})\b|\b(\d{3}[-]\d{3}[-]+\d{3})\b
Geo-Coordinate	Matching Geo-Coordinates, such as: 51.498134, -0.201755	\b([-+]?)([\d]{1,2})(((.)(\d+)(,)))(\s*)(([-+]?)([\d]{1,3})((.)(\d+))?)\b

For more information, see IBM Knowledge Center.

This basic list of regex might meet your needs. If it does not meet your needs, you can create a regex by selecting the **Add Regex** option, which opens a window in which you define a regex to be used by IBM Spectrum Discover. Enter a suitable name, description, and the regex pattern. Many useful regex and tutorials for creating them are available.

Figure 1-10 shows an example of creating a regex.



Add Regular Expression

Name
US-SSNbd

Description
US SSN that are delimited by whitespace

Regular Expression Pattern
\b\d{3}\s\d{2}\s\d{4}\b

Geo-Coordinate Matching Geo-Coordinates like: 51.498134, -0.201755 \b([-+]?)([\d]{1,2})(((.)(\d+)(,)))(\s*)(([-+]?)([\d]{1,3})((.)(\d+))?)\b

Figure 1-10 Creating a regular expression for use with IBM Spectrum Discover CONTENTSEARCH policies

Defining the CONTENTSEARCH policy

After you verify that the regex is available for your CONTENTSEARCH policy, proceed to defining the policy. As with other policies, browse to the **Metadata** page, select the **Policies** tab, and then select the **Add Policy** option.

Continuing with our example, suppose that we want a policy that identifies an SSN. To define the policy, complete the following steps:

1. Name the policy for example, `identifyFilesWithSSN`.
2. Select the policy type of **CONTENTSEARCH**.
3. Specify a filter pertinent to your scenario.
4. Select the agent that was created for CONTENTSEARCH policies. In this example scenario, that agent is `contentsearchagent`.
5. Identify the tags that the policy is to assign values to, for example, **ssn**.
6. Select the regex that the policy uses when searching the content of your data storage. In this example, we use two types of regex that identify US SSNs.
7. Specify whether you want the policy to set the value of **ssn** to the nine-digit SSN found or if you want to use the tag to identify whether it found an SSN (`True`) or not (`False`). In our example, we choose to identify whether the file includes an SSN.

After saving the policy configuration, your CONTENTSEARCH policy is ready to use.

For more information about this capability, see 2.1.2, “Using CONTENTSEARCH policy to extract DICOM metadata” on page 30.

DEEPINSPECT with custom agents

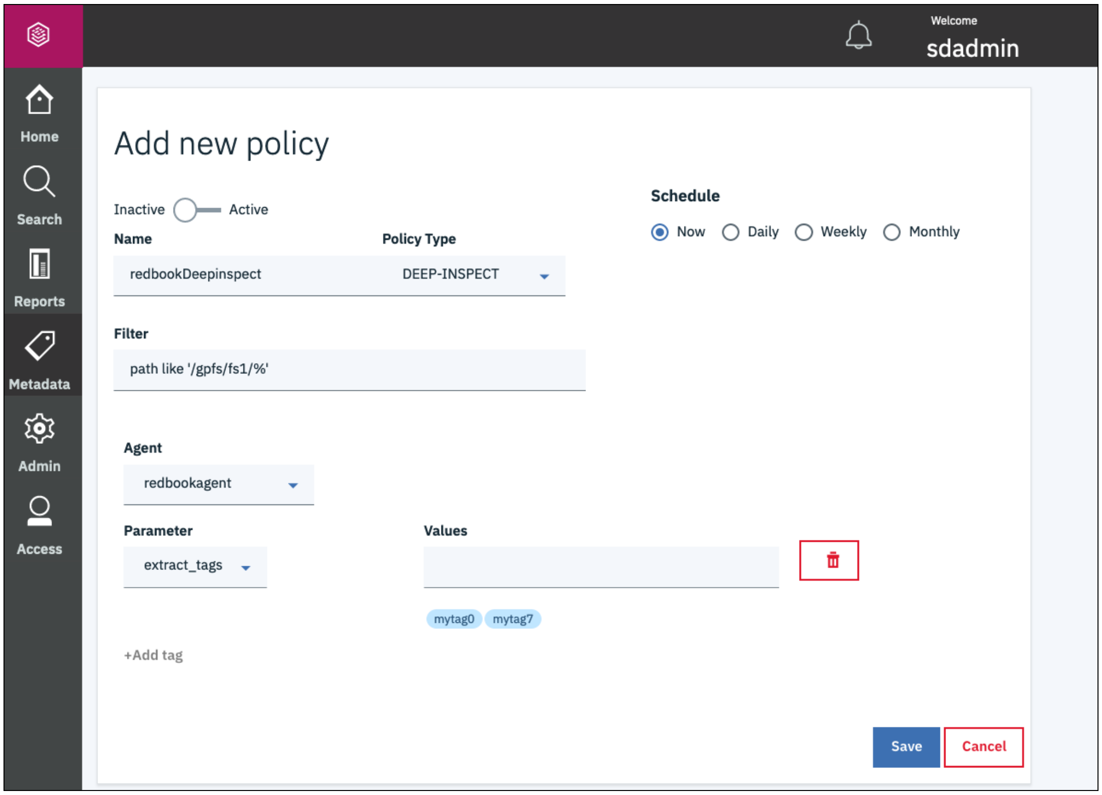
DEEPINSPECT is a policy that passes lists of files that are based on filter criteria to an analytics agent. It opens the source data file and extracts metadata information from it. The policy passes the data back to IBM Spectrum Discover in the form of tags so that you can run a search and perform the following tasks:

- Set up a filter to perform a search query that finds the candidates to apply the policy. For example, you can set an action for filtered candidates:

  ```
  AUTOTAG, tag1: value, tag2: value
  ```

- Set a schedule to apply the policy by specifying the following methods:
  - Immediately
  - Periodically

DEEPINSPECT policies are easily built and run from within the IBM Spectrum Discover GUI, as shown in Figure 1-11.



Add new policy

Inactive ○ Active

Name: redbookDeepinspect

Policy Type: DEEP-INSPECT

Schedule: ● Now ○ Daily ○ Weekly ○ Monthly

Filter: path like '/gpfs/fs1/%'

Agent: redbookagent

Parameter: extract_tags

Values: mytag0 mytag7

Figure 1-11 DEEPINSPECT policy from the IBM Spectrum Discover GUI

In some scenarios, collecting the wanted metadata cannot be accomplished by using AUTOTAG or CONTENTSEARCH policies. Consider a file format that facilitates a specific or proprietary API to collect metadata that is based on that format.

Another example is one in which files include location codes that must be converted to geographical metadata by using a mechanism that performs the conversion of location code to that metadata. To facilitate such use cases (those that do not fit easily into the AUTOTAG or CONTENTSEARCH policy paradigms), IBM Spectrum Discover provides an API to facilitate more complex metadata collection.

For more information about API, see IBM Knowledge Center.

To implement a custom metadata collection agent, complete the following steps:

1. Using the IBM Spectrum Discover GUI, define the tag or tags that are associated with the metadata that you plan to collect.
2. Develop an executable (custom agent) by using the REST API that collects the metadata and communicates it to your IBM Spectrum Discover system. This process includes registering your agent, establishing connectivity, and developing the necessary software to communicate with the IBM Spectrum Discover platform.
3. Define your DEEPINSPECT policy by using the IBM Spectrum Discover GUI.

Defining tags for use with DEEPINSPECT policies

Defining tags to be used with your DEEPINSPECT agent and policy is analogous to defining tags for any policy. Because your user-defined agent and policy depend on the tag names, plan and define the tags before moving on to agent development and policy definition.

Developing a DEEPINSPECT agent to collect metadata

This section describes how to deploy a custom metadata collection agent.

Developing a customer metadata collection agent depends on the IBM Spectrum Discover REST API. For more information about using the API for action agent management, see IBM Knowledge Center.

To operate the DEEPINSPECT agent by using a IBM Spectrum Discover system, complete the following steps:

1. Obtain the authorization token from the IBM Spectrum Discover system by using the credentials of an IBM Spectrum Discover data admin user.
2. Register your action agent by creating a JSON file by using the details of your action agent.
3. Develop the DEEPINSPECT agent.

   You can use any programming language that you choose to deploy the DEEPINSPECT agent. At a high level, your agent must establish a connection with the Kafka work and completion topics (message queues). For more information, see Apache Kafka.
4. To create the DEEPINSPECT policy, complete the following steps:
   a. Browse to the **Add new policy** pane by selecting the **Metadata** icon in the left pane. Select the **Policies** tab, and then, select the **Add Policy** option.

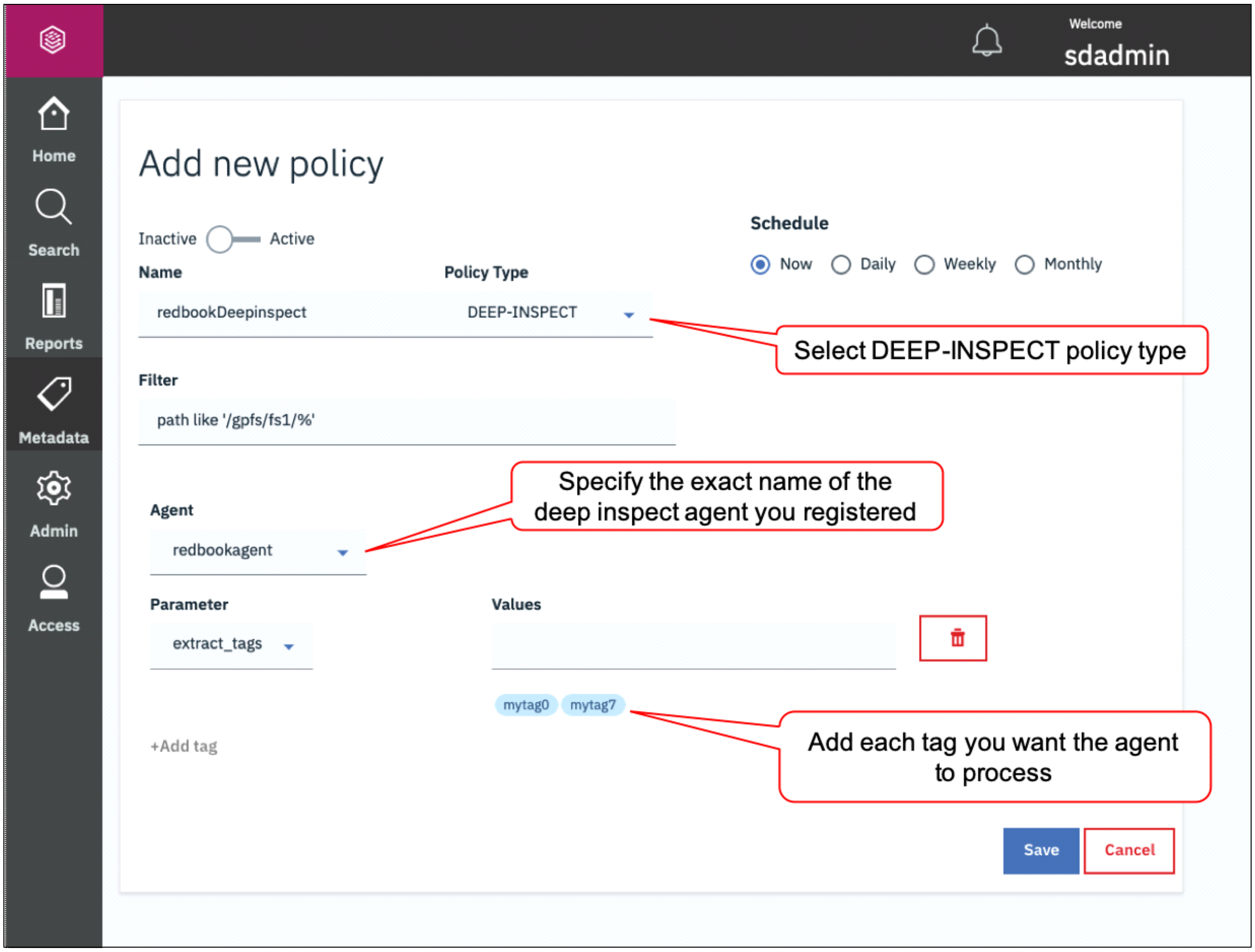
   b. Enter a name for the policy. Select **DEEP-INSPECT** in the Policy Type drop-down list and select your agent from the **Agent** drop-down list.

   c. Select the **+ Add tag** link, which displays selections for Parameter and Values. In the **Parameter** drop-down list, select **extract_tags**.

   d. Add each tag that you want the DEEPINSPECT agent to assign values to the list Values.

An example of defining a DEEPINSPECT policy that corresponds to our example in this section is shown in Figure 1-12.

> **Note:** For each tag that you want to add to the Values list, enter the tag name and then press Return or Enter so that the tag names appear in light blue bubbles below the Values entry bar.



Add new policy

Inactive ○ Active

Name: redbookDeepinspect

Policy Type: DEEP-INSPECT

Schedule: ● Now ○ Daily ○ Weekly ○ Monthly

Filter: path like '/gpfs/fs1/%'

Agent: redbookagent

Parameter: extract_tags

Values: mytag0 mytag7

Select DEEP-INSPECT policy type

Specify the exact name of the deep inspect agent you registered

Add each tag you want the agent to process

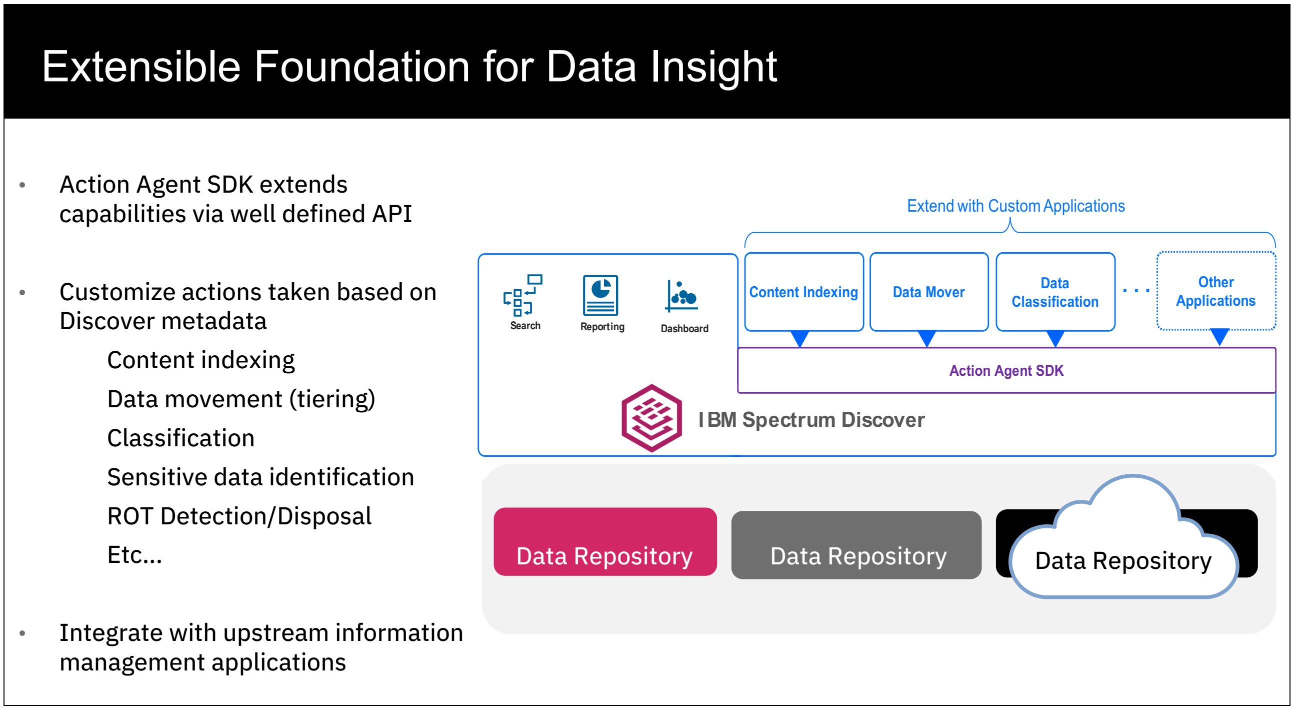
Figure 1-12 Adding a DEEPINSPECT policy to an IBM Spectrum Discover system

After the policy is defined, running that policy sends a list of files or objects to your DEEPINSPECT agent for processing.

For more information, see Chapter 4, “Using artificial intelligence in medical imaging: JFR Challenge” on page 63.

1.6.4 IBM Spectrum Discover Application Catalog and Software Development Kit

IBM Spectrum Discover enables users to customize their metadata extraction capability by using the DEEPINSPECT policy (see “DEEPINSPECT with custom agents” on page 18), as shown in Figure 1-13. That ability is facilitated by the IBM Spectrum Discover Software Development Kit (SDK) to aid in rapid development of agents is provided.



Extensible Foundation for Data Insight

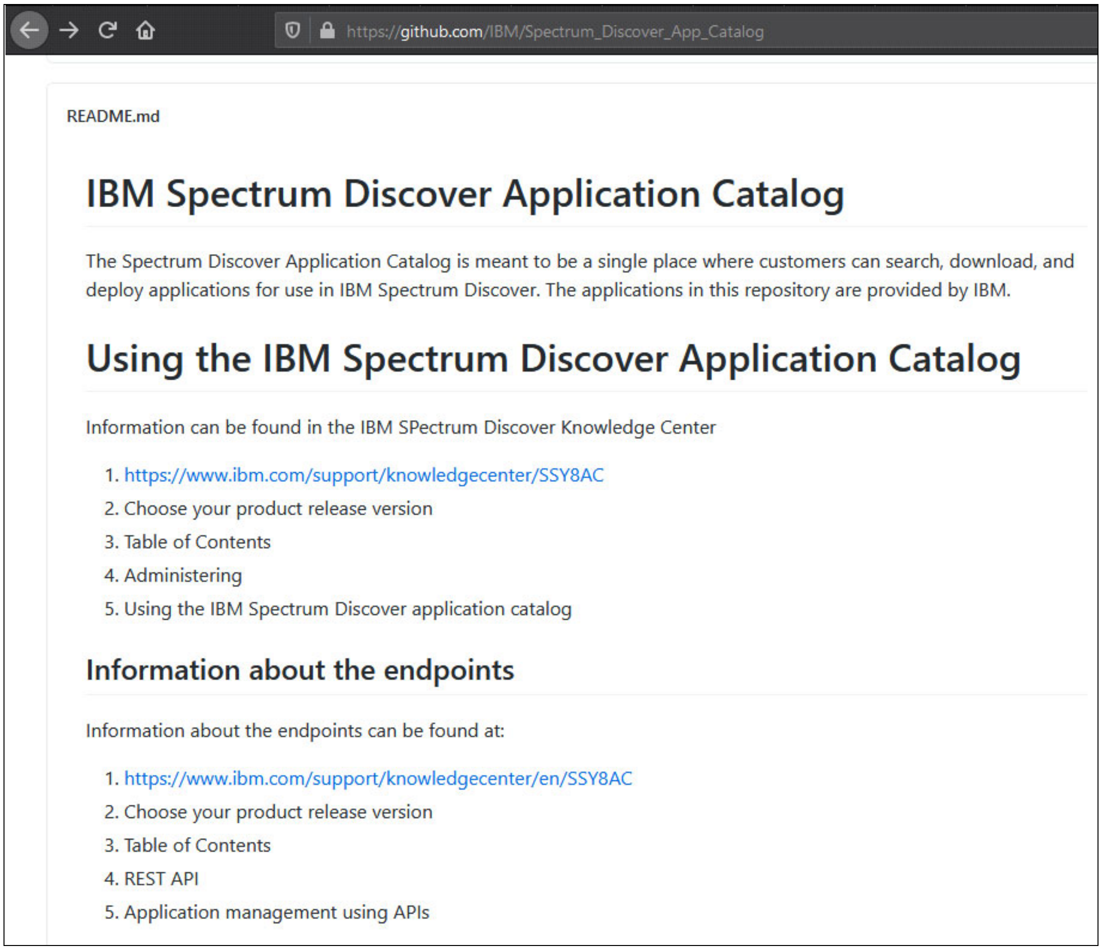
- Action Agent SDK extends capabilities via well defined API
- Customize actions taken based on Discover metadata
  - Content indexing
  - Data movement (tiering)
  - Classification
  - Sensitive data identification
  - ROT Detection/Disposal
  - Etc…
- Integrate with upstream information management applications

Figure 1-13 IBM Spectrum Discover as an extensible platform for adopting more application tools

The IBM Spectrum Discover SDK is available on GitHub, and a sample application is available on Dockerhub.

This SDK enables users to use IBM Spectrum Discover as a *platform* that can start custom applications and return the values that are identified from that application. This ability enhances the capability and customization options that are available on specific data sets that might be open formats or proprietary data formats.

To expand that capability, IBM established the IBM Spectrum Discover Application catalog as a single place where customers can search, download, and deploy applications for use in IBM Spectrum Discover, as shown in Figure 1-14. The applications in this repository are provided by IBM.



IBM Spectrum Discover Application Catalog

The Spectrum Discover Application Catalog is meant to be a single place where customers can search, download, and deploy applications for use in IBM Spectrum Discover. The applications in this repository are provided by IBM.

Using the IBM Spectrum Discover Application Catalog

Information can be found in the IBM SPectrum Discover Knowledge Center

1. https://www.ibm.com/support/knowledgecenter/SSY8AC
2. Choose your product release version
3. Table of Contents
4. Administering
5. Using the IBM Spectrum Discover application catalog

Information about the endpoints

Information about the endpoints can be found at:

1. https://www.ibm.com/support/knowledgecenter/en/SSY8AC
2. Choose your product release version
3. Table of Contents
4. REST API
5. Application management using APIs

Figure 1-14 GitHub app catalog

For more information about the IBM Spectrum Discover Application Catalog, see GitHub.

1.6.5 Data movement with IBM Spectrum Discover

With Version 2.0.3.1, IBM Spectrum Discover introduced the ability to start data movement across select data sources. Now, data administrators or data engineers can move or collect various data sets in various data sources and physical locations into one place based on various characteristics they determined based on the information they see from the tags and reporting tools in IBM Spectrum Discover.

It also means that those users can start a data archiving strategy from view-to-action based on characteristics, such data temperature. They identify older data and then start the action to move that data between data sources. These approaches are run by IBM Spectrum Discover as a policy action.

The following approaches are available to this data movement:

- ScaleILM: A built-in application to IBM Spectrum Discover that uses the Information Life Cycle Management (ILM) engine that is a part of IBM Spectrum Scale.
- External Agent based movement between data sources: This approach uses an external agent to facilitate that movement.

ScaleILM: IBM Spectrum Scale based

Included with IBM Spectrum Discover V2.0.3 is a built-in application that is named *ScaleILM*, which can be used in data management policies to set up tiers of select set of files or data set on a IBM Spectrum Scale data source connection. The ScaleILM application enables those selected files or data sets to move between internal IBM Spectrum Scale pools in a cluster; for example, between a flash-based pool and an NL-SAS based pool or other configuration, and uses the IBM Spectrum Scale ILM capability. In addition, the ScaleILM tool enables the movement to an external storage repository or archive pools that are managed by IBM Spectrum Archive, which creates an integrated approach to managing the data to the correct tier based on the data sets tags, attributes, and overall information.

External agent-based movement

Included with IBM Spectrum Discover Version 2.0.3.1 is the capability to move data between data repositories that are being monitored. By using the same type of reporting and understanding of the data that is inherent in IBM Spectrum Discover, data administrators can make decisions and move data based on business requirements. By using the deep inspect capability along with validated external applications to run the movement as a policy, such data movement tasks can be orchestrated. As of Version 2.0.3.1, the first external agent application that is validated for this use is Moonwalk.

1.7 Deployment patterns

IBM Spectrum Discover can be deployed by using a single node or multiple nodes, depending on the size of the catalog. For environments with more than 2 billion files to be cataloged, IBM recommends deploying multiple nodes. Consider the following points:

- Single nodes are for environments in which the file count of the catalog is not greater than 2 billion files.
- To provide greater performance for environments with more than 2 billion files and higher availability, deploy three nodes that use shared storage (Red Hat OpenShift deployment only).

The following platforms options are available for the deployment of the IBM Spectrum Discover nodes:

- VMware-based deployment

  IBM Spectrum Discover is deployed as an Open Virtual Appliance (OVA) image to be deployed on VMware ESXi 6.0 or later.

- KVM-based deployment

  IBM Spectrum Discover V2.0.3.1 now supports deployment as a Kernel-based Virtual Machine (KVM).

IBM announced a plan to support deployment of IBM Spectrum Discover on Red Hat OpenShift in an upcoming release.

1.8 Overview of the use cases in the book

Table 1-2 gives a general overview of the use cases that are covered in this book.

Note: JFR stands for Journées Francophones de Radiologie.

Table 1-2 Overview of the use cases

“Categorizing medical imaging data with content-search capability” on page 30	
Use case overview	Categorizing medical imaging data with content-search capability.
Products involved	IBM Spectrum Discover
Benefits	▸ Collection of metadata that is associated with imagery in Digital Imaging and Communications in Medicine (DICOM) format. ▸ Cataloging medical imagery by patient, facility, diagnosis, and so forth.
High-level implementation steps	1. Identify metadata of interest from DICOM header details. 2. Create regular expressions that are associated with metadata to be collected. 3. Create tags for each metadata entity to be collected. 4. Define CONTENTSEARCH policy to perform the metadata collection. 5. After running the policy, explore the DICOM metadata by using visual exploration and search capabilities.
“Extracting metadata from LIDAR imagery by using custom applications” on page 35	
Use case overview	Extracting metadata from LIDAR imagery with a custom application.
Products involved	▸ IBM Spectrum Discover ▸ Point Data Abstraction Library (PDAL)
Benefits	▸ Ability to collect metadata by way of third-party software for Light Detection and Ranging (LIDAR) imagery. ▸ Ability to identify set of images based on geographic coordinates by using LIDAR metadata.
High-level implementation steps	1. Install PDAL software from `http://pdal.io`. 2. Develop metadata collection Application by using the PDAL software. (Contact IBM for assistance with this LIDAR application, if necessary.) 3. Create tags for each metadata entity to be collected. 4. Define DEEPINSPECT policy to perform the metadata collection. 5. After running the policy, explore the LIDAR metadata by using visual exploration and search capabilities. Identify areas of interest by using the minimum and maximum coordinates in a search query.
“Organizing training data sets for artificial intelligence” on page 42	
Use case overview	Categorizing medical imaging data with content-search capability.
Products involved	IBM Spectrum Discover
Benefits	Easily collect project-related metadata for organizing applications, such as machine vision models.

High-level implementation steps	1. Identify directory locations of training set data. 2. Create tags for each metadata entity, such as project or category of image, to be collected. 3. Define Auto-tagging policy to perform the metadata collection by specification the level in the directory structure to use as the value for the project or category. 4. After running the policy, explore the project metadata by using visual exploration and search capabilities.
“Using artificial intelligence in medical imaging: JFR Challenge” on page 63	
Use case overview	A consortium of radiology services team to share resources (medical images) and use AI to help diagnosis of lung nodules. There are four scenarios: ► They index their local data sources in a single Spectrum Discover application to create one single data set of 3D CT-Scans of lungs in DICOM format. ► They catalog metadata, explore, and clean the data set. ► Data scientists train deep learning models by using data from all hospitals. ► Trained model can be deployed and queried by radiologists to get inference results. It can also automatically analyze images, even if the patient does not come to the hospital for lung nodules.
Products involved	► IBM Spectrum Discover ► IBM Spectrum Scale, IBM Cloud Object Storage or any other data source ► IBM Watson Machine Learning Community Edition
Benefits	► Unify multiple data sources in one single IBM Spectrum Discover application. ► Assist creation and cleansing of a data set and extraction of data insights to prepare a deep learning training. ► Help copying or moving data to the appropriate storage for training. ► Assist the deployment of deep learning models and track the inferences performed.
High-level implementation steps	1. Deploy IBM Spectrum Discover and index multiple data sources to it. 2. Use CONTENTSEARCH to extract metadata from DICOM files; enrich these metadata with a DEEPINSPECT application that connects to a database 3. Generate graphs of data, detect outliers, filter IBM Spectrum Discover data to cleanse them. 4. Perform copy of labeled data to the storage where AI training is performed. 5. Deploy an AI service (either inside an IBM Spectrum Discover application or as an independent service with an API front end). 6. Query this AI service with new images to get inference results (either in near real-time mode or on multiple images).
“Data Governance use case: Data staging for high-performance processing” on page 116	
Use case overview	Identify required files and stage them into high-performance storage tier for the following data processing phase.
Products involved	► IBM Spectrum Discover ► IBM Spectrum Scale
Benefits	► Ability to identify and locate the required data sets faster and easier. ► Ability to move required data sets to a premier tier to improve the overall data processing performance.
High-level implementation steps	1. Search for required file sets for data processing. 2. Identify them based on policy tags. 3. Run a policy movement of the data to the high-performance storage pool by using the ScaleILM DEEPINSPECT agent. 4. Confirm that the files are now in the target storage pool in IBM Spectrum Scale.
“Data Optimization use case: Data migration to tape for cost-efficient archiving” on page 124	

Use case overview	Identify aged or *cold* files and move them into IBM Spectrum Archive and onto tape.
Products involved	► IBM Spectrum Discover ► IBM Spectrum Scale ► IBM Spectrum Archive Enterprise Edition
Benefits	► Ability to match value of the data with the cost of the storage medium. ► Ability to move older inactive data sets to an ‘active archive’ tier to reduce cost per TB.
High-level implementation steps	1. Search for aged, cold or inactive file sets. 2. Identify them based on policy tags. 3. Run a policy movement of the data to the IBM Spectrum Archive pool by using the ScaleILM DEEPINSPECT agent. 4. Confirm that the files are now in a ‘migrated’ state inside the IBM Spectrum Archive pool.

2

Generic imagery use cases

One of the more common uses of IBM Spectrum Discover is the categorization and searching of health and medical information. Many types and formats of data are part of this domain, including data that is used for genetics research, health records, and imaging.

In this chapter, we discuss how to use IBM Spectrum Discover in generic imagery use cases. IBM Spectrum Discover CONTENTSEARCH policies are powerful tools for extracting metadata from document content, including DICOM file headers.

This chapter includes the following topics:

- 2.1, "Categorizing medical imaging data with content-search capability" on page 30
- 2.2, "Extracting metadata from LIDAR imagery by using custom applications" on page 35
- 2.3, "Organizing training data sets for artificial intelligence" on page 42

2.1 Categorizing medical imaging data with content-search capability

Digital Imaging and Communications in Medicine is the international standard for medical images and related information[1]. This standard, which often is referenced by its acronym *DICOM*, defines the formats for medical images that can be exchanged with the data and quality necessary for clinical use. For more information about the DICOM standards, see this web page.

IBM Spectrum Discover CONTENTSEARCH policies are powerful tools that are used for extracting metadata from document content, including DICOM file headers as demonstrated in this chapter.

2.1.1 Metadata provided in DICOM Files

As expected, DICOM files contain a significant amount of metadata. Examples of DICOM header information is shown in Example 2-1.

Example 2-1 Typical metadata included in DICOM header

```
(0010, 0010) Patient's Name PN: '734_201_164'
(0008, 0008) Image Type CS: ['ORIGINAL', 'PRIMARY']
(0008, 0012) Instance Creation Date DA: '20160459'
(0008, 0013) Instance Creation Time TM: '32075.922000'
(0008, 0070) Manufacturer LO: 'Siemens Healthcare'
(0008, 1090) Manufacturer's Model Name LO: 'Somatom Definition Flash'
…
```

2.1.2 Using CONTENTSEARCH policy to extract DICOM metadata

IBM Spectrum Discover's CONTENTSEARCH policy engine captures header information from various file types. As with other policy types, it assigns values to tags. To perform this assignment, CONTENTSEARCH policies requires regular expressions to identify what metadata is to be captured. DICOM header information is manifested as key/value pairs that are delimited by a colon. To capture the value of a specific *key*, the CONTENTSEARCH policy uses regular expressions to capture the *values*.

Regular expressions for CONTENTSEARCH

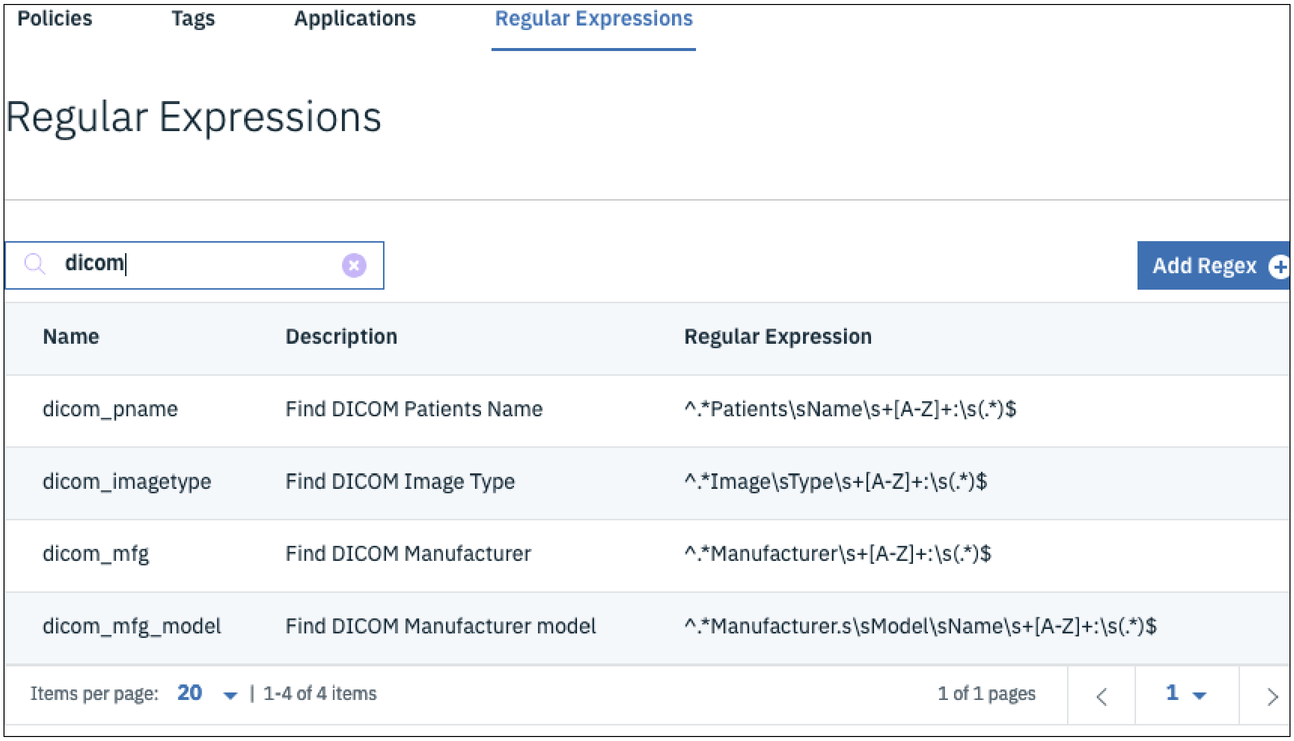
Consider capturing the value for the Manufacturer's Model Name given the format of the header metadata in a DICOM file. A regular expression must match the key (the Manufacturer's Model Name) and capture the "value"; that is, text after the colon. Such a regular expression (regex) is shown in the following example.

```
^.*Manufacturer.s\sModel\sName\s+[A-Z]+:\s(.*)$
```

When used to search the document's header, it starts at the beginning of the line (^) for the key text followed by combinations of spaces and text until it finds a colon. After the colon, the search continues to the first non-space character and then captures all text that is delimited by the parentheses until the end of the line.

[1] https://www.dicomstandard.org/

Regular expression syntax is beyond the scope of this document. For more information, see this web page. The regular expressions that are shown in Figure 2-1 are used in our example.



Regular Expressions

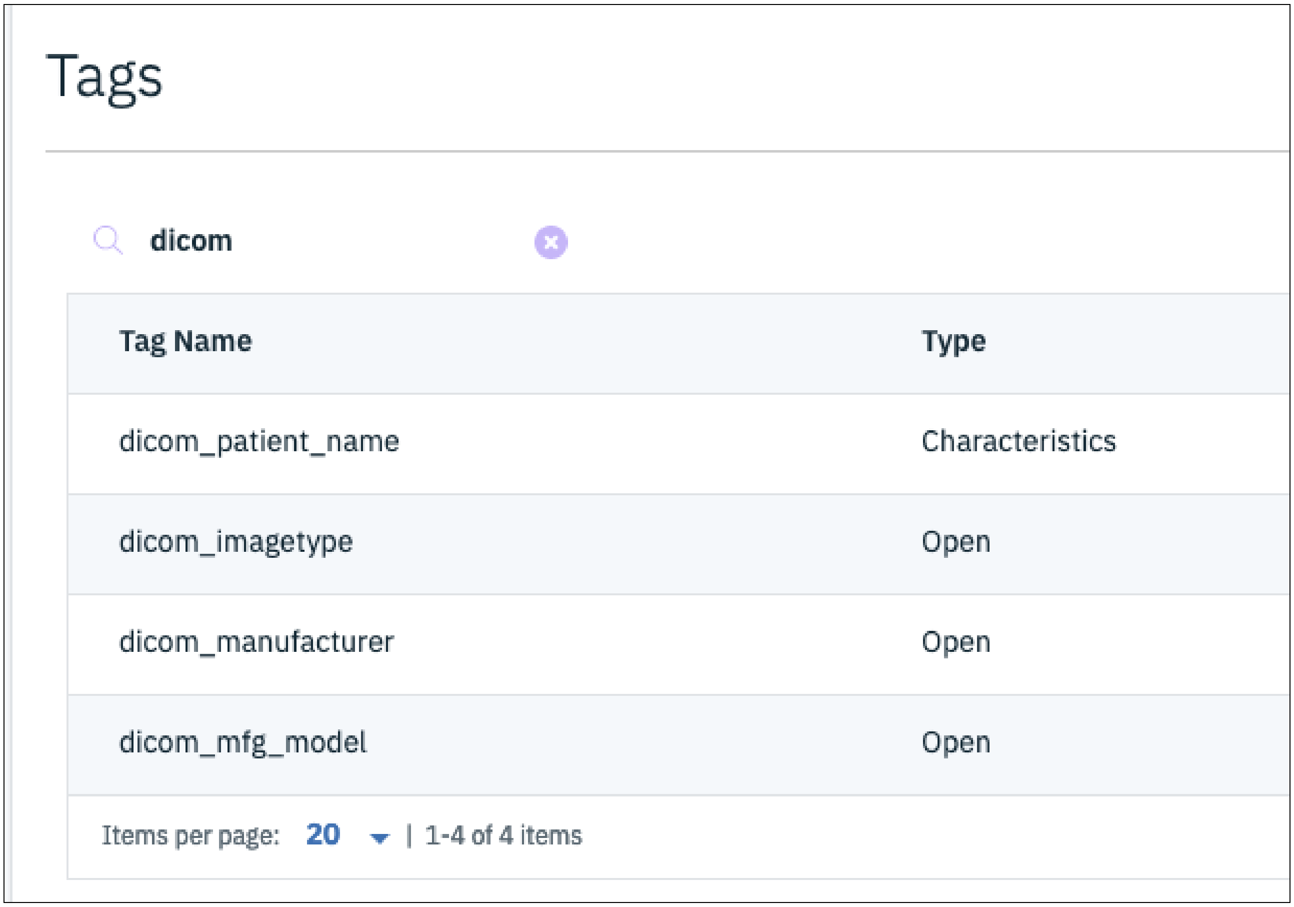
Name	Description	Regular Expression
dicom_pname	Find DICOM Patients Name	^.*Patients\sName\s+[A-Z]+:\s(.*)$
dicom_imagetype	Find DICOM Image Type	^.*Image\sType\s+[A-Z]+:\s(.*)$
dicom_mfg	Find DICOM Manufacturer	^.*Manufacturer\s+[A-Z]+:\s(.*)$
dicom_mfg_model	Find DICOM Manufacturer model	^.*Manufacturer.s\sModel\sName\s+[A-Z]+:\s(.*)$

Figure 2-1 Regular expressions used to capture values for various DICOM header metadata

Tag definition

Consider a scenario where we want to categorize DICOM images by the type of image, manufacturer, and model of the equipment that is used to collect the imagery. Also, assume that we want to analyze the results of the use of that equipment by following up with the patients that are involved. Other alternatives are available, such as collecting the physician's name and the location where the imagery was collected, but we limit our example to the tags we mentioned.

We create tags according to how we expect they will be used. Only the patient name is expected to be unique and therefore it is stored in a characteristics tag. The tags are named with a prefix of `dicom_` for clarity, as shown in Figure 2-2.



Tags

dicom

Tag Name	Type
dicom_patient_name	Characteristics
dicom_imagetype	Open
dicom_manufacturer	Open
dicom_mfg_model	Open

Items per page: 20 | 1-4 of 4 items

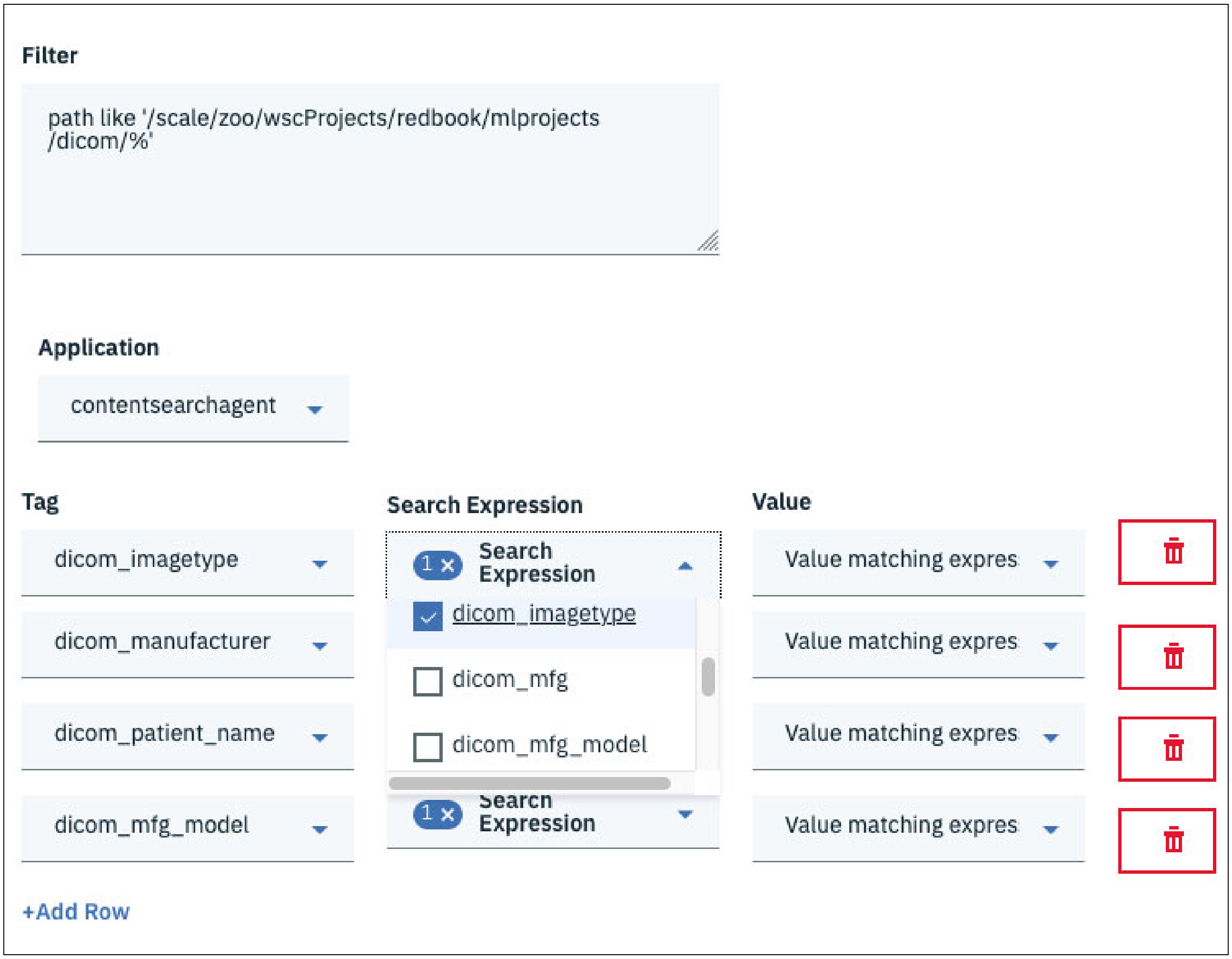
Figure 2-2 Tags to be used for the DICOM exploration use case

Creating the CONTENTSEARCH policy

When creating our policy, we select the CONTENTSEARCH policy type in the Define step. For efficiency, a CONTENTSEARCH policy can capture values for multiple tags and hence, the Policy step of creating the policy might be more detailed than other policy types.

The Collections and Filter attributes serve the same purpose as in other policy types. The Application attribute should be set to `contentsearchagent`. Then, each tag is listed along with selection of the regular expression to be used to capture the tag’s value.

Figure 2-3 shows our CONTENTSEARCH policy definition.



Filter

path like '/scale/zoo/wscProjects/redbook/mlprojects /dicom/%'

Application

contentsearchagent

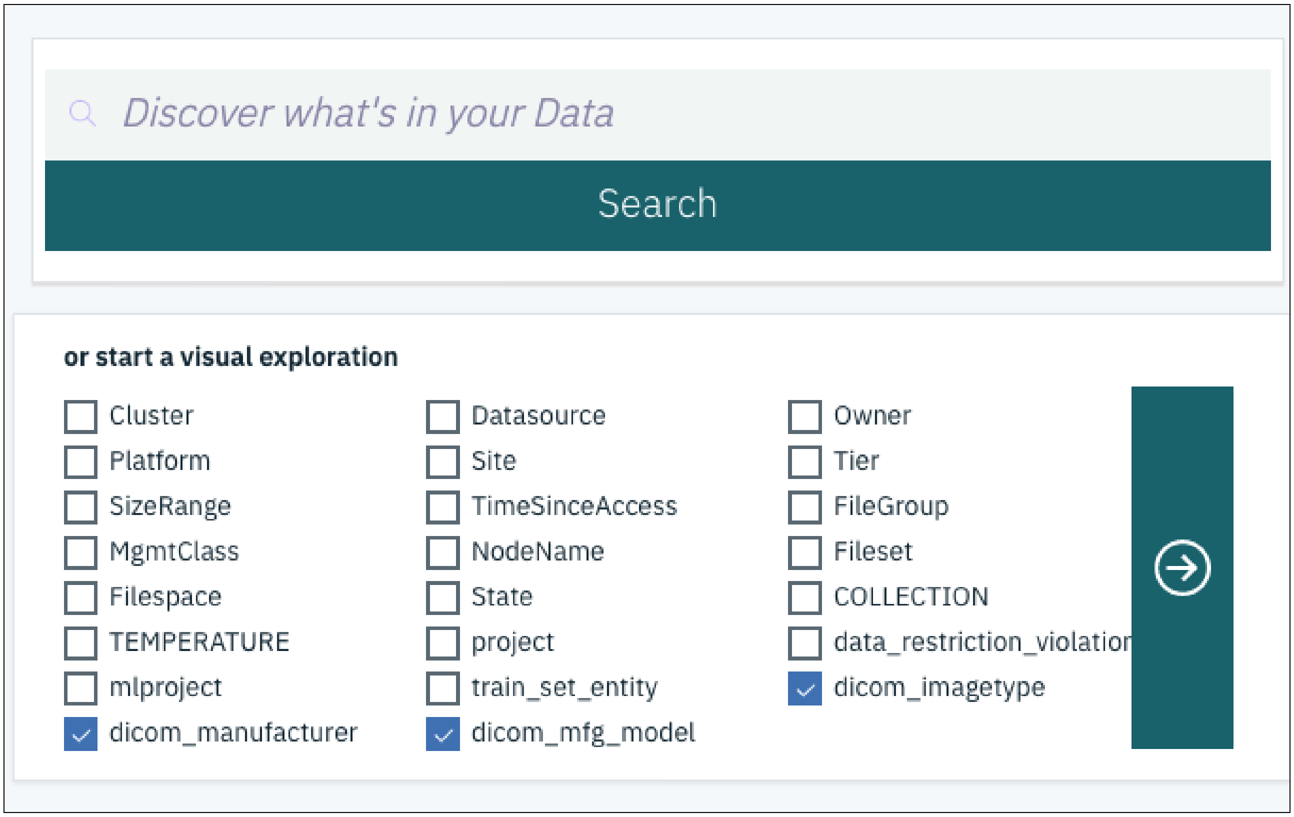
Tag	Search Expression	Value
dicom_imagetype	1 × Search Expression ☑ dicom_imagetype □ dicom_mfg □ dicom_mfg_model	Value matching expres
dicom_manufacturer		Value matching expres
dicom_patient_name		Value matching expres
dicom_mfg_model	1 × Search Expression	Value matching expres

+Add Row

Figure 2-3 Policy step for the DICOM CONTENTSEARCH policy definition

2.1.3 Exploring DICOM files by using IBM Spectrum Discover

After our policy is defined and it runs to completion, we can proceed with our visual exploration by cataloging our images based on the DICOM tags we have available. To do so, browse to the **Search** window and select the tags of interest. We then catalog by the various manufacturers, models, and image types, as shown in Figure 2-4 on page 34 - Figure 2-6 on page 35.

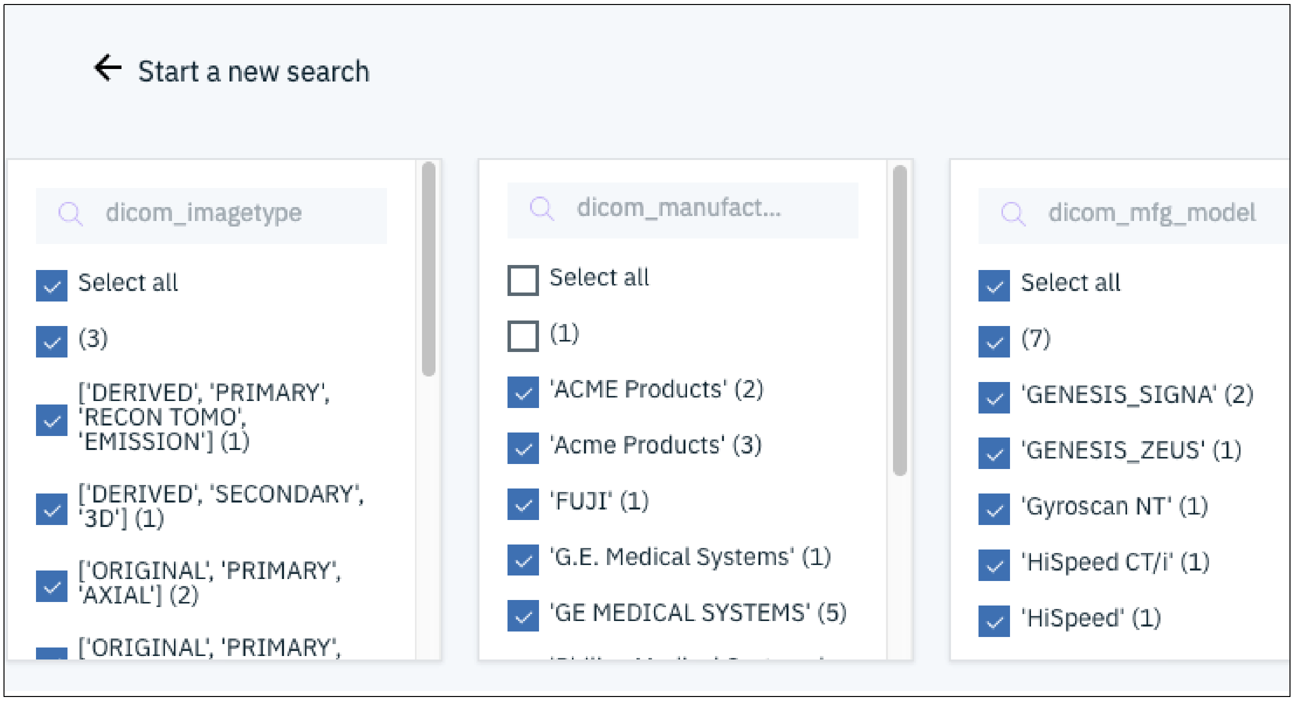

Discover what's in your Data

Search

or start a visual exploration

- □ Cluster
- □ Platform
- □ SizeRange
- □ MgmtClass
- □ Filespace
- □ TEMPERATURE
- □ mlproject
- ☑ dicom_manufacturer
- □ Datasource
- □ Site
- □ TimeSinceAccess
- □ NodeName
- □ State
- □ project
- □ train_set_entity
- ☑ dicom_mfg_model
- □ Owner
- □ Tier
- □ FileGroup
- □ Fileset
- □ COLLECTION
- □ data_restriction_violatior
- ☑ dicom_imagetype

Figure 2-4 Initiating the visual exploration of our DICOM metadata


← Start a new search

dicom_imagetype

- ☑ Select all
- ☑ (3)
- ☑ ['DERIVED', 'PRIMARY', 'RECON TOMO', 'EMISSION'] (1)
- ☑ ['DERIVED', 'SECONDARY', '3D'] (1)
- ☑ ['ORIGINAL', 'PRIMARY', 'AXIAL'] (2)
- ☑ ['ORIGINAL', 'PRIMARY',

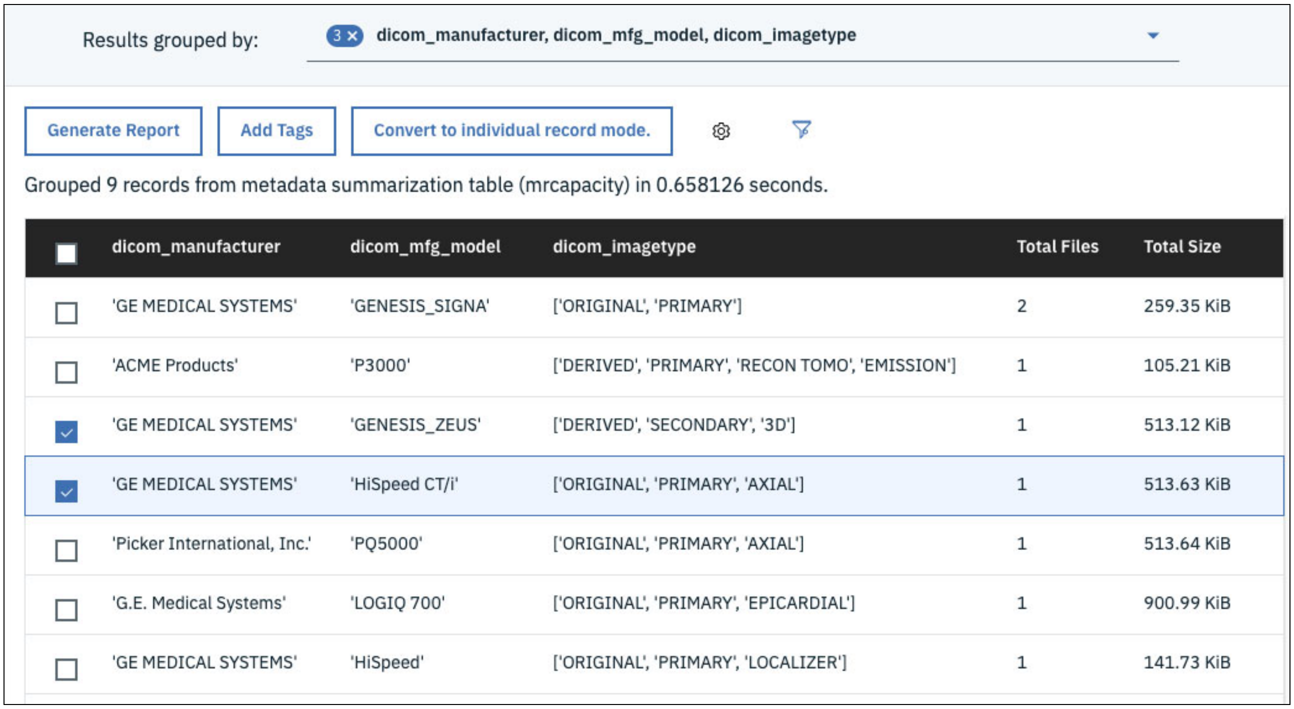
dicom_manufact...

- □ Select all
- □ (1)
- ☑ 'ACME Products' (2)
- ☑ 'Acme Products' (3)
- ☑ 'FUJI' (1)
- ☑ 'G.E. Medical Systems' (1)
- ☑ 'GE MEDICAL SYSTEMS' (5)

dicom_mfg_model

- ☑ Select all
- ☑ (7)
- ☑ 'GENESIS_SIGNA' (2)
- ☑ 'GENESIS_ZEUS' (1)
- ☑ 'Gyroscan NT' (1)
- ☑ 'HiSpeed CT/i' (1)
- ☑ 'HiSpeed' (1)

Figure 2-5 Selecting the values of interest for image type, manufacturer, and model



Results grouped by: 3 × dicom_manufacturer, dicom_mfg_model, dicom_imagetype

Grouped 9 records from metadata summarization table (mrcapacity) in 0.658126 seconds.

□	dicom_manufacturer	dicom_mfg_model	dicom_imagetype	Total Files	Total Size
□	'GE MEDICAL SYSTEMS'	'GENESIS_SIGNA'	['ORIGINAL', 'PRIMARY']	2	259.35 KiB
□	'ACME Products'	'P3000'	['DERIVED', 'PRIMARY', 'RECON TOMO', 'EMISSION']	1	105.21 KiB
☑	'GE MEDICAL SYSTEMS'	'GENESIS_ZEUS'	['DERIVED', 'SECONDARY', '3D']	1	513.12 KiB
☑	'GE MEDICAL SYSTEMS'	'HiSpeed CT/i'	['ORIGINAL', 'PRIMARY', 'AXIAL']	1	513.63 KiB
□	'Picker International, Inc.'	'PQ5000'	['ORIGINAL', 'PRIMARY', 'AXIAL']	1	513.64 KiB
□	'G.E. Medical Systems'	'LOGIQ 700'	['ORIGINAL', 'PRIMARY', 'EPICARDIAL']	1	900.99 KiB
□	'GE MEDICAL SYSTEMS'	'HiSpeed'	['ORIGINAL', 'PRIMARY', 'LOCALIZER']	1	141.73 KiB

Figure 2-6 Search results, categorized by the various manufacturers, models, and image types

We now have results that we can combine and analyze based on the various manufacturers, models, and image types. Researchers can now easily examine DICOM imagery that was collected by specific equipment and for a specific type of image. Attempting to organize this data without IBM Spectrum Discover can be tedious and require a combination of custom and third-party applications. However, as shown here, DICOM metadata can be collected fairly easily. After the policy and search expressions are defined, they can be reused with other DICOM data sets with little or no effort.

2.2 Extracting metadata from LIDAR imagery by using custom applications

Light Detection and Ranging (LIDAR) is an active remote sensing system that can be used to measure vegetation height across large geographical areas. Engineers in agriculture and forestry use LIDAR imagery to estimate forest canopy height because it is not practical to measure every plant or tree.

Typically, LIDAR systems emit light from a rapidly firing laser. This light travels toward the ground and is then reflected by objects on the ground, such as treetops and buildings. The time that is required for the light to make the trip to the object and back is recorded by the LIDAR system. That time is used to calculate the distance that was traveled and to determine the height or altitude of the object. LIDAR systems use Global Positioning System (GPS) to ultimately identify and record X, Y, and Z coordinates of the object's “top.”

Data from LIDAR systems can be used to derive details of a vegetation structure, such as canopy height and density, and species identification. Commonly used formats for LIDAR imagery are .las and the .laz format, which is a highly compressed version of the .las format.

A basic use case when developing an information architecture for LIDAR data is to identify where the image was captured, how (equipment and software), and when it was collected. To collect this information, IBM Spectrum Discover uses software that is collects metadata from LIDAR image files.

2.2.1 Using the Point Data Abstraction Library with LIDAR imagery

The *Point Data Abstraction Library (PDAL)* is an open source library and applications for processing point cloud data, including LIDAR imagery. PDAL software is available from this website and is released under terms of the Berkeley Software Distribution (BSD) open source license (copyright Hobu, Inc., `howard@hobu.co`). Among the many capabilities of PDAL, the command line demonstrates its ability to extract metadata from a specific LIDAR image, as shown in Example 2-2.

Example 2-2 Extracting metadata from a LIDAR image

```
# pdal info --metadata ./image_32075_1793.laz
{
  "filename": "./32075_1793.laz",
  "metadata":
  {
    "comp_spatialreference": "",
    "compressed": true,
    "count": 694173,
    "creation_doy": 20,
    "creation_year": 2013,
    "dataformat_id": 3,
    "dataoffset": 333,
    "filesource_id": 0,
    "global_encoding": 1,
    "global_encoding_base64": "AQA=",
    "header_size": 227,
    "major_version": 1,
    "maxx": 306429.1911,
    "maxy": 5091371.437,
    "maxz": 265.4639387,
    "minor_version": 2,
    "minx": 304030.8023,
    "miny": 5088054.325,
    "minz": 160.5603562,
    "offset_x": 0,
    "offset_y": 0,
    "offset_z": 0,
    "point_length": 34,
    "project_id": "00000000-0000-0000-0002-000000000173",
    "scale_x": 0.010000000000000000208,
    "scale_y": 0.010000000000000000208,
    "scale_z": 0.010000000000000000208,
    "software_id": "XPro SGM",
    "spatialreference": "",
    "system_id": "Leica ADS Imagery",
    "vlr_0":
  },
  "pdal_version": "1.9.1 (git-version: 0ca608)"
}
```

2.2.2 User-defined tagging for the metadata from PDAL

IBM Spectrum Discover's Software Development Toolkit (SDK) can be used to implement a deep inspect application to collect metadata by way of PDAL from LIDAR imagery. Tags are defined that correspond to the metadata of interest and a policy us deployed that uses the deep inspect application to collect values for those tags.

Tag definition

In the interest of simplicity, tags are defined by using the same metadata labels that are provided by PDAL as shown in Example 2-2 on page 36. Based on our goal of collecting location, source system information, and date of collection, we define tags as listed in Table 2-1.

Table 2-1 Tag list for LIDAR imagery with type and description

Tag	Tag Type	Description
minx	Characteristic	Lower bound X coordinate. Fairly unique (not suited for categorization).
maxx	Characteristic	Upper bound X coordinate. Fairly unique.
miny	Characteristic	Lower bound Y coordinate. Fairly unique.
maxy	Characteristic	Upper bound Y coordinate. Fairly unique.
minz	Characteristic	Lower bound Z (height) coordinate. Expected to be fairly unique.
maxz	Characteristic	Upper bound Z (height) coordinate.
creation_year	Open	Short list of values
creation_doy	Characteristic	Hundreds of values, probably not well-suited for categorization.
system_id	Open	LIDAR system identifier
software_id	Open	LIDAR system software identifier

2.2.3 Creating a policy to collect tag values from PDAL

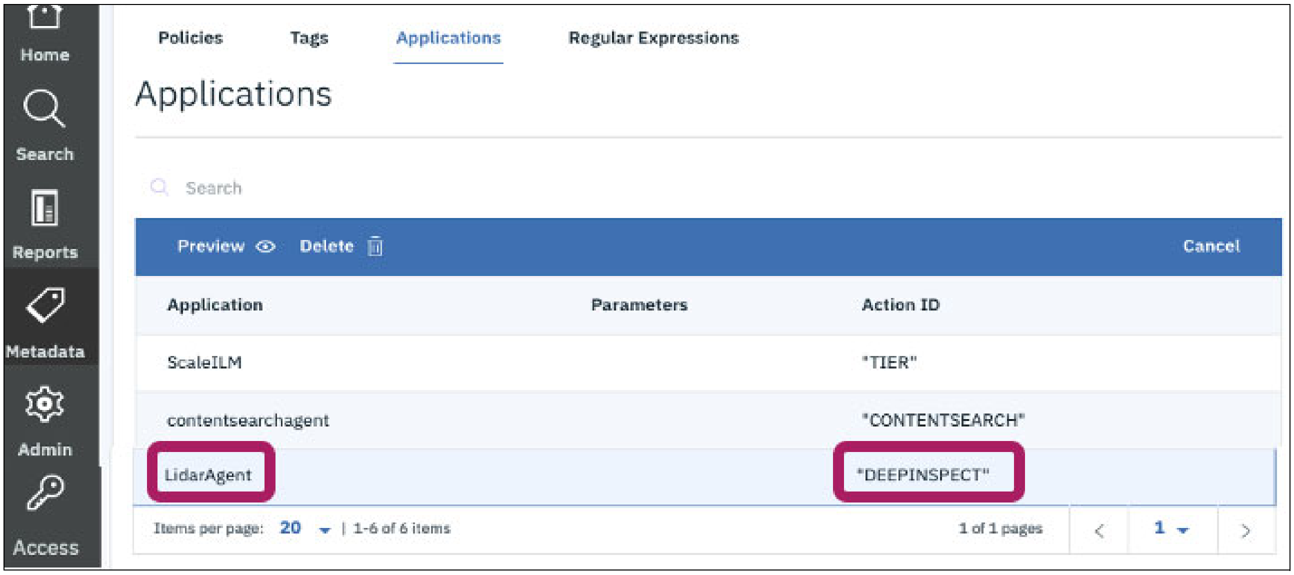
As is true for any policy in IBM Spectrum Discover, the tags it uses must exist; that is, the tags are defined before a DEEPINSPECT policy is created. The other prerequisite for DEEPINSPECT policies is that the application it uses must be running and be registered with the IBM Spectrum Discover node. The SDK provides an environment that authenticates and registers the application.

Running the deep inspect application

For our example, we name the application `LidarAgent`, as described in the IBM Spectrum Discover product documentation of deep inspect applications. Assuming that the application is running on a LINUX system, this process is done by exporting the environment variable APPLICATION_NAME before starting the application:

```
$ export APPLICATION_NAME=LidarAgent
```

After the application begins running, it registers itself with the IBM Spectrum Discover node, which lists it on the Metadata window under the Applications tab, as shown in Figure 2-7.



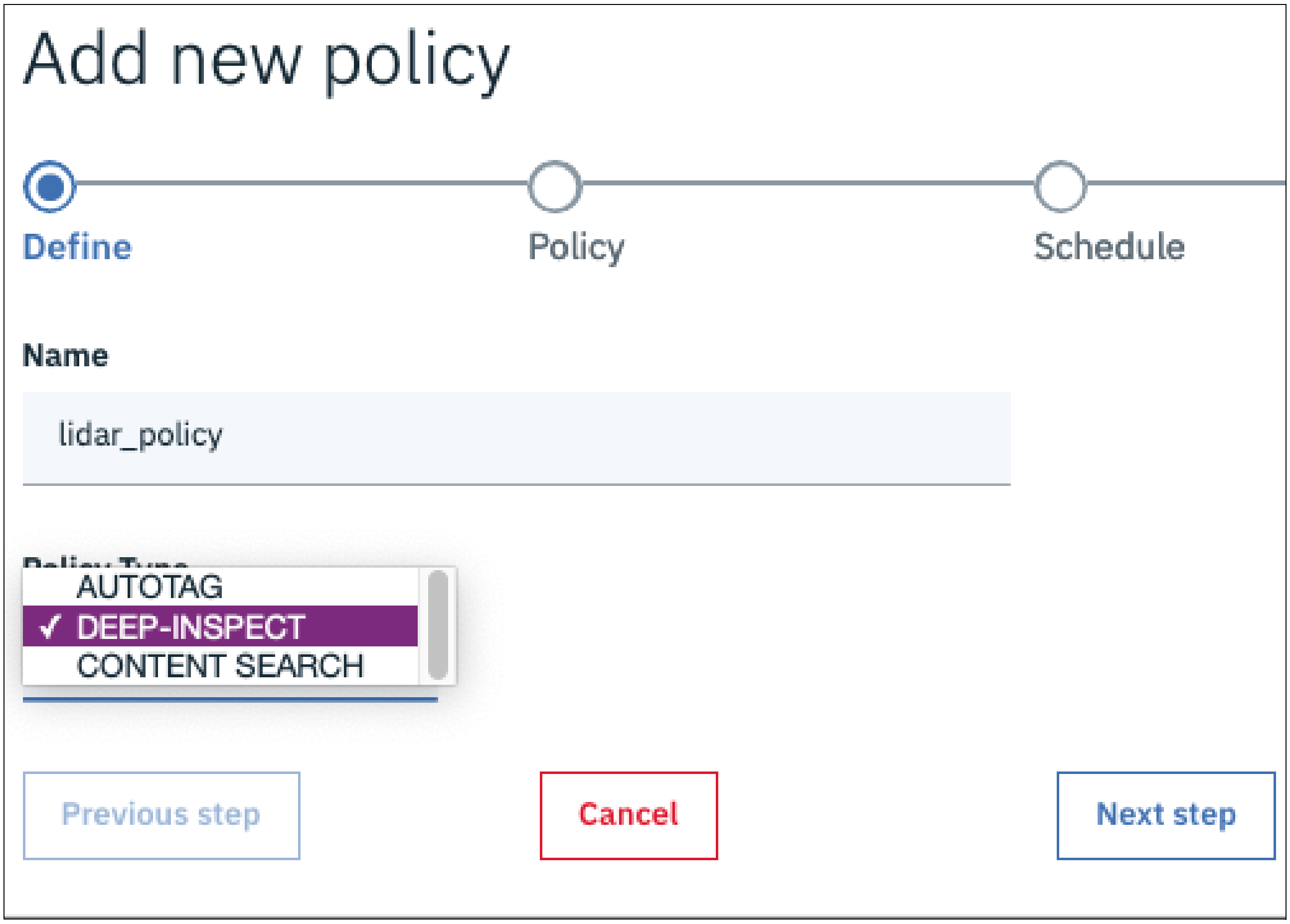
Application	Parameters	Action ID
ScaleILM		"TIER"
contentsearchagent		"CONTENTSEARCH"
LidarAgent		"DEEPINSPECT"

Figure 2-7 Application listing example, showing the deep inspect application LidarAgent

Creating the DEEPINSPECT policy

With the tags defined and the application registered and running, we can now create the policy. To do so, we browse to the **Metadata** window under the Policies tab.

The first step in creating the policy is the Define step where the policy is named and the Policy Type is selected. As shown in Figure 2-8, we select the **DEEP-INSPECT** type.



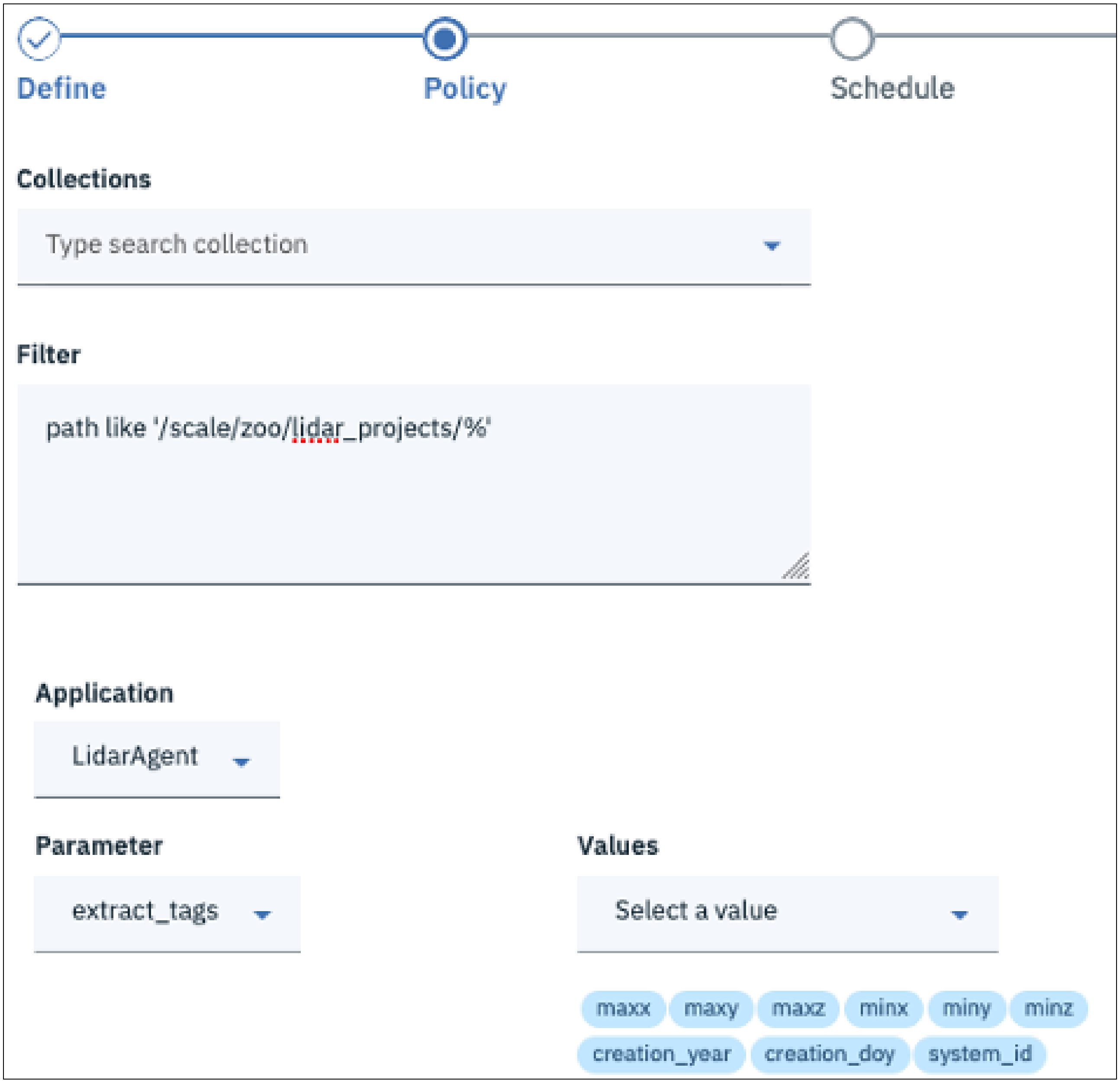
Add new policy

Name: lidar_policy

Policy Type: ✓ DEEP-INSPECT

Figure 2-8 Defining the DEEPINSPECT policy

Advancing to the policy step, we must select the Application to be used. In our example, we select the **LidarAgent** application. The `extract_tags` parameter must be provided with a list of tag names. As shown in Figure 2-9 on page 40, all of the tags from Table 2-1 on page 37 are entered in the list that appears under Values.



Collections

Type search collection

Filter

path like '/scale/zoo/lidar_projects/%'

Application

LidarAgent

Parameter

extract_tags

Values

Select a value

Figure 2-9 Selecting the suitable application in the Policy step

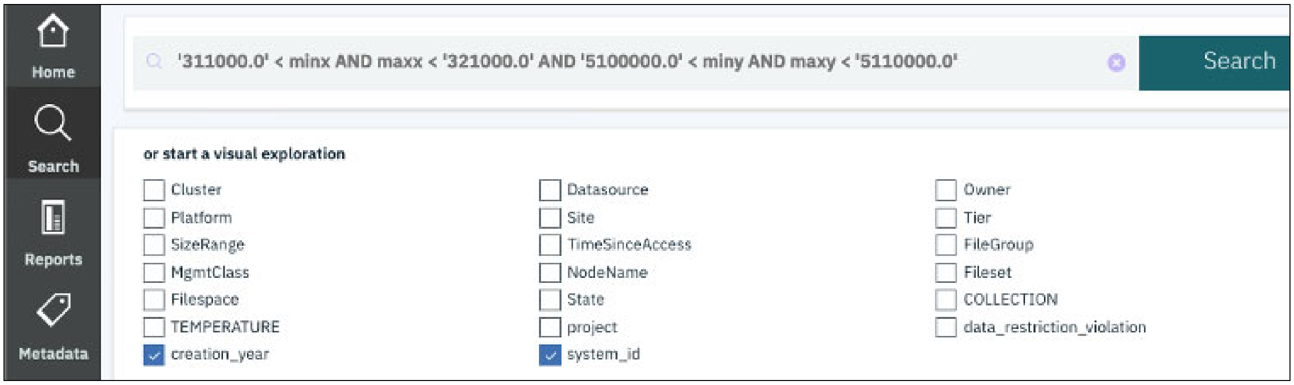
Complete the policy creation with the wanted schedule, and so forth. Running the policy triggers communication with the LidarAgent application and metadata collection starts.

2.2.4 Using the metadata to identify locations of interest

After the policy finishes, assume that the agriculture team is tasked with evaluating vegetation characteristics in a particular geographical area. The x and y coordinates of the area are bounded as follows:

```
311,000 < x < 321,000,   and  5,100,000 < y < 5,110,000
```

The team accesses the IBM Spectrum Discover GUI and browses to the **Search** page. They specify the search criteria and decide to categorize the images by `creation_year` and `system_id`, as shown in Figure 2-10 and Figure 2-11.

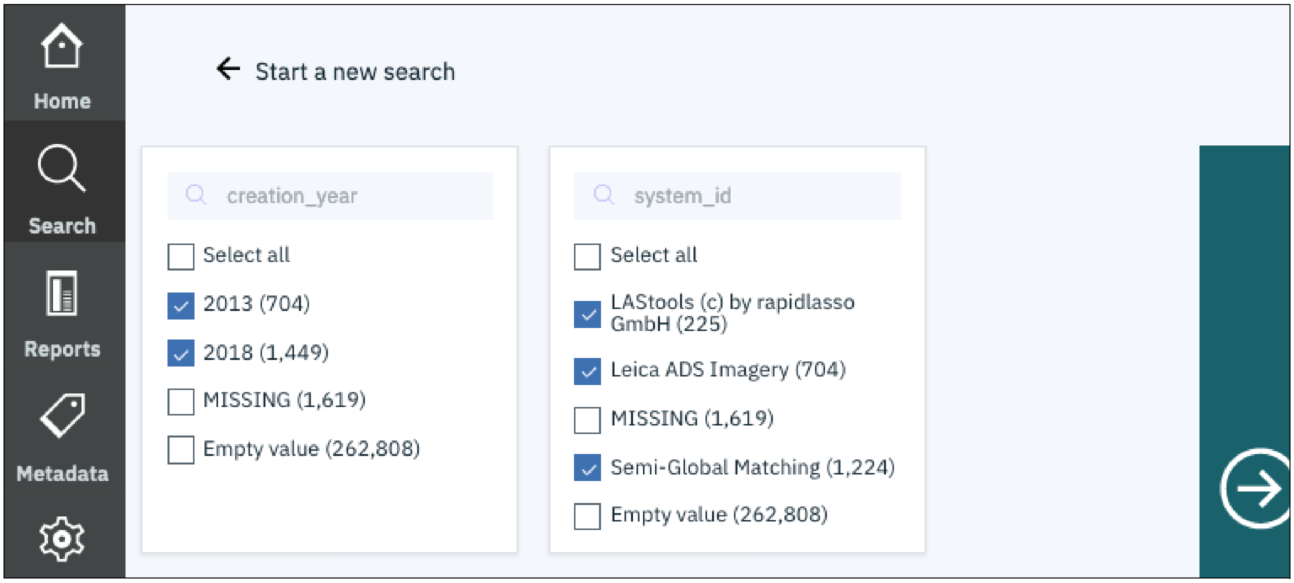


'311000.0' < minx AND maxx < '321000.0' AND '5100000.0' < miny AND maxy < '5110000.0'

or start a visual exploration

☐ Cluster
☐ Platform
☐ SizeRange
☐ MgmtClass
☐ Filespace
☐ TEMPERATURE
☑ creation_year
☐ Datasource
☐ Site
☐ TimeSinceAccess
☐ NodeName
☐ State
☐ project
☑ system_id
☐ Owner
☐ Tier
☐ FileGroup
☐ Fileset
☐ COLLECTION
☐ data_restriction_violation

Figure 2-10 Using the Search window to specify the rectangular area of interest



creation_year

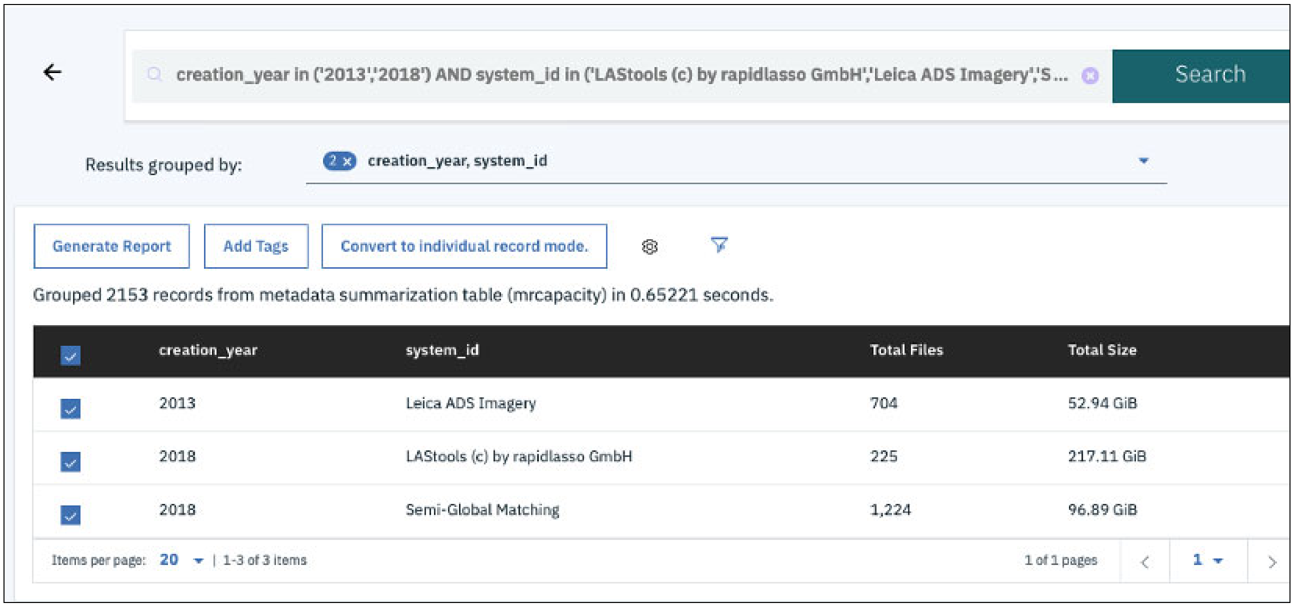
☐ Select all
☑ 2013 (704)
☑ 2018 (1,449)
☐ MISSING (1,619)
☐ Empty value (262,808)

system_id

☐ Select all
☑ LAStools (c) by rapidlasso GmbH (225)
☑ Leica ADS Imagery (704)
☐ MISSING (1,619)
☑ Semi-Global Matching (1,224)
☐ Empty value (262,808)

Figure 2-11 Categories for the images covering the area of interest for evaluation

Searching through its database, IBM Spectrum Discover quickly identifies and categorizes over 2,000 files that satisfy the team's parameters. As shown in Figure 2-12, the results present the team with many options.


creation_year in ('2013','2018') AND system_id in ('LAStools (c) by rapidlasso GmbH','Leica ADS Imagery','S...
Search
Results grouped by:
2 x
creation_year, system_id
Generate Report
Add Tags
Convert to individual record mode.
Items per page: 20
| 1-3 of 3 items
1 of 1 pages
1

Grouped 2153 records from metadata summarization table (mrcapacity) in 0.65221 seconds.

☑	creation_year	system_id	Total Files	Total Size
☑	2013	Leica ADS Imagery	704	52.94 GiB
☑	2018	LAStools (c) by rapidlasso GmbH	225	217.11 GiB
☑	2018	Semi-Global Matching	1,224	96.89 GiB

Figure 2-12 Second step of the Search sequence - categories for the images covering the area of interest for evaluation

The team can immediately generate a report in the various formats that are supported by IBM Spectrum Discover.

For future reference, they might create a tag that identifies this region with a value that can prove useful.

This data might need to be colocated for processing or visual examination. In this case, they can use the search criteria (or the new tag) to specify the filter for a data management policy that moves these files to the wanted location.

2.3 Organizing training data sets for artificial intelligence

The problem of "Where's my data?" continues to challenge data analytics applications regardless of the type of application, such as MapReduce, machine learning, or general high-performance computing. The sheer magnitude of new, unstructured data increases the problem's complexity by orders of magnitude. Data wrangling (identifying, collecting, and organizing data) uses more time now than ever before.

IBM Spectrum Discover is an important tool in creating an information architecture for artificial intelligence (IA for AI). Data identification, collection, and organization is simpler and easier to realize. Storage administrators and data scientists spend less time wrangling data, which leads to faster and higher-quality results.

An important part of developing machine learning systems involves training the model. Maintaining the organization of data sets that are used to train the model is critical to refining the system's accuracy and value. To effectively develop a machine learning model, the training data sets must be organized so that the images that are used to identify an entity can be clearly identified (or tagged) as being images of that entity.

The Electric Power Research Institute (EPRI) is investigating expanded use of Unmanned Aircraft Systems (UAS) to perform comprehensive power transmission inspections. EPRI research indicates that artificial intelligence can improve the current UAS inspection capabilities. As part of their research, EPRI is constructing sets of training images for use in the development of image recognition applications to be used with UAS to automate the inspection of power transmission structures and components.

These training sets include images that represent various aspects, or states, of such components. Examples include, but are not limited to:

- Glass insulator good
- Glass insulator broken
- Glass insulator contaminated
- Polymer insulator good
- Polymer insulator flashed
- Polymer insulator contaminated
- Porcelain insulators good
- Porcelain insulators flashed

For our use case, suppose a user downloaded a recent training set to be used in developing machine learning models for power transmission inspections. The data sets were placed in the following directory:

```
/scale/zoo/wscProjects/redbook/mlprojects/powerXmit/Training Set Final/
```

They are grouped by subdirectory name, as shown in the following example:

```
/scale/zoo/wscProjects/redbook/mlprojects/powerXmit/Training Set Final/Bird Guards
```

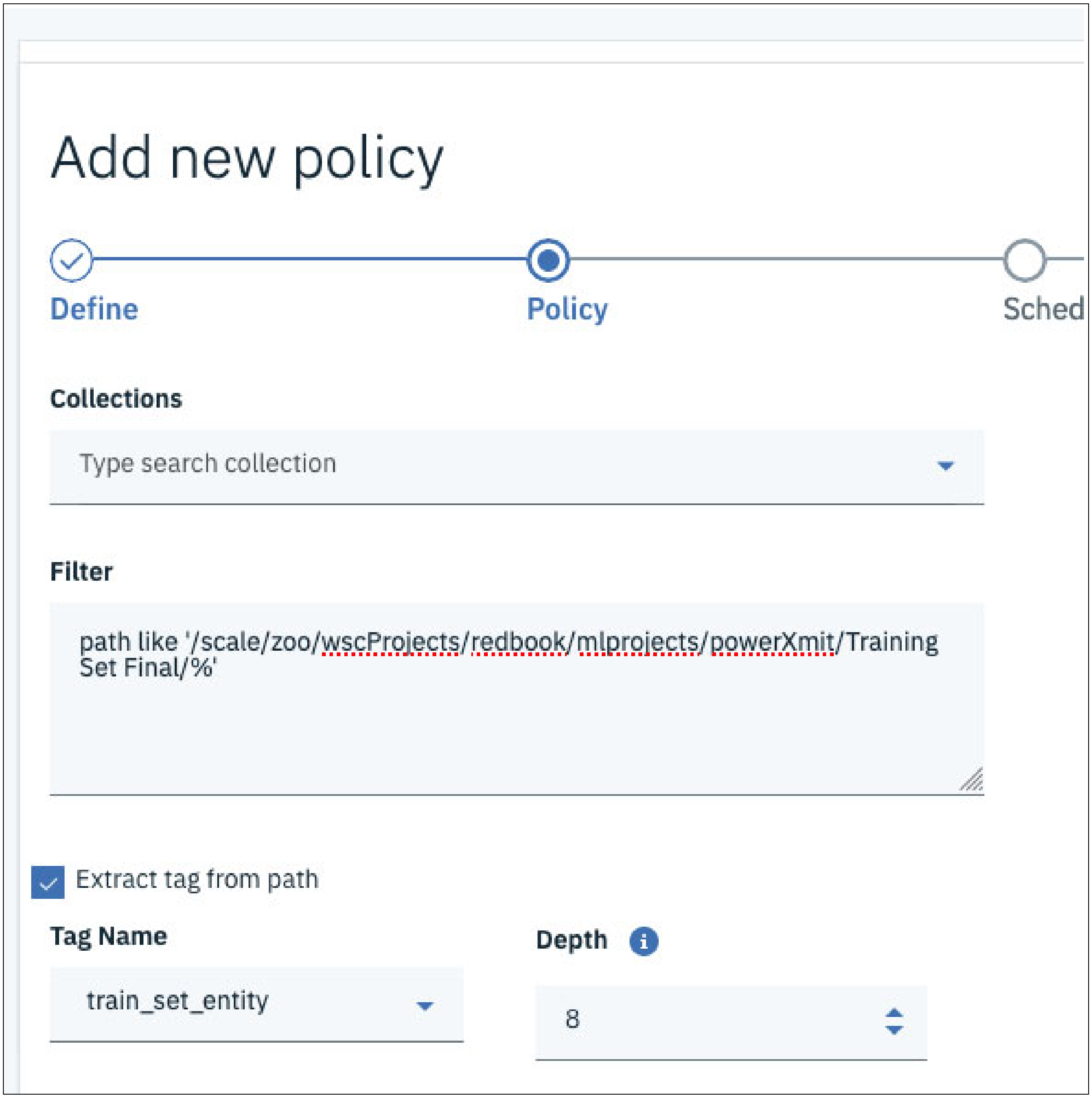
IBM Spectrum Discover is well-suited to easily identify and categorize such data sets as described next.

2.3.1 Extracting training set metadata

For this exercise, two user-defined tags are created on our IBM Spectrum Discover system, `mlproject` and `train_set_entity`, to identify which machine learning project these data sets are associated with and what entity the data sets describe, respectively.

With these tags defined, create an AUTOTAG policy `catalog_transmission_inspect` so that the filter focuses on the files that were uploaded in the `powerXmit/Training Set Final/` subdirectory. We set the value of `training_set_entity` to the directory name at the suitable depth.

This step of creating the policy is shown in Figure 2-13 on page 44.



Add new policy

Collections

Type search collection

Filter

path like '/scale/zoo/wscProjects/redbook/mlprojects/powerXmit/Training Set Final/%'

☑ Extract tag from path

Tag Name

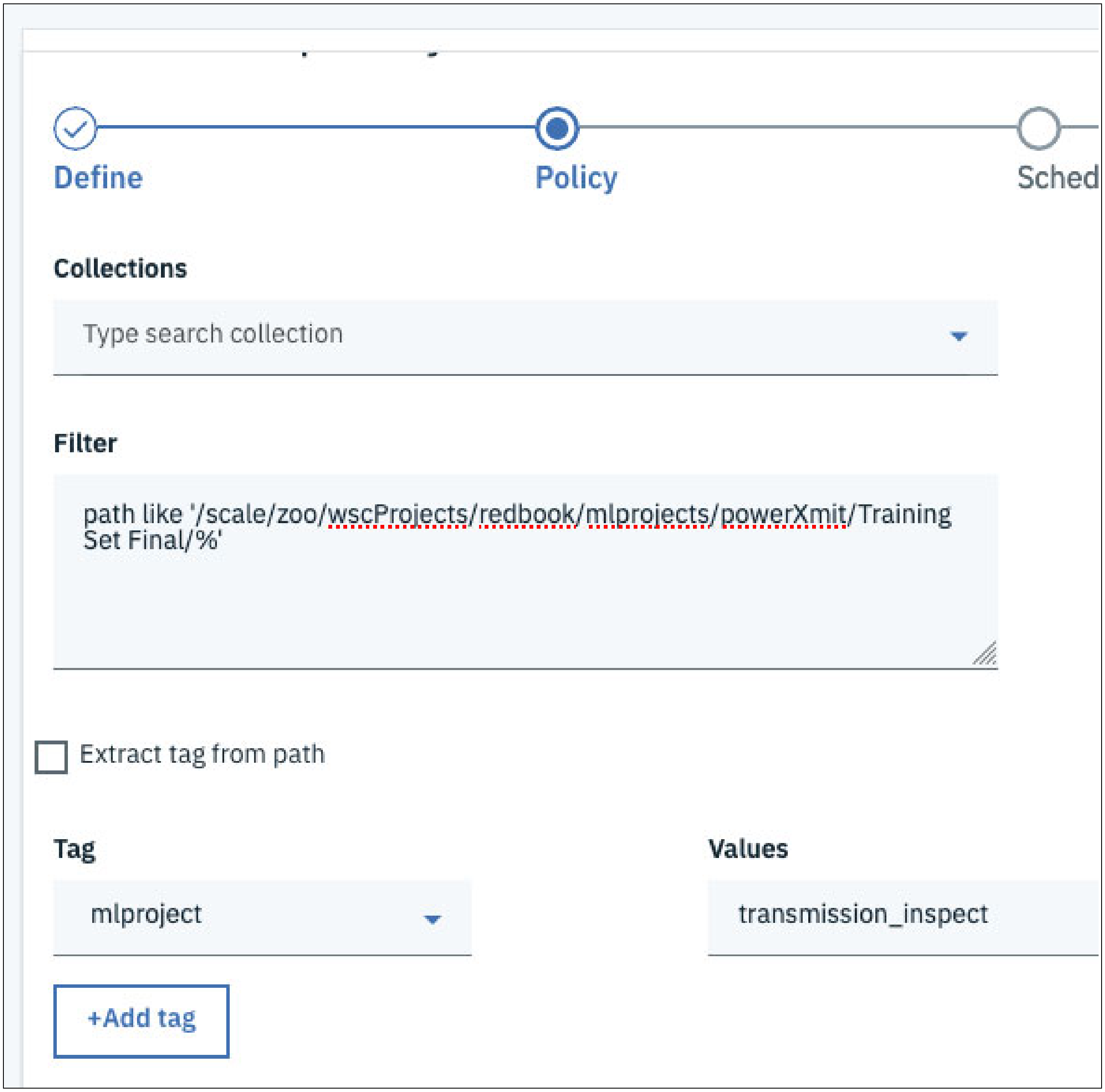
train_set_entity

Depth

8

Figure 2-13 The Policy step of creating the training_set_entity policy

We also can use an AUTOTAG policy to set the `mlproject` tag value to `transmission_inspect` in several ways. One simple example is to reuse the same path as used in the `catalog_transmission_inspect` policy with the resulting Policy step that is shown in Figure 2-14.



Collections

Filter

path like '/scale/zoo/wscProjects/redbook/mlprojects/powerXmit/Training Set Final/%'

□ Extract tag from path

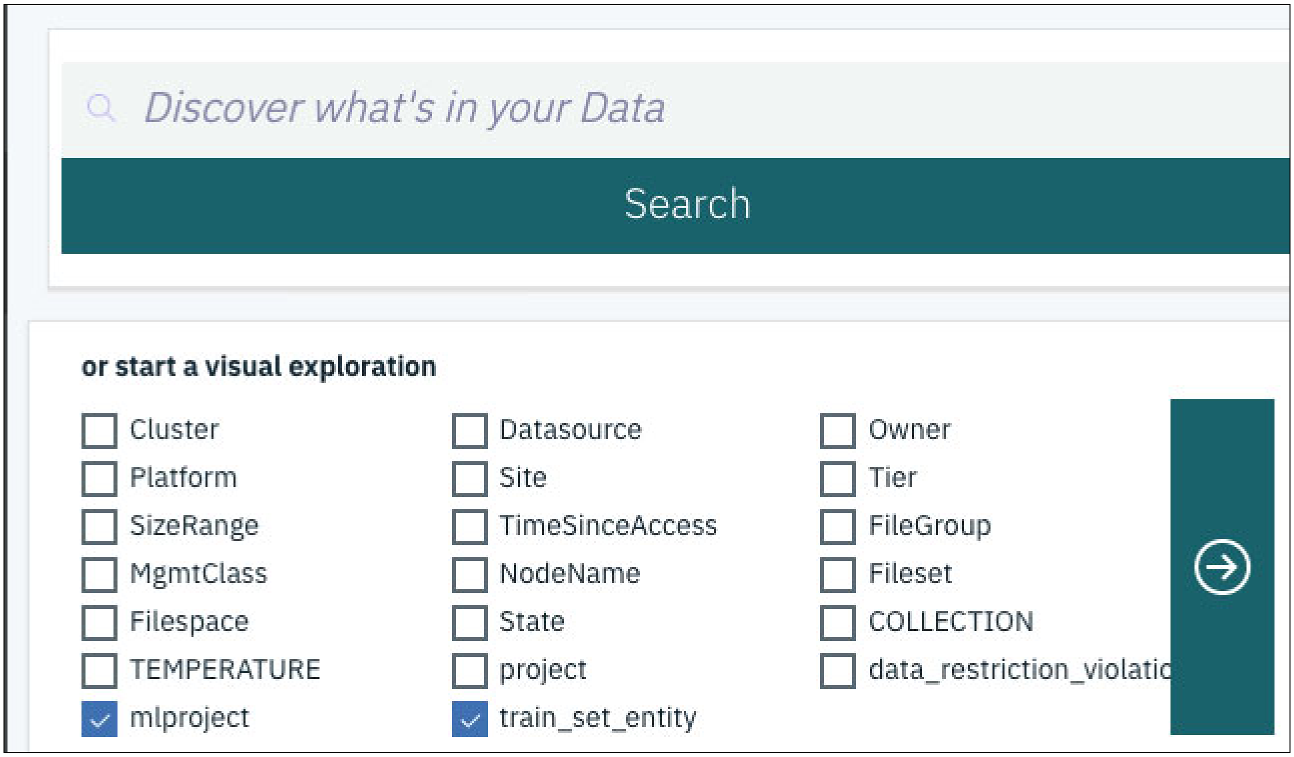
Tag	Values
mlproject	transmission_inspect

Figure 2-14 Creating the identify_mlproject policy to identify the mlproject name of the files as identified by the filter

2.3.2 Visual exploration of data in the transmission_inspect project

With the data ingested and cataloged in the IBM Spectrum Discover system, it is now simple to identify all the data sets that are associated with the transmission_inspect project.

At Search page, select the **mlproject** and **train_set_entity** tags for visual exploration and generate the resulting catalog, as shown in Figure 2-15 - Figure 2-17 on page 48.



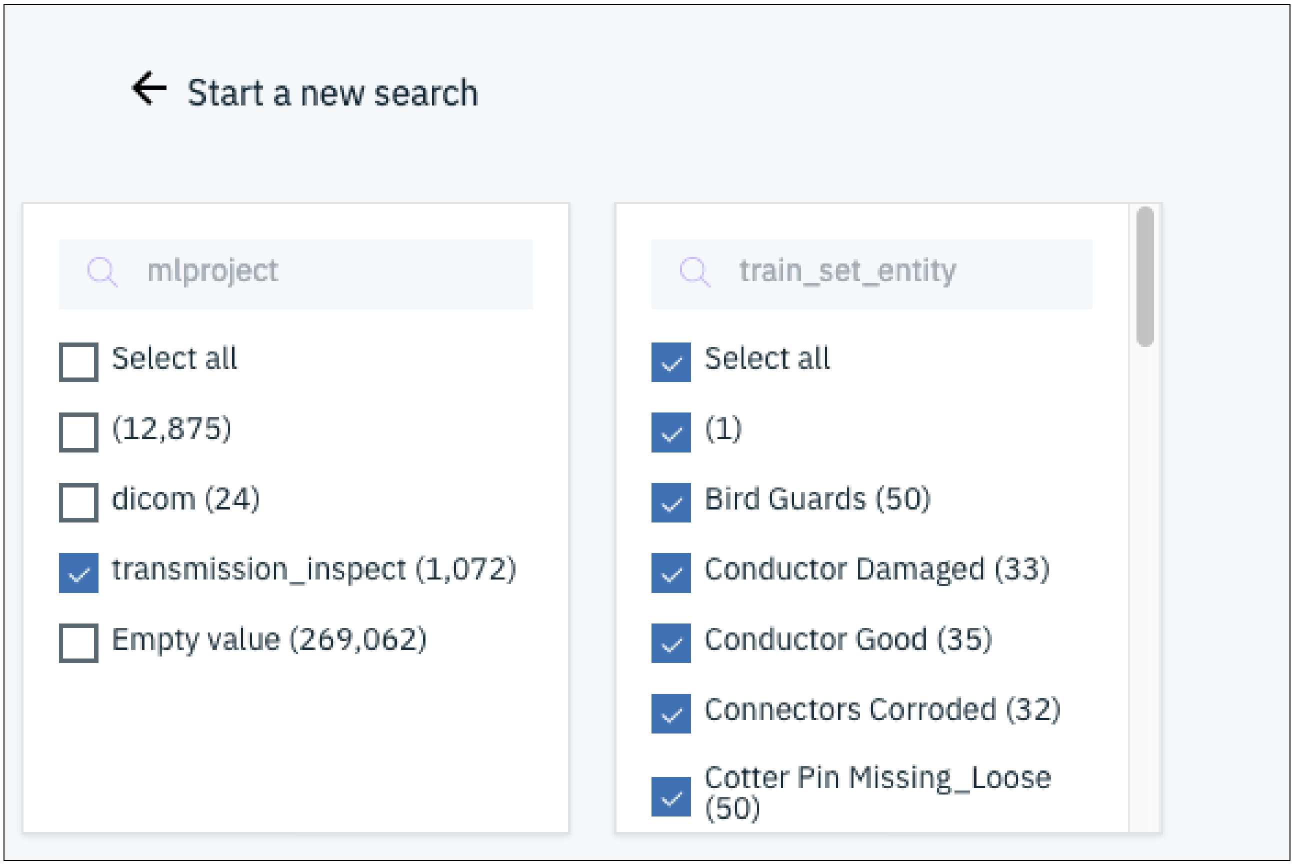
or start a visual exploration

□ Cluster
□ Platform
□ SizeRange
□ MgmtClass
□ Filespace
□ TEMPERATURE
☑ mlproject

□ Datasource
□ Site
□ TimeSinceAccess
□ NodeName
□ State
□ project
☑ train_set_entity

□ Owner
□ Tier
□ FileGroup
□ Fileset
□ COLLECTION
□ data_restriction_violatic

Figure 2-15 Selecting the mlproject and train_set_entity tags for visual exploration



← Start a new search

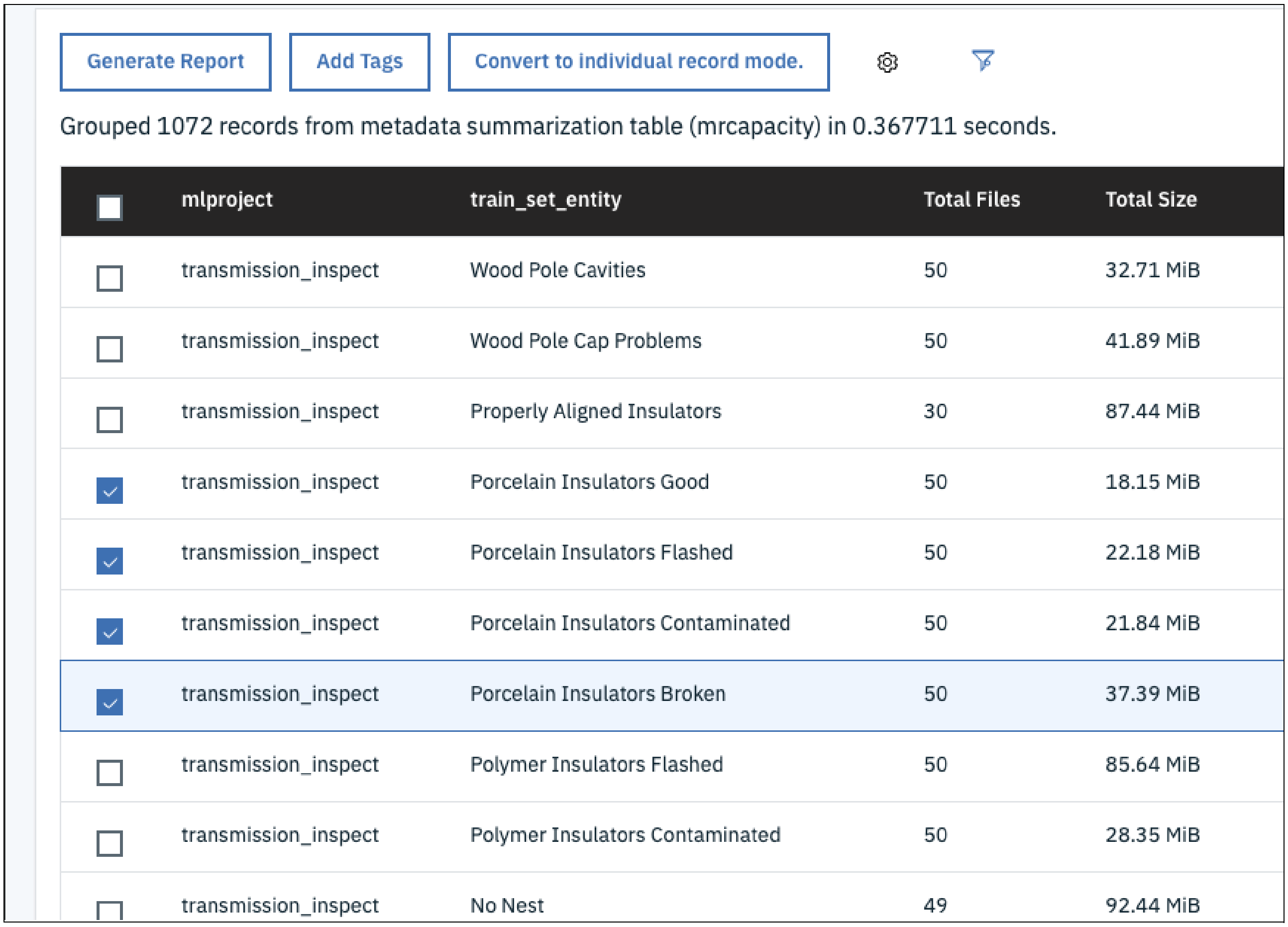
mlproject

□ Select all
□ (12,875)
□ dicom (24)
☑ transmission_inspect (1,072)
□ Empty value (269,062)

train_set_entity

☑ Select all
☑ (1)
☑ Bird Guards (50)
☑ Conductor Damaged (33)
☑ Conductor Good (35)
☑ Connectors Corroded (32)
☑ Cotter Pin Missing_Loose (50)

Figure 2-16 Selecting the mlproject of interest and all possible train_set_entity categories



Grouped 1072 records from metadata summarization table (mrcapacity) in 0.367711 seconds.

□	mlproject	train_set_entity	Total Files	Total Size
□	transmission_inspect	Wood Pole Cavities	50	32.71 MiB
□	transmission_inspect	Wood Pole Cap Problems	50	41.89 MiB
□	transmission_inspect	Properly Aligned Insulators	30	87.44 MiB
☑	transmission_inspect	Porcelain Insulators Good	50	18.15 MiB
☑	transmission_inspect	Porcelain Insulators Flashed	50	22.18 MiB
☑	transmission_inspect	Porcelain Insulators Contaminated	50	21.84 MiB
☑	transmission_inspect	Porcelain Insulators Broken	50	37.39 MiB
□	transmission_inspect	Polymer Insulators Flashed	50	85.64 MiB
□	transmission_inspect	Polymer Insulators Contaminated	50	28.35 MiB
□	transmission_inspect	No Nest	49	92.44 MiB

Figure 2-17 Results of visually exploring the transmission_inspect machine learning project

As shown in Figure 2-17, the data scientist can quickly identify training data sets that relate to some aspect of the model; for example, porcelain insulator inspection. They can generate a list of files by selecting the **Generate Report** option and use the file list as input for their machine learning model. By capturing the Search syntax, the storage administrator can move the selected data sets to a particular storage system or tier by using a data management policy.

Many next steps are possible, but the difficult part (data wrangling) is done. The next time the administrator or scientist must find the next subset of training data, IBM Spectrum Discover reduces the data wrangling effort to a simple visual exploration and a few checked boxes.

2.4 Summary

The use cases that are described in this chapter all focus on cataloging imagery while using different approaches. Each approach features a different policy, including basic auto-tagging, content search, and deep inspection. The latter policy type shows the power of extensibility and flexibility that is provided by IBM Spectrum Discover to collect metadata from almost any source.

DEEPINSPECT policies rely entirely on *applications*, as discussed in the use case for extracting metadata from LIDAR imagery. IBM provides an application catalog that includes multiple applications along with an example that can be used as a baseline to develop a custom application.

For more information, see *Using the IBM Spectrum Discover application catalog*: *Administration Guide*, SC27-9602-08. More applications might be available from IBM; therefore, contact your local IBM representative for more information.

Table 2-2 summarizes the using artificial intelligence in medical imaging use case.

Table 2-2 Summary of DICOM metadata extraction use case

Use case overview	Categorizing medical imaging data with content-search capability.
Products involved	IBM Spectrum Discover
Benefits	► Collection of metadata associated with imagery in Digital Imaging and Communications in Medicine (DICOM) format. ► Cataloging medical imagery by patient, facility, diagnosis, and so forth.
High-level implementation steps	1. Identify metadata of interest from DICOM header details. 2. Create regular expressions that are associated with metadata to be collected. 3. Create tags for each metadata entity to be collected. 4. Define CONTENTSEARCH policy to perform the metadata collection. 5. After running the policy, explore the DICOM metadata by using visual exploration and search capabilities.

Table 2-3 summarizes the LIDAR metadata collection use case.

Table 2-3 Summary of the LIDAR metadata collection use case

Use case overview	Extracting metadata from LIDAR imagery with a custom application.
Products involved	► IBM Spectrum Discover ► Point Data Abstraction Library
Benefits	► Ability to collect metadata by way of third-party software for LIDAR imagery. ► Ability to identify set of images based on geographic coordinates by using LIDAR metadata.
High-level implementation steps	1. Install PDAL software from `http://pdal.io`. 2. Develop metadata collection Application by using the PDAL software. (Contact IBM for assistance with this LIDAR application, if necessary.) 3. Create tags for each metadata entity to be collected. 4. Define DEEPINSPECT policy to perform the metadata collection. 5. After running the policy, explore the LIDAR metadata by using visual exploration and search capabilities. Identify areas of interest by using the minimum and maximum coordinates in a search query.

Table 2-4 on page 50 provides a summary of the organizing training data sets for AI use case.

Table 2-4 Summary of the use case for organizing training data sets for AI

Use case overview	Categorizing medical imaging data with content-search capability.
Products involved	IBM Spectrum Discover
Benefits	Easily collect project-related metadata for organizing applications such as machine vision models.
High-level implementation steps	1. Identify directory locations of training set data. 2. Create tags for each metadata entity, such as project or category of image, to be collected. 3. Define Auto-tagging policy to perform the metadata collection by specifying the level in the directory structure to use as the value for the project or category. 4. After running the policy, explore the project metadata by using visual exploration and search capabilities.

3

AI pipeline that uses IBM Spectrum Discover

In this chapter, we discuss the design of an artificial intelligence (AI) data pipeline that was created by using IBM Spectrum Discover. IBM Spectrum Discover's functions can be extended beyond the basic metadata scanning and tagging, by using deep inspection on agents and policies. These features allow access to data content, beyond metadata scan that is performed by IBM Spectrum Discover. Further, with help of orchestration and scheduling, this capability can be used to build powerful and automated data processing workflows.

This chapter includes the following topics:

3.1 Introduction to AI pipeline

An artificial intelligence pipeline (AI pipeline) depicts a typical processing model that is made with IBM Spectrum Discover. It denotes a typical data flow design with the help of application components that are outside IBM Spectrum Discover. IBM Spectrum Discover features various mechanisms to provide an interconnect to other applications. After it is completed, IBM Spectrum Discover can orchestrate key actions within the system to keep the data flowing within the pipeline.

As shown in Figure 3-1, an artificial pipeline denotes flow of data within an AI system. A typical AI pipeline can be represented in four stages: Ingest, curate, analyze, and infuse.

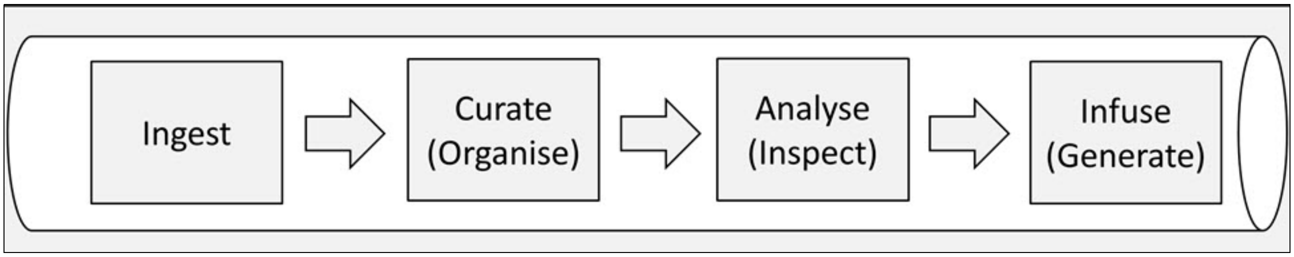


Figure 3-1 Simplified AI data pipeline

3.1.1 Ingest

Data ingestion is the first step in the AI pipeline. AI-enabled applications all share a common objective at their core: to ingest data from many sources and derive actionable insights or intelligence from it.

Data ingestion entails the complete process from acquiring the raw data to have it prepared for model training. The first step of this process involves parsing the raw data and representing it in a standardized format to be used in the next steps according to the problem definition. However, standardizing it is a tedious task because of the various data sources, lack of consistency, and the massive amount of data.

Data ingestion approaches emphasize the use of massive amounts of data in place, where analysis tools are used against data sources that were not moved or transformed. These approaches allow for much more flexibility because the transformation of data is isolated to each pipeline that is accessing a large data source.

Large data stores are the norm in large enterprises. Different pipelines can be used to flexibly transform data from large data stores. The concept of a data lake reflects this reality. *Data lakes* are large collections of data that are stored in their natural formats (often as object blobs or files). Today’s data scientist must be proficient in building data pipelines that tap directly into such large collections of raw data, then process the data to gain insights.

After selecting data from the sources of origin, data ingestion procedures resolve problems in the data and prepare it for research and modeling. With large data sets and complex use cases, data ingestion involves the ability to use data from a wide variety of sources, mixing and matching those sources to create data pipelines that feed machine learning models.

Cleaning, parsing, assembling, and gut checking data is among the most time-consuming tasks that data scientists must perform. There is no doubt that significant amounts of time are often devoted to data ingestion pipeline.

Data engineers work to facilitate the burden of data ingestion process because they are dealing with various types and sources, such as flat files, relational databases, and open API feeds (such as Twitter to XML web services). Connecting all of these heterogeneous sources in a standardized way can be a non-trivial task. More sources of data coming in increases the complexity of the ingestion process.

3.1.2 Curate

Curation activities enable data discovery and retrieval, maintain quality, add value, and provide for reuse over time. Data curation emerges as a key data management process in which an increase occurs in the number of data sources and platforms for data generation, which makes it more useful for users engaging in data discovery and analysis.

In practice, data curation is more concerned with maintaining and managing the metadata rather than the database itself and to that end, a large part of the process of data curation revolves around ingesting metadata, such as schema, table and column popularity, usage popularity, top joins, filters, and queries.

Data curators collect data from diverse sources, integrating it into repositories that are many times more valuable than the independent parts. Data curators not only create, manage, and maintain data, but might also be involved in determining best practices for working with that data.

Data curators often present the data in a visual format, such as a chart, dashboard, or report. Data curation includes data authentication, archiving, management, preservation, retrieval, and representation. Risks of poor or no data curation include factually inaccurate information, incorrect guidelines, and knowledge gaps.

Data curation starts with the *data set*. These data sets are the atoms of data curation. Determining which of these data sets are the most useful or relevant is the job of the data curator. The ability to present the data effectively is also important. Although some best practices apply, the data curator must make an educated decision about which data assets are suitable to use.

It is important to know the context of the data before it can be trusted. Data curation uses such arbiters of modern taste as lists, popularity rankings, annotations, relevance feeds, comments, articles, and the upvoting or downvoting of data assets to determine their relevancy.

Data curation was much more manageable when enterprises had only a few data sources from which to extract data. With the proliferation of big data, enterprises today have many more disparate data sources to extract data from, which makes it much more difficult to maintain a consistent method to curate data. Further complicating the problem is the fact that much of today’s data is created in an ad hoc way that cannot be anticipated by the people who intended to use data for analysis.

In a modern data catalog, all of this metadata is collected along with information about the assets themselves and organized within a catalog interface that is more readily searched and browsed than the legacy systems.

Data curation can also be described as the process of adding value to data. A data-driven organization naturally wants to maximize the value of that data. Therefore, establishing people, processes, and tools for data curation should be a part of any technical manager’s plans. This process might mean establishing strict rules about which data can and should be used, and putting business rules or other metrics in place that apply to all data sets no matter where they physically are stored.

Data curation is a necessary endeavor for any organization that is attempting to enable self-service analytics because it provides data consumers with faster access to the data that they need to make intelligent business decisions that affect the enterprise.

3.1.3 Inspect

AI-enabled applications lie a data pipeline that moves starting from data ingestion from multiple sources, through several stages of data classification, transformation, analytics, machine learning, and deep learning model training where applications can derive actionable insights or intelligence from the ingested data.

Tremendous value and intelligence is extracted from large, captured data sets (big data) that leads to actionable insights through today's analytics. The data inspection stage is uncovering trends, patterns and associations, new connections, and precise predictions that are helping businesses achieve better outcomes. Therefore, it is not only about the matter of storing data any longer, but capturing, preserving, accessing, and transforming it to take advantage of its possibilities and the value it can deliver.

Interactive model analysis is a combination of understanding the problem and available data, unified with understanding, diagnosing, and refining a machine learning model task. This analysis enables a suitable machine learning model selection process to achieve the ideal solution result. Machine learning's goal is helping businesses to manage, analyze, and use their data far more effectively than ever before for faster and better predictive decisions.

Data inspection includes data exploration and preparation to ensure data diversity that is unbiased and abundant for better prediction. Data augmentation is a step that is carried out to improve the diversification of data that was sourced.

Data Inspection also includes feature engineering. This step involves the art and science of transforming raw data into features that better represent a pattern to the machine learning algorithms. For example, data can be decomposed into multiple parts to capture more specific relationships.

Many of today's machine learning models are trained to run a specific task or provide insights that are derived from "What happened?" to "What will likely happen?" (predictive analysis). These models are iteratively applied to the previous result and improved upon each time to get closer approximations to solving the problem.

Training is the core of machine learning as we train a model to take inputs and predict an output with the lowest error possible. With larger models, and especially with large training sets, this step can quickly become difficult to manage. Because memory is generally a finite resource for our computations, the efficient distribution of the model training is crucial.

To proceed in machine learning training, data must be split into two sets: one for a training machine learning algorithm, and another for evaluation purposes. Whenever new training data becomes available, same workflow that includes data validation, preprocessing, model training, analysis, and deployment should be triggered.

Also, metadata extraction and the discovered correlations between metadata insights are foundation of machine learning models. The analysis phase is used to go beyond metadata scan and gain insights into data contents. After a model is sufficiently trained, it can be put into production to deliver faster determinations.

Data inspection stages that involve aggregating, normalizing, classifying data, and enriching it with useful metadata require high throughput, with small and large I/O. Model training requires a performance tier that can support the highly parallel processes that are involved in machine learning training and deep learning models with high throughput and low latency.

Retraining of models with inference does not require as much throughput, but still demands low latency. Also, archiving demands a highly scalable capacity tier for cold and active archive data that is throughput-oriented, and supports large I/O, streaming, and sequential writes. Any of this data inspection can occur on-premises or in private or public clouds, depending on requirements.

These varying requirements for building better machine learning models, including scalability, performance, deployment flexibility, and interoperability are a tall order. But, data science productivity depends on the efficacy of the overall data pipeline and not just the performance of the infrastructure that hosts the machine learning/deep learning workloads. It requires a portfolio of software and system technologies that can satisfy these requirements, along the entire data pipeline.

3.1.4 Generate

At this stage, you should have a trained model and are ready to conduct evaluation techniques on its performance.

For evaluation, we use a partition of the refined data, often referred to as the *test data*. The test data was not seen during the model training. They are also representative of examples of data that is expected to be encountered in practical scenarios.

The evaluation objective is to estimate the performance of the machine learning model, fit a model to the training data, and predict the labels of the test set. Further, the number of wrong predictions on the test dataset are counted to compute the model’s prediction accuracy. The prediction measures the performance of the model by determining the outcomes on the test data set that is not used for any training or cross-validation activities.

Therefore, a machine learning pipeline starts with the ingestion of new training data and ends with receiving some kind of feedback about how your newly trained model is performing. This feedback can be a production performance metric or feedback from users of your AI-enabled application. Per the feedback, retraining through inference is occurring to yield increasingly accurate decisions or insights.

3.2 AI pipeline by using IBM Spectrum Discover

A typical reference architecture to set up an AI pipeline is shown in Figure 3-2. This generic architecture can be applied to most AI and metadata processing use cases.

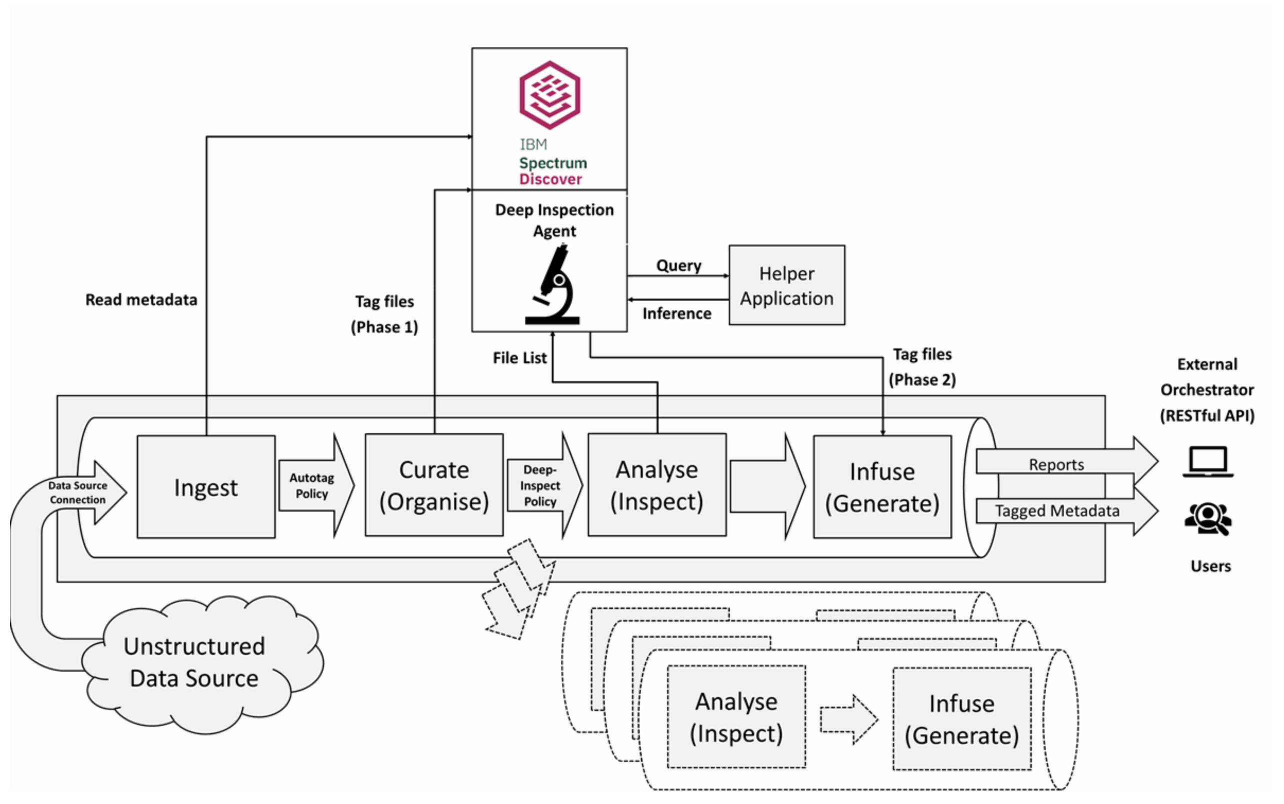


Figure 3-2 Data flow in data pipeline that is created by using IBM Spectrum Discover

Each component is described next.

3.2.1 Ingest

By using the data source connections, IBM Spectrum Discover scans the metadata at preset schedule or when started manually. Each scan reads the target file system metadata (or Object Storage, depending on the connection type) and generates a highly efficient search catalog.

A modern data lake can be filled with data that is coming from various applications. Data can arrive from sensors and IoT devices, user-generated document files, application-generated files (reports/logs) and many more. Multiple data sources can be configured within single IBM Spectrum Discover instance. Configuring multiple data sources within single IBM Spectrum Discover instance allows gathering data from all types of applications into a unified catalog. The catalog can be searched or filtered by using visual exploration or by providing filter queries in the search section of the GUI.

3.2.2 Curation

The curation allows us to build a higher level of metadata abstraction on top of available catalog information. This ability provides a way to filter the data for more specific analysis, which is achieved by completing the following tasks:

- Creating custom tags that can be attached to information in the catalog.
- Creating AUTOTAG policies that can filter files for specific criteria and applying the custom tag to those criteria.

In-training phase inference data files with lower confidence scores can be quickly identified in IBM Spectrum Discover and cataloging helps in retraining the model for improving accuracy. This feature supports effective data cataloging for model optimization

Multiple policies can be defined in one IBM Spectrum Discover instance so that more than one data set can be prepared. Multiple policies are shown in the dotted section of Figure 3-2 on page 56. Defining multiple policies in one IBM Spectrum Discover instance makes a single IBM Spectrum Discover work as an orchestrator for all possible organizational pipeline orchestration.

3.2.3 Analysis

The analysis phase is used to go beyond metadata scans and gain insights into data contents. IBM Spectrum Discover can include custom scripts that are written by users and developers. These scripts can interconnect with any IBM or third-party applications.

The sub phases of the analysis phase are described next.

CONTENTSEARCH

As described in “CONTENTSEARCH” on page 15, IBM Spectrum Discover can enrich metadata through content inspection of source data by using the built-in CONTENTSEARCH agent. To use this function, you define regular expressions (regex) to search for and create policies that use these regular expressions. IBM Spectrum Discover also provides predefined regular expressions to support commonly required searches, such as email, IP addresses, date formats, credit card numbers, and web URLs.

When the policy runs, the files or objects are retrieved from the source system by the CONTENTSEARCH agent, converted to text format if necessary, and searched by using the defined regular expressions. The results of the search are returned to IBM Spectrum Discover and the metadata of the files that are updated according to the policy’s definition.

DEEPINSPECT

Dealing with non-text file formats requires a mediator application that is aware of the file format in question. This application must be registered with IBM Spectrum Discover to use it. Registered applications can be manually started in the IBM Spectrum Discover GUI by selecting **Metadata** → **Policies** section. These policies can also be scheduled at specific intervals.

As described in “DEEPINSPECT with custom agents” on page 18, the DEEPINSPECT agent receives a list of all candidate files from the policy that starts it. The agent can iterate through the list and retrieve contents for reading. Agent reads each file and provides feedback to IBM Spectrum Discover. Therefore, the agent provides updated custom tag information and these updated tags are available for users to search.

Note: The DEEPINSPECT agent needs access to the source data for analysis because IBM Spectrum Discover collects only metadata during a scan. The agent can access data source connections and credentials that are saved in IBM Spectrum Discover if it is running as an IBM Spectrum Discover application (as a pod). Otherwise, the agent must be configured independently to connect same data source.

3.2.4 Helper application

The simplest implementation of DEEPINSPECT agent can perform independently, without the use of any third-party libraries other than those provided by the operating system. The CONTENTSEARCH agent that is provided is perfect example of that. It runs on the IBM Spectrum Discover host and does not require dedicated system resources.

The DEEPINSPECT agent can also use external applications to perform content analysis. This feature allows classic code to be reused where code libraries or APIs can be used. Also, the inspection agent can use more than one helper application at the same time.

The helper application can be outside of the IBM Spectrum Discover system. In this way, the suitable system and platform configuration can be sized for both. For example, the helper application can be hosted on a Microsoft Windows Server platform while IBM Spectrum Discover runs on Linux.

3.2.5 Query and inference

The x-ray must use a common interface to interact with the helper application. This feature provides a layered approach and independent deployment can be managed by specific teams. Some of the following methods are typical examples:

- REST API
- JSON query
- Secured HTML request

A typical query passes data and requests to the helper and the helper can return a suitable response. It is up to the DEEPINSPECT agent to decide whether the entire file data must be shared with helper application. The agent can perform a query in one of the following ways:

- Pass entire data to helper

  This method might require good bandwidth between the agent and helper to speed up the deep inspection phase.

- Read entire data, extract some content, and send a query to helper

  This method can intelligently control the data exchange between the two.

- Pass only file location

  This method works if the helper application has direct access to the source data.

Helper application provides an inference in response to the query it receives. The DEEPINSPECT agent can then process this inference and in turn, provide final response back to IBM Spectrum Discover.

Some of the use cases are listed in Table 3-1 to understand how it can work.

Table 3-1 Use cases.

Helper application	Agent query	Inference	Deep scanning action
Microsoft Office	Return data from file	Complete file data stream	► Search for specific patterns, legal keywords ► Classify document as public or confidential by way of classification AUTOTAG
PDF Reader	Return author, title from file	Name of document author and title	Update author name (`author`) and document title (`title`) by way of autotag
IBM PowerAI Vision	Run the x-ray file through a trained model and provide insights	Provided x-ray probably belongs to a tuberculosis patient with 80% confidence in inference	Update autotag `tuberculosis` as `yes`
	Run the x-ray file through a trained model and provide insights	Provided x-ray probably belongs to a tuberculosis patient with 70% confidence in inference	Update autotag `tuberculosis` as `no`

Applications, such as IBM PowerAI Vision and IBM Watson, are versatile and suitable as helpers, primarily because of their ability to be trained for various purposes. After a data model is created, it can be used for providing inference in an automated and continuous way.

3.2.6 Generate

In this final phase of data pipeline, IBM Spectrum Discover processes responses that are received from the DEEPINSPECT agent. The DEEPINSPECT agent provides return values as multiple (tag/value) pairs. These pairs are incorporated in the IBM Spectrum Discover catalog against the respective file's metadata.

Users can now search on newly acquired tag information by using the GUI's Search section.

The newly generated metadata can also be used as a source for new analysis, as shown in the following example:

- A hospital uses a data lake to collect x-rays, prescriptions, patient records, and diagnoses from various branches.
- IBM Spectrum Discover scans the data and marks x-ray files with new autotags:
  - Location (during ingestion, this tag is set based on the file path)
  - Diagnosis
- A DEEPINSPECT policy selects x-ray files, which include empty Diagnosis tag values. These file names are passed to a deep-inspect agent that scans files for "tuberculosis".
- An AI model is prepared with manually diagnosed tuberculosis x-rays.

- The deep-inspect agent shares each undiagnosed x-ray image with the AI model by using REST API calls and the AI model returns an inference to indicate whether the image might contain a tuberculosis x-ray.
- This diagnosis is returned to IBM Spectrum Discover and the `Diagnosis` tag is updated as `Yes` or `No`.
- A medical scientist can search on these autotags, along with metadata tags (creation date) to perform historical trend analysis.
- The diagnosed x-rays can again be fed to the AI model to increase accuracy of the inference.

This process forms a continuously running data pipeline with minimal manual user interaction.

3.2.7 Reports

IBM Spectrum Discover can provide tabular reports on an on-demand and on-scheduled basis. Each report can be customized to enlist files for a specific tag selection criterion that is decided by user.

On-demand reports can be retrieved by data scientists or users to determine whether their metadata filter generates any files of interest. This process is like running a search query, but users can save time in preparing filter in search boxes.

Scheduled reports make automated actions in IBM Spectrum Discover easier. For example, a script can use a report that lists all files that are classified as `confidential`. The script can move these files to a location, which is backed up daily.

3.2.8 Pipeline orchestration

IBM Spectrum Discover provides two prime means of automation to control every component of its own. These two facilities can be used to orchestrate entire data pipelines and create a fully automated solution for the customer:

- Scheduling

  A preset time schedule can be set to start tasks at a specific time:

  - Ingestion: Data source connections can be scanned daily to automatically check for new additions.
  - Curation: AUTOTAG policies can be started at periodic intervals to add values to tags.
  - Analysis: DEEPINSPECT and CONTENTSEARCH policies can be started at periodic intervals to perform an ongoing analysis on the data.
  - Generation: Reports can be prepared and refreshed at selected intervals.

- REST API control

  All schedulable components can be triggered by an external controller by making REST API calls to IBM Spectrum Discover instance. Instead of relying on a scheduled run, the external controller can trigger the tasks at-will and control the overall data movement in the pipeline.

  For more information about REST APIs that are supported by IBM Spectrum Discover, see IBM Knowledge Center.

3.3 Summary and value proposition

The features and components that are discussed in this chapter provide insight into how IBM Spectrum Discover can build an AI data pipeline with ease. IBM Spectrum Discover provides the following features and benefits for creating a data pipeline effectively:

- Ingestion from multiple sources:
  - Truly enterprise class ingestion allows scanning from all data lakes that are in the organization.
  - Ability to scan on-premises and cloud-based data resources make it a unified, future-ready tool.
- Speed and performance: The high-speed scanning engine can work at high speeds and scan billions of files daily:
  - Automation: Scheduling and API-based access to component execution allows designing customizable workflows that are aligned to business lifecycle. The pipeline can be kept moving as and when data is generated by the application. The data pipeline can also generate data in-time for providing inference when the user is expecting it.
  - Extendibility and versatility: By using the DEEPINSPECT agent, any external helper application can be integrated. This ability allows creating every possible data analysis design if the required data modeling tool is available. The possible solutions can be built for more than AI data processing.
  - Unification: Integration with multiple applications at once is possible by creating multiple policies and interfacing different deep-inspect agents. A single IBM Spectrum Discover instance can host all data processing workflows for the organization to provide maximum return on investment to customer.

For more information about a full-fledged AI pipeline use case to help understand the implementation, see Chapter 4, “Using artificial intelligence in medical imaging: JFR Challenge” on page 63.

4

Using artificial intelligence in medical imaging: JFR Challenge

In this chapter, we describe an IBM Spectrum Discover use case that uses artificial intelligence (AI) in medical imaging.

This chapter includes the following topics:

Note: For more information about the code that is described in this chapter, see the GitHub repository.

4.1 Introduction and overview

In this section, we provide an overview of this chapter's use case and give some context to the project (a data challenge) that served as groundwork to develop the IBM Spectrum Discover demonstration.

4.1.1 Context of AI in medical imaging

The potential of AI to transform almost all industries including healthcare (through the work and understanding of AI's potential in healthcare by organizational leaders and medical professionals) is increasingly evident as more real-world clinical applications emerge. As patient data sets become larger, manual analysis is becoming less feasible. AI has the power to efficiently process data far beyond our own capacity and enabled innovation in areas including healthcare data management, chemotherapy regimens, patient care, breast cancer risk, and even intensive care unit (ICU) death prediction.

The AI's role in healthcare, working on employing different types of AI technology, applications, managing limitations, sorting out challenges, and revealing industry opportunities, is growing. AI strategies are successfully deployed within the health care sector. Techniques, such as natural language processing, data analytics, and machine learning, are deployed across different contexts, such as hospital management, drug development, patient monitoring, and disease diagnosis, including medical imaging diagnosis.

The "Journées Francophones de Radiologie" 2019 (JFR 2019)

The "Journées Francophones de Radiologie" 2019 (JFR 2019) is a French national radiology congress that is focused on empowering radiologists with the knowledge to understand the transformative role of AI in healthcare. Its theme is the *augmented radiologist*, augmented in all its human, ethical, algorithmic, and therapeutic dimensions.

The following excerpt is from the JFR 2019's website and describes this theme:

Radiology is changing rapidly. Medical imaging is becoming more easily accessible, more and more precise, and interventional radiology continues to grow. This facilitates the involvement in all stages of the care journey: screening, diagnosis, therapeutic follow-up, targeted treatments, ethical decisions to stop care, and many others. All the technological advances do not only raise questions of medical nature, but also human and societal. The radiologist of tomorrow will be, even more than today, at the crossroads of new missions including; explaining to patients the images detected, cooperating with surgeons for specialized interventions, establishing effective care pathways with generalists, collaborating with public health physicians to interpret lots of data, interacting with engineers to explain the limitations of data science and communicating with patient organizations to better understand needs and to be able to inform widely about good practices.[1]

During its FR 2019 congress, the Société Française de Radiologie or SFR (French Radiology Society, website) organized a data challenge. An IBM French team, along with a Business Partner, decided to answer that challenge that consisted in building a deep learning model to detect lung nodules in 3D scans.

[1] The "Journées Francophones de Radiologie" 2019 (JFR 2019)

The IBM team won that challenge thanks to skills and IBM technologies. Following that achievement, they collaborated with an IBM Storage team to explore capabilities of IBM Storage solutions and build assets around it, which is described within the scope of this chapter.

4.1.2 The JFR Challenge

The challenge objectives were to help radiologists achieve better diagnoses by showing missed problems or problems that are challenging to be detected by using artificial intelligence, and to evaluate the state of the art of artificial intelligence algorithms for medical imaging applications. This challenge provides a great opportunity for imaging professionals to meet, debate, share experiences, and update their knowledge around the latest technological and scientific advances in medical imaging. It also facilitates the collaboration of clinical, fundamental, and industrial research, which is an essential keystone for efficient imaging that leads to the most appropriate therapeutic care for patients.

The competition is based on real cases. The following challenges were organized:

- Computation of muscles surface in sarcopenia (loss of skeletal muscle mass and function)
- Prediction of disabilities for patients with multiple sclerosis
- Classification of lung nodules according to their sizes in 3D scan images

The IBM France Systems team tackled the last classification challenge and won. The challenge represents an area where the contribution of artificial intelligence is undeniable. So, successful AI delivery imposed constituting a team with several skills, including data scientists, experts from imaging companies, one or more radiologists and students.

The IBM team was consisted of French IBMers from Montpellier and Paris who represented the data scientists and students. They partnered with a startup company that specializes in medical imaging who brought the imaging know-how and business experience.

In addition, radiologists from the cancer center were responsible for verifying the data sets, analyzing the annotations, and then controlling the quality of the work that was provided by the AI. All of those skilled profiles composed the IBM team and complemented each other.

This classification problem is challenging and complex. Usually, no consensus on the annotations of nodules exists among radiologists. This is mainly because of the entanglement with the respiratory and circulatory systems networks, and because nodules occur in different shapes, textures, and sizes.

Moreover, from a medical perspective, image classification has a significant impact, because lung cancer nodules often are diagnosed late because of the absence of characteristic symptoms, which makes lung cancer the leading cause of mortality in France. Therefore, a system that can assist radiologists in this detection and classification task can save patients' lives and have a great impact on public health.

Despite the problem difficulties, the IBM team was one of the three winners of the 2019 Data Challenge that was organized during the French Speaking Days of Radiology. The team used deep learning to detect nodules in lung scanners. The team also beat 12 other teams with an Area Under Curve (AUC) score of 89.9%. For more information, see this web page.

Challenging technical aspects

The contest organizers proposed three successive data sets. Each data set included approximately 350 computerized tomography scans (CT-scans, see Figure 4-21 on page 95 for examples), annotated by radiologists, anonymized, and delivered in DICOM format.

Scanners were showing nodules of various sizes, with nodules of more than 100 mm3 annotated. Each scanner corresponds to 100 - 200 MB of data and each data set contains approximately 80 GB of information, which represents a challenge in terms of processing capacity.

The competition occurred August - October 2019, and its objective was to classify whether 3D scanner exams had nodules bigger than 100 mm3.

A data set was provided by the organizers in two releases (350 images 2 months before the challenge, and 350 images 3 days before), both labeled with the ground truth by a panel of radiologists from multiple hospitals.

For the evaluation phase, an unlabeled set of 350 images was released with only 2 hours to process and return prediction results, on which was computed their performance for the final ranking.

Complex model inference is time-consuming; therefore, this 2-hour constraint required an efficient and resilient pipeline. The IBM solution ran on a cluster of three Power AC922 servers that are equipped with Nvidia V100 GPUs. The processing power together with the capacity to handle large models allowed to process successfully the test images in the required time with the best result among different teams in the challenge.

Designed prototype

The IBM solution included a pipeline that is designed to ingest 3D DICOM images as the pipeline input and return binary classification for each image whether it includes at least a nodule bigger than 100 mm3.

The 3D DICOM images processing chain consisted of the following main tasks:

- Data preprocessing

  Data preprocessing occurred on the original JFR data sets. Preprocessing includes isolating the lungs from the rest of the body by using a deep learning model for lungs detection, which introduced automatic rejection of artifacts that were detected outside the lungs. This is followed by extracting lung annotations normalizing density, and both are performed on CPUs.

- Model inference

  A 3D Retina UNet model (for more information, see this web page) state-of-the-art deep learning nodules segmentation model was trained by using the preprocessed images of JFR training and validation data sets and an open source data set named Lung Image Database Consortium image collection (LIDC-IDRI). For more information, see this web page. Inference was performed on GPUs.

- Postprocessing

  A Support Vector Machine (SVM), which is an SVM that predicted if a patient is pathological, having at least one nodule greater than 100 mm3, relaying on the extraction and nodules segmentation output of the previous tasks.

The entire pipeline was managed by a workload orchestrator to maximize CPUs and GPUs usage. Because of the lack of time and resources to implement efficient shared storage, the workload orchestrator was responsible for the following tasks:

- Copying the data to the suitable server.
- Monitoring resource usage and triggering a processing when resources where idle.
- Tracking progress and triggering next pipeline phases when one was done.
- Fetching back results.
- Handling errors by retrying and raising alerts.

Next steps

As highlighted in ““Challenging technical aspects” on page 65”, fitting the inference of 350 images in 2 hours was a challenge in itself. The lack of shared storage complicated the task and revealed the need for an efficient storage solution.

Following the challenge and with the skills that were gained in AI for medical imaging, the IBM Montpellier team conducted new studies to evaluate how IBM Storage solutions can enhance the capabilities and the performance of such an AI pipeline.

Following the challenge, the white paper *Storage for AI & IBM Power Systems Video: Medical data challenge*) was published.

> **Note:** To access this white paper, you must log in with your IBMid. If you do not have an IBMid, create one at this web page.

The white paper objectives are evaluated afterward how IBM storage solutions might help in this challenge by facilitating the solution development or improving the pipeline performance. It also describes how to define a target architecture that is based on IBM Power Systems and IBM Storage (IBM Spectrum Discover and IBM Spectrum Scale) for building a medical AI solution for 3D medical images.

4.1.3 Managing complex medical data

Among AI applications, handling medical images is more complex in processing and visualization because of the factors that are described in this section.

Data format

Medical images, such as lung images for instance are captured by way of Computed Tomography (CT) scans that uses x-rays to draw a 3D representation of the subject’s internal structure (in this case, the lung).

These medical imaging data can be stored in various formats that makes them complicated to manage, including the following examples:

- Digital Imaging and Communications in Medicine or DICOM (.dcm), which is the standard for communication and management of medical imaging information and its related metadata data.
- Neuroimaging Informatics Technology Initiative or NIfTI (.nii)
- Nearly Raw Raster Data or Nrrd (.nrrd)

For example, the MedPy library supports around more than 10 formats that are specific to medical imaging, in addition to the commonly used image formats, such as .jpg, and .png.

> **Note:** For more information about supported medical image formats, see this web page.

Many other libraries are available for managing files, such as pydicom (for managing DICOM files), and pynrrd (for managing Nrrd files). Visualization tools, such as SimpleITK, also are available. Learning how to use such specific libraries and tools, understanding the different data formats and specifications, their limitations and how they store information and metadata, require more work from data scientists.

Data complexity

Because medical images are often 3D data, their size complicates the management and copy of such data sets (on average resolution 5123 3D image contains 16 times more points than a 4 K image). Therefore, it requires large storage capacity.

In addition, the raw pixels a large amount of image metadata are stored, including the following examples:

- Pixel spacing: The real distance between two pixels on a specific image dimension.
- Orientation: Which dimension is the coronal, sagittal, or transverse direction.
- Data encoding: What the values represent, such as tissue density and type of organ.

Domain knowledge is often required to make full use of the information. Therefore, data scientists must acquire the required knowledge to develop deep learning models around it.

3D images visualization is also challenging and requires specialized 3D viewers, such as Medical Imaging Interaction Toolkit (MITK) or ITK-Snap. Visualization occurs on a different machine rather than the one that is used for deep learning, which increases network capacity that is required for files transfer.

Data availability

Most medical AI applications use supervised algorithms, in which AI models learn from labeled data. In the JFR challenge, for example, data was annotated by a team of radiologists; these labels are then used to train deep learning models.

But annotating a 3D lungs scan, usually at a segmentation level, is extremely complicated because segmentation requires to mark each pixel as belonging to a nodule (to determine the precise shape) in the set of slices where that nodule appears. Radiologists use a coloring tool to facilitate annotation, but each scan requires nevertheless tens of minutes to be carefully annotated.

In addition, because detecting a nodule requires deep medical expertise, this task can only be performed by radiologists. Even in that case, most radiologists do not agree on what is a nodule because nodules can occur in varying shapes and textures, which make them difficult to be differentiated from blood vessels and lung tissues. Therefore, annotation should be performed by a team of radiologists and results merged and averaged by using a consensus algorithm.

The difficulty of these constraints for creating data sets results in the limited availability of data sets. For example, regarding the lung cancer problem, the biggest identified open source data set was Lung Image Database Consortium image collection (LIDC-IDRI), which included more than 1010 annotated scans only. On other research medical problems, the shortage of annotated data can block AI research. Data augmentation techniques are explored, but they still cannot sort out this issue.

Integration with existing tools

Radiological Information Systems (RIS) are the core systems of images management in radiology services. They rely on technologies that are called *Picture Archiving and Communication Systems (PACS)* that can be on-premises or cloud-based installations. They are responsible for storing, managing, and displaying medical images of various types.

Any solution or prototype to be deployed and used in radiology services must be integrated within RIS, and more specifically to be integrated with PACS, which is the main interface for radiologists to work with medical imaging data.

4.1.4 Use case description

Consider a consortium of hospitals (more precisely, their radiology services) that plan to share and exchange data to expand their AI capabilities, each having their own storage solution, architecture, and organization.

Their main needs include:

- Sharing medical data between hospitals
- Using data from all hospitals to train deep learning models
- Having common deployed models that they can query to get real-time results for each new patient scan

Therefore, we propose the following architecture, as shown in Figure 4-1:

- One storage platform per hospital that is used by their radiology tools.
- A common IBM Spectrum Discover deployment.
- A common cluster of servers that are suitable for training and inference services.

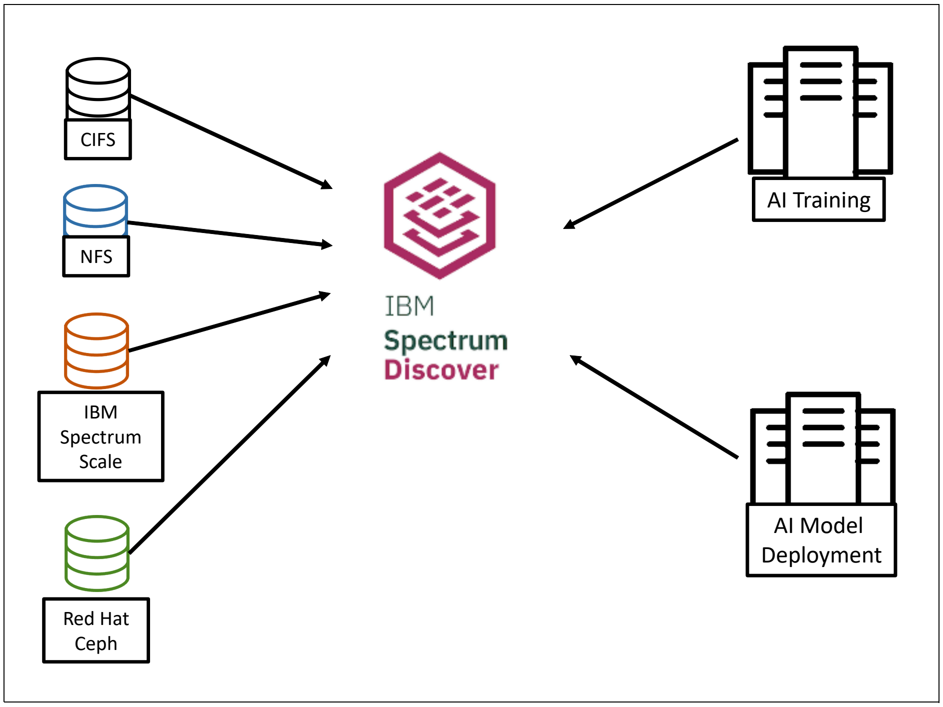


Figure 4-1 Use case architecture

4.2 Use case products

The following products were used in the use case:

- IBM Spectrum Discover
- Storage solutions: IBM Spectrum Scale, IBM Cloud Object Storage, or any other data source
- IBM Watson Machine Learning Community Edition

4.3 Benefits

In this section, we describe a detailed use case that explains how we use IBM Spectrum Discover features and how it can benefit this AI workflow.

In 4.5, "Implementation steps" on page 74, we provide implementation details, and share thoughts about the best practices and the motive behind the different technical decisions.

4.3.1 Unified data sources

IBM Spectrum Discover can ingest multiple types of data sources as described in Chapter 1, "IBM Spectrum Discover overview" on page 1.

In our use case, this feature implies that IBM Spectrum Discover can federate the hospitals data sources no matter the storage architecture they use. This feature helps to build an AI pipeline without changing the storage architecture, which limits the extra required cost.

4.3.2 CONTENTSEARCH policies

Metadata can be enriched through CONTENTSEARCH. As described in "CONTENTSEARCH" on page 15, this functionality is provided with IBM Spectrum Discover and examines the files to extract metadata.

Based on Apache Tika, the CONTENTSEARCH can process hundreds of different file types by applying the relevant process to extract information, such as the following examples:

- Run optical character recognition (OCR) over image files.
- Extract text from PDF files.
- Extract compressed archive and dig into the content.

The CONTENTSEARCH also natively supports DICOM metadata extraction, which is relevant in our use case.

In IBM Spectrum Discover, data is identified by policy filter and passed to the CONTENTSEARCH agent. Then, regular expressions are applied to each file content, and if values matched, matching metadata is cataloged in IBM Spectrum Discover (see Figure 4-2).

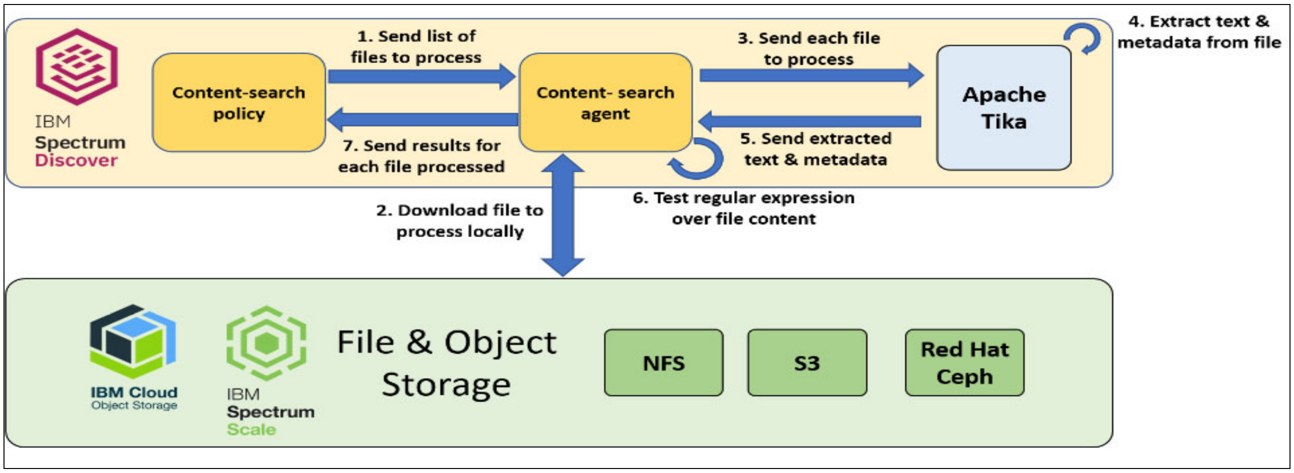

IBM Spectrum Discover
Content-search policy
1. Send list of files to process
Content- search agent
3. Send each file to process
Apache Tika
4. Extract text & metadata from file
7. Send results for each file processed
5. Send extracted text & metadata
2. Download file to process locally
6. Test regular expression over file content
IBM Cloud Object Storage
IBM Spectrum Scale
File & Object Storage
NFS
S3
Red Hat Ceph

Figure 4-2 IBM Spectrum Discover content search

4.3.3 Capabilities extension

The CONTENTSEARCH allows to gather metadata from files, but has its limitations. You might want to get metadata from an unsupported file type and want run a specific process to retrieve the relevant metadata.

As discussed in “DEEPINSPECT with custom agents” on page 18, DEEPINSPECT allows to solve this problem and to extend the capabilities of IBM Spectrum Discover. To perform DEEPINSPECT, an external application is required.

The external application interfaces with IBM Spectrum Discover by way of API and can access the source storage. Data is identified by IBM Spectrum Discover by policy filter and passed to the application as pointers through a messaging queue. Then, the application performs the convenient actions on source data and returns a completion status to IBM Spectrum Discover, which might include enriched metadata for the processed data records. If it does include enriched metadata, IBM Spectrum Discover catalogs that metadata and makes it immediately searchable.

Figure 4-3 shows this use case.

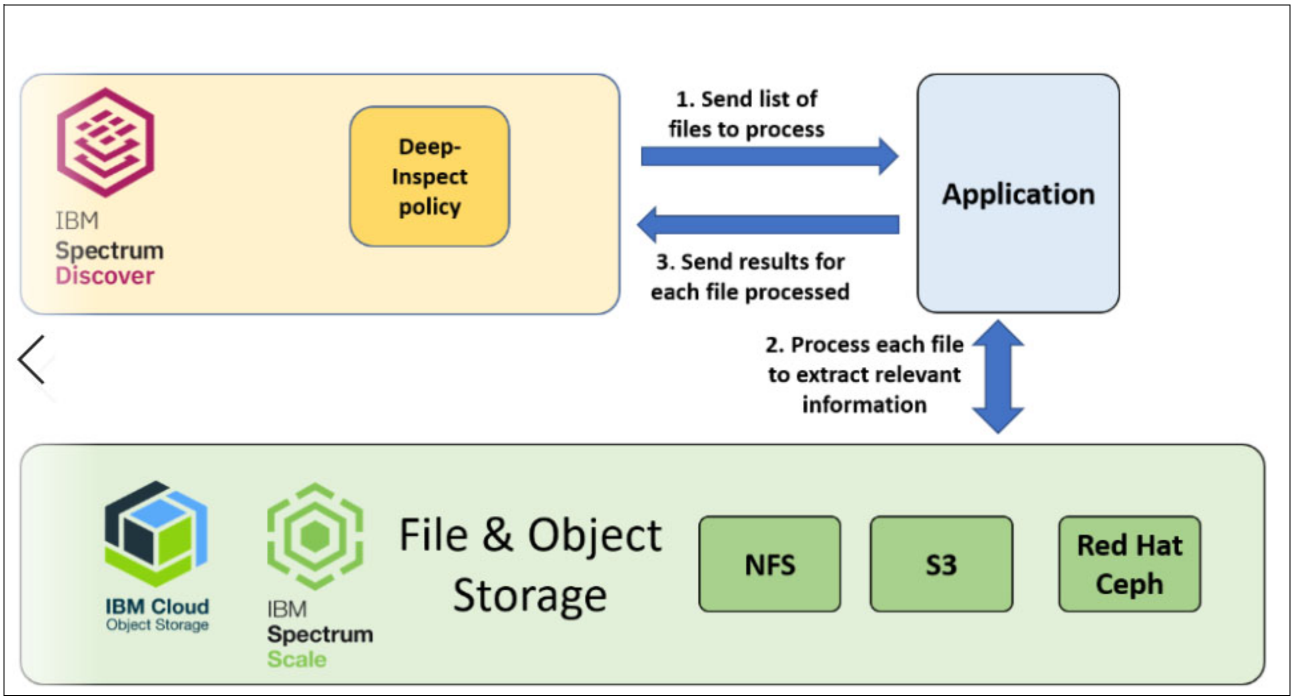


Figure 4-3 Deep inspection

This external application interface is an important part of this use case and we use it to complete the following tasks:

- Extract metadata from an external database.
- Send files to inference and catalog the result.

4.3.4 Data copy

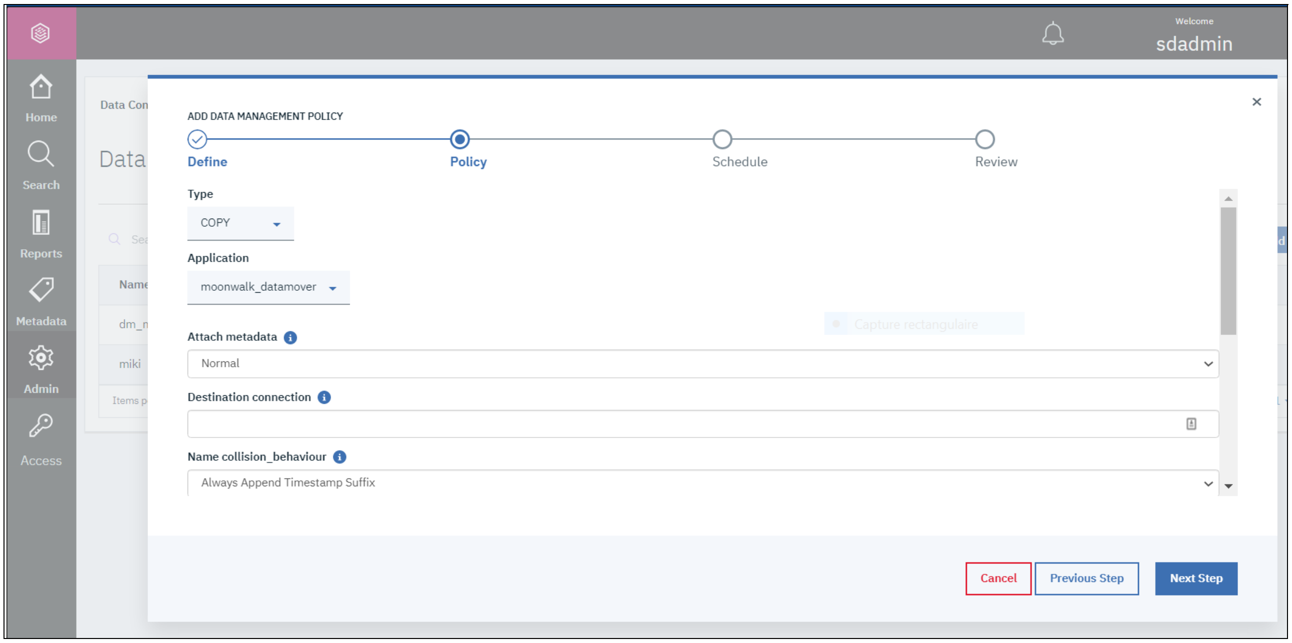
Thanks to IBM Spectrum Discover, we can build a data set from multiple sources of data.

However, when we need that full data set to train deep learning models on it, data must be gathered on a single storage source; that is, the source that is closest to the training server.

At the time of writing in a beta version, IBM Spectrum Discover provides a data movement feature that is based on Moonwalk, which is a software technology that helps implement, run, and automate data management and that must be installed independently (for more information, see “External agent-based movement” on page 24).

A special policy that uses Moonwalk can be created and used in COPY or MOVE mode, which is specified when the data movement policy is defined.

Figure 4-4 shows the configuration of such a policy.



ADD DATA MANAGEMENT POLICY

Type: COPY

Application: moonwalk_datamover

Attach metadata: Normal

Destination connection:

Name collision_behaviour: Always Append Timestamp Suffix

Figure 4-4 Moonwalk policy configuration

4.3.5 API interfacing

Full support of REST APIs is one of the important features of IBM Spectrum Discover, which facilitates applying almost all actions programmatically through HTTP requests. Full support of REST APIs facilitates many different use cases. In our example, we use APIs to query IBM Spectrum Discover and produce graphs that provide insights about our data set.

4.4 Use case architecture

The AI pipeline includes how to ingest, organize, and analyze data; then, ultimately train models to create AI-driven insights from that data. All of these tasks are essential for efficient data science. Efficiency of an AI pipeline is directly tied to addressing the challenges that are explored 4.1.3, "Managing complex medical data" on page 67 related to this healthcare use case, with the correct IT infrastructure.

The AI pipeline process focuses on two main phases: the training phase where a model is produced, and the inference phase where this trained model is used to infer new images. Before applying both phases, we first must index the hospitals' different local data sources in the common IBM Spectrum Discover infrastructure.

The following ideas are associated with this use case. These ideas have the double objective of being a realistic use of such a system and representative of a wide range of IBM Spectrum Discover features:

- Index the hospitals local data sources (of varying types: Spectrum Scale, Cloud Object Storage, NFS, and so forth) in the common IBM Spectrum Discover infrastructure.
- Training use cases:
  - Extract metadata from DICOM files, query a database to bring more data into IBM Spectrum Discover metadata; and use reports to extract data set insights.
  - Curate data set by using the previous analysis, eliminate or fix outliers, detect imbalanced classes and other data curation activities.
  - Run AI training with DICOM files indexed in IBM Spectrum Discover, which requires copying the data to the storage pool that is closest to the training server, which is common to all hospitals in this use case.
  - Deploy or redeploy the containerized inference services that radiologists query for new patient diagnosis.
- Inference use cases:
  - On-demand inference workflow, where radiologists submit new scans to the inference servers and expect detection results in a few minutes.
  - New model release, when a new model version is available, existing scans are reprocessed, the results are compared, and any significant difference in diagnosis raises an alert.

Note: Such an inference service assists the radiologists in their diagnosis, and provides a way to run automatic detections on scans, even if the patient had a scan for another pathology.

This case was discussed and validated with radiologists. CT scans are stored in a raw format and then post-processed (with filters), depending on the organ that the radiologist wants to inspect. But the existence of this raw image means that we can inspect all organs that are visible in this image.

Considering the diseases, such as lung cancer, that are particularly lethal because of their late diagnosis, such a service increases early detection and benefits public health with little extra time spent by a radiologist (only to inspect when an alert is raised by one of these models).

4.5 Implementation steps

In this section, we provide the implementation details of the components that were involved in the use case and discuss implementation choices that were made.

4.5.1 AI inference service

An AI service was built during the JFR challenge. It embeds a nodules detection deep learning model and is used as a root element of this use case.

In the following sections, we describe how this AI service works and how it is deployed.

Deep learning model

The model used for nodules detection is a 3D Retina U-Net and provides nodules segmentation filter as output. The version that is used in this demonstration was trained on the LIDC-IDRI data set.

Inference includes a preprocessing phase and takes approximately 90 seconds to complete on a Nvidia V100 GPU.

Software components

This model is based on Medical Detection Toolkit, an open source framework for medical deep learning applications, which is based on Pytorch 1.4.0.

This framework and its dependencies were installed on an IBM Power AC922 server by using the Watson Machine Learning Community Edition toolkit, which is a distribution of data science and libraries that are optimized for Power Systems and provided by IBM through a set of Conda channels. It is used for the training and the inference phases of this model.

Deployment

The AI service is deployed in a Docker container on a Red Hat OpenShift Version 3.11 cluster running on IBM Power servers. A Nvidia V100 GPU is attached and dedicated to this container.

This container has a persistent volume on a Spectrum Scale cluster (accessed by NFS; see this web page) that is indexed by IBM Spectrum Discover. Therefore, any data that is indexed in IBM Spectrum Discover that is moved or copied on that local storage is accessible by the AI service.

This container includes an external access through SSH that is used for inference. It also exposes an API that is used to submit files to inference (see “Deployment options” on page 93 the on-demand inference section for discussion about deployment options of that AI service).

An application that needs to run an inference (typically an IBM Spectrum Discover application) therefore connects through SSH and calls a **run_demo.py** Python script or send a query to the API that calls the same script. This script takes as parameter the path of the file to infer and calls the Medical Detection Toolkit framework and outputs the results on standard output in a JSON format.

Note: The storage path of an image as seen by IBM Spectrum Discover (and provided as parameter to the `run_demo.py` script) is not necessarily the same as the one where this is mounted in the Docker container running the inference.

For example, the path on the storage is `/export/hospitals-data/hospital-10/DICOM-1287.dcm`

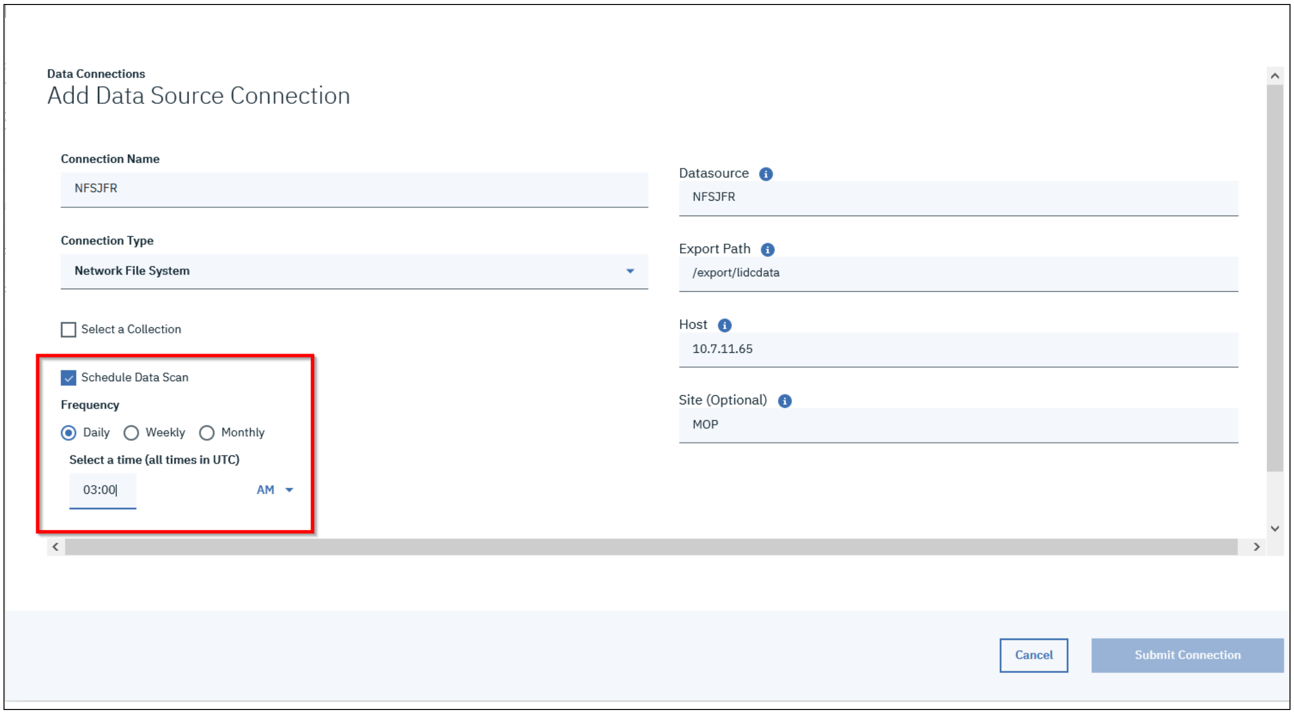
The directory `hospitals-data/` is mounted in `/data-source` in the container.

The image path (as seen by the inference code) is `/data-source/hospitals-data/hospital-10/DICOM-1287.dcm`. Therefore, users must be careful about the paths: match the paths (mount in the same absolute path) or "translate" when the inference code receives it.

4.5.2 Unified data sources

One of the features of IBM Spectrum Discover is the ability to connect to multiple types of storage to index their files and metadata. This feature is especially useful in our use case because it allows each hospital to keep their storage solution and infrastructure. It reduces the cost because they need to invest only in the hardware running IBM Spectrum Discover to federate all data sources.

Adding and scanning data sources allows to catalog their files and retrieve system metadata, such as owner, creation date, and size. As shown in Figure 4-5, scanning can be run manually or scheduled periodically.



Data Connections

Add Data Source Connection

Connection Name: NFSJFR

Connection Type: Network File System

□ Select a Collection

☑ Schedule Data Scan

Frequency: ● Daily ○ Weekly ○ Monthly

Select a time (all times in UTC): 03:00 AM

Datasource: NFSJFR

Export Path: /export/lidcdata

Host: 10.7.11.65

Site (Optional): MOP

Figure 4-5 Add data source

For more information about connecting to these data sources, see IBM Knowledge Center.

As described in 1.5, “Architecture” on page 8,IBM Spectrum Discover supports several data source connections. In this use case, we use NFS.

4.5.3 Training

In this section, we describe the first use-cases that are related to the data cataloging, cleansing, and training of the deep learning model.

Metadata cataloging

The DICOM standard defines a specific amount of optional metadata that is embedded in medical imaging DICOM files; giving characteristics about the image, scanner, or patient.

In that flow, we want show that we can extract DICOM metadata and use an external database to query more metadata and store them in IBM Spectrum Discover.

DICOM data set

As described in Chapter 2, “Generic imagery use cases” on page 29, we can work with DICOM data sets. In this example, we use the LIDC-IDRI data set (containing 1010 NRRD scans). Because NRRD does not hold patient metadata but only image metadata (such as spacing), we converted these images to DICOMs.

We also generated random metadata to simulate real data that a hospital provide because nearly all DICOMs that are available online are anonymized. In the detail, we complete the following metadata:

- PatientID: Social Security number referencing uniquely the patient
- PatientName
- PatientAge
- PatientSex

These metadata is defined in the DICOM standard (see this web page).

We also define a database with more metadata to represent the patient’s database of the hospital. It is a simple CSV file that includes the following fields:

- Patient SSN: Key that links patient to the `PatientID` field in IBM Spectrum Discover
- Address
- Name
- Sex
- Age
- Mail
- Blood group
- Smoker (if the patient is or was a smoker)

DICOM metadata extraction

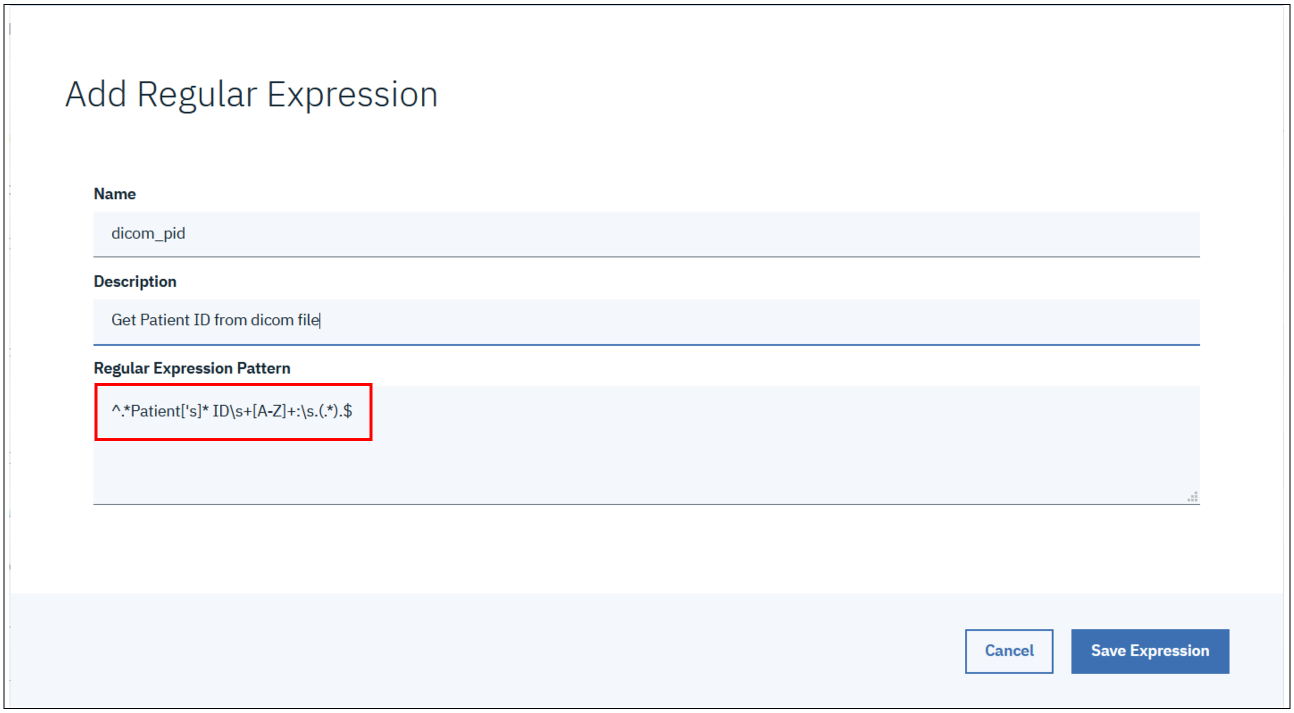
IBM Spectrum Discover natively allows to extract DICOM metadata thanks to CONTENTSEARCH policies.

To extract metadata from DICOM files, we performed the following steps:

1. Create specific regular expressions.
2. Create more metadata tags and characteristics.
3. Define and run a CONTENTSEARCH policy.
4. View the cataloged metadata.

Creating specific regular expressions

In IBM Spectrum Discover GUI, select **Metadata** → **Regular Expressions**. Then, add several entries by clicking **Add RegEx** (see Figure 4-6).



Add Regular Expression

Name

dicom_pid

Description

Get Patient ID from dicom file

Regular Expression Pattern

^.*Patient['s]* ID\s+[A-Z]+:\s.(.*).$

Figure 4-6 Add Regular Expression

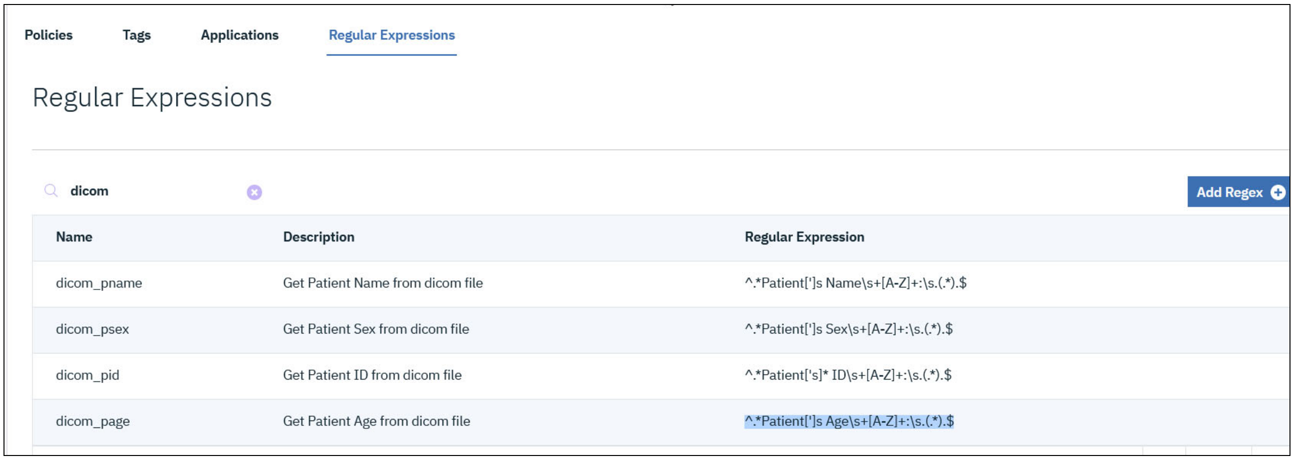
In this example, we define a regular expression for the Patient ID in DICOM files.

This regular expression helps to match and capture a specific pattern as Patient ID. The regular expression follows the Python regular expressions syntax. For this use case, we define the regular expressions as listed in Table 2-1.

The regular expressions that we defined in Figure 4-7 on page 78 are listed in Table 4-1.

Table 4-1 Regular expressions defined for the use case

Name	**Description**	**Regular expression pattern**
dicom_pname	Get Patient Name from dicom file	^.*Patient[']s Name\s+[A-Z]+:\s.(.*).$
dicom_psex	Get Patient Sex from dicom file	^.*Patient[']s Sex\s+[A-Z]+:\s.(.*).$
dicom_pid	Get Patient ID from dicom file	^.*Patient[']s* ID\s+[A-Z]+:\s.(.*).$
dicom_page	Get Patient Age from dicom file	^.*Patient[']s Age\s+[A-Z]+:\s.(.*).$



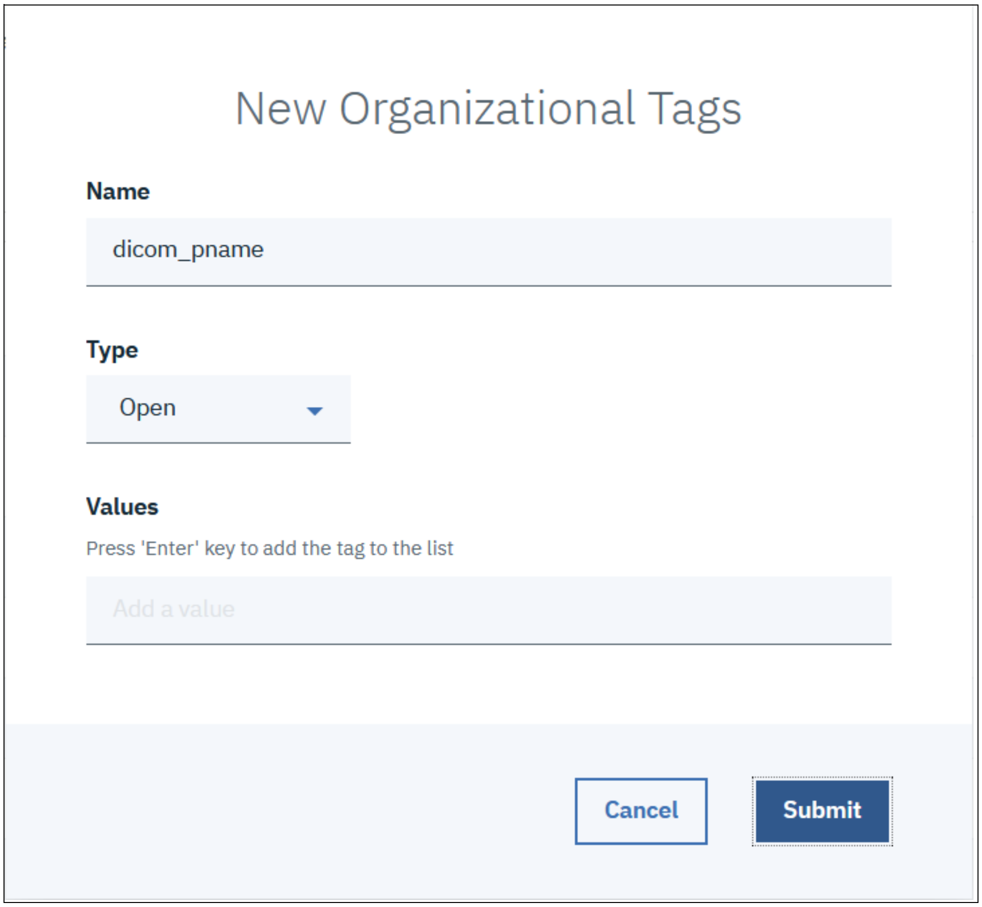
Regular Expressions

Name	Description	Regular Expression
dicom_pname	Get Patient Name from dicom file	^.*Patient[']s Name\s+[A-Z]+:\s.(.*).$
dicom_psex	Get Patient Sex from dicom file	^.*Patient[']s Sex\s+[A-Z]+:\s.(.*).$
dicom_pid	Get Patient ID from dicom file	^.*Patient['s]* ID\s+[A-Z]+:\s.(.*).$
dicom_page	Get Patient Age from dicom file	^.*Patient[']s Age\s+[A-Z]+:\s.(.*).$

Figure 4-7 Regular Expressions

Creating more metadata tags and characteristics

In IBM Spectrum Discover GUI, select **Metadata** → **Tags**, add several entries that match the extra metadata by clicking **Add**. The New Organizational Tags window opens (see Figure 4-8).


New Organizational Tags

Name: dicom_pname

Type: Open

Values

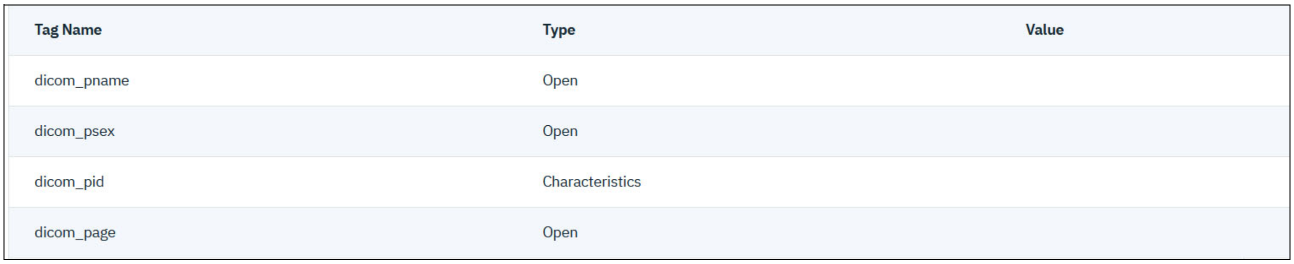
Press 'Enter' key to add the tag to the list

Figure 4-8 New Organizational Tags

For this use case, we create the tags that are filled by the CONTENTSEARCH (see Table 4-2). Figure 4-9 shows the tags created for this scenario.

Table 4-2 More tags created

Name	Type
dicom_pid	Characteristics
dicom_pname	Open
dicom_psex	Open
dicom_page	Open

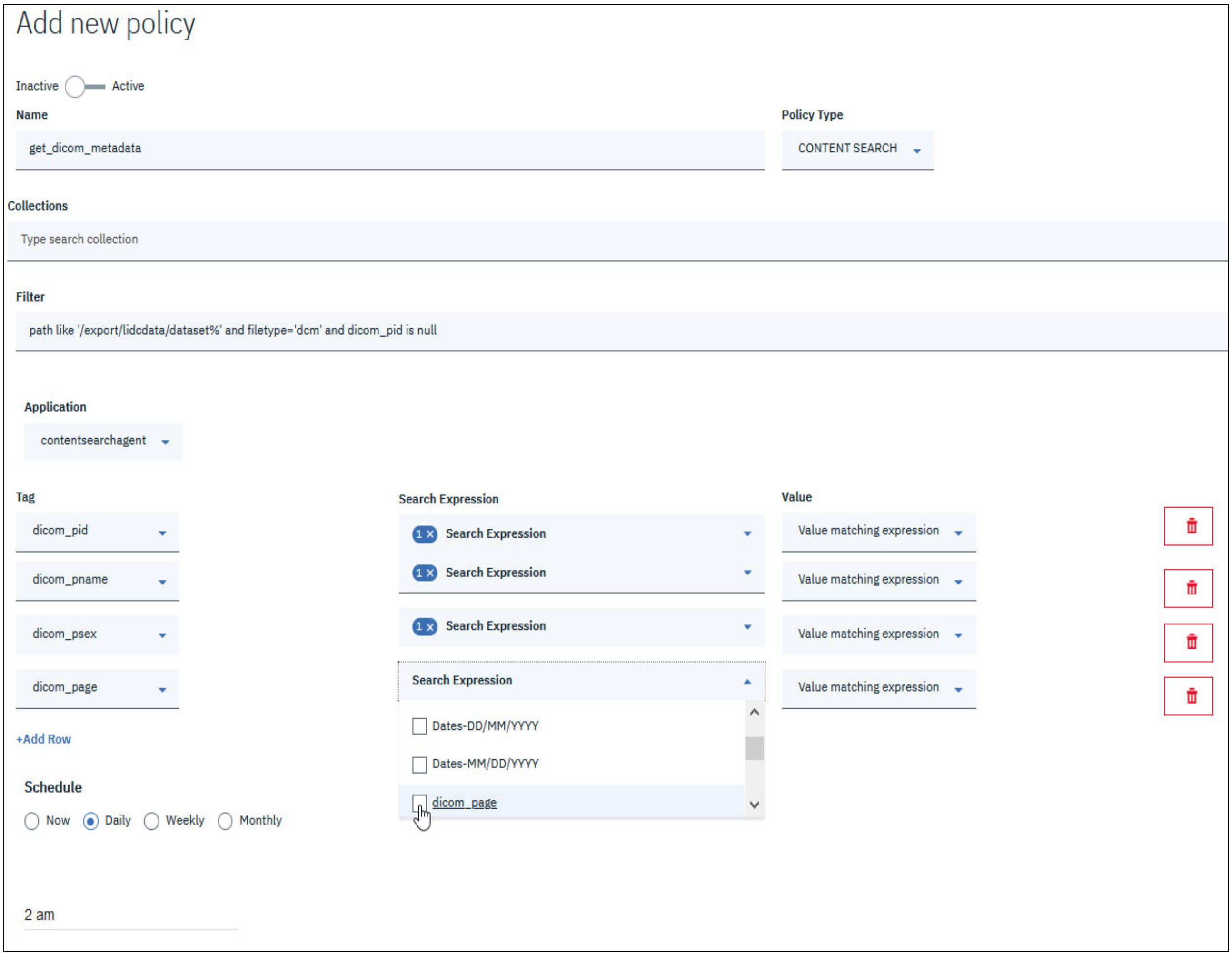


Tag Name	Type	Value
dicom_pname	Open	
dicom_psex	Open	
dicom_pid	Characteristics	
dicom_page	Open	

Figure 4-9 DICOM tag list -1/2

Defining and running a CONTENTSEARCH policy

After defining the regular expressions and extra metadata tags, a CONTENTSEARCH policy must be defined by clicking **Add Policy** (see Figure 4-10 on page 80).



Add new policy

Inactive ○ Active

Name: get_dicom_metadata

Policy Type: CONTENT SEARCH

Collections: Type search collection

Filter: path like '/export/lidcdata/dataset%' and filetype='dcm' and dicom_pid is null

Application: contentsearchagent

Tag	Search Expression	Value
dicom_pid	1 × Search Expression	Value matching expression
dicom_pname	1 × Search Expression	Value matching expression
dicom_psex	1 × Search Expression	Value matching expression
dicom_page	Search Expression	Value matching expression

□ Dates-DD/MM/YYYY

□ Dates-MM/DD/YYYY

□ dicom_page

+Add Row

Schedule

○ Now ● Daily ○ Weekly ○ Monthly

2 am

Figure 4-10 Add new policy

A CONTENTSEARCH policy consists of the following components:

- **Policy name**: `get_dicom_metadata`
- **Policy type**: CONTENTSEARCH
- **Filter**: To run the policy against a subset of the cataloged metadata path, such as `/export/lidcdata/dataset%` and `ffiletype='dcm'` and `dicom_pid` is not null and `dicom_smoker` is null

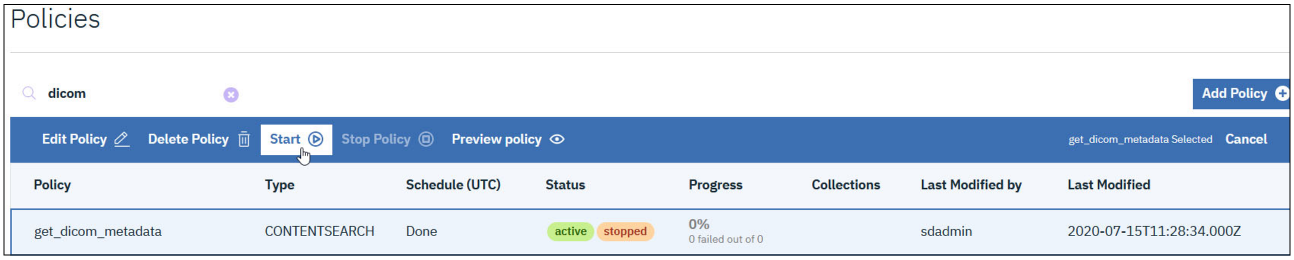
  The policy runs on only the `dcm` files in the `/export/lidcdata/dataset` folder that were not yet processed (`dicom_pid` is not null, which means the CONTENTSEARCH was run, but the `dicom_smoker` is null, which means the DEEPINSPECT was not yet processed.
- **Application**: In this use case, the CONTENTSEARCH agent `contentsearchAgent`.
- **Metadata list**: Used along with the matching regular expression.
- **Action**: Used to retrieve the value or return TRUE/FALSE if a match exists.
- **Schedule**: To run a policy now, daily, weekly, or monthly.

Table 4-3 lists the search expression that is used for each tag.

Table 4-3 Search expression used for each tag

Tag	Search expression	Value
dicom_pname	dicom_pname	Value matching expression
dicom_pid	dicom_pid	Value matching expression
dicom_psex	dicom_psex	Value matching expression
dicom_page	dicom_page	Value matching expression

After the policy is created, it automatically runs according to the schedule and it can be started manually by selecting the policy and clicking **Start** (see Figure 4-11).



Policy	Type	Schedule (UTC)	Status	Progress	Collections	Last Modified by	Last Modified
get_dicom_metadata	CONTENTSEARCH	Done	active stopped	0% 0 failed out of 0		sdadmin	2020-07-15T11:28:34.000Z

Figure 4-11 Starting policy

The Progress field displays the number of processed files. A policy can also be created, edited, deleted, or started through the REST API.

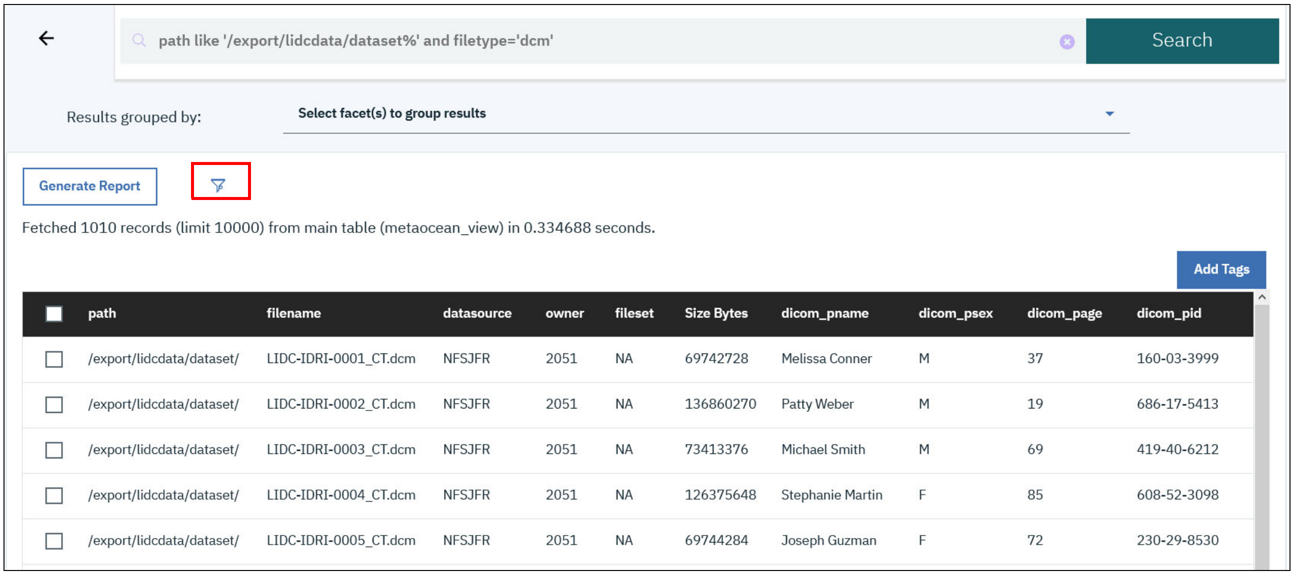
Viewing the cataloged metadata

After the policy completion, we can start browsing the metadata that was gathered by IBM Spectrum Discover.

From the Search page, we can, for example, run a path query, such as:

```
path like '/export/lidcdata/dataset%' and filetype='dcm'
```

Columns can be added by clicking the **funnel** icon and display the relevant DICOM metadata (see Figure 4-12).


path like '/export/lidcdata/dataset%' and filetype='dcm'
Search
Results grouped by:
Select facet(s) to group results
Generate Report
Add Tags

Fetched 1010 records (limit 10000) from main table (metaocean_view) in 0.334688 seconds.

□	path	filename	datasource	owner	fileset	Size Bytes	dicom_pname	dicom_psex	dicom_page	dicom_pid
□	/export/lidcdata/dataset/	LIDC-IDRI-0001_CT.dcm	NFSJFR	2051	NA	69742728	Melissa Conner	M	37	160-03-3999
□	/export/lidcdata/dataset/	LIDC-IDRI-0002_CT.dcm	NFSJFR	2051	NA	136860270	Patty Weber	M	19	686-17-5413
□	/export/lidcdata/dataset/	LIDC-IDRI-0003_CT.dcm	NFSJFR	2051	NA	73413376	Michael Smith	M	69	419-40-6212
□	/export/lidcdata/dataset/	LIDC-IDRI-0004_CT.dcm	NFSJFR	2051	NA	126375648	Stephanie Martin	F	85	608-52-3098
□	/export/lidcdata/dataset/	LIDC-IDRI-0005_CT.dcm	NFSJFR	2051	NA	69744284	Joseph Guzman	F	72	230-29-8530

Figure 4-12 Selecting the funnel icon

Metadata enrichment

Metadata often is not fully stored in the DICOM file because of security reasons (you might want to share DICOM data, but not the patient information), accessibility reasons (it is complicated and slow to query), and limitations of the DICOM standard.

For these reasons, a database often is used that contain more patient data. Some of this data can be queried and added in IBM Spectrum Discover to enrich its metadata.

This database can be any type or system that you can query from a Python script. For simplicity and because database query is not within the scope of this book, we simulate a patient database with the CSV file that is shown in Example 4-1, which contains for each patient their SSN, name, address, blood group, mail, age, sex and if they are or were a smoker (generated data).

Example 4-1 Sample CSV file

```
680-81-4061,Harold Hayes,AB-,dana06@hotmail.com,49,M,False
206-24-7335,Crystal James,AB-,nicole98@yahoo.com,44,F,True
209-75-1785,Sean Odonnell MD,AB-,marksmith@gmail.com,30,M,False
214-75-3175,John Miller,O+,kelleykyle@gmail.com,82,M,True
370-49-5867,Noah May,B-,edwingalloway@gmail.com,62,F,False
820-19-6236,Mr. Louis Merritt PhD,A+,david55@gmail.com,68,M,False
024-30-9976,Heather Lawson,O-,samantha09@yahoo.com,82,F,False
370-15-0515,Michelle Kemp,A+,tyleredward@gmail.com,32,F,False
633-27-5940,Shannon Koch,AB+,tpalmer@yahoo.com,69,F,True
```

Creating an IBM Spectrum Discover application

An IBM Spectrum Discover application is a program that interfaces with IBM Spectrum Discover and, for each file to process, performs whatever is necessary to retrieve the relevant metadata.

In this case, the application performs the following steps:

1. Queries IBM Spectrum Discover to retrieve the DICOM Patient ID (its Social Security number).
2. Queries the database to retrieve the extra metadata.
3. Returns the result to IBM Spectrum Discover.

Developing a custom, business-fitted application is made easy by using the IBM Spectrum Discover application SDK. For more information, see this web page.

The SDK manages most communication tasks between the application and IBM Spectrum Discover. The only part to develop is the part that is related to extract the relevant metadata for each file. The full code of this application is available at GitHub.

Running the IBM Spectrum Discover application

An IBM Spectrum Discover application can be run by using one of the following methods:

- As a simple Python program running on a server, this program should run as a daemon to be available at any time when the DEEPINSPECT policy runs.
- As a Docker container, which makes the portability and deployment easier.
- As a Kubernetes application, which simplifies the portability, deployment, scalability, and application updates. The application can run on the IBM Spectrum Discover Kubernetes cluster without issue.

> **Note:** In the first two options, you must ensure access to the data source that is cataloged by IBM Spectrum Discover if file must be read.

When running as a Kubernetes application on the IBM Spectrum Discover infrastructure, the application can benefit from the IBM Spectrum Discover connectivity to the data sources.

In this use case, we chose to run the application as a Kubernetes application by using the following process:

1. Build the container image: `$ docker build -t myagent/db-metadata-agent .`
2. Get an IBM Spectrum Discover API token (see Example 4-2).

Example 4-2 Get an IBM Spectrum Discover API token

```
$ TOKEN=$(curl -k -u <USER>:<PASSWORD> https://<DISCOVER_HOST>/auth/v1/token -I | grep
X-Auth-Token |cut -f 2 -d " ")
```

3. Create a JSON file that describes the application and the required environment variables (see Example 4-3).

Example 4-3 JSON file

```
$ cat db-metadata-agent.json
{
   "repo_name": "myagent/db-metadata-agent",
   "version": "0.0.1",
   "description": "Agent to extract metadata from database",
   "application_name": "db-metadata-agent",
   "SPECTRUM_DISCOVER_HOST"="https://10.3.74.20",
   "APPLICATION_USER"="sdadmin",
   "APPLICATION_USER_PASSWORD"="Passw0rd"
}
```

4. Run the application, as shown in Example 4-4.

Example 4-4 Run the application

```
$ curl -s -k -H "Authorization: Bearer $TOKEN" -H 'Content-type: application/json' -X POST
https://<DISCOVER_HOST> /api/application/appcatalog/helm -d@db-metadata-agent.json
```

5. Confirm that the application is running by running the command that is shown in Example 4-5).

Example 4-5 Confirm that the application is running

```
$ kubectl get pods -n spectrum-discover | grep db-metadata-agent
db-metadata-agent-application-1594826353-5f9b6bd8dc-6gg2v      1/1     Running     0
3m12s
```

Creating more metadata tags and characteristics

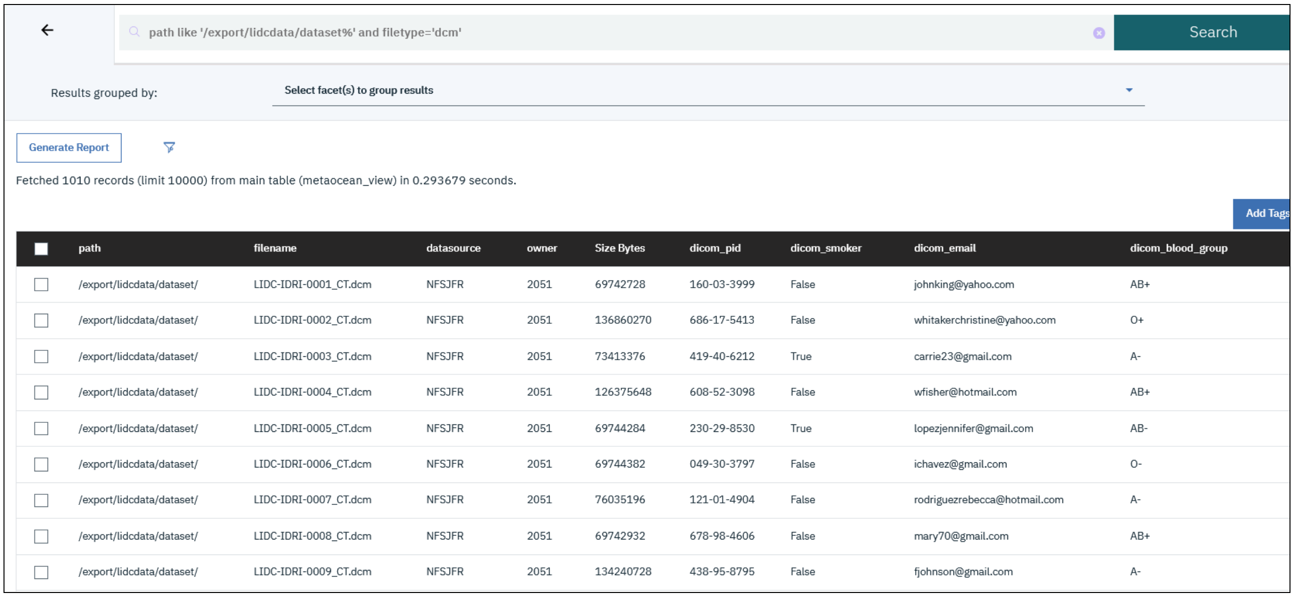
In IBM Spectrum Discover GUI, from **Metadata** → **Tags**, we added several entries that matched the extra metadata to be retrieved from the database by clicking **Add**.

For this use case, we create the tags that are listed in Table 4-4 that are filled by the DEEPINSPECT policy:.

Table 4-4 Extra tags

Name	Type
dicom_smoker	Characteristics
dicom_email	Characteristics
dicom_blood_group	Characteristics

Figure 4-13 shows these extra metadata tags.



path like '/export/lidcdata/dataset%' and filetype='dcm'

Fetched 1010 records (limit 10000) from main table (metaocean_view) in 0.293679 seconds.

□	path	filename	datasource	owner	Size Bytes	dicom_pid	dicom_smoker	dicom_email	dicom_blood_group
□	/export/lidcdata/dataset/	LIDC-IDRI-0001_CT.dcm	NFSJFR	2051	69742728	160-03-3999	False	johnking@yahoo.com	AB+
□	/export/lidcdata/dataset/	LIDC-IDRI-0002_CT.dcm	NFSJFR	2051	136860270	686-17-5413	False	whitakerchristine@yahoo.com	O+
□	/export/lidcdata/dataset/	LIDC-IDRI-0003_CT.dcm	NFSJFR	2051	73413376	419-40-6212	True	carrie23@gmail.com	A-
□	/export/lidcdata/dataset/	LIDC-IDRI-0004_CT.dcm	NFSJFR	2051	126375648	608-52-3098	False	wfisher@hotmail.com	AB+
□	/export/lidcdata/dataset/	LIDC-IDRI-0005_CT.dcm	NFSJFR	2051	69744284	230-29-8530	True	lopezjennifer@gmail.com	AB-
□	/export/lidcdata/dataset/	LIDC-IDRI-0006_CT.dcm	NFSJFR	2051	69744382	049-30-3797	False	ichavez@gmail.com	O-
□	/export/lidcdata/dataset/	LIDC-IDRI-0007_CT.dcm	NFSJFR	2051	76035196	121-01-4904	False	rodriguezrebecca@hotmail.com	A-
□	/export/lidcdata/dataset/	LIDC-IDRI-0008_CT.dcm	NFSJFR	2051	69742932	678-98-4606	False	mary70@gmail.com	AB+
□	/export/lidcdata/dataset/	LIDC-IDRI-0009_CT.dcm	NFSJFR	2051	134240728	438-95-8795	False	fjohnson@gmail.com	A-

Figure 4-13 Extra metadata tags

Creating and running a DEEPINSPECT policy

After running the IBM Spectrum Discover application, we create metadata tags and a DEEPINSPECT policy that mist be defined by clicking **Add Policy**.

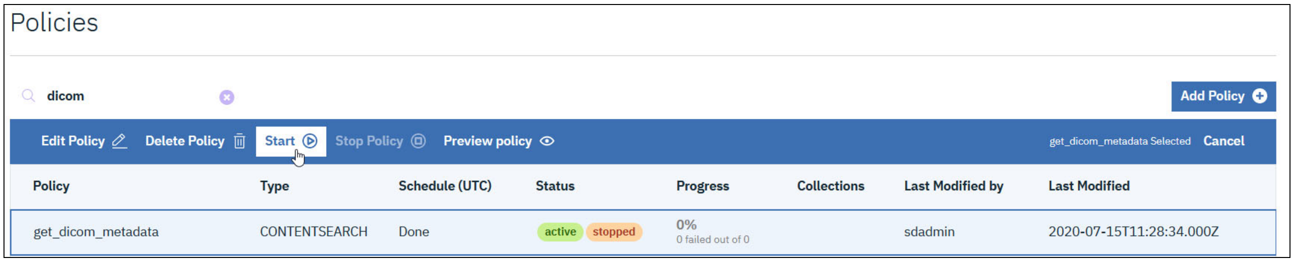
A CONTENTSEARCH policy consists of:

- **Policy name**: `get_dicom_metadata_from_db`
- **Policy type**: DEEPINSPECT
- **Filter**: To run the policy against a subset of the cataloged metadata: `path like '/export/lidcdata/dataset%'` and `filetype='dcm' and dicom_pname is not null`

  The policy runs only on the `dcm` files in the `/export/lidcdata/dataset` folder that were not yet processed (`dicom_pname` is null, not defined)
- **Application**: Application name to send the processed files:

  `db-metadata-agent-application`
- **Metadata list**: Metadata list that the application should extract: `dicom_smoker`, `dicom_email`, and `dicom_blood_group`
- **Schedule**: To run a policy now, daily, weekly, or monthly

After the policy is created, it automatically runs according to the schedule and it can also be started manually by selecting the policy and clicking **Start** (see Figure 4-14).



Policy	Type	Schedule (UTC)	Status	Progress	Collections	Last Modified by	Last Modified
get_dicom_metadata	CONTENTSEARCH	Done	active stopped	0% 0 failed out of 0		sdadmin	2020-07-15T11:28:34.000Z

Figure 4-14 Running the policy

The Progress field displays the number of processed files. A policy can also be created, edited, deleted, or started through the REST API.

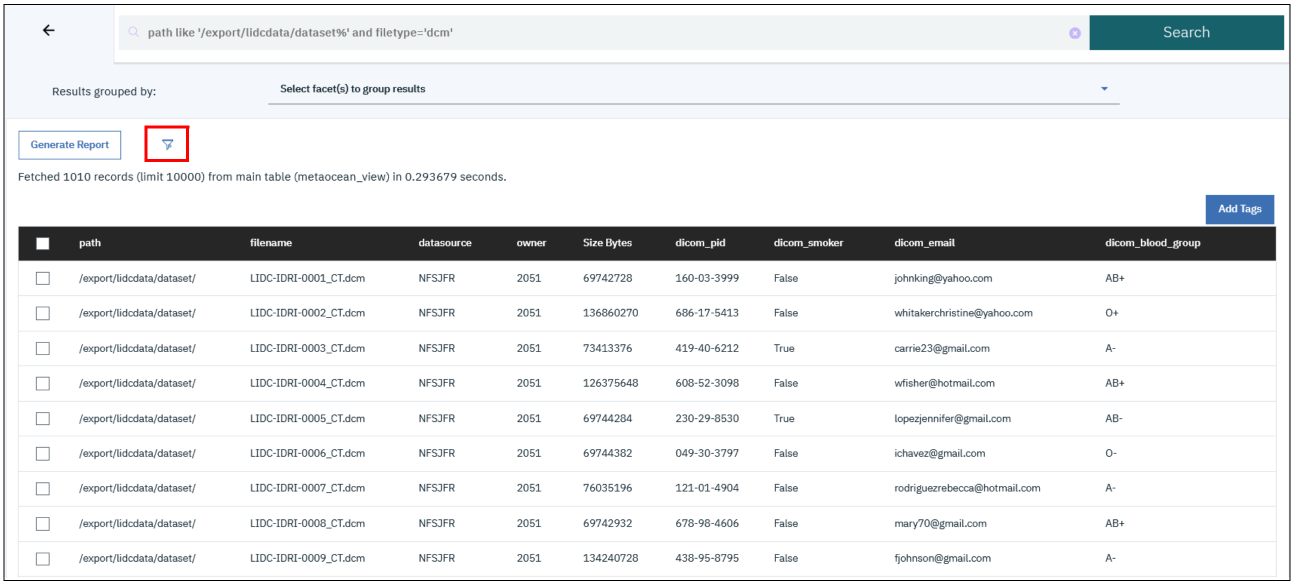
Viewing the cataloged metadata

After the policy completion, we can start browsing the metadata that must be gathered by IBM Spectrum Discover.

From the **Search** page, we can, for example, run a query, such as:

`path like '/export/lidcdata/dataset%'` and `filetype='dcm'`

More columns can be added by clicking the **funnel** icon and displaying the relevant DICOM metadata (see Figure 4-15).


←
path like '/export/lidcdata/dataset%' and filetype='dcm'
Search
Results grouped by:
Select facet(s) to group results
Generate Report
Add Tags

Fetched 1010 records (limit 10000) from main table (metaocean_view) in 0.293679 seconds.

☐	path	filename	datasource	owner	Size Bytes	dicom_pid	dicom_smoker	dicom_email	dicom_blood_group
☐	/export/lidcdata/dataset/	LIDC-IDRI-0001_CT.dcm	NFSJFR	2051	69742728	160-03-3999	False	johnking@yahoo.com	AB+
☐	/export/lidcdata/dataset/	LIDC-IDRI-0002_CT.dcm	NFSJFR	2051	136860270	686-17-5413	False	whitakerchristine@yahoo.com	O+
☐	/export/lidcdata/dataset/	LIDC-IDRI-0003_CT.dcm	NFSJFR	2051	73413376	419-40-6212	True	carrie23@gmail.com	A-
☐	/export/lidcdata/dataset/	LIDC-IDRI-0004_CT.dcm	NFSJFR	2051	126375648	608-52-3098	False	wfisher@hotmail.com	AB+
☐	/export/lidcdata/dataset/	LIDC-IDRI-0005_CT.dcm	NFSJFR	2051	69744284	230-29-8530	True	lopezjennifer@gmail.com	AB-
☐	/export/lidcdata/dataset/	LIDC-IDRI-0006_CT.dcm	NFSJFR	2051	69744382	049-30-3797	False	ichavez@gmail.com	O-
☐	/export/lidcdata/dataset/	LIDC-IDRI-0007_CT.dcm	NFSJFR	2051	76035196	121-01-4904	False	rodriguezrebecca@hotmail.com	A-
☐	/export/lidcdata/dataset/	LIDC-IDRI-0008_CT.dcm	NFSJFR	2051	69742932	678-98-4606	False	mary70@gmail.com	AB+
☐	/export/lidcdata/dataset/	LIDC-IDRI-0009_CT.dcm	NFSJFR	2051	134240728	438-95-8795	False	fjohnson@gmail.com	A-

Figure 4-15 View the cataloged metadata

Data set insights

The previous workflow of adding metadata is useful only to an AI pipeline if we then make use of the cataloged metadata. In that case, we extract them to get insights on the data set by plotting statistics, detecting and curating outliers, and making sure that the data set is balanced and includes representative population.

We randomly generated data that we inserted. Therefore, graphs of this chapter have no value other than demonstrating how to visualize data from IBM Spectrum Discover.

Metadata query

We use the IBM Spectrum Discover API to query the list of DICOM files in our data set. The Python code to retrieve data is available in the GitHub repository.

From a high-level perspective, you must first retrieve a token and then send a POST request to `https://discover-ip/db2whrest/v1/search` with the JSON content that is shown in Example 4-6 on page 87.

Example 4-6 JSON content

```
query = {
“query”: “path like '/export/lidcdata/dataset%'”,
“filters”: [],
“group_by”: [],
“sort_by”: [],
   “limit”: 10000
}
```

The content includes the following parameters:

- `query`: The search query string (SQL-like syntax (for more information, see this web page
- `filters`: Filters are applied to the results of the query
- `group_by`: Fields to group by values to summarize results
- `sort_by`: Fields on which to sort results
- `limit` (optional): Maximum number or rows returned

If the request passed successful, a JSON response is returned. With this mechanism, IBM Spectrum Discover data retrieval can be embedded in any script or application, which is greatly simplifying communication and the use of data that is stored in it.

To demonstrate the use of IBM Spectrum Discover API to query the DCOM files in our dataset, we can use these data to explore and improve the data set quality that is indexed in IBM Spectrum Discover.

Data set improvement

Performance and accuracy of a deep learning model mainly depends on the quality of the underlying data set on which it was trained.

One common workflow in AI is to explore the data set, plot different data to establish trends, and information about its content:

- Detect outliers

  Find data with an aberrant value; for example, a negative value in the age column. When found, they can be removed or their value fixed.

- Make sure that the data set is balanced

  Verify that classes of the data set are representative of the population; for example. make sure that the age distribution matches the one of people having lung nodules.

> **Note:** One issue might be that if your data set is exclusively composed of elders, the model might *overfit*, which means learning only to predict scans of elder’s lungs and have a poor accuracy on a younger population. To avoid this issue, data scientists ensure that all classes are represented (and not over-represented) in their data set.

Possibilities and best practices for data set cleansing are highly dependent on the data set that is used and the problem tackled. In our simple example, we extract the age, sex, and smoking habit of the patient to plot several graphs, which were generated by using the Seaborn library (the code for the generation is available at this web page).

Figure 4-16 shows histogram of age per sex.

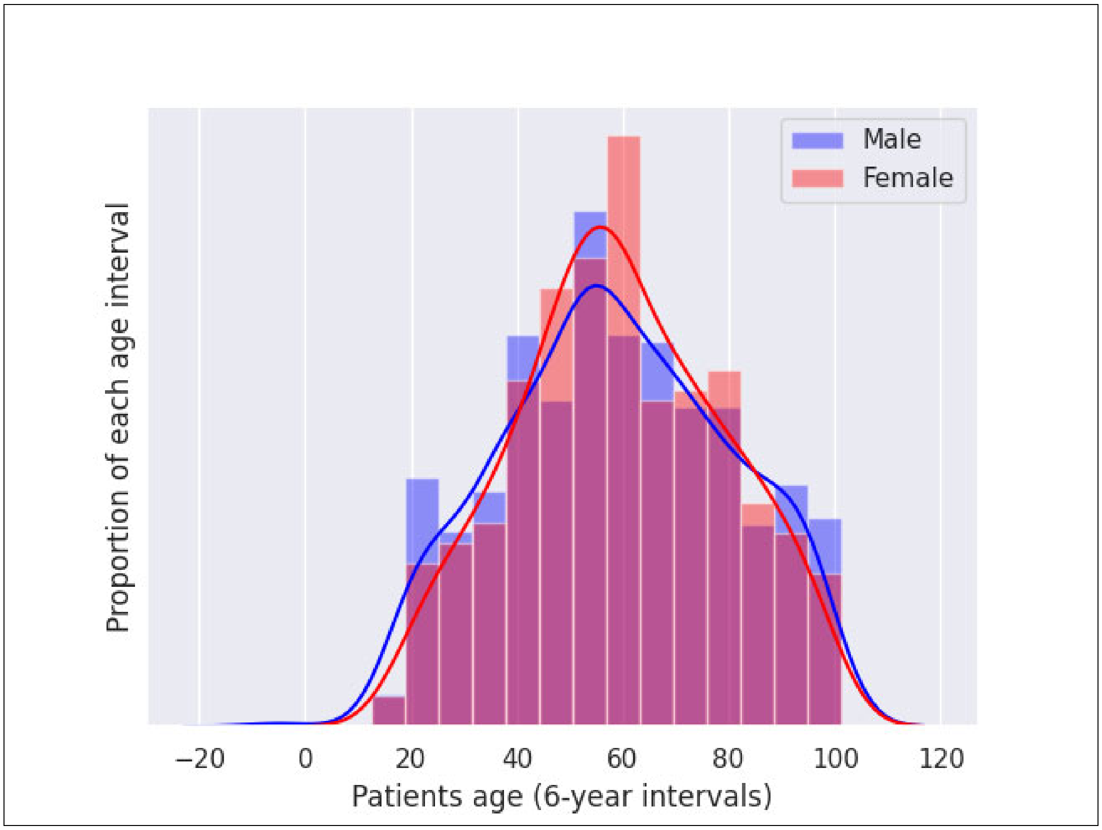


Figure 4-16 Histogram of age per sex

In the graph that is shown in Figure 4-16, we plot the histogram of ages per sex (blue for men, red for women). We see that the curves seem balanced and realistic (again, those are generated data made to demonstrate the use case, with no scientific value).

Figure 4-17 shows the patient count per sex and per smoking habit. Proportions again seem realistic.

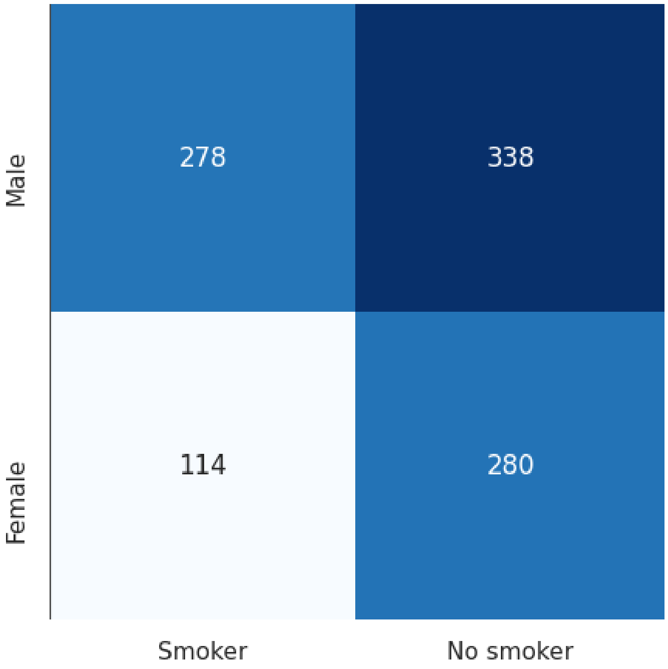


Figure 4-17 Smokers count per sex

Figure 4-18 shows the third graph in which we draw box plots of the distribution of ages per sex. We acknowledge that even if median and quartiles look consistent, the minimum age value for men is below 0.

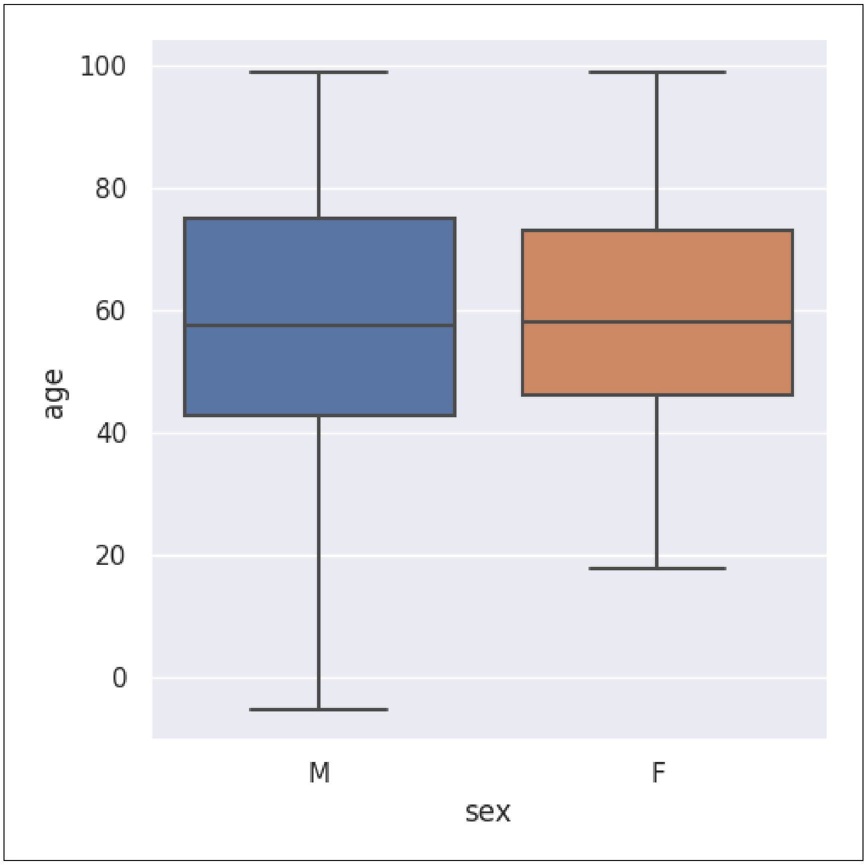
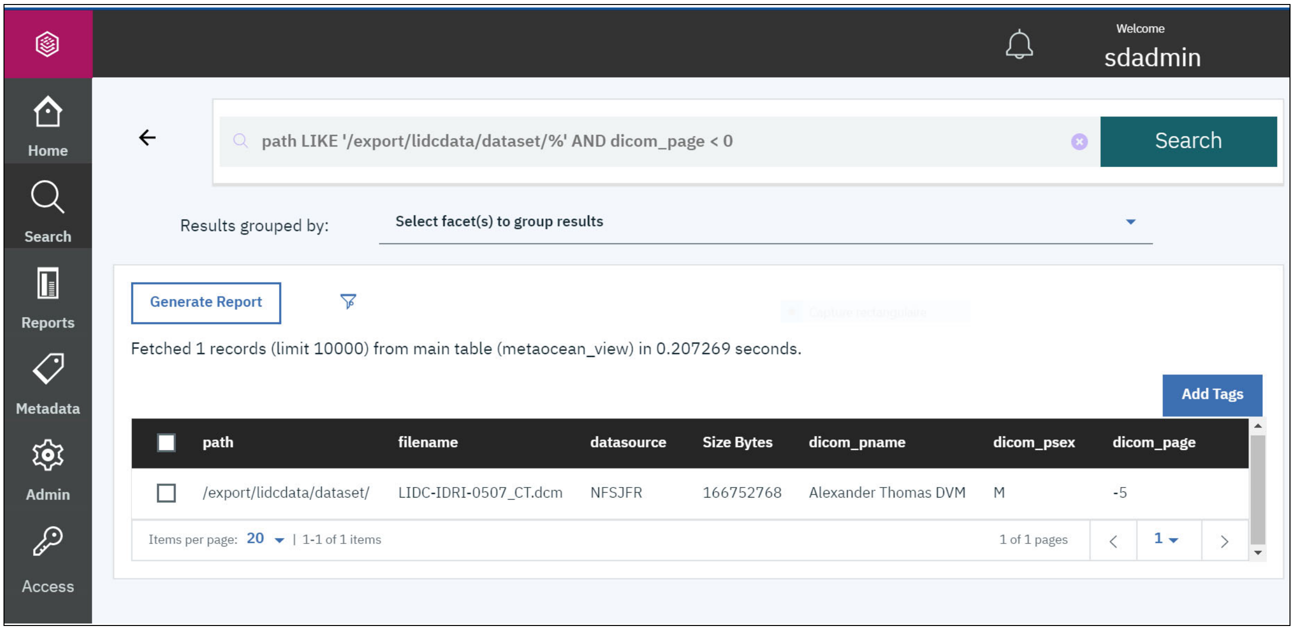


Figure 4-18 Box plots of the distribution of ages per sex

We detect here an outlier, and we can run the following query in IBM Spectrum Discover to extract patients with a negative age:

```
path LIKE '/export/lidcdata/data set/%' AND dicom_page < 0
```

We indeed detect one patient with a negative age; we can then handle that outlier and redraw the graphs again (see Figure 4-19).



path LIKE '/export/lidcdata/dataset/%' AND dicom_page < 0

Results grouped by: Select facet(s) to group results

Fetched 1 records (limit 10000) from main table (metaocean_view) in 0.207269 seconds.

■	path	filename	datasource	Size Bytes	dicom_pname	dicom_psex	dicom_page
□	/export/lidcdata/dataset/	LIDC-IDRI-0507_CT.dcm	NFSJFR	166752768	Alexander Thomas DVM	M	-5

Items per page: 20 | 1-1 of 1 items   1 of 1 pages   1

Figure 4-19   Patient with negative age

Thanks to IBM Spectrum Discover, we can also detect metadata with unexpected values. For example, we might ingest some incorrect “age” metadata from the DICOM file inspection, which are characters string instead of an integer.

Such aberrant values can be detected by using an IBM Spectrum Discover query, as shown in Example 4-7.

Example 4-7   IBM Spectrum Discover query

```
path like '/export/lidcdata/dataset%' and filetype='dcm' and LENGTH(RTRIM(TRANSLATE(dicom_page,
'*', ' 0123456789'))) <> 0
```

Note: TRANSLATE replaces all spaces with '*' and all digits with a white space. The second field of TRANSLATE is padded with white spaces to match the length of the third field (notice that it starts with a white space followed by the 10 digits), and characters of the third field are then replaced by the character of same index in the second field:

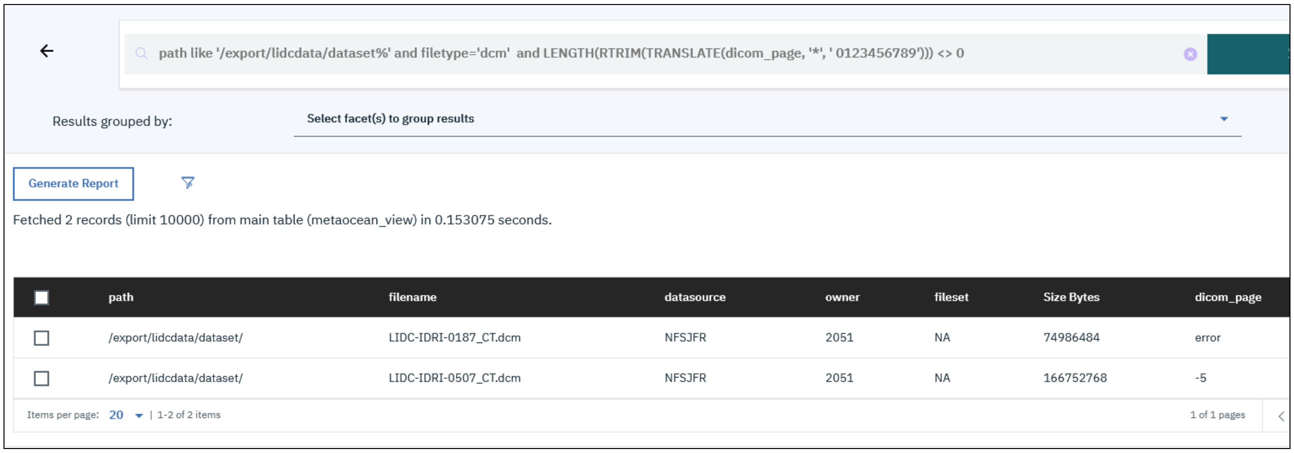
- '10' is translated to ' ' (two white spaces)
- '29' is translated '* ' (* plus two white spaces)
- 'az10' is translated to 'az ' (az plus two white spaces)
- '10 54' is translated to ' * ' (two white spaces, *, two white spaces)

In this way, the result is all white spaces only if the initial value was all digits.

RTRIM then removes white spaces from the right side (the result of RTRIM are empty if the result of TRANSLATE was all white spaces).

Finally, LENGTH returns the length of the RTRIM result, and the query filter on fields where that length is not zero, meaning that the initial value was not only made of digits.

Figure 4-20 shows the result of this query.



```
path like '/export/lidcdata/dataset%' and filetype='dcm'  and LENGTH(RTRIM(TRANSLATE(dicom_page, '*', ' 0123456789'))) <> 0
```

Results grouped by: Select facet(s) to group results

Fetched 2 records (limit 10000) from main table (metaocean_view) in 0.153075 seconds.

☐	path	filename	datasource	owner	fileset	Size Bytes	dicom_page
☐	/export/lidcdata/dataset/	LIDC-IDRI-0187_CT.dcm	NFSJFR	2051	NA	74986484	error
☐	/export/lidcdata/dataset/	LIDC-IDRI-0507_CT.dcm	NFSJFR	2051	NA	166752768	-5

Items per page: 20 | 1-2 of 2 items

1 of 1 pages

Figure 4-20 Result of the query

AI training

Another part of the use case is the ability to train AI models by using data that is provided by all radiology services.

This training is performed by AI teams on the same accelerated server or cluster that is used for inference or in a hospital if they want to train on their own servers.

In both cases, it is required to get a local copy of all labeled data in the data set to make them accessible from the training process.

Therefore, we use the data copy feature that is provided by Moonwalk, which is available in the next release of IBM Spectrum Discover that is targeted for 4Q 2020 as of this writing. Because Moonwalk in IBM Spectrum Discover is not released yet, we did not implement that feature; therefore, this area is for future enhancement for this use case. We are planning to implement it in the next edition of this publication.

Management of labeled data

Data that is uploaded by radiologists can be labeled or not (*labeled* means that a separate file exists that includes the annotation, which is a pixel mask of nodules drawn by the labeler). Because that annotation is required for deep learning training, it must be copied to the storage that is local to the training server (what we refer to as the *training storage* in this section).

However, we only need to copy data that includes an associated label, because all images cannot be labeled because of the time that is required to annotate an image. The solution depends on the *naming format*, followed by hospitals on their respective storage.

Case 1: Labeled data is in a separate directory

This case is the easiest because Moonwalk needs only to copy the directory that contains the labeled data from each hospital storage to the training storage.

> **Note**: Because the data source connection name is required when configuring a Moonwalk (move or copy) policy, we must create one policy per data source (per hospital then). Therefore, hospitals do not necessarily need to agree on a naming convention for the labeled directory.

Case 2: Labeled and unlabeled data is not separated

In this use case, we would need a filter to copy only the images for which there is an associated label. It is not possible at the time to do such complex queries for data copy, so we need a custom agent that tags the labeled data.

Assuming that hospitals follow a naming format (for example, `image_0267.dcm` and `image_0267_label.dcm`), that application queries the list of all files that are with “_label.dcm”, removes that extension, and tags all images and their corresponding labels with a tag `is_labeled` that is set to `True`. A data movement policy, based on Moonwalk, can then filter on this `is_labeled` tag to perform its data copy.

Inference service

Finally, we want to deploy a trained deep learning model to make it available for inference. Then, available radiologists can submit new scans and get results to assist and fasten their diagnosis (or even infer scans that were performed for another pathology).

In the following sections, we explore the deployment options of the inference service and share our approach’s results.

Deployment options

We identify two main cases of deployment:

- The inference is running on a remote server

  The first option is that the inference service runs as a stand-alone application on an external server. The service can be called in two ways:

  - Data moving option (first solution)

    IBM Spectrum Discover moves the data to be inferred on the server’s own storage and provides the inference script the file path as a parameter.

    The main drawback of this solution is the implementation complexity, because it needs to move the data to the inference storage, trigger the inference through an application, and move back the image and its prediction to the initial storage. It also requires a way to store the origin source of the image.

  - API front-end option (second solution)

    Inference is still deployed independently from IBM Spectrum Discover; however, in that case, we have an API front-end on the inference server with the ability to receive an image, process it, and return the result. An IBM Spectrum Discover agent, when triggered, retrieves the file from its original data source and calls the API with the full image attached in the request.

    This solution is simpler to implement but still requires developing the API interface. In addition, data is transferred twice: when it is read by the application and when it is sent through the API.

- The inference is running on IBM Spectrum Discover nodes (third solution)

  In this use case, we deploy the trained model as an IBM Spectrum Discover application. The policy triggers that application, which directly runs the inference and returns the result. Because the inference is running as a Kubernetes deployment on the IBM Spectrum Discover nodes in this case, it can use the IBM Spectrum Discover SDK to access the data sources.

However, because inference runs on GPU-enabled servers for performance, this solution requires that IBM Spectrum Discover is deployed on a Kubernetes cluster, which contains such nodes to specifically schedule the inference application to be deployed on these nodes.

The drawback is that it requires embedding the inference service inside a Kubernetes deployment. Therefore, the inference is no longer a separated service. By using the existing samples of custom applications (see this web page), this solution is simpler.

Note: The third solution in which the inference is deployed as an IBM Spectrum Discover application is the most relevant to us because it minimizes data transfers and is the most integrated solution.

However, it requires IBM Spectrum Discover to be deployed on a Kubernetes cluster, a feature, which is targeted to be available in a future release of IBM Spectrum Discover. Therefore, we decided to implement the second solution with an API front end on which data to infer can be received.

We describe how we implemented it and how the IBM Spectrum Discover application queries this API.

We developed a simple API front end that uses Flask, which is a micro web framework that is written in Python that can receive a file, start the inference on that file, and return a JSON result that contains the prediction output.

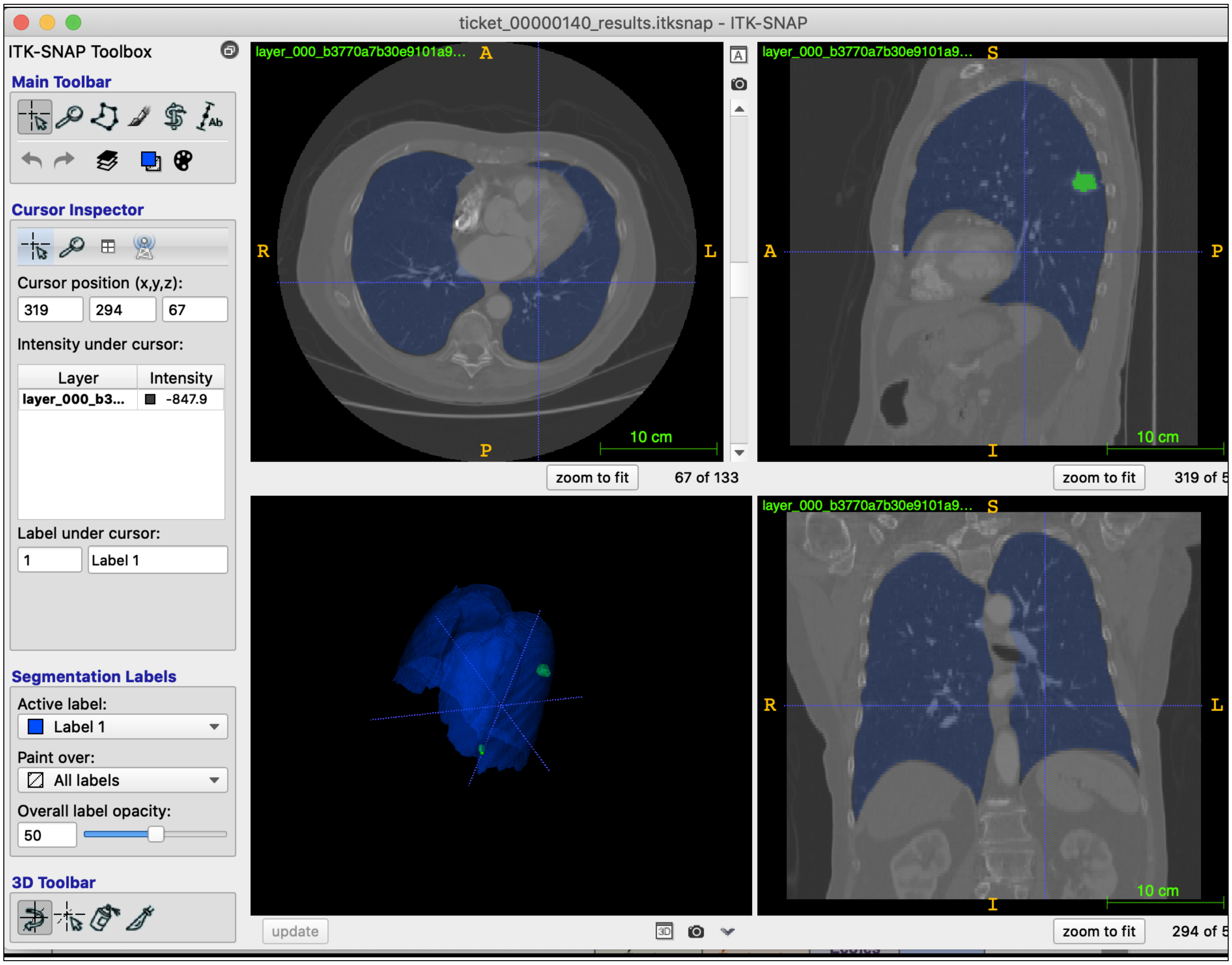
The code is available at this web page.

The prediction returns the following fields:

- `filename_seg`: Path of the segmentation file.
- `model_version`: An ID that references which model was used for inference.
- `obj_count`: The number of objects (in our example, nodules) that are detected by the model.
- `result`: List of bounding boxes that are detected (for each box, coordinates and specific metrics, such as confidence or overlaps).

The `filename_seg` field represents the path of the inference output. This output is a mask of the segmentation of nodules (as shown in green in Figure 4-21), which means that an image of the same size of the input with pixels of nodules highlighted. The use of pixels is more precise than bounding boxes because it shows exactly where nodules are at a pixel level.

In the example that is shown in Figure 4-21, we also applied a lungs segmentation model that highlights lungs (in blue).


ticket_00000140_results.itksnap - ITK-SNAP

ITK-SNAP Toolbox

Cursor position (x,y,z): 319 294 67

Intensity under cursor:

Layer	Intensity
layer_000_b3...	■ -847.9

Label under cursor: 1 Label 1

Active label: Label 1

Paint over: All labels

Overall label opacity: 50

Figure 4-21 Lungs segmentation model

4.5.4 Inference

In this section, we explain how we use the deployed inference service. We describe the following identified options:

- On-demand inference

  An example of this model is when a medical image is sent to the model by a radiologist who needs diagnosis insights or by an automated system that infers any new scan for potential lung nodules, and results are expected in real-time or near real-time fashion (order of a few minutes).

- New model release

  When a better detection model replaces the previous model, a system triggers a new inference on all scans that were generated to verify that the previous model version did not miss nodules. In that case, a larger number of inferences are triggered (and possibly handled in batches), and the results are not expected in real-time.

On-demand inference

This inference use case is in two parts: how to get notifications to automatically index new files in IBM Spectrum Discover, and how the data flow works to provide the radiologist real-time results of detections.

Notifications issue

The first requirement is the ability for IBM Spectrum Discover to be notified when a new file appears on the storage to first index it and to run its inference policy. Notifying IBM Spectrum Discover when a new file appears on the storage requires running full storage scans at repeated intervals, which cannot be scheduled more than daily and therefore users cannot receive inference results in real-time.

As of this writing, notifications are supported on IBM Spectrum Scale and IBM Cloud Object Storage only. It means that whenever a file is created (but it also supports all notifications available in Spectrum Scale: access, modification and so forth), a notification is sent to IBM Spectrum Discover, which is then consumed to be indexed.

> **Note**: IBM Spectrum Discover uses Kafka to communicate with external applications, such as file scanners and custom applications. When a scan of a data source is performed, information about new files is written in Kafka topics.
>
> IBM Spectrum Discover runs consumer processes, one per storage type: Scale consumer, Cloud Object Storage consumer, and File consumer for both NFS and CIFS. Accordingly, these consumers read from these Kafka topics and update the Discover database.
>
> The notification ability of Spectrum Scale and Cloud Object Storage means that this software can write on these Kafka topics when a file is created without being scanned.

In our case, we also want to be notified when a new file is created on other types of storage. Theoretically, it is possible to implement that feature by using a custom system that is based on *inotify*; for example, Linux kernel feature for receiving notifications about the file system (for more information, see this web page).

In this use case, we assume that we have a working notifications system, which ensures that all files are indexed in IBM Spectrum Discover when they are created on the storage.

Data flow

We assume that the notification issue is sorted out: each new file is automatically indexed by IBM Spectrum Discover. The inference workflow is started by the radiologist that selects **Detect nodules** in the PACS GUI that then communicates through API calls.

Following that trigger mechanism, the radiologist expects two outputs:

- The number of nodules that are detected by the model.
- A segmentation file that contains a mask of detected nodules, which is available on its local storage for the radiologist to visualize it, if needed.

We also assume that we have a working API front end for the inference (as described in the deployment options) to which we can send files to infer.

Retrieving the output segmentation file increases complexity of that workflow. We considered the following potential solutions:

- Solution 1

  This solution included the following steps (see Figure 4-22):

  a. The API button of the PACS creates an inference policy to trigger inference on the concerned file, which was indexed thanks to the notification system.

  b. The custom IBM Spectrum Discover application loads the file by using the SDK and sends it to the API.

  c. The inference is performed, and the number of nodules is sent back and stored by the application as a metadata.

  d. Because the output segmentation is generated on the local storage of the inference, the PACS can query directly the API to retrieve that file. Querying the API directly implies that another endpoint was developed for that API to handle sending files.

  e. The PACS can query IBM Spectrum Discover to get the metadata (number of nodules, for example).

The main issue here is that the IBM Spectrum Discover application likely has read-only access to storage (the same rights that were given to IBM Spectrum Discover) and therefore cannot retrieve the segmentation file and store it.

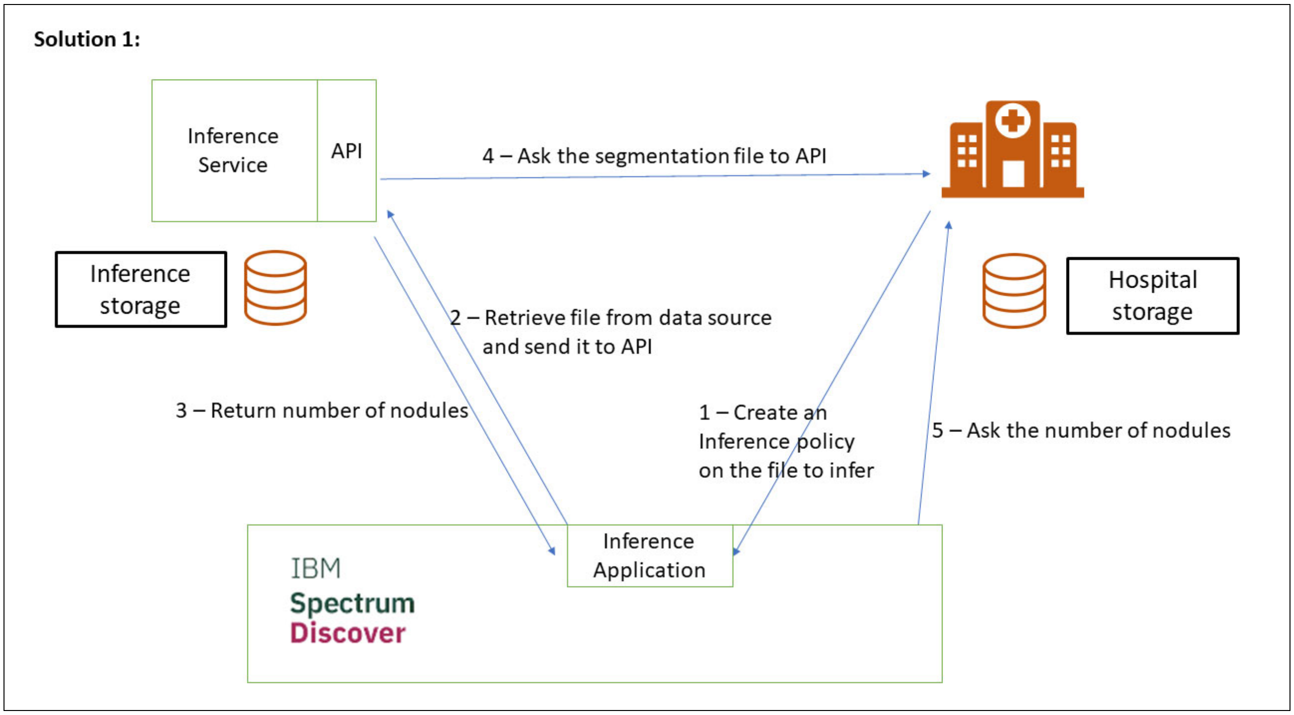

Solution 1:
Inference Service
API
4 – Ask the segmentation file to API
Inference storage
Hospital storage
2 – Retrieve file from data source and send it to API
3 – Return number of nodules
1 – Create an Inference policy on the file to infer
5 – Ask the number of nodules
Inference Application
IBM
Spectrum
Discover

Figure 4-22 Solution 1

- Solution 2

  This solution (see Figure 4-23 on page 98) covers the case in which data moving policies are available. In that case, the flow is similar to Solution 1, but the IBM Spectrum Discover application can, in addition to sending the file to inference, create a data moving policy to move the output segmentation file from the inference local storage to the hospitals storage.

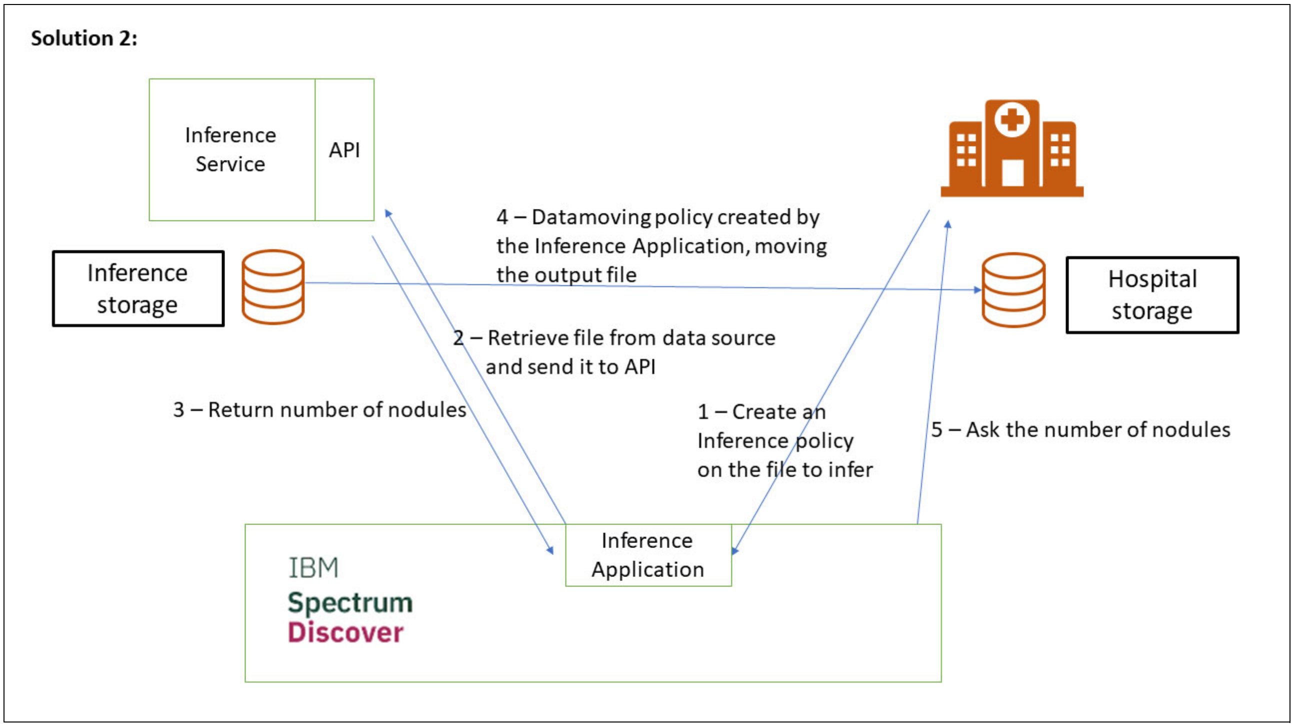

Solution 2:
Inference Service
API
Inference storage
Hospital storage
4 – Datamoving policy created by the Inference Application, moving the output file
2 – Retrieve file from data source and send it to API
3 – Return number of nodules
1 – Create an Inference policy on the file to infer
5 – Ask the number of nodules
IBM Spectrum Discover
Inference Application

Figure 4-23 Solution 2

- Solution 3

  This solution, as shown in Figure 4-24, is provided for comparison. In that case, the PACS can directly use the inference API. IBM Spectrum Discover is then queried by the inference service to store its results as metadata, only acting as a type of archive to keep track of inferences.

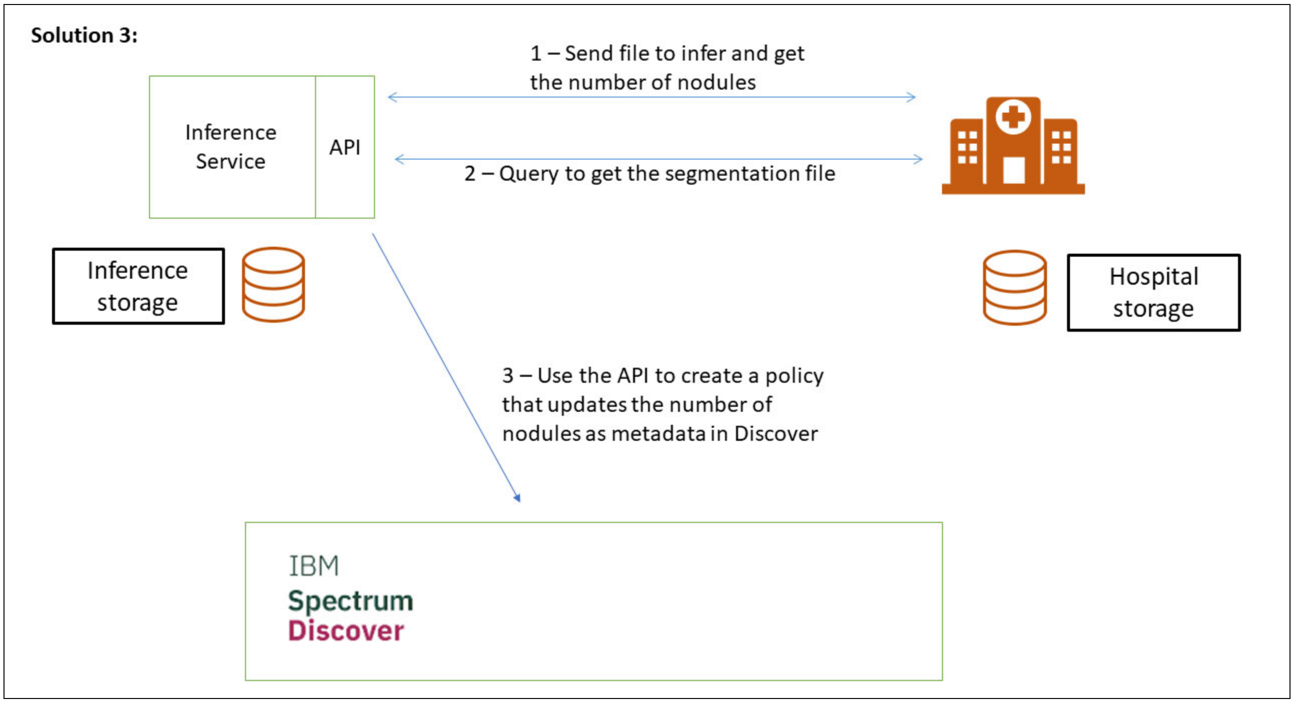

Solution 3:
Inference Service
API
1 – Send file to infer and get the number of nodules
2 – Query to get the segmentation file
Inference storage
Hospital storage
3 – Use the API to create a policy that updates the number of nodules as metadata in Discover
IBM Spectrum Discover

Figure 4-24 Solution 3

4.5.5 New model release

When a new model is released, we can define a DEEPINSPECT policy that selects all of the ingested DICOM files and sends them through an IBM Spectrum Discover application to the new inference model.

The results of this new release are cataloged in IBM Spectrum Discover as new tags, which allows us to perform queries that compare the previous results to the new results.

For example, we can issue a request that compares the amount of nodules that are detected by the former model and the new one, as shown in Example 4-8.

Example 4-8 IBM Spectrum Discover query comparing the number of nodules detected

```
filetype='dcm' and inf_nodules_count>newinf_nodules_count
```

This information can help us to evaluate the performance of the new model.

For example, is it normal that, for a set of files, the new inference model finds less nodules than the first one? Is it normal that the new model finds 25 nodules when the previous one was finding only two nodules?

The code of this agent is available in the GitHub repository.

4.6 Online resources

Table 4-5 lists the online resources that are available for this use case.

Table 4-5 Online resources

Online resource	Location
GitHub repository of the chapter	https://github.com/IBMRedbooks/SG248448-Making-Data-Smarter-with-Spectrum-Discover-Practical-AI-Solutions/tree/master/chapter3
Storage for AI & IBM Power Systems Video: Medical data challenge white paper	https://www.ibm.com/it-infrastructure/services/client-experience-portal/offeringdetail.jsp?offid=2048&sourcenamekey=CATALOG_LOGICAL
Area Under Curve (AUC)	https://developers.google.com/machine-learning/crash-course/classification/roc-and-auc
Supported medical image formats	https://loli.github.io/medpy/information/imageformats.html
3D Retina U-Net model	https://arxiv.org/abs/1811.08661
Medical Detection Toolkit	https://github.com/MIC-DKFZ/medicaldetectiontoolkit
Watson Machine Learning Community Edition toolkit	https://developer.ibm.com/linuxonpower/deep-learning-powerai/releases/
Pydicom documentation to add custom DICOM metadata	https://pydicom.github.io/pydicom/stable/auto_examples/metadata_processing/plot_add_dict_entries.html#sphx-glr-auto-examples-metadata-processing-plot-add-dict
Accessing Spectrum Scale through NFS	https://www.ibm.com/support/knowledgecenter/STXKQY_5.0.2/com.ibm.spectrum.scale.v5r02.doc/bl1ins_nfsexportoverview.htm

Online resource	Location
GitHub repository of the chapter	https://github.com/IBMRedbooks/SG248448-Making-Data-Smarter-with-Spectrum-Discover-Practical-AI-Solutions/tree/master/chapter3
Connecting to data sources with IBM Spectrum Discover	https://www.ibm.com/support/knowledgecenter/SSY8AC_2.0.3/com.ibm.spectrum.discover.v2r03.doc/ins_configuredatasourceconnections.html
DICOM custom metadata	https://pydicom.github.io/pydicom/stable/auto_examples/metadata_processing/plot_add_dict_entries.html#sphx-glr-auto-examples-metadata-processing-plot-add-dict-entries-py
Standard DICOM metadata	https://www.dicomlibrary.com/dicom/dicom-tags/
IBM Spectrum Discover Application SDK	https://github.com/IBM/Spectrum_Discover_Application_SDK

4.7 Summary

Table 4-6 summarizes the use of artificial intelligence in our medical imaging use case.

Table 4-6 Summary of the using artificial intelligence in medical imaging use case

Use case overview	A consortium of radiology services team to share resources (medical images) and use AI to help diagnosis of lung nodules. There are four scenarios in this use case: ► They index their local data sources in a single Spectrum Discover application to create one single data set of 3D CT-Scans of lungs in DICOM format. ► They catalog metadata, explore and clean the data set. ► Data scientists train deep learning models using data from all hospitals. ► Trained model can be deployed and queried by radiologists to get inference results. It can also automatically analyze images even if the patient doesn't come to the hospital for lung nodules.
Products involved	► IBM Spectrum Discover ► IBM Spectrum Scale, IBM Cloud Object Storage or any other data source ► IBM Watson Machine Learning Community Edition
Benefits	► Unify multiple data sources in one single IBM Spectrum Discover instance. ► Assist creation and cleansing of a data set and extraction of data insights to prepare a deep learning training. ► Help copying or moving data to the appropriate storage for training. ► Assist the deployment of deep learning models and keep track of the inferences performed.
High-level implementation steps	1. Deploy IBM Spectrum Discover and index multiple data sources to it. 2. Use CONTENTSEARCH to extract metadata from DICOM files; enrich these metadata with a DEEPINSPECT application that connects to a database 3. Generate graphs of data, detect outliers, filter IBM Spectrum Discover data to cleanse them. 4. Perform copy of labeled data to the storage where AI training is performed. 5. Deploy an AI service (inside an IBM Spectrum Discover application or as an independent service with an API front end). 6. Query this AI service with new images to get inference results (in near real-time mode or on multiple images).

5

IBM Spectrum Discover integration with IBM Spectrum Archive Enterprise Edition

In this chapter, we provide an overview of the IBM Spectrum Discover integration with the IBM Spectrum Archive Enterprise Edition solution and its integration with IBM Spectrum Scale. This integration helps storage administrators and data stewards retain more data, enable a cold storage tier inside their environment, create an air gap capability for their data, and better match the value of the data to the type of media on which it is stored.

Based on this integration, we describe the Data and Storage Optimization and the Data Governance use cases in this chapter.

This chapter includes the following topics;

5.1 Use cases introduction and overview

As described in Chapter 1, “IBM Spectrum Discover overview” on page 1, various use cases are supported by IBM Spectrum Discover. Primary to this chapter and solution are the Data and Storage Optimization use case and the Data Governance use case to a slightly lesser extent. These use cases highlight IBM Spectrum Discover’s ability to integrate and work with two other members of the IBM Spectrum Storage family: IBM Spectrum Scale and IBM Spectrum Archive.

IBM Spectrum Scale is one of the primary data sources that IBM Spectrum Discover supports. It is an enterprise-grade parallel file system that provides superior resiliency, scalability, and control. Storage administrators can combine flash, disk, cloud, and tape storage options into a unified system with higher performance and lower cost than traditional approaches.

IBM Spectrum Scale also manages the placement of data across these different storage media by using its Information Lifecycle Management (ILM) policy engine. This engine facilitates the moment of files in the file system to different storage pools.

IBM Spectrum Archive Enterprise Edition (EE) uses the Linear Tape File System (LTFS) standard format for reading and writing to tape and integrates those tape assets into the IBM Spectrum Scale file system. Therefore, IBM Spectrum Scale can contain tape storage as an active archive tier inside the file system. This tier enables storage administrators to extend the capacity of the file system beyond the available storage on flash and disk and into tape libraries.

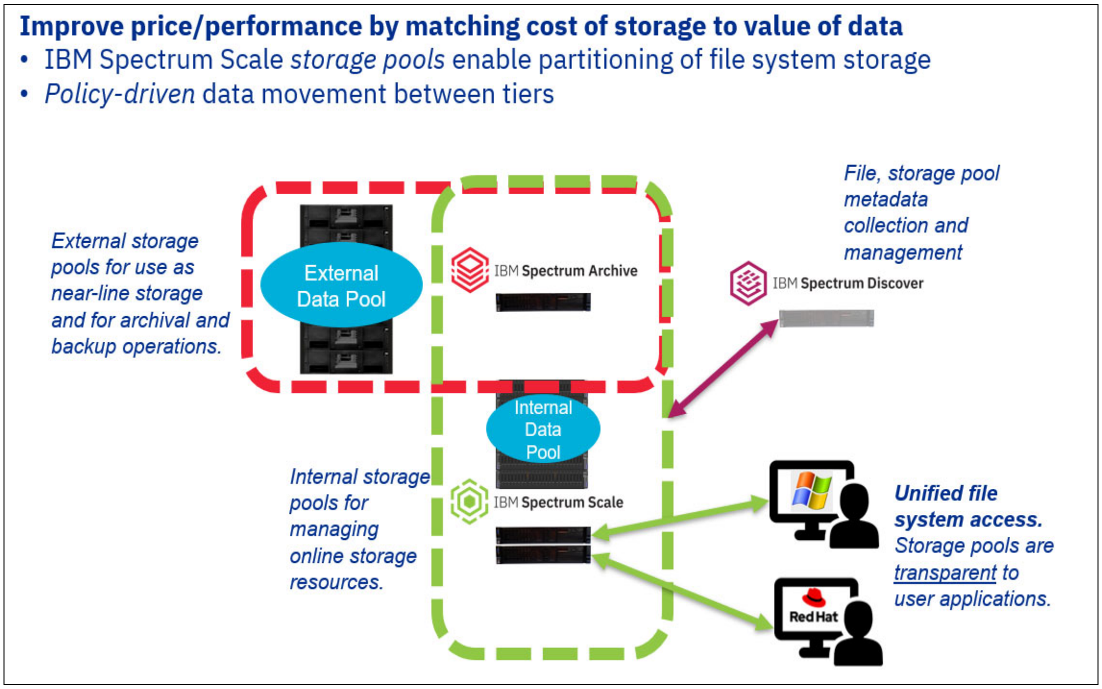
IBM Spectrum Discover supports IBM Spectrum Scale and IBM Spectrum Archive EE as data sources that it can see. Because of this feature, storage administrators can search across both systems by using IBM Spectrum Discover to see the data in active tiers and the cold tiers of the extended file system.

Beginning with release 2.0.3, IBM Spectrum Discover can also start the IBM Spectrum Scale policy engine to move files between the different tiers in the files system. IBM Spectrum Discover orchestrated the movement of data between IBM Spectrum Scale and IBM Spectrum Archive EE.

IBM Spectrum Discover features the following levels of integration with IBM Spectrum Archive:

- Viewing the state of data in IBM Spectrum Archive from IBM Spectrum Discover graphical user interface, generate reports, and include the data sets in IBM Spectrum Archive in the data export functions of IBM Spectrum Discover.
- The ability to move data between active storage medial pools, such as flash or disk, to and from tape pools by using the IBM Spectrum Scale Information Lifecycle Management (ILM) tool.

Figure 5-1 on page 103 shows the IBM Spectrum Discover integration with IBM Spectrum Archive EE and IBM Spectrum Scale.


Improve price/performance by matching cost of storage to value of data

- IBM Spectrum Scale *storage pools* enable partitioning of file system storage
- *Policy-driven* data movement between tiers

External storage pools for use as near-line storage and for archival and backup operations.

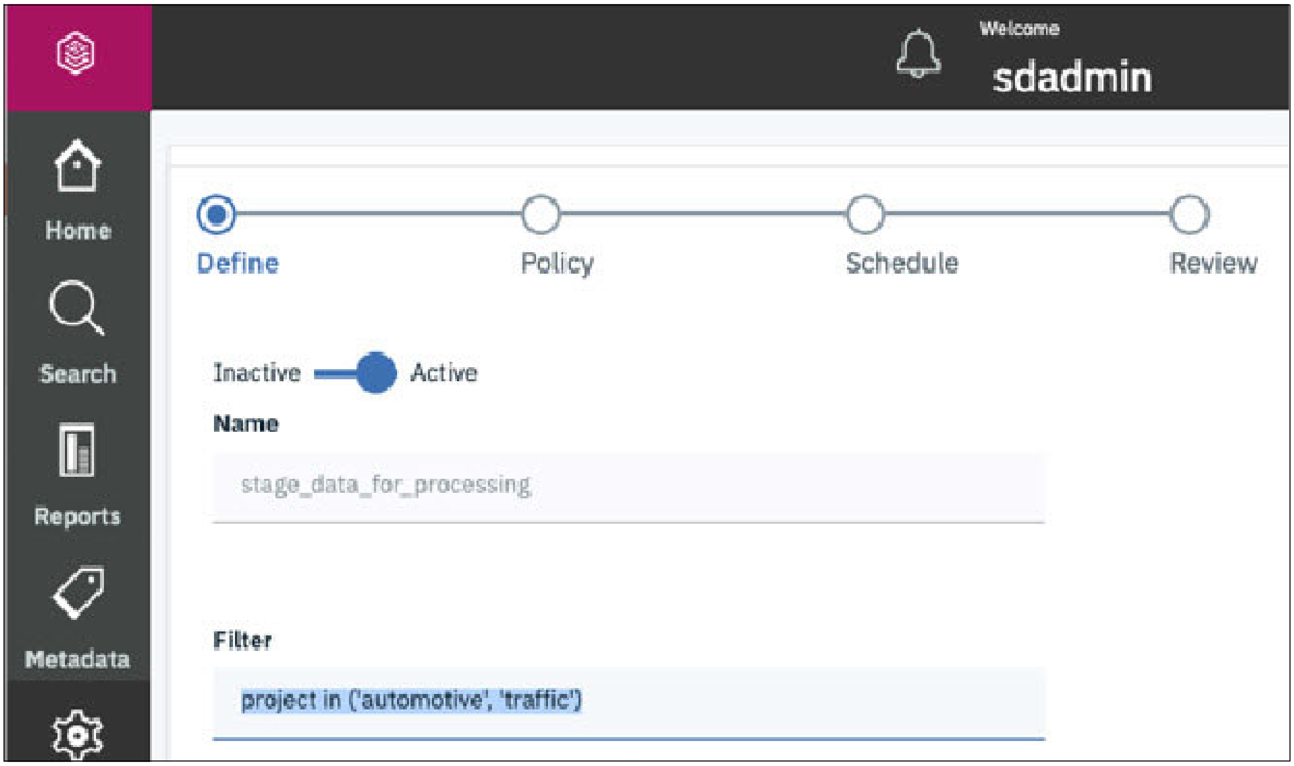
File, storage pool metadata collection and management

Internal storage pools for managing online storage resources.

Unified file system access. *Storage pools are* *transparent* *to user applications.*

Figure 5-1 IBM Spectrum Discover with IBM Spectrum Archive and IBM Spectrum Scale

Figure 5-2 shows policy-driven data movement as a function of system or user-defined metadata.



Inactive ● Active

Name

stage_data_for_processing

Filter

project in ('automotive', 'traffic')

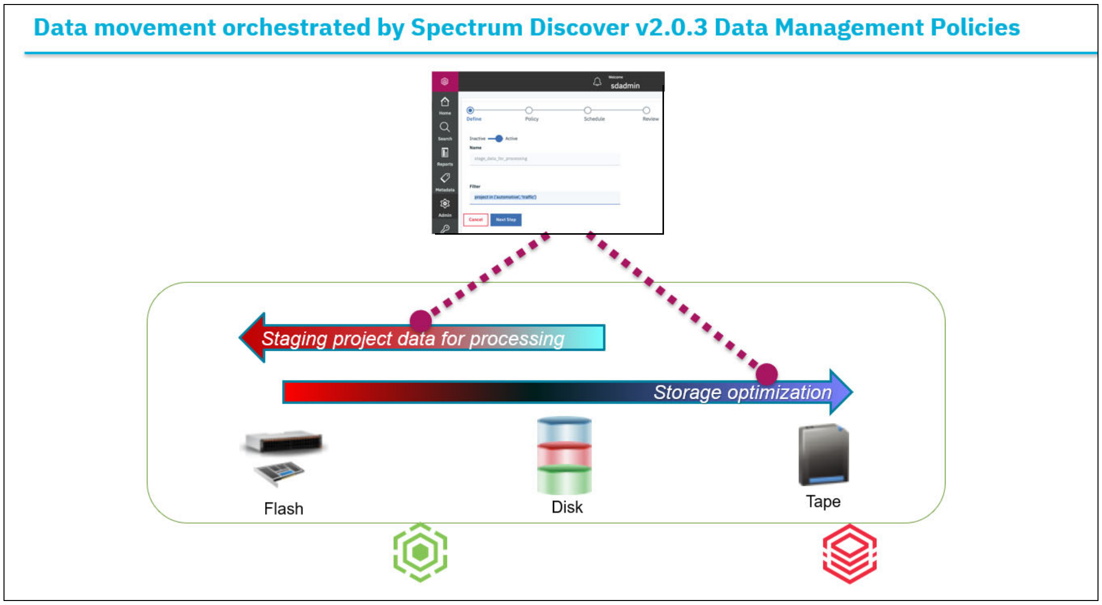
Figure 5-2 Policy-driven data movement

5.2 Benefits

The integration of IBM Spectrum Discover with IBM Spectrum Scale and IBM Spectrum Archive creates a series of potential benefits for storage administrators and data stewards, including the following examples:

- The ability to visualize the state of data across various storage media and pools, including data that was moved to tape.
- The ability to search metadata tags across storage pools that are on flash, disk, or in an active archive in one or more tape pools.
- The ability to decide about data that can be moved across pools and down to tape based on any of the criteria that is inherent in the file system, correlated against any of the metadata tags that were applied to data based on policies applied by using custom tags, or CONTENTSEARCH or DEEPINSPECT policies.
- Identify data sets across multiple pools that are deemed “cold” or inactive based on any number of business criteria and move that data to a tape pool.
- The ability to assemble data sets that exist across disk and tape platforms that are inside an IBM Spectrum Scale cluster.
- The ability to identify data sets and move them to or from tape media that is based on requirements.
- The ability to identify a data set to be protected and move it as a second copy to a tape pool, for data copy protection requirement, or an air gap copy for Cyber Resiliency use cases.

Figure 5-3 shows storage movement across various types of media inside IBM Spectrum Scale.



Data movement orchestrated by Spectrum Discover v2.0.3 Data Management Policies

Name
stage_data_for_processing

Filter
project in ('automotive','traffic')

Figure 5-3 Storage movement across various types of media inside IBM Spectrum Scale

5.3 Products involved

The following products are used in this use case:

- IBM Spectrum Scale

  IBM Spectrum Scale is an enterprise-grade parallel file system that provides superior resiliency, scalability, and control. It delivers scalable capacity and performance to handle demanding data analytics, content repositories, and technical computing workloads.

  Storage administrators can combine flash, disk, cloud, and tape storage into a unified system with higher performance and lower cost than traditional approaches. Supporting various protocols, including Portable Operating System Interface (POSIX), Hadoop Distributed File System (HDFS), NFS, SMB, and Swift/S3 it is delivered as a software-only solution, cloud deployment, or as part of an integrated system.

- IBM Spectrum Archive Enterprise Edition

  IBM Spectrum Archive EE gives organizations an easy way to use cost-effective IBM tape drives and libraries within a tiered storage infrastructure. By using tape libraries instead of disks for Tier 2 and Tier 3 data storage-data that is stored for long-term retention, organizations can improve efficiency and reduce costs that are related to storing growing amounts of data.

  Based on the Linear Tape File System (LTFS) standard it integrates with the scalability, manageability, and performance of IBM Spectrum Scale and provides direct, intuitive, and graphical access to data that is stored in Linear Tape-Open (LTO) Ultrium tape cartridges and IBM 3592 tape cartridges. It eliminates the need for more tape-management and data-access software.

- IBM Tape libraries and technology

  With a long history of research and innovation, IBM Storage includes various tape solutions, including modular and full-size libraries that can support petabytes of retained data. Included in the IBM Tape family are two different media options that are supported by IBM Spectrum Archive EE: Linear Tape-Open (LTO) Ultrium media, which follows the LTO standard, and IBM 3592 tape media, which enable media reuse by enabling the drive to reformat and upgrade older generation media cartridges.

> **Note:** For more information about these products, see Appendix A, “IBM Spectrum Scale, IBM Spectrum Archive, and IBM Tape libraries product details” on page 135.

5.4 IBM Spectrum Discover integration with IBM Spectrum Scale and IBM Spectrum Archive EE architecture

As described in 5.1, “Use cases introduction and overview” on page 102, IBM Spectrum Discover features the following levels of integration with IBM Spectrum Archive EE:

- Viewing detail information on migration status of data in IBM Spectrum Archive from IBM Spectrum Discover GUI to provide more finely grained data dashboard and reports.
- Moving data from IBM Spectrum Scale managed disk pools into IBM Spectrum Archive managed pools with IBM Spectrum Discover in a more effective way.

5.4.1 Data view of migration status with IBM Spectrum Discover

As part of the metadata of the files that are managed by IBM Spectrum Archive EE, the migration status is extracted by IBM Spectrum Discover through scanning or live events. This status consists of the following key columns when you perform metadata search:

- State

  This metadata is the most important metadata to identify the file in the internal storage pools or in the external tape pool, or both. This metadata can include the following values:

  – `migrtd` (Migrated): Indicates that the file was migrated to tape. It is in the external tape pool *only*.

  – `resdnt` (Resident): Indicates that the file is in the internal storage pool *only*.

  – `premig` (Premigrated): Indicates that the file is pre-migrated to tape, which means it is in the external tape pool *and* in the internal storage pool.

- migloc

  This information shows the location of the file on tape, and used the following format:

  ```
  1 tape cartridge label@tape storage pool id@tape library id
  ```

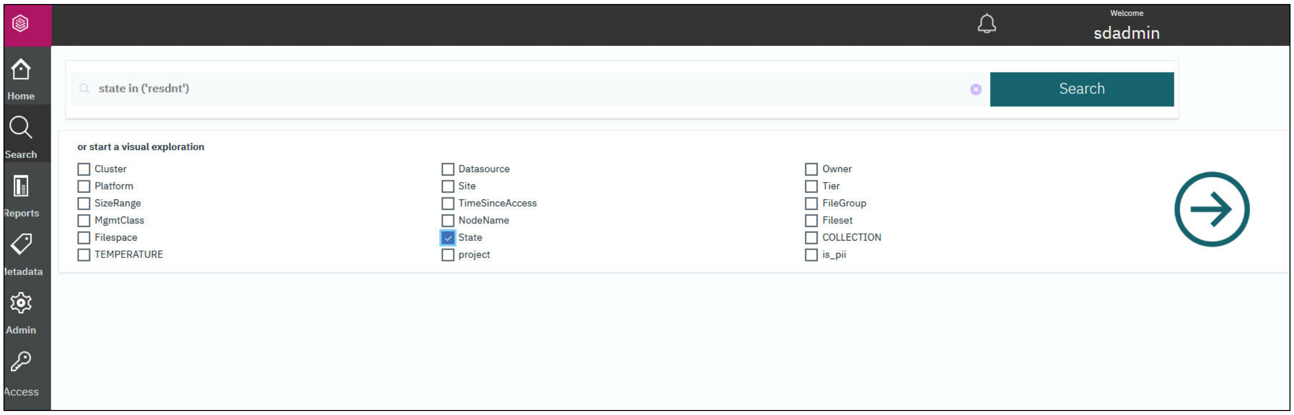
  Any other copies are separated by colons.

- Size Consumed Bytes

  This column in the search results shows the size of file in the associated IBM Spectrum Scale file system, which displays a zero if the file is moved to tape.

Complete the following steps to obtain the information in these columns:

1. Select the search icon in left pane of the IBM Spectrum Discover GUI, and then, select **State**, and click the **right-pointing arrow** to continue, as shown in Figure 5-4.



state in ('resdnt')

or start a visual exploration

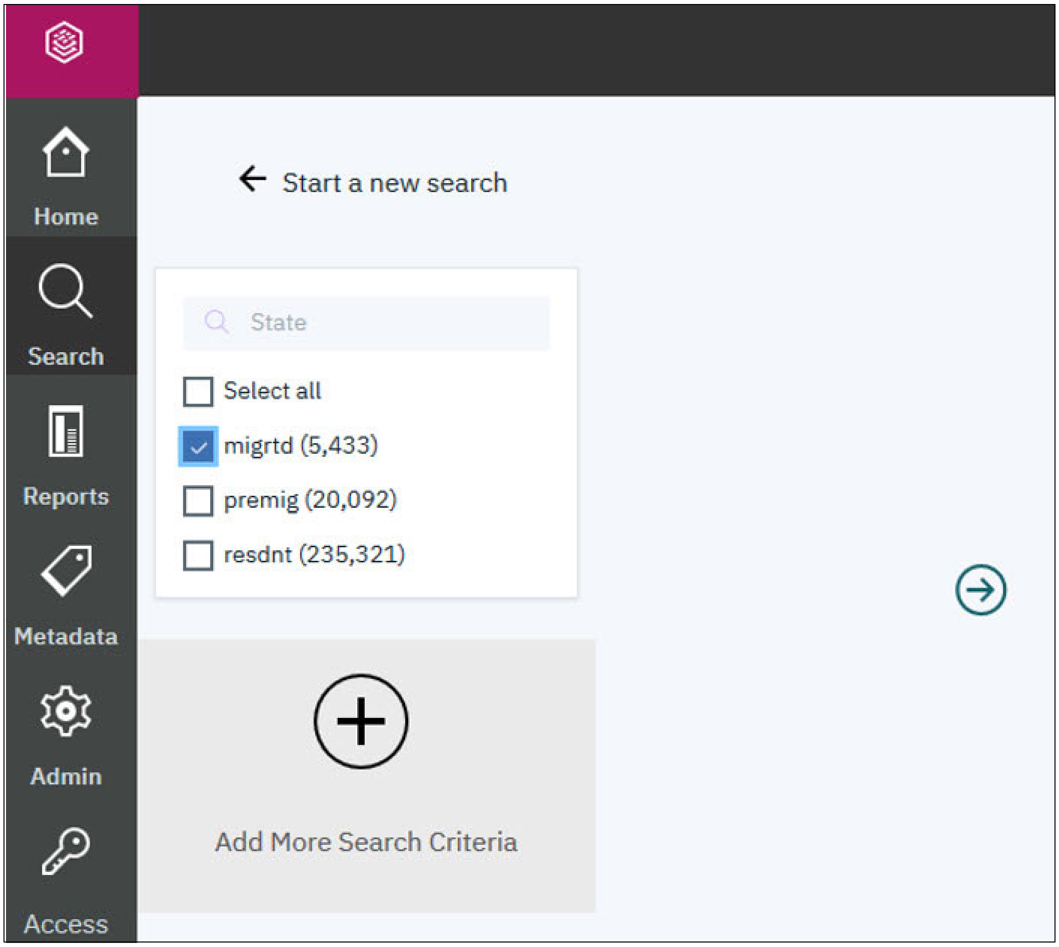
□ Cluster
□ Platform
□ SizeRange
□ MgmtClass
□ Filespace
□ TEMPERATURE

□ Datasource
□ Site
□ TimeSinceAccess
□ NodeName
☑ State
□ project

□ Owner
□ Tier
□ FileGroup
□ Fileset
□ COLLECTION
□ is_pii

Figure 5-4 Search for migration status metadata

2. The three types of State: `migrtd`, `premig`, and `resdnt` are available options. Select **migrtd**, for example, and click the **right-pointing arrow** to continue, as shown in Figure 5-5.



Start a new search

State

□ Select all

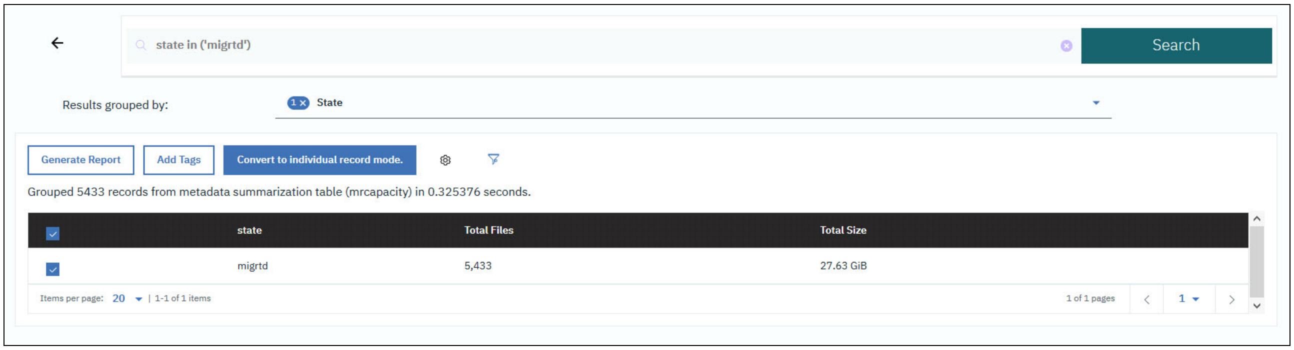
☑ migrtd (5,433)

□ premig (20,092)

□ resdnt (235,321)

Figure 5-5 Search for migrtd files

3. As shown in Figure 5-6, 5,433 files in `migrtd` state are found in this search. Select this file group and then, select **Convert to individual record mode**.



state in ('migrtd')

Results grouped by: 1 × State

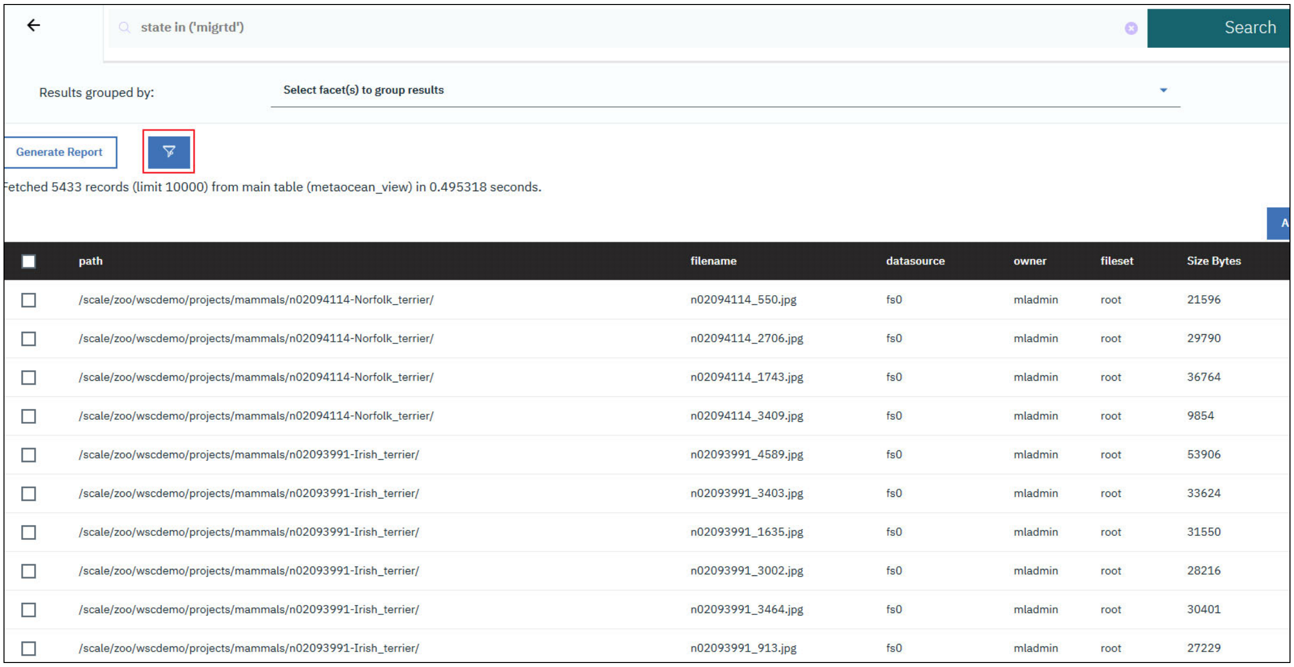
Grouped 5433 records from metadata summarization table (mrcapacity) in 0.325376 seconds.

☑	state	Total Files	Total Size
☑	migrtd	5,433	27.63 GiB

Items per page: 20 | 1-1 of 1 items   1 of 1 pages   1

Figure 5-6 Group view of the migrtd files

4. In the individual record mode, click the funnel to add more columns into the table, as shown in Figure 5-7.

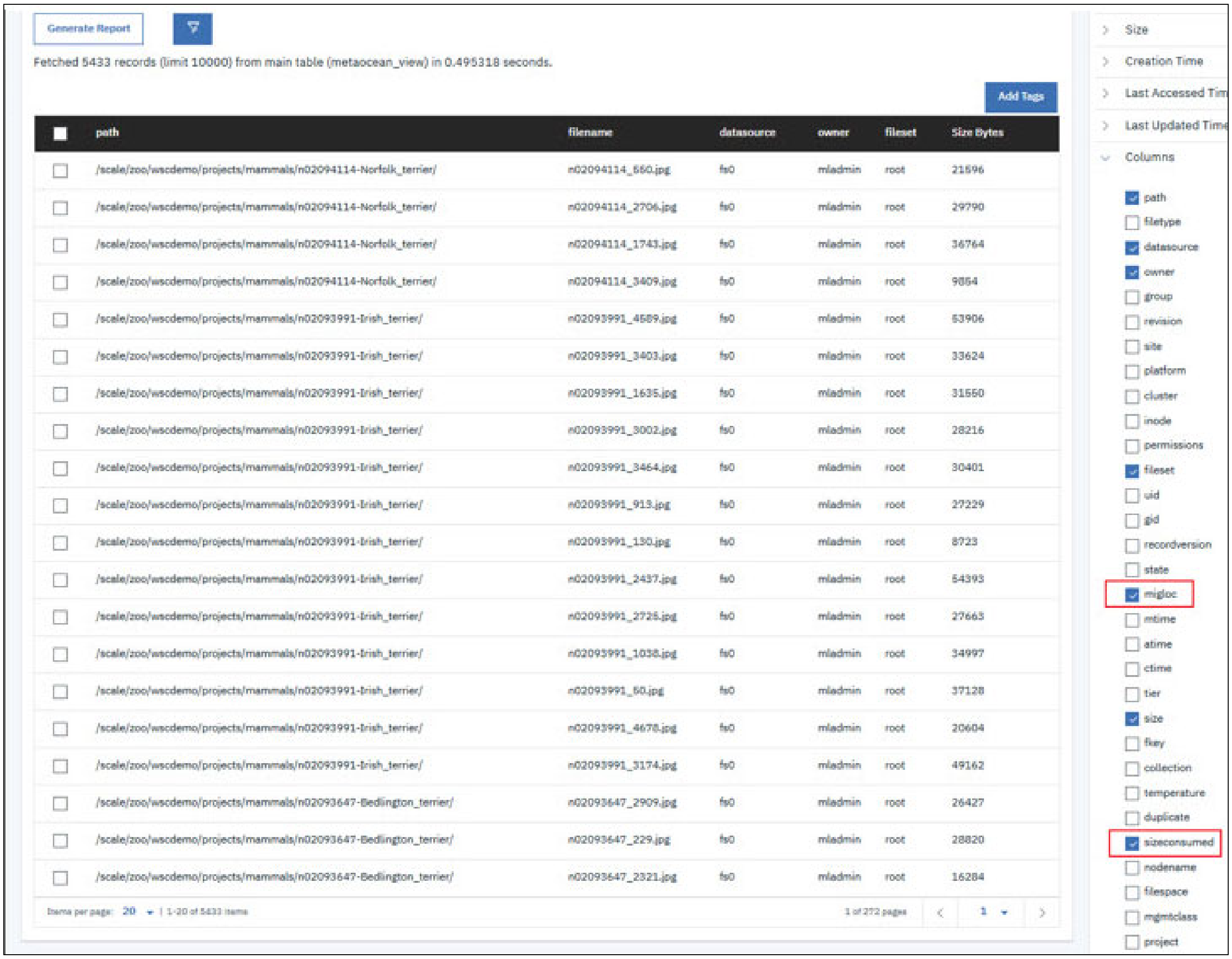


Fetched 5433 records (limit 10000) from main table (metaocean_view) in 0.495318 seconds.

□	path	filename	datasource	owner	fileset	Size Bytes
□	/scale/zoo/wscdemo/projects/mammals/n02094114-Norfolk_terrier/	n02094114_550.jpg	fs0	mladmin	root	21596
□	/scale/zoo/wscdemo/projects/mammals/n02094114-Norfolk_terrier/	n02094114_2706.jpg	fs0	mladmin	root	29790
□	/scale/zoo/wscdemo/projects/mammals/n02094114-Norfolk_terrier/	n02094114_1743.jpg	fs0	mladmin	root	36764
□	/scale/zoo/wscdemo/projects/mammals/n02094114-Norfolk_terrier/	n02094114_3409.jpg	fs0	mladmin	root	9854
□	/scale/zoo/wscdemo/projects/mammals/n02093991-Irish_terrier/	n02093991_4589.jpg	fs0	mladmin	root	53906
□	/scale/zoo/wscdemo/projects/mammals/n02093991-Irish_terrier/	n02093991_3403.jpg	fs0	mladmin	root	33624
□	/scale/zoo/wscdemo/projects/mammals/n02093991-Irish_terrier/	n02093991_1635.jpg	fs0	mladmin	root	31550
□	/scale/zoo/wscdemo/projects/mammals/n02093991-Irish_terrier/	n02093991_3002.jpg	fs0	mladmin	root	28216
□	/scale/zoo/wscdemo/projects/mammals/n02093991-Irish_terrier/	n02093991_3464.jpg	fs0	mladmin	root	30401
□	/scale/zoo/wscdemo/projects/mammals/n02093991-Irish_terrier/	n02093991_913.jpg	fs0	mladmin	root	27229

Figure 5-7 Individual view of the migrtd files

5. Then, on the right side, select **migloc** and **sizeconsumed**, and click **Apply**, as shown in Figure 5-8.



Fetched 5433 records (limit 10000) from main table (metaocean_view) in 0.495318 seconds.

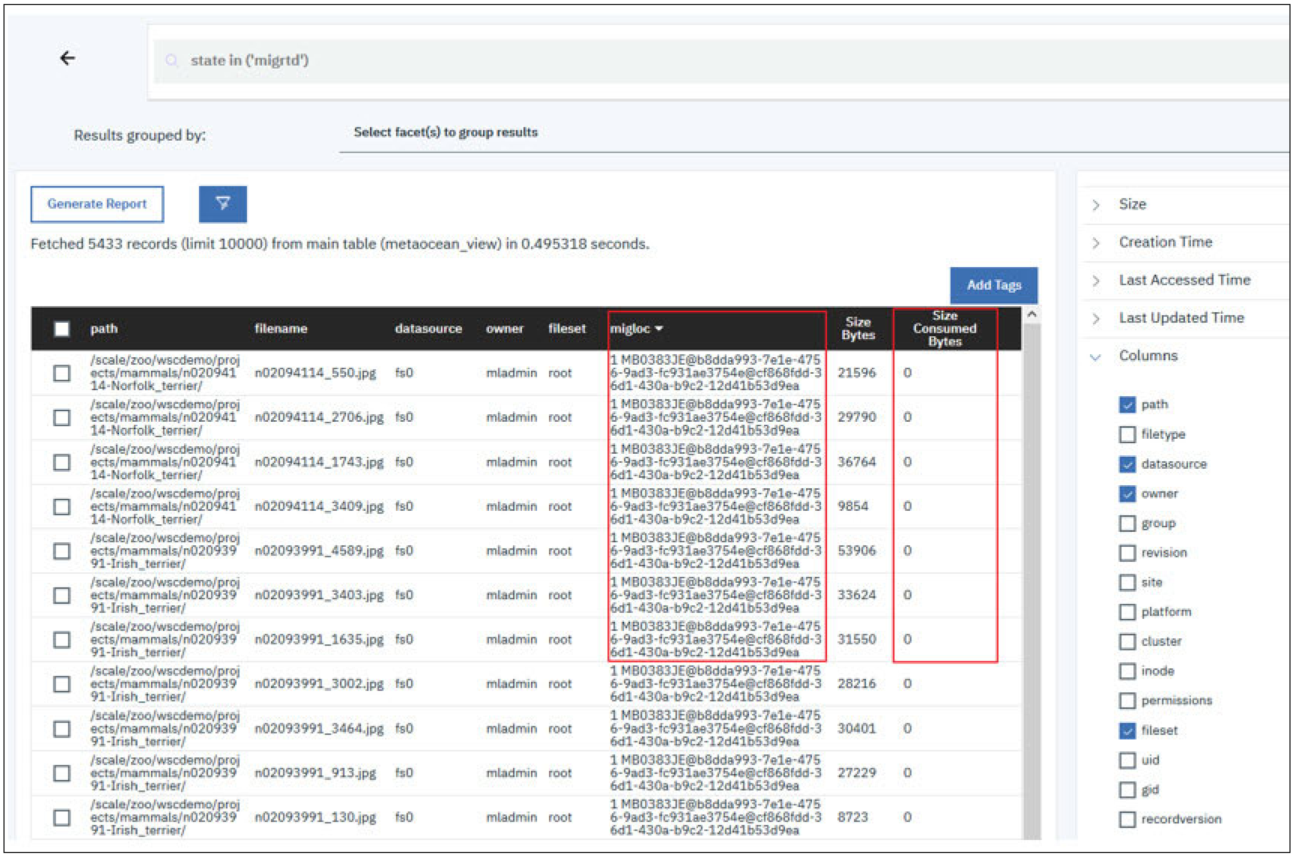
☐	path	filename	datasource	owner	fileset	Size Bytes
☐	/scale/zoo/wscdemo/projects/mammals/n02094114-Norfolk_terrier/	n02094114_660.jpg	fs0	mladmin	root	21596
☐	/scale/zoo/wscdemo/projects/mammals/n02094114-Norfolk_terrier/	n02094114_2706.jpg	fs0	mladmin	root	29790
☐	/scale/zoo/wscdemo/projects/mammals/n02094114-Norfolk_terrier/	n02094114_1743.jpg	fs0	mladmin	root	36764
☐	/scale/zoo/wscdemo/projects/mammals/n02094114-Norfolk_terrier/	n02094114_3409.jpg	fs0	mladmin	root	9864
☐	/scale/zoo/wscdemo/projects/mammals/n02093991-Irish_terrier/	n02093991_4589.jpg	fs0	mladmin	root	53906
☐	/scale/zoo/wscdemo/projects/mammals/n02093991-Irish_terrier/	n02093991_3403.jpg	fs0	mladmin	root	33624
☐	/scale/zoo/wscdemo/projects/mammals/n02093991-Irish_terrier/	n02093991_1635.jpg	fs0	mladmin	root	31560
☐	/scale/zoo/wscdemo/projects/mammals/n02093991-Irish_terrier/	n02093991_3002.jpg	fs0	mladmin	root	28216
☐	/scale/zoo/wscdemo/projects/mammals/n02093991-Irish_terrier/	n02093991_3464.jpg	fs0	mladmin	root	30401
☐	/scale/zoo/wscdemo/projects/mammals/n02093991-Irish_terrier/	n02093991_913.jpg	fs0	mladmin	root	27229
☐	/scale/zoo/wscdemo/projects/mammals/n02093991-Irish_terrier/	n02093991_130.jpg	fs0	mladmin	root	8723
☐	/scale/zoo/wscdemo/projects/mammals/n02093991-Irish_terrier/	n02093991_2437.jpg	fs0	mladmin	root	54393
☐	/scale/zoo/wscdemo/projects/mammals/n02093991-Irish_terrier/	n02093991_2726.jpg	fs0	mladmin	root	27663
☐	/scale/zoo/wscdemo/projects/mammals/n02093991-Irish_terrier/	n02093991_1038.jpg	fs0	mladmin	root	34997
☐	/scale/zoo/wscdemo/projects/mammals/n02093991-Irish_terrier/	n02093991_60.jpg	fs0	mladmin	root	37128
☐	/scale/zoo/wscdemo/projects/mammals/n02093991-Irish_terrier/	n02093991_4678.jpg	fs0	mladmin	root	20604
☐	/scale/zoo/wscdemo/projects/mammals/n02093991-Irish_terrier/	n02093991_3174.jpg	fs0	mladmin	root	49162
☐	/scale/zoo/wscdemo/projects/mammals/n02093647-Bedlington_terrier/	n02093647_2909.jpg	fs0	mladmin	root	26427
☐	/scale/zoo/wscdemo/projects/mammals/n02093647-Bedlington_terrier/	n02093647_229.jpg	fs0	mladmin	root	28820
☐	/scale/zoo/wscdemo/projects/mammals/n02093647-Bedlington_terrier/	n02093647_2321.jpg	fs0	mladmin	root	16284

› Size
› Creation Time
› Last Accessed Tim
› Last Updated Time
⌄ Columns

☑ path
☐ filetype
☑ datasource
☑ owner
☐ group
☐ revision
☐ site
☐ platform
☐ cluster
☐ inode
☐ permissions
☑ fileset
☐ uid
☐ gid
☐ recordversion
☐ state
☑ migloc
☐ mtime
☐ atime
☐ ctime
☐ tier
☑ size
☐ fkey
☐ collection
☐ temperature
☐ duplicate
☑ sizeconsumed
☐ nodename
☐ filespace
☐ mgmtclass
☐ project

Figure 5-8 Adding columns into the individual view of the migrtd files

More information about the migration status is displayed to help storage administrators or data engineers to manage the data in the IBM Spectrum Archive, as shown in Figure 5-9.

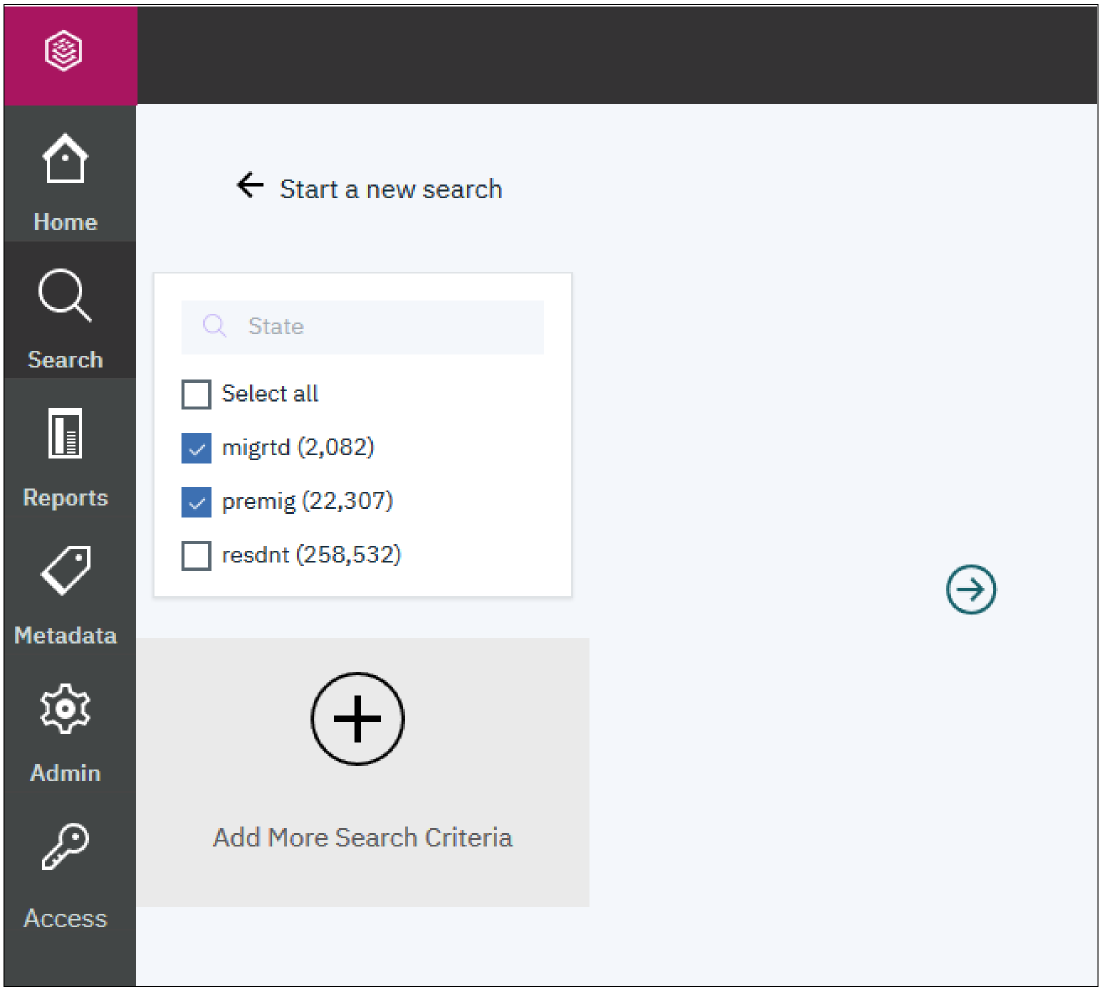


Fetched 5433 records (limit 10000) from main table (metaocean_view) in 0.495318 seconds.

□	path	filename	datasource	owner	fileset	migloc	Size Bytes	Size Consumed Bytes
□	/scale/zoo/wscdemo/projects/mammals/n02094114-Norfolk_terrier/	n02094114_550.jpg	fs0	mladmin	root	1 MB0383JE@b8dda993-7e1e-4756-9ad3-fc931ae3754e@cf868fdd-36d1-430a-b9c2-12d41b53d9ea	21596	0
□	/scale/zoo/wscdemo/projects/mammals/n02094114-Norfolk_terrier/	n02094114_2706.jpg	fs0	mladmin	root	1 MB0383JE@b8dda993-7e1e-4756-9ad3-fc931ae3754e@cf868fdd-36d1-430a-b9c2-12d41b53d9ea	29790	0
□	/scale/zoo/wscdemo/projects/mammals/n02094114-Norfolk_terrier/	n02094114_1743.jpg	fs0	mladmin	root	1 MB0383JE@b8dda993-7e1e-4756-9ad3-fc931ae3754e@cf868fdd-36d1-430a-b9c2-12d41b53d9ea	36764	0
□	/scale/zoo/wscdemo/projects/mammals/n02094114-Norfolk_terrier/	n02094114_3409.jpg	fs0	mladmin	root	1 MB0383JE@b8dda993-7e1e-4756-9ad3-fc931ae3754e@cf868fdd-36d1-430a-b9c2-12d41b53d9ea	9854	0
□	/scale/zoo/wscdemo/projects/mammals/n02093991-Irish_terrier/	n02093991_4589.jpg	fs0	mladmin	root	1 MB0383JE@b8dda993-7e1e-4756-9ad3-fc931ae3754e@cf868fdd-36d1-430a-b9c2-12d41b53d9ea	53906	0
□	/scale/zoo/wscdemo/projects/mammals/n02093991-Irish_terrier/	n02093991_3403.jpg	fs0	mladmin	root	1 MB0383JE@b8dda993-7e1e-4756-9ad3-fc931ae3754e@cf868fdd-36d1-430a-b9c2-12d41b53d9ea	33624	0
□	/scale/zoo/wscdemo/projects/mammals/n02093991-Irish_terrier/	n02093991_1635.jpg	fs0	mladmin	root	1 MB0383JE@b8dda993-7e1e-4756-9ad3-fc931ae3754e@cf868fdd-36d1-430a-b9c2-12d41b53d9ea	31550	0
□	/scale/zoo/wscdemo/projects/mammals/n02093991-Irish_terrier/	n02093991_3002.jpg	fs0	mladmin	root	1 MB0383JE@b8dda993-7e1e-4756-9ad3-fc931ae3754e@cf868fdd-36d1-430a-b9c2-12d41b53d9ea	28216	0
□	/scale/zoo/wscdemo/projects/mammals/n02093991-Irish_terrier/	n02093991_3464.jpg	fs0	mladmin	root	1 MB0383JE@b8dda993-7e1e-4756-9ad3-fc931ae3754e@cf868fdd-36d1-430a-b9c2-12d41b53d9ea	30401	0
□	/scale/zoo/wscdemo/projects/mammals/n02093991-Irish_terrier/	n02093991_913.jpg	fs0	mladmin	root	1 MB0383JE@b8dda993-7e1e-4756-9ad3-fc931ae3754e@cf868fdd-36d1-430a-b9c2-12d41b53d9ea	27229	0
□	/scale/zoo/wscdemo/projects/mammals/n02093991-Irish_terrier/	n02093991_130.jpg	fs0	mladmin	root	1 MB0383JE@b8dda993-7e1e-4756-9ad3-fc931ae3754e@cf868fdd-36d1-430a-b9c2-12d41b53d9ea	8723	0

Figure 5-9 More information about the migration status displayed for the migrtd files

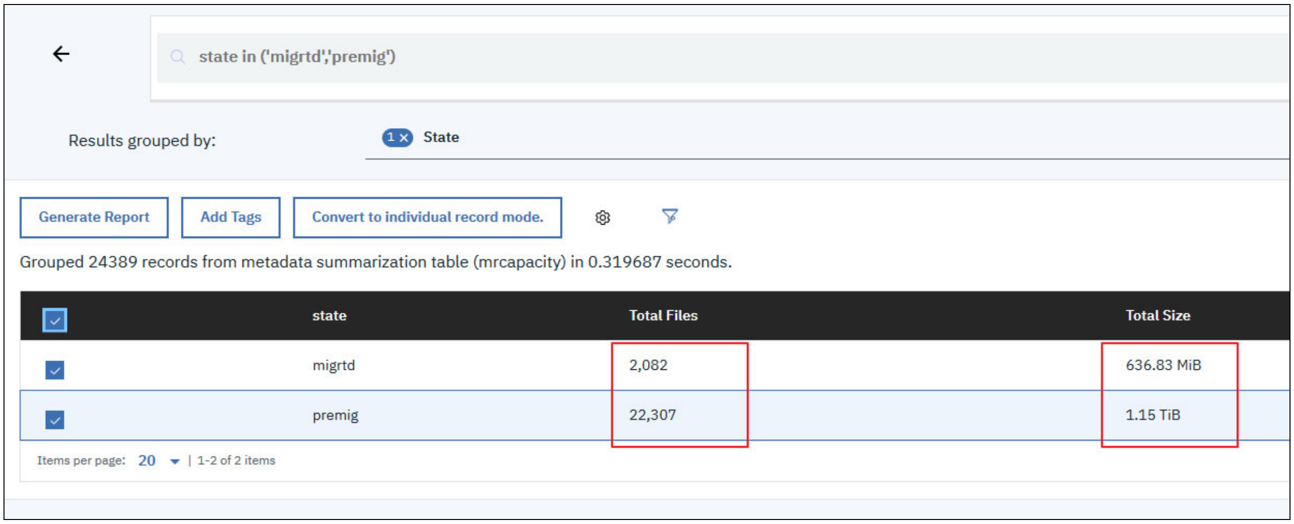
6. To see an overview of the data that is stored on tape, search by using `migrtd` and `premig` in State section, as shown in Figure 5-10.



□ Select all
☑ migrtd (2,082)
☑ premig (22,307)
□ resdnt (258,532)

Figure 5-10 Search for migrtd and premig files

7. The search result includes the total number of files in `migrtd` and `premig` state and their accumulated total size, which uses tape capacity, as shown in Figure 5-11.



state in ('migrtd','premig')
Results grouped by:
1 × State
Generate Report
Add Tags
Convert to individual record mode.

Grouped 24389 records from metadata summarization table (mrcapacity) in 0.319687 seconds.

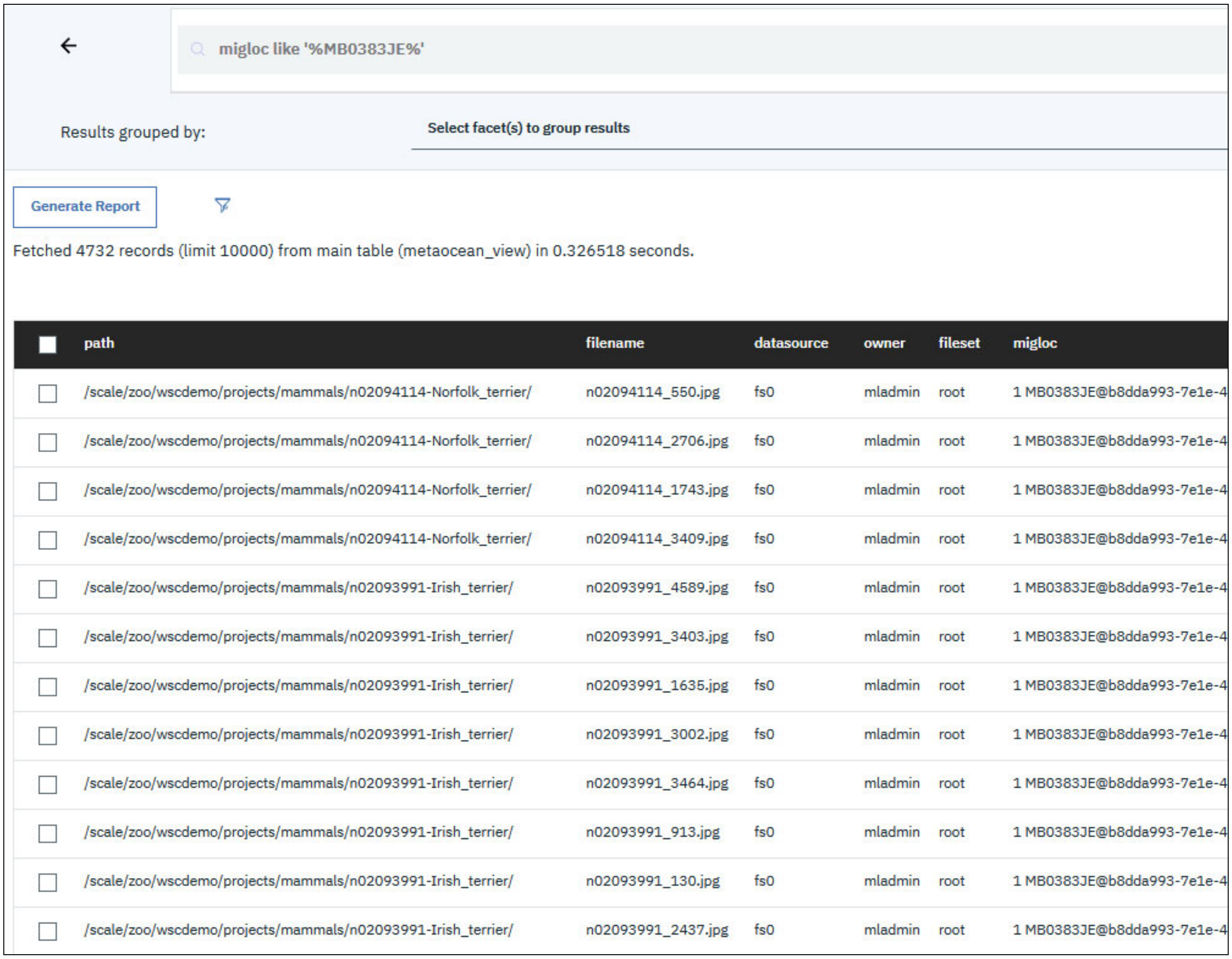
☑	state	Total Files	Total Size
☑	migrtd	2,082	636.83 MiB
☑	premig	22,307	1.15 TiB

Items per page: 20 | 1-2 of 2 items

Figure 5-11 View of the migrtd and premig files

8. Another perspective that data engineers or storage administrators might be interested in is to determine what data is on a specific tape.

   Because the tape cartridge to be worked on is MB0383JE, the following search criteria, "`migloc like '%MB0383JE%'`", extracts the metadata of all files that are stored on It, as shown in Figure 5-12.



migloc like '%MB0383JE%'

Results grouped by: Select facet(s) to group results

Fetched 4732 records (limit 10000) from main table (metaocean_view) in 0.326518 seconds.

□	path	filename	datasource	owner	fileset	migloc
□	/scale/zoo/wscdemo/projects/mammals/n02094114-Norfolk_terrier/	n02094114_550.jpg	fs0	mladmin	root	1 MB0383JE@b8dda993-7e1e-4
□	/scale/zoo/wscdemo/projects/mammals/n02094114-Norfolk_terrier/	n02094114_2706.jpg	fs0	mladmin	root	1 MB0383JE@b8dda993-7e1e-4
□	/scale/zoo/wscdemo/projects/mammals/n02094114-Norfolk_terrier/	n02094114_1743.jpg	fs0	mladmin	root	1 MB0383JE@b8dda993-7e1e-4
□	/scale/zoo/wscdemo/projects/mammals/n02094114-Norfolk_terrier/	n02094114_3409.jpg	fs0	mladmin	root	1 MB0383JE@b8dda993-7e1e-4
□	/scale/zoo/wscdemo/projects/mammals/n02093991-Irish_terrier/	n02093991_4589.jpg	fs0	mladmin	root	1 MB0383JE@b8dda993-7e1e-4
□	/scale/zoo/wscdemo/projects/mammals/n02093991-Irish_terrier/	n02093991_3403.jpg	fs0	mladmin	root	1 MB0383JE@b8dda993-7e1e-4
□	/scale/zoo/wscdemo/projects/mammals/n02093991-Irish_terrier/	n02093991_1635.jpg	fs0	mladmin	root	1 MB0383JE@b8dda993-7e1e-4
□	/scale/zoo/wscdemo/projects/mammals/n02093991-Irish_terrier/	n02093991_3002.jpg	fs0	mladmin	root	1 MB0383JE@b8dda993-7e1e-4
□	/scale/zoo/wscdemo/projects/mammals/n02093991-Irish_terrier/	n02093991_3464.jpg	fs0	mladmin	root	1 MB0383JE@b8dda993-7e1e-4
□	/scale/zoo/wscdemo/projects/mammals/n02093991-Irish_terrier/	n02093991_913.jpg	fs0	mladmin	root	1 MB0383JE@b8dda993-7e1e-4
□	/scale/zoo/wscdemo/projects/mammals/n02093991-Irish_terrier/	n02093991_130.jpg	fs0	mladmin	root	1 MB0383JE@b8dda993-7e1e-4
□	/scale/zoo/wscdemo/projects/mammals/n02093991-Irish_terrier/	n02093991_2437.jpg	fs0	mladmin	root	1 MB0383JE@b8dda993-7e1e-4

Figure 5-12 Search files on a certain tape

5.4.2 Data movement with IBM Spectrum Discover

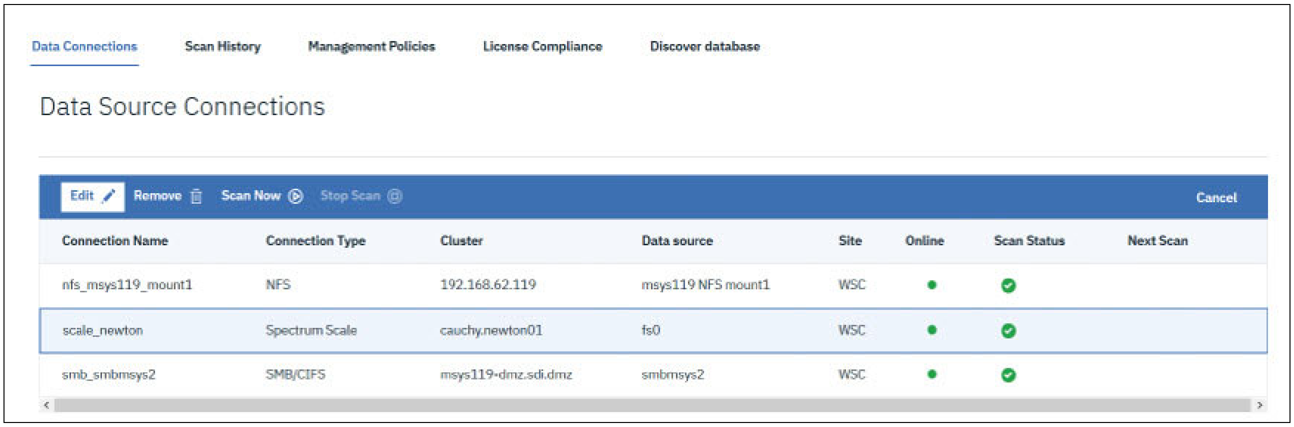
As described in 1.6.5, "Data movement with IBM Spectrum Discover" on page 23, the 2.0.3.1 release of IBM Spectrum Discover made the data movement between select data sources much easier. Spectrum Discover can identify the data sets in the migration scope with tags, reports, and views, and define the migration characteristics and trigger the data movement. It uses the following internal and external approaches:

- ScaleILM, an IBM Spectrum Scale based management policy inside Spectrum Discover, uses the Information Lifecycle Management (ILM) engine of IBM Spectrum Scale to define and realize the data migration actions. In the following sample use cases, ScaleILM is used to demonstrate the data optimization and governance.
- An external agent that integrates external application capabilities with Spectrum Discover DEEPINSPECT policies to orchestrate the data movement.

5.5 Implementation key points

The IBM Spectrum Discover integration with IBM Spectrum Archive EE is enabled through IBM Spectrum Scale data source connections. Therefore, most of the implementation steps to integrate IBM Spectrum Discover with IBM Spectrum Archive EE are the same as the steps with IBM Spectrum Scale.

As shown in Figure 5-13, `scale_newton` is a data source connection with IBM Spectrum Scale as the Connection Type, but it involves IBM Spectrum Scale and IBM Spectrum Archive EE technology behind. This data source is used later in sample use cases.



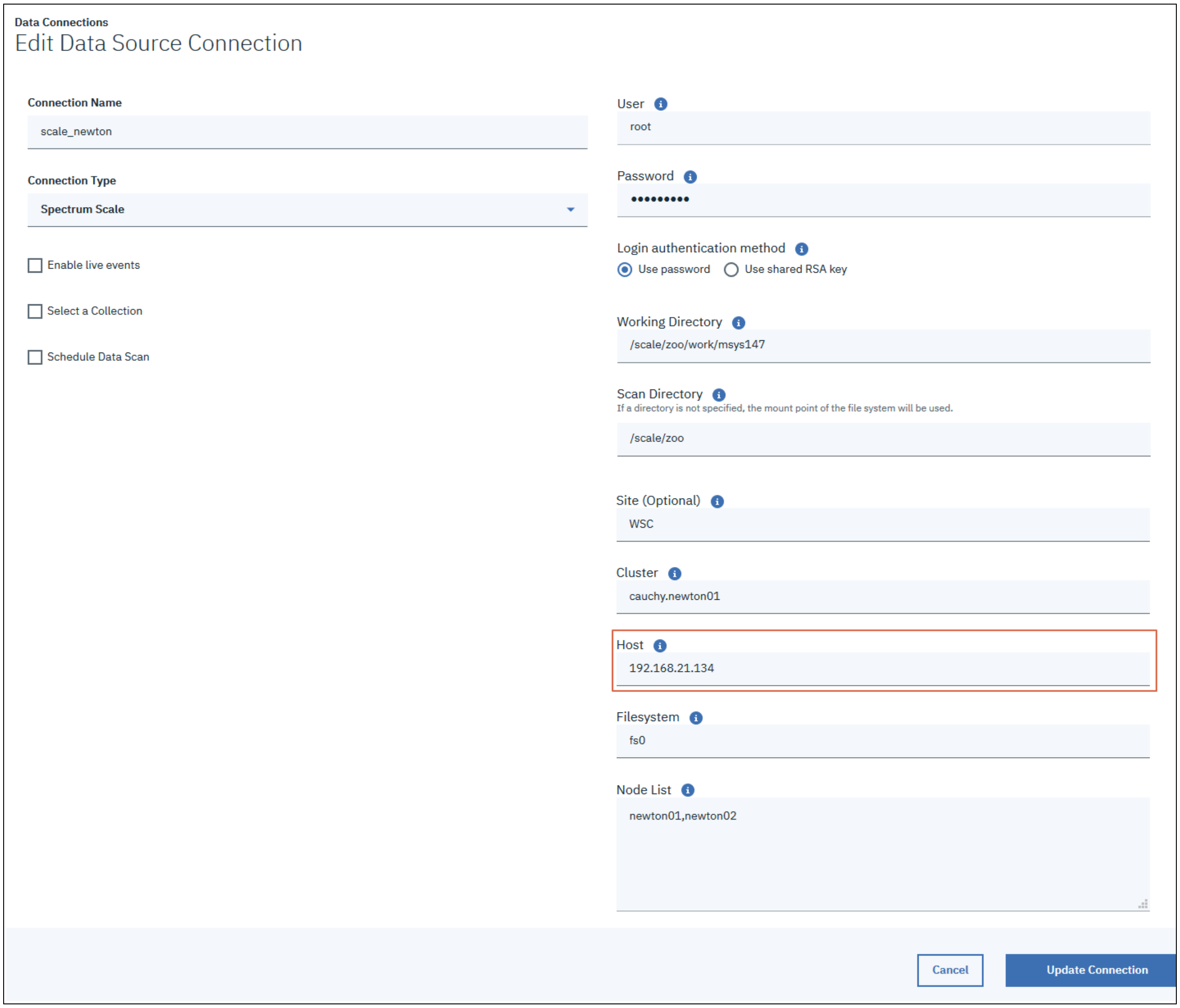
Connection Name	Connection Type	Cluster	Data source	Site	Online	Scan Status	Next Scan
nfs_msys119_mount1	NFS	192.168.62.119	msys119 NFS mount1	WSC	●	✓	
scale_newton	Spectrum Scale	cauchy.newton01	fs0	WSC	●	✓	
smb_smbmsys2	SMB/CIFS	msys119-dmz.sdi.dmz	smbmsys2	WSC	●	✓	

Figure 5-13 Data Source Connection defined for IBM Spectrum Scale and IBM Spectrum Archive

For more information about how to create a IBM Spectrum Scale data connection in IBM Spectrum Discover, see IBM Knowledge Center.

However, consider the following points about setting up data source connections that cover IBM Spectrum Scale and IBM Spectrum Archive in IBM Spectrum Discover if the data movement to external tape pool by using ScaleILM management policies is required:

- IBM Spectrum Archive EE 1.3.0.6 or later must be installed.
- When creating the IBM Spectrum Scale and IBM Spectrum Archive data source connection in IBM Spectrum Discover, the Host setting of a connection must specify one of the IBM Spectrum Archive nodes, as shown in Figure 5-14 on page 115.



Data Connections

Edit Data Source Connection

Connection Name
scale_newton

Connection Type
Spectrum Scale

□ Enable live events

□ Select a Collection

□ Schedule Data Scan

User
root

Password
•••••••••

Login authentication method
● Use password ○ Use shared RSA key

Working Directory
/scale/zoo/work/msys147

Scan Directory
If a directory is not specified, the mount point of the file system will be used.
/scale/zoo

Site (Optional)
WSC

Cluster
cauchy.newton01

Host
192.168.21.134

Filesystem
fs0

Node List
newton01,newton02

Figure 5-14 Data Source Connection detail defined for IBM Spectrum Scale and IBM Spectrum Archive

- IBM Spectrum Discover ScaleILM application uses the same user that is configured in the IBM Spectrum Scale and IBM Spectrum Archive data source connection to run the Data Movement policies. Therefore, this user has enough privileges (unless the user is root) to run the related IBM Spectrum Scale and IBM Spectrum Archive commands and access the affected files.

> **Note:** IBM Spectrum Discover uses the systems that are specified in Node List to perform the scan. The Host is used to perform the data movement.

For more information, see IBM Knowledge Center.

5.6 Sample use cases

In this section, the following data movement use cases are demonstrated that use ScaleILM in IBM Spectrum Discover, which show opposite data migration directions between storage tiers for different purposes:

- Data Governance

  Data staging for high-performance processing (moving data to a higher performance tier).

- Data Optimization

  Data migration to tape for cost-efficient archiving (moving data to a tape tier by way of IBM Spectrum Archive.

5.6.1 Data Governance use case: Data staging for high-performance processing

During the data preparation phase in data processing in various industries, required data sets often must be staged to the high-performance storage tier to improve the overall processing efficiency to get the results faster and better use the expensive computing resources.

The following typical data processing use cases require such data staging action:

- Computer Aided Design (CAD) and Computer Aided Engineering (CAE)
- Big data to support decision making
- Model training for machine learning and deep learning

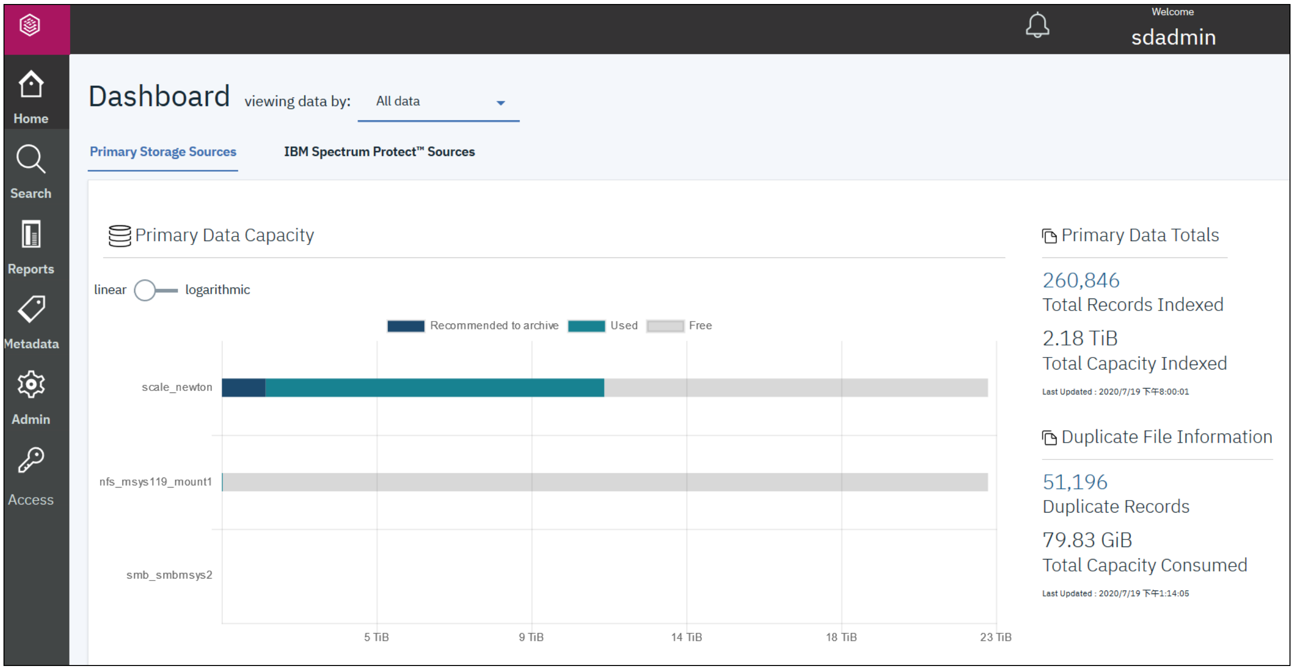
Today, the preferred high-performance storage tier is SSD/Flash based. Because the market price went down significantly in last several years, more enterprises deployed SSD/Flash technology in their data centers to fulfill the increasing demand for low latency and high throughput storage resources.

To complete this data staging activity, the following general procedures are used:

- Identify the data sets that required to be staged.
- Confirm the target storage tier, which meets the performance and capacity requirement.
- Perform the data migration with handy and applied technology.
- Check the migration result.

With IBM Spectrum Discover, IBM Spectrum Scale, and IBM Spectrum Archive, plus built-in ScaleILM application, storage administrators, or data engineers can complete data staging jobs much easier with more visualized and automated aids.

To demonstrate this use case, a sample automotive image data set is used. Such a data set is on IBM Spectrum Scale and managed by IBM Spectrum Discover, as shown in Figure 5-15 on page 117.



Primary Data Totals

260,846
Total Records Indexed

2.18 TiB
Total Capacity Indexed

Last Updated : 2020/7/19 下午8:00:01

Duplicate File Information

51,196
Duplicate Records

79.83 GiB
Total Capacity Consumed

Last Updated : 2020/7/19 下午1:14:05

Figure 5-15 IBM Spectrum Discover Dashboard

In Figure 5-15, `scale_newton` represents an IBM Spectrum Scale file system from a data source connection that consists of IBM Spectrum Scale and IBM Spectrum Archive in IBM Spectrum Discover. This file system is built on two internal storage pools, which are also called two tiers in IBM Spectrum Discover terms. As shown in Figure 5-16, the `system` pool, which is an essential pool in any IBM Spectrum Scale file system, consists of NVMe-based storage in this file system.

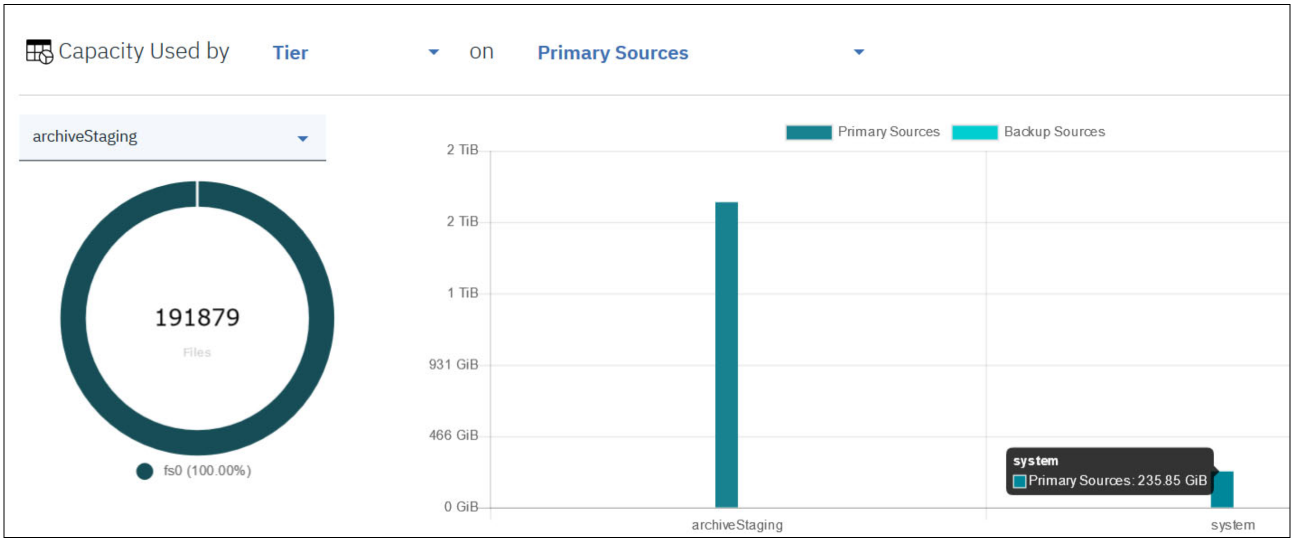


Figure 5-16 System tier in IBM Spectrum Scale file system

However, the other pool, `archiveStaging`, is backed by traditional, low-cost spindle drive technology with larger capacity, as shown in Figure 5-17.

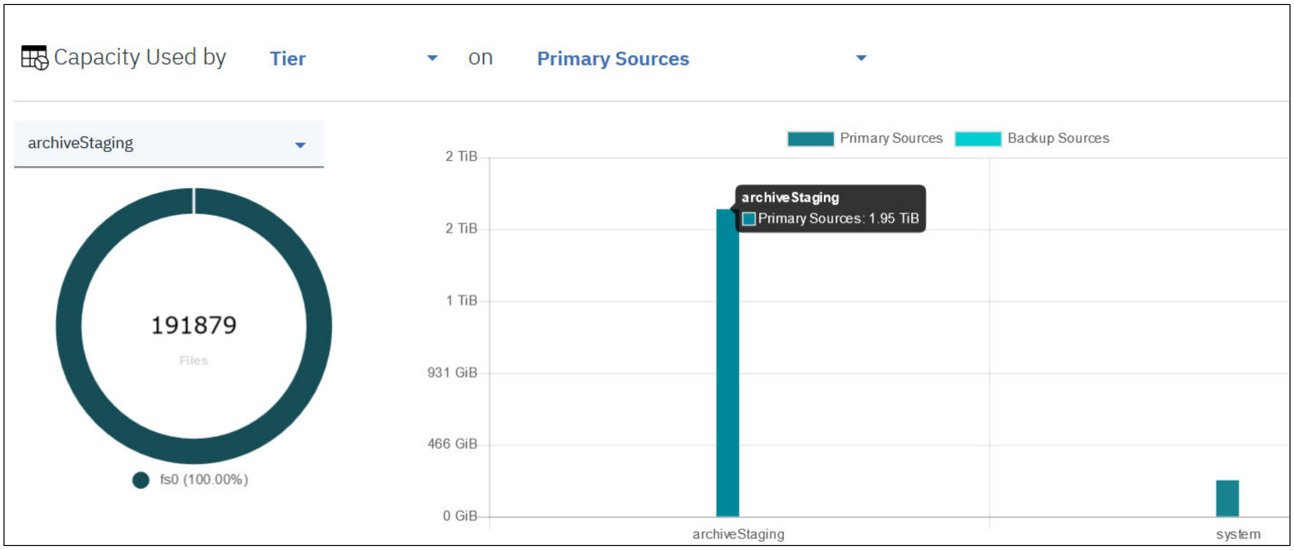


Figure 5-17 ArchiveStaging tier in IBM Spectrum Scale file system

To address the data staging requirement on this IBM Spectrum Scale file system with two storage tiers, the automotive data set is identified and moved the system tier in IBM Spectrum Discover by using ScaleILM.

Complete the following steps:

1. Select the **Search** icon in left pane of the IBM Spectrum Discover GUI. Then, select **State**, **project** and **Tier** tags. Click the **right-pointing arrow** to continue (see Figure 5-18).

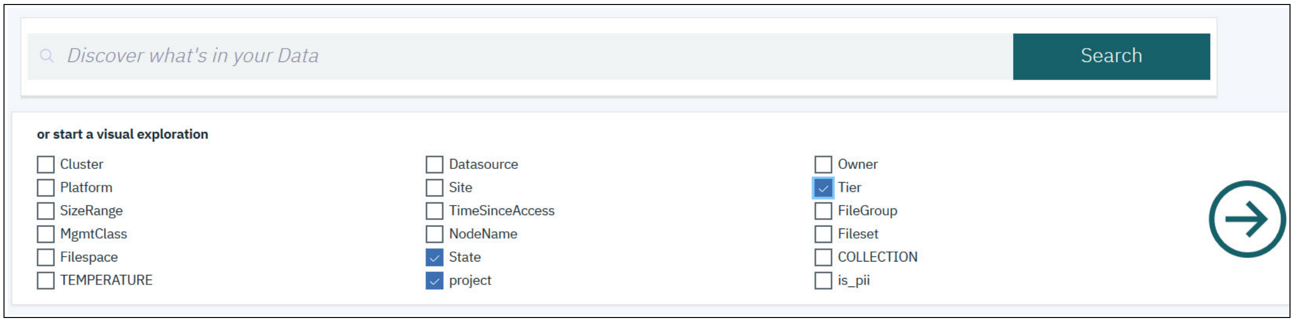
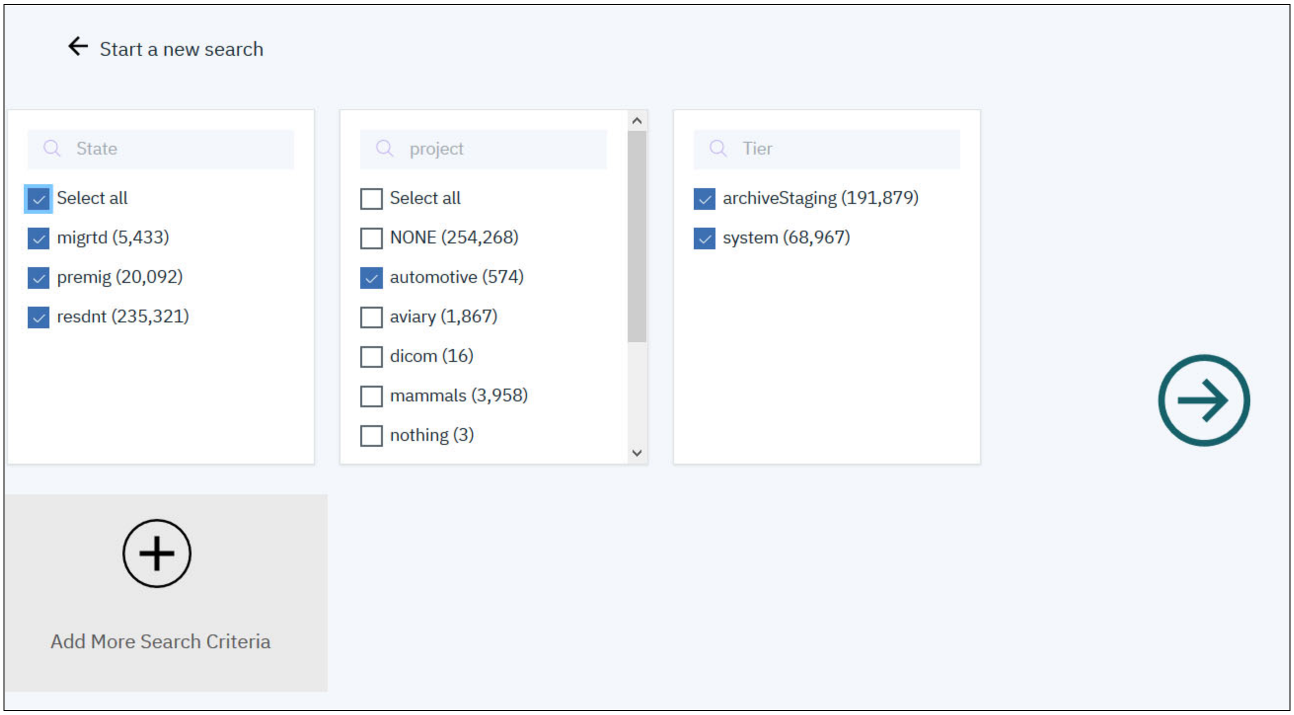


Figure 5-18 Starting the search with State, Project, and Tier tags

2. Select **automotive** in project tag because it is the data set to be used in this example. Select all tiers and states in **Tier** and **State** tag, which shows where the data set is, as shown in Figure 5-19 on page 119. Then, click the **right-pointing arrow** to continue.

> **Note:** The project tag is a customized tag to help administrators identify specific data sets for different projects. The tagging work was done previously and is not covered in this publication. (For more information, see IBM Knowledge Center.) This tagging feature reduces the administrators' workload in data management by reducing the time to find required data sets.



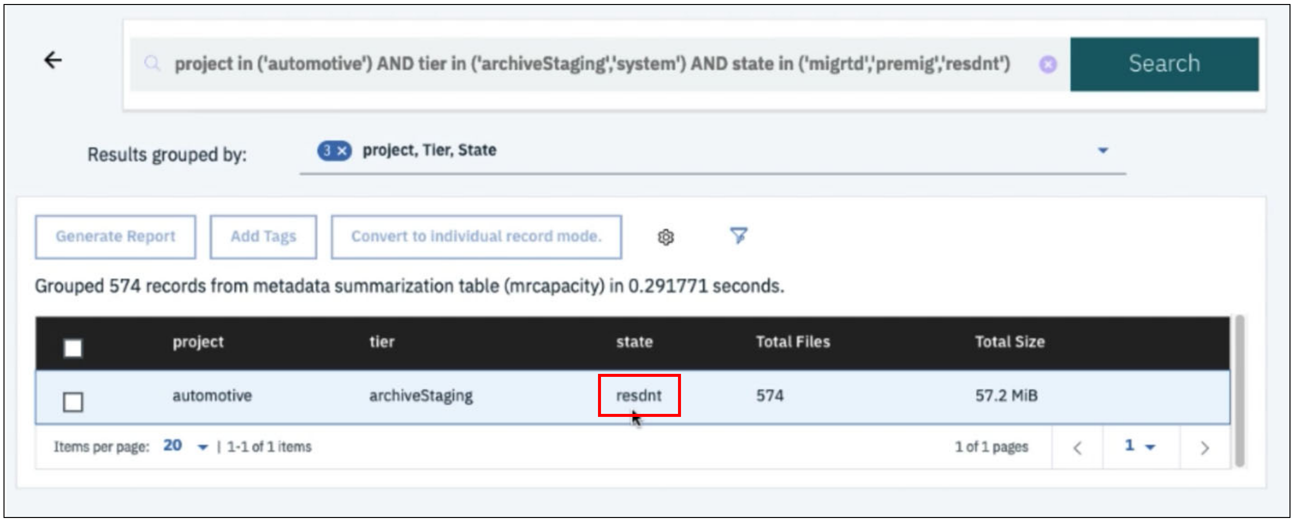
State
- ☑ Select all
- ☑ migrtd (5,433)
- ☑ premig (20,092)
- ☑ resdnt (235,321)

project
- ☐ Select all
- ☐ NONE (254,268)
- ☑ automotive (574)
- ☐ aviary (1,867)
- ☐ dicom (16)
- ☐ mammals (3,958)
- ☐ nothing (3)

Tier
- ☑ archiveStaging (191,879)
- ☑ system (68,967)

Figure 5-19 Refine the search with State, Project, and Tier tags

The search result shows all of images in the `automotive` data set are on the `archiveStaging` tier and all of them are in `resdnt` state, which means they are all on the spindle drives and were not migrated to the tape system (see Figure 5-20).



project in ('automotive') AND tier in ('archiveStaging','system') AND state in ('migrtd','premig','resdnt')

Results grouped by: 3 × project, Tier, State

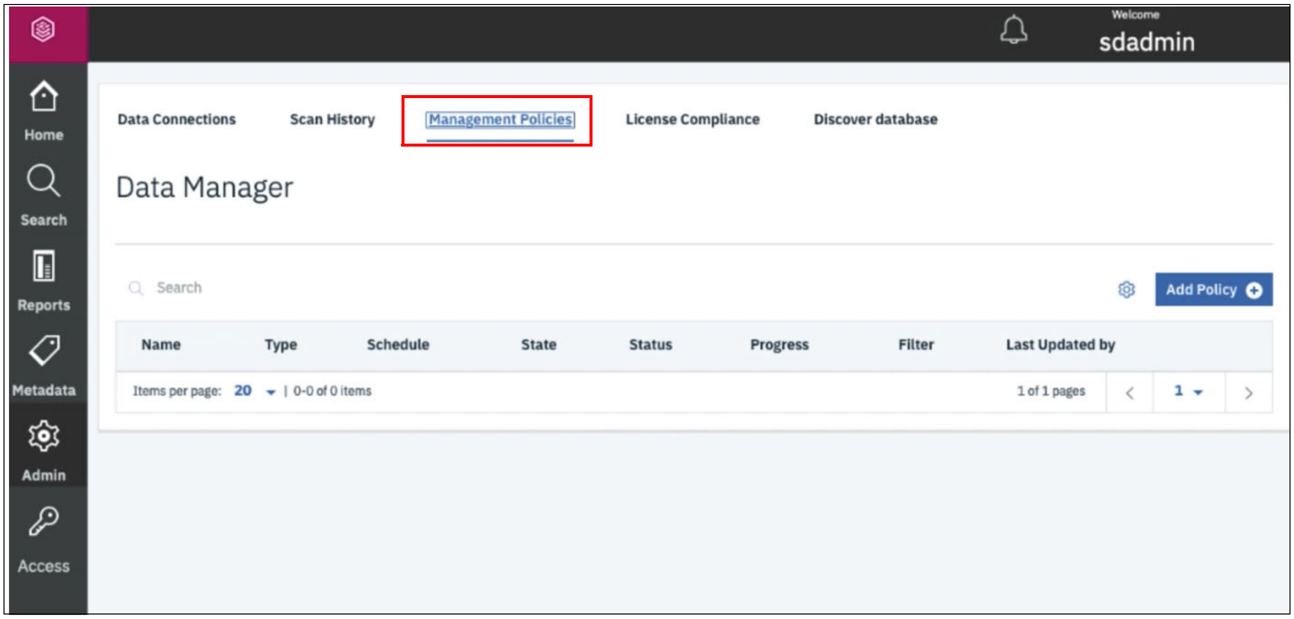
Grouped 574 records from metadata summarization table (mrcapacity) in 0.291771 seconds.

☐	project	tier	state	Total Files	Total Size
☐	automotive	archiveStaging	resdnt	574	57.2 MiB

Items per page: 20 | 1-1 of 1 items 1 of 1 pages 1

Figure 5-20 Search results for automotive data set

3. To create the ScaleILM management policy, select the **Admin** icon in left pane of the IBM Spectrum Discover GUI and click the **Management Policies** tab, as shown in Figure 5-21 on page 120. Then, click **Add Policy** to start the wizard.



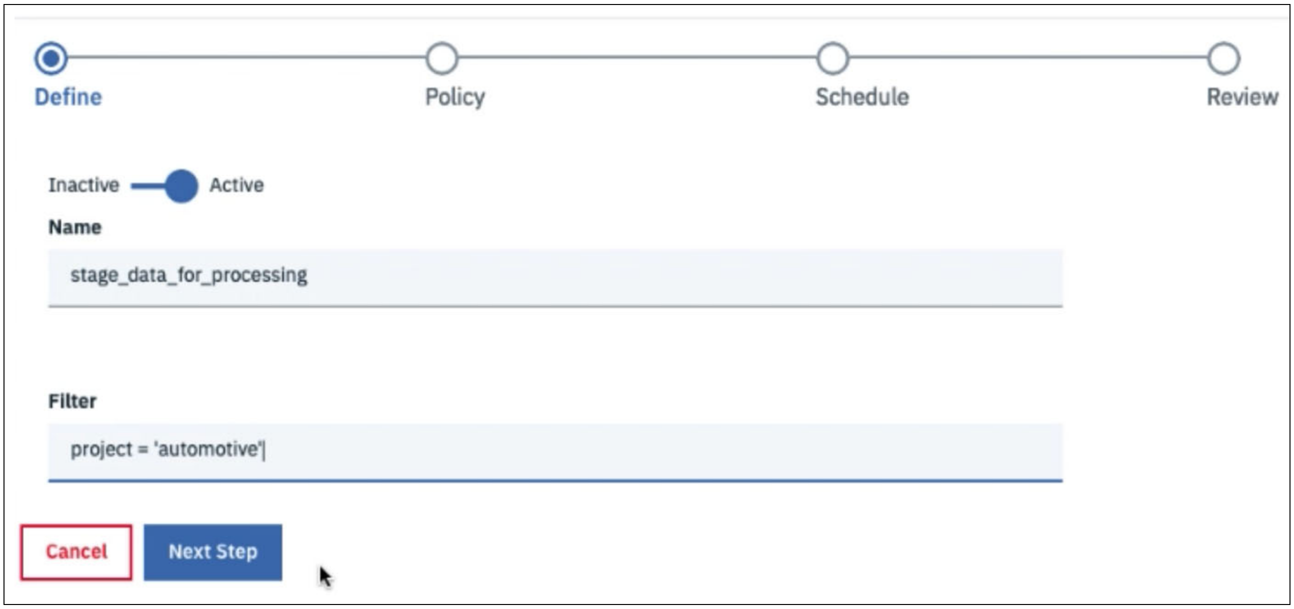
Data Manager

Name	Type	Schedule	State	Status	Progress	Filter	Last Updated by

Items per page: 20 | 0-0 of 0 items 1 of 1 pages 1

Figure 5-21 Create Management Policy

4. In the Define page of the management policy creation wizard, switch the policy to **Active** first, and then name the policy with the phrase that helps easily identify the purpose of this policy in the Name field. The Filter field is used to choose the correct data set for this policy to be applied. In this example, set the **project** tag to **automotive** and click **Next Step** (see Figure 5-22).



Inactive Active

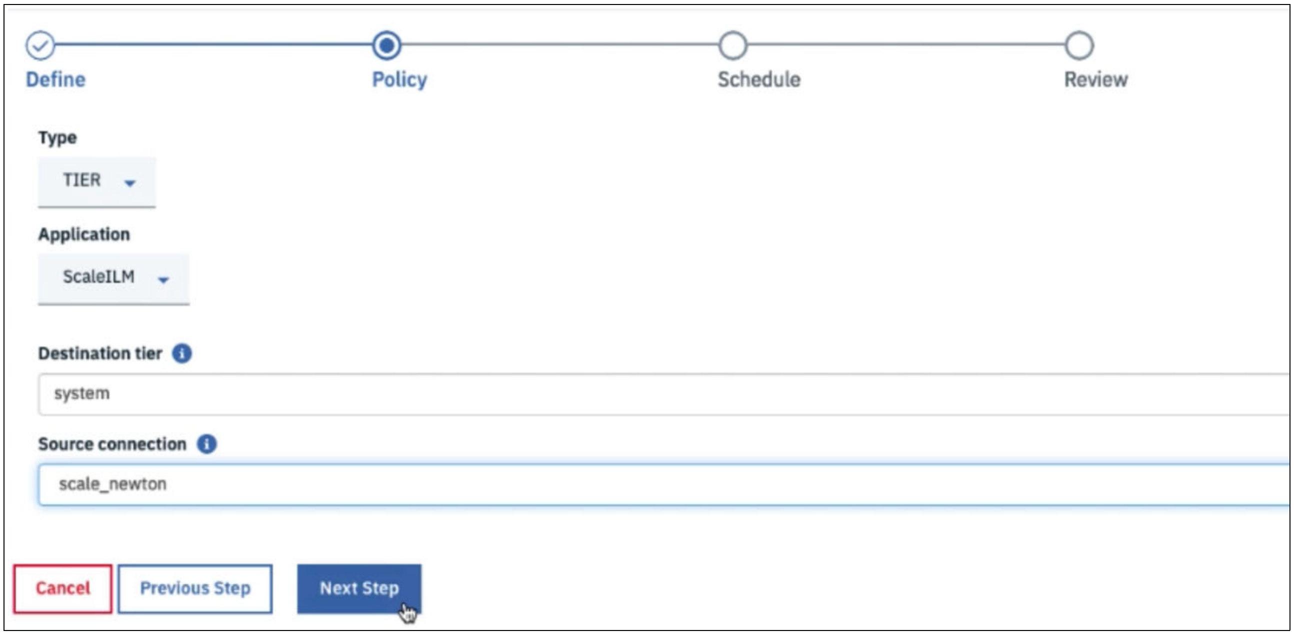
Name: stage_data_for_processing

Filter: project = 'automotive'

Figure 5-22 Define Management Policy for data staging

5. In the next **Policy** page, more information is needed for this policy. Select **TIER** in the Type field, which indicates the policy takes effect in moving data between tiers or storage pools. Then, select **ScaleILM** in Application field to use the information lifecycle management capability that is provided by IBM Spectrum Scale and IBM Spectrum Archive.

   Specify the target tier in **Destination tier** field, which is `system`, and the file system for this policy to work on in the **Source connection** field, which is `scale_newton`, in this example. Then, click **Next Step**, as shown in Figure 5-23.



Type
TIER

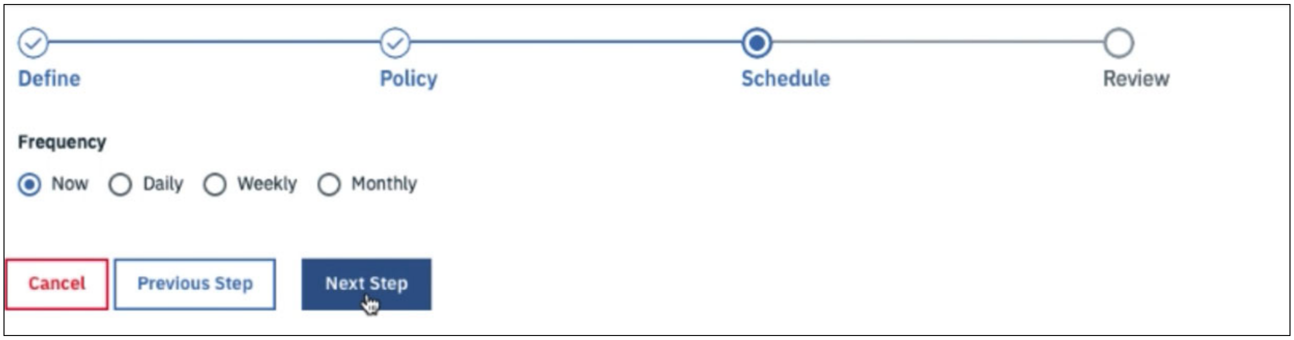
Application
ScaleILM

Destination tier
system

Source connection
scale_newton

Figure 5-23 Fill Management Policy required information for data staging policy

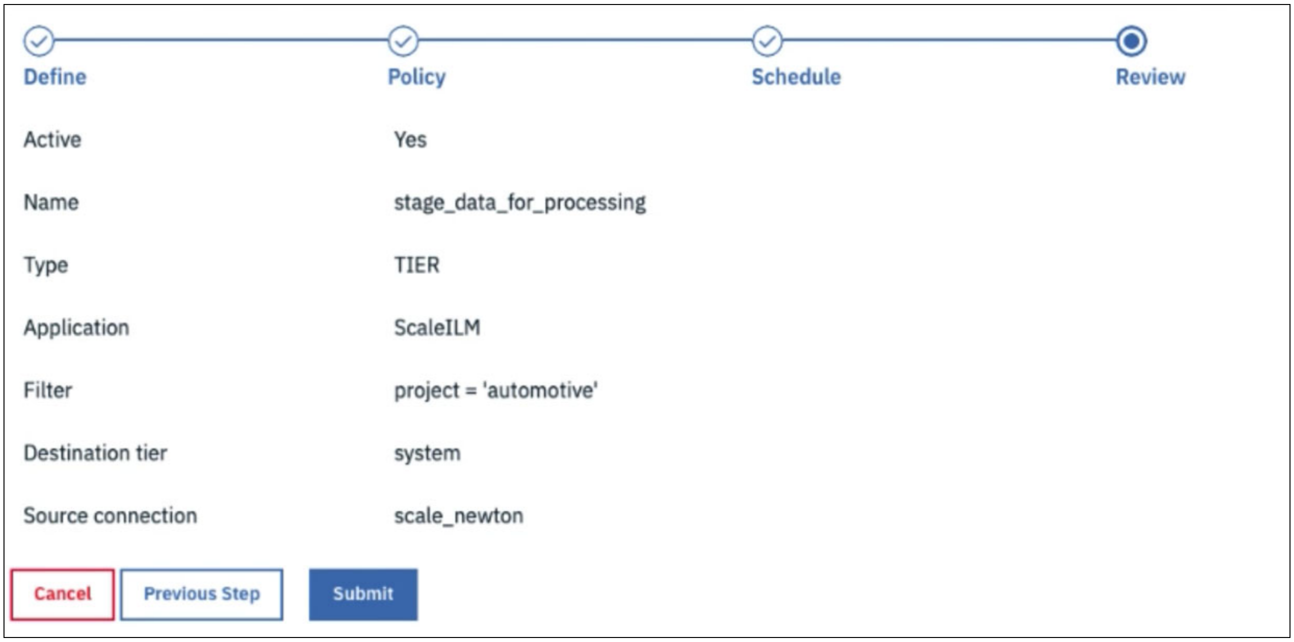
6. On the **Schedule** page, set the frequency for this policy to run. Because the policy is expected to run immediately here, select **Now** and then, click **Next Step**, as shown in Figure 5-24.



Frequency
● Now ○ Daily ○ Weekly ○ Monthly

Figure 5-24 Schedule Management Policy for data staging

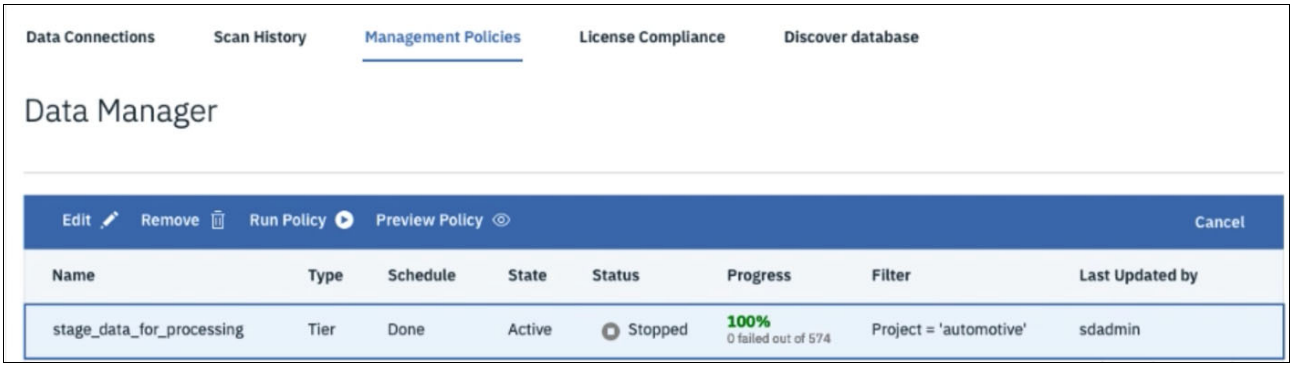
7. Double check the policy setting in the **Review** page and click **Submit**, as shown in Figure 5-25. Then, the policy runs.



Active	Yes
Name	stage_data_for_processing
Type	TIER
Application	ScaleILM
Filter	project = 'automotive'
Destination tier	system
Source connection	scale_newton

Figure 5-25 Review Management Policy for data staging

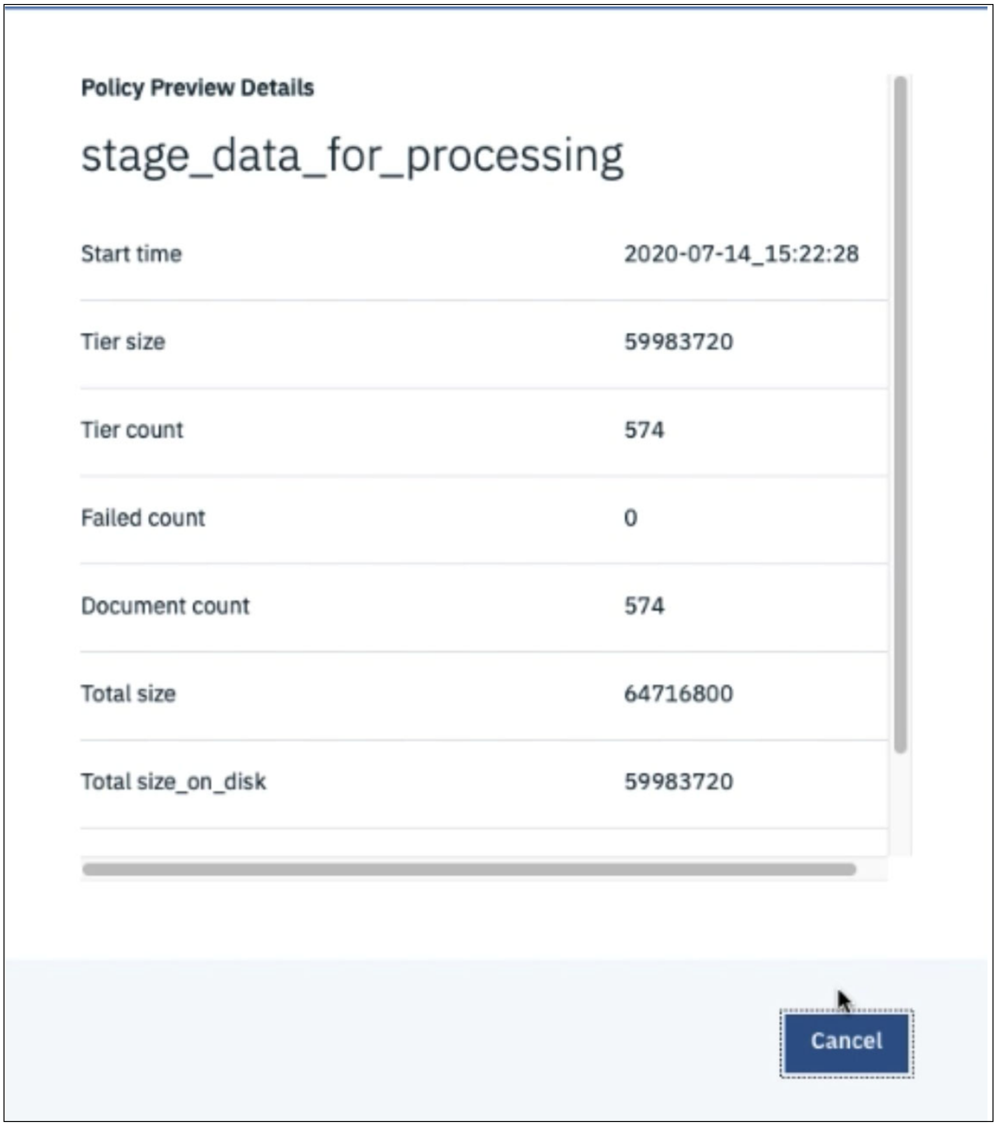
Depending on the different environment setup and different scope of the data sets, the running time of the policy can vary. In this example, the policy execution is completed in several minutes with 100% success, as shown in Figure 5-26.



Name	Type	Schedule	State	Status	Progress	Filter	Last Updated by
stage_data_for_processing	Tier	Done	Active	Stopped	100% 0 failed out of 574	Project = 'automotive'	sdadmin

Figure 5-26 Run Management Policy of data staging

8. Click **Preview Policy** for more information about the results. As shown in Figure 5-27, a total 574 files were identified for movement by the policy and all of them were successfully migrated.



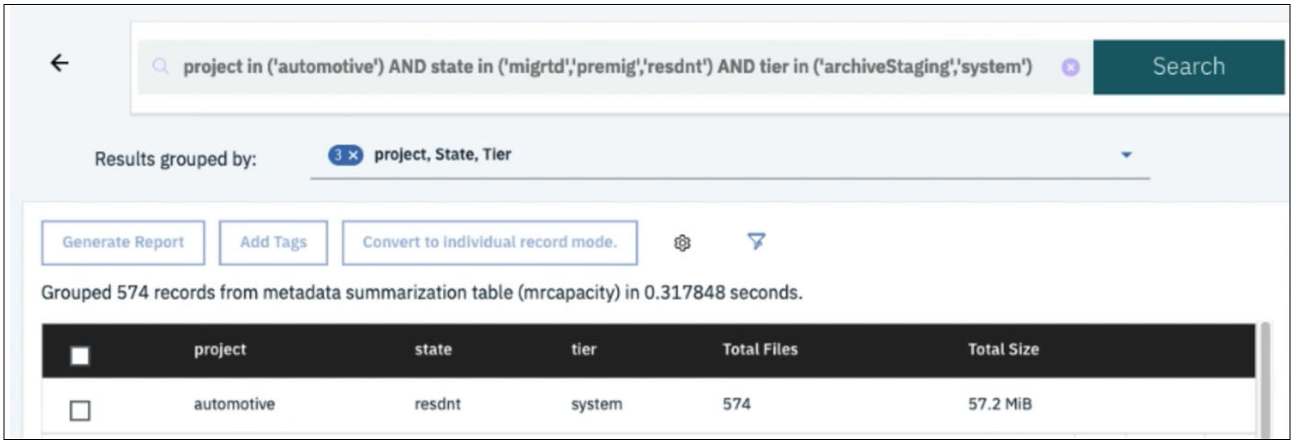
Policy Preview Details

stage_data_for_processing

Start time	2020-07-14_15:22:28
Tier size	59983720
Tier count	574
Failed count	0
Document count	574
Total size	64716800
Total size_on_disk	59983720

Figure 5-27 View the policy detail of data staging

9. To verify that the data was moved to the wanted tier, review the search logic again in steps 1 and 2 and get the updated result for the `automotive` data set. As shown in Figure 5-28 on page 124, the result indicates that all data in the `automotive` data set is in the system tier. By referring to step 3, a successful data staging for the `automotive` data set to the high-performance system tier can be confirmed.



project in ('automotive') AND state in ('migrtd','premig','resdnt') AND tier in ('archiveStaging','system')

Results grouped by: 3 × project, State, Tier

Grouped 574 records from metadata summarization table (mrcapacity) in 0.317848 seconds.

■	project	state	tier	Total Files	Total Size
□	automotive	resdnt	system	574	57.2 MiB

Figure 5-28 Verity the Data Staging results

5.6.2 Data Optimization use case: Data migration to tape for cost-efficient archiving

In the data explosion era, because of government or industry requirements or to get meaningful insights from the data, companies are keeping more data and for longer durations. Although data is rarely accessed, it accumulated day by day, month by month, and year by year, which is expensive to store.

As described in “IBM tape technologies overview” on page 143, tape technology continues to evolve as the greenest storage in saving energy and expense. The physical and mechanical characteristics of tape might not be a good fit as general-purpose storage option. However, it proved to be the best choice for cold data, which is not accessed for months or even years.

IBM Spectrum Scale, IBM Spectrum Archive, and IBM Tape system work together to build an optimized data architecture to automatically handle the full data lifecycle storage requirement with customized policies, from hot data, to warm data, to cold data. Also, IBM Spectrum Discover can integrate into this use case to provide better data and metadata management. Such a modern unstructured data storage use case fits almost any industry, including but not limited to, Financial Services, Government, Telecommunication, Media, Manufacturing, and Transportation.

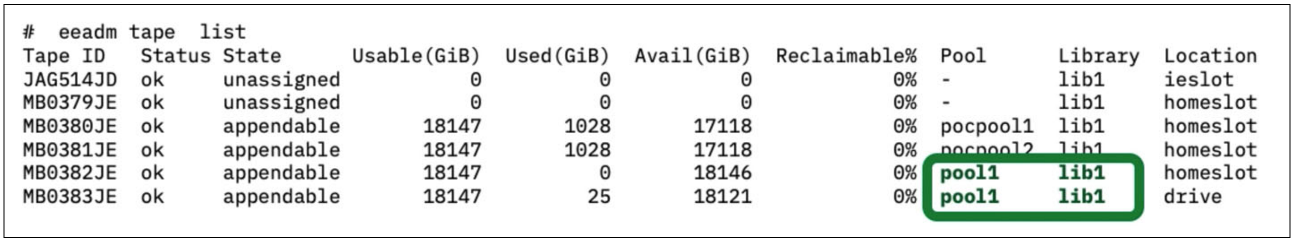
Particular to AI practice, the data sets are archived after training for future use because of its potential value instead of being deleted, including images, videos, and texts. Some might be reused soon, but some might need to wait for months to be accessed again. In such cases, IBM Spectrum Archive, plus IBM Spectrum Scale and IBM Spectrum Discover, can help to make an effective archive use case for cold data.

In this example use case, cold data in the aviary machine learning project is migrated to tape by using ScaleILM in the IBM Spectrum Discover to demonstrate how IBM Spectrum Discover can help to make the data lifecycle activities more administrator-friendly.

Tip: The reverse of this use case is also a common example: a mass data migration for data scaling moving large data sets from a tape tier that is managed by IBM Spectrum Archive EE to an active tier of disk or flash.

The environment to be used in this use case is the same as the environment that was used in the use case with `scale_newton` data source.

Moreover, another external tape pool in this IBM Spectrum Scale and IBM Spectrum Archive setup is introduced, as shown in F Figure 5-29. In the output of IBM Spectrum Archive command **eeadm**, `pool1` is the name of the external tape pool and `lib1` is the name of the tape system (both are defined in IBM Spectrum Archive). Such information is required before the ScaleILM management policy is created in IBM Spectrum Discover.

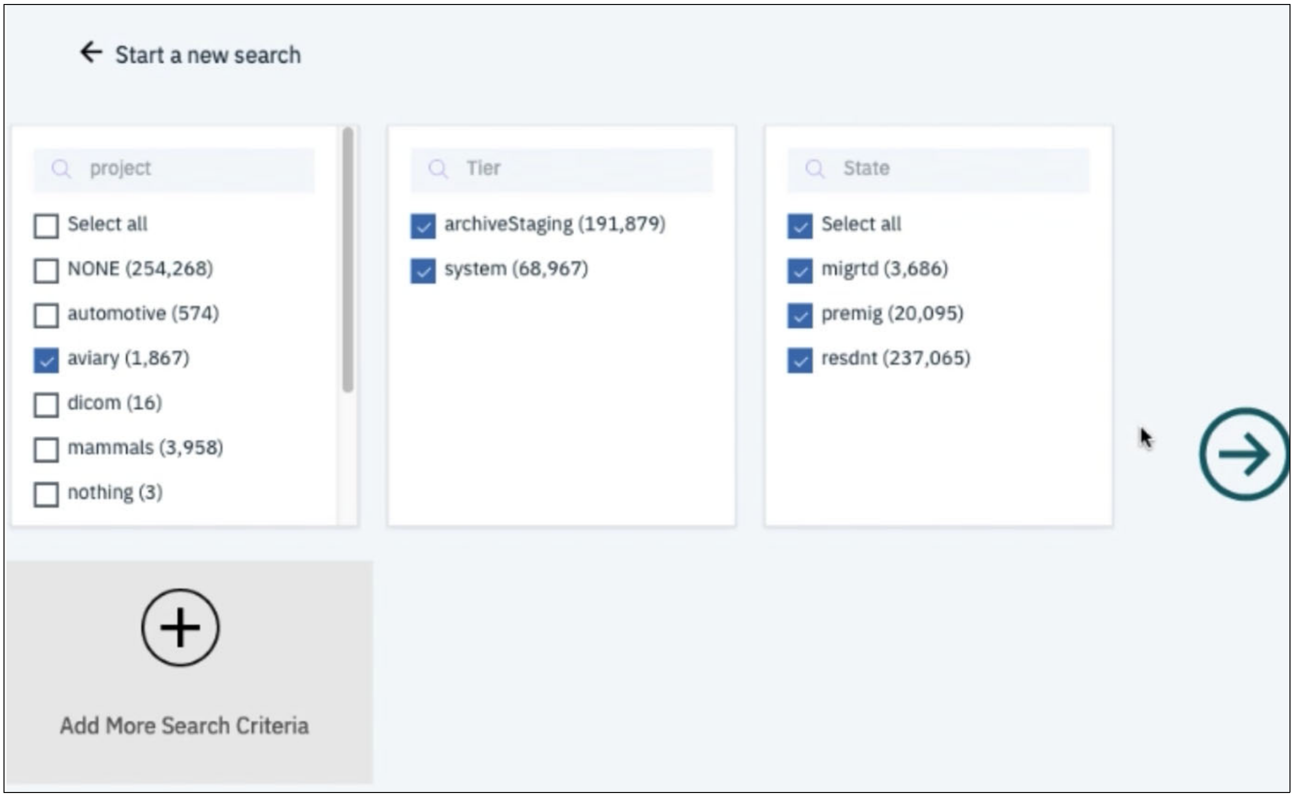


```
#  eeadm tape  list
Tape ID   Status State      Usable(GiB)  Used(GiB)  Avail(GiB)  Reclaimable%  Pool      Library  Location
JAG514JD  ok     unassigned           0          0           0            0%  -         lib1     ieslot
MB0379JE  ok     unassigned           0          0           0            0%  -         lib1     homeslot
MB0380JE  ok     appendable       18147       1028       17118            0%  pocpool1  lib1     homeslot
MB0381JE  ok     appendable       18147       1028       17118            0%  pocpool2  lib1     homeslot
MB0382JE  ok     appendable       18147          0       18146            0%  pool1     lib1     homeslot
MB0383JE  ok     appendable       18147         25       18121            0%  pool1     lib1     drive
```

Figure 5-29   IBM Spectrum Archive external tape pool

The following procedure is used in this data migration to tape use case:

1. In the search engine in IBM Spectrum Discover (as described in 5.6.1, “Data Governance use case: Data staging for high-performance processing” on page 116), select **aviary** in the **project** tag, all tiers in **Tier** tag, and states in **State** tag. Click the right-pointing arrow to continue (see Figure 5-30).



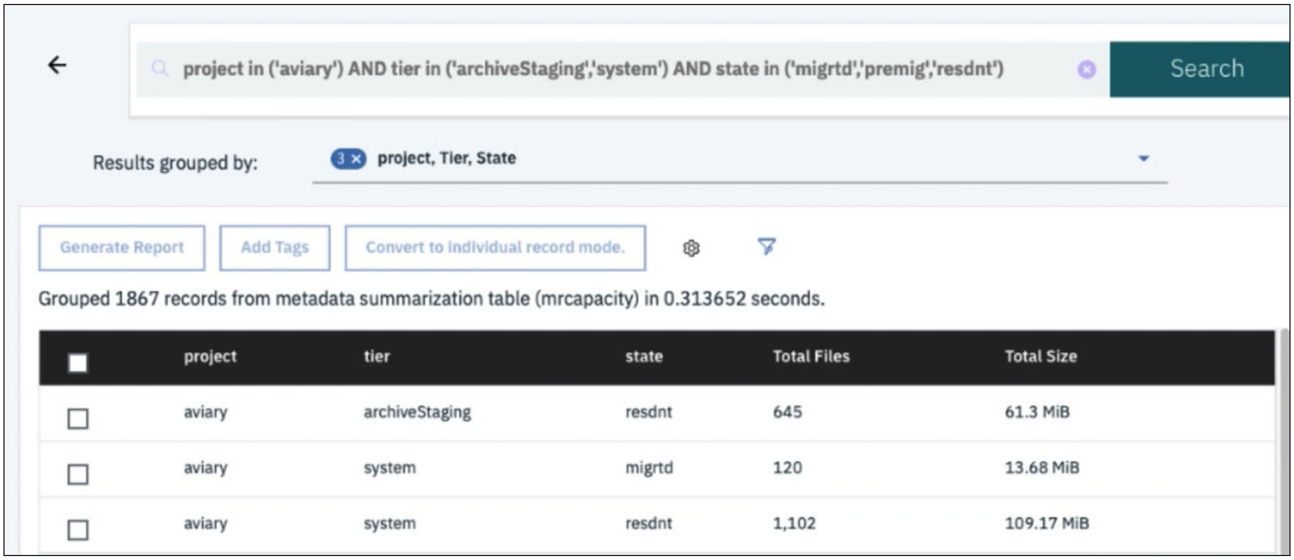
□ Select all
□ NONE (254,268)
□ automotive (574)
☑ aviary (1,867)
□ dicom (16)
□ mammals (3,958)
□ nothing (3)

☑ archiveStaging (191,879)
☑ system (68,967)

☑ Select all
☑ migrtd (3,686)
☑ premig (20,095)
☑ resdnt (237,065)

Figure 5-30   Search for aviary data set in IBM Spectrum Discover

2. As shown in Figure 5-31, the search result shows the images in the `aviary` data set are on different tiers and states. Some of the images were migrated to tape, as indicated by the `migrtd` state.



project in ('aviary') AND tier in ('archiveStaging','system') AND state in ('migrtd','premig','resdnt')

Results grouped by: 3 × project, Tier, State

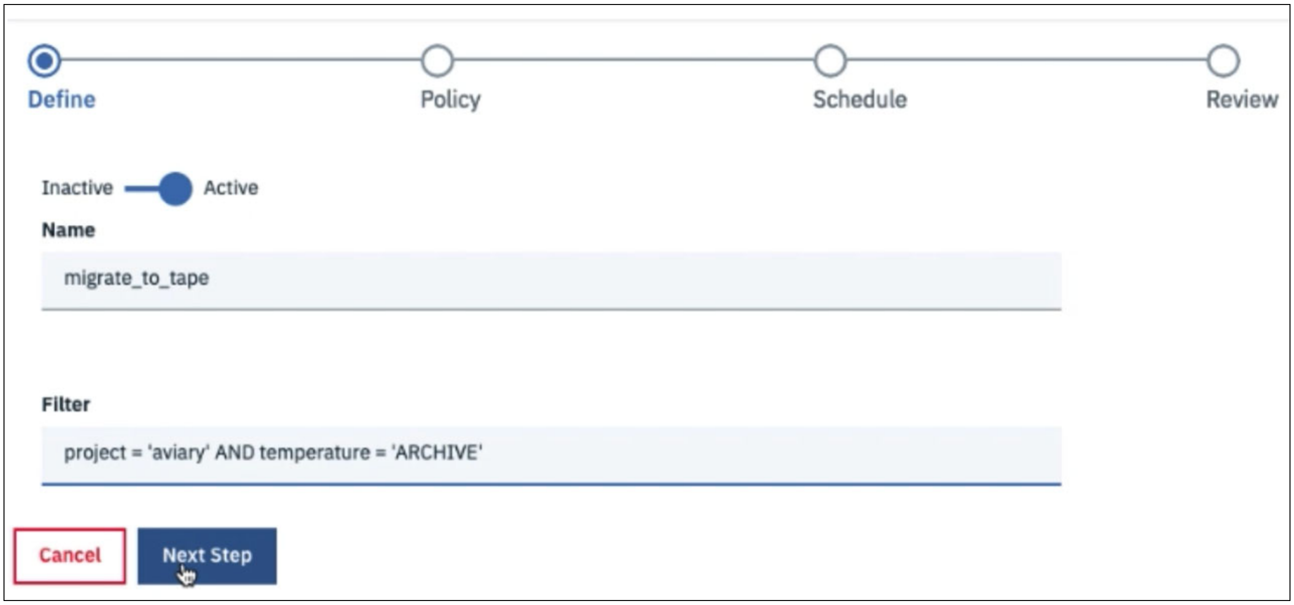
Grouped 1867 records from metadata summarization table (mrcapacity) in 0.313652 seconds.

■	project	tier	state	Total Files	Total Size
□	aviary	archiveStaging	resdnt	645	61.3 MiB
□	aviary	system	migrtd	120	13.68 MiB
□	aviary	system	resdnt	1,102	109.17 MiB

Figure 5-31 Aviary data set summary in IBM Spectrum Discover

3. In the **Management Policies** tab, activate the Policy Creation wizard to add a policy. In the **Define** page, switch the policy to **Active**, and enter a name for the policy in the **Name** field. Then, in the **Filter** field, set the **project** tag to **aviary** and narrow the policy-applied scope to images that feature the **temperature** tag with the value of **ARCHIVE**, which means only the cold data in the **aviary** data set is moved to tape. Click **Next Step**, as shown in Figure 5-32 on page 127.

> **Note:** The temperature tag is one of the pre-defined but customizable tags in IBM Spectrum Discover to help administrators to categorize data with its access frequency. For more information about how to work with the temperature tag, see IBM Knowledge Center.
>
> In addition, more filters can be applied to narrow the data scope that is to be migrated. For more examples of the use of search filters, see IBM Knowledge Center.



Inactive ● Active

Name

migrate_to_tape

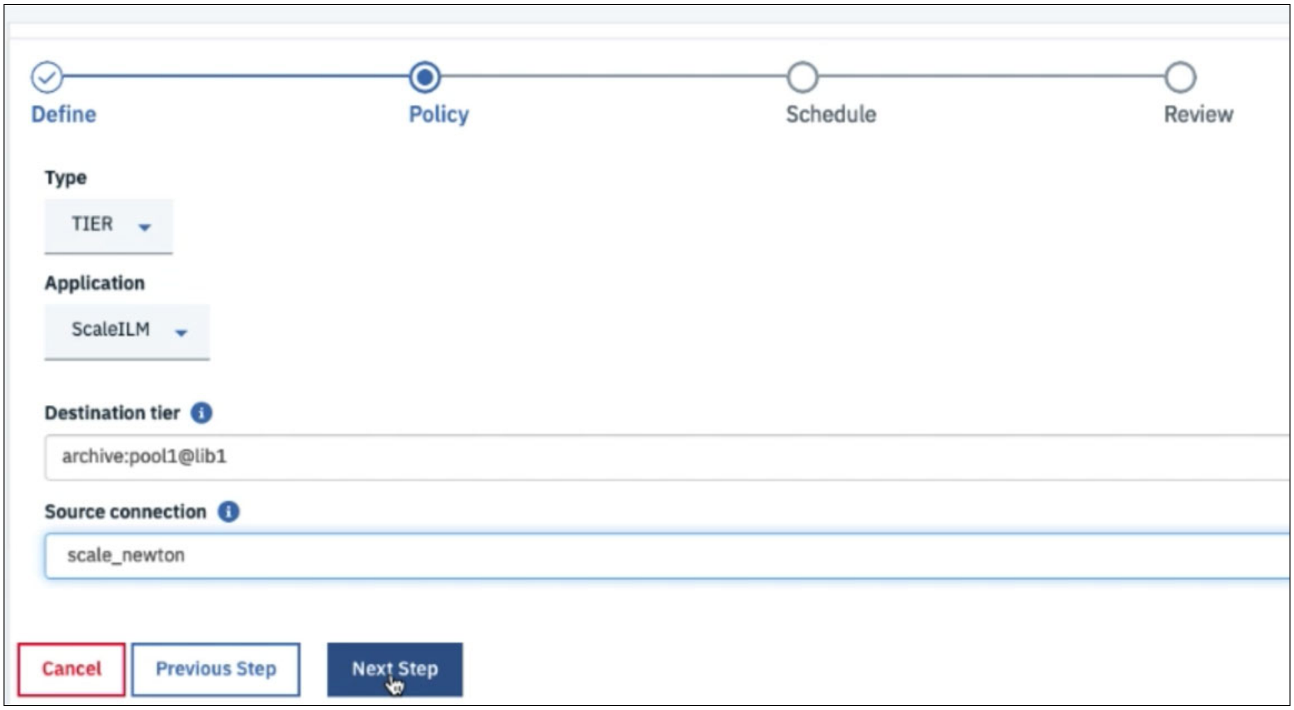
Filter

project = 'aviary' AND temperature = 'ARCHIVE'

Figure 5-32 Define Management Policy for data archiving

4. In the next **Policy** page, select **TIER** in **Type** field, and select **ScaleILM** in the **Application** field, as you did in the previous use case. In the **Destination tier** field, to declare the action of archiving to tape, the value that is filled in must follow the syntax of "`archive:<poolName>@<libraryName>`".

   As described at the beginning of this use case, the pool name and library name that are used must match the output of `eeadm` on the IBM Spectrum Archive nodes. Then, complete the **Source connection** field with the correct data source, which is `scale_newton` in our example. Click **Next Step**, as shown in Figure 5-33.



Type

TIER

Application

ScaleILM

Destination tier

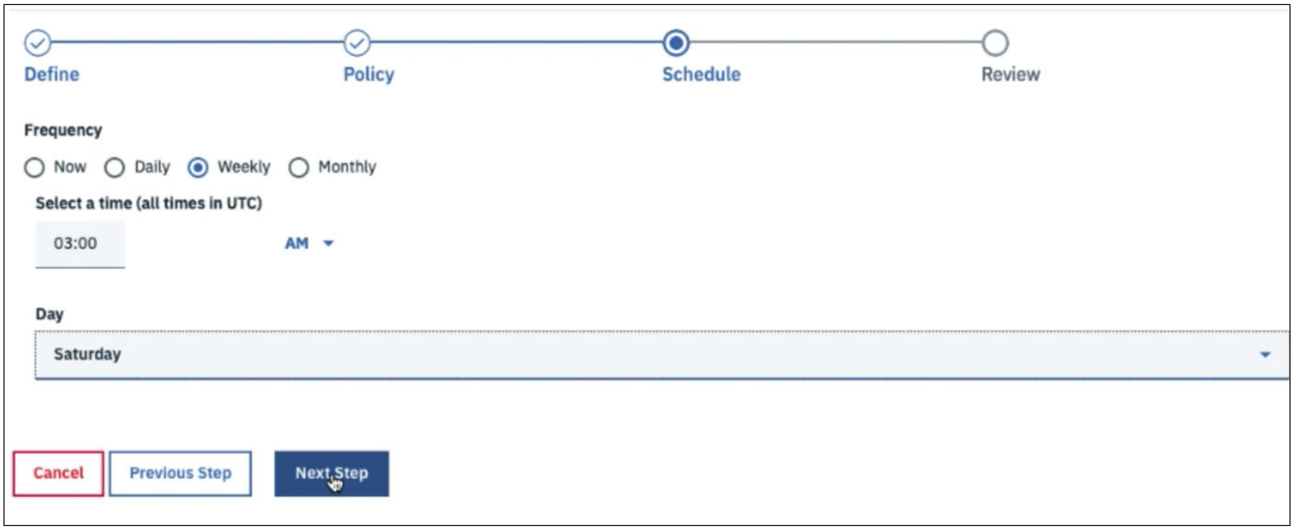
archive:pool1@lib1

Source connection

scale_newton

Figure 5-33 Fill Management Policy required information for data achieving

5. In the next **Schedule** page, set the frequency for this policy to run weekly at 3 AM on Saturday, as an example. Click **Next Step**, as shown in Figure 5-34.



Frequency

○ Now ○ Daily ● Weekly ○ Monthly

Select a time (all times in UTC)

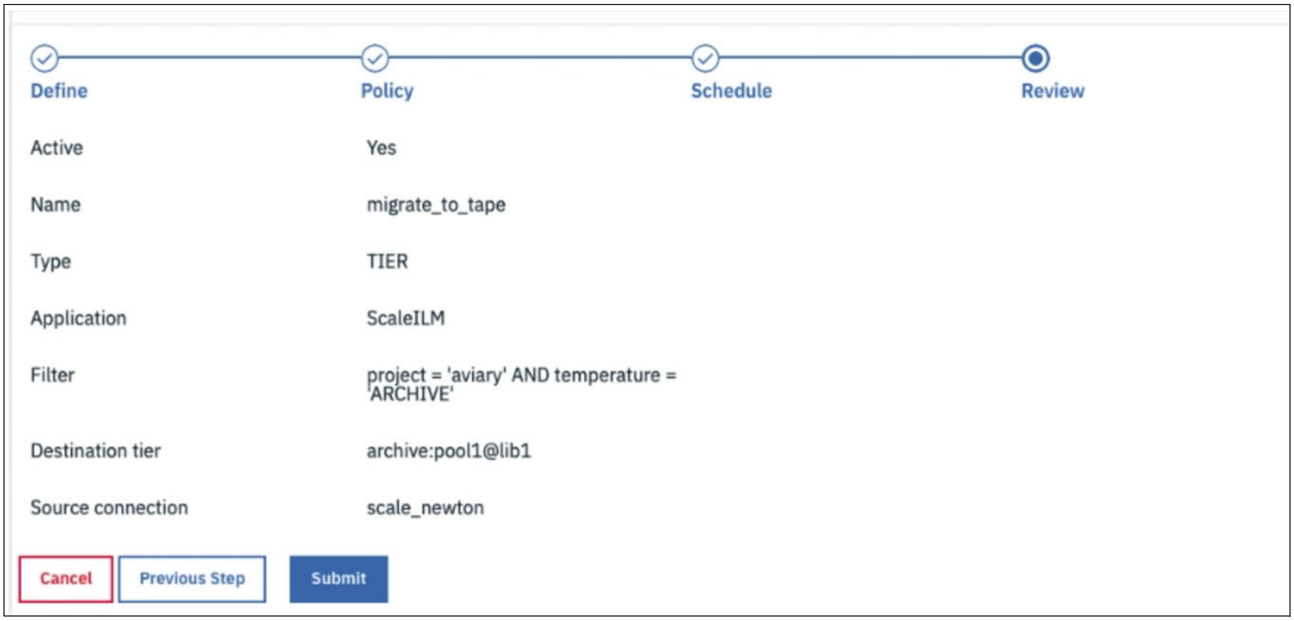
03:00 AM

Day

Saturday

Figure 5-34 Schedule Management Policy for data

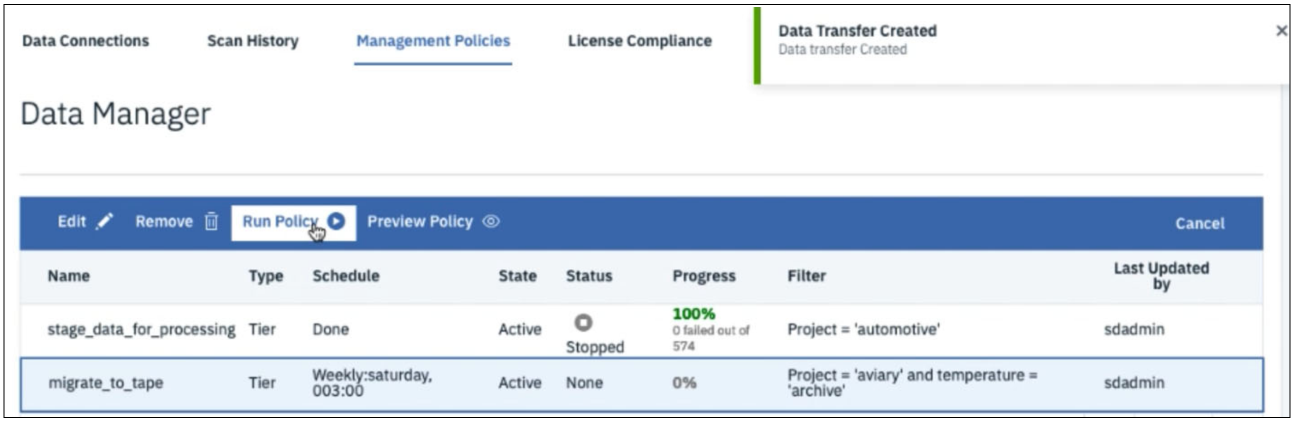
6. Review the policy setting in the **Review** page and click **Submit** (see Figure 5-35).



Active	Yes
Name	migrate_to_tape
Type	TIER
Application	ScaleILM
Filter	project = 'aviary' AND temperature = 'ARCHIVE'
Destination tier	archive:pool1@lib1
Source connection	scale_newton

Figure 5-35 Submitting the policy

7. A ScaleILM management policy is created with the parameters set in the previous steps and scheduled to run weekly. To start the policy, click **Run Policy**, as shown in Figure 5-36 on page 129.



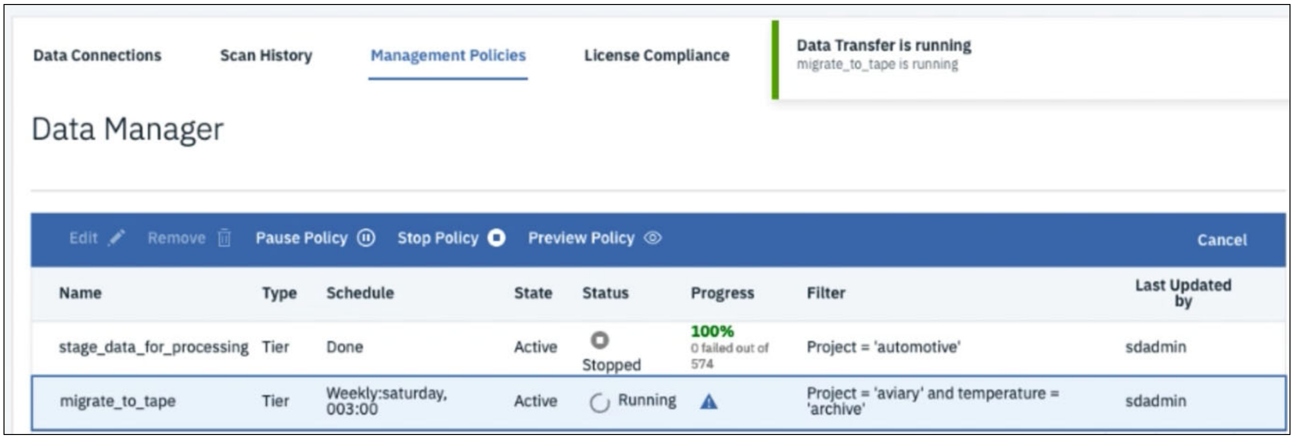
Data Transfer Created
Data transfer Created

Data Manager

Name	Type	Schedule	State	Status	Progress	Filter	Last Updated by
stage_data_for_processing	Tier	Done	Active	Stopped	100% 0 failed out of 574	Project = 'automotive'	sdadmin
migrate_to_tape	Tier	Weekly:saturday, 003:00	Active	None	0%	Project = 'aviary' and temperature = 'archive'	sdadmin

Figure 5-36 ScaleILM Management Policy created for data archiving

The data archiving management policy runs and the selected data is migrated to tape in the designated tape library, as shown in Figure 5-37.



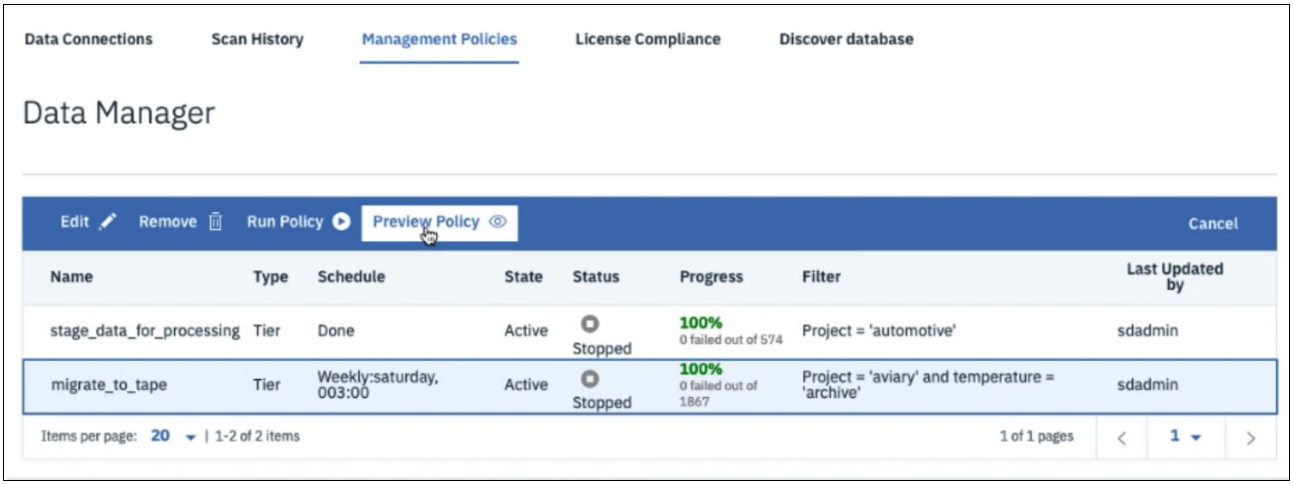
Data Transfer is running
migrate_to_tape is running

Data Manager

Name	Type	Schedule	State	Status	Progress	Filter	Last Updated by
stage_data_for_processing	Tier	Done	Active	Stopped	100% 0 failed out of 574	Project = 'automotive'	sdadmin
migrate_to_tape	Tier	Weekly:saturday, 003:00	Active	Running	⚠	Project = 'aviary' and temperature = 'archive'	sdadmin

Figure 5-37 ScaleILM Management Policy is running for data archiving

The data archiving policy process completes successfully, as shown in Figure 5-38.



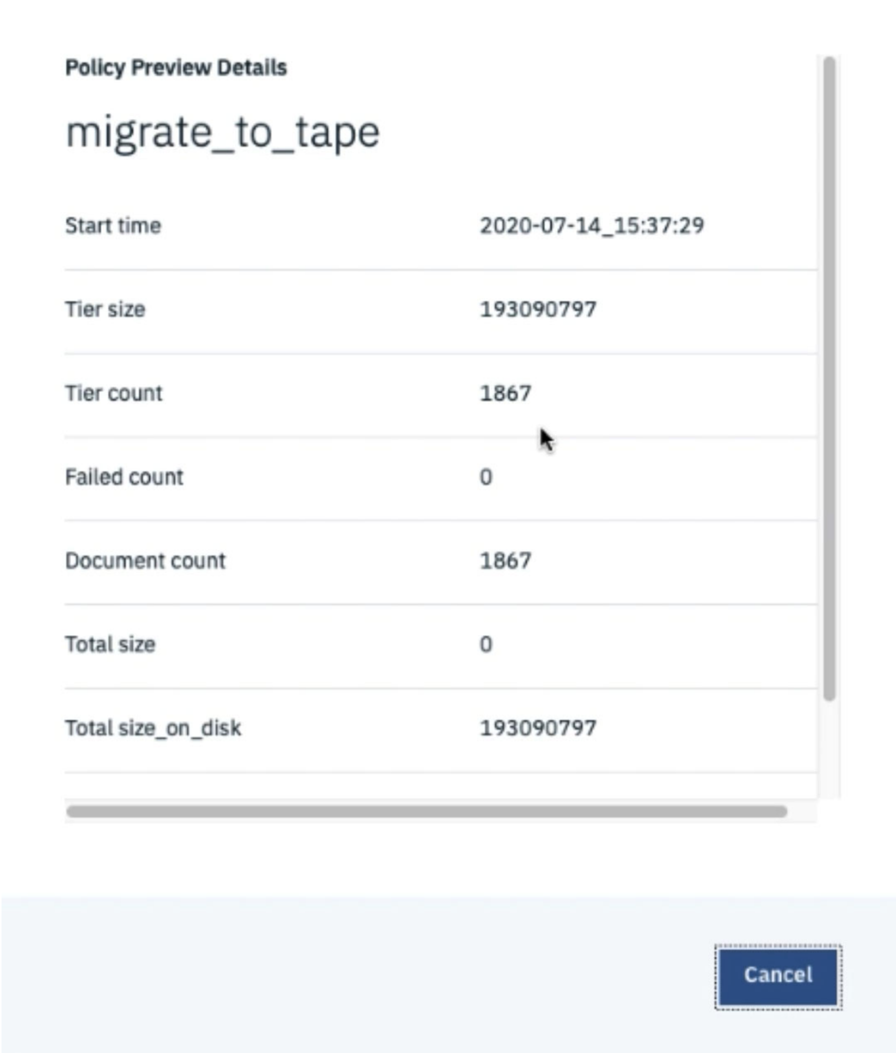
Data Manager

Name	Type	Schedule	State	Status	Progress	Filter	Last Updated by
stage_data_for_processing	Tier	Done	Active	Stopped	100% 0 failed out of 574	Project = 'automotive'	sdadmin
migrate_to_tape	Tier	Weekly:saturday, 003:00	Active	Stopped	100% 0 failed out of 1867	Project = 'aviary' and temperature = 'archive'	sdadmin

Items per page: 20 | 1-2 of 2 items   1 of 1 pages 1

Figure 5-38 ScaleILM Management Policy successfully completed for data archiving

8. Click **Preview Policy** for more information about the results. As shown in Figure 5-39, total of 1867 files were identified for movement by the policy and all of the files were successful migrated.



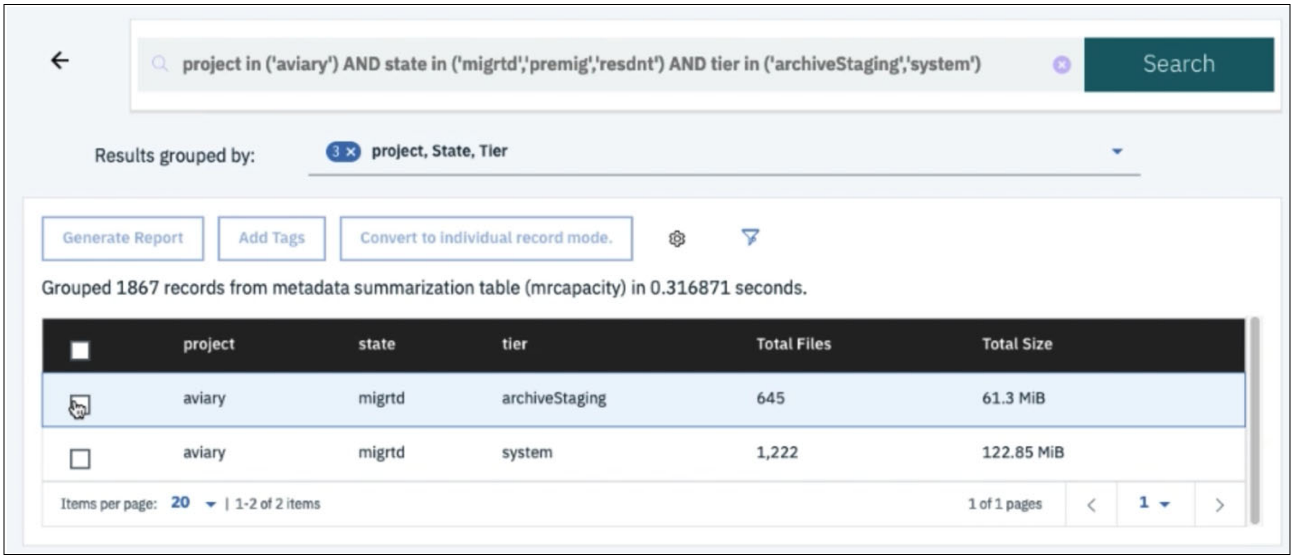
Policy Preview Details

migrate_to_tape

Start time	2020-07-14_15:37:29
Tier size	193090797
Tier count	1867
Failed count	0
Document count	1867
Total size	0
Total size_on_disk	193090797

Figure 5-39 View the policy detail of data archiving

9. To verify that the data was moved to tape, go through the search logic again in step 1 and review the updated result of the `aviary` data set. As shown in Figure 5-40 on page 131, the results indicate that all of the data in the `aviary` data set is the `migrtd` state, which shows they are all on tape. Select the records that are shown in the search result and click **Convert to individual record mode** for more information.



project in ('aviary') AND state in ('migrtd','premig','resdnt') AND tier in ('archiveStaging','system')

Results grouped by: 3 project, State, Tier

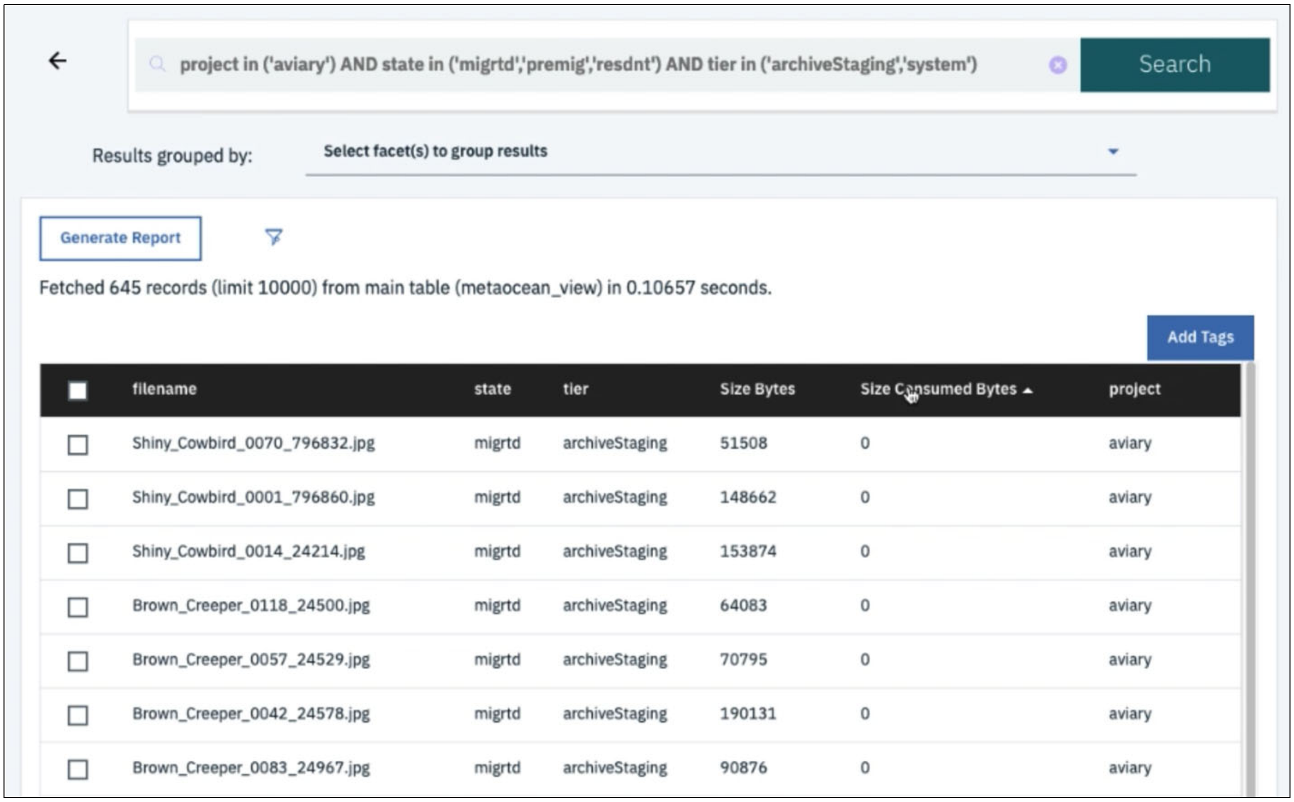
Grouped 1867 records from metadata summarization table (mrcapacity) in 0.316871 seconds.

□	project	state	tier	Total Files	Total Size
□	aviary	migrtd	archiveStaging	645	61.3 MiB
□	aviary	migrtd	system	1,222	122.85 MiB

Items per page: 20 | 1-2 of 2 items 1 of 1 pages 1

Figure 5-40 Verify the data archiving results in group mode

10. After the data records are shown in individual mode instead of group mode, click the funnel icon and add the **Size Consumed Bytes** column into the table, as shown in Figure 5-41. The value of Size Consumed Bytes column of all the records shows 0 as expected, which means with the archiving migration completed and the disk space was freed to contain more active data when applications write.



project in ('aviary') AND state in ('migrtd','premig','resdnt') AND tier in ('archiveStaging','system')

Results grouped by: Select facet(s) to group results

Fetched 645 records (limit 10000) from main table (metaocean_view) in 0.10657 seconds.

□	filename	state	tier	Size Bytes	Size Consumed Bytes ▲	project
□	Shiny_Cowbird_0070_796832.jpg	migrtd	archiveStaging	51508	0	aviary
□	Shiny_Cowbird_0001_796860.jpg	migrtd	archiveStaging	148662	0	aviary
□	Shiny_Cowbird_0014_24214.jpg	migrtd	archiveStaging	153874	0	aviary
□	Brown_Creeper_0118_24500.jpg	migrtd	archiveStaging	64083	0	aviary
□	Brown_Creeper_0057_24529.jpg	migrtd	archiveStaging	70795	0	aviary
□	Brown_Creeper_0042_24578.jpg	migrtd	archiveStaging	190131	0	aviary
□	Brown_Creeper_0083_24967.jpg	migrtd	archiveStaging	90876	0	aviary

Figure 5-41 Verify the data archiving results in individual mode

5.7 Online resources

The following online resources are available for this use case:

- Storage Accelerate with IBM Storage: Optimization with IBM Spectrum Discover video:

  https://www.youtube.com/watch?v=7INo3j4BZzk&feature=emb_title

- Accelerate with IBM Storage website

  https://www.youtube.com/watch?v=_YNfFDdMEa4

- IBM Knowledge Center resources:
  - https://www.ibm.com/support/knowledgecenter/SSY8AC
  - https://www.ibm.com/support/knowledgecenter/STXKQY/ibmspectrumscale_welcome.html
  - https://www.ibm.com/support/knowledgecenter/ST9MBR
  - https://www.ibm.com/support/knowledgecenter/en/STAKKZ/ts4300_kc/3U_kchome.html
  - https://www.ibm.com/support/knowledgecenter/STQRQ9/com.ibm.storage.ts4500.doc/ts4500_ichome.html
- IBM Spectrum Archive EE V1.3.0.6 Installation and Configuration Guide IBM Redbooks:

  https://www.redbooks.ibm.com/redbooks/pdfs/sg248333.pdf

5.8 Summary

Table 5-1 summarizes the Data Governance use case.

Table 5-1 Summary of the Data Governance use case

Use case overview	Identify required files and stage them into high-performance storage tier for the following data processing phase.
Products involved	▶ IBM Spectrum Discover ▶ IBM Spectrum Scale
Benefits	▶ Ability to identify and locate the required data sets faster and easier. ▶ Ability to move required data sets to a premier tier to improve the overall data processing performance.
High-level implementation steps	1. Search for required file sets for data processing. 2. Identify them based on policy tags. 3. Run a policy movement of the data to the high-performance storage pool by using the ScaleILM DEEPINSPECT agent. 4. Confirm that the files are in the target storage pool in IBM Spectrum Scale.

Table 5-2 summarizes the Data Optimization use case.

Table 5-2 Summary of the Data Optimization use case

Use case overview	Identify aged or “cold” files and move them into IBM Spectrum Archive and onto tape.
Products involved	► IBM Spectrum Discover ► IBM Spectrum Scale ► IBM Spectrum Archive Enterprise Edition
Benefits	► Ability to match value of the data with the cost of the storage medium. ► Ability to move older inactive data sets to an ‘active archive’ tier to reduce cost per TB.
High-level implementation steps	1. Search for aged, cold, or inactive file sets. 2. Identify them based on policy tags. 3. Run a policy movement of the data to the IBM Spectrum Archive pool by using the ScaleILM DEEPINSPECT agent. 4. Confirm that the files are now in a `migrated` state inside the IBM Spectrum Archive pool.

A

IBM Spectrum Scale, IBM Spectrum Archive, and IBM Tape libraries product details

This appendix describes the following IBM products and technologies that are used in the use cases that covered in this book:

- IBM Spectrum Scale
- IBM Spectrum Archive
- IBM Tape libraries and technology

This appendix includes the following topics:

IBM Spectrum Scale overview

IBM Spectrum Scale is a proven, scalable, high-performance file management solution. IBM Spectrum Scale provides world-class storage management with extreme scalability, flash accelerated performance, and automatic policy-based storage tiering from flash to disk, then to tape. IBM Spectrum Scale reduces storage costs up to 90% while improving security and management efficiency in cloud, big data, and analytics environments.

First introduced in 1998, this mature technology enables a maximum volume size of 8 YB, a maximum file size of 8 EB, and up to 18.4 quintillion (two to the 64th power) files per file system. IBM Spectrum Scale provides simplified data management and integrated information lifecycle tools, such as software-defined storage for cloud, big data, and analytics. It introduces enhanced security, flash accelerated performance, and improved usability. It also provides capacity quotas, access control lists (ACLs), and a powerful snapshot function.

Key capabilities

IBM Spectrum Scale adds elasticity with the following capabilities:

- Global name space with high-performance access scales from departmental to global.
- Automated tiering, data lifecycle management from flash (6x acceleration) to tape (10 times savings)
- Enterprise ready with data security (encryption), availability, reliability, and large scale
- POSIX compliant and supports containers
- Integrated with OpenStack Swift. and Hadoop

Benefits

IBM Spectrum Scale provides the following benefits:

- Improves performance by removing data-related bottlenecks.
- Automated tiering, data lifecycle management from flash (acceleration) to tape (savings).
- Enables sharing of data across multiple applications.
- Reduces cost per performance by placing data on most applicable storage (flash to tape or cloud).

IBM Spectrum Scale is part of the IBM market-leading software-defined storage family. Consider the following points:

- As a software-only solution: Runs on virtually any hardware platform and supports almost any block storage device. IBM Spectrum Scale runs on Linux (including Linux on IBM Z®), IBM AIX, and Windows systems.
- As an integrated IBM Elastic Storage System: A bundled hardware, software, and services offering that includes installation and ease of management with a graphical user interface. Elastic Storage System provides unsurpassed end-to-end data availability, reliability, and integrity with unique technologies that include IBM Spectrum Scale RAID.
- As a cloud service: IBM Spectrum Scale delivered as a service provides high-performance, scalable storage, and integrated data governance for managing large amounts of data and files in the IBM Cloud.

- IBM Spectrum Scale features enhanced security with native encryption and secure erase. It can increase performance by using server-side flash cache to increase I/O performance up to six times.
- IBM Spectrum Scale provides improved usability through data replication capabilities, data migration capabilities, Active File Management (AFM), transparent cloud tiering (TCT), IBM Spectrum Scale Native RAID, and Erasure Code Edition (ECE).

An example of the IBM Spectrum Scale architecture is shown in Figure A-1.

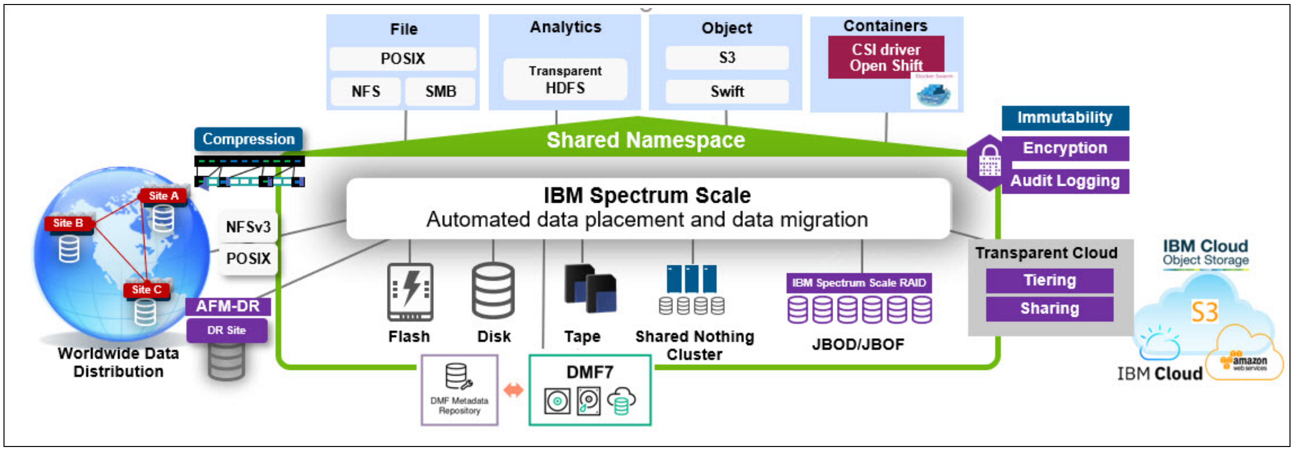


Figure A-1 IBM Spectrum Scale architecture

Three key concepts in IBM Spectrum Scale, which are of interest in the IBM Spectrum Discover integration with IBM Spectrum Archive Enterprise Edition use case are:

- Storage pools
- Policy engine
- Mirroring, replication, and migration capabilities

Storage pools

A storage pool is a collection of disks or arrays with similar attributes. It is an organizational structure that allows the combination of multiple storage locations with identical characteristics. Three types of pools are available, System, Data, and External.

Policy engine

The policy engine uses an SQL style syntax to query or operate on files based on file attributes. Policies can be used to migrate all data that was accessed for 6 months (for example) to less expensive storage or used to query the contents of a file system.

Management policies support advanced query capabilities, although what makes the policy engine most useful is the performance. The policy engine can scan billions of objects. Although an average `find` across 1 billion files took ~ 47 hours, the IBM Spectrum Scale policy engine can satisfy the request within 5 hours (assuming that the metadata is on SSDs or such.)

Mirroring, replication, and migration capabilities

In IBM Spectrum Scale, you can replicate a single file, a set of files, or the entire file system. You can also change the replication status of a file at any time by using a policy or running a command. By using these capabilities, you can achieve a replication factor of two, which equals mirroring, or a replication factor of three.

For migration, IBM Spectrum Scale provides the capability to add storage to the file system, migrate the data to the new storage, and remove the old storage from the file system. All of this can be done online without disruption to your business.

Active File Management

Active File Management (AFM) enables the sharing of data across unreliable or high latency networks. With AFM, you can create associations between IBM Spectrum Scale clusters and define the location and flow of file data. AFM allows you to implement a single name space view across clusters, between buildings, and around the world.

Transparent cloud tiering

Transparent cloud tiering is a software-defined capability that enables the use of public, private, and on-premises cloud Object Storage as a secure, reliable, and transparent storage tier that is natively integrated with IBM Spectrum Scale without introducing hardware appliances or new management touch points. It uses the ILM policy language semantics that are available in IBM Spectrum Scale. The semantics allow administrators to define the following policies for tiering cooler and cold data to a cloud Object Storage:

- IBM Cloud Object Storage
- Amazon Web Services S3
- OpenStack Swift

For more information, see *Enabling Hybrid Cloud Storage for IBM Spectrum Scale Using Transparent Cloud Tiering*, REDP-5411.

IBM Spectrum Archive overview

A member of the IBM Spectrum Storage family, IBM Spectrum Archive enables direct, intuitive, and graphical access to data that is stored in IBM tape drives and libraries by incorporating the IBM Linear Tape File System (LTFS) format standard for reading, writing, and exchanging descriptive metadata on formatted tape cartridges. IBM Spectrum Archive eliminates the need for extra tape management and software to access data.

IBM Spectrum Archive offers the following software solutions for managing your digital files with the LTFS format:

- IBM Spectrum Archive Single Drive Edition (SDE)
- IBM Spectrum Archive Library Edition (LE)
- IBM Spectrum Archive Enterprise Edition (EE)

With IBM Spectrum Archive Enterprise Edition and IBM Spectrum Scale, tape can now add savings as a low-cost storage tape tier. The use of a tier of tape for active but “cold” data enables enterprises to look at new ways to optimize the cost of their unstructured data storage.

This also allows storage administrators to match the value of the data, or the value of the copies of data to the most appropriate storage media.

Key capabilities

IBM Spectrum Archive options support small, medium, and enterprise businesses with the following advantages:

- Seamless virtualization of storage tiers
- Policy-based placement of data
- Single universal name space for all file data
- Security and protection of assets
- Open, non-proprietary, cross platform interchange
- Integrated functions with IBM Spectrum Scale

Benefits

IBM Spectrum Archive enables direct, intuitive, and graphical access to data that is stored in IBM tape drives and libraries by incorporating the LTFS format standard for reading, writing, and exchanging descriptive metadata on formatted tape cartridges. IBM Spectrum Archive eliminates the need for more tape management and software to access data.

IBM Spectrum Archive takes advantage of the low cost of tape storage while making it easy to use. IBM Spectrum Archive provides the following benefits:

- Easily access and manage all data in stand-alone tape environments as though it were on disk
- Enable easy-as-disk access to single or multiple cartridges in a tape library
- Improve efficiency and reduce costs for long-term, tiered storage
- Optimize data placement for cost and performance
- Enable data file sharing without proprietary software
- Scalable and low cost

Linear Tape File System

IBM developed LTFS and then contributed it to SNIA as an open standard so that all tape vendors can participate. LTFS is the first file system that works with Linear Tape-Open (LTO) generation 8, 7, 6, and 5 tape technology (or IBM TS1160, TS1155, TS1150, and TS1140 tape drives) to set a new standard for ease of use and portability for open systems tape storage.

With this application, accessing data that is stored on an IBM tape cartridge is as easy and intuitive as using a USB flash drive. Tapes are self-describing, and you can quickly recall any file from a tape without having to read the entire tape from beginning to end.

Also, any LTFS-capable system can read a tape that is created by any other LTFS-capable system (regardless of the operating system and platform). Any LTFS-capable system can identify and retrieve the files that are stored on it. LTFS-capable systems have the following characteristics:

- Files and directories are displayed to you as a directory tree listing.
- More intuitive searches of cartridge and library content are now possible because of the addition of file tagging.
- Files can be moved to and from LTFS tape by using the familiar drag-and-drop metaphor common to many operating systems.

- Many applications that were written to use files on disk can now use files on tape without any modification.
- All standard File Open, Write, Read, Append, Delete, and Close functions are supported.

IBM Spectrum Archive Editions

IBM Spectrum Archive is available in different editions that support small, medium, and enterprise businesses, as described next

IBM Spectrum Scale Single Drive Edition

The IBM Spectrum Archive Single Drive Edition implements the LTFS Format and allows tapes to be formatted as LTFS Volumes. These LTFS Volumes can then be mounted by using LTFS to allow users and applications direct access to files and directories that are stored on the tape. No integration with tape libraries exists in this edition. You can access and manage all data in stand-alone tape environments as though it were on disk.

IBM Spectrum Archive Library Edition

IBM Spectrum Archive Library Edition extends the file management capability of the IBM Spectrum Archive SDE. It enables easy-as-disk access to single or multiple cartridges in a tape library. The IBM Spectrum Archive LE software automatically controls the tape library robotics to load and unload the necessary LTFS Volumes to provide access to the stored files.

IBM Spectrum Archive LE enables the reading, writing, searching, and indexing of user data on tape and access to user metadata. IBM Spectrum Archive LE supports Linux and Windows. Each LTFS tape cartridge in the library appears as an individual folder within the file space. The user or application can browse to these folders to access the files that are stored on each tape.

IBM Spectrum Archive Enterprise Edition

IBM Spectrum Archive Enterprise Edition (EE) gives organizations an easy way to use cost-effective IBM tape drives and libraries within a tiered storage infrastructure. By using tape libraries instead of disks for Tier 2 and Tier 3 data storage (data that is stored for long-term retention), organizations can improve efficiency and reduce costs.

In addition, IBM Spectrum Archive EE seamlessly integrates with the scalability, manageability, and performance of IBM Spectrum Scale, which is an IBM enterprise file management platform that enables organizations to move from adding storage to optimizing data management.

IBM Spectrum Archive EE includes the following highlights:

- Simplify tape storage with the IBM LTFS format, which is combined with the scalability, manageability, and performance of IBM Spectrum Scale
- Help reduce IT expenses by replacing tiered disk storage (Tier 2 and Tier 3) with IBM tape libraries
- Expand archive capacity by adding and provisioning media without affecting the availability of data that is in the pool
- Add extensive capacity to IBM Spectrum Scale installations with lower media, floor space, and power costs
- Support for attaching up to two tape libraries to a single IBM Spectrum Scale cluster

IBM Spectrum Archive EE for the IBM TS4500, IBM TS4300, IBM TS3500, and IBM TS3310 tape libraries provide seamless integration of IBM Spectrum Archive with Spectrum Scale by creating an LTFS tape tier that can be controlled by the ILM policy management engine in IBM Spectrum Scale. You can run any application that is designed for disk files on tape by using IBM Spectrum Archive EE.

The integration of IBM Spectrum Archive EE archive solution with IBM Spectrum Scale is shown in Figure A-2.

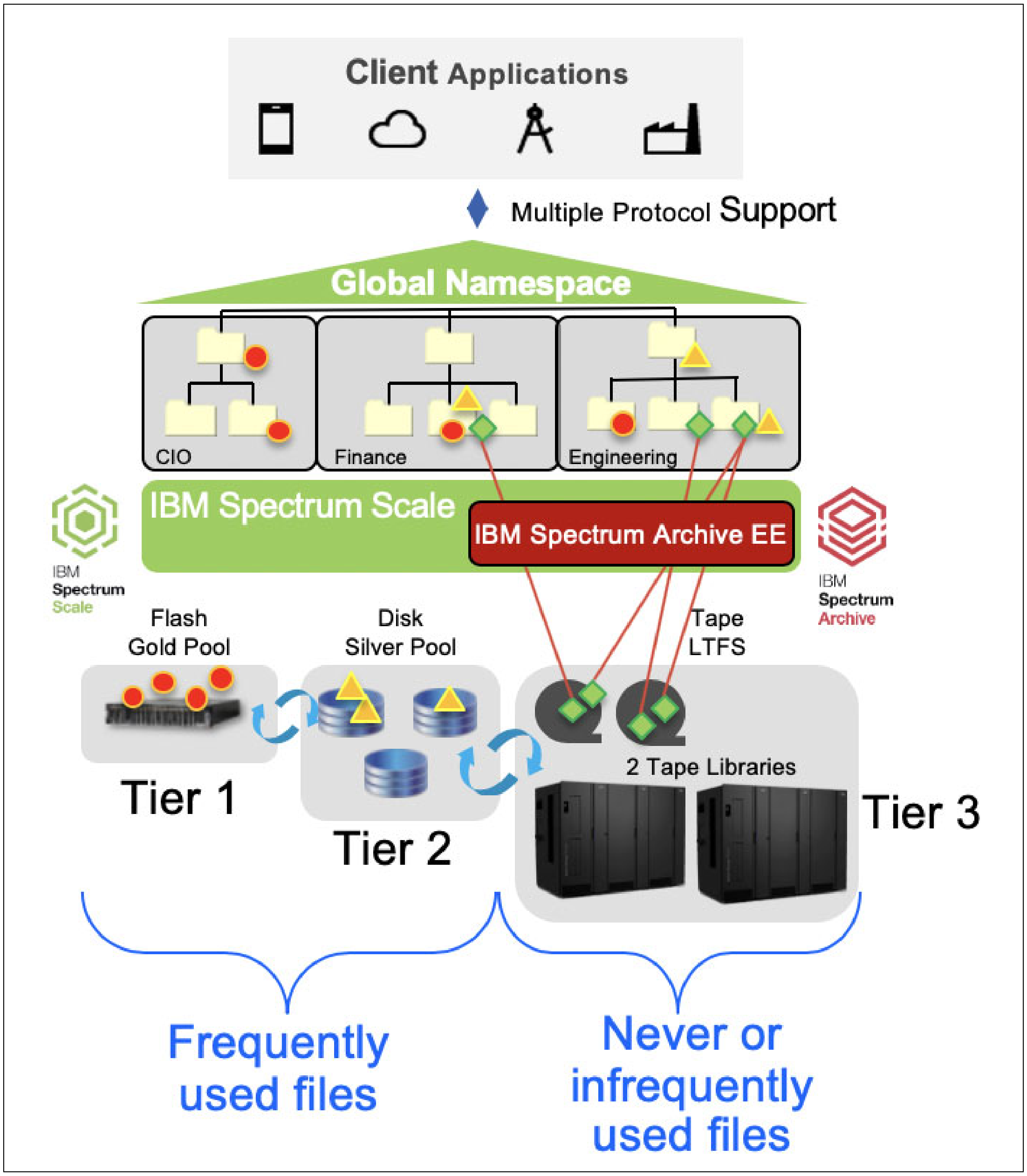


Figure A-2 Integration of IBM Spectrum Scale and IBM Spectrum Archive EE

The seamless integration offers transparent file access in a continuous name space. It provides file level write and read caching with disk staging area, policy-based movement from disk to tape, creation of multiple data copies on different tapes, load balancing, and high availability in multi-node clusters.

It also offers data exchange on LTFS tape by using import and export functions, fast import of file name space from LTFS tapes without reading data, built-in tape reclamation and reconciliation, and simple administration and management.

For more information, see IBM Knowledge Center.

OpenStack, SwiftHLM, and IBM Spectrum Archive

IBM Spectrum Archive Enterprise Edition can also be used to provide Object Storage by using OpenStack Swift. By using this configuration, objects can be stored in the file system and exist on disk or tape tiers within the enterprise.

For more information about creating an Object Storage Active Archive with IBM Spectrum Scale and IBM Spectrum Archive, see *Active Archive Implementation Guide with IBM Spectrum Scale Object and IBM Spectrum Archive*, REDP-5237.

States of data inside IBM Spectrum Archive

Within IBM Spectrum Archive EE, the following states of data can exist (see Figure A-3 on page 143):

- Resident: Data is on disk only. This state is the initial state of files.
- Premigrated: Data is on disk and tape, which enables rapid access to the data on disk, while redundant data is available on tape.
- Migrated: Data is on tape only, which is the most economical form of storage.

As shown in Figure A-3 on page 143, files can be moved between the states individually or in bulk. Based on the policies that are in place, files can automatically return to tape after they are accessed or remain on disk for a predetermined amount of time.

> **Tip:** IBM Spectrum Archive also includes the capability to create multiple copies on tape and maintain up to three copies of the data, with one being on flash or disk, and two being on tape potentially in two different tape libraries.

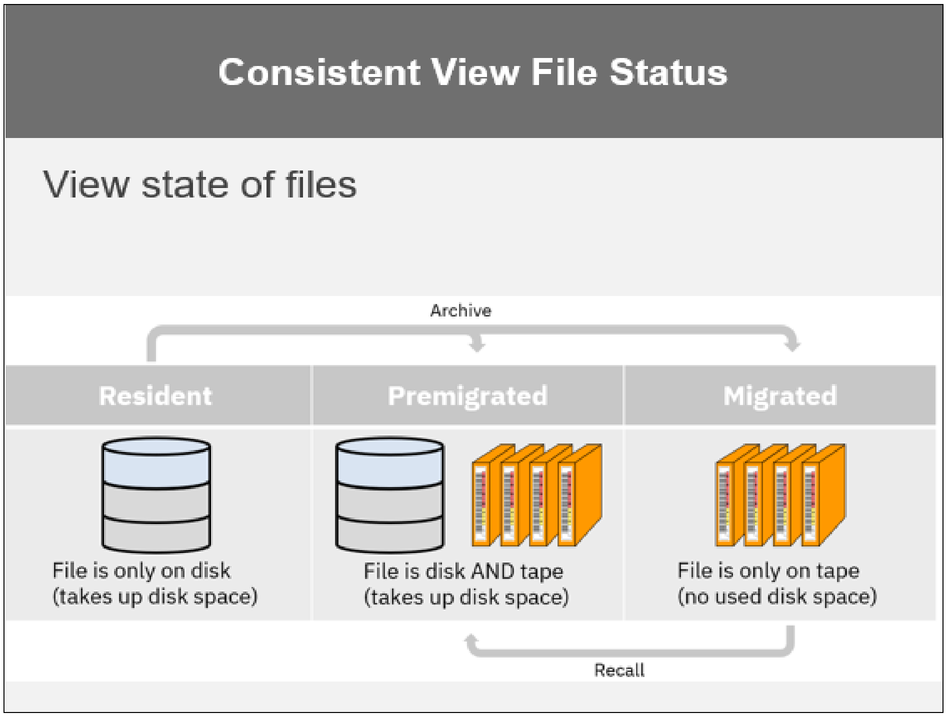


Figure A-3 Different states of data in a IBM Spectrum Archive

IBM tape technologies overview

Tape storage provides reliable data protection and long-term retention. With more than 60 years of tape innovation, IBM continues to push the limits of capacity and the boundaries of tape technology innovation. It is scalable, durable, secure, and energy-efficient at a compellingly lower cost than other media.

Automated tape libraries

Tape is receiving a resurgence of interest as many leading IT organizations recognize the unique features of the technology with the added benefit of low cost of ownership.

Moreover, cloud providers use tape as part of their infrastructure to offer low-cost, reliable archive solutions.

This section describes the latest IBM enterprise tape libraries that provide long-term data storage with the lowest cost footprint.

IBM TS4500 tape library

The IBM TS4500 tape library is a next-generation storage solution that is designed to help midsize and large enterprises respond to storage challenges. Among these challenges are high data volumes and the growth in data centers. These factors in turn increase the cost of data center storage footprints, the difficulty of migrating data across vendor platforms, and the complexity of IT training and management as staff resources shrink.

In the TS4500, IBM delivers the density that today’s and tomorrow’s data growth requires, along with the cost efficiency and the manageability to grow with business data needs while preserving existing investments in IBM tape library products.

You can now achieve a low cost per terabyte (TB) and a high TB density per square foot. The TS4500 can store up to 8.76 PBs of uncompressed data in a single 10-square foot library frame by using LTO 8 cartridges or 11 PBs with 3592 cartridges. The number of frames per library is one base frame and up to 17 expansion frames. The storage capacity with 3592 advanced cartridges is up to 351 PB per library (1.05 EBs with 3:1 compression) and the capacity with LTO Ultrium eight cartridges is up to 278 PBs per library (up to 695 PB with 2.5:1 compression).

The TS4500 tape library includes the following highlights:

- Improve storage density with more than two times the expansion frame capacity and support for 33% more tape drives.
- Proactively monitor archived data with policy-based automatic media verification.
- Improve business continuity and disaster recovery with automatic control path and data path failover.
- Help ensure security and regulatory compliance with tape-drive encryption and Write Once Read Many (WORM) media.
- Support Linear Tape-Open (LTO) Ultrium 8, LTO Ultrium 7, LTO Ultrium 6, LTO Ultrium 5, and IBM TS1160, TS1155, TS1150, and TS1140 tape drives.
- Increase mount performance and overall system availability with dual robotic accessors.
- Provide a flexible upgrade path for users who want to expand their tape storage as their needs grow.
- Reduce the storage footprint and simplify cabling with 10U of rack space on top of the library.

For more information, see the *IBM TS4500 R7 Tape Library Guide,* SG24-8235.

IBM TS4300 tape library

IBM TS4300 tape library is a high-density, highly scalable, easy-to-manage solution that is designed to keep data securely stored long term, while helping to reduce the costs that are associated with data center space and utilities. Its modular design enables you to increase cartridge and drive capacity as needed; scale vertically up to seven modules with expansion for Linear Tape-Open (LTO) Ultrium cartridges, drives, and redundant power supplies. IBM TS4300 enhanced data protection helps meet security and compliance requirements.

Modular design for scalability

The modular design enables users to increase cartridge and drive capacity as needed. Built around a 3U-high, modular base library, TS4300 can scale vertically with up to seven modules, with expansion for Linear Tape-Open (LTO) Ultrium 8, 7, and 6 cartridges, drives, and redundant power supplies. A single robot manages all modules in the stack. In a seven-module configuration, TS4300 offers a capacity of more than 270 LTO cartridges.

Security and compliance capabilities

Supports WORM cartridges to enhance data protection and provides library-managed encryption with IBM Security™ Key Lifecycle Manager.

Base library for complete management of system

The TS4300 base library contains all of the necessary robotics and intelligence to manage the base library system, with a maximum of 40 slots of LTO cartridge capacity per module, five input/output (I/O) slots, and support for three combinations of full- or half-height (FH or HH) LTO drives per module. The TS4300 base library is rack or tabletop mountable. Module expansions are rack mountable only.

Fast deployment and simplified management

The embedded, open source software architecture helps speed deployment and simplify management. Best-in-class error recovery and reporting, library health monitoring and alerting, and auto-recovery features help to speed diagnostics and resolution. An automation drive interface significantly increases drive communication speed for faster code updates and log downloads.

For more information, see *IBM Tape Library Guide for Open Systems*, SG24-5946.

Tape drives

IBM tape drives deliver performance and capacity for midrange to high-end storage for low cost, long-term storage needs.

IBM TS1100 tape drive family

The IBM TS1160 (Machine type 3592-60G), TS1155 (Machine Type 3592-55G), TS1150 (Machine Type 3592-E08), and TS1140 (Machine Type 3592-E07) offer a design that is focused on high capacity, performance, and high reliability for storing mission-critical data. With the TS1100 family of tape drives, IBM advanced its high-end, half-inch cartridge tape technology to a new level.

The 3592 family was enlarged and improved with the addition of the IBM TS1160, 3592 Model 60G tape drives. The TS1160 is the sixth generation of the 3592 tape drive family. It provides the unprecedented capacity of 20 TB of uncompressed data on a single tape and new physical connection options.

IBM TS1000 Linear Tape-Open Ultrium tape drives

Linear Tape-Open (LTO) Ultrium tape drives are in their eighth generation. The newest of these drives, the IBM TS1080 tape drive Model F8A, is a dual-ported drive that facilitates 8 Gbps Fibre Channel connectivity. It features a native data transfer of up to 360 MBps for FH tape drives.

The TS1080 Tape Drive supports the LTO generation 8 media capacity specification up to 30 TB with a 2.5:1 compression ratio (up to 12 TB native capacity), which is twice the compressed capacity of the previous version.

For more information about LTO technology, see this website.

B

Additional material

This book refers to additional material that can be downloaded from the internet as described in the following sections.

Locating the GitHub material

The web material that is associated with this book is available in softcopy on the internet from the IBM Redbooks GitHub location:

https://github.com/IBMRedbooks/SG248448-Making-Data-Smarter-with-Spectrum-Discover-Practical-AI-Solutions

Cloning the GitHub material

Complete the following steps to clone the GitHub repository for this book:

1. Download and install `Git` client if not installed from this web page.
2. Run the following command to clone the GitHub repository:

   ```
   git clone
   https://github.com/IBMRedbooks/SG248448-Making-Data-Smarter-with-Spectrum-Disco
   ver-Practical-AI-Solutions.git.
   ```

Related publications

The publications that are listed in this section are considered particularly suitable for a more detailed discussion of the topics that are covered in this book.

IBM Redbooks

The following IBM Redbooks publications provide more information about the topic in this document. Note that some publications that are referenced in this list might be available in softcopy only:

- *Enabling Hybrid Cloud Storage for IBM Spectrum Scale Using Transparent Cloud Tiering*, REDP-5411
- *Active Archive Implementation Guide with IBM Spectrum Scale Object and IBM Spectrum Archive*, REDP-5237
- IBM Spectrum Archive Enterprise Edition V1.3.0.6 Installation and Configuration Guide. SG24-8333
- *IBM DS8000 SafeGuarded Copy,* REDP-5506

You can search for, view, download, or order these documents and other Redbooks, Redpapers, Web Docs, draft, and additional materials, at the following website:

ibm.com/redbooks

Online resources

The following websites are also relevant as further information sources:

- IBM Knowledge Center link for IBM Spectrum Discover:

  https://www.ibm.com/support/knowledgecenter/SSY8AC
- IBM Knowledge Center link for IBM TS4500 Tape Library:

  https://www.ibm.com/support/knowledgecenter/STQRQ9/com.ibm.storage.ts4500.doc/ts4500_ichome.html
- IBM Knowledge Center link for IBM TS4300 Tape Library

  https://www.ibm.com/support/knowledgecenter/en/STAKKZ/ts4300_kc/3U_kchome.html
- IBM Knowledge Center link for IBM Spectrum Archive Enterprise Edition (EE):

  https://www.ibm.com/support/knowledgecenter/ST9MBR
- IBM Knowledge Center link for IBM Spectrum Scale:

  https://www.ibm.com/support/knowledgecenter/STXKQY/ibmspectrumscale_welcome.html
- Storage for AI & IBM Power Systems Video: Medical data challenge white paper:

  https://www.ibm.com/it-infrastructure/services/client-experience-portal/offeringdetail.jsp?offid=2048&sourcenamekey=CATALOG_LOGICAL

- Medical Detection Toolkit:

  https://github.com/MIC-DKFZ/medicaldetectiontoolkit
- Watson Machine Learning Community Edition toolkit:

  https://developer.ibm.com/linuxonpower/deep-learning-powerai/releases/
- Connecting to data sources with Spectrum Discover:

  https://www.ibm.com/support/knowledgecenter/SSY8AC_2.0.3/com.ibm.spectrum.discover.v2r03.doc/ins_configuredatasourceconnections.html
- Accelerate with IBM Storage website:

  https://www.youtube.com/watch?v=_YNfFDdMEa4

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services

Redbooks

Making Data Smarter with IBM Spectrum Discover

(0.2”spine)
0.17”<->0.473”
90<->249 pages

SG24-8488-00

ISBN 0738459135

Printed in U.S.A.