ISILON ONEFS ENTERPRISE FEATURES FOR HADOOP

VERSION 1.00

Abstract

This white paper details the capabilities of the Isilon OneFS architecture and how its enterprise features are ideally suited to support the storage centric data manageability and operational requirements of today's enterprise HDFS storage and analytics workflows.

April 2017

EMC Corporation
Hopkinton, Massachusetts 01748-9103
1-508-435-1000 In North America 1-866-464-7381
www.EMC.com

Publication History

Version	Date	Description
1.00	31 May 2017	Initial version.

Contents

Introduction

This document provides a high level overview of how the enterprise features available in Isilon OneFS can be integrated and leveraged when the Dell EMC's Isilon storage platform is used as the underlying storage layer within a Hadoop cluster.

Audience

This guide is intended for Hadoop systems administrators, storage administrators, IT architects, and IT managers who will be running OneFS with Hadoop.

Overview

The Isilon OneFS scale-out network-attached storage (NAS) platform provides Hadoop clients with direct access to Big Data through a Hadoop Distributed File System (HDFS) protocol interface. An Isilon cluster powered by the OneFS operating system delivers a scalable pool of storage with a global namespace.

Hadoop compute clients can access the data that is stored on an Isilon cluster by connecting to any node over the HDFS protocol. All nodes configured for HDFS provide NameNode and DataNode functionality. Each node boosts performance and expands the cluster's capacity. For Hadoop analytics, the Isilon scale-out distributed architecture minimizes bottlenecks, rapidly serves Big Data, and optimizes performance for MapReduce jobs.

In a traditional Hadoop deployment, the Hadoop compute nodes run analytics jobs against large sets of data. A NameNode directs the compute nodes to the data stored on a series of DataNodes. The NameNode is a separate server that holds metadata for every file that is stored on the DataNodes. Often data is stored in production environments and then copied to a landing zone server to be loaded on to HDFS. This process is network intensive and exposes the NameNode as a potential single point of failure.

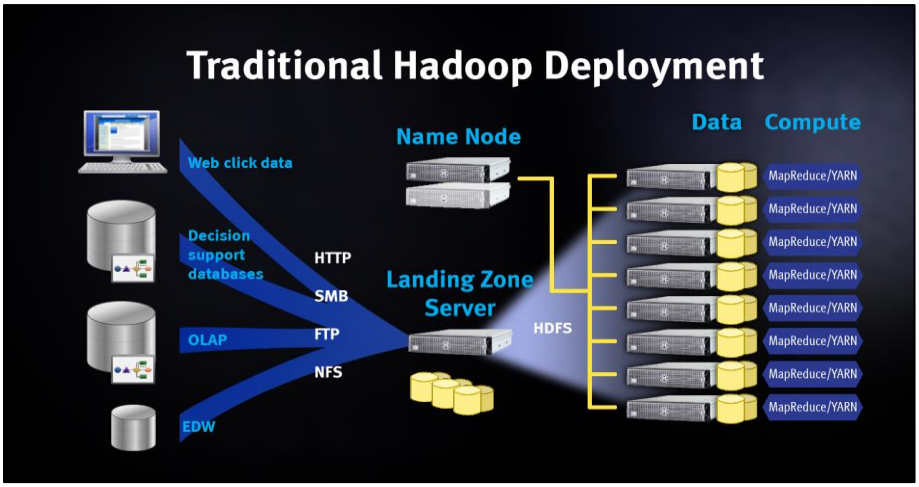


Figure 1: Traditional Hadoop Deployment

In an Isilon OneFS cluster with Hadoop deployment, OneFS serves as the file system for Hadoop compute clients. On an Isilon OneFS cluster, every node in the cluster acts as a NameNode and DataNode, providing automated failover protection.

When a Hadoop client runs a job, the clients access the data that is stored on an Isilon OneFS cluster by connecting over HDFS. The HDFS protocol is native to the OneFS operating system, and no data migration is required.

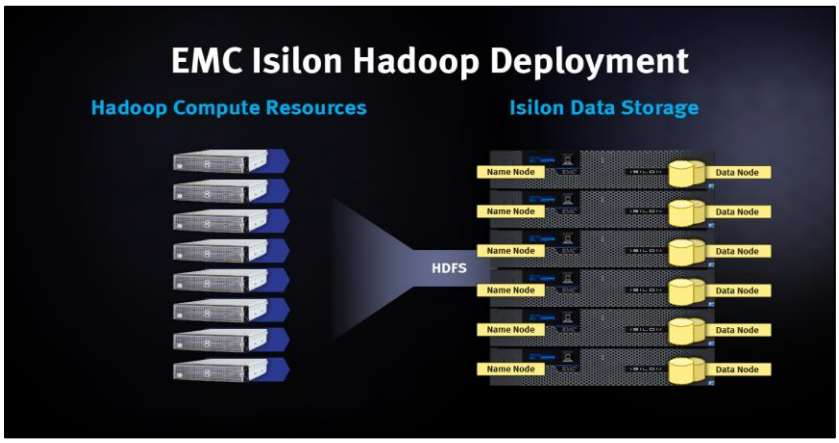


Figure 2: Isilon Hadoop Deployment

Compatibility

For supported versions, see the Hadoop Distributions and Products Supported by OneFS compatibility matrix.

Why Isilon

Isilon is implemented as a node-based clustered storage appliance running a combined operating and file system known as OneFS. It was originally developed to solve storage challenges at scale, and customers have relied on its enterprise features for over a decade. The unique capabilities and integration of support for the HDFS protocol have made it an ideal platform for scale-out Hadoop data storage.

As Isilon OneFS assumes the role of both the NameNode and all DataNodes in the Hadoop HDFS cluster, this provides a number of benefits.

- Since all nodes in an Isilon OneFS cluster function as a NameNode, no secondary NameNode or NameNode High Availability (HA) is required; any OneFS node will function as a NameNode and will provide continued data access without complex HA configuration. This substantially increases operational availability and decreases administrative overhead.
- Worker nodes in the cluster no longer function as DataNodes, The roles that remain on these nodes are now NodeManager roles and other dedicated roles for services, for example, Hive, HBase, Spark, and Impala, as quantity and distribution is dictated. This removal of the data and data management allows significantly more freedom around compute server selection and the required quantity. This alone can provide substantial consolidation of the physical space and power draw requirements of a given configuration.

Namespace Access

Since Isilon implements access to OneFS resident data through native HDFS protocol access via the HDFS namespace—with all Isilon nodes acting as a NameNode and a DataNode—there is no requirement to integrate directly with a compute cluster. Once the HDFS service and Isilon OneFS are configured for HDFS access, any data can be accessed by the HDFS URI directly, assuming the compute resource has the required security access. We will see many uses for the capabilities and flexibility this brings to the Data Lake later in the paper.

With Isilon OneFS providing HDFS access to any data on the cluster through its highly scalable clustered capabilities, the administrator is fully able to take full advantage of all of the native OneFS enterprise features.

OneFS Enterprise Capabilities

This whitepaper will discuss OneFS capabilities in depth; they are summarized in the following table:

Feature	Summary
Access Zones Multitenancy	Segregation and isolation of data with secure separation and identity management in each separate zones
On-Disk Data Protection	Optimized Erasure coding based protection of data, with granularity and customization down to the file level
Multiprotocol	Data access and unified authentication of the same data through multiple protocols—SMB, NFS, HDFS, and others
SmartConnect	Connection management and load balancing of client connections
SmartPools	Policy-driven tiering engine to control and manage data layout and location within OneFS and Isilon nodes
SnapshotIQ	Snapshots, fast, efficient data backups and recovery
SmartDedupe	Space savings through deduplication of data
SmartQuotas	Quota management with enforcement and thin provisioning
SyncIQ	Fast and flexible asynchronous replication for disaster recovery protection
CloudPools	Off-cluster tiering of infrequently accessed data to Cloud Platforms; Dell-EMC ECS, Amazon S3, Microsoft Azure, Google Cloud and Virtustream
Backup	Integration and support for standard backup technologies and methodologies
SmartLock	Policy-based WORM data protection
Data Security	In-flight data encryption and Data at Rest Encryption (DARE)
InsightIQ	Performance monitoring and reporting to manage storage resources

Table 1 : Enterprise capabilities

These features allow us the capability to separate access, backup and replicate data, restrict data growth, tier data, and many more. The ability to leverage these native data features outside of a HDFS data store is a primary example of how Isilon OneFS is the platform of choice for the scale out Data Lake.

The goal of this paper is to illustrate how these features can be leveraged specifically by Hadoop workflows achieving many enhanced capabilities for HDFS data management.

Workflow Segregation

Unlike a regular NameNode managed HDFS datastore an Isilon OneFS cluster allows the creation of segregated zones to effectively partition the cluster into multiple data zones. These zones can provide separation of data, authentication, and access. This gives you the ability to create multiple Hadoop workspaces on the same OneFS cluster, which can be all managed and administered from a single Isilon cluster. Throughout this whitepaper, we will configure and implement multitenant Hadoop and multiprotocol access to illustrate the enterprise features available, and describe how many OneFS features can extend the capabilities of large data management in the Hadoop space.

An ideal use case to showcase many enterprise features is the implementation of three Hadoop environments against a single Isilon cluster—production, QA, and development. In traditional Hadoop deployments this would require the deployment and management of three separate NameNode and DataNode environments across many servers.

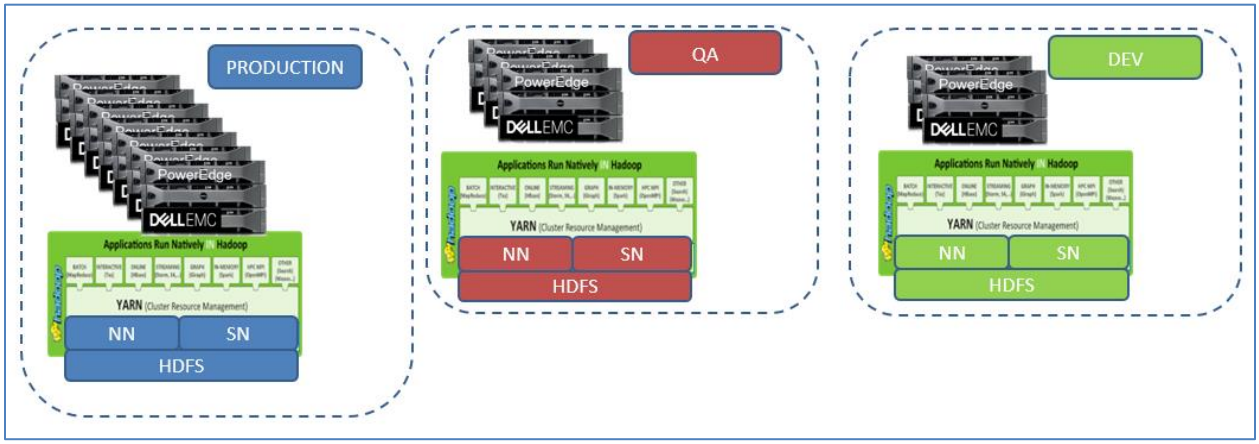


Figure 3: Traditional hadoop deployment; multiple NameNode DAS clusters to manage

The use case presented throughout this paper is a representative architecture used to highlight the capabilities and operational flexibility of a single Isilon cluster. Many of the implemented features can be used in a single use case or scale up to across many HDFS workflows.

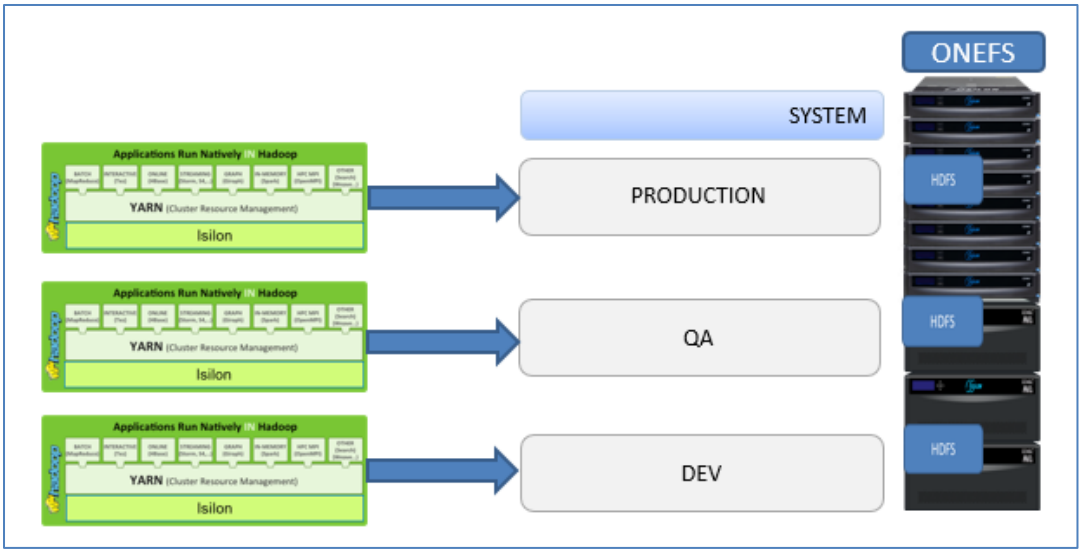


Figure 4: The multitenant cluster Isilon cluster; single cluster supports multitenant HDFS data

In Figure 4, the Isilon OneFS cluster is configured to support three separate Hadoop workflows. The data for all three Hadoop clusters resides on a single OneFS cluster, massively reducing the number of servers required as no DataNodes are required to provide HDFS data storage.

The ability to isolate data sets and user access to an OneFS Access Zone allows you to, in effect; create three separated dedicated environments within a single OneFS cluster. In this scenario, each zone is connected to a different Hadoop installation. OneFS gives us the ability to leverage different capabilities and features in each zone, allowing flexibility but single-point administration of the three HDFS datasets without needing to manage three separate HDFS data stores.

Production

The production environment includes the primary production data set and compute environment. This zone is secure and leverages many features to protect and manage the data. Data is backed up and protected in many ways—for example, replication, snapshots, and offsite—and is located on high performance nodes, providing maximum throughput and availability.

QA

The QA environment provides us the ability to qualify and validate features, while reviewing and testing Hadoop capabilities on representative data without impacting our production workflows. Data in this zone is periodically loaded from production to provide representative data sets and workflow.

Development

The development environment is used to test and validate new code, new workflows or tools without requiring any dependencies on production requirements or controls. Data in this zone is not backed up or regulated, and is freely available to provide easy access and flexibility to the Hadoop operational administrators and developers. The compute services could even be virtualized or run from a developer's machine as we have removed the requirement for large amount of local storage.

Delivering these different environments with traditional Direct Attached Storage (DAS) Hadoop clusters would require the deployment and management of multiple different servers with the disk space to support them. While, OneFS based deployment massively simplify the deployment and management of multiple Hadoop workspaces.

Hadoop Cluster Service Separation and Distribution Upgrades

Since Isilon OneFS is, in effect, a separation of the NameNode and DataNodes from the underlying core Hadoop services, we have decoupled the HDFS data layer from any dependencies on your Hadoop compute cluster. Cloudera Manager and Hortonworks Ambari both continue to manage and administer the core Hadoop cluster services—YARN, Mapred, Hive, Hbase, Impala, Oozie, and all the other Hadoop ecosystem services, except the underlying HDFS data, is now separated and can be managed independently. This decoupling provides many benefits to data management but also to Hadoop cluster management including:

- Data can grow independent of the compute environment; additional storage capacity can be added independent of compute resources; cold archive data can be tiered off automatically.
- Additional compute resources can be added independently of storage, diskless servers can be added into the compute cluster as data storage is separated from local disk.
- Data management is not dependent on Hadoop services—backups, snapshots, replication, and extract, transform, and load (ETL) can all be achieved outside of the core Hadoop services.
- Hadoop cluster upgrades are now decoupled from the HDFS data minimizing risk of NameNode issues and data loss.
- Different versions of Hadoop compute clusters can be running against a single Isilon OneFS cluster with separation of namespaces.
- Storage is now consolidated within the Isilon OneFS cluster and does not take up rack space or require additional administration.
- The Hadoop administrator is removed from managing and maintaining many aspects of the Hadoop cluster; the need for NameNode High Availability (HA) or backup is removed; DataNode management is simplified and removed from compute clients and centralized with OneFS.

On Disk Data Protection

OneFS is the underlying file system within an Isilon cluster. It's important to recognize that OneFS still protects data but in a manner different from traditional HDFS file system. OneFS employs the popular Reed-Solomon erasure coding algorithm for its protection calculations (OneFS does have the capability to mirror data similar to traditional HDFS, but erasure coding based protection provides a number of benefits). Protection is applied at the file-level, enabling the cluster to recover data quickly and efficiently. Inodes, directories, and other metadata are protected at the same or higher level as the data blocks they reference. Since all data, metadata, and FEC blocks are striped across multiple nodes, dedicated parity drives are not required. This both guards against single points of failure and bottlenecks, and allows file reconstruction to be a highly parallelized process. Today, OneFS provides +1n through +4n protection levels, providing protection against up to four simultaneous component failures respectively. A single failure can be as little as an individual disk or, at the other end of the spectrum, an entire node.

OneFS also supports several hybrid protection schemes. These include +2d:1n and +3d:1n, which protect againsttwo drive failures or one node failure, and three drive failures or one node failure, respectively. With larger drive density nodes, additional classes of hybrid protection schema exist, +3d:1n1d, which is designed to tolerate the failure of three drives or 1 node and 1 drive. Many other protection schemes exist also.

Protection Level	Description
+1n	Tolerate failure of 1 drive OR 1 node
+2d:1n	Tolerate failure of 2 drives OR 1 node
+2n	Tolerate failure of 2 drives OR 2 nodes
+3d:1n	Tolerate failure of 3 drives OR 1 node
+3d:1n1d	Tolerate failure of 3 drives OR 1 node AND 1 drive
+3n	Tolerate failure of 3 drives or 3 nodes
+4d:1n	Tolerate failure of 4 drives or 1 node
+4d:2n	Tolerate failure of 4 drives or 2 nodes
+4n	Tolerate failure of 4 nodes
2x to 8x	Mirrored over 2 to 8 nodes, depending on configuration

Figure 5: OneFS protection schemes

Since OneFS does not by default implement traditional 3x mirroring of HDFS data blocks, you see considerable space saving against HDFS Direct Attached Storage (DAS) storage, leading to a smaller footprint in the data center and considerably less wasted space when providing the underlying data protection.

Let's consider the difference in storage efficiency between mirroring and OneFS based protection schemes.

The parity overhead for mirrored data protection is not affected by the number of nodes in the cluster. The following table describes the parity overhead for requested mirrored protection.

2x	3x	4x	5x	6x	7x	8x
50%	67%	75%	80%	83%	86%	88%

Figure 6: Storage overhead with mirrored protection

Traditional HDFS 3X mirroring has a high parity overhead leaving only 33% of available space usable, as the remaining 67% is required to store the 2 additional mirrors.

The OneFS erasure coding based protection schemes have very different overhead requirements and are also customizable from the entire cluster down to the file level.

The following table describes the estimated percentage of overhead depending on the requested protection and the size of the cluster or node pool. The table does not show recommended protection levels based on

cluster size. It can be seen the used space efficiency with erasure coding based protection are substantial when compared to mirroring without loss of data protection.

Number of nodes	[+1n]	[+2d:1n]	[+2n]	[+3d:1n]	[+3d:1n1d]	[+3n]	[+4d:1n]	[+4d:2n]	[+4n]
3	2 +1 (33%)	4 + 2 (33%)	—	6 + 3 (33%)	3 + 3 (50%)	—	8 + 4 (33%)	—	—
4	3 +1 (25%)	6 + 2 (25%)	2 + 2 (50%)	9 + 3 (25%)	5 + 3 (38%)	—	12 + 4 (25%)	4 + 4 (50%)	—
5	4 +1 (20%)	8 + 2 (20%)	3 + 2 (40%)	12 + 3 (20%)	7 + 3 (30%)	—	16 + 4 (20%)	6 + 4 (40%)	—
6	5 +1 (17%)	10 + 2 (17%)	4 + 2 (33%)	15 + 3 (17%)	9 + 3 (25%)	3 + 3 (50%)	16 + 4 (20%)	8 + 4 (33%)	—
7	6 +1 (14%)	12 + 2 (14%)	5 + 2 (29%)	15 + 3 (17%)	11 + 3 (21%)	4 + 3 (43%)	16 + 4 (20%)	10 + 4 (29%)	—
8	7 +1 (13%)	14 + 2 (12.5%)	6 + 2 (25%)	15 + 3 (17%)	13 + 3 (19%)	5 + 3 (38%)	16 + 4 (20%)	12 + 4 (25%)	4 + 4 (50%)
9	8 +1 (11%)	16 + 2 (11%)	7 + 2 (22%)	15 + 3 (17%)	15+3 (17%)	6 + 3 (33%)	16 + 4 (20%)	14 + 4 (22%)	5 + 4 (44%)
10	9 +1 (10%)	16 + 2 (11%)	8 + 2 (20%)	15 + 3 (17%)	15+3 (17%)	7 + 3 (30%)	16 + 4 (20%)	16 + 4 (20%)	6 + 4 (40%)
12	11 +1 (8%)	16 + 2 (11%)	10 + 2 (17%)	15 + 3 (17%)	15+3 (17%)	9 + 3 (25%)	16 + 4 (20%)	16 + 4 (20%)	8 + 4 (33%)
14	13 + 1 (7%)	16 + 2 (11%)	12 + 2 (14%)	15 + 3 (17%)	15+3 (17%)	11 + 3 (21%)	16 + 4 (20%)	16 + 4 (20%)	10 + 4 (29%)
16	15 + 1 (6%)	16 + 2 (11%)	14 + 2 (13%)	15 + 3 (17%)	15+3 (17%)	13 + 3 (19%)	16 + 4 (20%)	16 + 4 (20%)	12 + 4 (25%)
18	16 + 1 (6%)	16 + 2 (11%)	16 + 2 (11%)	15 + 3 (17%)	15+3 (17%)	15 + 3 (17%)	16 + 4 (20%)	16 + 4 (20%)	14 + 4 (22%)
20	16 + 1 (6%)	16 + 2 (11%)	16 + 2 (11%)	16 + 3 (16%)	16 + 3 (16%)	16 + 3 (16%)	16 + 4 (20%)	16 + 4 (20%)	16 + 4 (20%)
30	16 + 1 (6%)	16 + 2 (11%)	16 + 2 (11%)	16 + 3 (16%)	16 + 3 (16%)	16 + 3 (16%)	16 + 4 (20%)	16 + 4 (20%)	16 + 4 (20%)

Figure 7: OneFS estimated protection overhead depending on cluster size and requested protection

Additional information can be found at: https://www.emc.com/collateral/hardware/white-papers/h10588-isilon-data-availability-protection-wp.pdf

Data on OneFS can also be natively protected in many other ways, for example, snapshots, replication to another cluster, or backups. The ability to customize the protection down to file level provides increased flexibility and the ability to protect the data at a level appropriate for the amount of nodes and data you are using. OneFS will suggest the recommended protection level to maximize data availability based on the nodes implemented in the cluster. It is always recommended to protect at this level or high to ensure data integrity.

Multitenancy

The key to the implementation of multiple Hadoop clusters running on a single OneFS cluster is the ability to create segregated namespace within the single file filesystem.

Access Zones

OneFS Access Zones represent a boundary to define access and security; each zone is tied to a specific data path within the cluster. A client reaches the data in an Access Zone through a dedicated DNS namespace (the SmartConnect Zone Name) and it is tied to a specific set of authentication providers to control access into the zone. An Access Zone is the fundamental unit of multitenant separation within OneFS, and many features of OneFS can be leveraged at a zone level.

The built in System Access Zone should only be used for administrative access or for protocols that are not zone aware. For additional information on the System Access Zone, see the OneFS Web Administration Guides.

Zone-based Access

Connecting to an OneFS Access Zone is administered and managed by OneFS SmartConnect, a connection management and load balancing module. It provides the ability to control which nodes and interfaces are available for client connections, how connections are load balanced, and how network resiliency is achieved. The flexibility to define connection policy on a zone level allows the administrator to create tiers of network connectivity to control which nodes are participating in client workflows. This can be very useful in mixed node clusters where different requirements need to be met for each tenant.

Zone-based Authentication

Since each OneFS Access Zone, in effect, represents a segregated environment within the cluster's namespace, each zone can be attached to different authentication providers to create secure dedicated access within that zone. A zone can be configured against multiple different providers—Active Directory, a Kerberos provider, NIS, and local providers. The ability to provide different access controls to different workflows gives the administrator the freedom to choose.

Zone-aware Multiprotocol

A core tenant of OneFS is the ability to provide true multiprotocol unified access to data residing within OneFS. This has huge benefits in the management and lifecycle of data. Data can now be natively written and read from multiple protocols—HDFS, NFS, SMB, and others—without the need to move or load data from disparate siloes of storage. Data can be natively written to and then accessed by other protocols, removing the requirement for extensive ETL operations between systems. Like all capabilities within OneFS, multiprotocol data access is managed at the Access Zone level.

Core Architectural Features of OneFS Clusters

The building blocks of an OneFS cluster are Isilon node pools; the minimum number of like nodes in a cluster node pool is 3. Having built our cluster we can define our implemented OneFS configuration on these pools of nodes or across the whole cluster consisting of different node pools.

Often an OneFS feature is layered upon a foundational feature or defined within the scope of another feature. The following outlines an example of how OneFS cluster architecture is defined from a node pool through protocol enablement and how the features interact:

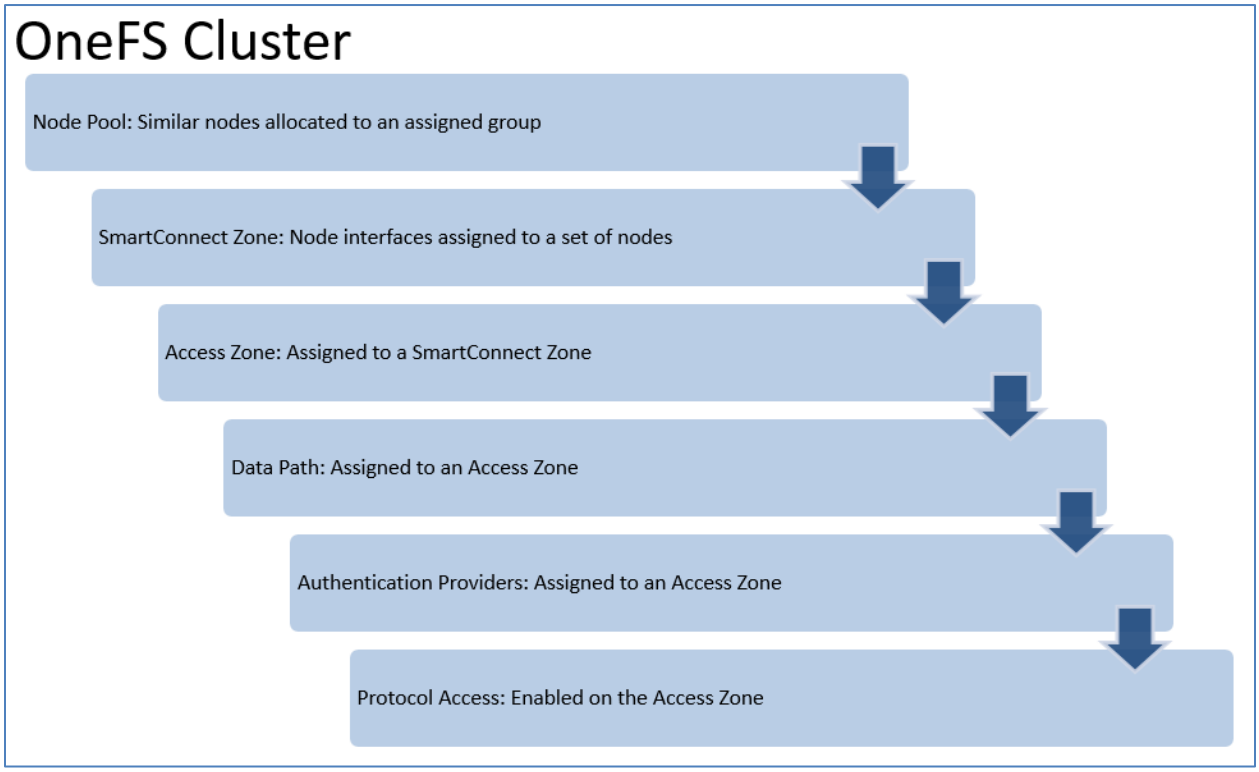


Figure 8: OneFS architecture

Understanding these dependencies is critical when defining the cluster architecture to support our requirements. As we build out our cluster, we will see how to configure different features to support our workflow.

Cluster Architecture

In the OneFS cluster utilized in this whitepaper, we have a mix of nodes types that are designed to support different workflows. The cluster compromises a mix of X node and NL node types. Clearly a cluster could be constructed out of many different nodes types of a similar class and configuration to meet any requirement. In this example, we have two pools of nodes that can provide different performance characteristics and will be used to support our workflows.

Node Type	Primary Role in Cluster
X410	High throughput, active production; Hot Data
NL410	Data Protection, Recovery, and Archiving; Cold Data and Development Data

Table 2: Node types in the cluster

In effect, we have created two distinct tiers of storage within the single OneFS cluster, which we will define and manage the data across. The X410 nodes will support our production and QA data workflows while hosting our hot active data. The NL nodes will be used by our development workflow and host our cold archive data. OneFS can automatically manage data movement between our tiers through SmartPools and segregate access with SmartConnect as we shall see.

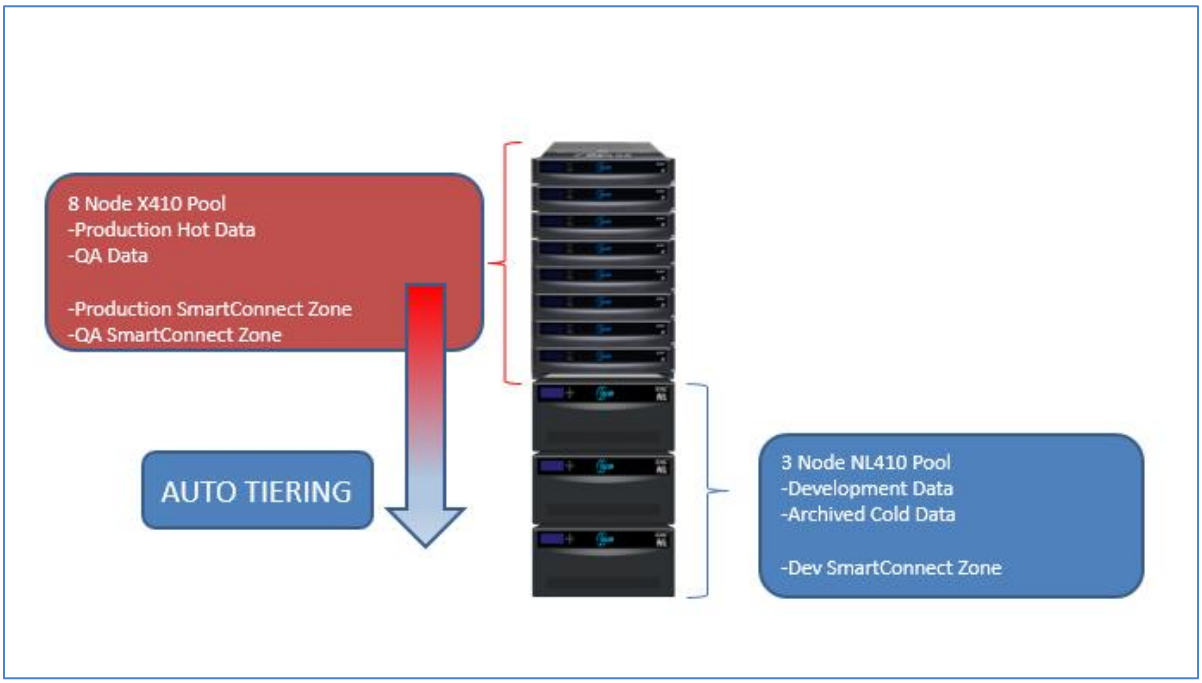


Figure 9: Cluster node pools

Having defined our two tiers of nodes within OneFS, we can continue to define how a multi workflow configuration is defined.

SmartConnect Zones and Pools

To control access and define connectivity to our cluster, we define network pools with an associated SmartConnect Zone name. This is a defined DNS namespace and is how clients connect to each network pool. Each SmartConnect name will be the ingress point into our Access Zone; it will control authentication and which nodes are available to connect to as defined by the pool membership.

Since any network card interface on any node within the cluster can be associated with a network pool, in order to meet our throughput requirements and to segregate our connections to the appropriate nodes we can assign specific interface to the pool to manage connectivity with SmartConnect.

The primary design consideration when developing a connectivity strategy is to provide the appropriate connectivity to our Hadoop workflows to ensure appropriate throughput and availability.

Not only does SmartConnect manage the load balancing of client connections to nodes and interfaces, SmartConnect can also manage the failover of IP addresses to maintain connectivity with stateless protocols. In a single rack or simple split rack deployment, OneFS can provide NameNode and DataNode access with a

single Dynamic pool (Dynamic IP allocation dictates IP addresses failover to other interfaces if that interface is down), all the Isilon node interfaces are presented to all Hadoop clients with IP address failover to provide the maximum number of interfaces with the highest level of resiliency.

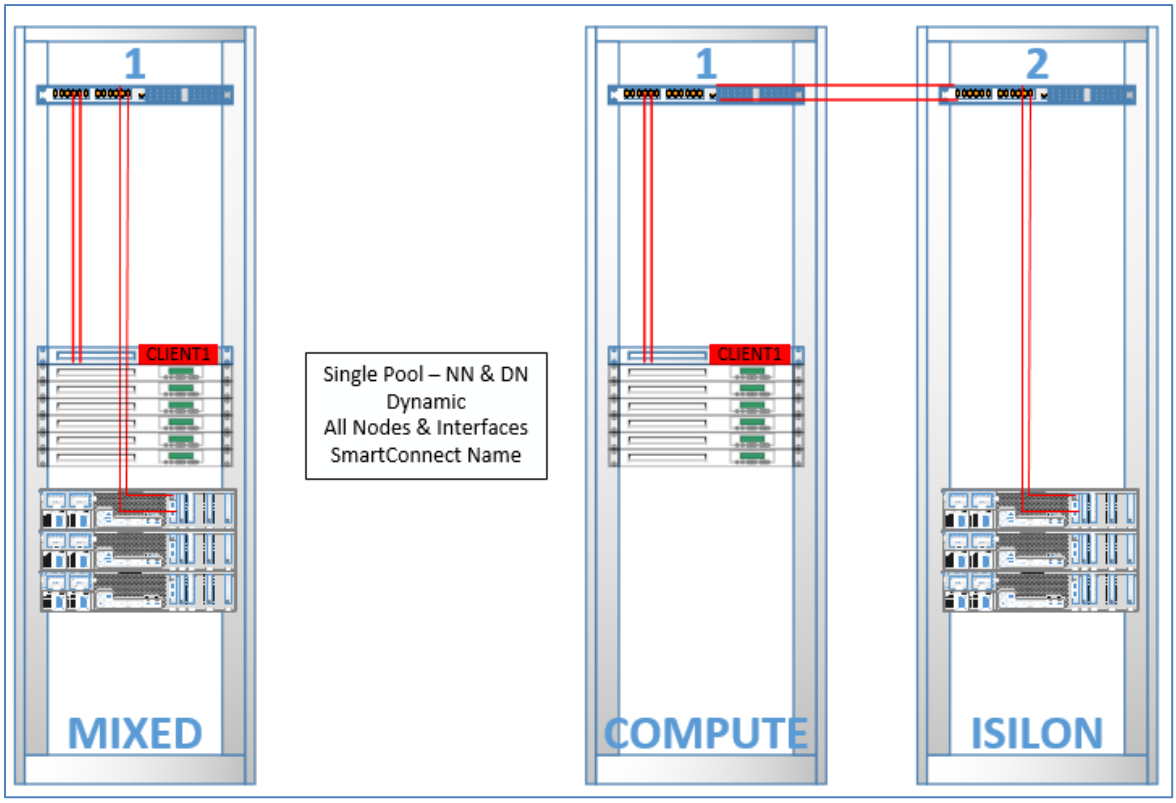


Figure 10: Single SmartConnect pool deployments

An important extension to using SmartConnect IP pools is the ability to manage inter-rack data traffic and therefore limit inter switch network traffic. Traditional hadoop distributions are built to be data location aware from the NameNode and DataNode, Hadoop clients will attempt to locate data blocks located as close to them as possible to limit network traffic. Since every node in an Isilon can be a NameNode and a DataNode we can define pools of interfaces to act as NameNode or DataNode pools. The ability to create groups of nodes to act as DataNodes can be implemented with Isilon Hadoop virtual racks to create location aware HDFS data storage. The goal with a rack design is to limit client connections to Isilon interfaces collocated within the same rack creating location awareness with OneFS. Since all Isilon nodes are connected by a high speed internal network all nodes continue to have access to data residing anywhere within the OneFS cluster.

Isilon racks and SmartConnect provide rack awareness by defining groups of Isilon nodes that are preferred by groups of client IP's. In the example below, All Isilon nodes are defined as NameNodes but all the source client IP's in Rack1 will prefer to get data from the interfaces on Isilon nodes in Rack1. While clients IP's located in rack2 will prefer to make connections and get data from Isilon nodes in Rack2.

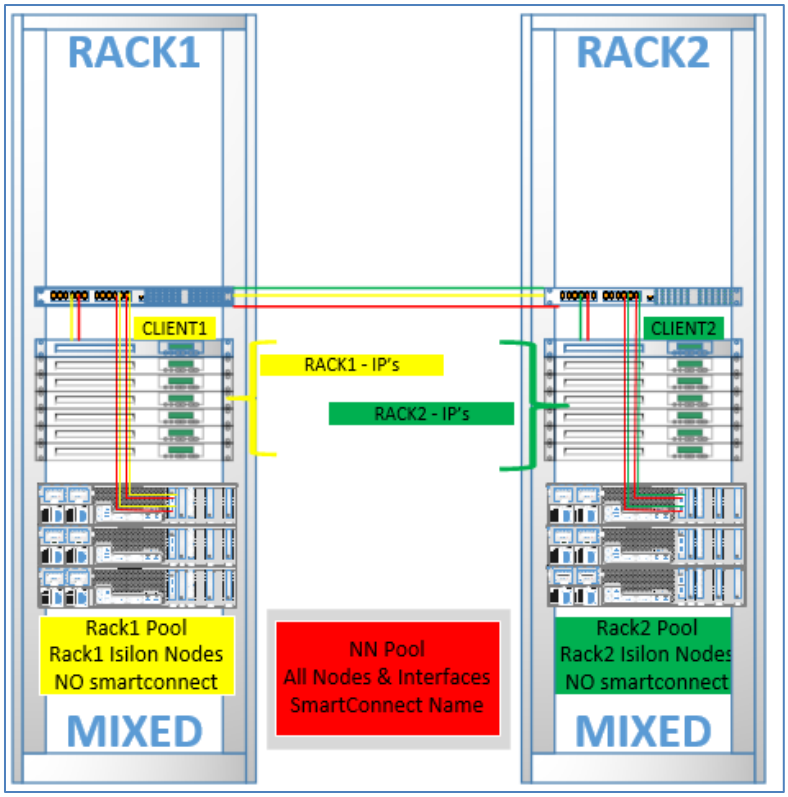


Figure 11: OneFS HDFS rack implementations

By defining multiple SmartConnect pools and implementing HDFS racks we can create location aware Isilon nodes, when the Isilon responds to a NameNode request it will attempt to send the client to the defined rack. When collocating clients and nodes in the same rack we can limit the network traffic leaving top of rack switches to provide location aware responses to data blocks from Isilon.

If no client and Isilon node rack colocation is implemented, then virtual rack are not required. OneFS 8.0.1.x introduced two new HDFS features which changes the way OneFS handles HDFS connection and these changes impact the recommended approaches to IP pool configurations for Hadoop Isilon integrated clusters.

DataNode Load Balancing

Previously the HDFS service on OneFS used a round robin scheme to provide IP address within a SmartConnect zone for new HDFS client to connect to or it leveraged racks. Typically, this works reasonably well but in highly loaded Hadoop cluster. There was a chance that new HDFS clients will be connected to a node that is already highly loaded with DataNode connections potential creating connectivity skew. In 8.0.1, OneFS is built with the intelligence to ensure that new HDFS client will always be given an IP address where the node's total number of TCP connection count is the lowest, allowing the client to have the best chance to leverage the least loaded node to get the best performance and throughput. This in effect removes the requirement for racks to achieve better load balancing when location awareness is not required.

DataNode Write Recovery

Now with HDFS DataNodes the client has the ability to recovery from node failure or network failures by using a feature called Pipeline Recovery. This feature allows the Hadoop clients to continue writing their data to a different node in the event that the current node is unreachable or returns an error for some reason. This feature creates greater resiliency with regard to Hadoop clients and Isilon DataNodes, as in effect we respond with three potential DataNodes for the client to connect to as opposed to one in previous versions of OneFS.

The implementation of these features provides better load balancing and resiliency within OneFS and allows us to change our recommended approaches to IP pools and the use of racks. We are no longer dependent on using racks to marshal separate NameNode and DataNode connections to optimize load balancing. Racks are still a completely valid configurations and should still be used to create location aware client DataNode connections if the cluster architecture dictates optimization of rack awareness.

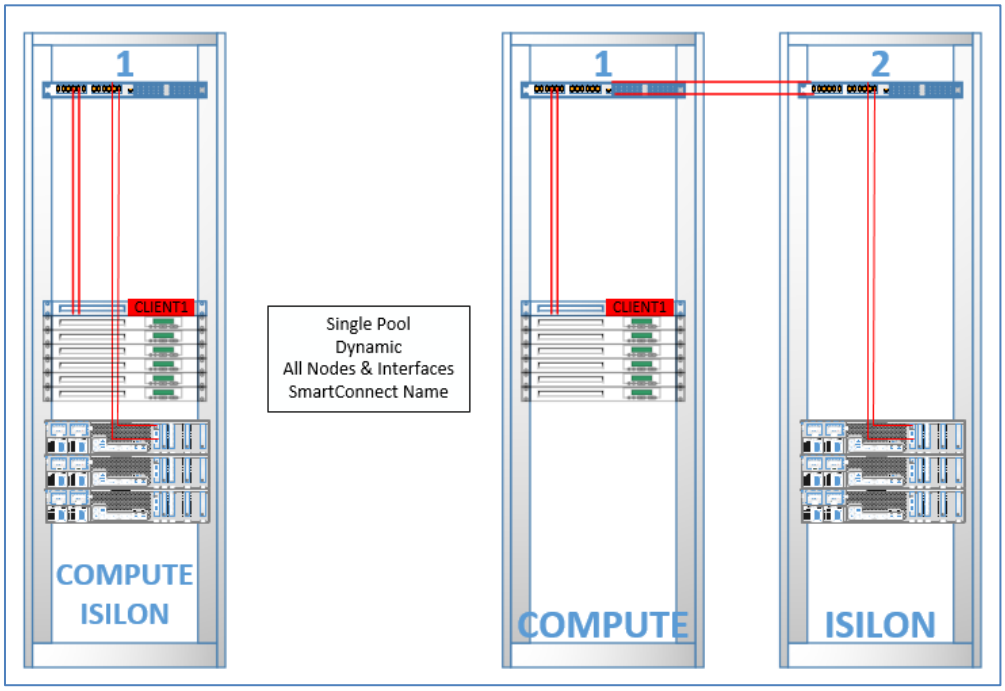


Figure 12: Pool strategies for OneFS that leverage DataNode Load Balancing

The primary considerations when defining and IP strategy are:

- Availability of IP addresses*
- OneFS node and interface assignment and use
- Rack and location of compute clients and OneFS nodes with respect to each other

* Any deployed pool strategy is dependent on the appropriate number of IP's being available for assignment within the pool to meet the requirement of allocation or failover.

Advanced configuration of racks, network pools and SmartConnect is beyond the scope of this paper. Consult the Isilon OneFS Networking guide for additional details.

Authentication Providers

Not only do Access Zones define data and connectivity boundaries, they create a security domains in which different types of access can be enforced. OneFS allows the configuration of multiple authentications sources from the cluster. Once defined, each authentication provider is attached to an Access Zone to determine what security is evaluated when a user attempts to access data in that zone.

Authentication and Identity Sources:

- Active Directory – Microsoft's implementation of Kerberos and LDAP
- MIT Kerberos - encrypted negotiated authentication based on tickets
- LDAP – Directory Service providing identity management, used in conjunction with Kerberos
- NIS – Network Information Service provides authentication and identity management
- File Provider – an authoritative-supplied file providing user and group information
- Local – local users and groups added to the cluster by an administrator

AIMA – Authorization, Identity Management and Authorization

Having defined which authentication providers will be used within an Access Zone to manage access, we can discuss briefly the process with which OneFS ultimately manages file access. Isilon OneFS presents a true unified model with access to a file or directory being consistent for all protocols. It achieves this through two primary mechanisms:

1. A unified access token, representing the user's persona to OneFS is created
2. A unified permission model, providing consistent permissions regardless of protocol access is implemented

This model can be summarized as follows:

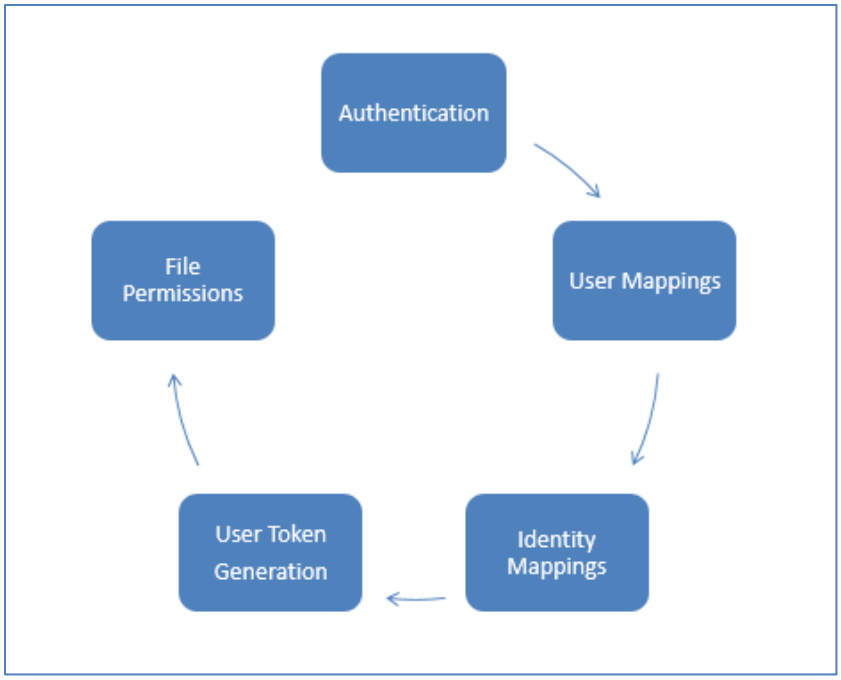


Figure 13: OneFS AIMA

1. The user is authenticated against defined authentication providers
2. User Mapping maps users between defined identity providers
3. OneFS maps the appropriate identities to users
4. OneFS constructs an access token including all the user's identity and all group membership; the users persona to the OneFS cluster
5. The access token is evaluated against the defined permissions to evaluate access

This unified model allows OneFS to present a single access and authentication experience to any client attempting to access data. The ability to control access in the context of a segregated workflow is again critical to our multi-Hadoop cluster implementation.

The specific details of implementing identity management and authorization is beyond the scope of this paper. A number of excellent reference are included in the Appendix.

Reference Cluster Architecture

If we pull our base architecture together, we can understand how these core OneFS capabilities have created a scale-out but segregated platform that provides secure multi-Hadoop workflow capabilities with the flexibility to permit each Hadoop workflow to operate within its own business requirements. Having now defined this architecture and how it is implemented, we can look at how additional enterprise features can be used across the cluster, only within an Access Zone and to external Isilon clusters and cloud services.

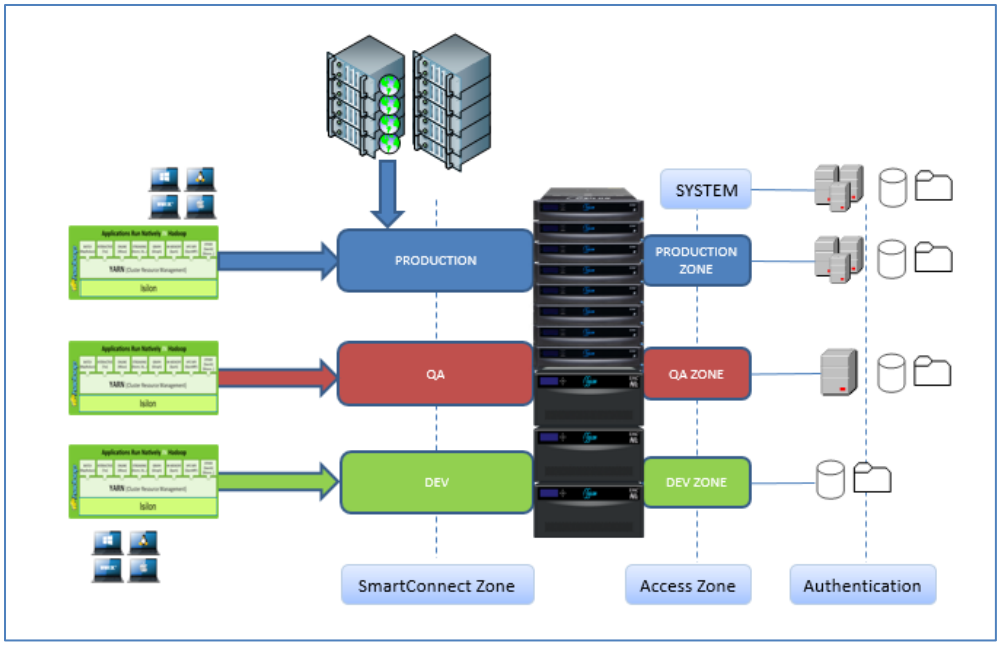


Figure 14: Multitenant OneFS Hadoop cluster

Figure 14 summarizes our base implemented architecture, which we will use to illustrate additional feature capabilities and operation of OneFS. In the upcoming section, we will look at each feature and how it integrates into our defined architecture, as well as the operational capabilities it brings to managing HDFS data on Isilon OneFS.

Node Pools, SmartPools and CloudPools – Data Tiering

As we saw earlier in this paper, Isilon considers nodes of a similar type to be equivalent and defines them as a node pool. If your cluster is made up of different node pools—these different node pools more than likely consist of different node types (memory & CPU) and/or disk types—they, in effect, create different performance characteristics that can be utilized as a workflow tier. This gives Hadoop administrators workflow isolation, higher granularity of data placement, and independent scalability within a single OneFS cluster.

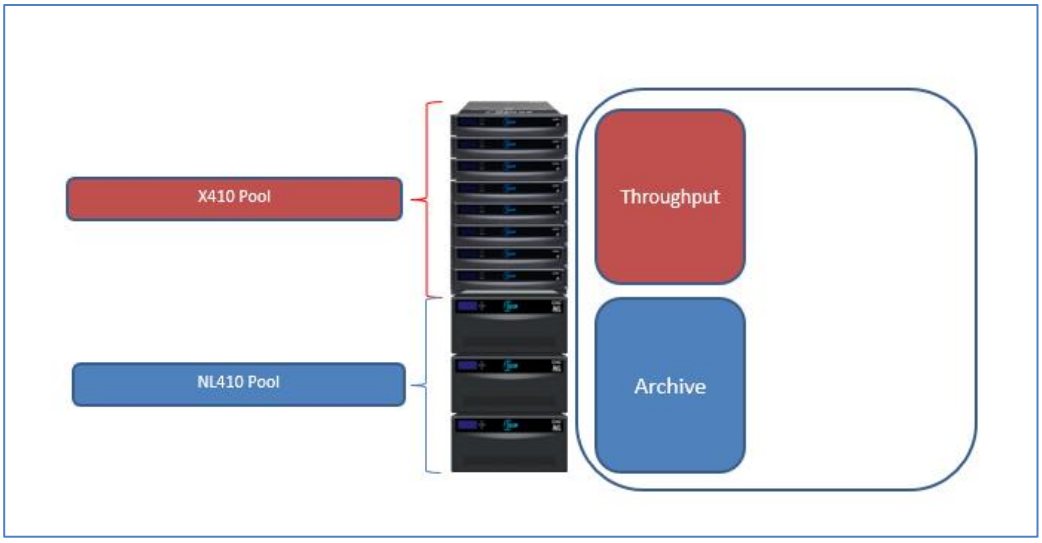


Figure 15: Node tiers within OneFS

OneFS continues to see the entire file system as a single scaled-out namespace. It is unaware of the different characteristic of the groups of similar nodes that exist within the cluster through node pools. At the highest level, we can categorize our storage containing two workflow tiers:

- Throughput – XNodes
- Archive – NLNodes

OneFS can either manage files by moving them between node pools (XNodes or NLNodes in this case), or we can define a tiers within OneFS, in effect grouping them as having similar characteristics. This is important, as multiple node pools can be grouped within a single tier if needed. By default OneFS will use all nodes in the cluster regardless of which nodes were accessed. If SmartPools and file pool policies are not used data can be written across different node type creating an imbalance in data access profiles. In order to maintain consistent throughput the location of data should always be managed in mixed node clusters.

File Pool Policies

Having defined tiers to differentiate our storage nodes, we can use SmartPools file pool policies to define data alignment and behavior. A file pool policy is used to locate or move data within OneFS within storage tiers. Data movement is seamless, including in-flight read/write activity, locking semantics, backup application interaction, and underlying file availability. With file-level granularity and control, automatic policies, manual overrides, or an Application Programming Interface (API), it is possible to tune performance and layout, storage tier alignment, and protection settings, with zero impact to end-users. The SmartPools capability is native to OneFS, which allows for unprecedented flexibility, granularity, and ease of management.

This capability far exceeds what is available natively from Hadoop in that data management is automated and policy-driven. File Pool policy criteria for defining the automation of file selection include:

- Filename
- Path

- File Type
- File Attribute
- Last Modified Date
- Creation Date
- Last Accessed Date
- Metadata Change
- File Size

Matching can be simple or consist of several permutations from the choices above. Once the filter is defined, then items such as Tiers and Protection Policy changes can be applied to the data as it proceeds through the data lifecycle.

Note: By default, OneFS will place files anywhere. This can have performance implications if the Isilon nodes in different node pools have different performance characteristics. It is a best practice to direct the default file policy towards a specific node pool or tier.

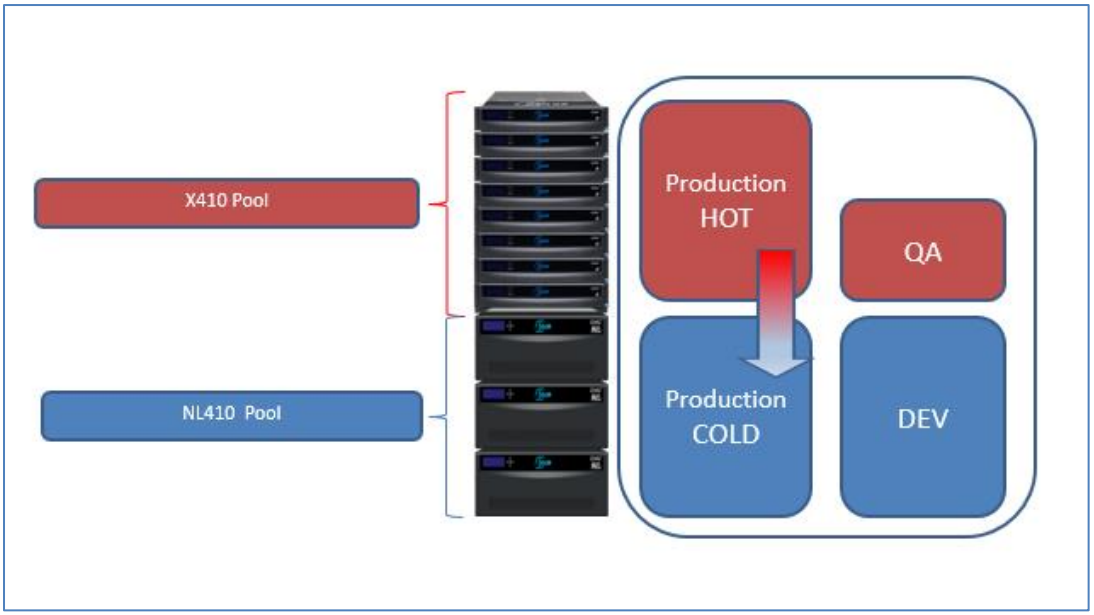


Figure 16: File Pool Policy define where data is located

File Pool policies can define where data is written to initially, or based on a defined characteristic where data blocks are moved to by a SmartPools job, based on the rules of the file pool policy. A SmartPools job is a scheduled process that runs processing of file pool policies and the movement of data within OneFS between defined storage pools.

In our example cluster, we wish to define where the data for our three Hadoop workflows resides in our tiers of storage. By defining file pool policies, we instruct OneFS where to locate the initial data written by the three Hadoop workflows. With these file pool policy, we instruct OneFS to locate the data blocks on the defined tier to provide the best performance for that workflow. By locating all the data within a node pool, we get consistency of access and can make the best use of disk tiers economically.

Hadoop Workflow	Tier
Production	Production HOT – all new and active data
QA	Production HOT – all qa data
Development	Production COLD – all dev data

Table 3: File Pool policies defining the location of data for each workflow

It is common within large datasets to have new, active hot data that is frequently accessed but as time moves on, we often see large amount of this data become inactive, frequently less used or cold. This is where file pool policies and SmartPools can actively manage this data for us after it was initial written with automated tiering.

Hadoop Workflow	File Pool Policy	Tier
Production	Data not accessed in 90 days	Production COLD – data not active in 90 days

Table 4: A sample File Pool policy to tier down cold data to the archive nodes

Adding another policy to move any data that has not been actively accessed within the last 90 days is seamless, automatic, and transparent to the user. The data is still fully-accessible to the compute cluster and can be re-promoted back to the hot tier if needed. The advantage of moving this data off the hot tier is that it frees up disk space and uses the more cost effective larger but slower drives of our NL Nodes. This policy-driven behavior automatically manages our storage infrastructure for us while providing the best use of the available disk to maximize our workflows.

CloudPools

OneFS also has the capability of extending its tiering capabilities further by extending its data tier off the OneFS cluster and into a cloud service object store (Amazon S3, Microsoft Azure, Virtustream and Google Cloud), or as an on-premise object store (Dell EMC ECS) or even another Isilon cluster. Typically, an Isilon Data Lake can store and manage up to around 50PB of unstructured data. CloudPools allows this to be expanded to a virtually limitless scale. All file metadata for a CloudPool tiered file, including security and file attributes, is stored local to the Primary and DR Isilon (CloudPools integrates with SyncIQ). The stub is typically very small in size while the data resides in the cloud tier thereby freeing up actual local storage. If this file is accessed, the data is recalled from the off-cluster location and served to the client.

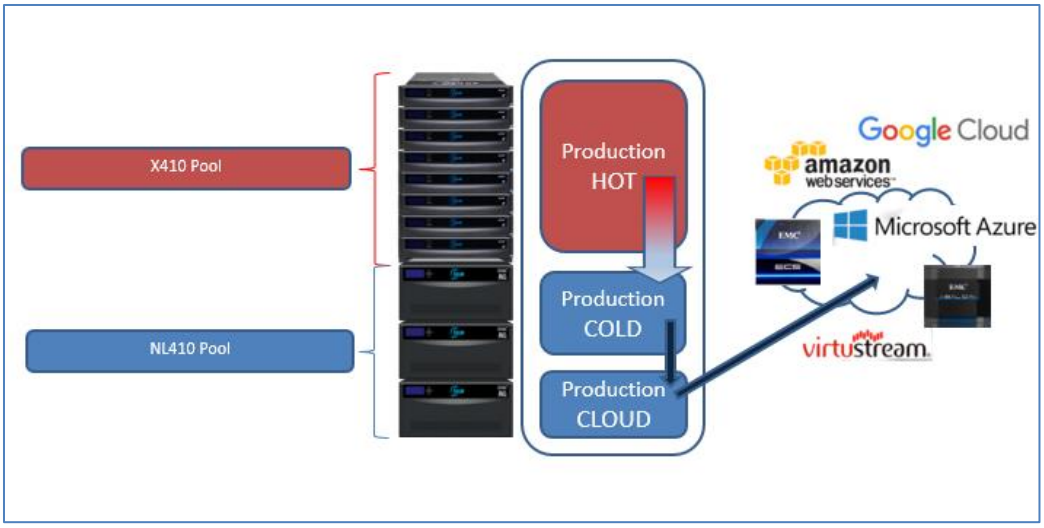


Figure 17: The additions of a CloudPools tier

A CloudPools tier is simple to deploy and manage because it uses the same flexible and powerful policy engine of File Pool policies and SmartPools. CloudPools can be deployed in minutes and is easy for administrators to deploy and manage while data movement through the tiering lifecycle is abstracted from end users. The data transmission leverages encryption in order to protect data while being moved to the offsite or onsite object storage. In addition, compression is an available option to optimize the network bandwidth for data transfers to and from the cloud while also reducing end-user latency. CloudPools is also integrated with the other OneFS storage management and protection services including SnapshotIQ, SyncIQ replication, SmartQuotas, and NDMP backup, making it an additional tool in data lifecycle management.

CloudPools with Hadoop

Cloudpools can easily be used to extend the HDFS namespace into the off cluster object storage. The data residing in the object store is presented to any protocol served by OneFS as native. So, for Hadoop, the CloudPools tiered data just appears as part of the local HDFS root being served by OneFS. It is critical when implementing CloudPools to understand that no SLA on data retrieval exists, the data in the cloud tier should be considered frozen and not used for active analytics. If you need to run analytics on cloud-tiered data, we recommend that you recall the data from the cloud tier to the local OneFS file system.

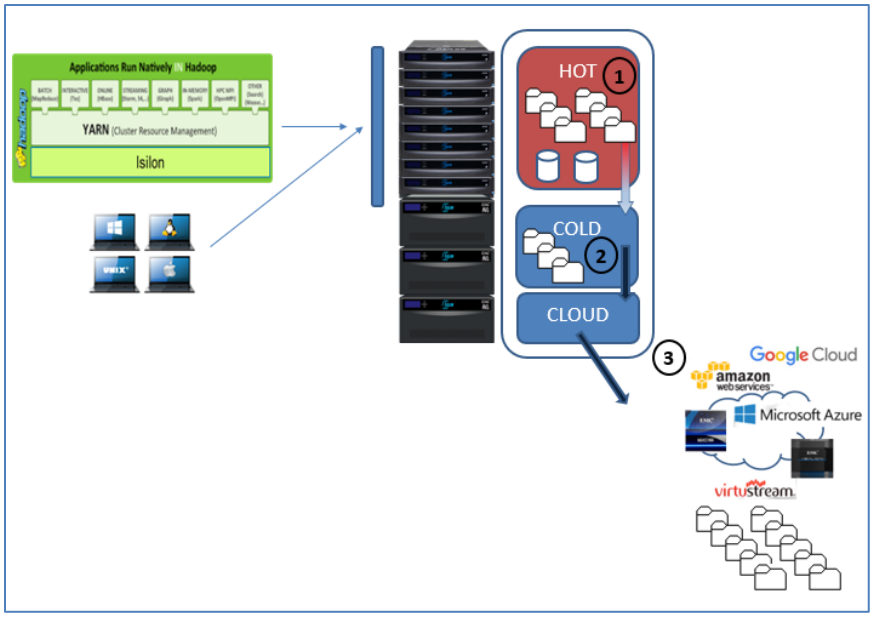


Figure 18: Data lifecycle of HDFS production data

Hadoop Workflow	File Pool Policy	Tier
Production	Data not accessed in 180 days	Cloud COLD – data not active in 180 days

Figure 18 illustrates:

- Hot Active Data – All active HDFS data, HDFS SQL data; Hive, Impala, HBase ①
- Cold Data – HDFS data that is no longer in active use but is occasionally accessed ②
- Frozen Data – HDFS data that is not actively used but must be retained ③

Through the use of node pools, file pool policies, CloudPools and SmartPools, we can develop an information lifecycle management strategy for our HDFS data to maximize the capabilities of our OneFS cluster.

The advantages of using these features with HDFS data is clear:

- Active data sits on our most-performant tier
- When data becomes cold it is automatically moved to our efficient archive tier
- Frozen data can be cost-effectively stored in the cloud freeing up OneFS disk space

The flexibility and native capabilities to manage and tier data with OneFS are much more comprehensive than with the traditional HDFS file system. OneFS provides both Hadoop and storage administrators many options in maximizing and optimizing both the compute and storage requirements of today's modern Data Lake architectures.

SmartQuotas – Quota Management

The enterprise Data Lake is allowing the unprecedented growth of data and with it comes the requirement for storage administrators to control how much space a user or group of users are permitted to use. The ability to monitor, report on, and ultimately deny the ability to consume space is a critical element in the information data lifecycle. The OneFS SmartQuotas feature provides the native capabilities to manage storage utilization of the HDFS data store of the attached compute platform.

OneFS quotas are defined by four criteria:

- The directory to monitor
- Whether snapshots are tracked as part of the quota
- Whether protection overhead is tracked as part of the quota
- The quota type:
    - **Directory Quota** – A directory and subdirectory only
    - **User Quota** – Specific user or default(every user)
    - **Group Quota** – Members of a specific group or default group (every group)

Multiple quotas can be created on the same directory, but they are required to be of different types, and any specific quota will take precedence over a default quota. Quotas can be defined on any directory and nested to create complex storage policies to manage your specific requirements.

OneFS supports tracking and storage limit quotas:

- **Tracking** – An accounting option to review, monitor, and report on storage usage
- **Limit** – Track storage usage and limit writes on being exceeded

Having determined the type of quota to be used, OneFS has the ability to enforce different threshold behaviors within the defined quota:

- **Advisory** – An informational limit that can be exceeded, logged and reported on
- **Soft** – A limit once exceeded provides a grace period until writes are denied
- **Hard** – Once exceeded writes fail

All thresholds provide logging and notification when exceeded, and write access will resume once usage fails below the threshold defined to re-allow access.

Let's review how the different quota options will be enforced in our multitenant Hadoop cluster to support the different requirements of the HDFS workflows. The advantage of using native OneFS quotas limits the administrative overhead required to manage storage utilization of the HDFS data store. SmartQuotas also presents a single point of administration to managing all HDFS data storage and utilization as opposed to managing separate silos of storage.

Workflow	Requirement
Production	-User home directory not to exceed 5TB, writes denied, ignore overhead, include snapshots -Monitor the HFDS root include snapshots, including data protection but no alerts

Table 5: Quota requirements

Looking at our storage management requirement for our production data, SmartQuotas can be configured to automatically provide reporting and enforcement of these different policies without having to manually run accounting and enforce limits. Let's review the specific quotas in detail and how OneFS can manage and enforce the storage utilization for us.

Production Quotas

In order to limit all HDFS users from creating more than 5TB of data in their home directory, we can use a user quota. Since each user has a dedicated secure home directory a default user policy on the Hadoop users root path '/users' directory will enforce this, we will limit that directory and each user to a 5TB limit as follows:

- Create a default user quota on the Hadoop root user directory
- Include snapshots
- Make it a hard quota limit – writes will be denied at 5TB
- Use the default system notification to warn users when the quota is exceeded

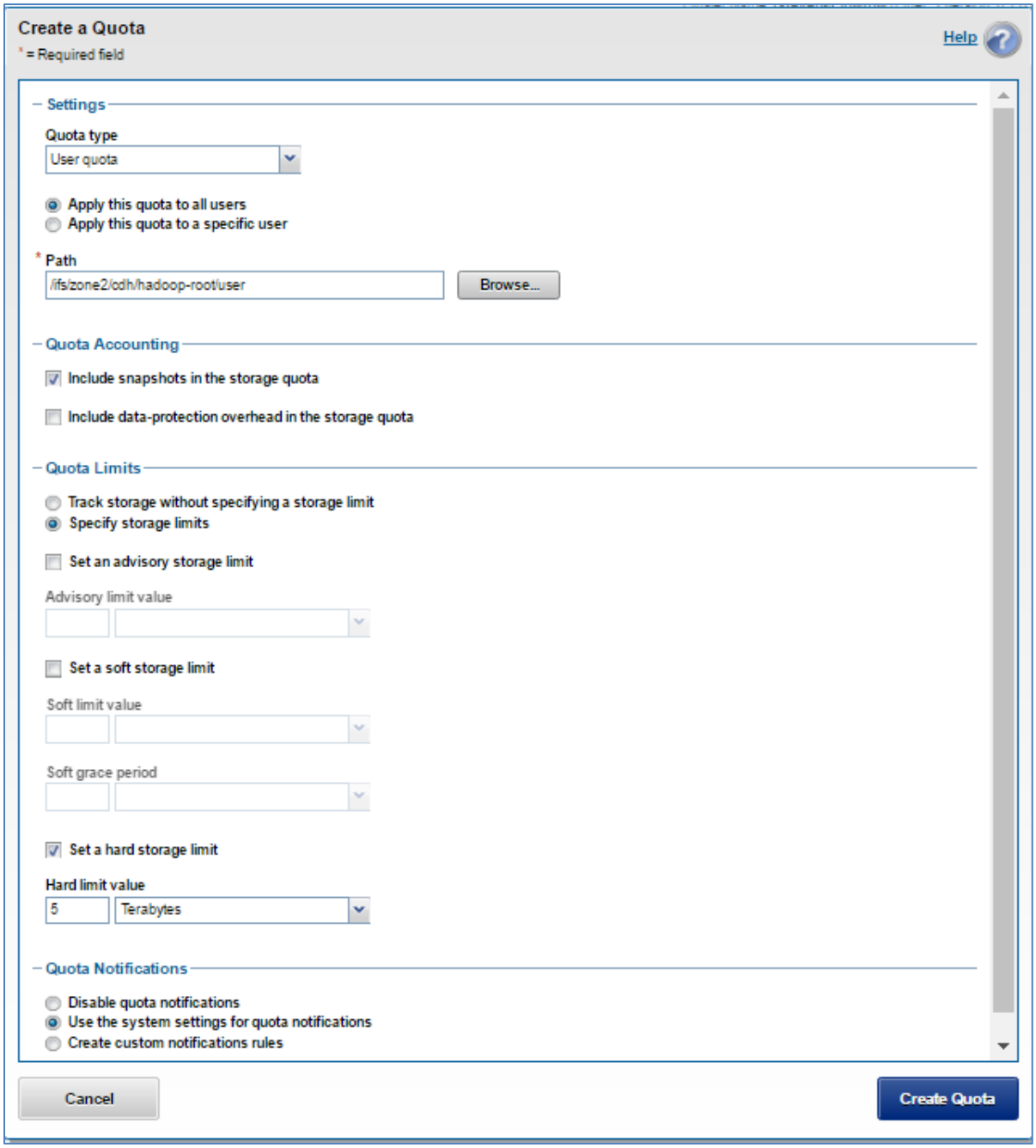


Figure 19: Default user home directory quota, denying writes at 5 TB

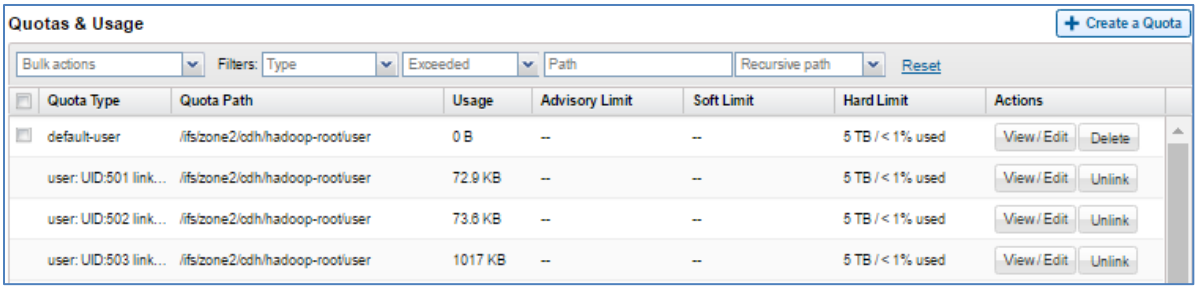


Figure 20: A default user quota

To track storage utilization of the entire HDFS root, we create a directory accounting quota. This policy does not enforce any limits; it is used for accounting purposes and to report on the overall usage of the HDFS root. Accounting quotas can be particular useful to determine the actual disk space used by a workflow, including snapshots and data protection overhead that are not accounted for in the data storage. An accounting quota of this type will give a precise accounting of the disk space used by HDFS data and not just the underlying data blocks.

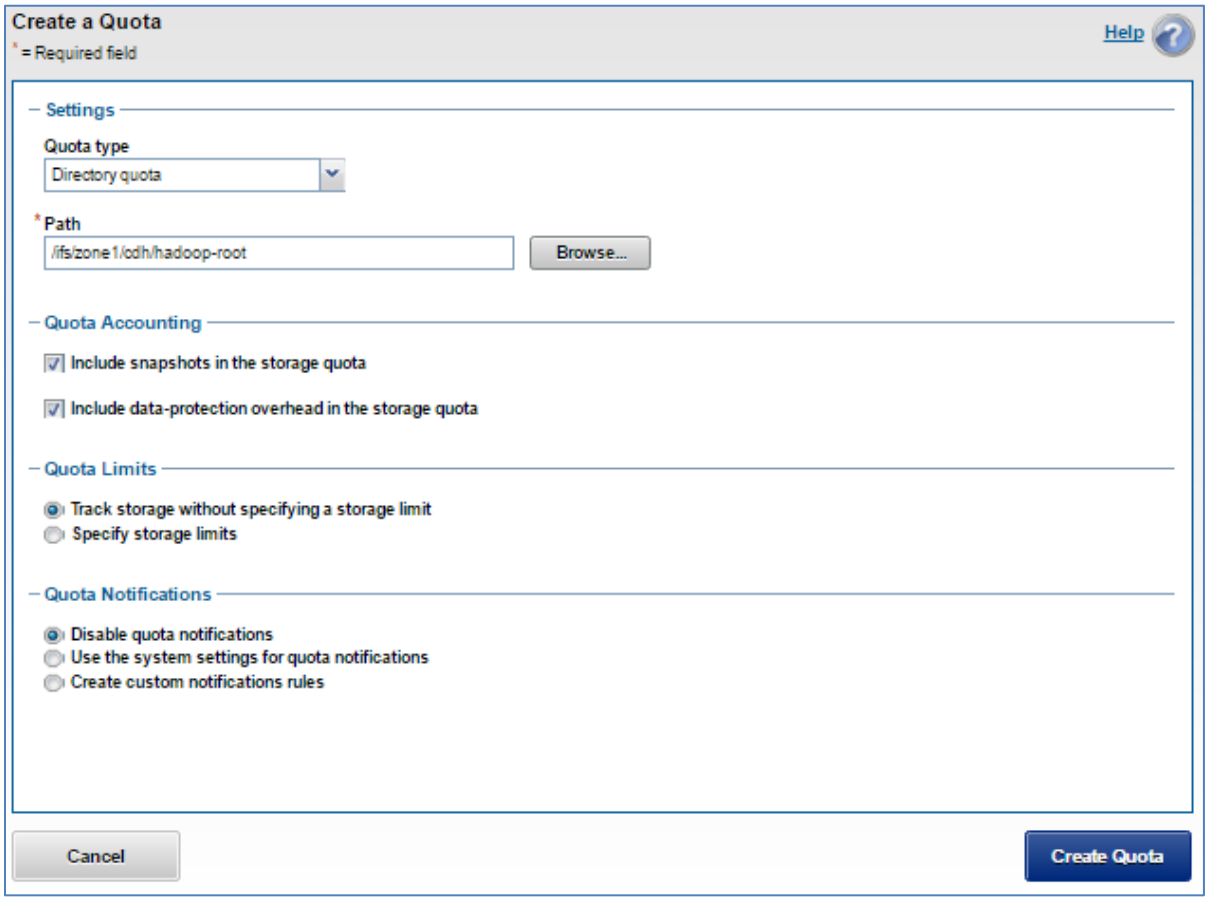


Figure 21: Directory accounting quota

As shown, the built in capabilities of quotas in OneFS provide an extremely powerful and flexible set of tools to monitor, manage, and limit storage usage on a cluster. This feature set can be especially useful when dealing with very large data sets. SmartQuotas frees the Hadoop and storage administrator from manually managing individual user utilization and capability not present in today's native directed-attached HDFS data storage.

Replication – SyncIQ

The Isilon OneFS SyncIQ module provides high performance asynchronous replication of data from one Isilon cluster to another. This enables you to develop data recovery and failover strategies in the event of the primary cluster becoming unavailable. SyncIQ is based on a policy which defines a directory on the source cluster that is replicated to a target OneFS cluster. The source cluster is the writable copy of the data, and all file changes are replicated to a read-only copy of the data on the target cluster. SyncIQ policies can be failover, failback, and fail reverted to support different operational requirements. The client data access is made through the source cluster to the data, and all data modifications are then replicated by SyncIQ to the target cluster; the target cluster does support read-only access to the data it contains if needed. This can be useful in some workflows that only require a read-only copy of the data. In this scenario, the target cluster can be used to offload traffic from your primary source cluster.

Local Target or Geographic Separation

SyncIQ can be optimized for either LAN or WAN based architectures in order to replicate over short or long distances. Typically these configuration are used to support two primary use cases:

- Local Target Cluster – Backup and data recovery to support business continuity
- Offsite Target Cluster – Geographically remote cluster to support Disaster Recovery

HDFS data sets can represent significant challenges in maintaining backup copies due to the size of the data. Once SyncIQ has completed its initial full synchronization, only changed files will be replicated, thereby minimizing traffic while maintaining full parity between the copies of the data. If needed, SyncIQ can also significantly increase the time to restore data. The target data can also be snapshotted to provide additional point-in-time recovery of data that may have been removed from the original source data set.

Data Replication

SyncIQ's primary capability is the ability to replicate any data defined within a SyncIQ policy to a remote cluster. The policy will define the data that is replicated, and how frequently changes made on the source data are replicated to the target OneFS cluster. The goal of any SyncIQ policy is meet the data recovery objectives of the business.

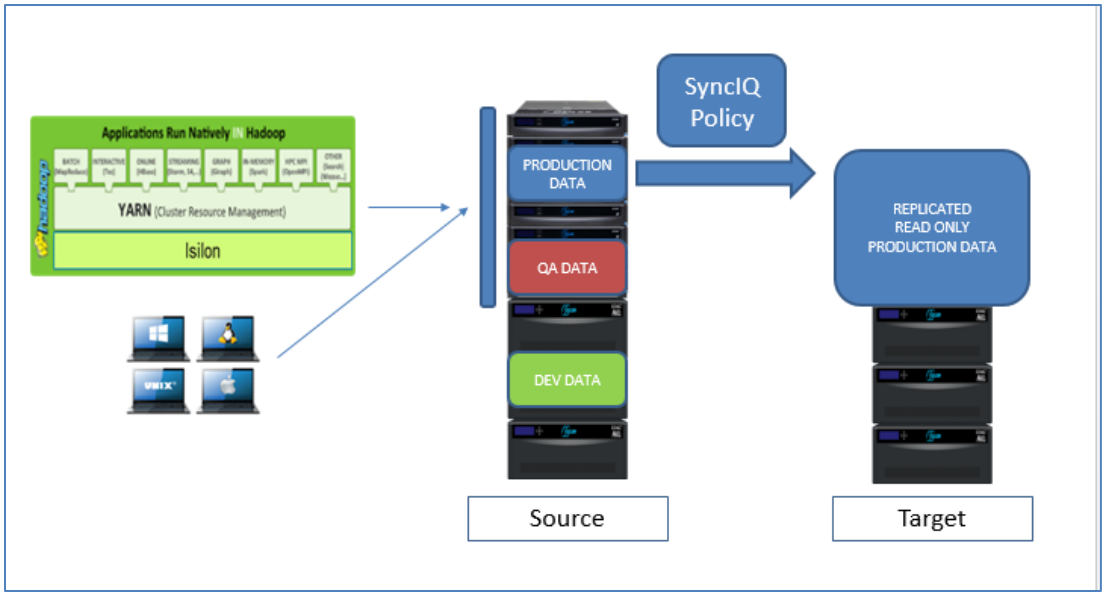
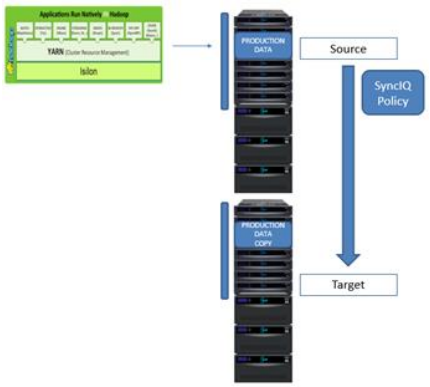
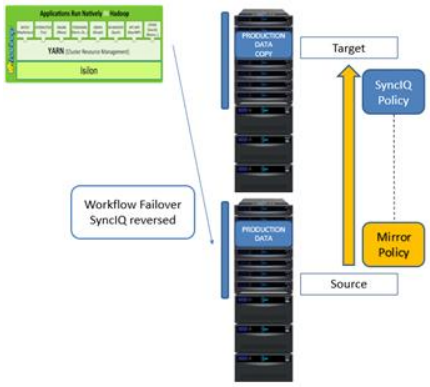
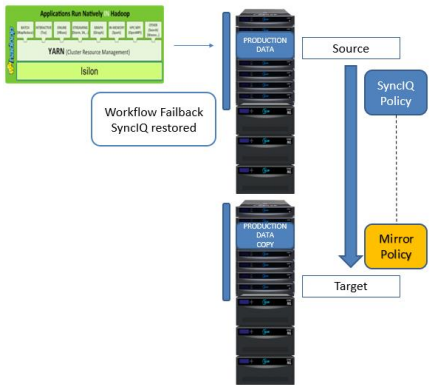


Figure 22: Using SyncIQ to replicate production data to a secondary source

SyncIQ policies define both the data to be replicated, the schedule and also the frequency of data replication. Multiple policies can be defined (up to 1,000 defined) and run simultaneously (50 running concurrently) to meet a broad range of recovery point objectives (RPO) and recovery time objectives (RTO).

Data Failover and Failback

In the event of the primary source cluster becoming unavailable, a SyncIQ policy can be failed over to the secondary target cluster. During this time, the administrators of the workflow will redirect all client activity to the mirrored target cluster. Workflow can now execute read and write operations against this target until the primary cluster is restored, at which point any new data is resynchronized back to the primary cluster with a failback workflow with a mirror policy.

		
Steady State: Primary – Source (RW)	FAILOVER: Primary – Source (RO)	FAILBACK: Primary – Source (RW)

Secondary - Target (RO)	Secondary – Target (RW)	Secondary - Target (RO)

Figure 23: OneFS SyncIQ failover and failback

The decoupling of the HDFS storage from the compute infrastructure introduces great flexibility since the SyncIQ replicated data is an exact copy of the source data. All data structures and permissions are replicated as-is, providing seamless integration to your compute cluster in the event of a failover. The compute cluster just needs to be redirected to the secondary cluster to gain access to the same HDFS data it was previously accessing. No underlying configuration ties the compute cluster to the primary HDFS data.

The capability to—write new data to the secondary cluster, and then replicate those changes back to your primary cluster (mirror policy), followed by a failback—provides business continuity that now removes many of the dependency of the Hadoop cluster managing the HDFS local storage through a local NameNode and DataNodes.

Disaster Recovery

With complete decoupling of data from the compute infrastructure even catastrophic events can be guarded against. Even with the entire loss of a site, services can be up and running quickly as all the HDFS data has been replicated and new compute infrastructure can be used to access it.

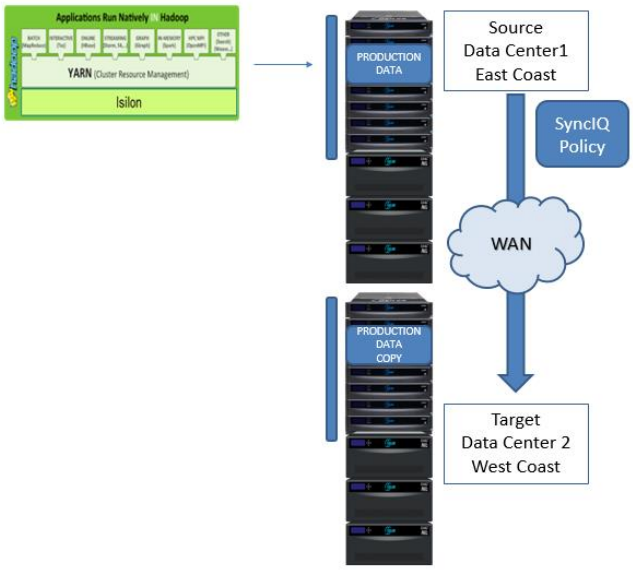
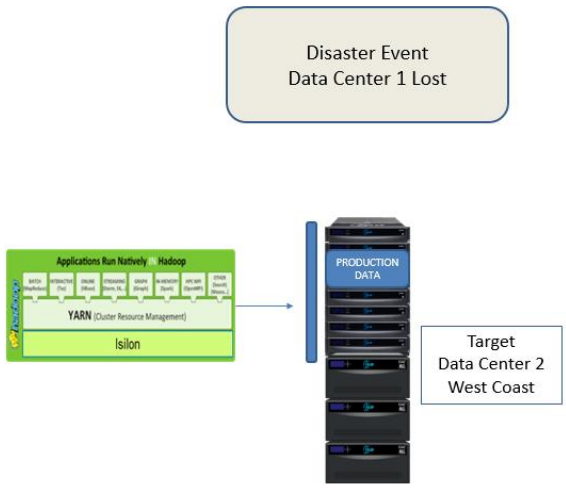
<table>
<tr>
<td> 
</td>
<td> 
</td>
</tr>
<tr>
<td>Steady State
1. Primary Data Center replicates to Secondary</td>
<td>Disaster Event occurs
1. Primary Data Center lost
2. Secondary Data Center assumes Primary role
3. New compute cluster built
4. Compute integrated with Isilon
5. Data Access continuity</td>
</tr>
</table>

Figure 24: Disaster Recovery with a geographically separate datacenter

SyncIQ policies are highly tunable and provide a high level of customization to manage the behavior and performance of a SyncIQ replication job and data transfer. One of the primary use cases and requirements with data replication is the ability meet explicit RPO and RTO objectives. SyncIQ supports three primary policy types to meet use cases from a point-in-time recovery of files from an off cluster copy to full recovery from a Disaster Recovery scenario.

Manual Replication

A SyncIQ policy is manually executed to create a point-in-time target copy of the source data on the target cluster.

Scheduled Replication

The most common and the most used policy is scheduled replication; the policy is scheduled to run on a defined scheduled basis to meet a RPO or RTO objectives as defined on the data being replicated.

Continuous Replication

Continuous replication in effect will replicate all changes to the target cluster as they occur on the source cluster. This type of policy is still considered asynchronous data replication and not HA. This type of replication is specifically suited to specific workflows and should not be used by most normal workflows which are better suited to scheduled replication.

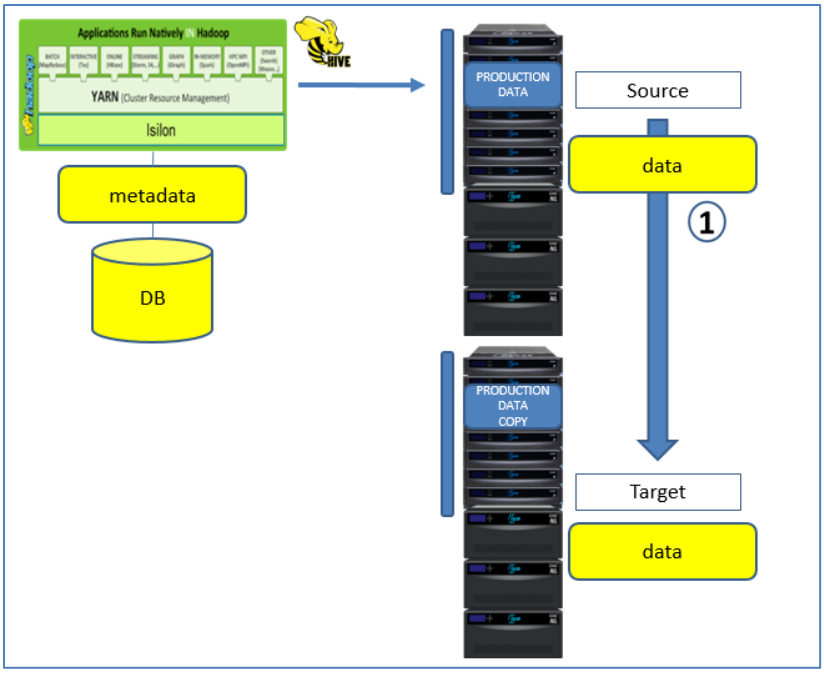
Additional Consideration for Hadoop

If leveraging OneFS for HDFS data storage and SyncIQ-based replication for data protection, consider the Hadoop cluster elements that do not reside natively on the HDFS file system. Predominately, these are the databases hosting service configuration and metadata in addition to the underlying HDFS data for Hadoop services (Cloudera Manger, Ambari, Hive, Impala, etc.). In this situation, simply replicating the underlying data does not fully protect the service. Additional planning and administration must be implemented to fully protect and replicate all the required data to support a service recovery.

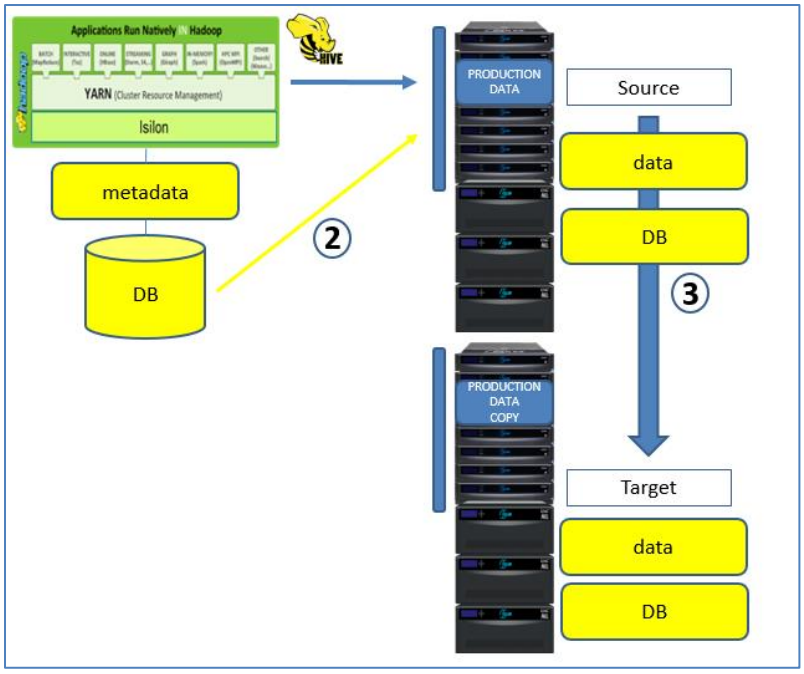
In these scenarios, OneFS can still be used to protect a Hadoop service and data—assuming the appropriate additional backup tasks are executed and data is replicated to OneFS.

An example would be Hive; Hive has two components we need to replicate to recovery from a failure.

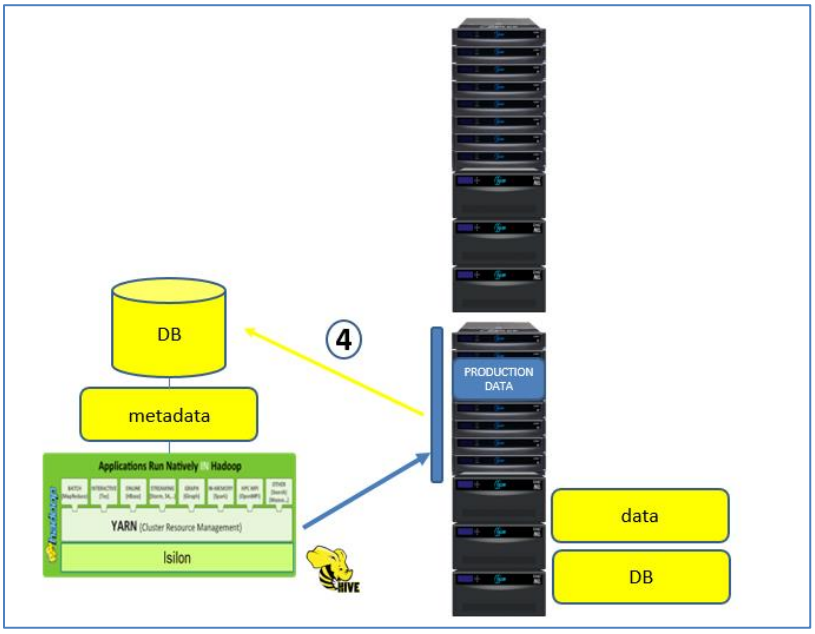
- The base Hive data tables are stored in OneFS, and are easily replicated by SyncIQ ①



- The Hive Metadata must also be protected, but it is currently stored in a database outside of HDFS
- The Hive metadata database is backed up and is copied to OneFS ② , this can be scheduled and automated from within the Hadoop cluster to meet RPO's



- SyncIQ can now replicate this backed up metadata data database ③



- In the event of a failover or DR event, the metadata can be restored from the SyncIQ replicated backup ④
- Hive data is available, and Hive is back online

Data Seeding

A useful feature of SyncIQ is the ability to run it locally within the cluster. An ideal use case is to create copies of production data in another location on the cluster for testing and destructive validation. Other methods can be used to copy data in this manner, but only with SyncIQ do you get the scale-out performance of the SyncIQ engine and control of the OneFS job engine.

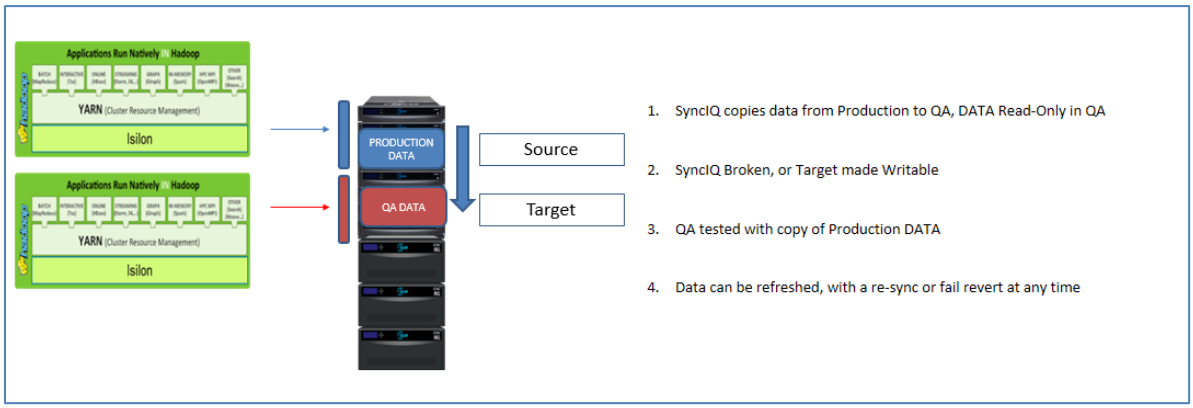


Figure 25: SyncIQ data seeding an Access Zone

In this scenario, we use an internal SyncIQ job to replicate a data set from the production data to the QA HDFS data path. The job is extremely efficient and replicates the data exactly as is to QA. Once the data is synchronized, the policy is broken or set to be writable at the target, and the data becomes writable and usable to the QA compute cluster.

At any time we can recreate the data in QA by recreating a new sync or executing a SyncIQ revert. A SyncIQ revert will discard any writes made to the data since the policy was made writable, basically returning the data to the original state. At this point, a sync could be used to update the data in QA if needed.

Additional SyncIQ Architectures

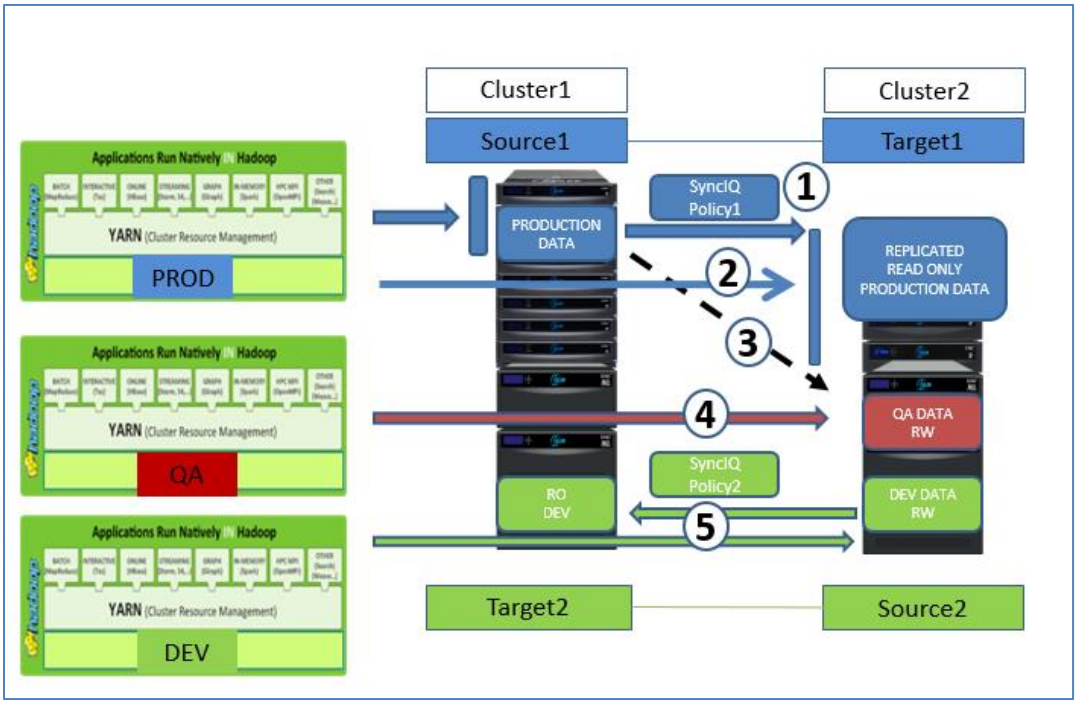


Figure 26: Multi cluster SyncIQ architectures

The above SyncIQ and cluster architecture illustrates a number of additional features of using SyncIQ to support Data Lake deployment strategies.

- Production data is replicated from cluster1 to cluster2 for Disaster recovery ①
- Read-only access to a copy of the production data accessed by the HDFS namespace URI directly, reduces load on Production Cluster ②
- Data Seeded to QA by SyncIQ, policy is made writable for QA testing ③, policy broken
- QA data resides as a writable copy on cluster2 ④
- Development data resides natively on cluster2. SyncIQ replicates back to cluster1 ⑤

This architecture clearly illustrates the flexibility SyncIQ brings to data management and the ease of replicating large data sets. Combined with the ability to failover and failback, we can optimize our cluster utilization with no impact to the Hadoop compute clusters.

SnapshotIQ

OneFS SnapshotIQ takes point-in-time copies of any file or directory within OneFS, either on-demand or through the use of rich policies and schedules. OneFS snapshots are highly scalable, very quick, and take little resources to create. Also, only changed blocks are stored between snapshots, ensuring high efficient storage utilization when storing snapshots. OneFS snapshots are per directory-based. This is unlike many traditional implementations of snapshots that are taken at the file system or volume boundary. Snapshots are relatively instantaneous, with just a small amount of work executed by OneFS to prepare the file system for the snapshot. The advantage is that the snapshot consumes close to zero space until file system operations occur after the snapshot was taken. In addition, when using OneFS SmartPools with storage tiers, snapshots can be stored on a different tier than where the original data is stored in order to maximize disk usage. When taking snapshots on our production data on our throughput node pool, we can use SmartPools to automatically locate the snapshots on the archive tier to optimize the disk usage in our hot tier.

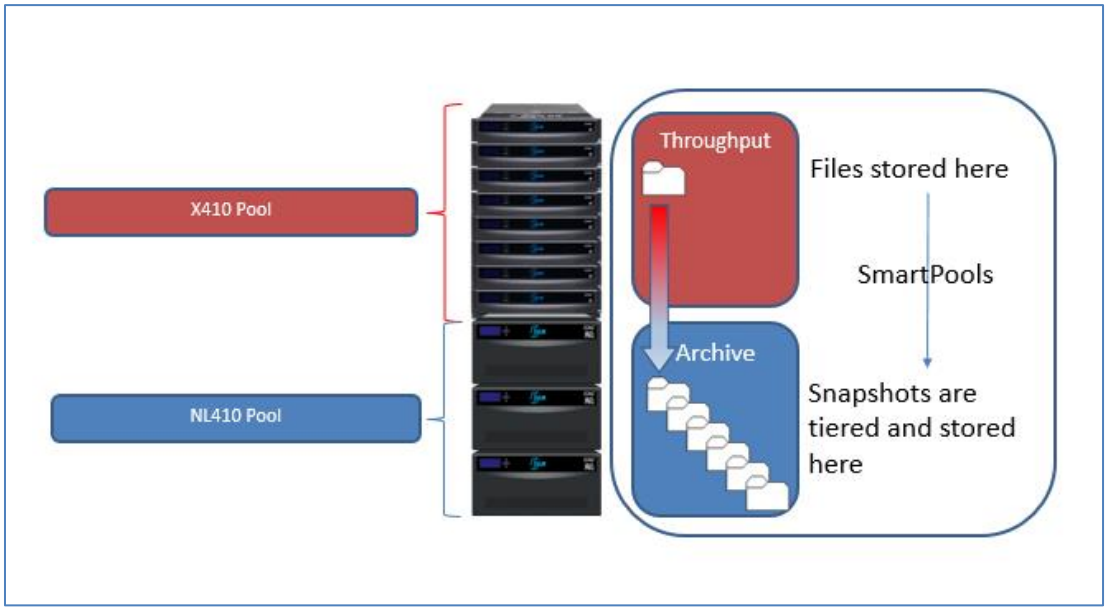


Figure 27: SmartPools tiering snapshot to a different node pool to optimize disk usage

Snapshot scheduling

SnapshotIQ provides advanced scheduling options with daily, weekly, monthly, and yearly options with single or multiple job frequency per the defined scheduled. The jobs can also support automatic deletion of old snapshots.

Snapshot deletes

Although SnapshotIQ can retain a very large number of snapshots, it often makes sense to remove unneeded or excessively old snapshots, as they no longer provide any value but may consume disk space that can be reclaimed. Snapshots should always be deleted under the control of the OneFS job engine and in reverse order—oldest to newest—in order to maintain lineage of data blocks within snapshots.

Snapshot Restore

File restoration of snapshots can be achieved in a number of ways to support your business requirements:

- Restore a file or directory from a snapshot through the OneFS command line.
- Clone a file from a snapshot through the OneFS command line.
- Access a file or directory from a snapshot through HDFS natively.
- Access a file or directory from a snapshot through NFS natively.
- Restore a file or directory from a snapshot through a SMB share. This will use the Microsoft Shadow Copy Client installed on your computer. You can use it to restore files and directories that are stored in snapshots directly.

User access to snapshots via a client using HDFS can be restricted from OneFS to make all recovery options an administrative task from the OneFS command line interface (CLI).

Snap Revert

SnapshotIQ also contains the ability to restore an entire directory back to its original or HEAD state of the initial snapshot. This efficient snapshot restoration of the entire contents of a directory to a previously known good state can provide additional data recovery capabilities in the event of data loss or data invalidation testing. A SnapRevert domain must be created on a directory prior to snapshots being created to leverage a Snap Revert, planning ahead with Snap revert can very quickly restore data to an original state after destructive testing.

SyncIQ – Target-aware Snapshots

An additional Snapshot capability is the ability to integrate Snapshot generation with SyncIQ replication. When a SyncIQ policy is configured to take target snapshots, OneFS will generate a snapshot on the target cluster on completion of the SyncIQ replication job. This can free space and resources on the source cluster thereby offloading snapshot and snapshot management to the target cluster.

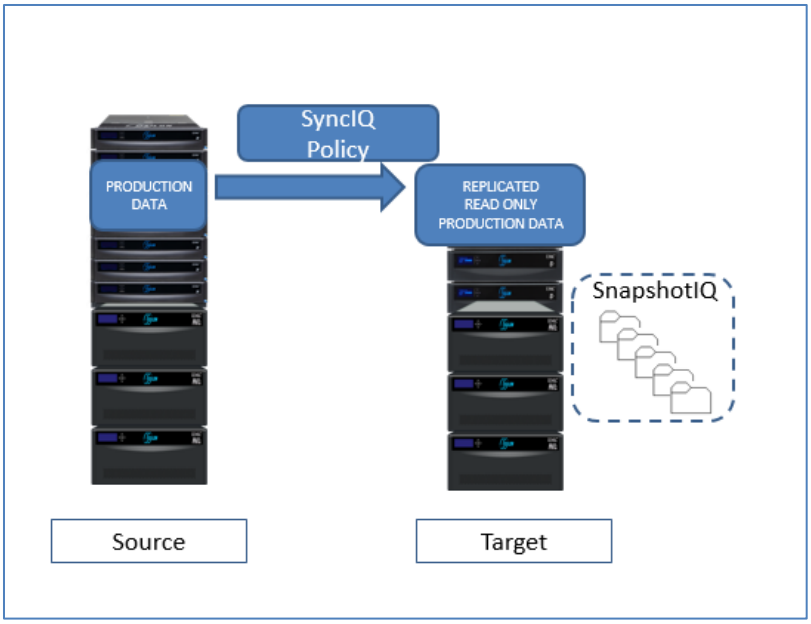


Figure 28: Target-aware snapshots

The ability to create additional snapshots on a replicated copies of the data opens the option to maintaining different snapshot retention policies without impacting storage overhead. It could meet two different data recovery requirements, for example:

- Source production snapshot – every 6 hours/5 day retention
- Target replicated snapshot – daily replication and snapshot/ 90 day retention

This gives you extended point-in-time recovery options using the native capabilities of SyncIQ and SnapshotIQ without additional management or administration.

Having reviewed on-cluster data protection mechanisms, let's take a look at additional OneFS features designed to protect data at scale.

Backup, Data Protection and Recovery

As the enterprise Data Lake grows, providing traditional backup and recovery of these new quantities of data is becoming extremely challenging. The previously relied upon data backup methodology, whether tape or disk, can no longer keep up and scale to support the volume and quantity of data being created and consumed. The high recovery time objectives and recovery point objectives (RPO and RTOs) —often involving a retrieval of tapes from secure, offsite storage, such as tape backup—is typically the mechanism of last resort for data recovery in the face of a disaster. With that being said, Isilon clusters and OneFS can provide additional features and capabilities to help facilitate large backups to traditional backup media.

Backup Accelerator

OneFS does support the ability to perform large-scale backup and restore functions across massive, single-volume data sets while leveraging an enterprise's existing, SAN-based tape and VTL infrastructure. This can be

enabled by the Isilon A100 Backup Accelerator (BA) node, which features a quad-port 4GB/s Fiber Channel card, quad-core processors, and 8GB of RAM.

A single Isilon A100 Backup Accelerator can concurrently stream backups at 480MB/s, or 1.7TB/hour, across its four Fiber Channel ports. Additionally, as data grows, multiple Backup Accelerator nodes can be added to a single cluster to support a wide range of RPO/RTO windows, throughput requirements, and backup devices.

Backup from Snapshots

In addition to the benefits provided by SnapshotIQ for snapshot point in time file recovery, SnapShotIQ also offers a powerful way to perform backups while minimizing the impact on the file system. Initiating backups leveraging a snapshot can provide substantial benefits. The most significant of these is that the file system does not need to be quiesced, since the backup is taken directly from the read-only snapshot. This eliminates lock contention issues around open file, and allows users full access to data throughout the duration of the backup job. SnapshotIQ can also automatically create an alias which points to the latest version of each snapshot on the cluster. This facilitates the backup process by allowing the backup to always refer to that alias. Since a snapshot is, by definition, a point-in-time (PIT) copy, by backing up from a snapshot the consistency of the file system or sub-directory is maintained.

This process can be further streamlined by using the Network Data Management Protocol (NDMP) snapshot capability to create a snapshot as part of the backup job, and then deleting it upon successful completion of the backup.

NDMP

OneFS facilitates performant backup and restore functionality through its support of the Network Data Management Protocol (NDMP). NDMP is an open-standard protocol that provides interoperability with leading data-backup products. OneFS supports both NDMP versions 3 and 4. The OneFS NDMP module includes the following functionality:

- Full and incremental backups and restores using NDMP
- Direct Access Restore/Directory Direct Access Restore (DAR/DDAR), single-file restores, and three-way backups
- Restore-to-arbitrary systems
- Seamless integration with access control lists (ACLs), alternate data streams, and resource forks
- Selective file recovery
- Replicate then backup
- Multi-stream NDMP backup

While some backup software vendors may support backing up OneFS over SMB and NFS, the advantages of using NDMP include:

- Increased performance
- Retention of file attributes and security and access controls

- Backups utilize automatically-generated snapshots for point-in-time consistency.
- Extensive support by backup software vendors

OneFS provides support for NDMP version 4, and both direct NDMP (referred to as 2-way NDMP), and remote NDMP (referred to as 3-way NDMP) topologies.

Direct NDMP model

This is the most efficient model and results in the fastest transfer rates. Here, the data management application (DMA) uses NDMP over the Ethernet front-end network to communicate with the Backup Accelerator. On instruction, the Backup Accelerator, which is also the NDMP tape server, begins backing up data to one or more tape devices which are attached to it through Fiber Channel.

The Backup Accelerator is an integral part of the Isilon cluster and communicates with the other nodes in the cluster through the internal InfiniBand network. The DMA, a separate server, controls the tape library's media management. File History, the information about files and directories, is transferred from the Backup Accelerator through NDMP to the DMA, where it is maintained in a catalog.

Direct NDMP is the fastest and most efficient model for backups with OneFS and obviously requires one or more Backup Accelerator nodes to be present within a cluster.

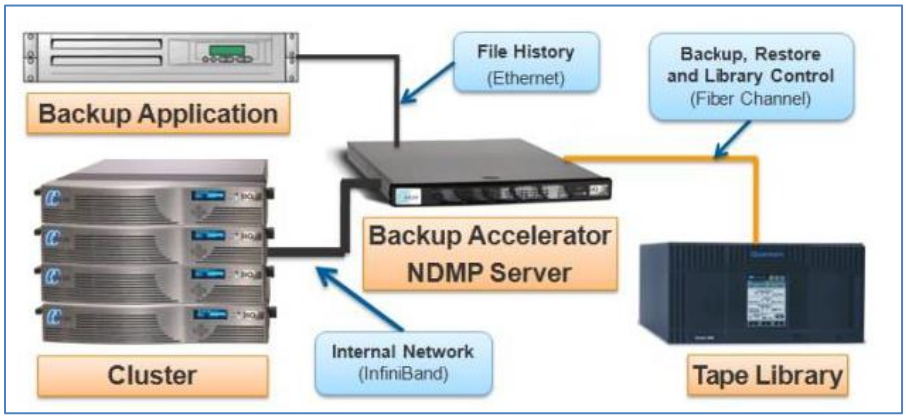


Figure 29: Two-way NDMP backup with backup accelerator

Remote NDMP Model

In this model, there is no Backup Accelerator. The DMA uses NDMP over the LAN to instruct the cluster to start backing up data to the tape server, which is either attached to the LAN or directly attached to the DMA host. In this model, the DMA also acts as the Backup/Media Server. During the backup, file history is transferred from the cluster through NDMP over the LAN to the backup server, where it is maintained in a catalog. In some cases, the backup application and the tape server software is on the same server.

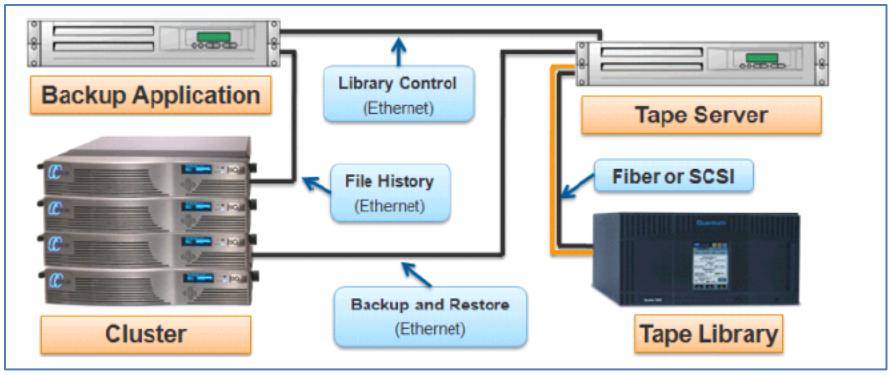


Figure 30: Three-way NDMP remote backup

Additional features supported by OneFS:

- Incremental Backups – supports backup levels 0-9
- Direct Access Recovery – ability to go directly to the location of a file within an archive and quickly recover
- Directory DAR – ability to go directly to the location of a directory within an archive and quickly recover
- Selective File Recovery – NDMP recovery of a subset of files within a backup archive.
- Alternate Path Restore – supports the ability to restore to alternate path locations.

Backup and Restore Architectures

The ability to utilize multiple independent features on the same data set allows the data administrator to create multiple levels of data protection to backup and restore data, depending on the business requirements that are defined on the data.

A multi-tier approach to data backup and recovery may look something like this in an OneFS cluster.

- SyncIQ – Replication ①
    - Failover
    - Recovery of data from the replication target
- SnapshotIQ – ②
    - Point-in-time recovery of files
    - User self-service or administrator-managed restores
    - Snap Revert
    - Snapshot based backups

- Offline backup – tape or VTL ③
- SyncIQ Target Aware Snapshots ④

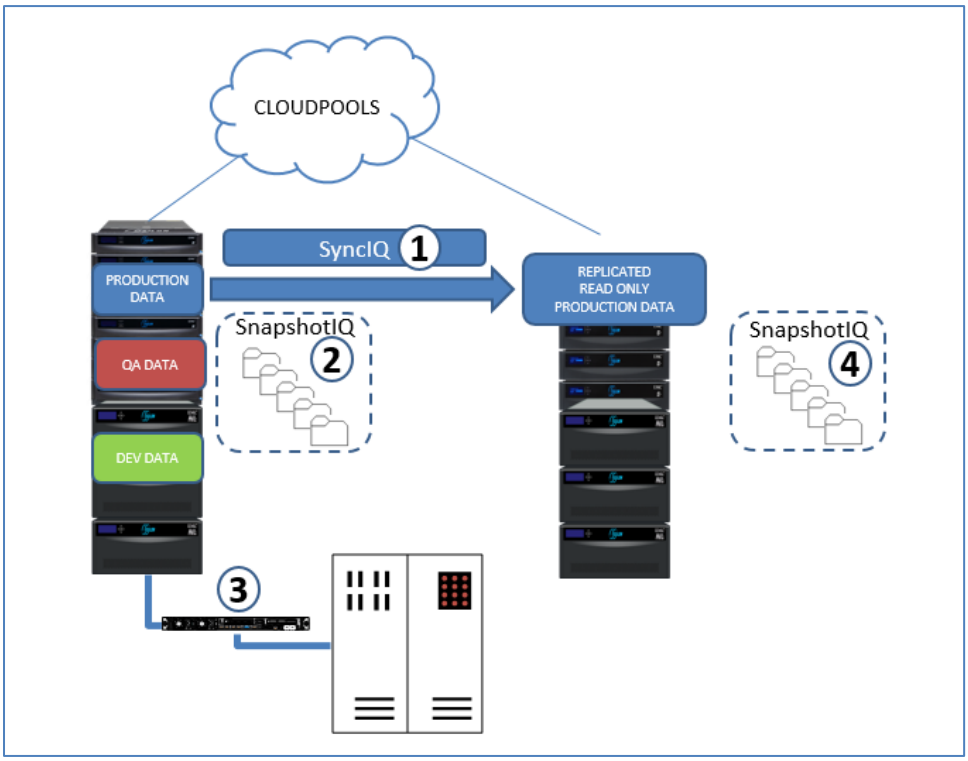


Figure 31: Multi architecture data backup and restore

Since all these capabilities are available at the directory level, the flexibility to craft multilevel protection and recovery schemes on any data in the cluster gives the data administrator capabilities far exceeding native Hadoop backup tools.

Multiprotocol

One of the foundational capabilities of OneFS is the ability to provide access to any data from any protocol OneFS supports (HDFS, NFS, SMB, FTP, HTTP and Swift). Since OneFS presents a unified permission model as discussed earlier, the ability to securely read or write to any common data is the cornerstone of the enterprise Data Lake. The ability to write directly to OneFS and then access the same data set securely can entirely remove any data load or ingestion processes that may exist with native HDFS data stores. Isilon OneFS is the only Hadoop-compatible product on the market to provide this level of protocol integration.

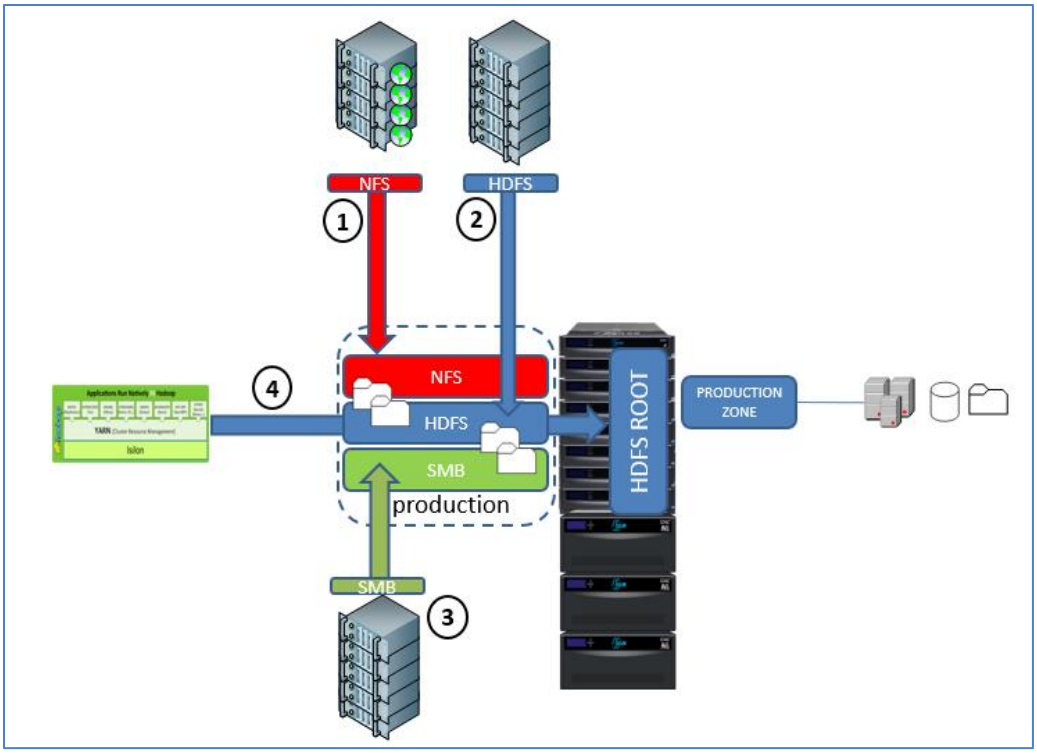


Figure 32: Multiprotocol data access to an Access Zone

An ideal multiprotocol workflow for OneFS is log file ingestion.

- Web servers all mount directories directly on OneFS using NFS exports and write log files directly to directories located in the HDFS root ①
- Application Server logs are written natively with HDFS directly to OneFS ②
- Database Server backup SQL tables written with SMB directly to OneFS ③
- All the ingested data is immediately accessible to the Hadoop compute cluster and all the Hadoop services running ④

Hadoop tools can now natively access this data through HDFS without any additional data movement or transport. This ability to limit server generated data needing to be copied into the HDFS file system can reduce the time that was previously needed to load data between silos and also dramatically reduce the overall space consumed. This protocol-agnostic secure access model illustrates that the Isilon OneFS Data Lake has truly become the scale-out storage platform for all data in the enterprise. Since OneFS is true scale out, we also remove the need to purge data from the valuable HDFS storage space, using the additional capabilities we have with OneFS it becomes much easier to manage our data storage through consolidation to a single platform.

SmartDedupe

Many analytic-based workflows often yield duplicate data sets being used and created. Users often create their own copy of a particular data set to run against their own custom workflow. Considering the size of many of these data sets, this can represent a large and inefficient use of storage. Implementing a deduplication strategy with OneFS SmartDedupe can provide storage efficiencies and maximize the storage capacity of the cluster.

OneFS deduplication is applied at the directory level, targeting all files and directories underneath one or more root directories. Using SmartDedupe, you have the ability to first assess a directory for deduplication and then determine the estimated amount of space you can expect to save. You can then decide whether to deduplicate the directory. After you begin deduplicating a directory, you can monitor how much space is saved by deduplication in real time. Deduplication is most effective when applied to static data sets or archived files and directories.

File Clones

OneFS file clones provide an efficient mechanism for generating multiple read/write copies of files. File clones provide space efficiency and rapid provisioning by logically creating a second copy of a file that shares common blocks between the original file and the clone. Clones are ideal for seeding data for multiple users with the same large file.

Audit

OneFS provides both system configuration and protocol event audit capabilities. All audit data is stored and protected in the cluster file system and organized by audit topics. Auditing can detect many potential sources of data loss, including fraudulent activities, inappropriate entitlements, and unauthorized access attempts. Protocol auditing tracks and stores activity performed through HDFS protocol connections and Access Zones in a cluster. If you enable protocol auditing for an Access Zone, file-access events through the SMB, NFS, and HDFS protocols are recorded in the protocol audit topic directories. You can specify which events to log in each Access Zone. For example, you might want to audit the default set of protocol events in the production Access Zone but audit only successful attempts to delete files in a different Access Zone.

Note: For the NFS and HDFS protocols, the rename and delete events might not be enclosed with the corresponding create and close events. Also the EMC Common Event Enabler (CEE), which is used to aggregate and forward events to enterprise audit tools does not currently support HDFS protocol event forwarding.

Role Based Access Control (RBAC)

OneFS fully supports the capability to delegate administrative control through Role Based Access Control (RBAC). Through the use of roles and privileges, administrative access can be designated in place of root user access. Selected users and groups can be assigned the right to perform a particular administrative action that has been delegated to them through a Security Administrator. RBAC enables individual Hadoop cluster administrators the ability to be designated a subset of administrative access to OneFS in order to manage configuration related to the HDFS data store and other OneFS features without requiring root level access.

SmartLock

Immutable storage or write once, read many (WORM) is provided by the OneFS Smartlock file locking capability. The ability to provide tamper-proof data archiving for critical data for disaster recovery or

compliance reasons is available through a simple-to-manage and automated feature set within OneFS. The use of WORM directories with HDFS data flows should be evaluated for suitability, Read-Only workflows will be compatible, but many workflows that are writing to the data store will encounter issues as renames and move operations will fail.

InsightIQ

The InsightIQ (IIQ) virtual appliance monitors and analyzes the performance of your Isilon cluster to help you optimize storage resources and forecast capacity.

Cluster Performance Monitoring

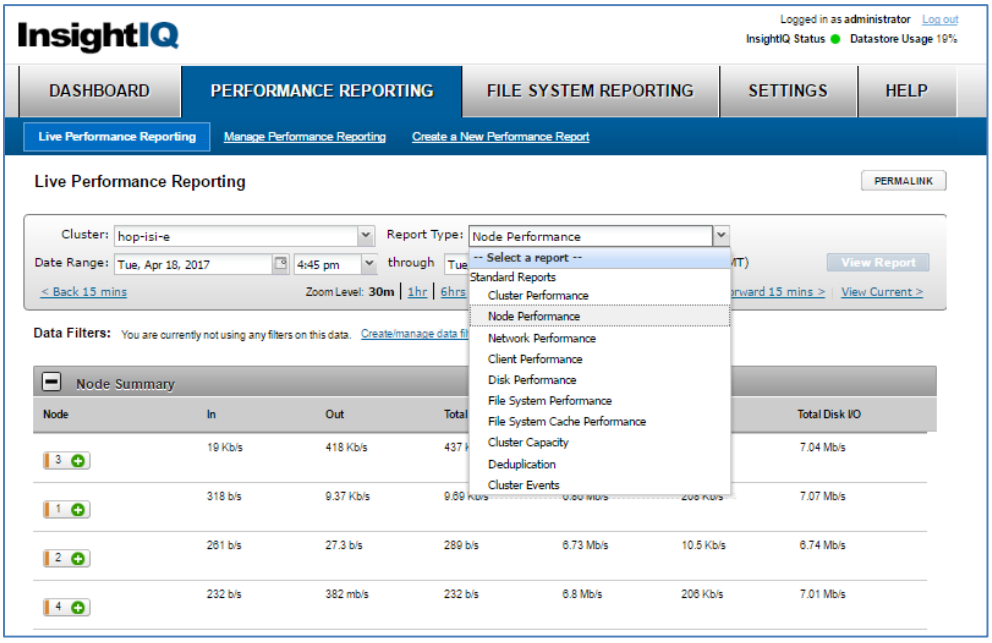


Figure 33: Performance Monitoring in IIQ

Using IIQ, the live performance of the cluster can be monitored and observed as workflows are executed against OneFS. The ability to drill down and filter extensively into specific areas of OneFS can provide real time insight into how a workflow and the cluster is operating.

File System Monitoring

Using IIQ File System Reporting features, you can monitor and observe: Capacity Planning, File System Analytics, Deduplication Reporting, and Quota Reporting as shown:

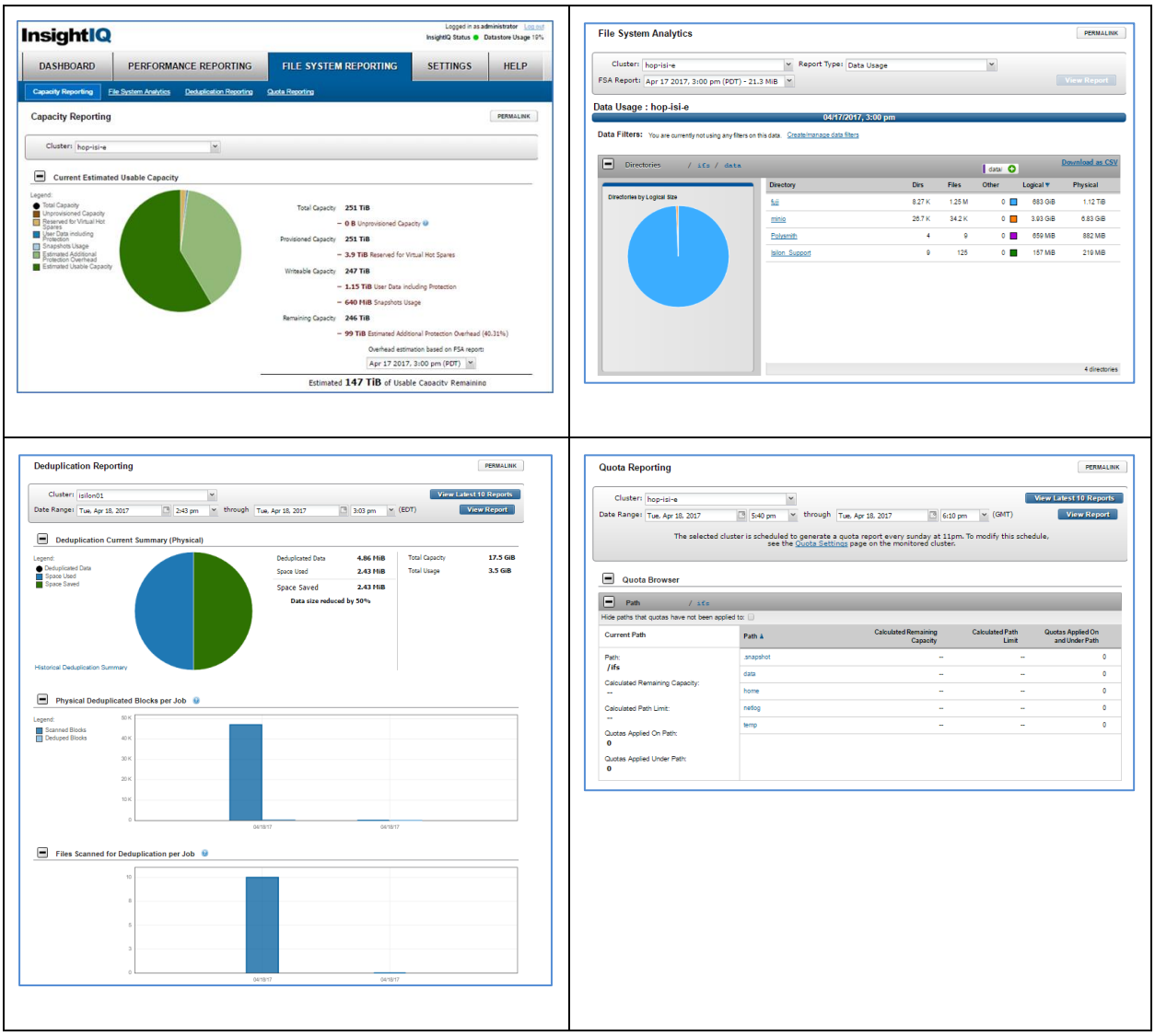


Figure 34: File system reporting

InsightIQ provides an extremely rich and powerful interface to understand the workflow, identify hotspots, and tune performance. The ability to monitor data utilization from a directory and user layout perspective—including quota and deduplication statistics—allows IIQ to provide detailed storage utilization and forecast information. Providing you the ability to anticipate and grow storage when needed to meet demands.

Data at Rest Encryption

OneFS provides support for the security of data at rest. It involves dedicated storage nodes containing self-encrypting drives (SEDs), in combination with an encryption key management system embedded within OneFS. Data is encrypted on disk using the AES-256 cipher, and each SED has a unique data encryption key (DEK) which is used to encrypt and decrypt data as it's read from and written to disk. OneFS automatically generates an authentication key (AK) that wraps and secures the DEK. This means that the data on any SED

which is removed from its source node cannot be unlocked and read, thereby guarding against the data security risks of physical drive theft.

The Isilon Data Encryption at Rest solution also allows SED drives to be securely wiped before being re-purposed or retired, using cryptographic erasure. Cryptographic erasure involves 'shredding' the encryption keys to wipe data, and can be done in a matter of seconds. To achieve this, OneFS irreversibly overwrites the vendor-provided password, or MSID, on each drive, resulting in all the on-disk data being scrambled.

Isilon encryption of data at rest satisfies a number of industries' regulatory compliance requirements, including U.S. Federal FIPS 140-2 Level 2 and PCI-DSS v2.0 section 3.4.

HDFS Wire Encryption

HDFS wire encryption enables the transmission of encrypted data over the HDFS protocol between OneFS and HDFS clients. HDFS wire encryption enables OneFS to encrypt data that is transmitted between OneFS and HDFS to meet regulatory requirements. Wire Encryption uses Advanced Encryption Standard (AES) to encrypt the data. 128-bit, 192-bit, and 256-bit key lengths are available. This feature is Access Zone aware and leverages Kerberos authentication to manage the transport encryption.

Monitoring and Alerting

Using the OneFS dashboard from the web administration interface, you can monitor the status and health of the OneFS system. Information is available for individual nodes, including node-specific network traffic, internal and external network interfaces, and details about node pools, tiers, and overall cluster health. A rich event and alerting framework is available to administrators to receive and manage notification from the cluster for administrative notifications or maintenance tasks that need to be addressed.

ESRS

Isilon OneFS is fully integrated and supported with EMC Secure Remote Services (ESRS) for remote support and alerting activities.

Treedelete

Delete very large quantities of data very quickly with this multimode highly scalable delete job managed by the OneFS job engine.

Integrity Scan

A cluster-wide data validation and protection mechanism to validate data integrity.

Permission Repair

A highly parallel and scalable process to correct and set permissions on large file trees efficient and quickly.

OneFS – The True Enterprise Data Lake

As the enterprise Data Lake continues to grow, it can become extremely challenging to backup and protect data efficiently and quickly enough to support everything from a single file recovery to a complete data center failover in response to a DR event. As we have seen, Isilon OneFS provides a complete set of enterprise features to fully protect all the data it contains in many different integrated ways. These features far out-

achieve what is capable with normal HDFS data storage today and would require many different tools and administrative tasks to achieve. OneFS provides a single point of administration, with many of the required operations handled automatically, schedulable, and massively configurable.

Hadoop Capability	HDFS based Hadoop Cluster	Isilon OneFS - HDFS
Data Protection	3 x mirroring	High efficiency erasure coding
NameNode Redundancy	NameNode HA Secondary NameNode	Every Isilon node acts as a NameNode
Scale out data storage	Add/manage many storage nodes	Add high-capacity storage nodes into the existing cluster
Upgrade and compute integration	Fully integrated-data tied to NameNode	Data is fully decoupled from compute
Multiple Protocol Access	HDFS and NFS	HDFS, NFS, SMB, HTTP, FTP, and Swift
Multiple HDFS roots	Multiple Hadoop cluster installs	Access Zone/SmartConnect
Load balancing traffic	Manual configuration	SmartConnect
HDFS data Tiering	HDFS tiers (not automatic)	Node Pools – different performance tiers SmartPools – automatic, job based Cloudpools – local and cloud File Pool Policies – management data locality
Snapshots	HDFS Snapshots	SnapshotIQ SnapRevert
Data Replication and DR support (local and geographical)	Replication cluster	SyncIQ – failover, failback, revert SyncIQ – WAN replication
Data Backup	-	NDMP
Data Deduplication	-	SmartDedupe
Data and User Quota	-	SmartQuotas – accounting, soft, hard
File Clones	-	OneFS File Clones
Audit	-	OneFS Audit
WORM	-	SmartLock
Secure Remote Support and Access	-	ESRS
Storage growth monitoring	-	InsightIQ

Table 6: Hadoop capabilities versus OneFS

This paper does not address the full capabilities of each OneFS feature in depth, but many of the OneFS features that do have an equivalent Hadoop cluster capability will contain many more enterprise-grade capabilities not to be found in the native Hadoop tools. The centralized availability of these features being managed and administered from a single Isilon cluster will also make the protection and administration of the Data Lake easier and provide confidence in your ability to make data available to the ever increasing demands of the enterprise.

Conclusion

As can be seen in this whitepaper, the enterprise storage features of Isilon OneFS provide additional capabilities to the data management of Hadoop data in large data lake deployments. The benefits of locating the HDFS data on OneFS provides the storage and hadoop administrators with the ability to manage the data from a single location and with many enterprise features still unavailable or not native to traditional DAS deployments of HDFS. This uniquely positions OneFS as the storage centric solution for Hadoop and analytic deployments.

Appendix

Compatibility information

Hadoop Distributions and Products Supported by OneFS

Information specific to Isilon

Using Hadoop with Isilon - Isilon Info Hub

OneFS

HDFS Reference Guide

OneFS CLI Administration Guide

OneFS Web Administration Guides

Multiprotocol

Isilon Multiprotocol Concepts Series

Contacting EMC Isilon Technical Support

Online Support: https://support.emc.com/

Telephone Support:
United States: 800-782-4362 (800-SVC-4EMC)
Canada: 800-543-4782
Worldwide: +1-508-497-7901
Additional worldwide access numbers

Help with Online Support Tools:

For questions specific to HDFS, contact support@emc.com