IBM Cloud Object Storage System

Splunk SmartStore with IBM Cloud Object Storage

Architecture, Configuration and Integration Guide

Version 1.1

Table of Contents

Revision History

Version	Date	Description
1.1	10/4/2019	Publication

Integration information

Splunk Review	7/1/2019
Target On-Premise ClevOS Version	ClevOS 3.14.3.15
Target IBM COS public cloud	Current release(Regional ClevOS 3.14.4.96 /Cross Regional 3.14.4.96)
Target Partner Product Version	Splunk Enterprise 7.3.0

Introduction

This document provides information on the interoperability between Splunk Enterprise's SmartStore feature that was released with Splunk version 7.2 and IBM Cloud Object Storage for both On-Premise installations and public cloud in terms of functionality. The actual procedure of integrating these two entities is documented in the configuration guide which is in Appendix A.

Intended Audience

This document is intended for anyone interested in understanding the functional aspects of the integration of Splunk with IBM Cloud Object Storage (COS).

Scope

This document assumes that the readers are aware the functional aspects of Splunk Enterprise and wants to understand how Splunk's SmartStore feature works with IBM Cloud Object Storage. The intention of the document is to cover ***Splunk Enterprise's SmartStore*** functionality with IBM Cloud Object over the published Cloud Object Storage API.

This document only covers testing ISV specific features and functionality at the time of the qualification. Hardware, software and overall environmental differences can affect overall performance.

Splunk Enterprise with the addition of the SmartStore feature now has native S3 integration with IBM Cloud Object Storage and serves as the warm tier for indexed data to Splunk for a highly scalable cost-effective data storage solution.

Terminologies

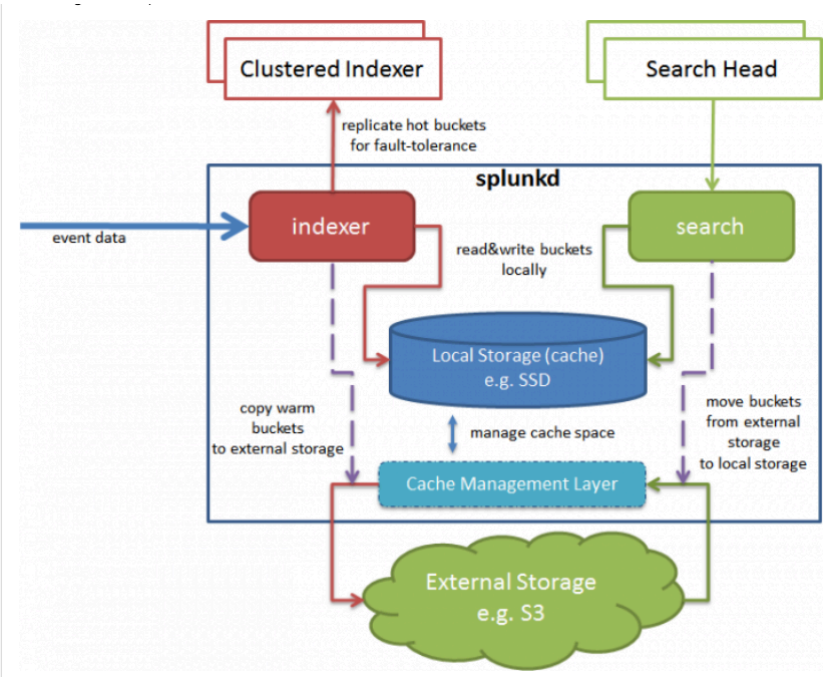
The following terms are helpful in understanding the Splunk/IBM COS solution:

- **Search Head:** Manages searches
- **Indexer:** Index and store data. Indexers also search across data.
- **Forwarder:** Ingest raw data and forward the data to another component, either another forwarder or an indexer.
- **Splunk S2:** Splunk SmartStore
- **Indexer cluster:** A specially configured group of Splunk indexers that replicate external data, so that they maintain multiple copies of the data. Indexer clusters promote high availability and disaster recovery.
- **Search head cluster:** A group of Splunk search heads that serve as a central resource for searching. The search heads in a cluster are interchangeable. You can run or access the same searches, dashboards, knowledge objects, and so on, from any member of the cluster. To achieve this interchangeability, the search heads in the cluster share configurations, apps, search artifacts, and job loads.
- **TSIDX reduction:** A process that eliminates all full-size tsidx files in a bucket and replaces them with mini versions of those files. These mini versions contain only some essential metadata. The resulting reduced buckets are smaller in size but slower to search. TSIDX reduction is not compatible with SmartStore.
- **Deployment:** A set of distributed Splunk Enterprise instances, working together. A typical deployment scenario consists of a number of forwarders and one or more indexers, with the forwarders sending data to the indexers to index and search. Distributed search is another type of Splunk Enterprise deployment. A single deployment scenario might combine both forwarding and distributed search.
- **Archiving:** The action of adding to and maintaining a collection of historical data. In Splunk Enterprise, you can define archiving policy to fit the needs of your organization. You can specify that indexed data be archived according to the size or age of an index. If archiving to NFS or S3, the archived data is no longer searchable. If archiving to HDFS, the archived data is searchable via mapreduce.
- **Splunk Cloud:** Splunk Cloud is a Splunk-hosted and operated SaaS solution currently using AWS infrastructure. Primarily in US-East(Virginia), but also available upon request in other availability zones. Splunk has tested SmartStore on Splunk Cloud prior to SmartStore being generally available.
- **Splunk Enterprise Security:** A premium Splunk application that is licensed independently from Splunk Enterprise. Splunk Enterprise Security (ES) is a security information and event management (SIEM) solution that provides insight into machine data generated from security technologies such as network, endpoint, access, malware, vulnerability and identity information.
- **Cloud storage:** TCP/IP based on-premise or cloud-based storage pools
- **IDA:** Information Dispersal Algorithm
- **Manager:** performs FCAPS for IBM COS (Fault management, Configuration management, Accounting management, Performance management, Security management)
- **CSO:** Cloud Storage Object(IBM's S3 compatible interface)
- **Accesser:** presents a stateless S3/CSO interface to Splunk
- **Slicestor:** IBM COS appliances that store slices of erasure coded object

- **External Agent:** IBM COS has a service called “External Agent” that allows the installation of custom software on ClevOS appliances. This is utilized by our largest customers to allow Splunk Universal Forwarders to send logs to Splunk Enterprise so that customers can monitor IBM COS performance.
- **IBM Cloud Object Storage Archive Tier:** IBM® Cloud Object Storage Archive Tier is our lowest cost option for data that is rarely accessed. You can store data by transitioning from any of the storage tiers (Standard, Vault, Cold Vault and Flex) to long-term offline archive or use the online Cold Vault option. In order to access an archived object on the archive tier, one must programmatically restore it to the original storage tier.

Solution Introduction

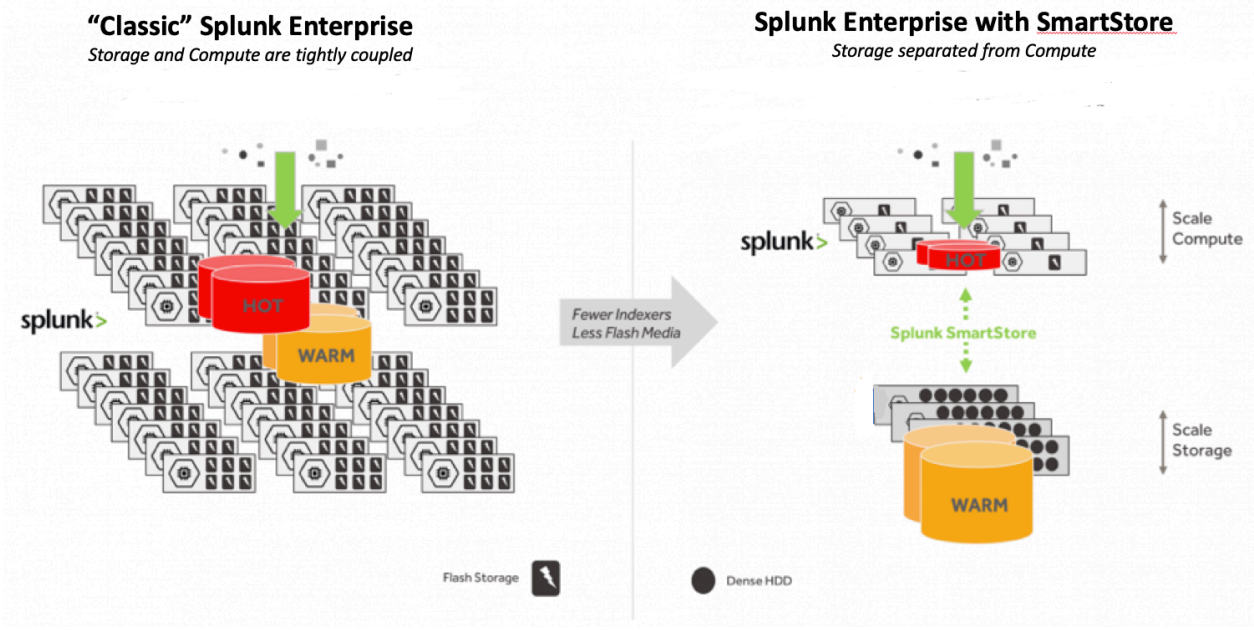
We have validated Splunk's native S3 integration with IBM COS in Splunk Enterprise 7.3.0 with the SmartStore(S2) feature. This feature allows Splunk customers to scale storage independently from compute. Using SmartStore with IBM Cloud Object Storage allows your Splunk instances to store/access petabytes of storage without having to add additional Splunk indexers. Both IBM Cloud Object Storage on premise, and IBM Cloud Object Storage in our public cloud are supported.



With this new feature, Splunk allows customers to extend their ***Splunk Local Storage*** with IBM Cloud Object Storage as a ***Warm Bucket***. Since most Splunk searches will be on recently indexed data, this allows for fast searches from local storage. Splunk's Cache Management Layer is central to the SmartStore data flow, and it evicts older/less searched copies of indexed data from cache. The Cache Manager also fetches copies of indexed data from IBM COS, as necessary, to handle search requests.

The integration of Splunk's SmartStore with IBM Cloud Object Storage allows clients to minimize the amount of data on local storage, while maintaining the fast indexing and search capabilities characteristic of Splunk Enterprise deployments. Splunk claims that except in a few uncommon scenarios, indexers return search results for SmartStore enabled indices with speeds similar to those for non SmartStore indices.

Here's another way to show how Splunk's SmartStore feature decouples compute from storage, and shows IBM Cloud Object Storage storing Splunk's Warm buckets on cost effective HDDs:



Splunk with SmartStore changes the deployment model by rendering indexers stateless for warm and cold data, making the indexer highly elastic and simplifying the management of a Splunk deployment. Not only is it easier to scale Splunk indexers as necessary, Splunk with SmartStore also allows easy migration of your indexers to higher performance hardware, without having to worry about the typical storage migration.

Splunk with SmartStore also provides more options than the normal Splunk cluster backup and restore processes by allowing an entire set of indexers in the cluster to be shutdown/replaced and later revived by bootstrapping the data from IBM Cloud Object Storage.

Solution Scope

In our solution certification, we focused on the use case which we see as the best fit for a COS solution.

- Splunk SmartStore(S2) warm bucket

We will characterize Splunk Enterprise search/query functionality with the SmartStore workflow for both IBM Cloud Object Storage on-premise and IBM Cloud Object Storage Public Cloud.

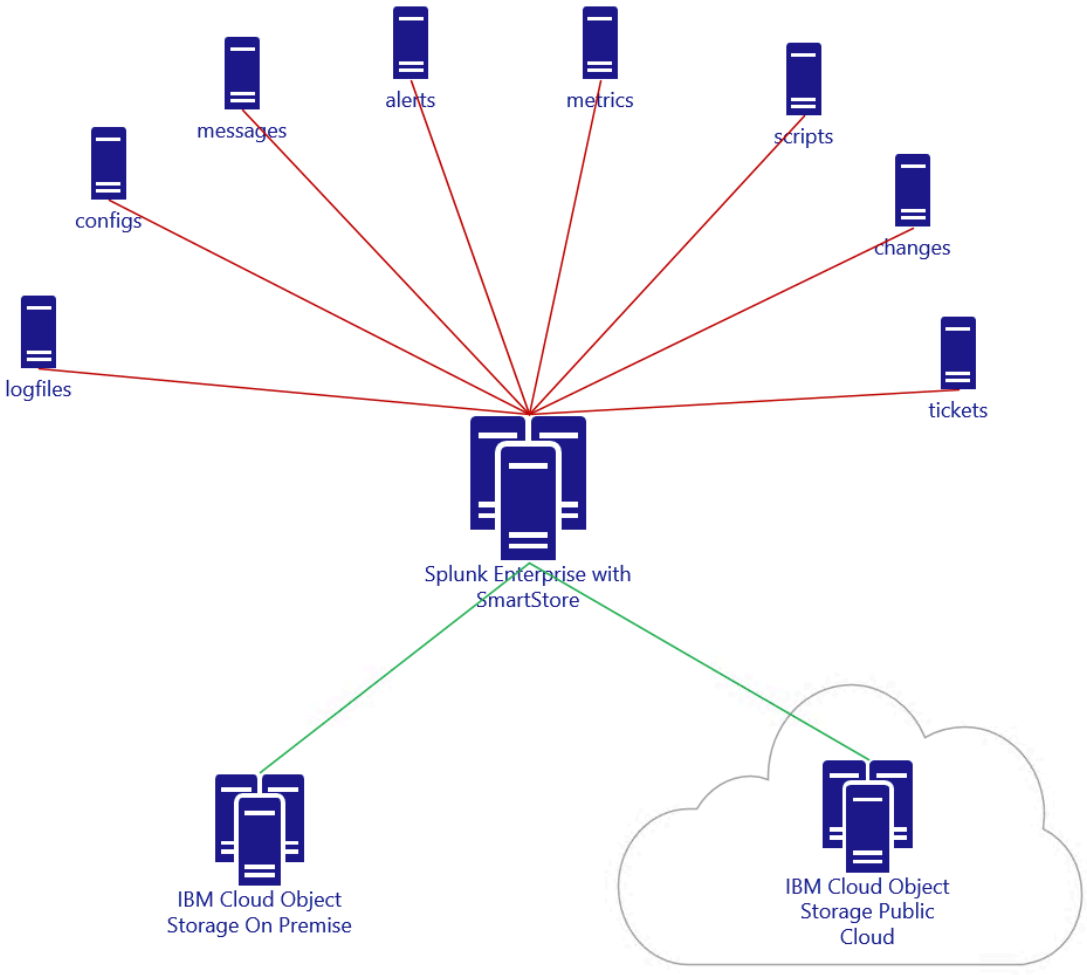
We have validated Splunk Enterprise w/ SmartStore against both our regional and cross regional offerings. We have also validated Splunk Enterprise w/ SmartStore against all our storage class choices (Standard, Vault, Flex, and Cold Vault), with the exception of our Archive Tier.

The solution will not cover other back-end cloud storage providers that are supported by Splunk Enterprise w/ Smart Store such as Amazon AWS and Microsoft Azure.

We also wanted to mention that we do see our largest on premise IBM COS customers already utilize IBM COS's "external agent" feature to install the Splunk Universal Forwarders on our Accessers. They've set up the Universal Forwarders to send our "Access Logs" to a Splunk Instance, so they are able to monitor the performance of their IBM COS dsNet. If one is interested in setting up this IBM COS monitoring via Splunk Enterprise, please contact IBM COS customer success for more details.

Solution Deployment

IBM Cloud Object Storage typically sits at the storage foundation of the Splunk/IBM COS solution. IBM's COS public cloud cross regional service is designed to withstand the loss of an entire region of the United States while maintaining data availability and integrity. Together, the Splunk/IBM COS solution gives Splunk administrators, more options, more flexibility, and at lower price points.



Splunk allows independent Smartstore configuration on a per index basis, so one can have certain indices use IBM COS on premise, and other indices can use IBM COS Public Cloud.

Solution Limitations

Splunk currently does not work with IBM COS's Archive Tier nor does it work with IBM COS's retention API.

IBM Cloud Object Storage Archive Tier is our lowest cost option for data that is rarely accessed. You can store data by transitioning from any of the storage tiers (Standard, Vault, Cold Vault and Flex) to long-term offline archive or use the online Cold Vault option.

While Splunk will have no problems writing to IBM COS's Archive Tier, when Splunk's Cache Manager goes to read the objects, and the objects have been tiered, the object retrieval will not work without manual user intervention.

Due to Splunk SmartStore's requirement of being able to recall data written to S3 at any time, Splunk's SmartStore is currently not suitable for use with IBM COS archive tier.

Splunk's SmartStore allows for SSE-C, customer provided keys, which IBM COS supports. However, there is no example of how to provide our own key in Splunk's documentation. The examples in Splunk's documentation refer to AWS's KMS Key ID's… We've noted this as an open issue and will track the inquiry there.

SmartStore is also not currently compatible with IBM COS's WORM/Compliance Enabled Vaults. This is due to the fact that Splunk does require being able to write a new version of an object, which would not be allowed in WORM/CEV vaults.

SmartStore prior to 7.3.0 was not compatible with any Splunk application that utilized report acceleration or data acceleration summaries. An example of an application that previously did not work with SmartStore was Splunk Enterprise Security, as they used the data model acceleration summaries. However, with Splunk 7.3.0, SmartStore now supports report/data acceleration summaries, so as a result, SmartStore now supports Splunk Enterprise Security workflows.

Also, we wanted to point out that currently Splunk doesn't allow migrations for a SmartStore enabled index back to the traditional local bucket. There is no feature to revert index config back to non-remote. So, our recommendation when trying out the SmartStore feature is to enable it on a "test index" initially, before enabling SmartStore on all the other indices.

Another SmartStore limitation we wanted to mention is that you cannot thaw an archived bucket into a SmartStore index, even if the bucket, prior to freezing, was part of a SmartStore index. Splunk recommends thawing the bucket into the thawed directory of a non-SmartStore index. When doing so, one should be sure that the bucket ID is unique within that index. Also, as a best practice if thawing buckets is frequently part of your

normal workflow, Splunk recommends creating a set of non-SmartStore indexes that parallel the SmartStore indexes in name.

We also wanted to mention that Splunk's TSIDX reduction feature is not supported for SmartStore enabled indexes. Also, Hadoop data roll also is not supported with SmartStore indexes.

Splunk also details situations where SmartStore isn't the best fit for. Splunk points out a few situations where using local storage might be a better fit, and they are as follows:

- If you have frequent need to run rare searches, SmartStore might not be appropriate for your purposes, as rare searches can require the indexer to copy large amounts of data from remote to local storage, causing a performance impact. This is particularly the case with searches that cover long timespans. If, however, the searches are across recent data and thus the necessary buckets are already in the cache, then there is no performance impact.
- If you run frequent long lookback searches, you might need to increase your cache size or continue to rely on local storage.

As you can see, care should be taken when deploying SmartStore so that end users whom utilize Splunk Enterprise for time critical searches will be less impacted.

S3 Object Structure, Sizes, Usage

We've observed Splunk writing objects to COS as small as 6 bytes to as large as 2 GiB in size.

The largest files looked to be gzip compressed rawdata/journals. Followed by *.tsidx files. All of these larger files were stored on IBM COS via S3 multipart upload, which is very efficient. The default part size utilized by SmartStore when writing to COS is **128MB**. This is specified in the `remote.s3.multipart_upload.part_size` variable and can be changed via a configuration file.

SmartStore also utilized S3 ranged reads, of which the range by default is **128MB**. This is specified in the `remote.s3.multipart_download.part_size` variable can also be changed via a configuration file. Refer to Appendix A(Configuration Guide) for more details.

SmartStore by default utilizes **8 simultaneous connections** to IBM COS when performing Multipart Uploads to IBM COS as well as when performing ranged reads. If one wants to change from the default, that can be done via the `remote.s3.multipart_max_connections` variable.

Splunk appears to use an upper level "directory" within the S3 vault to store the SmartStore warm bucket data:

- main/db/xx/yy/bucket-UUID

Within each bucket-UUID, there appear to be a common set of files consisting of:

```
"main/db/3a/dd/12~8CEE0BE8-0AE4-439B-9870-352754BEDC1A/guidSplunk-8CEE0BE8-0AE4-439B-9870-352754BEDC1A/.rawSize",
"main/db/3a/dd/12~8CEE0BE8-0AE4-439B-9870-352754BEDC1A/guidSplunk-8CEE0BE8-0AE4-439B-9870-352754BEDC1A/.sizeManifest4.1",
"main/db/3a/dd/12~8CEE0BE8-0AE4-439B-9870-352754BEDC1A/guidSplunk-8CEE0BE8-0AE4-439B-9870-352754BEDC1A/1557025784-1556972278-5967662167173358205.tsidx",
"main/db/3a/dd/12~8CEE0BE8-0AE4-439B-9870-352754BEDC1A/guidSplunk-8CEE0BE8-0AE4-439B-9870-352754BEDC1A/1557097424-1557025784-92870534617008893.tsidx",
"main/db/3a/dd/12~8CEE0BE8-0AE4-439B-9870-352754BEDC1A/guidSplunk-8CEE0BE8-0AE4-439B-9870-352754BEDC1A/1557130302-1556929400-12040765146535885427.tsidx",
"main/db/3a/dd/12~8CEE0BE8-0AE4-439B-9870-352754BEDC1A/guidSplunk-8CEE0BE8-0AE4-439B-9870-352754BEDC1A/1557133520-1557130302-11761191590501817662.tsidx",
"main/db/3a/dd/12~8CEE0BE8-0AE4-439B-9870-352754BEDC1A/guidSplunk-8CEE0BE8-0AE4-439B-9870-352754BEDC1A/1557133917-1557133521-9083977671087333142.tsidx",
"main/db/3a/dd/12~8CEE0BE8-0AE4-439B-9870-352754BEDC1A/guidSplunk-8CEE0BE8-0AE4-439B-9870-352754BEDC1A/1557134035-1557133917-9590165178793312860.tsidx",
"main/db/3a/dd/12~8CEE0BE8-0AE4-439B-9870-352754BEDC1A/guidSplunk-8CEE0BE8-0AE4-439B-9870-352754BEDC1A/1557134035-1557134035-9589914253536476171.tsidx",
"main/db/3a/dd/12~8CEE0BE8-0AE4-439B-9870-352754BEDC1A/guidSplunk-8CEE0BE8-0AE4-439B-9870-352754BEDC1A/Hosts.data",
"main/db/3a/dd/12~8CEE0BE8-0AE4-439B-9870-352754BEDC1A/guidSplunk-8CEE0BE8-0AE4-439B-9870-352754BEDC1A/SourceTypes.data",
"main/db/3a/dd/12~8CEE0BE8-0AE4-439B-9870-352754BEDC1A/guidSplunk-8CEE0BE8-0AE4-439B-9870-352754BEDC1A/Sources.data",
"main/db/3a/dd/12~8CEE0BE8-0AE4-439B-9870-352754BEDC1A/guidSplunk-8CEE0BE8-0AE4-439B-9870-352754BEDC1A/Strings.data",
"main/db/3a/dd/12~8CEE0BE8-0AE4-439B-9870-352754BEDC1A/guidSplunk-8CEE0BE8-0AE4-439B-9870-352754BEDC1A/bloomfilter",
"main/db/3a/dd/12~8CEE0BE8-0AE4-439B-9870-352754BEDC1A/guidSplunk-8CEE0BE8-0AE4-439B-9870-352754BEDC1A/bucket_info.csv",
"main/db/3a/dd/12~8CEE0BE8-0AE4-439B-9870-352754BEDC1A/guidSplunk-8CEE0BE8-0AE4-439B-9870-352754BEDC1A/merged_lexicon.lex",
"main/db/3a/dd/12~8CEE0BE8-0AE4-439B-9870-352754BEDC1A/guidSplunk-8CEE0BE8-0AE4-439B-9870-352754BEDC1A/rawdata/journal.gz",
"main/db/3a/dd/12~8CEE0BE8-0AE4-439B-9870-352754BEDC1A/guidSplunk-8CEE0BE8-0AE4-439B-9870-352754BEDC1A/rawdata/slicemin.dat",
"main/db/3a/dd/12~8CEE0BE8-0AE4-439B-9870-352754BEDC1A/guidSplunk-8CEE0BE8-0AE4-439B-9870-352754BEDC1A/rawdata/slicesv2.dat",
"main/db/3a/dd/12~8CEE0BE8-0AE4-439B-9870-352754BEDC1A/guidSplunk-8CEE0BE8-0AE4-439B-9870-352754BEDC1A/splunk-autogen-params.dat",
"main/db/3a/dd/12~8CEE0BE8-0AE4-439B-9870-352754BEDC1A/receipt.json",
```

Splunk by default uses **AWSv4** authentication, and we've observed no issues with SmartStore using AWSv4 authentication against IBM COS. If AWSv2 auth is preferred Splunk does have a variable, `remote.s3.signature_version`, you can set to make it use AWSv2 authentication. See Appendix A(Configuration Guide) for more details.

By default, SmartStore uses "**Signed Payloads**" when uploading to IBM COS. To change to using unsigned payloads, one would modify the following variable: `remote.s3.enable_signed_payloads`.

SmartStore utilizes **AWS v1 listing requests** by default. SmartStore also allow one to change to AWSv2 listing requests via the `remote.s3.list_objects_version` configuration file variable.

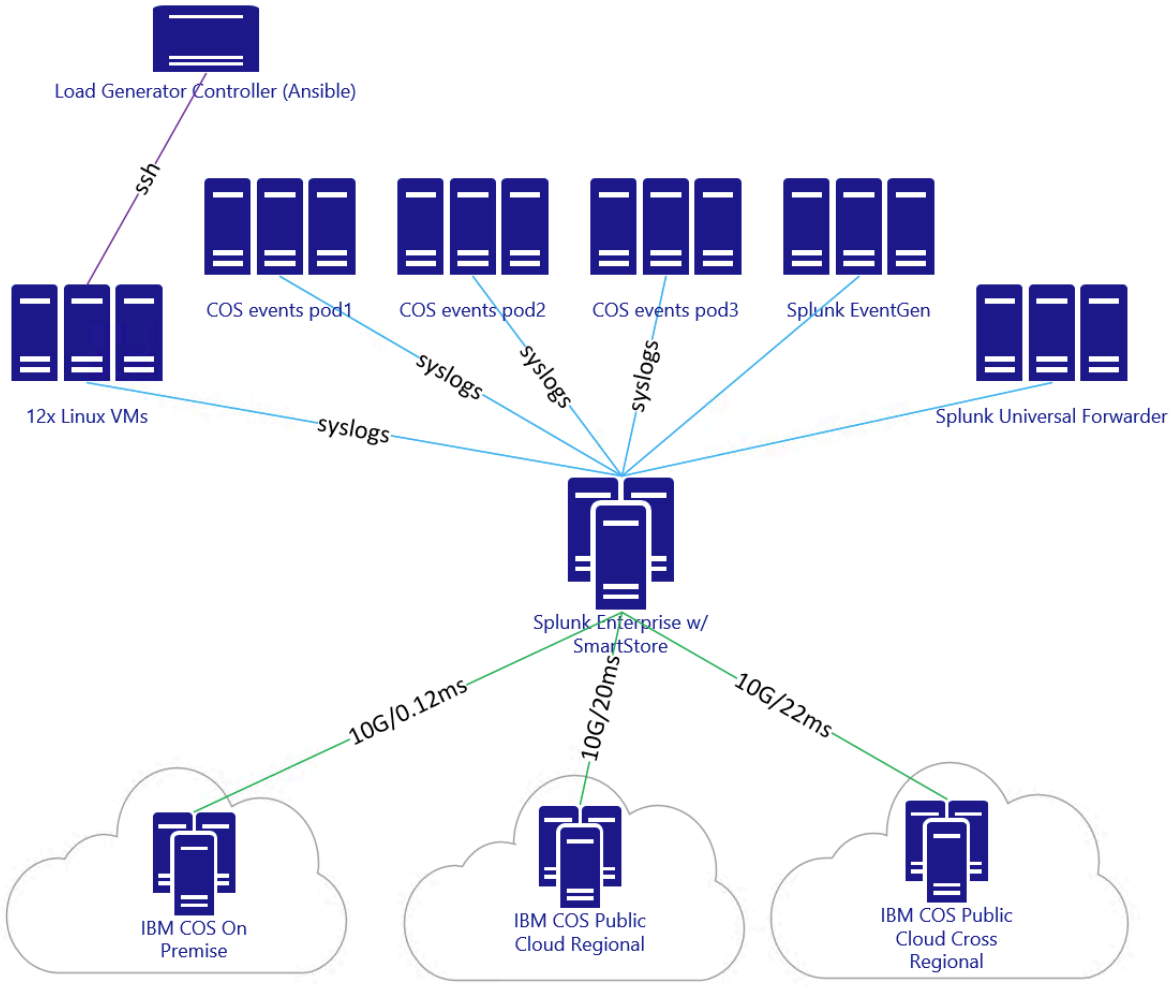
Splunk SmartStore also has numerous S3 timeout/retries configurable via the configuration file. The default is **9 retries** per part, and **128 retries per object** in total. The default S3 connect timeout is set at **5 seconds**. The default read/write timeouts are set at **60 seconds**.

When HTTPS is specified, Splunk SmartStore by default uses **TLS1.2**, which is supported by IBM COS Public Cloud and IBM COS on premise.

More details on these remote.s3.* variables are in Appendix A(Configuration Guide) of this document.

Test Architectures

The following diagram shows the test architecture used for our certification. The Splunk Enterprise Server was co-located with an IBM Cloud Object Storage on premise storage pool for the on-premise testing. Our virtual infrastructure we used for the testing was also collocated in the same datacenter. We utilized our lab's 10Gbps WAN connection on the tests to the IBM COS Public Cloud endpoints.



Splunk Hardware

The Splunk Enterprise Server being used for the performance characterization was built off of a Ubuntu Server 18.04.2 LTS VM. The resources we provisioned it with is targeted towards medium sized enterprise environments. Splunk describes this as their "High-performance specification" in their capacity planning guide. The **Splunk Enterprise** instance was comprised of a single VM, with the following hardware specifications:

- 48x Intel® Xeon® CPU E5-2697A v4 2.6GHz

- 128 GB DDR4 ECC RAM
- 1x 800GB SSD for Ubuntu OS and Splunk Enterprise Components
- 1x 10GbE network interface
- 64-bit Ubuntu Server 18.04.2 LTS

Splunk Enterprise

Splunk Enterprise 7.3.0

Load Generator Environment

The VM client environment consists of 12 VM's across 4 hypervisors, each with the following specifications:

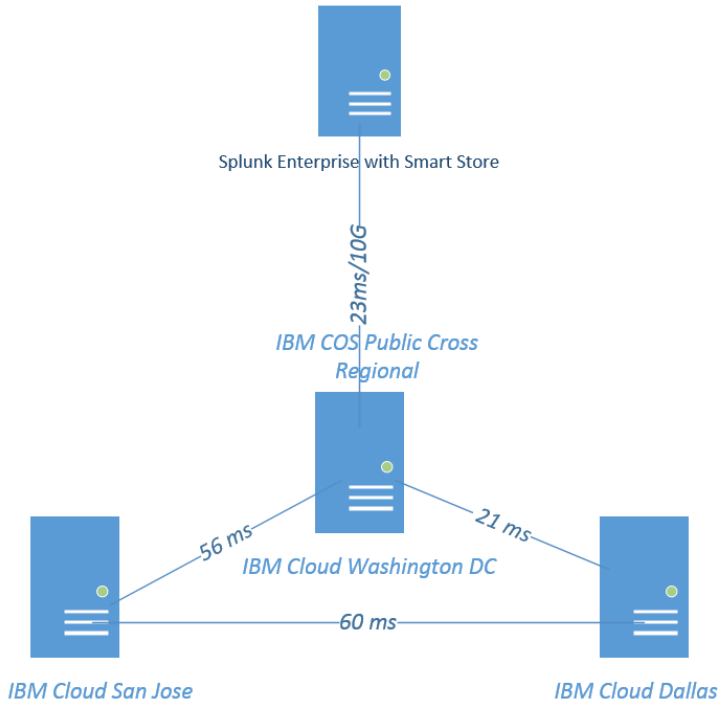
- 32 cores
- 32 GiB RAM
- 100GB storage
- 10 Gbps networking
- Ubuntu 16.04.4 LTS server OS
- genset (AIT developed data set generation tool)
- VMware vCenter 6.5
- Ansible Controller with syslog generator playbooks

IBM COS – On-Premise

- ClevOS 3.14.3.15
- 1 Accesser 4105 (40 Gbps client side links)
- 12 Slicestor 2448 – Single Site
- Vault Configuration
  - 12/7/9 IDA
  - SecureSlice: Disabled
  - Name Index: Enabled

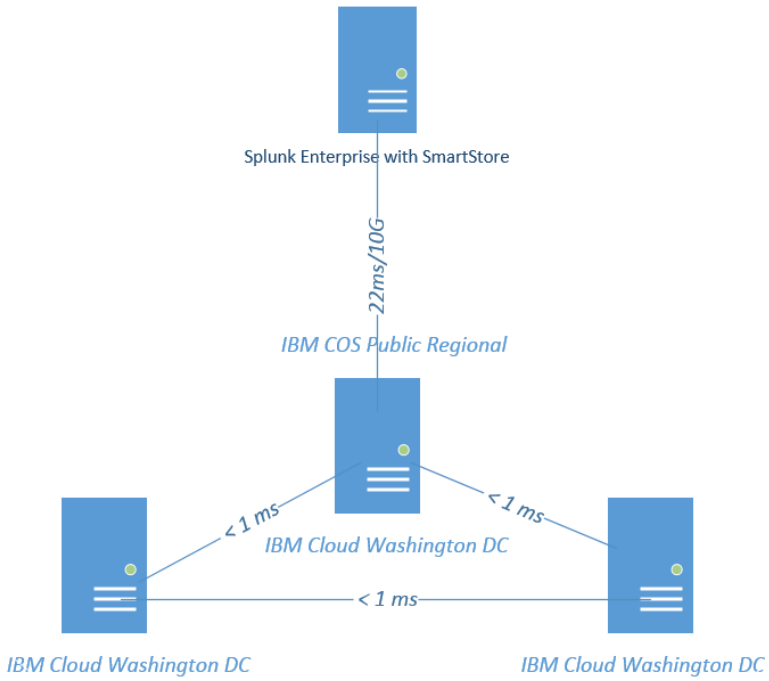
IBM COS – IBM COS Public Standard Cross Regional

- ClevOS 3.14.4.96
- 24/12/15 IDA (3 datacenters across 3 regions)
- Load balanced endpoint used was https://s3-api.dal-us-geo.objectstorage.softlayer.net
- Dallas, San Jose, Washington DC



IBM COS – IBM COS Public Standard Regional

- ClevOS 3.14.4.96
- 24/12/15 IDA (3 datacenters across 1 region)
- Load balanced endpoint used was https://s3.us-east.cloud-object-storage.appdomain.cloud
- Washington DC



Functional Characterization

With our single Splunk instance, we were unable to saturate a single A4105 Accesser during our testing, but in speaking with a Splunk/COS customer who has enabled SmartStore on all of their Splunk instances in production for the past 6 months, they have previously overrun the resources on their virtual Accessers. This customer's virtualization team has standard "templates" that they deploy, and due to their IT policies, the virtual Accessers they originally had were underpowered for the expected workflow. Remember, during searches that go back a long time(warm buckets on S3 object storage), all indexers that require data for processing the search will make ranged read GET requests to IBM COS near simultaneously.

The default maximum number of concurrent bucket downloads/uploads to/from COS by default is set to 8. This value can be configured in **server.conf** via the ***max_concurrent_downloads*** and ***max_concurrent_uploads*** settings.

With our single indexer Splunk instance, we didn't observe any Accesser resource issues, however, in a big Splunk deployment, we can see all the indexers hitting object storage near simultaneously, all downloading 8 buckets, which IBM COS can certainly handle when properly sized.

We wanted to mention that Splunk's S2 cache manager can be configured with 5 different cache eviction policies. However, Splunk documention recommends not to change from the default(LRU) without contacting Splunk Support. We saw good performance and functionality with the LRU eviction policy.

Eviction policy	Description
lru (default)	Evict the least recently used bucket.
clock	Evict the bucket with the oldest events first, unless it has been accessed recently.
lrlt	Evict the bucket with the oldest events first.
random	Randomly evict a bucket.
noevict	Don't evict.

Do not change `eviction_policy` from its default value of "lru" without consulting Splunk Support.

You may be wondering how Data Retention for SmartStore index data stored on IBM COS is configured in Splunk. With SmartStore indexes, data retention is managed cluster wide, and not on a per indexer basis. SmartStore indexes do not support the cold bucket state, so buckets roll to frozen directly from warm. Once a bucket in the index meets the criteria for freezing, the cluster will remove the bucket from the system both from IBM COS and any local caches where copies of it exist. The following parameters that can be set in index.conf, allow one to configure data retention policies for a SmartStore index:

- maxGlobalDataSizeMB
- maxGlobalRawDataSizeMB
- frozenTimePeriodInSecs

Lastly, we wanted to mention that Splunk Enterprise has no requirement for internet access. With the Splunk / IBM Cloud Object Storage solution, everything can be on-premise without internet access if client requirements require air gaps.

Performance Scaling

The Splunk / IBM Cloud Object storage solution allows Splunk administrators to seamlessly increase storage as well as storage performance via IBM COS without having to scale up compute at the same time. The traditional method of scaling up Splunk storage prior to Splunk's SmartStore feature required adding additional index nodes where one gains both storage and compute.

Scaling IBM Cloud Object Storage ***performance*** is as simple as adding more Accessers serving the storage pool. If the dsNet becomes storage pool constrained, then IBM COS allows real time addition of additional sets of Slicestors to the storage pool to increase storage pool performance without any downtime.

Likewise, scaling Splunk's search and indexing ***compute performance*** can be done independently of storage. Splunk with SmartStore/IBM COS deployment model allows the Splunk administrator to scale up easily, by adding additional indexers(when necessary) to the cluster and performing a sync of the hot buckets.

This also allows for migration of indexer cluster hardware. In fact, we learned of this specific use case from an existing customer who is using Splunk with SmartStore on IBM COS. They have seen the benefits of scaling compute when needed as well as seen the ease of performing indexer hardware migrations.

One item of note is, when scaling Splunk's compute performance by adding additional indexers, one should be careful that the IBM COS Access Pool serving the Splunk requests have sufficient capacity to handle requests from the additional indexers being added. Remember, during a search that goes back beyond what is on Splunk's Hot buckets, SmartStore's CacheManager will fetch the buckets it needs to perform the search from IBM COS. So while one does not need to add storage(by adding Slicestors) while adding indexers, one should be careful to add additional Accesser resources as necessary.

An additional method of scaling performance from an IBM COS perspective is to use Smartstore's ability to have different endpoints for each index. One can have certain indices use one particular dsNet, and other indices use another dsNet. This would split the load amongst two IBM COS deployments. And if needed, one can keep adding IBM COS deployments as one adds/migrate more indices. As you can see, this solution is truly a scalable from both Splunk and IBM COS's perspective.

Capacity Scaling

The Splunk/IBM Cloud Object Storage solution allows Splunk administrators to have an always available, infinitely scalable, and cost effective, searchable capacity tier for any index of their choice.

Splunk's SmartStore allows for true decoupling of compute and storage.

If one is using IBM Cloud Object Storage Public Cloud, IBM handles the capacity scaling for you behind the scenes.

Scaling IBM Cloud Object Storage on premise is as simple as adding an additional set of Slicestors when the existing storage pool is near the capacity limit. It is important to note that adding an additional set of Slicestors to the storage pool not only increases capacity, but also increases storage performance.

Open Issues

1) We observed that Splunk's SmartStore S3 client didn't populate the user_agent field:

You can see that Splunk shows userAgent field as null. While populating the user_agent field isn't required, we generally recommend our ISV's to include a unique identifier in this field, as we have seen that being helpful during IBM COS troubleshooting.

2) Splunk's SmartStore allows for SSE-C, customer provided keys, which IBM COS supports. However, there is no example of how to provide our own key in Splunk's documentation. The examples in Splunk's documentation refer to AWS's KMS Key ID's. Can you show us an example of how to configure SSE-C without using AWS's KMS?

3) We think there may be an error in the online 7.2.6 and 7.3.0 docs.splunk.com indexes.conf spec file. From these URLs:
https://docs.splunk.com/Documentation/Splunk/7.2.6/admin/Indexesconf
https://docs.splunk.com/Documentation/Splunk/7.3.0/admin/Indexesconf

It states:

```
remote.s3.supports_versioning = true | false
* Optional.
* Specifies whether the remote storage supports versioning.
* Versioning is a means of keeping multiple variants of an object
  in the same bucket on the remote storage.
* Defaults to: true
```

However, we've performed all of our testing against IBM COS vaults that have S3 versioning disabled. From earlier conversations with Splunk, we do not believe Splunk has yet started using the S3 versioning feature.

Appendix A: Configuration Guide

The significant segments of deploying Splunk Enterprise together with IBM Cloud Object Storage are explained through the following steps taken.

1. Prepare IBM COS as a remote store for SmartStore.
2. Configure the Splunk Indexer(or cluster peer nodes) to access the remote store.
3. Test the SmartStore deployment.

Assumptions

This guide assumes that existing IBM COS and Splunk Enterprise, systems are operating, and with certain capabilities already configured.

IBM COS

- Storage pool
- S3(CSO) Accesser pool
- COS Manager access with privileges to assign roles and administer the system including:
    - Assign user roles
- A manually created vault for Splunk. The vault can be either IBM Cloud Object Storage on premise, or IBM Cloud Object Storage Public Cloud. It should be noted that vaults will not be created automatically by Splunk SmartStore. Please be sure that the vault(s) created for Splunk utilize the settings recommended here before proceeding. **Credentials to the manually created vault must have read, write, and delete permissions.**
- Note, Splunk SmartStore allows for both:
    - all Splunk indexes using a single IBM COS vault
    - one Splunk index using a single IBM COS vault

Splunk Enterprise

- Splunk Enterprise 7.3.0 or newer
- DNS resolver and internet access for public access to Accesser(s)
- Access to the Splunk Enterprise server as an administrative user.
- Splunk Enterprise License with enough TB/day for your use case.

- Performance Tier with enough storage capacity for the use case. Splunk recommends using direct attached SSD's for this tier.
- All peer nodes on an indexer cluster must use the same SmartStore settings.

Prerequisites

IBM COS

- Tested Software Version
  - 3.14.3.15

Splunk Enterprise

- Tested Software Version
  - 7.3.0

---

1. Prepare IBM COS as a remote store for SmartStore

To add IBM Cloud Object Storage as a Splunk SmartStore destination, we will need to obtain the following:

- Access key ID
- Secret Access Key
- Hostname(Endpoint)
- Vault/Bucket

For the on-premise use case, a storage administrator with access to IBM Cloud Object Storage will be required to create the required vault with the following settings:

SecureSlice Technology:	Disabled
Versioning:	Disabled
Delete Restricted:	No
SSE-C:	Disabled
Name Index:	Enabled
Recovery Listing:	Disabled

Note, the name index is required since Splunk utilizes prefix based listings for their delete workflow. Because Splunk has an option to encrypt data uploaded to object storage, we are recommending disabling SecureSlice for on premise deployments.

For the public cloud use case, a storage administrator with access to the IBM Cloud will be required to create the vault(s). Our name index is always enabled in the IBM Public

Cloud. ***It is important to note that for the IBM Cloud vault, the storage administrator can choose the resiliency, location, and storage class at time of vault creation.*** The resiliency, location, and storage class cannot be migrated or changed post vault creation, so care should be taken during the initial vault creation to be sure that the client's needs and expectations are met. AIT has validated Splunk against all current IBM Cloud resiliency, location, and storage classes with the exception of our Archive Tier.

Before progressing to the next step, be sure to have an IBM COS vault created as well as the vault name, endpoint, access key ID, and secret access key.

2. Configure the Splunk Indexer(or cluster peer nodes) to access the remote store.

If you are configuring SmartStore on a standalone indexer, you may continue with step a. If your configuring SmartStore on a cluster of indexers, you will need to enable a single indexer as a master node prior to proceeding to step a. After enabling the master node, be sure to set the indexer cluster's replication factor and search factor to equal values.

a) Shell into the Splunk Enterprise Server and become the root user. If you are configuring SmartStore for a cluster of indexers, shell into the Splunk Enterprise master node as the root user.
b) For a standalone indexer, edit the following file:
$SPLUNK_HOME/etc/system/local/indexes.conf.
For a cluster of indexers, edit the following file on the master node:
$SPLUNK_HOME/etc/master-apps/_cluster/local/indexes.conf

There are at minimum 8 variables you will want to set under the global settings. By enabling SmartStore globally, SmartStore would then be enabled for all indices. If you would like to use a different IBM COS vault for each index, you will need to create multiple remote_store volumes as well as making sure the remotePath setting on each index is pointing to the right remote_store volume. The following example is enabling SmartStore for all indices on the Splunk instance.

```
[volume:remote_store]
storageType = remote
remote.s3.access_key = kDiJvzm0We4pwNXkZ0dN
remote.s3.secret_key = ExmLULz0NEUFhrccoIDDNSXXdyWQQc6rtFLGEUpV
remote.s3.endpoint = http://172.20.26.12
remote.s3.supports_versioning = false
path = s3://splunk0510
repFactor = auto
.
.
.
maxDataSize = auto
remotePath = volume:remote_store/$_index_name
```

For those organizations concerned about having cleartext credentials in a configuration file, Splunk will automatically overwrite both the access and secret keys upon starting with an encrypted string.

The **`maxDataSize`** should be set to auto when using SmartStore. By setting this to auto, the default will be **750MB** for each SmartStore index.

For the remote.s3.endpoint, you may also specify HTTPS instead of HTTP. If the SSL certificate is valid, there's nothing more to do. However, if your COS/S3 endpoint is using a self-signed certificate, one should import the CA used to generate the self-signed certificate as trusted on the Splunk Indexer so that SSL negotiation does not fail. One can also specifically specify the custom root CA path by using the **`remote.s3.sslRootCAPath`** variable in indexes.conf. Some organizations have internal CA's, so as long as those CA's are properly trusted from the Splunk Indexer, the Indexer should be able to read/write from IBM COS securely via HTTPS.

The **`remote.s3.supports_versioning`** needs to be set to false, as Splunk SmartStore's default is set to true. While IBM COS on premise supports versioning, IBM COS Public Cloud doesn't have versioning support.

The path variable refers to the IBM COS vault/bucket to be used. The bucket should have been created in step 1.

Also, before moving onto the next step, be sure that the following 2 variables are set correctly in indexes.conf.

```
# SmartStore-enabled indexes do not use thawedPath or coldPath,
# but you must still specify them here.
coldPath = $SPLUNK_DB/cs_index/colddb
thawedPath = $SPLUNK_DB/cs_index/thaweddb
```

The coldPath setting for each SmartStore index requires a value, even though the setting is ignored except in the case of migrated indexes. The thawedPath setting for each SmartStore index requires a value, even though the setting has no practical purpose because you cannot thaw data to a SmartStore index.

Within indexes.conf, there are many more remote.s3.* settings that can be modified and changed, as you've read about earlier. Here's the full list of them at the time of this publication:

```
###############################################################

##### S3 specific settings

###############################################################


remote.s3.header.<http-method-name>.<header-field-name> = <String>

* Optional.
```

```
* Enable server-specific features, such as reduced redundancy, encryption, and so on,
  by passing extra HTTP headers with the REST requests.
  The <http-method-name> can be any valid HTTP method. For example, GET, PUT, or ALL,
  for setting the header field for all HTTP methods.
* Example: remote.s3.header.PUT.x-amz-storage-class = REDUCED_REDUNDANCY

remote.s3.access_key = <String>
* Optional.
* Specifies the access key to use when authenticating with the remote storage
  system supporting the S3 API.
* If not specified, the indexer will look for these environment variables:
  AWS_ACCESS_KEY_ID or AWS_ACCESS_KEY (in that order).
* If the environment variables are not set and the indexer is running on EC2,
  the indexer attempts to use the access key from the IAM role.
* Default: unset

remote.s3.secret_key = <String>
* Optional.
* Specifies the secret key to use when authenticating with the remote storage
  system supporting the S3 API.
* If not specified, the indexer will look for these environment variables:
  AWS_SECRET_ACCESS_KEY or AWS_SECRET_KEY (in that order).
* If the environment variables are not set and the indexer is running on EC2,
  the indexer attempts to use the secret key from the IAM role.
* Default: unset

remote.s3.list_objects_version = v1|v2
* The AWS S3 Get Bucket (List Objects) Version to use.
* See AWS S3 documentation "GET Bucket (List Objects) Version 2" for details.
* Default: v1

remote.s3.signature_version = v2|v4
* Optional.
* The signature version to use when authenticating with the remote storage
  system supporting the S3 API.
* If not specified, it defaults to v4.
* For 'sse-kms' server-side encryption scheme, you must use signature_version=v4.

remote.s3.auth_region = <String>
* Optional
* The authentication region to use for signing requests when interacting with the remote
  storage system supporting the S3 API.
```

```
* Used with v4 signatures only.
* If unset and the endpoint (either automatically constructed or explicitly set with
  remote.s3.endpoint setting) uses an AWS URL (for example, https://s3-us-west-1.amazonaws.com)
,
  the instance attempts to extract the value from the endpoint URL (for
  example, "us-west-1").  See the description for the remote.s3.endpoint setting.
* If unset and an authentication region cannot be determined, the request will be signed
  with an empty region value.
* Defaults: unset

remote.s3.use_delimiter = true | false
* Optional.
* Specifies whether a delimiter (currently "guidSplunk") should be
  used to list the objects that are present on the remote storage.
* A delimiter groups objects that have the same delimiter value
  so that the listing process can be more efficient as it
  does not need to report similar objects.
* Defaults to: true

remote.s3.supports_versioning = true | false
* Optional.
* Specifies whether the remote storage supports versioning.
* Versioning is a means of keeping multiple variants of an object
  in the same bucket on the remote storage.
* Defaults to: true

remote.s3.endpoint = <URL>
* Optional.
* The URL of the remote storage system supporting the S3 API.
* The scheme, http or https, can be used to enable or disable SSL connectivity
  with the endpoint.
* If not specified and the indexer is running on EC2, the endpoint will be
  constructed automatically based on the EC2 region of the instance where the
  indexer is running, as follows: https://s3-<region>.amazonaws.com
* Example: https://s3-us-west-2.amazonaws.com

remote.s3.multipart_download.part_size = <unsigned int>
* Optional.
* Sets the download size of parts during a multipart download.
* This setting uses HTTP/1.1 Range Requests (RFC 7233) to improve throughput
  overall and for retransmission of failed transfers.
* A value of 0 disables downloading in multiple parts, i.e., files will always
```

```
  be downloaded as a single (large) part.
* Do not change this value unless that value has been proven to improve
  throughput.
* Minimum value: 5242880 (5 MB)
* Defaults: 134217728 (128 MB)

remote.s3.multipart_upload.part_size = <unsigned int>
* Optional.
* Sets the upload size of parts during a multipart upload.
* Minimum value: 5242880 (5 MB)
* Defaults: 134217728 (128 MB)

remote.s3.multipart_max_connections = <unsigned int>
* Specifies the maximum number of HTTP connections to have in progress for
  either multipart download or upload.
* A value of 0 means unlimited.
* Default: 8

remote.s3.enable_data_integrity_checks = <bool>
* If set to true, Splunk sets the data checksum in the metadata field of the HTTP header
  during upload operation to S3.
* The checksum is used to verify the integrity of the data on uploads.
* Default: false

remote.s3.enable_signed_payloads  = <bool>
* If set to true, Splunk signs the payload during upload operation to S3.
* Valid only for remote.s3.signature_version = v4
* Default: true

remote.s3.retry_policy = max_count
* Optional.
* Sets the retry policy to use for remote file operations.
* A retry policy specifies whether and how to retry file operations that fail
  for those failures that might be intermittent.
* Retry policies:
  + "max_count": Imposes a maximum number of times a file operation will be
    retried upon intermittent failure both for individual parts of a multipart
    download or upload and for files as a whole.
* Defaults: max_count

remote.s3.max_count.max_retries_per_part = <unsigned int>
* Optional.
```

```
* When the remote.s3.retry_policy setting is max_count, sets the maximum number
  of times a file operation will be retried upon intermittent failure.
* The count is maintained separately for each file part in a multipart download
  or upload.
* Defaults: 9

remote.s3.max_count.max_retries_in_total = <unsigned int>
* Optional.
* When the remote.s3.retry_policy setting is max_count, sets the maximum number
  of times a file operation will be retried upon intermittent failure.
* The count is maintained for each file as a whole.
* Defaults: 128

remote.s3.timeout.connect = <unsigned int>
* Optional
* Set the connection timeout, in milliseconds, to use when interacting with S3 for this volume
* Defaults: 5000

remote.s3.timeout.read = <unsigned int>
* Optional
* Set the read timeout, in milliseconds, to use when interacting with S3 for this volume
* Defaults: 60000

remote.s3.timeout.write = <unsigned int>
* Optional
* Set the write timeout, in milliseconds, to use when interacting with S3 for this volume
* Defaults: 60000

remote.s3.sslVerifyServerCert = <bool>
* Optional
* If this is set to true, Splunk verifies certificate presented by S3 server and checks
  that the common name/alternate name matches the ones specified in
  'remote.s3.sslCommonNameToCheck' and 'remote.s3.sslAltNameToCheck'.
* Defaults: false

remote.s3.sslVersions = <versions_list>
* Optional
* Comma-separated list of SSL versions to connect to 'remote.s3.endpoint'.
* The versions available are "ssl3", "tls1.0", "tls1.1", and "tls1.2".
* The special version "*" selects all supported versions.  The version "tls"
  selects all versions tls1.0 or newer.
* If a version is prefixed with "-" it is removed from the list.
```

```
* SSLv2 is always disabled; "-ssl2" is accepted in the version list but does nothing.
* When configured in FIPS mode, ssl3 is always disabled regardless
  of this configuration.
* Defaults: tls1.2


remote.s3.sslCommonNameToCheck = <commonName1>, <commonName2>, ..
* If this value is set, and 'remote.s3.sslVerifyServerCert' is set to true,
  splunkd checks the common name of the certificate presented by
  the remote server (specified in 'remote.s3.endpoint') against this list of common names.
* Defaults: unset


remote.s3.sslAltNameToCheck = <alternateName1>, <alternateName2>, ..
* If this value is set, and 'remote.s3.sslVerifyServerCert' is set to true,
  splunkd checks the alternate name(s) of the certificate presented by
  the remote server (specified in 'remote.s3.endpoint') against this list of subject alternate
names.
* Defaults: unset


remote.s3.sslRootCAPath = <path>
* Optional
* Full path to the Certificate Authrity (CA) certificate PEM format file
  containing one or more certificates concatenated together. S3 certificate
  will be validated against the CAs present in this file.
* Defaults: [sslConfig/caCertFile] in server.conf


remote.s3.cipherSuite = <cipher suite string>
* Optional
* If set, uses the specified cipher string for the SSL connection.
* If not set, uses the default cipher string.
* Must specify 'dhFile' to enable any Diffie-Hellman ciphers.
* Defaults: TLSv1+HIGH:TLSv1.2+HIGH:@STRENGTH


remote.s3.ecdhCurves = <comma separated list of ec curves>
* Optional
* ECDH curves to use for ECDH key negotiation.
* The curves should be specified in the order of preference.
* The client sends these curves as a part of Client Hello.
* We only support named curves specified by their SHORT names.
  (see struct ASN1_OBJECT in asn1.h)
* The list of valid named curves by their short/long names can be obtained
  by executing this command:
  $SPLUNK_HOME/bin/splunk cmd openssl ecparam -list_curves
```

```
* e.g. ecdhCurves = prime256v1,secp384r1,secp521r1
* Defaults: unset


remote.s3.dhFile = <path>
* Optional
* PEM format Diffie-Hellman parameter file name.
* DH group size should be no less than 2048bits.
* This file is required in order to enable any Diffie-Hellman ciphers.
* Defaults:unset.


remote.s3.encryption = sse-s3 | sse-kms | sse-c | none
* Optional
* Specifies the scheme to use for Server-side Encryption (SSE) for data-at-rest.
* sse-s3: Check http://docs.aws.amazon.com/AmazonS3/latest/dev/UsingServerSideEncryption.html
* sse-kms: Check http://docs.aws.amazon.com/AmazonS3/latest/dev/UsingKMSEncryption.html
* sse-c: Check http://docs.aws.amazon.com/AmazonS3/latest/dev/ServerSideEncryptionCustomerKeys.
html
* none: no Server-side encryption enabled. Data is stored unencrypted on the remote storage.
* Defaults: none


remote.s3.encryption.sse-c.key_type = kms
* Optional
* Determines the mechanism Splunk uses to generate the key for sending over to
  S3 for SSE-C.
* The only valid value is 'kms', indicating AWS KMS service.
* One must specify required KMS settings: e.g. remote.s3.kms.key_id
  for Splunk to start up while using SSE-C.
* Defaults: kms.


remote.s3.encryption.sse-c.key_refresh_interval = <unsigned int>
* Optional
* Specifies period in seconds at which a new key will be generated and used
  for encrypting any new data being uploaded to S3.
* Defaults: 86400


remote.s3.kms.key_id = <String>
* Required if remote.s3.encryption = sse-c | sse-kms
* Specifies the identifier for Customer Master Key (CMK) on KMS. It can be the
  unique key ID or the Amazon Resource Name (ARN) of the CMK or the alias
  name or ARN of an alias that refers to the CMK.
* Examples:
  Unique key ID: 1234abcd-12ab-34cd-56ef-1234567890ab
```

```
  CMK ARN: arn:aws:kms:us-east-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab
  Alias name: alias/ExampleAlias
  Alias ARN: arn:aws:kms:us-east-2:111122223333:alias/ExampleAlias
* Defaults: unset

remote.s3.kms.access_key = <String>
* Optional.
* Similar to 'remote.s3.access_key'.
* If not specified, KMS access uses 'remote.s3.access_key'.
* Default: unset

remote.s3.kms.secret_key = <String>
* Optional.
* Similar to 'remote.s3.secret_key'.
* If not specified, KMS access uses 'remote.s3.secret_key'.
* Default: unset

remote.s3.kms.auth_region = <String>
* Required if 'remote.s3.auth_region' is unset and Splunk can not
  automatically extract this information.
* Similar to 'remote.s3.auth_region'.
* If not specified, KMS access uses 'remote.s3.auth_region'.
* Defaults: unset

remote.s3.kms.max_concurrent_requests = <unsigned int>
* Optional.
* Limits maximum concurrent requests to KMS from this Splunk instance.
* NOTE: Can severely affect search performance if set to very low value.
* Defaults: 10

remote.s3.kms.<ssl_settings> = <...>
* Optional.
* Check the descriptions of the SSL settings for remote.s3.<ssl_settings>
  above. e.g. remote.s3.sslVerifyServerCert.
* Valid ssl_settings are sslVerifyServerCert, sslVersions, sslRootCAPath, sslAltNameToCheck,
  sslCommonNameToCheck, cipherSuite, ecdhCurves and dhFile.
* All of these are optional and fall back to same defaults as
  remote.s3.<ssl_settings>.
```

c) Optional: Configure Data Retention Policy.

Data retention policy for SmartStore indexes is configured with settings similar to those for non-SmartStore indexes. On indexer clusters, data retention for SmartStore indexes is managed cluster-wide. Once a bucket in the index meets the criteria for freezing, the cluster removes the bucket entirely from the system, both from remote storage and from any local caches where copies of it exist.

SmartStore uses the following 3 variables located in the indexes.conf file to configure data retention policies for a SmartStore index.

• **maxGlobalDataSizeMB** - This setting specifies the maximum size, in MB, for all warm buckets in a SmartStore index. When the size of an index's set of warm buckets exceeds this value, the system freezes the oldest buckets, until the size again falls below this value. This setting defaults to 0, which means that it does not limit the amount of space that the warm and cold buckets on an index can occupy.

• **maxGlobalRawDataSizeMB** - This setting specifies the maximum size, in MB, of raw data residing in all warm buckets in a SmartStore index. When the size of an index's raw data in the set of warm buckets exceeds this value, the system freezes the oldest buckets, until the size again falls below this value. This setting defaults to 0, which means that it does not limit the amount of raw data in an index.

• **frozenTimePeriodInSecs** - This setting is the same setting used with non-SmartStore indexes. It specifies that buckets freeze when they reach the configured age. The default value is 6 years(188697600 seconds).

For a standalone indexer, edit the following file:
$SPLUNK_HOME/etc/system/local/indexes.conf.
For a cluster of indexers, edit the following file on the master node:
$SPLUNK_HOME/etc/master-apps/_cluster/local/indexes.conf

And configure your data retention policy using the 3 variables above. Another optional SmartStore tunable is the ability to set cache retention periods based on data recency. The two variables you would want to tune would be:

- hotlist_recency_secs
- hotlist_bloom_filter_recency_hours

More details on these settings are available on https://docs.splunk.com

d) Now that the SmartStore configuration details are set, it is time to restart Splunk.
If deploying SmartStore to a standalone indexer, restart the indexer by executing **`$SPLUNK_HOME/bin/splunk restart`** as the root user.
If deploying SmartStore to an indexer cluster, apply the cluster-bundle so that SmartStore settings are deployed to all the peer nodes by executing **`$SPLUNK_HOME/bin/splunk apply cluster-bundle –answer-yes`** as the root user.

3. Test the SmartStore deployment.

To confirm remote storage access, one can wait until a hot bucket naturally rolls to warm(IBM COS), then perform the following command to list any files that are present in the remote store:

```
splunk cmd splunkd rfs -- ls --starts-with volume:remote_store
```

If you do not want to wait, Splunk recommends either putting a sample text file in the remote store, and then running the command above, or manually rolling a bucket with the following command:

```
splunk _internal call /data/indexes/<index_name>/roll-hot-buckets -auth <admin>:<password>
```

You can rerun the listing command periodically to verify that warm buckets are being uploaded to IBM COS.

At this point, you should be able to run normal searches against this data. In the majority of cases, you will not be transferring any data from the remote storage, because the data will already be in the local cache. Therefore, to validate data transfer from the remote store, it is recommended that you first evict all buckets from the local cache. Run the following on a peer node or a standalone indexer to force all cached buckets to be evicted.

```
curl -ku username:password
"https://localhost:8089/services/admin/cacheman/_evict" -d
path=$SPLUNK_HOME/var/lib/splunk/defaultdb/db/ -d mb=99999999999
```

After this command is run, you may need to wait a few minutes before the cache data gets evicted. You can run the following to verify that locally cached data does not exist on the indexer(peer node):

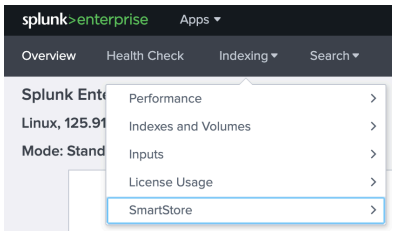
```
ls -lR $SPLUNK_HOME/var/lib/splunk/defaultdb/db | grep
cachemanager_local.json
```

If you see any cachemanager_local.json files in the output, the SmartStore cache was not properly evicted. If you see no files returned, congratulations, your SmartStore cache is now empty.

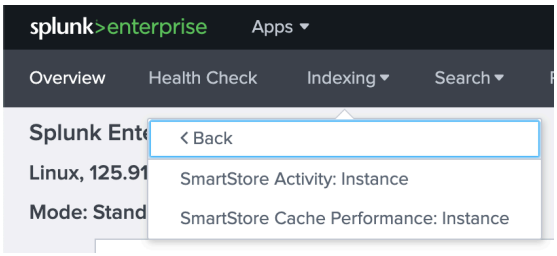
You can run a search now, and if the search requires data from the evicted bucket(s), you will see the reads from IBM COS. After the search completes, or even during the search, you may run the listing command above to see the cache getting repopulated from SmartStore's CacheManager.

Appendix-B: Tools/Troubleshooting

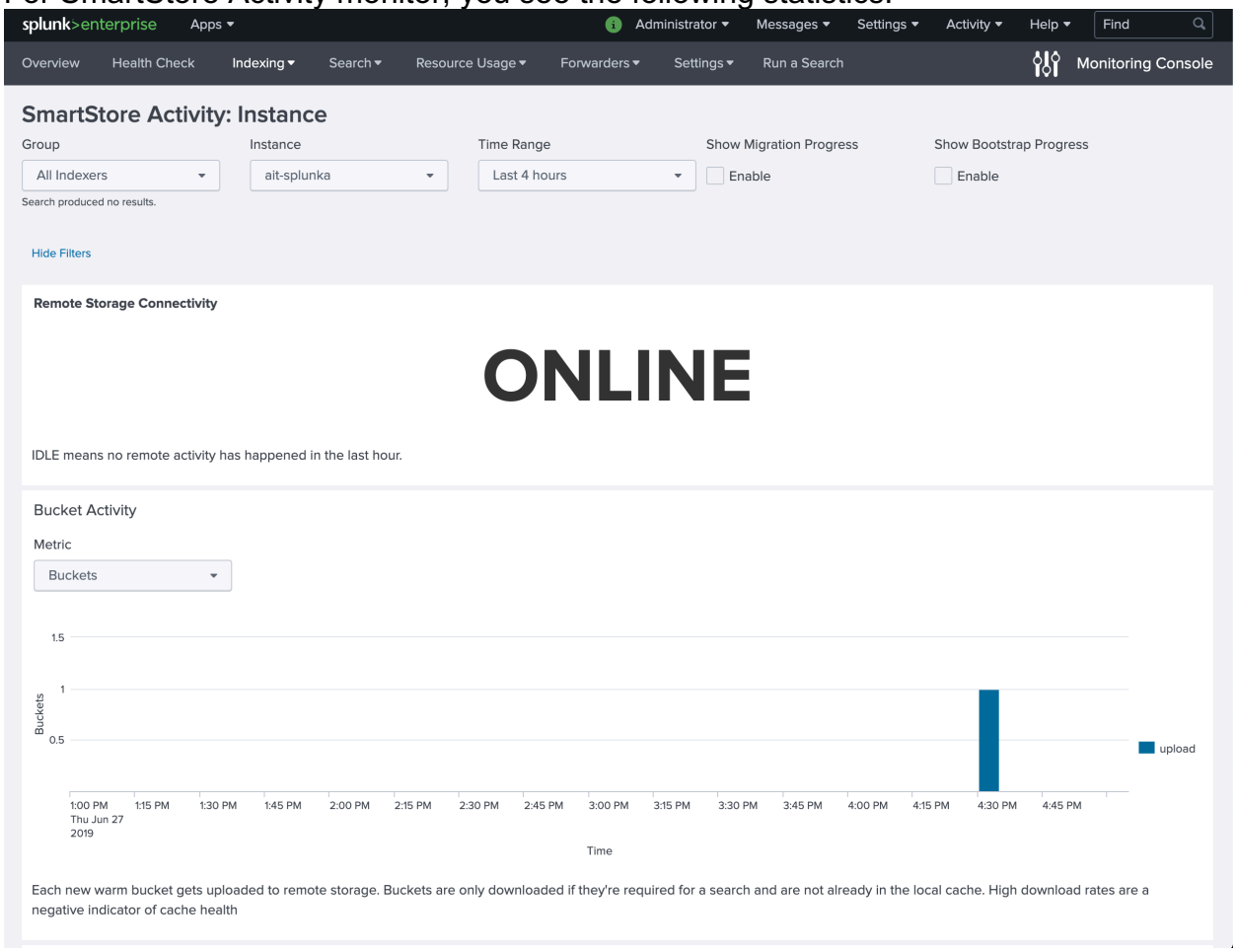
• In Splunk's indexing monitoring console, one can see that Splunk has support for monitoring SmartStore:

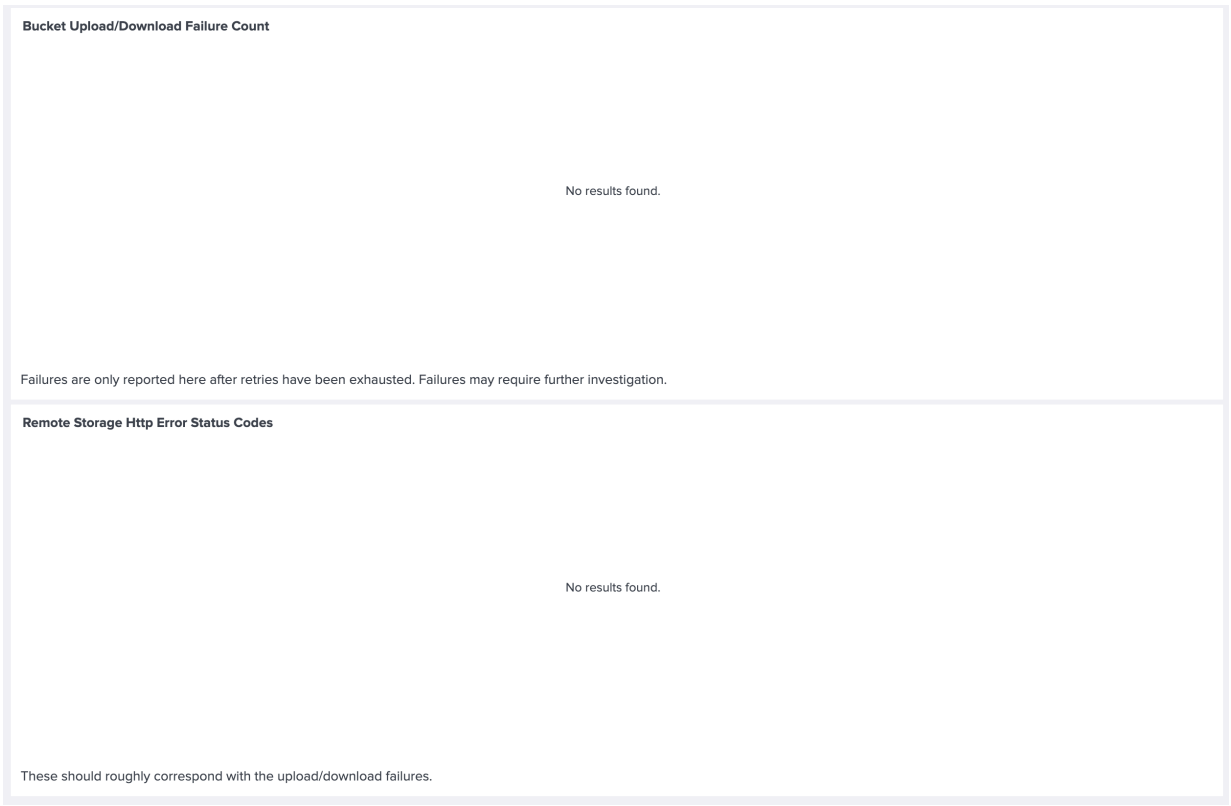


There's a SmartStore Activity monitor as well as a SmartStore Cache Performance monitor:

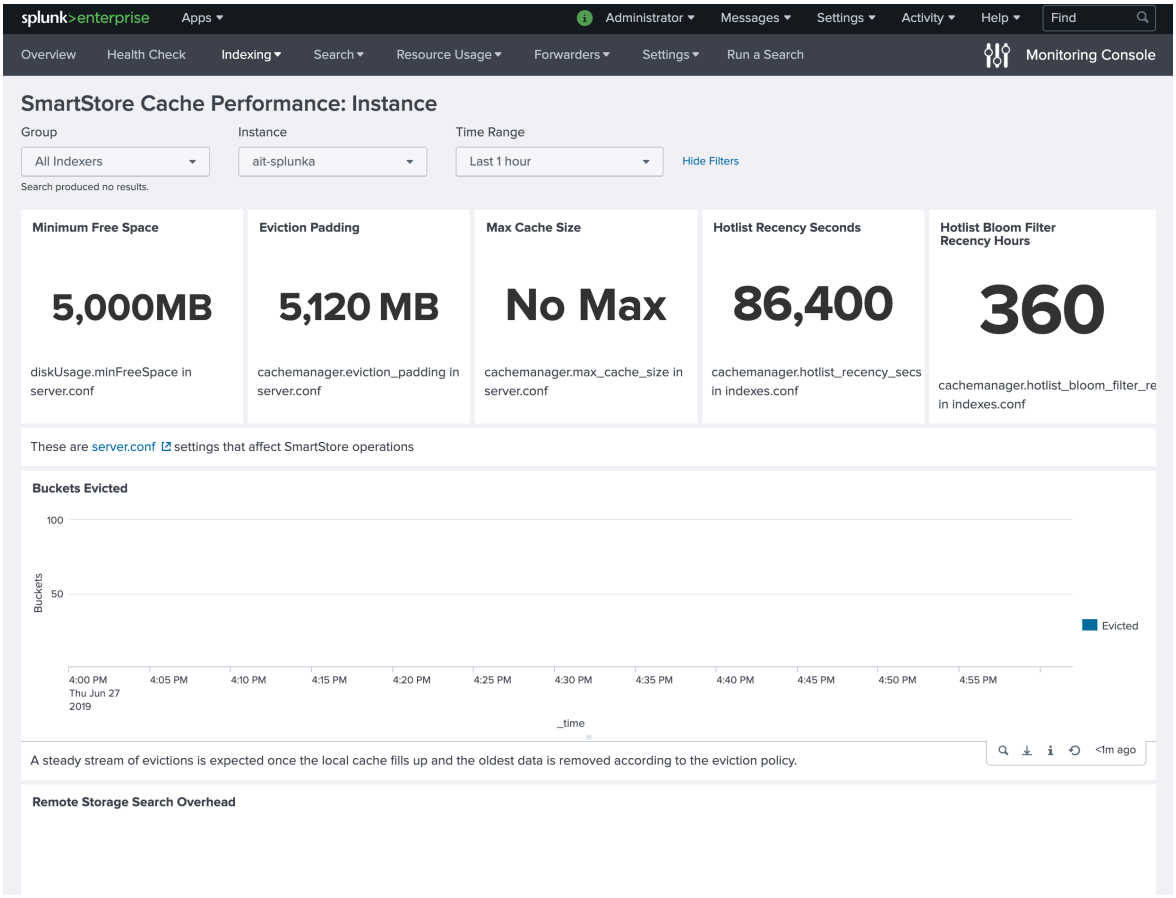


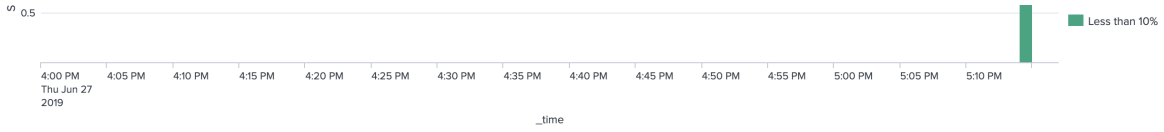
For SmartStore Activity monitor, you see the following statistics:





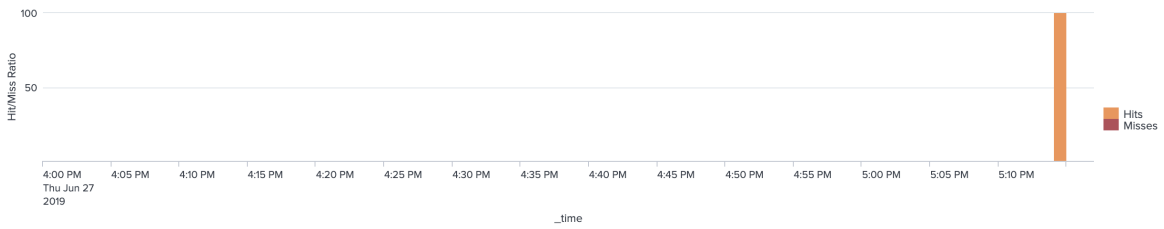
For the SmartStore Cache Performance monitor, you see the following statistics:





Portion of search time spent downloading buckets from remote storage.

Cache Hits/Misses



When a search needs a bucket and it's already in cache, that's a hit. If the bucket needs to be downloaded, that's a miss.

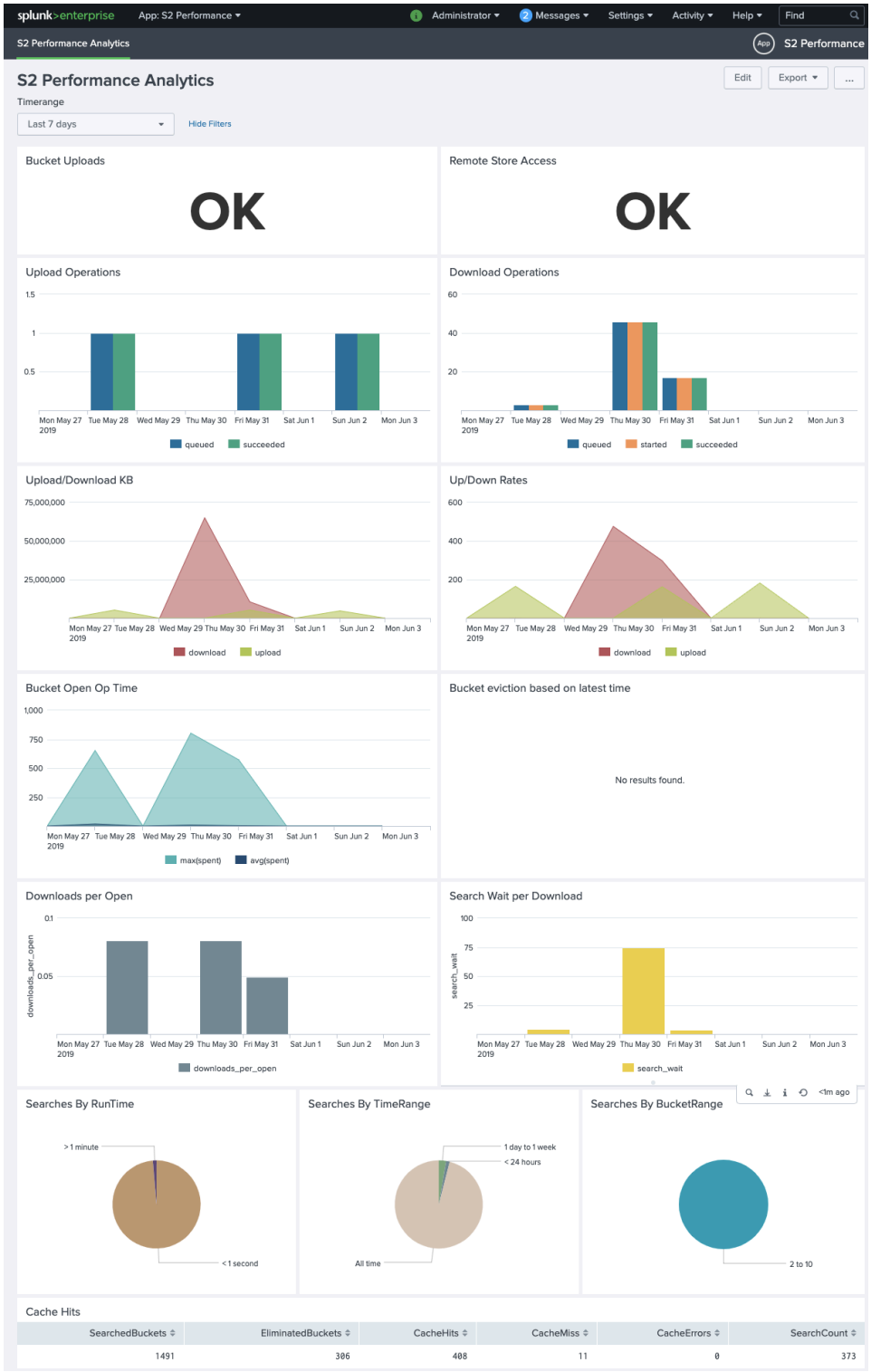
2m ago

Cache Thrash by Index

No results found.

This is the percentage of bucket downloads that are repeats. This happens when a bucket is downloaded for a search, evicted based on policy, and then needed by a search again. "Excessively Repeated Downloads" are buckets that have been downloaded more than 10 times.

• Splunk also provides a tool for monitoring SmartStore's S3 performance called "S2 Performance Analytics" and can be found under splunk_s2_performance under Splunk Applications.



As you can see, this Splunk tool shows Splunk's uploads/downloads activity to/from IBM COS, bandwidth utilization as well as how much search time indexers were waiting on downloads from IBM COS. Also provided are cache hit ratio, search runtime, searches by time range, as well as searches by bucket range.

Testing connectivity with IBM Cloud Object Storage

Splunk provides a method using “splunk cmd rfs” command to test connectivity to IBM Cloud Object Storage. The commands needed to perform the following are in bold.

• List contents of the “main” index on COS.

```
root@ait-splunk:/opt/splunk/bin# /opt/splunk/bin/splunk cmd splunkd rfs ls
index:main
#for full paths run: splunkd rfs -- ls --starts-with volume:remote_store/main/
size,name
12,main/db/08/2a/31~8CEE0BE8-0AE4-439B-9870-352754BEDC1A/guidSplunk-8CEE0BE8-0AE4-
439B-9870-352754BEDC1A/.rawSize
10,main/db/08/2a/31~8CEE0BE8-0AE4-439B-9870-352754BEDC1A/guidSplunk-8CEE0BE8-0AE4-
439B-9870-352754BEDC1A/.sizeManifest4.1
1181867984,main/db/08/2a/31~8CEE0BE8-0AE4-439B-9870-352754BEDC1A/guidSplunk-
8CEE0BE8-0AE4-439B-9870-352754BEDC1A/1558782302-1558714956-
5480741011260453768.tsidx
1388864456,main/db/08/2a/31~8CEE0BE8-0AE4-439B-9870-352754BEDC1A/guidSplunk-
8CEE0BE8-0AE4-439B-9870-352754BEDC1A/1558880686-1558782302-
3795428924668247385.tsidx
652012552,main/db/08/2a/31~8CEE0BE8-0AE4-439B-9870-352754BEDC1A/guidSplunk-
8CEE0BE8-0AE4-439B-9870-352754BEDC1A/1558927574-1558880686-
1444184996192417987.tsidx
101258096,main/db/08/2a/31~8CEE0BE8-0AE4-439B-9870-352754BEDC1A/guidSplunk-
8CEE0BE8-0AE4-439B-9870-352754BEDC1A/1558934826-1558927574-
1343097920064287844.tsidx
6989792,main/db/08/2a/31~8CEE0BE8-0AE4-439B-9870-352754BEDC1A/guidSplunk-8CEE0BE8-
0AE4-439B-9870-352754BEDC1A/1558935319-1558934827-16861688876283282892.tsidx
1799024,main/db/08/2a/31~8CEE0BE8-0AE4-439B-9870-352754BEDC1A/guidSplunk-8CEE0BE8-
0AE4-439B-9870-352754BEDC1A/1558935441-1558935319-17384913681757580489.tsidx
.
.
.
110,main/db/f1/65/5~8CEE0BE8-0AE4-439B-9870-352754BEDC1A/guidSplunk-8CEE0BE8-0AE4-
439B-9870-352754BEDC1A/Sources.data
344,main/db/f1/65/5~8CEE0BE8-0AE4-439B-9870-352754BEDC1A/guidSplunk-8CEE0BE8-0AE4-
439B-9870-352754BEDC1A/Strings.data
4237361,main/db/f1/65/5~8CEE0BE8-0AE4-439B-9870-352754BEDC1A/guidSplunk-8CEE0BE8-
0AE4-439B-9870-352754BEDC1A/bloomfilter
67,main/db/f1/65/5~8CEE0BE8-0AE4-439B-9870-352754BEDC1A/guidSplunk-8CEE0BE8-0AE4-
439B-9870-352754BEDC1A/bucket_info.csv
25289287,main/db/f1/65/5~8CEE0BE8-0AE4-439B-9870-352754BEDC1A/guidSplunk-8CEE0BE8-
0AE4-439B-9870-352754BEDC1A/merged_lexicon.lex
1836285399,main/db/f1/65/5~8CEE0BE8-0AE4-439B-9870-352754BEDC1A/guidSplunk-
8CEE0BE8-0AE4-439B-9870-352754BEDC1A/rawdata/journal.gz
1426979,main/db/f1/65/5~8CEE0BE8-0AE4-439B-9870-352754BEDC1A/guidSplunk-8CEE0BE8-
0AE4-439B-9870-352754BEDC1A/rawdata/slicemin.dat
13422941,main/db/f1/65/5~8CEE0BE8-0AE4-439B-9870-352754BEDC1A/guidSplunk-8CEE0BE8-
0AE4-439B-9870-352754BEDC1A/rawdata/slicesv2.dat
90,main/db/f1/65/5~8CEE0BE8-0AE4-439B-9870-352754BEDC1A/guidSplunk-8CEE0BE8-0AE4-
439B-9870-352754BEDC1A/splunk-autogen-params.dat
2143,main/db/f1/65/5~8CEE0BE8-0AE4-439B-9870-352754BEDC1A/receipt.json
root@ait-splunk:/opt/splunk/bin#
```

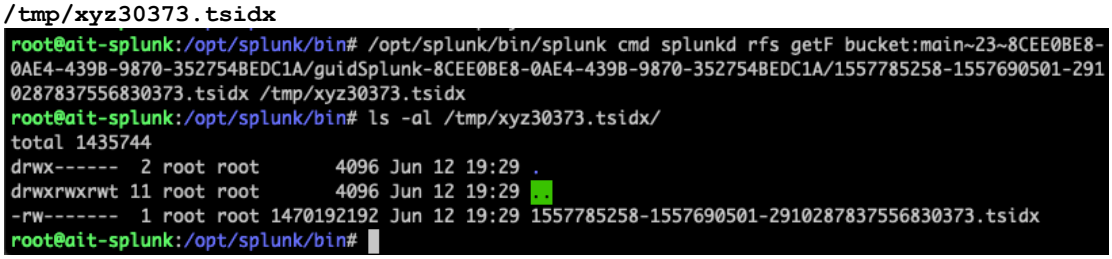
• List the contents of a given bucket on COS

```
root@ait-splunk:/opt/splunk/bin# /opt/splunk/bin/splunk cmd splunkd rfs ls
bucket:main~14~8CEE0BE8-0AE4-439B-9870-352754BEDC1A
#for full paths run: splunkd rfs -- ls --starts-with
volume:remote_store/main/db/b0/38/14~8CEE0BE8-0AE4-439B-9870-352754BEDC1A/
size,name
12,main/db/b0/38/14~8CEE0BE8-0AE4-439B-9870-352754BEDC1A/guidSplunk-8CEE0BE8-0AE4-
439B-9870-352754BEDC1A/.rawSize
10,main/db/b0/38/14~8CEE0BE8-0AE4-439B-9870-352754BEDC1A/guidSplunk-8CEE0BE8-0AE4-
439B-9870-352754BEDC1A/.sizeManifest4.1
```

```
28947,main/db/b0/38/14~8CEE0BE8-0AE4-439B-9870-352754BEDC1A/guidSplunk-8CEE0BE8-
0AE4-439B-9870-352754BEDC1A/1557328312-1557328308-7064755924731286341.tsidx
192105800,main/db/b0/38/14~8CEE0BE8-0AE4-439B-9870-352754BEDC1A/guidSplunk-
8CEE0BE8-0AE4-439B-9870-352754BEDC1A/1557493962-1557417729-
16294997783346694069.tsidx
1513719376,main/db/b0/38/14~8CEE0BE8-0AE4-439B-9870-352754BEDC1A/guidSplunk-
8CEE0BE8-0AE4-439B-9870-352754BEDC1A/1557504696-1557328983-
16762039824263159457.tsidx
1076515496,main/db/b0/38/14~8CEE0BE8-0AE4-439B-9870-352754BEDC1A/guidSplunk-
8CEE0BE8-0AE4-439B-9870-352754BEDC1A/1557506380-1557328312-
16493139279139614947.tsidx
2411,main/db/b0/38/14~8CEE0BE8-0AE4-439B-9870-352754BEDC1A/guidSplunk-8CEE0BE8-
0AE4-439B-9870-352754BEDC1A/Hosts.data
128,main/db/b0/38/14~8CEE0BE8-0AE4-439B-9870-352754BEDC1A/guidSplunk-8CEE0BE8-
0AE4-439B-9870-352754BEDC1A/SourceTypes.data
110,main/db/b0/38/14~8CEE0BE8-0AE4-439B-9870-352754BEDC1A/guidSplunk-8CEE0BE8-
0AE4-439B-9870-352754BEDC1A/Sources.data
346,main/db/b0/38/14~8CEE0BE8-0AE4-439B-9870-352754BEDC1A/guidSplunk-8CEE0BE8-
0AE4-439B-9870-352754BEDC1A/Strings.data
2394670,main/db/b0/38/14~8CEE0BE8-0AE4-439B-9870-352754BEDC1A/guidSplunk-8CEE0BE8-
0AE4-439B-9870-352754BEDC1A/bloomfilter
67,main/db/b0/38/14~8CEE0BE8-0AE4-439B-9870-352754BEDC1A/guidSplunk-8CEE0BE8-0AE4-
439B-9870-352754BEDC1A/bucket_info.csv
14944075,main/db/b0/38/14~8CEE0BE8-0AE4-439B-9870-352754BEDC1A/guidSplunk-
8CEE0BE8-0AE4-439B-9870-352754BEDC1A/merged_lexicon.lex
61447,main/db/b0/38/14~8CEE0BE8-0AE4-439B-9870-352754BEDC1A/guidSplunk-8CEE0BE8-
0AE4-439B-9870-352754BEDC1A/rawdata/49580602727
1686037685,main/db/b0/38/14~8CEE0BE8-0AE4-439B-9870-352754BEDC1A/guidSplunk-
8CEE0BE8-0AE4-439B-9870-352754BEDC1A/rawdata/journal.gz
1345013,main/db/b0/38/14~8CEE0BE8-0AE4-439B-9870-352754BEDC1A/guidSplunk-8CEE0BE8-
0AE4-439B-9870-352754BEDC1A/rawdata/slicemin.dat
12523350,main/db/b0/38/14~8CEE0BE8-0AE4-439B-9870-352754BEDC1A/guidSplunk-
8CEE0BE8-0AE4-439B-9870-352754BEDC1A/rawdata/slicesv2.dat
93,main/db/b0/38/14~8CEE0BE8-0AE4-439B-9870-352754BEDC1A/guidSplunk-8CEE0BE8-0AE4-
439B-9870-352754BEDC1A/splunk-autogen-params.dat
2326,main/db/b0/38/14~8CEE0BE8-0AE4-439B-9870-352754BEDC1A/receipt.json
```

• Test downloading a file from COS

```
root@ait-splunk:/opt/splunk/bin# /opt/splunk/bin/splunk cmd splunkd rfs getF
bucket:main~23~8CEE0BE8-0AE4-439B-9870-352754BEDC1A/guidSplunk-8CEE0BE8-0AE4-439B-
9870-352754BEDC1A/1557785258-1557690501-2910287837556830373.tsidx
/tmp/xyz30373.tsidx
```



Be sure to remove the downloaded files/clean up after your done with testing/troubleshooting.

Troubleshoot SmartStore with log files

Several log files can provide insight into the state of SmartStore operations.

splunkd.log Examine these log channels:

- S3Client. Communication with S3.
- StorageInterface. External storage activity (at a higher level than S3Client).
- CacheManager. Activity of the cache manger component.
- CacheManagerHandler. Cache manager REST endpoint activity (both server and client side).

search.log Examine these log channels:

- CacheManagerHandler. Bucket operations with cache manger REST endpoint activity.
- S2BucketCache. Search-time bucket management (open, close, and so on).
- BatchSearch, CursoredSearch, IndexScopedSearch, ISearchOperator. Search activity related to buckets.

audit.log

- Contains information on bucket operations, such as upload, download, evict, and so on.

metrics.log

- Contains metrics concerning operations on external storage.

splunkd_access.log

- Contains a trail of the search process activity against the cache manger REST endpoint.

Appendix-C: splunk/s3-tests

Splunk has a series of tests to determine whether object stores are s3 compliant. Splunk's s3-tests are located here:
https://github.com/splunk/s3-tests

We've run Splunk's S3 tests, analyzed the results, and confirmed from actual SmartStore testing, that those s3 test failures/errors do not impact current Splunk Enterprise SmartStore functionality.

Out of 212 tests run, there were fails/errors against the following 22 tests.

s3_test	s3_result
s3tests.functional.test_s3.test_put_max_tags	FAIL
s3tests.functional.test_s3.test_versioning_multi_object_delete_with_marker_create	FAIL
s3tests.functional.test_s3.test_bucket_create_naming_bad_short_two	ERROR
s3tests.functional.test_s3.test_bucket_create_naming_bad_short_one	ERROR
s3tests.functional.test_s3.test_object_delete_key_bucket_gone	FAIL
s3tests.functional.test_s3.test_object_write_to_nonexist_bucket	FAIL
s3tests.functional.test_s3.test_versioning_bucket_multipart_upload_return_version_id	FAIL
s3tests.functional.test_s3.test_versioning_bucket_atomic_upload_return_version_id	FAIL
s3tests.functional.test_s3.test_put_max_kvsize_tags	FAIL
s3tests.functional.test_headers.test_bucket_create_bad_ua_unreadable_aws2	ERROR
s3tests.functional.test_headers.test_object_create_bad_ua_unreadable_aws2	ERROR
s3tests.functional.test_headers.test_bucket_create_bad_authorization_unreadable	FAIL
s3tests.functional.test_headers.test_bucket_create_bad_contentlength_empty	FAIL
s3tests.functional.test_headers.test_bucket_create_bad_expect_unreadable	ERROR
s3tests.functional.test_headers.test_bucket_create_bad_expect_empty	ERROR
s3tests.functional.test_headers.test_bucket_create_bad_expect_mismatch	ERROR
s3tests.functional.test_headers.test_object_create_bad_authorization_unreadable	FAIL
s3tests.functional.test_headers.test_object_create_bad_expect_unreadable	ERROR
s3tests.functional.test_headers.test_object_create_bad_expect_empty	ERROR
s3tests.functional.test_headers.test_object_create_bad_expect_mismatch	ERROR
s3tests.functional.test_headers.test_object_create_bad_md5_unreadable	FAIL

We've sent the S3 test results to Splunk, and Splunk engineering came back saying the *"problems" identified in the API test results should not be a problem.*

References

IBM Cloud Object Storage information can be found:
https://ibm.com/marketplace/cloud-object-storage-system

The latest version of Splunk's documentation can always be found:
https://docs.splunk.com/Documentation/Splunk

Specifically, SmartStore documentation is available at:
https://docs.splunk.com/Documentation/Splunk/7.3.0/Indexer/AboutSmartStore

Notices

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

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

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan, Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

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

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

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

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

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Director of Licensing IBM
Corporation
North Castle Drive, MD-NC119 Armonk,
NY 10504-1785
US

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

The performance data discussed herein is presented as derived under specific operating conditions. Actual results may vary.

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

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

All IBM prices shown are IBM's suggested retail prices, are current and are subject to change without notice. Dealer prices may vary.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

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

COPYRIGHT LICENSE:

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

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Trademarks

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

Accesser®, Cleversafe®, ClevOS™, Dispersed Storage®, dsNet®, IBM Cloud Object Storage Accesser®, IBM Cloud Object Storage Dedicated™, IBM Cloud Object Storage Insight™, IBM Cloud Object Storage Manager™, IBM Cloud Object Storage Slicestor®, IBM Cloud Object Storage Standard™, IBM Cloud Object Storage System™, IBM Cloud Object Storage Vault™, SecureSlice™, and Slicestor® are trademarks or registered trademarks of Cleversafe, an IBM Company and/or International Business Machines Corp.

Other product and service names might be trademarks of IBM or other companies.