Da Xu
Senior Engineer Manager | Splunk

Bill Ern
Splunk Product Owner | Lockheed Martin
During the course of this presentation, we may make forward-looking statements regarding future events or plans of the company. We caution you that such statements reflect our current expectations and estimates based on factors currently known to us and that actual events or results may differ materially. The forward-looking statements made in this presentation are being made as of the time and date of its live presentation. If reviewed after its live presentation, it may not contain current or accurate information. We do not assume any obligation to update any forward-looking statements made herein.

In addition, any information about our roadmap outlines our general product direction and is subject to change at any time without notice. It is for informational purposes only, and shall not be incorporated into any contract or other commitment. Splunk undertakes no obligation either to develop the features or functionalities described or to include any such feature or functionality in a future release.

Splunk, Splunk>, Turn Data Into Doing, The Engine for Machine Data, Splunk Cloud, Splunk Light and SPL are trademarks and registered trademarks of Splunk Inc. in the United States and other countries. All other brand names, product names, or trademarks belong to their respective owners. © 2019 Splunk Inc. All rights reserved.
1. High Level Overview
2. Buckets
   • Hot/Warm/Cold
3. Searching
4. SmartStore at Lockheed Martin
   • Architecture
   • Learnings
High Level Overview

SmartStore brings about the separation of Storage and Compute

Before SmartStore: the filesystem was the storage tier
  • Keep buckets on disk for search
  • Keep buckets on disk for retention

After SmartStore: the filesystem is for compute!
  • Keep buckets on disk for search
  • Keep buckets on disk for retention
**Architecture**

**Components**

**Classic Architecture**

- **Log**
- **splunk>**
- **Hot/Warm Storage**
- **Cold Storage**

**S2 Architecture**

- **Log**
- **splunk>**
- **Hot/Cache Storage**
- **Remote Storage**
- **S3**
- **Hot/Cache Storage**
Key Advantages

Our Storage Tier is no longer tied to hardware
• Separation of storage and compute
• Indexer failures is no longer tied to storage failure

Local Storage is now simply a Search-Cache
• No longer need to size local storage to hold long-term retention
  – Local storage used to store - 90 days, 6 months, 3 years, etc
• Just need enough local storage for search
  – Most search is just for 1 day or 7 days
Key Advantages

Classic Architecture
Key Advantages

Classic Architecture

Hot/Warm Storage

Cold Storage

Indexer

Cold Storage

Hot/Warm Storage

Indexer
Key Advantages
Classic Architecture

- Hot/Warm Storage
- Cold Storage
- Indexer

- Hot/Warm Storage
- Cold Storage
- Indexer
Key Advantages
Smart Store Architecture
Key Advantages
Smart Store Architecture

[Diagram showing the architecture with components labeled: Indexer, Hot/Search Cache, Remote Storage, and S3.]
Key Advantages
Smart Store Architecture
Deep Dive – Buckets
Buckets Overview - Hot

Hot Buckets are the same as non SmartStore

- When using Indexer Clustering, hot buckets are replicated just like normal

There is no information about hot buckets on remote storage

Search works the same as non SmartStore
Buckets Overview - Hot

When the bucket transitions to warm, that’s when it will upload to remote storage

• In the clustering case, only the source will initially upload
• The targets will start a timer, and upload later ONLY if the source failed to upload within the timer expiration

After uploading the bucket to remote storage, we also keep the bucket on local disk.

• It’s a recent hot bucket – and will likely be searched!
• In clustering, we keep the source, but the targets will be evicted to free up space
Buckets Overview – Warm/Cold

Warm (and cold) buckets may or may not be fully existing!

- The folder is there, and they are in memory, but the actual contents may be missing!
- If the content is missing, these are considered evicted buckets
- When a search comes in against an evicted bucket, Splunk will download the bucket to serve the search – more on this process later
Buckets Overview – Warm/Cold

```bash
splunk@idx-i-0f8facba5ef9ac497:$ ls -ltr var/lib/splunk/_internal/db/ | head -n 400 | tail
drwx------ 2 splunk splunk 4096 Sep 19 15:27 db_1567408339_1567408235_8287_508A66A1-4E37-497D-A038-E8D37664D72
drwx------ 2 splunk splunk 4096 Sep 19 15:28 db_1566684199_1566868870_5851_00D246A1-32FE-4C7F-8A96-548D40C50073
drwx------ 2 splunk splunk 4096 Sep 19 15:28 db_1566522183_1566522177_5858_83C9A849-D062-4370-89A9-52FF1539079D
drwx------ 2 splunk splunk 4096 Sep 19 15:28 db_1566912744_1566912729_5860_00D246A1-32FE-4C7F-8A96-548D40C50073
drwx------ 2 splunk splunk 4096 Sep 19 15:28 db_1567096738_1567058261_5876_00D246A1-32FE-4C7F-8A96-548D40C50073
drwx------ 2 splunk splunk 4096 Sep 19 15:29 db_1567122755_1567120695_5881_00D246A1-32FE-4C7F-8A96-548D40C50073
drwx------ 2 splunk splunk 4096 Sep 19 15:29 db_1567124087_1567121687_5884_00D246A1-32FE-4C7F-8A96-548D40C50073
drwx------ 2 splunk splunk 4096 Sep 19 15:29 db_1567130519_1567126875_5892_00D246A1-32FE-4C7F-8A96-548D40C50073
drwx------ 2 splunk splunk 4096 Sep 19 15:29 db_1567147947_1567131875_5897_00D246A1-32FE-4C7F-8A96-548D40C50073
drwx------ 2 splunk splunk 4096 Sep 19 15:29 db_1567147265_1567146441_5898_00D246A1-32FE-4C7F-8A96-548D40C50073
splunk@idx-i-0f8facba5ef9ac497:$ ls -ltr var/lib/splunk/_internal/db/db_1567147265_1567146441_5898_00D246A1-32FE-4C7F-8A96-548D40C50073
total 464
drwx------ 2 splunk splunk 4096 Sep 19 15:29 .
drwx------ 3511 splunk splunk 466944 Oct 21 06:43 ..
splunk@idx-i-0f8facba5ef9ac497:$
```
Getting Data In

Clustered Deployments

1. Data arrives and is written to a Hot bucket
2. Hot bucket streams to cluster peer(s) according to RF
3. Replication completes and the buckets roll to warm
4. Buckets are registered with their cache managers
5. Cache manager on source peer uploads the bucket to the remote store
6. Source peer notifies replication peers that the bucket was uploaded successfully
7. Cached copies remains on the peers until evicted by the local cache manager
Searching

Hot Buckets are the same as non SmartStore

Warm buckets behavior can be different depending on the bucket state:

<table>
<thead>
<tr>
<th>Bucket State</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>hot</td>
<td>Same as non SmartStore</td>
</tr>
<tr>
<td>warm – all files local</td>
<td>Same as non SmartStore</td>
</tr>
<tr>
<td>warm – some files local</td>
<td>Might need to download more:</td>
</tr>
<tr>
<td></td>
<td>If local files are enough for search, then we don’t!</td>
</tr>
<tr>
<td></td>
<td>Ex:</td>
</tr>
<tr>
<td></td>
<td>Bloomfilter filters out this bucket ➔ nothing more is needed</td>
</tr>
<tr>
<td></td>
<td>Tstats search and tsidx is local ➔ nothing more is needed</td>
</tr>
<tr>
<td>warm – no files local</td>
<td>Start downloading files for search</td>
</tr>
</tbody>
</table>
Searching with S2

1. Search request is received
2. Indexer generates a list of relevant buckets to be searched
3. Search process is spawned
4. Spawned process reads the bucket list
5. Hot buckets are searched in the same manner as “classic” search
Searching with S2

Cached Buckets

1. Search process “opens” the bucket with the Cache manager
2. Cache manager tells the search process that the bucket is local and available for search
3. Search process searches the bucket
4. Search process "closes" the bucket with the cache manager
Searching with S2

Remote Buckets

1. Search process “opens” the bucket with the Cache manager, but it isn’t in cache
2. Search process waits
3. Cache manager fetches the bucket from the remote store
4. Cache manager tells the search process that the bucket is local and available for search
5. Search process searches the bucket
6. Search process “closes” the bucket with the cache manager
7. Bucket remains in cache until evicted by the cache manager
Cache Manager

Localizing Data

Cache manager offers the ability to fetch specific bucket files

Ex: bloomfilter, TSIDX, metadata, journal

Some search commands only need specific files from the bucket

• Don’t need the raw data
  – Ex: metadata, tstats
• Don’t need any bucket content
  – Ex: eventcount, dbinspect

Lookahead

• Cache manager will attempt to pre-fetch buckets needed for a search
  – Heuristic will adjust itself based upon the thruput from the remote store
Example - searching

```
splunk@idx-i-0f8faca5ef9ac497:~$ ls -latr var/lib/splunk/internaldb/db/ | head -n 400 | tail
drw--------  2 splunk splunk  4096 Sep 19 15:27 db_1567409839_1567400823_5827_508A66A1-4E47-497D-A038-E8D37664D72
drw--------  2 splunk splunk  4096 Sep 19 15:28 db_1566884199_1566868870_5851_00D246A1-32FE-4C7F-8A96-548D40C50073
drw--------  2 splunk splunk  4096 Sep 19 15:28 db_1566522183_1566522177_5858_83C9ABA9-D062-4370-89A9-52FF1539D79D
drw--------  2 splunk splunk  4096 Sep 19 15:28 db_1566912744_1566902296_5860_00D246A1-32FE-4C7F-8A96-548D40C50073
drw-x-------  2 splunk splunk  4096 Sep 19 15:28 db_1567096738_1567058261_5876_00D246A1-32FE-4C7F-8A96-548D40C50073
drw-x-------  2 splunk splunk  4096 Sep 19 15:29 db_1567122755_1567120695_5881_00D246A1-32FE-4C7F-8A96-548D40C50073
drw-x-------  2 splunk splunk  4096 Sep 19 15:29 db_1567124087_1567121687_5884_00D246A1-32FE-4C7F-8A96-548D40C50073
drw-x-------  2 splunk splunk  4096 Sep 19 15:29 db_1567130519_1567126875_5892_00D246A1-32FE-4C7F-8A96-548D40C50073
drw-x-------  2 splunk splunk  4096 Sep 19 15:29 db_1567147947_1567131875_5897_00D246A1-32FE-4C7F-8A96-548D40C50073
drw-x-------  2 splunk splunk  4096 Sep 19 15:29 db_1567147265_1567146441_5898_00D246A1-32FE-4C7F-8A96-548D40C50073
splunk@idx-i-0f8faca5ef9ac497:~$ ls -latr var/lib/splunk/internaldb/db/db_1567147265_1567146441_5898_00D246A1-32FE-4C7F-8A96-548D40C50073
```

464 total

```
drw-x-------  2 splunk splunk  4096 Sep 19 15:29 .
drw--------  3511 splunk splunk  466944 Oct 21 06:43 ..
splunk@idx-i-0f8faca5ef9ac497:~$
```
Example - searching

```
$ ls -ltr var/lib/splunk/_internal/db/
```

```
-splunk@idx-i-0f8facba5ef9ac497:-$ ls -ltr var/lib/splunk/_internal/db/ | head -n 400 | tail
```

```
-splunk@idx-i-0f8facba5ef9ac497:-$ bin/splunk search "| tstats count where index=._internal earliest=1567146441 latest=1567147265"
```
Example - searching

```
splunk@idx-i-0f8facba5ef9ac497:$ bin/splunk search "| tstats count where index=_internal earliest=1567146441 latest=1567147265"

INFO: Your timerange was substituted based on your 'where' clause
count
-------
715445
```
Example - searching

```bash
splunk@idx-i-0f8facba5ef9ac497:~$ bin/splunk search "| tstats count where index=_internal earliest=1567146441 latest=1567147265"

INFO: Your timerange was substituted based on your 'where' clause

count
-------
715445
```
```bash
splunk@idx-i-0f8facba5ef9ac497:~$ ls -ltr var/lib/splunk/_internaldb/db/db_1567147265_1567146441_5898_00d246a1-32fe-4c7f-8a96-548d40c50073
```
```bash
total 1072
-rw------- 1 splunk splunk 88 Oct 21 06:44 splunk-autogen-params.dat
-rw------- 1 splunk splunk 8 Oct 21 06:44 .rawSize
-rw------- 1 splunk splunk 75 Oct 21 06:44 bucket_info.csv
-rw------- 1 splunk splunk 6 Oct 21 06:44 .sizeManifest4.1
-rw------- 1 splunk splunk 600072 Oct 21 06:44 1567147265-1567146441-16742391476385632200.tsidx
-rw------- 1 splunk splunk 49 Oct 21 06:44 cachemanager_local.json
drwxr-xr-x 2 splunk splunk 4096 Oct 21 06:44 .
drwx------ 3511 splunk splunk 466944 Oct 21 06:45 ..
```
Example - searching

- **cachemanager_local.json**
  - contains the local set of files for a particular bucket
Example - searching

```
splunk@idx-i-0f8facba5ef9ac497:~$ bin/splunk search "index=_internal earliest=1567146441 latest=1567147265" | stats count
```

Example - searching

```
splunkidx-i-0f8facba5ef9ac497:-$ bin/splunk search "index=_internal earliest=1567146441 latest=1567147265 stats count"
```

INFO: Your timerange was substituted based on your search string

```
count
---
715445
```

```
splunkidx-i-0f8facba5ef9ac497:-$ ls -altr var/lib/splunk/_internaldb/db_db_1567147265_1567146441_5898_000246A1-32FE-4C7F-8A96-548D40C50073/*
-rw------- 1 splunk splunk 600072 Oct 21 06:44 var/lib/splunk/_internaldb/db_db_1567147265_1567146441_5898_000246A1-32FE-4C7F-8A96-548D40C50073/156714665-156714676
```

```
splunkidx-i-0f8facba5ef9ac497:-$ ls -altr var/lib/splunk/_internaldb/db_db_1567147265_1567146441_5898_000246A1-32FE-4C7F-8A96-548D40C50073/splunk-autogen-part
-rw------- 1 splunk splunk 88 Oct 21 06:45 var/lib/splunk/_internaldb/db_db_1567147265_1567146441_5898_000246A1-32FE-4C7F-8A96-548D40C50073/splunk-autogen-part
```

```
splunkidx-i-0f8facba5ef9ac497:-$ ls -altr var/lib/splunk/_internaldb/db_db_1567147265_1567146441_5898_000246A1-32FE-4C7F-8A96-548D40C50073/bucket_info.csv
```

```
splunkidx-i-0f8facba5ef9ac497:-$ ls -altr var/lib/splunk/_internaldb/db_db_1567147265_1567146441_5898_000246A1-32FE-4C7F-8A96-548D40C50073/SourceTypes.data
```

```
splunkidx-i-0f8facba5ef9ac497:-$ ls -altr var/lib/splunk/_internaldb/db_db_1567147265_1567146441_5898_000246A1-32FE-4C7F-8A96-548D40C50073/Sources.data
```

```
splunkidx-i-0f8facba5ef9ac497:-$ ls -altr var/lib/splunk/_internaldb/db_db_1567147265_1567146441_5898_000246A1-32FE-4C7F-8A96-548D40C50073/Strings.data
```

```
splunkidx-i-0f8facba5ef9ac497:-$ ls -altr var/lib/splunk/_internaldb/db_db_1567147265_1567146441_5898_000246A1-32FE-4C7F-8A96-548D40C50073/Hosts.data
```

```
splunkidx-i-0f8facba5ef9ac497:-$ ls -altr var/lib/splunk/_internaldb/db_db_1567147265_1567146441_5898_000246A1-32FE-4C7F-8A96-548D40C50073/Bloomfilter
```

```
splunkidx-i-0f8facba5ef9ac497:-$ ls -altr var/lib/splunk/_internaldb/db_db_1567147265_1567146441_5898_000246A1-32FE-4C7F-8A96-548D40C50073/cachemanager_local
```

```
var/lib/splunk/_internaldb/db_db_1567147265_1567146441_5898_000246A1-32FE-4C7F-8A96-548D40C50073/rawdata:
```

```
total 288
```

```
rw------- 1 splunk splunk 539 Oct 21 06:45 slicesv2.dat
```

```
rw------- 1 splunk splunk 84 Oct 21 06:45 slicein.dat
```

```
rw------- 1 splunk splunk 275361 Oct 21 06:45 journal.gz
```

```
drw-------- 2 splunk splunk 4096 Oct 21 06:45 .
```

```
drw--x--- 3 splunk splunk 4096 Oct 21 06:45 ..
```

```
splunkidx-i-0f8facba5ef9ac497:-$ splunk> 
```
```
Deep Dive – Eviction
Cache Manager – Eviction Policies

Cached Data

When do we evict?

- Hot buckets are always local
- Warm buckets are not immediately evicted from the cache
  - We look at available storage first, then…
  - Cache manager will attempt to keep buckets that contain events with timestamps from the last 24 hours [hotlist_recency_secs]
- TSIDX and Journals are evicted quicker than other bucket files
  - Cache manager will attempt to keep smaller bucket files for 15 days [hotlist_bloom_filter_recency_hours]

<table>
<thead>
<tr>
<th>Policy Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>clock</td>
<td>Prefer to evict bucket with the oldest events first, unless it has been accessed recently</td>
</tr>
<tr>
<td>lru (default)</td>
<td>Evict the least recently used bucket</td>
</tr>
<tr>
<td>random</td>
<td>Randomly evict a bucket</td>
</tr>
<tr>
<td>lru_lt</td>
<td>Evict the bucket with the oldest events first</td>
</tr>
<tr>
<td>noevict</td>
<td>Don’t evict – This can be used to provide data resiliency instead of indexer clustering</td>
</tr>
</tbody>
</table>
Cache Manager – Eviction Policies

Cached Data

When do we evict?

• As we approach using up all the available cache space.
  – max cache size
  – max volume size
• Clustered target buckets on hot → warm bucket transitions
Eviction
Cached Data

Very similar to CPU Memory Caching!
Eviction
Cached Data

Very similar to CPU Memory Caching!

Cache and Main Memory

Word Transfer
Block Transfer

CPU
Cache
Main Memory

Hot/Cache Storage
Remote Storage

Splunk
Monitoring Console – Cache Performance

SmartStore Cache Performance: Deployment

- **Minimum Disk Space Conf Setting**: 10%
  - `diskUsage.minFreeSpace` in `server.conf`

- **Eviction Padding**: 5,120 MB
  - `cachemanager.eviction_padding` in `server.conf`

- **Max Cache Size**: No Max
  - `cachemanager.max_cache_size`

- **Hotlist Recency Seconds**: 86,500
  - `cachemanager.hotlist_rencency_sec`

- **Hotlist Bloom Filter Recency Hours**: 360
  - `cachemanager.hotlist_bloom_filter_rencency_hours`

These are `server.conf` settings that affect SmartStore operations.

**Buckets Evicted**

- A steady stream of evictions is expected once the local cache fills up and the oldest data is removed according to the eviction policy.
Logs - eviction

splunk@idx-i-0f8facba5ef9ac497:~/var/log/splunk$ grep -e evicted metrics.log*

metrics.log:10-21-2019 05:36:47.392 +0000 INFO Metrics - group=spacemgr, evict_requested_kb=18908, evicted_kb=778592, elapsed_ms=55, tested=12958, evicted=5, partial_evict=770, insignificant_size=12179, cleaned=7, reserved_bytes=53687091200, free_bytes=428460789760, eviction_runs=1
Monitoring Console - eviction

Remote Storage Connectivity

ONLINE

IDLE means no remote activity has happened in the last hour.

Bucket Activity

Each new warm bucket gets uploaded to remote storage. Buckets are only downloaded if they're required for a search and are not already in the local cache. High download rates are a negative indicator of cache health.
Monitoring Console - cache hits / misses

Remote Storage Search Overhead

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.
Deep Dive – Migration
Migrations

Migration steps:
• Smart store should be enabled cluster-wide simultaneously
  1. Shut down the cluster
  2. Start the Cluster Master
  3. Enable Smart Store
  4. Push a bundle – this will finish instantly when no indexers are connected
  5. Bring up the indexers
  6. Migration will start and run in the background
Migrations – in the background

How S2 migration upload works

- Peers will upload all their searchable copies to remote storage
- Only one copy will “win” and remain on remote storage
- Restarting an indexer before S2 migration completes causes the indexer to resume migration
Migrations – in the background

Any limitations? Caveats? Scalability concerns?
  • Initialization is resource heavy
  • Uploading 10K’s of buckets may take several hours
  • Search is impacted during this time

Any capacity considerations eg. disk usage, cpu?
  • Greatest impact is on the networking
  • About 30% performance degradation on searching during S2 migration
Migrations – in the background

Tracking start (one entry per index):
• 03-29-2017 21:31:15.178 -0700 INFO DatabaseDirectoryManager - Remote storage migration needed for idx=foo for a bucket count=9

Tracking end of migration (all indexes):
• 03-29-2017 21:31:24.417 -0700 INFO CacheManager - Remote storage migration of buckets and summaries completed (duration_sec=9 upload_jobs=9)
Migrations – in the background

Endpoint that tracks status (in progress):

```
./master/bin/splunk search "|rest /services/admin/cacheman/_metrics |fields splunk_server migration.*"
```

```
splunk_server migration.current_job  migration.start_epoch  migration.status  migration.total_jobs
-----------------  -----------------  -----------------  -----------------
fool13-peer      3     1485306460  running                        43
fool15-peer      6     1485306476  running                        46
fool14-peer      5     1485306468  running                        44
```
Other Items

• Must be using a single object store.
• If multi-region in AWS, the endpoint must be against a specific region’s S3 bucket
• No support yet for non gzip compression, tsidx minification
Background

Interfaces Across 4 Main Categories

- Enterprise Splunk offering since 2016
- Splunk infrastructure is built in AWS
- Interfaces: ~27,394
  - Agent Based
  - Agentless
  - API-Based
  - IoT Sensors (IAI Premium Module)
- Users
  - 1,266 Users
  - 166 Power Users
- Daily Data Ingest
  - ~10.4 TB of data
- Version as of the Migration was 7.2.3
Before - High Level Indexer Tier Architecture

Before the Migration to SmartStore

Indexer Cluster Master

EC2 Compute C5-18.xlarge
72 vCPU | 144 GB RAM

GP2 - Volume

GP2 - RAID 0

ST1 - RAID 0
Overall Goals

1. Change the Indexing Architecture to SmartStore

2. Move to I3 Reserved Instances (RI) 3 Year convertible

3. Use “Native AWS Services”
Moving to S2 Architecture

Clustered Indexer

Search Head

indexer

replicate hot buckets for fault-tolerance

splunkd

read & write buckets locally

Local Storage (cache) e.g. SSD

manage cache space

Cache Management Layer

move buckets from external storage to local storage

copy warm buckets to external storage

External Storage e.g. S3

event data
After - High Level Indexer Tier Architecture

After the Migration to Smart Store

Indexer Cluster Master

EC2 Compute I3.8Xlarge
32 vCPU | 244 GB RAM
4 X 1900 NVMe SSD
Smart Store Cache

GP2 - Volume
Monitoring the Migration Process

1. Enterprise Splunk Version 7.3 Supports SmartStore Metrics Views
2. Continuously monitor indexer tier

```
$ splunk search "|rest /services/admin/cacheman/_metrics |fields splunk_server migration.*" -auth admin:passwd
```

<table>
<thead>
<tr>
<th>splunk_server</th>
<th>migration.current_job</th>
<th>migration.start_epoch</th>
<th>migration.status</th>
<th>migration.total</th>
</tr>
</thead>
<tbody>
<tr>
<td>cluster1-master</td>
<td>-</td>
<td>-</td>
<td>not_started</td>
<td>-</td>
</tr>
<tr>
<td>peer1.ajax.com</td>
<td>8</td>
<td>1484942186</td>
<td>running</td>
<td>35</td>
</tr>
<tr>
<td>peer2.ajax.com</td>
<td>7</td>
<td>1484942190</td>
<td>running</td>
<td>37</td>
</tr>
<tr>
<td>peer2.ajax.com</td>
<td>5</td>
<td>1484942194</td>
<td>running</td>
<td>36</td>
</tr>
</tbody>
</table>
After the Migration

Observations

1. Performance issues with the C5.8Xlarge instances

2. Understand the Cache Hits/Misses, bucket evictions for performance

3. Upgraded Enterprise Splunk to version 7.3 – ability to troubleshoot with the MC

4. Migrated the indexer tier to I3.8xlarge

5. Started to add additional indexers to the tier
1. Use the Monitoring Console
2. Log files
3. CLI commands
4. Rest endpoints
1. You cannot revert to non-SmartStore after you migrate – Can be in a mixed mode

2. Read the documentation to understand the Prerequisites

3. Test the configuration on a standalone instance

4. Look at the documentation for common issues
What were the Results
Cost Reduction Efforts

ST1 Storage
~15% Savings

GP2 Storage
~12% Savings

Cheaper I3 RI
~5% Savings