Splunk Upgrade Pre-, In-situ-, and Post-Validation Steps

Author: David Paper, dpaper@splunk.com, @cerby on Splunk-usergroups Slack
Date: 2019-07-11
Version: 1.0.4

Table of Contents

Purpose

The purpose of this document is to provide a set of read-only operations that a Splunk admin would execute as part of the upgrade process for the core deployment of Splunk Enterprise. Guidelines (example: high bucket counts) that include pointers to external resources for configuration updates are not intended as a requirement for adoption prior to kicking off an upgrade. They are present to assist in making the upgrade as smooth as possible.

The target audience for this document is a Splunk admin who is comfortable with day-to-day administration of the platform, has had experience in the past executing application code upgrades and is looking for a soup-to-nuts checklist for what to look for pre-, in-situ-, and post- upgrade for their core Splunk Enterprise environment.

Documentation Review

- ❏ If new to the specific Splunk environment about to be upgraded https://docs.splunk.com/Documentation/Splunk/latest/InheritedDeployment/Introduction
- ❏ Review specific version documentation
  - ❏ All Deprecated Features for the version being upgraded to https://docs.splunk.com/Documentation/Splunk/latest/ReleaseNotes/Deprecatedfeatures
  - ❏ All Known Issues for the version being upgraded to https://docs.splunk.com/Documentation/Splunk/latest/ReleaseNotes/Knownissues
  - ❏ All considerations outlined in https://docs.splunk.com/Documentation/Splunk/latest/Installation/AboutupgradingREADTHISFIRST

Pre-upgrade validation steps

- ❏ Take configuration backups of, and validate ability to restore, all Splunk components
  - ❏ Deployer, Deployment Server, License Master, Cluster Master, Search Heads, Indexers
  - ❏ Backup any KVstores in use on standalone and SHC nodes
    - ❏ For 7.1 and later, backups can be taken without shutting the Search Head down https://docs.splunk.com/Documentation/Splunk/latest/Admin/BackupKVstore
    - ❏ For 7.0 and earlier, backups must be done with the Search Head stopped https://docs.splunk.com/Documentation/Splunk/7.0.0/Admin/BackupKVstore
- ❏ Monitoring Console - benchmark system health
  - ❏ Is completely configured: all SH, IDX, DS, LM, CM (if in use), Deployer (if in use) and HF (if desired) are visible
  - ❏ Review existing resource utilization (CPU, RAM, Disk) for SH and IDX tier, take screenshots for comparison after upgrade in case MC has issues
  - ❏ Review search scheduling and performance, try to correct skipped & deferred searches before upgrade
  - ❏ Review ingestion queues on IDX, ensure they are not filling and not recovering
  - ❏ If using SHC
    - ❏ Review replication latency (***MC -> Search -> SHC -> SHC Configuration Replication***) for errors (top of view) and consistency for time taken (bottom of view)
    - ❏ Ensure that KVStore role is applied to SHC members
    - ❏ Review KVstore oplog (***MC -> Search -> KVStore -> KVStore: Deployment***), specifically “Operations Log Window of KV Store Captain” looking for a value of at least 1 hour, 3-4 hours is ideal for a busy SHC, the higher the better. Values below 15 minutes are problematic and investigation & fixing should commence before upgrade.
    - ❏ On the same view, ensure KVstore in SHC has a captain and one or more secondaries, total queued=0 for all nodes, and Instances by Average Replication Latency view should be in the 0-10s range. Exception for SHCs running ITSI, which can be 30s or higher and be OK.
- ❏ Cluster Master
  - ❏ All data is searchable, RF & SF are fully met
  - ❏ Bundle push to indexers can be completed without issue
  - ❏ Plaintext pass4SymmKey is known in case it needs to be re-keyed into configurations after upgrade
  - ❏ Unique and total bucket counts. If unique bucket count is close to or at 5M (6.6, 7.0, 7.1) or 9M (7.2), investigate the reason(s) for high bucket counts, and consider setting high bucket count configurations on the CM and IDX servers before upgrading. Da Xu’s excellent talk at .conf2017 goes into further detail (https://conf.splunk.com/files/2017/slides/indexer-clustering-internals-scaling-and-performance-testing.pdf, slide 21)
    - ❏ `| rest splunk_server=local /services/cluster/master/peers | stats sum(bucket_count) AS bucket_count_all | eval bucket_count = round(bucket_count_all / 1000 / 1000,2)."M" | eval replication_factor = [| rest splunk_server=local /services/cluster/config | return $replication_factor] | eval unique = round(bucket_count_all / replication_factor /`

```
1000 / 1000,2)."M"| fields bucket_count unique| rename
bucket_count AS "Total Buckets", unique AS "Unique Buckets”
```

Above search to run on the CM

- ❏ License Master
  - ❏ All indexers checking in
  - ❏ Copies of license(s) are archived off host or included in backups
  - ❏ _* indexes are successfully forwarding data to indexing tier (if configured to do so)
- ❏ Deployer/SHC
  - ❏ Validate status of cluster, expect to see fully healthy
  - ❏ Validate ability to complete a bundle push to all SHC nodes without issue
  - ❏ If static captain is in use, know which SHC node is set to captain
  - ❏ Validate kvstore(s) replicate without issue
- ❏ Deployment Server
  - ❏ Validate config reload successful
  - ❏ Validate all FWDs that should be phoning home are doing so successfully
- ❏ Forwarders
  - ❏ Validate current installation base will work with new version of Indexers (SSL & cipher configurations)

Determine forwarder-indexer compatibility

The following table shows the versions of forwarder and indexer that can be used together. As a best practice, use indexers that are the same or higher version than the forwarders.

- An **X** in a cell indicates that this version of forwarder can send event data to the corresponding version of indexer.
- An **M** in a cell indicates that this version of forwarder can send both event data and metrics data to the corresponding version of indexer.
- An **S** in a cell indicates that this version of forwarder can send data to this version of indexer after you change the Secure Sockets Layer (SSL)/Transport Layer Security (TLS) version and cipher suite on the forwarder. See Known Issues in the Splunk Enterprise *Release Notes* for instructions on changing the SSL/TLS version and cipher suite.
- An empty cell indicates that Splunk does not support sending any type of data from this version of forwarder to the corresponding version of indexer.

This table lists version 5.0 for informational purposes only. Version 5.0 forwarders are technically compatible with higher versions of indexer, but Splunk does not provide support for version 5.0 software. Version 5.0 reached its End of Life on November 30, 2017.

Forwarder version	Indexer version			
	5.0 (not supported)	6.0-6.5	6.6	7.x
5.0 (not supported)	X	X	S	S
6.0-6.5	X	X	S	S
6.6		S	X	X
7.x		S	X	M

  - ❏ https://docs.splunk.com/Documentation/Forwarder/latest/Forwarder/Compatibilitybetweenforwardersandindexers#Determine_forwarder-indexer_compatibility
  - ❏ If using DBX, JMX or other apps that require HFs and/or makes external queries, validate they work with the new version
  - ❏ Ensure Forwarder code management tooling can reach all forwarders to be upgraded
- ❏ Indexers

- ❏ Ensure sufficient disk space for new code deploy and local backups
- ❏ Validate indexers aren't running scheduled searches
  - ❏ `index=_internal source="*/scheduler.log" search_group=dmc_group_indexer sourcetype=scheduler |dedup host savedsearch_name | stats count(savedsearch_name) by savedsearch_name`
- ❏ Verify basic searches worked and that all the Indexers replying
  - ❏ `| tstats count where earliest=-5m by splunk_server`

- ❏ Search Tier
  - ❏ Validate upgrade target version works with all apps (searches, dashboards, TAs, external inputs)
    - ❏ Check version compatibility via Splunkbase for Premium and non-Premium apps
    - ❏ Don't forget to test homegrown apps
    - ❏ https://docs.splunk.com/Documentation/Splunk/latest/Installation/UpgradeyourdistributedSplunkEnterpriseenvironment#Test_apps_prior_to_the_upgrade
  - ❏ Have copies of SSL keys, SAML configs, external auth credentials like passwords available in plaintext
  - ❏ Look for failing searches due to missing users in external auth and correct prior to upgrade
  - ❏ Evaluate size of search bundle being pushed to indexers to determine if close to maximum setting, check each search head unless in SHC config
    - ❏ `index=_internal sourcetype=splunkd group=bundles_uploads search_group=dmc_group_search_head | eval baseline_bundle_size_mb=round((average_baseline_bundle_bytes/1024)/1024,1) | chart max(baseline_bundle_size_mb) AS Max_bundle_size by host | eval Max_bundle_size=Max_bundle_size . "M"`

In-situ validation steps

Order of operations for upgrades is important - refer to docs for order based on which components are in the environment and how they are configured

- ❏ https://docs.splunk.com/Documentation/Splunk/latest/Installation/HowtoupgradeSplunk#Choose_the_proper_upgrade_procedure_based_on_your_environment

Several components are single-step upgrades, and only the pre- and post- validation steps apply

- ❏ License Manager
- ❏ Deployment Server
- ❏ Deployer
- ❏ Monitoring Console

After the code upgrade, validate UI login for each component is successful.

- ❏ Cluster Master
  - ❏ The additional steps not outlined in the docs fall into one of two categories:
    - ❏ Pausing the process at key points to let the Cluster Master recover fully
    - ❏ Validation searches at each step
  - ❏ The pausing process is something that hangs a lot of upgrades. Relevant techniques to know when to declare the CM ready for the next step in the upgrade process. A few key points:
    - ❏ The act of monitoring the CM can have significant negative impacts
    - ❏ Specifically, the Clustering UI makes a lot of expensive rest calls that can compete for other tasks on the busy CM as the cluster stitches itself together
    - ❏ It's hard from our logs to determine when the CM was stable and ready for the next step in the upgrade. Reasonable indicators that can be viewed at the OS layer without adding load to the CM
      - ❏ Load average
      - ❏ "`iostats -xz 1`" or "`sar -d`" to determine disk IO
      - ❏ Thread utilization via "`top -H`" or turn on the thread view once top initializes normally ("H"), looking for when threads are no longer pegging a single CPU at 99%+
      - ❏ "`tail -f var/log/splunk/splunkd.log`" The rate that data gets dumped into this log slows significantly when the CM catches up - the type of messages also tend to change.
      - ❏ As soon as the CM load average dropped, IO counts/await times returned to what they were before the upgrade, thread utilization is no longer pegging cores at 99%+ and splunkd.log returns to normal, the CM seemed ready for the next step
        - ❏ Compare data collected in advance of the upgrade from ***MC -> Resource Usage -> Machine*** to what you see happening live as the CM and cluster come up
- ❏ Forwarders
  - ❏ Using MC, ensure that data ingestion continues to flow at the expected rate for the time of day and/or day of the week

- ❏ Indexers (generic)
  - ❏ As indexers are upgraded and brought back online, ensure they are ingesting and participating in search
    - ❏ `index=_internal component=Metrics per_index_thruput earliest=-30m | eval mb=(kb/1024) | timechart span=5m sum(mb) by host`
    - ❏ `| tstats count where earliest=-5m by splunk_server`
- ❏ Indexers (Clustered)
  - ❏ Verify indexers rejoin the cluster as they come back online and are marked “Status=up” and “Fully Searchable=yes” in ***MC -> Indexing -> Indexer Clustering -> Indexer Clustering: Status***.

Post-upgrade validation

- ❏ Monitoring Console
  - ❏ Verify All SH, IDX, utility servers and HF (if desired) are visible
  - ❏ Verify components have correct roles associated with them
  - ❏ Review existing resource utilization (CPU, RAM, Disk) for SH and IDX tier, and compare to before upgrade to see if significant changes are afoot. Use screenshots taken before the upgrade if necessary.
  - ❏ Review search scheduling and performance, determine if searches are skipping when they didn't previously and if so investigate
  - ❏ _* indexes are successfully forwarding data to indexing tier
- ❏ License Master
  - ❏ Verify all indexers checking in
  - ❏ _* indexes are successfully forwarding data to indexing tier
- ❏ CM upgrade and entering Maintenance Mode
  - ❏ Use the load average and iops to determine the CM had calmed down (CM's are general not IO intensive, but IO jumps up considerably when indexer rolling restarts occur)
  - ❏ Watch for kernel swapping regularly via your favorite indicator that the CM is hitting swap
    - ❏ "`vmstat 1`" show pages swapping in/out, "si" and "so" columns
    - ❏ "`iostat 1`" looking at swap device for activity (can get device name from fstab)
  - ❏ After a steady state, review the Clustering Dashboard to ensure
    - ❏ The cluster is searchable
    - ❏ If RF/SF fixup tasks are queued, some of the fixups are in progress
      - ❏ https://conf.splunk.com/files/2017/slides/indexer-clustering-fixups-how-a-cluster-recovers-from-failures.pdf
      - ❏ https://docs.splunk.com/Splexicon:Bucketfixing
  - ❏ Look for search peers that are flapping or restarting outside of a rolling restart
    - ❏ `index=_internal source=*splunkd.log sourcetype=splunkd host=cluster_master component=CMPeer peer transitioning NOT bid | eval transition = from." -> ".to | timechart count by transition`
  - ❏ Search to validate forwarders are heart beating again
    - ❏ `index=_internal sourcetype=splunkd component=CMIndexerDiscovery`
  - ❏ Verify the MC could still see the CM as a search peer
  - ❏ Bundle push to indexers can be completed without issus

- ❏ Search tier, generic
  - ❏ Check to make sure external auth (if configured) is working, including certificates if using SAML or other SSO outside of AD.

- ❏ Validate new version works with all apps (searches, dashboards, TAs, external inputs)
- ❏ Verify basic searches work from each standalone SH, and that all the Indexers replying
  - ❏ `| tstats count where earliest=-5m by splunk_server`
- ❏ Look for skipped or deferred searches that were not exhibiting this behavior prior to the upgrade
- ❏ Evaluate size of search bundle being pushed to indexers to determine if close to maximum setting, check each search head unless in SHC config (run in Monitoring Console search context)
  - ❏ `index=_internal sourcetype=splunkd group=bundles_uploads search_group=dmc_group_search_head | eval baseline_bundle_size_mb=round((average_baseline_bundle_bytes/1024)/1024,1) | chart max(baseline_bundle_size_mb) AS Max_bundle_size by host | eval Max_bundle_size=Max_bundle_size . "M"`
- ❏ Validate users can login utilizing remote auth (if configured) on each SH node

❏ Search tier, SHC specific. In addition to all generic Search tier steps

- ❏ Validate all SHC members visible in the MC
- ❏ Validate the SHC captain and member details in the MC
- ❏ Use the SHC Scheduler Delegation dashboard in the MC and sort the first panel by instance to validate the even distribution of search traffic
- ❏ With a time range before and after the upgrade, measure the time a SHC member is taking to spin up the Search Apparatus, major swings could indicate a problem on the members
  - ❏ `index=_internal uri=*delegatejob* | timechart median(spent) as median_spent max(spent) as max_spent`
- ❏ Look for anything suspicious, specifically errors and warnings in the logs
  - ❏ `index=_internal sourcetype=mongod earliest=-15m`
- ❏ Validate SHC can push a bundle successfully to all indexers, especially if there are multiple indexing clusters a SHC talks to
- ❏ Validate that KVstore comes online on each node and replicates correctly across nodes.
  - ❏ ***MC -> Search -> KVStore -> KVStore: Deployment***

❏ Deployer

- ❏ Validate a bundle can be pushed from Deployer out to all SHC nodes

❏ Deployment Server

- ❏ Validate config reload successful
- ❏ Validate all FWDs that should be phoning home are doing so successfully

❏ Indexers

- ❏ After the upgrade and restart, exercise patience letting the cluster stitch itself together (i.e. go get a cup of coffee or 2)
  - ❏ Validate all the nodes are present in the UI
  - ❏ Validate all data is searchable
  - ❏ Validate cleanup/fixup tasks are moving forward while watching load and IO on the CM
- ❏ Verify basic searches worked and that all the Indexers replying
  - ❏ `| tstats count where earliest=-5m by splunk_server`

- ❏ Verify all indexers are ingesting data
    - ❏ Check S2S port(s)
    - ❏ Check HEC port if configured
- ❏ Review ingestion queues in MC, ensure they are not filling. If queues are filling and not recovering quickly, investigate why.
- ❏ Use the load average and iops to determine the CM has calmed down
    - ❏ Refer to the host resource utilization metrics collected in the pre- steps to determine when the CM has returned to its normal state of operations
- ❏ Scan the internal logs for warnings and errors on the CM and to a lesser extent the indexers (which often are fairly noisy with parsing errors, etc)
    - ❏ `index=_internal sourcetype=splunkd source=*splunkd.log log_level!=info`
- ❏ Repeat earlier steps on the SHC to ensure searches complete and are timely

Conclusion

When all else fails during the upgrade, it's time to engage support (https://docs.splunk.com/Documentation/Splunk/latest/Troubleshooting/HowtofileagreatSupportcase) to help get you back online. If the content of this doc is beyond the comfort level of the Splunk admin team, engaging PS for upgrade help is a good next step.

Acknowledgements

David Paper wishes to thank Mike Barrie for the original inspiration, Nadine Miller and Jane Mulcaster for their thorough feedback on this document.