Nmon Performance monitor Splunk app for Unix and Linux systems Documentation

Release 1.9.0

Guilhem Marchand

Nov 06, 2019

Contents

Nmon Performance is now associated with Octamis to provide professional solutions for your business, and professional support for the Nmon Performance solution.

For more information: Octamis professional support for business

CHAPTER 1

Overview:

1.1 About Nmon Performance monitor for Splunk

- Author: Guilhem Marchand
- First release was published on starting 2014
- Purposes:

The Nmon Performance application for Splunk implements the excellent and powerful nmon binary known as Nigel's performance monitor. Originally developed for IBM AIX performance monitoring and analysis, it is now an Open source project that made it available to many other systems. It is fully available for any Linux flavor, and thanks to the excellent work of Guy Deffaux, it also available for Solaris 10/11 systems using the sarmon project.

The Nmon Performance monitor application for Splunk will generate performance and inventory data for your servers, and provides a rich number of monitors and tools to manage your AIX / Linux / Solaris systems.

Nmon Performance is now associated with Octamis to provide professional solutions for your business, and professional support for the Nmon Performance solution.

For more information: Octamis professional support for business

1.1.1 Splunk versions

It is recommended to use Splunk 6.5.x or superior to run the latest core application release. (in distributed deployments, only search heads may have this requirement)

The last release can be downloaded from Splunk base: https://splunkbase.splunk.com/app/1753

Compatibility matrix for core application:

- **Current major release Version 1.9.x:** Splunk 6.5.x or superior are officially supported

Splunk 6.4 will globally perform as expected, but there might be some unwanted behaviors such as css issues as this Splunk version is not supported anymore by the core application.

Stopped versions for older Splunk releases:

- Last version compatible with Splunk 6.4.x with release 1.7.9 (Splunk certified): https://github.com/guilhemmarchand/nmon-for-splunk/releases
- Last version compatible with Splunk 6.2.x with release 1.6.15 (Splunk certified): https://github.com/guilhemmarchand/nmon-for-splunk/releases
- Last version compatible with Splunk 6.1.x, with release 1.4.902 (not Splunk certified): https://github.com/guilhemmarchand/nmon-for-splunk/blob/last_release_splunk_61x

Compatibility matrix for TA-nmon addon:

Consult the TA-nmon documentation: http://ta-nmon.readthedocs.io

- Both add-ons are compatible with any Splunk version 6.x (full instance of Universal Forwarder)

The TA-nmon add-on is designed to be deployed on full Splunk instances or Universal Forwarders, **it is only compatible with Splunk 6.x.**

The PA-nmon_light add-on is a minimal add-on designed to be installed on indexers (clusters or standalone), this package contains the default "nmon" index definition and parsing configuration. It excludes any kind of binaries, inputs or scripts, and does not collect nmon data.

1.1.2 Index time operations

The application operates index time operations, the PA-nmon_light add-on must be installed in indexers in order for the application to operate normally.

If there are any Heavy forwarders acting as intermediate forwarders between indexers and Universal Forwarders, the TA-nmon add-on must deployed on the intermediate forwarders to achieve successfully index time extractions.

1.1.3 Index creation

The Nmon core application does not create any index at installation time.

An index called "nmon" must be created manually by Splunk administrators to use the default TA-nmon indexing parameters. (this can be tuned)

However, deploying the PA-nmon_light will automatically defines the default "nmon" index. (pre-configured for clusters replication)

Note: The application supports any index starting with the "nmon*" name, however the default index for the TA-nmon inputs is set to "nmon" index.

In distributed deployments using clusters of indexers, the PA-nmon add-on will automatically creates the "nmon" replicated index.

1.1.4 Summarization implementation

Accelerated data models:

Nmon for Splunk App intensively uses data model acceleration in almost every user interfaces, reports and dashboards.

Splunk certification requirements prohibit the default activation of data models acceleration.

Since version 1.9.12, none of the data models are accelerated by default, this is your responsibility to decide if you wish to do so, bellow are the recommended acceleration parameters:

- metrics related data models accelerated over a period of 1 year
- non metrics data models accelerated over the last 30 days (Nmon config, Nmon processing)

Splunk Accelerated data models provide a great and efficient user experience.

Accelerated reports:

The following report(s) use report acceleration feature:

- Volume of Data indexed Today, accelerated for last 7 days
- Number of notable events in Data Processing or Data Collect since last 24 Hours, accelerated for last 24 hours

Please review the *Large scale deployment considerations* documentation.

1.1.5 About Nmon Performance Monitor

Nmon Performance Monitor for Splunk is provided in Open Source, you are totally free to use it for personal or professional use without any limitation, and you are free to modify sources or participate in the development if you wish.

Feedback and rating the application will be greatly appreciated.

- Join the Google group: https://groups.google.com/d/forum/nmon-splunk-app
- App's Github page: https://github.com/guilhemmarchand/nmon-for-splunk
- Videos: https://www.youtube.com/channel/UCGWHd40x0A7wjk8qskyHQcQ
- Gallery: https://flic.kr/s/aHskFZcQBn

1.1.6 Open source and licensed materials reference

- css materials from http://www.w3schools.com
- d3 from Michael Bostock: https://bl.ocks.org
- various extensions and components from the Splunk 6.x Dashboard Examples application: https://splunkbase.splunk.com/app/1603
- dark.css from: http://www.brainfold.net/2016/04/splunk-dashboards-looks-more-beautiful.html
- Take the tour component from https://github.com/ftoulouse/splunk-components-collection
- hover.css from http://ianlunn.github.io/Hover
- free of use icons from /www.iconfinder.com
- Javascript tips (inputs highlighting) from https://splunkbase.splunk.com/app/3171 - https://blog.octoinsight.com/splunk-dashboards-highlighting-required-inputs

1.2 Release notes

1.2.1 Requirements

- Splunk 6.5.x and later Only, for 6.4.x and prior download release: V1.7.9, for prior to 6.2.x download release: V1.6.15, for 6.1.x and prior download release: V1.4.902
- Universal Forwarder v6.x is required for clients
- Universal Forwarders clients system lacking a Python 2.7.x interpreter requires Perl WITH Time::HiRes module available

1.2.2 What has been fixed by release

V1.9.20:

Version 1.9.20

- fix: AIX - field alias for VP_Idle_PCT results in missing field after Splunk behaviour change regarding field aliasing to non existing fields #122

V1.9.19:

Version 1.9.19

- fix: Solaris NMON ANALYSER view issue with drilldown field names causing URL malformed #121

V1.9.18:

Version 1.9.18

- fix: AIX lpar measurement issue in some queries when comparing to cpu_all #118
- fix: Safecenter - user feedback on headings color #119

V1.9.17:

Version 1.9.17

- fix: JQuery vulnerability issue for the integrated viz addons (bullet chart amd radial meter, CVE-2016-10707/CVE-2015-9251) #114
- fix: KVstore collections management interfaces improvements #115
- fix: Nav improvement: merge search and builtin menu into search menu #116
- fix: Nmon summary improvement #117

V1.9.16:

Version 1.9.16 - multiple updates #112

- review props.conf sourcetypes definition for Splunk best practices
- update of horseshoe-meter and bullet-graph to their latest version

- removal of calendar heatmap views managing processing and nmon data availability
- New dashboard: Heatmap daily CPU usage calendar with drilldown
- New alerting scheme with multi-layer KVstore based: Threshold management, frameID mapping and thresholds templating
- Splunk 7.1 minor compatibility issues

V1.9.15:

CAUTION: For Splunk 6.5 and later (for prior versions of Splunk, see requirements below)

This is a major release of the Nmon application and the TA-nmon:

Migration from 1.7.x and prior: http://nmon-for-splunk.readthedocs.io/en/latest/upgrade.html#migrating-from-release-prior-to-version-1-7-x

Migration from 1.8.x: http://nmon-for-splunk.readthedocs.io/en/latest/upgrade.html#migrating-from-release-prior-to-version-1-9-x

- fix: NMON_Data_PAGE data model issue: Comparator '=' has an invalid term #106
- fix: PAGE interface for AIX - duplicated ID #107

V1.9.14:

- intermediate release unpublished

V1.9.13:

CAUTION: For Splunk 6.5 and later (for prior versions of Splunk, see requirements below)

This is a major release of the Nmon application and the TA-nmon:

Migration from 1.7.x and prior: http://nmon-for-splunk.readthedocs.io/en/latest/upgrade.html#migrating-from-release-prior-to-version-1-7-x

Migration from 1.8.x: http://nmon-for-splunk.readthedocs.io/en/latest/upgrade.html#migrating-from-release-prior-to-version-1-9-x

- fix: CONFIG DF dead link in home pages (was replaced by STORAGE ui in 1.9.12)
- fix: props.conf and Nmon config datamodel issue with AIX combo cpu #100
- fix: Nmon Summary dashboard - nmon_span referenced instead of variable #102
- feature request: allow deactivation for auto-refresh feature #103
- fix: Summary dashboard stacking issues with Splunk 7 #104

V1.9.12:

CAUTION: For Splunk 6.5 and later (for prior versions of Splunk, see requirements below)

This is a major release of the Nmon application and the TA-nmon:

Migration from 1.7.x and prior: http://nmon-for-splunk.readthedocs.io/en/latest/upgrade.html#migrating-from-release-prior-to-version-1-7-x

Migration from 1.8.x: http://nmon-for-splunk.readthedocs.io/en/latest/upgrade.html#migrating-from-release-prior-to-version-1-9-x

- fix: Splunk certification requirements update, avoid global default parameter in ui-prefs.conf (config file has been removed) #98
- fix: Splunk certification requirements update, default activation of data model acceleration is now prohibited #98
- fix: DF_STORAGE vs JFSFILE compatibility for STORAGE UI and Dark monitoring

V1.9.11:

CAUTION: For Splunk 6.5 and later (for prior versions of Splunk, see requirements below)

This is a major release of the Nmon application and the TA-nmon:

Migration from 1.7.x and prior: http://nmon-for-splunk.readthedocs.io/en/latest/upgrade.html#migrating-from-release-prior-to-version-1-7-x

Migration from 1.8.x: http://nmon-for-splunk.readthedocs.io/en/latest/upgrade.html#migrating-from-release-prior-to-version-1-9-x

For the TA-nmon complete release notes: http://ta-nmon.readthedocs.io/en/latest/releasenotes.html

- feature: DF_STORAGE and DF_INODES implementation in replacement of JFSFILE (extended file system utilisation statistics)
- feature: New interface for STORAGE statistics management
- feature: metric_catalog lookup implementation
- feature: review and refresh of various interfaces, including comparative and predictive interfaces
- fix: dynamic tokens in dashboard improvements
- fix: Nmon Config datamodel OStype extraction #95

V1.9.10:

CAUTION: For Splunk 6.5 and later (for prior versions of Splunk, see requirements below)

This is a major release of the Nmon application and the TA-nmon:

Migration from 1.7.x and prior: http://nmon-for-splunk.readthedocs.io/en/latest/upgrade.html#migrating-from-release-prior-to-version-1-7-x

Migration from 1.8.x: http://nmon-for-splunk.readthedocs.io/en/latest/upgrade.html#migrating-from-release-prior-to-version-1-9-x

For the TA-nmon complete release notes: http://ta-nmon.readthedocs.io/en/latest/releasenotes.html

- feature: index and search time configuration for the TA-nmon-hec / nmon-logger-splunk-hec (agent less package using the Splunk http input)
- fix: UI Compare - fix frameID mapping for non CSV source data (nmon-logger) #92
- fix: UI Predictive - issue when time range is changed, bad MEM metric label #93
- fix: UI Summary / WOF - token auto-selection issue when time range is changed #94

V1.9.9:

CAUTION: For Splunk 6.5 and later (for prior versions of Splunk, see requirements below)

This is a major release of the Nmon application and the TA-nmon:

Migration from 1.7.x and prior: http://nmon-for-splunk.readthedocs.io/en/latest/upgrade.html#migrating-from-release-prior-to-version-1-7-x

Migration from 1.8.x: http://nmon-for-splunk.readthedocs.io/en/latest/upgrade.html#migrating-from-release-prior-to-version-1-9-x

For the TA-nmon complete release notes: http://ta-nmon.readthedocs.io/en/latest/releasenotes.html

- fix: Large scale issue - Optimize Nmon inventory generation runtime #85
- fix: Nmon inventory - Uptime data analysis issue #86
- fix: Nmon Dark dashboard - missing reset auto-refresh #87
- fix: TOP datamodel issue - error in distributed search for ALL_OS node (nmon summary. . .) #88
- fix: Drilldown correction for the number of last 7 days hosts in home pages #89
- evolution: Large scale consideration - restricted default limits for datamodel acceleration (1y for metrics) #90
- fix: Use nmon_inventory to retrieve configuration data instead of using datamodel #91

V1.9.8:

- intermediate unpublished release

V1.9.7:

CAUTION: For Splunk 6.5 and later (for prior versions of Splunk, see requirements below)

This is a major release of the Nmon application and the TA-nmon:

Migration from 1.7.x and prior: http://nmon-for-splunk.readthedocs.io/en/latest/upgrade.html#migrating-from-release-prior-to-version-1-7-x

Migration from 1.8.x: http://nmon-for-splunk.readthedocs.io/en/latest/upgrade.html#migrating-from-release-prior-to-version-1-9-x

For the TA-nmon complete release notes: http://ta-nmon.readthedocs.io/en/latest/releasenotes.html

- fix: Large scale issue - Optimize search refresh values for large deployments #84
- fix: Nmon Config data model issues in some clustered environments #83
- fix: baseline future charting not working due to mismatch between host and hostname #82
- fix: Large scale issue - Optimize the run time of the Hosts with data within last 7 days #81
- fix: Large scale issue - restrict the nmon_processing data model to the last 30 days by default #80
- fix: report issue - TA-nmon package deployment reporting can includes non deployment events #79
- fix: Large scale issue - Optimize run time of the Volume of Data indexed today report #78
- fix: Large scale issue - Nmon inventory generation report may fail due to report lengh #77

V1.9.6:

CAUTION: For Splunk 6.5 and later (for prior versions of Splunk, see requirements below)

This is a major release of the Nmon application and the TA-nmon:

Migration from 1.7.x and prior: http://nmon-for-splunk.readthedocs.io/en/latest/upgrade.html#migrating-from-release-prior-to-version-1-7-x

Migration from 1.8.x: http://nmon-for-splunk.readthedocs.io/en/latest/upgrade.html#migrating-from-release-prior-to-version-1-9-x

For the TA-nmon complete release notes: http://ta-nmon.readthedocs.io/en/latest/releasenotes.html

- fix: Alerting for CPU is broken since 1.9.5 due to unexpected missing sort _time #73
- fix: nmon data from syslog, missing indexed time creation and OStype and type fields #74
- fix: nmon data from syslog - uptime extraction failure #75
- fix: Alerting - Show the real number of alerts instead of triggered alerts #76

V1.9.5:

CAUTION: For Splunk 6.5 and later (for prior versions of Splunk, see requirements below)

This is a major release of the Nmon application and the TA-nmon:

Migration from 1.7.x and prior: http://nmon-for-splunk.readthedocs.io/en/latest/upgrade.html#migrating-from-release-prior-to-version-1-7-x

Migration from 1.8.x: http://nmon-for-splunk.readthedocs.io/en/latest/upgrade.html#migrating-from-release-prior-to-version-1-9-x

For the TA-nmon complete release notes: http://ta-nmon.readthedocs.io/en/latest/releasenotes.html

- fix: missing oshost tag for ITSI
- fix: Nmon Summary dashboard not retrieving expected results in CPU usage summary with Splunk 6.6.1

V1.9.4:

CAUTION: For Splunk 6.5 and later (for prior versions of Splunk, see requirements below)

This is a major release of the Nmon application and the TA-nmon:

Migration from 1.7.x and prior: http://nmon-for-splunk.readthedocs.io/en/latest/upgrade.html#migrating-from-release-prior-to-version-1-7-x

Migration from 1.8.x: http://nmon-for-splunk.readthedocs.io/en/latest/upgrade.html#migrating-from-release-prior-to-version-1-9-x

For the TA-nmon complete release notes: http://ta-nmon.readthedocs.io/en/latest/releasenotes.html

- fix alerting macros issues: transaction incorrect usage filter out events in excess of allowed limits #70
- fix eventtype related messages for nmon:performance:cpu/mem due to WLM stats #71
- fix Safe Center: reduce the number of searches and add refresh selector dropdown
- fix: CIM compliance improvements and corrections
- feature: introduce a smart auto refresh feature to prevent from having auto refresh enabled when out of current time range

- feature: red highlighting of forms waiting for inputs in views
- feature: Take the tour update

V1.9.3:

CAUTION: For Splunk 6.5 and later (for prior versions of Splunk, see requirements below)

This is a major release of the Nmon application and the TA-nmon:

Migration from 1.7.x and prior: http://nmon-for-splunk.readthedocs.io/en/latest/upgrade.html#migrating-from-release-prior-to-version-1-7-x

Migration from 1.8.x: http://nmon-for-splunk.readthedocs.io/en/latest/upgrade.html#migrating-from-release-prior-to-version-1-9-x

For the TA-nmon complete release notes: http://ta-nmon.readthedocs.io/en/latest/releasenotes.html

- fix certification issues: TA-nmon and PA-nmon_light are not anymore embedded in the core application and must be downloaded externally
- Lower data model acceleration load with per data model schedule configuration #68
- Net stats not associated with time range selector in Nmon Summary
- IOPS and NET stats rendering improvements in Analyser views

V1.9.2:

CAUTION: For Splunk 6.5 and later (for prior versions of Splunk, see requirements below)

This is a major release of the Nmon application and the TA-nmon:

Migration from 1.7.x and prior: http://nmon-for-splunk.readthedocs.io/en/latest/upgrade.html#migrating-from-release-prior-to-version-1-7-x

Migration from 1.8.x: http://nmon-for-splunk.readthedocs.io/en/latest/upgrade.html#migrating-from-release-prior-to-version-1-9-x

For the TA-nmon complete release notes: http://ta-nmon.readthedocs.io/en/latest/releasenotes.html

- Splunk 6.6 tstats issue over non existing field generates nan value instead of null values #67
- Introducing the Dark monitoring dashboard, interfaces review
- Linux Nmon Analyser view issue in DG chart for IOPS
- Nmon external load average extraction failure for some OS
- Be time relative to show indexing evolution in home page
- UPTIME external collection integration
- TA-nmon local/nmon.conf from the SHC deployer is not compatible #23, AIX issues with old topas-nmon, external collection stops on AIX 6.1/7.1, . . .

V1.9.1:

CAUTION: For Splunk 6.5 and later (for prior versions of Splunk, see requirements below)

This is a major release of the Nmon application and the TA-nmon:

Migration from 1.7.x and prior: http://nmon-for-splunk.readthedocs.io/en/latest/upgrade.html#migrating-from-release-prior-to-version-1-7-x

Migration from 1.8.x: http://nmon-for-splunk.readthedocs.io/en/latest/upgrade.html#migrating-from-release-prior-to-version-1-9-x

For the TA-nmon complete release notes: http://ta-nmon.readthedocs.io/en/latest/releasenotes.html

- TA-nmon new branch: fantastic foot print reduction with the fifo implementation, extend data with nmon external, various bug fixes (read TA-nmon release notes)
- PA-nmon and TA-nmon_selfmode are now deprecated (unified by the new TA-nmon features)
- Optimization and rationalisation (globally use the host Splunk Metadata instead of historical hostname field)
- Nmon cores issues (multisearch and tstats incompatible in distributed for the Disk KV generation)

V1.8.6:

CAUTION: For Splunk 6.5 and later (for prior versions of Splunk, see requirements below)

Please review update notes: http://nmon-for-splunk.readthedocs.io/en/latest/Userguide.html#additional-upgrade-notes-migrating-from-release-prior-to-version-1-7-x

- Invalid error number of events count in TCO dashboard when running multiple indexes
- Update of Nmon baseline generation for Disk I/O, and relevant macro update (use DG stats when available)
- app certification failure correction (custom viz issues in savedsearches.conf)
- Addons update to version 1.2.54
- Removal of the static “nmon” index abstraction layer: the app supports natively any index(es) starting with the “nmon” pattern
- Native support for multiple indexes
- Introducing the new frameID management using KVstore, and the frameID mapping management interface
- Improved multi-line events management for rsyslog with nmon-logger agent
- TA-nmon issue: implementation of linux disks groups caused issues with old nmon releases
- Improvement of multi line event management for rsyslog deployments
- populating forms issues in DG interface

V1.8.5:

- Intermediate release unpublished

V1.8.4:

- Intermediate release unpublished

V1.8.3:

CAUTION: For Splunk 6.5 and later (for prior versions of Splunk, see requirements below)

Please review update notes: http://nmon-for-splunk.readthedocs.io/en/latest/Userguide.html#additional-upgrade-notes-migrating-from-release-prior-to-version-1-7-x

- Octamis release, Nmon Performance suite is now a company supported software
- ITSI better compatibility (most ITSI OS module builtin will work, entities dynamic inventory. . .)
- Nmon WOF dashboard correction (single forms mot linked to shared time picker)
- Adding direct link to Data model manager, updating to datasets link, correction to removed interfaces (UI RT)
- Implementation of Linux disks extended statistics (DG* sections), new data model, interfaces, Howto
- Nmon Analyser update, Nmon Summary and WOF will now automatically choose disks extended statistics when available
- Implementation of monitors assets description (monitor description enrichment)
- Allow nmon.conf on a per server basis (/etc/nmon.conf can be set to customize parameters on a per server basis)
- Generic Nmon binaries not recognized for Linux 32 bits systems
- TA-nmon and PA-nmon update to v1.2.51

V1.8.2:

CAUTION: For Splunk 6.5 and later (for prior versions of Splunk, see requirements below)

Please review update notes: http://nmon-for-splunk.readthedocs.io/en/latest/Userguide.html#additional-upgrade-notes-migrating-from-release-prior-to-version-1-7-x

- Drilldown error with Splunk 6.5.1 #60 - Various drilldown errors since 6.5 when a pipeline is split in more than one line (carriage return)
- Errors in Nmon analyser views (Since 6.5 renming an non existing field removes the existing field, this was causing various Disks charts not to be displayed)
- TA-nmon update - Allow host name override #58 (feature request)
- TA-nmon and PA-nmon update to v1.2.50

V1.8.1:

CAUTION: For Splunk 6.5 and later (for prior versions of Splunk, see requirements below)

Please review update notes: http://nmon-for-splunk.readthedocs.io/en/latest/Userguide.html#additional-upgrade-notes-migrating-from-release-prior-to-version-1-7-x

- Technical addons issue with Oracle Solaris 10 using Python interpreter (https://github.com/guilhemmarchand/TA-nmon/issues/11)
- TA-nmon and PA-nmon update to v1.2.48

V1.8.0:

CAUTION: For Splunk 6.5 and later (for prior versions of Splunk, see requirements below)

Please review update notes: http://nmon-for-splunk.readthedocs.io/en/latest/Userguide.html#additional-upgrade-notes-migrating-from-release-prior-to-version-1-7-x

- Implementation of Splunk 6.5 auto refresh features
- Minor improvements and evolutions for best Splunk 6.5 compatibility

V1.7.9:

Please review update notes: http://nmon-for-splunk.readthedocs.io/en/latest/Userguide.html#additional-upgrade-notes-migrating-from-release-prior-to-version-1-7-x

- Adding the PA-nmon_light add-on for indexers that need parsing configuration only (for people that do not want or must not monitor performance of indexers such as Splunk cloud indexers instances)
- Documentation update

V1.7.8:

Please review update notes: http://nmon-for-splunk.readthedocs.io/en/latest/Userguide.html#additional-upgrade-notes-migrating-from-release-prior-to-version-1-7-x

- Add-ons update to 1.2.47 (Linux_unlimited_capture improvement #9, Nmon binary issue with SLES 11.3 #10)
- Adding CONFIG df (filesystems stats) reports & dashboard

V1.7.7:

Please review update notes: http://nmon-for-splunk.readthedocs.io/en/latest/Userguide.html#additional-upgrade-notes-migrating-from-release-prior-to-version-1-7-x

- Drilldown to inventory issues and improvements (Issue #55)
- Performance improvement of the TCO per server search (use datamodel for dcount)
- Add-ons Perl parser (nmon2csv.pl) is lacking OStype field in raw data for TOP/UARG, causing data to be unavailable
- Removal of nmon_inventory OStype mapping had removed OStype mapping for historical data
- Add-ons update (PA-nmon/TA-nmon/TA-nmon_selfmode) to 1.2.46

V1.7.6:

Please review update notes: http://nmon-for-splunk.readthedocs.io/en/latest/Userguide.html#additional-upgrade-notes-migrating-from-release-prior-to-version-1-7-x

- Fix TCO scheduling searches analysis when running in Search Head Cluster
- Updating alerting menu
- Broken links to removed django views (Issue #54)

V1.7.5:

Please review update notes: http://nmon-for-splunk.readthedocs.io/en/latest/Userguide.html#additional-upgrade-notes-migrating-from-release-prior-to-version-1-7-x

- Prevent unwanted server filtering in nmon inventory interfaces due to null fields in nmon_inventory KV
- Correct labels for LPAR stats (for Powerlinux), correct series name to match Physical raw field names
- Integrating the TA-nmon_selfmode as an alternative to the standard TA in case of unsolved unarchive processor failure
- Rewritten Internal dashboard as the Total Cost of Ownership dashboard
- Rewritten Add-ons reporting to provide the global picture of add-ons deployment
- The Nmon app customization tool now offers the option to build a core app that supports Linux only
- Nmon core app Fix Git Issues: #48 to #53
- TA-nmon and PA-nmon V1.2.45

V1.7.5:

Please review update notes: http://nmon-for-splunk.readthedocs.io/en/latest/Userguide.html#additional-upgrade-notes-migrating-from-release-prior-to-version-1-7-x

- Prevent unwanted server filtering in nmon inventory interfaces due to null fields in nmon_inventory KV
- Correct labels for LPAR stats (for Powerlinux), correct series name to match Physical raw field names
- Integrating the TA-nmon_selfmode as an alternative to the standard TA in case of unsolved unarchive processor failure
- Rewritten Internal dashboard as the Total Cost of Ownership dashboard
- Nmon core app Fix Git Issues: #48 to #53
- TA-nmon and PA-nmon V1.2.44

V1.7.4: Major release

Please review update notes: http://nmon-for-splunk.readthedocs.io/en/latest/Userguide.html#additional-upgrade-notes-migrating-from-release-prior-to-version-1-7-x

- Removing of the django deprecated django stack, all views were migrated to simple xml views
- New global bootstrap navigation scheme for easy and efficient user experience with the integrated navigation
- New dynamic help messages will inform about each step of required user action for better user experience
- New major view with the Nmon Wall Of Performance (Nmon WOF)
- Major improvement of Nmon Summary and Nmon Analyser views (active tokens, bar visualization for file systems and much more)
- Rewritten Nmon predictive interface for improved predictive experience
- Embedded Splunk 6.4.x custom viz with fallback to compatibility mode for Splunk 6.3.x
- Improved Power architectures support (PowerLinux Little / Big endian management, LPAR monitor support for Linux, LPAR parsing model)

- Binaries upgrade for Linux (16e / 16f), Linux binaries are now stored in tgz archive and will be uncompressed on client if applicant
- Various bug fixes (Issues #29 to #49)
- Certification app path: The nmon index is not anymore created at installation time for standalone instances
- Certification app path: The core application does contain anymore data generation related object, the TA-nmon must be installed for this to be achieved
- Certification app path: The nmon_inventory file base lookup table were migrated to KV store collection
- inline_customspan macro were renamed to span_nmon for easier usage
- TA-nmon and PA-nmon new packages (V1.2.40)

V1.6.15:

- App certification path, issue 1 execute permission
- App certification path, issue 2 invalid json detected
- App certification path, issue 3/4 duplicated stanzas
- App certification path, issue 5 new line chars in savedsearches.conf

V1.6.14:

- eventtypes / tags implementation over hard index/sourcetype (allow easier multi-index scenarios)
- CIM 4.3 implementation over Performance, Application State, Inventory, Network
- NEW Deployment scenario using Sysog as the transport layer with the nmon-logger third party tool
- #16 (nmon2csv.py logging)
- #17 execute permission in appserver
- #18 html iframe in help
- #19 which python error
- #20 html panel resize
- #21 rename eventgen.conf to .conf.spec
- #22 SuSE Linux identification failure
- #23 nmon 16d / 16c upgrade for Linux binaries
- #24 Prevents bin modifications from customization tools
- TA-nmon and PA-nmon new packages (V1.2.34)

V1.6.13:

- modal windows conversion of transition pages (operating system choice…)
- Fix file text busy error in sh cluster deployment with search head generating nmon data by the core app
- nmon_helper.sh update: Linux and Solaris clients hosts will now cache binaries in run directory
- New monitor: POOLS for AIX systems (extended pools statistics)

- TA-nmon and PA-nmon new packages (V1.2.32)
- Various UI improvements: simplification of multi-series charting, baseline interfaces updates and optimization, custom span macro update (2-3x faster)
- CPU data model update, AIX Nmon Analyser update, new POOLS monitor interface
- App customization Python tool fix (broken links for new app nav bar)

V1.6.12:

- Oracle Solaris 10 clients generates duplicated sarmon processes with TA-nmon v1.2.30 #13
- TA-nmon and PA-nmon new packages (V1.2.31)
- New Application bar navigation menu for better user experience
- Removed single decoration on home pages for better Splunk 6.3 compatibility
- Minor corrections

V1.6.11:

- sarmon (Nmon for Solaris) update to new v1.11 for sparc and x86
- TA-nmon and PA-nmon new packages (V1.2.30)

V1.6.10:

- Removing Home pages searches schedule to limit Splunk load due to the Nmon App (schedules with low interest over cost)
- Smoothing alerting schedule reports (prevents from running them on same round step of 5 minutes)
- Manage artifacts time to live (ttl) for Baseline generation reports and other scheduled reports (limit file system usage on search heads, limit number of artifacts)

V1.6.09:

- nmon2csv.sh hotfix: V1.6.07 changed the temp directory from /tmp to $SPLUNK_HOME/var/run/nmon, but it was lacking creating the directory if required
- This only affects system running the App (core / PA or TA) BUT not generating itself nmon data (such like managing external nmon data)
- TA-nmon and PA-nmon new packages (V1.2.29)

V1.6.08:

- Splitting the kvstore per Performance metric
- Major improvements of baseline generation reports to be valuable at scale
- Baseline interfaces corrections

V1.6.07:

- New feature: Introducing the baseline KV store and baseline interface, chart system key metrics over the baseline to detect system resources utilization derivation and anomalies
- css & html code improvements, code cleaning and xml re-indentation
- Linux binaries 15e/15g updates, set Linux embedded binaries utilization priority by default
- Updates for upcoming sarmon new release
- TA-nmon and PA-nmon new packages (V1.2.28)
- Processing errors detection improvements
- Howto TOP corrections
- Fix for Nmon inventory generation (get latest information instead of last)

V1.6.06:

- New Howtos interfaces: semi interactive SPL request repositories for main monitors
- New pre-built Panels interfaces for main monitors
- Support for CPUnn (CPU usage per logical core), Interfaces and CPU Data Model update
- nmon2csv Python and Perl backend improvements: Manage sections status store per server (allows managing multiple files in realtime mode), fixed blanck space issue in device for nmon2csv.py
- nmon2csv.sh backend will now restrict nmon2csv.py usage to 2.7.x interpreter versions (other will use Perl)
- Nmon App customization Python tool fix (management of token URLs)
- Various interfaces corrections, Home OS pages update
- Removed singlevalue.css for Splunk 6.3.0 compatibility, pre and post label single issue workaround for Splunk 6.3.0
- TA-nmon and PA-nmon new packages (V1.2.27)

V1.6.05:

- Data gaps in Real Time deployment for some random monitors and random timestamp #5
- Data gaps between Nmon collections (occurs between 2 Nmon processes iteration) #6
- Added support for DISKREADSERV / DISKWRITESERV
- TA-nmon and PA-nmon new packages (V1.2.26)

V1.6.04:

- Splunkd unexpected crashes with Splunk version 6.2.4 #4
- TA-nmon and PA-nmon new packages (V1.2.25)

V1.6.03:

- SAFE Center error in events panel for FS Analysis #3
- PA-nmon and TA-nmon add-on tgz archives where wrongly named and affected create_agent.py and Nmon customize script
- Global review of UI and Dashboards names and descriptions for better visibility
- Corrections and improvements of views
- Simple xml conversion of heatmap calendar views
- Added the Help menu in App bar

V1.6.02:

- AIX Hotfix: nmon_helper.sh on AIX generates splunkd error with grep call #2
- TA-nmon and PA-nmon new packages (V1.2.24)

V1.6.01:

- Hotfix for PA-nmon add-on, corrects non working Performance generation on standalone indexers
- Hotfix for Nmon_SplunkApp_Customize.py script: Broken triggered link in Home page when the root directory of App is customized
- Hotfix for create_agent.py: Manage creation of custom agents packages using the shell wrapper
- Improved single alerts drilldown of active alerts to match active time range (Home and Safe Center UI)
- TA-nmon and PA-nmon new packages (V1.2.23)

V1.6.0:

- New nmon2csv wrapper that will automatically choose between Python and Perl tool to convert Nmon raw data, deploy the TA-nmon much more easier than ever
- Introducing the SAFE Center as a central place to manage real time hosts alerting using performance data
- Introducing the TA-NMON management interface to get the better vision of your Nmon and Splunk clients deployment
- Reviewed Home pages for global App, and per type of Operating System
- Eventgen configuration and data samples for chosen main monitors (CPU, LPAR, TOP…) relevant for AIX, Linux and Solaris template hosts, test the App without deploying real clients
- New Wiki documentation now Online hosted at http://nmonsplunk.wikidot.com, Help page now refers to Online Wiki
- TA-nmon and PA-nmon new packages (V1.2.22)
- Various UI corrections

V1.5.30:

- SUSE Linux hotfix: nmon_helper.sh typo error leading in failing to identify best binary for Suse Linux clients
- nmon_helper.sh hotfix: Some cases still lead to processes duplication at boot time for some OS, improved and simplified code will prevent this
- TA-nmon and PA-nmon new packages (V1.2.21)

V1.5.29:

- nmon_helper.sh hotfix: Under certain circumstances and after reboot, multiple nmon instances may be generated, this new improved version will prevent this.
- TA-nmon and PA-nmon new packages (V1.2.20)

V1.5.28:

- Simple xml conversion of Nmon Internal interface, TOP Usage (bubblechart) dashboards
- Simplification of custom span definition in views, added a new form input "span" available in all interfaces
- Correction of IBM Pool usage alerting (bad CPU % reported), added file systems excluding lookup
- nmon_helper.sh update: Improvements code (All OS) to help preventing launching multiple nmon instances
- TA-nmon and PA-nmon new packages (V1.2.19)

V1.5.27:

- AIX Pool usage interface correction (relative and real time interfaces): non working token for monitor other than VP usage reporting (VP usage in % of its capacity)
- CPU_ALL / LPAR data model update: correcting evaluation of VP usage in % of capacity
- Data dictionary update (formula correcton for VP usage in %)

V1.5.26:

- nmon2csv.pl (Perl Nmon converter) update: Fix BBB config section extraction failure when BBB is lately generated (mainly for Linux hosts)
- nmon_helper.sh update: for AIX, prevents nmon instance identification failure if not using topas-nmon
- nmon_helper.sh update: for Linux (Ubuntu), added support for older releases (with no os-release file available)
- nmon2csv.py (Python Nmon converter) update: Windows Hotfix, broken directory creation fixed
- TA-nmon and PA-nmon new packages (V1.2.18)
- Nmon customization Python tool update: Fix customization failure due to the TA-nmon removing in V1.5.25 (only the tgz archive is kept now, for size optimization)
- Data dictionary visualization update: Added overflow scollbar and fixed low resolution truncation

V1.5.25:

- SEA Data model correction (SEACHPHY not reported)
- Correction of data volume comparison in Home page
- nmon_helper.sh maj update for Linux: Linux identification allows using best embedded nmon binary
- TA-nmon now brings nmon binaries for most common Linux OS and hardware
- New nmon.conf option allows giving priority to local nmon binary in PATH or embedded binaries
- TA-nmon and PA-nmon new packages (V1.2.17)
- TOP UI maj update: Aggregate stats per host or globally, Active drilldown links to stats per PID for the clicked Command invocation
- New embedded alert to watch for potential nmon processes duplication on hosts
- Internal Stats UI update: Added message for admin rights acess to internal indexes
- Web FrameWork dashboards maj update: Improved html code to correct fit to screen issues

V1.5.24:

- nmon_helper.sh hotfix: Corrections and improvement for App related nmon instances identification
- Introducing the very first version of Nmon Splunk Alerting, Alerting templates rules for common monitors (% CPU, Real and Virtual Memory…)
- Added support for SEA AIX Statistics (Shared Ethernet Adapter)
- Corrected NFS V4 AIX options which was incorrectly verified in nmon_helper.sh
- TA-nmon and PA-nmon new packages (V1.2.16)
- New data model for SEA statistics, associated SEA interface
- Data dictionary update (inclusion of SEA metrics)
- Home and Home AIX pages update

V1.5.23:

- Rewritten version of the nmon_helper.sh to definitively solve trouble with the input script
- The nmon_helper.sh has been a root cause of various troubles because it was (with more or less success) attempting to manage process duplication and so
- Part of the script has been rewritten from scratch, to be simple and effective with very few conditions
- The script won't try to kill anything now (common trouble for people) and will be based pid file to get its current status
- TA-nmon and PA-nmon new packages (V1.2.15)

V1.5.19 - V1.5.22:

- nmon_helper.sh update

V1.5.18:

- IOADAPT interface hotfix: Missing span in tstats command causing avg eval deviation and charting issues
- nmon2csv.py / nmon2csv.pl update: Added support for AIX Fiber Chanel metrics (FC*)
- nmon_helper.sh update: Prevent from trying to verify non existing processes (error message in Solaris, no such process)
- TA-nmon and PA-nmon new packages (V1.2.10)
- New data model for FC statistics, associated FC interface
- AIX Nmon Analyser update: set IOADAPT charts in stack mode
- Data dictionary update (inclusion of FC metrics)
- Home and Home AIX pages update

V1.5.17:

- Solaris update: Added Solaris specific Performance monitors, specially WLM statistics for Zone management
- New Solaris interfaces and Django Dashboard for WLM Statistics, Disks service and wait time
- nmon2csv.py / nmon2csv.pl update: Code improvement, Solaris update
- nmon_helper.sh / nmon.conf update: Solaris update (deactivation of CPUnn data, management of VxVM activation)
- TA-nmon and PA-nmon new packages (V1.2.09)
- New Data Model for Solaris WLM Stats, Disks Service and wait time
- Nmon Config Data Model update for type of processor identification corretion for Solaris
- Data dictionary update

V1.5.16:

- Linux maximum number of devices is now overcharged by nmon.conf to allow easy customization for very large systems
- nmon_helper.sh update for Linux max devices overcharged update
- nmon2csv.py / nmon2csv.pl hotfix: Prevent partial Configuration extraction in Real time mode for very large systems (BBB collects may occurs after Performance collect starts)
- TA-nmon and PA-nmon new packages (V1.2.08)
- Nmon Inventory Data Model update to prevent OSfilter being null in case of unexpected Operating System recognition (hosts would be listed in Any OS)
- Nmon Inventory Data Model update to improve Linux distribution and vendor identification, inventory saved-search update and minor Linux sections update in inventory interfaces
- Minor corrections in CPU_ALL interfaces (2 decimals rounding)
- Help update

V1.5.15:

- Data Model conversion and important performance optimization of Nmon Analyser views for AIX / Linux / Solaris
- MEM Linux interface correction for table stats
- Various optimizations of interfaces

V1.5.14:

- Introducing the new Data Dictionary to provide through a dendogram user interface the capacity to explore the App data definition: Which metrics are available, Operating systems applicable… and more !
- Major update of the nmon_helper.sh input script update: Improvement of process identification, prevents from killing non App related nmon instances, analysis of Linux return code…
- TA-nmon and PA-nmon new packages (V1.2.06)
- MEM Linux interface correction (duplicated OS filter, _time shown in chart)
- Minor AIX File datamodel update
- Global update of interface to add metric definitions for more complex interfaces
- Added information panel in Nmon Analyser views and Nmon Summary
- Nmon_SplunkApp_Customize.py script update for dendogram compatibility
- Update of scheduled search for error reporting (added the Data collect error reporting), Home page update
- Added the Know Issues, available as link from the Help page, Help page update

V1.5.13:

- Missing Wildcard in Disks DataModels that would lead to ignore devices in Data Model stats (introduced in V1.5.12 that was not published as public release)

V1.5.12:

- Data Models rebuild for disks sections: Main Disk datamodel has been split by type (DISKXFER, DISKBUSY…) for better acceleration building (large data volume) and better search performances
- Update of Disks interfaces and Nmon Summary interface
- Minor css correction for django interfaces

V1.5.11:

- shebang correction in nmon_cleaner.py
- python subversion check correction in nmon_cleaner.sh

V1.5.10:

- Migration of var directories used by the App to generate, monitor, index and clean nmon and associated data
- The main var directory is now $SPLUNK_HOME/var/run/nmon, this especially prevents from loosing data during indexing time if app upgrade occurs (deployment process)
- New versions of all third party scripts
- TA-nmon and PA-nmon new packages (V1.2.05)
- Documentation update
- Correction for Volume of data indexed saved search (bad volume reported in cluster), Home update
- Nmon Inventory update: regular expression to ignore Linux LSB_version patterns (improvement of Linux distributions recognition)
- First level of drilldown UI update

V1.5.09:

- nmon_helper.sh corrective hotfix (collision when nmon is in bin/)
- nmon_cleaner.sh improvement: Verify Python version meets 2.7.x requirements before using py script (User Perl version if not met)
- TA-nmon and PA-nmon new packages (V1.2.04)

V1.5.08:

- nmon_cleaner.sh corrective Hotfix
- TA-nmon and PA-nmon new packages (V1.2.03)

V1.5.07:

- New frontal sh script nmon_cleaner.sh to encapsulate both Python and Perl cleaners, if Python not locally available, the Perl version is now automatically used (configuration simplification)
- TA-nmon and PA-nmon new packages (V1.2.02)
- macros.conf update for custom span definition: 1 minute minimal span value is now the default standard (equal to the default value of nmon.conf)
- Minor correction of Nmon Inventory views (single forms drilldown issue)
- New source stanza in props.conf to Allow managing nmon.gz gzip compressed file archives without further more configuration (cold nmon repositories)
- nmon_helper.sh update: Definitively fixed detaching issue for Solaris!
- nmon2csv.py update and correction (data not being reported if count less than 3 events)
- Hotfix 20150211 for Windows users: fix non compatible epoch time conversion leading to nmon2csv failure
- source default field override by default to prevent multiplication of Metadata entries
- Nmon customization resource script cleaning improvement

V1.5.06:

- Error in CPU_ALL tables stats for Wait % value
- Broken image link in Nmon_ANALYSER_AIX

V1.5.05:

- New Application logo !
- Incorrect link to django interfaces in TOP processes views
- Data Model update for VM section (Linux, Solaris), update of associated interfaces
- Data Model conversion of heatmap cal view (data), improvement of processing calendar views
- Data Model conversion of Nmon Analyser views

V1.5.04:

- TOP Processes Activity (CPU, MEM) dj dashboards improvements: Added a table stats to link Commands by associated hosts

V1.5.03:

- OStype filtering error in Nmon Summary interface
- Nmon Compare interface corrections and improvements

V1.5.02:

- Error in LPAR Pool interface for Pool ID identification in table stats
- Nmon Summary interface corrections and Data Model conversion
- TOP Data Model update (added All OS node to allow Nmon Summary update)
- Various minor corrections of Interfaces
- Nmon Analyser views populating inputs update
- Home pages update for OS Filter token to be passed to Nmon Summary & Analyser

V1.5.01:

- Minor corrections in LPAR interfaces (hostname populating not associated with frameID)
- Fixed AIX compatibility with nmon_helper.sh
- NFS macro correction (macros.conf)
- Minor width corrections for redesigned django interfaces
- New version of TA-nmon: Version 1.2.01 and PA-nmon: 1.2.01
- Schedule of Nmon Inventory data from accelerated datamodel to run every hour

V1.5.0:

- Important new releases of Python and Perl nmon2csv converters with now real time capacity
- The App can now manage a single real time Nmon file (nmon binary is running) with the capacity of real time / cold data analysis detection
- Main nmon options (interval and snapshot, NFS activation) can now be controlled through a Splunk fashion default/local nmon.conf file (upgrade resilient)
- All new Data Models for each type of Nmon data, Using the Data Model acceleration, the App run faster than ever
- Global review of All interfaces and dashboard, take benefit of Data models acceleration, improved design, best functionalities
- Important improvement of the Nmon inventory data generation using the Data model acceleration (specially solves performance issue while generating nmon inventory)
- Brings new Python and Perl nmon_cleaner tools to manage retention of nmon raw data files and prevent potential issues with temporary csv data

V1.4.92:

- New Accelerated Data Model for Nmon Config: Configuration items extraction
- Updated associated saved search and home page

V1.4.91:

- Improved Linux Memory interface Analysis
- Update of Linux Nmon Analyser interface
- Minor views improvements
- Include the optional Python script "nmon_cleaner.py" that can be used to purge csv repositories, based on file retention
- New version of TA-nmon: Version 1.1.34 and PA-nmon: 1.1.27
- Nmon SplunkApp Customize tool updated: Deleted useless removal of pyo files (now forbidden files for package creation)

V1.4.90:

- Decimals rounding for evolution trend JavaScript decoration (home page and comparison ui decoration)
- Applying a dispatch ttl of 4 hours for Nmon Inventory lookup table generation savedsearch to prevent affecting user quota
- nmon2csv Python converter update: Fix for old Linux Nmon releases that have unexepected timestamp id in csv header, code cleaning (redundant espaced chars)
- New version of TA-nmon: Version 1.1.33 and PA-nmon: 1.1.26

V1.4.89:

- Home page improvements with volume of data indexed and reported errors trends decorations
- Comparison interface improvements with range icon decoration (equal, increase, decrease)
- New improved version of calendar data Analysis
- Improvements of Nmon Summary interface
- Improvement of hosts accounting (mainly for AIX, redundant hostnames are now accounted by serial numbers)
- nmon_helper.sh input script update: Allow master node execution for cluster monitoring
- New version of TA-nmon: Version 1.1.32 and PA-nmon: 1.1.25
- Nmon SplunkApp Customize tool updated: Missing string replacement for dispatch ui in savedsearches.conf
- Missing AIX_LEVEL in table stats of Nmon inventory interfaces
- Help update with a proper and improved Splunk Distributed Cluster monitoring using Nmon App (includes Splunk 6.2 search head clustering compatibility)

V1.4.88:

- nmon2csv Python converter update: Correction for bad header identification due to unexpected blank space after comma, String replacement correction that could affect LPAR section for partitions with no pools (IBM P5)
- New versions of TA-nmon: Version 1.1.31 and PA-nmon: 1.1.24
- props.conf of core App update (workaround for LPAR section with data previously indexed and affected by the string replacement error)
- Update of default metadata macros system export

V1.4.87:

- Remove the App setting page (setup.xml) which generates more troubles than benefits, replaced by links to main items in the configuration menu
- Corrected Volume Index today savedsearch
- Important correction of auto-span macros: under some circumstances, the macro was generating unexpected span values, and gaps in charts or “too much data” error message
- Correction of MEM views for Linux and Solaris
- Added missing Host pattern filtering in Predictive Web framework view
- Help update

V1.4.86:

- Nmon SplunkApp Customize tool updated: Missing string replacement for UARG links in Web Framework views
- Missing Host populating filter in Web Framework views: “D3chart: Processes CPU and Memory Usage”
- Corrected scale names in MEM interfaces
- Activated acceleration over report “Generate NMON Inventory Lookup Table”

- Pivot models update
- Added the number of nmon files proceeded in Application Internal Statistics

V1.4.85:

- Added Host populating filter in all views to facilitate management of very large number of hosts
- Improved Nmon Summary interface: Added Single links, improved memory analysis accuracy
- Navbar color changed
- Limited the minimal span to 20 sec instead of 10 sec, sometimes the Nmon collect may miss a measure which generates gaps in charts when looking at very small time ranges This will prevent this and does not change the minimal interval definition if the Nmon data has been generated out of Splunk. (unless interval inferior to 20 seconds)
- Nmon Analyser views update: Added NFS sections for AIX/Linux, migrated row grouping to panel mechanism
- Removed useless LPAR views for Linux
- Update and improvements of Web Frameworks dashboards

V1.4.84:

- Typo error in unarchive_cmd configuration line for props.conf of the core App (repeated unarchive_cmd but does not affect the good work of the Application)

V1.4.83:

- The nmon2csv converter is now officially available in 2 flavors, Python as the default converter, and Perl as the alternative converter
- Systems lacking Python or having trouble with it can use the Perl converter that has the same level of functionalities: Processing statistics, Prevention of data inconsistency, error logging…
- Release V1.0.9 of the Python nmon2csv converter (log truncated prevention)
- Updated help page
- New version of TA-nmon: Version 1.1.30 and PA-nmon: 1.1.23

V1.4.82:

- nmon2csv converter updated: Improvement of logging Splunk compliance, portable shebang update
- Nmon SplunkApp Customize tool updated: Important correction for non working calendar heatmap views due to customization, portable shebang update
- Removed useless nmon_data source overwrite in inputs.conf for csv indexing state
- Added report for NMON related splunkd events
- New versions of TA-nmon: Version 1.1.29 nd PA-nmon: 1.1.22

V1.4.81:

- Improved version of the "Nmon_SplunkApp_Customize.py" Python customizer tool (v1.0.2): Code improvement, backward compatibility with Python 2.6.x
- Added a new advanced macro with args used with manual interacts in Prediction UI (code improvement)
- Web Framework views improvements, minor corrections

V1.4.8:

- **nmon2csv Python converter update:** . PEP 8 Python compliance, various syntax corrections . Added the Parameters section to facilitate user customizations
- New versions of TA-nmon: Version 1.1.28 nd PA-nmon: 1.1.21
- Help update
- minor macros.conf update for Solaris inventory improvement, improved version of Solaris inventory UI

V1.4.7:

- Introducing the "Nmon_SplunkApp_Customize.py", a simple to use Python tool that allows customizing the Application to fit your needs and company criteria, such as:
- Customize the Appication Index Name (default: nmon)
- Customize the Application Root Directory (default: nmon)
- Customize the TA NMON Root Directory (default: TA-nmon)
- Customize the PA NMON Root Directory (default: PA-nmon)
- Customize the local CSV Repository (default:csv_repository)
- Customize the local Config Repository (default:config_repository) The Python tool uses optional command line arguments and can be used over each future release, such that your customizations are automatically integrated and updating the Application is easy as usual.
- Help update

V1.4.6:

- Missing PID filter in AIX TOP processes view, Added UARG interface link and PID filter in Web Framework TOP views
- Migrated default nmon repository from monitor to batch to prevent local nmon data missing when indexing large nmon volumes from central shares (does not affect central shares configuration, only for local host monitoring)
- **nmon2csv converter update:** . UARG section correction for AIX systems . Inconsistency Data prevent improvements . Logging improvements (some functional messages were logged instead of indexed within nmon_processing sourcetype)
- nmon_helper collecter update: Avoir deleting existing nmon files in default nmon_repository to prevent missing local nmon data, this operation is now done by Splunk (migrating from monitor to batch)
- New versions of TA-nmon: Version 1.1.27 nd PA-nmon: 1.1.20
- Corrected UARG Interfaces for AIX
- Inventory macros corrections, Improved versions of Inventory Interfaces for AIX, Linux

- Help update

V1.4.5:

- **nmon2csv converter update:** . Avoid blank line creation when running under Windows OS . Added NFS Statistics extraction: Sections NFSSVRV2 / NFSSVRV3 / NFSSVRV4 for Server, NFSCLIV2 / NFSCLIV3 / NFSCLIV4 for client . Added UARG data extraction (full command argument of TOP processes, needs to be activated in nmon command line to be available)
- New interfaces for NFS Statistics (AIX / Linux)
- nmon_helper collecter update: Improved default command line options for AIX / Linux
- New UARG interface, updated versions of TOP interfaces with link to UARG, improvement of Nmon Config interfaces
- New versions of TA-nmon: Version 1.1.26 nd PA-nmon: 1.1.19
- Help Page improvements: Various corrections, new Table of content with sections links, updated FAQ

V1.4.4:

- nmon2csv converter update: Added interval and snapshots values in data, to be used in conjunction with the new custom span macro embedded within this release
- **New version of custom span macros used with the App to identify the better span value for data accuracy, the new version**
  . Always use a minimal span value that matches the lower level of the Nmon interval between 2 measures . Always have charts with no gaps no matters the Nmon interval in data (if there is no gaps in data) . Allow an automatic identification of the interval per host, so that you can have hosts with different interval values . No more requirement of setting a local version of macros.conf if your Nmon data is less accurate than the proposal one in Nmon Collect
- All views updated to match the new macro syntax (args required, type and hostname)
- Help update
- OSfilter correction in some views
- New versions of TA-nmon: Version 1.1.25 and PA-nmon: Version 1.1.18

V1.4.3:

Windows OS compatiblity for Nmon Data conversion: * nmon2csv.py (Version 1.0.3) update for Windows Compatibility * Added OS type, Python version and Splunk Root Directory in output processing * Added inputs.conf.forWindows and props.conf.forWindows to allow users who need to convert Nmon files under Windows OS * Help update * New versions of TA-nmon as of Version 1.1.24 and PA-nmon as of Version 1.1.17

V1.4.2:

- Review and improvement of default config files inputs.conf and props.conf
- Using variable path instead of full path ($SPLUNK_HOME)
- Change the source stanza in props.conf to match any nmon file no matters where it is located to simplify adding custom repositories (now possible from Splunk Web)
- Using the Python emebedded interpreter for standard Application and PA-nmon (Forwarders don't have Python embedded, so must have the host running TA-nmon)

- Web Framework views improvement: Added auto_cancel parameter to prevent Real time searches from running after leaving interfaces
- New Versions of Calendar views: Data Processing and Performance Monitors Analysis
- Home page update: Added the Number of errors reported
- Help update
- Various minor corrections
- nmon2scv converter update: Minor version with code cleaning
- New versions of TA-nmon as of Version 1.1.23 and PA-nmon as of Version 1.1.16

V1.4.1:

- nmon2csv converter update: Minor regex optimizations, added nmon2csv version in output processing (nmon_processing sourcetype)
- Default host field override based on events data for nmon_data and nmon_config: corrects the host field when indexing nmon files from central shares instead of Forwarder hosts
- Increased the number of max event lines for nmon_config (prevents event breaking for very large system)
- New versions of TA-nmon as of Version 1.1.22 and PA-nmon as of Version 1.1.15
- Duration evaluation corrected in Application Internal Statistics interface
- Help updated mainly for the new Python nmon2csv converter and some other corrections

V1.4.0:

- The Nmon converter tool (formerly nmon2csv) has been fully rewritten in Python 2.x: More Data control, better processing output, lower resources usage, lower volume of data generated, no more empty files generation… and much more !
- Application Internal Statistics update to take advantage of the new Python converter (conversion stats: elapsed time, volume of Nmon raw data converted, numbers of encountered errors…)
- Reports updates (Activity and Errors in Data Collect / Processing)
- Added pre-packaged Nmon binary for powerlinux systems (ppc32/64)
- Removed the Nmon cleaner (nmon_cleaner.sh) which is not required anymore (no more generation of empty csv files with the new nmon2csv Python converter)
- New versions of TA-nmon as of Version 1.1.21 and PA-nmon as of Version 1.1.14
- Various updates and corrections

V1.3.6:

- nmon2csv converter update, Blank line issue correction: If the nmon file contains several blank lines, this could lead the script not to be able to convert data successfully, this is has been corrected in this release by filtering blank lines while reading from stdin
- Added text input filter in Nmon_Summary and Nmon_Analyser views to allow pre-filtering hosts using a user pattern

- Corrected Nmon_Summary and Nmon_Summary to keeps stats in "Waiting for input" mode until user's selection
- Added the CPU datasource identification for Nmon_Summary and Nmon_Analyser views
- Update of nmon_helper.sh to prevent users from trying to launch nmon data collect non supported systems
- New input script "nmon_cleaner.sh", prevents empty csv files kept undeleted by Splunk which may sometime happen
- Added reports nmon_cleaner activity / Nmon collect errors
- New versions of TA-nmon as of Version 1.1.20 and PA-nmon as of Version 1.1.13

V1.3.5:

- Intregated type of OS filtering based on csv lookup table instead of raw Nmon data to improve time required to populate hosts lists (requires a first run to be available)
- nmon2csv converter update: improved processing output logging (nmon_collect sourcetype)
- minor regex update for nmon_config
- New versions of TA-nmon as of Version 1.1.19 and PA-nmon as of Version 1.1.12
- Removed "Inactive" OS type choice when useless within interfaces

V1.3.4:

- OS type identification optimization: time of treatment drastrically reduced using dedup at top of nmon_config based search
- New UI "NMON Summary" for Light System load Analysis, available ton top of Home pages
- Nmon inventory important update, complete regex extraction of available config elements for AIX/Linux/Solaris
- Corrections for NMON Analyser views: Missing wildcards in some charts for disks aggregation
- New scheduled savedsearches which generates NMON inventory data used in inventory UIs, update NMON App setup page to allow customization
- nmon2csv converter update: added nmon data structure verification to prevent data inconsistency: Buggy nmon files (ZZZZ lines truncated) and obsolete Nmon versions
- Added a simple report to show NMON Processing Errors
- Added a simple report that shows NMON Collect Activity
- nmon_helper.sh update to clean Solaris sadc output
- New versions of TA-nmon as of Version 1.1.18 and PA-nmon as of Version 1.1.11

V1.3.3:

- Improved nmon2cv.pl time format for processing output, correction in props.conf
- Increased number of devices taken in charge while converting data, up to 150x5 devices for very large systems (nmon2csv update)
- Improved the identification of the number of logical CPUs for TOP section
- Introduced CPU load increase factor by SMT mode for AIX TOP processes views

- New section for AIX: DISKRIO and DISKWIO for read/write I/O and new AIX Interface
- New versions of TA-nmon as of Version 1.1.17 and TA-nmon as of Version 1.1.10
- Improved nmon_data section in props.conf
- Corrected nmon_processing django analysis interface (number of nmon files processed per day)
- Corrected default metadata (admin as default owner of views)
- Global review of all Interfaces with various corrections and improvements
- Interfaces with devices (NET*, DISK*, JFS*, IOADADPT) have been converted into multi-hosts selection,multi-series charts
- FileSystem filtering by pattern input (JFS* monitor)
- Pivot Models update

V1.3.2:

- Update of nmon converter (mmon2csv.pl): Corrected TOP section header and timestamp pattern to match updated props.conf
- New versions of TA-nmon as of Version 1.1.16 and TA-nmon as of Version 1.1.9
- Improved timestamp recognition of events
- setup.xml correction (wrong description in polling interval)
- Web Framework Toolkit upgraded to version 1.1
- Updated django Processes views "D3chart: Processes CPU and Memory Usage" to limit timecharts to top 20 processes (prevents browser hangs)
- Various minor corrections in views

V1.3.1:

- All New rewritten Comparison Interface in Simple XML: Compare various Metrics (CPU, I/Os, Network…), Evolution Trend with Single value decoration, Overlapped chart of periods, Multi Hosts selection
- Added Time Filtering input forms for all Interfaces (filter statistics by hour and type of days, business days, nights…)
- NMON logo and margin insertion in simple xml views (css customization)
- Added filter to prevent bad identified devices for NET section under Linux
- Added auto-refreshed indexing volume of the day in Home page
- Help update

V1.3.0:

- Solaris issue with nmon_helper.sh

V1.2.9:

- Optimization of CPU Load generated by the nmon App for Forwarders and Indexers by avoiding multiple nmon files to be kept in nmon_repository directory
- Removed input script "purge_nmon_repository.sh" from bin and App setup
- Updated nmon_helper.sh third party script
- New resources versions: PA-nmon (1.1.7) and TA-nmon (1.1.14) versions
- Update is highly recommended, please clean the old input "purge_nmon_repository.sh" from your local/inputs.conf, if any.

V1.2.8:

- Deactivated third party scripts nmon_helper.sh and purge_nmon_respository.sh in default App configuration to prevent splunkd crash on Max OS X installation

V1.2.7:

- Views and dashboards updates: Auto refresh for single forms in home page, Improved placements of forms in views for better options visualization
- Macro custom span definition update to correct Real Time span definition (issue introduced in last version with span accuracy improvements)
- Update of nmon_helper.sh to suppress useless log pollution of Solaris sadc binary in nmon_collect sourcetype
- New resources versions: PA-nmon (1.1.6) and TA-nmon (1.1.13) versions

V1.2.6:

- Update of purge nmon repository third party script to correct compatibility issue with Solaris 10
- New resources versions: PA-nmon (1.1.5) and TA-nmon (1.1.12) versions
- Update of nmon_helper.sh to improve accuracy of nmon measures, one measure each step of 10 seconds in default configuration
- Accuracy improvement of custom span definition macros for small time ranges (added 10s / 30s)
- Update of setup.xml to allow interval custom settings of nmon_helper.sh execution
- In default configuration, data will be refreshed each minute (2 minutes before) for Real Time monitoring accuracy
- Web Framework views corrections for Real Time search compatibility
- Help update

V1.2.5:

- Components from Web FrameWork Toolkit have been incorporated within the App core, it is not required anymore to install the WFT as a requirement
- Various corrections and optimizations of Web Framework dashboards
- Added missing OS Type filtering in Web Framework views

- Adding textinput filtering by Command in TOP interfaces for AIX / Linux / Solaris
- Added FAQ in Help Page
- Updated Installation section of Help Page
- Removed useless indexes.conf in TA-nmon, new TA-nmon as of Version 1.1.11

V1.2.4:

- An error has been introduced in Version 1.2.2 and 1.2.3 in props.conf of TA-nmon and PA-nmon
- Corrected Versions of TA-nmon / PA-nmon

V1.2.3:

- nmon2csv.pl correction for to clean cksum hash reference file upon check operation iteration
- New TA-nmon (V1.1.9) and PA-nmon (V1.1.3) versions
- Help updated for incorrect splunkforwarder rc-init management when a Splunk instance is present in the same machine (Cluster topology)

V1.2.2:

- nmon2csv.pl correction for missing timestamp in nmon_processing sourcetype
- New TA-nmon (V1.1.8) and PA-nmon (V1.1.2) versions
- Indexes First and Last Events statistics correction

V1.2.1:

- Update and improvement of all simple xml views (Nmon Metric interfaces) to implement the Multiselect module for multi Hosts / Devices selection that came with Splunk 6.1
- Various views corrections

V1.2.0:

- Introducing the “PA-nmon” App available in resources directory for Cluster Topology (cluster bundle configuration) to be installed in peer nodes of a cluster
- Help update with a new full tutorial for Cluster topology integration
- All pieces of a Splunk Cluster can now be analysed with Nmon Performance data

V1.1.10:

- Solaris 10 correction for sparc arch (nmon_helper.sh update)
- New Forwarder version as of Version 1.1.7 (Solaris 10 sparc arch issue)

V1.1.9:

- Solaris 10 incompatibility correction with nmon_helper.sh third party script
- New Forwarder version as of Version 1.1.6 (Solaris 10 incompatibility with previous version)

V1.1.8:

- New version of Forwarder App "TA-nmon" As of version 1.1.5 (nmon_helper.sh update, pre-packages for Solaris sparc and X86)
- Update of nmon_helper.sh third party script which includes now pre-packages for Solaris sparc and X86
- CSS updates
- Help page update

V1.1.7:

- Unification of various scripts for both nmon and TA-nmon (local data collect, remote collect through agents)
- md5sum operations has been replaced by cksum for AIX compliance
- Data collect is now fully compatible with AIX OS

V1.1.6:

- Images paths corrections for reverse proxy compliance

V1.1.5:

- New version of NMON Forwarder App (for Linux and Solaris, upcoming for AIX) which is now unified to be fully compliant with Splunk Deployment schemas
- Forwarder App renamed to "TA-nmon", input script unified for Solaris and Linux
- Help updated with deployment server tutorial, integration of Deployment server configuration and NMON forwarder App deployment
- Broken link correction in Home page for AIX JFSINODE
- NMON Analyser OS filtering missing for Solaris

V1.1.4:

- New version of third party script nmon2csv.pl to integrate auto extraction of full host configuration (AAA and BBB Nmon sections)
- New version of lightweight Nmon App forwarder version (version 1.1.2)
- New User Interface, Nmon Hosts Configuration Show Interface
- New User Interfaces, Nmon Hosts Inventory Interface for All systems and per OS type
- New Pivot Model to exploit Nmon Config data
- Purge script update

- Added Application setup confuguration to allow users activating NMON inputs at installation time
- Added access to Setup from navigation bar within the application
- migrated from full path references in default/inputs.conf to relative path due to incompatibility with setup.xml design (and REST endpoints update)
- Minor corrections of NMON Analyser pages
- Help page update

V1.1.3:

- Various corrections of views
- MEM views update with OS kind distinction
- Pivot Model updates to manage OS specific Metrics by OS type

V1.1.2:

- Dashboard “PieChart: TOP Hosts CPU and Memory Usage” Memory section correction

V1.1.1: Important update of NMON App which introduces distributed NMON Data collect and Real Time compatibility

- Indexers (or standalone instance) can now activate NMON local data collect upon installation (collect every 2 minutes in default config with 30 seconds data interval)
- A lightweight version of NMON App specifically designed for Splunk forwarders is available in “resources” directory, install it on forwarders and activate the input for your related OS to begin distributed NMON Data collect
- Custom span definition update: The macro is now much more accurate, generated charts give you the better of Splunk charting
- Real Time compatibility: Views can now do Real Time, thus with a limitation (for now) to a 12 hours window
- Important update of Documentation with Deployments scenarios
- Added Inline Help page available within the App
- Added scheduled purge of default NMON repository

V1.1.0:

- Major update of NMON App which introduces compatibility layer with AIX, Linux and Solaris OS Metrics
- New Home Page and navigation scheme between metrics and interfaces that have specific definitions and analysis depending on System type. (eg. NMON TOP sections for example will have different metrics available if you are analyzing an AIX, Linux or Solaris host)
- Global Metrics and Interfaces update for OS compatibility
- The “Global Analysis by host” interface has been renamed as “NMON Analyser” and exists in different versions depending on OS choice
- Processes System resources usage (known as TOP Section) NMON data is now converted with dynamic fields for OS compatibility, users with Linux or Solaris data already indexed should re-index these data
- Corrections on LPAR interface for AIX Virtualized Partitions

- Pivot Model update

V1.0.9:

- Various views corrections and improvements
- New Dashboard (django view) for Process Usage Analysis (NMON TOP Section)
- Span definition macro correction (no span value under certain circumstances)
- Home page margin correction for Firefox browser
- Calendar icon replacement
- Dashboards Django views corrections (empty fields with CPU % monitor)

V1.0.8:

- Icon gray theme changes
- Pivot Model corrections
- README update

V1.0.7:

- third party script corrections (blank lines in csv data generating streaming warn messages in splunkd, various corrections)
- Added support and views for File-Systems Metrics (JFSFILE, JFSINODE)
- Added Support and views for Linux Kernel Virtual Memory Statistics (VM)
- Pivot Model update

V1.0.6:

- Introducing NMON Pivot Data Models in very first versions

V1.0.5:

- Minor views update
- System App dj Page corrected for indexed data summary

V1.0.4:

- Solved NMON data conversion resulting in events duplication within Splunk, if you previously indexed data with anterior version, please delete index and restart Splunk, data will be re-indexed with no duplicates

V1.0.3:

- Minor corrections of various views
- TOP Process section analysis corrections

V1.0.2:

- Documentation update

V1.0.1:

- Home page correction

V1.0.0 beta:

- First Beta Release, V1.0.0 Beta

1.3 Known Issues

There are not currently known issues that you should be aware of.

Major or minor bug, enhancement requests will always be linked to an opened issue on the github project issue page:

https://github.com/guilhemmarchand/nmon-for-splunk/issues

Please note that once issues are considered as solved, by a new release or support exchanges, the issue will be closed. (but closed issues can still be reviewed)

1.4 Support

1.4.1 Octamis professional support for business

Nmon Performance is now available with professional support contract by Octamis limited.

Contact us at: sales@octamis.com

1.4.2 Community support

Nmon Performance Monitor for Splunk is provided in Open Source, you are totally free to use it for personal or professional use without any limitation, and you are free to modify sources or participate in the development if you wish.

This application and all of its components are provided under the Apache 2.0 licence, please remember that it comes with no warranty even if i intend to do my best in helping any people interested in using the App.

DISCLAIMER:

Unlike professional services, community support comes in “best effort” with absolutely no warranties.

Companies using this great piece are kindly invited to subscribe for a professional support contract to help us continuing developing the Nmon Performance solution!

Github

nmon-for-splunk (front-end application for search heads)

- https://github.com/guilhemmarchand/nmon-for-splunk

TA-nmon (Technology Addon)

- https://github.com/guilhemmarchand/TA-nmon

TA-nmon-hec (Technology Addon HEC version)

- https://github.com/guilhemmarchand/TA-nmon-hec

PA-nmon_light (Support Addon for indexers)

- https://github.com/guilhemmarchand/PA-nmon_light

Use Github to open an issue for errors and bugs to be reported, or to ask for enhancements requests.

You can even provide your own improvements by submitting a pull request.

Splunk Answers

Splunk has a strong community of active users and Splunk Answers is an important source of information.

Access previous messages of users or open your own discussion for the Nmon core application:

- http://answers.splunk.com/answers/app/1753

Google Group Support

An App dedicated Google Group has been created:

- https://groups.google.com/d/forum/nmon-splunk-app

This is also a great source of support from people using the Application, and you can also (if you subscribe to mailing news) receive important notifications about the App evolution, such as main release announcements.

1.5 Issues and enhancement requests

For any bug reporting, or enhancement request about the Nmon Performance application,you can:

Open a question on Splunk Answers:

- https://answers.splunk.com/app/questions/1753.html

Open an issue on the Git project home page:

- https://github.com/guilhemmarchand/nmon-for-splunk/issues

For any bug reporting, or enhancement request about the TA-nmon technical addon,you can:

Open a question on Splunk Answers:

- https://answers.splunk.com/app/questions/3248.html

Open an issue on the Git project home page:

- https://github.com/guilhemmarchand/TA-nmon/issues

email support:

- Get in touch by mail: guilhem.marchand@gmail.com

1.6 Scripts and Binaries

This depends on the stack component:

Component name	Contains scripts and binaries ?
nmon-for-splunk	No
PA-nmon_light	No
TA-nmon	Yes
TA-nmon-hec	Yes

1.6.1 nmon-for-splunk

The core front-end application does **NOT** contain any kind of script or binary.

1.6.2 PA-nmon_light

The Support Add-on does **NOT** contain any kind of script or binaries.

1.6.3 Technical Addons

The Technical Add-on contains various scripts and binaries:

- http://ta-nmon.readthedocs.io
- http://ta-nmon-hec.readthedocs.io

1.6.4 Additional tools

Customizer script

- resources/Nmon_SplunkApp_Customize.py.gz:

This Python script (must be uncompressed before execution) is a tool provided to execute different kind of automated customizations, such as restricting the application purpose to a given operating system for instance. (hide AIX and Solaris)

Detailed documentation: *Nmon_SplunkApp_Customize.py: Customize the Application*

https://github.com/guilhemmarchand/nmon-for-splunk/blob/master/nmon/resources/Nmon_SplunkApp_Customize.py.gz

Create agent script

- create_agent.py available in the Git repositories:

https://github.com/guilhemmarchand/TA-nmon

This Python script is a tool provided to create different version of the TA-nmon technical addon.

For example, you can use it to create a specific TA-nmon version for your critical production servers, and another version for your non production servers.

Each of the TA-nmon version would have its own parameters, such as the indexes, the data accuracy (interval between measures), etc.

Detailed documentation: *create_agent.py: Create multiple TA packages*

Study of usage: *01 - Splitting index for different users populations*

1.7 licence

CHAPTER 2

Documentation:

2.1 Introduction

NMON is short for Nigel's Performance Monitor, it is available on AIX Systems, Solaris (with Sarmon), and Linux Systems.

- nmon for Linux is open source under GPL: http://nmon.sourceforge.net
- nmon for AIX is not open source but is integrated into topas command from:
  - AIX 5.3 TL09
  - AIX 6.1 TL02

See: http://www.ibm.com/developerworks/wikis/display/WikiPtype/nmon

- nmon for Solaris (formerly sarmon) is open source and available for Solaris 10/11: http://www.geckotechnology.com/sarmon

This is a great "all in one" Performance Monitor tool that provides a very large amount of system performance information, and it can be used in different scenarios.

The first way to use NMON is running the "nmon" command in terminal, which opens a Real time monitoring terminal interface, giving you access to many system metrics within a single screen:

nmon is also very often used as a Capacity Planning and Performance tool by running the nmon process in a csv generating mode all along it's run time, for later cold Analyse.

Here are some useful links about NMON:

- http://nmon.sourceforge.net/pmwiki.php
- http://www.ibm.com/developerworks/aix/library/au-analyze_aix
- http://www-01.ibm.com/support/knowledgecenter/ssw_aix_71/com.ibm.aix.cmds4/nmon.htm
- http://www-01.ibm.com/support/knowledgecenter/ssw_aix_71/com.ibm.aix.cmds5/topas.htm

- http://nmon.sourceforge.net/pmwiki.php
- http://www.geckotechnology.com/fr/sarmon

Analysing NMON csv data is not easy because it has a very specific format Splunk cannot directly manage. (one big problem stands in the event time stamp identification which is very uncommon and defined by a non time stamp pattern)

This is why I decided to develop this App, based on my own professional experience in Unix systems Capacity Planning, to provide to anyone interested a powerful too to Analyse NMON data with an Enterprise Class Application.

2.1.1 How it works

In a few words, here is how the App works:

- The Nmon core application contains all the views, data models, configurations and related objects
- The TA-nmon which is the technical addon for the Nmon Performance application contains binaries and scripts to manage the nmon data
- The TA-nmon once installed starts immediately to collect and transforms nmon performance and configuration data
- The default configuration indexes data into the “nmon” index (by default)

2.1.2 Splunk Data structure

nmon_data

Performance metrics data ordered by the key “type” which corresponds to the nmon section metric item (CPU_ALL, LPAR…):

```
index=nmon sourcetype=nmon_data
```

Eventtype::

```
eventtype=nmon:performance
```

nmon_config

Configuration data extracted by nmon2csv converters, corresponds to AAA and BBB* sections of nmon raw data:

```
index=nmon sourcetype=nmon_config
```

Eventtype::

```
eventtype=nmon:config
```

nmon_collect

Output of the nmon_helper.sh script which is responsible for nmon instances launches:

```
index=nmon sourcetype=nmon_collect
```

Eventtype::

```
eventtype=nmon:collect
```

nmon_processing

Output of nmon2csv Python and Perl converters (conversion of nmon raw data into csv data):

```
index=nmon sourcetype=nmon_processing
```

Eventtype::

```
eventtype=nmon:collect
```

nmon_clean

Output of the nmon_cleaner.sh script (interface to nmon_helper.py | nmon_helper.pl) which is responsible for nmon raw data file cleaning:

```
index=nmon sourcetype=nmon_clean
```

Eventtype::

```
eventtype=nmon:clean
```

2.1.3 Available packages

There are different packages:

- The ***Nmon core*** Application: this is the whole package you download in Splunk App (directory called "nmon")
- The **PA-nmon_light** addon, available in the Git repository https://github.com/guilhemmarchand/PA-nmon_light (tgz archive), this package is expected to be deployed in indexers
- The **TA-nmon** addon, available in Splunk base https://splunkbase.splunk.com/app/3248 and https://github.com/guilhemmarchand/TA-nmon (tgz archive), can be deployed to any AIX / Linux / Solaris Full or Universal forwarder instance, master node of a cluster, deployment server, standalone instances, clustered indexers. . .

2.2 Deployment Matrix

2.2.1 What goes where ?

Application stack components

Component name	Purpose	Installation
nmon-for-splunk	Front-user core application	Search Heads
PA-nmon_light	Support Add-on for index-time configurations	Indexers / intermediate HF
TA-nmon	Technical Add-on for metrics and inventory generation	Each monitored server
TA-nmon-hec	Technical Add-on for metrics and inventory generation	Each monitored server

ONLY one technical addon must be deployed on the same host, **BUT** you can mix any both types of addons in your deployment.

Splunk Standalone deployment

A standalone Splunk installation means that all the Splunk roles are performed by the same instance, most likely for testing and development purposes.

Splunk roles	nmon-for-splunk	PA-nmon_light	TA-nmon-*
Standalone	X	X (optional)	X (optional)

The Technical Add-ons provide performance and configuration collection for the host than runs the add-on, which is optional

Distributed deployment

A Splunk distributed deployment is a Splunk infrastructure where specific Splunk roles are dedicated to specific instances.

For more information: http://docs.splunk.com/Documentation/Splunk/latest/Deploy/Indexercluster

The application stack is fully compatible with any kind of Splunk distributed deployment.

Splunk roles	nmon-for-splunk	PA-nmon_light	TA-nmon-*
Search head	X		X (optional)
Indexer		X	X (optional)
Master node			X (optional)
Deployment server			X (optional)
Heavy Forwarder		X (if TA is not installed)	X
Universal Forwarder			X

The Technical Add-on provides performance and configuration collection for the host than runs the add-on, which is optional

2.3 Deployment topologies

2.3.1 1. Splunk native deployment

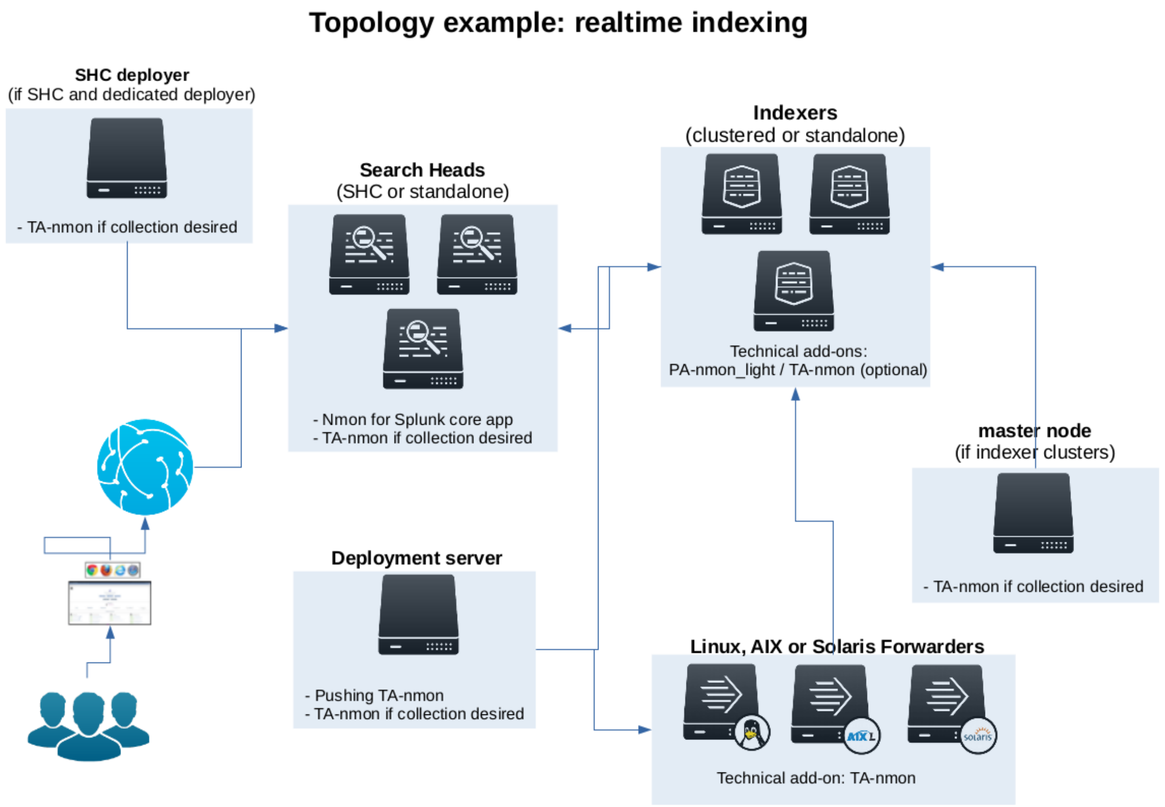
There are different ways to deploy and use the Nmon for Splunk App, basically the application works in 2 modes:

- Real Time: An nmon process runs on the server and generates performance and configuration for the host, Splunk retrieves, transforms and indexes the data using the TA-nmon
- Cold Mode: The Application is being used to manage collection of nmon raw files generated out of Splunk (nmon instances have terminated and files are closed)

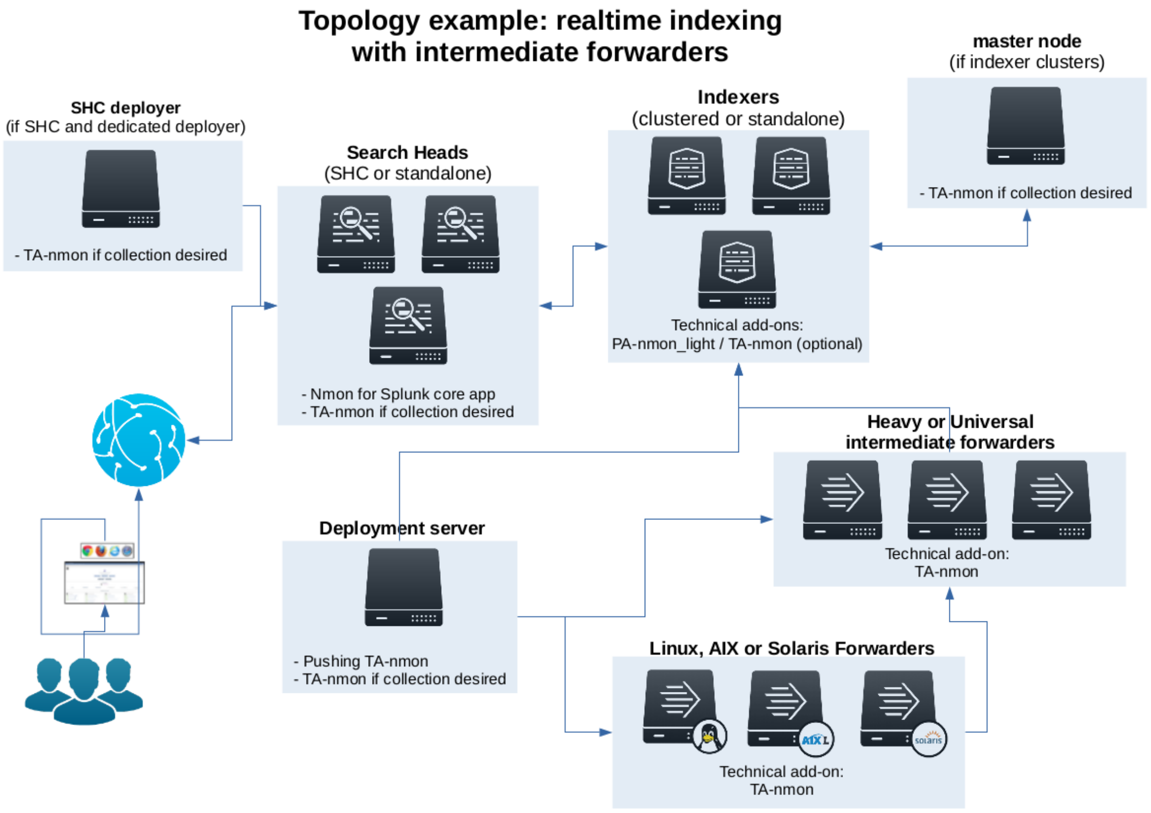
These deployment scenarios are detailed in the deployment section, as in an introduction, here are some deployment scenarios.

Real time deployment scenarios:

This is an example of deployment for standard scenario with multiple *nix clients running the addon "TA-nmon", managed trough a deployment server, indexing Nmon data into a Splunk indexer cluser running the addon "PA-nmon_light" and optionally the "TA-nmon", exploiting Nmon data in a Search Head cluster running the core "nmon" application and optionally the "TA-nmon" addon.

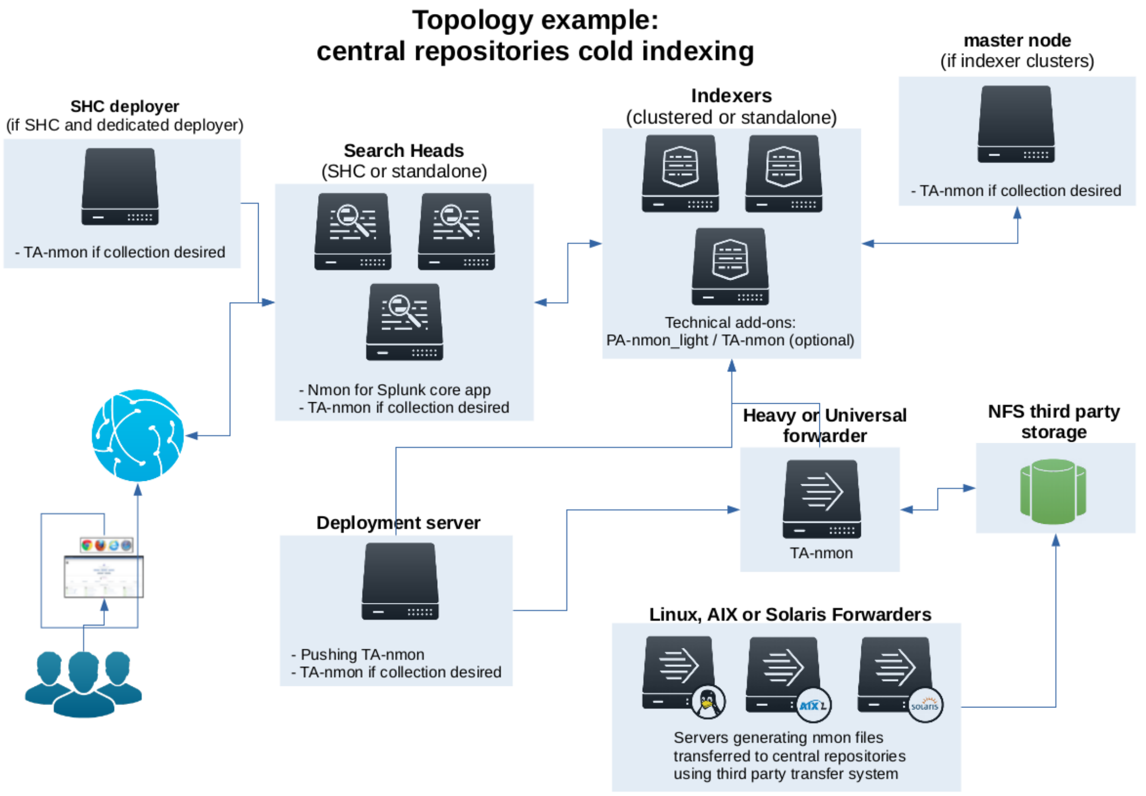

Topology example: realtime indexing
SHC deployer
(if SHC and dedicated deployer)
- TA-nmon if collection desired
Search Heads
(SHC or standalone)
- Nmon for Splunk core app
- TA-nmon if collection desired
Indexers
(clustered or standalone)
Technical add-ons:
PA-nmon_light / TA-nmon (optional)
master node
(if indexer clusters)
- TA-nmon if collection desired
Deployment server
- Pushing TA-nmon
- TA-nmon if collection desired
Linux, AIX or Solaris Forwarders
AIX
solaris
Technical add-on: TA-nmon

This is an example of deployment for standard scenario with multiple *nix clients running the addon "TA-nmon" and sending data to intermediate Heavy or Universal Forwarders, all managed trough a deployment server, indexing Nmon data into a Splunk indexer cluser running the addon "PA-nmon_light" and optionally the "TA-nmon", exploiting Nmon data in a Search Head cluster running the core "nmon" application and optionally the "TA-nmon" addon.


Topology example: realtime indexing
with intermediate forwarders
master node
(if indexer clusters)
- TA-nmon if collection desired
SHC deployer
(if SHC and dedicated deployer)
- TA-nmon if collection desired
Indexers
(clustered or standalone)
Technical add-ons:
PA-nmon_light / TA-nmon (optional)
Search Heads
(SHC or standalone)
- Nmon for Splunk core app
- TA-nmon if collection desired
Heavy or Universal
intermediate forwarders
Technical add-on:
TA-nmon
Deployment server
- Pushing TA-nmon
- TA-nmon if collection desired
Linux, AIX or Solaris Forwarders
Technical add-on:
TA-nmon

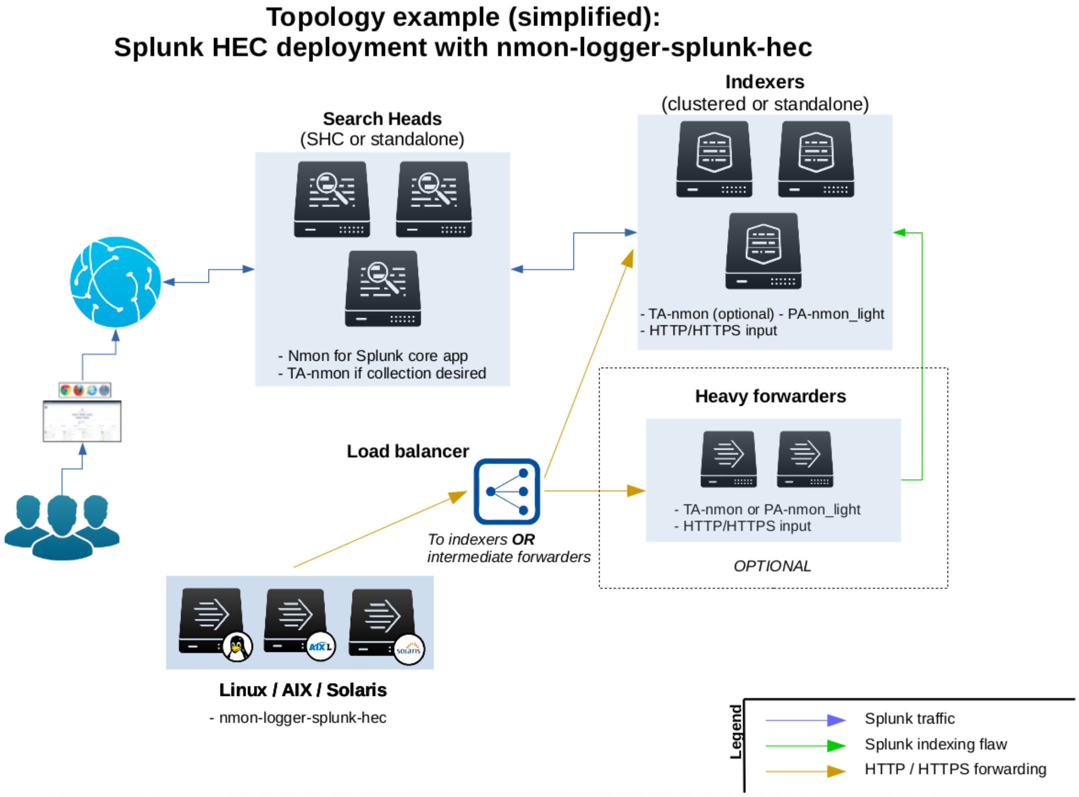
Cold data deployment scenario:

This is an example of deployment for standard scenario with a single Splunk forwarder instance (Universal or Heavy) running the addon “TA-nmon”, indexing Nmon data from central NFS repositories into a Splunk indexer cluser running the addon “PA-nmon_light” and optionally the “TA-nmon”, exploiting Nmon data in a Search Head cluster running the core “nmon” application and optionally the TA-nmon addon.


Topology example:
central repositories cold indexing
SHC deployer
(if SHC and dedicated deployer)
- TA-nmon if collection desired
Search Heads
(SHC or standalone)
- Nmon for Splunk core app
- TA-nmon if collection desired
Indexers
(clustered or standalone)
Technical add-ons:
PA-nmon_light / TA-nmon (optional)
master node
(if indexer clusters)
- TA-nmon if collection desired
Heavy or Universal
forwarder
TA-nmon
NFS third party
storage
Deployment server
- Pushing TA-nmon
- TA-nmon if collection desired
Linux, AIX or Solaris Forwarders
Servers generating nmon files
transferred to central repositories
using third party transfer system

2.3.2 2. Agentless deployment with Splunk HEC and nmon-logger

Since the version 1.9.10, the nmon-logger for Splunk HEC provides a 100% agent less configuration using the Splunk http input:



This deployment provides the following features:

- **clients easy set up:** the nmon-logger is provided as deb/rpm package, easy and fast deployment
- **server easy set up:** Splunk http input is easy to configure and implement
- **100% agent less:** the nmon-logger uses only native system features (cron, logrotate. . .)
- **secure:** Splunk http traffic can easily be encrypted via SSL and integrated into any DMZ or similar restricted networking layer
- **resilient and scalable:** using load balancers and multiple nodes provides resiliency and horizontal scalability
- **network friendly:** as Web service, it can be easily used across wide networks and over the Internet
- **easy management:** since the http input is managed on a token basis, you can easily configure different token to ingest the data into different indexes without any package modification or complexity

See the detailed section: *Splunk HEC / nmon-logger deployment*

2.3.3 3. Syslog deployment

Additionally and since the version 1.6.14, it is possible to use Syslog as the transport layer associated with a third party package called "nmon-logger"

This deployment topology provides all the application features without any deployment of Universal Forwarders on end servers, using rsyslog or syslog-ng. Such a deployment answers the need for people that cannot or do not want to install any third party agent.

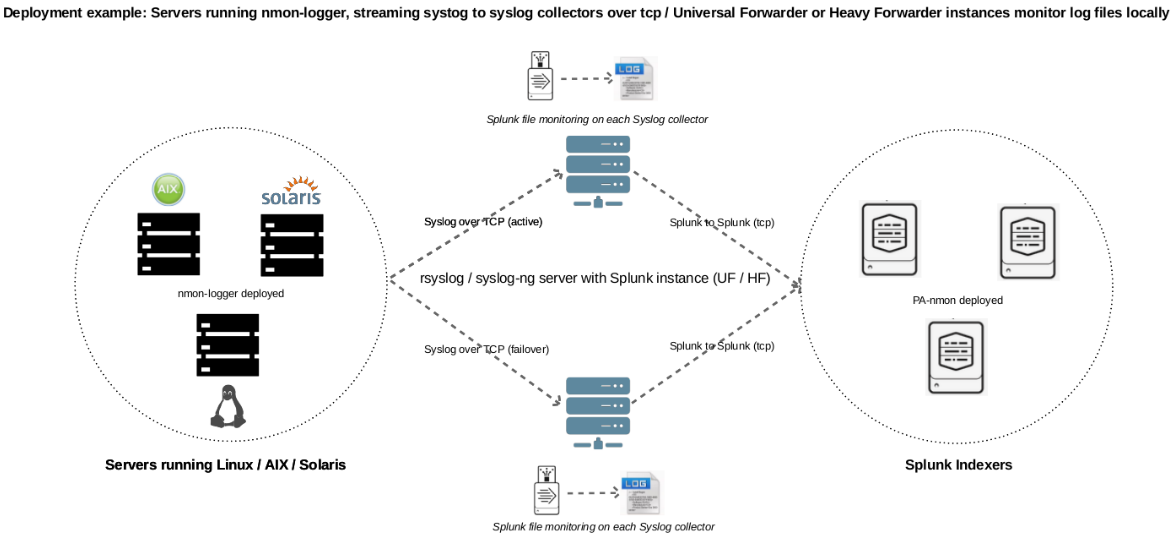
The nmon-logger package is available for download in GitHub: https://github.com/guilhemmarchand/nmon-logger

The deployment will use and require and/or recommended:

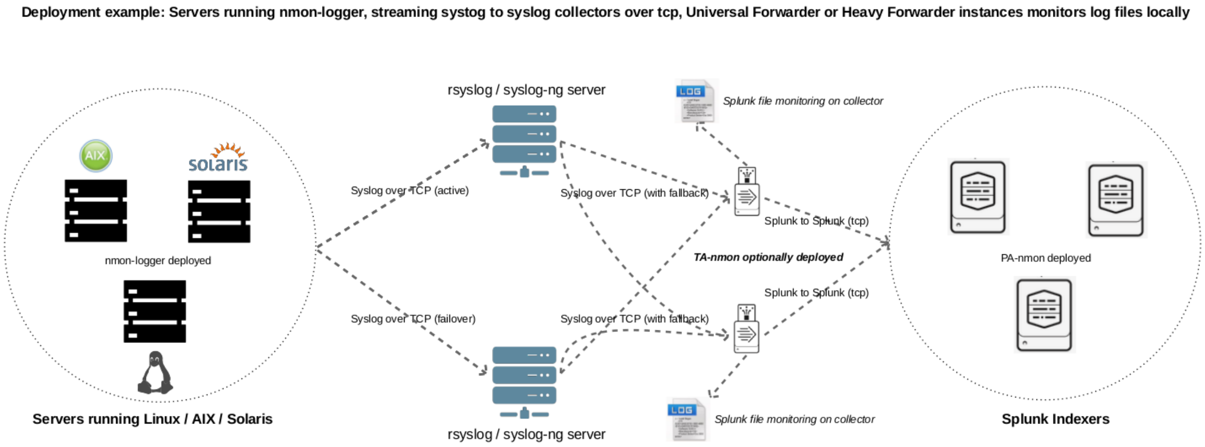
- nmon-logger
- rsyslog or syslog-ng
- cron
- logrotate

Note that this deployment scenario would be recommended mostly with a modern Linux deployment. Although all pieces of software should work fine too on AIX and Solaris, this requires quite up to date versions (for rsyslog / syslog-ng), which could be complex on older OS.

Example 1: Splunk Universal or Heavy forwarder installed on main syslog-ng collectors:


Deployment example: Servers running nmon-logger, streaming systog to syslog collectors over tcp / Universal Forwarder or Heavy Forwarder instances monitor log files locally
Splunk file monitoring on each Syslog collector
AIX
solaris
nmon-logger deployed
Servers running Linux / AIX / Solaris
Syslog over TCP (active)
Syslog over TCP (failover)
rsyslog / syslog-ng server with Splunk instance (UF / HF)
Splunk to Splunk (tcp)
Splunk to Splunk (tcp)
PA-nmon deployed
Splunk Indexers
Splunk file monitoring on each Syslog collector

Example 2: Splunk Universal or Heavy forwarder installed third party servers running syslog-ng:


Deployment example: Servers running nmon-logger, streaming systog to syslog collectors over tcp, Universal Forwarder or Heavy Forwarder instances monitors log files locally
rsyslog / syslog-ng server
Splunk file monitoring on collector
AIX
solaris
nmon-logger deployed
Servers running Linux / AIX / Solaris
Syslog over TCP (active)
Syslog over TCP (with fallback)
Splunk to Splunk (tcp)
TA-nmon optionally deployed
PA-nmon deployed
Splunk to Splunk (tcp)
Syslog over TCP (failover)
Syslog over TCP (with fallback)
Splunk file monitoring on collector
Splunk Indexers
rsyslog / syslog-ng server

For rsyslog, see: *rsyslog / nmon-logger deployment*

For syslog-ng, see: *syslog-ng / nmon-logger deployment*

2.4 Download

2.4.1 nmon-for-splunk (front-end application for search heads)

Download with Splunk Base at the following URL:

- https://splunkbase.splunk.com/app/1753

2.4.2 TA-nmon (Technology Addon)

Download with Splunk Base at the following URL:

- https://splunkbase.splunk.com/app/3248

2.4.3 TA-nmon-hec (Technology Addon HEC version)

Download with Splunk Base at the following URL:

- https://splunkbase.splunk.com/app/3668

2.4.4 PA-nmon_light (Support Addon for indexers)

Download with Splunk Base at the following URL:

- https://splunkbase.splunk.com/app/4067

2.4.5 Github releases

The Nmon Performance Monitor is hosted on a Github project, you can freely download the application from the Github project page: https://github.com/guilhemmarchand/nmon-for-splunk

The TA-nmon has also its own Github project: https://github.com/guilhemmarchand/TA-nmon

About main branches and associated versions:

Github branch	master	release	testing
Stable and eligible for Splunk Base publication	X		
Pre-release under qualification cycle, can break things		X	
Unstable and under heaby testing, can break things			X

Downloading and installing from Github:

nmon-for-splunk (front-end application for search heads)

Download the latest tgz archive at the root of the repository:

- https://github.com/guilhemmarchand/nmon-for-splunk

TA-nmon (Technology Addon)

Download the latest tgz archive at the root of the repository:

- https://github.com/guilhemmarchand/TA-nmon

TA-nmon-hec (Technology Addon HEC version)

Download the latest tgz archive at the root of the repository:

- https://github.com/guilhemmarchand/TA-nmon-hec

PA-nmon_light (Support Addon for indexers)

Download the latest tgz archive at the root of the repository:

- https://github.com/guilhemmarchand/PA-nmon_light

2.4.6 Stopped releases for old Splunk versions

As the Nmon Performance Monitor attempts to get the better from Splunk new features, it is possible that new releases will stop being compatible with old Splunk versions.

Currently, here are stopped versions for older Splunk releases:

- Last version compatible with Splunk 6.1.x, with release 1.4.902 (not Splunk certified)

https://github.com/guilhemmarchand/nmon-for-splunk/blob/last_release_splunk_61x

- Last version compatible with Splunk 6.2.x with release 1.6.15 (Splunk certified)

https://github.com/guilhemmarchand/nmon-for-splunk/releases

- Last version compatible with Splunk 6.4.x and Splunk 6.3.x with release 1.7.9 (Splunk certified)

https://github.com/guilhemmarchand/nmon-for-splunk/releases

2.5 Running on Windows

It is NOT possible to generate Nmon data of a Windows machine!

But you can install and run the applications on Windows for different purposes:

Splunk roles	nmon-for-splunk	PA-nmon_light	TA-nmon
Search head	X		
Indexer		X	
Master node			
Deployment server		Conditional	Conditional
Heavy Forwarder		Conditional	
Universal Forwarder			X

- Using Windows as a deployment server to push applications to Unix/Linux based servers is strongly discouraged as required file permissions will be lost, and manual actions would be required

2.6 Deploy to single server instance

2.6.1 Installation for standalone instance

Standalone deployment: A single Splunk instance performs all roles

Splunk roles	nmon-for-splunk	PA-nmon_light	TA-nmon-*
Standalone	X	X (optional)	X (optional)

optional: The Technical Add-on provide nmon performance and configuration collection for the host than runs the add-on, which is optional

If you are new to Splunk, checkout:

https://docs.splunk.com/Documentation/Add-ons/released/Overview/Singleserverinstall

VIDEO TUTORIAL

Checkout this video demo: https://www.youtube.com/watch?v=-0H-CJDIGDI

Installing with Splunk Web

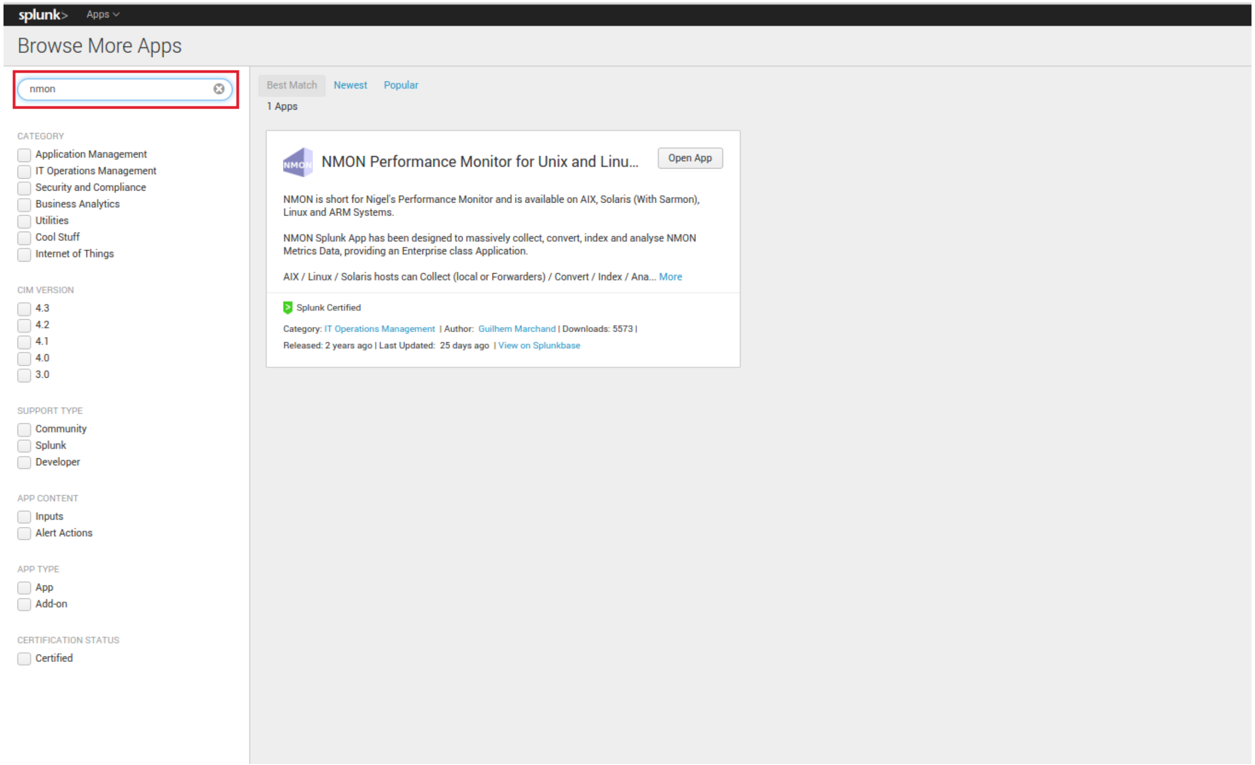
You can install the Application directly within Splunk Application management:

1. Access the Application manager:

Application Menu > Manage Apps

2. Browse online for Nmon Performance Monitor App and follow Splunk standard app installation:

Search for "nmon":



NMON Performance Monitor for Unix and Linu...

NMON is short for Nigel's Performance Monitor and is available on AIX, Solaris (With Sarmon), Linux and ARM Systems.

NMON Splunk App has been designed to massively collect, convert, index and analyse NMON Metrics Data, providing an Enterprise class Application.

AIX / Linux / Solaris hosts can Collect (local or Forwarders) / Convert / Index / Ana... More

Splunk Certified

Category: IT Operations Management | Author: Guilhem Marchand | Downloads: 5573 |

Released: 2 years ago | Last Updated: 25 days ago | View on Splunkbase

Manual installation

1. Download the tgz archive of Nmon Performance in Splunk Base:

https://splunkbase.splunk.com/app/1753

See the Download page for more information and options

2. Unarchive

To install the application, simply unarachive the tgz file in the apps directory of Splunk, example:

```
cd /opt/splunk/etc/apps/
tar -xvf nmon-performance-monitor-for-unix-and-linux-systems_*.tgz
```

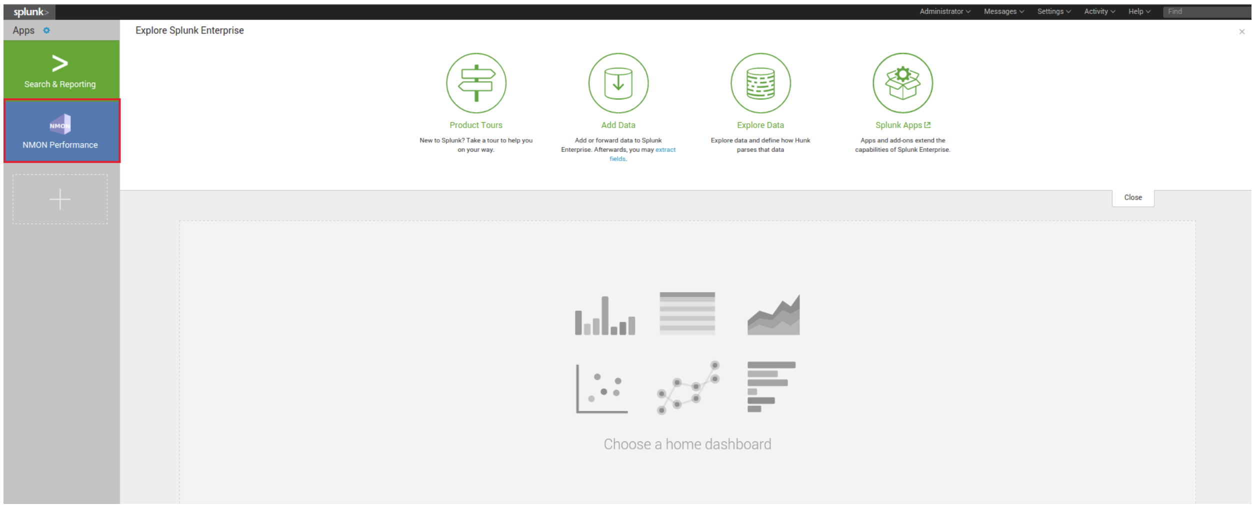
3. And restart Splunk

```
/opt/splunk/bin/splunk restart
```

Access the Application

After Splunk restart, you can directly access the application:

through its App icon:



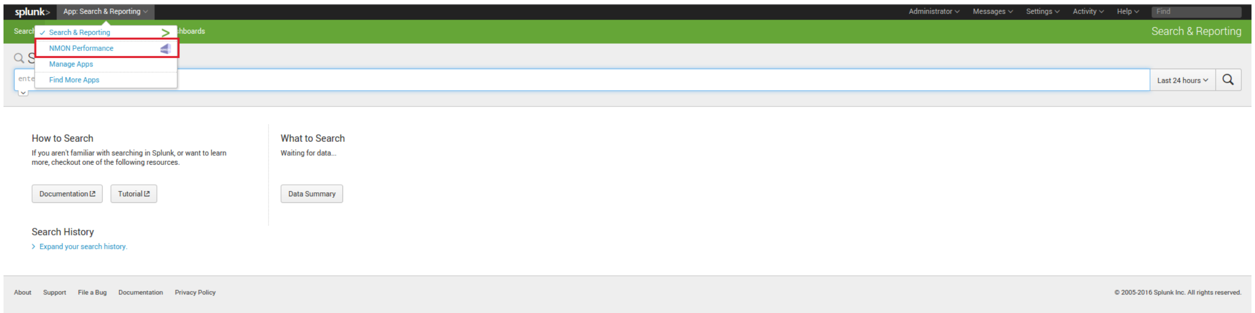
New to Splunk? Take a tour to help you on your way.

Add or forward data to Splunk Enterprise. Afterwards, you may extract fields.

Explore data and define how Hunk parses that data

Apps and add-ons extend the capabilities of Splunk Enterprise.

Or the app menu bar:



How to Search

If you aren't familiar with searching in Splunk, or want to learn more, checkout one of the following resources.

What to Search

Waiting for data...

Search History

Expand your search history.

© 2005-2016 Splunk Inc. All rights reserved.

Generating performance and configuration data

If you are running Splunk on **Linux, AIX or Solaris**, then you can generate Nmon performance data for the local machine running Splunk.

Replace the PA-nmon_light_XXXX.tgz with current release of the PA-nmon_light:

```
cd /opt/splunk/etc/apps/
tar -xvf <YOUR PATH>/PA-nmon_light_XXXX.tgz
```

And restart Splunk:

```
/opt/splunk/bin/splunk restart
```

Replace the TA-nmon_XXXX.tgz with current release of the TA-nmon:

```
cd /opt/splunk/etc/apps/
tar -xvf <YOUR PATH>/TA-nmon_XXXX.tgz
```

And restart Splunk:

```
/opt/splunk/bin/splunk restart
```

2.7 Deploy to distributed deployment

2.7.1 Installation for distributed deployments

Distributed deployment matrix:

Software components:

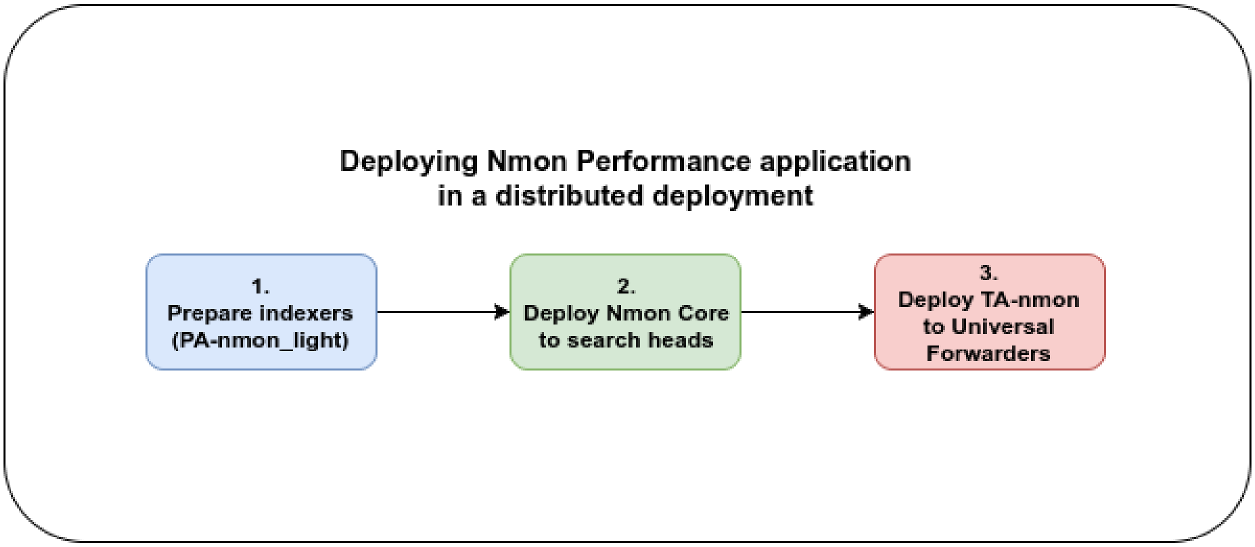
Splunk roles	nmon-for-splunk	PA-nmon_light	TA-nmon-*
Search head	X		X (optional)
Indexer		X	X (optional)
Master node			X (optional)
Deployment server		Conditional	Conditional
Heavy Forwarder		Conditional	Conditional
Universal Forwarder			X

The Technology Add-ons provide metrics and configuration collection for the host than runs the add-on, which is optional.

The Support Add-on does not generate any collection, but defines indexes and contains index time configuration settings.

The following installation tutorial covers all aspects of a distributed deployment scenario:

- Standalone indexers
- Single site or multi site indexer clusters
- Standalone search heads
- Search heads in a sh cluster



1. Deploying the PA-nmon_light and TA-nmon (optional) on indexers

1.1. Deploying on clustered indexers

We will assume your indexers are already operational, in the case of a new installation, remember to activate port receiving to allow the indexer to retrieve data.

If required (eg. new installation), this can be easily achieved:

in CLI:

```
/opt/splunk/bin/splunk enable listen 9997
Where 9997 (default) will be the receiving port for Splunk Forwarder connections
```

Deploying the PA-nmon_light on clustered indexers

ALL THESE ACTION MUST BE DONE ON THE MASTER NODE

Remind:

- If you don't want to collect performance and configuration data from your indexers, deploy only the PA-nmon_light
- If you want to collect performance and configuration data from your indexers, deploy both the PA-nmon_light and TA-nmon

Download the Application tar.gz archive from:

https://splunkbase.splunk.com/app/1753/

Extract the content of the archive on your master node in a temporary directory, example:

```
cd /tmp/
<upload the archive here>

tar -xvzf nmon-performance-monitor-for-unix-and-linux-systems*.tgz
```

TA-nmon: (optional)

The TA-nmon tgz archive must be uncompressed and installed in the Master Node in $SPLUNK_HOME/etc/master_apps/ (where $SPLUNK_HOME refers to the root directory of your Splunk installation)

```
cd /opt/splunk/etc/master/apps

tar -xvzf <YOUR PATH>/TA-nmon_*.tar.gz
```

PA-nmon_light:

The PA-nmon_light must be uncompressed and installed in the Master Node in $SPLUNK_HOME/etc/master_apps/ (where $SPLUNK_HOME refers to the root directory of your Splunk installation)

```
cd /opt/splunk/etc/master/apps

tar -xvzf >YOUR PATH>/PA-nmon_light_*.tar.gz
```

Publish the cluster bundle to indexers, this implies an automatic rolling restart of indexers:

```
/opt/splunk/bin/splunk apply cluster-bundle
```

To see the current status of the indexer cluster:

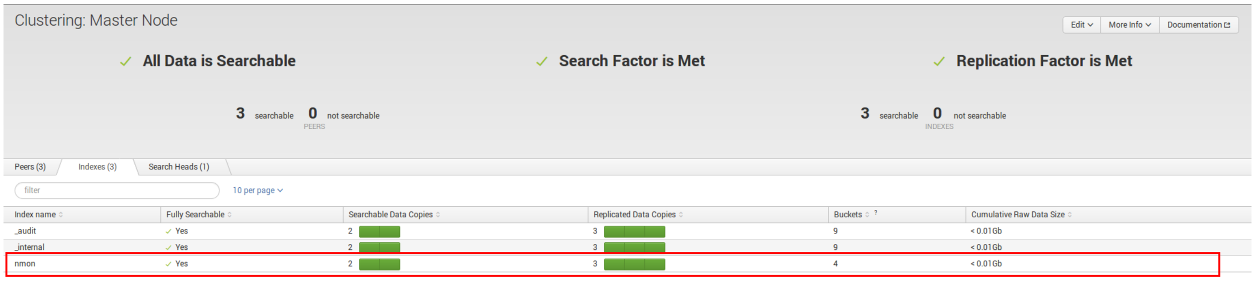
In CLI:

```
/opt/splunk/bin/splunk show cluster-bundle-status
```

In Splunk Web, connected to the master node console:

Settings –> Indexer Clustering

Upon Rolling Restart of the indexer cluster, and if the local data performance collecting is activated, a new clustered index will be available in the indexer clustering console from the Master node:



Clustering: Master Node

Index name	Fully Searchable	Searchable Data Copies	Replicated Data Copies	Buckets	Cumulative Raw Data Size
_audit	✓ Yes	2	3	9	< 0.01Gb
_internal	✓ Yes	2	3	9	< 0.01Gb
nmon	✓ Yes	2	3	4	< 0.01Gb

1.2. Deploying the PA-nmon_light and TA-nmon (optional) on standalone indexers

ALL THESE ACTION MUST BE DONE FOR EACH STANDALONE INDEXER

Download the Application tar.gz archive from:

https://splunkbase.splunk.com/app/1753/

Extract the content of the archive on your indexer in a temporary directory, example:

```
cd /tmp/

<upload the archive here>

tar -xvzf nmon-performance-monitor-for-unix-and-linux-systems*.tgz
```

Remind:

- If you don't want to collect performance and configuration data from your indexers, deploy only the PA-nmon_light
- If you want to collect performance and configuration data from your indexers, deploy both the PA-nmon_light and TA-nmon

TA-nmon: (optional)

The TA-nmon tgz archive must be uncompressed and installed in the Master Node in $SPLUNK_HOME/etc/master_apps/ (where $SPLUNK_HOME refers to the root directory of your Splunk installation)

```
cd /opt/splunk/etc/apps

tar -xvzf <YOUR PATH>/TA-nmon_*.tar.gz
```

PA-nmon_light:

The PA-nmon_light tgz archive must be uncompressed and installed in the Master Node in $SPLUNK_HOME/etc/master_apps/ (where $SPLUNK_HOME refers to the root directory of your Splunk installation)

```
cd /opt/splunk/etc/apps

tar -xvzf <YOUR PATH>/PA-nmon_light_*.tar.gz
```

Restart the indexer:

```
splunk restart
```

2. Deploying the Core App and TA-nmon (optional) to search heads

2.1. Deploying the Nmon Core in a sh cluster

ALL THESE ACTION MUST BE DONE ON THE SHC DEPLOYER

Download the Application tar.gz archive from:

https://splunkbase.splunk.com/app/1753/

Upload the archive to the search head in a temporary directory, example:

```
cd /tmp/

<upload archive here>
```

NOTE: For more information about search head clustering and application deployment, see:

http://docs.splunk.com/Documentation/Splunk/latest/DistSearch/PropagateSHCconfigurationchanges

On the SHC deployer, the configuration bundle resides under the $SPLUNK_HOME/etc/shcluster directory. The set of files under that directory constitutes the configuration bundle.

The directory has this structure:

```
$SPLUNK_HOME/etc/shcluster/
     apps/
          <app-name>/
          <app-name>/
          ...
     users/
```

Extract the content of the core Application (the tar archive you downloaded from Splunk base) to the "apps" directory.

The core application does not generate nmon data, if you want to get performance and configuration data from your search heads, extract the content of the TA-nmon addon to the "apps" directory.

```
cd /opt/splunk/etc/shcluster/apps/
tar -xvf <YOUR PATH>/nmon-performance-monitor-for-unix-and-linux-systems_*.tgz
tar -xvf <YOUR PATH>/TA-nmon*.tgz
```

Finally push the configuration bundle to publish the Nmon core application to all search heads:

```
splunk apply shcluster-bundle -target <URI>:<management_port> -auth <username>:
↪<password>
```

2.2. Deploying the Nmon Core in standalone search heads

For each search head:

Download the Application tar.gz archive from:

https://splunkbase.splunk.com/app/1753/

Upload the archive to the search head in a temporary directory, example:

```
cd /tmp/

<upload archive here>
```

Uncompress the content of the tar.gz archive in $SPLUNK_HOME/etc/apps/ (where $SPLUNK_HOME refers to the Application root directory)

```
tar -xvzf nmon-performance-monitor-for-unix-and-linux-systems*.tgz
```

Since the release V1.7, the core application does not generate anymore nmon data, if you want to get performance and configuration data from your search heads, extract the content of the TA-nmon addon to the "apps" directory.

```
cd /opt/splunk/etc/apps/
tar -xvf <YOUR PATH>/TA-nmon*.tgz
```

Restart each search head manually:

```
splunk restart
```

3. Deploying the TA-nmon to Heavy or Universal Forwarders

The next step is to deploy the TA-nmon in every machine that must be monitored.

The following tutorial assumes that you will be using the Splunk deployment server to publish the TA-nmon package to clients.

However, any other automation solution (Ansible, Chef, Puppet. . .) can be used with no issue.

3.1 Preparing the TA-nmon on deployment servers

ALL THESE ACTION MUST BE DONE ON INSTANCE(S) ACTING AS DEPLOYMENT SERVERS

Download the Application tar.gz archive from:

https://splunkbase.splunk.com/app/1753/

Extract the content of the archive on your indexer in a temporary directory, example:

```
cd /tmp/

<upload the archive here>

tar -xvzf nmon-performance-monitor-for-unix-and-linux-systems*.tgz
```

The TA-nmon tgz archive must be uncompressed and installed in $SPLUNK_HOME/etc/deployment-apps/ (where $SPLUNK_HOME refers to the root directory of Splunk installation)

```
cd /opt/splunk/etc/deployment-apps

tar -xvzf <YOUR PATH>/TA-nmon_*.tar.gz
```

Then , ask the deployment server to update its configuration:

```
/opt/splunk/bin/splunk reload deploy-server
```

3.2. Configuring the deployment server to push the TA-nmon to Universal Forwarders

Connecting Universal Forwarders to the Deployment Server:

If this is a new installation or if you haven't already, you must connect your Universal Forwarders clients to your deployment server:

in CLI:

```
/opt/splunkforwarder/bin/splunk set-poll <mydeploymentserver>:8089
```

Where <mydeploymentserver> corresponds to the hostname of your Deployment Server

For more information, see:

http://docs.splunk.com/Documentation/Splunk/latest/Forwarding/Setupforwardingandreceiving

Deploying forwarding configuration (outputs.conf) to Universal Forwarders clients:

Most of the time in an existing deployment of Universal Forwarders, you will probably want to host the copy of the configuration "outputs.conf" in a dedicated configuration (eg. application) being pushed to all connected clients.

For more information, see:

http://docs.splunk.com/Documentation/Splunk/latest/Forwarding/Setupforwardingandreceiving

You can also host the outputs.conf within the TA-nmon package, such as:

```
cd /opt/splunk/etc/deployment-apps/TA-nmon

mkdir local

<create outputs.conf and set the list of indexers and desired options, example>

[tcpout]
defaultGroup = indexer_pool

[tcpout:indexer_pool]
server=splunk-peer1:9997,splunk-peer2:9997,splunk-peer3:9997
autoLB = true
```

The next step will reside in configuring the Deployment Server to push the TA-nmon to connected clients, by creating the associated server class and linked application

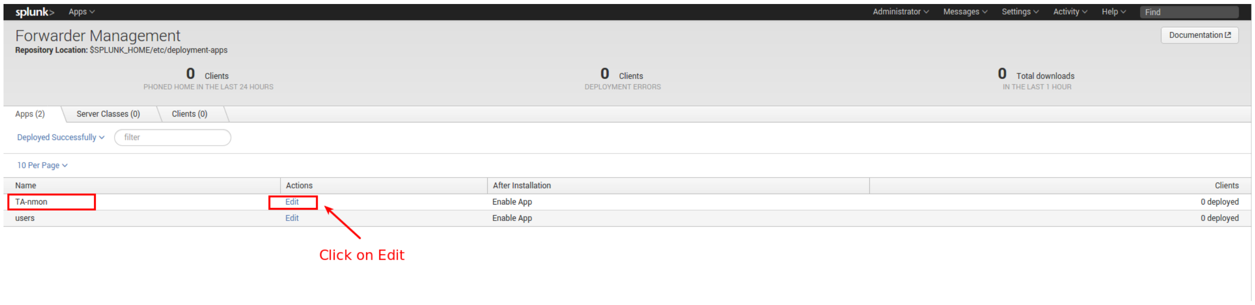
ALL THESE ACTION MUST BE DONE ON INSTANCE(S) ACTING AS DEPLOYMENT SERVERS IN SPLUNK WEB

Connect to Splunk Web of your Deployment Server

Access the Forwarder Management Interface (Settings —> Forwarder Management)

Follow these steps examples to set up a server class that will push to clients the TA-nmon package

Edit the TA-nmon application:



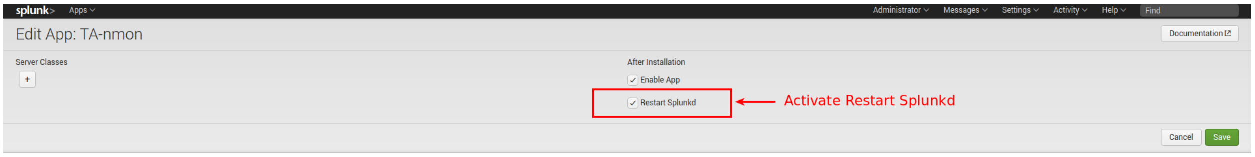
Forwarder Management
Repository Location: $SPLUNK_HOME/etc/deployment-apps

0 Clients PHONED HOME IN THE LAST 24 HOURS
0 Clients DEPLOYMENT ERRORS
0 Total downloads IN THE LAST 1 HOUR

Name	Actions	After Installation	Clients
TA-nmon	Edit	Enable App	0 deployed
users	Edit	Enable App	0 deployed

Click on Edit

*Ensure to activate “restart splunkd”:



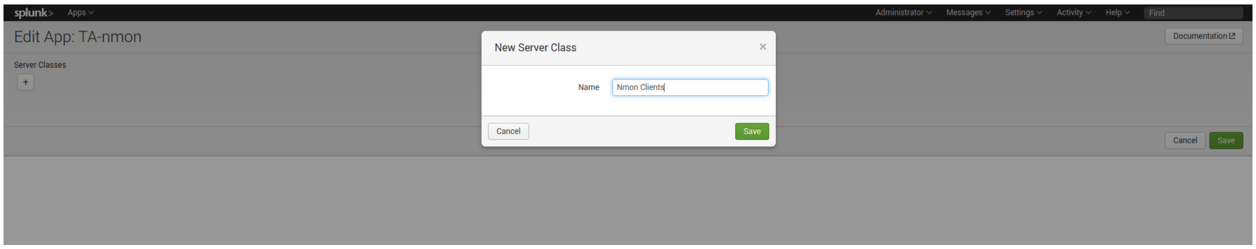
Edit App: TA-nmon

Server Classes

After Installation
☑ Enable App
☑ Restart Splunkd

Activate Restart Splunkd

Create a new server class:

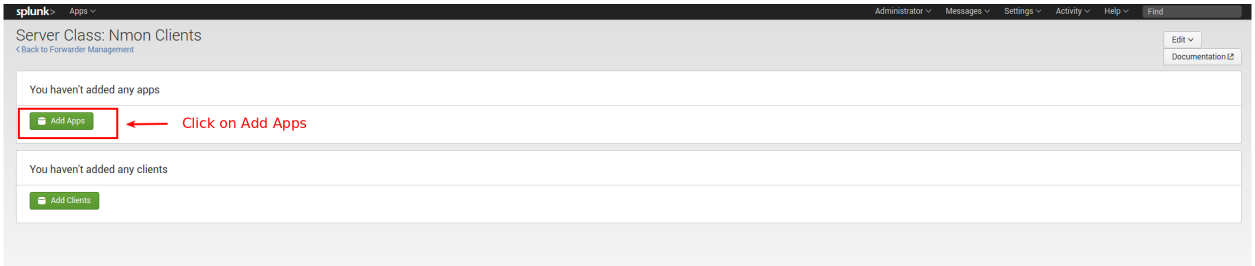


Edit App: TA-nmon

New Server Class

Name: Nmon Clients

Associate the TA-nmon with the server class:

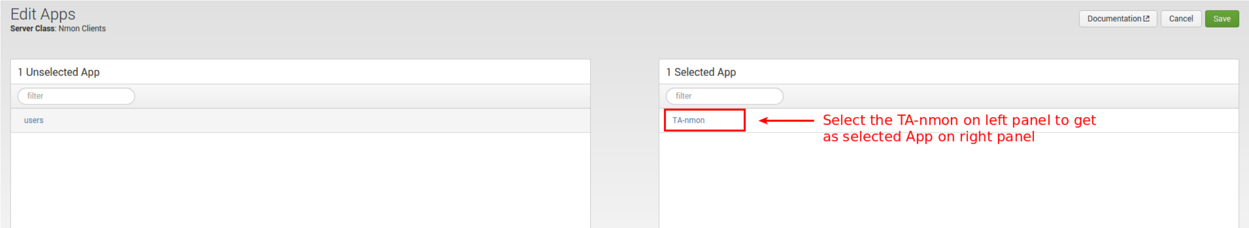


Server Class: Nmon Clients
< Back to Forwarder Management

You haven't added any apps

Click on Add Apps

You haven't added any clients



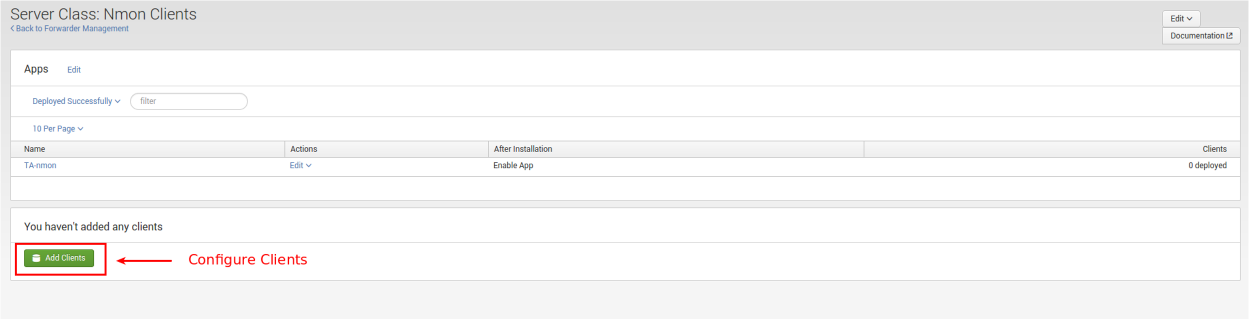
Edit Apps
Server Class: Nmon Clients

1 Unselected App
users

1 Selected App
TA-nmon

Select the TA-nmon on left panel to get as selected App on right panel

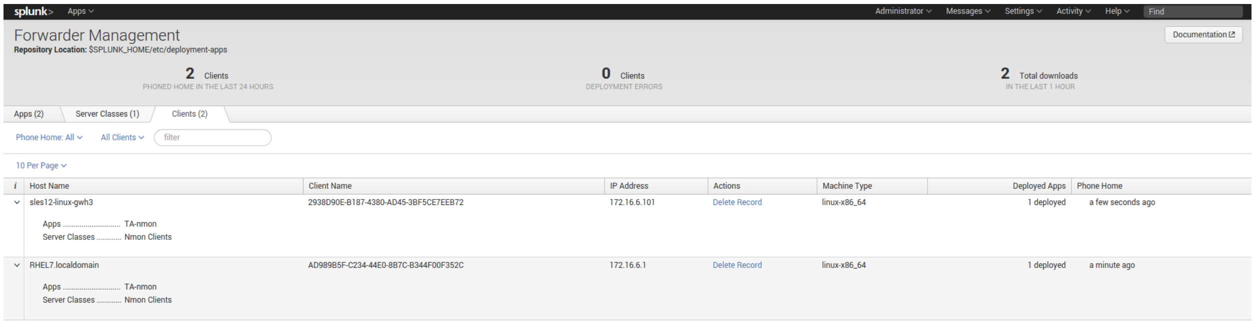
And finally add required clients:



Server Class: Nmon Clients

Name	Actions	After Installation	Clients
TA-nmon	Edit	Enable App	0 deployed

You haven't added any clients



Forwarder Management

Repository Location: $SPLUNK_HOME/etc/deployment-apps

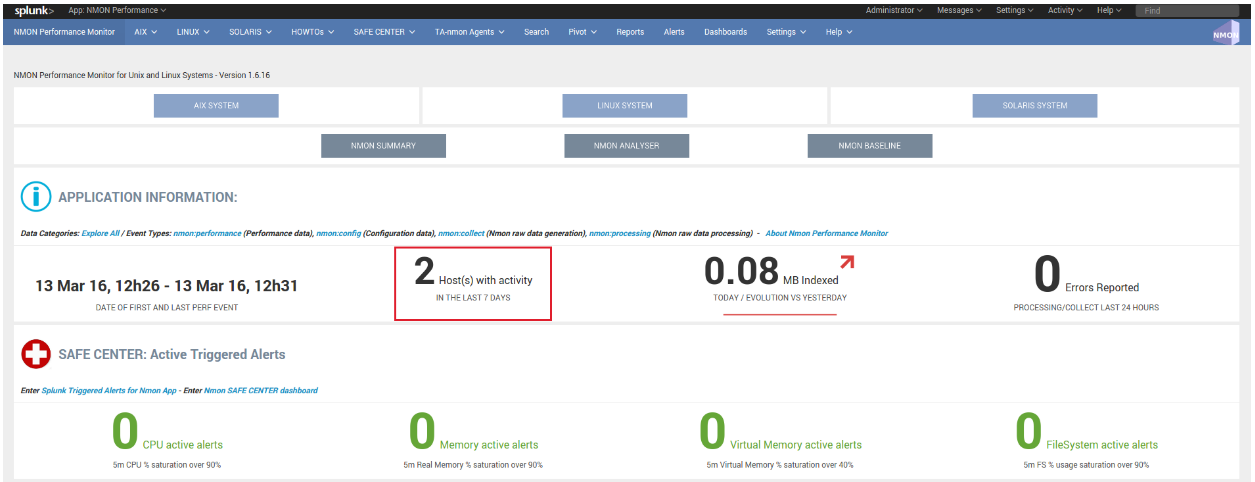
2 Clients PHONED HOME IN THE LAST 24 HOURS

0 Clients DEPLOYMENT ERRORS

2 Total downloads IN THE LAST 1 HOUR

i	Host Name	Client Name	IP Address	Actions	Machine Type	Deployed Apps	Phone Home
	sles12-linux-gwh3 Apps .... TA-nmon Server Classes .... Nmon Clients	2938D90E-B187-4380-AD45-3BF5CE7EEB72	172.16.6.101	Delete Record	linux-x86_64	1 deployed	a few seconds ago
	RHEL7.localdomain Apps .... TA-nmon Server Classes .... Nmon Clients	AD989B5F-C234-44E0-8B7C-B344F00F352C	172.16.6.1	Delete Record	linux-x86_64	1 deployed	a minute ago

A few minutes later, you will start collecting data from your hosts, refresh the Application Home page and you should see the number of hosts in activity being increased:



NMON Performance Monitor for Unix and Linux Systems - Version 1.6.16

APPLICATION INFORMATION:

Data Categories: Explore All / Event Types: nmon:performance (Performance data), nmon:config (Configuration data), nmon:collect (Nmon raw data generation), nmon:processing (Nmon raw data processing) - About Nmon Performance Monitor

13 Mar 16, 12h26 - 13 Mar 16, 12h31 — DATE OF FIRST AND LAST PERF EVENT

2 Host(s) with activity — IN THE LAST 7 DAYS

0.08 MB Indexed — TODAY / EVOLUTION VS YESTERDAY

0 Errors Reported — PROCESSING/COLLECT LAST 24 HOURS

SAFE CENTER: Active Triggered Alerts

Enter Splunk Triggered Alerts for Nmon App - Enter Nmon SAFE CENTER dashboard

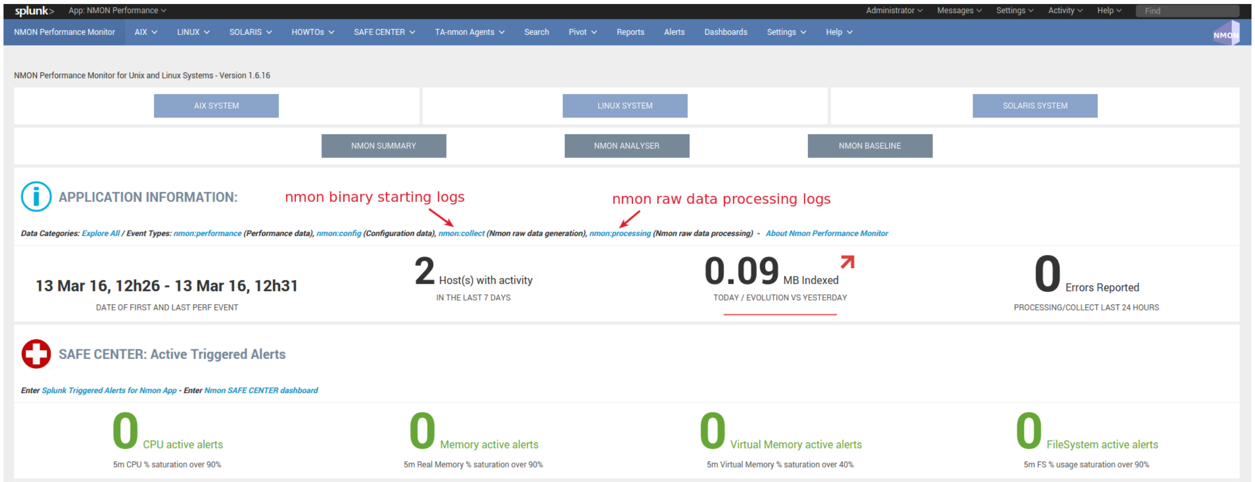
0 CPU active alerts — 5m CPU % saturation over 90%

0 Memory active alerts — 5m Real Memory % saturation over 90%

0 Virtual Memory active alerts — 5m Virtual Memory % saturation over 40%

0 FileSystem active alerts — 5m FS % usage saturation over 90%

You can check nmon binary starting logs and processing logs in associated eventtypes:



NMON Performance Monitor for Unix and Linux Systems - Version 1.6.16

Data Categories: Explore All / Event Types: nmon:performance (Performance data), nmon:config (Configuration data), nmon:collect (Nmon raw data generation), nmon:processing (Nmon raw data processing) - About Nmon Performance Monitor

13 Mar 16, 12h26 - 13 Mar 16, 12h31 DATE OF FIRST AND LAST PERF EVENT

2 Host(s) with activity IN THE LAST 7 DAYS

0.09 MB Indexed TODAY / EVOLUTION VS YESTERDAY

0 Errors Reported PROCESSING/COLLECT LAST 24 HOURS

Enter Splunk Triggered Alerts for Nmon App - Enter Nmon SAFE CENTER dashboard

0 CPU active alerts 5m CPU % saturation over 90%

0 Memory active alerts 5m Real Memory % saturation over 90%

0 Virtual Memory active alerts 5m Virtual Memory % saturation over 40%

0 FileSystem active alerts 5m FS % usage saturation over 90%

3.3. Deploying TA-nmon on master node, deployment server for performance data generation

For other nodes that won't have the TA-nmon published automatically (such as the master node and deployment servers), you will need to manually install the TA-nmon addon.

It must be uncompressed and installed in $SPLUNK_HOME/etc/apps/ (where $SPLUNK_HOME refers to the root directory of Splunk installation)

```
cd /opt/splunk/etc/apps

tar -xvzf <YOUR PATH>/TA-nmon_*.tar.gz
```

And restart:

```
splunk restart
```

2.8 Deploying Nmon Performance Monitor in SH Clusters

This is the recommended deployment scenario for search heads.

Installation is covered in details in the distributed deployment guide: *Installation for distributed deployments*

- See the detailed section: deploy_sh_cluster

2.9 Deploy to Splunk Cloud

2.9.1 Splunk Cloud compatibility and limitations

In a nutshell, the "mnmon-for-splunk" application will be deployed to your ad-hoc search head and the "PA-nmon_light" will be deployed by Splunk Cloud operations to your indexers stack.

To achieve this, submit a ticket to Splunk Cloud Ops to request the deployment of the Support Addon. (which gets deployed to the cluster master and then pushed to the indexers)

Then the application must be deployed on the Splunk Cloud search head, when an application has been vetted for Cloud Ops, this can be done as a self-service. (Otherwise submit the request to Cloud Ops teams)

Finally, you will use your own on premise tools to push and deploy the Technology Addon to your servers, which can forward either directly to your Splunk Cloud indexers (recommended) or through your on-prem intermediate forwarders.

The “nmon” index creation can be done as a self-service (Settings / Indexes) or requested to Cloud Ops in the same time.

2.9.2 Splunk Cloud deployment matrix

Splunk Cloud components:

Splunk roles	nmon-for-splunk	PA-nmon_light	TA-nmon-*
Search head	X		
Indexer		X	

The Support Add-on does not generate any collection, but defines the replicated nmon index and contains index time configuration settings.

On premise components: (you may not have all these roles on-premise depending on your configuration)

Splunk roles	nmon-for-splunk	PA-nmon_light	TA-nmon-*
Search head	X		X (optional)
Indexer		X	X (optional)
Master node			X (optional)
Deployment server		Conditional	Conditional
Heavy Forwarder		Conditional	Conditional
Universal Forwarder			X

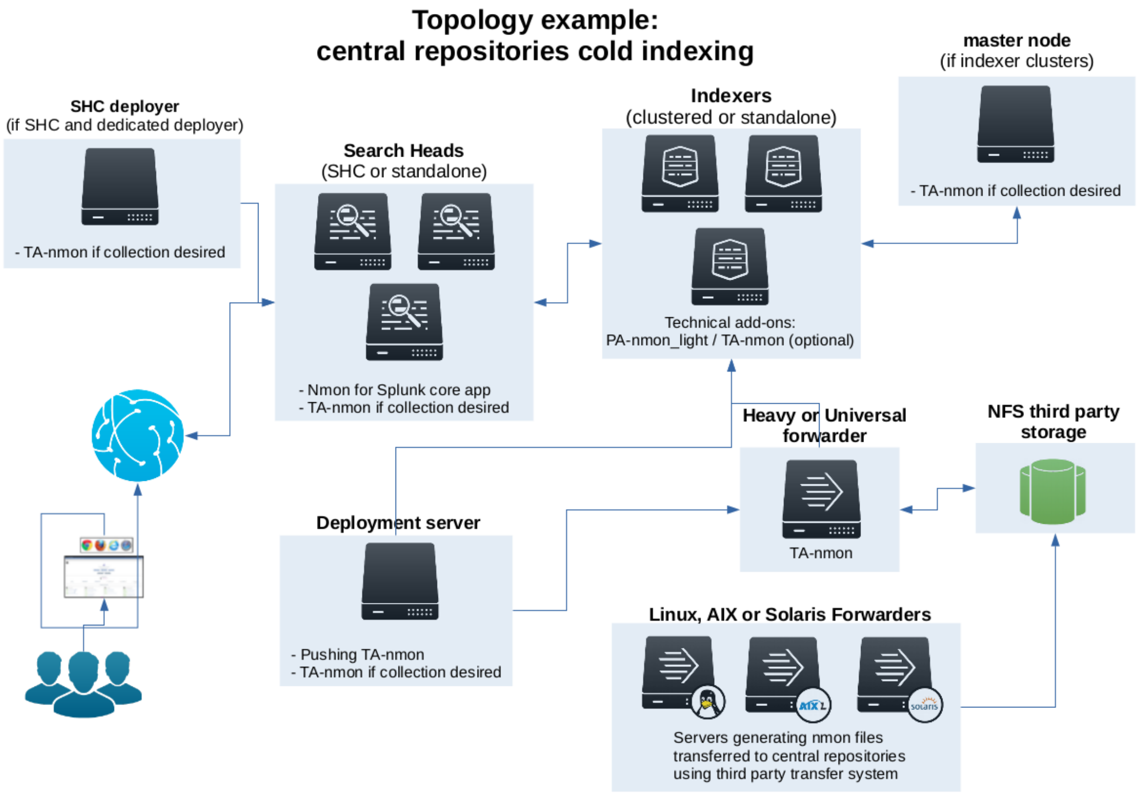
The Technology Add-ons provide metrics and configuration collection for the host than runs the add-on, which is optional.

The Support Add-on does not generate any collection, but defines the replicated nmon index and contains index time configuration settings.

2.10 Managing Nmon Central Repositories

A common scenario of Nmon Splunk App resides in using the Application to manage large and massive collections of cold Nmon raw data that have been generated out of Splunk.

Topology example:


Topology example:
central repositories cold indexing
master node
(if indexer clusters)
- TA-nmon if collection desired
SHC deployer
(if SHC and dedicated deployer)
- TA-nmon if collection desired
Indexers
(clustered or standalone)
Technical add-ons:
PA-nmon_light / TA-nmon (optional)
Search Heads
(SHC or standalone)
- Nmon for Splunk core app
- TA-nmon if collection desired
Heavy or Universal forwarder
TA-nmon
NFS third party storage
Deployment server
- Pushing TA-nmon
- TA-nmon if collection desired
Linux, AIX or Solaris Forwarders
Servers generating nmon files transferred to central repositories using third party transfer system

Such a configuration is quite easy to achieve, the only requirement is having a Splunk instance (Heavy or Universal Forwarder) having custom input monitors to watch for your Nmon files. (most of the time they would be hosted in NFS shares).

INFORMATION: The application can now manage cold and hot Nmon data even when centralized in a common place (see above)

2.10.1 Indexing Nmon data generated ouf of Splunk

The Application can manage multiple custom input monitors to automatically index your collection of Nmon raw data files:

It is important to create a custom input in a “local/inputs.conf” such that it does not get overwritten when the Application gets updated.

Here are some examples of monitor configuration:

```
[monitor:///mnt/NFS-SHARE/nmon-repository/*/*nmon]
disabled = false
index = nmon
sourcetype = nmon_processing
crcSalt = <SOURCE>
```

Or an alternative version using whitelist and manage any nmon files in sub folders:

```
[monitor:///mnt/NFS-SHARE/nmon-repository/]
disabled = false
```

(continues on next page)

(continued from previous page)

```
whitelist = \.nmon$
index = nmon
sourcetype = nmon_processing
crcSalt = <SOURCE>
```

And even:

```
[monitor:///mnt/NFS-SHARE/nmon-repository/.../*.nmon]
disabled = false
index = nmon
sourcetype = nmon_processing
crcSalt = <SOURCE>
```

If you are synchronizing HOT nmon data:

After adding the input, please restart Splunk.

Immediately after restart, the App should start managing available Nmon files, look in the nmon_processing sourcetype to get the current activity of the Nmon processing steps:

```
index=nmon sourcetype=nmon_processing
```

Or open the report: "Activity of NMON Data Processing"

2.11 Eventgen testing

2.11.1 Testing Nmon performance with evengen

Splunk Evengen is a pretty good and straightforward way to test the application.

Starting the TA-nmon version 1.3.28 and TA-nmon-hec version 1.3.32, we provide sample data for 2 AIX and 2 Linux servers. The data has been generated on IBM Power Development Cloud servers.

Finally, we use to run a system stress tool on 1 server of each category, such that you will have quickly active alerts and system statistic anomalies.

Eventgen will generate data for:

- performance metrics (sourcetype=nmon_data / eventtype=nmon:performance)
- configuration data (sourcetype=nmon_config / eventtype-nmon:config)

Additional data normally available within the application is related to the nmon data collection and will not be generated by Eventgen.

2.11.2 Get it working in 30 seconds

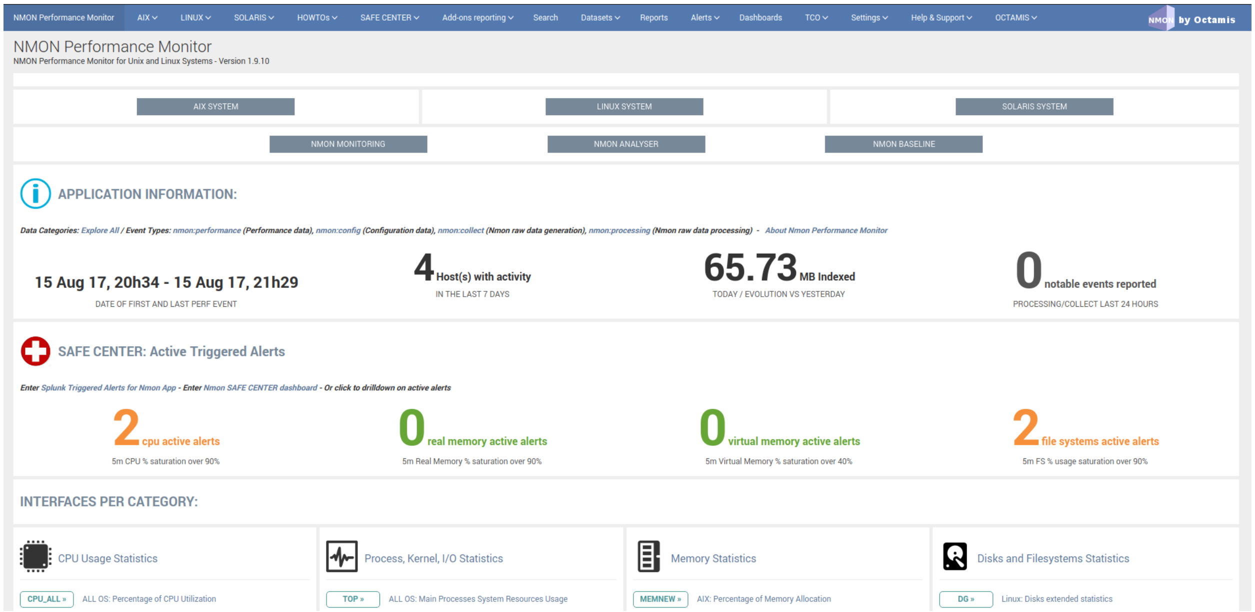
- Have a Splunk instance up and running
- Download the current eventgen version from https://github.com/splunk/eventgen
- Install the eventgen application, you should name the application directory as:

```
$SPLUNK_HOME/etc/apps/SA-Eventgen
```

- If not done already, install the Nmon Performance application (obvious!)

- Install either the TA-nmon or the TA-nmon-hec on this instance
- Create an index called "nmon"
- Restart Splunk

Immediately after Splunk restart, eventgen starts to generate nmon data, as visible from the application home page:



NMON Performance Monitor

NMON Performance Monitor for Unix and Linux Systems - Version 1.9.10

APPLICATION INFORMATION:

Data Categories: Explore All / Event Types: nmon:performance (Performance data), nmon:config (Configuration data), nmon:collect (Nmon raw data generation), nmon:processing (Nmon raw data processing) - About Nmon Performance Monitor

15 Aug 17, 20h34 - 15 Aug 17, 21h29
DATE OF FIRST AND LAST PERF EVENT

4 Host(s) with activity
IN THE LAST 7 DAYS

65.73 MB Indexed
TODAY / EVOLUTION VS YESTERDAY

0 notable events reported
PROCESSING/COLLECT LAST 24 HOURS

SAFE CENTER: Active Triggered Alerts

Enter Splunk Triggered Alerts for Nmon App - Enter Nmon SAFE CENTER dashboard - Or click to drilldown on active alerts

2 cpu active alerts
5m CPU % saturation over 90%

0 real memory active alerts
5m Real Memory % saturation over 90%

0 virtual memory active alerts
5m Virtual Memory % saturation over 40%

2 file systems active alerts
5m FS % usage saturation over 90%

INTERFACES PER CATEGORY:

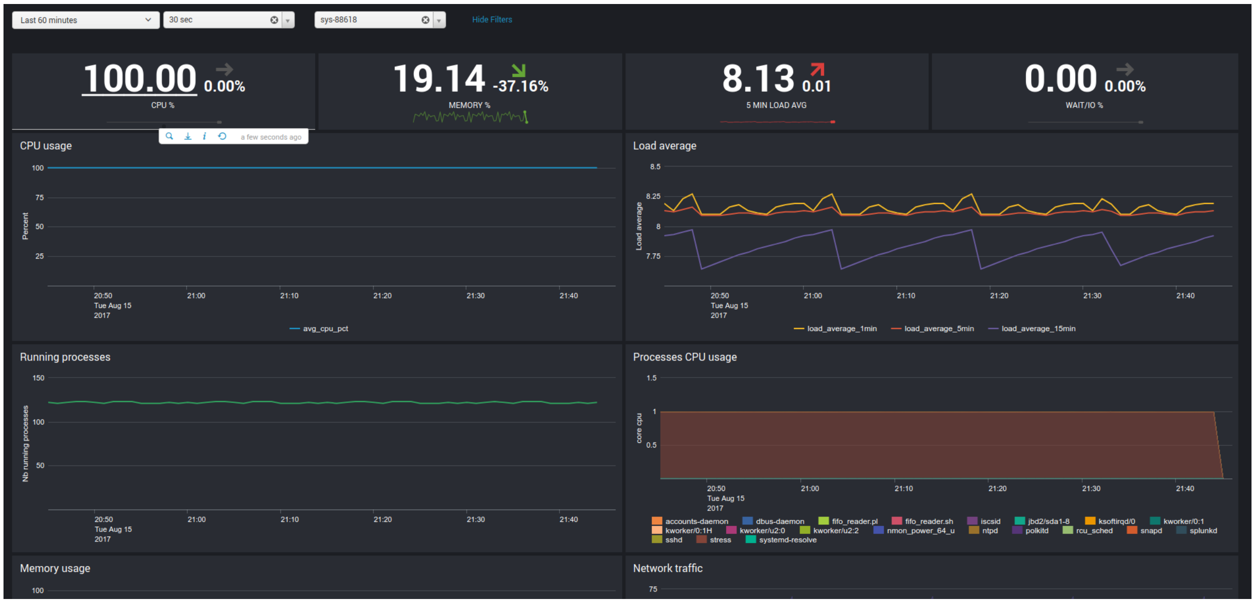
CPU Usage Statistics
CPU_ALL » ALL OS: Percentage of CPU Utilization

Process, Kernel, I/O Statistics
TOP » ALL OS: Main Processes System Resources Usage

Memory Statistics
MEMNEW » AIX: Percentage of Memory Allocation

Disks and Filesystems Statistics
DG » Linux: Disks extended statistics

Example of a server running with abnormal load:



2.12 Upgrade

2.12.1 01 - Upgrade Standalone Instance

Upgrade or Update the Nmon Splunk App in a Splunk standalone instance

Updating the Nmon App on a minor release or upgrade to a major new release is totally transparent and uses Splunk standard.

IMPORTANT: As for any other Splunk Application, do never modify configuration files in the default directory but create your own copy in the local directory, such that updating the Application will not overwrite your custom settings

To update or upgrade Nmon Splunk App in a standalone installation, you can:

- Use the Splunk App manage builtin, Splunk automatically notifies you when a new version is available, the update can be done on the air through the Manager
- Download the new version on Splunk base https://splunkbase.splunk.com/app/1753/ and use the Manager to proceed to update
- Uncompress directly the content of the tar.gz archive in $SPLUNK_HOME/etc/apps and restart Splunk

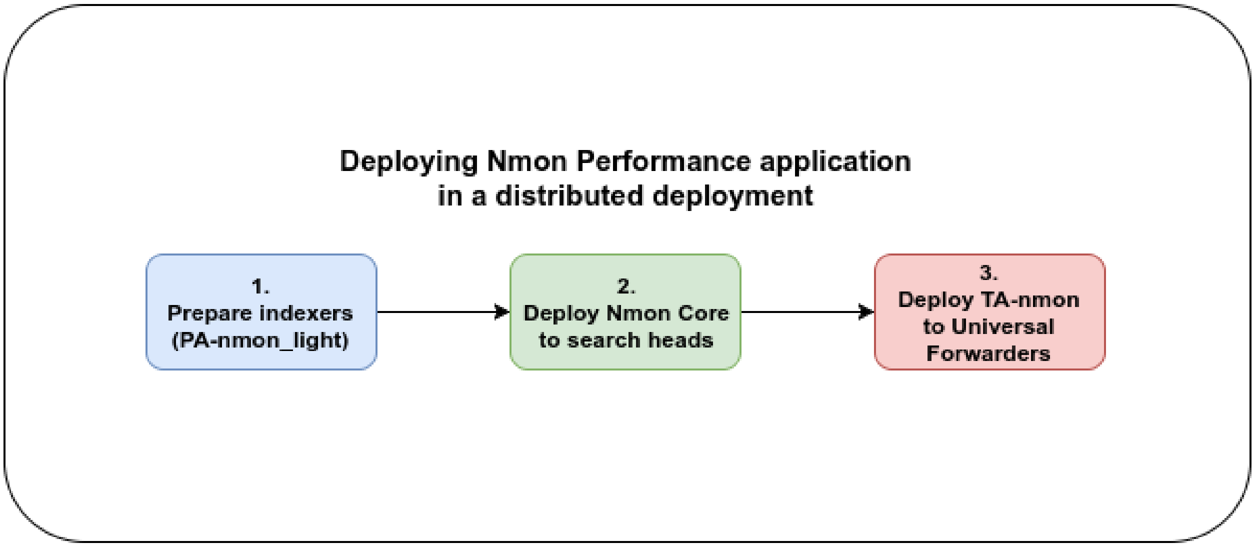
2.12.2 02 - Upgrade Distributed Deployment

Upgrade or Update the Nmon Splunk App in a Splunk Distributed Deployment

Updating the Nmon App on a minor release or upgrade to a major new release is totally transparent and uses Splunk standard.

IMPORTANT: As for any other Splunk Application, do never modify configuration files in the default directory but create your own copy in the local directory, such that updating the Application will not overwrite your custom settings

Updating the Application in a Distributed Deployment context follows the same tracking than initial deployment, with three major pieces of the App:



So, proceed in the order:

- Update PA-nmon_light and TA-nmon (optional)
- Update Nmon Core App and TA-nmon (optinal)

- Update TA-nmon and reload your deployment server to update your end clients

Please consult the Distributed Deployment manual to get details instructions of each step for upgrade: *Installation for distributed deployments*

2.12.3 03 - Migrating from release prior to Version 1.7.x

Upgrade notes

The release V1.7.x is a major release of the Nmon Performance Monitor application, follow this procedure when migrating from an existing installation running a version previous to the V1.7.x.

SUMMARY OF MAJOR CHANGES

- The Nmon core application does not create anymore the "nmon" index at installation time (for app certification purposes), the index must be declared manually
- The Nmon core application does not implement anymore data collection, if you want to get performance data of your search heads you must deploy the TA-nmon
- The TA-nmon working directory has been migrated from $SPLUNK_HOME/var/run to $SPLUNK_HOME/var/log for certification purposes
- The nmon_inventory lookup table is now stored in a KVstore collection, after upgrade you must re-generate the nmon inventory data to fill the KVstore (or wait for the next auto iteration)
- Different old components were removed from the core application (such a the django views), extracting using tar will not clean these files
- The span definition macro "custom_inlinespan" where renamed to "nmon_span" for easier usage, if you used to customize the minimal span value previously, you must update your local configuration (the original macro still exists in case of users would be using it, but it is not used anymore in views)

FILES AND DIRECTORY THAT WERE REMOVED FROM THE CORE APPLICATION

Bellow is the list of files and directory that were removed in the Version 1.7.x, at the end of your update you can clean these files with no issue.

If you are running standalone search head, remove them from:

```
$SPLUNK_HOME/etc/apps/nmon
```

If you are running a Search Head Cluster, remove them the deployer and apply the bundle configuration to the search head

```
$SPLUNK_HOME/etc/shcluster/apps/nmon
```

FILES AND DIRECTORIES TO BE REMOVED:

- nmon/bin
- nmon/django
- nmon/default/inputs.conf
- nmon/default/inputs.conf_forWindows
- nmon/default/indexes.conf
- nmon/lookups/nmon_inventory.csv
- nmon/samples

All these files, directories and sub-directories can be removed safety.

PRE-CHECK - HAVE YOU DECLARED YOUR INDEX ?

The nmon core application does create anymore the "nmon" index at startup time.

This is a requirement for Splunk application certification, as this task should be managed by Splunk administrators.

If you running in Indexer cluster, then your index has necessarily be declared and you are not concerned.

If you running standalone instances, ensure you have set your index explicitly, you can create the "nmon" index the local/ directory of the Nmon core application for example.

STEP 1. UPDATE THE CORE APPLICATION

If you are running on a standalone installation only, you should declare the "nmon" index manually before upgrading, or at least before restarting.

Refer to the standalone installation guide: *Installation for standalone instance*

If you running the PA-nmon or an indexer cluster where you have already manually declared the nmon index, you are not affected by this change.

Apply the installation procedure following your configuration, checkout:

- Upgrade a standalone server: *01 - Upgrade Standalone Instance*
- Upgrade a distributed deployment: any:*upgrade_distributed*

inputs.conf

Clean the default/inputs.conf and local/inputs.conf on the search head If you were generating performance and configuration data at the search head level using the Nmon core application, you should delete these files as they are not useful anymore.

STEP 2. DEPLOY THE TA-NMON ON SEARCH HEADS IF RELEVANT

Since the release V1.7.4, you must deploy the TA-nmon on the search head level if you want to collect performance and configuration data from the search heads

This will be easily achieved by the the deploying the TA-nmon along with the Nmon core application on the SHC deployer, checkout: *Installation for distributed deployments*

STEP 3. CHECKOUT YOUR LOCAL CONFIGURATION ACCORDING TO MAJOR CHANGES

According to the summary of major changes, you may have to:

- Review your local/macros.conf if you are using a custom minimal value for the span definition, see *Custom Span definition macro*
- Manually re-generate the nmon inventory data by running the "Generate NMON Inventory Lookup Table" report, for more information, see: *nmon_inventory*

2.12.4 04 - Migrating from release prior to Version 1.9.x

Migrate from version 1.7.x to version 1.9.x

Please refer to: *03 - Migrating from release prior to Version 1.7.x*

Migrate from version 1.8.x to version 1.9.x

The release 1.9.x is a new main release for the Nmon Core application, and as well for the TA-nmon technical addon

There are some changes in these releases than can require specific actions:

- The PA-nmon has been deprecated, it is now unified with the TA-nmon (the TA-nmon replaces the PA-nmon on indexers)
- The TA-nmon_selfmode has been deprecated, it is as well unified with the TA-nmon
- The TA-nmon introduces the fifo implementation which provides the lower level of foot print on servers

What is the upgrade path then ?

If you have previously deployed the PA-nmon in your clustered indexers, follow these simple steps:

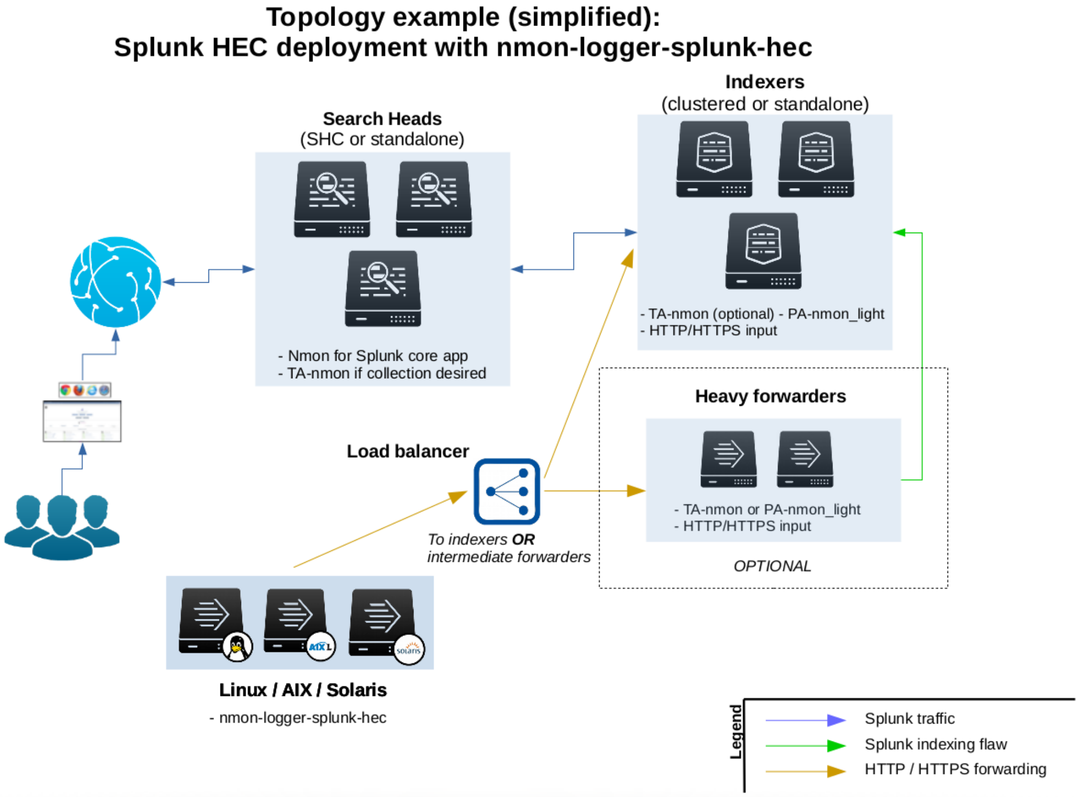
- if you have defined any custom index in the PA-nmon, backup this configuration and move it to the PA-nmon_light (see above)
- remove the PA-nmon from the "master-apps" of the master node
- extract the PA-nmon_light archive in the master node "master-apps" directory
- extract the TA-nmon archive in the master node "master-apps" directory if you want to collect performance statistics from your indexers
- apply the cluster bundle
- after the indexers rolling restart, kill any existing nmon processes, or wait their end and assume a gap of 2 hours maximum in the performance data

If you have customized the interval and/or snapshot values in "nmon.conf":

- the new TA-nmon does not use any more the same variables in nmon.conf (see http://ta-nmon.readthedocs.io/en/latest/nmon_config.html)
- the reason why is that with the fantastic gain in TA-nmon foot print, it is not required anymore to run short life nmon cycles to limit the CPU and other resources costs
- the default and recommended life time for an nmon process is 24 hours
- if you used to modify the "interval" value to reduce the volume of data (which is already very low!), back port this configuration in the new variables

2.13 Splunk HEC / nmon-logger deployment

The "nmon-logger" package for Splunk HEC provides a 100% agent less configuration using the Splunk http input:



The nmon-logger is **not** a Splunk application, this is an independent package to be deployed to your Operating System.

This deployment provides the following features:

- **clients easy set up:** the nmon-logger is provided as deb/rpm package, easy and fast deployment
- **server easy set up:** Splunk http input is easy to configure and implement
- **100% agent less:** the nmon-logger uses only native system features (cron, logrotate. . .)
- **secure:** Splunk http traffic can easily be encrypted via SSL and integrated into any DMZ or similar restricted networking layer
- **resilient and scalable:** using load balancers and multiple nodes provides resiliency and horizontal scalability
- **network friendly:** as a Web service, it can be easily used across wide networks and over the Internet
- **easy management:** since the http input is managed on a token basis, you can easily configure different tokens to ingest the data into different indexes without any package modification or complexity

2.13.1 Deployment matrix

Splunk Instance (role)	Core App	PA-nmon_light	TA-nmon	nmon-logger
Search head (single or clustered)	X		X (optional)	
Indexer (single or clustered)		X	X (optional)	
Master node			X (optional)	
Deployment servers			X (optional)	
Heavy Forwarder			X	
Universal Forwarder			X	
Client servers				X

Notes:

- Indexing time parsing operations require the PA-nmon_light or the TA-nmon (or both) to be deployed on the host running the http input
- The Nmon core app **version 1.9.10** minimal, TA-nmon **version 1.3.27** minimal and PA-nmon_light **version 1.3.19** are required on the Splunk infrastructure
- The http input can run either on indexers, or one or more heavy forwarders

Fast testing using Vagrant and Ansible:

If you are interested in a very fast and automated way to test the Nmon Performance Application with an HEC nmon-logger deployment, checkout the provided configuration using the excellent Vagrant (https://www.vagrantup.com/) and Ansible configuration management (http://docs.ansible.com/ansible/index.html)

- Checkout: https://github.com/guilhemmarchand/nmon-logger/tree/master/vagrant-ansible-demo-splunk-hec

In about 5 minutes, have a running and automated deployment working !

2.13.2 HEC performance considerations

For best HEC performance purposes, the nmon-logger works the following way:

- performance and configuration data are streamed in "batch" mode, which means we only generate one HEC connection for each during an occurrence of the nmon_processing (which occurs every minute)
- collection, processing and other data being generated by the nmon-logger work as well in batch mode, one connection per processing streams the full data
- most of Metadata are part of each event sent to the HEC

See: http://dev.splunk.com/view/event-collector/SP-CAAAE73

2.13.3 Download the nmon-logger-splunk-hec package

The nmon-logger-splunk-hec package is available in the Github repository of the nmon-logger:

- https://github.com/guilhemmarchand/nmon-logger

The nmon-logger is provided as a deb and rpm package for Linux OS and AIX, it has been tested against:

- Ubuntu (x86 and Powerpc)
- Debian (x86)
- CentOS (x86)

- RHEL (x86 and Powerpc)
- Suse (x86 and Powerpc)
- OpenSuse (x86)
- AIX 7.1
- AIX 7.2

2.13.4 Activate the Splunk http input and create a token

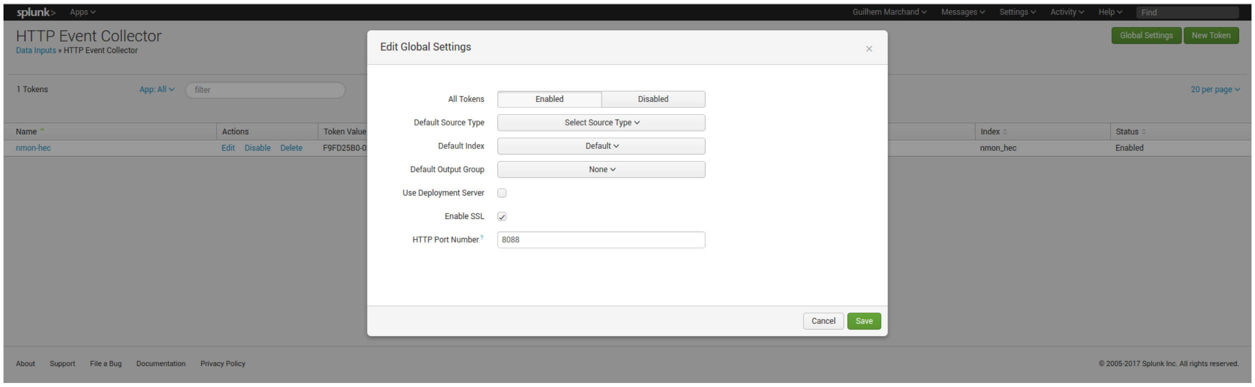
The Splunk configuration is really straightforward, it is all about:

- Activating and the http input: configuring the http port, choosing between http and https
- Creating a token for the nmon data (1 token for all data, but you can create multiple tokens for different servers deployement)

Notes:

- http and https are supported
- indexer acknowledgment is not currently supported (configured per token)
- the nmon-logger will not explicitly specify an index, you choose the index to be used on a per token basis
- Any index name starting by "nmon" is natively taken in charge by the Nmon Performance application
- If you choose a different index name that does not match the rule above, you just need to customize the eventtypes.conf and macros.conf of the Nmon app
- it is not required to define any sourcetype / source by default

In a nutshell:



HTTP Event Collector

Data Inputs » HTTP Event Collector

1 Tokens   App: All   filter   20 per page

Name	Actions	Token Value	Index	Status
nmon-hec	Edit Disable Delete	F9FD25B0-0	nmon_hec	Enabled

Edit Global Settings

All Tokens: Enabled | Disabled

Default Source Type: Select Source Type

Default Index: Default

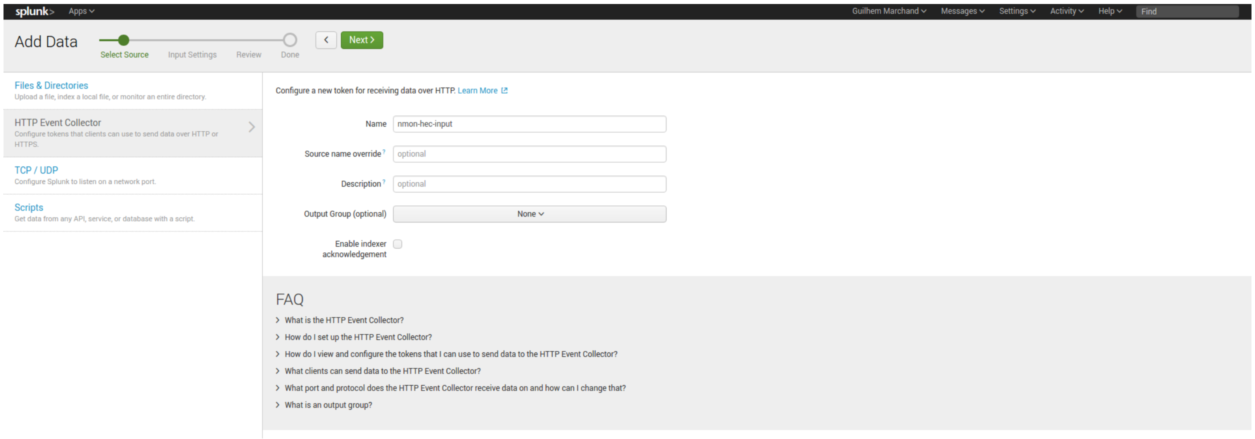
Default Output Group: None

Use Deployment Server: □

Enable SSL: ☑

HTTP Port Number: 8088

Cancel   Save



Files & Directories
Upload a file, index a local file, or monitor an entire directory.

HTTP Event Collector
Configure tokens that clients can use to send data over HTTP or HTTPS.

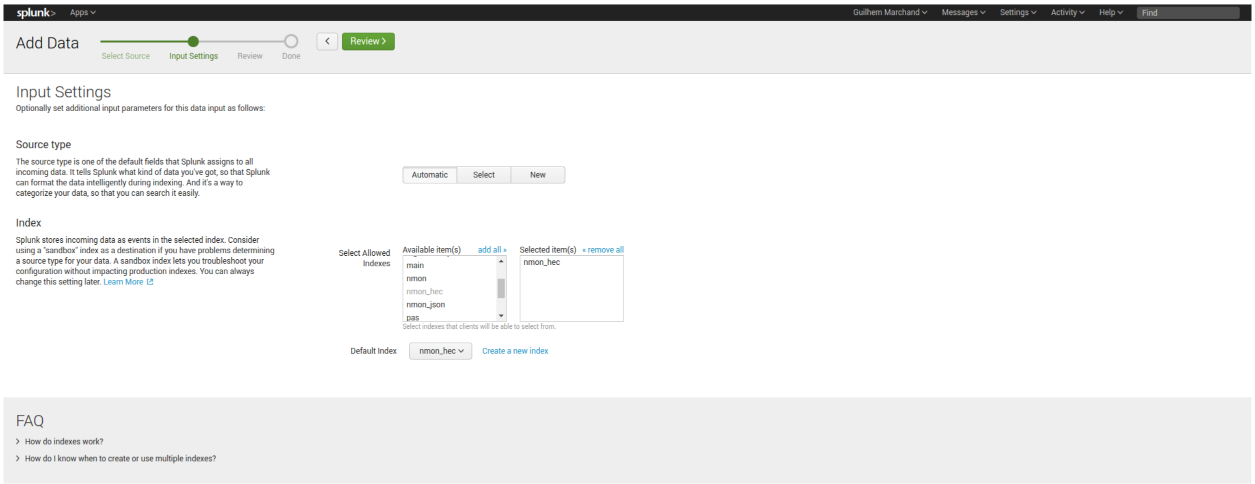
TCP / UDP
Configure Splunk to listen on a network port.

Scripts
Get data from any API, service, or database with a script.

Configure a new token for receiving data over HTTP. Learn More

Name: nmon-hec-input
Source name override: optional
Description: optional
Output Group (optional): None
Enable indexer acknowledgement: □

FAQ
- What is the HTTP Event Collector?
- How do I set up the HTTP Event Collector?
- How do I view and configure the tokens that I can use to send data to the HTTP Event Collector?
- What clients can send data to the HTTP Event Collector?
- What port and protocol does the HTTP Event Collector receive data on and how can I change that?
- What is an output group?



Input Settings
Optionally set additional input parameters for this data input as follows:

Source type
The source type is one of the default fields that Splunk assigns to all incoming data. It tells Splunk what kind of data you've got, so that Splunk can format the data intelligently during indexing. And it's a way to categorize your data, so that you can search it easily.

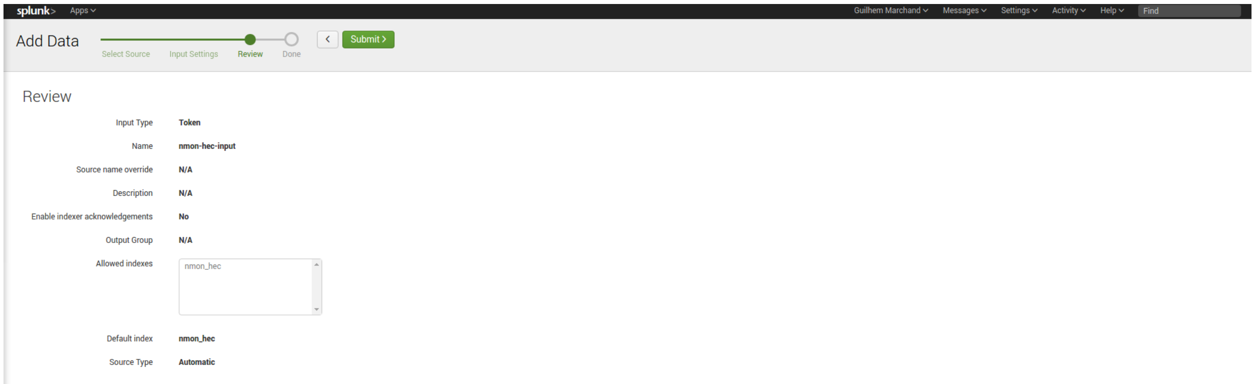
Automatic | Select | New

Index
Splunk stores incoming data as events in the selected index. Consider using a "sandbox" index as a destination if you have problems determining a source type for your data. A sandbox index lets you troubleshoot your configuration without impacting production indexes. You can always change this setting later. Learn More

Select Allowed Indexes
Available item(s) add all »: main, nmon, nmon_hec, nmon_json, pas
Selected item(s) « remove all: nmon_hec
Select indexes that clients will be able to select from.

Default Index: nmon_hec
Create a new index

FAQ
- How do indexes work?
- How do I know when to create or use multiple indexes?



Review

Input Type	Token
Name	nmon-hec-input
Source name override	N/A
Description	N/A
Enable indexer acknowledgements	No
Output Group	N/A
Allowed indexes	nmon_hec
Default index	nmon_hec
Source Type	Automatic

Configuration files:

- "$SPLUNK_HOME/etc/apps/splunk_http_input/local/inputs.conf":

```
[http]
disabled = 0
```

- "$SPLUNK_HOME/etc/apps/<appname>/local/inputs.conf":

Notes: replace <appname> with the application context where you want to store the configuration inputs.conf file

```
# inputs.conf

# Enable the HEC
[http]
disabled = 0
enableSSL = 1

# HEC endpoint for clients
[http://nmon-hec-input]
disabled = 0
index = nmon_hec
indexes = nmon_hec
token = CEE56643-BA2D-48EE-94EF-AD0909718B2A
```

2.13.5 Deploying the nmon-logger to your servers

Linux OS

This is package (no arch) to be deployed, which is obviously straight forward:

deb based OS:

```
dpkg -i nmon-logger-splunk-hec-*.deb
```

rpm based OS:

```
rpm -i nmon-logger-splunk-hec-*.rpm
```

Notes:

- Host running SeLinux (likely RHEL for instance) need to have the “permissive mode” enabled for the rpm installation or the groupadd operation might fail:

```
sudo setenforce 0
```

- Some systems (likely on RHEL), the perl-Time-HiRes may not be installed by default:

```
yum install -y perl-Time-HiRes
```

AIX OS

Download the rpm package according to your version, and install as usual:

rpm based OS:

```
rpm -i nmon-logger-splunk-hec-*.rpm
```

Notes about AIX 6.1: the nmon-logger has not been tested against out of support AIX version but is expected to operate normally

Installing rpm package manager:

See: https://ftp.software.ibm.com/aix/freeSoftware/aixtoolbox/ezinstall/ppc/README-yum

2.13.6 Configuring the nmon-logger

The data collection starts 1 minute maximum after the package deployment, as long as you don't have configured the URL and token, **the data is only generated locally on the file system**.

Create a local directory:

```
mkdir /etc/nmon-logger/local
```

Create a local/nmon.conf and insert your URL / Token:

/etc/nmon-logger/local/nmon.conf, example:

```
# HEC server configuration

nmon2csv_options="--mode fifo --silent --splunk_http_url https://192.168.33.100:8088/
↪services/collector/event --splunk_http_token CEE56643-BA2D-48EE-94EF-AD0909718B2A"
```

Et voila!

Once the nmon-logger package is configured and if the networking configuration is properly configured, Splunk will start receiving data through the http input !

2.13.7 Foot-print and benchmarking

The **nmon-logger** globally shares the same components than the **TA-nmon**, as the difference that the CSV data is being transformed into key value data and streamed to the Splunk http input. (nmon2csv parsers are nmon2kv!)

See:

- http://ta-nmon.readthedocs.io/en/latest/processing_overview.html
- http://ta-nmon.readthedocs.io/en/latest/data_processing.html
- http://ta-nmon.readthedocs.io/en/latest/footprint.html

The foot-print related to the generation, processing and streaming of the performance and configuration data is very low, it is actually even lower than the TA-nmon since there are no overhead due to the Splunk instance.

benchmarking reports will be added shortly!

2.14 rsyslog / nmon-logger deployment

Syslog deployment topology - Generate and forward Nmon performance data from rsyslog clients to your centralized rsyslog servers

Introduced with the Nmon Performance Monitor 1.6.14, you can now get real time Nmon data from end servers without any Splunk Universal Forwarders deployment.

This will be achieved using:

- The nmon-logger package to be deployed on servers: https://github.com/guilhemmarchand/nmon-logger
- rsyslog locally available on servers and configured to send data to central syslog servers
- Splunk Universal or Heavy Forwarder instance installed in rsyslog servers (collector or additional relays) that will monitor and send data in a "key=value" format to Splunk
- Deploying the nmon-logger package to your end-servers, the package is now provided in **rpm** and **deb** packages

Please review requirements in the above section.

Fast testing using Vagrant and Ansible:

If you are interested in a very fast and automated way to test the Nmon Performance Application with an rsyslog deployment, checkout the provided configuration using the excellent Vagrant (https://www.vagrantup.com/) and Ansible configuration management (http://docs.ansible.com/ansible/index.html)

- Checkout: https://github.com/guilhemmarchand/nmon-logger/tree/master/vagrant-ansible-demo-rsyslog

In about 5 minutes, have a running and automated deployment working !

2.14.1 Key concepts of an rsyslog deployment

Why an syslog topology versus Universal Forwarders deployment ?:

- At first, this provides a powerful and resilient alternative way to deploy the Nmon Performance monitor
- Syslog is a Unix / Linux standard, and available on many servers
- Because you may already have an rsyslog centralization available and you do not want to deploy any additional software on servers
- Because sometimes companies don't want to rely on proprietary software on end servers, deploying Universal Forwarders is not an option

Key concepts :

- 100% of Application features over a traditional Universal Forwarders deployment
- In a standard deployment of the TA-nmon and the Nmon Performance application, hosts generates Nmon Performance in csv structured data
- To be transported over syslog, csv data are being transformed in a key=value format
- While csv structured data is known in Splunk as "an indexing field structure", key=value format will be extracted on the fly (parsing at extraction time)
- csv structured data has an higher disk space cost but offers a very low level of volume of data to index (and so a low level of licence cost), and offers best performances at search time in SPL (Search Language Processing)
- key=value format generates an higher volume of data and requires more power at extraction time, but it does not generate indexed fields, this also represents less disk space
- Nmon Performance massively uses data model acceleration, key=value searching performance versus csv structured data will not be different from standard deployment

Can i send data directory from syslog to Splunk ?

Yes it is possible, but this is not the deployment topology i would recommend for multiple reasons, as exposed above:

- You cannot guarantee that rsyslog will send data in the order it should, this is especially true with multi-line events that could be mixed between hosts

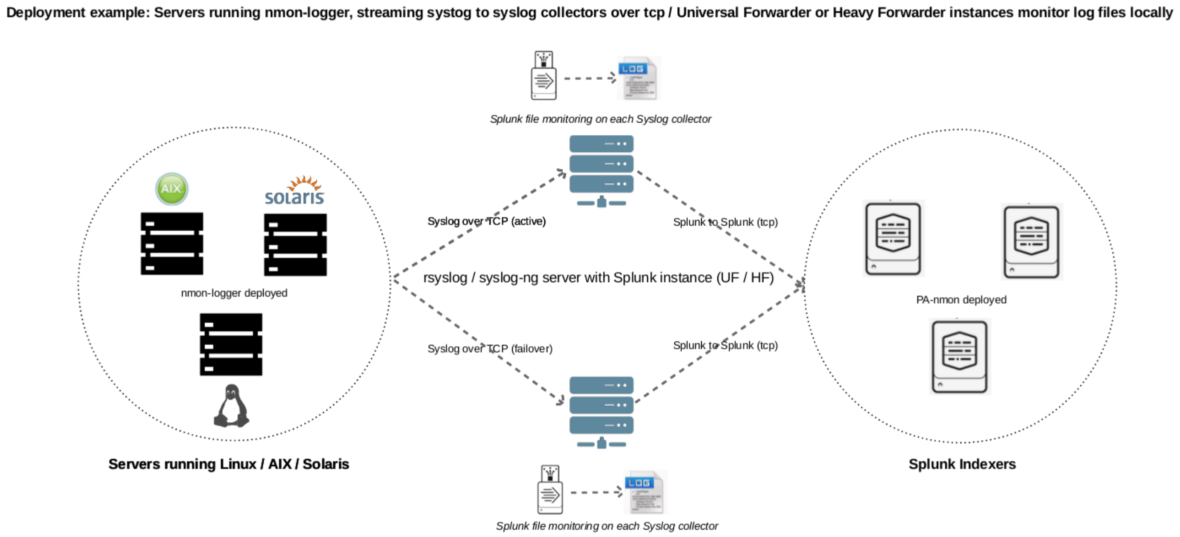
- rsyslog can easily identify the host origin of the incoming data and generate per host files, which guarantees management of Nmon data (like large multi-line configuration data)
- For resilient reasons, once it is configured, you will few often restart rsyslog. (most often while rebooting the machine)
- This is less true with Splunk as you will want to upgrade it from time to time, or deploy new configuration or application to manage new data
- Finally syslog speaks to syslog, with the same native implementation and features

What about the nmon-logger deployment management ?:

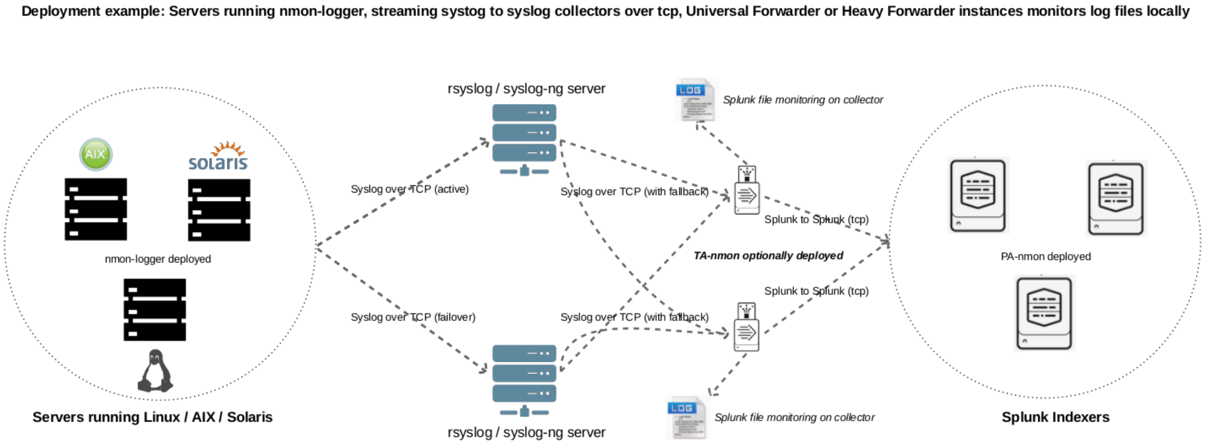
- In standard Universal Forwarder management, you can easily rely on native Splunk deployment servers to push and maintain the TA-nmon package
- In the syslog deployment, you will deploy nmon-logger packages (rpm, deb) or deploy nmon-logger manually

Topology: Examples of possible implementations:

Example 1: Splunk Universal or Heavy forwarder installed on main rsyslog collectors:


Deployment example: Servers running nmon-logger, streaming systog to syslog collectors over tcp / Universal Forwarder or Heavy Forwarder instances monitor log files locally
Splunk file monitoring on each Syslog collector
AIX
solaris
nmon-logger deployed
Servers running Linux / AIX / Solaris
Syslog over TCP (active)
Syslog over TCP (failover)
rsyslog / syslog-ng server with Splunk instance (UF / HF)
Splunk to Splunk (tcp)
Splunk to Splunk (tcp)
PA-nmon deployed
Splunk Indexers
Splunk file monitoring on each Syslog collector

Example 2: Splunk Universal or Heavy forwarder installed third party servers running rsyslog:


Deployment example: Servers running nmon-logger, streaming systog to syslog collectors over tcp, Universal Forwarder or Heavy Forwarder instances monitors log files locally
rsyslog / syslog-ng server
Splunk file monitoring on collector
AIX
solaris
nmon-logger deployed
Servers running Linux / AIX / Solaris
Syslog over TCP (active)
Syslog over TCP (with fallback)
Splunk to Splunk (tcp)
TA-nmon optionally deployed
Splunk to Splunk (tcp)
Syslog over TCP (failover)
Syslog over TCP (with fallback)
PA-nmon deployed
Splunk Indexers
rsyslog / syslog-ng server
Splunk file monitoring on collector

PRE-REQUISITES:

- Splunk + Nmon Performance app:

First of all, have a Splunk working installation, and the Nmon Performance up and running ! (yeah, songs like an evidence :-)

Some specific requirements must be respected to achieve a deployment that uses rsyslog as the transport layer:

- RSYSLOG V8.x:

RSYSLOG V8.x is required to forward Nmon Performance data to centralized rsyslog servers, see:

http://www.rsyslog.com/

On end servers, the "imfile" plugin will be used to read and collect Nmon Performance data.

- Python 2.7.x **OR** Perl with the module Time::HiRes:

The nmon-logger will by default search for a Python 2.7.x environment. If it is not available, scripts will use Perl, when using Perl note that the Time::HiRes module is required.

STEP 1 : Rsyslog configuration for central collectors

A minimal configuration is required on Syslog collectors, this will make rsyslog to listen on a dedicated TCP port to receive incoming data from end servers.

In the following example, rsyslog will listen to the TCP / 514 port:

Let's create a central configuration:

NOTE: Ensure the prefix value is always higher than the prefix value for your nmon_performance logging config file (20 in the example above) to prevent data duplication

```
/etc/rsyslog/rsyslog.d/99-central_logging.conf

# rsyslog configuration for central logging
# Note: 'rsyslog-central' must be replaced to match your hostname
# 'localhost' is expected to work, but some persistent cases shown that only
# setting to the real value of the host name prevents from logging local log␣
↪duplicated
# in remote location

# provides TCP syslog reception
$ModLoad imtcp
$InputTCPServerRun 514

# Set the global dynamic file
$template PerHost, "/var/log/remote-hosts/%HOSTNAME%/%HOSTNAME%.log"

if ($hostname != 'syslog-central') then ?PerHost
& stop
```

Create the directory and correctly set permissions for syslog user:

```
mkdir /var/log/remote-hosts

chown syslog:syslog /var/log/remote-hosts
```

Finally, create a configuration file that catch Nmon Performance events and log it in dedicated files:

```
/etc/rsyslog/rsyslog.d/20-nmon-performance.conf

# Nmon Performance configuration (validated over rsyslog 8.15)

# Turn off non printable chars replacing
$EscapeControlCharactersOnReceive off

# Set the global dynamic file
$template DynFile,"/var/log/nmon-performance/%HOSTNAME%/%programname%.log"

if $programname startswith 'nmon_performance' then ?DynFile
& stop

if $programname startswith 'nmon_config' then ?DynFile
& stop

if $programname startswith 'nmon_collect' then ?DynFile
& stop

if $programname startswith 'nmon_processing' then ?DynFile
& stop

if $programname startswith 'nmon_clean' then ?DynFile
& stop
```

Create the directory and correctly set permissions for syslog user:

```
mkdir /var/log/nmon-performance

chown syslog:syslog /var/log/nmon-performance
```

Restart Rsyslog:

```
sudo service rsyslogd restart
```

STEP 2 : Rsyslog configuration for end servers

Each of your end servers must be configured to send its syslog data to the central rsyslog server.

First, let's activate the imfile module that will be used to read and send Nmon Performance data:

```
/etc/rsyslog.conf
```

In the MODULES section, add:

```
module(load="imfile")    # if you want to tail files
```

Now, create the central client configuration that forwards local log to rsyslog central servers:

```
/etc/rsyslog.d/01-central-syslog.conf

*.* @@syslog-central:514
```

rsyslog has native support for fail over data forwarding, if you have a backup rsyslog collectors:

If the first rsyslog server is unavailable, data will be forwarded to the backup server, if none are available, data is being temporily stored until one of remote servers is up again:

See: http://wiki.rsyslog.com/index.php/FailoverSyslogServer

```
*.* @@syslog-central:514
$ActionExecOnlyWhenPreviousIsSuspended on
*.* @@syslog-central:514
*.* @@syslog-central2:514
& /var/log/localbuffer
$ActionExecOnlyWhenPreviousIsSuspended off
```

Restart Rsyslog:

```
sudo service rsyslogd restart
```

Immediately after the restart, rsyslog starts to forward data to central rsyslog servers.

STEP 3 : Deploy the nmon-logger to your end servers

On each end server, you must deploy the "nmon-logger" package:

https://github.com/guilhemmarchand/nmon-logger

Using your package manager

For compatible operating systems using the "deb" Debian package manager (Debian, Ubuntu. . .) and the "rpm" Redhat package manager (CentOS, RHEL. . .) you can easily deploy the pre-configured package matching your system:

- https://github.com/guilhemmarchand/nmon-logger/tree/master/deb for deb packages
- https://github.com/guilhemmarchand/nmon-logger/tree/master/rpm for rpm packages

Manual deployment

Deploying manually must be achieve the following way:

If not existing, create a system account for the non privilege "nmon" user:

```
useradd -r -m -c "system account for nmon-logger" nmon
```

Copy each file and directory to its destination by respecting the files and directories structure from the package

Package content description:

```
###################################
### Content:                    ###
###################################

### nmon-logger-rsyslog: ###

etc/
    cron.d/nmon-logger
    logrotate.d/nmon-logger
    nmon-logger/
        bin/(various)
        default/nmon.conf
    rsyslog.d/20-nmon-logger.conf
```

Set correct permissions for each piece of the package:

Execute these commands as root:

```
mkdir /var/log/nmon-logger; chown nmon:nmon /var/log/nmon-logger; chmod 0755 /var/log/
↪nmon-logger

chown -R nmon:nmon /etc/nmon-logger; chmod -R 0755 /etc/nmon-logger

chown root:root /etc/cron.d/nmon-logger; chmod 0644 /etc/cron.d/nmon-logger

chown root:root /etc/logrotate.d/nmon-logger; chmod 0644 /etc/logrotate.d/nmon-logger

chown root:root /etc/rsyslog.d/20-nmon-logger.conf; chmod 0644 /etc/rsyslog.d/20-nmon-
↪logger.conf
```

OPTIONAL : Verification on end server(s)

For trouble shooting or verification purposes, you may want to verify that things are working fine on the server where the nmon-logger has been deployed.

Nmon processes:

After a few minutes upon the deployment, a new nmon process must be running:

```
root@syslog-client:/var/log/nmon-logger# ps -ef | grep nmon
nmon      7029     1  0 22:07 ?        00:00:00 /etc/nmon-logger/bin/linux/generic/
↪nmon_linux_x86_64 -f -T -d 1500 -s 60 -c 120 -p
```

Various log will be generated about nmon data management:

```
root@syslog-client:/var/log/nmon-logger# ls -ltr /var/log/nmon-logger/
total 156
-rw-rw-r-- 1 nmon nmon   3441 janv. 26 21:15 nmon_clean.log
drwxrwxr-x 6 nmon nmon   4096 janv. 27 22:07 var
-rw-rw-r-- 1 nmon nmon  18719 janv. 27 22:10 nmon_collect.log
-rw-rw-r-- 1 nmon nmon 122781 janv. 27 22:10 nmon_processing.log
```

And Nmon Performance data:

```
root@syslog-client:/var/log/nmon-logger# ls -ltr /var/log/nmon-logger/var/*
-rw-rw-r-- 1 nmon nmon     5 janv. 27 22:07 /var/log/nmon-logger/var/nmon.pid

/var/log/nmon-logger/var/config_repository:

total 112
-rw-rw-r-- 1 nmon nmon 111509 janv. 27 22:07 nmon_configdata.log

/var/log/nmon-logger/var/perf_repository:
total 6068
-rw-rw-r-- 1 nmon nmon 6206333 janv. 27 22:12 nmon_perfdata.log
```

Et voila !

OPTIONAL : Verifications on syslog collectors

On active rsyslog collectors, a directory with the name of the server will host Nmon logs:

```
root@syslog-central:~# ls -ltr /var/log/nmon_performance/*/*
-rw-r----- 1 syslog adm   670272 janv. 27 22:07 /var/log/nmon_performance/syslog-
↪client/nmon_config.log
-rw-r----- 1 syslog adm    74711 janv. 27 22:55 /var/log/nmon_performance/syslog-
↪client/nmon_clean.log
-rw-r----- 1 syslog adm   299929 janv. 27 22:56 /var/log/nmon_performance/syslog-
↪client/nmon_collect.log
-rw-r----- 1 syslog adm 35814228 janv. 27 22:56 /var/log/nmon_performance/syslog-
↪client/nmon_performance.log
-rw-r----- 1 syslog adm  2554165 janv. 27 22:56 /var/log/nmon_performance/syslog-
↪client/nmon_processing.log
```

STEP 4 : Splunk it !

The last step is getting the data indexed in Splunk:

Have Splunk forwarding data to your indexer(s) Deploy the TA-nmon to your instance

Create a local/inputs.conf to index Nmon Performance data, example:

```
# inputs.conf

[monitor:///var/log/nmon_performance/.../nmon_performance.log]
disabled = false
index = nmon
sourcetype = nmon_data:fromsyslog
source = perfdata:syslog

[monitor:///var/log/nmon-performance/.../nmon_config.log]
disabled = false
index = nmon
sourcetype = nmon_config:fromsyslog
source = configdata:syslog

[monitor:///var/log/nmon-performance/.../nmon_collect.log]
disabled = false
index = nmon
sourcetype = nmon_collect:fromsyslog
source = nmon_collect:syslog

[monitor:///var/log/nmon-performance/.../nmon_clean.log]
disabled = false
index = nmon
sourcetype = nmon_clean:fromsyslog
source = nmon_cleaner:syslog

[monitor:///var/log/nmon-performance/.../nmon_processing.log]
disabled = false
index = nmon
sourcetype = nmon_processing:fromsyslog
source = nmon_processing:syslog
# Wait additional time to avoid incorrect event breaking
multiline_event_extra_waittime = true
```

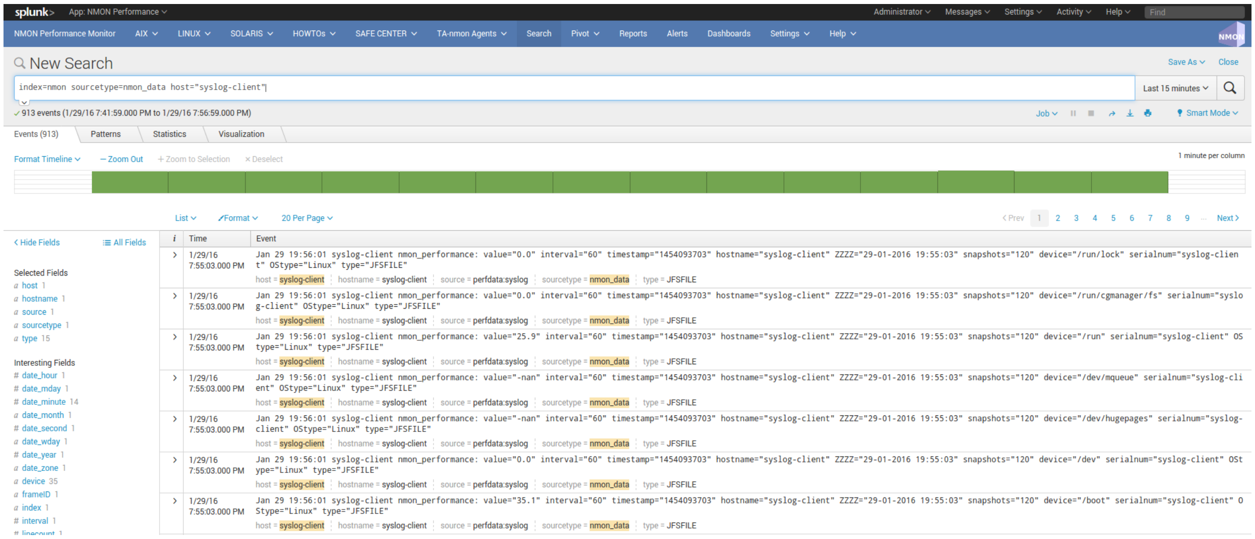
Restart Splunk

Et voilà !

If everything is fine in your configuration, you should start to receive incoming data in Nmon Performance monitor application.

OPTIONAL : Check your work !

Running a search over the hostname of the end server:

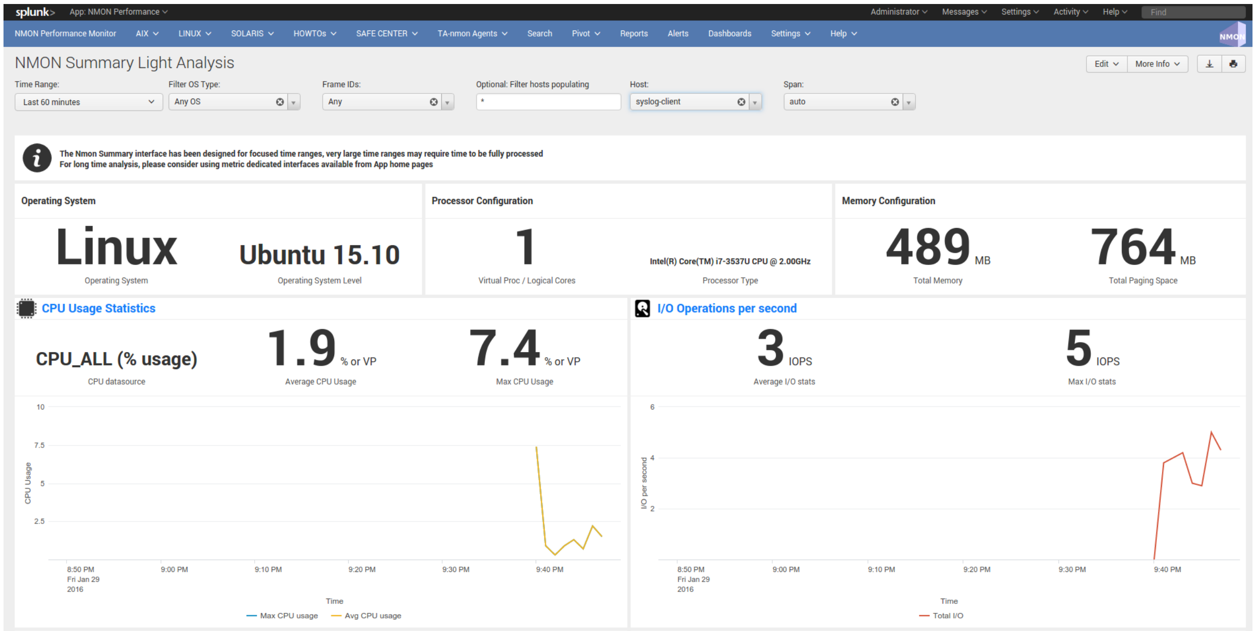


```
index=nmon sourcetype=nmon_data host="syslog-client"
```

✓ 913 events (1/29/16 7:41:59.000 PM to 1/29/16 7:56:59.000 PM)

```
Time                 Event
1/29/16              Jan 29 19:56:01 syslog-client nmon_performance: value="0.0" interval="60" timestamp="1454093703" hostname="syslog-client" ZZZZ="29-01-2016 19:55:03" snapshots="120" device="/run/lock" serialnum="syslog-clien
7:55:03.000 PM       t" OStype="Linux" type="JFSFILE"
                     host = syslog-client   hostname = syslog-client   source = perfdata:syslog   sourcetype = nmon_data   type = JFSFILE
1/29/16              Jan 29 19:56:01 syslog-client nmon_performance: value="0.0" interval="60" timestamp="1454093703" hostname="syslog-client" ZZZZ="29-01-2016 19:55:03" snapshots="120" device="/run/cgmanager/fs" serialnum="syslo
7:55:03.000 PM       g-client" OStype="Linux" type="JFSFILE"
                     host = syslog-client   hostname = syslog-client   source = perfdata:syslog   sourcetype = nmon_data   type = JFSFILE
1/29/16              Jan 29 19:56:01 syslog-client nmon_performance: value="25.9" interval="60" timestamp="1454093703" hostname="syslog-client" ZZZZ="29-01-2016 19:55:03" snapshots="120" device="/run" serialnum="syslog-client" OS
7:55:03.000 PM       type="Linux" type="JFSFILE"
                     host = syslog-client   hostname = syslog-client   source = perfdata:syslog   sourcetype = nmon_data   type = JFSFILE
1/29/16              Jan 29 19:56:01 syslog-client nmon_performance: value="-nan" interval="60" timestamp="1454093703" hostname="syslog-client" ZZZZ="29-01-2016 19:55:03" snapshots="120" device="/dev/mqueue" serialnum="syslog-cli
7:55:03.000 PM       ent" OStype="Linux" type="JFSFILE"
                     host = syslog-client   hostname = syslog-client   source = perfdata:syslog   sourcetype = nmon_data   type = JFSFILE
1/29/16              Jan 29 19:56:01 syslog-client nmon_performance: value="-nan" interval="60" timestamp="1454093703" hostname="syslog-client" ZZZZ="29-01-2016 19:55:03" snapshots="120" device="/dev/hugepages" serialnum="syslog-
7:55:03.000 PM       client" OStype="Linux" type="JFSFILE"
                     host = syslog-client   hostname = syslog-client   source = perfdata:syslog   sourcetype = nmon_data   type = JFSFILE
1/29/16              Jan 29 19:56:01 syslog-client nmon_performance: value="0.0" interval="60" timestamp="1454093703" hostname="syslog-client" ZZZZ="29-01-2016 19:55:03" snapshots="120" device="/dev" serialnum="syslog-client" OSt
7:55:03.000 PM       ype="Linux" type="JFSFILE"
                     host = syslog-client   hostname = syslog-client   source = perfdata:syslog   sourcetype = nmon_data   type = JFSFILE
1/29/16              Jan 29 19:56:01 syslog-client nmon_performance: value="35.1" interval="60" timestamp="1454093703" hostname="syslog-client" ZZZZ="29-01-2016 19:55:03" snapshots="120" device="/boot" serialnum="syslog-client" O
7:55:03.000 PM       Stype="Linux" type="JFSFILE"
                     host = syslog-client   hostname = syslog-client   source = perfdata:syslog   sourcetype = nmon_data   type = JFSFILE
```

Interface example:



splunk> App: NMON Performance
Administrator Messages Settings Activity Help Find
NMON Performance Monitor AIX LINUX SOLARIS HOWTOs SAFE CENTER TA-nmon Agents Search Pivot Reports Alerts Dashboards Settings Help
Edit
More Info
10
7.5
5
2.5
CPU Usage
8:50 PM
Fri Jan 29
2016
9:00 PM
9:10 PM
9:20 PM
9:30 PM
9:40 PM
Time
Max CPU usage
Avg CPU usage
6
4
2
I/O per second
8:50 PM
Fri Jan 29
2016
9:00 PM
9:10 PM
9:20 PM
9:30 PM
9:40 PM
Time
Total I/O

NMON Summary Light Analysis

Time Range: Last 60 minutes
Filter OS Type: Any OS
Frame IDs: Any
Optional: Filter hosts populating: *
Host: syslog-client
Span: auto

The Nmon Summary interface has been designed for focused time ranges, very large time ranges may require time to be fully processed
For long time analysis, please consider using metric dedicated interfaces available from App home pages

Operating System
Linux — Operating System
Ubuntu 15.10 — Operating System Level

Processor Configuration
1 — Virtual Proc / Logical Cores
Intel(R) Core(TM) i7-3537U CPU @ 2.00GHz — Processor Type

Memory Configuration
489 MB — Total Memory
764 MB — Total Paging Space

CPU Usage Statistics
CPU_ALL (% usage) — CPU datasource
1.9 % or VP — Average CPU Usage
7.4 % or VP — Max CPU Usage

I/O Operations per second
3 IOPS — Average I/O stats
5 IOPS — Max I/O stats

2.15 syslog-ng / nmon-logger deployment

Syslog deployment topology - Generate and forward Nmon performance data from syslog-ng clients to your centralized syslog-ng servers

Introduced with the Nmon Performance Monitor 1.6.14, you can now get real time Nmon data from end servers even without any Splunk Universal Forwarders deployment.

This will be achieved using:

- The nmon-logger package to be deployed on servers: https://github.com/guilhemmarchand/nmon-logger
- syslog-ng locally available on servers and configured to send data to central syslog servers
- Splunk Universal or Heavy Forwarder instance installed in syslog-ng servers (collector or additional relays) that will monitor and send data in a "key=value" format to Splunk
- Deploying the nmon-logger package to your end-servers, the package is now provided in **rpm** and **deb** packages

Optionally, a deployment tool manager like (Ansible, Chef. . .) is recommended, note that Playbooks for Ansible are provided with the nmon-logger package

Please review requirements in the above section.

Fast testing using Vagrant and Ansible:

If you are interested in a very fast and automated way to test the Nmon Performance Application with an rsyslog deployment, checkout the provided configuration using the excellent Vagrant (https://www.vagrantup.com/) and Ansible configuration management (http://docs.ansible.com/ansible/index.html)

- Checkout: https://github.com/guilhemmarchand/nmon-logger/tree/master/vagrant-ansible-demo-syslog-ng

In about 5 minutes, have a running and automated deployment working !

Key concepts of an syslog-ng deployment:

Why an syslog topology versus Universal Forwarders deployment ?:

- At first, this provides a powerful and resilient alternative way to deploy the Nmon Performance monitor
- Syslog is a Unix / Linux standard, and available on many servers
- Because you may already have an syslog-ng centralization available and you do not want to deploy any additional software on servers
- Because sometimes companies don't want to rely on proprietary software on end servers, deploying Universal Forwarders is not an option

Key concepts :

- 100% of Application features over a traditional Universal Forwarders deployment
- In a standard deployment of the TA-nmon and the Nmon Performance application, hosts generates Nmon Performance in csv structured data
- To be transported over syslog, csv data are being transformed in a key=value format

- While csv structured data is known in Splunk as "an indexing field structure", key=value format will be extracted on the fly (parsing at extraction time)
- csv structured data has an higher disk space cost but offers a very low level of volume of data to index (and so a low level of licence cost), and offers best performances at search time in SPL (Search Language Processing)
- key=value format generates an higher volume of data and requires more power at extraction time, but it does not generate indexed fields, this also represents less disk space
- Nmon Performance massively uses data model acceleration, key=value searching performance versus csv structured data will not be different from standard deployment

Can i send data directory from syslog to Splunk ?

Yes it is possible, but this is not the deployment topology i would recommend for multiple reasons, as exposed above:

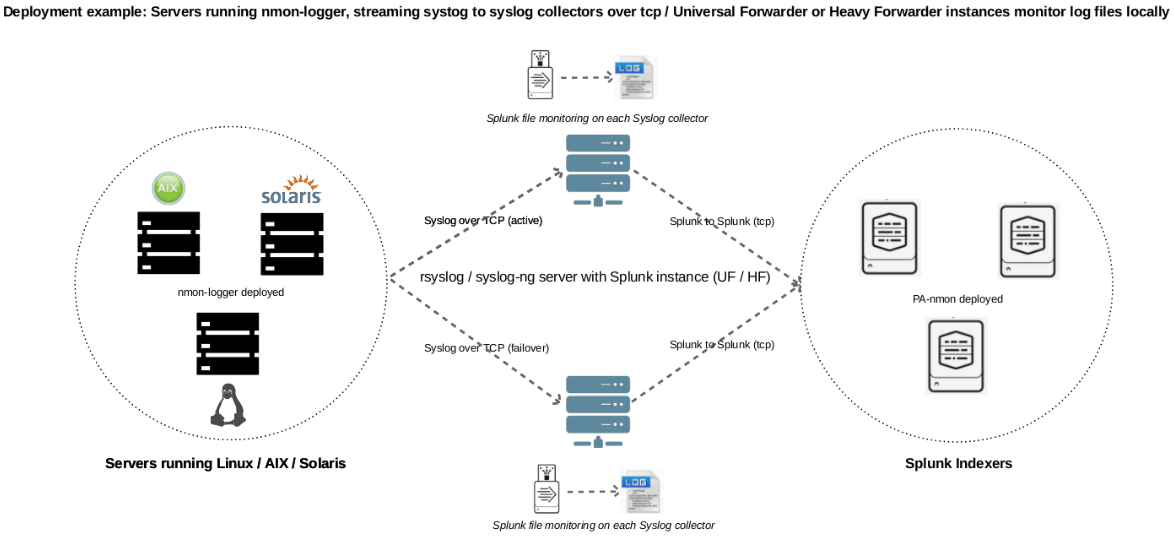
- You cannot guarantee that syslog-ng will send data in the order it should, this is especially true with multi-line events that could be mixed between hosts
- syslog-ng can easily identify the host origin of the incoming data and generate per host files, which guarantees management of Nmon data (like large multi-line configuration data)
- For resilient reasons, once it is configured, you will few often restart syslog-ng. (most often while rebooting the machine)
- This is less true with Splunk as you will want to upgrade it from time to time, or deploy new configuration or application to manage new data
- Finally syslog speaks to syslog, with the same native implementation and features

What about the nmon-logger deployment management ?:

- In standard Universal Forwarder management, you can easily rely on native Splunk deployment servers to push and maintain the TA-nmon package
- In the syslog deployment, you will deploy nmon-logger packages (rpm, deb) or deploy nmon-logger manually

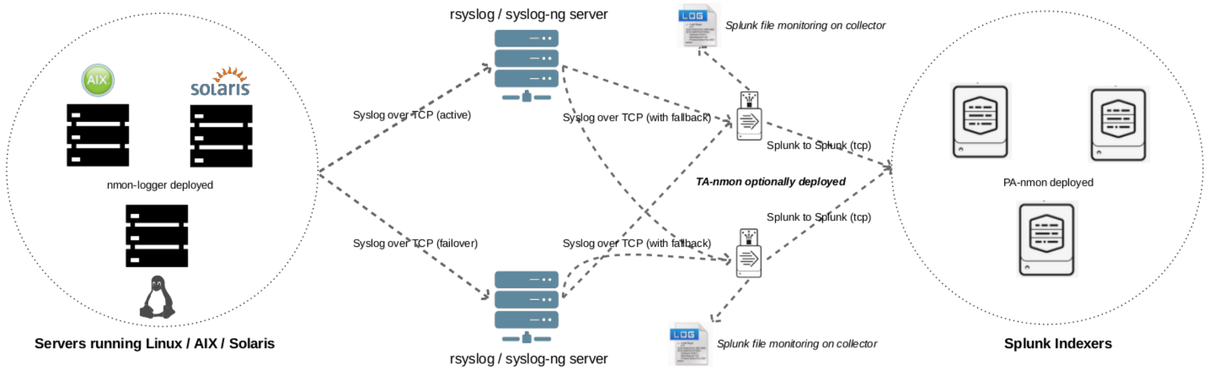
Topology: Examples of possible implementations:

Example 1: Splunk Universal or Heavy forwarder installed on main syslog-ng collectors:


Deployment example: Servers running nmon-logger, streaming systog to syslog collectors over tcp / Universal Forwarder or Heavy Forwarder instances monitor log files locally
Splunk file monitoring on each Syslog collector
AIX
solaris
nmon-logger deployed
Servers running Linux / AIX / Solaris
Syslog over TCP (active)
Syslog over TCP (failover)
rsyslog / syslog-ng server with Splunk instance (UF / HF)
Splunk to Splunk (tcp)
Splunk to Splunk (tcp)
PA-nmon deployed
Splunk Indexers
Splunk file monitoring on each Syslog collector

Example 2: Splunk Universal or Heavy forwarder installed third party servers running syslog-ng:

Deployment example: Servers running nmon-logger, streaming systog to syslog collectors over tcp, Universal Forwarder or Heavy Forwarder instances monitors log files locally


AIX
solaris
nmon-logger deployed
Servers running Linux / AIX / Solaris
rsyslog / syslog-ng server
Syslog over TCP (active)
Syslog over TCP (failover)
rsyslog / syslog-ng server
Syslog over TCP (with fallback)
Syslog over TCP (with fallback)
LOG
Splunk file monitoring on collector
Splunk to Splunk (tcp)
TA-nmon optionally deployed
Splunk to Splunk (tcp)
LOG
Splunk file monitoring on collector
PA-nmon deployed
Splunk Indexers

2.15.1 PRE-REQUISITES:

- Splunk + Nmon Performance app:

First of all, have a Splunk working installation, and the Nmon Performance up and running ! (yeah, songs like an evidence :-)

Some specific requirements must be respected to achieve a deployment that uses syslog-ng as the transport layer:

- SYSLOG-NG (V3.x minimal recommended):

Syslog-ng is required to forward Nmon Performance data to centralized syslog-ng servers, see:

https://syslog-ng.org/

- Python 2.7.x **OR** Perl with the module Time::HiRes:

The nmon-logger will by default search for a Python 2.7.x environment. If it is not available, scripts will use Perl, when using Perl note that the Time::HiRes module is required.

2.15.2 STEP 1 : Syslog-ng configuration for central collectors:

A minimal configuration is required on Syslog collectors, this will make syslog-ng to listen on a dedicated TCP port to receive incoming data from end servers.

In the following example, syslog-ng will listen to the TCP / 514 port:

Let's create a central configuration that will both log remote hosts messages and nmon-logger data (without duplicating them in both locations):

/etc/syslog-ng/conf.d/central.conf

```
# syslog-ng configuration for central logging

options {
        time-reap(30);
        mark-freq(10);
        keep-hostname(yes);
        create-dirs(yes);
};

```

(continues on next page)

(continued from previous page)

```
source s_tcp {
        tcp(port(514));
};

destination d_host-specific {
        file("/var/log/remote-hosts/$HOST/$HOST.log");
};

log {
       source(s_tcp);
       filter(f_nmon_performance); destination(d_nmon_performance); flags(final);
};

log {
       source(s_tcp);
       filter(f_nmon_config); destination(d_nmon_config); flags(final);
};

log {
       source(s_tcp);
       filter(f_nmon_collect); destination(d_nmon_collect); flags(final);
};

log {
       source(s_tcp);
       filter(f_nmon_processing); destination(d_nmon_processing); flags(final);
};

log {
       source(s_tcp);
       filter(f_nmon_clean); destination(d_nmon_clean); flags(final);
};

log {
        source(s_tcp);
        destination(d_host-specific);
};
```

Now create the nmon-logger configuration file:

```
*/etc/syslog-ng/conf.d/nmon-logger.conf*

# nmon-logger.conf

# Generic options
options {
        keep-hostname(yes);
        create-dirs(yes);
};

# setup destination for Nmon performance data
destination d_nmon_performance {
        file("/var/log/nmon-performance/$HOST/nmon_performance.log" );
};
destination d_nmon_config {
        file("/var/log/nmon-performance/$HOST/nmon_config.log" );
};
```

(continues on next page)

(continued from previous page)

```
destination d_nmon_collect {
        file("/var/log/nmon-performance/$HOST/nmon_collect.log" );
};
destination d_nmon_processing {
        file("/var/log/nmon-performance/$HOST/nmon_processing.log" );
};
destination d_nmon_clean {
        file("/var/log/nmon-performance/$HOST/nmon_clean.log" );
};

# filter all messages, on the "program" field.
filter f_nmon_performance {
        program("nmon_performance");
};
filter f_nmon_config {
        program("nmon_config");
};
filter f_nmon_collect {
        program("nmon_collect");
};
filter f_nmon_processing {
        program("nmon_processing");
};

filter f_nmon_clean {
        program("nmon_clean");
};
```

Restart syslog-ng:

```
sudo service syslog-ng restart
```

2.15.3 STEP 2 : syslog-ng configuration for end servers:

Each of your end servers must be configured to send its syslog data to the central syslog-ng server.

Create the central client configuration that forwards local log to central servers:

/etc/syslog-ng/conf.d/client.conf

```
# Client configuration for central logging
# log all syslog messages to remote syslog-ng server

destination d_net { tcp("syslog-ng-central" port(514) log_fifo_size(1000)); };
log { source(s_src); destination(d_net); };
```

IMPORTANT: syslog-ng does not natively support fail over mechanism, such mechanism must be operating on Operating system level (OS cluster) or using third party software such as HA-proxy.

Restart syslog-ng:

```
sudo service syslog-ng restart
```

Immediately after the restart, syslog-ng starts to forward data to central syslog-ng server.

2.15.4 STEP 3 : Deploy the nmon-logger to your end servers

On each end server, you must deploy the "nmon-logger" package:

https://github.com/guilhemmarchand/nmon-logger

Using your package manager

For compatible operating systems using the "deb" Debian package manager (Debian, Ubuntu. . .) and the "rpm" Redhat package manager (CentOS, RHEL. . .) you can easily deploy the pre-configured package matching your system:

- https://github.com/guilhemmarchand/nmon-logger/tree/master/deb for deb packages
- https://github.com/guilhemmarchand/nmon-logger/tree/master/rpm for rpm packages

Manual deployment

Ansible Playbooks are available in the Git repository, with Ansible the nmon-logger package is being totally deployed, up and running in a few seconds !!!

Deploying manually must be achieve the following way:

- If not existing, create a system account for the non privilege "nmon" user:

```
useradd -r -m -c "system account for nmon-logger" nmon
```

- Copy each file and directory to its destination by respecting the files and directories structure from the package

Package content description:

```
###################################
### Content:                    ###
###################################

### nmon-logger-syslog-ng: ###

etc/
    cron.d/nmon-logger
    logrotate.d/nmon-logger
    nmon-logger/
                bin/(various)
                default/nmon.conf
    syslog-ng/conf.d/nmon-logger.conf
```

- Set correct permissions for each piece of the package:

Execute these commands as root:

```
mkdir /var/log/nmon-logger; chown nmon:nmon /var/log/nmon-logger; chmod 0755 /var/log/
↪nmon-logger

chown -R nmon:nmon /etc/nmon-logger; chmod -R 0755 /etc/nmon-logger

chown root:root /etc/cron.d/nmon-logger; chmod 0644 /etc/cron.d/nmon-logger

chown root:root /etc/logrotate.d/nmon-logger; chmod 0644 /etc/logrotate.d/nmon-logger

chown root:root /etc/syslog-ng/conf.d/nmon-logger.conf; chmod 0644 /etc/syslog-ng/
↪conf.d/nmon-logger.conf
```

(continues on next page)

(continued from previous page)

2.15.5 OPTIONAL : Verification on end server(s)

For trouble shooting or verification purposes, you may want to verify that things are working fine on the server where the nmon-logger has been deployed.

Nmon processes:

After a few minutes upon the deployment, a new nmon process must be running:

```
root@syslog-client:/var/log/nmon-logger# ps -ef | grep nmon
nmon      7029     1  0 22:07 ?        00:00:00 /etc/nmon-logger/bin/linux/generic/
↪nmon_linux_x86_64 -f -T -d 1500 -s 60 -c 120 -p
```

Various log will be generated about nmon data management:

```
root@syslog-client:/var/log/nmon-logger# ls -ltr /var/log/nmon-logger/
total 156
-rw-rw-r-- 1 nmon nmon   3441 janv. 26 21:15 nmon_clean.log
drwxrwxr-x 6 nmon nmon   4096 janv. 27 22:07 var
-rw-rw-r-- 1 nmon nmon  18719 janv. 27 22:10 nmon_collect.log
-rw-rw-r-- 1 nmon nmon 122781 janv. 27 22:10 nmon_processing.log
```

And Nmon Performance data:

```
root@syslog-client:/var/log/nmon-logger# ls -ltr /var/log/nmon-logger/var/*
-rw-rw-r-- 1 nmon nmon    5 janv. 27 22:07 /var/log/nmon-logger/var/nmon.pid

/var/log/nmon-logger/var/config_repository:
total 112
-rw-rw-r-- 1 nmon nmon 111509 janv. 27 22:07 nmon_configdata.log

/var/log/nmon-logger/var/perf_repository:
total 6068
-rw-rw-r-- 1 nmon nmon 6206333 janv. 27 22:12 nmon_perfdata.log
```

Et voila !

2.15.6 OPTIONAL : Verifications on syslog-ng collector(s)

On syslog-ng collector(s), a directory with the name of the server will host Nmon logs:

```
root@syslog-ng-central:~# ls -ltr /var/log/nmon_performance/syslog-ng-client/
total 1960
-rw-r----- 1 root adm   35220 janv. 30 12:54 nmon_config.log
-rw-r----- 1 root adm    5604 janv. 30 13:50 nmon_clean.log
-rw-r----- 1 root adm   23343 janv. 30 13:53 nmon_collect.log
-rw-r----- 1 root adm  193058 janv. 30 13:53 nmon_processing.log
-rw-r----- 1 root adm 1724814 janv. 30 13:53 nmon_performance.log
```

2.15.7 STEP 4 : Splunk it !

The last step is getting the data indexed in Splunk:

- Have Splunk forwarding data to your indexer(s)
- Deploy the TA-nmon to your instance
- Create a local/inputs.conf to index Nmon Performance data, example:

```
# inputs.conf

[monitor:///var/log/nmon-performance/.../nmon_performance.log]
disabled = false
index = nmon
sourcetype = nmon_data:fromsyslog
source = perfdata:syslog

[monitor:///var/log/nmon-performance/.../nmon_config.log]
disabled = false
index = nmon
sourcetype = nmon_config:fromsyslog
source = configdata:syslog

[monitor:///var/log/nmon-performance/.../nmon_collect.log]
disabled = false
index = nmon
sourcetype = nmon_collect:fromsyslog
source = nmon_collect:syslog

[monitor:///var/log/nmon-performance/.../nmon_clean.log]
disabled = false
index = nmon
sourcetype = nmon_clean:fromsyslog
source = nmon_cleaner:syslog

[monitor:///var/log/nmon-performance/.../nmon_processing.log]
disabled = false
index = nmon
sourcetype = nmon_processing:fromsyslog
source = nmon_processing:syslog
# Wait additional time to avoid incorrect event breaking
multiline_event_extra_waittime = true
```

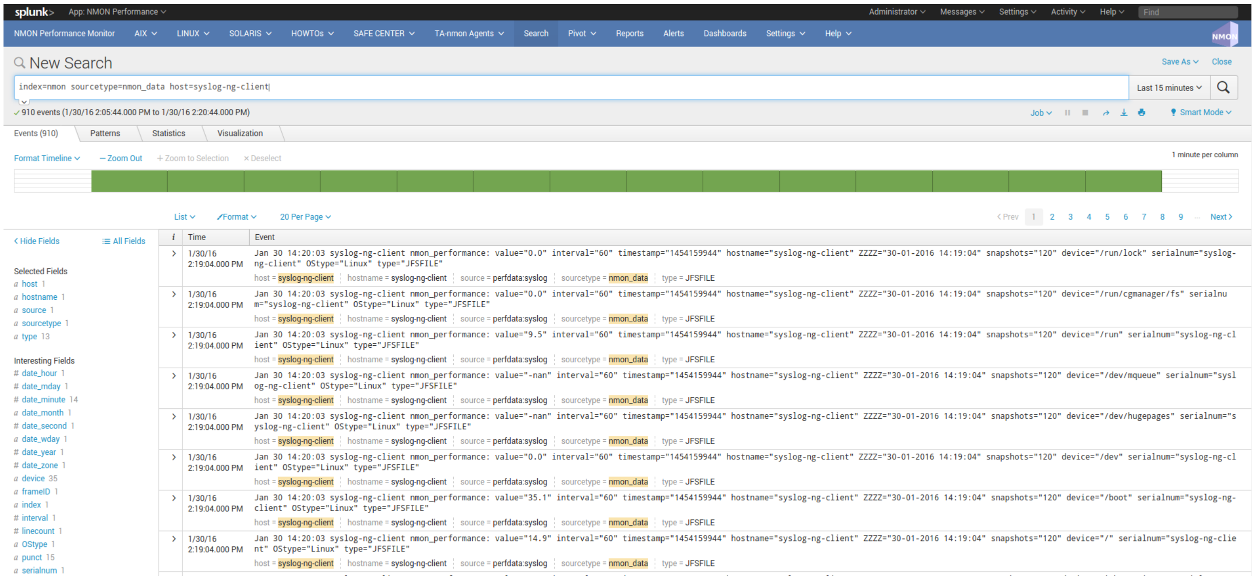
Restart Splunk

Et voilà !

If everything is fine in your configuration, you should start to receive incoming data in Nmon Performance monitor application.

2.15.8 OPTIONAL : Check your work !

Running a search over the hostname of the end server:

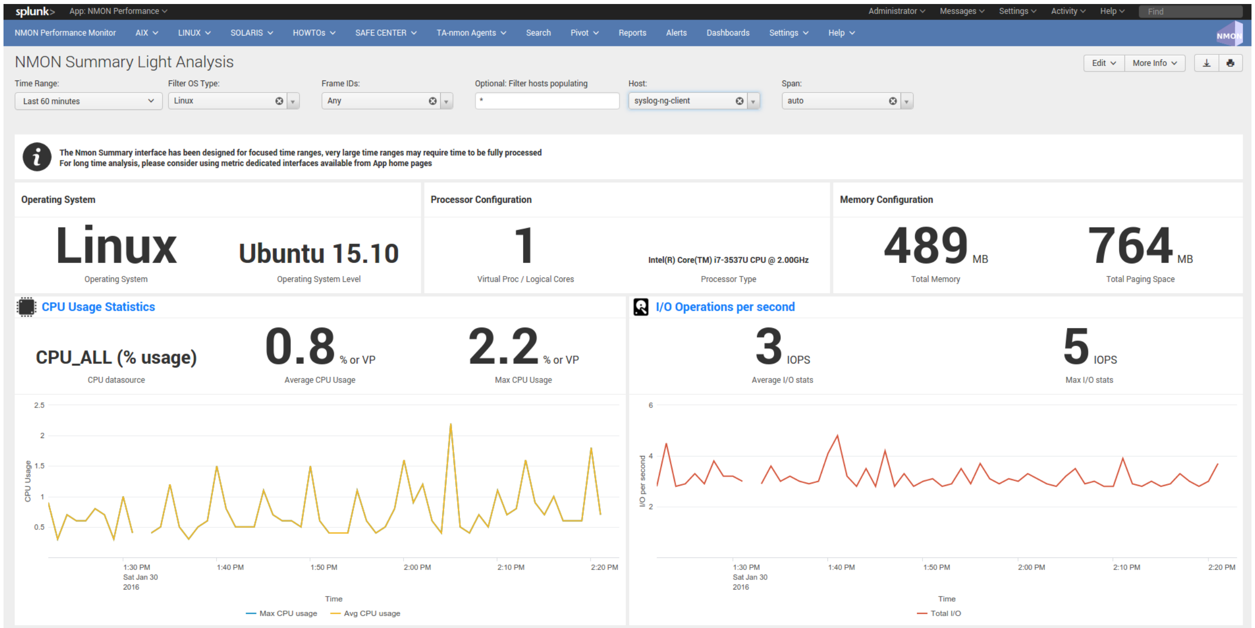


```
index=nmon sourcetype=nmon_data host=syslog-ng-client
```

✓ 910 events (1/30/16 2:05:44.000 PM to 1/30/16 2:20:44.000 PM)

```
i  Time                      Event
>  1/30/16 2:19:04.000 PM    Jan 30 14:20:03 syslog-ng-client nmon_performance: value="0.0" interval="60" timestamp="1454159944" hostname="syslog-ng-client" ZZZZ="30-01-2016 14:19:04" snapshots="120" device="/run/lock" serialnum="syslog-ng-client" OStype="Linux" type="JFSFILE"
                             host = syslog-ng-client  hostname = syslog-ng-client  source = perfdata:syslog  sourcetype = nmon_data  type = JFSFILE
>  1/30/16 2:19:04.000 PM    Jan 30 14:20:03 syslog-ng-client nmon_performance: value="0.0" interval="60" timestamp="1454159944" hostname="syslog-ng-client" ZZZZ="30-01-2016 14:19:04" snapshots="120" device="/run/cgmanager/fs" serialnum="syslog-ng-client" OStype="Linux" type="JFSFILE"
                             host = syslog-ng-client  hostname = syslog-ng-client  source = perfdata:syslog  sourcetype = nmon_data  type = JFSFILE
>  1/30/16 2:19:04.000 PM    Jan 30 14:20:03 syslog-ng-client nmon_performance: value="9.5" interval="60" timestamp="1454159944" hostname="syslog-ng-client" ZZZZ="30-01-2016 14:19:04" snapshots="120" device="/run" serialnum="syslog-ng-client" OStype="Linux" type="JFSFILE"
                             host = syslog-ng-client  hostname = syslog-ng-client  source = perfdata:syslog  sourcetype = nmon_data  type = JFSFILE
>  1/30/16 2:19:04.000 PM    Jan 30 14:20:03 syslog-ng-client nmon_performance: value="-nan" interval="60" timestamp="1454159944" hostname="syslog-ng-client" ZZZZ="30-01-2016 14:19:04" snapshots="120" device="/dev/mqueue" serialnum="syslog-ng-client" OStype="Linux" type="JFSFILE"
                             host = syslog-ng-client  hostname = syslog-ng-client  source = perfdata:syslog  sourcetype = nmon_data  type = JFSFILE
>  1/30/16 2:19:04.000 PM    Jan 30 14:20:03 syslog-ng-client nmon_performance: value="-nan" interval="60" timestamp="1454159944" hostname="syslog-ng-client" ZZZZ="30-01-2016 14:19:04" snapshots="120" device="/dev/hugepages" serialnum="syslog-ng-client" OStype="Linux" type="JFSFILE"
                             host = syslog-ng-client  hostname = syslog-ng-client  source = perfdata:syslog  sourcetype = nmon_data  type = JFSFILE
>  1/30/16 2:19:04.000 PM    Jan 30 14:20:03 syslog-ng-client nmon_performance: value="0.0" interval="60" timestamp="1454159944" hostname="syslog-ng-client" ZZZZ="30-01-2016 14:19:04" snapshots="120" device="/dev" serialnum="syslog-ng-client" OStype="Linux" type="JFSFILE"
                             host = syslog-ng-client  hostname = syslog-ng-client  source = perfdata:syslog  sourcetype = nmon_data  type = JFSFILE
>  1/30/16 2:19:04.000 PM    Jan 30 14:20:03 syslog-ng-client nmon_performance: value="35.1" interval="60" timestamp="1454159944" hostname="syslog-ng-client" ZZZZ="30-01-2016 14:19:04" snapshots="120" device="/boot" serialnum="syslog-ng-client" OStype="Linux" type="JFSFILE"
                             host = syslog-ng-client  hostname = syslog-ng-client  source = perfdata:syslog  sourcetype = nmon_data  type = JFSFILE
>  1/30/16 2:19:04.000 PM    Jan 30 14:20:03 syslog-ng-client nmon_performance: value="14.9" interval="60" timestamp="1454159944" hostname="syslog-ng-client" ZZZZ="30-01-2016 14:19:04" snapshots="120" device="/" serialnum="syslog-ng-client" OStype="Linux" type="JFSFILE"
                             host = syslog-ng-client  hostname = syslog-ng-client  source = perfdata:syslog  sourcetype = nmon_data  type = JFSFILE
```

Interface example:

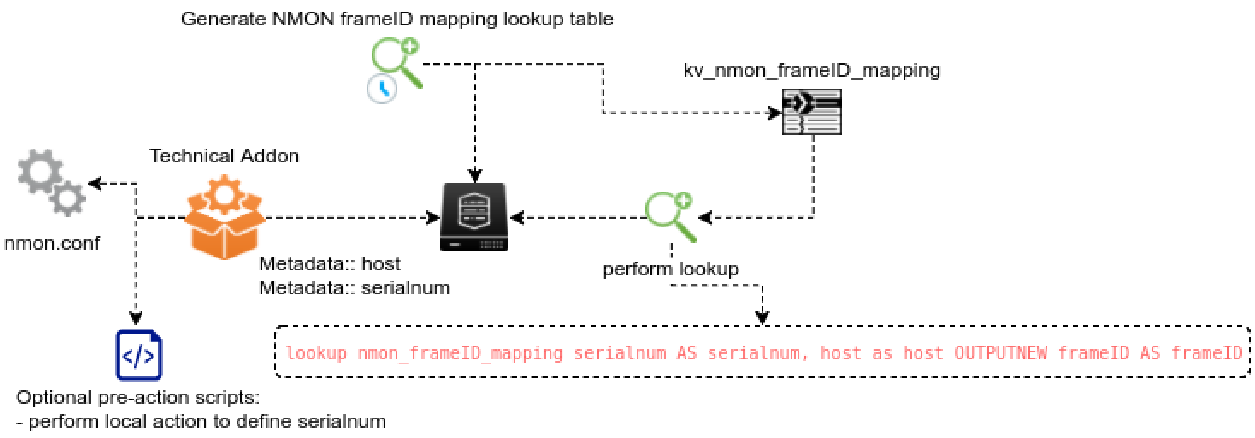


Time Range: Last 60 minutes
Filter OS Type: Linux
Frame IDs: Any
Optional: Filter hosts populating: *
Host: syslog-ng-client
Span: auto

The Nmon Summary interface has been designed for focused time ranges, very large time ranges may require time to be fully processed
For long time analysis, please consider using metric dedicated interfaces available from App home pages

2.16 frameID mapping management

frameID mapping overview:


Generate NMON frameID mapping lookup table
kv_nmon_frameID_mapping
Technical Addon
nmon.conf
Metadata:: host
Metadata:: serialnum
perform lookup
Optional pre-action scripts:
- perform local action to define serialnum

```
lookup nmon_frameID_mapping serialnum AS serialnum, host as host OUTPUTNEW frameID AS frameID
```

The "frameID" feature is used within the application to programmatically link servers with a logical identifier, aka the frame identifier.

The frameID mapping is first natively achieved against the serial number value retrieved from Nmon data, plus several other options which are exposed in this documentation.

Using frameID allows:

- filtering for servers more effectively within any human interface of the application
- managing alerting thresholds and exclusions based on the frameID template
- for analytic purposes operated against the frameID and server associations, like performing capacity planning analytics using this logical grouping

The frameID value can be configured by several options:

- at raw level using default values retrieved by Nmon, valuable for AIX operating systems only (the serial number will be the serial number from the frame hosting the partition)
- at raw level using static definition in nmon.conf
- at raw level using dynamic definition in nmon.conf configured by a pre-action script
- at search time level using the frameID mapping KVstore collection and Splunk lookup feature

All these options are described in detail within the present documentation.

2.16.1 frameID mapping lookup table generation

First, The application has a scheduled report called "Generate NMON frameID mapping lookup table".

The purpose of this scheduled report is generating and updating the KVstore collection that defines the frameID mapping.

The report is scheduled by default to run every hour against the last 7 seven days of raw data using the high performance **mcatalog** command.

Due to its high level of optimisation, its cost in term of Splunk resources is negligible.

2.16.2 Default frameID value

Without any kind of configuration, the default value for the serial number (serialnum) used automatically to define the frameID field value will be:

- for AIX: serial number of the frame hosting the partition at Nmon binary startup
- for Linux: equal to the hostname value
- for Solaris: equal to the hostname value

As such, the default frameID mapping is valuable essentially for AIX systems, if your using Linux OS and/or Solaris OS, it is highly recommended to perform your own mapping such that you can get benefits from the feature.

2.16.3 frameID value at raw level with nmon.conf and pre-action scripts

As exposed in nmon.conf, you can statically configure the frameID mapping using a custom option in the nmon.conf configuration file.

This feature is driven by:

```
#####################
# frameID definition:
#####################

# The frameID definition is an enrichment mechanism used within the application to associate a given host with a given frame identifier
# By default, the mapping is operated against the value of "serialnum" which is defined at the raw level by nmon binaries

# On AIX systems, the serialnum value is equal to the serial number of the frame hosting the partition
# On Linux and Solaris systems, the serialnum is equal to the value of the hostname

# Using this option allows you to override the serialnum value by a static value defined in the nmon.conf configuration file
# nmon.conf precedence allows defining the serialnum value on per deployment basis (local/nmon.conf) or on a per server basis (/etc/nmon.conf)

# default is:
# override_sys_serialnum="0"
# which lets nmon set the serialnum value

# Set this value to:
# override_sys_serialnum="1"
# to activate the serialnum override based on the value defined in:

# override_sys_serialnum_value="<sting>"
# Acceptable values for <string> are letters (lower and upper case), numbers and "-" / "_"

override_sys_serialnum="0"
override_sys_serialnum_value="none"
```

Most likely, this feature is interesting being used in conjonction with a pre-action script.

pre-action scripts

A pre-action script is literally nothing more than a home made script that can be set and run automatically by the Technical Addon, and specially run by the nmon_helper.sh script.

pre-actions scripts should be stored in:

```
TA-nmon/bin/pre_action_scripts/
```

Any pre-action script deployed must be named with ".sh" extension and must be executable by the Unix user owning the Splunk processes.

A pre-action script could be defined to set the frameID based on:

- A conditional operation based of the server hostname if you have a naming convention that can be used
- Any command that run on the server to retrieve the information to be used as the serial number value

Review the TA-nmon/bin/pre_action_scripts/README for example and more information.

2.16.4 frameID mapping at search time with the KVstore collection

A KVstore collection and its associated lookup are available to define the frameID mapping at search time.

KVstore collection	Lookup name	Expected fields
kv_nmon_frameID_mapping	nmon_frameID_mapping	serialnum, frameID, host, host_description

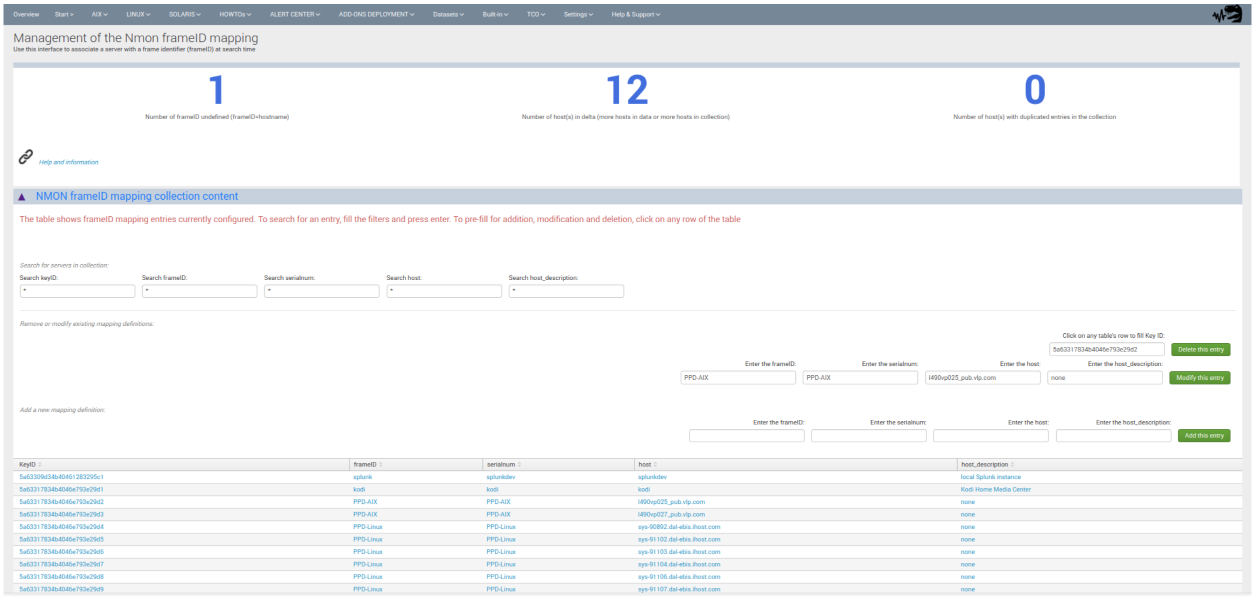
The following macro is being used to perform the frameID mapping at search time:

```
#########################################
# frameID mapping
#########################################

[mapping_frameID]
definition = lookup nmon_frameID_mapping host as host OUTPUT frameID\
| eval frameID=if(isnull(frameID), host, frameID)
iseval = 0
```

Finally, a configuration interface is provided to perform the mapping collection:

```
Menu Settings / FRAMEID mapping enrichment
```



Management of the Nmon frameID mapping

Use this interface to associate a server with a frame identifier (frameID) at search time

1

Number of frameID undefined (frameID=hostname)

12

Number of host(s) in delta (more hosts in data or more hosts in collection)

0

Number of host(s) with duplicated entries in the collection

▲ NMON frameID mapping collection content

The table shows frameID mapping entries currently configured. To search for an entry, fill the filters and press enter. To pre-fill for addition, modification and deletion, click on any row of the table

Search for servers in collection:

Search keyID: *

Search frameID: *

Search serialnum: *

Search host: *

Search host_description: *

Remove or modify existing mapping definitions:

Click on any table's row to fill Key ID: 5a63317834b4046e793e29d2

Enter the frameID: PPD-AIX

Enter the serialnum: PPD-AIX

Enter the host: l490vp025_pub.vlp.com

Enter the host_description: none

Add a new mapping definition:

Enter the frameID:

Enter the serialnum:

Enter the host:

Enter the host_description:

KeyID	frameID	serialnum	host	host_description
5a63309d34b40461283295c1	splunk	splunkdev	splunkdev	local Splunk instance
5a63317834b4046e793e29d1	kodi	kodi	kodi	Kodi Home Media Center
5a63317834b4046e793e29d2	PPD-AIX	PPD-AIX	l490vp025_pub.vlp.com	none
5a63317834b4046e793e29d3	PPD-AIX	PPD-AIX	l490vp027_pub.vlp.com	none
5a63317834b4046e793e29d4	PPD-Linux	PPD-Linux	sys-90892.dal-ebis.ihost.com	none
5a63317834b4046e793e29d5	PPD-Linux	PPD-Linux	sys-91102.dal-ebis.ihost.com	none
5a63317834b4046e793e29d6	PPD-Linux	PPD-Linux	sys-91103.dal-ebis.ihost.com	none
5a63317834b4046e793e29d7	PPD-Linux	PPD-Linux	sys-91104.dal-ebis.ihost.com	none
5a63317834b4046e793e29d8	PPD-Linux	PPD-Linux	sys-91106.dal-ebis.ihost.com	none
5a63317834b4046e793e29d9	PPD-Linux	PPD-Linux	sys-91107.dal-ebis.ihost.com	none

The mapping interface allows you to:

- View the current lookup table content
- Modify any entry and field values
- Add or delete any entry

Finally, all human interfaces provide selection filtering against the frameID, automatic lookup is achieved at search time for Nmon events data, and alerting thresholds and exceptions will use the frameID value to perform their tasks.

2.17 Userguide

2.17.1 Key concepts

The Nmon Performance application implements the nmon/sarmon binaries to generates rich and accurate performance data for your AIX, Linux and Solaris systems.

Keys concepts of the app can be summarize as the following:

- The nmon core application is deployed to the Splunk search head level
- The TA-nmon package is to be deployed to *nix clients running the Splunk Universal Forwarder or full Splunk instance
- On search head / standalone instances, the core app can generate the nmon data without having to deploy the TA-nmon
- When the nmon_helper input script starts, it attempts find the best suitable binary for your system or can fallback to locally nmon binary available
- Once the nmon binary process has been started, the data collection begins until the current process ends. (each nmon process has a time to live)
- Every time the nmon data gets updated, Splunk read the nmon files and calls the nmon2csv parsers, the data gets structured and indexed in Splunk
- Performance metrics once indexed are immediately available in Splunk for analysis
- Every time a new nmon process is started, new configuration data will be generated and indexed in Splunk
- The nmon_inventory data (stored in the nmon_inventory lookup table) is generated every hour using efficient data model, it is being used to enrich the performance data and provide inventory interfaces

2.17.2 Data Types (sourcetype)

sourcetype=nmon_data

The "**nmon_data**" sourcetype available in the **eventtype=nmon:performance** contains all the data related to performance metrics of your systems.

In the nmon:performance data, the "key" is the **type** field.

This field contains the monitor identifier that matches a category of metrics, such as "type=CPU_ALL". (global CPU usage in percentage)

sourcetype=nmon_config

The “**nmon_config**” sourcetype available in the **eventtype=nmon:config** contains all the data related to the configuration of your systems.

These are the AAA and BBB* sections of nmon raw data, generated during the nmon binary startup. The events are long multi-lines events stored per host, in default configuration these data will be extracted almost every 2 hours as the data will not change unless a new nmon process gets launched.

The nmon:config data is associated with the generation of the nmon_inventory lookup and the Nmon_Config data model.

sourcetype=nmon_collect

The “**nmon_collect**” sourcetype available in the **eventtype=nmon:collect** contains all the data related to the nmon processes generation on your systems.

These are the ouput of the input script “nmon_helper.sh” thats gets automatically launched by Splunk when deploying the Nmon App. By default, the nmon_helper.sh script gets started every minute, its main job is to verify the status of the current nmon process, and start a new one if conditions requires it.

Many nmon starting options can be controlled through the “nmon.conf” configuration file during the deployment of the App.

sourcetype=nmon_processing

The “**nmon_processing**” sourceytpe available in the **eventtype=nmon:collect** contains all the data related to the nmon processing steps that converts nmon data into usable data for Splunk.

These are the ouput of nmon2csv Python and Perl parsers provided within the App. Every time an existing raw nmon file is updated, or a new one gets created, Splunk will call parsers scripts and generate appropriated data.

sourcetype=nmon_clean

The “**nmon_clean**” sourceytpe available in the **eventtype=nmon:clean** contains all the data related to various cleaning steps operated by nmon_cleaner scripts.

These scripts are responsible in cleaning raw nmon data file periodically, and also cleaning csv raw data in case of an unexpected Splunk failure to prevent from filling the file system with unconsumed csv data.

2.17.3 Lookups and KV Store

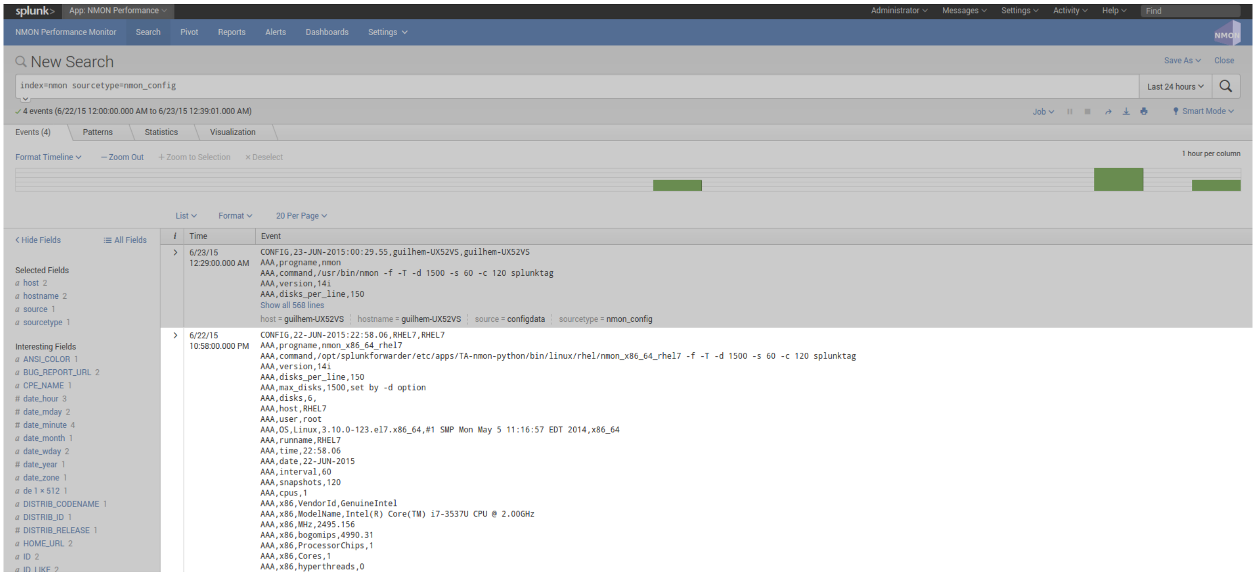
nmon_inventory

Nmon Inventory (nmon_inventory): Inventory Extraction of NMON data

The Nmon Inventory data is an important piece of the Application, it is being used to provide useful inventory information about your servers with main configuration items. (CPU and memory configuration, uptime. . .)

Since the major release V1.7, the nmon inventory data is stored in a KVstore collection

The nmon_inventory data is build over the nmon_config sourcetype which contains the extraction of AAA and BBB* sections of Nmon:


splunk> App: NMON Performance
Administrator Messages Settings Activity Help Find
NMON Performance Monitor Search Pivot Reports Alerts Dashboards Settings
New Search
Save As Close
Last 24 hours
Job
Smart Mode
Events (4) Patterns Statistics Visualization
Format Timeline — Zoom Out + Zoom to Selection × Deselect
1 hour per column
List Format 20 Per Page
< Hide Fields ☰ All Fields

```
index=nmon sourcetype=nmon_config
```

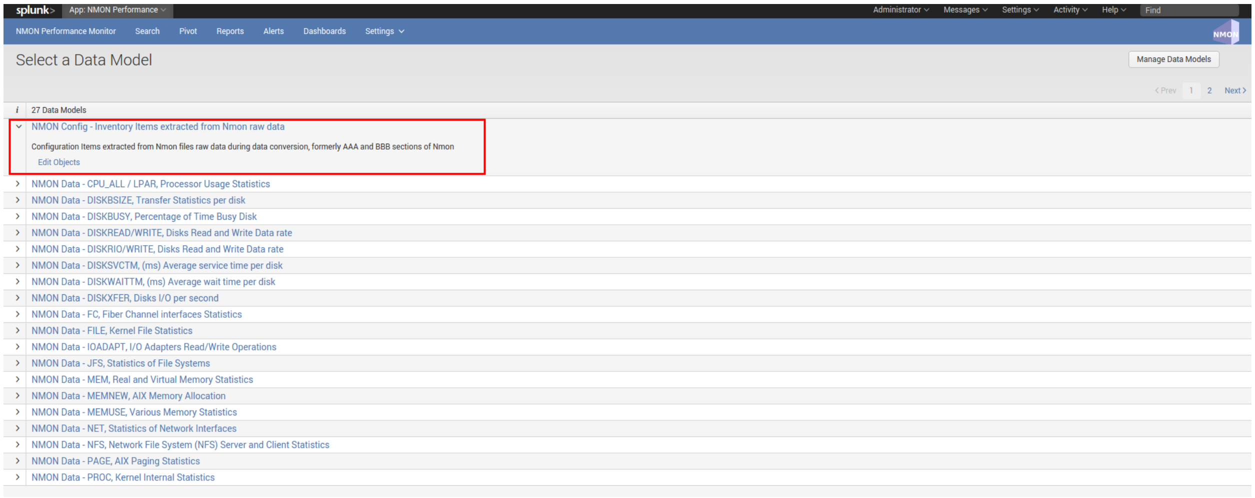
✓ 4 events (6/22/15 12:00:00.000 AM to 6/23/15 12:39:01.000 AM)

Selected Fields
- a host 2
- a hostname 2
- a source 1
- a sourcetype 1

Interesting Fields
- a ANSI_COLOR 1
- a BUG_REPORT_URL 2
- a CPE_NAME 1
- # date_hour 3
- # date_mday 2
- # date_minute 4
- a date_month 1
- a date_wday 2
- # date_year 1
- a date_zone 1
- a de 1 × 512 1
- a DISTRIB_CODENAME 1
- a DISTRIB_ID 1
- # DISTRIB_RELEASE 1
- a HOME_URL 2
- a ID 2
- a ID_LIKE 2

i	Time	Event
>	6/23/15 12:29:00.000 AM	CONFIG,23-JUN-2015:00:29.55,guilhem-UX52VS,guilhem-UX52VS AAA,progname,nmon AAA,command,/usr/bin/nmon -f -T -d 1500 -s 60 -c 120 splunktag AAA,version,14i AAA,disks_per_line,150 Show all 568 lines host = guilhem-UX52VS ⁞ hostname = guilhem-UX52VS ⁞ source = configdata ⁞ sourcetype = nmon_config
>	6/22/15 10:58:00.000 PM	CONFIG,22-JUN-2015:22:58.06,RHEL7,RHEL7 AAA,progname,nmon_x86_64_rhel7 AAA,command,/opt/splunkforwarder/etc/apps/TA-nmon-python/bin/linux/rhel/nmon_x86_64_rhel7 -f -T -d 1500 -s 60 -c 120 splunktag AAA,version,14i AAA,disks_per_line,150 AAA,max_disks,1500,set by -d option AAA,disks,6, AAA,host,RHEL7 AAA,user,root AAA,OS,Linux,3.10.0-123.el7.x86_64,#1 SMP Mon May 5 11:16:57 EDT 2014,x86_64 AAA,runname,RHEL7 AAA,time,22:58.06 AAA,date,22-JUN-2015 AAA,interval,60 AAA,snapshots,120 AAA,cpus,1 AAA,x86,VendorId,GenuineIntel AAA,x86,ModelName,Intel(R) Core(TM) i7-3537U CPU @ 2.00GHz AAA,x86,MHz,2495.156 AAA,x86,bogomips,4990.31 AAA,x86,ProcessorChips,1 AAA,x86,Cores,1 AAA,x86,hyperthreads,0

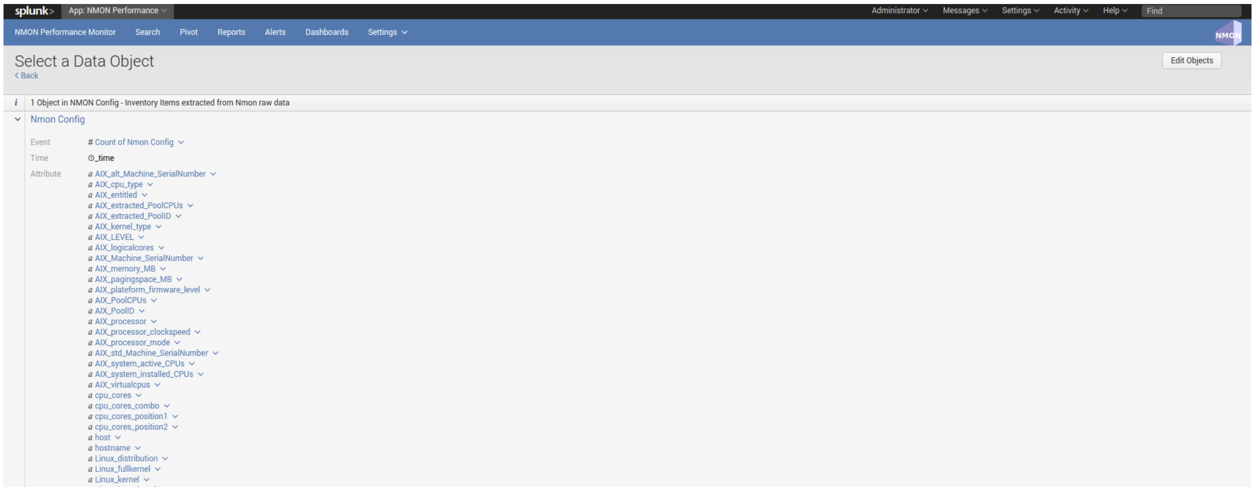
To build with efficiency the nmon_inventory data, the Application uses the accelerated data model “NMON Config - Inventory Items extracted from Nmon raw data” and intensive regular expressions:


splunk> App: NMON Performance
Administrator Messages Settings Activity Help Find
NMON Performance Monitor Search Pivot Reports Alerts Dashboards Settings
Manage Data Models
< Prev 1 2 Next >

Select a Data Model

i 27 Data Models

- NMON Config - Inventory Items extracted from Nmon raw data
  Configuration Items extracted from Nmon files raw data during data conversion, formerly AAA and BBB sections of Nmon
  Edit Objects
- NMON Data - CPU_ALL / LPAR, Processor Usage Statistics
- NMON Data - DISKBSIZE, Transfer Statistics per disk
- NMON Data - DISKBUSY, Percentage of Time Busy Disk
- NMON Data - DISKREAD/WRITE, Disks Read and Write Data rate
- NMON Data - DISKRIO/WRITE, Disks Read and Write Data rate
- NMON Data - DISKSVCTM, (ms) Average service time per disk
- NMON Data - DISKWAITTM, (ms) Average wait time per disk
- NMON Data - DISKXFER, Disks I/O per second
- NMON Data - FC, Fiber Channel interfaces Statistics
- NMON Data - FILE, Kernel File Statistics
- NMON Data - IOADAPT, I/O Adapters Read/Write Operations
- NMON Data - JFS, Statistics of File Systems
- NMON Data - MEM, Real and Virtual Memory Statistics
- NMON Data - MEMNEW, AIX Memory Allocation
- NMON Data - MEMUSE, Various Memory Statistics
- NMON Data - NET, Statistics of Network Interfaces
- NMON Data - NFS, Network File System (NFS) Server and Client Statistics
- NMON Data - PAGE, AIX Paging Statistics
- NMON Data - PROC, Kernel Internal Statistics


splunk> App: NMON Performance
Administrator Messages Settings Activity Help Find
NMON Performance Monitor Search Pivot Reports Alerts Dashboards Settings
Edit Objects
< Back

Select a Data Object

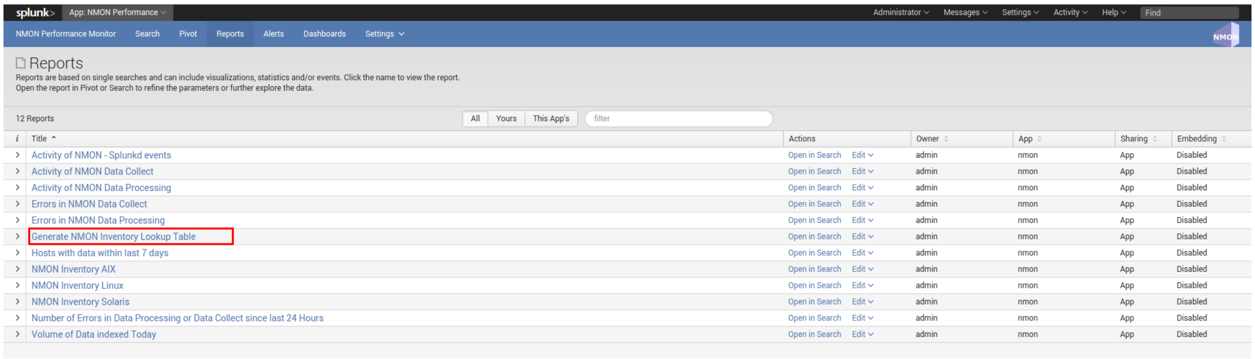
i 1 Object in NMON Config - Inventory Items extracted from Nmon raw data

Nmon Config

Event: # Count of Nmon Config

Time: _time

Attribute:
- a AIX_alt_Machine_SerialNumber
- a AIX_cpu_type
- a AIX_entitled
- a AIX_extracted_PoolCPUs
- a AIX_extracted_PoolID
- a AIX_kernel_type
- a AIX_LEVEL
- a AIX_logicalcores
- a AIX_Machine_SerialNumber
- a AIX_memory_MB
- a AIX_pagingspace_MB
- a AIX_plateform_firmware_level
- a AIX_PoolCPUs
- a AIX_PoolID
- a AIX_processor
- a AIX_processor_clockspeed
- a AIX_processor_mode
- a AIX_std_Machine_SerialNumber
- a AIX_system_active_CPUs
- a AIX_system_installed_CPUs
- a AIX_virtualcpus
- a cpu_cores
- a cpu_cores_combo
- a cpu_cores_position1
- a cpu_cores_position2
- a host
- a hostname
- a Linux_distribution
- a Linux_fullkernel
- a Linux_kernel



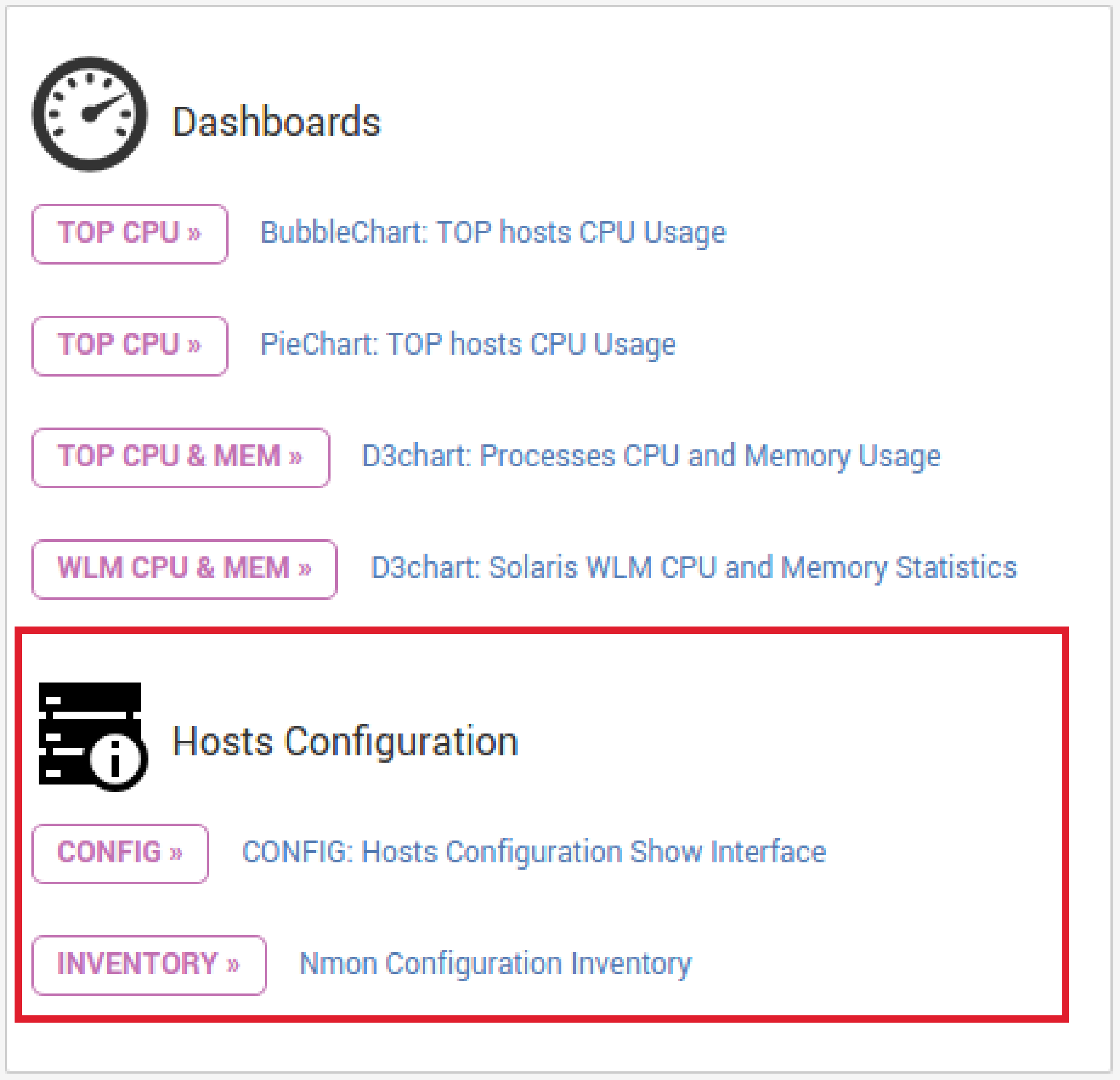
Reports

Reports are based on single searches and can include visualizations, statistics and/or events. Click the name to view the report.
Open the report in Pivot or Search to refine the parameters or further explore the data.

12 Reports

All | Yours | This App's | filter

i	Title	Actions		Owner	App	Sharing	Embedding
>	Activity of NMON - Splunkd events	Open in Search	Edit	admin	nmon	App	Disabled
>	Activity of NMON Data Collect	Open in Search	Edit	admin	nmon	App	Disabled
>	Activity of NMON Data Processing	Open in Search	Edit	admin	nmon	App	Disabled
>	Errors in NMON Data Collect	Open in Search	Edit	admin	nmon	App	Disabled
>	Errors in NMON Data Processing	Open in Search	Edit	admin	nmon	App	Disabled
>	Generate NMON Inventory Lookup Table	Open in Search	Edit	admin	nmon	App	Disabled
>	Hosts with data within last 7 days	Open in Search	Edit	admin	nmon	App	Disabled
>	NMON Inventory AIX	Open in Search	Edit	admin	nmon	App	Disabled
>	NMON Inventory Linux	Open in Search	Edit	admin	nmon	App	Disabled
>	NMON Inventory Solaris	Open in Search	Edit	admin	nmon	App	Disabled
>	Number of Errors in Data Processing or Data Collect since last 24 Hours	Open in Search	Edit	admin	nmon	App	Disabled
>	Volume of Data indexed Today	Open in Search	Edit	admin	nmon	App	Disabled



Dashboards

TOP CPU » BubbleChart: TOP hosts CPU Usage

TOP CPU » PieChart: TOP hosts CPU Usage

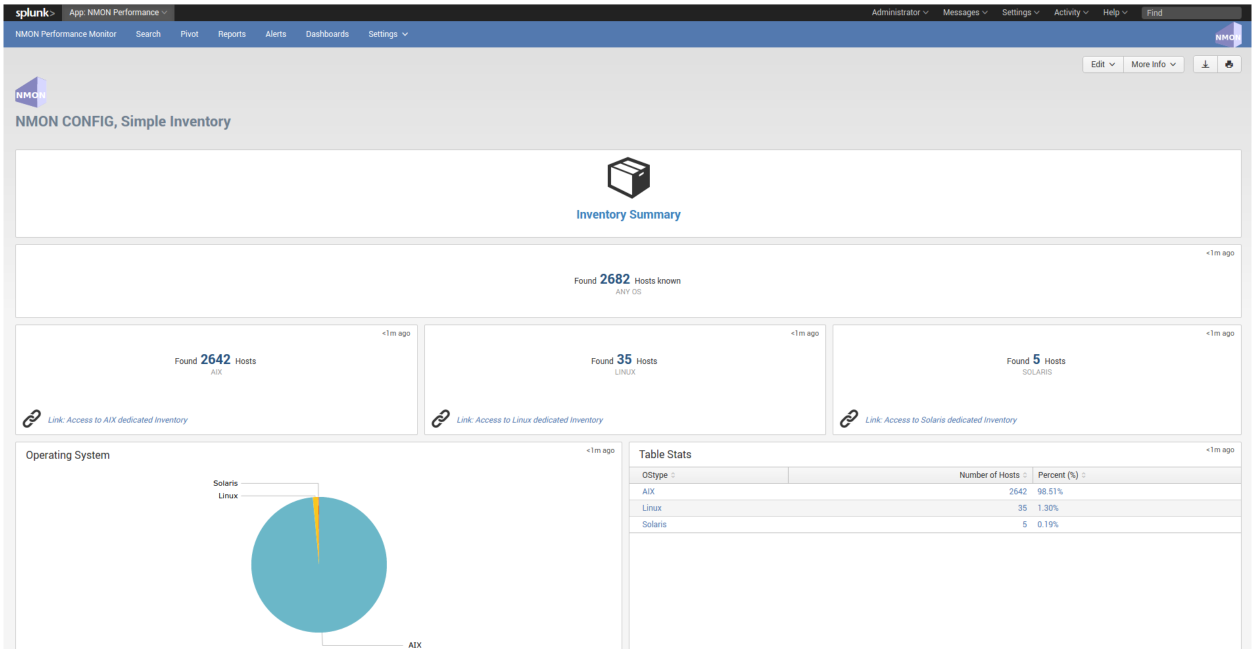
TOP CPU & MEM » D3chart: Processes CPU and Memory Usage

WLM CPU & MEM » D3chart: Solaris WLM CPU and Memory Statistics

Hosts Configuration

CONFIG » CONFIG: Hosts Configuration Show Interface

INVENTORY » Nmon Configuration Inventory



NMON CONFIG, Simple Inventory

Inventory Summary

Found 2682 Hosts known
ANY OS

Found 2642 Hosts
AIX

Link: Access to AIX dedicated Inventory

Found 35 Hosts
LINUX

Link: Access to Linux dedicated Inventory

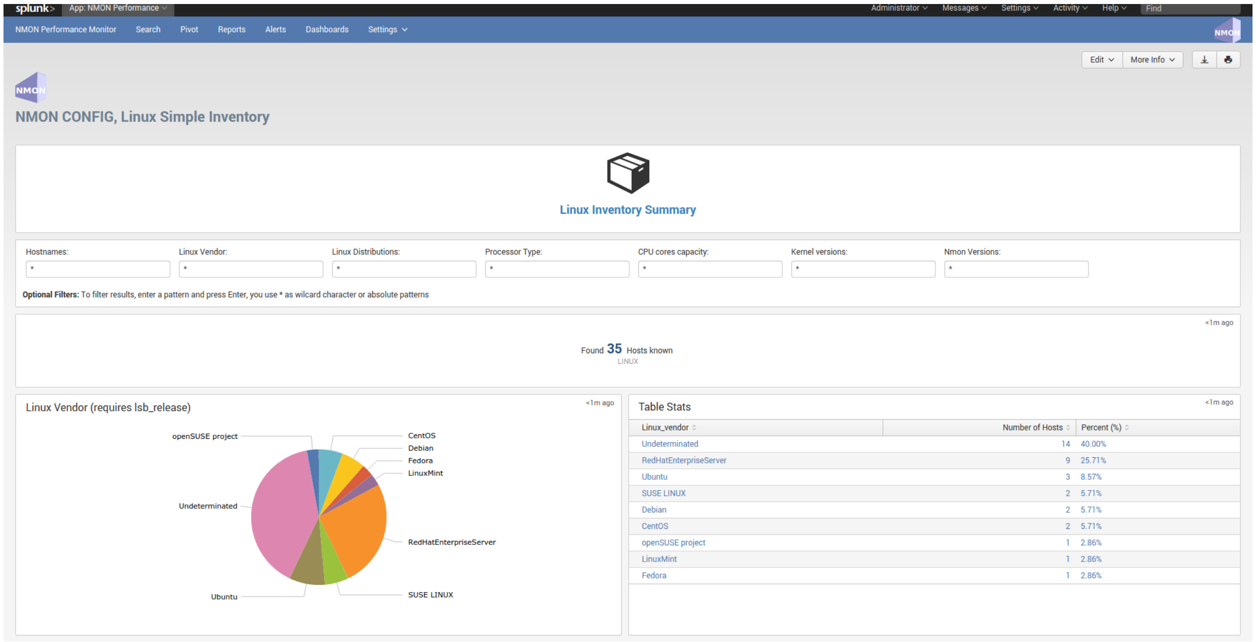
Found 5 Hosts
SOLARIS

Link: Access to Solaris dedicated Inventory

Operating System

Table Stats

OStype	Number of Hosts	Percent (%)
AIX	2642	98.51%
Linux	35	1.30%
Solaris	5	0.19%



NMON CONFIG, Linux Simple Inventory

Linux Inventory Summary

Hostnames: *
Linux Vendor: *
Linux Distributions: *
Processor Type: *
CPU cores capacity: *
Kernel versions: *
Nmon Versions: *

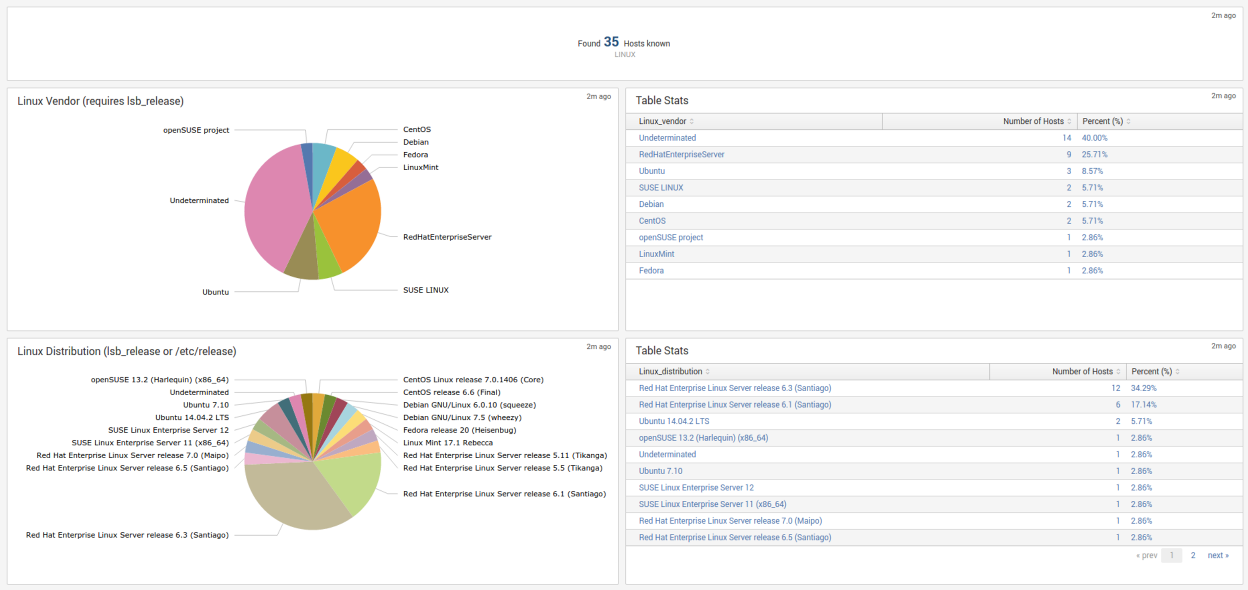
Optional Filters: To filter results, enter a pattern and press Enter, you use * as wilcard character or absolute patterns

Found 35 Hosts known
LINUX

Linux Vendor (requires lsb_release)

Table Stats

Linux_vendor	Number of Hosts	Percent (%)
Undeterminated	14	40.00%
RedHatEnterpriseServer	9	25.71%
Ubuntu	3	8.57%
SUSE LINUX	2	5.71%
Debian	2	5.71%
CentOS	2	5.71%
openSUSE project	1	2.86%
LinuxMint	1	2.86%
Fedora	1	2.86%


Linux Vendor (requires lsb_release)
2m ago
openSUSE project
CentOS
Debian
Fedora
LinuxMint
Undeterminated
RedHatEnterpriseServer
Ubuntu
SUSE LINUX
Linux Distribution (lsb_release or /etc/release)
2m ago
openSUSE 13.2 (Harlequin) (x86_64)
Undeterminated
Ubuntu 7.10
Ubuntu 14.04.2 LTS
SUSE Linux Enterprise Server 12
SUSE Linux Enterprise Server 11 (x86_64)
Red Hat Enterprise Linux Server release 7.0 (Maipo)
Red Hat Enterprise Linux Server release 6.5 (Santiago)
Red Hat Enterprise Linux Server release 6.3 (Santiago)
CentOS Linux release 7.0.1406 (Core)
CentOS release 6.6 (Final)
Debian GNU/Linux 6.0.10 (squeeze)
Debian GNU/Linux 7.5 (wheezy)
Fedora release 20 (Heisenbug)
Linux Mint 17.1 Rebecca
Red Hat Enterprise Linux Server release 5.11 (Tikanga)
Red Hat Enterprise Linux Server release 5.5 (Tikanga)
Red Hat Enterprise Linux Server release 6.1 (Santiago)

2m ago

Found **35** Hosts known
LINUX

Table Stats

2m ago

Linux_vendor	Number of Hosts	Percent (%)
Undeterminated	14	40.00%
RedHatEnterpriseServer	9	25.71%
Ubuntu	3	8.57%
SUSE LINUX	2	5.71%
Debian	2	5.71%
CentOS	2	5.71%
openSUSE project	1	2.86%
LinuxMint	1	2.86%
Fedora	1	2.86%

Table Stats

2m ago

Linux_distribution	Number of Hosts	Percent (%)
Red Hat Enterprise Linux Server release 6.3 (Santiago)	12	34.29%
Red Hat Enterprise Linux Server release 6.1 (Santiago)	6	17.14%
Ubuntu 14.04.2 LTS	2	5.71%
openSUSE 13.2 (Harlequin) (x86_64)	1	2.86%
Undeterminated	1	2.86%
Ubuntu 7.10	1	2.86%
SUSE Linux Enterprise Server 12	1	2.86%
SUSE Linux Enterprise Server 11 (x86_64)	1	2.86%
Red Hat Enterprise Linux Server release 7.0 (Maipo)	1	2.86%
Red Hat Enterprise Linux Server release 6.5 (Santiago)	1	2.86%

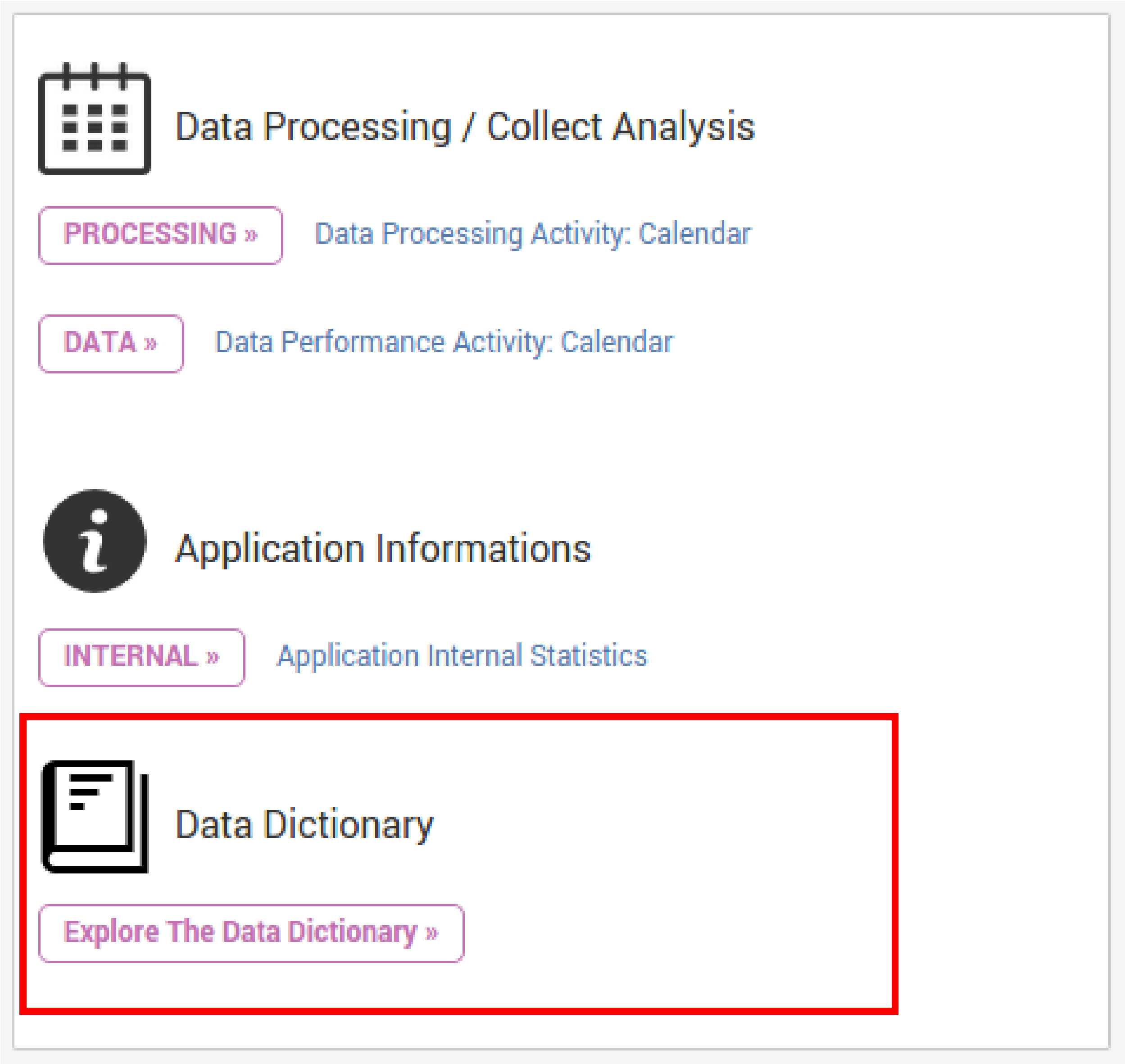
« prev 1 2 next »

data_dictionary

data-dictionary lookup : Dictionary of Nmon Application data

The “data-dictionary” lookup is a csv file provided by the Application, it contains the definition of every piece of data available within the Application.

It is being used in the “Data Dictionary” interface to provide a extensible view of metrics and data available in the context of the application, with a hierarchy by type of operating system:



Data Processing / Collect Analysis

PROCESSING » Data Processing Activity: Calendar

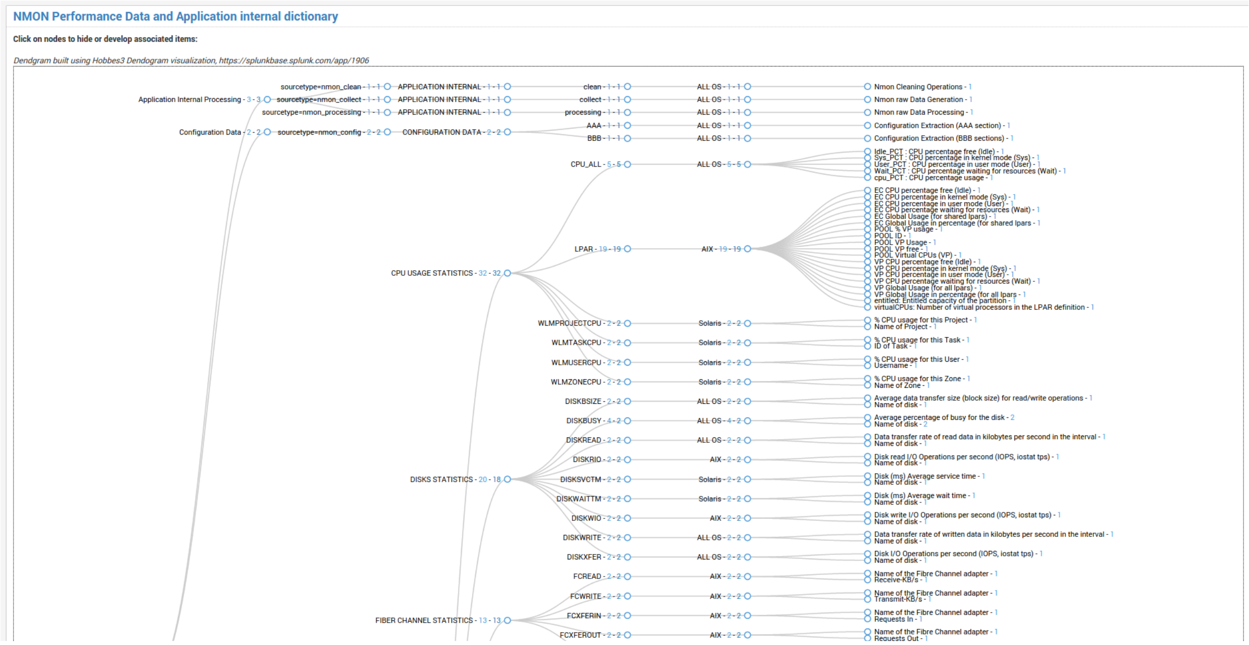
DATA » Data Performance Activity: Calendar

Application Informations

INTERNAL » Application Internal Statistics

Data Dictionary

Explore The Data Dictionary »

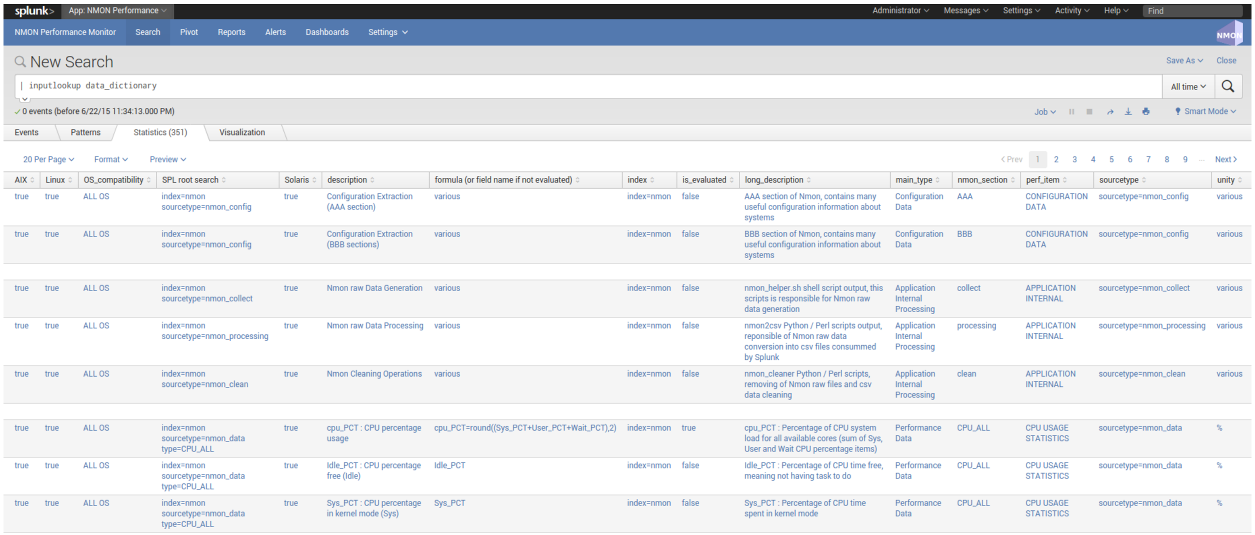


NMON Performance Data and Application internal dictionary

Click on nodes to hide or develop associated items:

Dendgram built using Hobbes3 Dendogram visualization, https://splunkbase.splunk.com/app/1906

Application Internal Processing - 3 - 3
sourcetype=nmon_clean - 1 - 1
sourcetype=nmon_collect - 1 - 1
sourcetype=nmon_processing - 1 - 1
APPLICATION INTERNAL - 1 - 1
APPLICATION INTERNAL - 1 - 1
APPLICATION INTERNAL - 1 - 1
clean - 1 - 1
collect - 1 - 1
processing - 1 - 1
Configuration Data - 2 - 2
sourcetype=nmon_config - 2 - 2
CONFIGURATION DATA - 2 - 2
AAA - 1 - 1
BBB - 1 - 1
ALL OS - 1 - 1
ALL OS - 1 - 1
ALL OS - 1 - 1
ALL OS - 1 - 1
ALL OS - 1 - 1
Nmon Cleaning Operations - 1
Nmon raw Data Generation - 1
Nmon raw Data Processing - 1
Configuration Extraction (AAA section) - 1
Configuration Extraction (BBB sections) - 1
CPU USAGE STATISTICS - 32 - 32
CPU_ALL - 5 - 5
ALL OS - 5 - 5
Idle_PCT : CPU percentage free (Idle) - 1
Sys_PCT : CPU percentage in kernel mode (Sys) - 1
User_PCT : CPU percentage in user mode (User) - 1
Wait_PCT : CPU percentage waiting for resources (Wait) - 1
cpu_PCT : CPU percentage usage - 1
LPAR - 19 - 19
AIX - 19 - 19
EC CPU percentage free (Idle) - 1
EC CPU percentage in kernel mode (Sys) - 1
EC CPU percentage in user mode (User) - 1
EC CPU percentage waiting for resources (Wait) - 1
EC Global Usage (for shared lpars) - 1
EC Global Usage in percentage (for shared lpars - 1
POOL % VP usage - 1
POOL ID - 1
POOL VP Usage - 1
POOL VP free - 1
POOL Virtual CPUs (VP) - 1
VP CPU percentage free (Idle) - 1
VP CPU percentage in kernel mode (Sys) - 1
VP CPU percentage in user mode (User) - 1
VP CPU percentage waiting for resources (Wait) - 1
VP Global Usage (for all lpars) - 1
VP Global Usage in percentage (for all lpars - 1
entitled: Entitled capacity of the partition - 1
virtualCPUs: Number of virtual processors in the LPAR definition - 1
WLMPROJECTCPU - 2 - 2
Solaris - 2 - 2
% CPU usage for this Project - 1
Name of Project - 1
WLMTASKCPU - 2 - 2
Solaris - 2 - 2
% CPU usage for this Task - 1
ID of Task - 1
WLMUSERCPU - 2 - 2
Solaris - 2 - 2
% CPU usage for this User - 1
Username - 1
WLMZONECPU - 2 - 2
Solaris - 2 - 2
% CPU usage for this Zone - 1
Name of Zone - 1
DISKS STATISTICS - 20 - 18
DISKBSIZE - 2 - 2
ALL OS - 2 - 2
Average data transfer size (block size) for read/write operations - 1
Name of disk - 1
DISKBUSY - 4 - 2
ALL OS - 4 - 2
Average percentage of busy for the disk - 2
Name of disk - 2
DISKREAD - 2 - 2
ALL OS - 2 - 2
Data transfer rate of read data in kilobytes per second in the interval - 1
Name of disk - 1
DISKRIO - 2 - 2
AIX - 2 - 2
Disk read I/O Operations per second (IOPS, iostat tps) - 1
Name of disk - 1
DISKSVCTM - 2 - 2
Solaris - 2 - 2
Disk (ms) Average service time - 1
Name of disk - 1
DISKWAITTM - 2 - 2
Solaris - 2 - 2
Disk (ms) Average wait time - 1
Name of disk - 1
DISKWIO - 2 - 2
AIX - 2 - 2
Disk write I/O Operations per second (IOPS, iostat tps) - 1
Name of disk - 1
DISKWRITE - 2 - 2
ALL OS - 2 - 2
Data transfer rate of written data in kilobytes per second in the interval - 1
Name of disk - 1
DISKXFER - 2 - 2
ALL OS - 2 - 2
Disk I/O Operations per second (IOPS, iostat tps) - 1
Name of disk - 1
FIBER CHANNEL STATISTICS - 13 - 13
FCREAD - 2 - 2
AIX - 2 - 2
Name of the Fibre Channel adapter - 1
Receive-KB/s - 1
FCWRITE - 2 - 2
AIX - 2 - 2
Name of the Fibre Channel adapter - 1
Transmit-KB/s - 1
FCXFERIN - 2 - 2
AIX - 2 - 2
Name of the Fibre Channel adapter - 1
Requests In - 1
FCXFEROUT - 2 - 2
AIX - 2 - 2
Name of the Fibre Channel adapter - 1
Requests Out - 1



```
| inputlookup data_dictionary
```

✓ 0 events (before 6/22/15 11:34:13.000 PM)

AIX	Linux	OS_compatibility	SPL root search	Solaris	description	formula (or field name if not evaluated)	index	is_evaluated	long_description	main_type	nmon_section	perf_item	sourcetype	unity
true	true	ALL OS	index=nmon sourcetype=nmon_config	true	Configuration Extraction (AAA section)	various	index=nmon	false	AAA section of Nmon, contains many useful configuration information about systems	Configuration Data	AAA	CONFIGURATION DATA	sourcetype=nmon_config	various
true	true	ALL OS	index=nmon sourcetype=nmon_config	true	Configuration Extraction (BBB sections)	various	index=nmon	false	BBB section of Nmon, contains many useful configuration information about systems	Configuration Data	BBB	CONFIGURATION DATA	sourcetype=nmon_config	various
true	true	ALL OS	index=nmon sourcetype=nmon_collect	true	Nmon raw Data Generation	various	index=nmon	false	nmon_helper.sh shell script output, this scripts is responsible for Nmon raw data generation	Application Internal Processing	collect	APPLICATION INTERNAL	sourcetype=nmon_collect	various
true	true	ALL OS	index=nmon sourcetype=nmon_processing	true	Nmon raw Data Processing	various	index=nmon	false	nmon2csv Python / Perl scripts output, reponsible of Nmon raw data conversion into csv files consummed by Splunk	Application Internal Processing	processing	APPLICATION INTERNAL	sourcetype=nmon_processing	various
true	true	ALL OS	index=nmon sourcetype=nmon_clean	true	Nmon Cleaning Operations	various	index=nmon	false	nmon_cleaner Python / Perl scripts, removing of Nmon raw files and csv data cleaning	Application Internal Processing	clean	APPLICATION INTERNAL	sourcetype=nmon_clean	various
true	true	ALL OS	index=nmon sourcetype=nmon_data type=CPU_ALL	true	cpu_PCT : CPU percentage usage	cpu_PCT=round((Sys_PCT+User_PCT+Wait_PCT),2)	index=nmon	true	cpu_PCT : Percentage of CPU system load for all available cores (sum of Sys, User and Wait CPU percentage items)	Performance Data	CPU_ALL	CPU USAGE STATISTICS	sourcetype=nmon_data	%
true	true	ALL OS	index=nmon sourcetype=nmon_data type=CPU_ALL	true	Idle_PCT : CPU percentage free (Idle)	Idle_PCT	index=nmon	false	Idle_PCT : Percentage of CPU time free, meaning not having task to do	Performance Data	CPU_ALL	CPU USAGE STATISTICS	sourcetype=nmon_data	%
true	true	ALL OS	index=nmon sourcetype=nmon_data type=CPU_ALL	true	Sys_PCT : CPU percentage in kernel mode (Sys)	Sys_PCT	index=nmon	false	Sys_PCT : Percentage of CPU time spent in kernel mode	Performance Data	CPU_ALL	CPU USAGE STATISTICS	sourcetype=nmon_data	%

metric_catalog

metric_catalog lookup : metric catalog definition

The “metric_catalog” lookup is a csv file provided by the Application, it contains various definition of metrics to be dynamically used by different interfaces.

The catalog contains the following definition:

- metric_name:

a logical name of the metric, with the following naming convention: <metric context>.<nmon_section_name>.<metric_fieldname>.

- metric_label:

a contextual label that describes the metric.

- is_AIX: [TRUE / FALSE]

a boolean value defining compatibility for AIX systems.

- is_Linux: [TRUE / FALSE]

a boolean value defining compatibility for Linux systems.

- is_Solaris: [TRUE / FALSE]

a boolean value defining compatibility for Solaris systems.

- metric_category:

a technical context for the metric category, such as cpu / memory / storage / network.

- metric_has_device: [TRUE / FALSE]

a boolean value defining if the metric has a device dimension available.

- metric_device_field:

the field name containing the device dimension.

- metric_unit:

the default metric unit.

- metric_volume_unit_choice: [TRUE / FALSE]

used by interfaces to dynamically provide a unit dropdown choice for unity conversion.

- metric_dimension_filter:

a string containing the SPL filters, such as “type=CPU_ALL” to be used within SPL native searches.

- metric_value_field:

the name of the field containing the metric value.

- metric_dm:

the data model name associated with this metric.

- metric_dm_node:

the data model node name associated with this metric.

- metric_dm_prefix:

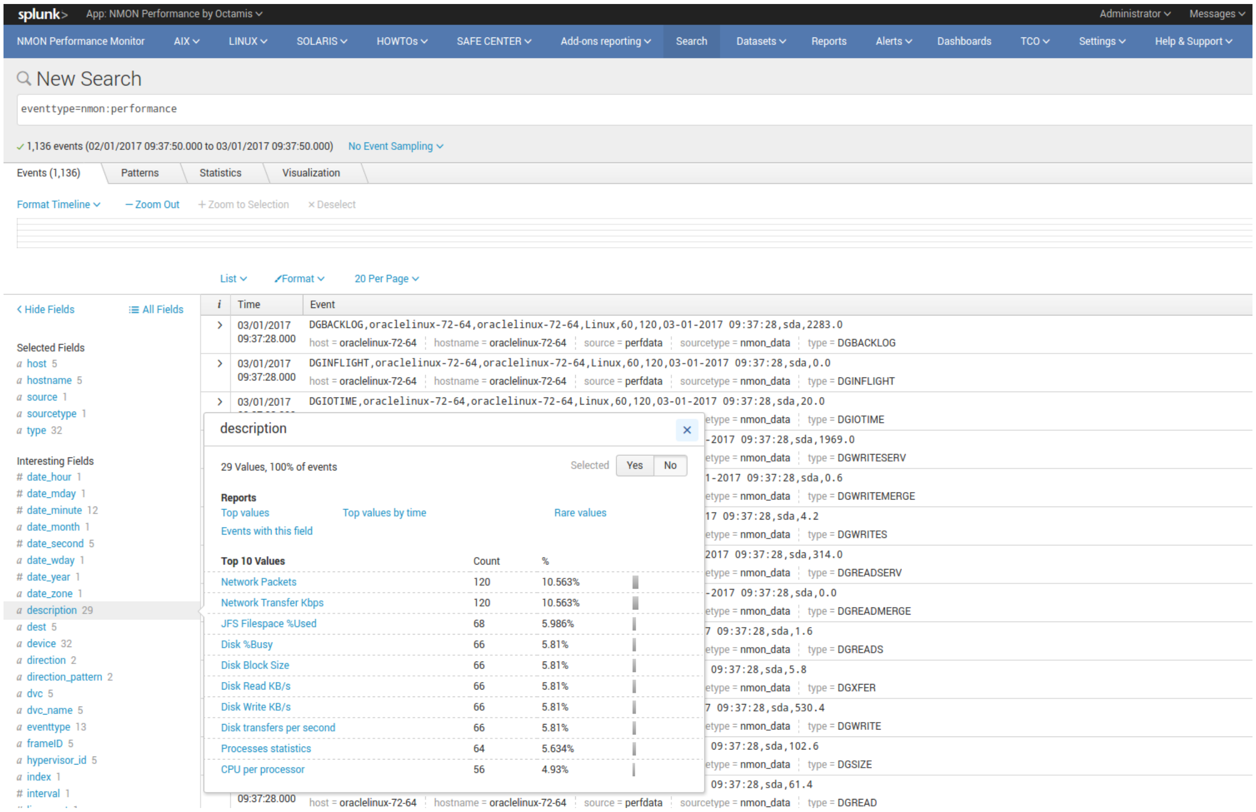
the data model prefix name associated with this metric.

nmon_data_asset_description

nmon_data_asset_description: Description enrichment of Nmon performance monitors categories

The “nmon_data_asset_description” lookup is a csv file provided by the Application, it is being used to statically enrich the nmon data.

Depending on the “type” field which determines the type of performance monitor (ex: CPU_ALL for Total CPU usage), a field “description” will contain a human readable description of what does this performance monitor.



```
eventtype=nmon:performance
```

✓ 1,136 events (02/01/2017 09:37:50.000 to 03/01/2017 09:37:50.000)

i	Time	Event
>	03/01/2017 09:37:28.000	DGBACKLOG,oraclelinux-72-64,oraclelinux-72-64,Linux,60,120,03-01-2017 09:37:28,sda,2283.0 host = oraclelinux-72-64 hostname = oraclelinux-72-64 source = perfdata sourcetype = nmon_data type = DGBACKLOG
>	03/01/2017 09:37:28.000	DGINFLIGHT,oraclelinux-72-64,oraclelinux-72-64,Linux,60,120,03-01-2017 09:37:28,sda,0.0 host = oraclelinux-72-64 hostname = oraclelinux-72-64 source = perfdata sourcetype = nmon_data type = DGINFLIGHT
>	03/01/2017	DGIOTIME,oraclelinux-72-64,oraclelinux-72-64,Linux,60,120,03-01-2017 09:37:28,sda,20.0

description

29 Values, 100% of events

Top 10 Values	Count	%
Network Packets	120	10.563%
Network Transfer Kbps	120	10.563%
JFS Filespace %Used	68	5.986%
Disk %Busy	66	5.81%
Disk Block Size	66	5.81%
Disk Read KB/s	66	5.81%
Disk Write KB/s	66	5.81%
Disk transfers per second	66	5.81%
Processes statistics	64	5.634%
CPU per processor	56	4.93%

nmon_baseline

Nmon Baseline (nmon_baseline): Key system metrics from the Nmon KV Store Baseline

The Nmon KV Store baseline is a feature that provides an advanced analysis of historical past data charted versus real time data to help detecting unexpected or unusual system resources usage.

The key concept is quite simple:

Every week (scheduled each Sunday starting at midnight by default), scheduled reports will generate data for different metrics and store the result in kvstore collections:

- CPU (CPU_ALL, LPAR)
- Real and Virtual Memory (MEM)
- Disks I/O per second (DISKXFER)

These reports will generate statistics per day of the week and per 5 minutes step of 3 statistics results for each metric per server: lower (perc05), Average and upper (perc95)

At the end, results are being stored in different kvstore Collections on search heads. (2016 records per server and per kvstore)

Specific macros called within the Baseline interface will retrieve current (or custom if you select your own time range) statistics for these metrics and selected host The macro will evaluate statistics per day of the week and per minute (data is being retrieved from indexers using data model acceleration)

The lookup command being called within the macro will retrieve stored values within the KV Store for associated days of week and minute to generate the metric baseline (eg. compare Mondays over Mondays, Tuesdays over Tuesdays. . .) This operation will fully occurs on search head within generating unnecessary loads for indexers

Finally, if the selected time range runs over the future (default of baseline interface starts at beginning of the day and finishes at the end of the current day), the baseline will be charted over the future in 2 available mode: Full Baseline using the predict rendering with lower, average and upper, of the Simple baseline which will only generate the Average baseline serie

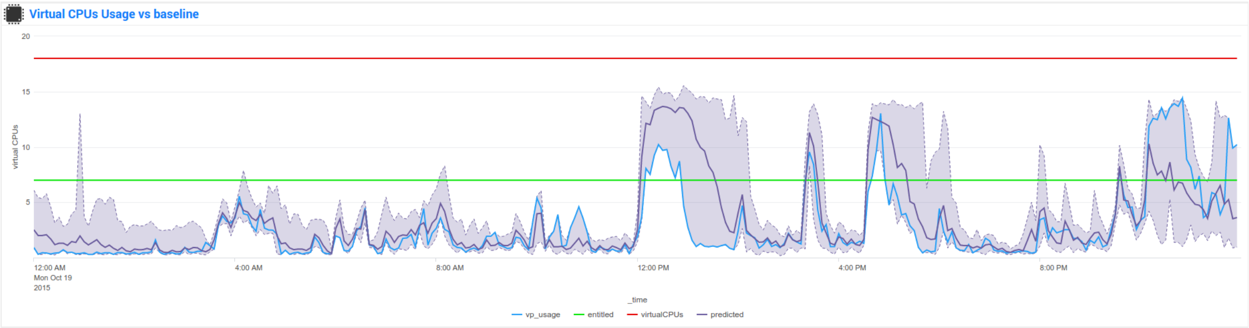
List of kvtore Collections:

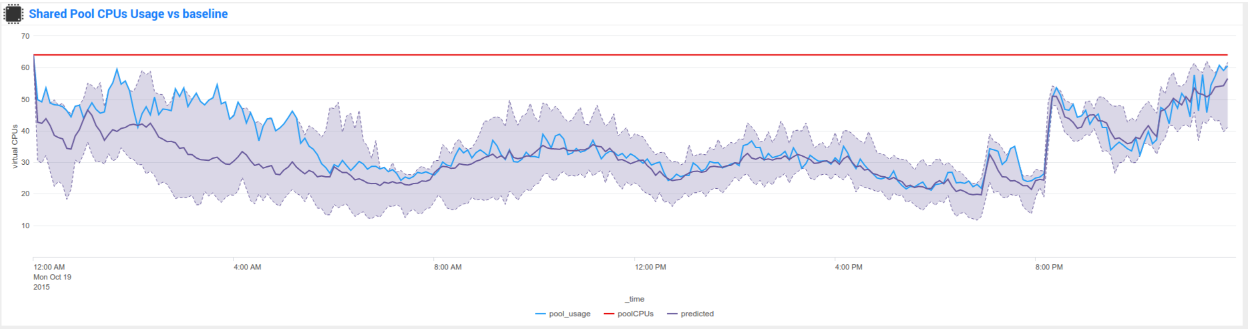
Here are kvstore Collections and corresponding lookup table references:

kvstore collection	lookup name	baseline generation report name
kv_nmon_baseline_CPU_ALL	nmon_baseline_CPU_ALL	Generate NMON Baseline KV Collection for CPU_ALL
kv_nmon_baseline_LPAR	nmon_baseline_LPAR	Generate NMON Baseline KV Collection for LPAR
kv_nmon_baseline_MEM	nmon_baseline_MEM	Generate NMON Baseline KV Collection for MEM
kv_nmon_baseline_DISKXFER	nmon_baseline_DISKXFER	Generate NMON Baseline KV Collection for DISKXFER

Note that only the LPAR kvstore and related report are specific for Power systems, if you are not using such systems, these objects can be safety deactivated.

Here are some examples of the baseline charting:





filesystem_excluding

The lookup table "filesystem_excluding" is file lookup that will contain mount point of file systems to be excluded from file system alerting.

The alert "NMON - File System % usage exceeds 90% (5 consecutive minutes minimal duration)" will exclude any mount point listed in this lookup table from its analysis. Note that this lookup table is case insensitive, can contain wildcards of pattern to be excluded (such as *cdrom*).

upgrade resiliency caution:

If you customize this lookup table, you will need to back it up before upgrading, and recover it from your backup after the update. This feature will probably be updated and improved in future releases!

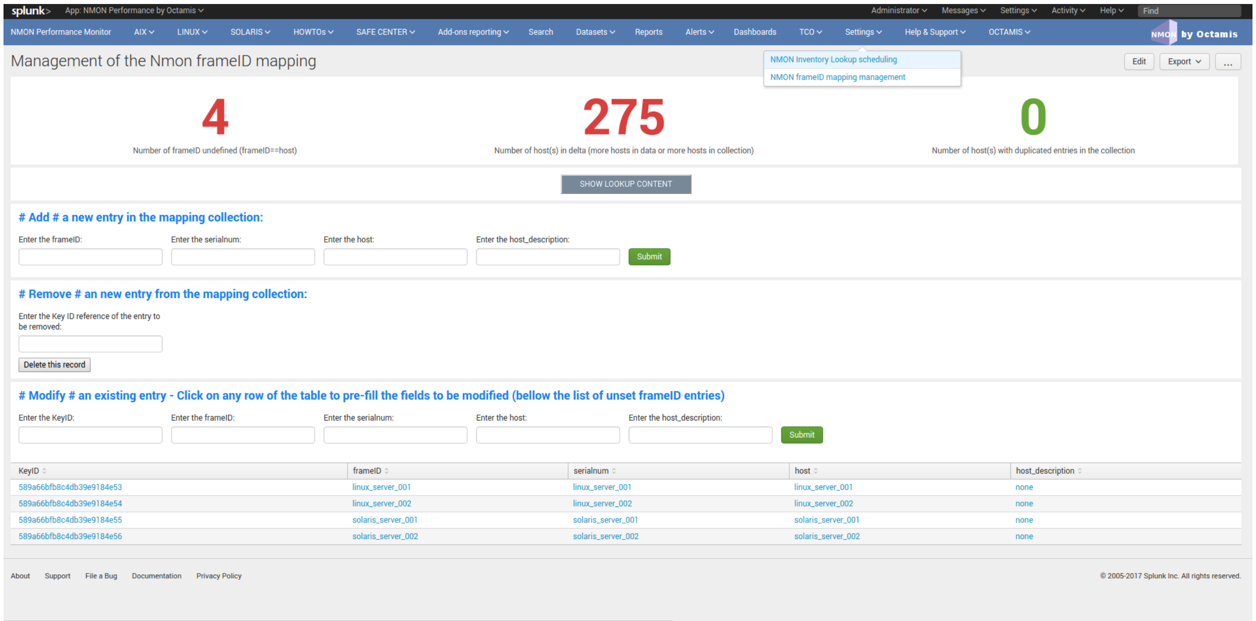
frameID mapping KVstore

nmon_frameID_mapping: logically group hostname in the frameID field

Since the release 1.8.4 of the Nmon core application, the frameID mapping has been improved and operates now against a KVstore collection.

The KVstore based lookup table "nmon_frameID_mapping" is generated automatically by the scheduled report "Generate NMON frameID mapping lookup table". (runs at Splunk startup and every hour by default)

Using the management interface available in the "Settings" application menu, you can directly edit and update the mapping within Splunk web:



Management of the Nmon frameID mapping

4
Number of frameID undefined (frameID==host)

275
Number of host(s) in delta (more hosts in data or more hosts in collection)

0
Number of host(s) with duplicated entries in the collection

Add # a new entry in the mapping collection:

Enter the frameID:
Enter the serialnum:
Enter the host:
Enter the host_description:

Remove # an new entry from the mapping collection:

Enter the Key ID reference of the entry to be removed:

Modify # an existing entry - Click on any row of the table to pre-fill the fields to be modified (bellow the list of unset frameID entries)

Enter the KeyID:
Enter the frameID:
Enter the serialnum:
Enter the host:
Enter the host_description:

KeyID	frameID	serialnum	host	host_description
589a66bfb8c4db39e9184e53	linux_server_001	linux_server_001	linux_server_001	none
589a66bfb8c4db39e9184e54	linux_server_002	linux_server_002	linux_server_002	none
589a66bfb8c4db39e9184e55	solaris_server_001	solaris_server_001	solaris_server_001	none
589a66bfb8c4db39e9184e56	solaris_server_002	solaris_server_002	solaris_server_002	none

By default, the frameID value will be generated using the following rule:

- For AIX, the frameID gets its value from the host serial number (actually the PSeries serial number)
- For Linux, the frameID values is equal to the host name
- For Solaris, the frameID values is equal to the host name

Every time that runs the scheduled report, Splunk will automatically update and add the new values to the KVstore, preserving the existing content.

The frameID feature allows you to easily group the servers in logical containers, and provides an easier and improved selection for a better user experience.

INFORMATION:

When modifying the frameID lookup definition, this will be applied almost immediately for any operation at search time.

Interfaces using data models will as well immediately reflect changes, however you will have to rebuild the data model acceleration if you want these changes to be applied for previously indexed data.

2.17.4 Main Configuration Files

The nmon core application does not collect performance and configuration data, for these items please refer to the TA-nmon documentation:

http://ta-nmon.readthedocs.io

props.conf

props.conf - nmon sourcetypes definition

01 - Nmon Performance Data definition

This stanza defines the nmon_data sourcetype wich contains Nmon Performance data.

```
[nmon_data]

FIELD_DELIMITER=,
FIELD_QUOTE="
HEADER_FIELD_LINE_NUMBER=1

# your settings
INDEXED_EXTRACTIONS=csv
NO_BINARY_CHECK=1
SHOULD_LINEMERGE=false
TIMESTAMP_FIELDS=ZZZZ
TIME_FORMAT=%d-%m-%Y %H:%M:%S

# set by detected source type
KV_MODE=none
pulldown_type=true

# Overwritting default host field based on event data for nmon_data sourcetype␣
↪(useful when managing Nmon central shares)
TRANSFORMS-hostfield=nmon_data_hostoverride
```

This uses csv format defined by csv header, and time stamp definition adapted for generated data from raw nmon data file.

It is being used by associated input monitor in props.conf that consumes csv data from csv_repository.

02 - Nmon Processing definition (output of nmon2csv)

This stanza sets the appropriated time format for indexing of nmon2csv converters.

```
[nmon_processing]

TIME_FORMAT=%d-%m-%Y %H:%M:%S
This sourcetype contains useful information about processing steps operated by␣
↪converters, such as the list of Nmon section proceeded, the number of events per␣
↪section, processing various information and more.
```

03 - Nmon Configuration Data definition

This stanza defines the nmon_config sourcetype wich contains Nmon Configuration data.

```
[nmon_config]

BREAK_ONLY_BEFORE=CONFIG,
MAX_EVENTS=100000
NO_BINARY_CHECK=1
SHOULD_LINEMERGE=true
TIME_FORMAT=%d-%b-%Y:%H:%M
TIME_PREFIX=CONFIG,
TRUNCATE=0

# Overwritting default host field based on event data for nmon_data sourcetype␣
↪(useful when managing Nmon central shares)
TRANSFORMS-hostfield=nmon_config_hostoverride
```

Events stored within this sourcetype are large multi line events containing items available in AAA and BBB* sections of Nmon.

transforms.conf

Notable configuration used in default transforms.conf:

```
###########################################
#                   nmon data stanza
###########################################

# Host override based on event data form nmon_data sourcetype

[nmon_data_hostoverride]
DEST_KEY = MetaData:Host
REGEX = ^\"{0,1}[a-zA-Z0-9\_]+\"{0,1},\"{0,1}[a-zA-Z0-9\-\_\.]+\"{0,1},\"{0,1}([a-zA-
↪Z0-9\-\_\.]+)\"{0,1},.+
FORMAT = host::$1

# New with 1.2.55, allows the perf data generation in json mode
# rewrite the sourcetype to regular nmon_data

[nmon_data_json_hostoverride]
DEST_KEY = MetaData:Host
REGEX = \"hostname\":\s\"([a-zA-Z0-9\-\_\.]+)\"
FORMAT = host::$1

[nmon_data_json_sourcetypeoverride]
DEST_KEY = MetaData:Sourcetype
REGEX = .*
FORMAT = sourcetype::nmon_data

# the following stanza will create **indexed time** fields, to be used when choosing␣
↪the json search time extraction only
# creating fields at indexing time is usually not recommended and not necessary
# however, we want to be able to use the tstats command over some basic fields␣
↪including Splunk meta but as well the basic fields in Nmon context: OStype, type
```

(continues on next page)

(continued from previous page)

```
[nmon_data_json_createindexed_OStype]
REGEX = \"OStype\":\s\"(?<OStype>[^\"]*)\"
WRITE_META = true
FORMAT = OStype::$1
DEFAULT_VALUE = NULL

[nmon_data_json_createindexed_type]
REGEX = \"type\":\s\"(?<type>[^\"]*)\"
WRITE_META = true
FORMAT = type::$1
DEFAULT_VALUE = NULL

###########################################
#                    nmon config stanza
###########################################

# Host override based on event data form nmon_config sourcetype

[nmon_config_hostoverride]
DEST_KEY = MetaData:Host
REGEX = CONFIG\,[a-zA-Z0-9\-\:\.]+\,([a-zA-Z0-9\-\_\.]+)\,[a-zA-Z0-9\-\_\.]+
FORMAT = host::$1
```

The reason for this is simple, when managing Nmon data that has been generated out of Splunk (so not by a Universal Forwarder or full Splunk instance runnning the Application), the “host” field which is a default Splunk field will have the value of the host that managed the data, and not the value of the real host that generated the Nmon data.

This will happens for example when using the Application in a central NFS repository scenario deployment.

Using the configuration above, Splunk will always and automatically rewrite the default host field based on the Nmon data, and not only on Splunk information

2.17.5 Configure

Various configurations and advanced administration tasks

01 - Manage Nmon Collection (generating Performance and Configuration data)

Configuration, tips and advanced configuration about Nmon Raw data generation

Edit AIX Nmon starting options

For AIX, you can manage the full list of Nmon options and control them from a central place using a “local/nmon.conf” configuration file.

Please refer to: http://ta-nmon.readthedocs.io/en/latest/nmon_config.html

To manage AIX Nmon options (but the activation of NFS collection), you will:

- Change the value of mode in your “local/nmon.conf” accorded to your needs
- Update your deployment servers
- The new package version will be pushed to clients, and next iteration of Nmon binaries will start using these values

Activate the Performance Data collection for NFS Statistics

The configuration by default will not collect NFS statistics for AIX / Linux (NFS statistics is currently not available on Solaris), its activation can be controlled through a “local/nmon.conf”:

Please refer to: http://ta-nmon.readthedocs.io/en/latest/nmon_config.html

To activate NFS collection, you will:

- Change the value of mode in your “local/nmon.conf” accorded to your needs
- Update your deployment servers

The new package version will be pushed to clients, and next iteration of Nmon binaries will start using these values

Manage Nmon parallel run between Nmon collections to prevent data gaps

The nmon_helper.sh script will automatically manage a temporarily parallel run of 2 Nmon instances to prevent data gaps between collections.

Please refer to: http://ta-nmon.readthedocs.io/en/latest/nmon_config.html

Things works the following way:

- Each time the nmon_helper.sh runs, the age in seconds of the current instance is evaluated
- It also evaluates the expected time to live in seconds of an Nmon instance based on parameters (interval * snapshot)
- A margin in seconds is applied to the time to live value
- If the age of the current instance gets higher than the time to live less the margin, a new Nmon instance will be launched
- On next iteration of nmon_helper.sh script, only the new Nmon instance will be watched and the time to live counter gets reset
- During the parallel run, both instances will run and generate Nmon raw data, nmon2csv converters will prevent any duplicated events and only new data will be indexed
- During the parallel run, more data will be temporarily indexed
- When the time to live of the old Nmon instance reaches its end, the instance will terminate and the parallel run will be finished

In default configuration, the parallel run uses a 4 minutes time margin (240 seconds) defined in default/nmon.conf, this value can be overwritten using a local/nmon.conf

If you have gaps in data due to Nmon collections, then you may need to increase the endtime_margin value, on very big systems Nmon may require more time to start the data collection and the 4 minutes parallel run may not be enough.

To solve this, you can create a local/nmon.conf to include your custom endtime_margin and deploy the update.

Note that this feature can also be totally disabled by setting the endtime_margin to a “0” value.

The nmon_collect sourcetype will contains information about the parallel run, age in seconds of the Nmon current instance and time to live less the endtime margin.

Manage the Volume of data generated by the Nmon Data collection

Each Universal Forwarder running the TA-nmon add-on generates a volume of data which will vary depending on Nmon options sent at binary startup. These settings can be totally managed from a central place using a "local/nmon.conf" configuration file.

Please refer to: http://ta-nmon.readthedocs.io/en/latest/nmon_config.html

To manage the volume of data to be generated, you will:

- Choose a different value for the "interval" variable (time between measures)
- Adapt the value for the "snapshot" variable (number of measures to be performed)

Since the branch 1.3.x of the TA-nmon, the recommended life cycle of nmon processes (computation of interval and snapshot) is 24 hours.

Prioritization of embedded nmon binaries OR locally available nmon binaries

Using nmon.conf configuration file, you can decide to give priority to embedded binaries OR locally available binaries, you should consider giving the priority to embedded binaries versus binaries available on hosts, this feature offers several advantages:

- Automatically use best Nmon binaries compiled for your systems and your architecture
- Manage from a central place binaries versions, updating results in updating only the TA-nmon add-on and pushing it to Deployment Servers

Since release 1.6.07, default configuration sets the priority to embedded binaries:

To enforce the embedded binaries priority:

- Create a "local/nmon.conf"
- Copy the parameter section "Linux_embedded_nmon_priority" from "default/nmon.conf" to your newly created "local/nmon.conf"

Priority to embedded binaries (default):

```
Linux_embedded_nmon_priority="0"
```

Priority to local binaries:

```
Linux_embedded_nmon_priority="1"
```

Update your deployment server and let the package be pushed to your clients

New iteration of Nmon will now use embedded binaries, to get information about the binary in use look in nmon_collect

Linux OS: Number of devices taken in charge at nmon boot time

The maximum number of devices taken in charge by nmon at boot time can be controlled using the "nmon.conf" configuration file.

By default 1500 devices maximum will be taken in charge, up to 3000 devices can be managed by the Application (current hard limit in nmon2csv.py/nmon2csv.pl), configure your "local/nmon.conf" file:

Please refer to: http://ta-nmon.readthedocs.io/en/latest/nmon_config.html

Take note that increasing the number of devices also increases processing and storage costs, but if you have more than 1500 devices and don't set this to a suitable value, Disks analysis would not be complete

- Set this value in your "local/nmon.conf"
- Update your Deloyment Servers
- Let your client have the new package pushed

On next iteration, the Nmon binary will start using the new option

Activate the Performance Data collection for Solaris VxVM Volumes

The configuration by default will not collect Solaris VxVM, its activation can be controlled through a "local/nmon.conf":

Please refer to: http://ta-nmon.readthedocs.io/en/latest/nmon_config.html

default/nmon.conf related settings:

```
# CHange to "1" to activate VxVM volumes IO statistics
Solaris_VxVM="0"
```

To activate NFS collection, you will:

- Change the value of mode in your "local/nmon.conf" accorded to your needs
- Update your deployment servers

The new package version will be pushed to clients, and next iteration of Nmon binaries will start using these values

02 - Manage Core Application: Mapping, Extraction, Restitution

Manage the Core Application

Custom Span definition macro

NMON Performance Monitor uses an advanced search (eg. macro) to dynamically define the more accurate interval time definition possible within charts.

Splunk has a charting limit of 1000 points per series, an adapted span value (time interval) has to be defined if we want charts to be more accurate than Splunk automatically affects

This is why this custom macro is being defined based on analysing Time ranges supplied by users, see:

```
$SPLUNK_HOME/etc/apps/nmon/default/macros.conf
```

Since the major release V1.7, the span management macro were renamed from "inline_customspan" to "nmon_span" for easier usage

If you have a different minimal time interval than 60 seconds between 2 measures at the lower level, you can customize these macro to adapt them to your data. (as for an example if you generate NMON data with an other process than Splunk)

NOTE: This custom configuration has to be done on search heads only

- Create an empty "local/macros.conf" configuration file

- Copy the full stanza of the macro “nmon_span” from “default/macros.conf” to “local/macros.conf”, the original macros contains the following:

```
[nmon_span]
definition = [ | stats count | addinfo\
| eval earliest=if(info_min_time == "0.000", info_search_time,info_min_time)\
| eval latest=if(info_max_time == "+Infinity", info_search_time,info_max_time)\
| eval searchStartTIme=strftime(earliest,"%a %d %B %Y %H:%M")\
| eval searchEndTime=strftime(latest,"%a %d %B %Y %H:%M")\
| eval Difference = (latest - earliest)\
| eval span=case(\
info_min_time == "0.000", "2m",\
Difference > (3000*24*60*60),"4d",\
Difference > (2000*24*60*60),"3d",\
Difference > (1000*24*60*60),"2d",\
Difference > (500*24*60*60),"1d",\
Difference > (333*24*60*60),"12h",\
Difference > (166*24*60*60),"8h",\
Difference > (83*24*60*60),"4h",\
Difference > (41*24*60*60),"2h",\
Difference > (916*60*60),"1h",\
Difference > (833*60*60),"55m",\
Difference > (750*60*60),"50m",\
Difference > (666*60*60),"45m",\
Difference > (583*60*60),"40m",\
Difference > (500*60*60),"35m",\
Difference > (416*60*60),"30m",\
Difference > (333*60*60),"25m",\
Difference > (250*60*60),"20m",\
Difference > (166*60*60),"15m",\
Difference > (83*60*60),"10m",\
Difference > (66*60*60),"5m",\
Difference > (50*60*60),"4m",\
Difference > (33*60*60),"3m",\
Difference > (16*60*60),"2m",\
Difference > (8*60*60),"1m",\
Difference <= (8*60*60),"1m"\
)\
| eval spanrestricted=case(\
info_min_time == "0.000", 2*60,\
Difference > (916*60*60),60*60,\
Difference > (833*60*60),55*60,\
Difference > (750*60*60),50*60,\
Difference > (666*60*60),45*60,\
Difference > (583*60*60),40*60,\
Difference > (500*60*60),35*60,\
Difference > (416*60*60),30*60,\
Difference > (333*60*60),25*60,\
Difference > (250*60*60),20*60,\
Difference > (166*60*60),15*60,\
Difference > (83*60*60),10*60,\
Difference > (66*60*60),5*60,\
Difference > (50*60*60),4*60,\
Difference > (33*60*60),180,\
Difference > (16*60*60),120,\
Difference > (8*60*60),60,\
Difference <= (8*60*60),60\
)\
```

(continues on next page)

(continued from previous page)

```
| eval span=case(spanrestricted < interval, interval, spanrestricted >= interval, span, isnull(interval), span)\
| eval span=if(spanrestricted <= 60, "1m", span)\
| return span ]
iseval = 0
```

They key is modifying that part of the macro code:

```
| eval span=if(spanrestricted <= 60, "1m", span)\
```

By default, if the value of spanrestricted is lower or equal to 60 seconds, a span value of 1 minute will be set

For example, if you want the span value to be never less than 4 minutes (the evaluation will still consider every value), you will set:

```
| eval span=if(spanrestricted <= 240, "4m", span)\
```

Which will give the full following code:

```
[nmon_span]
definition = [ | stats count | addinfo\
| eval earliest=if(info_min_time == "0.000", info_search_time,info_min_time)\
| eval latest=if(info_max_time == "+Infinity", info_search_time,info_max_time)\
| eval searchStartTIme=strftime(earliest,"%a %d %B %Y %H:%M")\
| eval searchEndTime=strftime(latest,"%a %d %B %Y %H:%M")\
| eval Difference = (latest - earliest)\
| eval span=case(\
info_min_time == "0.000", "2m",\
Difference > (3000*24*60*60),"4d",\
Difference > (2000*24*60*60),"3d",\
Difference > (1000*24*60*60),"2d",\
Difference > (500*24*60*60),"1d",\
Difference > (333*24*60*60),"12h",\
Difference > (166*24*60*60),"8h",\
Difference > (83*24*60*60),"4h",\
Difference > (41*24*60*60),"2h",\
Difference > (916*60*60),"1h",\
Difference > (833*60*60),"55m",\
Difference > (750*60*60),"50m",\
Difference > (666*60*60),"45m",\
Difference > (583*60*60),"40m",\
Difference > (500*60*60),"35m",\
Difference > (416*60*60),"30m",\
Difference > (333*60*60),"25m",\
Difference > (250*60*60),"20m",\
Difference > (166*60*60),"15m",\
Difference > (83*60*60),"10m",\
Difference > (66*60*60),"5m",\
Difference > (50*60*60),"4m",\
Difference > (33*60*60),"3m",\
Difference > (16*60*60),"2m",\
Difference > (8*60*60),"1m",\
Difference <= (8*60*60),"1m"\
)\
| eval spanrestricted=case(\
info_min_time == "0.000", 2*60,\
Difference > (916*60*60),60*60,\
```

(continues on next page)

(continued from previous page)

```
Difference > (833*60*60),55*60,\
Difference > (750*60*60),50*60,\
Difference > (666*60*60),45*60,\
Difference > (583*60*60),40*60,\
Difference > (500*60*60),35*60,\
Difference > (416*60*60),30*60,\
Difference > (333*60*60),25*60,\
Difference > (250*60*60),20*60,\
Difference > (166*60*60),15*60,\
Difference > (83*60*60),10*60,\
Difference > (66*60*60),5*60,\
Difference > (50*60*60),4*60,\
Difference > (33*60*60),180,\
Difference > (16*60*60),120,\
Difference > (8*60*60),60,\
Difference <= (8*60*60),60\
)\
| eval span=case(spanrestricted < interval, interval, spanrestricted >= interval,␣
↪span, isnull(interval), span)\
| eval span=if(spanrestricted <= 240, "4m", span)\
| return span ]
iseval = 0
```

Save the file, and update your search heads. (in sh cluster apply the bunde, in standalone restart)

FRAME ID: Mapping hostnames with a Frame Identifier

In large deployment scenarios, mapping hostnames with their Frame Identifier can be very useful to help Analysis, or simply finding the required host.

Before the version 1.8.4, the frameID feature was operated against an optional file based lookup table.

This is has been deprecated and it is now operated against a KVstore collection.

As well, an embedded interface is provided to edit and update the frameID mapping directly within Splunk web.

- See the detailed section: *frameID mapping KVstore*

The old fashion file based feature configuration above is left for history purposes:

Since Version 1.5.0, a Frame ID feature is included within interfaces, in default configuration the frame ID is mapped to the Serial Number of the host.

In AIX OS, the Serial Number is associated the PSeries Serial Number (in Pseries environments), in Linux / Solaris, this is equal to the hostname.

You can customize the Frame Identifier using any external lookup table which will contains one field for the frameIDs, and one field containing hostnames.*

To achieve this, please follow the configuration above:

1. Configure your table lookup in transforms.conf

Create a local/transforms.conf and set your lookup table:

```
[myframeidtable]
filename = my_frameid_lookup.csv
```

Example 1: Map Pseries with hostnames using the serial number field

```
PSERIES_NAME,serialnum
PSERIESfoo,xxxxxxxxxxx
PSERIESbar,xxxxxxxxxxx
```

Example 2: Map frameID with hostnames (using the hostname field)

```
FRAME_NAME,hostname
frame1,hostname1
frame1,hostname2
frame2,hostname3
frame3,hostname4
```

2. Map your hostnames with the frameID in props.conf

Create a local/props.conf and map your hosts within the nmon_data stanza:

Example 1: (Pseries with serial number field)

```
[nmon_data]
LOOKUP-myframeidtable = myframeidtable serialnum AS serialnum OUTPUT PSERIES AS␣
↪frameID
```

Example 2: (frameID with hostnames)

```
[nmon_data]
LOOKUP-myframeidtable = myframeidtable hostname OUTPUT FRAME_NAME AS frameID
```

NOTE: Use “OUTPUT” to generate the frameID field, don’t use OUTPUTNEW which wont’t overwrite the default frameID field

3. Restart Splunk to apply settings

4. Rebuild Acceleration for Datamodel

For each accelerated Data model, please rebuild the acceleration to update the frameID field. (Go in Pivot, manage datamodels, develop each data model and rebuild)

03 - Manage Application Packaging

Manage Application Packaging

create_agent.py: Create multiple TA packages

You may be interested in having different versions of the TA-nmon with the goal to manage different configurations, and target for example specific operating systems or versions with specific configurations.

A Python script utility is provided to allow creating on demand custom TA-nmon packages ready to be deployed, the Python tool allows to:

- Create a new TA-nmon package with the name of your choice
- Customize the target index name if required (eg. for example if you use the customization tool to change the default index name
- Choose between Python Data Processing, or Perl Data Processing

The "create_agent.py" Python tool is available in Git: https://github.com/guilhemmarchand/TA-nmon

Notice for updating the application: using this tool is upgrade resilient, you can create your package and repeat this operation for future release update

The tool requires Python 2.7.x or superior to operate, you can check your version with the following command:

```
python --version
```

```
python create_agent.py

create_agent.py

This utility had been designed to allow creating customized agents for the Nmon␣
↪Splunk Application, please follow these instructions:

- Download the TA-nmon tgz archive (from Splunk base or from Git) and the create_
↪agent.py script (from Git)
- Ensure to store both files in the same directory
- Run the tool: ./create_agent.py and check for available options
- After the execution, a new agent package will have been created in the directory
- Extract its content to your Splunk deployment server, configure the server class,␣
↪associated clients and deploy the agent
- Don't forget to set the application to restart splunkd after deployment
./create_agent.py -h
usage: create_agent.py [-h] [-f INFILE] [--indexname INDEX_NAME]
                       [--agentname TA_NMON] [--agentmode AGENTMODE]
                       [--version]

optional arguments:
  -h, --help            show this help message and exit
  -f INFILE             Name of the Nmon Splunk APP tgz Archive file
  --indexname INDEX_NAME
                        Customize the Application Index Name (default: nmon)
  --agentname TA_NMON   Define the TA Agent name and root directory
  --agentmode AGENTMODE
                        Define the Data Processing mode, valid values are:
                        python,perl / Default value is python
  --version             show program's version number and exit
```

Example of utilization: Create a custom TA package called "TA-nmon-perl" that will use "myindex" as the App index, and Perl as the Data processing language

```
./create_agent.py -f TA-nmon_1316.tgz --agentname TA-nmon-perl --agentmode perl --
↪indexname myindex
Extracting tgz Archive: TA-nmon_1316.tgz
INFO: Extracting Agent tgz resources Archives
INFO: Renaming TA-nmon default agent to TA-nmon-perl
Achieving files transformation...
Done.
INFO: Customizing any reference to index name in files
INFO: ************* Tar creation done of: TA-nmon-perl.tgz *************

*** Agent Creation terminated: To install the agent: ***

 - Upload the tgz Archive TA-nmon-perl.tgz to your Splunk deployment server
 - Extract the content of the TA package in $SPLUNK_HOME/etc/deployment-apps/
 - Configure the Application (set splunkd to restart), server class and associated␣
↪clients to push the new package to your clients
```

(continues on next page)

(continued from previous page)

```

Operation terminated.
```

Nmon_SplunkApp_Customize.py: Customize the Application

If for some reason you need to customize the Nmon Splunk Application, A Python command line tool is provided in the resources directory which will help you easily achieving your customizations.

The Python tool allows to:

- Customize the Application Index Name (deprecated since 1.8.4, prefer using the nmon pattern, see notes bellow)
- Customize the Application Root Directory (default: nmon)
- Customize the TA NMON Root Directory (default: TA-nmon)
- Customize the PA NMON Root Directory (default: PA-nmon)
- Customize the local CSV Repository (default:csv_repository)
- Customize the local Config Repository (default:config_repository)
- Focus on Linux OS only by hidding other systems specific views and setting a Linux navigation menu

Using this tool over releases, you can easily manage your customizations and update the Application as usual.

Since the release 1.8.4, the application is compatible with any indexes starting with the pattern "nmon" (one or x indexes)

As such, it is not required anymore to customize the application is you intend to use one or more custom indexes, as long as the indexes names start by "nmon".

This Python tool is available in the "resources" directory of the Nmon Core Application (as gzip file, uncompress the script before launching)

Notice for updating the application: using this tool is upgrade resilient, you can create your package and repeat this operation for future release update

The tool requires Python 2.7.x or superior to operate, you can check your version with the following command:

```
python --version
```

Launching the tool with no option:

```
python Nmon_SplunkApp_Customize.py

If for some reason you need to customize the Nmon Splunk Application, please follow␣
↪these instructions:

- Download the current release of Nmon App in Splunk Base: https://apps.splunk.com/
↪app/1753
- Uncompress the Nmon_SplunkApp_Customize.py.gz
- Place the downloaded tgz Archive and this Python tool in the directory of your␣
↪choice
- Run the tool: ./customize_indexname.py and check for available options

After the execution, the application will have been customized and is ready to be used
```

Getting help with available options:

```
python Nmon_SplunkApp_Customize.py --help

usage: Nmon_SplunkApp_Customize.py [-h] [-f INFILE] [-i INDEX_NAME]
                                   [-r ROOT_DIR] [-a TA_NMON] [-p PA_NMON]
                                   [--csvrepo CSV_REPOSITORY]
                                   [--configrepo CONFIG_REPOSITORY]
                                   [--linux_only] [--version] [--debug]

optional arguments:
  -h, --help            show this help message and exit
  -f INFILE             Name of the Nmon Splunk APP tgz Archive file
  -i INDEX_NAME         Customize the Application Index Name (default: nmon)
  -r ROOT_DIR           Customize the Application Root Directory (default:
                        nmon)
  -a TA_NMON            Customize the TA NMON Root Directory (default: TA-
                        nmon)
  -p PA_NMON            Customize the PA NMON Root Directory (default: PA-
                        nmon)
  --csvrepo CSV_REPOSITORY
                        Customize the local CSV Repository (default:
                        csv_repository)
  --configrepo CONFIG_REPOSITORY
                        Customize the local Config Repository (default:
                        config_repository)
  --linux_only          Deactivate objects for other operating systems than
                        Linux (AIX / Solaris), use thisoption if you only use
                        Linux and don't want non Linux related objects to be
                        visible.
  --version             show program's version number and exit
  --debug
```

Generic example of utilization

Replace "nmon-performance-monitor-for-unix-and-linux-systems_xxx.tgz" with the exact name of the tgz archive

```
python Nmon_SplunkApp_Customize.py -f nmon-performance-monitor-for-unix-and-linux-
↪systems_xxx.tgz -i my_custom_index -r my_custom_app -a my_custom_ta -p my_custom_pa␣
↪--csvrepo my_custom_csvrepo --configrepo my_custom_configrepo
Extracting tgz Archive: nmon-performance-monitor-for-unix-and-linux-systems_175.tgz
Extracting tgz Archive: PA-nmon_1244.tgz
Extracting tgz Archive: TA-nmon_1244.tgz
Extracting tgz Archive: TA-nmon_selfmode_1244.tgz
INFO: Changing the App Root Directory from default "nmon" to custom "my_custom_app"
Achieving files transformation:
INFO: Customizing any reference to default root directory in files
Achieving files transformation:
INFO: Customizing any reference to index name in files
INFO: Customizing indexes.conf
INFO: Customizing csv_repository to my_custom_csvrepo
INFO: Customizing config_repository to my_custom_configrepo
INFO: Removing tgz resources Archives
INFO: Customizing the TA-nmon Root directory from the default TA-nmon to my_custom_ta
INFO: ************* Tar creation done of: my_custom_ta_custom.tgz *************
INFO: Removing tgz resources Archives
INFO: Customizing the PA-nmon Root directory from the default PA-nmon to my_custom_pa
INFO: ************* Tar creation done of: my_custom_pa_custom.tgz *************
```

(continues on next page)

(continued from previous page)

```
INFO: Creating the custom nmon_performance_monitor_custom.spl archive in current root␣
↪directory
INFO: ************* Tar creation done of: nmon_performance_monitor_custom.spl␣
↪*************

*** To install your customized packages: ***

 - Extract the content of nmon_performance_monitor_custom.spl to Splunk Apps␣
↪directory of your search heads (or use the manager to install the App)
 - Extract the content of the PA package available in resources directory to your␣
↪indexers
 - Extract the content of the TA package available in resources directory to your␣
↪deployment server or clients

Operation terminated.
```

Linux OS example: build an app for Linux OS support only

```
python Nmon_SplunkApp_Customize.py -f nmon-performance-monitor-for-unix-and-linux-
↪systems_xxx.tgz --linux_only

INFO: No custom index name were provided, using default "nmon" name for index
INFO: No custom root directory of the nmon App core App were provided, using default
↪"nmon" name for root directory
INFO: No custom root directory of the TA-nmon were provided, using default "TA-nmon"␣
↪name for TA-nmon root directory
INFO: No custom root directory of the PA-nmon were provided, using default "PA-nmon"␣
↪name for PA-nmon root directory
INFO: No custom csv reposity directory were provided, using default "csv_repository"␣
↪name for csv repository root directory
INFO: No custom csv reposity directory were provided, using default "config_repository
↪" name for csv repository root directory
Extracting tgz Archive: nmon-performance-monitor-for-unix-and-linux-systems_175.tgz
Extracting tgz Archive: PA-nmon_1244.tgz
Extracting tgz Archive: TA-nmon_1244.tgz
Extracting tgz Archive: TA-nmon_selfmode_1244.tgz
INFO: Operating systems support, AIX operating system related objects have been␣
↪deactivated
INFO: Operating systems support, Solaris operating system related objects have been␣
↪deactivated
INFO: Linux only management, activate Linux only navigation
INFO: Creating the custom nmon_performance_monitor_custom.spl archive in current root␣
↪directory
INFO: ************* Tar creation done of: nmon_performance_monitor_custom.spl␣
↪*************

*** To install your customized packages: ***

 - Extract the content of nmon_performance_monitor_custom.spl to Splunk Apps␣
↪directory of your search heads (or use the manager to install the App)
 - Extract the content of the PA package available in resources directory to your␣
↪indexers
 - Extract the content of the TA package available in resources directory to your␣
↪deployment server or clients
```

(continues on next page)

(continued from previous page)

```
Operation terminated.
```

04 - Scenarios of advanced customizations

Advanced Customization

01 - Splitting index for different users populations

Vagrant testing:

Easily test this deployment scenario with Vagrant and Ansible !

See: https://github.com/guilhemmarchand/splunk-vagrant-ansible-collections

The goal:

The goal of this scenario is to ingest nmon data coming from different data centers that will be managed by different Unix administrator teams. As such, each user of those teams will be able to see and analyse only the data of the servers under their management.

For the demonstration purpose, we will assume:

- **Data center 1: datacenter_US**
  - **index name:** nmon_perf_unix_datacenter_US
  - **Technical addon name:** TA-nmon-datacenter-US
  - **role name for Unix admin users:** team-unix-admin-us
- **Data center 2: datacenter_UK**
  - **index name:** nmon_perf_unix_datacenter_UK
  - **Technical addon name:** TA-nmon-datacenter-UK
  - **role name for Unix admin users:** team-unix-admin-uk

STEP 1: Prepare your indexers

For the demonstration purpose, we assume having a single standalone indexer receiving data from both data centers, with the following indexes.conf:

```
[nmon_perf_unix_datacenter_UK]
coldPath = $SPLUNK_DB/nmon_perf_unix_datacenter_UK/colddb
homePath = $SPLUNK_DB/nmon_perf_unix_datacenter_UK/db
thawedPath = $SPLUNK_DB/nmon_perf_unix_datacenter_UK/thaweddb

[nmon_perf_unix_datacenter_US]
coldPath = $SPLUNK_DB/nmon_perf_unix_datacenter_US/colddb
homePath = $SPLUNK_DB/nmon_perf_unix_datacenter_US/db
thawedPath = $SPLUNK_DB/nmon_perf_unix_datacenter_US/thaweddb
```

STEP 2: Prepare the TA-nmon packages

We will want to have 2 different versions of the TA-nmon, one for each data center.

For the example purpose, I will assume you upload the tgz archive to /tmp

Create the packages:

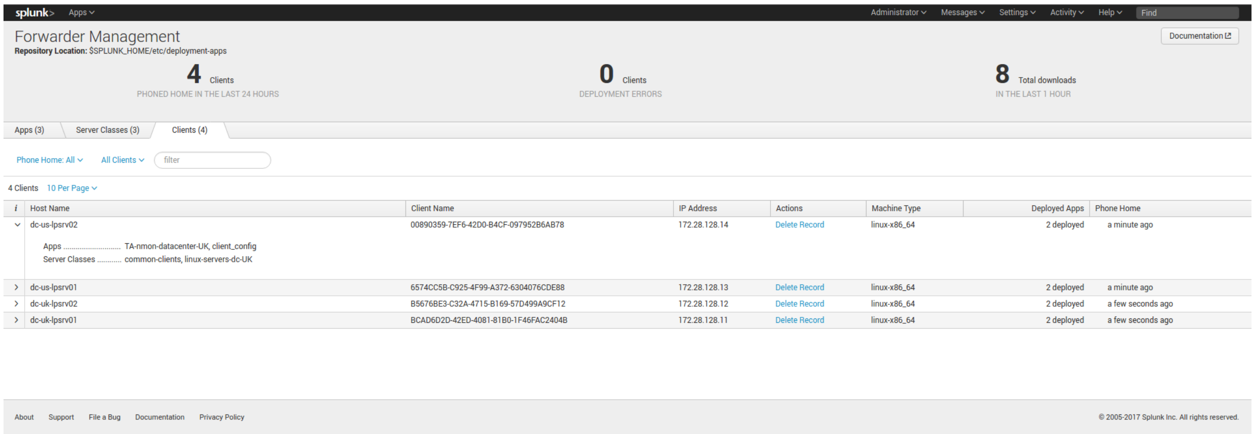
```
python create_agent.py --indexname nmon_perf_unix_datacenter_US --agentname TA-nmon-
↪datacenter-US -f /tmp/TA-nmon_<VERSION>.tgz
python create_agent.py --indexname nmon_perf_unix_datacenter_UK --agentname TA-nmon-
↪datacenter-UK -f /tmp/TA-nmon_<VERSION>.tgz
```

This will generate 2 TA-nmon packages to be deployed to each group of data center servers:

```
TA-nmon-datacenter-UK.tgz
TA-nmon-datacenter-US.tgz
```

STEP 3: Deploy the TA-nmon packages

Configure your deployment servers to deploy the packages to your servers:


splunk> Apps
Administrator Messages Settings Activity Help Find
Documentation
Apps (3) Server Classes (3) Clients (4)
Phone Home: All All Clients filter
10 Per Page
About Support File a Bug Documentation Privacy Policy
© 2005-2017 Splunk Inc. All rights reserved.

Forwarder Management

Repository Location: $SPLUNK_HOME/etc/deployment-apps

4 Clients PHONED HOME IN THE LAST 24 HOURS

0 Clients DEPLOYMENT ERRORS

8 Total downloads IN THE LAST 1 HOUR

4 Clients

i	Host Name	Client Name	IP Address	Actions	Machine Type	Deployed Apps	Phone Home
⌄	dc-us-lpsrv02 Apps .... TA-nmon-datacenter-UK, client_config Server Classes .... common-clients, linux-servers-dc-UK	00890359-7EF6-42D0-B4CF-097952B6AB78	172.28.128.14	Delete Record	linux-x86_64	2 deployed	a minute ago
›	dc-us-lpsrv01	6574CC5B-C925-4F99-A372-6304076CDE88	172.28.128.13	Delete Record	linux-x86_64	2 deployed	a minute ago
›	dc-uk-lpsrv02	B5676BE3-C32A-4715-B169-57D499A9CF12	172.28.128.12	Delete Record	linux-x86_64	2 deployed	a few seconds ago
›	dc-uk-lpsrv01	BCAD6D2D-42ED-4081-81B0-1F46FAC2404B	172.28.128.11	Delete Record	linux-x86_64	2 deployed	a few seconds ago

STEP 4: On the search heads, configure the roles and users

We will create 2 roles, each role inherits from the default user role and provides access to the relevant indexes.

Because by default the user role provides access to any indexes, you will want as well to restrict it:

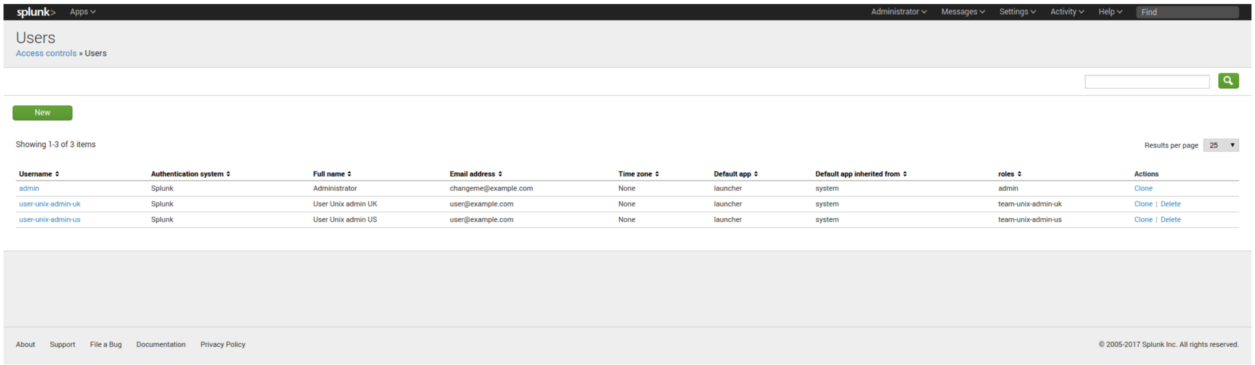
local authorized.conf content:

```
[role_team-unix-admin-us]
importRoles = user
srchIndexesAllowed = nmon_perf_unix_datacenter_US
srchIndexesDefault = nmon_perf_unix_datacenter_US

[role_team-unix-admin-uk]
importRoles = user
srchIndexesAllowed = nmon_perf_unix_datacenter_UK
srchIndexesDefault = nmon_perf_unix_datacenter_UK

# Restrict standard user role to main index only
[role_user]
srchIndexesAllowed = main
```

Finally, have your users belonging to the relevant roles, for the demonstration purpose:



Users

Access controls » Users

Showing 1-3 of 3 items

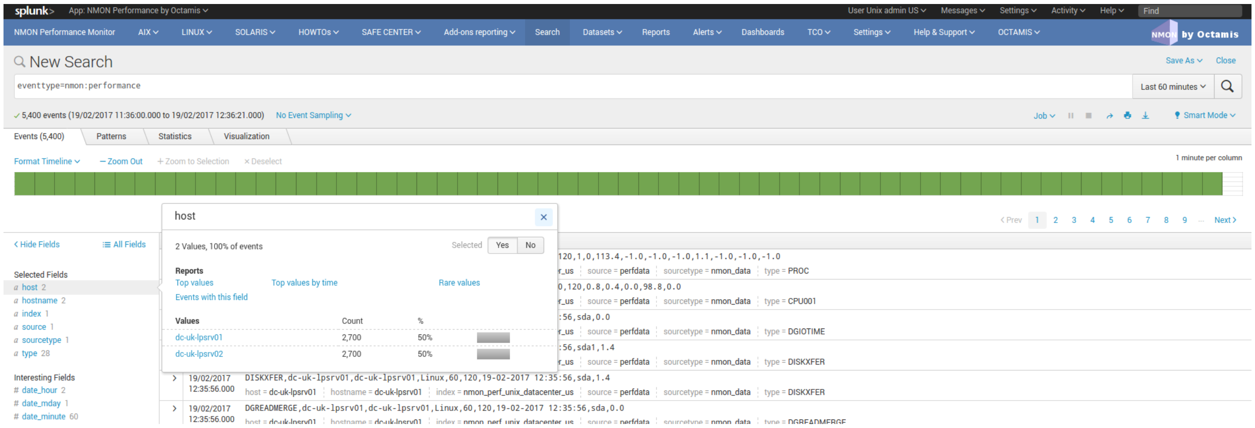
Results per page 25

Username	Authentication system	Full name	Email address	Time zone	Default app	Default app inherited from	roles	Actions
admin	Splunk	Administrator	changeme@example.com	None	launcher	system	admin	Clone
user-unix-admin-uk	Splunk	User Unix admin UK	user@example.com	None	launcher	system	team-unix-admin-uk	Clone | Delete
user-unix-admin-us	Splunk	User Unix admin US	user@example.com	None	launcher	system	team-unix-admin-us	Clone | Delete

About Support File a Bug Documentation Privacy Policy
© 2005-2017 Splunk Inc. All rights reserved.

FINAL: Splunk is ready

A user belonging to the role "team-unix-admin-us" will only see and access to data from the US data center:


splunk> App: NMON Performance by Octamis
User Unix admin US Messages Settings Activity Help Find
NMON Performance Monitor AIX LINUX SOLARIS HOWTOs SAFE CENTER Add-ons reporting Search Datasets Reports Alerts Dashboards TCO Settings Help & Support OCTAMIS
NMON by Octamis
New Search
Save As Close
Last 60 minutes
Job
Smart Mode
Events (5,400) Patterns Statistics Visualization
Format Timeline − Zoom Out + Zoom to Selection × Deselect
1 minute per column
‹ Prev 1 2 3 4 5 6 7 8 9 … Next ›
‹ Hide Fields ☰ All Fields
host
Selected Yes No
Reports
Top values Top values by time Rare values
Events with this field

```
eventtype=nmon:performance
```

✓ 5,400 events (19/02/2017 11:36:00.000 to 19/02/2017 12:36:21.000) No Event Sampling

2 Values, 100% of events

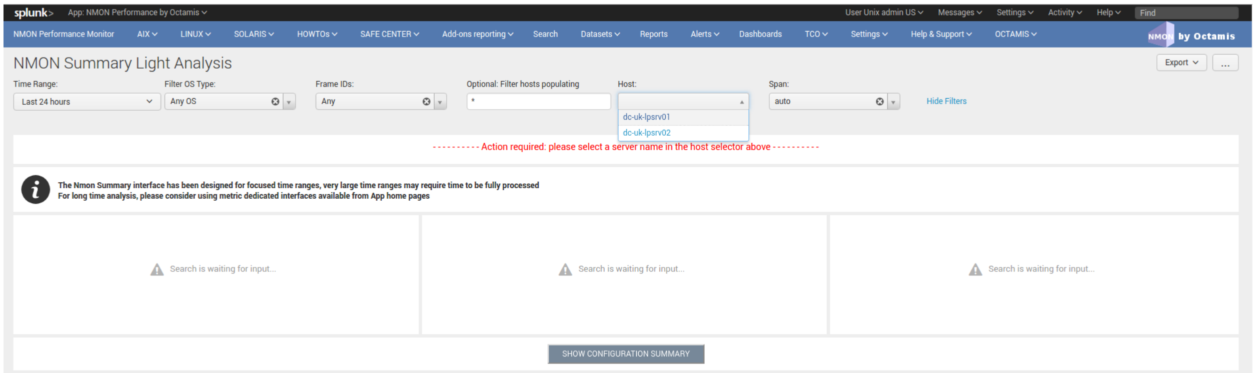
Values	Count	%
dc-uk-lpsrv01	2,700	50%
dc-uk-lpsrv02	2,700	50%

Selected Fields
- a host 2
- a hostname 2
- a index 1
- a source 1
- a sourcetype 1
- a type 28

Interesting Fields
- # date_hour 2
- # date_mday 1
- # date_minute 60

```
120,1,0,113.4,-1.0,-1.0,-1.0,1.1,-1.0,-1.0,-1.0
r_us | source = perfdata | sourcetype = nmon_data | type = PROC
0,120,0.8,0.4,0.0,98.8,0.0
r_us | source = perfdata | sourcetype = nmon_data | type = CPU001
:56,sda,0.0
r_us | source = perfdata | sourcetype = nmon_data | type = DGIOTIME
:56,sda1,1.4
r_us | source = perfdata | sourcetype = nmon_data | type = DISKXFER
> 19/02/2017    DISKXFER,dc-uk-lpsrv01,dc-uk-lpsrv01,Linux,60,120,19-02-2017 12:35:56,sda,1.4
  12:35:56.000  host = dc-uk-lpsrv01 | hostname = dc-uk-lpsrv01 | index = nmon_perf_unix_datacenter_us | source = perfdata | sourcetype = nmon_data | type = DISKXFER
> 19/02/2017    DGREADMERGE,dc-uk-lpsrv01,dc-uk-lpsrv01,Linux,60,120,19-02-2017 12:35:56,sda,0.0
  12:35:56.000  host = dc-uk-lpsrv01 | hostname = dc-uk-lpsrv01 | index = nmon_perf_unix_datacenter_us | source = perfdata | sourcetype = nmon_data | type = DGREADMERGE
```

And a user belonging to the role "team-unix-admin-uk" will have access to servers from the UK data center only:


splunk> App: NMON Performance by Octamis
User Unix admin US Messages Settings Activity Help Find
NMON Performance Monitor AIX LINUX SOLARIS HOWTOs SAFE CENTER Add-ons reporting Search Datasets Reports Alerts Dashboards TCO Settings Help & Support OCTAMIS
NMON by Octamis
Export ...
Hide Filters
SHOW CONFIGURATION SUMMARY

NMON Summary Light Analysis

Time Range: Last 24 hours

Filter OS Type: Any OS

Frame IDs: Any

Optional: Filter hosts populating: *

Host: dc-uk-lpsrv01, dc-uk-lpsrv02

Span: auto

---------- Action required: please select a server name in the host selector above ----------

The Nmon Summary interface has been designed for focused time ranges, very large time ranges may require time to be fully processed
For long time analysis, please consider using metric dedicated interfaces available from App home pages

Search is waiting for input...

Search is waiting for input...

Search is waiting for input...

2.17.6 Troubleshoot

01 - Troubleshooting guide from A to Z

Troubleshooting guide for Nmon Performance Monitor

So you've got trouble ? This guide will help in troubleshooting every piece of the Nmon Perf Application from the very beginning!

Note that this guide is oriented in distributed deployment scenario, such that it focuses on issues you may encounter between Splunk and end servers

STEP 1: Checking Splunk internal events

Checking Splunk internal events from your remote host (Universal or Heavy Forwarders) to Splunk

In case of trouble with remote hosts , you should always start by verifying that you successfully receive Splunk internal events from them, this is a simple verification that validates:

- That your remote hosts are able to send data to your indexers
- That your basic deployment items (such as outputs.conf) are correctly configured

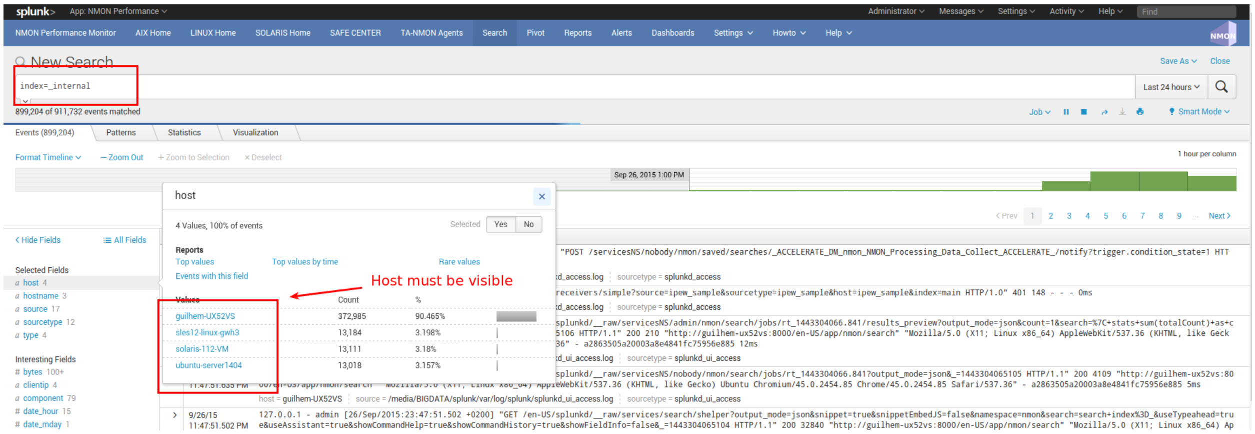
When a remote host running Splunk (Universal or Heavy forwarder) is connected to a Splunk infrastructure, it will always send its internal events into various internal indexes:

- _internal
- _audit
- _introspection

Between other log files, the main log you should care about is the "splunkd.log", you will find it in the "_internal" index, this is the data i strongly recommend to check

Ensure you successfully receive Splunk internal data:

INFORMATION: In default configuration, internal indexes cannot be accessed by standard users (unless Splunk admin gives the access rights), this step requires admin access or access authorization to internal indexes


splunk> App: NMON Performance
Administrator Messages Settings Activity Help Find
NMON Performance Monitor AIX Home LINUX Home SOLARIS Home SAFE CENTER TA-NMON Agents Search Pivot Reports Alerts Dashboards Settings Howto Help
New Search
Save As Close
Last 24 hours
Job Smart Mode
Events (899,204) Patterns Statistics Visualization
Format Timeline Zoom Out Zoom to Selection Deselect
1 hour per column
Sep 26, 2015 1:00 PM
Prev 1 2 3 4 5 6 7 8 9 ... Next
Hide Fields All Fields
Selected Fields
host 4
hostname 3
source 17
sourcetype 12
type 4
Interesting Fields
bytes 100+
clientip 4
component 79
date_hour 15
date_mday 1

```
index=_internal
```

899,204 of 911,732 events matched

host

4 Values, 100% of events — Selected: Yes No

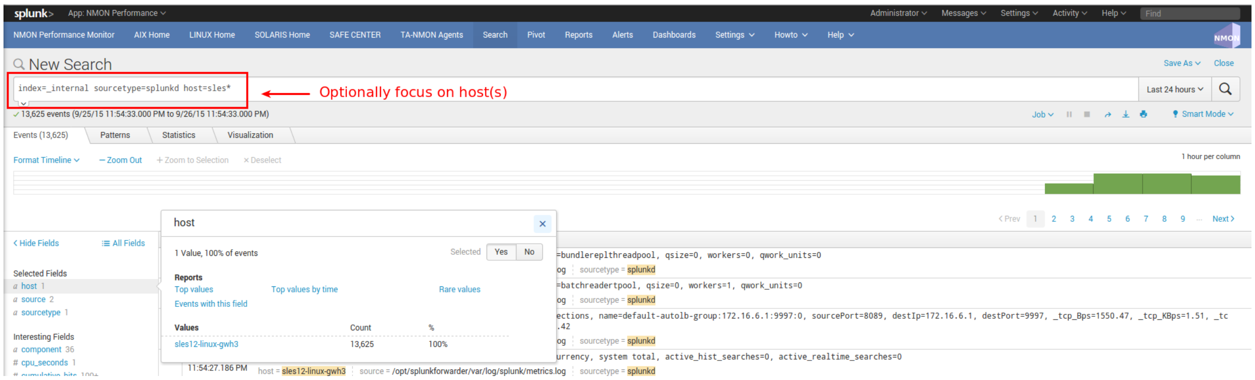
Reports: Top values · Top values by time · Rare values · Events with this field

Host must be visible

Values	Count	%
guilhem-UX52VS	372,985	90.465%
sles12-linux-gwh3	13,184	3.198%
solaris-112-VM	13,111	3.18%
ubuntu-server1404	13,018	3.157%

```
"POST /servicesNS/nobody/nmon/saved/searches/_ACCELERATE_DM_nmon_NMON_Processing_Data_Collect_ACCELERATE_/notify?trigger.condition_state=1 HTT
d_access.log   sourcetype = splunkd_access
receivers/simple?source=ipew_sample&sourcetype=ipew_sample&host=ipew_sample&index=main HTTP/1.0" 401 148 - - - 0ms
d_access.log   sourcetype = splunkd_access
splunkd/__raw/servicesNS/admin/nmon/search/jobs/rt_1443304066.841/results_preview?output_mode=json&count=1&search=%7C+stats+sum(totalCount)+as+c
5106 HTTP/1.1" 200 210 "http://guilhem-ux52vs:8000/en-US/app/nmon/search" "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Geck
36" - a2863505a20003a8e4841fc75956e885 12ms
d_ui_access.log   sourcetype = splunkd_ui_access
splunkd/__raw/servicesNS/nobody/nmon/search/jobs/rt_1443304066.841?output_mode=json&_=1443304065105 HTTP/1.1" 200 4109 "http://guilhem-ux52vs:80
11:47:51.635 PM   00/en-US/app/nmon/search"   Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Ubuntu Chromium/45.0.2454.85 Chrome/45.0.2454.85 Safari/537.36" - a2863505a20003a8e4841fc75956e885 5ms
host = guilhem-UX52VS   source = /media/BIGDATA/splunk/var/log/splunk/splunkd_ui_access.log   sourcetype = splunkd_ui_access
> 9/26/15         127.0.0.1 - admin [26/Sep/2015:23:47:51.502 +0200] "GET /en-US/splunkd/__raw/services/search/shelper?output_mode=json&snippet=true&snippetEmbedJS=false&namespace=nmon&search=search+index%3D_&useTypeahead=tru
11:47:51.502 PM   e&useAssistant=true&showCommandHelp=true&showCommandHistory=true&showFieldInfo=false&_=1443304065104 HTTP/1.1" 200 32840 "http://guilhem-ux52vs:8000/en-US/app/nmon/search" "Mozilla/5.0 (X11; Linux x86_64) Ap
```

Optionally focus on splunkd sourcetype and host(s) you are verifying:


splunk> App: NMON Performance
Administrator Messages Settings Activity Help Find
NMON Performance Monitor AIX Home LINUX Home SOLARIS Home SAFE CENTER TA-NMON Agents Search Pivot Reports Alerts Dashboards Settings Howto Help
New Search
Save As Close
Last 24 hours
Job Smart Mode
Events (13,625) Patterns Statistics Visualization
Format Timeline Zoom Out Zoom to Selection Deselect
1 hour per column
Prev 1 2 3 4 5 6 7 8 9 ... Next
Hide Fields All Fields
Selected Fields
host 1
source 2
sourcetype 1
Interesting Fields
component 36
cpu_seconds 1

```
index=_internal sourcetype=splunkd host=sles*
```

Optionally focus on host(s)

✓ 13,625 events (9/25/15 11:54:33.000 PM to 9/26/15 11:54:33.000 PM)

host

1 Value, 100% of events — Selected: Yes No

Reports: Top values · Top values by time · Rare values · Events with this field

Values	Count	%
sles12-linux-gwh3	13,625	100%

```
=bundlereplthreadpool, qsize=0, workers=0, qwork_units=0
og   sourcetype = splunkd
=batchreadertpool, qsize=0, workers=1, qwork_units=0
og   sourcetype = splunkd
ections, name=default-autolb-group:172.16.6.1:9997:0, sourcePort=8089, destIp=172.16.6.1, destPort=9997, _tcp_Bps=1550.47, _tcp_KBps=1.51, _tc
.42
og   sourcetype = splunkd
urrency, system total, active_hist_searches=0, active_realtime_searches=0
11:54:27.186 PM   host = sles12-linux-gwh3   source = /opt/splunkforwarder/var/log/splunk/metrics.log   sourcetype = splunkd
```

—> If you successfully found incoming events for your host(s), swith to step 2

—> If you can't find incoming events for your host(s), common root causes can be:

- Network connection failure between you host(s) and indexers (or intermediate collecters): Verify with a simple telnet connection test that you can access to destination IP and port(s)
- Bad information in outputs.conf (check IP / Port, syntax)
- No outputs.conf deployed to Universal or Heavy Forwarder

In such a case, connect directly to the host and verify messages in /opt/splunkforwarder/var/log/splunkd.log

STEP 2: Verify the TA-nmon behaviors

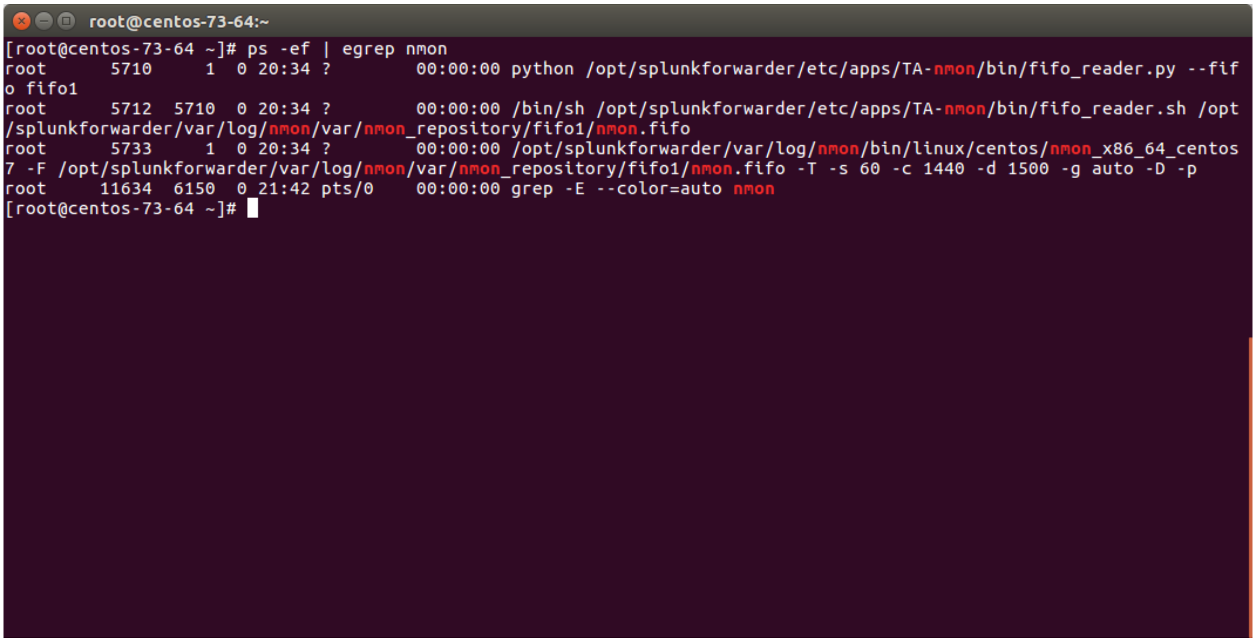
This section refers to the TA-nmon trouble shooting guide: http://ta-nmon.readthedocs.io/en/latest/troubleshoot.html

Expected running processes

Since the 1.3.x branch, you should find various processes running:

- 1 x nmon process (or 2 x nmon processes during the parallel interval)
- 1 x main Perl or Python fifo_reader process (or 2 x processes during the parallel interval)
- 1 x subshell fifo_reader process (or 2 x processes during the parallel interval)

On a Linux box:



```
[root@centos-73-64 ~]# ps -ef | egrep nmon
root      5710     1  0 20:34 ?        00:00:00 python /opt/splunkforwarder/etc/apps/TA-nmon/bin/fifo_reader.py --fif
o fifo1
root      5712  5710  0 20:34 ?        00:00:00 /bin/sh /opt/splunkforwarder/etc/apps/TA-nmon/bin/fifo_reader.sh /opt
/splunkforwarder/var/log/nmon/var/nmon_repository/fifo1/nmon.fifo
root      5733     1  0 20:34 ?        00:00:00 /opt/splunkforwarder/var/log/nmon/bin/linux/centos/nmon_x86_64_centos
7 -F /opt/splunkforwarder/var/log/nmon/var/nmon_repository/fifo1/nmon.fifo -T -s 60 -c 1440 -d 1500 -g auto -D -p
root     11634  6150  0 21:42 pts/0    00:00:00 grep -E --color=auto nmon
[root@centos-73-64 ~]# 
```

On AIX, the nmon process will be called "topas-nmon"

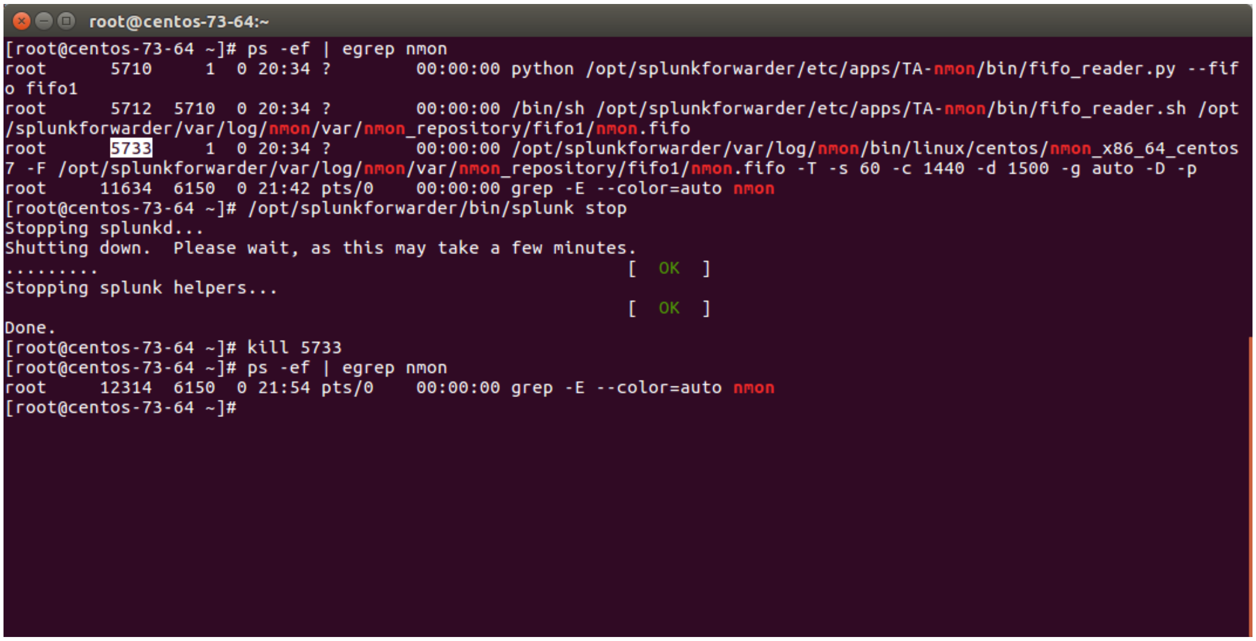
On Solaris, the sarmon process will be called "sadc"

Starting processes

If you run in trouble and want to troubleshoot the situation, the easiest approach is stopping Splunk, kill existing nmon process and run the tasks manually:

- Stop Splunk and kill the nmon process:

```
./splunk stop
```



```
[root@centos-73-64 ~]# ps -ef | egrep nmon
root      5710     1  0 20:34 ?        00:00:00 python /opt/splunkforwarder/etc/apps/TA-nmon/bin/fifo_reader.py --fif
o fifo1
root      5712  5710  0 20:34 ?        00:00:00 /bin/sh /opt/splunkforwarder/etc/apps/TA-nmon/bin/fifo_reader.sh /opt
/splunkforwarder/var/log/nmon/var/nmon_repository/fifo1/nmon.fifo
root      5733     1  0 20:34 ?        00:00:00 /opt/splunkforwarder/var/log/nmon/bin/linux/centos/nmon_x86_64_centos
7 -F /opt/splunkforwarder/var/log/nmon/var/nmon_repository/fifo1/nmon.fifo -T -s 60 -c 1440 -d 1500 -g auto -D -p
root     11634  6150  0 21:42 pts/0    00:00:00 grep -E --color=auto nmon
[root@centos-73-64 ~]# /opt/splunkforwarder/bin/splunk stop
Stopping splunkd...
Shutting down.  Please wait, as this may take a few minutes.
.........                                                  [  OK  ]
Stopping splunk helpers...
                                                           [  OK  ]
Done.
[root@centos-73-64 ~]# kill 5733
[root@centos-73-64 ~]# ps -ef | egrep nmon
root     12314  6150  0 21:54 pts/0    00:00:00 grep -E --color=auto nmon
[root@centos-73-64 ~]#
```

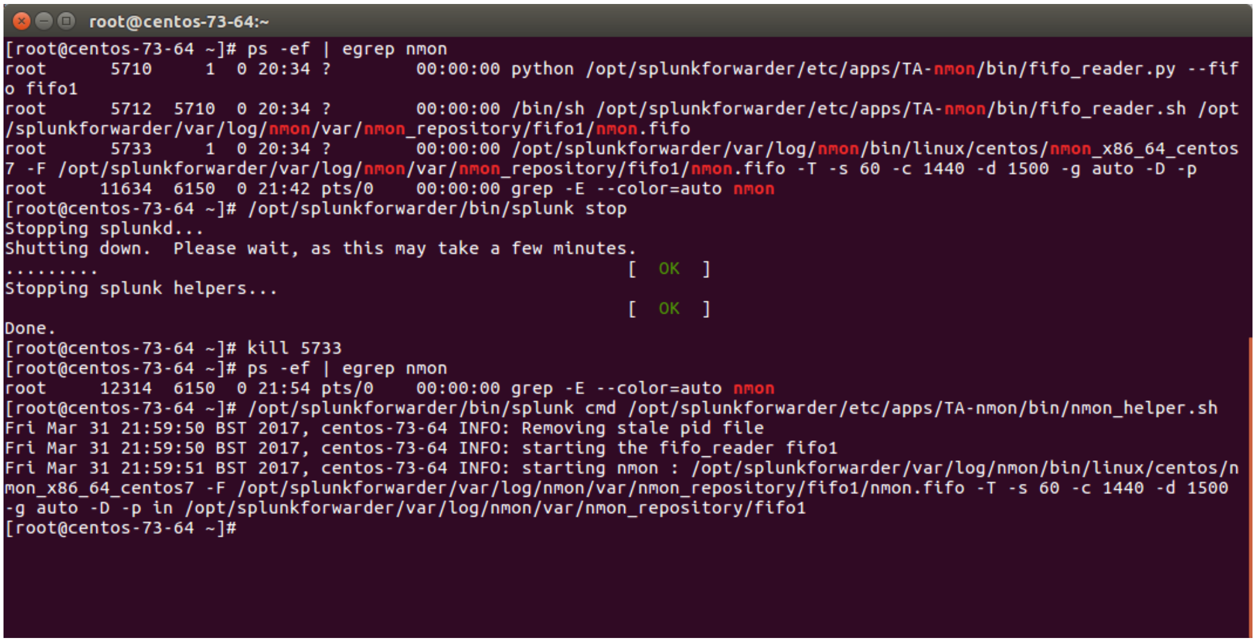
You will observe that killing the nmon process will automatically terminate the fifo_reader.pll.py and the subshell fifo_reader.sh. This the expected behavior, and mandatory.

If the processes do not stop, then your problem became mine and please open an issue !

- Now we can manually starting the processes, example:

```
/opt/splunkforwarder/bin/splunk cmd /opt/splunkforwarder/etc/apps/TA-nmon/bin/
↪nmon_helper.sh
```

Please adapt the paths to your context



```
[root@centos-73-64 ~]# ps -ef | egrep nmon
root      5710     1  0 20:34 ?        00:00:00 python /opt/splunkforwarder/etc/apps/TA-nmon/bin/fifo_reader.py --fif
o fifo1
root      5712  5710  0 20:34 ?        00:00:00 /bin/sh /opt/splunkforwarder/etc/apps/TA-nmon/bin/fifo_reader.sh /opt
/splunkforwarder/var/log/nmon/var/nmon_repository/fifo1/nmon.fifo
root      5733     1  0 20:34 ?        00:00:00 /opt/splunkforwarder/var/log/nmon/bin/linux/centos/nmon_x86_64_centos
7 -F /opt/splunkforwarder/var/log/nmon/var/nmon_repository/fifo1/nmon.fifo -T -s 60 -c 1440 -d 1500 -g auto -D -p
root     11634  6150  0 21:42 pts/0    00:00:00 grep -E --color=auto nmon
[root@centos-73-64 ~]# /opt/splunkforwarder/bin/splunk stop
Stopping splunkd...
Shutting down.  Please wait, as this may take a few minutes.
.........                                                  [  OK  ]
Stopping splunk helpers...
                                                           [  OK  ]
Done.
[root@centos-73-64 ~]# kill 5733
[root@centos-73-64 ~]# ps -ef | egrep nmon
root     12314  6150  0 21:54 pts/0    00:00:00 grep -E --color=auto nmon
[root@centos-73-64 ~]# /opt/splunkforwarder/bin/splunk cmd /opt/splunkforwarder/etc/apps/TA-nmon/bin/nmon_helper.sh
Fri Mar 31 21:59:50 BST 2017, centos-73-64 INFO: Removing stale pid file
Fri Mar 31 21:59:50 BST 2017, centos-73-64 INFO: starting the fifo_reader fifo1
Fri Mar 31 21:59:51 BST 2017, centos-73-64 INFO: starting nmon : /opt/splunkforwarder/var/log/nmon/bin/linux/centos/n
mon_x86_64_centos7 -F /opt/splunkforwarder/var/log/nmon/var/nmon_repository/fifo1/nmon.fifo -T -s 60 -c 1440 -d 1500
-g auto -D -p in /opt/splunkforwarder/var/log/nmon/var/nmon_repository/fifo1
[root@centos-73-64 ~]#
```

Let’s summarize what happened here:

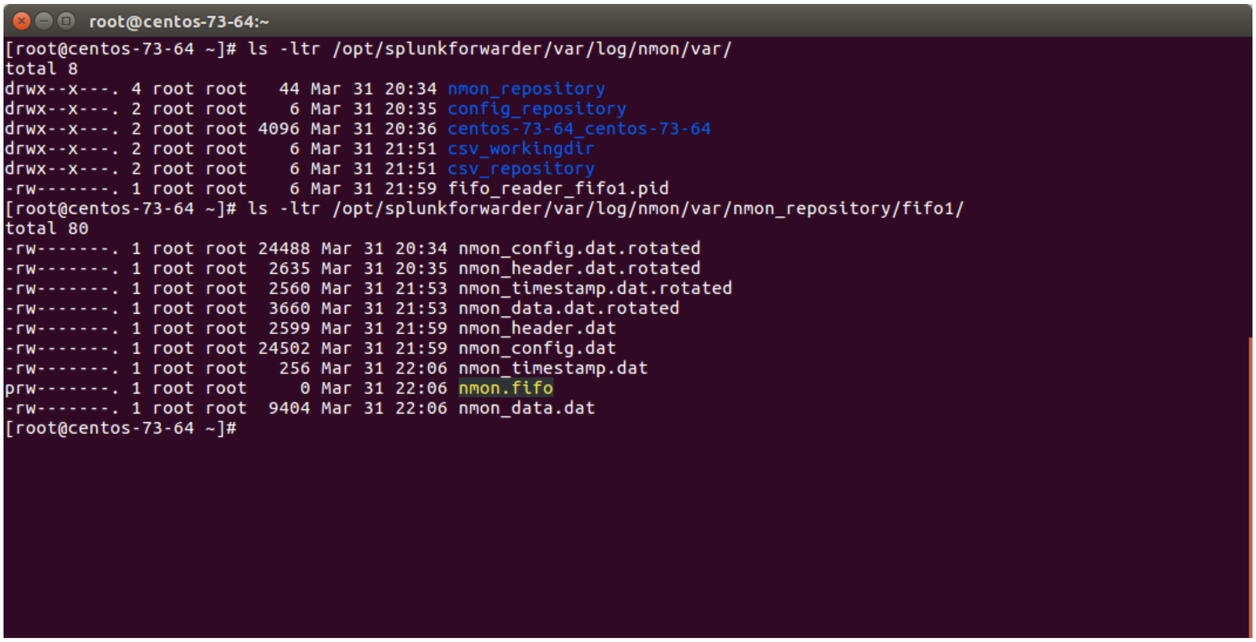
- nmon_helper.sh starts the fifo reader, if there is no fifo_reader running, the “fifo1” process will be started
- the fifo_reader.pll.py starts a fifo_reader.sh process in the background
- nmon_helper.sh starts the nmon process which will write its data to the relevant fifo file
- the nmon process cannot start if the fifo_reader has not started

If something unexpected happens and that the fifo_reader and nmon process do not start normally, you may want to trouble shoot the nmon_helper.sh script.

You can do very easily by commenting out “# set -x”, re-run the script and analyse the output. (you might need to add the set-x within the functions as well)

Checking fifo_reader processes

The fifo_reader processes will continuously read the fifo file writen by the nmon process, and generate various dat files that represent the different typologies of nmon data:



```
[root@centos-73-64 ~]# ls -ltr /opt/splunkforwarder/var/log/nmon/var/
total 8
drwx--x---. 4 root root   44 Mar 31 20:34 nmon_repository
drwx--x---. 2 root root    6 Mar 31 20:35 config_repository
drwx--x---. 2 root root 4096 Mar 31 20:36 centos-73-64_centos-73-64
drwx--x---. 2 root root    6 Mar 31 21:51 csv_workingdir
drwx--x---. 2 root root    6 Mar 31 21:51 csv_repository
-rw-------. 1 root root    6 Mar 31 21:59 fifo_reader_fifo1.pid
[root@centos-73-64 ~]# ls -ltr /opt/splunkforwarder/var/log/nmon/var/nmon_repository/fifo1/
total 80
-rw-------. 1 root root 24488 Mar 31 20:34 nmon_config.dat.rotated
-rw-------. 1 root root  2635 Mar 31 20:35 nmon_header.dat.rotated
-rw-------. 1 root root  2560 Mar 31 21:53 nmon_timestamp.dat.rotated
-rw-------. 1 root root  3660 Mar 31 21:53 nmon_data.dat.rotated
-rw-------. 1 root root  2599 Mar 31 21:59 nmon_header.dat
-rw-------. 1 root root 24502 Mar 31 21:59 nmon_config.dat
-rw-------. 1 root root   256 Mar 31 22:06 nmon_timestamp.dat
prw-------. 1 root root     0 Mar 31 22:06 nmon.fifo
-rw-------. 1 root root  9404 Mar 31 22:06 nmon_data.dat
[root@centos-73-64 ~]#
```

How this it work?

- The fifo_reader.sh reads every new line of data writen to the fifo file (named pipe) and sends the data to the fifo_reader.pll.py
- The fifo_reader.pll.py parses the lines and applies various regular expressions to decide where to write the data, depending on its content
- If there were existing *.dat files at the startup of the fifo_reader processes, those dat files are rotated and renamed to* “.rotated”
- The nmon.fifo is not regular file but a named pipe (observe the “prw——-“), its size will always be equal to 0

Checking the data parsing

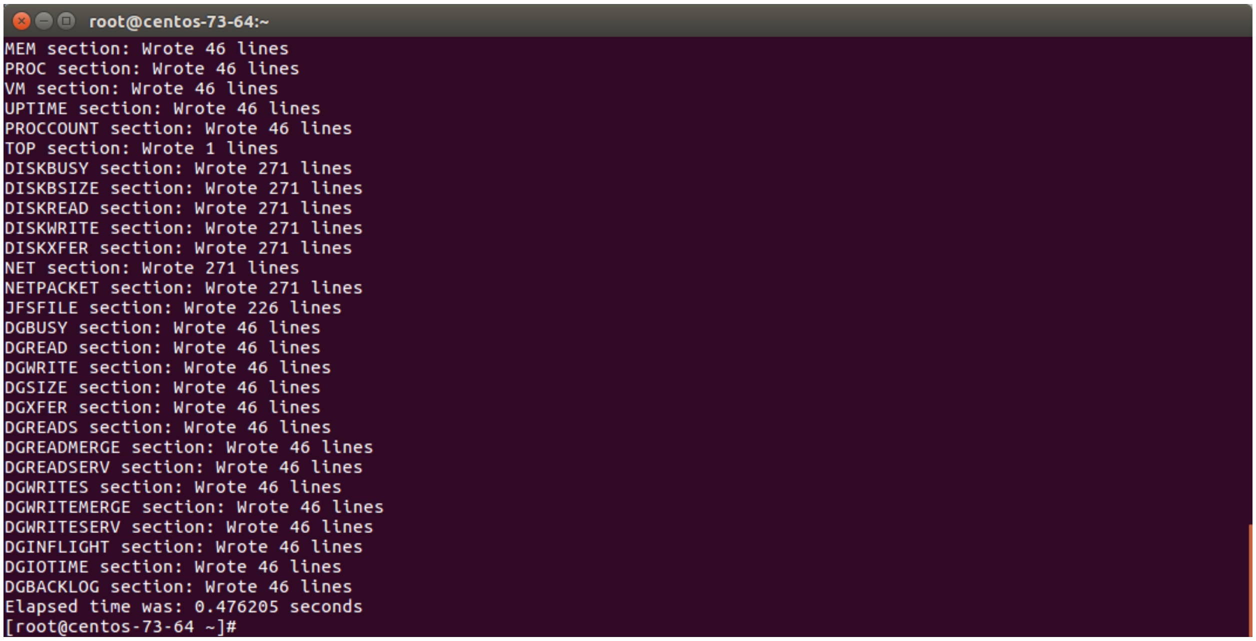
The parsing of those dat files is being achieved in 2 main steps:

- The "bin/fifo_consumer.sh" script is started every 60 seconds by Splunk
- This script will check if an nmon_data.dat file exists and that its size is greater than 0
- If the size of the nmon_dat.data file equals to 0, then the fifo_consumer.sh has nothing to do and will exit this fifo file
- If the size is greater than 0 but its modification time (mtime) is less than 5 seconds, the script will loop until the condition is true
- The fifo_consumer.sh reads the dat file, recompose the nmon file and stream its content to the "bin/nmon2csh.sh" shell wrapper
- After this operation, the nmon_data.dat file will be empty for the next cycle
- The shell wrapper reads in stdin the data, and send it to the nmon2csv parser (bin/nmon2csv.pll.py)
- The parser reads the nmon data, parses it and produces the final files to be indexed by Splunk

Easy no ;-)

You can easily run the fifo_consumer.sh manually:

```
/opt/splunkforwarder/bin/splunk cmd /opt/splunkforwarder/etc/apps/TA-nmon/bin/fifo_
↪consumer.sh
```



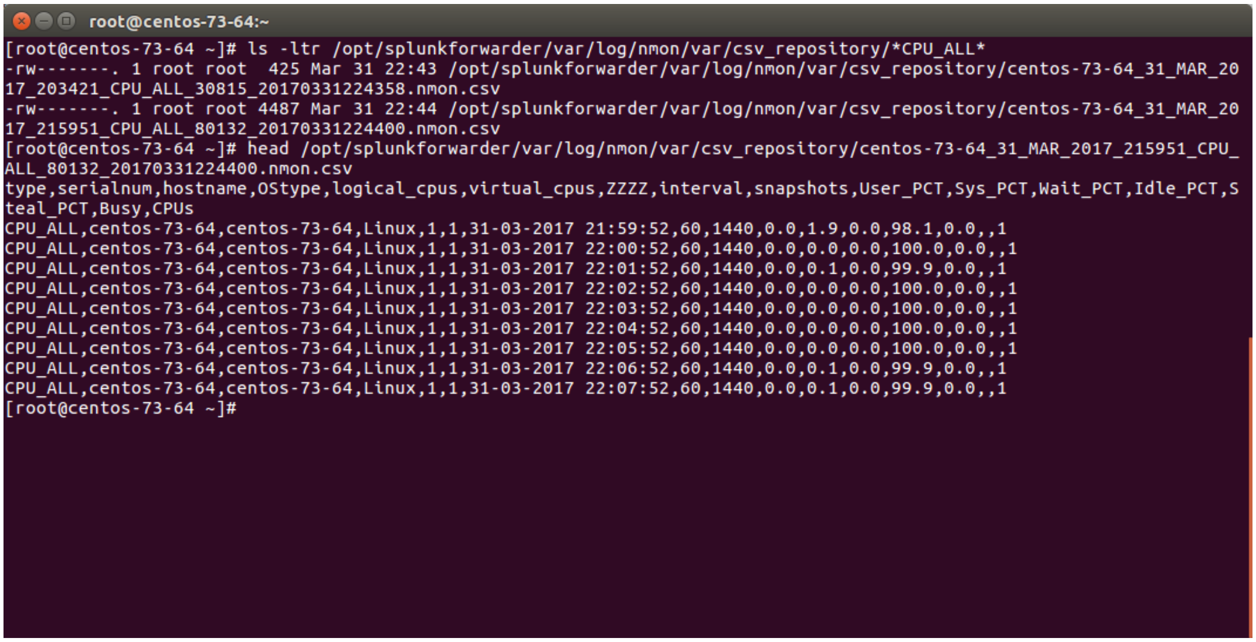
root@centos-73-64:~

```
MEM section: Wrote 46 lines
PROC section: Wrote 46 lines
VM section: Wrote 46 lines
UPTIME section: Wrote 46 lines
PROCCOUNT section: Wrote 46 lines
TOP section: Wrote 1 lines
DISKBUSY section: Wrote 271 lines
DISKBSIZE section: Wrote 271 lines
DISKREAD section: Wrote 271 lines
DISKWRITE section: Wrote 271 lines
DISKXFER section: Wrote 271 lines
NET section: Wrote 271 lines
NETPACKET section: Wrote 271 lines
JFSFILE section: Wrote 226 lines
DGBUSY section: Wrote 46 lines
DGREAD section: Wrote 46 lines
DGWRITE section: Wrote 46 lines
DGSIZE section: Wrote 46 lines
DGXFER section: Wrote 46 lines
DGREADS section: Wrote 46 lines
DGREADMERGE section: Wrote 46 lines
DGREADSERV section: Wrote 46 lines
DGWRITES section: Wrote 46 lines
DGWRITEMERGE section: Wrote 46 lines
DGWRITESERV section: Wrote 46 lines
DGINFLIGHT section: Wrote 46 lines
DGIOTIME section: Wrote 46 lines
DGBACKLOG section: Wrote 46 lines
Elapsed time was: 0.476205 seconds
[root@centos-73-64 ~]#
```

The files to be indexed by Splunk can be found in:

```
$SPLUNK_HOME/var/log/nmon/var/csv_repository
$SPLUNK_HOME/var/log/nmon/var/config_repository
$SPLUNK_HOME/var/log/nmon/var/json_repository
```

Example:



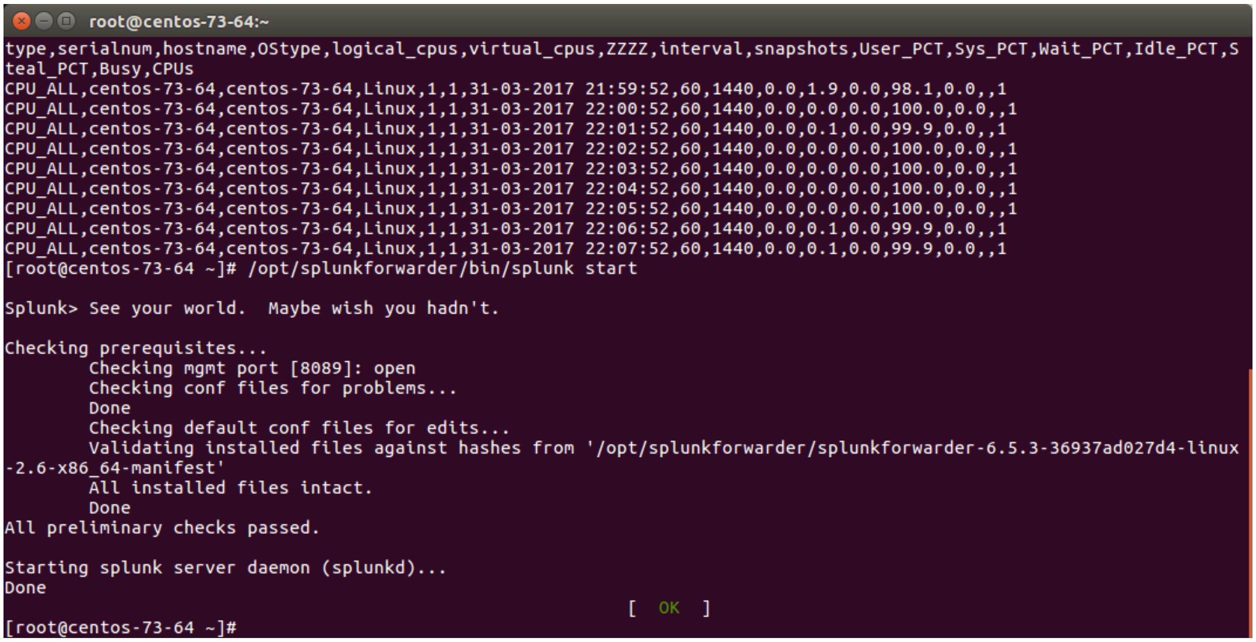
root@centos-73-64:~

```
[root@centos-73-64 ~]# ls -ltr /opt/splunkforwarder/var/log/nmon/var/csv_repository/*CPU_ALL*
-rw-------. 1 root root  425 Mar 31 22:43 /opt/splunkforwarder/var/log/nmon/var/csv_repository/centos-73-64_31_MAR_20
17_203421_CPU_ALL_30815_20170331224358.nmon.csv
-rw-------. 1 root root 4487 Mar 31 22:44 /opt/splunkforwarder/var/log/nmon/var/csv_repository/centos-73-64_31_MAR_20
17_215951_CPU_ALL_80132_20170331224400.nmon.csv
[root@centos-73-64 ~]# head /opt/splunkforwarder/var/log/nmon/var/csv_repository/centos-73-64_31_MAR_2017_215951_CPU_
ALL_80132_20170331224400.nmon.csv
type,serialnum,hostname,OStype,logical_cpus,virtual_cpus,ZZZZ,interval,snapshots,User_PCT,Sys_PCT,Wait_PCT,Idle_PCT,S
teal_PCT,Busy,CPUs
CPU_ALL,centos-73-64,centos-73-64,Linux,1,1,31-03-2017 21:59:52,60,1440,0.0,1.9,0.0,98.1,0.0,,1
CPU_ALL,centos-73-64,centos-73-64,Linux,1,1,31-03-2017 22:00:52,60,1440,0.0,0.0,0.0,100.0,0.0,,1
CPU_ALL,centos-73-64,centos-73-64,Linux,1,1,31-03-2017 22:01:52,60,1440,0.0,0.1,0.0,99.9,0.0,,1
CPU_ALL,centos-73-64,centos-73-64,Linux,1,1,31-03-2017 22:02:52,60,1440,0.0,0.0,0.0,100.0,0.0,,1
CPU_ALL,centos-73-64,centos-73-64,Linux,1,1,31-03-2017 22:03:52,60,1440,0.0,0.0,0.0,100.0,0.0,,1
CPU_ALL,centos-73-64,centos-73-64,Linux,1,1,31-03-2017 22:04:52,60,1440,0.0,0.0,0.0,100.0,0.0,,1
CPU_ALL,centos-73-64,centos-73-64,Linux,1,1,31-03-2017 22:05:52,60,1440,0.0,0.0,0.0,100.0,0.0,,1
CPU_ALL,centos-73-64,centos-73-64,Linux,1,1,31-03-2017 22:06:52,60,1440,0.0,0.1,0.0,99.9,0.0,,1
CPU_ALL,centos-73-64,centos-73-64,Linux,1,1,31-03-2017 22:07:52,60,1440,0.0,0.1,0.0,99.9,0.0,,1
[root@centos-73-64 ~]#
```

Checking Splunk indexing

Splunk monitors those directories in “batch” mode, which means index and delete.

Once you will have restarted Splunk, all the files will be consumed and disappear in a few seconds:



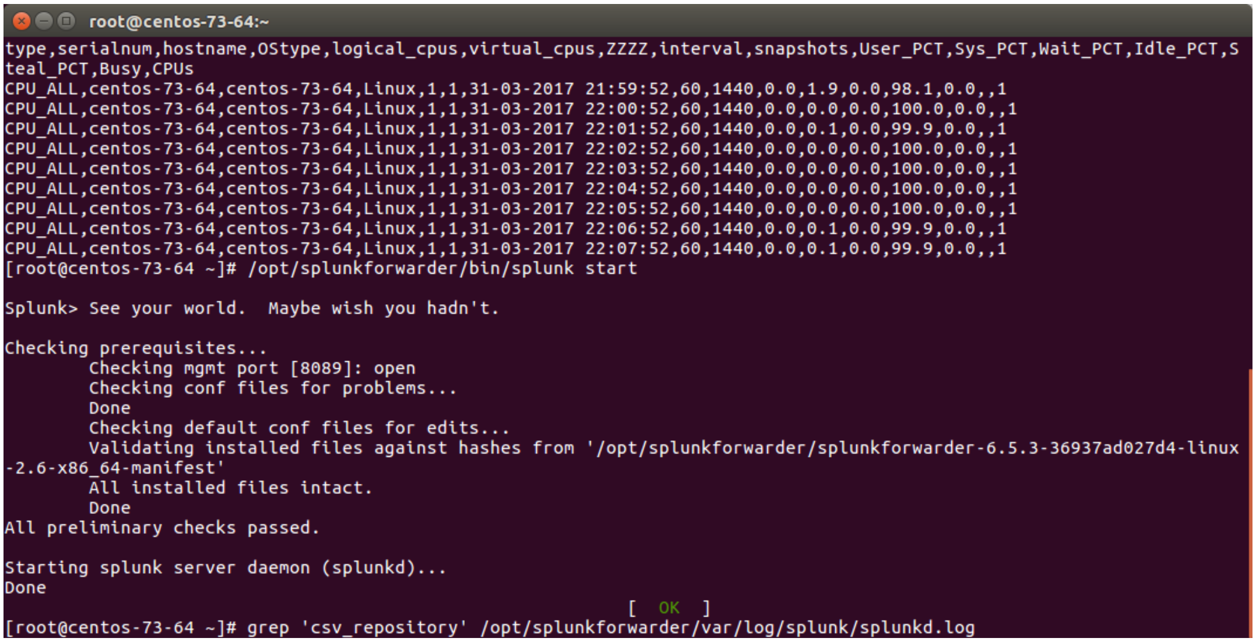
root@centos-73-64:~

```
type,serialnum,hostname,OStype,logical_cpus,virtual_cpus,ZZZZ,interval,snapshots,User_PCT,Sys_PCT,Wait_PCT,Idle_PCT,S
teal_PCT,Busy,CPUs
CPU_ALL,centos-73-64,centos-73-64,Linux,1,1,31-03-2017 21:59:52,60,1440,0.0,1.9,0.0,98.1,0.0,,1
CPU_ALL,centos-73-64,centos-73-64,Linux,1,1,31-03-2017 22:00:52,60,1440,0.0,0.0,0.0,100.0,0.0,,1
CPU_ALL,centos-73-64,centos-73-64,Linux,1,1,31-03-2017 22:01:52,60,1440,0.0,0.1,0.0,99.9,0.0,,1
CPU_ALL,centos-73-64,centos-73-64,Linux,1,1,31-03-2017 22:02:52,60,1440,0.0,0.0,0.0,100.0,0.0,,1
CPU_ALL,centos-73-64,centos-73-64,Linux,1,1,31-03-2017 22:03:52,60,1440,0.0,0.0,0.0,100.0,0.0,,1
CPU_ALL,centos-73-64,centos-73-64,Linux,1,1,31-03-2017 22:04:52,60,1440,0.0,0.0,0.0,100.0,0.0,,1
CPU_ALL,centos-73-64,centos-73-64,Linux,1,1,31-03-2017 22:05:52,60,1440,0.0,0.0,0.0,100.0,0.0,,1
CPU_ALL,centos-73-64,centos-73-64,Linux,1,1,31-03-2017 22:06:52,60,1440,0.0,0.1,0.0,99.9,0.0,,1
CPU_ALL,centos-73-64,centos-73-64,Linux,1,1,31-03-2017 22:07:52,60,1440,0.0,0.1,0.0,99.9,0.0,,1
[root@centos-73-64 ~]# /opt/splunkforwarder/bin/splunk start

Splunk> See your world.  Maybe wish you hadn't.

Checking prerequisites...
        Checking mgmt port [8089]: open
        Checking conf files for problems...
        Done
        Checking default conf files for edits...
        Validating installed files against hashes from '/opt/splunkforwarder/splunkforwarder-6.5.3-36937ad027d4-linux
-2.6-x86_64-manifest'
        All installed files intact.
        Done
All preliminary checks passed.

Starting splunk server daemon (splunkd)...
Done
                                                           [  OK  ]
[root@centos-73-64 ~]#
```



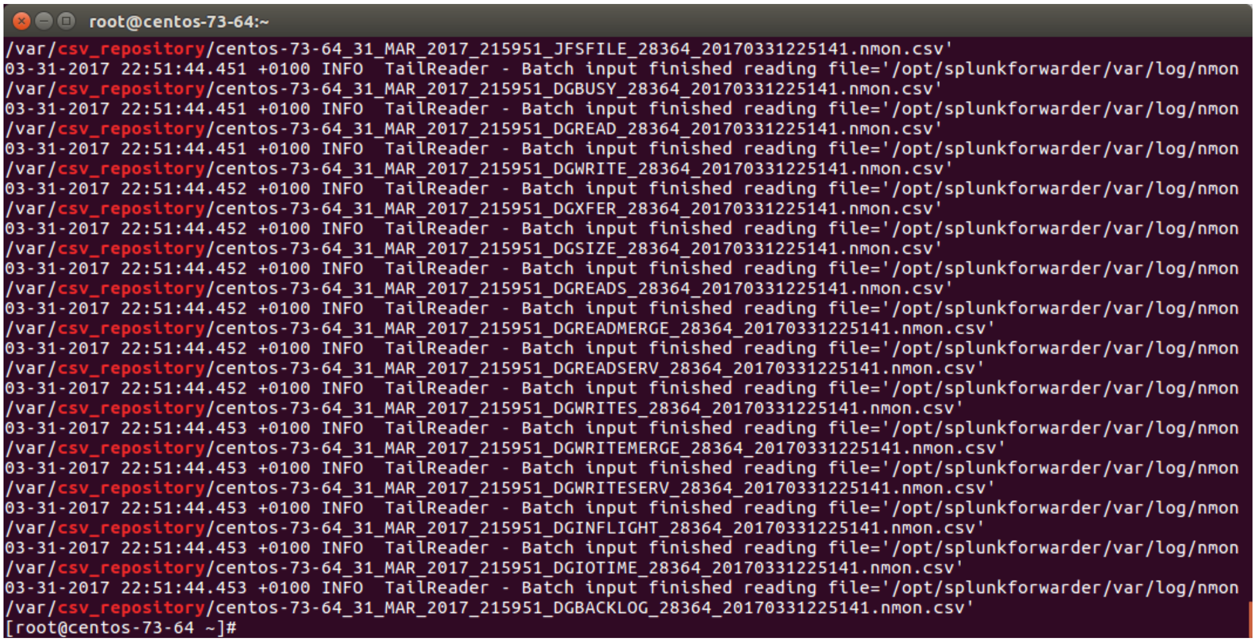
root@centos-73-64:~

```
type,serialnum,hostname,OStype,logical_cpus,virtual_cpus,ZZZZ,interval,snapshots,User_PCT,Sys_PCT,Wait_PCT,Idle_PCT,S
teal_PCT,Busy,CPUs
CPU_ALL,centos-73-64,centos-73-64,Linux,1,1,31-03-2017 21:59:52,60,1440,0.0,1.9,0.0,98.1,0.0,,1
CPU_ALL,centos-73-64,centos-73-64,Linux,1,1,31-03-2017 22:00:52,60,1440,0.0,0.0,0.0,100.0,0.0,,1
CPU_ALL,centos-73-64,centos-73-64,Linux,1,1,31-03-2017 22:01:52,60,1440,0.0,0.1,0.0,99.9,0.0,,1
CPU_ALL,centos-73-64,centos-73-64,Linux,1,1,31-03-2017 22:02:52,60,1440,0.0,0.0,0.0,100.0,0.0,,1
CPU_ALL,centos-73-64,centos-73-64,Linux,1,1,31-03-2017 22:03:52,60,1440,0.0,0.0,0.0,100.0,0.0,,1
CPU_ALL,centos-73-64,centos-73-64,Linux,1,1,31-03-2017 22:04:52,60,1440,0.0,0.0,0.0,100.0,0.0,,1
CPU_ALL,centos-73-64,centos-73-64,Linux,1,1,31-03-2017 22:05:52,60,1440,0.0,0.0,0.0,100.0,0.0,,1
CPU_ALL,centos-73-64,centos-73-64,Linux,1,1,31-03-2017 22:06:52,60,1440,0.0,0.1,0.0,99.9,0.0,,1
CPU_ALL,centos-73-64,centos-73-64,Linux,1,1,31-03-2017 22:07:52,60,1440,0.0,0.1,0.0,99.9,0.0,,1
[root@centos-73-64 ~]# /opt/splunkforwarder/bin/splunk start

Splunk> See your world.  Maybe wish you hadn't.

Checking prerequisites...
        Checking mgmt port [8089]: open
        Checking conf files for problems...
        Done
        Checking default conf files for edits...
        Validating installed files against hashes from '/opt/splunkforwarder/splunkforwarder-6.5.3-36937ad027d4-linux
-2.6-x86_64-manifest'
        All installed files intact.
        Done
All preliminary checks passed.

Starting splunk server daemon (splunkd)...
Done
                                                           [  OK  ]
[root@centos-73-64 ~]# grep 'csv_repository' /opt/splunkforwarder/var/log/splunk/splunkd.log
```



root@centos-73-64:~

```
/var/csv_repository/centos-73-64_31_MAR_2017_215951_JFSFILE_28364_20170331225141.nmon.csv'
03-31-2017 22:51:44.451 +0100 INFO  TailReader - Batch input finished reading file='/opt/splunkforwarder/var/log/nmon
/var/csv_repository/centos-73-64_31_MAR_2017_215951_DGBUSY_28364_20170331225141.nmon.csv'
03-31-2017 22:51:44.451 +0100 INFO  TailReader - Batch input finished reading file='/opt/splunkforwarder/var/log/nmon
/var/csv_repository/centos-73-64_31_MAR_2017_215951_DGREAD_28364_20170331225141.nmon.csv'
03-31-2017 22:51:44.451 +0100 INFO  TailReader - Batch input finished reading file='/opt/splunkforwarder/var/log/nmon
/var/csv_repository/centos-73-64_31_MAR_2017_215951_DGWRITE_28364_20170331225141.nmon.csv'
03-31-2017 22:51:44.452 +0100 INFO  TailReader - Batch input finished reading file='/opt/splunkforwarder/var/log/nmon
/var/csv_repository/centos-73-64_31_MAR_2017_215951_DGXFER_28364_20170331225141.nmon.csv'
03-31-2017 22:51:44.452 +0100 INFO  TailReader - Batch input finished reading file='/opt/splunkforwarder/var/log/nmon
/var/csv_repository/centos-73-64_31_MAR_2017_215951_DGSIZE_28364_20170331225141.nmon.csv'
03-31-2017 22:51:44.452 +0100 INFO  TailReader - Batch input finished reading file='/opt/splunkforwarder/var/log/nmon
/var/csv_repository/centos-73-64_31_MAR_2017_215951_DGREADS_28364_20170331225141.nmon.csv'
03-31-2017 22:51:44.452 +0100 INFO  TailReader - Batch input finished reading file='/opt/splunkforwarder/var/log/nmon
/var/csv_repository/centos-73-64_31_MAR_2017_215951_DGREADMERGE_28364_20170331225141.nmon.csv'
03-31-2017 22:51:44.452 +0100 INFO  TailReader - Batch input finished reading file='/opt/splunkforwarder/var/log/nmon
/var/csv_repository/centos-73-64_31_MAR_2017_215951_DGREADSERV_28364_20170331225141.nmon.csv'
03-31-2017 22:51:44.452 +0100 INFO  TailReader - Batch input finished reading file='/opt/splunkforwarder/var/log/nmon
/var/csv_repository/centos-73-64_31_MAR_2017_215951_DGWRITES_28364_20170331225141.nmon.csv'
03-31-2017 22:51:44.453 +0100 INFO  TailReader - Batch input finished reading file='/opt/splunkforwarder/var/log/nmon
/var/csv_repository/centos-73-64_31_MAR_2017_215951_DGWRITEMERGE_28364_20170331225141.nmon.csv'
03-31-2017 22:51:44.453 +0100 INFO  TailReader - Batch input finished reading file='/opt/splunkforwarder/var/log/nmon
/var/csv_repository/centos-73-64_31_MAR_2017_215951_DGWRITESERV_28364_20170331225141.nmon.csv'
03-31-2017 22:51:44.453 +0100 INFO  TailReader - Batch input finished reading file='/opt/splunkforwarder/var/log/nmon
/var/csv_repository/centos-73-64_31_MAR_2017_215951_DGINFLIGHT_28364_20170331225141.nmon.csv'
03-31-2017 22:51:44.453 +0100 INFO  TailReader - Batch input finished reading file='/opt/splunkforwarder/var/log/nmon
/var/csv_repository/centos-73-64_31_MAR_2017_215951_DGIOTIME_28364_20170331225141.nmon.csv'
03-31-2017 22:51:44.453 +0100 INFO  TailReader - Batch input finished reading file='/opt/splunkforwarder/var/log/nmon
/var/csv_repository/centos-73-64_31_MAR_2017_215951_DGBACKLOG_28364_20170331225141.nmon.csv'
[root@centos-73-64 ~]#
```

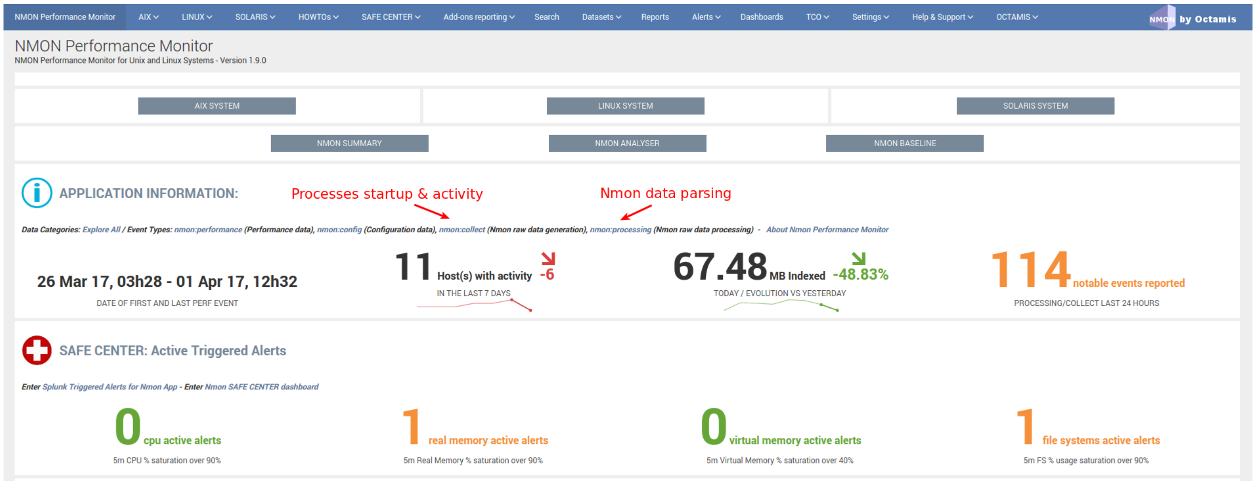
STEP 3: Nmon processing indexing in Splunk

The activity of the TA-nmon "bin/nmon_helper.sh" is logged in Splunk: (startup of fifo_reader and nmon processes)

```
eventtype=nmon:collect
```

The activity of the TA-nmon "bin/fifo_consumer.sh" and nmon2csv parsers is logged in Splunk:

```
eventtype=nmon:processing
```



NMON Performance Monitor

NMON Performance Monitor for Unix and Linux Systems - Version 1.9.0

APPLICATION INFORMATION:

Processes startup & activity

Nmon data parsing

Data Categories: Explore All / Event Types: nmon:performance (Performance data), nmon:config (Configuration data), nmon:collect (Nmon raw data generation), nmon:processing (Nmon raw data processing) - About Nmon Performance Monitor

26 Mar 17, 03h28 - 01 Apr 17, 12h32
DATE OF FIRST AND LAST PERF EVENT

11 Host(s) with activity -6
IN THE LAST 7 DAYS

67.48 MB Indexed -48.83%
TODAY / EVOLUTION VS YESTERDAY

114 notable events reported
PROCESSING/COLLECT LAST 24 HOURS

SAFE CENTER: Active Triggered Alerts

Enter Splunk Triggered Alerts for Nmon App - Enter Nmon SAFE CENTER dashboard

0 cpu active alerts
5m CPU % saturation over 90%

1 real memory active alerts
5m Real Memory % saturation over 90%

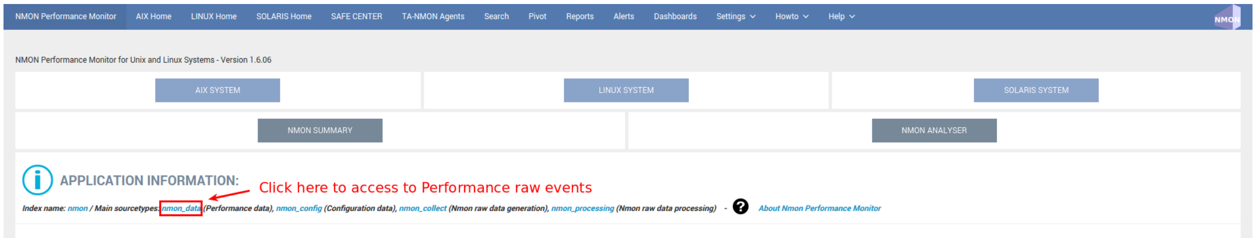
0 virtual memory active alerts
5m Virtual Memory % saturation over 40%

1 file systems active alerts
5m FS % usage saturation over 90%

STEP 4: Verify Nmon Performance and Configuration data

Next step of troubleshooting resides in verifying Performance data and Configuration data in Splunk:

Access to Performance raw events:



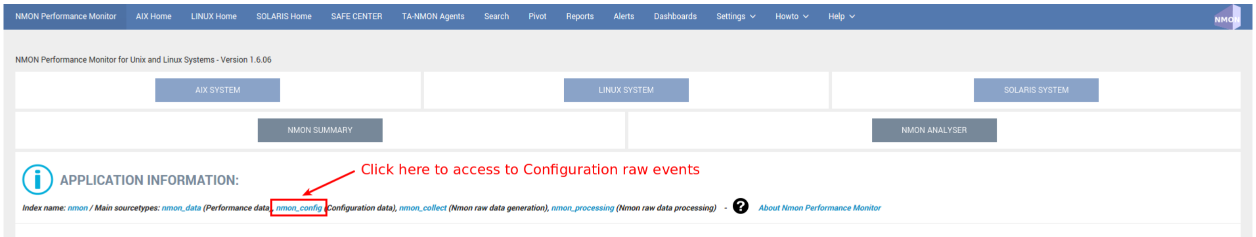
NMON Performance Monitor for Unix and Linux Systems - Version 1.6.06

APPLICATION INFORMATION:

Click here to access to Performance raw events

Index name: nmon / Main sourcetypes: nmon_data (Performance data), nmon_config (Configuration data), nmon_collect (Nmon raw data generation), nmon_processing (Nmon raw data processing) - About Nmon Performance Monitor

Access to Configuration raw events:



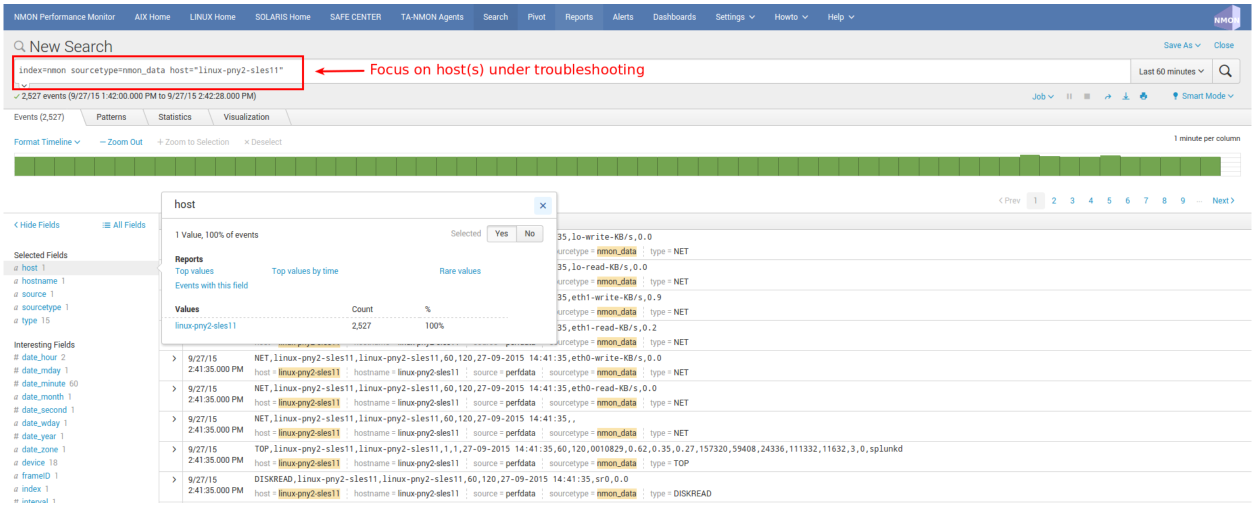
NMON Performance Monitor for Unix and Linux Systems - Version 1.6.06

APPLICATION INFORMATION:

Click here to access to Configuration raw events

Index name: nmon / Main sourcetypes: nmon_data (Performance data), nmon_config (Configuration data), nmon_collect (Nmon raw data generation), nmon_processing (Nmon raw data processing) - About Nmon Performance Monitor

Example of host returning Performance events:



```
index=nmon sourcetype=nmon_data host="linux-pny2-sles11"
```

Focus on host(s) under troubleshooting

✓ 2,527 events (9/27/15 1:42:00.000 PM to 9/27/15 2:42:28.000 PM)

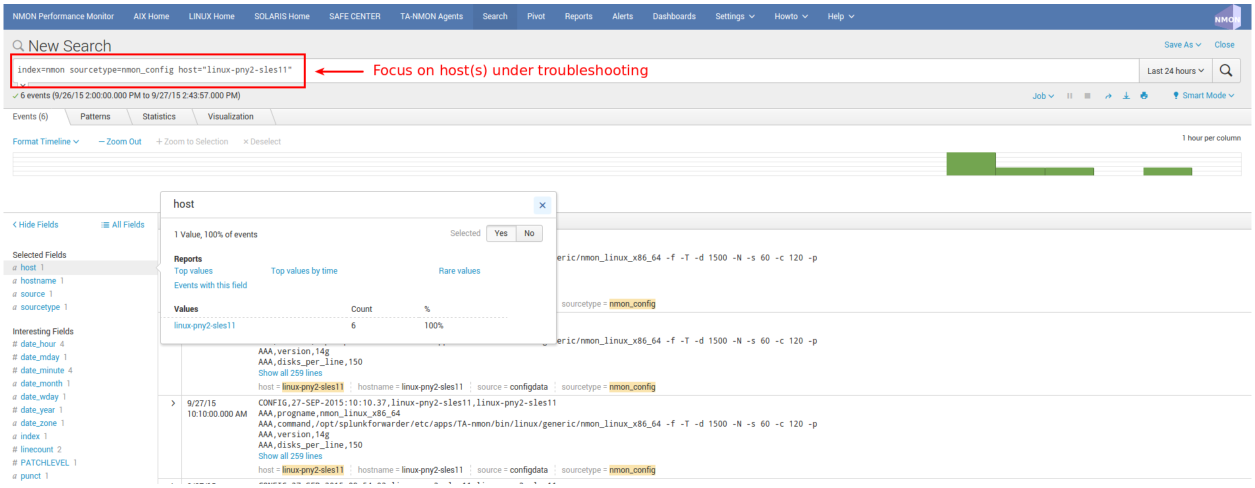
host

1 Value, 100% of events — Selected: Yes No

Values	Count	%
linux-pny2-sles11	2,527	100%

```
9/27/15 2:41:35.000 PM  NET,linux-pny2-sles11,linux-pny2-sles11,60,120,27-09-2015 14:41:35,eth0-write-KB/s,0.0
                        host = linux-pny2-sles11 | hostname = linux-pny2-sles11 | source = perfdata | sourcetype = nmon_data | type = NET
9/27/15 2:41:35.000 PM  NET,linux-pny2-sles11,linux-pny2-sles11,60,120,27-09-2015 14:41:35,eth0-read-KB/s,0.0
                        host = linux-pny2-sles11 | hostname = linux-pny2-sles11 | source = perfdata | sourcetype = nmon_data | type = NET
9/27/15 2:41:35.000 PM  NET,linux-pny2-sles11,linux-pny2-sles11,60,120,27-09-2015 14:41:35,,
                        host = linux-pny2-sles11 | hostname = linux-pny2-sles11 | source = perfdata | sourcetype = nmon_data | type = NET
9/27/15 2:41:35.000 PM  TOP,linux-pny2-sles11,linux-pny2-sles11,1,1,27-09-2015 14:41:35,60,120,0010829,0.62,0.35,0.27,157320,59408,24336,111332,11632,3,0,splunkd
                        host = linux-pny2-sles11 | hostname = linux-pny2-sles11 | source = perfdata | sourcetype = nmon_data | type = TOP
9/27/15 2:41:35.000 PM  DISKREAD,linux-pny2-sles11,linux-pny2-sles11,60,120,27-09-2015 14:41:35,sr0,0.0
                        host = linux-pny2-sles11 | hostname = linux-pny2-sles11 | source = perfdata | sourcetype = nmon_data | type = DISKREAD
```

Example of host returning Configuration events:



```
index=nmon sourcetype=nmon_config host="linux-pny2-sles11"
```

Focus on host(s) under troubleshooting

✓ 6 events (9/26/15 2:00:00.000 PM to 9/27/15 2:43:57.000 PM)

host

1 Value, 100% of events — Selected: Yes No

Values	Count	%
linux-pny2-sles11	6	100%

```
9/27/15 10:10:00.000 AM  CONFIG,27-SEP-2015:10:10.37,linux-pny2-sles11,linux-pny2-sles11
                         AAA,progname,nmon_linux_x86_64
                         AAA,command,/opt/splunkforwarder/etc/apps/TA-nmon/bin/linux/generic/nmon_linux_x86_64 -f -T -d 1500 -N -s 60 -c 120 -p
                         AAA,version,14g
                         AAA,disks_per_line,150
                         Show all 259 lines
                         host = linux-pny2-sles11 | hostname = linux-pny2-sles11 | source = configdata | sourcetype = nmon_config
```

INFORMATION:

You will notice the existence of "host" and "hostname" fields, they are totally equivalent, "host" field is a default Splunk field (Metadata) and "hostname" is directly extracted from Nmon data for Performance and Configuration. The "host" default field is overridden during indexing time to match Nmon data, this allows between other managing history nmon data transparently.

If you are fine with the step, you will have validated that incoming Performance and Configuration events are correctly indexed by Splunk

Since the release V1.6.15, the OStype is generated directly in the raw data, before it was associated with the nmon_inventory lookup table. It is not necessary anymore to verify the lookup table as it cannot be anymore a root cause of error for data identification

02 - Debugging nmon2csv parsers

nmon2csv Python / Perl converters operations can be debugged by manually running the conversion process:

If Splunk is running, stop Splunk:

```
$ /opt/splunkforwarder/bin/splunk stop
```

Have an nmon file ready to test, if you don’t have some to get the current copy in $SPLUNK_HOME/etc/apps/nmon/var/nmon_resposity when the Application is running

Use the shell wrapper to let him decide which converter will be used:

```
$ cat my_file.nmon | /opt/splunkforwarder/etc/apps/TA-nmon/bin/nmon2csv.sh
```

For Python version:

```
$ cat my_file.nmon | /opt/splunkforwarder/etc/apps/TA-nmon/bin/nmon2csv.py
```

For Perl version:

```
$ cat my_file.nmon | /opt/splunkforwarder/etc/apps/TA-nmon/bin/nmon2csv.pl
```

The converter will output its processing steps and generate various csv files in csv_repository and config_repository

Note that you can achieve the same operation in the proper normal Splunk directory, but if you do so, you need to stop Splunk before as it would immediately index and delete csv files

Additional Options

Some options can be used for testing purposes:

```
-debug
```

This option will show various debugging information like removal of events when running in real time mode.

```
-mode [ colddata | realtime ]
```

This option will force the converter to use the colddata mode (the file is entirely proceeded without trying any operation to identify already proceeded data) or real time mode.

real mode is much more complex because we need to identify already proceeded events over each iteration of processing steps.

The real time option should be used when the purpose is simulating the same operation that would do Splunk managing live Nmon data

03 - Troubleshooting FAQ

Problem: I have deployed the TA-nmon add-on to my hosts and i do not seem to receive data

Cause:

root causes can be multiple:

- Universal Forwarder (or full instance) not sending data at all
- Nmon binary does not start
- Nmon raw data converter failure
- input scripts not activated
- Universal Forwarder not compatible (see requirements)
- Clients sending data directly to indexers lacking the PA-nmon add-on

Resolution:

Please read and execute the trouble shooting guide procedure: *01 - Troubleshooting guide from A to Z*

Problem: Linux hosts are not identified as Linux Operating Systems

Cause:

Linux configuration can be split at indexing time, this requires indexing time parsing operation that will fail if the the PA-nmon is not installed in indexers (or if the TA-nmon is not installed on intermediate Heavy Forwarders acting as Collectors in front of your indexers)

Resolution:

- Install the PA-nmon add-on on indexers (as it is required in installation manual) or the TA-nmon add-on if your are using Heavy Forwarders as collectors in front of your indexers
- Update the nmon_inventory lookup by running the generation report (see here)

Problem: I have set frameID Mapping (see here) but past indexed data still have the original frameID value

Cause:

Data Acceleration will keep the previously known values for frameID as long as they won't be rebuilt

Resolution:

Enter the data model manager: pivot > Manage For each data model, click on Rebuild

2.18 Total cost of Ownership

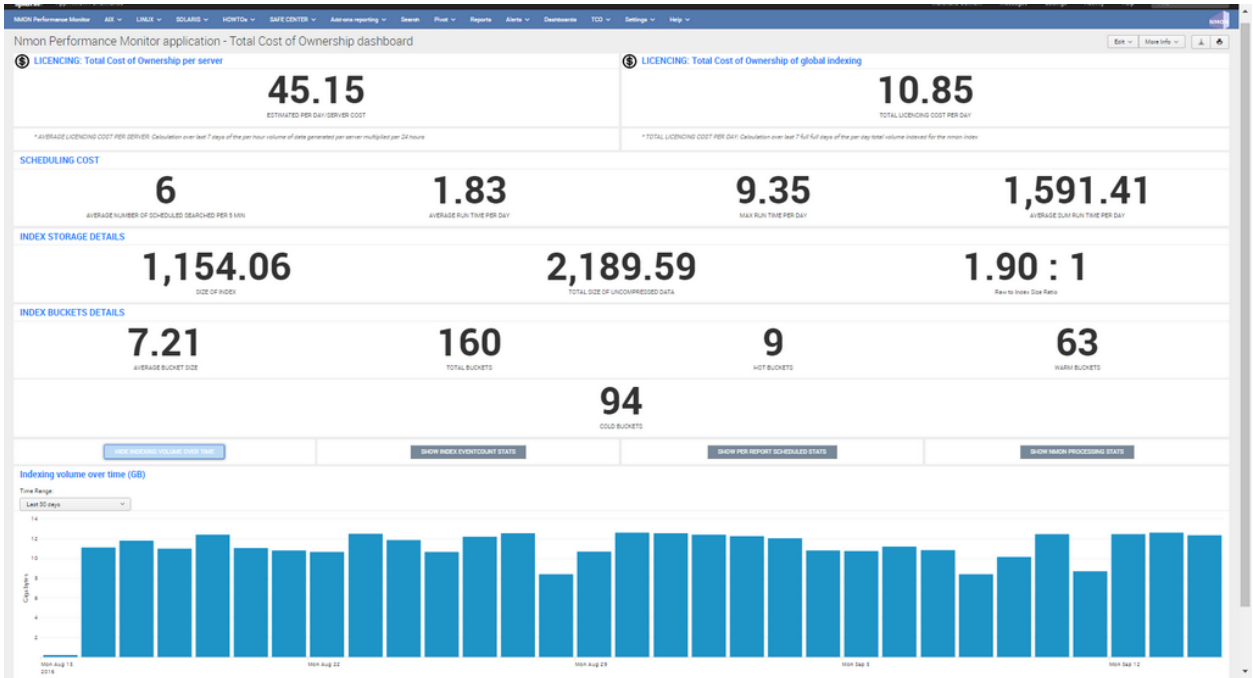
Nmon Performance monitor application is a global solution for Splunk, as such this is a rich and complete application that implies some costs you will want to control and monitor.

The following documentation will help you understanding and managing each part of the application.

This great piece of software is an open source application, as such nothing will ever be hidden and you will always have the complete control.

2.18.1 1. Total Cost of Ownership dashboard

The application provides a builtin dashboard "Total Cost Of Ownership dashboard" to help you understanding better what the real costs can be:



Nmon Performance Monitor application - Total Cost of Ownership dashboard

LICENCING: Total Cost of Ownership per server

45.15
ESTIMATED PER DAY/SERVER COST

* AVERAGE LICENCING COST PER SERVER: Calculation over last 7 days of the per hour volume of data generated per server multiplied per 24 hours

LICENCING: Total Cost of Ownership of global indexing

10.85
TOTAL LICENCING COST PER DAY

* TOTAL LICENCING COST PER DAY: Calculation over last 7 full full days of the per day total volume indexed for the nmon index

SCHEDULING COST

6	1.83	9.35	1,591.41
AVERAGE NUMBER OF SCHEDULED SEARCHED PER 5 MIN	AVERAGE RUN TIME PER DAY	MAX RUN TIME PER DAY	AVERAGE SUM RUN TIME PER DAY

INDEX STORAGE DETAILS

1,154.06	2,189.59	1.90 : 1
SIZE OF INDEX	TOTAL SIZE OF UNCOMPRESSED DATA	Raw to Index Size Ratio

INDEX BUCKETS DETAILS

7.21	160	9	63	94
AVERAGE BUCKET SIZE	TOTAL BUCKETS	HOT BUCKETS	WARM BUCKETS	COLD BUCKETS

Indexing volume over time (GB)

As builtin reporting, this dashboard will expose:

- The average cost of Splunk licence per server / per day in Megabytes
- The average cost of Splunk licence of the global deployment per day in Megabytes
- Various metrics about the cost of scheduled reports associated with the application (specially run time metrics of reports)
- Index storage details (storage size, repartition of buckets, compression data rate…)
- Indexing volume in Megabytes over time
- Fist and last events per data sources
- Detailed information of scheduled reports
- Detailed technical metrics of nmon files processing tasks

2.18.2 2. Managing storage costs

Following items will influence costs related to data storage:

- Data retention
- Acceleration period of data models: the period of time Splunk will use to accelerate application data models

Data retention:

By default when you create an index without any custo; configuration, Splunk will keep all events for 5 years, you can off course set custom retention for your data and decide how the data will be managed and removed.

For more information, read official related documentation: http://docs.splunk.com/Documentation/Splunk/latest/Indexer/Setaretirementandarchivingpolicy

Data model acceleration:

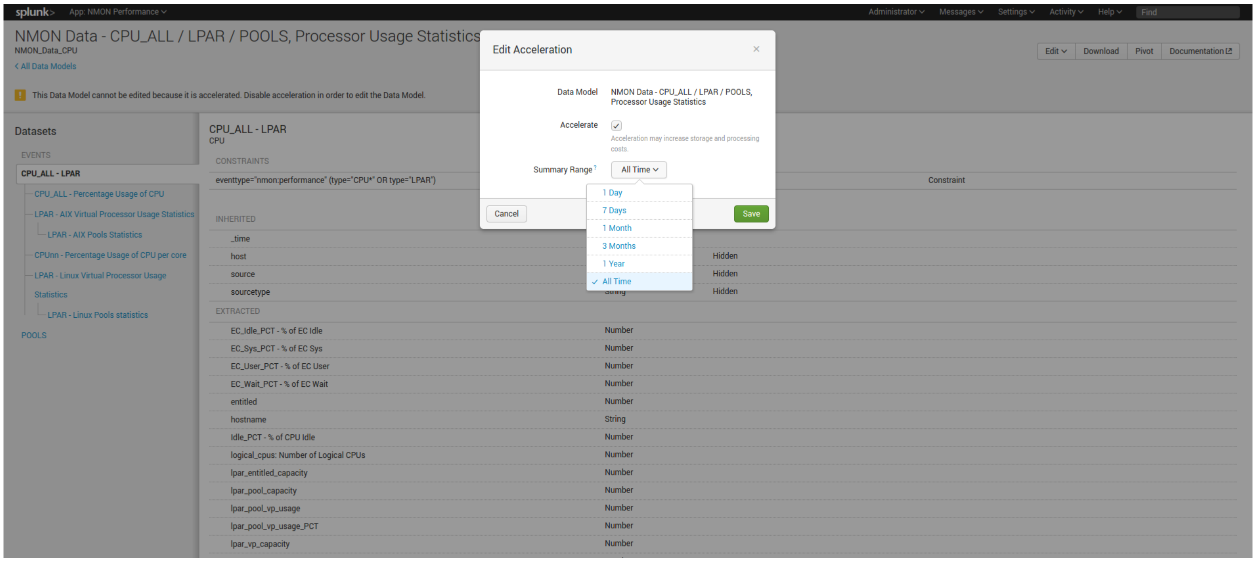
In the default configuration of the Nmon Performance application, every data models will be accelerated over all time.

With large set of data (large number of servers and/or large set of historical data), the accelerated data storage can become important, hopefully you can easily configure you own period of acceleration for each data model.

When you will analyse period of time out the accelerated period, interfaces will continue to work but at the price of much lower performances. (raw searches will not be affected at any time)

Configuring custom accelerated periods will be recommended depending on your configuration, needs and requirements.

For more information, read official related documentation: http://docs.splunk.com/Documentation/Splunk/latest/Knowledge/Acceleratedatamodels



NMON Data - CPU_ALL / LPAR / POOLS, Processor Usage Statistics
NMON_Data_CPU

This Data Model cannot be edited because it is accelerated. Disable acceleration in order to edit the Data Model.

Edit Acceleration

Data Model: NMON Data - CPU_ALL / LPAR / POOLS, Processor Usage Statistics

Accelerate: ☑
Acceleration may increase storage and processing costs.

Summary Range: All Time

Datasets

EVENTS
- CPU_ALL - LPAR
  - CPU_ALL - Percentage Usage of CPU
  - LPAR - AIX Virtual Processor Usage Statistics
    - LPAR - AIX Pools Statistics
  - CPUnn - Percentage Usage of CPU per core
  - LPAR - Linux Virtual Processor Usage Statistics
    - LPAR - Linux Pools statistics
- POOLS

CPU_ALL - LPAR
CPU

CONSTRAINTS		Constraint
eventtype="nmon:performance" (type="CPU*" OR type="LPAR")		
INHERITED		
_time		
host		Hidden
source		Hidden
sourcetype	String	Hidden
EXTRACTED		
EC_Idle_PCT - % of EC Idle	Number	
EC_Sys_PCT - % of EC Sys	Number	
EC_User_PCT - % of EC User	Number	
EC_Wait_PCT - % of EC Wait	Number	
entitled	Number	
hostname	String	
Idle_PCT - % of CPU Idle	Number	
logical_cpus: Number of Logical CPUs	Number	
lpar_entitled_capacity	Number	
lpar_pool_capacity	Number	
lpar_pool_vp_usage	Number	
lpar_pool_vp_usage_PCT	Number	
lpar_vp_capacity	Number	

2.18.3 3. Managing licencing costs

Each server that will be reporting performance and configuration generates data to be indexed in Splunk, which implies a level of utilization of your Splunk core licence.

The way the Nmon Performance application generates is specially performing and optimized from both performance and licensing perspective.

Most of the data generated within the application uses a comma separated format which implies an high level of performance, and the lowest level of data to be generated. (there are no redundant fields name definitions wich slightly reduce the volume of data to be generated)

However, it is also very easy to influence the volume of data to be generated by machine using builtin custom nmon.conf.

Using the function allows increasing the time in minute between performance measures, by default this value is set to 1 minute can be set to whatever you like.

For more information see: Manage the Volume of data generated by the Nmon Data collection

Basically, the volume of data per server can be estimated between 15 and 50 MB per day, but this may slightly vary depending on the system. (number of CPU, disks…)

2.18.4 4. Splunk core resources usage

Following items may influence how the application may impact your Splunk infrastructure: (CPU, memory, disks IOPS…)

- Number of servers to be managed
- Acceleration period of data models / size of indexes
- scheduling of reports (alerting reports, inventory data generation. . .)

By default, every piece of the application has been designed to be as optimized as possible, and will strictly respect best practices and the highest level of code quality.

Most of all and this is a global requirement, your Splunk deployment must be correctly designed and sized. If you have poor performances due to undersized servers or unadapted configurations (non distributed configuration, overcharged servers. . .), this is clearly where you need to start investigating.

How deploying Nmon performance can influence resources utilization of your Splunk deployment:

You can easily manage multiple thousands of servers from the same Splunk deployment, but obviously the more servers you will manage, the more you will:

- require storage capacity
- impact your Splunk licence
- require storage and physical memory on your search heads for the baseline KVstores
- require CPU and memory for data model acceleration building phase and maintenance operations (specially after indexers restart)

As such, you can control:

- The activation / deactivation and scheduling configuration of alerting reports (See Alerts within the application)
- The activation / deactivation and scheduling configuration of baseline KVstore (See Generate NMON baseline * reports within the application)
- The scheduling configuration of the nmon innventory KVstore generation (See Generate NMON Inventory Lookup Table, by default runs every hour)
- The volume of data to be generated per server (see section 3 of this document)
- Configure your data retirement policy (see section 2 of this document)
- Configure custom values for the acceleration period of the application data models (see section 2 of this document)

2.19 Large scale deployment considerations

If you are planing to deploy Nmon in a large scale scenario for thousands of servers, please read carefully the following documentation.

Nmon for Splunk Enterprise can easily be deployed to thousands and thousands of nodes, however there are things that should be considered to optimize at the best your Splunk and Nmon deployment.

The following items will help you controlling and optimizing your Nmon deployment at large scale.

Please also review the *Total cost of Ownership* documentation.

2.19.1 Data model acceleration

Accelerated data models are massively used in the application, this implementation provides exceptional performances of searches, and a great user experience. However, data models have a cost in term of storage and system resources utilization for acceleration build and maintenance.

Splunk certification requirements prohibit the default activation of data models acceleration.

Since version 1.9.12, none of the data models are accelerated by default, this is your responsibility to decide if you wish to do so, bellow are the recommended acceleration parameters:

- metrics related data models accelerated over a period of 1 year
- non metrics data models accelerated over the last 30 days (Nmon config, Nmon processing)

nmon/default/datamodels.conf

```
acceleration.earliest_time = -1y
```

Restricting the acceleration period will helping reducing:

- The amount of storage used per data model for the acceleration
- The amount of time required for initial build or total rebuild of the acceleration, as well as the amount of system resources (CPU, memory) that are temporarily required on indexers to build the acceleration

Note that The maintenance cost, which refers to operation that Splunk operates periodically to maintain the state of acceleration, will not necessary be different with a large or a small period.

Also, rolling restart of clustered indexers will generate a partial verification and/or rebuild of data model acceleration, with large set of data this can imply a temporarily high level of resource usage on indexers following the rolling restart.

Finally, take care not to reduce too much the acceleration period, searches out of the acceleration period are still possible but at the price of much more poor performances.

Restricting the acceleration period of data models:

please refer to Splunk documentation: https://docs.splunk.com/Documentation/Splunk/latest/Admin/Datamodelsconf

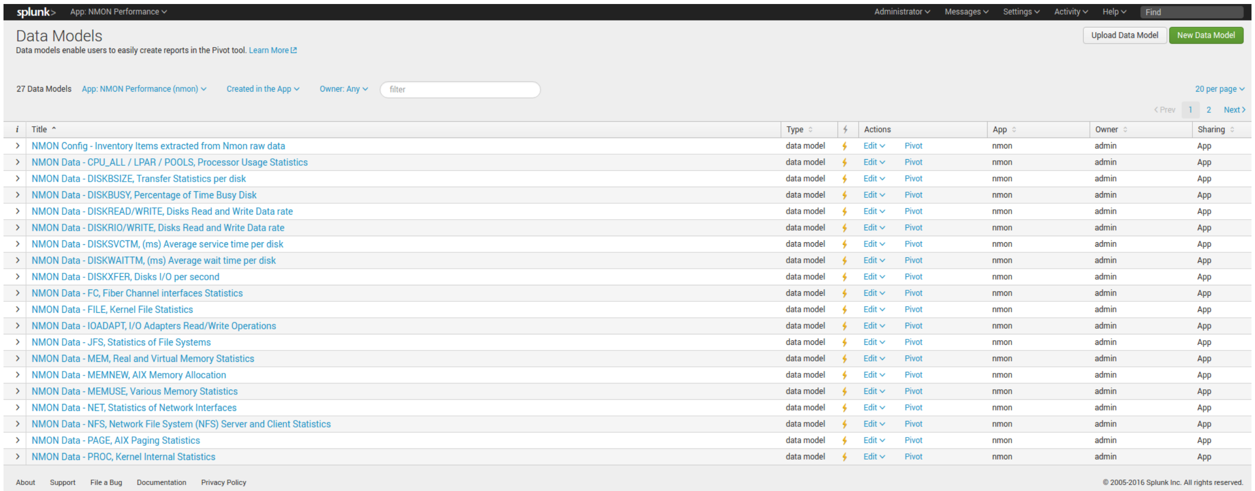
You can easily customize the acceleration period by creating a local copy of the datamodels.conf under the “local” directory.

Then, for each data model, ensure to set the required period, example with a 3 months period:

```
acceleration.earliest_time = -3mon
```

Acceleration setting of data models can also be managed directly in Splunk Web:

Settings / Data models:



Data Models

Data models enable users to easily create reports in the Pivot tool. Learn More

27 Data Models App: NMON Performance (nmon) Created in the App Owner: Any filter

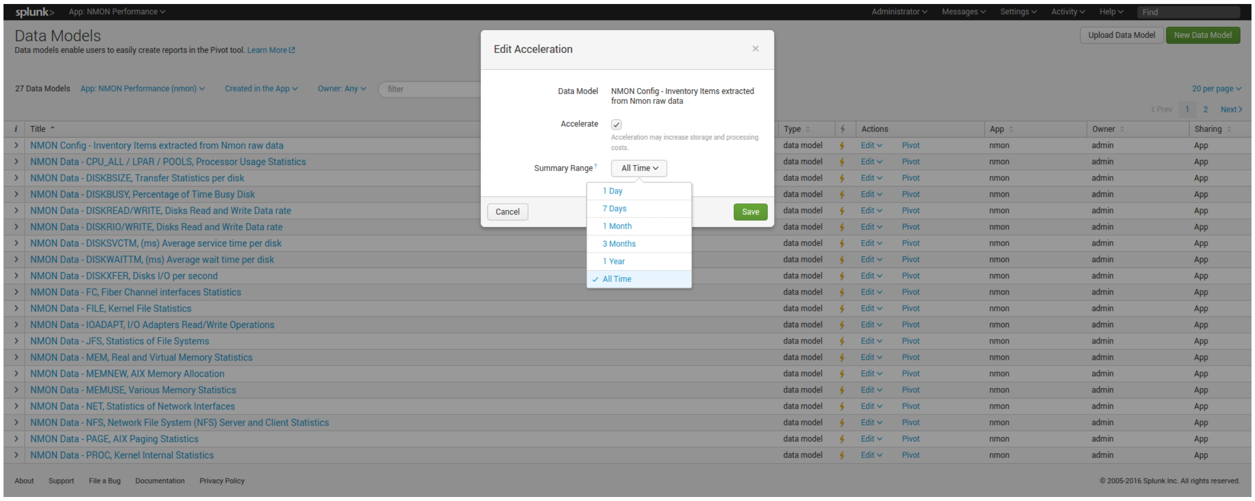
20 per page

‹ Prev 1 2 Next ›

i	Title ^	Type	⚡	Actions		App	Owner	Sharing
›	NMON Config - Inventory Items extracted from Nmon raw data	data model	⚡	Edit	Pivot	nmon	admin	App
›	NMON Data - CPU_ALL / LPAR / POOLS, Processor Usage Statistics	data model	⚡	Edit	Pivot	nmon	admin	App
›	NMON Data - DISKBSIZE, Transfer Statistics per disk	data model	⚡	Edit	Pivot	nmon	admin	App
›	NMON Data - DISKBUSY, Percentage of Time Busy Disk	data model	⚡	Edit	Pivot	nmon	admin	App
›	NMON Data - DISKREAD/WRITE, Disks Read and Write Data rate	data model	⚡	Edit	Pivot	nmon	admin	App
›	NMON Data - DISKRIO/WRITE, Disks Read and Write Data rate	data model	⚡	Edit	Pivot	nmon	admin	App
›	NMON Data - DISKSVCTM, (ms) Average service time per disk	data model	⚡	Edit	Pivot	nmon	admin	App
›	NMON Data - DISKWAITTM, (ms) Average wait time per disk	data model	⚡	Edit	Pivot	nmon	admin	App
›	NMON Data - DISKXFER, Disks I/O per second	data model	⚡	Edit	Pivot	nmon	admin	App
›	NMON Data - FC, Fiber Channel interfaces Statistics	data model	⚡	Edit	Pivot	nmon	admin	App
›	NMON Data - FILE, Kernel File Statistics	data model	⚡	Edit	Pivot	nmon	admin	App
›	NMON Data - IOADAPT, I/O Adapters Read/Write Operations	data model	⚡	Edit	Pivot	nmon	admin	App
›	NMON Data - JFS, Statistics of File Systems	data model	⚡	Edit	Pivot	nmon	admin	App
›	NMON Data - MEM, Real and Virtual Memory Statistics	data model	⚡	Edit	Pivot	nmon	admin	App
›	NMON Data - MEMNEW, AIX Memory Allocation	data model	⚡	Edit	Pivot	nmon	admin	App
›	NMON Data - MEMUSE, Various Memory Statistics	data model	⚡	Edit	Pivot	nmon	admin	App
›	NMON Data - NET, Statistics of Network Interfaces	data model	⚡	Edit	Pivot	nmon	admin	App
›	NMON Data - NFS, Network File System (NFS) Server and Client Statistics	data model	⚡	Edit	Pivot	nmon	admin	App
›	NMON Data - PAGE, AIX Paging Statistics	data model	⚡	Edit	Pivot	nmon	admin	App
›	NMON Data - PROC, Kernel Internal Statistics	data model	⚡	Edit	Pivot	nmon	admin	App

About Support File a Bug Documentation Privacy Policy

© 2005-2016 Splunk Inc. All rights reserved.



Data Models

Data models enable users to easily create reports in the Pivot tool. Learn More

27 Data Models App: NMON Performance (nmon) Created in the App Owner: Any filter

Edit Acceleration

Data Model: NMON Config - Inventory Items extracted from Nmon raw data

Accelerate: ☑

Acceleration may increase storage and processing costs.

Summary Range: All Time

- 1 Day
- 7 Days
- 1 Month
- 3 Months
- 1 Year
- ✓ All Time

i	Title	Type	⚡	Actions		App	Owner	Sharing
>	NMON Config - Inventory Items extracted from Nmon raw data	data model	⚡	Edit	Pivot	nmon	admin	App
>	NMON Data - CPU_ALL / LPAR / POOLS, Processor Usage Statistics	data model	⚡	Edit	Pivot	nmon	admin	App
>	NMON Data - DISKBSIZE, Transfer Statistics per disk	data model	⚡	Edit	Pivot	nmon	admin	App
>	NMON Data - DISKBUSY, Percentage of Time Busy Disk	data model	⚡	Edit	Pivot	nmon	admin	App
>	NMON Data - DISKREAD/WRITE, Disks Read and Write Data rate	data model	⚡	Edit	Pivot	nmon	admin	App
>	NMON Data - DISKRIO/WRITE, Disks Read and Write Data rate	data model	⚡	Edit	Pivot	nmon	admin	App
>	NMON Data - DISKSVCTM, (ms) Average service time per disk	data model	⚡	Edit	Pivot	nmon	admin	App
>	NMON Data - DISKWAITTM, (ms) Average wait time per disk	data model	⚡	Edit	Pivot	nmon	admin	App
>	NMON Data - DISKXFER, Disks I/O per second	data model	⚡	Edit	Pivot	nmon	admin	App
>	NMON Data - FC, Fiber Channel interfaces Statistics	data model	⚡	Edit	Pivot	nmon	admin	App
>	NMON Data - FILE, Kernel File Statistics	data model	⚡	Edit	Pivot	nmon	admin	App
>	NMON Data - IOADAPT, I/O Adapters Read/Write Operations	data model	⚡	Edit	Pivot	nmon	admin	App
>	NMON Data - JFS, Statistics of File Systems	data model	⚡	Edit	Pivot	nmon	admin	App
>	NMON Data - MEM, Real and Virtual Memory Statistics	data model	⚡	Edit	Pivot	nmon	admin	App
>	NMON Data - MEMNEW, AIX Memory Allocation	data model	⚡	Edit	Pivot	nmon	admin	App
>	NMON Data - MEMUSE, Various Memory Statistics	data model	⚡	Edit	Pivot	nmon	admin	App
>	NMON Data - NET, Statistics of Network Interfaces	data model	⚡	Edit	Pivot	nmon	admin	App
>	NMON Data - NFS, Network File System (NFS) Server and Client Statistics	data model	⚡	Edit	Pivot	nmon	admin	App
>	NMON Data - PAGE, AIX Paging Statistics	data model	⚡	Edit	Pivot	nmon	admin	App
>	NMON Data - PROC, Kernel Internal Statistics	data model	⚡	Edit	Pivot	nmon	admin	App

2.19.2 Indexes settings, retention and rolling buckets

Hot DB bucket size for large volume:

If you are indexing 10GB or more per day, then you should set the maxDataSize, according to Splunk spec: https://docs.splunk.com/Documentation/Splunk/latest/Admin/Indexesconf

```
maxDataSize = auto_high_volume
```

This settings can take place in a local/indexes.conf configuration file of the PA-nmon, or the indexes.conf if you arenot using the PA-nmon

Retention:

Ensure you set the retention of the nmon index according to your needs, See: http://docs.splunk.com/Documentation/Splunk/latest/Indexer/Setaretirementandarchivingpolicy

Rolling buckets and buckets management:

Ensure you set the better configuration possible according to your environment, such as using faster disks for hot and warm buckets.

For more information, See: https://docs.splunk.com/Documentation/Splunk/latest/Indexer/HowSplunkstoresindexes

2.19.3 Alerting customization

By default, the Nmon Performance application has several alerting reports configured:

- NMON - File System % usage exceeds 90% (5 consecutive minutes minimal duration)
- NMON - Real Memory % usage exceeds 90% (5 consecutive minutes minimal duration)
- NMON - Virtual Memory % usage exceeds 40% (5 consecutive minutes minimal duration)
- NMON - IBM PSERIES Pools CPU % usage exceeds 90% (5 consecutive minutes minimal duration)
- NMON - CPU % usage exceeds 90% (5 consecutive minutes minimal duration)
- NMON Collect - duplicated nmon instances may occur (excessive nbr of process launched)

These reports will run every five minutes. Excepting the “NMON Collect”, they all use the same variation of macros, by default these alerting reports will scan for all hosts.

For instance the CPU alert has the following definition:

```
`alerting_cpu_usage(*,*,90,300,5m)`
```

Which stands for the macro definition:

```
[alerting_cpu_usage(5)]
args = frameID,hostname,alert_usage,min_duration,max_pause
```

As exposed, these alerts will scan for every host available, you may want to restrict them to a given list of hosts, such as your production servers only, and so on.

You can restrict the scope of the search using wildcard characters (*), such as restricting frameIDs or hostnames, you can even create your own macros based on the provided models if you need more complex restrictions. (such as using booleans)

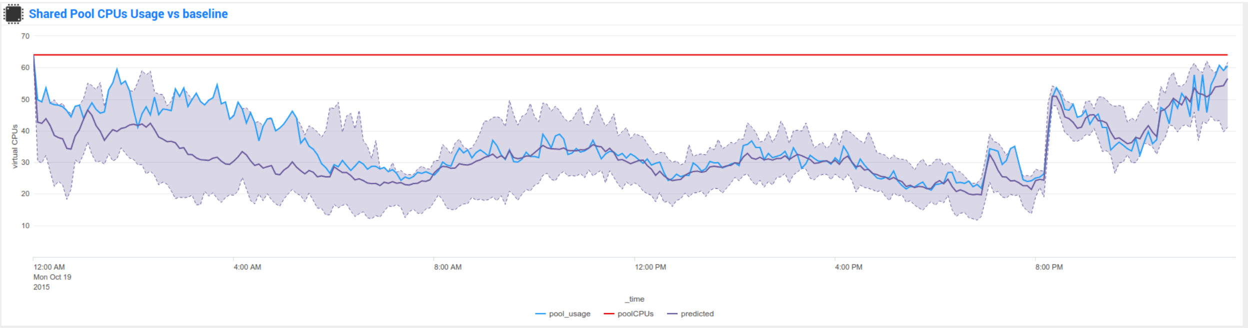
Note: If you are not using IBM frames, you san safely disable the schedule of the "NMON - IBM PSERIES Pools CPU % usage exceeds 90% (5 consecutive minutes minimal duration)"

Each customization must be achieved through Splunk Web, or stored in local version of configuration files to be upgrade resilient

2.19.4 Baseline KVstore management

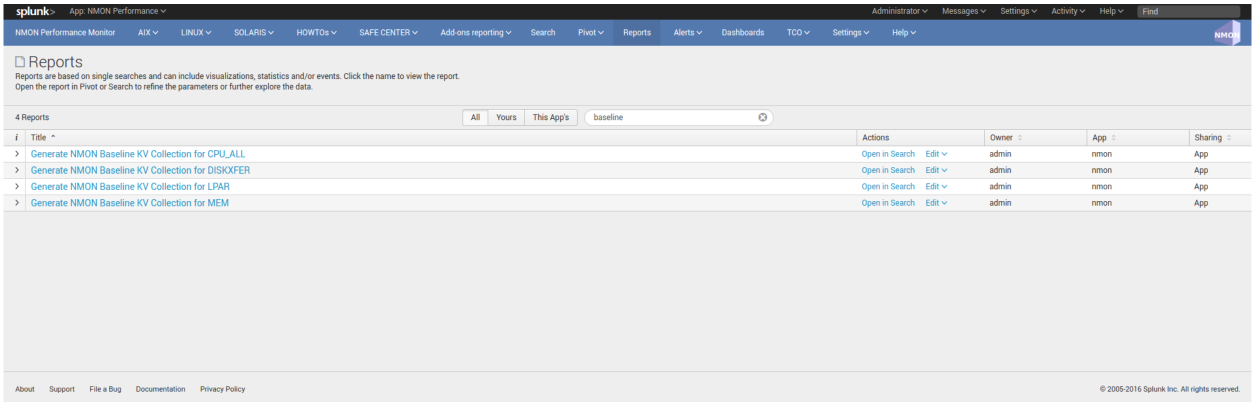
Nmon Performance for Splunk implements different KVstore known as the "baseline KVstores", and used within the baseline interfaces.

These KVstore are being filled by scheduled reports and provide advanced analysis of usual system resources usage to help you determining anomalies on your systems.



By default, the KVstores will contain data for all of the available servers within your deployment, in a large scale deployment you might want to limit these features to important servers, such as production servers only.

The following reports are being used to generate KVstore data once a week:



splunk> App: NMON Performance
Administrator Messages Settings Activity Help Find
NMON Performance Monitor AIX LINUX SOLARIS HOWTOs SAFE CENTER Add-ons reporting Search Pivot Reports Alerts Dashboards TCO Settings Help
NMON
All Yours This App's
About Support File a Bug Documentation Privacy Policy
© 2005-2016 Splunk Inc. All rights reserved.

Reports

Reports are based on single searches and can include visualizations, statistics and/or events. Click the name to view the report.
Open the report in Pivot or Search to refine the parameters or further explore the data.

4 Reports

baseline

i	Title	Actions		Owner	App	Sharing
>	Generate NMON Baseline KV Collection for CPU_ALL	Open in Search	Edit	admin	nmon	App
>	Generate NMON Baseline KV Collection for DISKXFER	Open in Search	Edit	admin	nmon	App
>	Generate NMON Baseline KV Collection for LPAR	Open in Search	Edit	admin	nmon	App
>	Generate NMON Baseline KV Collection for MEM	Open in Search	Edit	admin	nmon	App

You can optionally customize these reports to filter out servers or focus on particular environment such as production servers only, which will limit the volume of data being stored in these KVstores.

Kvstores are hosted by search heads and do not need to be replicated to your indexers, resources that will be used to host these KVstores:

- Storage: Very large KVstores containing data for thousands of server may require a few GB of storage on your search heads
- Physical memory: As well, KVstores have physical memory costs, very large KVstores can impact your memory utilization on search heads
- Reports runtime: The more server you have, the more time these reports might need to complete, they run by default on Sunday basis, you can manage the scheduling differently according to your own constraints

Open these reports in Splunk Web and modify the root search to limit the scope of the searches, you can also manage the searches in a local version of "savedsearches.conf".

For upgrade resiliency considerations, do not modify the default/savedsearches.conf configuration file.

2.19.5 Managing nmon collection and volume of data

By default, the technical add-ons provided with the Nmon Performance application will generate performance data with a 1 minute accuracy between 2 performances collection.

These features can be easily controlled through an internal process using a customized version of the "nmon.conf" configuration file.

See: *Manage the Volume of data generated by the Nmon Data collection*

The Nmon Performance technical add-ons generates csv flows of data, as such the volume of data to be generated is already really optimised and reduced to the maximum.

However, you can choose to limit licence usage and storing costs by increasing the time between 2 performance collections, a common choice might be to increase this time to 2 or 3 minutes.

2.20 Reference Materials

Please refer to *Scripts and Binaries*