Practical Splunk Administration in 2 Hours

- **Working with Configuration Files and Indexes**
- **Understanding SPLUNK Admin Basics AND License Management**

Telegram:
@Cybersecnerd (username)
https://t.me/+TI9qqxih70VlOTM1 (Splunk-workshops)
(group)

Email:
cybersecnerd@gmail.com

Overview

Splunk Enterprise Administrator responsibilities

Splunk Enterprise Components

- Indexers, search heads, heavy forwarders, oh my!

Licensing options

- Understand how licensing works

Handling license violations

- What to do when a license violation occurs?

A Day in Splunk Enterprise Administrator's Life

Splunk Enterprise Systems Administrators and Data Administrators have different responsibilities

Splunk Enterprise Systems Administrator Tasks

Install Splunk Software

Create and Manage Indexes

Manage Splunk Licenses

Configure Security

Monitor Splunk and respond to Monitoring Console Alerts

Splunk Enterprise Data Administrator Tasks

Create and manage inputs

Manage parsing of data including line-breaking and timestamp extraction

Design and establish new ingestion pipelines

Manage Splunk configuration files

Collaborate with users on data on-boarding

Splunk Enterprise Components

The Splunk Platform

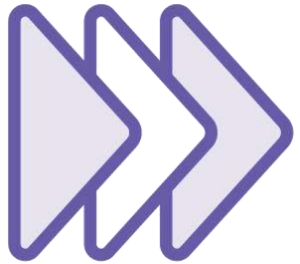
Three Core Components

Indexer

Receive, parse and store machine data in files. Serve search requests

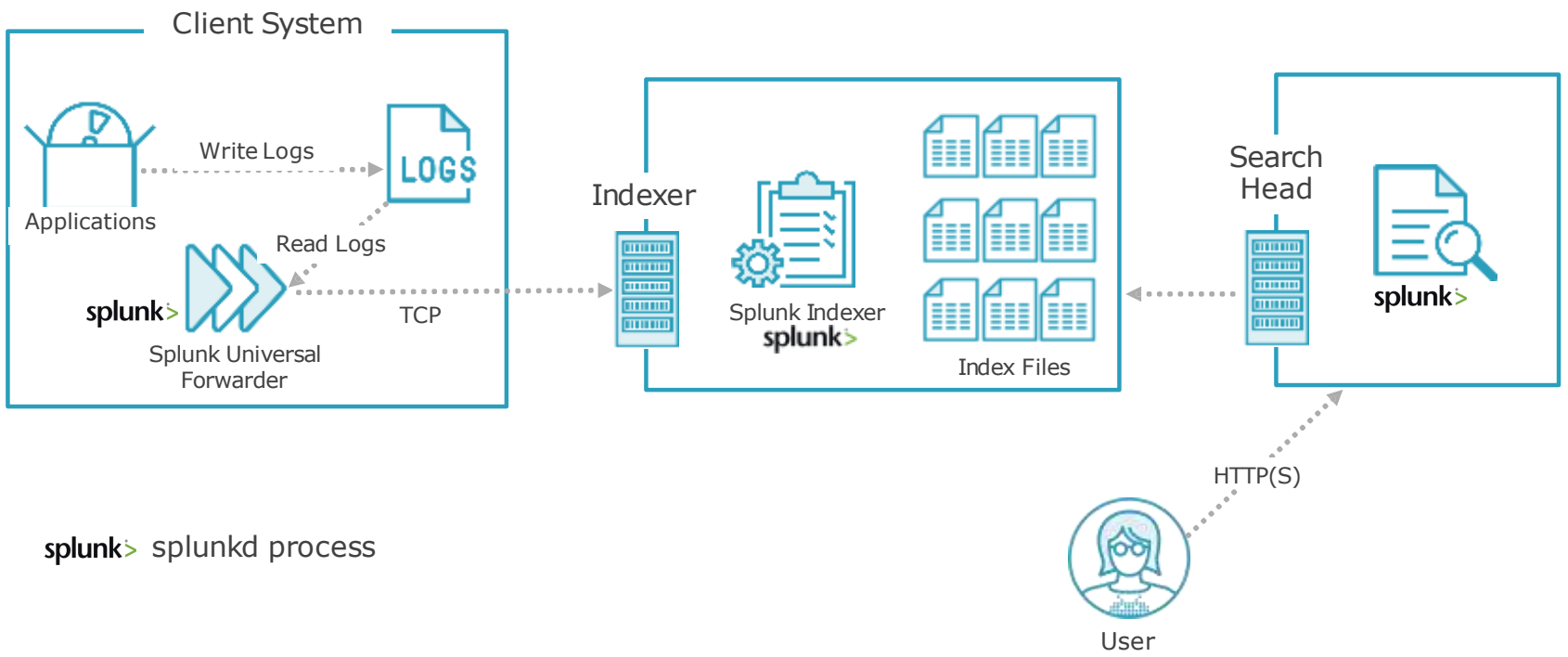
Search Head

Web interface for the users. Dispatch searches to Indexers



Universal Forwarder

Collects data from the clients and forwards for Indexing



splunk> splunkd process

Other Splunk Components

License Master

Deployment Server

Cluster Master

Search Head Deployer

Monitoring Console

Heavy Forwarder

The License Master manages Splunk licenses

Other Splunk Components are License Slaves

Can be a co-located with other components such as Monitoring Console

Licenses can be managed through Splunk Web

Manages the Configuration files on the Deployment Clients

Maintains configuration in serverclass.conf

Alternative solutions such as ansible/puppet can be used

Configuration files are packaged as apps

Deployment clients periodically poll Deployment Server

Cluster Master manages the Indexer Cluster

There is only one Cluster Master

Maintains data bucket status and handles replication

Distributes configuration files and apps to Cluster members

Search Head Deployer distributes apps and configuration files to Search Head Cluster Members

Keeps the files in SPLUNK_HOME/etc/shd-apps

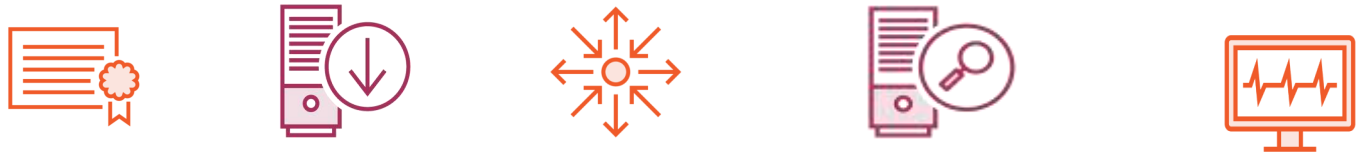
Cannot run on the same instance as a cluster member

Monitoring Console is a Web App that helps to monitor the system health

Rich set of charts and statistics

One-stop-shop to monitor everything

Only Administrators have access



Heavy forwarders can parse data before forwarding to indexer

Full Splunk Enterprise binary with distributed search disabled

Can also index data locally

Smaller footprint compared to indexer

Splunk Licensing Options

Splunk Licensing is based on the amount of data indexed

Types of Splunk Licenses

Enterprise License

Industrial IoT License

Free License

Forwarder License

Trial License

Dev/Test License

The Enterprise License can be bought for any indexing volume

Enables all Splunk features including clustering and distributed search

No-enforcement. Users can still search after a License violation

Licenses can be stacked





The Free license includes 500MB/day indexing, for life

Disabled features

- Clustering
- Authentication
- Distributed Search
- Alerting
- Deployment management





The Trial License provides full Splunk features for 60 days

After 60 days, it automatically becomes Free License

Maximum 500 MB daily volume

Sales Trial license can provided for customized limits

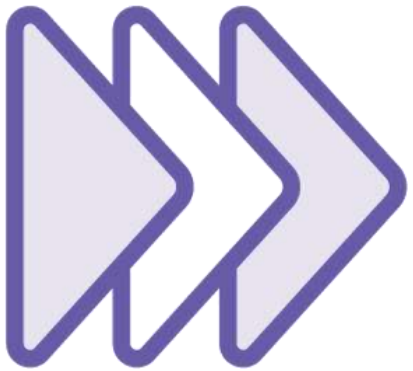




Splunk for Industrial IoT License

Not stackable

Access to Splunk Enterprise and a select premium Splunk Apps



Forwarder License allows forwarding of unlimited data

Cannot be used for indexing

No need to purchase them separately

Universal Forwarders automatically apply Forwarder License

Heavy Forwarders must be converted to Forwarder License group

Dev/Test license for running Splunk in Non Prod environments

Cannot be used in distributed environment

Not stackable

Can be used for Splunk App development

What counts towards daily License quota?

Full size of data flowing through the parsing pipeline

Not the disk storage

Does not include

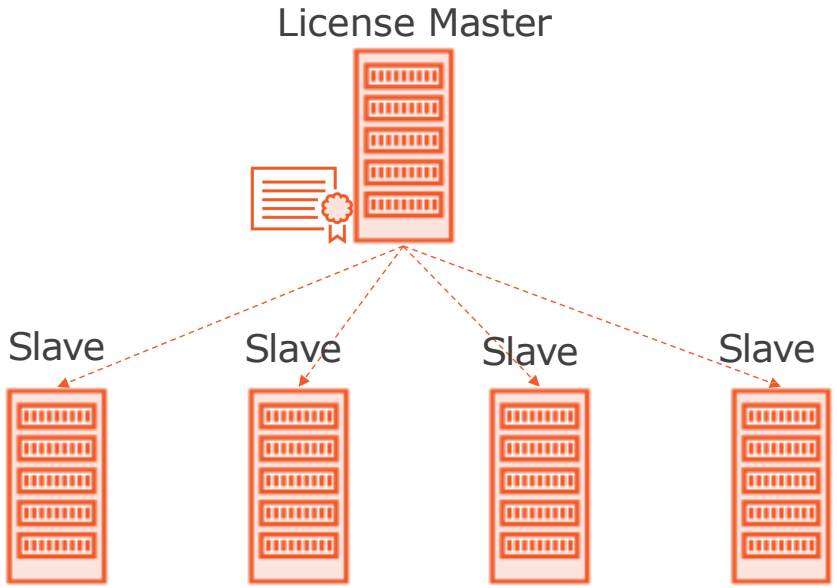
Replicated Data

Summary indexes

Internal Logs

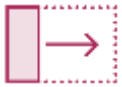
Metadata

Licensing in Distributed Environment



Managing License Violations

License Warnings and Violations



Exceeding daily volume quota results in a Warning

5 or more warnings in a 30 day rolling period is a Violation

Searching is NOT disabled in violation period

Alert logged in Messages on any Splunk Web pages

Monitoring License Warnings

Monitoring Console

Enable the license monitoring alerts

Licensing Page in Splunk Web

Current and permanent violations

Usage Report in Splunk Web

Current and previous 30 days usage

Handling License Violations

Review heavy hitters (Usage Report) and adjust intake

The daily limit resets at midnight

Buy more license

Demo

Licensing page in Splunk Web

Review Warnings and Violations

Usage Report

- Current
- Past 30 days

Enable Monitoring Console License Alerts

Working with Splunk Configuration Files

Overview

- Index-time
- Search-time

Precedence of configuration files Using btool to troubleshoot

Understanding Splunk configuration files

Structure of Splunk configuration files

Layering the configuration files

Understanding Splunk Configuration Files

Splunk platform is configured using set of text-based configuration files

Splunk Configuration Files

Text files located in SPLUNK_HOME/etc/…

.conf extension (Examples: server.conf, deploymentclient.conf)

Changes made via Splunk Web update configuration files

Contain stanzas and key value pairs

Govern a Splunk functionality

There are only two Splunk software packages: Splunk Enterprise and Splunk Universal Forwarder

Splunk Software Packages

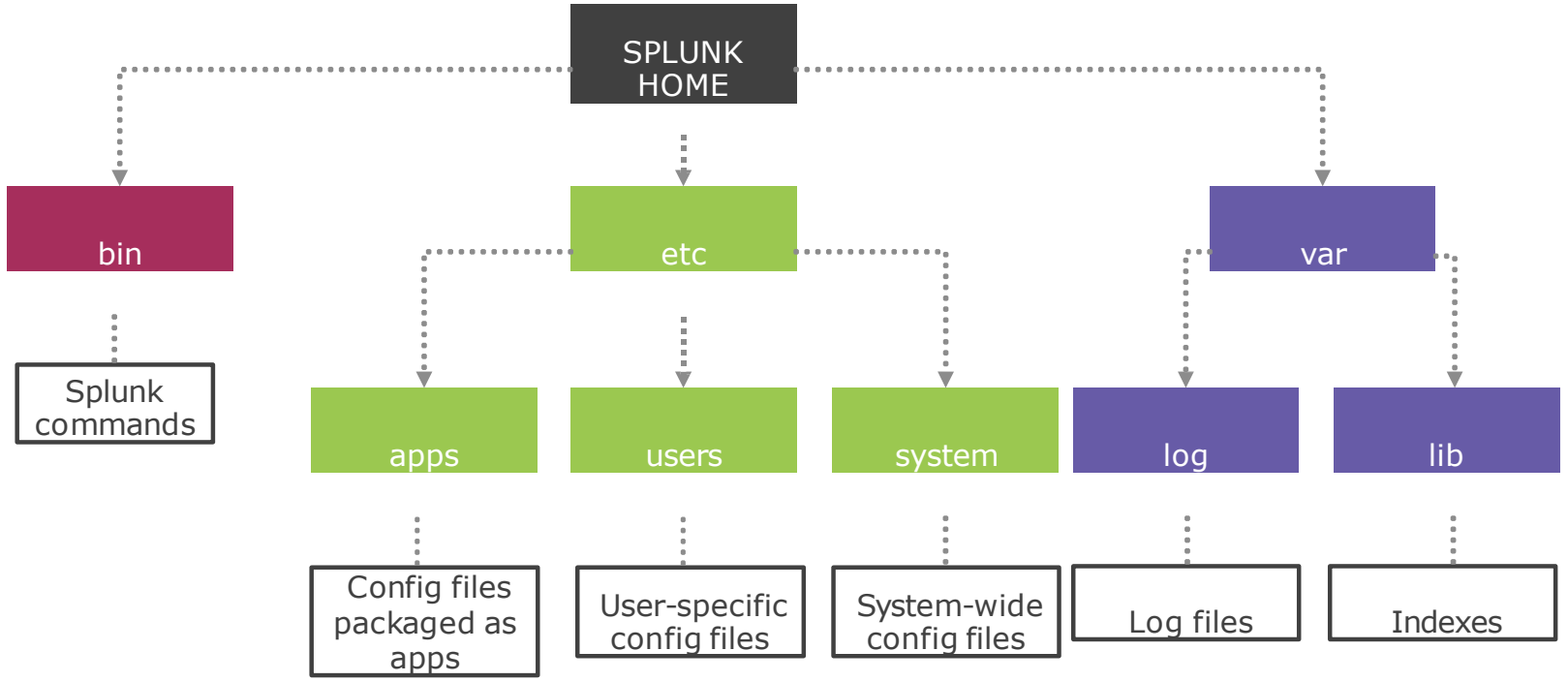
Splunk Enterprise

Indexer | Search Head | License Master | Deployment Server | Cluster Master | Search Head Deployer | Monitoring Console | Heavy Forwarder

Splunk Universal Forwarder

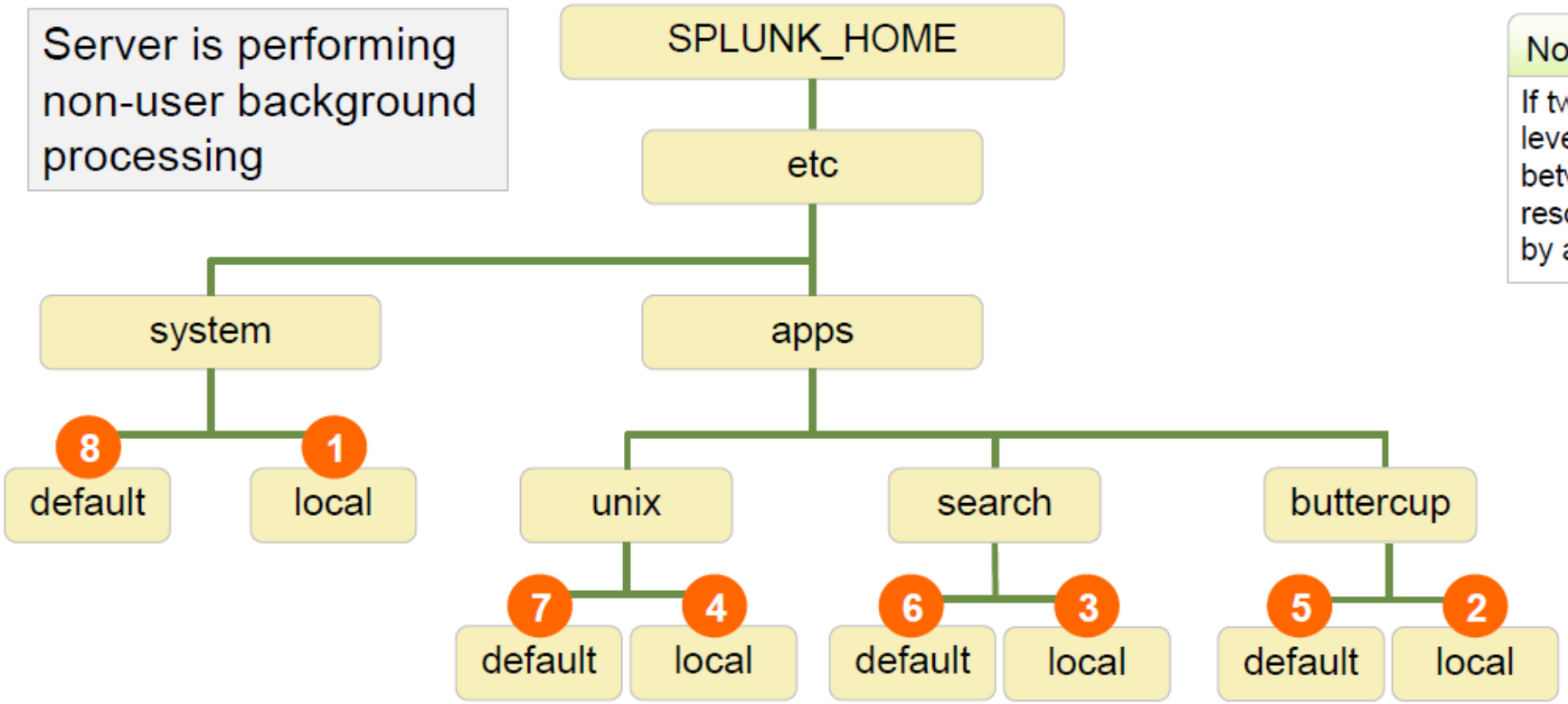
Deployment Client

Splunk Platform Directory Structure



Index Time Precedence – Adding an App

Server is performing non-user background processing



Note

If two or more apps at the same level of precedence have conflicts between them, the conflicts are resolved in **lexicographical** order by app directory name.

Structure of Splunk Configuration File

Configuration Files Are Made of Stanzas and Key-value Pairs

indexes.conf

```
[default]

maxTotalDataSizeMB = 650000

maxGlobalRawDataSizeMB = 0


# idx1 index settings

[idx1]

homePath = volume:hot1/idx1

coldPath = volume:cold2/idx1
```

inputs.conf

```
[monitor:///var/log/httpd]

sourcetype = access_common

ignoreOlderThan = 7d

index = web
```

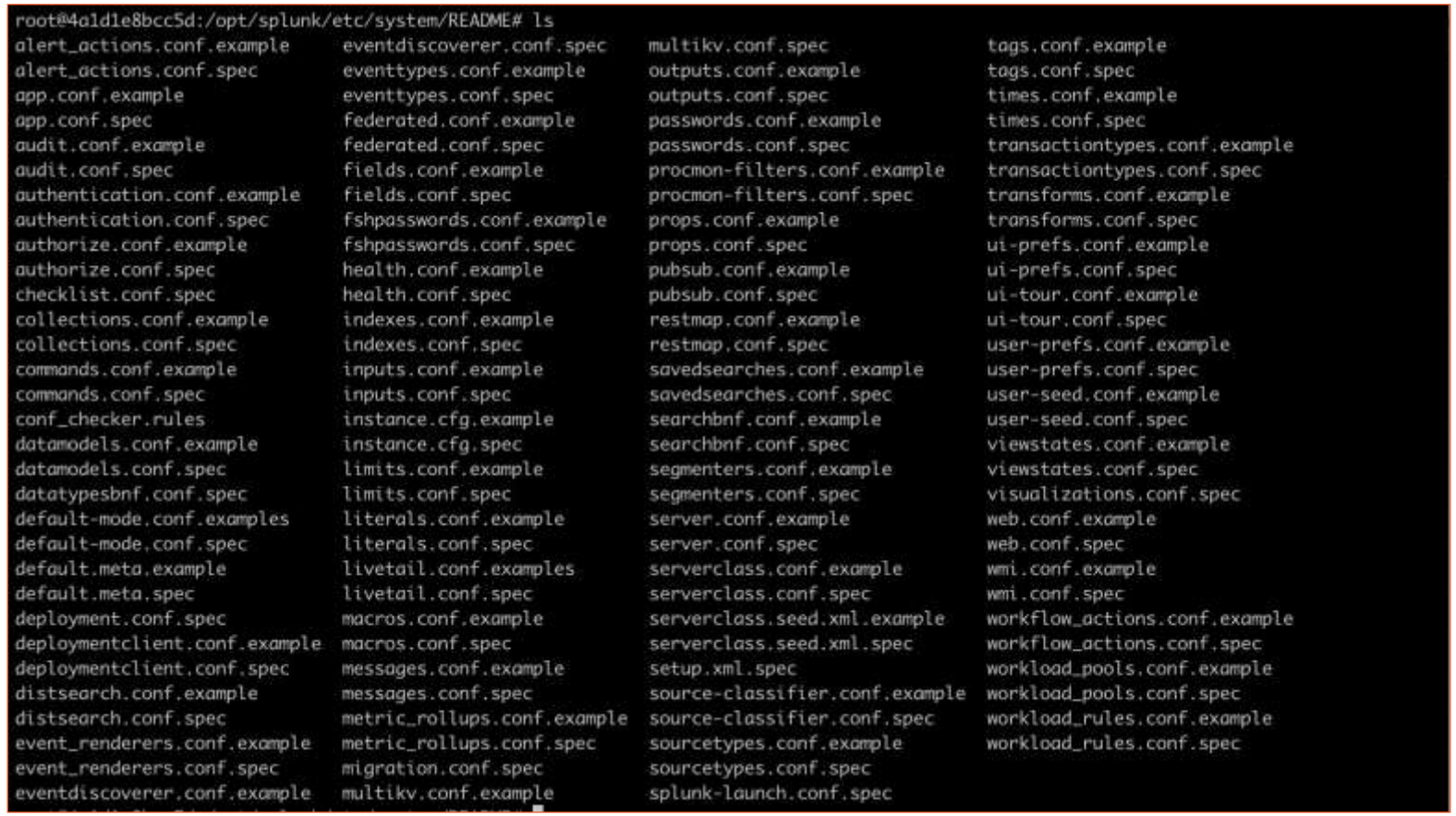
Two Ways to Learn Splunk Configuration Files

README files

$SPLUNK_HOME/etc/system/README directory contains example and spec files

Spec files at docs.splunk.com

https://docs.splunk.com/Documentation/Splunk


root@4a1d1e8bcc5d:/opt/splunk/etc/system/README# ls
alert_actions.conf.example
alert_actions.conf.spec
app.conf.example
app.conf.spec
audit.conf.example
audit.conf.spec
authentication.conf.example
authentication.conf.spec
authorize.conf.example
authorize.conf.spec
checklist.conf.spec
collections.conf.example
collections.conf.spec
commands.conf.example
commands.conf.spec
conf_checker.rules
datamodels.conf.example
datamodels.conf.spec
datatypesbnf.conf.spec
default-mode.conf.examples
default-mode.conf.spec
default.meta.example
default.meta.spec
deployment.conf.spec
deploymentclient.conf.example
deploymentclient.conf.spec
distsearch.conf.example
distsearch.conf.spec
event_renderers.conf.example
event_renderers.conf.spec
eventdiscoverer.conf.example
eventdiscoverer.conf.spec
eventtypes.conf.example
eventtypes.conf.spec
federated.conf.example
federated.conf.spec
fields.conf.example
fields.conf.spec
fshpasswords.conf.example
fshpasswords.conf.spec
health.conf.example
health.conf.spec
indexes.conf.example
indexes.conf.spec
inputs.conf.example
inputs.conf.spec
instance.cfg.example
instance.cfg.spec
limits.conf.example
limits.conf.spec
literals.conf.example
literals.conf.spec
livetail.conf.examples
livetail.conf.spec
macros.conf.example
macros.conf.spec
messages.conf.example
messages.conf.spec
metric_rollups.conf.example
metric_rollups.conf.spec
migration.conf.spec
multikv.conf.example
multikv.conf.spec
outputs.conf.example
outputs.conf.spec
passwords.conf.example
passwords.conf.spec
procmon-filters.conf.example
procmon-filters.conf.spec
props.conf.example
props.conf.spec
pubsub.conf.example
pubsub.conf.spec
restmap.conf.example
restmap.conf.spec
savedsearches.conf.example
savedsearches.conf.spec
searchbnf.conf.example
searchbnf.conf.spec
segmenters.conf.example
segmenters.conf.spec
server.conf.example
server.conf.spec
serverclass.conf.example
serverclass.conf.spec
serverclass.seed.xml.example
serverclass.seed.xml.spec
setup.xml.spec
source-classifier.conf.example
source-classifier.conf.spec
sourcetypes.conf.example
sourcetypes.conf.spec
splunk-launch.conf.spec
tags.conf.example
tags.conf.spec
times.conf.example
times.conf.spec
transactiontypes.conf.example
transactiontypes.conf.spec
transforms.conf.example
transforms.conf.spec
ui-prefs.conf.example
ui-prefs.conf.spec
ui-tour.conf.example
ui-tour.conf.spec
user-prefs.conf.example
user-prefs.conf.spec
user-seed.conf.example
user-seed.conf.spec
viewstates.conf.example
viewstates.conf.spec
visualizations.conf.spec
web.conf.example
web.conf.spec
wmi.conf.example
wmi.conf.spec
workflow_actions.conf.example
workflow_actions.conf.spec
workload_pools.conf.example
workload_pools.conf.spec
workload_rules.conf.example
workload_rules.conf.spec

Layering Splunk Configuration Files

Splunk App

Splunk's way of organizing configuration files

A directory under SPLUNK_HOME/etc/apps

Contains Splunk configuration files

Can also contain scripts and other necessary artifacts

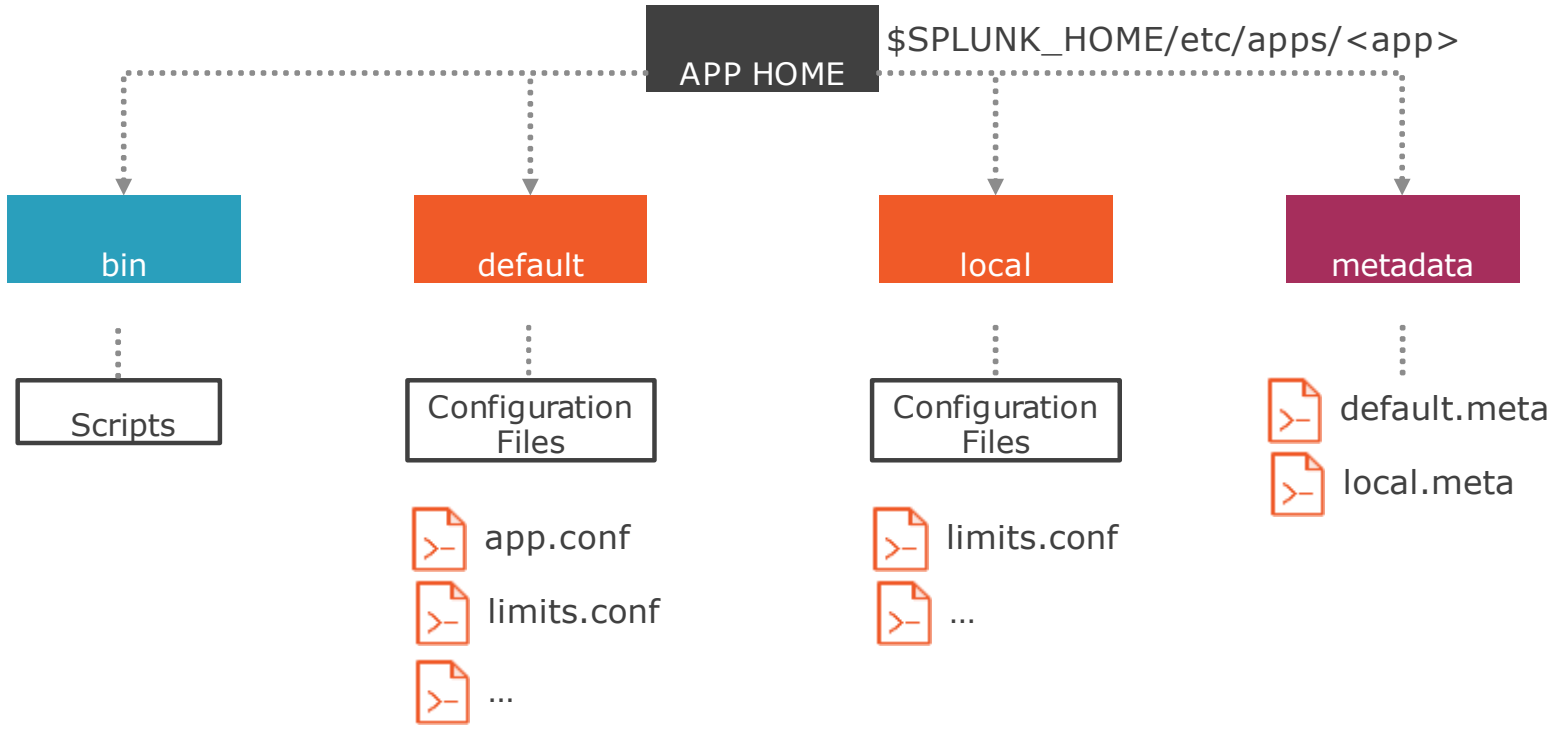
An add-on is an app that usually does not contain GUI

Search & Reporting app

Splunk ships with Search & Reporting app which is a prebuilt general-purpose app.

The configuration is stored under SPLUNK_HOME/etc/apps/search

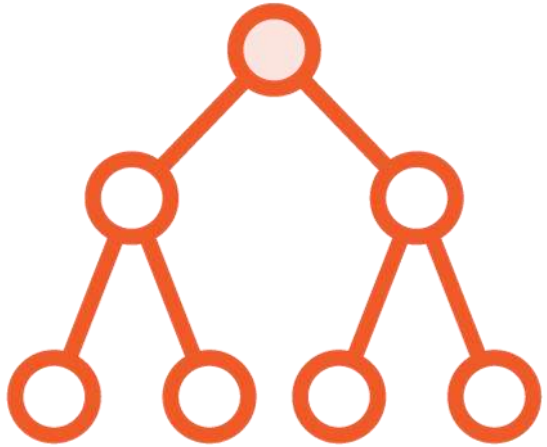
Splunk App Directory Structure



Default vs. Local Directories

Default	Local
Shipped with Splunk	User created
Will be overwritten upon Splunk upgrade	Will be preserved upon Splunk upgrade
Files should not be updated	Recommended place to modify files
Contains default settings	User-specific configuration changes
Does not override local	Always overrides default

All the Configuration File Locations



etc/system/default

etc/system/local

etc/apps/search/default

etc/apps/search/local

etc/apps/<app>/default

etc/apps/<app>/local

etc/users/<user>/<app>/local

Do not update files in etc/system/default

Splunk Configuration Files Precedence

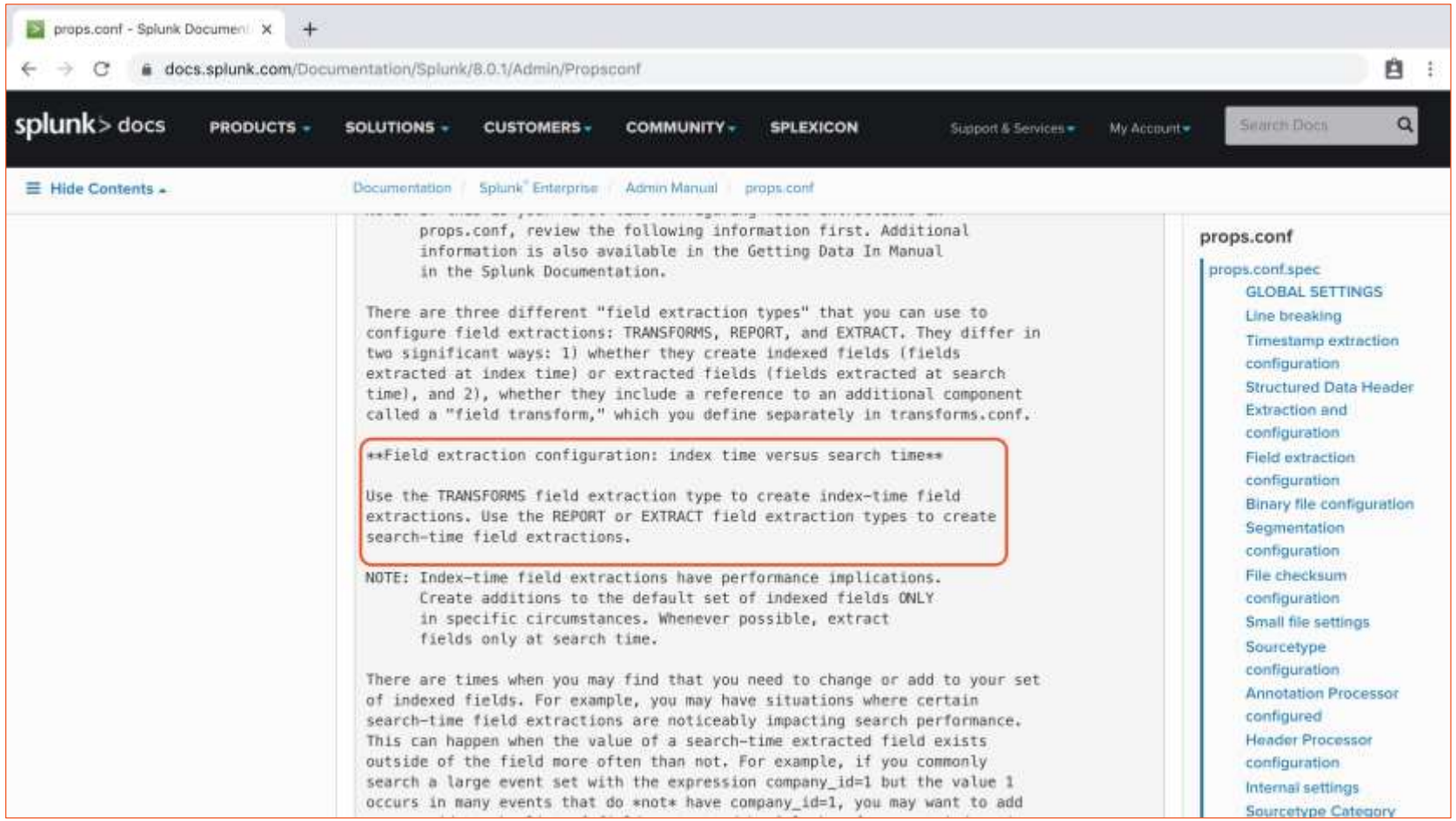
Index-time and Search-time

Index-time

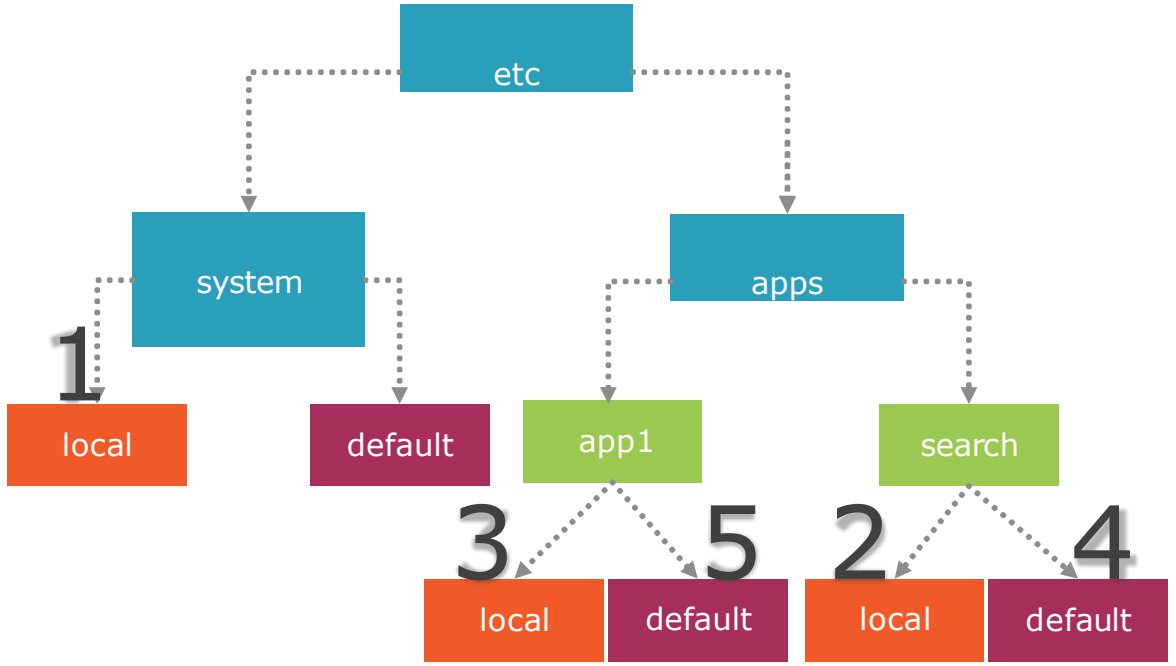
Global context, such as input/parsing configuration

Search-time

App/User scoped, such as a user's knowledge objects

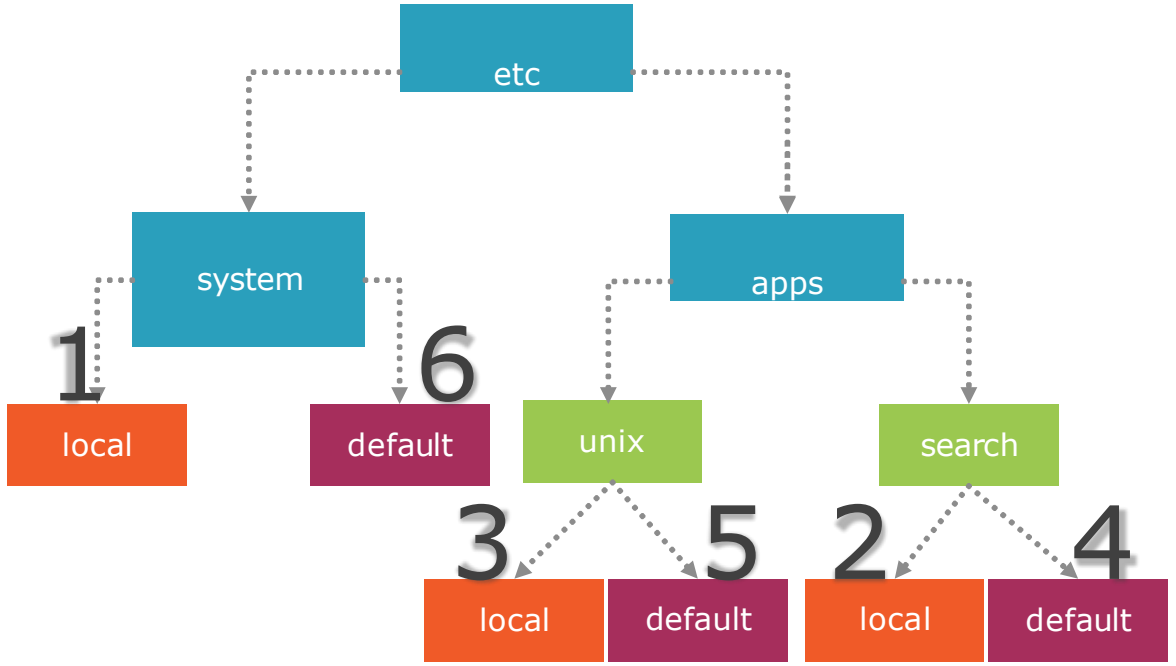

props.conf - Splunk Documen
docs.splunk.com/Documentation/Splunk/8.0.1/Admin/Propsconf
splunk> docs
PRODUCTS
SOLUTIONS
CUSTOMERS
COMMUNITY
SPLEXICON
Support & Services
My Account
Search Docs
Hide Contents
Documentation
Splunk® Enterprise
Admin Manual
props.conf
props.conf, review the following information first. Additional
information is also available in the Getting Data In Manual
in the Splunk Documentation.
There are three different "field extraction types" that you can use to
configure field extractions: TRANSFORMS, REPORT, and EXTRACT. They differ in
two significant ways: 1) whether they create indexed fields (fields
extracted at index time) or extracted fields (fields extracted at search
time), and 2), whether they include a reference to an additional component
called a "field transform," which you define separately in transforms.conf.
Field extraction configuration: index time versus search time
Use the TRANSFORMS field extraction type to create index-time field
extractions. Use the REPORT or EXTRACT field extraction types to create
search-time field extractions.
NOTE: Index-time field extractions have performance implications.
Create additions to the default set of indexed fields ONLY
in specific circumstances. Whenever possible, extract
fields only at search time.
There are times when you may find that you need to change or add to your set
of indexed fields. For example, you may have situations where certain
search-time field extractions are noticeably impacting search performance.
This can happen when the value of a search-time extracted field exists
outside of the field more often than not. For example, if you commonly
search a large event set with the expression company_id=1 but the value 1
occurs in many events that do *not* have company_id=1, you may want to add
props.conf
props.conf.spec
GLOBAL SETTINGS
Line breaking
Timestamp extraction configuration
Structured Data Header Extraction and configuration
Field extraction configuration
Binary file configuration
Segmentation configuration
File checksum configuration
Small file settings
Sourcetype configuration
Annotation Processor configured
Header Processor configuration
Internal settings
Sourcetype Category

Index-time Precedence

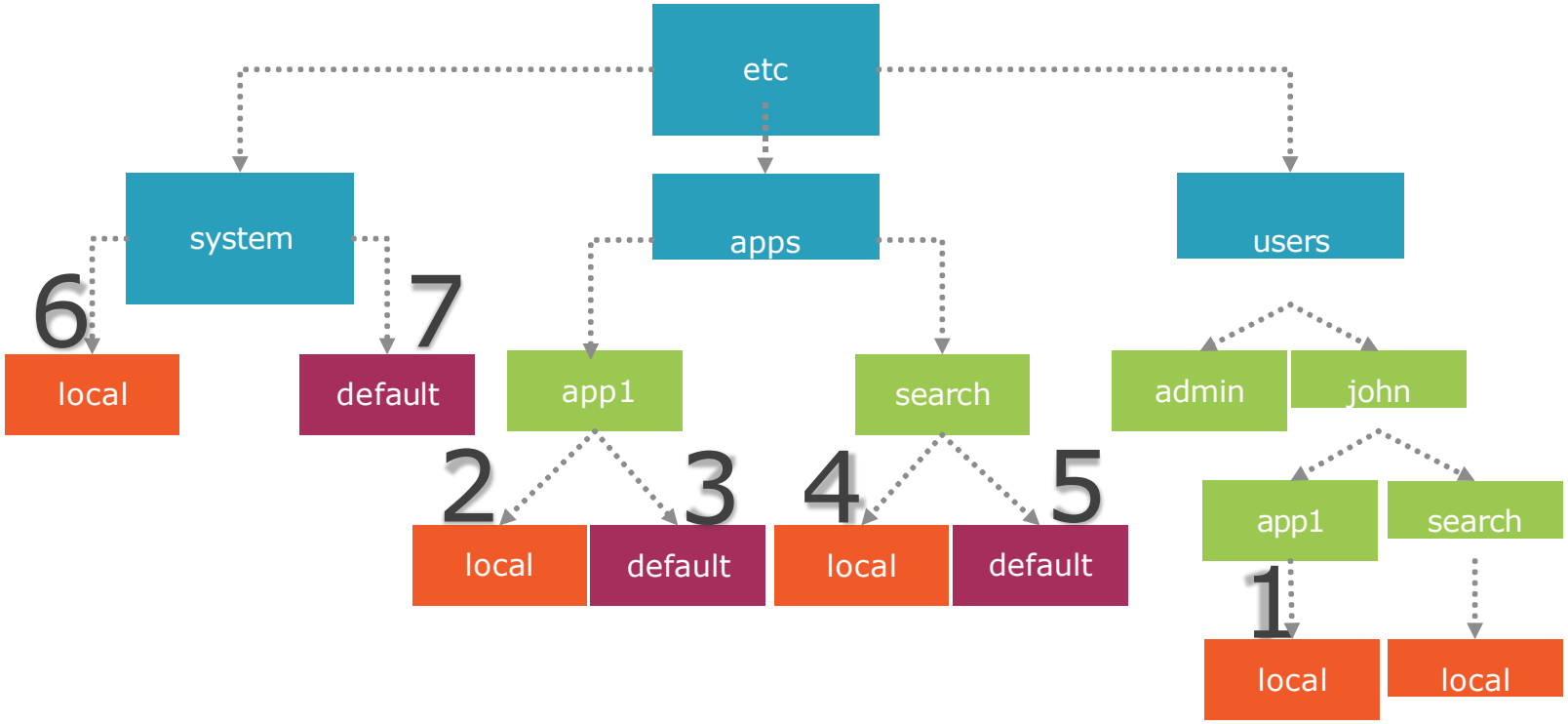


If two are more apps have conflicting settings, app directory name with highest ASCII order wins.

Index-time Precedence



Search-time Precedence



How Splunk Merges Configuration Files

Upon startup, Splunk merges configuration files for each type

The resulting file combines settings from various directory locations

Only one file per file type will be used at run-time

If there is a conflict, precedence is applied

Local takes precedence over default

Configuration File Merging Example 1

etc/apps/cybersecnerd_app_prop
s/local/ props.conf

```
[web:log]
SHOULD_LINEMERGE = false
LINE_BREAKER = ([\r\n]+)\d{4}-
```

etc/apps/chi_app_props/local/props.conf

```
[app:log]
TIME_FORMAT = %Y-%m-%d %H:%M:%S%Z
TIME_PREFIX = ^
```

etc/apps/jack_app_props/local/props.conf

```
[db:log]
BREAK_ONLY_BEFORE_DATE = true
SHOULD_LINEMERGE=true

[web:log]
category = web
```

Effective props.conf at runtime

```
[web:log]
SHOULD_LINEMERGE = false
LINE_BREAKER = ([\r\n]+)\d{4}-
category = web

[app:log]
TIME_FORMAT = %Y-%m-%d %H:%M:%S%Z
TIME_PREFIX = ^

[db:log]
BREAK_ONLY_BEFORE_DATE = true
SHOULD_LINEMERGE=true
```

Configuration File Merging Example 2

etc/system/default/limits.conf
[inputproc]
file_tracking_db_threshold_mb = 500
learned_sourcetypes_limit = 1000

etc/apps/cyber_limits_app/local/limits.conf
[inputproc]
max_fd = 500

etc/system/local/limits.conf
[inputproc]
max_fd = 300

Effective limits.conf at runtime

[inputproc]
file_tracking_db_threshold_mb = 500
learned_sourcetypes_limit = 1000
max_fd = 300

When overriding configuration with a local file, do not copy the entire file from default. Just add the overriding configuration.

Overriding the Default Configurations

Correct Way to Override Defaults

etc/system/default/props.conf

```
[default]  CHARSET =
UTF-8
LINE_BREAKER_LOOKBEHIND = 100
TRUNCATE = 10000

DATETIME_CONFIG = /etc/datetime.xml
ADD_EXTRA_TIME_FIELDS = True
ANNOTATE_PUNCT = True
…
```

etc/system/local/props.conf

```
[default]
TRUNCATE = 50000
```

Using btool to Work with Splunk Configuration Files

What is btool?

A Splunk command

Located in SPLUNK_HOME/bin

Retrieves the on-disk configuration of a Splunk configuration file

Syntax: splunk btool <conf file name> list [options]

--debug option shows the exact .conf file location

Btool Example

```
/opt/splunk/bin# ./splunk btool inputs list monitor:///var/log
[monitor:///var/log]
_rcvbuf = 1572864
disabled = false
host = cybersecnerd
index = main

/opt/splunk/bin# ./splunk btool inputs list monitor:///var/log -debug
/opt/splunk/etc/apps/search/local/inputs.conf [monitor:///var/log]
/opt/splunk/etc/system/default/inputs.conf     _rcvbuf = 1572864
/opt/splunk/etc/apps/search/local/inputs.conf disabled = false
/opt/splunk/etc/apps/search/local/inputs.conf host = cybersecnerd
/opt/splunk/etc/apps/search/local/inputs.conf index = main
```

MORE EXAMPLES for btool

Demo

Review a Splunk configuration file

Locate the btool command

Use btool to retrieve configuration

Use btool to analyze conflicts

Using REST API to retrieve configuration

Understanding Splunk Index

Overview

How Splunk organizes data

Index buckets

- Hot
- Warm
- Cold
- Frozen
- Thawed

Creating an index

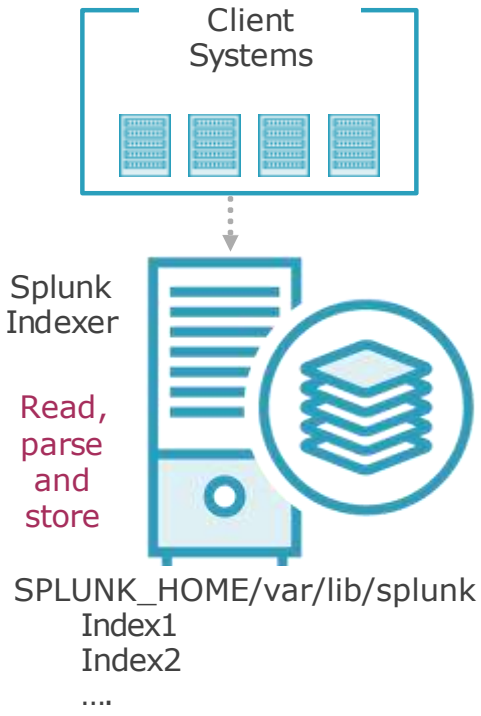
- Using Splunk Web
- Using configuration files
- One index vs multiple indexes

Index data integrity

How Splunk Organizes Data

Splunk stores data in indexes, which are organized in directories and files in disk

Splunk Indexes



Indexes are stored in SPLUNK_HOME/var/lib/splunk

The directory location is customizable for each index

Indexes contain raw data and index files

Indexes can be created by an Administrator

Many prebuilt indexes

- _internal
- _audit
- main

Inside a Splunk Index

Raw data

Raw data is stored in compressed format

Tsidx files

Time series index files that point to the raw data

metadata

Metadata files such as Sources.data, SourceTypes.data and Hosts.data

Splunk Data Buckets

Buckets

Indexes store data in buckets

Buckets are set of directories organized by age

Contain raw data, tsidx files and metadata

As the index grows, number of buckets grows as well

Types of Buckets

Hot

Warm

Cold

Frozen

Thawed

Hot

Hot buckets contain the newest data

They are open for both read and write

There can be more then one hot bucket in an index

Searchable

Roll to warm bucket when buckets reach certain size, or upon Splunk restart

Warm

Warm buckets are created when hot buckets roll

Not open for writing, but searchable

They reside in the same directory as hot buckets but renamed

Roll to cold buckets when exceeding maximum warm buckets setting




Cold

Starting from oldest bucket (based on time), warm buckets roll to cold

Reside in different directory from hot and warm

Searchable

The directory location can be configured

Possible to save cost by using cheaper storage

Frozen

After cold buckets age out based on retention policy, they roll to frozen

The default action is to delete; can be configured to archive

coldToFrozenDir or coldToFrozenScript in indexes.conf configures archiving

Archived frozen buckets are not searchable

Thawed

Frozen buckets can be thawed

Thawed buckets are rebuilt into the index and searchable

Location of thawed buckets can be configured in indexes.conf

No age restriction for thawed buckets

Use splunk rebuild command to rebuild

Bucket Locations

SPLUNK_HOME/var/lib/splunk/myfirstindex/db/

SPLUNK_HOME/var/lib/splunk/myfirstindex/db/

SPLUNK_HOME/var/lib/splunk/myfirstindex/colddb/

Frozen buckets are deleted by default. Can be optionally archived.

SPLUNK_HOME/var/lib/splunk/myfirstindex/thaweddb/

Bucket Naming

Hot

hot_v1_<local id>

hot_v1_5

Warm

db_<newest time>_<oldest_time>_<local id>

db_1559676230_1559676181_0

Local id is the ID of the bucket

Newest and oldest time are in UTC epoch time in seconds

During a search, Splunk uses the time range in the bucket name before opening it

Index Directory

```
/opt/splunk/var/lib/splunk
myfirstindex
    db
        .bucketManifest
        CreationTime
        hot_v1_0
            .rawSize
            bucket_info.csv
            splunk-autogen-params.dat
            Hosts.data
            Sources.data
            Strings.data
            SourceTypes.data
            1576088642-1575418501-12210397385432534163.tsidx
            rawdata
                l1Hashes_0_2013DDA2-630D-4FFE-BD9E-7EC7E2D8C74B.dat.tmp
                slicesv2.dat
                journal.gz
                24935359
    colddb
    thaweddb
    datamodel_summary
```

Creating Splunk Indexes

Why Create Multiple Indexes?

Security

Restrict access to index by Splunk role

Retention

Retention policies are applied at index level

```
authorize.conf

[role_myrole]

importRoles = user

srchIndexesAllowed = os,idx1

srchIndexesDefault = idx1


[role_mysuperuserrole]

importRoles = user

srchIndexesAllowed = os,idx1,idx2

srchIndexesDefault = idx2
```

- Security is defined in authorize.conf
- Definition for myrole
- Inherit user role's capabilities
- Allowed indexes are os and idx1
- Default index is idx1
- Allowed indexes are os,idx1,idx2
- Default index is idx2

```
indexes.conf

[myfirstindex]

coldPath =
$SPLUNK_DB/myfirstindex/colddb

enableDataIntegrityControl = 0

enableTsidxReduction = 0

homePath =
$SPLUNK_DB/myfirstindex/db

maxTotalDataSizeMB = 512000

frozenTimePeriodInSecs = 1209600


thawedPath =
$SPLUNK_DB/myfirstindex/thaweddb
```

- Indexes are configured in indexes.conf
- Definition for myfirstindex
- Location for cold buckets
- Disable Data Integrity Control
- Disable TSIDX reduction
- Location for hot and warm buckets
- Maximum size of the index
- Number of seconds after which bucket roles to frozen (every event in a bucket must be older than this limit)
- Location for thawed buckets

Creating Splunk Index

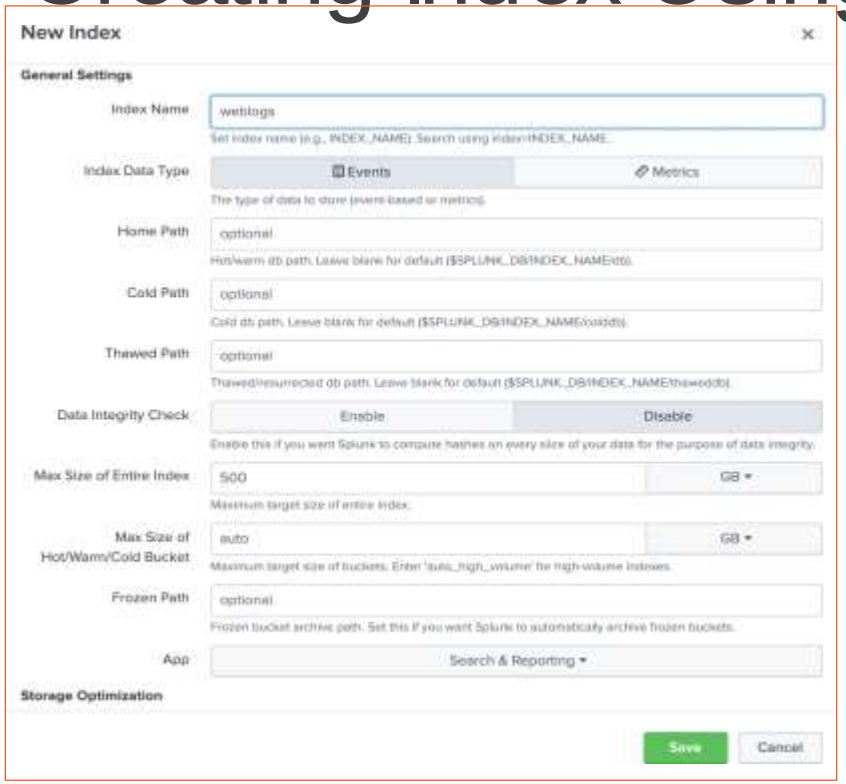
Using Splunk Web

Settings -> Indexes -> New Index

Using configuration files

Copy existing stanza in indexes.conf and update. Restart of Splunk required

Creating Index Using



Index Data Integrity

enableDataIntegrityControl=true in indexes.conf.

Check integrity:
./splunk check-integrity -index [index name]

Allows to ensure indexed data has not been tampered with.

Regenerate hash:
./splunk generate-hash-files -index [index name]

Creates hash files (using SHA 256) as the data is indexed.

Hash files are stored in rawdata directory within the index.

One Index vs. Multiple Indexes

Security

Create separate indexes if you want to allow access selectively

Retention

Create separate indexes if you want varying data retention periods

Management

Easier management for chargeback processes

Demo

Create a Splunk index using Splunk Web

Review the indexes.conf

Upload data

Review the index directory structure

- buckets
- tsidx files
- raw data

Check data integrity

Configuring Indexes

Overview

Learning to tune indexes.conf

Understanding fishbucket

Applying a data retention policy

Managing Splunk index

- Backup
- Deleting events
- Cleaning out

Using monitoring-console to understand index configuration

Splunk Index Configuration File

Indexes.conf file is used to configure Splunk indexes and their properties

Location of indexes.conf

There is a default indexes.conf in SPLUNK_HOME/etc/system/default directory.

DO NOT edit this file.

Create a new indexes.conf in SPLUNK_HOME/etc/system/local directory and add specific customization there.

You can also place indexes.conf as part of an app under SPLUNK_HOME/etc/apps/<app name>/local.

Restart of splunkd required for any configuration changes.

Structure of indexes.conf

The file can have a 'default' stanza for defining global properties

If a property is defined outside of any stanza, at the top of the file, it is considered a global property.

If a property is defined at both global level and in a specific stanza, value in the stanza takes precedence.

If there are multiple definitions of the same settings in a stanza, last setting wins.

Indexes.conf

```
# A basic index definition

[weblogs]

homePath = $SPLUNK_DB/weblogs/db

coldPath = $SPLUNK_DB/weblogs/colddb

thawedPath = $SPLUNK_DB/weblogs/thaweddb

tstatsHomePath = $SPLUNK_DB/weblogs/datamodel_summary

frozenTimePeriodInSecs = 5184000
```

```
homePath = $SPLUNK_DB/weblogs/db

coldPath = $SPLUNK_DB/weblogs/colddb

thawedPath = $SPLUNK_DB/weblogs/thaweddb

tstatsHomePath =
$SPLUNK_DB/weblogs/datamodel_summary

frozenTimePeriodInSecs = 5184000

maxHotBuckets = 3

maxDataSize = auto


maxTotalDataSizeMB = 400000

maxWarmDBCount = 300
```

- Directory for hot and warm buckets
- Directory for cold buckets
- Directory for thawed buckets
- Directory for accelerated data model summary tsidx files
- Freeze data older than this many seconds
- Maximum number of hot buckets
- Maximum size of hot buckets (default 750MB)
- Maximum size of an index
- Maximum number of warm buckets

Understanding Fishbucket

Splunk Fishbucket

fishbucket is a special Splunk internal index that is automatically created.

Checkpoint

Fishbucket keeps track of the ingestion progress of monitored files and directories

Location

It is located in SPLUNK_HOME/var/lib/splunk/fishbucket.

Use

Upon restart, using fishbucket, Splunk can start ingesting from where it left off

Using Fishbucket

You can use fishbucket to re-index files

To re-index a particular file

Use btprobe command

./splunk cmd btprobe -d SPLUNK_HOME/var/lib/splunk/fishbucket/splunk_private_db --file <file name> --reset

To re-index all monitored files

Remove the entire fishbucket directory

rm -rf /opt/splunk/var/lib/splunk/fishbucket

You must restart Splunk Forwarder

Applying a Data Retention Policy

Data Retention in Splunk

You must define a retention policy for the data indexed

Retention policy is applied at index level

Set maxTotalDataSizeMB and/or frozenTimePeriodInSecs

maxTotalDataSizeMB overrides frozenTimePeriodInSecs

indexes.conf is used to configure retention ploicy

Configuring Data Retention

maxTotalDataSizeMB

Maximum size of the index in Mega Bytes. Oldest data is frozen after this limit. Default is 500GB

frozenTimePeriodInSecs

Time period in seconds after which the data rolls to frozen. Default is 6 years

What Happens to the Expired Data?

When the bucket rolls from cold to frozen, by default the data is deleted

If coldToFrozenScript is configured, the script is executed

If coldToFrozenDir is configured, Splunk moves the expired buckets to this directory

To restore expired data, copy the archived buckets to thaweddb location and rebuild

- ./splunk rebuild SPLUNK_HOME/var/lib/splunk/<index name>/thaweddb/<bucket name>

Managing Splunk Indexes

Backing up Splunk

You must regularly backup Splunk. Daily incremental backups recommended.

Objects to backup:
Warm and cold buckets;
Entire etc directory.

Hot buckets can't be backed up without stopping Splunk.

In a clustered environment, you may not need to backup data buckets as data is replicated.

In distributed environments, ensure you backup search heads and heavy forwarders.

Deleting Events in a Splunk Index

Best way is to let the data expire instead of deleting

You can only do a virtual delete (i.e data is not removed from disk)

Even admins can't delete data by default

Create a user with can_delete capability.

Run a search to list the desired events to be deleted.

Pipe the delete command.

Cleaning out a Splunk Index

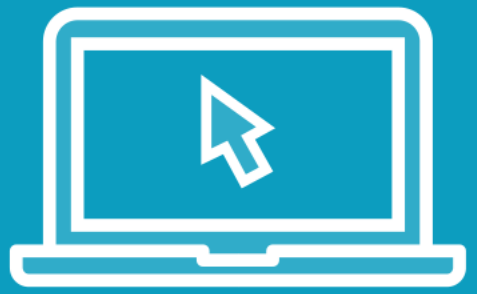
Extremely dangerous command in production

Destroys index data from disk

syntax:

splunk clean all -index <index name>

Demo



Review indexes.conf

- Data retention policy

Review location and contents of fishbucket

Reset fishbucket to re-index data

Deleting events from an Index

Use monitoring console to monitor indexes

Summary

Congratulations!

Licensing options and handling license violations

Configuration files layering and precedence

Using btool to troubleshoot

Creating and tuning Splunk indexes

Various Splunk data buckets

Configuring a data retention policy

Managing Splunk indexes

Practical Splunk for Users/PowerUsers in 2 Hours

- Working with Configuration Files and Indexes
- Understanding SPLUNK Admin Basics AND License Management