Running Splunk in an Air-gapped environment

Steve Schohn
Staff Sales Engineer | Splunk

Forward-Looking Statements

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

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

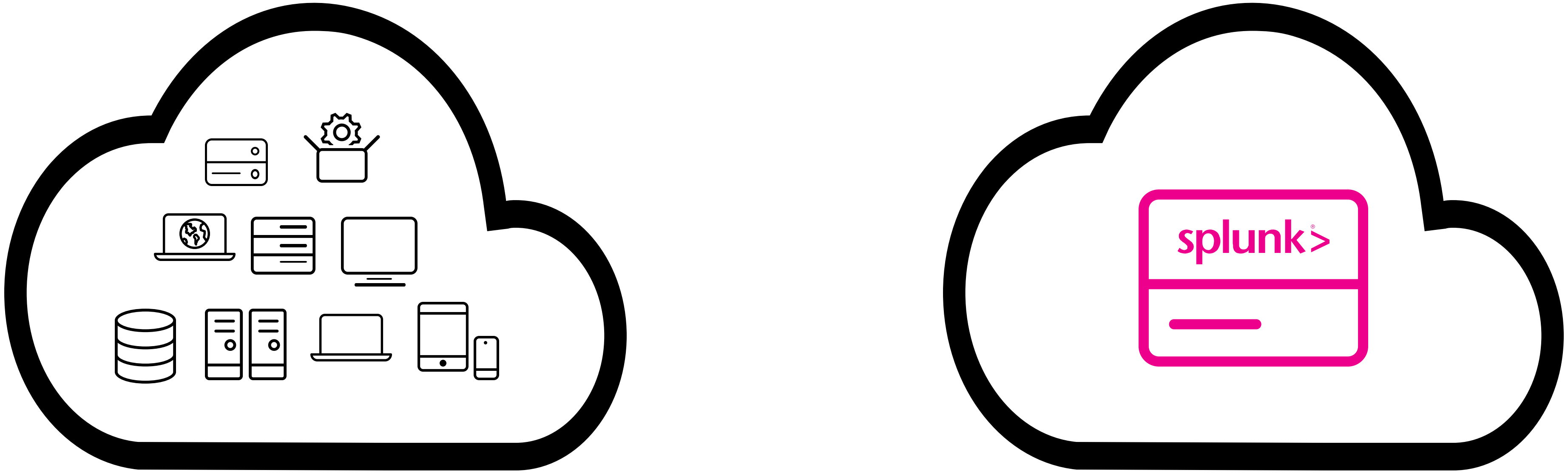
Splunk, Splunk>, Turn Data Into Doing, The Engine for Machine Data, Splunk Cloud, Splunk Light and SPL are trademarks and registered trademarks of Splunk Inc. in the United States and other countries. All other brand names, product names, or trademarks belong to their respective owners. © 2019 Splunk Inc. All rights reserved.

The Challenge

Nobody said it was easy

Are you in the right place?

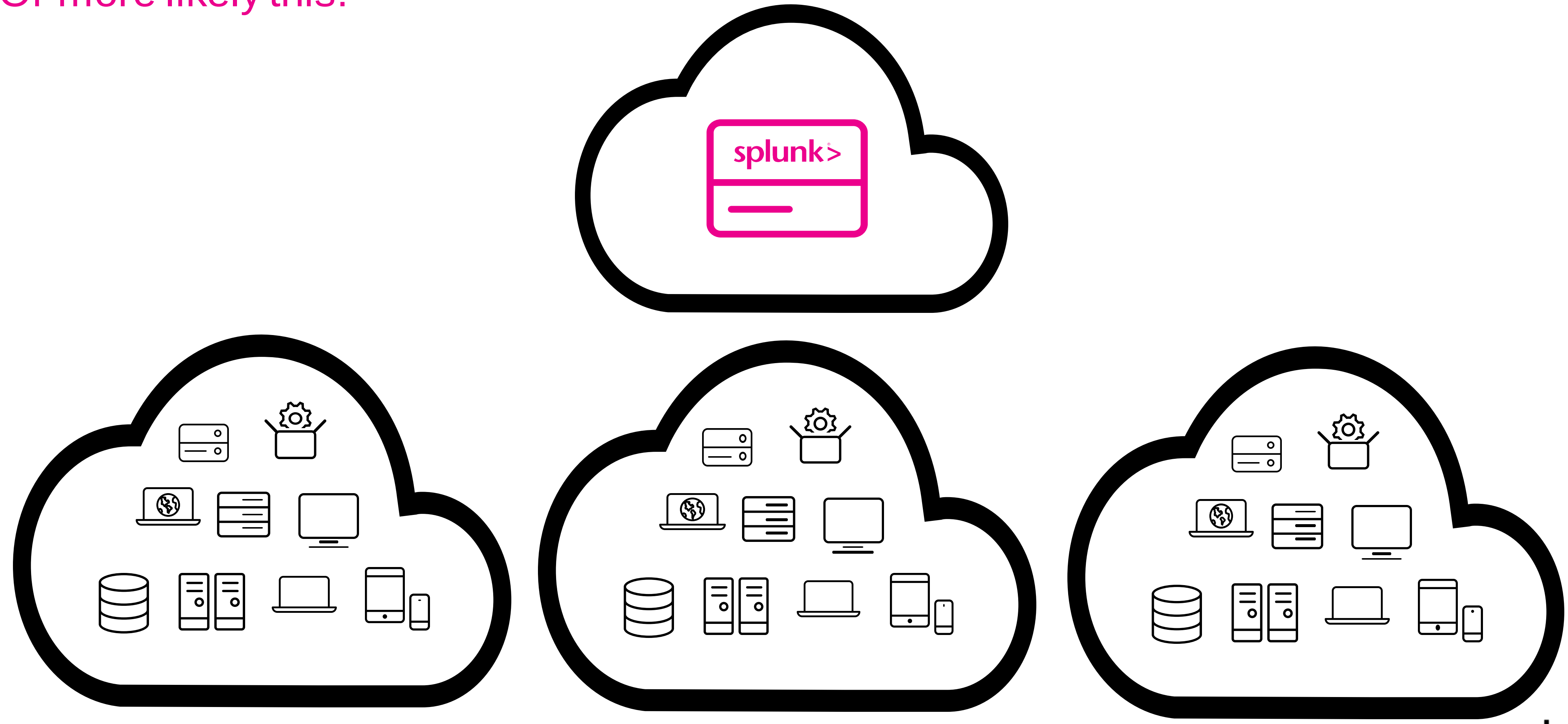
You have an architecture that looks like this:



Data sources are on one network. **Splunk is on another.** You cannot use forwarders to get data from one to another.

Are you in the right place?

Or more likely this:

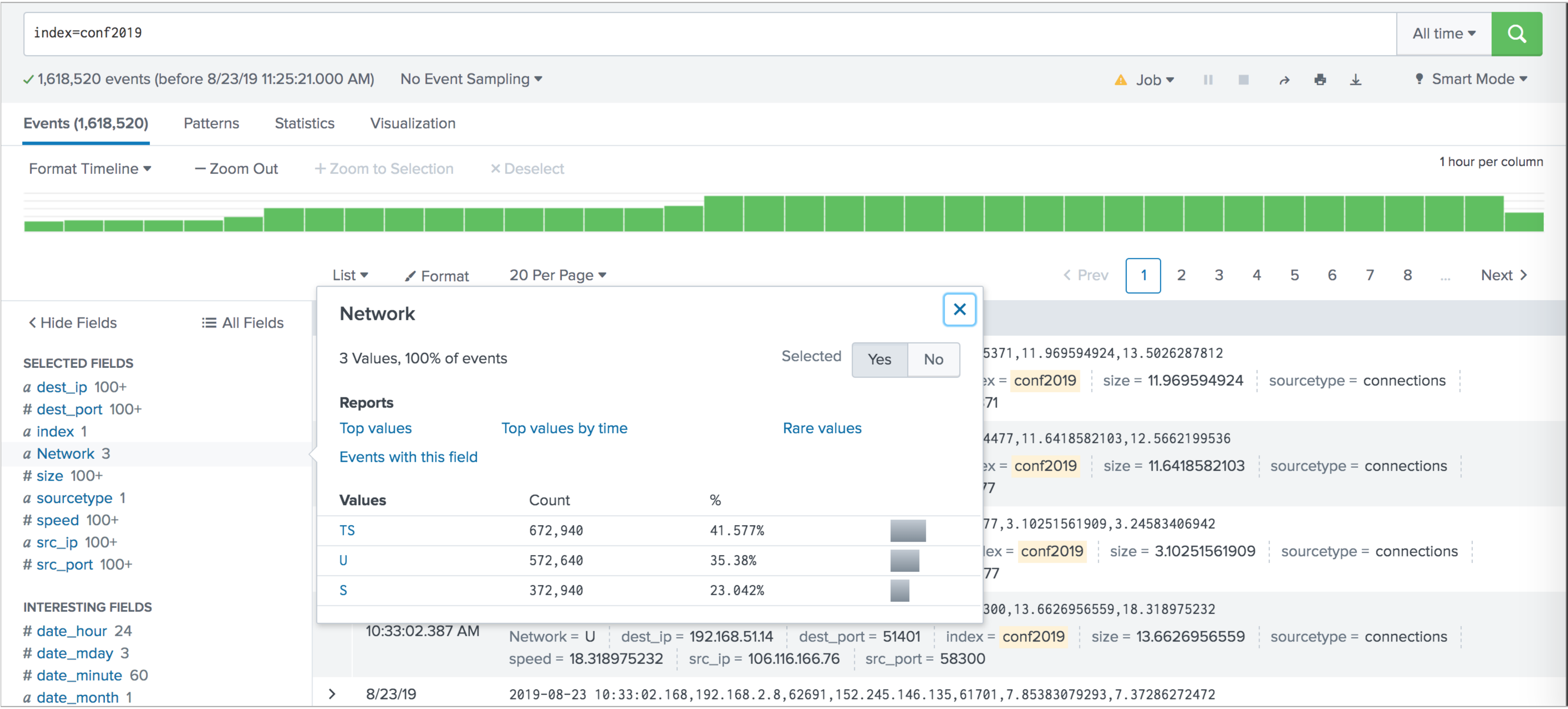


A Land Without TCP

- There is one-way connectivity from the network with data sources to the network running Splunk.
- The connectivity is limited to a few unidirectional protocols: FTP, UDP, etc
- If a true sneakernet: that way lies madness, and is outside the scope of this talk.
- TCP is specifically not an option: the SYN-SYNACK-ACK is unavailable, and therefore so is the UF->Indexer connection

Data can go up the stack, but never down

The Goal



```
index=conf2019
```

✓ 1,618,520 events (before 8/23/19 11:25:21.000 AM) No Event Sampling ▾

SELECTED FIELDS

- a dest_ip 100+
- # dest_port 100+
- a index 1
- a Network 3
- # size 100+
- a sourcetype 1
- # speed 100+
- a src_ip 100+
- # src_port 100+

INTERESTING FIELDS

- # date_hour 24
- # date_mday 3
- # date_minute 60
- a date_month 1

Network

3 Values, 100% of events

Selected [✓ Yes] No

Reports

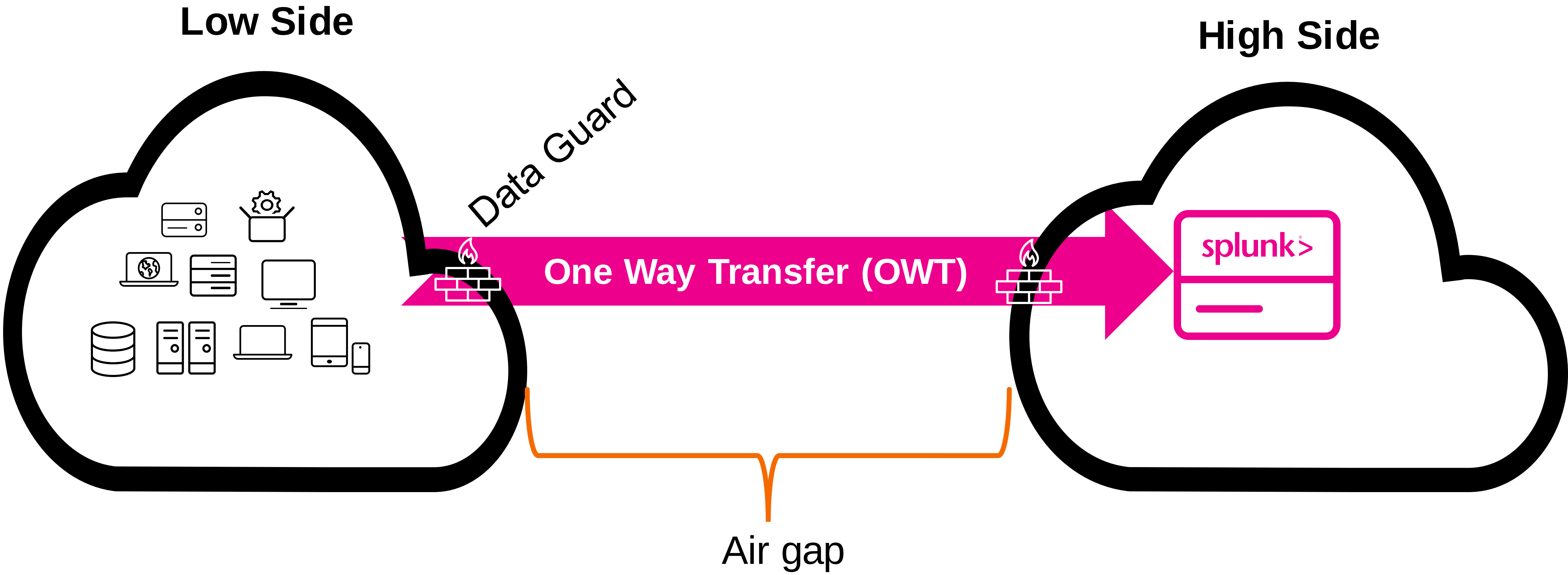
Top values Top values by time Rare values

Events with this field

Values	Count	%
TS	672,940	41.577%
U	572,640	35.38%
S	372,940	23.042%

```
5371,11.969594924,13.5026287812
ex = conf2019 | size = 11.969594924 | sourcetype = connections
71
4477,11.6418582103,12.5662199536
ex = conf2019 | size = 11.6418582103 | sourcetype = connections
77
77,3.10251561909,3.24583406942
lex = conf2019 | size = 3.10251561909 | sourcetype = connections
77
300,13.6626956559,18.318975232
10:33:02.387 AM   Network = U | dest_ip = 192.168.51.14 | dest_port = 51401 | index = conf2019 | size = 13.6626956559 | sourcetype = connections
                  speed = 18.318975232 | src_ip = 106.116.166.76 | src_port = 58300
> 8/23/19   2019-08-23 10:33:02.168,192.168.2.8,62691,152.245.146.135,61701,7.85383079293,7.37286272472
```

Some quick definitions

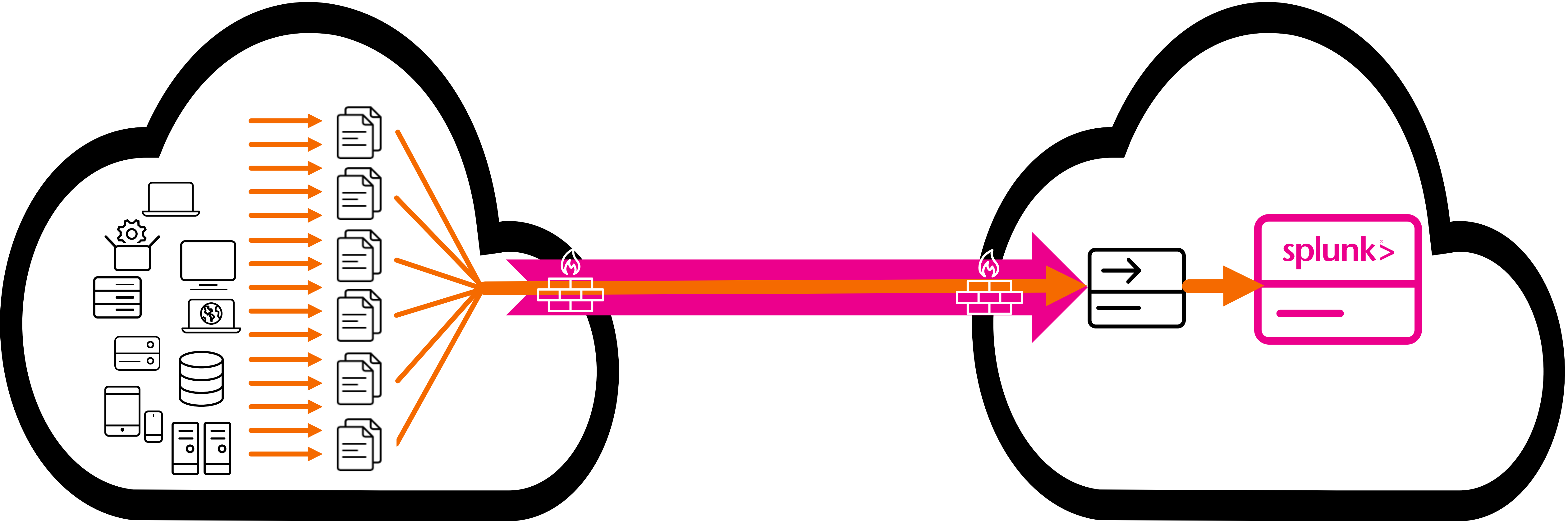


Getting Data In, Across the Air-gap

A Tour of What I’ve Seen

Method 1: Save data low as files, move files through OWT and index high

Use a combination of syslog, WEF, and native formats to push individual files through OWT



Should you do this?

No!

A resounding no!

Please no!

splunk> .conf19

Method 1: Save data low as files, move files through OWT and index high

Pros

Often the first thing people try?

Cons

I don’t care how good your naming convention is: at scale, eventually you will mess up the originating metadata

- Data will end up on high associated with the wrong host, source, and/or sourcetype

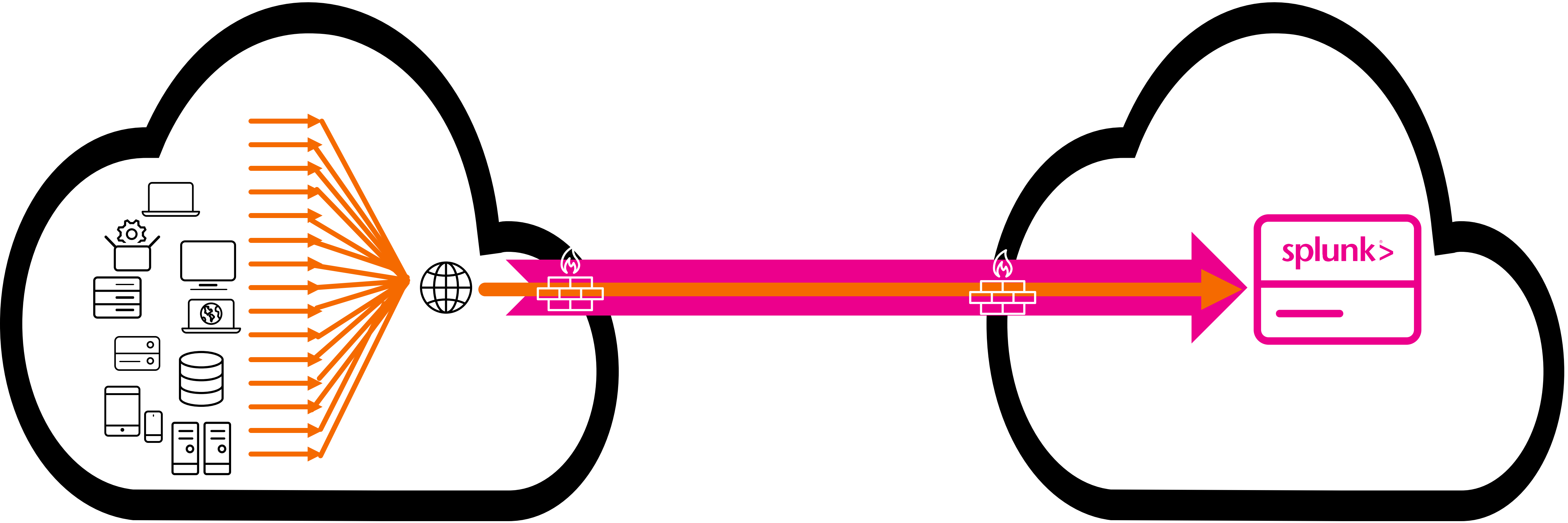
It’s a nightmare to maintain, let alone continually scale

- Imagine having thousands of hosts, and trying to copy up all the files in each’s /var/log/ while keeping the host correct

Windows events are especially ugly this way

Method 2: UDP data across OWT, index it high

Send data via syslog/UDP across OWT. Receive on the high side like standard UDP inputs.



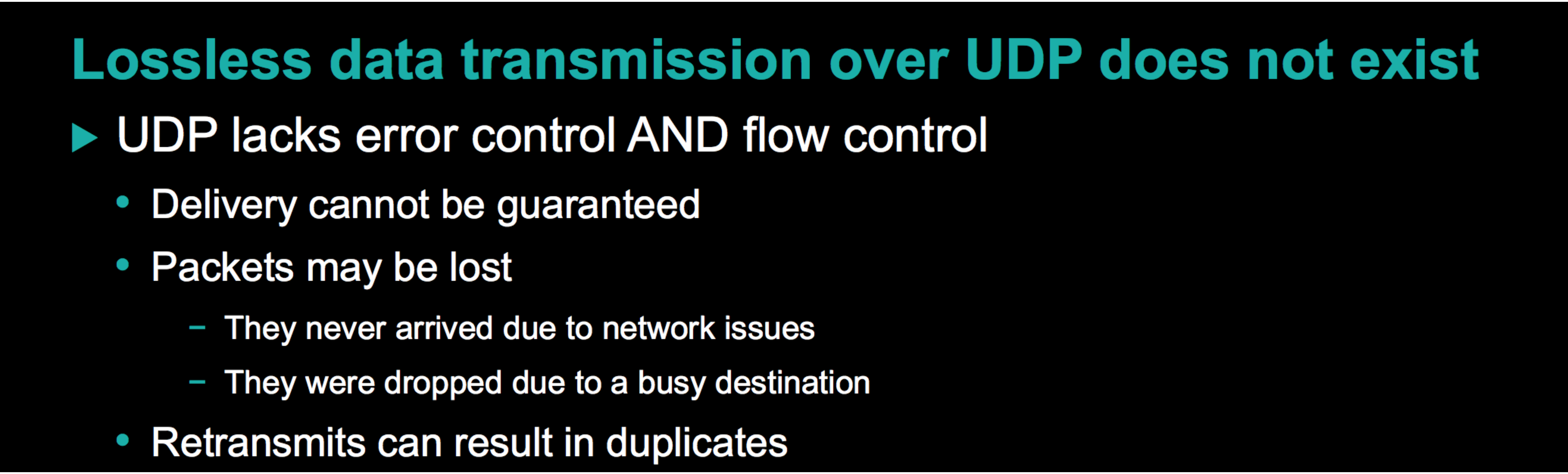
Should you do this?

Almost definitely not!

Method 2: UDP data across OWT, index it high

In short: You will lose data

All the reasons not to use UDP as described in Jeff Champagne’s “Worst Practices and How to Avoid Them” past .conf talks come into play here:



Lossless data transmission over UDP does not exist

- UDP lacks error control AND flow control
  - Delivery cannot be guaranteed
  - Packets may be lost
    - They never arrived due to network issues
    - They were dropped due to a busy destination
  - Retransmits can result in duplicates

Method 2: UDP data across OWT, index it high

Pros:
The one use case where this does make sense: when you care about real time way more than you care about potential data loss

If you are ok with getting less than 100% of your data, but you want it searchable immediately, this can be made to work

No low side hardware deployment is necessary

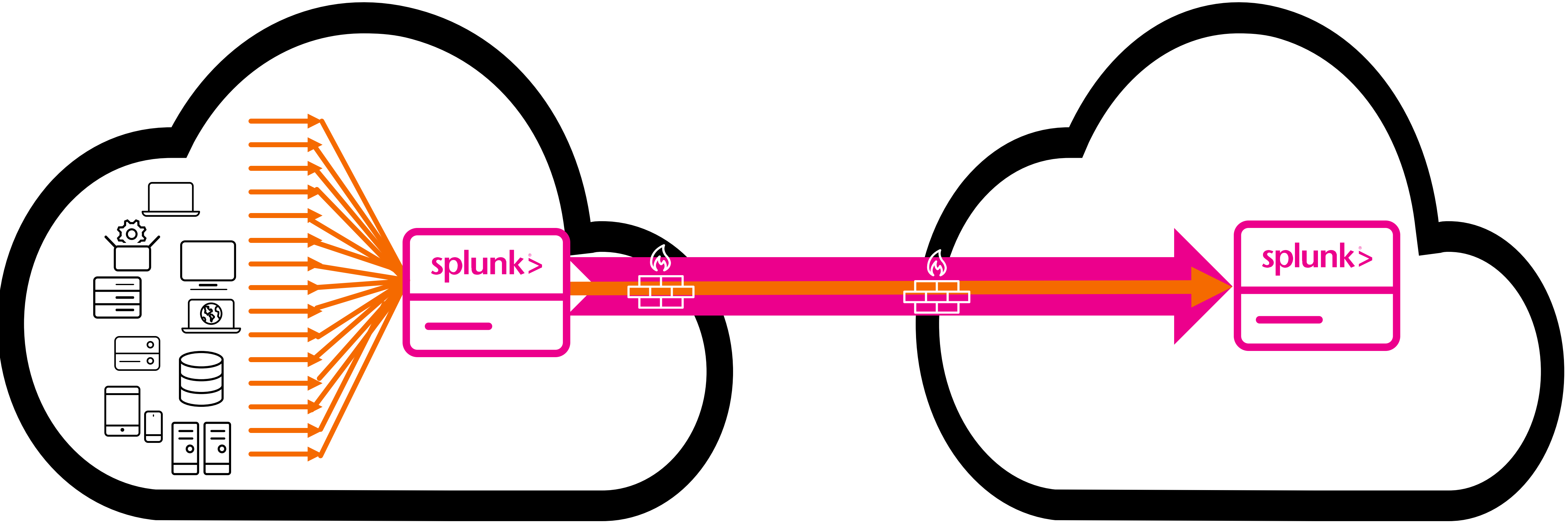
Cons:
Works best when you have mostly network device data. Endpoint host data will get messy in syslog form – you’ll have whitespace and truncation issues.

Again, you’re probably going to have host/source/sourcetype pain: make sure each feed comes in on its own port

And again, **YOU WILL LOSE DATA**

Method 3: Index data low, move the bucket files up

Set up indexers on both low side and high side. Index low side data as normal. On a schedule, move up the low side bucket files into the high side Splunk indexes.



Should you do this?

Probably not!

splunk> .conf19

Method 3: Index data low, move buckets up

Pros:
You're guaranteed fidelity of the data

Conceptually it's the most obvious to explain and maintain

If you don't need real time data, and are ok with a daily restart, this can be a workable model

Cons:
You need to wait for low-side buckets to roll to warm before moving them

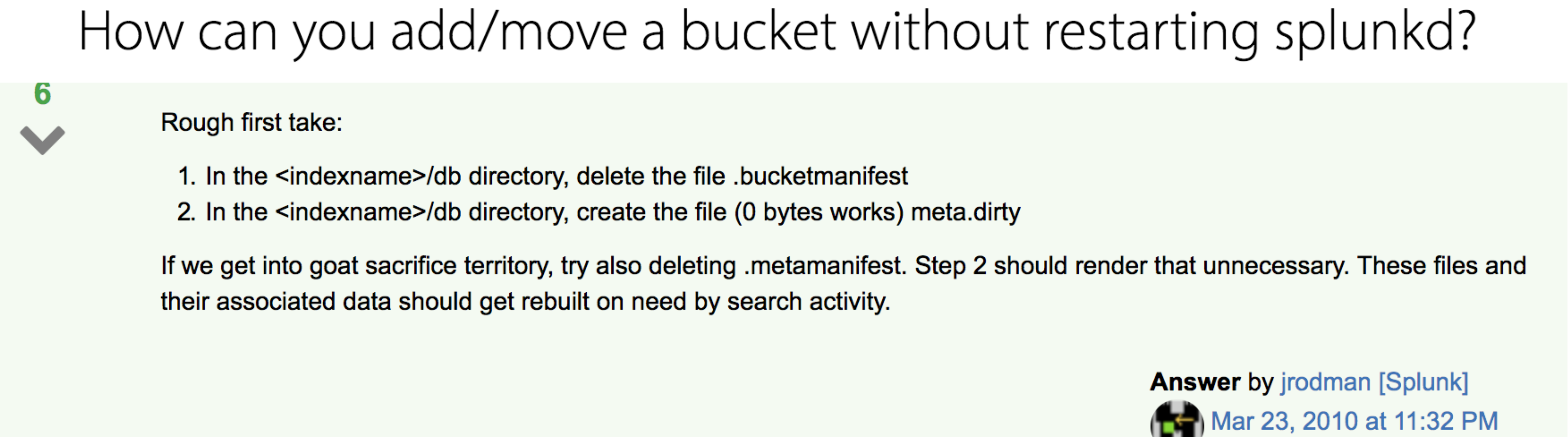
The buckets moved to high won't be searchable without a restart of splunkd

You need to ensure you don't have bucket id collisions: each bucket must have a unique id number. So prepare to rename the bucket, or have separate indexes on low and high.

Very difficult to move buckets from clustered systems -> non-clustered systems, and vice versa

Method 3: A Quick Addendum!

But but but Splunk Answers: https://answers.splunk.com/answers/838/how-can-you-add-move-a-bucket-without-restarting-splunkd.html



How can you add/move a bucket without restarting splunkd?

Rough first take:

1. In the <indexname>/db directory, delete the file .bucketmanifest
2. In the <indexname>/db directory, create the file (0 bytes works) meta.dirty

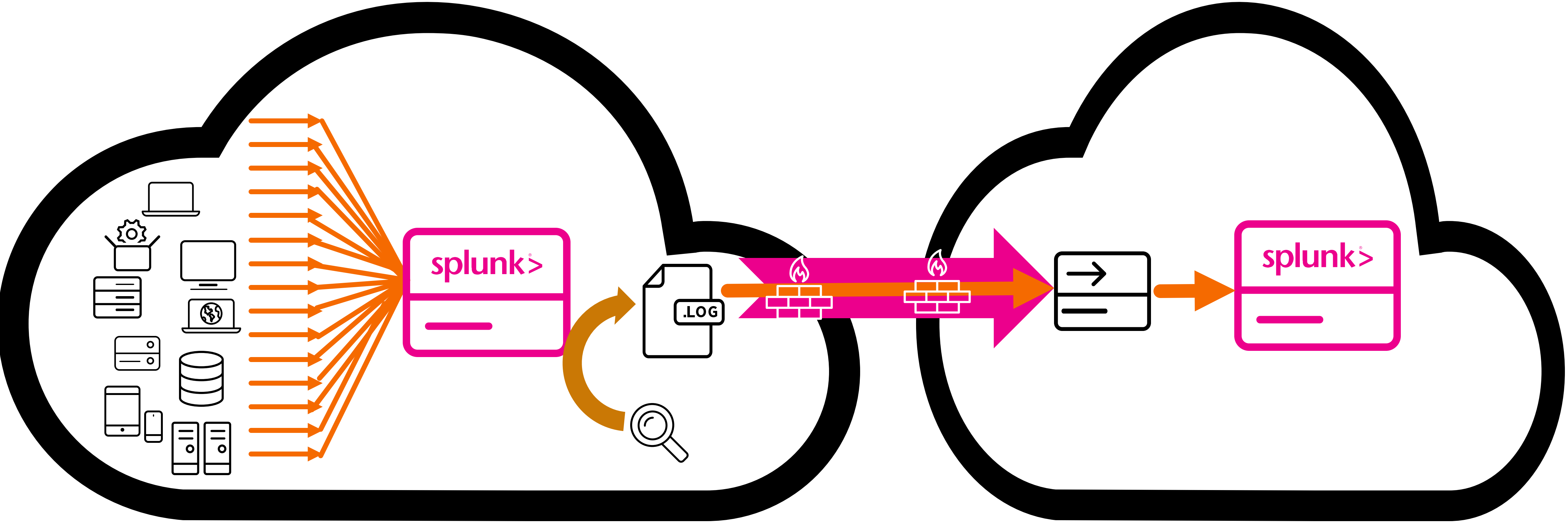
If we get into goat sacrifice territory, try also deleting .metamanifest. Step 2 should render that unnecessary. These files and their associated data should get rebuilt on need by search activity.

Answer by jrodman [Splunk]
Mar 23, 2010 at 11:32 PM

This is totally unsupported! One day a small upgrade will break this hack, and you’ll be in a world of pain.

Method 4: Index Low, Read Data out to a File

Instead of moving the buckets up, use a Splunk search to output a flat file of all _raw data on a set interval. Move the file up for index.



Should you do this?

Maybe!

Method 4: Index Low, Read Data out to a File

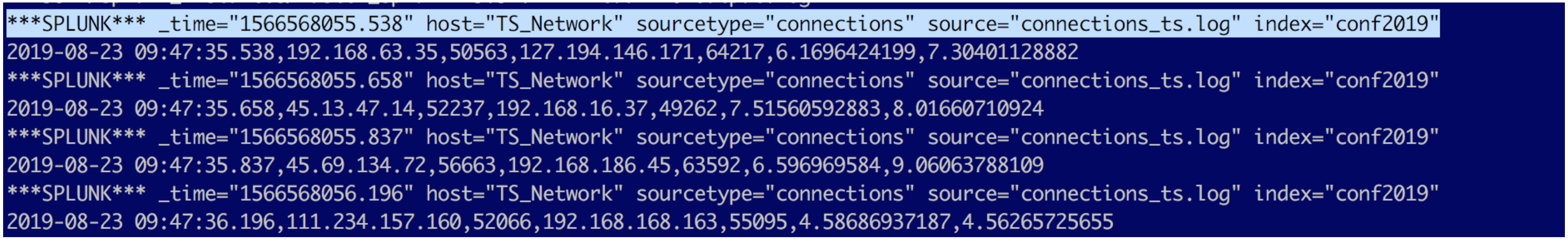
Using an example search like this… (done from CLI, as it’s normally part a cron’d script):

```
$ bin/splunk search 'index=* earliest=-2h@h latest=-1h@h

| eval headeroutput = "***SPLUNK*** _time=\""._time."\" host=\"".host."\"
sourcetype=\"".sourcetype."\" source=\"".source."\" index=\"".index."\"-
splitonthis-"._raw

| table headeroutput

| makemv delim="-splitonthis-" headeroutput

| mvexpand headeroutput' -maxout 0 -header F
```

(Note: You’ll need to tweak earliest/latest to match your cron schedule)

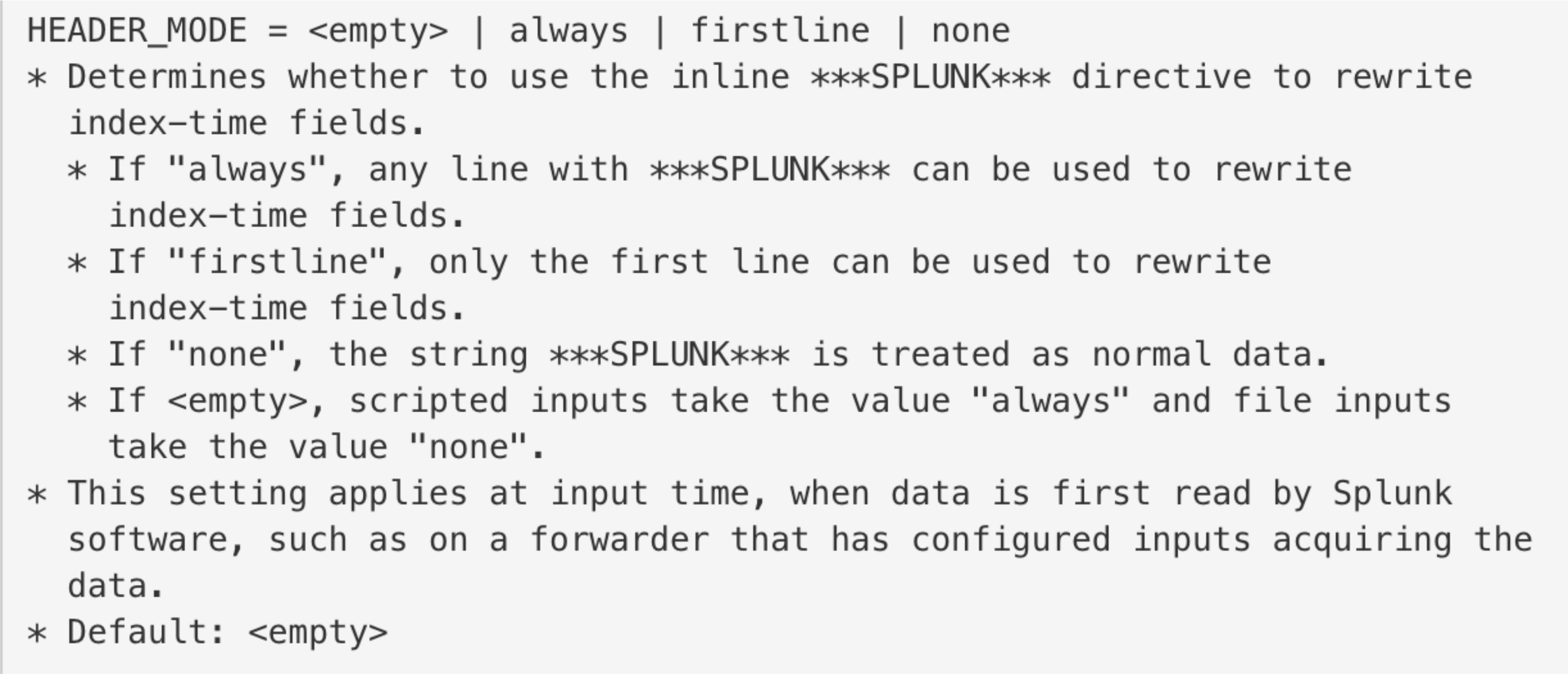
Method 4: Index Low, Read Data out to a File

- Low-side search output looks like:



```
***SPLUNK*** _time="1566568055.538" host="TS_Network" sourcetype="connections" source="connections_ts.log" index="conf2019"
2019-08-23 09:47:35.538,192.168.63.35,50563,127.194.146.171,64217,6.1696424199,7.30401128882
***SPLUNK*** _time="1566568055.658" host="TS_Network" sourcetype="connections" source="connections_ts.log" index="conf2019"
2019-08-23 09:47:35.658,45.13.47.14,52237,192.168.16.37,49262,7.51560592883,8.01660710924
***SPLUNK*** _time="1566568055.837" host="TS_Network" sourcetype="connections" source="connections_ts.log" index="conf2019"
2019-08-23 09:47:35.837,45.69.134.72,56663,192.168.186.45,63592,6.596969584,9.06063788109
***SPLUNK*** _time="1566568056.196" host="TS_Network" sourcetype="connections" source="connections_ts.log" index="conf2019"
2019-08-23 09:47:36.196,111.234.157.160,52066,192.168.168.163,55095,4.58686937187,4.56265725655
```

- On high-side input, use the HEADER_MODE = ALWAYS mode in props.conf (https://docs.splunk.com/Documentation/Splunk/latest/admin/Propsconf#Header_Processor_configuration)



```
HEADER_MODE = <empty> | always | firstline | none
* Determines whether to use the inline ***SPLUNK*** directive to rewrite
  index-time fields.
  * If "always", any line with ***SPLUNK*** can be used to rewrite
    index-time fields.
  * If "firstline", only the first line can be used to rewrite
    index-time fields.
  * If "none", the string ***SPLUNK*** is treated as normal data.
  * If <empty>, scripted inputs take the value "always" and file inputs
    take the value "none".
* This setting applies at input time, when data is first read by Splunk
  software, such as on a forwarder that has configured inputs acquiring the
  data.
* Default: <empty>
```

Method 4: Index Low, Read Data out to a File

Pros:
Moving events eliminates the challenges we discussed previously: cluster agnostic, bucket_ids a non-issue, metadata properly captured

- Your generating search will return the raw event and preserve the host/source/sourcetype/_time in an inserted header row for each event

There's nothing additional to install or maintain, outside the cron script

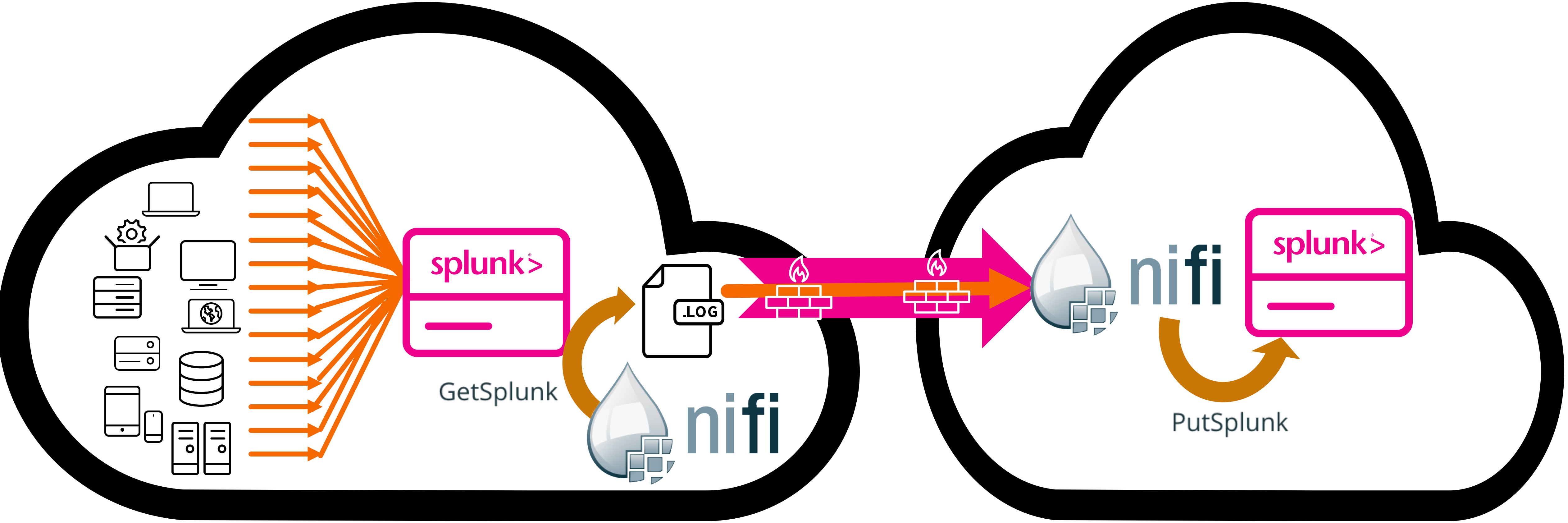
Cons:
This will eventually struggle to keep up at high enough ingest, at which point the low-side Splunk (which lived-reality shows tends to be underpowered) cannot read out the data fast enough over the specified interval

This can be brittle: you need to think about how to handle if/when the low-side server skips a generating search

This works very well at lower volumes if your use cases allow you to tolerate data coming in as a batch on a search schedule

Method 5: Index Low, NiFi out and up, NiFi back in

Instead of using a a Splunk search to output a log file of low side data, use Apache Niagra Files (NiFi).



Should you do this?

If you can!

Method 5: Index Low, NiFi out and up, NiFi back in

Pros:

NiFi is scalable, resilient, and highly-available

Customers are using NiFi to push multiple TB through OWTs each day

Biggest customers have started using NiFi on low, but a HF receiver on high, using HEC to push into their indexers

Cons:

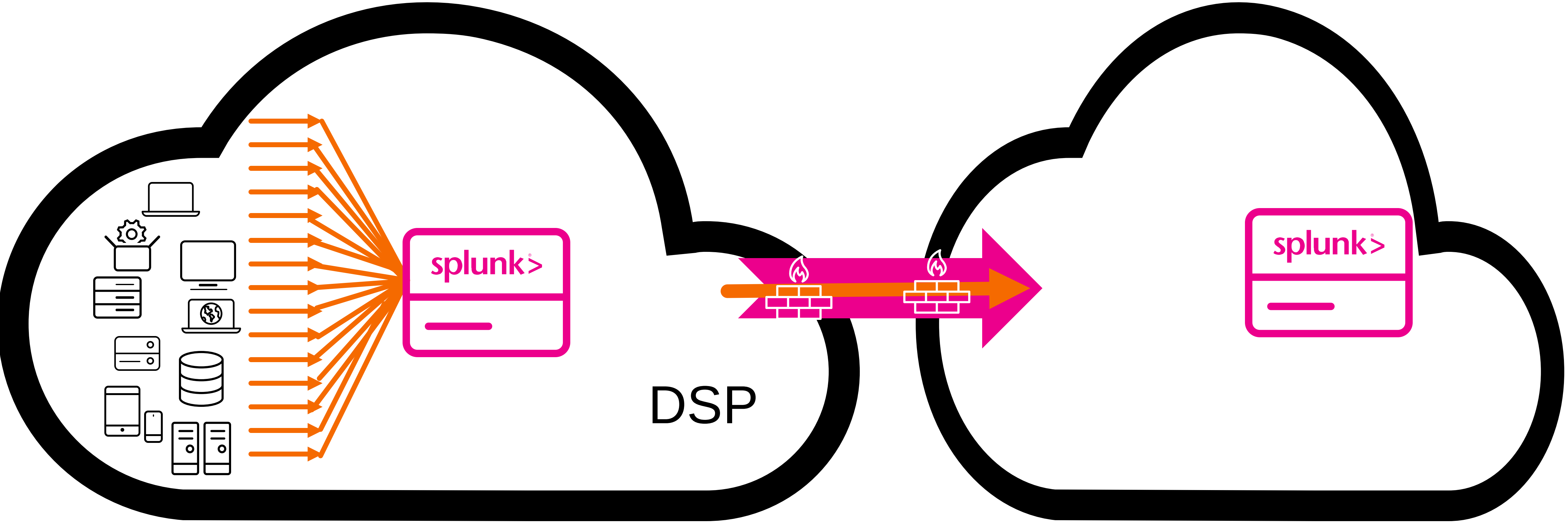
With the added capabilities comes added admin complexity

NiFi is a separate system to learn, install, and maintain

If your team is capable enough to manage a multi-TB deployment of Splunk, it's also capable enough to manage NiFi

Method 6: Data Stream Processor!

Instead of NiFi, use Data Stream Processor to write to a file



Should you do this?

It’s the holy grail.

Method 6: Data Stream Processor

Pros:
Splunk built and supported: one system to rule them all

Resilient, fast, highly available stream processing engine

WYSIWYG pipelines for real-time data manipulation

Heterogenous ingress/egress to non-Splunk endpoints

Cons:
Just became generally available this week

While it's obviously the future, most customers are only now getting started and establishing best practices

Version released this week can't write directly to a file – send data to Kafka and pipe to file there – but that functionality is coming soon

Key Takeaways

No one ever said it would be this hard

1. Move events instead of raw files or buckets
2. The most mature customers use NiFi today
3. DSP is the future

Offline Resources

splunk> .conf19

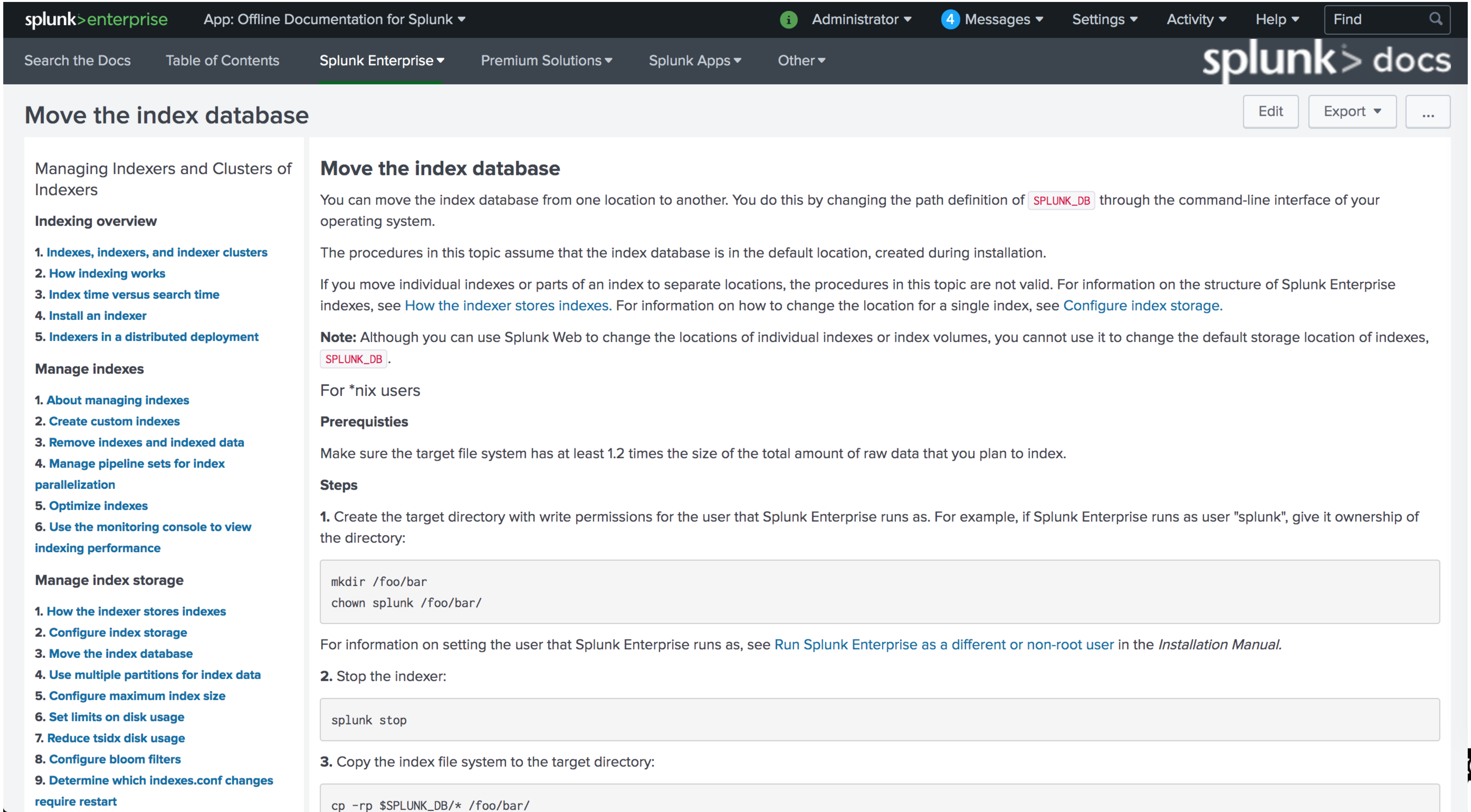
Offline Documention

Splunk Documentation is fantastic, but the offline experience leaves something to be desired

There’s got to be a better way!

DEMO

Offline Documentation for Splunk App



Move the index database

Managing Indexers and Clusters of Indexers

Move the index database

You can move the index database from one location to another. You do this by changing the path definition of `SPLUNK_DB` through the command-line interface of your operating system.

The procedures in this topic assume that the index database is in the default location, created during installation.

If you move individual indexes or parts of an index to separate locations, the procedures in this topic are not valid. For information on the structure of Splunk Enterprise indexes, see How the indexer stores indexes. For information on how to change the location for a single index, see Configure index storage.

Note: Although you can use Splunk Web to change the locations of individual indexes or index volumes, you cannot use it to change the default storage location of indexes, `SPLUNK_DB`.

For *nix users

Prerequisties

Make sure the target file system has at least 1.2 times the size of the total amount of raw data that you plan to index.

Steps

1. Create the target directory with write permissions for the user that Splunk Enterprise runs as. For example, if Splunk Enterprise runs as user "splunk", give it ownership of the directory:

```
mkdir /foo/bar
chown splunk /foo/bar/
```

For information on setting the user that Splunk Enterprise runs as, see Run Splunk Enterprise as a different or non-root user in the *Installation Manual.*

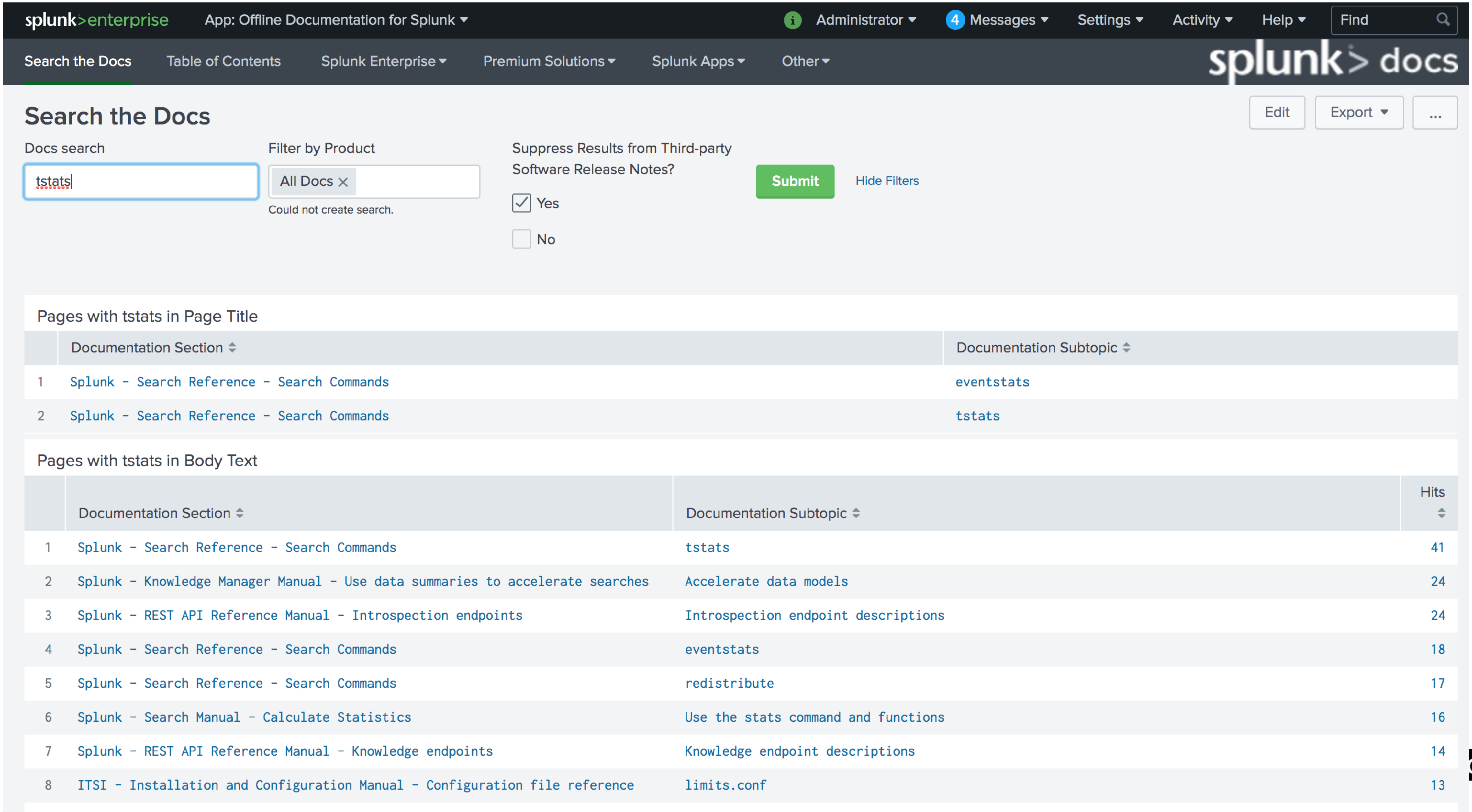
2. Stop the indexer:

```
splunk stop
```

3. Copy the index file system to the target directory:

```
cp -rp $SPLUNK_DB/* /foo/bar/
```

Offline Documentation for Splunk App



Search the Docs

Docs search: tstats

Filter by Product: All Docs ✕

Could not create search.

Suppress Results from Third-party Software Release Notes?

☑ Yes

□ No

Pages with tstats in Page Title

	Documentation Section ⇕	Documentation Subtopic ⇕
1	Splunk - Search Reference - Search Commands	eventstats
2	Splunk - Search Reference - Search Commands	tstats

Pages with tstats in Body Text

	Documentation Section ⇕	Documentation Subtopic ⇕	Hits ⇕
1	Splunk - Search Reference - Search Commands	tstats	41
2	Splunk - Knowledge Manager Manual - Use data summaries to accelerate searches	Accelerate data models	24
3	Splunk - REST API Reference Manual - Introspection endpoints	Introspection endpoint descriptions	24
4	Splunk - Search Reference - Search Commands	eventstats	18
5	Splunk - Search Reference - Search Commands	redistribute	17
6	Splunk - Search Manual - Calculate Statistics	Use the stats command and functions	16
7	Splunk - REST API Reference Manual - Knowledge endpoints	Knowledge endpoint descriptions	14
8	ITSI - Installation and Configuration Manual - Configuration file reference	limits.conf	13

Offline Documentation for Splunk App

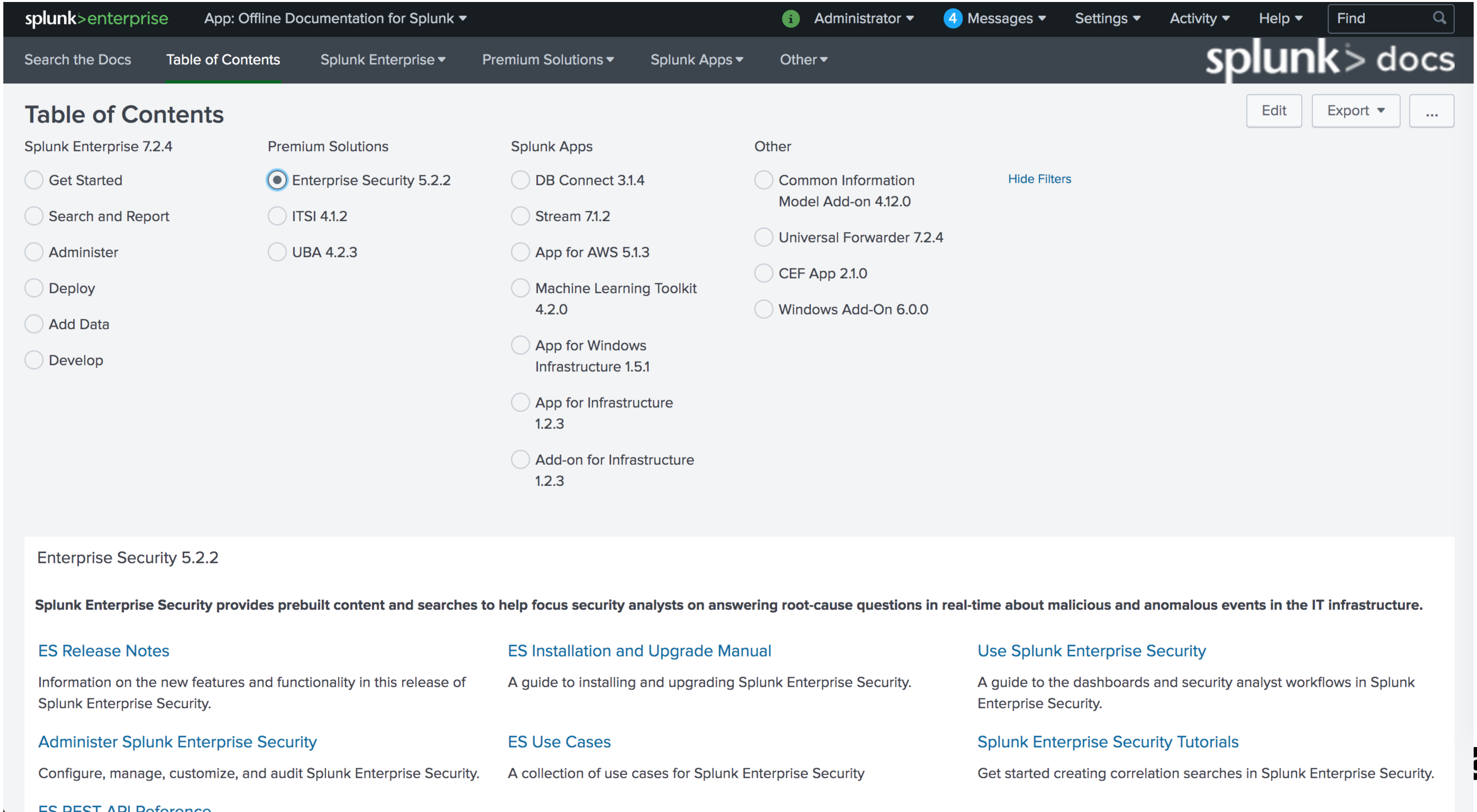


Table of Contents

Splunk Enterprise 7.2.4

- ○ Get Started
- ○ Search and Report
- ○ Administer
- ○ Deploy
- ○ Add Data
- ○ Develop

Premium Solutions

- ● Enterprise Security 5.2.2
- ○ ITSI 4.1.2
- ○ UBA 4.2.3

Splunk Apps

- ○ DB Connect 3.1.4
- ○ Stream 7.1.2
- ○ App for AWS 5.1.3
- ○ Machine Learning Toolkit 4.2.0
- ○ App for Windows Infrastructure 1.5.1
- ○ App for Infrastructure 1.2.3
- ○ Add-on for Infrastructure 1.2.3

Other

- ○ Common Information Model Add-on 4.12.0
- ○ Universal Forwarder 7.2.4
- ○ CEF App 2.1.0
- ○ Windows Add-On 6.0.0

Enterprise Security 5.2.2

Splunk Enterprise Security provides prebuilt content and searches to help focus security analysts on answering root-cause questions in real-time about malicious and anomalous events in the IT infrastructure.

ES Release Notes

Information on the new features and functionality in this release of Splunk Enterprise Security.

ES Installation and Upgrade Manual

A guide to installing and upgrading Splunk Enterprise Security.

Use Splunk Enterprise Security

A guide to the dashboards and security analyst workflows in Splunk Enterprise Security.

Administer Splunk Enterprise Security

Configure, manage, customize, and audit Splunk Enterprise Security.

ES Use Cases

A collection of use cases for Splunk Enterprise Security

Splunk Enterprise Security Tutorials

Get started creating correlation searches in Splunk Enterprise Security.

ES REST API Reference

Offline Documentation

Not a Splunk product

Cannot guarantee maintenance schedule

Built by me to help out a few of my customers.

Hosted at http://docsapp.splunk-nsp.com

I've tried to stay on top of it, updating it after the first maintenance release after every major release (so I'll update it for 8.0.1)

Q&A

Steve Schohn | Staff Sales Engineer

splunk> .conf19

Thank You!

Go to the .conf19 mobile app to

RATE THIS SESSION