

Administrators Anonymous

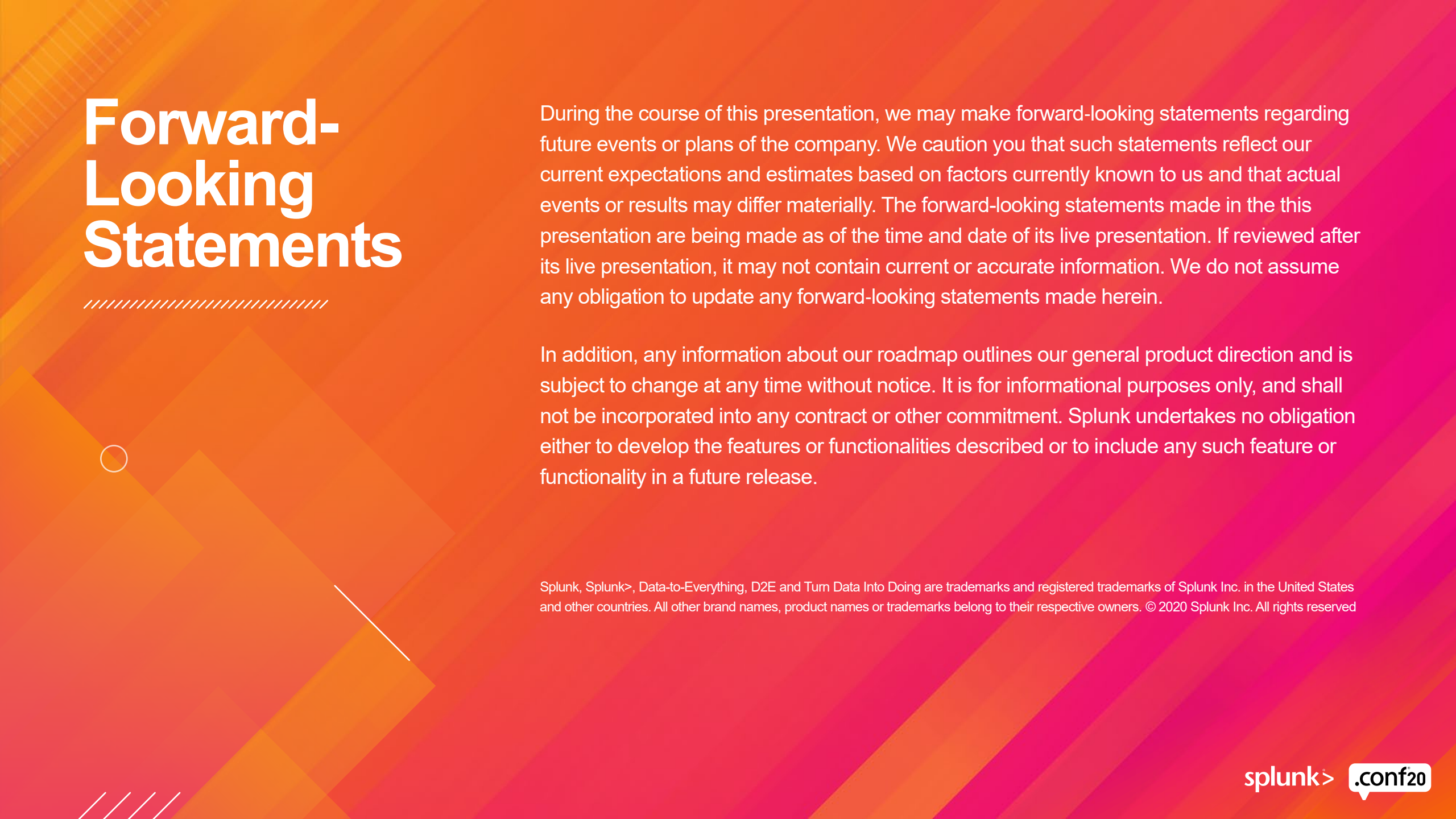
Splunk best practices (and useful tricks)
I learned the hard way

Tom Kopchak

Director of Technical Operations | Hurricane Labs



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>, Data-to-Everything, D2E and Turn Data Into Doing 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. © 2020 Splunk Inc. All rights reserved

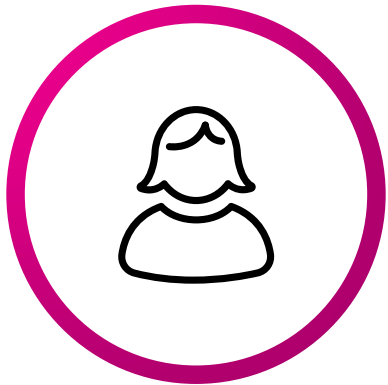
Tom Kopchak

Director of Technical Operations | Hurricane Labs



Who are You?

**Splunk
Administrators**



**Splunk
Architects**



**Splunk
Consultants**



**Recovering
Consultants**



Agenda

What we'll cover

1) Splunk Configuration & Configuration Management

There's a lot of gotchas and potentially bad advice out there

2) Splunk Architecture

Building a solid environment

3) Data Ingestion and Quality

Making your data the best it can be



Configuration & Management

Customize Splunk without breaking it



Configuration Precedence

Easy, but (very) important

Default vs. Local

- Unless you're writing an app, never modify anything in default
- Unless you're writing Splunk, never modify anything in etc/system/default
 - If you're reading this, you're probably not writing Splunk

Configuration Precedence

Easy, but (very) important

Default vs. Local

- Unless you're writing an app, never modify anything in default
- Unless you're writing Splunk, never modify anything in etc/system/default
 - If you're reading this, you're probably not writing Splunk

Highest level (generally): etc/system/local (ESL)

- This will override any other lower-level configuration setting!
- Except when it isn't: slave-apps on an indexer cluster takes precedence

Configuration Precedence

Easy, but (very) important

Default vs. Local

- Unless you're writing an app, never modify anything in default
- Unless you're writing Splunk, never modify anything in etc/system/default
 - If you're reading this, you're probably not writing Splunk

Highest level (generally): etc/system/local (ESL)

- This will override any other lower-level configuration setting!
- Except when it isn't: slave-apps on an indexer cluster takes precedence

Best Practice: Avoid making changes in etc/system/local

How Does etc/system/local Happen?

- Some configurations in the WebUI
- Using the `splunk edit` command
- Using the Splunk MSI installer
- Advice from blogs, Splunk Answers, Docs, Support

How Does etc/system/local Happen?

- Some configurations in the WebUI
- Using the `splunk edit` command
- Using the Splunk MSI installer
- Advice from blogs, Splunk Answers, Docs, Support

Best Practice: Use apps for all configurations

This is more difficult and requires planning, but gives better results

hurricane@splunk-demo: /opt/splunk/bin (ssh)

```
hurricane@splunk-demo:/opt/splunk/bin$ ./splunk edit cluster-config -mode master -secret donttellanyone
```

The cluster-config property has been edited.

You need to restart the Splunk Server (splunkd) for your changes to take effect.

```
hurricane@splunk-demo:/opt/splunk/bin$ cat /opt/splunk/etc/system/local/server.conf | grep master
```

```
mode = master
```

```
hurricane@splunk-demo:/opt/splunk/bin$
```




Download topic as PDF

10 minutes to read

Configure forwarding with outputs.conf

The [outputs.conf](#) file defines how forwarders send data to receivers. You can specify some output configurations at installation time (Windows universal forwarders only) or the CLI, but most advanced configuration settings require that you edit `outputs.conf`.

The topics that describe various forwarding topologies, such as [load balancing](#) and [intermediate forwarding](#), provide detailed examples on configuring `outputs.conf` to support those topologies.

Although `outputs.conf` is a required file for configuring forwarders, it addresses only the outputs from the forwarder, where you want the forwarder to send the data it collects. To specify the data that you want to collect from the forwarder, you must separately configure the inputs, as you would for any Splunk instance. See [Add data and configure inputs](#) in *Getting Data In*.

Edit outputs.conf to configure forwarding

This procedure details the steps you must take to edit the default `outputs.conf` which is in `$SPLUNK_HOME/etc/system/local`. You might have to edit the file in other places, as sections in this topic explain. For an example of what an `outputs.conf` file looks like, see [Examples of outputs.conf](#) later in this topic.

1. On the host that forwards that data that you want to collect, open a shell or command prompt or PowerShell window.
2. Go to the configuration directory for the forwarder.

Unix

Windows

Configure forwarding with outputs.conf

[Edit outputs.conf to configure forwarding](#)[Types of outputs.conf forwarding topologies](#)[Configuration levels for outputs.conf](#)[Define typical forwarding deployment topologies](#)[Common attributes for outputs.conf](#)

Why is This a Problem?

Glad you asked!

Universal forwarders

- Deployment server connection: deploymentclient.conf
- Inputs configuration: inputs.conf
- Outputs configuration: outputs.conf

Why is This a Problem?

Glad you asked!

Universal forwarders

- Deployment server connection: deploymentclient.conf
- Inputs configuration: inputs.conf
- Outputs configuration: outputs.conf

How do you handle if your deployment server changes?

- You should probably be using a DNS CNAME anyway
- Manually updating every UF isn't practical
- This can be fixed with a scripted input, but that's messy

Why is This a Problem?

Glad you asked!

Universal forwarders

- Deployment server connection: deploymentclient.conf
- Inputs configuration: inputs.conf
- Outputs configuration: outputs.conf

How do you handle if your deployment server changes?

- You should probably be using a DNS CNAME anyway
- Manually updating every UF isn't practical
- This can be fixed with a scripted input, but that's messy

Key Takeaway: Anything in etc/system/local is NOT managed (and cannot be easily changed) by the deployment server

Deleting default app configuration

Because something in an app's default configuration sometimes needs to go away

Don't ever modify ANYTHING in default

Create a blank entry in the appropriate stanza in local

```
root@hdf-cptc-02: ~ (ssh)
splunk@hdf-cptc-02:/opt/splunk/etc/apps/StupidApp$ /opt/splunk/bin/splunk btool props list mySourcetype --debug | grep StupidApp
/opt/splunk/etc/apps/StupidApp/default/props.conf [mySourcetype]
/opt/splunk/etc/apps/StupidApp/default/props.conf EVAL-action = true
/opt/splunk/etc/apps/StupidApp/default/props.conf FIELDALIAS-user = person AS user
splunk@hdf-cptc-02:/opt/splunk/etc/apps/StupidApp$
```

```
root@hdf-cptc-02: ~ (ssh)
splunk@hdf-cptc-02:/opt/splunk/etc/apps/StupidApp$ cat local/props.conf
[mySourcetype]
EVAL-action =
splunk@hdf-cptc-02:/opt/splunk/etc/apps/StupidApp$ /opt/splunk/bin/splunk btool props list mySourcetype --debug | grep StupidApp
/opt/splunk/etc/apps/StupidApp/local/props.conf [mySourcetype]
/opt/splunk/etc/apps/StupidApp/local/props.conf EVAL-action =
/opt/splunk/etc/apps/StupidApp/default/props.conf FIELDALIAS-user = person AS user
splunk@hdf-cptc-02:/opt/splunk/etc/apps/StupidApp$
```

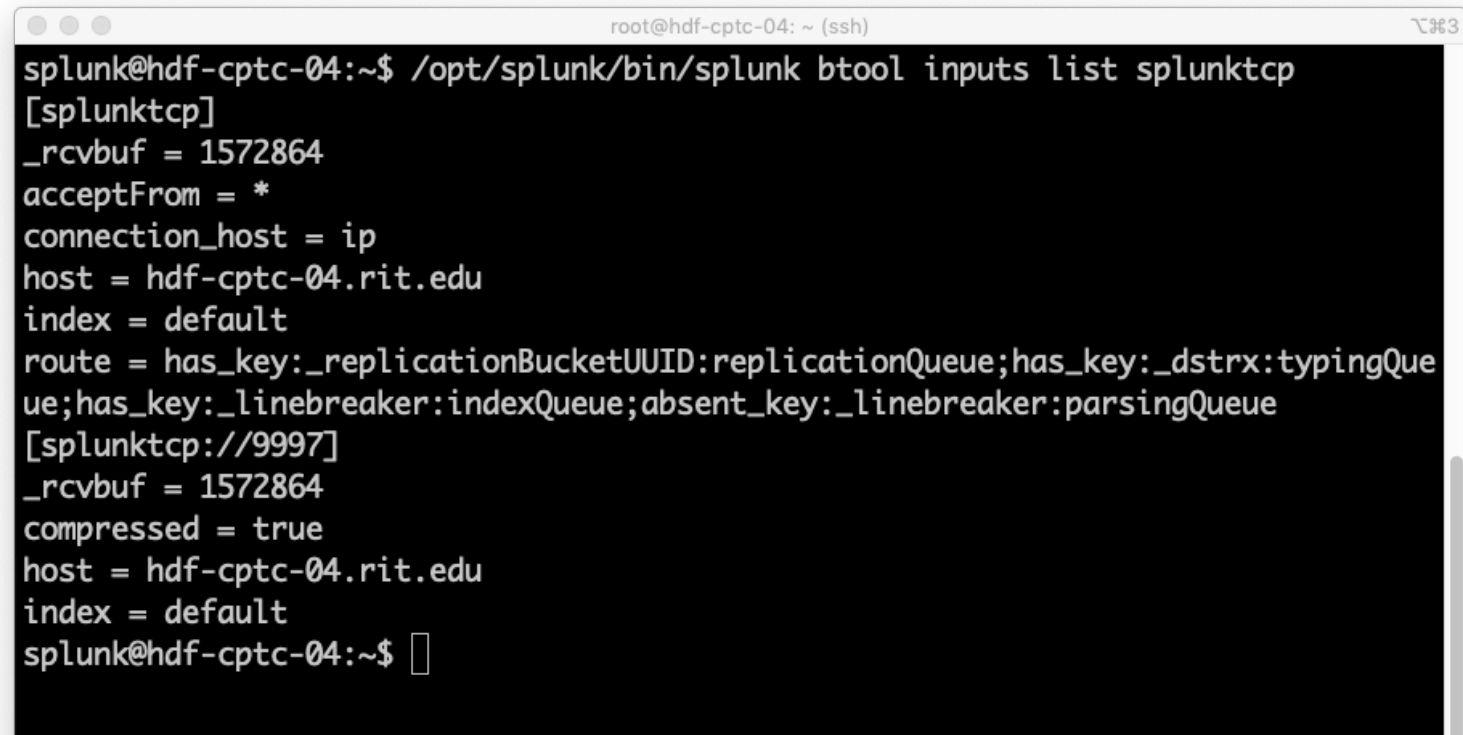
Understanding Configurations with btool

Btool – best way to understand what configuration exists on disk

Understand the limitations

- Not the currently running configuration

Many different flags/options based on what you're trying to do

A terminal window with a black background and white text. The window title is 'root@hdf-cptc-04: ~ (ssh)'. The command executed is 'splunk@hdf-cptc-04:~\$ /opt/splunk/bin/splunk btool inputs list splunktcp'. The output shows two configuration blocks for 'splunktcp'. The first block has settings: '_rcvbuf = 1572864', 'acceptFrom = *', 'connection_host = ip', 'host = hdf-cptc-04.rit.edu', 'index = default', and a long 'route' value. The second block has settings: '_rcvbuf = 1572864', 'compressed = true', 'host = hdf-cptc-04.rit.edu', and 'index = default'. The prompt 'splunk@hdf-cptc-04:~\$' is followed by a cursor.

```
root@hdf-cptc-04: ~ (ssh)
splunk@hdf-cptc-04:~$ /opt/splunk/bin/splunk btool inputs list splunktcp
[splunktcp]
_rcvbuf = 1572864
acceptFrom = *
connection_host = ip
host = hdf-cptc-04.rit.edu
index = default
route = has_key:_replicationBucketUUID:replicationQueue;has_key:_dstrx:typingQueue;has_key:_linebreaker:indexQueue;absent_key:_linebreaker:parsingQueue
[splunktcp://9997]
_rcvbuf = 1572864
compressed = true
host = hdf-cptc-04.rit.edu
index = default
splunk@hdf-cptc-04:~$
```


Btool Example: base config

```
root@hdf-cptc-04: ~ (ssh)
splunk@hdf-cptc-04:~$ /opt/splunk/bin/splunk btool inputs list splunktcp --debug
/opt/splunk/etc/system/default/inputs.conf [splunktcp]
/opt/splunk/etc/system/default/inputs.conf _rcvbuf = 1572864
/opt/splunk/etc/system/default/inputs.conf acceptFrom = *
/opt/splunk/etc/system/default/inputs.conf connection_host = ip
/opt/splunk/etc/system/local/inputs.conf host = hdf-cptc-04.rit.edu
/opt/splunk/etc/system/default/inputs.conf index = default
/opt/splunk/etc/system/default/inputs.conf route = has_key:_replicationBucketUUID:replicationQueue;has_key:_dstrx:typingQueue;has_key:_linebreaker:indexQueue;absent_key:_linebreaker:parsingQueue
/opt/splunk/etc/slave-apps/infra_inputs/local/inputs.conf [splunktcp://9997]
/opt/splunk/etc/system/default/inputs.conf _rcvbuf = 1572864
/opt/splunk/etc/slave-apps/infra_inputs/local/inputs.conf compressed = true
/opt/splunk/etc/system/local/inputs.conf host = hdf-cptc-04.rit.edu
/opt/splunk/etc/system/default/inputs.conf index = default
splunk@hdf-cptc-04:~$
```

```
root@hdf-cptc-04: ~ (ssh)
splunk@hdf-cptc-04:~$ /opt/splunk/bin/splunk btool inputs list splunktcp --debug | grep -v system/default
/opt/splunk/etc/system/local/inputs.conf host = hdf-cptc-04.rit.edu
/opt/splunk/etc/slave-apps/infra_inputs/local/inputs.conf [splunktcp://9997]
/opt/splunk/etc/slave-apps/infra_inputs/local/inputs.conf compressed = true
/opt/splunk/etc/system/local/inputs.conf host = hdf-cptc-04.rit.edu
splunk@hdf-cptc-04:~$
```

Btool Example: base config

```
root@hdf-cptc-04: ~ (ssh)
splunk@hdf-cptc-04:~$ /opt/splunk/bin/splunk btool inputs list splunktcp --debug
/opt/splunk/etc/system/default/inputs.conf      [splunktcp]
/opt/splunk/etc/system/default/inputs.conf      _rcvbuf = 1572864
/opt/splunk/etc/system/default/inputs.conf      acceptFrom = *
/opt/splunk/etc/system/default/inputs.conf      connection_host = ip
/opt/splunk/etc/system/local/inputs.conf         host = hdf-cptc-04.rit.edu
/opt/splunk/etc/system/default/inputs.conf      index = default
/opt/splunk/etc/system/default/inputs.conf      route = has_key:_replicationBucketUUID:replicationQueue
;has_key:_dstrx:typingQueue;has_key:_linebreaker:indexQueue;absent_key:_linebreaker:parsingQueue
/opt/splunk/etc/slave-apps/infra_inputs/local/inputs.conf [splunktcp://9997]
/opt/splunk/etc/system/default/inputs.conf      _rcvbuf = 1572864
/opt/splunk/etc/slave-apps/infra_inputs/local/inputs.conf compressed = true
/opt/splunk/etc/system/local/inputs.conf         host = hdf-cptc-04.rit.edu
/opt/splunk/etc/system/default/inputs.conf      index = default
splunk@hdf-cptc-04:~$
```



```
root@hdf-cptc-04: ~ (ssh)
splunk@hdf-cptc-04:~$ /opt/splunk/bin/splunk btool inputs list splunktcp --debug | grep -v system/default
/opt/splunk/etc/system/local/inputs.conf         host = hdf-cptc-04.rit.edu
/opt/splunk/etc/slave-apps/infra_inputs/local/inputs.conf [splunktcp://9997]
/opt/splunk/etc/slave-apps/infra_inputs/local/inputs.conf compressed = true
/opt/splunk/etc/system/local/inputs.conf         host = hdf-cptc-04.rit.edu
splunk@hdf-cptc-04:~$
```

Btool Example: base config

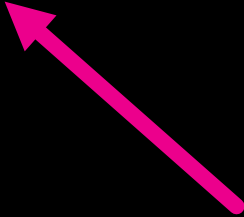
```
root@hdf-cptc-04: ~ (ssh)
splunk@hdf-cptc-04:~$ /opt/splunk/bin/splunk btool inputs list splunktcp --debug
/opt/splunk/etc/system/default/inputs.conf [splunktcp]
/opt/splunk/etc/system/default/inputs.conf _rcvbuf = 1572864
/opt/splunk/etc/system/default/inputs.conf acceptFrom = *
/opt/splunk/etc/system/default/inputs.conf connection_host = ip
/opt/splunk/etc/system/local/inputs.conf host = hdf-cptc-04.rit.edu
/opt/splunk/etc/system/default/inputs.conf index = default
/opt/splunk/etc/system/default/inputs.conf route = has_key:_replicationBucketUUID:replicationQueue;has_key:_dstrx:typingQueue;has_key:_linebreaker:indexQueue;absent_key:_linebreaker:parsingQueue
/opt/splunk/etc/slave-apps/infra_inputs/local/inputs.conf [splunktcp://9997]
/opt/splunk/etc/system/default/inputs.conf _rcvbuf = 1572864
/opt/splunk/etc/slave-apps/infra_inputs/local/inputs.conf compressed = true
/opt/splunk/etc/system/local/inputs.conf host = hdf-cptc-04.rit.edu
/opt/splunk/etc/system/default/inputs.conf index = default
splunk@hdf-cptc-04:~$
```

```
root@hdf-cptc-04: ~ (ssh)
splunk@hdf-cptc-04:~$ /opt/splunk/bin/splunk btool inputs list splunktcp --debug | grep -v system/default
/opt/splunk/etc/system/local/inputs.conf host = hdf-cptc-04.rit.edu
/opt/splunk/etc/slave-apps/infra_inputs/local/inputs.conf [splunktcp://9997]
/opt/splunk/etc/slave-apps/infra_inputs/local/inputs.conf compressed = true
/opt/splunk/etc/system/local/inputs.conf host = hdf-cptc-04.rit.edu
splunk@hdf-cptc-04:~$
```

Btool Example: base config

```
root@hdf-cptc-04: ~ (ssh)
splunk@hdf-cptc-04:~$ /opt/splunk/bin/splunk btool inputs list splunktcp --debug
/opt/splunk/etc/system/default/inputs.conf      [splunktcp]
/opt/splunk/etc/system/default/inputs.conf      _rcvbuf = 1572864
/opt/splunk/etc/system/default/inputs.conf      acceptFrom = *
/opt/splunk/etc/system/default/inputs.conf      connection_host = ip
/opt/splunk/etc/system/local/inputs.conf         host = hdf-cptc-04.rit.edu
/opt/splunk/etc/system/default/inputs.conf      index = default
/opt/splunk/etc/system/default/inputs.conf      route = has_key:_replicationBucketUUID:replicationQueue
;has_key:_dstrx:typingQueue;has_key:_linebreaker:indexQueue;absent_key:_linebreaker:parsingQueue
/opt/splunk/etc/slave-apps/infra_inputs/local/inputs.conf [splunktcp://9997]
/opt/splunk/etc/system/default/inputs.conf      _rcvbuf = 1572864
/opt/splunk/etc/slave-apps/infra_inputs/local/inputs.conf compressed = true
/opt/splunk/etc/system/local/inputs.conf         host = hdf-cptc-04.rit.edu
/opt/splunk/etc/system/default/inputs.conf      index = default
splunk@hdf-cptc-04:~$
```

```
root@hdf-cptc-04: ~ (ssh)
splunk@hdf-cptc-04:~$ /opt/splunk/bin/splunk btool inputs list splunktcp --debug | grep -v system/default
/opt/splunk/etc/system/local/inputs.conf         host = hdf-cptc-04.rit.edu
/opt/splunk/etc/slave-apps/infra_inputs/local/inputs.conf [splunktcp://9997]
/opt/splunk/etc/slave-apps/infra_inputs/local/inputs.conf compressed = true
/opt/splunk/etc/system/local/inputs.conf         host = hdf-cptc-04.rit.edu
splunk@hdf-cptc-04:~$
```



Btool Example: user config

Specify the user and app context

```
hurricane@splunk-demo: ~ (ssh)
hurricane@splunk-demo:~$ /opt/splunk/bin/splunk btool --app=search --user=tom savedsearches list --debug | grep -v system/default | grep -i local
/opt/splunk/etc/apps/search/default/savedsearches.conf search = | rest timeout=600 splunk_server=local /servicesNS/-/-/savedsearches add_orphan_field=yes count=0
/opt/splunk/etc/users/tom/search/local/savedsearches.conf [Terrible Search]
/opt/splunk/etc/users/tom/search/local/savedsearches.conf action.email.useNSSubject = 1
/opt/splunk/etc/users/tom/search/local/savedsearches.conf alert.track = 0
/opt/splunk/etc/users/tom/search/local/savedsearches.conf description = A really, really, really bad search
/opt/splunk/etc/users/tom/search/local/savedsearches.conf dispatch.earliest_time = -24h@h
/opt/splunk/etc/users/tom/search/local/savedsearches.conf dispatch.latest_time = now
/opt/splunk/etc/users/tom/search/local/savedsearches.conf display.visualizations.show = 0
/opt/splunk/etc/users/tom/search/local/savedsearches.conf request.ui_dispatch_app = search
/opt/splunk/etc/users/tom/search/local/savedsearches.conf request.ui_dispatch_view = search
/opt/splunk/etc/users/tom/search/local/savedsearches.conf search = index=* OR index=_* latest=now earliest=-10y@y
hurricane@splunk-demo:~$
```

Apps and Splunk Cloud

Making your app deployment experience faster and easier

Become familiar with what's allowed in Splunk Cloud and what isn't

- Field extractions, lookups, and other knowledge objects = self-install, generally no problem
- Indexes or inputs = will need to be removed first
- Custom python code ALWAYS will require manual review and approval
- One word: appinspect

Apps and Splunk Cloud

Making your app deployment experience faster and easier

Become familiar with what's allowed in Splunk Cloud and what isn't

- Field extractions, lookups, and other knowledge objects = self-install, generally no problem
- Indexes or inputs = will need to be removed first
- Custom python code ALWAYS will require manual review and approval
- One word: appinspect

Strip down apps to components needed for systems where they are deployed

- This is applicable to more than just Splunk Cloud
 - Indexers may need props/transforms for index-time operations, but not dashboards or other configurations

Apps and Splunk Cloud

Making your app deployment experience faster and easier

Become familiar with what's allowed in Splunk Cloud and what isn't

- Field extractions, lookups, and other knowledge objects = self-install, generally no problem
- Indexes or inputs = will need to be removed first
- Custom python code ALWAYS will require manual review and approval
- One word: appinspect

Strip down apps to components needed for systems where they are deployed

- This is applicable to more than just Splunk Cloud
 - Indexers may need props/transforms for index-time operations, but not dashboards or other configurations

REST Commands are your friend

Configuration, Naming, and App Sanity

Someone else might need to understand your deployment

Keep configurations and naming consistent

Configuration, Naming, and App Sanity

Someone else might need to understand your deployment

Keep configurations and naming consistent

Define an app naming convention that's logical and stick with it

- Eg, app names that define what configurations are in an app
 - Monitor stanza: `uf_linux_apache_inputs`
 - SplunkTCP input: `infra_inputs`

Configuration, Naming, and App Sanity

Someone else might need to understand your deployment

Keep configurations and naming consistent

Define an app naming convention that's logical and stick with it

- Eg, app names that define what configurations are in an app
 - Monitor stanza: `uf_linux_apache_inputs`
 - SplunkTCP input: `infra_inputs`

Define a server naming convention that's logical and stick with it

- This may be a lost cause at many companies



Splunk Architecture

Just because your license lets you install an unlimited number of Splunk instances, it doesn't mean you should

Splunk Architecture

More servers isn't always better (but sometimes it is)

- Splunk architecture = flexible
- Just because you can do something doesn't mean you should, or that it will perform well
- Some designs are better than others

Key Takeaway: Pay attention to the next few slides 😊

Indexers & Search Heads

The key to a solid Splunk infrastructure

Good indexer performance = good Splunk performance

- CPU, memory, and disk I/O requirements are not suggestions
- Don't use NFS for warm volumes (ever)
- SSDs, please! (especially for SmartStore cache)

Indexers & Search Heads

The key to a solid Splunk infrastructure

Good indexer performance = good Splunk performance

- CPU, memory, and disk I/O requirements are not suggestions
- Don't use NFS for warm volumes (ever)
- SSDs, please! (especially for SmartStore cache)

Appropriate Indexer:Search Head ratio

- Adding more search heads doesn't make searching faster, especially if indexing tier is insufficient
- Recommended ratio varies, but 1 search head for every 7 indexers is reasonable

Indexers & Search Heads

The key to a solid Splunk infrastructure

Good indexer performance = good Splunk performance

- CPU, memory, and disk I/O requirements are not suggestions
- Don't use NFS for warm volumes (ever)
- SSDs, please! (especially for SmartStore cache)

Appropriate Indexer:Search Head ratio

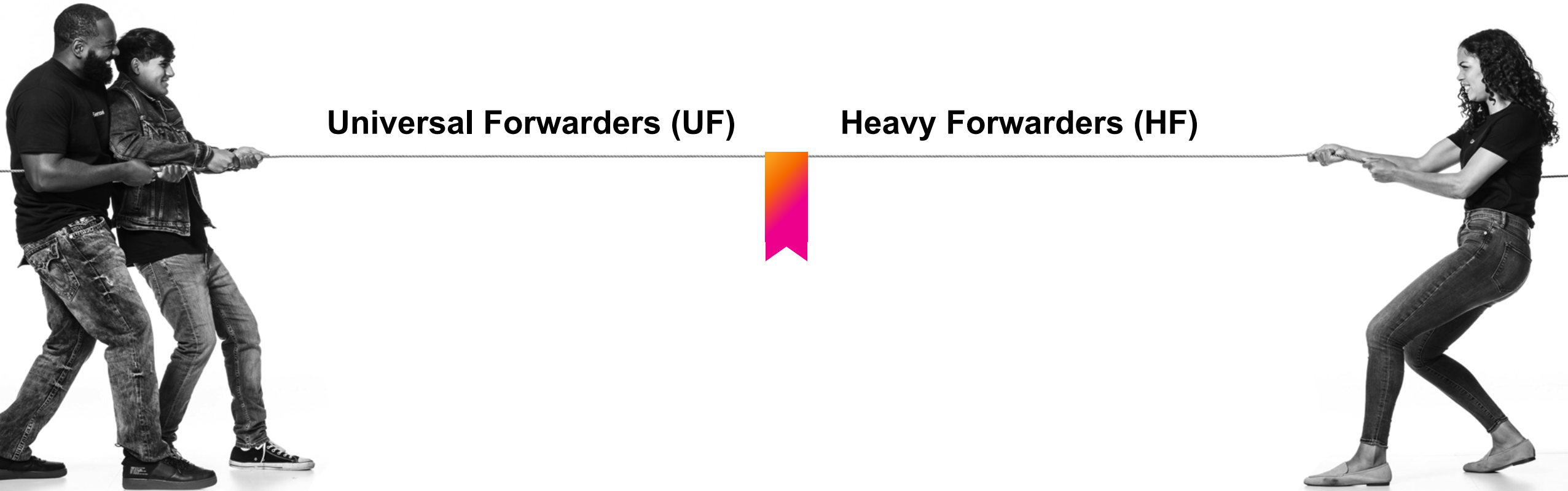
- Adding more search heads doesn't make searching faster, especially if indexing tier is insufficient
- Recommended ratio varies, but 1 search head for every 7 indexers is reasonable

Premium Apps

- Splunk Enterprise Security (ES) and IT Service Intelligence (ITSI) = dedicated search head
- Don't install other apps on these instances (other than TAs)

Universal Forwarders vs. Heavy Forwarders

Key takeaway: Universal forwarders are more efficient



Universal Forwarders vs. Heavy Forwarders

Key takeaway: Universal forwarders are more efficient



Universal Forwarders (UF)

Lightweight Splunk installation



Heavy Forwarders (HF)

Full Splunk Enterprise installation



Universal Forwarders vs. Heavy Forwarders

Key takeaway: Universal forwarders are more efficient



Universal Forwarders (UF)

Lightweight Splunk installation

Does not include Python



Heavy Forwarders (HF)

Full Splunk Enterprise installation

Includes Python



Universal Forwarders vs. Heavy Forwarders

Key takeaway: Universal forwarders are more efficient



Universal Forwarders (UF)

Lightweight Splunk installation

Does not include Python

Able to handle most data ingestion tasks



Heavy Forwarders (HF)

Full Splunk Enterprise installation

Includes Python

Able to run full Splunk apps



Universal Forwarders vs. Heavy Forwarders

Key takeaway: Universal forwarders are more efficient



Universal Forwarders (UF)

Lightweight Splunk installation

Does not include Python

Able to handle most data ingestion tasks

Efficient streaming of data to Splunk



Heavy Forwarders (HF)

Full Splunk Enterprise installation

Includes Python

Able to run full Splunk apps

Actually a full Splunk instance



Overuse of Heavy Forwarders

Often you will see Heavy Forwarders in front of Indexers

- Still see this in the field sometimes
- NOT recommended practice today
- Reference: <https://hlb.ninja/2mKJERu>

Overuse of Heavy Forwarders

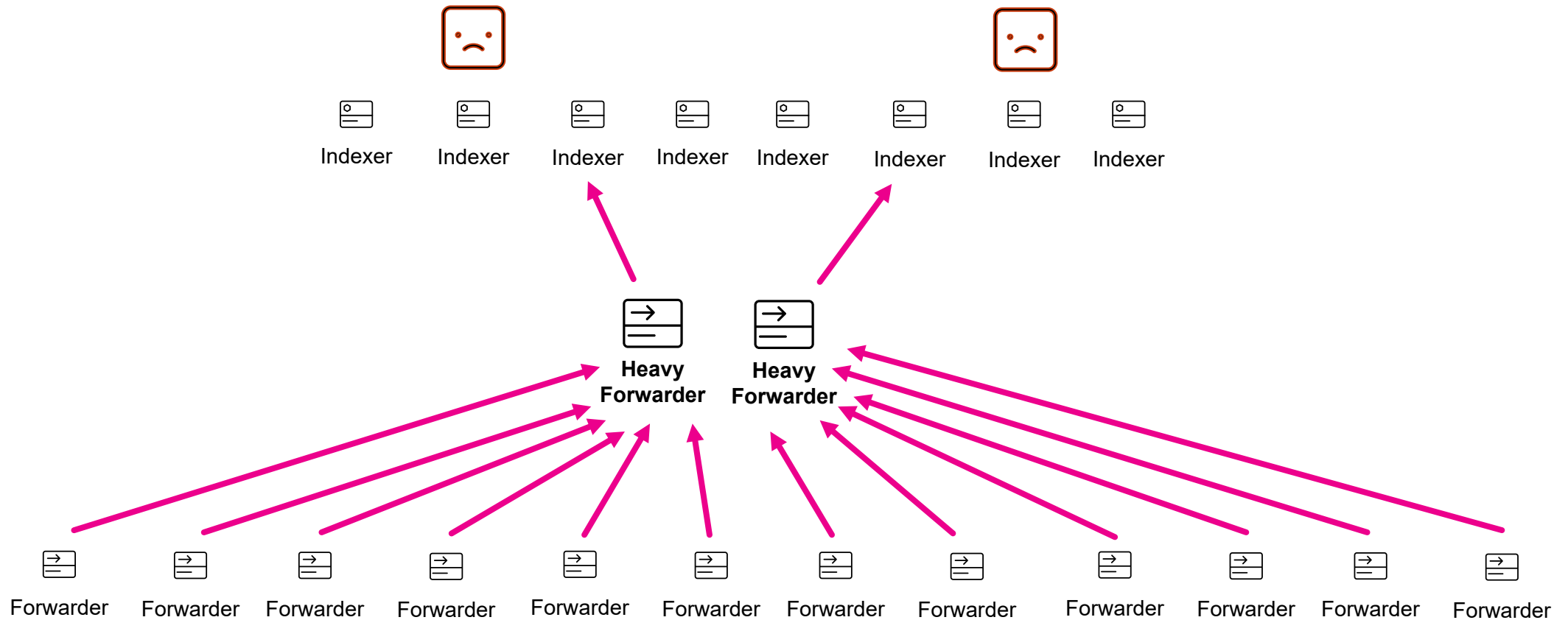
Often you will see Heavy Forwarders in front of Indexers

- Still see this in the field sometimes
- NOT recommended practice today
- Reference: <https://hlb.ninja/2mKJERu>

Best Practice: Have Universal Forwarders send data directly to indexers

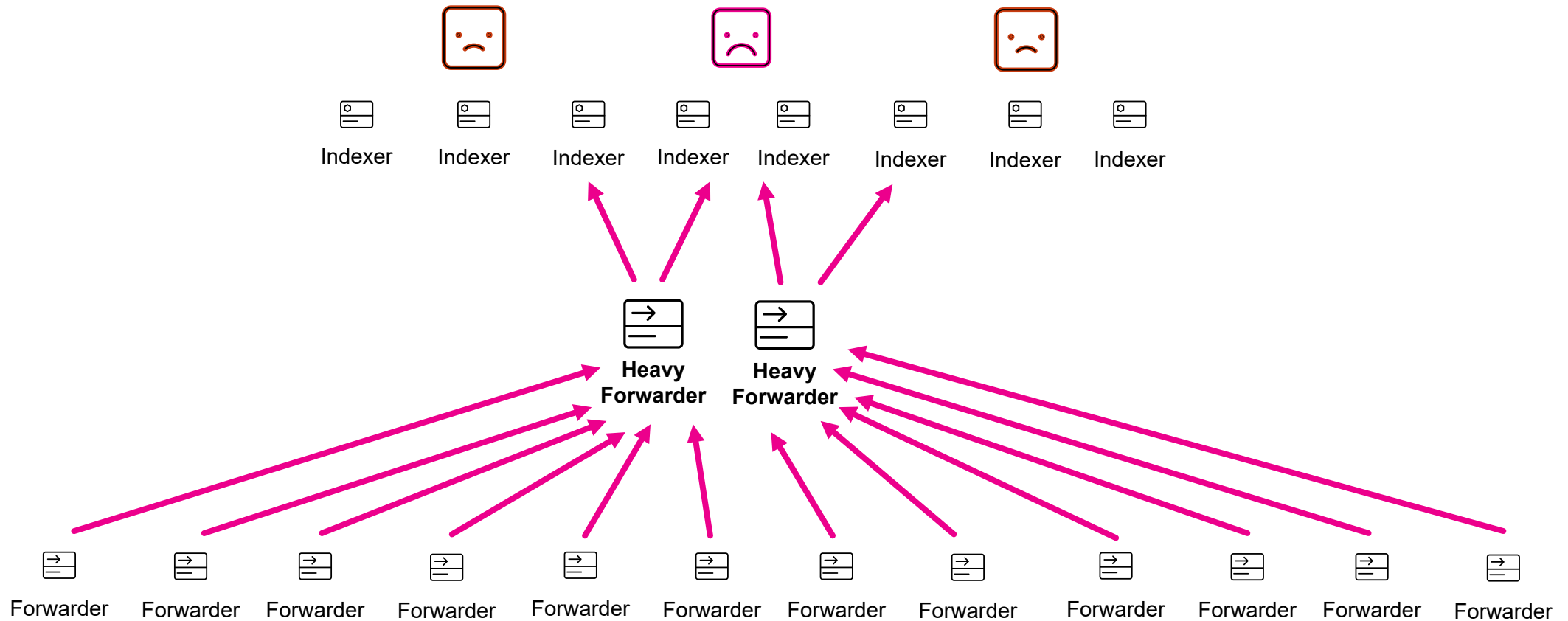
Why are HFs Bad for Data Distribution?

It creates a bottleneck



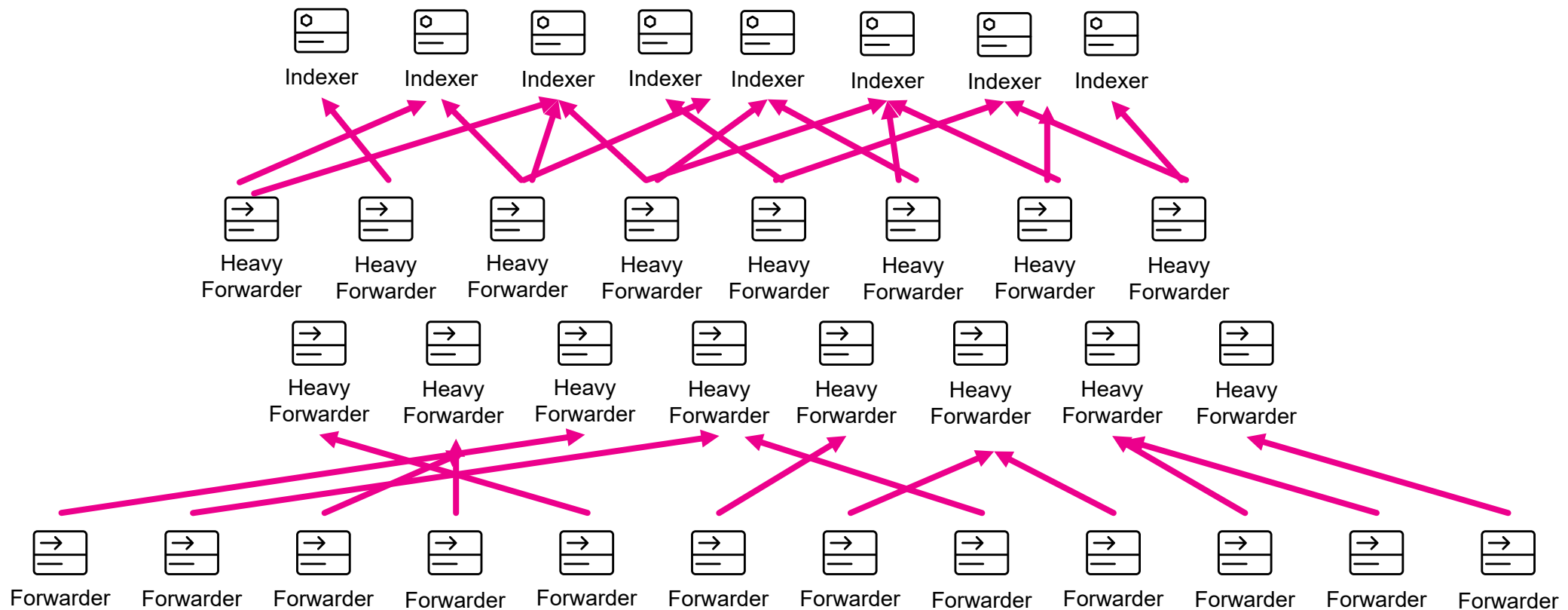
What About More Parallel Ingestion Pipelines?

SO MUCH BETTER!!!



So....You Really Want to Use HFs?

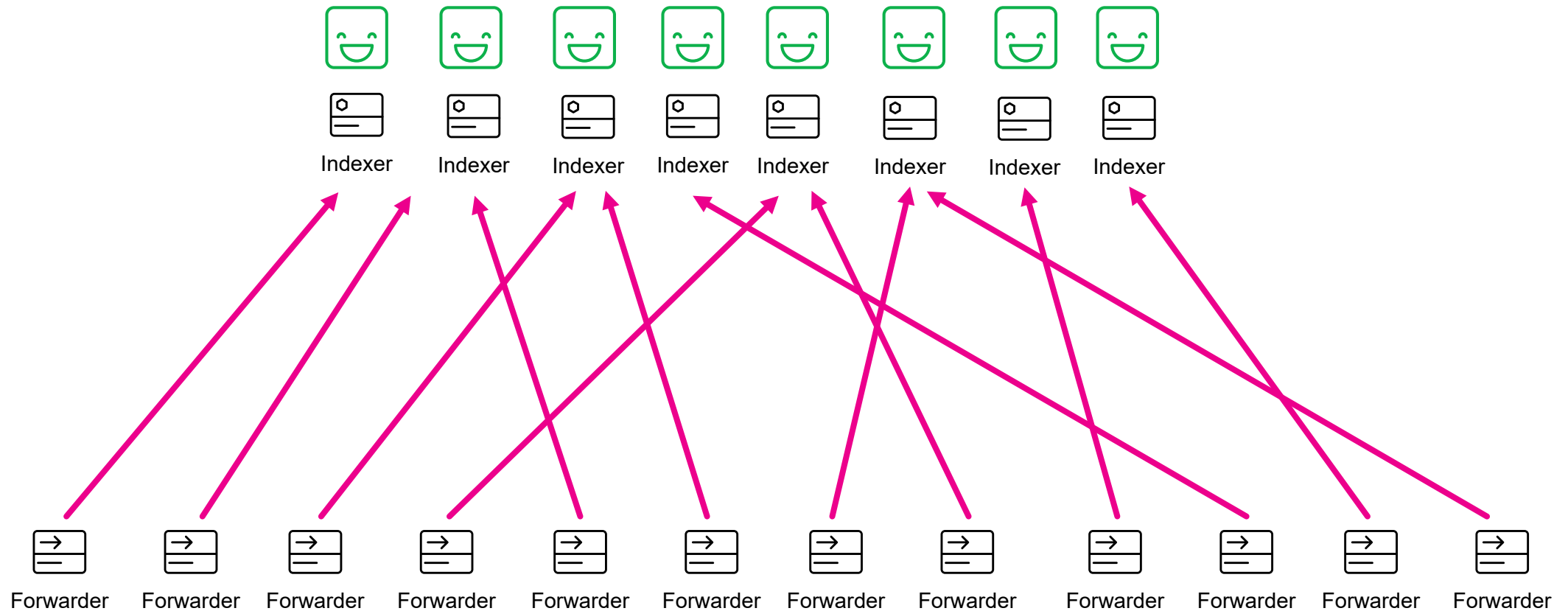
Okay, fine. Deploy a bunch of them. And make them UFs if you can too.



Note: Arrows not to scale

Send Data Directly to the Indexers

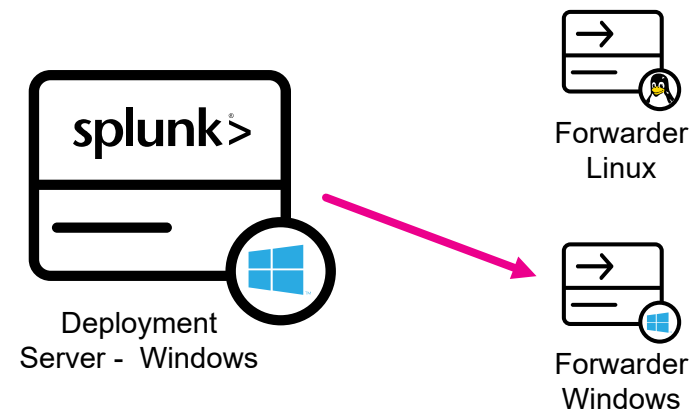
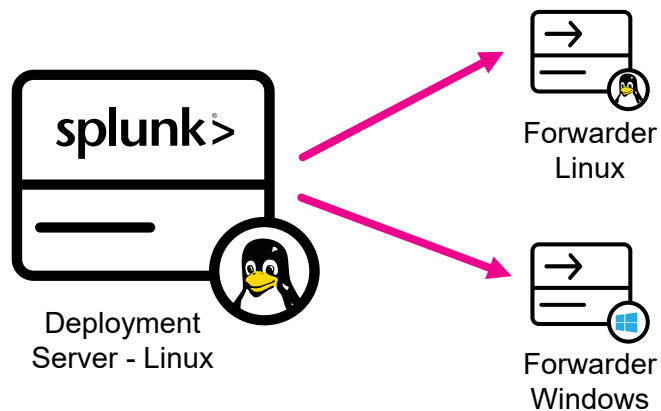
Bottleneck = gone!



Deployment Management

Use the proper Deployment Server (DS) platform for your environment

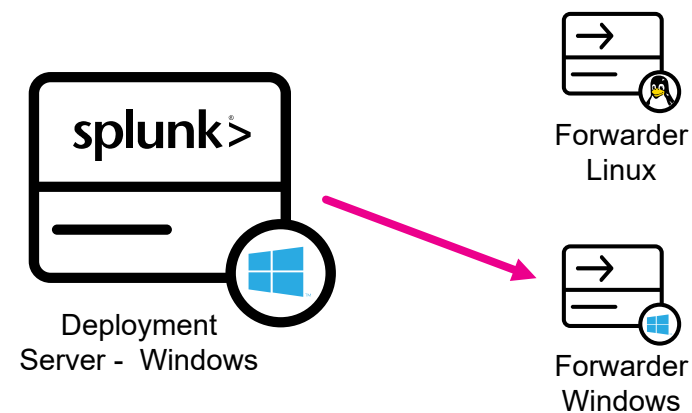
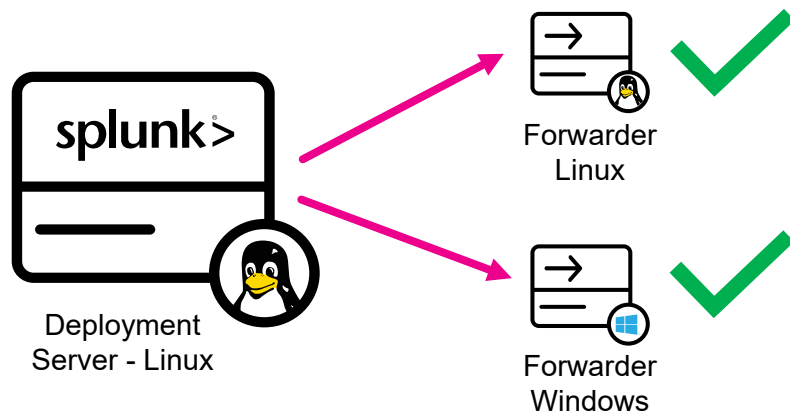
- A Windows DS should *only* manage Windows hosts (Linux is not supported)
- A Linux DS can manage both Windows and Linux hosts
- Reference: <https://h1b.ninja/2nHgltY>



Deployment Management

Use the proper Deployment Server (DS) platform for your environment

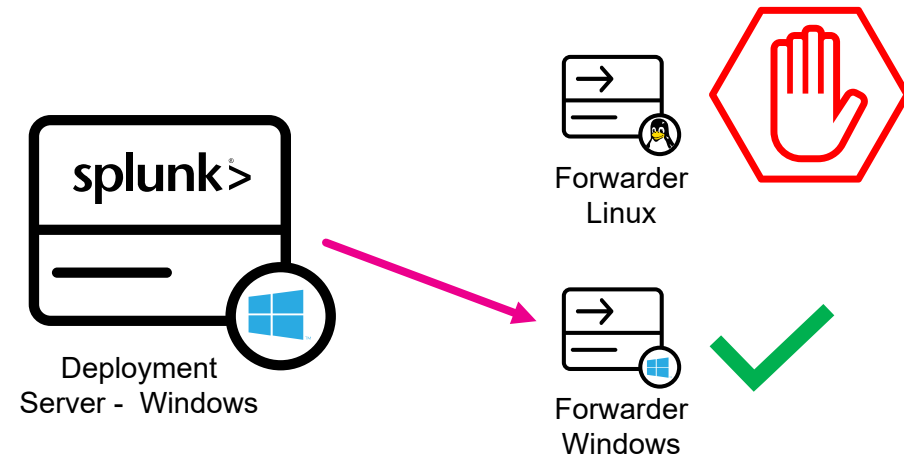
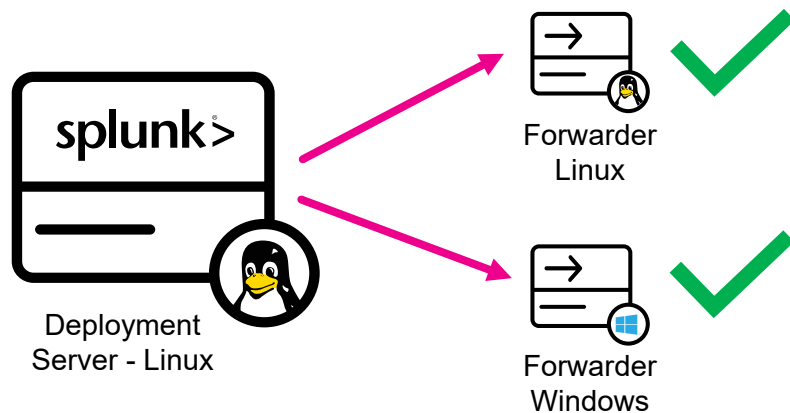
- A Windows DS should *only* manage Windows hosts (Linux is not supported)
- A Linux DS can manage both Windows and Linux hosts
- Reference: <https://h1b.ninja/2nHgltY>



Deployment Management

Use the proper Deployment Server (DS) platform for your environment

- A Windows DS should *only* manage Windows hosts (Linux is not supported)
- A Linux DS can manage both Windows and Linux hosts
- Reference: <https://hlb.ninja/2nHgltY>



Deployment Apps & Server Classes

Deployment apps are building blocks

- Single function apps are preferred
- Mix and match configurations as needed

Deployment Apps & Server Classes

Deployment apps are building blocks

- Single function apps are preferred
- Mix and match configurations as needed

Create server classes based on the task they need to accomplish

- DS creates a tarball per app+serverclass combination
 - 20 serverclasses with the same app = 20 different tarballs
 - Slow DS restarts = symptom of this issue
- Example: Rather than adding outputs apps to every server class, create general server classes to broadly cover inputs

Deployment Apps & Server Classes

Deployment apps are building blocks

- Single function apps are preferred
- Mix and match configurations as needed

Create server classes based on the task they need to accomplish

- DS creates a tarball per app+serverclass combination
 - 20 serverclasses with the same app = 20 different tarballs
 - Slow DS restarts = symptom of this issue
- Example: Rather than adding outputs apps to every server class, create general server classes to broadly cover inputs

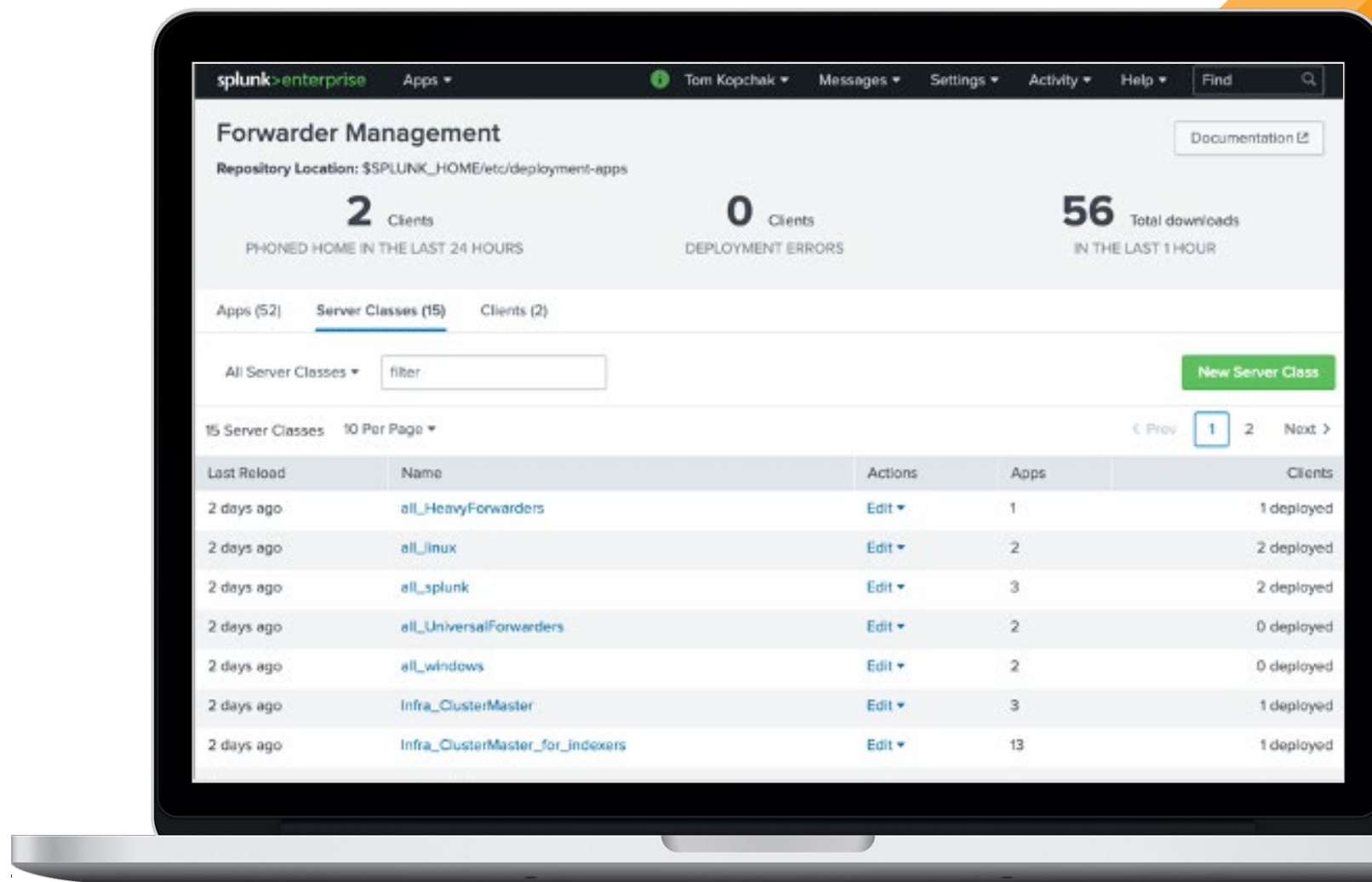
Key Takeaway: Logical names are a major benefit here

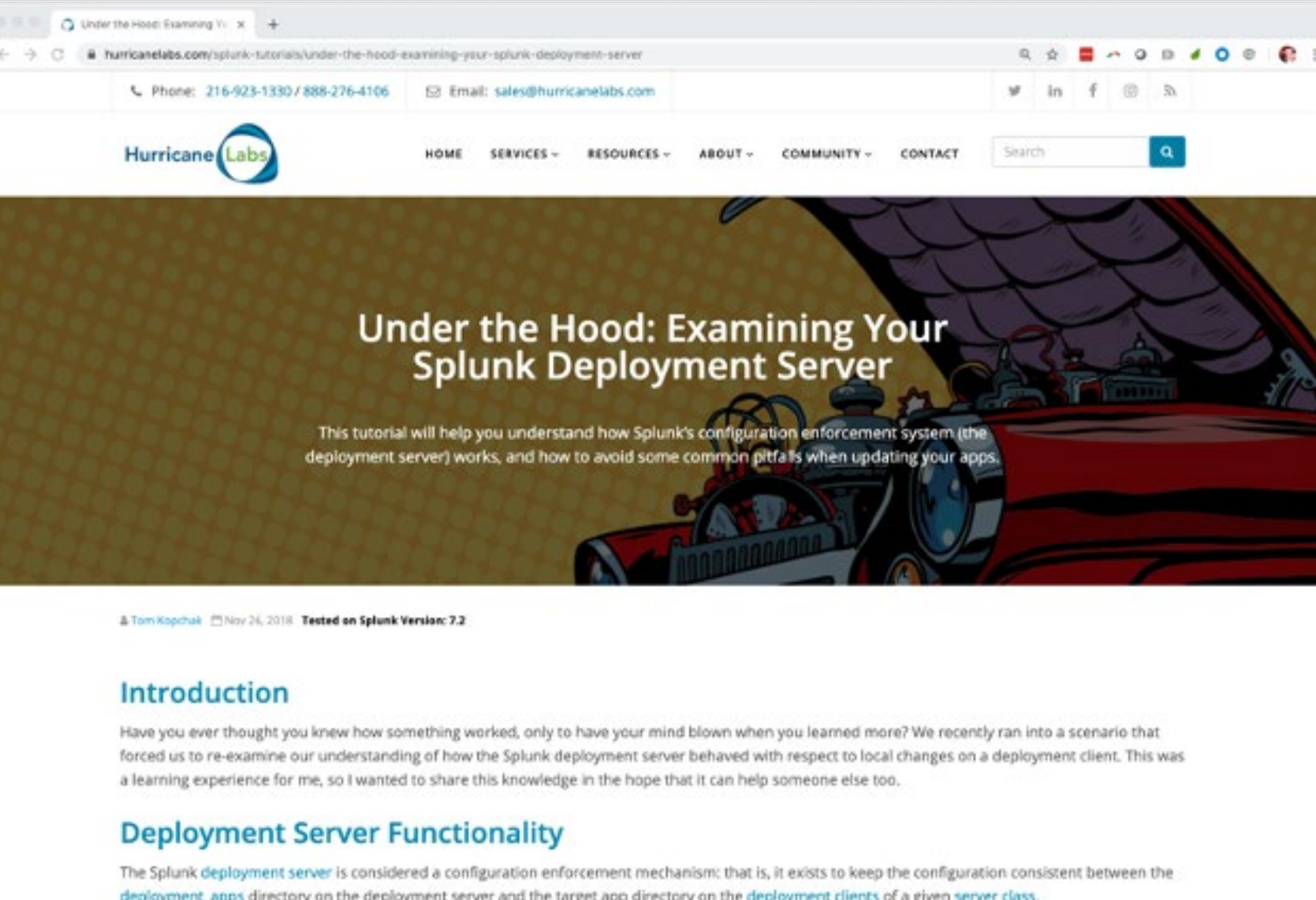
Example Server Classes

Server classes are defined
based on roles

- Apps assigned on the task for
each class

Minimizes DS complexity





Under the Hood: Examining Your
Splunk Deployment Server

This tutorial will help you understand how Splunk's configuration enforcement system (the deployment server) works, and how to avoid some common pitfalls when updating your apps.

Tom Kopchak Nov 26, 2018 Tested on Splunk Version: 7.2

Introduction

Have you ever thought you knew how something worked, only to have your mind blown when you learned more? We recently ran into a scenario that forced us to re-examine our understanding of how the Splunk deployment server behaved with respect to local changes on a deployment client. This was a learning experience for me, so I wanted to share this knowledge in the hope that it can help someone else too.

Deployment Server Functionality

The Splunk deployment server is considered a configuration enforcement mechanism: that is, it exists to keep the configuration consistent between the deployment apps directory on the deployment server and the target app directory on the deployment clients of a given server class.

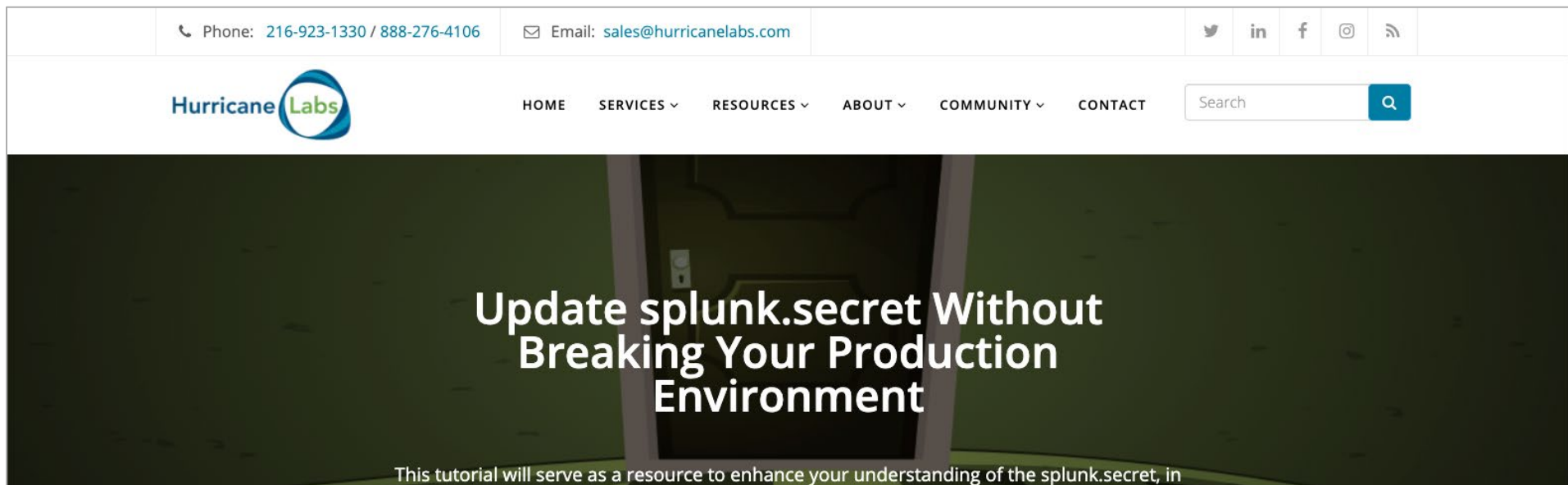
See my tutorial on the Deployment Server: <https://hlb.ninja/SplunkDS>

Splunk.Secret

Avoiding plaintext passwords in distributed configuration files

What is splunk.secret?

- Encryption key used by Splunk to encode passwords in configuration files
- Avoids keeping passwords in plaintext



Splunk.Secret

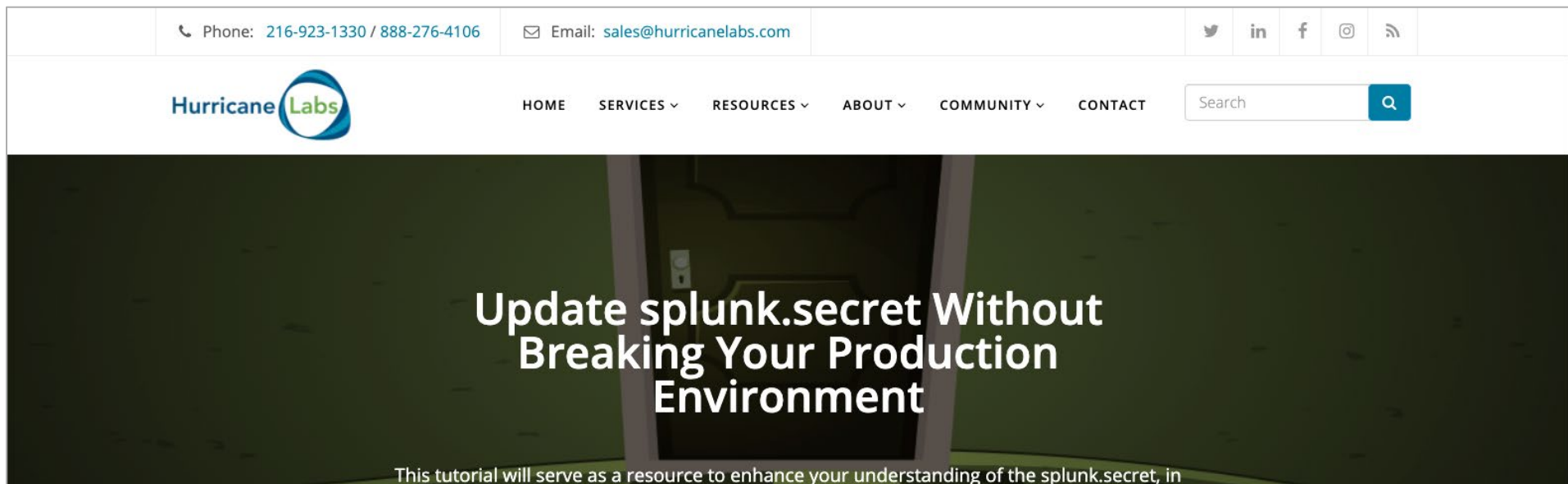
Avoiding plaintext passwords in distributed configuration files

What is splunk.secret?

- Encryption key used by Splunk to encode passwords in configuration files
- Avoids keeping passwords in plaintext

Why would you want it to be consistent?

- Allows for encrypted config files to be distributed via deployment server



Splunk.Secret

Avoiding plaintext passwords in distributed configuration files

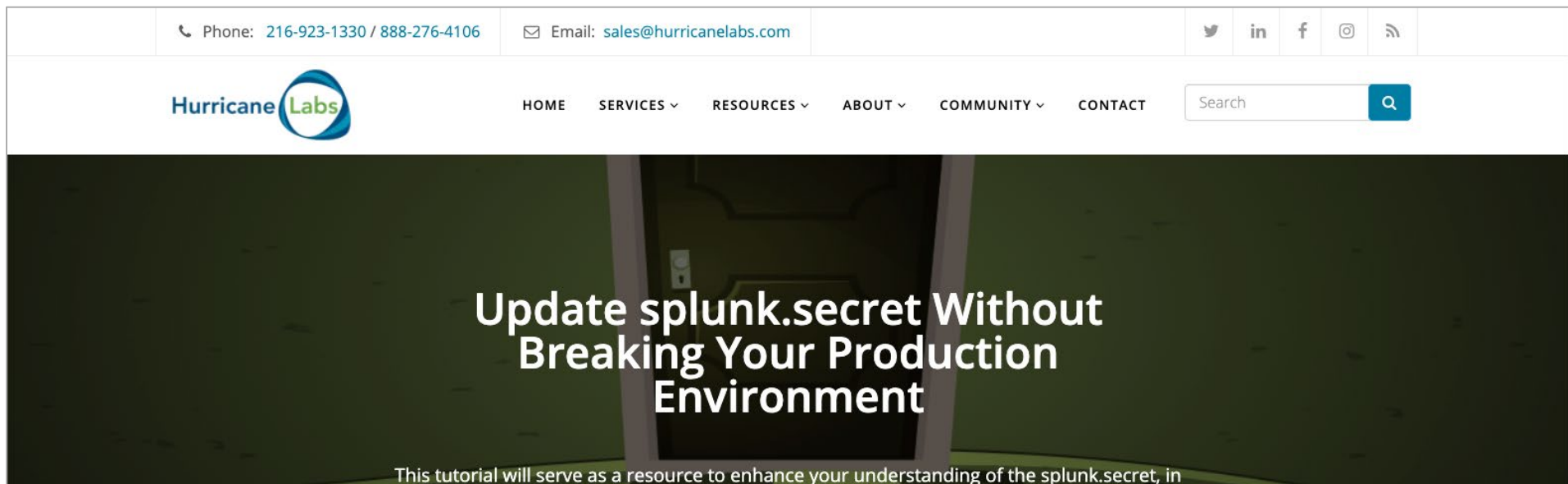
What is splunk.secret?

- Encryption key used by Splunk to encode passwords in configuration files
- Avoids keeping passwords in plaintext

Why would you want it to be consistent?

- Allows for encrypted config files to be distributed via deployment server

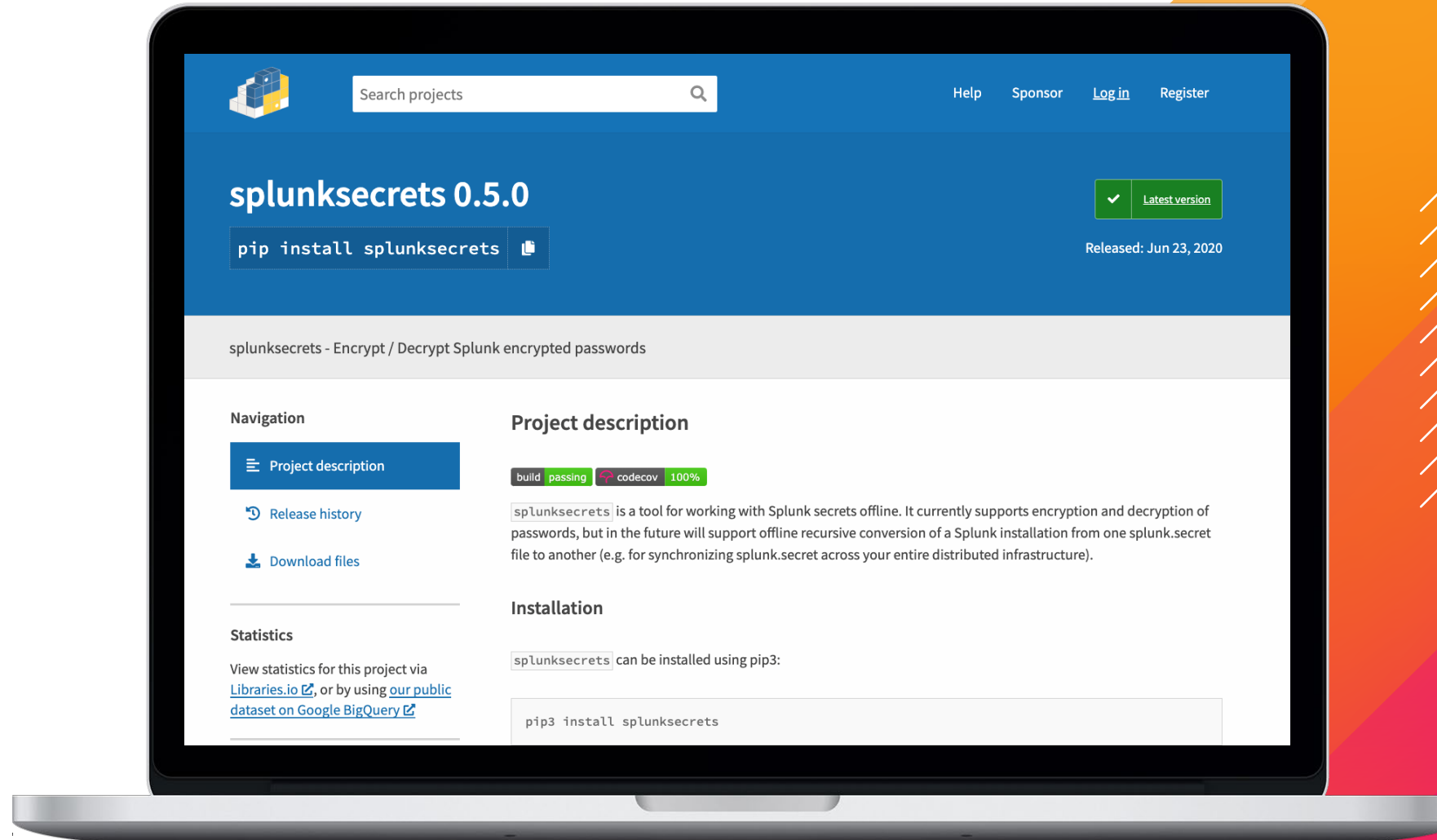
Great example: authentication config



Need to deal with Spunk Secrets?

Steve McMaster's
splunksecrets tool:

<https://hlb.ninja/SecretsPY>



Health Status Monitoring

A great place to start when investigating issues

Health Status of Splunkd

 **splunkd**

Data Forwarding

- Splunk-2-Splunk Forwarding
 -  TCPOutAutoLB-0

File Monitor Input

-  BatchReader-0
-  TailReader-0

Index Processor

-  Buckets
-  Disk Space
-  Index Optimization

Indexer Clustering

-  Search Head Connectivity

Search Scheduler

-  Search Lag
-  Searches Delayed
-  Searches Skipped

How to interpret this health report:

This health report displays information from the `/health/splunkd/details` endpoint. There are three potential states for a feature:

-  Green: The feature is functioning properly.
-  Yellow: The feature is experiencing a problem. The feature's status might automatically improve, or it might worsen over time. For details, see Root Cause.
-  Red: The feature has severe issues and is negatively impacting the functionality of your deployment. For details, see Root Cause.
-  Grey: Health report is disabled for the feature.

To manage red and yellow threshold values for the individual features, go to [Health Report Manager](#).

For more information on this health report, see [Learn more](#).



Data Ingestion Quality

Schema on the fly doesn't mean you can't plan ahead

Sourcetypes

Don't make them up! (unless you have to)

- If an existing TA (add-on) or app defines a sourcetype for your log – use it!

Sourcetypes

Don't make them up! (unless you have to)

- If an existing TA (add-on) or app defines a sourcetype for your log – use it!

If you must make up a sourcetype, make it logical

- vendor:product
- vendor:product:type

Sourcetypes

Don't make them up! (unless you have to)

- If an existing TA (add-on) or app defines a sourcetype for your log – use it!

If you must make up a sourcetype, make it logical

- vendor:product
- vendor:product:type

Avoid configurations that lead to <file>_toosmall sourcetypes

- Major culprit: Splunk *nix TA
- Better approach: explicitly assign sourcetypes to specific log files

```
[monitor:///var/log]
whitelist=(\.log|log$|messages|secure|auth|mesg$|cron$|acpid$|\.out)
blacklist=(lastlog|anaconda\.syslog)
disabled = 1
```

Sourcetypes

Don't make them up! (unless you have to)

- If an existing TA (add-on) or app defines a sourcetype for your log – use it!

If you must make up a sourcetype, make it logical

- vendor:product
- vendor:product:type

Avoid configurations that lead to <file>_toosmall sourcetypes

- Major culprit: Splunk *nix TA
- Better approach: explicitly assign sourcetypes to specific log files



```
[monitor:///var/log]
white = (log$|messages|secure|auth|msg$|cron$|acpid$|\.out)
blacklist = (log|anaconda\.syslog)
disabled = 1
```

Data Ingestion from Third-Party Cloud Products

What the HEC should you try first?

Data Ingestion from Third-Party Cloud Products

What the HEC should you try first?

Third-party APIs are often an adventure filled with landmines/sadness

Data Ingestion from Third-Party Cloud Products

What the HEC should you try first?

Third-party APIs are often an adventure filled with landmines/sadness

Third-party apps don't necessarily make sense for your environment

- App developers make a lot of assumptions

Data Ingestion from Third-Party Cloud Products

What the HEC should you try first?

Third-party APIs are often an adventure filled with landmines/sadness

Third-party apps don't necessarily make sense for your environment

- App developers make a lot of assumptions

HF + API App vs. HEC (HTTP Event Collector)

- Choose HEC every time if it's an option
- HF has limitations:
 - Delay in ingestion due to polling
 - Single point of failure, no automatic failover

Data Ingestion from Third-Party Cloud Products

What the HEC should you try first?

Third-party APIs are often an adventure filled with landmines/sadness

Third-party apps don't necessarily make sense for your environment

- App developers make a lot of assumptions

HF + API App vs. HEC (HTTP Event Collector)

- Choose HEC every time if it's an option
- HF has limitations:
 - Delay in ingestion due to polling
 - Single point of failure, no automatic failover

Don't be afraid to talk to the vendor/app developer

- We all win with a better app

Data Ingestion from Third-Party Cloud Products

What the HEC should you try first?

Third-party APIs are often an adventure filled with landmines/sadness

Third-party apps don't necessarily make sense for your environment

- App developers make a lot of assumptions

HF + API App vs. HEC (HTTP Event Collector)

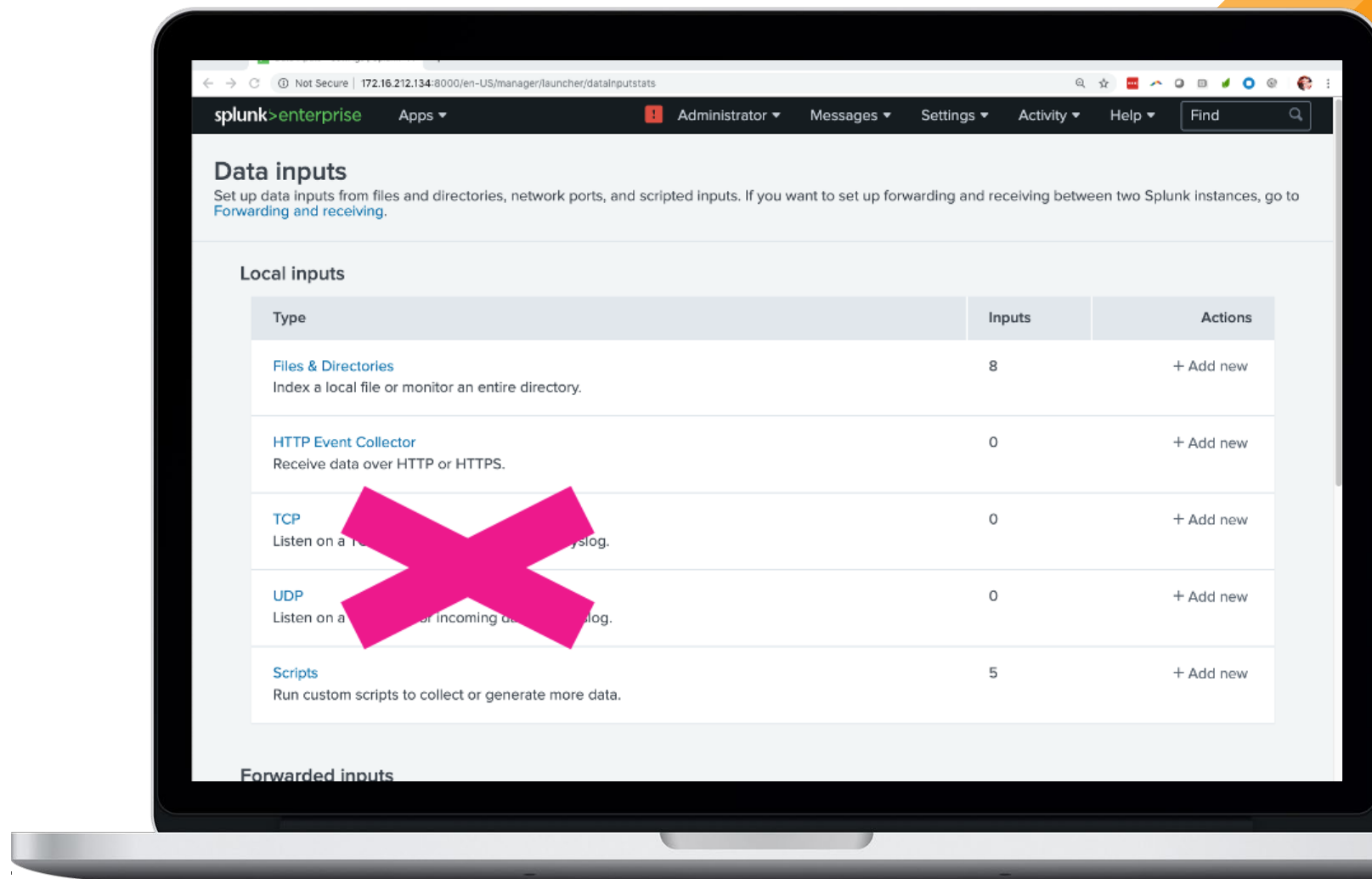
- Choose HEC every time if it's an option
- HF has limitations:
 - Delay in ingestion due to polling
 - Single point of failure, no automatic failover

Don't be afraid to talk to the vendor/app developer

- We all win with a better app

Key Takeaway: Stream data whenever possible

**Don't use
direct
TCP/UDP
inputs**
(if you like
your data)



Key Takeaways

1. All of this knowledge comes with experience
2. Don't be afraid to ask for help
3. Keep what you've learned in mind when seeking advice!
4. Download these slides for reference later



Thank You

Please provide feedback via the
SESSION SURVEY



Reference

More in-depth content and details

Splunk Secret:

- Update Splunk Secret: <https://hlb.ninja/2mbJ9zx>
- SplunkSecrets tool: <https://hlb.ninja/SecretsPY>

Splunk Deployment Server: <https://hlb.ninja/SplunkDS>

Splunk Architecture and Environment Design:

- Environment Design/Architecture: <https://hlb.ninja/2nM35cM>
- Sizing Storage: <https://hlb.ninja/2nJ9a9E>

Splunk password resets: <https://hlb.ninja/2or3lOC>

Search performance (wildcards) demo: <https://hlb.ninja/32iw1cl>

Reference

(Even) More in-depth content and details

Exporting (a lot of) data: <https://hlb.ninja/2m9h7o9>

All about Splunk Certificates: <https://hlb.ninja/SplunkCerts>

Splunk AppInspect + Cloud Vetting: <https://hlb.ninja/2Qf12c2>

Setting up HEC: <http://hlb.ninja/SplunkHEC>

Configuring LDAP Authentication: <http://hlb.ninja/SplunkLDAP>

Custom times in search: <https://hlb.ninja/32ij0jD>