Splunk Enterprise Data Administration

Document Usage Guidelines

- Should be used only for enrolled students
- Not meant to be a self-paced document, an instructor is needed
- Do not distribute

Course Prerequisites

- Required:
  - Splunk Fundamentals 1
  - Splunk Fundamentals 2
- Strongly Recommended:
  - Splunk Enterprise System Administration

Course Goals

- Understand sourcetypes
- Manage and deploy forwarders with Forwarder Management
- Configure data inputs
  - File monitors
  - Network inputs (TCP/UDP)
  - Scripted inputs
  - HTTP inputs (via the HTTP Event Collector)
- Customize the input phase parsing process
- Define transformations to modify raw data before it is indexed
- Define search time field extractions

Course Outline

Module 1: Introducing Splunk Data Administration

Module 2: Getting Data In - Staging

Module 3: Forwarder Configuration

Module 4: Heavy Forwarders & Forwarder Management

Module 5: Monitor Inputs

Module 6: Network and Scripted Inputs

Module 7: Windows and Agentless Inputs

Module 8: Fine-tuning Inputs

Module 9: Parsing Phase and Data Preview

Module 10: Manipulating Raw Data

Module 11: Supporting Knowledge Objects

Data Administrator vs System Administrator

Splunk Data Administrator

Data onboarding and management

- Work with users requesting new data sources
- Document existing and newly ingested data sources
- Design and manage inputs for UFs/HFs to capture data
- Manage parsing, event line breaking, timestamp extraction
- Move configuration through non-production testing as required
- Deploy changes to production
- Manage Splunk configuration files

Splunk System Administrator

System management

- Install, configure, and manage Splunk components
- Install and manage Splunk apps
- Manage Splunk licensing
- Manage Splunk indexes
- Manage Splunk users and authentication
- Manage Splunk configuration files
- Monitor MC and respond to system health alerts

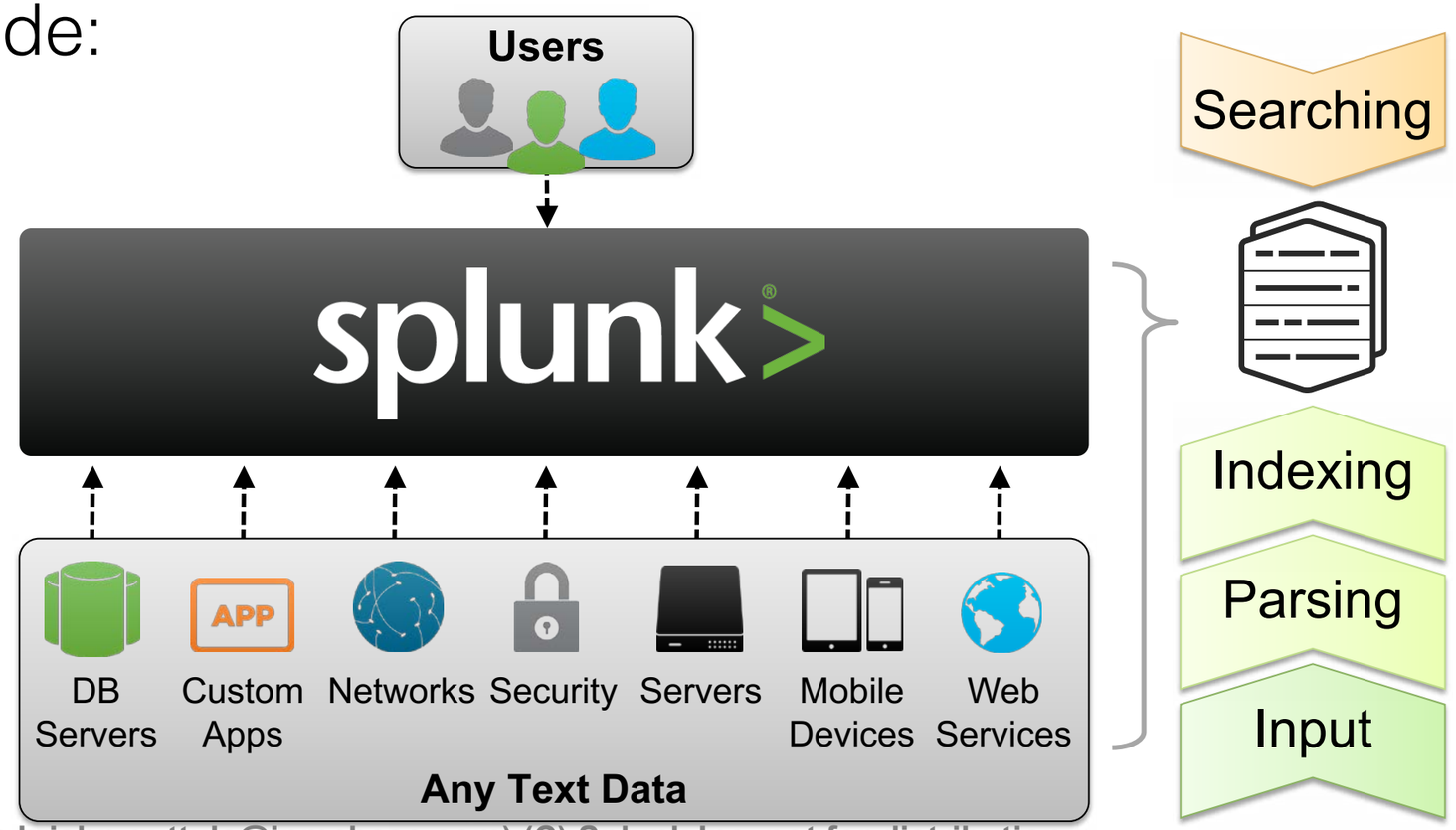
Module 1: Introducing Splunk Data Administration

Module 1 Objectives

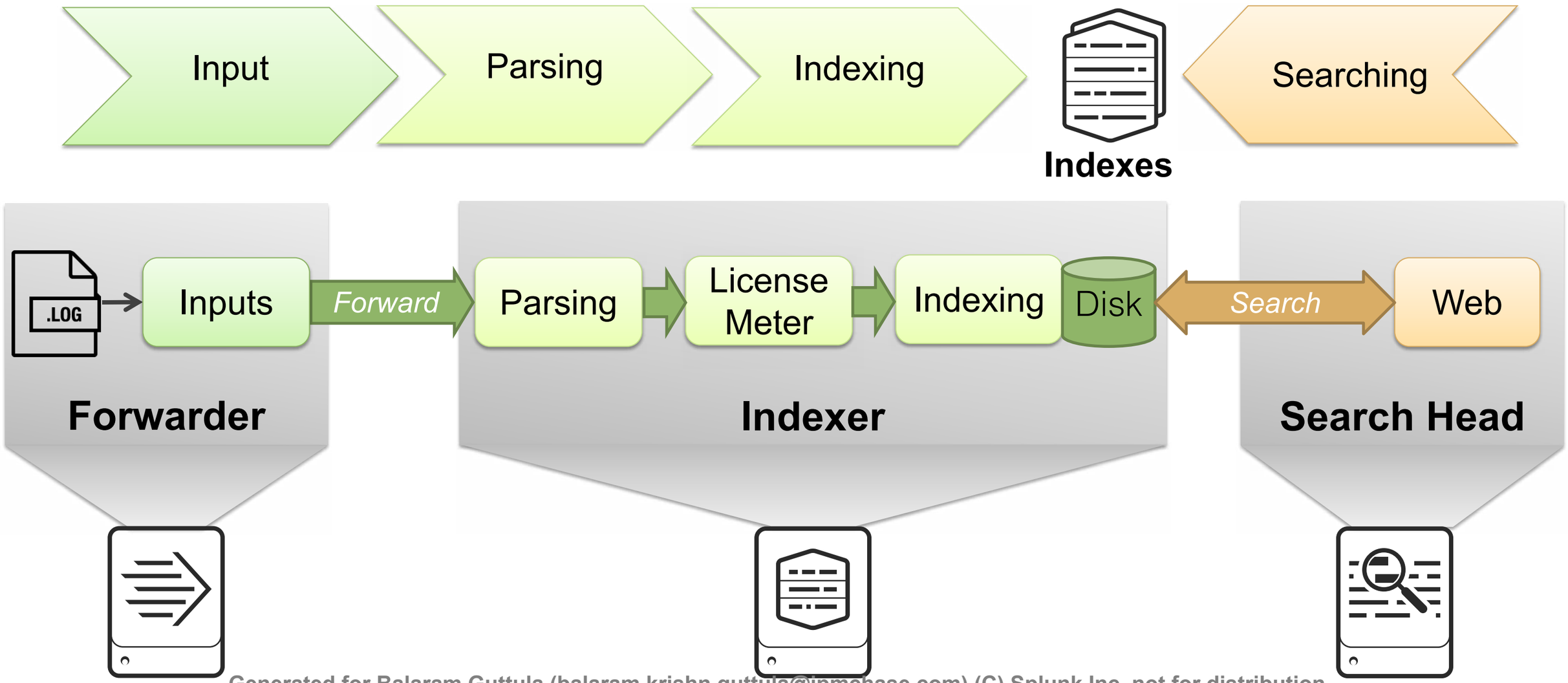
- Provide an overview of Splunk
- Describe the four phases of the distributed model
- Identify Splunk configuration files and directories
- Describe index-time and search-time precedence
- Use **`btool`** to retrieve Splunk configuration information

Splunk Overview

- Splunk can be deployed in a variety of configurations
- Scales from a single server to a distributed infrastructure
- Four stages of Splunk include:
  - **Input** any text data
  - **Parse** the data into events
  - **Index** and store events
  - **Search** and report

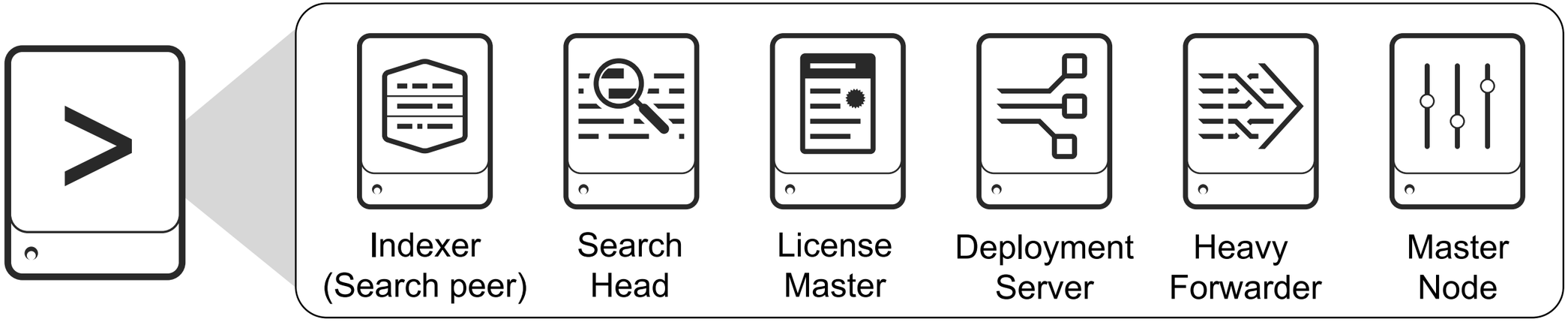


The Four Phases of the Distributed Model

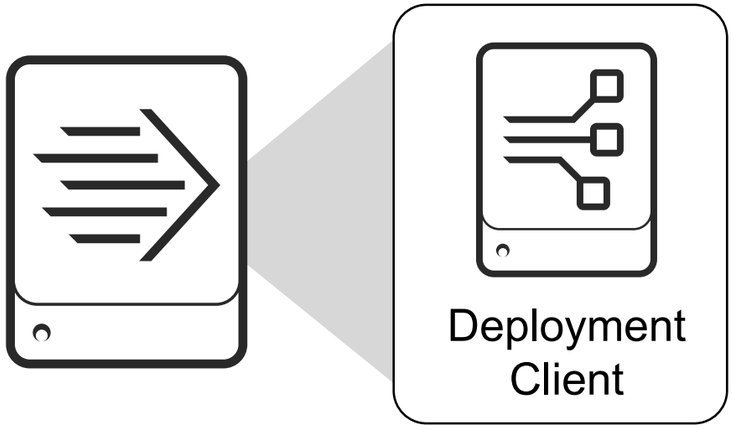


Software in Splunk Enterprise packages

Splunk Enterprise package



Universal Forwarder package



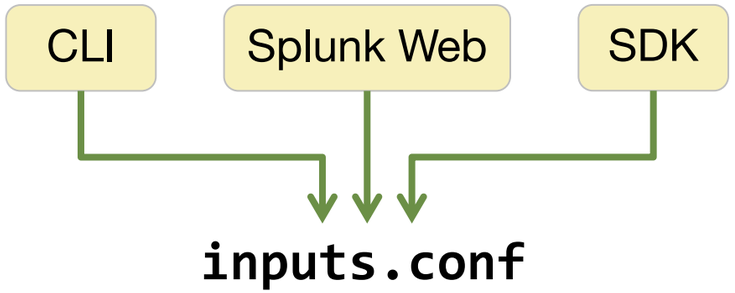
Note

The System Administrator is responsible for installing and configuring Splunk components.

Splunk Configuration Files

Configuration Files (`.conf`)

- Govern an aspect of Splunk functionality
- Text files using a generally case-sensitive **`[stanza]`** and **`attribute = value`** format
- Modified using Splunk Web, CLI, SDK, app install, or directly editing
- Saved under **`SPLUNK_HOME/etc`**
- Come with documentation and examples under **`SPLUNK_HOME/etc/system/README/`**

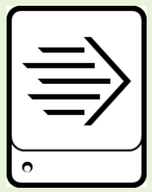



```
[default]
host=www

[monitor:///var/log/httpd]
sourcetype = access_common
ignoreOlderThan = 7d
index = web
```

Note

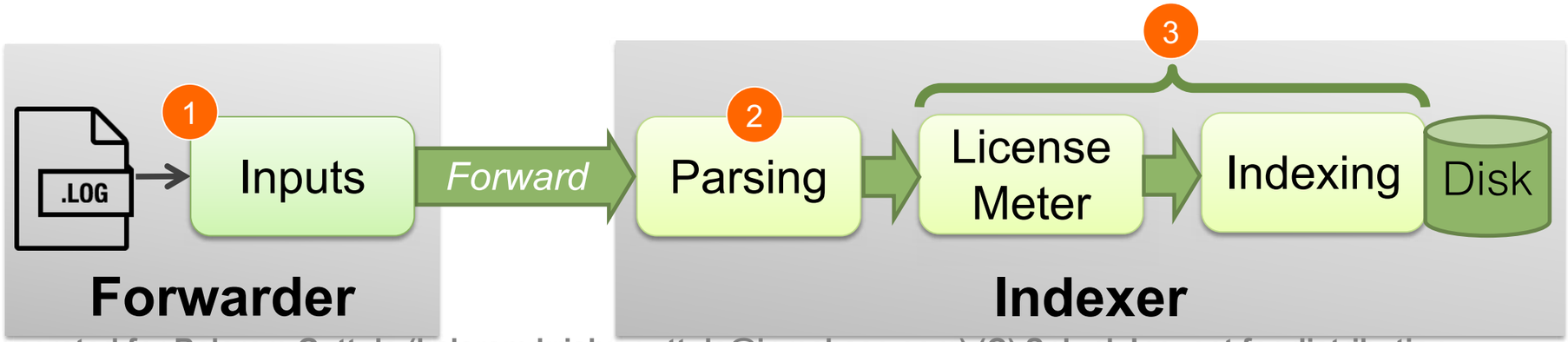
For **`.conf`** file documentation and examples view **`SPLUNK_HOME/etc/system/README/`**:
- **`*.conf.spec`**
- **`*.conf.example`**

Commonly Used Splunk Configuration Files

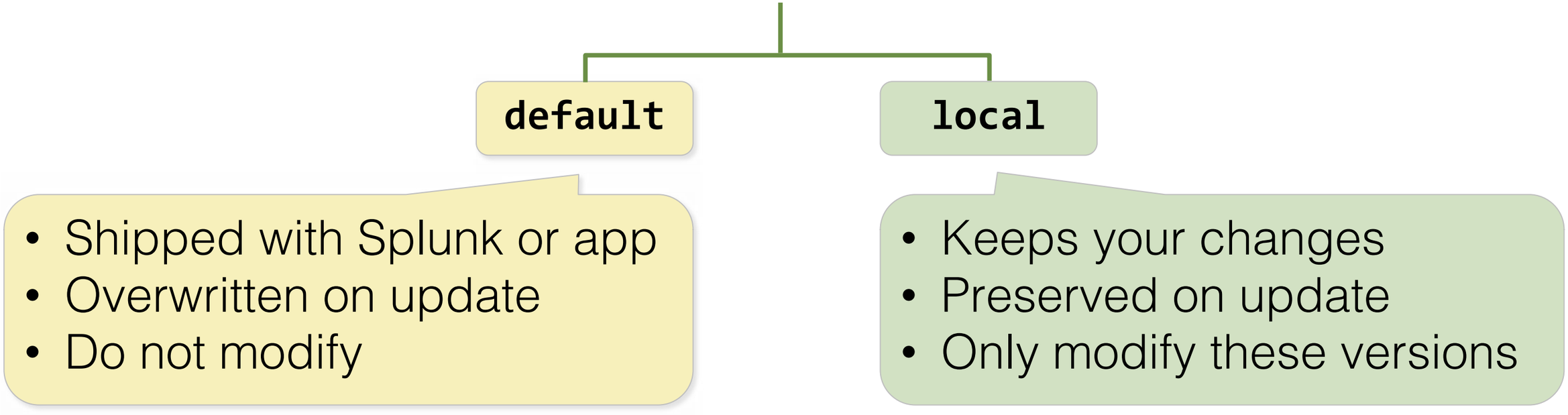
Component	inputs.conf	props.conf	outputs.conf
Universal Forwarder 	What data to collect (production logs)	Limited parsing (such as character encoding, refine MetaData, event breaks*)	Where to forward the data (generally to Indexers)
Indexer 	What data to collect; which ports to listen to	Refine MetaData at event level, event breaks, Time Extraction, TZ, data transformation	Not generally needed (generally Indexer does not forward data)
Search Head 	What data to collect (internal Splunk logs)	Field Extractions (search-time), lookups, and so on	Where to forward the data (such as the internal Splunk logs to an indexer)

Index-Time Process

1. **Input phase**: Handled at the source (usually a forwarder)
   - The data sources are being opened and read
   - Data is handled as streams; configuration settings are applied to the entire stream
2. **Parsing phase**: Handled by indexers (or heavy forwarders)
   - Data is broken up into events and advanced processing can be performed
3. **Indexing phase**: Handled by indexers
   - License meter runs as data is initially written to disk, prior to compression
   - After data is written to disk, it **cannot** be changed



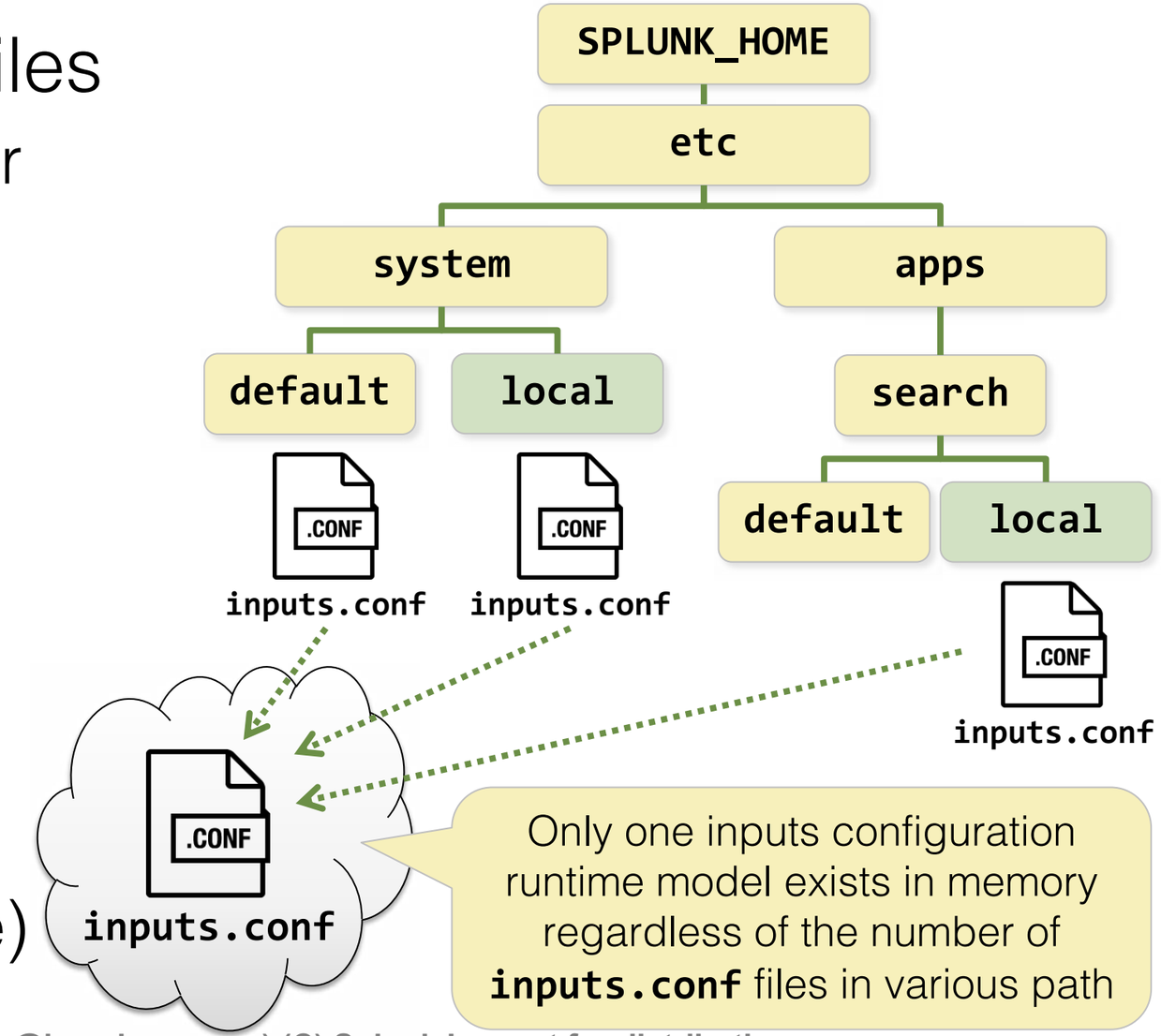
Default vs. Local Configuration



- Avoid storing configurations in **SPLUNK_HOME/etc/system/...**
  - Manage your configurations in the appropriate app under **SPLUNK_HOME/etc/apps/<*appname*>/local**
  - If you don't have an app, create one to manage system settings

Merging of Configuration Files

- Splunk merges configuration files
  - Generally when Splunk starts, or when searches are run
  - Into a single run-time model for each file type
  - As a union of all files if no duplicates/conflicts exist
- In case of conflicts, priority is based on the context:
  - **Global context** (index-time)
  - **App/User context** (search-time)

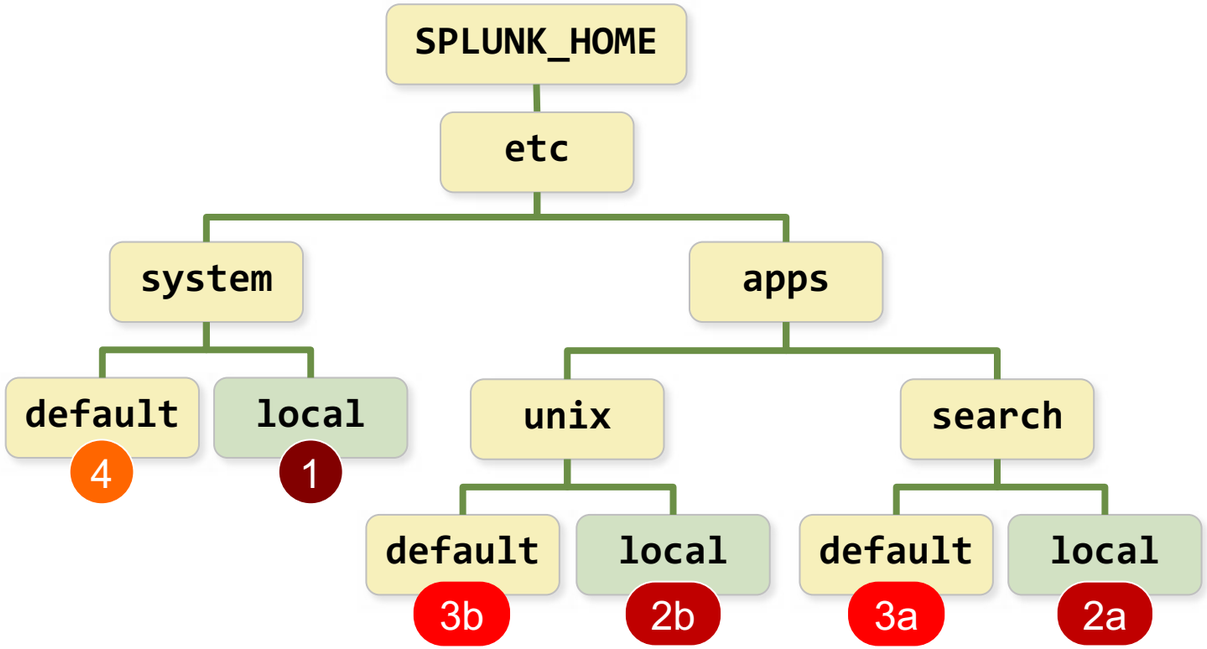


File Context and Index-time vs Search-time

	Global Context	App/User Context
Used during:	**Index-time**	**Search-time**
Used by:	• User-independent tasks • Background tasks • Input, parsing, indexing	• User-related activity • Searching • Search-time processing
Example use-case:	A network input to collect **syslog** data	Mary's private report in the **Search** app
Example files:	**`inputs.conf`** **`outputs.conf`** **`props.conf`**	**`macros.conf`** **`savedsearches.conf`** **`props.conf`**

docs.splunk.com/Documentation/Splunk/latest/Admin/Wheretofindtheconfigurationfiles

Index-Time Precedence (Global Context)



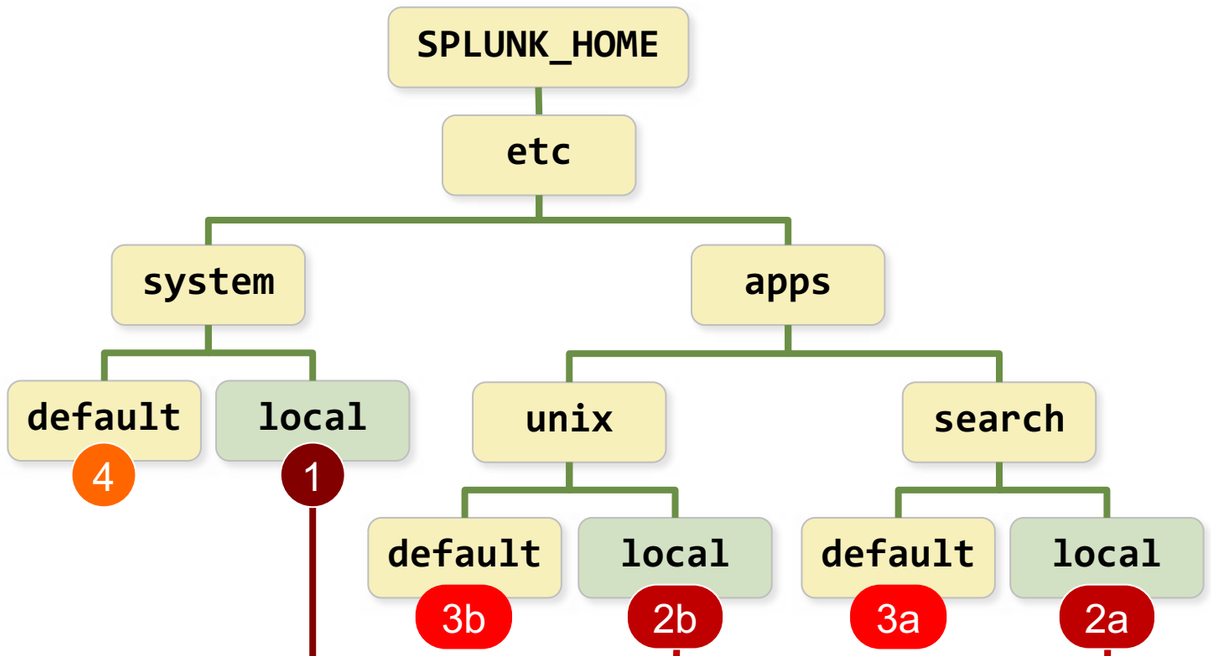
Precedence order

1. System local directory
   etc/system/local
2. App local directories*
   etc/apps/*appname*/local
3. App default directories*
   etc/apps/*appname*/default
4. System default directory
   etc/system/default

Note

* When determining priority of app directories in **global** context (for steps 2 and 3), Splunk uses *lexicographical* order. (Files in apps directory "A" have higher priority than files in apps directory "B".)

Example of Index-Time Precedence with **inputs.conf**



inputs.conf

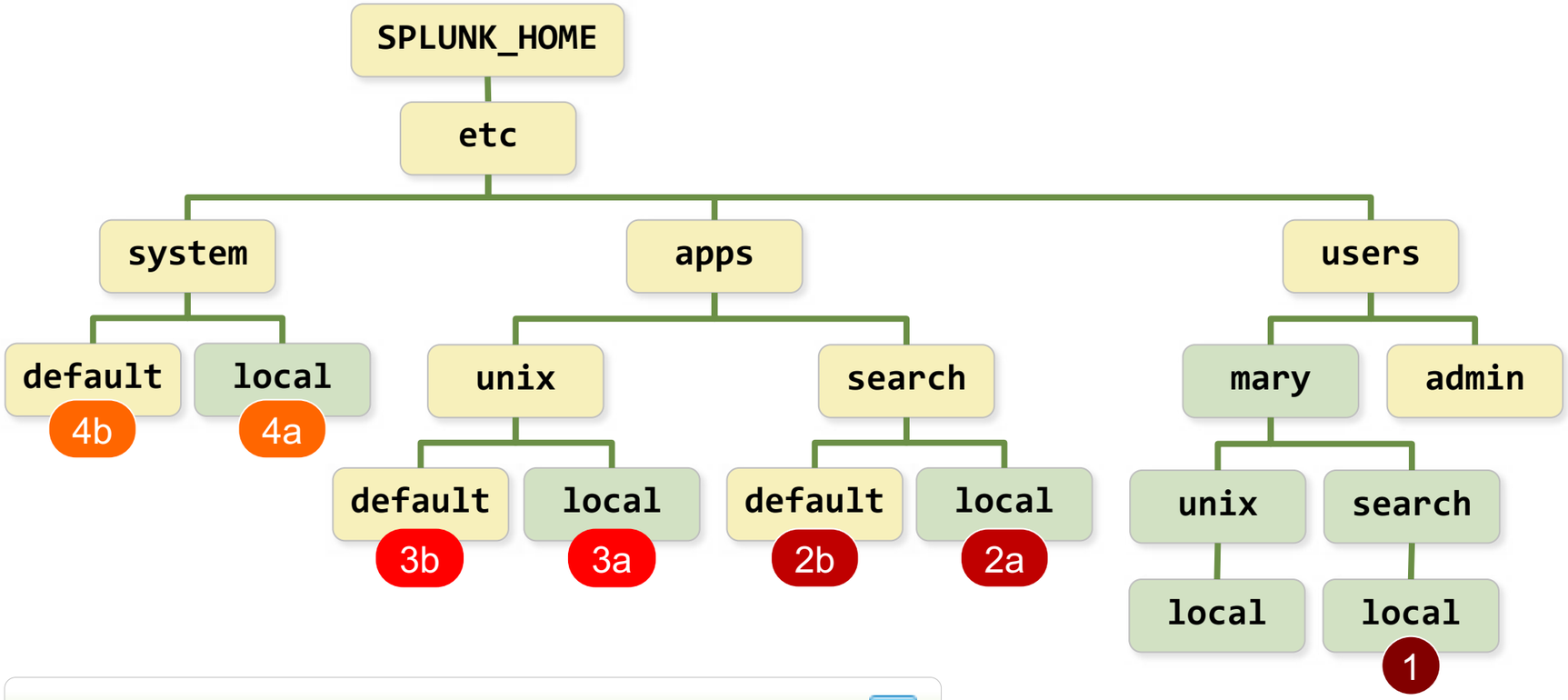
```
[default]
host = server1

[monitor:///var/log/secure.log]
host = logsvr1
sourcetype = linux_secure
index=security

[monitor:///opt/log/www1/access.log]
host = websvr1
sourcetype = access_combined_wcookie
```

```
[monitor:///var/log/secure.log]
sourcetype = access_combined
index=security
```

```
[monitor:///var/log/secure.log]
host = logsvr1
sourcetype = linux_secure

[monitor:///opt/log/www1/access.log]
host = www1
sourcetype = access_combined_wcookie
```

```
[default]
host = server1

[monitor:///opt/log/www1/access.log]
host = websvr1
```


Search-Time Precedence (App/User Context)



Precedence order

1. Current user directory for app
   etc/users/*user*/*appname*/local
2. App directory - running app
   etc/apps/*appname*/local
   etc/apps/*appname*/default
3. App directories - all other apps*
   etc/apps/*appname*/local
   etc/apps/*appname*/default
4. System directories
   etc/system/local
   etc/system/default

Note

* If objects from the app are exported globally with **.meta** file setting, evaluate all other app directories using *reverse lexicographical* order. (Files in apps directory "B" have higher priority than directory "A".)

Validating the Splunk configuration

Validating the in-memory configuration

- Performed with **`splunk show config`** CLI or REST API
- Syntax: **`splunk show config <conf_file>`**
- Example: **`splunk show config inputs`**

Validating the on-disk configuration

- Performed with **`splunk btool`** CLI
- Syntax: **`splunk btool <conf_file> list`**
- Example: **`splunk btool inputs list`**

Configuration Validation with **btool**

- **splunk btool <*conf-name*> list [*options*]**
  - Shows on-disk configuration for requested file
  - Run **splunk btool check** each time Splunk starts
  - Useful for checking the configuration scope and permission rules
    - Use **--debug** to display the exact **.conf** file location
    - Add **--user= <*user*> --app=<*app*>** to see the user/app context layering
- Examples:

```
splunk help btool
splunk btool check
splunk btool inputs list
splunk btool inputs list monitor:///var/log
splunk btool inputs list monitor:///var/log --debug
```

docs.splunk.com/Documentation/Splunk/latest/Troubleshooting/Usebtooltotroubleshootconfigurations

Example using **btool**

Scenario: What are the **/var/log/secure.log** input configurations and where are they specified?

```
> splunk btool inputs list monitor:///var/log/secure.log --debug

  etc/apps/search/local/inputs.conf    [monitor:///var/log/secure.log]
  etc/apps/search/local/inputs.conf    host = logsvr1
  etc/apps/unix/local/inputs.conf      index = security
  etc/apps/search/local/inputs.conf    sourcetype = linux_secure
```

etc/apps/unix/local/inputs.conf

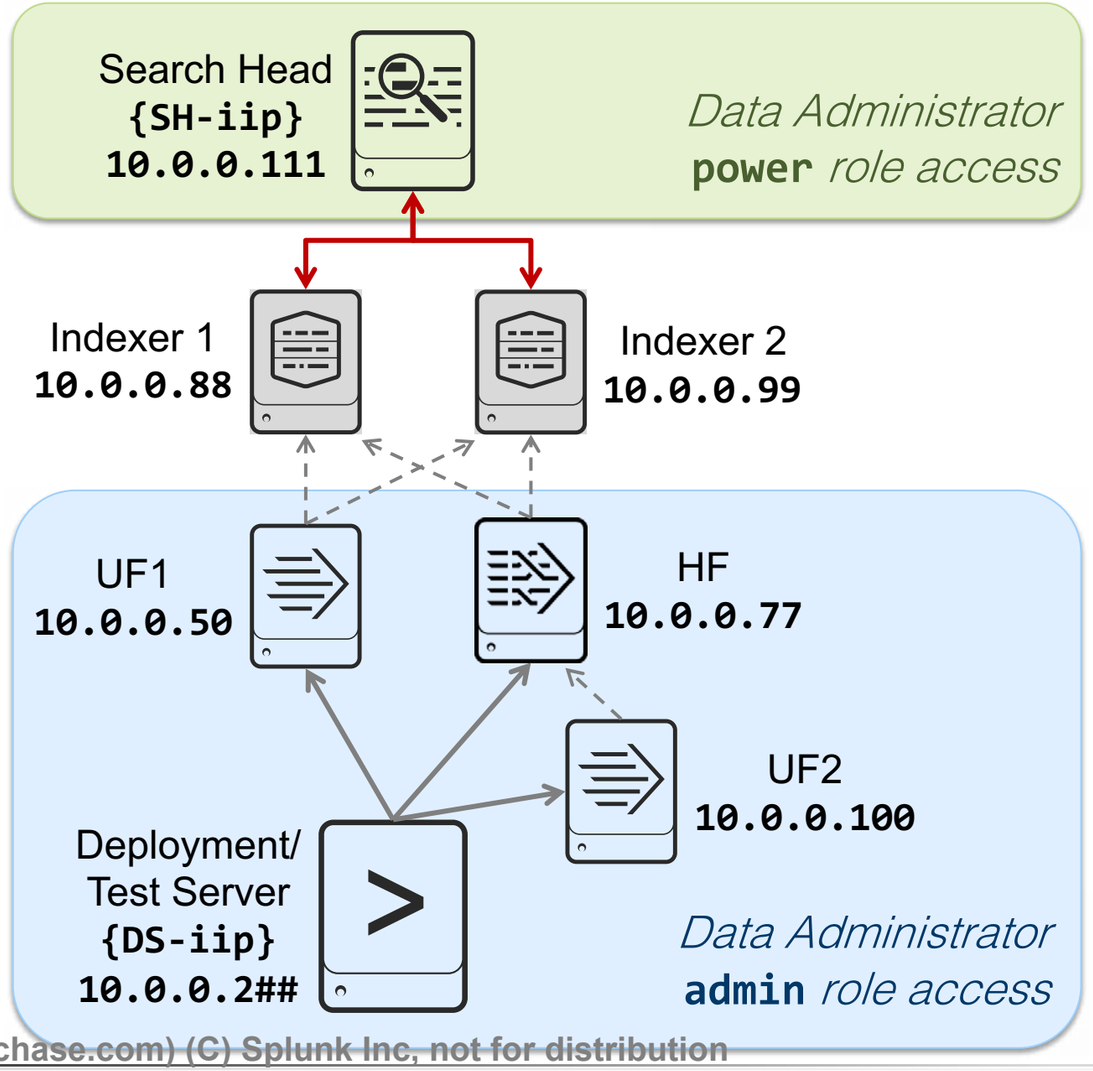
```
[monitor:///var/log/secure.log]
sourcetype = access_combined
index = security
```

etc/apps/search/local/inputs.conf

```
[monitor:///var/log/secure.log]
host = logsvr1
sourcetype = linux_secure
```


Data Administrator Access Scenario

Splunk instance	Access
Search Head (search / verify data configs)	**power** role
Indexers	No access
Deployment/Test Server Forwarders (data sources and inputs)	**admin** role



Module 1 Knowledge Check

- Which installer will the System Admin use to install the heavy forwarder?
- Which configuration file tells a Splunk instance to ingest data?
- True or False. The best place to add a parsing configuration on an indexer would be **`SPLUNK_HOME/etc/system/local directory`** as it has the highest precedence.

Module 1 Knowledge Check – Answers

- Which installer will the System Admin use to install the heavy forwarder?

  Splunk Enterprise

- Which configuration file tells a Splunk instance to ingest data?

  inputs.conf

- True or False. The best place to add a parsing configuration on an indexer would be the **SPLUNK_HOME/etc/system/local** directory, as it has the highest precedence.

  False. Best practice is to put the configuration in an *app's* **local** directory (**SPLUNK_HOME/etc/apps/<*appname*>/local**).

Module 1 Lab Exercise – Discover Lab Environment

Time: 15 minutes

Tasks:

- Log into search head and test/deployment server
- Discover Splunk Enterprise lab environment
- Use CLI to connect to Splunk components

Module 2:
Getting Data In – Staging

Module Objectives

- Describe data input types and default metadata settings
- Describe differences between the input and parsing phase
- Configure initial input testing with Splunk Web

Forwarding Data to Splunk

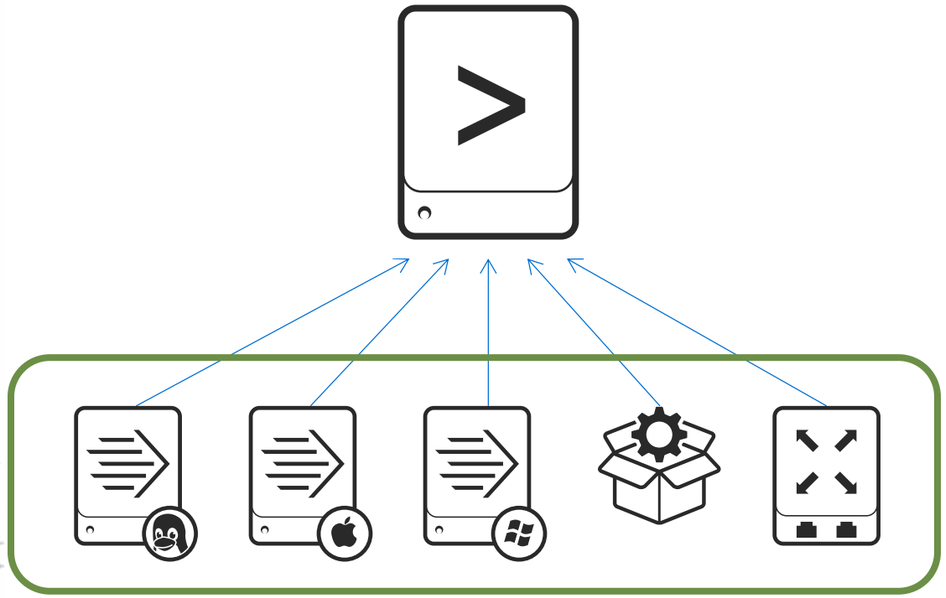
Any Source

- Computers
- Network devices
- Virtual Machines
- Internet of Things (IoT)
- Communication devices
- Sensors
- Databases

Any Text Data

- Logs
- Configurations
- Messages
- Call Detail Records
- Clickstream
- Alerts
- Metrics
- Scripts
- Changes
- Tickets

Splunk Enterprise



Indexes any text data from any source

Data Input Types

- Splunk supports many types of data input
  - **Files and directories:** monitoring text files and/or directory structures containing text files
  - **Network data:** listening on a port for network data
  - **Script output:** executing a script and using the output from the script as the input
  - **Windows logs:** monitoring Windows event logs, Active Directory, etc.
  - **HTTP**: using the HTTP Event Collector
  - And more...
- You can add data inputs with:
  - Apps and add-ons from Splunkbase
  - Splunk Web
  - CLI
  - Directly editing **`inputs.conf`**

Default Metadata Settings

- When Splunk indexes a data source, metadata values are assigned
  - Metadata is applied to the entire source
  - Defaults are used if alternates are not specified
  - Overriding values can be performed at input time or later

Metadata	Default
`source`	Path of input file, network hostname:port, or script name
`host`	Splunk hostname of the inputting instance (usually a forwarder)
`sourcetype`	Uses the source filename if Splunk cannot automatically determine
`index`	Defaults to the **`main`** index

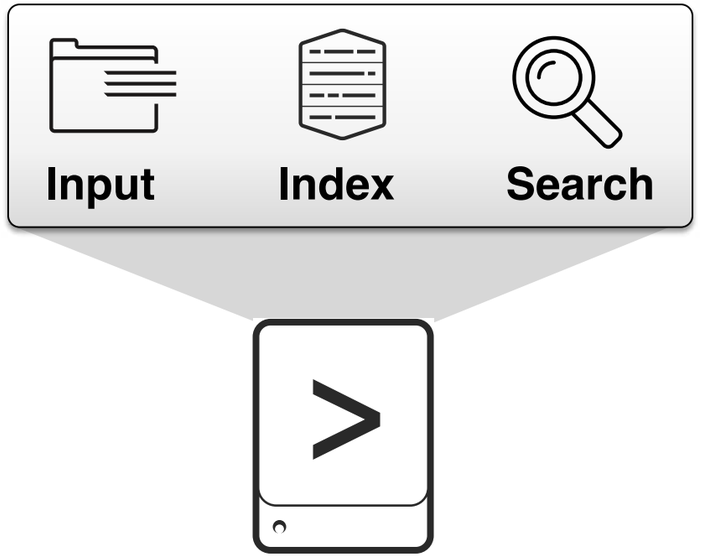
Input Phase vs. Parsing Phase

Input phase	Parsing phase

• **Most efficient, but less flexible**
• Acquires data from source
• Sets initial metadata fields: source, sourcetype, host, index, etc.
• Converts character encoding
• Operates on the entire data stream
• Most configuration done in **`inputs.conf`** on forwarder
  - Some configuration is in **`props.conf`**
	• **Less efficient, but finer control**
	• Breaks data into events with timestamps
	• Applies event-level transformations
	• Fine-tunes metadata settings from inputs phase
	• Operates on individual events
	• Most configuration done in **`props.conf`** on indexer
	  - Also: **`transforms.conf`**

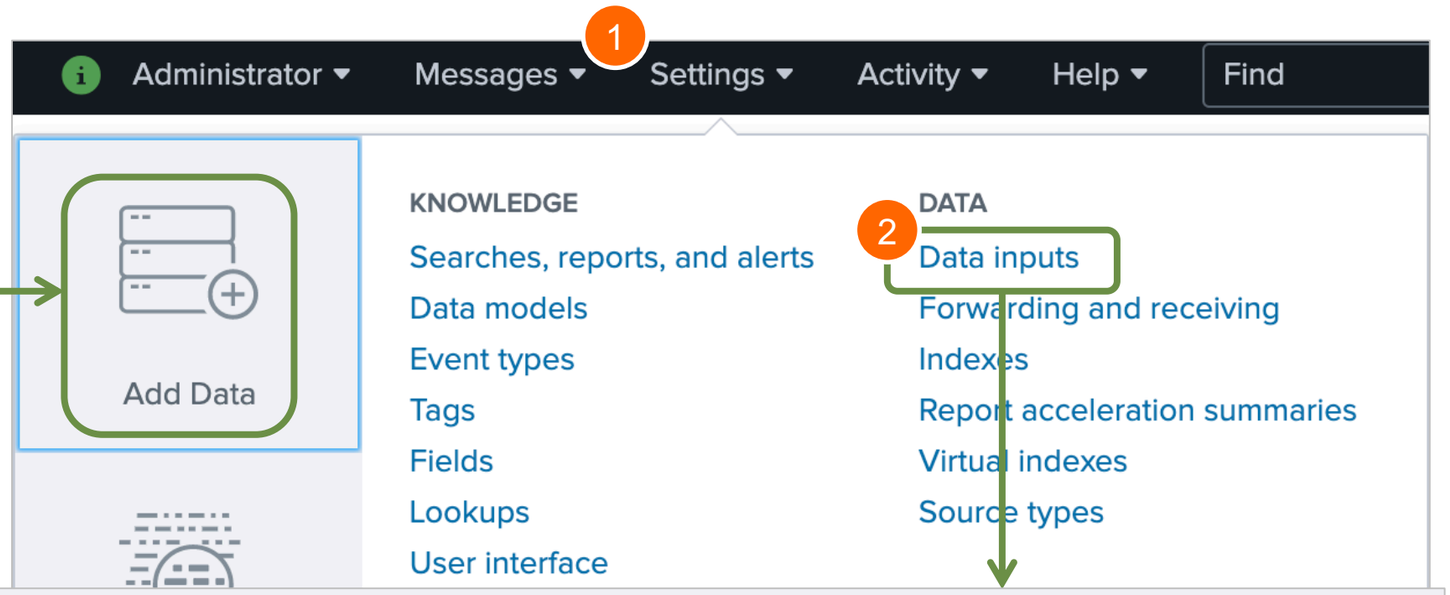
Testing Indexes with Input Staging

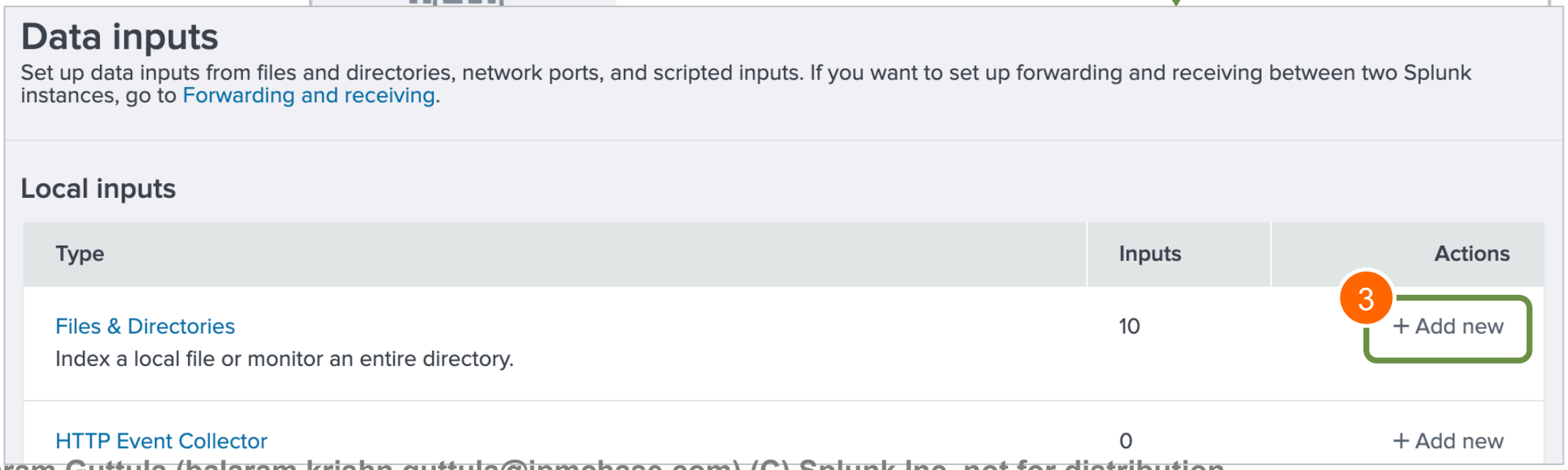
- Production data typically resides
  - On a forwarder (remote system)
  - Not on the indexer
- Test input data
  - Use Splunk Web > **Add Data**
  - Sample a log file on a test server
  - Check to see if **sourcetype** and other settings are applied correctly
  - Delete the test data, change your test configuration, and repeat as necessary



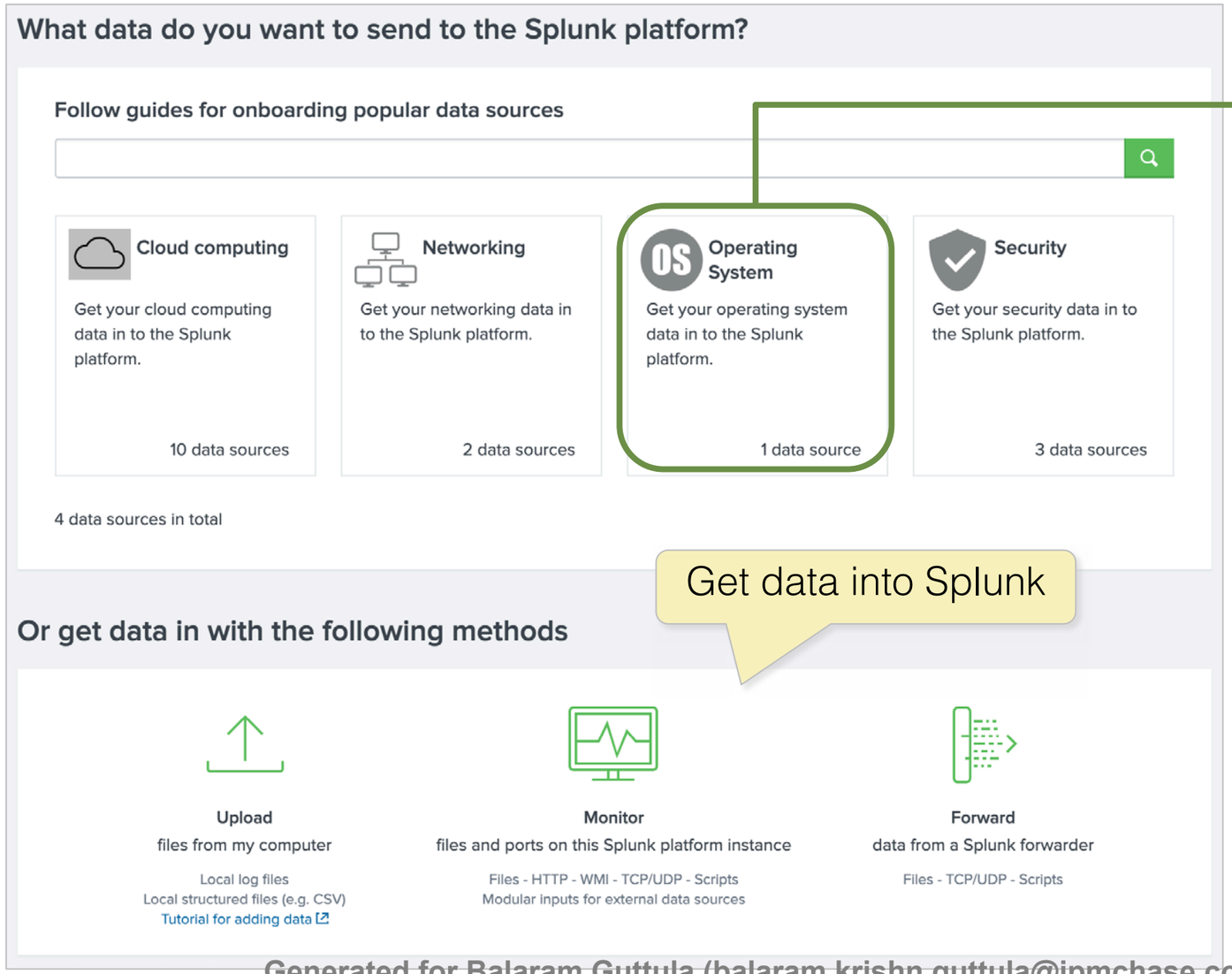
Adding an Input with Splunk Web

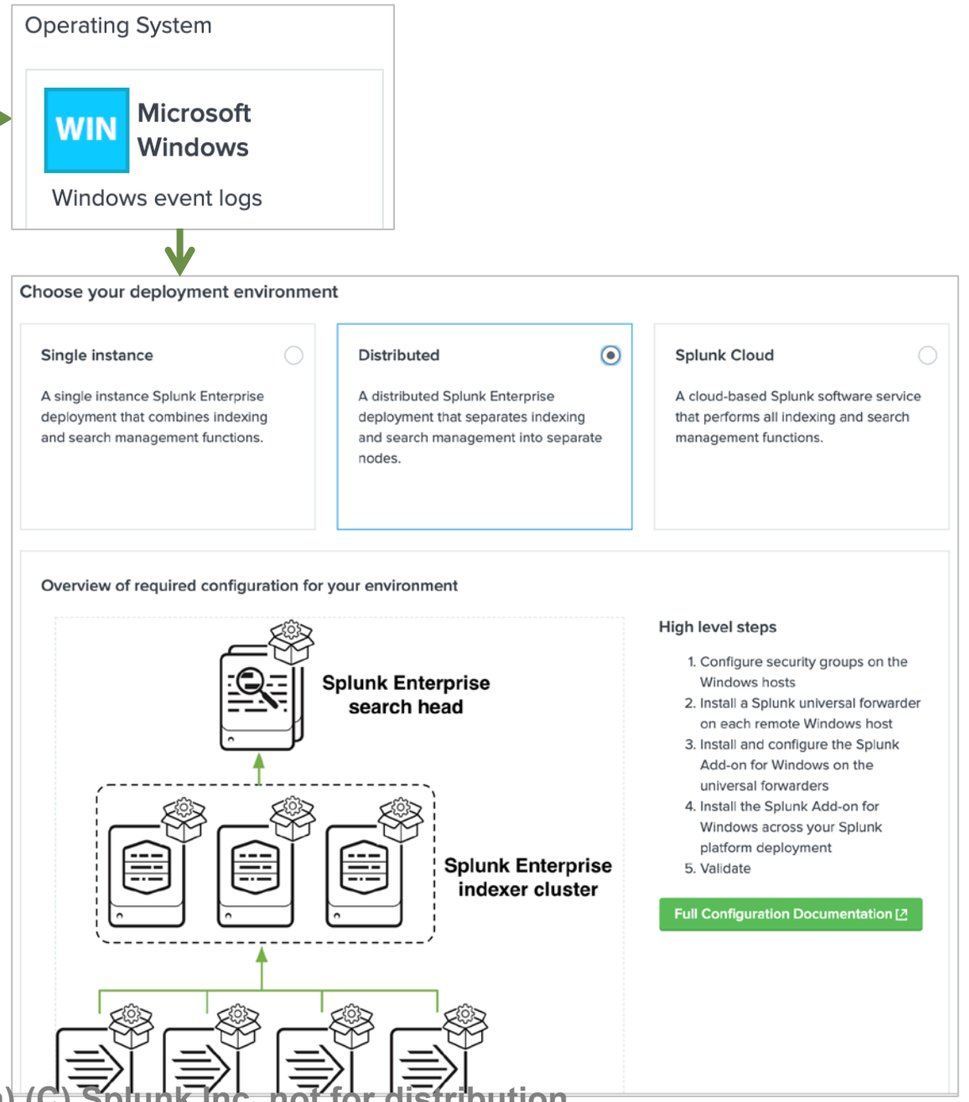
- Click the **Add Data** icon
  - On admin's **Home** page
  - On the **Settings** panel
- Or select:
  1. Settings
  2. Data inputs
  3. Add new



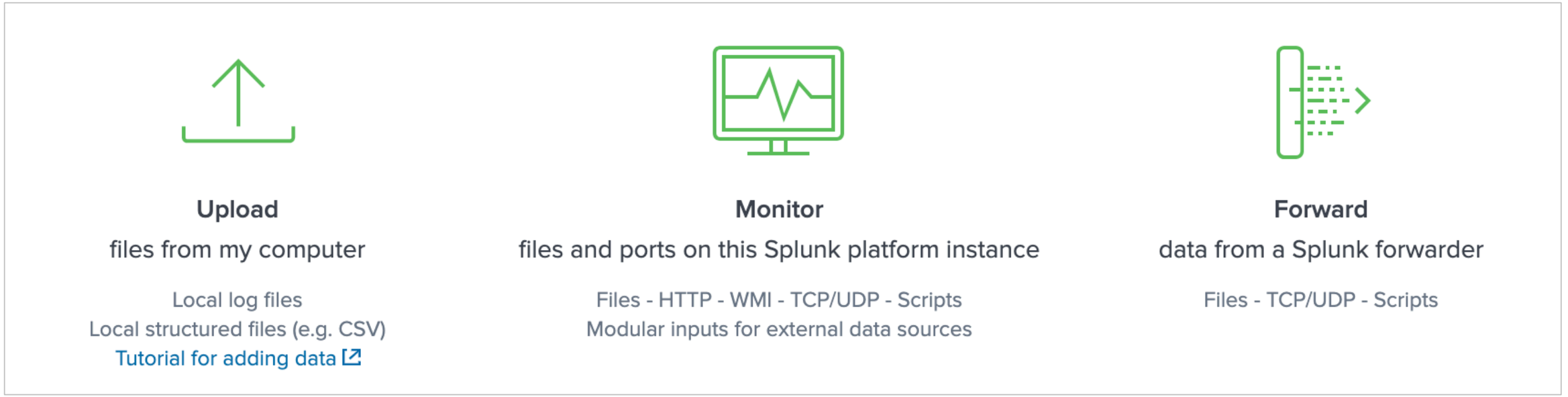


Add Data Menu





Add Data Menu (cont.)



Upload

- Allows uploading local files that only get indexed once
- Useful for testing or data that is created once and never gets updated
- Does not update **`inputs.conf`**

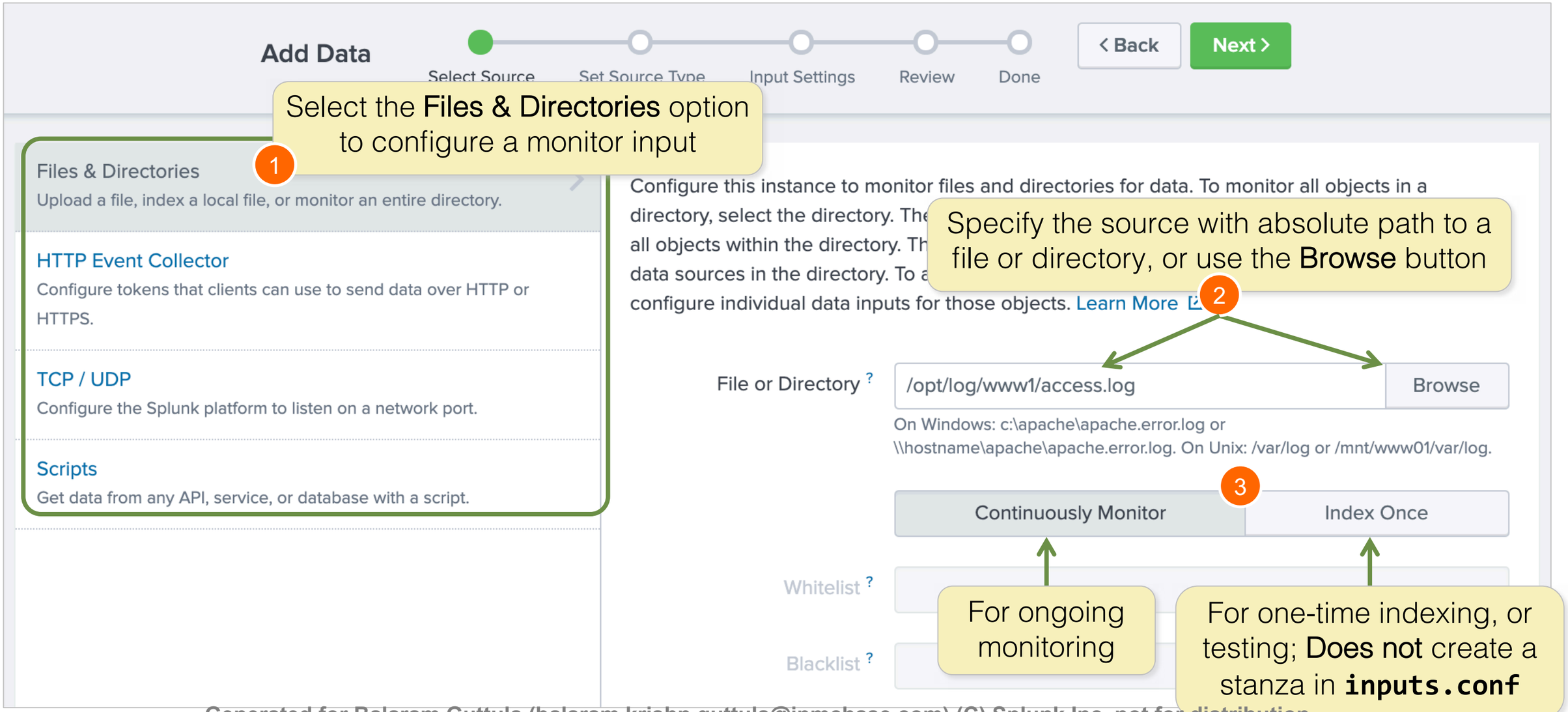
Monitor

- Provides one-time or continuous monitoring of files, directories, http events, network ports, or data gathering scripts located on Splunk Enterprise instances
- Useful for testing inputs

Forward

- Remote machines gather and forward data to indexers over a receiving port
- Main source of input in production environments
- Requires installation of forwarders

Select Source

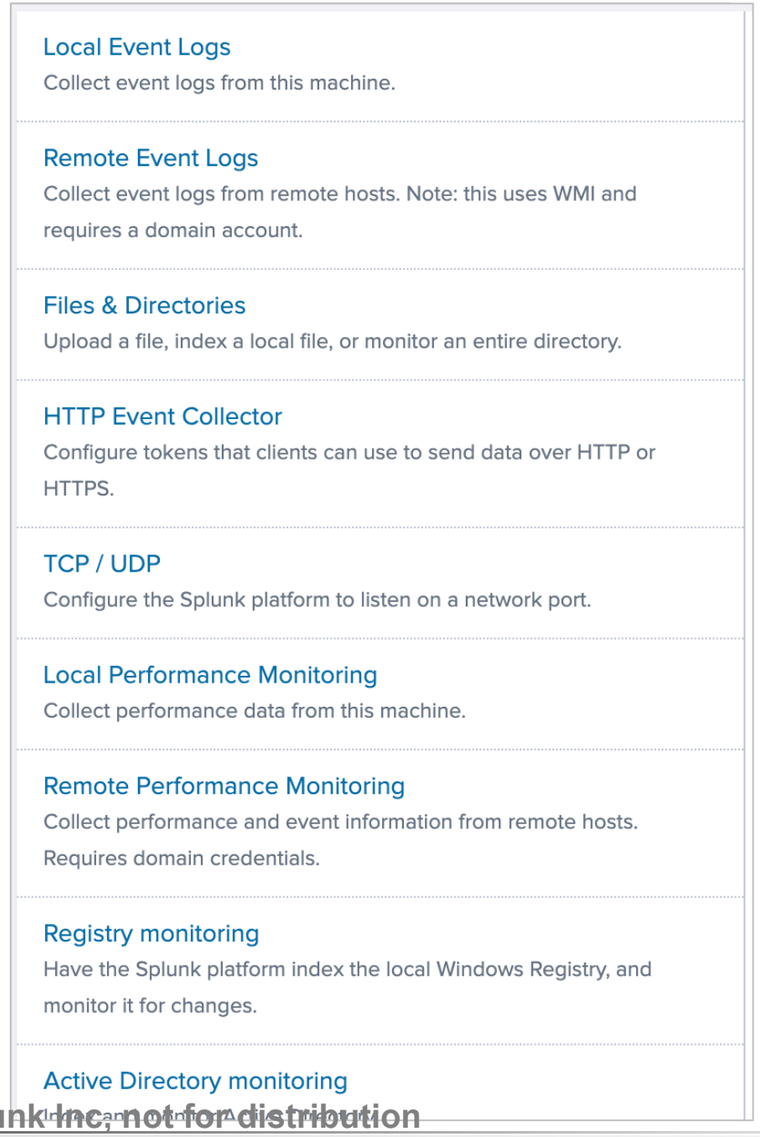


Select Source From Windows Instance

- On Windows Splunk instances, there are additional Windows-specific source options
- To monitor a shared network drive, enter the path manually:

*nix:	**<host>/<path>**
Windows:	**\\<host>\<path>**

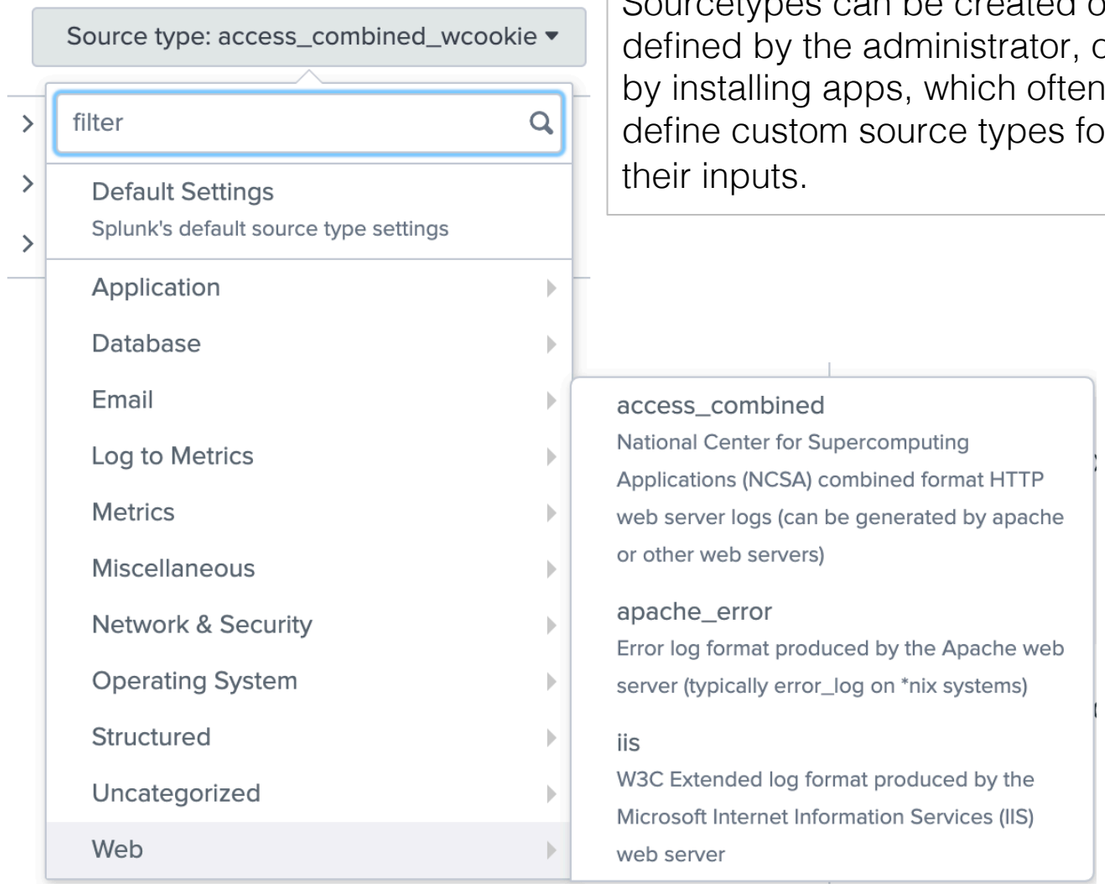
  - Make sure Splunk has read access to the mounted drive



Understanding Source Types

Source Types

- Splunk's way of categorizing data types
- Frequently used during index processes
- Used in searches, reports, apps, etc.
- Can be explicitly set with Splunk Web, CLI, or by modifying **`inputs.conf`**
- Assigned automatically when possible
- Can be set by administrators or apps

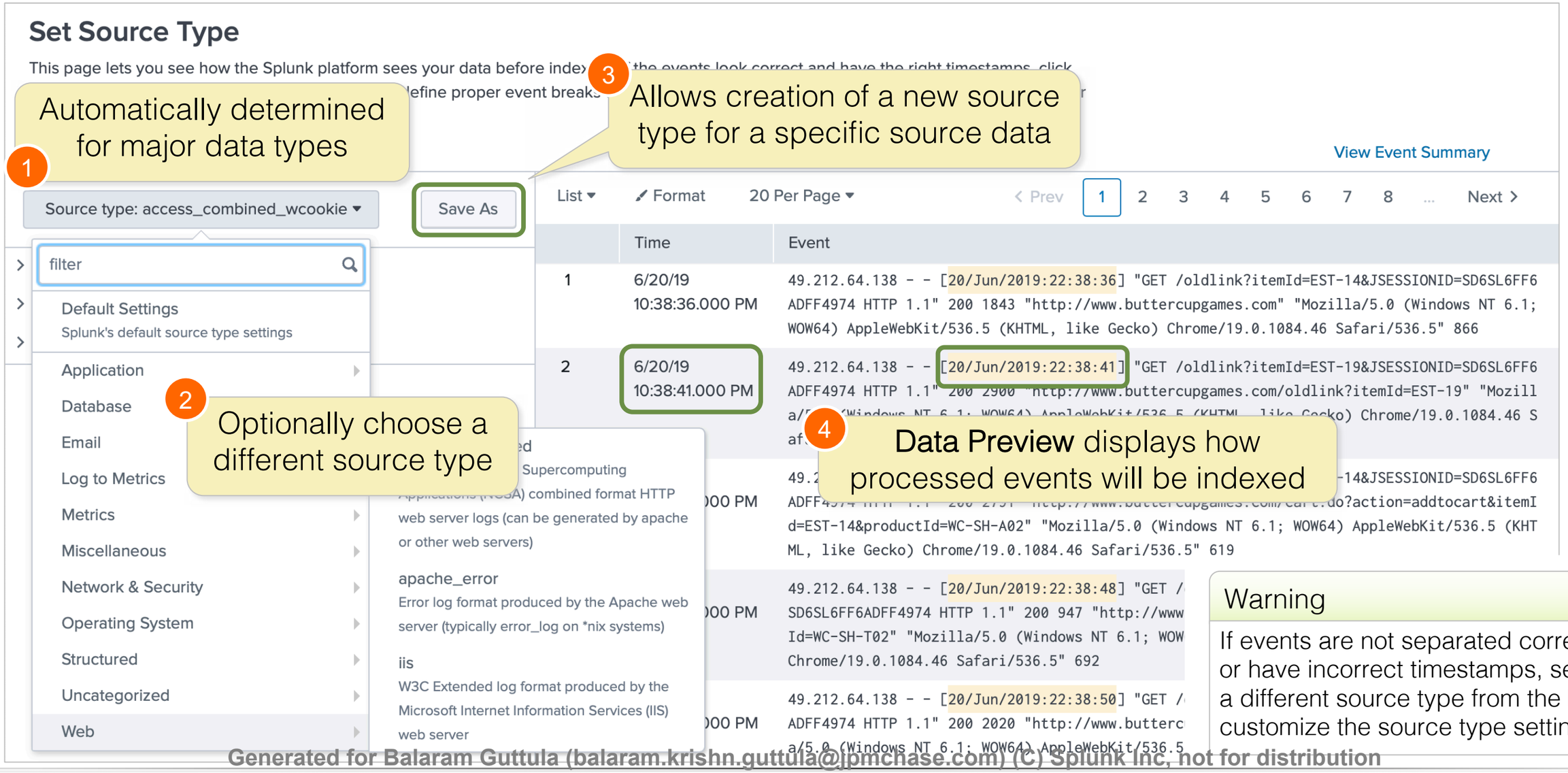


Note

Sourcetypes can be created or defined by the administrator, or by installing apps, which often define custom source types for their inputs.

 | 29 May 2020

Set Source Type (Data Preview Interface)



Pretrained Source Types

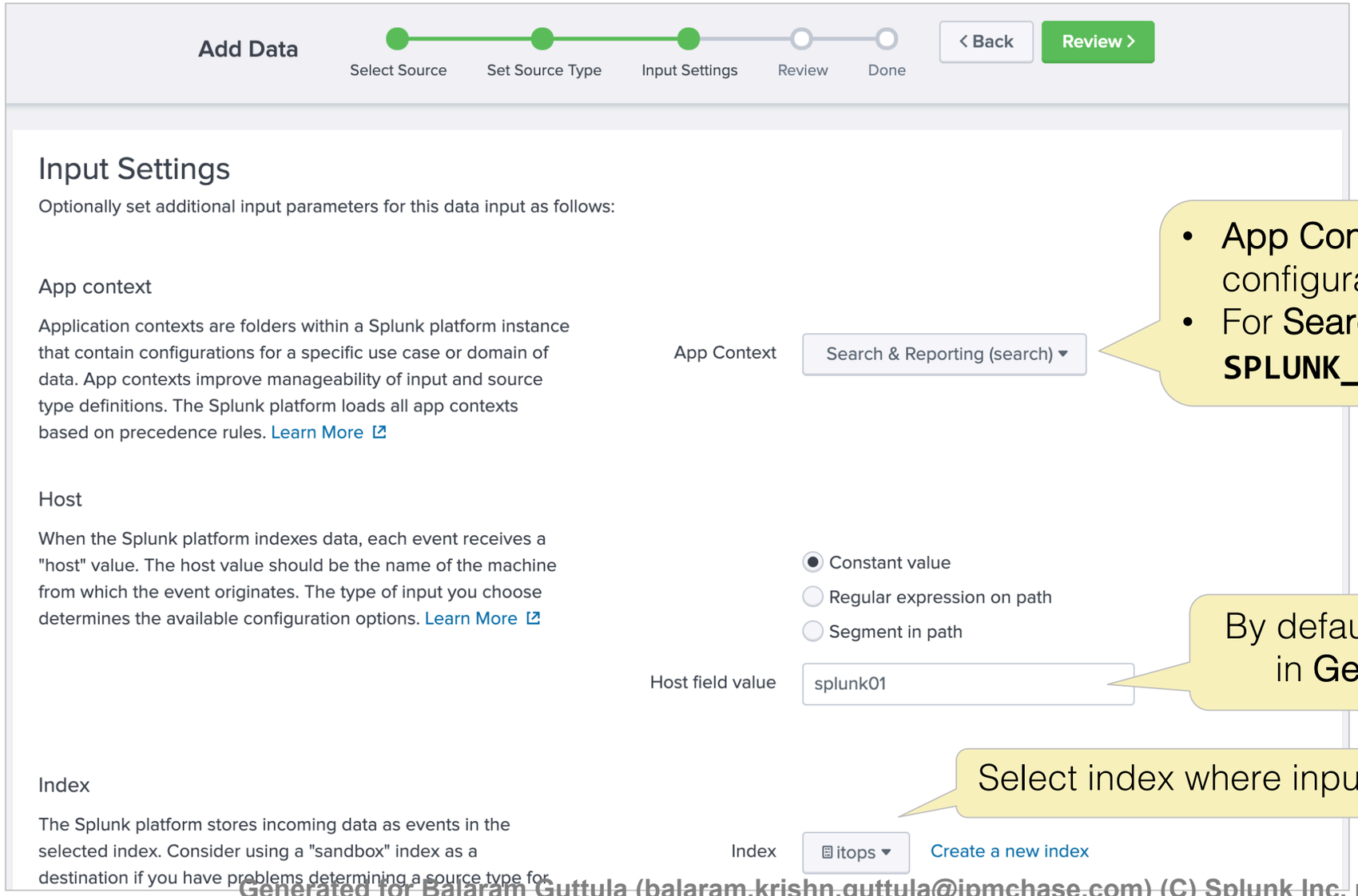
- Built-in source types shipped with Splunk
- Can be added to and defined by Splunk apps
- Listed in Splunk documentation:

  docs.splunk.com/Documentation/Splunk/latest/Data/Listofpretrainedsourcetypes

Automatically recognized source types [edit]

Source type name	Origin	Examples
access_combined	NCSA combined format http web server logs (can be generated by apache or other web servers)	`10.1.1.43 - webdev [08/Aug/2005:13:18:16 -0700] "GET / HTTP/1.0" 200 0442 "-" "check_http/1.10 (nagios-plugins 1.4)"`
access_combined_wcookie	NCSA combined format http web server logs (can be generated by apache or other web servers), with cookie field added at end	`"66.249.66.102.1124471045570513" 59.92.110.121 - - [19/Aug/2005:10:04:07 -0700] "GET /themes/splunk_com/images/logo_splunk.png HTTP/1.1" 200 994 "http://www.splunk.org/index.php/docs" "Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.7.8) Gecko/20050524 Fedora/1.0.4-4 Firefox/1.0.4" "61.3.110.148.1124404439914689"`
access_common	NCSA common format http web server logs (can be generated by	`10.1.1.140 - - [16/May/2005:15:01:52 -0700] "GET`

Input Settings



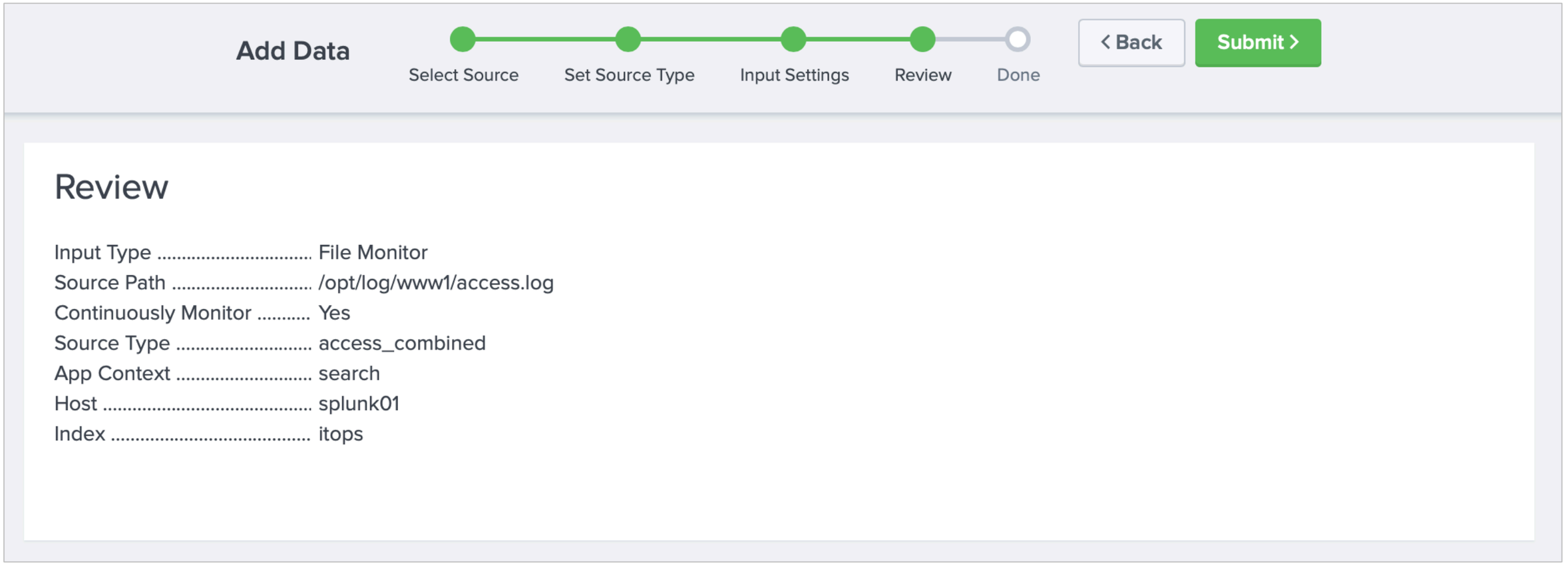
- **App Context** determines where input configuration is saved
- For **Search & Reporting (search)**: **`SPLUNK_HOME/etc/apps/search/local`**

By default, the **default host name** in **General settings** is used

Select index where input will be stored

Review

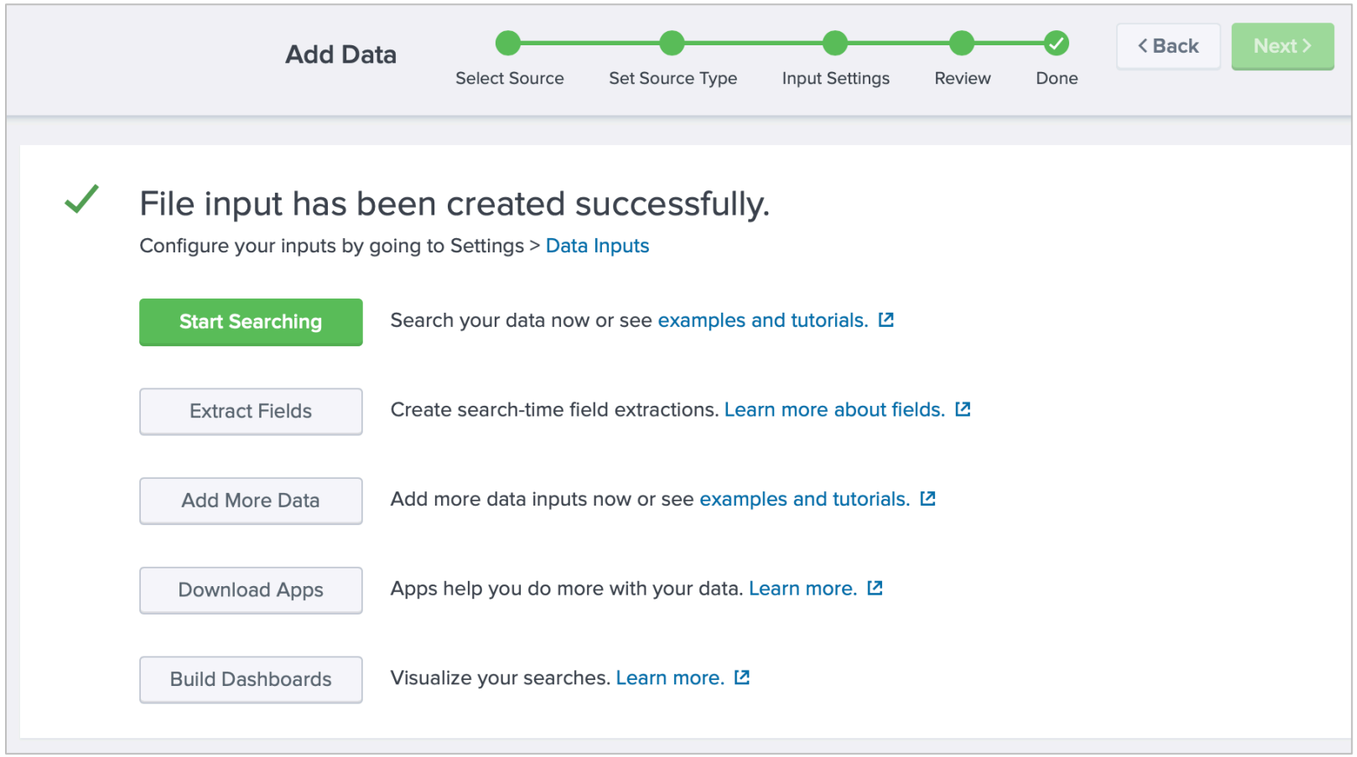
Review the input configuration summary and click **Submit** to finalize



What Happens Next?

- Indexed events are available for immediate search
  - Splunk may take a minute to start indexing the data
- You are given other options to do more with your data
- Input configuration is saved in:

SPLUNK_HOME/etc/apps/<*app*>/local/inputs.conf

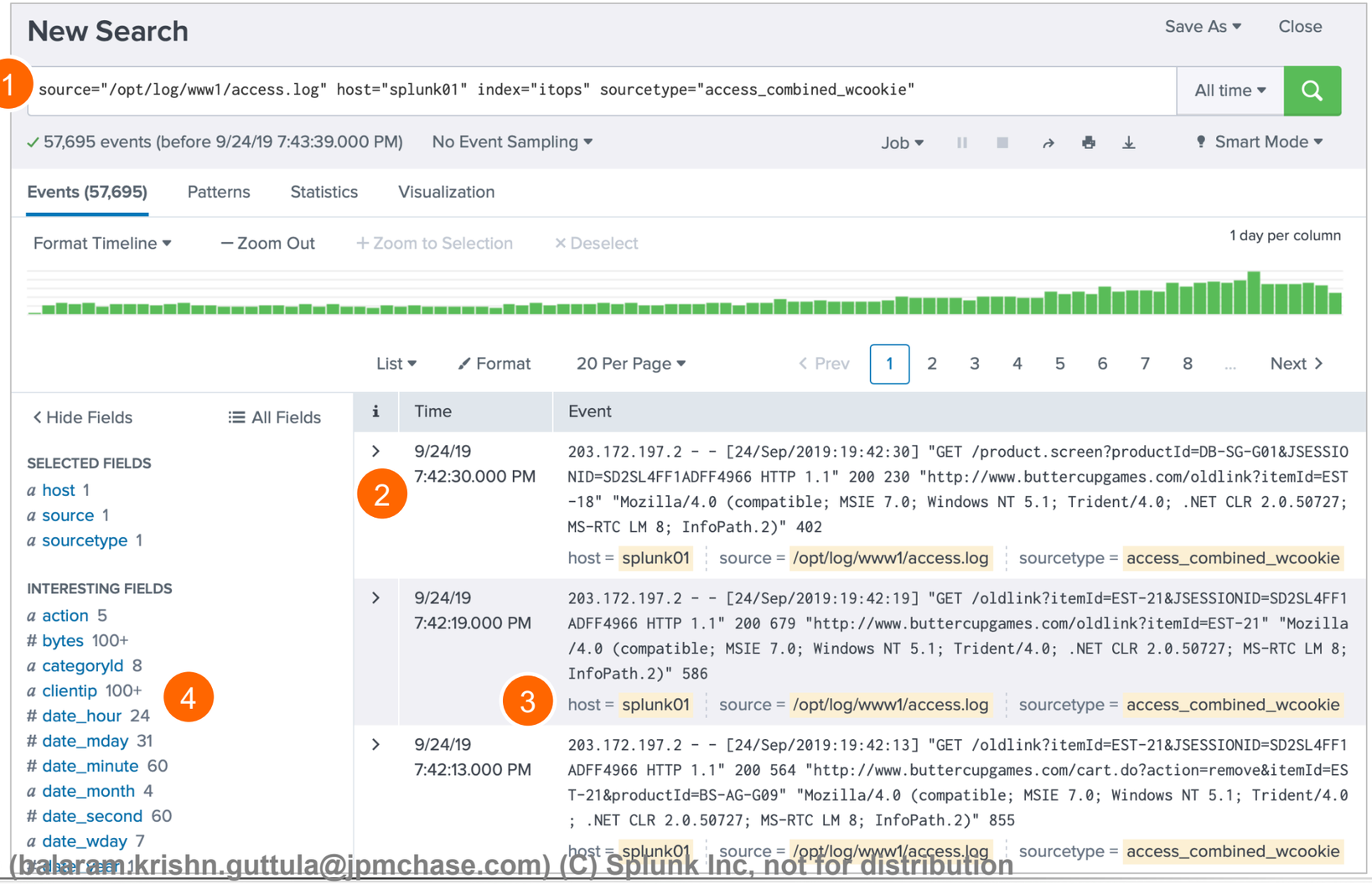


Note

Entries in the **inputs.conf** file are not created when **Upload** or **Index Once** is selected.

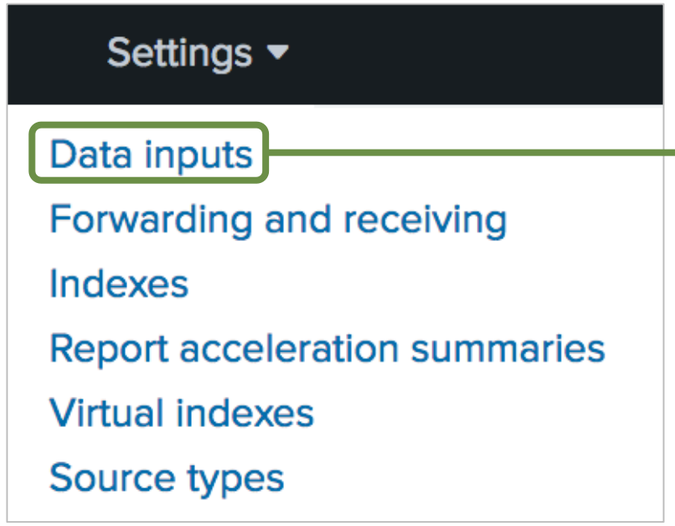
Verify your Input

1. Click **Start Searching** or search for **index=<*test_idx*>**
2. Verify the event timestamps
3. Confirm the **host**, **source**, and **sourcetype** field values
4. Check the auto-extracted field names



Viewing Configured Inputs

Select **Settings > Data Inputs**



Data inputs

Set up data inputs from files and directories, network ports, and scripted inputs. If you want to set up forwarding and receiving between two Splunk instances, go to Forwarding and receiving.

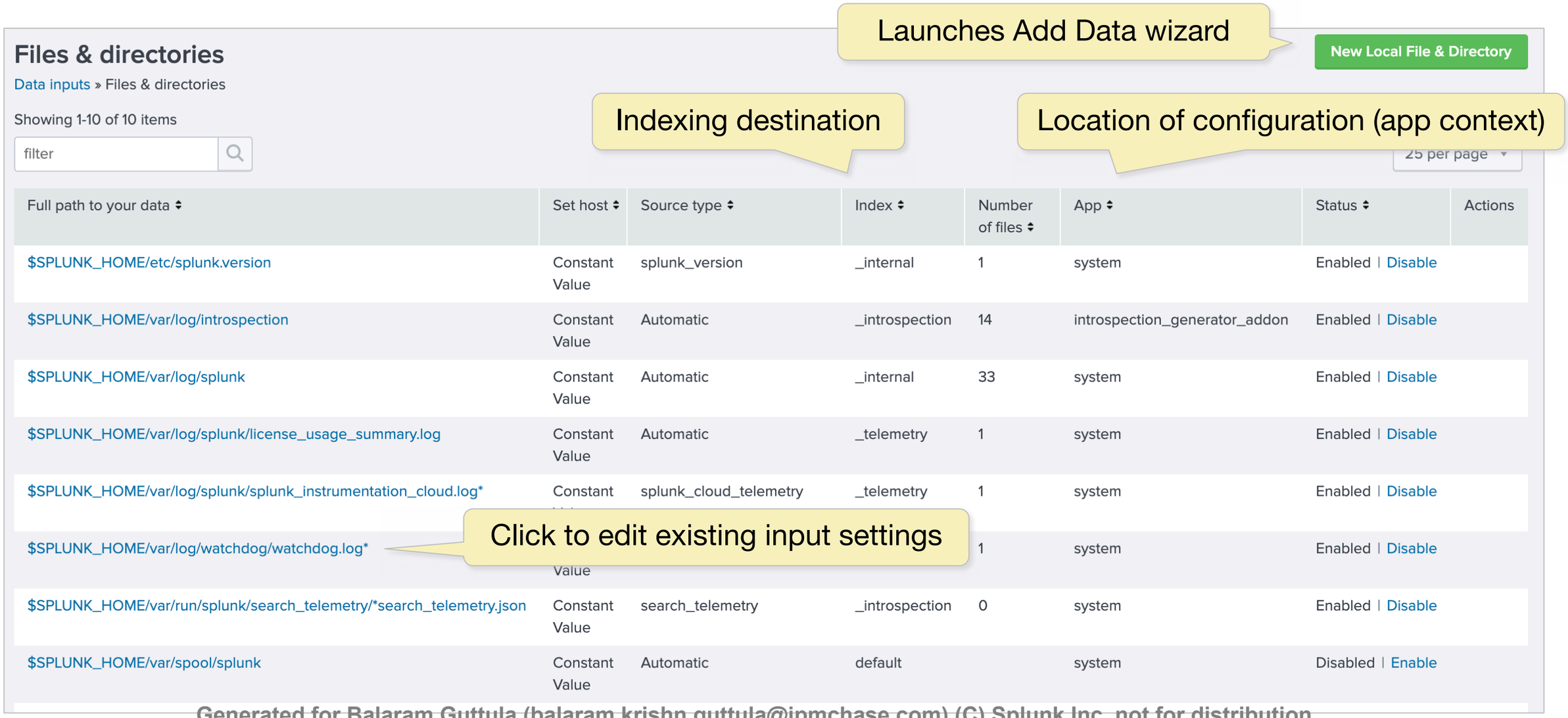
Local inputs — Inputs handled by this server

Type	Inputs	Actions
Files & Directories Index a local file or monitor an entire directory.	10	+ Add new
HTTP Event Collector Receive data over HTTP or HTTPS.	0	+ Add new
TCP Listen on a TCP port for incoming data, e.g. syslog.	0	+ Add new
UDP Listen on a UDP port for incoming data, e.g. syslog.	0	+ Add new
Scripts Run custom scripts to collect or generate more data.	5	+ Add new

Forwarded inputs — Inputs handled by remote instances but configured from this deployment server

Type	Inputs	Actions
Windows Event Logs Collect event logs from forwarders.	0	+ Add new

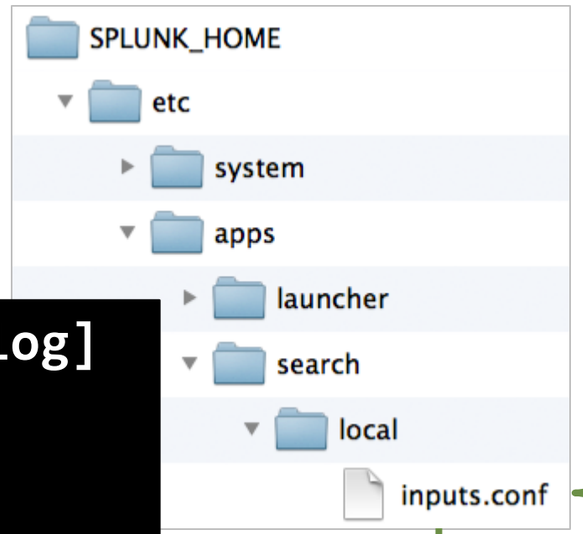
Viewing Configured Inputs: Files & Directories



Full path to your data	Set host	Source type	Index	Number of files	App	Status	Actions
$SPLUNK_HOME/etc/splunk.version	Constant Value	splunk_version	_internal	1	system	Enabled | Disable	
$SPLUNK_HOME/var/log/introspection	Constant Value	Automatic	_introspection	14	introspection_generator_addon	Enabled | Disable	
$SPLUNK_HOME/var/log/splunk	Constant Value	Automatic	_internal	33	system	Enabled | Disable	
$SPLUNK_HOME/var/log/splunk/license_usage_summary.log	Constant Value	Automatic	_telemetry	1	system	Enabled | Disable	
$SPLUNK_HOME/var/log/splunk/splunk_instrumentation_cloud.log*	Constant	splunk_cloud_telemetry	_telemetry	1	system	Enabled | Disable	
$SPLUNK_HOME/var/log/watchdog/watchdog.log*				1	system	Enabled | Disable	
$SPLUNK_HOME/var/run/splunk/search_telemetry/*search_telemetry.json	Constant Value	search_telemetry	_introspection	0	system	Enabled | Disable	
$SPLUNK_HOME/var/spool/splunk	Constant Value	Automatic	default		system	Disabled | Enable	

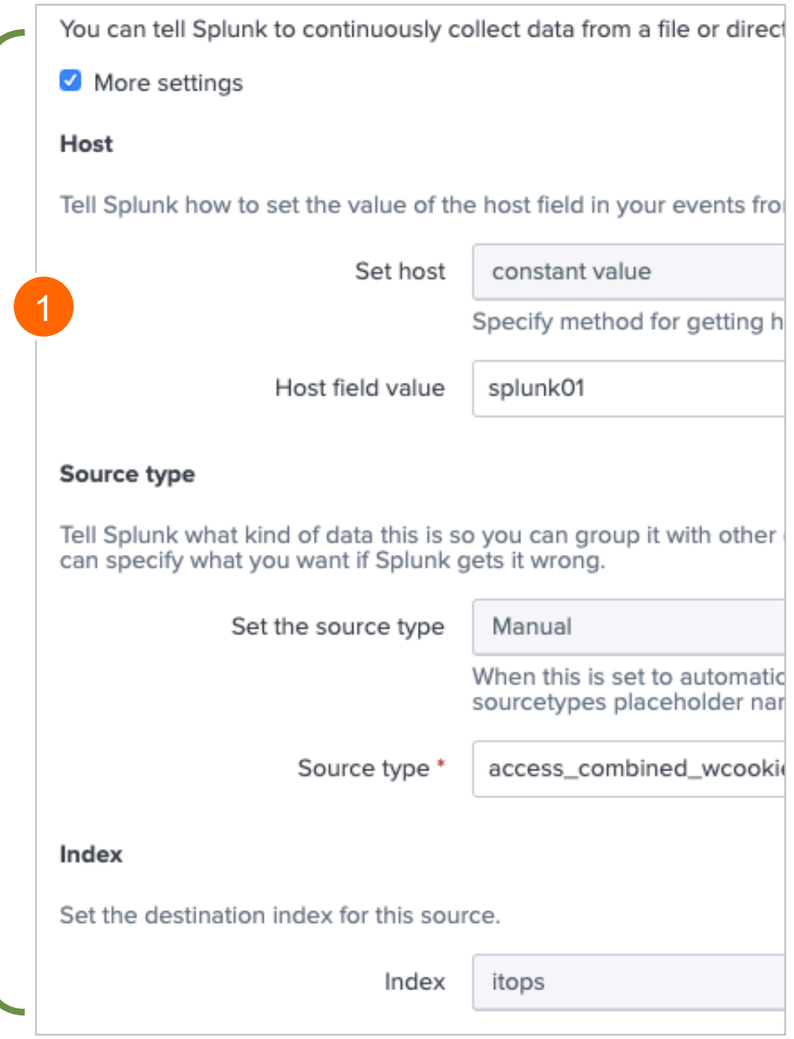
Viewing and Updating **inputs.conf**

1. Adding monitored data into Splunk updates the target app's **local/inputs.conf** file
2. Optionally edit **inputs.conf** manually




```
[monitor:///opt/log/www1/access.log]
disabled = false
host = splunk01 www_ca
index = itops websales
sourcetype = access_combined_wcookie
```

2



1

Module 2 Knowledge Check

- When you configure the inputs using **Settings > Add Data**, under what directory is the **`inputs.conf`** created?
- True or False. You cannot change the sourcetype when you go through the **Settings > Add Data** wizard.
- True or False. Splunk will not create an **`inputs.conf`** file when you use the **Upload** option in **Settings > Add Data**.

Module 2 Knowledge Check – Answers

- When you configure the inputs using **Settings > Add Data**, under what directory is the **`inputs.conf`** created?

  It depends on the **App Context** setting on the **Input Setting** stage. Best practice is to put the configuration file in the local directory of your app. If you have clustering enabled, then the **`SPLUNK_HOME/etc/system/local`** may not be the highest in the precedence order. More details are available in the Cluster Administration course.

- True or False. You cannot change the sourcetype when you go through the **Settings > Add Data** wizard.

  False. You can change the source type from the dropdown. In fact, you can even create a new source type. We will learn how to do this in Module 9.

- Splunk will not create an **`inputs.conf`** file when you use the **Upload** option in **Settings > Add Data**.

  True. Upload is a one-time process, so Splunk does not create an **`inputs.conf`**.

Module 2 Lab Exercise – Add a Local Data Input

Time: 20 minutes

Tasks:

- Create all local indexes required on the deployment/test server
- Index a file on the deployment server
- Verify the indexed events with their metadata values
- View the stanza in the saved **`inputs.conf`** file

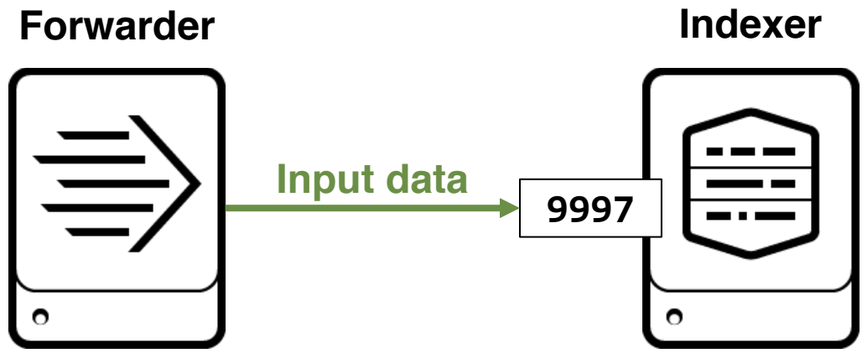
Module 3: Forwarder Configuration

Module 3 Objectives

- Understand the role of production indexers and forwarders
- Understand the functionality of Universal Forwarders
- Configure forwarders
- Identify additional forwarder options

Forwarders and Indexers

- In production environments:
  - Indexers run on dedicated servers
  - Most input data is on remote servers
- Install **forwarders** on remote servers to
  - Gather the data
  - Send it over the network to indexers
- Configure indexers to listen on a **receiving port** for the forwarded data

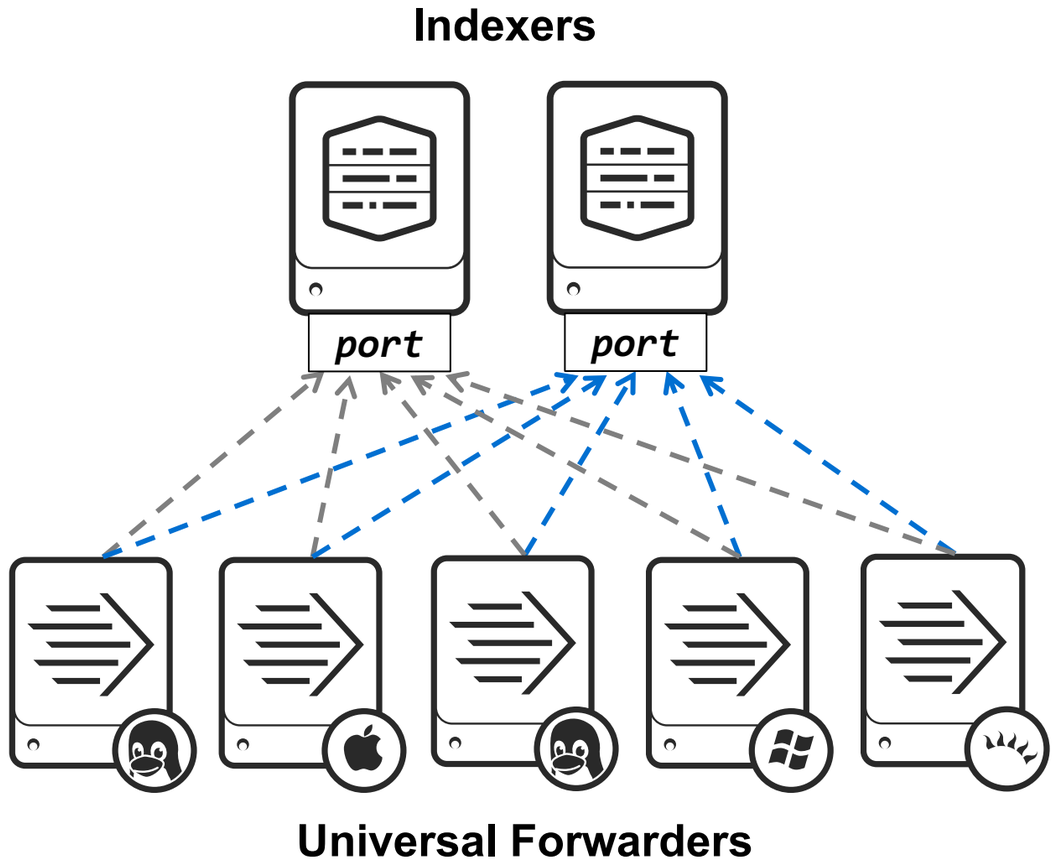


Understanding Universal Forwarders

Universal Forwarders (UF)

- Gathers data from a host, sending it over the network to receiving ports on indexers
- Provided as separate installation binary with a built-in license (no limits)
- Designed to run on production servers

  (minimal CPU / memory use, bandwidth constrained to 256 KBps by default, no web interface, cannot search or index)



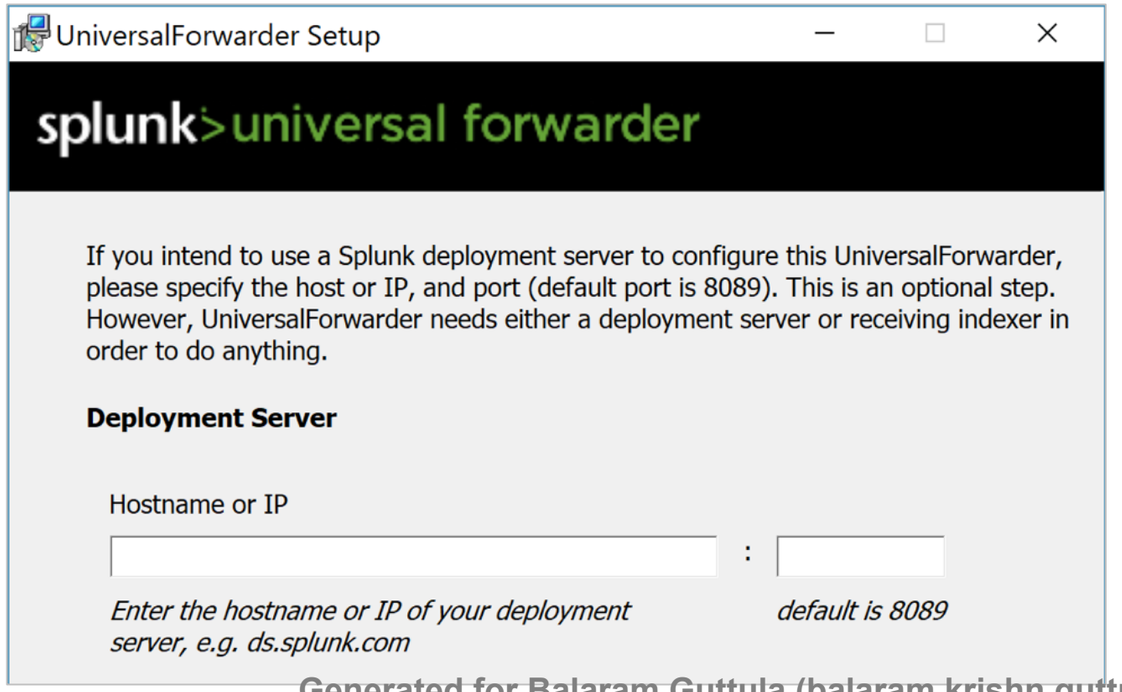
Installing Splunk Universal Forwarder

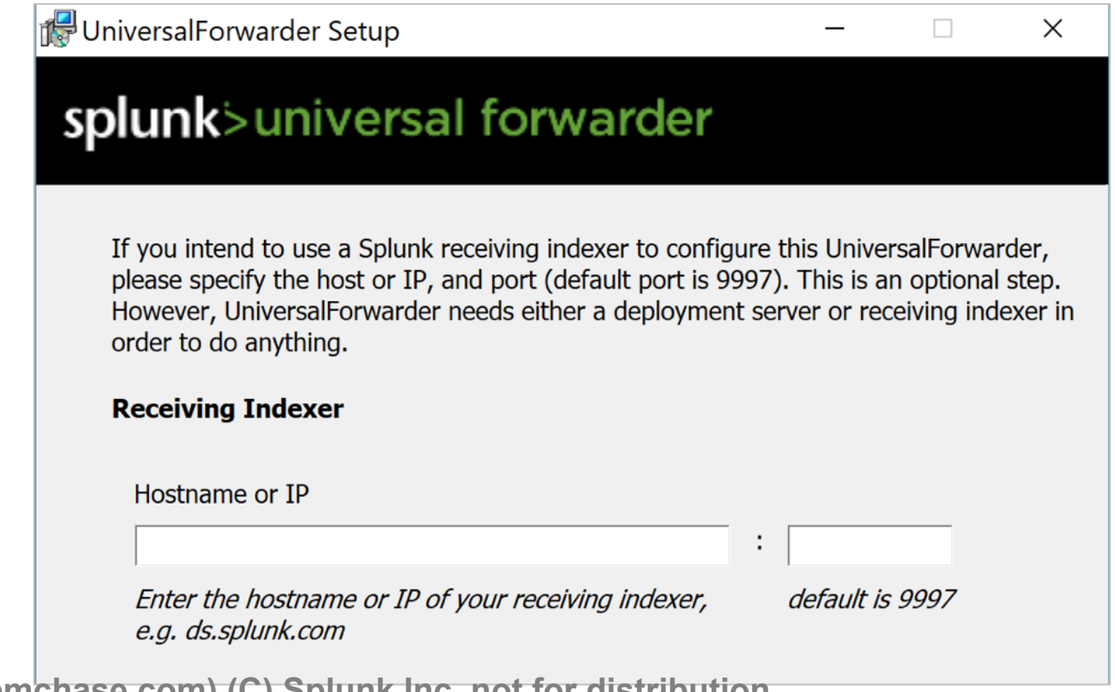
	*NIX	Windows
Download	www.splunk.com/en_us/download/universal-forwarder.html	
Install	• Un-compress **.tgz**, **.rpm**, or **.deb** file in the path Splunk will run from • Default **SPLUNK_HOME** is: **/opt/splunkforwarder**	• Execute **.msi** installer (or use the CLI) • Default **SPLUNK_HOME** is: **C:\Program Files\SplunkUniversalForwarder**

- Same **splunk** command-line interface in **SPLUNK_HOME/bin**
  - Same commands for start/stop, restart, etc.
  - An admin account and password are required
- When installing large numbers of forwarders, use an automated method

Using the Interactive Windows Installer

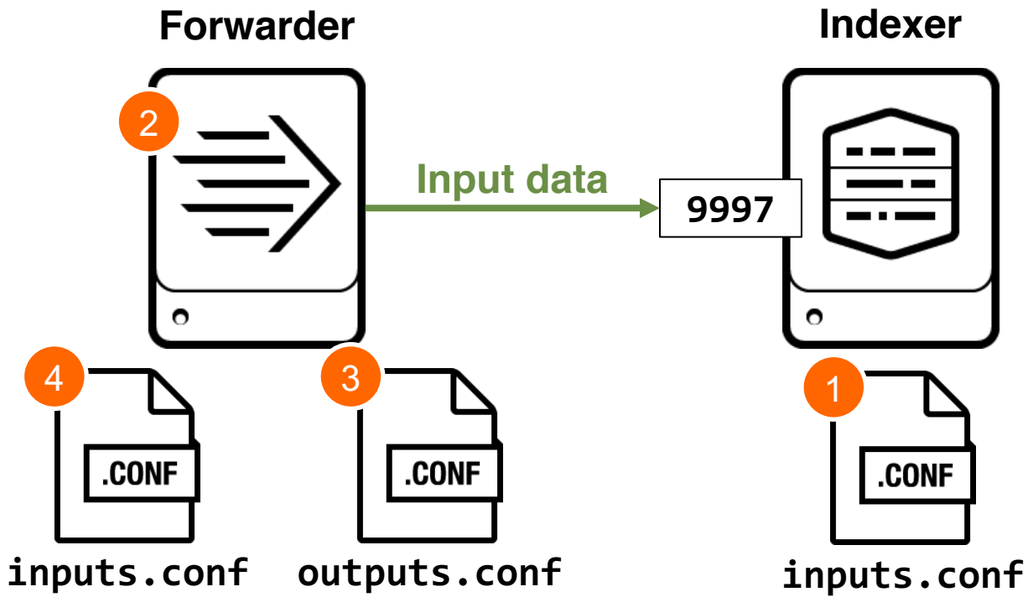
- Most forwarder settings can be configured using the installer wizard
  - Can run as a local or domain user without local administrator privileges
- CLI installation is available for scripted installations
  docs.splunk.com/Documentation/Forwarder/latest/Forwarder/InstallaWindowsuniversalforwarderfromthecommandline





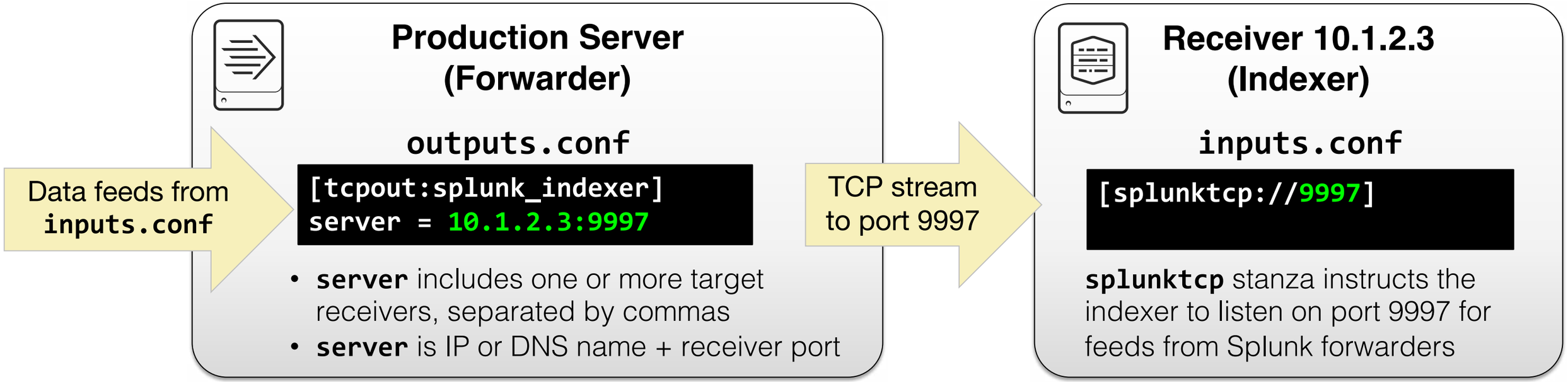
Universal Forwarder Configuration Steps

1. Set up a receiving port on each indexer
   - Task only needs to be performed once
2. Download and install Universal Forwarder
3. Set up forwarding on each forwarder, by either:
   - Editing **`outputs.conf`** manually
   - Using Deployment Server
4. Add inputs on forwarders, by either:
   - Editing **`inputs.conf`** manually
   - Using Deployment Server
   - Running Splunk commands (CLI)



Forwarder **outputs.conf** File

- Points the forwarder to the receivers
- Can specify additional options for load balancing, SSL, compression, alternate indexers, and indexer acknowledgement



Defining Target Indexers on the Forwarder

- Execute on the forwarder for each destination indexer:
  splunk add forward-server <*indexer:receiving_port*>
  - For example, **splunk add forward-server 10.1.2.3:9997** configures the **outputs.conf** as:

```
[tcpout]
defaultGroup = default-autolb-group

[tcpout-server://10.1.2.3:9997]

[tcpout:default-autolb-group]
disabled = false
server = 10.1.2.3:9997
```

docs.splunk.com/Documentation/Forwarder/latest/Forwarder/Configureforwardingwithoutputs.conf

Configuration and Connection Validation

- After running **`splunk add forward-server`**:
  - Forwarder should be communicating with the indexer
  - Forwarder Splunk logs are automatically sent to indexer's **`_internal`** index
- To check the configuration:
  - On indexer, run: **`splunk display listen`**
  - On forwarder, run: **`splunk list forward-server`**
- To verify successful connection:
  - Search: **`index=_internal host=<forwarder_hostname>`**
- To remove the target indexer setting:
  - On forwarder, run: **`splunk remove forward-server <indexer:port>`**

Troubleshooting Forwarder Connection

- Is the forwarder sending data to the indexer?
  - Check **SPLUNK_HOME/var/log/splunk/splunkd.log** on forwarder:

```
tail -f var/log/splunk/splunkd.log | egrep 'TcpOutputProc|TcpOutputFd'
```

- Does the indexer receive any data on the listening port?
  - Search on indexer:

```
index=_internal sourcetype=splunkd component=TcpInputConfig OR
(host=<uf> component=StatusMgr)
```

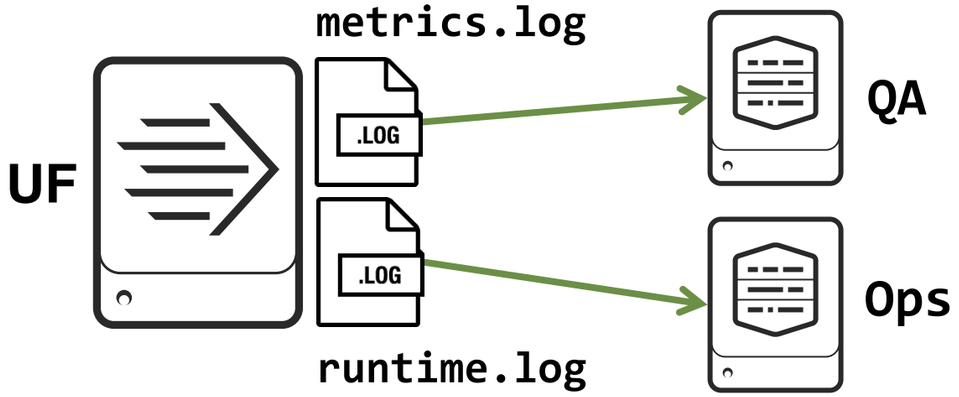
  - To get the hostname **<uf>**, run on the forwarder:

```
splunk show default-hostname
```

- Check the configuration files

Selectively Forwarding Data to Indexers

- Universal forwarder can route based on sources
  - Define multiple tcpout stanzas in **outputs.conf**
  - Specify **_TCP_ROUTING** identifying the **tcpout** stanza names in each source in **inputs.conf**
- Example:
  - QA team wants **metrics.log** sent to the QA team's indexer and Ops team wants **runtime.log** sent to the operations indexer



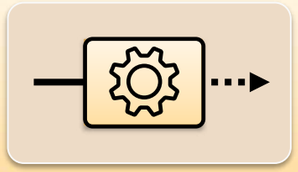

```
[tcpout:QA]
server=srv.qa:9997

[tcpout:Ops]
server=srv.ops:9997
```

outputs.conf

```
[monitor://path/metrics.log]
_TCP_ROUTING = QA

[monitor://path/runtime.log]
_TCP_ROUTING = Ops
```

inputs.conf

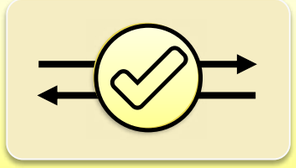
Additional Forwarding Options



Compressing the feed

Securing the feed

Automatic load balancing to multiple indexers

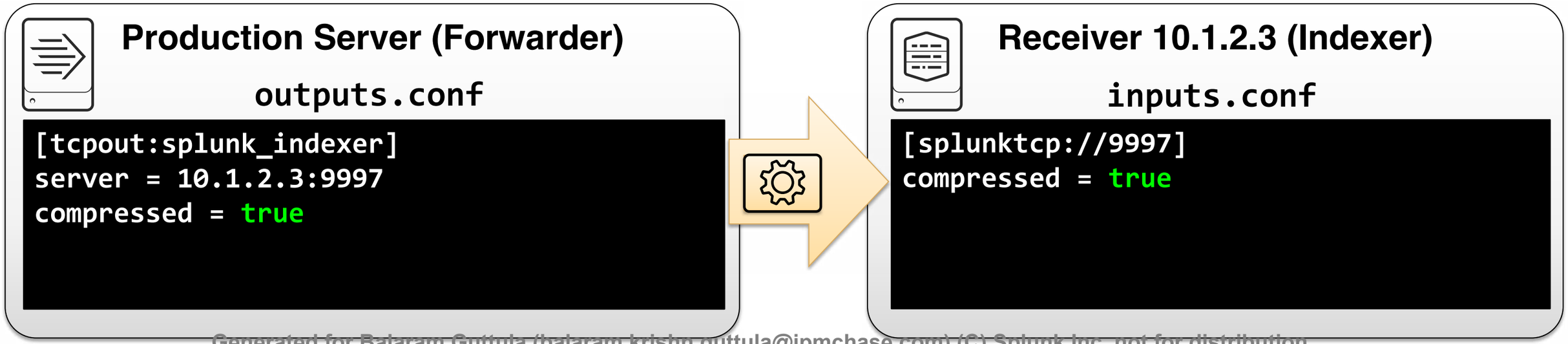


Indexer acknowledgement to forwarder

Forwarder queue size

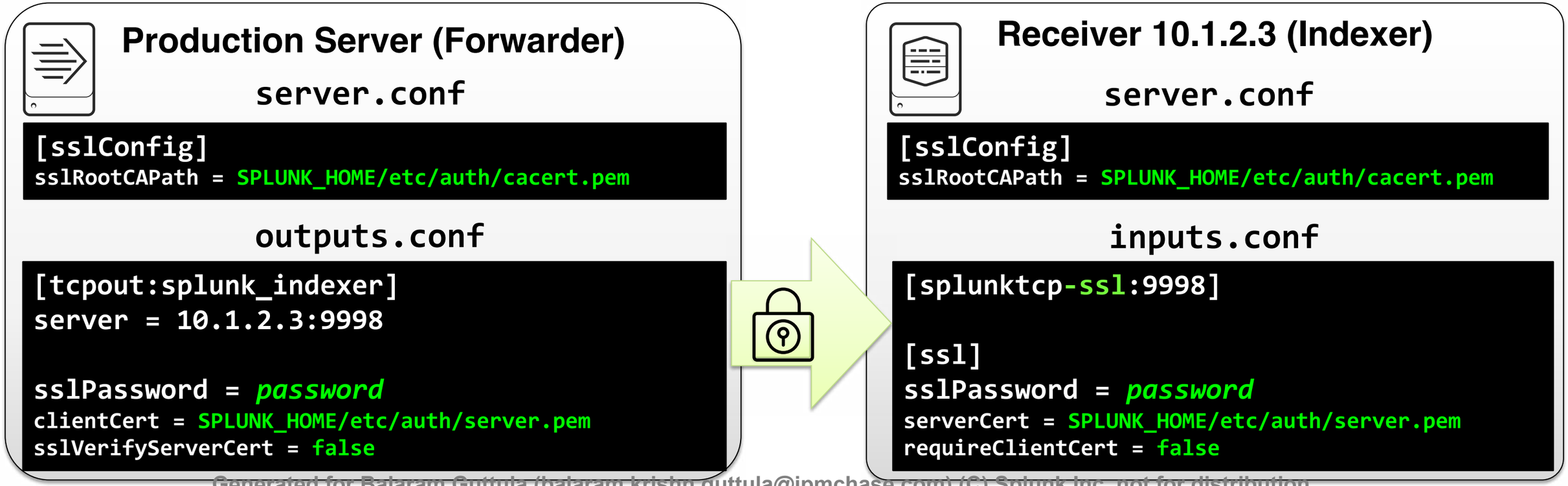
Compressing the Feed

- Reduces network utilization
- Increases CPU utilization slighty
- Compression can be set at either the forwarder or the indexer
  - Compress select feeds by setting compression on the forwarder
  - Compress all feeds by setting compression on the indexer



Securing the Feed with SSL

- Encrypts the feed
- Automatically compresses the feed
- Increases CPU utilization



Notes About SSL

- Splunk uses OpenSSL to generate its default certificates
  - Default certificate password is **`password`**
- Use external certs *or* create new ones using Splunk's OpenSSL
- Refer to:

docs.splunk.com/Documentation/Splunk/latest/Security/AboutsecuringyourSplunkconfigurationwithSSL

docs.splunk.com/Documentation/Splunk/latest/Security/Aboutsecuringdatafromforwarders

docs.splunk.com/Documentation/Splunk/latest/Security/ConfigureSplunkforwardingtousethedefaultcertificate

docs.splunk.com/Documentation/Splunk/latest/Security/ConfigureSplunkforwardingtousesignedcertificates

wiki.splunk.com/Community:Splunk2Splunk_SSL_DefaultCerts

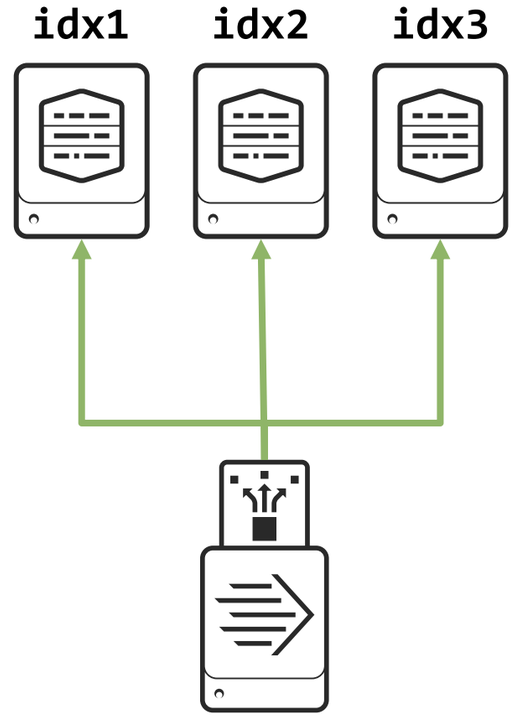
wiki.splunk.com/Community:Splunk2Splunk_SSL_SelfSignedCert_NewRootCA

Automatic Load Balancing

- Configured in forwarder's **outputs.conf** using static list target:

```
[tcpout:my_LB_indexers]
server = idx1:9997,idx2:9997,idx3:9997
```

- Causes forwarder to split data between multiple indexers
- Switching indexers is performed:
  - By time, every **autoLBFrequency** seconds (default: 30 sec.)
  - By volume, every **autoLBVolume** bytes (default: 0 = disabled)
  - When it is safe for the data stream (e.g. an **EOF** is detected)
  - When a receiving indexer goes down



docs.splunk.com/Documentation/Splunk/latest/Forwarding/Setuploadbalancingd

Defining Event Boundary on UF

- Event boundaries
  - Detecting when one event ends and another starts
  - Normally determined on the indexer (where data is parsed)
- UF switches safely when:
  - An **`EOF`** (End of File) is detected
  - There is a short break in I/O activity
- Potential side effects
  - Streaming data (**`syslog`**) can prevent a UF from switching
  - A multi-line data (**`log4j`**) can result in event splits
    - Especially if the application has pauses in writing its log file
- Solution:
  - Enable event breaker on the UF per sourcetype

Defining Event Boundary on UF (cont.)

- Add the event breaker settings on UF per sourcetype in **`props.conf`**
  - Single line event

```
[my_syslog]
EVENT_BREAKER_ENABLE = true
```

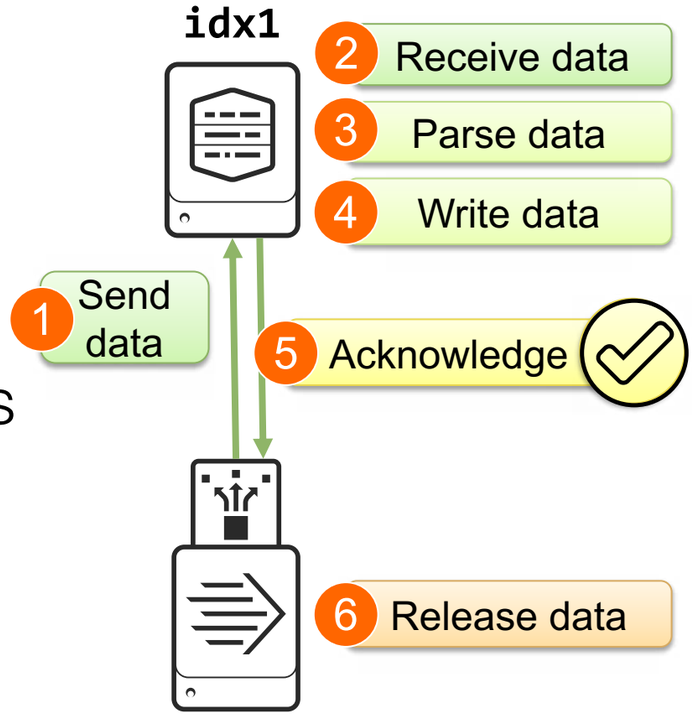
  - Multi-line event

```
[my_log4j]
EVENT_BREAKER_ENABLE = true
EVENT_BREAKER = ([\r\n]+)\d\d\d\d-\d\d-\d\d
```

docs.splunk.com/Documentation/Forwarder/latest/Forwarder/Configureloadbalancing

Indexer Acknowledgement

- Disabled by default (**`useACK=false`**)
- Enabled with **`useACK=true`** in **`outputs.conf`**
- Guards against loss of forwarded data
  1. **Forwarder**: Send data to indexer
  2. **Indexer**: Receives the block of data
  3. **Indexer**: Parses the data
  4. **Indexer**: Writes the data to the file system as events
  5. **Indexer**: Sends acknowledgement to the forwarder
  6. **Forwarder**: Releases data from memory
     - If no acknowledgement is received from indexer, forwarder instead resends the data

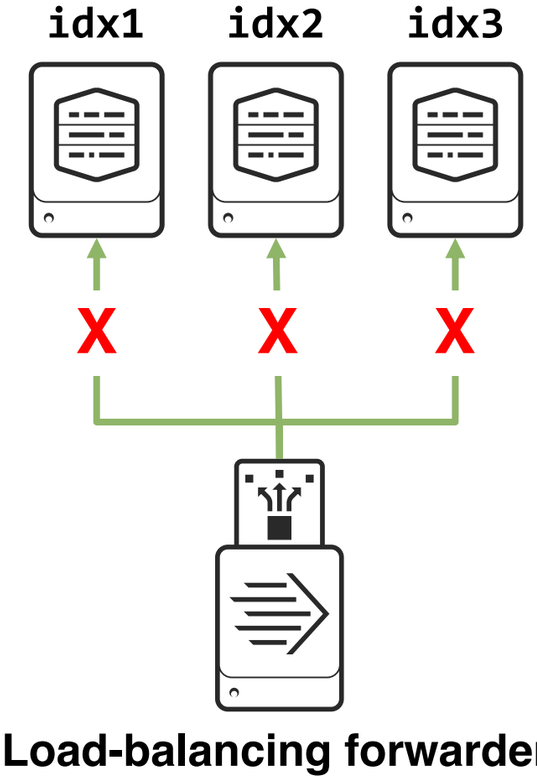


docs.splunk.com/Documentation/Splunk/latest/Forwarding/Protectagainstlossofin-flightdata

Forwarder Queue Size

- When forwarder can't reach an indexer, forwarder automatically switches to another indexer
- When forwarder can't reach **any** indexer, data is queued on the forwarder
- Output and wait queue sizes are affected by **`maxQueueSize`** and **`useACK`** in **`outputs.conf`**
  - Default: **`maxQueueSize=auto`**

maxQueueSize=	useACK=	Output queue	Wait queue
auto	false	500 KB	-
auto	true	7 MB	21 MB
20MB	true	20 MB	60 MB



Load-balancing forwarder

Forwarding Resources

- Overview of forwarders

  docs.splunk.com/Documentation/Splunk/latest/Data/Usingforwardingagents

- Forwarder deployment overview

  docs.splunk.com/Documentation/Splunk/latest/Forwarding/Aboutforwardingandreceivingdata

- Overview of enterprise installation
  - Link at the bottom of the web page has example install packages and Windows install

  wiki.splunk.com/Deploying_Splunk_Light_Forwarders

Useful Commands

Command	Operation
From the Forwarder	
`splunk add forward-server`	Configures the forwarder to connect the receiving indexer
`splunk list forward-server`	Displays the current receiving indexer configuration
`splunk remove forward-server`	Removes the receiving indexer from the forwarder
From the Receiver	
`splunk enable listen`	Configures the Splunk receiving port number
`splunk display listen`	Displays the current Splunk receiving port number

Module 3 Knowledge Check

- If the forwarder is set to send its data to 2 indexers at 30 second intervals, does it switch exactly at the 30th second?
- True or False. Turning SSL on between the forwarder and the receiver automatically compresses the feed.
- What configuration file on the forwarder defines where data is to be forwarded to?

Module 3 Knowledge Check - Answers

- If the forwarder is set to send its data to 2 indexers at 30 second intervals, does it switch exactly at the 30th second?

  Not always. The forwarder does not want to send half an event to indexer1 and the other half to indexer2. To avoid this situation, for example, if the forwarder is tailing a file, then it waits for an EOF or a pause in IO activity before it switches.

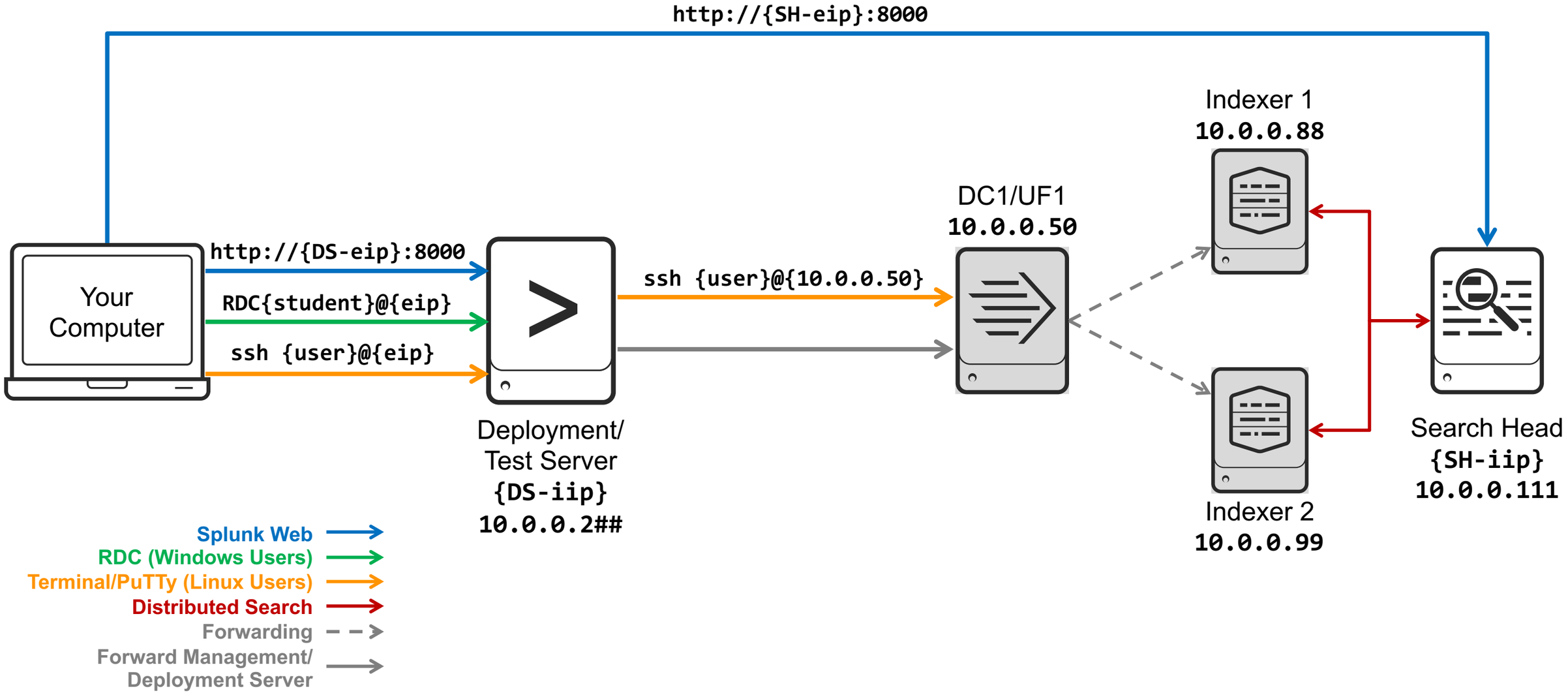
- True or False. Turning SSL on between the forwarder and the receiver automatically compresses the feed.

  True

- What configuration file on the forwarder defines where data is to be forwarded to?

  outputs.conf

Module 3 Lab Exercise – Environment Diagram



Module 3 Lab Exercise – Setting Up Forwarders

Time: 20 – 25 minutes

Tasks:

- Configure forwarder to send data to the Indexer 1 (**`10.0.0.88`**) and Indexer 2 (**`10.0.0.99`**)
- Confirm forwarder connection from your search head

Lab notes:

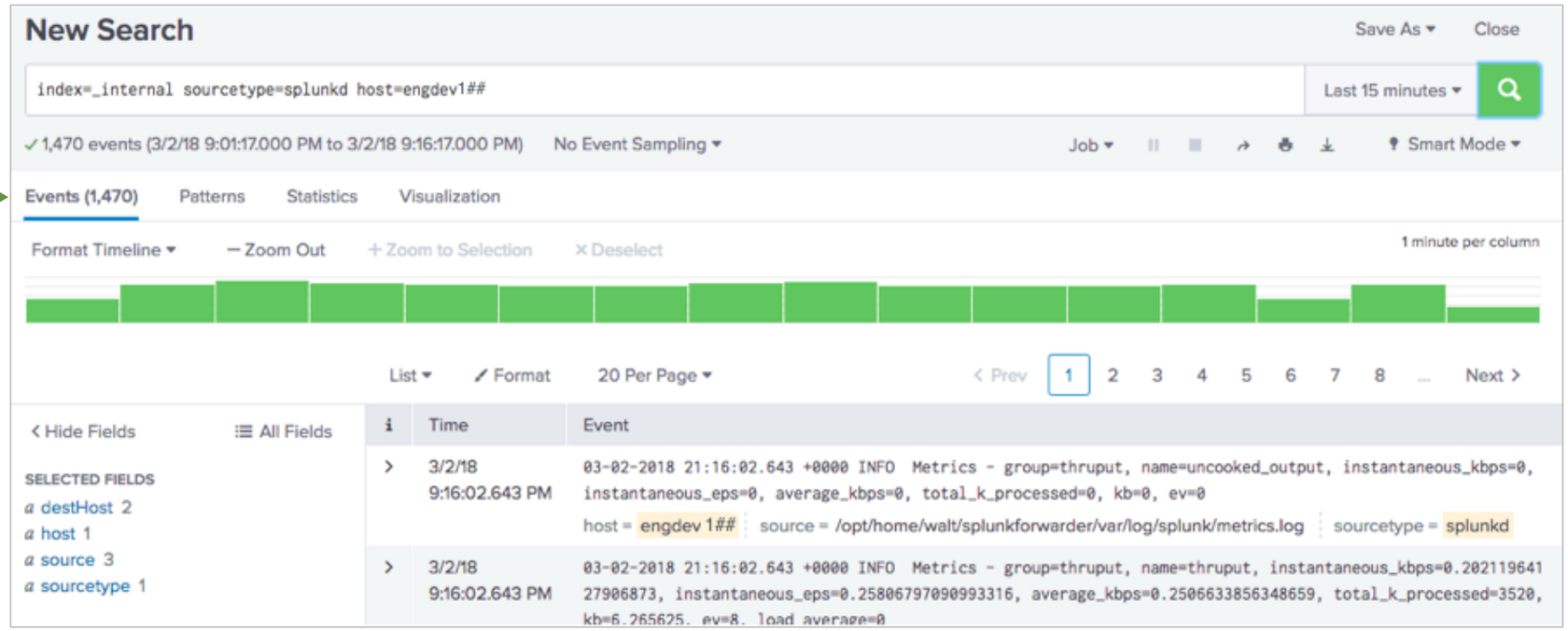
- You have a login on a remote Linux host that is your forwarder
- This lab exercise only establishes the connection between your UF and Indexer

Module 3 Lab Exercise – Setting up Forwarders (cont.)

Verification: Run a search to get forwarded internal logs from UF#1

`index=_internal sourcetype=splunkd host=engdev1##`

`http://{SH-IP}:8000`



Module 4: Heavy Forwarders & Forwarder Management

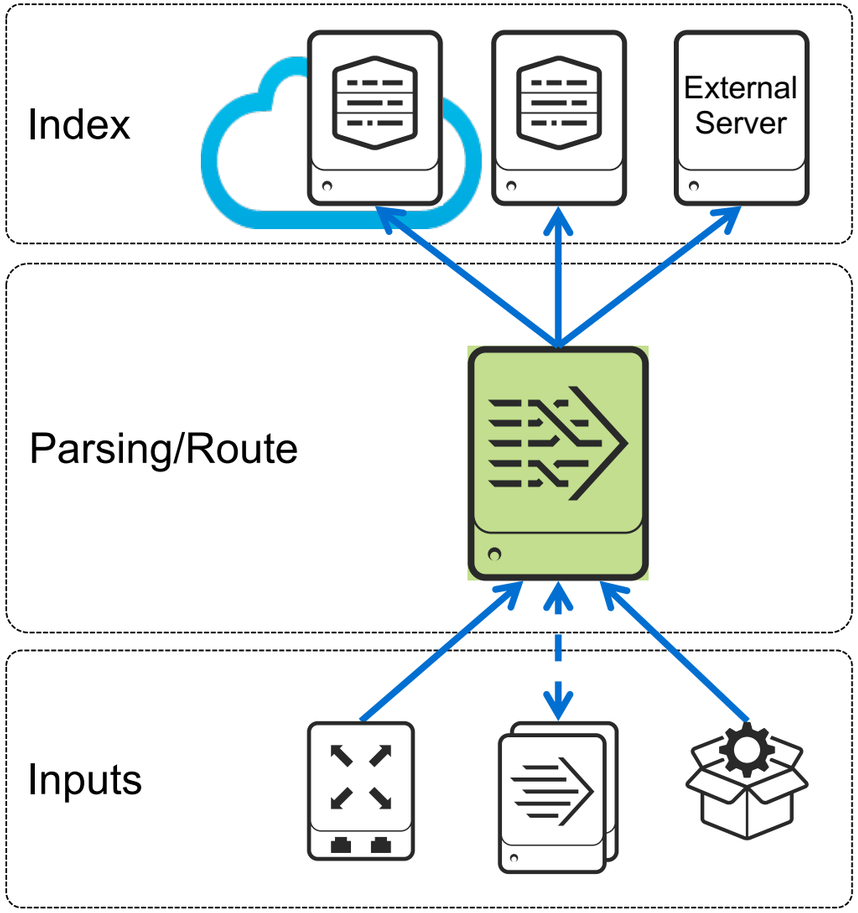
Module Objectives

- Introduction to the Heavy Forwarder (HF)
  - Describe what the heavy forwarder is and use cases
  - Perform heavy forwarder configuration
  - Deploy an app to the heavy forwarder
- Using Splunk Forwarder Management
  - Describe Splunk Deployment Server (DS)
  - Manage forwarders using deployment apps
  - Configure deployment clients and client groups
  - Monitor forwarder management activities

Understanding Heavy Forwarders (HF)

Heavy Forwarders (HF)

- Splunk Enterprise instance with the Forwarder License enabled
- Can parse data before forwarding it
- Can route data based on event criteria to different indexers or 3rd party receivers
- Cannot perform distributed searches



Deciding Between UF and HF

Universal Forwarder

- Ideal for most circumstances, including collecting files or as intermediate forwarder
- Minimal footprint on production servers
- Generally requires less bandwidth and has faster processing than same data on HF
- Supports simple routing or cloning data to separate indexers
- Does not support filtering based on regular expressions

vs.

Heavy Forwarder

- Able to do all UF tasks, and more
- Required by some apps, add-ons, or input types (such as HEC, DBconnect)
- Provides Splunk Web, if needed
- Supports complex, event-level routing to separate indexers or indexer clusters
- Can anonymize or mask data before forwarding to an indexer
- Predictable version of Python

HF Configuration – Inputs

- HF can receive data from other Splunk instances and UFs
- Setup can be performed using either:
  - CLI: **`splunk enable listen <port>`**
  - Deploying an **`inputs.conf`** file from the deployment server with the following stanza:

```
[splunktcp://<port>]
```

Note

During the lab exercises, you will manually set up the HF port using CLI.

HF Configuration – Outputs

- To configure the HF to forward the data to the indexers use either:
  - CLI to set up forwarding manually:

    splunk add forward-server:<*indexer:listening_port*>
  - Deploy an **outputs.conf** file from the DS with the following entry:

```
[tcpout]
defaultGroup = default-autolb-group

[tcpout-server://<indexer:listening_port>]

[tcpout:default-autolb-group]
disabled = false
server = <indexer:listening_port>
```

Note

In the lab exercises, you will configure forwarding using the deployment server (DS).

HF Configuration – Client

- To configure the HF as a deployment client to the Deployment Server by hostname or IP:

  splunk set deploy-poll <*deploymentServer:port*>

- This creates a **deploymentclient.conf** file with the following stanza and setting:

```
[target-broker:<deploymentServer>]
targetUri = 10.0.0.2##:8089
```


HF Configuration – Optimizations

- Based on your use case
- Disable indexing data on the HF:

outputs.conf

```
[indexAndForward]
index = false
```

- Disable Splunk Web on the HF:

web.conf

```
[settings]
startwebserver = 0
```


Selectively Routing Data

- Selectively send data from the UF directly to the indexers
- Route UF data through a HF to parse before sending to indexers

inputs.conf on UF2

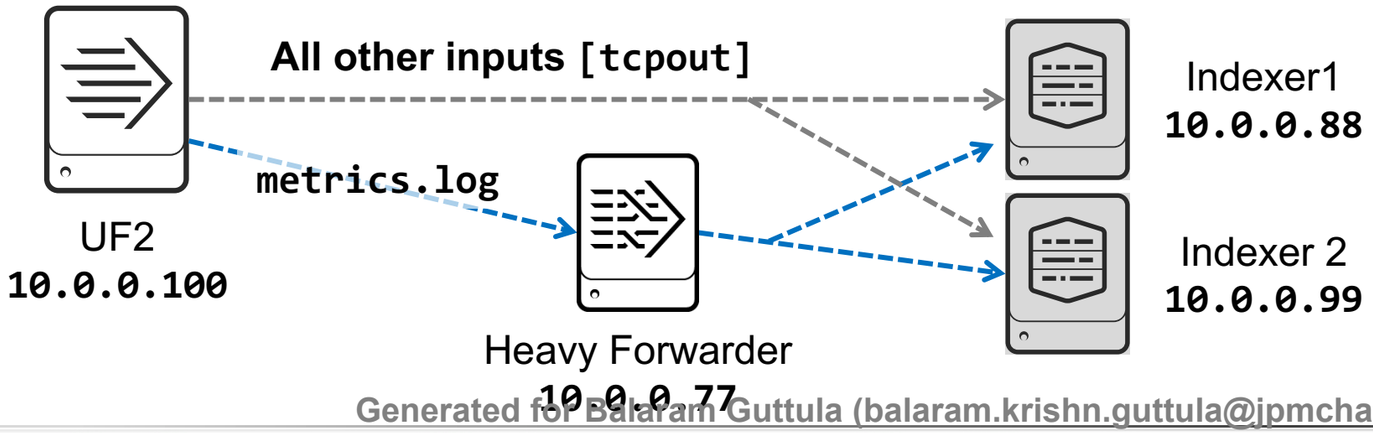
```
[monitor://path/metrics.log]
_TCP_ROUTING = HF_TheParser

[monitor://path/runtime.log]
```

outputs.conf on UF2

```
[tcpout]
defaultGroup = default-autolb-group

[tcpout:default-autolb-group]
disabled=false
server = 10.0.0.88:9997,10.0.0.99:9997

[tcpout:HF_TheParser]
server=10.0.0.77:99XX
```

Note

It is best practice to send your data to the indexers. Because of the limitations of the lab environment for this class, you will route all data from UF2 through the HF to the indexers.



Module 4 HF Knowledge Check

- True or False. The HF has a GUI.
- True or False. The UF and the HF can be used to mask data before transmitting to indexers.
- True or False. The listening port has to be 8089.

Module 4 HF Knowledge Check – Answers

- True or False. The HF has a GUI.

  True.

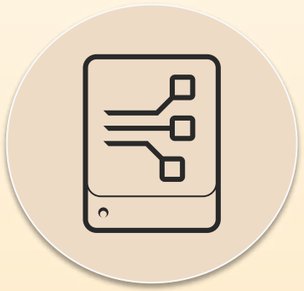
- True or False. The UF and the HF can be used to mask data before transmitting to indexers.

  False. Only the HF, specifically a Splunk Enterprise instance, can perform data masking.

- True or False. The default listening port is 8089.

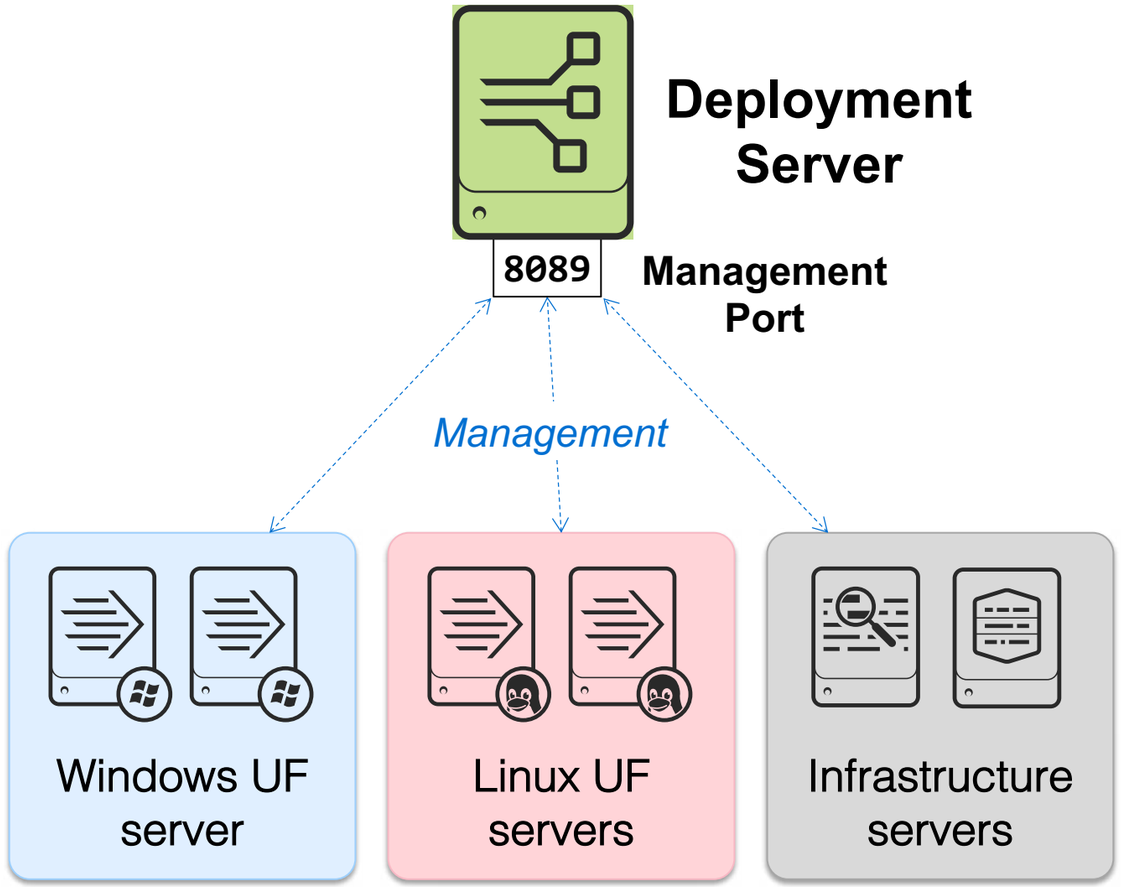
  False. 8089 is the default management port. The listening port can be any port.

Understanding the Deployment Server



Deployment Server (DS)

- Built-in tool for centrally managing configuration packages as apps for clients
- Includes **Forwarder Management** as the graphical user interface
- Can restart remote Splunk instances
- Requires an Enterprise license and should be on a dedicated server



Deployment Server Components

Deployment Apps

- Configuration files (such as **`inputs.conf`**) to be packaged into apps to be deployed to the deployment clients
- Reside in **`SPLUNK_HOME/etc/deployment-apps/`**

Server Class

- Groupings of deployment clients
- Define what apps should be deployed to which clients
- Saved in **`serverclass.conf`**

Deployment Clients

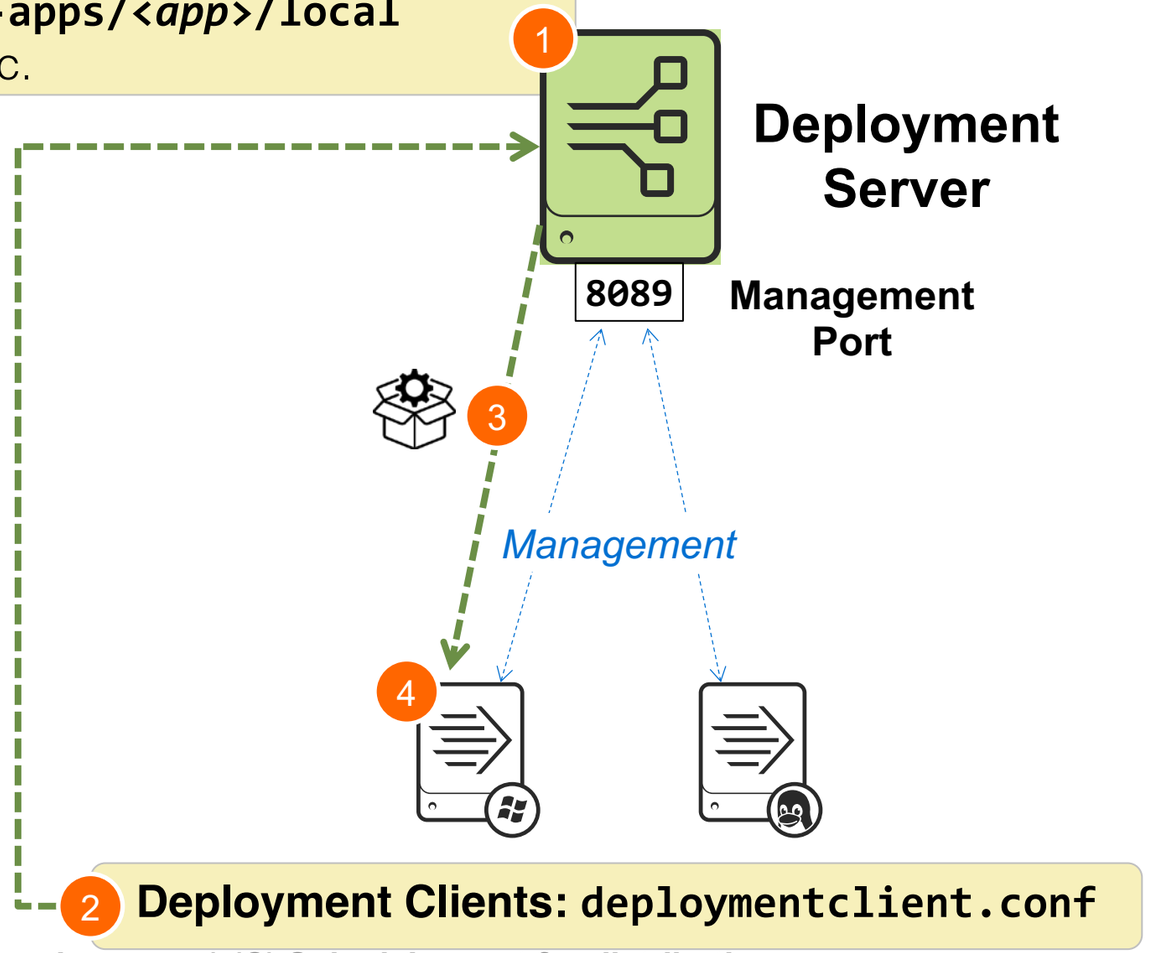
- Splunk instances (Enterprise or UF) that are connected to the Deployment Server (DS) and are phoning home
- Establish the connection from the Deployment Client

Deployment Server Configuration

Configuration on DS:

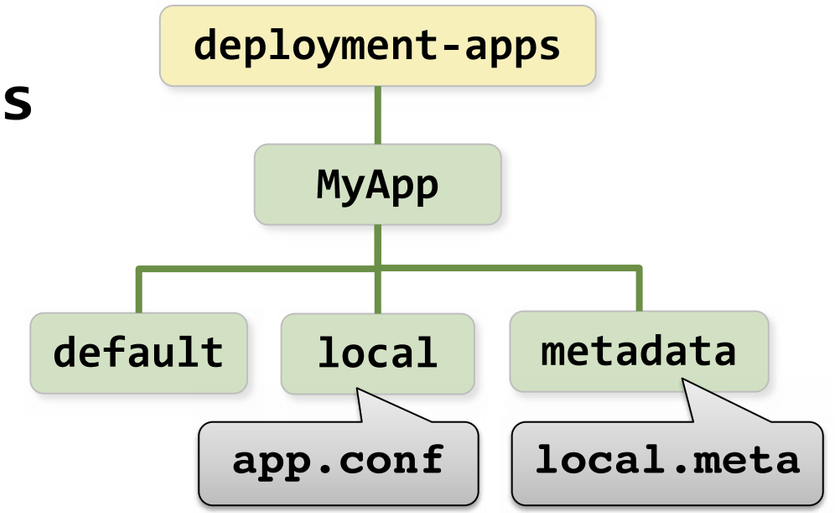
Map clients to apps:	**SPLUNK_HOME/etc/apps/<*app*>/local/serverclass.conf**
App repository:	**SPLUNK_HOME/etc/deployment-apps/<*app*>/local**
Apps/configs to deploy:	**outputs.conf**, **inputs.conf**, etc.

1. Configure DS, server class configuration, and app packages
2. Configure instances as deployment clients; phones home to deployment server
3. Client downloads subscribed apps as directed by DS
4. Client uses configuration; for example sending data to indexers configured in **outputs.conf**



Configuring a Deployment App

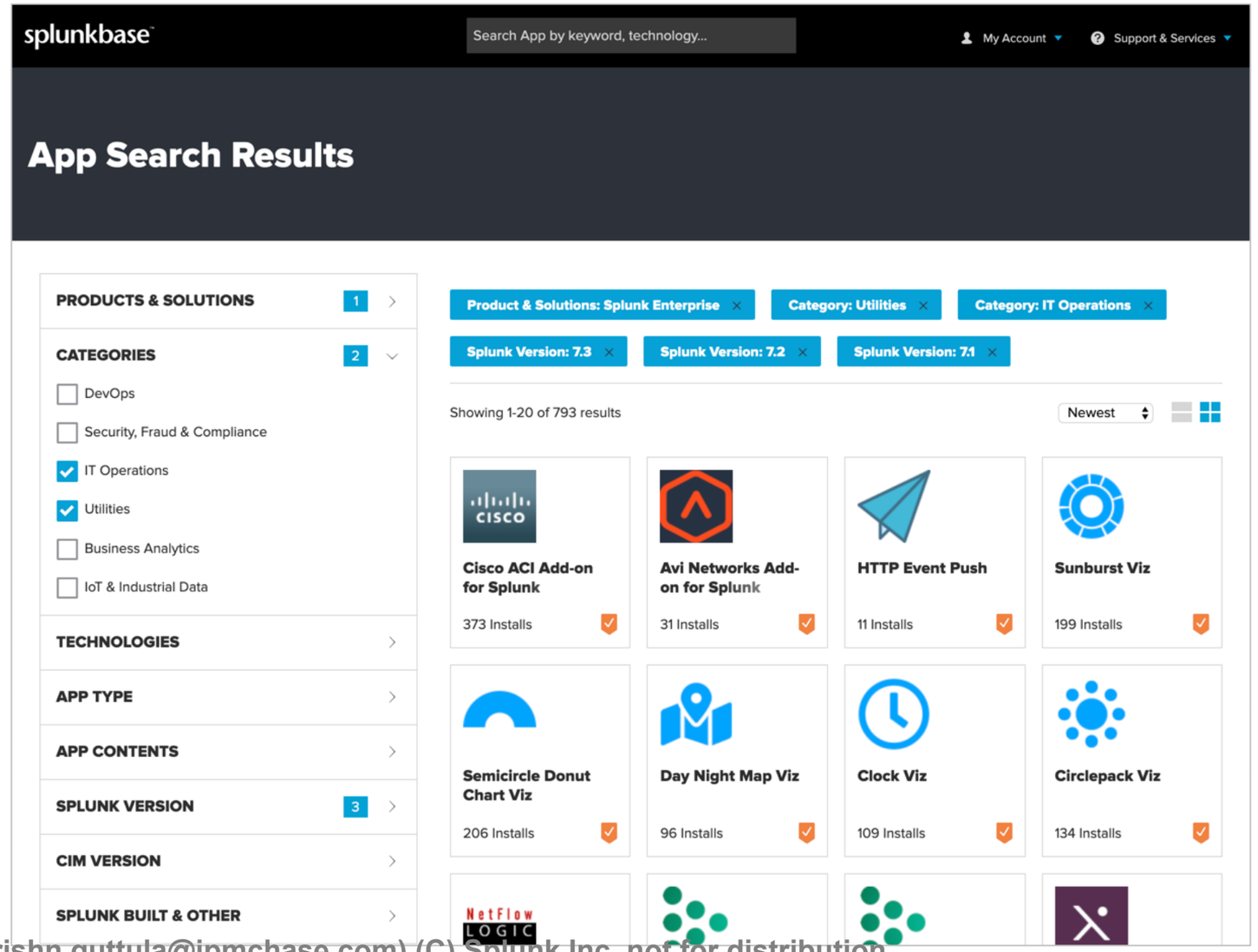
- Follows app structure and rules
  - Place files in **SPLUNK_HOME/etc/deployment-apps**
  - Required files:
    - **app.conf** (in **default** or **local**)
    - **local.meta** (in **metadata**)
  - Optionally may contain configuration files, scripts, and other resources



- Files are deployed to client's **SPLUNK_HOME/etc/apps** folder by default
- Best practice
  - Create small and discrete deployment apps
  - Take advantage of **.conf** file layering
  - Use a consistent naming convention

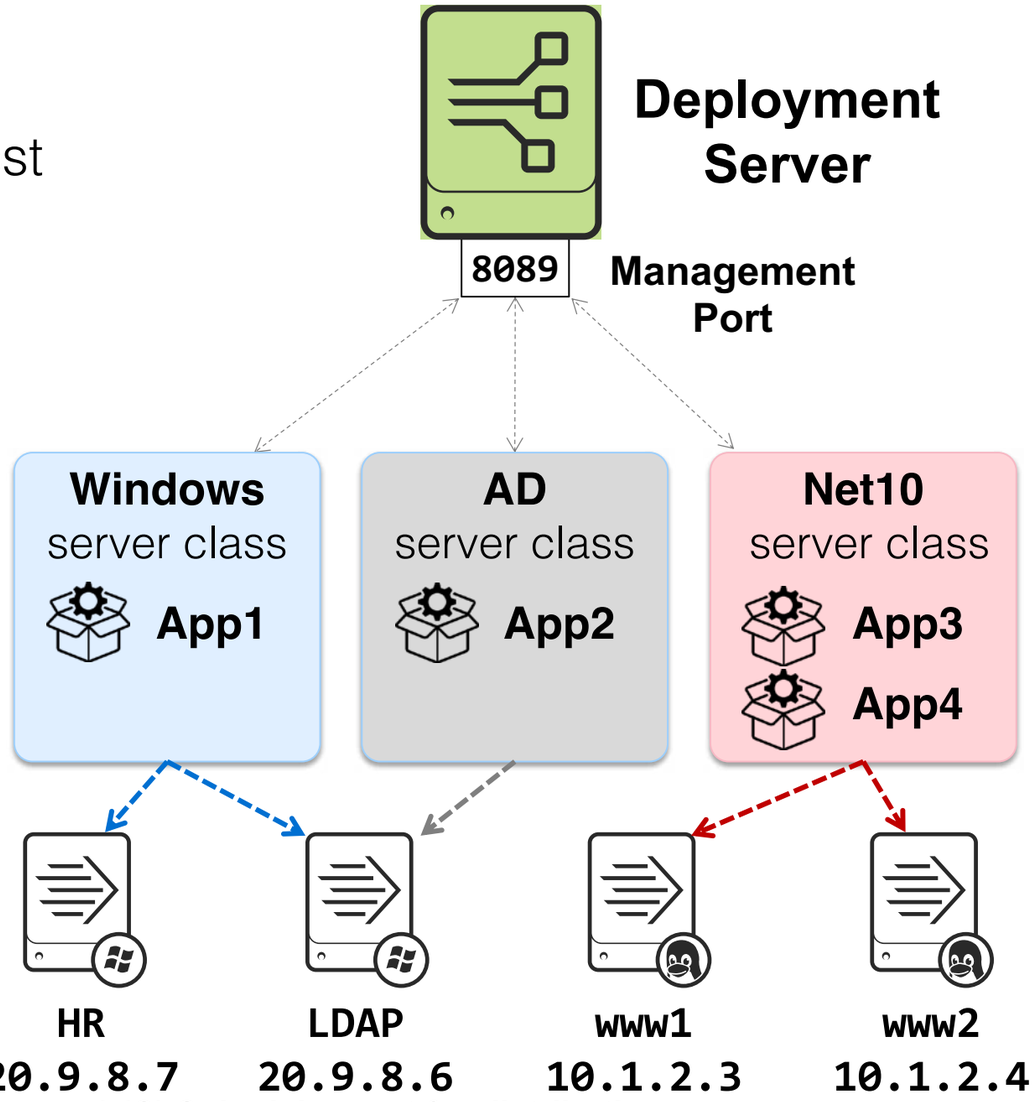
Apps and Add-ons

- Can be downloaded from Splunkbase
- Installed on a Splunk instance:
  - Using the Deployment Server
  - Using CLI on the instance
  - Manually by installing the app
- Deploy to **`SPLUNK_HOME/etc/apps`**
- Comes with documentation for details about settings for **`inputs.conf`**, and so on



What's a Server Class?

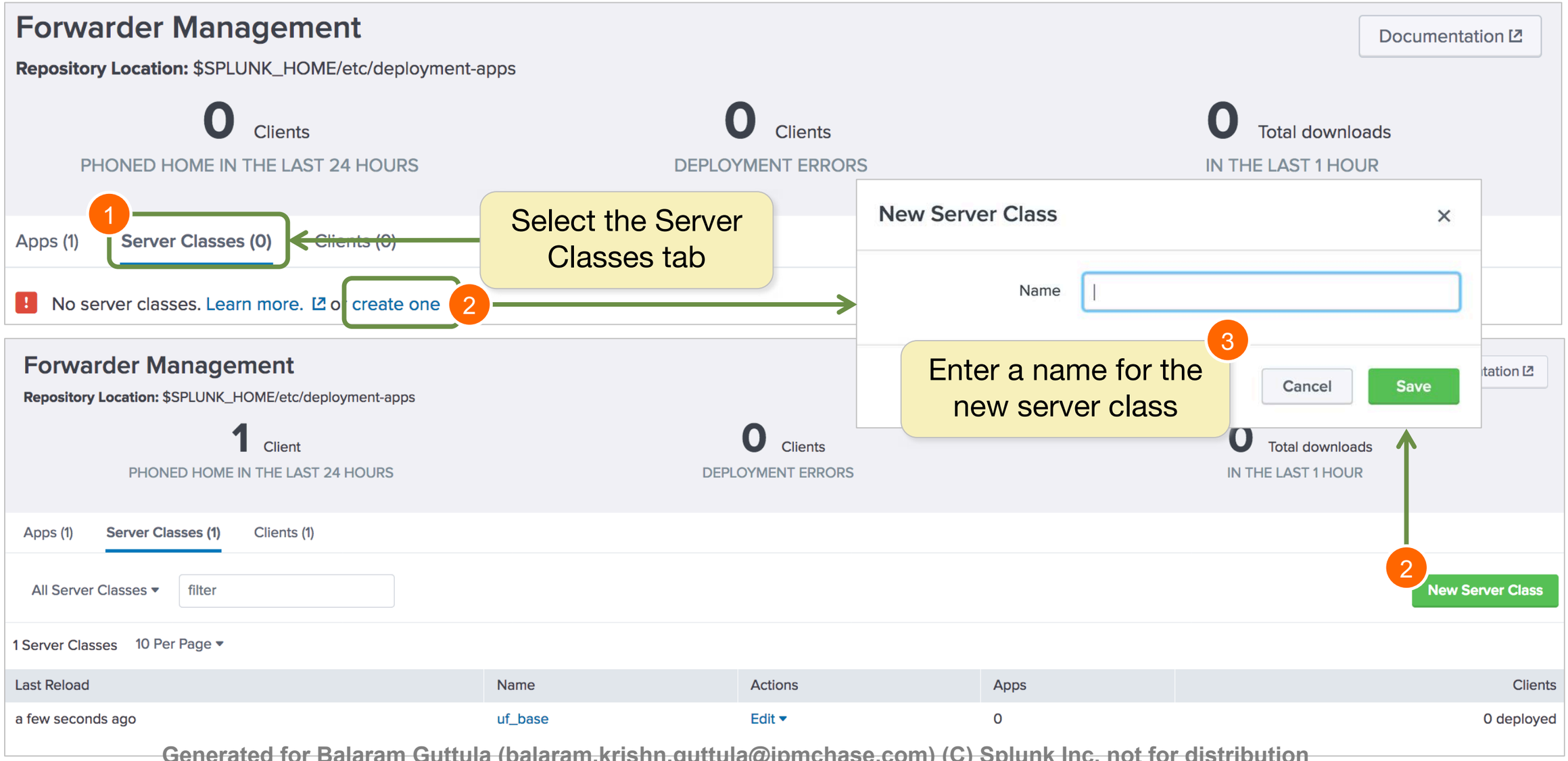
- Maps groups of clients to deployment apps
  - Clients can be grouped based on client name, host name, IP address, DNS name, or machine types
- Example:
  - **Windows** server class
    - Assigned to Windows systems
    - Installs **App1**
  - **AD** server class
    - Assigned to Active Directory servers
    - Installs **App2**
  - **Net10** server class
    - Assigned to hosts on **`10.1.2.*`** subnet
    - Installs **App3** and **App4**



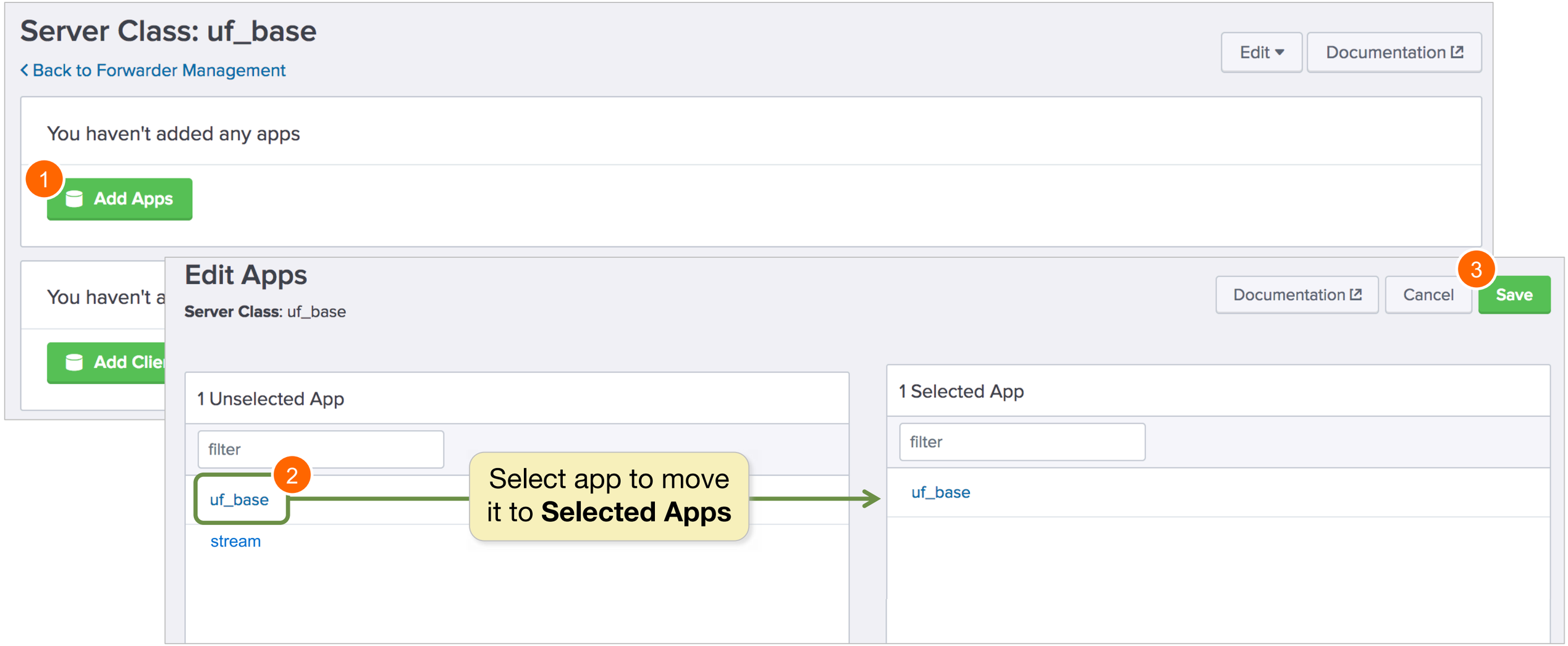
Enabling Forwarder Management

1. On deployment server: Add one or more apps in **SPLUNK_HOME/etc/deployment-apps**
2. In Forwarder Management UI: Create one or more server classes
3. On forwarders: Setup the deployment client
   - Run **splunk set deploy-poll <*DS:port*>** using the **splunkd** port on the deployment server (default **8089**)
   - Run **splunk restart**
4. On deployment server: Verify deployment clients and deployment status
5. On forwarders: Verify **SPLUNK_HOME/etc/apps** folder for deployed apps

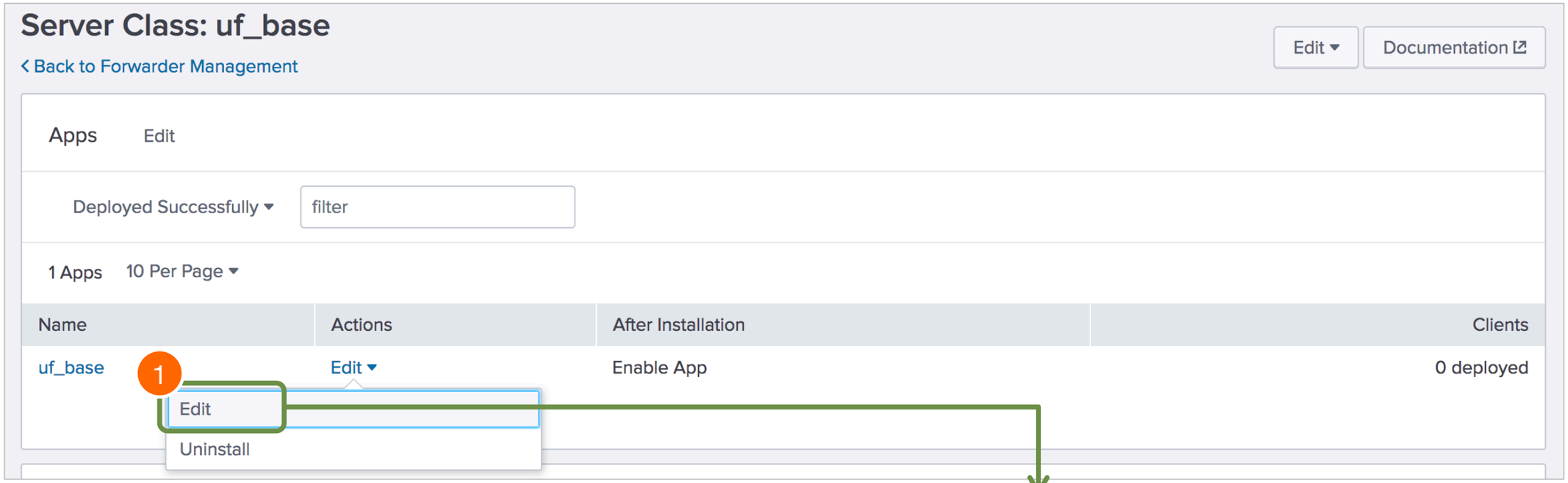
Adding a Server Class

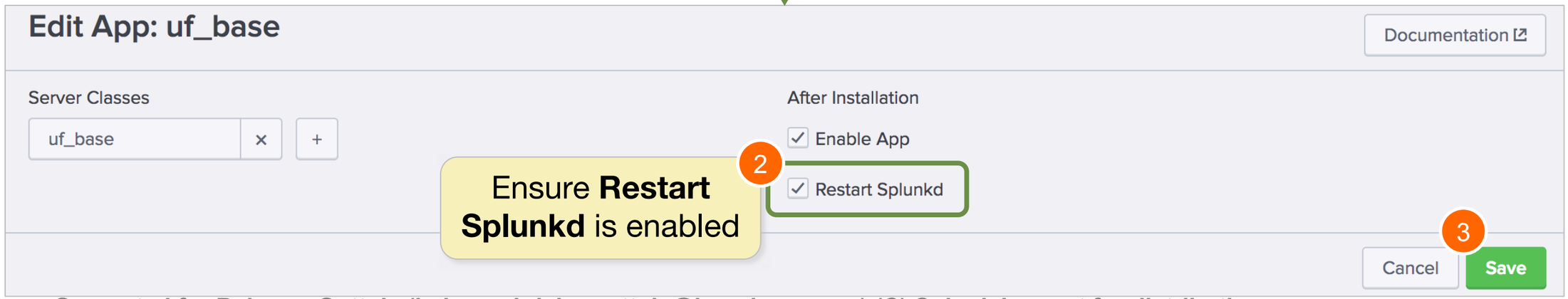


Selecting Apps for the Server Class

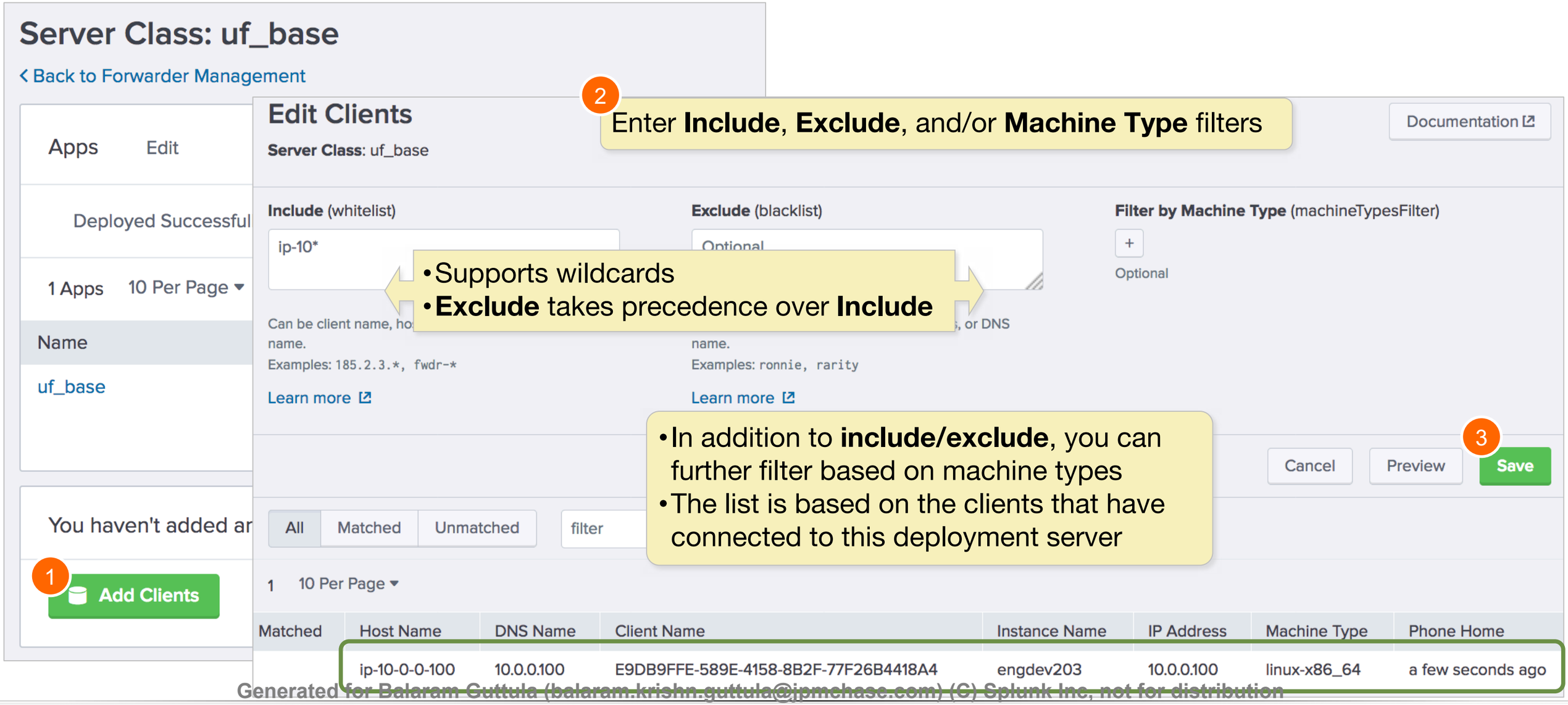


Post Deployment Behavior Setting





Selecting Clients for the Server Class



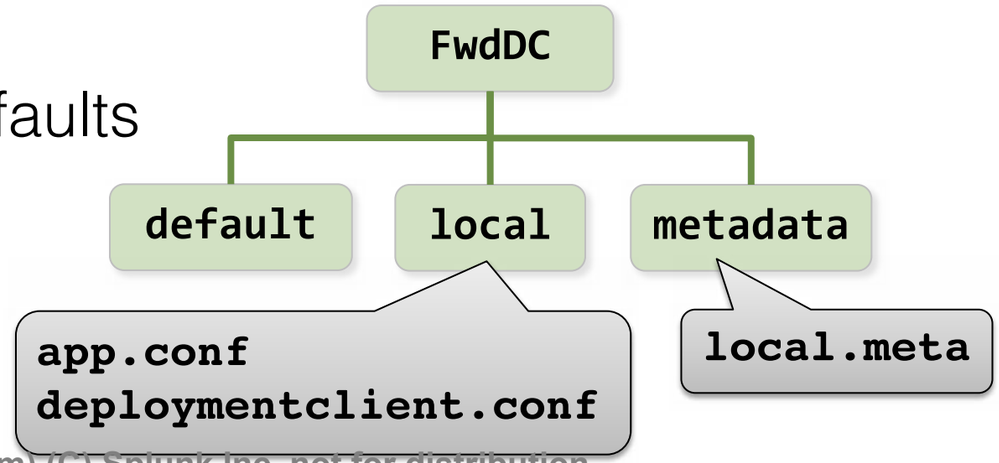
Configuring Deployment Clients

- Configure your forwarders to be deployment clients
  1. Run **`splunk set deploy-poll <DS:port>`**
     - ***DS*** = deployment server hostname or IP
     - ***port*** = splunkd port
     - Creates **`deploymentclient.conf`** in **`SPLUNK_HOME/etc/system/local`**

     (Alternately create **`deploymentclient.conf`** manually)
  2. Create an app and place the **`.conf`** file in the **`local`** directory
  3. Restart the deployment clients: **`splunk restart`**
- Edit **`[deployment-client]`** stanza to override defaults
  - Can be part of initial deployment app
  - Contains **`phoneHomeIntervalInSecs`** setting (default is 60 seconds)

`deploymentclient.conf`

```
[target-broker:deploymentServer]
targetUri = splunk_server:8089
...

[deployment-client]
clientName = webserver_1
phoneHomeIntervalInSecs = 600
```

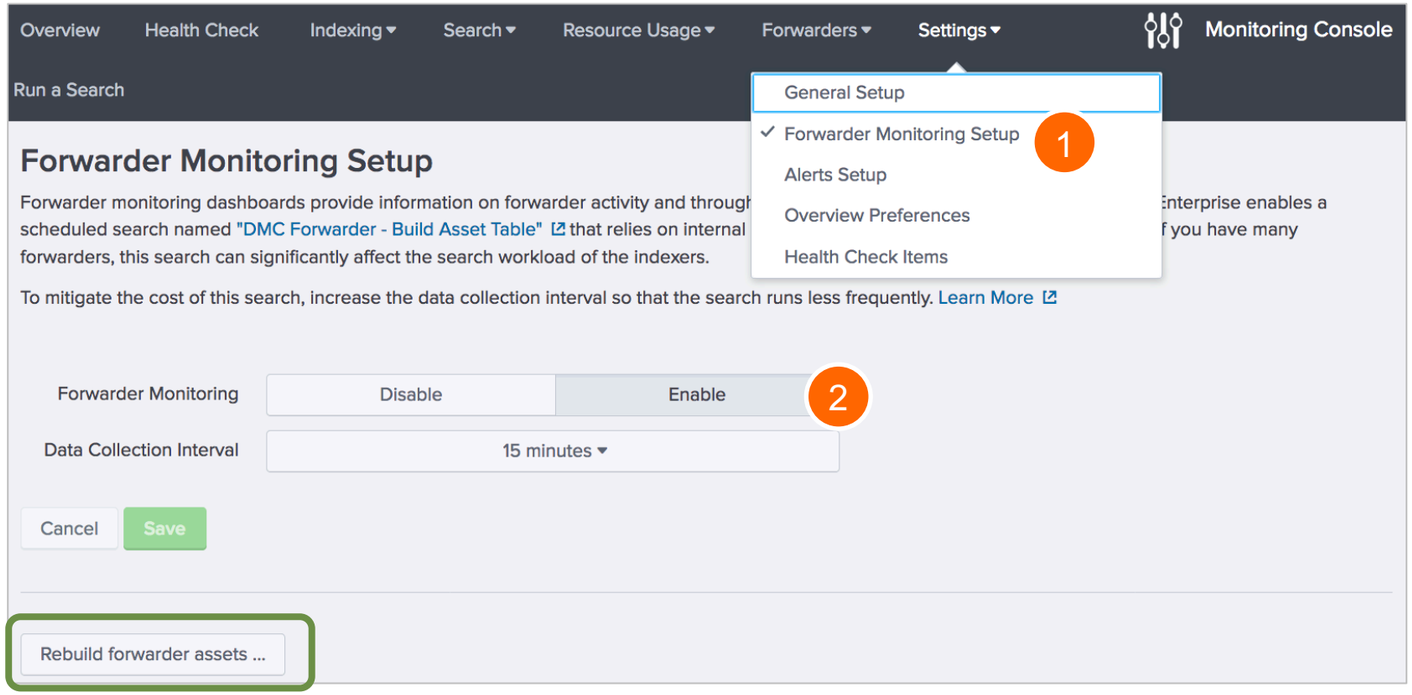


Configuring Deployment Clients (cont.)

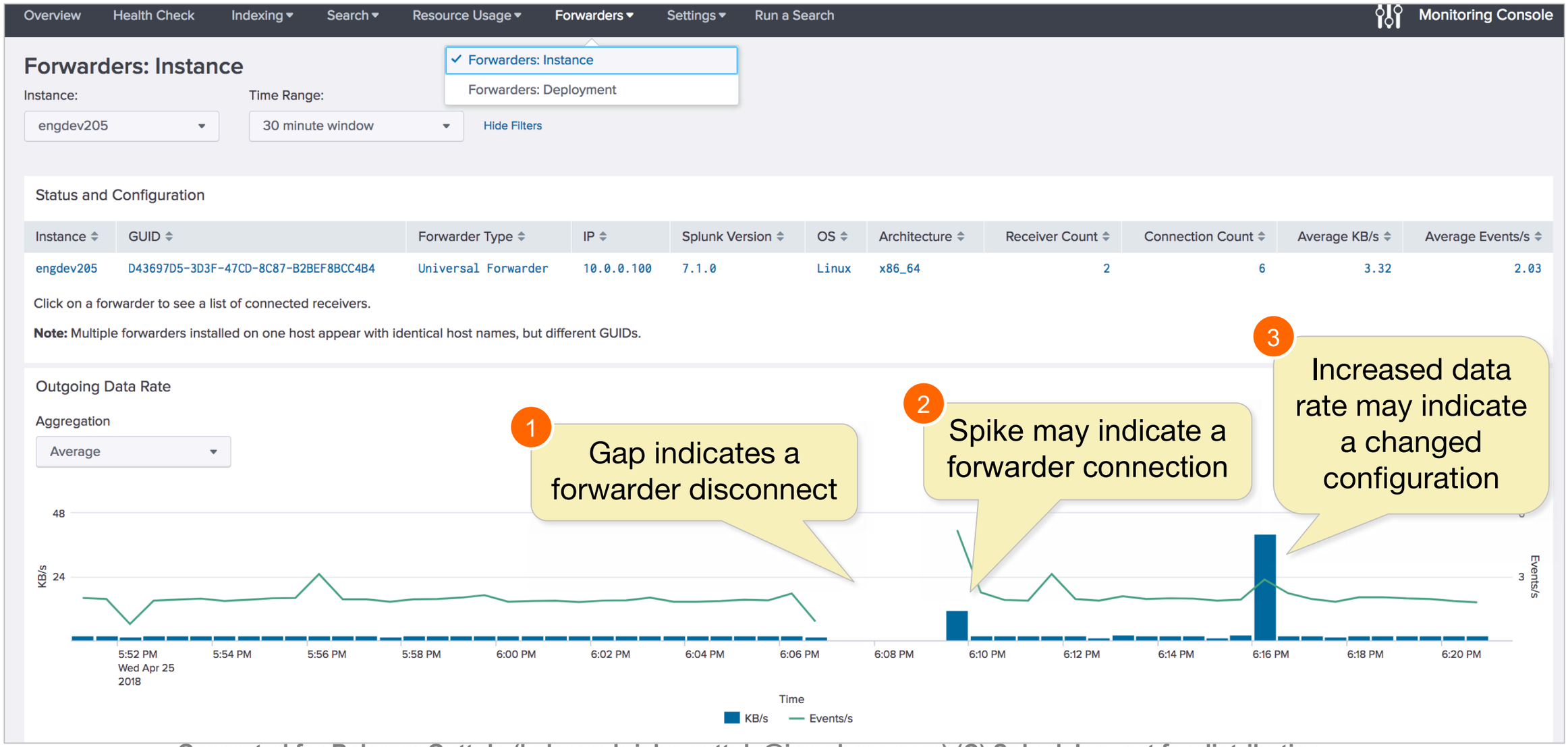
- On the deployment client (usually a forwarder)
  - Confirm that the expected app directories and contents have arrived in **`SPLUNK_HOME/etc/apps`**
- If the app is changed on the Deployment Server, then the forwarder will load the updated app after its next phone-home
  - To change the app settings using Forwarder Management, use the app's Edit menu associated with the server class
  - To change inputs for an app, go to **Settings > Data Inputs > Forwarded Inputs**
- If the post-deployment behavior option is set, the forwarder is restarted
- On the deployment client
  - Use **`splunk show deploy-poll`** to check the deployment server settings
  - Use **`splunk list forward-server`** to check the indexer destination settings

Forwarder Monitoring with Monitoring Console

- Provides valuable information on forwarder activity and throughput
- Enabled with:
  1. Monitoring Console: **Settings > Forwarder Monitoring Setup**
  2. **Forwarder Monitoring**: **Enable**
- Runs a scheduled search that builds a forwarder asset table
  - Runs every 15 minutes by default
  - Relies on forwarder internal logs
  - Can affect search workload if you have many forwarders
  - Can be rebuilt manually



Forwarder Monitoring with MC



Useful Commands

Command	Operation
From the Deployment Client	
`splunk set deploy-poll`	Connects the client to the deployment server and management port
`splunk show deploy-poll`	Displays the current deployment server and management port
`splunk disable deploy-client`	Disables the deployment client
From the Deployment Server (DS)	
`splunk reload deploy-server`	Checks all apps for changes and notifies the relevant clients the next time they phone home
`splunk list deploy-clients`	Displays information about the deployment clients
`splunk list forward-server`	Displays the current forward server configuration

Module 4 Knowledge Check

- On the DS, what is the difference between the apps sitting in the **`SPLUNK_HOME/etc/apps`** folder versus the **`SPLUNK_HOME/etc/deployment-apps`** ?
- When an app is deployed from the DS to the client, where will you find that app on the client by default?
- True or False. Deployment clients poll the DS on port 9997.

Module 4 Knowledge Check – Answers

- On the DS, what is the difference between the apps sitting in the **SPLUNK_HOME/etc/apps** folder versus the **SPLUNK_HOME/etc/deployment-apps**?

  The apps in the **.../etc/apps** folder are for the Deployment Server and the apps in the **…/etc/deployment-apps** are apps for deployment to a client.

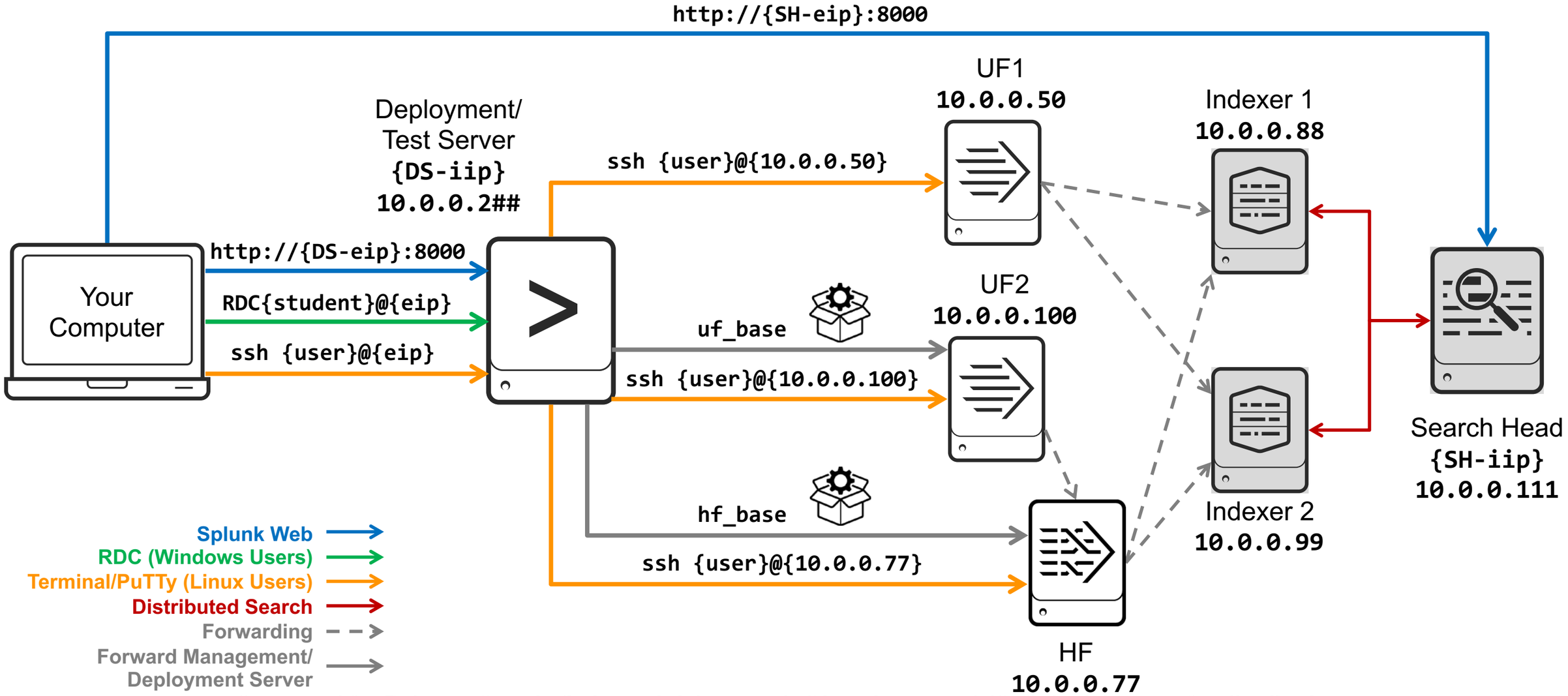
- When an app is deployed from the Deployment Server to the client, where will you find that app on the client by default?

  Apps by default are deployed from the DS to the client in the **SPLUNK_HOME/etc/apps** folder.

- True or False. Clients poll the DS on port 9997.

  False. Clients poll the DS on it's management port (**8089** by default.)

Module 4 Lab Exercise – Environment Diagram



Module 4 Lab Exercise – Forwarder Management

Time: 25 – 30 minutes

Tasks:

- Enable Forwarder Management by copying the **`uf_base`** and **`hf_base`** apps into the **`SPLUNK_HOME/etc/deployment-apps`** folder on the deployment/test server
- Configure UF2 as a deployment client to the deployment/test server
- Enable listening port on HF for data being transmitted from UF2
- Configure the HF as a deployment client to the deployment/test server
- Create two server classes to manage UF2 and the HF from the deployment/test server
- Confirm deployment of **`hf_base`** app on HF and **`uf_base`** app on UF2

Module 5: Monitor Inputs

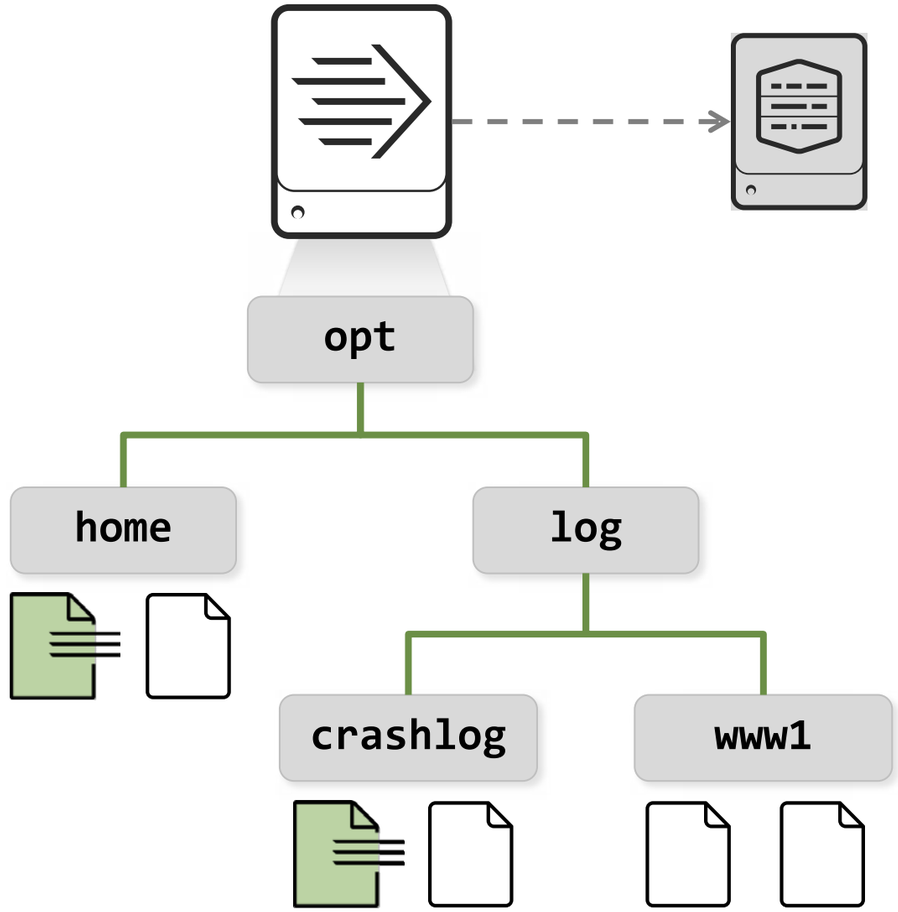
Module Objectives

- Create file and directory monitor inputs
- Use optional settings for monitor inputs
- Deploy a remote monitor input

Monitoring Input Files

Monitoring Files

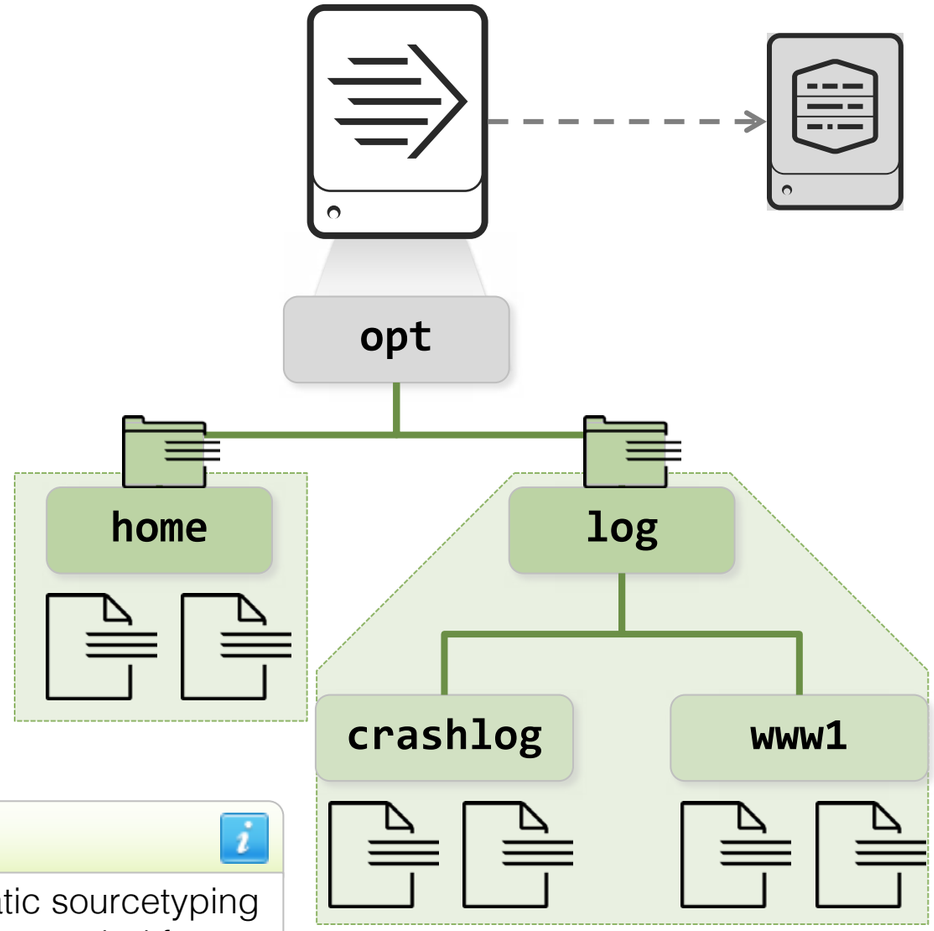
- Defines a single file as the source, with input settings (**sourcetype**, **index**, **host**, etc.)
- Ingests current contents of the file
- Continuously monitors for new content using the Splunk Fishbucket to keep a checkpoint
- Supports any text format, such as: plain text, structured text (**`CSV`**, **`XML`**, **`JSON`**), multi-line logs (**`Log4J`**), and files compressed with **`gzip`**



Monitoring Input Directories

Monitoring Directories

- Defines a directory tree as data source
- Recursively traverses directory for all discovered text files
- Unzips compressed files automatically before ingesting them, one at a time
- Includes new files added to the directories
- Detects and handles log file rotation
- Input settings applied to all contained files



Note

Automatic sourcetyping is recommended for directories with mixed file types.

Monitor Input Options in **inputs.conf**

- Defining the source
  - Place after **monitor://** in stanza header
  - Absolute path to a file or directory
  - Can contain wildcards
- Defining attributes
  - The sourcetype, host, index, etc. are optional
  - Default **host** is defined in **SPLUNK_HOME etc/system/local/inputs.conf**
  - Default **source** is the fully-qualified file name
  - Default **sourcetype** is **automatic**
- For more attributes and documentation
  - See **inputs.conf.spec** in **SPLUNK_HOME/etc/system/README**

inputs.conf format:

```
[monitor://<path>]
disabled=[0|1|false|true]
sourcetype=<string>
host=<string>
index=<string>
blacklist=<regular expression>
whitelist=<regular expression>
```

Example **monitor** path entries:

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

[monitor://C:\logs\system.log]

[monitor://C:\logs\]

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


File Pathname Wildcards in **`inputs.conf`**

Wildcard	Description
...	The ellipsis wildcard recurses through directories and subdirectories to match.
*	The asterisk wildcard matches anything in that specific directory path segment but does not go beyond that segment in the path. Normally it should be used at the end of a path.

File and Directory Matching

```
[monitor:///var/log/www1/secure.log]
sourcetype = linux_secure
```

✓ /var/log/www1/secure.log
✗ /var/log/www1/secure.1
✗ /var/log/www1/logs/secure.log
✗ /var/log/www2/secure.log

```
[monitor:///var/log/www1/secure.*]
sourcetype = linux_secure
```

✓ /var/log/www1/secure.log
✓ /var/log/www1/secure.1
✗ /var/log/www1/logs/secure.log
✗ /var/log/www2/secure.log

```
[monitor:///var/log/www*/secure.*]
sourcetype = linux_secure
```

✓ /var/log/www1/secure.log
✓ /var/log/www1/secure.1
✗ /var/log/www1/logs/secure.log
✓ /var/log/www2/secure.log

```
[monitor:///var/log/.../secure.*]
sourcetype = linux_secure
```

✓ /var/log/www1/secure.log
✓ /var/log/www1/secure.1
✓ /var/log/www1/logs/secure.log
✓ /var/log/www2/secure.log

✓ Matches
✗ Doesn't match

Additional Options

Follow tail (`followTail`)

- Splunk ignores file's existing content, indexing new data as it arrives
- DO NOT leave enabled indefinitely

Ignore older than (`ignoreOlderThan`)

- Only index events after the time window (such as only events since 60 days ago with **`ignoreOlderThan = 60d`**)
- Completely ignores files with modification time outside the time window (even if the file is updated later)

Whitelist and Blacklist

- Use regular expressions to filter files or directories from the input
- In case of a conflict, the blacklist prevails

Example: Using Whitelist to Include Files

- Files/directories that match the regular expression are indexed
- The syntax for blacklists is identical

```
[monitor:///var/log/www1/]
whitelist = \.log$
```

✓ /var/log/www1/access.log
✓ /var/log/www1/dbaccess.log
✓ /var/log/www1/access.1.log
✗ /var/log/www1/access.log.2

```
[monitor:///var/log/www1/]
whitelist = query\.log$|my\.log$
```

✓ /var/log/www1/query.log
✓ /var/log/www1/dbquery.log
✓ /var/log/www1/my.log
✗ /var/log/www1/my.log4j

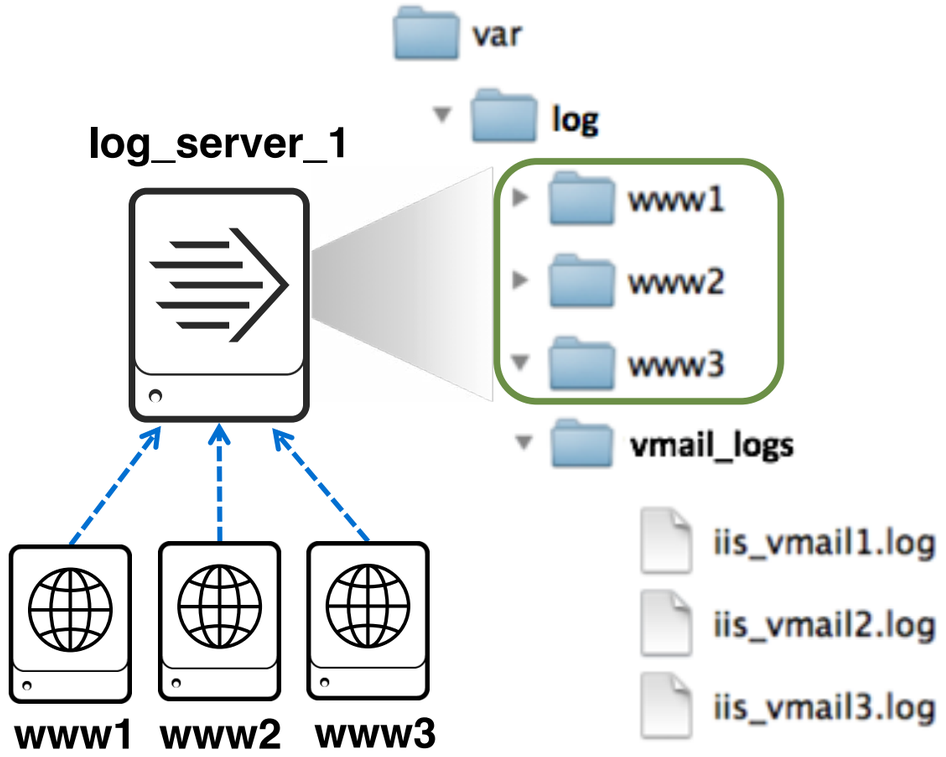
```
[monitor:///var/log/www1/]
whitelist = /query\.log$|/my\.log$
```

✓ /var/log/www1/query.log
✓ /var/log/www1/my.log
✗ /var/log/www1/dbquery.log
✗ /var/log/www1/my.log4j

✓ Matches
✗ Doesn't match

Overriding the Host Field

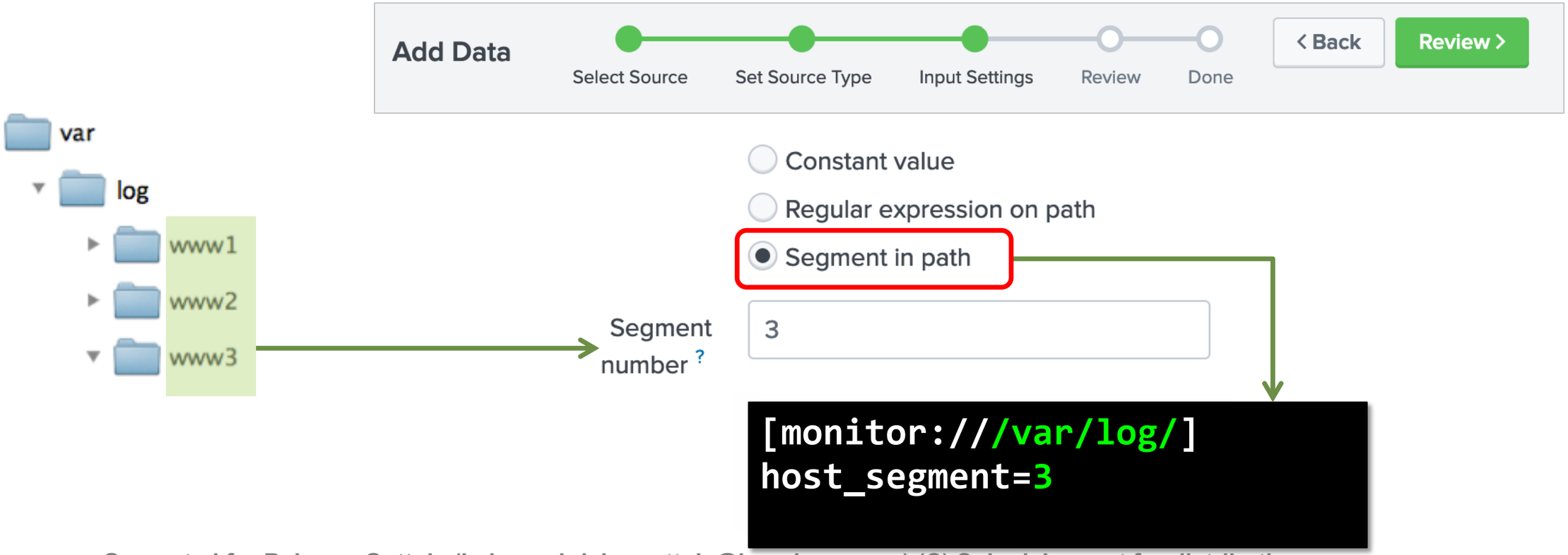
- When data is stored on a different server than its origin
  - Example: A web farm where each web server stores its log file on a centralized file server
- By explicitly setting the **host**
  - Using a specified value
  - Using a directory name
  - Using a regular expression



Setting the Host With a Directory Name

- Used with **`host_segment = <integer>`**

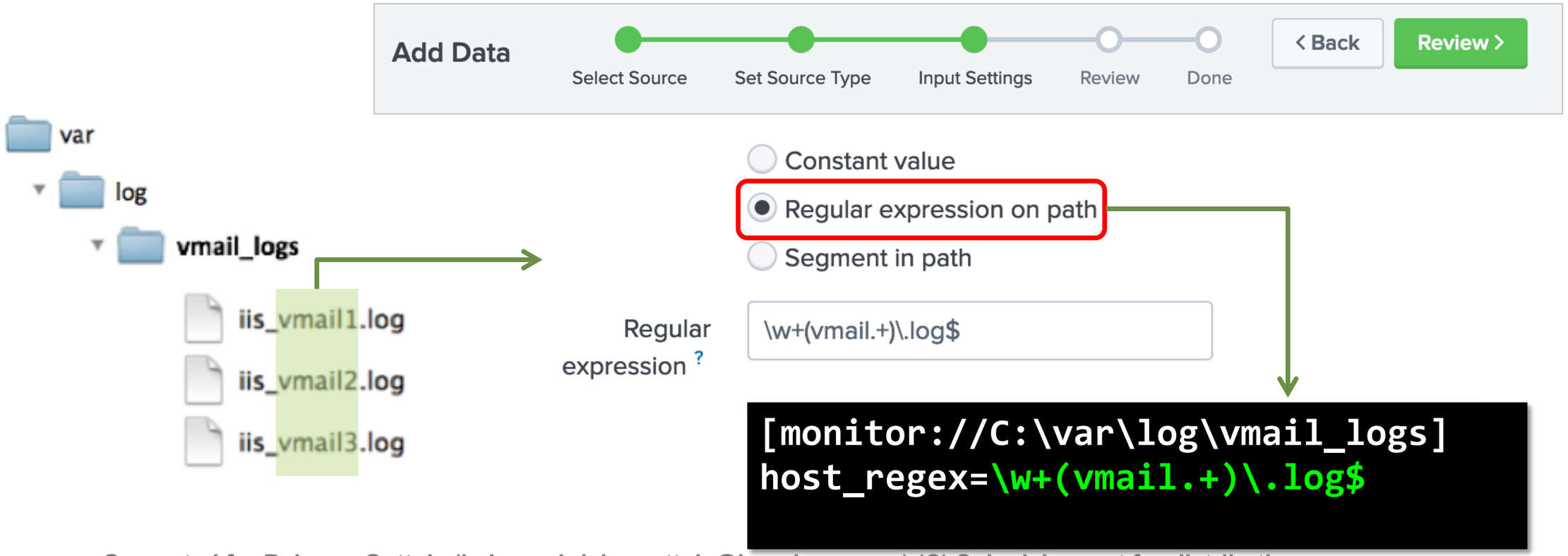
  Example: Setting **`host_segment`** to **`3`** uses the 3rd segment of the directory path as the host name for files in that directory



Setting the Host With a Regular Expression

- Used with **`host_regex = <regular expression>`**

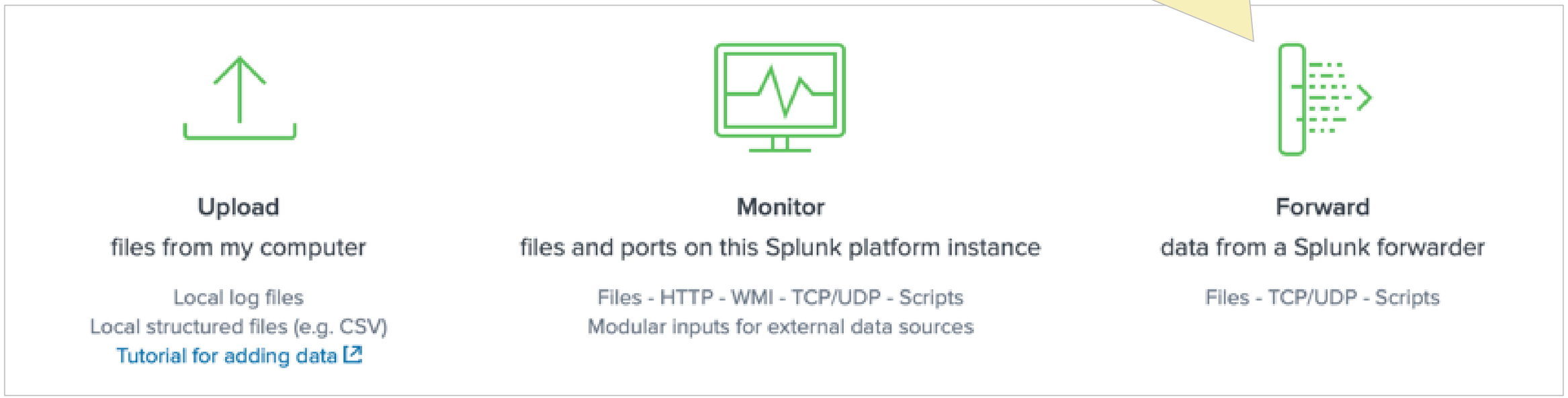
  Example: Setting **`host_regex`** to **`\w+(vmail.+)\.log$`** selects the second part of the log file name as its host name



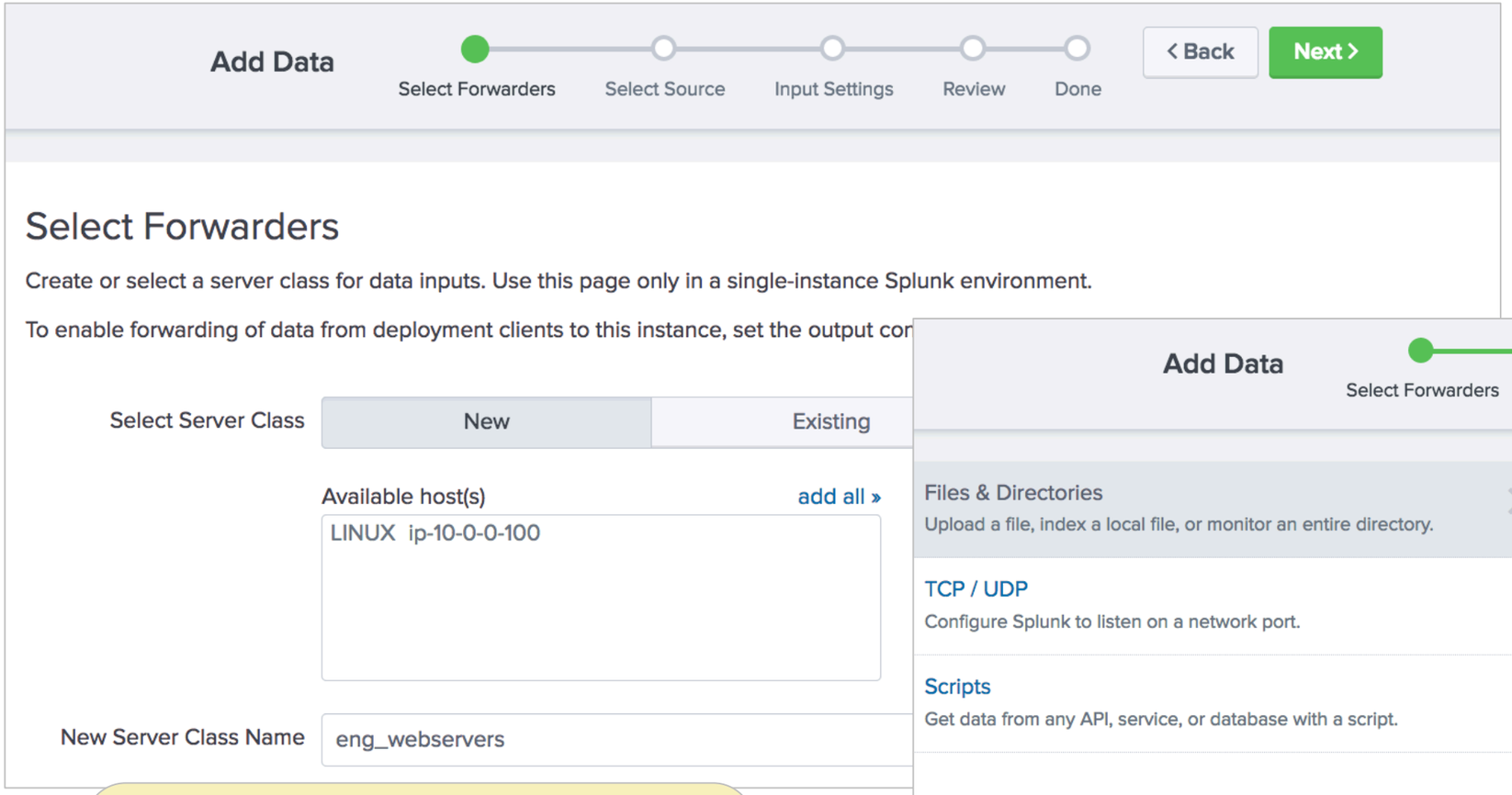
Creating a Remote Data Input

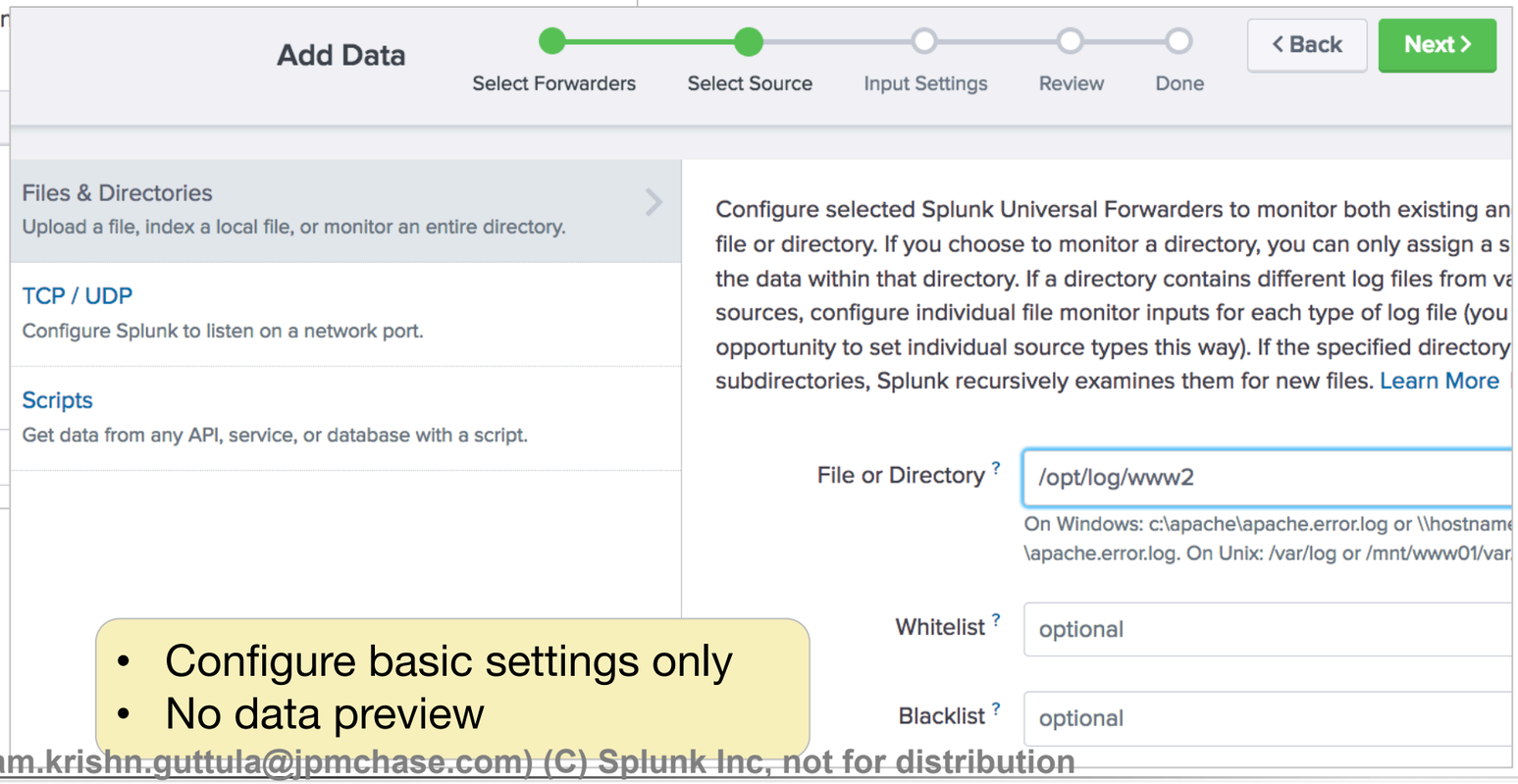
After deployment clients are working, optionally create deployment apps for configuring inputs on the clients

Uses deployment server to distribute the **inputs.conf**



Creating a Remote Data Input (cont.)

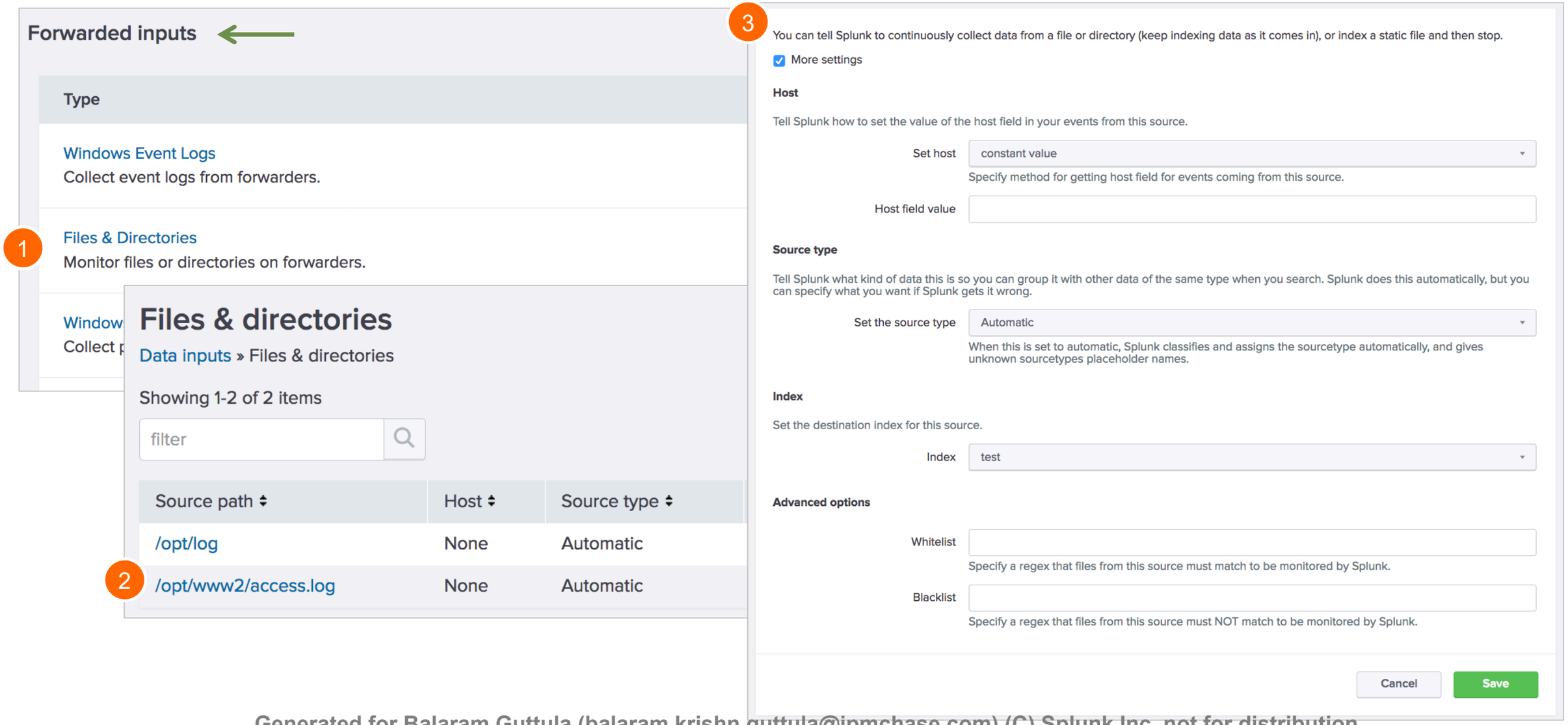




- Creates new server class or uses existing one
- Creates a new app for this input (or updates existing)

- Configure basic settings only
- No data preview

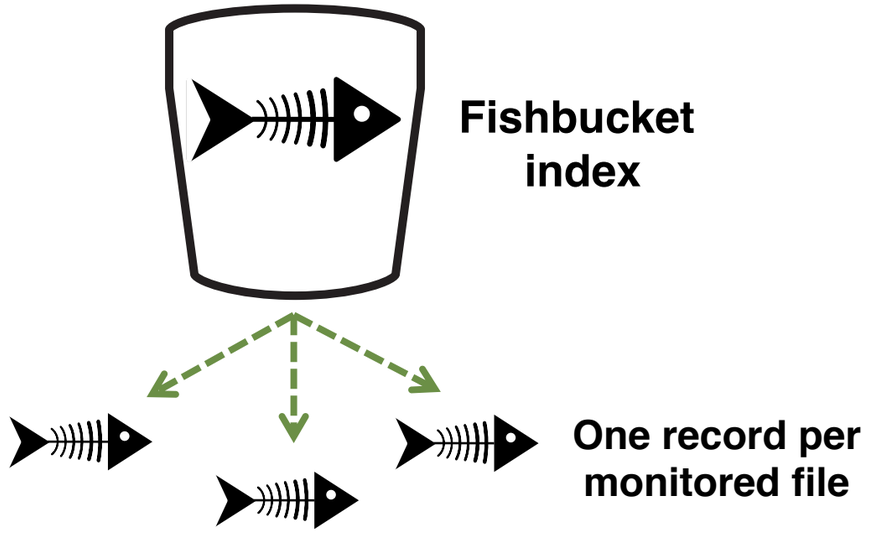
Editing Remote Data Input



What is the Fishbucket?

Fishbucket

- Allows Splunk to track monitored input files
- Contains file metadata which identifies a pointer to the file, and a pointer to where Splunk last read the file
- Exists on all Splunk instances
- Stored in a special subdirectory found at **`SPLUNK_DB/fishbucket`**



Includes:

- **Head**: Pointer to the file
- **Tail**: Pointer showing where Splunk last left off indexing in the file

Editing Inputs and Re-indexing Data

- Editing the **`inputs.conf`**
  - Only applies changes to new data
  - Does not change or cause re-indexing of existing ingested data
- To re-index:
  1. Delete the old, erroneous data on the indexers
     - May require assistance from the system administrator
  2. Change the **`inputs.conf`** on the deployment server (or forwarders)
  3. Reset the fishbucket checkpoint on the involved forwarders
  4. Restart Splunk forwarders

Resetting input file monitors

1. Stop Splunk
2. Reset applicable file monitors
   - Individually for each source:

```
splunk cmd btprobe -d SPLUNK_DB/fishbucket/splunk_private_db
  --file <source> --reset
```

   - All sources (use only on test systems / with extreme caution):

```
splunk clean eventdata -index _thefishbucket
```

or

```
rm -r SPLUNK_DB/fishbucket
```

3. Start Splunk

Warning

Resetting the fishbucket forces re-indexing of all file monitors affected. This results in more license usage.

Module 5 Knowledge Check

- True or False. You can use the wildcards **...** and ***** in the whitelist and blacklist.
- True or False. The **host_regex** setting in **inputs.conf** can extract the host from the filename only.
- After a file monitor is set up and is running, if you change the host value, will the new host value be reflected for already ingested data?
- In our environment, we have a UF, an Indexer and a SH. Which instance contains the **_thefishbucket**?

Module 5 Knowledge Check – Answers

- True or False. You can use the wildcards, **...** and ***** in the whitelist and blacklist.

  False. The wildcards, ... and * are meant for the stanzas.

- True or False. The **host_regex** setting in **inputs.conf** can extract the host from the filename only.

  False. It can extract the host from the path of the file.

- After a file monitor is set up and is running, if you change the host value, will the new host value be reflected for already ingested data?

  No. All changes apply to the new data only. To reflect changes for your old data, you need to delete and re-ingest the old data.

- In our environment, we have a UF, an Indexer and a SH. Which instance contains the **_thefishbucket**?

  Each instance will have its own local **_fishbucket**.

Module 5 Lab Exercise – File Monitors

Time: 20 – 25 minutes

Tasks:

- To test-collect remote data from UF#2, add a remote directory monitor input to the **`test`** index
- Modify the **`inputs.conf`** file using the following caveats and re-deploy
  - Send the source logs to the **`sales`** index
  - Override the **`default-host`** name value
  - To monitor only the **`www.*`** sub-directories, use **`whitelist`**
    - Exclude the indexing of the **`secure.log`** files, use **`blacklist`**

Module 6: Network and Scripted Inputs

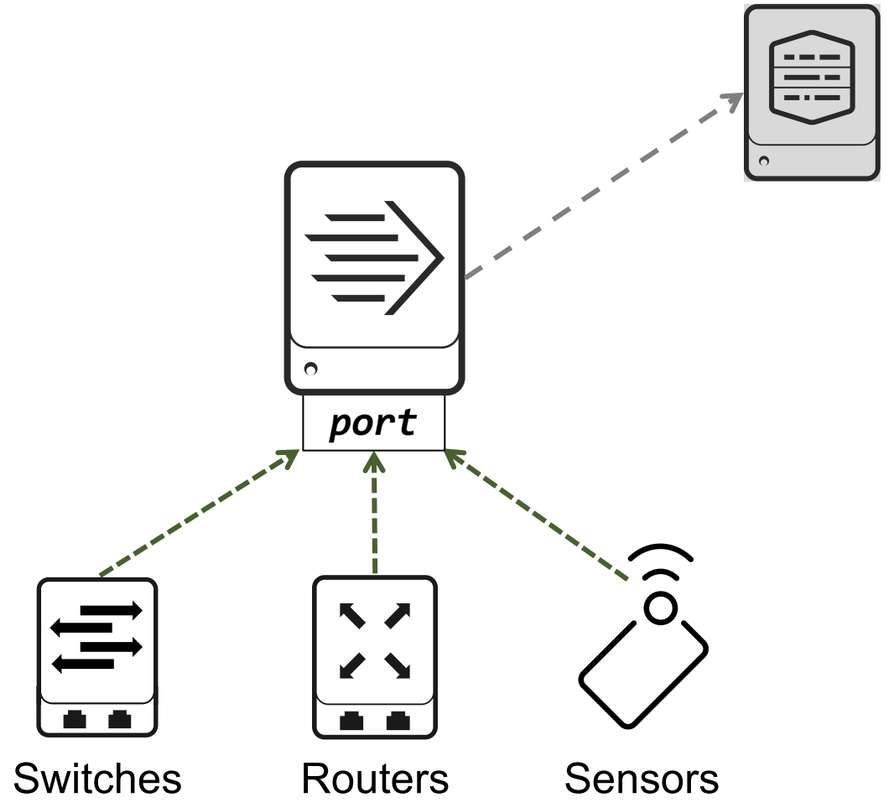
Module Objectives

- Create network (TCP and UDP) inputs
- Describe optional settings for network inputs
- Create a basic scripted input

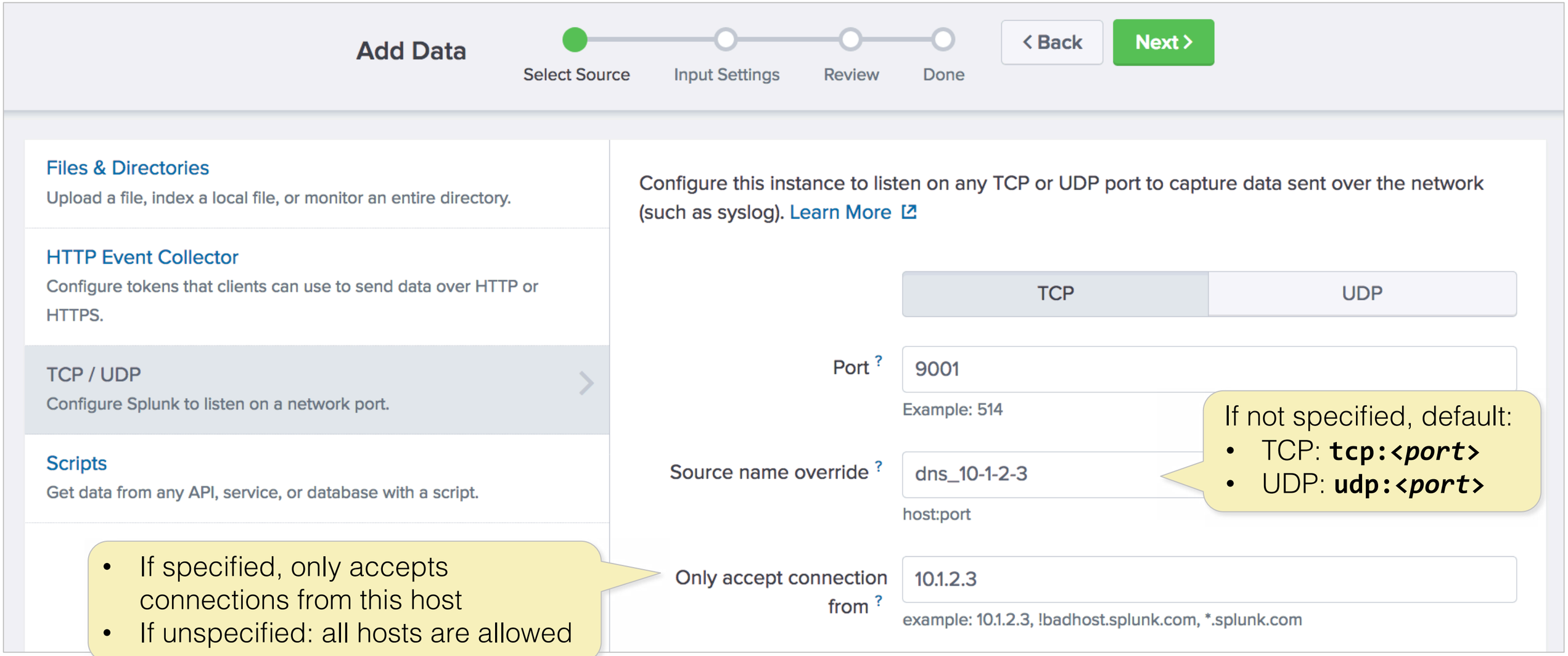
Network Inputs

Network Inputs

- Input data sent to a Splunk instance on a TCP/UDP port (for example: Syslog)
- Adds a layer of resiliency (buffering, load balancing, cloning, indexer restarts)
- Can minimize indexer workload by managing network connections on the forwarder (which can additionally bridge network segments)



Adding Network Input



Optional Network Input Settings

- Edit the stanza directly to fine-tune input settings:
  - Metadata override
  - Sender filtering options
  - Network input queues
    - Memory queues
    - Persistent queues

```
[udp://<[host:]port>]
connection_host = dns
sourcetype=<string>

[tcp://<[host:]port>]
connection_host = dns
source=<string>
```

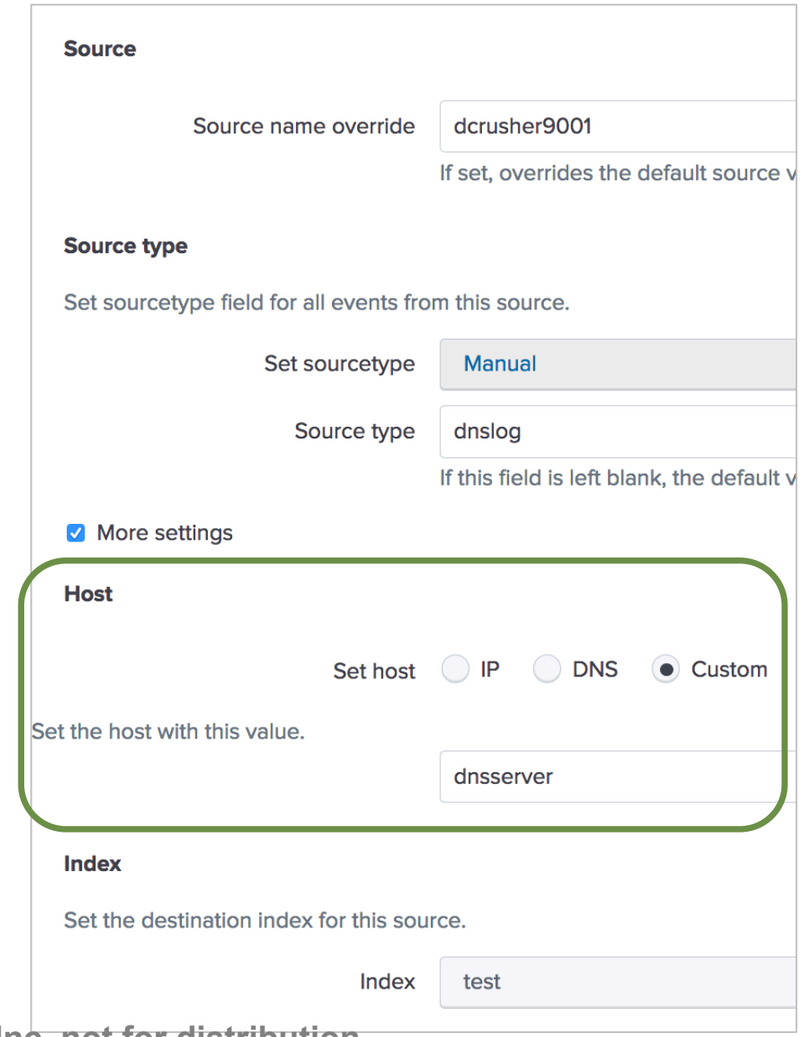
Examples:

```
[udp://514]
connection_host = dns
sourcetype=syslog

[tcp://10.1.2.3:9001]
connection_host = dns
source = dns_10-1-2-3
```


Network Input: Host Field

- Set in **inputs.conf** with the **connection_host** attribute:
  - **dns** (default for TCP inputs)
    - The host is set to a DNS name using reverse IP lookup
  - **ip** (default for UDP inputs)
    - The host is set to the originating host's IP address
  - **none** (Custom in the UI)
    - Requires explicitly setting the **host** value

```
[tcp://9002]
sourcetype=auth-data
connection_host=dns

[tcp://9003]
sourcetype=ops-data
connection_host=ip

[tcp://9001]
sourcetype=dnslog
connection_host=none
host=dnsserver
```



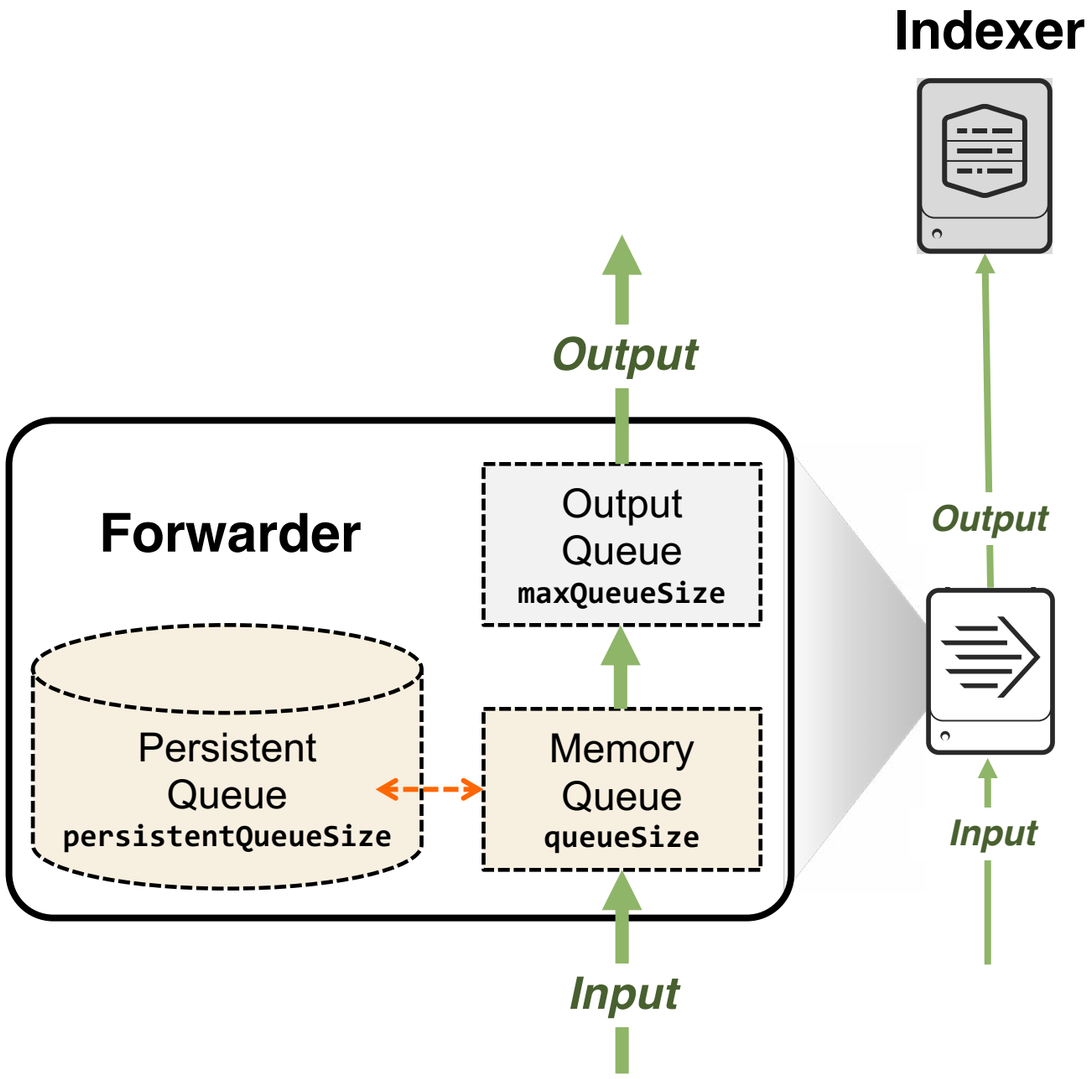
Network Input: Sender Filtering Options

- Specify which input streams are accepted by Splunk
- Example:
  - Network devices are sending syslog reports (UDP 514) to the Splunk network input, but want to accept UDP inputs selectively
- Use **`acceptFrom = <network_acl>`**
  - List address rules separated by commas or spaces
  - Available formats include:
    - Single IPv4 or IPv6 address
    - CIDR block of addresses
    - DNS name
    - Wildcards: **`*`** (any), **`!`** (not)

```
[udp://514]
sourcetype=syslog
connection_host=ip
acceptFrom=!10.1/16, 10/8
```

Network Input: Queues

- Queues provide input flow control
  - Applies to TCP, UDP, scripted input
  - Controls network data bursts, slow resources, or slow forwarding
  - If indexers can't be reached: Data is stored in the output queue
  - If the output queue is full: Data is stored in the memory queue
  - If the memory queue is full: Data is stored in the persistent queue on disk
- Persistent queue preserves across restarts
  - Not a solution for input failure



Network Input: Setting Queue Attributes

- Memory queue
  - Set with **queueSize** attribute (default = 500 KB)
  - Memory-resident queue that buffers data before forwarding
  - Useful if indexer receives data slower than forwarder is acquiring it
  - Independent of forwarder's **maxQueueSize** attribute
- Persistent queue
  - Set with **persistentQueueSize** attribute (doesn't exist by default)
  - Provides additional, file-system buffering of data
  - Written to **SPLUNK_HOME/var/run/splunk/...**
  - Useful for high-volume data and in the case of network outage to indexers

inputs.conf

```
[tcp://9001]
queueSize=10MB
persistentQueueSize=5GB
```

Special Handling and Best Practices

UDP

- Splunk merges UDP data until it finds a timestamp by default
- Default behavior can be overridden during the parsing phase

Syslog

- Send data to a syslog collector that writes into a directory structure (for example: **`/var/log/syslog1/syslog.txt`**)
- Monitor the directory and use **`host_segment`**
- docs.splunk.com/Documentation/Splunk/latest/Data/HowSplunkEnterprisehandlessyslogdata

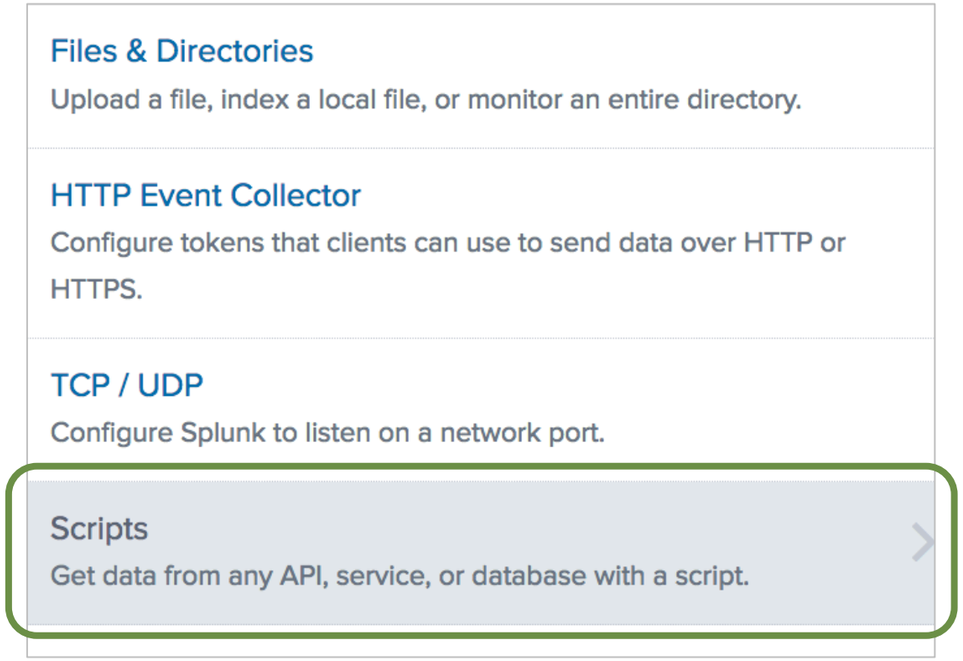
SNMP traps

- Write the traps to a file and use the monitor input
- docs.splunk.com/Documentation/Splunk/latest/Data/SendSNMPeventstoSplunk

Scripted Inputs

Scripted Inputs

- Schedules script execution and indexes the output
- Used to collect diagnostic data from OS commands (such as **`top`**, **`netstat`**, **`vmstat`**, **`ps`** etc.)
- Used by many Splunk apps to gather information from the OS or other server applications
- Can gather transient data that cannot be collected with Monitor or Network inputs (Examples: APIs, message queues, Web services, custom transactions)
- Supports Shell (**`.sh`**), Batch (**`.bat`**), PowerShell (**`.ps1`**) and Python (**`.py`**) scripts



Warning

Splunk only executes scripts from:

- **`SPLUNK_HOME/etc/apps/<app_name>/bin`**
- **`SPLUNK_HOME/bin/scripts`**
- **`SPLUNK_HOME/etc/system/bin`**

 | 29 May 2020

Defining a Scripted Input

1. Develop and test the script
2. Test your script from the context of a Splunk app
   - Copy the script to the app's **bin** directory on a test/dev server
   - Run script using the **splunk cmd *scriptname*** command

     Example: **splunk cmd SPLUNK_HOME/etc/apps/<*app*>/bin/myscript.sh**
3. Deploy the scripted input using the Deployment Server
   - Copy script to **SPLUNK_HOME/etc/deployment-apps/<*app*>/bin/**
   - Deploy script using **Add Data > Forward** from Splunk Web
4. Verify the output of the script is being indexed

Scripted Input Stanza

```
[script://<cmd>]
passAuth = <username>
host = <as indicated>
source = <defaults to script name>
sourcetype = <defaults to script name>
interval = <number in seconds or cron syntax>
```

Use **passAuth** to run the script as the specified OS user – Splunk passes the auth token via stdin to the script

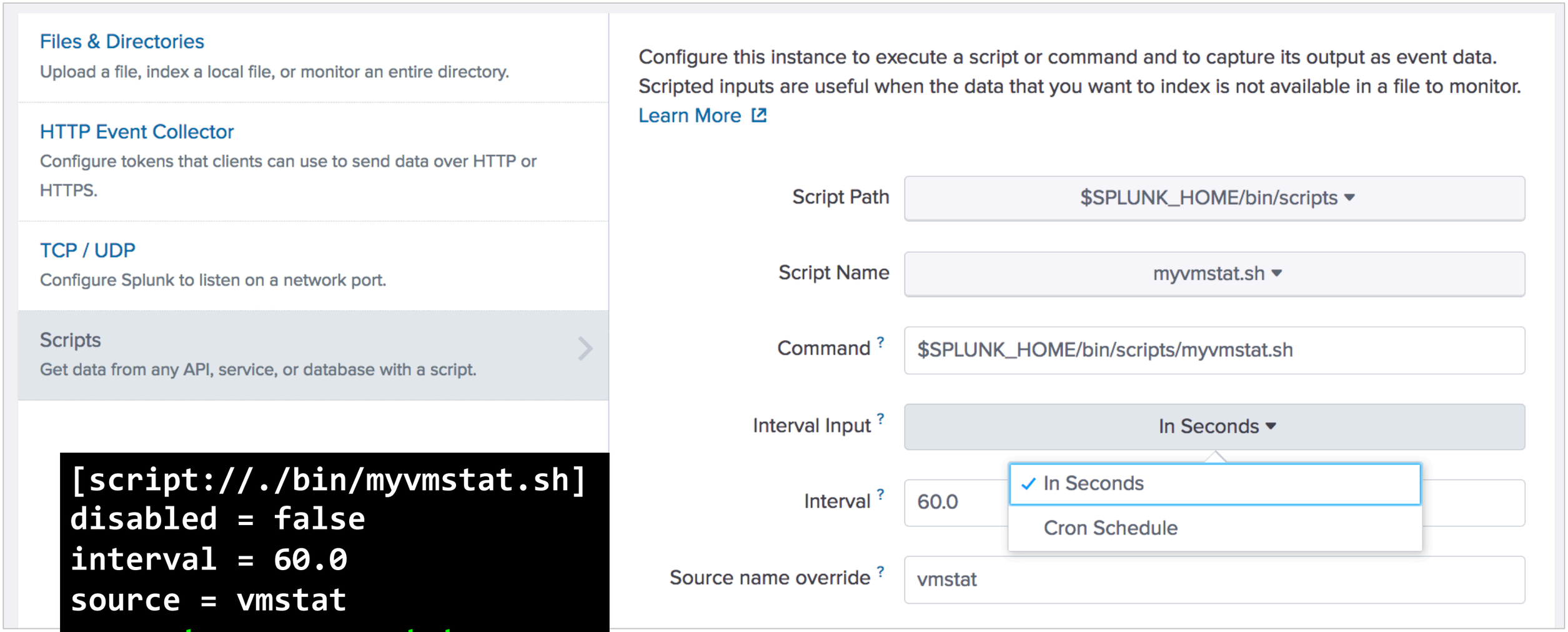
Interval is the time period between script executions (default: 60 seconds)

Warning

Splunk only executes scripts from:

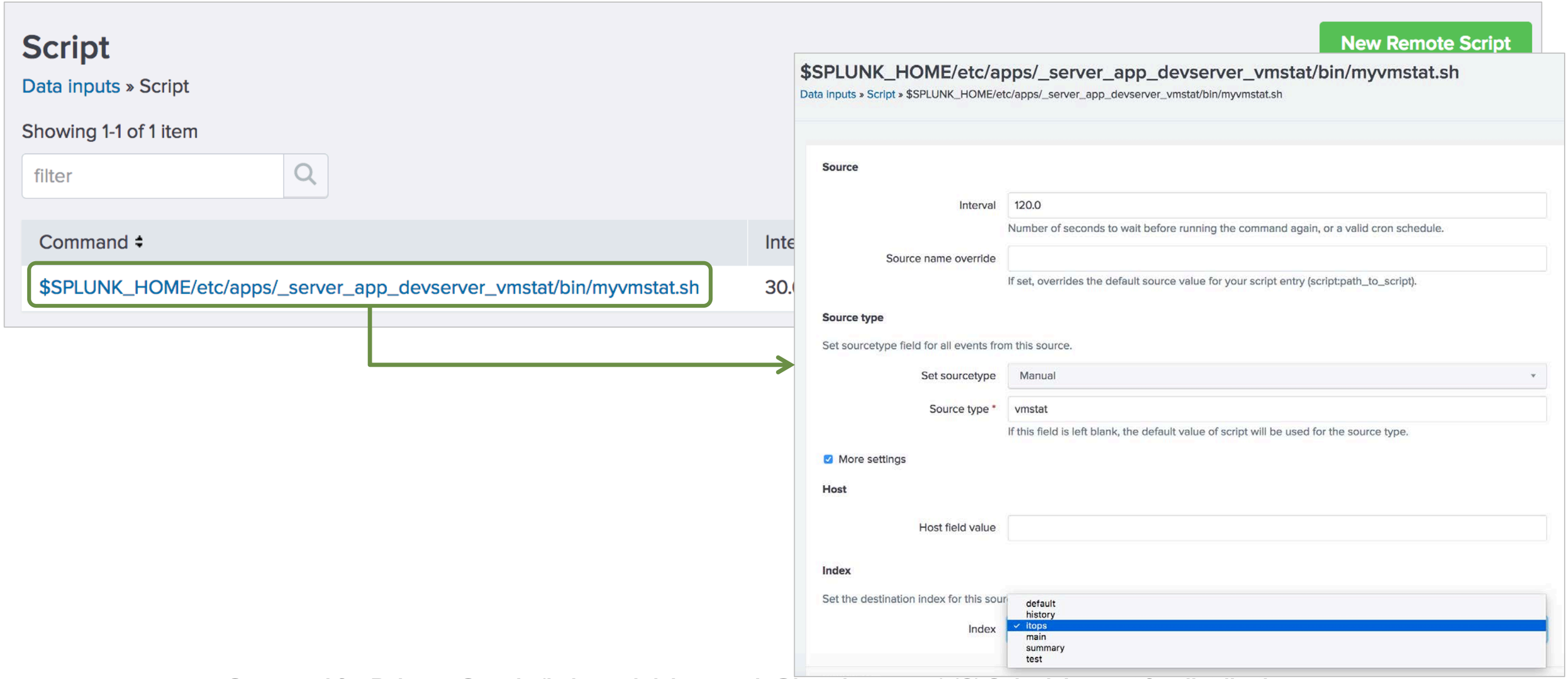
- **SPLUNK_HOME/etc/apps/<app_name>/bin**
- **SPLUNK_HOME/bin/scripts**
- **SPLUNK_HOME/etc/system/bin**

Scripted Inputs Example




```
[script://./bin/myvmstat.sh]
disabled = false
interval = 60.0
source = vmstat
sourcetype = myvmstat
```

Editing Scripted Inputs



Scripted Input Buffering

- Potential loss of data
  - Forwarder running the script is not able to connect to the indexer due to networking problems
- Workaround
  - The **`queueSize`** and **`persistentQueueSize`** attributes can be set for scripted input (in the **`[script://...]`** stanza)
  - Buffers data on the forwarder when the network or indexer is unavailable

Alternates to Using Scripted Input

Monitor a file containing the output of the script

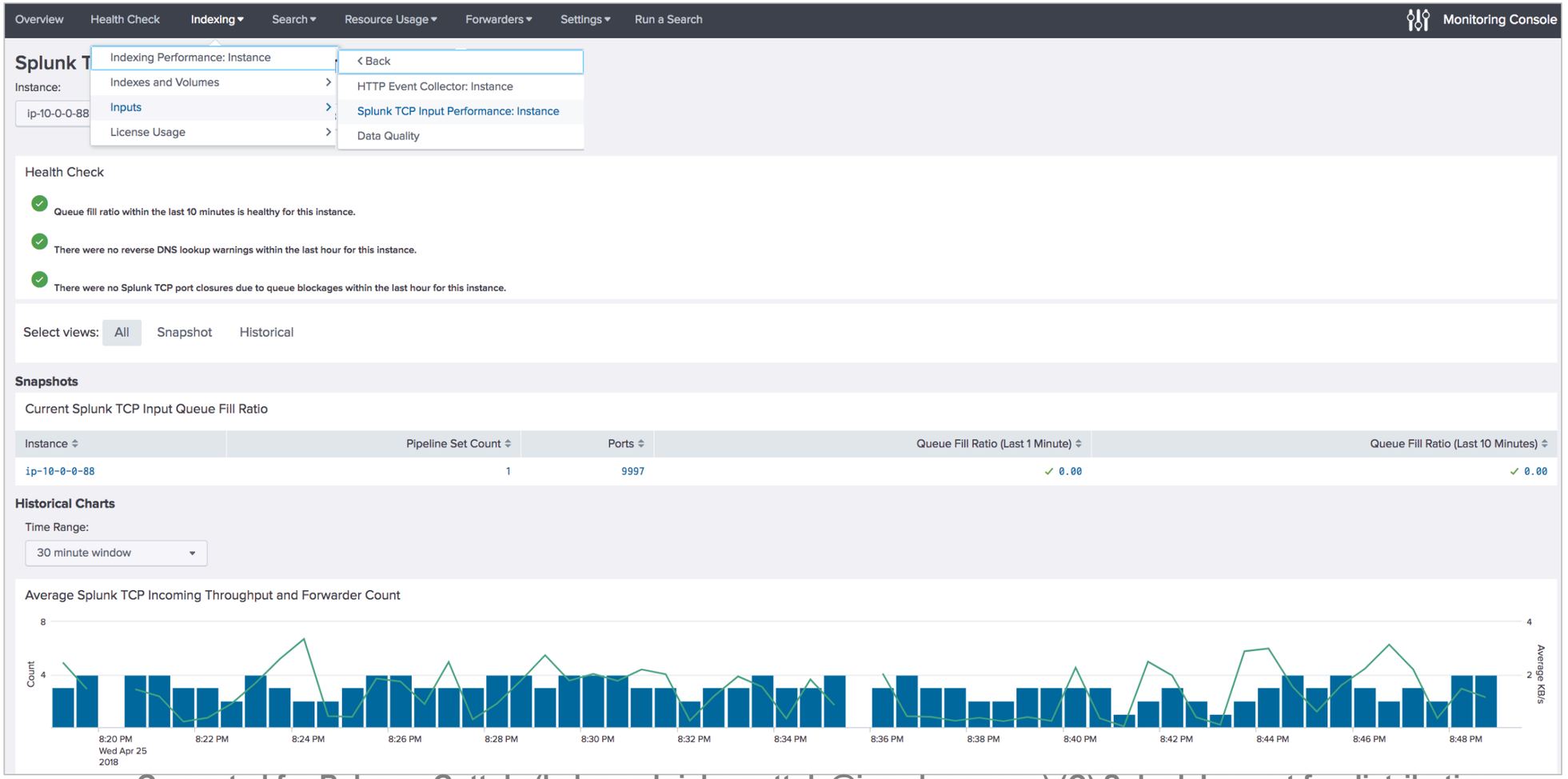
- Allows the use of Splunk's simple configuration of monitoring files
- Takes advantage of the file system and Splunk's robust file monitoring capabilities
- Can easily recover even when forwarder goes down
- Configured with a scripted log file:
  1. Schedule the script to run using an external scheduler (such as cron)
  2. Append script output to a log file
  3. Set up a monitor input to ingest the log file

Use Splunk's modular input

- Simple UI for configuring a scripted input
- Appears as its own type of input
- docs.splunk.com/Documentation/Splunk/latest/AdvancedDev/ModInputsScripts

Monitoring with MC: Splunk TCP Inputs

For remote input monitoring, click **Indexing > Inputs > Splunk TCP Input Performance**



Module 6 Knowledge Check

- True or False. Persistent Queue and Memory Queue can be applied to Network as well as Scripted inputs.
- Why is it a Best Practice to send data to a syslog collector that writes into a directory structure and then have a UF/HF ingest the data from the directory structure?
- True or False. An interval setting for scripted inputs can be specified in cron syntax.
- Is it possible to use the host value and not the DNS name or IP address for a TCP input? How?

Module 6 Knowledge Check – Answers

- True or False. Persistent Queue and Memory Queue can be applied to Network as well as Scripted inputs.

  True.

- Why is it a Best Practice to send data to a syslog collector that writes into a directory structure and then have a UF/HF ingest the data from the directory structure?

  If the UF has to be restarted, the **`_fishbucket`** will prevent data loss.

- True or False. An interval setting for scripted inputs can be specified in cron syntax.

  True. You can specify the interval in either number of seconds or cron syntax.

- Is it possible to use the host value and not the DNS name or IP address for a TCP input? How?

  Yes, it is possible. Under the stanza in **`inputs.conf`** set the **`connection_host`** to none and specify the host value.

Module 6 Lab Exercise – Network Input

Time: 15 – 20 minutes

Tasks:

- Create and test a simple TCP-based network input
- On the deployment/test server, add a test network input
- Modify the host value for the test network input
- Deploy the app to your forwarder

Lab Notes:

- Your instructor will run a script to send TCP data ports on the forwarder
- Use your assigned port to listen for the TCP data
- Deploy a remote scripted input

Module 7: Windows and Agentless Inputs

Module Objectives

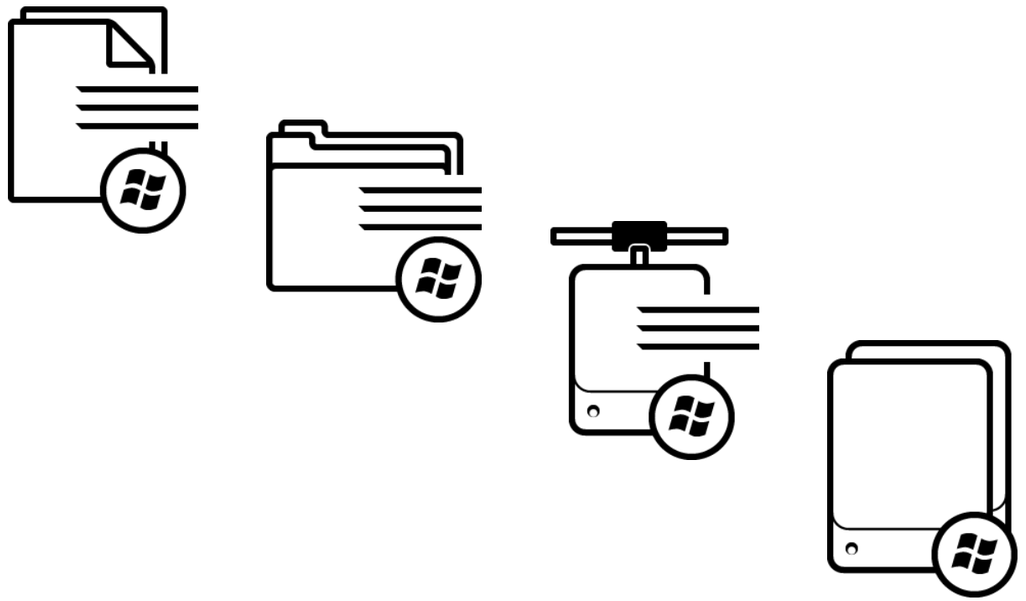
- Identify Windows specific **`inputs.conf`** stanzas and attributes
- Understand and configure Splunk HTTP Event Collector (HEC) agentless input
- Monitor HEC using MC (Monitoring Console)

Windows-Specific Inputs

Windows-Specific Inputs

- Generally stored in binary format (for example some state data and logs)
- Accessed using Microsoft APIs
- Use special Splunk input types
- Can be forwarded to an indexer running any OS platform
- May require that Windows Universal Forwarder run as a domain user

Windows



Windows-Specific Input Types

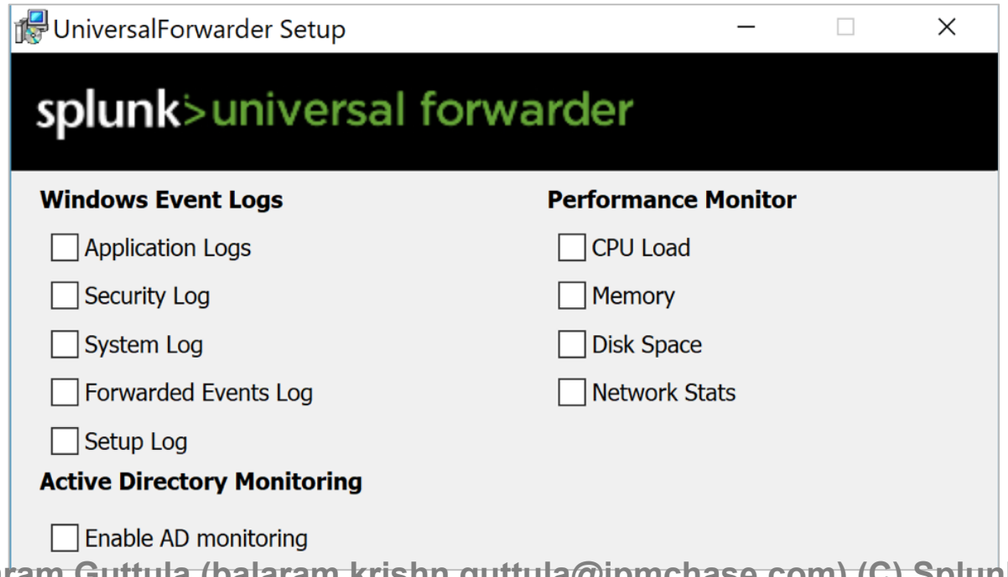
Input Type	Description
Event Log*	Consumes data from the Windows OS logs
Performance*	Consumes performance monitor data
Active Directory	Monitors changes in an Active Directory server
Registry	Monitors changes in a Windows registry
Host	Collects data about a Windows server
Network	Monitors network activity on a Windows server
Print	Monitors print server activity

* Supports both local and remote (WMI) data collection

Local Windows Inputs Syntax

- Configure inputs during the Windows Forwarder installation
- Optionally, configure manually
  - See **`inputs.conf.spec`** and **`inputs.conf.example`** for details on setting up each Windows input type

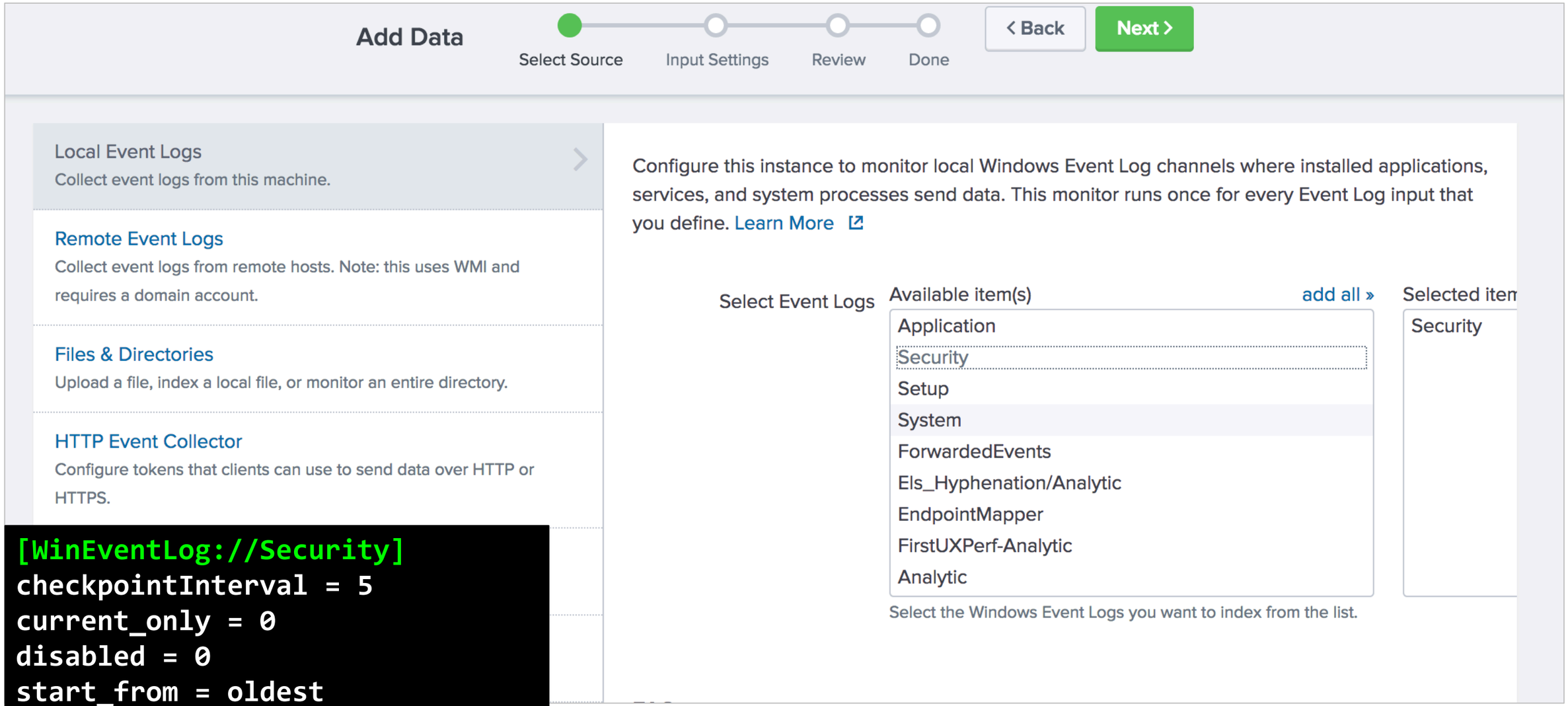
```
[admon://name]
[perfmon://name]
[WinEventLog://name]
[WinHostMon://name]
[WinNetMon://name]
[WinPrintMon://name]
[WinRegMon://name]
```



Note

While you can configure Windows inputs manually, Splunk recommends that you prepare the stanza using Splunk Web UI because it is easy to mistype the values for event log channels.

Windows Inputs: Using the Manager UI




```
[WinEventLog://Security]
checkpointInterval = 5
current_only = 0
disabled = 0
start_from = oldest
```

Windows Input Filtering Options

- Filter out non-essential events using whitelist and blacklist
  - Provides ability to keep specific events and filter out low-value events
  - Configure up to 10 whitelist and 10 blacklist per stanza
  - Set whitelist and blacklist based on event field names and regex:
    - **whitelist = <*list*> | *key=regex* [*key=regex*]**
    - **blacklist = <*list*> | *key=regex* [*key=regex*]**
  - In case of a conflict, the blacklist prevails

```
[WinEventLog://Security]
disabled=0
whitelist1= EventCode=/^[4|5].*$/ Type=/Error|Warning/
whitelist2= TaskCategory=%^Log.*$%
blacklist = 540
```

Windows Remote Inputs With WMI

- WMI is available for two types of Windows inputs:
  - Event logs
  - Performance monitor
- Advantage:
  - Collect information from Windows servers without installing a Splunk forwarder
- Disadvantage:
  - Uses WMI as a transport protocol
  - Not recommended in high latency networks
  - Requires Splunk to run as a domain account

Configuring WMI Inputs

- Remote inputs are configured in **wmi.conf**
- See **wmi.conf.spec** and **wmi.conf.example** for full details

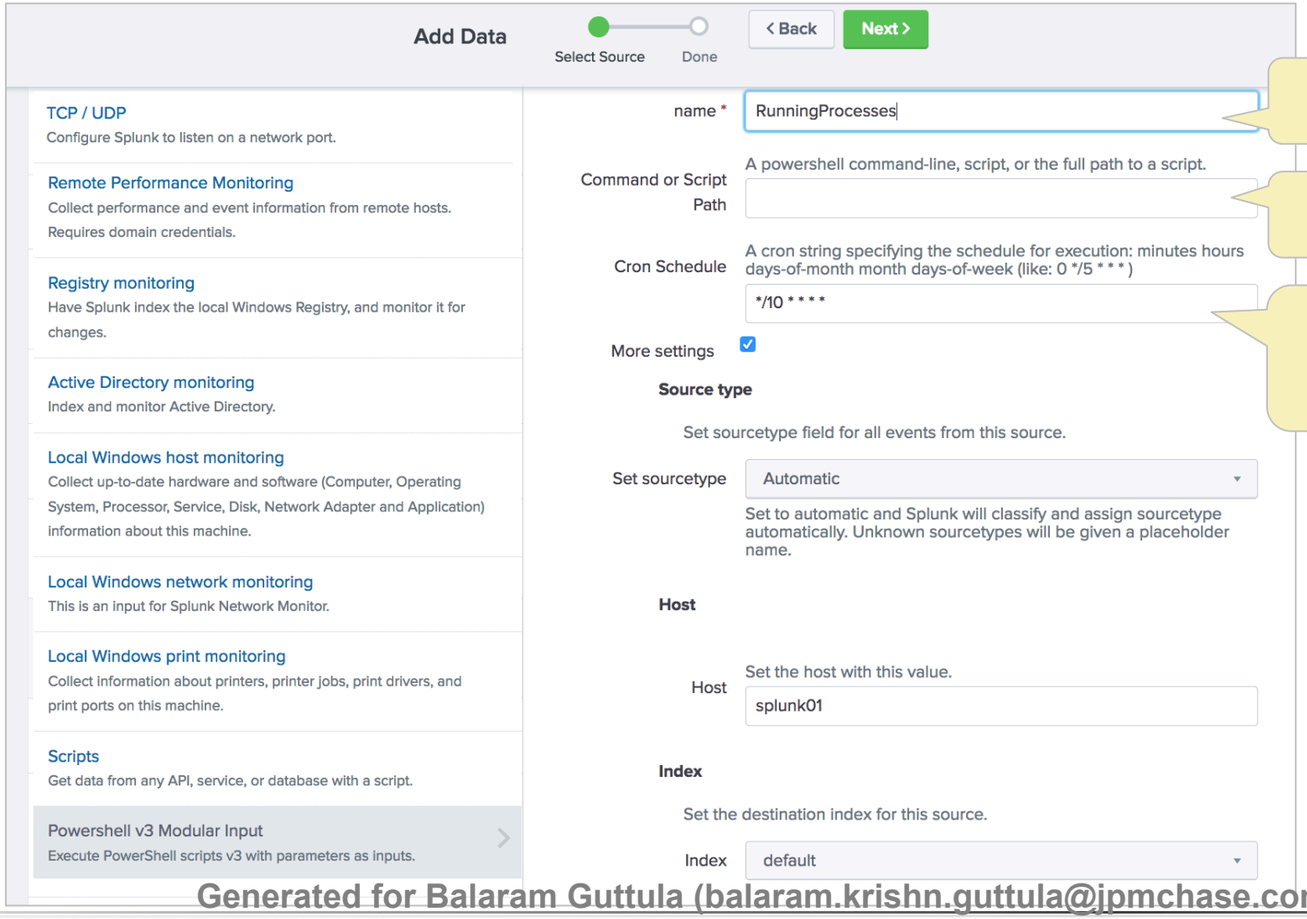
```
[WMI:remote-logs]
interval = 5
server = server1, server2, server3
event_log_file = Application, Security, System

[WMI:remote-perfmon]
interval = 5
server = server1,server2, server3
wql = Select DatagramsPersec
```


Special Field Extractions

- Several Microsoft products use a special multi-line header log format
  - Examples: IIS/W3C, JSON, and other delimited/structured sources
- Challenges:
  - These logs often get re-configured by the product administrator
  - Requires coordination between source administrator and Splunk administrator to sync the field extraction
- Solution:
  - Use indexed field extraction on the Windows forwarder
    - Normally the field extraction magic happens on the index/search tier

Powershell Input

- Uses built-in **`powershell.exe`** scripting facility in Windows
  - No custom external library dependencies



PowerShell v1 or v3

Command or a script file

Blank field executes once only

```
[powershell://<name>]
script = <command>
schedule = [<number>|<cron>]
```

Windows Inputs Resources

- About Windows data

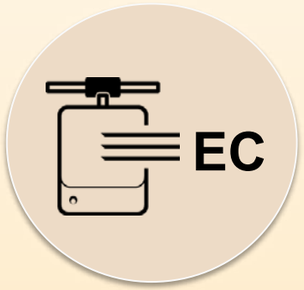
  http://docs.splunk.com/Documentation/Splunk/latest/Data/AboutWindowsdataandSplunk
- General information about event log

  https://docs.microsoft.com/en-us/windows/desktop/wes/windows-event-log
- Performance Counters Portal

  https://docs.microsoft.com/en-us/windows/desktop/PerfCtrs/performance-counters-portal
- Performance Counters Reference

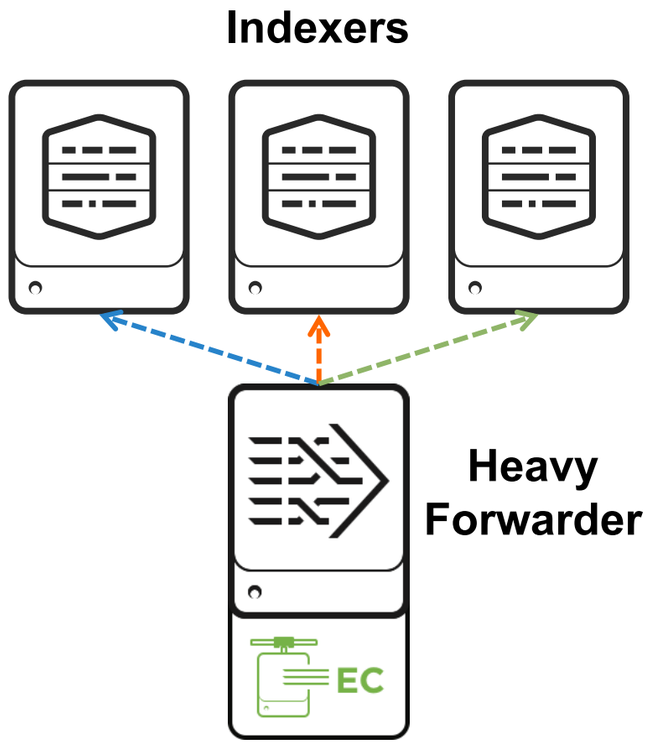
  https://docs.microsoft.com/en-us/windows/desktop/PerfCtrs/performance-counters-reference

HTTP Event Collector (HEC) Agentless Inputs



HTTP Event Collector (HEC)

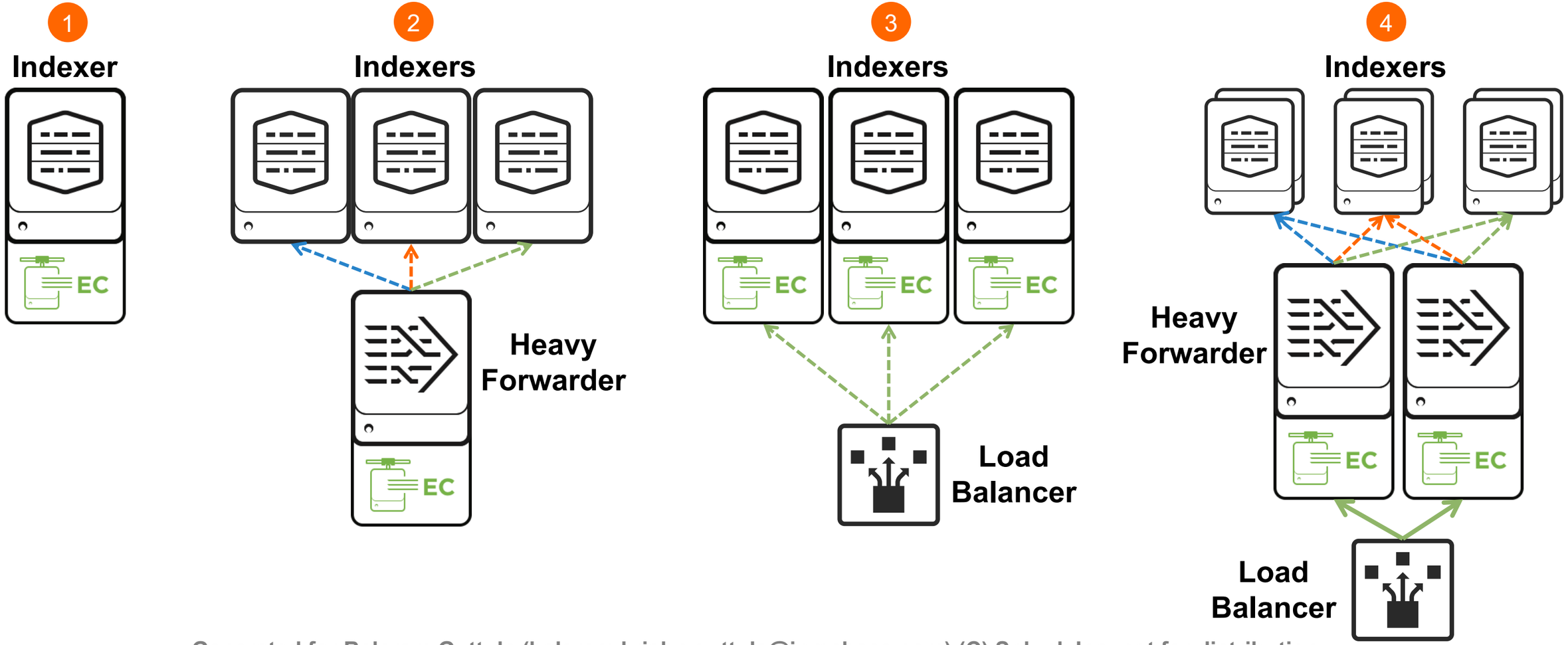
- A token-based HTTP input that is secure and scalable
- Sends events to Splunk without the use of forwarders (such as log data from a web browser, automation scripts, or mobile apps)
- Can facilitate logging from distributed, multi-modal, and/or legacy environments



Event collector enabled to receive HTTP events

Distributed HEC Deployment Options

HEC can scale by taking advantage of Splunk distributed deployment



Configuring HTTP Event Collector

1. Enable the HTTP event collector (disabled by default)
   - Navigate to **Settings > Data inputs > HTTP Event Collector**
   - Click **Global Settings > Enabled**
2. Generate a HTTP-input token by clicking New Token
   - The **Add Data** workflow starts
   - Name the input token and optionally set the default source type and index

Sending HTTP Events from a Device

- Create a request with its authentication header to include the input token
  - Can send data from any client
  - Simplify the process by using the Splunk logging libraries (which support JavaScript, Java and .NET)
- POST data in JSON format to the token receiver

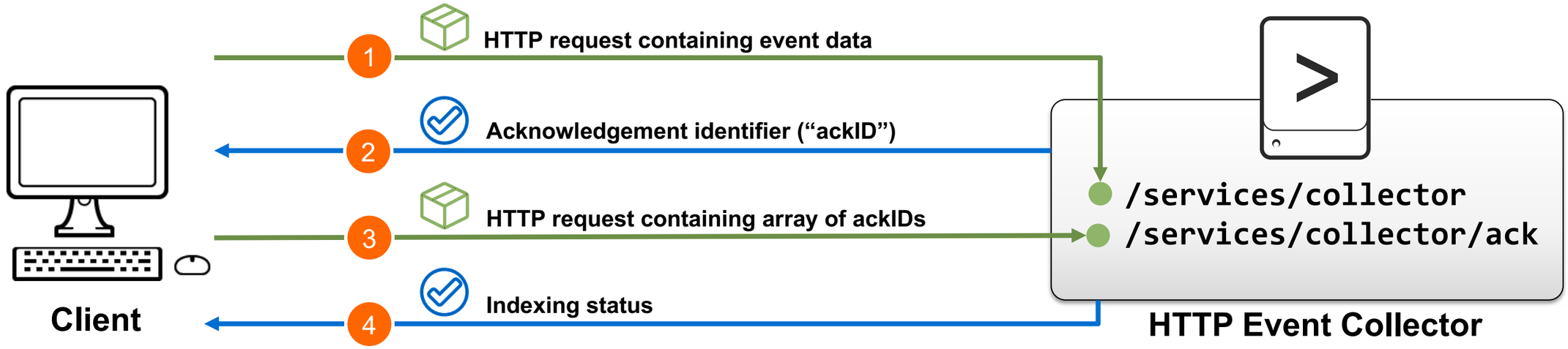
```
curl "http[s]://<splunk_server>:8088/services/collector"
 -H "Authorization: Splunk <generated_token>"
 -d '{
      "host":"xyz",
      "sourcetype":"fl01_S2",
      "source":"sensor125",
      "event": {"message":"ERR", "code":"401"}
     }'
```

HTTP Event Collector Options

- Enable HEC acknowledgments
- Send *raw* payloads
- Configure dedicated HTTP settings

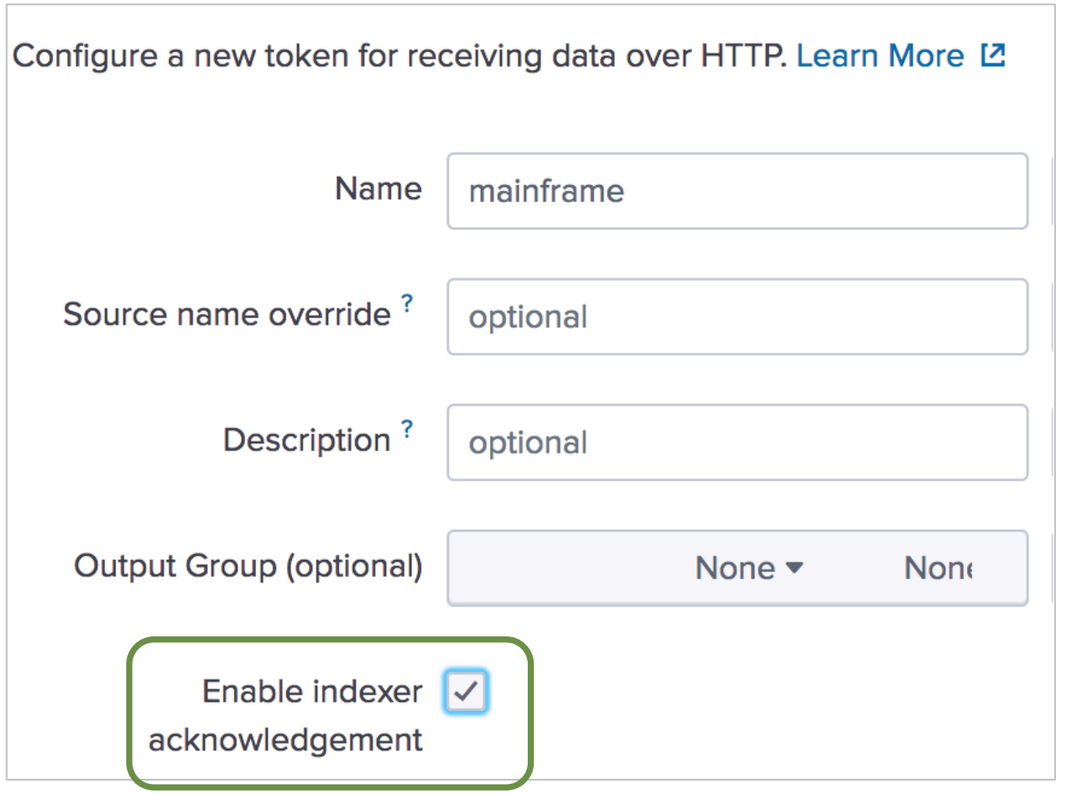
HEC Indexer Acknowledgement

1. Request sent from client to the HEC endpoint using a token, with indexer acknowledgment enabled
2. Server returns an acknowledgment identifier (**`ackID`**) to client
3. Client can query the Splunk server with the identifier to verify if all events in the send request have been indexed (HTTP request containing array of **`ackID`**'s)
4. Splunk server responds with status information of each queried request



HEC Indexer Acknowledgement Notes

- **ACK** is configured at the token level
- Each client request must provide a *channel* (a unique identifier created by the client)
- When an event is indexed, the channel gets the **ackID**
- Client polls a separate endpoint using one or more **ackID**'s
- Once an **ACK** has been received, it is released from memory
- Client polling functionality is not built into Splunk and requires custom programming



docs.splunk.com/Documentation/Splunk/latest/Data/AboutHECIDXAck

Sending Raw Payloads to HEC

- Example:
  - Application developers want to send data in a proprietary format
- Solution:
  - HEC allows any arbitrary payloads, not just JSON
- Configuration Notes:
  - No special configuration required
  - Must use channels similar to ACK
    - Supports ACK as well
  - Events MUST be bounded within a request

```
curl "http[s]://<splunk_server>:8088/services/collector/raw?channel=<client_provided_channel>"
 -H "Authorization: Splunk <generated_token>"
 -d 'ERR,401,-23,15,36'
```

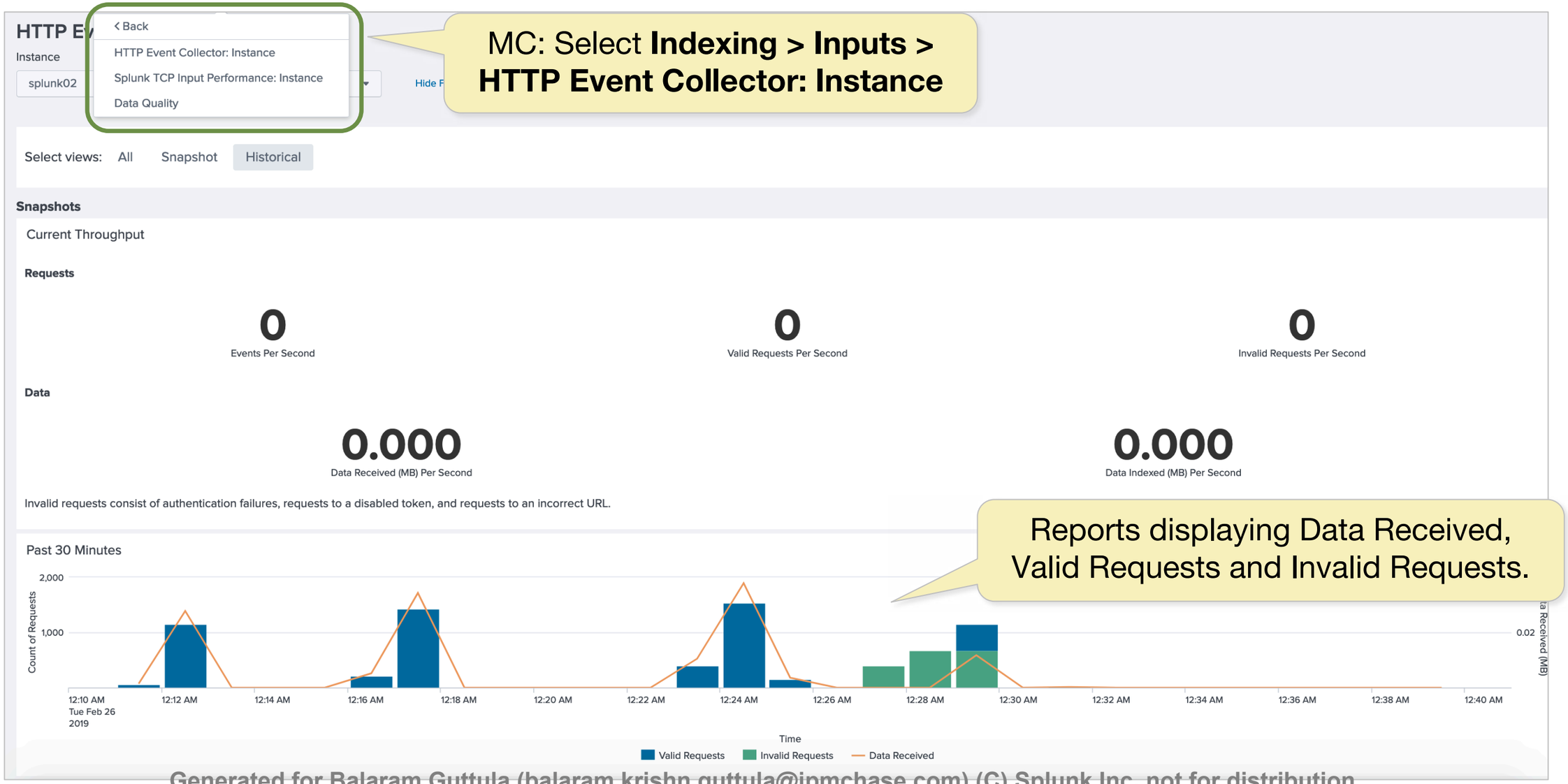

Configuring Dedicated HTTP Settings

- Example:
  - Splunk admins want to limit who can access the HEC endpoints
- Solution:
  - Manually add the dedicated server settings in **inputs.conf**
- Configuration Notes:
  - Available attributes under the **[http]** stanza
    - Configure a specific SSL cert for HEC and client certs
    - Enable cross-origin resource sharing (CORS) for HEC
    - Restrict based on network, hostnames, etc.

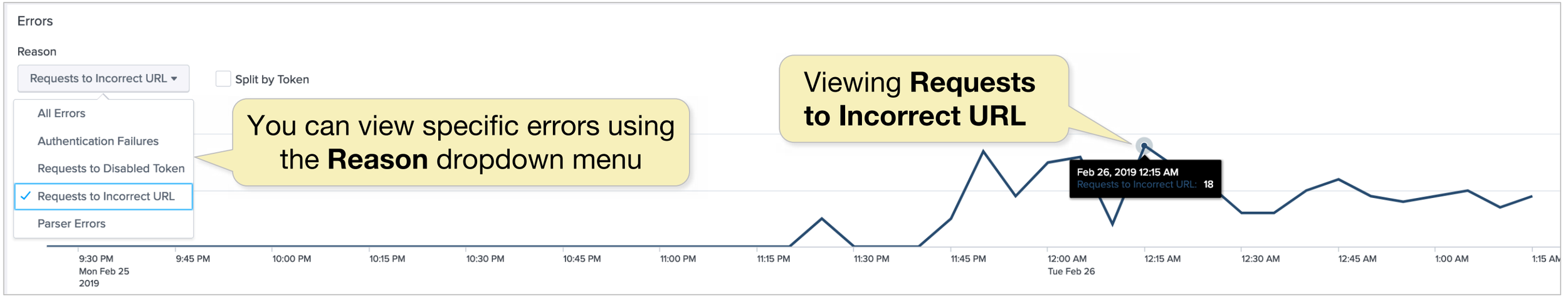
inputs.conf

```
[http]
enableSSL = 1
crossOriginSharingPolicy = *.splunk.com
acceptFrom = "!45.42.151/24, !57.73.224/19, *"
```


Monitoring HEC with MC



Monitoring HEC with MC – Viewing Errors



HTTP Event Collector (HEC) documentation

- Refer to:
  - Introduction to Splunk HTTP Event Collector
    dev.splunk.com/view/event-collector/SP-CAAAE6M
  - Blogs: Tips & Tricks on HTTP Event Collector
    blogs.splunk.com/2015/10/06/http-event-collector-your-direct-event-pipe-to-splunk-6-3

Module 7 Knowledge Check

- True or False. You can set up a windows input using a UF on the windows server and send the data to an Indexer running on Linux.
- True or False. You can collect Active Directory data from a Windows Server remotely using **`wmi.conf`**.
- True or False. Event Collector can be set up on a UF.
- True or False. Data can be sent in json or any raw data format to the event collector.

Module 7 Knowledge Check – Answers

- True or False. You can set up a windows input using a UF on the windows server and send the data to an Indexer running on Linux.

  True.

- True or False. You can collect Active Directory data from a Windows Server remotely using **`wmi.conf`**.

  False. Only event logs and performance monitoring logs can be collected using **`wmi.conf`**.

- True or False. Event Collector can be set up on a UF.

  False. Event collector can be set up on an Indexer or HF.

- True or False. Data can be sent in json or any raw data format to the event collector.

  True.

Module 7 Lab Exercise – HTTP Event Collector

Time: 10 – 15 minutes

Tasks:

- Enable HTTP event collector on the deployment/test server
- Create a HTTP event collector token
- Send HTTP events from your UF1 (**`10.0.0.50`**)

Module 8:
Fine-tuning Inputs

Module Objectives

- Understand the default processing that occurs during input phase
- Configure input phase options, such as source type fine-tuning and character set encoding

Testing New Inputs

- Use **Data Preview**
  - Evaluate new data sources without actually inputting them
- Create a test index
  - Send test inputs to this index
  - Delete index and start again as desired
  - Does not require **`splunkd`** restart
- Create a Splunk test environment
  - Use a laptop, virtual machine, or spare server
  - Send test inputs for new sources and evaluate, prior to production
  - Should be running same version of Splunk as in production

Things to Get Right at Index Time

Input phase	• Host • Source type • Source • Index
Parsing phase	• Line breaking (event boundary) • Date/timestamp extraction • Adjust all meta fields* • Mask raw data* • Eliminate events*

* Optional

What if I Don't Get It Right?

- On a testing / development system
  - This is what test/dev Splunk setups are for!
  - Clean or delete/recreate the test index, change your configurations, and try again
  - May need to reset the fishbucket
- On a production server
  - Leave the erroneous data in the system until it naturally “ages out” (reaches the index size or retention time limits)
  - Attempt to ***delete*** the erroneous data
  - Only re-index when it is absolutely necessary

The **`props.conf`** File

- Config file referenced during all phases of Splunk data processing (inputs, indexing, parsing and searching)
- Documentation:
  - The **`props.conf.spec`** and **`props.conf.example`** files in **`SPLUNK_HOME/etc/system/README`**
  - docs.splunk.com/Documentation/Splunk/latest/admin/Propsconf

Stanzas in props.conf

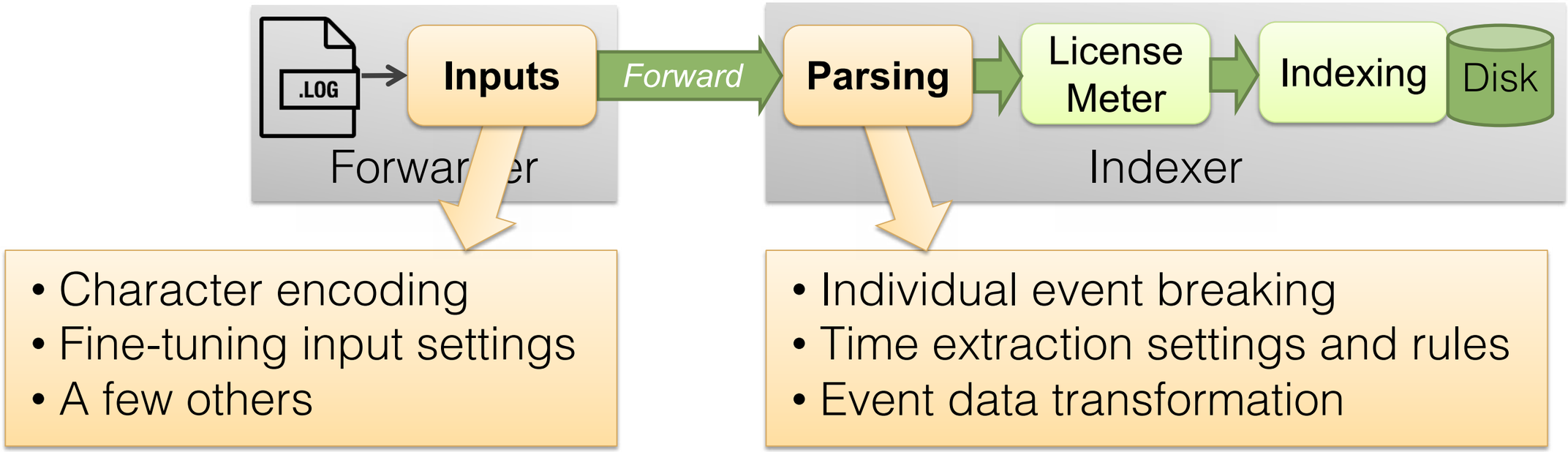
- All data modifications in **`props.conf`** are based on either source, sourcetype, or host

syntax	example
`[source::source_name]` `attribute = value`	`[source::/var/log/secure*]` `sourcetype = linux_secure`
`[host::host_name]` `attribute = value`	`[host::nyc*]` `TZ = US/Eastern`
`[sourcetype]` `attribute = value`	`[sales_entries]` `CHARSET = UTF-8`

- You can use wildcards (`*`) and regex in the **`source::`** and **`host::`** stanzas

Input and Parsing Phases and **`props.conf`**

- Settings from **`props.conf`** applied during phases:



- Configure **`props.conf`** on the appropriate Splunk instances

  wiki.splunk.com/Where_do_I_configure_my_Splunk_settings

Character Encoding

- During the input phase, Splunk sets all input data to UTF-8 encoding by default
  - Can be overridden, if needed, by setting the **CHARSET** attribute

```
[source::/var/log/locale/korea/*]
CHARSET=EUC-KR

[sendmail]
CHARSET=AUTO
```

- Use **AUTO** to attempt automatic encoding based on language

docs.splunk.com/Documentation/Splunk/latest/Data/Configurecharactersetencoding

Fine-tuning Directory Monitor Source Types

- When you add a directory monitor:
  - Specify a **sourcetype** to apply it to all files (contained recursively under that directory)
  - Omitting the **sourcetype** causes Splunk to try to use automatic pre-trained rules
- Override specific source types selectively in **props.conf**
  - Identify input with a **[source::<*source*>]** stanza and set the **sourcetype** attribute
  - Place this configuration on the source server, as this is an input phase process

inputs.conf

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

props.conf

```
[source::/var/log/mail.log]
sourcetype=sendmail

[source::/var/log/secure/]
sourcetype=secure
...
```

Note

If you explicitly set the source type in **inputs.conf** for a given source, you cannot override the source type value for the source in **props.conf**

Module 8 Knowledge Check

- In the **`props.conf`** example below, what is **`sendmail`**?

```
[sendmail]
CHARSET=AUTO
```

- Examine the **`props.conf`** example below. Is this an acceptable format for the stanzas?

```
[source::/var/…/korea/*]
CHARSET=EUC-KR

[sendm*]
CHARSET=AUTO
```


Module 8 Knowledge Check – Answers

- In the **props.conf** example below, what is **sendmail**?

```
[sendmail]
CHARSET=AUTO
```

It is a source type in **props.conf**. Source types are specified as a string value in the stanza without the **sourcetype::** prefix.

- Examine the **props.conf** example below. Is this an acceptable format for the stanzas?

```
[source::/var/…/korea/*]
CHARSET=EUC-KR

[sendm*]
CHARSET=AUTO
```

No. You cannot use a wildcard with source types in **props.conf**.

Module 8 Lab Exercise – Fine-Tuning Inputs

Time: 10 – 15 minutes

Tasks:

- Add a test directory monitor to sample the auto-sourcetype behavior
  - Make note of the source type value
- Override the auto-sourcetyping of a specific source by adding a source type declaration in **`props.conf`**
- Deploy it to your forwarder and check again

  Note: These input files are not being updated. Therefore, you must reset the file pointer and re-index the files

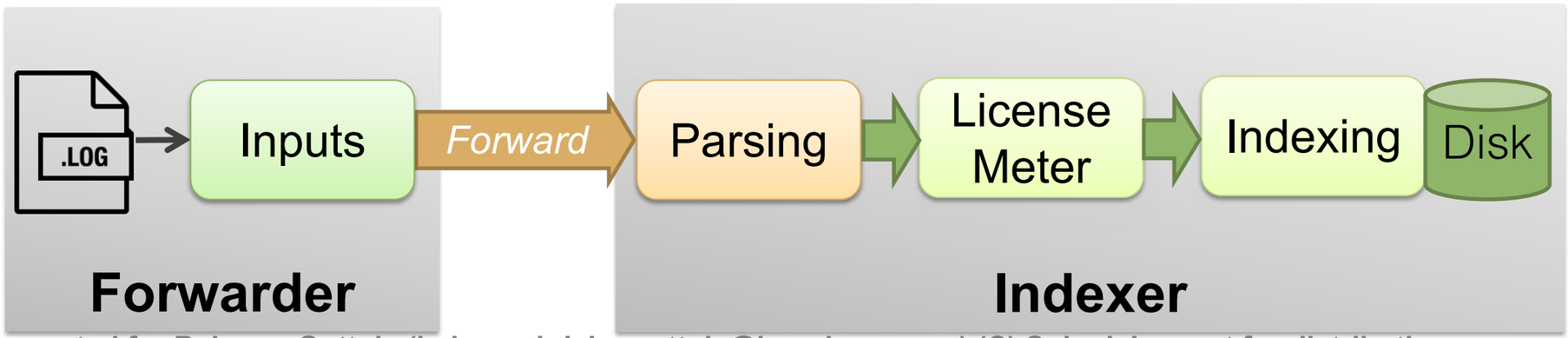
Module 9: Parsing Phase and Data Preview

Module Objectives

- Understand the default processing that occurs during parsing
- Optimize and configure event line breaking
- Explain how timestamps and time zones are extracted or assigned to events
- Use Data Preview to validate event creation during the parsing phase

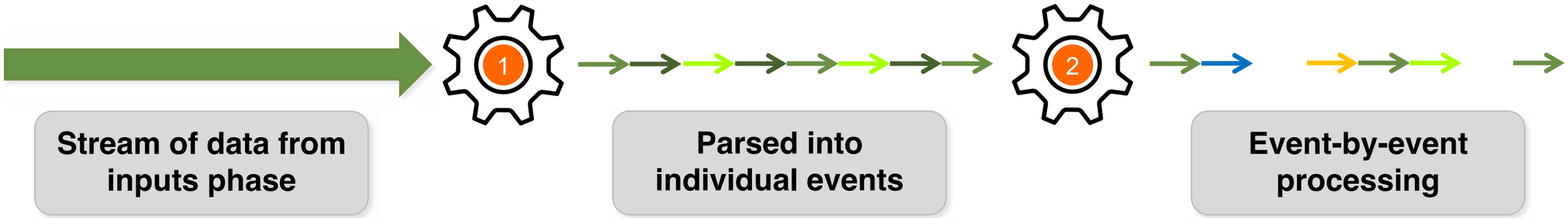
The Parsing Phase

- Occurs as data arrives at the indexer (or heavy forwarder)
- Breaks up input data stream into discrete **events**, each with a **timestamp** and **time zone**
- Creates, modifies, and redirects events
  - Applies additional transformation steps to modify the metadata fields or modify raw data



Event Creation

- Occurs during the parsing phase
  1. Data from input phase is broken up into individual events
  2. Event-level processing is performed



- Relies on **event boundaries**: distinguishing where events begin and end
  - Usually determined by line breaks
  - May be determined by other settings in **`props.conf`**
- Should be verified using **Data Preview**, with new source types

Determining Event Boundaries

- Performed in two steps:
  1. **Line breaking**
     - Splits the incoming stream of bytes into separate lines
     - Configured with **`LINE_BREAKER = <regular_expression>`**
     - Default is any sequence of new lines and carriage returns: **`([\r\n]+)`**
  2. **Line merging** (optional)
     - Takes separate lines and merges them to make individual events
     - Configured with **`SHOULD_LINEMERGE = true`** (default)
     - Uses additional settings to determine how to merge lines (such as **`BREAK_ONLY_BEFORE`**, **`BREAK_ONLY_BEFORE_DATE`**, and **`MUST_BREAK_AFTER`**)
     - Disabled if set to **`false`**, which improves performance

docs.splunk.com/Documentation/Splunk/latest/Data/Configureeventlinebreaking

Event Boundary Examples

Monitored input: Single line input with 3 events

```
[19/Sep/2019:18:22:32] VendorID=7033 Code=E AcctID=4390644811207834 ↲
[19/Sep/2019:18:22:48] VendorID=1239 Code=K AcctID=5822351159954740 ↲
[19/Sep/2019:18:22:59] VendorID=1243 Code=F AcctID=8768831614147676 ↲
```

props.conf

```
[sourcetype1]
LINE_BREAKER = ([\r\n]+)
SHOULD_LINEMERGE = false
```

Monitored input: Multi-line input with 3 events

```
Sep 12 06:11:58 host1.example.com storeagent[49597] <Critical>: Starting update scan ↲
Sep 12 06:11:58 host1.example.com storeagent[49597] <Critical>: UpdateController: Message tracing { ↲
       "power_source" = ac; ↲
       "start_date" = "2014-08-21 20:10:39 +0000"; ↲
} ↲
Sep 12 06:11:58 host1.example.com storeagent[49597] <Critical>: Asserted BackgroundTask power ↲
```

props.conf

```
[sourcetype2]
LINE_BREAKER = ([\r\n]+)
SHOULD_LINEMERGE = true
BREAK_ONLY_BEFORE_DATE = true
```


Date/timestamp Extraction

- Correct date/timestamp extraction is essential
- Always verify timestamps when setting up new data types
  - Pay close attention to timestamps during testing/staging of new data
  - Check UNIX time or other non-human readable timestamps
- Splunk works well with standard date/time format and well-known data types
- Custom timestamp extraction is specified in **`props.conf`**

Using **TIME_PREFIX**

- Syntax: **TIME_PREFIX = <*REGEX*>**
- Matches characters right BEFORE the date/timestamp
  - Use this syntax to specify where the timestamp is located in the event
    - Example data with "date-like" code at the start of the line

```
1989/12/31 16:00:00 ed May 23 15:40:21 2015 ERROR UserManager - Exception thrown
```

Start looking here for date/timestamp

props.conf

```
[my_custom_source_or_sourcetype]
TIME_PREFIX = \d{4}/\d{2}/\d{2} \d{2}:\d{2}:\d{2} \w+\s
```


Using **MAX_TIMESTAMP_LOOKAHEAD**

- Syntax: **MAX_TIMESTAMP_LOOKAHEAD = <*integer*>**
- Specifies how many characters to look beyond the start of the line for a timestamp
  - Works in conjunction with **TIME_PREFIX**
  - If set, it starts counting from the point the **TIME_PREFIX** indicates Splunk should start looking for the date/timestamp
  - Improves efficiency of timestamp extraction
  - The complete timestamp string must be present within the specified range

Using **TIME_FORMAT**

- Syntax: **TIME_FORMAT = <*strptime-style format*>**
- Specifies the format of the timestamp using a **strptime()** expression
  - For example, **2015-12-31** would be **%Y-%m-%d**
- For more detail and other options, check:
  - **SPLUNK_HOME\etc\system\README\props.conf.spec**
  - docs.splunk.com/Documentation/Splunk/latest/Data/ConfigureTimestampRecognition
  - docs.splunk.com/Documentation/Splunk/latest/Data/Handleeventtimestamps

Setting Time Zones – Splunk's Rules

- Use time zone offsets to ensure correct event time
- Splunk applies time zones in this order:
  1. A time zone indicator in the raw event data
     - **-0800**, **GMT-8** or **PST**
  2. The value of a TZ attribute set in **props.conf**
     - Checks the **host**, **source**, or **sourcetype** stanzas
  3. If a forwarder is used, the forwarder-provided time zone is used
     - en.wikipedia.org/wiki/List_of_zoneinfo_timezones
  4. If all else fails, Splunk applies the time zone of the indexer's host server

props.conf

```
[host::nyc*]
TZ = America/New York

[source::/mnt/cn_east/*]
TZ = Asia/Shanghai
```

Splunk Event Timestamp Processing

1. • Use **`TIME_FORMAT`** (from **`props.conf`**) to identify a timestamp in event
2. • If no **`TIME_FORMAT`** configured: Try to automatically identify timestamp from event
3. • If identify time+date, but no year: Determine a year
4. • If identify time, but no date: Try to find date in source name or file name
5. • If cannot identify a date: use file modification time
6. • Else no timestamp found:
   - If any timestamp from same source, use the most recent timestamp.
   - If no timestamps, use the current system time when indexing the event

http://docs.splunk.com/Documentation/Splunk/latest/Data/HowSplunkextractstimestamps

Using Splunk Data Preview

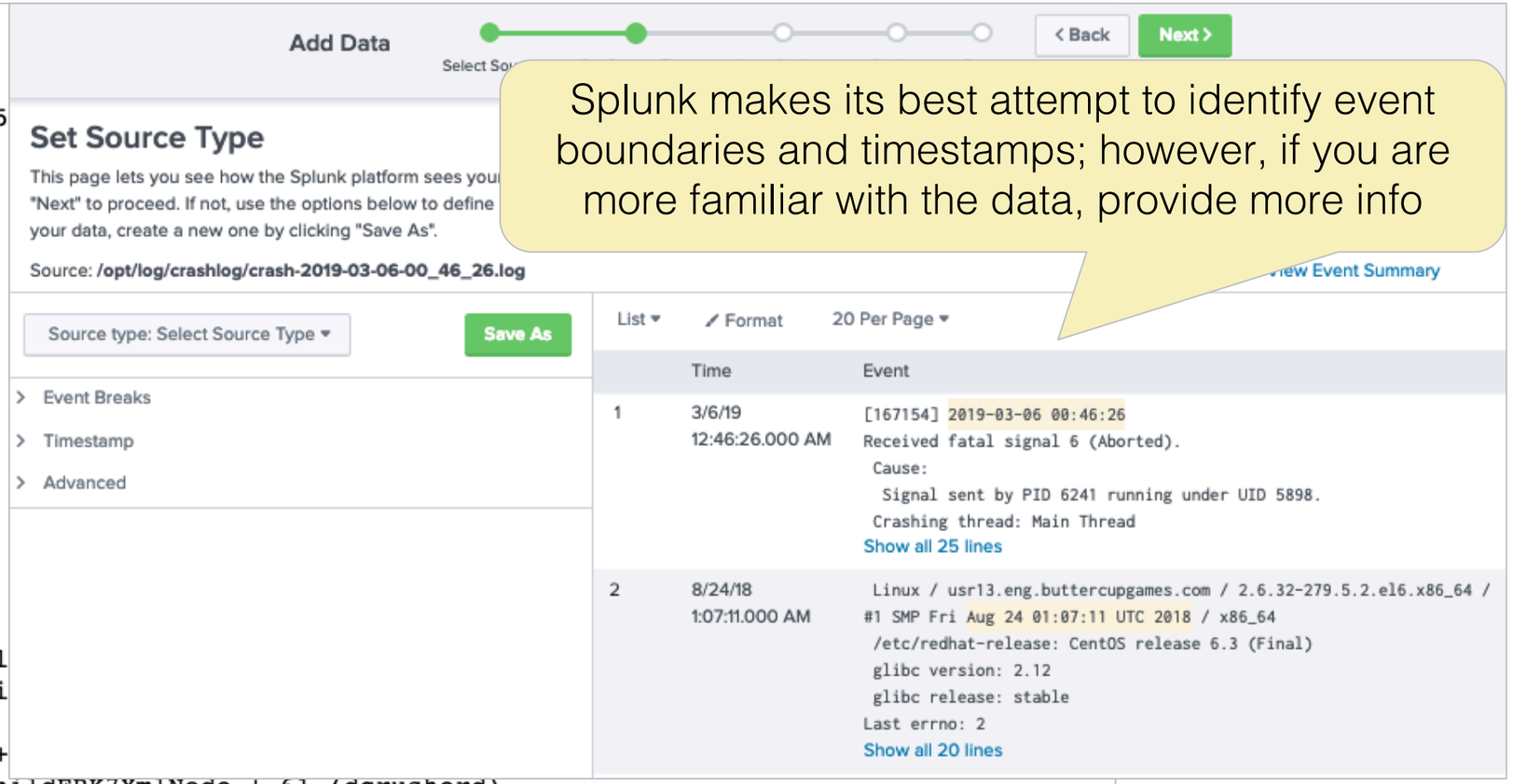
- Splunk attempts to auto-detect a source type
  - Alternatively select from a list or define your own source type
  - Supports both unstructured and structured data sources
  - CSV, JSON, W3C/IIS, XML, etc.
- Event breaking and date/timestamp settings are evaluated
  - Use your sandbox environment or test index to perfect your settings before taking a new data input into the production environment
- Use Data Preview configuration settings to create new source types

Previewing Unstructured Data

```
[167154] 2019-03-06 00:46:26
Received fatal signal 6 (Aborted).
 Cause:
  Signal sent by PID 6241 running under UID 5
 Crashing thread: Main Thread
 Registers
   RDI:  [0x00000B0500000C09]
   RSI:  [0x0F0097000009A300]
   RBP:  [0x0000000000002000]
   RSP:  [0x004B000000000D00]
   RAX:  [0x00042000010D0000]
   RBX:  [0x3005000000100000]
   RCX:  [0xE0E000000C010000]
   RDX:  [0x0000000A00000C00]
   EFL:  [0x0000000000020000]

 OS: Linux
 Arch: x86-64

 Backtrace:
  [0x04050A000000D000] gsignal + 53 (/lib64/l
  [0x0600000000000000] abort + 373 (/lib64/li
  [0x000C000000000000] ? (/lib64/libc.so.6)
  [0x80000000090300B0] __assert_perror_fail +
  [0x0F0000000E00B000] _ZN11XmlDocument8addChildERK7XmlNode + 61 (dcrusherd)
  [0x0800000070500C00] _Z18getSearchConfigXMLR11XmlDocumentPKPKc + 544 (dcrusherd)
  [0x0000100000000000] _Z22do_search_process_impliPKPKcP12BundlesSetupb + 6141 (dcrusherd)
 Linux / usr13.eng.buttercupgames.com / 2.6.32-279.5.2.el6.x86_64 / #1 SMP Fri Aug 24 01:07:11 UTC 2018 / x86_64
 /etc/redhat-release: CentOS release 6.3 (Final)
 glibc version: 2.12
 glibc release: stable
Last errno: 2
```

1

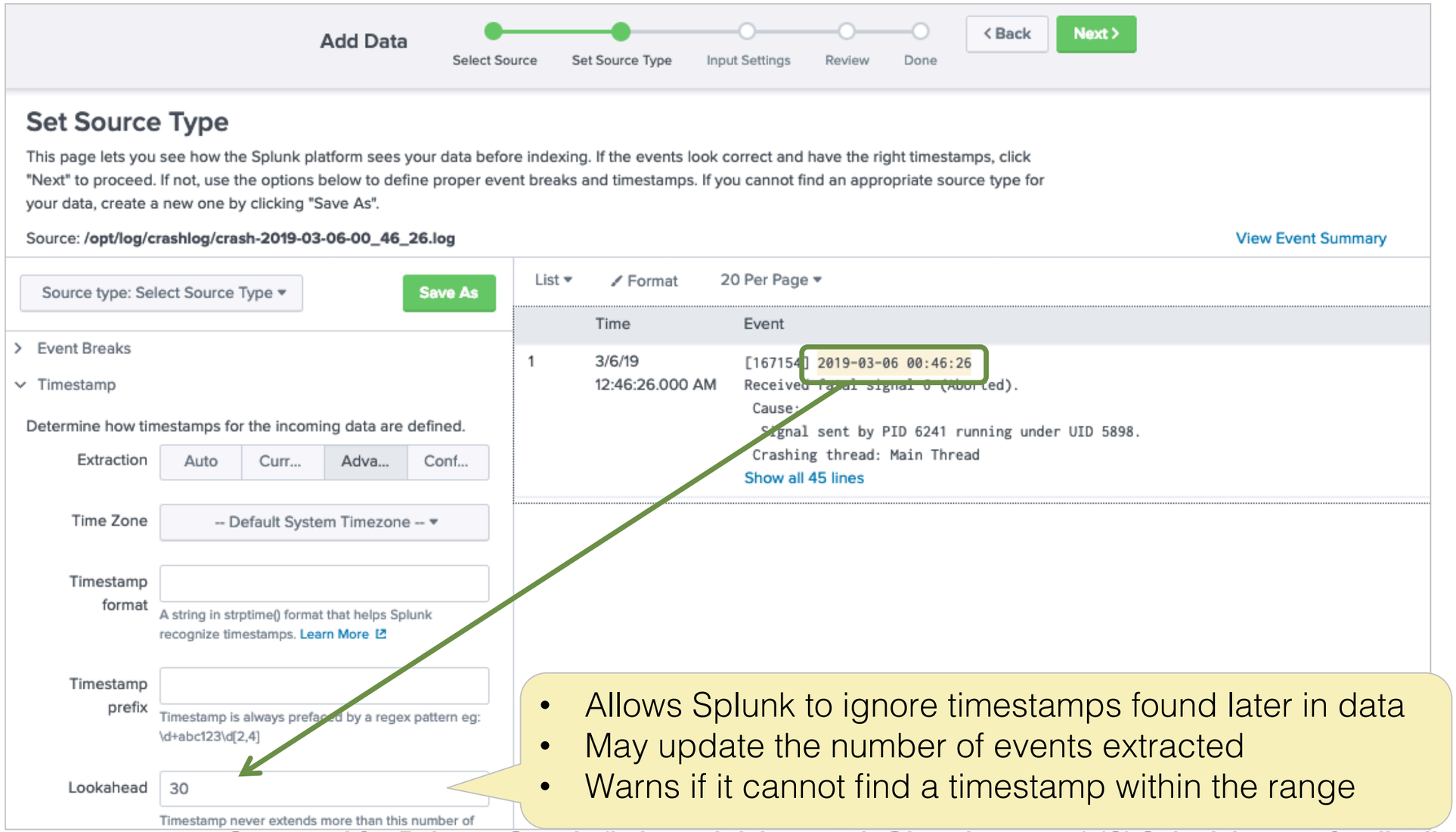
2



Splunk makes its best attempt to identify event boundaries and timestamps; however, if you are more familiar with the data, provide more info

 | 29 May 2020

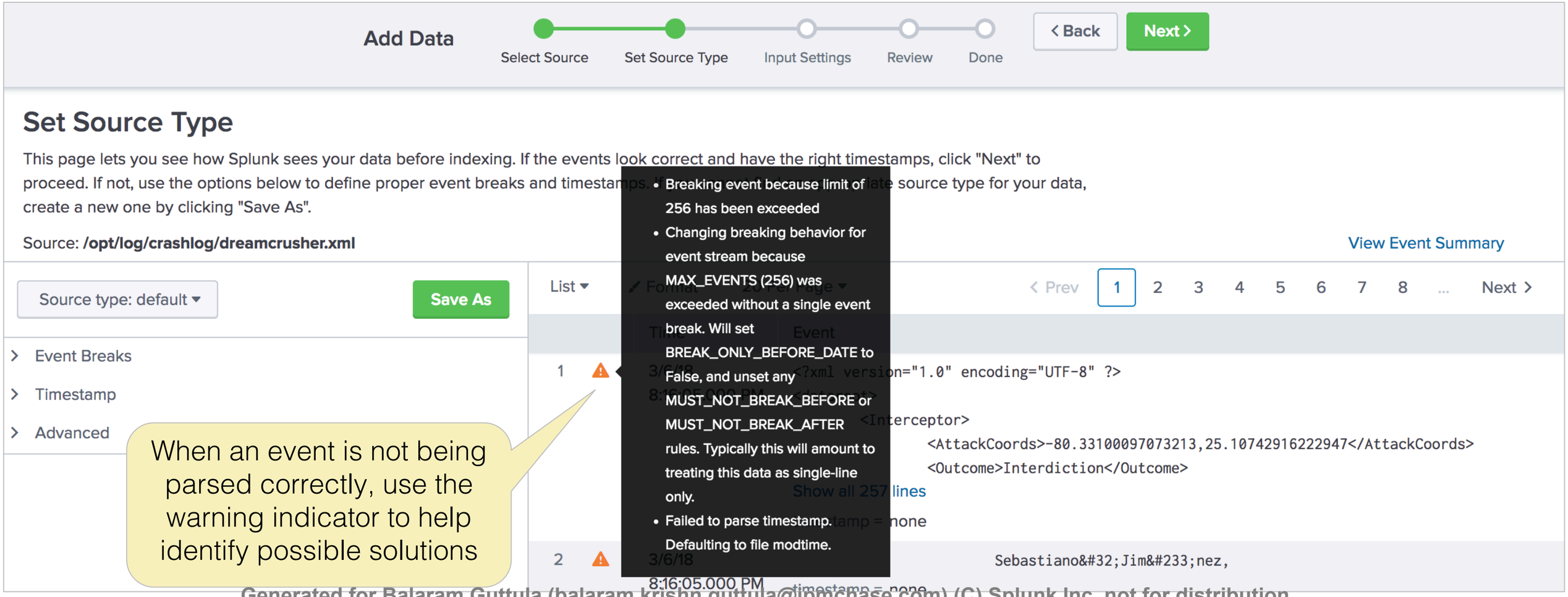
Previewing Unstructured Data (cont.)



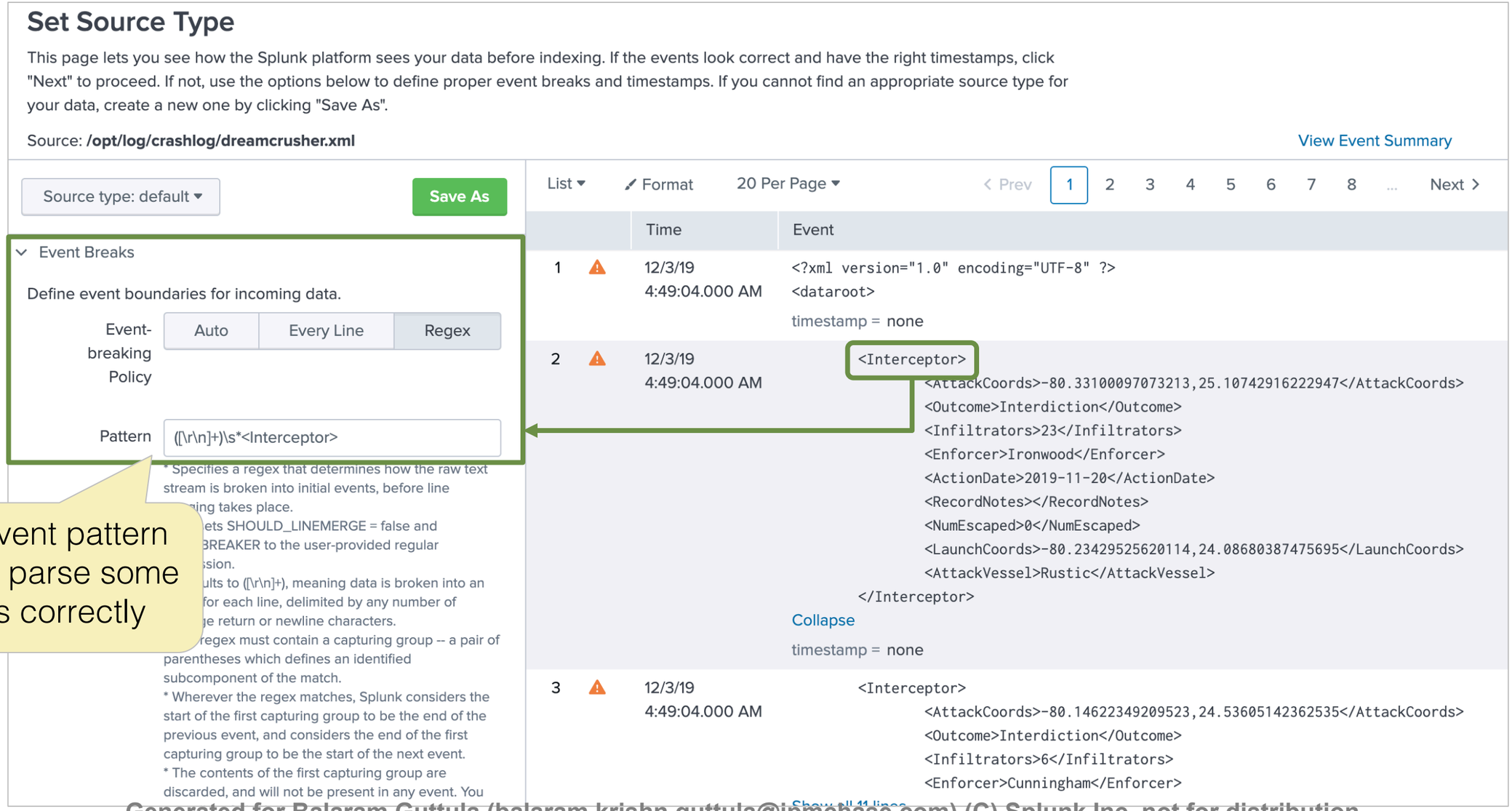
- Allows Splunk to ignore timestamps found later in data
- May update the number of events extracted
- Warns if it cannot find a timestamp within the range

Previewing Unstructured Data (cont.)

Another example – previewing **XML** file as unstructured data

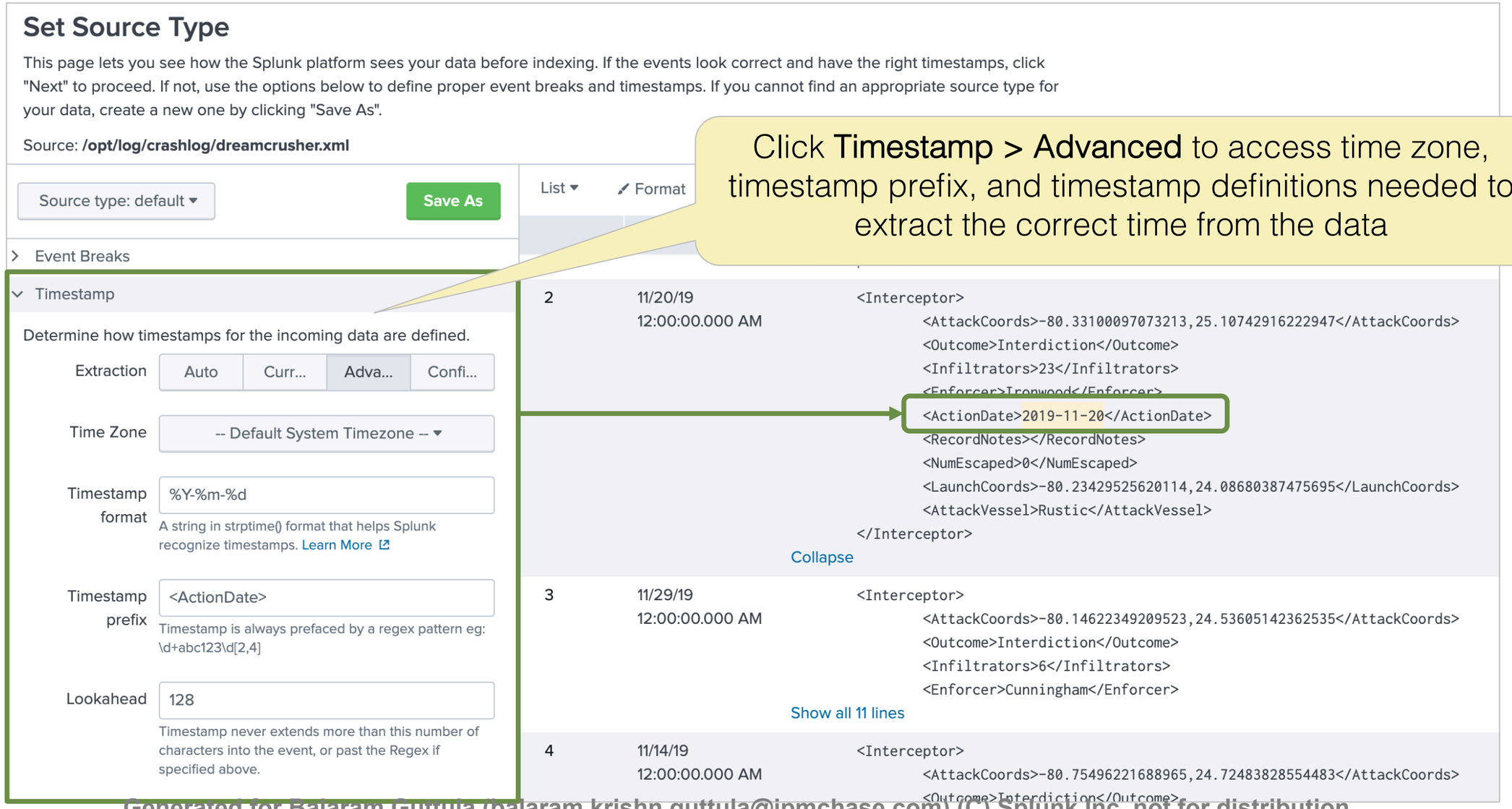


Previewing Unstructured Data (cont.)

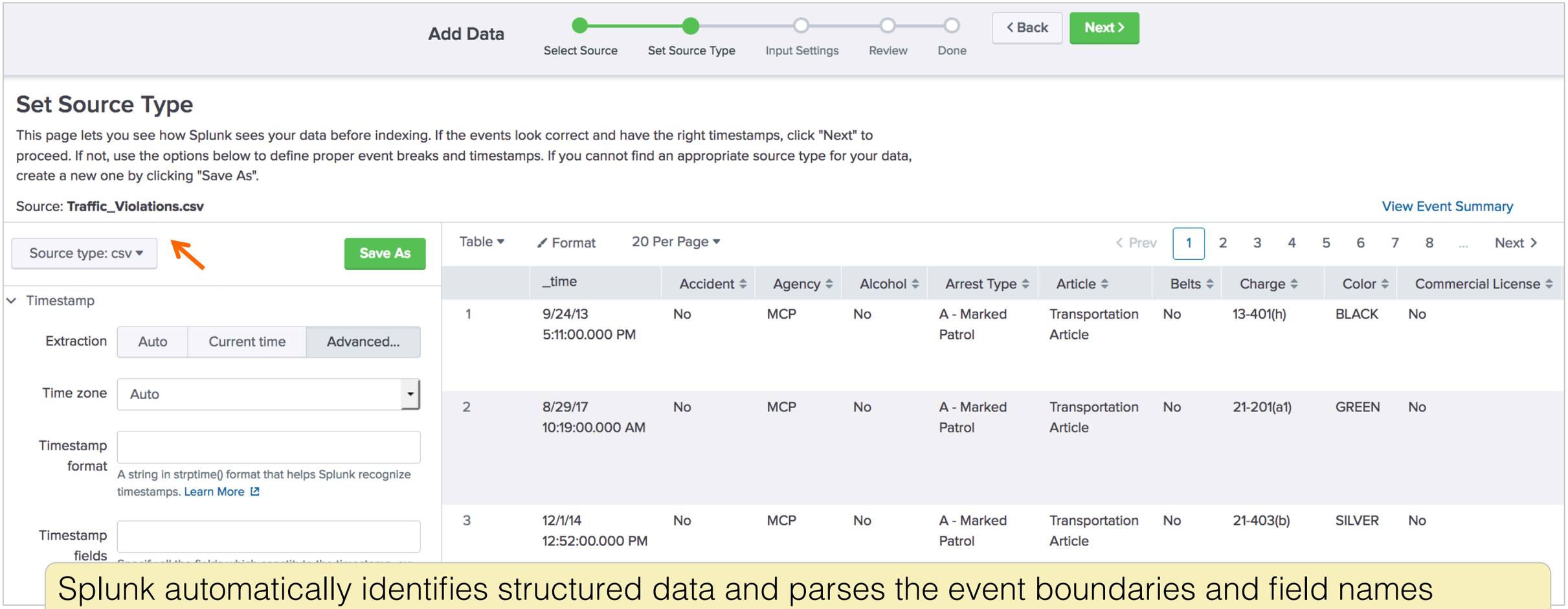


Enter event pattern prefix to parse some events correctly

Previewing Unstructured Data (cont.)



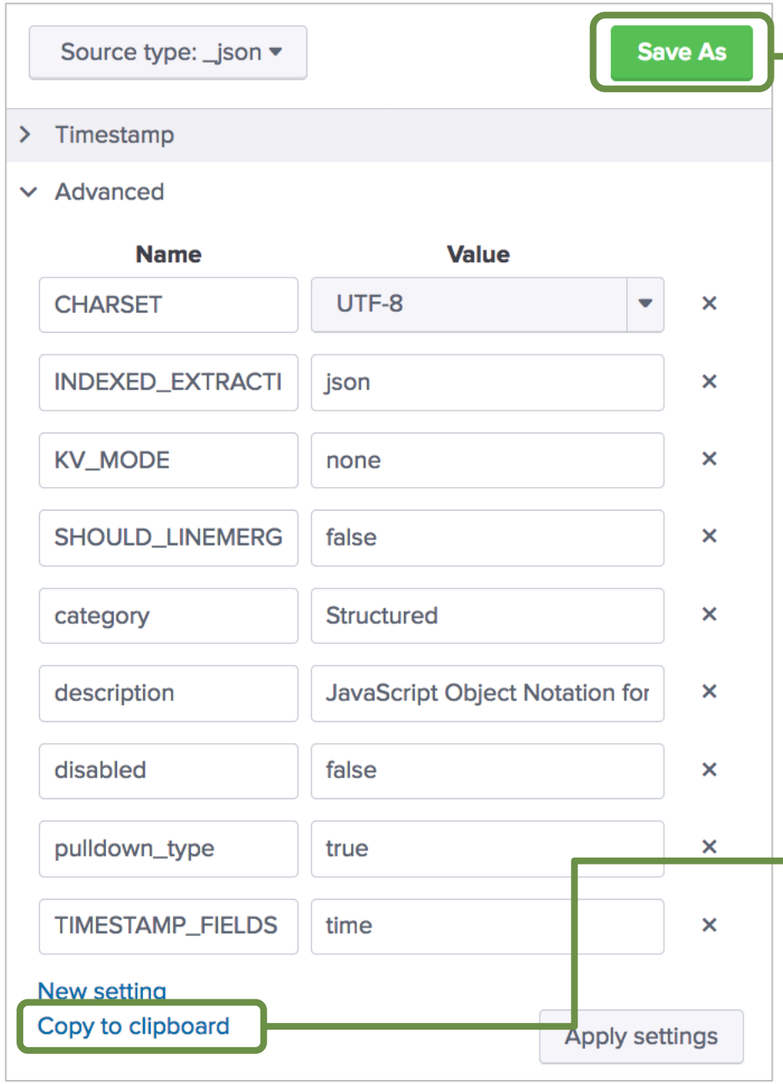
Previewing Structured Data

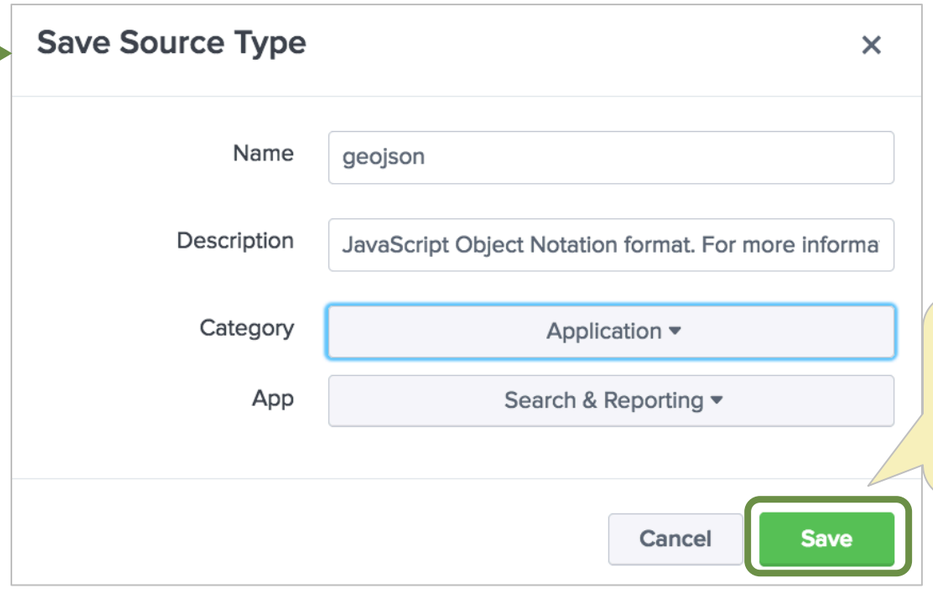


Splunk automatically identifies structured data and parses the event boundaries and field names

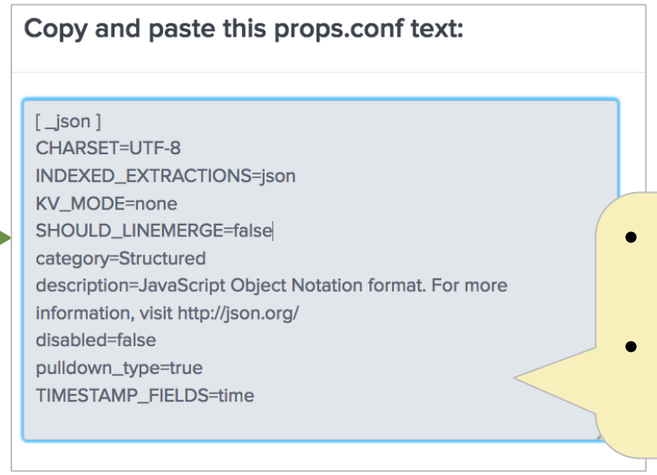
- Produces an **indexed extraction** stanza
- If you see a timestamp warning, indicate where to find a timestamp by specifying a field name

Saving New Source Type





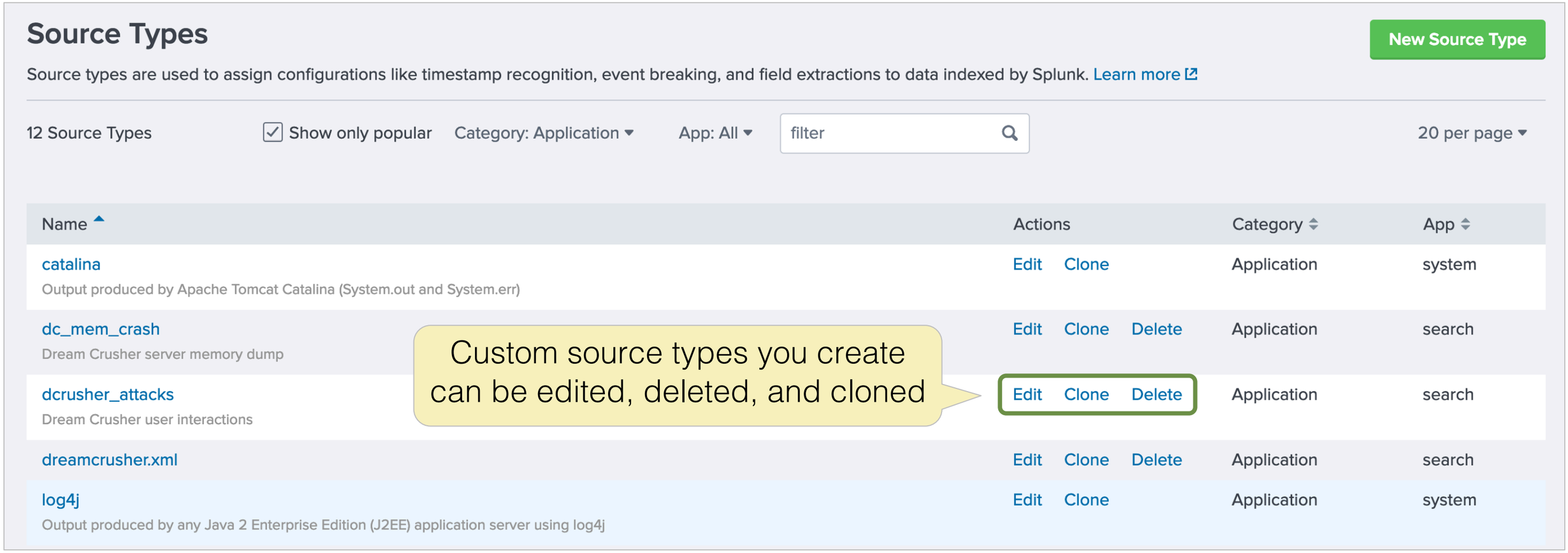
When saved, the source type becomes a custom source type that can be re-used



- Copy and deploy sourcetype settings manually to your forwarders
- Alternately get settings from **`props.conf`** stanza for the new source type

Source Type Manager

Click **Settings > Source types** to view and access the configured source types independent of the **Add Data** wizard



Module 9 Knowledge Check

- True or False. Time extraction can be done using **`props.conf`** on the UF and the HF.
- True or False. Event boundaries can be defined using **`props.conf`** at the UF.
- True or False. When extracting a timestamp, if the parser finds the indexer's OS time, it will use that as the first preference.

Module 9 Knowledge Check – Answers

- True or False. Time extraction can be done using **`props.conf`** on the UF and the HF.

  False. You will learn how to specify Time Extraction if the file contains a header line. But if it does not contain a header line, then time has to be extracted on the HF/ Indexer.

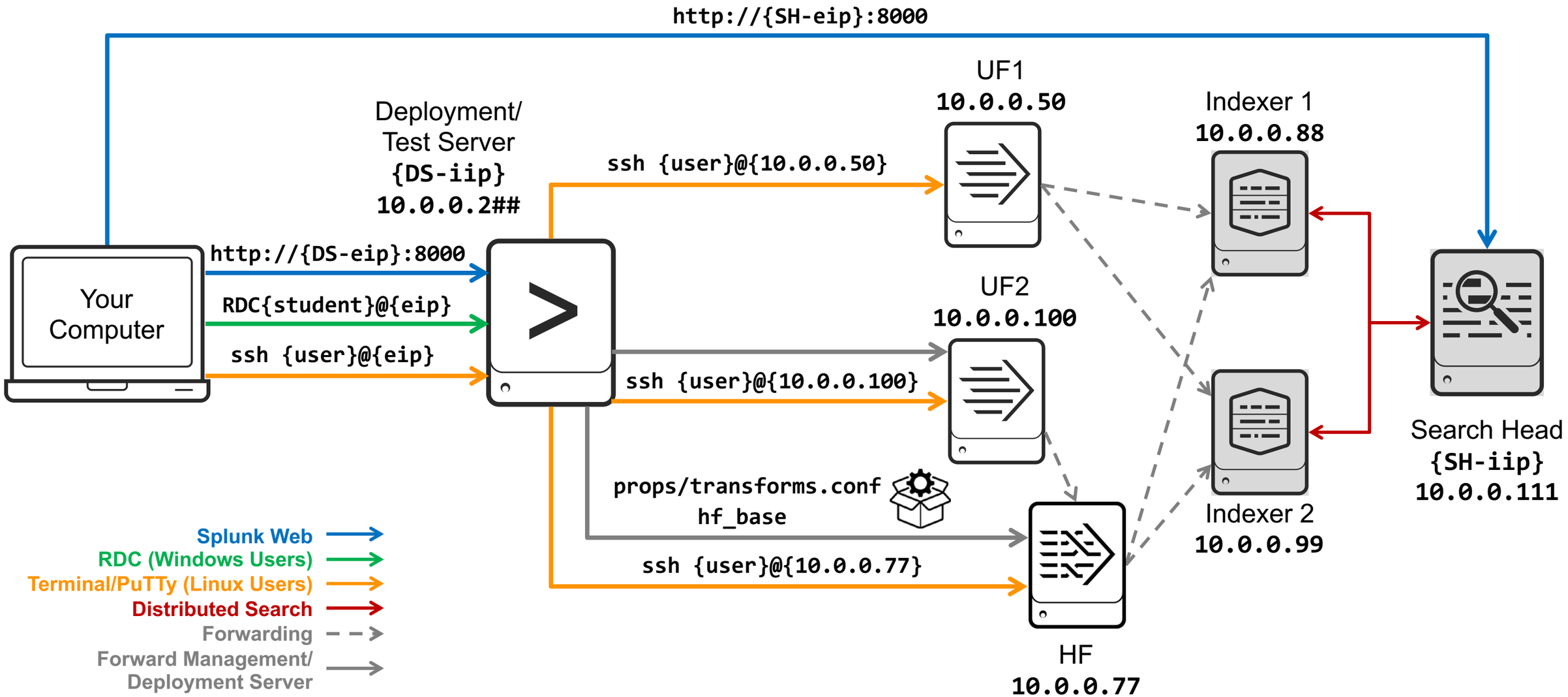
- True or False. Event boundaries can be defined using **`props.conf`** at the UF.

  True. You may want to define event boundaries for certain event types at the UF level. Remember the more you do at the UF level, the more resources you will need.

- True or False. When extracting a timestamp, if the parser finds the indexer's OS time, it will use that as the first preference.

  False. When all else fails, the Indexer's OS time is used as the *last* preference.

Module 9 Lab Exercise – Environment Diagram



Module 9 Lab Exercise – Create a New Source Type

Time: 20 – 25 minutes

Tasks:

- Use preview to evaluate two custom file types:
  - A new log sample that contains multiple timestamps
  - A new log sample that contains multi-line events in XML format
- Apply a custom line breaking rule and custom timestamp rules and save as a new sourcetype

Module 10: Manipulating Raw Data

Module Objectives

- Explain how data transformations are defined and invoked
- Use transformations with **`props.conf`** and **`transforms.conf`** to:
  - Mask or delete raw data as it is being indexed
  - Override sourcetype or host based upon event values
  - Route events to specific indexes based on event content
  - Prevent unwanted events from being indexed
- Use **`SEDCMD`** to modify raw data

Modifying the Raw Data

Sometimes necessary prior to indexing

- In cases of privacy concerns
  - **Healthcare**: Patient information
  - **Finance**: Credit card or account numbers
  - **Globalization**: Data transported across international borders
- According to business use cases
  - **Audit and security**: Routing all events to the **`web`** index, except credit card transactions which are sent to the **`credits`** index

Should be performed with care

- Unlike all other modifications discussed, these changes modify the raw data (**`_raw`**) before it is indexed
- Indexed data will not be identical to the original data source

Splunk Transformation Methods

- When possible, define meta field values during the input phase
  - Most efficient to use **inputs.conf**
- Splunk provides two methods of raw data transformations:

SEDCMD	TRANSFORMS
• Uses only **props.conf** • Only used to mask or truncate raw data	• Uses **props.conf** and **transforms.conf** • More flexible • Transforms matching events based on source, source type, or host

Using SEDCMD

- Splunk leverages a UNIX "sed-like" syntax for simplified data modifications
  - Provides “search and replace” using regular expressions and substitutions
  - Supported on both Linux and Windows versions of Splunk
- Example: Hide first 5 digits of account numbers in **vendor_sales.log** source

```
[22/Oct/2014:00:46:27] VendorID=9112 Code=B AcctID=4902636948
[22/Oct/2014:00:48:40] VendorID=1004 Code=J AcctID=4236256056
[22/Oct/2014:00:50:02] VendorID=5034 Code=H AcctID=8462999288
```

vendor_sales.log

Replace with **AcctID=xxxxx99288**

```
[source::.../vendor_sales.log]
SEDCMD-1acct = s/AcctID=\d{5}(\d{5})/AcctID=xxxxx\1/g
```

props.conf

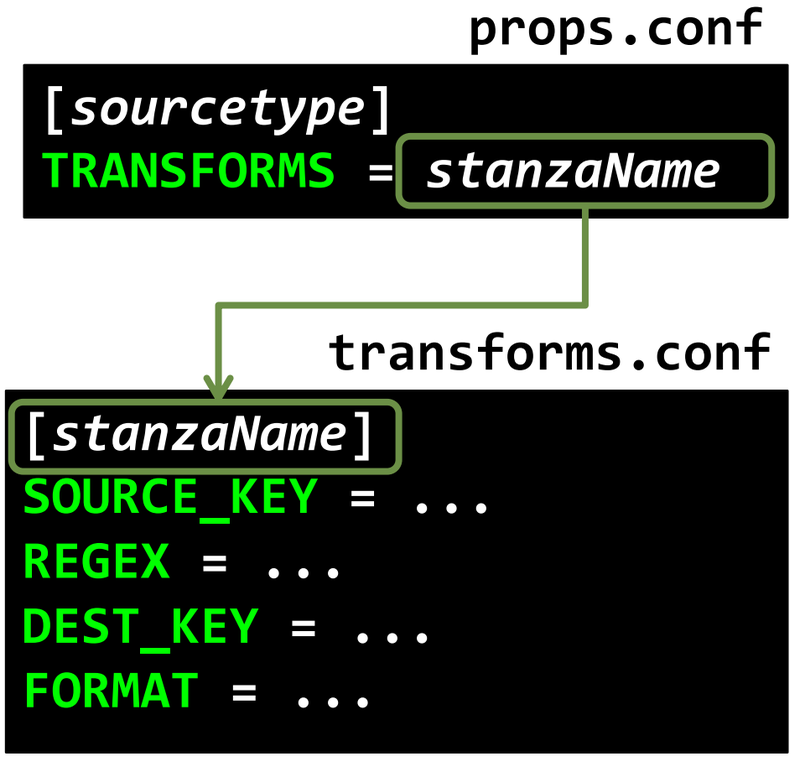
Indicates the capture group

- For more examples, see:

http://docs.splunk.com/Documentation/Splunk/latest/Data/Anonymizedata

Using TRANSFORMS

- Per event transformation is based on REGEX pattern matches
- Define the transformation in **transforms.conf**
- Invoke the transformation from **props.conf**
- Transformation is based on the following attributes:
  - **SOURCE_KEY** indicates which data stream to use as the source for pattern matching (default: **_raw**)
  - **REGEX** identifies the events from the **SOURCE_KEY** that will be processed (required)
    - Optionally specifies regex capture groups
  - **DEST_KEY** indicates where to write the processed data (required)
  - **FORMAT** controls how **REGEX** writes the **DEST_KEY** (required)



Masking Sensitive Data

```
[22/Apr/2014:00:46:27] VendorID=9112 CC_Num: 4217656647324534 Code=B
[22/Apr/2014:00:48:40] Sent to checkout TransactionID=100763
[22/Apr/2014:00:50:02] VendorID=5034 CC_Num: 6218651647508091 Code=H
```

props.conf

```
[source::...\\store\\purchases.log]
TRANSFORMS-1ccnum = cc_num_anon
```

- For the **purchases.log** source, send to the **cc_num_anon** transformation processor.
- The label **-1ccnum** identifies this transform namespace and is used to determine sequence.

transforms.conf

```
[cc_num_anon]
REGEX = (.*CC_Num:\s)\d{12}(\d{4}.*)
DEST_KEY = _raw
FORMAT = $1xxxxxxxxxxxx$2
```

- When **SOURCE_KEY** is omitted, **_raw** is used.
- This **REGEX** pattern finds two capture groups and rewrites the raw data feed with a new format.

```
[22/Apr/2014:00:46:27] VendorID=9112 CC_Num: xxxxxxxxxxxx4534 Code=B
[22/Apr/2014:00:48:40] Sent to checkout TransactionID=100763
[22/Apr/2014:00:50:02] VendorID=5034 CC_Num: xxxxxxxxxxxx8091 Code=H
```

Setting Per-Event Source Type

Should be your last option because it is more efficient to set the sourcetype during the inputs phase

```
[29/Apr/2017:07:08:32] VendorID=4119 Code=E AcctID=1808937180466558 Custom
[29/Apr/2017:07:09:42] VendorID=5012 Code=N AcctID=7905045242265135
[29/Apr/2017:07:11:10] VendorID=7015 Code=G AcctID=3283196485834211 Custom
```

props.conf

```
[source::udp:514]
TRANSFORMS = custom_sourcetype
```

transforms.conf

```
[custom_sourcetype]
SOURCE_KEY = _raw
REGEX = Custom$
DEST_KEY = MetaData:Sourcetype
FORMAT = sourcetype::custom_log
```

- Check events in network input source
- If an event contains “Custom” at the end, assign the new sourcetype value **custom_log**
- When **MetaData:** key is used, its **FORMAT** value must be prefixed by:
  - **host::**
  - **source::**
  - **sourcetype::**

Setting Per-Event Host Name

```
[22/Apr/2014:00:46:27] sales accepted server:A01R2 SID=107570
[22/Apr/2014:00:48:40] sales rejected server:B13R1 SID=102498
[22/Apr/2014:00:50:02] sales accepted server:A05R1 SID=173560
```

props.conf

```
[sales_entries]
TRANSFORMS-register = sales_host
```

transforms.conf

```
[sales_host]
SOURCE_KEY = _raw
REGEX = server:(\w+)
DEST_KEY = MetaData:Host
FORMAT = host::$1
```

- Check each events in the **_raw** source
- If an event contains "**server:**", capture the word and rewrite the value of the **MetaData:Host** key with the captured group
- When **MetaData:** key is used, its **FORMAT** value must be prefixed by:
  - **host::**
  - **source::**
  - **sourcetype::**

Per-Event Index Routing

Again, if at all possible, specify the index for your inputs during the input phase (**inputs.conf**)

props.conf

```
[mysrctype]
TRANSFORMS-itops = route_errs_warns
```

transforms.conf

```
[route_errs_warns]
REGEX = (Error|Warning)
DEST_KEY = _MetaData:Index
FORMAT = itops
```

If **Error** or **Warning** is found in the incoming **_raw**, change its **index** field value to **itops**

Filtering Unwanted Events

- You can route specific unwanted events to the **null queue**
  - Events discarded at this point do NOT count against your daily license quota

props.conf

```
[WinEventLog:System]
TRANSFORMS = null_queue_filter
```

transforms.conf

```
[null_queue_filter]
REGEX = (?i)^EventCode=(592|593)
DEST_KEY = queue
FORMAT = nullQueue
```

- The **(?i)** in the **REGEX** means "ignore case." Events with an **eventcode** of **592** or **593** should not be indexed
- Route to **queue** and use **nullQueue** format to discard events

Routing Events to Groups using HF

You can route specific events to different groups using the HF (another use case for HF)

props.conf

```
[default]
TRANSFORMS-routing=errorRouting

[syslog]
TRANSFORMS-routing=syslogRouting
```

transforms.conf

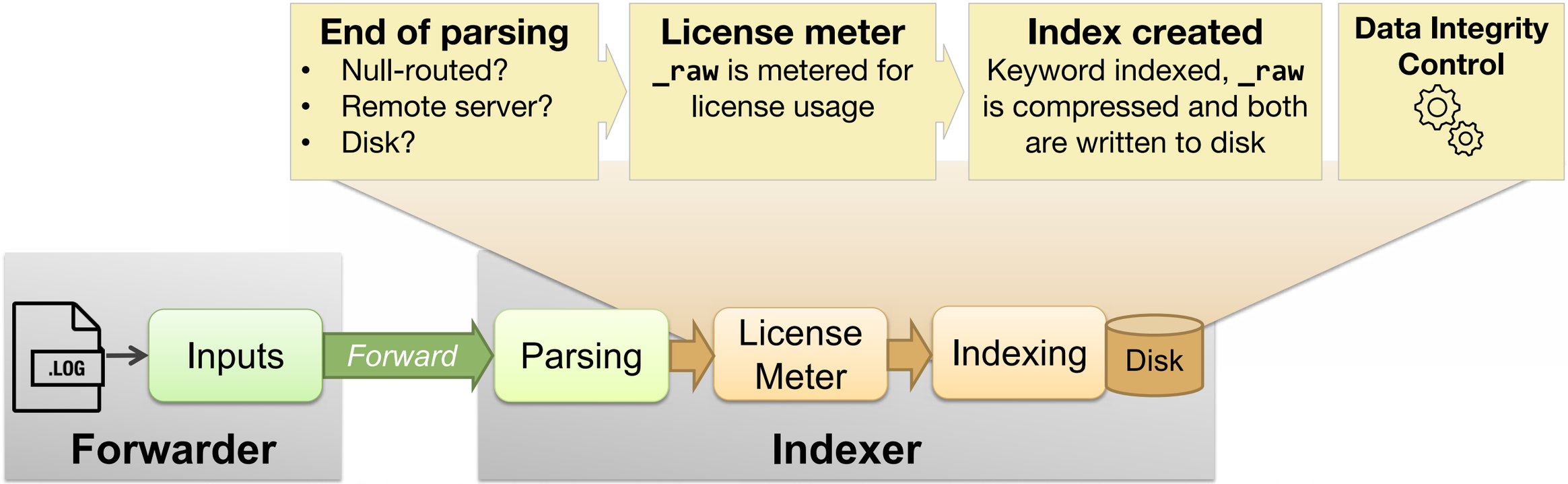
```
[errorRouting]
REGEX = error
DEST_KEY=_TCP_ROUTING
FORMAT = errorGroup

[syslogRouting]
REGEX = .
DEST_KEY=_TCP_ROUTING
FORMAT=syslogGroup
```

outputs.conf

```
[tcpout]
defaultGroup=everythingElseGroup

[tcpout:errorGroup]
server=10.1.1.200:9999

[tcpout:syslogGroup]
server=10.1.1.197:9996,10.1.1.198:9997

[tcpout:everythingElseGroup]
server=10.1.1.250:9998
```


Indexing Phase Details

After the parsing phase, Splunk passes the fully processed events to the index processor



Persisted to Disk

- Indexed data is written to disk
  - Includes all modifications and extractions
  - Includes raw data (**`_raw`**) and metadata (source, sourcetype, host, timestamp, punct, etc.)
- Changes to **`props.conf`** or **`transforms.conf`**
  - Only applies to new data
  - Requires restarting the indexer, or re-loading by visiting:

    http://*servername:splunkwebport*/debug/refresh
- Re-indexing is required to index old data with new settings

Module 10 Knowledge Check

- True or False. **sedcmd** can be used to eliminate unwanted events.
- True or False. When using **transforms.conf**, the **SOURCE_KEY** is set to **_raw** by default.
- In the **props.conf** file example below, what is **itops**?

```
[mysrctype]
TRANSFORMS-itops = route_errs_warns
```


Module 10 Knowledge Check – Answers

- True or False. **sedcmd** can be used to eliminate unwanted events.

  False. You have to use **transforms.conf**. **sedcmd** can only be used to mask or truncate data.

- True or False. When using **transforms.conf**, the **SOURCE_KEY** is set to **_raw** by default.

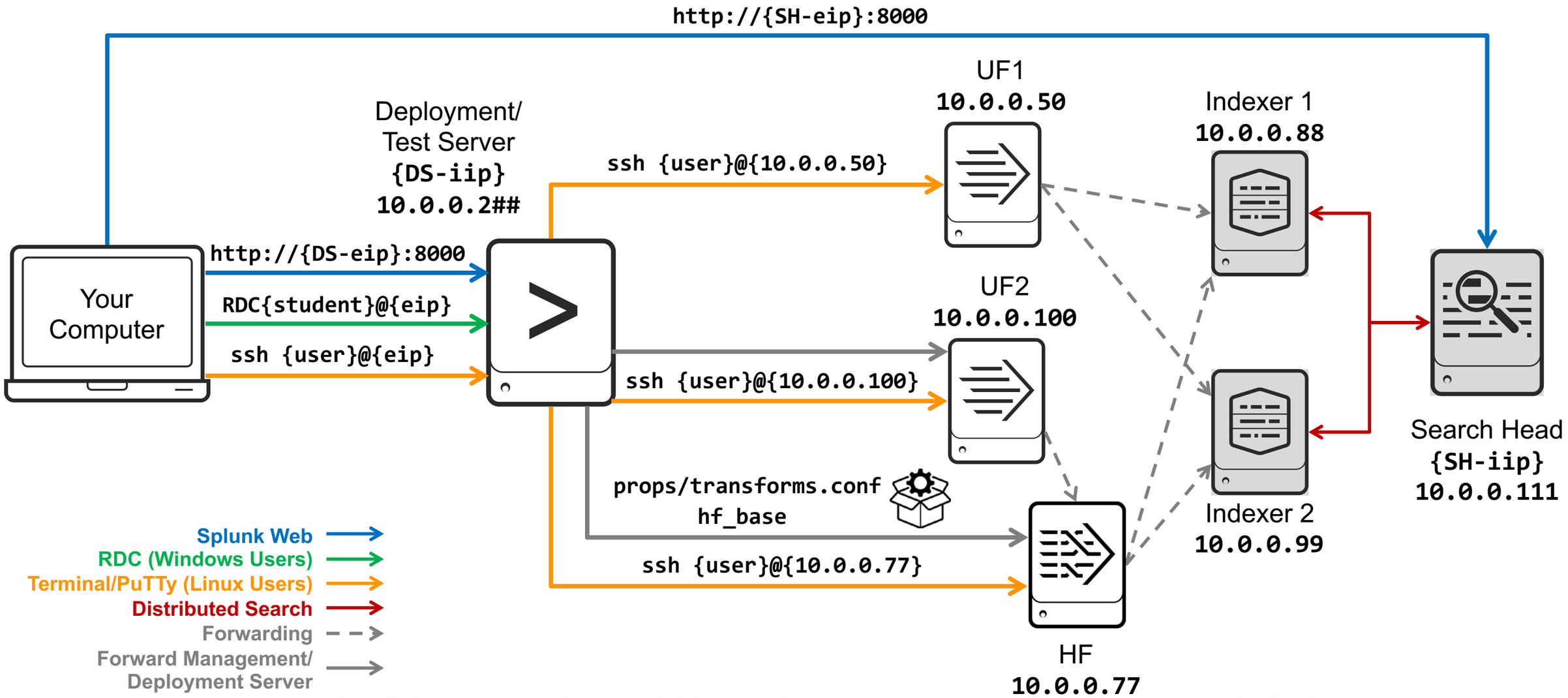
  True. If you do not specify the **SOURCE_KEY** in **transforms.conf**, it defaults to **_raw**.

- In the **props.conf** file example below, what is **itops**?

```
[mysrctype]
TRANSFORMS-itops = route_errs_warns
```

  Itops is the namespace and is used to determine the sequence.

Module 10 Lab Exercise – Environment Diagram



Module 10 Lab Exercise – Manipulating Data

Time: 10 - 15 minutes (20 - 25 minutes with optional lab)

Tasks:

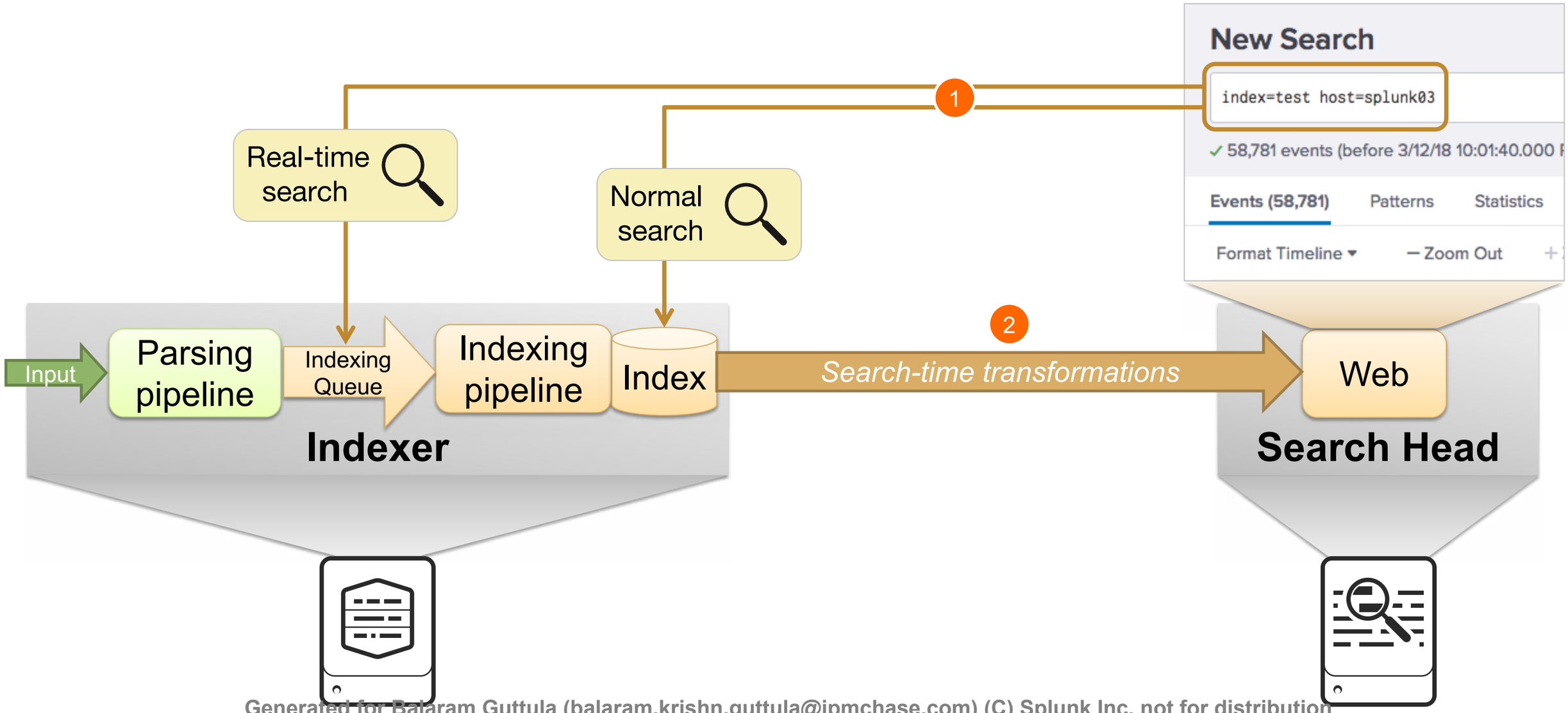
- Use **`transforms.conf`** to:
  - Mask sensitive data
  - Redirect events to specific indexes
  - Drop unwanted events
- (Optional lab) Use **`props.conf`** to:
  - Drop in sequence the filtering and redirecting events

Module 11: Supporting Knowledge Objects

Module Objectives

- Define default and custom search time field extractions
- Discuss the pros and cons of indexed time field extractions
- Configure indexed field extractions
- Describe default search time extractions
- Manage orphaned knowledge objects

Search Phase: The Big Picture



Index-Time Field Extraction

- During index-time, event data is stored in the index on disk
  - Default fields are extracted and added automatically
  - Custom fields are added based on customizations (by the data admin)
- Fields are generally extracted at search-time
- Certain use cases perform field extraction at index-time
  - On the forwarder for structured inputs
  - On the indexer for fields that may be negatively impacting search performance
- Add index-time custom fields only if necessary
  - Can negatively impact indexing performance and search times
  - Increases the size of the searchable index

Pros/Cons of Index Time Field Extractions

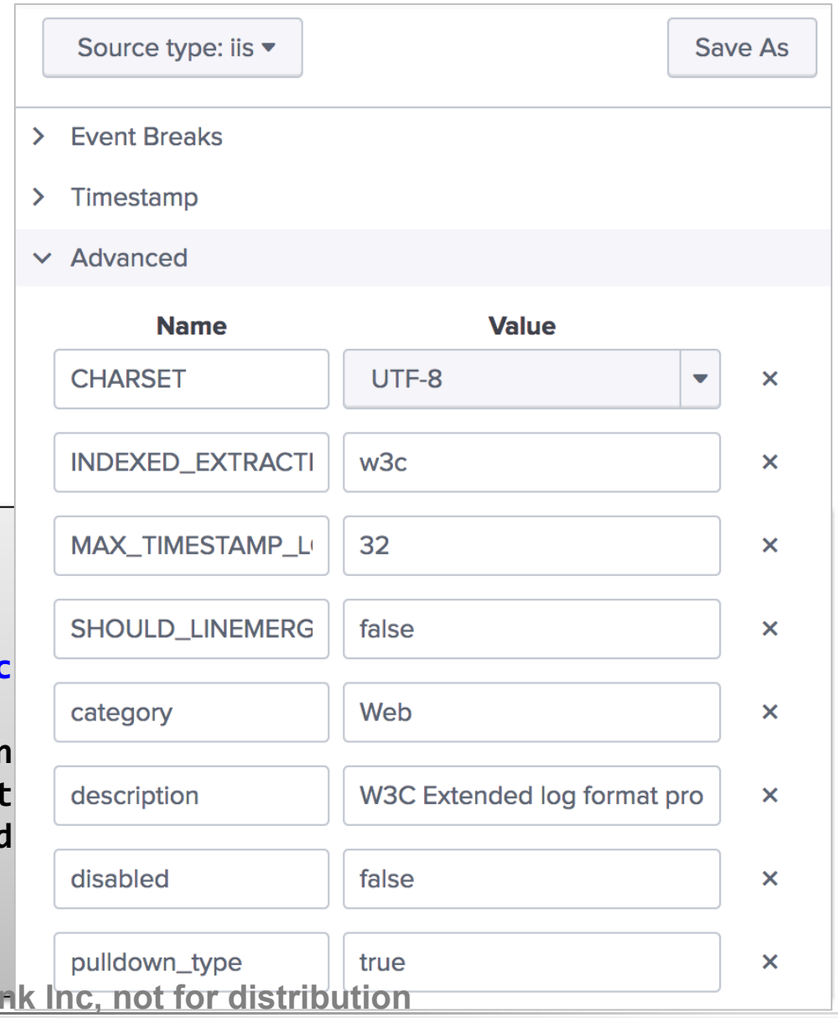
PROs	CONs
• Provision the extraction during the **input** or **parsing** phase • Can configure on the universal forwarder • Auto-formatting • Can drop useless headers and comments	• Increased storage size (2-5x the original size consumed on the indexer) • Static field names: additional step required for late-binding use cases • Possible performance implications • Less flexible: changes to fields require a re-index of the dataset, or only apply to new data

- Recommendations:
  - For frequently re-configured delimited sources, use indexed extractions (example: **`IIS`**)
  - For static CSV, use **`REPORT`** and **`DELIMS`**, or other search-time extractions
  - Use a dedicated index

Structured Data Field Extraction Example

- Indexed extractions are **input phase `props.conf`** settings
  - In this scenario, the settings belong on forwarder
  - Check **`props.conf.spec`** for more options

```
[my_structured_data]
INDEXED_EXTRACTIONS = w3c
HEADER_FIELD_LINE_NUMBER = 4
TIMESTAMP_FIELDS = date, time
```




```
#Software: Microsoft Internet Information Services 7.5
#Version: 1.0
#Date: 2015-06-08 00:00:00
#Fields: date time cs-method cs-uri-stem cs-uri-query c-ip cookie referer cs-host sc
2015-01-08 00:00:00 POST AutoComplete.asmx/GetCompletionList - 10.175.16.79
cApproved=1;+fParticipant=0000000695607440|urn:System-Services:GatewayTokenService_n
format:persistent|http://www.acme.com/2015/06/attributes/credentialidentifier; &nest
fc2df5;+style=normal https://search.acme.com/Account/Account.aspx?redirect=https://d
200 1113 0
...
```

Configuring Indexed Field Extractions

Define additional attributes in **`props.conf`**, **`transforms.conf`**, and fields in **`fields.conf`**

File	Splunk instance	Example
`props.conf`	Indexer, Heavy Forwarder	`[testlog]` `TRANSFORMS-netscreen = netscreen-error`

`transforms.conf`	Indexer, Heavy Forwarder	`[netscreen-error]`
		`REGEX = device_id=\[\w+\](?<error_code>[^:]+)`
		`FORMAT = error_code::"$1"`
		`WRITE_META = true`

`fields.conf`	Search Head	`[error_code]` `INDEXED=true`

Indexed Field Extractions – Caveat

- Splunk software does not parse structured data that has been forwarded to an indexer
  - If you have configured **`props.conf`** on the targeted forwarder with **`INDEXED_EXTRACTIONS`** and its associated attributes, the forwarded data skips the following queues on the indexer:
    - Parsing
    - Aggregation
    - Typing

http://docs.splunk.com/Documentation/Splunk/latest/Forwarding/Routeandfilterdatad#Caveats_for_routing_and_filtering_structured_data

Default Search Time Field Extractions

- Provided by Splunk for common source types
- Can be discovered by Splunk from your search results
  - Automatically detects key/value pairs (e.g. **a=1**)
- Can be added with add-ons and apps
  - ***nix app**: has many search time fields for standard UNIX logs
    - For example, **secure.log**, **messages.log**, etc.
  - **Windows app**: has many defaults for Windows data
  - **For other data**: look for an app on http://splunkbase.splunk.com specifically designed for that type of data

Custom Search Time Field Extractions

SPL

- Use **`rex`** (or similar) commands in the search language
- Requires knowledge of regular expressions (REGEX)
- All roles can use this command

Field Extractor

- Found in Splunk Web
- Handles REGEX-based and delimiter-based extractions
- Knowledge of regular expressions helpful, but not required

Configuration files

- Provides additional advanced extraction options
- Knowledge of REGEX required
- Available only to admins

Field Extractions and **`props.conf`**

- Field extraction happens during index-time (indexed fields) and/or search-time (extracted fields)
- Search-time extractions can be inline or a field transform
- Use extraction directives
  - **`EXTRACT`** (inline extraction)
    - Defined in **`props.conf`** as single field extraction
  - **`REPORT`** (field transform)
    - Defined in **`transforms.conf`**
    - Invoked from **`props.conf`**

REPORT Extractions in props.conf

- **REPORT** references a transform defined separately in **transforms.conf**
- In **transforms.conf**, you can
  - Define field extractions using delimiters
  - Apply other advanced extraction techniques
- For full details on **REPORT**, see:

docs.splunk.com/Documentation/Splunk/latest/Knowledge/Createandmaintainsearch-timefieldextractionsthroughconfigurationfiles

Using **EXTRACT** and **REPORT** in **props.conf**

- Applies to this sourcetype
- The REGEX pattern defines extracted field

Arbitrary namespace you assign to this extraction. Useful for ordering multiple transactions

props.conf

```
[tradelog]
EXTRACT-1type = type:\s(?<acct_type>\S+)

[sysmonitor]
REPORT-sysmon = sysmon-headers
KV_MODE = none
```

Extracted field name

Process this stanza in **transforms.conf**

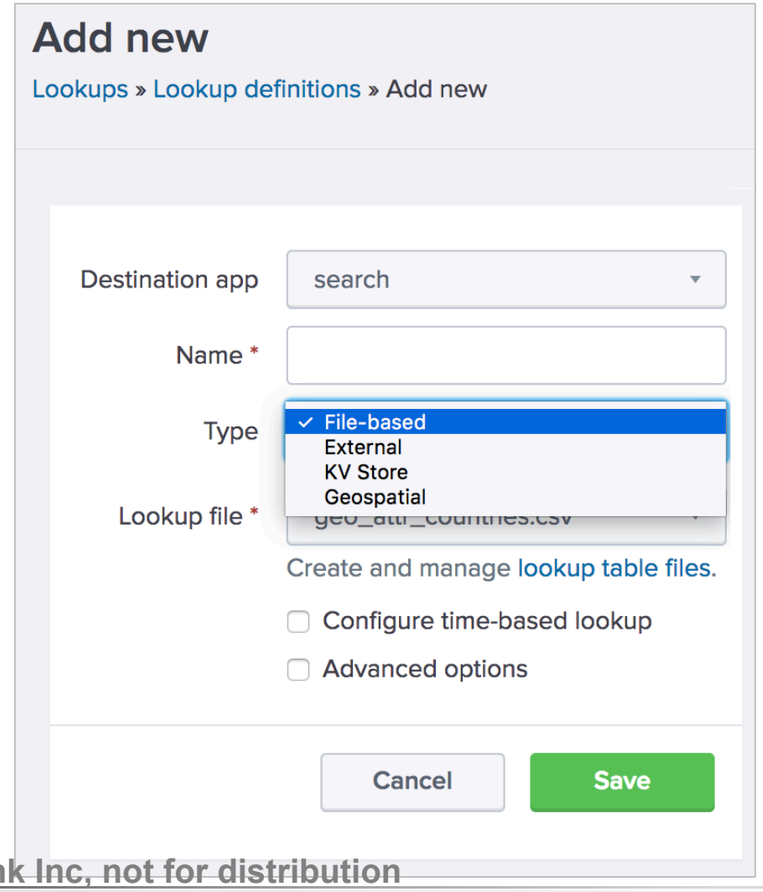
transforms.conf

```
[sysmon-headers]
DELIMS = ","
FIELDS = Time,EventCode,EventType,Type,ComputerName,LogName,RecordNumber
```


Lookups

- A Splunk data enrichment knowledge object
  - Uses stanzas defined in **`transforms.conf`** and **`props.conf`**
  - Used *only* during search time
- Four types:

Lookup type	Description
File-based	Uses a CSV file stored in the lookups directory
KV Store	Requires **`collections.conf`** that defines fields
External	Uses a python script or an executable in the **`bin`** directory
Geospatial	Uses a **`kmz`** saved in the lookups directory to support the choropleth visualization



Other Search Time Knowledge Objects

- KOs are stored in configuration files:
  - **`macros.conf`**, **`tags.conf`**, **`eventtypes.conf`**, **`savedsearches.conf`**, etc.
  - See docs and **`*.spec`** files in **`SPLUNK_HOME/etc/system/README`**
- Create or modify KOs using:
  - Splunk Web (automatically updates **`.conf`** files)
  - Editing **`.conf`** files manually (requires admin rights)
    - Use **`btool`** to verify changes
  - Splunk Web: **Advanced edit** (supports some system settings)

Search name	RSS feed	Scheduled time	Display view	Owner	App	Alerts	Sharing	Status	Actions
quake_L24h		None	None	emaxwell	search	0	Private | Permissions	Enabled | Disable	Run | Advanced edit | Clone | Move | Delete
quake_L24H		None	None	admin	search	0	Private | Permissions	Enabled | Disable	Run | Advanced edit | Clone | Move | Delete
Top five sourcetypes		None	None	No owner	search	0	App | Permissions	Enabled | Disable	Run | Advanced edit | Clone

Orphaned Knowledge Objects (KOs)

What are orphaned knowledge objects?

- KOs without a valid owner
- Occurs when a Splunk account is deactivated and the KOs associated with that account remain in the system

Issues with orphaned knowledge objects

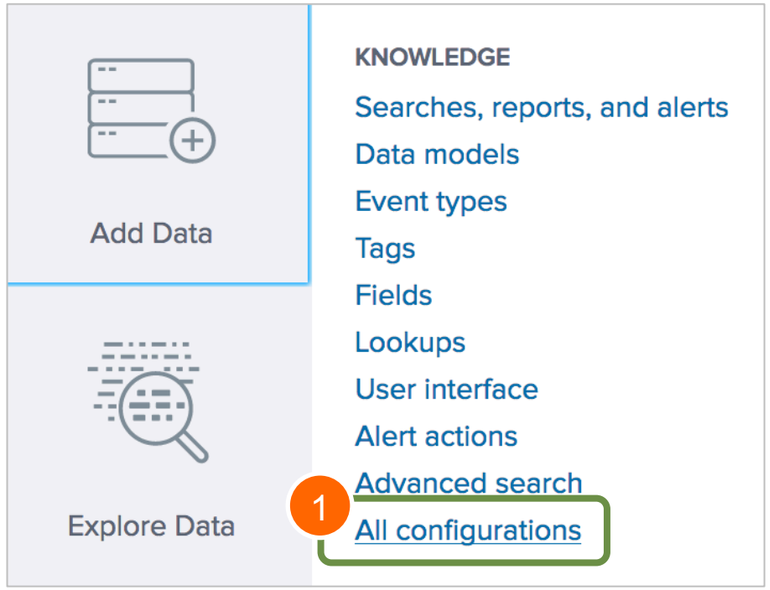
- Can cause performance problems and security concerns
- Searches that refer to an orphaned lookup may not work
- Search scheduler cannot run a report on behalf of a nonexistent owner

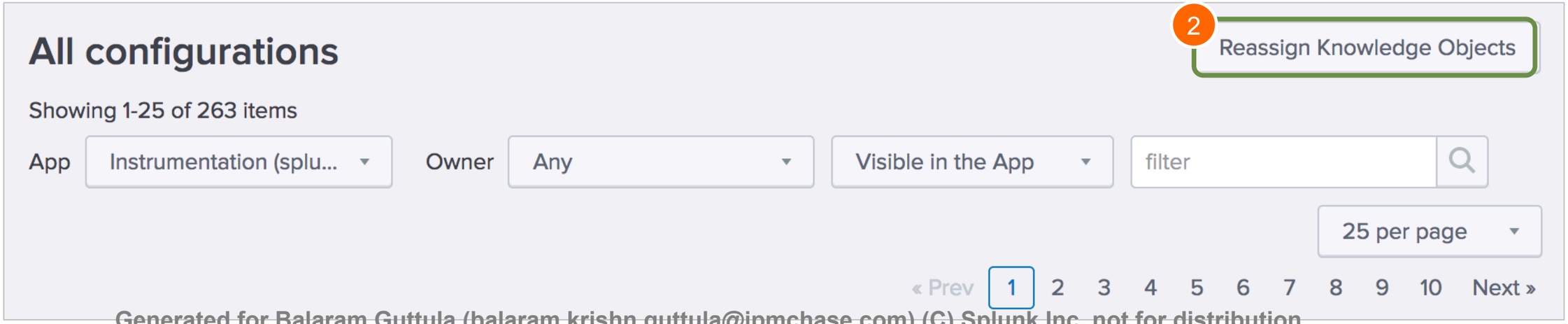
Locating Orphaned Knowledge Objects

- Splunk runs a default search on a daily schedule to detect orphaned scheduled reports
- Report on orphaned KO using any of these methods:
  - Click **Messages**, then click the message link to access the alerts dashboard
  - Run the search from **Search > Dashboards > Orphaned Scheduled Searches, Reports, Alerts**
  - Run the MC Health Check search to detect orphaned knowledge objects

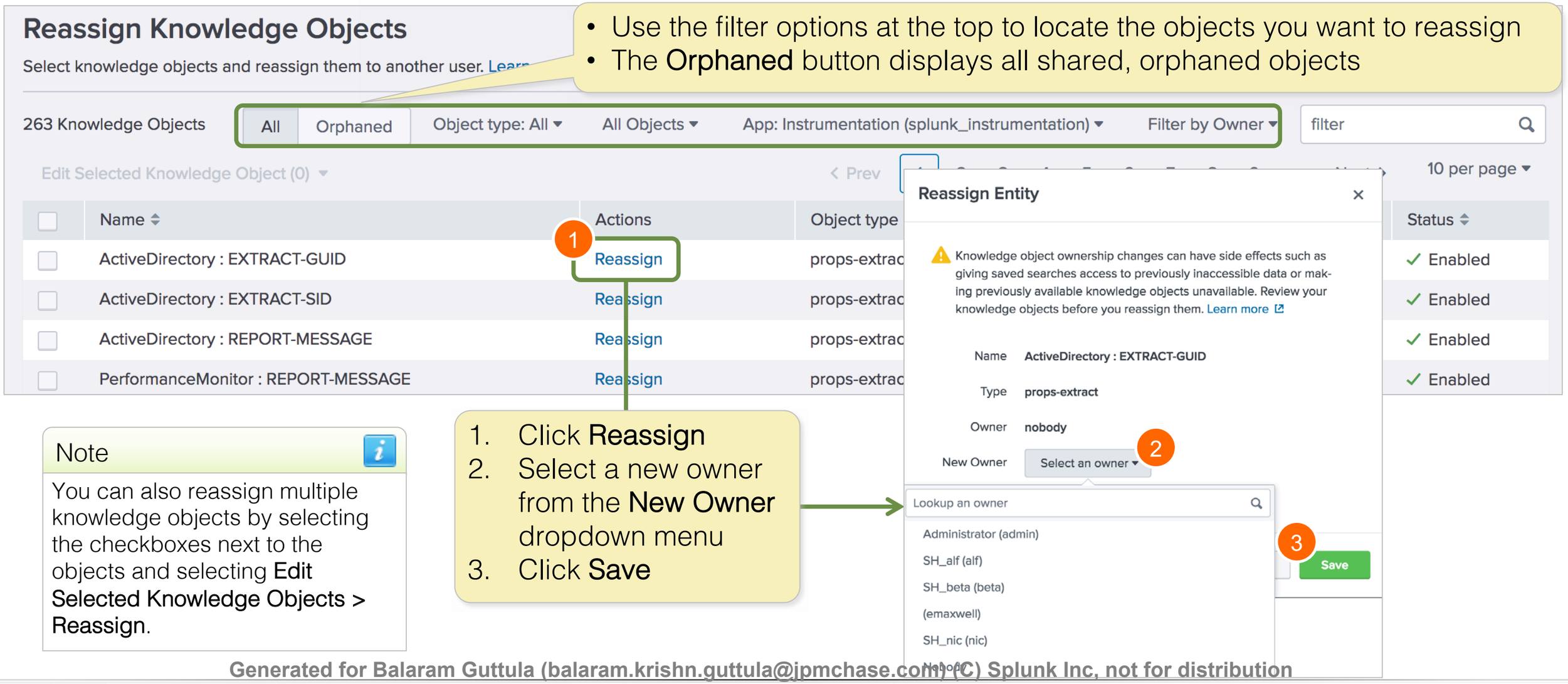
Reassigning Knowledge Objects

- Requires **`admin`** role capability
- Possible for both orphaned and owned KOs
- Performed in Splunk Web with:
  1. Select **Settings > All configurations**
  2. Click **Reassign Knowledge Objects**





Reassigning Knowledge Objects (cont.)



- Use the filter options at the top to locate the objects you want to reassign
- The **Orphaned** button displays all shared, orphaned objects

1. Click **Reassign**
2. Select a new owner from the **New Owner** dropdown menu
3. Click **Save**

Note

You can also reassign multiple knowledge objects by selecting the checkboxes next to the objects and selecting **Edit Selected Knowledge Objects > Reassign**.

Module 11 Knowledge Check

- True or False. **`props.conf`** and **`transforms.conf`** are used to store Field Extractions, Lookups, Saved Searches and macros.
- True or False. Any user belonging to any user role can reassign any KO.
- True or False. When you select the REGEX option in the Field Extractor in the GUI, it uses **`props.conf`** and **`transforms.conf`** in the background.

Module 11 Knowledge Check – Answers

- True or False. **`props.conf`** and **`transforms.conf`** are used to store Field Extractions, Lookups, Saved Searches and macros.

  False. They are used only for Field Extractions and Lookups.

- True or False. Any user belonging to any user role has the ability to reassign any KO.

  False. Only users belonging to the **`admin`** role can assign any KO.

- True or False. When you are using Splunk Web and select the REGEX option in the Field Extractor, it uses **`props.conf`** and **`transforms.conf`** in the background.

  False. It only uses **`props.conf`**. Delimiter based extractions entries in **`props.conf`** and **`transforms.conf`** are manually created.

Module 11 Lab Exercise – KO Administration

Time: 5 – 10 minutes

Tasks:

- Create a knowledge object (report)
- Search for orphaned knowledge objects
- Assign the report to the user, **`emaxwell`**

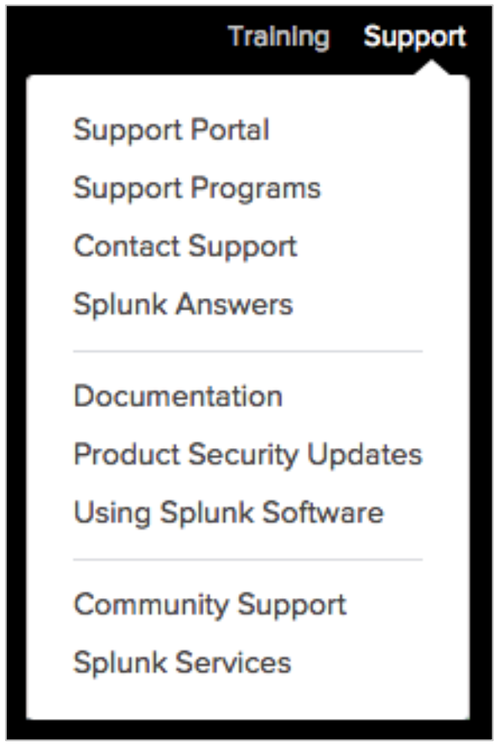
Course Wrap-up

Community

- Splunk Community Portal
  splunk.com/en_us/community.html
- Splunk Answers
  answers.splunk.com
- Splunkbase
  splunkbase.splunk.com/
- Splunk Blogs
  splunk.com/blog/
- Splunk Live!
  http://splunklive.splunk.com/
- Splunk .conf
  conf.splunk.com
- Splunk Wiki
  wiki.splunk.com
- Slack User Groups
  splk.it/slack
- Splunk Dev Google Group
  groups.google.com/forum/#!forum/splunkdev
- Splunk Docs on Twitter
  twitter.com/splunkdocs
- Splunk Dev on Twitter
  twitter.com/splunkdev
- IRC Channel
  #splunk on the EFNet IRC server

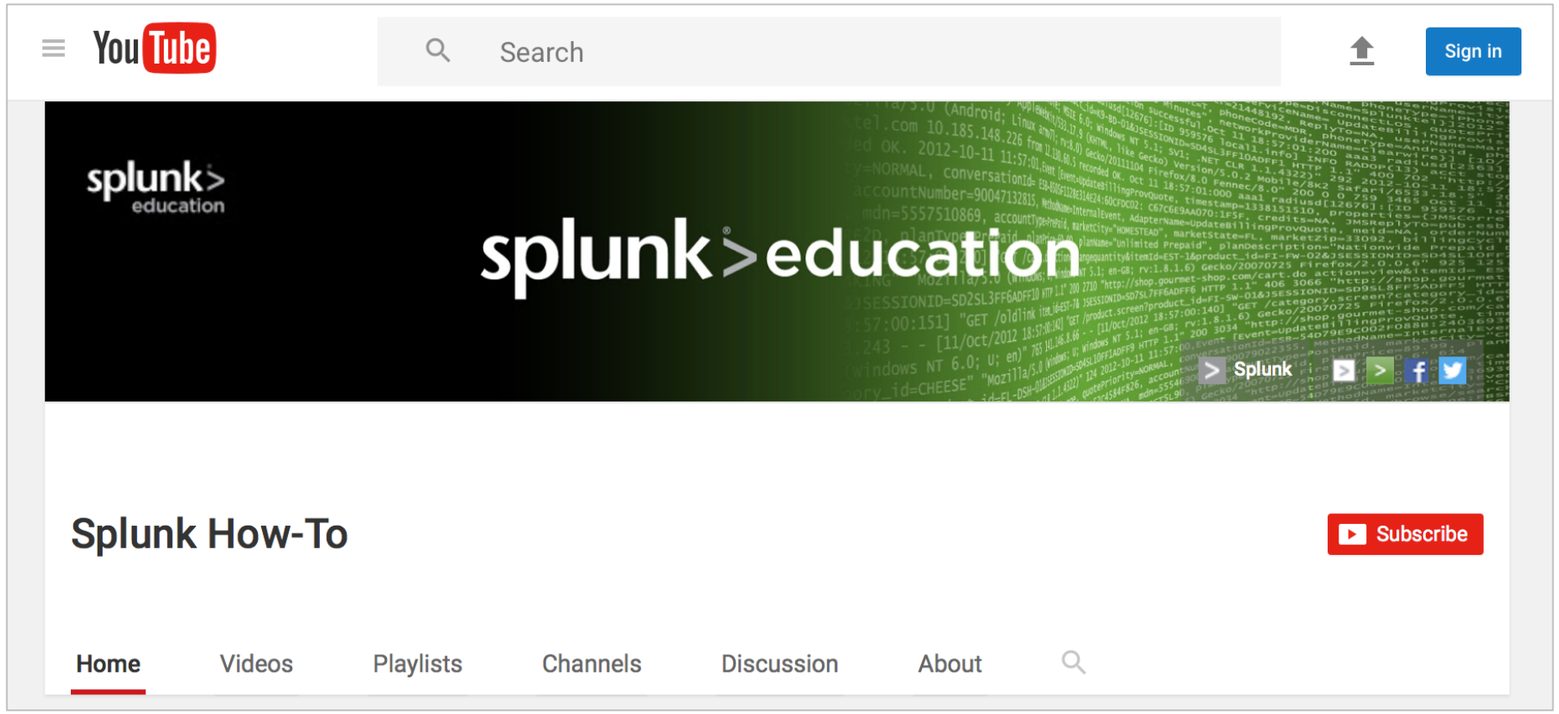
Support Programs

- Web
  - Documentation: docs.splunk.com and dev.splunk.com
  - Wiki: wiki.splunk.com
- Global Support
  - Support for critical issues, a dedicated resource to manage your account – 24 x 7 x 365
  - Web: splunk.com/index.php/submit_issue
  - Phone: (855) SPLUNK-S or (855) 775-8657
- Enterprise Support
  - Access customer support by phone and manage your cases online 24 x 7 (depending on support contract)



YouTube: The Splunk How-To Channel

- In addition to the roster of Splunk Education training courses, check out our How-To channel: http://www.youtube.com/c/SplunkHowTo
- Useful, short videos on a variety of Splunk topics



Splunk Certification

- Splunk Certification program

  https://www.splunk.com/en_us/training/faq-training.html

- Program information

  https://www.splunk.com/pdfs/training/Splunk-Certification-Candidate-Handbook.pdf

- Exam registration

  https://www.splunk.com/pdfs/training/Exam-Registration-Tutorial.pdf

- If you have further questions, send an email to:

  certification@splunk.com

splunk> .conf20

October 20-21, 2020

Join us for two days of innovation featuring dozens of educational sessions and numerous opportunities to do amazing things with data.

> “Splunk makes our imagination the only limit to unlocking and understanding our data.”
>
> - IT Specialist, US Public Sector

Learn more at

conf.splunk.com

Thank You