Splunk Enterprise 7.3 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 7.3 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: Introduction to Data Administration

Module 2: Getting Data In - Staging

Module 3: Forwarder Configuration

Module 4: Forwarder Management

Module 5: Monitor Inputs

Module 6: Network and Scripted Inputs

Module 7: Agentless Inputs

Module 8: Fine-tuning Inputs

Module 9: Parsing Phase and Data Preview

Module 10: Manipulating Raw Data

Module 11: Supporting Knowledge Objects

Module 12: Splunk Diag

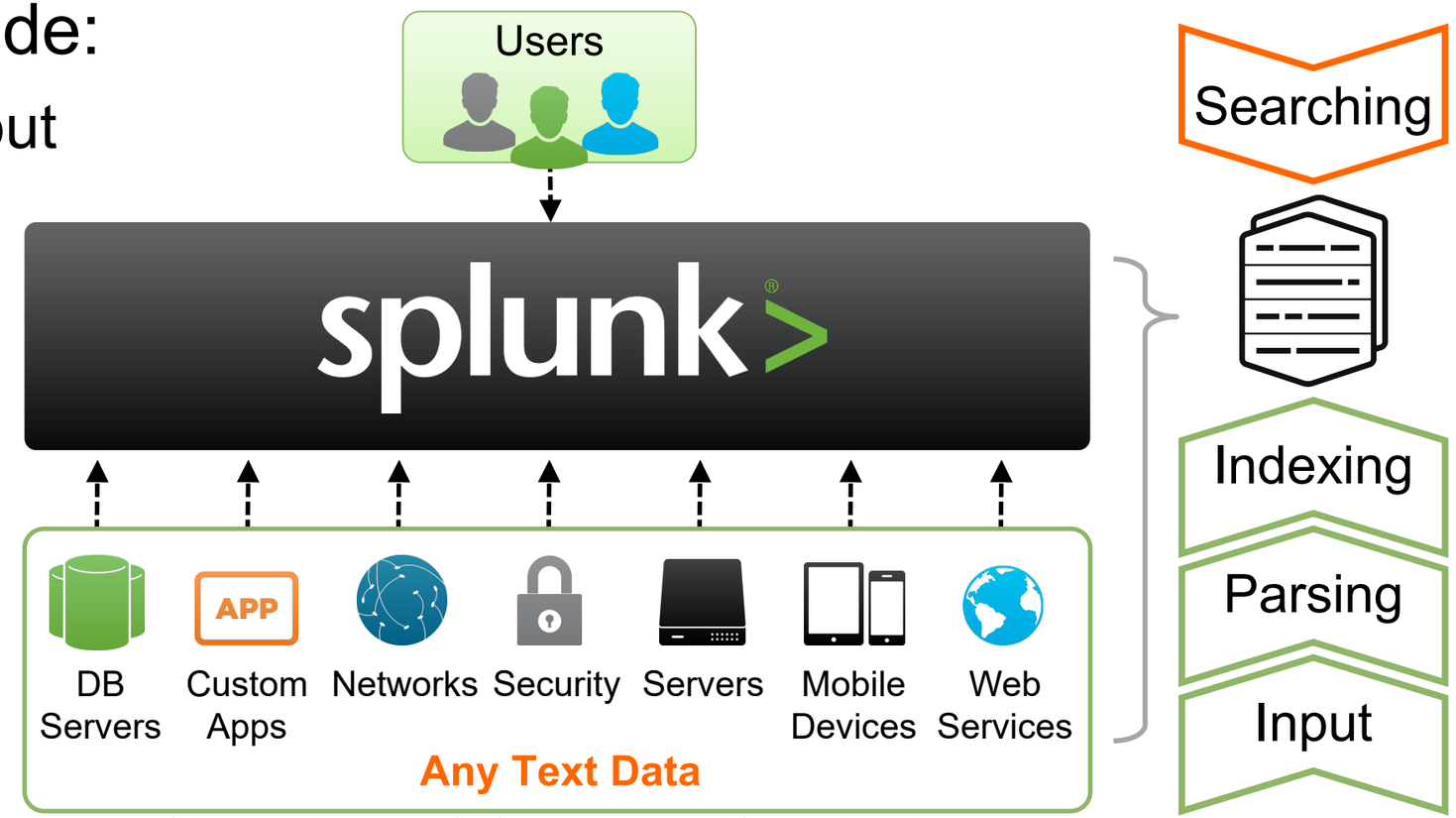
Module 1: Introduction to Data Administration

Module 1 Objectives

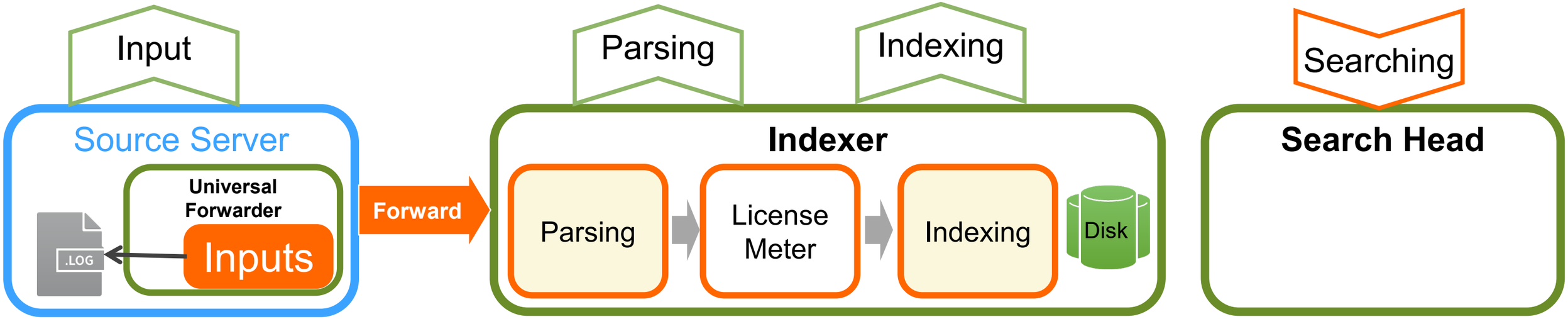
- Overview of Splunk
- Describe the four phases of the distributed model
- Identify Splunk configuration files and directories
- Describe index-time precedence and search time precedence
- Use **`btool`** to retrieve Splunk configuration information
- Identify Splunk Data and System Administrator roles

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:
  - Accepts any text data as input
  - Parses the data into events
  - Stores events in indexes
  - Searches and reports

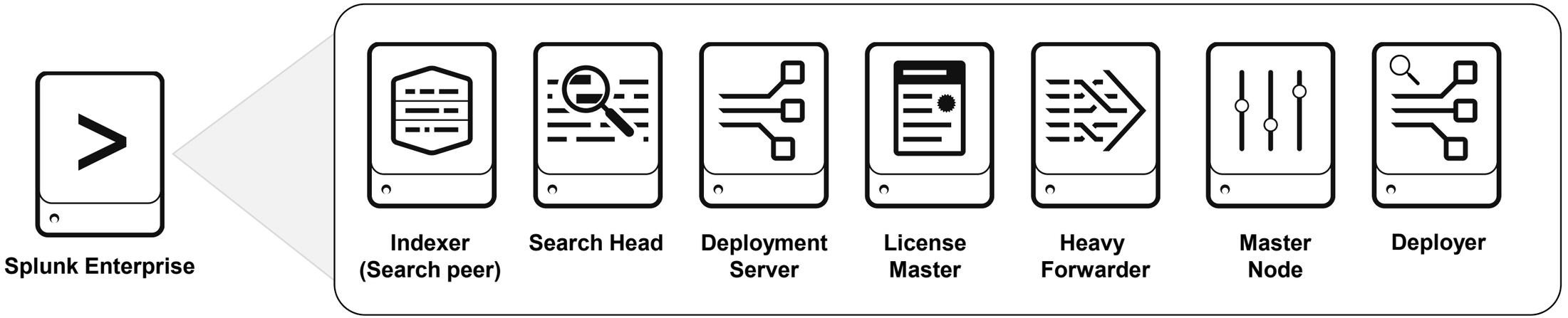


The Four Phases of the Distributed Model

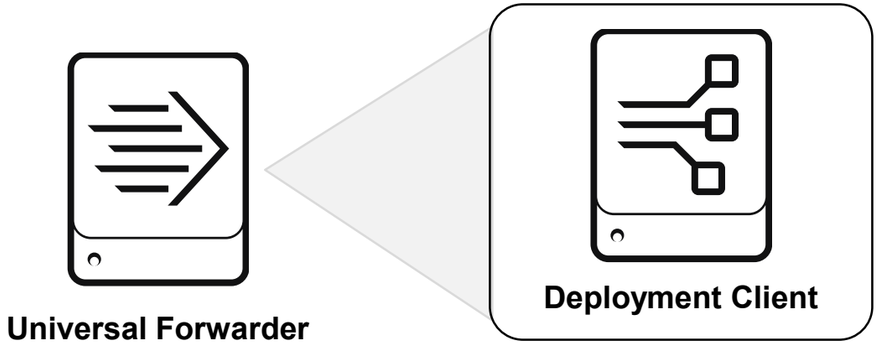


What Software Do You Install?

- Included in the Splunk Enterprise software package



- Included in the Universal Forwarder software package

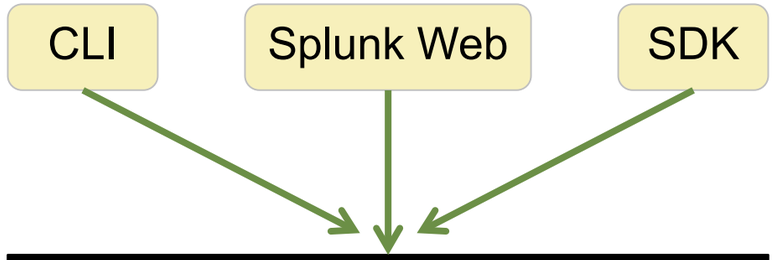


Note

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

Splunk Configuration Files

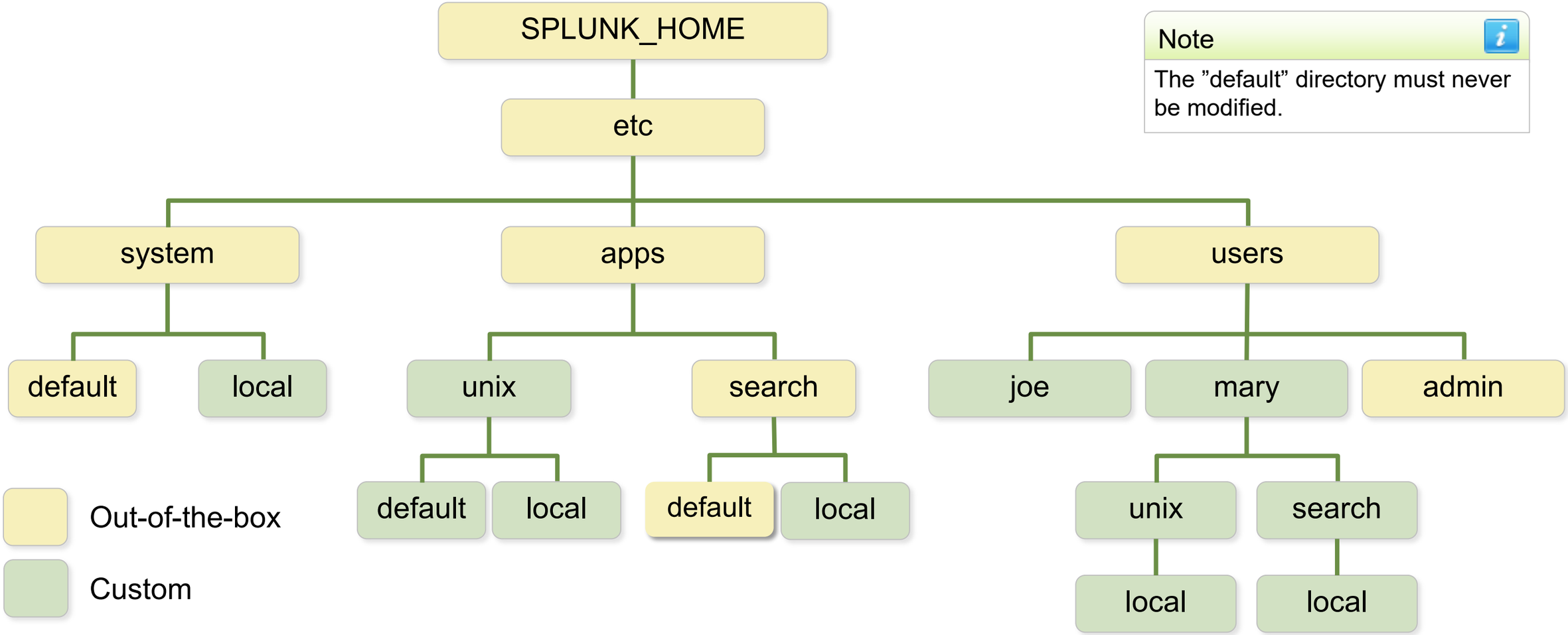
- Each configuration file governs a particular aspect of Splunk functionality
- Configuration changes are saved in **.conf** files under **SPLUNK_HOME/etc/**
  - **.conf** files are text files using a stanza and name/value (attribute) format
  - The syntax is case-sensitive
- You can change settings using Splunk Web, CLI, SDK, app install, and/or direct edit
- All **.conf** files have documentation and examples:
  - **SPLUNK_HOME/etc/system/README**
    - ***.conf.spec**
    - ***.conf.example**
    - Splunk documentation: docs.splunk.com




```
[default]
host=www

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

inputs.conf

Configuration Directories

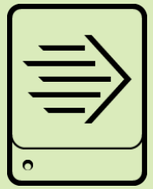
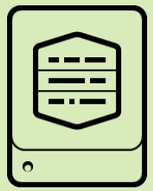


Note

The ”default” directory must never be modified.

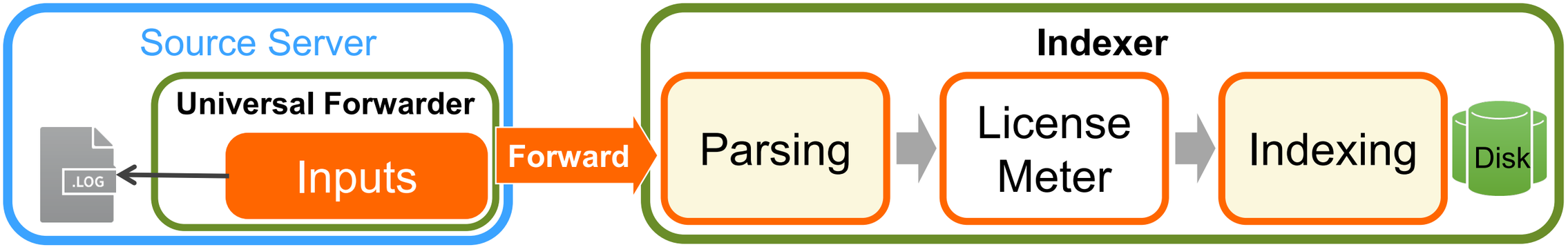
http://docs.splunk.com/Documentation/Splunk/latest/Admin/Listofconfigurationfiles

Commonly Used Configuration Files

Component	inputs.conf	props.conf	outputs.conf
Universal Forwarder 	Defines what data to collect	Limited parsing such as character encoding, refine MetaData, event breaks*	Defines where to forward the data
Indexer 	Defines what data to collect including data coming from forwarders	Refine MetaData at event level, event breaks, Time Extraction, TZ, data transformation	Does not need an **outputs.conf** as the indexer does not forward the data
Search Head 	Defines what data to collect including Splunk logs	Field Extractions (search-time), lookups, etc.	Defines where to forward the data. You may want to send the data to the indexer, especially the internal logs

Index-Time Process

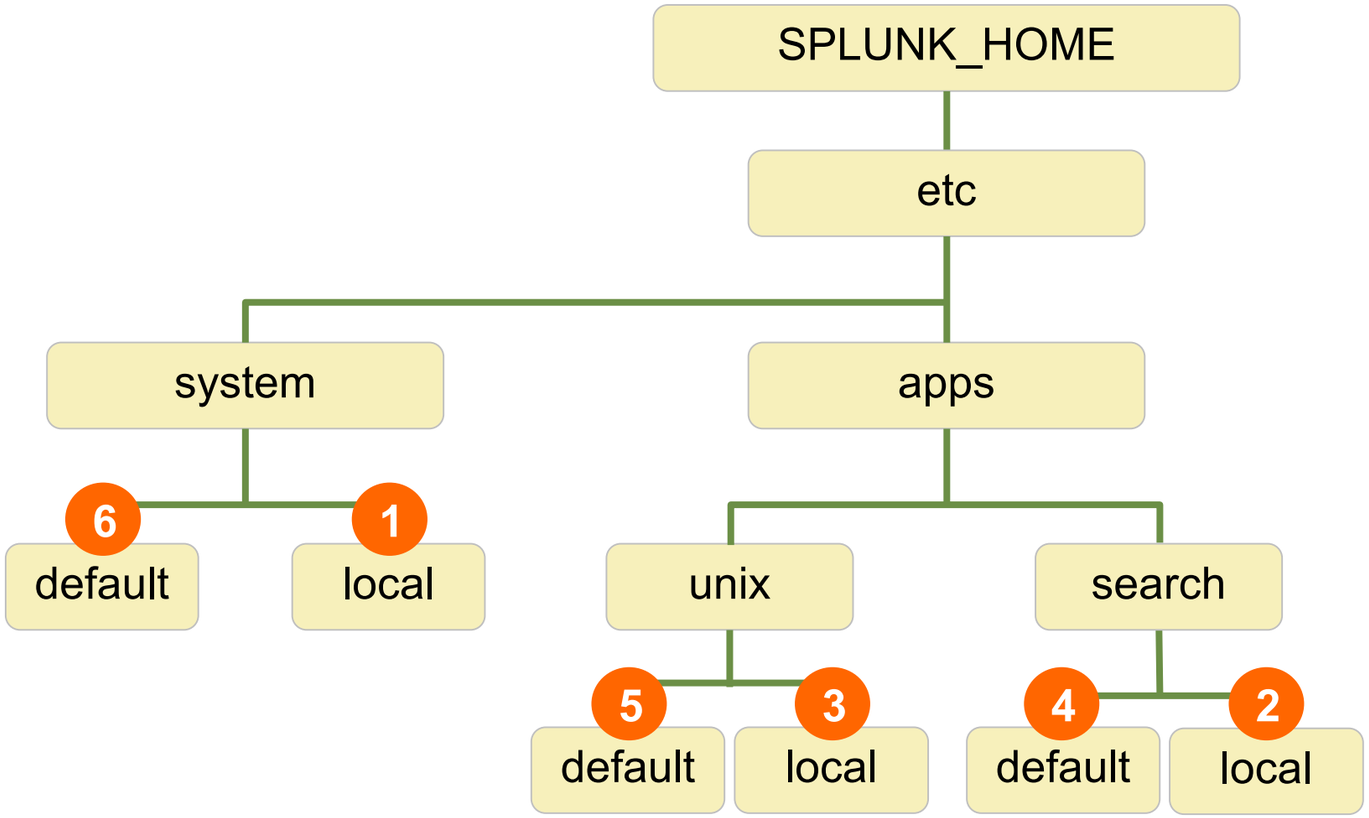
- Splunk index-time process (data ingestion) can be broken down into three phases:
  1. **Input phase**: handled at the source (usually a forwarder)
     - The data sources are being opened and read
     - Data is handled as streams and any configuration settings are applied to the entire stream
  2.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



Index-Time Merging of Configurations

- When Splunk starts, configuration files are merged together into a single run-time model for each file type
  - Regardless of the number of **`inputs.conf`** files in various apps or the system path, only one master inputs configuration model exists in memory at runtime
- If there are no duplicate stanzas or common settings between the files, the result is the union of all files
- If there are conflicts, the setting with the highest precedence is used
  - Remember that **`local`** always takes precedence over **`default`**

Index-Time Precedence

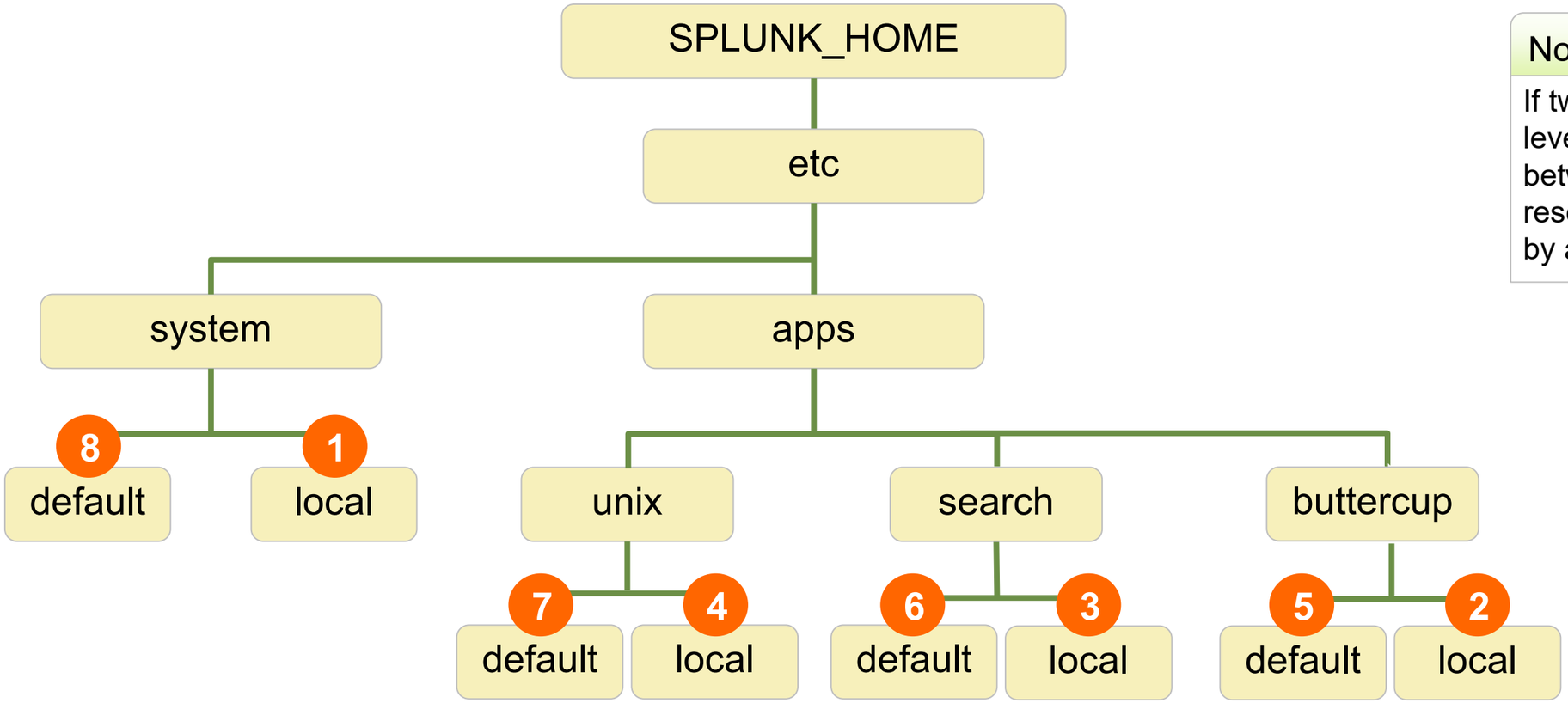


1. etc/system/local
2. etc/apps/search/local
3. etc/apps/unix/local
4. etc/apps/search/default
5. etc/apps/unix/default
6. etc/system/default

Note

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

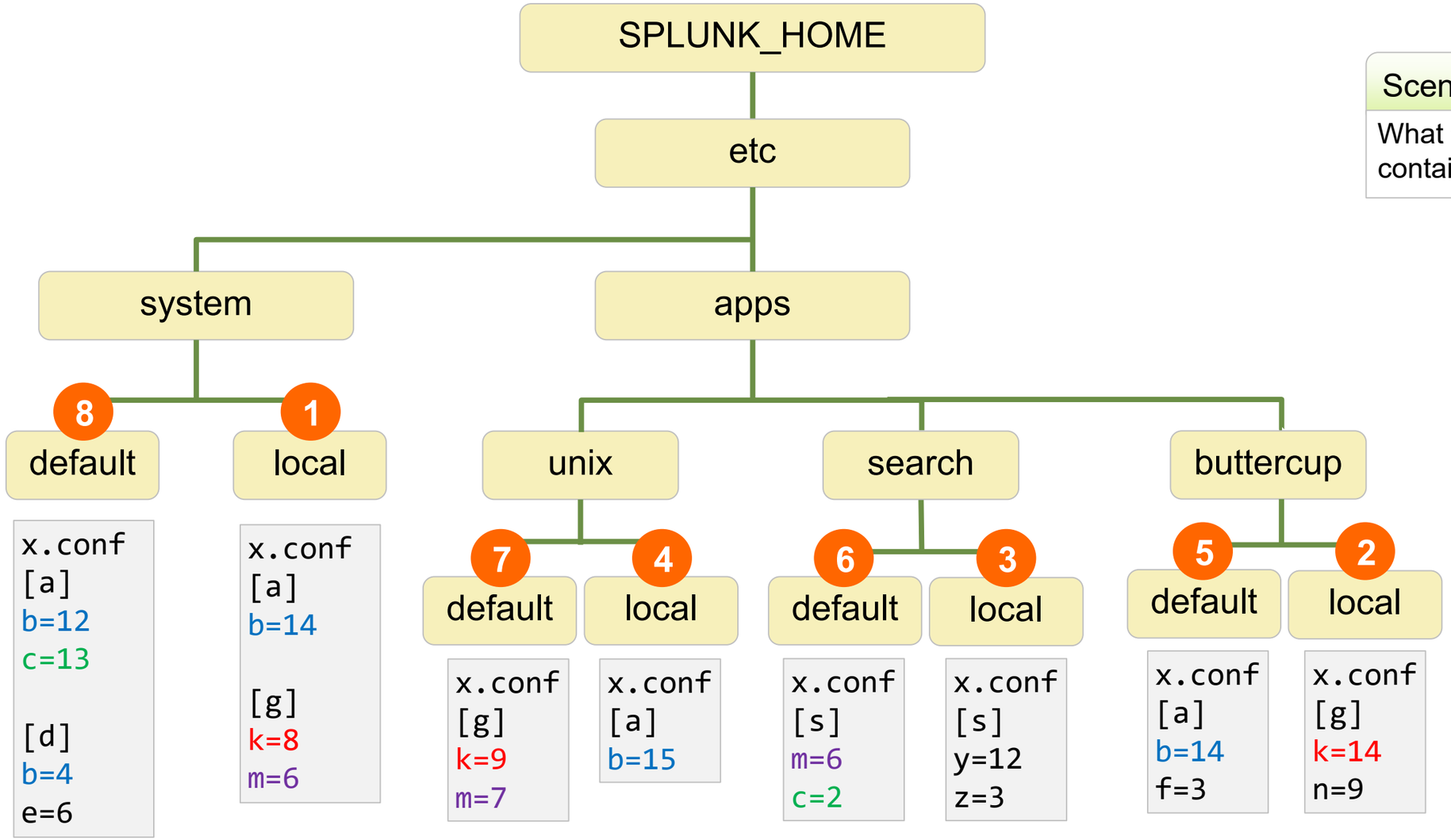
Index Time Precedence – Adding an App



Note

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

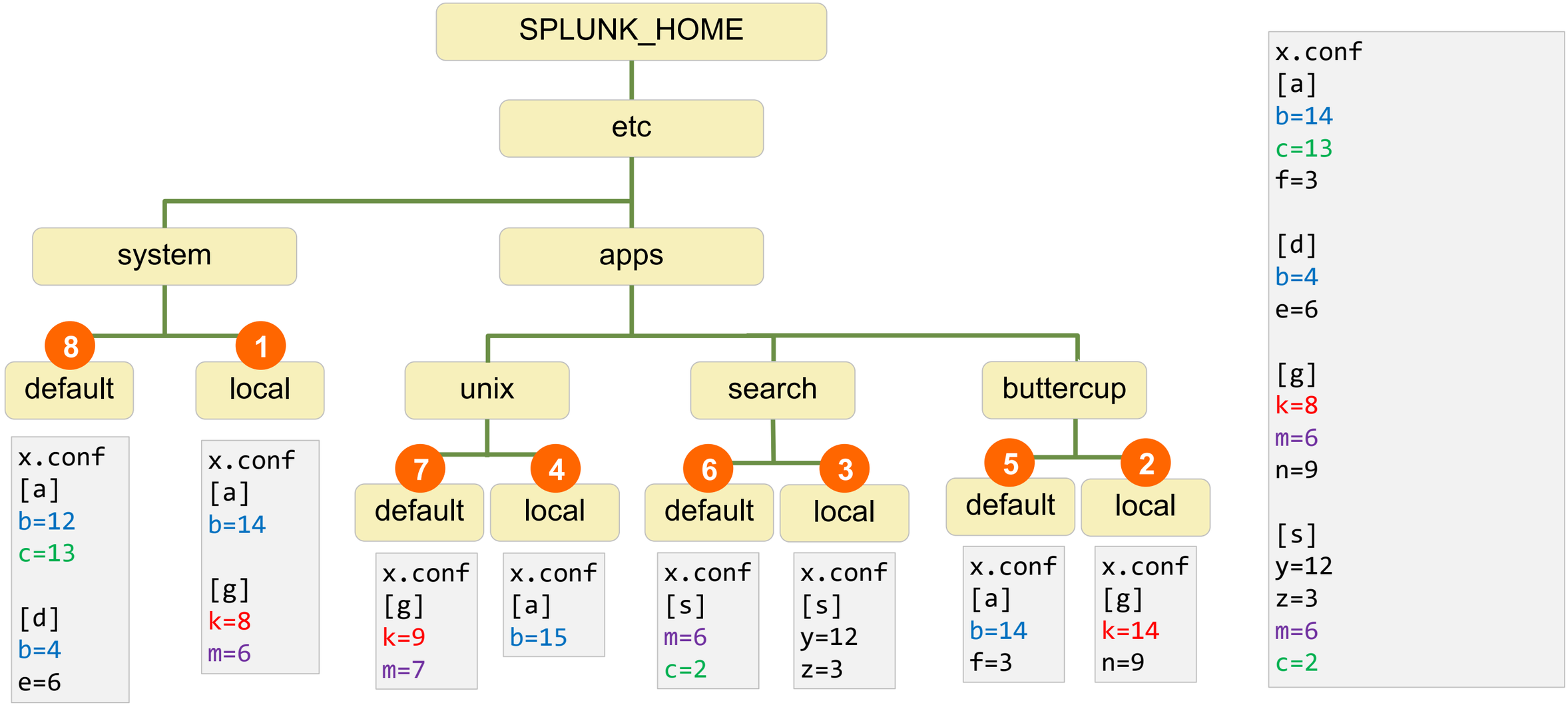
Index Time Precedence – Scenario



Scenario

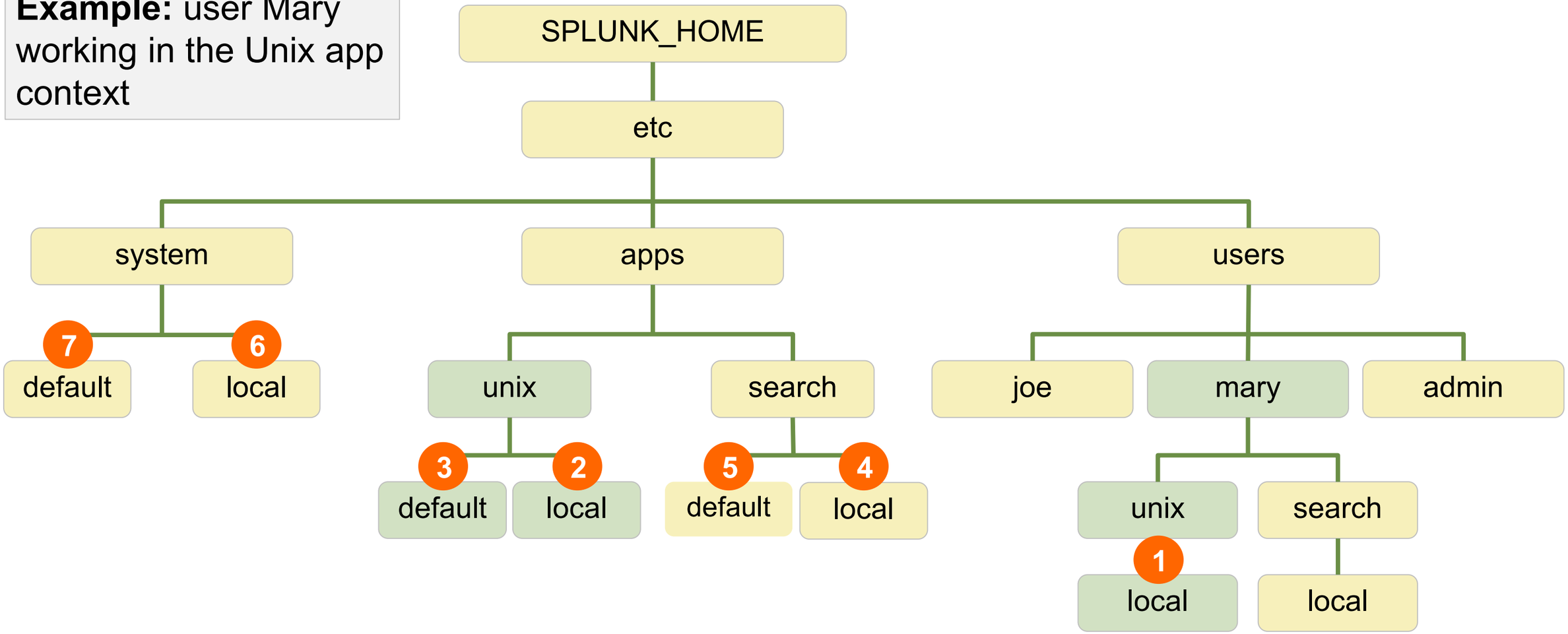
What will the merged **`x.conf`** file contain?

Index Time Precedence – Scenario (cont.)

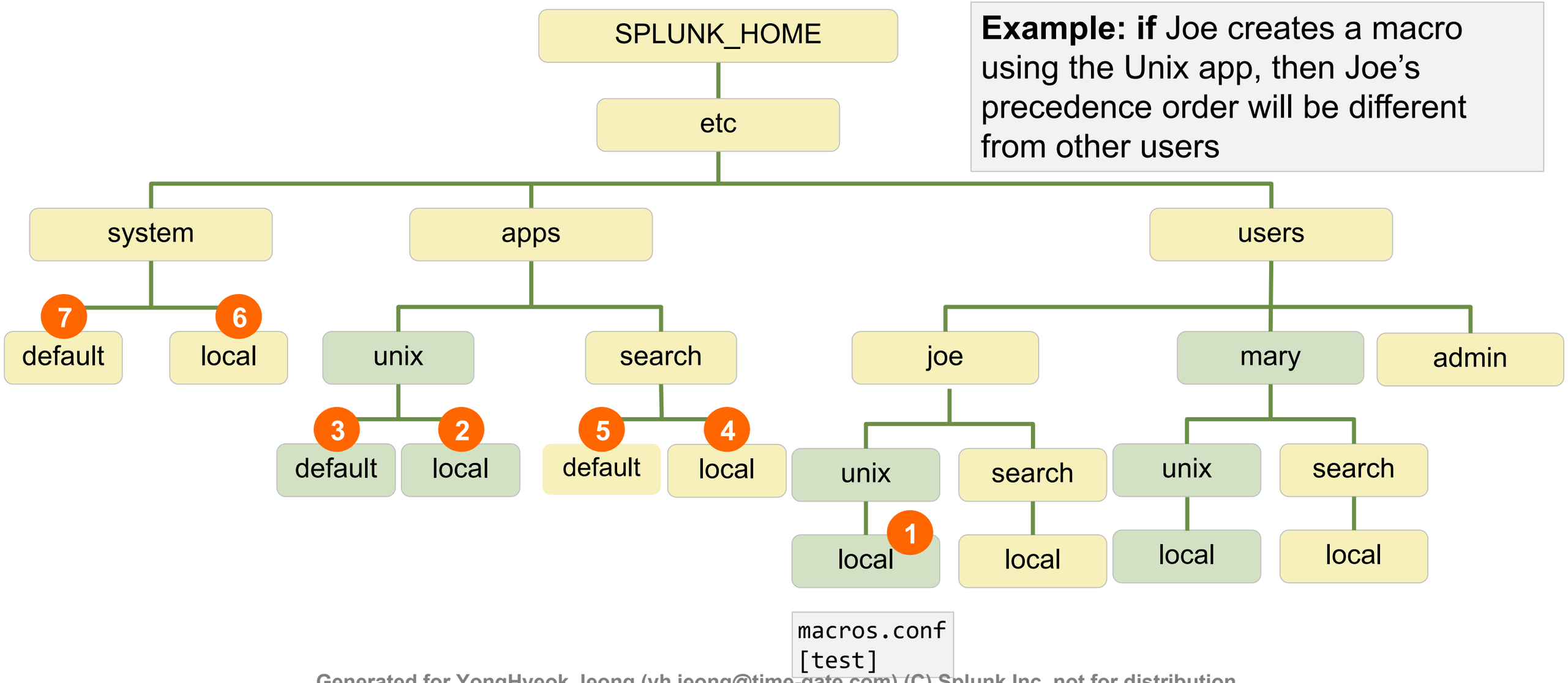


Search Time Precedence Order

Example: user Mary working in the Unix app context

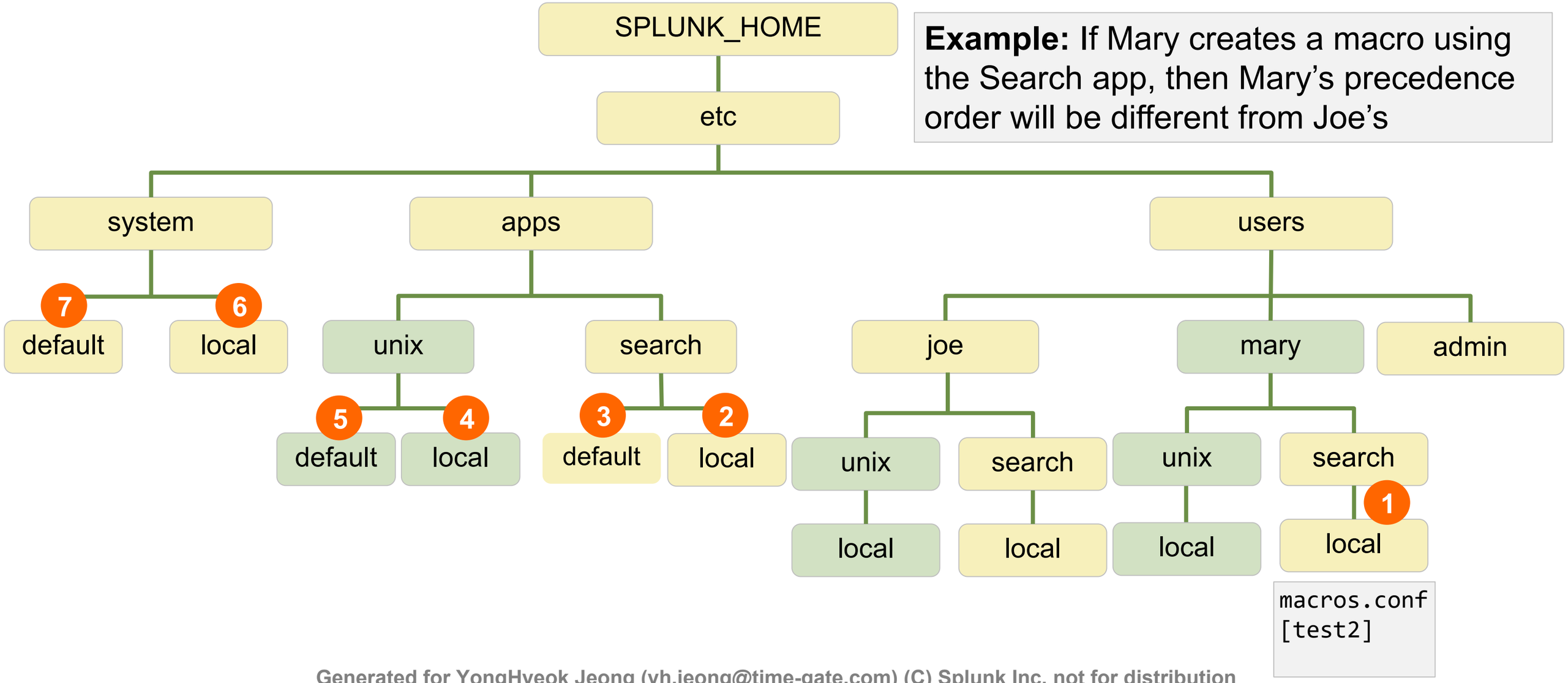


Search Time Precedence Order (Joe)



Example: if Joe creates a macro using the Unix app, then Joe's precedence order will be different from other users

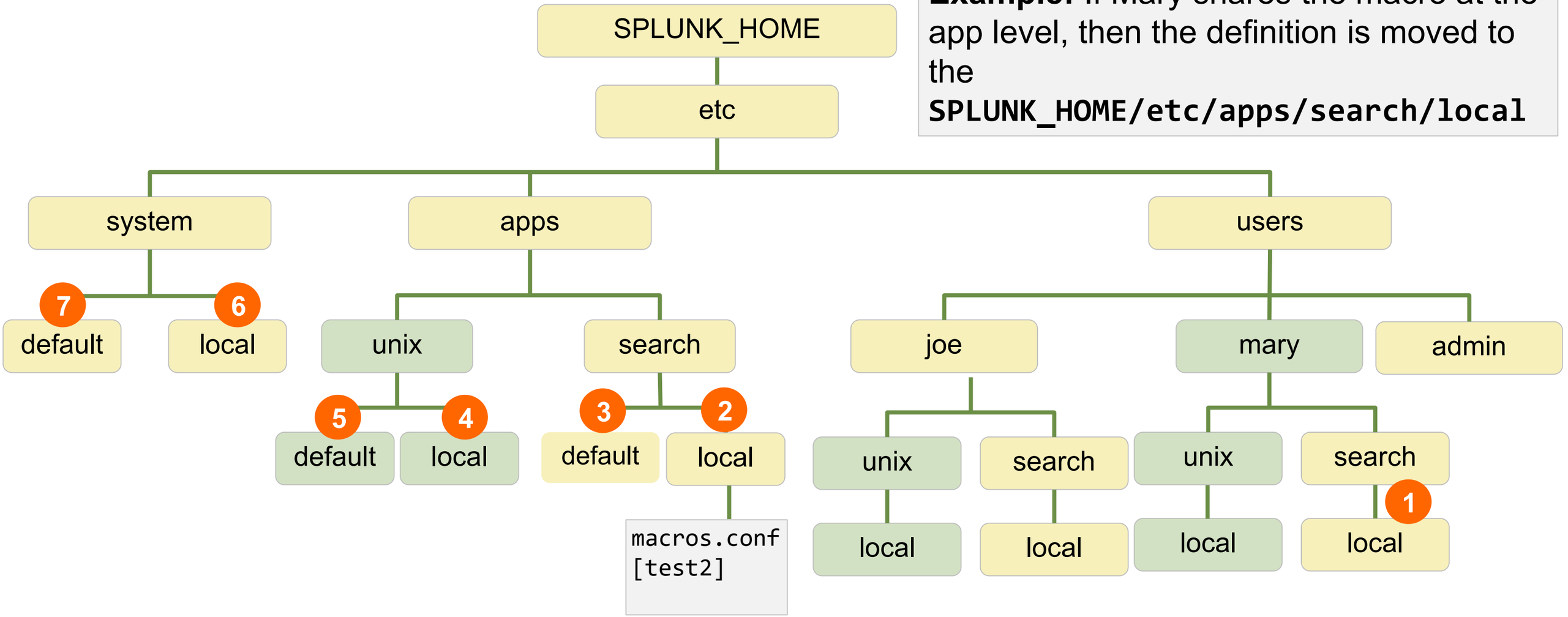
Search Time Precedence Order (Mary)



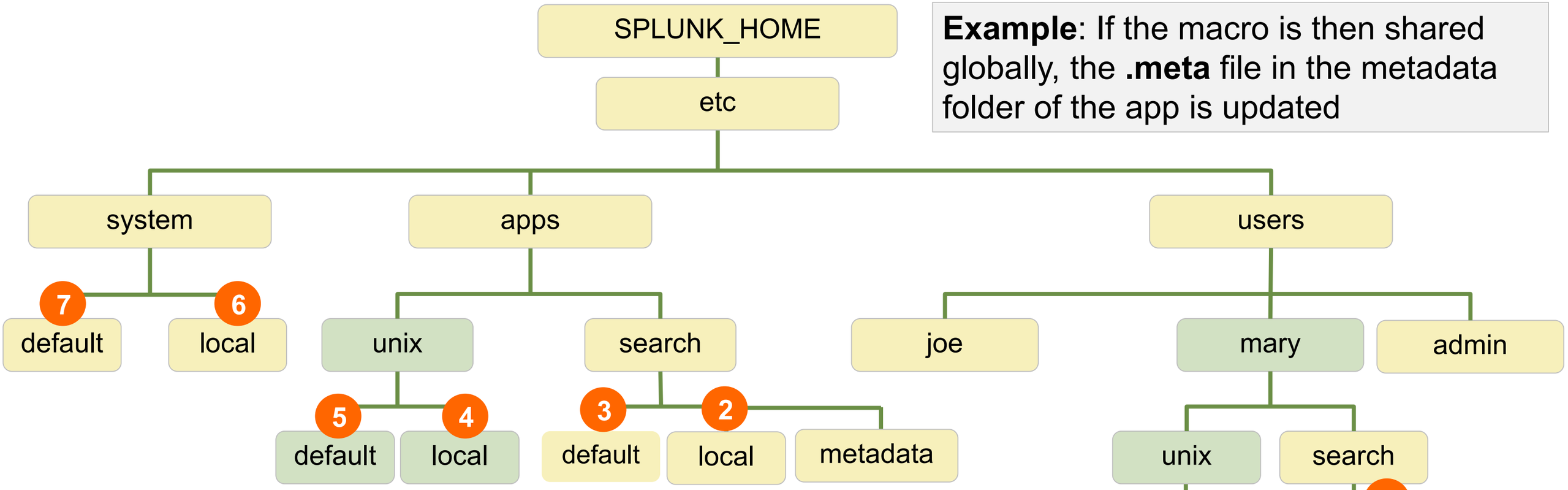
Example: If Mary creates a macro using the Search app, then Mary's precedence order will be different from Joe's

Search Time Precedence Order (Sharing KOs)

Example: If Mary shares the macro at the app level, then the definition is moved to the **`SPLUNK_HOME/etc/apps/search/local`**



Search Time Precedence Order (Sharing KOs) (cont.)



Example: If the macro is then shared globally, the **.meta** file in the metadata folder of the app is updated

```
local.meta
[macros/test2]
access = read : [ * ]
export = system
owner = mary
version = 7.1.0
modtime = 1537527828.580874000
```

Configuration Validation Command - **btool**

- **splunk btool** *conf-name* **list** [*options*]
  - Shows on-disk configuration for requested file
  - 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
```

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

`btool` Example

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/system/local/inputs.conf           host = myIndexer
  etc/system/default/inputs.conf         index = default
  etc/apps/search/local/inputs.conf      sourcetype = linux_secure
```

etc/system/local/inputs.conf

```
[monitor:///var/log/secure.log]
host=myIndexer
```

etc/apps/search/local/inputs.conf

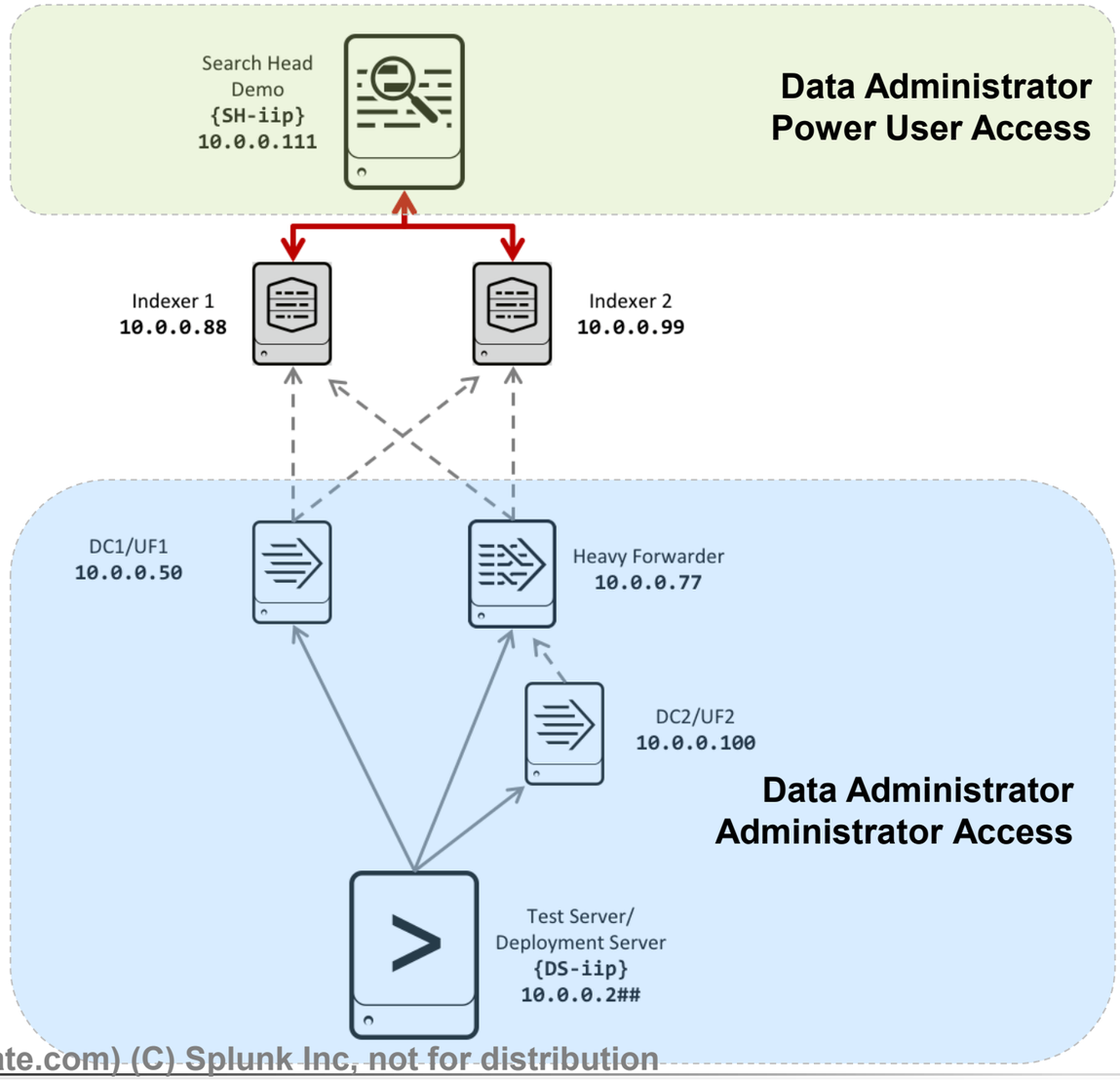
```
[monitor:///var/log/secure.log]
sourcetype=linux_secure
host=webserver
```


Data Administrator vs System Administrator

- The Splunk Data Administrator is primarily responsible for data onboarding and management efforts which include:
  - 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
- The Splunk System Administrator is primarily responsible for system management efforts which include:
  - Monitor MC and respond to system health alerts
  - Install and manage Splunk apps
  - Manage Splunk licensing
  - Manage Splunk configuration files and indexes
  - Manage Splunk users and authentication

Data Administrator Scenario

- The Data Administrator has admin role access to data sources and the data inputs
- The indexers are remote, the Data Administrator does not have access
- The Data Administrator has power role access to the search head
  - Searches will be used to verify data configuration



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

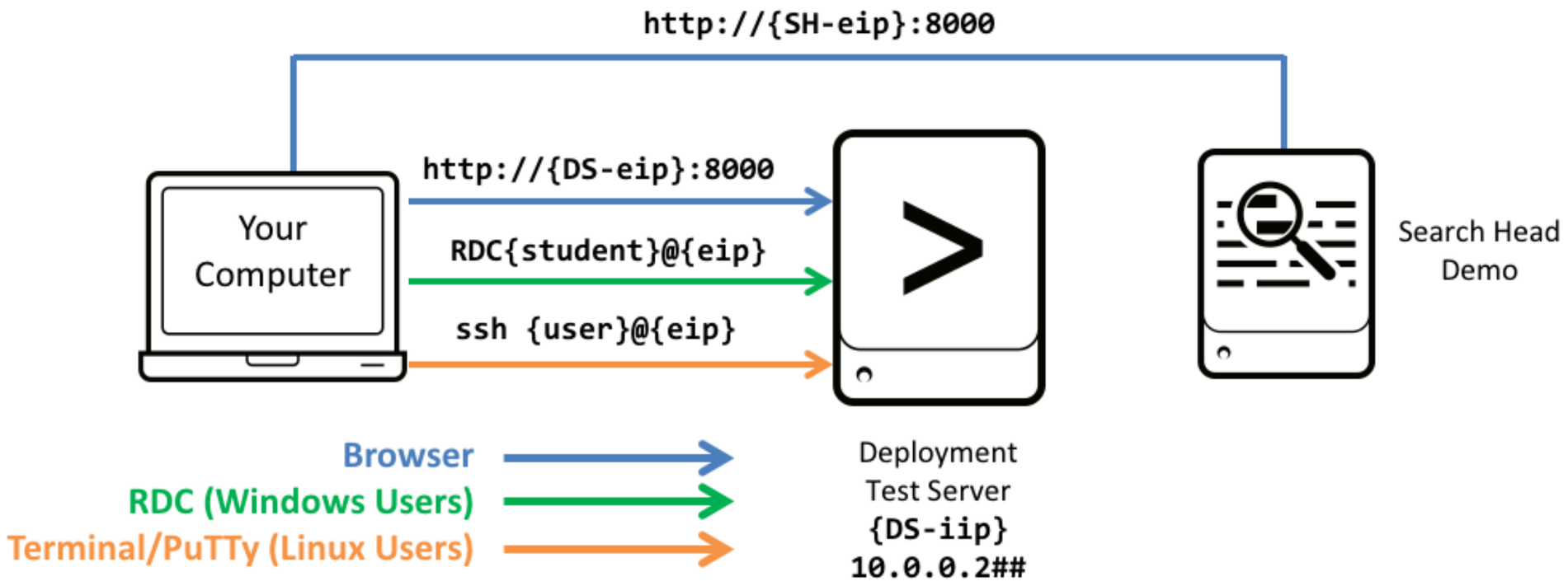
- Which configuration file tells a Splunk instance to ingest data?

  `inputs.conf`

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

  False. It is best is to put the configuration file in the **`local`** directory of your app.

Module 1 Lab Exercise – Environment Diagram



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

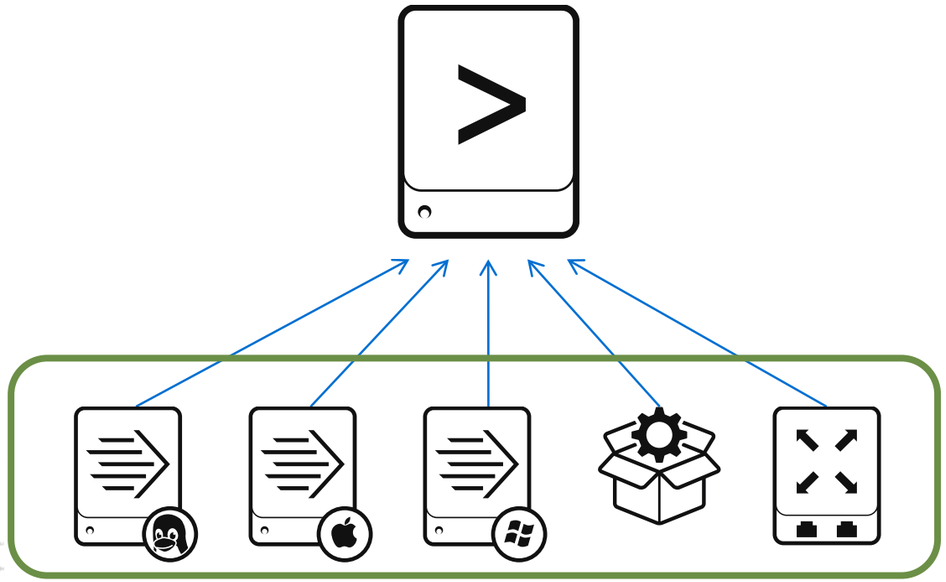
Module Objectives

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

Got Data?

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

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



Indexes any 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 you index a data source, Splunk assigns metadata values
  - The metadata is applied to the entire source
  - Splunk applies defaults if not specified
  - You can also override them 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 `main`

Input Phase vs. Parsing Phase

Input phase

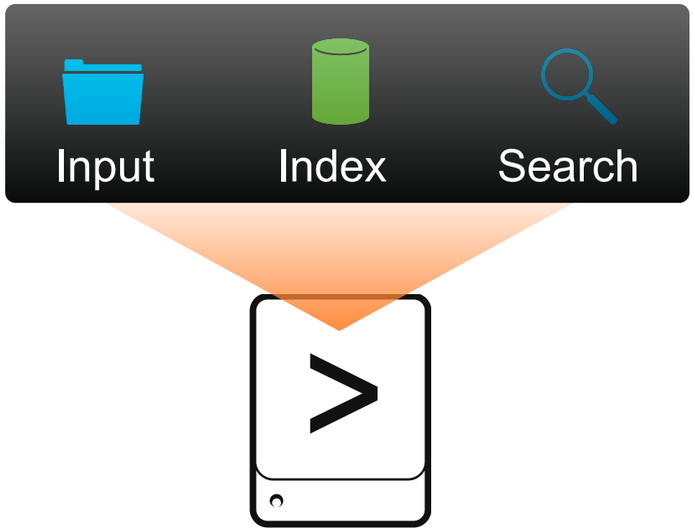
- **Most efficient, but low discrimination**
- Acquires data from source
- Sets initial metadata fields: source, sourcetype, host, index, etc.
- Converts character encoding
- Operates on the entire data stream
- Most configuration done in **`inputs.conf`** on forwarder
  - Some configuration is in **`props.conf`**

Parsing phase

- **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`**

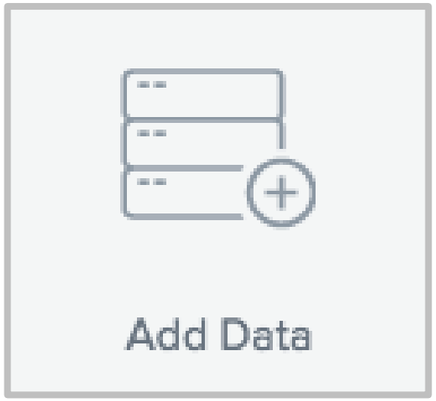
Initial Input Testing

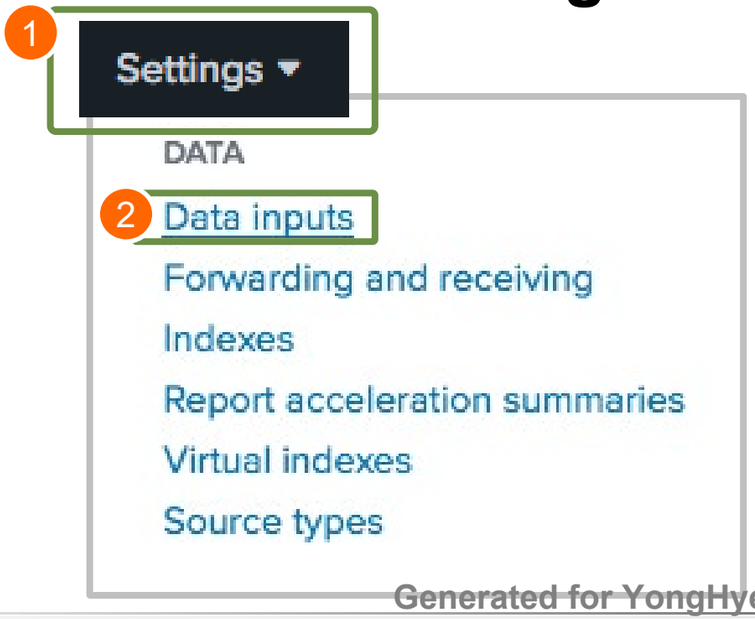
- Production data is usually on a remote system and is not on the indexer
  - Best Practice: Splunk forwarders forward the data
- For testing, you can use Splunk Web to sample a data file on a test server
- Use **Add Data**
  - Check to see if **sourcetype** and other settings are applied correctly
  - If not, delete the test data, change your test configuration, and try again

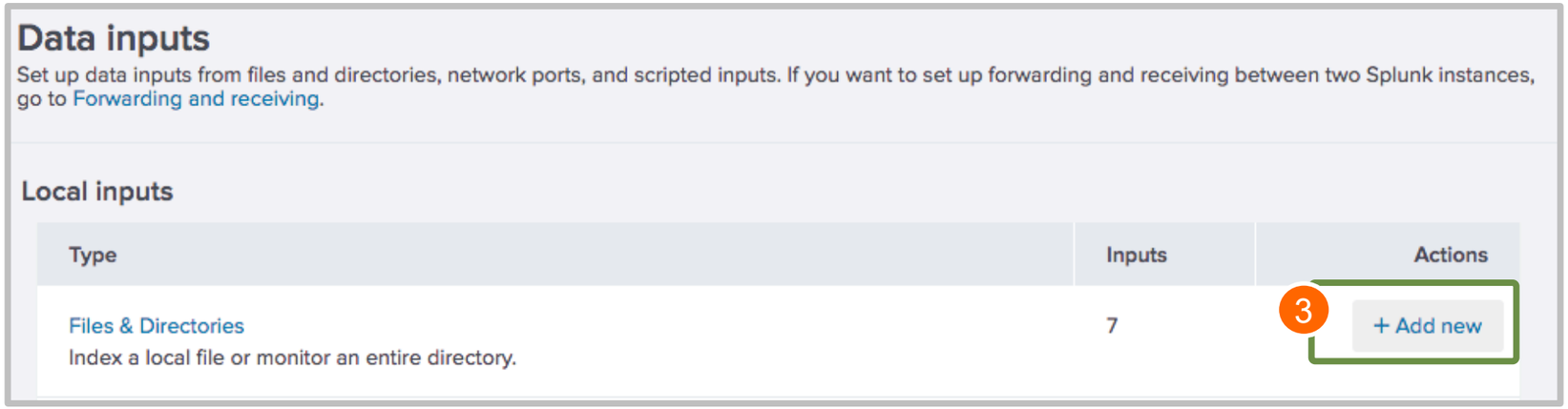


Adding an Input with Splunk Web

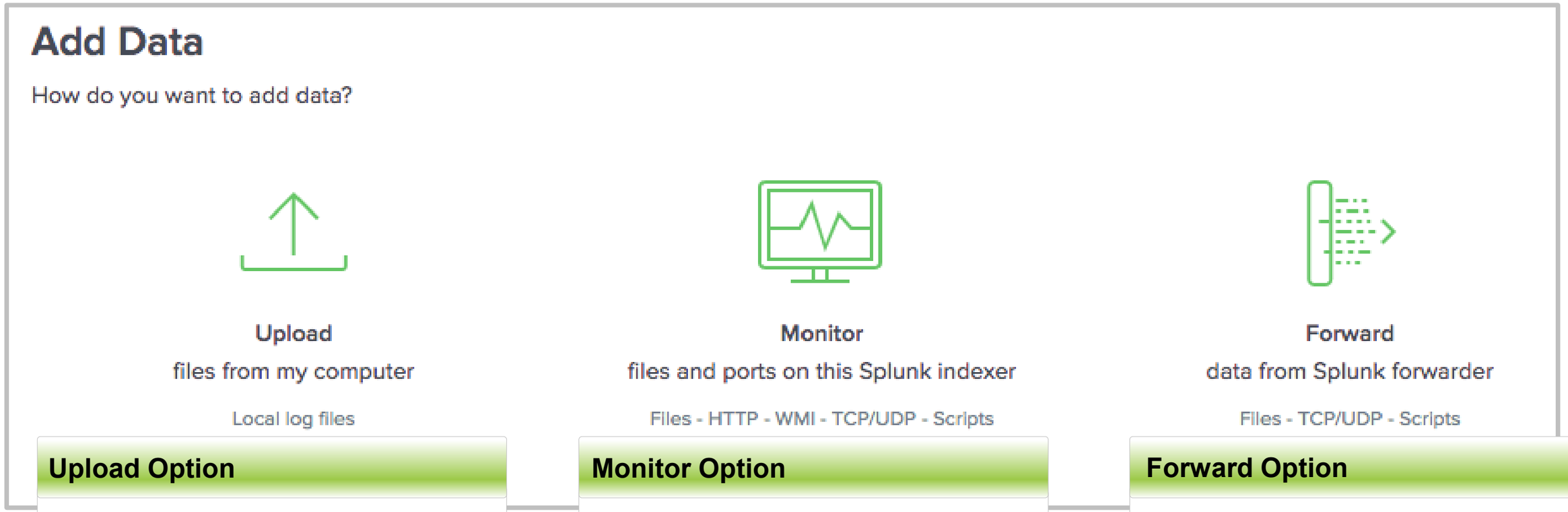
- Splunk admins have a number of ways to start the **Add Data** page
  - Click the **Add Data** icon
    - On the admin's **Home** page
    - On the **Settings** panel
  - Select **Settings** > **Data inputs** > **Add new**







Add Data Menu



Upload Option

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 create **`inputs.conf`**.

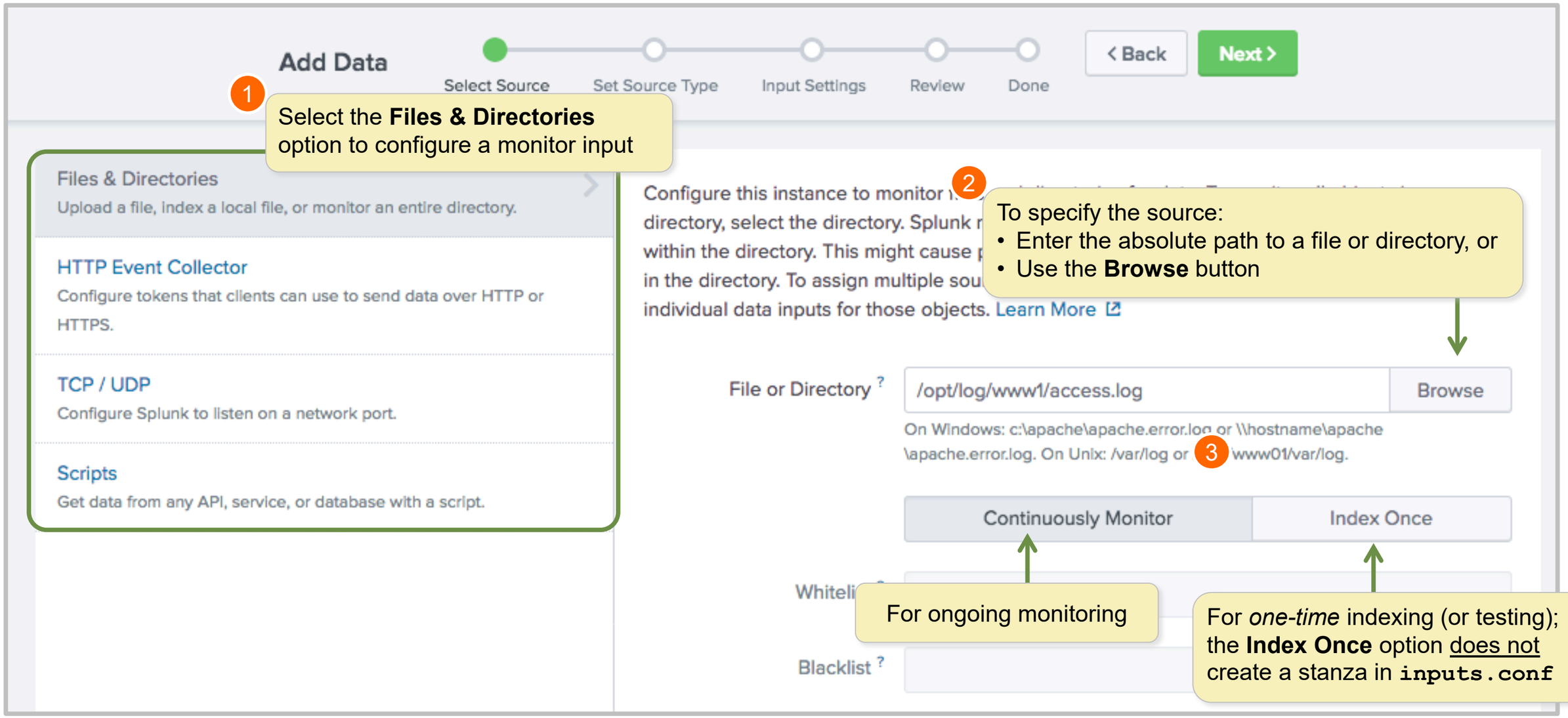
Monitor Option

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 Option

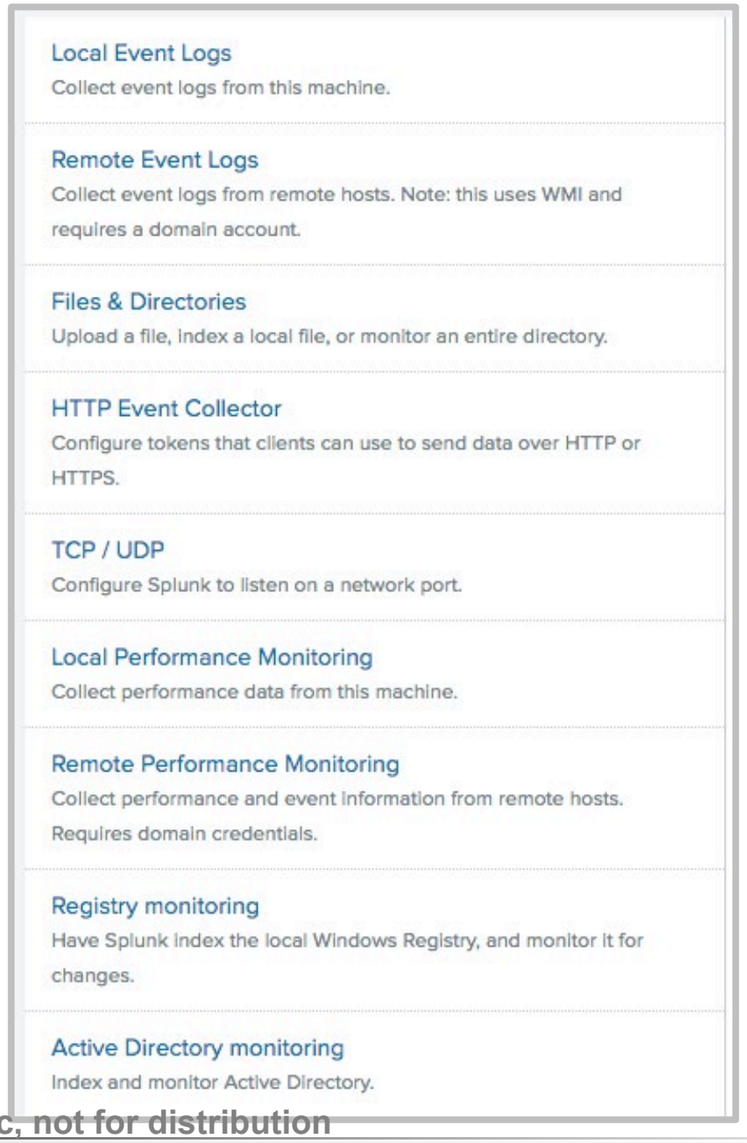
Main source of input in production environments. Remote machines gather and forward data to indexers over a receiving port.

Select Source



Select Source From Windows Instance

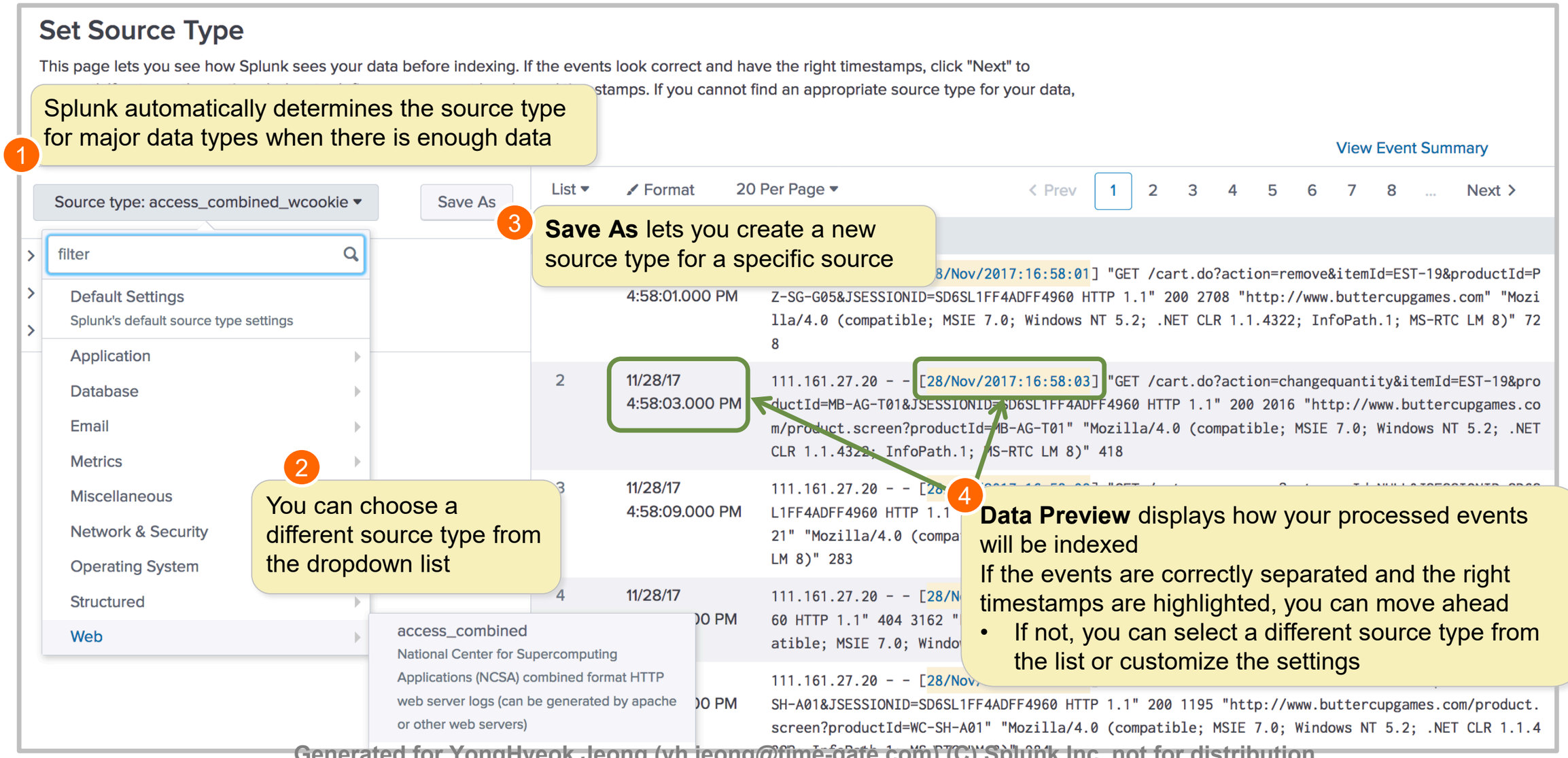
- On Windows Splunk instances, there are additional Windows-specific source options
- To monitor a shared network drive, enter the path manually:
  - **`<host>/<path>`** on *nix
  - **`\\<host>\<path>`** on Windows
  - Make sure Splunk has read access to the mounted drive



Understanding Source Types

- **Source type** is Splunk's way of categorizing the type of data
  - Splunk indexing processes frequently reference source type
  - Many searches, reports, dashboards, apps, etc. also rely on source type
- Splunk will try to determine the source type for you
  - If Splunk recognizes the data, then it assigns one from the pretrained sourcetypes
  - If one is explicitly specified, then Splunk will not try to determine the source type
    - You can explicitly set source type with Splunk Web, CLI, or by modifying **`inputs.conf`**
  - Otherwise, Splunk uses the name of the file as the source type
- You can also add source types by installing apps, which often define source types for their inputs

Set Source Type (Data Preview Interface)

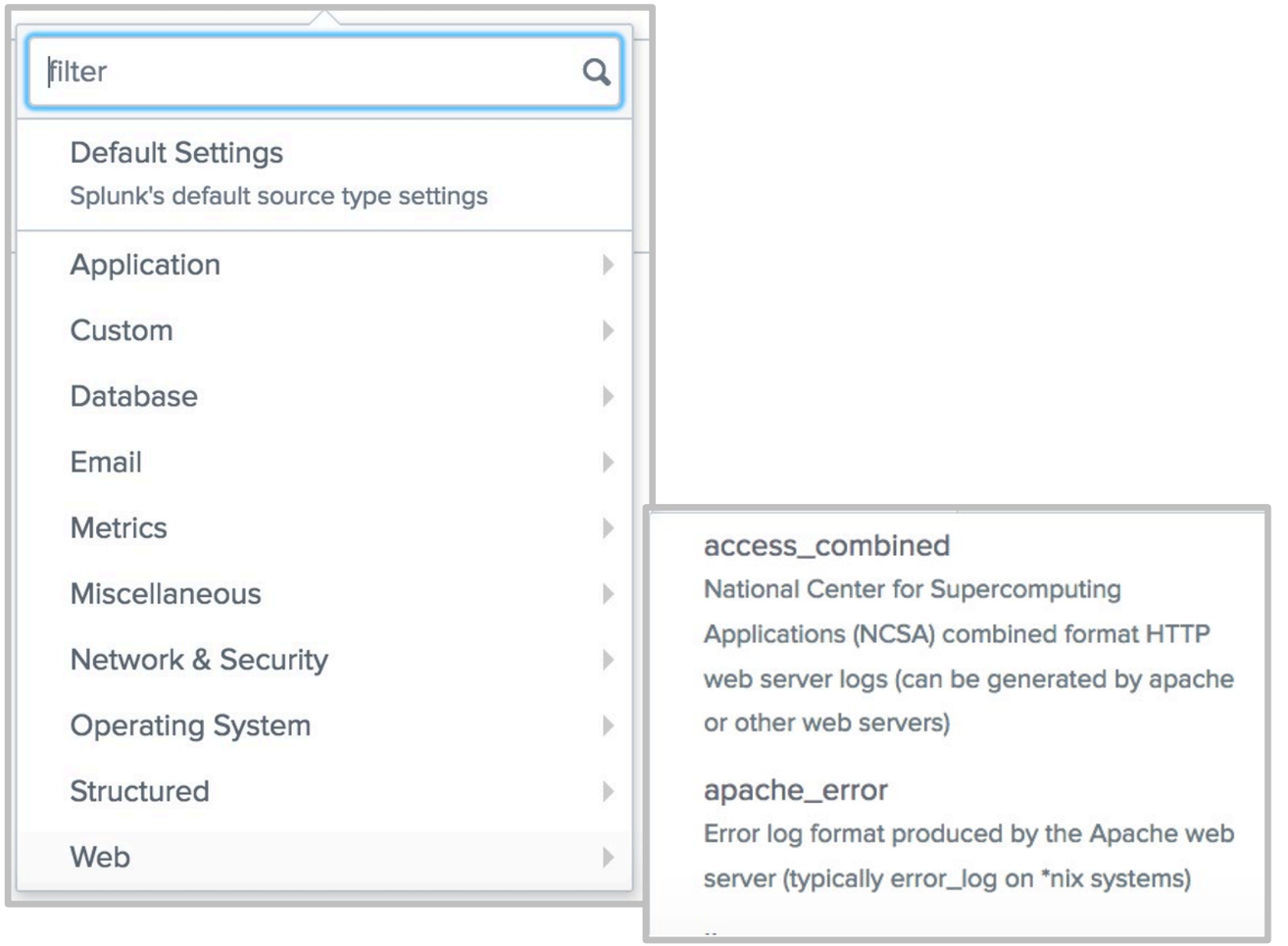


Set Source Type (Data Preview Interface) (cont.)

1. Splunk automatically determines the source type for major data types when there is enough data
2. You can choose a different source type from the dropdown list
3. Or, you can create a new source type name for the specific source
4. **Data preview** displays how your processed events will be indexed
   - If the events are correctly separated and the right timestamps are highlighted, you can move ahead
     - If not, you can select a different source type from the list or customize the settings

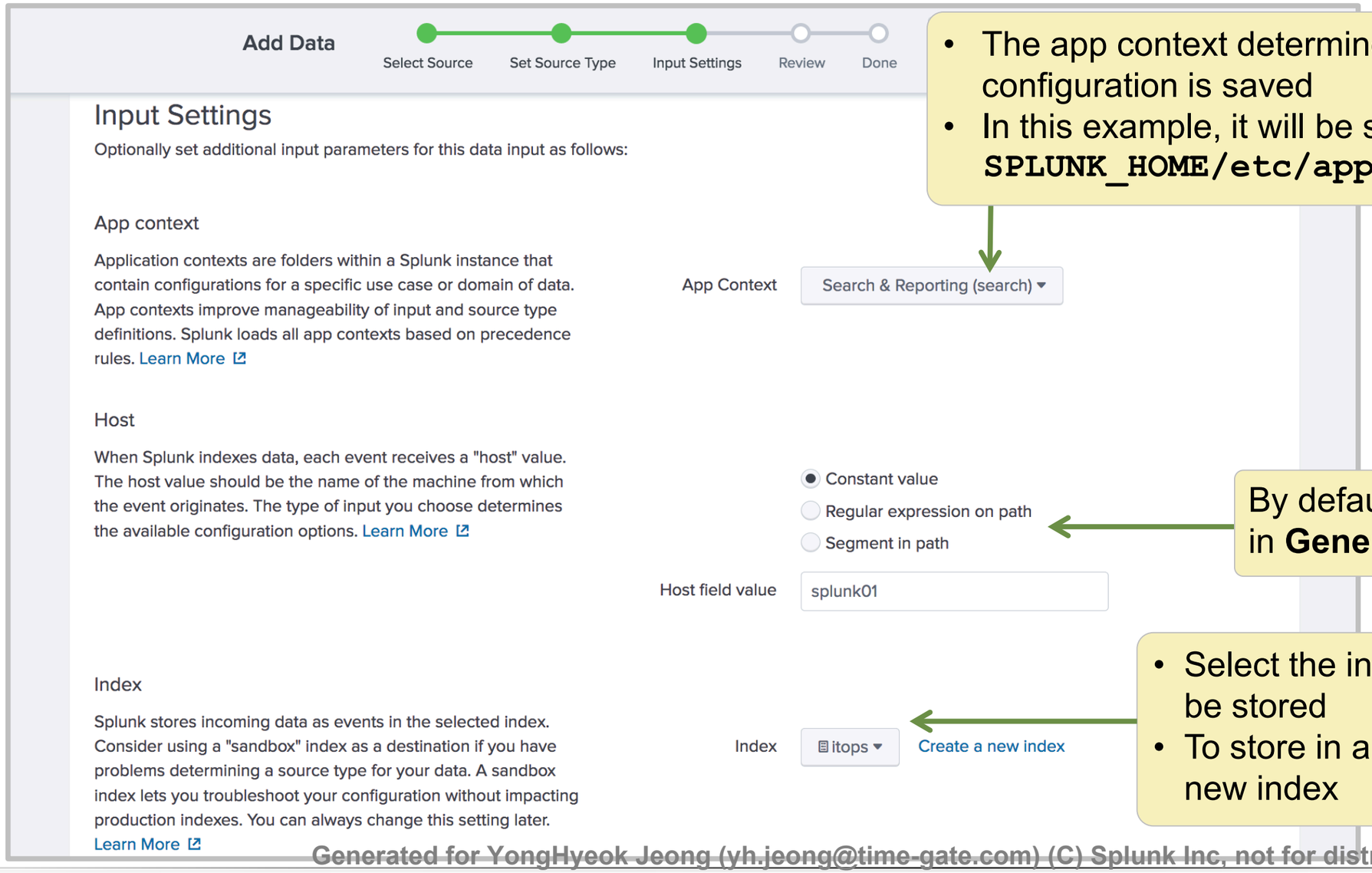
Pretrained Source Types

- Splunk has default settings for many types of data
- The docs also contain a list of source types that Splunk automatically recognizes
- Splunk apps can be used to define additional source types



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

Input Settings



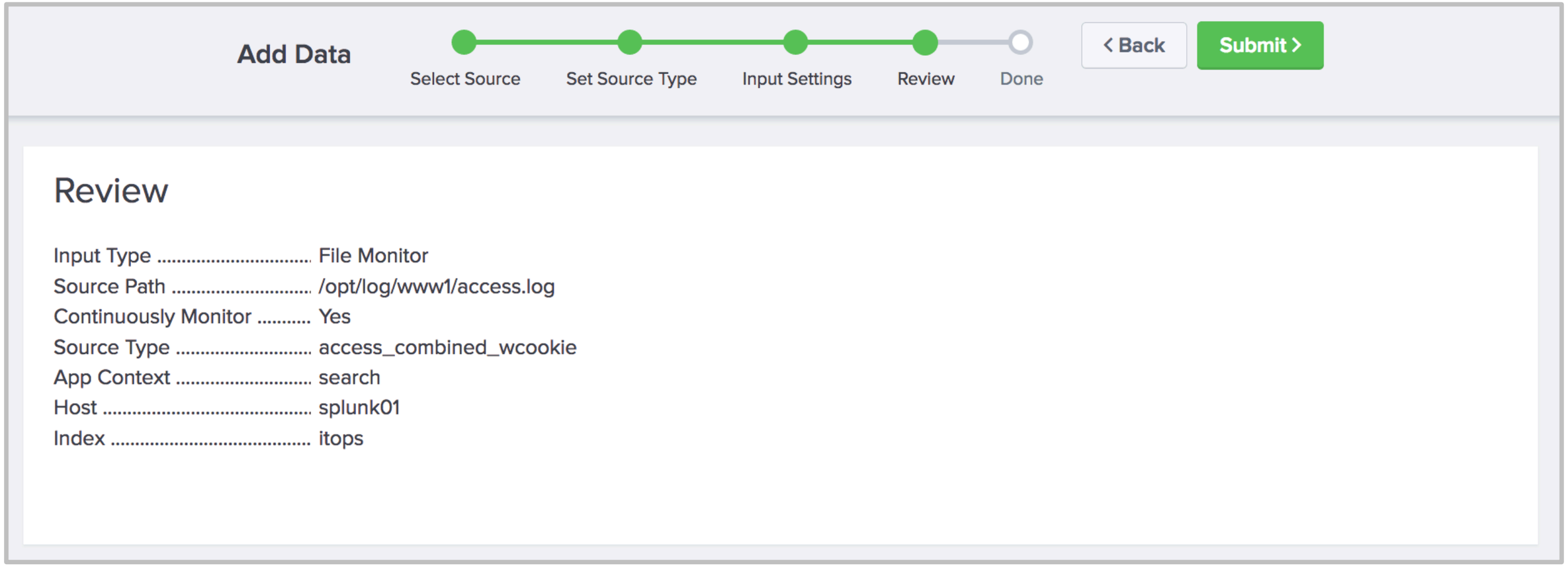
- The app context determines where your input configuration is saved
- In this example, it will be saved in: **`SPLUNK_HOME/etc/apps/search/local`**

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

- Select the index where this input should be stored
- To store in a new index, first create the new index

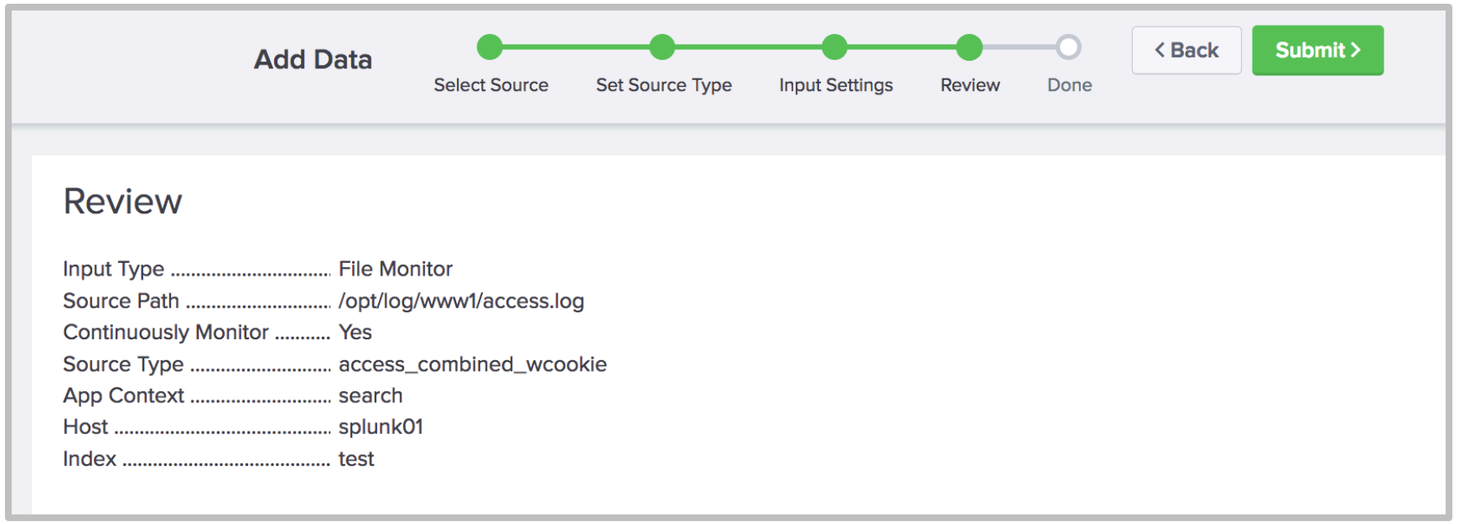
Review

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



What Happens Next?

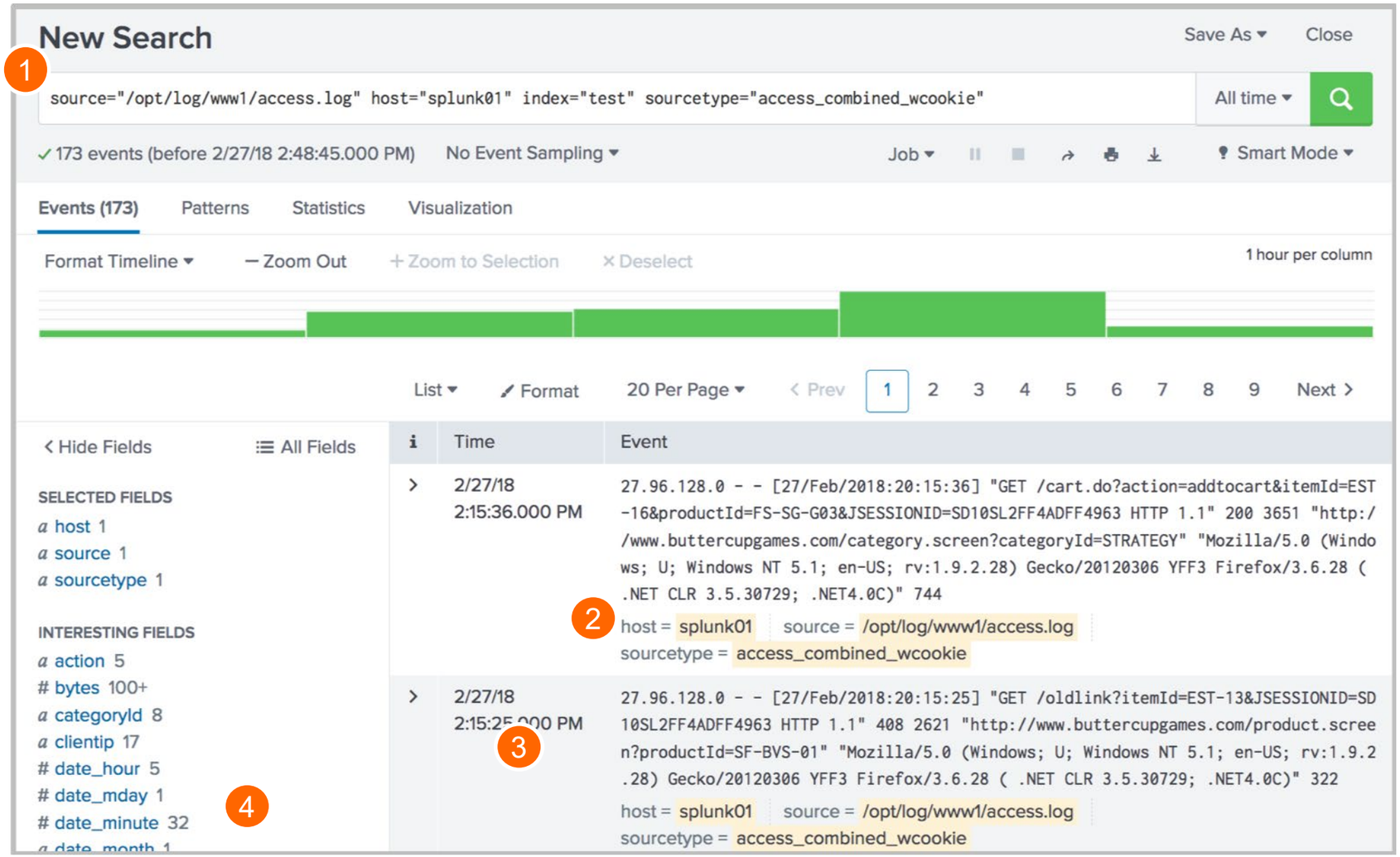
- Indexed events are available for immediate search
  - However, it may take a minute for Splunk to *start* indexing the data
- You are given other options to do more with your data
- The input configuration is saved in:
  `etc/apps/<app>/local/inputs.conf`



Note: **`inputs.conf`** is ***not*** created when **Upload** or **Index Once** is selected

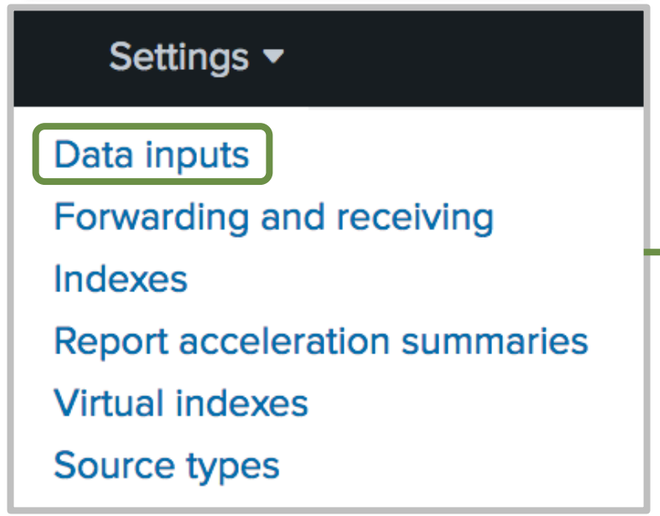
Verify your Input

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



View 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.

Inputs handled by this server

Local inputs

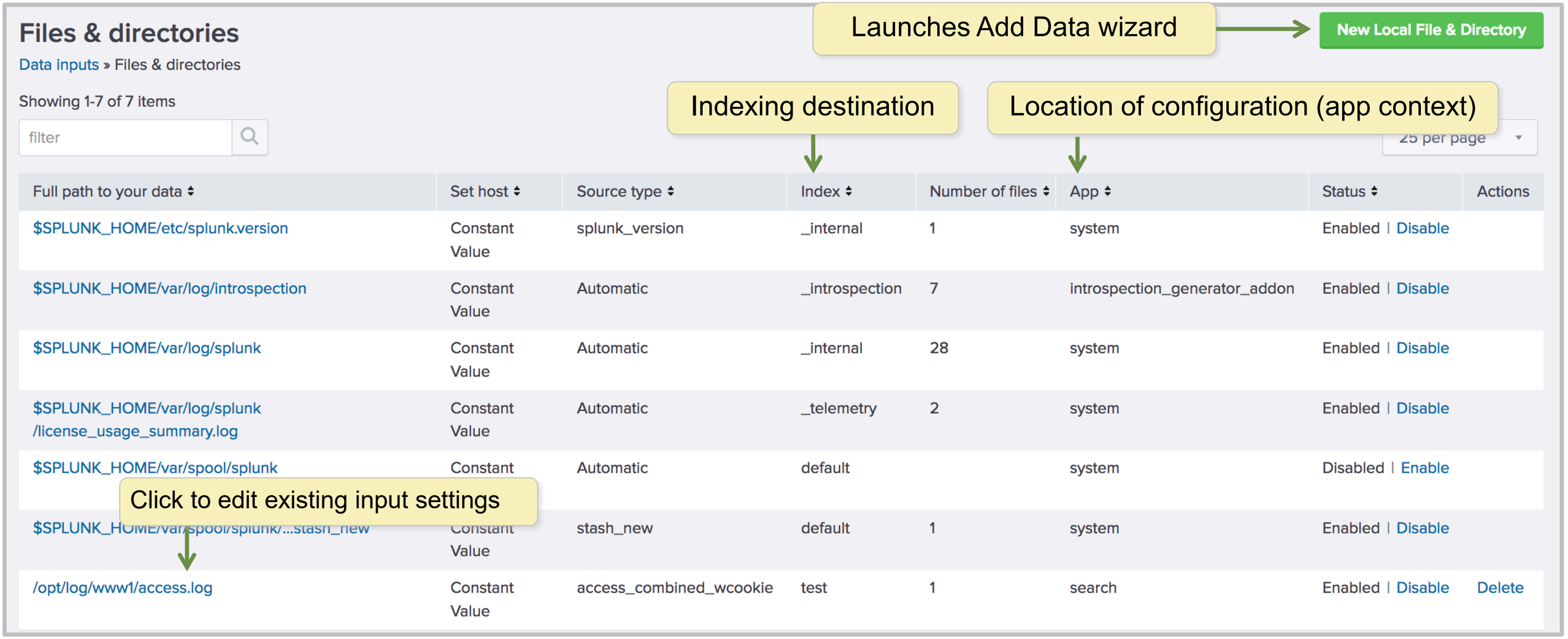
Type	Inputs	Actions
Files & Directories Index a local file or monitor an entire directory.	7	+ Add new
HTTP Event Collector Receive data over HTTP or HTTPS.	0	+ Add new
TCP Listen on a TCP port for incoming data, e.g. syslog.	0	+ Add new
UDP Listen on a UDP port for incoming data, e.g. syslog.	0	+ Add new
Scripts Run custom scripts to collect or generate more data.	5	+ Add new

Inputs handled by remote instances but configured from this deployment server

Forwarded inputs

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

View Configured Inputs (cont.)



Files & directories

Data inputs » Files & directories

Showing 1-7 of 7 items

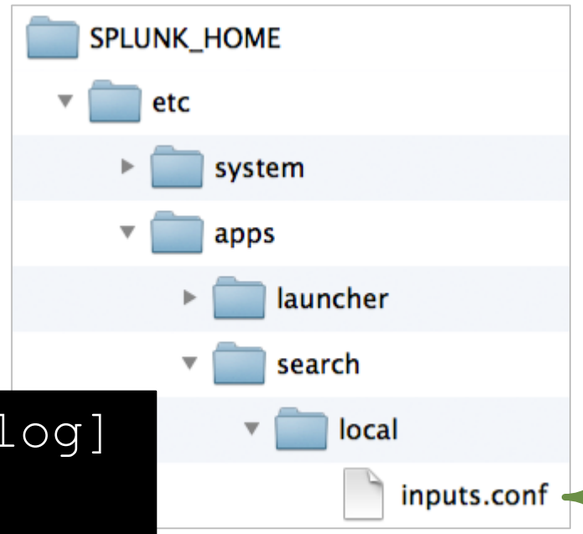
filter

25 per page

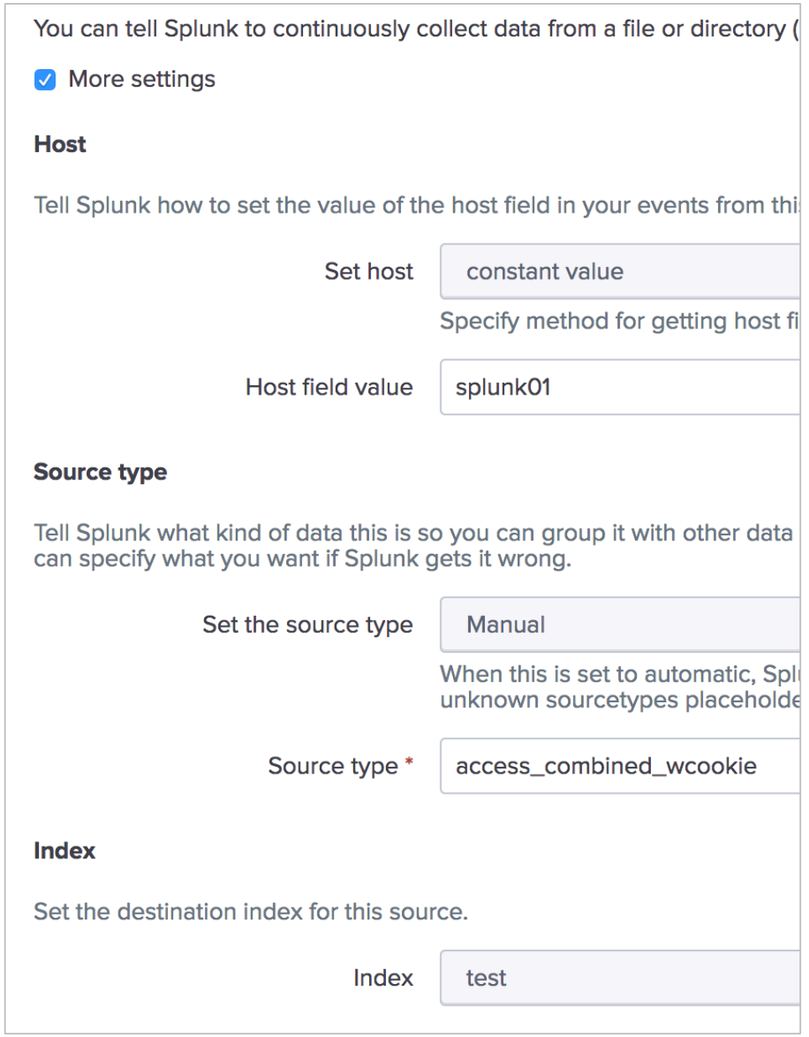
Full path to your data	Set host	Source type	Index	Number of files	App	Status	Actions
$SPLUNK_HOME/etc/splunk.version	Constant Value	splunk_version	_internal	1	system	Enabled | Disable	
$SPLUNK_HOME/var/log/introspection	Constant Value	Automatic	_introspection	7	introspection_generator_addon	Enabled | Disable	
$SPLUNK_HOME/var/log/splunk	Constant Value	Automatic	_internal	28	system	Enabled | Disable	
$SPLUNK_HOME/var/log/splunk /license_usage_summary.log	Constant Value	Automatic	_telemetry	2	system	Enabled | Disable	
$SPLUNK_HOME/var/spool/splunk	Constant	Automatic	default		system	Disabled | Enable	
$SPLUNK_HOME/var/spool/splunk/...stash_new	Constant Value	stash_new	default	1	system	Enabled | Disable	
/opt/log/www1/access.log	Constant Value	access_combined_wcookie	test	1	search	Enabled | Disable	Delete

Inputs.conf

- To put the input into production, edit the target index setting in **`inputs.conf`**, or
- Repeat the Add Data steps
  - Inputs you create are saved in the target app's **`local/inputs.conf`** file




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



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

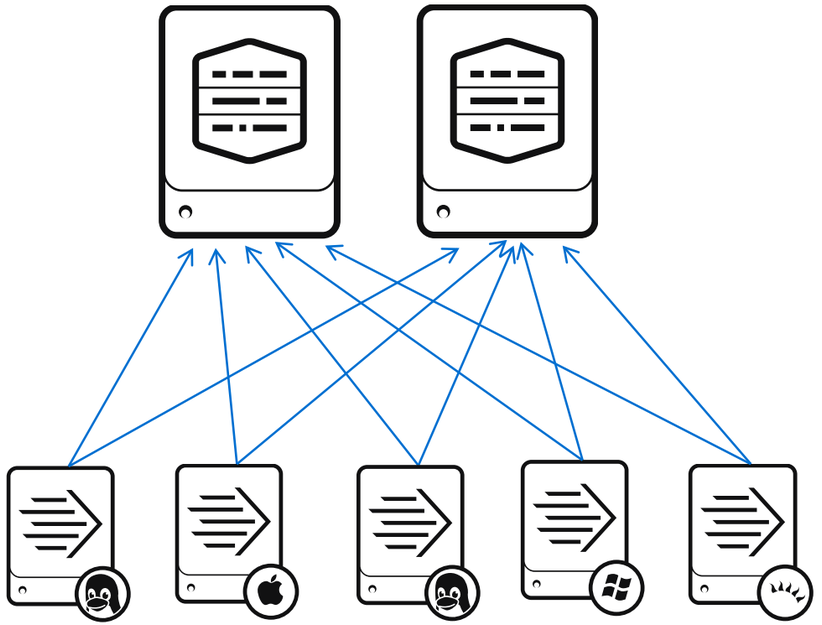
Module 3: Forwarder Configuration

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 a production environment
  - Splunk indexers run on dedicated servers
  - The data you want is on remote machines
  - Communicates via TCP
- Install a **forwarder** on a remote machine to
  - Gather data
  - Send it across the network to the Splunk indexer(s)
- Indexers listen on a **receiving port** for the forwarded data



Universal Forwarder

- Universal forwarder gathers data from a host and sends it to indexers
- Specifically designed to run on production servers
  - Minimal CPU and memory usage
  - Output bandwidth constrained to 256 KBps by default
  - No web interface, cannot search or index
- A separate installation binary
  - Built-in license, no limits

Best Practice: use the Universal Forwarder

https://www.splunk.com/blog/2016/12/12/universal-or-heavy-that-is-the-question.html

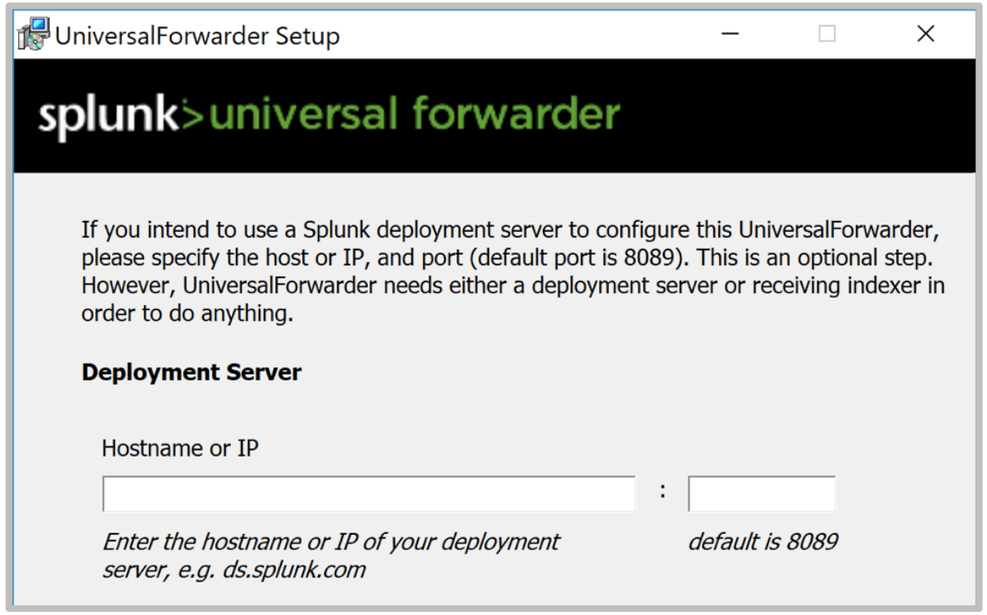
Installing Universal Forwarder

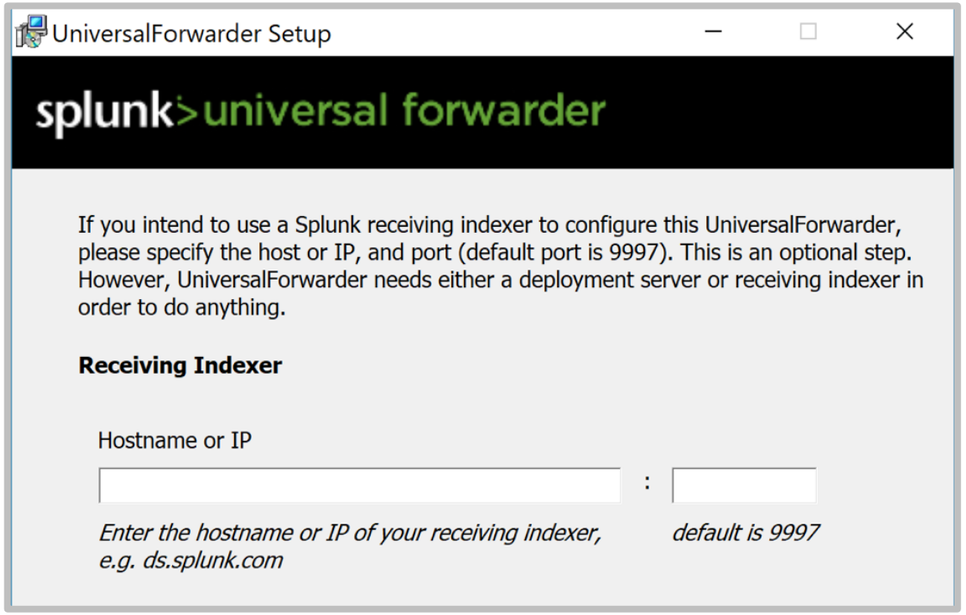
- *NIX: unpack the **`.tgz`** or **`.tgz.Z`** in the desired location
- Windows: execute the **`.msi`** or use the command line
  - Installed as a service
- **`SPLUNK_HOME`** is the installation directory:

  `/opt/splunkforwarder` or
  `c:\Program Files\SplunkUniversalforwarder`
- Same **`splunk`** command-line interface in **`SPLUNK_HOME/bin`**
  - Same commands for start/stop, restart, etc.
  - A password will have to be defined for the **admin** account
- When installing large numbers of forwarders, use an automated method

Interactive Windows UF or Forwarder Installer

- Splunk can run without administrator privileges
  - Splunk must be able to run as a service
- Some forwarder settings can be configured using the installer wizard
  - This is *not* recommended for production, but can be helpful for testing
- CLI installation is available for scripted installations





http://docs.splunk.com/Documentation/Forwarder/latest/Forwarder/InstallaWindowsuniversalforwarderfromthecommandline

Universal Forwarder Configuration Steps

System Admin Task

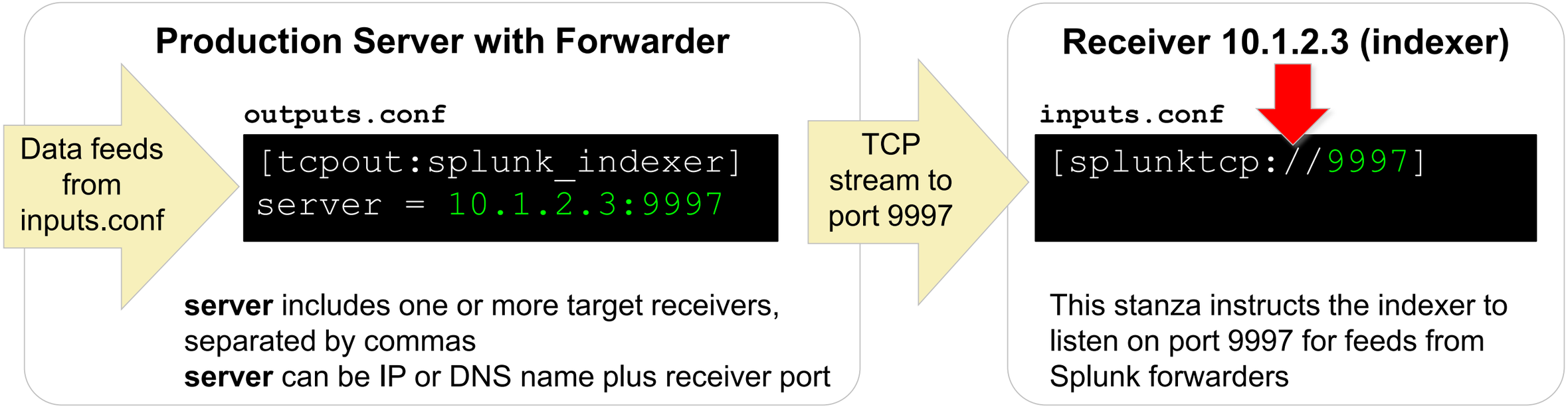
1. Set up a receiving port on each indexer
   - It is only necessary to do this once

Data Admin Task

2. Download and install Universal Forwarder
3. Set up forwarding on each forwarder
4. Add inputs on forwarders, using one of the following:
   - Forwarder management
   - CLI
   - Manually
   - Installing an add-on or app

Forwarder Configuration Files

- Forwarders require **`outputs.conf`**
  - **`outputs.conf`** points the forwarder to the receiver(s)
  - Can specify additional options for load balancing, SSL, compression, alternate indexers, and indexer acknowledgement



Defining Target Indexers on the Forwarder

- Execute on the forwarder:
  `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
```

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

Testing the Connection

- After running **`splunk add forward-server`**, the forwarder should be communicating with the indexer
  - Splunk forwarder logs are automatically sent to the indexer's **_internal** index
- To check for successful connection from the indexer:
  - In the GUI, search **`index=_internal host=`*`forwarder_hostname`***
  - From CLI, run **`splunk display listen`**
- On the forwarder:
  - To view current forwarder to indexer configuration, run **`splunk list forward-server`**
  - To remove the target indexer setting, 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 the 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)
```

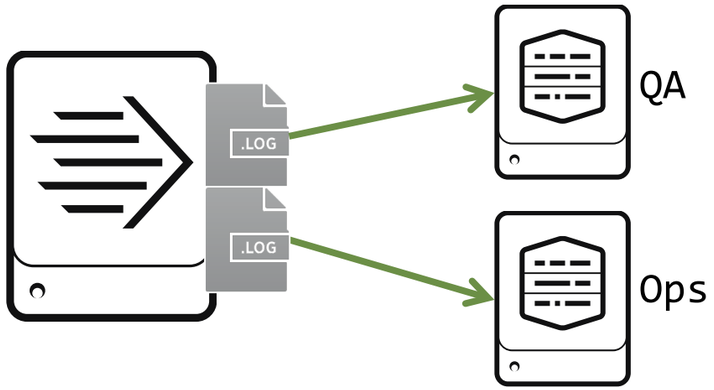
  - To get the **`<uf>`**, run on the forwarder:

```
splunk show default-hostname
```

- Check the configuration files

Selectively Forwarding Data to Indexers

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




```
[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
```

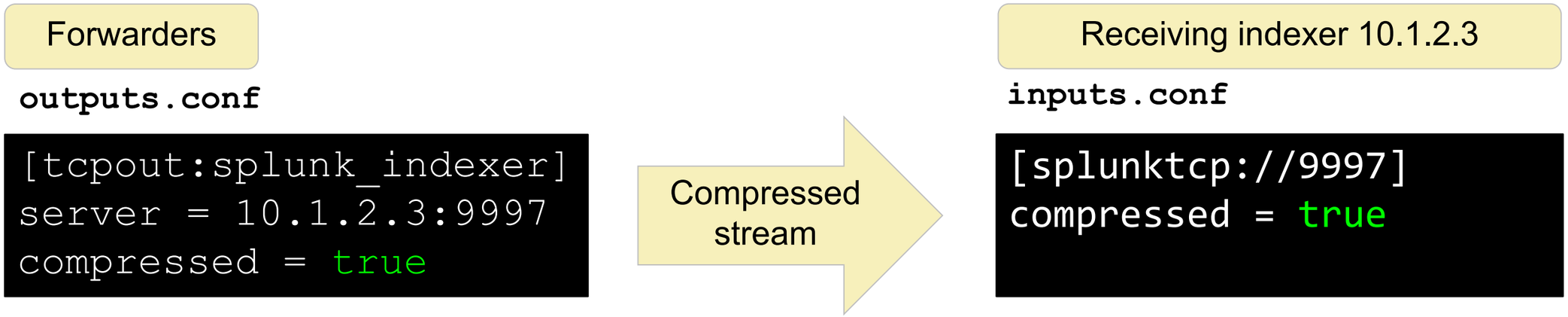
`inputs.conf`

Additional Forwarding Options

- Compressing the feed
- Securing the feed
- Automatic load balancing to multiple indexers
- Forwarder queue size
- Indexer acknowledgement to forwarder

Compressing the Feed

- Slightly increases CPU utilization
- Compression can be set at either the forwarder or the indexer
  - If you want to compress all feeds, set compression on the indexer
  - If you want to compress select feeds, set compression on the forwarder



Securing the Feed – SSL

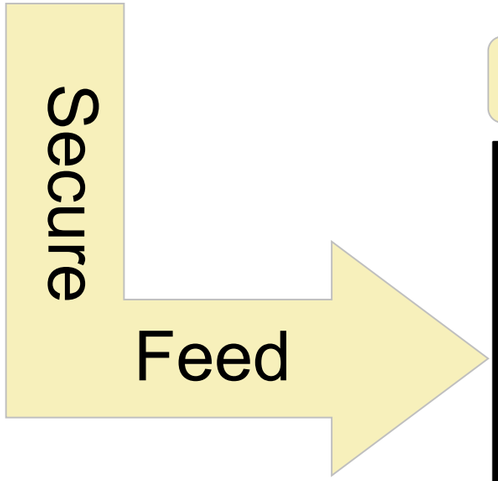
Forwarders

outputs.conf

```
[tcpout:splunk_indexer]
server = 10.1.2.3:9997

sslPassword = ssl_for_m3
sslCertPath = SPLUNK_HOME/etc/auth/cert1/server.pem
sslRootCAPath = SPLUNK_HOME/etc/auth/cert1/cacert.pem
```

Turning on SSL:

- Can increase the CPU usage
- Automatically compresses the feed
- Encrypts password



Receiver 10.1.2.3

inputs.conf

```
[splunktcp-ssl:9997]

[ssl]
password = ssl_for_m3
serverCert = SPLUNK_HOME/etc/auth/cert1/server.pem
rootCA = SPLUNK_HOME/etc/auth/cert1/cacert.pem
```

SSL settings for all inputs in dedicated SSL stanza

Notes About SSL

- Splunk uses OpenSSL to generate its default certificates
  - Default certificate password is **`password`**
- You should use external certs OR create new ones using Splunk's OpenSSL
- docs.splunk.com/Documentation/Splunk/latest/Security/AboutsecuringyourSplunkconfigurationwithSSL
- docs.splunk.com/Documentation/Splunk/latest/Security/Aboutsecuringdatafromforwarders
- wiki.splunk.com/Community:Splunk2Splunk_SSL_DefaultCerts
- wiki.splunk.com/Community:Splunk2Splunk_SSL_SelfSignedCert_NewRootCA

Automatic Load Balancing

- Automatic load balancing switches from server to server in a list
  - Switch happens only when the forwarder detects an EOF
  - Time-based load balancing default frequency is 30 seconds
  - Volume-based load balancing is set on how much data a forwarder sends before switching
- Load balancing is the key to making distributed search or clustering work efficiently

outputs.conf

```
[tcpout:splunk_indexer]
server = splunk1:9997,splunk2:9997,splunk3:9997
```

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

Defining Event Boundary on UF

- Normally, event boundary is determined on the indexer
- The UF does not know when it is safe to switch to the next indexer, unless:
  - An EOF is detected
  - Or, a short break in IO 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
```

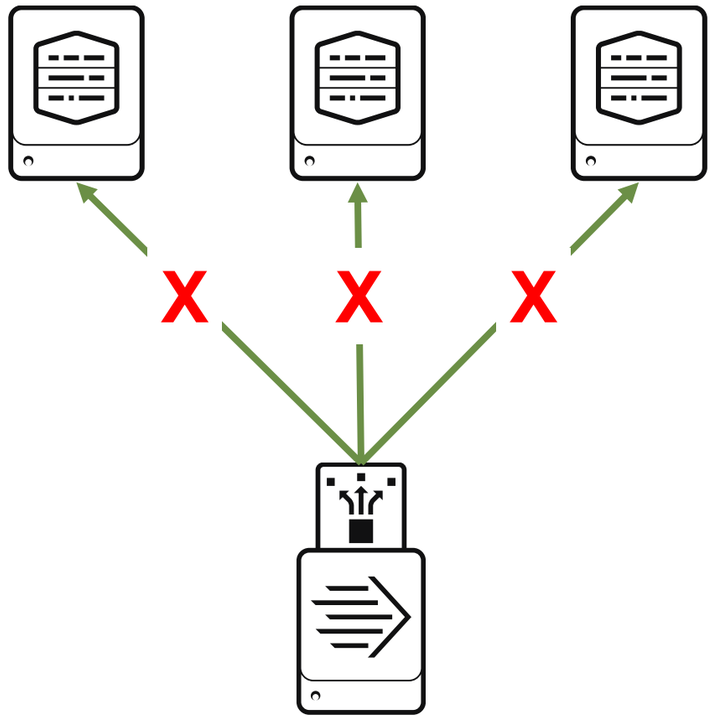
  - Multi-line event

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

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

Caching/Queue Size in `outputs.conf`

- **`maxQueueSize = 500kb`** (default) is the maximum amount of data the forwarder queues if the target receiver cannot be reached
  - In load-balanced situations, if the forwarder can't reach **any** of the indexers, it automatically switches to another and only queues if all indexers are down or unreachable
- See **`outputs.conf.spec`** for details and more queue settings



Indexer Acknowledgement

- Guards against loss of data when forwarding to an indexer
  - Forwarder resends any data not acknowledged as "received" by the indexer
- Disabled by default
- Can also be used for forwarders sending to an intermediate forwarder
- Automatically increases the wait queue to 3x the size of **`maxQueueSize`** to meet larger space requirement for acknowledgement

`outputs.conf`

```
[tcpout:splunk_indexer]
useACK = true
...
```

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

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

  http://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.

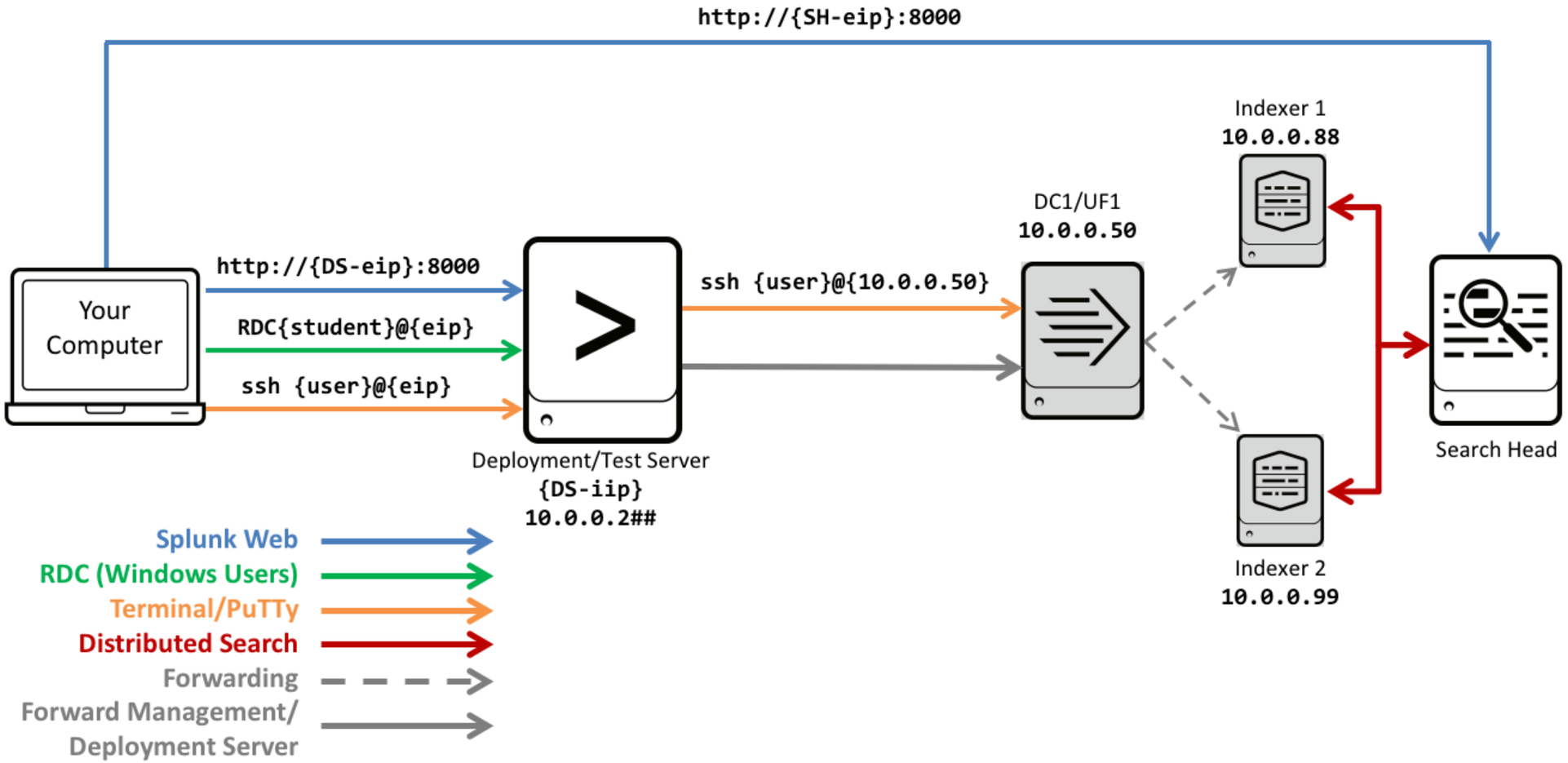
- True or False. Turning SSL on between the forwarder and the receiver automatically compresses the feed.

  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:

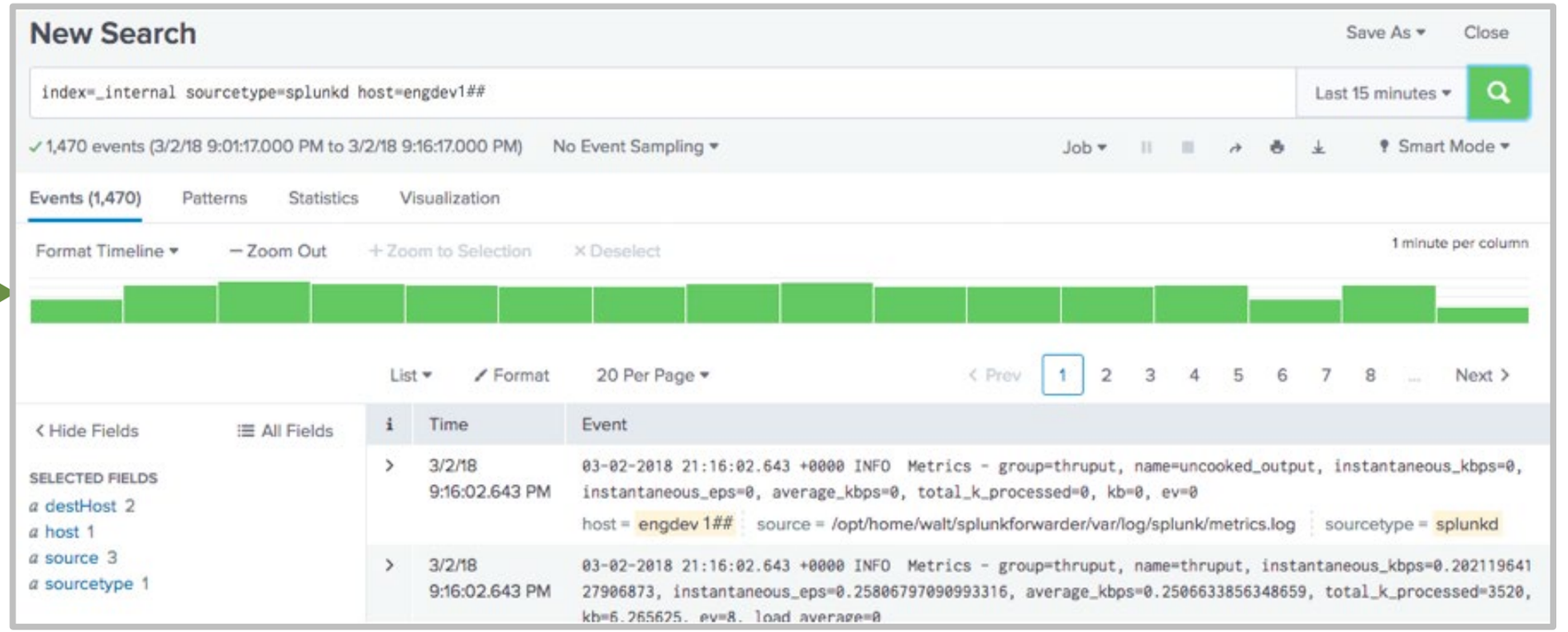
- You have a login on a remote Linux host that is your forwarder
- This lab exercise only establishes the connection between your UF and Indexer

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

Verification:

1. Run a search to get the forwarded internal logs from UF#1
   `index=_internal sourcetype=splunkd host=eng1##`

`http://{EIP}:8000`



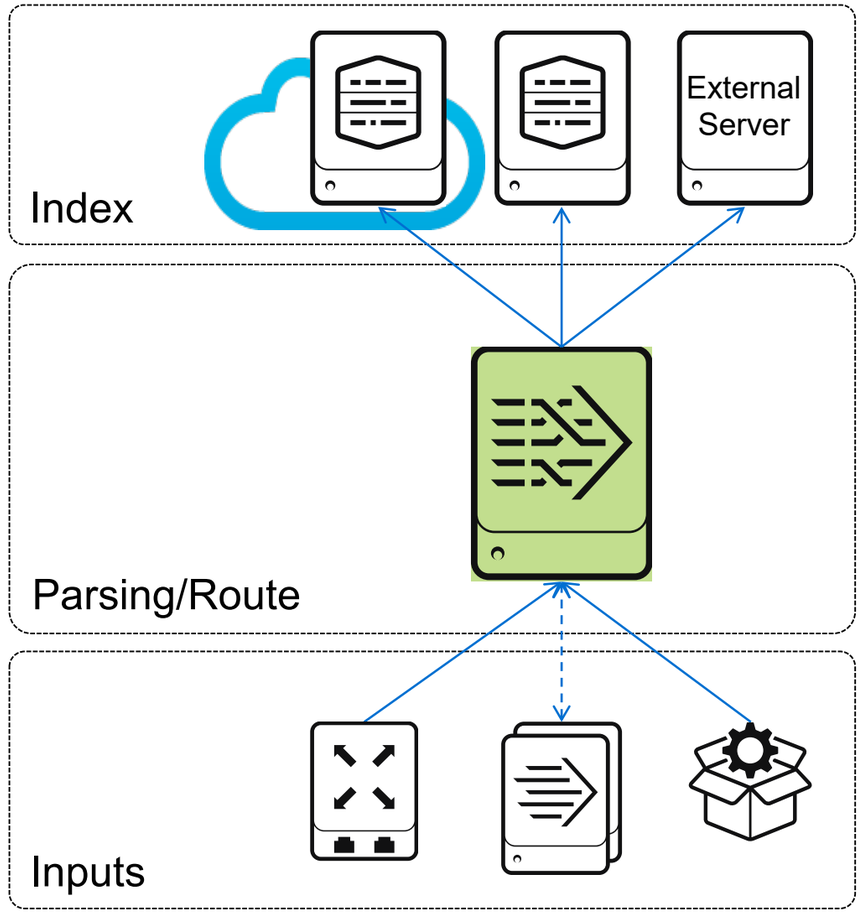
Module 4: Heavy Forwarders & Forwarder Management

Module Objectives

- Introduction to the Heavy Forwarder (HF)
  - Describe what the heavy forwarder is and use cases
  - Heavy forwarder configuration
  - Connect to the deployment server and 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

Heavy Forwarder (HF)

- Splunk Enterprise
  - Set the license group to **Forwarder License**
  - Initially, has full functionality; input, indexing, parsing, searching
  - Configure only the required processes
- Accepts all input types and can parse raw data
- Can route data to different indexers or 3rd party receivers
- Can be used as an intermediary receiver for one or more universal forwarders
  - As a mid-tier component in a multi-stage data routing design
  - To parse and route data to the indexers



Use Cases for the HF

- When a GUI is needed (remember UF does not have a GUI)
- Advanced event level routing to separate indexers or index clusters
- Anonymizing or masking of incoming data before forwarding to an indexer
- Predictable version of Python is needed
- No access to indexers
- Required by an App/Input
  - HEC, DBX

HF Configuration – Inputs

- The HF can receive data from other Splunk instances, such as UF
  - Setup can be done manually using CLI:

    ```
    ./splunk enable listen <port>
    ```

  - You can also deploy 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 – Client

- To configure the HF as a deployment client to the DS:

```
./splunk set deploy-poll <DS-IP/Hostname:port>
```

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

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


HF Configuration – Outputs

- To configure the HF to forward the data to the indexers, you can use either of the following methods:
  - Set up forwarding manually using CLI:

    `./splunk add forward-server:<Index-IP/Hostname:<listening_port>`
  - Deploy an **`outputs.conf`** file from the DS with the following entry:

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

[tcpout-server://Index-IP/Hostname:<Listening_Port>]

[tcpout:default-autolb-group]
disabled = false
server = Index-IP/Hostname:<Listening_Port>
```

Note

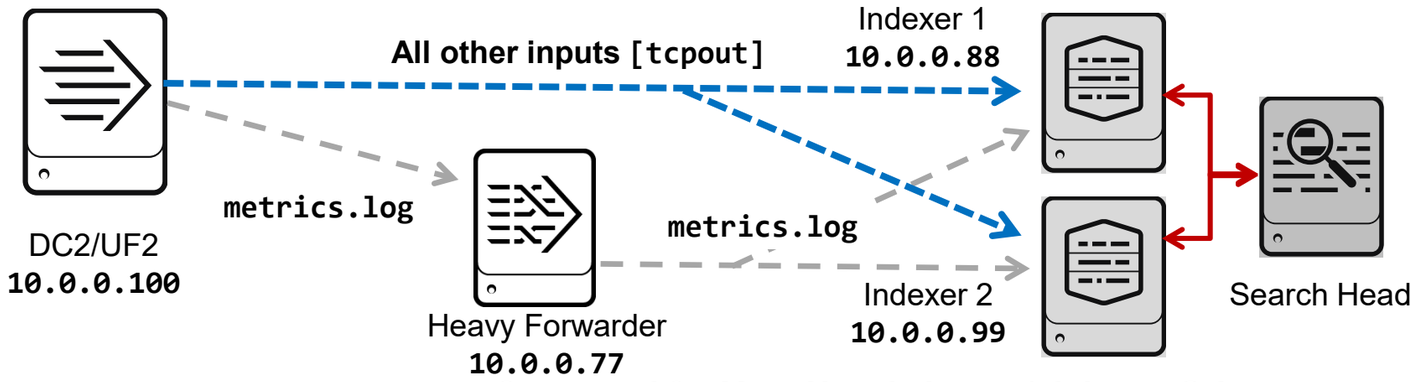
In the lab exercises, you will configure forwarding using the DS.

Selectively Routing Data

- You can selectively send data from the UF directly to the indexers or route the data through the HF to parse the data before sending it to the indexers

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.



inputs.conf on the **UF**

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

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

outputs.conf on the **UF**

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

[tcpout]
defaultGroup = default-autolb-group

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


 | 23 Aug 2019

HF Configuration – Outputs

- Based on your use case, you may not want to index any data on the HF:

outputs.conf

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

- Based on your use case, you want to disable Splunk Web

web.conf

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


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.

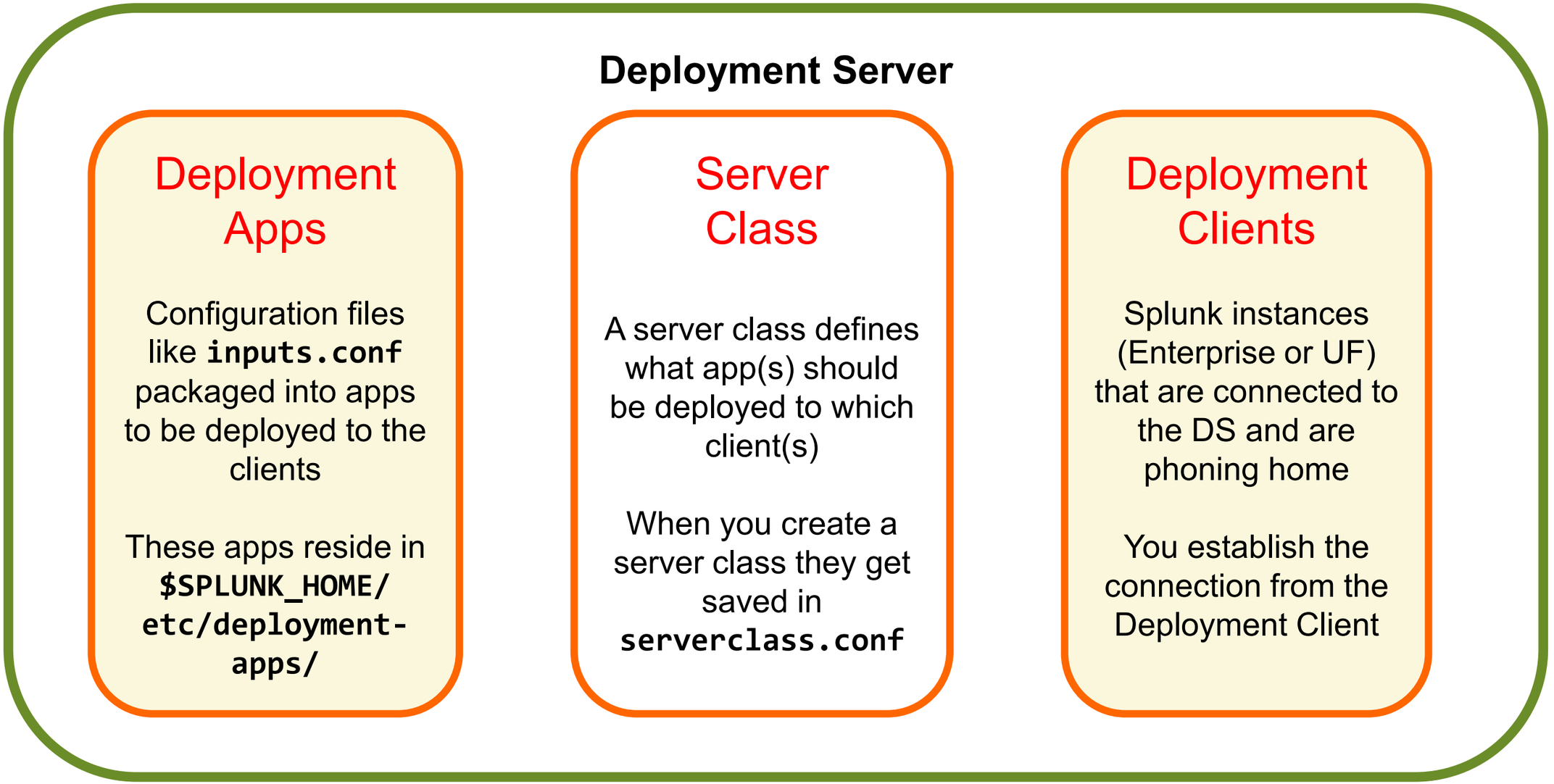
  False. 8089 is the default management port. The listening port can be any port.

Deployment Management

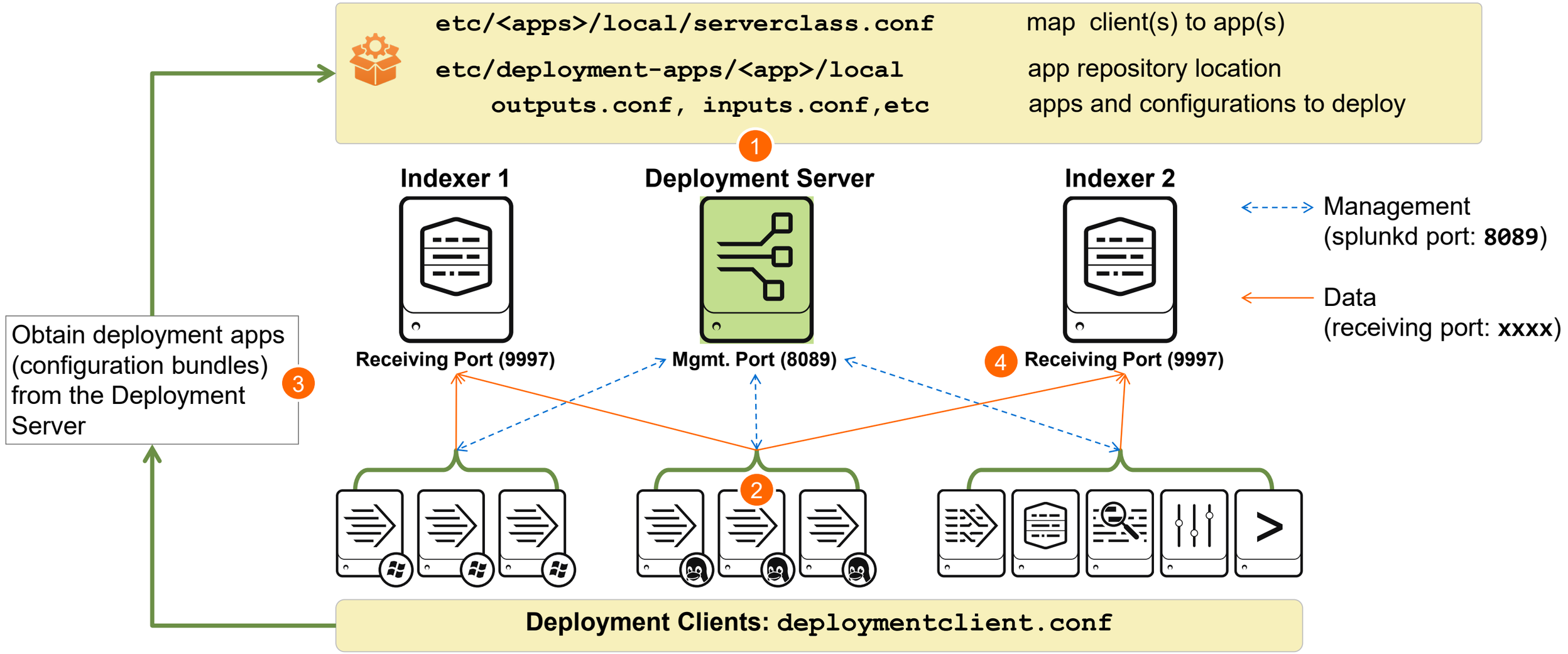
- **Deployment Server** is a built-in tool for managing configuration of Splunk instances
  - Allows you to manage remote Splunk instances centrally
  - Requires an Enterprise License
  - Handles the job of sending configurations (**`inputs.conf`**, **`outputs.conf`**, etc.) packaged as apps
  - Can automatically restart remote Splunk instances
- **Forwarder management** is a graphical interface on top of deployment server
- **Monitoring Console Forwarder dashboards** help you monitor the deployment
- **Best Practice:** The Deployment Server should be a dedicated Splunk instance
  - In this class, you will use your test server as a deployment server

http://docs.splunk.com/Documentation/Splunk/latest/Updating/Calculatedeploymentserverperformance

Deployment Server Components

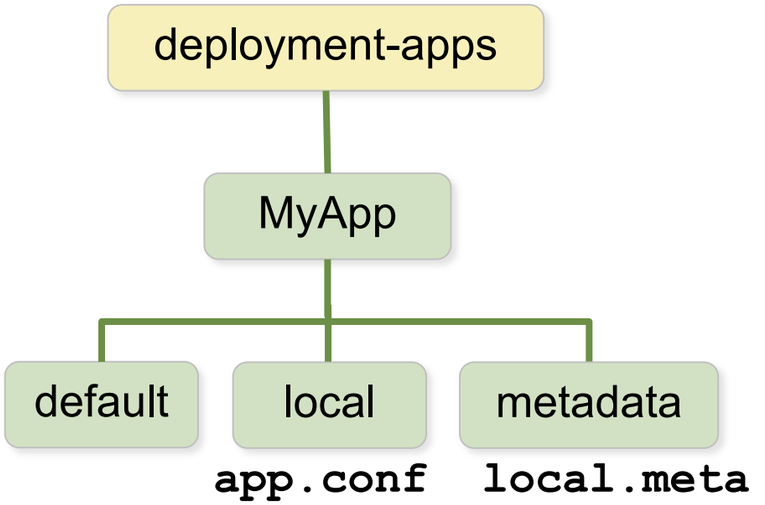


Ports and Configurations



What's in a Deployment App?

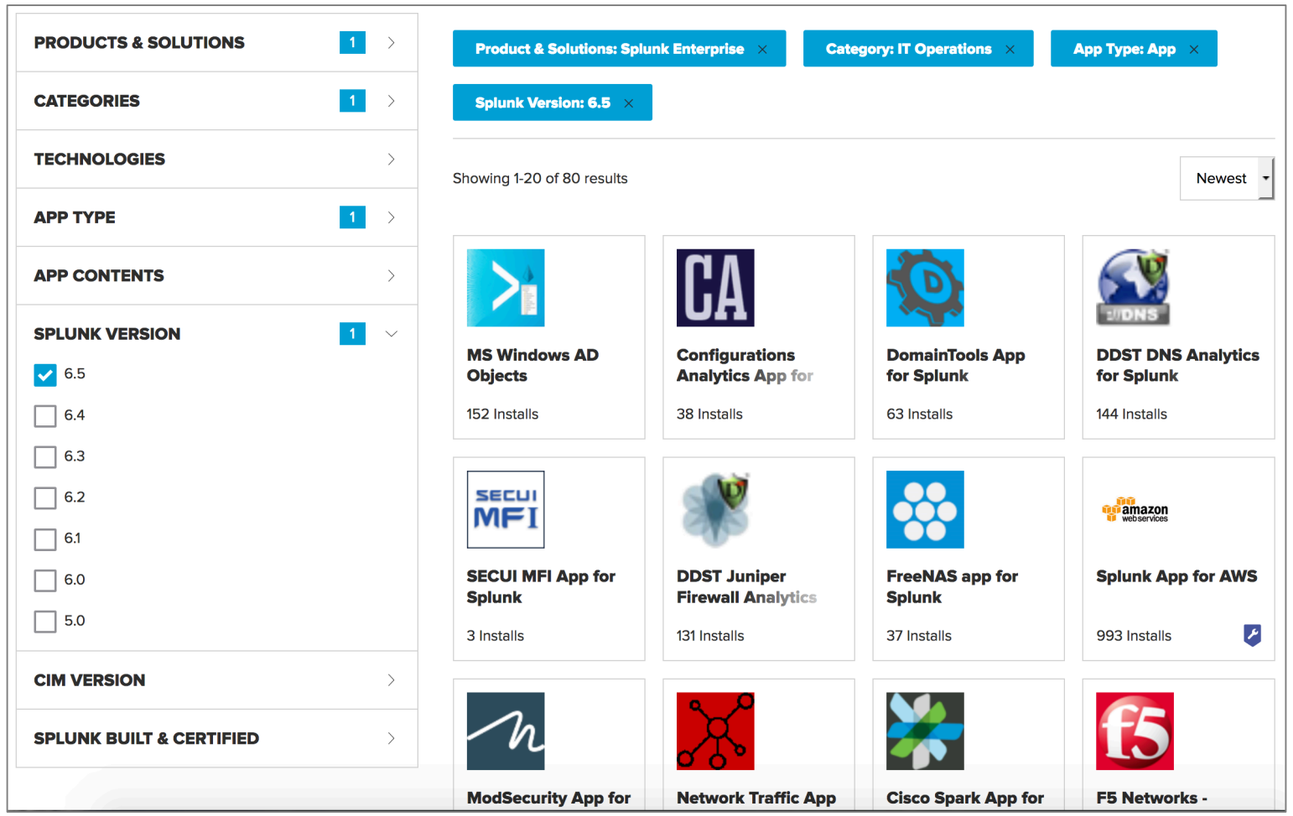
- Deployment Server/Forwarder Management works by deploying one or more apps from the **`SPLUNK_HOME/etc/deployment-apps`** folder to the remote forwarders (clients)
  - They are deployed to the forwarder's **`SPLUNK_HOME/etc/apps`** folder by default
- An app can have configuration files, scripts, and other resources
  - Apps must follow the normal app structure and rules. Required files:
    - **`app.conf`** (in **`default`** or **`local`**)
    - **`local.meta`** (in **`metadata`**)
- Best practice
  - Create small and discrete deployment apps
  - Take advantage of .conf file layering
  - Use a naming convention



Add-ons

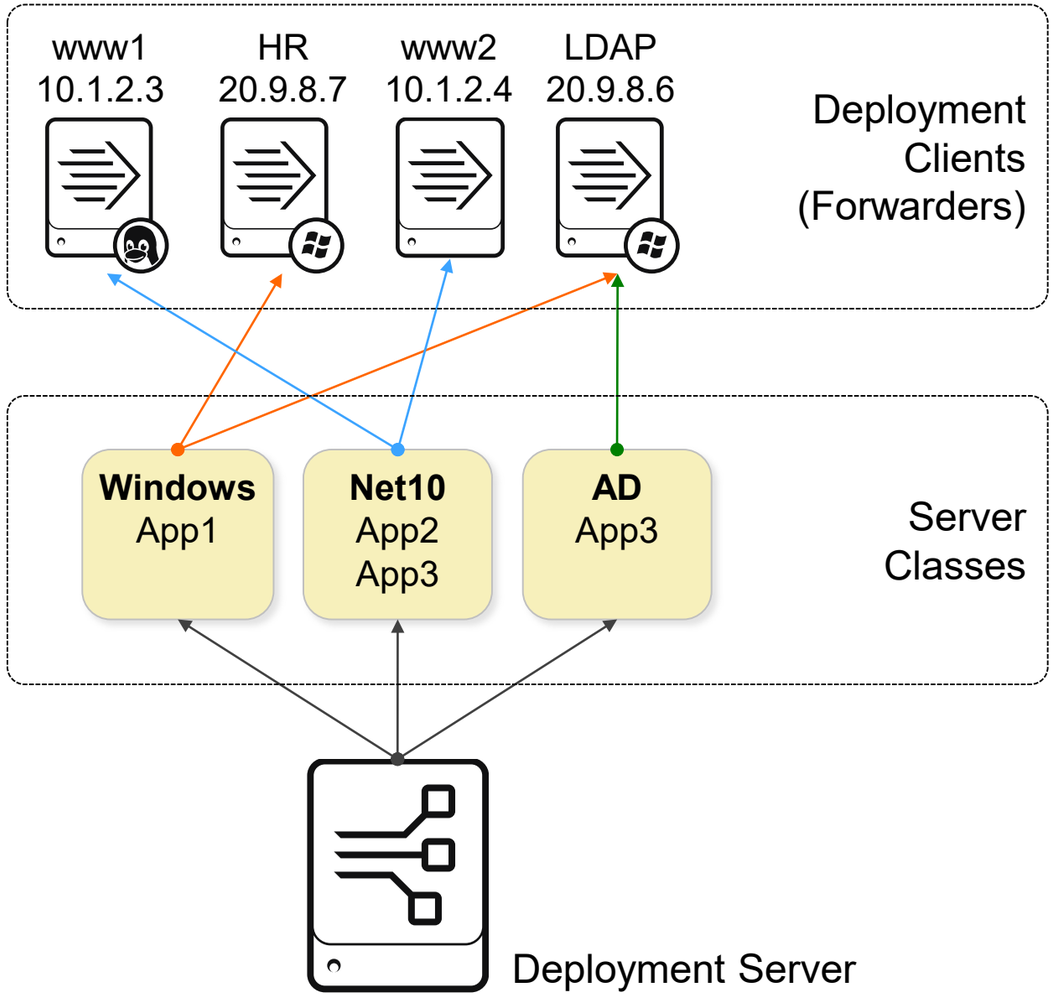
- Apps and add-ons can be downloaded from Splunkbase
- Installed on a forwarder:
  - Using the Deployment Server (deploys the app to **`SPLUNK_HOME/etc/apps`**)
  - Using CLI on the forwarder
  - Manually installing the app

    Note: Splunk Web does not exist on a universal forwarder!
- The app must be installed in **`SPLUNK_HOME/etc/apps`**
- See the add-on's documentation for details about its settings for **`inputs.conf`**



What's a Server Class?

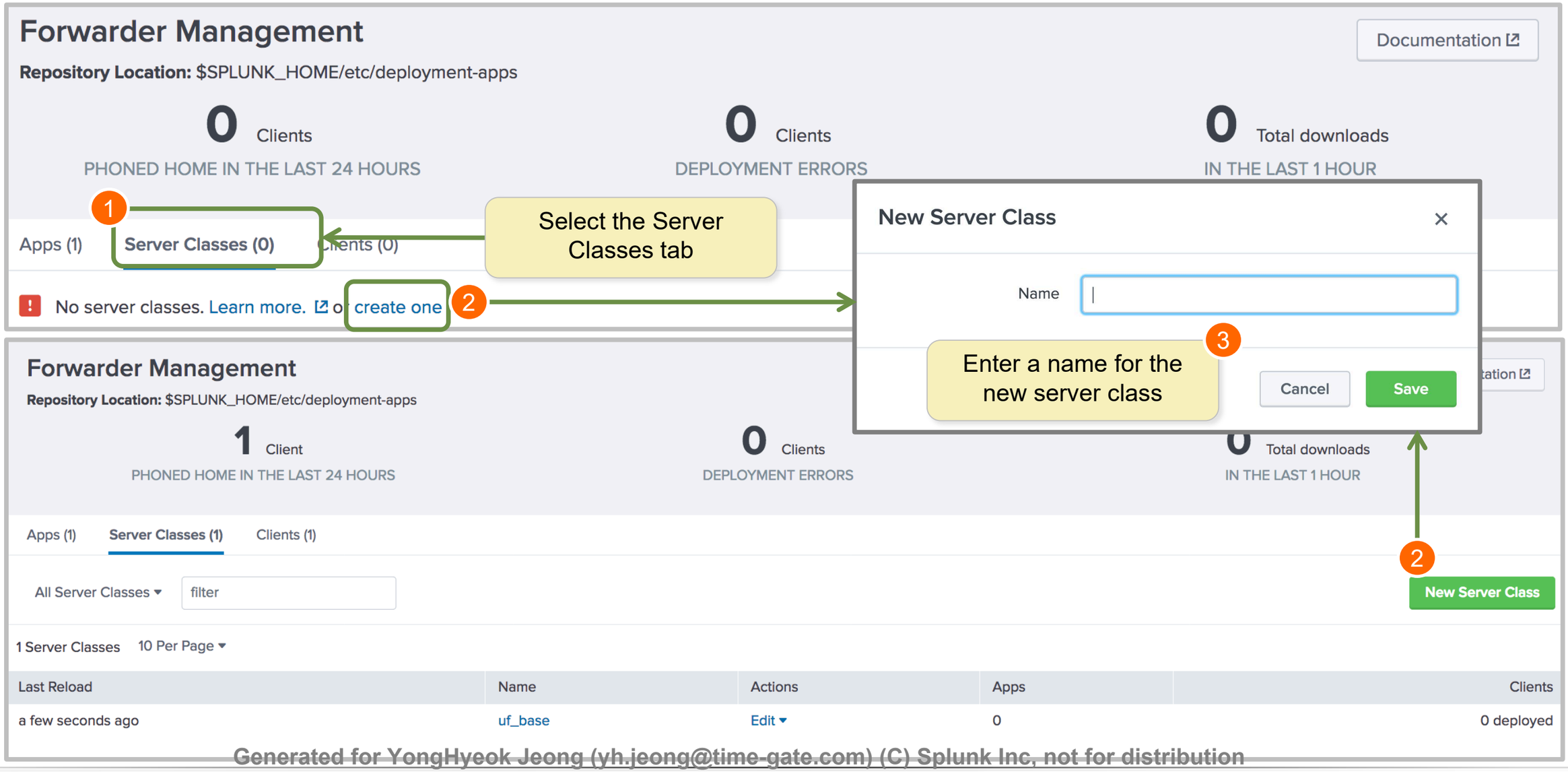
- A server class maps a group of clients to one or more deployment apps
- A set of clients can be grouped based on:
  - Client name, host name, IP address, or DNS name
  - Machine types
- Examples:
  - **Windows** server class
    - Systems running Windows get **App1**
  - **Net10** server class
    - Hosts on 10.1.2.* subnet get **App2** and **App3**
  - **AD** server class
    - AD servers get **App3**
- Notice that clients (like the LDAP server) can belong to multiple server classes



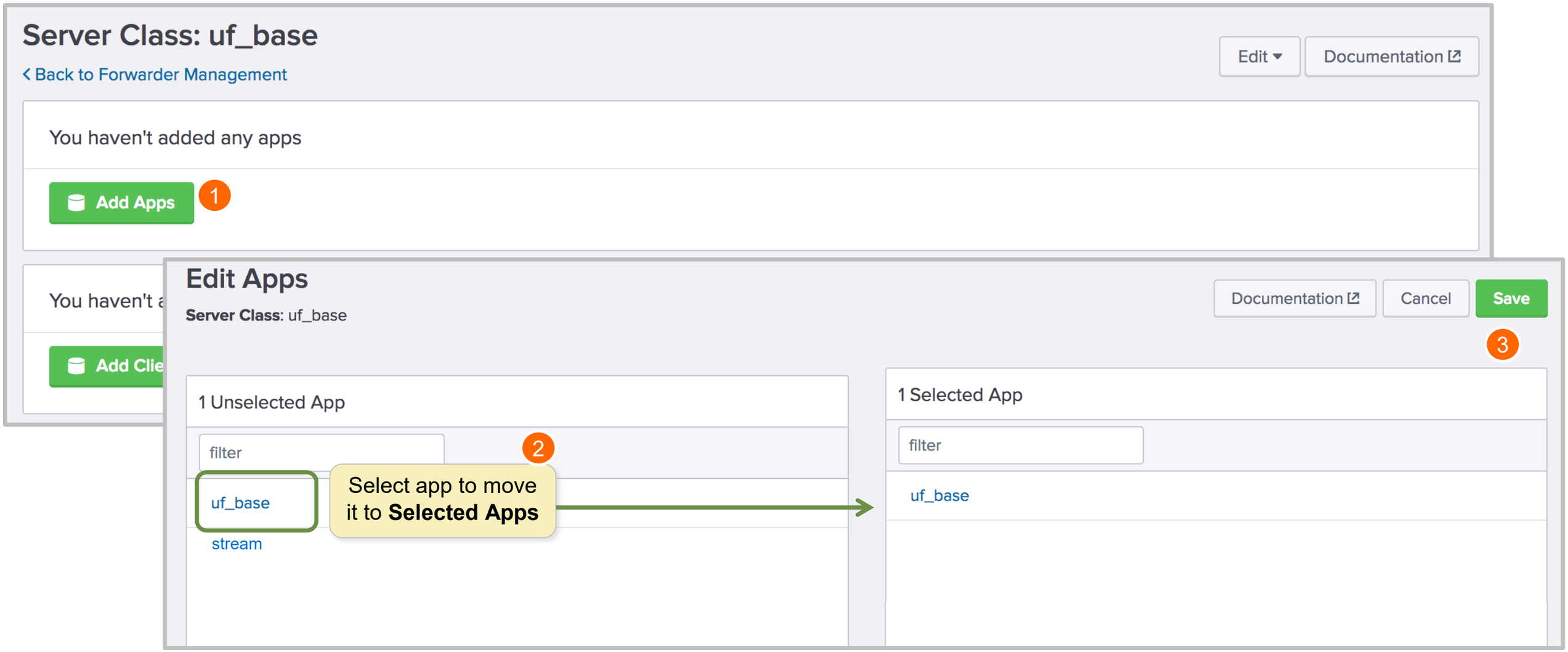
Enabling Forwarder Management

- Overview of how to set up Forwarder Management in your implementation:
  1. On the deployment server, add one or more apps in **`SPLUNK_HOME/etc/deployment-apps`**
  2. In the Forwarder Management UI, create one or more server classes
  3. On forwarders, run **`splunk set deploy-poll <deployServer:port>`**
     - Where **`port`** is the **`splunkd`** port on the deployment server - **`8089`** is the default
  4. Verify on deployment server:
     - List of clients phoning home
     - Deployment status
  5. Verify on forwarders:
     - **`etc/apps`** folder for deployed apps

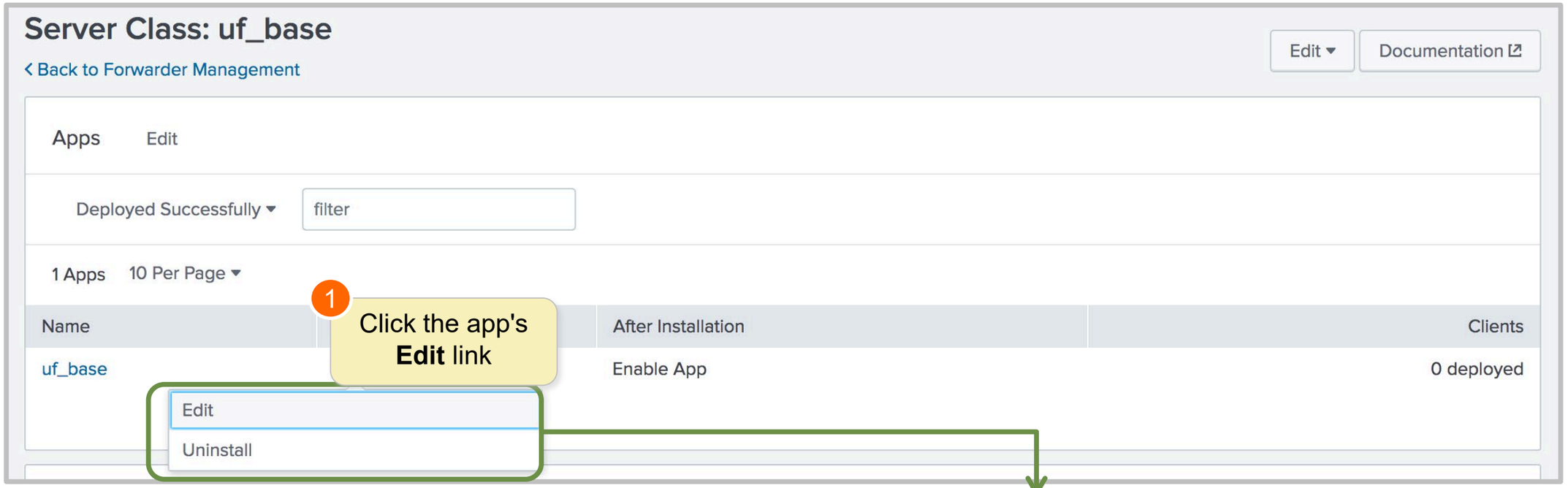
Adding a Server Class

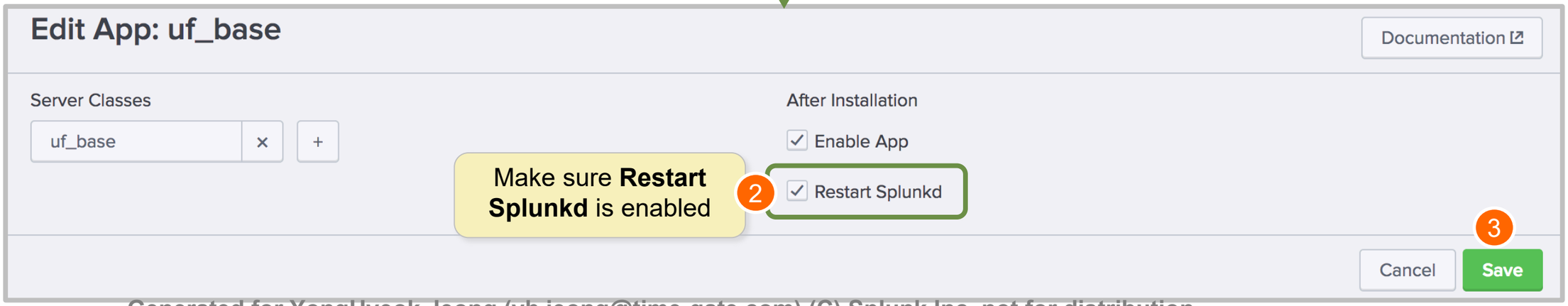


Selecting Apps for the Server Class

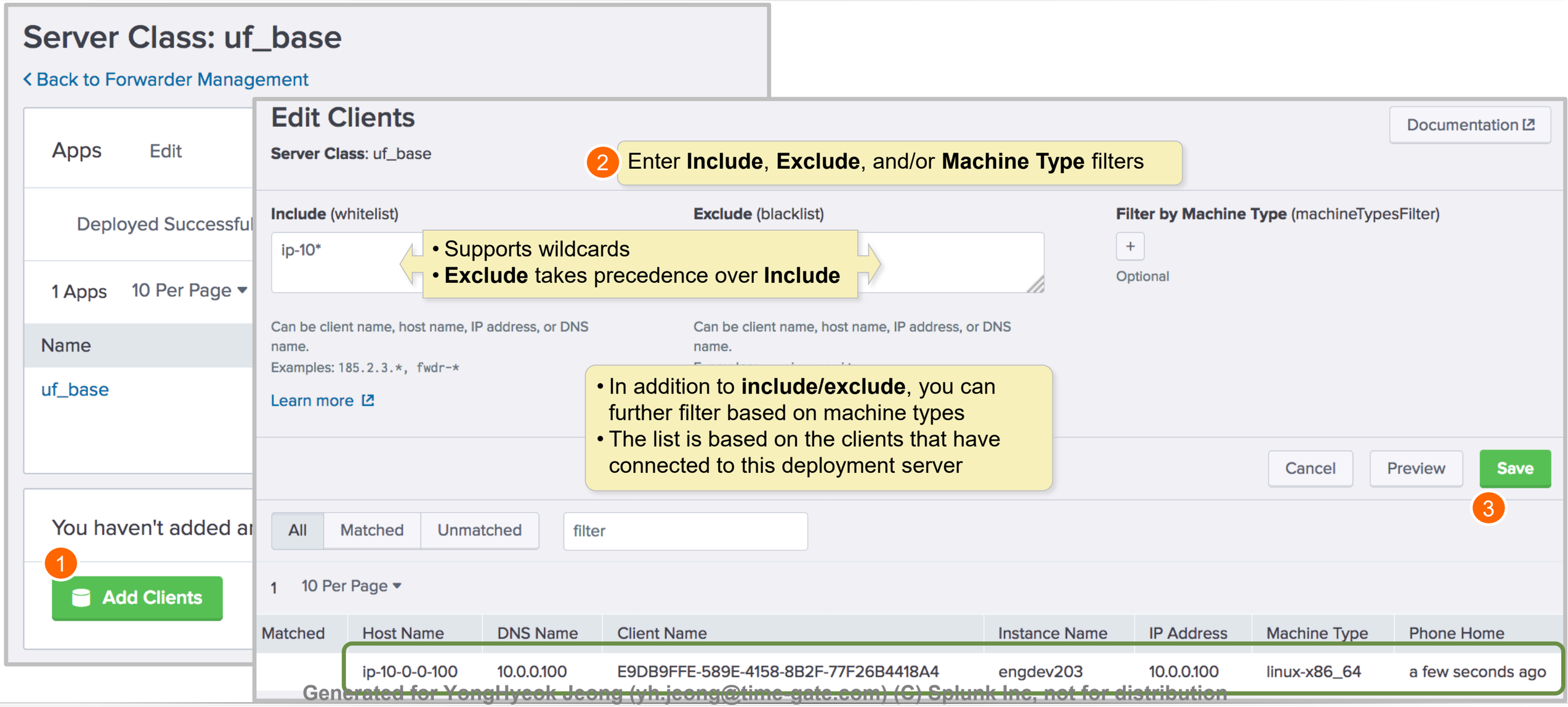


Post Deployment Behavior Setting





Selecting Clients for the Server Class



Configuring Deployment Clients

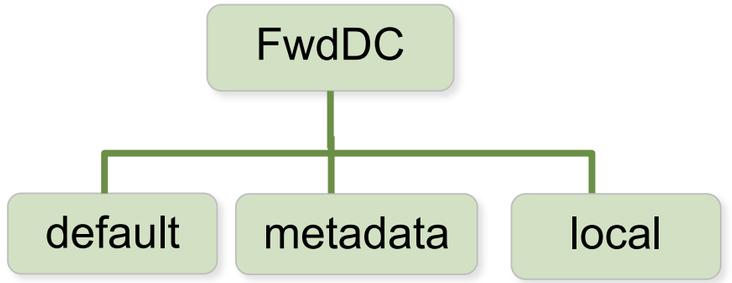
- Configure your forwarders to be deployment clients
  - Run this during forwarder installation or later
    `splunk set deploy-poll deployServer:port`
    - **`deployServer`** = deployment server hostname or IP
    - **`port`** = splunkd port
    - Creates **`deploymentclient.conf`** in **`SPLUNK_HOME/etc/system/local`**
    - OR, create deploymentclient.conf manually
  - Create an app and place the conf file in the local directory
  - Restart the deployment clients:
    - **`splunk restart`**
- To override the default attributes, edit the **`[deployment-client]`** stanza
  - Can be part of initial deployment app
  - The forwarder "phones home" once a minute (by default)

`deploymentclient.conf`

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

`deploymentclient.conf`

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

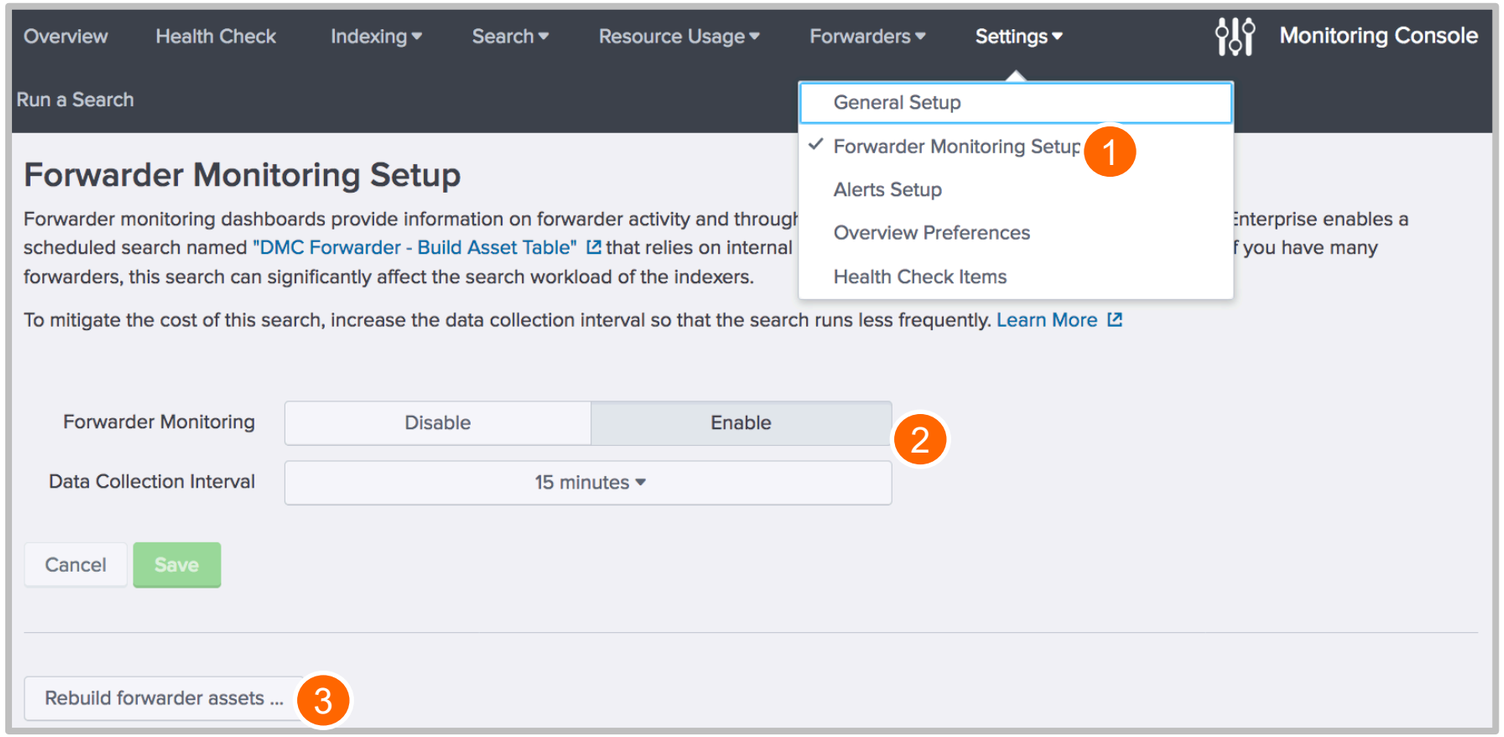


Configuring Deployment Clients (cont.)

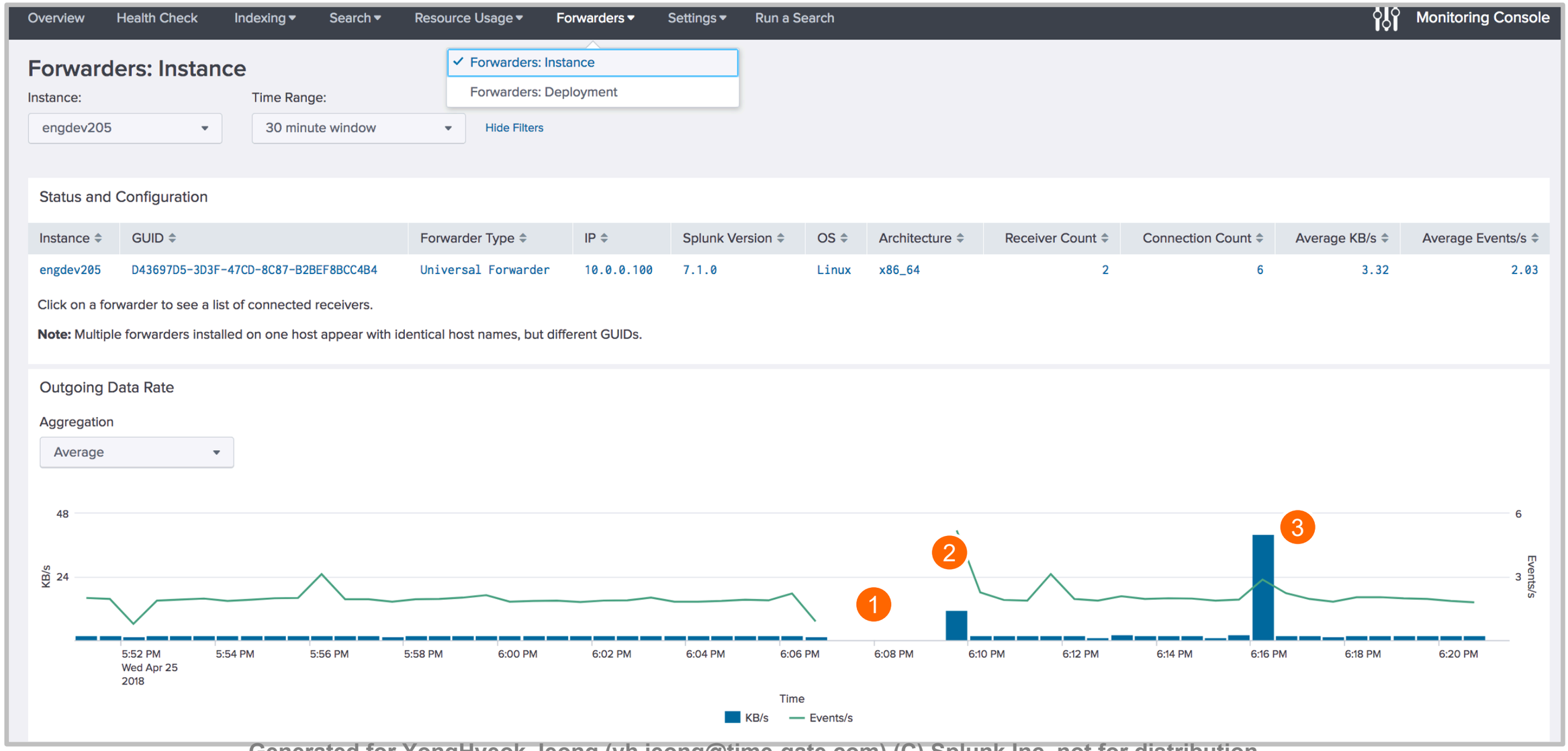
- On the deployment client (usually a forwarder)
  - Confirm that the expected app directories and contents have arrived in **`SPLUNK_HOME/etc/apps`**
- If the app is changed on the Deployment Server, then the forwarder will load the updated app after its next phone-home
  - To change the app settings 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

- The Monitoring Console (MC) provides valuable information on forwarder activity and throughput
  - Once enabled, a scheduled search runs to build a forwarder asset table
  - Runs every 15 minutes by default
  - Relies on the internal logs forwarded by forwarders
  - Can affect the search workload if you have many forwarders
  - Can rebuild 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 DS	
`./splunk reload deploy-server`	Checks all apps for changes and notifies the relevant clients
`./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.

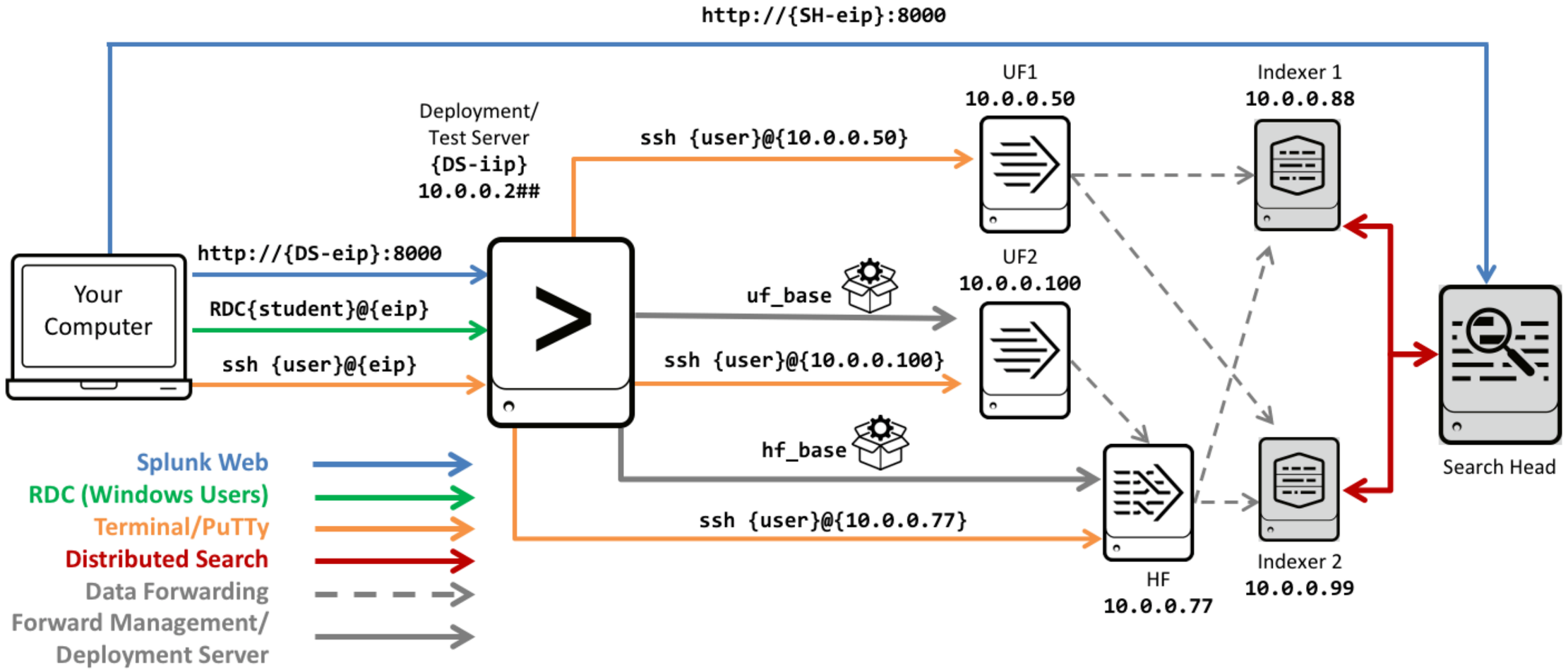
- When an app is deployed from the Deployment Server to the client, where will you find that app on the client by default?

  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`** app 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 the listening port on HF to listen 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 the deployment of the **`hf_base`** on the 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 Files

- A monitor input defines a specific file as the data source
  - The current content of the file is ingested
  - The file is continuously monitored for new content
  - Splunk tracks file status and automatically continues monitoring at the correct file location after a restart
- The file monitor supports any text file format, such as:
  - Plain text data files
  - Structured text files, such as CSV, XML, JSON
  - Multi-line logs, such as Log4J
  - Splunk can also read files compressed with gzip

Monitoring Directories

- A monitor input can define a directory tree as the data source
  - Splunk recursively traverses through the directory structure
  - All discovered text files are consumed, including compressed files
    - Unzips compressed files automatically before ingesting them, one at a time
  - Any files added to the directory tree in the future are included
    - Automatically detects and handles log file rotation
- The input settings are applied to all files in the directory
  - **`sourcetype`**, **`host`** and **`index`** -- if specified -- are applied to all files in the tree
  - **`source=`** the file name (absolute path)
  - Automatic sourcetyping is recommended for directories that contain mixed file types
    - Can override exceptions manually
    - Automatic sourcetyping is disabled if the sourcetype attribute is defined

Monitor Input Options in **inputs.conf**

- Source (after **`monitor://`** in stanza header) is an absolute path to a file or directory
  - Can contain a wildcard
- All attributes (sourcetype, host, index, etc.) are optional
- Defaults apply if omitted
  - Default host is defined in **`etc/system/local/inputs.conf`**
  - Default source is the fully-qualified file name
  - Default sourcetype is automatic
- There are many possible attributes
  - See **`inputs.conf.spec`** in **`SPLUNK_HOME/etc/system/README`**

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

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

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

[monitor://C:\logs\]

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

File Pathname Wildcards

Monitor stanzas in **`inputs.conf`** support two wildcards to help you specify the files/directories you want to index

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

- Whitelist and Blacklist
  - Regular expressions to filter files or directories from the input
  - In case of a conflict between a whitelist and a blacklist, the blacklist prevails
- Follow tail (**`followTail`**)
  - Splunk ignores existing content in the file, but indexes new data as it arrives
  - DO NOT leave **`followTail`** enabled indefinitely
- Consider using **`ignoreOlderThan`**, if applicable
  - A file whose modtime falls outside this time window will not be indexed
    - After a file is ignored, it will never be considered as an input again, even if it is updated
  - **`ignoreOlderThan = 60d`**

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

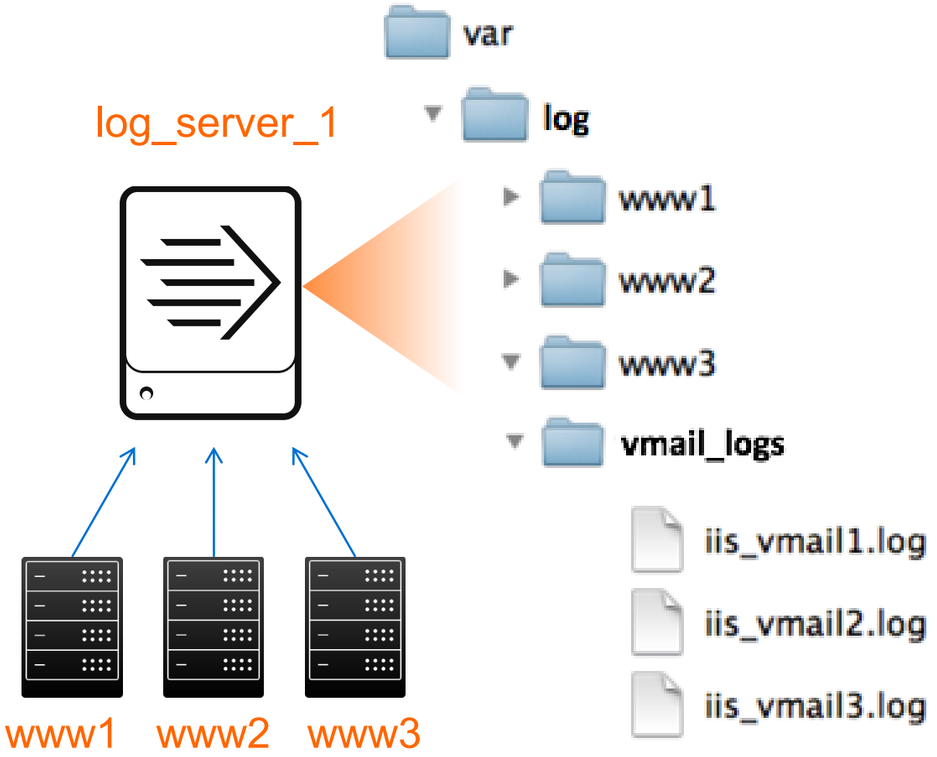
```
[monitor:///var/log/www1/]
whitelist = /query\.log$|/my\.log$
```

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

✓ Matches
✗ Doesn't match

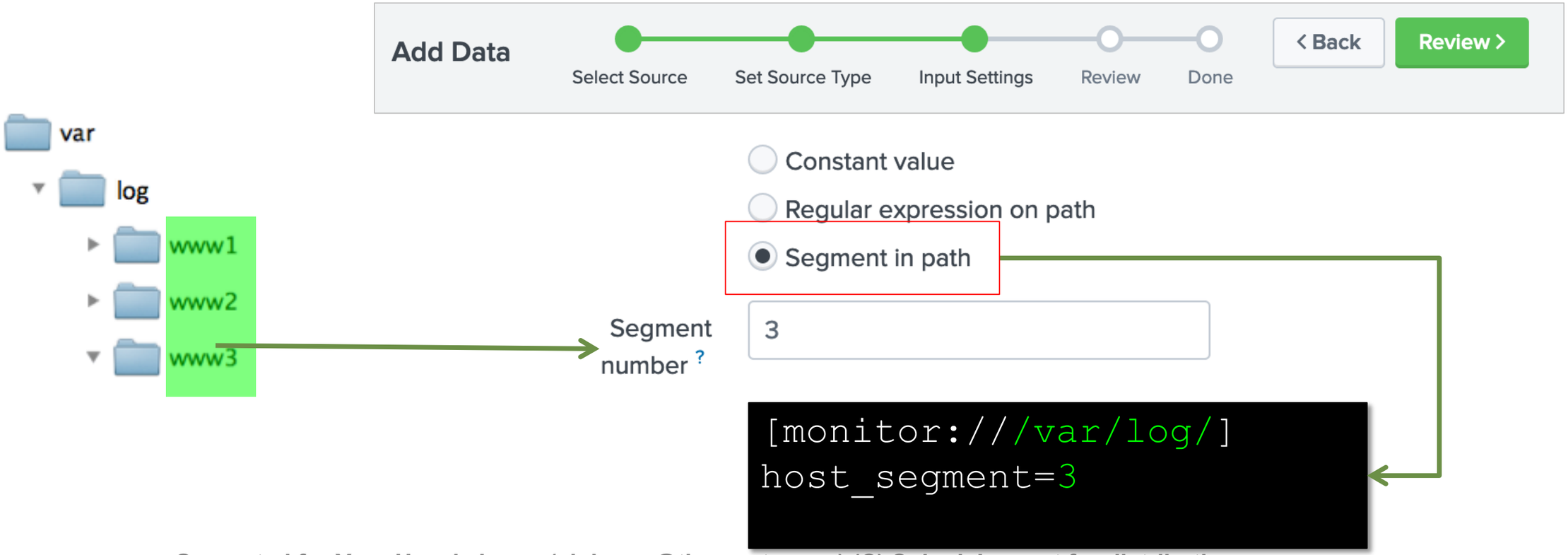
Overriding the Host Field

- Normally on a forwarder, the host can be left to its default value
- In some cases, the data might be stored on a different server than its origin
  - For example, a web farm where each web server stores its log file on a centralized file server
- You can override the default host value
  - Explicitly set the host value
  - Set the host based on a directory name
  - Set the host based on a regular expression



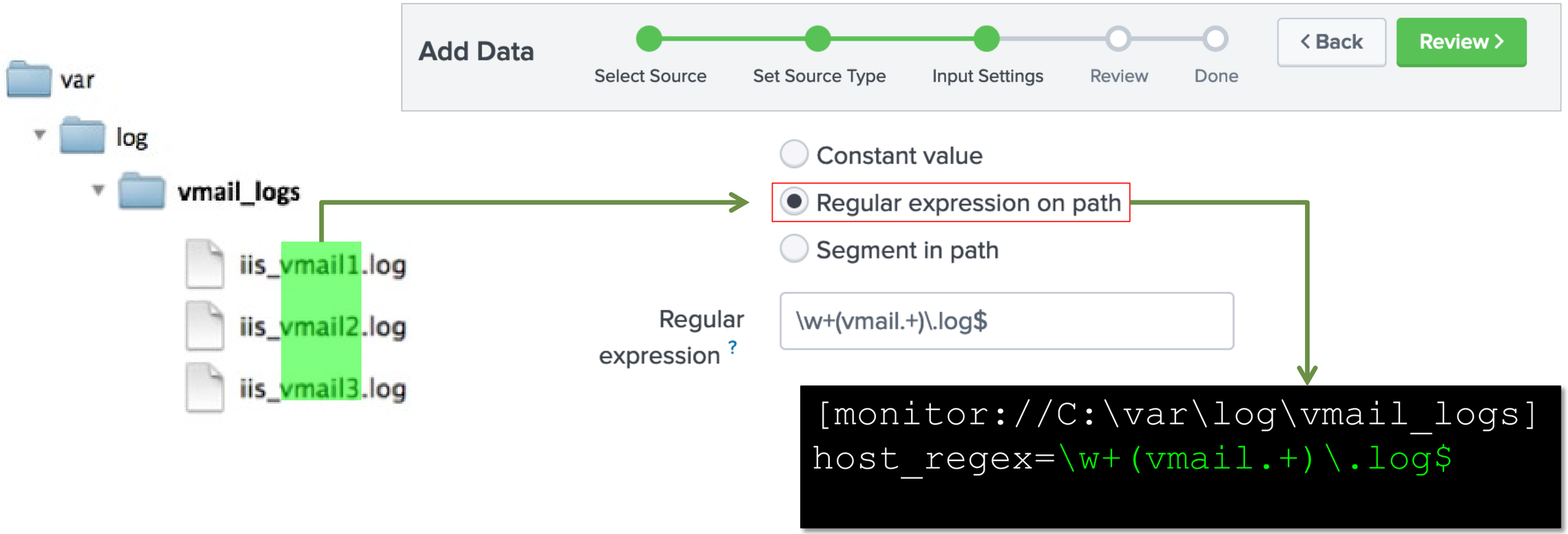
Setting the Host: `host_segment`

- **`host_segment`** = <integer>
  - Setting **`host_segment`** to **3** uses the 3rd segment of the directory path as the host name for files in that directory



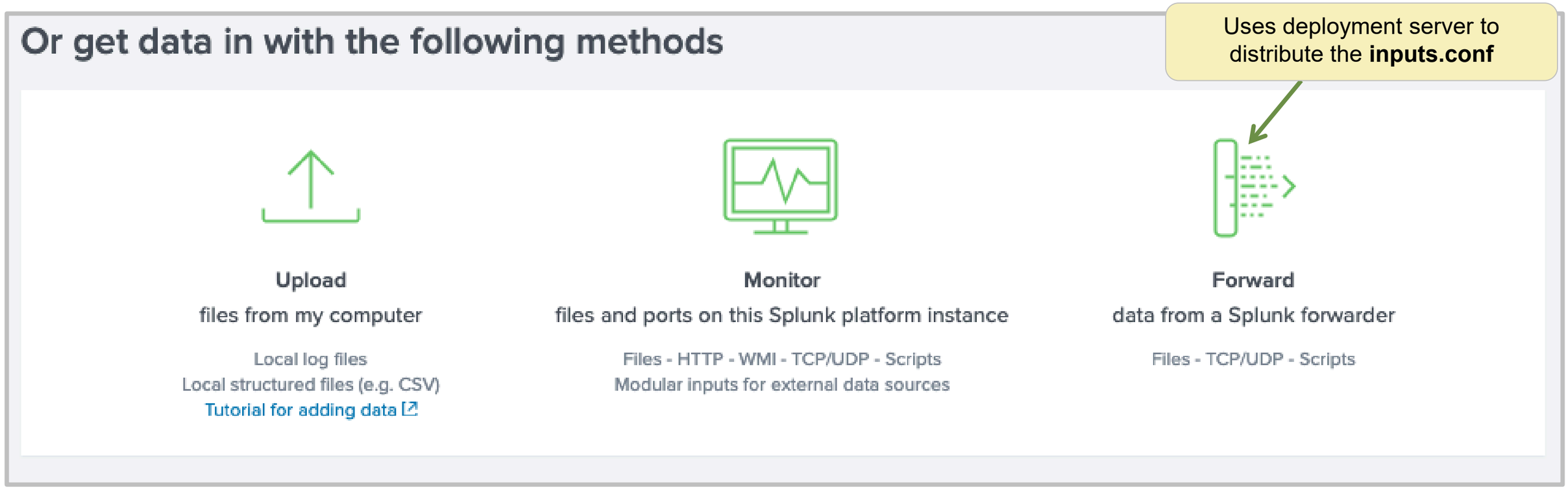
Overriding the Host: host_regex

- **host_regex =** <regular expression>
  - Setting **host_regex** to **\w+(vmail.+)\.log$** selects the second part of the log file name as its host name

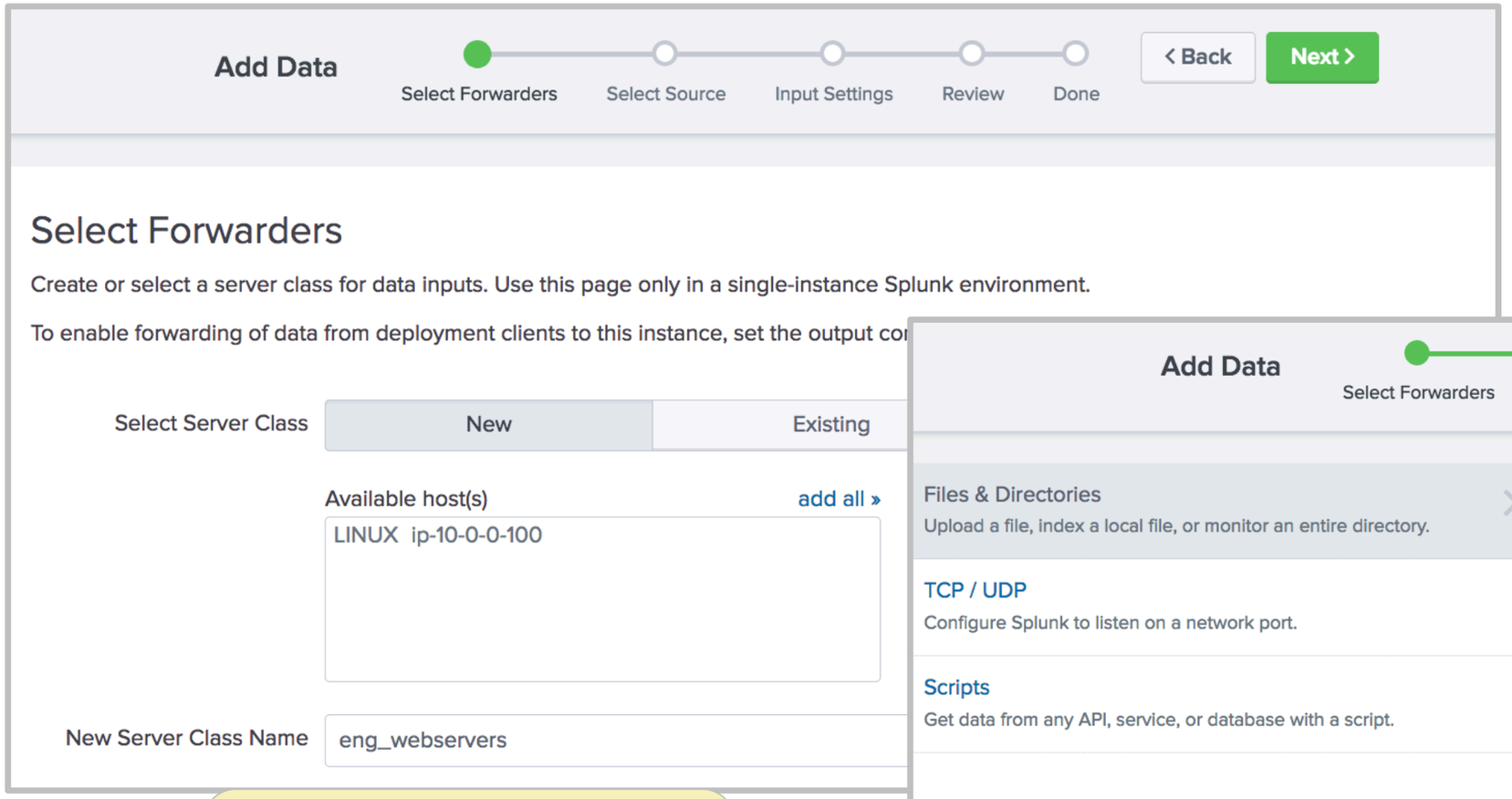


Creating a Remote Data Input

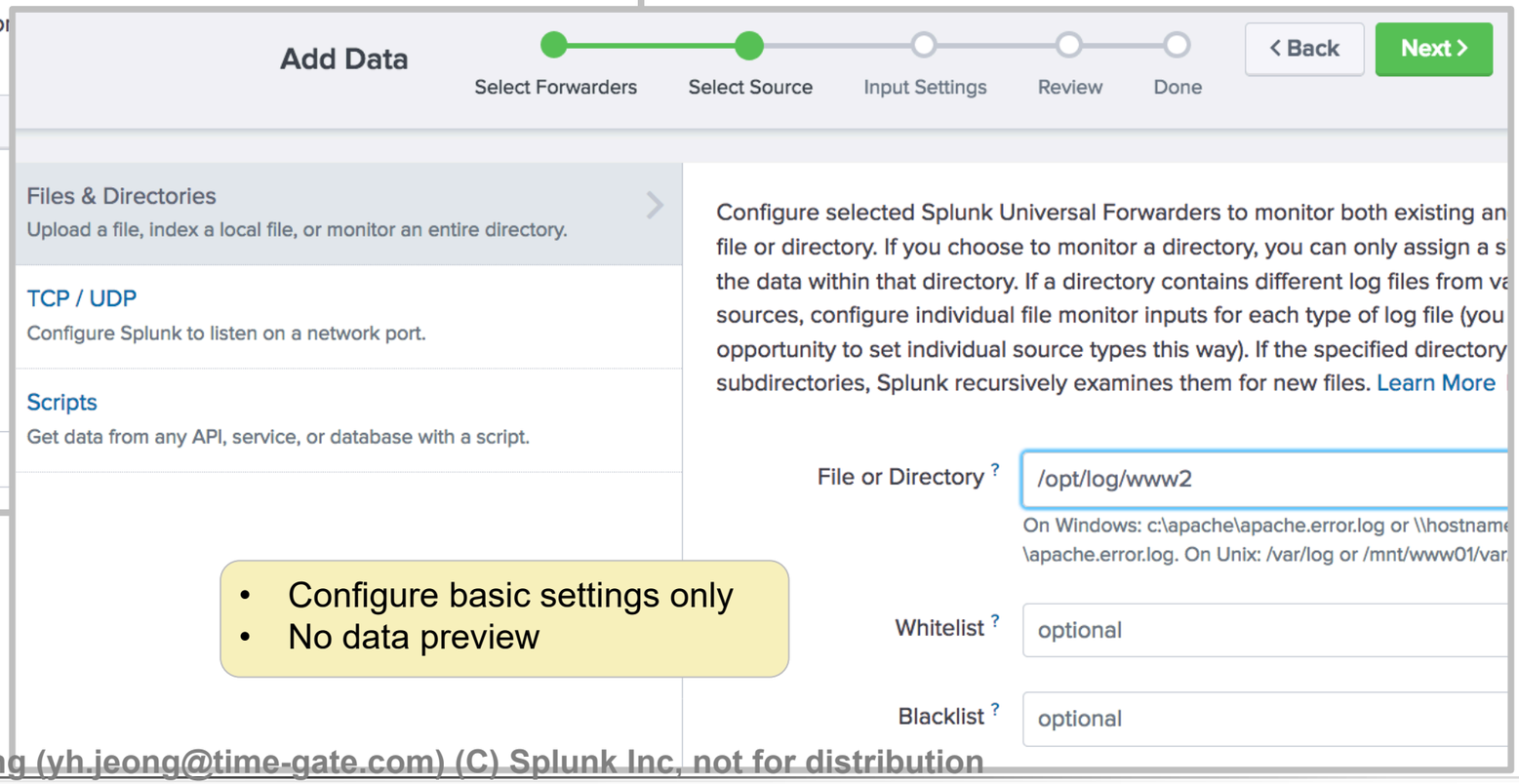
After deployment clients are working, you can create deployment apps for configuring inputs on the clients



Creating a Remote Data Input (cont.)

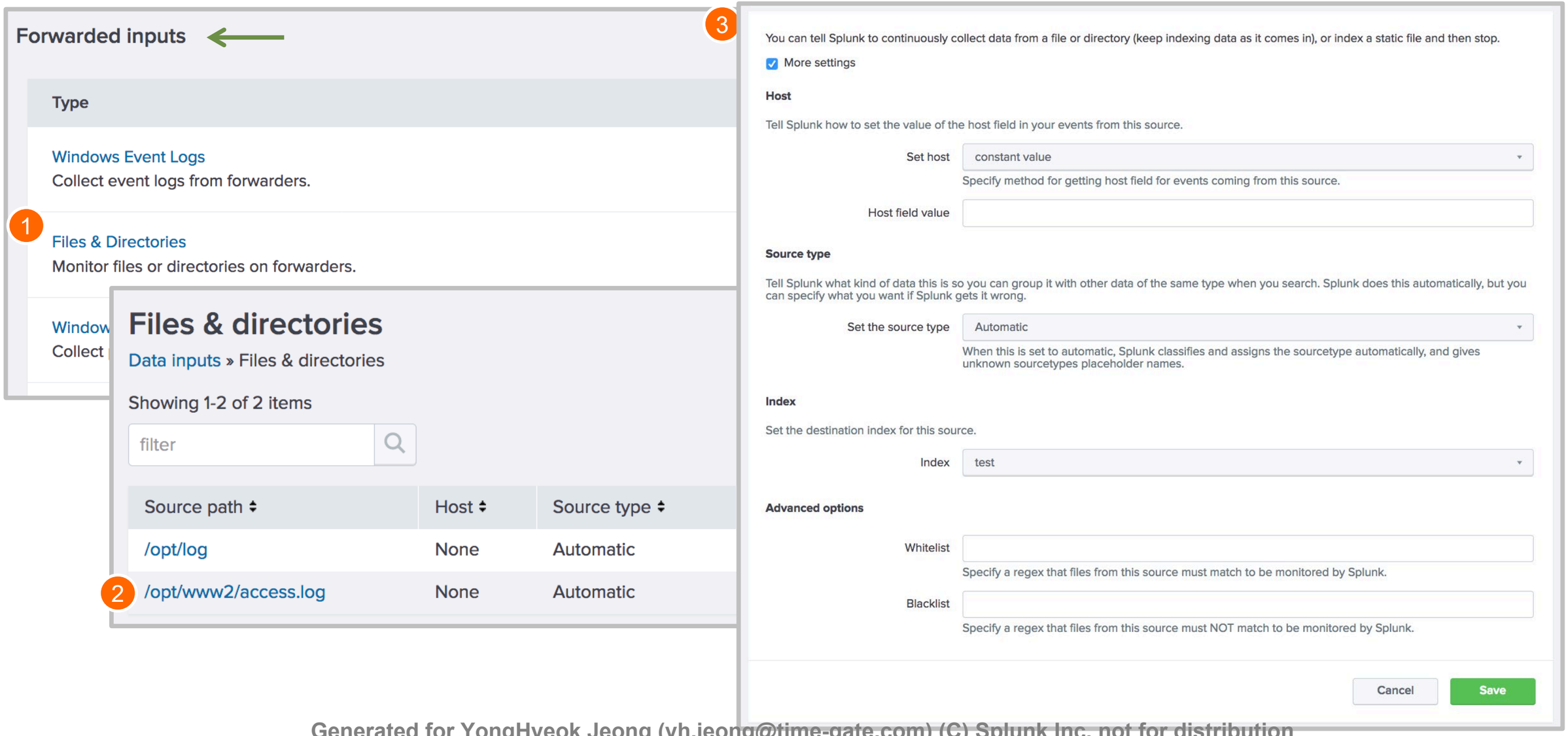


- Creates new server class or uses existing one
- Creates a new app for this input (or updates existing)



- Configure basic settings only
- No data preview

Editing Remote Data Input



Editing Inputs

- Editing **`inputs.conf`** only applies changes to new data, it does not change the data
- Splunk monitor (file or directory) inputs are tracked by the **fishbucket**
  - Ensures that data is not missed or duplicated
  - Keeps checkpoints and other information for each input
- Splunk does *not* re-index when **`inputs.conf`** is edited
- To re-index
  - Delete the old (erroneous) data on the indexer(s)
    - May require assistance from the System Admin
  - Change the **`inputs.conf`** on the deployment server (or forwarders)
  - Reset the **fishbucket** checkpoint on the involved forwarders
  - Restart Splunk forwarders

The Fishbucket and btprobe Command

- To reset the checkpoint for an individual input, use the **btprobe** command:

```
splunk cmd btprobe -d SPLUNK_HOME/var/lib/splunk/
fishbucket/splunk_private_db --file <source> --reset
```

- Requires stopping the forwarder or indexer
- It is possible to clear all checkpoints, but this is only recommended for test environments:
  - **splunk clean eventdata _thefishbucket**
    - Force re-indexing of all file monitors in the indexer
  - **rm -r ~/splunkforwarder/var/lib/splunk/fishbucket**
    - Manually deletes the entire **fishbucket** directory on forwarders
    - Forces the forwarder to resend all its monitor inputs

Note

Resetting the monitor checkpoint re-indexes ALL the data, resulting in more license usage and duplicate events.

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 decide to change the host value, will new host value be reflected for the old data that has already been ingested?
- In our environment, we have a UF, an Indexer and a SH. Which instance contains the **`_fishbucket`**?

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 decide to change the host value, will new host value be reflected for the old data that has already been ingested?

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

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

  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`**

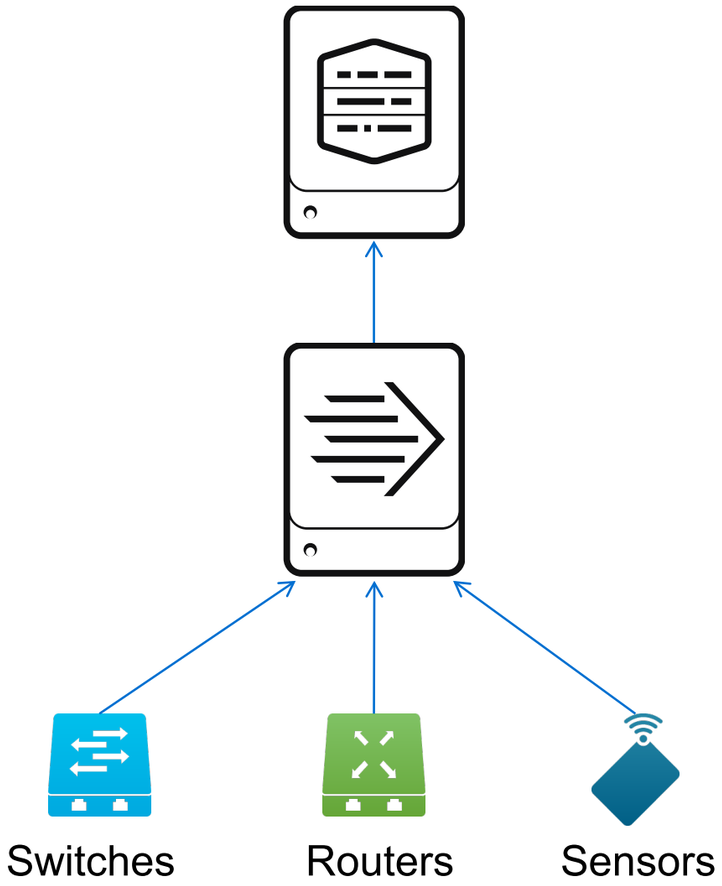
Module 6:
Network & Scripted Inputs

Module Objectives

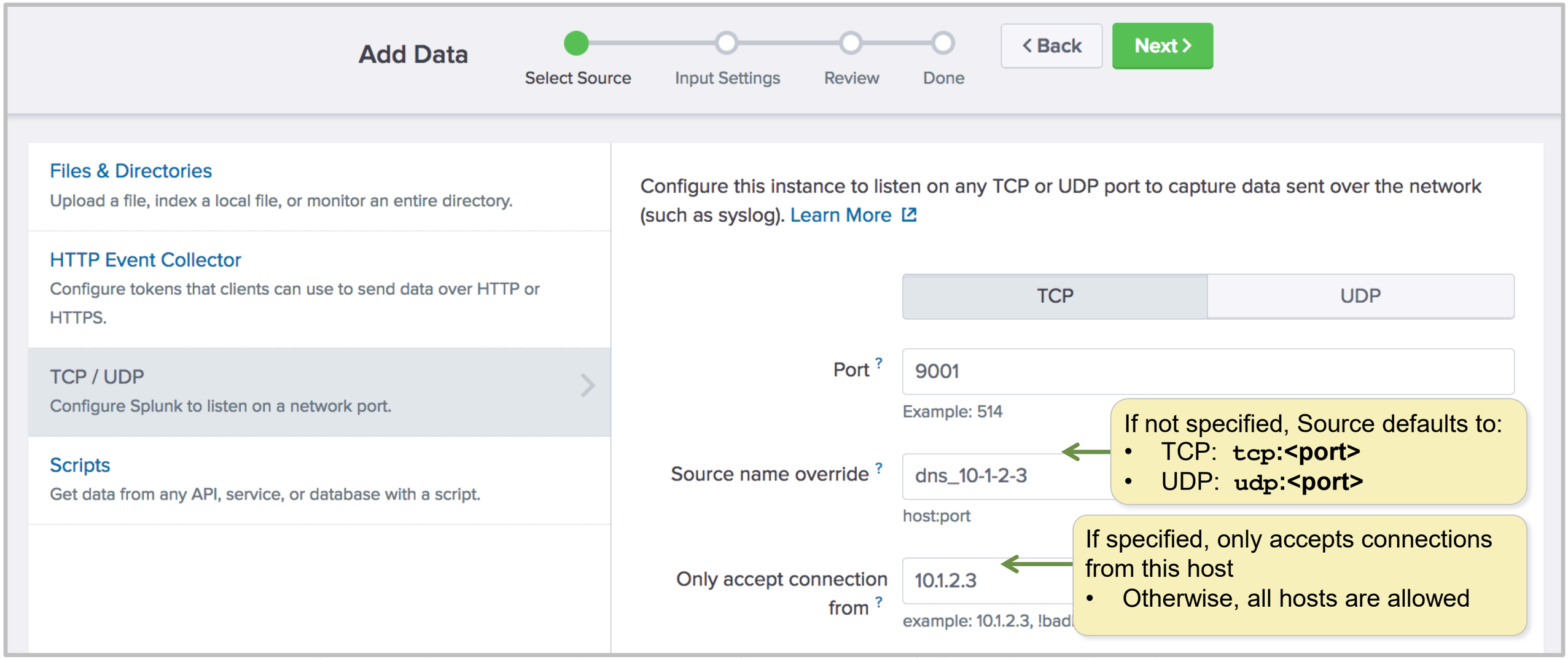
- Create network (TCP and UDP) inputs
- Describe optional settings for network inputs
- Create a basic scripted input

Network Inputs

- A Splunk instance (forwarder or indexer) can listen on a TCP or UDP port for incoming data
  - Syslog is a good example of network-based data
- Add the TCP or UDP input on a forwarder
- Adds a layer of resiliency to your topology
  - Buffering, load balancing, cloning, etc.
  - Indexer restart will not cause data loss of TCP or UDP inputs
  - Minimizes the workload that must be done by the indexer
    - Manage the network connections on the forwarder
    - Can also be useful to bridge network segments if needed



Adding Network Input



Optional Network Input Settings

- You can fine-tune the input settings by editing the stanza directly
  - Metadata override
  - Sender filtering options
  - Network input queues
    - Memory queues
    - Persistent queues
- These settings are described on the following slides

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

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

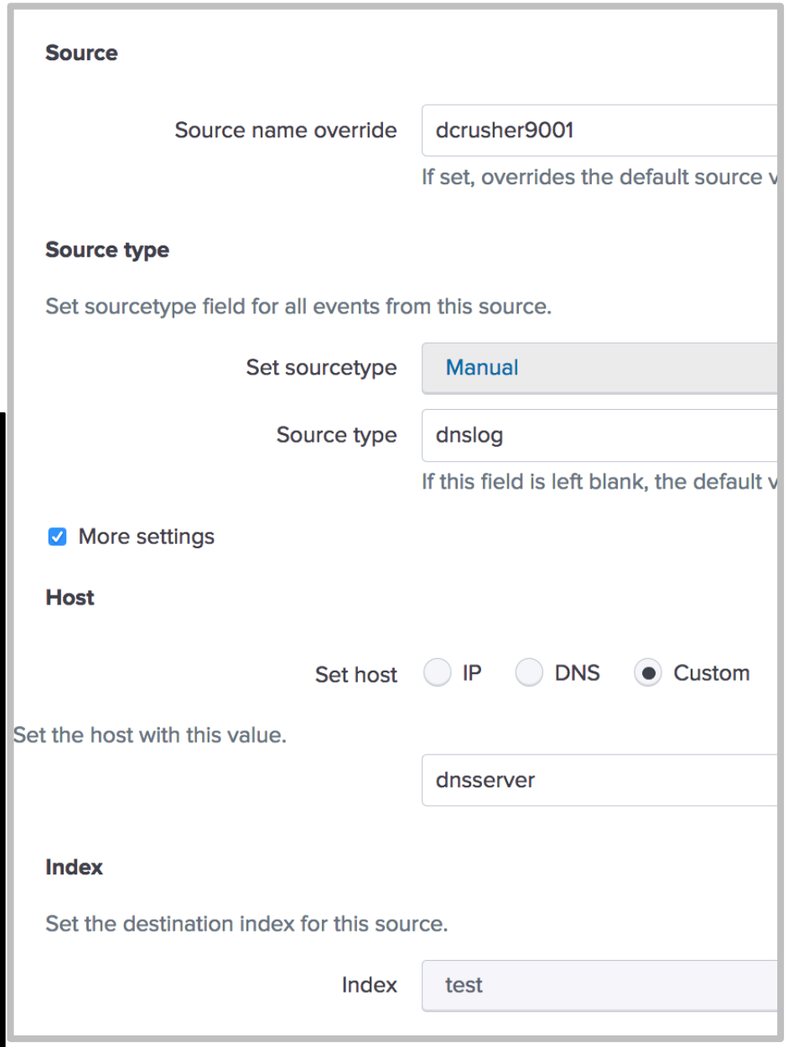
Examples:

```
[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

- The **`connection_host`** attribute defines how the **`host`** field is set:
  - **`dns`** (UI default)
    - The host is set to a DNS name using reverse IP lookup
  - **`ip`**
    - The host is set to the originating host's IP address
  - **`none`** (Custom)
    - Explicitly set 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
```



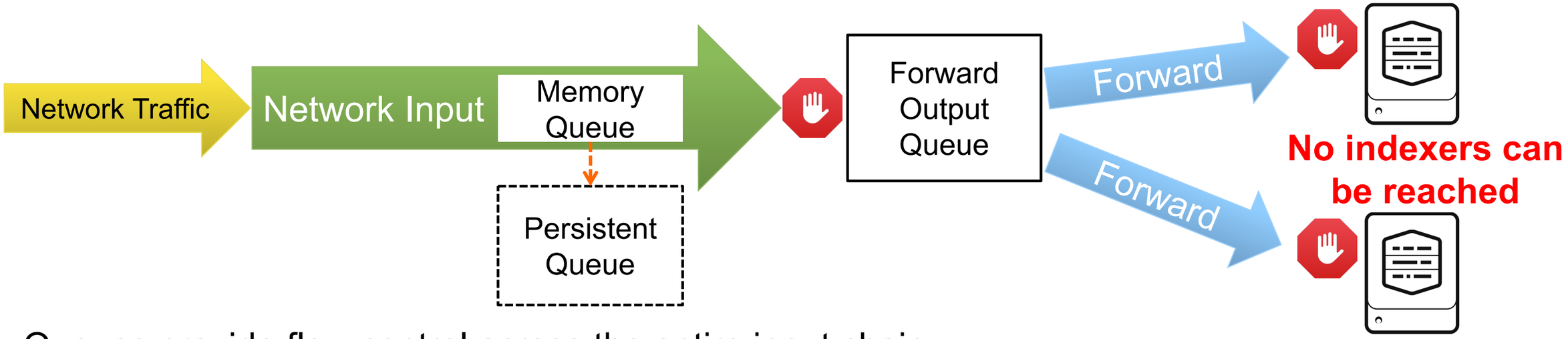
Network Input: `acceptFrom`

Several network devices are sending syslog reports (UDP 514) to my Splunk network input but I want to accept UDP inputs more selectively

- **`acceptFrom = <network_acl>`**
  - List address rules separated by commas or spaces
    - A single IPv4 or IPv6 address
    - A CIDR block of addresses
    - A DNS name
    - A wildcard '*' and '!'

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


Network Input: Queues



- Queues provide flow control across the entire input chain
  - Applies to TCP, UDP, Scripted Input
  - Controls bursts in data over network, slow resources, or slow forwarding
  - If the indexers can't be reached, the forwarder will maintain data in the output queue
    - If the forward output queue ***and*** the indexers cannot be reached, the forwarder uses memory queue and persistent queue
- When memory queue is full, persistent queue (writing to a file) is used and is preserves across restarts
  - Not a solution for input failure

Network Input: Memory Queues

- The **`queueSize`** attribute sets a queue size for input data in KB, MB, or GB
- This is a memory-resident queue that can buffer data before forwarding
- Defaults to 500KB
- Useful if the indexer cannot always receive the data as fast as the forwarder is acquiring it
- Independent of the forwarder's **`maxQueueSize`** attribute defined in **`outputs.conf`**

`inputs.conf`

```
[tcp://9001]
queueSize=10MB
```

Network Input: Persistent Queues

- Provides file-system buffering of data
- Adds additional buffer space after memory buffer
  - You must set a **`queueSize`** first
- A persistent queue is written to disk on the forwarder in **`SPLUNK_HOME/var/run/splunk/...`**
- Useful for high-volume data that must be preserved in situations where it cannot be forwarded, such as if the network is unavailable, etc.

`inputs.conf`

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


Special Handling and Best Practices

- **UDP** – Splunk merges the UDP data until it finds a timestamp by default
  - Can override during the parsing phase

Best practices:

- **Syslog** – Send data to a syslog collector that writes into a directory structure
  - For example, **`/sourcetype/host/syslog.txt`**
    - Monitor the **`sourcetype`** 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

- Splunk can execute scheduled scripts and index the generated output
- Commonly used to collect diagnostic data from OS commands
  - For example: **`top`**, **`netstat`**, **`vmstat`**, **`ps`**, etc.
  - Many Splunk apps use scripted inputs to gather specialized information from the OS or other applications running on the server
- Also good for gathering any transient data that cannot be collected with Monitor or Network inputs
  - APIs, message queues, Web services, or any other custom transactions
- Splunk can run:
  - Shell scripts (**`.sh`**) on *nix
  - Batch (**`.bat`**) and PowerShell (**`.ps1`**) on Windows
  - Python (**`.py`**) on any platform

Scripted Input Stanza

Splunk only executes scripts from **SPLUNK_HOME/etc/apps/<app_name>/bin**, **SPLUNK_HOME/bin/scripts**, OR **SPLUNK_HOME/etc/system/bin**

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

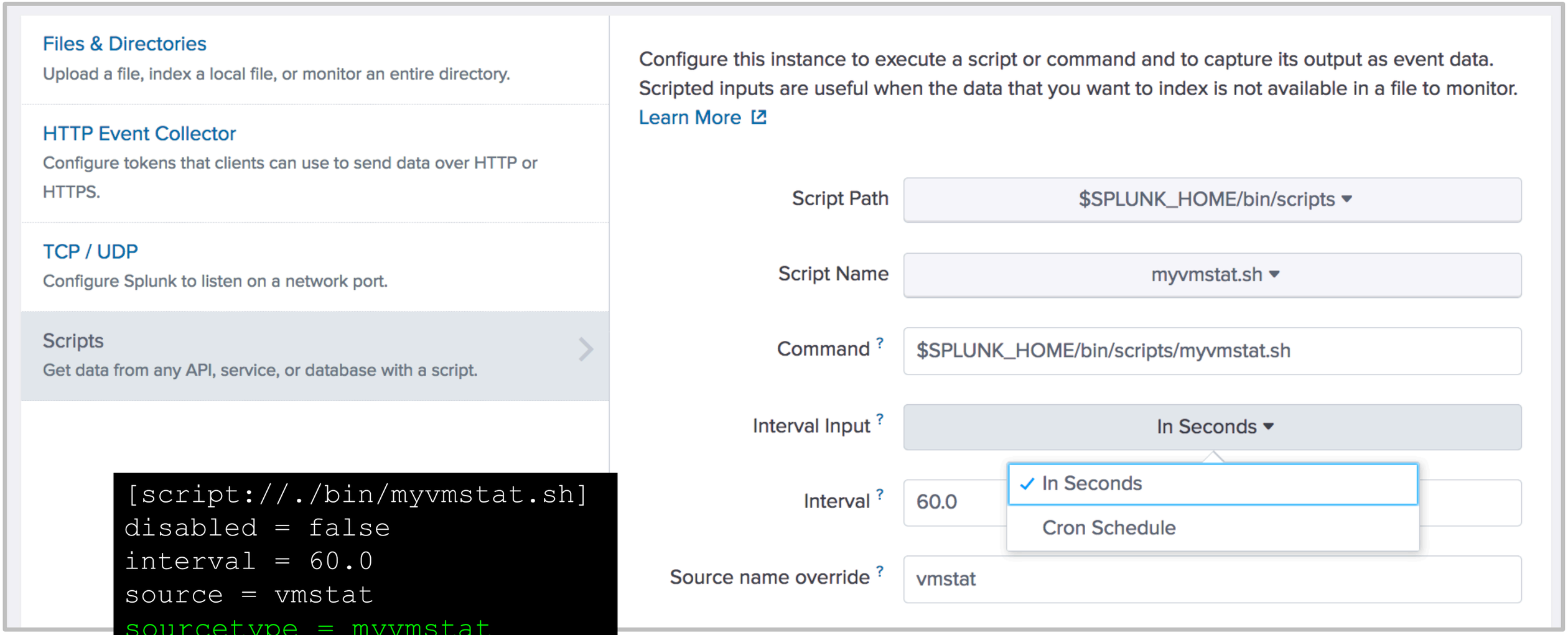
Use **passAuth** to run the script as the specified OS user – Splunk passes the auth token via stdin to the script

Interval is the time period between script executions – defaults to 60 seconds

Defining a Scripted Input

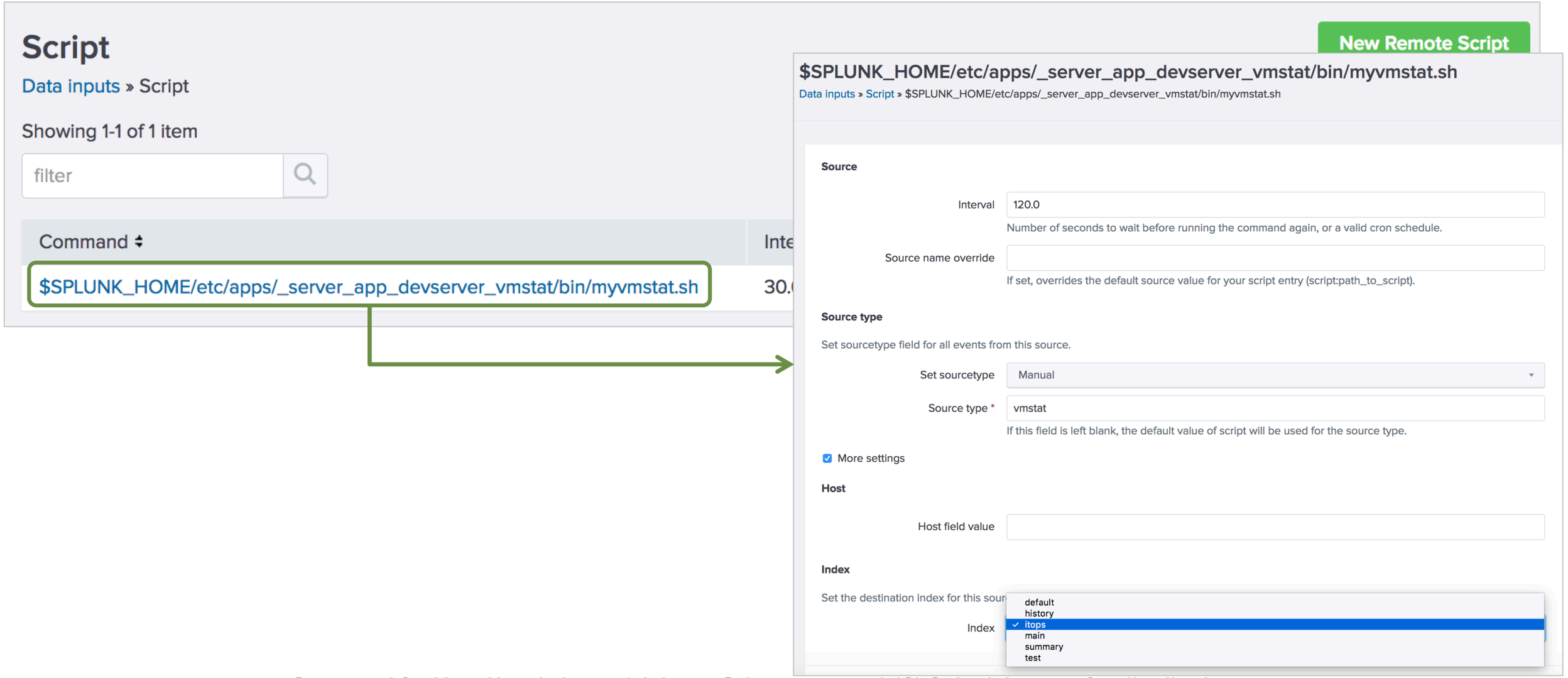
1. Develop and test the script
2. Always test your script from the context of an app and make sure it runs correctly
   - On the test/dev server, copy the script to an app's **`bin`** directory
   - To test the script from the Splunk perspective, run **`splunk cmd scriptname`**
     - **`./splunk cmd ../etc/apps/<app>/bin/<myscript.sh>`**
3. To deploy a scripted input using the Deployment Server
   - Copy the verified script to the appropriate directory first (**`deployment-apps/<app>/bin/`**)
   - Deploy the script using the **Add Data** wizard, Forward option from the Splunk Web UI
4. Verify the output of the script is being indexed

Scripted Inputs Example




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

Editing Scripted Inputs



Scripted Input Buffering

- One possible downside to scripted input is potential loss of data
  - Example: the forwarder that is running the script is not able to connect to the indexer due to networking problems
- You can declare the same **`queueSize`** and **`persistentQueueSize`** attributes for a script stanza as for network (TCP and UDP) inputs
  - Buffers data on the forwarder when the network or indexer is not available

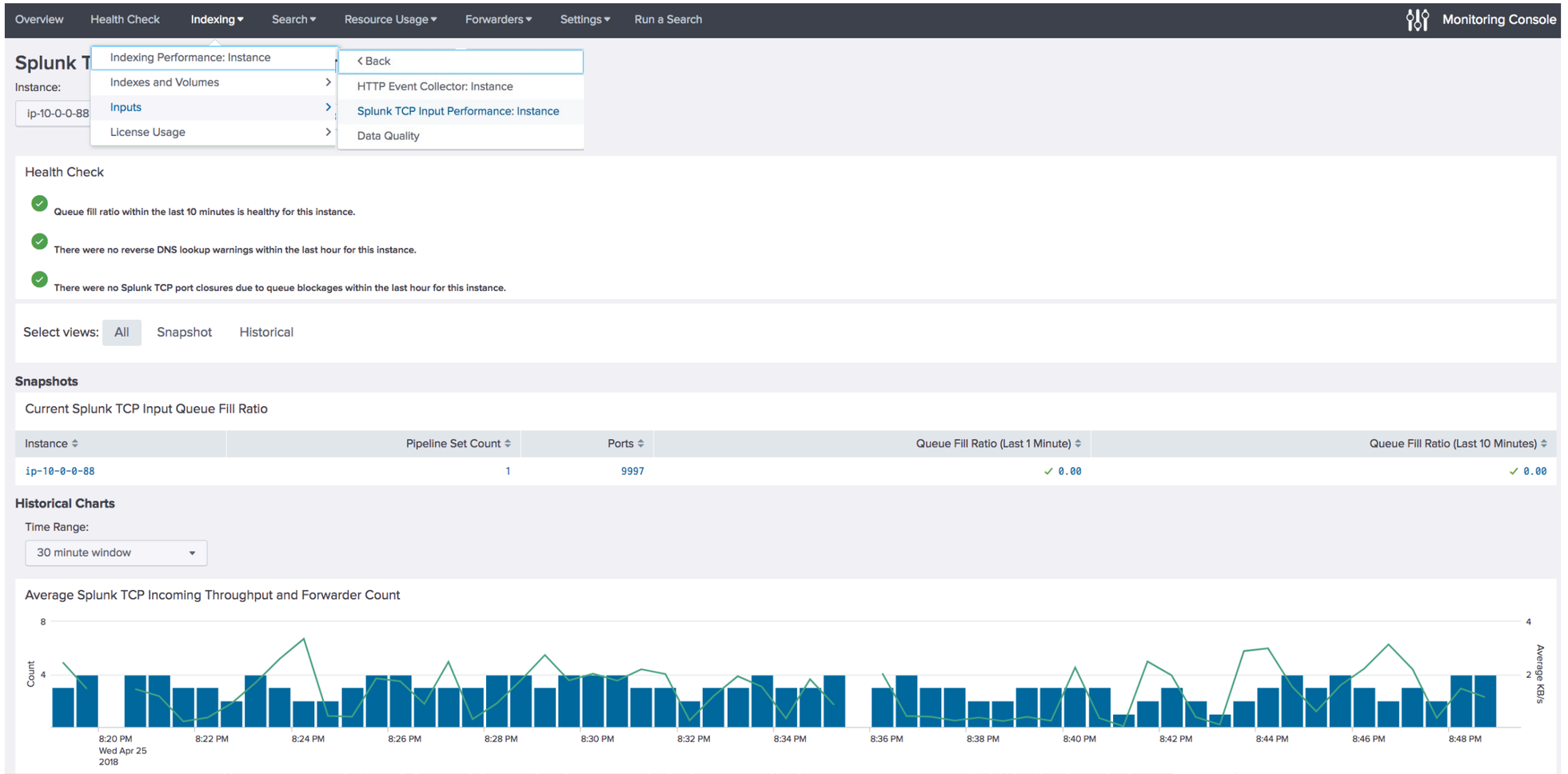
Alternate to Scripted Input

- Set up your script to run as a CRON job and append data to a log file
- Set up a monitor input to ingest the log file
  - Takes advantage of the file system and Splunk's robust file monitoring capabilities
  - Can easily recover even when forwarder goes down
- 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 & Agentless Inputs

Module Objectives

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

Windows-Specific Inputs

- Windows OS maintains much of its state data (logs, etc.) in binary format
  - Windows provides APIs that enable programmatic access to this information
- Splunk provides special input types to access this data
  - All other input types are also supported
  - Data can be forwarded to any Splunk indexer on any OS platform
  - Windows Universal Forwarder can run as domain user without the local administrator privilege

Windows-Specific Input Types

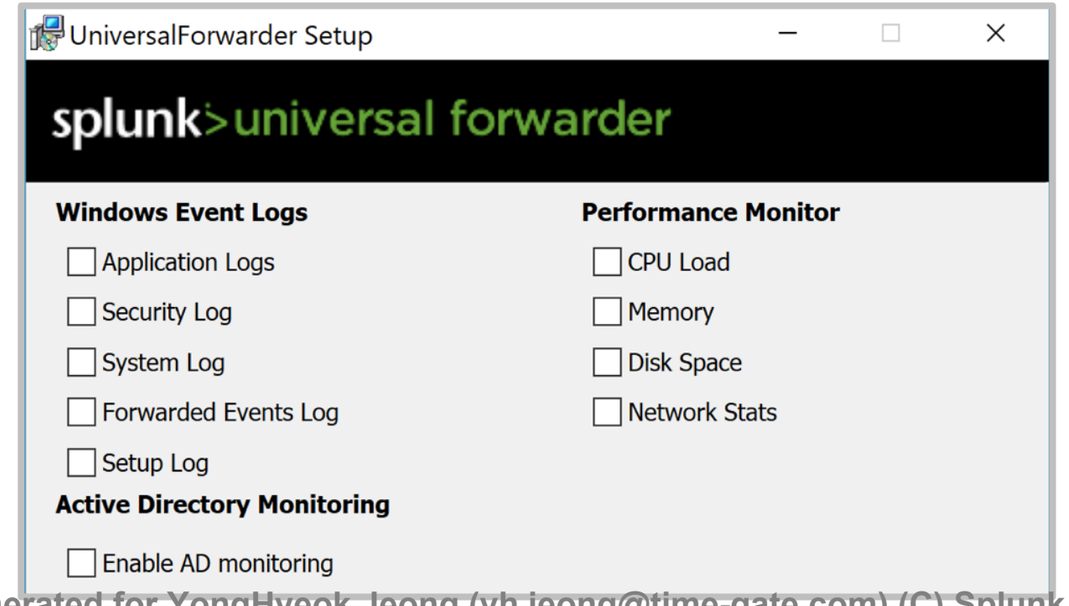
Input Type	Description
Event Log*	Consumes data from the Windows OS logs
Performance*	Consumes performance monitor data
Active Directory	Monitors changes in an Active Directory server
Registry	Monitors changes in a Windows registry
Host	Collects data about a Windows server
Network	Monitors network activity 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
- Or, configure them manually:
  - See **`inputs.conf.spec`** and **`inputs.conf.example`** for details on setting up each Windows input type

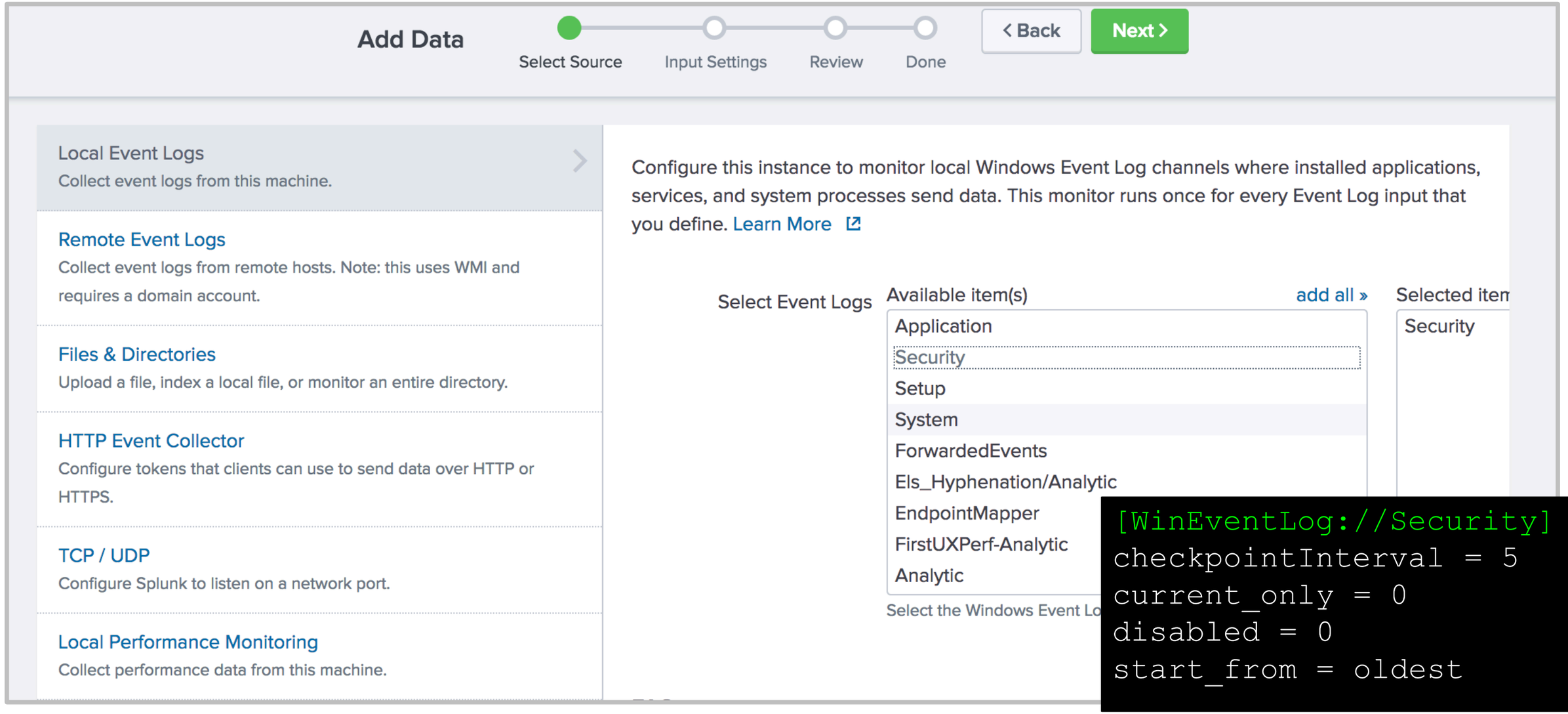
```
[admon://name]
[perfmon://name]
[WinEventLog://name]
[WinHostMon://name]
[WinNetMon://name]
[WinPrintMon://name]
[WinRegMon://name]
```



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



Windows Input Configuration Options

- You can filter out non-essential events on the Windows Universal Forwarder
  - Set **whitelist** and **blacklist** based on event field names and regex
  - Allows you to target specific events while filtering out lower value events
    - **`whitelist = <list> | key=regex [key=regex]`**
    - **`blacklist = <list> | key=regex [key=regex]`**
  - Can configure up to 10 whitelist and 10 blacklist per stanza
    - Blacklist overrides whitelist if conflicts occur

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

Local vs. Remote Windows (WMI) Inputs

- You can configure remote inputs (WMI) for two types of Windows inputs:
  - Event logs
  - Performance monitor
- Advantage:
  - You can collect the 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

WMI Inputs

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

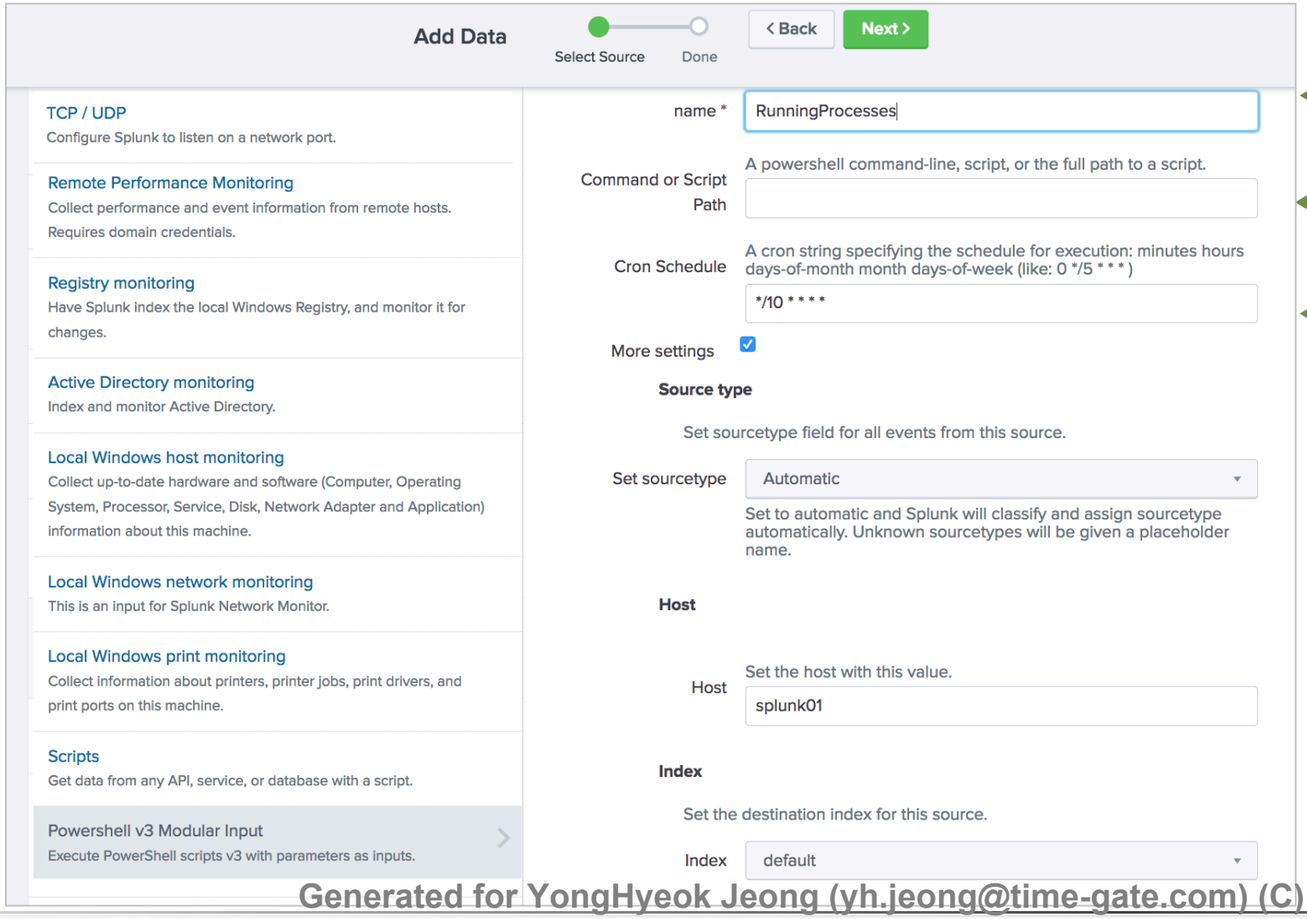
```
[WMI:remote-logs]
interval = 5
server = server1, server2, server3
event_log_file = Application, Security, System

[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
  - For example, IIS/W3C, JSON, and other delimited/structured sources
- Challenges:
  - These logs often get re-configured by the product administrator
    - Requires some sort of coordination between the source administrator and the Splunk administrator to sync up 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 executes once

```
[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

Splunk 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
    - Can facilitate logging from distributed, multi-modal, and/or legacy environments
    - Log data from a web browser, automation scripts, or mobile apps

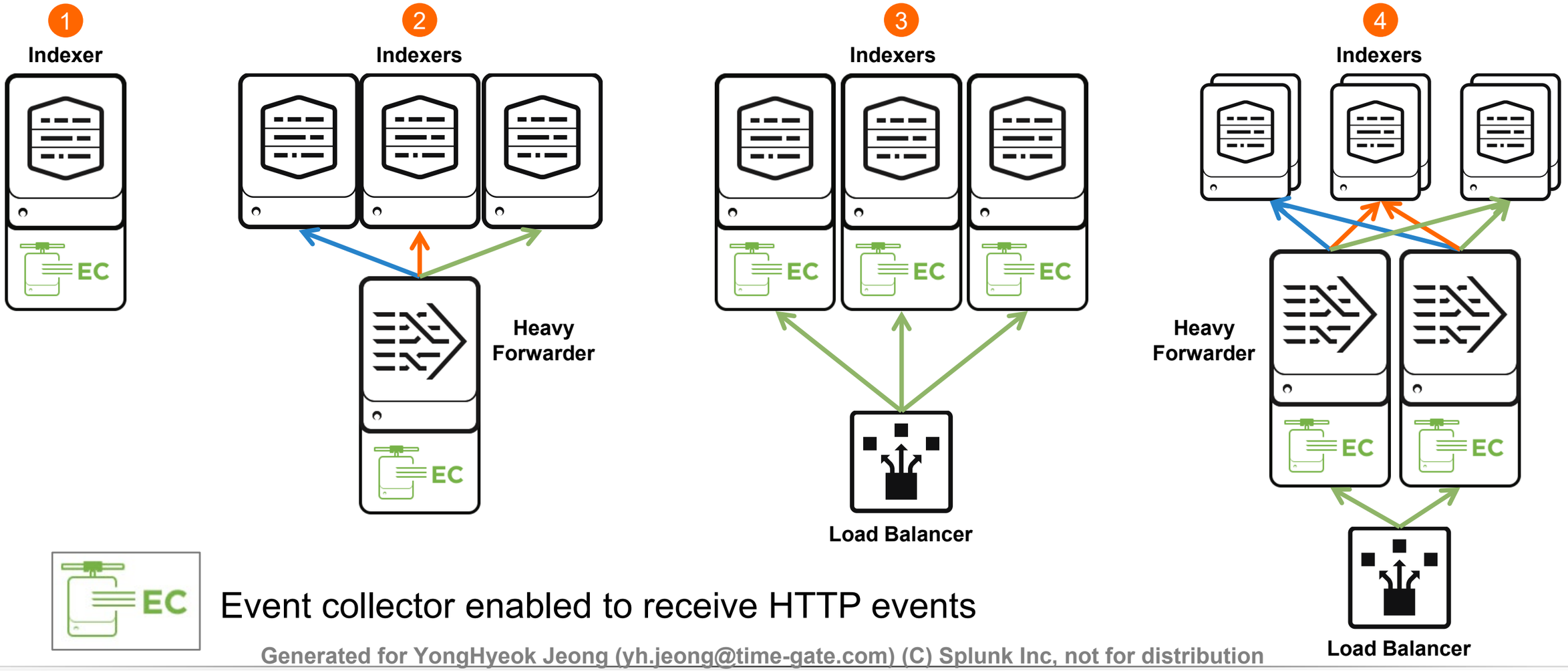
    http://dev.splunk.com/view/event-collector/SP-CAAAE6M
- **Splunk App for Stream** (Splunk-supported free app)
  - An alternative way to collect “difficult” inputs
    - No visibility into DB servers because DBAs refuse to install any agents on SQL servers
    - Web logs alone do not provide enough visibility into nefarious web traffic
  - Able to read data off the wire
  - Supports Windows, Mac, and Linux

    http://docs.splunk.com/Documentation/StreamApp
  - Refer to Blogs: Tips & Tricks on HTTP Event Collector

http://blogs.splunk.com/2015/10/06/http-event-collector-your-direct-event-pipe-to-splunk-6-3

Distributed HEC Deployment Options

HEC can scale by taking advantage of Splunk distributed deployment



Event collector enabled to receive HTTP events

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
  - While you can send data from any client, you can simplify the process by using the Splunk logging libraries
    - Supports JavaScript, Java and .NET logging libraries
- POST data in JSON format to the token receiver

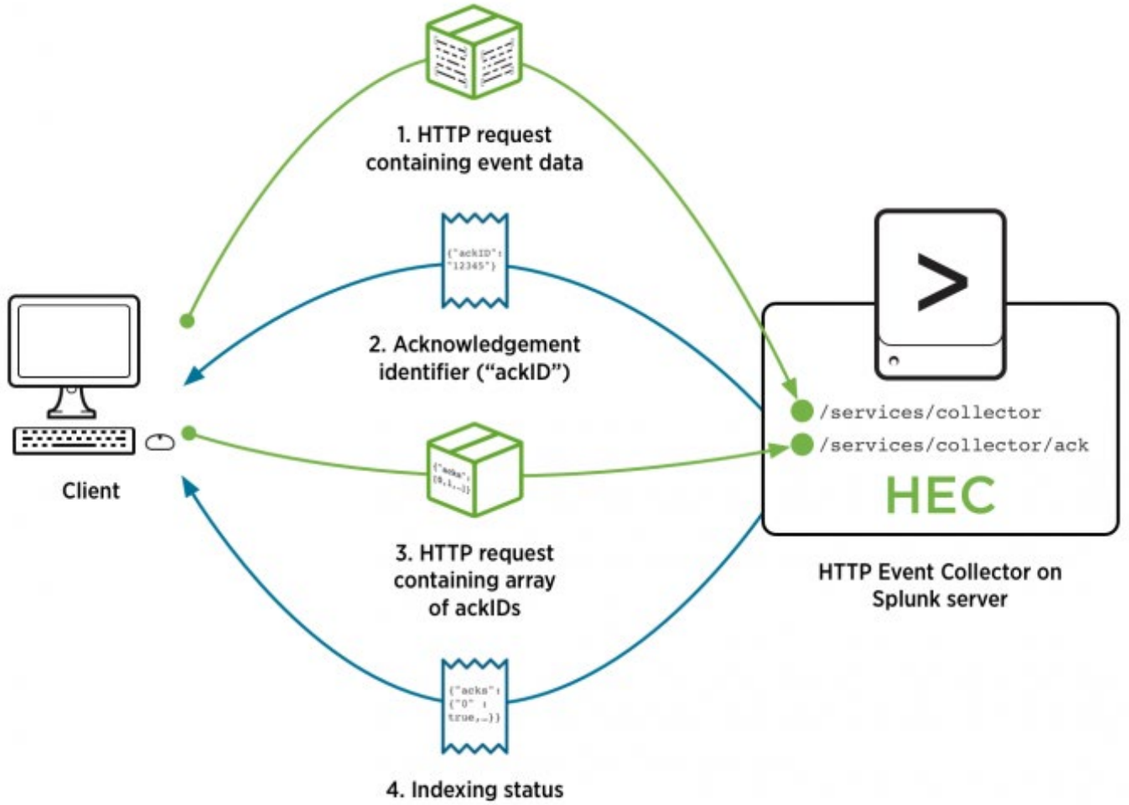
```
curl "http[s]://<splunk_server>:8088/services/collector"
 -H "Authorization: Splunk <generated_token>"
 -d '{
      "host":"xyz",
      "sourcetype":"fl01_S2",
      "source":"sensor125",
      "event": {"message":"ERR", "code":"401"}
     }'
```

HTTP Event Collector Options

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

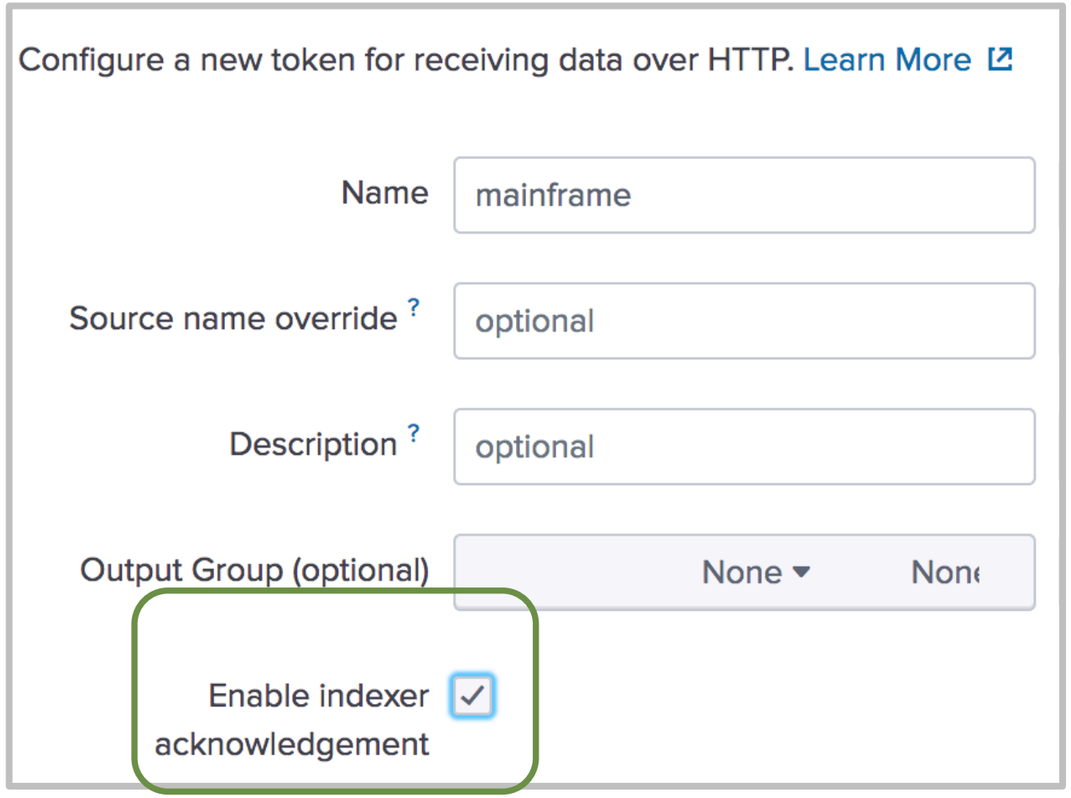
HEC Indexer Acknowledgement

1. A request is sent from a client to the HEC endpoint using a token with indexer acknowledgment enabled
2. The server returns an acknowledgment identifier (`ackID`) to the client
3. The client can then query the Splunk server with the identifier to verify whether all the events in the sent request of have been indexed
4. The Splunk server responds with the status information of each queried request



Enabling HEC Indexer Acknowledgments

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



http://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'
```

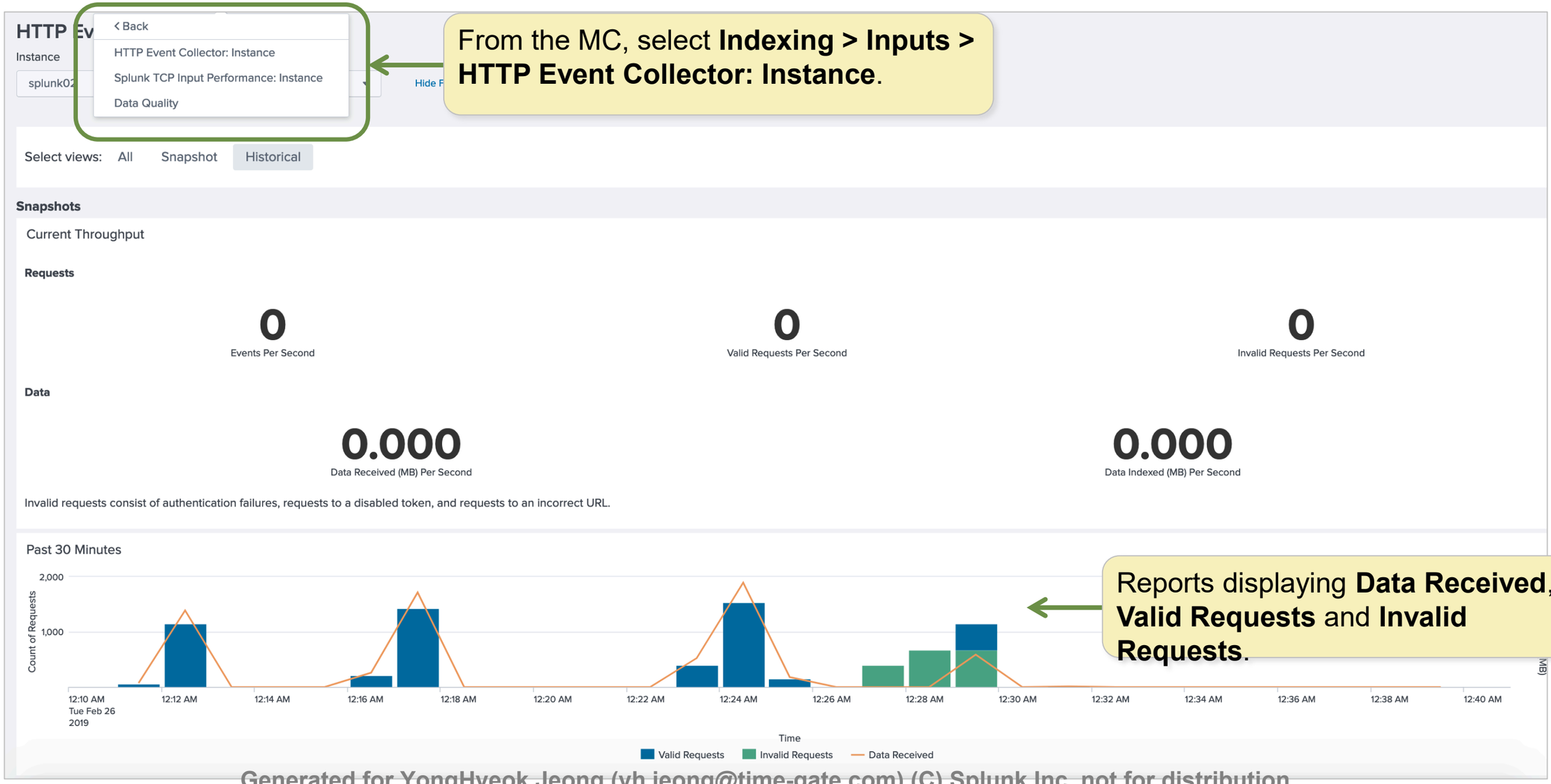

Configuring Dedicated HTTP Settings

- 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.

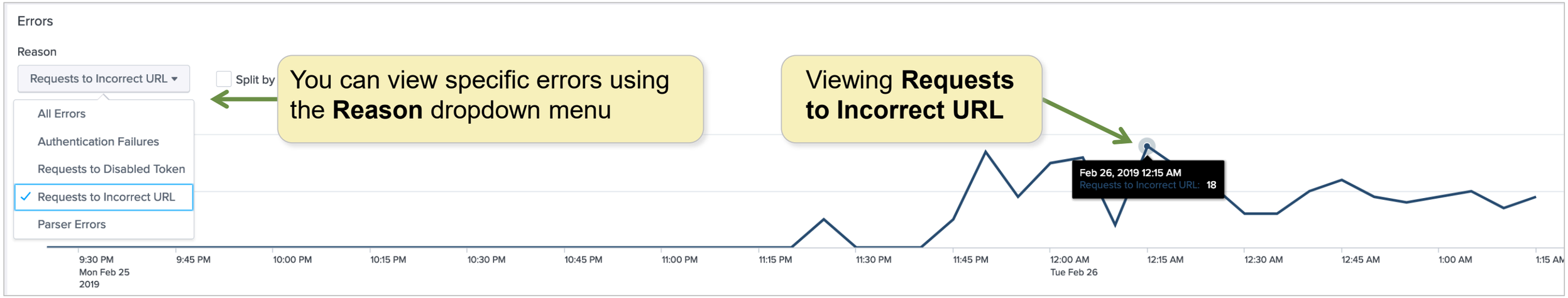
`inputs.conf`

```
[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



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 ActiveDirectory 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 ActiveDirectory 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

- Every Splunk deployment should have a test environment
  - It can be a laptop, virtual machine, or spare server
  - It should have the same version of Splunk running in production
- Test your input for a new source of data and evaluate it in a test environment
- If not:
  - Create a **`test`** index and send test inputs to this index
    - You can delete it when needed
    - Does not require **`splunkd`** restart
  - Use **Data Preview** to evaluate new data sources without actually inputting

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

- Required for all inputs
  - Optional

What if I Don't Get It Right?

- On a testing and development system
  - It’s okay; 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

- **`props.conf`** is a config file that is referenced during all phases of Splunk data processing
  - Inputs, indexing, parsing and searching
- See **`props.conf.spec`** and **`props.conf.example`** files in **`SPLUNK_HOME/etc/system/README`** for specifics
  docs.splunk.com/Documentation/Splunk/latest/admin/Propsconf

`props.conf` Stanza

- 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

`props.conf` in the Input and Parsing Phases

- Some settings in **`props.conf`** are applied during the input phase:
  - Character encoding
  - Fine-tuning source types
  - A few others
- Some settings in **`props.conf`** are applied during the parsing phase:
  - Individual event breaking
  - Time extraction settings and rules
  - Event data transformation
- Configure **`props.conf`** on your forwarders if you have input phase settings

  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
  - This 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 and specify a **source type** explicitly, it applies to all files in the directory and subdirectories
- You can omit the **source type** attribute in **inputs.conf**
  - Splunk will try to use automatic pre-trained rules
- You can then selectively override the source type with **props.conf**
  - Identify the input with a **[source::<source>]** stanza and set the **sourcetype** attribute
  - This is an input phase process
  - **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**

inputs.conf

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

props.conf

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

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

...
```


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 sourcetypes 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

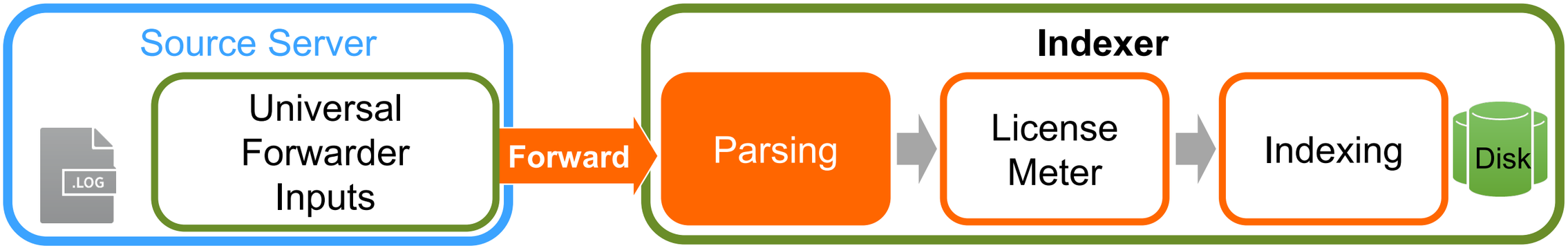
Module 9: Parsing Phase and Data Preview

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

- As data arrives at the indexer, it goes through the **parsing** phase
  - The data is broken up into discrete **events**, each with a **timestamp** and a **time zone**
- The parsing phase is all about creating, modifying, and redirecting events
  - Apply additional transformation steps to modify the metadata fields or modify raw data
  - Both indexers and heavy forwarders parse events
    - In this module, we assume parsing is happening on an indexer



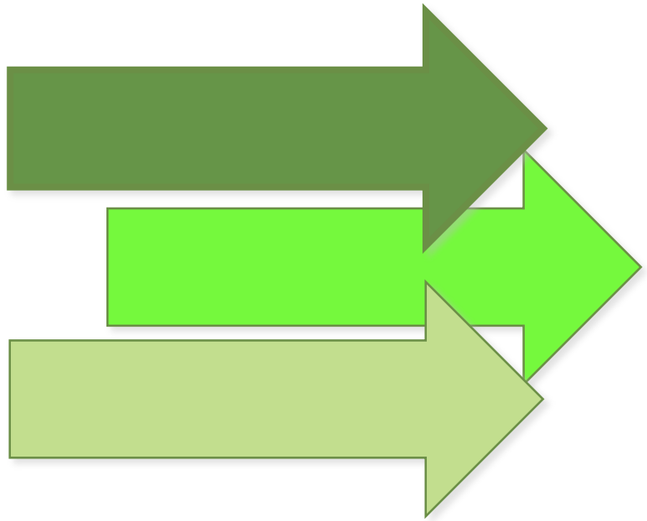
Event Creation

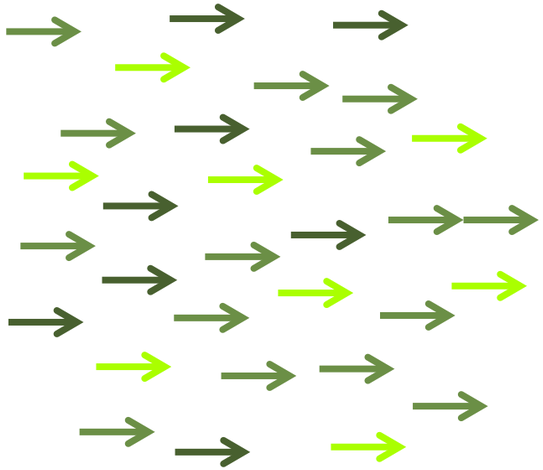
During the parsing phase, data from input phase is broken up into individual events, and then event-level processing is performed

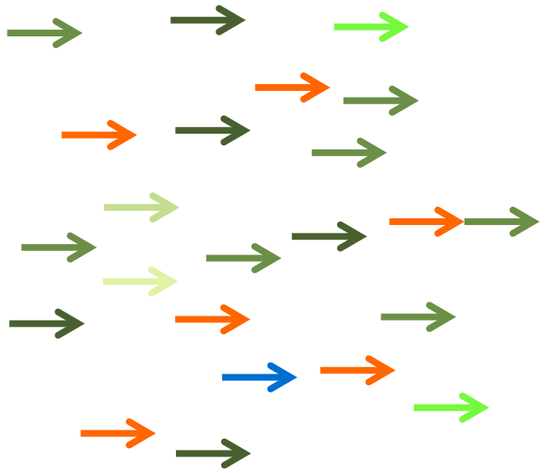
"Streams" of data from inputs phase

Parsed into individual events

Event-by-event processing

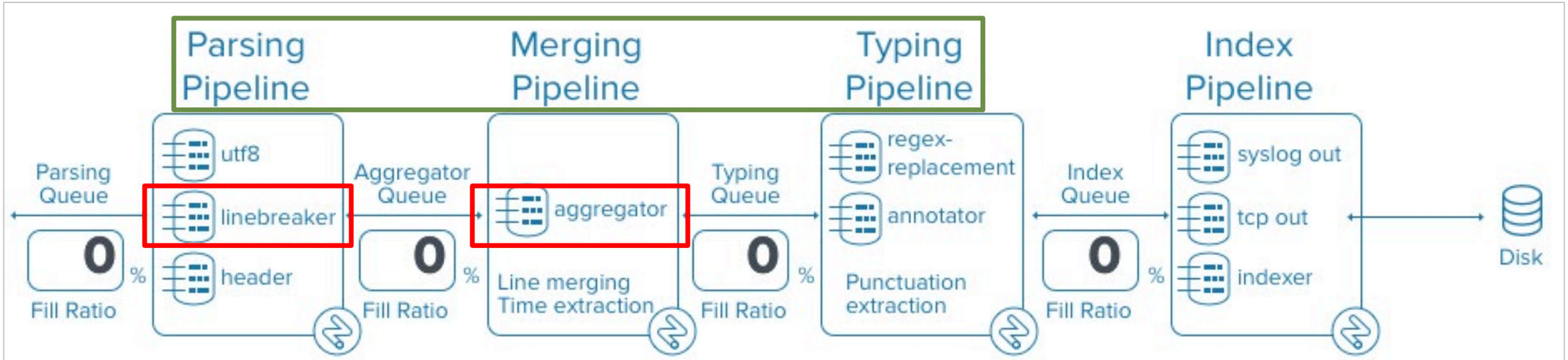






Event Boundaries

- Splunk parsing phase determines where one event ends and the next one begins
  - Automatically handles line breaking for common source types – even multi-line events
- This ***line breaking*** process involves a series of pipelines
  - Each pipeline consists of a set of queues and processors
- Use **Data Preview** when on-boarding a new source type



Handling Single Line Events

- Splunk handles single line event sourcetypes with automatic line breaking
- It is more efficient to explicitly set:
  - **SHOULD_LINEMERGE = false**
  - Default is **true** and assumes events can span over more than one line

SPLUNK_HOME/etc/apps/mycustom_addon/local/props.conf

```
[my_custom_one_event_per_line_sourcetype]
SHOULD_LINEMERGE = false
```

Configuring Line Breaking

- Splunk determines event boundaries in two steps:
  - Line breaking: `LINE_BREAKER = <regular_expression>`
    - Splits the incoming stream of bytes into separate lines
    - The default value is `([\r\n]+)` which is any sequence of new lines and carriage returns
    - Correct use of regular expression can produce results in first step
  - Line merging: `SHOULD_LINEMERGE = true`
    - When set to `true` (the default) it uses all other line merging settings (such as `BREAK_ONLY_BEFORE`, `BREAK_ONLY_BEFORE_DATE`, `MUST_BREAK_AFTER`)
    - When set to `false`, the line merging step does not run

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

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`**

TIME_PREFIX

- **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
```

MAX_TIMESTAMP_LOOKAHEAD

- **`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

TIME_FORMAT

- **`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

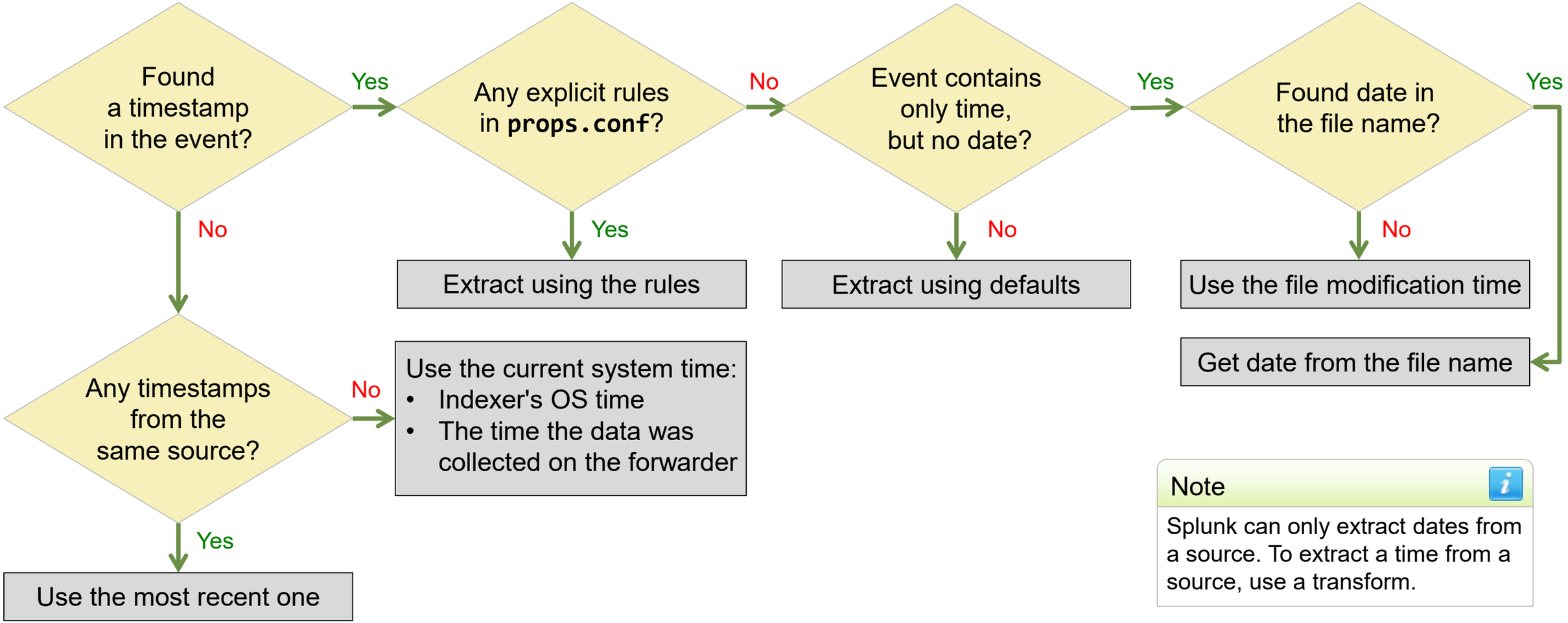
Setting Time Zones – Splunk's Rules

- 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



Note

Splunk can only extract dates from a source. To extract a time from a source, use a transform.

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

Using Splunk Data Preview

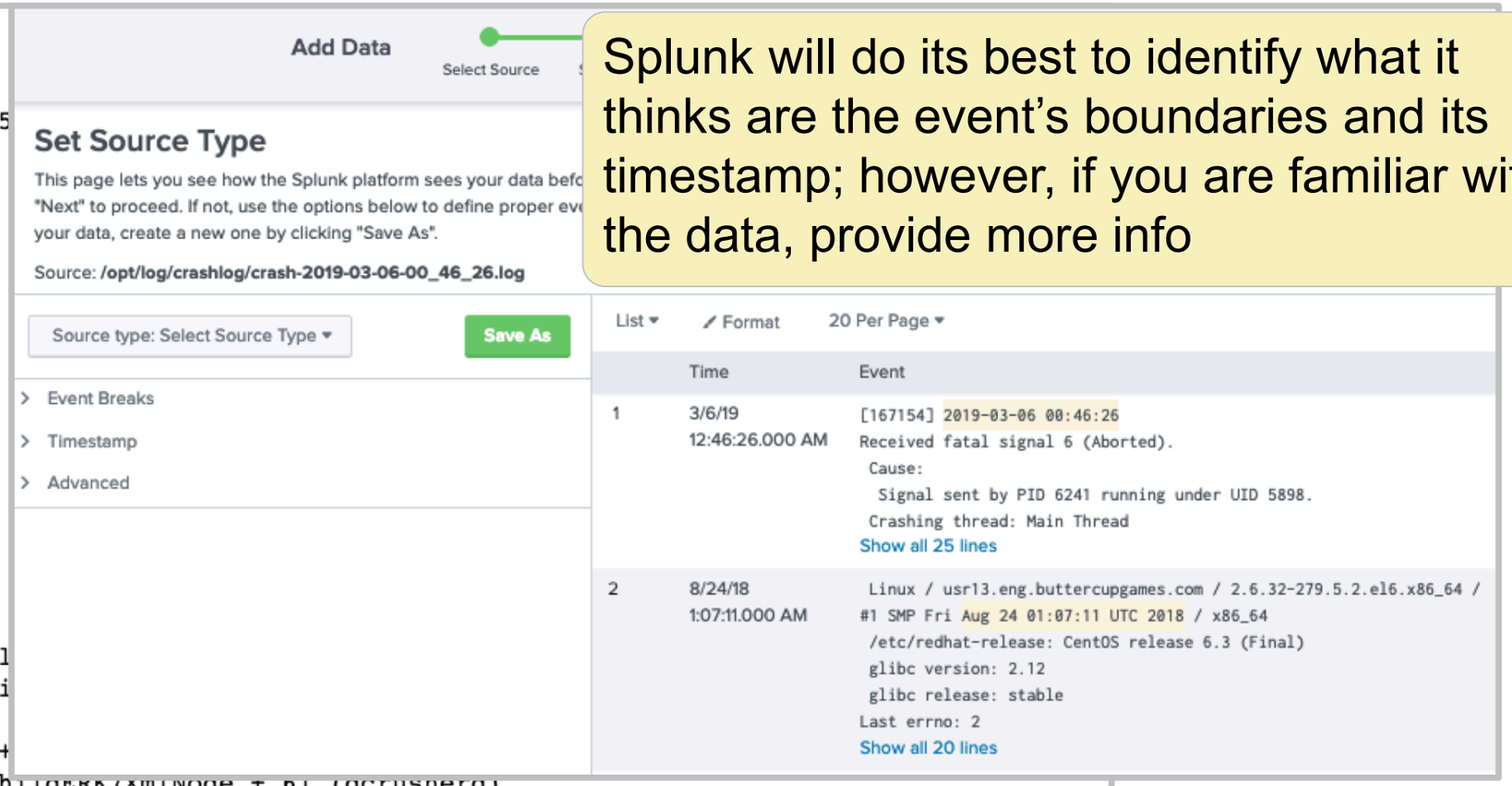
- Splunk attempts to auto-detect a source type
  - You can also select from a list or define your own source type
  - Supports both unstructured and structured data sources
  - CSV, JSON, W3C/IIS, XML, etc.
- Event breaking and date/timestamp settings are evaluated
  - Use your sandbox environment or 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
```

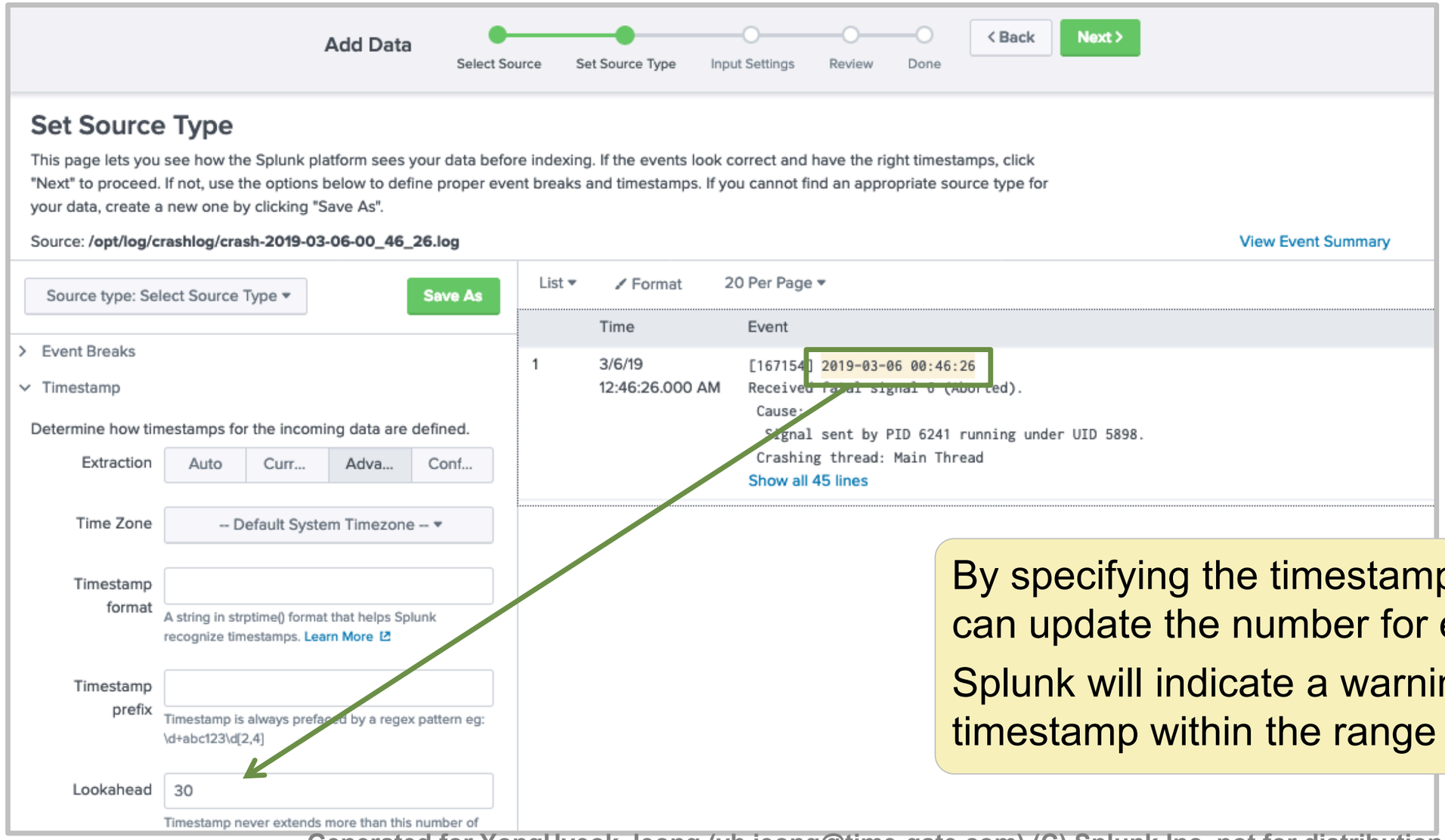
1

2



Splunk will do its best to identify what it thinks are the event's boundaries and its timestamp; however, if you are familiar with the data, provide more info

Previewing Unstructured Data (cont.)

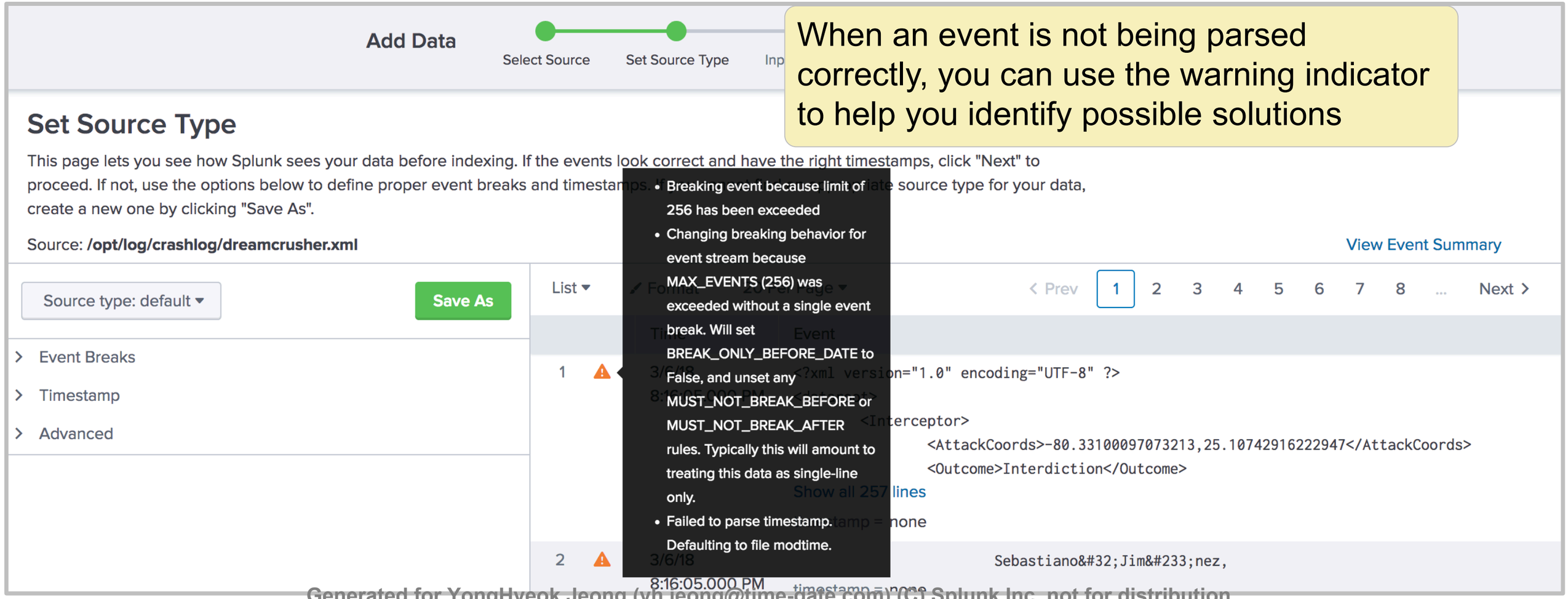


By specifying the timestamp location, Splunk can update the number for events extracted

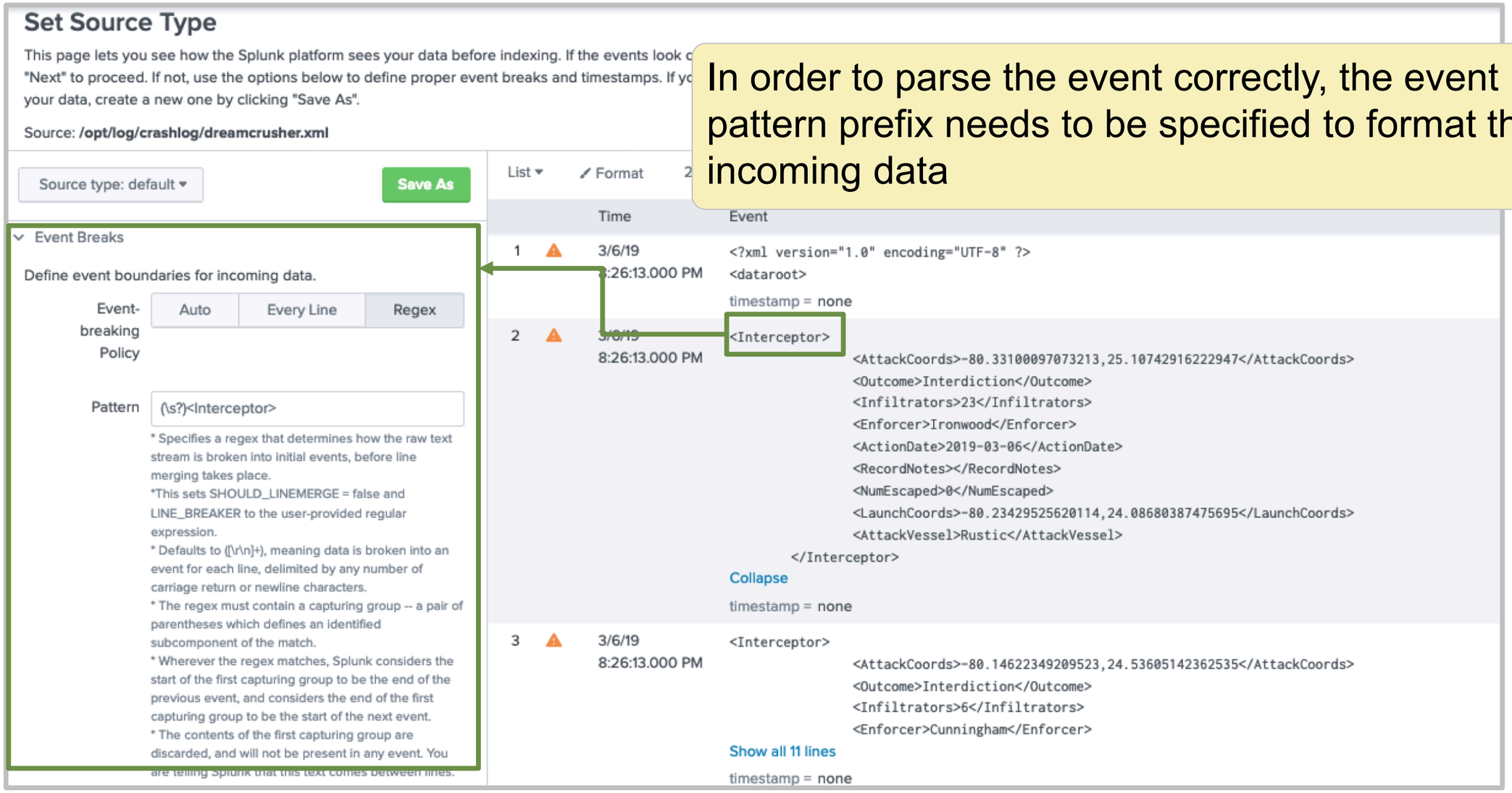
Splunk will indicate a warning if it cannot find a timestamp within the range

Previewing Unstructured Data (cont.)

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

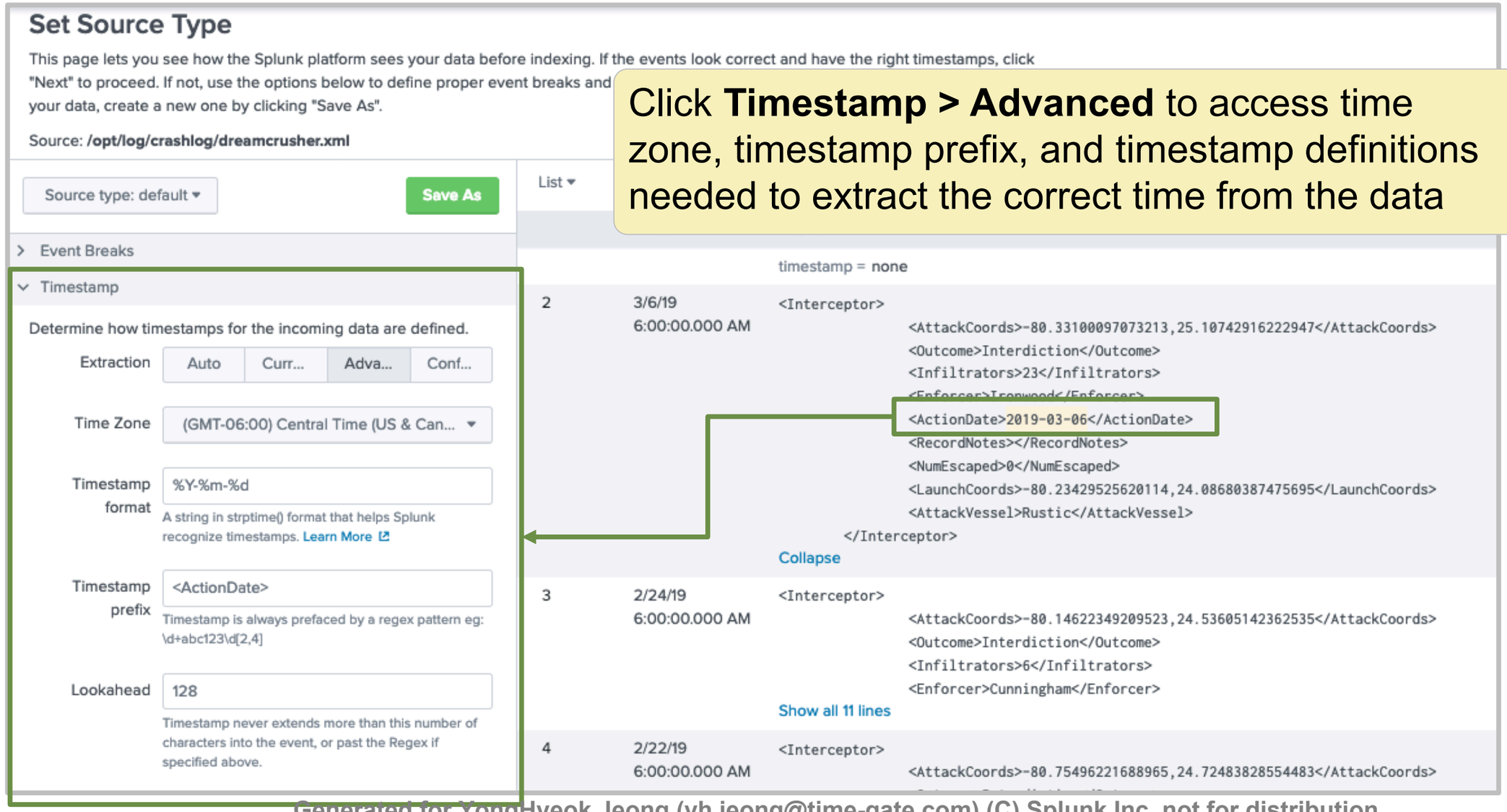


Previewing Unstructured Data (cont.)



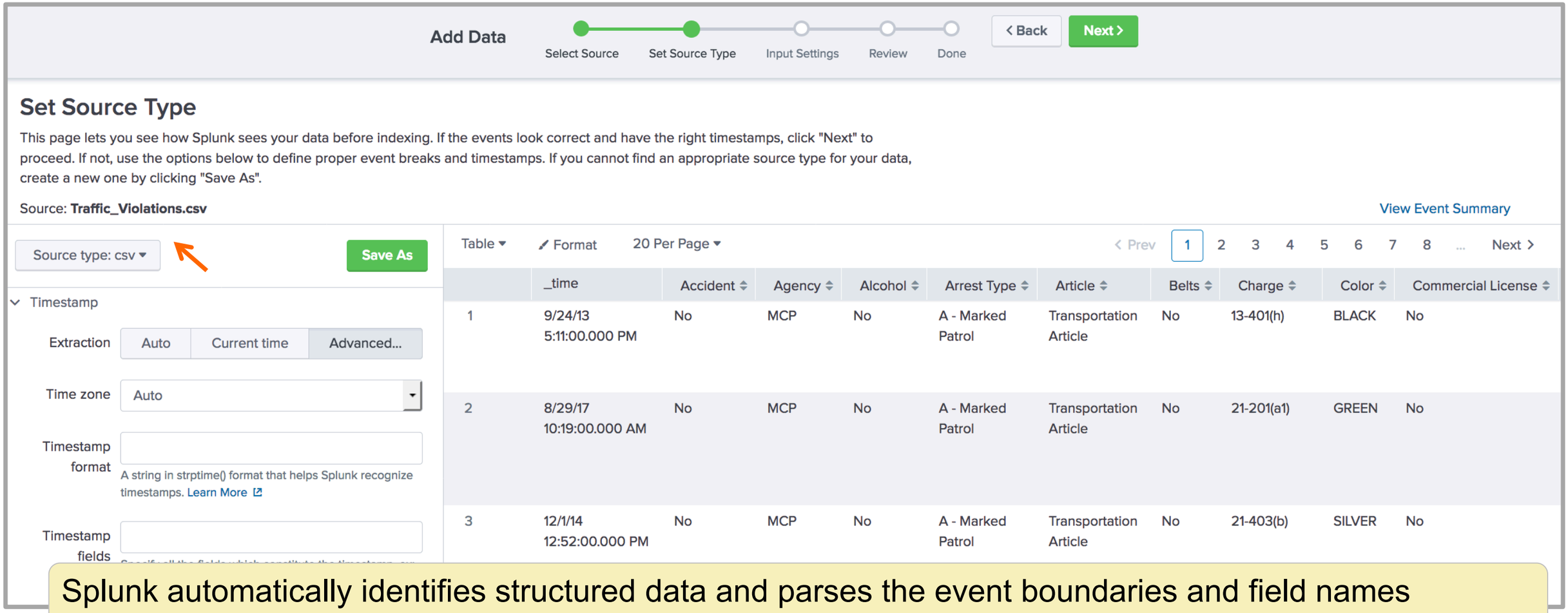
In order to parse the event correctly, the event pattern prefix needs to be specified to format the incoming data

Previewing Unstructured Data (cont.)



Click **Timestamp > Advanced** to access time zone, timestamp prefix, and timestamp definitions needed to extract the correct time from the data

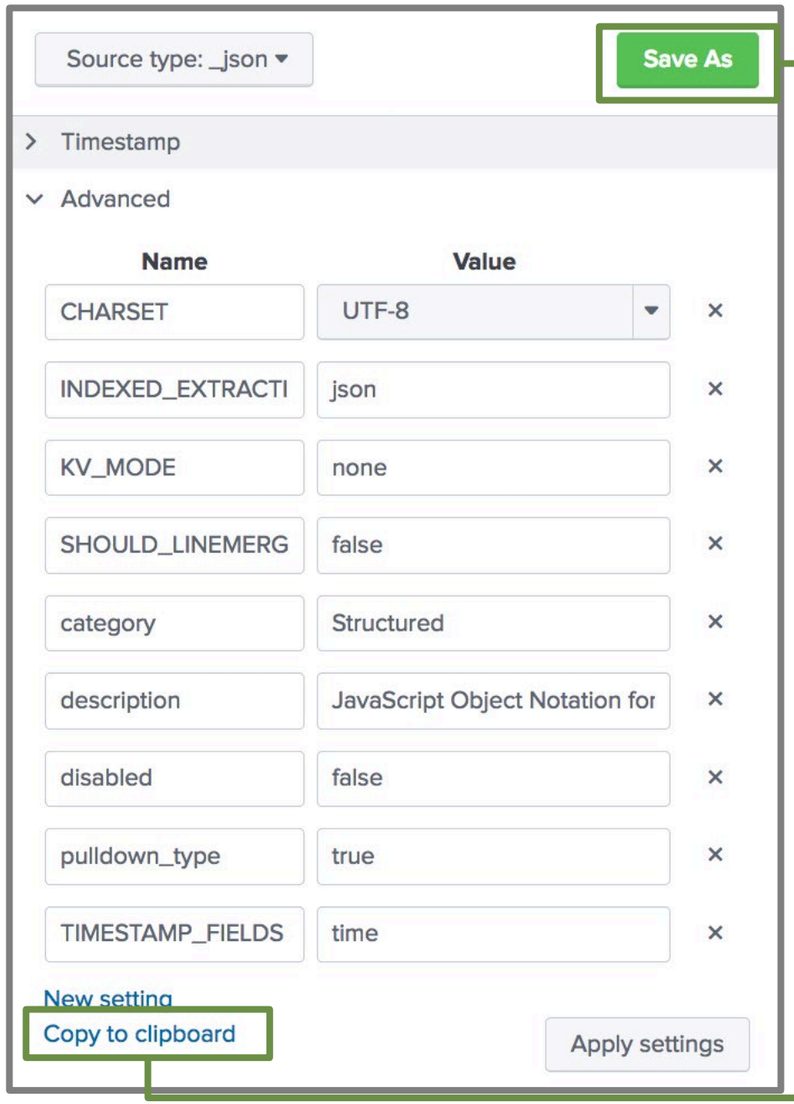
Previewing Structured Data



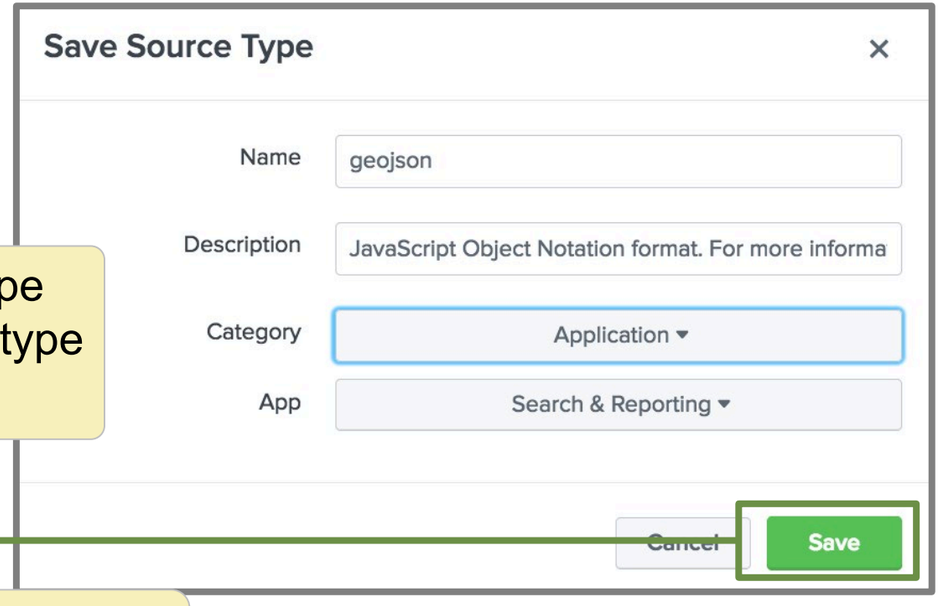
Splunk automatically identifies structured data and parses the event boundaries and field names

- 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

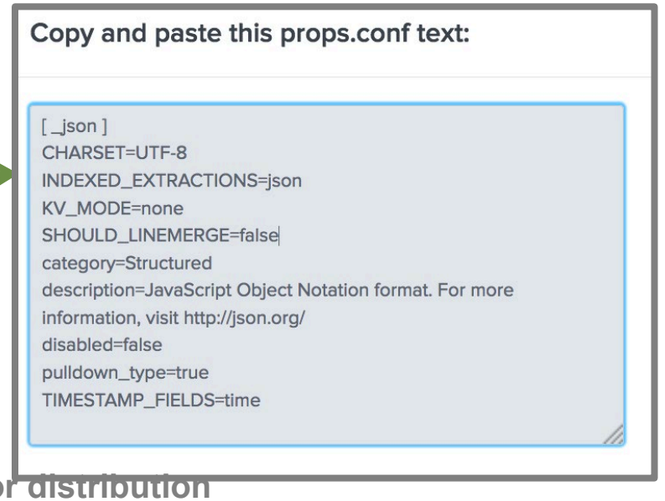


When saved, the source type becomes a custom source type that you can re-use



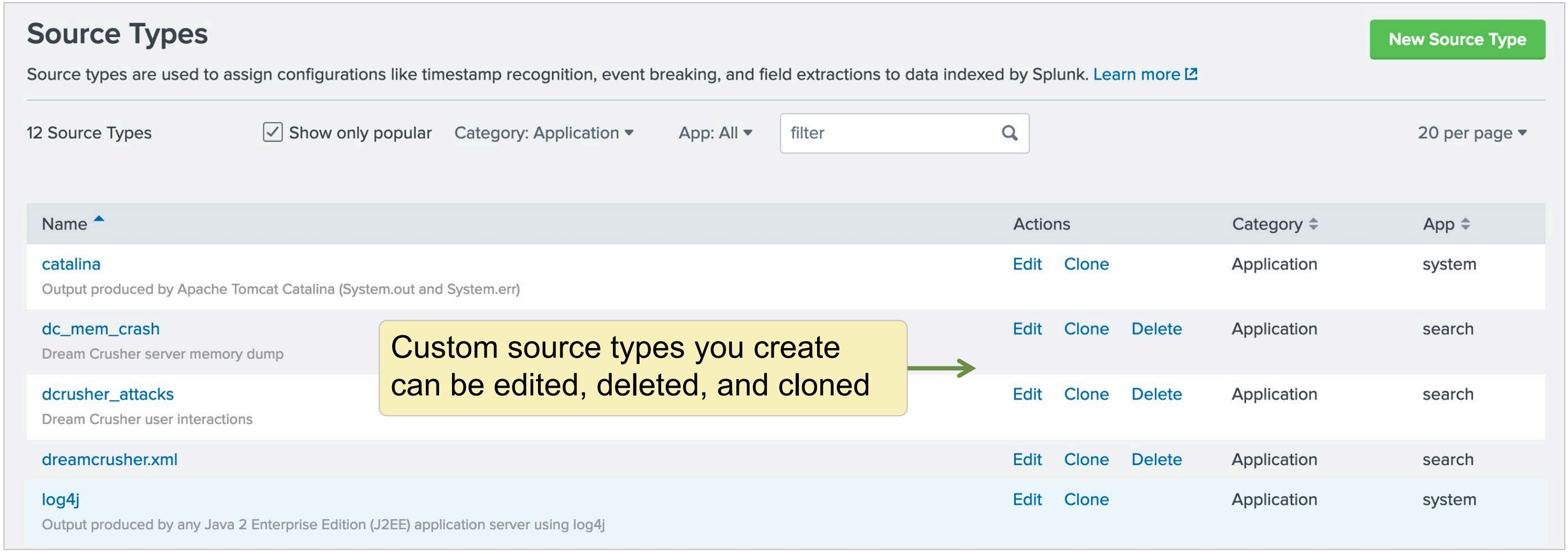
Or, check the resulting **`props.conf`** stanza for your new source type

You can copy and deploy sourcetype settings manually to your forwarders



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.

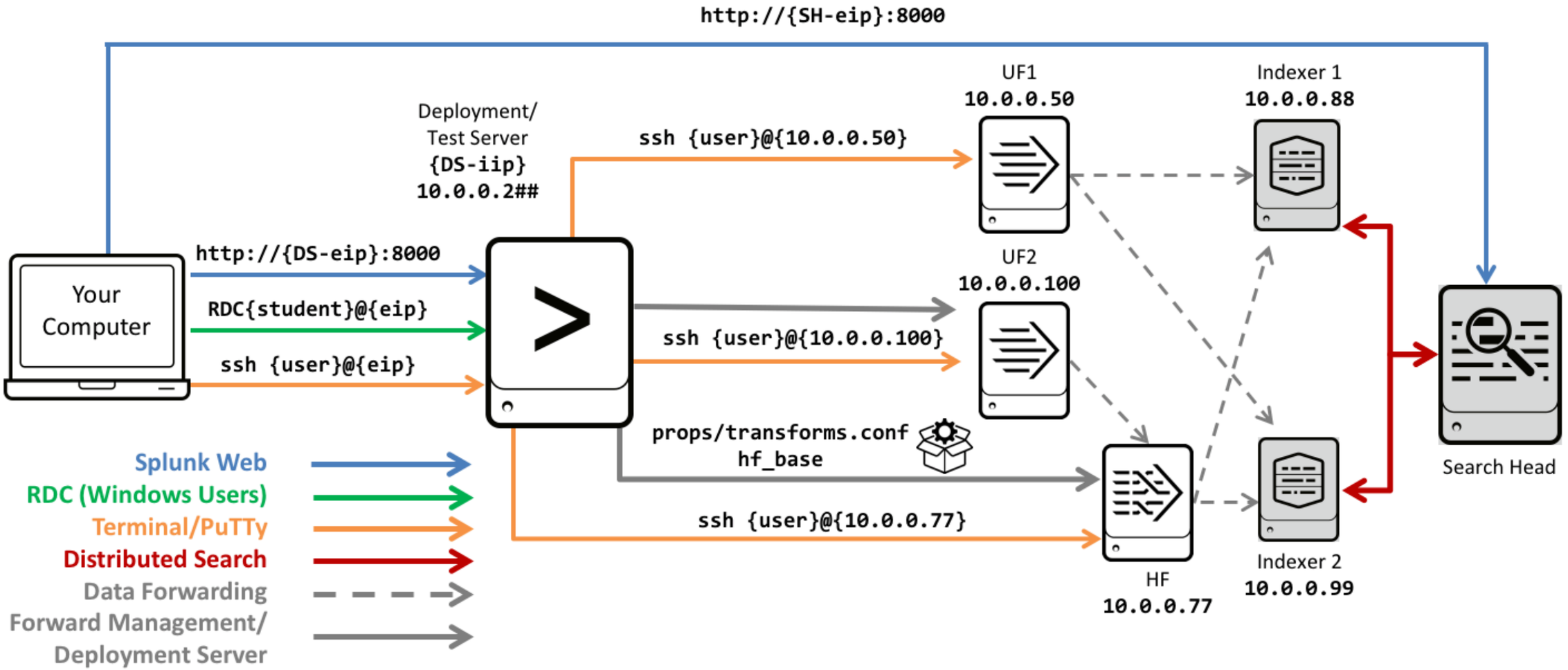
- True or False. Event boundaries can be defined using **`props.conf`** at the UF.

  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 it's necessary to modify the underlying raw data before it is indexed
- Examples:
  - The case of privacy concerns:
    - Patient information in a healthcare environment
    - Credit card or account numbers in a financial environment
    - Data being transported across international boundaries
  - Event routing according to business use cases (e.g. audit and security):
    - All events go to the **web** index, except credit card transactions which are sent to the **credits** index
- Care should be taken when modifying raw events (**`_raw`**)
  - Unlike all other modifications discussed, these change the raw data 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`**
    - Uses only **`props.conf`**
    - Only used to mask or truncate raw data
  - **`TRANSFORMS`**
    - Uses **`props.conf`** and **`transforms.conf`**
    - More flexible
    - Transforms matching events based on source, source type, or host

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 the first 5 digits of an account number in the **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

Match this and replace with **AcctID=xxxxx99288**

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

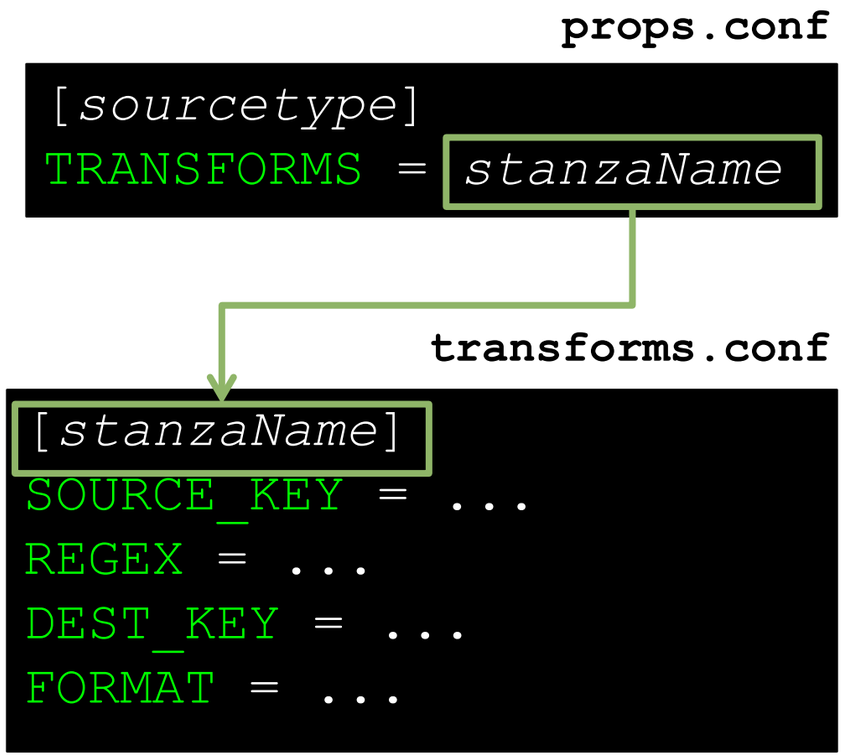
props.conf

Indicates the capture group

- For more examples, see:
  http://docs.splunk.com/Documentation/Splunk/latest/Data/Anonymizedata

TRANSFORMS

- Per event transformation is based on REGEX pattern matches
- Define the transformation in **transforms.conf** and invoke it 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
```

transforms.conf

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

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

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=790504524226513 5
[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 event 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 = (?!)^EventCode=(592|593)
DEST_KEY = queue
FORMAT = nullQueue
```

The **`(?!)`** 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

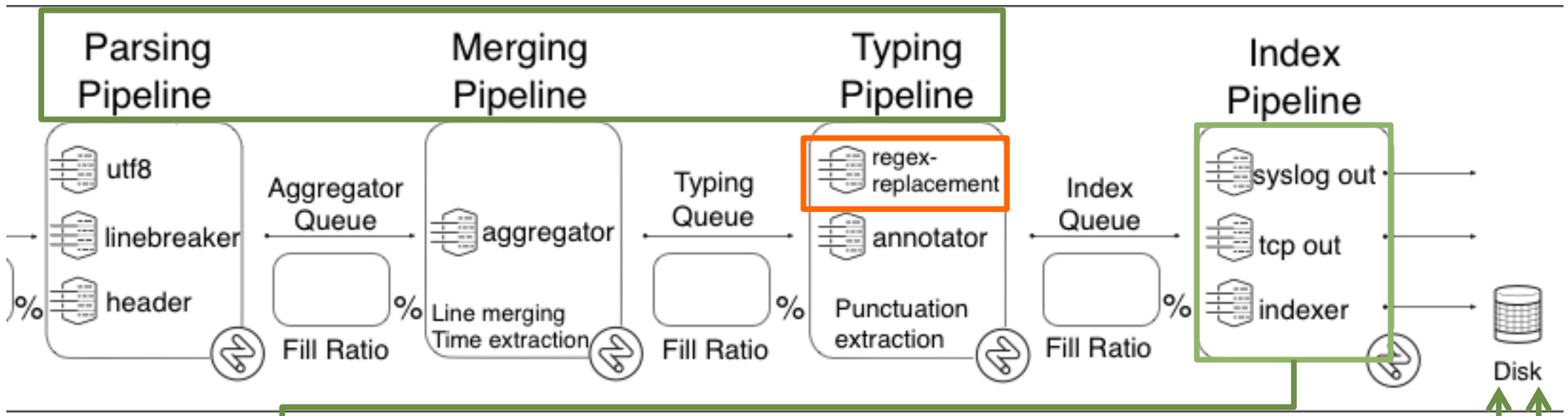
```
[errorRouting]
REGEX = error
DEST_KEY=_TCP_ROUTING
FORMAT = errorGroup

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

outputs.conf

```
[tcpout]
defaultGroup=everythingElseGroup

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

[tcpout:errorGroup]
server=10.1.1.200:9999

[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



End of the parsing
- Null-routed?
- Remote server?
- Disk?

license meter
`_raw` is metered for license usage

index created
Keyword indexed, **`_raw`** is compressed and both are written to disk

Data Integrity Control

Persisted to Disk

- All modifications and extractions are written to disk along with **`_raw`** and metadata
  - source, sourcetype, host, timestamp, punct, etc.
- Indexed data cannot be changed
  - Changes to **`props.conf`** or **`transforms.conf`** only apply to new data
  - Indexed data cannot be changed without re-indexing
- **Tip**:
  - When adding or changing stanzas in **`props.conf`**, you can call the following URL endpoint to re-load the modified **`props.conf`** and **`transforms.conf`** without restarting your indexer:

  http://*servername:splunkwebport*/debug/refresh

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.

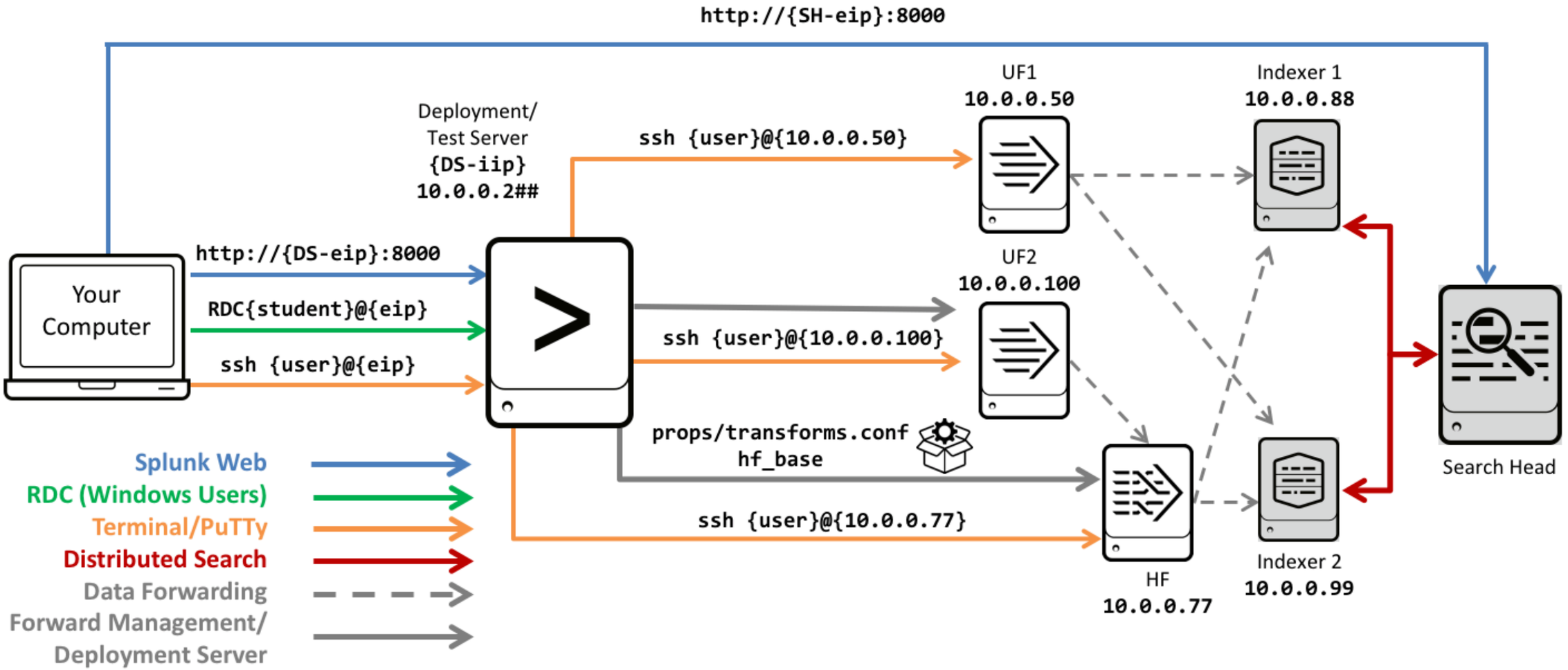
  True. If you do not specify the **SOURCE_KEY** in **transforms.conf**, it defaults to **_raw**.

- 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: 20 – 25 minutes

Tasks:

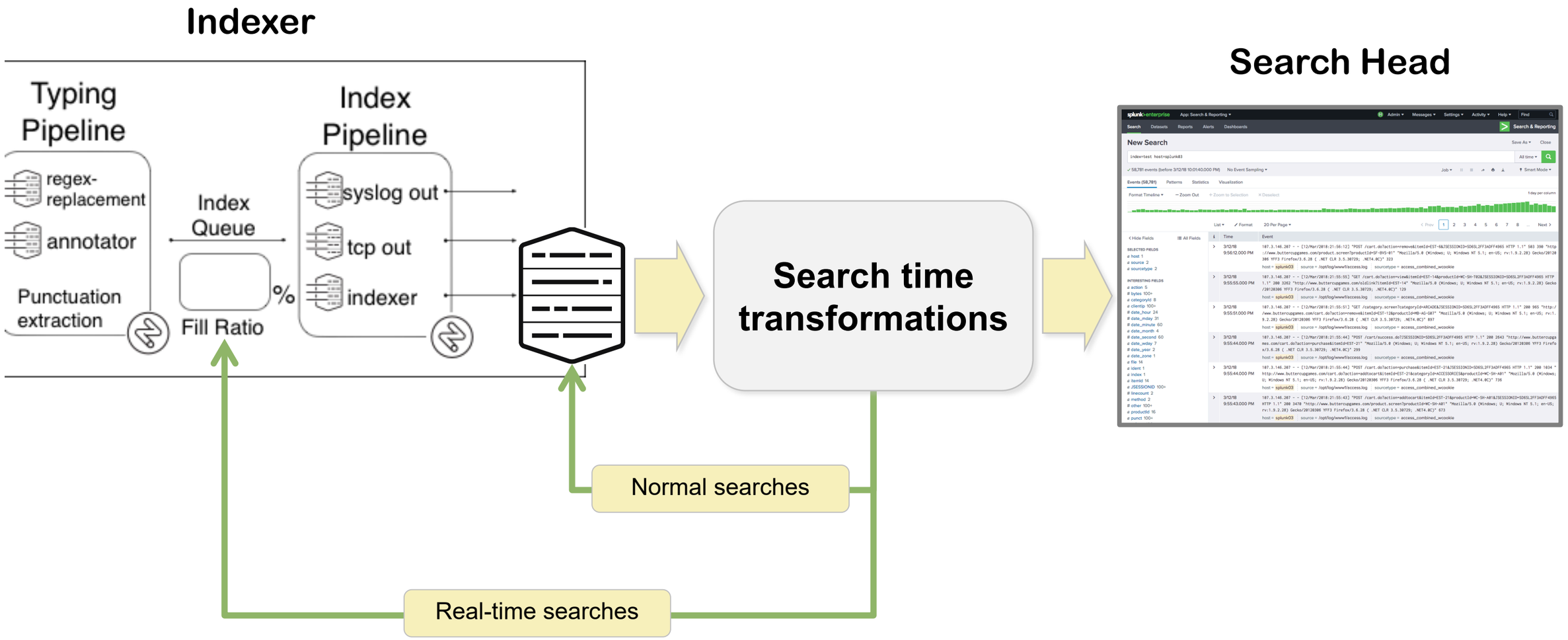
- Use **`transforms.conf`** to:
  - Mask sensitive data
  - Redirect events to specific indexes
  - Drop unwanted events
- Use **`props.conf`** to sequence the filtering and redirecting events

Module 11:
KO Administration

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

- When Splunk parses and indexes data, it adds (extracts) a number of fields to the event data that are stored in the index
  - Default fields are added automatically by Splunk to each event
  - Custom fields are fields that are specified by the data admin
- Generally, fields should be extracted at search time, however there are certain use cases when index time field extractions can be used
  - On the forwarder for structured inputs
  - On the indexer for fields that may be negatively impacting search performance
- Do not add custom fields to the set of default fields that splunk extracts and indexes (**`host`**, **`_time`**, **`source`**, **`sourcetype`**, etc.) at index time unless absolutely necessary
  - Can negatively impact indexing performance and search times
  - Increases the size of the searchable index

Pros/Cons of Index Time Field Extractions

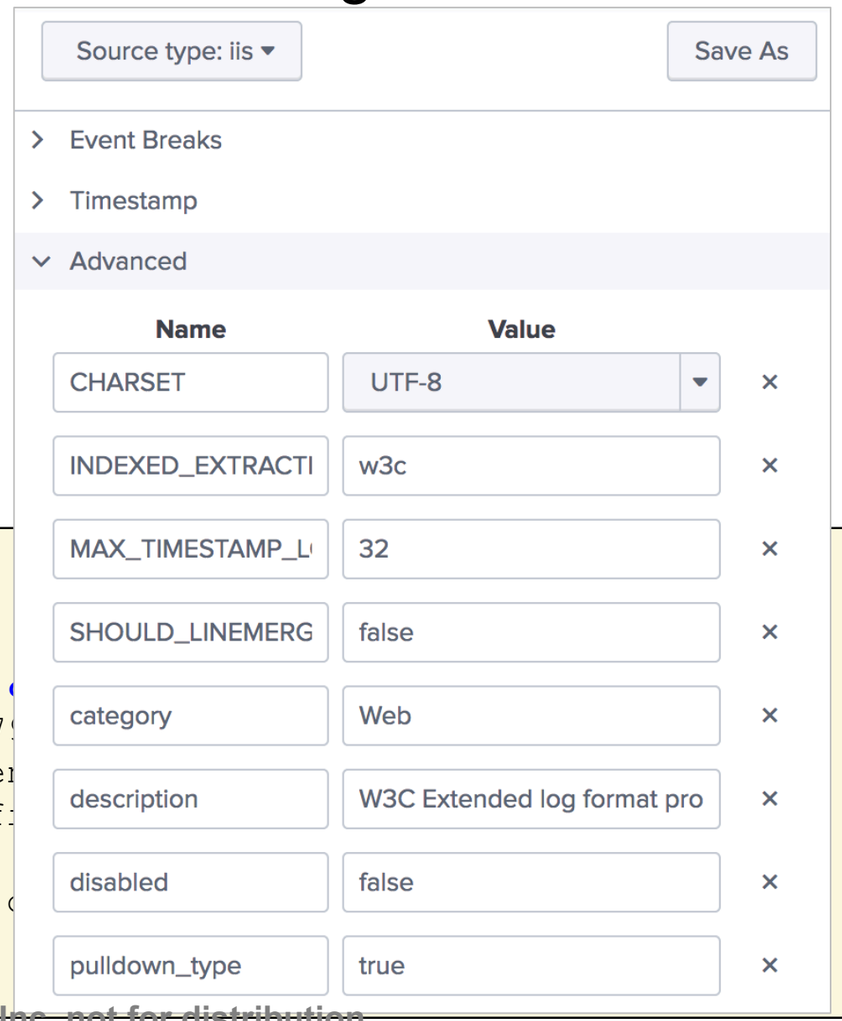
PROs	CONs
• Provision the extraction during the **input** or **parsing** phase • Can configure on the universal forwarder • Auto-formatting • Can drop useless headers and comments	• Increased storage size (2x -5x the original size consumed on the indexer) • Static field names – additional step required 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

- Again, 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
2015-01-08 00:00:00 POST AutoComplete.asmx/GetCompletionList - 10.175.16.7
cApproved=1;+fParticipant=0000000695607440|urn:System-Services:GatewayToke
format:persistent|http://www.acme.com/2015/06/attributes/credentialidentif
4dfe-bf45-fc2df5;+style=normal
https://search.acme.com/Account/Account.aspx?redirect=https://direct.acme.
0
...
```

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

Configuring Indexed Field Extractions

- Define additional attributes in **`props.conf`**, **`transforms.conf`**, and fields in **`fields.conf`**
  - In a distributed environment, **`props.conf`** and **`transforms.conf`** are deployed to the indexers (search peers) or HF (if being used)
  - Deploy **`fields.conf`** changes to the search head

`props.conf`

```
[testlog]
TRANSFORMS-netscreen = netscreen-error
```

`fields.conf`

```
[error_code]
INDEXED=true
```

`transforms.conf`

```
[netscreen-error]
REGEX =  device_id=\[\w+\](?<err_code>[^:]+)
FORMAT = err_code::"$1"
WRITE_META = true
```


Default Search Time Field Extractions

- For common source types, Splunk has default search time field extractions
- Fields can be discovered by Splunk from your search results
  - Automatically detects key/value pairs (e.g. **`a=1`**)
- Additional default extractions are easy to add with add-ons and apps
  - The *nix app has many search time fields for standard UNIX logs
    - For example, **`secure.log`**, **`messages.log`**, etc.
  - The Windows app has similar defaults for Windows data
  - For other data, look for an app specifically designed for that type of data on **splunkbase.splunk.com**

Custom Search Time Field Extractions

- Use the **`rex`** command, or similar commands, in the search language
  - All roles can use this command
  - Requires knowledge of regular expressions (REGEX)
- Use the Field Extractor in Splunk Web
  - Handles REGEX-based and delimiter-based extractions
  - Knowledge of regular expressions helpful, but not required
- Edit configuration files
  - Available only to admins and provides additional advanced extraction options
  - Knowledge of REGEX required

Field Extractions in `props.conf`

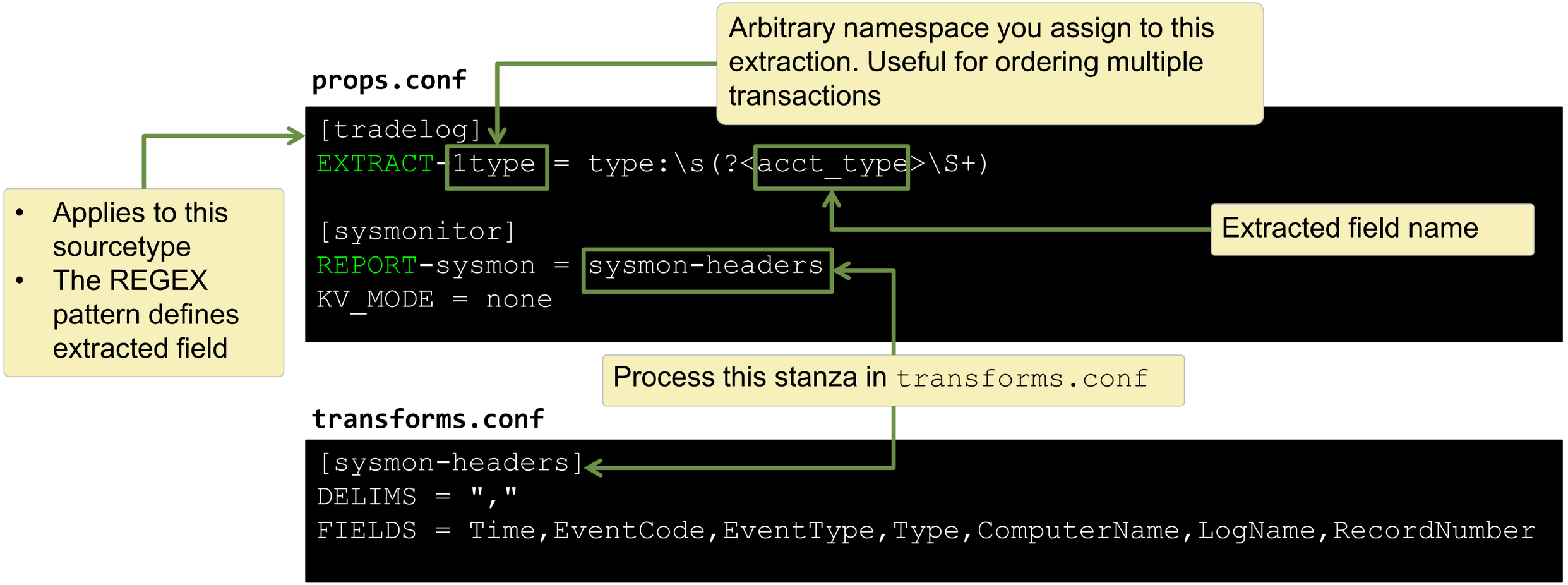
- Field extraction happens during index-time (indexed fields) and/or search-time (extracted fields)
- The search-time extractions can be an inline or a transform
- Use extraction directives, **`EXTRACT`** and **`REPORT`**, in **`props.conf`**
  - **`EXTRACT`** (inline extraction) is defined in **props.conf** as single field extraction
  - **`REPORT`** (field transform) is defined in **`transforms.conf`** and 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:

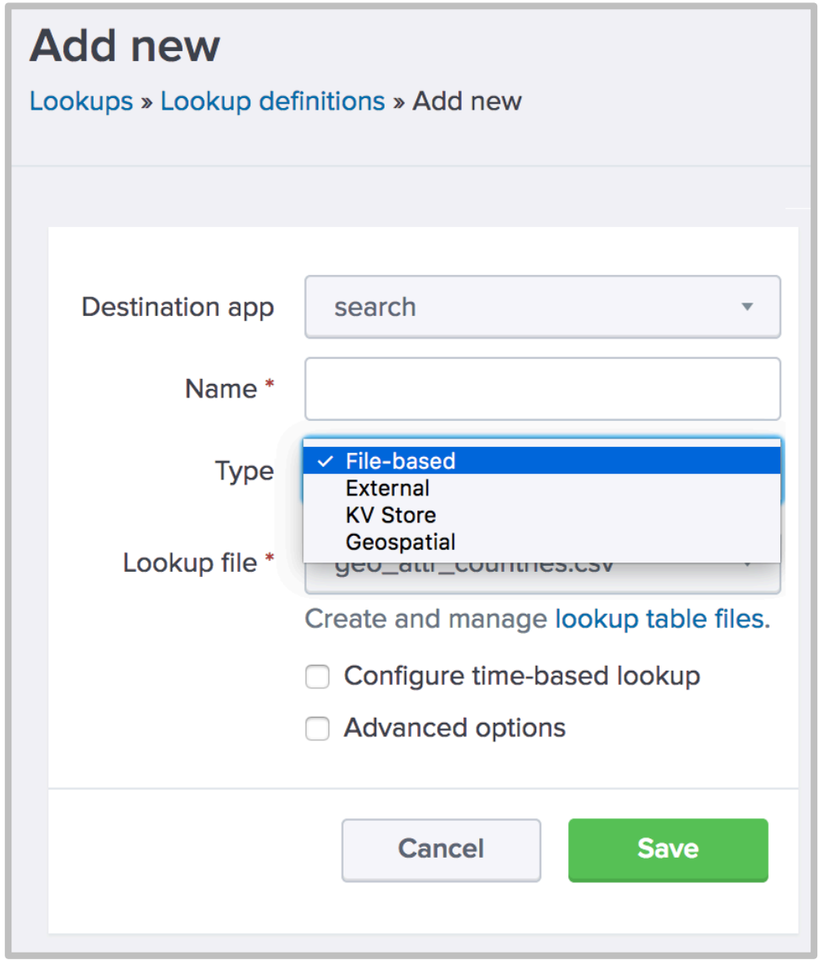
docs.splunk.com/Documentation/Splunk/latest/Knowledge/Createandmaintainsearch-timefieldextractionsthroughconfigurationfiles

Extract and Report in `props.conf`



Lookups

- A lookup is a Splunk data enrichment knowledge object
  - Used ONLY during search time
  - The lookup stanzas are defined in **`transforms.conf`** and **`props.conf`**
- Four types:
  - **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

- Knowledge objects are stored in configuration files:
  - **`macros.conf`**, **`tags.conf`**, **`eventtypes.conf`**, **`savedsearches.conf`**, etc.
  - See specific **`.spec`** files in **`SPLUNK_HOME/etc/system/README`** and the docs for details
- When users create or modify knowledge objects, Splunk Web automatically updates the **`.conf`** files
- Use Splunk Web UI as much as possible
  - Admins can use **`btool`** and edit the **`.conf`** files directly
- Some system settings can be checked and changed with **Advanced edit**

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
Splunk errors last 24 hours		None	None	No owner	search	0	App | Permissions	Enabled | Disable	Run | Advanced edit | Clone
Messages by minute last 3 hours		None	report_builder_display	No owner	search	0	App | Permissions	Enabled | Disable	Run | Advanced edit | Clone

Managing Orphaned Knowledge Objects

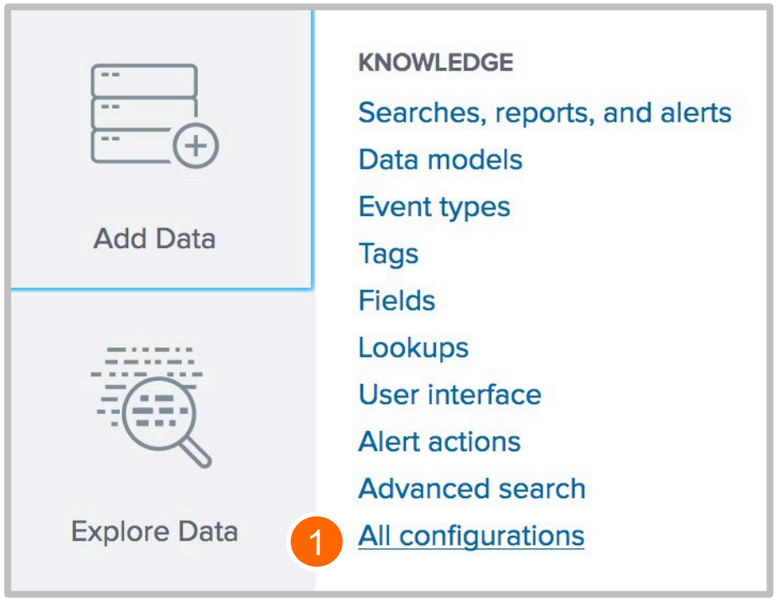
- What are orphaned knowledge objects?
  - Knowledge objects without a valid owner
  - Occurs when a Splunk account is deactivated and the knowledge objects associated with that account remain in the system
- Can cause performance problems and security concerns
  - Searches that refer to an orphaned lookup may not work
  - Search scheduler cannot run a report on behalf of a nonexistent owner

Locating Orphaned Knowledge Objects

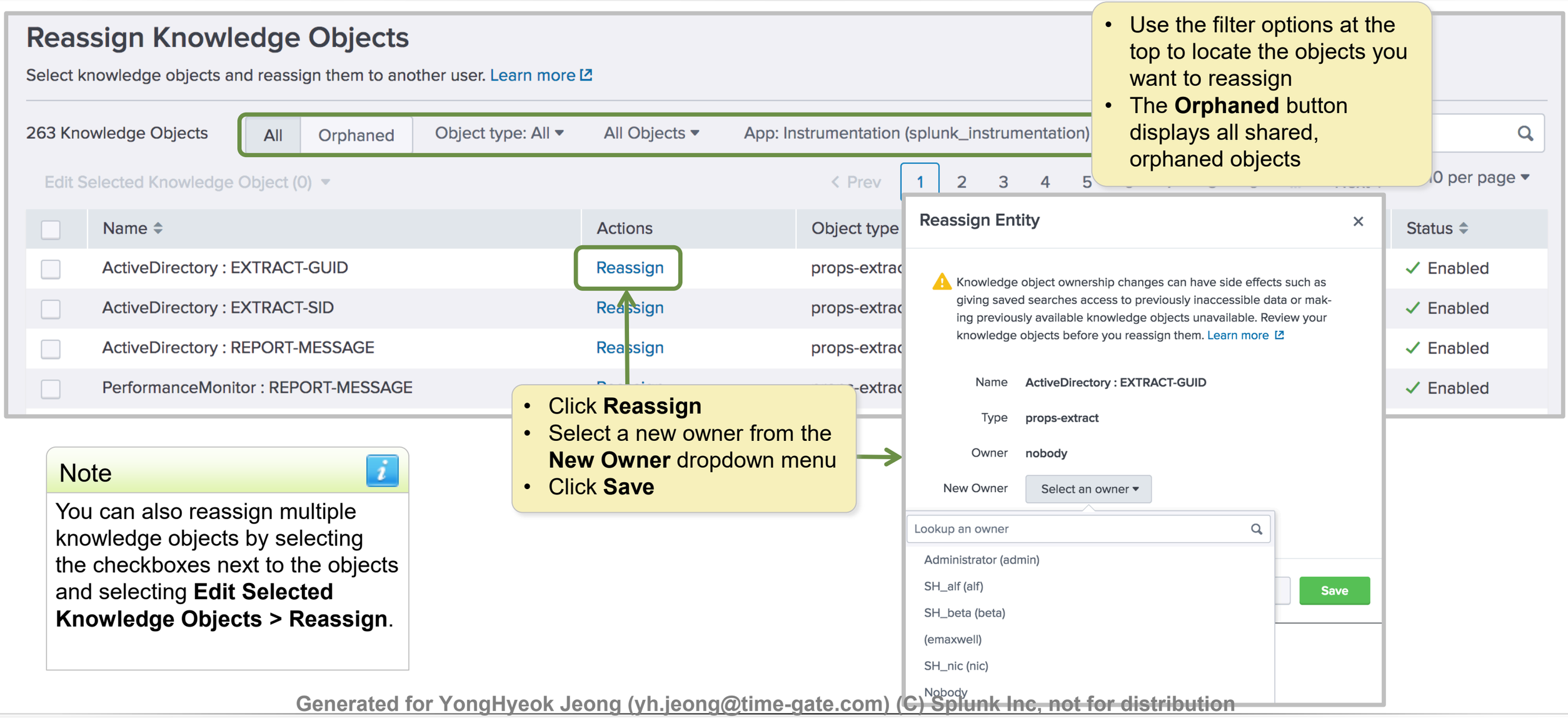
- Splunk runs a default search on a daily schedule to detect orphaned scheduled reports
- Click **Messages**, then click the message link to access the alerts dashboard
  - You can also run the search from **Search > Dashboards > Orphaned Scheduled Searches, Reports, Alerts**
  - The MC Health Check also has a search to detect orphaned knowledge objects

Reassigning Knowledge Objects

- Users with the Admin role can reassign ownership of knowledge objects (orphaned and owned)
  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

- Click **Reassign**
- Select a new owner from the **New Owner** dropdown menu
- 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`**

Module 12: Creating a Splunk Diag

Module Objectives

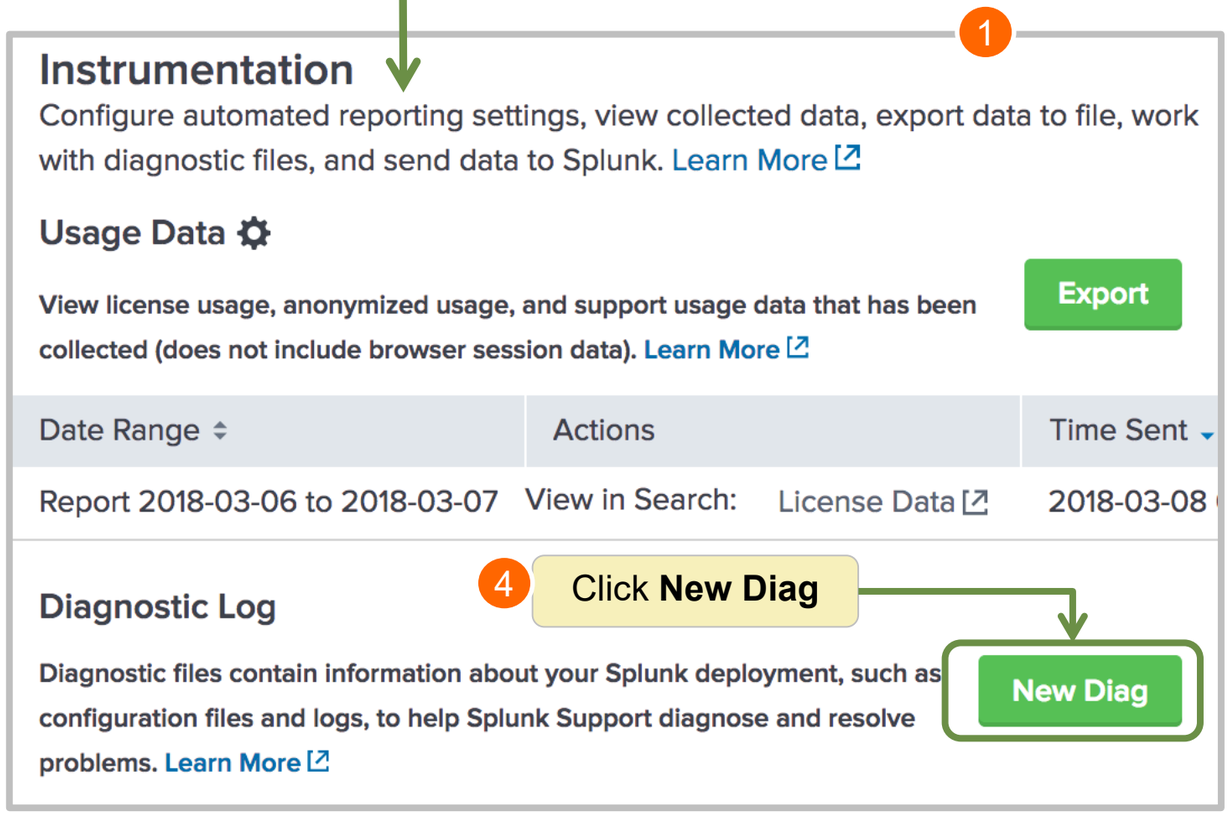
- Describe Splunk diag
- Generate a Splunk diag

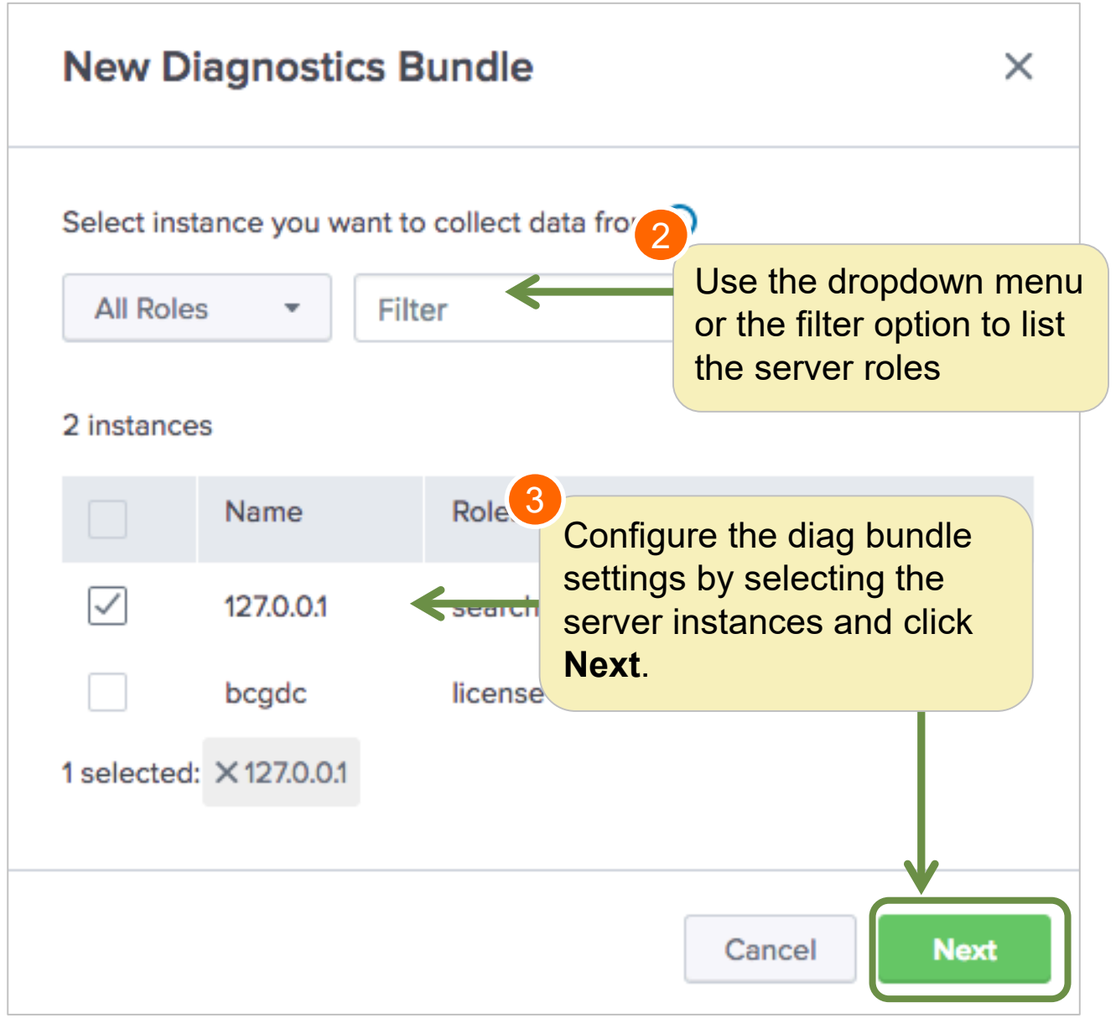
What is a Splunk Diag?

- Diag gathers data and provides insight to your instance:
  - Server specs
    - Configuration, OS version, file system, and current open connections
  - Splunk platform
    - Contents of **`$SPLUNK_HOME/etc`** such as app configurations, Splunk log files, and index metadata
- Produces a **`tar.gz`** file and **`diag.log`**
- No customer data or index data is retrieved, but you should examine the tarball to ensure no proprietary data is included

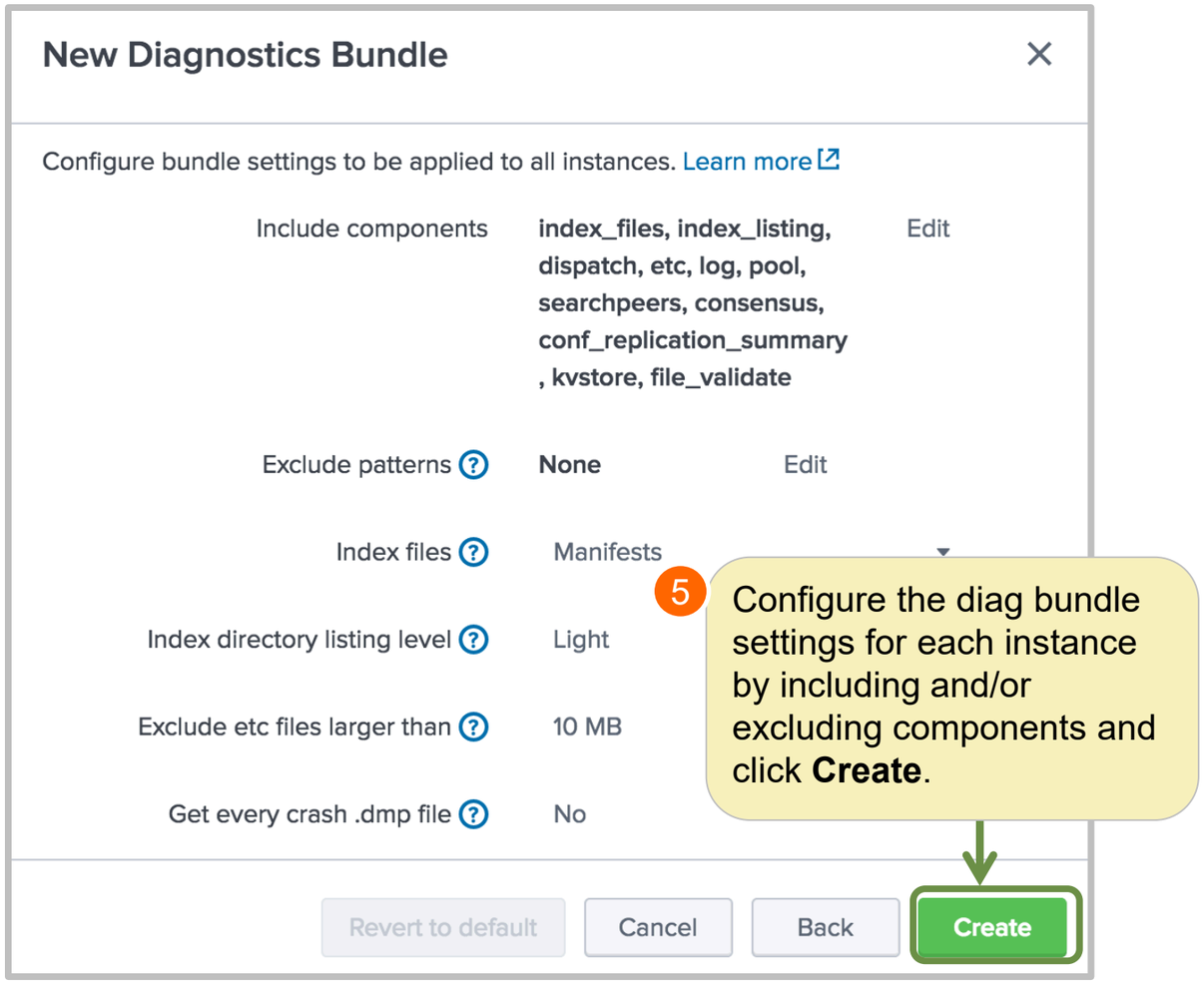
Using the Splunk Diag UI

From Splunk Web, select **Settings > Instrumentation**





Using Splunk Diag UI (cont.)



Diag Example

```
[splunker@ip-10-0-0-201 bin]$ ./splunk diag
Collecting components: conf_replication_summary, consensus, dispatch, etc,
file_validate, index_files, index_listing, kvstore, log, pool, searchpeers
Skipping components: rest
Selected diag name of: diag-ip-10-0-0-201-2016-10-26_18-58-13
Starting splunk diag...
Skippping REST endpoint gathering...
Determining diag-launching user...
Getting version info...
Getting system version info...
Getting file integrity info...
Getting network interface config info...
Getting splunk processes info...
```

Diag reports the components it will collect and the ones it will skip

```
The following certificates were excluded from the diag output
automatically.
  /opt/splunk/etc/apps/framework/contrib/requests/requests/cacert.pem
  /opt/splunk/etc/auth/appsCA.pem
  /opt/splunk/etc/auth/cacert.pem
  /opt/splunk/etc/auth/cloudCA.pem
  /opt/splunk/etc/auth/server.pem
  /opt/splunk/etc/auth/ca.pem
```

Diag also reports certificates that were not auto-detected or skipped

```
Copying Splunk log files...
Copying Search Pool files...
Copying bucket info files...
Copying Splunk dispatch files...
Copying Splunk consensus files...
Adding manifest files...
Cleaning up...
Splunk diagnosis file created: /opt/splunk/diag-ip-10-0-0-201-2016-10-
26_18-58-13.tar.gz
```

When the diag is complete, the output is saved and the file and location are displayed

Module 12 Knowledge Check

- As an admin, you want to look at the contents of the zip file created as a result of executing a diag. How would you check it?

Module 12 Knowledge Check – Answers

- As an admin, you want to look at the contents of the zip file created as a result of executing a diag. How would you check it?

  Splunk it! Ingest the zip file on your test server into a test index.

Module 12 Lab Exercise – Create a Diag

Time: 10 minutes

Task:

Create and index a basic diag file

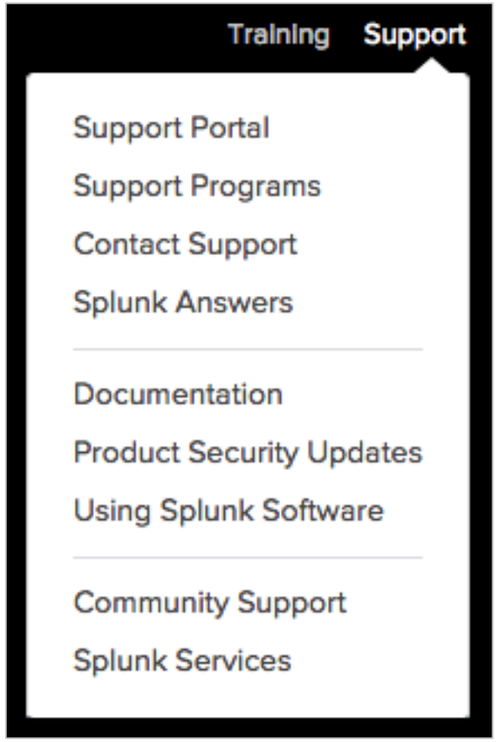
Course Wrap-up

Community

- **Splunk Community Portal**
  splunk.com/en_us/community.html
  - **Splunk Answers**
    answers.splunk.com
  - **Splunkbase**
    splunkbase.com
  - **Splunk Blogs**
    splunk.com/blog/
  - **Splunk Live!**
    splunklive.splunk.com
  - **conf**
    conf.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

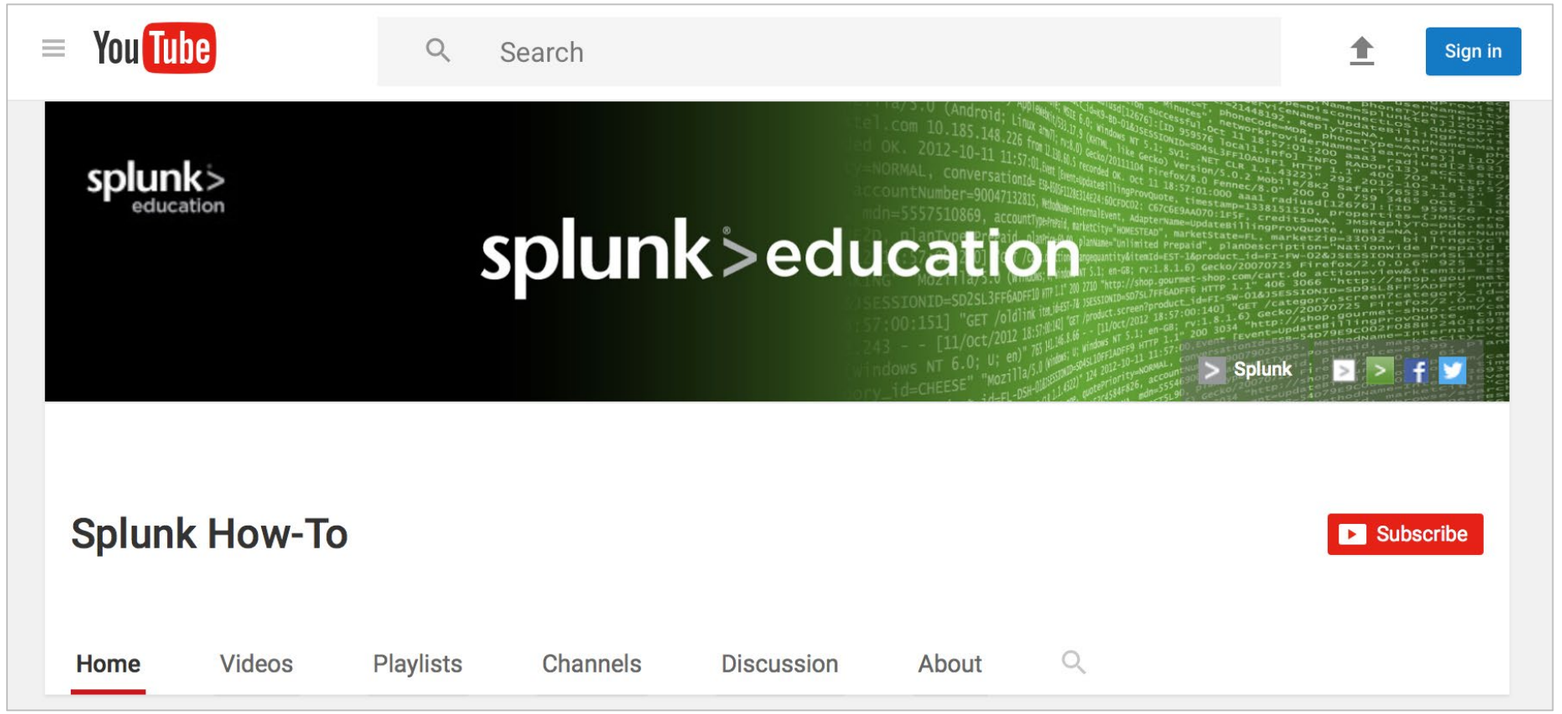
Support Programs

- **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 our roster of training courses, check out the Splunk Education How-To channel: http://www.youtube.com/c/SplunkHowTo
- This site provides useful, short videos on a variety of Splunk topics



What's Next?

- Splunk Certification program

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

- Program information

  https://www.splunk.com/pdfs/training/Splunk-Certification-Handbook-v.8.31.2018.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> .conf19

.conf19
October 21-24, 2019

Splunk University
October 19-21, 2019

Las Vegas, NV
The Venetian Sands Expo

4 Days of Innovation

350 Education Sessions

20 Hours of Networking

sign up for notifications @ conf.splunk.com

Thank You