Splunk Enterprise Data Administration

Document Usage Guidelines

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

Course Prerequisites

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

Course Goals

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

Course Outline

Module 1: Getting Data Into Splunk

Module 2: Configuration Files

Module 3: Forwarder Configuration

Module 4: Forwarder Management

Module 5: Monitor Inputs

Module 6: Network Inputs

Module 7: Scripted Inputs

Module 8: Agentless Inputs

Module 9: Operating System Inputs

Module 10: Fine-tuning Inputs

Module 11: Parsing Phase and Data Preview

Module 12: Manipulating Raw Data

Module 13: Supporting Knowledge Objects

System Administrator versus Data Administrator

Splunk System Administrator

System Management

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

Splunk Data Administrator

Data Onboarding and Management

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

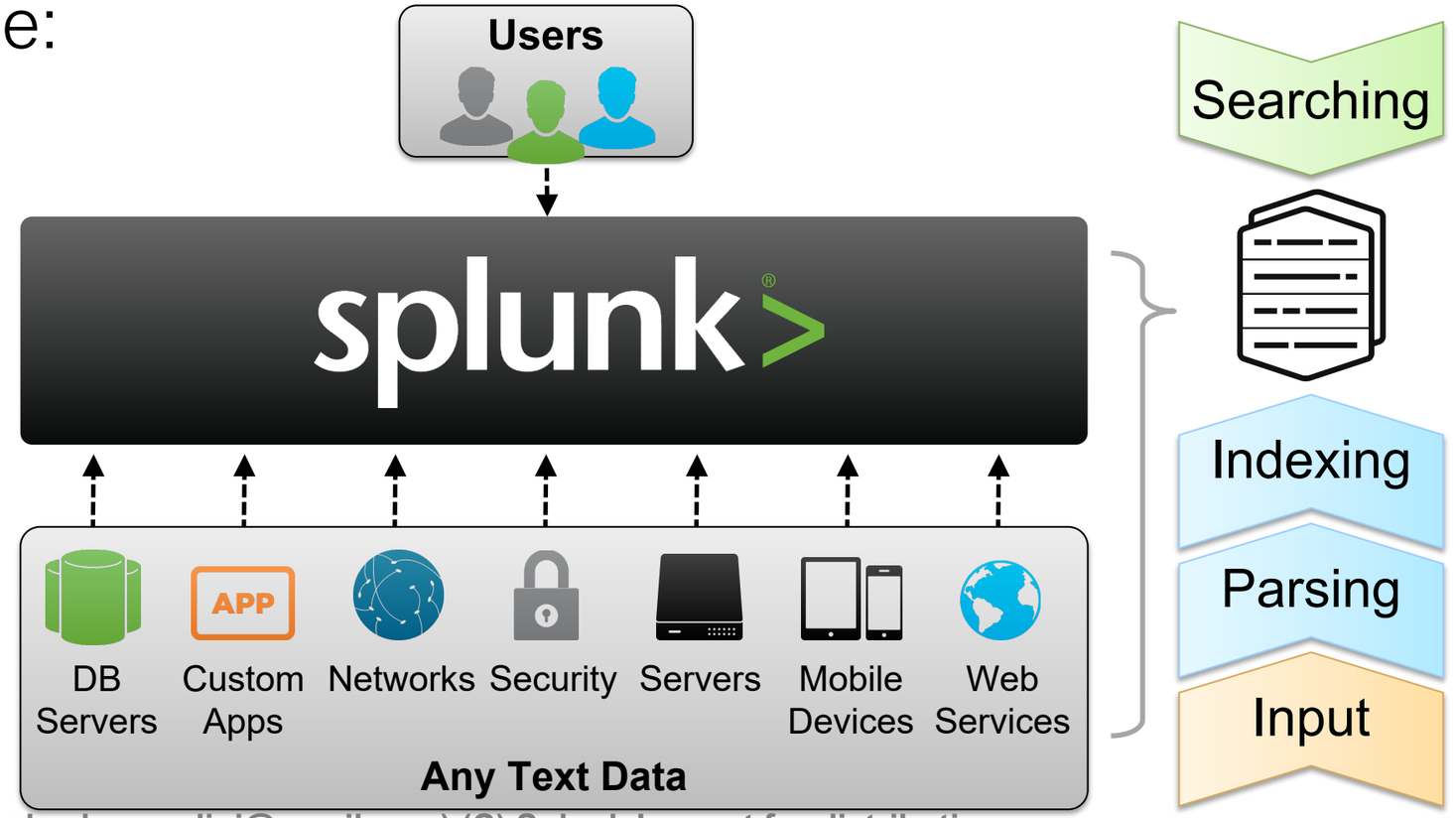
Module 1:
Getting Data Into Splunk

Module Objectives

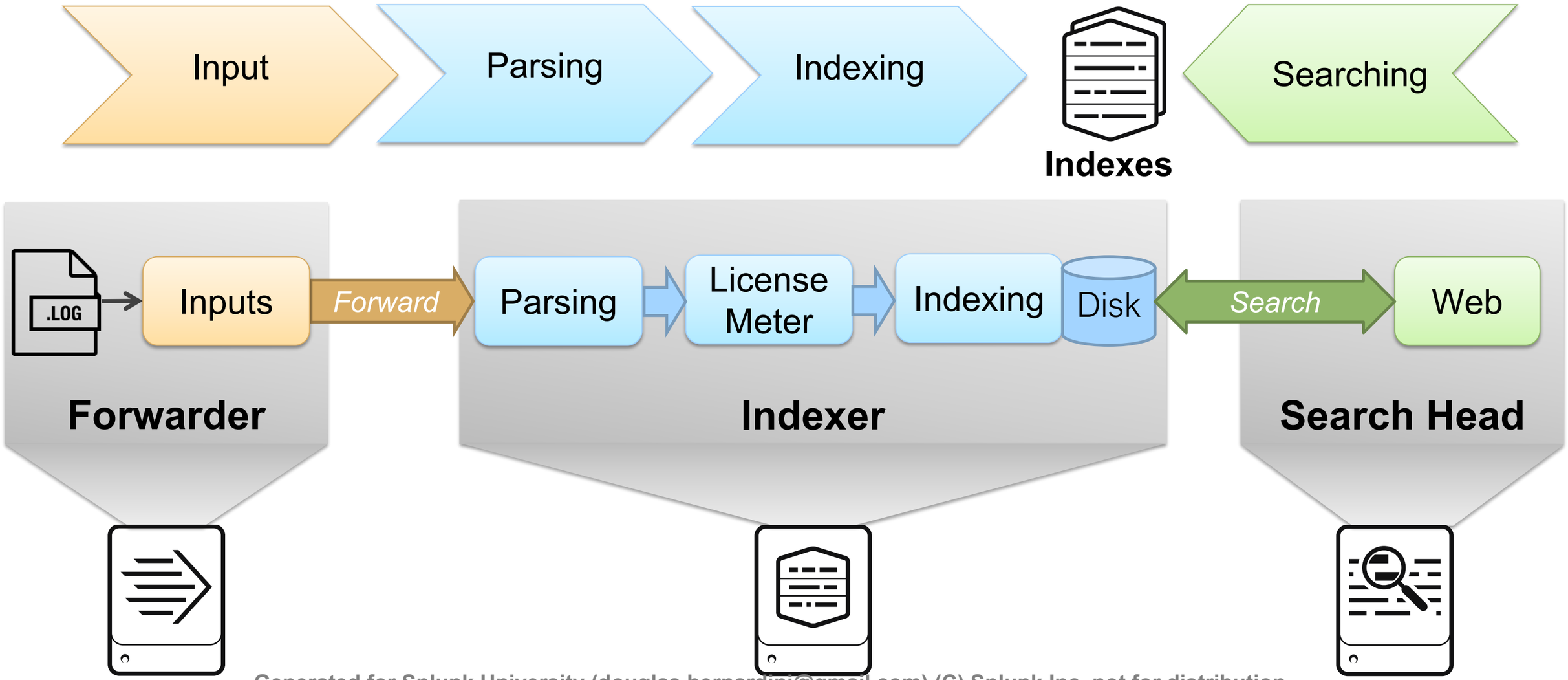
- Provide an overview of Splunk
- Describe the four phases of the distributed model
- Describe data input types and metadata settings
- Configure initial input testing with Splunk Web
- Testing Indexes with Input Staging

Splunk Overview

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

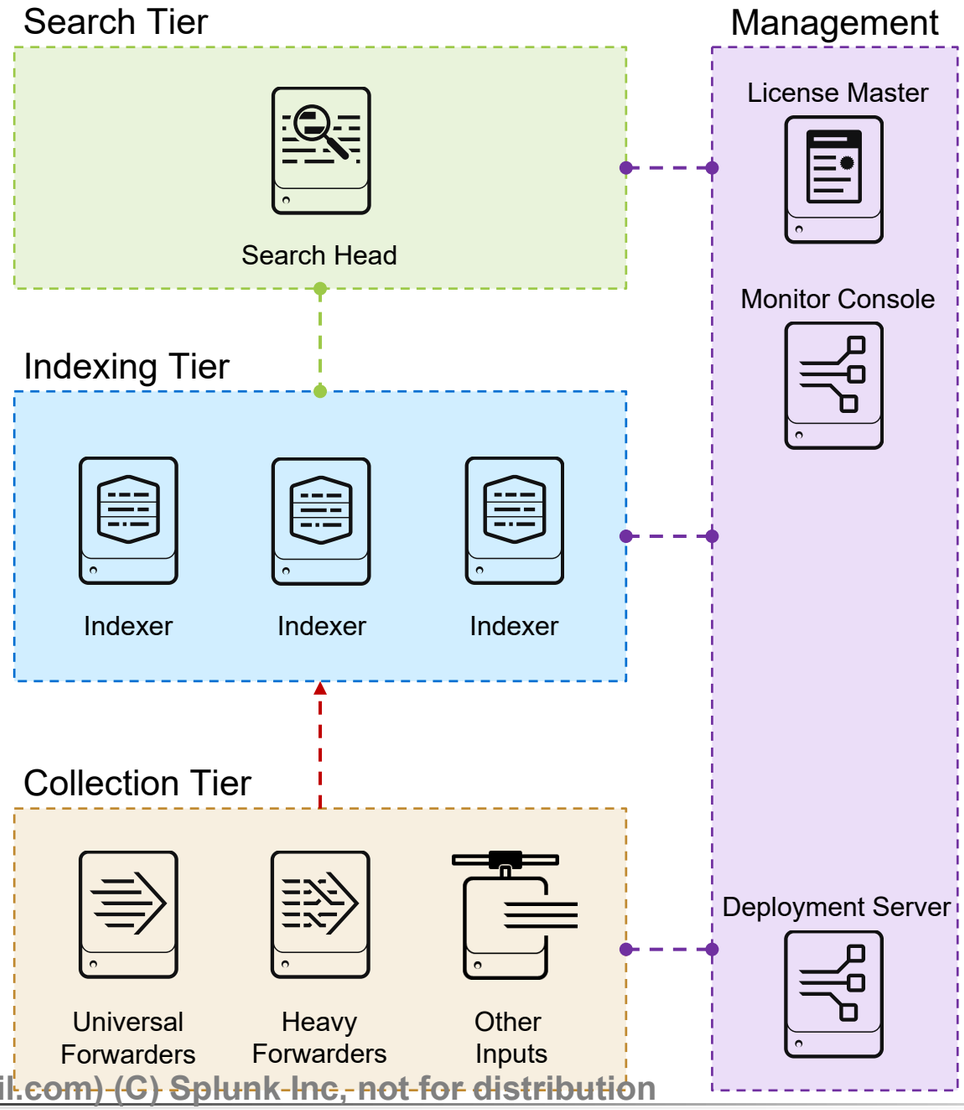


The Four Phases of the Distributed Model



Distributed Non-Cluster Environment

- Scale Splunk in various ways
  - Add indexers to handle more inputs
  - Add indexers and search heads to handle more searching
- Centralize management using dedicated servers including:
  - Deployment server for forwarder management
  - License Master
  - Monitoring Console

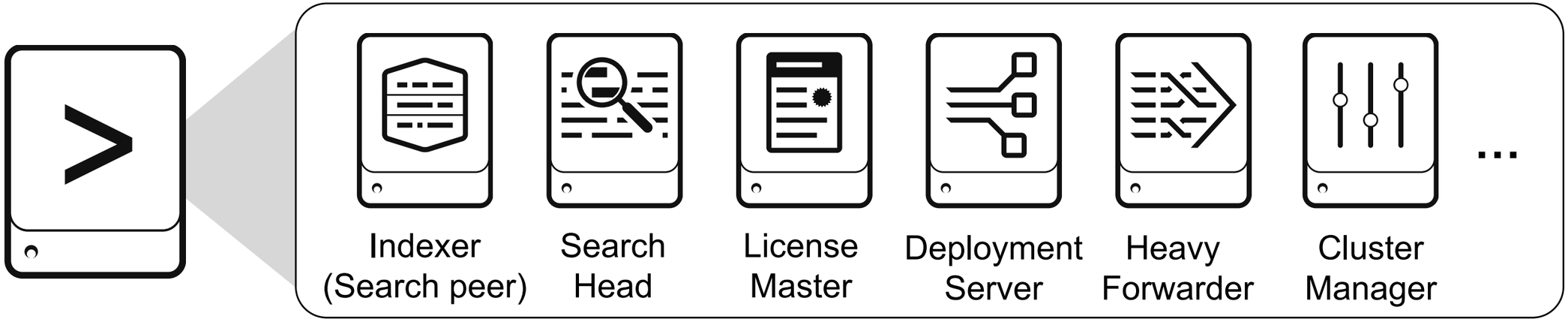


Note

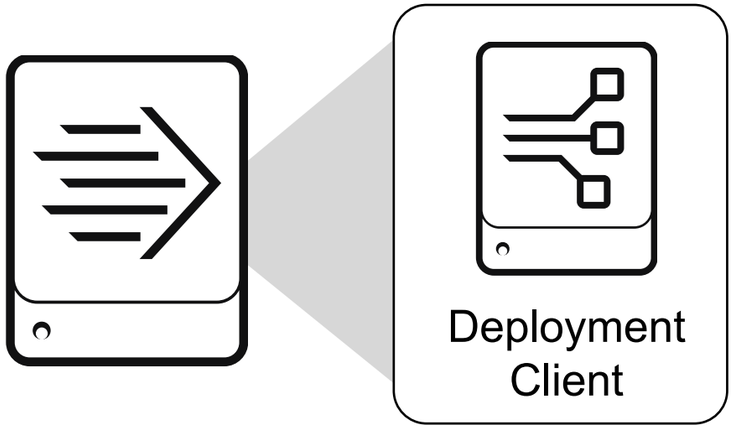
You will configure a Deployment Server and different types of forwarders in later lab exercises.

Software in Splunk Enterprise Packages

Splunk Enterprise package



Universal Forwarder package

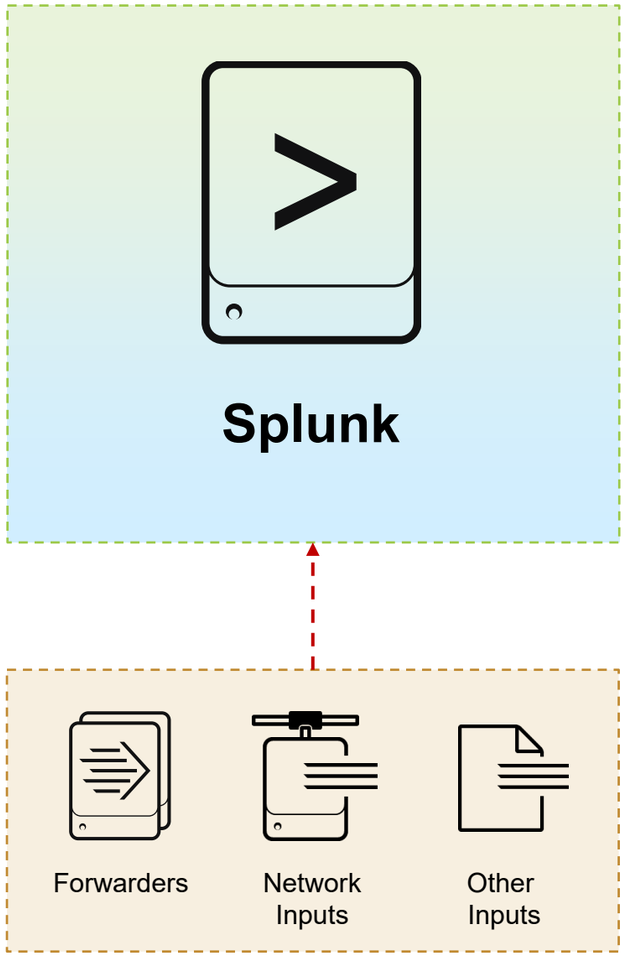


Note

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

Data Input Types

- Supported types of data input
  - **Files and directories**
  - **Network data**
  - **Script output**
  - **Linux and Windows logs**
  - **HTTP**
  - And more...
- You can add data inputs with:
  - Apps and add-ons
  - Splunk Web
  - CLI
  - Editing **`inputs.conf`**



Indexes any text data from any source

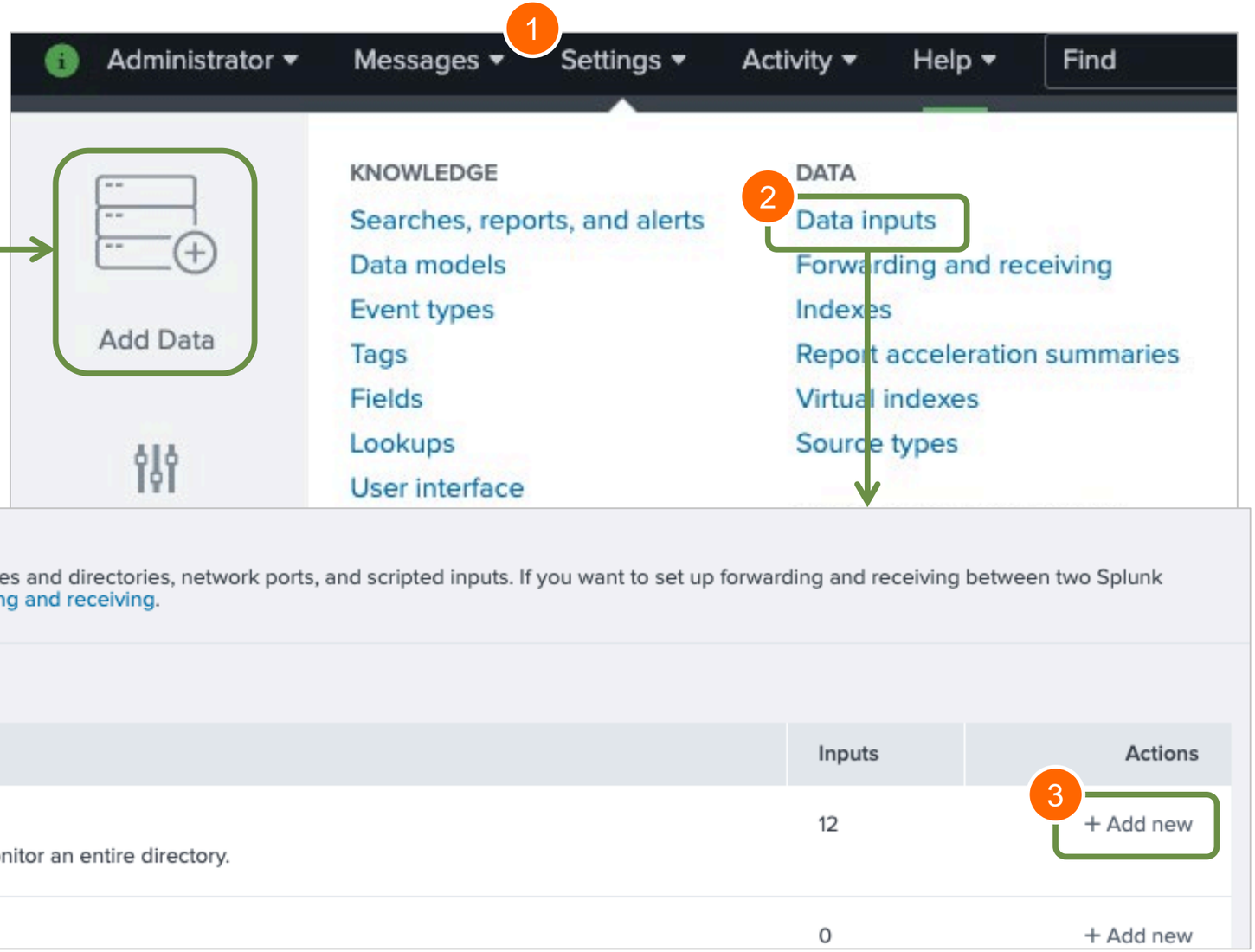
Metadata Settings

- Assigned when Splunk indexes event data
- Generally assigned to entire source during input phase
- Defaults are used if alternates are not specified
  - Overriding values can be performed at input time or later

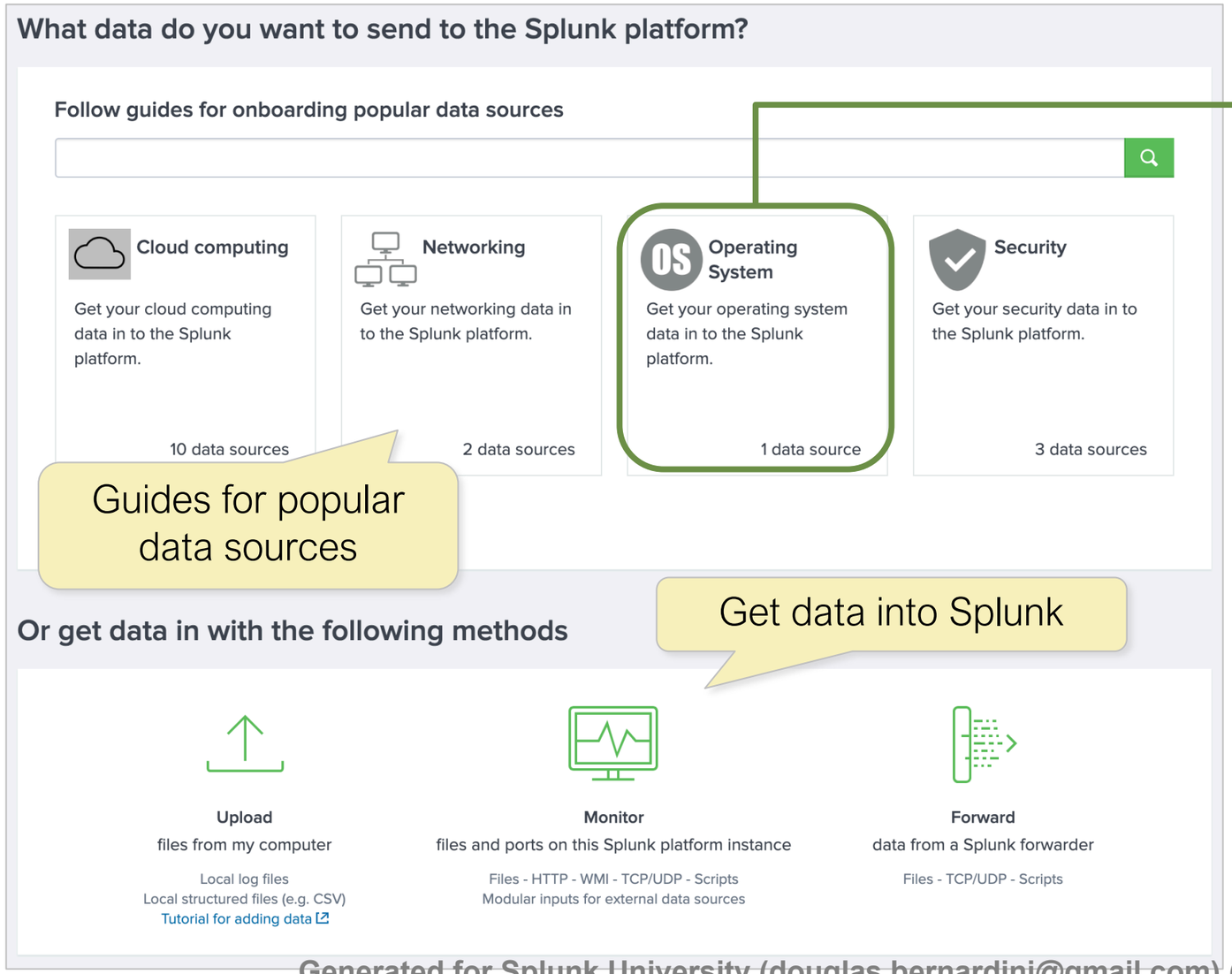
Metadata	Description	Examples
host	Host where an event originates	**websvr1.example.com** **10.0.21.55**
source	Source file, stream or input of an event	**/var/log/messages** **UDP:514**
sourcetype	Format and category of the data input	**access_combined** **cisco_syslog**
index	Where data is stored by Splunk	**main** (default) **itops**

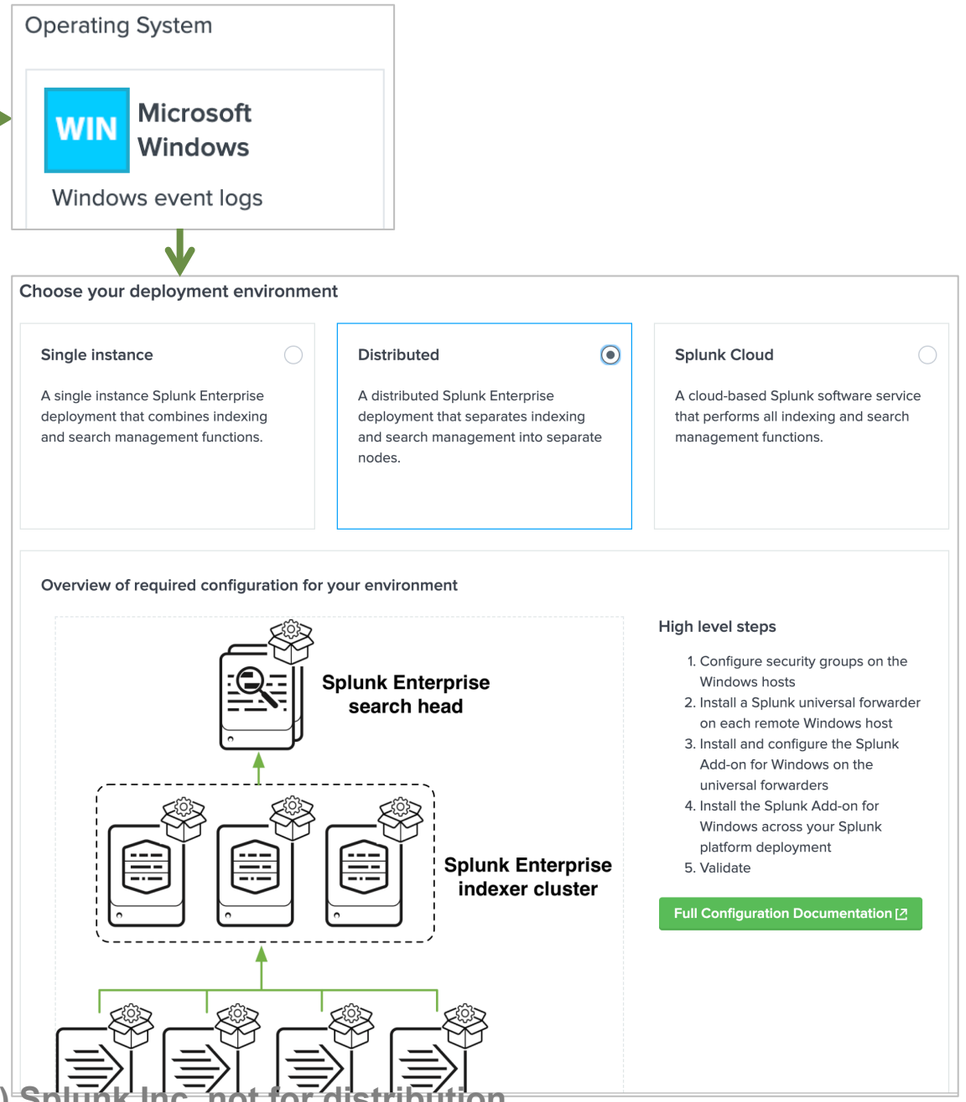
Adding an Input with Splunk Web

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

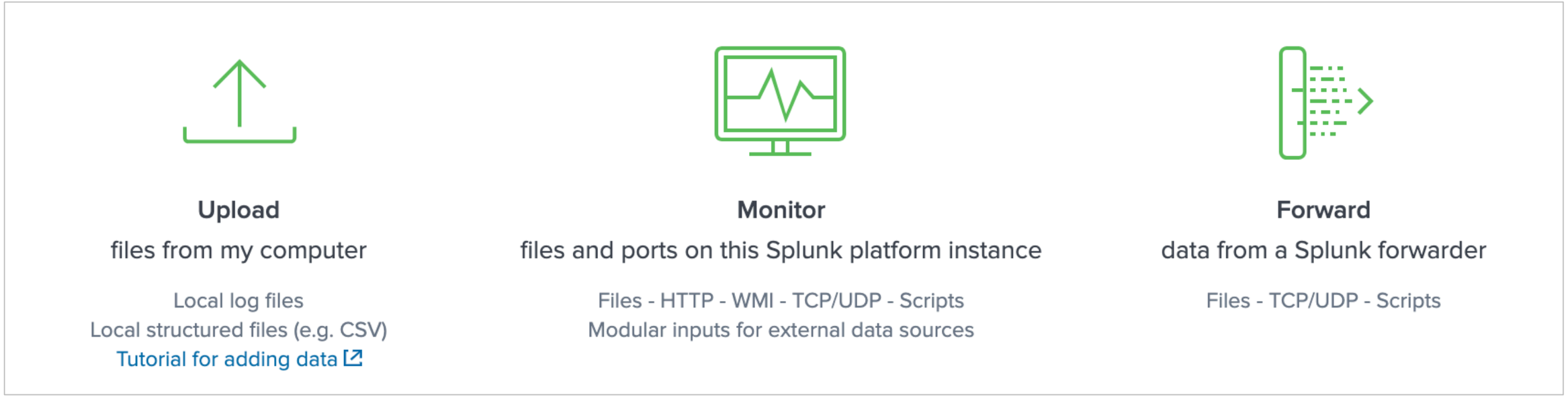


Add Data Menu





Add Data Menu (cont.)



Upload

- Indexed once, for data that never gets updated
- Useful for testing
- File on the local machine
- Does not update **`inputs.conf`**

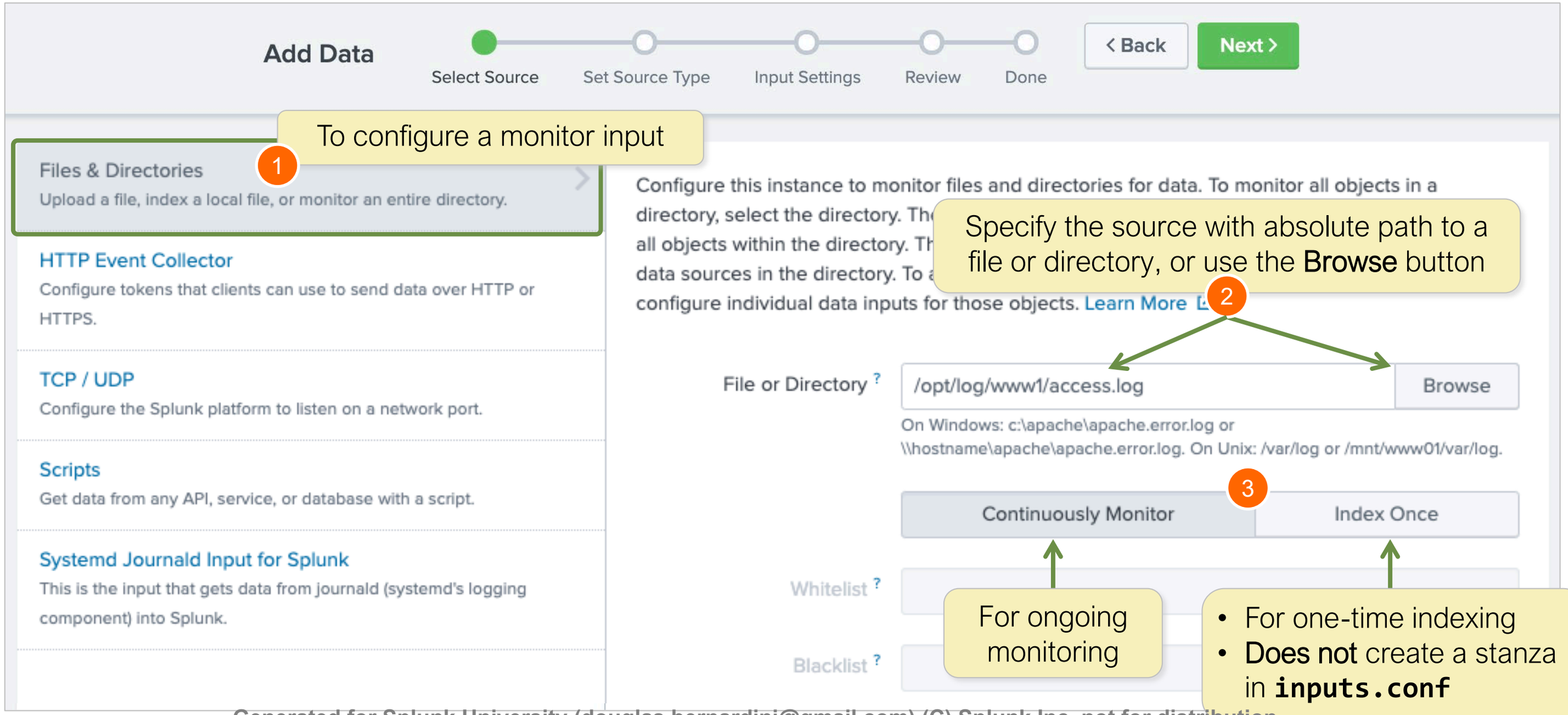
Monitor

- Indexed once or continuously
- Useful for testing or production
- File on the remote Splunk server
- Updates **`inputs.conf`**
- Supports files, directories, http events, network ports, and scripts

Forward

- Data from forwarders managed by this Deployment Server
- Sent to indexers' receiving port
- Main source of input in production
- Updates **`inputs.conf`**

Select Source

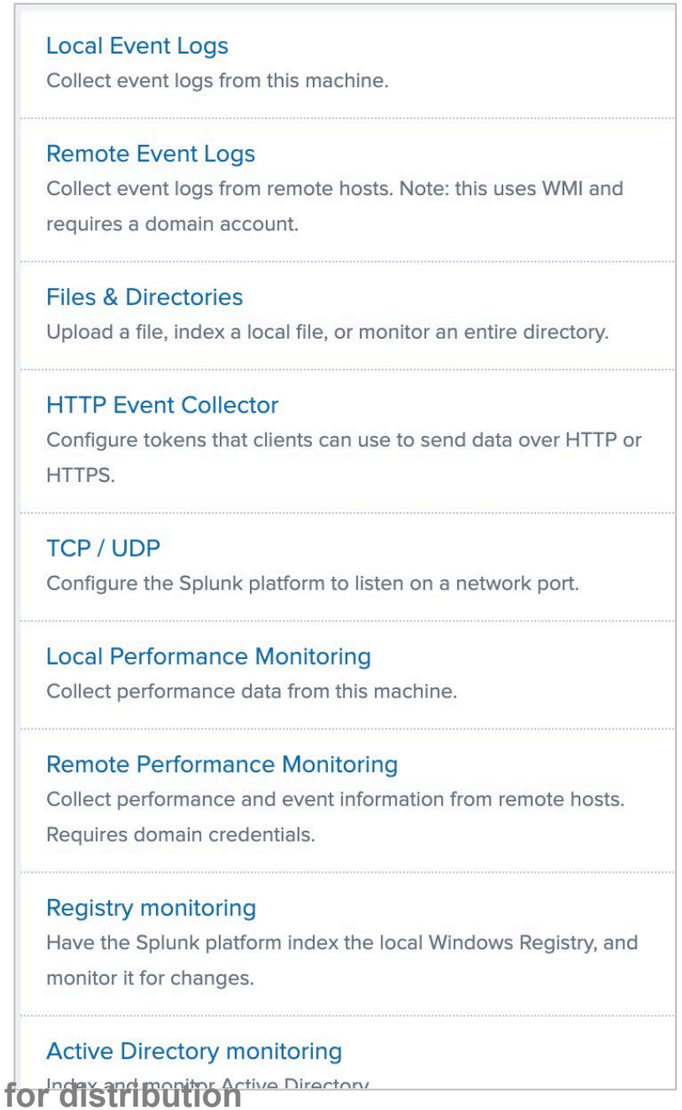


Select Source: Additional Information

- To monitor a shared network drive, enter:

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

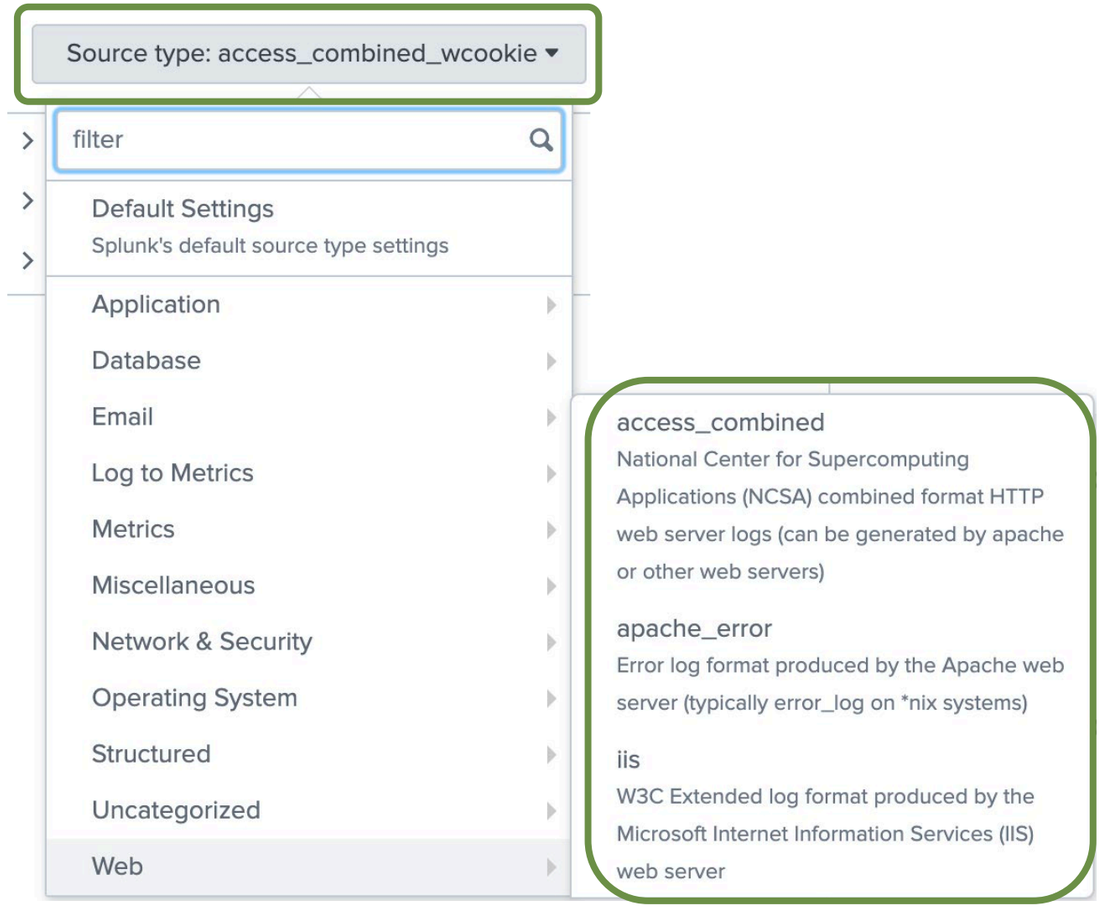
  - Splunk requires read access to the share
- Additional sources on Linux Splunk instances
  - **Systemd Journald Input**
- Additional sources on Windows Splunk instances
  - Including **Event Logs**, **Performance Monitoring**, **Registry monitoring**, and **Active Directory monitoring**



Understanding Source Types

Source Types

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



Pretrained Source Types

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

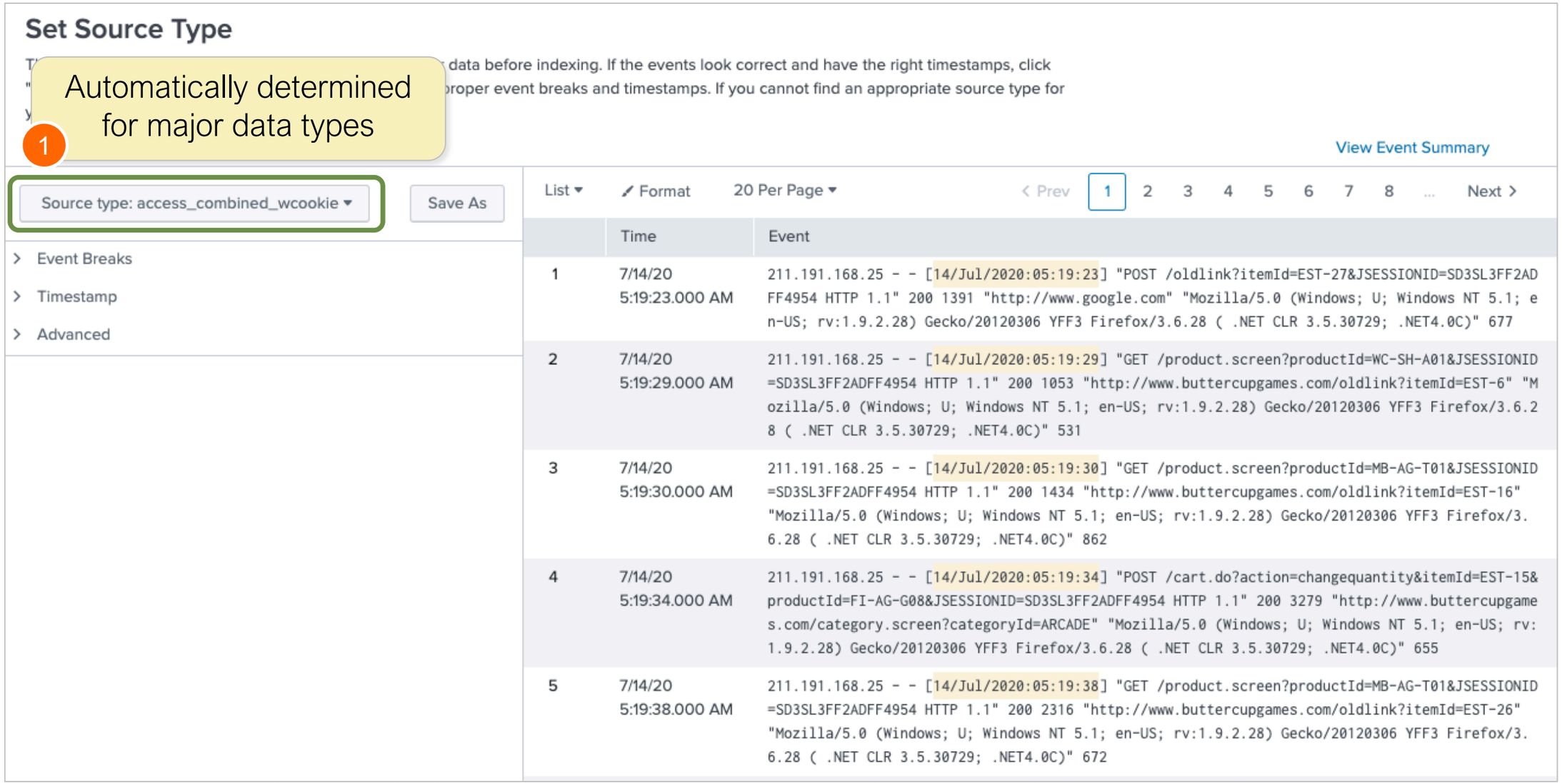
  docs.splunk.com/Documentation/Splunk/latest/Data/Listofpretrainedsourcetypes

Automatically recognized source types [edit]

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

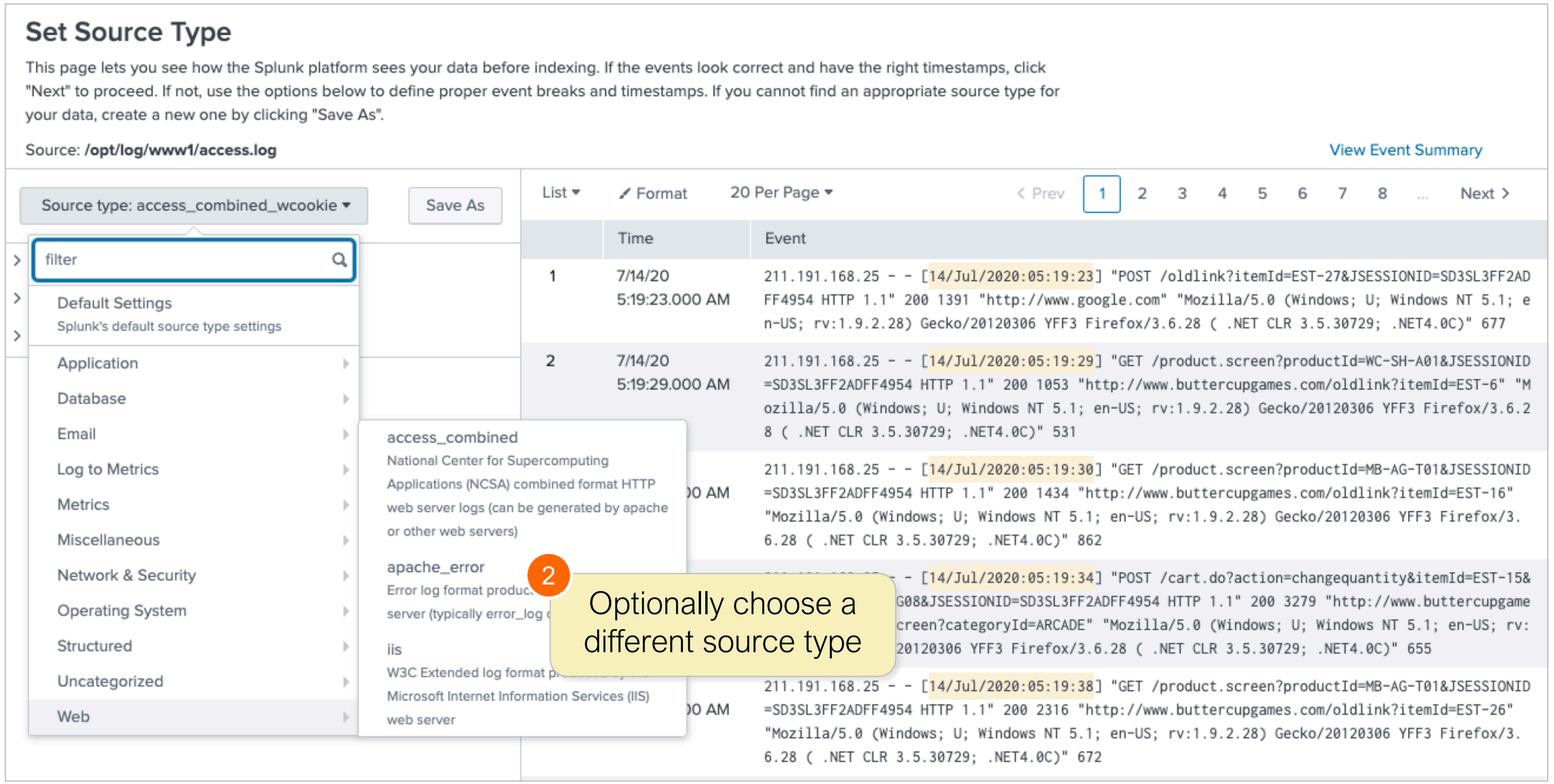
 | 15 September 2021

Set Source Type (Data Preview - 1)

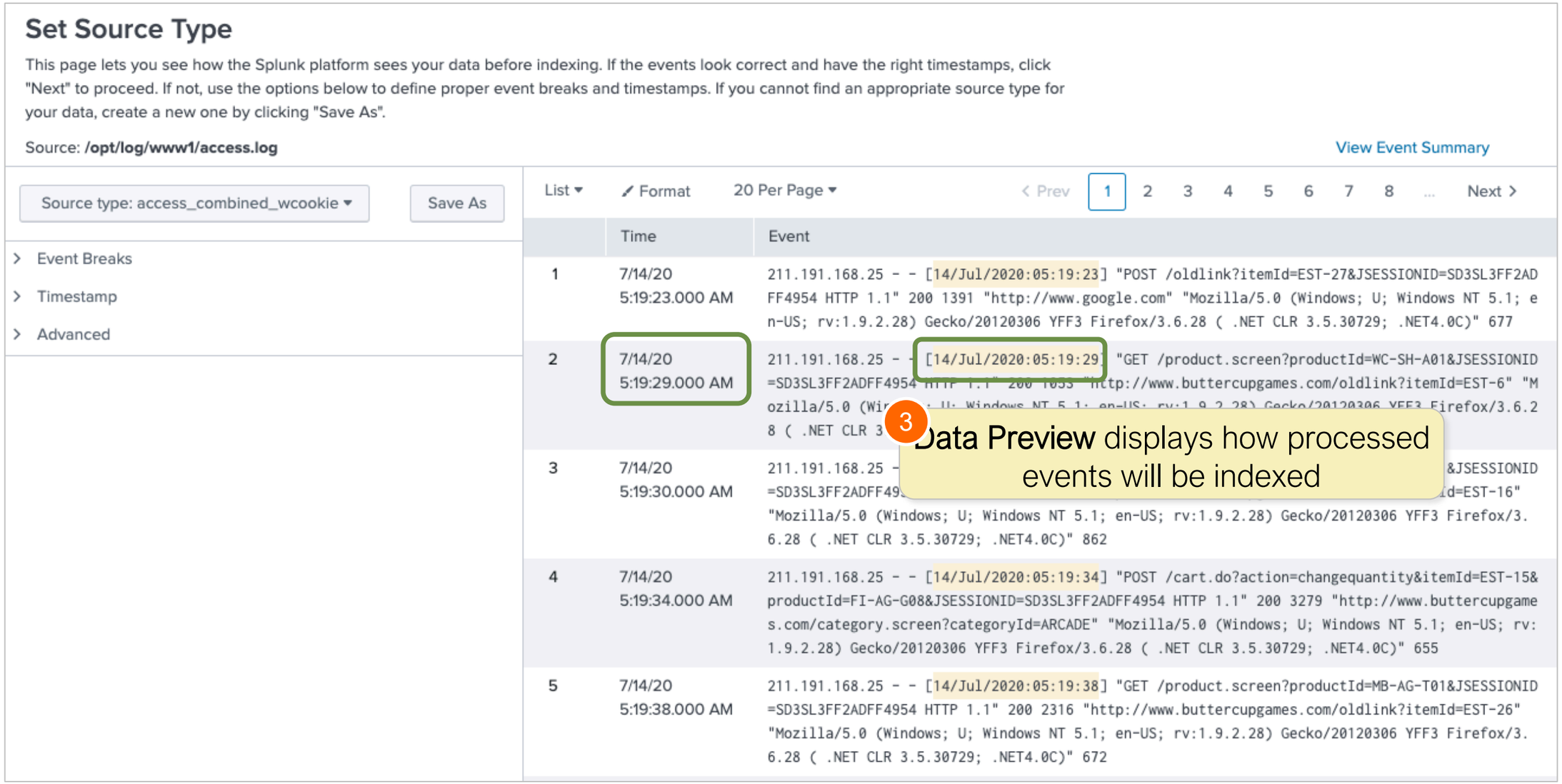


 | 15 September 2021

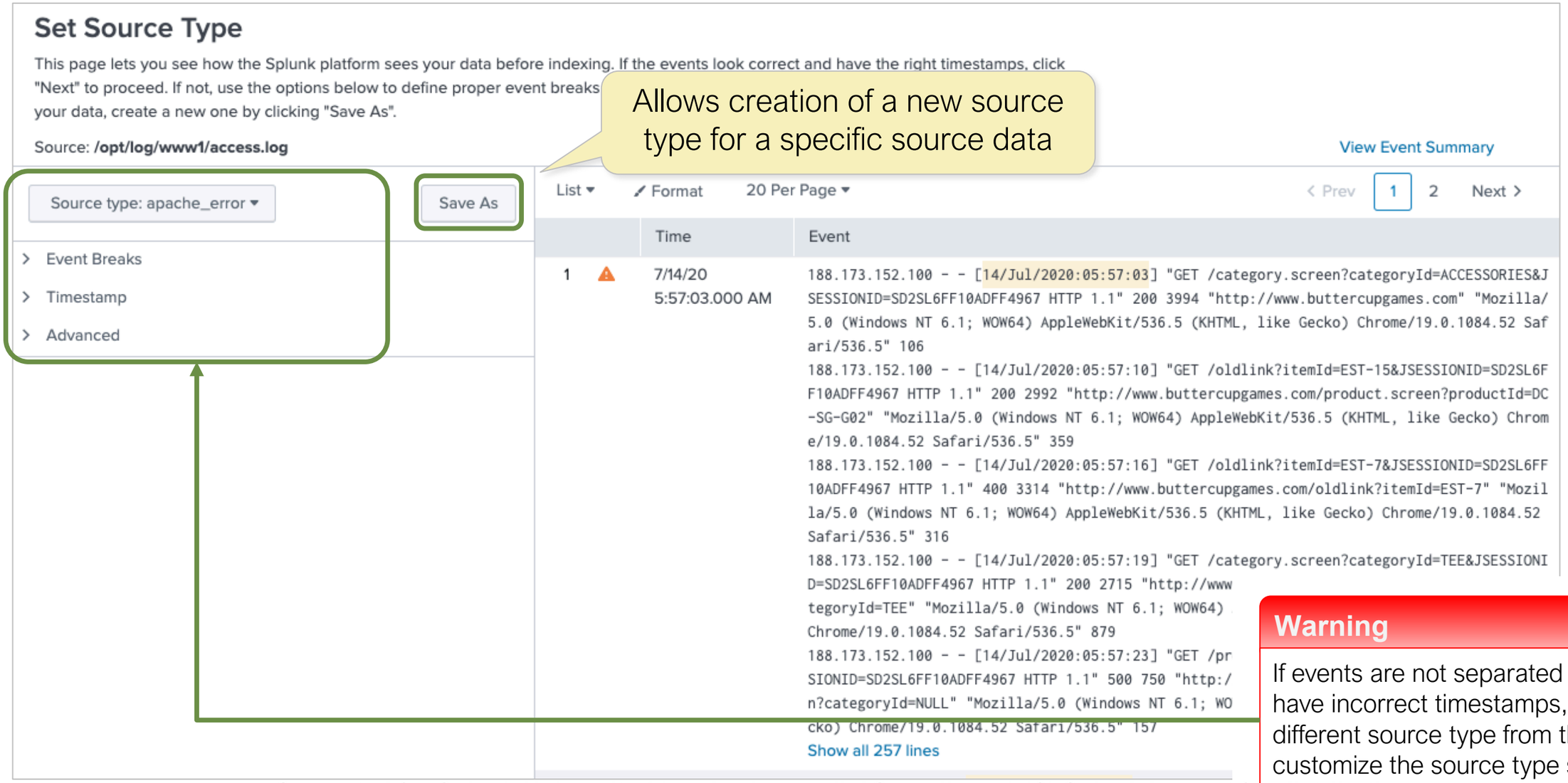
Set Source Type (Data Preview - 2)



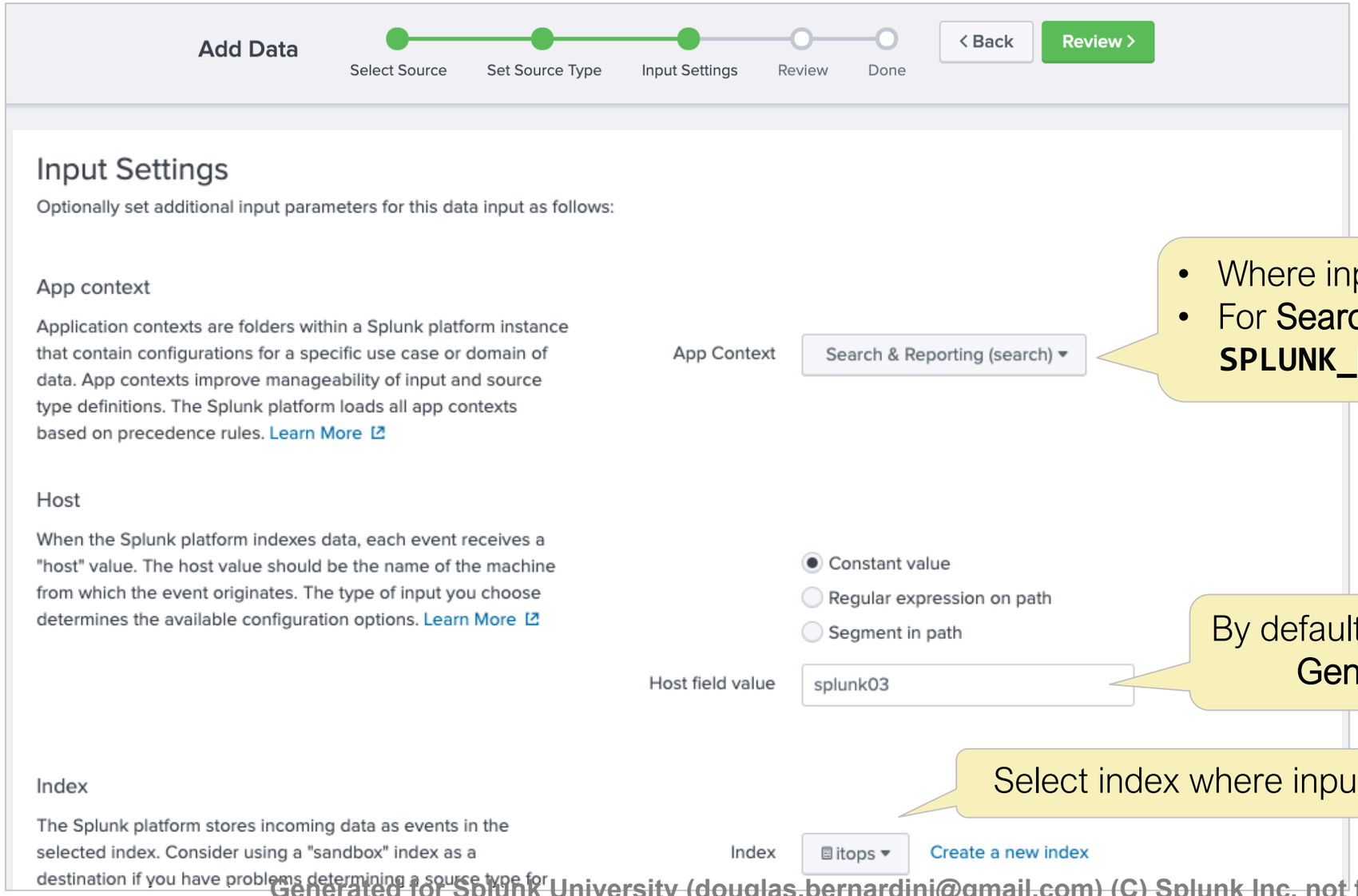
Set Source Type (Data Preview - 3)



Set Source Type (Data Preview - Warning)



Input Settings



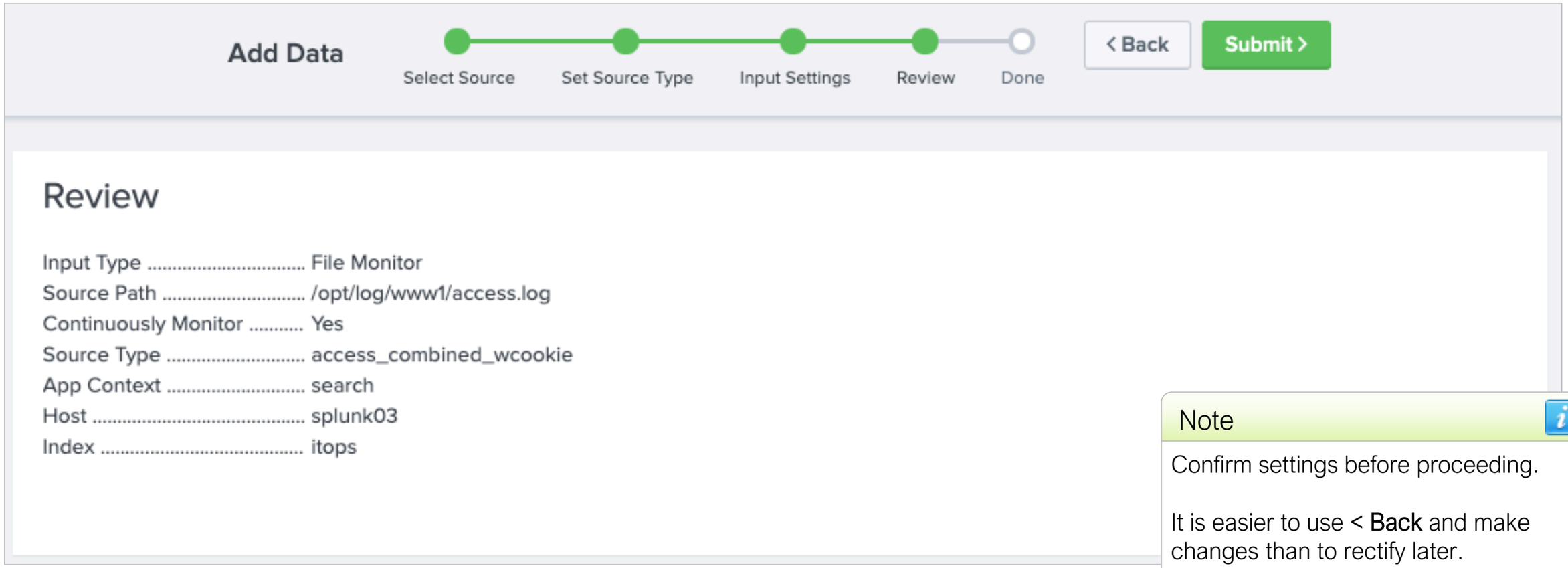
- Where input configuration is saved
- For **Search & Reporting (search)**: **`SPLUNK_HOME/etc/apps/search/local`**

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

Select index where input will be stored

Review

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



Note

Confirm settings before proceeding.

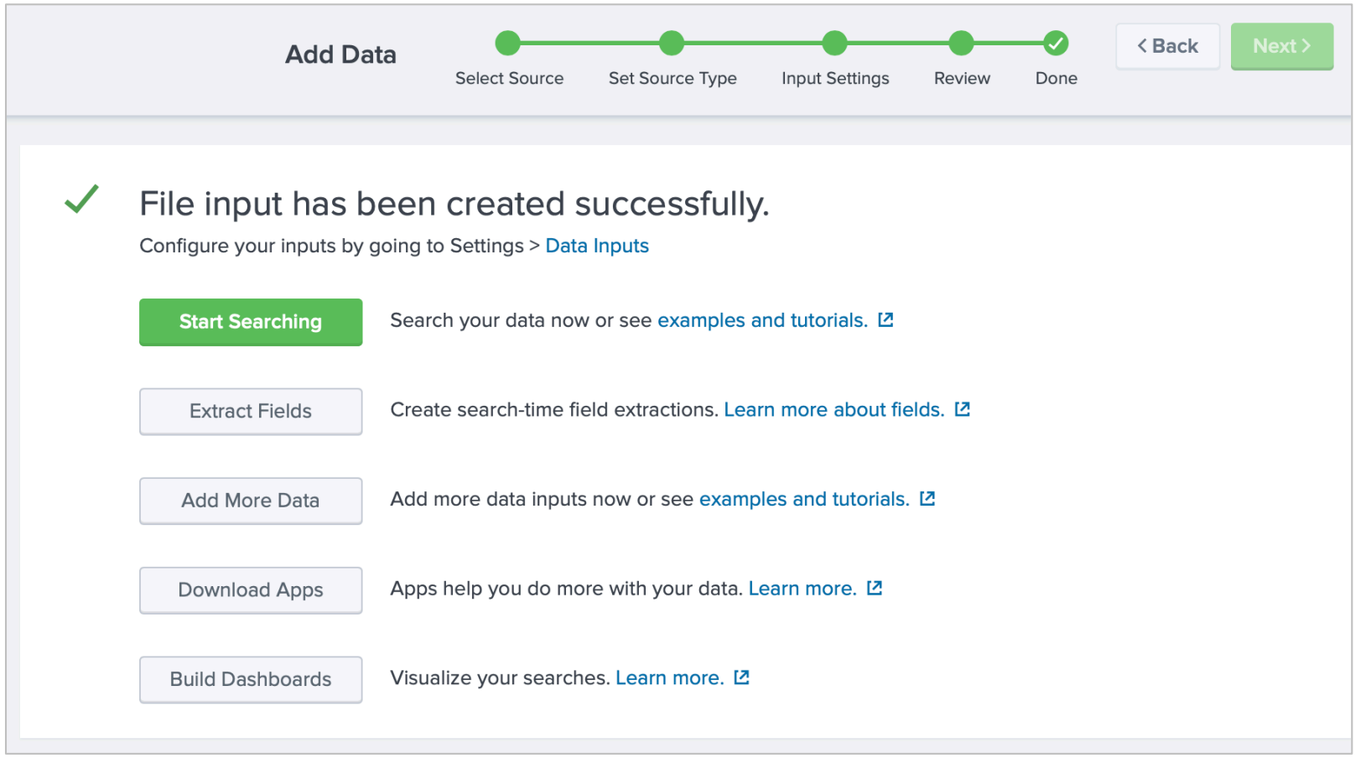
It is easier to use < **Back** and make changes than to rectify later.

 | 15 September 2021

What Happens Next?

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

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

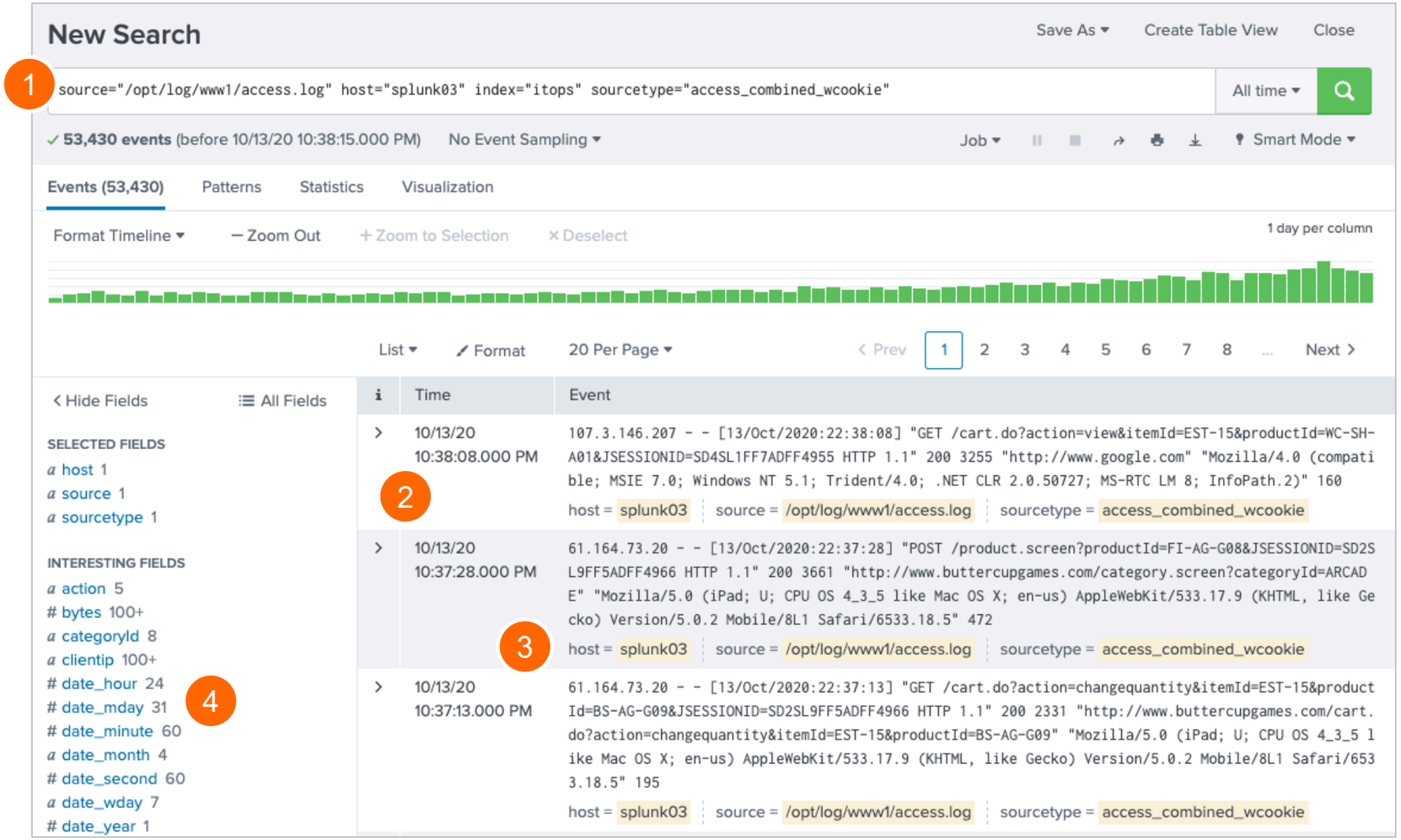


Note

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

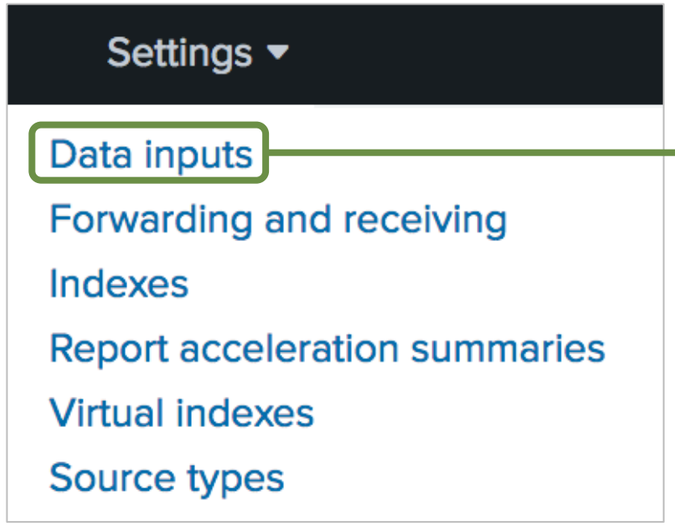
Verify your Input

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



Viewing Configured Inputs

Select **Settings > Data Inputs**



Data inputs

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

Local inputs — Inputs handled by this server

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

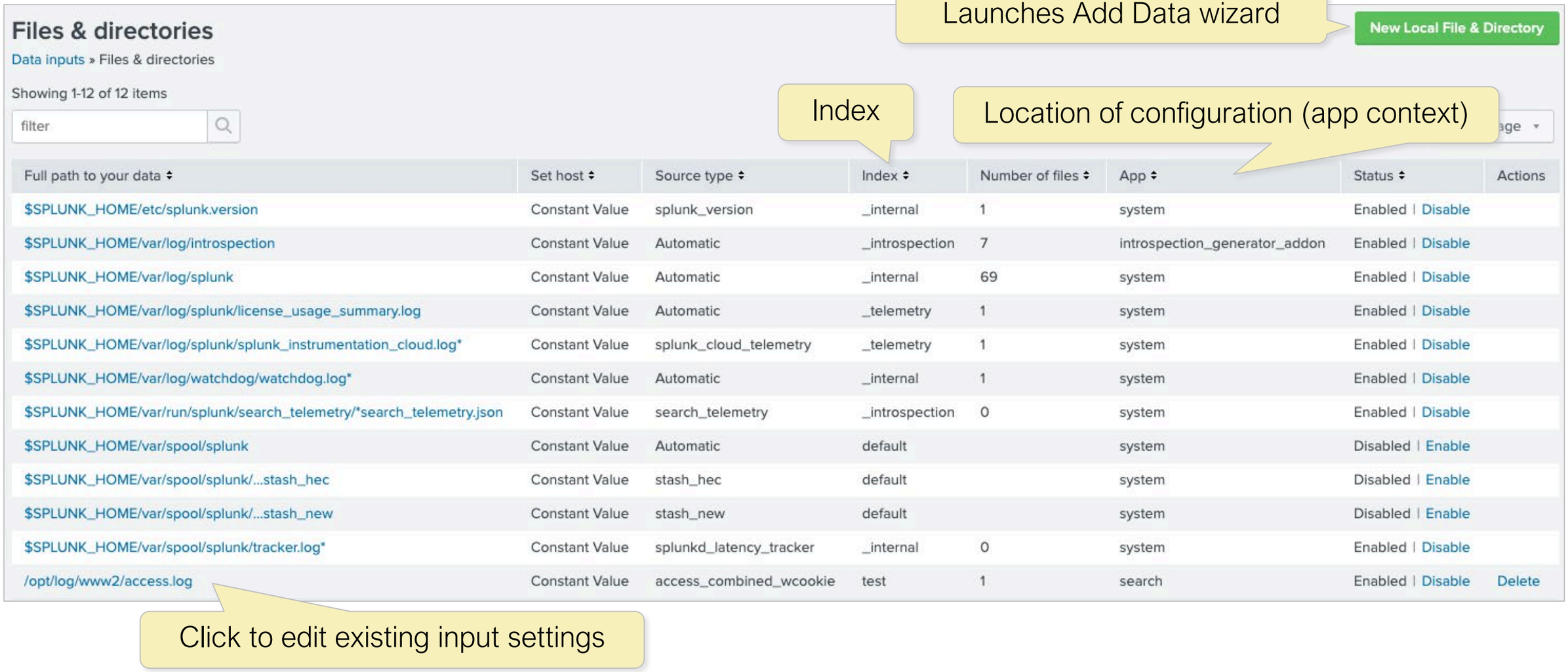
• • •

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

Type	Inputs	Actions
Windows Event Logs Collect event logs from forwarders.	0	+ Add new
Files & Directories Monitor files or directories on forwarders.	0	+ Add new

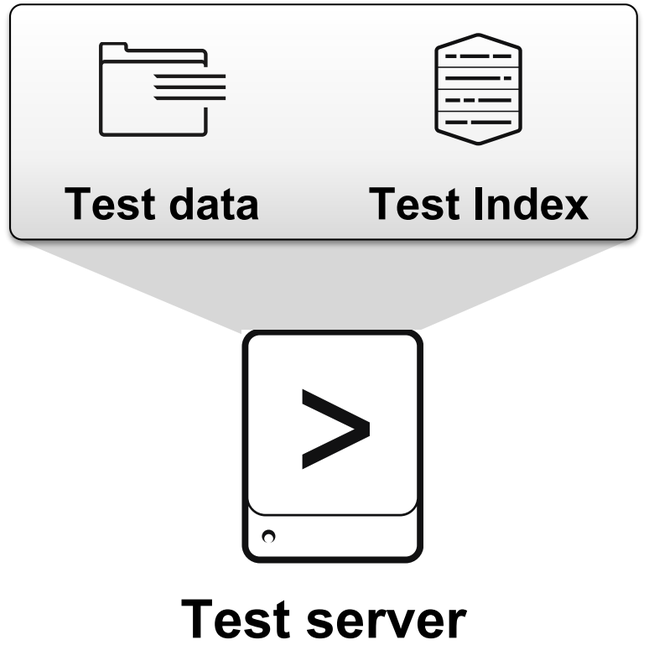
 | 15 September 2021

Viewing Configured Inputs: Files & Directories



Initial Data Input Testing

- Use a Splunk test server
  - Should be running same version as production
- Use test indexes
- Procedure:
  1. Copy production data to test server
  2. Use Splunk Web > **Add Data**
  3. Check to see if **sourcetype** and other settings are applied correctly
  4. Delete the test data, change your test configuration, and repeat as necessary



Module 1 Knowledge Check

- True or False. You cannot change the sourcetype when you go through the **Settings** > **Add Data** wizard.
- True or False. Splunk will not update an **`inputs.conf`** file when you use the **Upload** option in **Settings** > **Add Data**.

Module 1 Knowledge Check – Answers

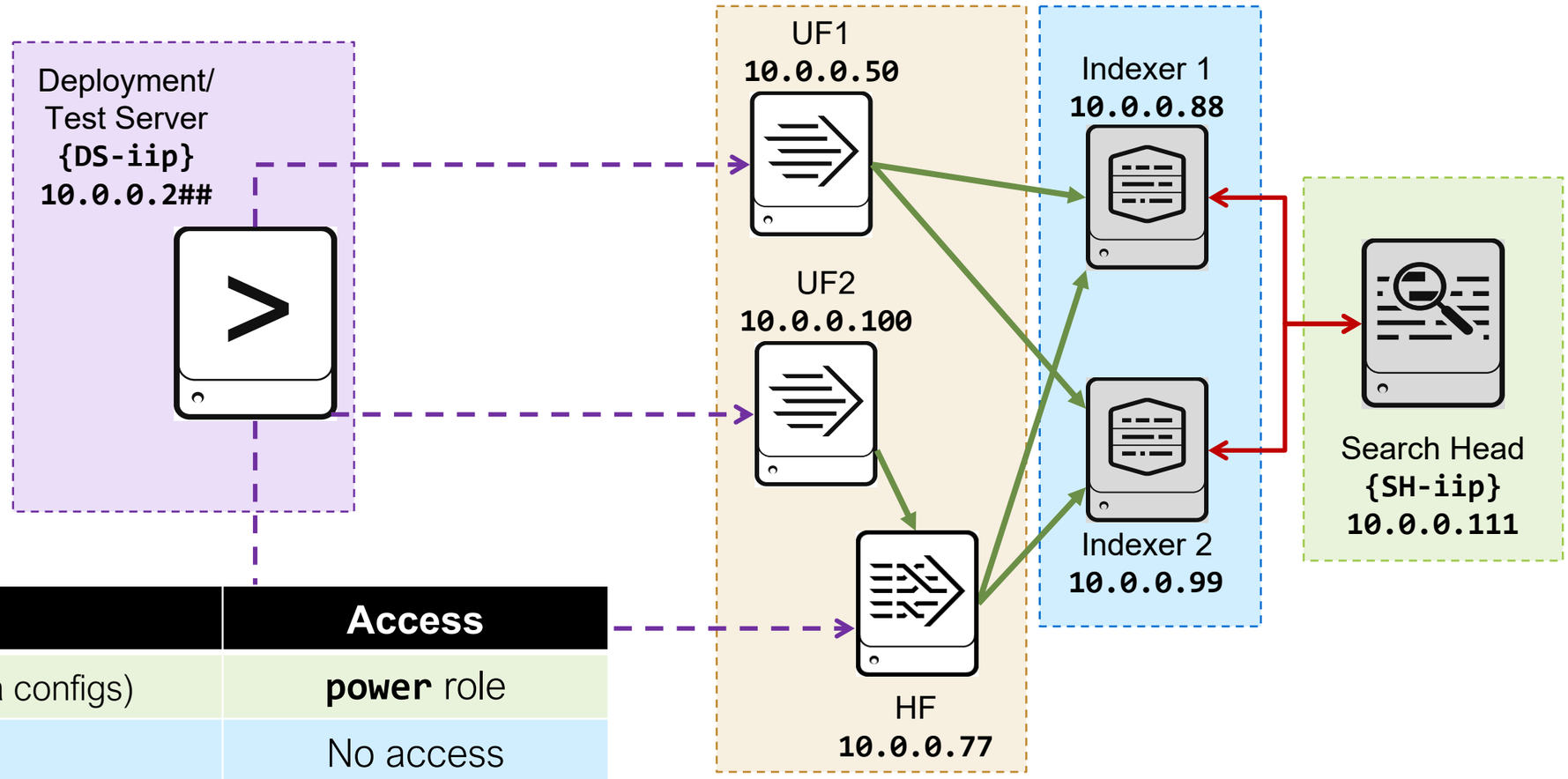
- True or False. You cannot change the sourcetype when you go through the **Settings > Add Data** wizard.

  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.

- True or False. Splunk will not update 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 update an **`inputs.conf`**.

Access Scenario For Course Labs



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

Module 1 Lab Exercise

Time: 20 minutes

Description: Add a Local Data Input

Tasks:

- Discover Splunk Enterprise lab environment
- Log into search head and test/deployment server
- Create a test index on the deployment/test server
- Index a file on the deployment server
- Verify the indexed events with their metadata values

Module 2: Configuration Files

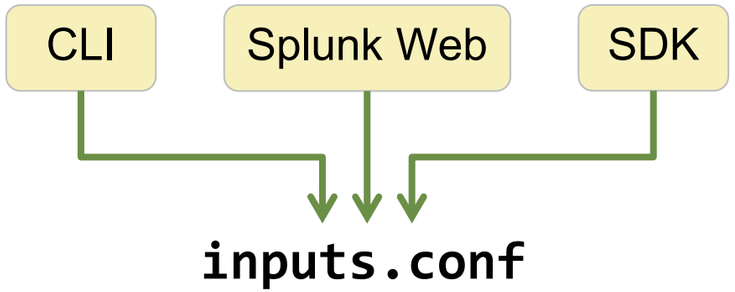
Module Objectives

- Identify Splunk configuration files and directories
- Describe index-time and search-time precedence
- Validating and updating configuration files

Splunk Configuration Files

Configuration Files (`.conf`)

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




```
[default]
host=www

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

Note

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

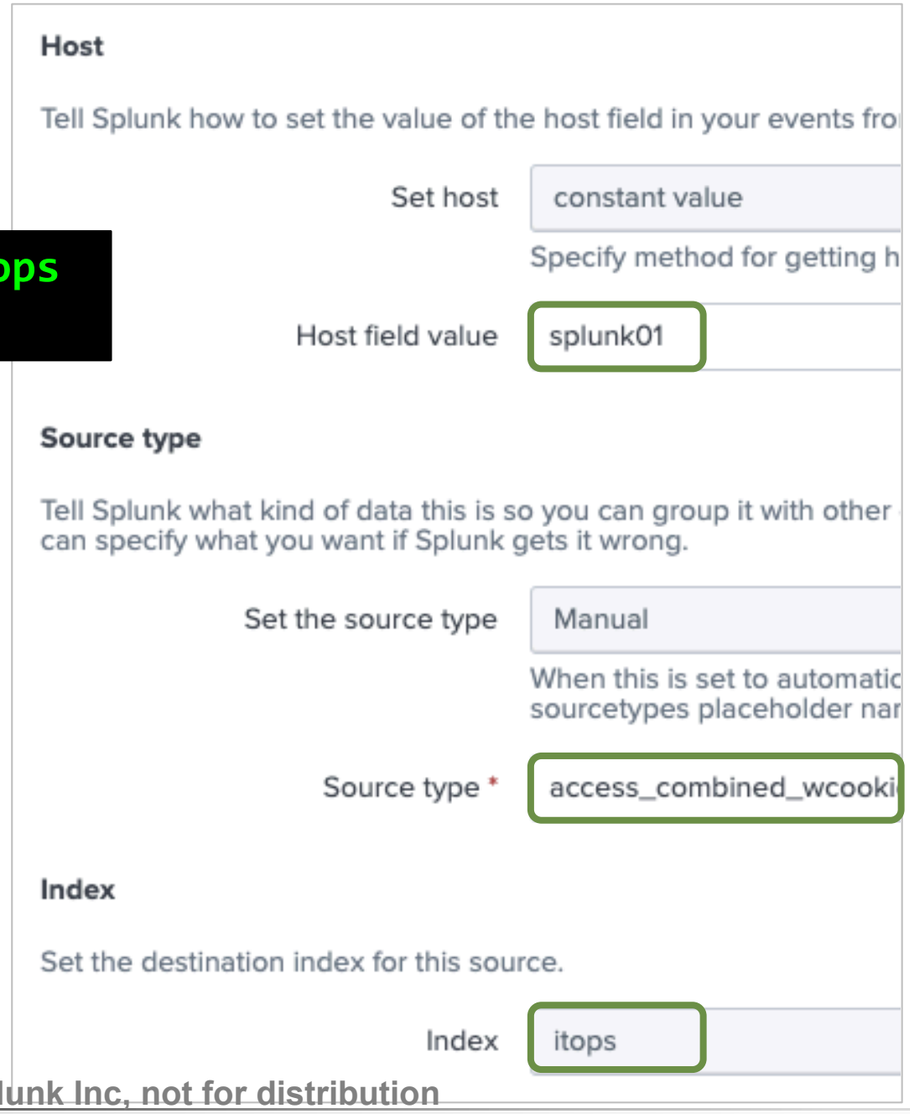
Methods for Modifying Splunk Configurations

- Splunk Web
- Splunk CLI

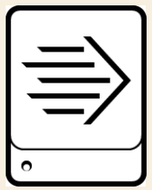
```
./splunk add monitor /opt/log/www1/access.log -index itops
   -sourcetype access_combined_wcookie -host splunk01
```

- Editing **.conf** files

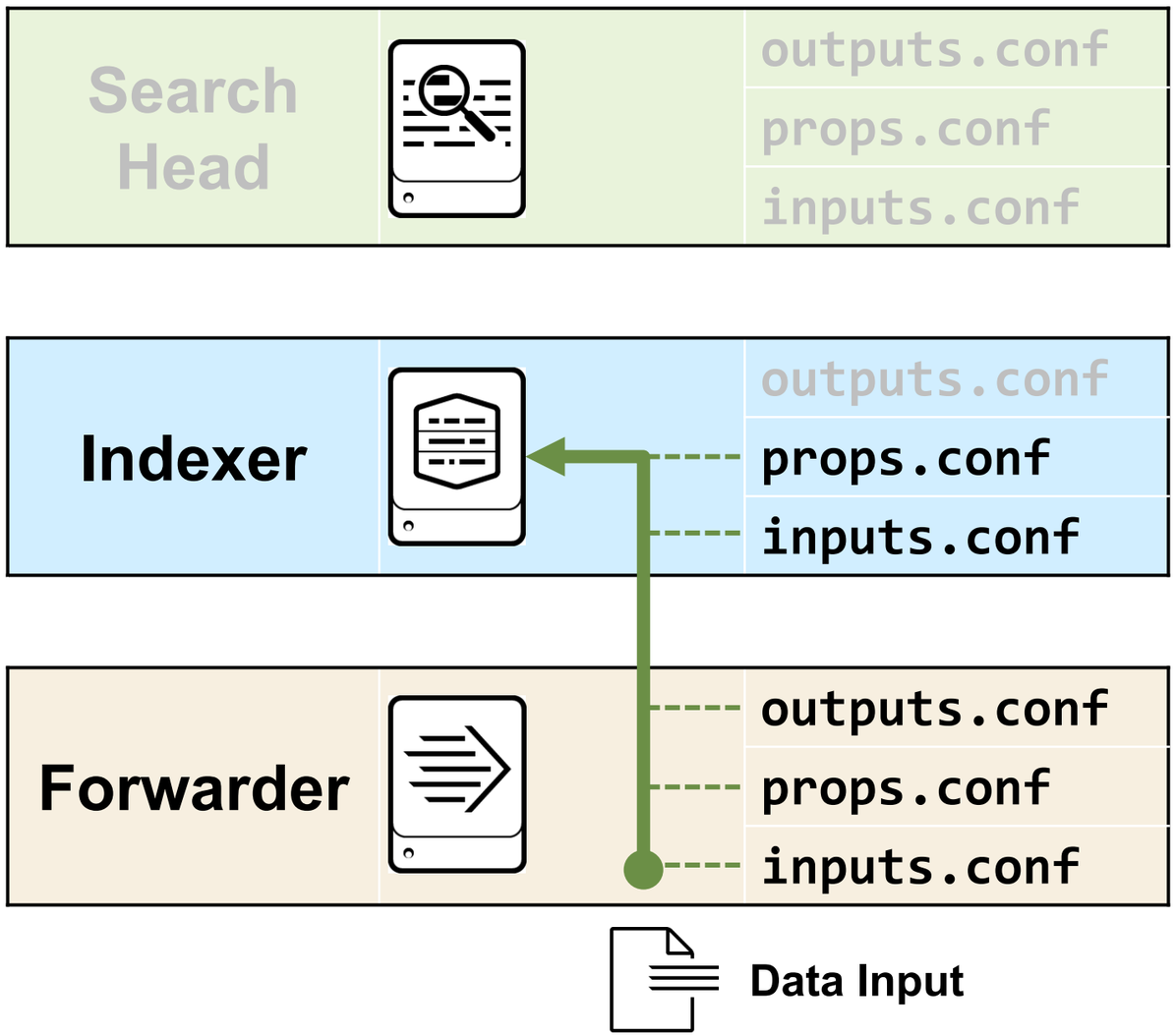
```
[monitor:///opt/log/www1/access.log]
disabled = false
host = splunk01
index = itops
sourcetype = access_combined_wcookie
```



Commonly Used Splunk Configuration Files

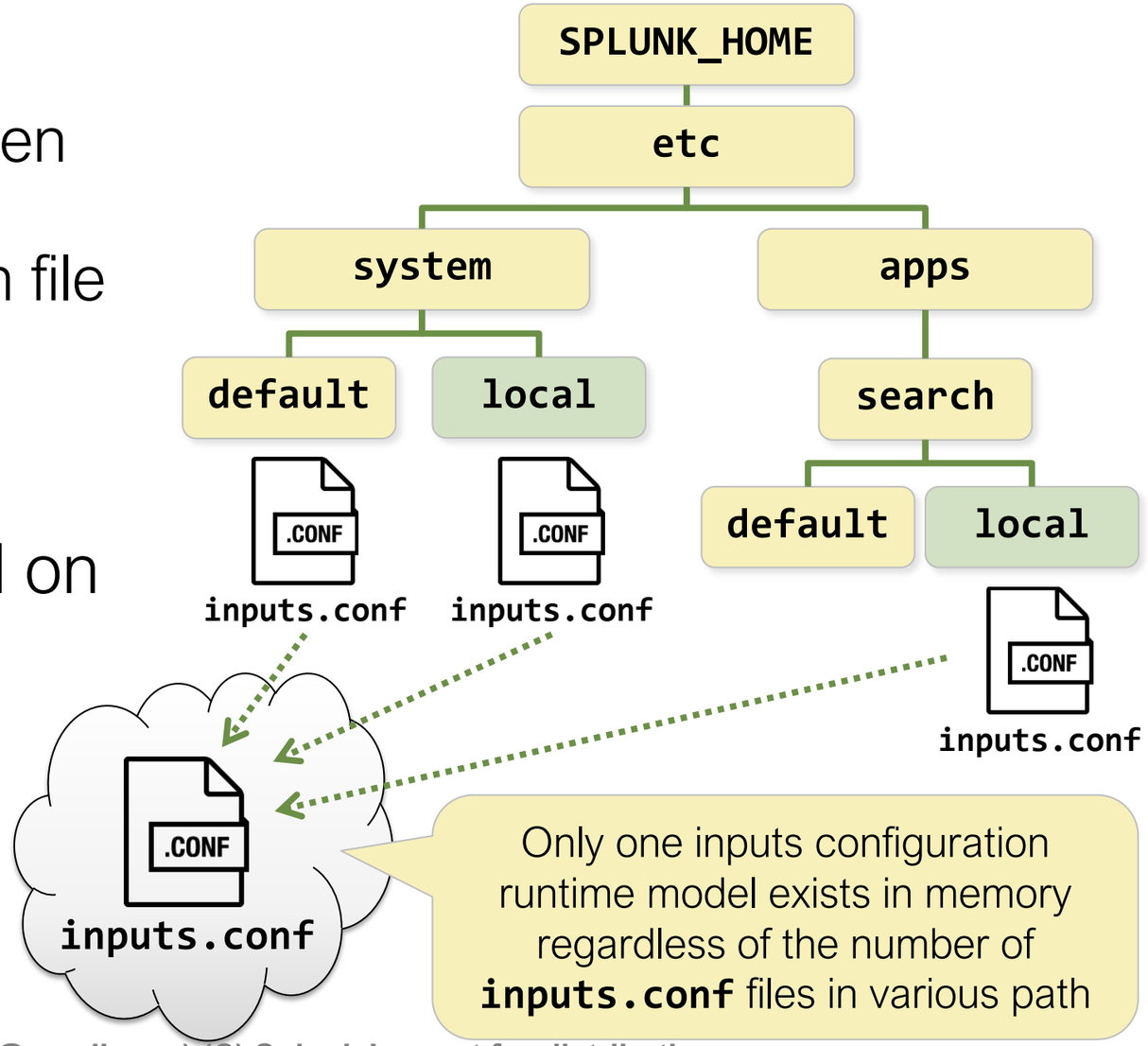
Search Head		`outputs.conf`	Where to forward data
		`props.conf`	Search-time Field Extractions, lookups, and so on
		`inputs.conf`	What data is collected (*internal Splunk logs*)
Indexer		`outputs.conf`	Not generally needed
		`props.conf`	Parsing
		`inputs.conf`	What data is collected; Which ports to listen to
Forwarder		`outputs.conf`	Where to forward data
		`props.conf`	Limited parsing
		`inputs.conf`	What data is collected (*production data*)

Configuration Files Used During Data Input

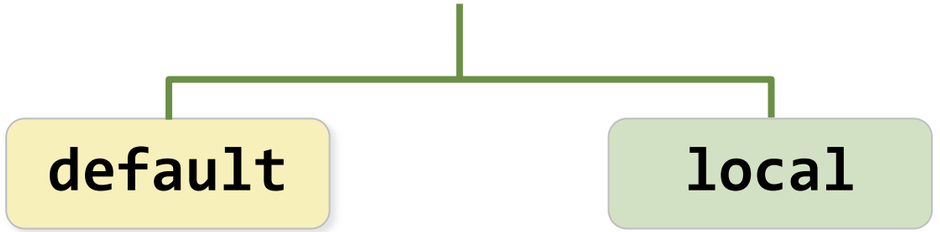


Merging of Configuration Files

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



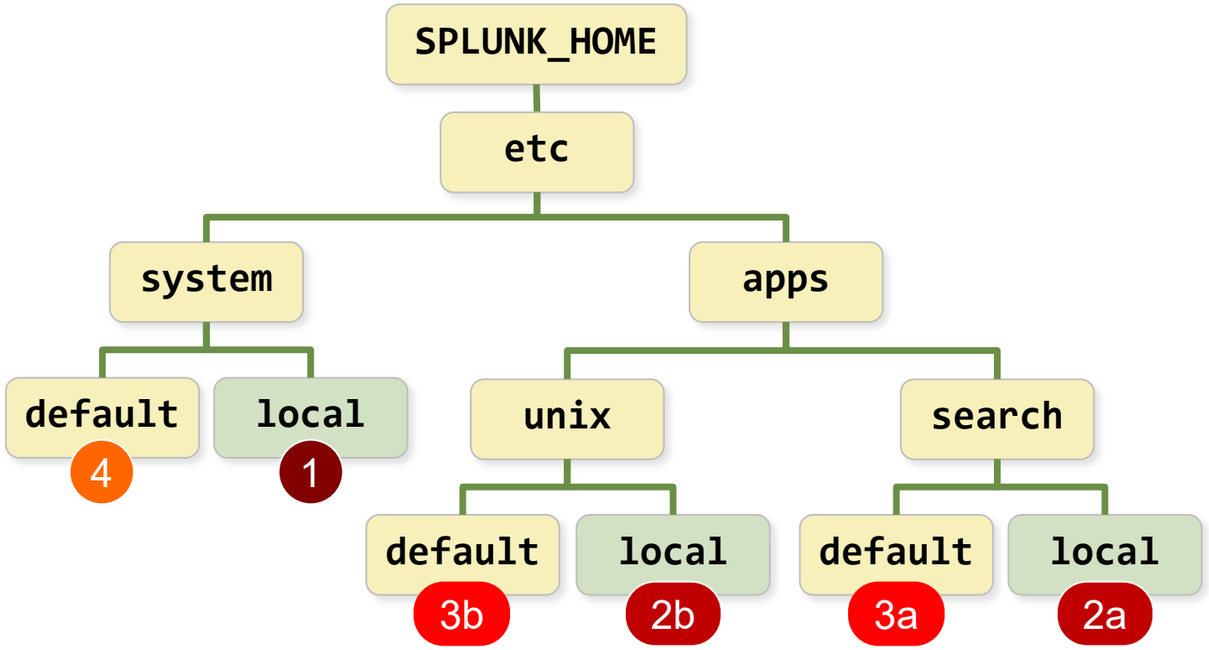
Default versus Local Configuration



- Shipped with Splunk or app
- Overwritten on update
- Do not modify

- Keeps your changes
- Preserved on update
- Only modify these versions
- Overrides **`default`** settings

Index-Time Precedence (Global Context)



Precedence order

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

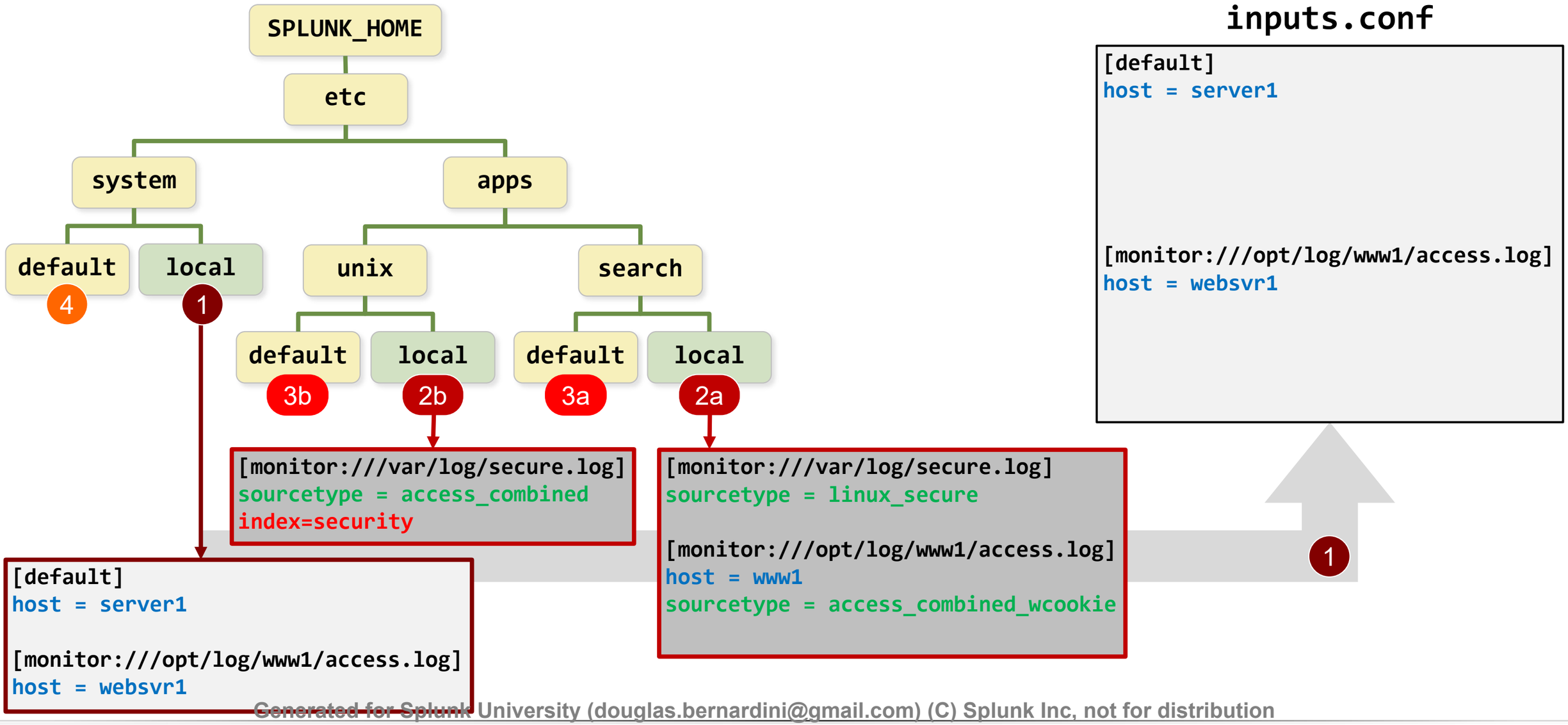
Note

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

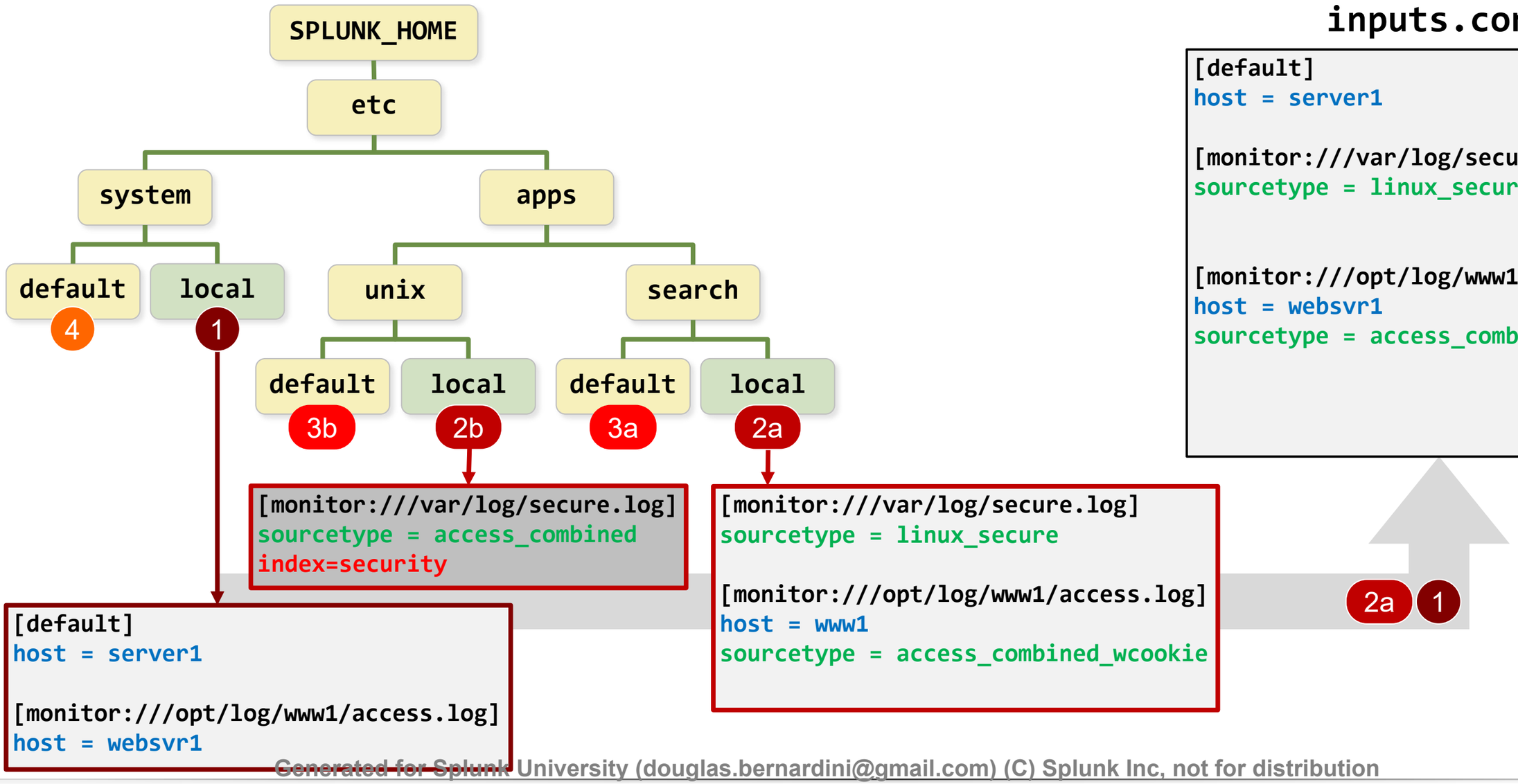
Note

This precedence is different for indexer cluster peers.

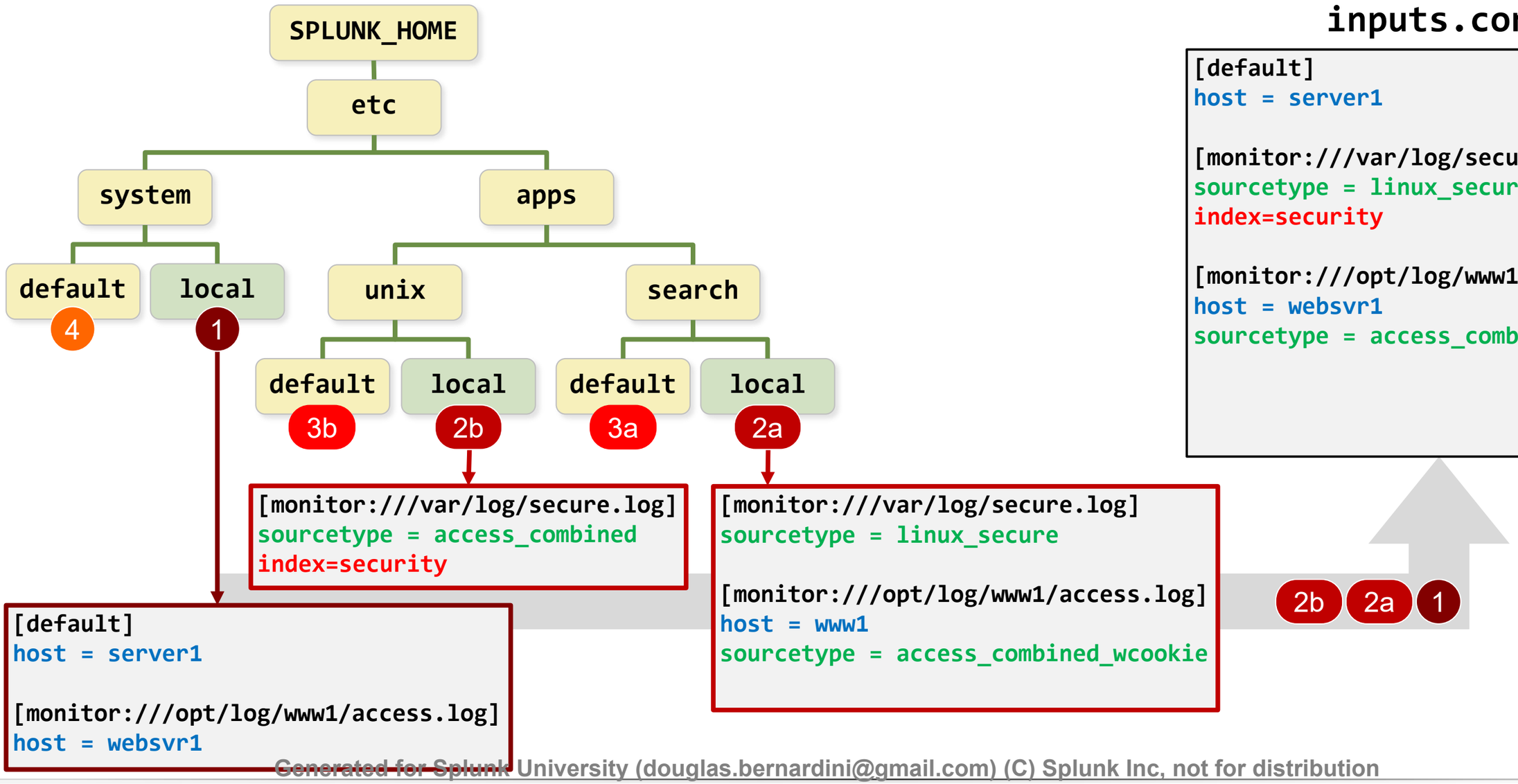
Example of Index-Time Precedence (1)



Example of Index-Time Precedence (2)

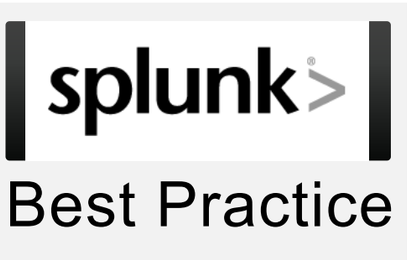


Example of Index-Time Precedence (3)



Configuration Best Practices

- Avoid storing configurations in **`SPLUNK_HOME/etc/system/local`**
  - Local context settings *always* take precedence
  - Attempting to override index-time settings in an app will fail
  - Managing these settings with a deployment server is impossible
- Create an app to manage system settings
  - Allows you to manage settings with a deployment server
  - Manage system configurations in an app (e.g. **`DC_app`**) under **`SPLUNK_HOME/etc/apps/<appname>/local`**
  - Refer to the *Forwarder Management* module



Validating the Splunk Configuration

Validating the on-disk configuration

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

Validating the in-memory configuration

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

Configuration Validation with **btool**

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

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

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

Example using **btool**

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

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

  etc/apps/search/local/inputs.conf    [monitor:///var/log/secure.log]
  system/local/inputs.conf             host = server1
  etc/apps/unix/local/inputs.conf      index = security
  etc/apps/search/local/inputs.conf    sourcetype = linux_secure
```

etc/apps/unix/local/inputs.conf

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

etc/apps/search/local/inputs.conf

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


Module 2 Knowledge Check

- Which configuration file tells a Splunk instance to ingest data?
- True or False. **`btool`** shows on-disk configuration for requested file
- 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 2 Knowledge Check – Answers

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

  inputs.conf

- True or False. **btool** shows on-disk configuration for requested file.

  True.

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

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

Module 2 Lab Exercise

Time: 10 minutes

Description: Configuration Files

Tasks:

- Use CLI to connect to Splunk components
- View the **`inputs.conf`** stanzas manually and using **`btool`**

Module 3: Forwarder Configuration

Module Objectives

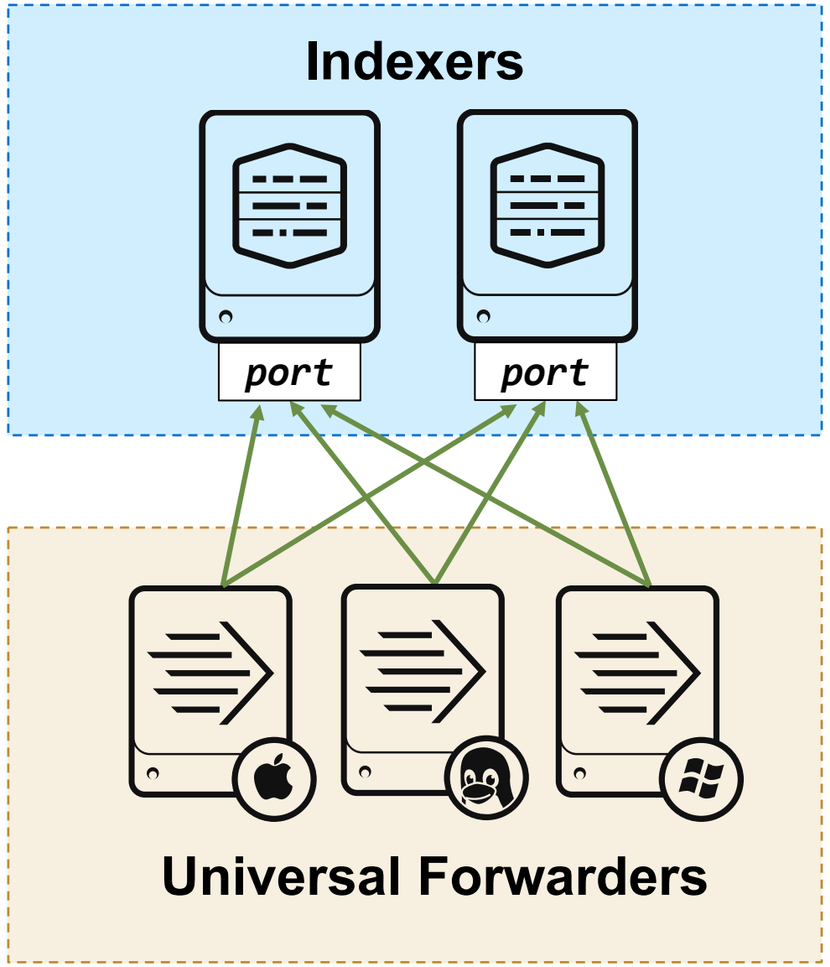
- Identify the role of production indexers and forwarders
- Understand and configure Universal Forwarders
- Understand and configure Heavy Forwarders
- Understand and configure intermediate forwarders
- Identify additional forwarder options

Understanding Universal Forwarders

Universal Forwarders (UF)

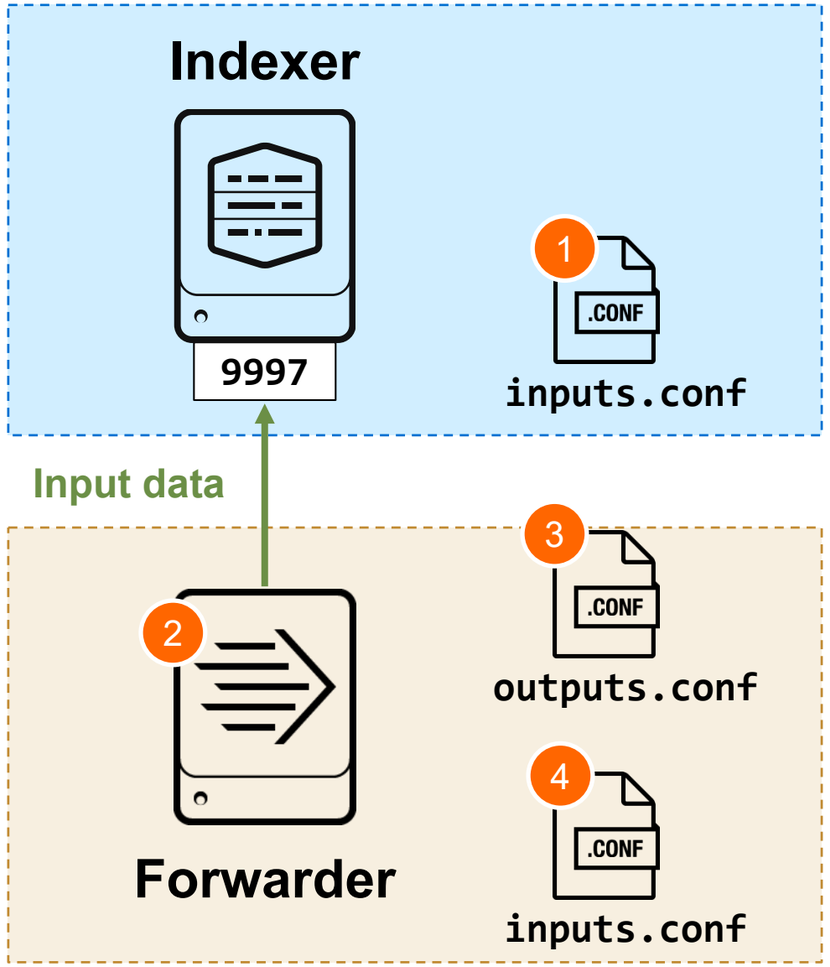
- Gathers data from a host
- Sends data over the network to receiving ports on receivers (usually an indexer)
- Provided as separate installation binary with a built-in license (no limits)
- Designed to run on production servers

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



Universal Forwarder Configuration Steps

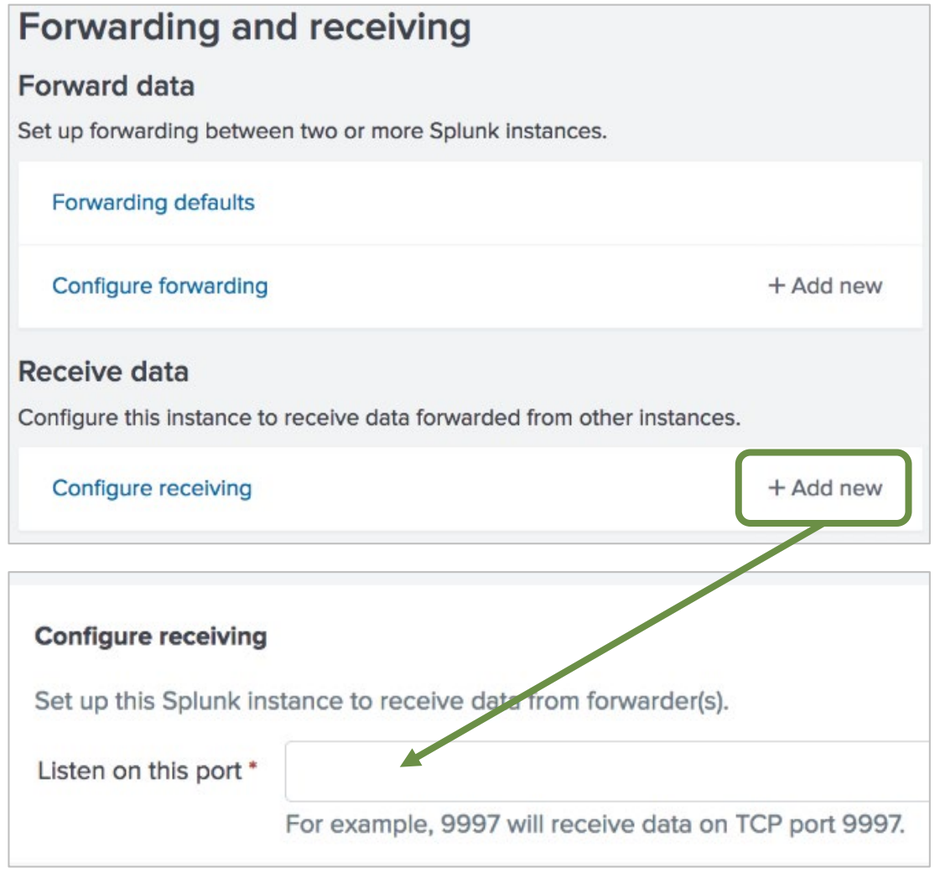
1. Set up a receiving port on each indexer
   - Task only needs to be performed once
2. Download and install Universal Forwarder
3. Set up forwarding on each forwarder
4. Add inputs on forwarders



Configure the Receiving Port on Each Indexer

- Using Splunk Web:
  1. Select **Settings > Forwarding and receiving**
  2. Next to **Configure receiving**, select **Add new**
  3. Enter a port number and click **Save**
  - Stored in most recently visited app: **`SPLUNK_HOME/etc/apps/<app>/local`**
- Using CLI:
  - Run **`splunk enable listen <port>`**
  - Stored in **`SPLUNK_HOME/etc/apps/search/local`**
- Manually in **`inputs.conf`** as:

  `[splunktcp://port]`



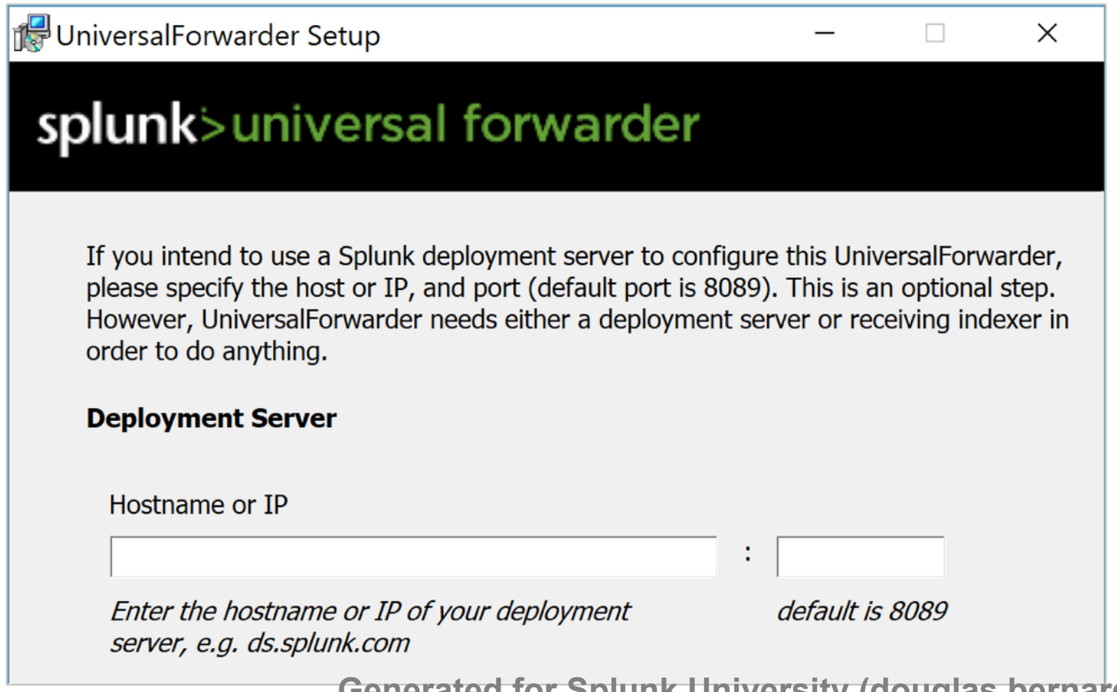
Installing a Universal Forwarder

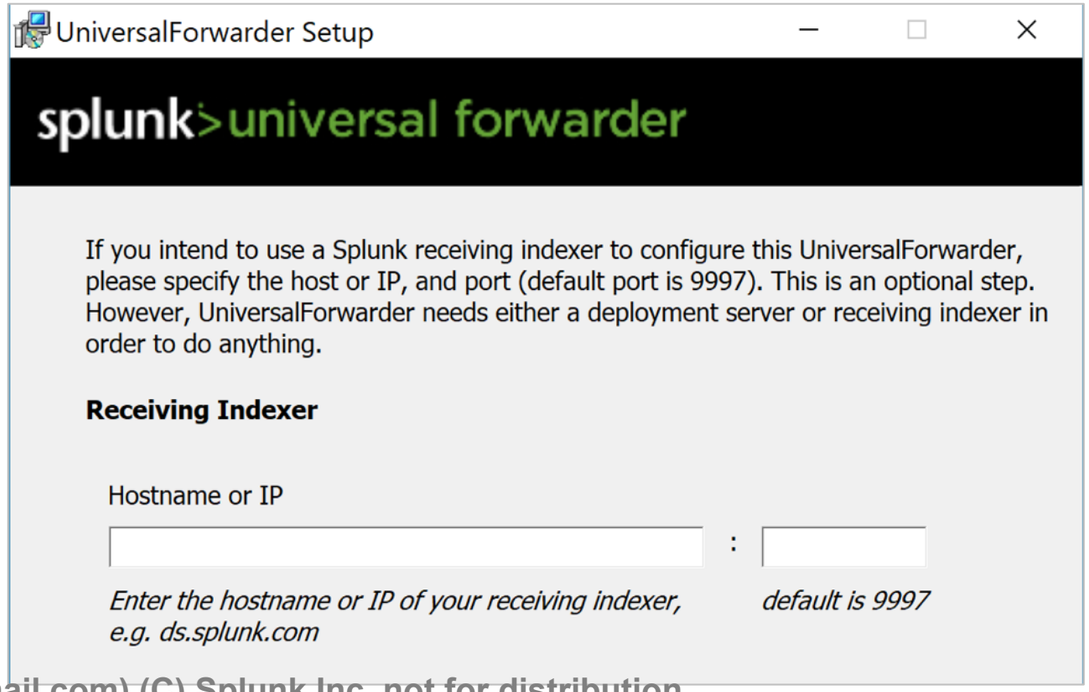
	*NIX	Windows
Download	www.splunk.com/en_us/download/universal-forwarder.html	
Install	• Un-compress **.tgz**, **.rpm**, or **.deb** file in the path Splunk will run from • Default **SPLUNK_HOME** is: **/opt/splunkforwarder**	• Execute **.msi** installer (or use the CLI) • Default **SPLUNK_HOME** is: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

- Silent installation methods exist on all platforms
- Same **splunk** command-line interface in **SPLUNK_HOME/bin**
  - Same commands for start/stop, restart, etc.
  - An admin account and password are required

Using the Interactive Windows Installer

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





Defining Target Indexers on the Forwarder

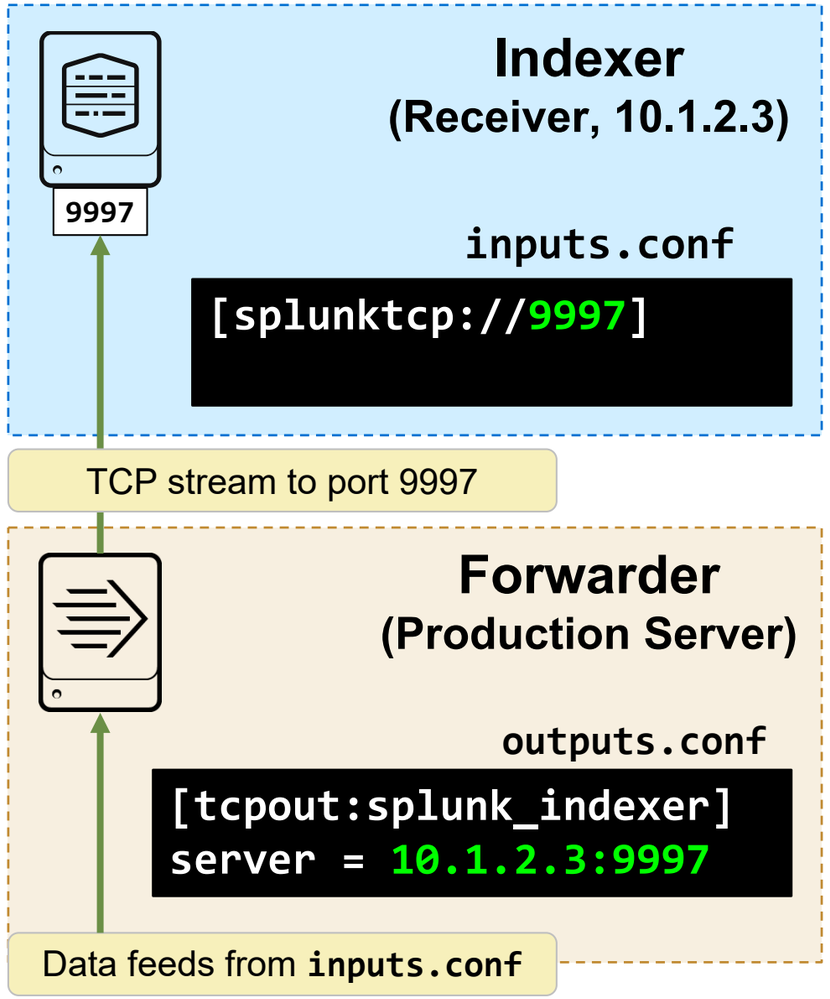
- To configure target indexers on forwarders, either:
  - Run **`splunk add forward-server <indexer:receiving_port>`**
  - Modify **`outputs.conf`**
- Splunk logs are automatically sent to indexer's **`_internal`** index
- Example: **`splunk add forward-server 10.1.2.3:9997`** configures **`outputs.conf`** as:

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

[tcpout-server://10.1.2.3:9997]

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

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

Forwarder **outputs.conf** File

- Points the forwarder to the receivers
  - **splunktcp** stanza sets the indexer to listen on a port for feeds from Splunk forwarders
  - **server** sets a forwarder's destination to one or more receivers (IP or DNS name + receiver port), separated by commas
- Can specify additional options:
  - Load balancing
  - SSL
  - Compression
  - Alternate indexers
  - Indexer acknowledgement

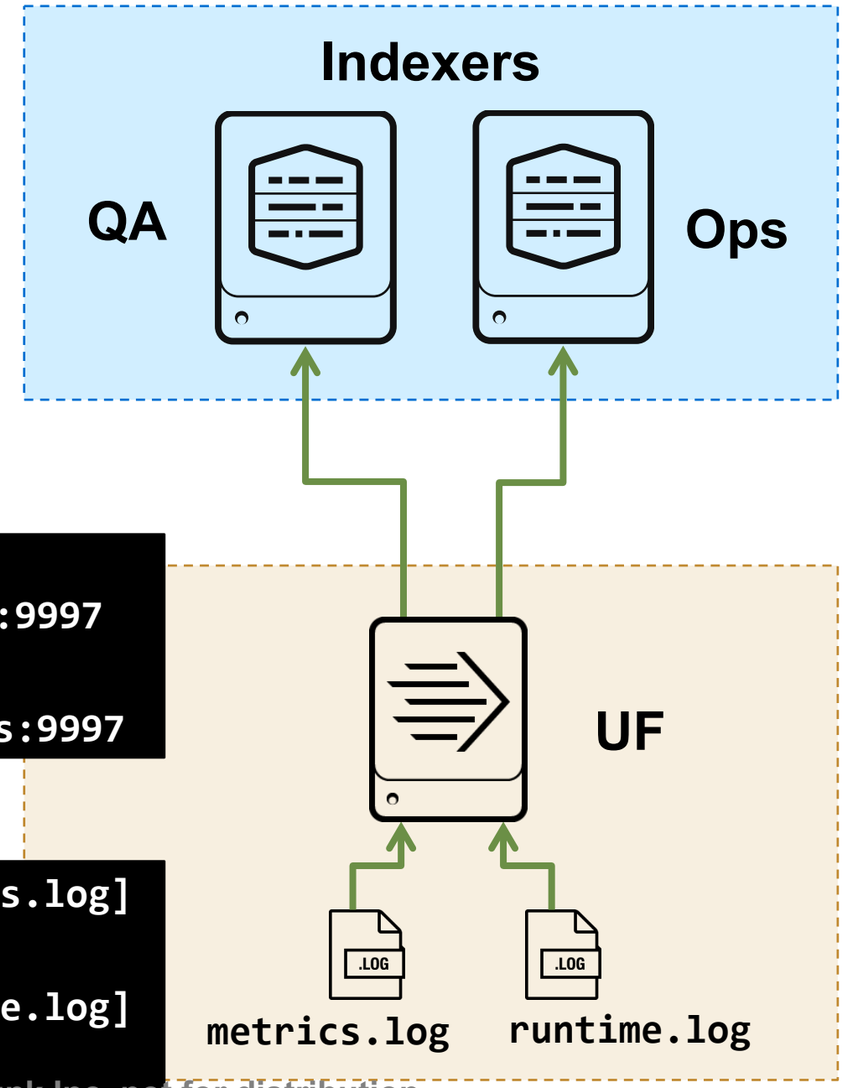


Configuration Validation and Troubleshooting

- To verify the configuration:
  - On forwarder, run: **`splunk list forward-server`**
  - On indexer, run: **`splunk display listen`**
- To verify successful connection:
  - On search head, search: `index=_internal host=<forwarder_hostname>`
- Troubleshooting forwarder connection
  - Check **`SPLUNK_HOME/var/log/splunk/splunkd.log`** on forwarder:
    `tail -f splunkd.log | egrep 'TcpOutputProc|TcpOutputFd'`

Selectively Forwarding Data to Indexers

- Universal forwarder can route based on sources
- Example:
  - **metrics.log** → QA indexer
  - **runtime.log** → Ops indexer



Define multiple **tcpout** stanzas in **outputs.conf**

outputs.conf

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

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

Specify **_TCP_ROUTING** for each source in **inputs.conf**

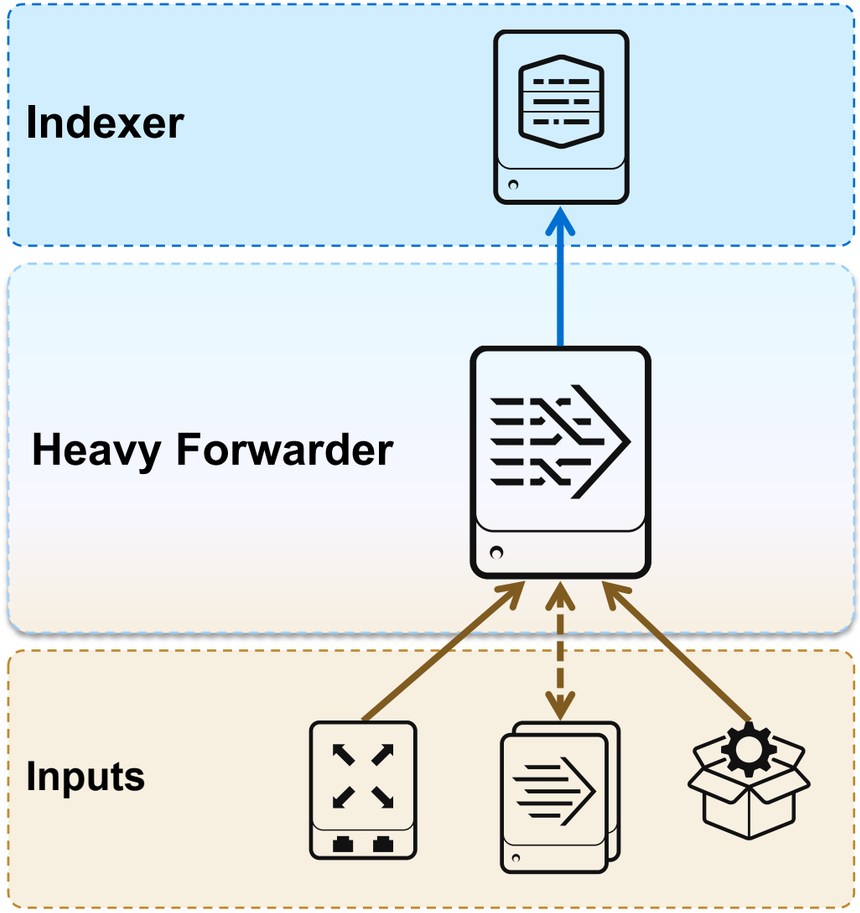
inputs.conf

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

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

Understanding Heavy Forwarders (HF)

Heavy Forwarders (HF)

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



Deciding Between UF and HF

Universal Forwarder

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

vs.

Heavy Forwarder

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

Optimizing the Heavy Forwarder

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

Best Practice

outputs.conf

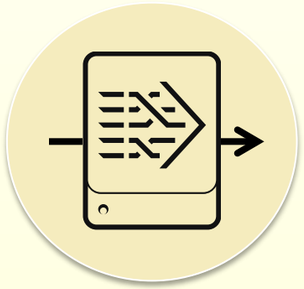
```
[indexAndForward]
index = false
```

- Disable Splunk Web on the HF:

web.conf

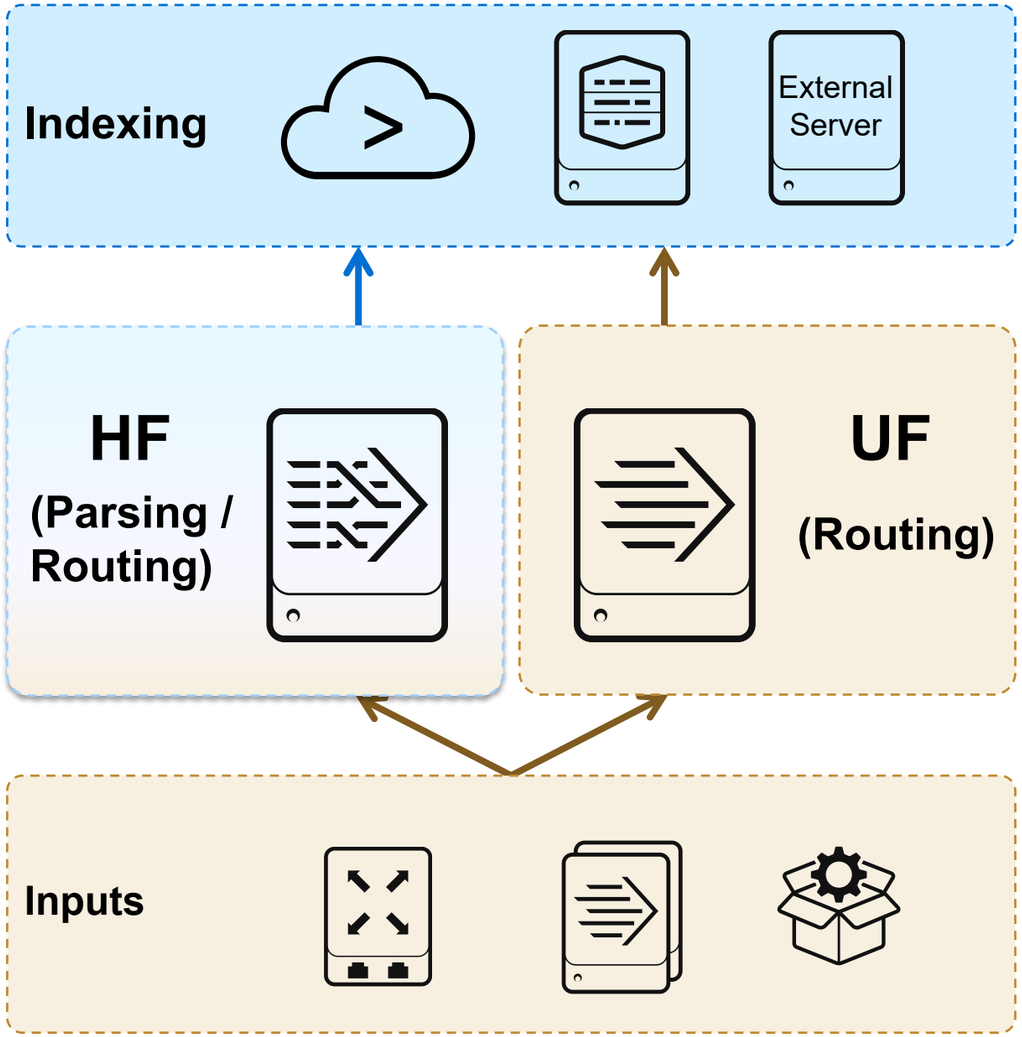
```
[settings]
startwebserver = 0
```


Understanding Intermediate Forwarders



Intermediate Forwarders

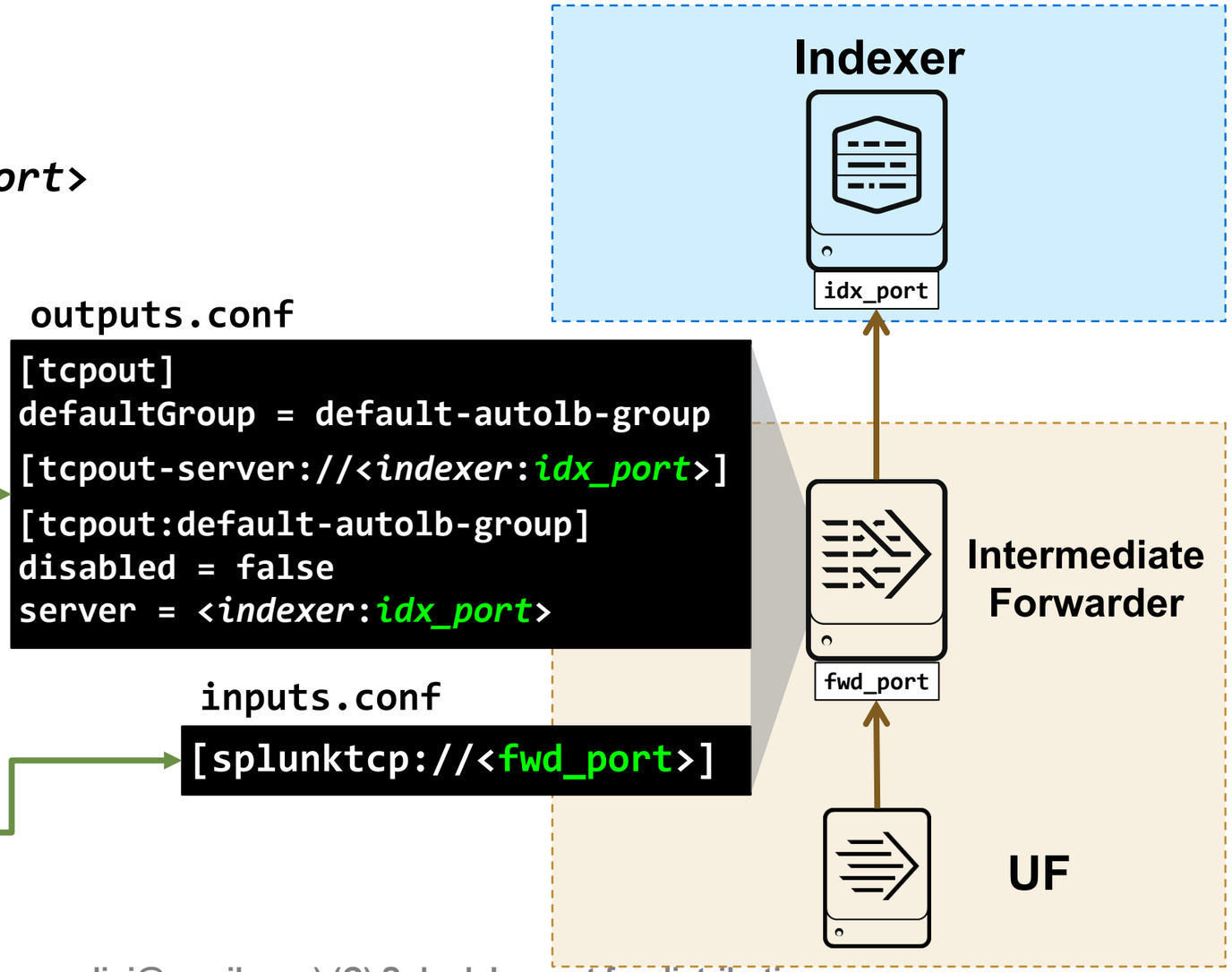
- Can be a Universal or Heavy Forwarder
- Route data from inputs to indexers or other intermediate forwarders
- Can reduce or limit bandwidth on specific network segments
- Can limit security concerns (DMZ, firewalls)
- Can parse, filter or index data if a HF



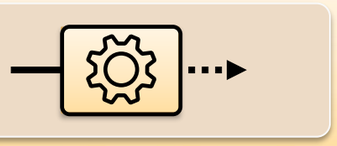
Configuring an Intermediate Forwarder

- Configure forwarding by either:
  - Running:
    `splunk add forward-server:<idx:port>`
  - Modifying **`outputs.conf`**

- Configure receiver by either:
  - Running:
    `splunk enable listen <port>`
  - Modifying **`inputs.conf`**
  - Using Splunk Web (if a HF)



Additional Forwarding Options

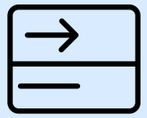


Compressing the feed

Securing the feed

Automatic load balancing to multiple indexers

Indexer acknowledgement to forwarder

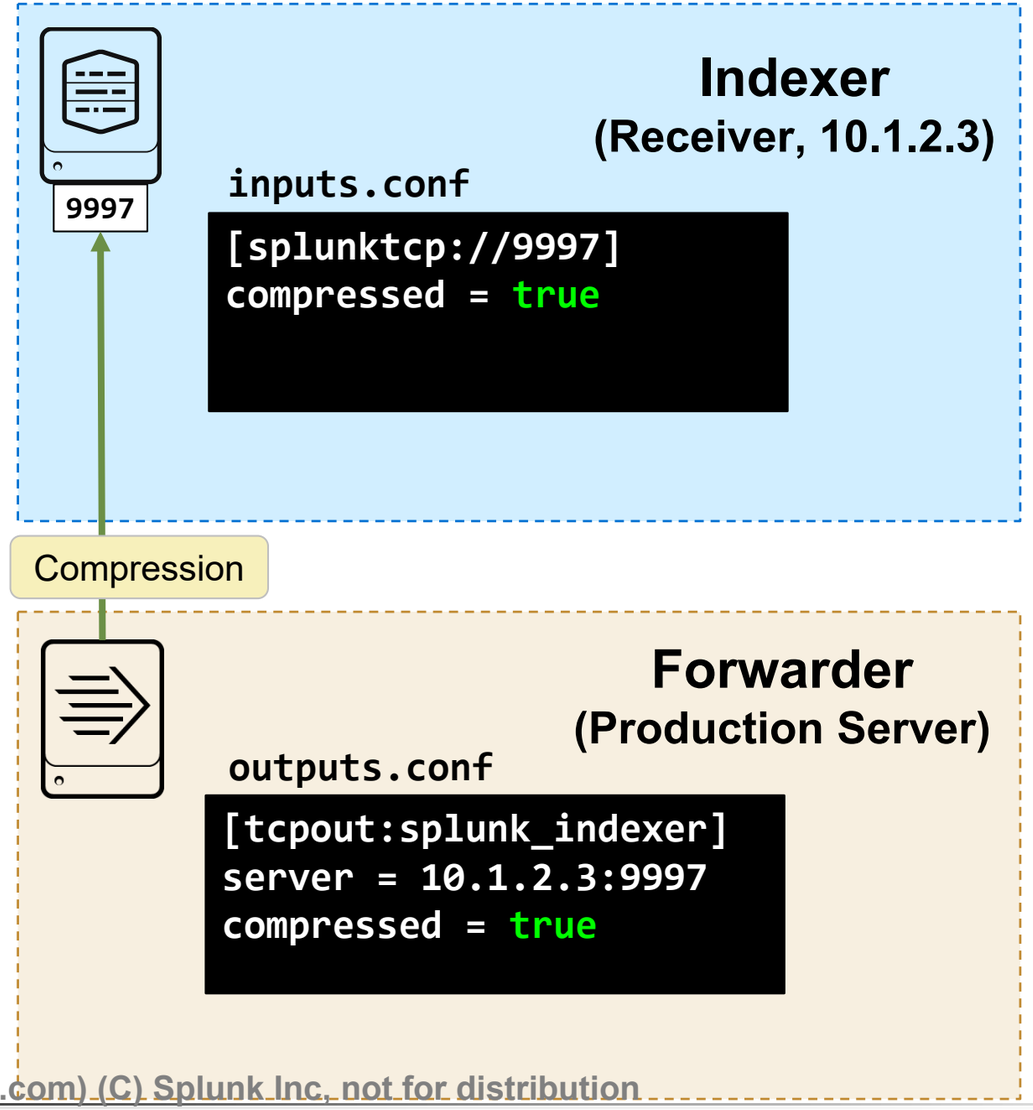


Forwarder queue size

Send the feed over HTTP

Compressing the Feed

- Reduces network utilization
- Increases CPU utilization slightly
- Set either at the forwarder or the indexer
  - Compress select feeds by setting on the forwarder
  - Compress all feeds by setting on the indexer



Securing the Feed with SSL

- Encrypts the feed
- Automatically compresses the feed
- Increases CPU utilization
- Requires the use of certificates
  - To configure with default root certificates:
    - On a *nix indexer: **server.conf**

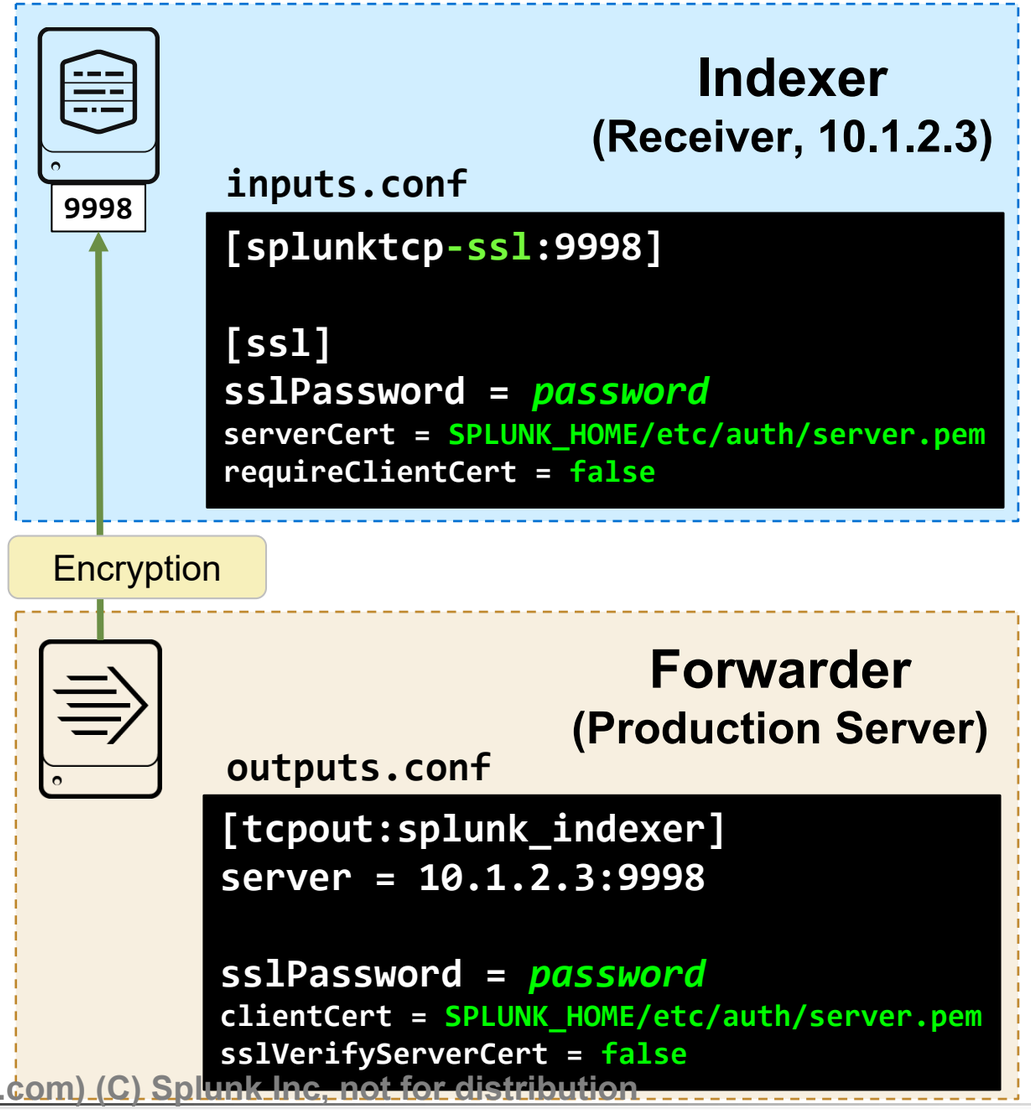
```
[sslConfig]
sslRootCAPath = SPLUNK_HOME/etc/auth/cacert.pem
```

    - On a Windows indexer: Nothing required
    - On a *nix forwarder: **server.conf**

```
[sslConfig]
sslRootCAPath = SPLUNK_HOME/etc/auth/cacert.pem
```

    - On a Windows forwarder: **server.conf**

```
[sslConfig]
caCertFile = cacert.pem
caPath = SPLUNK_HOME\etc\auth
```



Notes About SSL

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

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

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

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

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

wiki.splunk.com/Community:Splunk2Splunk_SSL_DefaultCerts

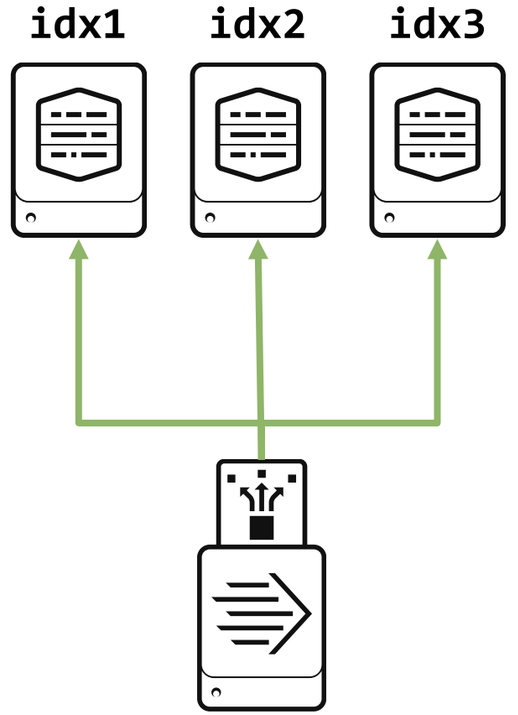
wiki.splunk.com/Community:Splunk2Splunk_SSL_SelfSignedCert_NewRootCA

Automatic Load Balancing

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

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

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



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

Defining Event Boundary on UF

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

Defining Event Boundary on UF (cont.)

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

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

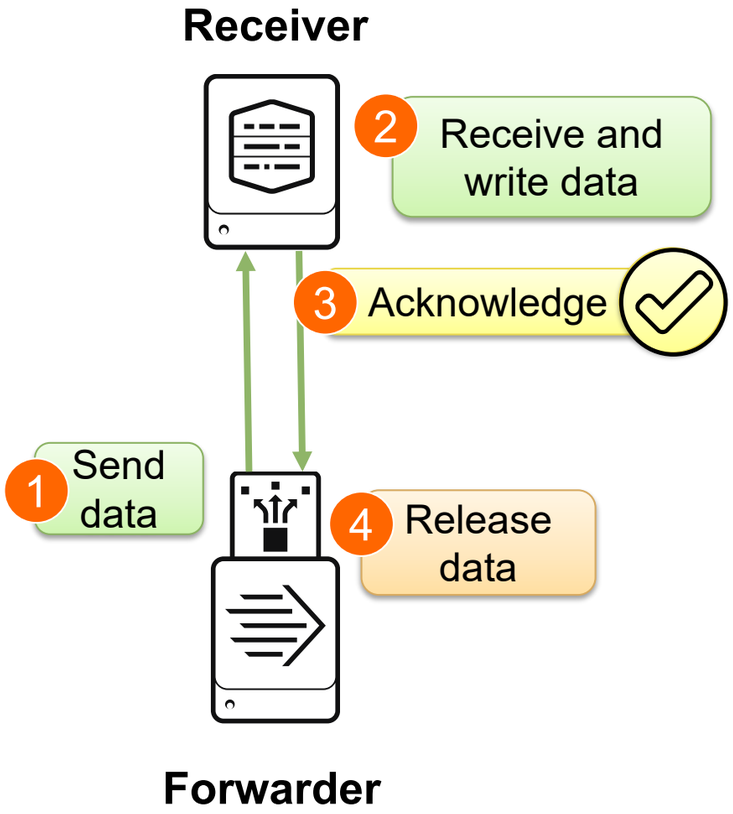
  - Multi-line event

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

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

Indexer Acknowledgement

- Configured in **`outputs.conf`**
  - Disabled by default (**`useACK=false`**)
  - Enabled with **`useACK=true`**
- Guards against loss of forwarded data
  - If no acknowledgement is received, forwarder instead resends the data
- Enable along all segments of data path if using intermediate forwarders

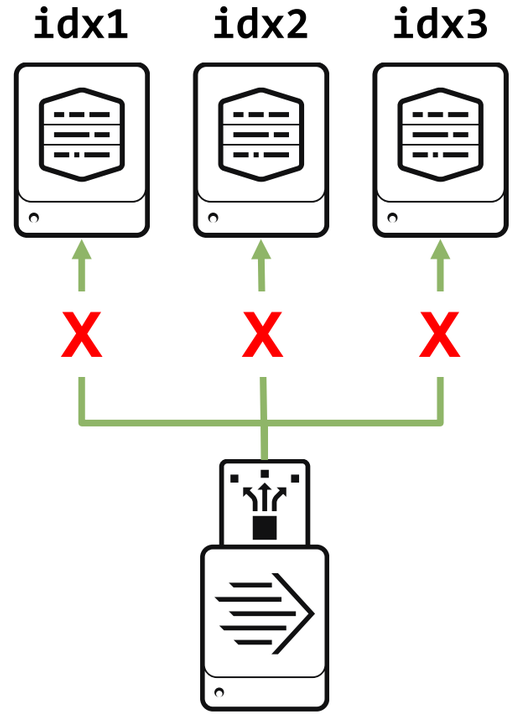


docs.splunk.com/Documentation/Splunk/latest/Forwarding/Protectagainstlossofin-flightdata

Forwarder Queue Size

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

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



Load-balancing forwarder

 | 15 September 2021

Configuring a UF to Send Data over HTTP

- Use cases
  - Use existing network rules for HTTP
  - Easily supports off-the-shelf Load Balancers
- Limitations:
  - UF performs httpout or tcpout, but not both simultaneously
  - No support for indexer acknowledgements
- To break events on the UF for sending over HTTP:

outputs.conf

```
[httpout]
httpEventCollectorToken = <authToken>
uri = https://<ip>:8088
batchSize = 65536          (default: 64 KB)
batchTimeout = 30          (default: 30 sec)
```

props.conf

```
LB_CHUNK_BREAKER = ([\r\n]+)     (default)
```

Forwarding Resources

- Overview of forwarders

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

- Forwarder deployment overview

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

- Splunk Blog: Universal or Heavy, that is the question?

  www.splunk.com/en_us/blog/tips-and-tricks/universal-or-heavy-that-is-the-question.html

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

  wiki.splunk.com/Deploying_Splunk_Light_Forwarders

Useful Commands

Command	Operation
From the Forwarder:	
`splunk add forward-server`	Configures the forwarder to send data to the receiver
`splunk list forward-server`	Displays the current receivers
`splunk remove forward-server`	Removes the receiver 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?
- Which installer will the System Admin use to install the heavy forwarder?
- True or False. The UF and the HF can be used to mask data before transmitting to indexers.

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. To prevent sending a partial event to an indexer, the forwarder waits for an EOF or a pause in I/O 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

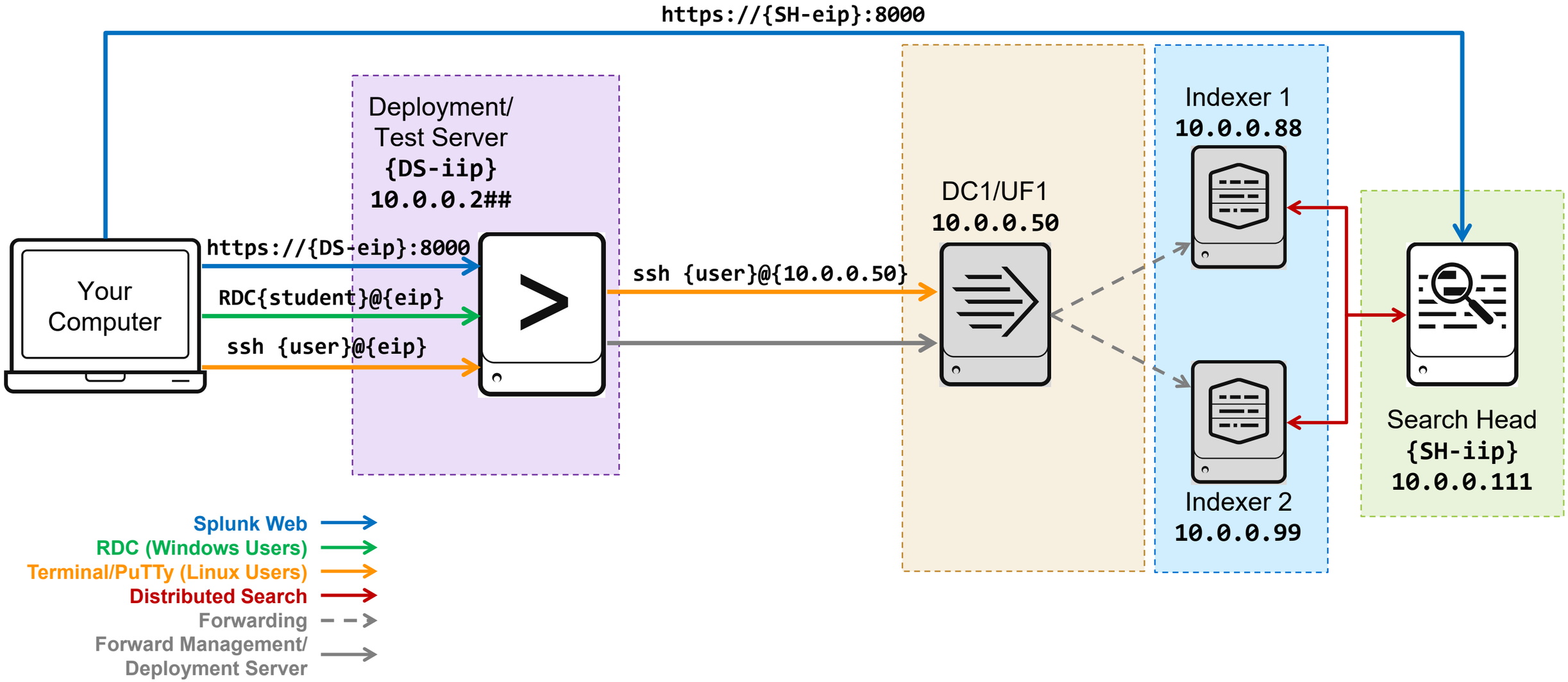
- Which installer will the System Admin use to install the heavy forwarder?

  Splunk Enterprise

- 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.

Module 3 Lab Exercise – Environment Diagram



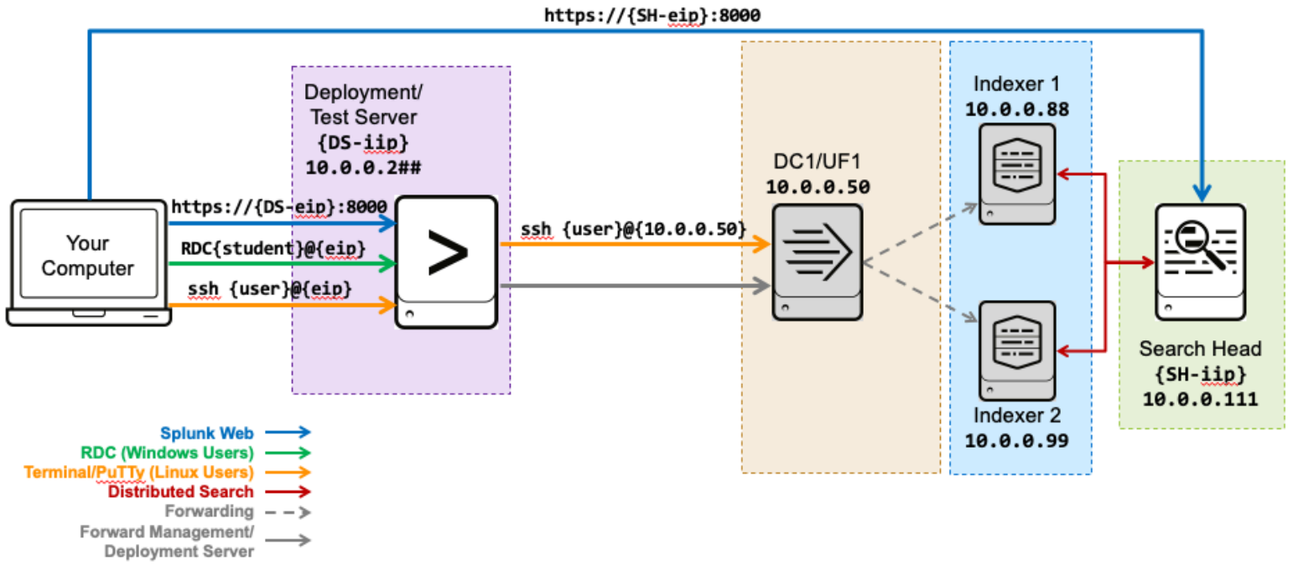
Module 3 Lab Exercise

Time: 20-25 minutes

Description: Set up forwarders

Tasks:

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



Note

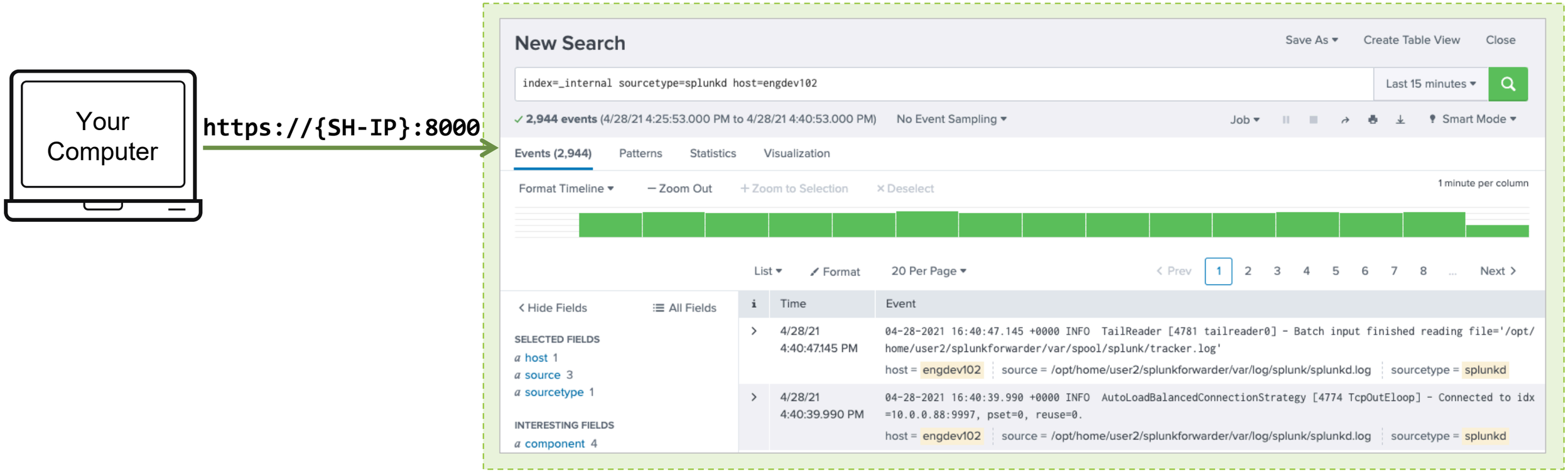
You have a login on a remote Linux host that is your forwarder.

This lab exercise only establishes the connection between your UF and indexer.

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

Verification: Run a search to get forwarded internal logs from UF1

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

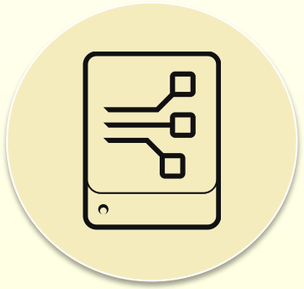


Module 4:
Forwarder Management

Module Objectives

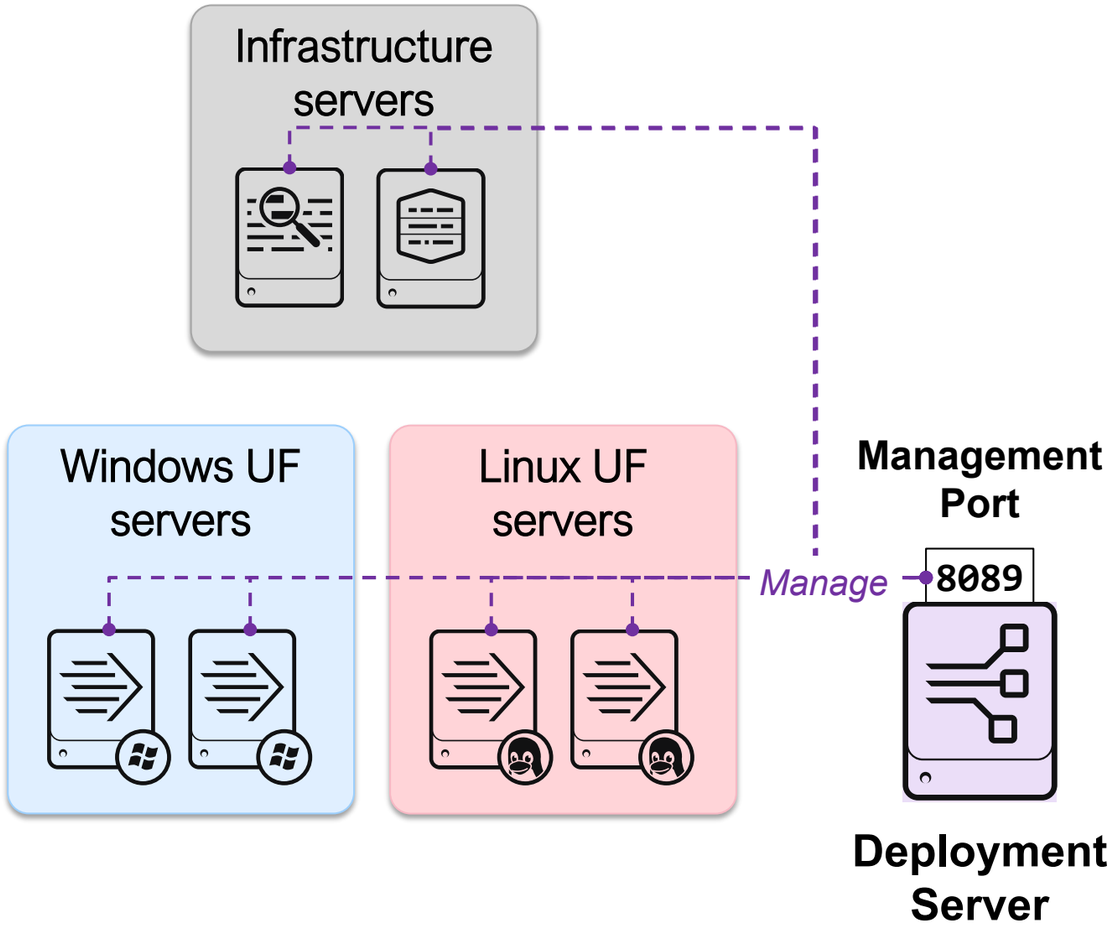
- Describe Splunk Deployment Server (DS)
- Manage forwarders using deployment apps
- Configure deployment clients and client groups
- Monitor forwarder management activities

Understanding the Deployment Server



Deployment Server (DS)

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



Deployment Server Components

Deployment Apps

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

Server Classes

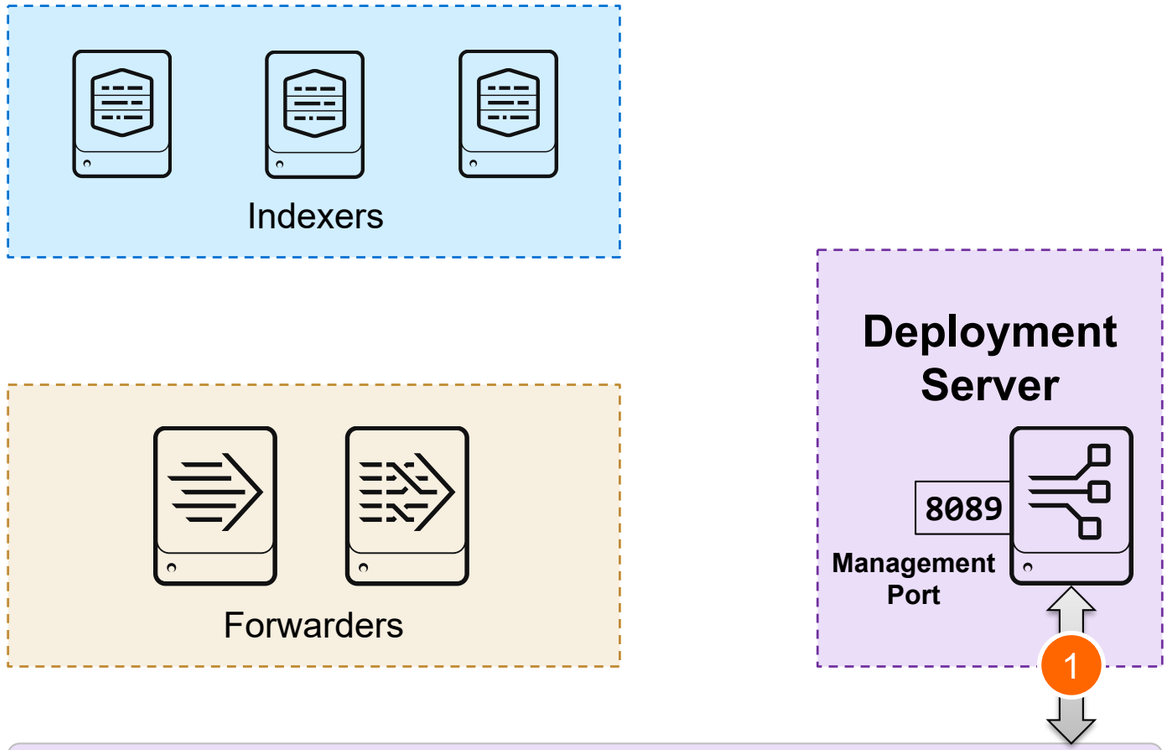
- Groupings of deployment clients
- Define what apps should be deployed to which clients
- Saved in **`serverclass.conf`**

Deployment Clients

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

Deployment Server Configuration (1)

1. Configure DS, server classes, and app packages



Configuration on DS

Map clients to apps:

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

App repository:

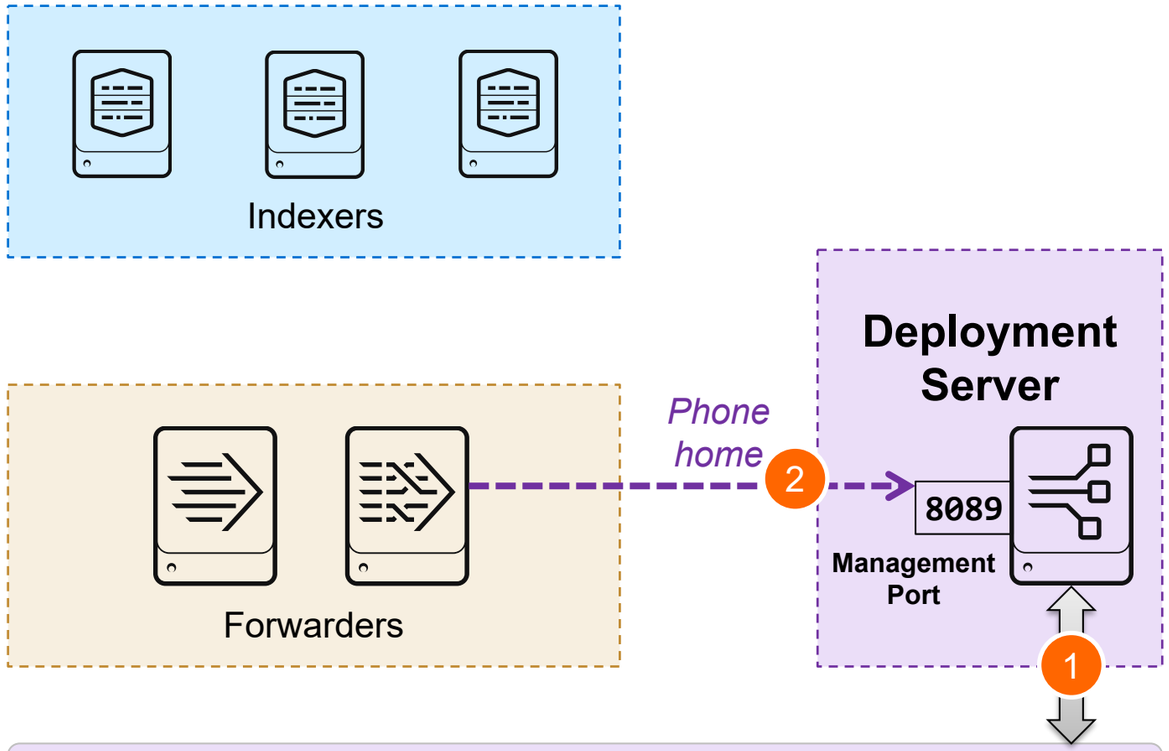
SPLUNK_HOME/etc/deployment-apps/<*app*>/local

Apps/configs to deploy:

outputs.conf, **inputs.conf**, etc.

Deployment Server Configuration (2)

1. Configure DS, server classes, and app packages
2. Use **`deploymentclient.conf`** to configure instances as deployment clients; phones home to DS



Configuration on DS

Map clients to apps:

`SPLUNK_HOME/etc/apps/<app>/local/serverclass.conf`

App repository:

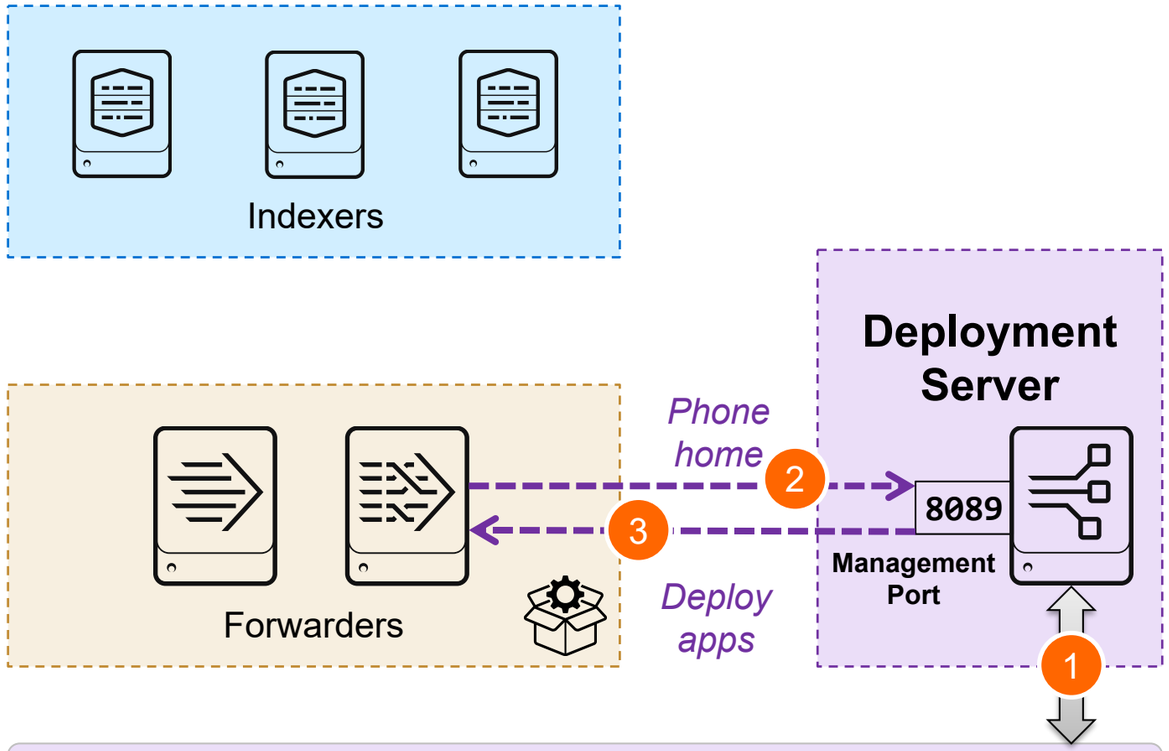
`SPLUNK_HOME/etc/deployment-apps/<app>/local`

Apps/configs to deploy:

`outputs.conf`, **`inputs.conf`**, etc.

Deployment Server Configuration (3)

1. Configure DS, server classes, and app packages
2. Use **`deploymentclient.conf`** to configure instances as deployment clients; phones home to DS
3. Client downloads subscribed apps as directed by server classes on DS



Configuration on DS

Map clients to apps:

`SPLUNK_HOME/etc/apps/<app>/local/serverclass.conf`

App repository:

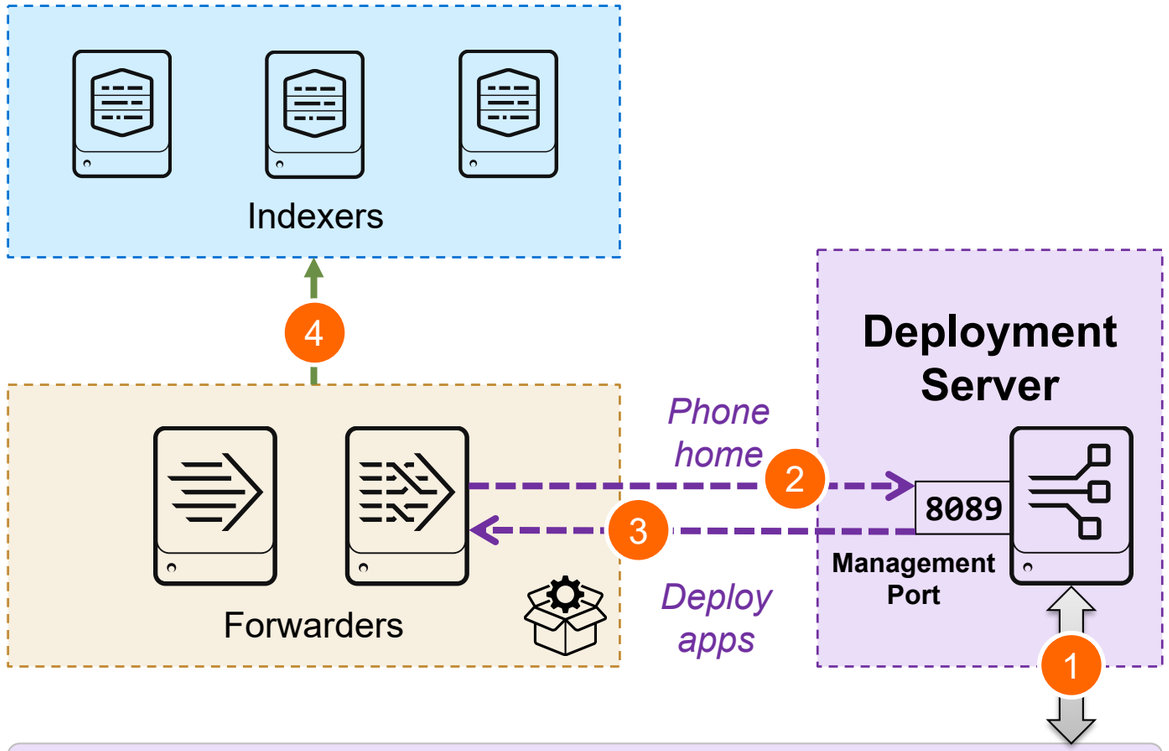
`SPLUNK_HOME/etc/deployment-apps/<app>/local`

Apps/configs to deploy:

`outputs.conf`, **`inputs.conf`**, etc.

Deployment Server Configuration (4)

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



Configuration on DS

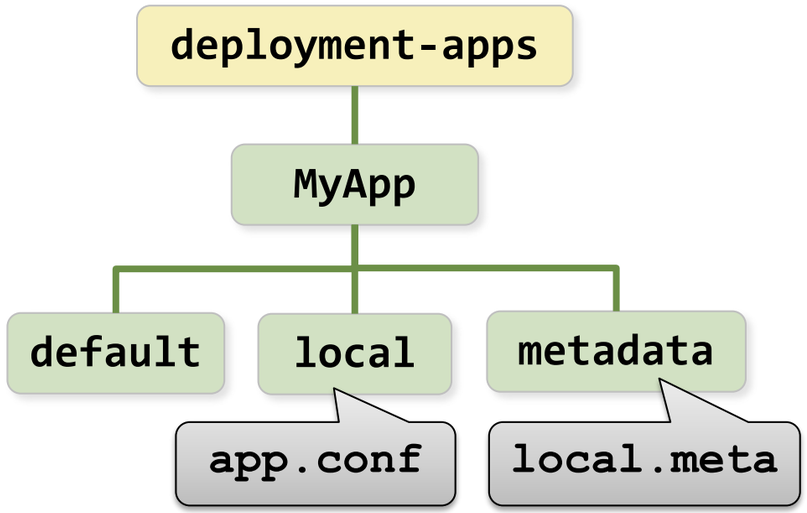
Map clients to apps:
`SPLUNK_HOME/etc/apps/<app>/local/serverclass.conf`

App repository:
`SPLUNK_HOME/etc/deployment-apps/<app>/local`

Apps/configs to deploy:
`outputs.conf`, **`inputs.conf`**, etc.

Configuring a Deployment App

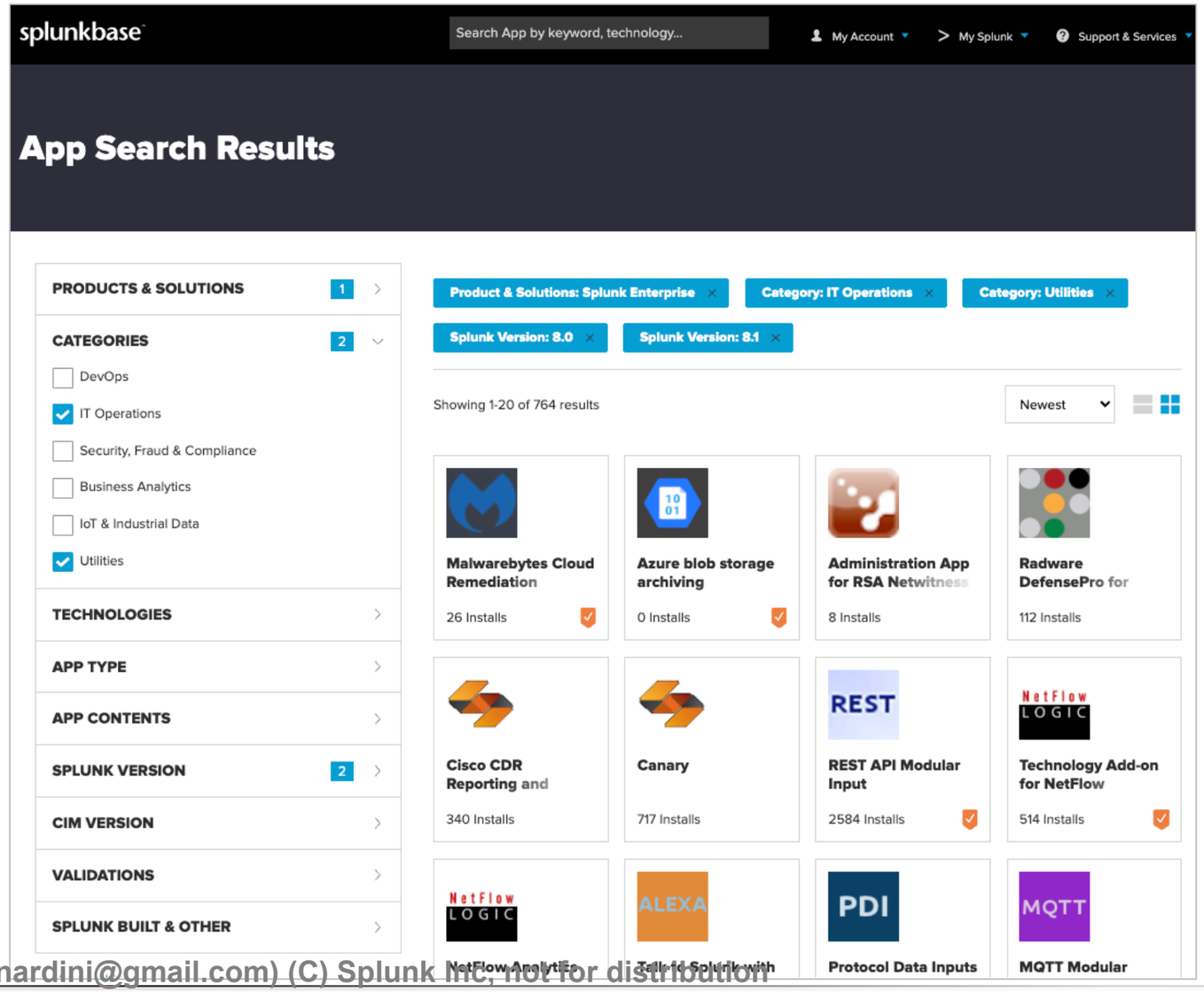
- Follows app structure and rules
  - Place files in **SPLUNK_HOME/etc/deployment-apps**
  - Required files:
    - **app.conf** (in **default** or **local**)
    - **local.meta** (in **metadata**)
  - Optionally may contain configuration files, scripts, and other resources



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

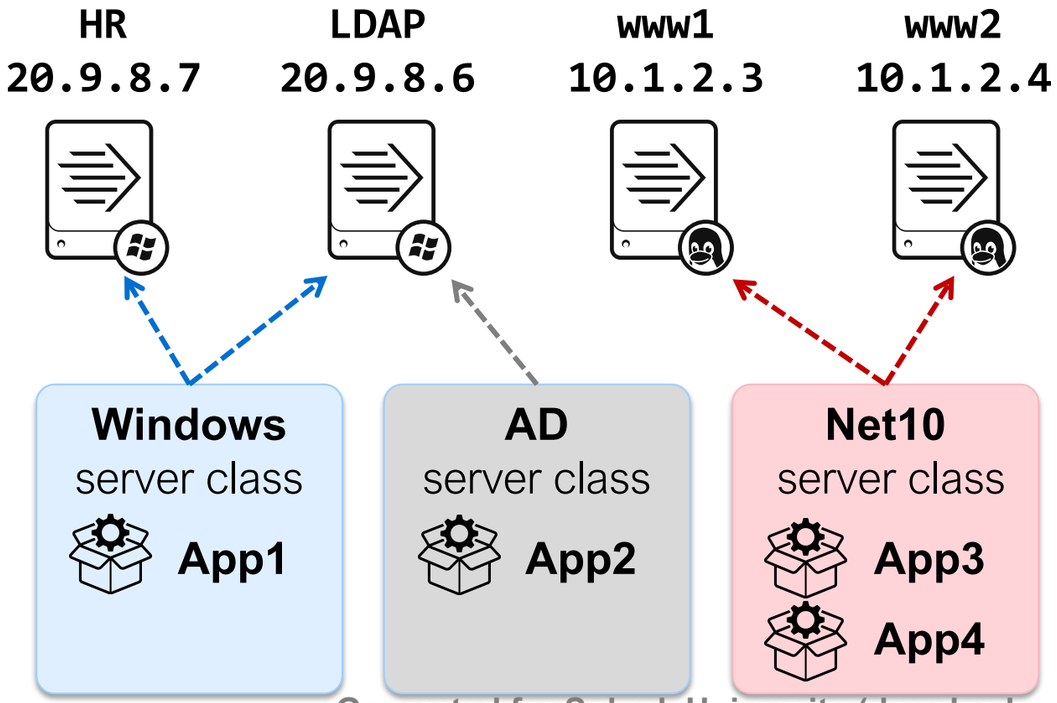
Apps and Add-ons

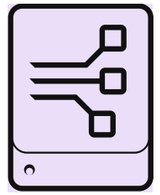
- Can be downloaded from Splunkbase
- Installed on a Splunk instance:
  - Using the Deployment Server
  - Using CLI on the instance
  - Manually by installing the app
- Deploy to **`SPLUNK_HOME/etc/apps`**
- Comes with documentation for details about settings for **`inputs.conf`**, and so on



What's a Server Class?

- Maps groups of clients to deployment apps
  - Can be based on client name, host name, IP address, DNS name, or machine types





Deployment Server

Server class	Rules
Windows	• Assigned to Windows systems • Installs **App1**
AD	• Assigned to Active Directory servers • Installs **App2**
Net10	• Assigned to hosts on **`10.1.2.*`** subnet • Installs **App3** and **App4**

Enabling Forwarder Management

1. On deployment server: Add one or more apps in **SPLUNK_HOME/etc/deployment-apps**
2. On forwarders: Set up the deployment client
   - Run **splunk set deploy-poll <*deployment_server*:*splunkd_port*>**
   - Run **splunk restart**
3. In Forwarder Management UI: Create one or more server classes
4. On deployment server: Verify deployment clients and deployment status
5. On forwarders: Verify **SPLUNK_HOME/etc/apps** folder for deployed apps

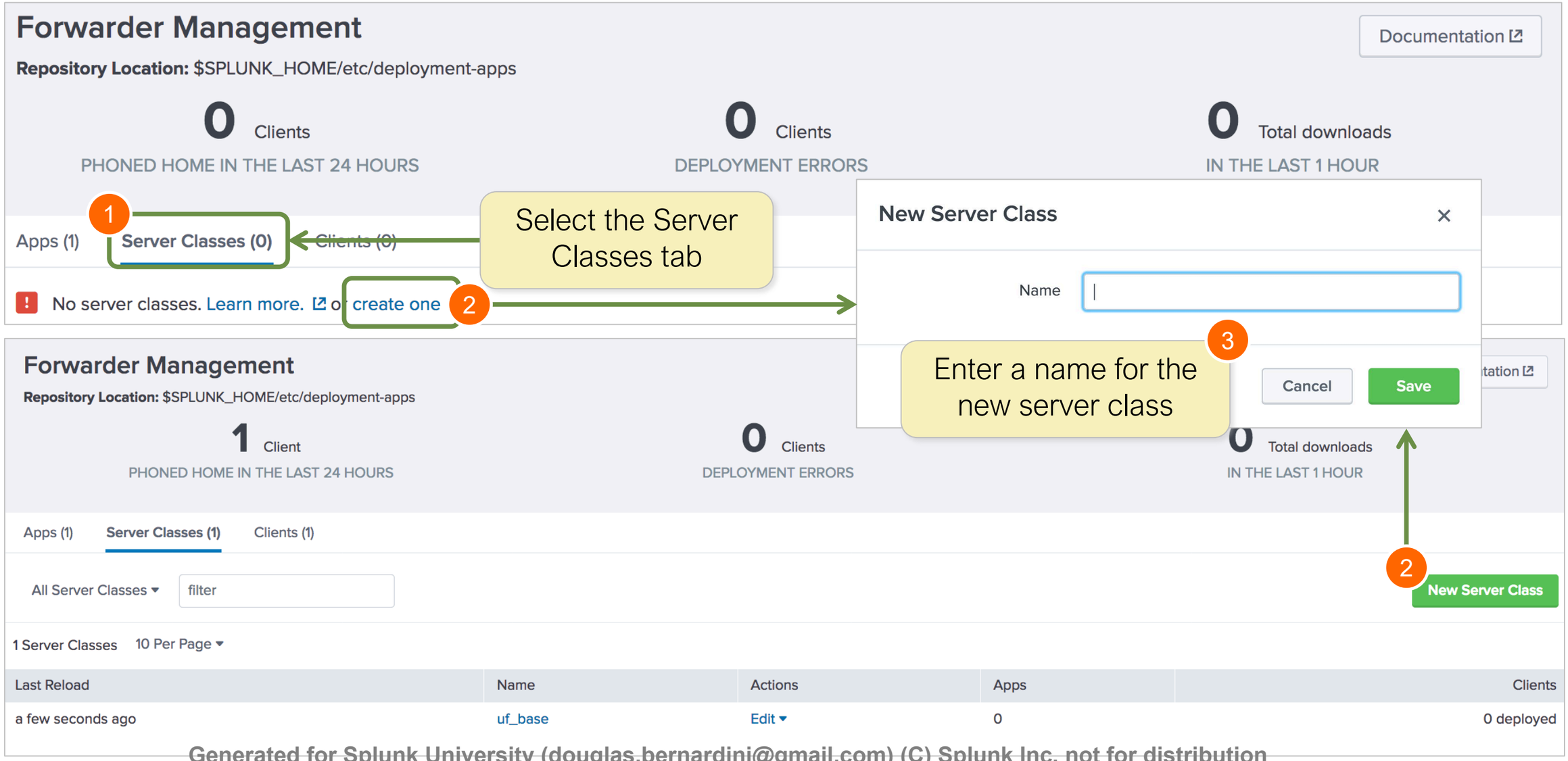
Configuring Deployment Clients

- On prospective deployment clients (usually forwarders):
  1. Run: **`splunk set deploy-poll <deployment_server:splunkd_port>`**
     - Creates **`deploymentclient.conf`** in **`SPLUNK_HOME/etc/system/local`**
     - Alternatively create **`deploymentclient.conf`** manually
  2. Restart the deployment clients: **`splunk restart`**
- Edit **`[deployment-client]`** stanza to override defaults
  - Can be part of initial deployment app
  - Contains phone home setting (default: 60 seconds)

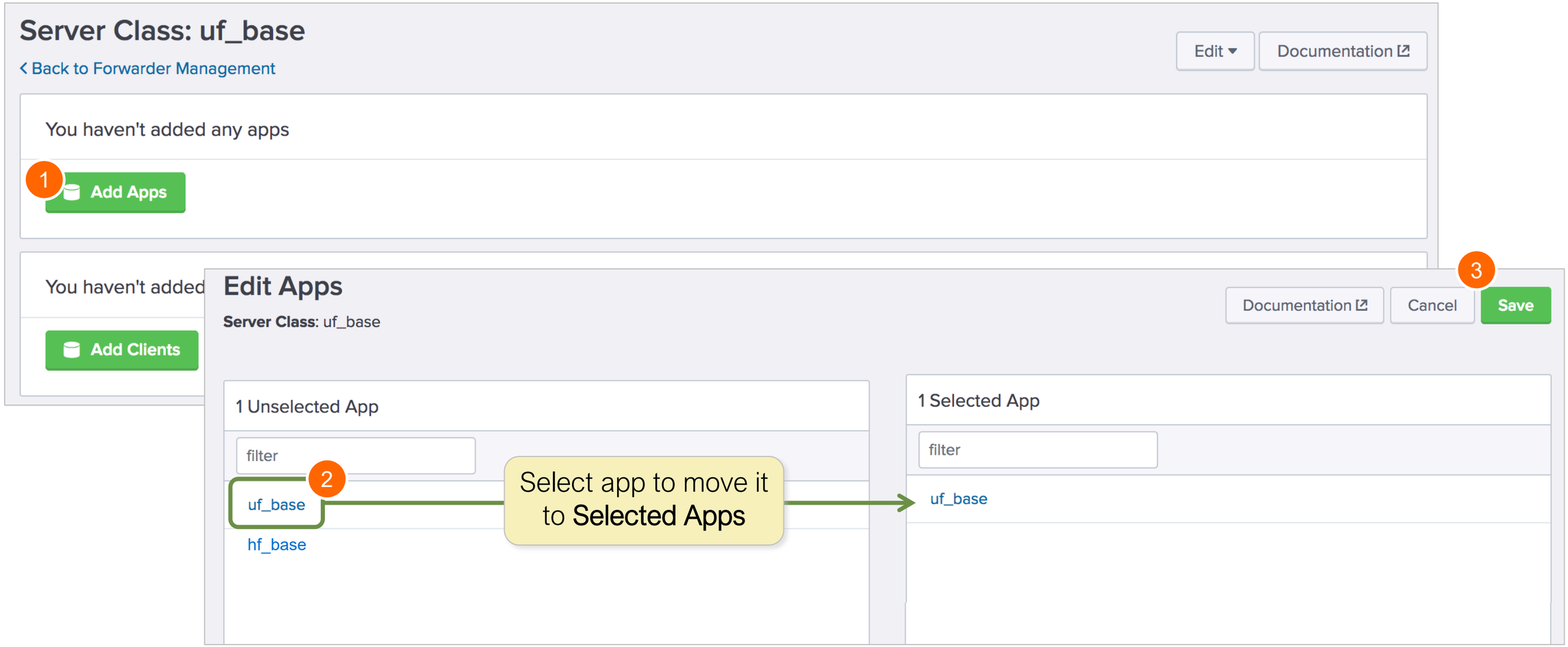
`deploymentclient.conf`

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

...

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

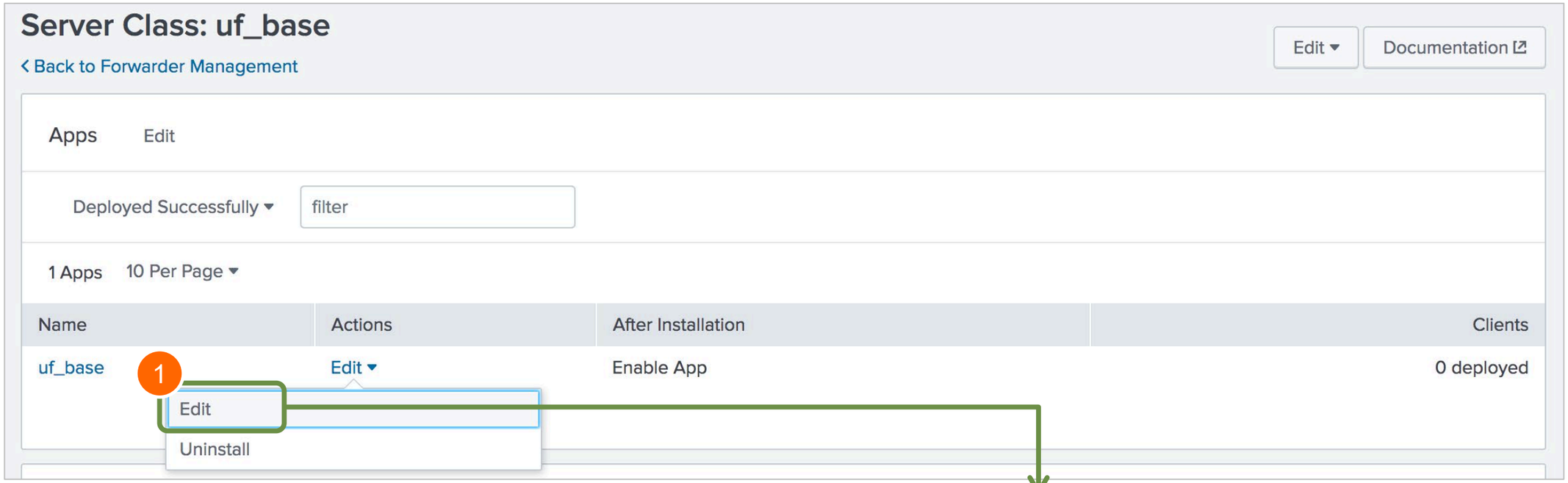

Adding a Server Class

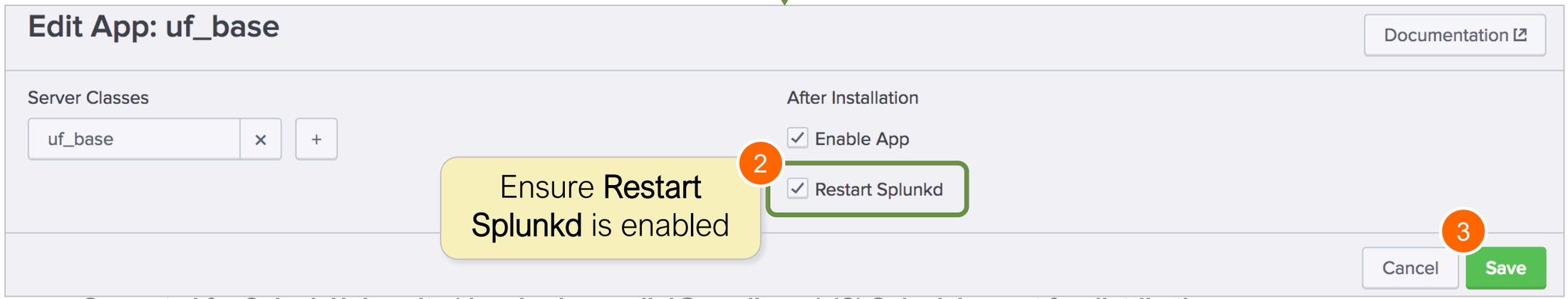


Selecting Apps for the Server Class

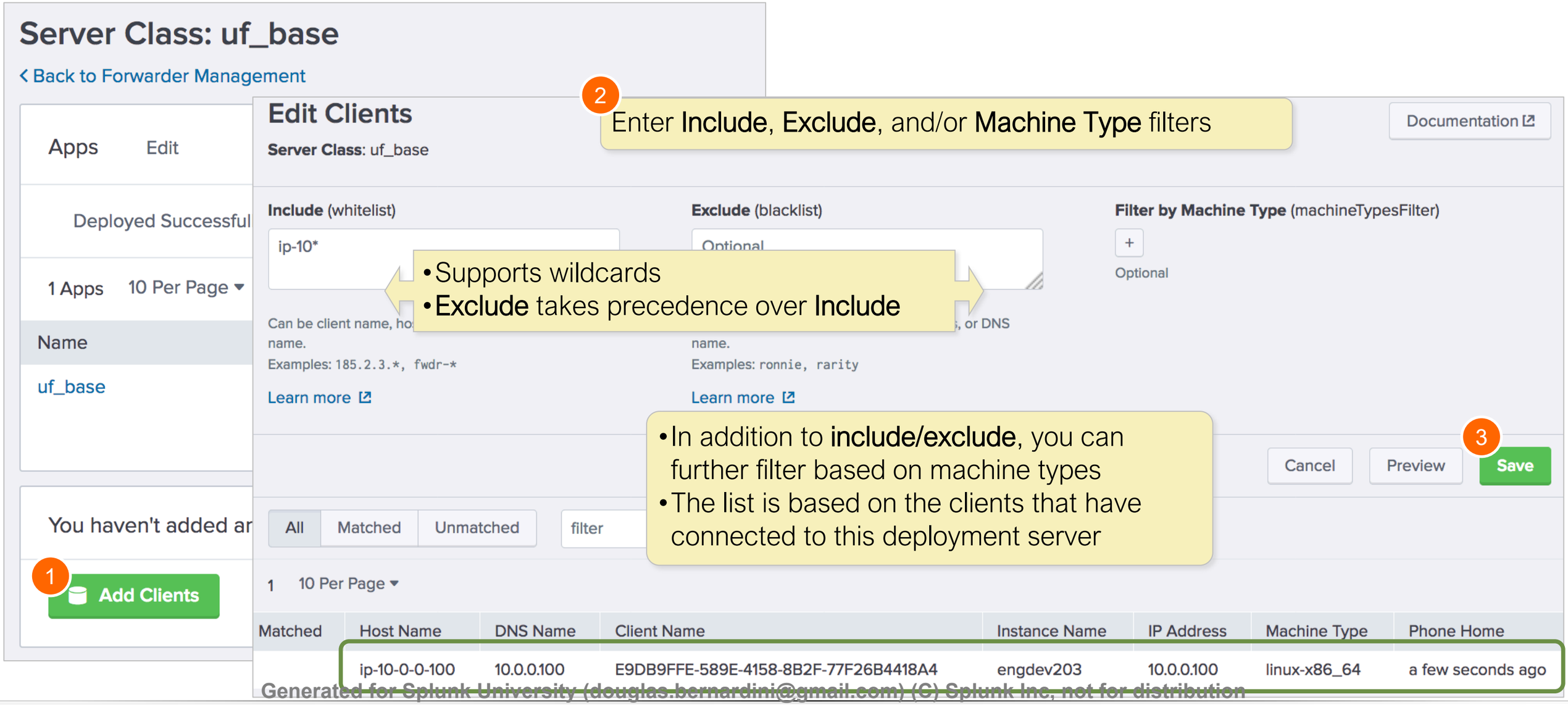


Post Deployment Behavior Setting





Selecting Clients for the Server Class



Verifying Clients Are Receiving Apps

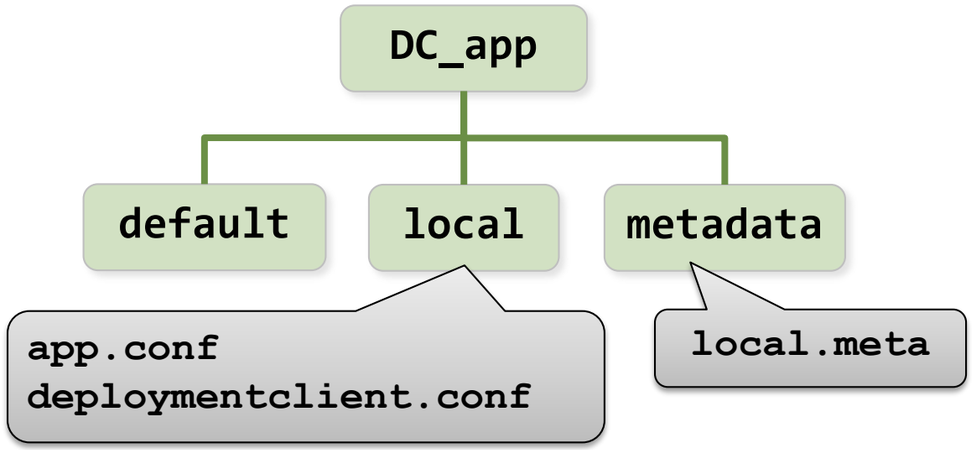
- Confirm expected app directories and contents
  - As **`SPLUNK_HOME/etc/apps/`*`app_name`*** on deployment clients
- App changes on DS causes client to reload
  - Occurs after client's next phone-home
  - To change the app settings using Forwarder Management, use app's **Edit** menu associated with the server class
  - To change inputs for an app: **Settings > Data Inputs > Forwarded Inputs**
- Set post-deployment behavior to automatically restart the forwarder
- To troubleshoot the deployment client
  - Check the deployment server settings: **`splunk show deploy-poll`**

Reload Deployment Server

- DS uses checksums to compare app on server with client
- Checksums are updated during Splunk start
- Issue:
  - DS is unaware if deployed app configuration files are edited manually
  - Restarting Splunk on DS may be costly
- Solution:
  - Run **`splunk reload deploy-server`** on the DS to re-cache the deployable apps and update checksums (without Splunk restart).
  - Next time client phones home, app checksums are different, causing the app to be re-deployed

Manage Deployment Client Settings Centrally

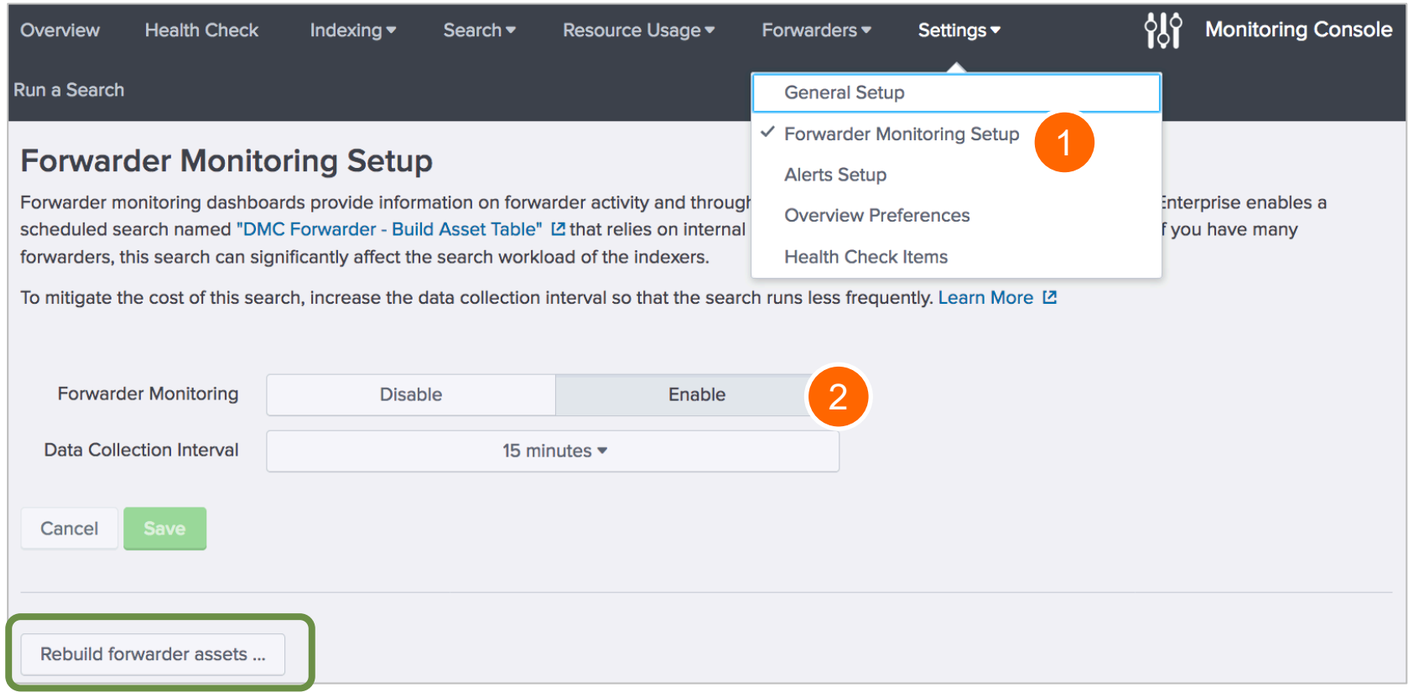
- Use an app to manage deployment client settings
  - Create a deployment client settings app (example: **DC_app**)
  - Move **deploymentclient.conf** settings from **etc/system/local/** to **etc/apps/DC_app/local/**
  - Deploy **DC_app** to clients using a Server Class



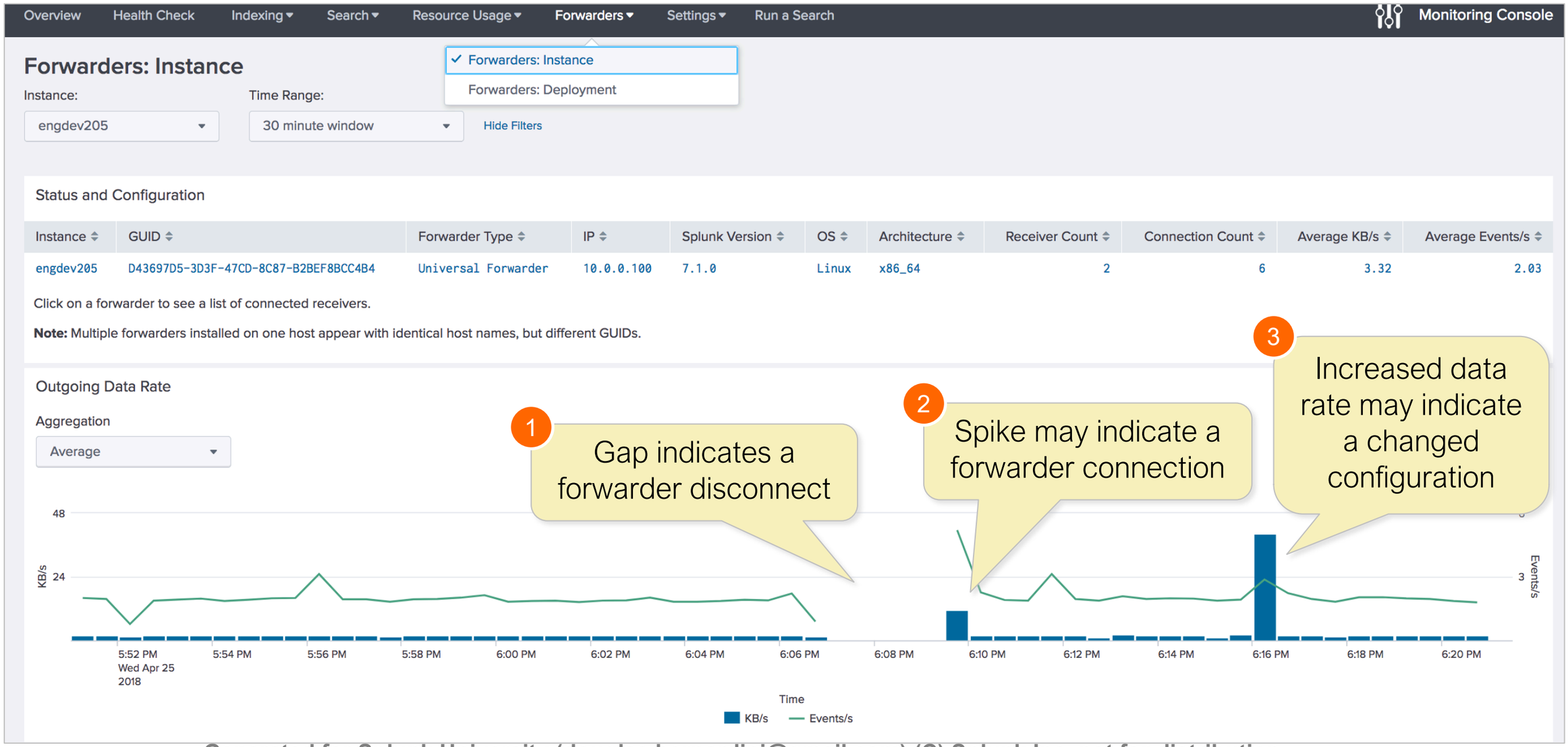
splunk> Best Practice

Forwarder Monitoring with Monitoring Console

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



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 list forward-server`	Displays the current forward server configuration
`splunk disable deploy-client`	Disables the deployment client
From the Deployment Server (DS):	
`splunk reload deploy-server`	Checks all apps for changes and notifies the relevant clients the next time they phone home
`splunk list deploy-clients`	Displays information about the deployment clients

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.

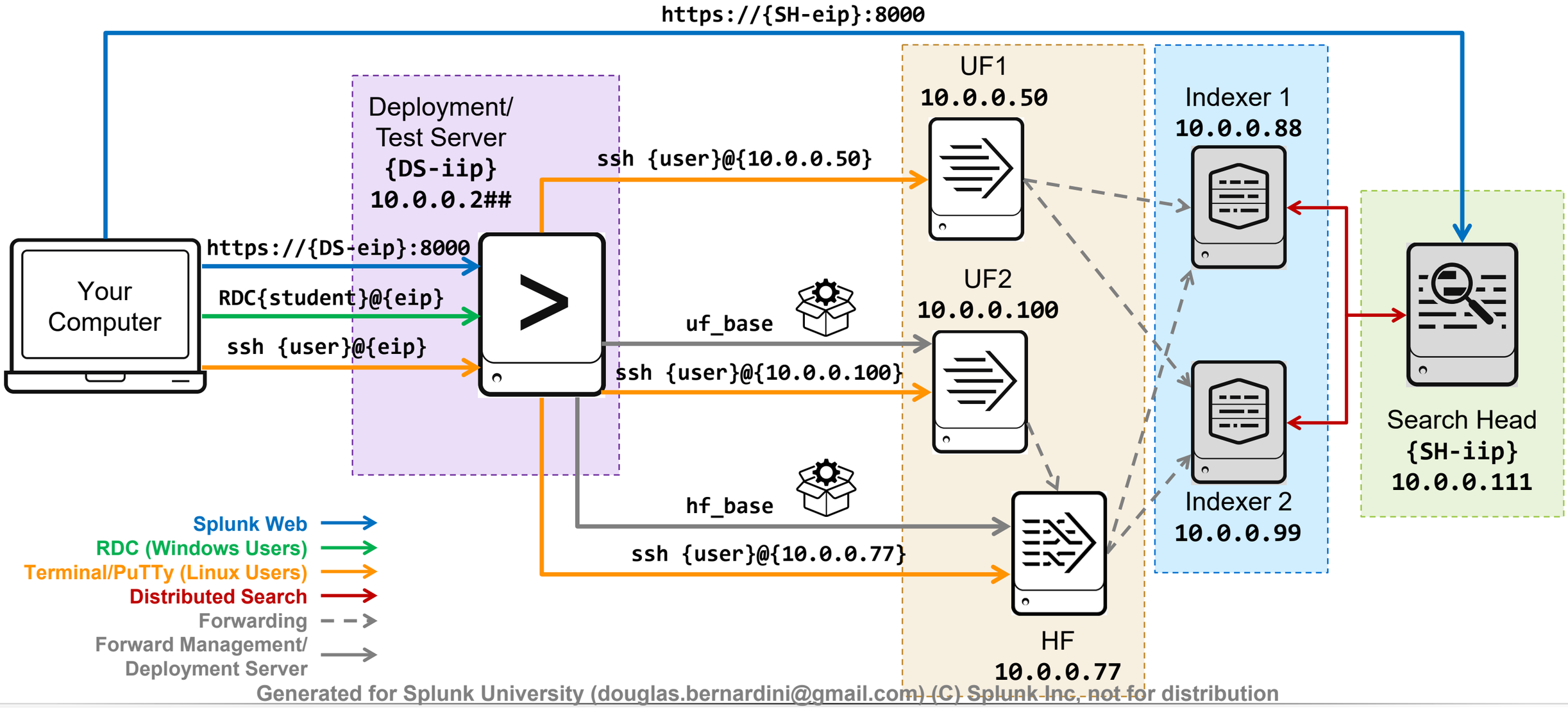
- When an app is deployed from the Deployment Server to the client, where will you find that app on the client by default?

  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 its management port (**8089** by default.)

Module 4 Lab Exercise – Environment Diagram



Module 4 Lab Exercise

Time: 25-30 minutes

Description: Configure Forwarder Management

Tasks:

- Copy deployment apps to the DS folders
- Configure UF2 as a deployment client
- Enable listening port on HF (as an intermediate forwarder)
- Configure the HF as a deployment client
- Create two server classes to manage UF2 and the HF from the DS
- Confirm deployment of deployment apps on UF2 and HF

Module 5:
Monitor Inputs

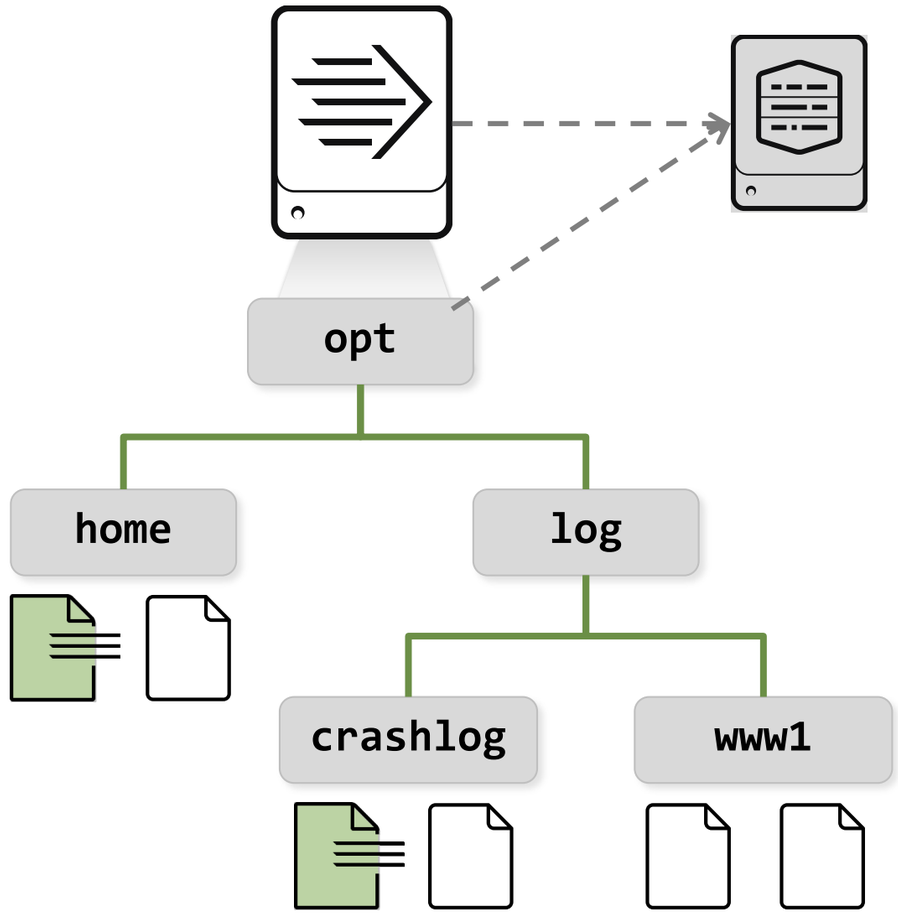
Module Objectives

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

Monitoring Input Files

Monitoring Files

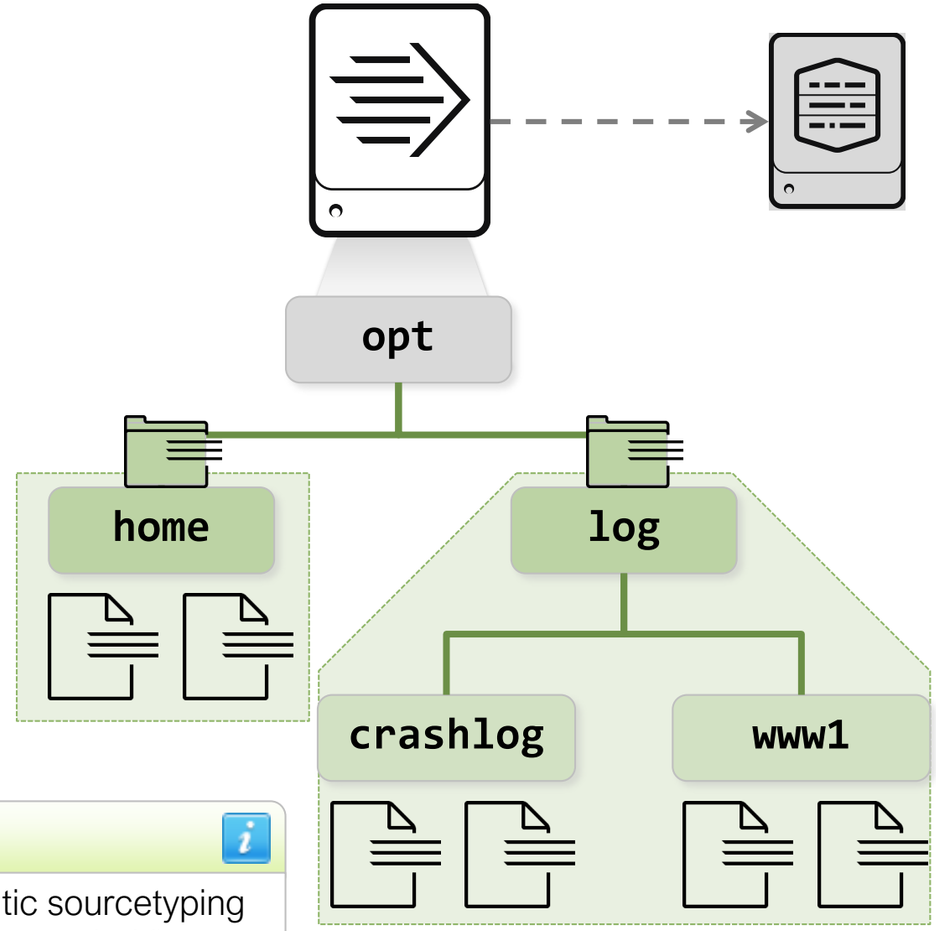
- Defines a single file as the source, with input settings (**sourcetype**, **index**, **host**, etc.)
- Ingests current contents of the file
- Continuously monitors for new content using the Splunk Fishbucket to keep a checkpoint
- Supports any text format, such as: plain text, structured text (**`CSV`**,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



Monitoring Input Directories

Monitoring Directories

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



Note

Automatic sourcetyping is recommended for directories with mixed file types.

Monitor Input Options in **inputs.conf**

- Defining the source
  - Place after **monitor://** in stanza header
  - Absolute path to a file or directory
  - Can contain wildcards
- Defining attributes
  - All attributes are optional
  - Default **host** is defined in **SPLUNK_HOME/etc/system/local/inputs.conf**
  - Omitting **sourcetype** causes Splunk to try to determine it automatically
- For more attributes and documentation
  - See **inputs.conf.spec** in **SPLUNK_HOME/etc/system/README**

inputs.conf format:

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

Example **monitor** path entries:

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

[monitor:///var/log/]

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

[monitor://C:\logs\]
```

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

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

File and Directory Matching

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

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

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

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

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

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

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

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

✓ Matches
✗ Doesn't match

Additional Options

Follow tail (`followTail`)

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

Ignore older than (`ignoreOlderThan`)

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

Whitelist and Blacklist

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

Example: Using Whitelist to Include Files

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

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

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

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

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

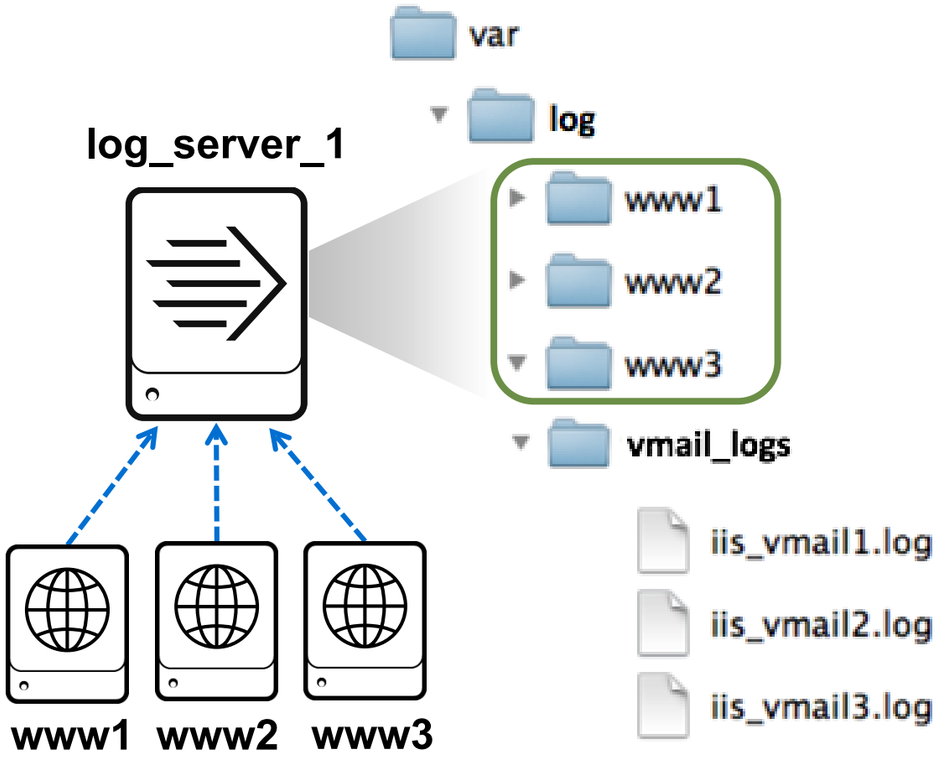
```
[monitor:///var/log/www1/]
whitelist = /query\.log$|/my\.log$
```

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

✓ Matches
✗ Doesn't match

Overriding the Host Field

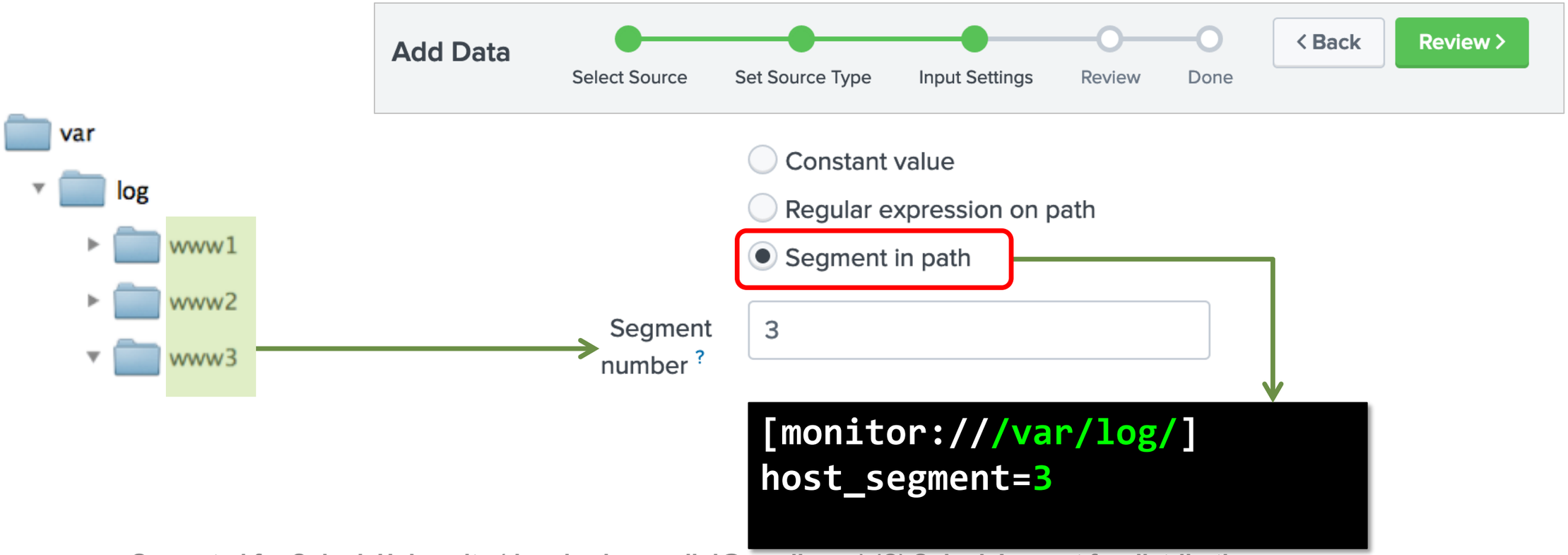
- When data is stored on a different server than its origin
  - Example: A web farm where each web server stores its log file on a centralized file server
- By explicitly setting the host
  - Using a specified value
  - Using a directory name
  - Using a regular expression



Setting the Host With a Directory Name

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

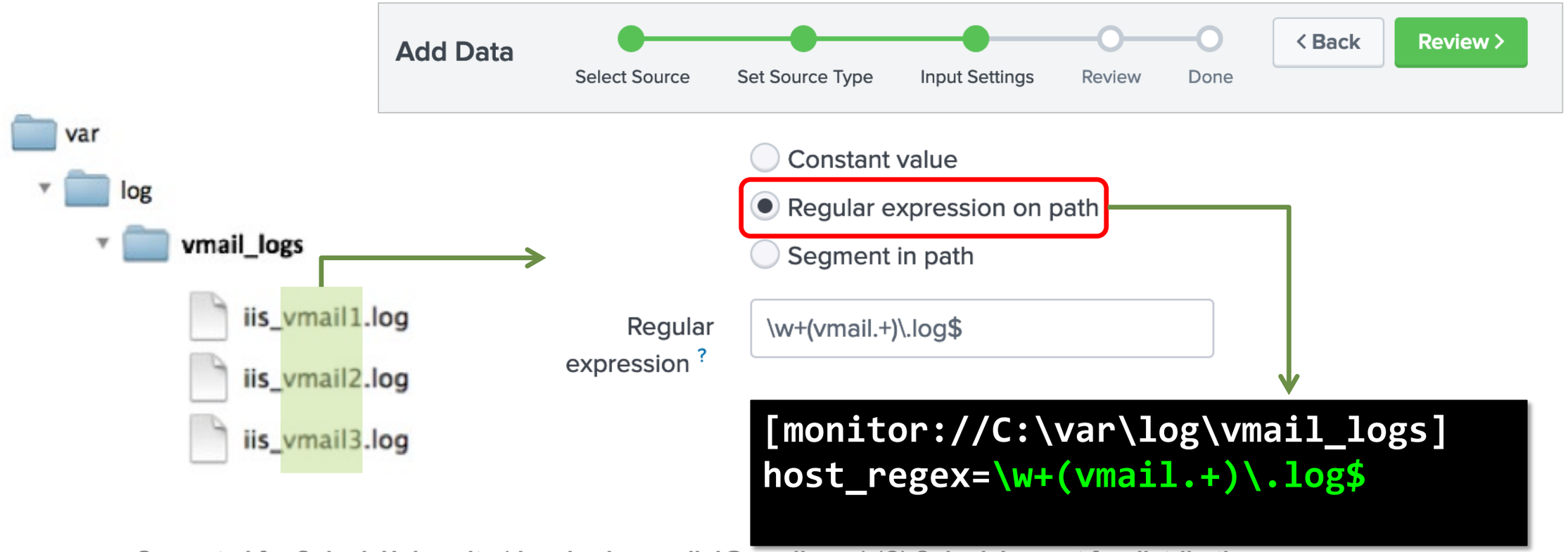
  Example: Setting **`host_segment`** to **`3`** uses the 3rd segment of the directory path as the host name for files in that directory



Setting the Host With a Regular Expression

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

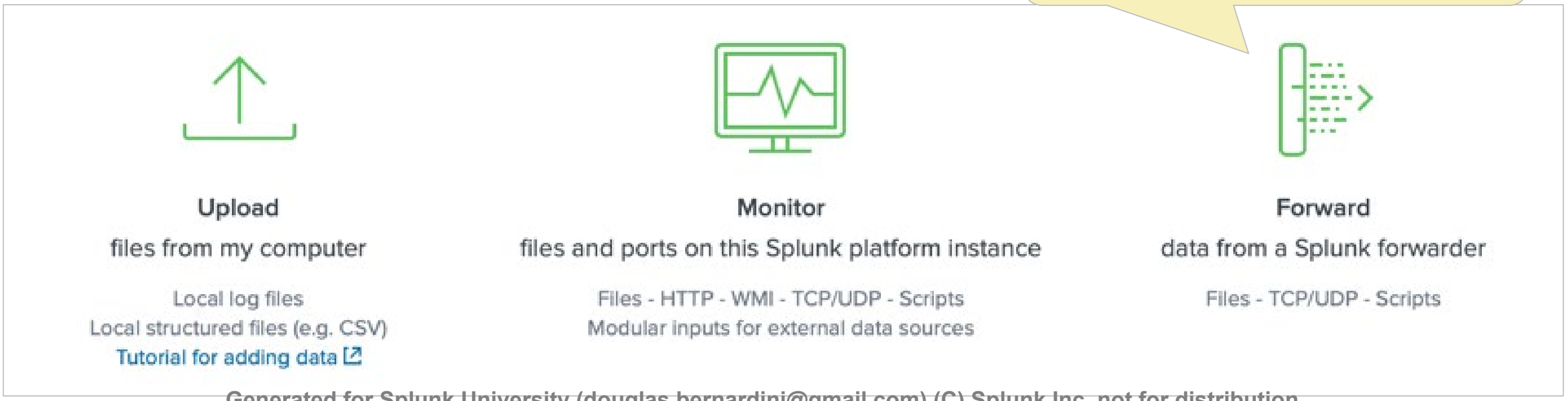
  Example: Setting **`host_regex`** to **`\w+(vmail.+)\.log$`** selects the second part of the log file name as its host name



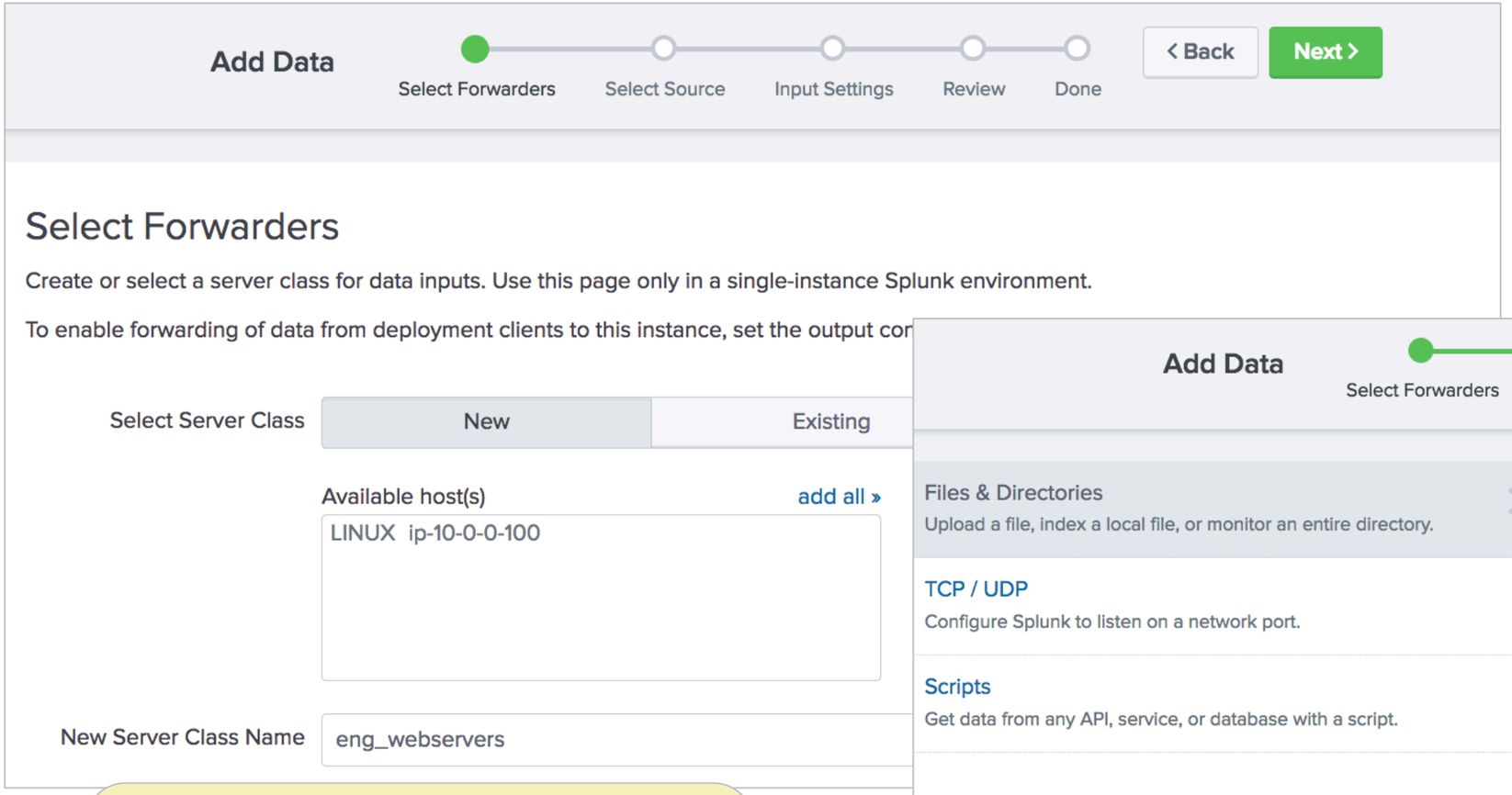
Creating Forwarded Inputs

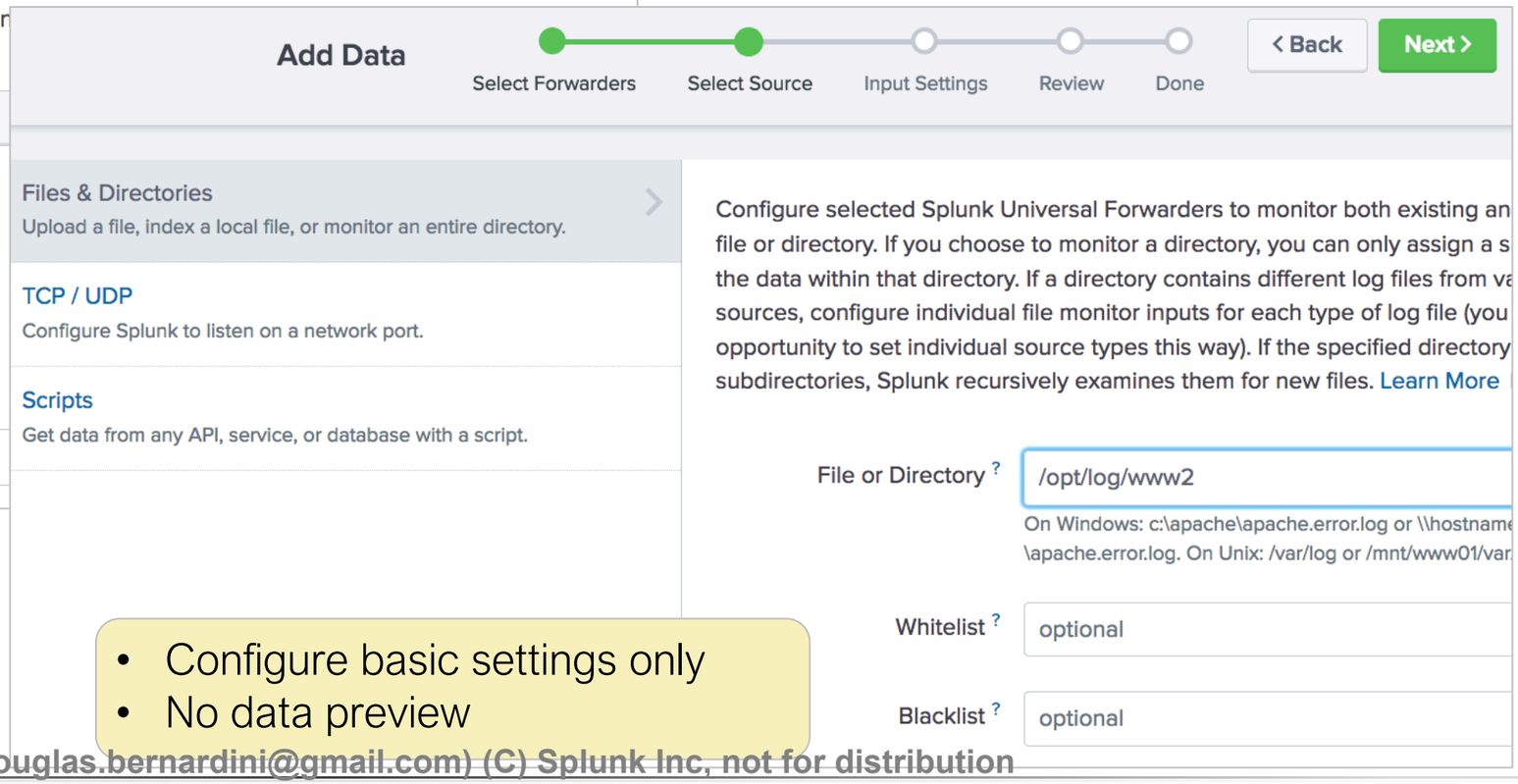
- Use the deployment server to create forwarded inputs
- Optionally create deployment apps for configuring inputs on deployment clients

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



Creating Forwarded Inputs (cont.)

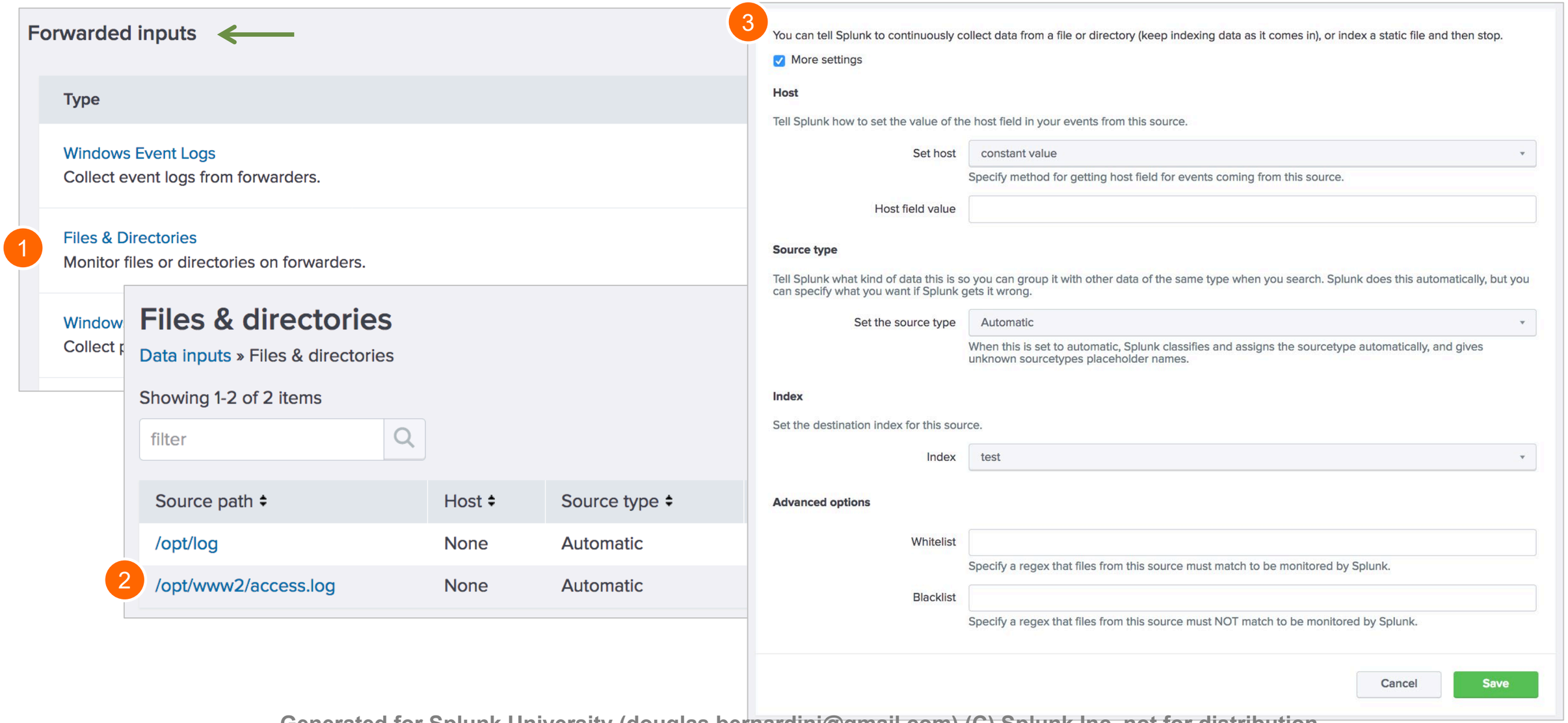




- Creates new server class or uses existing one
- Creates a new app for this input (or updates existing)

- Configure basic settings only
- No data preview

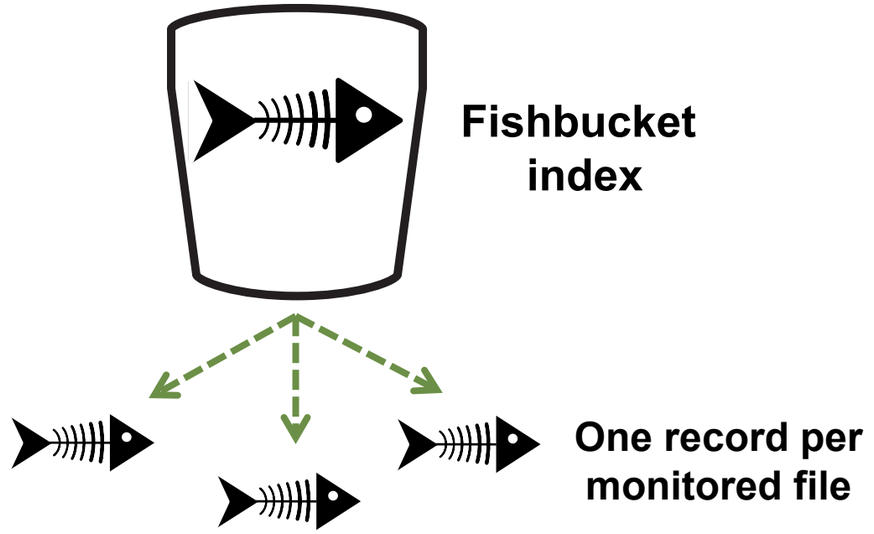
Editing Forwarded Inputs



What is the Fishbucket?

Fishbucket

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



Includes:

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

Editing Inputs and Re-indexing Data

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

Resetting Input File Monitors

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

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

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

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

or

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

3. Start Splunk

Warning

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

Module 5 Knowledge Check

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

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

Time: 20-25 minutes

Description: File Monitor Input

Tasks:

- Add a monitor input for a remote directory on UF2 to the **`test`** index
- Modify the **`inputs.conf`** file using the following caveats
  - Send the source logs to the **`sales`** index
  - Override the **`default-host`** name value
  - Monitor only the **`www.*`** sub-directories
  - Exclude the indexing of the **`secure.log`** files
- Re-deploy the **`inputs.conf`** file

Module 6:
Network Inputs

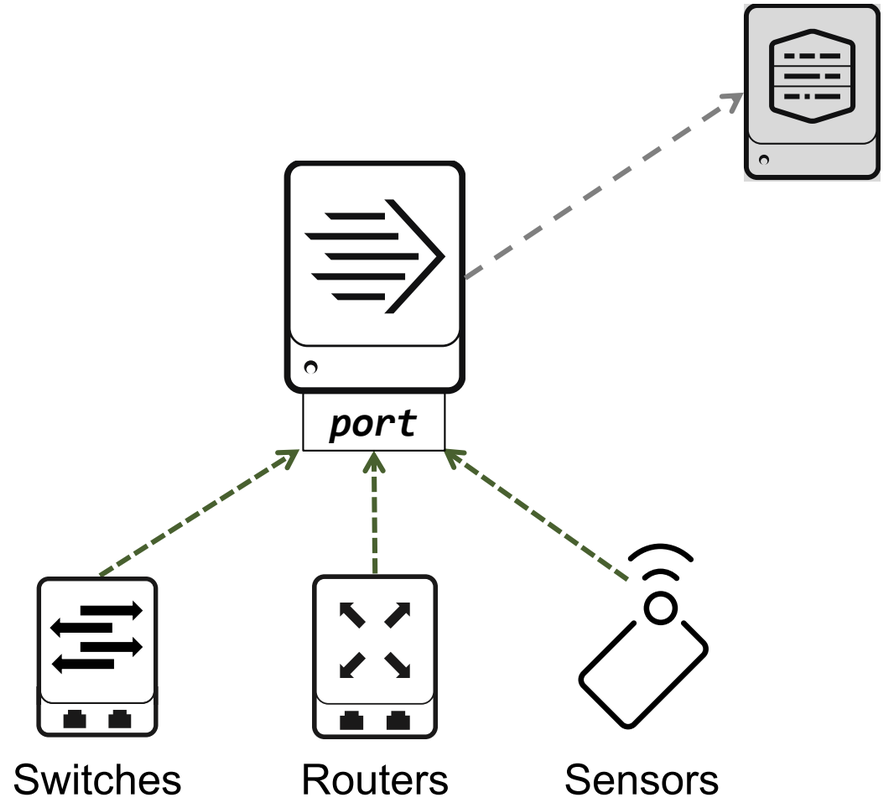
Module Objectives

- Create network (TCP and UDP) inputs
- Describe optional settings for network inputs

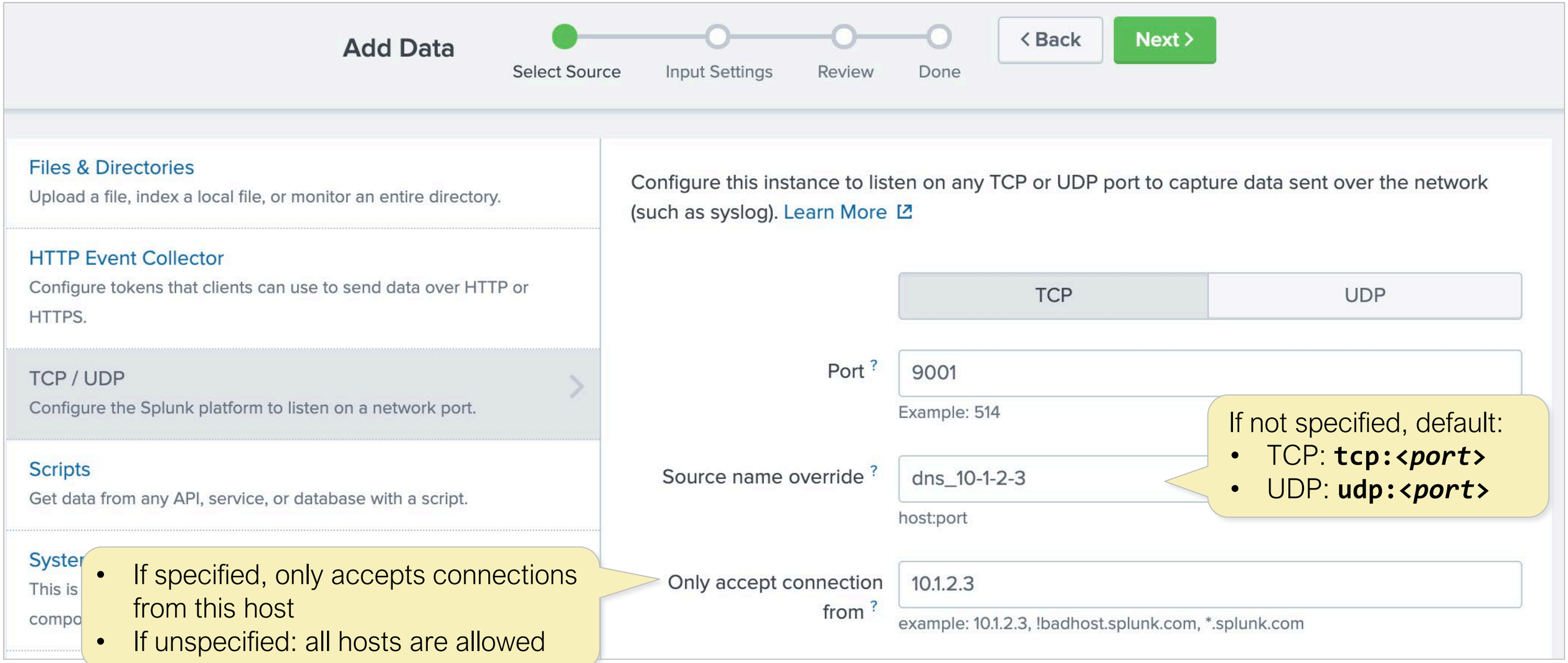
Network Inputs

Network Inputs

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



Adding Network Input



Optional Network Input Settings

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

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

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

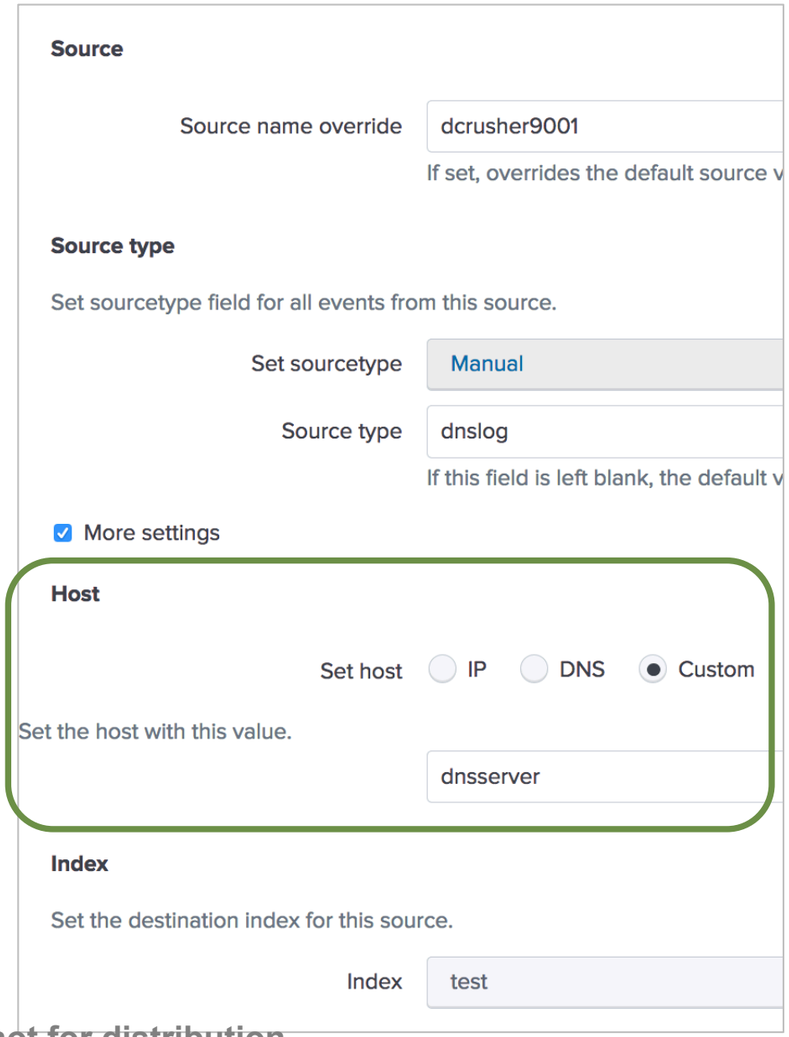
Examples:

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

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


Network Input: Host Field

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

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

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

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



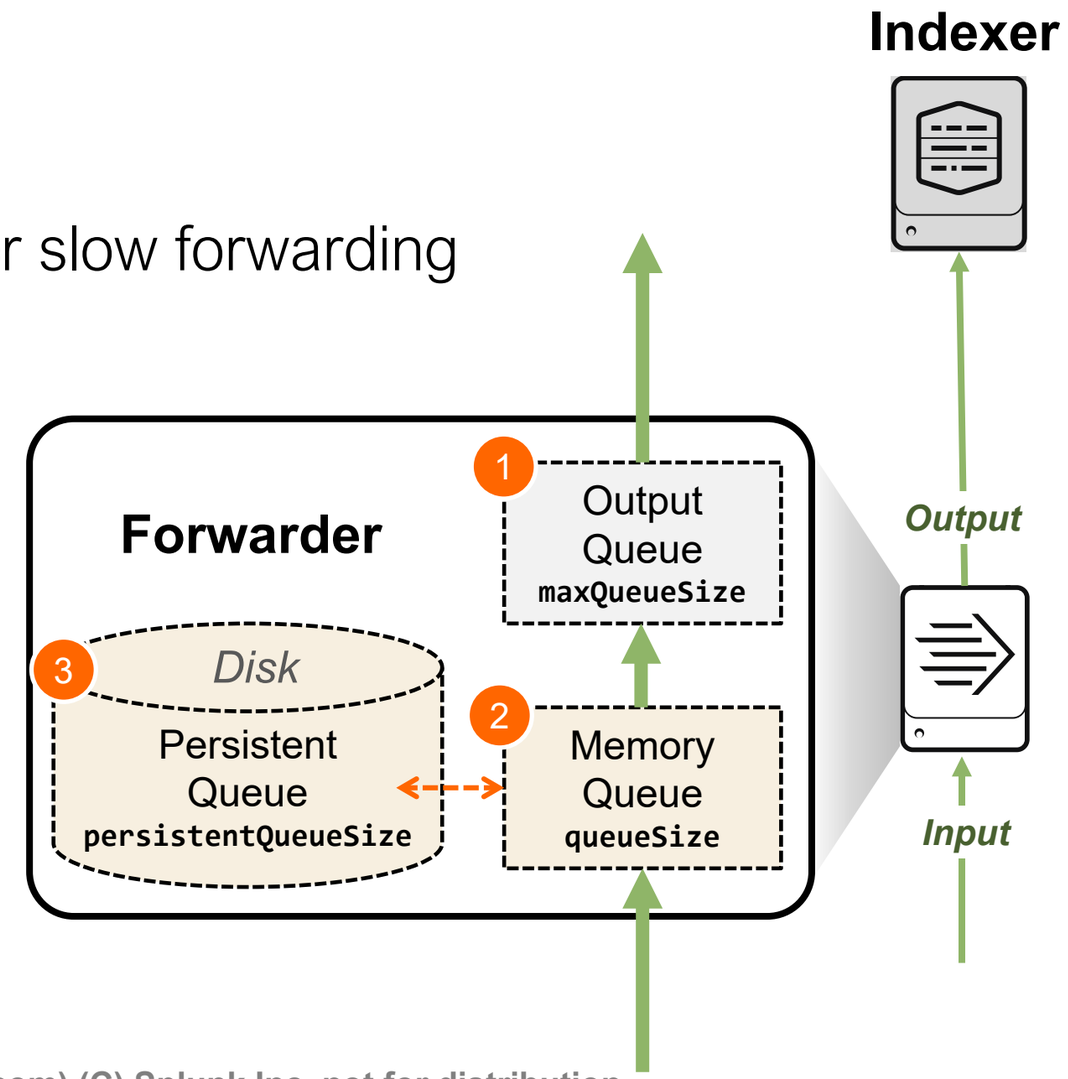
Network Input: Sender Filtering Options

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

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


Network Input: Queues

- Provide input flow control
- Apply to TCP, UDP, scripted input
- Control network data bursts, slow resources, or slow forwarding
  1. If indexers can't be reached:
     → Data is stored in the output queue
  2. If the output queue is full:
     → Data is stored in the memory queue
  3. If the memory queue is full:
     → Data is stored in the persistent queue
- Persistent queue preserves across restarts
  – Not a solution for input failure



Network Input: Setting Queue Attributes

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

`inputs.conf`

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


Special Handling and Best Practices

UDP

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

Syslog

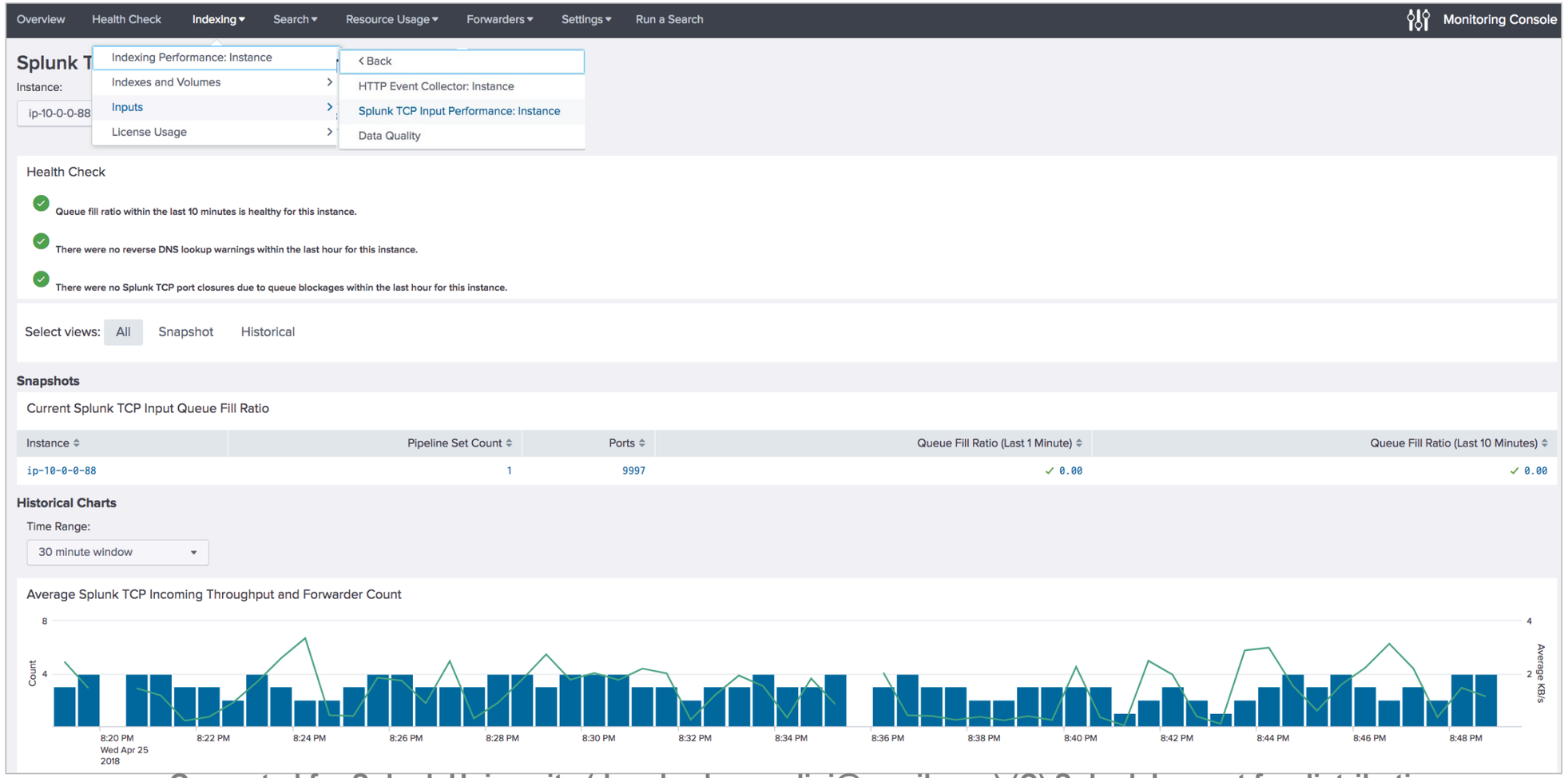
- Send data to a syslog collector that writes into a directory structure (for example: **`/var/log/syslog/`*`servername`*`/`*`filename`*`.txt`**)
- Monitor the directory and use **`host_segment`**
- docs.splunk.com/Documentation/Splunk/latest/Data/HowSplunkEnterprisehandlessyslogdata

SNMP traps

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

Monitoring with MC: Splunk TCP Inputs

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

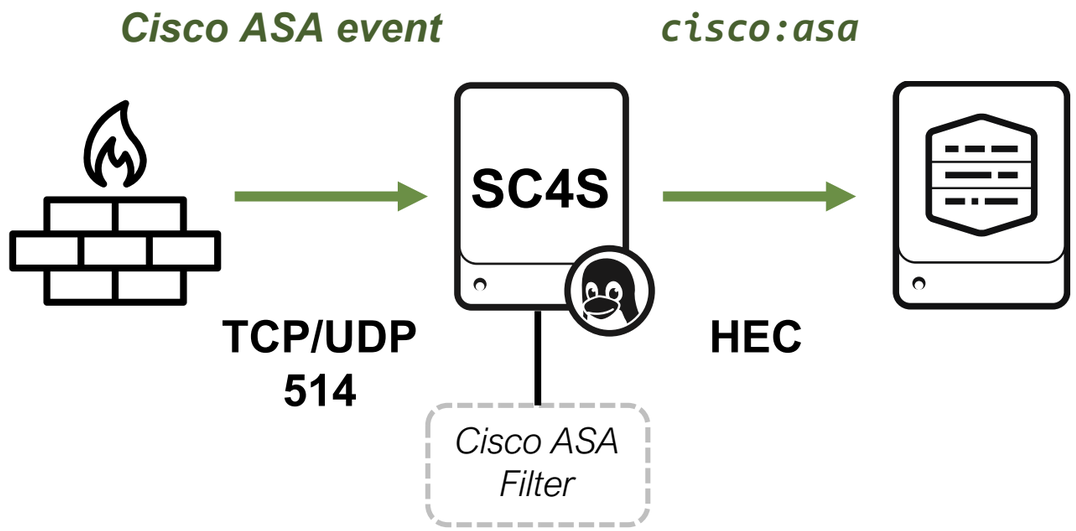


 | 15 September 2021

Understanding Splunk Connect for Syslog (SC4S)

Splunk Connect for Syslog

- Lower burden of getting syslog into Splunk
- Consistent, documented, repeatable
- Turnkey data ingestion for common source types
- Lower Splunk overhead for improved scaling and data distribution
- Containerized Syslog appliance



Identify / Parse / Format

Module 6 Knowledge Check

- 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?
- 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

- 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.

- 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

Time: 15 minutes

Description: Network Inputs

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

Note

Your instructor will run a script to send TCP data ports on the forwarder.

Use your assigned port to listen for the TCP data.

Module 7: Scripted Inputs

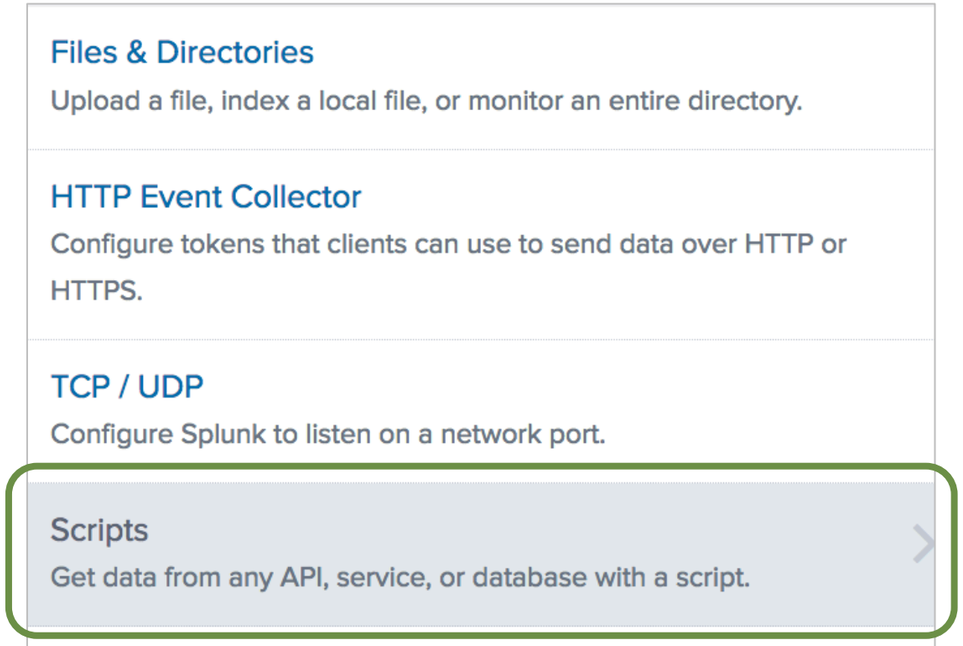
Module Objectives

Create a basic scripted input

Scripted Inputs

Scripted Inputs

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



Warning

Splunk only executes scripts from:

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

 | 15 September 2021

Defining a Scripted Input

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

     Example: **`splunk cmd SPLUNK_HOME/etc/apps/<app>/bin/myscript.sh`**
3. Deploy the script to production servers, for example if using a deployment server:
   - Copy script to **`SPLUNK_HOME/etc/deployment-apps/<app>/bin/`**
   - Deploy script using **Add Data** > **Forward** from Splunk Web
4. Verify the output of the script is being indexed

Scripted Input Stanza

inputs.conf

```
[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 a specified OS user; Splunk passes an authorization token via stdin to the script

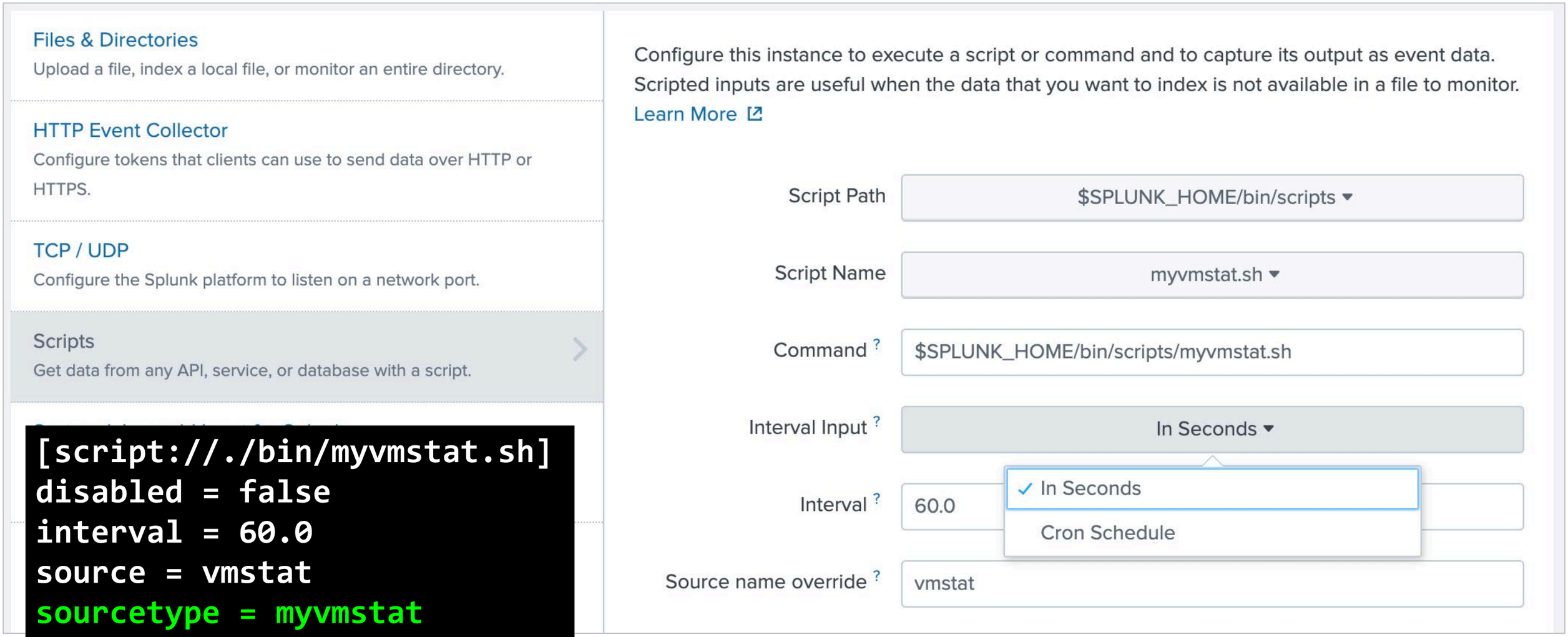
Interval is the time period between script executions (default: 60 seconds)

Warning

Splunk only executes scripts from:

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

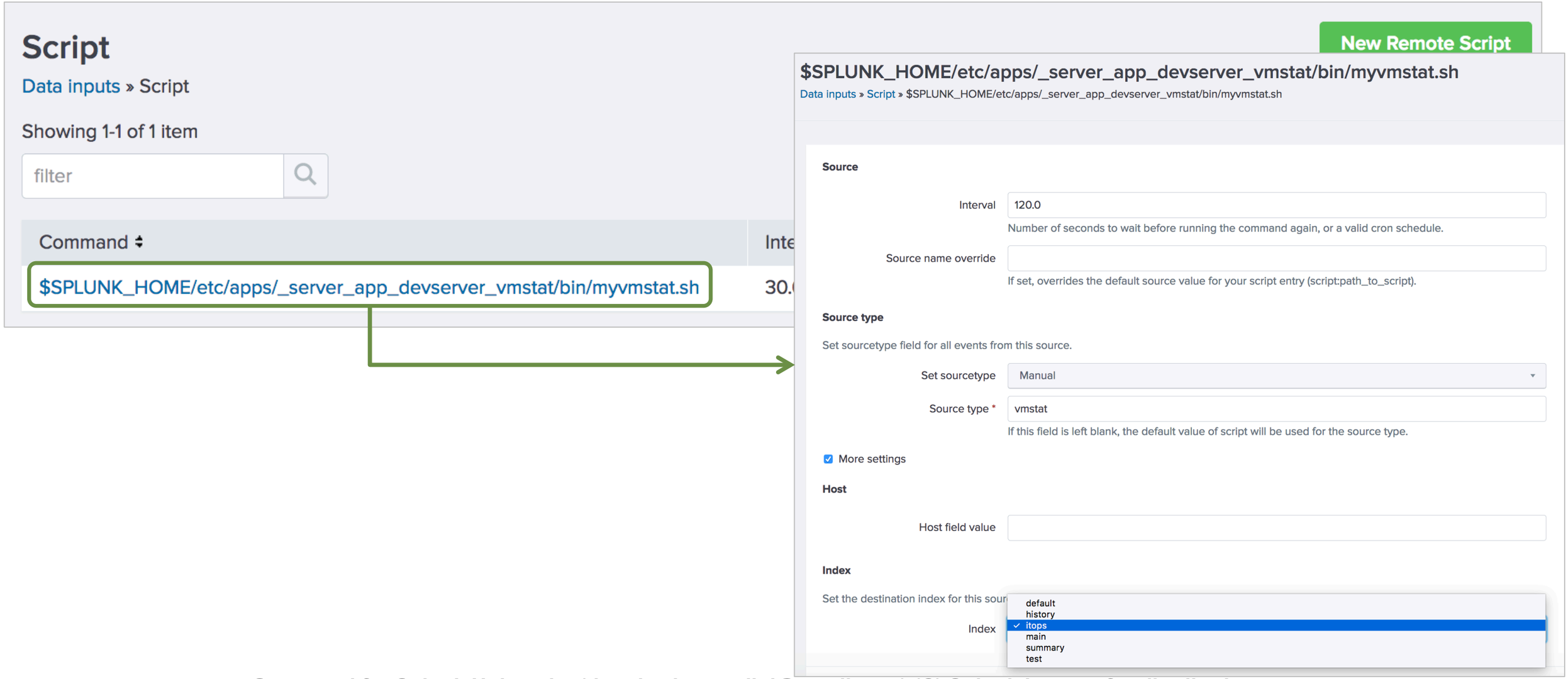
Scripted Inputs Example




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

inputs.conf

Editing Scripted Inputs



Scripted Input Buffering

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

Alternates to Using Scripted Input

Monitor a file containing the output of the script

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

Use Splunk's modular input

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

Module 7 Knowledge Check

- True or False. Persistent Queue and Memory Queue can be applied to Network as well as Scripted inputs.
- True or False. An interval setting for scripted inputs can be specified in **`cron`** syntax.

Module 7 Knowledge Check – Answers

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

  True.

- 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.

Module 7 Lab Exercise

Time: 10 minutes

Description: Scripted Inputs

Tasks:

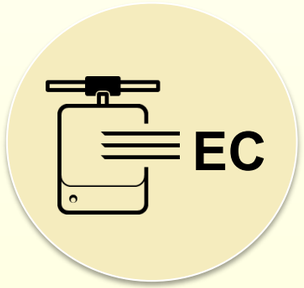
- Add a scripted input on your deployment server
- Deploy the scripted input to your forwarder

Module 8:
Agentless Inputs

Module Objectives

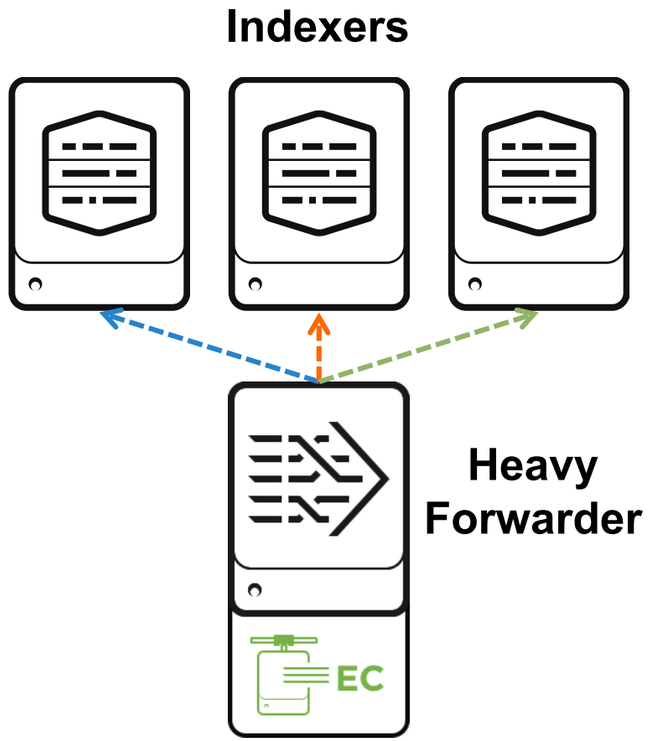
- Configure Splunk HTTP Event Collector (HEC) agentless input
- Describe Splunk App for Stream

HTTP Event Collector (HEC) Agentless Inputs



HTTP Event Collector (HEC)

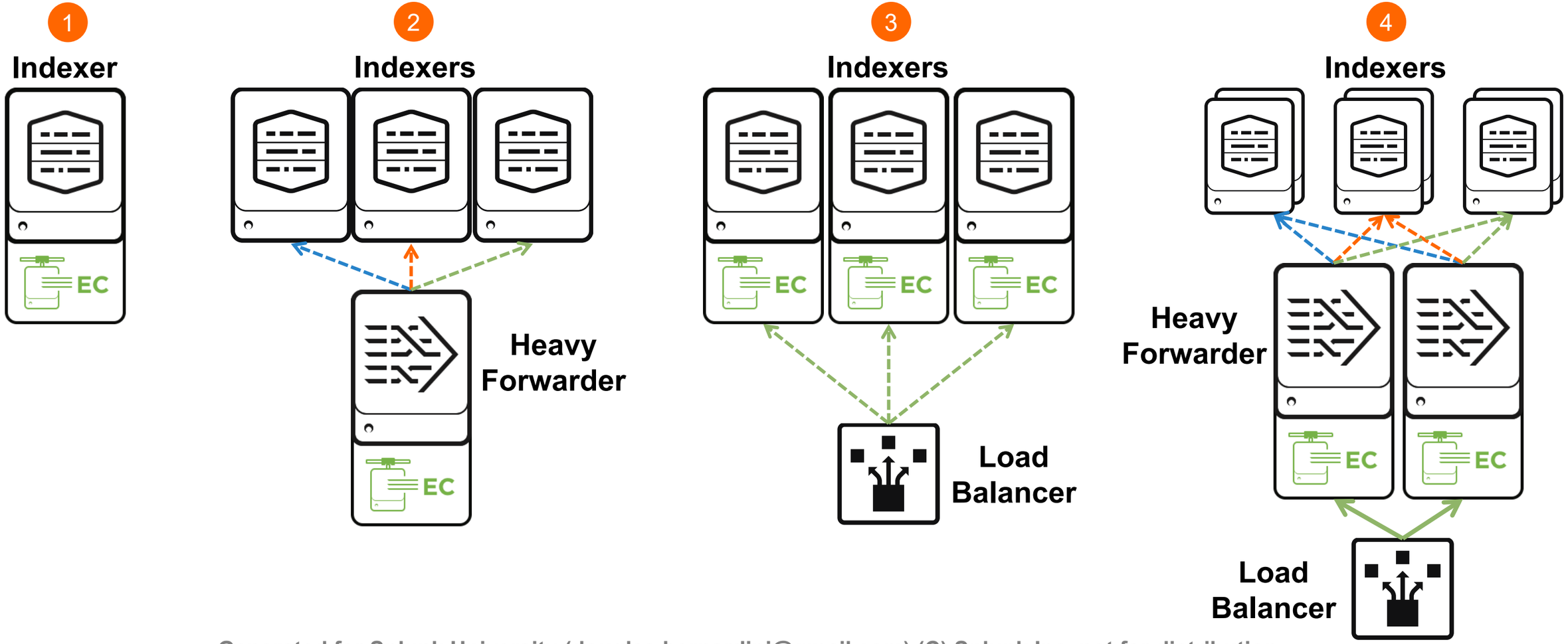
- A token-based HTTP input that is secure and scalable
- Sends events to Splunk without the use of forwarders (such as log data from a web browser, automation scripts, or mobile apps)
- Can facilitate logging from distributed, multi-modal, and/or legacy environments



Event collector enabled to receive HTTP events

Distributed HEC Deployment Options

HEC can scale by taking advantage of Splunk distributed deployment



 | 15 September 2021

Configuring HTTP Event Collector

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

Sending HTTP Events from a Device

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

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

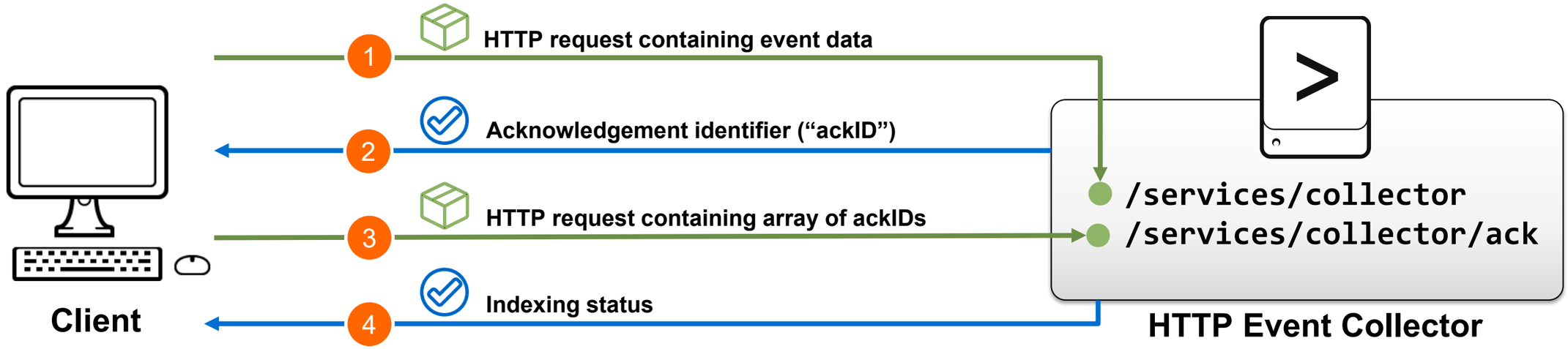
HTTP Event Collector Options

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

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

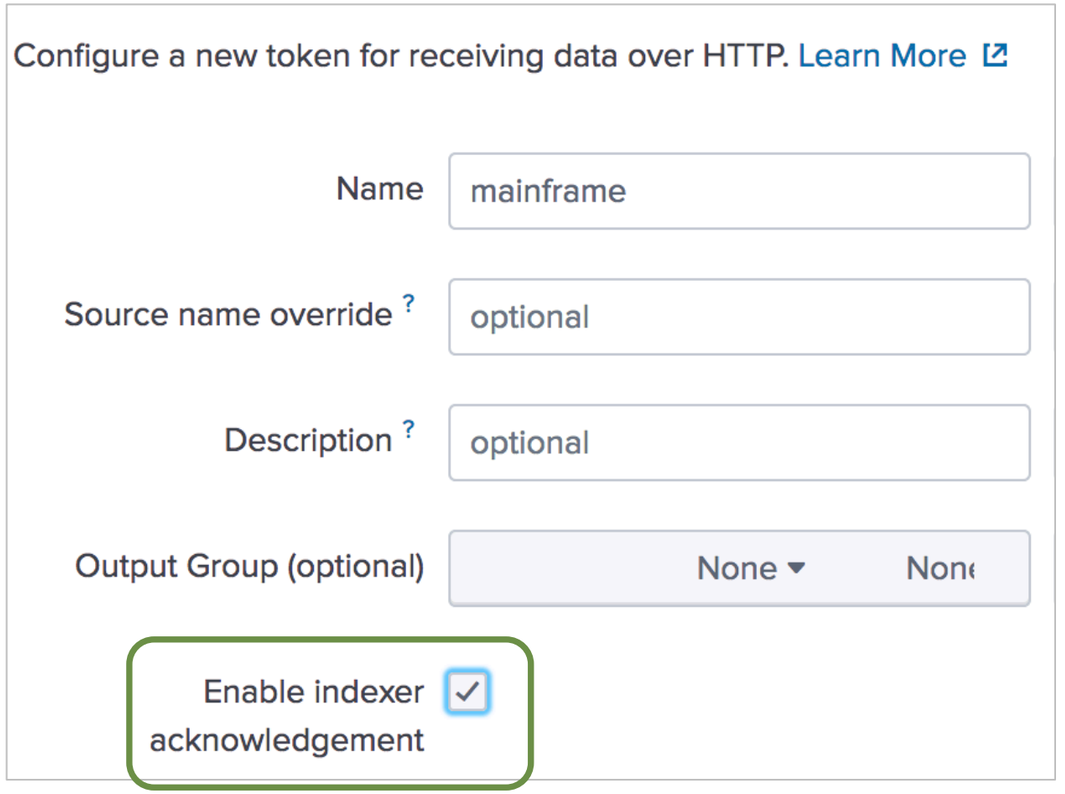
HEC Indexer Acknowledgement

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



HEC Indexer Acknowledgement Notes

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



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

Sending Raw Payloads to HEC

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

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

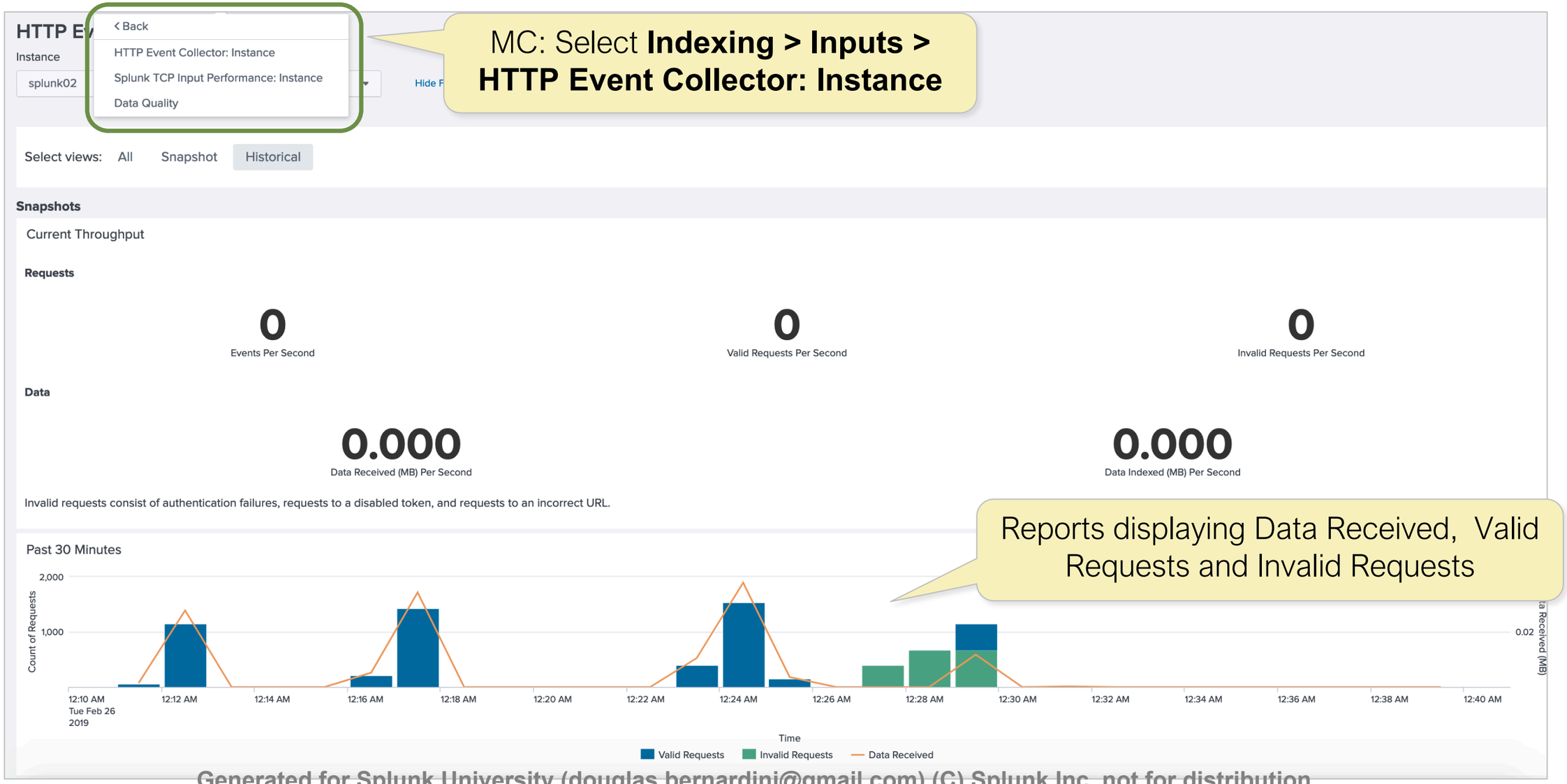
Configuring Dedicated HTTP Settings

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

inputs.conf

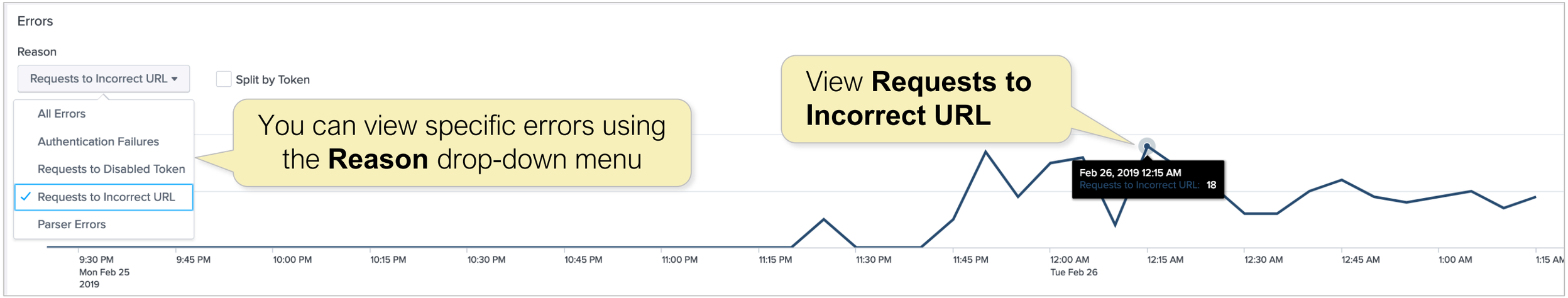
```
[http]
enableSSL = 1
crossOriginSharingPolicy = *.splunk.com
acceptFrom = "!45.42.151/24, !57.73.224/19, *"
```


Monitoring HEC with MC



 | 15 September 2021

Monitoring HEC with MC – Viewing Errors



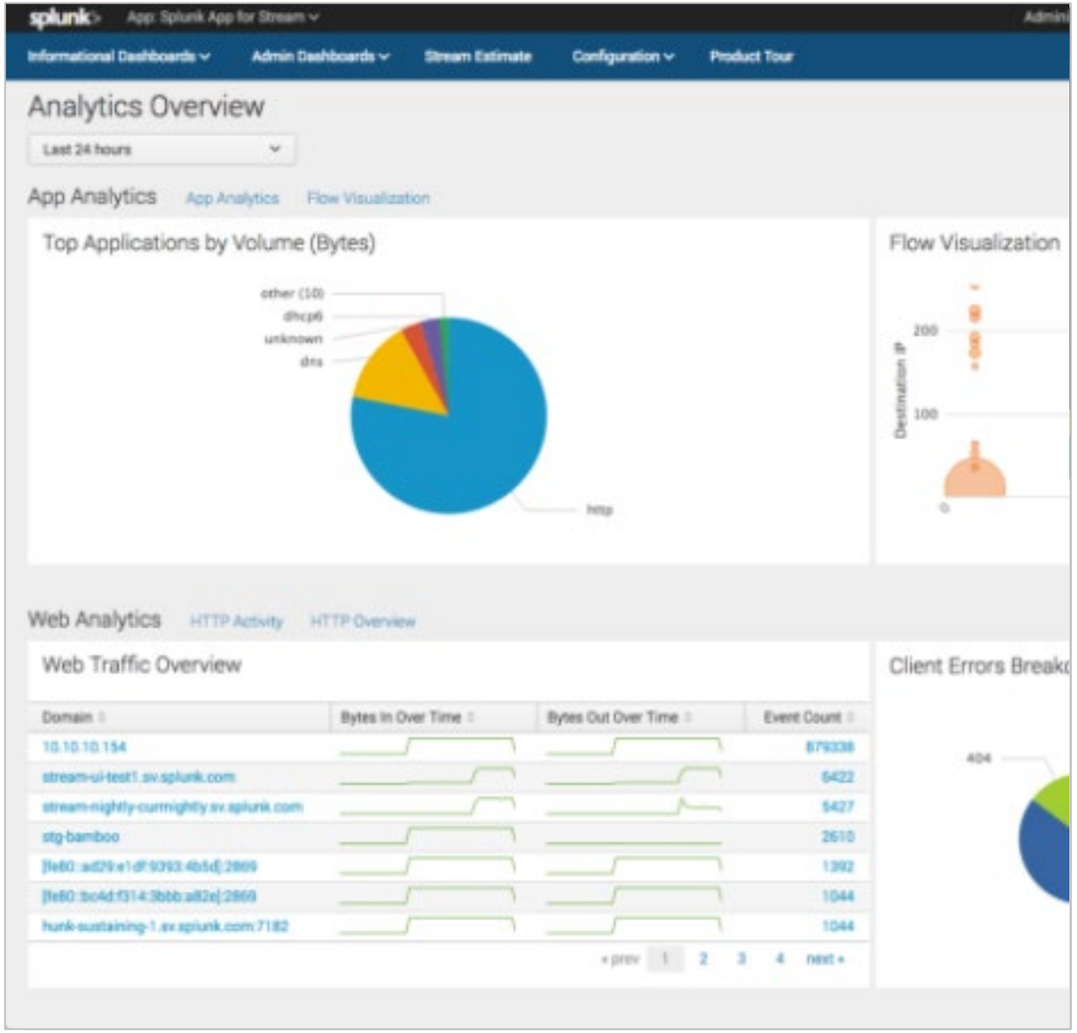
HTTP Event Collector (HEC) Documentation

- Refer to:
  - Introduction to Splunk HTTP Event Collector
    dev.splunk.com/view/event-collector/SP-CAAAE6M
  - Blogs: Tips & Tricks on HTTP Event Collector
    www.splunk.com/en_us/blog/tips-and-tricks/http-event-collector-your-direct-event-pipe-to-splunk-6-3.html

Understanding Splunk App for Stream

Splunk App for Stream

- Part of purpose-built wire data collection and analytics solution from Splunk
- An alternative way to collect “difficult” inputs
  - Database servers without forwarders
  - Network traffic not visible to web logs
- Able to read data off the wire
- Supports Windows, Mac, and Linux



Module 8 Knowledge Check

- ❑ 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 8 Knowledge Check – Answers

- 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 8 Lab Exercise

Time: 15 minutes

Description: HTTP Event Collector

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 9:
Operating System Inputs

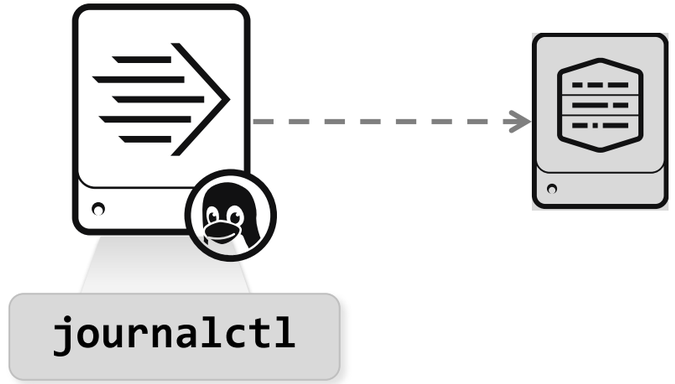
Module Objectives

- Identify Linux-specific inputs
- Identify Windows-specific inputs

Identifying JournalD Inputs For UF

JournalD Inputs on Linux

- Natively supports **`journalctl`** command for viewing logs collected by **`systemd`**
- Collects thousands of events per second with minimal impact
- Only requires **`inputs.conf`** configuration
- Supported in Splunk 8.1 and later



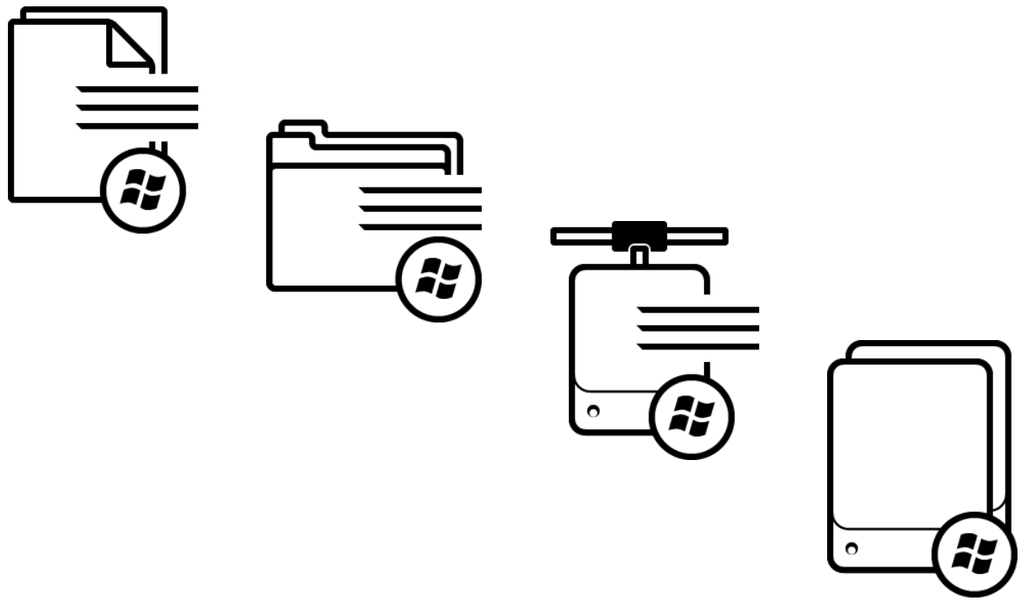
`inputs.conf`

```
[journald://my-stanza]
journalctl-include-list = PRIORITY, CMD, EXE
journalctl-exclude-list =
journalctl-filter = _SYSTEMD_UNIT=my.service
   _PID=232 + _SYSTEMD_UNIT=sshd
journalctl-grep = ^WARN.*disk,
   .*errno=\d+\S+restarting
journalctl-user-unit = unit1, unit2
```

Windows-Specific Inputs

Windows-Specific Inputs

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



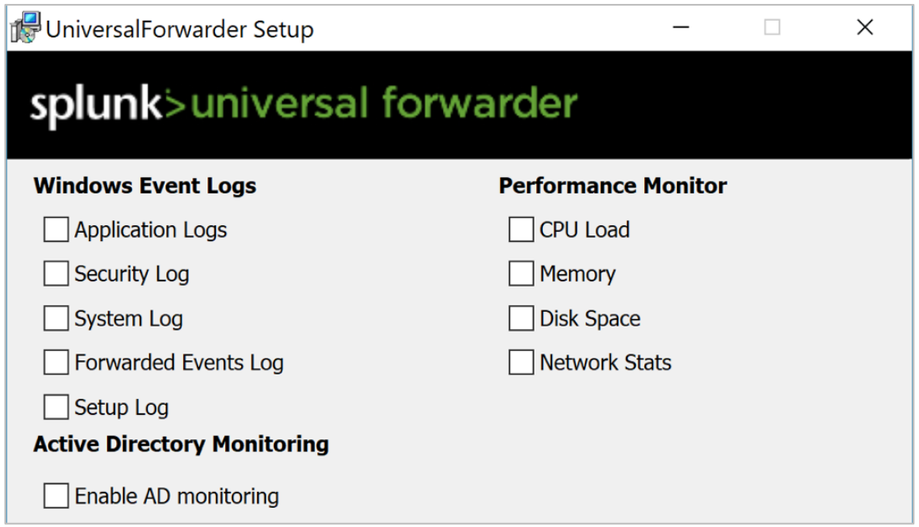
Windows-Specific Input Types

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

Options for Configuring Local Windows Inputs

- During the Windows forwarder install
  - Easy to use for testing and proof of concept (PoC)
  - Entries created in the app **`SplunkUniversalForwarder`**
  - Presents issues when centrally managing configuration with Deployment Server (DS)

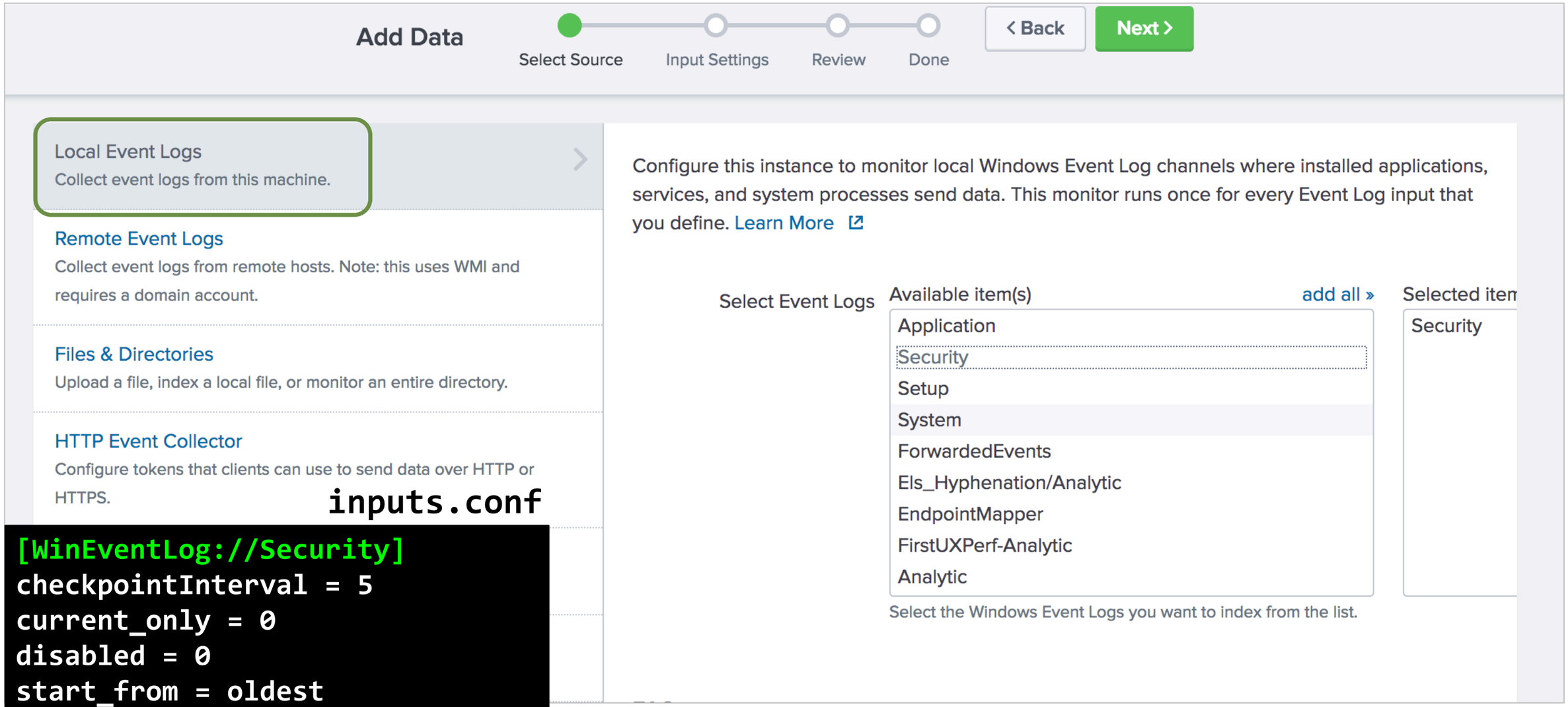


- Manually (Best Practice)
  - Create entries in custom app or use **Splunk Add-on for MS Windows**:

    splunkbase.splunk.com/app/742/
  - Easy to manage using a DS
  - For details refer to:
    - **`inputs.conf.spec`**
    - **`inputs.conf.example`**

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

Configuring Local Windows Inputs Using Add Data



inputs.conf

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

Windows Input Filtering Options

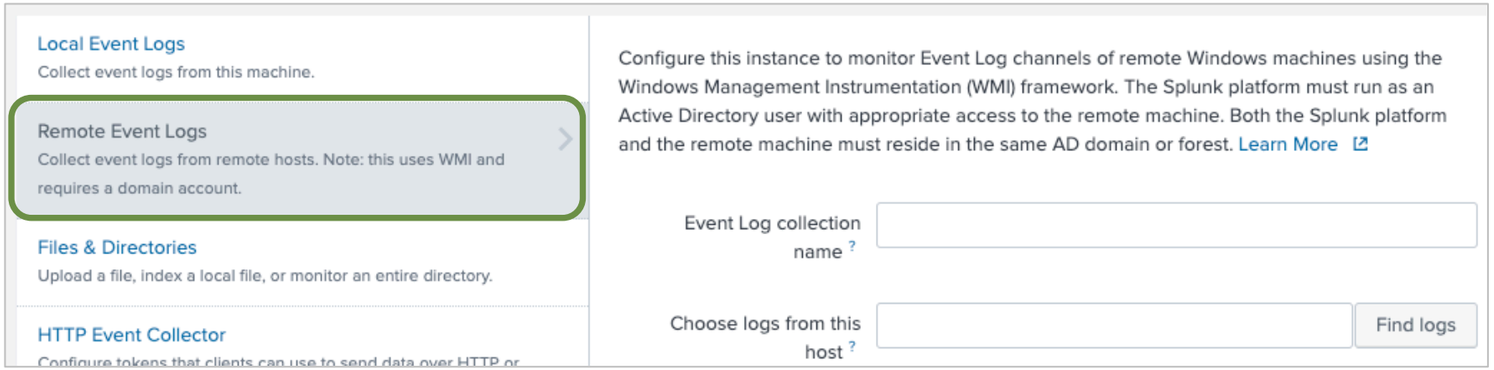
- Filter out non-essential events
  - Use include lists (**whitelist**) and exclude lists (**blacklist**)
  - Configure up to 10 entries for each list per stanza
  - Set entries based on event field names and regex:
    - **whitelist[1-9] = <*list*> | *key=regex* [*key=regex*]**
    - **blacklist[1-9] = <*list*> | *key=regex* [*key=regex*]**
  - In case of a conflict, the exclude lists (**blacklist**) prevails

inputs.conf

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

Windows Remote Inputs With WMI

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



Configuring WMI Inputs

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

wmi.conf

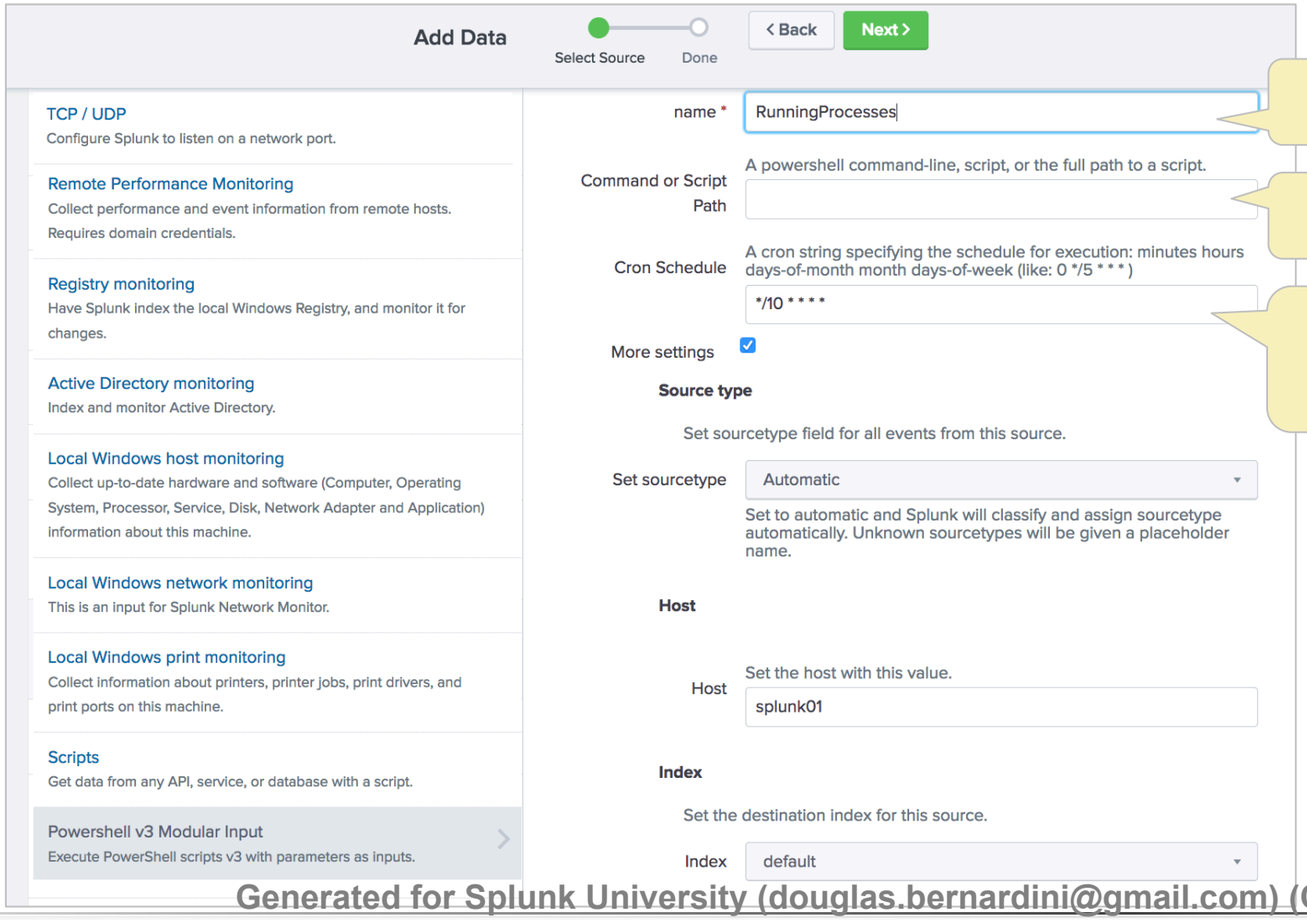
```
[WMI:remote-logs]
interval = 5
server = server1, server2, server3
event_log_file = Application, Security, System

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


Special Field Extractions

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

Powershell Input

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



PowerShell v3 or higher

Command or a script file

Blank field executes once only

`inputs.conf`

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

Windows Inputs Resources

- Monitoring Windows data with Splunk Enterprise

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

- Microsoft: Diagnostics - Windows Event Log

  docs.microsoft.com/en-us/windows/desktop/wes/windows-event-log

- Microsoft: Diagnostics - Performance Counters

  docs.microsoft.com/en-us/windows/desktop/PerfCtrs/performance-counters-portal

- Microsoft: Diagnostics - Performance Counters Reference

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

Module 9 Knowledge Check

- True or False. JournalD input only requires Splunk Enterprise 8.1 and **`inputs.conf`** settings.
- True or False. Windows input from a Windows UF must be forwarded to an Indexer running Windows.
- True or False. You can collect Active Directory data from a Windows Server remotely using **`wmi.conf`**.

Module 9 Knowledge Check – Answers

- True or False. JournalD input only requires Splunk Enterprise 8.1 and **`inputs.conf`** settings.

  True.

- True or False. Windows input from a Windows UF must be forwarded to an Indexer running Windows.

  False. Any platform indexer can be used.

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

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

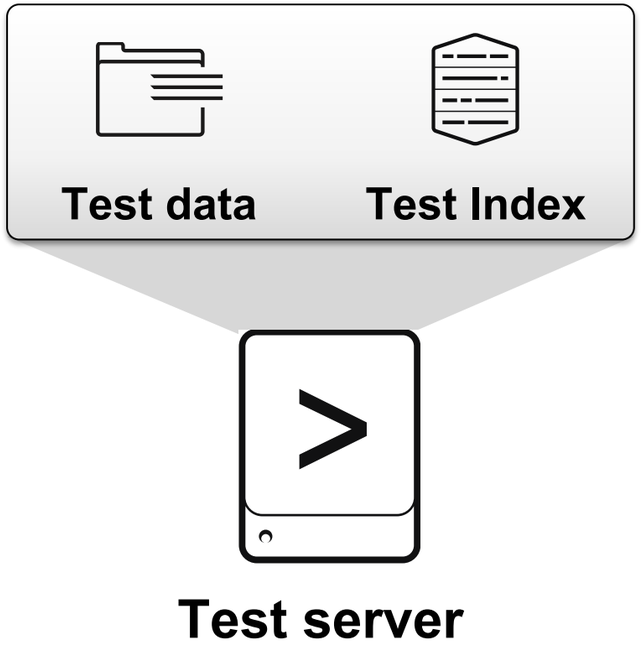
Module 10:
Fine-tuning Inputs

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

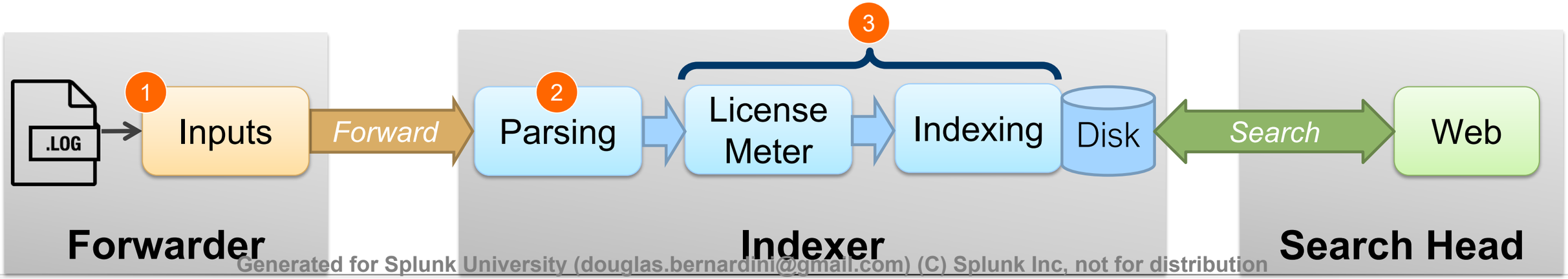
Review: Initial Data Input Testing

- Use a Splunk test server
  - Should be running same version as production
- Use test indexes
- Procedure:
  1. Copy production data to test server
  2. Use Splunk Web > **Add Data**
  3. Check to see if **sourcetype** and other settings are applied correctly
  4. Delete the test data, reset fishbucket if needed, change test configuration, and repeat as necessary



Index-Time Process

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



Things to Get Right at Index Time

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

* Optional

What if I Don't Get It Right?

On a testing / development server

- This is what a test/dev server is for!
- Clean or delete+recreate test index, change configuration, try again
- May need to reset the fishbucket

On a production server

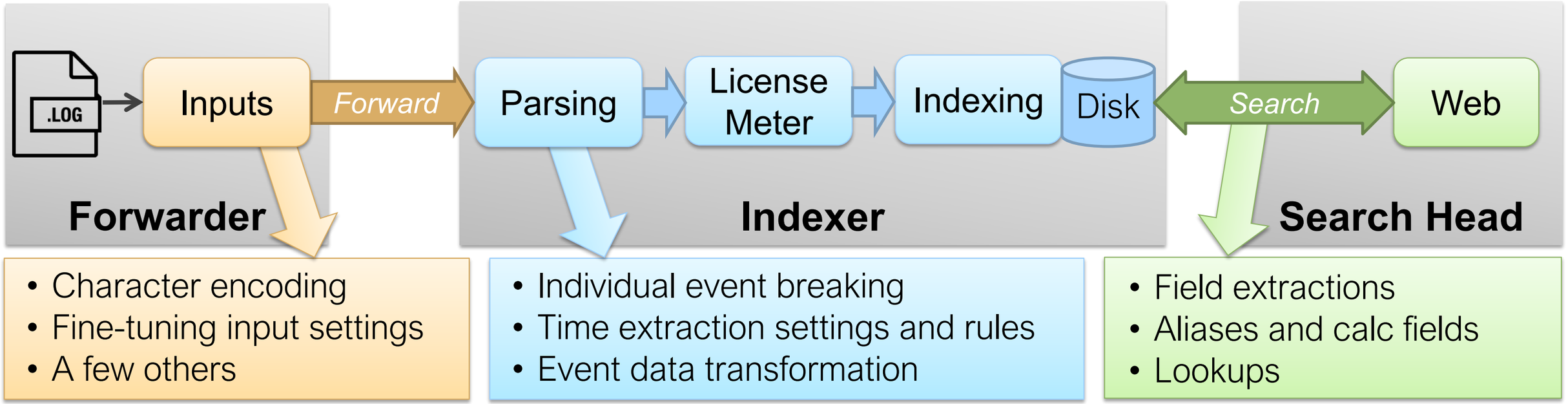
- Leave erroneous data in the system until it naturally “ages out” (reaches the index size or retention time limits)
- Attempt to delete the erroneous data
- Only re-index when it is absolutely necessary

The `props.conf` File

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

Phases and `props.conf`

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



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

  wiki.splunk.com/Where_do_I_configure_my_Splunk_settings

Stanzas in props.conf

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

syntax

```
[source::source_name]
attribute = value
```

```
[host::host_name]
attribute = value
```

```
[sourcetype_name]
attribute = value
```

example

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

```
[host::nyc*]
TZ = US/Eastern
```

```
[sales_entries]
CHARSET = UTF-8
```

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

Character Encoding

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

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

[sendmail]
CHARSET=AUTO
```

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

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

Fine-tuning Directory Monitor Source Types

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

inputs.conf

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

props.conf

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

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

Note

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

 | 15 September 2021

Module 10 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 10 Knowledge Check – Answers

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

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

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

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

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

[sendm*]
CHARSET=AUTO
```

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

Module 10 Lab Exercise

Time: 10-15 minutes

Description: Fine-tuning Inputs

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.

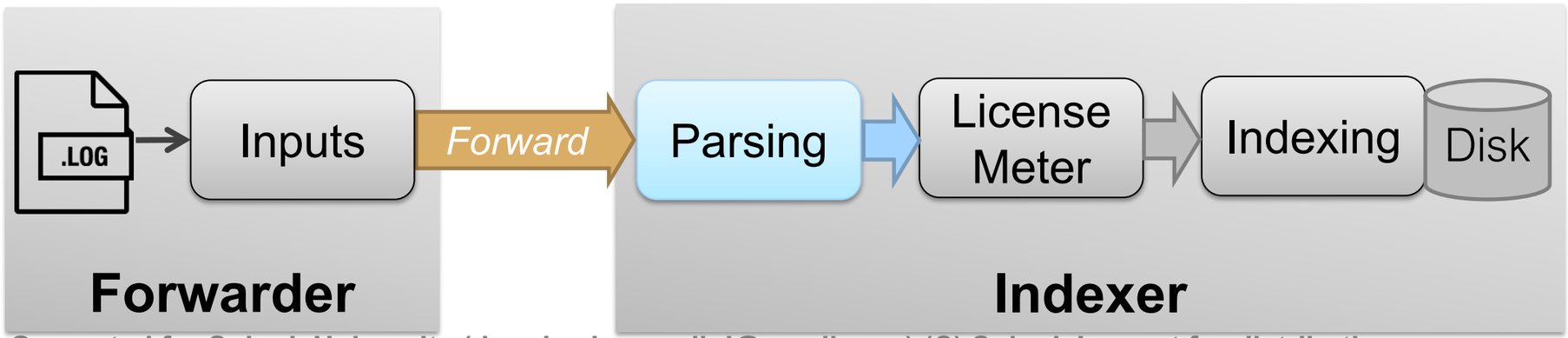
Module 11:
Parsing Phase and Data Preview

Module Objectives

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

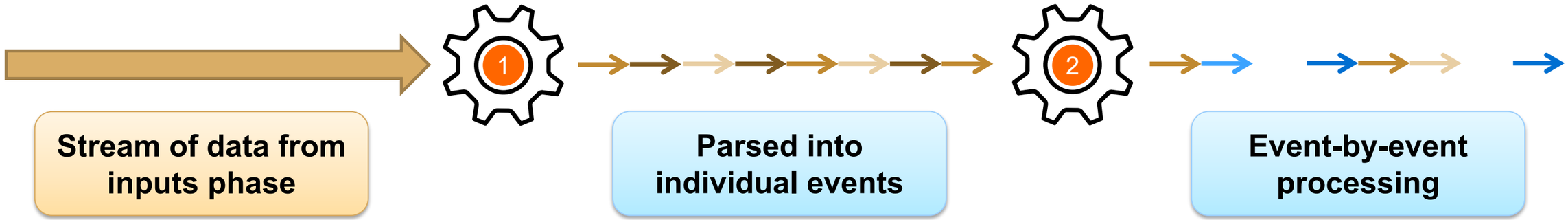
The Parsing Phase

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



Event Creation

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



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

Determining Event Boundaries

Step 1: Line breaking

- Splits the incoming stream of bytes into separate lines
- Configured with **`LINE_BREAKER = <regular_expression>`**
- Default is any sequence of new lines and carriage returns: **`([\r\n]+)`**

Step 2: Line merging (optional)

- Merges separate lines to make individual events
- Configured with **`SHOULD_LINEMERGE = true`** (default)
- Uses additional settings to determine how to merge lines (such as **`BREAK_ONLY_BEFORE`**, **`BREAK_ONLY_BEFORE_DATE`**, and **`MUST_BREAK_AFTER`**)
- If each event is a separate line, disable (set to **`false`**) to improve performance

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

Event Boundary Examples

Monitored input: Single line input with 3 events

```
[19/Sep/2020:18:22:32] VendorID=7033 Code=E AcctID=4390644811207834 ↲
[19/Sep/2020:18:22:48] VendorID=1239 Code=K AcctID=5822351159954740 ↲
[19/Sep/2020:18:22:59] VendorID=1243 Code=F AcctID=8768831614147676 ↲
```

props.conf

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

Monitored input: Multi-line input with 3 events

```
Sep 12 06:11:58 host1.example.com storeagent[49597] <Critical>: Starting update scan ↲
Sep 12 06:11:58 host1.example.com storeagent[49597] <Critical>: UpdateController: Message tracing { ↲
       "power_source" = ac; ↲
       "start_date" = "2018-08-21 20:10:39 +0000"; ↲
} ↲
Sep 12 06:11:58 host1.example.com storeagent[49597] <Critical>: Asserted BackgroundTask power ↲
```

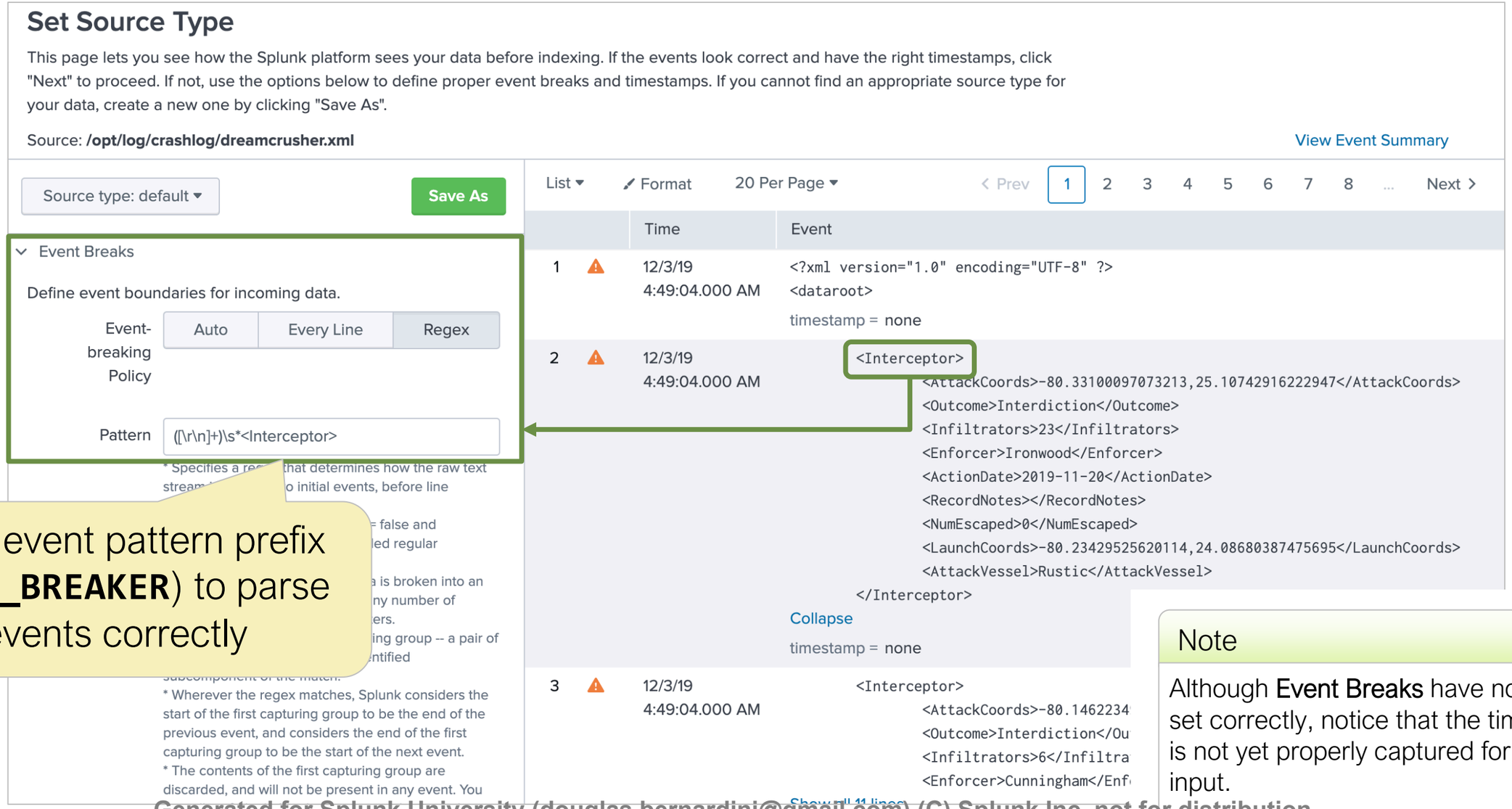
props.conf

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


Using Splunk Data Preview

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

Setting Event Breaks in Data Preview



Enter event pattern prefix (**LINE_BREAKER**) to parse events correctly

Note

Although **Event Breaks** have now been set correctly, notice that the timestamp is not yet properly captured for this input.

Date/timestamp Extraction

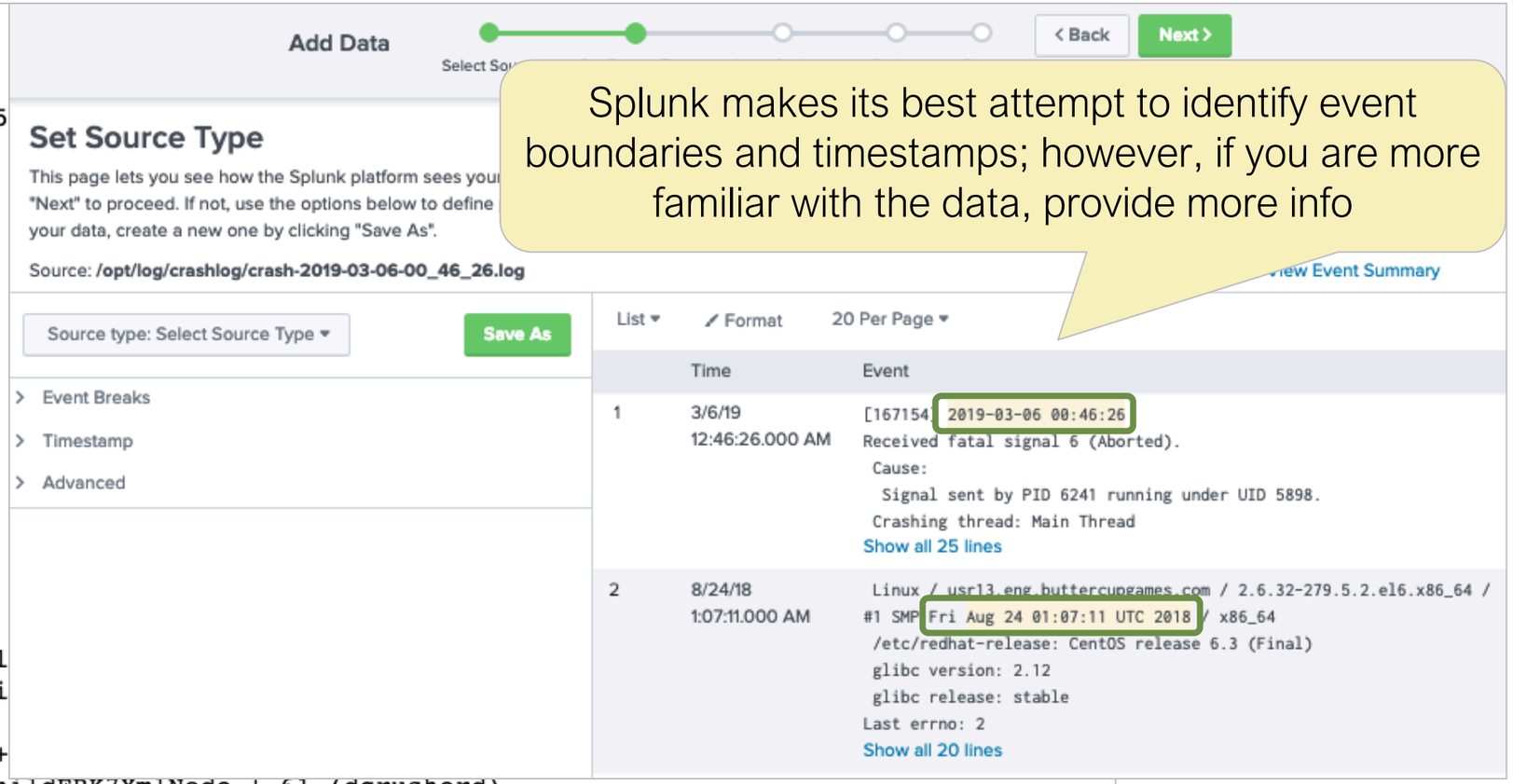
- Correct date/timestamp extraction is essential
  - Splunk works well with standard date/time formats and well-known data types
- Always verify timestamps when setting up new data types
  - Pay close attention to timestamps during testing/staging of new data
  - Check UNIX time or other non-human readable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

Incorrectly Determined Timestamps

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

 OS: Linux
 Arch: x86-64

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

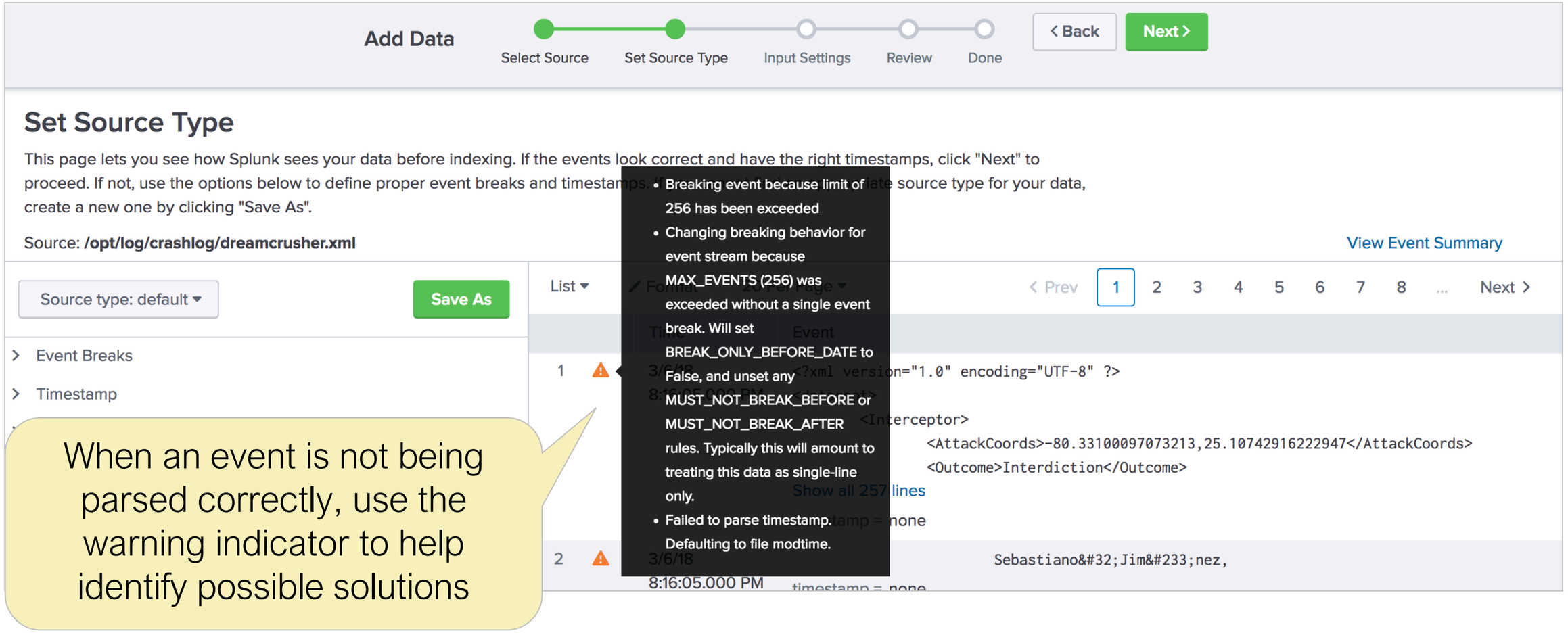
1

2



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

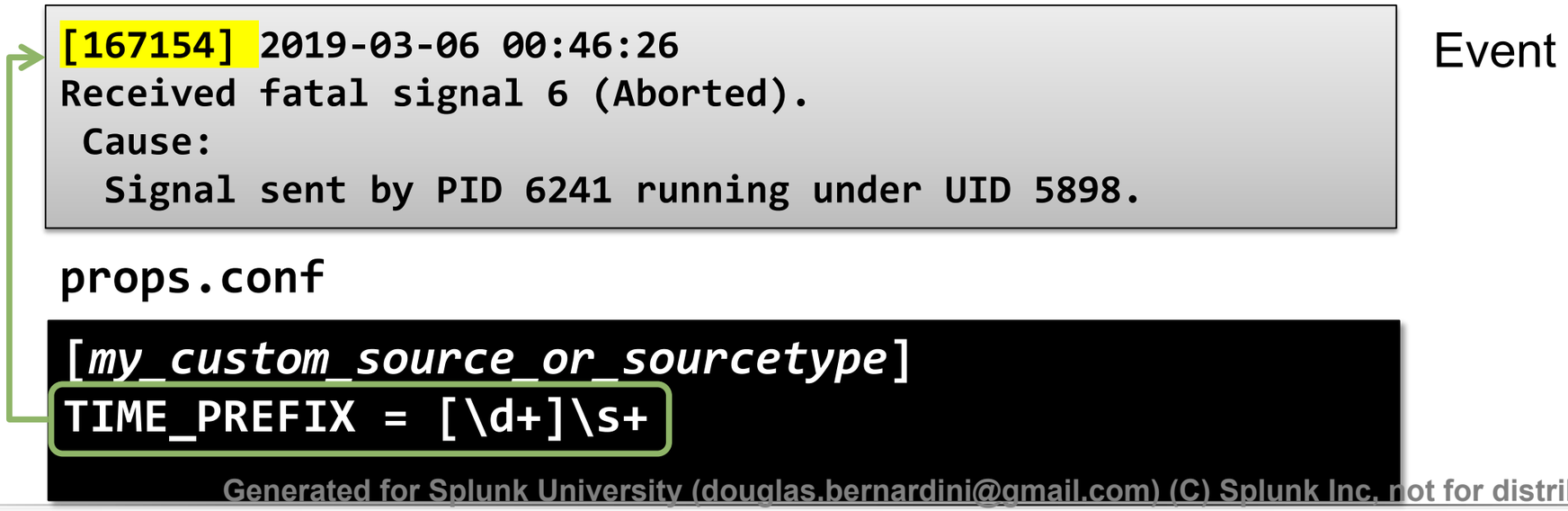
Failed To Parse Timestamps



When an event is not being parsed correctly, use the warning indicator to help identify possible solutions

Using **TIME_PREFIX**

- Syntax: **TIME_PREFIX = <*REGEX*>**
- Matches characters right BEFORE the date/timestamp
  - Use this syntax to specify where the timestamp is located in the event



Using **MAX_TIMESTAMP_LOOKAHEAD**

- Syntax: **MAX_TIMESTAMP_LOOKAHEAD = *<integer>***
- Specifies how many characters to look for a timestamp
  - Generally, starts from beginning of the event
  - If **TIME_PREFIX** is set, starts from the point the **TIME_PREFIX** indicates
  - Improves efficiency of timestamp extraction

```
[167154] 2019-03-06 00:46:26
Received fatal signal 6 (Aborted).
 Cause:
  Signal sent by PID 6241 running under UID 5898.
```

Event

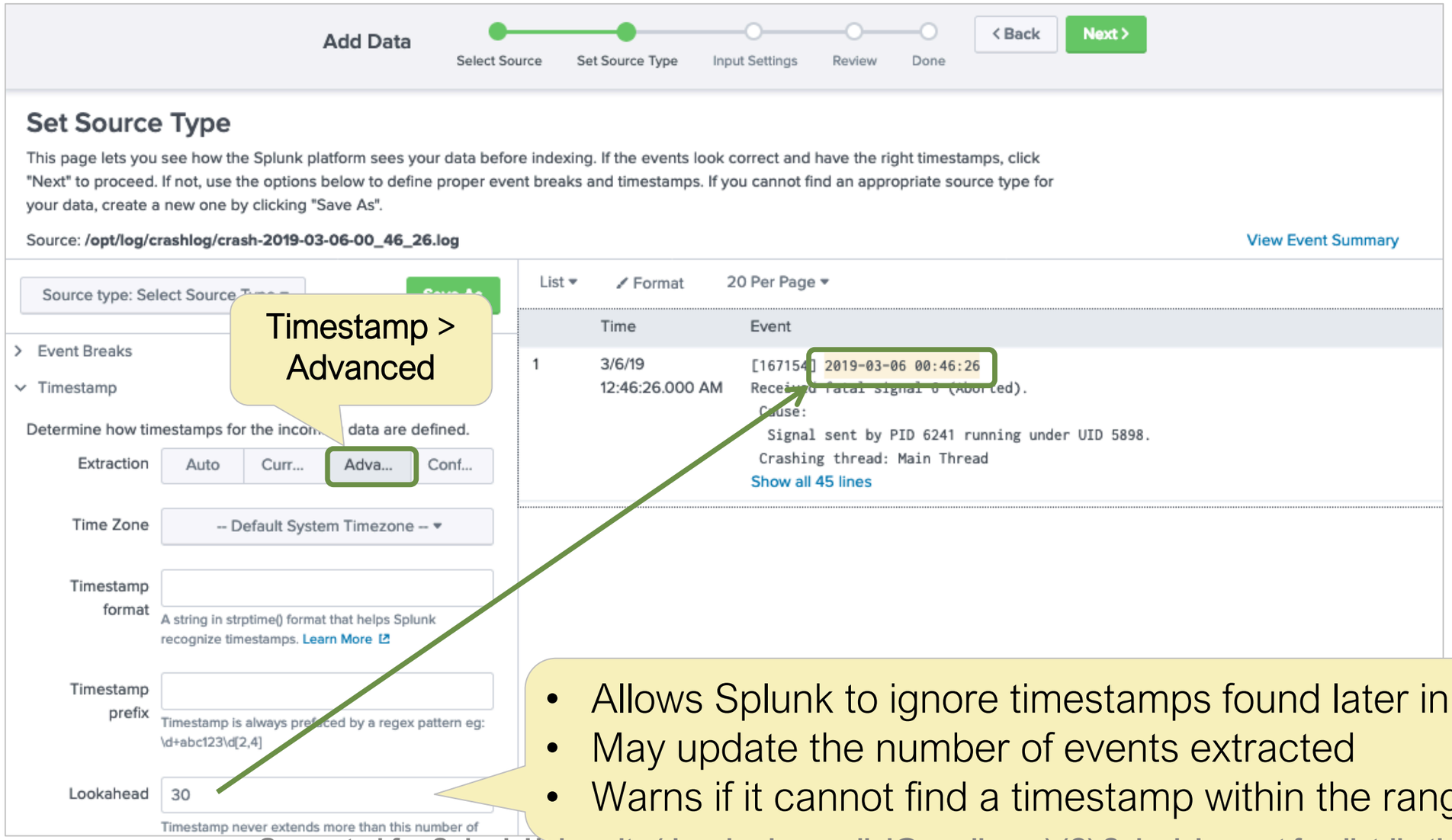
props.conf

```
[my_custom_source_or_sourcetype]
TIME_PREFIX = [\d+]\s+
MAX_TIMESTAMP_LOOKAHEAD = 30
```

Note

The complete timestamp string must be present within the specified range.

Using Timestamp Lookahead In Splunk Web



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

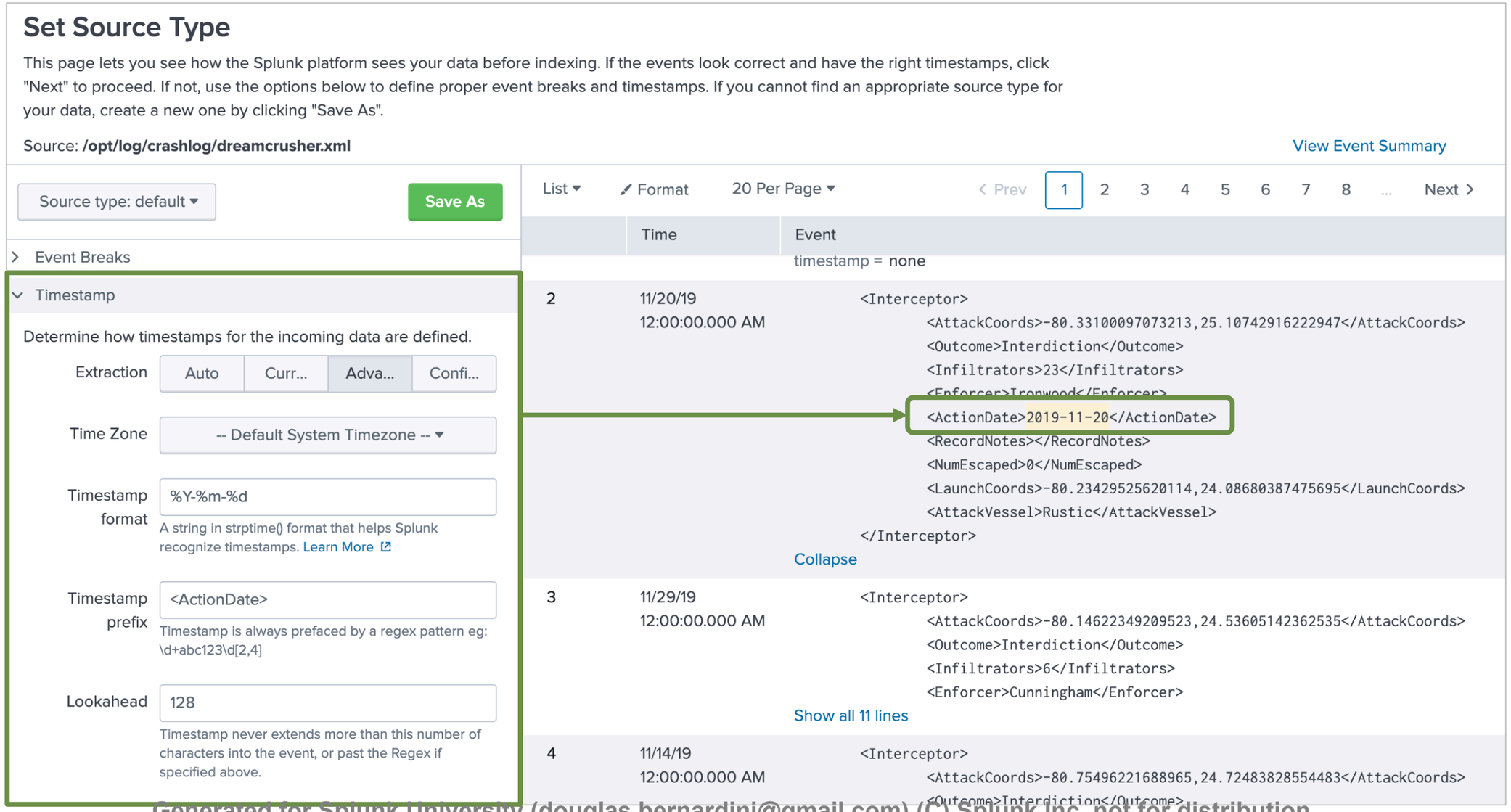
Using TIME_FORMAT

- Syntax: **TIME_FORMAT = <*strptime-style format*>**
- Examples:

Timestamp	TIME_FORMAT entry
2020-10-31	%Y-%m-%d
January 24, 2003	%B %d, %Y

- 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

Splunk Web: Advanced Timestamp Extraction



Setting Time Zone 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
     - en.wikipedia.org/wiki/List_of_zoneinfo_timezones
  3. If a forwarder is used, the forwarder-provided time zone is used
  4. If all else fails, Splunk applies the time zone of the indexer's host server

props.conf

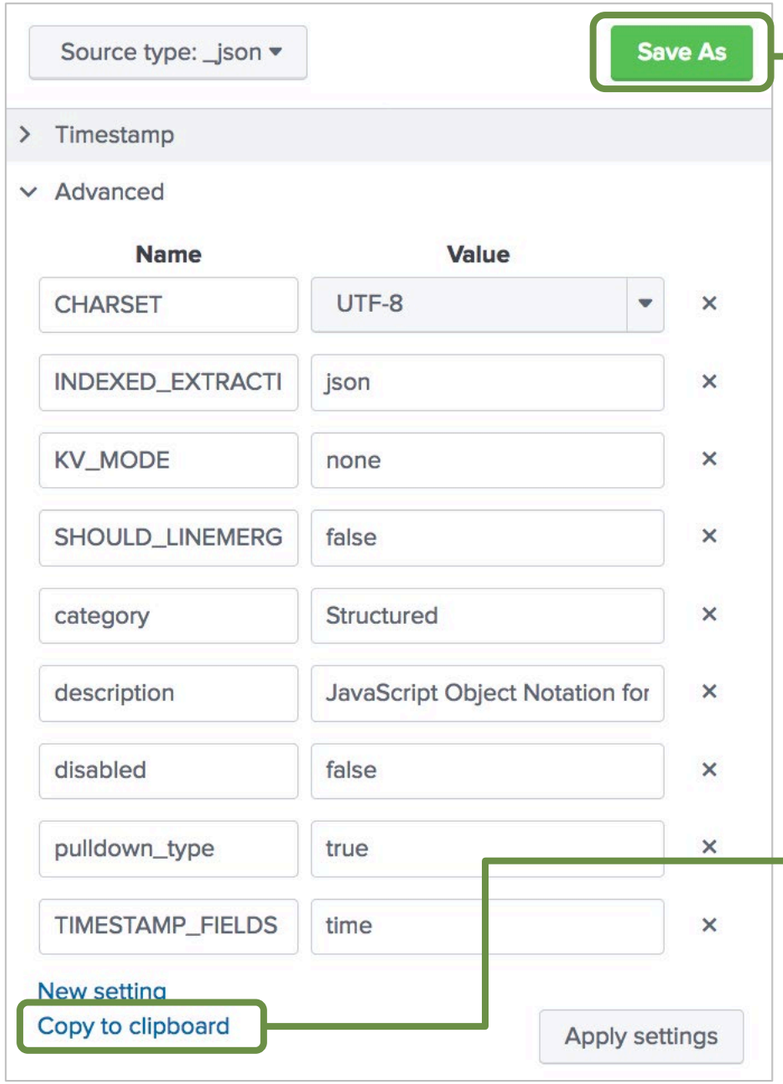
```
[host::nyc*]
TZ = America/New York

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

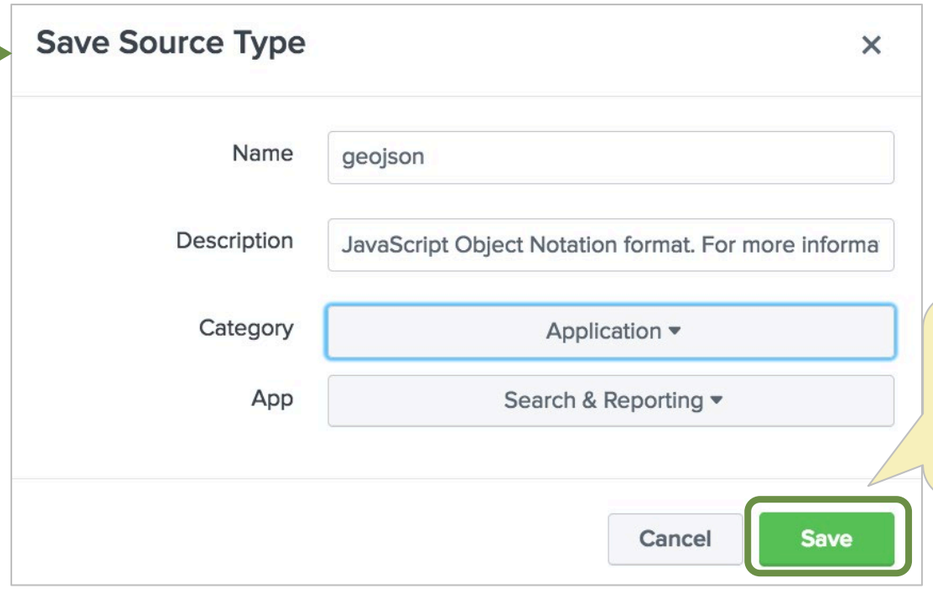

Splunk Event Timestamp Processing

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

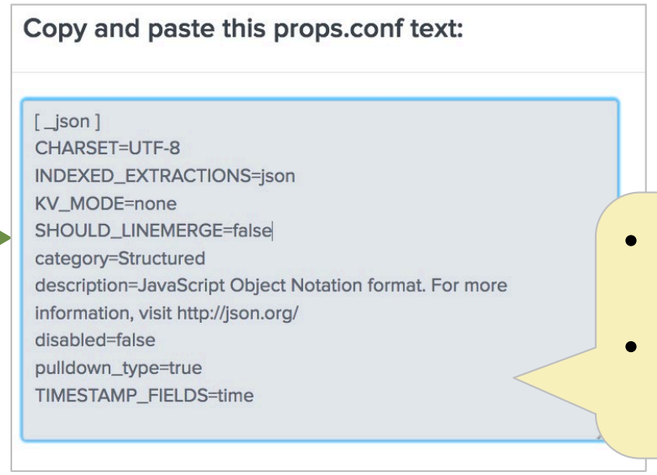
http://docs.splunk.com/Documentation/Splunk/latest/Data/HowSplunkextractstimestamps

Saving New Source Type





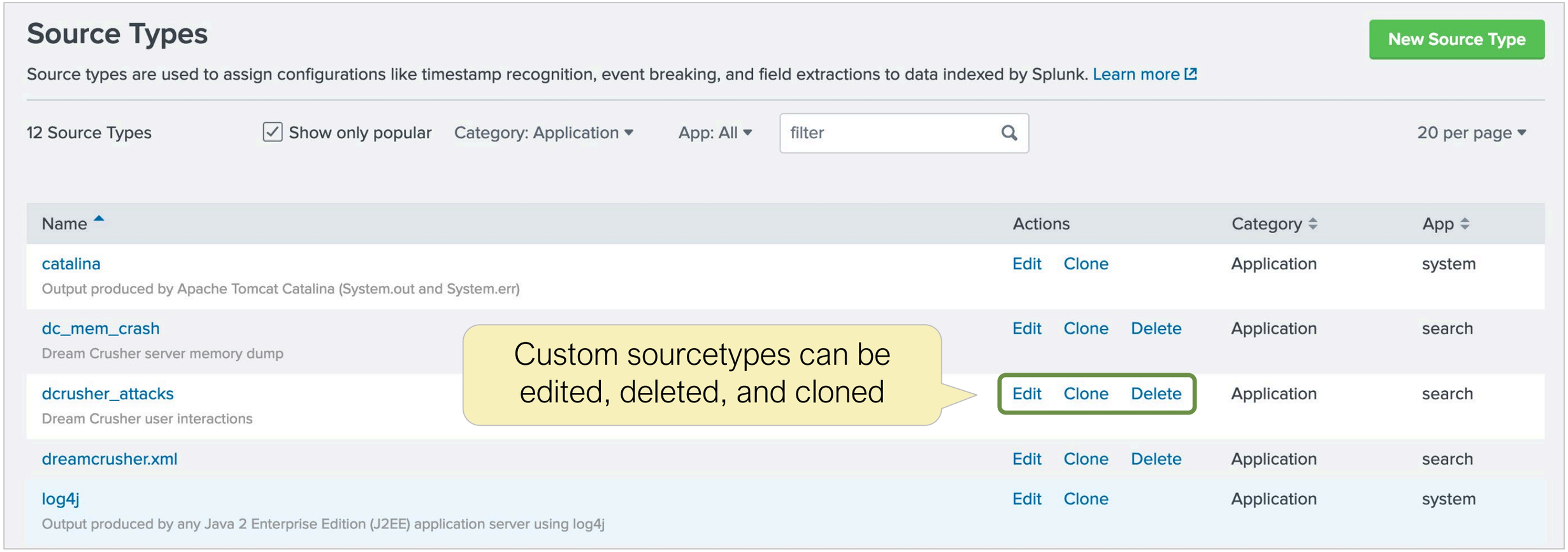
When saved, the source type becomes a custom source type that can be re-used



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

Source Type Manager

Settings > **Source types** allows access to configured sourcetypes independent of the **Add Data** wizard



Module 11 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 11 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.

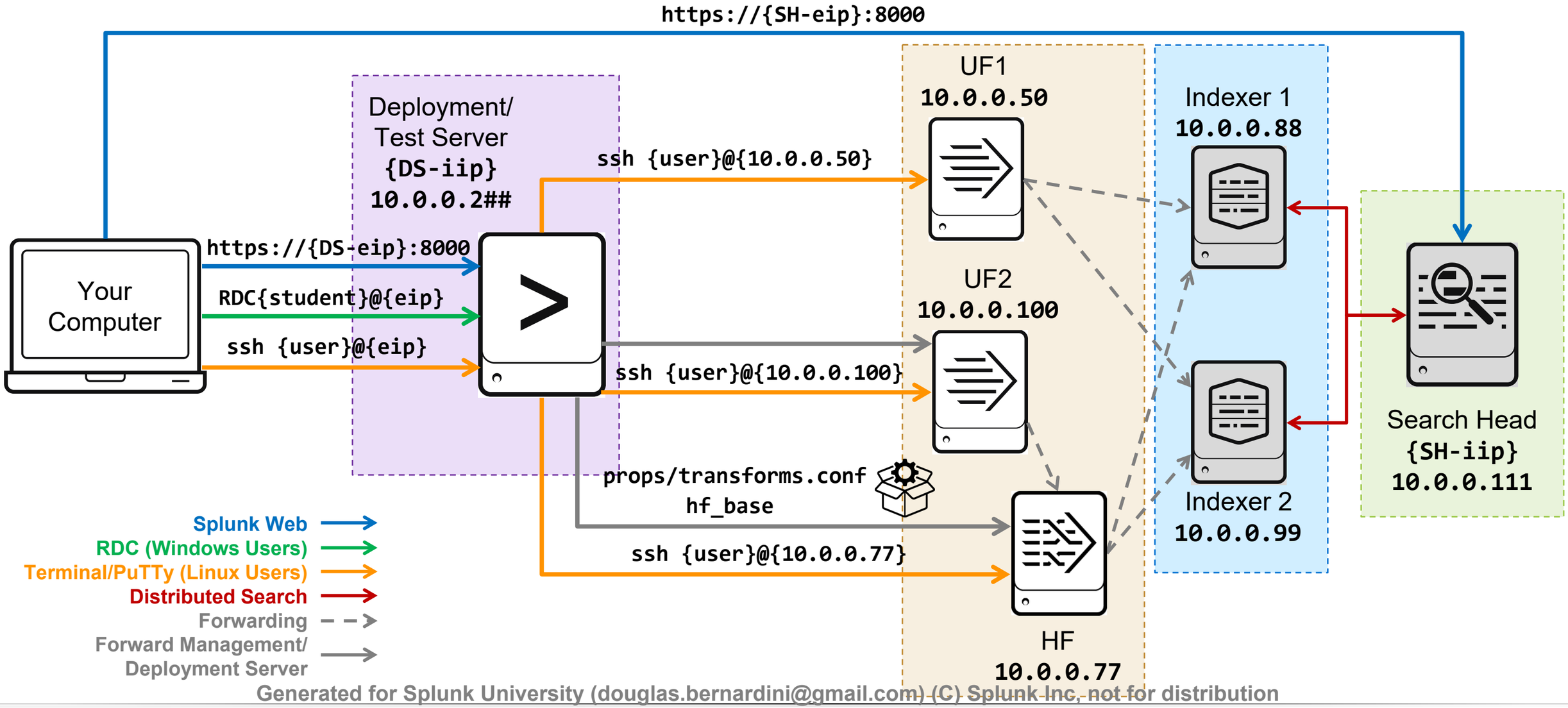
- True or False. Event boundaries can be defined using **`props.conf`** at the UF.

  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 11 Lab Exercise – Environment Diagram



Module 11 Lab Exercise

Time: 20-25 minutes

Description: Create a New Source Type

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 12: Manipulating Raw Data

Module Objectives

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

Modifying the Raw Data

Sometimes necessary prior to indexing

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

Should be performed with *extreme* care

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

Splunk Transformation Methods

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

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

Using SEDCMD

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

```
[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
```

Replace with **AcctID=xxxxx99288**

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

props.conf

\1 Indicates the capture group

- Refer to: docs.splunk.com/Documentation/Splunk/latest/Data/Anonymizedata

Using TRANSFORMS

- Per event transformation based on REGEX pattern matches
- Invoked from **`props.conf`**
- Defined in **`transforms.conf`**
- Based on attributes:

`SOURCE_KEY`	Which field to use as source for pattern matching (default: **`_raw`**: unprocessed text of all events)
`REGEX` *	Events from the **`SOURCE_KEY`** that will be processed, with optional regex capture groups
`DEST_KEY` *	Where to write the processed data
`FORMAT` *	Controls how **`REGEX`** writes the **`DEST_KEY`**

`props.conf`

```
[sourcetype]
TRANSFORMS = stanzaName
```

`transforms.conf`

```
[stanzaName]
SOURCE_KEY = ...
REGEX = ...
DEST_KEY = ...
FORMAT = ...
```

* required

Masking Sensitive Data

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

props.conf

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

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

transforms.conf

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

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

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


Setting Per-Event Source Type

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

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

props.conf

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

transforms.conf

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

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

Setting Per-Event Host Name

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

props.conf

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

transforms.conf

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

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

Per-Event Index Routing

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

props.conf

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

transforms.conf

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

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

Filtering Unwanted Events

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

props.conf

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

transforms.conf

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

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

Routing Events to Groups using HF

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

props.conf

```
[default]
TRANSFORMS-routing=errorRouting

[syslog]
TRANSFORMS-routing=syslogRouting
```

transforms.conf

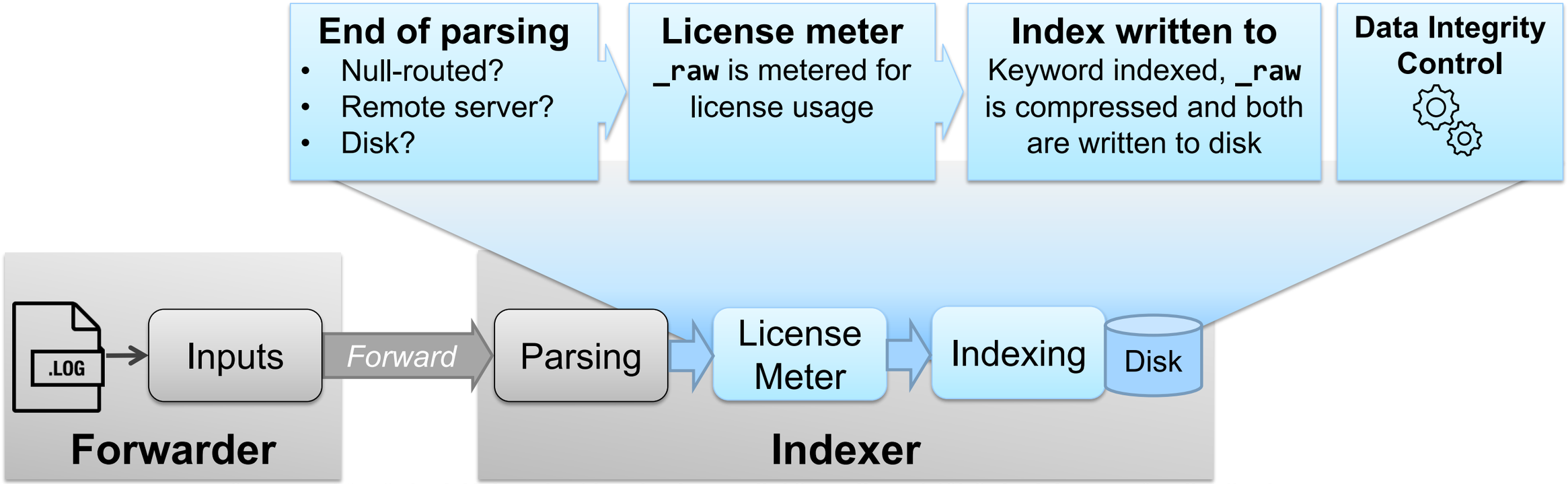
```
[errorRouting]
REGEX = error
DEST_KEY=_TCP_ROUTING
FORMAT = errorGroup

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

outputs.conf

```
[tcpout]
defaultGroup=everythingElseGroup

[tcpout:errorGroup]
server=10.1.1.200:9999

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

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


Indexing Phase Details

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



Persisted to Disk

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

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

Module 12 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 12 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.

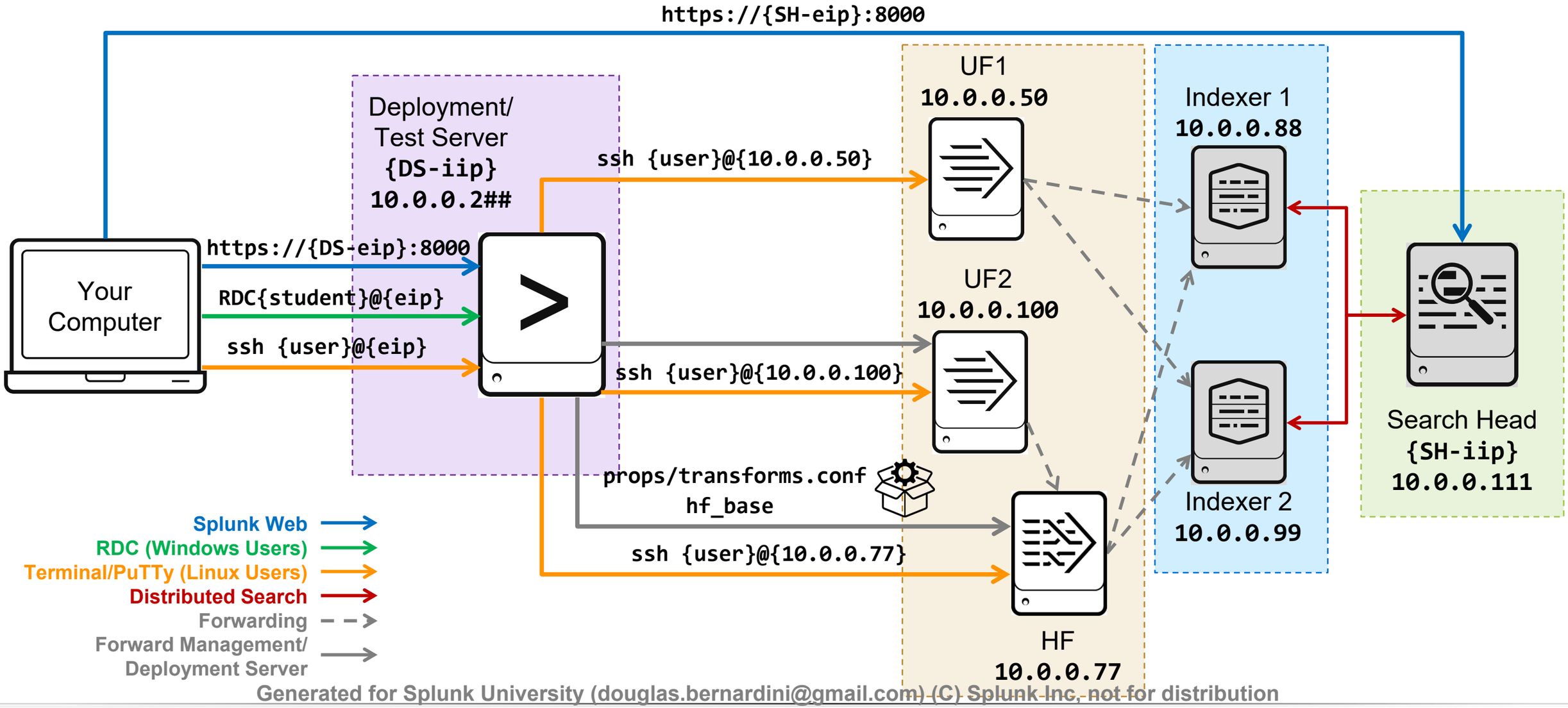
  True. If you do not specify the **SOURCE_KEY** in **transforms.conf**, it defaults to **_raw**.

- 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 12 Lab Exercise – Environment Diagram



Module 12 Lab Exercise

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

Description: Manipulating Data

Tasks:

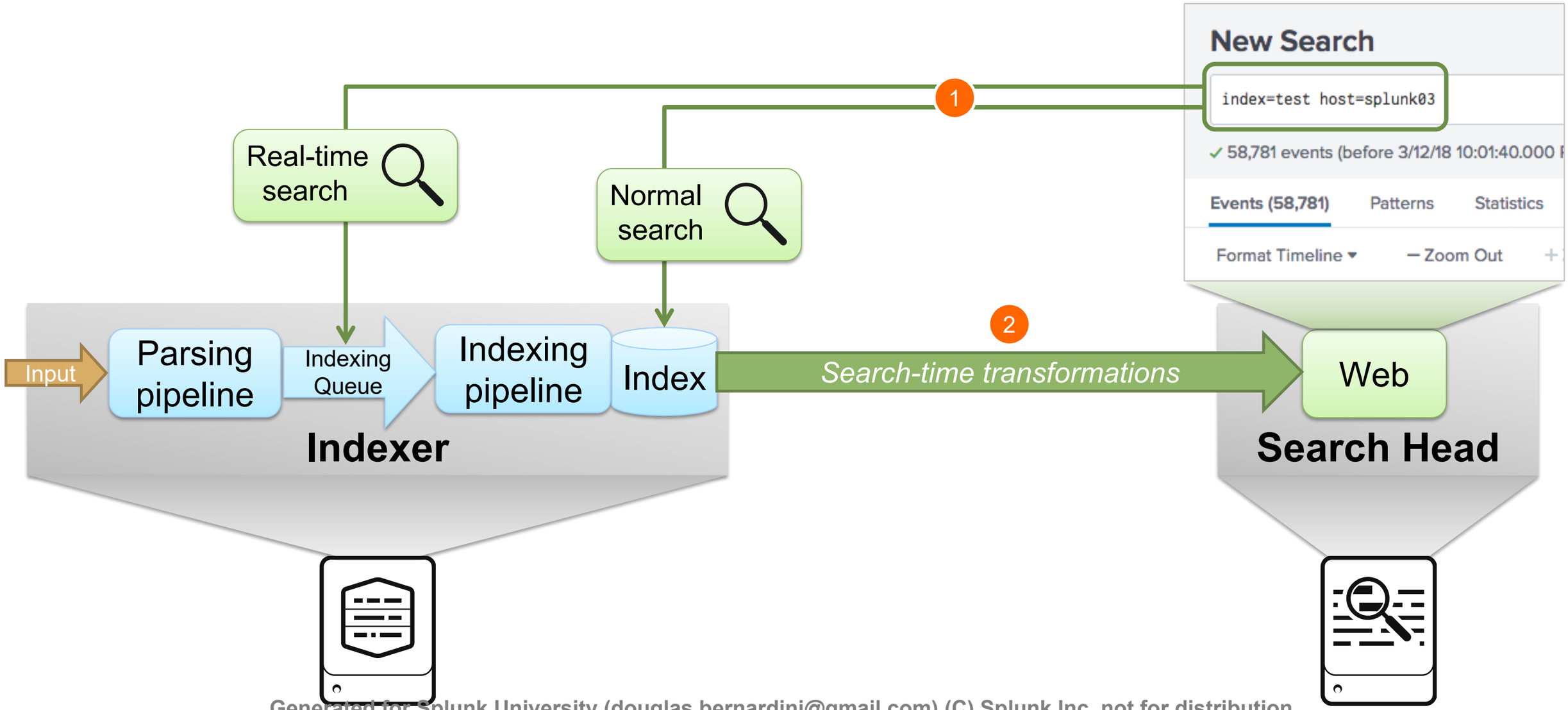
- Use **`props.conf`** and **`transforms.conf`** to:
  - Mask sensitive data
- (Optional lab exercise)
  Use **`props.conf`** and **`transforms.conf`** to:
  - Redirect events to specific indexes
  - Drop unwanted events

Module 13:
Supporting Knowledge Objects

Module Objectives

- Define default and custom search time field extractions
- Identify 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

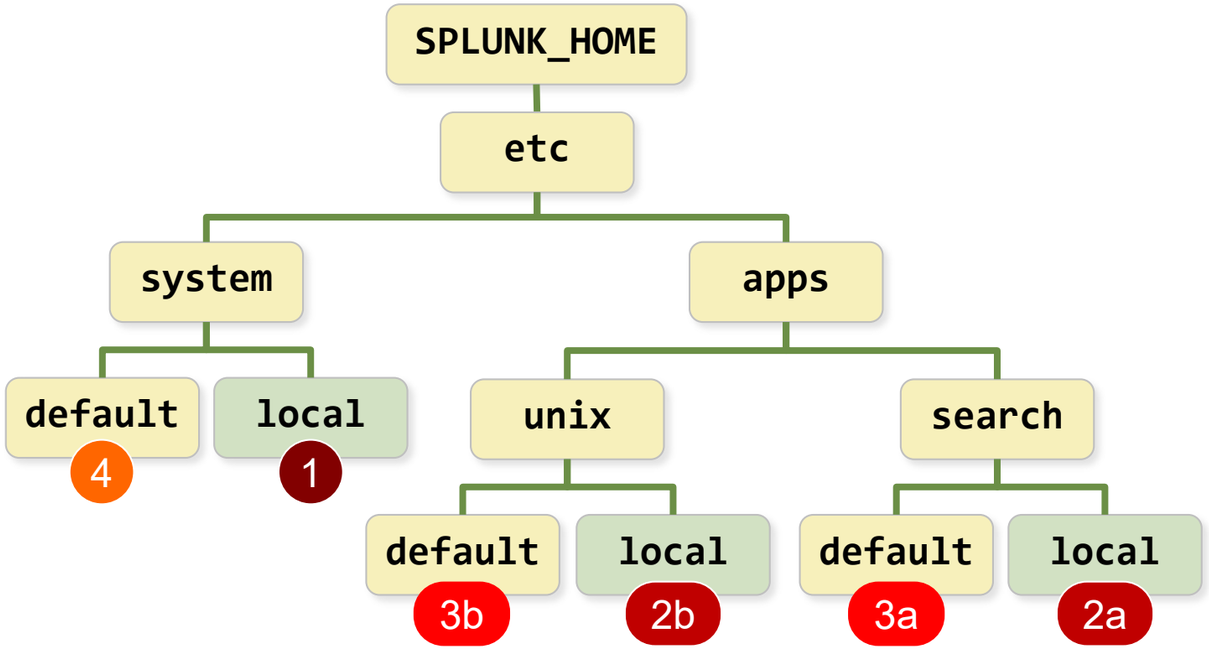


File Context and Index-time versus Search-time

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

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

Review: Index-Time Precedence (Global Context)



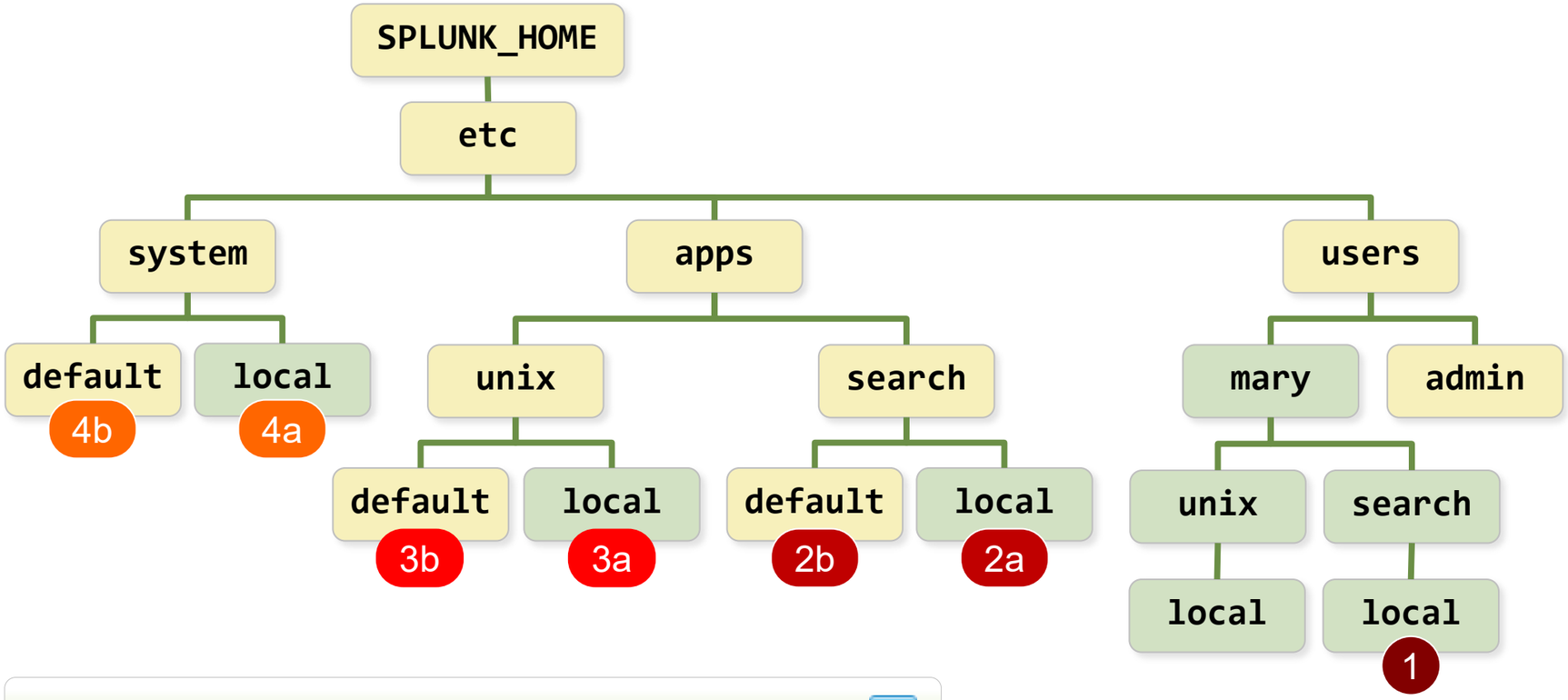
Precedence order

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

Note

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

Search-Time Precedence (App/User Context)



Precedence order

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

Note

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

Indexed Field Extraction

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

Pros/Cons of Indexed Field Extractions

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

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

Configuring Indexed Field Extractions

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

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

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

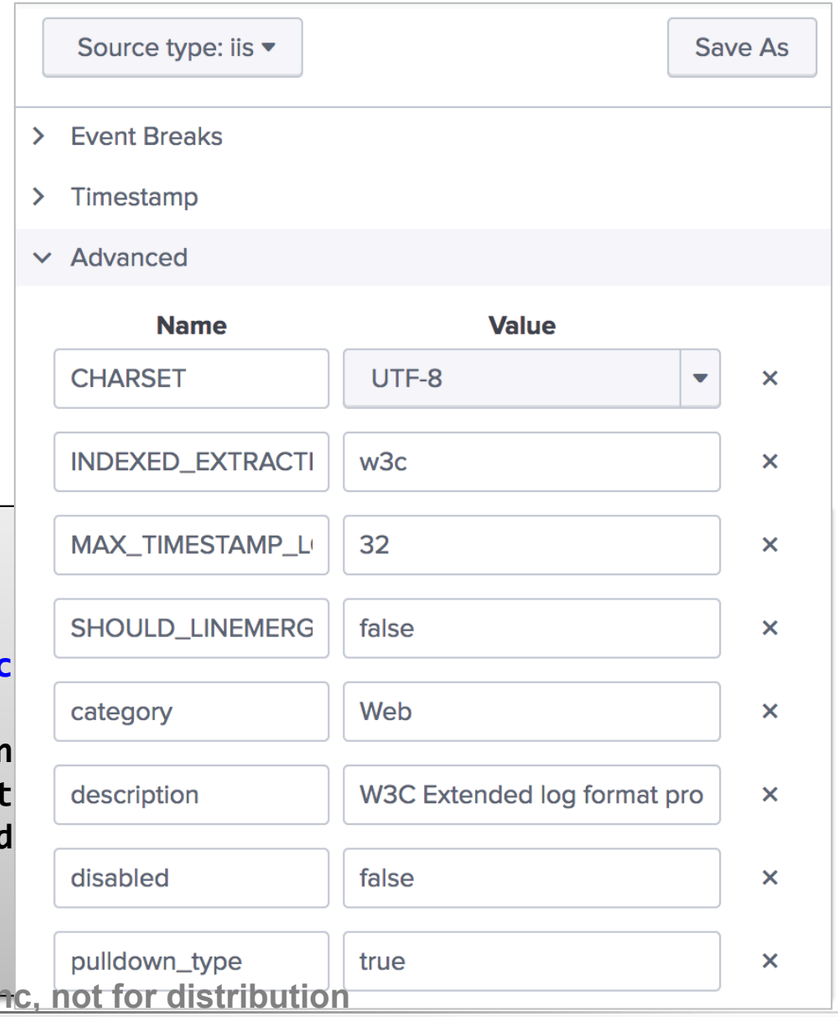
`fields.conf`	Search Head	`[error_code]` `INDEXED=true`

Structured Data Field Extraction Example

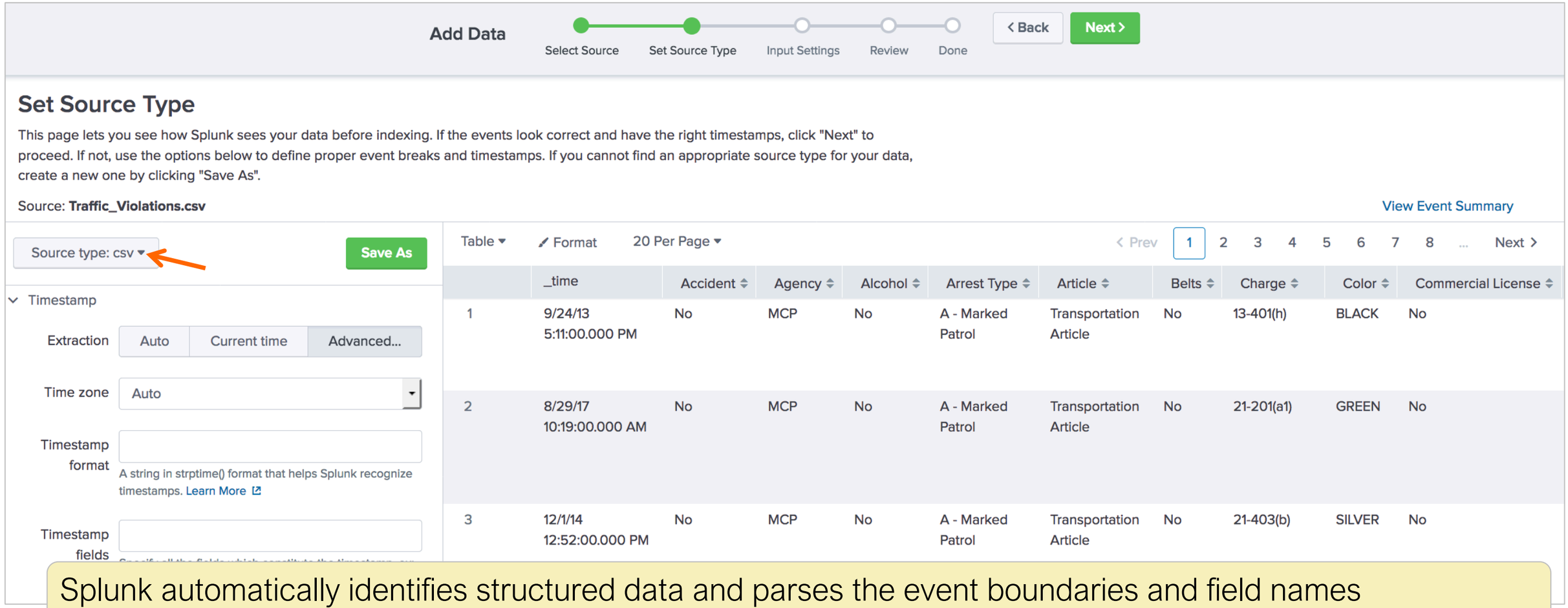
- Indexed extractions are **input phase `props.conf`** settings
  - In this scenario, the settings belong on forwarder
  - Check **`props.conf.spec`** for more options

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

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



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

Indexed Field Extractions – Caveat

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

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

Default Search Time Field Extractions

- Provided by Splunk for common source types
- Can be discovered by Splunk from your search results
  - Automatically detects key/value pairs (e.g. **a=1**)
- Can be added with add-ons and apps

*NIX app	Has many search time fields for standard UNIX logs, such as **secure.log**, **messages.log**, and so on
Windows app	Has many defaults for Windows data
For other data	Look for an app on splunkbase.splunk.com specifically designed for that type of data

Custom Search Time Field Extractions

SPL

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

Field Extractor

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

Configuration files

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

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

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

REPORT Extractions in props.conf

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

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

Using EXTRACT and REPORT in props.conf

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

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

props.conf

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

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

Extracted field name

Process this stanza in **transforms.conf**

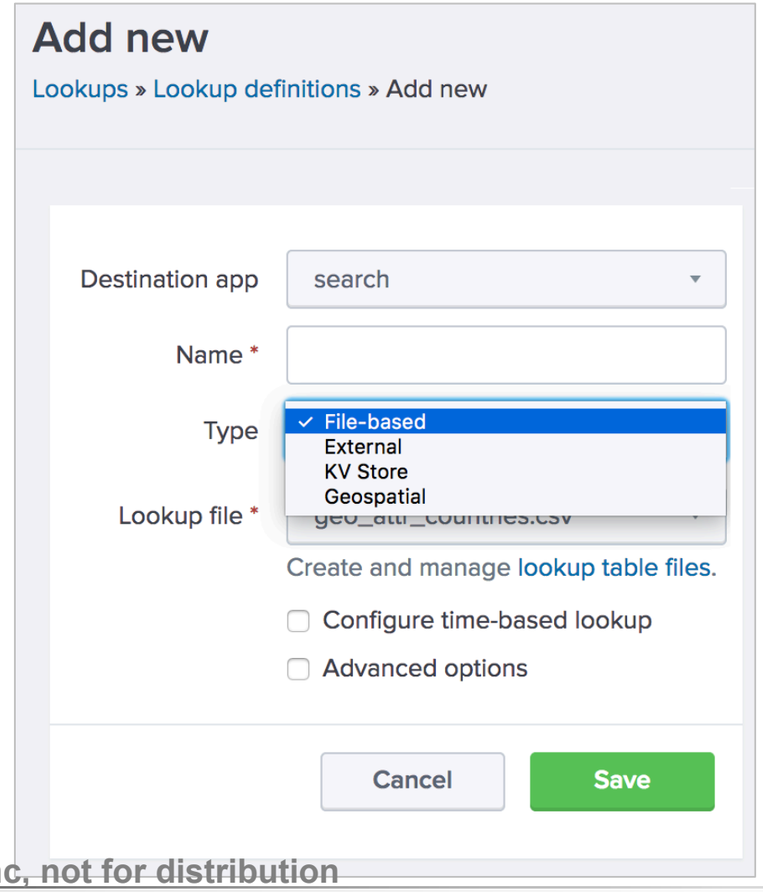
transforms.conf

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

Lookups

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

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



Other Search Time Knowledge Objects

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

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

Orphaned Knowledge Objects (KOs)

What are orphaned knowledge objects?

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

Issues with orphaned knowledge objects

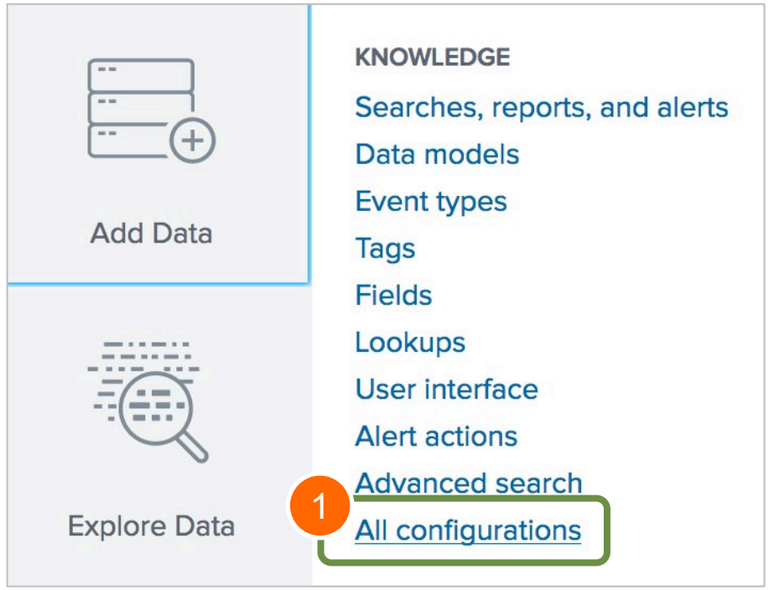
- Can cause performance problems and security concerns
- Searches that refer to an orphaned lookup may not work
- Search scheduler cannot run a report on behalf of a nonexistent owner

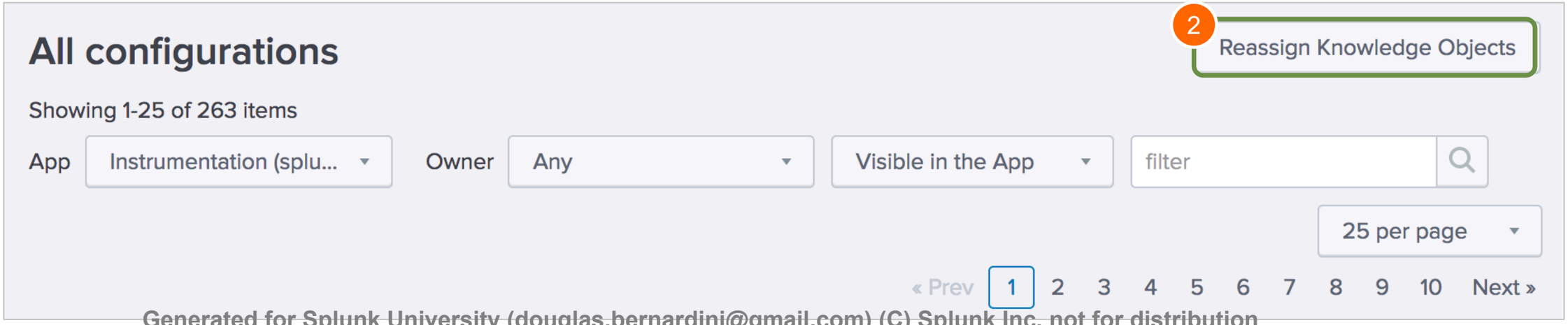
Locating Orphaned Knowledge Objects

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

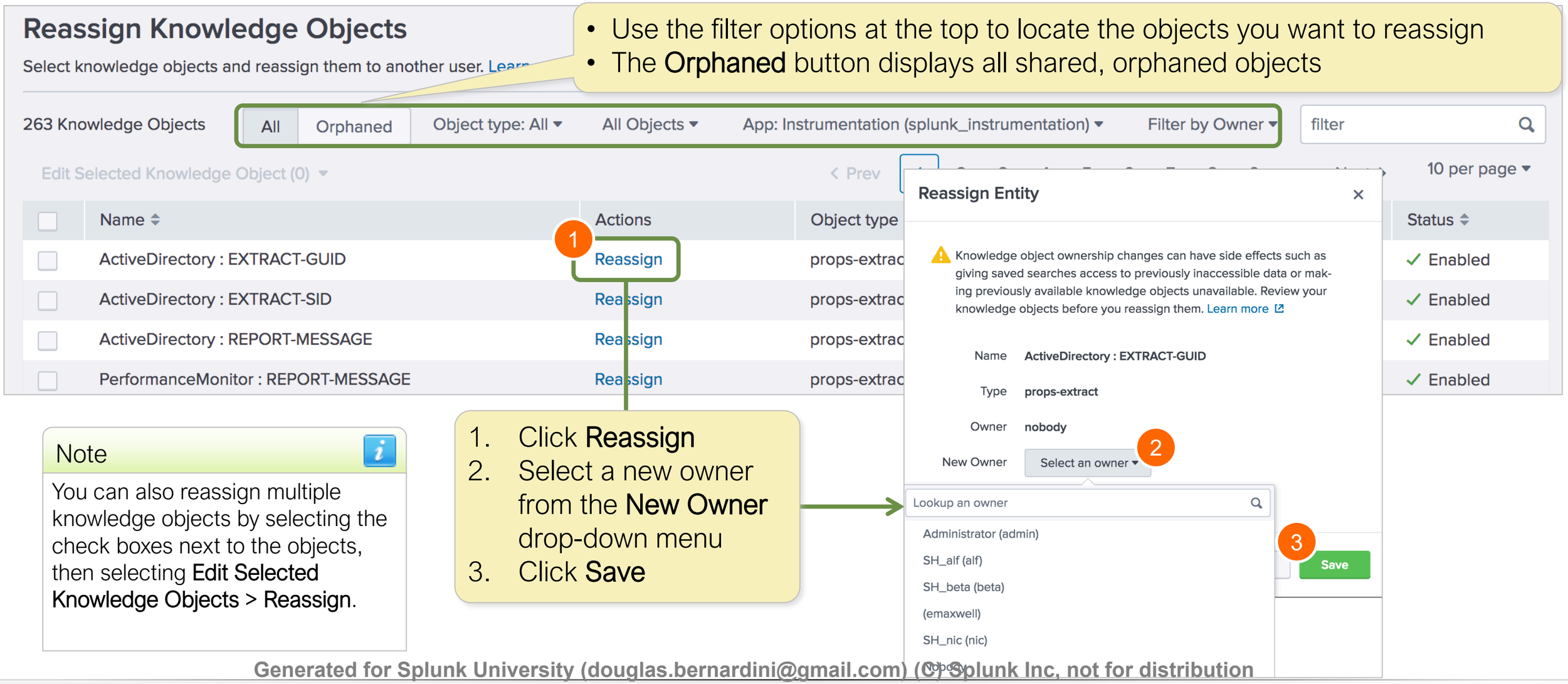
Reassigning Knowledge Objects

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





Reassigning Knowledge Objects (cont.)



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

Note

You can also reassign multiple knowledge objects by selecting the check boxes next to the objects, then selecting **Edit Selected Knowledge Objects > Reassign**.

Module 13 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 13 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 13 Lab Exercise

Time: 5-10 minutes

Description: Knowledge Object (KO) Administration

Tasks:

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

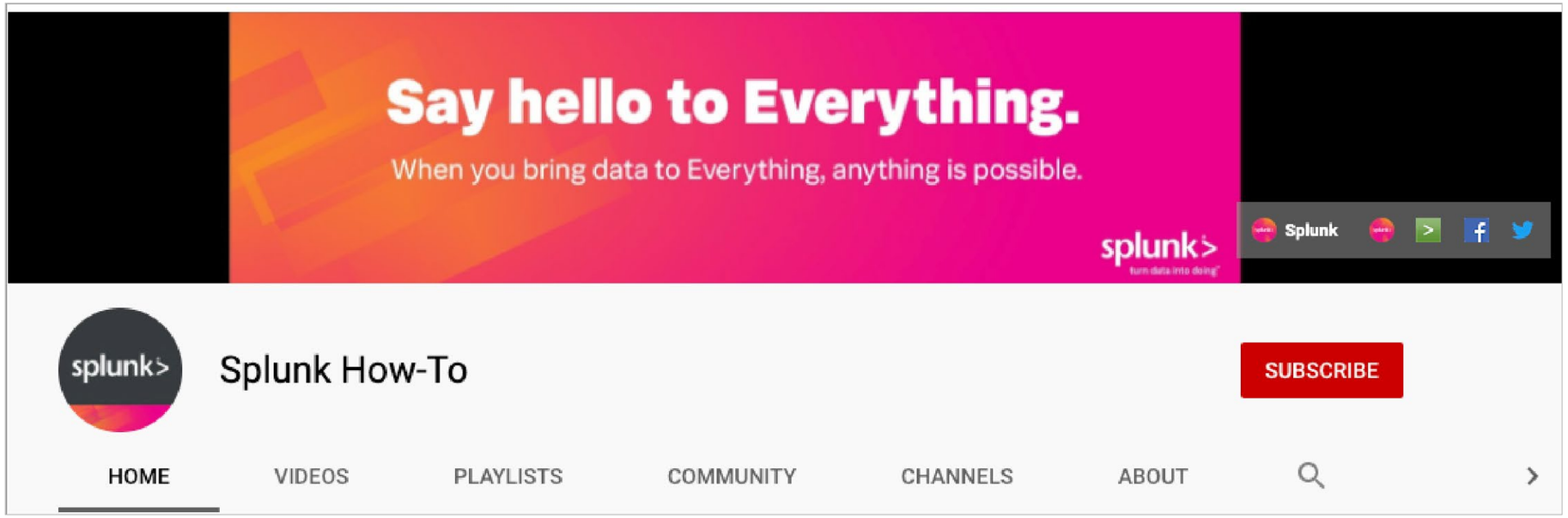
Wrap-up Slides

Community

- Splunk Community Portal
  splunk.com/en_us/community.html
  - Splunk Answers
    answers.splunk.com
  - Splunk Apps
    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

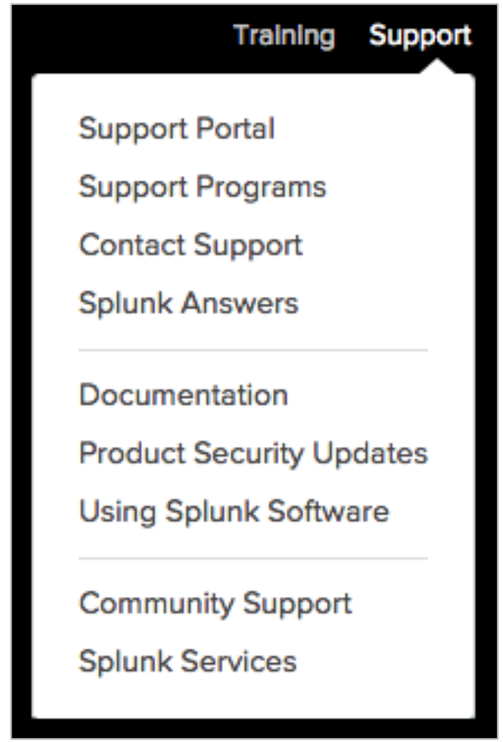
Splunk How-To Channel

- Check out the Splunk Education How-To channel on YouTube: splk.it/How-To
- Free, short videos on a variety of Splunk topics



Support Programs

- **Web**
  - Documentation: dev.splunk.com and docs.splunk.com
  - Wiki: wiki.splunk.com
- **Splunk Lantern**
  Guidance from Splunk experts
  - lantern.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)



splunk> .conf21

.conf21 Las Vegas

October 18–21

.conf21 Virtual

October 19–20

Splunk University

October 16–18

Thank You