

# Splunk Federated Data Management Workshop

## Lab Guide

### Overview

This workshop will help you understand the power of **Ingest Actions (IA)**, **Edge Processor (EP)**, and **Ingest Processor (IP)** and how they help manage your data quickly as it streams into Splunk. You'll learn how to use these tools from a web browser and how to filter, mask, transform, enrich, and route data to multiple destinations. You'll also see that by using these tools, you can get faster feedback that your changes are working.

In this hands-on workshop, you will work with all three tools in a Splunk Cloud environment. This practical setting allows you to see where these components reside, when you should use them, and how they each process data. The interactive approach will ensure that you leave the workshop with a confident understanding of these tools and their applications.

Throughout the labs in this workshop, you will work with Cisco ASA logs. Using Ingest Actions, you will complete the base use case and build on it with Edge Processor and, finally, Ingest Processor.

# Table of Contents

<b>Overview.....</b>	<b>1</b>
<b>Table of Contents.....</b>	<b>2</b>
Base Use Case.....	4
Lab Connection Info.....	4
Sample Data.....	4
Accessing Your Environment.....	5
<b>Lab Exercise 1: Ingest Actions.....</b>	<b>6</b>
Description.....	6
Accessing Ingest Actions.....	6
Creating a Ruleset.....	7
Select Sample Data.....	7
Route Data to S3.....	9
Filter Data.....	12
Mask Data.....	13
Send Data to a Splunk Index.....	16
Save and Check Your Work.....	18
Splunk Federated Search for Amazon S3.....	19
<b>Lab Exercise 2: Edge Processor.....</b>	<b>21</b>
Description.....	21
Access Your Splunk Cloud Instance.....	22
Create a Workspace.....	23
Add a new Edge Processor.....	25
Install a New Edge Processor Instance.....	28
Restart Your Splunk Heavy Forwarder (Manual Step for Today's Environment).....	29
Verify Edge Processor is Receiving Data.....	30
Create a Pipeline.....	30
Save Your Pipeline Progress.....	36
Author Your Pipeline.....	38
Filtering.....	40
Masking.....	41
Enrichment Via Lookup.....	42
Drop Fields & Set Index.....	44
Apply the Pipeline to your Edge Processor.....	46
Verify Edge Processor is Receiving Data.....	48
Check Your Data in Splunk Cloud.....	50
<b>Lab Exercise 3: Ingest Processor.....</b>	<b>51</b>
Description.....	51
Access Your Ingest Processor Console.....	51
Create a Workspace.....	52
Create a Pipeline.....	53
Create a Preview Snapshot.....	56

Save Your Pipeline Progress.....	61
Author Your Pipeline.....	63
Filtering.....	65
Masking.....	67
Drop Fields & Set Index.....	68
Apply Your Pipeline.....	70
Check Your Data in Splunk Cloud.....	71
Convert Logs To Metrics.....	71
Check Your Data in Splunk Observability.....	74
<b>Lab Summary.....</b>	<b>77</b>
<b>Get Started with Splunk Data Management Today.....</b>	<b>78</b>
Request access to Edge Processor or Ingest Processor.....	78
Join the Slack channel.....	78
Splunk Adoption Hub.....	78
Splunk Federated Data Management Lab Guide.....	2

## Base Use Case

The base use case you will use for all three labs follows. Edge Processor and Ingest Processor build on this. Those details are in their respective lab sections.

### Ingest Cisco ASA Logs

The environment already has the Cisco ASA logs set up for you. You will need to work with them to achieve the lab goals.

#### Remove Events with Lower-Level Information

You will send information you don't need daily to S3 instead of a Splunk index. As part of your configuration, removing events with lower-level details is important to streamline your data and focus on the most relevant logs. You still want to keep these logs in a lower-cost object store for compliance. Send logs with the following information to S3 instead of your Splunk index:

1. Built inbound UDP connection.
2. Built inbound TCP connection.
3. Teardown UDP connection.
4. Teardown TCP connection.
5. Disallowing new connections.

#### Mask the User Name Field

Before indexing the Cisco ASA logs in Splunk, you want to mask the user name, which we regard as sensitive.

## Lab Connection Info

Your workshop instance URL and credentials will be provided on a handout. Please refer to that on how to access the workshop.

## Sample Data

Sample data for this workshop can be downloaded below. Both links point to the same sample file and are provided for ease of access:

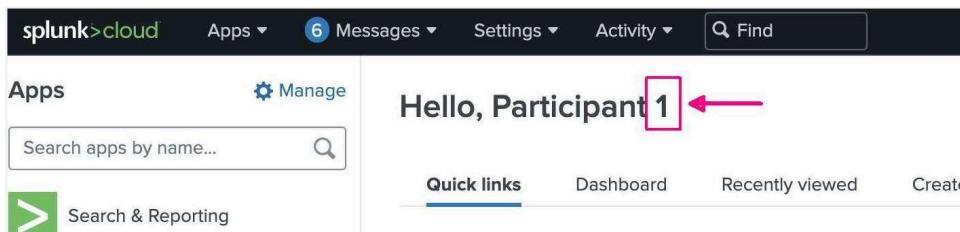
Direct Link: [cisco\\_logs\\_small.sample](#)

## Accessing Your Environment

1. Connect to the lab environment as described in the section above. Labs with unique components may provide specific access instructions for those components.
2. Log into your assigned Splunk Cloud instance via the URL provided. After logging in, you may see a Terms of Service screen. If so, click the “Accept” checkbox as appropriate and continue.



3. Once logged into your Splunk Cloud instance, you'll see this screen.



- a. The highlighted number corresponds to the sourcetypes you will use throughout the labs. In the example image above, your number would be **1**, so you would use the following sourcetypes through the lab:
  - i. Ingest Actions lab uses the `cisco:asa:ia:#` sourcetype.
  - ii. Edge Processor lab uses the `cisco:asa:ep:#` sourcetype.
  - iii. Ingest Processor lab uses the `cisco:asa:ip:#` sourcetype.
4. The lab is set up this way so lab participants don't interfere with each other.
5. **Write your participant number down, as you will need it for all labs.**

# Lab Exercise 1: Ingest Actions

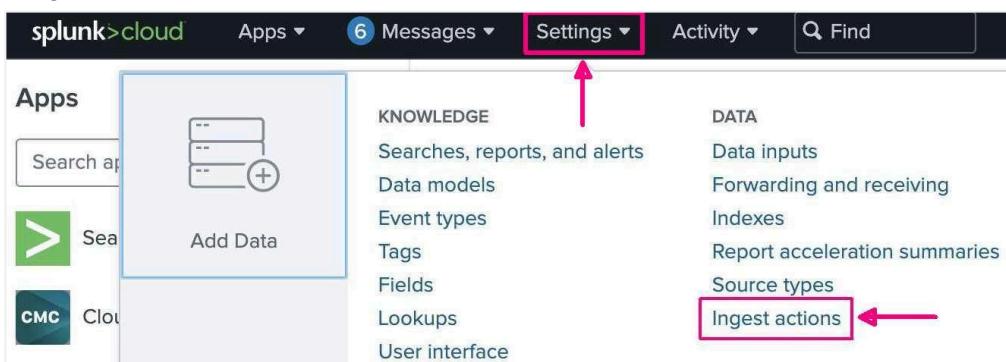
## Description

In this lab, you will filter, mask, and route Cisco ASA logs using Ingest Actions ingest-time transformation and routing rules. You will implement the [Base Use Case](#) described at the beginning of this guide.

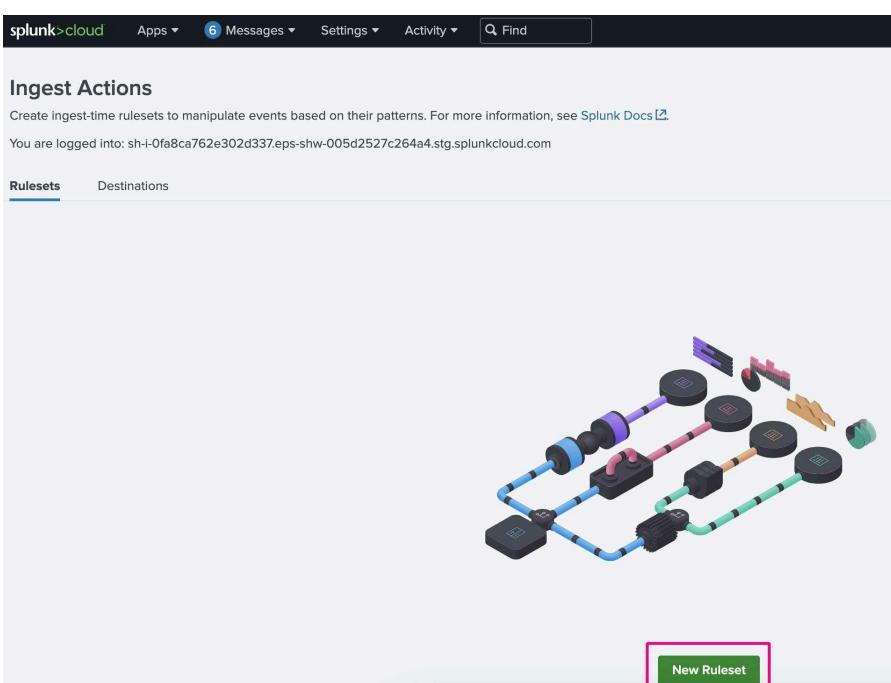
## Accessing Ingest Actions

1. Be sure you've read and followed instructions from the [Lab Connection Info](#) and [Accessing Your Environment](#) above.

Log in to your Splunk Cloud environment. Browse to the **Settings** menu at the top of your screen and choose **Ingest Actions** in the upper right area of the menu, as shown in the image below.



2. This will take you to the main Ingest Actions configuration page. Click on the **New Ruleset** button near the bottom of your screen.



## Creating a Ruleset

3. You're now on the *Create New Ruleset* configuration screen. The first thing to do is name your ruleset where you see **Enter Ruleset Name**. Since multiple people are using the same host as you, simply use **Participant #, REPLACING # WITH YOUR NUMBER FROM THE Accessing Your Environment SECTION.**

The screenshot shows the Splunk Cloud interface with the title 'Create New Ruleset'. At the top, there is a navigation bar with links for 'splunk>cloud', 'Apps', 'Messages', 'Settings', 'Activity', 'Find', and a search icon. Below the title, there is a large input field labeled 'Enter Ruleset Name' which is currently empty. Underneath it is a smaller input field for 'Enter Ruleset Description'. To the right of the main form area, there is a sidebar titled 'Event Stream' and a 'Data Preview for Event' section which is currently empty. At the bottom left of the main form, there are two buttons: 'Indexed Data' and 'Sample File'.

## Select Sample Data

4. Looking down this screen, find the **Sourcetype** pull-down menu, as shown in the image below.

The screenshot shows the 'Sample' configuration screen. At the top, there is a dropdown menu labeled 'Sourcetype' with the option 'Select sourcetype...' selected. Below it are three more dropdown menus: 'Sample size' set to '100', 'Sample ratio' set to 'No event sampling', and 'Time Range' set to 'Last 60 minutes'. At the bottom right of the form is a green button labeled 'Sample' with a magnifying glass icon.

5. Expand the **Select sourcetype** menu by clicking on it. You will see several sourcetype categories. In the filter text box at the top, type in **cisco:asa:ia:#**, **REPLACING # WITH YOUR NUMBER FROM THE [Accessing Your Environment SECTION](#)**. Finally, click on **Use cisco:asa:ia:#** in the pull-down to select it for use. See the image below for an example.



6. Now, set the **Sample size** to **5000** and the **Time Range** to **Last 7 days** so you get a good sample of data. Finally, click the **Sample** button to bring in data to work with. You should see data populated on the right side of the screen.

Sample size ?  
5,000 ←

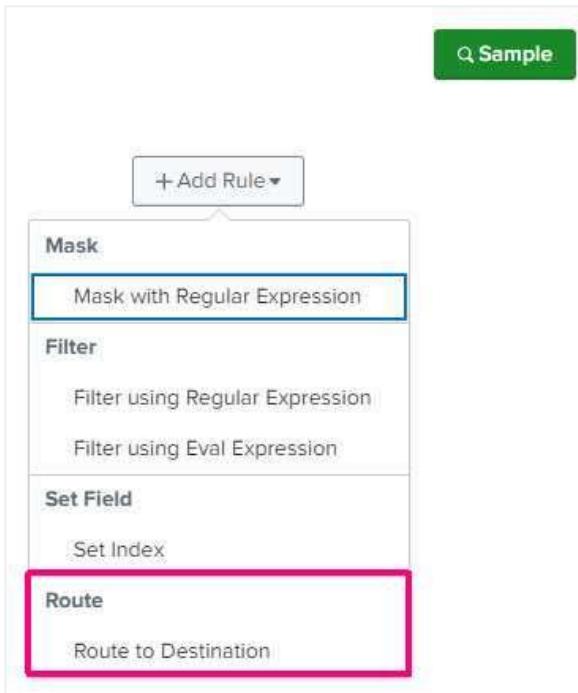
Sample ratio ?  
No event sampling

Time Range ?  
Last 7 days ←

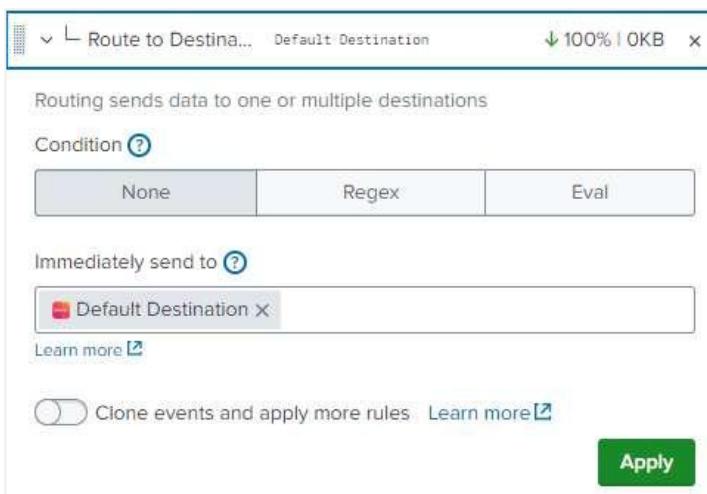
Sample

## Route Data to S3

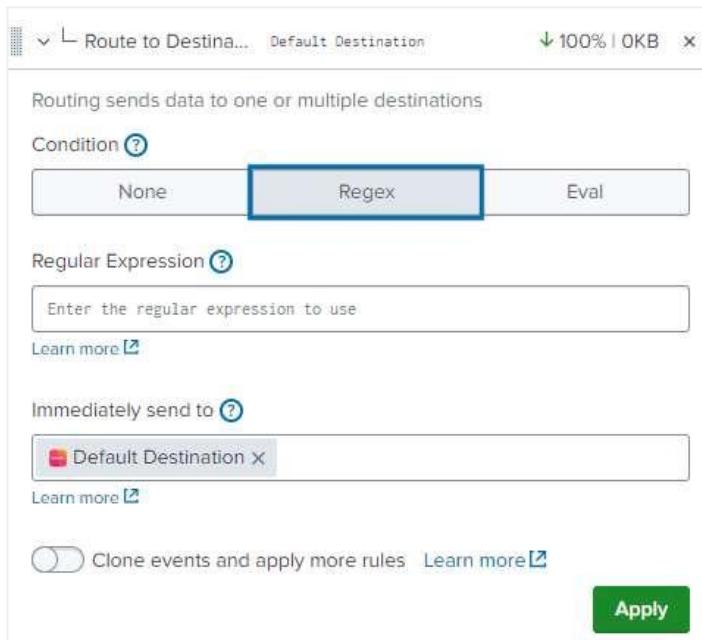
7. Now, we can add rules to our ruleset. You'll see several choices when you click the **Add Rule** dropdown, as shown below. Remember, the first thing you want to do is send low-level information events to S3. This means you want to **Route to Destination**.



8. Selecting **Route to Destination** will add a new rule to your ruleset, as shown in the image below.



9. Since we want to filter out events with various text, let's use a Regex to do that. Click the Regex button to show a textbox where you can enter a regular expression to target specific events



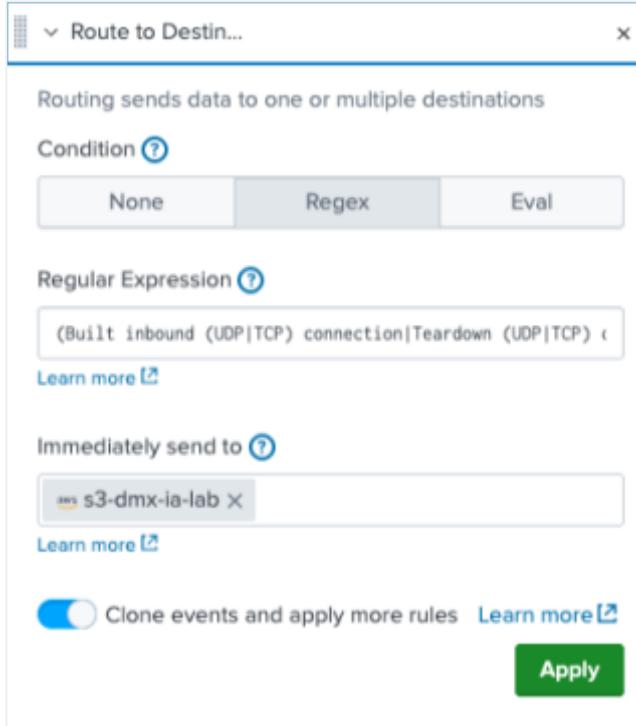
10. Refer back to the [Base Use Case](#) in the “Remove events with lower-level information” section on what events you need to target. Create a regular expression to match all of them and put it in the **Regular Expression** text field.

**NOTE:** If you need help creating a regular expression, you can use this one:

(Built inbound (UDP|TCP) connection|Teardown (UDP|TCP)  
connection|Disallowing new connections)

11. Next, you'll choose the destination to send events to and make sure to "clone" events so you can apply more rules.

- For the destination, you will choose the S3 location called `s3-dmx-ia-lab` and ensure you toggle the **Clone events and apply more rules**. If you don't toggle cloning, any rules after this point won't apply.
- Also, remove the **Default Destination** so events don't get sent to a Splunk index.
- See the image below for how your configuration should look.



**NOTE:** Make sure you click the "Apply" button after configuring each rule. Otherwise, Ingest Actions will not save the configuration when you move on to the next rule.

12. Once you click the "Apply" button, look to the upper right of the events view on the right side. You will see three buttons: *All Events*, *Affected Events*, and *Unaffected Events*. These can help you ensure your rules are working properly.

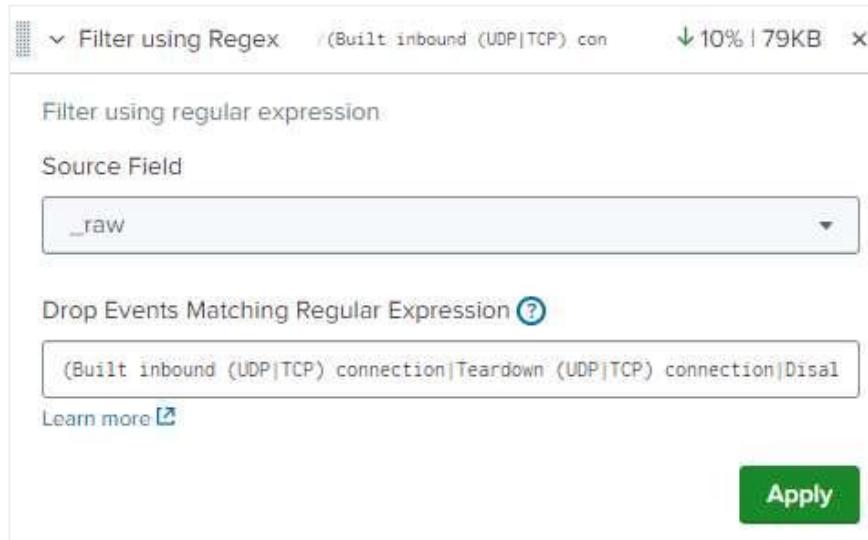
- All Events** shows all events being sampled.
- Affected Events** shows which events are affected by the current rule.
- Unaffected Events** shows which events are not affected by the current rule.
- The numbers you see next to these will likely not match the screenshot below, which is OK. In the image, we can see that out of 500 events, the S3 destination routing regex affected 44 events and did not affect 456 events. This means we'll get those 44 events in our S3 bucket.



## Filter Data

13. Now that events are properly routed to S3, let's add the next rule. Since you only routed the events to S3 and didn't filter them, they will still pass through the event cloning and end up in a Splunk index, which you don't want. This means you need to filter them out.

- a. Click **Add Rule** and choose **Filter using Regular Expression** to prevent them from being indexed in Splunk. The great part is that you already created the regular expression you need in the routing rule.
- b. Put the same regular expression in the **Drop Events Matching Regular Expression** text box and click the *Apply* button.
- c. You should see the same number of Affected and Unaffected events as before since you're filtering the same events you routed to S3.
- d. In the upper-right corner of the below image, you can also see the 10% reduction in sample data after we filter events.



**NOTE:** Use the same regular expression as you did in the route to S3 rule:

(Built inbound (UDP|TCP) connection|Teardown (UDP|TCP) connection|Disallowing new connections)

## Mask Data

14. Now that you've sent the low-value data to S3 **and** filtered those events out of the data stream, you need to mask the username value in the events before indexing the data to Splunk. There are several places where a username value exists, so you will use three rules to target them. For this, we will replace the user name with a value of <REDACTED>.
15. Add another rule, this time choosing **Mask with Regular Expression**. You will see this rule has two inputs - one for the match and one for the replacement. Create a regular expression to match the username. Then, enter in the replacement. Remember, you want to replace the username value with <REDACTED>.
16. Remember to click the "Apply" button after configuring the rule.

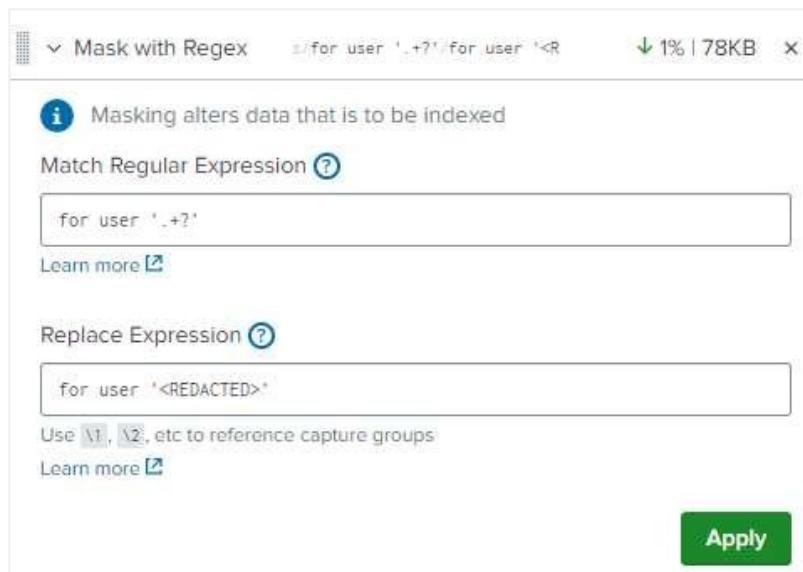
**NOTE:** These events look like this:

```
Authorization denied (acl=acmetechinbound) for user 'admin'  
from
```

**NOTE:** If you need help creating a regular expression, you can use this one:

Match Regular Expression: for user '.+?'

Replace Expression: for user '<REDACTED>'



17. Add another **Mask with Regular Expression** rule to target the next username value events.

**NOTE:** These events look like this:

```
Group = Acme_Techcde, Username = practical_lovelace, IP =  
10.11.36.23,
```

**NOTE:** If you need help creating a regular expression, you can use this one:

Match Regular Expression: Username = .+?,

Replace Expression: Username = <REDACTED>,

The screenshot shows the 'Mask with Regex' configuration page. At the top, there's a header with a dropdown arrow, the title 'Mask with Regex', a file path 's/Username = .+?, /Username =', a download progress bar '0% | 658KB', and a close button 'X'. Below the header, there's an information icon with the text 'Masking alters data that is to be indexed'. Underneath, there are two sections: 'Match Regular Expression' with the input 'Username = .+?,', and 'Replace Expression' with the input 'Username = <REDACTED>,'. Both sections have 'Learn more' links. At the bottom right is a green 'Apply' button.

18. Add another **Mask with Regular Expression** rule to target the next username value events.

**NOTE:** These events look like this:

```
local database : user = focused_euclid
```

And this:

```
between 10.11.36.18 and 10.11.36.13 (user= intelligent_haslett) has been created.
```

**NOTE:** If you need help creating a regular expression, you can use this one:

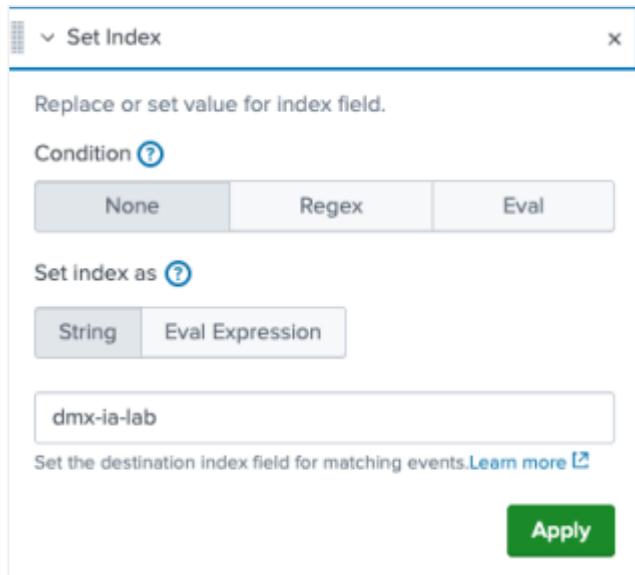
Match Regular Expression: `(user\s?= ).+?(\)|$)`

Replace Expression: `\1<REDACTED>\2`

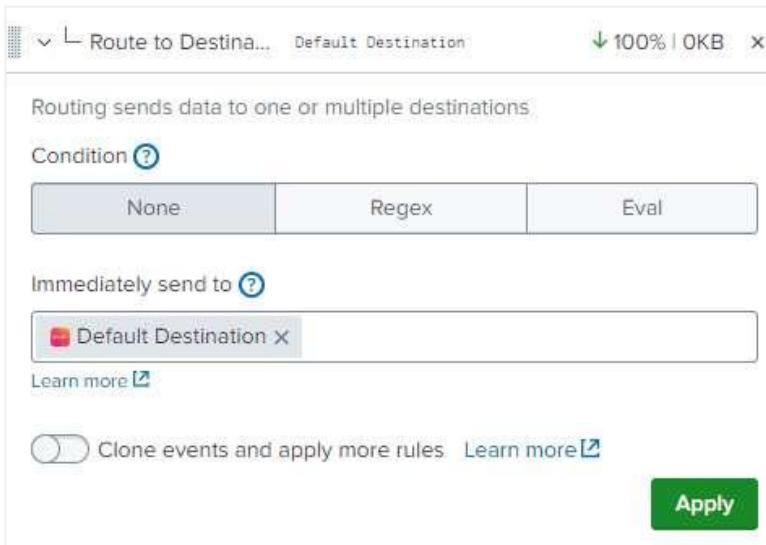
The screenshot shows the 'Mask with Regex' configuration page. At the top, there's a note about masking: 'Masking alters data that is to be indexed'. Below that, the 'Match Regular Expression' field contains the pattern `(user\s?= ).+?(\)|$)`. The 'Replace Expression' field contains the pattern `\1<REDACTED>\2`. A note below the replace field says 'Use \1, \2, etc to reference capture groups'. At the bottom right is a green 'Apply' button.

## Send Data to a Splunk Index

19. Now, you can send the filtered and masked data to a Splunk index. For this, set the index to **dmx-ia-lab** and route the data there. You do this by adding a **Set Index** rule to the ruleset.
- Since you are setting the index name to a static value, you can leave the rule configurations to default settings and put the index name in the text box.
  - Since you didn't do a *Regex* or *Eval*, this affects 100% of the remaining events.



20. Almost there - you just need to route the remaining data to Splunk. This is done similar to how you routed data to S3. Add a new rule and choose **Route to Destination**. You'll see the destination as **Default Destination**. This means that the events will get re-inserted into Splunk's normal pipeline at the indexing stage. Since you set the index to **dmx-ia-lab**, that's where the data will get indexed.



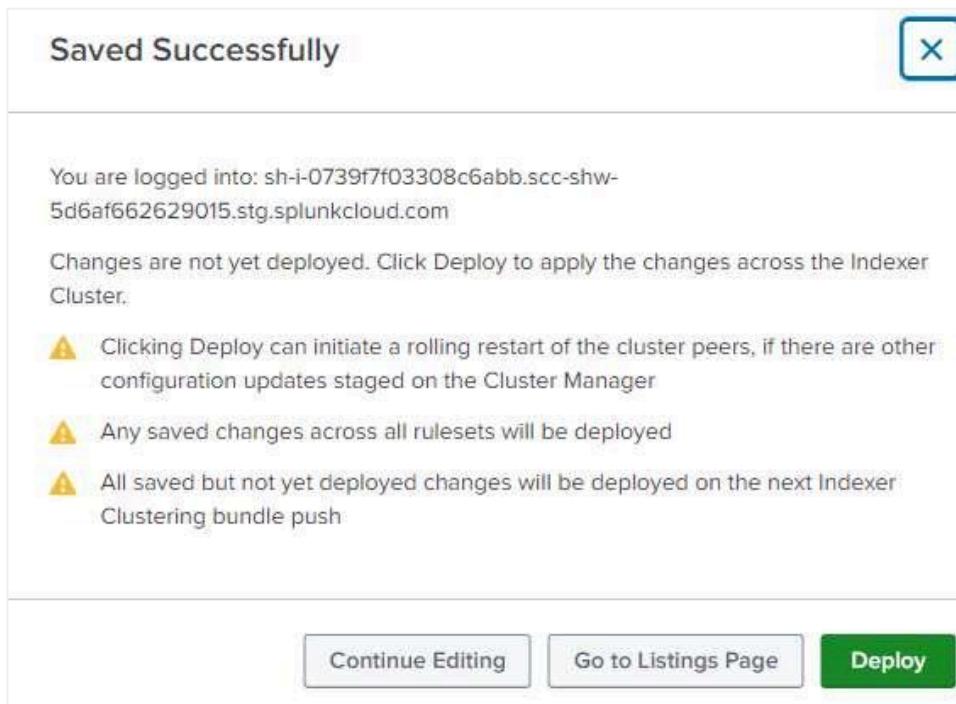
**NOTE:** Make sure you click the “Apply” button.

## Save and Check Your Work

21. Now that the ruleset is complete, in the upper-right corner of the screen, click the **Save** button.



22. A modal window may open, as shown below. If it does, click the green **Deploy** button to send your changes to the indexers. Once set, you can check your work.



23. In the upper-left corner of the screen, select **Apps** menu and then select the **Search & Reporting** app. This will take you to a search bar where you can ensure the correct data ended up in the correct index.



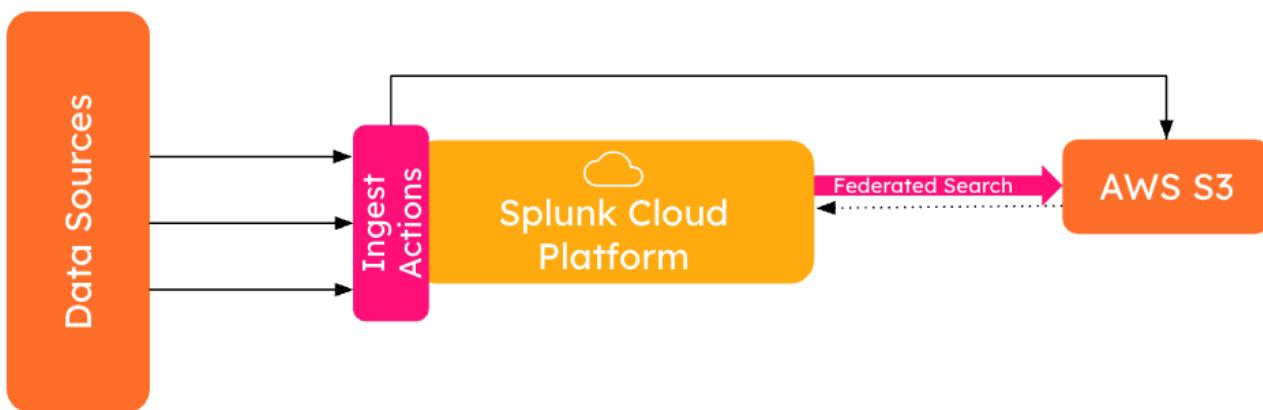
**NOTE:** You may see a modal window pop up to take a tour of Splunk. In the lower-right corner of the pop-up, click the “Skip tour” link if this happens.

24. Use the search below, replacing the sourcetype **#** portion highlighted in yellow **WITH YOUR ASSIGNED NUMBER**. You will see all **user** field values are set to <**REDACTED**>.

```
index=dmx-ia-lab sourcetype=cisco:asa:ia:# earliest=-7d@d
```

### Splunk Federated Search for Amazon S3

Federated search for Amazon S3 lets you search data in your Amazon S3 buckets from your Splunk Cloud Platform deployment.



25. Use the S3 federated search below, replacing the sourcetype **#** portion highlighted in yellow **WITH YOUR ASSIGNED NUMBER**, to see the data you sent to the S3 bucket.

```
| sdselect * FROM federated:s3-ia-lab-index WHERE  
sourcetype="cisco:asa:ia:#"
```

## Lab Exercise 2: Edge Processor

### Description

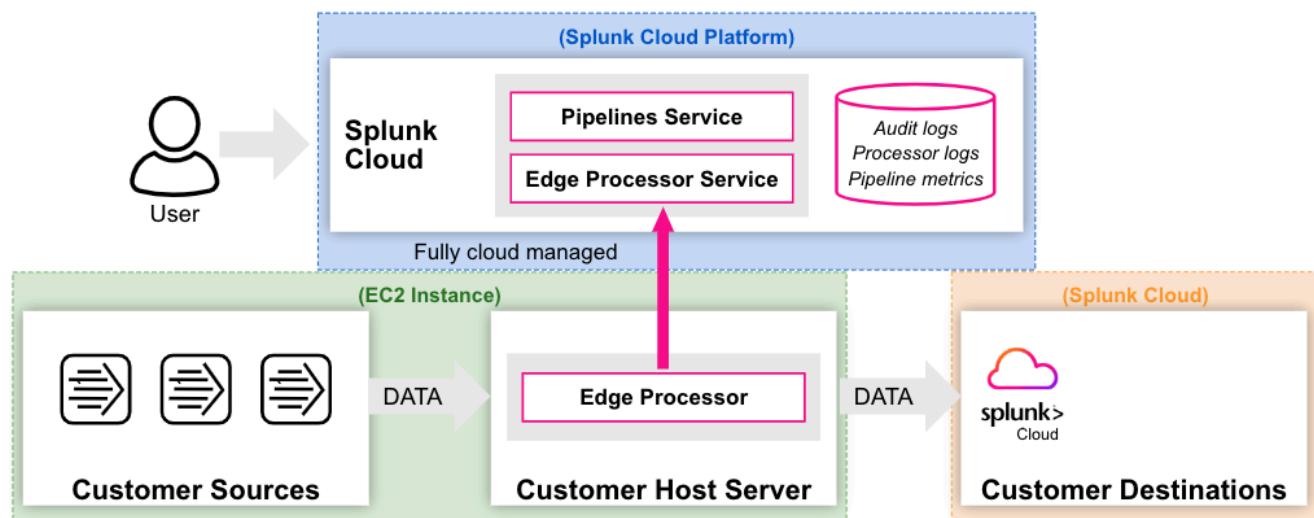
In this lab, you will filter, mask, and route Cisco ASA logs using Edge Processor. You will implement the [Base Use Case](#) described at the beginning of this guide. Additionally, the lab guide will demonstrate the ability to enrich log events using lookup commands.

An environment and an Edge Processor have been set up, and you have been provided with the credentials. In case you are interested in architectural background, today's lab environment consists of the following:

**Splunk Cloud** - Your data will be indexed here once it has been processed by Edge Processor.

**Edge Processor Service** - Hosted in Splunk Cloud Platform, this is the cloud-based console through which you will set up, configure, and manage Edge Processors, including creating pipelines for data processing (more on this later)!

**EC2 Instance** (with demo data) - An Edge Processor node has been installed for you prior to this exercise in an EC2 instance. The edge processor is configurable via the cloud-based Edge Processor service. Processed data will be forwarded to Splunk Cloud. In the lab environment, this is referred to as 'EP - Edge Node'.

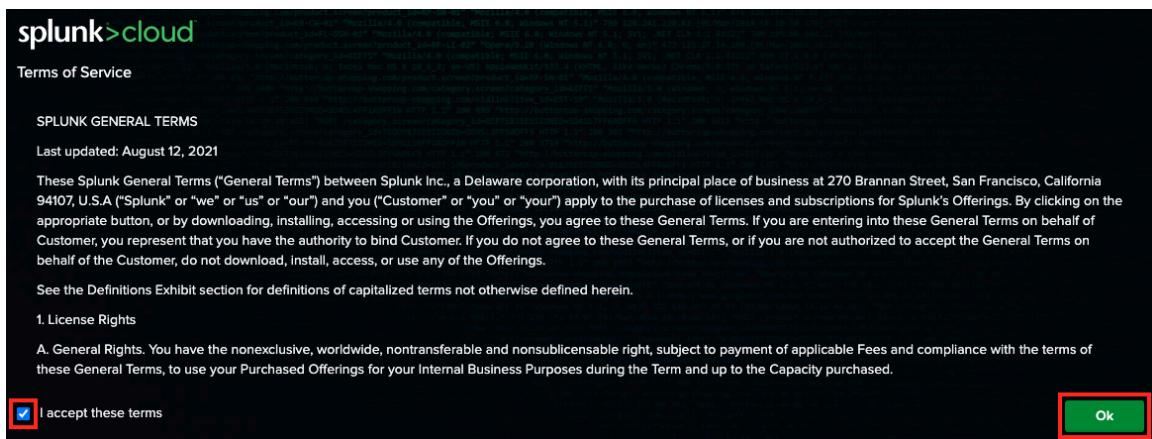


## Access Your Splunk Cloud Instance

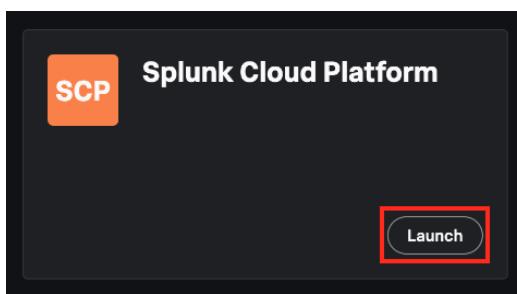
- As part of the exercise, You will be given credentials to the EP that has been set up. Use the admin credentials provided here to log in to your account.



- Accept the Terms of Service and click on **OK**.



- On the welcome page, click on **Launch** under **Splunk Cloud Platform**.



You will now be taken to the data management environment.

## Create a Workspace

4. Enter the shared workspace by clicking on the Shared Workspaces button on the left side.

NAME ↑	KIND ↓	CREATED BY ↓	UPDATED ↓
participant2	Workspace	P2 Participant 2 (me)	a few seconds ago
pipelines	Workspace	P2 Participant 2 (me)	a day ago

5. Create a new workspace for your participant by pressing the **New Workspace** button in the upper right. Name the workspace **participant# (REPLACE # WITH YOUR NUMBER)**. Select the **Create** button. You can skip this step if you have already created this workspace in the past.

**Create workspace**

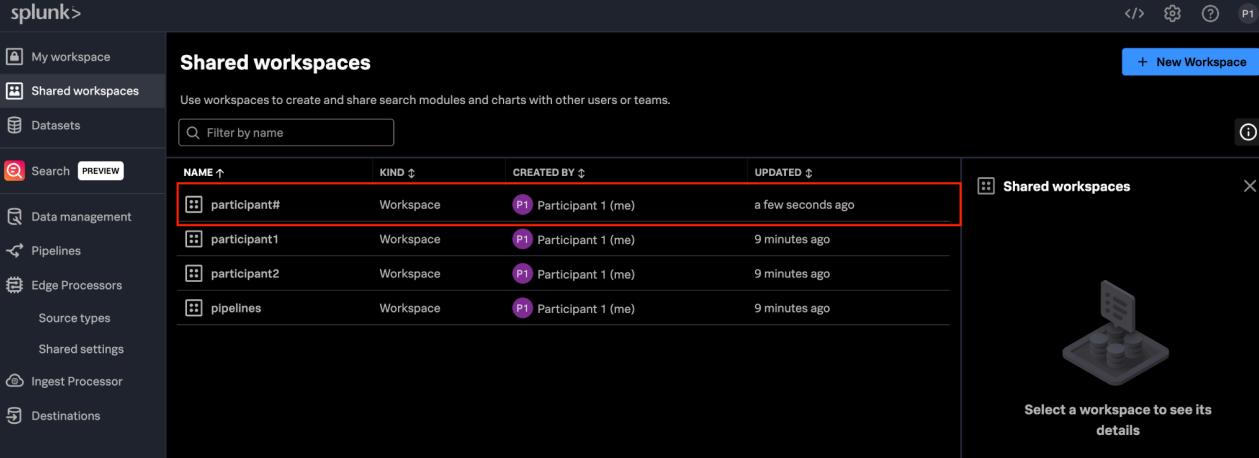
Workspace Name  
participant#  
12/80

Description (optional)  
0/512

Namespace: /shared. participant  
Edit namespace

Cancel **Create**

6. Double-click the workspace you created to enter into the workspace.



The screenshot shows the Splunk interface for managing shared workspaces. On the left, there's a sidebar with various navigation options like My workspace, Shared workspaces, Datasets, Search, and others. The 'Shared workspaces' section is currently selected. The main area is titled 'Shared workspaces' and contains a table of workspaces. The table has columns for NAME, KIND, CREATED BY, and UPDATED. There are four rows:

NAME ↑	KIND ↓	CREATED BY ↓	UPDATED ↓
participant#	Workspace	P1 Participant 1 (me)	a few seconds ago
participant1	Workspace	P1 Participant 1 (me)	9 minutes ago
participant2	Workspace	P1 Participant 1 (me)	9 minutes ago

A modal window titled 'Shared workspaces' is open on the right, showing a preview of a workspace. It includes a small icon of a dashboard with charts, a 'Select a workspace to see its details' message, and a close button.

## Add a new Edge Processor

7. In your home page, from the menu on the left of the page click on **Edge Processors**.

The screenshot shows the Splunk Data Management interface. On the left, there's a sidebar with various options like My workspace, Shared workspaces, Datasets, Search, Data management, Pipelines, Edge Processors (which is highlighted with a red box), Source types, Shared settings, Ingest Processor, Usage summary, Destinations, Education center, and Add-ons. The main content area has a heading 'Get started with the Edge Processor solution' with a sub-instruction about configuring and installing an Edge Processor. It features two buttons: 'Add an Edge Processor' and 'Create Edge Processor pipeline'. Below this, another section titled 'Process your data at ingest time' with a similar sub-instruction and a 'Create Ingest Processor pipeline' button.

8. In the middle of the **Edge Processors** page click on **+ New Edge Processor**.

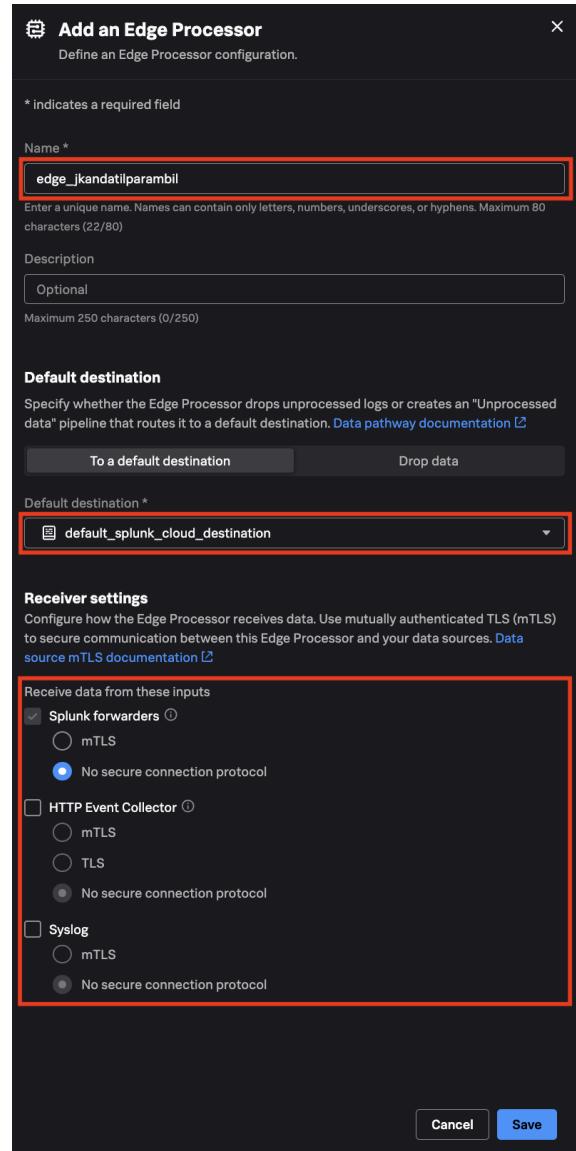
This screenshot shows a page titled 'No Edge Processors to display'. It features a central button labeled '+ New Edge Processor' which is highlighted with a red box. Above the button, there's a sub-instruction: 'Add an Edge Processor to transform and route data'.

9. The **Add an Edge Processor** window will appear. Give your Edge Processor a name. Since each environment is shared among multiple people, use the name '**edge\_**' followed by your own name to help identify this as your Edge Processor. For example: ***edge\_jsmith***

For Default destination select '**default\_splunk\_cloud\_destination**'.

To help simplify our setup for this workshop, disable TLS for Splunk forwarders by checking the **No secure connection protocol option** (this will remove the checks against the options below). Uncheck the HTTP Event Collector and Syslog options.

Click **Save** to create your Edge Processor.



10. You will now see a pane detailing information about your Edge Processor on the right hand side of the page. We also see configuration information about our Edge Processor and can see "**Instances (0)**", meaning that there are currently no Edge Processor instances installed.

**edge\_jkandatilparambil**

ID: 728dcb94-17ef-4b96-8b76-ab00e93088aa  
Number of pipelines: 0  
Default destination: default\_splunk\_cloud\_destination

**Receivers**  
Splunk forwarders: TLS off

**Instances (0)**  
To install an Edge Processor instance, run the following commands.  
Once installed, the software will be automatically updated periodically and will send service-related logs, metrics, and usage data to Splunk Cloud Platform.

11. Under “**Instances**” information, you will see a clipboard containing the installation commands, which we can use to install an Edge Processor instance onto a Linux server.

Copy this command by clicking on the icon.

```

Instances (0)
To install an Edge Processor instance, run the following commands.
Once installed, the software will be automatically updated
periodically and will send service-related logs, metrics, and usage
data to Splunk Cloud Platform.

# Download the installation package for an Edge Pro: 
curl "https://beam.scs.splunk.com/splunk-edge/v0.0.1"

# Validate the package
export SPLUNK_EDGE_PACKAGE_CHECKSUM=$(echo "$(sha512

if [[ "$SPLUNK_EDGE_PACKAGE_CHECKSUM" != "627af01767

then \
# Report if the package is invalid
echo "The installation package is invalid. The downl

else \
# Extract the installation package. This command cre
tar -xvzf splunk-edge.tar.gz

# Create a configuration file that determines which
echo "groupId: ebf4350e-599f-4f3f-bab0-ff315401e37e"
echo "tenant: eps-shw-d8f6d52b19ea31" >> ./splunk-ed
echo "env: staging" >> ./splunk-edge/etc/config.yaml

# Create a token file containing the authentication
echo "eyJhbGciOiJSUzI1NiIsImtpZCI6Inl5VW1ETlBmVTRaR1

# Install the Edge Processor instance
mkdir -p ./splunk-edge/var/log
nohup ./splunk-edge/bin/splunk-edge run >> ./splunk-
fi

```

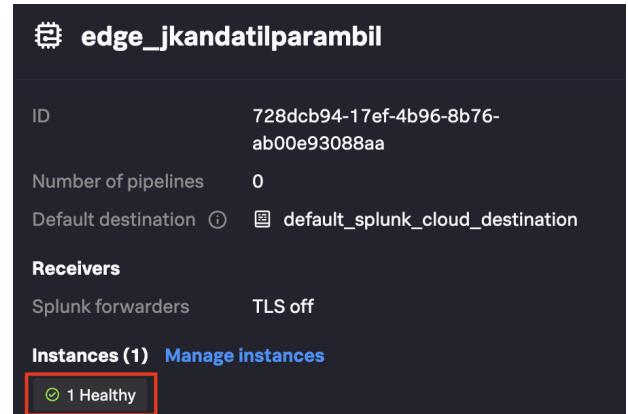
## Install a New Edge Processor Instance

12. With your installation command already copied to your clipboard, open your SSH connection to your EC2 instance (see [Access Your EC2 Instance](#)) and paste the command. The command will run automatically.

Due to the way the command is formatted, you will need to hit enter a second time after the initial command has run. You should now see a green prompt like before.

```
jkandatilparambil@splunk@Domane-Demo-i-0e5cc73c3e9c16b: ~ -- ssh -p 2222 splunk@44.201.96.115 -- 111x34
> tar -xvzf splunk-edge.tar.gz
>
> # Create a configuration file that determines which Splunk Cloud Services (SCS) environment, tenant, and Edge Processor cluster the instance gets associated with
> echo "groupId: c208c0c3-e19a-40cb-9e4a-ffd5f5494b85e" > ./splunk-edge/etc/config.yaml
> echo "tenant: eps-shw-3c23343dca48c3" >> ./splunk-edge/etc/config.yaml
> echo "env: staging" >> ./splunk-edge/etc/config.yaml
>
> # Create a token file containing the authentication token that allows the Edge Processor instance to connect to SCS
> echo "eyJhbGciOiJSUzI1NiIsImpZCI6Inl5VW1ET1BmVTRaR1huRm4rQ3JIZitEWjlldlRDQVlwQXI2dTVGcUt3bGsiLCJ0eXAiOiJKV1Qifo.eyJhdWQiOlsIXBp0i8vc2NzlWRlZmf1bHQ1XSwiY21kJoiMG9h0GcybWtxbndpNG5ka3gzNTcilCJ1Yi6eyJzaWQiOii2OTYwN2N1MT1nWR1M2RkyBmNDkwMDNmMDA5YjRhyisInN1Yi6InNjx2fkbluIn0sImV4cC16MTY5mj2MDE4NsawiwF0IjoxNjkyNzE20Tg1LCjpZHAi0iJlcHNfc2h3XzNjMjMzNDNkY2E0OGMzLmVjLkcC5pZCIsIm1zcyl6Inhh0dHBzOi8vYXV0aC5zdGFnaW5nLnNjcy5zcGx1bmsuY29tiwiianRpIjoiY2ppY3ZLY2trbjBxODRpMws5bdAaiLCJzdWIoiJzY19hZGipbkBAZBzLXNodyoZyIzMzQzGNHNdjmMyIsInR1bmFudC16ImVwcyIzaHctM2MyMzM0M2RjYTQ4YzMiFQ.R-DwmZqPq4efw7UaiijbJTeduCW29hd3Jxvc3d3xy3UCw6sq2Kochw-sekBSo8X8MZrcctcqWEoumAisI80q0LlbjissKN0APWX2t9txj2xp0Dtow0z1tGwpZ1K99wP70ZmYJySNS-1s6HzDjtDs_NEjbStIDnqNUdcg90p73IUKPv6ms0tjzuWgQW7tUroQlpwkunck51feQMk2T-dnLGj1MYZILQ9833uoYDrf1_rkCfbMGeG1l7uI6Jau4_9Vn58JjlFr91DmBcfwCGfpzcB_vgYd1Cv0nXZtA1ifvf8hsbsUaJC8zBKPd1Uaej_W0Fg7EnjJYPociCZU8w" > splunk-edge/var/token
>
> # Install the Edge Processor instance
> mkdir -p ./splunk-edge/var/log
> nohup ./splunk-edge/bin/splunk-edge run >> ./splunk-edge/var/log/install-splunk-edge.out
2>&1 </dev/null &
>
> fi
splunk-edge/
splunk-edge/bin/
splunk-edge/bin/splunk-edge
splunk-edge/var/
splunk-edge/etc/
splunk-edge/etc/splunk-edge.service
[1] 75427
splunk@Domane-Demo-i-0e5cc73c3e9c16b:~$
```

13. After a brief moment (it could take a couple of minutes in today's shared environment) you should see the number of instances associated with your Edge Processor turn to **(1)** with the status of **1 Pending**. This will eventually change to **1 Healthy**. You may need to refresh your browser page to see the status update.



The screenshot shows the Splunk Cloud Services interface for an Edge Processor instance. The instance ID is 728dcb94-17ef-4b96-8b76-ab00e93088aa. It has 0 pipelines and is connected to the default\_splunk\_cloud\_destination. Under Receivers, Splunk forwarders are set to TLS off. The Instances section shows 1 instance, which is currently healthy (indicated by a green circle with a checkmark). A red box highlights the '1 Healthy' status.

ID	728dcb94-17ef-4b96-8b76-ab00e93088aa
Number of pipelines	0
Default destination	default_splunk_cloud_destination
<b>Receivers</b>	
Splunk forwarders	TLS off
<b>Instances (1)</b>	<a href="#">Manage instances</a>
<b>1 Healthy</b>	

If you click on the blue **Manage instances** link you will see a list of instances, including the one you just installed. A green tick means the instance is healthy.

The screenshot shows the 'Manage instances' interface for the edge\_jkandatilparambil app. It displays a single instance named 'Domane-Demo-i-0e5cccec73c3e9c1'. The instance status is marked with a green circle and the text 'Healthy'. Below the instance list are filters for 'Hostname', 'CPU usage', and 'Memory usage'.

## **i** Source types and Destinations

In a real world environment you would also need to configure source types and destinations (i.e. destinations where your data can be sent to) for your Edge Processor but to simplify today's workshop these have been pre-configured for you.

### Restart Your Splunk Heavy Forwarder (Manual Step for Today's Environment)

14. For today's lab you will need to manually restart the Splunk Heavy Forwarder that is running on your EC2 instance. Note that you would not need to do this in a production environment.

Run the following command in the SSH session on your EC2 instance:

```
sudo /opt/splunk/bin/splunk restart
```

Confirm that Splunk restarts successfully:

If you get stuck, we're here to help.  
Look for answers here: <http://docs.splunk.com>

The Splunk web interface is at <https://Domane-Demo-i-03397b9fa7444e973>

```
splunk@Domane-Demo-i-03397b9fa7444e973:~$
```

## Verify Edge Processor is Receiving Data

15. Browse to the Edge Processor page and double-click on your Edge Processor.

The screenshot shows the Splunk interface with the sidebar navigation bar on the left. The 'Edge Processors' option is selected and highlighted with a red box. The main content area is titled 'Edge Processors' and displays a summary of instances: All instances (1), Instances with Error (0), Instances with Warning (0), Instances with Disconnected (0), and Instances with Healthy (1). Below this, a table lists the edge processor named 'edge\_kioskprocessor'. The table has columns for Name, Number of instances, Instance health, and Number of pipelines. The instance 'edge\_kioskprocessor' has 1 instance, is healthy, and has 0 pipelines. A search bar at the top allows filtering by name or instance health.

16. Under the **Data sources** pane on the left side of the page, click on **Received data** and verify that your Edge Processor is receiving inbound **cisco:asa:ep:#** source type data.

The screenshot shows the 'Received data' tab selected in the sidebar. It displays a list of inbound data entries. The first entry is 'audittrail' with 159.06 KB / 367 events. Below it is a list of entries starting with 'cisco:asa:ep:1' up to 'cisco:asa:ep:13', each with its respective size and event count. A search bar and a 'Configure data sources' button are also visible.

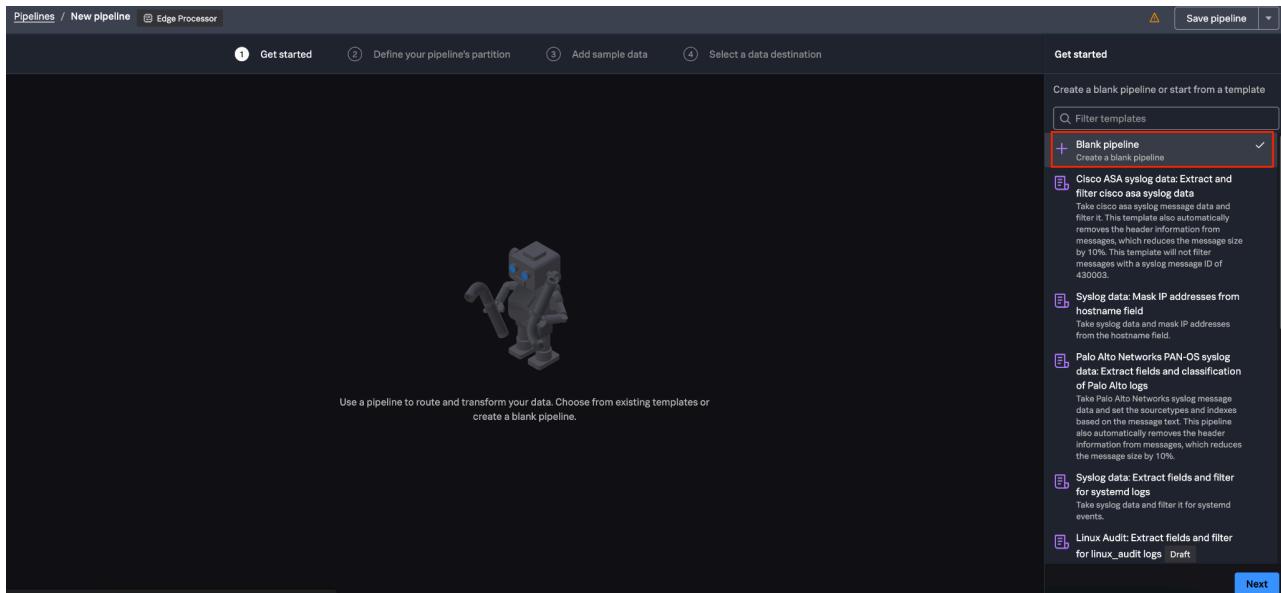
Source type	Inbound data
audittrail	159.06 KB / 367 events
cisco:asa:ep:1	80.32 KB / 449 events
cisco:asa:ep:10	76.46 KB / 428 events
cisco:asa:ep:11	73.47 KB / 412 events
cisco:asa:ep:12	75.9 KB / 424 events
cisco:asa:ep:13	83.33 KB / 466 events

## Create a Pipeline

17. In your participant workspace, create a new EP pipeline by selecting **New → Edge Processor Pipeline** in the upper right.

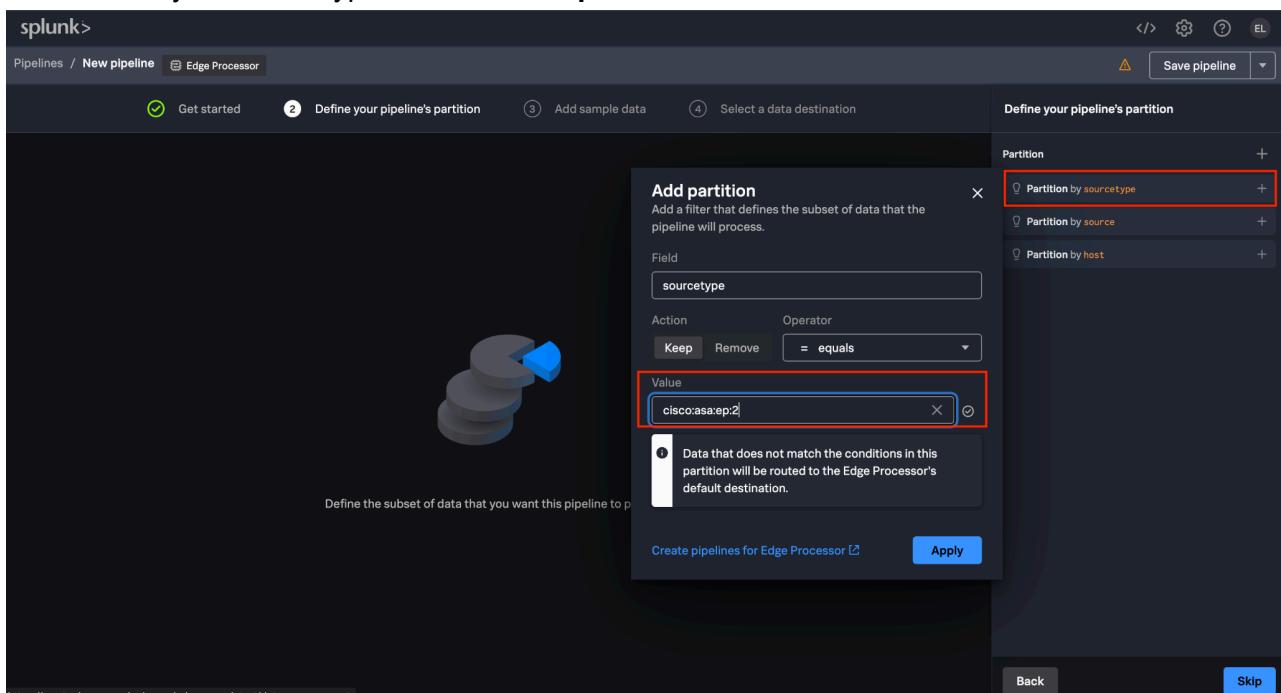
The screenshot shows the 'participant#' workspace. The sidebar includes options like My workspace, Shared workspaces, Datasets, Search (PREVIEW), Data management, Pipelines, Edge Processors, Source types, Shared settings, Ingest Processor, and Destinations. The 'Edge Processors' option is selected. In the top right, there is a '+ New' button with a dropdown menu. The 'Module' option is selected, and 'Edge Processor Pipeline' is highlighted with a red box. Below the workspace title, there is a search bar and a 'Create related content' section with buttons for 'New module', 'New Edge Processor pipeline', and 'New Ingest Processor pipeline'.

18. On the Get Started screen, you can choose from several templates, including Cisco ASA. For the purposes of this workshop, we will select the **Blank pipeline** and choose **Next**.



19. The pipeline's partition allows us to select which sourcetype is being ingested into Edge Processor that we want to process. Select **Partition by sourcetype** and enter the Cisco ASA sourcetype corresponding to your participant number. The format is `cisco:asa:ep:#`. Click **Apply**.

**Reminder:** Your sourcetype should be your participant number. For example, if your participant number is 2, your sourcetype is `cisco:asa:ep:2`



20. Note the partition information now shows the entered sourcetype. Choose **Next**.

The screenshot shows the Splunk Pipeline Editor interface. The top navigation bar includes 'splunk>', 'Pipelines / New pipeline / Edge Processor', and a 'Save pipeline' button. Below the navigation is a progress bar with four steps: 'Get started' (green checkmark), 'Define your pipeline's partition' (yellow exclamation mark), 'Add sample data' (grey), and 'Select a data destination' (grey). The main area displays a pie chart icon and the text 'Define the subset of data that you want this pipeline to process.' On the right, a panel titled 'Define your pipeline's partition' shows a list with one item: 'Partition (1)' followed by 'sourceType equals cisco:asa:ep:2'. The 'Next' button at the bottom right is highlighted with a red box.

21. Choose to **Add sample data**. Download and use the data found in the following sample file:

Direct Link: [cisco\\_logs\\_small.sample](#)

The screenshot shows the Splunk Pipeline Editor interface, specifically the 'Add sample data' step. The top navigation bar includes 'splunk>', 'Shared workspaces / participant / New pipeline / Edge Processor', and a 'Save pipeline' button. Below the navigation is a progress bar with four steps: 'Get started' (green checkmark), 'Define your pipeline's partition' (green checkmark), 'Add sample data' (blue circle with '3'), and 'Select a data destination' (grey). The main area features a robot icon holding a stack of boxes. Below it, the text reads 'Sample data allows you to test your data flow to ensure that your pipeline processes data as desired.' To the right, a panel titled 'Add sample data' contains a large input field with the placeholder 'Enter or upload sample data'. Below the input field is a note '0 / 30,000 characters'. At the bottom of the input field is a red-bordered box containing the text 'Drop your file anywhere or upload file...'. Underneath the input field, there is a section for 'Sample data format' with 'Raw text' and 'CSV' options. The 'Skip' button at the bottom right is highlighted with a red box.

22. After uploading the sample data, press next

splunk>

Shared workspaces / participant / New pipeline Edge Processor

Get started Define your pipeline's partition Add sample data Select a data destination

Add sample data

Enter or upload sample data

Sample data allows you to test your data flow to ensure that your pipeline processes data as desired.

Raw text CSV

Back Next

23. Select a data destination. For this exercise, let's use the pre-configured destination “`default_splunk_cloud_destination`”

splunk>

My workspace / New pipeline Edge Processor

Get started Define your pipeline's partition Add sample data Select a data destination

Select a data destination

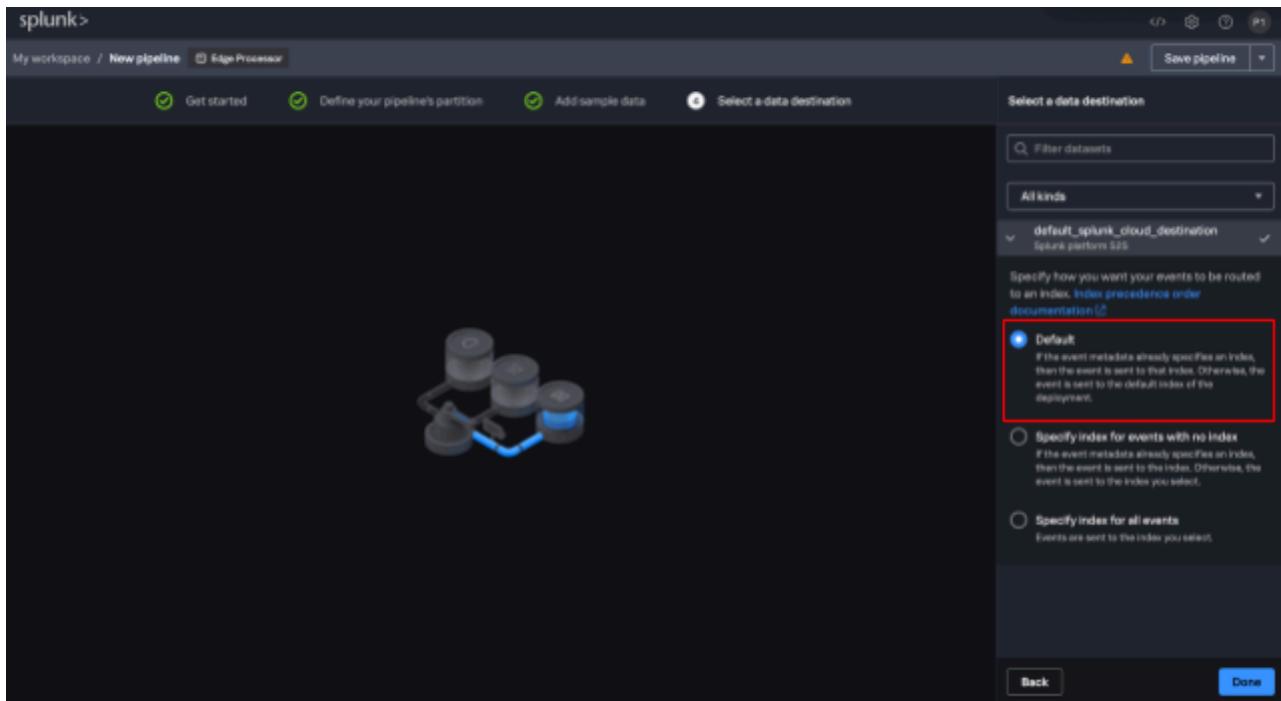
Filter datasets All kinds

default\_spark\_cloud\_destination  
Splunk platform S2S



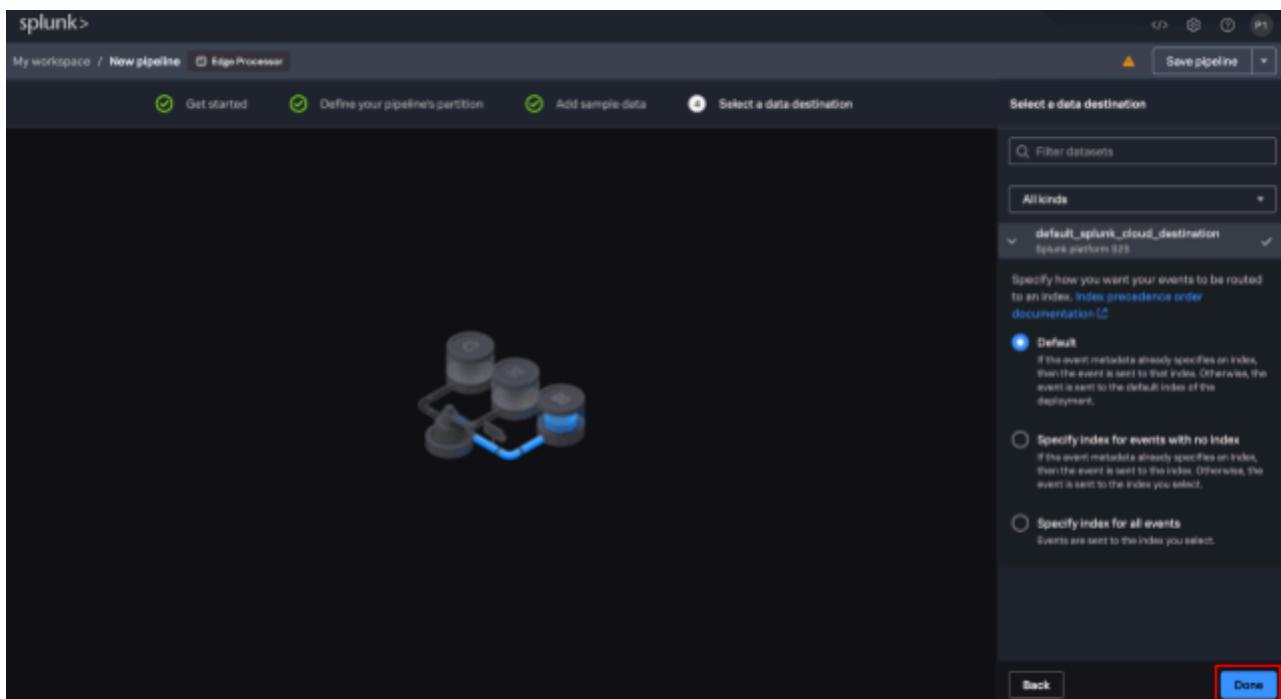
Back Done

24. After selecting **default\_splunk\_cloud\_destination**, press **Next**.



The screenshot shows the Splunk interface for creating a new pipeline. The top navigation bar includes 'My workspace / New pipeline / Edge Processor' and a 'Save pipeline' button. Below the navigation, there are five steps: 'Get started' (green checkmark), 'Define your pipeline's partition' (green checkmark), 'Add sample data' (green checkmark), 'Select a data destination' (blue circle), and 'Done'. The current step is 'Select a data destination'. On the right, a sidebar titled 'Select a data destination' shows a list of datasets: 'All kinds' and 'default\_splunk\_cloud\_destination (splunk platform S2S)'. Below the list, instructions say 'Specify how you want your events to be routed to an index. Index precedence order documentation'. Three radio button options are listed: 'Default' (selected, highlighted with a red box), 'Specify index for events with no index', and 'Specify index for all events'. At the bottom right of the sidebar are 'Back' and 'Done' buttons.

25. For this exercise, we will not select a target index. Press **Done**



This screenshot is identical to the previous one, showing the 'Select a data destination' step. The 'Default' radio button is selected and highlighted with a red box. The 'Done' button at the bottom right of the sidebar is also highlighted with a red box.

26. A screen similar to the one below will appear:

```

splunk>
Shared workspaces / participant / New pipeline Edge Processor
+ Add pipeline : 1 /*
2 A valid SPL2 statement for a pipeline must start with "$pipeline", and include "from $source"
3 and "into $destination".
4 */
5 $pipeline = | from $source | into $destination;

Click Preview ▶ or Command + Enter to preview the pipeline
Use SPL2 to define the conditions for processing and routing your data before sending it
to your data destination. Data is processed by all pipelines for which it meets the
conditions.
Use Edge Processors ↗
< Collapse

```

27. Test your pipeline by first clicking on the \$pipeline SPL. The blue Preview pipeline button ( ) should appear in the top right corner of the screen. When you click it, you should see a preview of the sample events you uploaded earlier.

```

Pipelines / p3pipeline Edge Processor
+ Add pipeline : 5 $pipeline = | from $source | into $destination;
First click on $pipeline

Previewing $pipeline 102 results - 3:36:00 PM - 4:06:37 PM, 22 May 2024
Show only _raw & _time
</> _raw
1 Mar 5 16:01:03 HHHHHHHH Mar 05 2013 16:01:03 HHHHHHHH : %ASA-4-400013 IPS:2003 ICMP redirect from ##IP_ADDRESS## to ##IP_ADDRESS##
## on interface dmz
2 Mar 5 16:01:03 HHHHHHHH Mar 05 2013 16:01:03 HHHHHHHH : %ASA-4-400032 IPS:4051 UDP Snort attack from ##IP_ADDRESS## to ##IP_ADDRESS##
ESS## on interface outside
3 Mar 5 16:01:03 HHHHHHHH Mar 05 2013 16:01:03 HHHHHHHH : %ASA-6-109025: Authorization denied (acl=acmetechinbound) for user 'UUUU
UUUU' from ##IP_ADDRESS##/55424 to ##IP_ADDRESS##/53 on interface Outside using UDP
4 Mar 5 16:01:03 HHHHHHHH Mar 05 2013 16:01:03 HHHHHHHH : %ASA-6-106012: Deny IP from ##IP_ADDRESS## to ##IP_ADDRESS##, IP option
s: "Router Alert"
5 Mar 5 16:01:03 HHHHHHHH Mar 05 2013 16:01:03 HHHHHHHH : %ASA-6-106012: Deny IP from ##IP_ADDRESS## to ##IP_ADDRESS##, IP option
s: "Router Alert"
6 Apr 19 11:24:32 HHHHHHHH Apr 19 2013 11:24:32: %ASA-6-106100: access-list access-epn2-dmz2-in permitted udp epn2-dmz2##IP_ADDRESS##
$##(54877) -> epn2-win##IP_ADDRESS##(53) hit-cnt 1 first hit [0xf0c8454b, 0x0]
7 Apr 19 11:24:32 HHHHHHHH Apr 19 2013 11:24:32: %ASA-4-733100: [ Scanning] drop rate-2 exceeded. Current burst rate is 0 per sec
ond, max configured rate is 8; Current average rate is 16 per second, max configured rate is 4; Cumulative total count is 58913
8 Apr 19 11:24:32 HHHHHHHH Apr 19 2013 11:24:32: %ASA-6-106100: access-list access-epn2-dmz2-in permitted udp epn2-dmz2##IP_ADDRESS##
$##(58736) -> epn2-win##IP_ADDRESS##(53) hit-cnt 1 first hit [0xf0c8454b, 0x0]

< Collapse

```

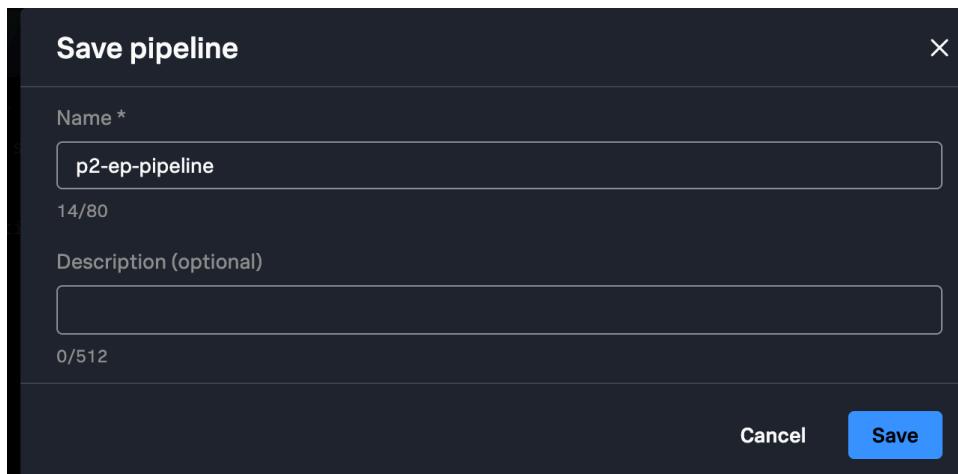
## Save Your Pipeline Progress

28. We can inspect the data in the table to understand the fields and sample values that were captured. To save our progress thus far before we start editing the SPL2, choose the **Save Pipeline** button on the top right.

The screenshot shows the Splunk interface with a pipeline configuration. On the left, there's a sidebar with '+ Add' and 'pipeline'. The main area shows a preview of 102 results from Mar 5, 2013, to Apr 19, 2013. The pipeline configuration on the right includes a 'Sample data' section with 'Inserted Sample', a 'Partition (1)' section with a condition 'sourcetype equals cisco:asa:ep:2', and an 'Actions (1)' section with 'Send data to \$destination' and 'default\_splunk\_cloud\_destination'. The 'Fields (1)' section has a 'Coverage' dropdown. The top right corner features a 'Save pipeline' button, which is highlighted with a red box.

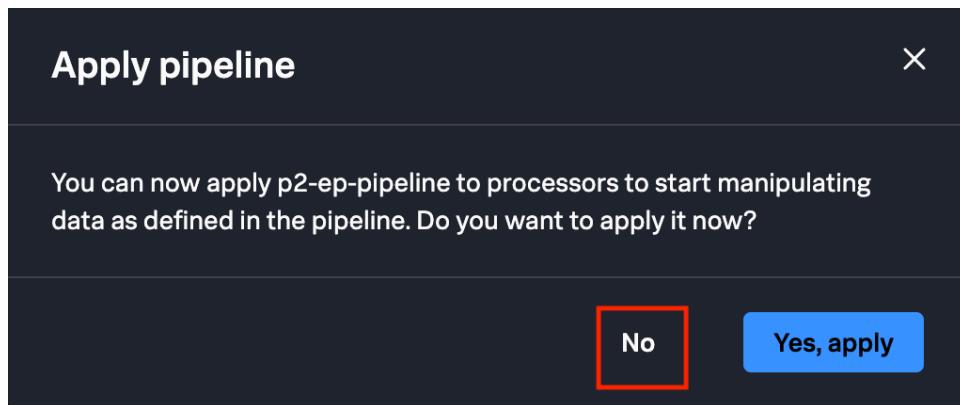
29. In the Save Pipeline modal that appears, name the pipeline **p#-ep-pipeline** **REPLACING # WITH YOUR NUMBER FROM PARTICIPANT NUMBER** and select **Save**.

a. For example, if your **participant number is 2**, name your pipeline **p2-ep-pipeline**



30. After saving, we will be prompted in another window to Apply the pipeline. The Apply step will push out the pipeline configuration and begin transforming the data according to the SPL2 we wrote. Since we haven't written any SPL2 yet, we will select **No**.

Note: If you accidentally selected “Yes”, don’t worry. We will overwrite the pipeline when we apply our Cisco ASA SPL2 later.



## Author Your Pipeline

31. Now that we have our pipeline setup, we can start writing some SPL2! The first task will be to extract useful fields out of `_raw`. To do this, we will create an SPL2 function to extract the fields we care about. Copy/paste the following text into the SPL2 editor, replacing what was there before:

```
function extract_useful_fields($source) {
    return | from $source
    /* Extracted message matches with ASA or FTD */
    | rex field=_raw /(?P<_raw>(%ASA|%FTD).*)/
    /* Extract message number */
    | rex field=_raw /(%ASA|%FTD)-\d+- (?P<message_id>\d+)/
    /* Extract username */
    | rex field=_raw /^[^'\n]*' (?P<username>[^']+)/
}

$pipeline = | from $source
// extract the useful fields
| extract_useful_fields
| into $destination;
```

The `rex` functions inside of `extract_useful_fields` will do the following:

- Ensure that all `_raw` fields start with either ASA or FTD
- Extract the message ID and store it in the `message_id` field
- Extract the username (if present) and store it in the `username` field

Pressing the preview button () again will show that the fields have been extracted. Note that sometimes the field list can get quite long, so we have unchecked several fields on the bottom right to reduce the size of the preview results table. You can also rearrange the columns however you want.

phonelookup  
visualforce.com/apex/simplppr\_\_app?f=0|org...  
pipelines / **pipeline** Edge Processor

+ Add

```
1 function extract_useful_fields($source) {
2     return | from $source
3     /* Extracted message matches with ASA or FTD */
4     | rex field=_raw /(^Pc|raw=(ASA|FTD).+)
5     /* Extract message number */
6     | rex field=_raw /(^ASA|FTD)-d+(?P<message_id>d+)/
7     /* Extract username */
8     | rex field=_raw /(^|\n|^)(?P<username>[^`]+)
9
10
11
12 $Pipeline = | from $source
13     // extract the useful fields
14     | extract_useful_fields
15
16 }
```

Previewing \$Pipeline 100 results - 4:02:00 PM - 4:32:15 PM, 22 May 2024

	# message_id	Aa username
1 ASA-6-109025: Authorization denied (acl=acmetechinbound) for user 'stupefied_matsumoto' from 10.11.36.16/64852 to 10.11.36.617753 on interface Outside using UDP	109025	stupefied_matsumoto
2 ASA-6-109025: Authorization denied (acl=acmetechinbound) for user 'hungry_goldstine' from 10.11.36.21/50136 to 10.11.36.7753 on interface Outside using UDP	109025	hungry_goldstine
3 ASA-6-302814: Teardown TCP connection 1671352851 For outside:130.253.37.97/63455 to inside:10.90.151.105/52311 duration 0:00:00 bytes 935 TCP FIN (competent_hertz@ACMETECH.COM)	302814	NULL
4 ASA-6-109025: Authorization denied (acl=acmetechinbound) for user 'hopeful_heyrovsky' from 10.11.36.47/64845 to 10.11.36.36/53 on interface Outside using UDP	109025	hopeful_heyrovsky
5 ASA-6-302815: Built inbound UDP connection 517992 for outside:200.6.134.23/59488 (10.11.36.18/59488) to Inside:10.11.36.9/59488 (10.11.36.49/53) (wizardly_pasture)	302815	NULL
6 ASA-6-109025: Authorization denied (acl=acmetechinbound) for user 'romantic_lehmann' from 10.11.36.32/51888 to 10.11.36.17/53 on interface Outside using UDP	109025	romantic_lehmann
7 ASA-6-109025: Authorization denied (acl=acmetechinbound) for user 'nifty_lamport' from 10.11.36.29/53030 to 10.11.36.9/53 on interface Outside using UDP	109025	nifty_lamport
8 ASA-6-109025: Authorization denied (acl=acmetechinbound) for user 'compassionate_mendel' from 10.11.36.17/52395 to 10.11.36.4/53 on interface Outside using UDP	109025	compassionate_mendel
9 ASA-6-302816: Teardown UDP connection 215721343 for outside:130.253.37.97/58156(LOCAL/heuristic_hofstadter) to inside:10.159.39.202/53 duration 0:00:00 bytes 535 (sleepy_feistci)	302816	NULL

Show only \_raw & \_time

pipeline

Sample data Inserted Sample

Partition (1)

sourceType equals cisco:asa:ep:2

Actions (2)

extract\_useful\_fields

Send data to \$destination default\_spunk\_cloud\_destination

Filter values

Filter fields

Remove rows where \_raw matches ...

Fields (3)

Filter fields Coverage

0 fields hidden

# message\_id 12 checked

Aa username 60 checked

</> \_raw 1 checked

## Filtering

- Next up, we are going to filter certain message IDs related to security that are noisy. Add the `drop_security_noise` SPL2 function to our editor, then pipe the parsed logs into our new function.

The new function to add in is:

```
function drop_security_noise($source) {
    return | from $source
    | where message_id != "302013"
    | where message_id != "302015"
}
```

Update your previous pipeline SPL to:

```
$pipeline = | from $source
    // extract the useful fields
    | extract_useful_fields
    // Filter "302013", "302015" message ID number
    | drop_security_noise
    | into $destination;
```

- Preview again and notice the 302013 and 302015 message IDs have been filtered.

The screenshot shows the Splunk interface for pipeline management. On the left, the pipeline code is displayed:

```
splunk>
Pipelines / p3pipeline Edge Processor
+ Add
10
11     function drop_security_noise($source) {
12         return | from $source
13         | where message_id != "302013"
14         | where message_id != "302015"
15     }
16
17 $pipeline = | from $source
18     // extract the useful fields
19     | extract_useful_fields
20     // Filter "302013", "302015" message ID number
21     | drop_security_noise
22     | into $destination;
```

On the right, the pipeline configuration details are shown, including a preview of 82 results. The preview table includes columns for \_raw, # message\_id, and Aa username. The results list various log entries, such as XASA-6-109025 Authorization denied and Teardown TCP connection entries, with message IDs like 302014 and 302016 being filtered out.

_raw	# message_id	Aa username
1 XASA-6-109025: Authorization denied (acl=acmetechinbound) for user 'stupefied_matsumoto' from 10.11.36.16/64852 to 10.11.36.17/53 on interface Outside using UDP	109025	stupefied_matsumoto
2 XASA-6-109025: Authorization denied (acl=acmetechinbound) for user 'hungry_goldstine' from 10.11.36.21/50136 to 10.11.36.7/53 on interface Outside using UDP	109025	hungry_goldstine
3 XASA-6-302014: Teardown TCP connection 1671358261 for outside:130.253.37.97/63455 to inside:10.90.151.105/52311 duration 0:00:00 bytes 935 TCP FINs (competent_hertz@ACMETECH.COM)	302014	NULL
4 XASA-6-109025: Authorization denied (acl=acmetechinbound) for user 'hopeful_heyrovsky' from 10.11.36.47/64845 to 10.11.36.36/53 on interface Outside using UDP	109025	hopeful_heyrovsky
5 XASA-6-109025: Authorization denied (acl=acmetechinbound) for user 'romantic_lehmann' from 10.11.36.32/51088 to 10.11.36.1/53 on interface Outside using UDP	109025	romantic_lehmann
6 XASA-6-109025: Authorization denied (acl=acmetechinbound) for user 'nifty_lamport' from 10.11.36.29/53038 to 10.11.36.9/53 on interface Outside using UDP	109025	nifty_lamport
7 XASA-6-109025: Authorization denied (acl=acmetechinbound) for user 'compassionate_mendel' from 10.11.36.17/52395 to 10.11.36.4/53 on interface Outside using UDP	109025	compassionate_mendel
8 XASA-6-302016: Teardown UDP connection 215721343 for outside:130.253.37.97/58156(LOCAL\heuristic_hofstadter) to inside:10.11.36.29/53 duration 0:00:00 bytes 535 (sleepy_feistei)	302016	NULL
9 XASA-6-109025: Authorization denied (acl=acmetechinbound) for user 'pedantic_varahamihira' from 10.11.36.17/59333 to 10.11.36.1/53 on interface Outside using UDP	109025	pedantic_varahamihira

## Masking

- When we send this data to the index, we've decided to mask the usernames to maintain compliance. Here's the mask\_usernames function to replace usernames found in \_raw:

```
function mask_usernames($source) {
    return | from $source
    | eval _raw=if(isnull(username), _raw, replace(_raw, username, "[NAME_REDACTED]"))
}
```

Don't forget to invoke `mask_usernames` in your pipeline:

```
$pipeline = | from $source
    // extract the useful fields
    | extract_useful_fields
    // Filter "302013", "302015" message ID number
    | drop_security_noise
    // Mask usernames to protect PII
    | mask_usernames
    | into $destination;
```

When previewing your results, you will see the username has been redacted in `_raw`. Note that the username is still present in the `username` field.

The screenshot shows the Splunk UI with a pipeline named '\$pipeline' in the 'Pipelines' tab. The pipeline configuration is as follows:

```
+ Add
|> _raw
|> extract_useful_fields
|> drop_security_noise
|> mask_usernames
|> pipeline
|> into $destination;
```

The preview pane shows 82 results. A red box highlights the first few rows of the table, which are as follows:

	# message_id	Aa.username
1 XASA-6-109025: Authorization denied (acl=acmetechinbound) for user [NAME_REDACTED]	from 10.11.36.16/64852 to 10.11.36.1	109025 stupefied_matsumoto
2 XASA-6-109025: Authorization denied (acl=acmetechinbound) for user [NAME_REDACTED]	from 10.11.36.21/50136 to 10.11.36.7/	109025 hungry_goldstine
3 XASA-6-302014: Teardown TCP connection 1671358261 for outside:130.253.37.97/63455 to inside:10.90.151.105/52311 duration 0:00:00 bytes 935 TCP FINs competent_hertz@ACMETECH.COM		302014 NULL
4 XASA-6-109025: Authorization denied (acl=acmetechinbound) for user [NAME_REDACTED]	from 10.11.36.47/64845 to 10.11.36.3	109025 hopeful_heyrovsky
5 XASA-6-109025: Authorization denied (acl=acmetechinbound) for user [NAME_REDACTED]	from 10.11.36.32/51088 to 10.11.36.1/	109025 romantic_lehmann
6 XASA-6-109025: Authorization denied (acl=acmetechinbound) for user [NAME_REDACTED]	from 10.11.36.29/53038 to 10.11.36.9/	109025 nifty_lamport
7 XASA-6-109025: Authorization denied (acl=acmetechinbound) for user [NAME_REDACTED]	from 10.11.36.17/52395 to 10.11.36.4/	109025 compassionate_mendel
8 XASA-6-302016: Teardown UDP connection 215721343 for outside:130.253.37.97/58156(LOCAL\heuristic_hofstadter) to inside:10.159.39.200/53 duration 0:00:00 bytes 535 (sleepy_feistel)		302016 NULL
9 XASA-6-109025: Authorization denied (acl=acmetechinbound) for user [NAME_REDACTED]	from 10.11.36.17/59333 to 10.11.36.1/	109025 pediatric_varahamihira_gerry

## Enrichment Via Lookup

- Next, we can enrich our log events by looking up the message ID with some explanations via a pre-uploaded lookup file.

For lookup import, everyone would have a different path. Your import path is  
**/envs.splunk.'[insert your demo tenant]'.lookups**

Tenant information can be found at your URL. For example, if your URL starts with <https://px.staging.scs.splunk.com/eps-shw-0899d525c30f3f/>, your tenant is **eps-shw-0899d525c30f3f**. Then your import path would be  
**/envs.splunk.'eps-shw-0899d525c30f3f'.lookups**

```
// import statement should be on the top of the file
import 'cisco_msg_id.csv' from /envs.splunk.'eps-shw-0899d525c30f3f'.lookups

function enrich_with_explanation($source) {
    return | from $source |
        lookup 'cisco_msg_id.csv' message_id AS message_id OUTPUT explanation AS explanation
}
```

Then, we can apply this function in our pipeline:

```
$pipeline = | from $source
    // extract the useful fields
    | extract_useful_fields
    // Filter "302013", "302015" message ID number
    | drop_security_noise
    // Mask usernames to protect PII
    | mask_usernames
    // enrich log events with explanations based on message ID
    | enrich_with_explanation
    | into $destination;
```

You will now see a new column with an explanation for each log event:

The screenshot shows the Splunk Pipeline interface. On the left, the pipeline code is displayed:

```

splunk>
Pipelines / p3pipeline Edge Processor
+ Add
28
29     function remove_duplicate_fields($source) {
30         return $source
31     }
32     | $fields->username, message_id, FTD
33
34     $pipeline = | from $source
35     // extract the useful fields
36     | extract_useful_fields
37     // Filter "302013", "302015" message ID number
38     | drop_security_noise
39     // Mask username to protect PII
40     | mask_usernames
41     // enrich with explanation
42     | enrich_with_explanation
43     // Set target index
44     | eval index="conf24-ep-lab"
45     | into $destination;

```

On the right, the pipeline configuration is shown:

- pipeline**: sourcetype equals cisco:asa:ep:2
- Actions (6)**:
  - extract\_useful\_fields
  - drop\_security\_noise
  - mask\_usernames
  - enrich\_with\_explanation
  - Target index conf24-ep-lab
  - Send data to \$destination default\_splunk\_cloud\_destination
- Fields (5)**:
  - Filter fields
  - Coverage
  - 0 fields hidden
  - \_raw
  - explanation
  - index
  - message\_id
  - username

Below the pipeline code, a preview of 82 results is shown. A red box highlights the 'explanation' column. The table has columns: \_raw, explanation, index, message\_id, and username.

_raw	explanation	index	message_id	username
1 %ASA-6-109025: Authorization denied (acl=acmetechinbound) for user '[N AME_REDACTED]' from 10.11.36.16/64852 to 10.11.36.17/53 on interface [N AME_REDACTED]	The access control list check failed. The check either...	conf24-ep-lab	109025	stupefied_matsumoto
2 %ASA-6-109025: Authorization denied (acl=acmetechinbound) for user '[N AME_REDACTED]' from 10.11.36.21/58136 to 10.11.36.7/53 on interface [N AME_REDACTED]	The access control list check failed. The check either...	conf24-ep-lab	109025	hungry_goldstine
3 %ASA-6-302014: Teardown TCP connection 1671358261 for outside:138.253.37.97/63455 to inside:10.98.151.165/52371 duration 0:00:00 bytes 395	A TCP connection between two hosts was deleted.	conf24-ep-lab	302014	NULL
4 %ASA-6-109025: Authorization denied (acl=acmetechinbound) for user '[N AME_REDACTED]' from 10.11.36.47/64845 to 10.11.36.36/53 on interface [N AME_REDACTED]	The access control list check failed. The check either...	conf24-ep-lab	109025	hopeful_heyrovsky
5 %ASA-6-109025: Authorization denied (acl=acmetechinbound) for user '[N AME_REDACTED]' from 10.11.36.32/51888 to 10.11.36.1/53 on interface [N AME_REDACTED]	The access control list check failed. The check either...	conf24-ep-lab	109025	romantic_lemann
6 %ASA-6-109025: Authorization denied (acl=acmetechinbound) for user '[N AME_REDACTED]' from 10.11.36.29/53838 to 10.11.36.9/53 on interface [N AME_REDACTED]	The access control list check failed. The check either...	conf24-ep-lab	109025	nifty_lamport
7 %ASA-6-109025: Authorization denied (acl=acmetechinbound) for user '[N AME_REDACTED]' from 10.11.36.17/52305 to 10.11.36.4/53 on interface [N AME_REDACTED]	The access control list check failed. The check either...	conf24-ep-lab	109025	compassionate_mendel

## Drop Fields & Set Index

- Finally, we can remove the parsed fields as they are no longer needed, and we don't want them sent to the indexer. Create the remove\_duplicate\_fields function:

```
function remove_duplicate_fields($source) {  
    return | from $source  
    | fields -username, message_id, FTD  
}
```

Add the remove\_duplicate\_fields function to your pipeline definition and set the index to dmx-ep-lab.

```
$pipeline = | from $source  
// extract the useful fields  
| extract_useful_fields  
// Filter "302013", "302015" message ID number  
| drop_security_noise  
// Mask usernames to protect PII  
| mask_usernames  
// enrich log events with explanations based on message ID  
| enrich_with_explanation  
// Remove duplicate fields  
| remove_duplicate_fields  
// Set target index  
| eval index="dmx-ep-lab"  
| into $destination;
```

Preview now shows the results without these fields along with the index field set:

Splunk > Pipelines / p3pipeline Edge Processor

```
+ Add
fo enrich_with_expl...
fo extract_useful_fi...
fo drop_security_no...
fo mask_usernames
fo remove_duplicate...
<- pipeline ...
  29 | return | from source
  30 |   | fields -username, message_id, FTD
  31 }
  32 }
  33 | drop_security_noise <source>
  34 spipe Undocumented function
  35 // Parameters
  36 | $source -Undocumented parameter| _ ID number
  37 | drop_security_noise
  38 // Mask usernames to protect PII
  39 | mask_usernames
  40 // enrich log events with explanation based on message ID
  41 | enrich_with_explanation
  42 | enrich_with_explanation
  43 // Remove duplicate fields
  44 | remove_duplicate_fields
  45 // Set target index
  46 | eval index="conf24-ep-lab"
  47 | into $destination;
```

Sample Preview "pipeline" (Command + Enter)

Partition(1)

- sourceType equals cisco:asa-ep-2

Actions (7)

- extract\_useful\_fields
- drop\_security\_noise
- mask\_usernames
- enrich\_with\_explanation
- remove\_duplicate\_fields
- Target Index conf24-ep-lab
- Send data to destination default\_splunk\_cloud\_destination

Fields (3)

Show only \_raw & \_time

Previewing Pipeline 82 results > 4:17:00 PM - 4:47:39 PM, 22 May 2024 ▲

	Aa explanation	Aa index
1 NASA-6-108925: Authorization denied (acl=acmetechinbound) for user '[NAME_REDACTED]' from 10.11.36.16/64852 to 10.11.36.17/53 on interface Outside using UDP	The access control list che conf24-ep-lab ck failed. The check eithe..	conf24-ep-lab
2 NASA-6-109025: Authorization denied (acl=acmetechinbound) for user '[NAME_REDACTED]' from 10.11.36.21/50136 to 10.11.36.7/53 on interface Outside using UDP	The access control list che conf24-ep-lab ck failed. The check eithe..	conf24-ep-lab
3 NASA-6-109024: Teardown TCP connection 1671358261 for outside:138.253.37.97/63455 to inside:10.90.151.105/52311 duration 0:00:00 bytes 935 TCP FINs (competent_hertz@ACMETECH.COM)	A TCP connection between tw conf24-ep-lab o hosts was deleted.	conf24-ep-lab
4 NASA-6-109025: Authorization denied (acl=acmetechinbound) for user '[NAME_REDACTED]' from 10.11.36.47/64845 to 10.11.36.36/53 on interface Outside using UDP	The access control list che conf24-ep-lab ck failed. The check eithe..	conf24-ep-lab
5 NASA-6-109025: Authorization denied (acl=acmetechinbound) for user '[NAME_REDACTED]' from 10.11.36.32/51088 to 10.11.36.1/53 on interface Outside using UDP	The access control list che conf24-ep-lab ck failed. The check eithe..	conf24-ep-lab
6 NASA-6-109025: Authorization denied (acl=acmetechinbound) for user '[NAME_REDACTED]' from 10.11.36.29/50300 to 10.11.36.9/53 on interface Outside using UDP	The access control list che conf24-ep-lab ck failed. The check eithe..	conf24-ep-lab
7 NASA-6-109025: Authorization denied (acl=acmetechinbound) for user '[NAME_REDACTED]' from 10.11.36.17/52395 to 10.11.36.4/53 on interface Outside using UDP	The access control list che conf24-ep-lab ck failed. The check eithe..	conf24-ep-lab

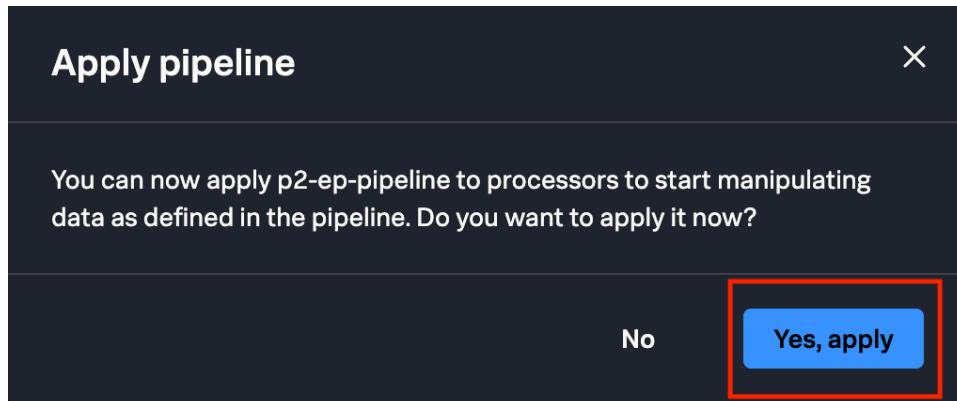
Collapse

## Apply the Pipeline to your Edge Processor

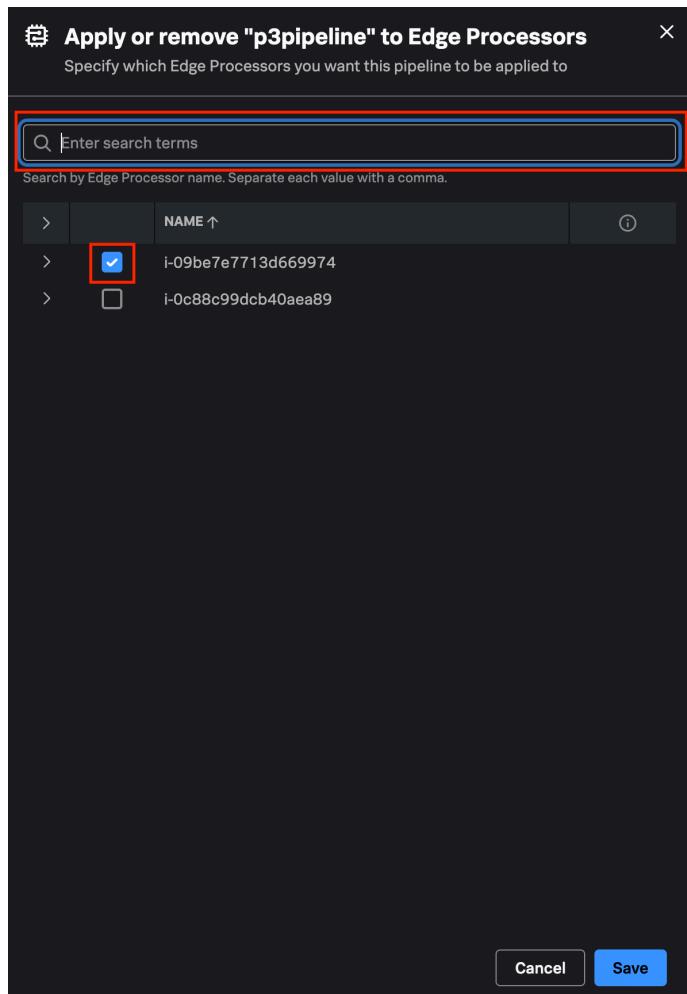
- Being satisfied with the results of our pipeline, we will now apply the pipeline. As we did previously, select the **Save pipeline** button on the top right.

The screenshot shows the Splunk pipeline editor interface. On the left, there's a code editor window displaying a pipeline configuration. On the right, there are several panels: a preview panel showing 82 results, a sidebar with pipeline parameters, and a main configuration panel with sections for 'Actions', 'Target\_index', and 'Fields'. A red box highlights the 'Save pipeline' button in the top right corner of the main configuration area.

If prompted, select **Yes, apply** to apply the pipeline. If you are not prompted, the pipeline will be automatically applied on save.



Search and select the edge processor instance that is assigned to you. Then press **Save**



## Verify Edge Processor is Receiving Data

- After applying your pipeline successfully, click the Splunk logo on the top left to navigate back to the home page. Then click **Edge Processors** on the left panel

The screenshot shows the Splunk Data Management interface. On the left sidebar, under the "Data management" section, the "Edge Processors" option is highlighted with a red box. The main content area is titled "Process your data at the edge" and includes a call-to-action button "Get started with Edge Processors". Below this, there's a "What's new" section with four cards:

- Edge Processor acknowledgement for HTTP Event Collector (HEC)**: May 30, 2024. Edge Processor. Verify whether the Edge Processor received data that was sent through HEC.
- HTTP Event Collector (HEC) token authentication**: April 2, 2024. Edge Processor. Secure the HEC receiver in your Edge Processor by using token authentication.
- User interface support for hashing functions**: Mar 26, 2024. Edge Processor, Ingest Processor. Use the "Compute hash of" pipeline action to leverage a variety of hashing functions to obfuscate sensitive data.
- User interface support for lookup command**: Mar 12, 2024. Edge Processor. Use the "Enrich events with lookup" pipeline action to configure the lookup command.

The URL shown in the browser is <https://nx.staging.scs.splunk.com/ens-shw-0899d525c30f3f/workspaces/shared>.

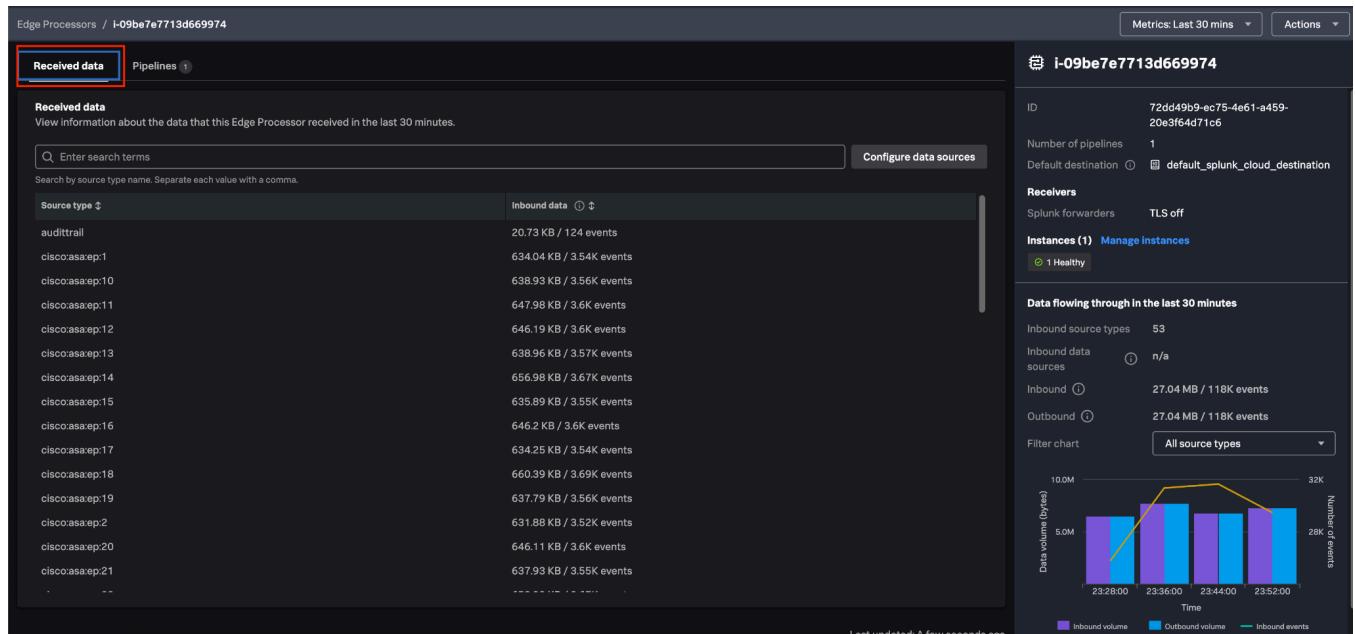
- Search for the instance that is assigned to you and double-click on it

The screenshot shows the Splunk Edge Processors list interface. On the left sidebar, the "Edge Processors" option is highlighted with a red box. The main content area is titled "Edge Processors" and displays summary statistics: 2 instances total, 0 with errors, 0 with warnings, 0 disconnected, and 2 healthy. Below this is a search bar with a placeholder "Q. Filter by name" and a dropdown menu "Filter by instance health". A table lists two instances:

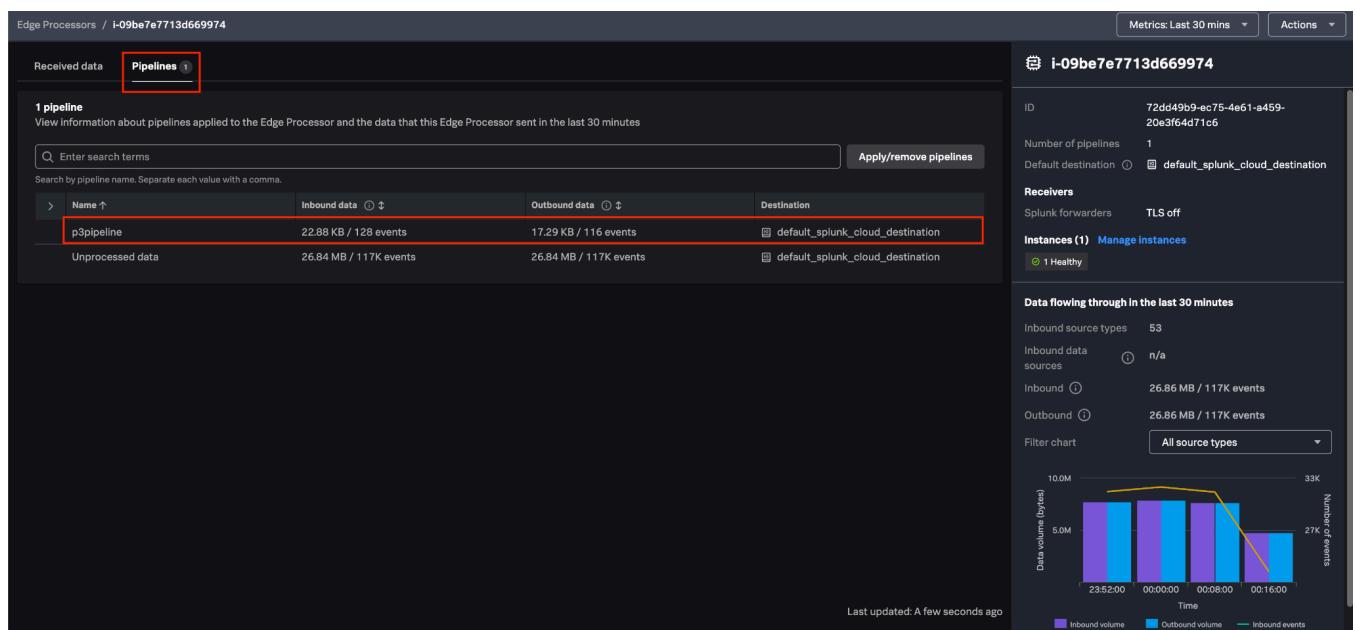
Name ↑	Number of Instances ↓	Instance health	Number of pipelines ↓
i-09be7e7713d669974	1	<div style="width: 100%; background-color: green;"></div>	1
i-0c98c99dc40aea89	1	<div style="width: 100%; background-color: green;"></div>	0

- Under the **Data sources** pane on the left side of the page, click on **Received data** and verify that your Edge Processor is receiving inbound **cisco:asa:ep:#** sourcetype data. You will notice multiple sourcetypes with a different suffix number. This is to allow each participant to experiment with their data without interfering with other participants.

**Reminder:** if your **participant number** is **2**, please ensure your pipeline is set up to process data with sourcetype **cisco:asa:ep:2** in step 9.



- In the pipelines panel, you should be able to see the pipeline you just applied to your Edge Processor.



## Check Your Data in Splunk Cloud

- Log in to Splunk Cloud and open up the **Search & Reporting** app. Run the following search over the **last 15 minutes** and verify that you now see the redacted events:

**Reminder:** Your sourcetype should be your participant number. For example, if your participant number is 2, your sourcetype is **cisco:asa:ep:2**

```
index=dmx-ep-lab sourcetype=cisco:asa:ep:#
```

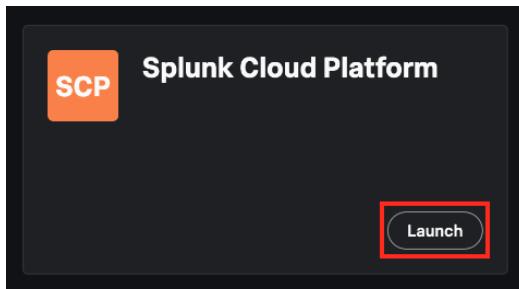
## Lab Exercise 3: Ingest Processor

### Description

In this lab, you will filter, mask, and route Cisco ASA logs using Ingest Processor. The transformation pipelines you write will execute immediately prior to indexing. You will implement the [Base Use Case](#) described at the beginning of this guide. You will also implement a basic Logs To Metrics conversion and send metrics to [Splunk Observability](https://www.splunk.com/en_us/products/observability.html) ([https://www.splunk.com/en\\_us/products/observability.html](https://www.splunk.com/en_us/products/observability.html)) (aka O11y).

### Access Your Ingest Processor Console

1. Log into the Splunk Cloud Ingest Processor UI. Refer to [Lab Connection Info](#) to find your login URL and credentials.
2. On the welcome page, click on **Launch** under **Splunk Cloud Platform**.



You will now be taken to the Ingest Processor management environment.

## Create a Workspace

*Note: You can skip this step if you have already created this workspace in the past.*

- Enter the shared workspace by clicking on the Shared Workspaces button on the left side.

NAME ↑	KIND ↓	CREATED BY ↓	UPDATED ↓
participant2	Workspace	Participant 2 (me)	a few seconds ago
pipelines	Workspace	Participant 2 (me)	a day ago

- Create a new workspace for your participant. Name the workspace **participant# (REPLACE # WITH YOUR PARTICIPANT NUMBER)**. Select the **Create** button.

Workspace Name  
participant2  
12/80

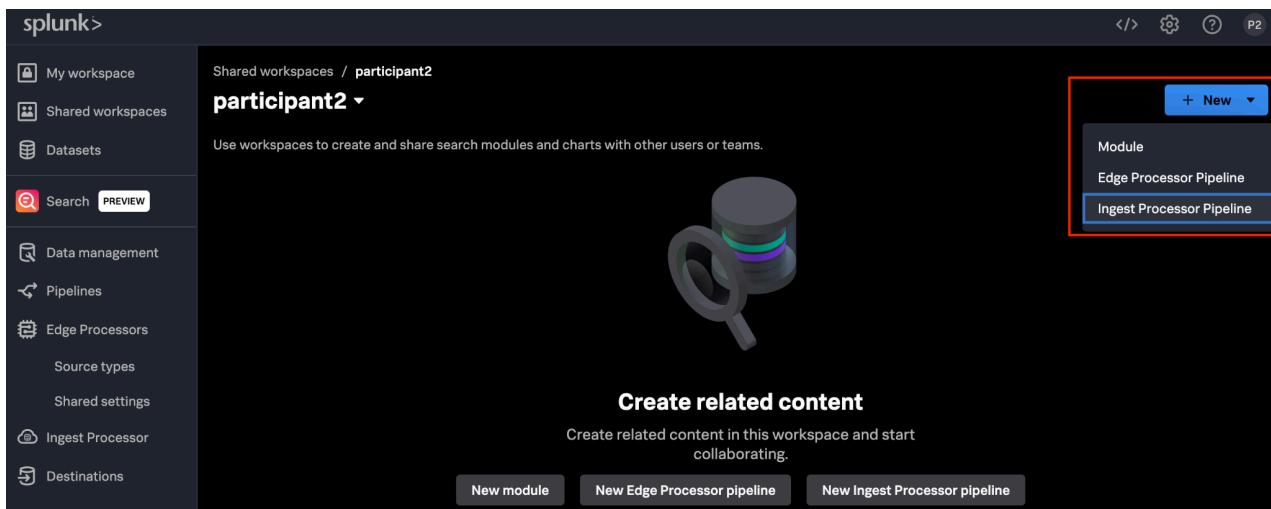
Description (optional)  
0/512

Namespace: /shared. participant2  
Edit namespace

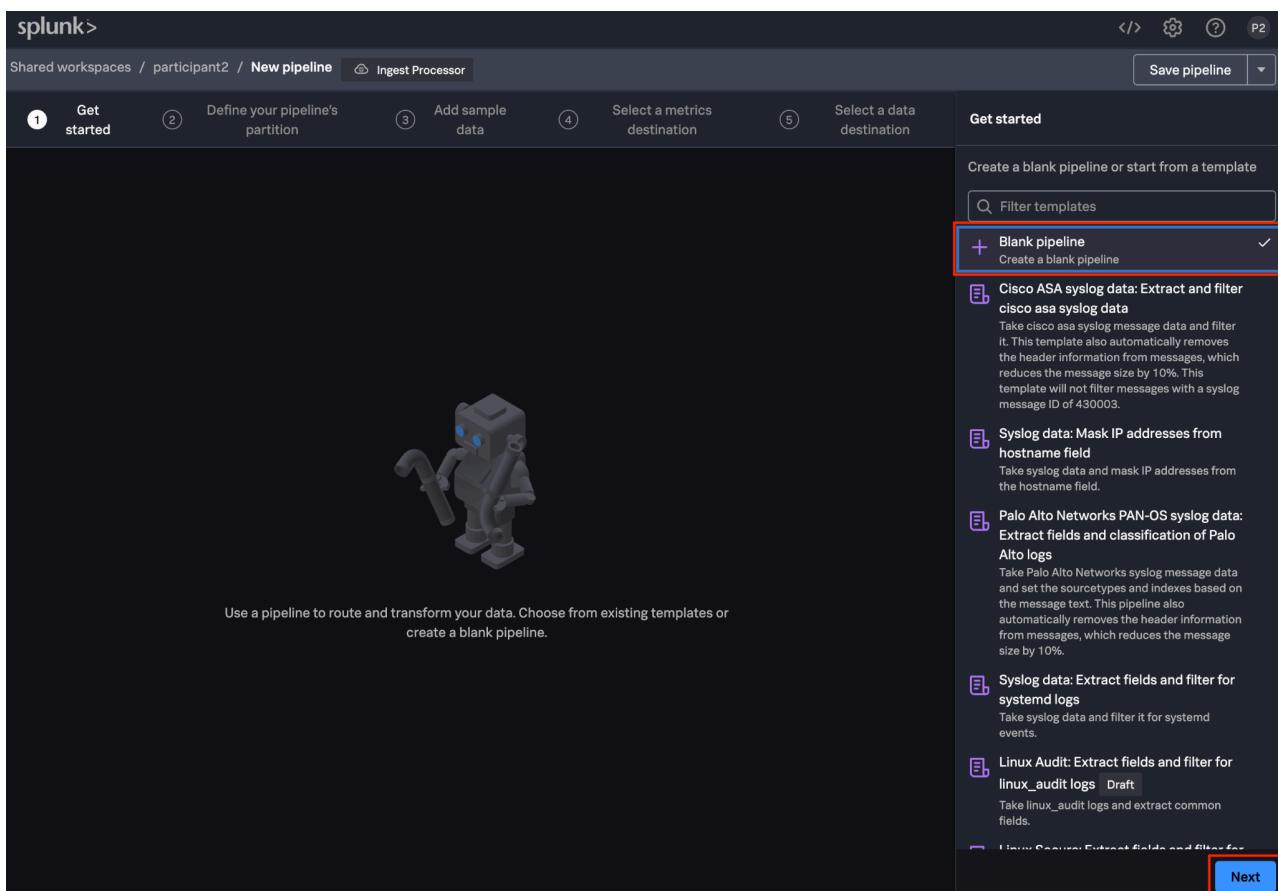
Cancel Create

## Create a Pipeline

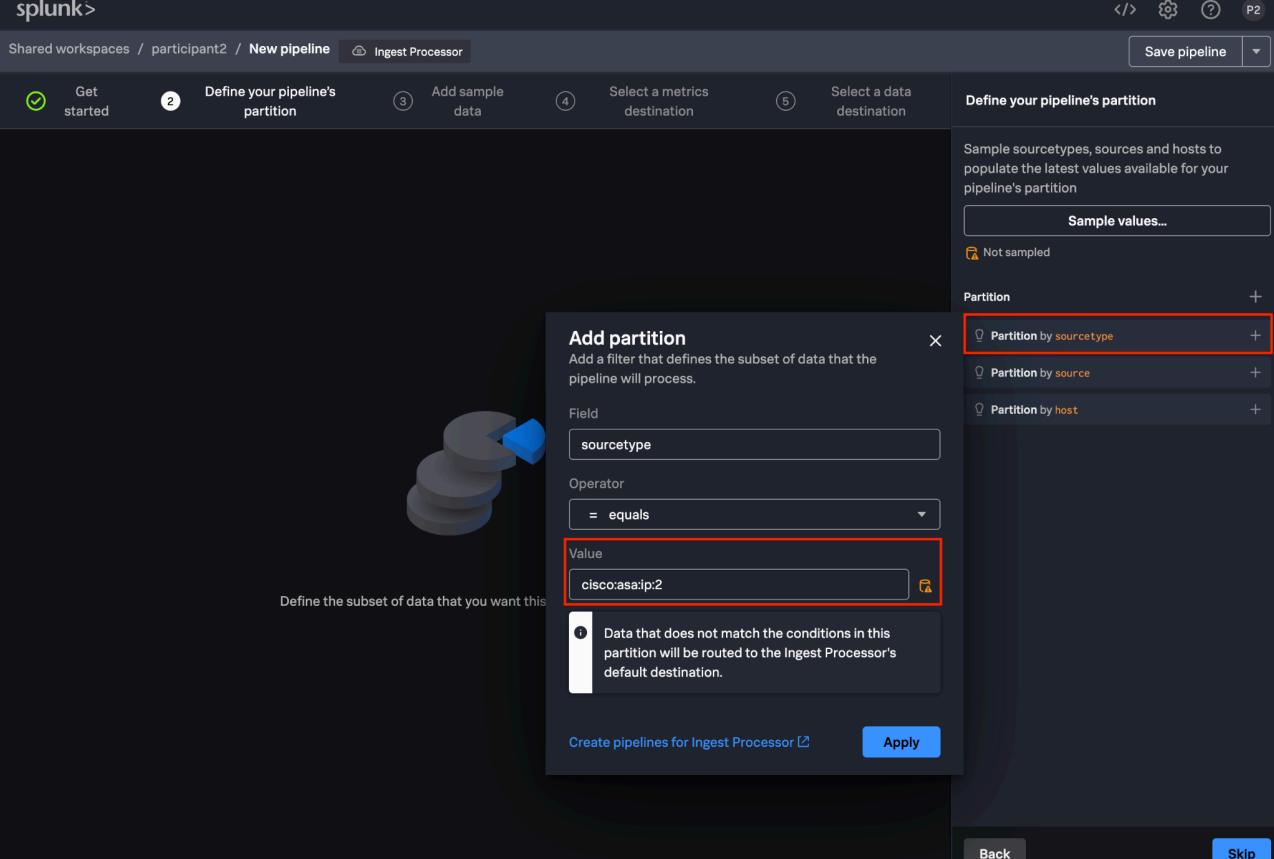
5. In your participant workspace, create a new IP pipeline by selecting **New → Ingest Processor Pipeline** in the upper right.



6. On the Get Started screen, you can select from several templates, including Cisco ASA. For the purposes of this workshop, we will select the **Blank Pipeline** and choose **Next**.



7. The pipeline's partition allows us to select which sourcetype is being ingested into Splunk Cloud that we want to process prior to indexing. Select **Partition by sourcetype** and enter the Cisco ASA sourcetype corresponding to your participant number. The format is `cisco:asa:ip:#`.  
**REPLACE # WITH YOUR PARTICIPANT NUMBER.** Click **Apply**.



The screenshot shows the Splunk interface for defining a pipeline's partition. The main header says "splunk> Shared workspaces / participant2 / New pipeline / Ingest Processor". Below it, a progress bar indicates steps 1 through 5. Step 2, "Define your pipeline's partition", is currently active. A modal window titled "Add partition" is open, showing a configuration for filtering data based on the "sourcetype" field. The "Value" field is set to "cisco:asa:ip:2". A note below the modal states: "Data that does not match the conditions in this partition will be routed to the Ingest Processor's default destination." At the bottom of the modal are "Create pipelines for Ingest Processor" and "Apply" buttons. To the right of the modal, the "Partition" section is visible, with "Partition by sourcetype" selected. There are other options like "Partition by source" and "Partition by host". Buttons for "Back" and "Skip" are at the bottom right of the main screen.

8. Note the partition information now shows the entered sourcetype. Choose **Next**.

splunk>

Shared workspaces / participant2 / New pipeline Ingest Processor

Get started Define your pipeline's partition Add sample data Select a metrics destination Select a data destination Define your pipeline's partition

Sample sourcetypes, sources and hosts to populate the latest values available for your pipeline's partition

Sample values...

Not sampled

Partition (1)

sourcetype equals cisco:asa:ip:2

Back Next

9. Choose to add sample data **From snapshot**.

splunk>

Pipelines / New pipeline Ingest Processor

Get started Define your pipeline's partition Add sample data Select a metrics destination Select a data destination Add sample data

Filter snapshots

Paste or upload sample data

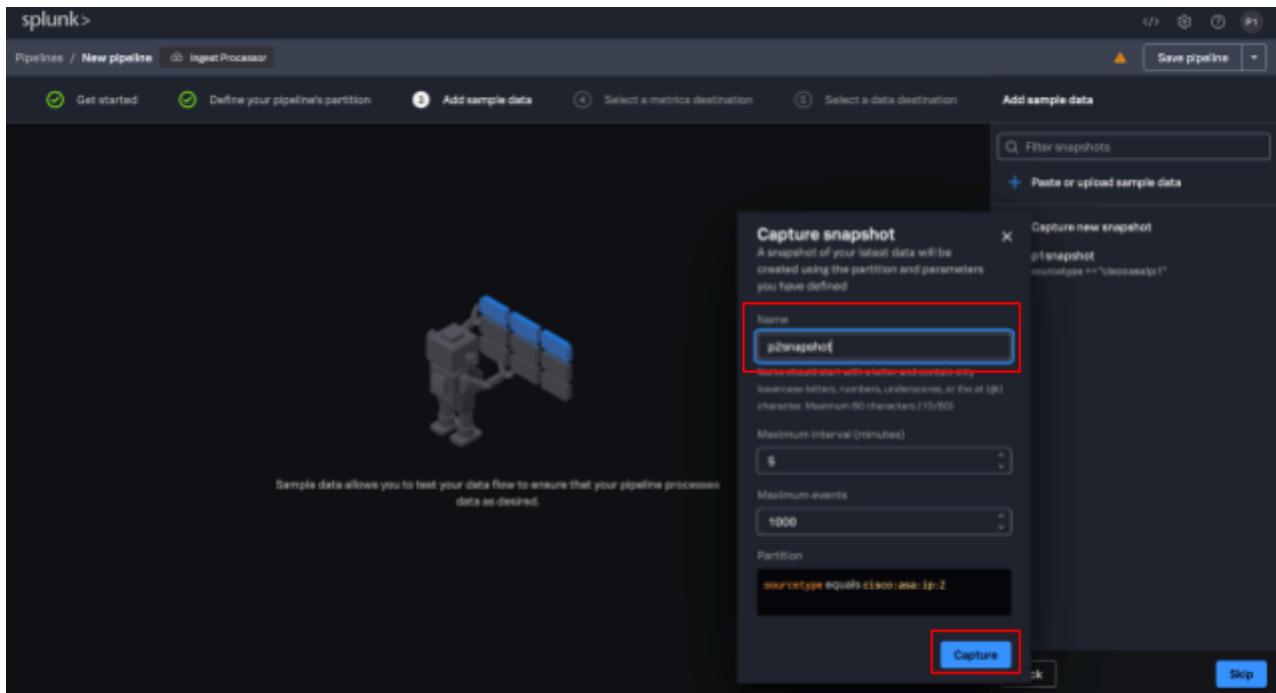
Capture new snapshot

a snapshot sourcetype == 'filebeat-ingest'

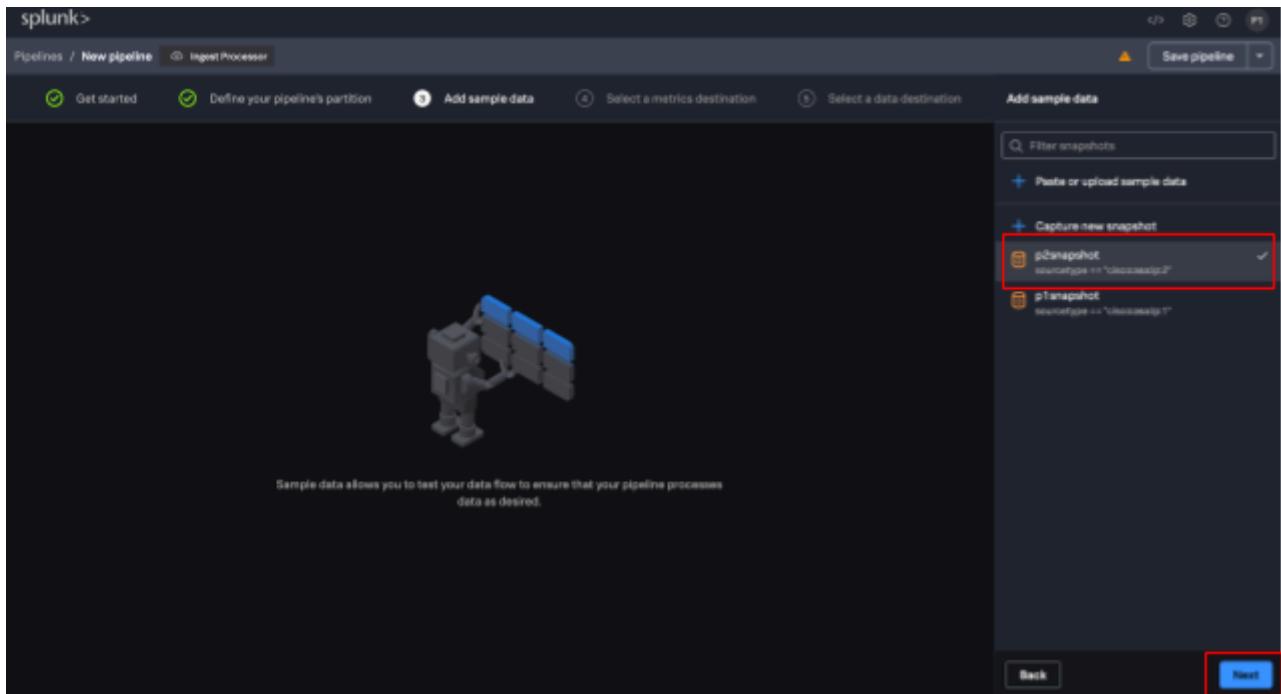
Back Skip

## Create a Preview Snapshot

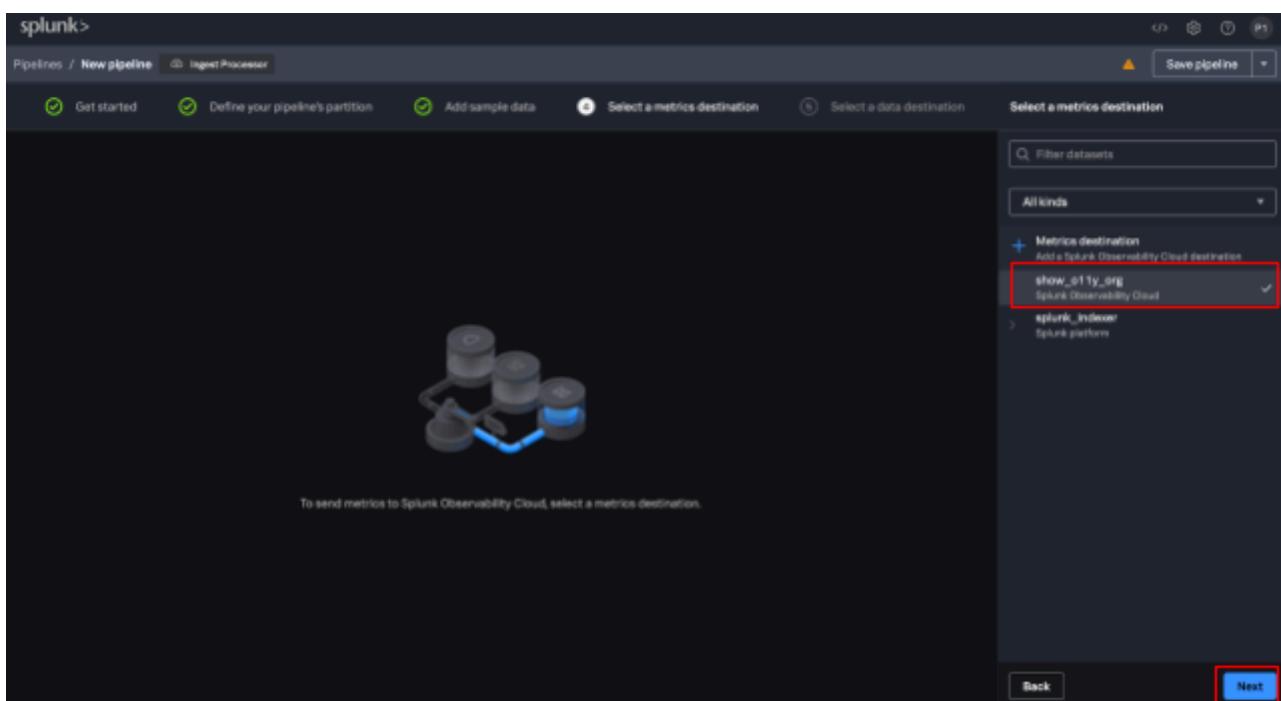
10. We will create a new snapshot from the data that is being ingested at this moment. Choose **Create new snapshot**, enter the snapshot name (such as `p#snapshot REPLACING # WITH YOUR PARTICIPANT NUMBER`), and then initiate the snapshot process by selecting **Capture**. This will begin capturing up to 1000 events over the next 5 minutes. Note that our pipeline sourcetype is automatically populated.



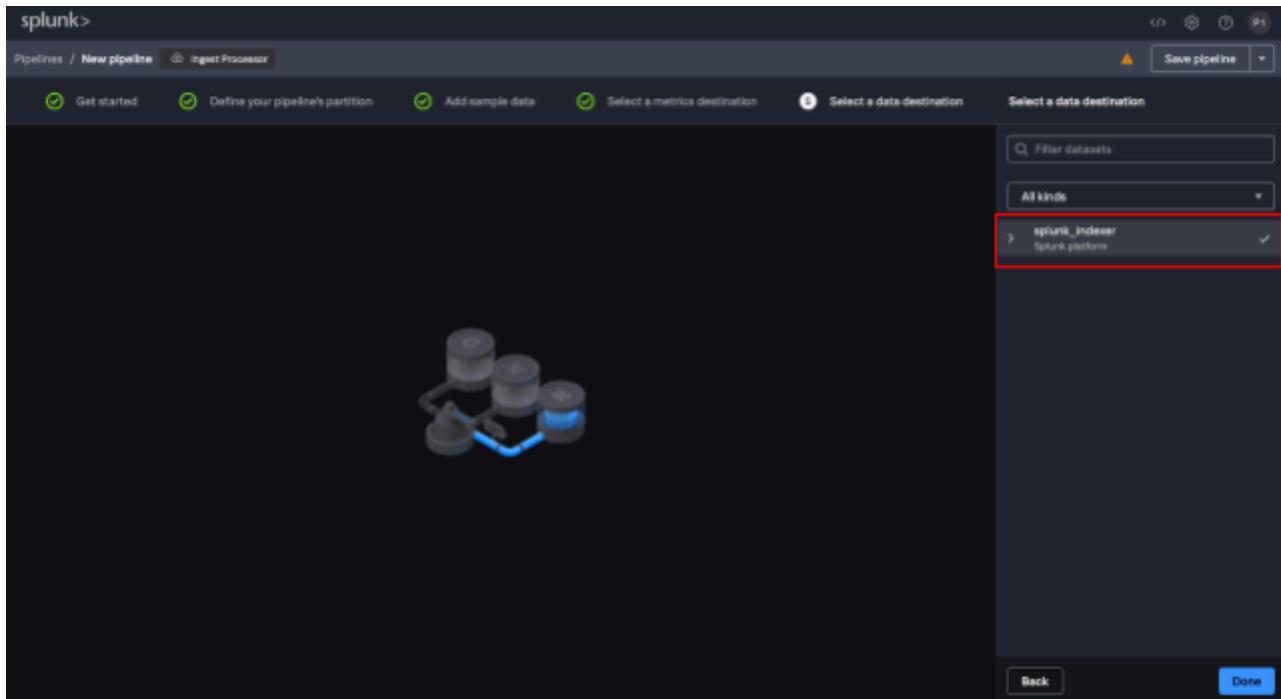
11. Select the newly created snapshot and choose **Next**.



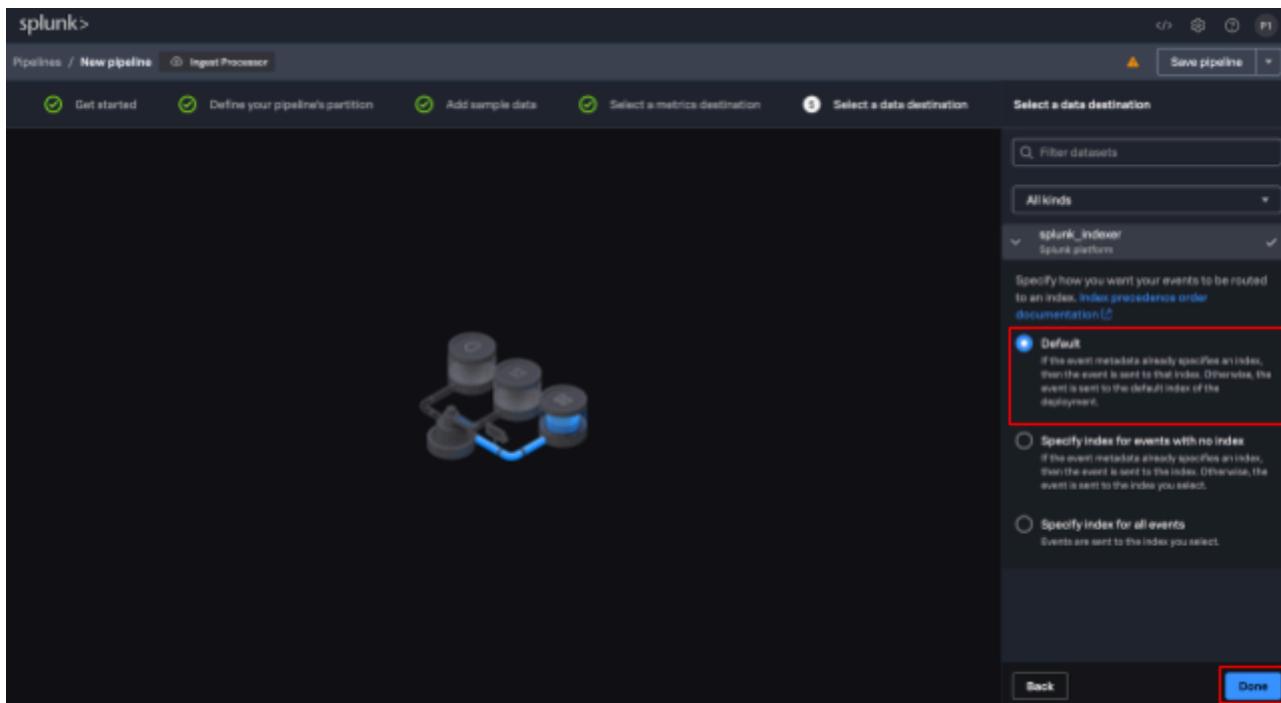
12. In this step, you are prompted for the default metrics destination. Choose the Splunk Show Splunk Observability destination that was automatically created for you, **show\_o11y\_org**. Then click **Next**.



13. In this step, you are prompted to go to the default logs destination. Choose the Splunk indexer destination that was automatically created for you called **splunk\_indexer**. Then click **Next**.



14. Leave the target index blank and select **Done**.



15. At this point, you will see the code editor experience, where you can write, edit, and test SPL2. This page should show a basic passthrough pipeline. Information about the pipeline is available on the right-hand side panel. The **Actions** section provides a UI-driven code editing experience with support for tasks like masking, filtering, selecting destinations, etc. To see the data that we captured in the snapshot in the previous steps, select the **play button** on the upper right.

The screenshot shows the Splunk code editor interface for creating a new pipeline. The left sidebar has a '+ Add' button and a 'pipeline' section. The main area contains the following SPL2 code:

```
/*
A valid SPL2 statement for a pipeline must start with "$pipeline", and include "from $source"
and "into $destination".
*/
$pipeline = | from $source | into $destination;
```

The right-hand side panel displays the pipeline configuration:

- pipeline**:
  - Sample data p2snapshot
- Partition (1)**:
  - sourcetype equals cisco:asa:ip:2
- Actions (1)**:
  - Send data to \$destination splunk\_indexer
  - Remove rows where \_raw matches ...
  - Set field to ...
  - Compute hash of ...
- Fields**: (empty)

At the bottom, there are instructions to preview the pipeline and links to use ingest processors.

16. The snapshot data that has been captured so far is shown in the preview table below.

The screenshot shows the Splunk interface with a pipeline named "p2snapshot". The pipeline configuration is as follows:

```

splunk>
Shared workspaces / participant2 / New pipeline / Ingest Processor
+ Add
pipeline
1 /*
2 A valid SPL2 statement for a pipeline must start with "$pipeline", and include "from $source"
3 and "into $destination".
4 */
5 $pipeline = | from $source | into $destination;
    
```

The preview table displays 614 results, showing log entries with columns for \_time, \_raw, host, index, sourcetype, source, token, and \_raw. The results are filtered by sourcetype equals cisco:asa:ip:2. The table is highlighted with a red border.

	_time	_raw	host
1	9:27:38 PM 21 May 2024	May 22 03:27:38 10.11.36.25 %ASA-6-109025: Authorization denied (acl=a cmetechoutbound) for user 'magical_johnson' from 10.11.36.10/1118 to ...	127.0.0.1
2	9:27:38 PM 21 May 2024	May 22 03:27:38 10.11.36.22 %ASA-6-302015: Built inbound UDP connection n 517979 for Outside:87.194.216.51/61397 (10.11.36.50/61397) to Inside... n 517979 for Outside:87.194.216.51/61397 (10.11.36.50/61397) to Inside...	127.0.0.1
3	9:27:39 PM 21 May 2024	May 22 03:27:39 10.11.36.25 %ASA-6-109025: Authorization denied (acl=a cmetechchode) for user 'vigilant_wilbur' from 10.11.36.31/58469 to 10.1...	127.0.0.1
4	9:27:39 PM 21 May 2024	May 22 03:27:39 10.11.36.22 %ASA-6-109025: Authorization denied (acl=a cmetechinbound) for user 'ecstatic_hypatia' from 10.11.36.6/64845 to ...	127.0.0.1
5	9:30:16 PM 21 May 2024	May 22 03:30:16 10.11.36.50 %ASA-6-302015: Built inbound UDP connection n 518016 for Outside:130.253.37.97/49234 (10.11.36.41/49234) to Inside... n 518016 for Outside:130.253.37.97/49234 (10.11.36.41/49234) to Inside...	127.0.0.1
6	9:30:16 PM 21 May 2024	May 22 03:30:16 10.11.36.49 %ASA-6-109025: Authorization denied (acl=a cmetechinbound) for user 'sweet_lalanne' from 10.11.36.34/64847 to 10.1...	127.0.0.1
7	9:30:16 PM 21 May 2024	May 22 03:30:16 10.11.36.39 %ASA-6-109025: Authorization denied (acl=a cmetechinbound) for user 'confident_sutherland' from 10.11.36.13/5239... n 5239 for Outside:130.253.37.97/49234 (10.11.36.41/49234) to Inside... n 5239 for Outside:130.253.37.97/49234 (10.11.36.41/49234) to Inside...	127.0.0.1
8	9:30:16 PM 21 May 2024	May 22 03:30:16 10.11.36.38 %ASA-6-109025: Authorization denied (acl=a cmetechinbound) for user 'serene_ramanujan' from 10.11.36.28/64844 to ...	127.0.0.1
9	9:26:44 PM 21 May 2024	May 22 03:26:44 10.11.36.8 %ASA-6-109025: Authorization denied (acl=a metechinbound) for user 'dazzling_torvalds' from 10.11.36.46/64843 to ...	127.0.0.1
10	9:26:44 PM 21 May 2024	May 22 03:26:44 10.11.36.17 %ASA-6-302020: Built ICMP connection for f addr 130.253.37.97/35949 gaddr 10.11.36.27/0 laddr 10.87.16.136/0	127.0.0.1
11	9:27:01 PM 21 May 2024	May 22 03:27:01 10.11.36.45 %ASA-6-109025: Authorization denied (acl=a cmetechinbound) for user 'serene_elion' from 10.11.36.12/64852 to 10.1...	127.0.0.1
12	9:27:06 PM 21 May 2024	May 22 03:27:06 10.11.36.37 %ASA-6-109025: Authorization denied (acl=a cmetechinbound) for user 'magical_johnson' from 10.11.36.10/1118 to ...	127.0.0.1

The pipeline also includes actions such as sending data to \$destination and filtering values. The fields section lists various fields like host, index, sourcetype, source, token, time, and raw.

## Save Your Pipeline Progress

17. We can inspect the data in the table to understand the fields and sample values that were captured during ingestion to Splunk Cloud. Note that these records shown in the preview are raw events that were also sent to the indexer. To save our progress thus far before we start editing the SPL2, choose the **Save Pipeline** button on the top right.

The screenshot shows the Splunk Ingest Processor interface. At the top, there's a header bar with buttons for </>, gear, help, and P2. A red box highlights the "Save pipeline" button on the right side of the header. Below the header is a navigation bar with "Shared workspaces / participant2 / New pipeline" and an "Ingest Processor" tab. On the left, a sidebar has a "+ Add" button and a "pipeline" section with a dropdown menu. The main area contains a code editor with the following SPL2 pipeline definition:

```

/*
A valid SPL2 statement for a pipeline must start with "$pipeline", and include "from $source"
and "into $destination".
*/
$pipeline = | from $source | into $destination;

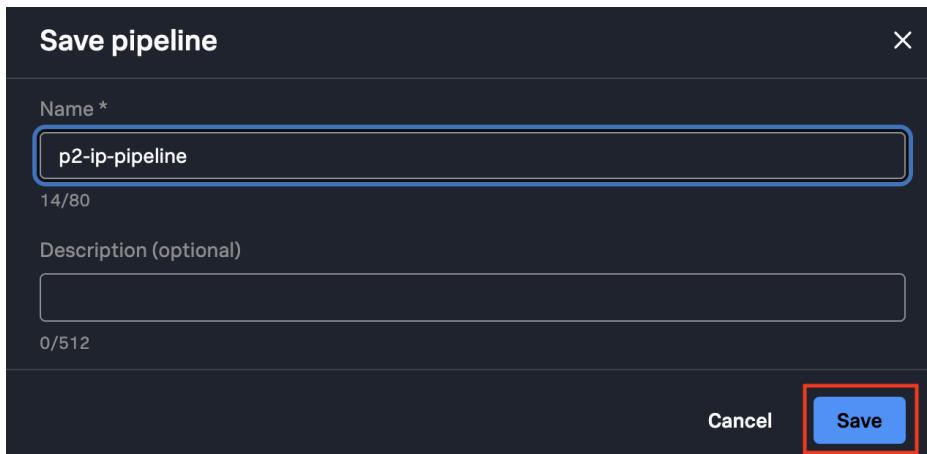
```

Below the code editor is a preview table titled "Previewing \$pipeline 614 results". The table has columns for \_time, \_raw, and host. The first few rows of data are:

	_time	_raw	host
1	9:27:38 PM 21 May 2024	May 22 03:27:38 10.11.36.25 %ASA-6-109025: Authorization denied (acl=a cmetechoutbound) for user 'magical_johnson' from 10.11.36.10/1118 to ...	127.0.0.1
2	9:27:38 PM 21 May 2024	May 22 03:27:38 10.11.36.22 %ASA-6-302015: Built inbound UDP connection n 517979 For Outside:87.194.216.51/61397 (10.11.36.50/61397) to Inside:10.11.36.22/517979	127.0.0.1
3	9:27:39 PM 21 May 2024	May 22 03:27:39 10.11.36.25 %ASA-6-109025: Authorization denied (acl=a cmetechcde) for user 'vigilant_wilbur' from 10.11.36.31/58469 to 10.11.36.25/58469	127.0.0.1
4	9:27:39 PM 21 May 2024	May 22 03:27:39 10.11.36.22 %ASA-6-109025: Authorization denied (acl=a cmetechinbound) for user 'ecstatic_hypatia' from 10.11.36.6/64845 to ...	127.0.0.1
5	9:30:16 PM 21 May 2024	May 22 03:30:16 10.11.36.50 %ASA-6-302015: Built inbound UDP connection n 518016 For Outside:138.253.37.97/49234 (10.11.36.41/49234) to Inside:10.11.36.50/518016	127.0.0.1
6	9:30:16 PM 21 May 2024	May 22 03:30:16 10.11.36.49 %ASA-6-109025: Authorization denied (acl=a cmetechinbound) for user 'sweet_lalande' from 10.11.36.34/64847 to 10.11.36.49/64847	127.0.0.1
7	9:30:16 PM 21 May 2024	May 22 03:30:16 10.11.36.39 %ASA-6-109025: Authorization denied (acl=a cmetechinbound) for user 'confident_sutherland' from 10.11.36.13/5239 to 10.11.36.39/5239	127.0.0.1
8	9:30:16 PM 21 May 2024	May 22 03:30:16 10.11.36.38 %ASA-6-109025: Authorization denied (acl=a cmetechinbound) for user 'serene_ramanujan' from 10.11.36.28/64844 to 10.11.36.38/64844	127.0.0.1
9	9:26:44 PM 21 May 2024	May 22 03:26:44 10.11.36.8 %ASA-6-109025: Authorization denied (acl=a metechinbound) for user 'dazzling_torvalds' from 10.11.36.46/64843 to 10.11.36.8/64843	127.0.0.1
10	9:26:44 PM 21 May 2024	May 22 03:26:44 10.11.36.17 %ASA-6-302020: Built ICMP connection for faddr 130.253.37.97/35949 gaddr 10.11.36.27/0 laddr 10.87.16.136/0	127.0.0.1
11	9:27:01 PM 21 May 2024	May 22 03:27:01 10.11.36.45 %ASA-6-109025: Authorization denied (acl=a cmetechinbound) for user 'serene_elion' from 10.11.36.12/64852 to 10.11.36.45/64852	127.0.0.1
12	9:27:06 PM 21 May 2024	May 22 03:27:06 10.11.36.37 %ASA-6-109025: Authorization denied (acl=a cmetechinbound) for user 'magical_johnson' from 10.11.36.10/1118 to ...	127.0.0.1

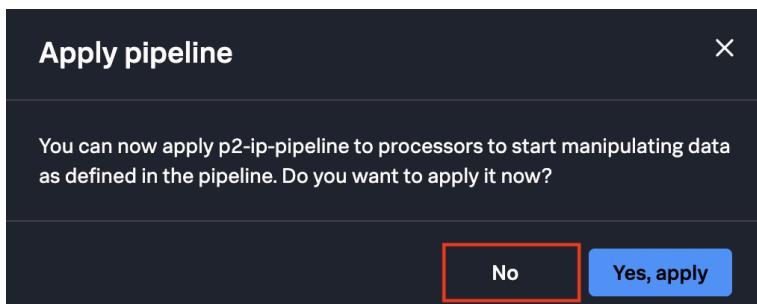
On the right side of the interface, there are sections for "pipeline", "Actions (1)", and "Fields (7)". The "Actions" section contains a single item: "Send data to \$destination splunk\_indexer". The "Fields" section lists several fields with checkboxes: host, index, sourcetype, source, token, \_time, and \_raw. The "Fields" section also includes a "Coverage" dropdown and a "Filter fields" search bar.

18. In the Save Pipeline modal that appears, name the pipeline **p#-ip-pipeline REPLACING # WITH YOUR PARTICIPANT NUMBER** and select **Save**.



19. After saving, we will be prompted in another window to Apply the pipeline. The Apply step will push out the pipeline configuration and begin transforming the data according to the SPL2 we wrote. Since we haven't written any SPL2 yet, we will select **No**.

Note: If you accidentally selected Yes, don't worry, we will overwrite the pipeline when we Apply our Cisco ASA SPL2 later.



## Author Your Pipeline

20. Now that we have our pipeline setup, we can start writing some SPL2! The first task will be to extract useful fields out of `_raw`. To do this, we will create an SPL2 function to extract the fields we care about. Copy/paste the following text into the SPL2 editor, replacing what was there before:

```
function extract_useful_fields($source) {
    return | from $source
    /* Extracted message matches with ASA or FTD */
    | rex field=_raw /(?P<_raw>(%ASA|%FTD).*)/
    /* Extract message number */
    | rex field=_raw /(%ASA|%FTD)-\d+- (?P<message_id>\d+)/
    /* Extract username */
    | rex field=_raw /^[^'\n]*' (?P<username>[^']+)/
}

$pipeline = | from $source
// extract the useful fields
| extract_useful_fields
| into $destination;
```

The `rex` functions inside of `extract_useful_fields` will do the following:

- Ensure that all `_raw` fields start with either ASA or FTD
- Extract the message ID and store it in the `message_id` field
- Extract the username (if present) and store it in the `username` field

Pressing the preview “play” button again will show that the fields have been extracted. Note that sometimes the field list can get quite long, so we have unchecked several fields on the bottom right to reduce the size of the preview results table. You can also rearrange the columns however you want.

The screenshot shows the Splunk interface for creating a new pipeline. On the left, the pipeline configuration is visible:

```

splunk>
Shared workspaces / pipelines / New pipeline Ingest Processor
+ Add
fo extract_useful_f...
-pipeline ...
fo drop_security_no...
fo mask_usernames
fo remove_duplicat...
1 function extract_useful_fields($source) {
2     return | from $source
3     /* Extracted message matches with ASA or FTD */
4     | rex field=_raw /(?P<_raw>(%ASA|%FTD).*)/
5     /* Extract message number */
6     | rex field=_raw /(%ASA|%FTD)-\d-(?P<message_id>\d+)/
7     /* Extract username */
8     | rex field=_raw /^[^\n]*(?P<username>[^"]+)/
9 }
10
11 $pipeline = | from $source
12 // extract the useful fields
13 | extract_useful_fields
14 | into $destination;
15

```

The preview pane shows 28 results from the pipeline. The first few rows of the preview table are:

	_time	_raw	# message_id	Aa username
1	9:58:24 AM 21 May 2024	%ASA-6-109025: Authorization denied (acl=acmetechinbound) for user 'intelligent_curi' e' from 10.11.36.10/55424 to 10.11.36.21/53 on interface Outside using UDP	109025	intelligent_curi
2	9:59:44 AM 21 May 2024	%ASA-6-109025: Authorization denied (acl=acmetechinbound) for user 'priceless_pare' from 10.11.36.38/64844 to 10.11.36.19/53 on interface Outside using UDP	109025	priceless_pare
3	10:00:34 AM 21 May 2024	%ASA-6-109025: Authorization denied (acl=acmetechoutbound) for user 'reverent_mendeleev' from 10.11.36.26/137 to 10.11.36.43/137 on interface Outside using UDP	109025	reverent_mendeleev
4	9:58:44 AM 21 May 2024	%ASA-6-302015: Built inbound UDP connection 518001 for Outside:141.146.8.66/65067 (1 0.11.36.35/65067) to Inside:10.86.79.167/53 (%happy_einstein)	302015	NULL
5	10:00:24 AM 21 May 2024	%ASA-6-109025: Authorization denied (acl=acmetechoutbound) for user 'romantic_yellow' from 10.11.36.6/138 to 10.11.36.3/138 on interface Outside using UDP	109025	romantic_yellow
6	10:02:34 AM 21 May 2024	%ASA-6-109025: Authorization denied (acl=acmetechinbound) for user 'focused_lalande' from 10.11.36.18/137 to 10.11.36.11/137 on interface Outside using UDP	109025	focused_lalande
7	9:58:54 AM 21 May 2024	%ASA-6-109025: Authorization denied (acl=acmetechinbound) for user 'bold_khayyam' from 10.11.36.35/137 to 10.11.36.21/137 on interface Outside using UDP	109025	bold_khayyam
8	10:01:24 AM 21 May 2024	%ASA-6-109025: Authorization denied (acl=acmetechdmz) for user 'quirky_tesla' from 10.11.36.39/1026 to 10.11.36.2/53 on interface Outside using UDP	109025	quirky_tesla
9	9:59:24 AM 21 May 2024	%ASA-6-109025: Authorization denied (acl=acmetechdmz) for user 'stoic_lichterman' from 10.11.36.36/1026 to 10.11.36.47/53 on interface Outside using UDP	109025	stoic_lichterman
10	10:01:04 AM	%ASA-6-109025: Authorization denied (acl=acmetechinbound) for user 'boring_williams'	109025	boring_williams

On the right side, the pipeline configuration is shown with a play button highlighted in red. The Fields section lists the extracted fields:

- 5 fields hidden
- # message\_id
- Aa username
- ⌚ \_time
- ⌚ \_raw
- Aa index
- Aa source
- Aa token
- Aa host
- Aa sourcetype

## Filtering

21. Next, we are going to filter certain message IDs related to security that are noisy. Add the `drop_security_noise` SPL2 function to our editor, then pipe the parsed logs into our new function. The complete SPL2 now looks like:

```
function extract_useful_fields($source) {
    return | from $source
    /* Extracted message matches with ASA or FTD */
    | rex field=_raw /(?P<_raw>(%ASA|%FTD).*)/
    /* Extract message number */
    | rex field=_raw /(%ASA|%FTD)-\d+- (?P<message_id>\d+)/
    /* Extract username */
    | rex field=_raw /[^'\n]*' (?P<username>[^']+)/
}

function drop_security_noise($source) {
    return | from $source
    | where message_id != "302013"
    | where message_id != "302015"
}

$pipeline = | from $source
// extract the useful fields
| extract_useful_fields
// Filter "302013", "302015" message ID number
| drop_security_noise
| into $destination;
```

Preview again and notice the 302013 and 302015 message IDs have been filtered.

The screenshot shows the Splunk interface for creating a new pipeline. On the left, the pipeline configuration is displayed:

```

splunk>
Shared workspaces / pipelines / New pipeline Ingest Processor
+ Add
fo extract_useful_f...
fo drop_security_no...
`-> pipeline ...
fo mask_usernames
fo remove_duplicat...
+-> pipeline
8 | rex fields=_raw /^[^(\n)*(<?<username>|[^>]+)/
9 }
10
11 function drop_security_noise($source) {
12     return | from $source
13     | where message_id != "302013"
14     | where message_id != "302015"
15 }
16
17 $pipeline = | from $source
18     // extract the useful fields
19     | extract_useful_fields
20     // Filter "302013", "302015" message ID number
21     | drop_security_noise
22     | into $destination
23
24

```

Below the code, a preview table shows 24 results:

○ _time	</> _raw	# message_id	Aa username
1 9:58:24 AM 21 May 2024	XASA-6-109025: Authorization denied (acl=acmetechinbound) for user 'intelligent_curi' from 10.11.36.19/55424 to 10.11.36.21/53 on interface Outside using UDP	109025	intelligent_curi
2 9:59:44 AM 21 May 2024	XASA-6-109025: Authorization denied (acl=acmetechinbound) for user 'priceless_pare' from 10.11.36.38/64844 to 10.11.36.15/53 on interface Outside using UDP	109025	priceless_pare
3 10:00:34 AM 21 May 2024	XASA-6-109025: Authorization denied (acl=acmetechoutbound) for user 'reverent_mendeleev' from 10.11.36.26/137 to 10.11.36.45/137 on interface Outside using UDP	109025	reverent_mendeleev
4 10:00:24 AM 21 May 2024	XASA-6-109025: Authorization denied (acl=acmetechoutbound) for user 'romantic_yallow' from 10.11.36.6/138 to 10.11.36.3/138 on interface Outside using UDP	109025	romantic_yallow
5 10:00:34 AM 21 May 2024	XASA-6-109025: Authorization denied (acl=acmetechinbound) for user 'focused_lalande' from 10.11.36.18/137 to 10.11.36.11/137 on interface Outside using UDP	109025	focused_lalande
6 9:58:54 AM 21 May 2024	XASA-6-109025: Authorization denied (acl=acmetechinbound) for user 'bold_khayyam' from 10.11.36.35/137 to 10.11.36.21/137 on interface Outside using UDP	109025	bold_khayyam
7 10:01:24 AM 21 May 2024	XASA-6-109025: Authorization denied (acl=acmetechdmz) for user 'quirky_tesla' from 10.11.36.39/1026 to 10.11.36.2/53 on interface Outside using UDP	109025	quirky_tesla
8 9:59:24 AM 21 May 2024	XASA-6-109025: Authorization denied (acl=acmetechdmz) for user 'stoic_lichterman' from 10.11.36.36/1025 to 10.11.36.47/53 on interface Outside using UDP	109025	stoic_lichterman
9 10:01:04 AM 21 May 2024	XASA-6-109025: Authorization denied (acl=acmetechinbound) for user 'boring_williams' from 10.11.36.7/52465 to 10.11.36.46/53 on interface Outside using UDP	109025	boring_williams
10 10:00:04 AM	XASA-6-113003: AAA group policy for user gifted_shamir is being set to Acme_Techcode	113003	NULL

On the right, the pipeline configuration details are shown:

- Sample data**: cisco\_asa\_snap
- Partition (1)**: sourcetype equals cisco:asa
- Actions (3)**:
  - extract\_useful\_fields
  - drop\_security\_noise
  - Send data to \$destination splunk\_indexer
- Filter values**, **Filter fields**, **Remove rows where \_raw matches ..**
- Fields (9)** (with checkboxes):
  - # message\_id
  - Aa username
  - \_time
  - </> \_raw
  - Aa index
  - Aa source
  - Aa token
  - Aa host
  - Aa sourcetype

## Masking

22. When we send this data to the index, we've decided to mask the usernames to maintain compliance. Here's the `mask_usernames` function to replace usernames found in `_raw`:

```
function mask_usernames($source) {
    return | from $source
    | eval _raw=replace(_raw, username, "[NAME_REDACTED]")
}
```

Don't forget to invoke `mask_usernames` in your pipeline:

```
$pipeline = | from $source
// extract the useful fields
| extract_useful_fields
// Filter "302013", "302015" message ID number
| drop_security_noise
// Mask usernames to protect PII
| mask_usernames
| into $destination;
```

When previewing your results, you will see the username has been redacted in `_raw`. Note that the username is still present in the `username` field.

The screenshot shows the Splunk Ingest Processor interface with a pipeline configuration and its preview results.

**Pipeline Configuration:**

```
+ Add
fo extract_useful_fi...
fo drop_security_no...
fo mask_usernames
↳ pipeline :
fo remove_duplicat...
16
17 function mask_usernames($source) {
18     return | from $source
19     | eval _raw=replace(_raw, username, "[NAME_REDACTED]")
20 }
21
22 $pipeline = | from $source
23 // extract the useful fields
24 | extract_useful_fields
25 // Filter "302013", "302015" message ID number
26 | drop_security_noise
27 // Mask usernames to protect PII
28 | mask_usernames
29 | into $destination;
```

**Preview Results:**

Showing 24 results

#	message_id	username
1	109025	intelligent_curie
2	109025	priceless_pare
3	109025	reverent_mendelev
4	109025	romantic_yellow
5	109025	focused_lalande
6	109025	bold_khayyam
7	109025	quirky_tesla
8	109025	stoic_lichterman
9	109025	boring_williams
10	113003	NULL

The `_raw` column shows the original log entry with the `username` field redacted by a red box. The `username` column shows the actual value used in the log entry.

## Drop Fields & Set Index

23. Finally, we can remove the parsed fields as they are no longer needed, and we don't want them sent to the indexer. Create the remove\_duplicate\_fields function:

```
function remove_duplicate_fields($source) {  
    return | from $source  
    | fields -username, message_id, FTD  
}
```

Add the remove\_duplicate\_fields function to your pipeline definition and set the index to dmx-ip-lab.

```
$pipeline = | from $source  
// extract the useful fields  
| extract_useful_fields  
// Filter "302013", "302015" message ID number  
| drop_security_noise  
// Mask usernames to protect PII  
| mask_usernames  
// Remove duplicate fields  
| remove_duplicate_fields  
| eval index="dmx-ip-lab"  
| into $destination;
```

Preview now shows the results without these fields:

```

splunk>
Pipelines / p2pipeline Ingest Processor
+ Add
23 |     return | from $source
24 |     | fields -username, message_id, FTD
25 |
26 }
27 $pipeline = | from $source
28 // extract the useful fields
29 | extract_useful_fields
30 // Filter "302015", "302015" message ID number
31 | drop_security_noise
32 // Mask usernames to protect PII
33 | mask_usernames
34 // Remove duplicate fields
35 | remove_duplicate_fields
36 | eval index="conf24-ip-lab"
37 | into $destination;

```

Previewing \$pipeline 533 results

	_time	_raw
1	3:36:03 PM 22 May 2024	%ASA-6-109025: Authorization denied (acl=acmetechcode) for user '[NAME_REDACTED]' from 10.11.36.2/52824 to 10.11.36.14/53 on interface outside using UDP
2	3:36:03 PM 22 May 2024	%ASA-6-109025: Authorization denied (acl=acmetechinbound) for user '[NAME_REDACTED]' from 10.11.36.43/64845 to 10.11.36.32/53 on interface Outside using UDP
3	3:38:15 PM 22 May 2024	%ASA-6-109025: Authorization denied (acl=acmetechinbound) for user '[NAME_REDACTED]' from 10.11.36.31/64852 to 10.11.36.16/53 on interface Outside using UDP
4	3:38:15 PM 22 May 2024	%ASA-6-109025: Authorization denied (acl=acmetechinbound) for user '[NAME_REDACTED]' from 10.11.36.30/49442 to 10.11.36.4/53 on interface Outside using UDP
5	3:38:17 PM 22 May 2024	%ASA-6-109025: Authorization denied (acl=acmetechinbound) for user '[NAME_REDACTED]' from 10.11.36.29/49245 to 10.11.36.13/53 on interface Outside using UDP
6	3:38:18 PM 22 May 2024	%ASA-6-109025: Authorization denied (acl=acmetechoutbound) for user '[NAME_REDACTED]' from 10.11.36.4/137 to 10.11.36.43/137 on interface Outside using UDP
7	3:38:18 PM 22 May 2024	%ASA-6-109025: Authorization denied (acl=acmetechinbound) for user '[NAME_REDACTED]' from 10.11.36.48/59140 to 10.11.36.3/53 on interface Outside using UDP
8	3:38:18 PM 22 May 2024	%ASA-6-109025: Authorization denied (acl=acmetechinbound) for user '[NAME_REDACTED]' from 10.11.36.45/53620 to 10.11.36.32/53 on interface Outside using UDP
9	3:37:42 PM 22 May 2024	%ASA-6-109025: Authorization denied (acl=acmetechdmz) for user '[NAME_REDACTED]' from 10.11.36.13/1026 to 10.11.36.26/53 on interface Outside using UDP
10	3:37:42 PM 22 May 2024	%ASA-6-109025: Authorization denied (acl=acmetechinbound) for user '[NAME_REDACTED]' from 10.11.36.50/64854 to 10.11.36.6/53 on interface Outside using UDP
11	3:37:43 PM	%ASA-6-109025: Authorization denied (acl=acmetechinbound) for user '[NAME_REDACTED]' from 10.11.36.50/64853 to 10.11.36.18/53 on interface Outside using UDP

Collapse

pipeline

Sample data p2snapshot

Partition (1)

Actions (6)

extract\_useful\_fields

drop\_security\_noise

mask\_usernames

remove\_duplicate\_fields

Target index conf24-ip-lab

Send data to destination splunk\_indexer

Filter values

Filter fields

Fields (7)

Filter fields Coverage

\_time

\_raw

host

index

sourcetype

source

token

## Apply Your Pipeline

24. We are satisfied with the results of our pipeline, so we will now apply it. Select the Save pipeline button at the top right.

The screenshot shows the Splunk pipeline editor interface. On the left, the pipeline code is displayed:

```

splunk>
Pipelines / p2pipeline Ingest Processor
+ Add
fo extract_useful_f...
fo drop_security_no...
fo mask_usernames
fo remove_duplicat...
↳ pipeline
23     return | from $source
24     | fields -username, message_id, FTD
25
26
27     pipeline = | from $source
28     | extract_useful_fields
29     | drop_security_noise
30     // Filter "302013", "302015" message ID number
31     | mask_usernames
32     // Mask usernames to protect PII
33     | mask_usernames
34     // Remove duplicate fields
35     | remove_duplicate_fields
36     | eval index="conf24-ip-lab"
37     | into $destination;

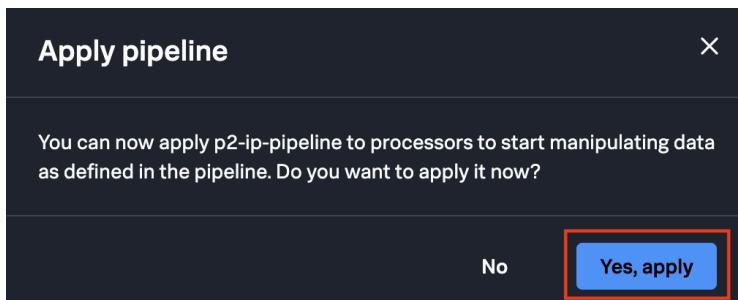
```

On the right, the pipeline configuration and preview results are shown. The pipeline configuration includes:

- Sample data:** p2snapshot
- Partition (1):** sourcetype equals cisco:asa:ip:2
- Actions (6):**
  - extract\_useful\_fields
  - drop\_security\_noise
  - mask\_usernames
  - remove\_duplicate\_fields
  - Send data to \$destination splunk\_indexer
  - Filter values
  - Filter fields
- Target index:** conf24-ip-lab
- Fields (7):**
  - \_time
  - \_raw
  - host
  - index
  - sourcetype
  - source
  - token

The preview results show 533 results, listing log entries from May 22, 2024, such as ASA-6-109025: Authorization denied (acl=acmetechinbound) for user '[NAME\_REDACTED]' from 10.11.36.4/137 to 10.11.36.4/137 on interface Outside using UDP.

If prompted, select **Yes, apply** to apply the pipeline. If you are not prompted, the pipeline will be automatically applied on save.



## Check Your Data in Splunk Cloud

25. Log in to Splunk Cloud and open up the Search & Reporting app. Run the following search over the last 15 minutes and verify that you now see the redacted events:

```
index=dmx-ip-lab sourcetype=cisco:asa:ip:#
```

(REPLACING # WITH YOUR PARTICIPANT NUMBER)

## Convert Logs To Metrics

We can instrument our incoming Cisco ASA log data using IP's capability to convert logs to metrics and send them to Splunk Observability (o11y). To illustrate this capability, we will create a counter metric that increments by one each time we see a Cisco ASA log. We will also set message\_id as a dimension, allowing us to compare message counts by ID. The metrics will be available in the Splunk Observability UI.

26. In the SPL2 code editor, at the top of the file, add the logs\_to\_metrics import statement:

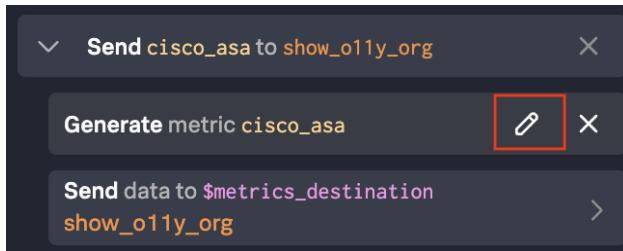
```
import logs_to_metrics from /splunk.ingest.commands
```

Replace the definition of \$pipeline with one that contains a thru command:

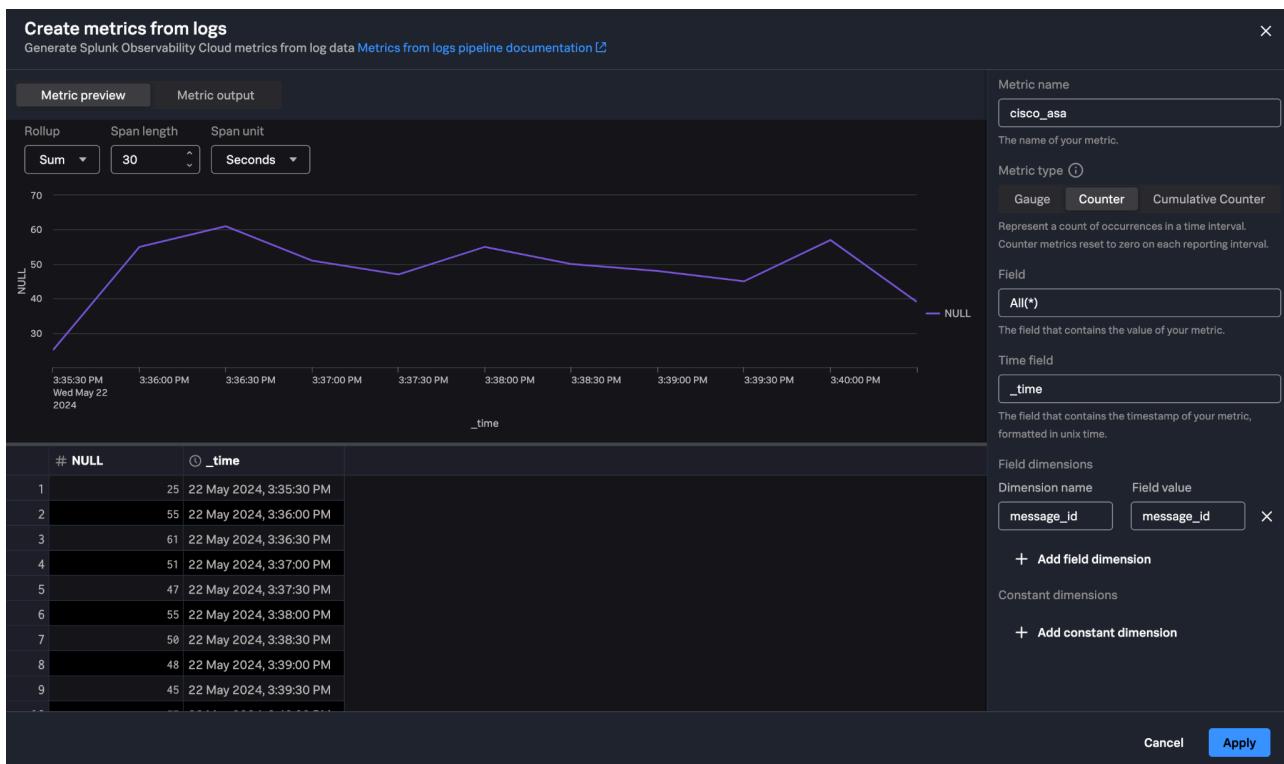
```
$pipeline = | from $source
    // extract the useful fields
    | extract_useful_fields
    // Filter "302013", "302015" message ID number
    | drop_security_noise
    // Mask usernames to protect PII
    | mask_usernames
    | thru [
        | logs_to_metrics time=_time name="cisco_asa" value=1 metrictype="counter"
dimensions={"message_id": message_id}
        | into $metrics_destination
    ]
    // Remove duplicate fields
    | remove_duplicate_fields
    | eval index="dmx-ip-lab"
    | into $destination;
```

The thru command creates a branch where each path gets a copy of the data, similar to the Unix “tee” command. In this way we are able to continue sending data to the Splunk indexer while also sending a copy of all events down the logs\_to\_metrics path.

27. Several new actions are available in the right-hand panel. Choose the “Edit” icon on the “Generate metrics cisco\_asa” to see a UI-driven metricization experience, including a preview of the metrics.

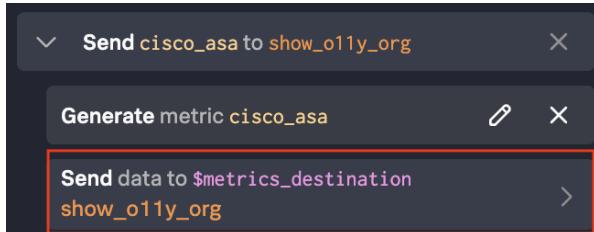


You will see a screen similar to this:



Choose cancel to exit out of this screen.

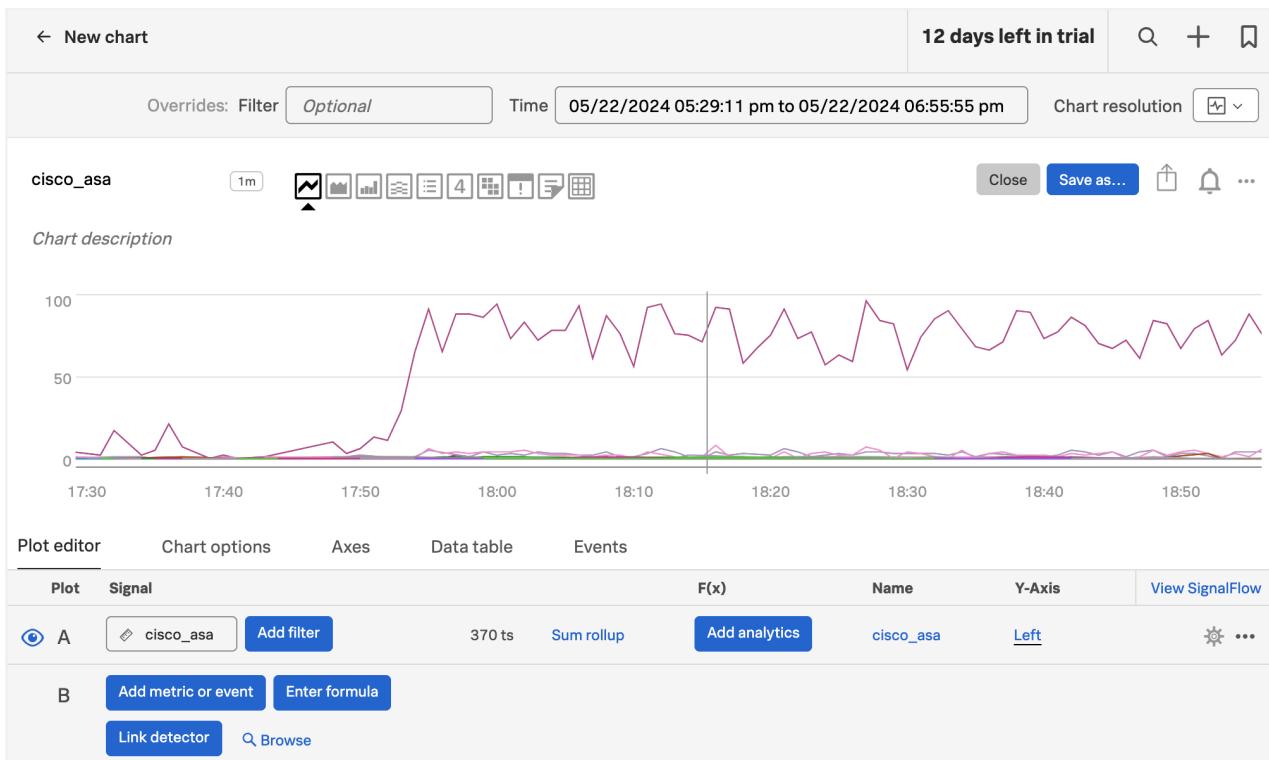
28. Because we selected show\_o11y\_org as our default metrics destination when we created this pipeline, **it should automatically be selected for you**. However if no metrics destination is selected, you can click on **Send data to \$metrics\_destination** and choose the o11y destination if necessary.



29. Save and apply your pipeline. IP will now send metrics to Splunk Observability while continuing to send logs to the Splunk Indexer.

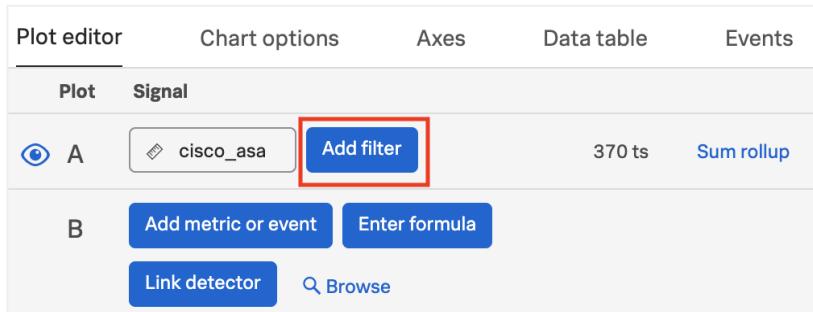
## Check Your Data in Splunk Observability

30. Log into [Splunk Observability using this link](#) with the username and password provided in [Lab Connection Info](#). If you open the link and only see a white or black screen, please log out of your existing Splunk Observability (SignalFx) account prior to proceeding (choose Settings > Sign Out).

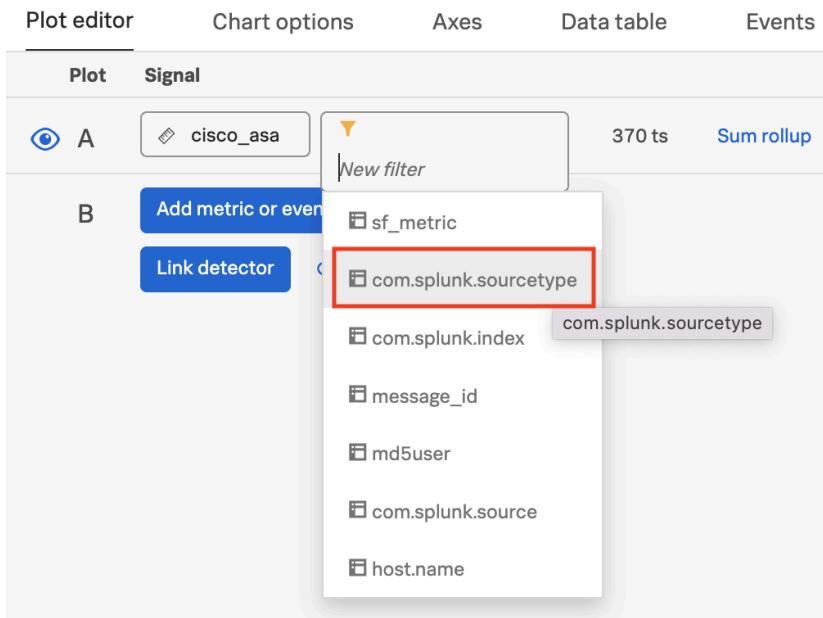


You will be taken immediately to the Metric Finder, which displays the metric values for the `cisco_asa` metric from the last 15 minutes. Metrics from other participants will also be present in this chart. Use the filtering functionality shown below to filter metrics to those sent by your IP pipeline only.

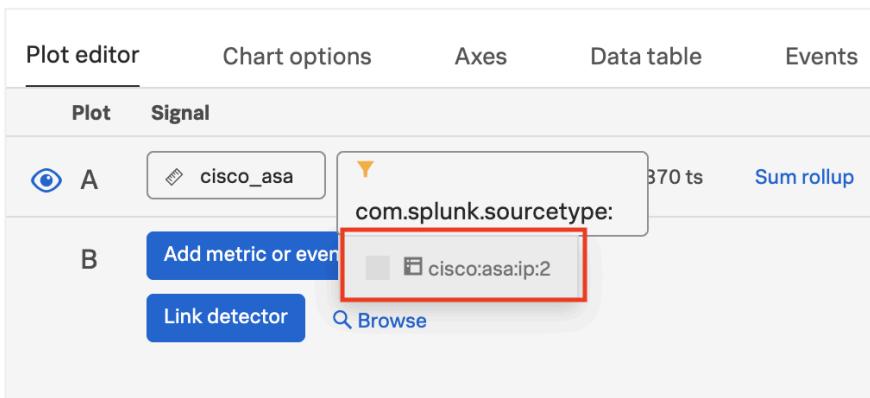
To filter by sourcetype, select Add filter → com.splunk.sourcetype → cisco:asa:ip:#.



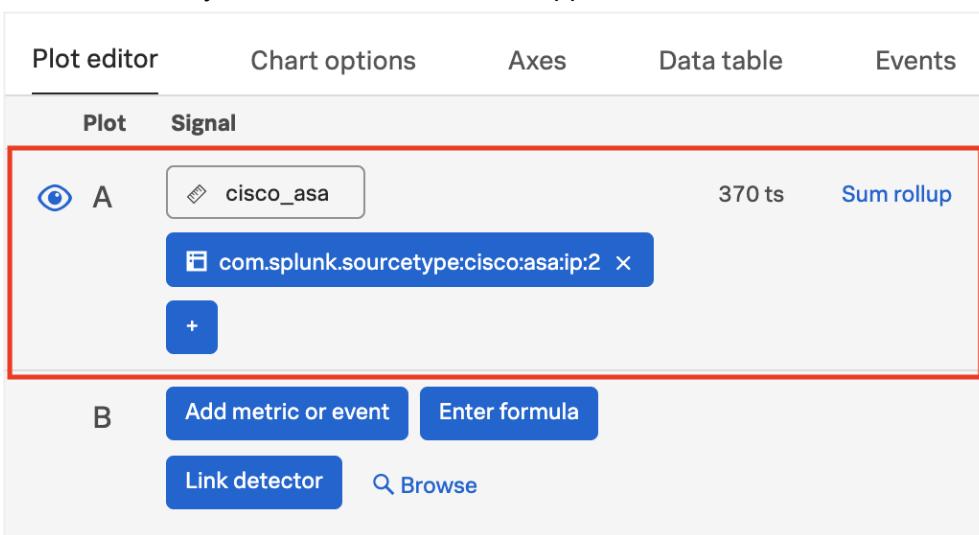
Then choose com.splunk.sourcetype.



And select the sourcetype corresponding to your participant ID.



Once selected, you will see the new filter applied to the metric.



Once your participant sourcetype has been filtered, you can confirm that Splunk o11y is receiving your metrics.

## Lab Summary

In this workshop, we explored three flexible and powerful ways to ingest data into the Splunk environment.

**Ingest Actions (IA)** allow you to quickly author, preview, and deploy transformation rules at ingest time through an intuitive user interface. These rulesets allow you to define one or more rules with a few clicks, masking, truncating, routing, or eliminating data without having to handwrite stanzas in configuration files.

In this workshop, we used Ingest Actions to route certain higher-volume, lower-value data sources to lower-cost S3 storage for future use with Splunk Federated Search for S3.

**Edge Processor (EP)** goes beyond the built-in “getting data in (GDI)” functionality available in the core platform with Ingest Actions. EP uses dedicated Edge Nodes to route, filter, mask, transform, and convert data where it resides in the customer-managed infrastructure. Edge Nodes are high-performance OTel data processors that reside proximally to the source data.

As we explored today, Edge Nodes are managed centrally by a Splunk-hosted management user interface. In the lab, we created a workspace, uploaded sample data, and built a data pipeline using SPL2. We then enriched the data in the stream via a lookup and validated functionality.

**Ingest Processor (IP)** is a data processing capability that works within your Splunk Cloud Platform deployment. Use IP to configure data flows, control data format, apply transformation rules prior to indexing, and route to destinations. The Ingest Processor solution is suitable for Splunk Cloud Platform administrators who use forwarders or HTTP Event Collector (HEC) to get data into their deployments. Ingest Processor. While IP/EP management interfaces are quite similar in scope and capability, the IP removes the administrative burden of managing and maintaining edge nodes.

In the lab, we extended the base use case that we built in the IA and EP labs. We created a workspace and a pipeline to process data in the stream. We were able to preview, save, and deploy filtering, masking, and routing use cases. Finally, we converted log data to metric-formatted data and reviewed our work.

The workshop authors hope that you enjoyed these labs and are able to take with you knowledge of both how to manage your data and when to use each Splunk capability to build more effective data pipelines.

Thank you for attending today's workshop. Happy Splunking!

## Get Started with Splunk Data Management Today

### Request access to Edge Processor or Ingest Processor

Fill out this [form](#) to request access to Data Management Pipeline Builders. If you already have access, you can directly navigate to it using the instructions on this form.

### Join the Slack channel

Get direct access to the Data Management team, ask questions, get help, and collaborate with the Community. Request access [here](#).

### Splunk Adoption Hub

Access curated [Splunk resources](#) to get started and maximize value—all in one place. Tailored to your needs and regularly updated by Splunk experts.