Splunk4Ninjas - Splunk Connect for SNMP

Version 1.2

Forward-Looking Statements

This presentation may contain forward-looking statements regarding future events, plans or the expected financial performance of our company, including our expectations regarding our products, technology, strategy, customers, markets, acquisitions and investments. These statements reflect management's current expectations, estimates and assumptions based on the information currently available to us. These forward-looking statements are not guarantees of future performance and involve significant risks, uncertainties and other factors that may cause our actual results, performance or achievements to be materially different from results, performance or achievements expressed or implied by the forward-looking statements contained in this presentation.

For additional information about factors that could cause actual results to differ materially from those described in the forward-looking statements made in this presentation, please refer to our periodic reports and other filings with the SEC, including the risk factors identified in our most recent quarterly reports on Form 10-Q and annual reports on Form 10-K, copies of which may be obtained by visiting the Splunk Investor Relations website at www.investors.splunk.com or the SEC's website at www.sec.gov. The forward-looking statements made in this presentation are made as of the time and date of this presentation. If reviewed after the initial presentation, even if made available by us, on our website or otherwise, it may not contain current or accurate information. We disclaim any obligation to update or revise any forward-looking statement based on new information, future events or otherwise, except as required by applicable law.

In addition, any information about our roadmap outlines our general product direction and is subject to change at any time without notice. It is for informational purposes only and shall not be incorporated into any contract or other commitment. We undertake no obligation either to develop the features or functionalities described, in beta or in preview (used interchangeably), or to include any such feature or functionality in a future release.

splunk>

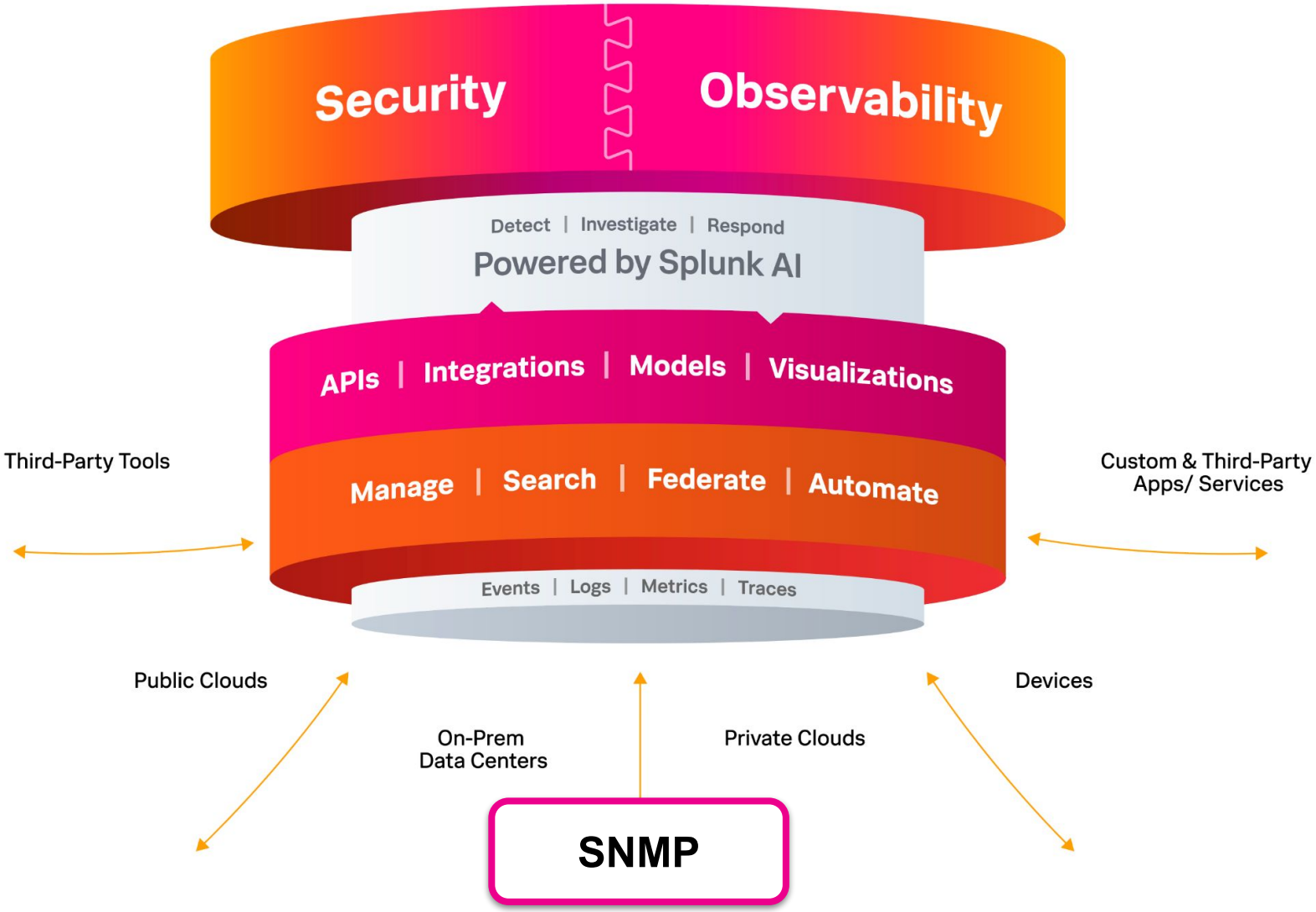
Please introduce yourself!

- Name
- Company/organisation
- Role
- Are you currently using Splunk?
- What are you interested in using Splunk for?

Workshop Agenda

- **SNMP Overview**
  - Polling, Traps and MIBs
- **Splunk Connect for SNMP (SC4SNMP)**
  - Architecture, capabilities, deployment scenarios, sizing and hardware
  - Configuration via k8s
- **Hands-on Labs:**
  - OpenTelemetry Collector deployment
  - Splunk Connect for SNMP deployment
  - Explore SNMP data in Splunk
  - Build Your Network Monitor

The Unified Security and Observability Platform



splunk>

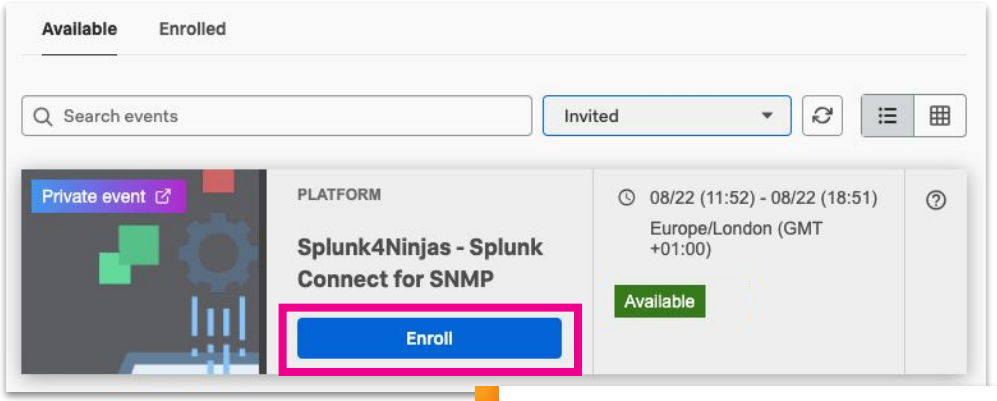
REGISTRATION 5 MINS

Enroll in Today's Workshop

Tasks

1. Get a splunk.com account if you don't have one yet: https://splk.it/SignUp
2. Enroll in the Splunk Show workshop event: https://show.splunk.com/event/<eventID>
3. Download a copy of today's slide deck: https://splk.it/S4N-SNMP-Attendee

Goal



Enroll in today's event

splunk>

SNMP: An Overview

Simple **N**etwork **M**anagement **P**rotocol

- It predates the dinosaurs and survived the asteroid impact
- Protocol invented to address management of IP based networking devices
- It has changed very little over the years since its inception

Key Components

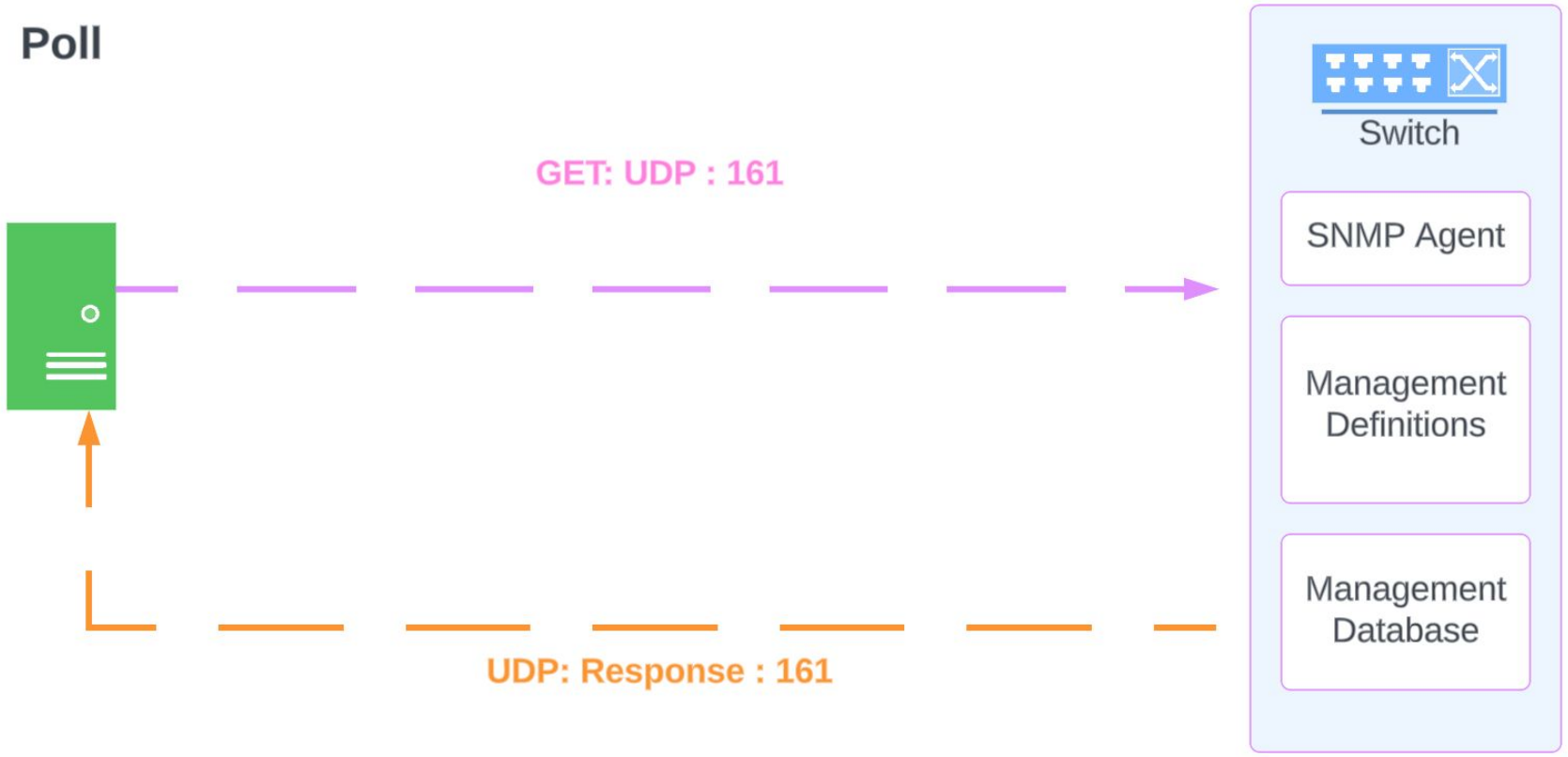
NMS - Network Management System, leverages SNMP to monitor network devices

SNMP Agent - A process(es) running on a network endpoint responding to SNMP requests

OID - Object Identifier, uniquely identify managed objects (“attributes”), which can be polled. e.g. device throughput

MIB - Management information base, contains or defines all attributes (OIDS) of an endpoint, which can be polled via SNMP

SNMP: Polling



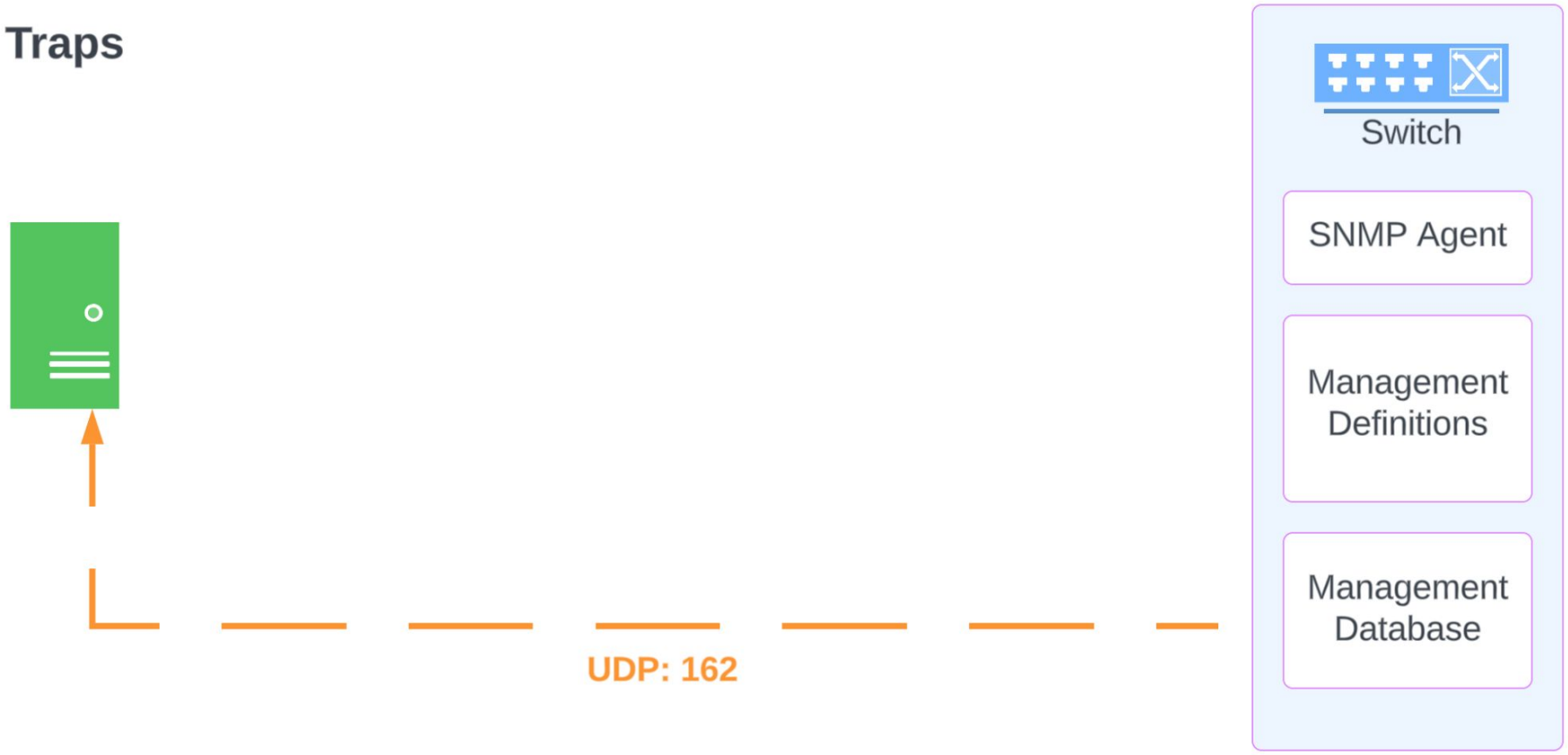
Message Types

GetRequest - Sent by NMS to the SNMP agent on the endpoint requesting a value of a attribute (OID).

GetNextRequest - Gets the next value for the an attribute.

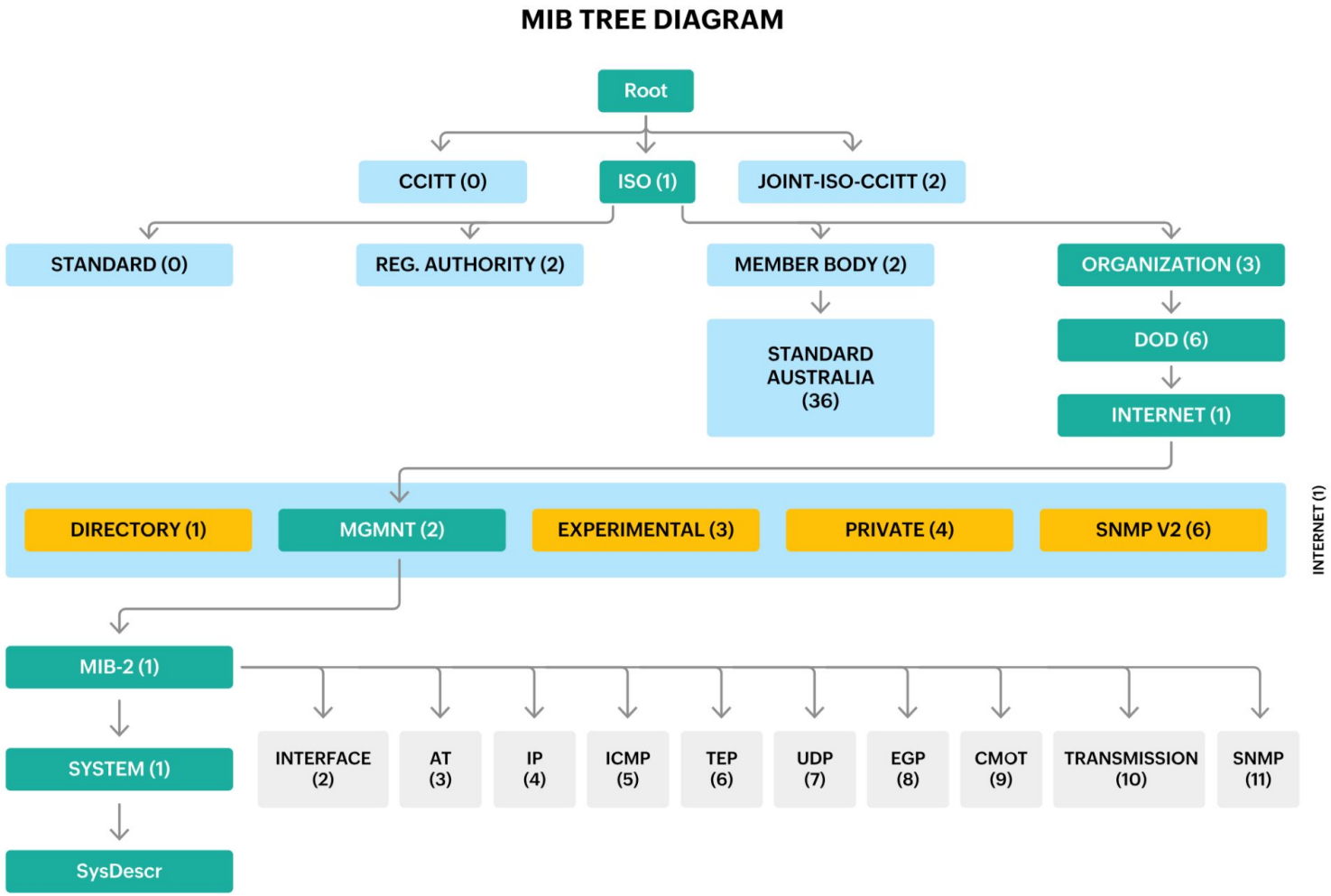
GetBulkRequest - Similar to the GetNextRequest , used to collect data in bulk.

SNMP: Traps



Traps messages are initialed by SNMP Agent on the endpont. Traps are configured on a endpoint

SNMP: MIBs



MIBs and OIDs

OID for SysDescr can be found by following the green paths down to SysDescr, e.g. 1.3.6.1.2.1.1.1

There are a few MIB browser sites which could be used as reference for generic or vendor/ device specific MIBs, e.g. https://bestmonitoringtools.com/mibdb

splunk>

Splunk Connect for SNMP: A Splunk History

Post SolarWinds supply chain hack in 2020, Splunk embarked on addressing the lack of SNMP polling and Traps GDI deficiency.

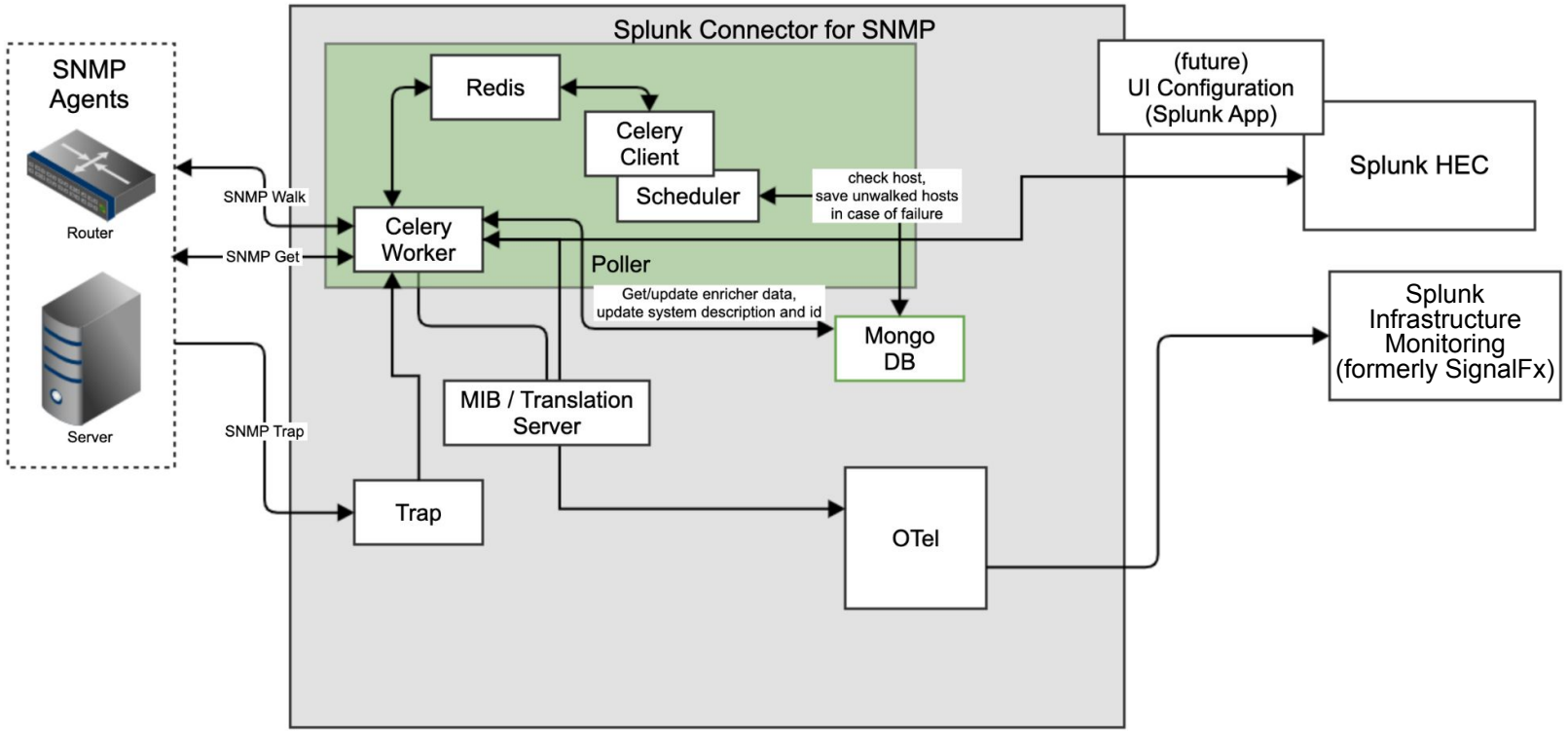
Version 0.1
Feb 2021

GA Release
Feb 2022

Current: Version 1.9.1
June 2023

splunk>

Splunk Connect for SNMP: Architecture



- **Trap** - responsible for re-register in SNMP agents and receiving traps and relaying them to the Splunk HEC (when using Splunk as an endpoint) or the OpenTelemetry (OTel) Collector (for Splunk Infrastructure Monitoring)
- **Poller** - responsible for getting data from SNMP agents and sending them to OTel/Splunk HEC
- **MIB Server** - contains a community collection of SNMP MIBs and serves them to the SNMP Workers
- **OTel Connector** - sends data to Splunk Infrastructure Monitoring. The Splunk distribution of the OTel Collector provides a binary that can be deployed as a standalone service (also known as a gateway) that can receive, process and export trace, metric and log data.

splunk>

Splunk Connect for SNMP: Architecture

Splunk Connect for SNMP is an edge-deployed, containerized, and highly available solution for collecting SNMP data for Splunk Enterprise, Splunk Cloud and Splunk Infrastructure Monitoring

- Builds on the approach used in Splunk Connect for Syslog
- Deployed into network zone using self contained Kubernetes (MicroK8s)
- CI/CD friendly headless configuration driven approach
- Can be managed by “someone other than Splunk admin”
- Support for thousands of MIBs and translation out-of-the-box

MicroK8s

redis

OpenTelemetry

Splunk Connect for SNMP: Capabilities

All these capabilities are available out-of-the-box

SNMP Trap Receiver

Supports multi-level traps
Parses fields in splunk friendly KV formats

SNMP Poller

Query Specific OIDs/MIBs

SNMP Walk of subtrees

Support operations across multiple collection times

Senses Data format and uses metrics or event indexes

MIB Library

Over 1000 MIBs loaded across major network gear manufacturers

Supported SNMP Versions

SNMP V1

- Community String Authentication
- Initial version
- Not recommended

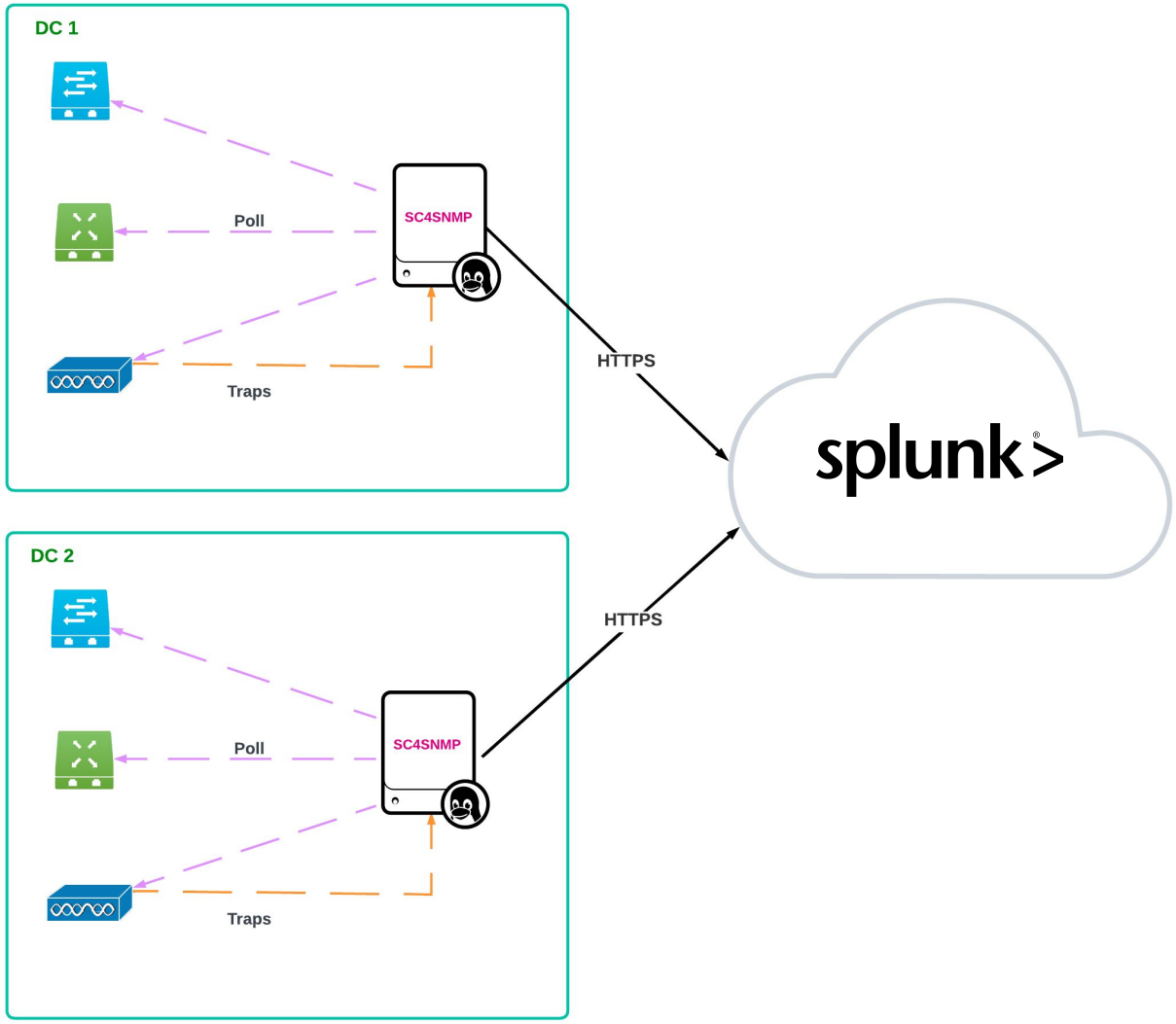
SNMP V2 (v2c)

- Community String Authentication
- Recommended for use only in internal networks if v3 is not an option

SNMP V3

- Community String Authentication
- Secure Authentication
- Auth Hash: SHA / MD5
- Privacy Encryption Algo: AES / DES

Splunk Connect for SNMP: Deployment Scenario



Splunk Connect for SNMP is designed to be deployed on the edge of a network, forwarding data to either **Splunk Enterprise**, **Splunk Cloud** or **Splunk Observability Cloud** (Splunk Infrastructure Monitoring)

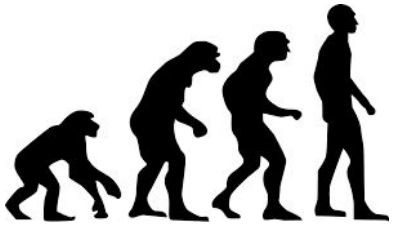
splunk>

Sizing and Hardware

- A supported deployment of MicroK8s
- 16 cores/32 threads, x64 architecture server or virtual machine, 12GB RAM
- HA requires 3 or more instances (odd numbers), 8 cores/16 threads, 16GB RAM
- 100GB root mount or in correct container mounts
- HTTP access (non proxy) allowed for the HTTP(S) connection from SC4SNMP to the Splunk destination
- Splunk Enterprise/Cloud 8.x and or Splunk Infrastructure Monitoring
  - Splunk Enterprise/Cloud specific requirements:
    - Splunk IT Service Intelligence (ITSI) or Splunk IT Essentials Work
    - Ability to create a HTTP Event Collector (HEC) token
    - Ability to create event and metrics indexes (or use existing)

*** Tested on ~ 2000 traps/second - polling numbers in the works ***

Kubernetes: A Very Brief Introduction



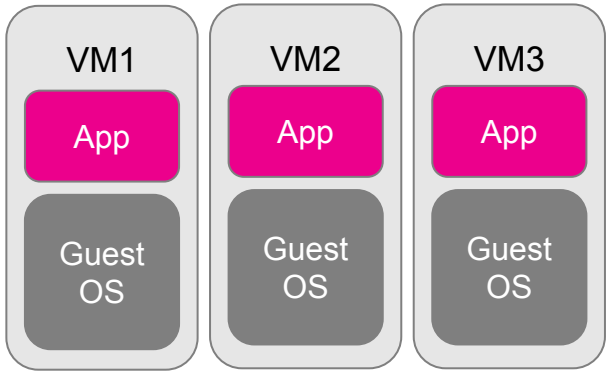
Physical

App

Host Operating System

Host Hardware

Virtual



Hypervisor

Host Operating System

Host Hardware

Containers

Container Runtime

Host Operating System

Host Hardware

splunk>

Some Useful Terminology

Node

Virtual or Physical machine where Pods are run.

Nodes can be part of a K8s cluster

Pod

Consists of 1 or more containers. Number of Pods can be scaled up or down as needed.

Namespace

A way of organizing cluster/ node resources into logical groups. A cluster can be have multiple namespaces.

Master

K8s clusters are comprised of Master and Worker nodes. Master nodes control the state of the cluster.

Worker

Worker nodes host pods where workloads are run.

A config file in YAML or JSON that describes the resources on a cluster. e.g. pods, services, deployments, etc.

splunk>

Splunk Connect for SNMP: Services and Pods

There are 4 main services for SC4SNMP:

Poller

Responsible for querying SNMP devices. The two main operations used by the poller are:

GET - to query OIDs defined in values.yaml file
WALK - query devices to collect metadata

Scheduler

Responsible for managing schedules for SNMP walks and GETs

Traps

Traps service is responsible for capturing SNMP traps sent by endpoints

Sender

Sender service handles forwarding of Splunk Connect for SNMP data to Splunk

Poller

values.yaml sample:

```
poller:
  usernameSecrets:
    - sc4snmp-hlab-sha-aes
    - sc4snmp-hlab-sha-des
  logLevel: "WARN"
  inventory: |
    address,port,version,community,secret,security_engine,
    walk_interval,   profile,smart_profiles,delete
    10.202.4.202,,2c,public,,,2000,,,
```

usernameSecrets
Secrets for SNMPv3

logLevel
Log level for poller. Default is `WARNING`. Can also be set to `DEBUG`, `INFO` or `ERROR`.

Inventory
Defines which endpoints and how to poll them

In this example:

Endpoint polled: `10.202.4.202`
Port: `default` (161)
SNMP Version: `2c`
Community String: `public`
Walk Interval: `2000`

Mandatory Fields:

- **address** - IP address which SC4SNMP will connect to to collect data
- **version** - SNMP Version. Allowed values: `1`, `2c` , `3`

splunk>

Scheduler: Profiles and Walks

values.yaml sample:

```
scheduler:
    logLevel: “WARN”
    profiles:
        smart_profile:
        frequency: 100
        condition:
          type: field
          field: "SNMPv2-MIB.sysDescr"
          patterns:
            - '.*linux.*'
        varBinds:
          - ['SNMPv2-MIB']
          - ['SNMPv2-MIB', 'sysName']
          - ['SNMPv2-MIB', 'sysUpTime',0]
      static_profile:
        frequency: 300
        varBinds:
          - ['IP-MIB']
      short_walk:
      condition:
        type: "walk"
      varBinds:
        - ['IF-MIB']
```

Profiles
There a two types which can be set up:

Static profile
Polling starts when the profile is added to the `profiles` field in the inventory of the device

Smart profile
polling starts when configured conditions are fulfilled. In this example the `smart_profile` settings mean polling will only occur if the sysDescr of the endpoint contains “`linux`”

Walk Profile
A setting to limit the scope of an SNMP WALK to the defined MIB. Used together with profile. In this the example `static_profile` has be configured to walk only the `IF-MIB`

Best Practice:
Use walks to improve performance

splunk>

Scheduler: varBinds and Custom Translations

values.yaml sample:

```
scheduler:
  logLevel: “WARN”
    profiles: |
        smart_profile:
        frequency: 100
        condition:
          type: field
          field: "SNMPv2-MIB.sysDescr"
          patterns:
            - '.*linux.*'
        varBinds:
          - ['SNMPv2-MIB']
          - ['SNMPv2-MIB', 'sysName']
          - ['SNMPv2-MIB', 'sysUpTime',0]
    customTranslations:
     IF-MIB:
            ifInDiscards: InterfaceDiscards
               ifOutErrors: InterfaceOutErrors
         SNMPv2-MIB:
              sysDescr: SystemDescr
```

varBinds (variable bindings)
Used for defining what OIDs should be requested from SNMP Agents.

e.g.
Entire MIB
`[‘IF-MIB’]`

Specific OIDS in IF-MIB
`[‘IF-MIB’, ‘ifDescr’]`

Best Practice:
Try to define specific OIDS from a MIB are some MIB are be very large.

customTranslations
Default OID names can be changed to custom values. All translations need to be grouped by MIB family e.g IF-MIB in this example `ifInDiscards` is being changed to “`InterfaceDiscards`”.

splunk>

Scheduler: Groups

values.yaml sample:

```
scheduler:
    logLevel: “WARN”
    groups: |
        group_1:
            - address: 10.202.4.202
              port: 161
            - address: 63.2.40.0
              port: 161
        group_2:
            - address: 103.0.0.1
              port: 1161
              version: '3'
              secret: 'my_secret'
poller:
      inventory: |
        address,port,version,community,secret,security_engine,
        walk_interval,profiles,smart_profiles,delete
        group_1,,2c,public,,,2000,static_profile,,
        group_2,,3,,my_secret,,2000,static_profile,,
```

Groups
Devices can be grouped and group name used in inventory, which will save manual creation for each device in the polling section and making management easier.

poller - inventory
Group reference in poller inventory. In the example `group_1` and `group_2` and being referenced.

Group Note: If an endpoint has an entry in **inventory** and **group** then the group settings are used

splunk>

Traps

values.yaml sample:

```
traps:
   communities:
        1:
          - public
        2c:
          - public
          - ninjaslab
   usernameSecrets:
        - secretv3
        - sc4snmp-Ninjas-secure-sha-des

    # Overrides the image tag whose default is the chart appVersion.
   logLevel: "WARN"
    loadBalancerIP: 10.202.4.202
```

Communities
Used to define SNMP version and community string

usernameSecrets
Used for v3 secrets for traps messages sent by endpoints

logLevel
Log level for poller. Default is `WARNING`. Can also be set to `DEBUG`, `INFO` or `ERROR`.

Workers

There are 3 critical worker pods for Splunk Connect for SNMP. The **worker** section is used to control the number of worker pods.

values.yaml sample:

```
worker:
  poller:
    autoscaling:
      enabled: true
      minReplicas: 4
      maxReplicas: 10
      targetCPUUtilizationPercentage: 80
  sender:
    replicaCount : 2
  trap:
    replicaCount: 0
  resources:
    limits:
      cpu: 500m
      memory: 512Mi
    requests:
      cpu: 200m
      memory: 256Mi
```

autoscaling
In this example poller pods will scale from min of 4 to max of 10 if CPU Utilization > 80%

no autoscaling
Static assignment of number of pods of the worker.
Best Practice:
2x the number of nodes

Disable worker
Setting a `replicaCount` to `0` disables the worker. In this example traps will be disabled.

resources
`limits` - maximum amount of resource that can be consumed.
`requests` - minimum guaranteed amount that will be allocated for the worker.

splunk>

MIBs and MIB Server Updates

To check for MIBs stored in the default install of Splunk Connect for SNMP:
https://pysnmp.github.io/mibs/index.csv

If a customer MIB needs to be added to the MIB server for polling there are two routes:

1. MIB file submission via repo or creating an issue:
   https://splunk.github.io/splunk-connect-for-snmp/main/mib-request/#submit-a-new-mib-file

2. Use of local MIB files (MIB Server version 1.15.0 +). MIB files can be added to mib server without the MIB request process:
   https://splunk.github.io/splunk-connect-for-snmp/main/mib-request/#beta-use-mib-server-with-local-mibs

Tips from the Trenches: Walks and Polls

For Scheduler Walks and Polls try **NOT** to use whole MIBs, e.g. TCP-MIB or UDP-MIB. Be selective about what you want to poll. Polls and Walks can very compute intensive for endpoints.

TCP and IF Recommendations:

TCP-MIB

```
- ["TCP-MIB", "tcpActiveOpens"]
- ["TCP-MIB", "tcpAttemptFails"]
- ["TCP-MIB", "tcpCurrEstab"]
- ["TCP-MIB", "tcpEstabResets"]
- ["TCP-MIB", "tcpInErrs"]
- ["TCP-MIB", "tcpInSegs"]
- ["TCP-MIB", "tcpMaxConn"]
- ["TCP-MIB", "tcpOutRsts"]
- ["TCP-MIB", "tcpOutSegs"]
- ["TCP-MIB", "tcpPassiveOpens"]
- ["TCP-MIB", "tcpRetransSegs"]
- ["TCP-MIB", "tcpRtoMax"]
- ["TCP-MIB", "tcpRtoMin"]
```

IF-MIB

```
- ["IF-MIB", "ifInDiscards"]
- ["IF-MIB", "ifInErrors"]
- ["IF-MIB", "ifInNUcastPkts"]
- ["IF-MIB", "ifInOctets"]
- ["IF-MIB", "ifInUcastPkts"]
- ["IF-MIB", "ifInUnknownProtos"]
- ["IF-MIB", "ifOutDiscards"]
- ["IF-MIB", "ifOutErrors"]
- ["IF-MIB", "ifOutNUcastPkts"]
- ["IF-MIB", "ifOutOctets"]
- ["IF-MIB", "ifOutQLen"]
- ["IF-MIB", "ifOutUcastPkts"]
```

Poll / Walk Frequency

Walks frequency:
Frequency : 86400 secs or higher
Endpoints will generally not change frequently

Polling Frequency:
Device Specific Health OIDS: 300 secs
e.g CPU, Mem, internal temp, etc.

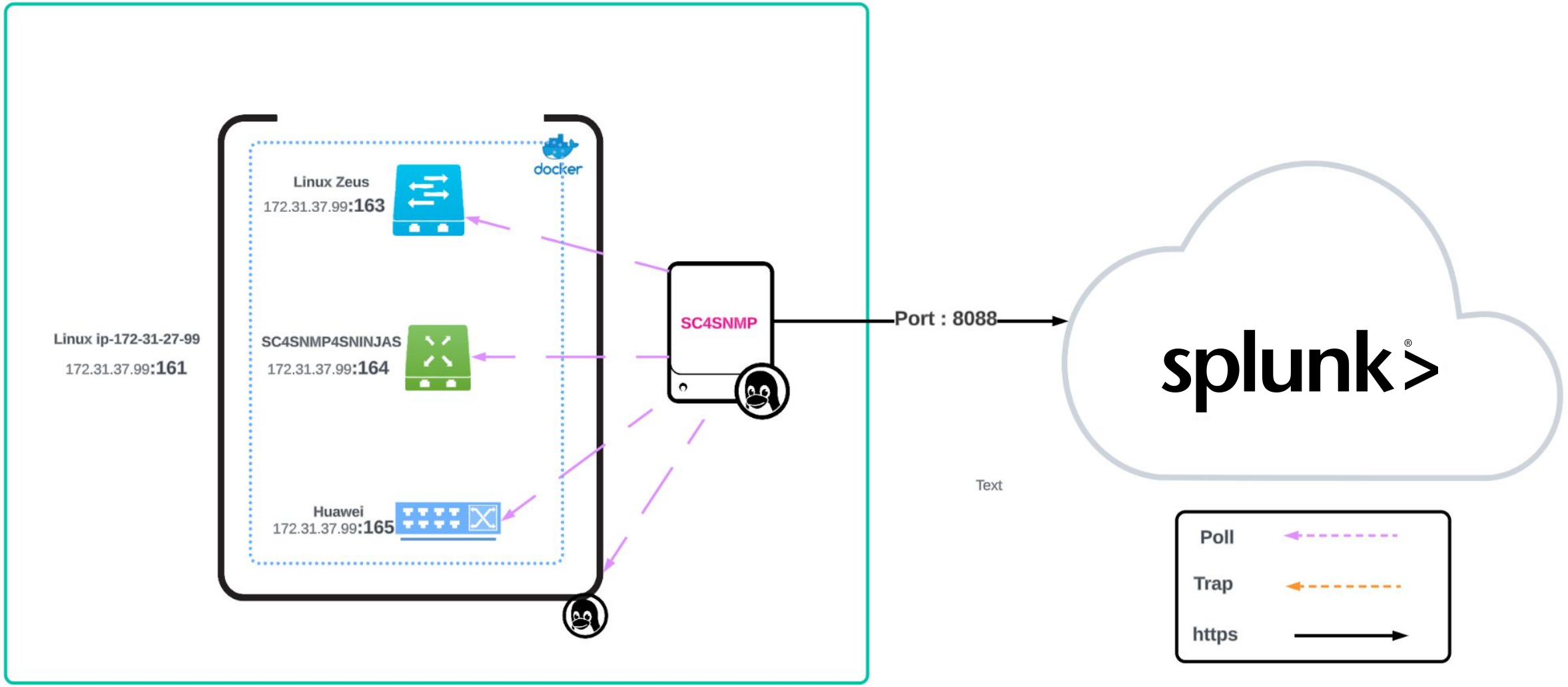
IF and TCP OIDs: 60 to 300 secs

splunk>

Hands-On Labs

splunk>

Lab Environment

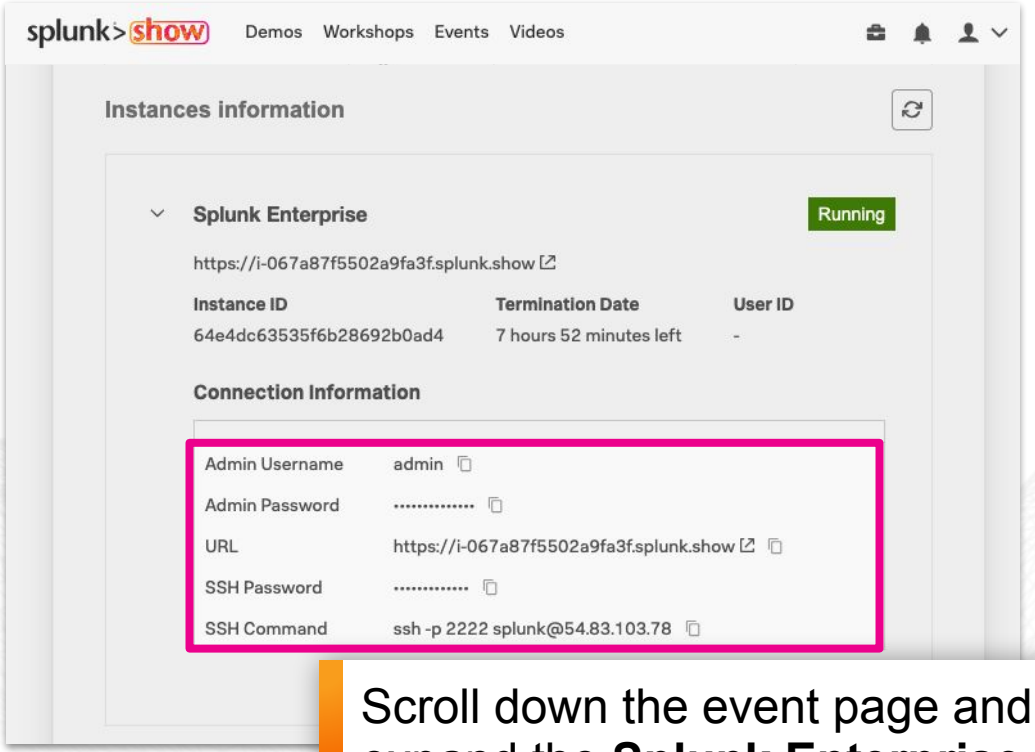


Lab Tasks

1. Splunk Connect for SNMP deployment
2. Splunk Cloud configuration
3. Configure Splunk Connect for SNMP for polling
4. Create dashboard views in Splunk

Login to Splunk

1. Locate your instance URL and credentials in the Splunk Show event: https://show.splunk.com



Scroll down the event page and expand the **Splunk Enterprise** section to view your login details

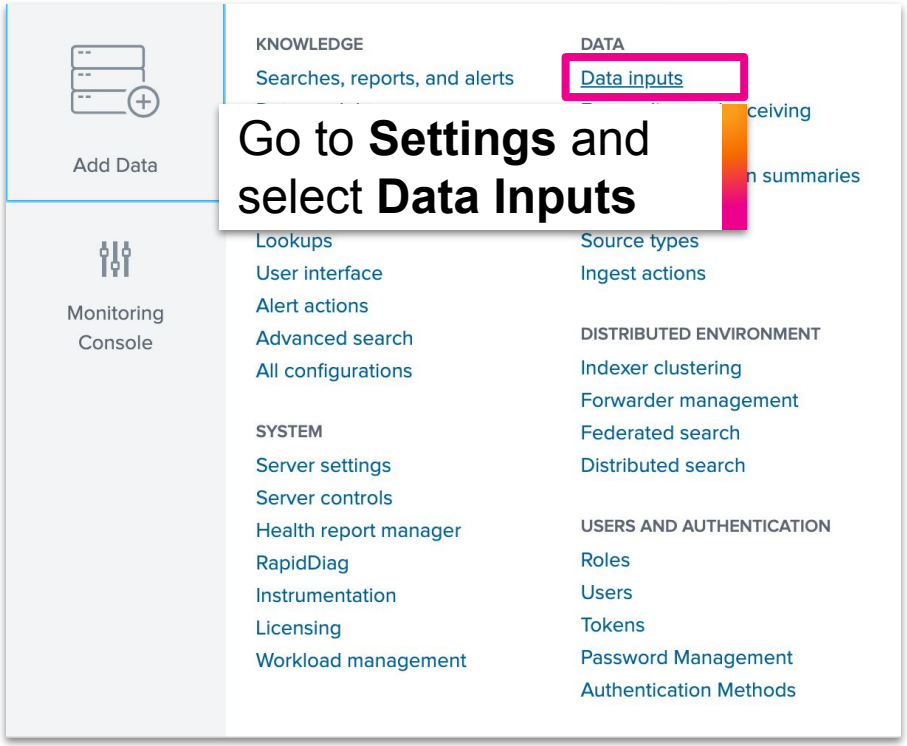
2. Log in to your Splunk instance using the Admin username and password from Splunk Show

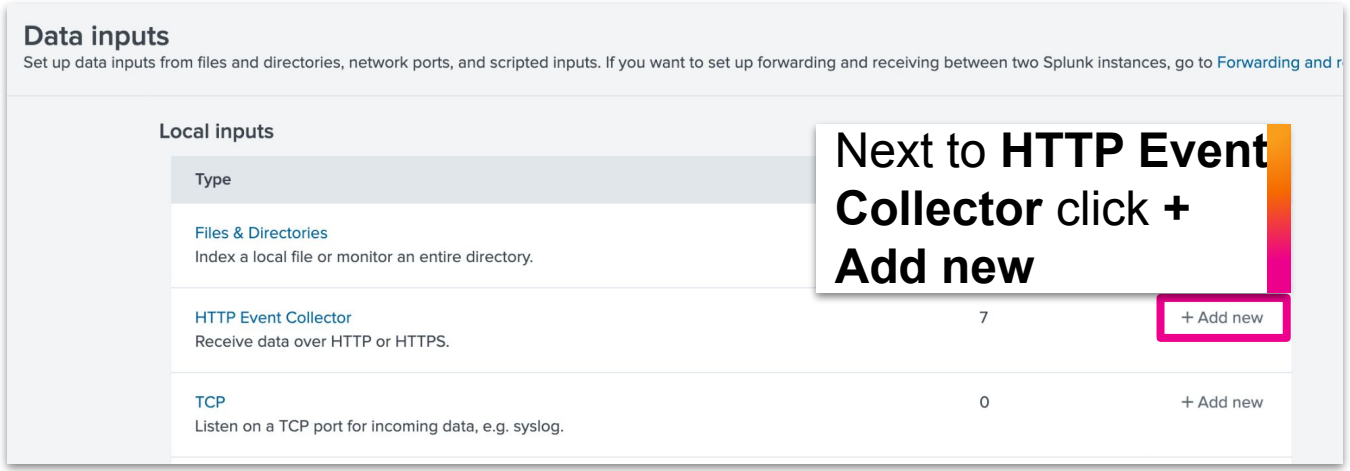
3. Test SSH connectivity using the SSH command and password from Splunk Show:

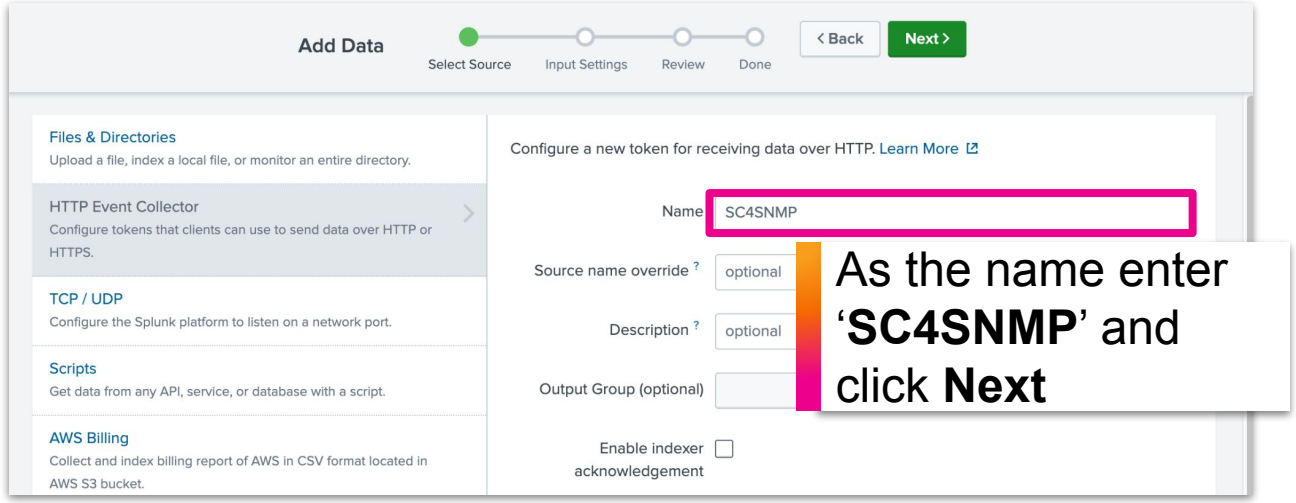
```
ssh -p 2222 splunk@<instance_IP>
```

splunk>

Generate a HEC token



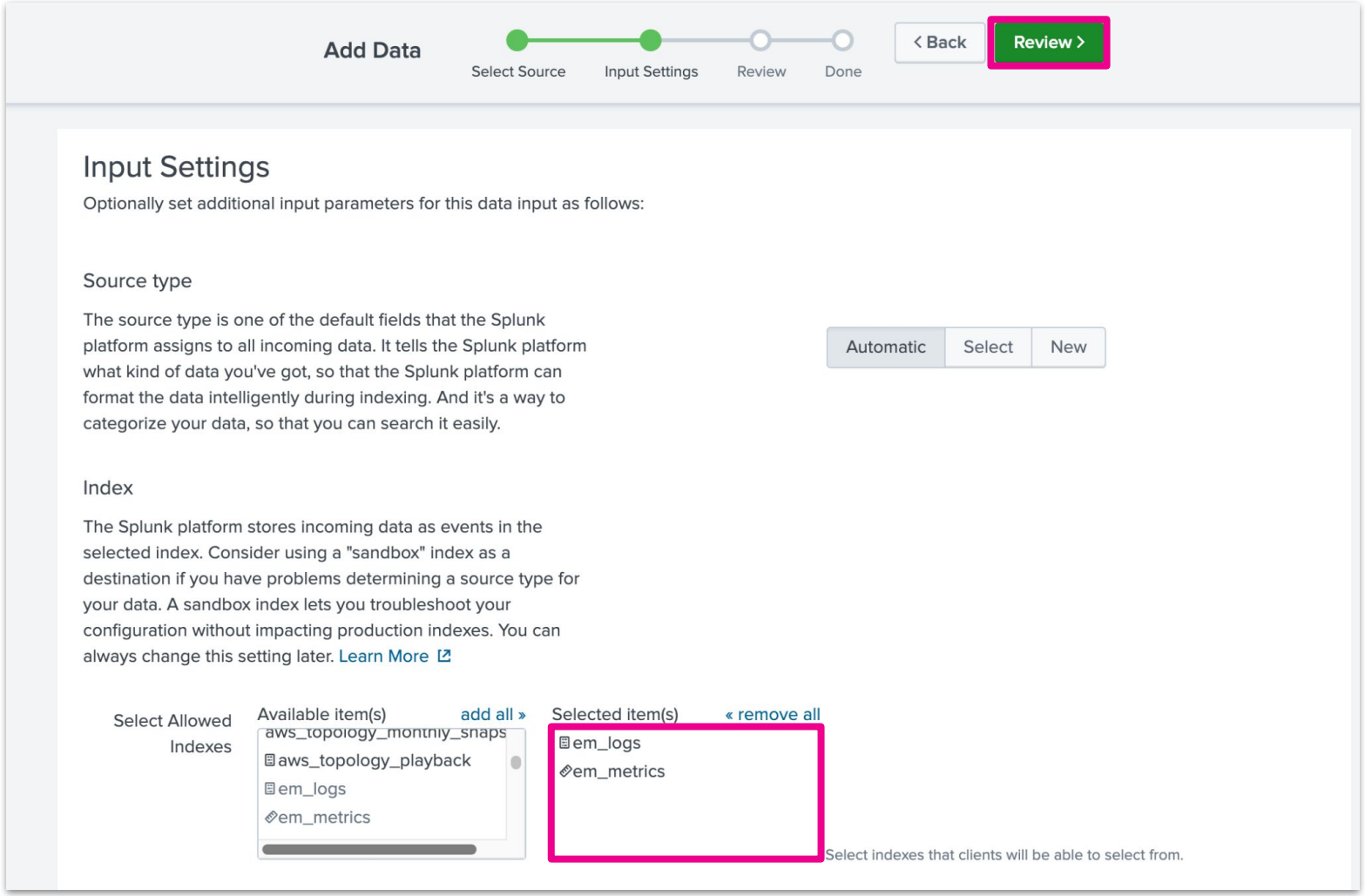




splunk>

Generate a HEC token

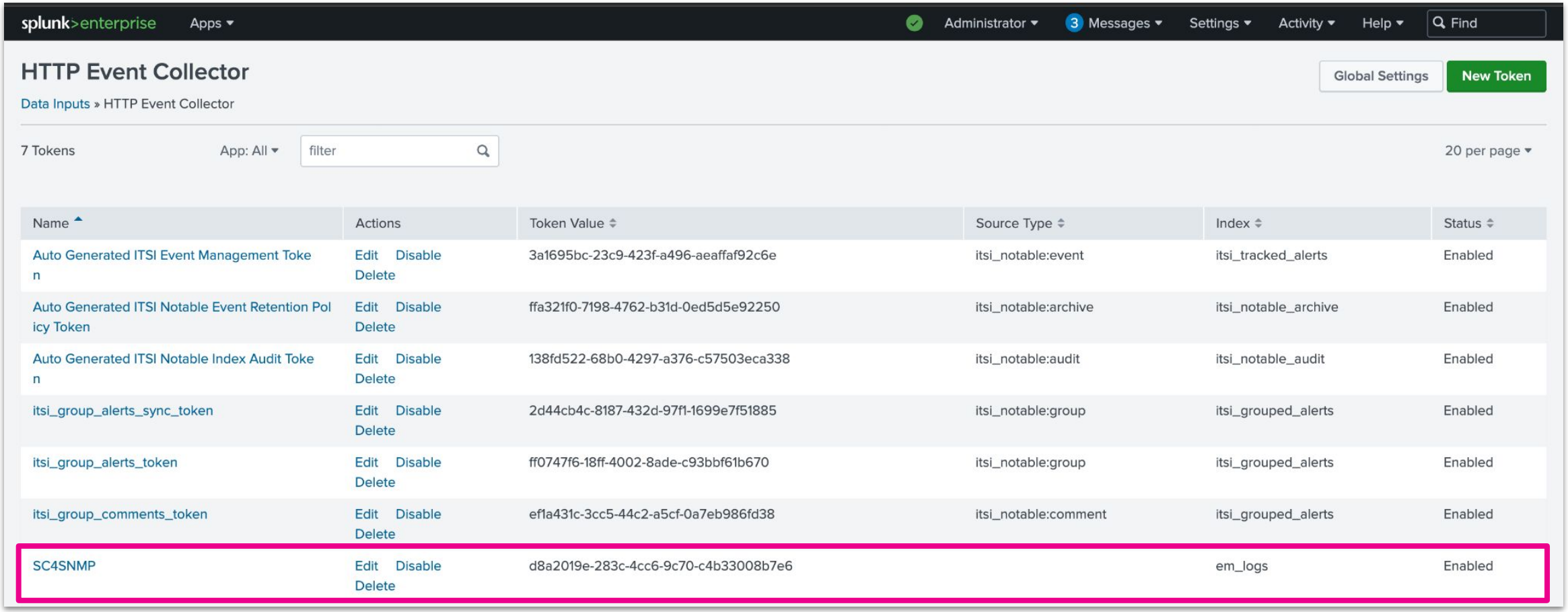
Add the **em_logs** and **em_metrics** indexes and click **Review**, then **Submit** to finish creating a HEC token.



splunk>

Generate a HEC token

Go to **Settings > Data Inputs > HTTP Event Collector** to confirm the creation of the token. This token will be used to configure Splunk Connect for SNMP.



Name	Actions	Token Value	Source Type	Index	Status
Auto Generated ITSI Event Management Token	Edit Disable Delete	3a1695bc-23c9-423f-a496-aeaffaf92c6e	itsi_notable:event	itsi_tracked_alerts	Enabled
Auto Generated ITSI Notable Event Retention Policy Token	Edit Disable Delete	ffa321f0-7198-4762-b31d-0ed5d5e92250	itsi_notable:archive	itsi_notable_archive	Enabled
Auto Generated ITSI Notable Index Audit Token	Edit Disable Delete	138fd522-68b0-4297-a376-c57503eca338	itsi_notable:audit	itsi_notable_audit	Enabled
itsi_group_alerts_sync_token	Edit Disable Delete	2d44cb4c-8187-432d-97f1-1699e7f51885	itsi_notable:group	itsi_grouped_alerts	Enabled
itsi_group_alerts_token	Edit Disable Delete	ff0747f6-18ff-4002-8ade-c93bbf61b670	itsi_notable:group	itsi_grouped_alerts	Enabled
itsi_group_comments_token	Edit Disable Delete	ef1a431c-3cc5-44c2-a5cf-0a7eb986fd38	itsi_notable:comment	itsi_grouped_alerts	Enabled
SC4SNMP	Edit Disable Delete	d8a2019e-283c-4cc6-9c70-c4b33008b7e6		em_logs	Enabled

splunk>

Connect via SSH

In Terminal (for Mac) or Command Prompt/PowerShell (for Windows) connect via SSH:

```
ssh -p 2222 splunk@<instance_IP>
```

```
Expanded Security Maintenance for Applications is not enabled.

51 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status


Last login: Mon Aug 21 00:40:45 2023 from 138.130.26.245
splunkadmin@ip-172-31-47-244:~$
```

You should be able to see this once logged in

SNMPWalk to Test Connectivity to Endpoints

Switch to the **root** user (`sudo su`) and run the following command:

```
snmpwalk -v2c -c public  localhost 1.3.6.1.2.1.1.1
```

- Command used to poll endpoint for a specific MIB or OIDs to gather meta for the device
- snmp version
- Community string for Authentication
- Endpoint IP address
- OID - SNMPv2-MIB sysDescr

```
root@ip-172-31-47-244:/home/splunkadmin# snmpwalk -v2c -c public localhost 1.3.6.1.2.1.1.1
iso.3.6.1.2.1.1.1.0 = STRING: "Linux ip-172-31-47-244 6.2.0-1009-aws #9~22.04.3-Ubuntu SMP Tue Aug  1 21:11:51 UTC 2023 x86_64"
root@ip-172-31-47-244:/home/splunkadmin#
```

splunk>

OpenTelemetry (OTel) Collector Deployment

Connect to your instance via SSH:

```
ssh -p 2222 splunk@<instance_IP>
```

Switch to the **root** user and run the following command:

```
microk8s helm3 upgrade --install sck \
  --set="clusterName=sc4snmp4ninjas" \
  --set="splunkPlatform.endpoint=https://<instanceIP>:8088/services/collector/event" \
  --set="splunkPlatform.insecureSkipVerify=true" \
  --set="splunkPlatform.token=<HEC TOKEN>" \
  --set="logsEngine=otel" \
  --set="splunkPlatform.metricsEnabled=true" \
  --set="splunkPlatform.metricsIndex=em_metrics" \
  --set="splunkPlatform.index=em_logs" \
  splunk-otel-collector-chart/splunk-otel-collector --version 0.79.0
```

OpenTelemetry (OTel) Collector Deployment

The output should be similar to the screenshot below:

```
root@ip-172-31-47-244:/home/ubuntu# microk8s helm3 upgrade --install sck \
  --set="clusterName=sc4snmp4ninjas" \
  --set="splunkPlatform.endpoint=https://localhost:8088/services/collector/event" \
  --set="splunkPlatform.insecureSkipVerify=true" \
  --set="splunkPlatform.token=5e657bb0-334e-45c3-b39d-53d51e4e813f" \
  --set="logsEngine=otel" \
  --set="splunkPlatform.metricsEnabled=true" \
  --set="splunkPlatform.metricsIndex=em_metrics" \
  --set="splunkPlatform.index=em_logs" \
  splunk-otel-collector-chart/splunk-otel-collector --version 0.79.0
Release "sck" has been upgraded. Happy Helming!
NAME: sck
LAST DEPLOYED: Mon Aug 21 09:03:31 2023
NAMESPACE: default
STATUS: deployed
REVISION: 7
TEST SUITE: None
NOTES:
Splunk OpenTelemetry Collector is installed and configured to send data to Splunk Platform endpoint "https://localhost:8088/services/collector/event".
```

splunk>

Splunk Connect for SNMP Deployment

While connected via SSH, run the following command as the **root** user:

```
microk8s kubectl get pods -A
```

```
root@ip-172-31-31-149:/home/ubuntu# microk8s kubectl get pods -all
error: unknown shorthand flag: 'a' in -all
See 'kubectl get --help' for usage.
root@ip-172-31-31-149:/home/ubuntu# microk8s kubectl get pods -A
NAMESPACE        NAME                                                      READY   STATUS    RESTARTS       AGE
kube-system      hostpath-provisioner-766849dd9d-d9rd7                     1/1     Running   0              14h
kube-system      calico-node-rl9mb                                         1/1     Running   0              14h
kube-system      calico-kube-controllers-d8b9b6478-2hzqw                   1/1     Running   0              14h
kube-system      coredns-d489fb88-2h54n                                    1/1     Running   0              14h
kube-system      metrics-server-6b6844c455-2qgms                           1/1     Running   0              14h
metallb-system   controller-56c4696b5-rd4pp                                1/1     Running   0              14h
metallb-system   speaker-vjpl4                                             1/1     Running   0              14h
default          sck-splunk-otel-collector-k8s-cluster-receiver-788b4bc958-g9mdc   1/1     Running   8 (33m ago)    13h
default          sck-splunk-otel-collector-agent-9clq7                     1/1     Running   56 (13m ago)   13h
```

Ensure that all pods are in a running state

splunk>

Splunk Connect for SNMP Deployment (cont.)

Add SC4SNMP to the k8s repo by running these two commands:

```
microk8s helm3 repo add splunk-connect-for-snmp https://splunk.github.io/splunk-connect-for-snmp
```

```
microk8s helm3 repo update
```

```
root@ip-172-31-31-149:/home/ubuntu# microk8s helm3 repo update
Hang tight while we grab the latest from your chart repositories...
...Successfully got an update from the "splunk-otel-collector-chart" chart repository
...Successfully got an update from the "splunk-connect-for-snmp" chart repository
Update Complete. ⎈Happy Helming!⎈
root@ip-172-31-31-149:/home/ubuntu#
```

SC4SNMP has now been added to the k8s repo

Check if SC4SNMP exists in the repo by running:

```
microk8s helm3 search repo snmp
```

```
root@ip-172-31-31-149:/home/ubuntu# microk8s helm3 search repo snmp
NAME                                            CHART VERSION   APP VERSION     DESCRIPTION
splunk-connect-for-snmp/splunk-connect-for-snmp 1.9.1           1.9.1           A Helm chart for SNMP Connect for SNMP
root@ip-172-31-31-149:/home/ubuntu# ||
```

You're now ready to configure Splunk Connect for SNMP!

splunk>

Splunk Connect for SNMP Deployment (cont.)

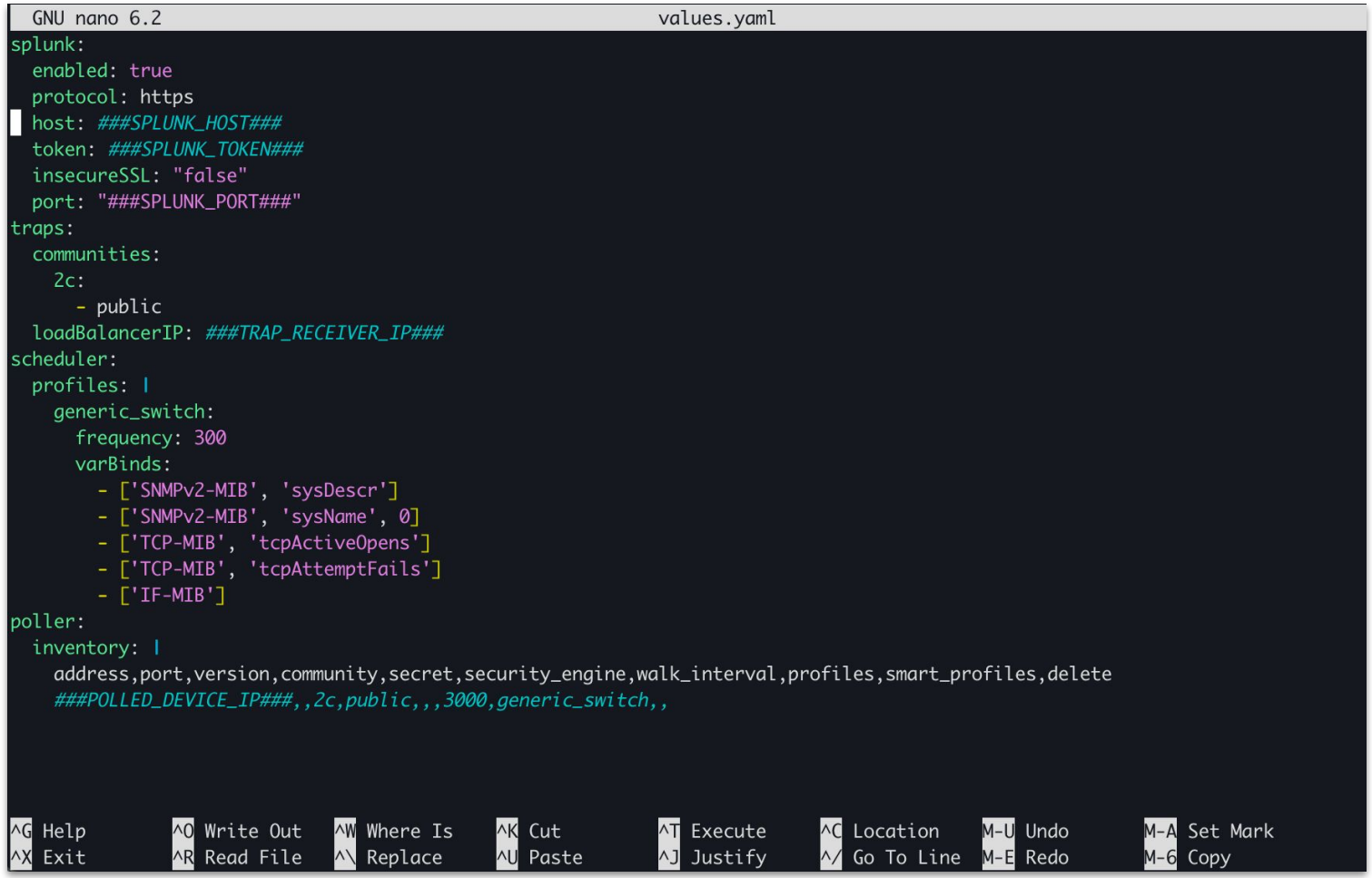
Use the **ls** command to confirm that **values.yaml** exists in the **/home/ubuntu** directory:

```
ls
```

```
root@ip-172-31-31-149:/home/ubuntu# ls
values.yaml
```

Edit **values.yaml** by running:

```
nano values.yaml
```



splunk>

Splunk Connect for SNMP Deployment (cont.)

Add the following entries in the yaml file:

```
  GNU nano 6.2                                values.yaml
splunk:
  enabled: true
  protocol: https
  host: 172.31.44.125
  token: d8a2019e-283c-4cc6-9c70-c4b33008b7e6
  insecureSSL: "true"
  port: "8088"
  metricsIndex: "em_metrics"
  eventIndex: "em_logs"
image:
  pullPolicy: "Always"
traps:
  communities:
    2c:
      public
  loadBalancerIP: 172.31.31.149
scheduler:
  profiles: |
    generic_switch:
      frequency: 60
      varBinds:
        - ['SNMPv2-MIB', 'sysDescr']
        - ['SNMPv2-MIB', 'sysName']
        - ['TCP-MIB', 'tcpActiveOpens']
        - ['TCP-MIB', 'tcpAttemptFails']
        - ['IF-MIB']
poller:
  inventory: |
    address,port,version,community,secret,security_engine,walk_interval,profiles,smart_profiles,delete
    172.31.27.99,161,2c,public,,,3000,generic_switch,,
    172.31.27.99,163,2c,public,,,3000,generic_switch,,
    172.31.27.99,164,2c,sc4snmp,,,3000,generic_switch,,
    172.31.27.99,165,2c,huawei,,,3000,generic_switch,,
                                [ Read 33 lines ]
^G Help      ^O Write Out   M-A Set Mark   M-] To Bracket
^X Exit      ^R Read File   M-6 Copy       ^Q Where Was
```

splunk:
host: <IP address of your Splunk instance>
token: <SC4SNMP HEC token>
insecureSSL: true (disregards self signed cert)
port:
metricsIndex: em_metrics
eventIndex: em_logs

traps:
loadBalancerIP:

poller:

```
address,port,version,community,secret,security_engine,walk_interval,profiles,smart_profiles,delete
<instanceIP>,161,2c,public,,,3000,generic_switch,,
<instanceIP>,163,2c,public,,,3000,generic_switch,,
<instanceIP>,164,2c,sc4snmp,,,3000,generic_switch,,
<instanceIP>,165,2c,huawei,,,3000,generic_switch,,
```

Note: Use **ctrl+O** to save changes and **ctrl+X** to exit from nano

splunk>

Splunk Connect for SNMP Deployment (cont.)

Apply the updated **values.yaml** by running the following command:

```
microk8s helm3 upgrade snmp -f values.yaml splunk-connect-for-snmp/splunk-connect-for-snmp
--namespace=sc4snmp --create-namespace
```

```
root@ip-172-31-31-149:/home/ubuntu# microk8s helm3 upgrade  snmp -f values.yaml splunk-connect-for-snmp/splunk-connect-for-snmp --namespace=sc
4snmp --create-namespace
Release "snmp" has been upgraded. Happy Helming!
NAME: snmp
LAST DEPLOYED: Sun Jun 25 07:07:31 2023
NAMESPACE: sc4snmp
STATUS: deployed
REVISION: 7
NOTES:
Walk profiles no longer include IF-MIB family by default.
If you've used this functionality before, please update the walk profile with ['IF-MIB'] varBind.
root@ip-172-31-31-149:/home/ubuntu#
```

Your helm chart has been upgraded!

splunk>

Splunk Connect for SNMP Deployment (cont.)

Check if all SC4SNMP pods are in a running state by running the following command:

```
microk8s kubectl get pods -n sc4snmp
```

```
root@ip-172-31-47-244:/home/ubuntu# microk8s kubectl get pods -n sc4snmp
NAME                                                READY   STATUS    RESTARTS   AGE
snmp-splunk-connect-for-snmp-worker-poller-c77c7d8bb-vmv9b   1/1     Running   0          20m
snmp-splunk-connect-for-snmp-worker-trap-68cb4b789b-gtdfc    1/1     Running   0          20m
snmp-mibserver-847c5bf574-9wjmh                     1/1     Running   0          20m
snmp-splunk-connect-for-snmp-worker-sender-7894685f44-z8l2w  1/1     Running   0          20m
snmp-splunk-connect-for-snmp-worker-trap-68cb4b789b-hf4c2    1/1     Running   0          20m
snmp-splunk-connect-for-snmp-worker-poller-c77c7d8bb-rp428   1/1     Running   0          20m
snmp-splunk-connect-for-snmp-scheduler-d89787bbd-j2qxw       1/1     Running   0          20m
snmp-splunk-connect-for-snmp-trap-6dbd6bccc-4c8w7            1/1     Running   0          20m
snmp-splunk-connect-for-snmp-trap-6dbd6bccc-g4fm8            1/1     Running   0          20m
snmp-mongodb-75b89b595f-n27wr                       2/2     Running   0          20m
snmp-redis-master-0                                 1/1     Running   0          20m
```

splunk>

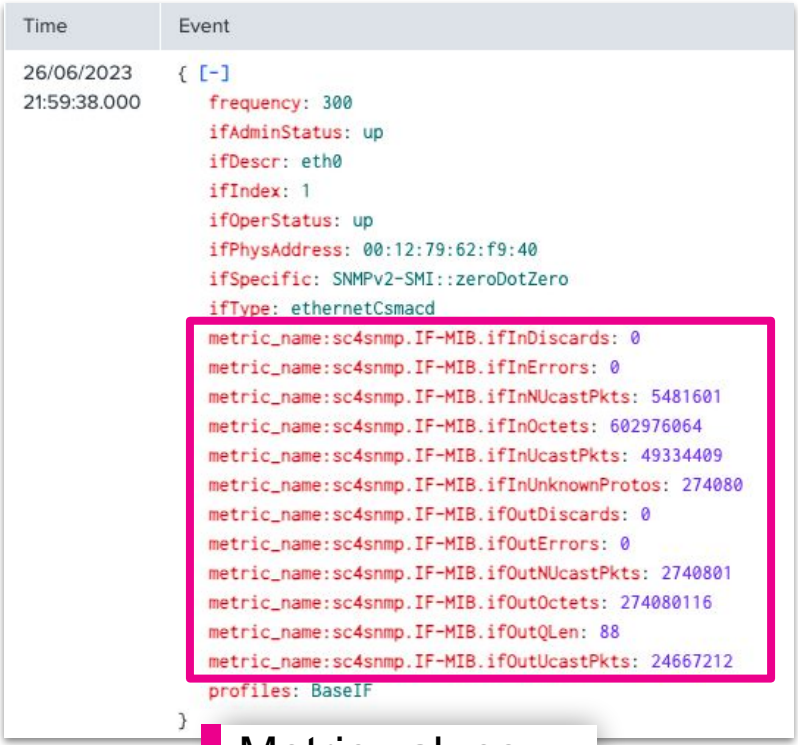
SNMP Indexes: em_logs or em_metrics?

If you recall, the Splunk section of the yaml file had two indexes defined:

```
  GNU nano 6.2                                    values.yaml
splunk:
  enabled: true
  protocol: https
  host: 172.31.44.125
  token: d8a2019e-283c-4cc6-9c70-c4b33008b7e6
  insecureSSL: "true"
  port: "8088"
  metricsIndex: "em_metrics"
  eventIndex: "em_logs"
```

Rule: If an endpoint is polled with a profile with at least one **metric value**, it will go to the **metrics** index

No metrics = **event** index (em_logs)



Metric values = **metrics** index (em_metrics)

splunk>

Explore Event Data

Log onto your splunk instance and open the **Search and Reporting** app. Search for SNMP events by running following search:

```
index=em_logs sourcetype=sc4snmp:event
 | stats latest("SNMPv2-MIB.sysDescr.value") as SysName by host
```

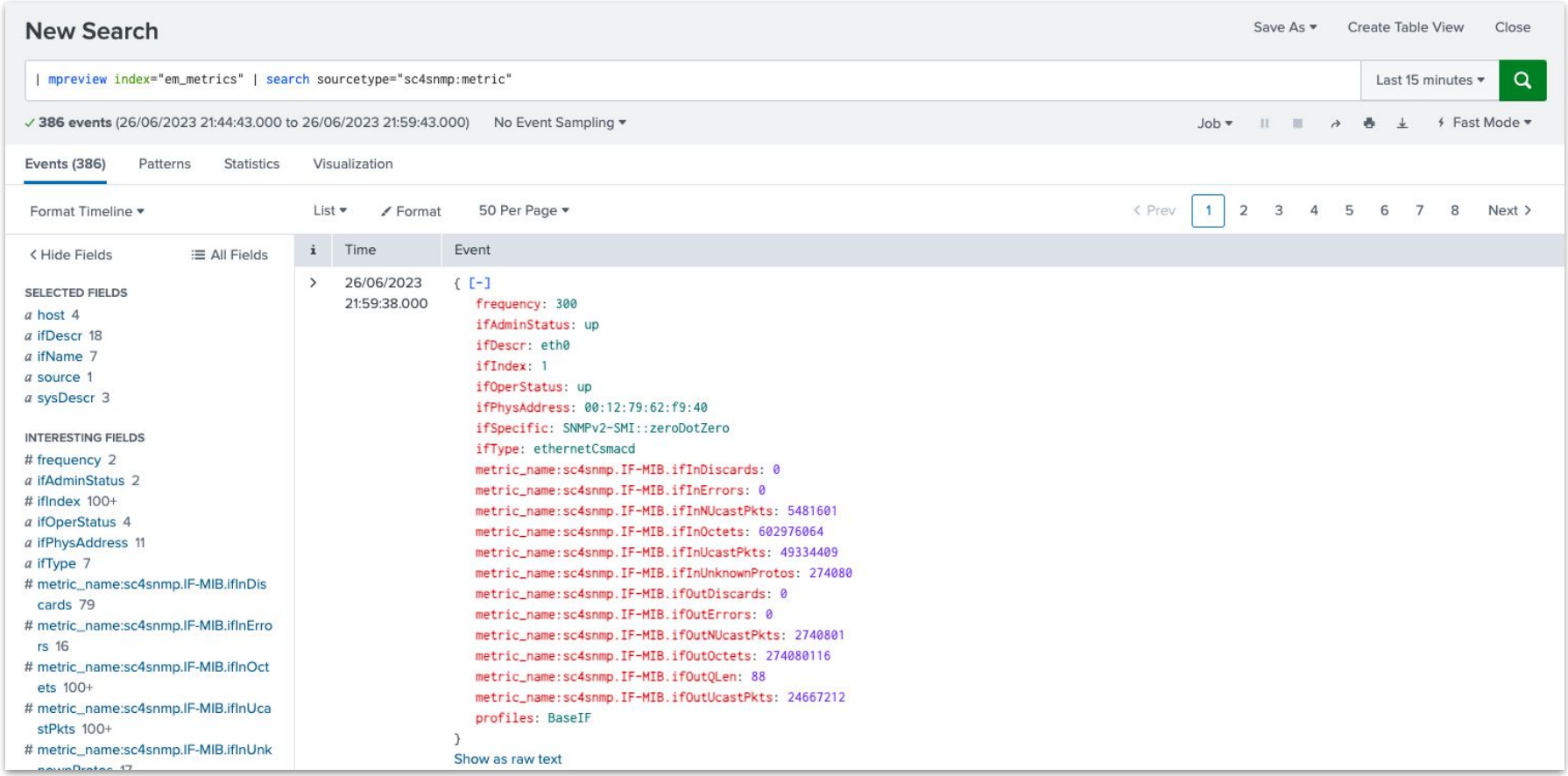
	host	SysName
1	172.31.27.99	Linux ip-172-31-27-99.us-east-2.compute.internal 6.1.29-50.88.amzn2023.x86_64 #1 SMP PREEMPT_DYNAMIC Tue Jun 6 00:47:06 UTC 2023 x86_64
2	172.31.27.99:163	Linux zeus 4.8.6.5-smp #2 SMP Sun Nov 13 14:58:11 CDT 2016 i686
3	172.31.27.99:164	SC4SNMP4NINJAS Switch
4	172.31.27.99:165	Huawei Integrated Access Software

Explore Metric Data

Using Search

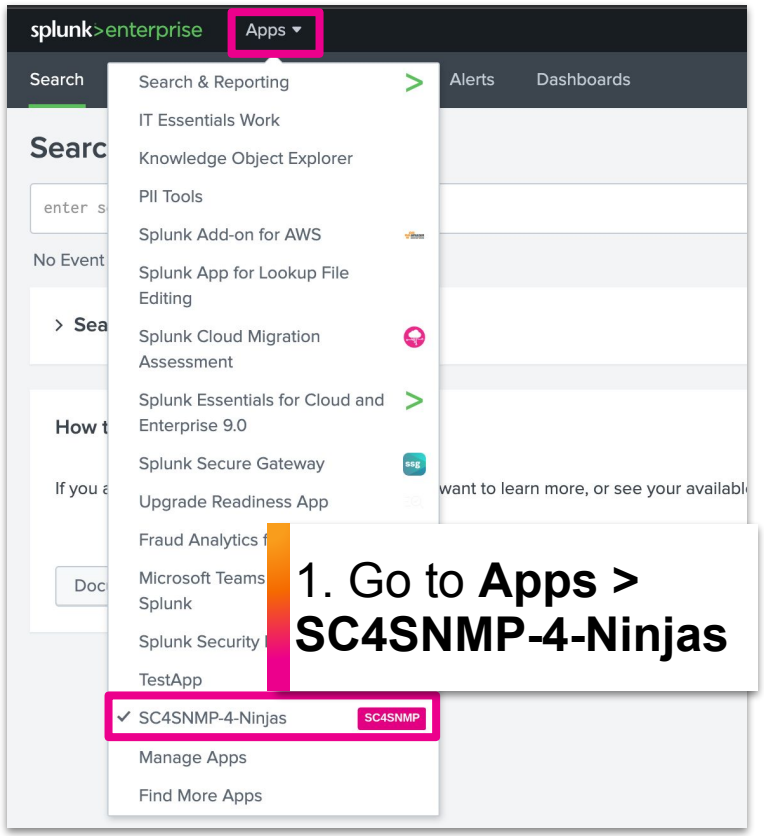
Search for SNMP metrics by running following search:

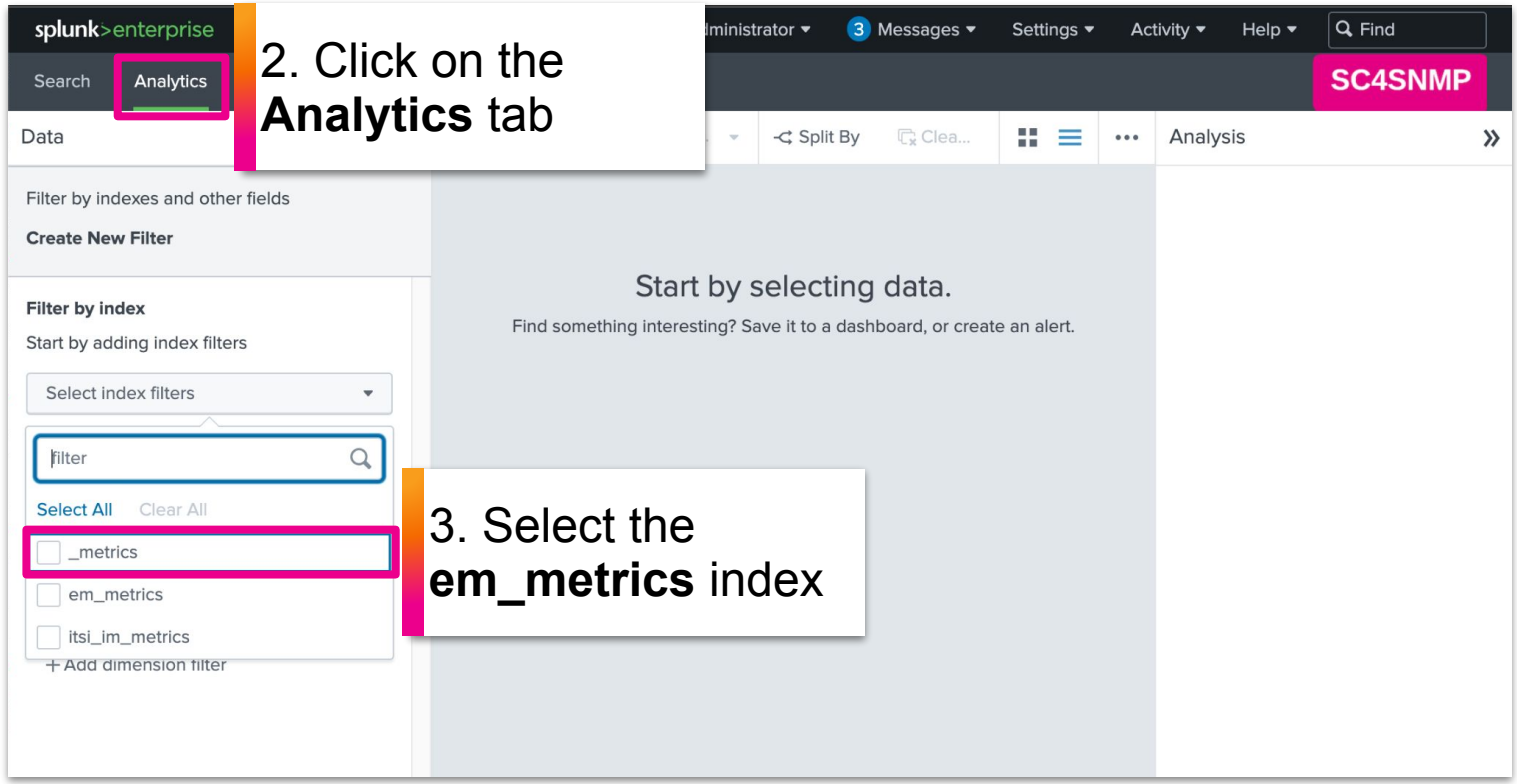
```
| mpreview index=em_metrics | search sourcetype=sc4snmp:metric
```



Explore Metric Data

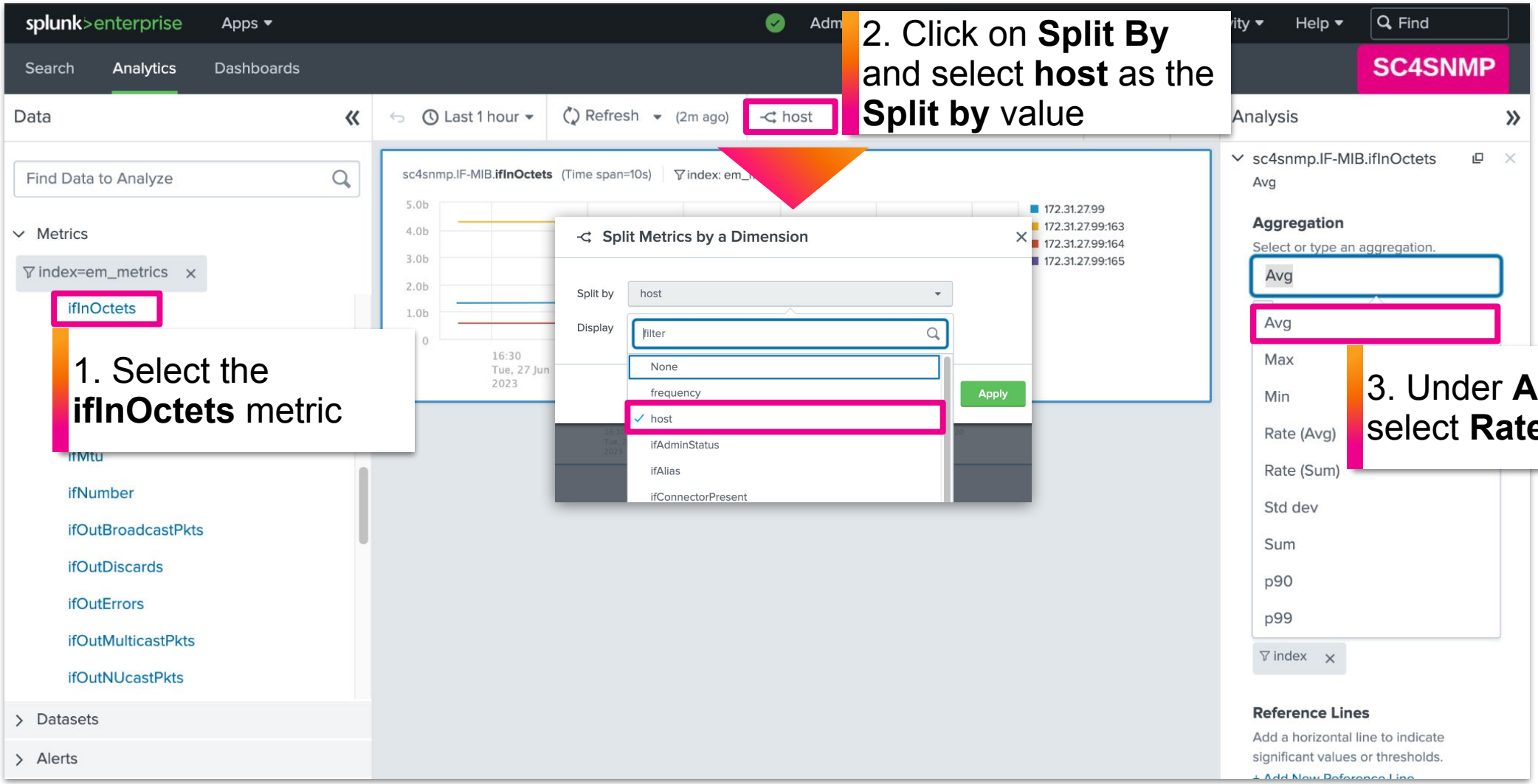
Using Analytics Workspace





Explore Metric Data (cont.)

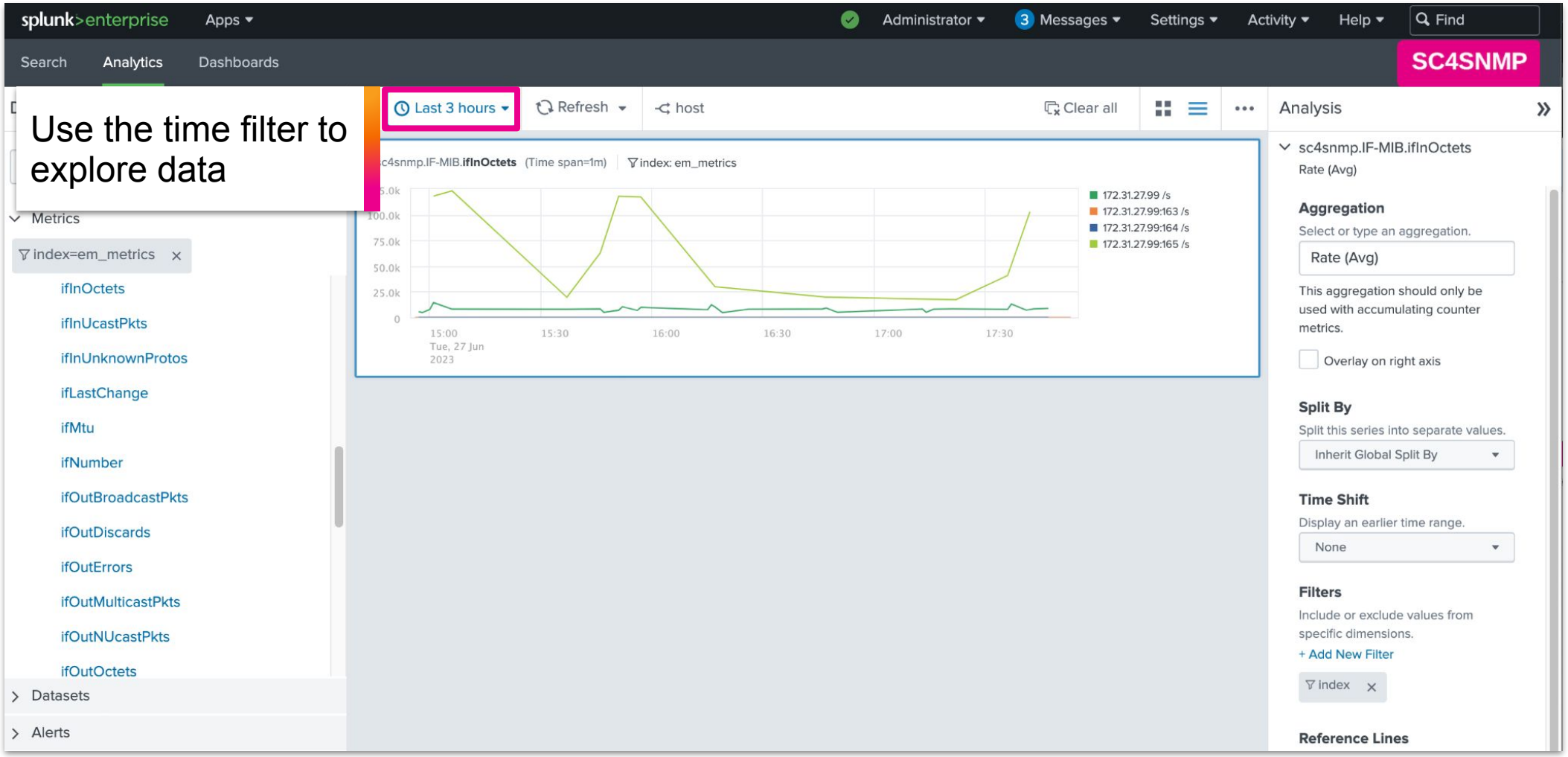
Using Analytics Workspace



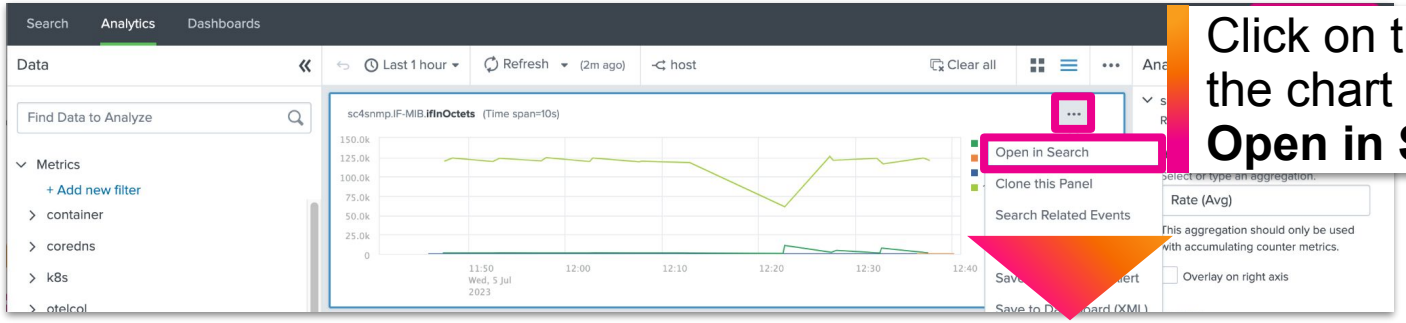
splunk>

Explore Metric Data (cont.)

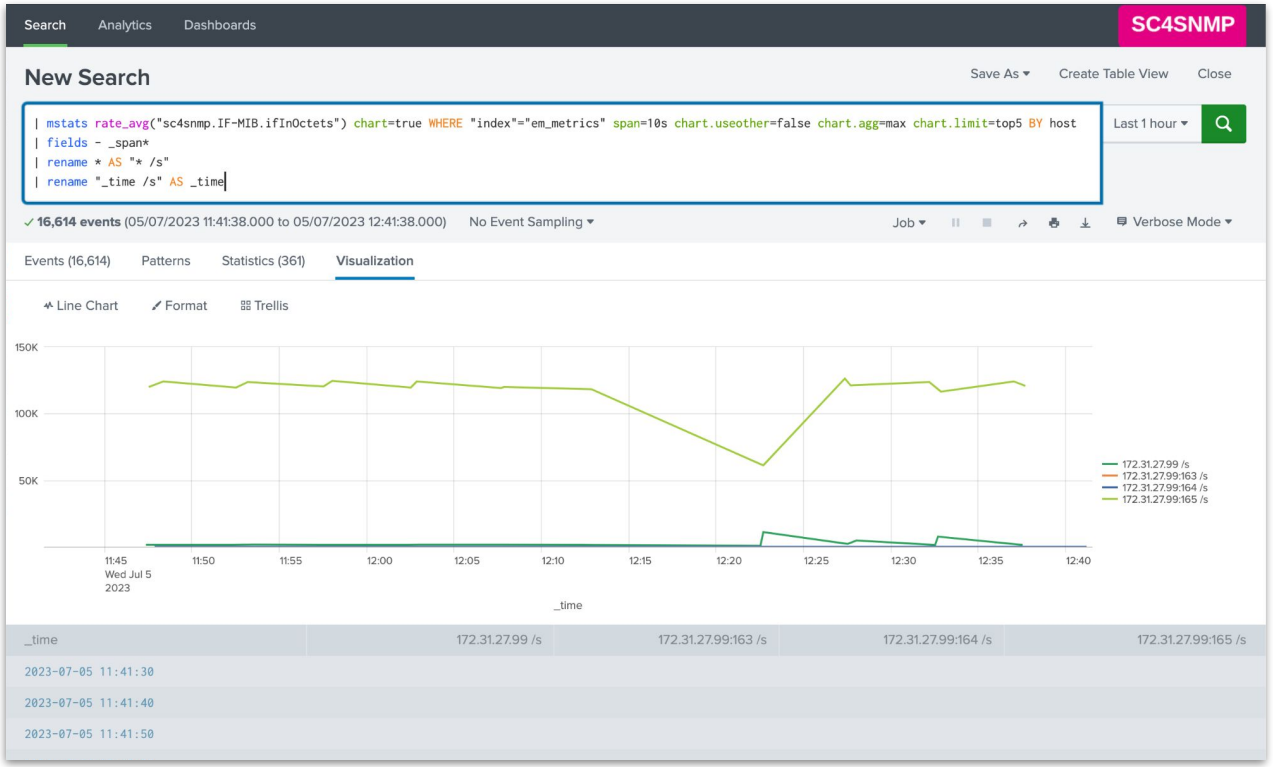
Using Analytics Workspace



Open in Search



Click on the "..." to open the chart menu and select **Open in Search**

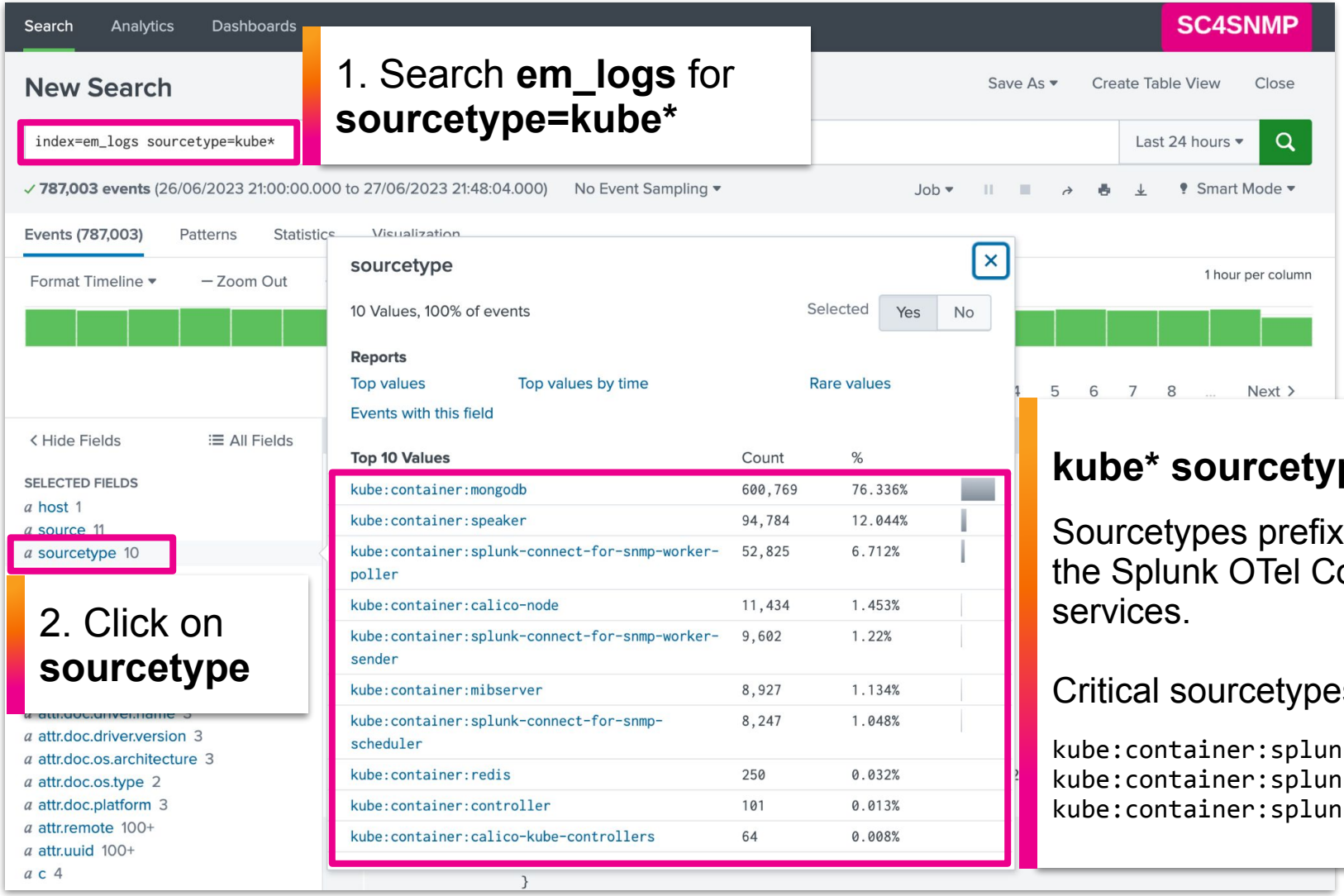



```
| mstats rate_avg("sc4snmp.IF-MIB.ifInOctets")
  chart=true WHERE "index"="em_metrics" span=10s
  chart.useother=false chart.agg=max
  chart.limit=top5 BY host
| fields - _span*
| rename * AS "* /s"
| rename "_time /s" AS _time
```

The search is auto-generated from the chart

splunk>

What Does SC4SNMP Log About Itself?



kube* sourcetypes

Sourcetypes prefixed with **kube*** are everything that the Splunk OTel Collector logs about itself and its services.

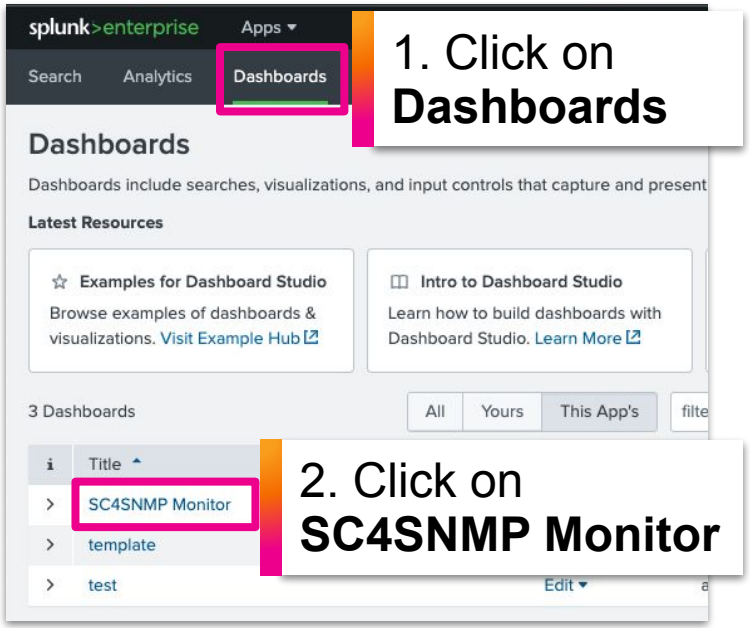
Critical sourcetypes for SC4SNMP are:

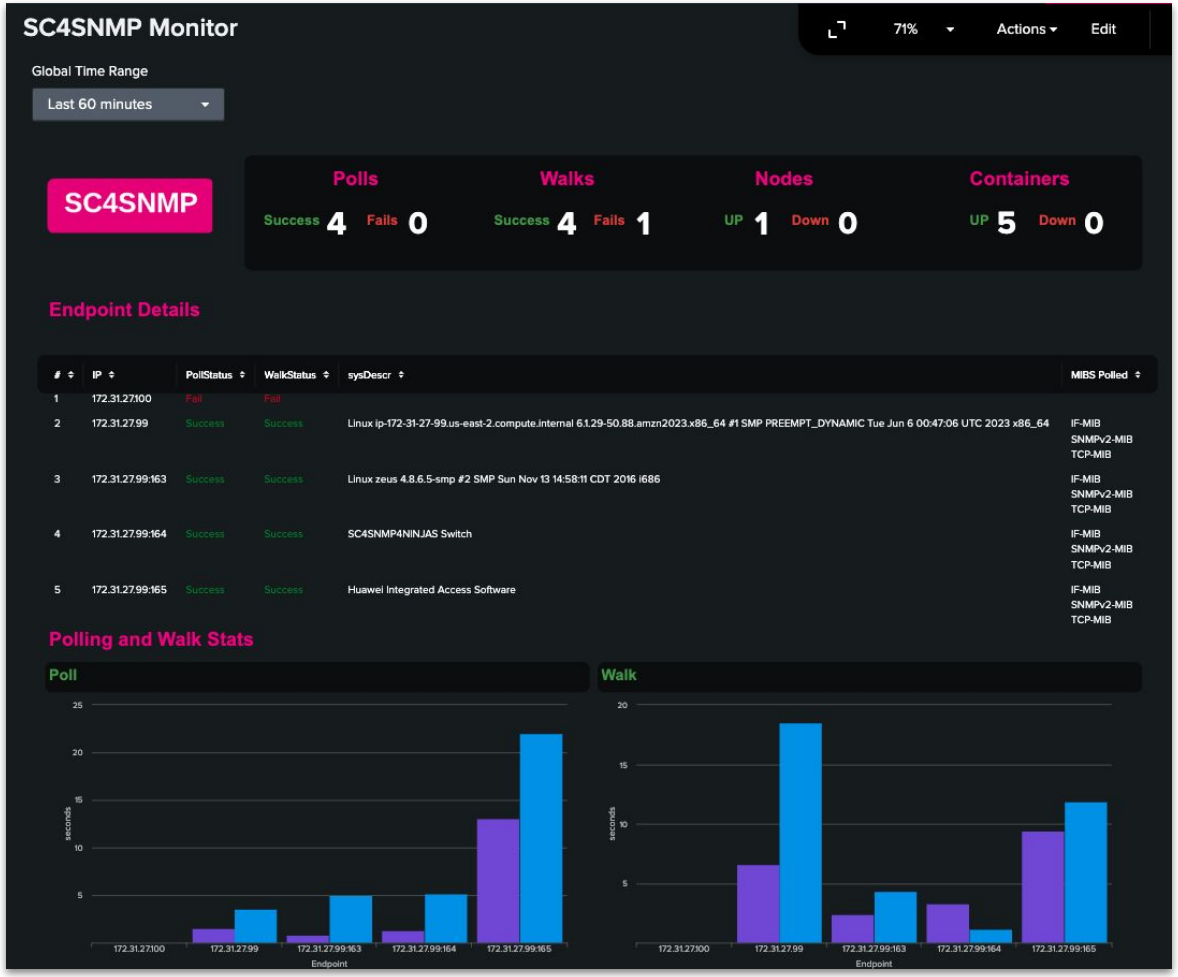
```
kube:container:splunk-connect-for-snmp-worker-poller
kube:container:splunk-connect-for-snmp-worker-sender
kube:container:splunk-connect-for-snmp-worker-scheduler
```

splunk>

SC4SNMP Monitor

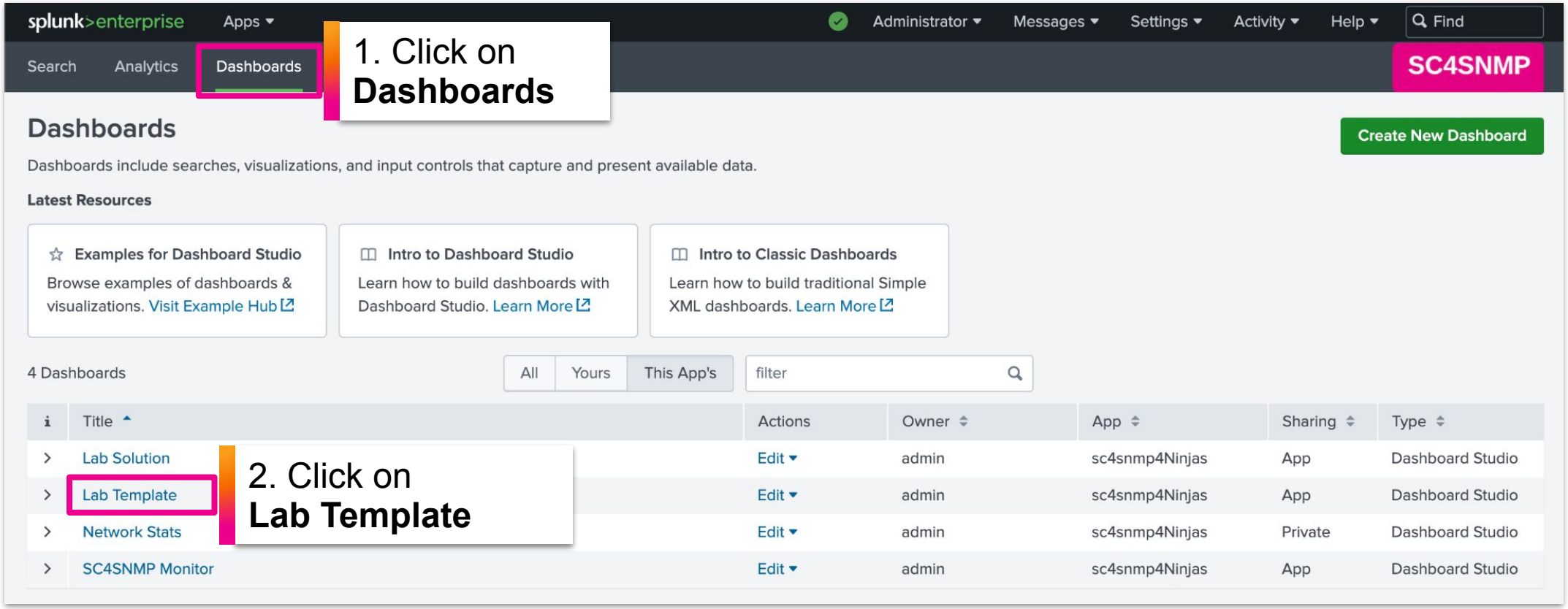
A dashboard designed to monitor SC4SNMP functions, e.g. Polls, Walks and stats, as well as k8s node/container health





splunk>

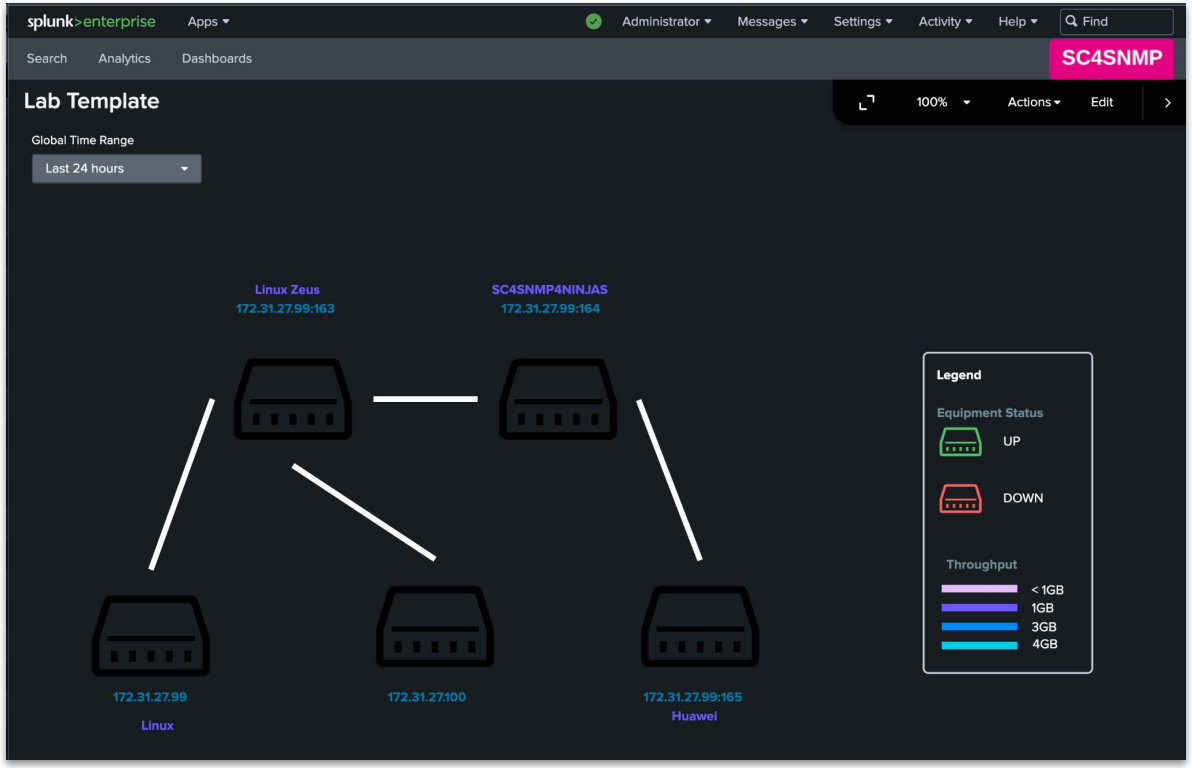
Build Your Network Monitor Lab

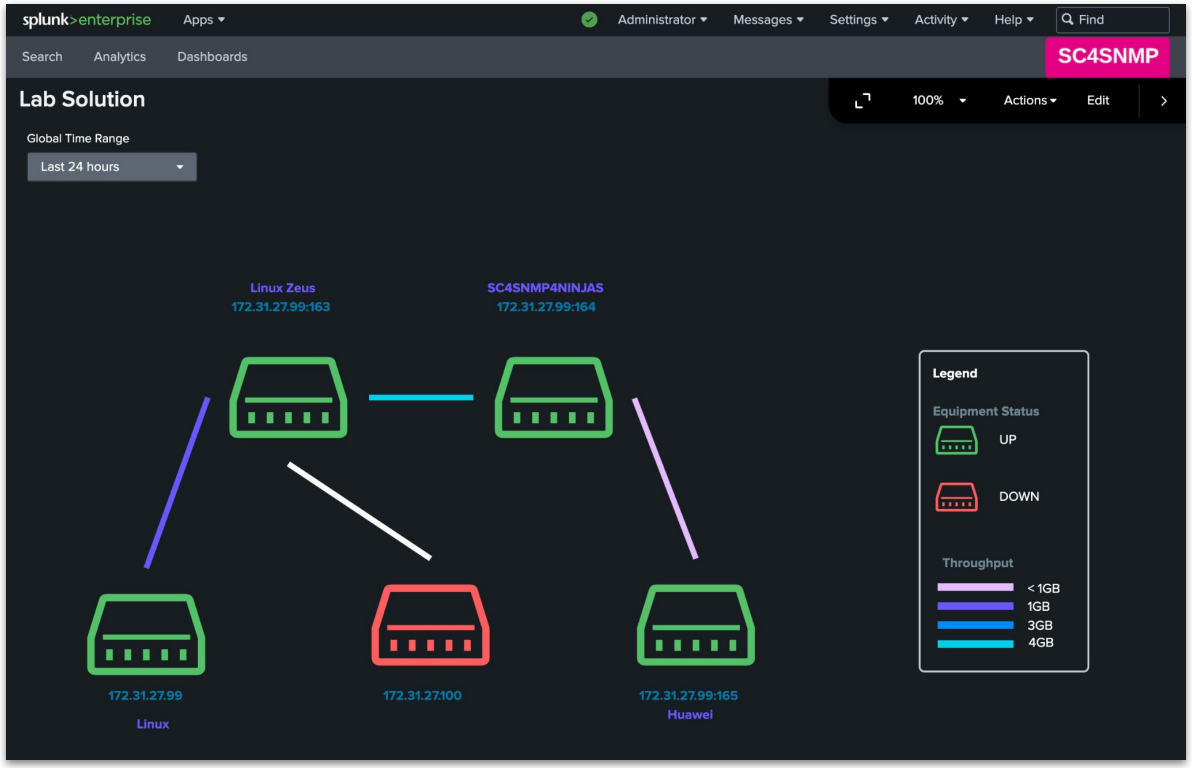


splunk>

Build Your Network Monitor Lab

Lab Goal: Transform the lab template into a functioning dashboard

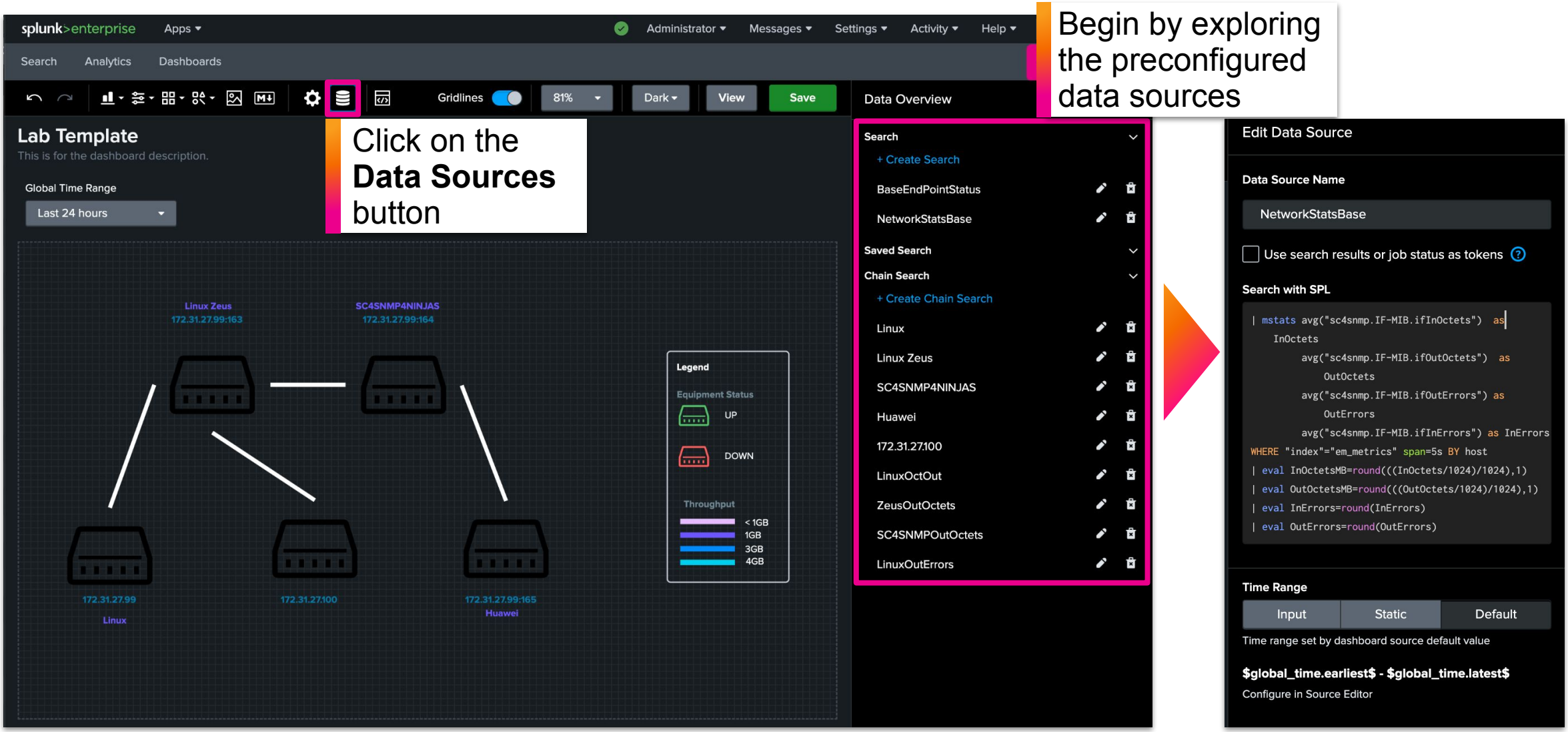




The completed solution can be found in the **Lab Solution** dashboard

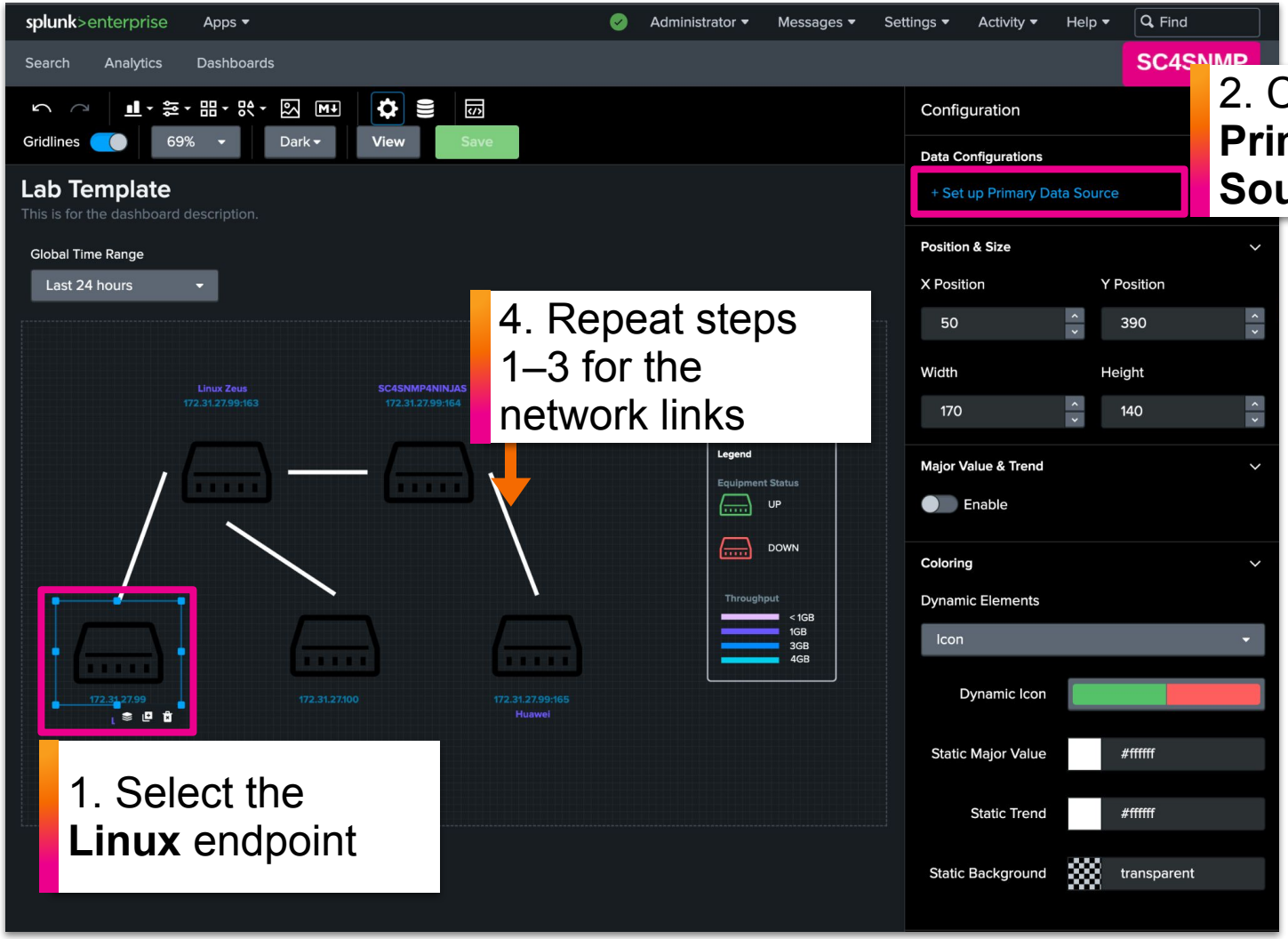
splunk>

Build Your Network Monitor Lab



splunk>

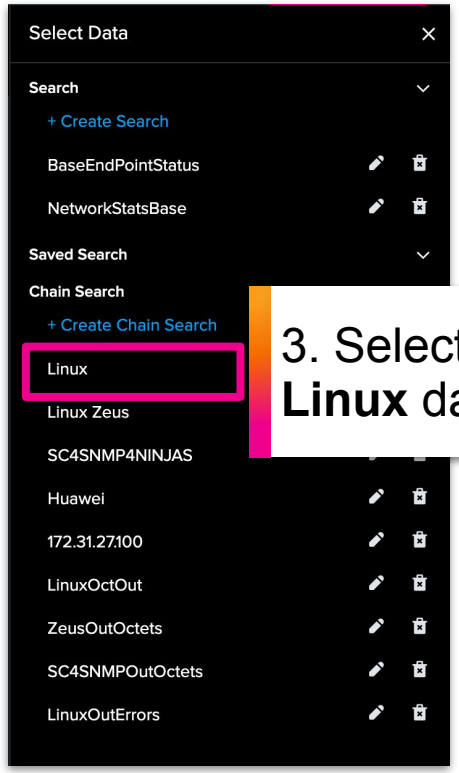
Build Your Network Monitor Lab



1. Select the **Linux** endpoint

4. Repeat steps 1–3 for the network links

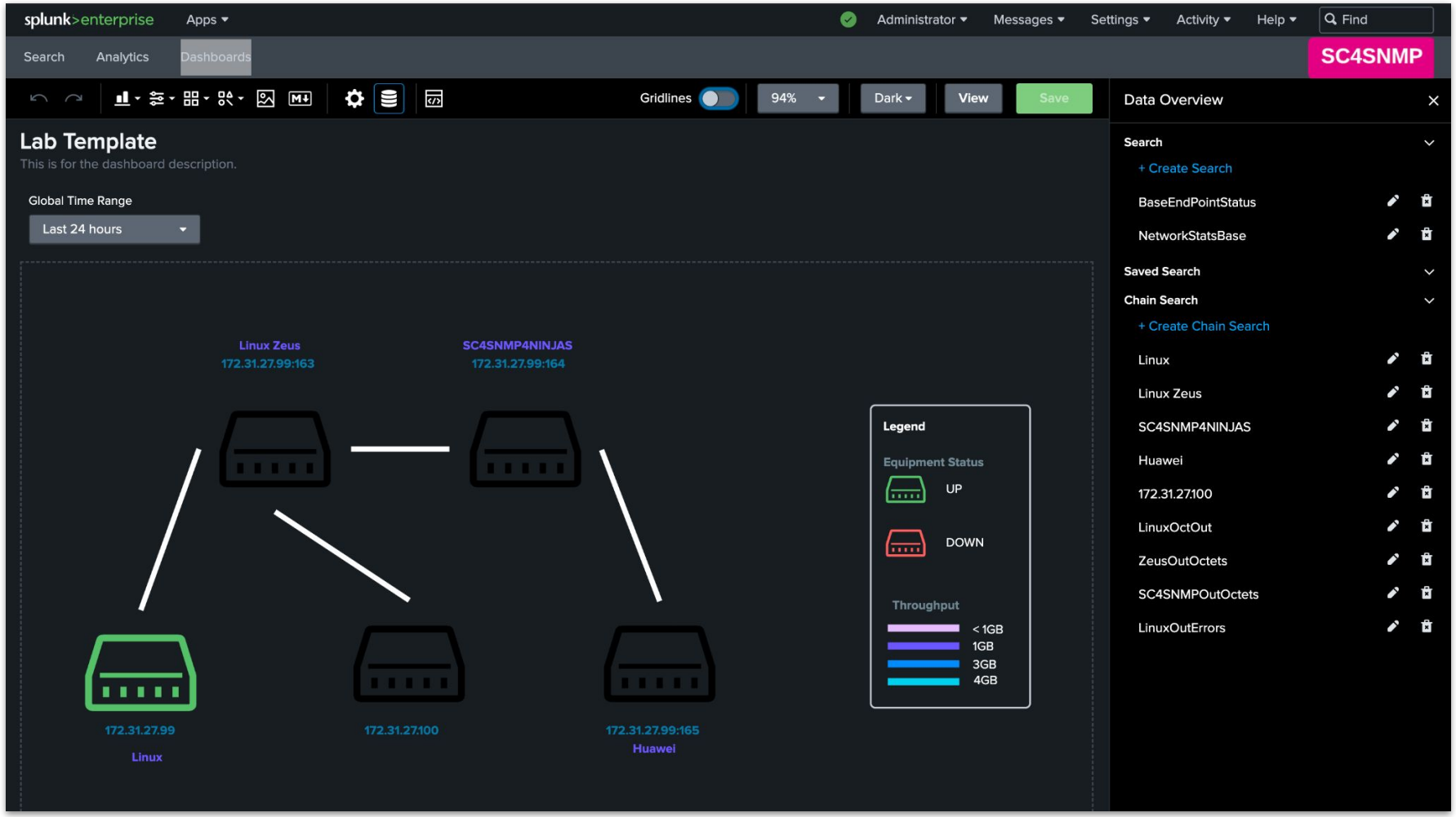
2. Click **+ Set up Primary Data Source**



3. Select the **Linux** data source

splunk>

Build Your Network Monitor Lab: Tips



Use Chain Searches

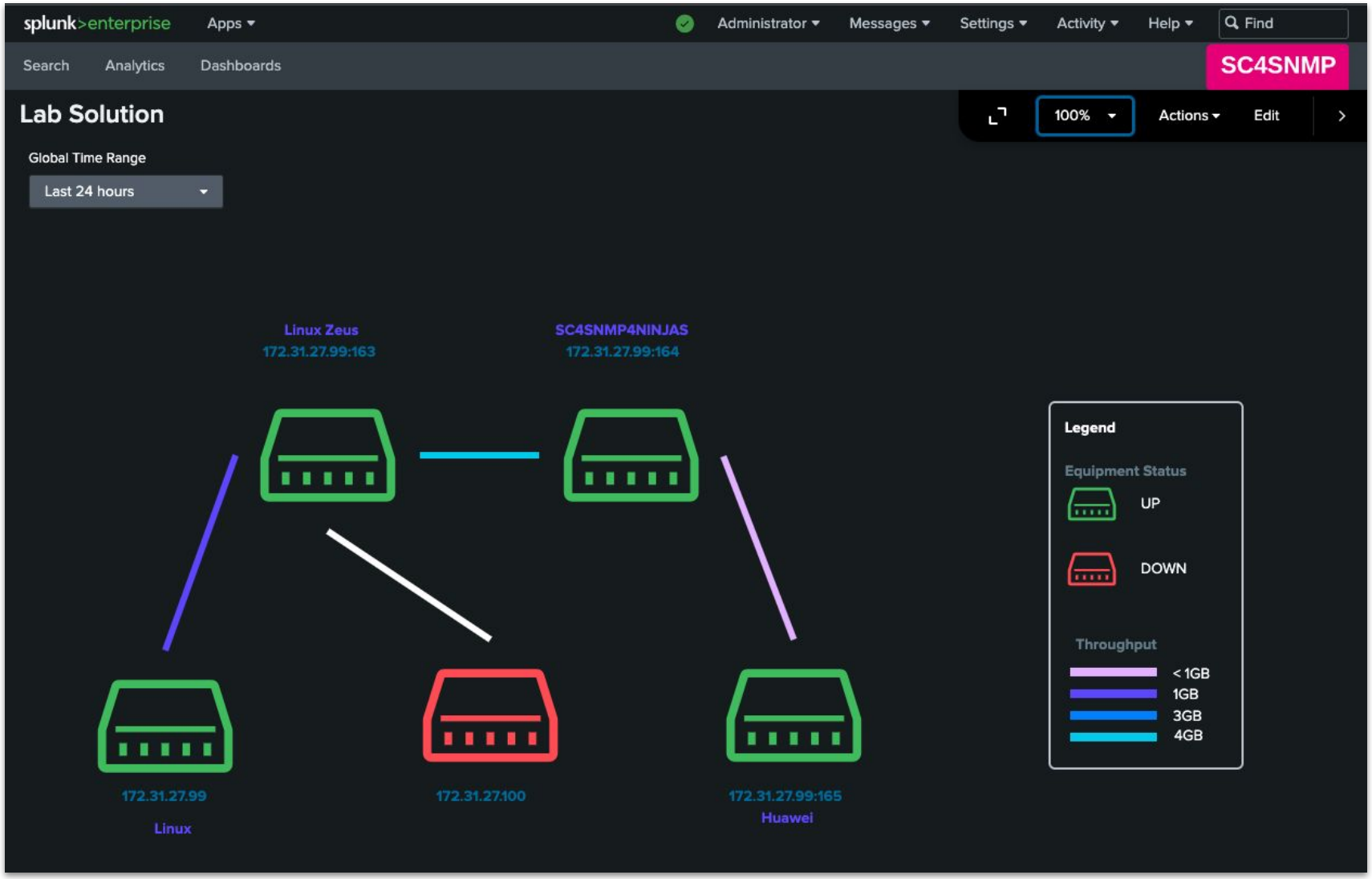
All chain searches have been generated of the two base searches:

BaseEndPointStatus
Leverages k8s logging data for SC4SNMP

NetworkStatsBase
Uses Mstats to query metrics events in em_metrics index

splunk>

Build Your Network Monitor Lab: Solution



Final Solution

The final result should mirror the dashboard on the left.

If not, check the **Lab Solution** dashboard for help!

splunk>

Thank You!

splunk>