

# Splunk Administration

# Introduction

---

Name

Total Experience

Background – Development / Infrastructure / Database / Network

Experience on monitoring tools

Your Expectations from this training

# Machine Data

# Machine Data

---

Data generated by machines, computer processing, applications, databases and sensor data.

Machine data is everywhere and this data is stored in the form of logs every moment.

Millions of data gets produced within a distributed environment.

Machine data is the fastest growing, most complex, most valuable of big data.

# 1. Server Logs

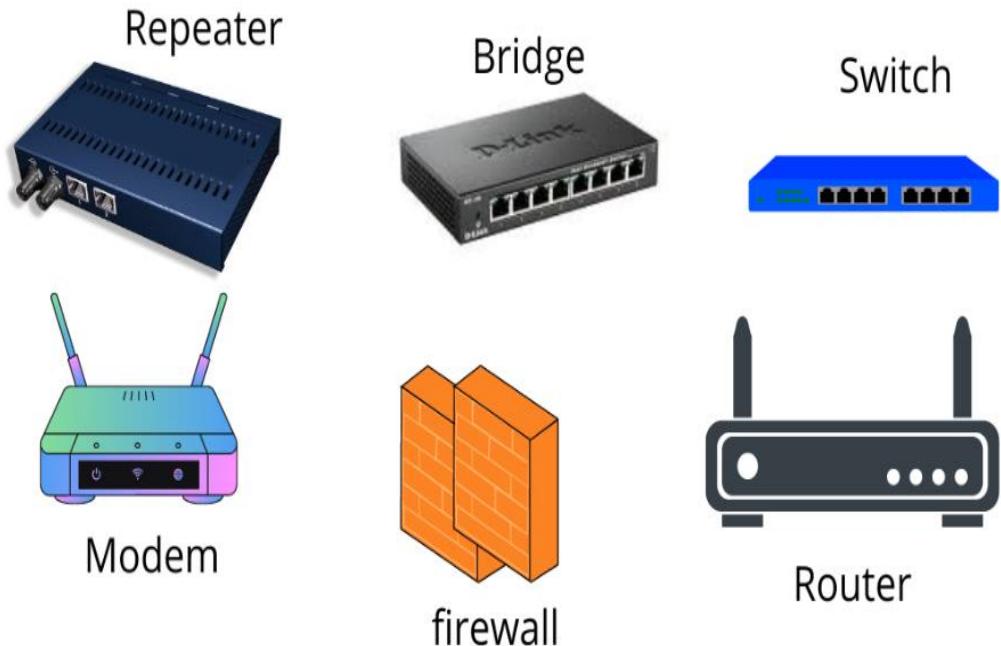
---



- Linux/Windows
- Log files
- Access
- File system

## 2. Network Logs

---



- Firewall
- Warnings
- Alerts
- IP addresses

### 3. Database Logs

---



- Audit Logs
- Configurations
- Schemas
- Tables
- Queries

## 4. Web Logs

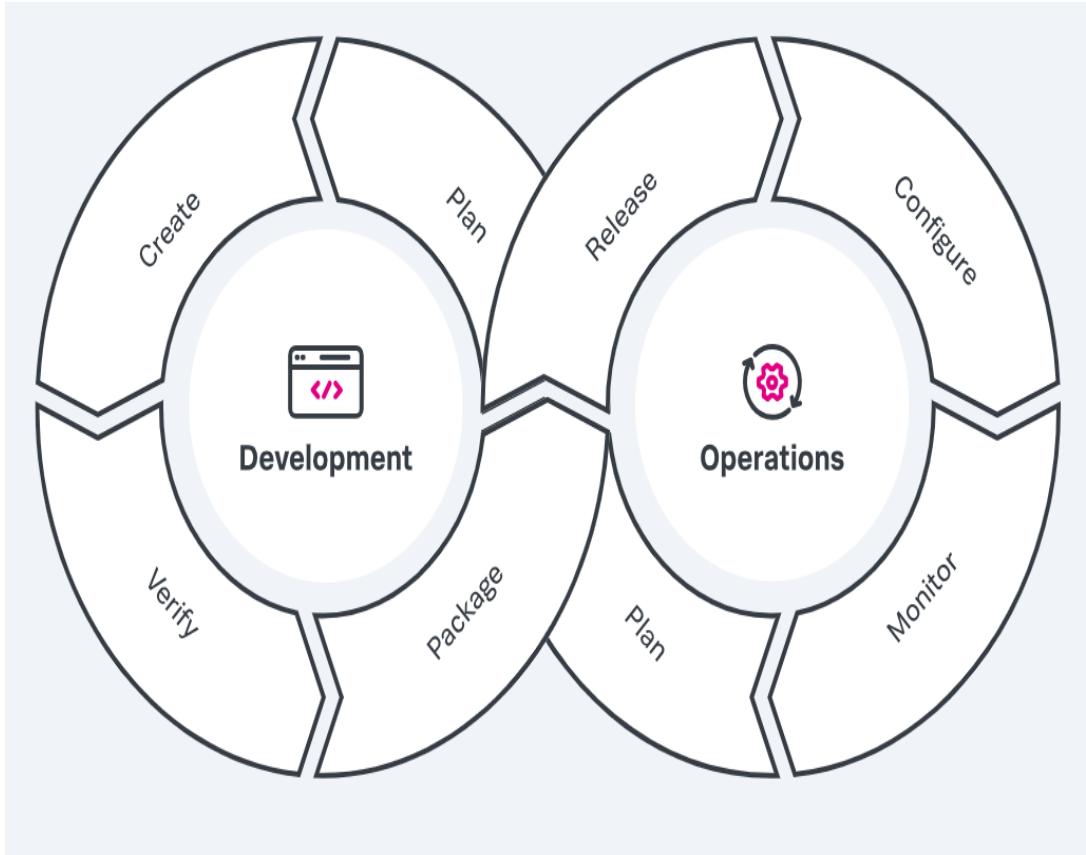
---



- Click Streams
- Location
- Browser
- Transactions
- Time

# 5. DevOps Logs

---



- Test Logs
- Alerts
- Code Check-in
- Event Logs

## 5. IOT Logs

---



- GPS
- Temperature
- RFID
- Biometric
- Limitless

# **What is Splunk?**

# Splunk

---

Splunk makes it simple to collect, analyze and act upon the untapped value of the big data generated by your technology infrastructure, security systems and business applications—giving you the insights to drive operational performance and business results.

Splunk provides **a platform for IT and Government Operations** to gain visibility, insights and intelligence from all machine data

**Strong ecosystem of apps** deliver end-to-end operational visibility enabling IT to reduce costs, consolidate tools and eliminate silos

**Splunk delivers Operational Intelligence** allowing IT to go beyond maintenance to enabling any organizational insights, security and Mission operational improvements

# What is Splunk

---



# Why to choose Splunk for log Analytics

---

- Splunk was designed specifically to address the challenges of collecting, monitoring, analyzing and reporting on machine data.
- The logs that come in any format, from any source, anywhere in your environment.
- Splunk provides huge advantages whether analyzing logs for operational intelligence, security, IoT or other critical use cases.
- Stop issues before they impact your customers
- Rebound faster when the unexpected occurs
- Faster time to market for new apps

# Why to choose Splunk for log Analytics

---

- Over 1000 free apps—many published by your top vendors
- Universal data collection—any amount, any location, any source
- Schema-on-the-fly for true schema-less data ingestion, indexing and storage
- Powerful point-and-click visualizations, analytics and dashboards
- Customize and extend Splunk with your preferred technologies

# Powerful Marketplace

---

Splunk has its own marketplace as Splunkbase where people can submit their applications or addons.

This allows customers to use out of the box solution for wide variety of use cases.

**Browse Splunk Apps by Category**

 <b>IT Operations</b> 1685 Apps	 <b>Security, Fraud &amp; Compliance</b> 1816 Apps	 <b>Business Analytics</b> 261 Apps
 <b>Utilities</b> 1110 Apps	 <b>IoT &amp; Industrial Data</b> 218 Apps	 <b>DevOps</b> 287 Apps

# More than a Log Monitoring solution

---

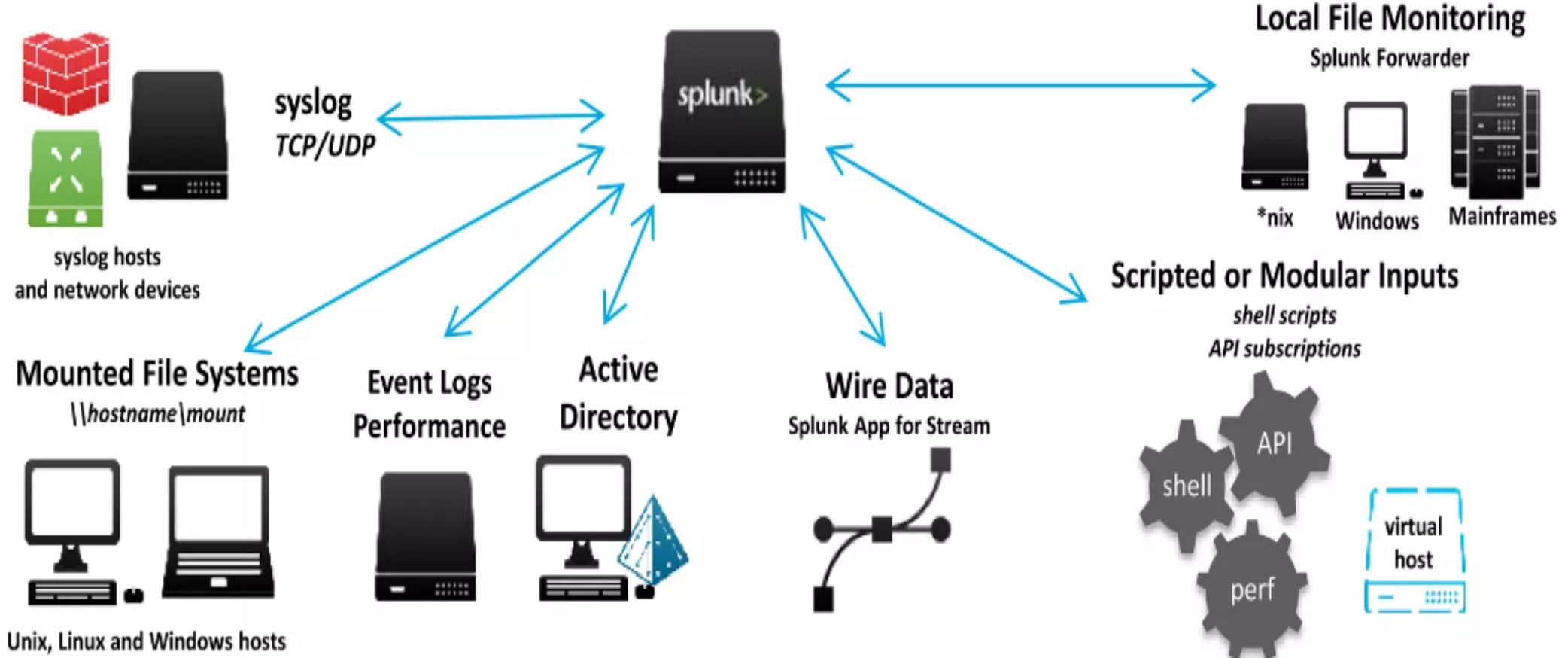
Splunk is a powerful tool which is used to search, analyze and visualize, this makes it more than just a log monitoring solution.

The above feature will help Splunk to promote new apps in various areas like :

- Security information and event management (SIEM)
- Splunk IT Service Intelligence
- Splunk User Behaviour Analytics

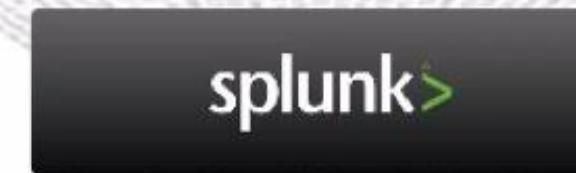


# *Efficient Time Based Indexing*

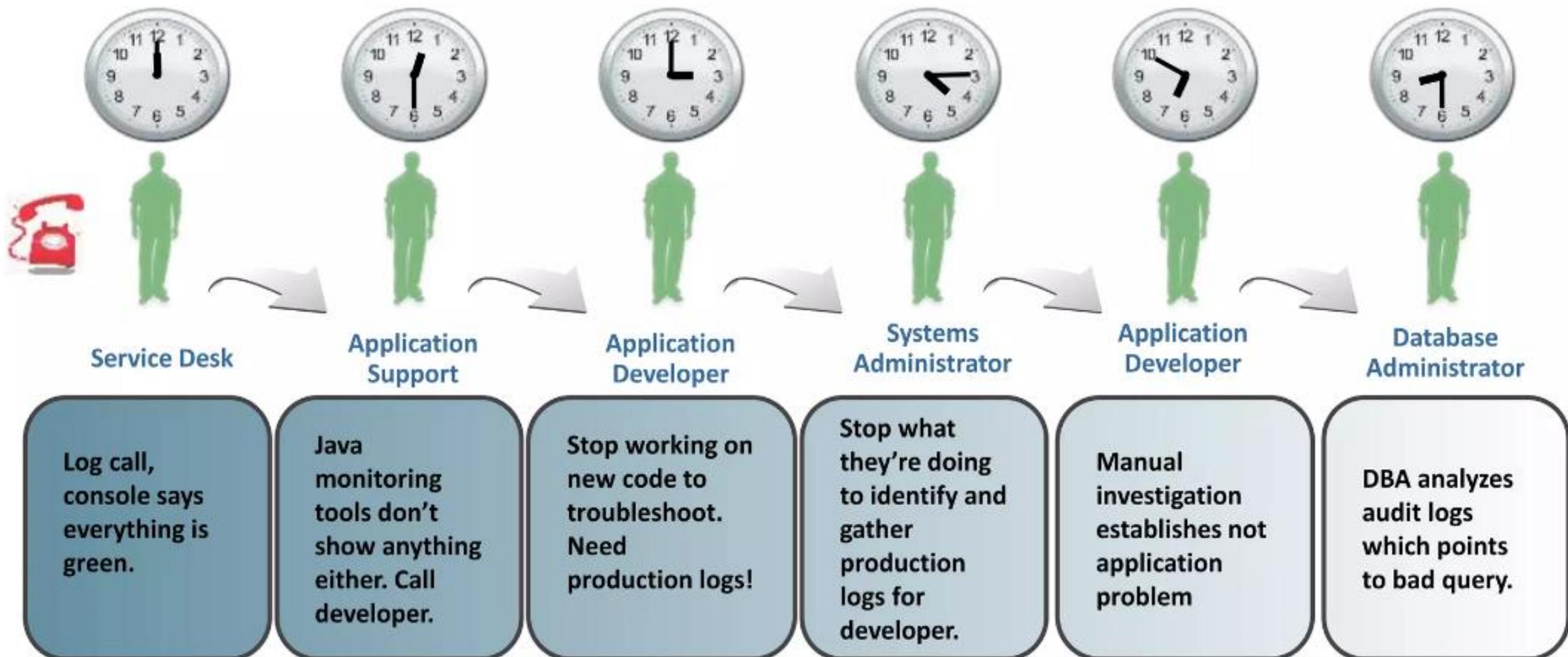


# **INGENST ONCE USE MANY TIMES**

**Reduce Costs: Consolidate tools, eliminate silos, find root cause faster!**



# *Splunk breaking the silos*



# Why it is Important?

---

- Production Monitoring and Debugging
- Resource Usage
- HTTP Errors
- Slow Queries
- Rouge Automated Robots
- Security Issues
- Getting more values out of network and security infrastructures
- Tracking the visitors on a site or platform

# Why it is Important?

---

- Situational awareness and new threat discovery
- Extracting what is really actionable automatically
- Measuring security
- Compliance the regulations
- Incident response

# Log Analysis: what to look for

---

- Unauthorized access
- Password changes
- Login failures
- New login events
- Malware attacks and detection
- Scanning firewall ports
- DDOS attacks

# Log Analysis: what to look for

---

- File related logs (access or name changes)
- Network devices errors
- Data exported
- Processes stopped or new processes started
- Disconnected events
- File auditing
- User accounts
- Modified registry values

# Log Analysis Best Practices:

---

**Pattern Detection and Recognition** refers to filtering incoming messages based on a pattern book. Detecting patterns is an integral part of log analysis as it helps spot anomalies.

**Log Normalization** is the function of converting log elements such as IP addresses or timestamps, to a common format.

**Classification and Tagging** is the process of tagging messages with keywords and categorizing them into classes. This enables you to filter and customize the way you visualize data.

# Log Analysis Best Practices:

---

**Correlation Analysis** refers to collecting data from different sources and finding messages that belong to a specific event. It helps to make connections between logs since multiple systems record an incident. **For example**, in the case of malicious activity, it allows you to filter and correlate logs coming from your network devices, firewalls, servers, and other sources. Correlation analysis is usually associated with alerting systems – based on the pattern you identified, you can create alerts for when your log analyzer spots similar activity in your logs.

**Artificial Ignorance** is a machine learning process that recognizes and discards log entries that are not useful and is used to detect anomalies. When it comes to logging analysis, it means to ignore routine messages generated from the normal operation of the system like regular system updates, thus labeling them as uninteresting. Artificial ignorance alerts you about new and unusual events, even about common events that should have occurred but did not – for example, if a weekly update has failed. These should be investigated.

# Log Analysis

---

## Stage 1



### Collection

Maintain Log Integrity  
Source Identification  
Export Log Files  
Log File Size Reduction

## Stage 2



### Preparation

Log Platform Selection  
Determine Record Structure  
Column Data Types  
Log Tidy/Clean Up

## Stage 3



### Modeling

Investigation Categories  
Investigative Models  
Multivariate Analysis  
Data Relationships

## Stage 4

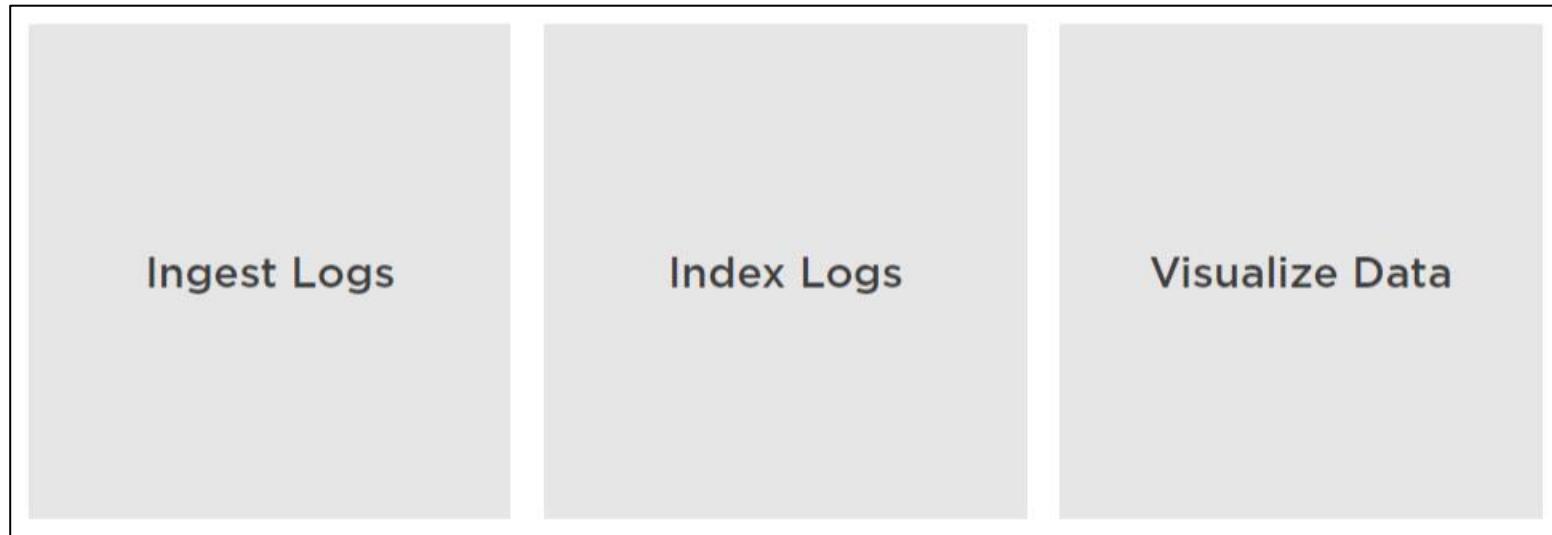


### Report/Presentation

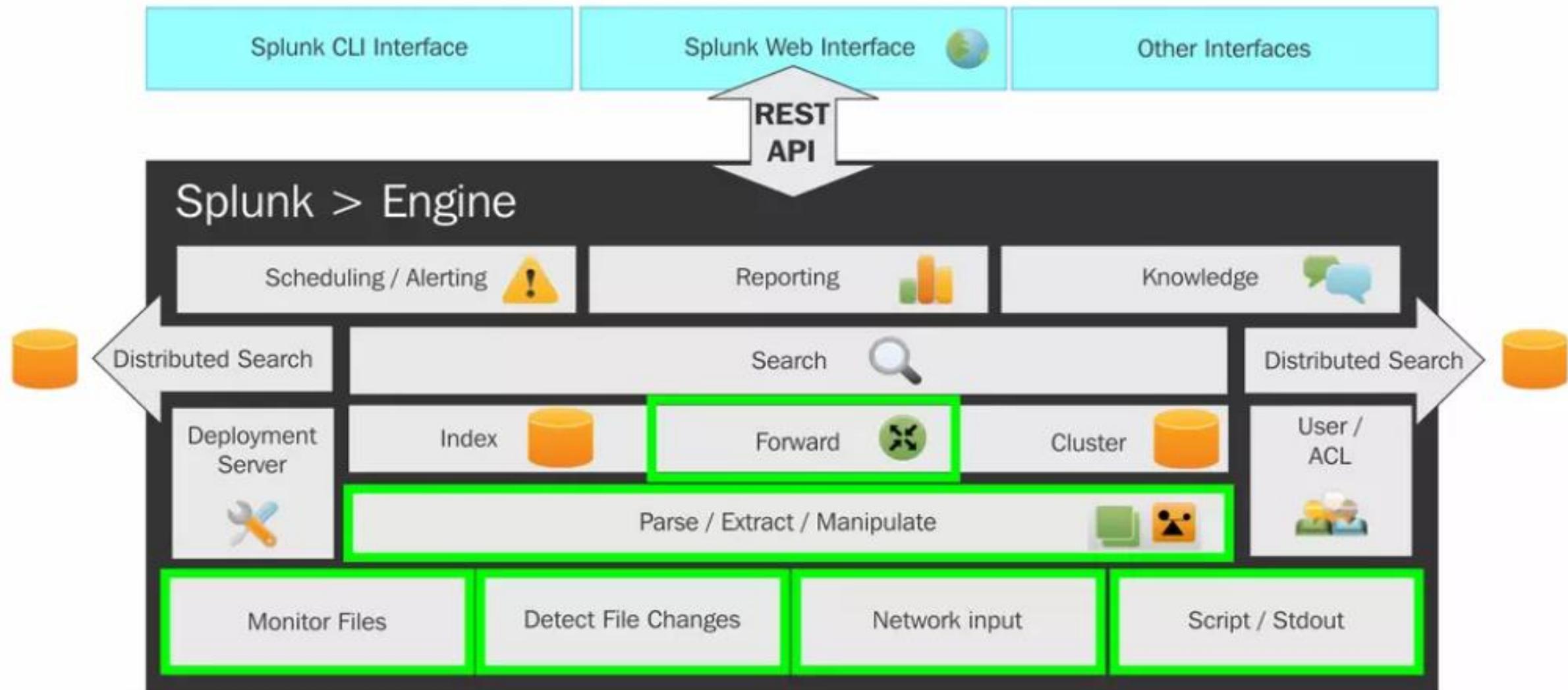
Graphs  
Charts  
Tables  
Interactive Plots

# Splunk Workflow

---



# Splunk Architecture



# Splunk Architecture Components

---

- **Universal Forwarder (UF):** **Splunk Universal Forwarder** is considered a lightweight component that helps in pushing data to the heavy Splunk forwarder. Here, the task of the component is to forward the log data from the server. The **universal forwarder** can also be installed on the client-side or application side.
- **Load Balancer (LB):** The main task of the load balancers is to distribute the workloads over the network or the application traffic over a cluster of servers.
- **Heavy Forwarder (HF):** **Splunk Heavy Forwarder** is acknowledged as a heavy component. It mainly filters the data that is collecting only error logs.

# Splunk Architecture Components

---

- **Indexer:** The indexer stores and indexes the filtered data. It also improves Splunk's performance and automatically implements indexing.
- **Search Head (SH):** It helps in distributing the searches to the other indexers, and is also used to achieve intelligence and perform reporting.
- **Deployment Server (DS):** In the **deployment server** sharing of data is performed between the components. The deployment of the configurations like the update of the UF configuration file plays a main role in the **deployment server in Splunk**.
- **License Master (LM):** A license slave is controlled by a License Master. It is based on quality and usage.

# Splunk Versions

---

Enterprise

Cloud

Light

# Splunk Versions

---

- **Splunk Enterprise:** **Splunk enterprise components** are the paid version with unlimited access to the IT businesses. Its architecture supports single and multi-site clustering for disaster recovery. **Splunk Enterprise** also gathers and analyzes the data from websites, applications, etc.
- **Splunk Cloud:** **Splunk Cloud** is the hosted platform provided as a service with subscription pricing. The features included in this package are similar to the **Splunk enterprise** version. In the architecture, clustering is managed by Splunk.
- **Splunk Light:** **Splunk Light** is the free version with up to 500MB indexing per day. In this version, the features and functionalities are limited as compared to other versions. The architecture supports only a single instance.

# **Splunk Pricing**

# Plans

---

- Workload Pricing
- Ingest pricing
- Entity Pricing
- Activity based pricing

# 1. Workload Pricing

---

This pricing approach is based on the amount of Splunk Virtual Compute (SVC) and storage resources required to run your workloads. It allows you to bring vast amounts of data into Splunk for potential future investigations.

## Workload Pricing



Industry standard, value-oriented metric  
that aligns your spend with search activity

## What you value

### Great if you value:

- Ability to tackle many use cases all in one place
- Tying spend to value
- Industry standard pricing
- Complete control over your usage and infrastructure

## What is it?

### Workload Pricing:

- Frees you up to put all your data in one place to explore more use cases than ever with Splunk
- Provides you control over product expansion — more search vs. indexing more data
- Licensing model is widely used in the industry (AWS, VMware, etc.)
- Similar to your open source vendors

- Increase your data flexibility
- Gain greater control
- Maximize your Splunk ROI

## Workload Types

The Workloads are the activities that you run against data that you have ingested into Splunk to extract insights and analysis.

		GB Ingested Per Day
x	Compliance Storage 1,000 Daily Searches	 1000
	Data Lake (Exploration / Use Case Development)	 1000
	Basic Reporting 50,000 Daily Searches	 1000
	Ad-hoc Investigation 100,000 Daily Searches	 1000
	Continuous Monitoring 300,000 Daily Searches	 1000

# 2. Ingest Pricing

---

This is a traditional model and the pricing is based on how many gigabytes of data are ingested into Splunk products per day. If you need to ingest more data, you can upgrade to the next volume tier.

## Ingest Pricing



Well-known pricing metric – add users, increase activity/searches within your ingest plan to get the maximum value from ingested data

## What you value

### Great if you value:

- Flexibility to tackle use cases in IT, security and more
- Freedom to add users, increase activity and search as much as you want
- Ability to actively administer your search efficiency

## What is it?

### Ingest Pricing:

- Allows you to continue using your current pricing if you have built plans around it; i.e., customers with a very defined / stable data volume needs
- Allows you to get the maximum value from the data you choose to ingest

## Scale up within the confines of your data volume

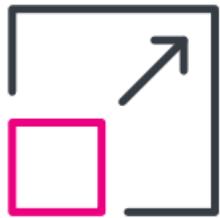
Add users and increase search activity within your ingest plan — all without any additional costs.

# 3. Entity Pricing

---

Tap into a predictable, controllable plan that is based on the number of hosts using Splunk observability products.

## Entity Pricing



Clear, predictable pricing that scales with your business. Align spend to the number of assets in your environment

## What you value

### Great if you value:

- Solving defined Security, ITOps or DevOps use cases with bespoke capabilities
- Ability to easily grow your investment as you find more value in Splunk

## What is it?

### Entity Pricing:

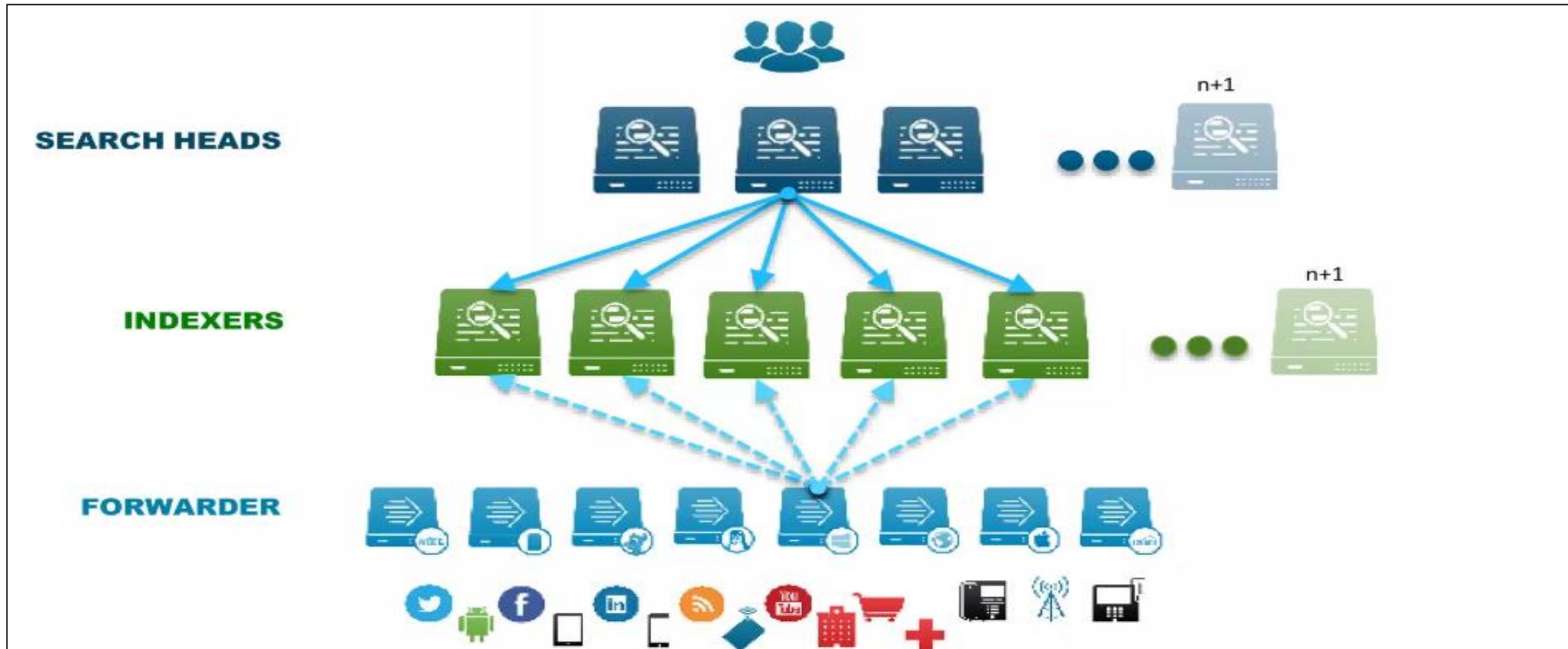
- Allows buying tied directly to the number of assets in your environment
- Assets include protected devices or hosts depending on your product selection
- Allows you to experiment with new capabilities and scale to unlimited data volume with control over your service

## 4. Activity Based Pricing

---

Connect costs directly to activities being monitored by Splunk observability products like metric time series (MTS), traces analyzed per minute, sessions or uptime requests.

# How does Splunk work?

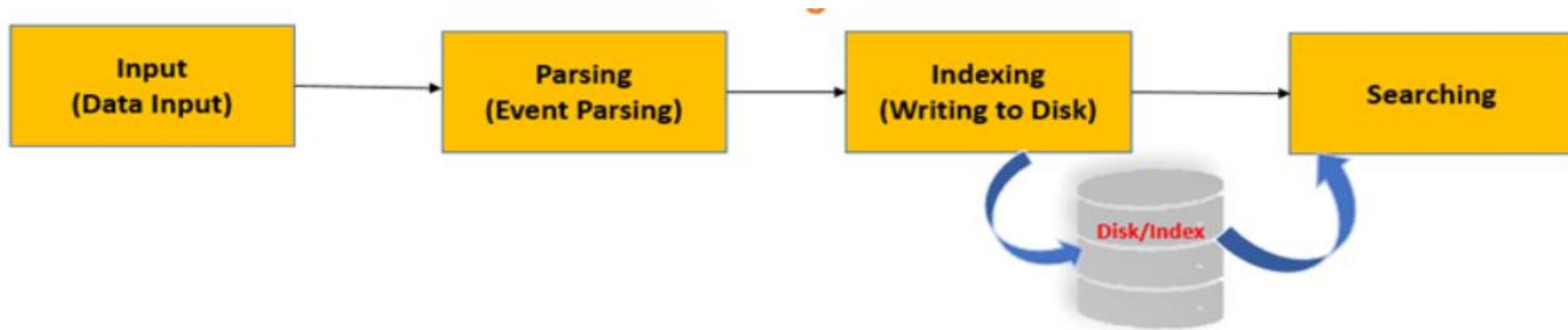


# How does Splunk work?

---

There are 3 different stages in Splunk.

- Data Input stage
- Data Storage stage
- Data Searching stage



# How does Splunk work?

---

## **Input**

Input Data moves to Parsing stage,

## **Parsing**

In Parsing Stage, relevant data is converted into events:

- Customer Region
- Revenue per order
- Time of Order (Morning, Afternoon, Evening, Night)
- A device used by customers (Mobile, PC, Tablet)
- Discount Coupons applied

# How does Splunk work?

---

## **Indexing stage**

In this stage, events are sorted and indexed for storage based on:

- Sales by Geographical location
- Order Revenue
- Time of order (Morning, Afternoon, Evening, Night)
- Device use by the customer
- Coupon offered applied

## **Search Head**

It is used to gain intelligence and perform reporting.

Mac- Donald used it to get the following information:

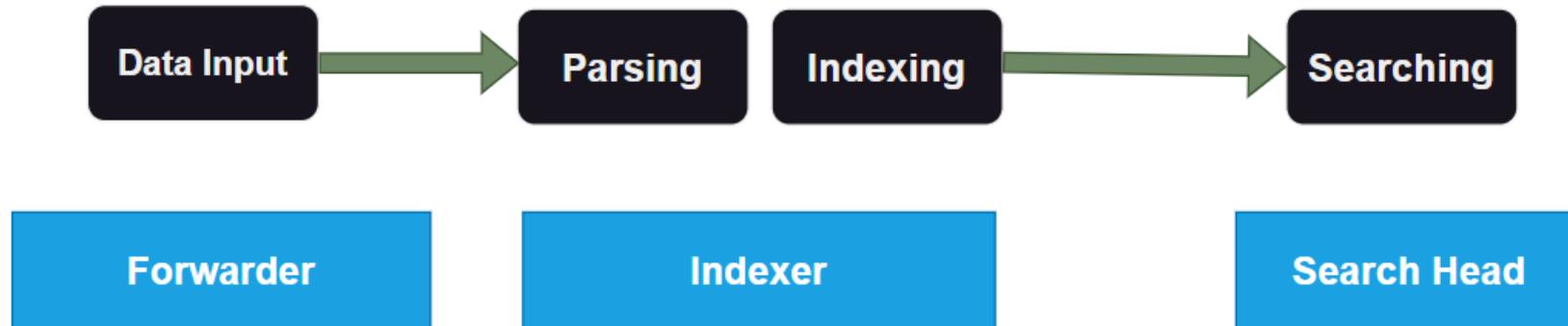
- Which sales offer works best in which geographical location?
- How does customer behavior changes in order revenue?
- What is the best time to apply burger or combo offers?

# Splunk Components

---

Splunk is made up of three major components:

- Splunk Forwarder is a data forwarding tool.
- Splunk Indexer, which is used for data parsing and indexing.
- Search Head is a graphical user interface (GUI) for searching, analyzing, and reporting.



# Splunk Components : Forwarder

---

## **Splunk Forwarder:**

Splunk Forwarder seems to be the component that you must use to collect logs. If you want to collect logs from a remote machine, you can do so by using Splunk's remote forwarders, which are separate from the main Splunk instance.

In reality, users could indeed install multiple forwarders on different machines to forward log data to a Splunk Indexer for processing and storage. What if you want to perform real-time data analysis? Splunk forwarders can also be used for this purpose. The forwarders can indeed be configured to send data to Splunk indexers in real time. You can install them in multiple systems and collect data in real time from multiple machines at the same time.

# Splunk Components : Forwarder

---

- **Universal forwarder:** If you really want to send the raw data collected at the source, you can use a universal forwarder. It is a straightforward component that performs minimal processing on incoming data streams before passing them on to an indexer.

With almost every tool on the market, data transfer is a major issue. Because the data is processed minimally before being forwarded, a large amount of unnecessary data is also forwarded to the indexer, resulting in performance overheads. Why bother transmitting all the information to the Indexers and then filtering out only the relevant information? Isn't it best to send just the necessary data to the Indexer, saving bandwidth, time, and money? It can be fixed by employing heavy forwarders, as explained below.

- **Heavy forwarder:**

You could use a Heavy forwarder to solve half of your problems because one level of data processing occurs at the source before data is forwarded to the indexer. The Heavy Forwarder generally parses and indexes data at the source and intelligently routes it to the Indexer, saving bandwidth and storage space. As a result, when a heavy forwarder parses the data, the indexer only has to deal with the indexing segment.

# Splunk Components : Indexer

---

## **Splunk Indexer:**

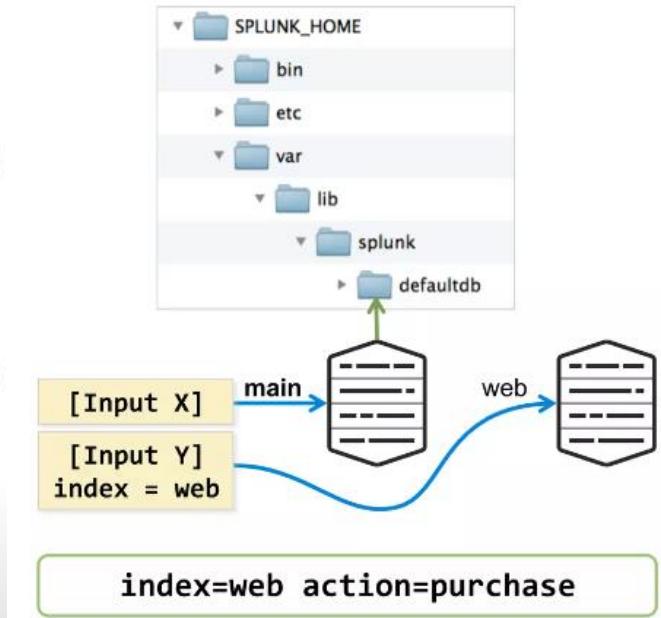
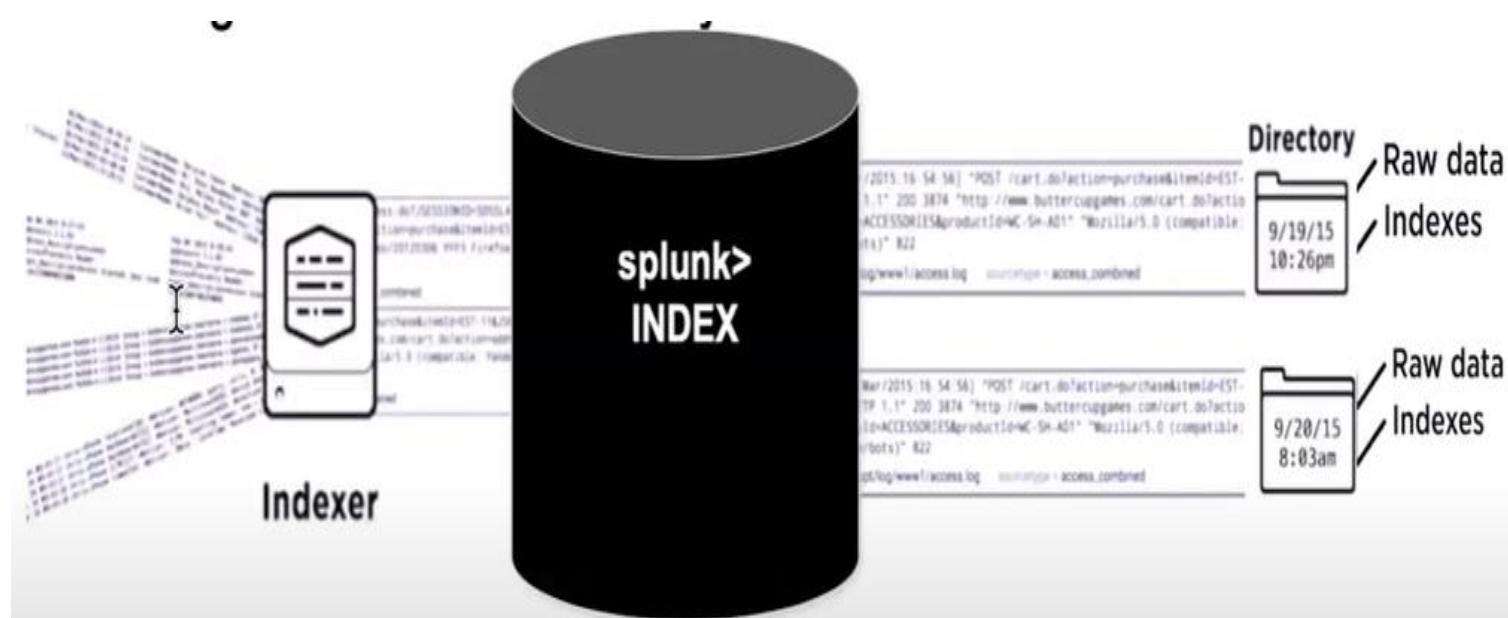
The indexer is the Splunk component that will be used to index and store the data from the forwarder. The Splunk instance converts incoming data into events and stores it in indexes for efficient search operations. If the data is received from a Universal forwarder, the indexer will first parse it before indexing it. Data parsing is used to remove unwanted data. However, if the data is received from a Heavy forwarder, the indexer will only index the data.

**As your data is indexed by Splunk, it generates a number of files. These files contain one or more of the following:**

- 1.Compressed raw data
- 2.Indexes pointing to raw data (index files, also known as tsidx files), as well as some metadata files
- 3.These files are stored in buckets, which are collections of directories.

# Splunk Components : Indexer

An index is a collection of directories and files. These are located under **\$SPLUNK\_HOME/var/lib/splunk**. Index directories are also called buckets and are organised by age



# Splunk Components : Indexer

---

**Splunk processes incoming data to allow for quick search and analysis. It improves the data in a variety of ways, including:**

1. Dividing the data stream into discrete, searchable events
2. Creating or determining timestamps
3. Obtaining information such as host, source, and source-type
4. Performing user-defined actions on incoming data, such as identifying custom fields, masking sensitive data, creating new or modified keys, breaking rules for multi-line events, filtering unwanted events, and routing events to specified indexes or servers.

# Splunk Components : Indexer

---

This **indexing** procedure is also referred to as **event processing**. Splunk keeps multiple copies of indexed data, so you don't have to worry about data loss. This is known as **index replication** or **indexer clustering**.

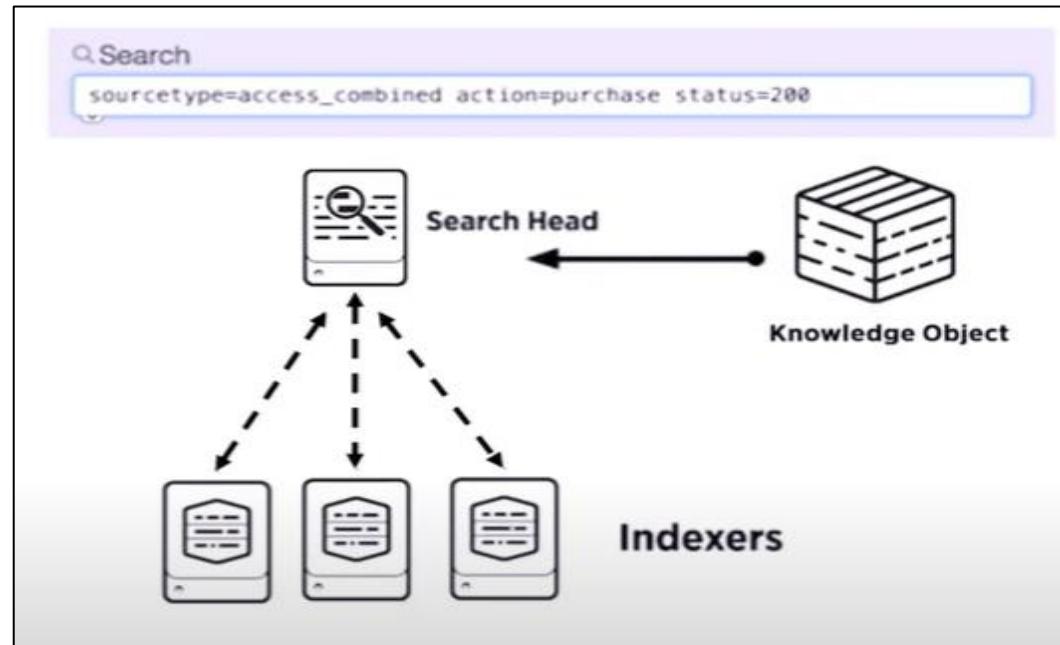
By Default, data you feed to Splunk is stored in the “main” index, but you can create and specify other indexes for Splunk to use for different data inputs.

The files reside in directories organized by age. These directories are called buckets.

# Splunk Components : Search Head

## Splunk Search Head:

The search head seems to be the component that interacts with Splunk. It gives users a graphical user interface through which they can perform various operations. You can search and query the Indexer's data by entering search terms, and you will get the expected results.



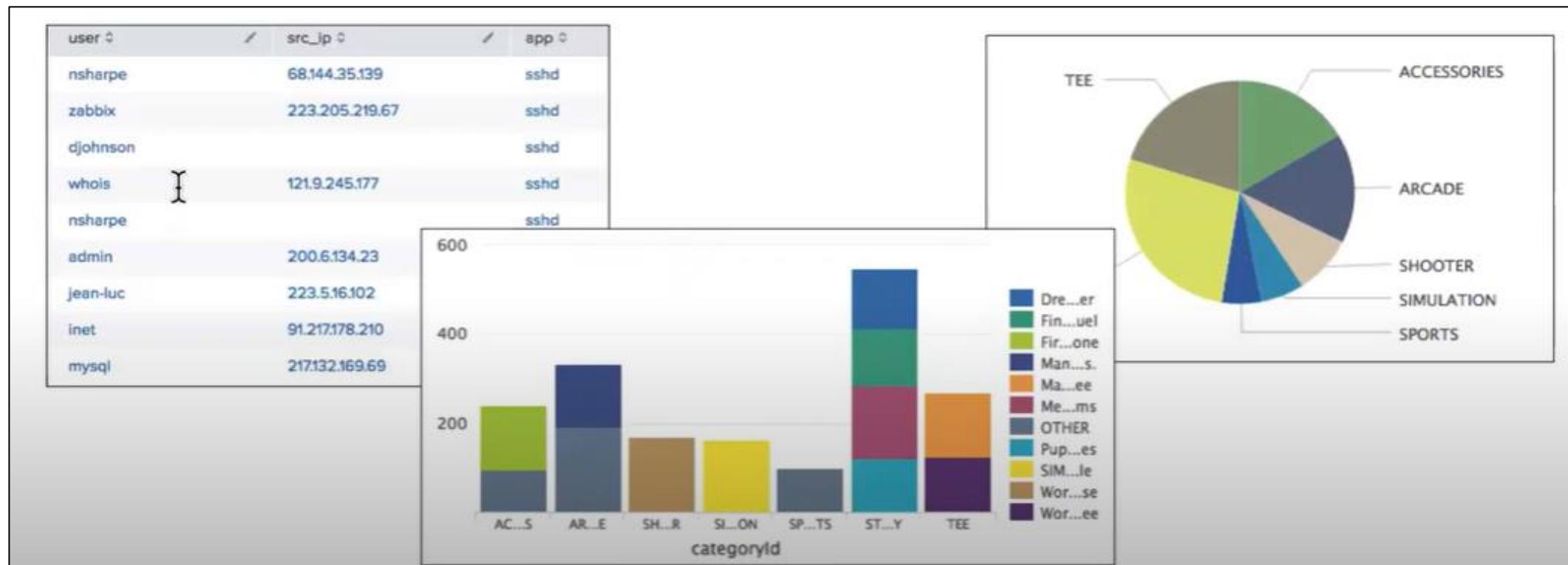
# Splunk Components : Search Head

---

- The search head could be installed on separate servers or on the same server as other Splunk components. There is no separate installation file for the search head; to enable it, simply enable the **splunkweb** service on the Splunk server.
- A search head in a Splunk instance can send search requests to a group of indexers, or search peers, who perform the actual searches on their indexes. The search head then combines the results and returns them to the user. This is a faster data search technique known as **distributed searching**.
- **Knowledge objects** on the Search Heads can be created to extract additional fields and transform the data without changing the underlying index data.

# Splunk Components : Search Head

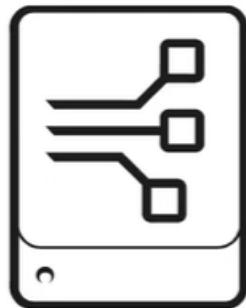
Search Heads provide tools to enhance the search experience such reports, dashboards and visualizations.



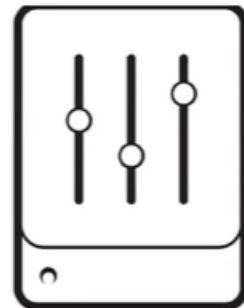
# Additional Splunk components

---

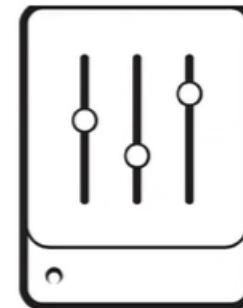
These are less common components:



**Deployment  
Server**



**Cluster Master**



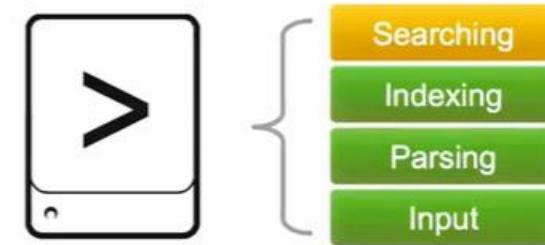
**License Master**

# Splunk Deployment - Standalone

---

## ➤ Single Server

- All functions in a single instance of Splunk
- For testing, proof of concept, personal use and learning
- this is what you get when you download Splunk and install with default setting



## ➤ Recommendation

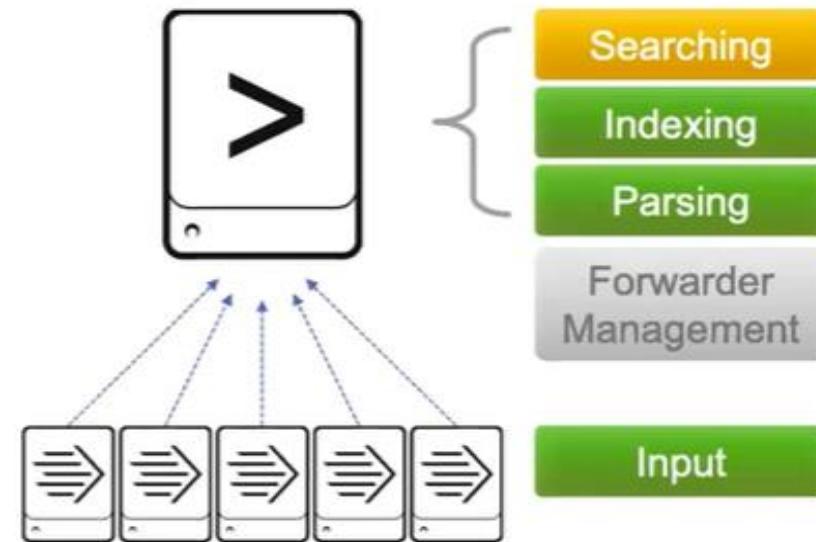
- Have at least one test/development setup at your site

# Splunk Deployment - Basic

- **Splunk server**
  - Similar to server in standalone configuration
  - Manage deployment of forwarder configurations

- **Forwarders**

- Forwarders collect data and send it to Splunk servers
- Install forwarders at data source (usually production servers)

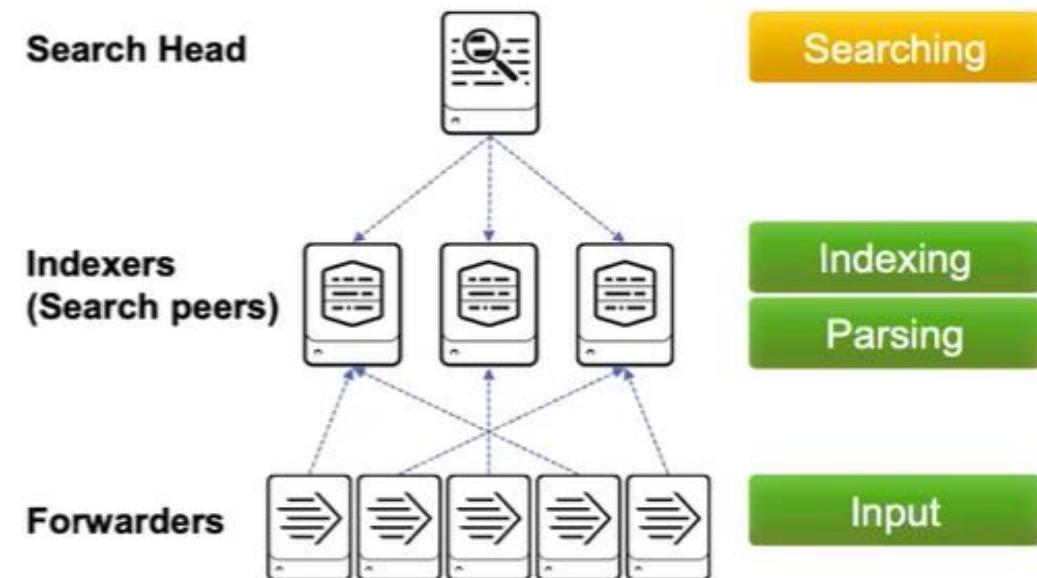


Basic Deployment for organizations:

- Indexing less than 20GB per day
- With under 20 users
- Small amount of forwarders

# Splunk Deployment – Multi-instance

- Increases indexing and searching capacity
- Search management and index functions are split across multiple machines

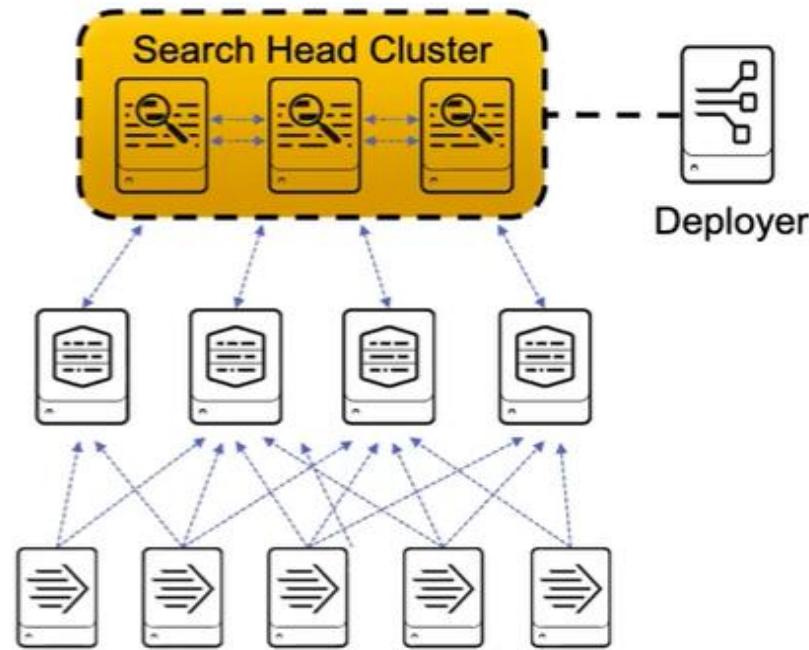


Deployment for organizations:

- Indexing up to 100 GB per day
- Supports 100 users
- Supports several hundred forwarders

# Splunk Deployment – Increasing Capacity

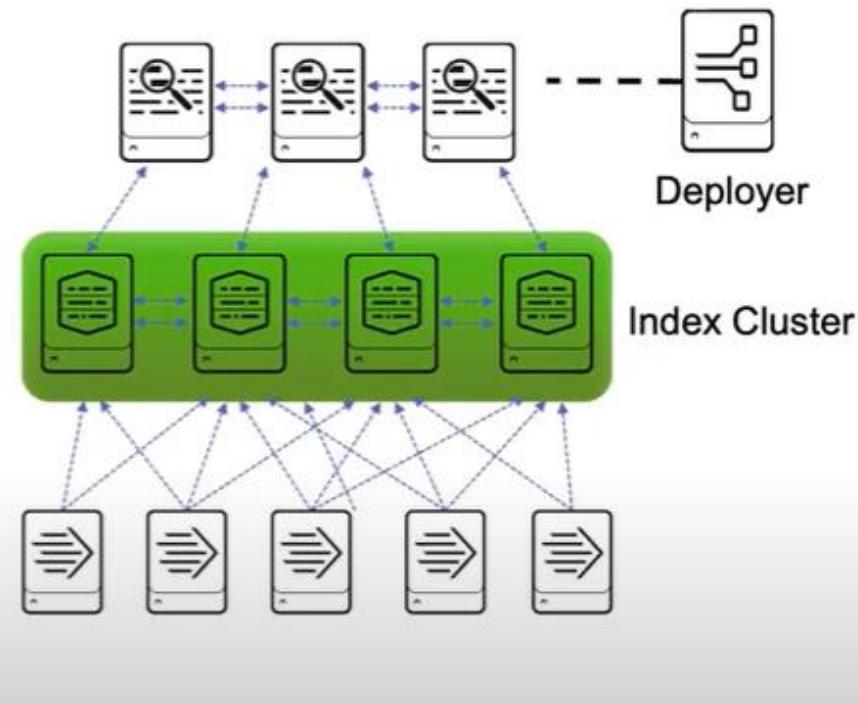
- Adding a Search Head Cluster:
  - Services more users for increased search capacity
  - Allows users and searches to share resources
  - Coordinate activities to handle search requests and distribute the requests across the set of indexers
- Search Head Clusters require a minimum of three Search Heads
- A Deployer is used to manage and distribute apps to the members of the Search Head Cluster



# Splunk Deployment – Index Cluster

---

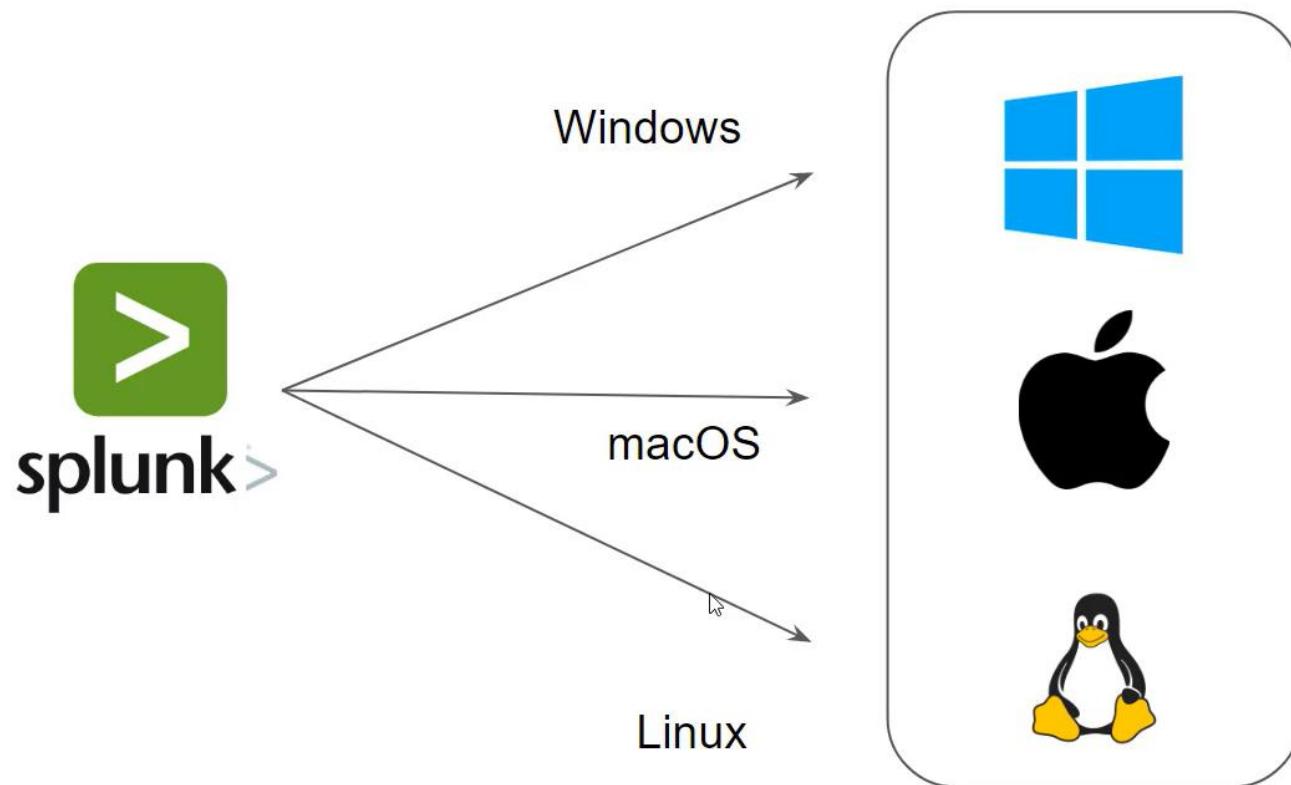
- Traditional Index Clusters:
  - Configured to replicate data
  - Prevent data loss
  - Promote availability
  - Manage multiple indexers
- Non-replicating Index Clusters
  - Offer simplified management
  - Do not provide availability or data recovery



# **Splunk Installation**

# Splunk Installation

---



- Splunk can be installed in wild varieties of operating system.
- Not all of the versions of the operating systems are supported

# Splunk Installation

---

- Create a Splunk account
- Go to the downloads for a specific package

## Free trials and downloads

### **Splunk Platform**

[Splunk Cloud](#)

[Splunk Enterprise](#)

[Splunk Universal Forwarder](#)

### **Splunk Observability**

[Splunk Infrastructure Monitoring](#)

[Splunk On-Call](#)

### **Splunk for Security**

[Splunk SOAR](#)

# Splunk Enterprise Install Package

## Splunk Enterprise 9.1.0.2

Index 500 MB/Day. Sign up and download now. After 60 days you can convert to a perpetual free license or purchase a Splunk Enterprise license to continue using the expanded functionality designed for enterprise-scale deployments.

### Choose Your Installation Package



Windows



Linux



Mac OS

64-bit

3.x+, 4.x+, or 5.4.x kernel Linux  
distributions

.deb 441.22 MB

[Download Now](#)

.tgz 574.28 MB

[Download Now](#)

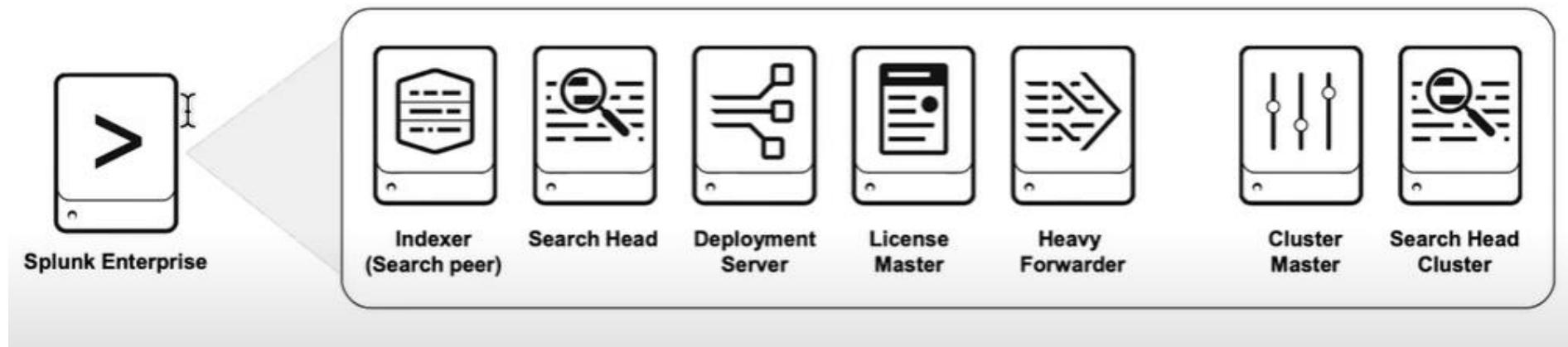
.rpm 489.89 MB

[Download Now](#)

# Splunk Enterprise Install Package

There are multiple Splunk components installed from the Splunk Enterprise package

- 500MB/day for 60 days indexing.
- After 60 days, it will be a free version with limited functionality.
- Required ports should be opened for Splunkweb, Splunkd and forwarder
- Download the Enterprise version from [https://www.splunk.com/en\\_us/download.html](https://www.splunk.com/en_us/download.html).



# Splunk Components Installation Overview

---

- Installing Splunk Enterprise as an Indexer or Search Head is identical to installing a single deployment instance.
- The difference happens at the configuration level
  - Installation as configuration is an iterative and ongoing event as you build and scale your deployment
  - Administrators need to be in control of the environment to fulfil emerging needs
  - Before installing Indexers or Search Head, be sure to keep in mind the different hardware requirements.

# Splunk Cloud Platform

GET STARTED

## Splunk Cloud Trial

Search, analyze, and visualize 5 GB/day of your own data in a Splunk hosted cloud environment for fast insights. Didn't want a cloud trial? Review our [Free Trials and Downloads page](#) for other options.

[Start Trial](#)

- Splunk-hosted cloud environment
- 5GB of data/day for 14 days
- Go to [https://www.splunk.com/en\\_us/download.html](https://www.splunk.com/en_us/download.html) and start the Free Trial.

# Splunk Ports

---

Usage	Splunk Enterprise	Universal Forwarder
splunkd	<b>8089</b>	<b>8089</b>
Splunk Web	<b>8000</b>	-
Web app-server proxy	<b>8065</b>	-
KV Store	<b>8191</b>	-
S2S receiving port(s)	No default	-
Any network/http input(s)	No default	No default
Index replication port(s)	Optional (no default)	-
Search replication port(s)	Optional (no default)	-

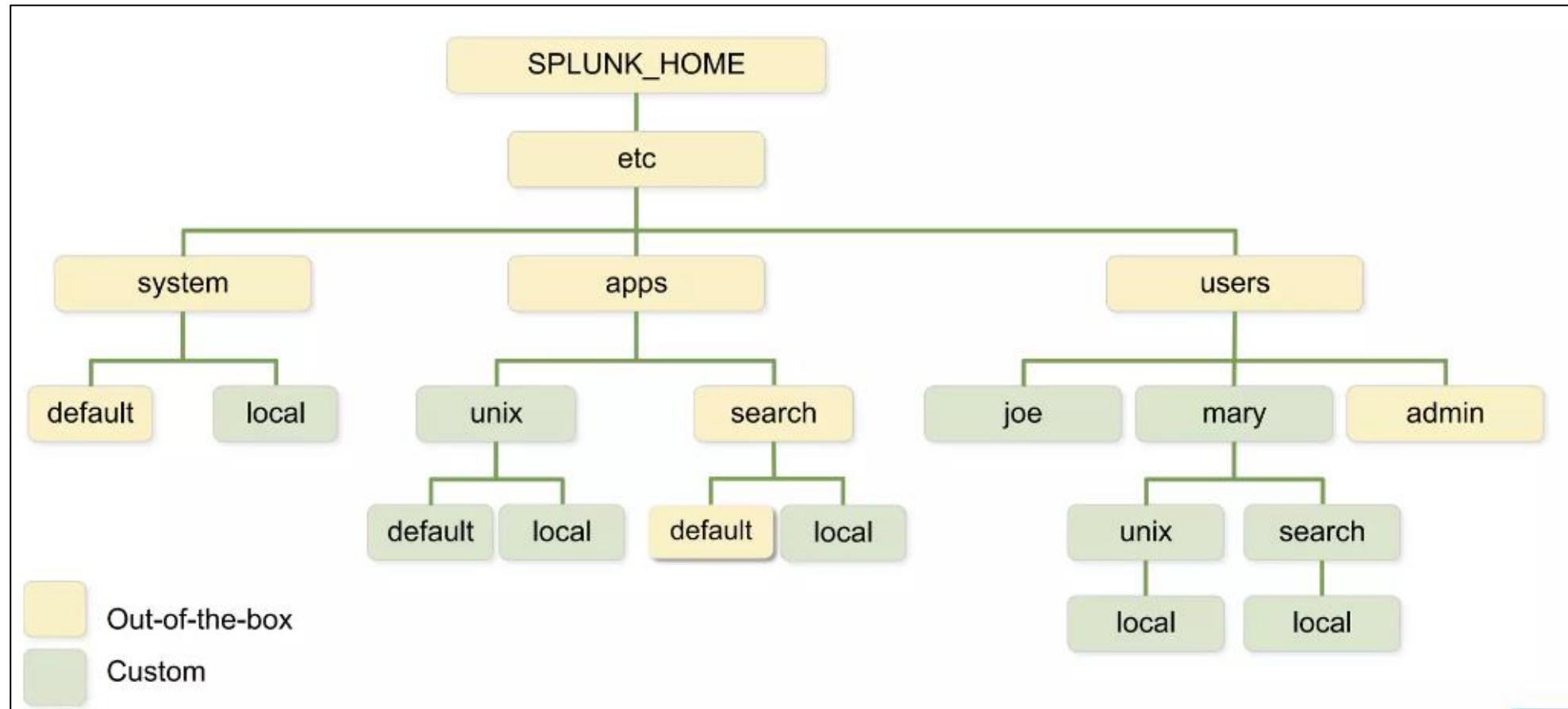
# Splunk Directory Structure

---

```
-r--r--r-- 1 10777 10777 520 Jul 1 05:32 README-splunk.txt
drwxr-xr-x 4 10777 10777 4096 Jul 1 05:52 bin
drwxr-xr-x 2 10777 10777 4096 Jul 1 05:50 cmake
-r--r--r-- 1 10777 10777 57 Jul 1 05:29 copyright.txt
drwxr-xr-x 17 10777 10777 4096 Aug 24 04:59 etc
drwxr-xr-x 3 10777 10777 4096 Jul 1 05:50 include
drwxr-xr-x 9 10777 10777 4096 Jul 1 05:52 lib
-r--r--r-- 1 10777 10777 85405 Jul 1 05:29 license-eula.txt
drwxr-xr-x 3 10777 10777 4096 Jul 1 05:50 openssl
drwxr-xr-x 3 10777 10777 4096 Aug 22 12:33 quarantined_files
drwxr-xr-x 4 10777 10777 4096 Jul 1 05:50 share
-r--r--r-- 1 10777 10777 3200662 Jul 1 05:52 splunk-9.1.0.1-77f73c9edb85-linux-2.6-x86_64-manifest
drwxr-xr-x 2 10777 10777 4096 Jul 1 05:51 swidtag
drwx--- 7 root root 4096 Aug 22 12:31 var
```

Directory	Description
bin	It contains primary Splunk binary as well as other tools
var	It contains all the data that get indexed as well as log files
etc	It contains all the configuration files as well as the apps and add-ons that you install
lib	It contains necessary libraries needed for Splunk to run

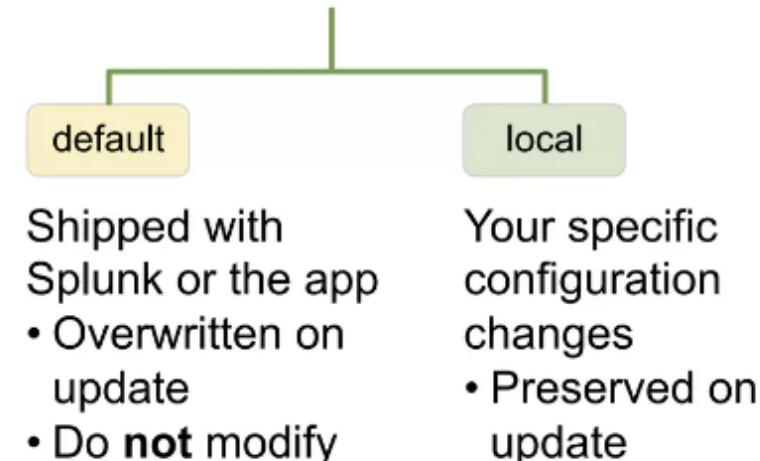
# Splunk Directory Structure



# Default vs Local Configuration

---

- Splunk ships with default **.conf** files
  - Stored in the **default** directories
- Add all configurations and edits to the **local** directory
  - Most configurations apply to only one app
- Avoid storing configurations in **SPLUNK\_HOME/etc/system**
  - Use the **Searching and Reporting** app as the default location for storing your configurations that are not app-specific



# Splunk Directory Structure

---

Other important directories

Directory	Description
Splunk/etc/apps	It contains Apps and add-ons
Splunk/etc/system/default/	Default configuration files
Splunk/var/log/	Logs related to Splunk and add-ons
Splunk/var/lib/splunk/	Indexed data gets stored here

# Splunk configuration files

---

A single Splunk instance typically has multiple versions of configuration files across several directories within the filesystem.

We can have configuration files with same names in **default**, **local** and **app** directories.

**Local** will get the higher preference over **app** followed by **default**.

**Local > app > default**

The default directory contains preconfigured versions of the configuration files/

Never make the changes in default directory, infact we can make changes in local directory which will override the default settings.

```
root@ip-172-31-9-191:~/splunk/etc/system# ls default/
alert_actions.conf    conf.conf      eventdiscoverer.conf   inputs.conf      multikv.conf      segmenters.conf    transactiontypes.conf  web.conf
app.conf              data          eventtypes.conf     limits.conf      outputs.conf      server.conf       transforms.conf    workflow_actions.conf
audit.conf            datamodels.conf federated.conf     literals.conf    procmon-filters.conf serverclass.conf  ui-prefs.conf     workload_policy.conf
authentication.conf  datatypesbnf.conf fields.conf      global-banner.conf messages.conf    props.conf        source-classifier.conf  ui-tour.conf     workload_pools.conf
authorize.conf        default-mode.conf   fields.conf     health.conf      metric_alerts.conf  restmap.conf      sourcetypes.conf  viewstates.conf   workload_rules.conf
collections.conf      distsearch.conf  global-banner.conf  health.conf     metric_rollups.conf savedsearches.conf telemetry.conf  visualizations.conf
commands.conf         event_renderers.conf indexes.conf    metric_alerts.conf  searchbnf.conf  sourcetypes.conf  times.conf      web-features.conf
root@ip-172-31-9-191:~/splunk/etc/system#
root@ip-172-31-9-191:~/splunk/etc/system# ls local/
README  migration.conf  server.conf
root@ip-172-31-9-191:~/splunk/etc/system#
```

# Splunk configuration files

---

By default Splunk is working on port number 8000 which has been mentioned in  
**/splunk/etc/system/default/web.conf**

We can create same file in the directory **/splunk/etc/system/local/web.conf**

```
[default]

[settings]

# enable/disable the appserver
startwebserver = 1

# First party apps:
splunk_dashboard_app_name = splunk-dashboard-studio

# enable/disable splunk dashboard app feature
enable_splunk_dashboard_app_feature = true

# port number tag is missing or 0 the server will NOT start an http listener
# this is the port used for both SSL and non-SSL (we only have 1 port now).
httpport = 8080
```

Restart the Splunk server and you can access it on **8080** as above file will override **default/web.conf**

# Splunk Precedence order

---

A Splunk platform deployment can have many copies of the same configuration file. These file copies are usually layered in directories that affect either the **users**, an **app**, or the **system** as a whole.

To determine the order of directories for evaluating configuration file precedence, Splunk software considers each file's context. Configuration files operate in either a **global context** or in the context of the current **app** and **user**:

- **Global.** Activities like indexing take place in a global context. They are independent of any app or user. For example, configuration files that determine monitoring or indexing behavior occur outside of the app and user context and are global in nature.
- **App/user.** Some activities, like searching, take place in an app or user context. The app and user context is vital to search-time processing, where certain knowledge objects or actions might be valid only for specific users in specific apps.

Index time	Global context	User-independent and background tasks such as inputs, parsing, indexing, etc.
Search time	App/User context	User-related activity, such as searching

# Splunk Precedence order

---

## Precedence within global context

When the file context is global, directory priority descends in this order:

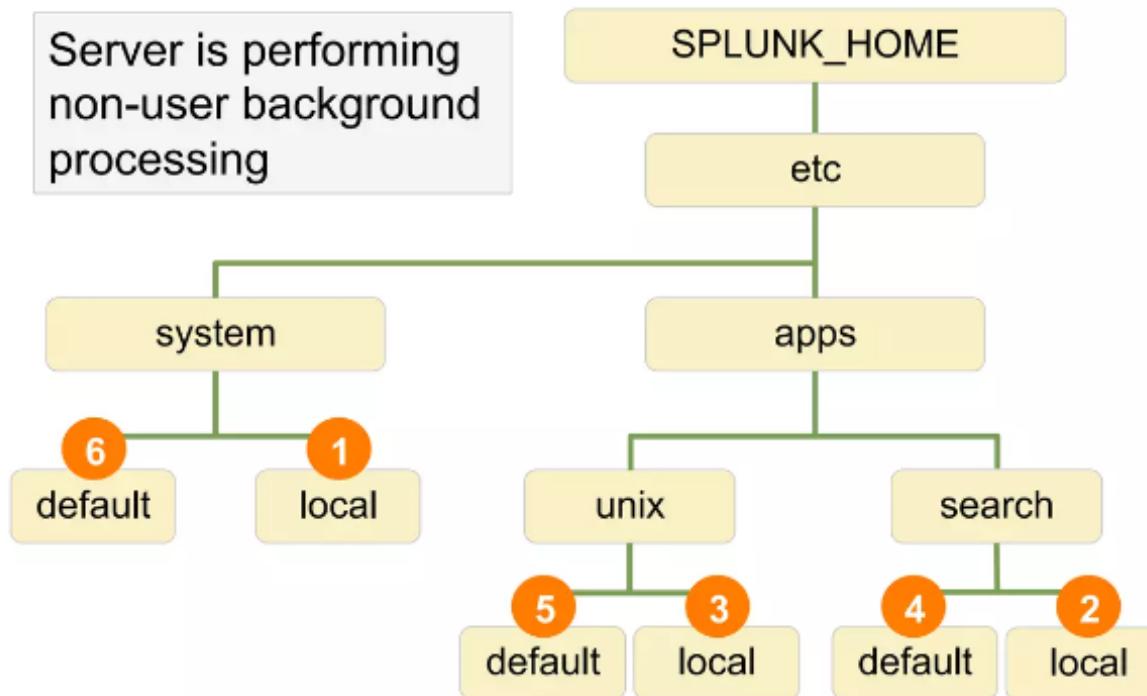
- 1.** System local directory -- highest priority
- 2.** App local directories
- 3.** App default directories
- 4.** System default directory -- lowest priority

## Precedence within app or user context

For files with an app/user context, directory priority descends from user to app to system:

- 1.** User directories for current user -- highest priority
- 2.** App directories for currently running app (local, followed by default)
- 3.** App directories for all other apps (local, followed by default) -- for exported settings only
- 4.** System directories (local, followed by default) -- lowest priority

# Index time precedence



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

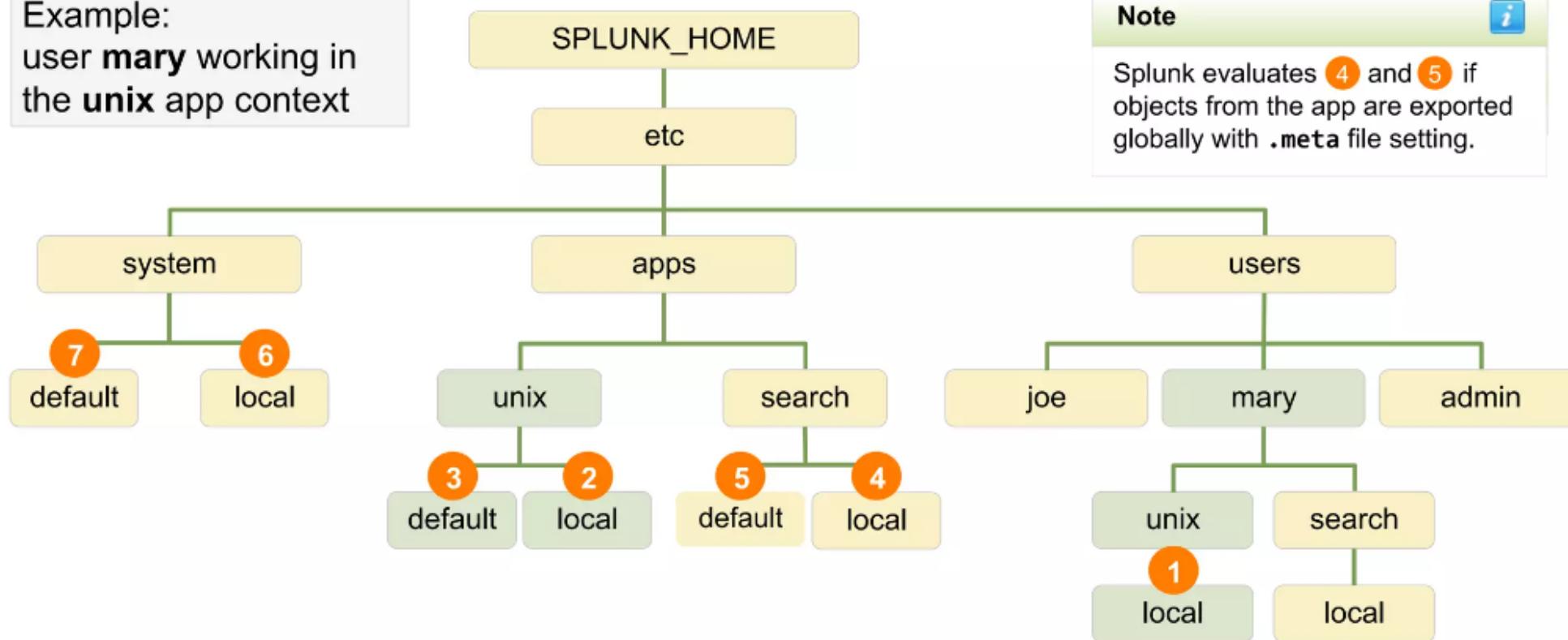
## Note



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

# Search time Precedence

Example:  
user **mary** working in  
the **unix** app context



## Note

Splunk evaluates 4 and 5 if objects from the app are exported globally with .meta file setting.

# Splunk Precedence order

---

The effect of app directory names varies depending on whether the context is global or local.

## **App directory names in the global context**

When determining priority in the global context, Splunk software uses **lexicographical** order to determine priority among the collection of apps directories. For example, files in an apps directory named "A" have a higher priority than files in an apps directory named "B", and so on.

## **App directory names in the app/user context**

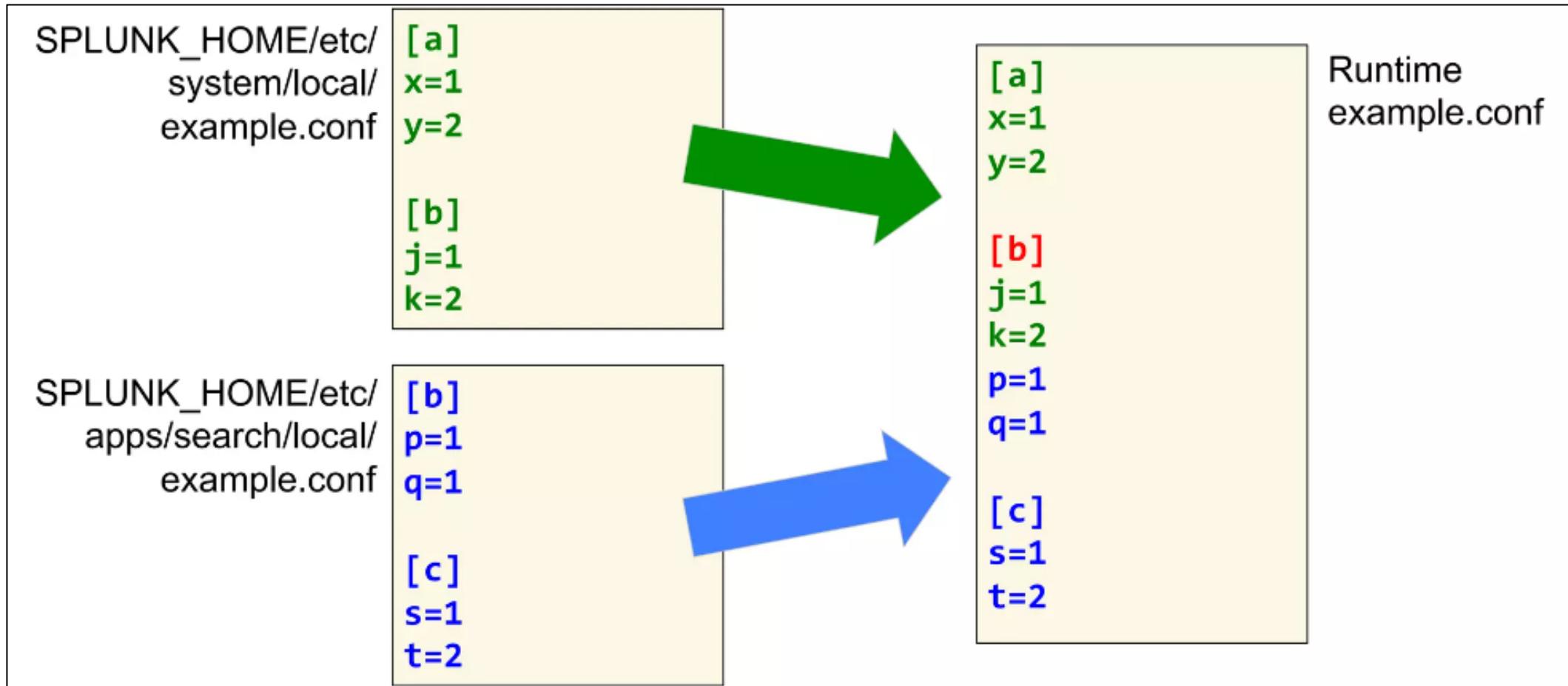
When determining priority in the app/user context, Splunk software uses **reverse-lexicographical** order to determine priority among the collection of apps directories, For example, files in an apps directory named "B" have a higher priority than files in an apps directory named "A", and so on.

# Run Time merging of configurations

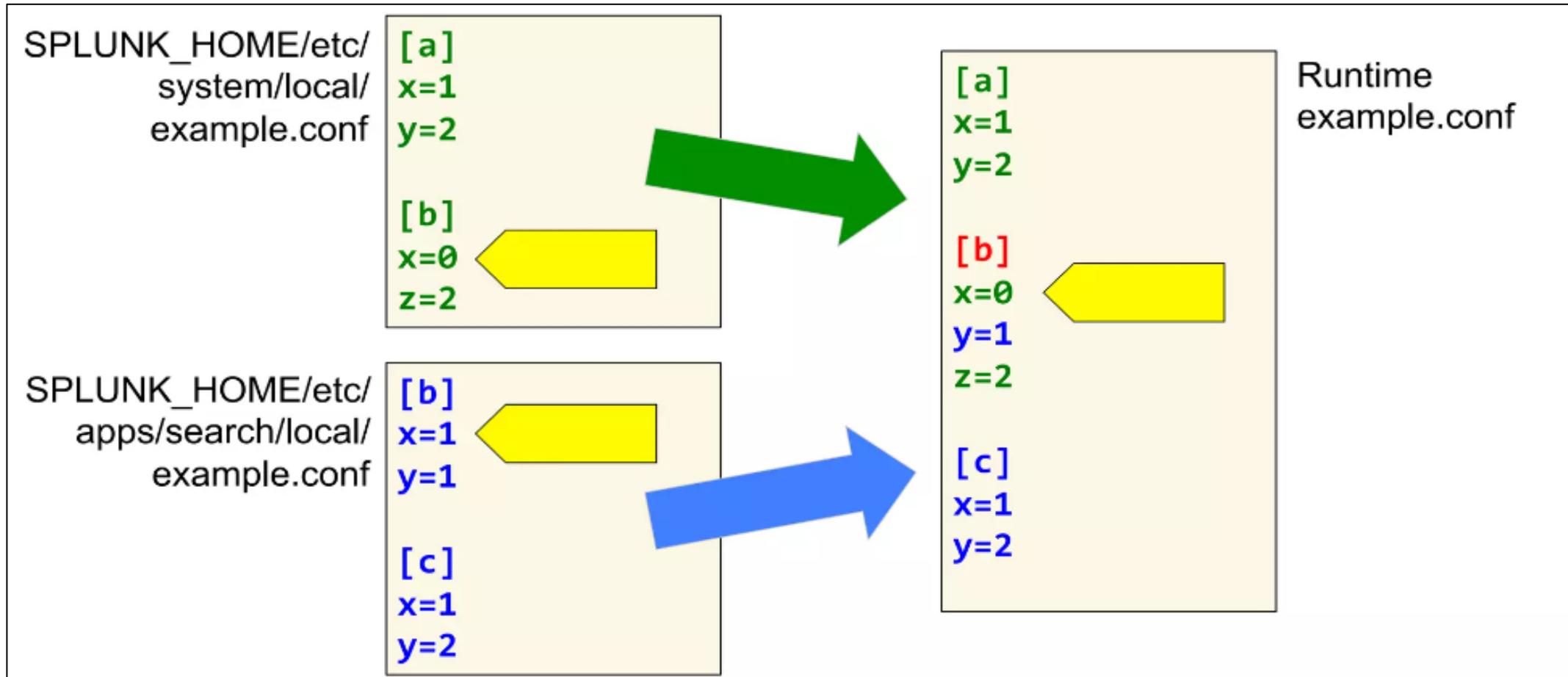
---

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

# Run Time merging (with no conflict)



# Run Time merging (with conflict)



# Overriding Defaults

---

- There are default settings in **SPLUNK\_HOME/etc/system/default** and **SPLUNK\_HOME/etc/apps/search/default**
- The correct method to override these settings, if needed, is to do so in the **local** directory at the same scope
  - Only add the items you are overriding—not a whole copy of the default conf file
- Example:
  - To disable a default attribute **TRANSFORMS** for **[syslog]**:

```
# etc/system/default/props.conf
[syslog]
TRANSFORMS = syslog-host
REPORT-syslog = syslog-extractions
...
```

```
# etc/system/local/props.conf
[syslog]
TRANSFORMS =
```

# Common Splunk Commands

---

Splunk is the program in the bin directory to run the CLI

Command	Operation
<code>splunk help</code>	Display a usage summary
<code>splunk [start   stop   restart]</code>	Manage the Splunk processes
<code>splunk start --accept-license</code>	Automatically accept the license without prompt
<code>splunk status</code>	Display the Splunk process status
<code>splunk show splunkd-port</code>	Show the port that the <code>splunkd</code> listens on
<code>splunk show web-port</code>	Show the port that Splunk Web listens on
<code>splunk show servername</code>	Show the servername of this instance
<code>splunk show default-hostname</code>	Show the default host name used for all data inputs
<code>splunk enable boot-start -user</code>	Initialize script to run Splunk Enterprise at system startup

# Splunk UI

By default, It works on port no 8000

The screenshot shows the Splunk Enterprise UI homepage. At the top left is the 'splunk>enterprise' logo. To its right are navigation links: 'Apps ▾', 'Administrator ▾', 'Messages ▾', 'Settings ▾', 'Activity ▾', 'Help ▾', and a search bar labeled 'Find'. On the far right is a magnifying glass icon.

The main content area starts with a greeting 'Hello, Administrator'. Below it is a 'Quick links' menu with options: 'Dashboard', 'Recently viewed', 'Created by you', and 'Shared with you'. A 'Common tasks' section follows, containing six cards:

- Add data**: Add data from a variety of common sources.
- Search your data**: Turn data into doing with Splunk search.
- Visualize your data**: Create dashboards that work for your data.
- Add team members**: Add your team members to Splunk platform.
- Manage permissions**: Control who has access with roles.
- Configure mobile devices**: Login or manage mobile devices using Splunk Secure Gateway.

Below this is a 'Learning and resources' section with six cards:

- Product tours**: New to Splunk? Take a tour to help you on your way.
- Learn more with Splunk Docs**: Deploy, manage, and use Splunk software with comprehensive guidance.
- Get help from Splunk experts**: Actionable guidance on the Splunk Lantern Customer Success Center.
- Extend your capabilities**: Browse thousands of apps on Splunkbase.
- Join the Splunk Community**: Learn, get inspired, and share knowledge.
- See how others use Splunk**: Browse real customer stories.

# Lab: Exercise1

---

- Download and install Splunk
- Login to Splunk via web
- Check the directory structure of Splunk Instance

# **Getting Data in**

# What type of data to index?



# Splunk Index Time Process

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



# Data Input Types

---

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

# Default Metadata Settings

---

- When you index a data source, Splunk assigns metadata values
  - The metadata is applied to the entire source
  - Splunk applies defaults if not specified
  - You can also override them at input time or later

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

# Adding an Input with Splunk Web

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



The screenshot illustrates the navigation path and the 'Data inputs' configuration screen. On the left, a sidebar menu is shown with the following items:

- 1 Settings ▾
- 2 Data inputs (highlighted)
- Forwarding and receiving
- Indexes
- Report acceleration summaries
- Virtual indexes
- Source types

On the right, the main content area displays the 'Data inputs' page with the following details:

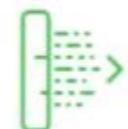
**Data inputs**  
Set up data inputs from files and directories, network ports, and scripted inputs. If you want to set up forwarding and receiving between two Splunk instances, go to [Forwarding and receiving](#).

**Local inputs**

Type	Inputs	Actions
Files & Directories	7	<b>3 + Add new</b>

# Add Data Menu

Add Data menu provides three options depending on the source to be used

Add Data		
How do you want to add data?		
 Upload files from my computer	 Monitor files and ports on this Splunk indexer	 Forward data from Splunk forwarder
Upload Option	Monitor Option	Forward Option
Upload allows uploading local files that only get indexed once. Useful for testing or data that is created once and never gets updated. Does not create <b>inputs.conf</b> .	Provides one-time or continuous monitoring of files, directories, http events, network ports, or data gathering scripts located on Splunk Enterprise instances. Useful for testing inputs.	Main source of input in production environments. Remote machines gather and forward data to indexers over a receiving port.

# Select Source

Add Data     Select Source   Set Source Type   Input Settings   Review   Done   < Back   **Next >**

**1** Select the **Files & Directories** option to configure a monitor input

**2** To specify the source:

- Enter the absolute path to a file or directory, or
- Use the **Browse** button

**3** On Windows: c:\apache\apache.error.log or \\hostname\apache\apache.error.log. On Unix: /var/log or www01/var/log.

File or Directory ?  **Browse**

Continuously Monitor   Index Once

For ongoing monitoring   For one-time indexing (or testing); the **Index Once** option does not create a stanza in **inputs.conf**

Whitelists ?   Blacklist ?

Files & Directories  
Upload a file, index a local file, or monitor an entire directory.

HTTP Event Collector  
Configure tokens that clients can use to send data over HTTP or HTTPS.

TCP / UDP  
Configure Splunk to listen on a network port.

Scripts  
Get data from any API, service, or database with a script.

# Set Source Type

## Set Source Type

This page lets you see how Splunk sees your data before indexing. If the events look correct and have the right timestamps, click "Next" to proceed. If not, use the options below to define proper event breaks and timestamps. If you cannot find an appropriate source type for your data, create a new one by clicking "Save As".

Source: /opt/log/www1/access.log

[View Event Summary](#)

The screenshot shows the 'Set Source Type' page with the following interface elements:

- Source type:** access\_combined\_wcookie (highlighted with a red circle 1)
- Save As:** button
- Filter:** input field containing 'filter' (highlighted with a red circle 2)
- Search icon:** magnifying glass icon next to the filter input (highlighted with a red circle 3)
- Event List:** A table with columns 'Time' and 'Event'. The first event is highlighted with a red box and a red circle 4.
- Event Details:** The first event's timestamp and log line are shown in detail: 11/28/17 4:58:01.000 PM, 111.161.27.28 - - [Nov/2017:16:58:01] "GET /cart.do?action=remove&itemId=EST-19&productId=PZ-SG-G85&JSESSIONID=SD6SL1FF4ADFF4960 HTTP/1.1" 200 2708 "http://www.buttercupgames.com" "Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.2; .NET CLR 1.1.4322; InfoPath.1; MS-RTC LM 8)" 728
- Page Navigation:** List, Format, 20 Per Page, Prev, Next, page numbers 1-8
- Source Type Description:** access\_combined, National Center for Supercomputing Applications (NCSA) combined format HTTP web server logs (can be generated by apache or other web servers)

# Overview of Source Type

---

The **source type** is one of the default fields that the Splunk platform assigns to all incoming data.

It tells the platform what kind of data you have, so that it can format the data intelligently during indexing.

```
122.162.148.139 - - [23/Aug/2023:08:18:55 +0000] "GET / HTTP/1.1" 200 3459 "-" "Mozilla/5.0  
(Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/116.0.0.0  
Safari/537.36"
```

# HTTP access logs

---

**122.162.148.139 - - [23/Aug/2023:08:18:55 +0000] "GET / HTTP/1.1" 200 3459 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/116.0.0.0 Safari/537.36"**

Extracted Information	Description
122.162.148.139	Client IP Address
[23/Aug/2023:08:18:55 +0000]	Timestamp
GET	HTTP method
200	Response Code
3459	Byte

# Set Source Type

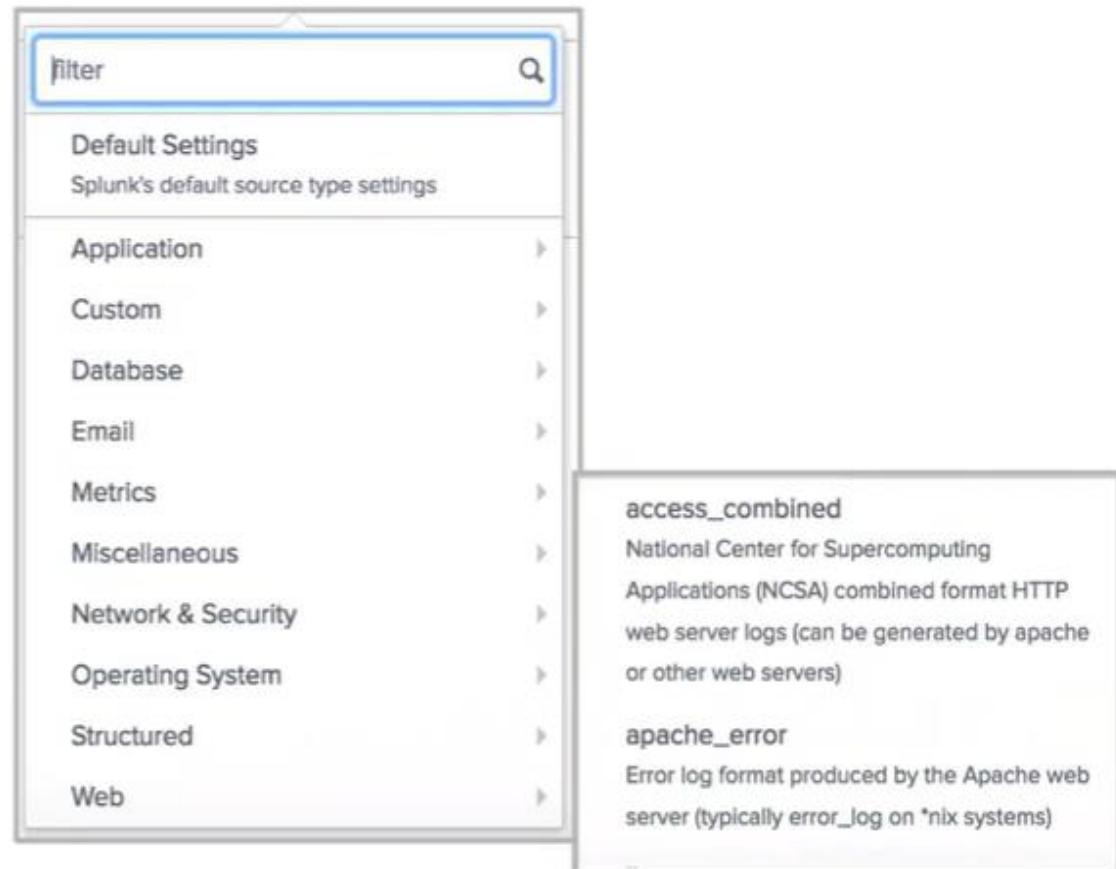
---

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

# Pretrained Source Types

---

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



# Input Settings

The app context determines where your input configuration is saved

- In this example, it will be saved in:  
**SPLUNK\_HOME/etc/apps/search/local**

App context

Application contexts are folders within a Splunk instance that contain configurations for a specific use case or domain of data. App contexts improve manageability of input and source type definitions. Splunk loads all app contexts based on precedence rules. [Learn More](#)

Host

When Splunk indexes data, each event receives a "host" value. The host value should be the name of the machine from which the event originates. The type of input you choose determines the available configuration options. [Learn More](#)

Index

Splunk stores incoming data as events in the selected index. Consider using a "sandbox" index as a destination if you have problems determining a source type for your data. A sandbox index lets you troubleshoot your configuration without impacting production indexes. You can always change this setting later.

Host field value:

Index:  [Create a new index](#)

# Review

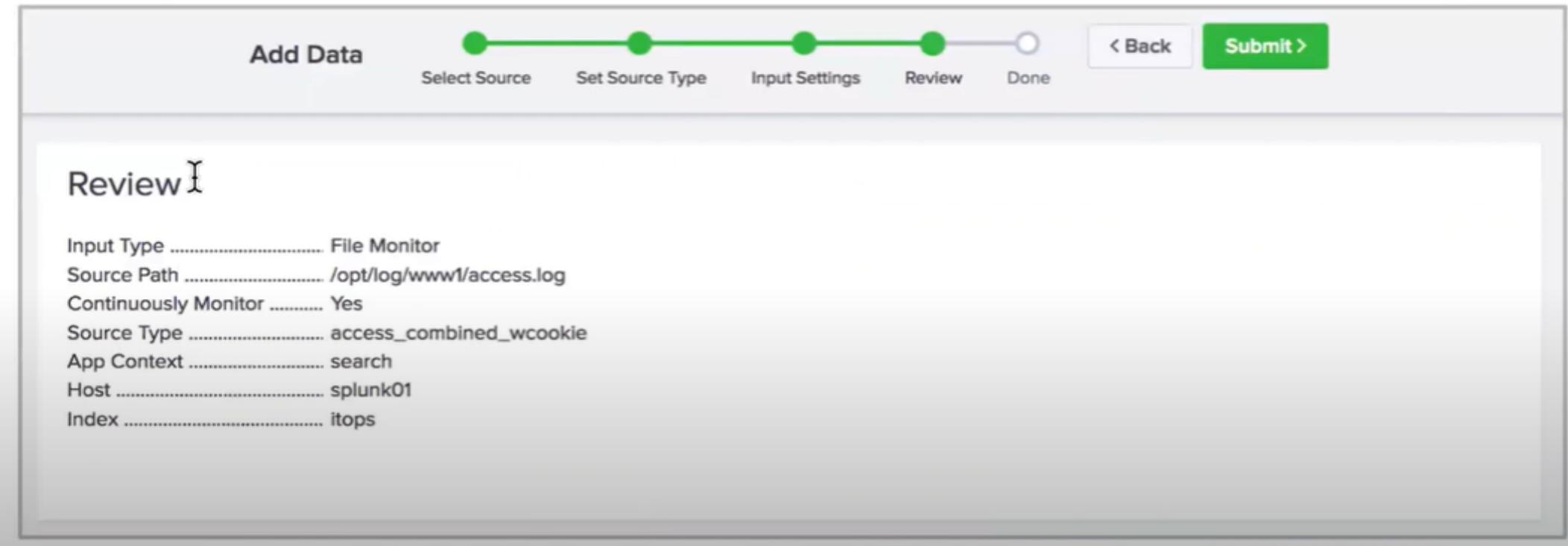
- Review the input configuration summary and click **Submit** to finalize

Add Data

Review

Input Type ..... File Monitor  
Source Path ..... /opt/log/www1/access.log  
Continuously Monitor ..... Yes  
Source Type ..... access\_combined\_wcookie  
App Context ..... search  
Host ..... splunk01  
Index ..... itops

< Back **Submit >**



# What Happens Next?

---

- Indexed events are available for immediate search
  - However, it may take a minute for Splunk to *start* indexing the data
- You are given other options to do more with your data

The screenshot shows a progress bar at the top with five steps: 'Add Data', 'Select Source', 'Set Source Type', 'Input Settings', 'Review', and 'Done'. The 'Review' step is highlighted with a green circle. Below the progress bar, the word 'Review' is displayed. A table lists the configuration settings:

Setting	Value
Input Type	File Monitor
Source Path	/opt/log/www1/access.log
Continuously Monitor	Yes
Source Type	access_combined_wcookie
App Context	search
Host	splunk01
Index	test

Buttons for 'Back' and 'Submit' are located at the bottom right of the form.

# Important Notes

---

Most of the common types of data are easily parsed by Splunk considering right source type are associated with it.

Additional Splunk Add Ons are available in marketplaces that can also parse the data for a specific source type.

For custom data, you can create your own parser that can parse the data.

# **Demo: Import Log Data**

# Import the logs data

---

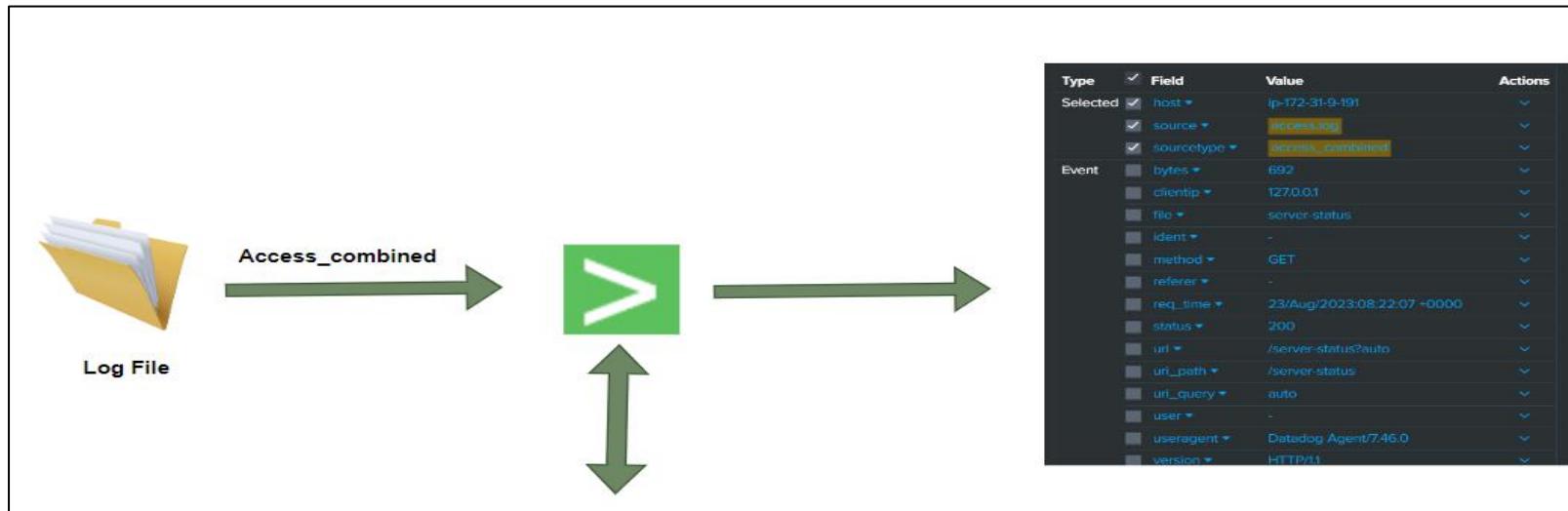
We will upload the below data which are stored in a file.

Data Types	Description
HTTP Access logs	Records data of requests processed by Web-Server
Linux Authentication Logs	Authentication Success and Failure Related Messages

# Log Parsing

Whenever we upload a log file, it is important to set right source type associated with it.

This allows Splunk to parse the log accordingly.



Sourcetype	Parser
access_combined	Parser1
Linux_audit	parser2

# Upload files from the computer

---



**Upload**  
files from my computer

Local log files  
Local structured files (e.g. CSV)  
[Tutorial for adding data](#) ↗



**Monitor**  
files and ports on this Splunk platform instance

Files - HTTP - WMI - TCP/UDP - Scripts  
Modular inputs for external data sources



**Forward**  
data from a Splunk forwarder

Files - TCP/UDP - Scripts

# Upload access.log

Add Data

Select Source Set Source Type Input Settings Review Done

< Back Next >

## Select Source

Choose a file to upload to the Splunk platform, either by browsing your computer or by dropping a file into the target box below. [Learn More](#)

Selected File: **access.log**

Select File

Drop your data file here

The maximum file upload size is 500 Mb

 File Successfully Uploaded

# Set Source Type

Add Data

Select Source Set Source Type Input Settings Review Done

Next < Back

## Set Source Type

This page lets you see how the Splunk platform sees your data before indexing. If the events look correct and have the right timestamps, click "Next" to proceed. If not, use the options below to define proper event breaks and timestamps. If you cannot find an appropriate source type for your data, create a new one by clicking "Save As".

Source: **access.log**

View Event Summary

Source type: access\_combined ▾ Save As

List ▾ Format 20 Per Page ▾

< Prev 1 2 3 4 5 6 7 8 9 Next >

	Time	Event
1	8/23/23 7:48:52.000 AM	127.0.0.1 - - [23/Aug/2023:07:48:52 +0000] "GET /server-status?auto HTTP/1.1" 200 629 "-" "Datadog Agent/7.46.0"
2	8/23/23 7:49:07.000 AM	127.0.0.1 - - [23/Aug/2023:07:49:07 +0000] "GET /server-status?auto HTTP/1.1" 200 661 "-" "Datadog Agent/7.46.0"

Event Breaks

Timestamp

Advanced

# Input Settings

Add Data           [< Back](#) [Review >](#)

## Input Settings

Optionally set additional input parameters for this data input as follows:

**Host**

When the Splunk platform indexes data, each event receives a "host" value. The host value should be the name of the machine from which the event originates. The type of input you choose determines the available configuration options. [Learn More ↗](#)

Constant value  
 Regular expression on path  
 Segment in path

Host field value

**Index**

The Splunk platform stores incoming data as events in the selected index. Consider using a "sandbox" index as a destination if you have problems determining a source type for your data. A sandbox index lets you troubleshoot your configuration without impacting production indexes. You can always change this setting later. [Learn More ↗](#)

Index [Default ▾](#) [Create a new index](#)

# Review

---

Add Data

Select Source   Set Source Type   Input Settings   Review   Done

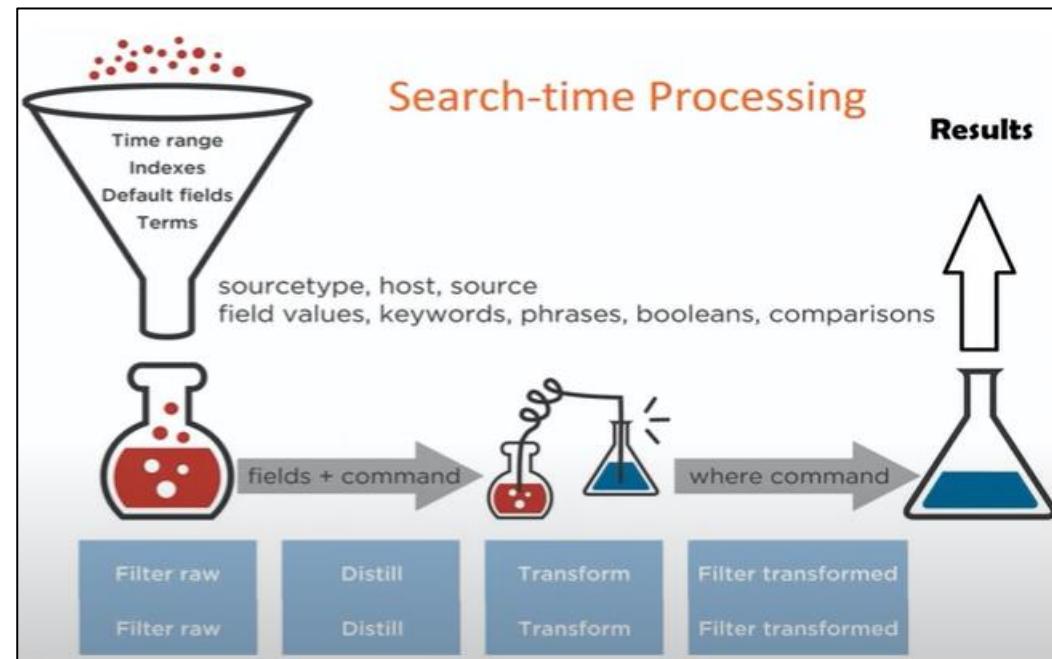
< Back   **Submit >**

Review

Input Type ..... Uploaded File  
File Name ..... access.log  
Source Type ..... access\_combined  
Host ..... ip-172-31-9-191  
Index ..... Default

# Search Time Processing

- When a search starts, matching indexed events are retrieved from disk, fields are extracted from the event's text.
- The event is classified by matching against event type definitions. (eg error, login)
- The event returned from a search can then be powerfully transformed using Splunk's search language to generate reports that live on dashboards.



# Search Data

The screenshot shows the Splunk Enterprise search interface. At the top, there's a dark header bar with the "splunk>enterprise" logo, a "Find" icon, and various navigation links like "Administrator", "Messages", "Settings", "Activity", "Help", and a search bar. Below the header is a secondary navigation bar with links for "Search", "Analytics", "Datasets", "Reports", "Alerts", and "Dashboards". The "Search" link is highlighted with a green underline. To the right of this bar is a "Search & Reporting" button with a green arrow icon.

The main content area is titled "Search" and contains a search bar with the placeholder "enter search here...". To the right of the search bar are filters for "Last 24 hours" and a magnifying glass icon. Below the search bar is a dropdown menu set to "No Event Sampling" and another for "Smart Mode".

On the left side, there's a sidebar with a "How to Search" section containing text about available resources and three buttons: "Documentation", "Tutorial", and "Data Summary". The "Data Summary" button is highlighted with a blue border. On the right side, there's a section titled "Analyze Your Data with Table Views" with text about Table Views, a "Create Table View" button, and links to learn more or view datasets.

# Search data

Data Summary

Hosts (1) Sources (1) Sourcetypes (1)

filter

Host	Count	Last Update
ip-172-31-9-191	161	8/23/23 8:31:02.000 AM

Data Summary

Hosts (1) Sources (1) Sourcetypes (1)

filter

Source	Count	Last Update
access.log	161	8/23/23 8:31:02.000 AM

Data Summary

Hosts (1) Sources (1) Sourcetypes (1)

filter

Sourcetype	Count	Last Update
access_combined	161	8/23/23 8:31:02.000 AM

# Search Data

Splunk > enterprise Apps ▾

Administrator ▾ 2 Messages ▾ Settings ▾ Activity ▾ Help ▾ Find

Search Analytics Datasets Reports Alerts Dashboards > Search & Reporting

New Search Save As ▾ Create Table View Close

source="access.log" All time ▾ Q

✓ 161 events (before 8/23/23 9:20:27.000 AM) No Event Sampling ▾ Job ▾ II ■ ⌂ ⌂ ⌂ Smart Mode ▾

Events (161) Patterns Statistics Visualization

Format Timeline ▾ – Zoom Out + Zoom to Selection × Deselect 1 minute per column

List ▾ Format 20 Per Page ▾ ◀ Prev 1 2 3 4 5 6 7 8 9 Next >

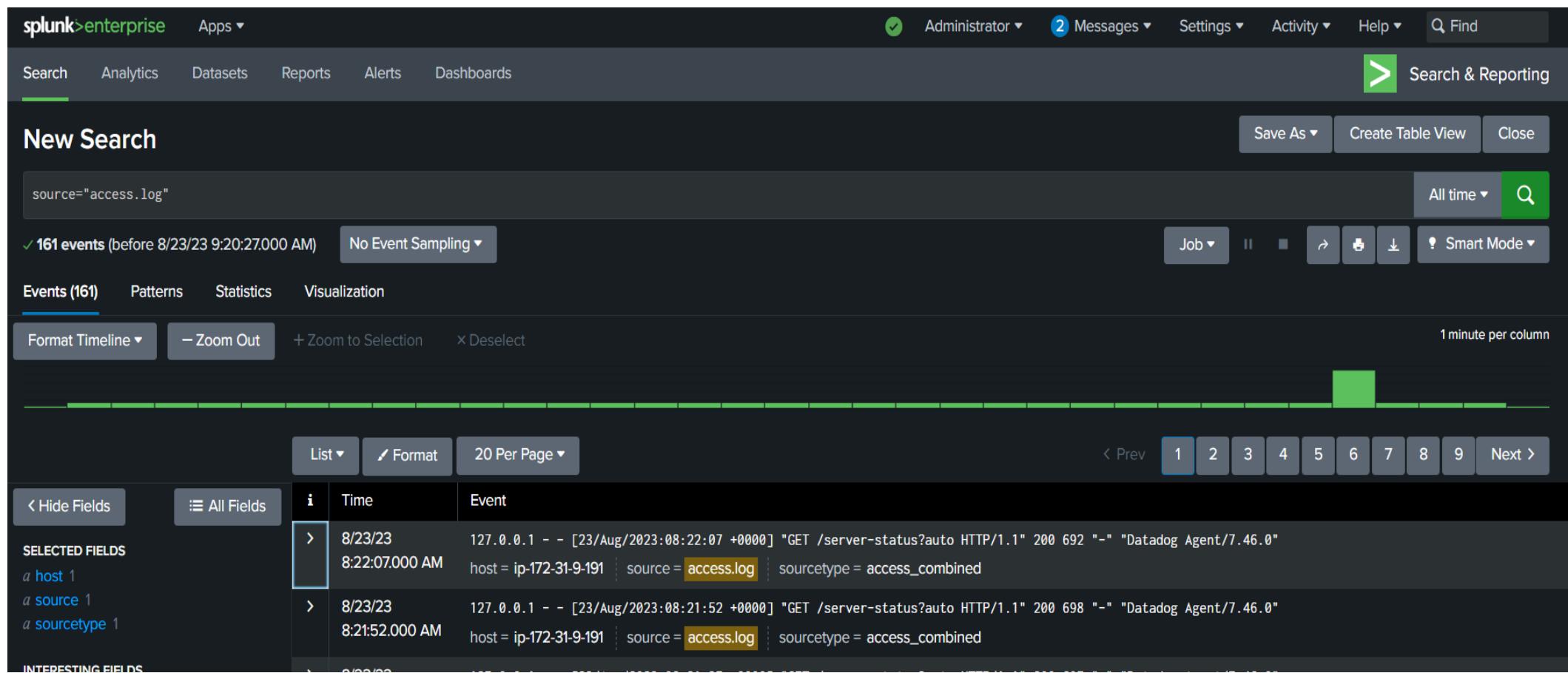
◀ Hide Fields All Fields i Time Event

SELECTED FIELDS a host 1 a source 1 a sourcetype 1

Time	Event
8/23/23 8:22:07.000 AM	127.0.0.1 - - [23/Aug/2023:08:22:07 +0000] "GET /server-status?auto HTTP/1.1" 200 692 "-" "Datadog Agent/7.46.0" host = ip-172-31-9-191   source = access.log   sourcetype = access_combined
8/23/23 8:21:52.000 AM	127.0.0.1 - - [23/Aug/2023:08:21:52 +0000] "GET /server-status?auto HTTP/1.1" 200 698 "-" "Datadog Agent/7.46.0" host = ip-172-31-9-191   source = access.log   sourcetype = access_combined

INTERESTING FIELDS

8/23/23



# Parsed Data

Splunk intelligently parse the data from the logs which can be seen below.

The screenshot shows the Splunk parse configuration interface. On the left, under "SELECTED FIELDS", there are three entries: host, source, and sourcetype, each with a count of 1. Under "INTERESTING FIELDS", there is a long list of various log fields with their counts: bytes (26), clientip (2), date\_hour (2), date\_mday (1), date\_minute (35), date\_month (1), date\_second (17), date\_wday (1), date\_year (1), date\_zone (1), file (3), ident (1), index (1), linecount (1), method (1), punct (4), referer (2), req\_time (100+), splunk\_server (1), status (2), timeendpos (2), timestamppos (2), uri (4), uri\_path (4), uri\_query (1), and user (1). The main area displays a single event log entry. The event details are as follows:

Type	Field	Value	Actions
Selected	host	ip-172-31-9-191	▼
	source	access.log	▼
	sourcetype	access_combined	▼
Event	bytes	692	▼
	clientip	127.0.0.1	▼
	file	server-status	▼
	ident	-	▼
	method	GET	▼
	referer	-	▼
	req_time	23/Aug/2023:08:22:07 +0000	▼
	status	200	▼
	uri	/server-status?auto	▼
	uri_path	/server-status	▼
	uri_query	auto	▼
	user	-	▼
	useragent	Datadog Agent/7.46.0	▼
	version	HTTP/1.1	▼
Time	_time	2023-08-23T08:22:07.000+00:00	▼
Default	index	main	▼
	linecount	1	▼

# Search Options

---

## 1. Search from a file uploaded

New Search

```
source="access.log"
```

✓ 161 events (before 8/23/23 9:20:27.000 AM) No Event Sampling ▾

## 2. Search along with sourcetype

```
source="access.log" sourcetype="access_combined"
```

✓ 161 events (before 8/23/23 9:45:09.000 AM) No Event Sampling ▾

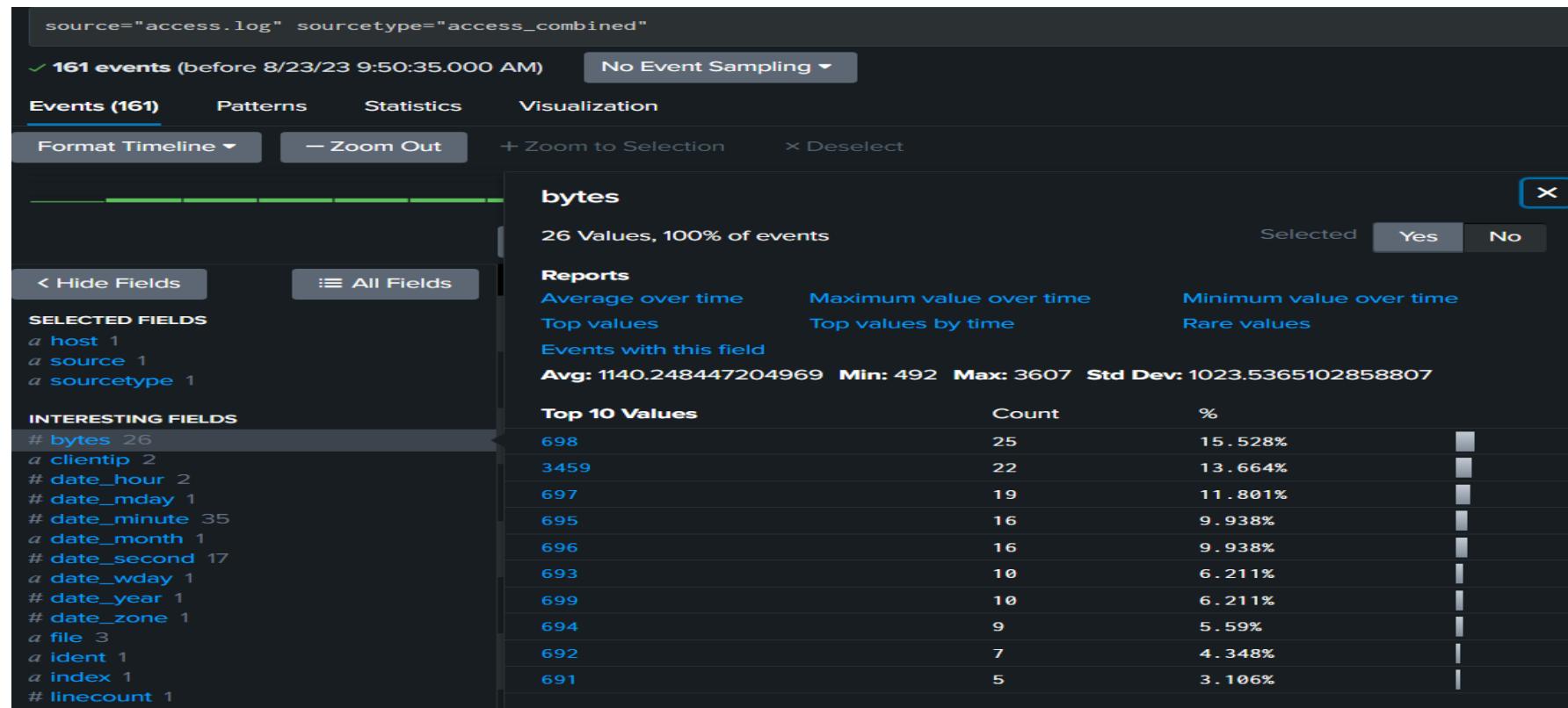
## 3. Search along with byte information

```
source="access.log" sourcetype="access_combined" bytes=698
```

✓ 25 events (before 8/23/23 9:49:02.000 AM) No Event Sampling ▾

# Search Options

We can filter the data from the fields mentioned on the left hand side.



Search

enter search here...

No Event Sampling

How to Search

If you are not familiar with the search features, o

[Documentation](#)[Tutorial](#)[> Search History](#)

## Data Summary

Hosts (5) Sources (8) Sourcetypes (3)

filter

Sourcetype	all	Count	Last Update
access_combined_wcookie	all	39,532	2/28/18 1:39:10.000 PM
secure	all	40,088	2/28/18 1:39:10.000 PM
vendor_sales	all	30,244	2/28/18 1:39:10.000 PM

Last 24 hours



Verbose Mode

20 hours ago

LATEST EVENT

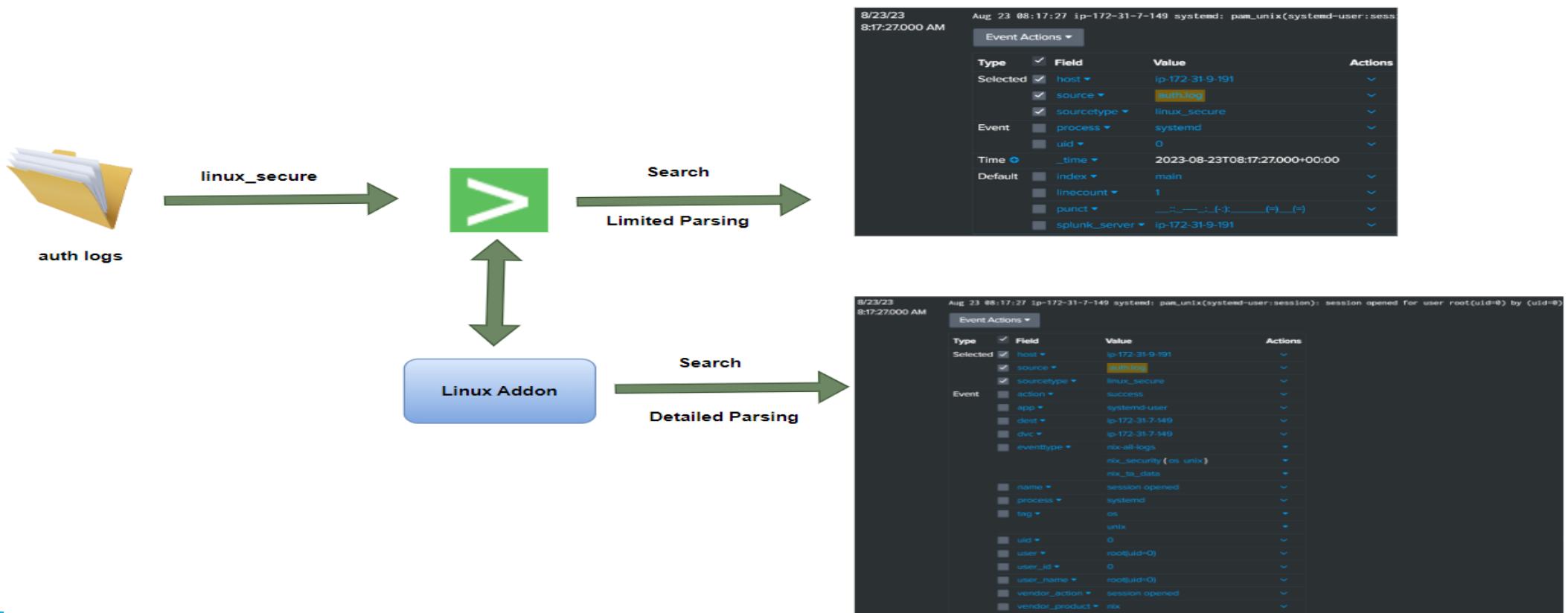
After adding data use the Search view to run searches, design data visualizations, save reports, and create dashboards. To know more about the data in your deployment, see the Data Summary.

[Skip tour](#)

# **Demo: Parsing auth data**

# Parsing Linux Authentication logs

For the logs that are not parsed by Splunk, you can install various Addons from Splunk marketplace that can do the parsing for us



# Parsing Linux Authentication logs

- Upload the access logs
- Select sourcetype as linux\_secure

You will find the result as shown below

Data Summary

Hosts (1) Sources (2) Sourcetypes (2)

filter

Source	Count	Last Update
access.log	161	8/23/23 8:31:02.000 AM
auth.log	393	8/23/23 10:24:54.000 AM

Data Summary

Hosts (1) Sources (2) Sourcetypes (2)

filter

Sourcetype

Sourcetype	Count	Last Update
access_combined	161	8/23/23 8:31:02.000 AM
linux_secure	393	8/23/23 10:24:54.000 AM

# Limited Parsing

Time Event

8/23/23 Aug 23 08:17:27 ip-172-31-7-149 systemd: pam\_unix(systemd-user:session): session opened for user root(uid=0) by (uid=0)  
8:17:27.000 AM

Event Actions ▾

Type	✓ Field	Value	Actions
Selected	✓ host	ip-172-31-9-191	▼
	✓ source	auth.log	▼
	✓ sourcetype	linux_secure	▼
Event	process	systemd	▼
	uid	0	▼
Time	_time	2023-08-23T08:17:27.000+00:00	▼
Default	index	main	▼
	linecount	1	▼
	punct	__::__--__:_(-):____(=)___(=)	▼
	splunk_server	ip-172-31-9-191	▼

# Install the Linux Addon for detailed parsing

splunk>enterprise Apps ▾

**Browse More Apps**

linux X

Best Match   Newest   Popular

58 Apps

CATEGORY

- IT Operations
- Security, Fraud & Compliance
- Business Analytics
- Utilities
- IoT & Industrial Data
- DevOps
- Directory Service
- Email
- Endpoint
- Firewall
- Generic

**Splunk Add-on for Unix and Linux** **Install**

\*\*\* Important: Read upgrade Instructions and test add-on update before deploying to production \*\*\*

There are changes to default indexes and .conf changes in version 6.0 of Splunk Add-on for Unix and Linux that can break an existing installation if upgrade instructions are not followed in detail. If an existing Splunk Add-on for Unix and Linux is be... [More](#)

Category: IT Operations, Utilities | Author: Splunk Inc. | Downloads: 291534 |

Released: 2 months ago | Last Updated: 2 months ago | [View on Splunkbase](#)

Login and Install X

Enter your Splunk.com username and password to download the app.

[Forgot your password?](#)

The app, and any related dependency that will be installed, may be provided by Splunk and/or a third party and your right to use these app(s) is in accordance with the applicable license(s) provided by Splunk and/or the third-party licensor. Splunk is not responsible for any third-party app and does not provide any warranty or support. If you have any questions, complaints or claims with respect to an app, please contact the applicable licensor directly whose contact information can be found on the Splunkbase download page.

[Splunk Add-on for Unix and Linux](#) is governed by the following license:

[Splunk Software License Agreement](#)

I have read the terms and conditons of the license(s) and agree to be bound by them. I also agree to Splunk's [Website Terms of Use](#).

**Cancel** **Agree and Install**

# Detailed Parsing

- Once the add on installed search the data related to auth.
- Now you will find more data has been parsed.

Type	Field	Value	Actions
Selected	host	ip-172-31-9-191	
	source	auth.log	
	sourcetype	linux_secure	
Event	action	success	
	app	systemd-user	
	dest	ip-172-31-7-149	
	dvc	ip-172-31-7-149	
	eventtype	nix-all-logs nix_security (os_unix) nix_ta_data	
	name	session opened	
	process	systemd	
	tag	os unix	
	uid	0	
	user	root(uid=0)	
	user_id	0	
	user_name	root(uid=0)	
	vendor_action	session opened	
	vendor_product	nix	

# **Demo: Finding Attack vectors**

# Finding Attack vectors

---

You might be requested to find certain attack vector related to unauthorized login attempts.

Lets find out below requirements:

Sr. No	Requirement
1	Find total number of SSH failed login attempts
2	Find how many failed logins from every IP
3	Find list of Countries from which the failed login attempts were made
4	Create a visualization of Countries in world map based on failed logins

# SPL commands

---

We can use SPL commands in order to get the desired result. List of search commands can be find in the below given link.

<https://docs.splunk.com/Documentation/Splunk/9.1.0/SearchReference/ListOfSearchCommand>

S

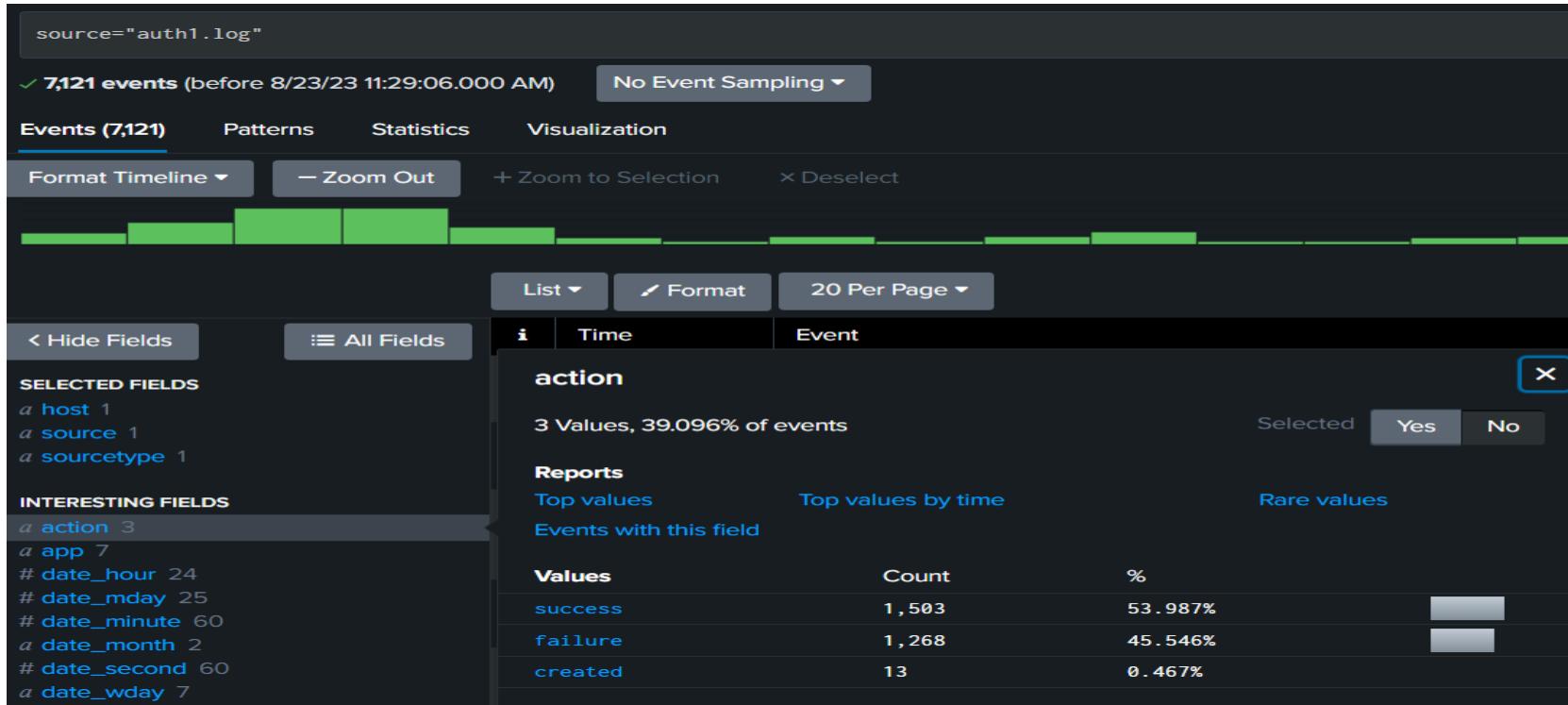
Command	Description	Related commands
<code>abstract</code>	Produces a summary of each search result.	<code>highlight</code>
<code>accum</code>	Keeps a running total of the specified numeric field.	<code>autoregress, delta, trendline, streamstats</code>
<code>addcoltotals</code>	Computes an event that contains sum of all numeric fields for previous events.	<code>addtotals, stats</code>
<code>addinfo</code>	Add fields that contain common information about the current search.	<code>search</code>
<code>addtotals</code>	Computes the sum of all numeric fields for each result.	<code>addcoltotals, stats</code>
<code>analyzefields</code>	Analyze numerical fields for their ability to predict another discrete field.	<code>anomalousvalue</code>
<code>anomalies</code>	Computes an "unexpectedness" score for an event.	<code>anomalousvalue, cluster, kmeans, outlier</code>
<code>anomalousvalue</code>	Finds and summarizes irregular, or uncommon, search results.	<code>analyzefields, anomalies, cluster, kmeans, outlier</code>
<code>anomalydetection</code>	Identifies anomalous events by computing a probability for each event and then detecting unusually small probabilities.	<code>analyzefields, anomalies, anomalousvalue, cluster, kmeans, outlier</code>
<code>append</code>	Appends subsearch results to current results.	<code>appendcols, appendcsv, appendlookup, join, set</code>
<code>appendcols</code>	Appends the fields of the subsearch results to current results. first results to first result, second to	<code>append, appendcsv, join, set</code>

# 1. Find total number of SSH failed login attempts

---

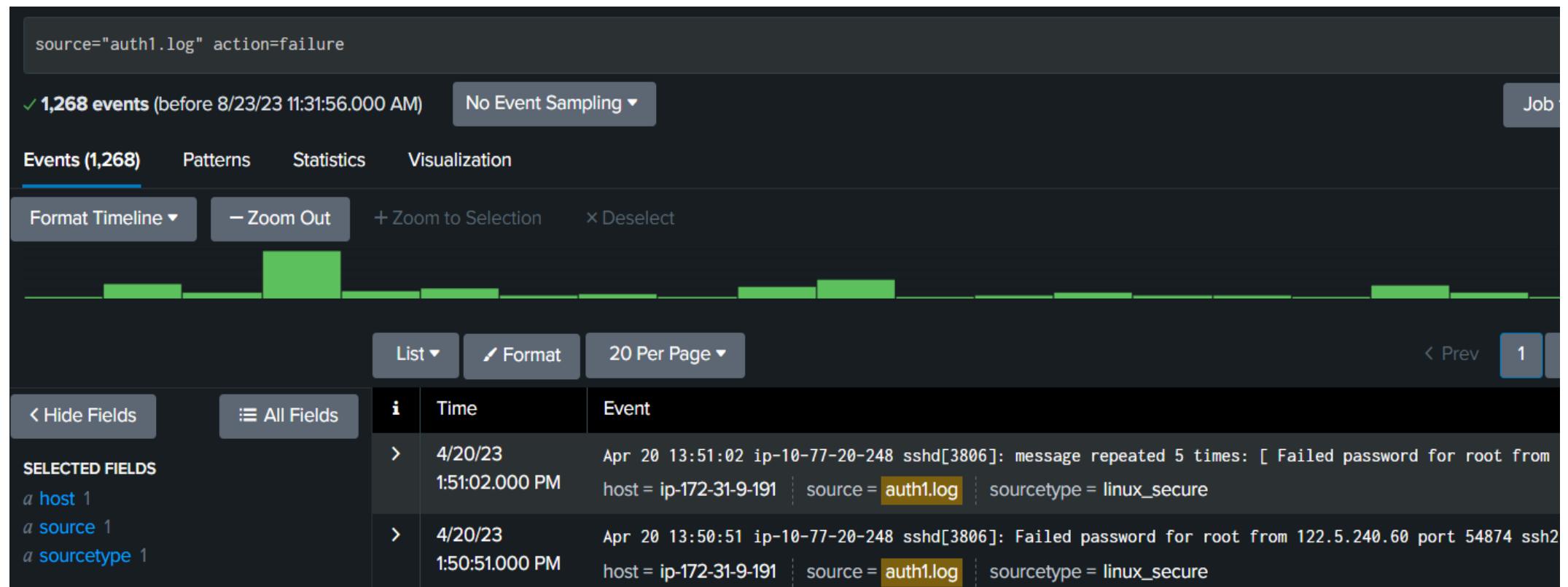
From our uploaded data we can find the number of attempts made under action field.

**Success: 53.987% , Failure : 45.546%**



# 1. Find total number of SSH failed login attempts

**source="auth1.log" action=failure** will give us the desired result



## 2. Find how many failed logins from every IP

To get this result we will use **stats** from SPL command.

It calculates aggregate statistics, such as average, count, and sum over the result set.

Stats function options				
stats-func				
<b>Syntax:</b> The syntax depends on the function that you use. Refer to the table below.				
<b>Description:</b> Statistical and charting functions that you can use with the <code>stats</code> command. Each time you invoke the <code>stats</code> command, you can use one or more functions. However, you can only use one <code>BY</code> clause. See <a href="#">Usage</a> .				
The following table lists the supported functions by type of function. Use the links in the table to see descriptions and examples for each function. For an overview about using functions with commands, see <a href="#">Statistical and charting functions</a> .				
Type of function	Supported functions and syntax			
Aggregate functions	<code>avg()</code> <code>count()</code> <code>distinct_count()</code> <code>estdc()</code> <code>estdc_error()</code>	<code>exactperc&lt;num&gt;()</code> <code>max()</code> <code>median()</code> <code>min()</code> <code>mode()</code>	<code>perc&lt;num&gt;()</code> <code>range()</code> <code>stdev()</code> <code>stdevp()</code>	<code>sum()</code> <code>sumsq()</code> <code>upperperc&lt;num&gt;()</code> <code>var()</code> <code>varp()</code>
Event order functions	<code>first()</code>	<code>last()</code>		
Multivalue stats and chart functions	<code>list()</code>	<code>values()</code>		
Time functions	<code>earliest()</code> <code>earliest_time()</code>	<code>latest()</code> <code>latest_time()</code>	<code>rate()</code>	

## 2. Find how many failed logins from every IP

---

So we can use stats with count function to get the desired result.

**source="auth1.log" action=failure | stats count by src**

The screenshot shows a Splunk search interface with the following details:

- Search Bar:** source="auth1.log" action=failure | stats count by src
- Results Summary:** 1,268 events (before 8/23/23 11:45:41.000 AM)
- Event Sampling:** No Event Sampling
- Time Range:** All time
- Job Status:** Job
- Visual Mode:** Smart Mode
- Statistics View:** Statistics (130) is selected.
- Table Headers:** src (sorted), count (sorted)
- Data Rows:** A list of IP addresses and their corresponding failed login counts.

src	count
1.189.205.173	3
1.30.211.144	9
103.230.120.26	9
105.101.221.33	9
106.57.58.19	3
110.78.174.75	9
111.40.166.130	9
111.40.168.90	3

### 3. Find list of Countries from which the failed login attempts were made

---

For this information we need another SPL command **iplocation**.

It extracts information from IP address by using 3<sup>rd</sup> party databases. This command supports IPv4 and IPv6.

The IP address that you specify in the ip-address-fieldname argument, is looked up in a database. Fields from that database that contain location information are added to each event. The setting of the allfields argument determines which fields are added to the events.

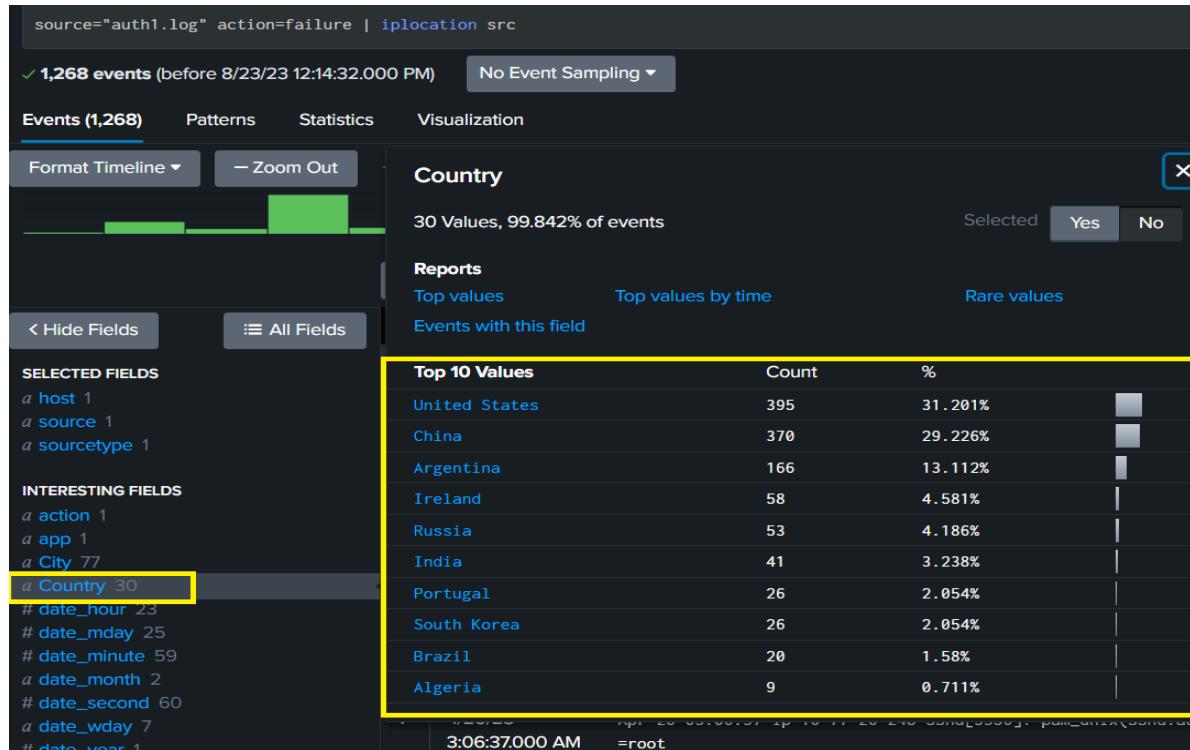
Because all the information might not be available for each IP address, an event can have empty field values.

For IP addresses which do not have a location, such as internal addresses, no fields are added.

# 3. Find list of Countries from which the failed login attempts were made

We can use the below SPL command to get the desired result.

```
source="auth1.log" action=failure | iplocation src
```



# 4. Create a visualization of Countries in world map based on failed logins

---

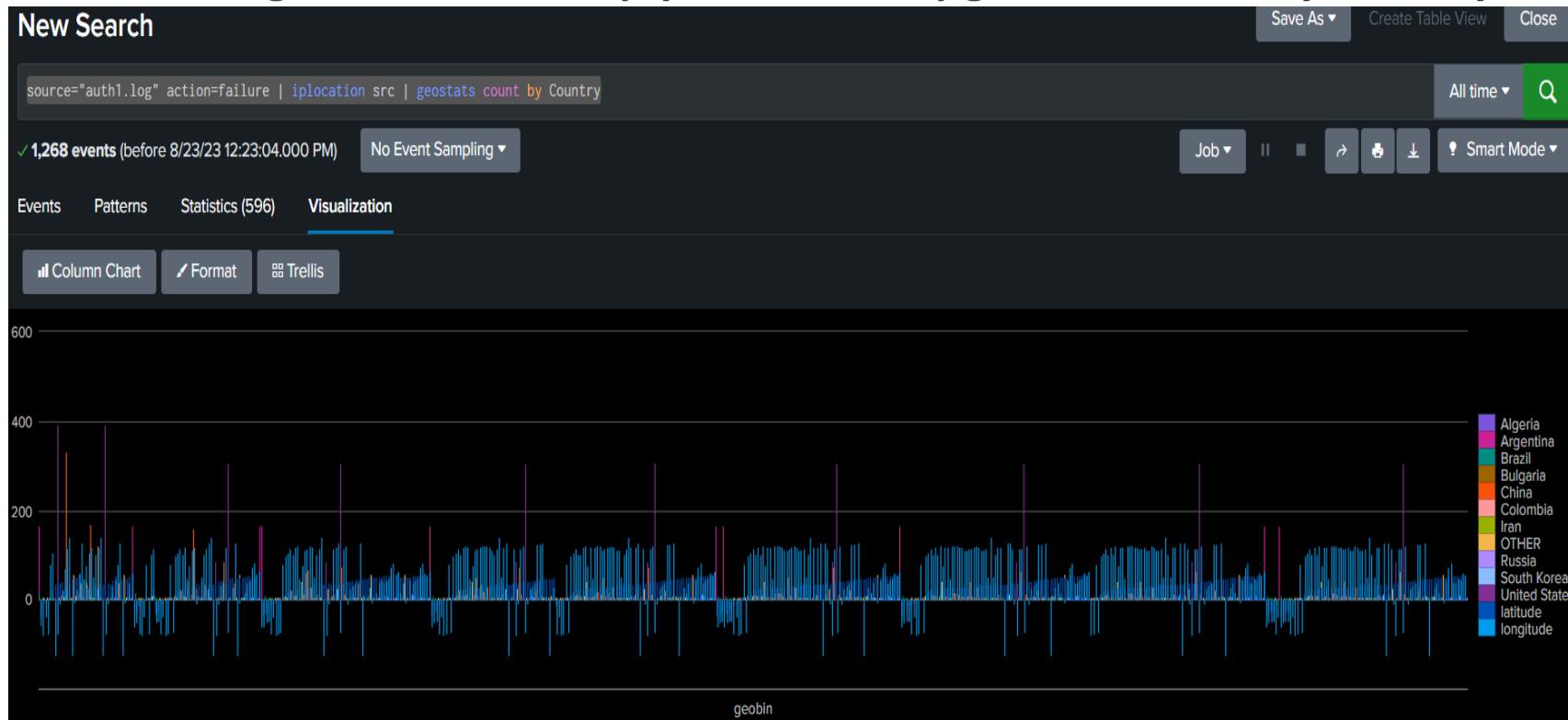
Use the **geostats** command to generate statistics to display geographic data and summarize the data on maps.

The command generates statistics which are clustered into geographical bins to be rendered on a world map. The events are clustered based on latitude and longitude fields in the events. Statistics are then evaluated on the generated clusters. The statistics can be grouped or split by fields using a BY clause.

# 4. Create a visualization of Countries in world map based on failed logins

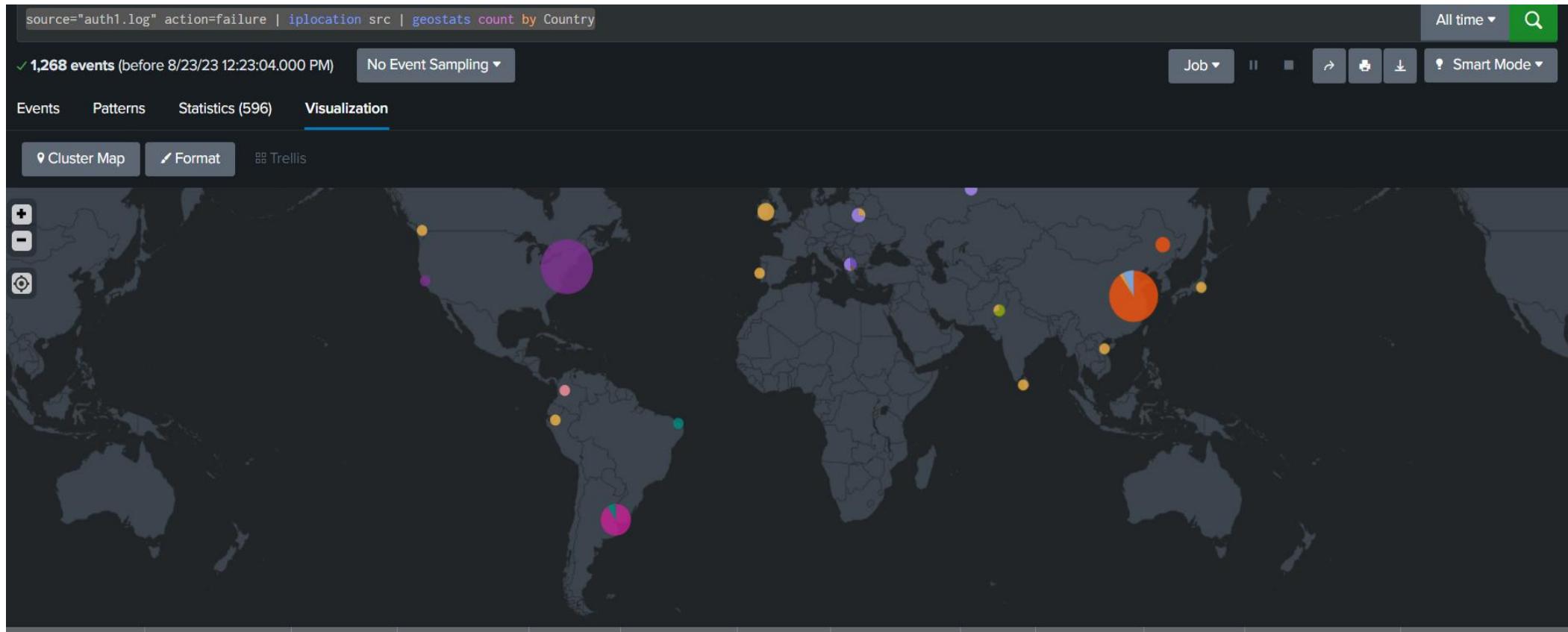
Use the below command to get the desired result.

```
source="auth1.log" action=failure | iplocation src | geostats count by Country
```



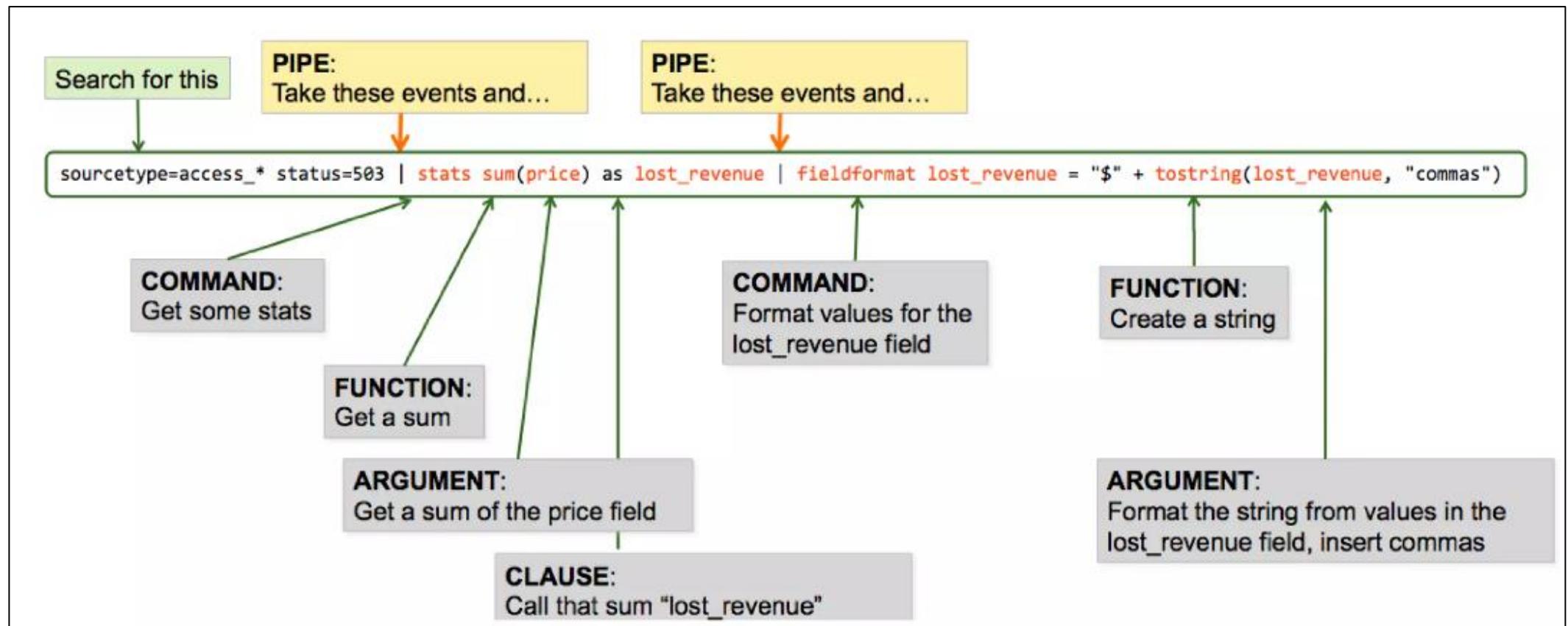
# 4. Create a visualization of Countries in world map based on failed logins

Another graph (recommended)



# **Basic Search**

# Search Syntax Components



# Search basics

---

The easiest way to search for a specific data is to type in the search string.

The string should be the exact match.

String	
failure	Give result about the events having “failure” or “Failure”. (exact match)
fail	Give result about the events having exact string “fail” or “Fail” but will not include the string starting with “fail”
Fail*	Give result about the events having string started with “fail”

# Search basics

## Time Range Picker:

We can filter our search using a time range which is an easiest and most effective way to optimize the searches.

Time range can also be used for troubleshooting at a specific timeframe.

New Search

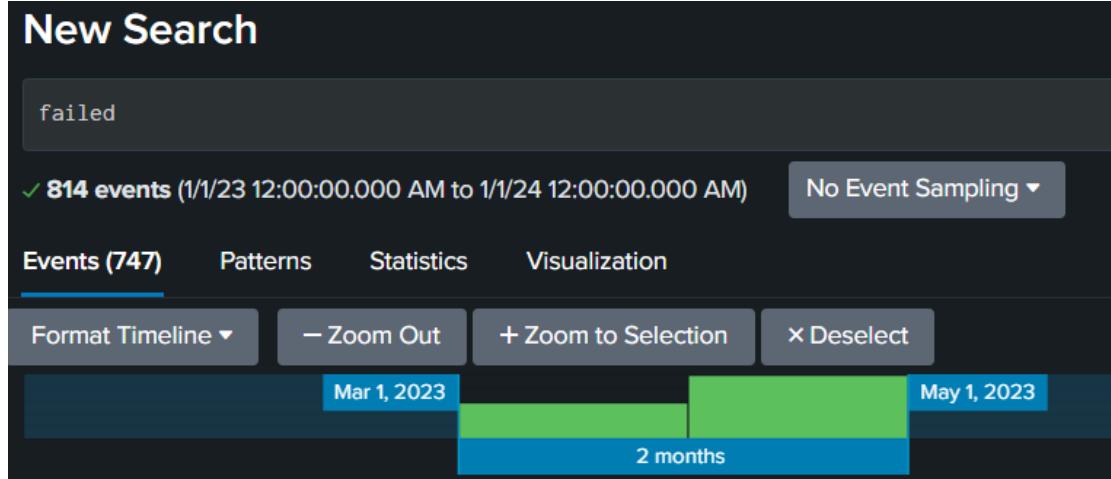
failed

✓ 814 events (1/1/23 12:00:00.000 AM to 1/1/24 12:00:00.000 AM) No Event Sampling ▾

Events (747) Patterns Statistics Visualization

Format Timeline ▾ – Zoom Out + Zoom to Selection × Deselect

Mar 1, 2023 [redacted] May 1, 2023  
2 months



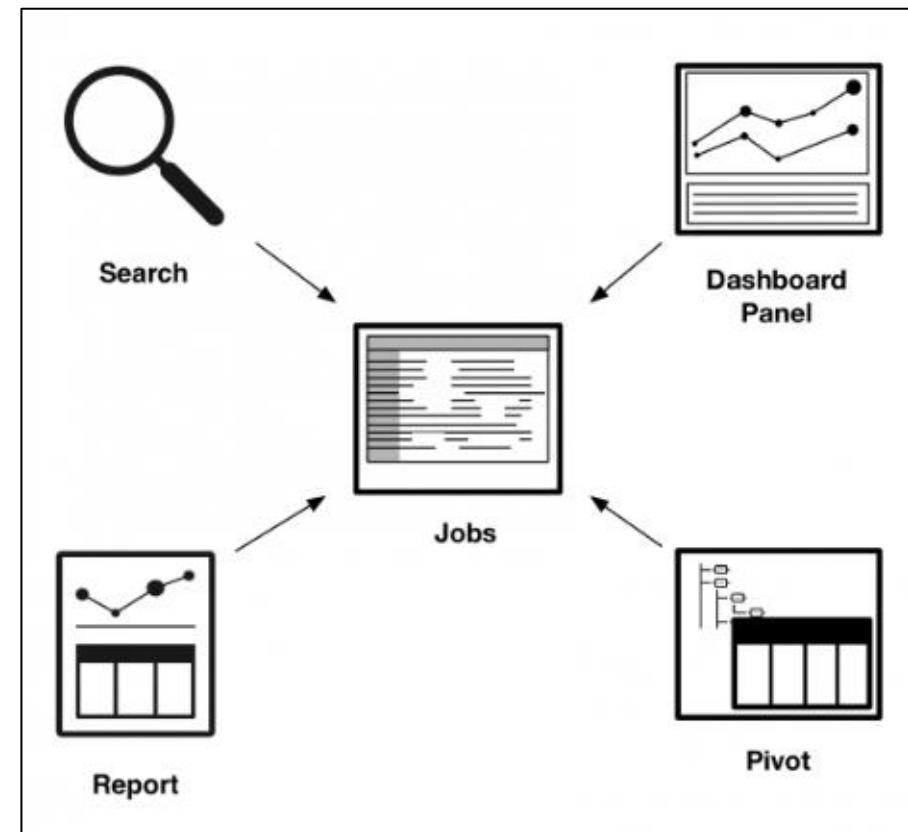
# Search: Jobs and Jobs management

Each time you run a search, create a pivot, open a report, or load a dashboard panel, the Splunk software creates a job in the system.

When you run a search, you are creating an ad hoc search. Pivots, reports, and panels are powered by saved searches.

A job is a process that tracks information about the ad hoc search or saved search. The information that is tracked includes the owner of the job, the app that the job was run on, how many events were returned, and how long the job took to run.

Each job process creates a **search artifact**. The artifact contains the results and associated metadata

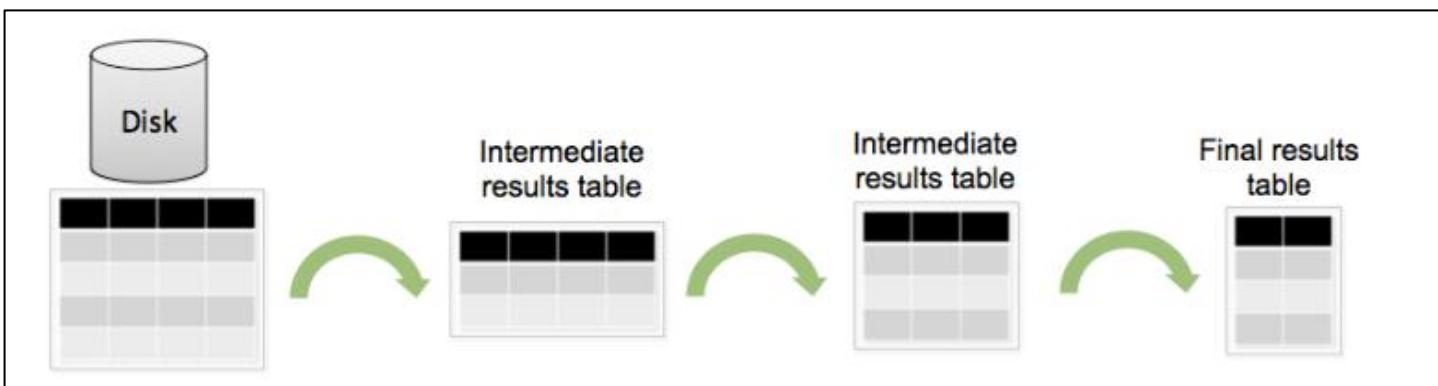


# Search Pipeline

It helps to visualize all your indexed data as a table. Each search command redefines the shape of your table.

The "search pipeline" refers to the structure of a Splunk search, in which consecutive commands are chained together using a pipe character, "|". The pipe character tells Splunk software to use the output or result of one command (to the left of the pipe) as the input for the next command (to the right of the pipe).

```
|sourcetype=syslog ERROR | top user | fields - percent
```



# Quotes and Escaping Characters

---

- you need quotes around phrases and field values that include white spaces, commas, pipes, quotes, or brackets.
- The backslash character (\) is used to escape quotes, pipes, and itself

Escaping double quotes	index=myindex "This is a \"quoted\" string"
Escaping pipes	index=myindex source="some\\ file.log"
Escaping Backslash itself	index=myindex "C:\\Windows\\System32\\file.exe"

# Search: Inspecting Jobs

---

You can inspect a job or you can manage a job using below methods.

Search Job Inspector	Use the Search Job Inspector to view information about the current job, such as job execution costs and search job properties. You get a closer look at what your search is doing and see where the Splunk software is spending most of its time.
Job Details dashboard	The Job Details dashboard provides a clear and concise overview of a search job process. You can access the Job Details dashboard through the Search Job Inspector.
Jobs manager page	Use the Jobs manager page to view information about recent jobs. If you have the Admin role or a role with an equivalent set of capabilities, you can manage the search jobs run by other users.

# Search: Jobs Menu

---

- Edit the job settings
- Send the job to the background.
- Inspect the job
- Delete the job
- Sharing Jobs
- Jobs lifetime (Default: 10min)
- Auto-pause long running jobs (Enabled by default for Summary dashboards)
- Managing jobs when search web running computer goes to sleep mode

# Search: Administering Jobs

---

Administering jobs (to restrict how many jobs a given user can run, and how much space their job artifacts can take up)

Scope	Description
System-wide	Create the authorize.conf file in local directory for the system under \$SPLUNK_HOME/etc/system/local .
Application Specific	Create the authorize.conf file in the local directory for the application under \$SPLUNK_HOME/etc/apps/<app_name>/local.

# Search: Administering Jobs

---

[role\_ninja]

rtsearch = enabled (for real time searches)

importRoles = user

srchFilter = host=foo

srchIndexesAllowed = \*

srchIndexesDefault = mail;main

srchJobsQuota = 8 (The default value is 3)

rtSrchJobsQuota = 8

srchDiskQuota = 500 (The default value is 100MB)

srchTimeWin = 86400 (allowed to run searches that span a maximum of one day)

srchTimeEarliest = 2592000 ( allowed to run searches on data that is newer than 30 days ago)

# Search: Extending Job lifetime

---

When you run a new search job, the job is retained in the system for a period of time, called the **job lifetime**. During the lifetime, you can access the job and view the data returned by the job. If the job is not accessed within the specified lifetime, the job expires and is removed from the system.

There are two lifetime settings, 10 minutes and 7 days.

**Unscheduled Searches** : Default is 10 minutes

**Scheduled Searches** : The interval of the scheduled search multiplied by two. For example, if the search runs every 6 hours, the resulting jobs expire in 12 hours.

Whenever you access an active job, such as when you view the results of a search job, the lifetime is reset.

# Search: Extending Job Lifetime

Jobs	
Unscheduled Jobs	<p>Open the local <b>limits.conf</b> file for the <b>Search</b> app. For example, <code>\$SPLUNK_HOME/etc/apps/&lt;app_name&gt;/local</code>. In the <b>[search]</b> stanza, change the <b>default_save_ttl</b> value to a number that is appropriate for your needs in seconds, and defaults to 604800 seconds, or one week.</p>
Scheduled Jobs	<p>Open the local <b>savedsearches.conf</b> file. For example, <code>\$SPLUNK_HOME/etc/apps/&lt;app_name&gt;/local</code>. Locate the scheduled search, and change the <b>dispatch.ttl</b> setting to a different interval multiple</p>

# Search: Jobs Dispatch Directory and Search Artifacts

---

Each search or alert you run creates a search artifact that must be saved to disk. For each search job, there is one search-specific directory. When the job expires, the search-specific directory is deleted.

The path to the dispatch directory is `$SPLUNK_HOME/var/run/splunk/dispatch`.

The dispatch directory reaper iterates over all of the artifacts every 30 seconds. The reaper deletes artifacts that have expired, based on the last time that the artifacts were accessed and their configured time to live (TTL), or lifetime.

As more and more artifacts are added to the dispatch directory, it is possible that the volume of artifacts will cause a adverse effect on search performance or that a warning appears in the UI. The **warning threshold is based on the `dispatch_dir_warning_size` attribute in the `limits.conf` file**.

The default value for the `dispatch_dir_warning_size` attributes is **5000**.

You can move search-specific directories from the dispatch directory to another, destination, directory. The destination directory must be on the same file system as the dispatch directory.

Run the command `$SPLUNK_HOME/bin/splunk clean-dispatch help` to learn how to use the clean-dispatch command.

# Search: Limits memory usage by a Job

---

Splunk software can be configured to automatically terminate search job processes that exceed a threshold of a configured quantity of resident memory in use.

Steps:

- Use the `$SPLUNK_HOME/etc/apps/search/local` path to apply this change only to the Search app.
- Under the [search] stanza, change the setting for the `enable_memory_tracker` setting to `true`.  
You can set the limit to an absolute amount or a percentage of the identified system maximum, using `search_process_memory_usage_threshold` or `search_process_memory_usage_percentage_threshold`, respectively.
- To enable the configuration changes, restart Splunk Enterprise.

# Search: What is a Field and how to use it

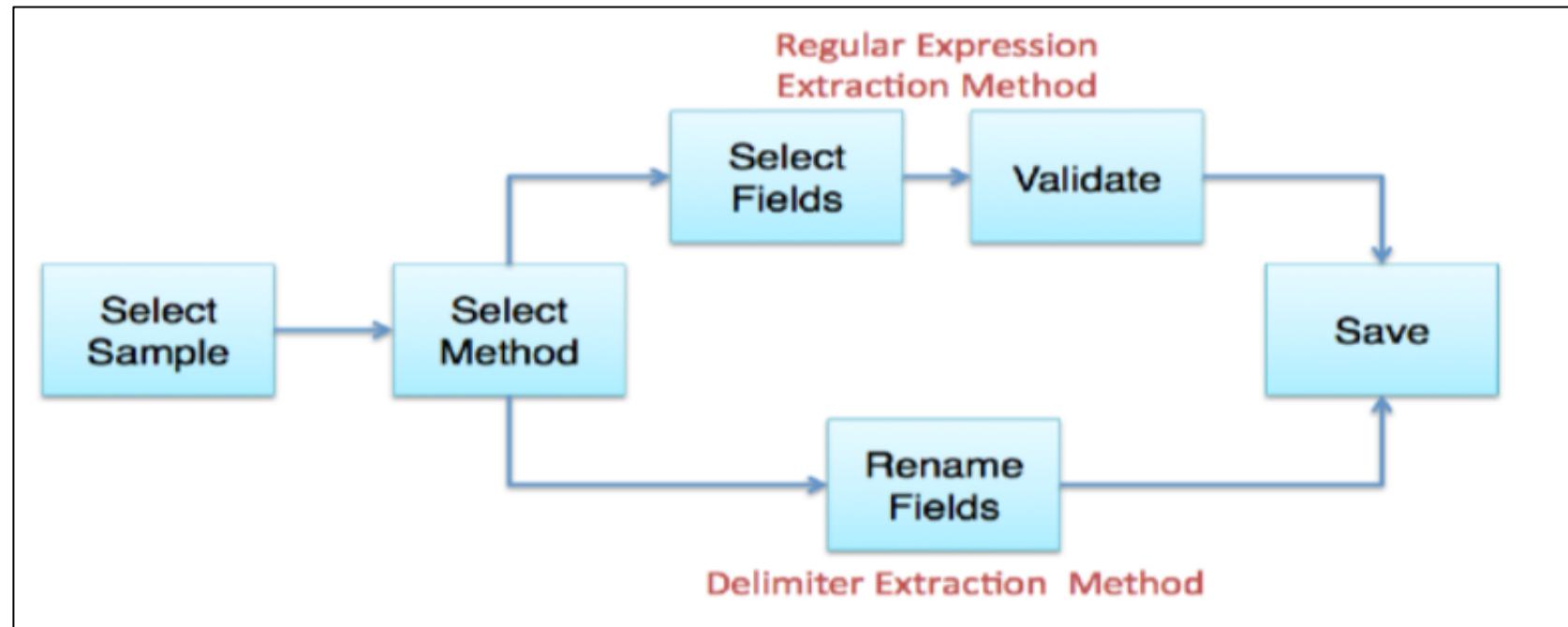
---

The `fields` command specifies which fields to keep or remove from the search results. By default, the internal fields `_raw` and `_time` are included in the output.

Commands	Description
<code>  fields name, salary</code>	Include only name and salary
<code>  fields - country, age</code>	Exclude only country and age
<code>  fields - *_</code>	Exclude <code>_raw</code> and <code>_time</code>
<code>  fields - _raw</code>	Exclude only <code>_raw</code>
<code>  fields - '_*', host, src</code>	Exclude <code>_raw</code> , <code>_time</code> , <code>host</code> and <code>src</code>

# Search: Build Field extractions with field extractor

Use the **field extractor** utility to create new fields. The field extractor provides two field extraction methods: regular expression and delimiters.



# Search: Build Field extractions with field extractor

---

**Regular Expression:** if the event that you have selected is derived from unstructured data such as a system log. The field extractor can attempt to generate a regular expression that matches similar events and extracts your fields.

**Delimiters** if the fields in your selected event are:

- cleanly separated by a common delimiter, such as a space, a comma, or a pipe character.
- consistent across multiple events (each value is in the same place from event to event).

# Search Basics: Boolean Expressions

---

The Splunk search processing language (SPL) supports the Boolean operators:

Operators	
AND	implied between terms, so you do not need to write it.
OR	used to specify that either one of two or more arguments should be true.
NOT	used to filter out events containing a specific word.

# Boolean Expressions : Use cases

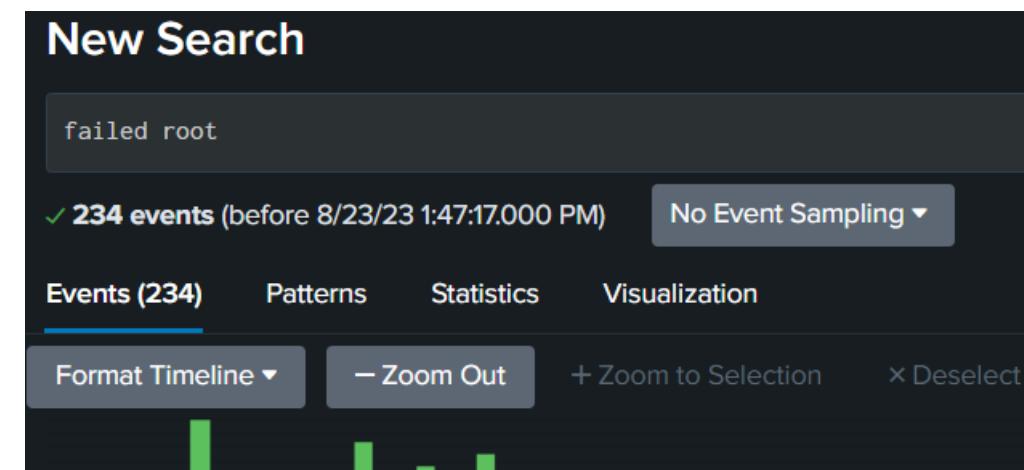
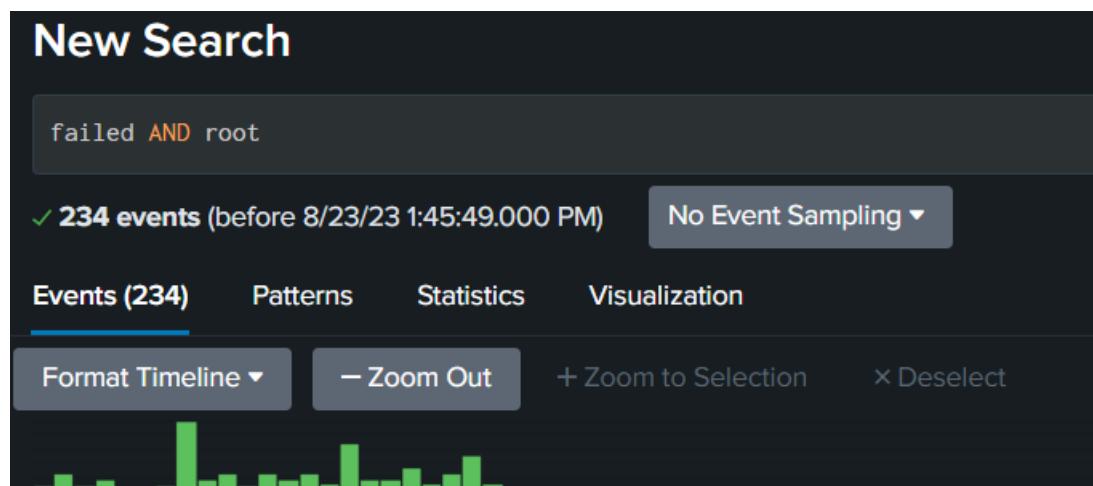
---

Use Cases	SPL
Search for all failed login attempts by user root	Root AND failed
Search for all failed logins for all user except root	Failed NOT root
Search failed logins for user root or admin	Failed admin OR root

# Boolean Expressions : Use cases

**AND operator:** (Search for all failed login attempts by user root)

Below both will give the same result



# Boolean Expressions : Use cases

**NOT operator:** Search for all failed logins for all user except root

New Search

```
failed NOT root
```

✓ 580 events (before 8/23/23 1:51:18.000 PM) No Event Sampling ▾

Events (580) Patterns Statistics Visualization

Format Timeline ▾    - Zoom Out    + Zoom to Selection    × Details



i	Time	Event
>	8/20/23 12:25:50	Aug 20 12:25:50 ip-172-31-7-149 sshd[17875]: Failed password for invalid user p from 45.162.37.27 port 40462 ssh2 host = ip-172-31-9-191   source = auth.log   sourcetype = linux_secure
>	8/20/23 12:21:45	Aug 20 12:21:45 ip-172-31-7-149 sshd[17077]: Failed password for invalid user kuku from 45.162.37.27 port 47500 ssh2 host = ip-172-31-9-191   source = auth.log   sourcetype = linux_secure
>	8/20/23 12:19:43	Aug 20 12:19:43 ip-172-31-7-149 sshd[16620]: Failed password for invalid user postgres from 45.162.37.27 port 51014 ssh2 host = ip-172-31-9-191   source = auth.log   sourcetype = linux_secure
>	8/20/23 12:18:37	Aug 20 12:18:37 ip-172-31-7-149 sshd[16421]: Failed password for invalid user irene from 45.162.37.27 port 38662 ssh2 host = ip-172-31-9-191   source = auth.log   sourcetype = linux_secure
>	8/20/23 12:17:38	Aug 20 12:17:38 ip-172-31-7-149 sshd[16218]: Failed password for invalid user scanner from 45.162.37.27 port 54536 ssh2 host = ip-172-31-9-191   source = auth.log   sourcetype = linux_secure
>	8/20/23 12:16:40	Aug 20 12:16:40 ip-172-31-7-149 sshd[15991]: Failed password for invalid user slim from 45.162.37.27 port 42180 ssh2 host = ip-172-31-9-191   source = auth.log   sourcetype = linux_secure
>	8/20/23 12:15:40	Aug 20 12:15:40 ip-172-31-7-149 sshd[15792]: Failed password for invalid user rh from 45.162.37.27 port 58050 ssh2 host = ip-172-31-9-191   source = auth.log   sourcetype = linux_secure
>	8/20/23 12:14:40	Aug 20 12:14:40 ip-172-31-7-149 sshd[15593]: Failed password for invalid user test from 45.162.37.27 port 45692 ssh2 host = ip-172-31-9-191   source = auth.log   sourcetype = linux_secure

# Boolean Expressions : Use cases

**OR operator:** Search failed logins for user root or admin

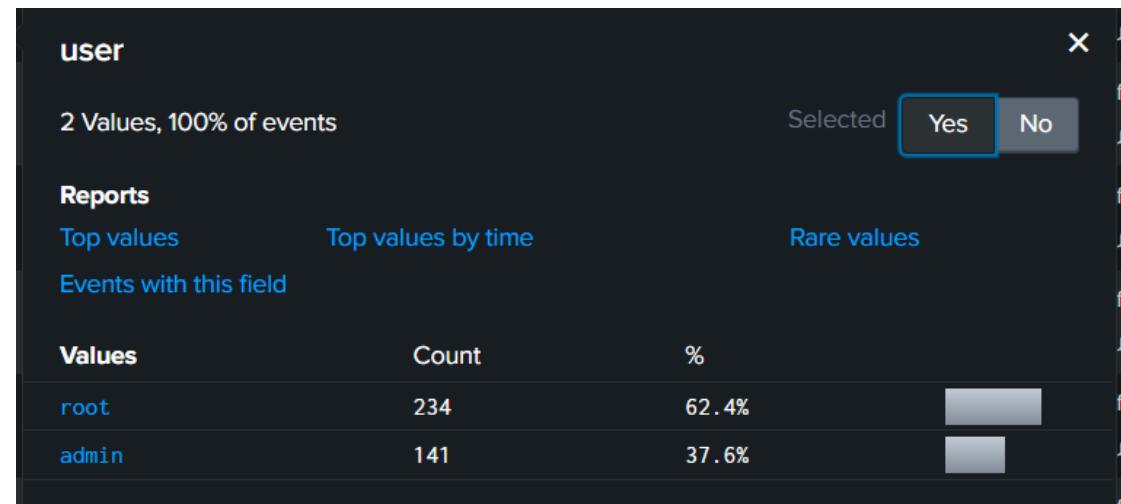
New Search

```
failed AND (root OR admin)
```

✓ 375 events (before 8/23/23 1:57:45.000 PM) No Event Sampling ▾

Events (375) Patterns Statistics Visualization

Format Timeline ▾ - Zoom Out + Zoom to Selection × Deselect



# Search and Filter

---

1	<b>Keyword Search :</b>
	Sourcetype=access* http
2	<b>Filter :</b>
	Sourcetype=access* http host=webserver-01
3	<b>Combined :</b>
	Sourcetype=access* http host=webserver-01 (503 or 504)

# Eval: Modify or Create New fields and Values

Common Eval Functions		The eval command calculates an expression and puts the resulting value into a field (e.g. "...  eval force = mass * acceleration"). The following table lists some of the functions used with the eval command. You can also use basic arithmetic operators (+ - * / %), string concatenation (e.g., "...  eval name = last . ",". first"), and Boolean operations (AND OR NOT XOR < > <= >= != == LIKE).
Function	Description	Examples
<code>abs(X)</code>	Returns the absolute value of X.	<code>abs(number)</code>
<code>case(X,"Y",...)</code>	Takes pairs of arguments X and Y, where X arguments are Boolean expressions. When evaluated to TRUE, the arguments return the corresponding Y argument.	<code>case(error == 404, "Not found", error == 500,"Internal Server Error", error == 200, "OK")</code>
<code>ceil(X)</code>	Ceiling of a number X.	<code>ceil(1.9)</code>
<code>cidrmatch("X",Y)</code>	Identifies IP addresses that belong to a particular subnet.	<code>cidrmatch("123.132.32.0/25",ip)</code>
<code>coalesce(X,...)</code>	Returns the first value that is not null.	<code>coalesce(null(), "Returned val", null())</code>
<code>cos(X)</code>	Calculates the cosine of X.	<code>n=cos(0)</code>
<code>exact(X)</code>	Evaluates an expression X using double precision floating point arithmetic.	<code>exact(3.14*num)</code>
<code>exp(X)</code>	Returns eX.	<code>exp(3)</code>
<code>if(X,Y,Z)</code>	If X evaluates to TRUE, the result is the second argument Y. If X evaluates to FALSE, the result evaluates to the third argument Z.	<code>if(error==200, "OK", "Error")</code>
<code>isbool(X)</code>	Returns TRUE if X is Boolean.	<code>isbool(field)</code>
<code>isint(X)</code>	Returns TRUE if X is an integer.	<code>isint(field)</code>
<code>isnull(X)</code>	Returns TRUE if X is NULL.	<code>isnull(field)</code>
<code>isstr()</code>	Returns TRUE if X is a string.	<code>isstr(field)</code>
<code>len(X)</code>	This function returns the character length of a string X.	<code>len(field)</code>
<code>like(X,"Y")</code>	Returns TRUE if and only if X is like the SQLite pattern in Y.	<code>like(field, "addr%")</code>

# Eval: Modify or Create New fields and Values

---

1	<b>Calculation :</b>
	Sourcetype=access*   eval KB=bytes/1024
2	<b>Evaluation:</b>
	Sourcetype=access*   eval http_response;if(status!=200,"Error","OK")
3	<b>Concatenation:</b>
	Sourcetype=access*   eval connection = device.":".clientip

# Transforming Commands

---

A type of search command that orders the results into a data table. Transforming commands "transform" the specified cell values for each event into numerical values that Splunk Enterprise can use for statistical purposes. Searches that use transforming commands are called [transforming searches](#).

Transforming commands include

1. chart
2. timechart
3. stats
4. top
5. rare
6. contingency
7. highlight.

# Transforming Commands: Chart

---

The chart command is used to transform data into a tabular format suitable for charting and visualization.

It's one of the many commands you can use to analyze and visualize your data effectively. The chart command groups and aggregates data based on specified fields and functions and presents the results in a table format that can be used for creating charts and graphs.

```
| chart <aggregation_function>(field) AS alias BY group_field
```

**aggregation\_function:** The aggregation function to apply to the specified field. Common aggregation functions include count, sum, avg, min, max, etc.

**field:** The field you want to aggregate.

**alias:** An optional alias name for the resulting field.

**BY group\_field:** An optional clause that specifies how the data should be grouped by a particular field. You can group data by multiple fields by separating them with commas.

# Transforming Commands: Chart

Count the number of events by a specific field:	<code>...   chart count(event) as EventCount by status</code>
Calculate the average response time by user:	<code>...   chart avg(response_time) as AvgResponseTime by user</code>
Find the sum of sales by product category and month:	<code>...   chart sum(sales) as TotalSales by product_category, month</code>
Count unique values in a field:	<code>...   chart dc(ip_address) as UniqueIPs by country</code>
Stacked column chart with multiple values:	<code>...   chart sum(value1) as Value1, sum(value2) as Value2 by category</code>

# Transforming Commands: Timechart

---

The timechart command is used to transform and summarize data over time, making it particularly useful for time-series analysis and visualization. The timechart command aggregates data based on a specified time field and allows you to perform various statistical and aggregation functions on that data. Here's the basic syntax of the timechart command:

```
| ... | timechart <aggregation_function>(field) as alias
```

**<aggregation\_function>**: The aggregation function you want to apply to the specified field over time. Common functions include sum, avg, min, max, count, etc.

**field**: The field you want to aggregate over time.

**alias**: An optional alias name for the resulting time-series column.

**\_time**: This field gets added automatically

# Transforming Commands: Timechart

Summarize events over time:	...   timechart count as EventCount
Calculate the average response time over time:	...   timechart avg(response_time) as AvgResponseTime
Find the maximum temperature per day:	...   timechart max(temperature) as MaxTemperature by day
Count the number of errors per hour:	...   timechart count(status=error) as ErrorCount span=1h
Create a time series of unique users per day:	...   timechart dc(user) as UniqueUsers span=1d
Calculate the total sales per week:	...   timechart sum(sales) as TotalSales span=1w

# Timechart- Visualize Statistics Over Time

---

1	<b>Visualize stats over time:</b>
	Index=new-index   timechart avg(bytes)
2	<b>Add a tredline:</b>
	Index=new-index   timechart avg(bytes) as bytes   tredline sma5(bytes)
3	<b>Add a prediction overlay:</b>
	Index=new-index   timechart avg(bytes) as bytes   predict future_timespan=5 bytes

# Transforming Commands: Stats

---

The stats command is used for statistical transformations of your data. It's a powerful command that allows you to perform various statistical calculations and aggregations on your search results. Here's the basic syntax of the stats command:

```
| ... | stats <aggregation_function>(field) as alias
```

**<aggregation\_function>**: The aggregation function you want to apply to the specified field. Common aggregation functions include count, sum, avg (average), min, max, list, values, dc (distinct count), and more.

**field**: The field on which you want to perform the aggregation.

**alias**: An optional alias name for the resulting field.

# Transforming Commands: Stats

---

Here are some common use cases for the stats command:

- **Aggregation:** Calculate sums, averages, counts, minimums, maximums, etc., for numeric fields.
- **Grouping:** Group data based on one or more fields using the by clause to perform aggregations within those groups.
- **Distinct Count:** Calculate the distinct count of values in a field (dc(field)).
- **List Values:** Create a list of values for a field within a group (list(field)).
- **Percentiles:** Calculate percentiles for numeric fields (percentile(field, p)).
- **Advanced Statistics:** Perform more advanced statistical operations like standard deviation, variance, skewness, and kurtosis.

# Stats: Calculate Statistics based on Field Values

---

1	<b>Calculate stats and rename:</b>
	index=new-index   stats avg(bytes) AS "Avg Bytes"
2	<b>Multiple Statistics:</b>
	index=new-index   stats avg(bytes) AS "Avg Bytes" sparkline(avg(bytes)) AS Bytes_Trend min(bytes) max(bytes)
3	<b>By another field: (A sparkline is a small representation of some statistical information without showing the axes) and (sorting in descending order by avg_bytes)</b>
	index=new-index   stats avg(bytes) AS "Avg Bytes" sparkline(avg(bytes)) AS Bytes_Trend min(bytes) max(bytes) by clientip   sort - avg_bytes

# Common Stats Function

Common Stats Functions		Common statistical functions used with the chart, stats, and timechart commands. Field names can be wildcarded, so avg(*delay) might calculate the average of the delay and xdelay fields.
<b>avg(X)</b>	Returns the average of the values of field X.	
<b>count(X)</b>	Returns the number of occurrences of the field X. To indicate a specific field value to match, format X as eval(field="value").	
<b>dc(X)</b>	Returns the count of distinct values of the field X.	
<b>earliest(X)</b>	Returns the chronologically earliest seen value of X.	
<b>latest(X)</b>	Returns the chronologically latest seen value of X.	
<b>max(X)</b>	Returns the maximum value of the field X. If the values of X are non-numeric, the max is found from alphabetical ordering.	
<b>median(X)</b>	Returns the middle-most value of the field X.	
<b>min(X)</b>	Returns the minimum value of the field X. If the values of X are non-numeric, the min is found from alphabetical ordering.	
<b>mode(X)</b>	Returns the most frequent value of the field X.	
<b>perc&lt;X&gt;(Y)</b>	Returns the X-th percentile value of the field Y. For example, perc5(total) returns the 5th percentile value of a field "total".	
<b>range(X)</b>	Returns the difference between the max and min values of the field X.	
<b>stdev(X)</b>	Returns the sample standard deviation of the field X.	
<b>stdevp(X)</b>	Returns the population standard deviation of the field X.	
<b>sum(X)</b>	Returns the sum of the values of the field X.	
<b>sumsq(X)</b>	Returns the sum of the squares of the values of the field X.	
<b>values(X)</b>	Returns the list of all distinct values of the field X as a multi-value entry. The order of the values is alphabetical.	
<b>var(X)</b>	Returns the sample variance of the field X.	

# Transforming commands: Top

---

The top command in Splunk's Search Processing Language (SPL) is used to identify and display the top values in a result set based on a specific field. It's a useful command for finding the most significant or frequently occurring values in your data.

The top command is valuable for identifying outliers, frequent occurrences, or significant values within your data. It's commonly used in data exploration, anomaly detection, and generating summary reports.

```
| ... | top limit=<number> field=<fieldname>
```

**limit:** Specifies the number of top values you want to display.

**field:** Specifies the field based on which you want to find the top values.

# Transforming commands: Top

top 10 values in the user field along with their frequencies.	...   top limit=10 field=user
top 10 users in each department based on their frequency.	...   top limit=10 field=user by=department
top values in the user field that collectively account for 10% of the total.	...   top limitperc=10 field=user
top 10 cities based on the unique count of users within each city.	...   top limit=10 field=city countfield=user
top 5 product_id values based on the sum of quantity_sold for each product.	...   top limit=5 field=product_id countfield=quantity_sold

# Transforming commands: Top

---

the top 3 department values based on the average salary within each department.	...   top limit=3 field=department countfield=salary
the top 10 users based on their frequency of occurrence in the last 24 hours.	...   top limit=10 field=user span=24h
the top 5 users in the "Sales" department based on their frequency	...   where department="Sales"   top limit=5 field=user

# Transforming commands: Rare

---

The rare command in Splunk's Search Processing Language (SPL) is used to find and display rare or infrequent values in a specific field within your search results. It's essentially the opposite of the top command, which identifies the most frequent values. The rare command helps you spot anomalies or unusual occurrences in your data. Here's the basic syntax of the rare command:

```
| ... | rare limit=<number> field=<fieldname>
```

**limit:** Specifies the number of rare values you want to display.

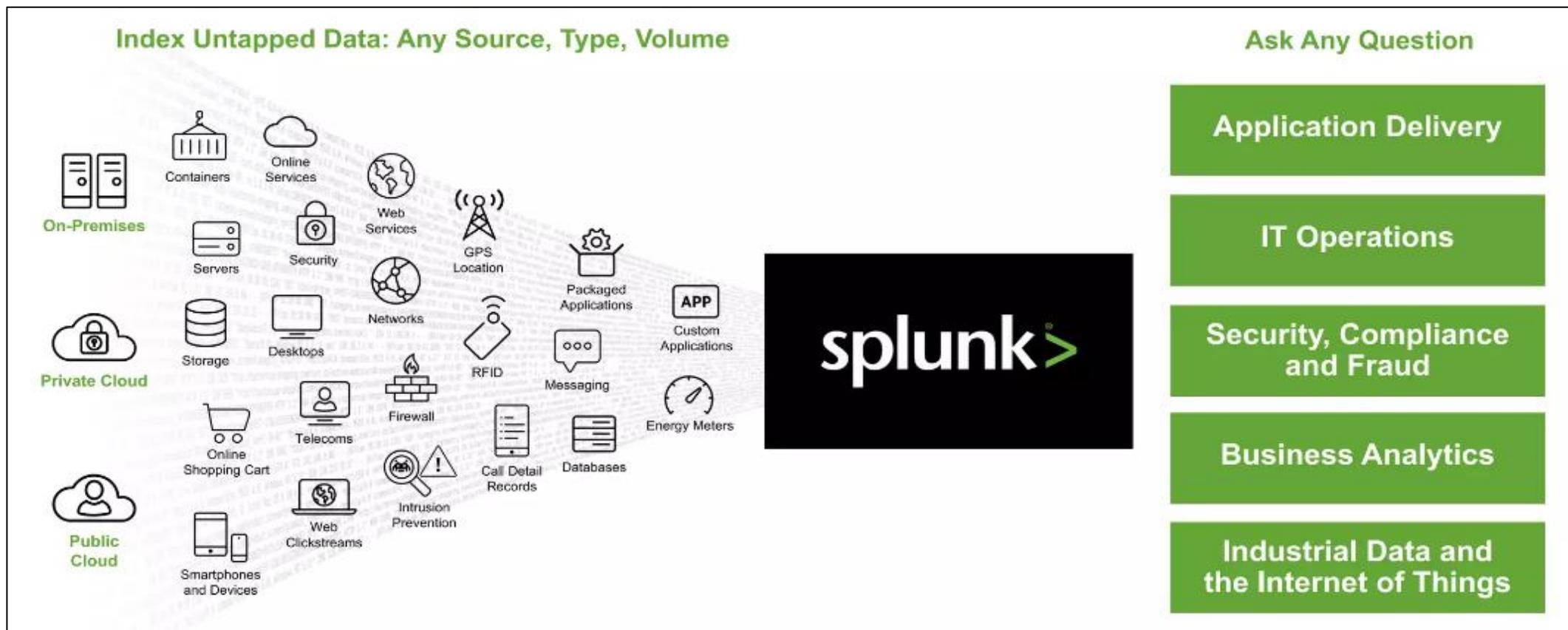
**field:** Specifies the field in which you want to find rare values.

It is used for Anomaly detection, quality control, security analysis, data validation and reporting.

# Transforming commands: Rare

the rarest 10 values in the error_code field,	...   rare limit=10 field,error_code
To display the 5 rarest error_code values for each server in your data.	...   rare limit=5 field,error_code by=server
the rarest 10 IP addresses and calculates their average response times.	...   rare limit=10 field=ip_address   stats avg(response_time) as AvgResponseTime by ip_address
to find the rarest HTTP response codes over time	...   rare limit=5 field=response_code   timechart count by response_code
groups events by user and then identifies the rarest 5 users based on the count of their transactions.	...   transaction user   rare limit=5 field=user
the rarest 10 error codes that occurred after January 1, 2023.	...   where time >= "2023-01-01"   rare limit=10 field,error_code

# Converging Data Sources



# Iplocation Geographic Data

---

1	<b>Assign Lat/Lon to IP addresses:</b>
	Sourcetype=access_combined   iplocation clientip
2	<b>Visualize statistics geographically:</b>
	Sourcetype=access_combined   iplocation clientip   geostats sum(price) by product
3	<b>Use custom choropleths:</b>
	geom <featureCollection> <featureid>
4	<b>Track Object movements</b>
	index=drive-index latitude=* longitude=*   table _time latitude longitude vehicleid

# Search Modes

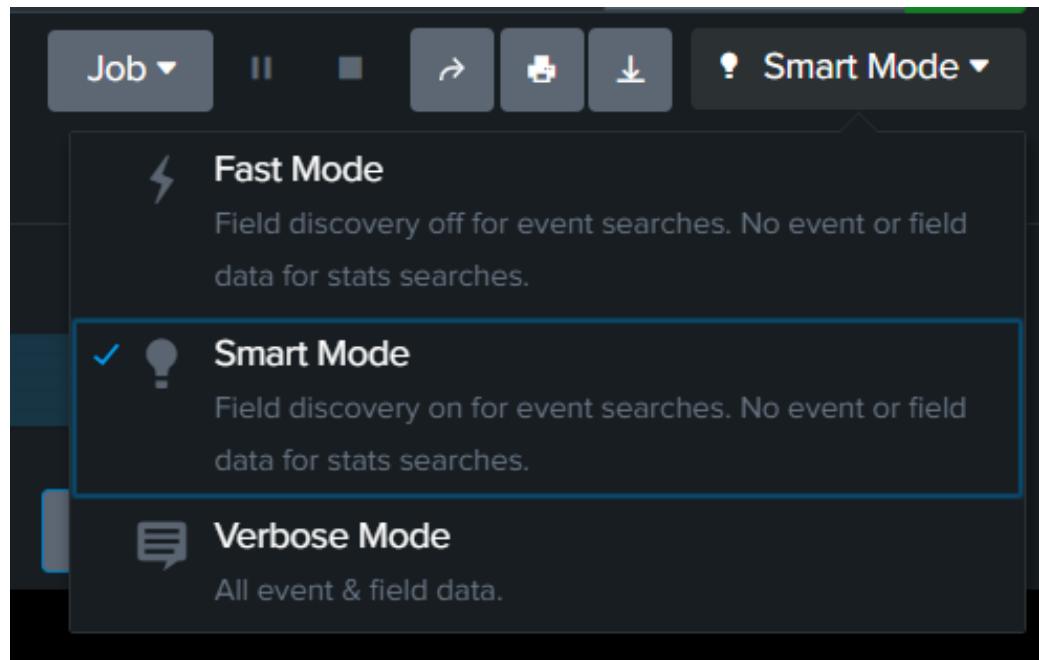
---

You can use the Search Mode selector to provide a search experience that fits your needs.

Search Mode	Description
<b>Fast Mode</b>	<ul style="list-style-type: none"><li>The Fast mode prioritizes the performance of the search and does not return nonessential field or event data</li><li><b>Disables field discovery</b></li><li><b>Only depicts search results as report result tables or visualizations when you run a reporting search</b></li></ul>
<b>Verbose Mode</b>	<ul style="list-style-type: none"><li><b>Discovers all of the fields it can</b></li><li><b>Returns an event list view of results and generates the search timeline</b></li></ul>
<b>Smart Mode</b>	<ul style="list-style-type: none"><li>All reports run in Smart mode, the <b>default search mode</b>, after they are first created.</li><li>When you run a Smart mode search that <b>does not include transforming commands</b>, the search behaves as if it were in <b>Verbose mode</b>.</li><li>When you run a Smart mode search that <b>includes transforming commands</b>, the search behaves as if it were in <b>Fast mode</b>.</li></ul>

# Search Modes

---



Transforming commands include **chart**, **timechart**, **stats**, **top**, **rare**, **contingency**, and **highlight**.

# Splunk Search Assistant

When you type a few letters or a term into the Search bar, the search assistant shows you terms and searches that match what you typed. It gets these values from the suggestions or history.

New Search

failed | stats count by

✓ 814 events (1/1/23 12:00:00)

Events Patterns St 20 Per Page ▾ Form

src ↴ 1.189.205.173 1.30.211.144 103.230.120.26

stats count by action, host  
stats count by host  
stats count by host, port  
stats count by src  
stats count by src\_ip,dest\_ip,dest\_port  
failed | stats count by src

stats

Provides statistics, grouped optionally by field.

Example:

sourcetype=access\_combined | top limit=100 referer\_domain | stats sum(count)

Learn More ↗

# Splunk Search Assistant modes

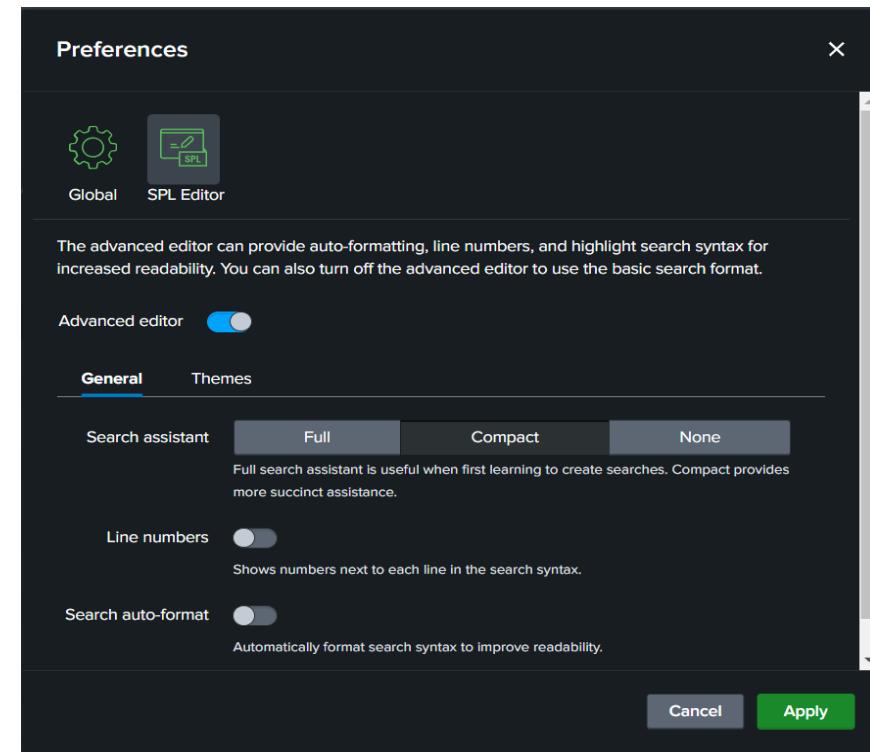
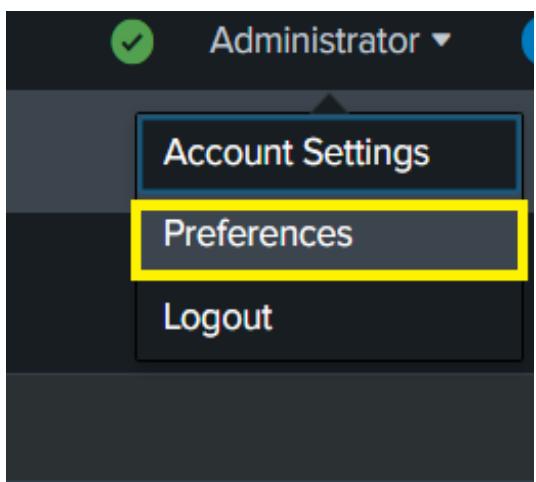
---

Splunk Search Assistant has three modes:

Modes	Description
<b>Full</b>	<ul style="list-style-type: none"><li>The Full mode displays a list of matching terms and searches when you type, along with a count of how many times a term appears in your indexed data.</li></ul>
<b>Compact</b>	<ul style="list-style-type: none"><li>This is the default mode</li><li>The Compact mode displays a list of matching terms and searches when you type. When you type a pipe (   ) character to indicate that you want to use a command, a list of the SPL commands appears.</li></ul>
<b>None</b>	<ul style="list-style-type: none"><li>It will not give any suggestion.</li></ul>

# Splunk Search Assistant modes

## Changing Modes:



# Splunk Search Assistant modes : Full Mode

## Full Mode

New Search

failed | stats count

Matching Searches

failed | stats count by src

Command History

... | stats count by action, host  
... | stats count by host  
... | stats count by src  
... | stats count by src\_ip,dest\_ip,dest\_port  
... | stats count by host, port

stats

Help [Less](#)

Provides statistics, grouped optionally by field.

Details

Calculate aggregate statistics over the dataset, optionally grouped by a list of fields. Aggregate statistics include:

- \* count, distinct count
- \* mean, median, mode
- \* min, max, range, percentiles
- \* standard deviation, variance
- \* sum
- \* earliest and latest occurrence
- \* first and last (according to input order into stats command) occurrence

Similar to `sql` aggregation. If called without a by-clause, one row is produced, which represents the aggregation over the entire incoming result set. If called with a by-clause, one row is produced for each distinct value of the by-clause. The 'partitions' option, if specified, allows stats to partition the input data based on the split-by fields for multithreaded reduce. The 'allnum' option, if true (default = false), computes numerical statistics on each field if and only if all of the values of that field are numerical. The 'delim' option is used to specify how the values in the 'list' or 'values' aggregation are delimited. (default is a single space) When called with the name `prestats`, it will produce intermediate results (internal).

Syntax

[More >](#)

# Splunk Search Assistant modes : Compact Mode

---

## Compact Mode

New Search

```
failed | stats count by
```

✓ 814 events (1/1/23 12:00)

Events Patterns

20 Per Page ▾

src ↴

1.189.205.173

1.30.211.144

103.230.120.26

stats count by action, host  
stats count by host  
stats count by host, port  
stats count by src  
stats count by src\_ip,dest\_ip,dest\_port  
failed | stats count by src

**stats** Command History

Provides statistics, grouped optionally by field.

Example:

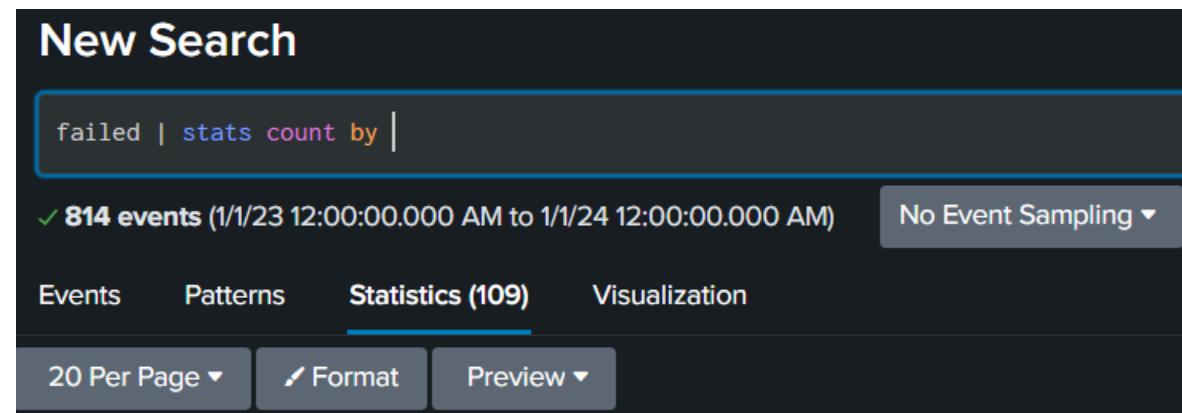
sourcetype=access\_combined | top limit=100 referer\_domain | stats sum(count)

Learn More ↗

# Splunk Search Assistant modes : None Mode

---

## None Mode



# Splunk Reports

---

Reports are created when you save a search or a pivot for later reuse.

We can do the following with reports :

- Manually Create and edit reports
- Accelerate slow completing reports
- Set up scheduled reports
- Configure the priority of scheduled reports
- Generate PDFs of reports, dashboard, searched and pivots

# Summary Indexing

---

Summary Indexing enable you to efficiently search on large volumes of data. When you create a summary index you design a scheduled search that runs in the background, extracting a precise set of statistical information from a large and varied dataset.

The results of each run of the search are stored in a summary index that you designate. It is much faster than similar searches run against the source dataset.

The summary index is "faster" because it is smaller than the original dataset and contains only data that is relevant to the search that you run against it.

The summary index is also statistically accurate, in part because the scheduled search that updates the summary runs on an interval that is shorter than the average time range of the searches that you run against the summary index

# Type of Summary Indexes

---

You can create two types of summary indexes:

- **summary events indexes:** Summary events indexes store the statistical event data as [events](#)
- **summary metrics indexes:** summary metrics indexes convert that statistical event data into [metric data points](#) as part of their summarization process

Metrics indexes store metric data points in a way that makes searches against them notably fast, and which reduces the space they take up on disk, compared to events indexes. You may find that a summary metrics index provides faster search performance than a summary events index, even when both indexes are summarizing data from the same source dataset.

All summarized data has a special default source type. Events summarized in a summary events index have a source type of stash. Metric data points summarized in a summary metrics index have a source type of mcollect\_stash

# Summary Indexing use cases

---

## 1. Run reports over long time ranges for large datasets more efficiently

Your instance of the Splunk platform indexes tens of millions of events per day. You want to set up a dashboard with a panel that displays the number of page views and visitors each of your Web sites had over the past 30 days, broken out by site.

you set up a saved search that collects website page view and visitor information into a designated summary index on a weekly, daily, or even hourly basis. You'll then run your month-end report on this smaller summary index, and the report should complete far faster than it would otherwise because it is searching on a smaller and better-focused dataset.

## 2. Building rolling reports

If you want total number of downloads then schedule a saved search to return the total number of downloads over a specified slice of time as a summary index. You can then run a report any time you want on the data in the summary index to obtain the latest count of the total number of downloads.

# Create a summary index

Summary index can be created with charts, rare, stats, timechart, and top using the “si” prefix

The screenshot shows the Splunk Search & Reporting interface. The top navigation bar includes links for Search, Analytics, Datasets, Reports, Alerts, and Dashboards. On the right, there's a green button labeled "Search & Reporting". Below the navigation is a search bar containing the command: `source="personal-details" | dbxlookup lookup="employment-info" | sistats count by country, name, employee_salary, age`. To the right of the search bar are buttons for "Save As", "Create Table View", and "Close". The main area is titled "New Search" and displays the results of the search. It shows 83,000 events from before 9/1/23 6:51:22.000 PM. The search results are presented in a table with the following columns: source, \_time, \_raw, host, sourcetype, index, linecount, splunk\_server, punct, id, name, user\_id, email, age, employee\_salary, employee\_profile, country, and psrsvd. The table contains six rows of data:

source	_time	_raw	host	sourcetype	index	linecount	splunk_server	punct	id	name	user_id	email	age	employee_salary	employee_profile	country	psrsvd
										Michelle Boyd			80	121427		Afghanistan	
										Sue McKinney			37	79591		Afghanistan	
										Troy Wright			39	32456		Albania	
										Andrew Hernandez II			48	147918		Algeria	
										Christopher Delgado			24	65149		Algeria	
										Dr. Rachel Summers DDS			45	131479		Algeria	

# Create a summary index

Save the report and schedule it.

Save As Report

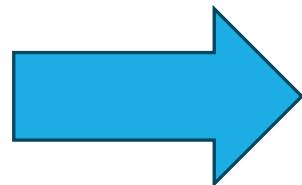
Title: salary\_summary\_report

Description: optional

Content: Statistics Table

Time Range Picker: Yes

Cancel Save



Edit Schedule

Scheduling this report results in removal of the time picker from the report display.

Report: salary\_summary\_report

Schedule Report:  Learn More

Schedule: Run every day ▾

At: 0:00

Time Range: All time ▾

Schedule Priority: Default ▾

Schedule Window: No window ▾

Trigger Actions: + Add Actions ▾

Cancel Save

# Create a summary index

Go to searches, reports and alerts → edit summary index

**Edit Summary Index**

Report `salary_summary_report`

Enable summary indexing  Create summaries of event or metric data. [Learn More](#)

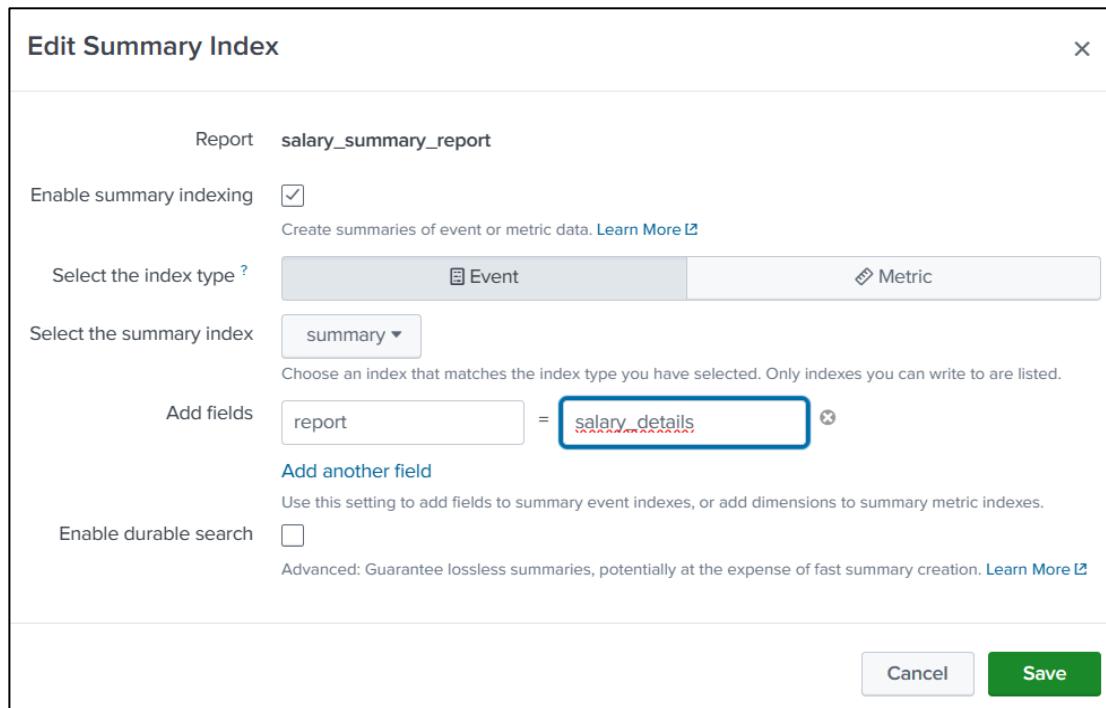
Select the index type ?  Event  Metric

Select the summary index `summary` ▾ Choose an index that matches the index type you have selected. Only indexes you can write to are listed.

Add fields `report` = `salary_details` [Add another field](#) Use this setting to add fields to summary event indexes, or add dimensions to summary metric indexes.

Enable durable search  Advanced: Guarantee lossless summaries, potentially at the expense of fast summary creation. [Learn More](#)

[Cancel](#) [Save](#)



# Splunk Reports: Create a report

1. Select **Settings > Searches, reports, and alerts**

2. Click **New Report**.

**Create Report**

Title	Failed SSH events
Description	optional
Search	<pre>source="auth1.log" action=failure   iplocation src   table src, Country</pre>
Earliest time	optional Time specifiers: y, mon, d, h, m, s <a href="#">Learn More</a>
Latest time	optional Time specifiers: y, mon, d, h, m, s <a href="#">Learn More</a>
App	Search & Reporting (search) ▾
Time Range Picker	Yes <input type="radio"/> No <input type="radio"/>
<input type="button" value="Cancel"/> <input type="button" value="Save"/>	

**Searches, Reports, and Alerts**

Searches, reports, and alerts are saved searches created from pivot or the search page. [Learn more](#)

Name	Actions	Type	Next Scheduled Time
Failed SSH events	Edit Run	Report	none

The "Edit" button for the "Failed SSH events" search has a dropdown menu open, showing the following options:

- Edit Search (selected)
- Edit Permissions
- Edit Schedule
- Edit Acceleration
- Edit Summary
- Indexing
- Disable
- Advanced Edit
- Clone
- Embed
- Move
- Delete

# Report

# Splunk Reports: Saving a search

---

You can create a report out of your search and edit some of the settings like Permissions, Schedule, Acceleration, Embed

The screenshot shows the Splunk "New Search" interface. At the top, there is a search bar containing the command: `source="auth1.log" action=failure | iplocation src | table src, Country`. Below the search bar, it displays **1,268 events** (before 8/23/23 6:09:18.000 PM) and "No Event Sampling". The "Statistics (1,268)" tab is selected. At the bottom, there are buttons for "Events", "Patterns", "Statistics (1,268)", "Visualization", "20 Per Page", "Format", and "Preview". On the right side, there is a "Save As" dropdown menu with options: Report (selected), Alert, Job, Existing Dashboard, New Dashboard, and Event Type. A page number "1" is also visible.

# Splunk Reports : List of reports

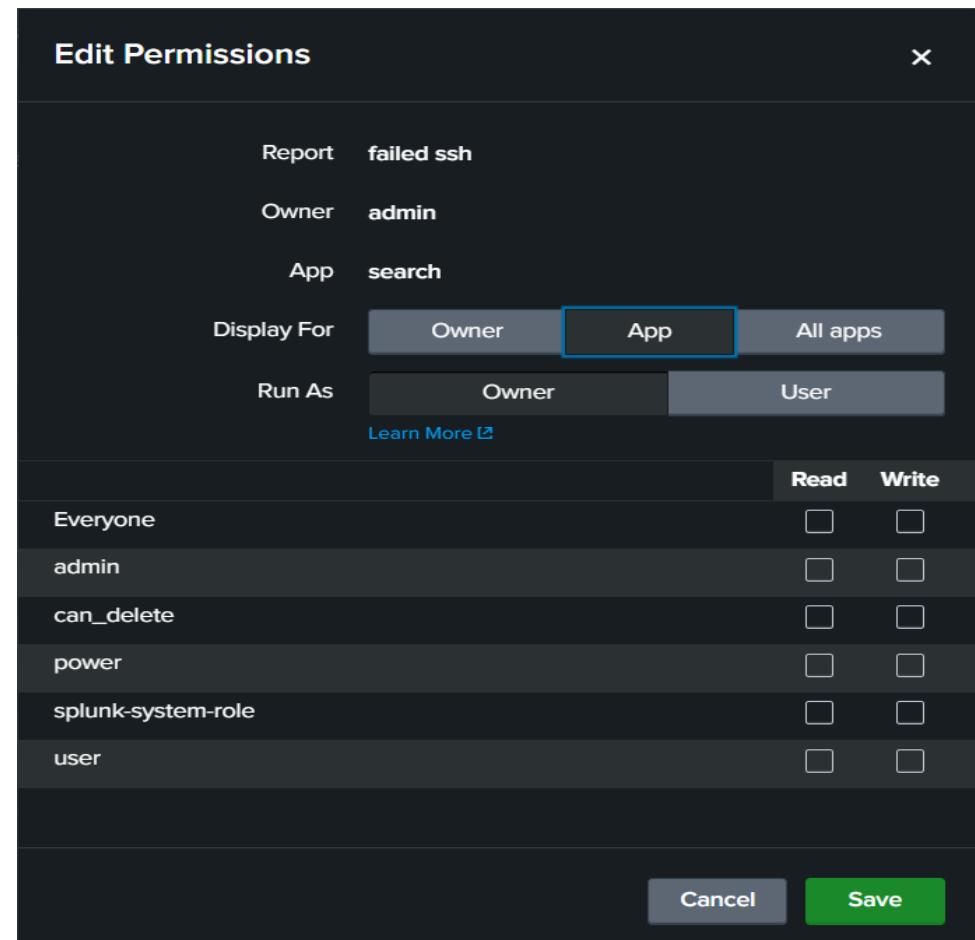
The screenshot shows the Splunk interface with the 'Reports' tab selected. The page displays a list of 9 reports, each with a title, actions (Open in Search, Edit), next scheduled time, owner, app, and sharing status.

i	Title	Actions	Next Scheduled Time	Owner	App	Sharing
>	Bucket Merge Retrieve Conf Settings	Open in Search Edit	None	nobody	search	App
>	Errors in the last 24 hours	Open in Search Edit	None	nobody	search	App
>	Errors in the last hour	Open in Search Edit	None	nobody	search	App
>	Failed SSH events	Open in Search Edit	None	admin	search	Private
>	License Usage Data Cube	Open in Search Edit	None	nobody	search	App
>	Messages by minute last 3 hours	Open in Search Edit	None	nobody	search	App
>	Orphaned scheduled searches	Open in Search Edit	None	nobody	search	App
>	Splunk errors last 24 hours	Open in Search Edit	None	nobody	search	App
>	failed ssh	Open in Search Edit	None	admin	search	Private

# Splunk Reports : Permission

We can set report permission using below options

- Select either **App** or **All apps**.  
All reports are created in the context of a specific app. Select **App** to share the report with other users of the app that the report belongs to.
- Select **All apps** to share the report with all users of your Splunk platform implementation.
- Determine whether the report runs as **Owner** or **User**.
- Set the **Read** and **Write** permissions by role.



# Splunk Reports : Permission

---

Why to set the permission??

All reports run as **Owner** by default.

- **Control access to report results**

The **Owner** and **User** settings let you control access to the data returned by your reports

- **Keep concurrent search job limits for report owners from being exceeded**

The **Owner** and **User** controls determine whether a run of the report counts against the concurrent search job limit of the report "owner" or the report "user."

- **Permissions for Pivot-based reports**

Your Splunk deployment can have two apps installed: Search and Security. When you are in the context of the Security app, you use its External Threats data model to create a Pivot-based report titled "Top Firewall Attacks by IP."

This file can be given the permission for App (should have same Security App) or All Apps (can get the result from the search app too).

# Splunk Reports: Accelerate report

---

- If your report has a large number of events and is slow to complete when you run it, you may be able to accelerate it so it completes faster when you run it in the future.
- When you accelerate a report, Splunk software runs a background process that builds a data summary based on the results returned by the report.
- When you next run the search, it runs against this summary rather than the full index.
- The base search must use a transforming command (such as chart, timechart, stats, and top) in order to qualify for Acceleration

# Splunk Reports: Accelerate report

## How Summary Range works

**Summary Range** sets the approximate range of time that a report's data summary will cover. When you run the report in the future only the portion of it that falls within that range will benefit from acceleration.

For example, if you choose a **Summary Range** of *7 Days*, you're saying that going forward you want a summary that always covers at least the last seven days. As time passes, Splunk software will delete data from this summary that is older than seven days while it continues to summarize incoming new data.

**Edit Acceleration** X

Report failure count by src

Accelerate Report

Acceleration might increase storage and processing costs.  
Acceleration can return invalid results if you change definitions  
of knowledge objects used in the search string after you acceler-  
ate the report. [Learn More](#) ↗

Summary Range ? 7 Days ▾

Cancel Save

# Splunk Reports: Schedule Reports

A scheduled report is a report that runs on a scheduled interval, and which can trigger an action each time it runs. You can define up to four actions for a scheduled report:

- Send a report summary by email
- Write the report results to a CSV lookup file
- Set up a webhook that sends a message to an external web resource, such as a chatroom
- Log and index searchable events
- Run a script
- Output results to telemetry endpoint
- Send to Splunk mobile

**Edit Schedule**

**⚠ Scheduling this report results in removal of the time picker from the report display.**

Report **report3**

Schedule Report  [Learn More ↗](#)

Schedule **Run every week ▾**

On **Friday ▾** at **0:00 ▾**

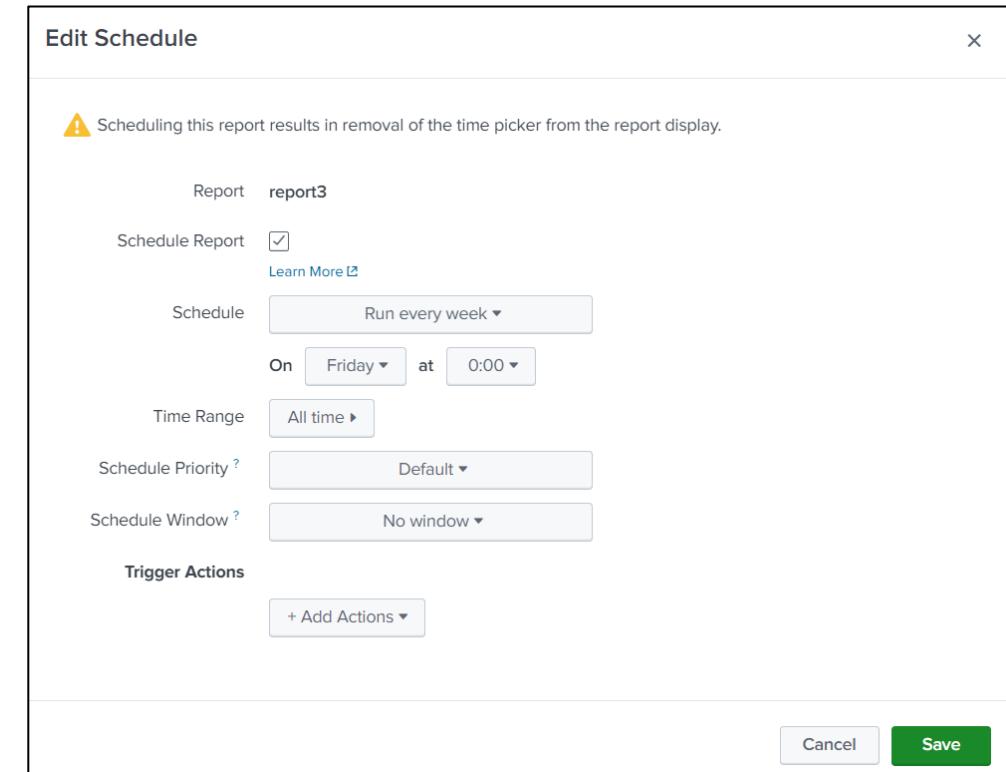
Time Range **All time ▾**

Schedule Priority **Default ▾**

Schedule Window **No window ▾**

Trigger Actions **+ Add Actions ▾**

**Cancel** **Save**



# Splunk Reports: Schedule Reports

You have to configure the email settings in order to get the mails from Splunk.

Go to settings >> server settings >> email settings

Email settings  
[Server settings](#) » [Email settings](#)

**Mail Server Settings**

Mail host  Set the host that sends mail for this Splunk instance.

Email security  none  Enable SSL  Enable TLS  
Check with SMTP server admin. When SSL is enabled, mail host should include the port. IE: smtp.splunk.com:465

Username

Username to use when authenticating with the SMTP server. Leave empty for no authentication.

Password

Password to use when authenticating with the SMTP server.

Confirm password

**Email Domains**

Allowed Domains \*

Provide a comma-separated list of allowed email domains. Leave empty for no restriction.

**Email Format**

# Splunk Reports: Embed scheduled reports

Report embedding lets you bring the results of your reports to large numbers of report stakeholders. With report embedding, you can embed scheduled reports in external (non-Splunk) websites, dashboards, and portals.

Embedded reports can display results in the form of event views, tables, charts, maps, single values, or any other visualization type.

You can't embed a report until it is scheduled to run on a regular interval.

An embedded report always displays the results from its last scheduled run.

**Embed** X

⚠ Embedded Report will not have data until the scheduled search runs.

Copy and paste this code into your HTML-based web page.

```
<iframe height="636" width="480" frameborder="0" src="http://54.67.59.152:8000/en-US/embed?s=%2FservicesNS%2Fadmin%2Fsearch%2Fsaved%2Fsearches%2Freport3&oid=ArhfM%5EV%5EbQkl%5E5iaKUsbdRMwLdIcF_c6iiXdw38a%5EUSHZ1TnVTtR0zTceRLnZATbA9rpK4"/>
```

Disable embedding if you no longer want to share this report outside of Splunk.

Disable Embedding Done

# **Splunk App & Addons**

# Overview of Splunk Add-ons

---

Splunk add-ons support and extend the functionality of the Splunk platform and the apps that run on it, usually by providing inputs for a specific technology or vendor.

It provides required field extractions, lookups, saved searches and others.

Add-ons can also be used to extract data from a specific destination (like AWS).

These Add-ons will have a specific script to fetch the data from the destination and parse it in the Splunk environment.

# Splunk Add-ons: Splunk Add-on for Unix and Linux

We can change the data to be fetched by this plugin

Now you will see the below Sourcetypes

Data Summary			
Hosts (1)	Sources (219)	Sourcetypes (73)	X
<input type="text"/> filter <input type="button"/>			
Sourcetype	Count	Last Update	
01-vendor-ubuntu-too_small	2	8/23/23 8:04:29.000 PM	
01autoremove-too_small	1	8/23/23 8:04:28.000 PM	
10periodic-too_small	3	8/23/23 8:04:28.000 PM	
15update-stamp-too_small	1	8/23/23 8:04:29.000 PM	
20archive-too_small	3	8/23/23 8:04:29.000 PM	
20auto-upgrades-too_small	2	8/23/23 8:04:29.000 PM	
20packagekit-too_small	11	8/23/23 8:04:28.000 PM	
50command-not-found-too_small	1	8/23/23 8:04:28.000 PM	
50unattended-upgrades	1	8/23/23 8:04:28.000 PM	
70debconf-too_small	3	8/23/23 8:04:28.000 PM	

**Splunk Add-on for Unix and Linux: Setup**

The Splunk Add-on for Unix and Linux provides pre-built data inputs to facilitate Linux and Unix system monitoring using Splunk. Check out the [Splunk for Unix Technical Add-on](#) page on Splunkbase for support information, the latest updates, and more.

**File and Directory Inputs:**

Name	Enable (All)	Disable (All)
/etc	<input checked="" type="radio"/>	<input type="radio"/>
/home/*/.bash_history	<input checked="" type="radio"/>	<input type="radio"/>
/Library/Logs	<input checked="" type="radio"/>	<input type="radio"/>
/root/.bash_history	<input checked="" type="radio"/>	<input type="radio"/>
/var/adm	<input checked="" type="radio"/>	<input type="radio"/>
/var/log	<input checked="" type="radio"/>	<input type="radio"/>

**Scripted Metric Inputs:**

Name	Enable (All)	Disable (All)	Interval (sec)	Index
cpu_metric.sh	<input checked="" type="radio"/>	<input type="radio"/>	30	Select... <input type="button"/>
df_metric.sh	<input checked="" type="radio"/>	<input type="radio"/>	300	Select... <input type="button"/>
interfaces_metric.sh	<input checked="" type="radio"/>	<input type="radio"/>	60	Select... <input type="button"/>
iostat_metric.sh	<input checked="" type="radio"/>	<input type="radio"/>	60	Select... <input type="button"/>
ps_metric.sh	<input checked="" type="radio"/>	<input type="radio"/>	30	Select... <input type="button"/>
vmstat_metric.sh	<input checked="" type="radio"/>	<input type="radio"/>	60	Select... <input type="button"/>

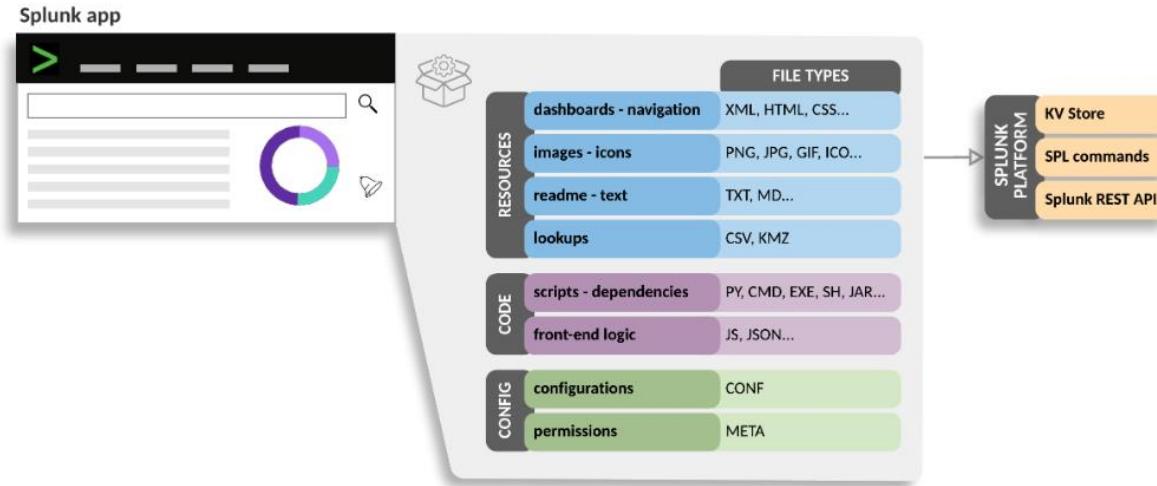
**Scripted Event Inputs:**

Search produced no results.

# Splunk App

Apps delivers user experience that makes data immediately useful typically with pre-built dashboards that makes data easy to analyze

A Splunk app is a packaged collection of knowledge objects and extensions, most of which are represented as files in the Splunk platform installation in your app's directory, **\$SPLUNK\_HOME/etc/apps/appname/**. The following diagram shows the anatomy of an app by categorizing objects, like resources and code, by file types:

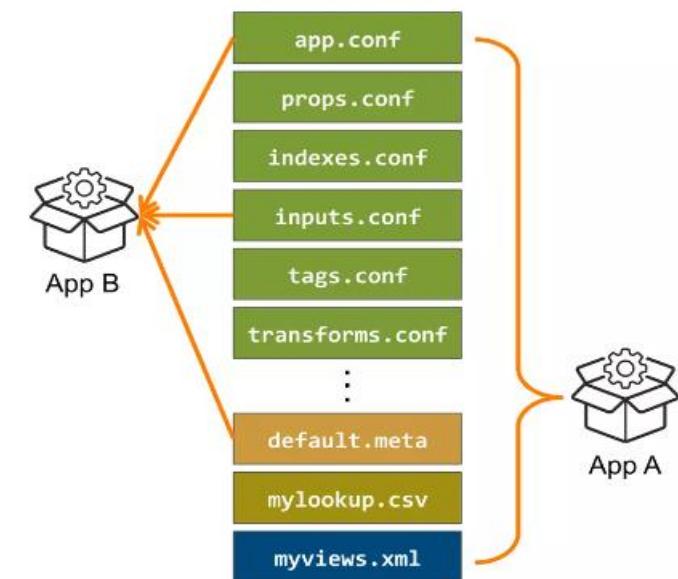


Anatomy of a Splunk app

# What is an APP?

---

- An app is an independent collection
  - configuration files
    - Defining inputs, indexes, sourcetypes, field extractions, transformations
    - Providing eventtypes, tags, reports, dashboards and other knowledge objects
  - Scripts, web assets etc
- Most apps are focused on:
  - A specific type of data from a vendor, operating system or industry
  - A specific business need
- Apps may be installed on any Splunk Instance
- Splunk includes a number of default apps



# View all Installed Apps

---

- From the Search app, Select Apps → Manage Apps
- Apps can be Visible or hidden
  - Several apps are installed by default that are hidden or disabled
    - Internal apps used by Splunk should not be modified
    - Legacy apps
    - Sample apps
- Apps are installed under **SPLUNK\_HOME/etc/apps**

# Installing an App

---

- You can install an App from [splunkbase.splunk.com](http://splunkbase.splunk.com)
- Click **Apps** → **find More Apps** → install the app
- You can also install an app from a file
  - Download the file for the app from [splunkbase.splunk.com](http://splunkbase.splunk.com) (in .tar, .gz, .tgz, .zip or .spl format)
  - install it using the CLI

```
splunk install app path-to-appfile
```

- or extract the app in the proper location which is **SPLUNK\_HOME/etc/apps**
- Some apps may require a restart

# Apps and Add-ons Support

---

There are three types of support criteria that you will generally see in Splunkbase

- Splunk Supported
- Developer Supported
- Community Supported

**Note:** Splunk Add-ons and Apps can consume heavy resources which might impact our server, so check the system requirements before installing them.

# Splunk Add-ons for AWS

The Splunk Add-on for Amazon Web Services allows us to collect AWS related data and logs.

<u>Use case</u>	<u>Add-on inputs</u>
Use the Splunk Add-on for AWS to calculate the <b>cost of your Amazon Web Service usage</b> over different lengths of time.	<ul style="list-style-type: none"><li>Billing (Cost and Usage report)</li><li>Billing (Legacy)</li></ul>
Use the Splunk Add-on for AWS to push <b>CloudTrail log data</b> to the Splunk platform. CloudTrail allows you to audit your AWS account.	<ul style="list-style-type: none"><li>CloudTrail</li><li>Kinesis data</li><li>S3 Access Logs</li></ul>
Use the Splunk Add-on for AWS to push <b>IT and performance data</b> on your Amazon Web Service into the Splunk platform.	<ul style="list-style-type: none"><li>Amazon CloudWatch data</li><li>CloudFront Access Logs</li><li>ELB Access Logs</li><li>Config and Config Rules data</li><li>Description data</li><li>Kinesis data</li><li>S3 Access Logs</li><li>SQS-based Access Logs</li><li>VPC flow log data</li></ul>
Use the Splunk Add-on for AWS to push <b>security data</b> on your Amazon Web Service into the Splunk platform.	<ul style="list-style-type: none"><li>Inspector data</li><li>Inspector (v2) data</li><li>Config and Config Rules data</li><li>Description data</li><li>Kinesis data</li><li>S3 Access Logs</li><li>SQS-based Access Logs</li><li>VPC flow log data</li></ul>

# Splunk Add-ons for AWS

---

This add-on supports for both pull based and push based data collection.

<u>Push Data</u>	<u>Pull Data</u>
For high volume, streaming data.	For low volume, rarely changing data.
If high availability and scale are required for your deployment.	For normal availability and scale.
Sends data directly to indexers so you do not need to manage forwarders.	Unless your deployment is in Splunk Cloud, you must manage the forwarders.

# Splunk Add-ons for AWS

---

Important points to remember before installing a Splunk Add-on

- Add-on must support your Splunk version
- Hardware requirements for the Add-on
- Support Type
- For this AWS add-on we need the authentication keys or IAM role if the Splunk server is running on AWS account only.

# Splunk Add-ons for AWS

Steps:

- **Install Splunk add-on for Amazon Web Services**
- Open the app and configure the Account details or IAM role
- Create a data input for the data you want to get.

Add Account

Name	splunk-user
Key ID	XXXXXXXXXXXXXX
Secret Key	.....
Region Category	Global

Cancel Add

Add IAM Role

Name	splunk-role
Role ARN	arn:aws:iam::XXXXXXXXX role/splunk-role

Cancel Add

# Splunk Add-ons for AWS

Create a data input for aws metadata.

Select the user or IAM role.

Select the region and add the report.

AWS Input Configuration [Learn more](#)

Name	aws-metadata
AWS Account	splunk-user
Assume Role	optional
AWS Regions	US West (N. California) X

APIs/Interval (in seconds)	API	Interval (in seconds)
	<input checked="" type="checkbox"/> ec2_volumes	3600
	<input checked="" type="checkbox"/> ec2_instances	3600
	<input checked="" type="checkbox"/> ec2_reserved_instances	3600
	<input checked="" type="checkbox"/> ebs_snapshots	3600
	<input checked="" type="checkbox"/> classic_load_balancers	3600
	<input checked="" type="checkbox"/> application_load_balancers	3600
	<input checked="" type="checkbox"/> vpcs	3600
	<input checked="" type="checkbox"/> vnc_network_acls	3600

# Splunk Add-ons for AWS

Explore the AWS data on Splunk now.

New Search

sourcetype="aws:metadata"

✓ 1,477 events (8/23/23 5:00:00.000 AM to 8/24/23 5:27:06.000 AM) No Event Sampling ▾

Events (1,477) Patterns Statistics Format Timeline ▾ — Zoom Out

source X

11 Values, 100% of events Selected Yes No

Reports

Top values Top values by time Rare values

Events with this field

< Hide Fields All Fields

SELECTED FIELDS

a host 1  
a source 11  
a sourcetype 1

INTERESTING FIELDS

# AccountId 1  
a Arn 100+  
# AttachmentCount 11  
a CreateDate 100+  
a DefaultVersionId 25  
a index 1  
a IsAttachable 1  
# linecount 1

Top 10 Values

	Count	%
685421549691:us-west-1:iam_list_policy	1,320	89.37%
685421549691:us-west-1:iam_users	100	6.77%
685421549691:us-west-1:s3_buckets	25	1.693%
685421549691:us-west-1:ec2_security_groups	14	0.948%
685421549691:us-west-1:ec2_volumes	4	0.271%
685421549691:us-west-1:vpc_subnets	4	0.271%
685421549691:us-west-1:ec2_key_pairs	3	0.203%
685421549691:us-west-1:ec2_instances	2	0.135%
685421549691:us-west-1:vpc_network_acls	2	0.135%
685421549691:us-west-1:vpcs	2	0.135%

Show as raw text

	Count	%
685421549691:us-west-1:iam_list_policy	1,320	89.37%
685421549691:us-west-1:iam_users	100	6.77%
685421549691:us-west-1:s3_buckets	25	1.693%
685421549691:us-west-1:ec2_security_groups	14	0.948%
685421549691:us-west-1:ec2_volumes	4	0.271%
685421549691:us-west-1:vpc_subnets	4	0.271%
685421549691:us-west-1:ec2_key_pairs	3	0.203%
685421549691:us-west-1:ec2_instances	2	0.135%
685421549691:us-west-1:vpc_network_acls	2	0.135%
685421549691:us-west-1:vpcs	2	0.135%

# **Splunk DB connect**

# What is Splunk DB connect

---

- With Splunk DB Connect 3, you can combine your structured data from databases with your unstructured machine data, and then use Splunk Enterprise to provide insights into all of that combined data.
- When you use Splunk DB Connect, you create additional data inputs for Splunk Enterprise, giving Splunk Enterprise more sources of data. Splunk DB Connect connects your relational database data to Splunk Enterprise and makes that data consumable by Splunk Enterprise.
- In addition, Splunk DB Connect can do the reverse, writing Splunk Enterprise data back to your relational database.
- DB Connect also performs database lookups, which let you reference fields in an external database that match fields in your event data. Using these matches, you can add more meaningful information and searchable fields to enrich your event data.

# Who DB connect is for?

---

- Quickly get data from a database into Splunk Enterprise.
- Run on-the-fly lookups from data warehouses or state tables within Splunk Enterprise.
- Index structured data stored in databases in streams or batches using Splunk Enterprise.
- Write Splunk Enterprise data into databases in streams or batches.
- Preview data and validate settings such as locale and time zone, rising column and metadata choice, and so on before indexing begins, to prevent accidental duplication or other problems in the future.
- Scale, distribute, and monitor database read/write jobs to prevent overload and receive notice of failures.
- Know what databases are accessible to which Splunk Enterprise users to prevent unauthorized access.

# Setup Splunk DB Connect

---

To set up Splunk DB Connect, download [Splunk DB Connect](#) from Splunkbase and it is an add-on that bridges Splunk Enterprise with relational databases through Java Database Connectivity (JDBC).

All DB Connect instances require Java Runtime Environment (JRE) version 11 or higher in order to enable JDBC.

You must also install a Java Database Connectivity (JDBC) driver so that Splunk Enterprise can communicate with your databases.

# Setup Splunk DB Connect

---

<b>Identities</b>	An identity, which consists of a username and password, defines the database user through which Splunk Enterprise connects to your database. A single identity can be used by many connections, so that service accounts can be easily shared across multiple systems
<b>Connections</b>	create a connection, which contains the information necessary to connect to a specific database. It consists of the address of your database (the host name), the database's type, and the name of the database.
<b>Database Inputs</b>	A database input enables you to retrieve and index data from a database using Splunk Enterprise.
<b>Search</b>	After you set up identities, connections, and database inputs, and Splunk Enterprise has indexed your data, you are ready to search. Indexed data obtained through Splunk DB Connect from relational databases is searchable just like the rest of your Splunk Enterprise data. DB Connect provides the <code>dbxquery</code> command for querying remote databases and generating events in Splunk Enterprise from the database query result set.

# Setup Splunk DB Connect

<b>Database outputs</b>	Splunk DB Connect also enables you to write Splunk Enterprise data back to your relational database using <b>database outputs</b> . You can do this interactively from a search head or by setting up an automatic output from a heavy forwarder.
<b>Database lookups</b>	Splunk DB Connect allows you to interact with your external database. Database lookups give you real-time contextual information from a database during ad hoc search in the Splunk platform.  Use the <i>dbxlookup</i> command to perform lookups by using remote database tables as lookup tables.
<b>Health Monitoring</b>	Splunk DB Connect includes a health dashboard that allows you to monitor numerous aspects of your database connections and transactions with Splunk Enterprise.

# Setup Splunk DB Connect

## Steps:

1. Install Splunk DB connect app
2. Install JRE in your Splunk machine and set the env JAVA\_HOME
3. Configure the Splunk DB connect app and save it.

The screenshot shows the Splunk DB Connect Settings page. The 'General' tab is selected under the 'Databases' section. The 'Settings' tab is also visible. The 'JRE Installation Path (JAVA\_HOME)' field contains the value '/usr/lib/jvm/java-11-openjdk-amd64/'. A large green arrow points upwards from the bottom left towards this field, with the text 'Set the JAVA\_HOME as configured in Splunk Machine.' overlaid. Other fields include 'Task Server JVM Options' (-Ddw.server.applicationConnectors[0].port=9998), 'Query Server JVM Options' (-Dport=9999), 'Task Server Port' (9998), and 'Query Server Port' (9999). Buttons for 'Reset' and 'Save' are at the top right.

# Setup Splunk DB Connect

4. Install the **driver** to connect with a database (for example **Splunk\_JDBC\_mysql**)
5. Create the **identity** (Username and password)
6. Create a **connection** to connect with the database using the above identity.
7. Go to **Data Lab** and create a **new input** to search from the database. Set the properties to index them using a specific **sourcetype**.
8. Now we are ready to search the data.

Settings    Permissions

Connection Name  
mysql-connection

Identity  
mysql-identity

Connection Type  
MySQL

Timezone  
Select...  
The time zone used by DB Connect to read time-related fields. By default the JVM time zone setting is used. [Learn More](#)

JDBC URL Settings

Host  
54.153.37.222

Port  
3306

Default Database  
mysql

JDBC URL Preview  
jdbc:mysql://54.153.37.222:3306/mysql?useSSL=true

Edit JDBC URL

Connection Properties

Key	Value

# Searching from the database

The screenshot shows a Splunk search interface with the following details:

**Source:** source="personal-details"

**Events:** 13,000 events (8/30/23 9:00:00.000 PM to 8/31/23 9:06:49.000 PM) | No Event Sampling

**Event Details:**

Type	Field	Value	Actions
Selected	host	54.153.37.222	▼
Selected	source	personal-details	▼
Selected	sourcetype	mysqld	▼
Event	age	40	▼
Event	email	brandonvillanueva@example.com	▼
Event	id	500	▼
Event	name	Damon Wallace	▼
Event	user_id	user-500	▼
Time	_time	2023-08-31T21:06:24.702+00:00	
Default	index	main	▼
Default	linecount	1	▼
Default	punct	-_--::--=---=_=_@.=--	▼

# Using the lookup

---

We can use lookup to connect with remote database tables, to get the required information which is connected to our indexed data. (for example using and User-ID).

Steps:

1. Navigate to **Data Lab → Lookups → New Lookup (Set reference search)**

The screenshot shows the 'New Lookup' wizard interface. The top navigation bar has five steps: 'Set Reference', 'Set Lookup SQL', 'Field Mapping', 'Set Properties', and 'Complete'. The first step, 'Set Reference', is highlighted with a green dot. Below the steps is a search interface with 'Search' and 'Saved Search' buttons. A code editor window displays a search query: `source="personal-details" | fields user_id`. At the bottom, a table lists three user records with columns 'user\_id' and '\_raw':

	user_id	_raw
1	user-500	2023-08-31 20:06:25.020, id="500", name="Damon Wallace", user_id="user-500", email="brandonvillanueva@example.com", age="40"
2	user-499	2023-08-31 20:06:25.020, id="499", name="Joseph Smith", user_id="user-499", email="wesleydavis@example.net", age="27"
3	user-498	2023-08-31 20:06:25.020, id="498", name="Sherri Hernandez", user_id="user-498", email="tabitha91@example.com", age="36"

# Using the lookup

## 2. Set lookup SQL which will look for the data from our remote database.

New Lookup      Set Reference Search      Set Lookup SQL      Field Mapping      Set Properties      Complete      <      Next      Cancel

Choose Table	Lookup SQL	SQL Columns																																																
Connection mysql-connection	<b>SQL Editor</b> <pre>1 select * from employment_info;</pre> <b>Format</b> <b>Execute SQL</b>	<code>id</code> <code>user_id</code> <code>salary</code> <code>country</code> <code>designation</code>																																																
Catalog Select...		<code>user_id</code> <code>_bkt</code> <code>_cd</code> <code>_indextime</code> <code>_kv</code> <code>_raw</code> <code>_serial</code> <code>_si</code> <code>_sourcetype</code> <code>_subsecond</code> <code>_time</code>																																																
Schema Select...																																																		
Table Search	<table border="1"><thead><tr><th></th><th><code>id</code> ▾</th><th><code>user_id</code> ▾</th><th><code>salary</code> ▾</th><th><code>country</code> ▾</th><th><code>designation</code> ▾</th></tr></thead><tbody><tr><td>1</td><td>1</td><td>user-001</td><td>74596</td><td>Belize</td><td>Clinical molecular geneticist</td></tr><tr><td>2</td><td>2</td><td>user-002</td><td>129416</td><td>Saint Helena</td><td>Artist</td></tr><tr><td>3</td><td>3</td><td>user-003</td><td>36699</td><td>Vietnam</td><td>Camera operator</td></tr><tr><td>4</td><td>4</td><td>user-004</td><td>148448</td><td>Tajikistan</td><td>Lexicographer</td></tr><tr><td>5</td><td>5</td><td>user-005</td><td>84038</td><td>United Kingdom</td><td>Risk manager</td></tr><tr><td>6</td><td>6</td><td>user-006</td><td>60442</td><td>Uruguay</td><td>Advertising copywriter</td></tr><tr><td>7</td><td>7</td><td>user-007</td><td>82737</td><td>Martinique</td><td>Psychologist clinical</td></tr></tbody></table>		<code>id</code> ▾	<code>user_id</code> ▾	<code>salary</code> ▾	<code>country</code> ▾	<code>designation</code> ▾	1	1	user-001	74596	Belize	Clinical molecular geneticist	2	2	user-002	129416	Saint Helena	Artist	3	3	user-003	36699	Vietnam	Camera operator	4	4	user-004	148448	Tajikistan	Lexicographer	5	5	user-005	84038	United Kingdom	Risk manager	6	6	user-006	60442	Uruguay	Advertising copywriter	7	7	user-007	82737	Martinique	Psychologist clinical	
	<code>id</code> ▾	<code>user_id</code> ▾	<code>salary</code> ▾	<code>country</code> ▾	<code>designation</code> ▾																																													
1	1	user-001	74596	Belize	Clinical molecular geneticist																																													
2	2	user-002	129416	Saint Helena	Artist																																													
3	3	user-003	36699	Vietnam	Camera operator																																													
4	4	user-004	148448	Tajikistan	Lexicographer																																													
5	5	user-005	84038	United Kingdom	Risk manager																																													
6	6	user-006	60442	Uruguay	Advertising copywriter																																													
7	7	user-007	82737	Martinique	Psychologist clinical																																													

# Using the lookup

## 3. Perform the field mapping and mention which data to fetch.

New Lookup      Set Reference Search      Set Lookup SQL      **Field Mapping**      Set Properties      Complete      <      Next      Cancel

**Search Fields Mapping**  
Map your selected search results fields to table columns.

Search Fields	Match	Table Columns
user_id	→	user_id

Add Search Field ▾

**Lookup Fields**  
Add your table columns as new Splunk fields.

Table Columns	AS	Aliases
salary	→	employee_salary
designation	→	employee_profile

Add Column ▾

**Preview Results**  
Preview lookup results with the following SPL

```
(...) | dbxlookup connection="mysql-connection" query="select * from employment_info;" "user_id" AS "user_id" OUTPUT "salary" AS "employee_salary", "designation" AS "employee_profile"
```

# Using the lookup

## 4. Set the Properties

New Lookup      Set Reference Search      Set Lookup SQL      Field Mapping      Set Properties      Complete      <      **Finish**      Cancel

Basic Information

Name: employment-info

Description: Optional

Application: Splunk DB Connect

Summary

Append this command to your search query to enrich your search results once it has been saved.

| dbxlookup lookup="employment-info"

# Using the lookup

Now you can search the relevant details using the lookup query.

New Search

```
source="personal-details" | dbxlookup lookup="employment-info"
```

✓ 16,000 events (8/30/23 9:00:00.000 PM to 8/31/23 9:22:37.000 PM) No Event Sampling ▾

Events (16,000) Patterns Statistics Visualization

Format Timeline ▾ - Zoom Out + Zoom to Selection × Deselect

List ▾ Format 20 Per Page ▾

Time	Event
8/31/23 9:21:24.697 PM	2023-08-31 21:21:24.697, id="500", name="Damon Wallace", user_id="user-500", email=brandonvillanueva@example.com, host=54.153.37.222, source=personal-details, sourcetype=mysql, age=40, employee_profile=Buyer, retail, employee_salary=75689

Event Actions ▾

Type	Field	Value	Actions
Selected	host	54.153.37.222	▼
	source	personal-details	▼
	sourcetype	mysql	▼
Event	age	40	▼
	email	brandonvillanueva@example.com	▼
	employee_profile	Buyer, retail	▼
	employee_salary	75689	▼
	id	500	▼
	name	Damon Wallace	▼
	user_id	user-500	▼

Time time ▾ 2023-08-31T21:21:24.697+00:00

◀ Hide Fields ▶ All Fields

SELECTED FIELDS

- a host 1
- a source 1
- a sourcetype 1

INTERESTING FIELDS

- # age 63
- a email 100+
- a employee\_profile 100+
- # employee\_salary 100+
- # id 100+
- a index 1
- # linecount 1
- a name 100+
- a punct 5
- a splunk\_server 1
- a user\_id 100+

+ Extract New Fields

# **Dashboard and Panels**

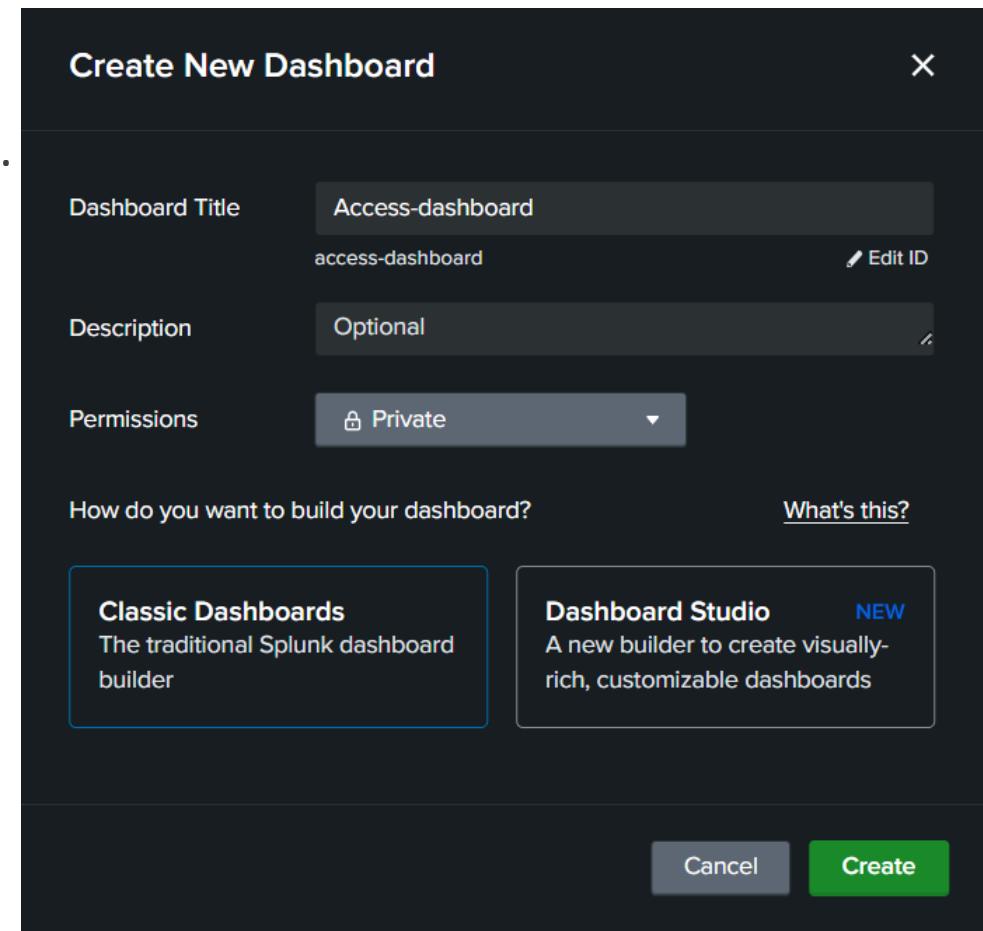
# Dashboard and Panels

A dashboard contains one or more panels.

Dashboard panels use searches to generate visualization.

Go to Dashboard → create a Dashboard

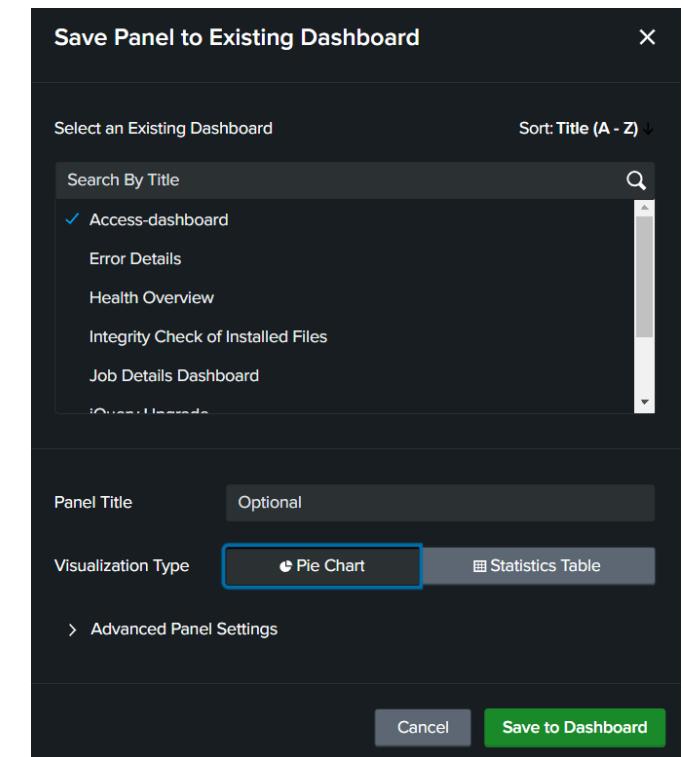
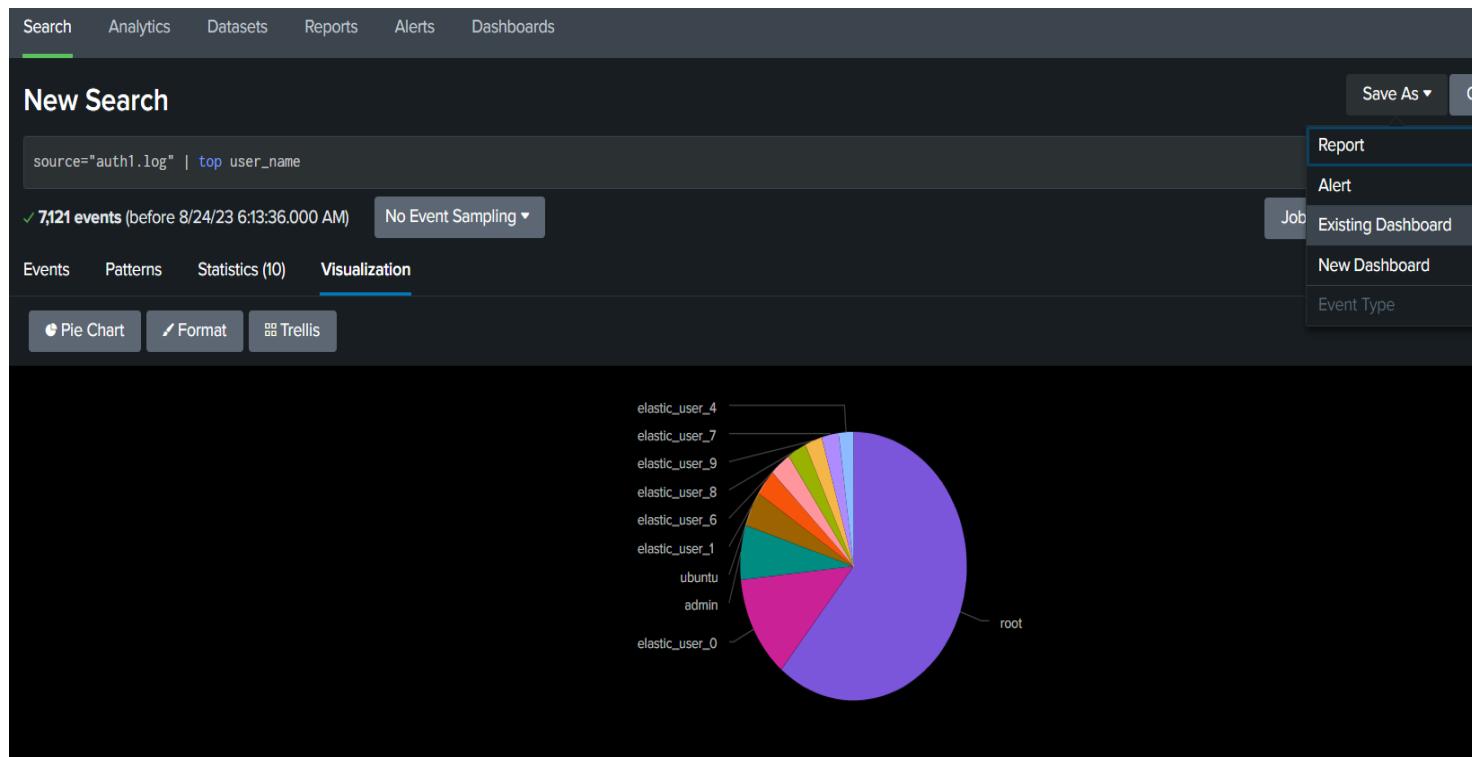
- Classic Dashboard
- Dashboard Studio



# Dashboard and Panels

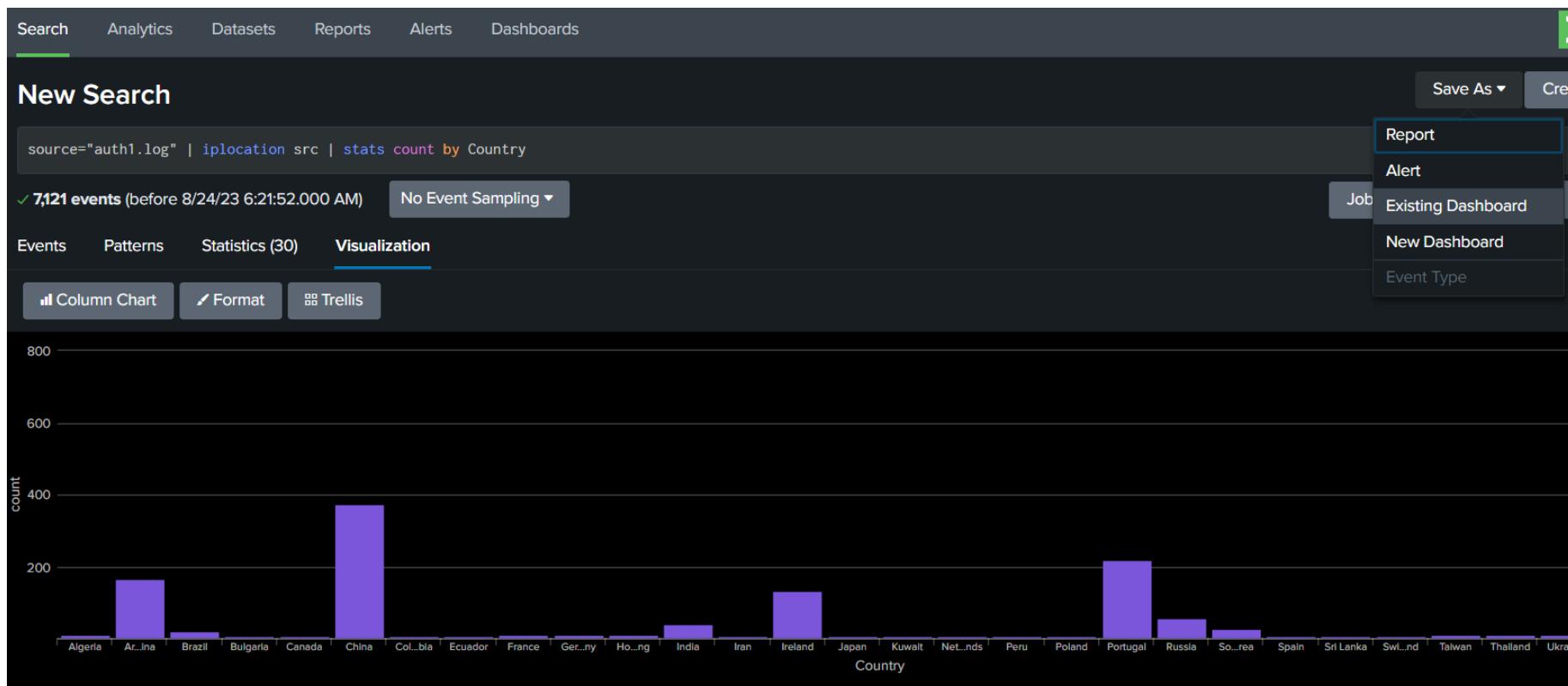
We need to add the panels which will visualize the searches.

Create a search and add it to the existing dashboard.



# Dashboard and Panels

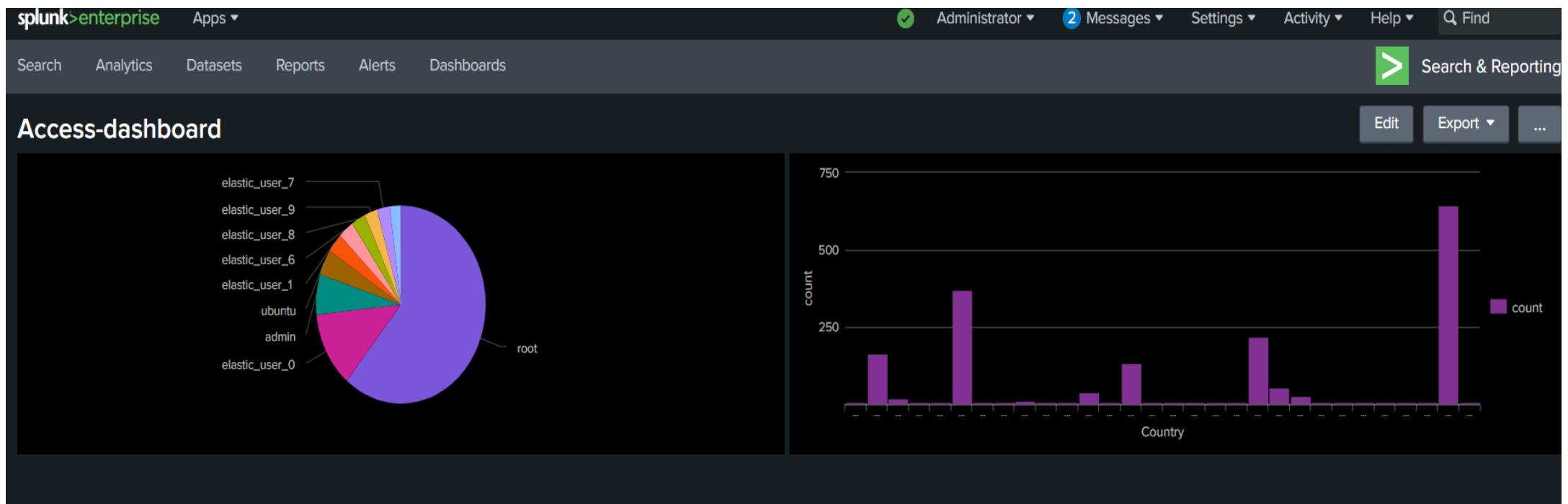
Add another one to the existing Dashboard.



# Dashboard and Panels

Now our dashboard is ready.

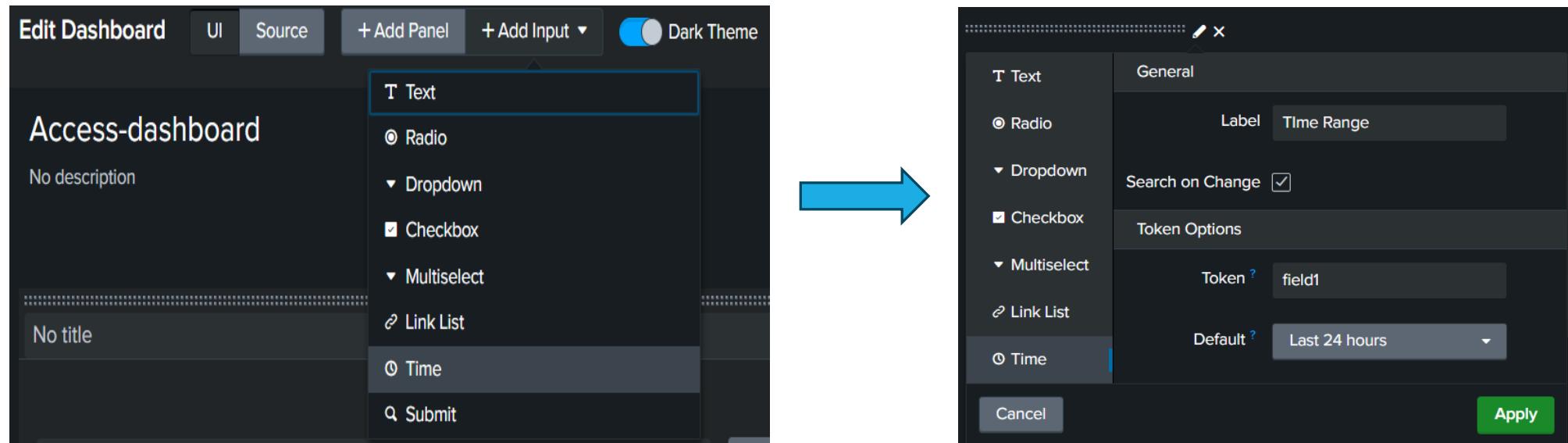
We can edit this chart by changing the search details or we can also make the changes within the source code of the chart.



# Dashboard and Panels: Time Range Picker

## Adding Time Range Picker:

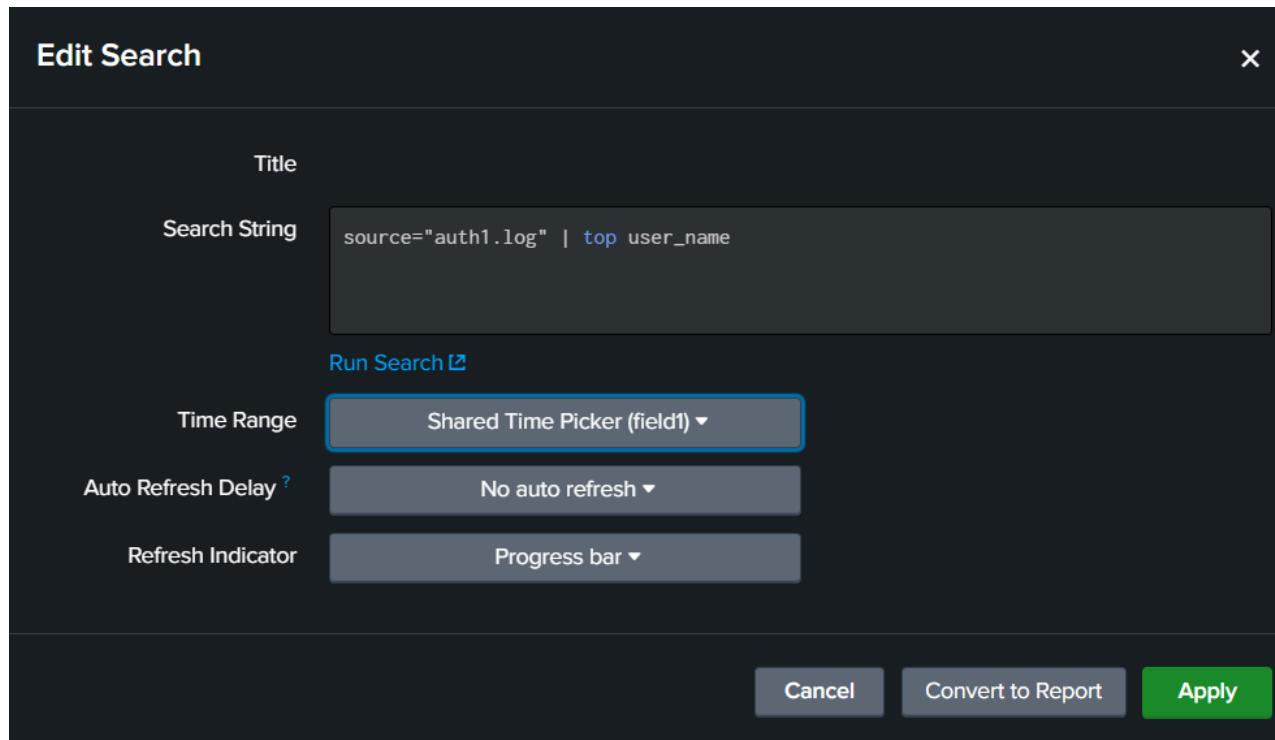
We want all the panels to sync with the time we select, Panels will be automatically updated with the time search or we can use a submit button to apply the changes.



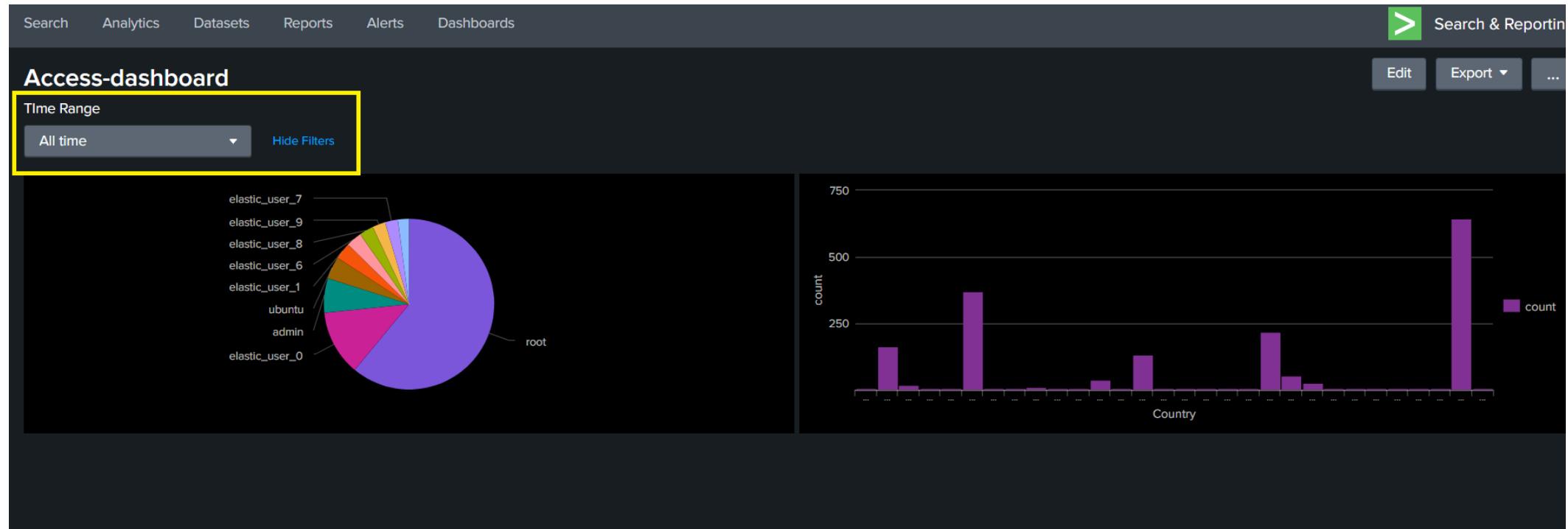
# Dashboard and Panels: Time Range Picker

---

Now we need to edit the panels to use the Time Range Picker by mentioning the token information.



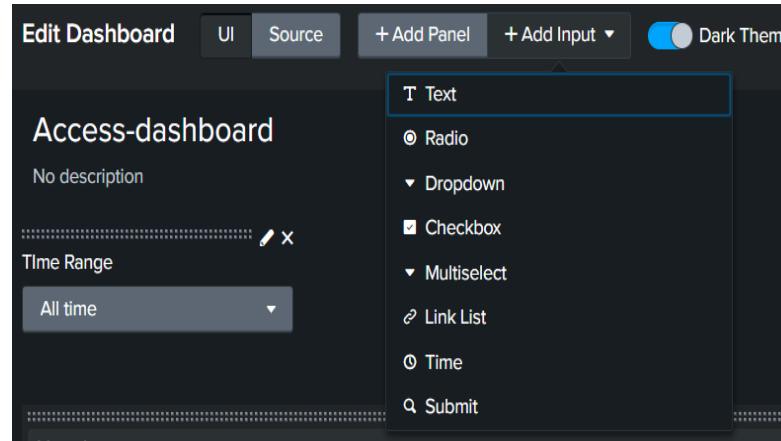
# Dashboard and Panels: Time Range Picker



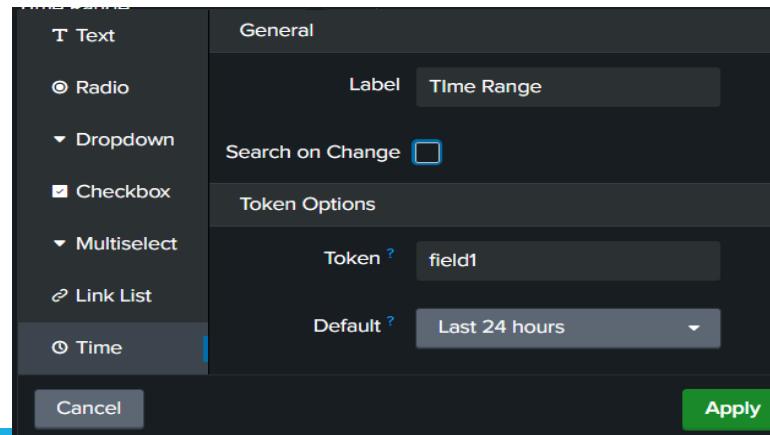
# Dashboard and Panels: Time Range Picker and Submit button

We want the changes once the user press the submit button.

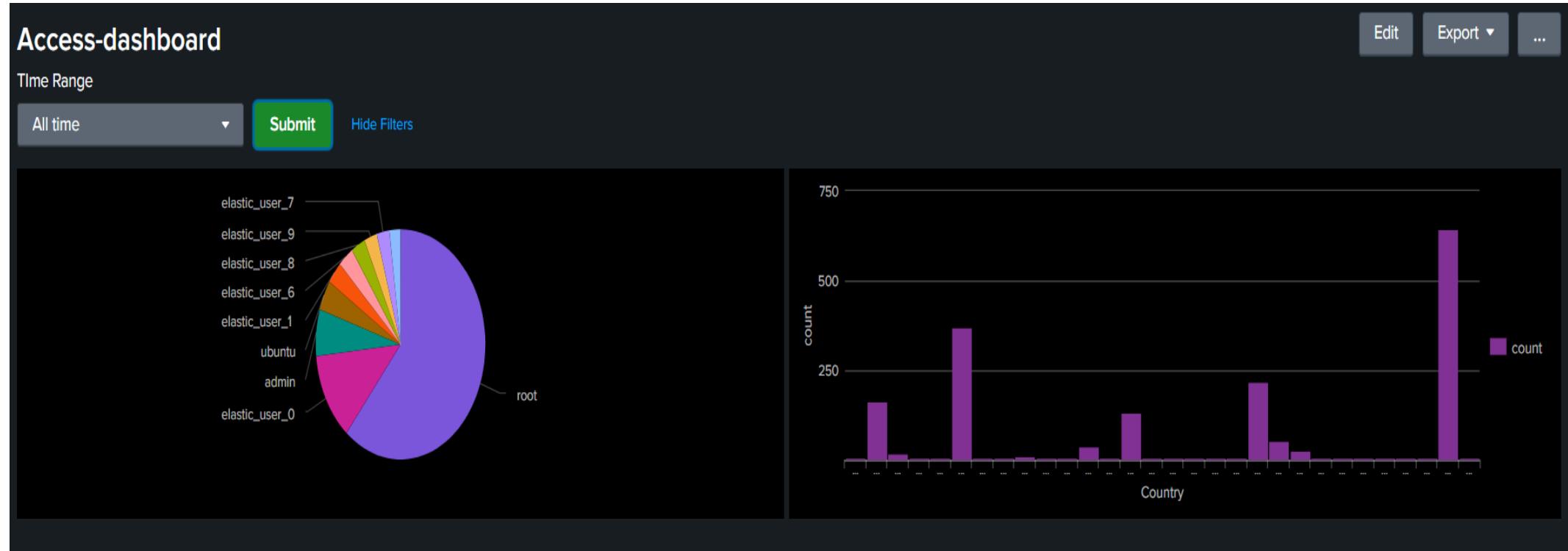
- Add a submit button



- Uncheck “search on change”

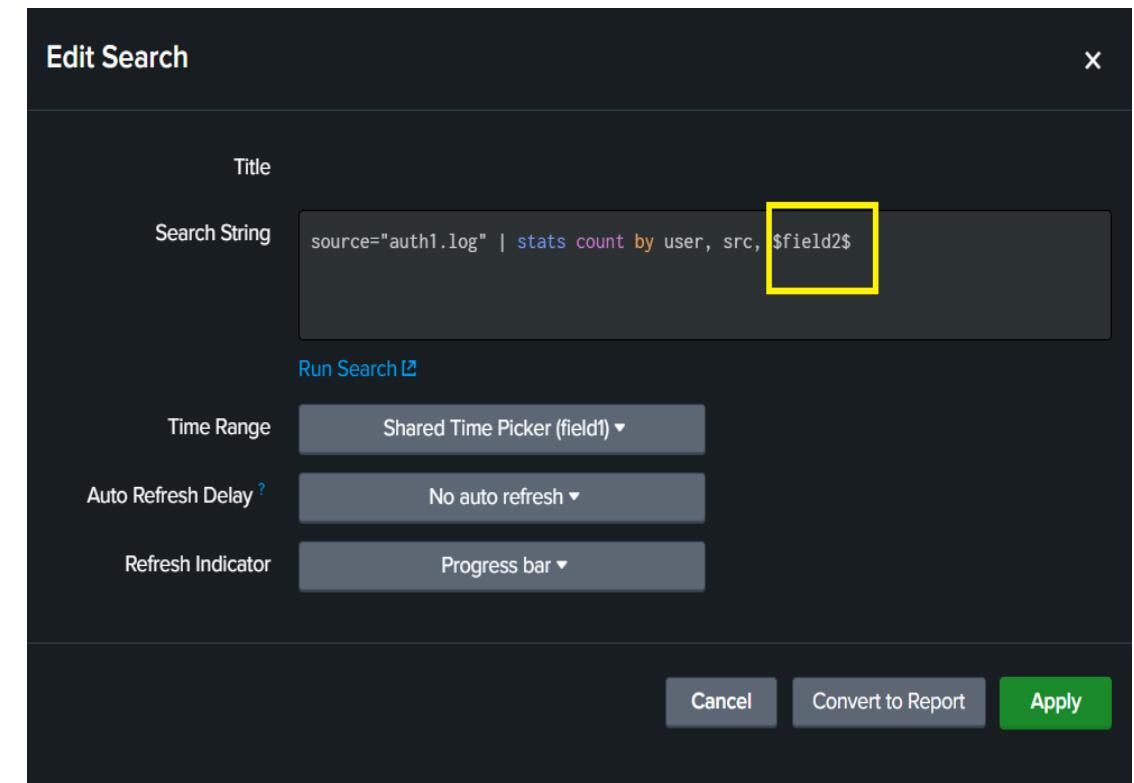
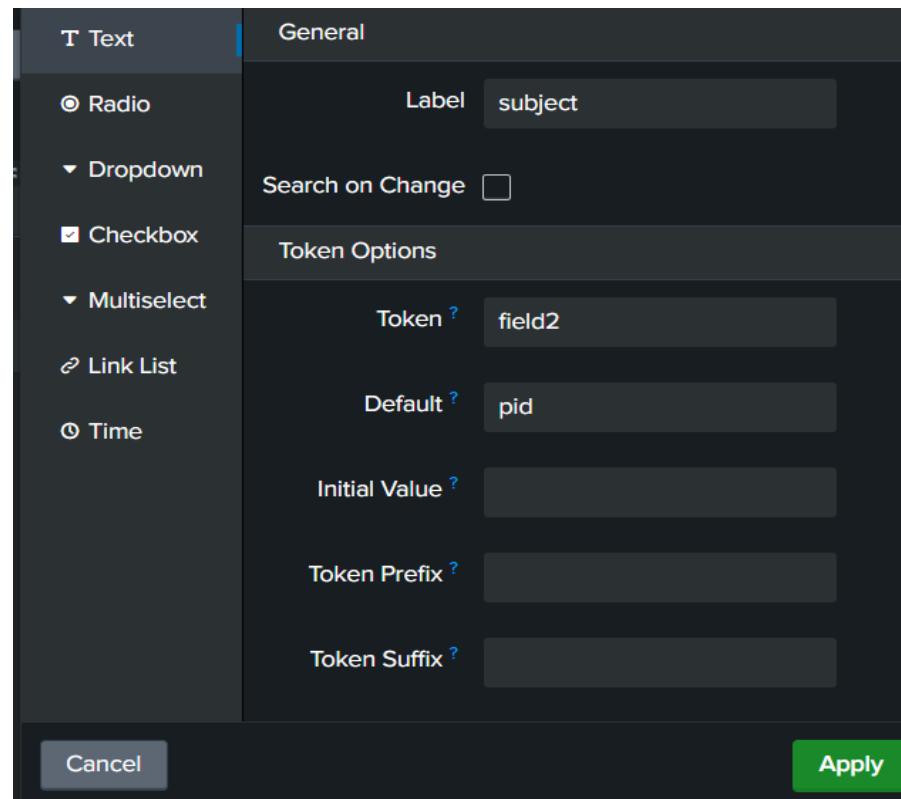


# Dashboard and Panels: Time Range Picker and Submit button

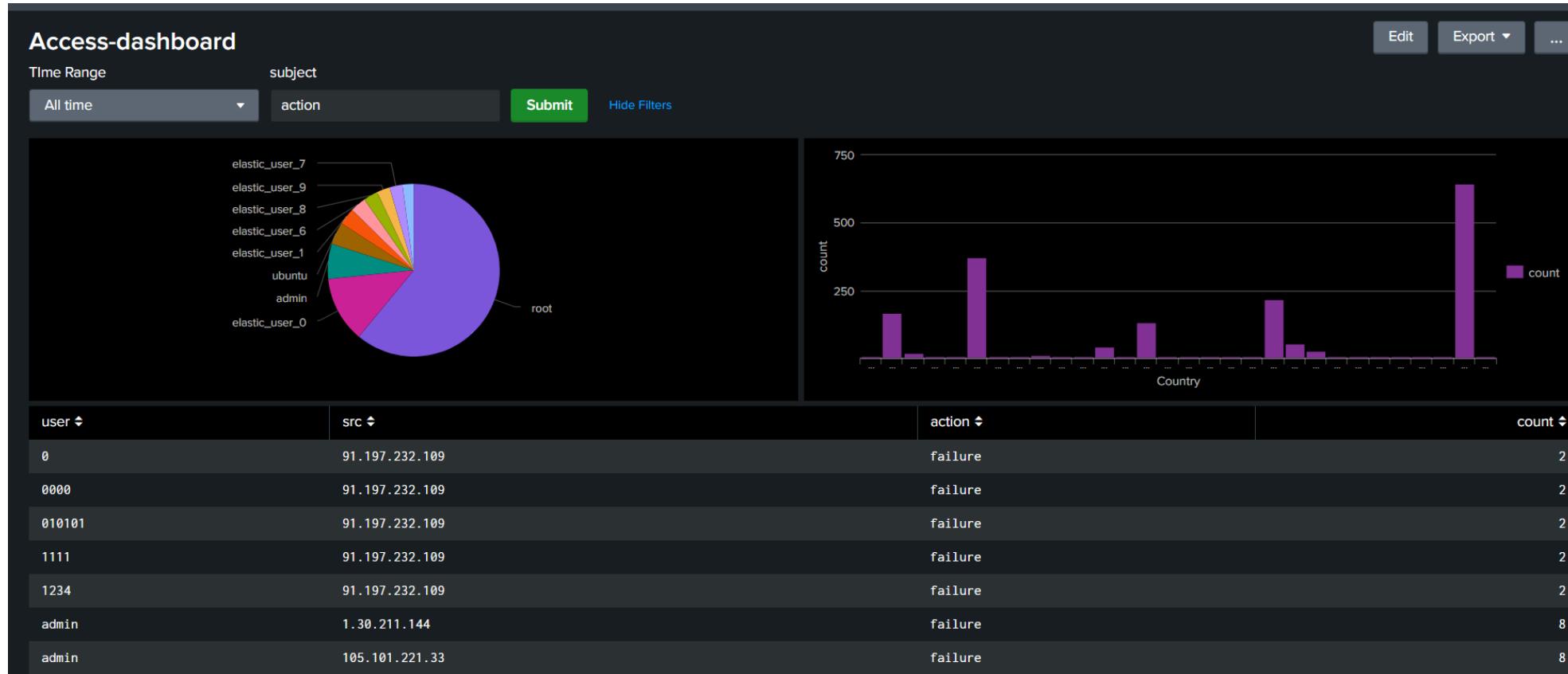


# Dashboard and Panels: Text Based Input

What if we want some additional field in the Panel using a variable. To achieve this, we need to create a text based input and mention the token in the search option of the Panel.

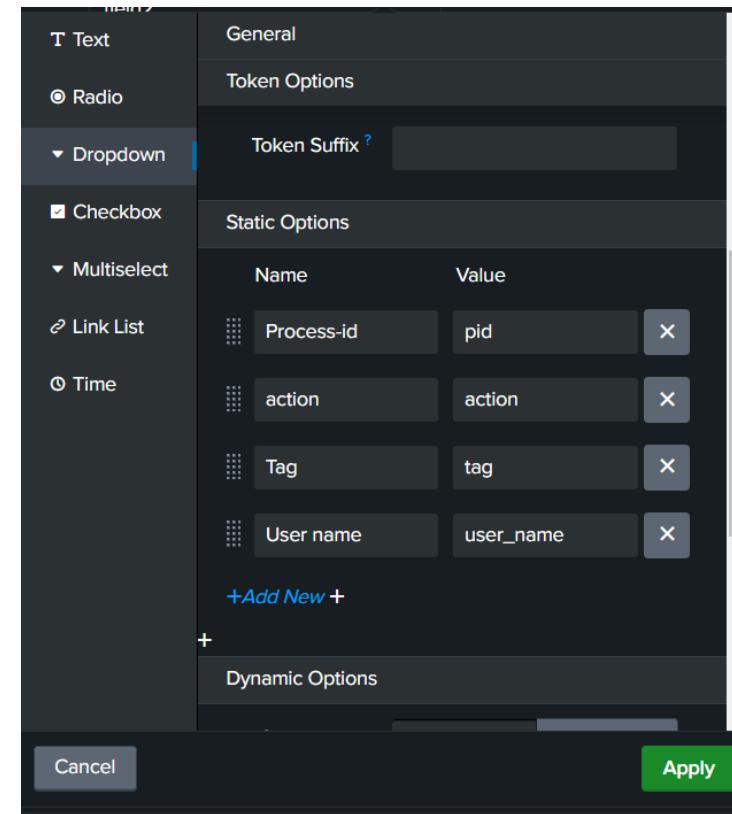
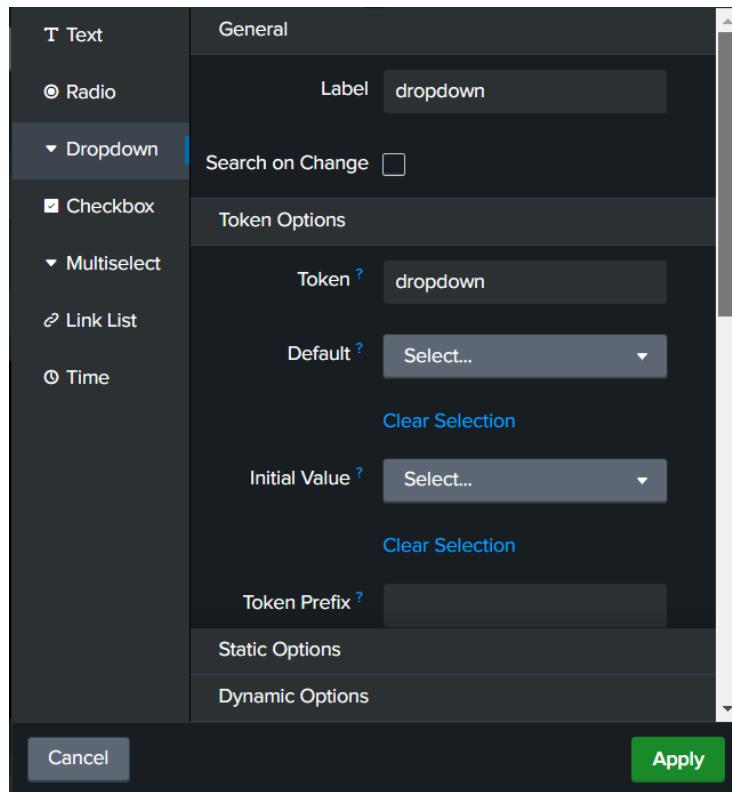


# Dashboard and Panels: Text Based Input



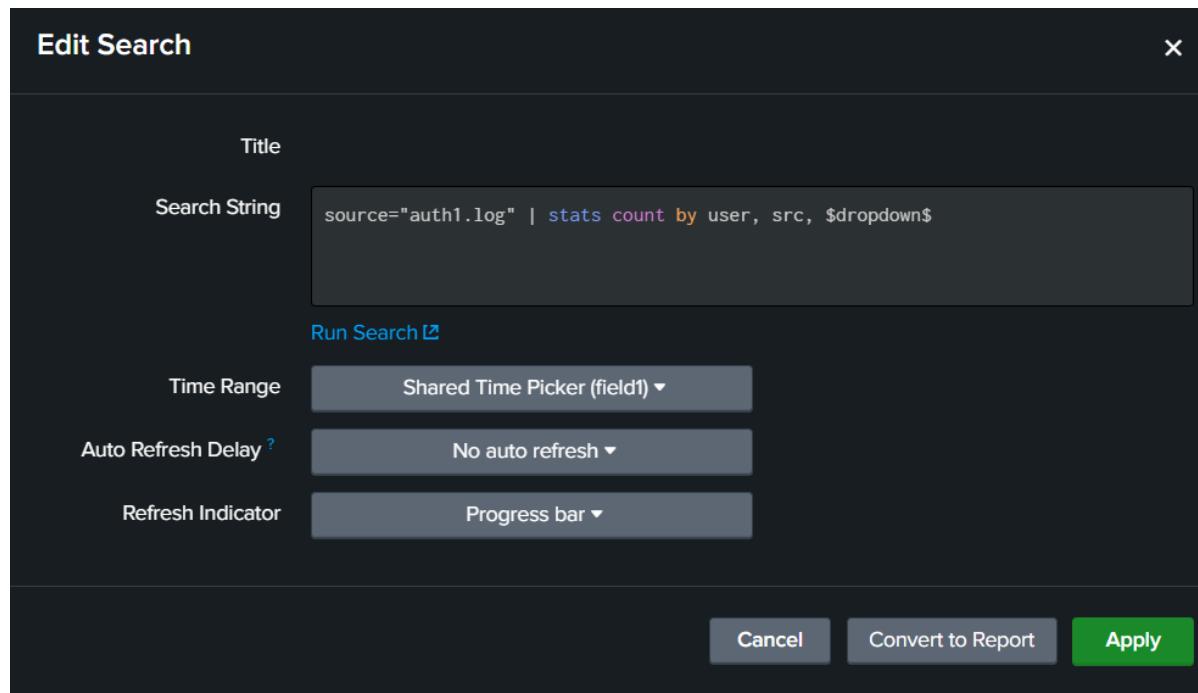
# Dashboard and Panels: Dropdown

If we want use a dropdown having a list of fields then we have to create one.

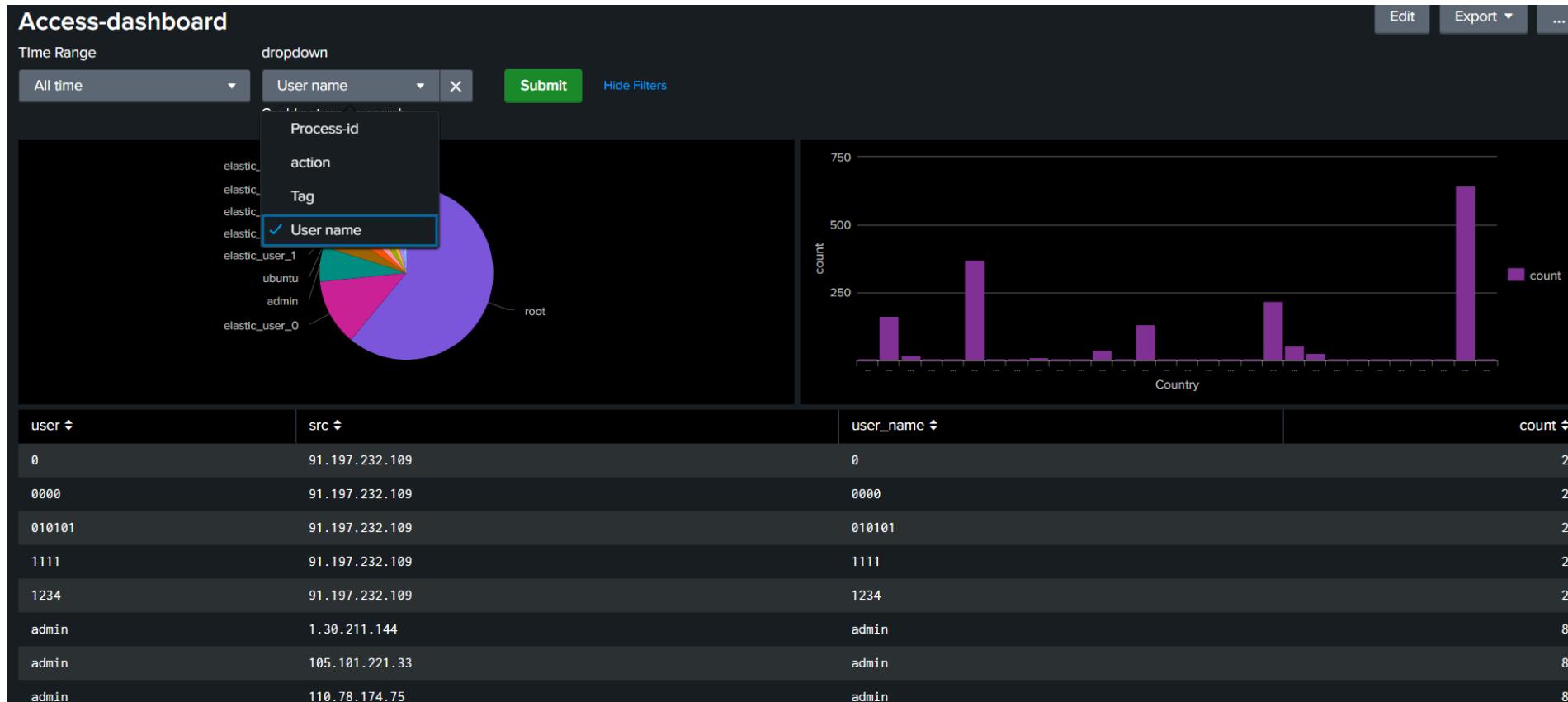


# Dashboard and Panels: Dropdown

Now edit the search using the dropdown token and check the latest dashboard.



# Dashboard and Panels: Dropdown



# Indexes

# What's in an Index?

---

Splunk Enterprise stores the data it processes in indexes. An index consists of a collection of subdirectories, called **buckets**. Buckets consist mainly of two types of files: **rawdata files** and **index files**.

Once data has been added to an index, you cannot edit or otherwise change the data. You can delete all data from an index or you can delete, and optionally archive, individual index buckets based on policy, but you cannot selectively delete individual events from storage.

## Default set of indexes

Splunk Enterprise comes with a number of preconfigured indexes, including:

- **main**: This is the default Splunk Enterprise index. All processed data is stored here unless otherwise specified.
- **\_internal**: Stores Splunk Enterprise internal logs and processing metrics.
- **\_audit**: Contains events related to the file system change monitor, auditing, and all user search history.

# Overview of Indexes

---

The index is a repository of Splunk data.

Splunk transforms incoming data into events which it stores in the indexes. Splunk Enterprise ships with several indexes, and you can create additional indexes as needed.

When Splunk indexes your data, it creates a number of files, these files fall into categories:

- the raw data in compressed form (raw data)
- Indexes that point to raw data (tsidx files, plus some meta-data files).

These files reside in a set of directories organized by age. These directories are called [buckets](#).

# Overview of Indexes

---

Indexes are being created in a specific directory `splunk/var/lib/splunk/`

```
root@ip-172-31-9-191:~# ls splunk/var/lib/splunk/
audit.dat           _internaldb          _metrics.dat        authDb      historydb    persistentstorage
_configtracker      _introspection      _metrics_rollup   defaultdb   kvstore      summary.dat
_configtracker.dat  _introspection.dat  _telemetry       fishbucket  main.dat     summarydb
internal.dat        _metrics            audit           hashDb      modinputs
root@ip-172-31-9-191:~#
```

Splunk Enterprise manages its indexes to facilitate flexible searching and fast data retrieval, eventually archiving them according to a user-configurable schedule.

Splunk Enterprise handles everything with flat files; it doesn't require any third-party database software running in the background.

Splunk Enterprise, by default, puts all user data into a single, preconfigured index. It also employs several other indexes for internal purposes.

# Overview of Indexes

---

## Index types

Splunk Enterprise supports two types of indexes:

- **Events indexes.** Events indexes impose minimal structure and can accommodate any type of data, including metrics data. Events indexes are the default index type.
- **Metrics indexes.** Metrics indexes use a highly structured format to handle the higher volume and lower latency demands associated with metrics data. Putting metrics data into metrics indexes results in faster performance and less use of index storage, compared to putting the same data into events indexes.

# How indexing works

Splunk Enterprise can **index** any type of time-series data (data with **timestamps**). When Splunk Enterprise indexes data, it breaks it into **events**, based on the timestamps.

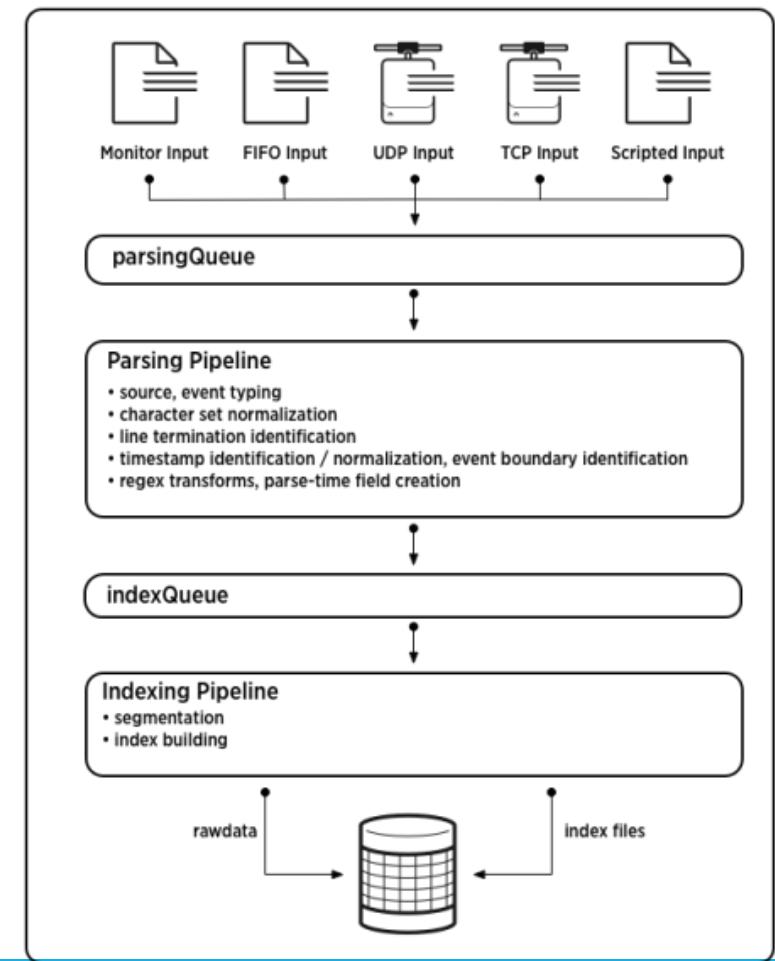
Data enters the indexer and proceeds through a pipeline where **event processing** occurs.

Event processing occurs in two main stages, parsing and indexing.

During parsing, Splunk Enterprise breaks these chunks into events which it hands off to the **indexing pipeline**.

In the indexing pipeline, Splunk Enterprise performs additional processing, including:

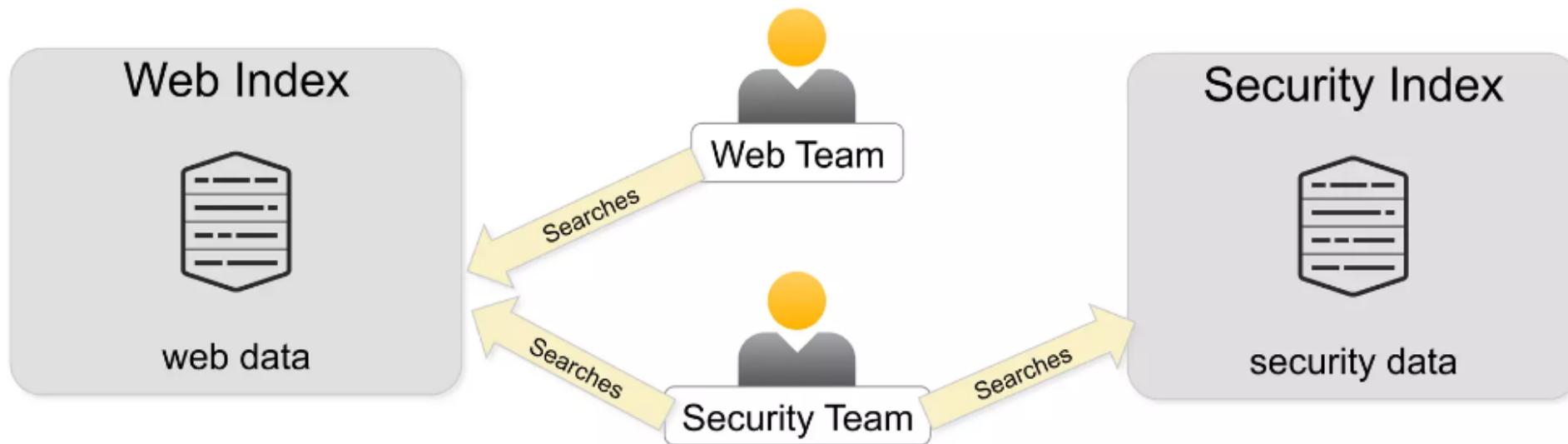
- Breaking all events into **segments** that can then be searched upon and it affects indexing and searching speed, search capability, and efficiency of disk compression.
- Building the index data structures.
- Writing the raw data and index files to disk, where post-indexing compression occurs.



# Why to create an index?

---

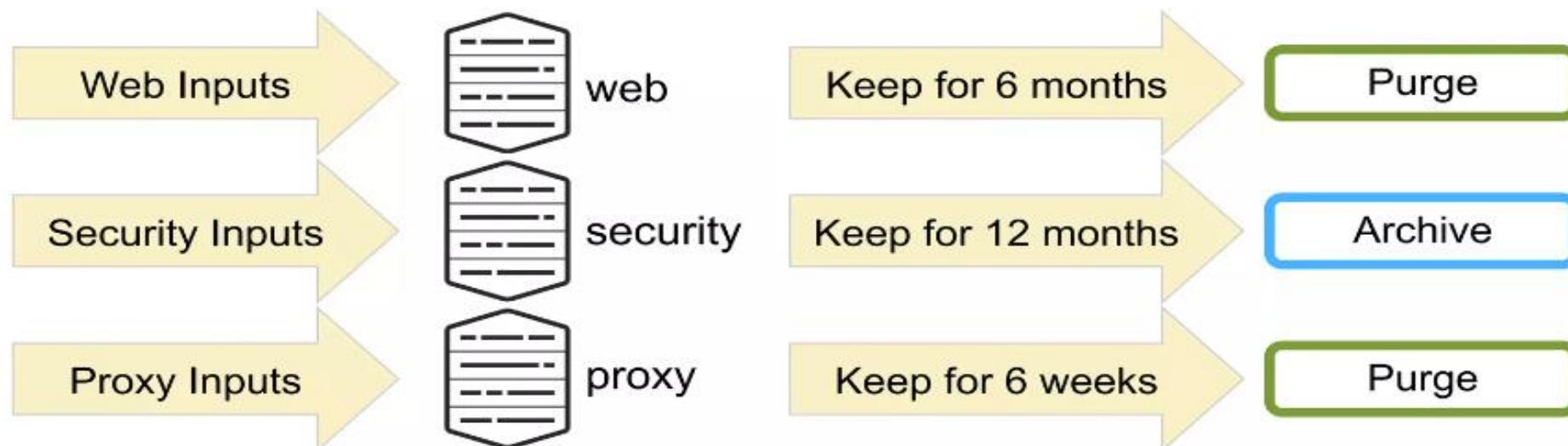
- Access control – segregate data into separate indexes to limit access by Splunk role



# Why to create an index?

---

- Retention
  - Retention is set on a per-index basis
  - Separate data into different indexes based on retention time
  - Splunk data retention can be managed by data age and/or by size



# What is tsidx file

---

In Splunk, the "**tsidx**" (**timestamp index**) file is one of the primary files associated with an index. It is responsible for storing metadata and indexing information related to timestamp values in the data that has been ingested and indexed by Splunk.

Here are some key points about the tsidx file:

- 1. Timestamp Indexing:** The tsidx file is used to index timestamp values within the events or log data. Splunk heavily relies on timestamps to organize and search through data efficiently.
- 2. Metadata Storage:** The tsidx file stores metadata about timestamps, such as their locations within the index, their offsets, and other information that helps Splunk quickly locate and retrieve events by time.
- 3. Search Optimization:** Splunk uses the information in the tsidx file to optimize searches based on time ranges. This allows for fast retrieval of events that fall within a specified time frame, which is a common use case when working with log data.

# What is tsidx file

---

4. **Reduction of Data Scans:** By indexing timestamps, the tsidx file helps Splunk avoid scanning the entire dataset when executing time-based searches, significantly improving search performance.
5. **Internal to Splunk:** The tsidx file is an internal file used by Splunk to manage and organize data. Users typically don't interact directly with tsidx files.
6. **Compressed:** The tsidx file may be compressed to save disk space, and it's often stored alongside other index files.

It's important to note that the tsidx file is part of the internal workings of Splunk, and it is not something that administrators or users typically need to manage or manipulate directly. Splunk takes care of creating, maintaining, and using these files to provide efficient searching and retrieval of data based on timestamps.

Search from raw data	index=_internal   stats count by sourcetype
Search from tsidx file	tstats count where index=_internal by sourcetype

# What is tsidx file

---

- Read in a line of data, apply segmentation, store tokens in TSIDX files
- Minor breakers: / : = @ . - \$ # % \ \_
- Major breakers: \r\n\s\t [] <> () {} | ! ; , ' " etc.
- Can be configured in segmenters.conf – but very rarely should!

127.0.0.1 - mm [24/Jun/2016:18:11:03.404] +0200]

# What is tsidx file

127.0.0.1 - mm [24/Jun/2016:18:11:03.404 +0200]

bin>splunk cmd walklex ..\var\lib\splunk\conf2016\_segmentation\db  
\hot\_v1\_1\1466784663-1466784663-15369347184008592423.tsidx ""

my needle:	10 1 127.0.0.1
3 1 -	11 1 18
4 1 0	12 1 2016
5 1 0200	13 1 24
6 1 03	14 1 24/jun/2016:18:11:03.404
7 1 1	15 1 404
8 1 11	27 1 jun
9 1 127	29 1 mm

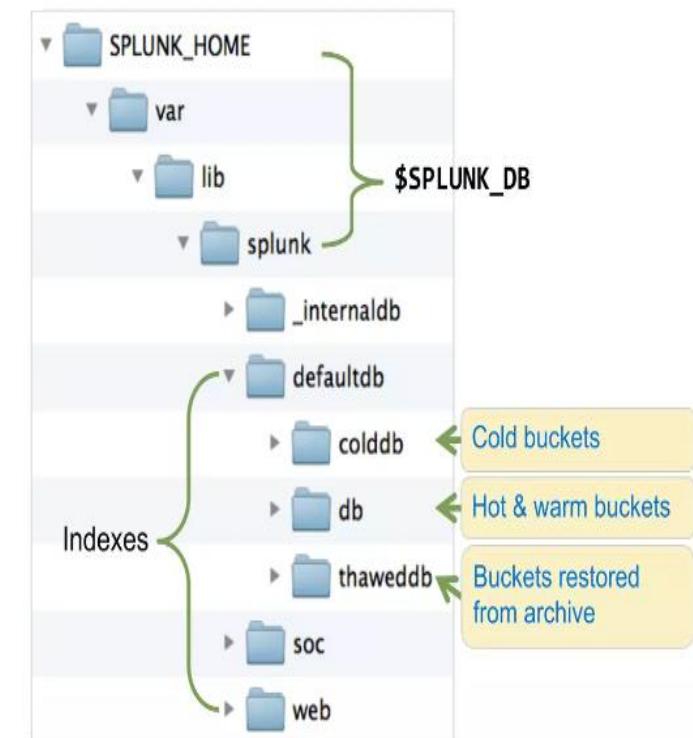
Each token is a pointer  
to the raw event

# Bucket Lifecycle

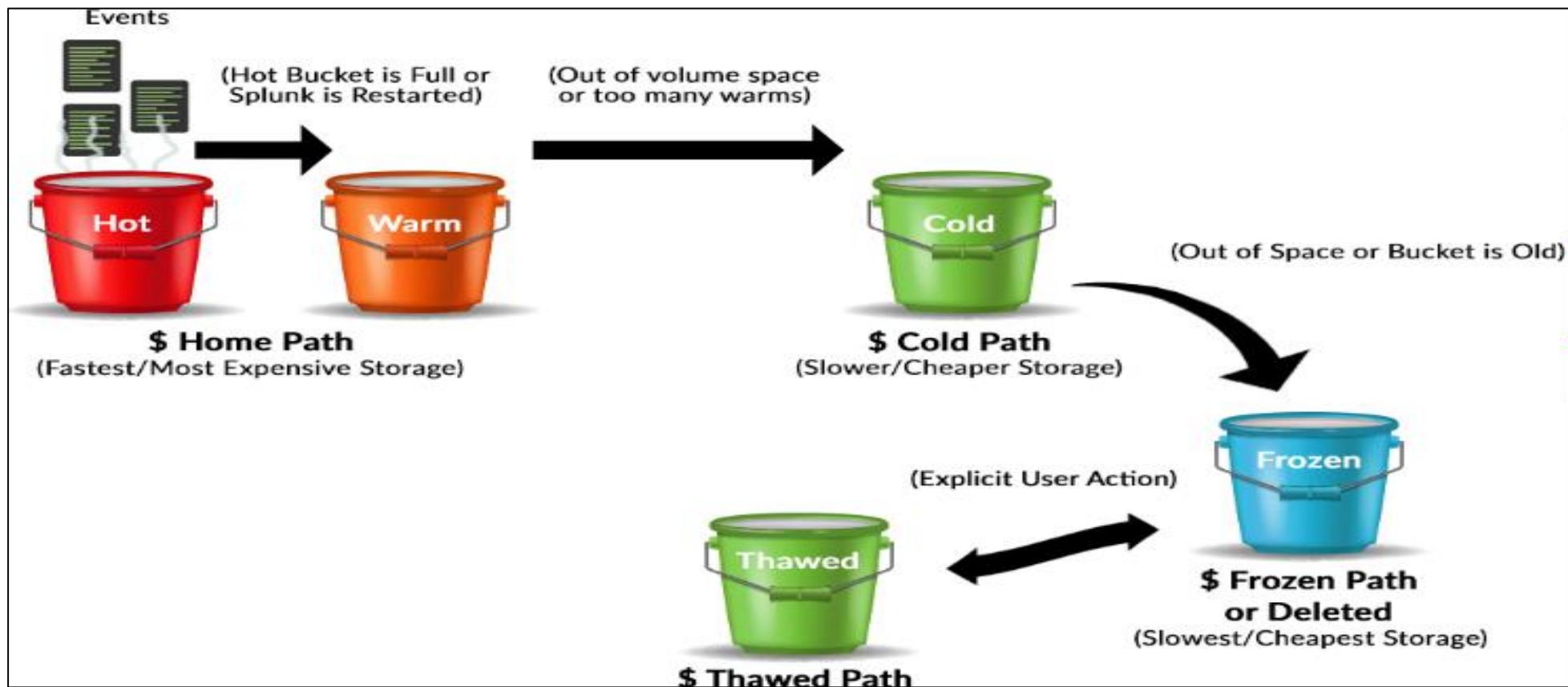
Splunk stores all its data in directories and these directories are called buckets.

A bucket moves through several stages as it ages which are mentioned below.

Lifecycle	Description
Hot	The new and most recent data stays here. Data is actively written here.
Warm	Data is rolled from hot and data is not actively written to warm buckets.
Cold	Data rolled from warm. The data is aged or archived so rarely searched
Frozen	Data rolled from cold and it is deleted but can be archived
Thawed	If data in frozen bucket is archived, it can be indexed again by thawed process.



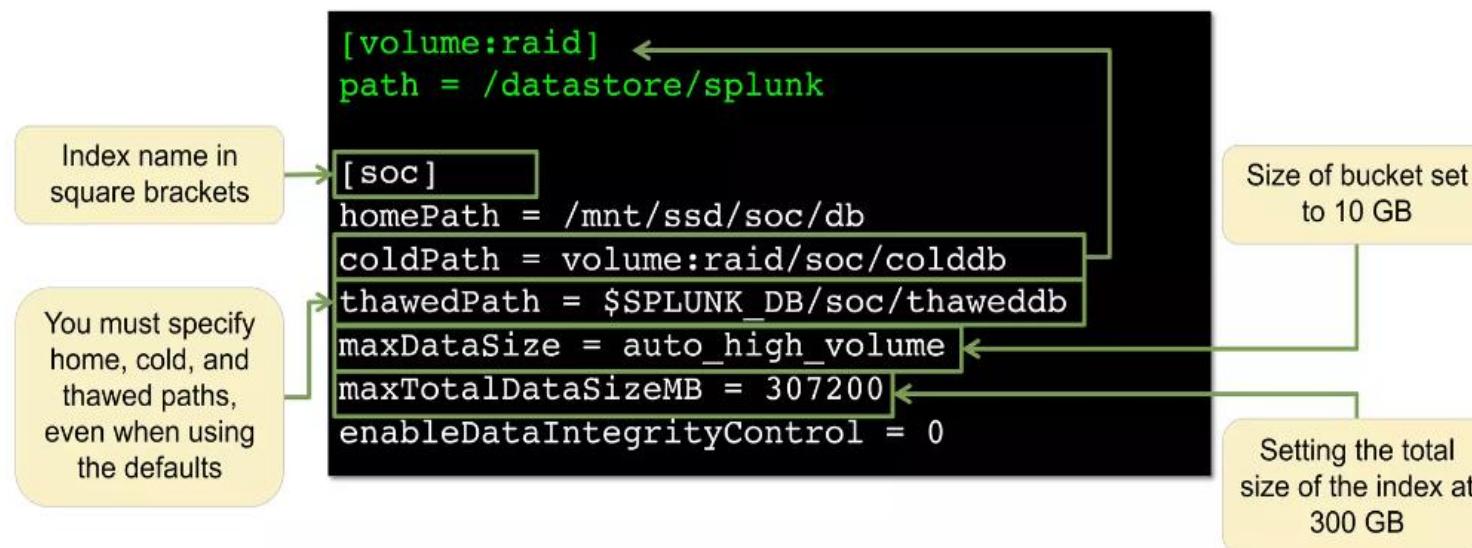
# Bucket Lifecycle



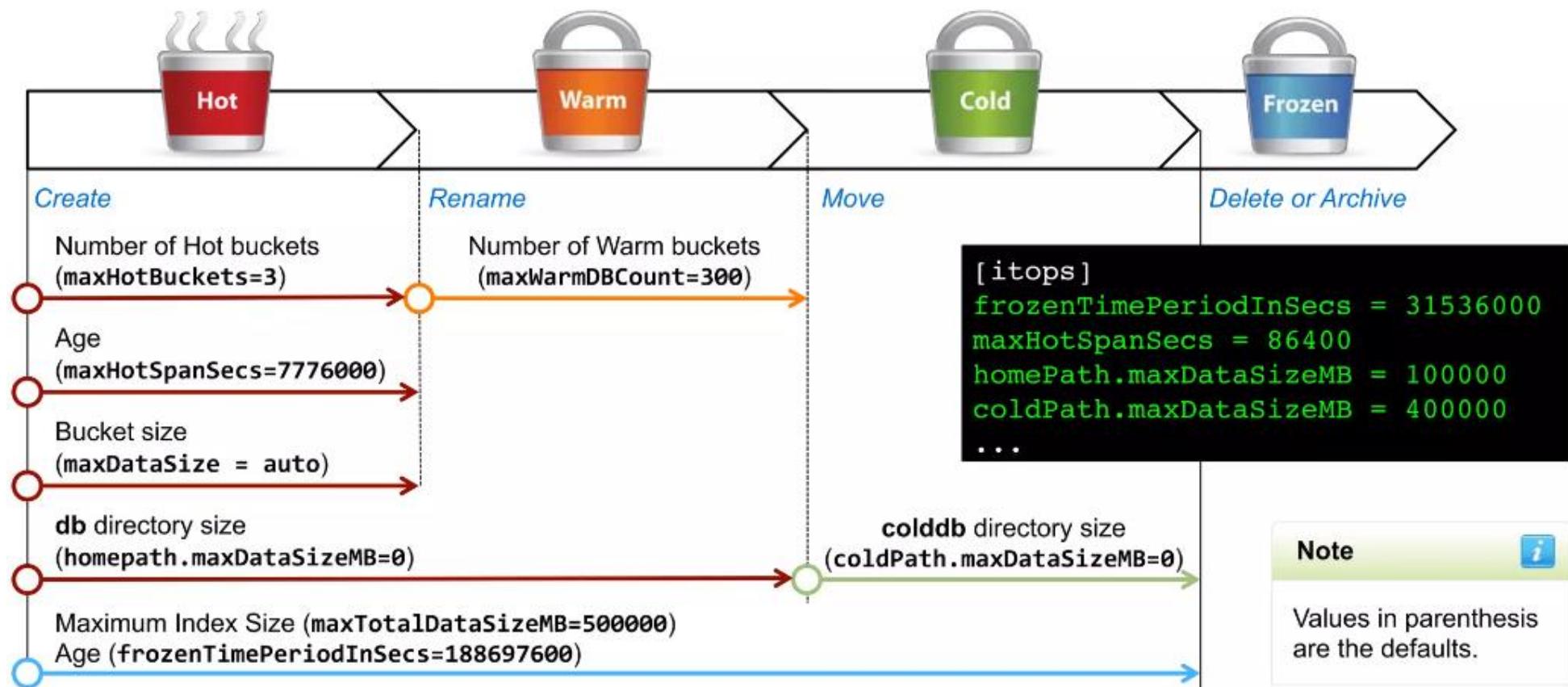
# Indexes.conf

Many advanced/optional attributes are not available in Splunk Web.

- The index stanza is created in **local/indexes.conf** of the selected app.
- New indexes default to 3 hot buckets at a time
- High Volume indexes should have up to 10 hot buckets (use **maxHotBuckets** key)



# Indexes.conf options



# Hot bucket to Warm Bucket

---

Buckets are rolled from hot to warm in following condition:

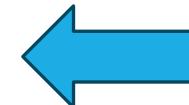
- We get too many hot buckets.
- Hot bucket has not received data for a while.
- Timespan of buckets is too large
- Bucket meta-data files have grown large.
- Index clustering replication error
- Splunk restarted.

# Hot bucket to Warm Bucket : Demo

Create a new index. Go to **settings → indexes → new Index**

Index will be created at below path.

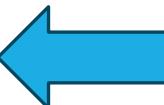
```
root@ip-172-31-9-191:~#  
root@ip-172-31-9-191:~# tree splunk/var/lib/splunk/new-index/  
splunk/var/lib/splunk/new-index/  
├── colddb  
├── datamodel_summary  
├── db  
└── CreationTime  
└── thaweddb
```



New Index

Index Name	new-index
Set index name (e.g., INDEX_NAME). Search using index=INDEX_NAME.	
Index Data Type	<input checked="" type="radio"/> Events <input type="radio"/> Metrics
The type of data to store (event-based or metrics).	
Home Path	optional
Hot/warm db path. Leave blank for default (\$SPLUNK_DB/INDEX_NAME/db).	
Cold Path	optional
Cold db path. Leave blank for default (\$SPLUNK_DB/INDEX_NAME/colddb).	
Thawed Path	optional
Thawed/resurrected db path. Leave blank for default (\$SPLUNK_DB/INDEX_NAME/thaweddb).	
Data Integrity Check	<input checked="" type="checkbox"/> Enable <input type="checkbox"/> Disable
Enable this if you want Splunk to compute hashes on every slice of your data for the purpose of data integrity.	
Max Size of Entire Index	5      GB ▾
Maximum target size of entire index.	
Max Size of Hot/Warm/Cold Bucket	auto      GB ▾
Maximum target size of buckets. Enter 'auto_high_volume' for high-volume indexes.	
Frozen Path	optional
Frozen bucket archive path. Set this if you want Splunk to automatically archive frozen buckets.	
<b>Save</b> <b>Cancel</b>	

```
root@ip-172-31-9-191:~# cat splunk/etc/apps/search/local/indexes.conf  
[new-index]  
coldPath = $SPLUNK_DB/new-index/colddb  
enableDataIntegrityControl = 0  
enableTsidxReduction = 0  
homePath = $SPLUNK_DB/new-index/db  
maxTotalDataSizeMB = 5120  
thawedPath = $SPLUNK_DB/new-index/thaweddb  
root@ip-172-31-9-191:~#  
root@ip-172-31-9-191:~#
```



# Hot bucket to Warm Bucket : Demo

Lets store some data in this new index.

Once data is stored, a hot directory will be created containing our data. As of now colddb is empty.

Lets restart our Splunk server.

Now you will see that hot data has been transferred to a warm directory.

This warm directory can be used to backup also.

```
root@ip-172-31-9-191:~#  
root@ip-172-31-9-191:~# tree splunk/var/lib/splunk/new-index/  
splunk/var/lib/splunk/new-index/  
├── colddb  
├── datamodel_summary  
└── db  
    └── CreationTime  
        └── hot_v1_0  
            ├── 1692904288-1692904288-18357661535984193142.tsidx  
            ├── Hosts.data  
            ├── SourceTypes.data  
            ├── Sources.data  
            ├── Strings.data  
            └── bucket_info.csv  
        └── rawdata  
            ├── 131099  
            │   ├── journal.zst  
            └── slicesv2.dat  
    └── thaweddb
```

```
root@ip-172-31-9-191:~#  
root@ip-172-31-9-191:~# tree splunk/var/lib/splunk/new-index/  
splunk/var/lib/splunk/new-index/  
├── colddb  
├── datamodel_summary  
└── db  
    └── CreationTime  
        └── GlobalMetaData  
            └── db_1692904288_1692904288_0  
                ├── 1692904288-1692904288-18357661535984193142.tsidx  
                ├── Hosts.data  
                ├── SourceTypes.data  
                ├── Sources.data  
                ├── Strings.data  
                ├── bloomfilter  
                ├── bucket_info.csv  
                └── optimize.result  
        └── rawdata  
            ├── journal.zst  
            └── slicemin.dat  
    └── thaweddb
```

# Warm to Cold bucket

---

Historical data goes to the cold buckets.

Allows older data to be kept on slower storage.

**Buckets roll from warm to Cold when we have too many Warm buckets.**

Default warm bucket counts are 300.

<b>Conf File</b>	indexes.conf
<b>Parameter</b>	[<index name>] coldPath = \$SPLUNK_DB/\$_index_name/colddb maxWarmDBCount = 300

# Warm to Cold bucket: Demo

Add more data to our index and it will create a hot data directory.

We can see here that there are warm and hot buckets available.

Lets add **maxWarmDBCount =1** to the configuration file.

**vi splunk/etc/apps/search/local/indexes.conf**

```
[new-index]
coldPath = ${SPLUNK_DB}/new-index/colddb
enableDataIntegrityControl = 0
enableTsidxReduction = 0
homePath = ${SPLUNK_DB}/new-index/db
maxTotalDataSizeMB = 5120
thawedPath = ${SPLUNK_DB}/new-index/thaweddb
maxWarmDBCount = 1
```

```
root@ip-172-31-9-191:~#
root@ip-172-31-9-191:~# tree splunk/var/lib/splunk/new-index/
splunk/var/lib/splunk/new-index/
├── colddb
├── datamodel_summary
└── db
    ├── CreationTime
    ├── GlobalMetaData
    ├── db_1692904288_1692904288_0
    │   ├── 1692904288-1692904288-18357661535984193142.tsidx
    │   ├── Hosts.data
    │   ├── SourceTypes.data
    │   ├── Sources.data
    │   ├── Strings.data
    │   ├── bloomfilter
    │   ├── bucket_info.csv
    │   ├── optimize.result
    │   └── rawdata
    │       ├── journal.zst
    │       ├── slicemin.dat
    │       └── slicesv2.dat
    └── hot_v1_1
        ├── 1692778927-1692776932-7773043623742932049.tsidx
        ├── Hosts.data
        ├── SourceTypes.data
        ├── Sources.data
        ├── Strings.data
        ├── bucket_info.csv
        └── rawdata
            └── 0
└── thaweddb
```

# Warm to Cold bucket: Demo

Now lets **restart our Splunk server**.

Now the old data has been rolled to colddb whereas the recent data is in the warm directory.

Since we have used the **maxWarmDBCount =1**, it reached to maximum and data has been transferred to the cold storage.

```
root@ip-172-31-9-191:~#
root@ip-172-31-9-191:~# tree splunk/var/lib/splunk/new-index/
splunk/var/lib/splunk/new-index/
├── colddb
│   └── db_1692778927_1692776932_1
│       ├── 1692778927-1692776932-7773043623742932049.tsidx
│       ├── Hosts.data
│       ├── SourceTypes.data
│       ├── Sources.data
│       ├── Strings.data
│       ├── bloomfilter
│       ├── bucket_info.csv
│       ├── optimize.result
│       ├── rawdata
│       │   ├── journal.zst
│       │   ├── slicemin.dat
│       │   └── slicesv2.dat
│       └── datamodel_summary
└── db
    ├── CreationTime
    └── GlobalMetaData
        └── db_1692904288_1692904288_0
            ├── 1692904288-1692904288-18357661535984193142.tsidx
            ├── Hosts.data
            ├── SourceTypes.data
            ├── Sources.data
            ├── Strings.data
            ├── bloomfilter
            ├── bucket_info.csv
            ├── optimize.result
            ├── rawdata
            │   ├── journal.zst
            │   ├── slicemin.dat
            │   └── slicesv2.dat
            └── thaweddb
```

# Cold to Frozen bucket

---

The data available in this storage is no longer searchable.

Data rolls from Cold to Frozen when :

- The total size of the index (Hot + Warm + Cold) grows too large.
- The oldest event in a bucket exceeds a specific age.

Default freezing process

- TSIDX file is removed
- Bucket is copied to a destination you specify
- Splunk no longer manages the data – you are in charge

Custom freezing process

- You provide a custom script

Conf File	indexes.conf
	[<index name>] maxTotalDataSizeMB = frozenTimePeriodInSecs = coldToFrozenDir = coldToFrozenScript = thawedPath =

# Cold to Frozen bucket: Demo

Add **heavy data** to surpass the max size of our index.

You can see now that the cold data has been deleted and new data has been stored in hot directory.

Now lets add the **coldToFrozenDir** in the configuration file, in order to save the cold data from deletion.

**Note: If we do not mention the path then the data will be deleted.**

```
root@ip-172-31-9-191:~# tree splunk/var/lib/splunk/new-index/
splunk/var/lib/splunk/new-index/
├── colddb
└── datamodel_summary
    └── db
        ├── CreationTime
        ├── GlobalMetaData
        ├── hot_quar_v1_3
        │   ├── 1555157821-1552682907-1546908377340879235.tsidx
        │   ├── 1587271076-1552682907-1555136778765756770.tsidx
        │   ├── 1587392069-1552682907-1550956036125012725.tsidx
        │   ├── 1587392069-1555157821-1547606021763645167.tsidx
        │   ├── 1587392069-1587271076-1558276769226326200.tsidx
        │   ├── Hosts.data
        │   ├── SourceTypes.data
        │   ├── Sources.data
        │   ├── Strings.data
        │   ├── bucket_info.csv
        │   └── rawdata
        │       ├── 6033230
        │       │   ├── journal.zst
        │       │   └── slicesv2.dat
        │       └── splunk-autogen-params.dat
        └── hot_v1_2
            ├── 1680188101-1679922416-1540930019088064494.tsidx
            ├── 1682000069-1679922416-1558248800399293623.tsidx
            ├── 1682034435-1680260879-1549911577388015657.tsidx
            ├── 1682034936-1679922416-1546655824673946321.tsidx
            ├── 1682034936-1679922416-1549324386639154864.tsidx
            ├── 1682034936-1680188101-1541199523990897566.tsidx
            ├── 1682034936-168194591-1554256387779508825.tsidx
            ├── Hosts.data
            ├── SourceTypes.data
            ├── Sources.data
            ├── Strings.data
            ├── bucket_info.csv
            └── rawdata
                ├── 9181018
                │   ├── journal.zst
                │   └── slicesv2.dat
                └── splunk-autogen-params.dat
        └── hot_v1_4
            ├── 1617013972-1616850416-1545354883374943264.tsidx
            ├── 1650464069-1616850416-1551101510962287719.tsidx
            ├── 1650464069-1616850416-1558487510386670553.tsidx
            ├── 1650464069-1617013972-1546668374568361074.tsidx
            ├── Hosts.data
            └── SourceTypes.data
```

# Cold to Frozen bucket: Demo

Now restart the server.

Hot data will be moved to the warm data.

Cold data will be moved to the frozen directory.

```
root@ip-172-31-9-191:~#  
root@ip-172-31-9-191:~# tree /tmp/frozen  
/tmp/frozen  
└── db_1587392069_1552682907_3  
    └── rawdata  
        └── journal.zst  
└── db_1650464069_1616850416_4  
    └── rawdata  
        └── journal.zst
```

```
root@ip-172-31-9-191:~# tree splunk/var/lib/splunk/new-index/  
splunk/var/lib/splunk/new-index/  
├── colddb  
├── datamodel_summary  
├── db  
│   ├── CreationTime  
│   └── GlobalMetaData  
├── db_1682034936_1679922416_2  
│   ├── 1680188101-1679922416-1540930019088064494.tsidx  
│   ├── 1682000069-1679922416-1558248800399293623.tsidx  
│   ├── 1682034435-1680260879-1549911577388015657.tsidx  
│   ├── 1682034936-1679922416-1546655824673946321.tsidx  
│   ├── 1682034936-1679922416-1549324386639154864.tsidx  
│   ├── 1682034936-1680188101-1541199523990897566.tsidx  
│   ├── 1682034936-1681949591-1554256387779508825.tsidx  
│   ├── Hosts.data  
│   ├── SourceTypes.data  
│   ├── Sources.data  
│   ├── Strings.data  
│   └── bucket_info.csv  
└── rawdata  
    ├── journal.zst  
    ├── slicemin.dat  
    └── slicesv2.dat  
splunk-autogen-params.dat  
thaweddb
```

# Thawing Data

---

Bringing data back from the deep freeze

It is a manual process:

- copy frozen buckets to thawed path
- use the rebuild command to re-index the data

**Re-indexing:**

- Does not count against your license
- Takes time (use the same estimates for indexing new data)

# Thawing Data : Demo

Copy the data from the frozen directory to thaweddb

```
cp -R /tmp/frozen/db_1587392069_1552682907_3 splunk/var/lib/splunk/new-index/thaweddb
```

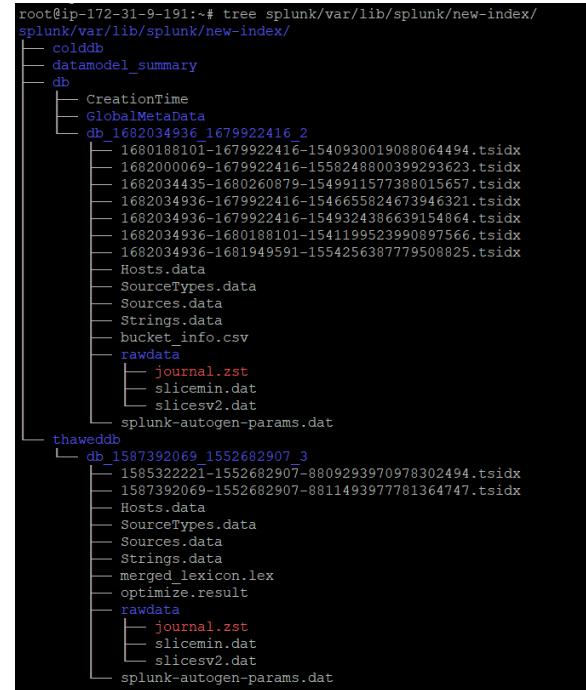
**Rebuild command:**

```
./splunk/bin/splunk rebuild splunk/var/lib/splunk/new-index/thaweddb/db_1587392069_1552682907_3
```

**Restart Splunk:**

```
./splunk/bin/splunk restart
```

Now we can access our data



# Workflow actions

---

Splunk workflow actions allows us to add interactivity between the indexed fields and other web resources

## **Example:**

There is a field called as src in access\_combined or linux\_secure log file

You can add option for lookup on the IP address mentioned in src field.

# Workflow actions: Demo

1. Create a workflow action (settings → field → workflow action)

Label \* whois\_lookup  
Enter the label that appears for this action. Optionally, incorporate a field's value by enclosing the field name in dollar signs, e.g. 'Search for ticket number : \$ticketnum\$'.

Apply only to the following fields src  
Specify a comma-separated list of fields that must be present in an event for the workflow action to apply to it. When fields are specified, the workflow action only appears in the field menus for those fields; otherwise it appears in all field menus.

2. Mention the field and URI

Apply only to the following event types  
Specify a comma-separated list of event types that an event must be associated with for the workflow action to apply to it.

Show action in Event menu  
Action type \* link

## Link configuration

URI \* https://www.maxmind.com/en/high-risk-ip-sample/\$src\$  
Enter the location to link to. Optionally, specify fields by enclosing the field name in dollar signs, e.g. http://www.google.com/search?q=\$host\$.

Open link in New window  
Link method get

# Workflow actions: Demo

List ▾ Format 20 Per Page ▾

i	Time	Event
▼	8/20/23 12:25:51.000 PM	Aug 20 12:25:51 ip-172-31-7-149 sshd[17875]: Failed password for invalid user p from 185.220.100.241 port 40462 ssh2

Event Actions ▾

Build Event Type	Value	Actions
Extract Fields	ip-172-31-27-153	▼
Show Source	risky-logs.txt	▼
whois_lookup	linux_secure	▼

Event	action ▾	failure	▼
	app ▾	ssh	▼
	dest ▾	ip-172-31-7-149	▼
	dvc ▾	ip-172-31-7-149	▼
	eventtype ▾	failed_login ( authentication )	▼
		nix_errors ( error )	▼

# **Workload Management**

# Workload Management

---

**Align your Splunk resources to your business priorities**

- **Admission rules framework** to filter rogue searches
- Prioritize critical searches using **workload rules**
- Reduce the impact and surface area of rogue searches using **automated remediation**
- Flexibly manage workload during peak/off-peak using **schedule based rules**
- Guide users faced with aborted searches with **custom user messaging**

# Workload Management: Demo

<b>name</b>	<b>Condition</b>
Alltime_searchduration	search_time_range=alltime AND (NOT (role=admin OR role=sc_admin))
Peaktime_newjoinee	role=newjoinee
Wildcard_index	index=* AND (NOT role=admin) AND (NOT app=splunk_instance_monitoring)

## Admission Rules

Admission Rule	Predicate (Condition)	Rule Action	User Message	Schedule	Actions	Status
alltime_searchduration	search_time_range=alltime AND (NOT (role=admin OR role=sc_admin))	Filter search	the time duration should be less than 7 days	Always On	Edit Delete	<input checked="" type="checkbox"/> Enabled
peaktime_newusers	role=newjoinee	Filter search	you are not allowed to search during the peak hours	Time Range 2024-07-15 (0:00) - 2024-07-16 (0:00)	Edit Delete	<input checked="" type="checkbox"/> Enabled
wildcard_index	index=* AND (NOT role=admin) AND (NOT app=splunk_instance_monitoring)	Filter search	please specify an index	Always On	Edit Delete	<input checked="" type="checkbox"/> Enabled

# Workload Pools

---

## Search Pools

Category	Workload Pool	CPU Weight <a href="#">?</a>	Allocated CPU % <a href="#">?</a>	Memory Limit % <a href="#">?</a>	Allocated Memory Limit % <a href="#">?</a>	Default Pool
search	high_perf	60	30.00%	100%	65.00%	
search	limited_perf	5	2.50%	100%	65.00%	
search	standard_perf	35	17.50%	100%	65.00%	

# Workload Pools: Demo

Workload Rule	Predicate (Condition)	Rule Action	User Message
abort_longrunning	role=newusers AND run time>10	Abort search	search time exceeded
low_priority_users	role=newusers	Place search in a Pool: limited_perf	
high_priority_users	role=security OR role=admin OR role=sc_admin	Place search in a Pool: high_perf	
throttle_longrunning	role=security AND runtime>10	Move search to alternate Pool: limited_perf	the search run for longer time so throttled
high_priority_apps	app=splunk_instance_monitoring	Place search in a Pool: high_perf	

# Workload Pools: Demo

---

## Workload Rules

Order	Workload Rule	Predicate (Condition)	Rule Action	User Message	Schedule	Actions	Status
1	abort_longrunning	role=newusers AND runtime>10	Abort search	search time exceeded	Always On	Edit Delete	<input checked="" type="button"/> Enabled
2	low_priority_users	role=newusers	Place search in a Pool: limited_perf		Always On	Edit Delete	<input checked="" type="button"/> Enabled
3	high_priority_users	role=security OR role=admin OR role=sc_admin	Place search in a Pool: high_perf		Always On	Edit Delete	<input checked="" type="button"/> Enabled
4	throttle_longrunning	role=security AND runtime>10	Move search to alternate Pool: limited_perf	the search run for longer time so throttled	Always On	Edit Delete	<input checked="" type="button"/> Enabled
5	high_priority_apps	app=splunk_instance_monitoring	Place search in a Pool: high_perf		Always On	Edit Delete	<input checked="" type="button"/> Enabled

# **Forwarder**

# Overview of Universal Forwarder

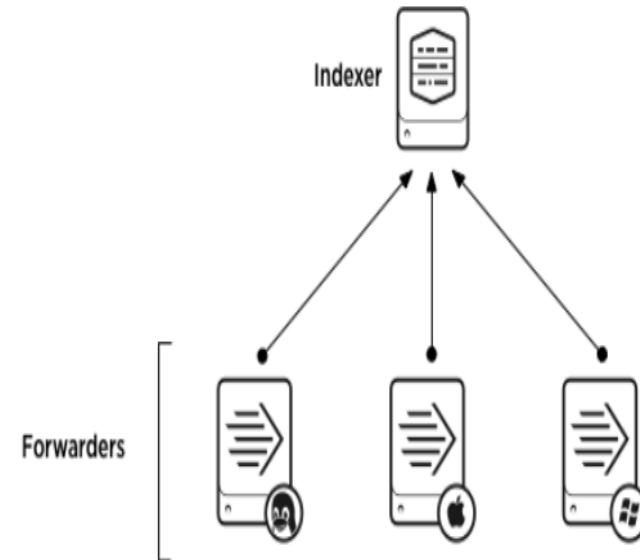
---

**Universal forwarders** stream data from your machine to a data receiver. This receiver is usually a **Splunk index** where you store your Splunk data.

Universal forwarder streaming lets you monitor data in real time.

The universal forwarder also ensures that your data is correctly formatted before sending it to Splunk.

You can also manipulate your data before it reaches the indexes or manually add the data.



# Benefit of Universal Forwarder

---

**Universal forwarders** are highly scalable. Universal Forwarders use significantly less hardware resources than other Splunk products. You can install thousands of them without impacting network performance and cost. The universal forwarder does not have a user interface, which helps minimize resource use.

Forwarders provide the following capabilities:

- metadata tagging, including source, source type, and host.
- configurable buffering
- data compression
- SSL security
- Use of any available network ports

## Computer Hardware Prerequisites

The universal forwarder has the following minimum processing, RAM, and disk space requirements:

Processing	1.5Ghz
RAM	512MB
Free Disk Space	5GB

see [Supported Operating Systems](#) in the Splunk Enterprise installation manual

# Universal Forwarder : Installation

Go to the link [https://www.splunk.com/en\\_us/download/universal-forwarder.html?locale=en\\_us](https://www.splunk.com/en_us/download/universal-forwarder.html?locale=en_us)

## Splunk Universal Forwarder 9.1.0.1

Universal Forwarders provide reliable, secure data collection from remote sources and forward that data into Splunk software for indexing and consolidation. They can scale to tens of thousands of remote systems, collecting terabytes of data.

### Choose Your Installation Package



Windows



Linux



Mac OS



Free BSD



Solaris



AIX

ARM

4.14+, 5.4+ kernel Linux distributions with  
libc v2.21+, Graviton+ Servers 64-bit

.tgz

29.72 MB

[Download Now](#)

.rpm

29.7 MB

[Download Now](#)

.deb

20.18 MB

[Download Now](#)

PPCLE

3.x+, 4.x+, or 5.x+ kernel Linux  
distributions

.rpm

28.68 MB

[Download Now](#)

.tgz

28.63 MB

[Download Now](#)

# Universal Forwarder : Installation

---

Once downloaded , unzip it and install it using below method.

**Install:** ./splunkforwarder/bin/splunk start

**Check status :** ./splunkforwarder/bin/splunk status

**Files being monitored:** ./splunkforwarder/bin/splunk list monitor

by default only splunk logs are being monitored.

**Configuration Files:** splunkforwarder/etc/

**log :** splunkforwarder/var/log/splunk/splunkd.log

# Universal Forwarder : Configuration

## 1. First of all we need to use a port on Splunk Server which will receive the data.

Navigate to **settings → data → forwarding and receiving → add receive data port** (any port)

## 2. We need to tell forwarder whom to connect, so add a forward-server

```
./splunkforwarder/bin/splunk add forward-server SPLUNK-IP-ADDR:RECEIVING-PORT
```

check the status:

```
./splunkforwarder/bin/splunk list forward-server
```

Information get stored in this file (`splunkforwarder/etc/system/local/outputs.conf`)

## 3. Ask forwarder to monitor a directory.

```
splunkforwarder/bin/splunk add monitor /var/log
```

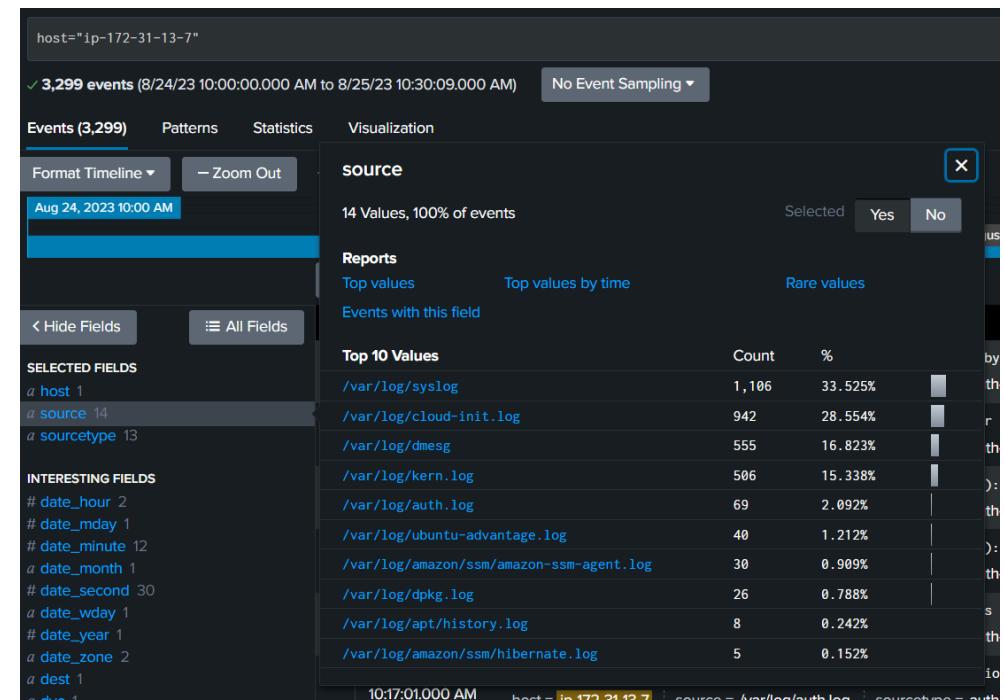
Will store the information here :

`/splunkforwarder/etc/apps/search/local/inputs.conf`

## 4. check the status again

```
splunkforwarder/bin/splunk list monitor
```

## 5. Go to the Splunk server and check the events.



# Universal Forwarder : Configuration

---

## **1. To remove the forward-server from the client machine**

```
./bin/splunk remove forward-server 54.153.52.90:9997
```

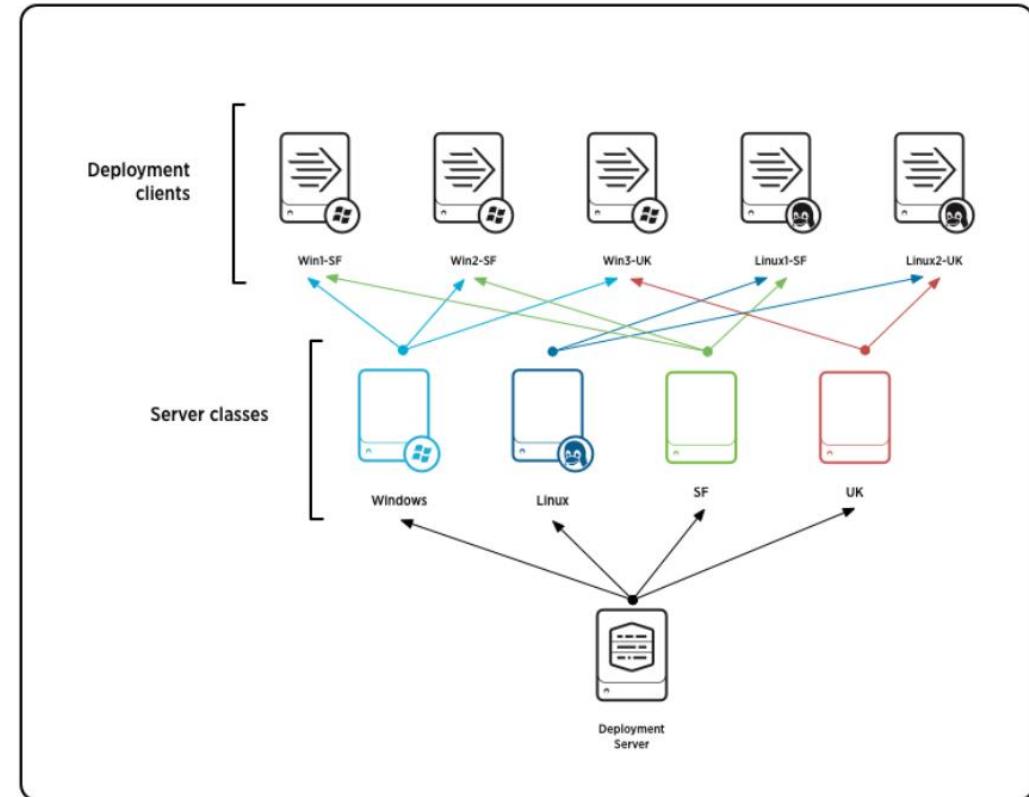
## **2. To remove the monitor list**

```
./bin/splunk remove monitor /var/log
```

# Overview of Deployment Server

**Deployment server** is a tool for distributing **configuration, apps and content updates** to group of Splunk enterprise instances like **universal forwarders, search heads** and others, grouped into **server classes**.

Forwarder management is a GUI build on top of deployment server that provides an easy way to configure the deployment server and monitor the status of deployment updates.



# Overview of Deployment Server

Term	Meaning
<b>deployment server</b>	A Splunk Enterprise instance that acts as a centralized configuration manager. It deploys configuration updates to other instances. Also refers to the overall configuration update facility comprising deployment server, clients, and apps.
<b>deployment client</b>	A remotely configured Splunk Enterprise instance. It receives updates from the deployment server. Deployment clients can be universal forwarders, heavy forwarders, indexers, or search heads.
<b>server class</b>	A deployment configuration category shared by a group of deployment clients. A deployment client can belong to multiple server classes. By creating a server class, you are telling the deployment server that a specific set of clients should receive configuration updates in the form of a specific set of apps.
<b>deployment app</b>	A unit of content deployed to the members of one or more server classes. A deployment app might consist of just a single configuration file, or it can consist of many files. Over time, an app can be updated with new content and then redeployed to its designated clients

# How Deployment Updates happen?

---

The deployment update process works like this:

- 1.** Each deployment client periodically polls the deployment server, identifying itself.
- 2.** The deployment server determines the set of deployment apps for the client, based on which server classes the client belongs to.
- 3.** The deployment server gives the client the list of apps that belong to it, along with those apps' current checksums.
- 4.** The client compares the app info from the deployment server with its own app info, to determine whether there are any new or changed apps that it needs to download.
- 5.** If there are new or updated apps, the deployment client downloads them.
- 6.** Depending on the configuration for a given app, the client might restart itself before the app changes take effect.

# Forwarder Management

---

Whenever we install an universal forwarder, we configure it with **add monitor** and **add forward-server** commands, which is not a scalable solution as we can have many servers.

To Automate the process, we can user **Forwarder management**.

We can also configure Forwarder management to monitor the specific directory based on their IP network or Hostname etc.

All the forwarder are connected to forwarder management.

So we define rules in Forwarder Management, based on that rule it will instruct the forwarder which log file to monitor.

**The forwarder management interface is an interactive, visual tool for creating server classes, which map deployment clients to deployment apps. You can also use forwarder management to manage and monitor your deployment.**

# Forwarder Management

---

The interface saves server class configurations to a **serverclass.conf** file, located under **\$SPLUNK\_HOME/etc/system/local** on the deployment server.

The main purpose of the forwarder management interface is to create and edit server classes. You can also use it for a number of other purposes:

- To track the status of the system
- To monitor deployment activity
- To view the associations between apps, clients, and server classes
- To configure app behavior
- To uninstall apps from clients

# Forwarder Management

---

To enable forwarder management UI, use the CLI and paste an app to deployment-apps folder.

```
cp -R splunk/etc/apps/sample_app/ splunk/etc/deployment-apps/
```

Restart our Splunk Server

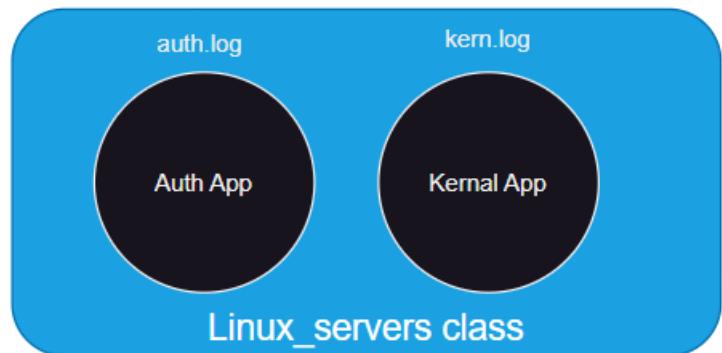
```
./splunk/bin/splunk restart
```

Now navigate to settings → Forwarder management

Our Forwarder management UI is ready.

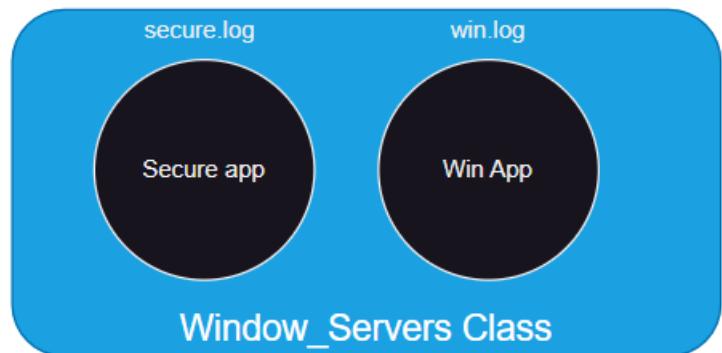
# Server Class and Deployment Apps

---



## Rules

- Deploy to Linux machines
- With IP starting with 192.168
- Exclude 192.168.1.10



## Rules

- Deploy to Windows servers
- With IP starting with 172.12.

# Server Class and Deployment Apps

Lets create a server class:

Navigate to **settings** → **forwarder management** → **server classes** → **add new**

**Add clients:**

Mention the machines to monitor based on IP address, hostname etc

**Server Class: linux\_server**

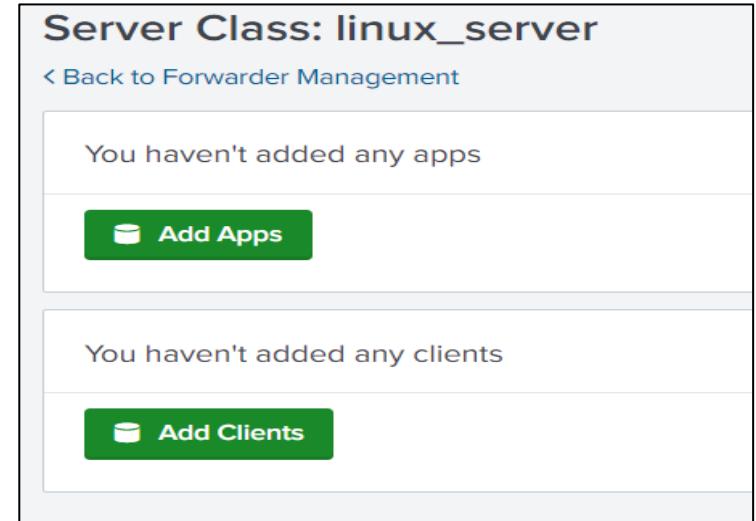
[Back to Forwarder Management](#)

You haven't added any apps

[Add Apps](#)

You haven't added any clients

[Add Clients](#)



**Edit Clients**

Server Class: linux\_server

[Documentation](#)

**Include (includelist)**

Can be client name, host name, IP address, or DNS name.  
Examples: 185.2.3.\* , fwdr-\*  
[Learn more](#)

**Exclude (excludelist)**

Can be client name, host name, IP address, or DNS name.  
Examples: ronnie, rarity  
[Learn more](#)

**Filter by Machine Type (machineTypesFilter)**

Optional

[+](#)

**Cancel** **Preview** **Save**

All Matched Unmatched filter



# Server Class and Deployment Apps

---

## Add Apps:

- We can add multiple apps which will be a part of this server class.

Now we need to use below command on the client machine having forwarder installed to connect with Splunk server.

**./splunkforwarder/bin/splunk set deploy-poll [SPLUNK-IP]:8089**

Restart our forwarder client

**./splunkforwarder/bin/splunk restart**

Now check **splunkforwarder/etc/apps/** which will show that sample-app has been configured on this server too.

# Forwarder UI

## Forwarder Management

Repository Location: \$SPLUNK\_HOME/etc/deployment-apps

**1 Client**  
PHONED HOME IN THE LAST 24 HOURS

**0 Clients**  
DEPLOYMENT ERRORS

**1 Total download**  
IN THE LAST 1 HOUR

[Documentation](#)

Host Name	Client Name	Instance Name	IP Address	Actions	Machine Type	Deployed Apps	Phone Home
ip-172-31-13-7	C16F91CE-A8D6-4D22-9F95-21D13EF82050	ip-172-31-13-7	54.153.59.18	Delete Record	linux-x86_64	1 deployed	▲ 5 minutes ago
Apps ..... sample_app Server Classes ..... linux_server							

# Create Deployment App

---

A [deployment app](#) consists of any arbitrary content that you want to download to a set of [deployment clients](#). The content can include:

- A Splunk Enterprise app (such as those on [Splunkbase](#))
- A set of Splunk Enterprise configurations
- Other content, such as scripts, images, and supporting files

You add a deployment app by creating a directory for it on the deployment server. Once you create the directory, you can use the forwarder management interface to map the app to deployment clients

# Create Deployment App

---

To push log files, we have to create a deployment app with relevant **inputs.conf** and **outputs.conf**.

Go to Splunk server Deployment-apps directory

```
cd splunk/etc/deployment-apps/
```

```
mkdir linux_servers
```

```
cd linux_servers
```

```
mkdir local
```

```
cd local
```

```
touch inputs.conf outputs.conf
```

It contains the information which log files universal forwarder should monitor

Inputs.conf

It contains the information where should these log files forwarded to.

outputs.conf

Deployment App

# Create Deployment App

---

Inputs.conf	outputs.conf
[monitor:///var/log] disabled = false <b>Index = new-index</b>	[tcpout] defaultGroup = default-autolb-group  [tcpout:default-autolb-group] server = 54.153.77.95:9997  [tcpout-server://54.153.77.95:9997]

**Note:** Replace the IP with your Splunk Server IP address and 9997 we have set a receiving port on Splunk server. The data will be stored in new-index.

# Create Deployment App

Navigate to **forwarder management UI → apps → linux\_servers (edit)**

### Edit App: linux\_servers

Server Classes	After Installation
<input type="text" value="linux_server"/> <span>x</span> <span>+</span>	<input checked="" type="checkbox"/> Enable App <input checked="" type="checkbox"/> Restart Splunkd

# Create Deployment App

---

Now the app has been installed in our client machine too

**Forwarder Management**

Repository Location: \$SPLUNK\_HOME/etc/deployment-apps

**1 Client**  
PHONED HOME IN THE LAST 24 HOURS

**0 Clients**  
DEPLOYMENT ERRORS

**1 Total download**  
IN THE LAST 1 HOUR

Apps (2) Server Classes (1) **Clients (1)**

Phone Home: All ▾ All Clients ▾ filter

1 Clients 10 Per Page ▾

i	Host Name	Client Name	Instance Name	IP Address	Actions	Machine Type	Deployed Apps	Phone Home
>	forwarder	C16F91CE-A8D6-4D22-9F95-21D13EF82050	ip-172-31-13-7	54.183.60.48	Delete Record	linux-x86_64	1 deployed	a few seconds ago

# Basics of Regular Expressions

---

Regular expression (regex) is a sequence of characters that defines a search pattern.

“There is a Rainbow which arises on the south shore of Mumbai”

Rainbow – Literal Character

Meta Character is a character or sequence of character that has special meaning that provides information about the other characters.

# Meta Characters

---

\d – Any digit from 0-9

\w – Any word (A-Z, a-z, 0-9)

\s – Whitespace

. - Any character

[] - Matches characters in brackets

[^] - Matches characters in brackets

# Parsing Web Server Logs

---

There are two ways to have data parsed in Splunk.

1. Create an addon and write custom regex
2. Use Add-ons from marketplace which has build in regex for specific log

# Named Capturing Group

---

Named Capturing group makes understanding the parsed data in much more easier manner.

It is used extensively in various Splunk Add-ons available in the marketplace.

Sample Syntax:

(?<name>regex)

# Importance of Source Types

---

In Splunk, field extractions and regex are generally defined at the source types level.

They can be defined in **props.conf** as well as **transforms.conf**

If source type of your log is incorrect then it will not get parsed properly.

Splunk comes with some built-in source types and it is associated regex for common logs.

# Interactive Field Extractor (IFX)

Interactive Field Extractor allows us to teach splunk on how to extract fields from your data without writing regex.

Navigate to **Extract New Fields** if we want more fields as per our requirements.

1. Select a Sample code
2. Select a method (regular Expression or Delimiters)
3. Rename fields and save.

The screenshot shows the Splunk IFX interface with the following sections:

- Selected Fields:** host 1, source 1, sourcetype 1
- Interesting Fields:** date\_hour 2, date\_mday 1, date\_minute 5, date\_month 1, date\_second 8, date\_wday 1, date\_year 1, date\_zone 1, eventtype 6, index 1, linecount 1, punct 28, splunk\_server 1, timeendpos 1, timestamppos 1
- Buttons:** Hide Fields, All Fields, + Extract New Fields (highlighted with a yellow box)
- Text at bottom:** 13 more fields

# Creating Custom Source Types

Splunk comes with default source types and field extractions for common log files.

However we can create our own custom source type as well

Every source type has some associated configuration settings

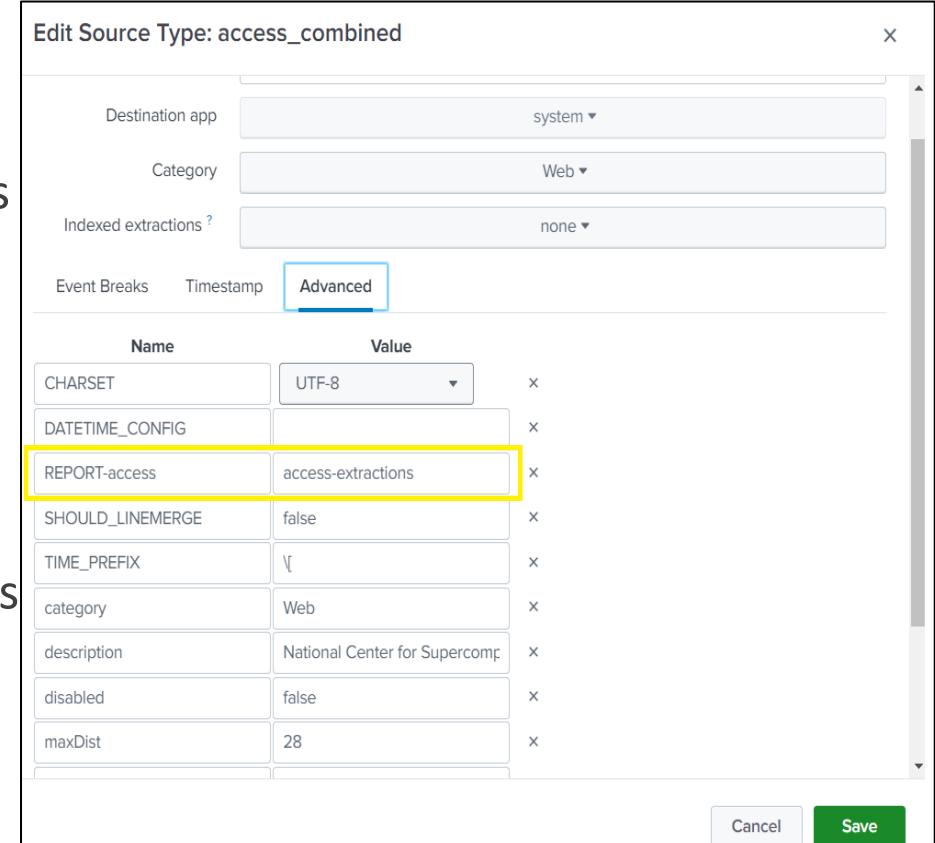
These configuration parameters and source type details are stored in props and transforms.

We can find all the source types under **Settings → source types**.

Edit Source Type: access\_combined

Name	Value
CHARSET	UTF-8
DATETIME_CONFIG	
REPORT-access	access-extractions
SHOULD_LINEMERGE	false
TIME_PREFIX	\
category	Web
description	National Center for Supercomputing
disabled	false
maxDist	28

Cancel Save



# Creating Custom Source Types

---

Default configuration of **props** and **transforms** file we can find under **/splunk/etc/system/default** directory.

Like **access\_combined** source type setting details is present under **props.conf** file.

**REPORT-access** is a method which is using a global parameter **access-extractions**

**REPORT-access = access-extractions**

Now **access-extractions** is defined under **transforms.conf** file and below regex belongs to it which helps **access\_combined** source type to name all the fields included in the data.

```
[access-extractions]
# matches access-common or access-combined apache logging formats
# Extracts: clientip, clientport, ident, user, req_time, method, uri, root, file, uri_domain, uri_query, version, status, bytes, referer_url, r
eferer_domain, referer_proto, useragent, cookie, other (remaining chars)
# Note: referer is misspelled in purpose because that is the "official" spelling for "HTTP referer"
REGEX = ^[[nspaces:clientip]]\s++[[nspaces:ident]]\s++[[nspaces:user]]\s++[[sbstring:req_time]]\s++[[access-request]]\s++[[nspaces:status]]\s++
[[nspaces:bytes]](?:\s++(<referer>[[bc_domain:referer_]]?+[^"]*+)"(?:\s++[[qstring:useragent]](?:\s++[[qstring:cookie]])?+)?+)?[[all:other]]
```

# LAB

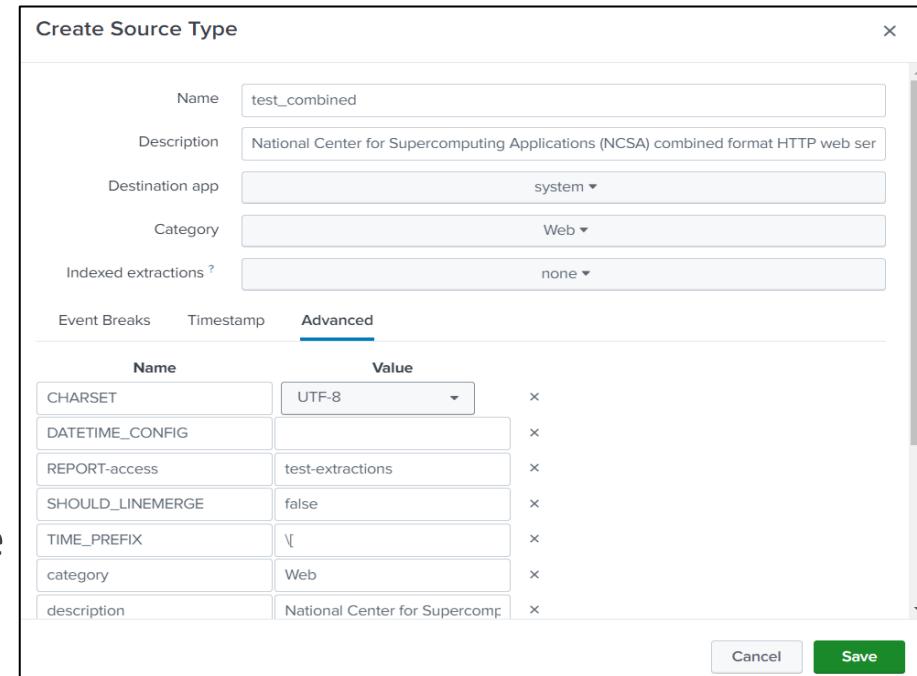
---

## Create Custom Source

1. Create or clone a source type
2. update the function setting (example REPORT-access = test-extractions)
3. Upload the data
4. Check if the data is formatted properly if not then create test-extractions inside transforms.conf under local directory (since this is a custom source type).
5. Search the result and check if the data is formatted again.

# Solution

1. Create or clone a source type (test\_combined)
2. Update the function (REPORT\_access=test-extractions)
3. Add a new data to test.
4. New props.conf will be created under **/splunk/etc/system/local** directory
5. Create a **transforms.conf** under directory and paste the regex under test-extractions.



```
[test-extractions]
# matches access-common or access-combined apache logging formats
# Extracts: clientip, clientport, ident, user, req_time, method, uri, root, file, uri_domain, uri_query, version, status, bytes, referer_url, referer_domain, referer_proto, useragent, cookie, other (remaining chars)
# Note: referer is misspelled in purpose because that is the "official" spelling for "HTTP referer"
REGEX = ^[[nspaces:clientip]]\s+[[nspaces:ident]]\s+[[nspaces:user]]\s+[[sbstring:req_time]]\s+[[access-request]]\s+[[nspaces:status]]\s+[[nspaces:bytes]](?:\s+"(?<referer>[[bc_domain:referer_]]?+[^"]*+)"(?:\s+[[qstring:useragent]](?:\s+[[qstring:cookie]])?+)?+)?+)?+[[all:other]]
~
```

# Event Types

---

EventTypes are categorization system to help you make sense of your data.

For example:

**Sourcetype=acess\_combined status=200 action=purchase**

If you save the above as an eventtype **success\_purchase**, any event that gets returned by the search gets associated eventtype.

Default event types we can find under **settings → eventtypes** and we can create our own which will be added there too.

# Event Types: Limitations

---

We can not have event type based on search which has following aspects:

1. Includes pipe operator after a simple search
2. Includes a sub-search

# Lab

---

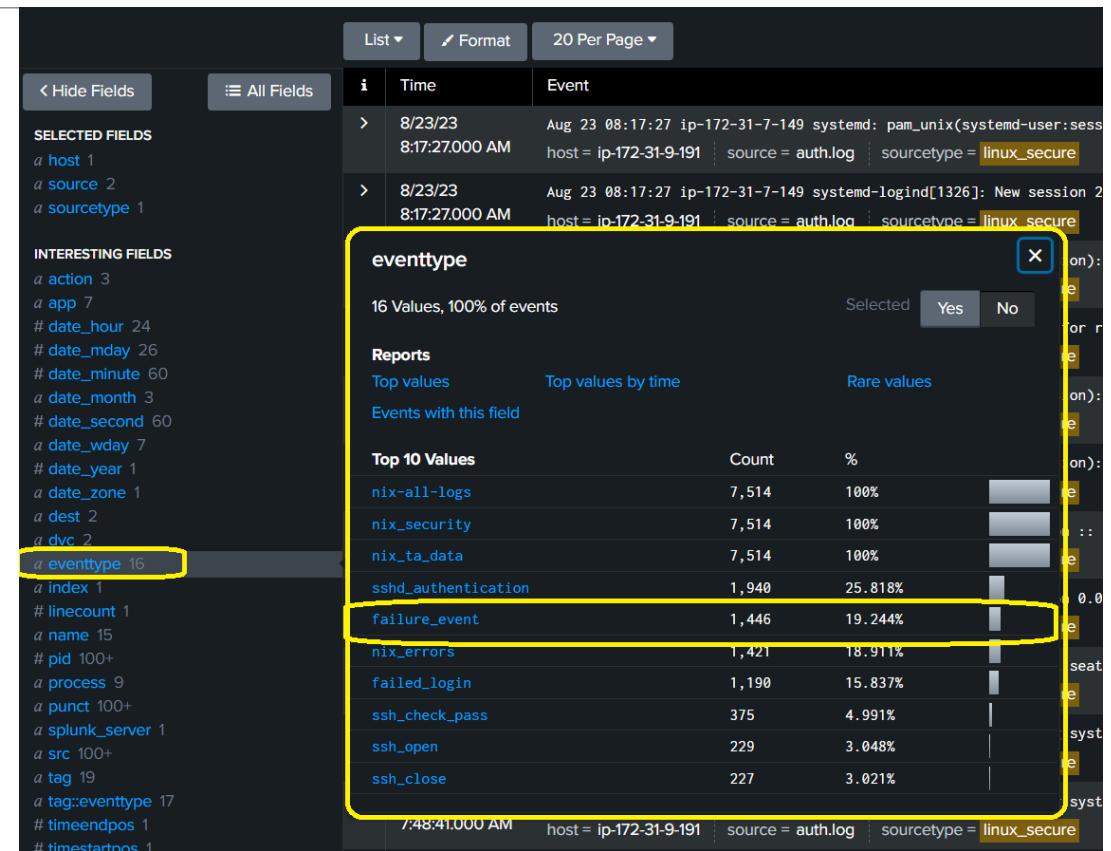
1. Add a data
2. Create an eventtype using the below query.

```
sourcetype="linux_secure" action="failure »
```

3. save it as failure\_events
4. check the status

# Lab

A new event type will be created under eventtype field.



# Tags

Tags enable you to assign names to specific field and value combinations, including event type, host, source, or source type.

By default Splunk provides the tags to our search events. You can use tags to help you track abstract field values, like IP addresses or ID numbers.

You can use a tag to group a set of field values together, so that you can search for them with one command.

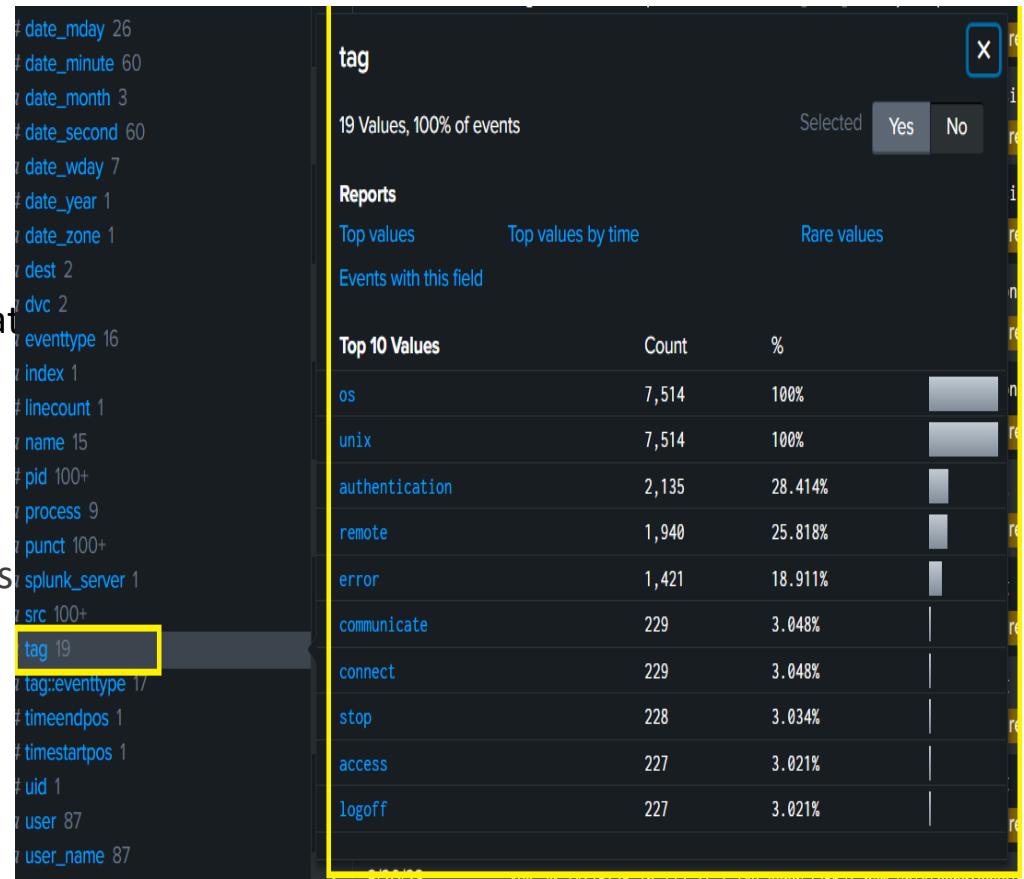
Example Use Case :

Your Network logs has IP addresses belonging to three subnets

192.168.1.0/24 Tag: Singapore region

172.17.0.0/16 Tag: Mumbai region

10.66.0.0/16 Tag: Frankfurt region



# Tags: Creation

---

1. Navigate to settings → eventtypes
2. Select an eventtype and add the tags to it. Multiple tags should be comma-separated.
3. save it
4. Search the events related to that tag. (like below search the event using the search **tag=failure-action**)

The screenshot shows a search interface with the following fields:

- Search string \***: sourcetype="linux\_secure" action="failure"
- Tag(s)**: NotAuthenticated, failure-action (highlighted with a blue border)
- Color**: none
- Priority**: 1 (Highest)

At the bottom right are two buttons: **Cancel** and **Save**.

# Event types priority and Coloring

Typically event type field get attached to the matching events when wildcard search is used.

Coloring is based on eventtypes so we need to search the relevant events and assign the color to those events as per our requirement.

## Steps:

1. Search for the events which have status=200 and save it as success-events and apply the color
2. Search for the events which have status=404 and save it as failure-events and apply the color

Save As Event Type

Name	success_events
Tags	Optional
Color	green ▾
Priority	1 (Highest) ▾

Determines which style wins, when an event has more than one event type.

Cancel Save

Save As Event Type

Name	failure-events
Tags	Optional
Color	red ▾
Priority	1 (Highest) ▾

Determines which style wins, when an event has more than one event type.

Cancel Save

# Event types priority and Coloring

---

Make sure you do not have the overlapping eventtypes, otherwise the color will not be applied to it.

>	8/23/23 8:18:45.000 AM	122.162.148.139 - - [23/Aug/2023:08:18:45 +0000] "GET / HTTP/1.1" 200 3459 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/116.0.0.0 Safari/537.36" host = ip-172-31-9-191   source = access.log   sourcetype = test_combined
>	8/23/23 8:18:44.000 AM	122.162.148.139 - - [23/Aug/2023:08:18:44 +0000] "GET / HTTP/1.1" 200 3459 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/116.0.0.0 Safari/537.36" host = ip-172-31-9-191   source = access.log   sourcetype = access_combined
>	8/23/23 8:18:44.000 AM	122.162.148.139 - - [23/Aug/2023:08:18:44 +0000] "GET / HTTP/1.1" 200 3459 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/116.0.0.0 Safari/537.36" host = ip-172-31-9-191   source = access.log   sourcetype = test_combined
>	8/23/23 8:18:43.000 AM	122.162.148.139 - - [23/Aug/2023:08:18:43 +0000] "GET /favicon.ico HTTP/1.1" 404 492 "http://18.144.177.168/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/116.0.0.0 Safari/537.36" host = ip-172-31-9-191   source = access.log   sourcetype = access_combined
>	8/23/23 8:18:43.000 AM	122.162.148.139 - - [23/Aug/2023:08:18:43 +0000] "GET /favicon.ico HTTP/1.1" 404 492 "http://18.144.177.168/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/116.0.0.0 Safari/537.36" host = ip-172-31-9-191   source = access.log   sourcetype = test_combined
>	8/23/23 8:18:42.000 AM	122.162.148.139 - - [23/Aug/2023:08:18:42 +0000] "GET /icons/ubuntu-logo.png HTTP/1.1" 200 3607 "http://18.144.177.168/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/116.0.0.0 Safari/537.36" host = ip-172-31-9-191   source = access.log   sourcetype = access_combined

# Macros

# Macros

---

**Search macros** are reusable chunks of Search Processing Language (SPL) that you can insert into other searches.

Search macros can be any part of a search, such as an eval statement or search term and do not need to be a complete command.

Macros are used heavily in **Splunk Enterprise Security, Splunk IT service Intelligence and Splunk common information model (CIM)**.

Instead of using the search query over and over again, we can create Macros and that Macro will be called by different searches or different dashboards.

Macros are called using back tick character ( ` )

```
sourcetype=access_* | `mymacro`
```

# Macros

---

If macro expands to an SPL string that begins with a Generating command like **from**, **search**, **metadata**, **inputlookup**, **pivot**, and **tstats**. You need to put a pipe character before the search macro.

Macros can use the variable as well.

Your search macro definition can include the following:

- A validation expression that determines whether the arguments you enter are valid.
- A validation error message that appears when you provide invalid arguments.

A macro can inherit other macro too.

# Create Macro

---

## Steps

- Select **Settings > Advanced Search > Search macros.**
- Click **New** to create a search macro. Enter a unique **Name** for the search macro(Mention the number of arguments if used).
- In **Definition**, enter the search string that the macro expands to when you reference it in another search.
- (Optional) Click **Use eval-based definition?** to indicate that the **Definition** value is an eval expression that returns a string that the search macro expands to.
- (Optional) Enter any **Arguments** for your search macro. This is a comma-delimited string of argument names. Argument names may only contain alphanumeric characters (a-Z, A-Z, 0-9), underscores, and dashes. The string cannot contain repetitions of argument names.
- (Optional) Enter a **Validation** expression that verifies whether the argument values are acceptable.
- (Optional) Enter a **Validation error message** if you defined a validation expression. This message appears when the argument values that invoke the search macro fail the validation expression.
- Click **Save** to save your search macro.

# Splunk Lookups

---

Lookups enhances the power of Splunk by enabling correlation of search results with 3<sup>rd</sup> part data like databases, directories, csv files and others.

It allows us to co-relate external information with the search results.

**For example:**

Event in Splunk might contain CustomerID field.

We want to get more information like customerName, customerNumber which are stored externally in files.

# Splunk Lookups: Lab

1. Add a new test data.
2. Rename the fields using IFX

The screenshot shows the Splunk Event Detail view for an event. The event details are as follows:

Time	Event
8/28/23 10:37:21.000 AM	1050,Widget B,75.00

The event actions pane shows field renamings:

Type	Field	Value	Actions
Selected	host	ip-172-31-9-191	▼
	source	purchase.txt	▼
	sourcetype	purchase	▼
Event	customerID	1050	▼
	itemPrice	75.00	▼
	itemPurchased	Widget B	▼
	timestamp	none	▼
Time	_time	2023-08-28T10:37:21.000+00:00	▼
Default	index	main	▼
	linecount	1	▼
	punct	---	▼
	splunk_server	ip-172-31-9-191	▼

A yellow box highlights the renamed fields under the "Event" type: customerID, itemPrice, and itemPurchased.

# Splunk Lookup

---

## 3. Create a lookup.

    Navigate to settings → lookups

## 4. Upload the csv file having Customer information

which will be stored at a specific

**location/root/splunk/etc/users/admin/search/lookups/purchase.csv**

The screenshot shows the 'Create New Lookup' dialog box. At the top, the 'Destination app' dropdown is set to 'search'. Below it, the 'Upload a lookup file' section contains a 'Choose File' button with the path 'purchase.csv' displayed. A note below the button specifies that the file can be a plaintext CSV, a gzipped CSV, or a KMZ/KML file, with a maximum size of 500MB. The 'Destination filename \*' field is filled with 'purchase.csv'. A detailed note below this field explains the naming conventions for different file types. At the bottom right are two buttons: 'Cancel' and a green 'Save' button.

Destination app	search
Upload a lookup file	<input type="button" value="Choose File"/> purchase.csv Select either a plaintext CSV file, a gzipped CSV file, or a KMZ/KML file. The maximum file size that can be uploaded through the browser is 500MB.
Destination filename *	purchase.csv Enter the name this lookup table file will have on the Splunk server. If you are uploading a gzipped CSV file, enter a filename ending in ".gz". If you are uploading a plaintext CSV file, we recommend a filename ending in ".csv". For a KMZ/KML file, we recommend a filename ending in ".kmz"/".kml".
<input type="button" value="Cancel"/> <input type="button" value="Save"/>	

# Splunk Lookup

5. Search the details and map the customerID of the event with CustomerID of purchase.csv file.

The screenshot shows a Splunk search interface with the following details:

- Search Bar:** source="purchase.txt" | lookup purchase.csv customerID as customerID
- Results Summary:** ✓ 51 events (8/27/23 10:00:00.000 AM to 8/28/23 10:57:07.000 AM) | No Event Sampling
- Event List:** Events (51) | Patterns | Statistics | Visualization | Format Timeline | Zoom Out | + Zoom to Selection | Deselect
- Event View:** A single event is selected, showing:
  - Time: 8/28/23 10:37:21.000 AM
  - Event: 1050,Widget B,75.00
  - Event Actions dropdown
  - Selected Fields table:

Type	Field	Value	Actions
Selected	host	ip-172-31-9-191	▼
Selected	source	purchase.txt	▼
Selected	sourcetype	purchase	▼
  - Event table:

Type	Field	Value	Actions
Event	customerID	1050	▼
Event	customerLocation	Boston	▼
Event	customerName	Emily White	▼
Event	itemPrice	75.00	▼
Event	itemPurchased	Widget B	▼
Event	timestamp	none	▼
  - Time: \_time | 2023-08-28T10:37:21.000+00:00
  - Default: index | main

# Splunk Alerts

---

Alerts are used to monitor and respond to a particular event.

An alert is a type of saved search. Alerts run in real time or on a scheduled interval and are triggered when they return results that meet user-defined conditions. When an alert is triggered, it can initialize one or more alert actions.

Alerts not necessarily mean to send an email on a specific action, it can do much more better things like event driven action.

It can be connected with a specific scripts which will do the next action.

Throttling of alerts is also an important factor during an outage.

# Splunk Alerts

---

## Alert Types:

- Scheduled
- Real Time

## Trigger Condition:

Trigger alert when : when to trigger



Trigger: Once or each result

Throttle: After an event is triggered, subsequent events will not be triggered until after the throttle period.

<input checked="" type="checkbox"/> <b>Number of Results</b>
Triggers based on a number of search results during a scheduled search.
<b>Number of Hosts</b>
Triggers based on a number of hosts during a scheduled search.
<b>Number of Sources</b>
Triggers based on a number of sources during a scheduled search.
<b>Custom</b>
Triggers based on a custom condition during a scheduled search.

# Splunk Alerts

---

## Trigger Actions:

Sending email notifications when alerts trigger	<a href="#">Email notification action</a>
Displaying a message in a chat room or updating another web resource	<a href="#">Use a webhook alert action</a>
Writing the results of the triggered alert or scheduled report to a CSV lookup file	<a href="#">Output results to a CSV lookup</a>
Logging and indexing searchable alert events	<a href="#">Log events</a>
Adding an alert to a list of recently triggered alerts for monitoring	<a href="#">Monitor triggered alerts</a>
Sending an alert to Splunk Mobile users	<a href="#">Send alerts and dashboards to Splunk Mobile users</a>

# Splunk Alerts: Lab

1. Search for a suspicious activity
2. Set an alert for it (Save as → alert)
3. Name it and choose the appropriate action.
4. Add a new data again for testing the alert.
5. Check if the alert has been triggered.

Save As Alert

Permissions	Private	Shared in App
Alert type	Scheduled	Real-time
Expires	24	hour(s) ▾

**Trigger Conditions**

Trigger alert when  Per-Result ▾

Throttle ?

Suppress results containing field value

Suppress triggering for  60 second(s) ▾

**Trigger Actions**

+ Add Actions ▾

When triggered >  Remove

# Splunk Alert

# **Access Control**

# Why do we need Access Control?

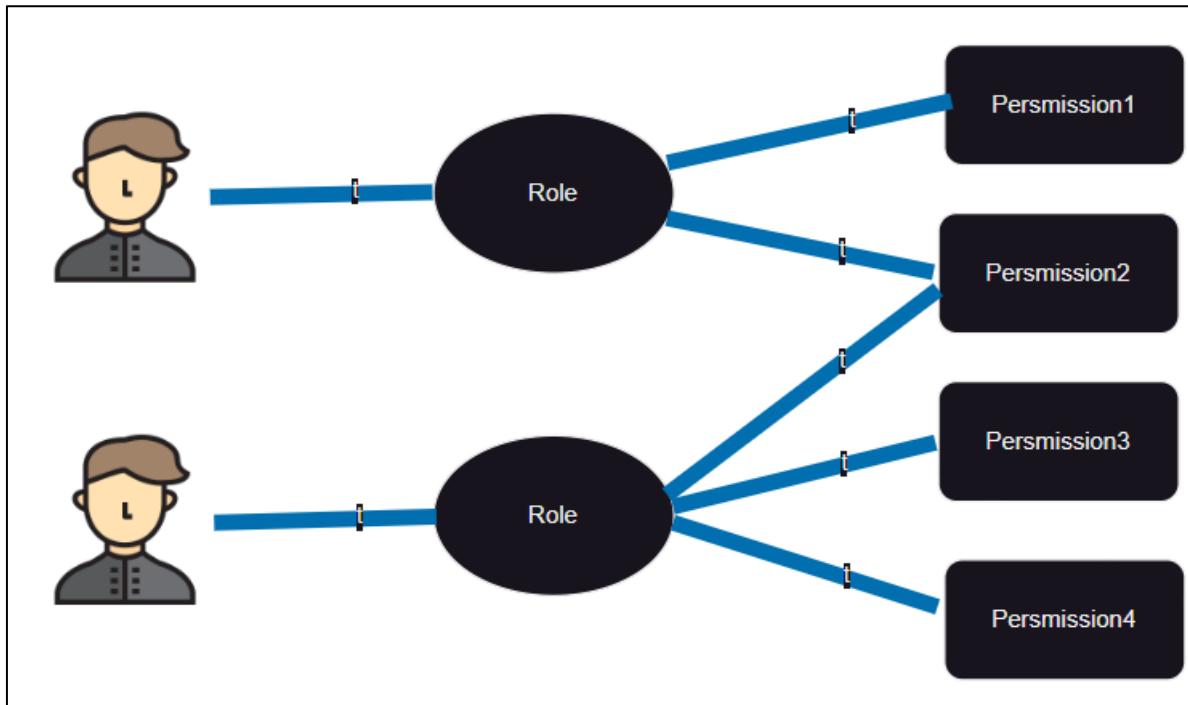
---

1. Extremely sensitive data having access to system might involve legal risk, consider using an isolated separate Splunk instance for relevant audience.
2. When you have sensitive data then you can restrict access based on index to users.
3. When there are security concerns, but not so much legal risks, you can restrict access using apps. For example, you can create an app with static dashboards and assign roles with lower clearance to those dashboards. This limits the type of information that the user that holds the role may access.
4. Field encryption, search exclusions, and field aliasing to redacted data are also great ways to tighten up a limited search case.

# User Authentication

---

Splunk Enterprise Authentication allows us to create users, add them to roles(groups) and assign permissions to those roles.



# Role based user access

---

Roles let users perform actions on the Splunk platform. You can use roles to control access to platform resources. When you configure role-based user access, you determine what [permissions](#) and [capabilities](#) that users have through the roles that they hold.

As users do not receive permissions and capabilities directly, roles connect users to how they interact with the Splunk platform.

The Splunk platform comes with the following predefined roles:

**admin:** This role is for administrators who manage all or most of the users, objects, and configurations. and comes with the most capabilities.

**power:** This role can edit all shared objects (saved searches, etc) and alerts, tag events, and perform other similar tasks.

**user:** This role can create, edit, and run its own searches, save those searches, edit its own preferences, create and edit event types, and perform other similar tasks.

**can\_delete:** This role lets the user delete by keyword. This capability is necessary when using the delete search operator.

**Splunk-system-role:** This role lets users create other users and roles, but does not grant any other administrative capabilities.

# Role based user access

---

Custom roles let you make granular adjustments to user access, including the following:

**Role inheritance:** You can have a role inherit certain properties from one or more existing roles. Users that hold multiple roles receive the most permissive amount of capabilities that each role that they hold can provide.

**Capabilities:** You can specify which actions that a user that holds the role can perform, for example, change their password, change forwarder settings, and so on. See [About defining roles with capabilities](#) for more information.

**Allowed and default indexes:** You can limit access to specific indexes and set which indexes the Splunk platform searches by default.

**Search restrictions:** In addition to specifying the indexes that users that hold the role can search, you can also specify a search filter that limits the search results that these users can see. For additional information, see "Search restrictions" in this topic.

**Resource access:** You can control how many standard and real-time searches that all users that hold the role can run at one time, as well as individual limits for each user. You can restrict searches to a certain time window, and control how much disk space is available for search jobs that a user with this role creates.

# Distributed Splunk Architecture

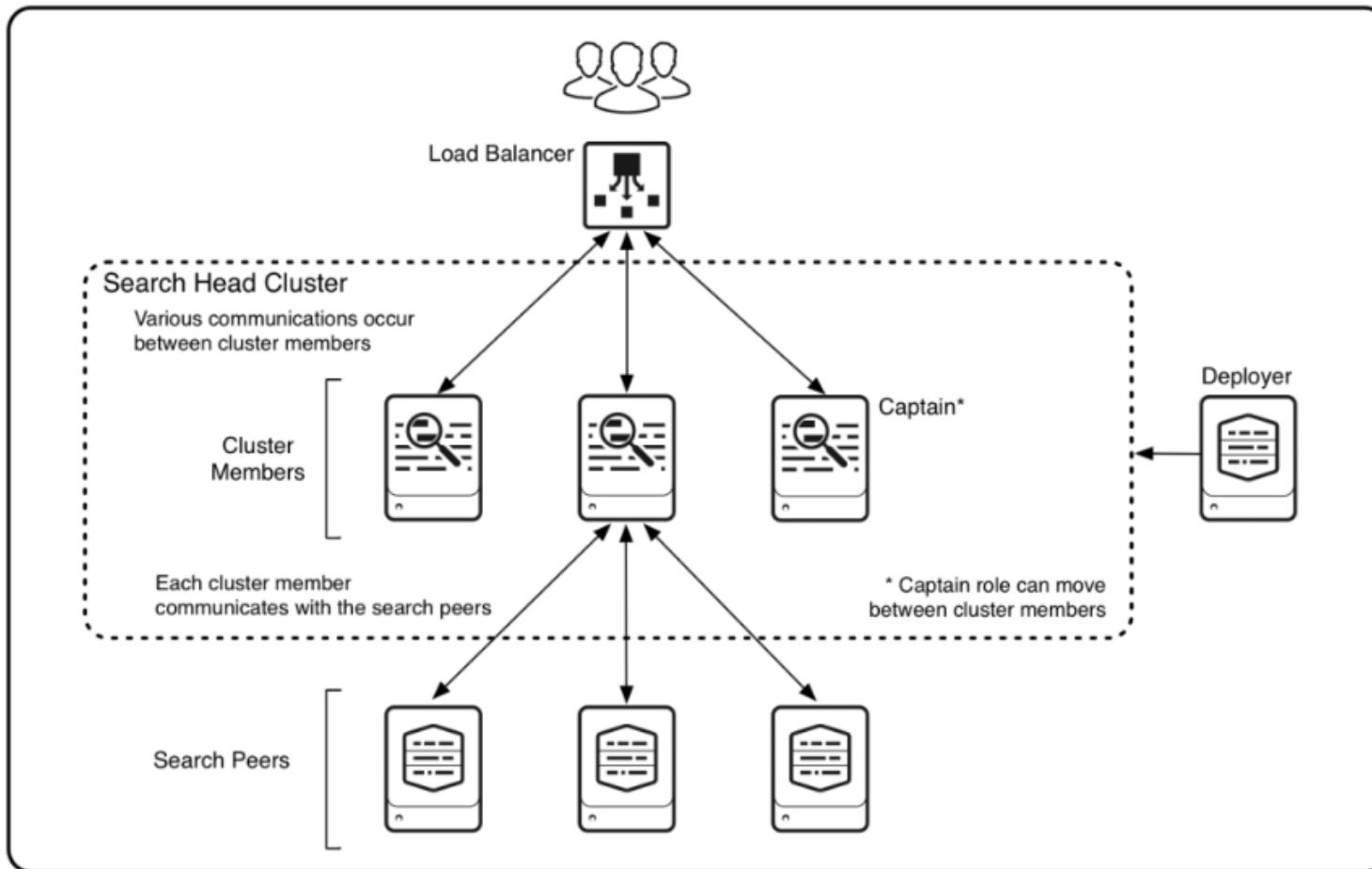
# Splunk Components

---

Splunk enterprise has various other sub-components which are :

- Indexer
- Search Head
- Deployment Server
- Forwarders
- License Master
- Monitoring Console

# Splunk Cluster



# Splunk Cluster

---

Splunk Enterprise supports clustering features for two major components:

- Indexer
- Search Heads

For components like license master, heavy forward etc, Splunk supports Active-Passive failover.

# Splunk Cluster: License Master

---

- Splunk Enterprise ingests external data, indexes it and stores it on disk.
- Licenses specify how much external data you can index per day
- All Splunk Enterprise instances require a license
- If you have a standalone indexer, you can install the license locally
- In case of distributed, we need to configure a license master

# Splunk Cluster: License Master

This server is acting as a standalone license server

[Change to peer](#)

**Trial license group**

This server is configured to use licenses from the **Trial license group**

[Add license](#) [Usage report](#)

**Alerts**

Licensing alerts notify you of excessive indexing warnings and licensing misconfigurations. [Learn more](#)

**Current**

- No licensing alerts

**Permanent**

- No licensing violations

**Local server information**

Indexer name	ip-172-31-9-191
License expiration	Oct 21, 2023, 12:32:04 PM
Licensed daily volume	500 MB
Volume used today	0 MB (0.012% of quota)
Warning count	0
Debug information	<a href="#">All license details</a> <a href="#">All indexer details</a>

# Splunk Cluster: License Master creation

- Make one of the Splunk instance as the license manager.
- Add the license

**Change manager association**  
Licensing > Change manager association

**Change manager association**

This server, ip-172-31-9-191, is currently acting as a manager license server.

Designate this Splunk instance, ip-172-31-9-191, as the manager license server  
Choosing this option will:

- Point the local indexer at the local manager license server
- Disconnect the local indexer from any remote license server

Designate a different Splunk instance as the manager license server  
Choosing this option will:

- Deactivate the local manager license server
- Point the local indexer at license server specified below
- Discontinue license services to remote indexers currently pointing to this server

Manager license server URI  
  
For example: https://splunk\_license\_server:8089  
Use https and specify the management port.



**Licensing**  
This server is acting as a standalone license server [Change to peer](#)

**Trial license group** [Change license group](#)  
This server is configured to use licenses from the Trial license group

[Add license](#) [Usage report](#)

**Alerts**  
Licensing alerts notify you of excessive indexing warnings and licensing misconfigurations. [Learn more](#)

Current: No licensing alerts

Permanent: No licensing violations

**Local server information**

Indexer name	ip-172-31-9-191
License expiration	Oct 21, 2023, 12:32:04 PM
Licensed daily volume	500 MB
Volume used today	0 MB (0.012% of quota)
Warning count	0
Debug information	All license details All indexer details

# Splunk Cluster: License Master

- Start some Splunk instances which would work as slaves.
- Provide the IP address of the license Manager to sync with it.
- The slaves will become a part of this cluster

Change manager association

Licensing > Change manager association

This server, ip-172-31-9-191, is currently acting as a manager license server.

Designate this Splunk instance, ip-172-31-9-191, as the manager license server  
Choosing this option will:

- Point the local indexer at the local manager license server
- Disconnect the local indexer from any remote license server

Designate a different Splunk instance as the manager license server  
Choosing this option will:

- Deactivate the local manager license server
- Point the local indexer at license server specified below
- Discontinue license services to remote indexers currently pointing to this server

Manager license server URI

https://

For example: https://splunk\_license\_server:8089  
Use https and specify the management port.



# Splunk Cluster: License Pool

- Splunk License resides in license stack as Splunk Enterprise Stack.
- The stack has default license pool called `auto_generated_pool_enterprise`
- Any license slave that connects to this license master has access to the default pool.
- Best approach is to create different pool for each team and designate the volume usage.
- One instance can be a part of only one pool.

Licensing

This server is acting as a manager license server [Change to peer](#)

Enterprise license group [Change license group](#)

This server is configured to use licenses from the **Enterprise license group**

[Add license](#) [Usage report](#)

Alerts

This deployment is subject to license enforcement. Search is disabled after 45 warnings over a 60-day window [Learn more](#)

Licensing alerts notify you of excessive indexing warnings and licensing misconfigurations. [Learn more](#)

Current

- No licensing alerts

Permanent

- No licensing violations

Splunk Enterprise License stack [Learn more](#)

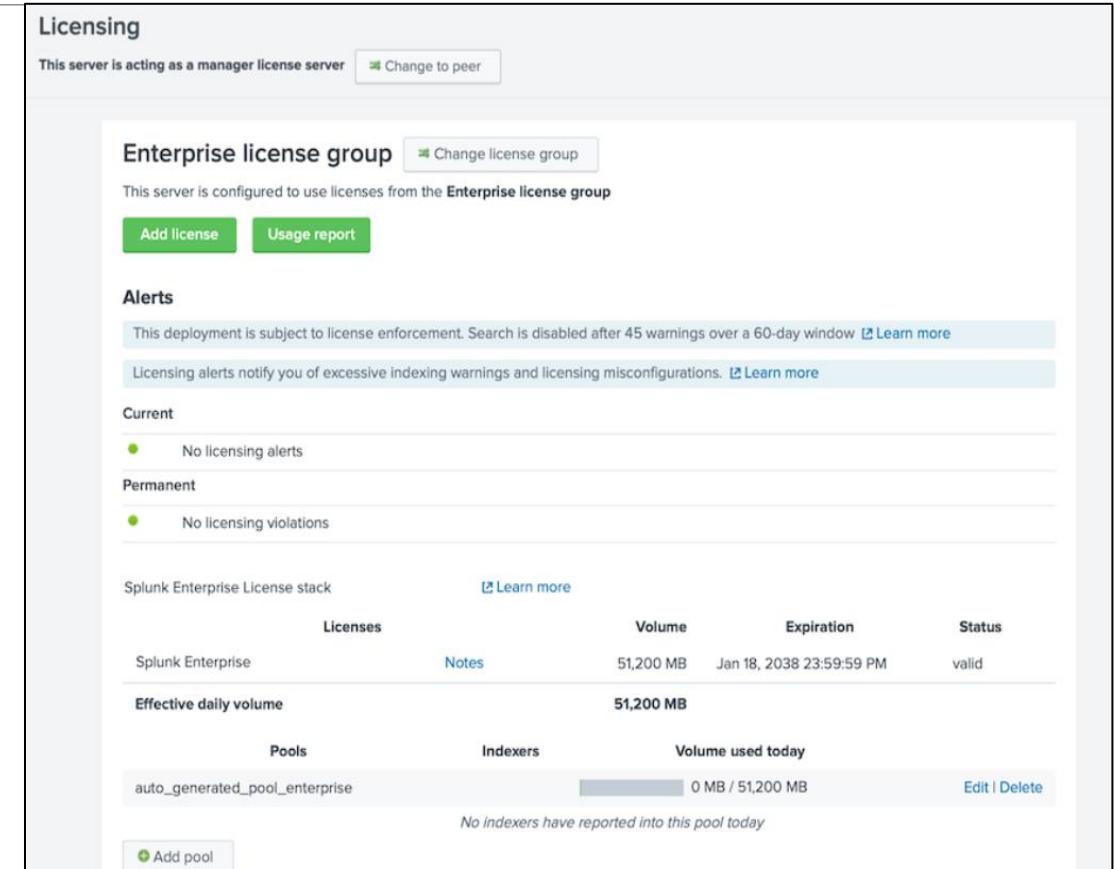
Licenses	Volume	Expiration	Status	
Splunk Enterprise	Notes	51,200 MB	Jan 18, 2038 23:59:59 PM	valid

Effective daily volume **51,200 MB**

Pools	Indexers	Volume used today	Actions
auto_generated_pool_enterprise		0 MB / 51,200 MB	<a href="#">Edit</a>   <a href="#">Delete</a>

No indexers have reported into this pool today

[Add pool](#)

The screenshot shows the Splunk Licensing interface. At the top, it says "Licensing" and "This server is acting as a manager license server". Below that is the "Enterprise license group" section, which states "This server is configured to use licenses from the Enterprise license group". There are two green buttons: "Add license" and "Usage report". Under "Alerts", it says "This deployment is subject to license enforcement. Search is disabled after 45 warnings over a 60-day window" and "Licensing alerts notify you of excessive indexing warnings and licensing misconfigurations". The "Current" and "Permanent" sections both show "No licensing alerts" and "No licensing violations". The "Splunk Enterprise License stack" section shows a single license entry for "Splunk Enterprise" with a volume of 51,200 MB, expiration on Jan 18, 2038, and a status of "valid". Below this is the "Effective daily volume" section showing "51,200 MB". A table then lists "Pools" (auto\_generated\_pool\_enterprise), "Indexers" (empty), and "Volume used today" (0 MB / 51,200 MB). A note at the bottom says "No indexers have reported into this pool today". At the bottom left is a button for "Add pool".

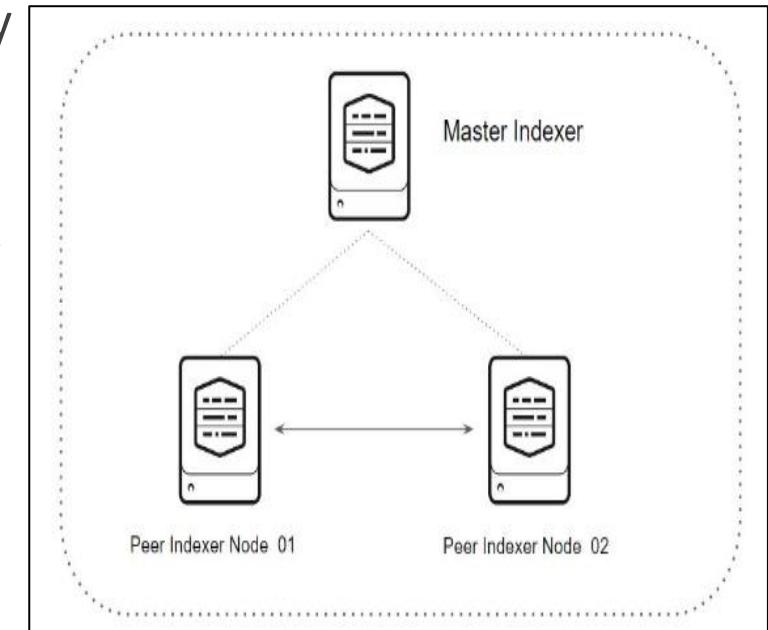
# Splunk Cluster: Indexer

Indexer is a component in Splunk Enterprise whose responsibility is to index data, transform data into searchable events and placing results into an index.

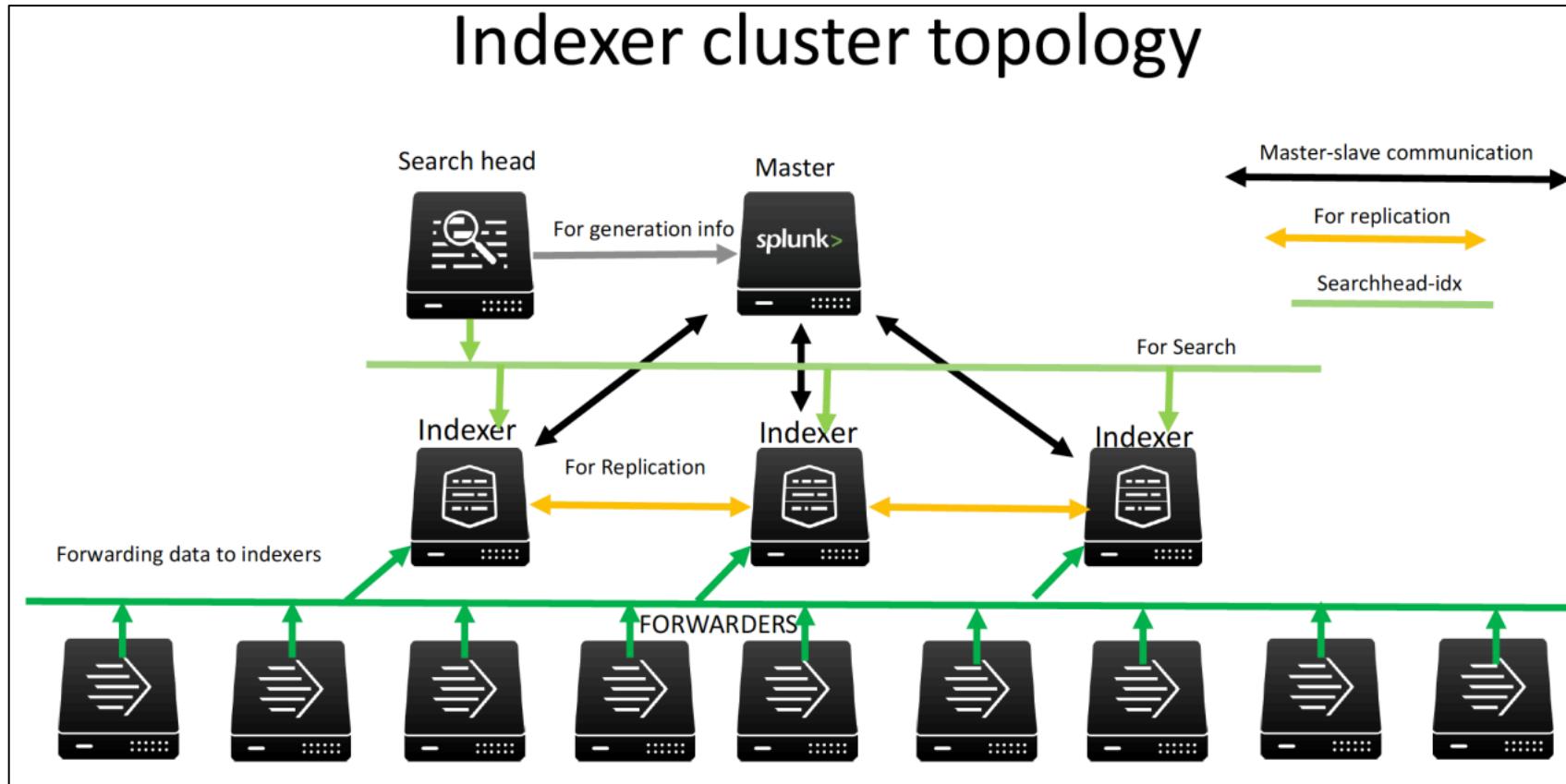
To ensure high availability of data, we can deploy indexer cluster.

Each cluster has three types of nodes:

- A single **manager node** to manage the cluster.
- Several to many **peer nodes** to index and maintain multiple copies of the data and to search the data.
- One or more **search heads** to coordinate searches across the set of peer nodes.



# Splunk Cluster: Indexer Cluster



# Splunk Cluster: Indexer

---

The **manager node** manages the cluster. It coordinates the replicating activities of the peer nodes and tells the search head where to find data. It also helps manage the configuration of peer nodes and orchestrates remedial activities if a peer goes down.

The **peer nodes** receive and index incoming data, just like non-clustered, stand-alone indexers. Unlike stand-alone indexers, however, peer nodes also replicate data from other nodes in the cluster. A peer node can index its own incoming data while simultaneously storing copies of data from other nodes. You must have at least as many peer nodes as the replication factor. That is, to support a replication factor of 3, you need a minimum of three peer nodes.

The **search head** runs searches across the set of peer nodes. You must use a search head to manage searches across indexer clusters.

# Splunk Cluster: Master Indexer

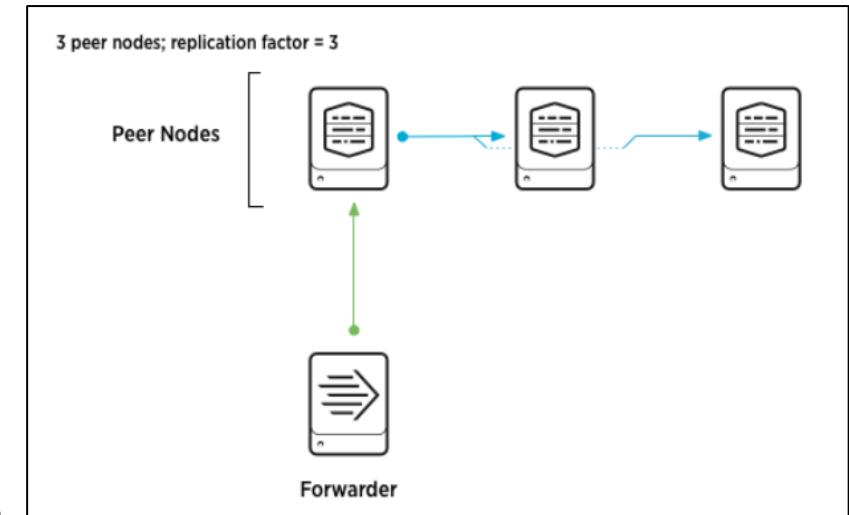
---

A cluster has one and only one master node. **Master node** coordinates activities of the **peer nodes**. It does not itself store or replicate data.

**Replication Factor** determines how many copies of data the cluster maintains.

This is a key factor since it determines the cluster's fault tolerance.

For example, if we want to ensure that system can handle the failure of two peer nodes, we must configure a replication factor of 3, which means the cluster will store 3 identical copies of data on separate nodes.



# Splunk Cluster: Master Indexer

**Search Factor** determines the number of immediately searchable copies of data the cluster maintains.

Searchable copies of data require more storage than non-searchable copies.

The difference between a searchable and a non-searchable copy of some data is this: The searchable copy contains both the data itself and some extensive index files that the cluster uses to search the data.

The non-searchable copy contains just the data. Even the data stored in the non-searchable copy, however, has undergone initial processing and is stored in a form that makes it possible to recreate the index files later, if necessary..

Splunk Enterprise stores indexed data in **buckets**, which are directories containing files of data.

A **complete** cluster maintains replication factor number of copies of each bucket, with each copy residing on a separate peer node. The bucket copies are either searchable or non-searchable.

Manager Node Configuration

Replication Factor: 3  
The number of copies of raw data that you want the cluster to maintain. A higher replication factor protects against loss of data if peer nodes fail.

Search Factor: 2  
The number of searchable copies of data the cluster maintains. A higher search factor speeds up the time to recover lost data at the cost of disk space. Must be less than or equal to Replication Factor.

Security Key: .....  
This key authenticates communication between the manager and the peers and search heads.

Cluster Label: splunk-master  
Name your cluster using this field. This label is also used to identify this cluster in the Monitoring Console.

[Back](#) [Enable Manager Node](#)

# Splunk Cluster: Peer node

You can add peer nodes to the indexer cluster at any time. To do so, just enable a Splunk Enterprise instance as a peer node.

To enable an indexer as a peer node:

1. Click Settings → Indexer clustering
2. Select Enable indexer clustering. Select Peer node and click Next.
3. There are a few fields to fill out:

**Manager URI.** Enter the manager node's URI, including its management port. For example: <https://Manager-Node-IP:8089>.

Peer node configuration

Manager URI   
E.g. https://10.152.31.202:8089

Peer replication port   
The port peer nodes use to stream data to each other (Eg: 8080).

Security key   
This key authenticates communication between the manager and the peers and search heads.

[Back](#) [Enable peer node](#)

# Splunk Cluster: Peer node

---

**Peer replication port.** This is the port on which the peer receives replicated data streamed from the other peers. You can specify any available, unused port for this purpose. This port must be different from the management or receiving ports.

**Security key.** This is the key that authenticates communication between the manager node and the peers and search heads. The key must be the same across all cluster nodes. Set the same value here that you previously set on the manager node.

6. Click **Enable peer node**.

The message appears, "You must restart Splunk for the peer node to become active."

# Lab: Create Indexer cluster and replication

---

1. Create a Master node
2. Create 2 peer nodes and provide Master URI
3. Add data to one peer node
4. Check if the replication is done on another peer node too.
5. Stop peer1 node and check it the data is still searchable.
6. upload the a new data on peer2 and stop it too.
7. Check the status and check if the data is searchable
8. Start peer1 and check if the replication from peer2 is done(peer2 is still down).
9. Start peer2 and check the replication status

# Result

---

Task	Result
1. Add data to peer1 node	Will be replicated to another peer2 node as well.
2. Stop peer1 node	Data is still searchable as it is on peer2 as well.
3. Upload the a new data on peer2 and stop it	Data is not searchable as both peer1 and peer2 are down
4. Start peer1 and check if the replication from peer2 is done	Replication will not be done
5. Start peer2 and check the replication status	Replication will be done automatically

# Splunk Cluster: Indexer Configuration Bundle

---

Configuration Bundle is a set of configuration files and apps common to all the peers.

Managed by the master and distributed to peers via bundle push operation.

It is important to note that we should never configure things directly within the peer nodes.

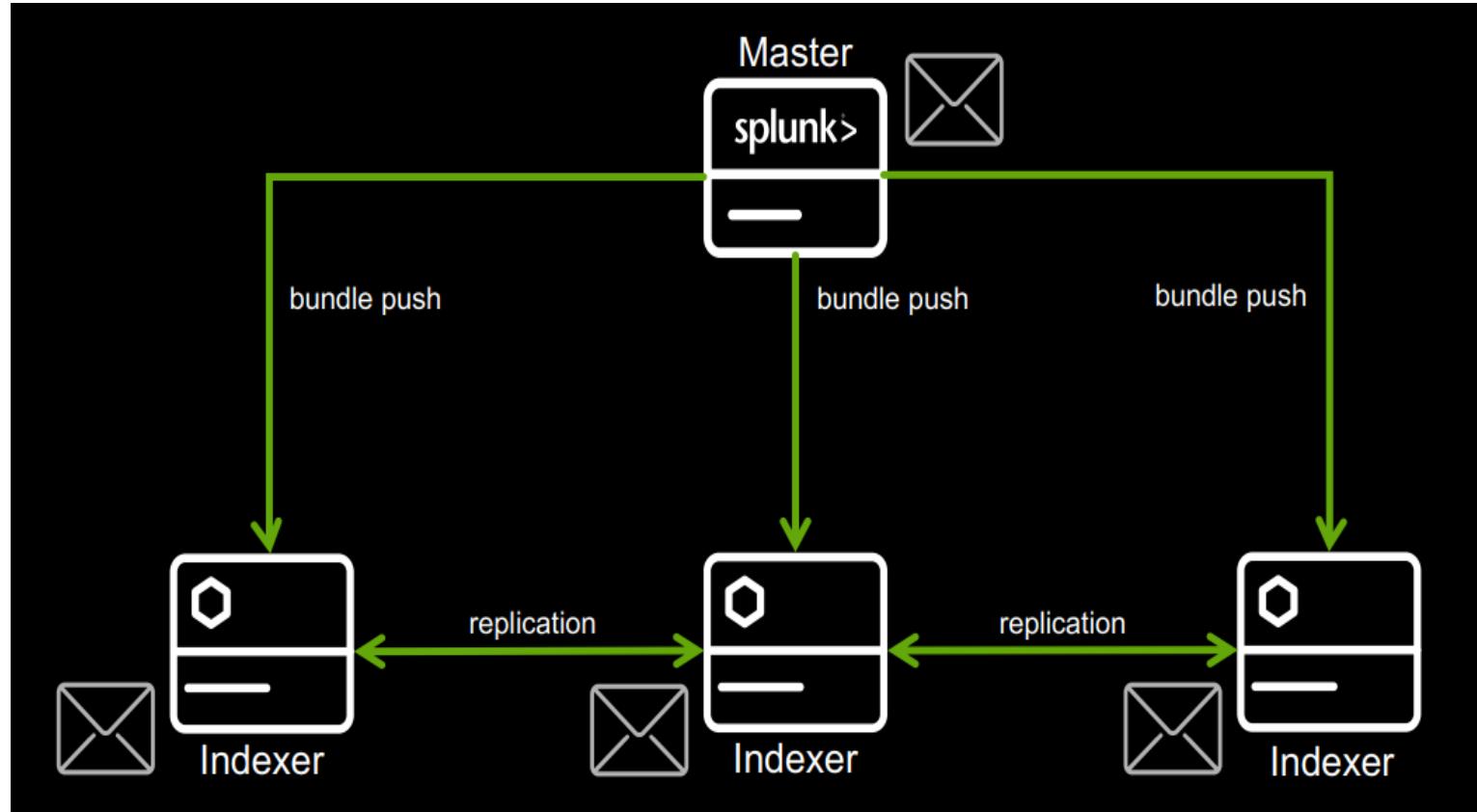
It Distributes to the peers in a single operation

- i.e. indexes.conf, props.conf, and transforms.conf
- apps will also include versions of these configuration files

Component	Location
Master Indexer	\$SPLUNK_HOME/etc/master-apps
Peer Indexer	\$SPLUNK_HOME/etc/slave-apps

# Splunk Cluster: Indexer Configuration Bundle

It is not like replication.



# Approach to send data to Indexer Cluster

---

Once we have Indexer cluster build, universal forwarder would need to start sending logs to the peer indexer nodes.

There are two ways to connect forwarders to index nodes:

1. Connect forwarders directly to the peers nodes.
2. Using the Indexer Discovery Feature (recommended)

# Approach to send data to Indexer Cluster: Direct method

---

## **1. Connect forwarders directly to the peers nodes:**

In this approach, the universal forwarders will have the IP addresses of all the peer nodes configured and the logs will be uploaded accordingly.

Steps:

- Add the file to monitor on forwarder machine

```
splunkforwarder/bin/splunk add monitor /var/log
```

- add indexer peer manually

```
splunkforwarder/bin/splunk add add forward-server peer-node1-ip:9997
```

```
splunkforwarder/bin/splunk add add forward-server peer-node1-ip:9997
```

- Check the logs on Manager node. Stop one of the peer node and check the logs again.

Verify the logs >>>

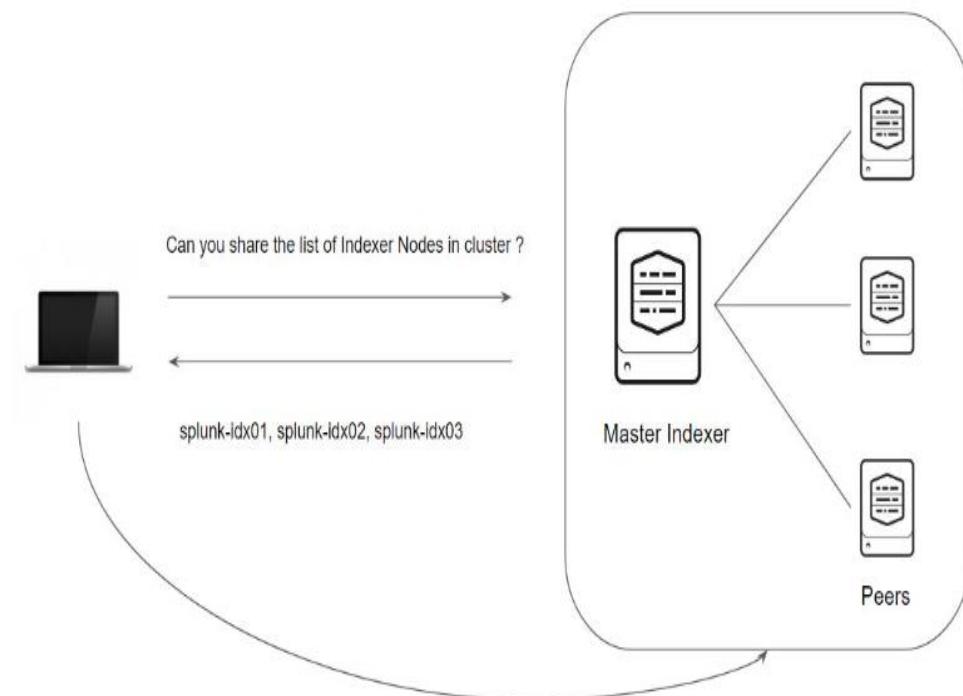
```
tail -f splunkforwarder/var/log/splunk/splunkd.log
```

# Approach to send data to Indexer Cluster: Discovery Method

## 2. Using the Indexer Discovery Feature

With the Indexer Discovery method, each forwarder queries the master node for a list of peer nodes within the cluster.

It then uses load balancing to forward data to a set of peer nodes.



# Approach to send data to Indexer Cluster: Discovery Method

1	Go to <b>splunk/etc/system/local/server.conf</b> and add below configuration (to enable the discovery feature of the Master Indexer.)
	<pre>[indexer_discovery] pass4SymmKey = password123</pre>
2	Go to <b>splunkforwarder/etc/system/local/outputs.conf</b> and add below configuration(tell universal forwarder to connect with Master Indexer)
	<pre>[indexer_discovery:master] pass4SymmKey = password123 master_uri = https://[Master-IP]:8089  [tcpout:group1] indexerDiscovery = master</pre>
3	Restart Forwarder Instance
	<pre>splunkforwarder/bin/splunk restart</pre>

# **Search Head Clustering**

# Splunk Cluster: Search Head

---

Search head is a component in Splunk Enterprise whose responsibility is to handle the search management functions, directing search requests to search peers and then migrating the results back to the users.

Search heads are used for number of functions, some of the primary one includes:

- Search related functions
- Building Dashboards and Reports
- Data Models
- Alerting Related Functionality

# Splunk Cluster: Search Head

---

Indexer nodes are primarily designed to index data from multiple sources and if we do not have search head instances then master indexer works as a search head.

Search and reporting is a resource intensive application and can slow down an indexer node if larger queries are run.

So we should avoid searching via indexer node and better prepare search heads to handle the search.

A **search head cluster** consists of a group of **search heads** that share configurations, job scheduling, and **search artifacts**. The search heads are known as the cluster **members**.

One cluster member has the role of **captain**, which means that it coordinates job scheduling and replication activities among all the members. It also serves as a search head like any other member, running search jobs, serving results, and so on. Over time, the role of captain can shift among the cluster members.

# Splunk Cluster: Search Head

---

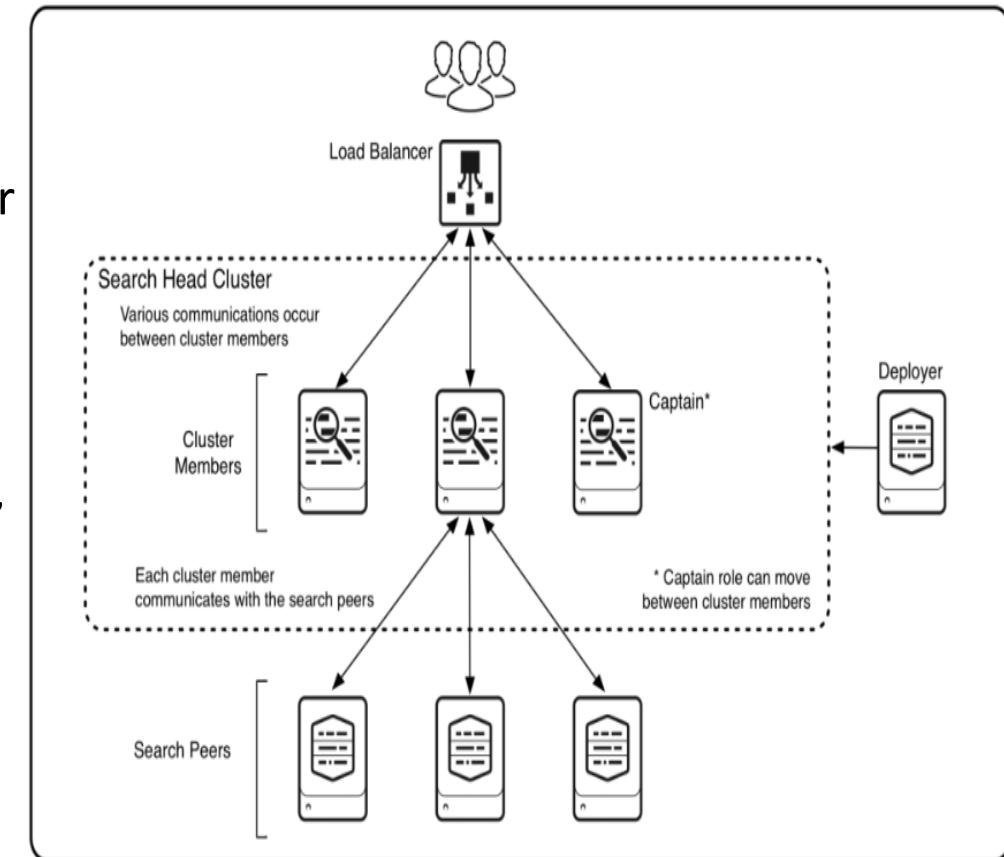
a functioning cluster requires several other components:

- **The deployer.** This is a Splunk Enterprise instance that distributes apps and other configurations to the cluster members. It stands outside the cluster and cannot run on the same instance as a cluster member. It can, however, under some circumstances, reside on the same instance as some other Splunk Enterprise components, such as a deployment server or an indexer cluster manager node.
- **Search peers.** These are the indexers that cluster members run their searches across. The search peers can be either independent indexers or nodes in an indexer cluster.
- **Load balancer.** This is third-party software or hardware optionally residing between the users and the cluster members. With a load balancer in place, users can access the set of search heads through a single interface, without needing to specify a particular search head.

# Splunk Cluster: Search Head

The **captain** is a cluster member with additional responsibilities, beyond the search activities common to all cluster members. It serves to coordinate the activities of the cluster. Any member can perform the role of captain, but the cluster has just one captain at any time. Over time, if failures occur, the captain changes and a new member gets elected to the role.

The elected captain is known as a **dynamic captain**, because it can change over time. A cluster that is functioning normally uses a dynamic captain. You can deploy a **static captain** as a temporary workaround during disaster recovery, if the cluster is not able to elect a dynamic captain.



# Splunk Cluster: Search Head Cluster Setup

---

Steps	Search Head Cluster Setup
1	Create a Splunk instance which will be working as a deployer and add the below configuration in file <b>splunk/etc/system/local/server.conf</b>
	<pre>[shclustering] pass4SymmKey = password123 shcluster_label = tech-shcluster</pre>
2	Restart Splunk server
	<pre>splunk/bin/splunk restart</pre>

# Splunk Cluster: Search Head Cluster Setup

---

Steps	
3	Setting up search heads (create 3 Splunk instances) and execute below command in all the nodes by editing the IP address of Deployer and Search Head instances.
	<code>splunk/bin/splunk init shcluster-config -mgmt_uri https://[SH-IP]:8089 -replication_port 8080 -replication_factor 1 -shcluster_label tech-shcluster -conf_deploy_fetch_url https://[DEPLOYER-IP] -secret password123</code>
4	Restart Splunk Search Heads after the above configuration
	<code>splunk/bin/splunk restart</code>

# Splunk Cluster: Search Head Cluster Setup

---

Steps	
5	Creating a caption out of all the Search Heads. Paste the below command in one of the search head instance which we want to designate as a caption.
	<code>splunk/bin/splunk bootstrap shcluster-captain -servers_list "https://[SH-01-IP]:8089","https://[SH-02-IP]:8089","https://[SH-03-IP]:8089"</code>
6	Check Search Head Cluster status
	<code>splunk show shcluster-status -auth &lt;username&gt;:&lt;password&gt;</code>

# Splunk Cluster: Search Head replication

---

There are various aspects which gets replicated within the members.

- Alert Actions
- Data Models
- Users
- Workflow actions
- Macros
- Lookups
- Event Types
- Saved searches and many more

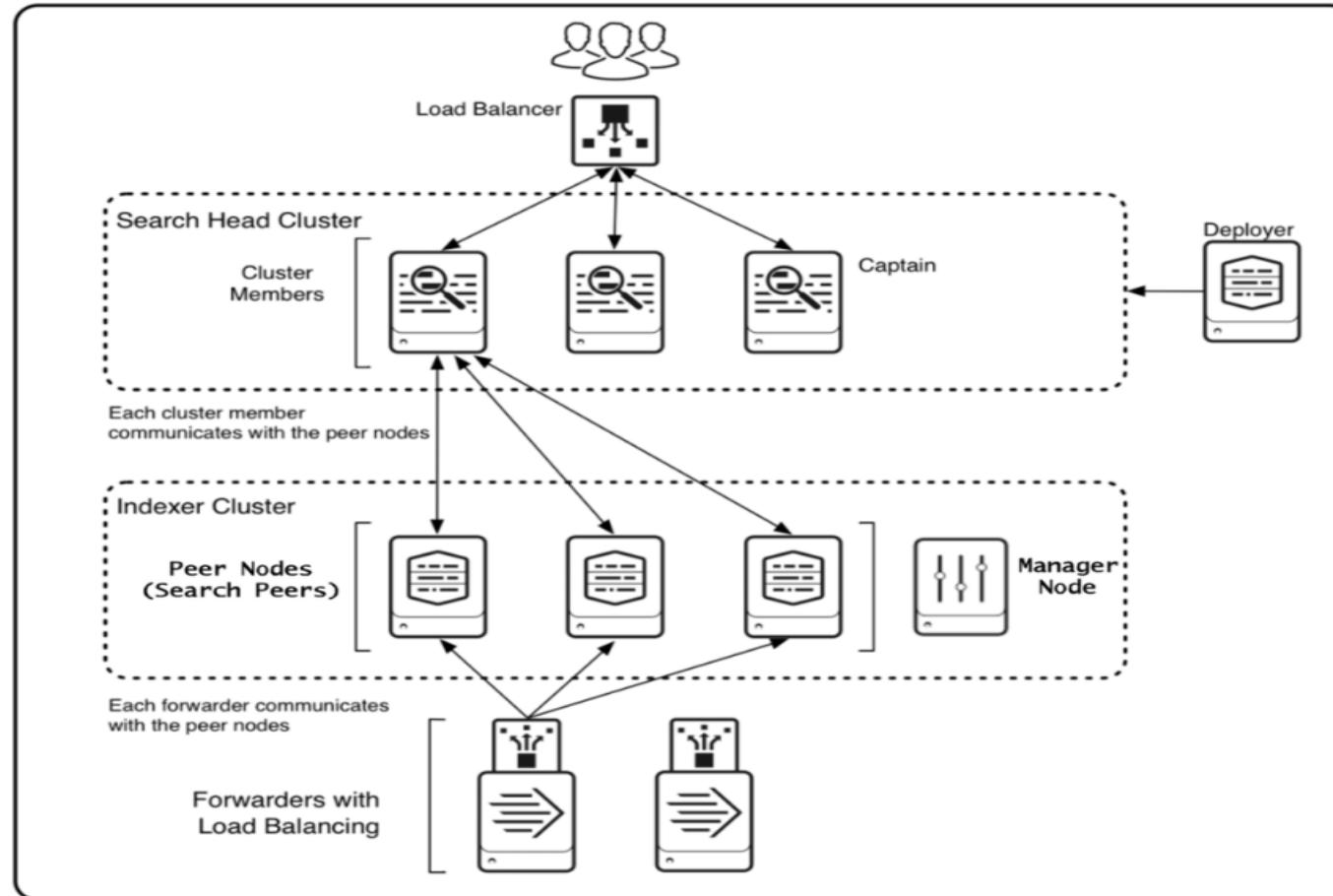
# Splunk Cluster: Pushing bundle setup via Deployer

---

The apps should be installed on deployer and deployer will push those apps to Search Head instances.

Steps	
1	Access Deployer via UI and install any app (for example splunk_app_for_nix) which will be installed at a specific location /opt/splunk/etc/apps
2	Move the application to shcluster directory
	<code>mv splunk_app_for_nix/ /opt/splunk/etc/shcluster/apps/</code>
3	Push that application to search head using the below command
	<code>splunk/bin/splunk apply shcluster-bundle -target https://[SH-IP]:8089</code>

# Splunk Cluster: Integrate Search Head Cluster with Indexer Cluster



# Splunk Cluster: Integrate Search Head Cluster with Indexer Cluster

---

Steps	
1	Go to each search heads and use the below command to connect with master indexer  splunk/bin/splunk edit cluster-config -mode searchhead -master_uri https://[Master-indexer-IP]:8089 -secret password123
2	Check Master Indexer UI if the Search Heads are added
3	Now you can search any data from the Search Heads

# **Monitoring Console**

# Monitoring Console

---

Monitoring Console allows us to view detailed information about the topology and performance of your Splunk Enterprise deployment.

The Monitoring Console provides pre-built dashboards that give you visibility into many areas of your deployment, including search and indexing performance, resource usage, license usage and more.

To access Monitoring console, navigate to **Settings → Monitoring Console**

# Monitoring Console

---

The available dashboards provide insight into the following areas of your deployment or instance:

- search performance and distributed search framework
- indexing performance
- operating system resource usage
- Splunk app key value store performance
- search head and indexer clustering
- index and volume usage
- forwarder connections and Splunk TCP performance
- HTTP Event Collector performance
- and license usage.

# **Advanced Splunk Concepts**

# Troubleshooting Configuration issues: Btool

---

The Splunk Enterprise configuration file system supports many overlapping configuration files in many different locations like local, app and default location.

The btool command simulates the merging process using the on-disk conf files and creates a report showing the merged settings.

This flexibility can make it hard to figure out exactly which configuration value Splunk Enterprise is using.

Btool is a command line tool that can help us troubleshoot configuration file issues or see what values are being used by your Splunk Enterprise installation.

Btool shows you the merged settings in the .conf files.

# Btool commands

---

<b>Review the merged settings for a conf file</b>	<code>./splunk btool inputs list</code>
<b>Review the merged settings for a conf file in an app context</b>	<code>./splunk btool --app=search inputs list</code>
<b>Review the merged settings for a conf file in an app context</b>	<code>./splunk btool --app=search inputs list</code>
<b>Review the settings for a conf file and see where the settings are merged from</b>	<code>./splunk btool inputs list --debug</code>

# Btool commands

---

<b>Find a specific setting for a conf file</b>	<code>./splunk btool inputs list   grep splunktcp</code>
<b>Find a specific setting for a conf file and see where the setting is merged from</b>	<code>./splunk btool inputs list --debug   grep splunktcp</code>
<b>Find a specific setting for a conf file, see where the settings is merged from, and place the report into a file</b>	<code>./splunk btool inputs list --debug   grep splunktcp &gt; /tmp/inputs_splunktcp</code>
<b>Look for the error if you see a "typo in stanza" message</b>	<code>splunk btool check</code>

# Data Models

---

Why we need Data Models??

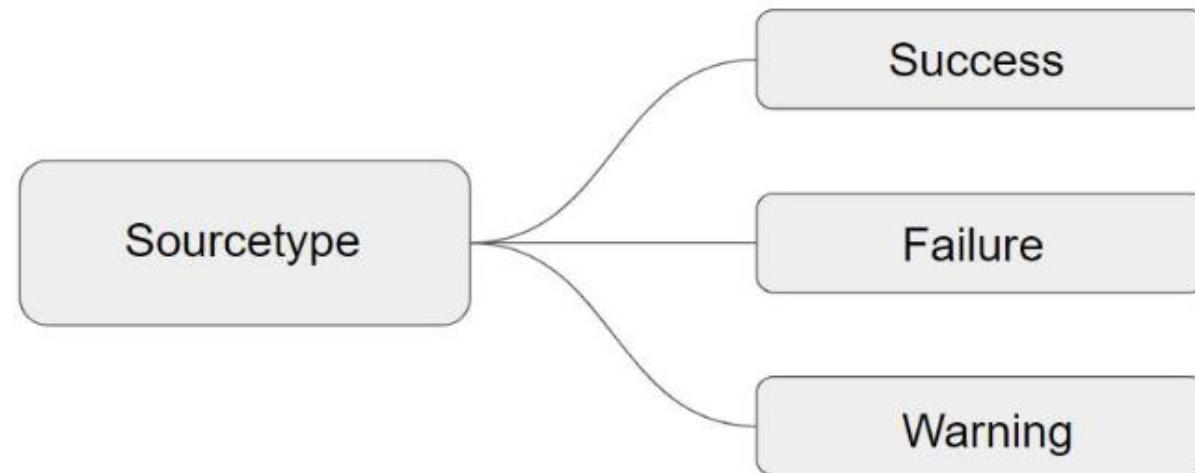
- You have data which contains certain critical business indicators.
- The data structure is multi-layered and complicated
- The users who wants to see business indicators are less technical
- You do not want to be point of contact for everything they want out of data

Data models enable users of Pivot to create compelling reports and dashboards without designing the searches that generate them

# Data Models

---

Data Model allows us to provide meaningful representation of underlying data.



# **Cloud Inputs**

# Splunk Connect for Syslog

---

Splunk Connect for Syslog (SC4S) is a distribution of syslog-ng that simplifies getting your syslog data into Splunk Enterprise and Splunk Cloud.

SC4S provides a runtime-agnostic solution that lets you deploy using the container runtime environment of choice and a configuration framework. This lets you process logs out-of-the-box from many popular devices and systems.

# Splunk Connect for Syslog

---

Create the following default indexes that are used by SC4S:

- email
- epav
- fireeye
- gitops
- infraops
- netauth
- netdlp
- netdns
- netfw
- netids
- netops
- netwaf
- netproxy
- netipam
- oswinsec
- osnix

# Use cases of SC4S

---

## **Enterprise Network Monitoring:**

- Collects syslog data from network devices (routers, switches, firewalls) to monitor and troubleshoot network issues.

## **Security Information and Event Management (SIEM):**

- Aggregates syslog data from various security appliances (IDS/IPS, firewalls, anti-virus) to detect and respond to security incidents.

## **Compliance Monitoring:**

- Collects and forwards logs required for regulatory compliance (e.g., PCI-DSS, HIPAA) to ensure that the organization meets legal and regulatory requirements.

## **Operational Visibility:**

- Provides operational insights by collecting logs from various IT infrastructure components (servers, applications) for performance monitoring and troubleshooting.

# Splunk Connect for Syslog: Demo

---

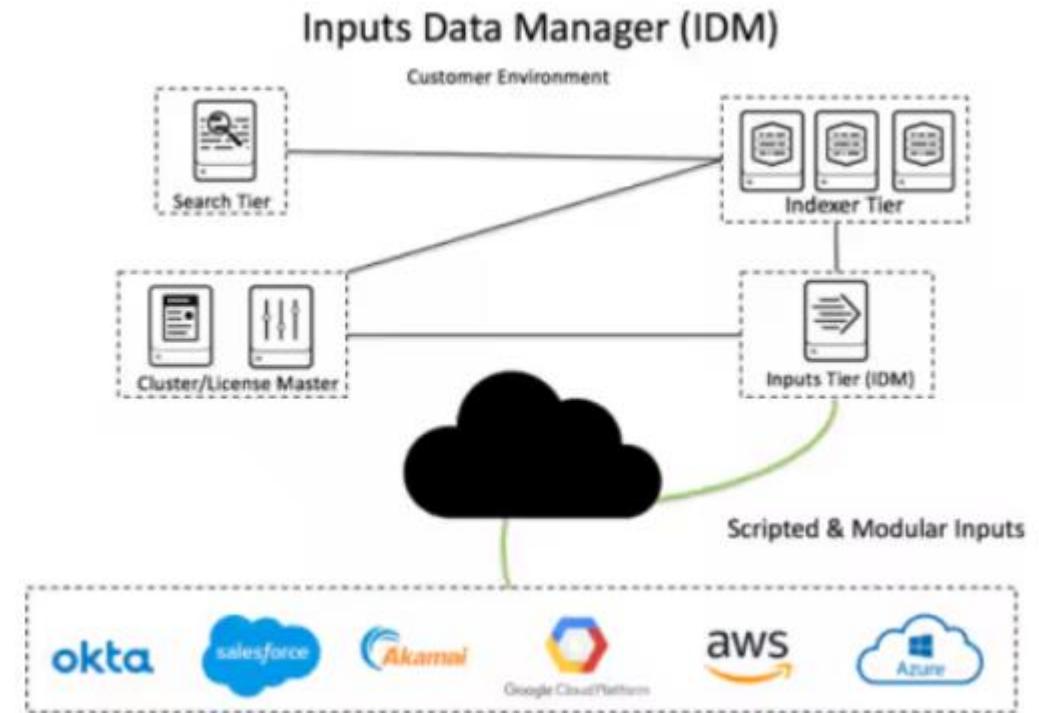
1. Create a HEC Token
2. run the below script.

<https://github.com/CloudSihmar/splunk-docs/blob/main/sc4s%20script>

3. Check the logs in Splunk server (index=\* sourcetype="sc4s:\*)

# Inputs Data Manager

- Inputs Data Manager (IDM), is a Splunk instance within a Cloud Stack that provides users an ability to set up and configure modular and scripted inputs.
- As a part of a stack, IDM is managed by Splunk. IDM is a unique instance, meaning that it exists independently and separately from a Search Head, and does not belong to a Search or Indexing cluster.
- Search capabilities are enabled, however this is reserved to app-only default reports and scheduled searches.
- Many of the inputs reside in Cloud contexts, such as **AWS, Salesforce, Azure, GCP**, and many others. The Inputs Data Manager was introduced to aid the ingestion of these cloud data sources.



# Inputs Data Manager

---

IDM is neither a forwarder, nor a heavy forwarder. As opposed to a forwarder/heavy forwarder, IDM are not suitable to perform these tasks:

- Parsing on inputs and anonymization,
- Network inputs such as UDP/TCP,
- Inputs via HTTP Event Collector (HEC),
- Receiving syslog input,
- Apps and Technical Add-ons directly integrated with Enterprise Security (ES)

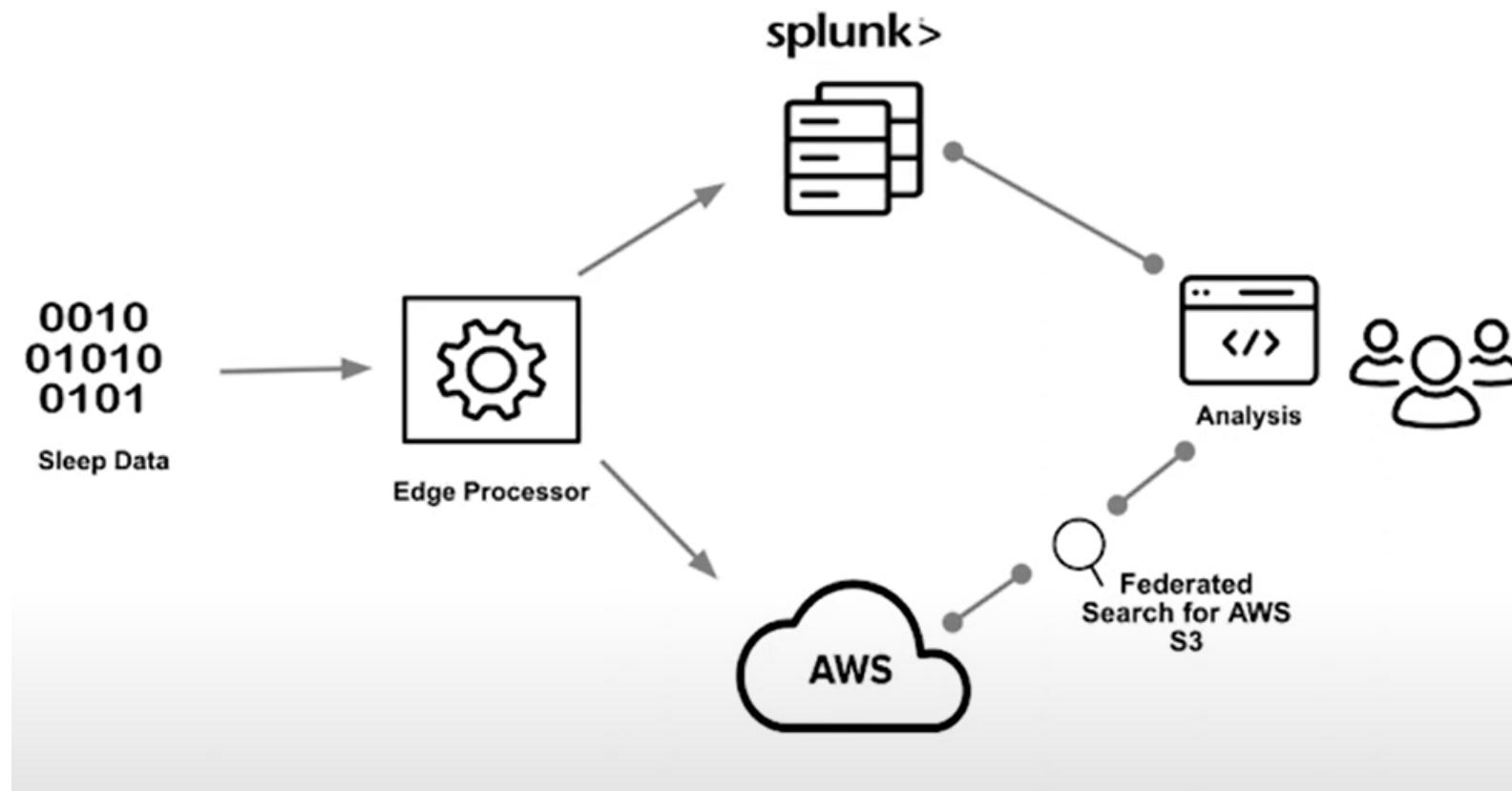
# Splunk Edge Processor

---

- Edge Processor helps customers achieve greater efficiencies in data transformation close to the data source, and improved visibility into data in motion.
- Edge Processor provides customers new abilities to filter and mask, and otherwise transform their data, before routing it to supported destinations.
- Edge Processor joins Ingest Actions as part of Splunk's pre-ingest data transformation capabilities.
- Edge Processor nodes are easily installed and configured on customer servers or customer cloud infrastructure using a single command, and managed completely from Splunk Cloud Platform.
- These nodes are an intermediate forwarding tier, and receive data from edge sources. Customers manage their entire fleet of edge processors and have visibility into both inbound and outbound data volumes through their edge processor network, all from a single place.

# Splunk Edge Processor

---



# Splunk Edge Hub

---

- It is a physical device that can help collect, distribute and act on data from edge devices and sensors, making it easier to capture and act on data that can be difficult to access physically or digitally.
- The Splunk Edge Hub device's screen lets you observe sensor data in-person, in the Splunk Mobile and AR apps, as well as on the Splunk platform.

# Splunk Edge Hub

---

Splunk Edge Hub is able to gather data from a variety of sources including:

- **Built-In Sensors:** Temperature, Light, Humidity, Sound, Pressure, Gas, Gyroscopic, Acceleration.
- **Native Protocol Support:**
  - **MQTT:** An internal MQTT Broker enables data collection from a variety of IoT sensors.
  - **SNMP:** Monitor and poll current and legacy hardware such as power distribution units.
  - **MODBUS:** Communicate using the Modbus TCP protocol with sensors and devices.
  - **OPC UA:** Collect metrics from hardware using the OPC Unified Architecture protocol.
- **Connectivity:** The Edge Hub device has connectivity options including ethernet, WiFi, Bluetooth and LTE connectivity.

# Splunk Support Programs

---

Splunk offers a variety of support plans for customers.

These are primarily divided into the following categories:

- Community
- Base
- Standard
- Premium