Splunk Versions

There are three different versions of Splunk

- Splunk Enterprise
- Splunk Light
- Splunk Cloud

Splunk Enterprise

Big IT enterprise uses the Splunk Enterprise Version. With the help of the Splunk tool, we can collect and analyze the data from mobile phones, websites, and applications, etc.

Splunk Cloud

Splunk Cloud is a website that is the host. It possesses the same features as the company version. It can be used from Splunk or the cloud platform AWS.

Splunk Light

The free version of Splunk Illumination. It enables scanning, recording, and editing of your log data. Compared with other versions, it has limited functionalities and features.

Splunk Architecture: Components and Best Practices

Splunk is a distributed system that aggregates, parses and analyses log data. In this article we'll help you understand how the Splunk architecture, the Splunk big data pipeline works, how the Splunk components like the forwarder, indexer and search head interact, and the different topologies you can use to scale your Splunk deployment.
This is part of an extensive series of guides about data security.
In this article:

How Splunk Works: Stages in the Data Pipeline

There are three main stages in the Splunk data pipeline: data collection, data indexing, and finally, search and analysis.

Data Collection

The first stage of the Splunk data pipeline is data collection. Splunk can ingest data from a wide variety of sources, including files, directories, network events, and APIs. It supports common data formats such as CSV, JSON, and XML, as well as custom formats. Data collection is typically performed using forwarders, which are lightweight agents that can be installed on any machine that generates data. Learn more in the Splunk Components section below.

Data Indexing

Once data is collected, it moves on to the indexing stage. Splunk indexes the data by parsing it into individual events and extracting relevant fields, such as timestamps, source types, and host information. This process enables efficient searching and analysis of the data later on.

Indexing can be performed on a single Splunk instance or distributed across multiple indexers for scalability and redundancy. In a distributed environment, Splunk uses an indexing cluster to ensure that data is evenly distributed and replicated across multiple indexers.

Data Searching and Analysis

After data is indexed, it can be searched and analyzed using Splunk's powerful search language, the Search Processing Language (SPL). SPL allows users to perform a wide range of operations on the data, such as filtering, aggregation, correlation, and statistical analysis. Users can create custom reports, dashboards, and alerts based on the results of their searches and analyses.

Splunk also provides a variety of pre-built apps and add-ons that extend its capabilities and integrate with other systems, such as IT service management tools, security information and event management systems, and cloud platforms.

Splunk Enterprise vs Splunk Cloud: How Does it Affect Your Architecture?

Splunk is available in two versions:

- Splunk Enterprise – the paid version
- Splunk Cloud – provided as a service with subscription pricing

Your selection of a splunk edition will affect your architecture. This is summarized in the table below.

Splunk Edition	Limitations	Architectural Considerations
Enterprise	Unlimited	Supports single site clustering and multi-site clustering for disaster recovery
Cloud	Depending on service package	Clustering managed by Splunk

Note: The free version of Splunk, which was called Splunk Light, is no longer available (End of Life was May, 2021).

Splunk Components

The primary components in the Splunk architecture are the forwarder, the indexer, and the search head.

Splunk Forwarder

The forwarder is an agent you deploy on IT systems, which collects logs and sends them to the indexer. Splunk has two types of forwarders:

- Universal Forwarder – forwards the raw data without any prior treatment. This is faster, and requires less resources on the host, but results in huge quantities of data sent to the indexer.
- Heavy Forwarder – performs parsing and indexing at the source, on the host machine and sends only the parsed events to the indexer.

Splunk Indexer

The indexer transforms data into events (unless it was received pre-processed from a heavy forwarder), stores it to disk and adds it to an index, enabling searchability.
The indexer creates the following files, separating them into directories called buckets:

- Compressed raw data
- Indexes pointing to raw data (.TSIDX files)
- Metadata files

The indexer performs generic event processing on log data, such as applying timestamp and adding source, and can also execute user-defined transformation actions to extract specific information or apply special rules, such as filtering unwanted events.
In Splunk Enterprise, you can set up a cluster of indexers with replication between them, to avoid data loss and provide more system resources and storage space to handle large data volumes.

Splunk Search Head

The search head provides the UI users can use to interact with Splunk. It allows users to search and query Splunk data, and interfaces with indexers to gain access to the specific data they request.

Splunk provides a distributed search architecture, which allows you to scale up to handle large data volumes, and better handle access control and geo-dispersed data. In a distributed search scenario, the search head sends search requests to a group of indexers, also called search peers. The indexers perform the search locally and return results to the search head, which merges the results and returns them to the user.

There are a few common topologies for distributed search in Splunk:

- One or more independent search heads to search across indexers (each can be used for a different type of data)
- Multiple search heads in a search head cluster – with all search heads sharing the same configuration and jobs. This is a way to scale up search.
- Search heads as part of an indexer cluster – promotes data availability and data recovery.

Deployment server

A Splunk Enterprise instance can also serve as a **deployment server**. The deployment server is a tool for distributing configurations, apps, and content updates to groups of Splunk Enterprise instances. You can use it to distribute updates to most types of Splunk components: forwarders, non-clustered indexers, and non-clustered search heads. See About deployment server and forwarder management in the *Updating Splunk Enterprise Instances* manual.

Use the deployer to distribute apps and configuration updates

The **deployer** is a Splunk Enterprise instance that you use to distribute apps and certain other configuration updates to search head cluster members. The set of updates that the deployer distributes is called the **configuration bundle**.

The deployer distributes the configuration bundle in response to your command, according to the deployer push mode that you select. The deployer also distributes the bundle when a member joins or rejoins the cluster.

Which configurations does the deployer manage?

The deployer has these main roles:

- It handles migration of app and user configurations into the search head cluster from non-cluster instances and search head pools.
- It deploys baseline app configurations to search head cluster members.
- It provides the means to distribute non-replicated, non-runtime configuration updates to all search head cluster members.

You do not use the deployer to distribute search-related runtime configuration changes from one cluster member to the other members. Instead, the cluster automatically replicates such changes to all cluster members. For example, if a user creates a saved search on one member, the cluster automatically replicates the search to all other members. See Configuration updates that the cluster replicates. To distribute all other updates, you need the deployer.

Configurations move in one direction only: from the deployer to the members. The members never upload configurations to the deployer. It is also unlikely that you will ever need to force such behavior by manually copying files from the cluster members to the deployer, because the members continually replicate all runtime configurations among themselves.

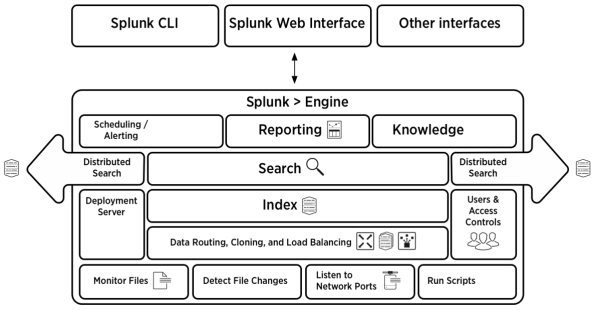
Types of updates that the deployer handles

These are the specific types of updates that require the deployer:

- New or upgraded apps.
- Configuration files that you edit directly.
- All non-search-related updates, even those that can be configured through the CLI or Splunk Web, such as updates to `indexes.conf` or `inputs.conf`.
- Settings that need to be migrated from a search head pool or a standalone search head. These can be app or user settings.

Putting it All Together: Splunk Architecture

The following diagram illustrates the Splunk architecture as a whole.



Source: Splunk Documentation

From top to bottom:

- Splunk gathers logs by monitoring files, detecting file changes, listening on ports or running scripts to collect log data – all of these are carried out by the Splunk forwarder.
- The indexing mechanism, composed of one or more indexers, processes the data, or may receive the data pre-processed by the forwarders
    - The deployment server manages indexers and search heads, configuration and policies across the entire Splunk deployment.
    - User access and controls are applied at the indexer level – each indexer can be used for a different data store, which may have different user permissions.
- The search head is used to provide on-demand search functionality, and also powers scheduled searches initiated by automatic reports.
- The user can define Scheduling, Reporting and Knowledge objects to schedule searches and create alerts.
- Data can be accessed from the UI, the Splunk CLI, or APIs integrating with numerous external systems.

Splunk Design Principles and Best Practices

Now that we have covered Splunk architecture in detail, let's review some best practices that will help you build the most effective architecture for your big data project.

Scalability

Splunk is designed to scale horizontally by adding additional indexers or search heads as needed. To ensure optimal performance and resource utilization in large deployments, it's essential to distribute the workload evenly across all available components. Load balancing techniques, such as round-robin DNS, can be used to achieve this.

High Availability

In a distributed Splunk deployment, it's crucial to ensure that data remains accessible even in the event of hardware failure or network issues. Splunk supports data replication and search head clustering to provide high availability and fault tolerance.

Security

Securing your Splunk environment is critical to protecting sensitive data and ensuring compliance with data protection regulations. Best practices for Splunk security include:

- Enabling encryption for data in transit and at rest
- Implementing strong access controls and authentication mechanisms
- Regularly monitoring and auditing Splunk activity for signs of unauthorized access or suspicious behavior

Data Retention and Archiving

It's important to define and implement a data retention policy that meets your organization's legal and operational requirements. Splunk allows you to configure data retention settings on a per-index basis, giving you granular control over how long data is retained and when it should be deleted or archived.

Monitoring and Optimization

Regularly monitoring and optimizing your Splunk environment is essential for maintaining optimal performance and resource usage. Key areas to monitor include:

- Search performance and resource utilization
- Indexing performance and disk space usage
- Forwarder health and data ingestion rates

By following these design principles and best practices, you can ensure that your Splunk architecture is scalable, secure, and efficient, enabling you to unlock the full potential of your machine-generated data and drive better decision-making across your organization.

Reduce Splunk Storage Costs by 70% with SmartStore and Cloudian

Splunk's new SmartStore feature allows the indexer to index data on cloud storage such as Amazon S3. Cloudian HyperStore is an S3-compatible, exabyte-scalable on-prem storage pool that SmartStore can connect to. Cloudian lets you decouple compute and storage in your Splunk architecture and scale up storage independently of compute resources.

You can configure SmartStore to retain hot data on the indexer machine, and move warm or cold data to on-prem Cloudian storage. Cloudian creates a single data lake with seamless, modular growth. You can simply add more Cloudian units, with up to 840TB in a 4U chassis, to expand from terabytes to an exabyte. It also offers up to 14 nines durability.

Learn more about Cloudian's big data storage solutionsLearn more about Cloudian's solution for Splunk storage.

Splunk default ports:

Port	Type	Description
9997	Convention	Splunk-to-Splunk (e.g., Forwarding Data)
8000	Default	Splunk Web (HTTP by Default)
8089	Default	API Access to Servers
8089	Default	Non-Forwarding Splunk-to-Splunk Communication
9100 / 8080	Convention	Index Cluster Replication. Different sources list different recommendation
200 / 9777	Convention	Search Head Cluster Replication Different sources list different recommendation
191	Default	KVStore, Internal and Replication
8088	Default	HTTP Event Collector
514	Convention – Not Recommended	Syslog, TCP or UDP. Recommendation is to send Syslog to a Syslog Collector tool (Syslog-NG, rsyslog, etc) instead of to Splunk

Splunk Cloud Default Ports

'port	Type	Description
443	Default – Immutable	Web Connection. Mandatory SSL
443	Default – Immutable	HTTP Event Collector
9997	Default – Immutable	Splunk-to-Splunk (e.g., Forwarding Data)
8089	Default – Immutable	API Access (the SH, Premium SH, or IDM)
8089	Default – Immutable	Federated Search
8089	Default – Immutable	Hybrid Search (While it lasts)

Splunk Observability Cloud OpenTelemetry Collector Default Ports

'port	Type	Description
13133	Default	Health Check Extension
6831, 6832, 14250, 14268	Default	Jaeger Receiver – Thrify and gRPC

55679	Default	ZPages extension
4317, 4318	Default	OLTP receiver – gRPC and http
6060	Default	HTTP Forwarder – Smart Agent
7276	Default	SAPM Trace receiver
8888	Default	Internal Prometheus
8006	Default	Fluent forward receiver
9080	Default	Smart Agent receiver – SignalFxForwarder
9411	Default	Zipkin Receiver
9943	Default	SignalFx receiver – metrics and events

Splunk SOAR Default Ports for Clustered Deployments – On-Prem

port	Type	Description
22	Default	SSH – Cluster admin
80	Default	HTTP (redirected to HTTPS)
443	Default	HTTPS (unprivileged install is changeable)
443	Default	REST API port
8443	Default	HTTPS default when using AMI-based deployment
4369	Default	RabbitMQ port mapper
5100 – 5120	Default	Daemon inter-process ports
5671	Default	RabbitMQ service
8300	Default	Consol RPC services

8301	Default	Consol internode communication
8302	Default	Consol internode communication
8888	Default	WebSocket server
15672	Default	RabbitMQ admin UI — Optional
25672	Default	RabbitMQ internode communications

Disclaimer: These ports are current as of January 2023. Most of these ports have been static through the years but expect more ports to support new services and offerings.