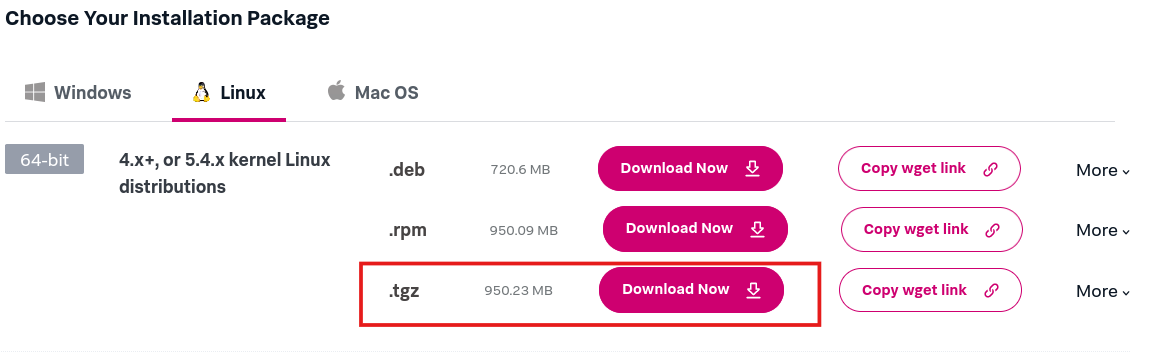
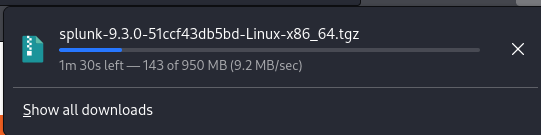
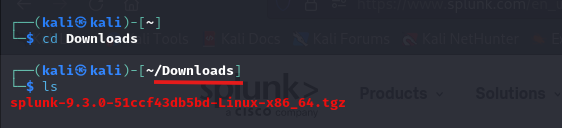
Splunk is a SIEM solution that allows us to collect, analyze, and correlate logs in a centralized server in real-time.

**Splunk Deployment on Linux Server:**

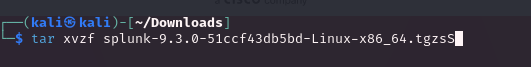
Splunk supports all major OS versions, Typically, we would **create an account on splunk.com** and go to [this](https://www.splunk.com/en_us/download/splunk-enterprise.html) Splunk Enterprise download link to select the installation package for the latest version.  
****

After accepting the terms and conditions the download will start automatically.  
****

Now open the terminal and navigate to downloads folder, where we can find the downloaded Splunk zip file.

****

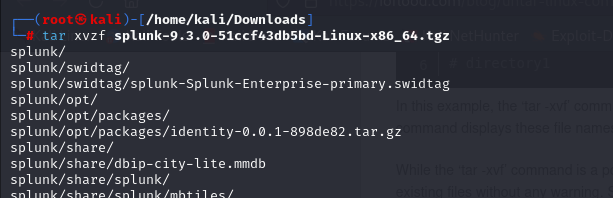
The file is in zip format, to unzip the file type below code to unzip.



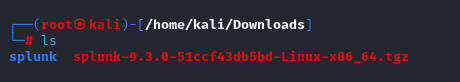
tar: is the Linux command to unzip the file. We can also use commands like unzip, gzip etc.

* + ‘x’: This option tells the tar command to extract the contents of a tar file.
  + ‘v’: The verbose option enables the command to display the progress in the terminal as it extracts files.
  + ‘z’: Filter the archive through gzip
  + ‘f’: This option allows you to specify the name of the tar file.

Note: To unzip the files, we must have root permissions.



A sperate folder with name of “splunk” will be appear after the successful unzipping the above tar zip.



Let's now move this(splunk) folder to the **/opt/** directory and start working on Splunk from there.

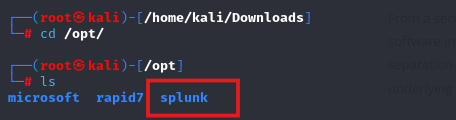
Note:

* The term "/opt" stands for "optional" and signifies that the directory is intended for the installation of optional or add-on software packages.
* the "/opt" directory provides a designated location for software installations that are not part of the core operating system.

To move splunk folder from Downloads folder to /opt/ folder we will use below command.

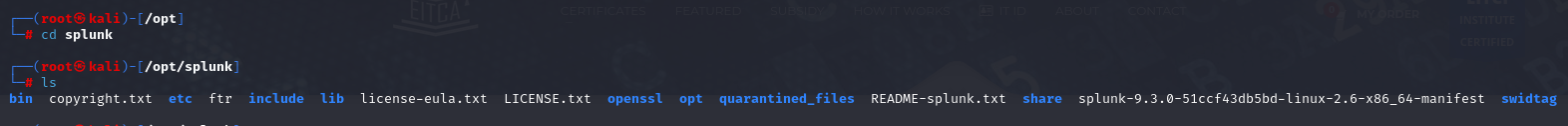


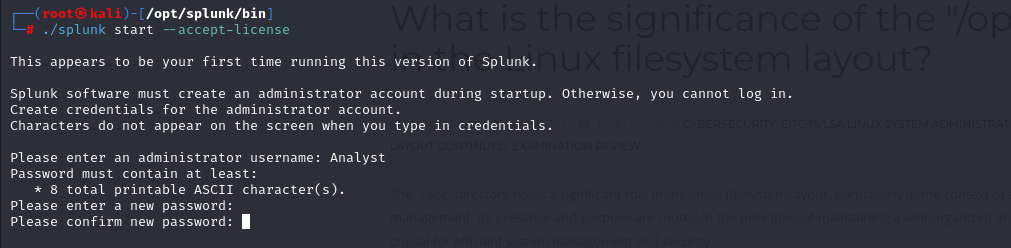
Now navigate to **/opt/** directory check for splunk folder in it.



The above step unzips the Splunk installer and installs all the necessary binaries and files on the system.

Once installed, go to the directory **/opt/splunk/bin** and run the following command to start Splunk. **./splunk start --accept-license**. As it is the first time, we are starting the Splunk instance, it will ask the user for admin credentials. Create a user account and proceed.





If the installation is successful we will see the below message on terminal.  
A screen shot of a computer

Description automatically generated

Accessing Splunk:  
We successfully installed Splunk on our Linux machine, To access Splunk, open the browser within the VM and go to the address [**http://MACHINE\_IP/host\_name:8000**](http://MACHINE_IP/host_name:8000)

**A screenshot of a computer

Description automatically generated**

Splunk: Interacting with CLI (Command Line Interface)

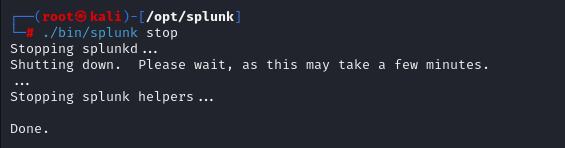
Now that we have installed Splunk, it's important to learn some key commands while interacting with Splunk instances through CLI. **These commands are run from the /opt/splunk/ directory.**

* The **splunk start** command is used to start the Splunk server.

A screen shot of a computer

Description automatically generated

* The **splunk stop** command is used to stop the Splunk server.

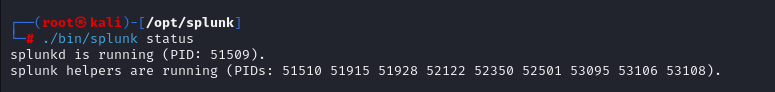


* The **splunk restart** command is used to restart the Splunk server.

A screen shot of a computer

Description automatically generated

* The **splunk status** command is used to check the status of the Splunk server.



* The **splunk add oneshot** command is used to add a single event to the Splunk index.
* The **splunk search** command is used to search for data in the Splunk index.
* The most important command is the help command which provides all the help options. (splunk help)

Splunk: Data Ingestion

Configuring data ingestion is an important part of Splunk. This allows for the data to be indexed and searchable for the analysts. Splunk accepts data from various log sources like Operating System logs, Web Applications, Intrusion Detection logs, Osquery logs, etc

Splunk Forwarders:

Splunk has two primary types of forwarders that can be used in different use cases

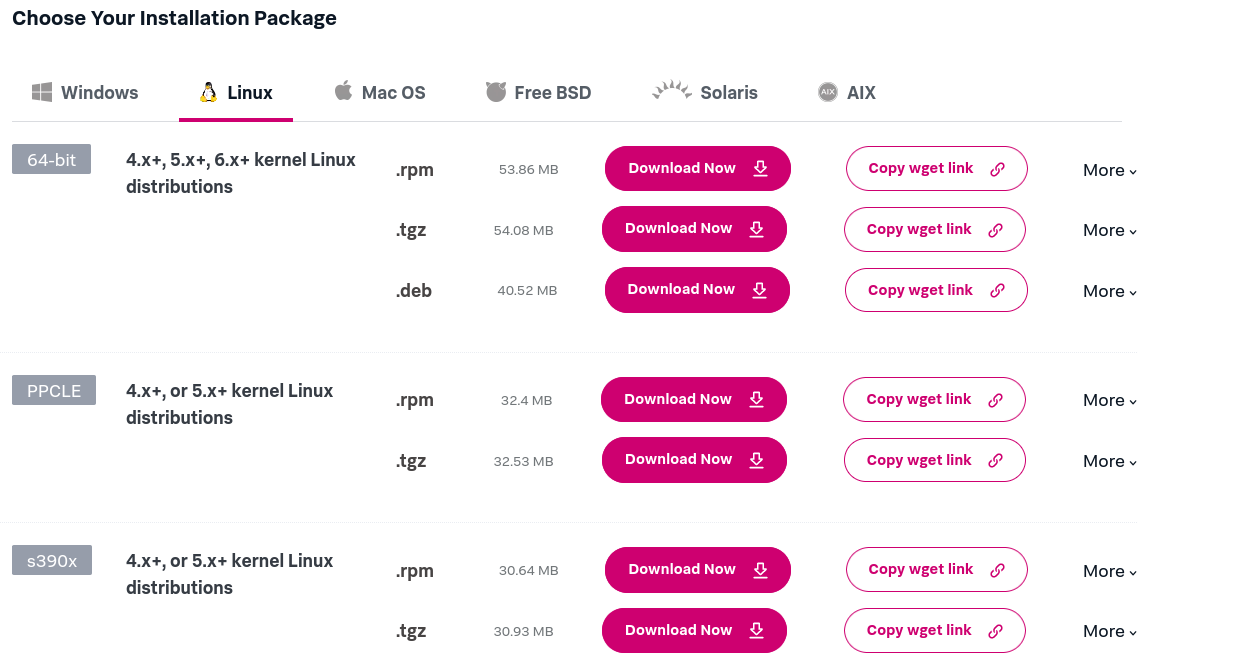
Heavy Forwarder:

Heavy forwarders are used when we need to apply a filter, analyze or make changes to the logs at the source before forwarding it to the destination

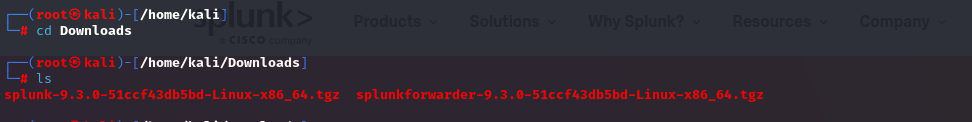
Universal Forwarder:

It is a lightweight agent that gets installed on the target host, and its main purpose is to get the logs and send them to the Splunk instance or another forwarder without applying any filters or indexing.

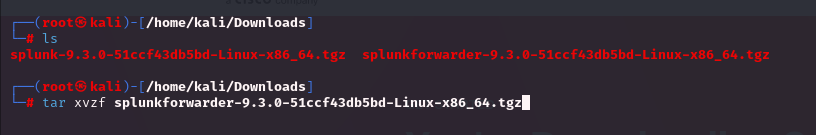
Universal forwarders can be downloaded from the official Splunk [website](https://www.splunk.com/en_us/download/universal-forwarder.html).

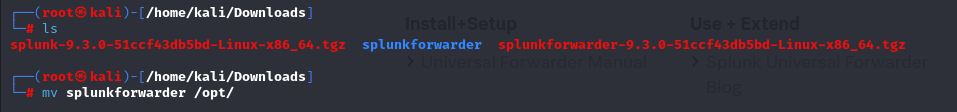


Navigate to Downloads Directory to unzip the downloaded file (universal forwarder)



Unzip the folder and move it to the /opt/ directory.





A screen shot of a computer

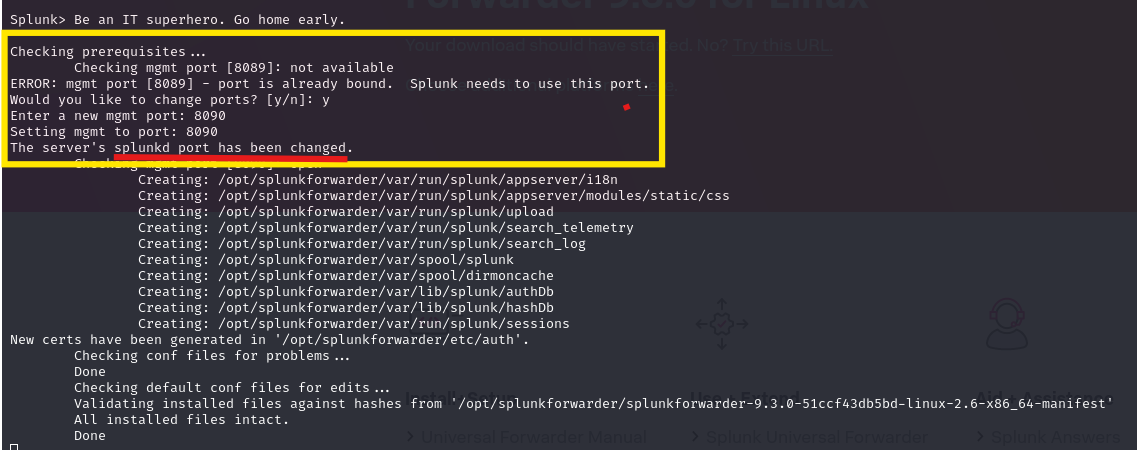
Description automatically generated

We will run the Splunk forwarder instance now and provide it with the new credentials

A screenshot of a computer

Description automatically generated

By default, Splunk forwarder runs on port 8089. If the system finds the port unavailable, it will ask the user for the custom port. If we want to change the default just enter ‘y’ and enter the custom port number, now I am changing 8089 port number to 8090 port number.



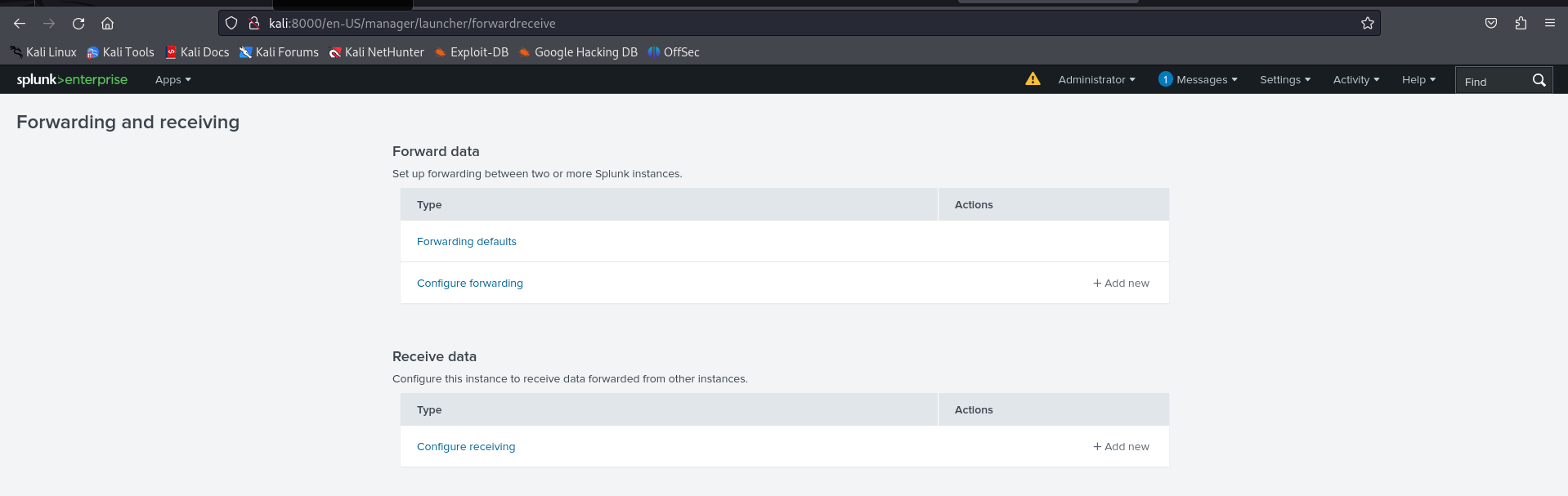
Configuring Forwarder on Linux:

Now that we have installed the forwarder, it needs to know where to send the data. So we will configure it on the host end to send the data and configure Splunk so that it knows from where it is receiving the data.

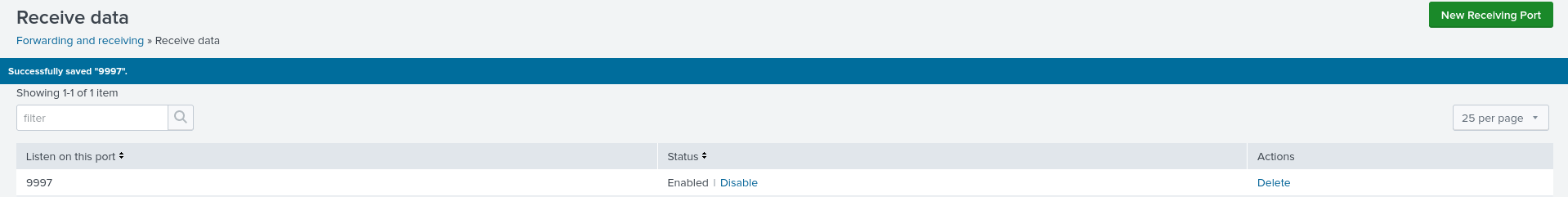
Splunk configuration:

Log into Splunk and Go to Settings -> Forward and receiving tab

It will show multiple options to configure both forwarding and receiving. As we want to receive data from the Linux endpoint, we will click on **Configure receiving** and then proceed by configuring a new receiving port.



By default, the Splunk instance receives data from the forwarder on the port **9997**. It's up to us to use this port or change it. For now, we will configure our Splunk to start **listening on port 9997** and **Save**, as shown below:



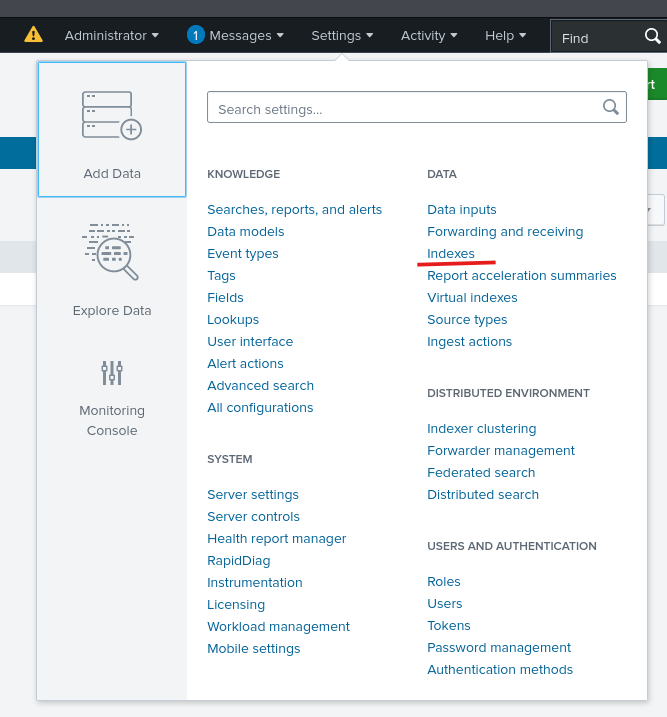
Our listening port 9997 is now enabled and waiting for the data.

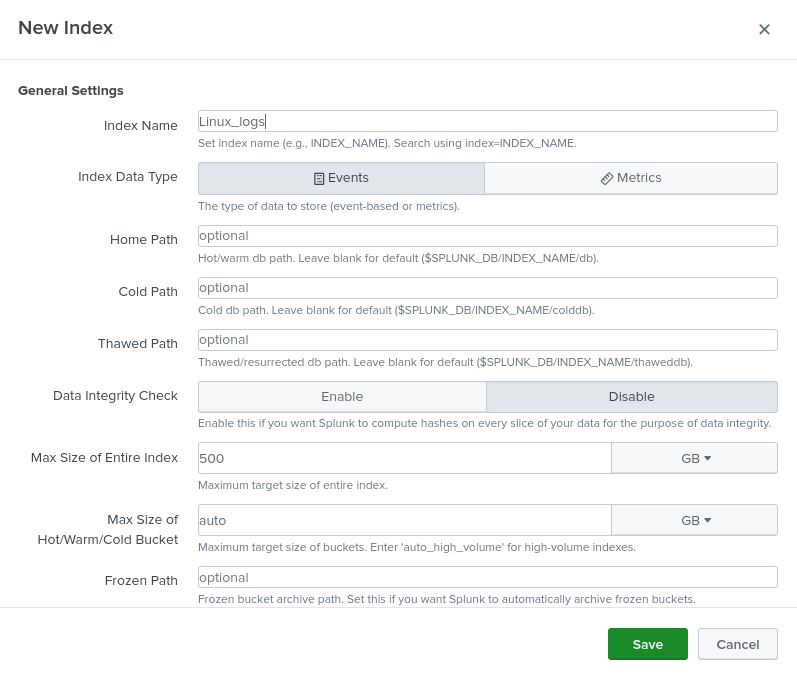
Creating Index:

Now that we have enabled a listening port, the important next step is to create an index that will store all the receiving data. If we do not specify an index, it will start storing received data in the default index, which is called the **main** index.

The indexes tab contains all the indexes created by the user or by default.

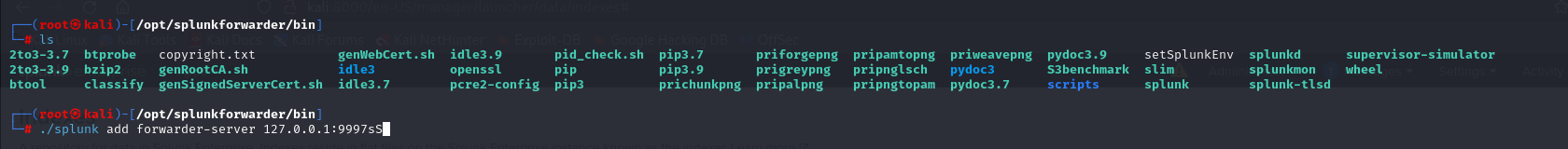
Click the **New Index** button, fill out the form, and click **Save**to create the index. Here we have created an index called **Linux\_host**



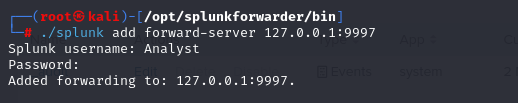


Configuring Forwarder:

It's time to configure the forwarder to ensure it sends the data to the right destination. Back in the Linux host terminal, go to the **/opt/splunkforwarder/bin** directory:



This command will add the forwarder server, which listens to port 9997.

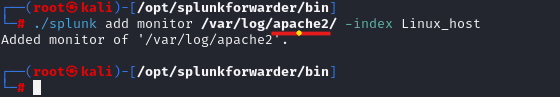


Linux Log sources:

Linux stores all its important logs into the **/var/log** file.

we will ingest syslog into Splunk. All other logs can be ingested using the same method.

we will tell Splunk forwarder which logs files to monitor. Here, we tell Splunk Forwarder to monitor the /var/log/apache2 file.



In the place Apache log file, we can replace with other log files to monitor.

I have used “logger” command to generate the log file



Splunk is a powerful data analytics platform used for searching, monitoring, and analyzing large amounts of machine-generated data. Data parsing in Splunk involves extracting relevant fields and transforming the data into a structured format for efficient analysis.

**Splunk Data Processing:**

Step 1: Understand the Data Format:

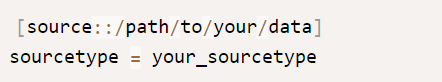
First, you need to understand the data format you want to parse. Splunk supports various data formats, such as CSV, JSON, XML, syslog, and more. Determine the format of your data source and the relevant fields you want to extract.

Step 2: Identify the Sourcetype:

In Splunk, **the sourcetype represents the format of the data being indexed**. It helps Splunk apply the appropriate parsing rules. If your data source does not have a pre-defined sourcetype, you can create a custom one in Splunk.

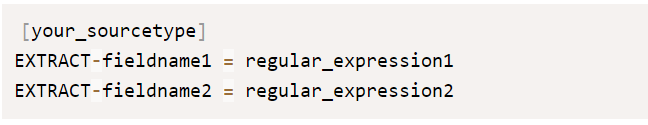
Step 3: Configure props.conf

The props.conf file defines data parsing settings for specific sourcetypes or data sources. It resides in the **$SPLUNK\_HOME/etc/system/local directory**. Here’s an example of how you can configure props.conf



Step 4: Define Field Extractions

You can define regular expressions or use pre-built extraction techniques to parse fields from the data. Here’s an example of defining field extractions in props.conf:



Replace **your\_sourcetype** with the actual **sourcetype** name you defined. **fieldname1** and **fieldname2** represent the names of the fields you want to extract, while regular\_expression1 and regular\_expression2 are the regular expressions used to match and extract the desired values.

Step 5: Save and Restart Splunk

After making changes to **props.conf**, save the file, and restart Splunk to apply the new configurations. You can do this using the Splunk web interface or by using the command line.

Step 6: Verify and Search the Data

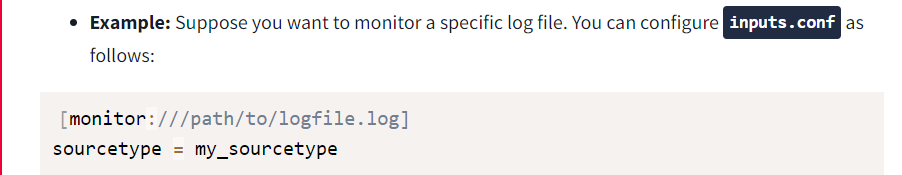
Once Splunk restarts, you can search and verify that the data is being parsed correctly. You can use the extracted fields to filter and analyze the data effectively.

**Exploring Splunk Configuration files**

Splunk uses several configuration files to control various data processing and indexing aspects.

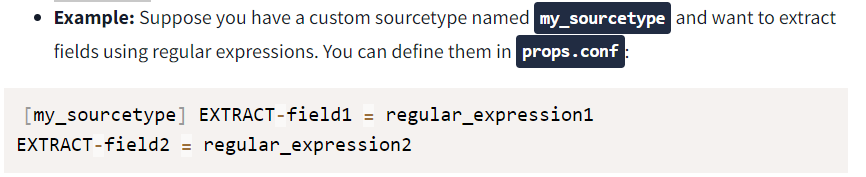
* **inputs.conf:**

Defines data inputs and how to collect data from different sources.



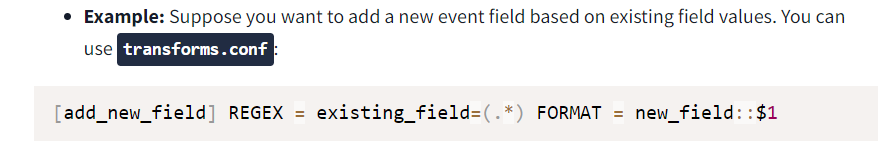
* **props.conf**

Specifies parsing rules for different sourcetypes to extract fields and define field extractions.



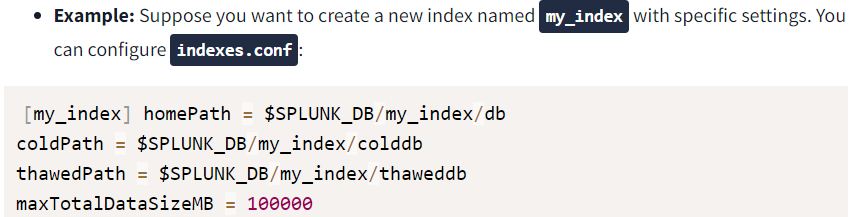
* **transforms.conf**

Allows you to define field transformations and enrichments on indexed events.



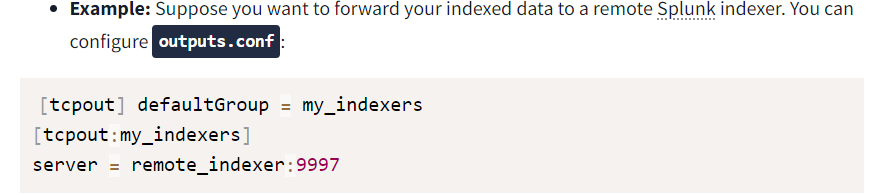
* **indexes.conf**

Manages the configuration of indexes in Splunk, including storage, retention policies, and access control.



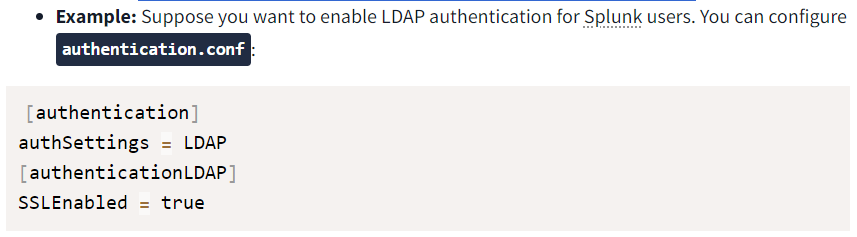
* **outputs.conf**

Specifies the destination and settings for sending indexed data to various outputs, such as remote Splunk instances or third-party systems.



* **authentication.conf**

Manages authentication settings and user authentication methods.



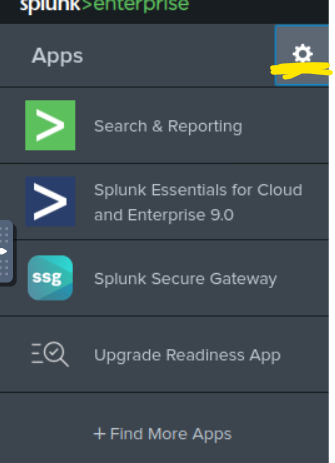
Reference: <https://docs.splunk.com/Documentation/Splunk/9.3.0/Admin/Aboutconfigurationfiles>

**Creating a Simple Splunk App**

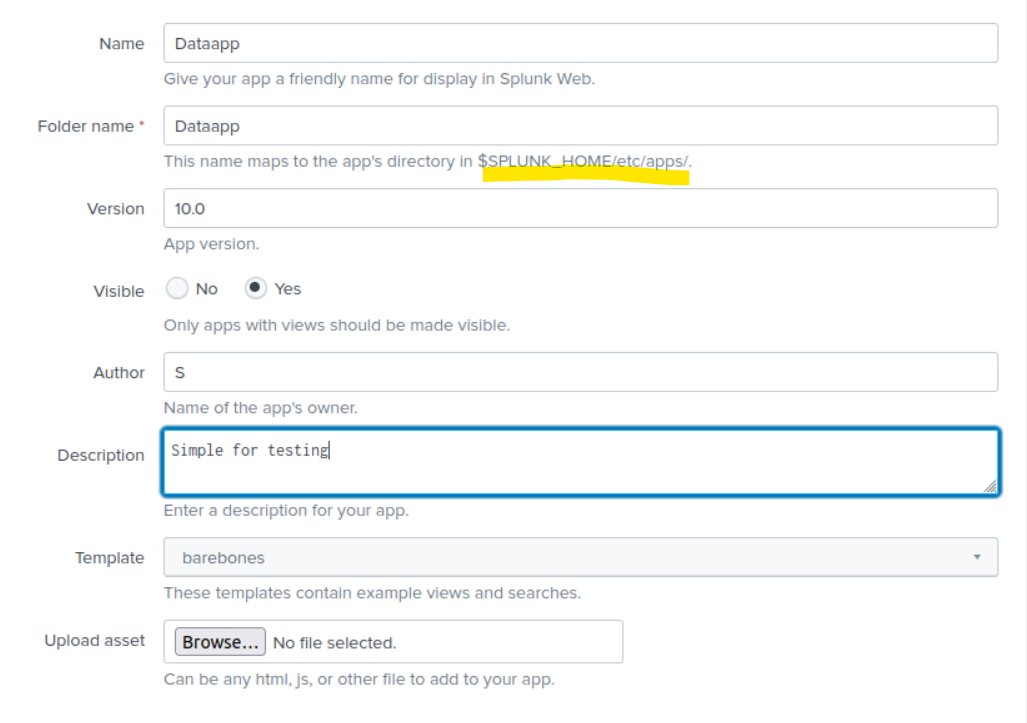
Splunk apps are pre-packaged software modules or extensions that enhance the functionality of the Splunk platform. The purpose of Splunk apps is to provide specific sets of features, visualizations, and configurations tailored to meet the needs of various use cases and industries.

Create a simple App

Once the Splunk Instance is loaded, click on the Manage App tab as highlighted below:

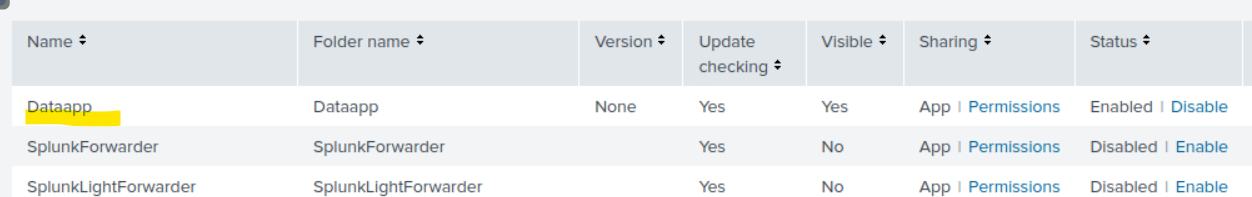


It will take us to the page that contains all the available apps in Splunk. To create a new app, Click on the **Create App**

****

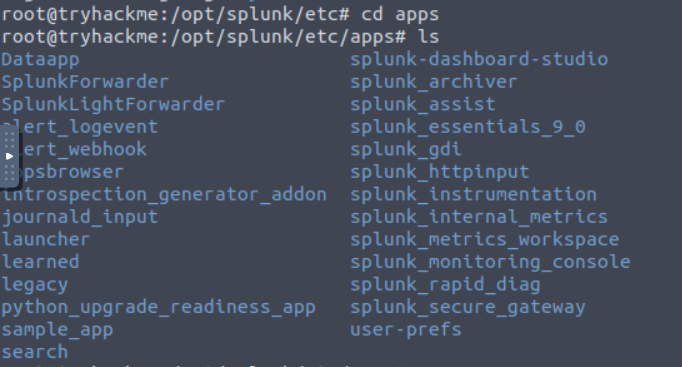
Next, fill in the details about the new app that we want to create. The new app will be placed in the **/opt/splunk/etc/apps** directory.

A new Splunk app has been created successfully and it can be shown on the Apps page. Click on the **Launch App** to see if there is any activity logged yet.



**Understand the App directory:**

Go to the app directory /opt/splunk/etc/apps , where we can locate our newly created app DataApp





Some of the key directories and files that are present in the app directory are explained briefly:

* + **app.conf**: Metadata file defining the app’s name, version, and more.
  + **bin (directory)**: Holds custom scripts or binaries required by the app.
  + **local (directory)**: Optionally used for overriding default UI configurations.
  + **default (directory):** Contains XML files defining app dashboards and views.