Day - 1

16 December 2024 10:01

1. Splunk.
2. What is splunk?
3. Use Cases
4. Pros & Cons of splunk.
5. Commands in splunk.

---

1. Data Analysis
   1. Graphical Representation
   2. Alerts & Report.
   3. Dashboard

2020

Data size ↑

↓

Storage Data.

↓

Data Warehouse

Any data.

Data lake

Data Warehouse

↓

Structured data.

→ Data Mart

→ Data Mart

→ Data Mart

Data Swamp :- No Schema Defined property.

Store → Analysis

1. Elastic
2. Splunk
3. Data Dog

1. Pull the data from all the sources.
2. Pull the data of all the type.

(1) Security → Palo Alto, Fireeye

Component

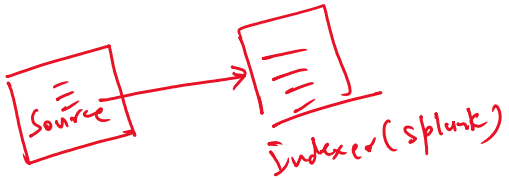
1. Indexer
2. Forwarder
3. search Head.
4. License Master.
5. Deployer.
6. Deployment server
7. cluster Master

→ Management Instances.

core package | App | Add-on

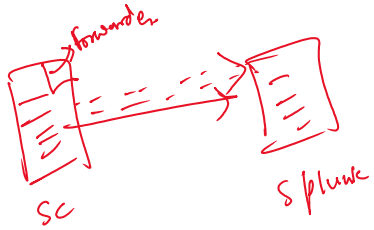
(1) splunk UF
(2) splunk Enterprise

1. Indexer: Store the data.

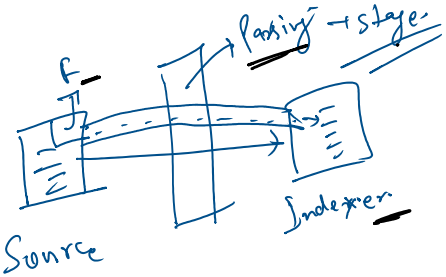


2. Forwarder: Forward data from the source to the splunk.

Forwarder
↓
Component
↓
That is installed at the source end.



Forwarder: (1) Universal forwarder
(2) Heavy forwarder



(1) Unwanted data / Corrupt data.

splnkforwad

Universal forwarder
↓
Package
↳ splunkforwarder

(1) Parsing will not be taken care by forwarder.
(2) Standalone package for UF. ex: 25MB tar Version.
↓
250 MB Unix Version
(3) No GUI

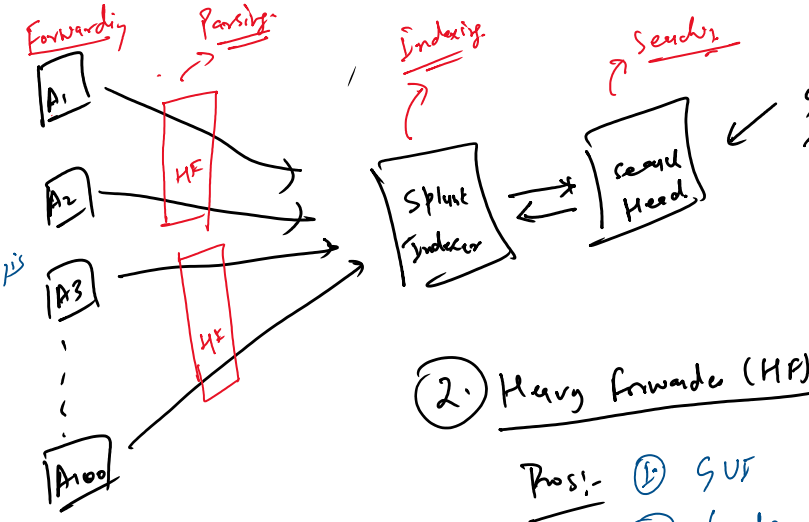
Heavy forwarder

(1) Parsing is done at the forwarder.
(2) sblunk Enterprise. Ex: 70+MB tar Version
↓

Heavy forwader ← Package splunk

1. Parsing is done at the forwarder.
2. Install the splunk Enterprise. Ex: 70+MB tar version → 4.5 GB Universal Version
3. GUI is enabled.

UF in all the sources:



1. Pros:-
   a. Size is small. Easy installation.
   b. Free.
   c. No Parse at source, Forwarding is quick.
2. Cons:-
   a. No GUI
   b. More load at the Indexer.

2. Heavy forwader (HF):-

Pros:- 1. GUI
2. Load on the indexer is less.

d. Limited resources used in UF.

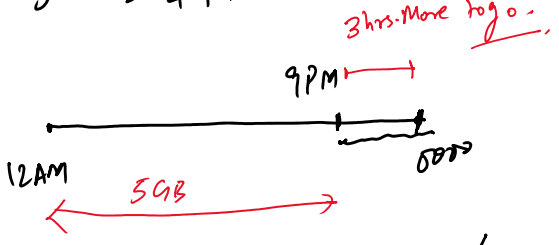
Cons:- 1. Consumption of Computational resource will be High.
2. Conf. will be Complex.

3. SearchHead:- GUI where user can go & do the searching Activity.
Dashboard, Alert, Report etc.

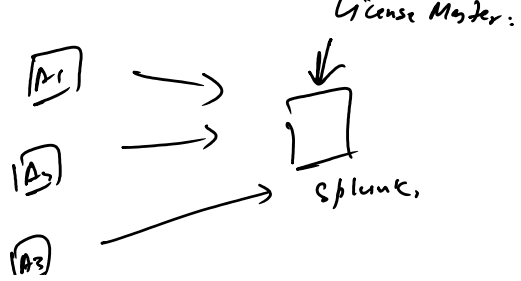
4. License Master:-

1. Amount of Data ingestion per day over a period of time.

5 GB data per day per year = $$$
24 hour cycle



1. Data Ingestion will Continue.
2. Stop your Data searching.
   - System will come
   - Dashboard, Report



↓
System will come in stale Mode.

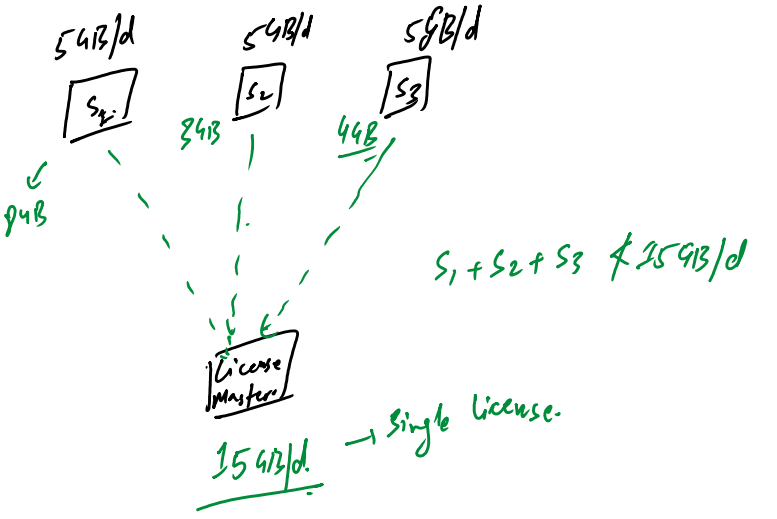
Dashboard
Report
Alert.
Knowledge object

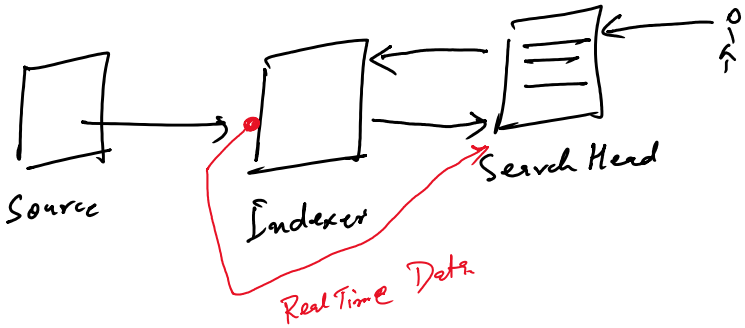
splunk.

(3) Max. License Breach of 5 times in a 30days Window.

(4) If License Breach > 5 time, splunk will Black list you. you have contact splunk sale team.

* License Pooling :-





Relative Search.

Real Time Search → Load on the Indexer will be Higher.

Host
Source
Sourcetype
_time.

Host = 10.0.0.1
Source = /tmp/log/current.log
Sourcetype = log [data type]
_time = 12PM IST [default_time Value]

*** Imp.

Index.

→ Predefined Index. → ① Internal log of splunk Application. Index = _*
② No license consumed / calculated

→ Default Index → ① Index = main → C → program files
② License will be calculated.

→ Custom Index. → ① which is created by developer to store data.
② index = VK_idx → Custom index.
③ License will be calculated.

Searching data = Extraction of data + List down the number of event.

* SPL (search Processing lang.) :-

1. Table → output in the tabular format. Syntax :- | table field1, field2, field3
2. Rename. → Change the name at search time.
3. stats → count, avg, sum, list & Values. → Statistical info.
4. eval → Used for evaluation purpose. Initialize the variable. Ex → a+b => c Syn: eval Var_name
5. Sort. → Sorting the data → sort +/- → + Descending, - Ascending.
6. Dedup. → Remove duplicate value.
7. Addcoltotal. →
8. Addtotal. →
9. fillnull → fill the empty cell in the output.
10. where → filter the data, by comparing two diff. fields.
11. search → filter the data from the same field with specific value.
12. Top → Top values. By default, it will only give top 10 values. Limit = 0 → All the values.
13. Rare → Ascending order, By default, it show top 10 least values, limit = 0 → All the values but in ascending order.

```
index=vk_idx | stats count by current_ticket_state | eval threshold = 10 | where count < threshold
```

```
index=_internal |  top source limit=0 |  fields - percent
```

```
index=_internal |  rare source limit=0 |  fields - percent
```

Eval → ① Calculation. —

```
index=_internal | eval vk_kb = round(bytes/1024,3)." KB"| table bytes, vk_kb
```

② If-else statement. — if (Condition, True, false)

```
index=_internal | eval vk_kb = round(bytes/1024,3)." KB"| table bytes, vk_kb| eval state = if(bytes>192260, "High", "Low")
```

③ Case statement. —

— case(Cond1, "1", Cond2, "2", Cond3, "3", 1=1, "4")

— execution will work from left to right. 1=1 → Universal Condition.

```
index="vk_idx" sourcetype=csv
| dedup severity
| table severity
|  eval state = case(severity=1, "Critical", severity=2, "High", 1=1, "Normal")
| sort severity
```