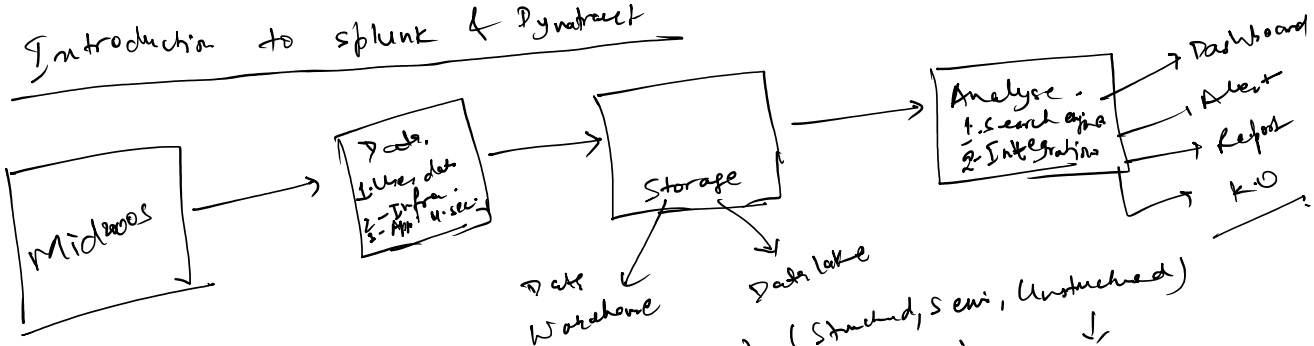
Day - 1

04 August 2025 08:23

1. Introduction to splunk & Dynatrace

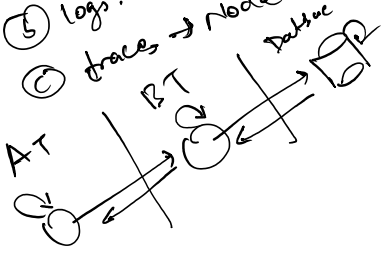



Splunk:- (1) log Monitoring tool
(2) Dashboard, Report, Alert
(3) ML TK, Predictive Analytics

Monitoring → (1) Reactive Approach

Dynatrace :-
(1) Observability Tool
(2) APM, RUM, Log Monitoring, Container, cloud, Database visibility
(3) Dashboard, Alert & Report

Observability
1. Proactive Approach
2. (1) Metrics → CPU = 40%
(2) Logs → timestamp value
(3) traces → Node level.




Splunk:-
1. Forwarder - Agent. → input.conf
Universal | Heavy forwarder
What? where!

Data Ingestion
Agent
Agentless

Agent based
2. Scripted Input → Python
3. Http Event Collector (HEC) → Token
4. Apps or Add-on. → Servicenow | AWS logger
5. Syslog (TCP or UDP) →

Dynatrace:-
1. OneAgent → Agent
2. API
Native Gate → Agentless

2. API.
3. Active Gate.
4. Custom App / Add-on.
5. Open telemetry.
6. Syslog.

→ Agentless

## Component of Splunk

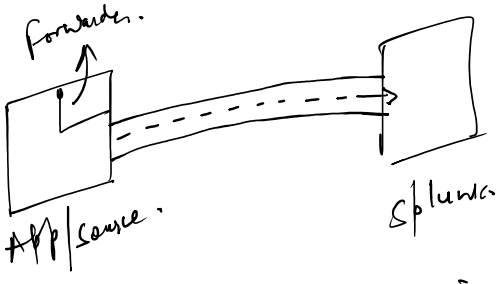
1. Forwarder.
2. Indexer.
3. Search Head.
4. License Master.

→ Core Component.

5. Deployment Server. → Manage forwarder.
6. Deployer. → Manage your SH
7. Cluster Master → Manage your Indexer.

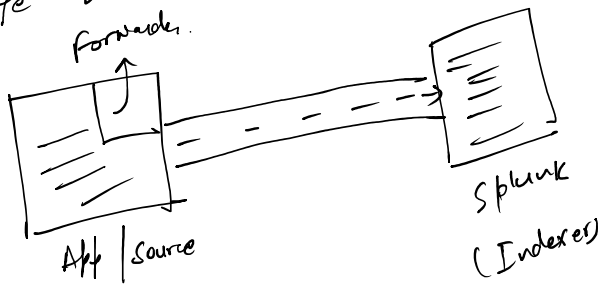
Management Server.

### 1) Forwarder



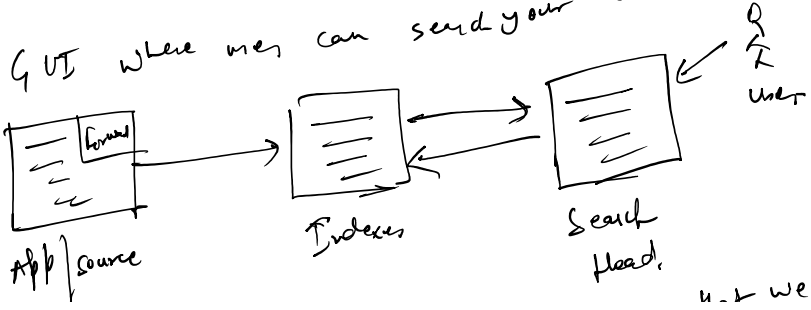

### 2. Indexer

Storage where the data is stored.




### 3. Search Head:-

GUI where we can send your data.




u-> we will

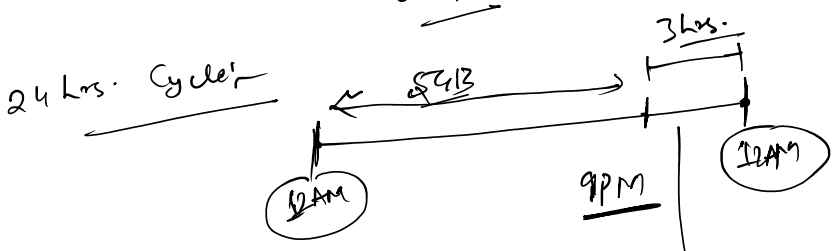
App | source → Indexes → Search Head.

4. License Master:- Policy yaml that will make sure, that we will adhere with the license policy.

How much data ingested on daily basis?

5GB/day → 1 year.

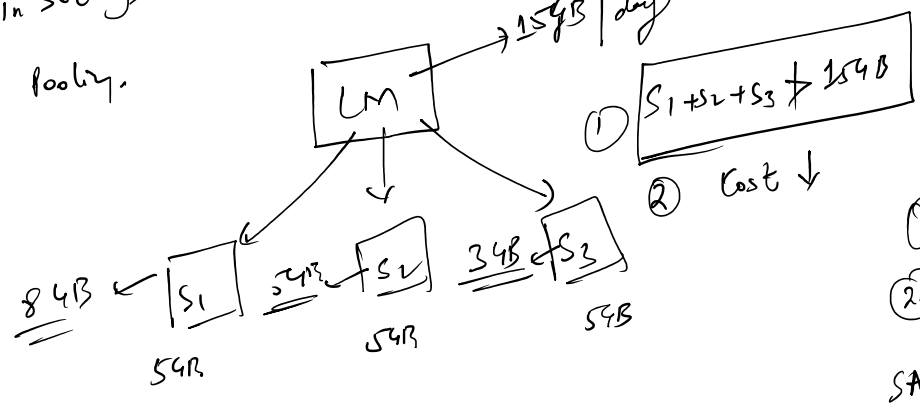
License




Alert | Report | Dashboard
↓
System will come on Halt

1. Data ingestion will continue.
2. searching of data will be disabled

* 5 times in 30 days window
* license Pooling.

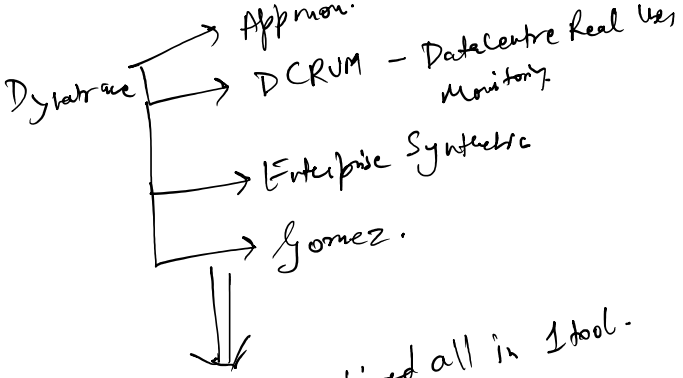



(1) Splunk Enterprise. managed
(2) Splunk cloud.
↓
SAAS

Component in Dynatrace! →

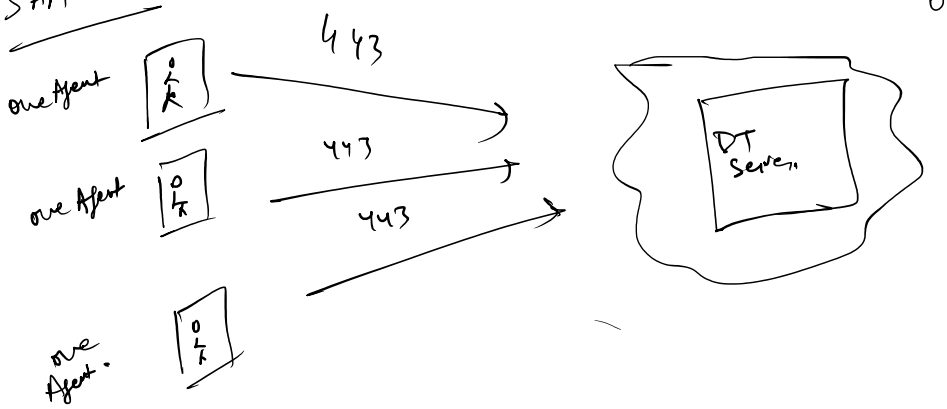
(1) SAAS
(2) Managed

Around 2014, DT multiple Monitoring Tool



Ruxit → Combined all in ...
↓
Dynatrace

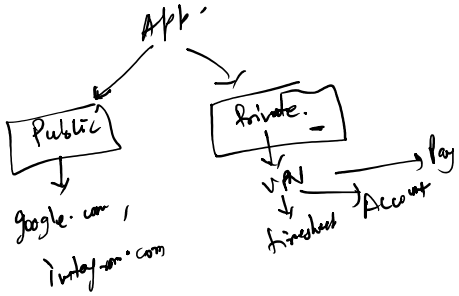
(1) SAAS:-

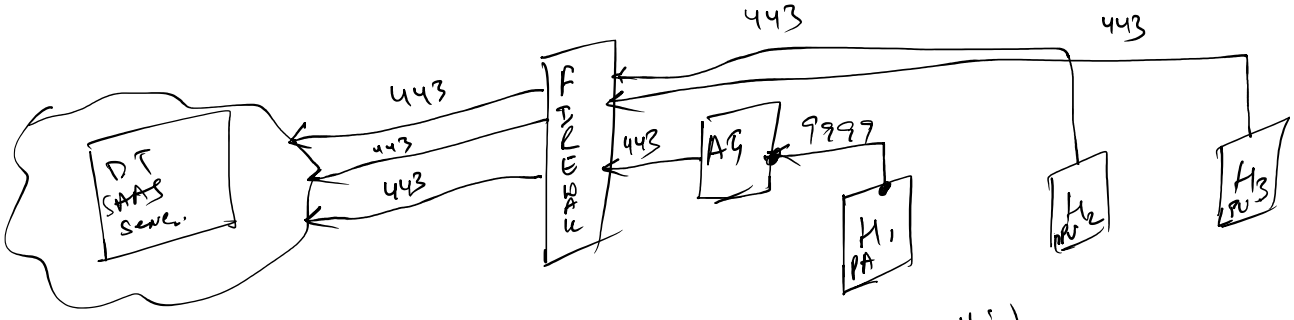



OneAgent:- Doesn't depends on the Application language. It depends on the OS. (Window / Linux)

Active Gate:-

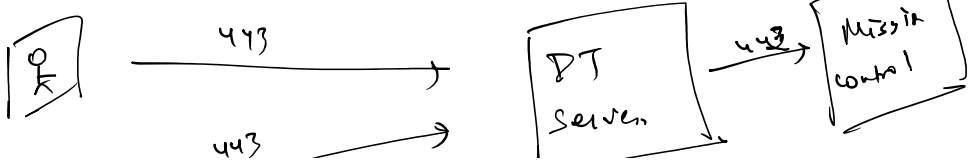
1. Private App Monitoring.
2. Cloud Monitoring.
3. Synthetic Private Monitoring.



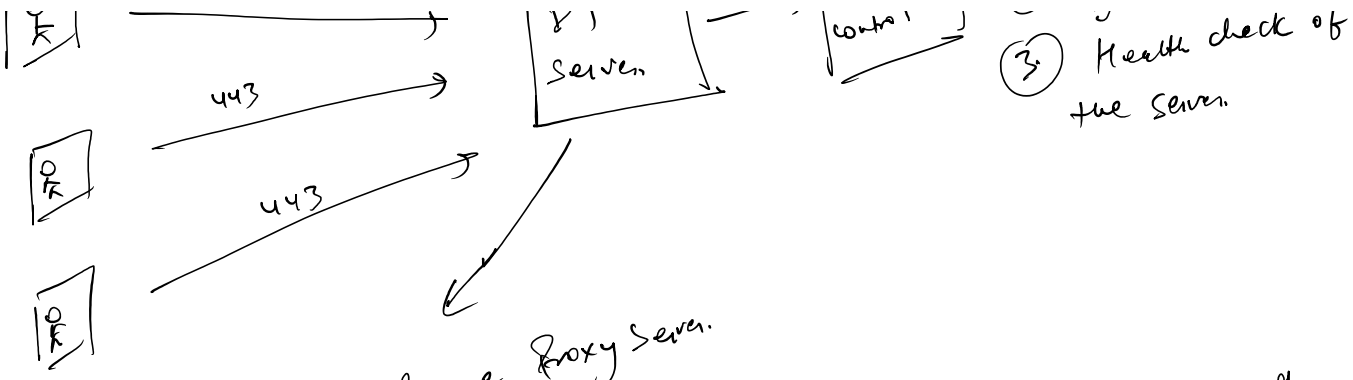




EUM (end user monitoring)
- RUM (Real User Monitoring)
- Synthetic (Bot deployed particular region, from their we will send the req.)

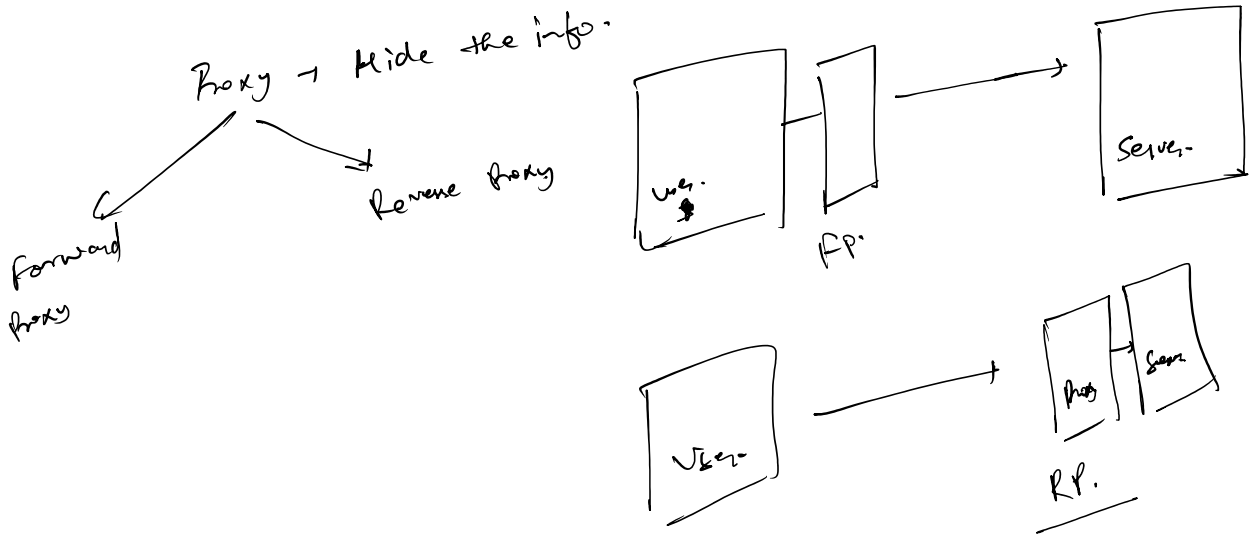
DT Managed Architecture:-




(1) check license.
(2) Upgrade Availability
(3) Health check of




1. nginx → Reverse Proxy Server.
2. Elastic Search → Analyse the data.
3. Cassandra Hypercube → Distributed Database where data is stored.
4. DT Server. → Collect, Process & Analyze the data




## License in DT:-

1. classic licensing (legacy)
2. Dynatrace Platform Subscription (DPS) — April 2023

### 1. classic license:- Based on fixed Allocation of unit

1. Host Unit / Host Unit hours.
2. Davis Data Unit (DDU) - logs, traces, custom metrics
   Monitoring (DEM) - Real or Synthetic

(2) Davis Data Unit (DDU) - typo.

(3) Digital experience Monitoring (DEM) - Real or Synthetic

(4) Application Security Monitoring (ASU)

Active → expired → Inactive → Deleted.

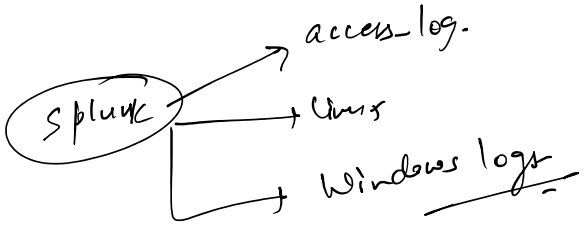
SAAS data get deleted after 60 days post expiration if not renewed.

(2.) Dynatrace Platform subscription (DPS) :- Modern Model.

(1) Commit for minimum spend at the platform level, pay for actual usage - 1 year upto 3 years

(2) Usage based model.

(3) On-demand at the same cost after crossing annual commitment




(1.) Log ingestion → Props & transforms.conf

(2.) EventBreaking.

(3.) Timestamp extraction

Field Name is case sensitive

Field value is ~~is~~ case insensitive

Commands :-

(1.) table → Tabular format.

(2.) rename → Rename | New Name of field → Search level

output → (1) Count → count the event (3) Sum. → |stats sum(f1) as sum_f1 (5) Values →

(2) AVG → |stats (4) list → Categorise data on the basis of certain fields

2. rename → Rem

3. stats → Statistical output → 1. count → count the event  2. avg → stats avg(fs) as avg-fs  4. list → Categorise data on basis of certain fields

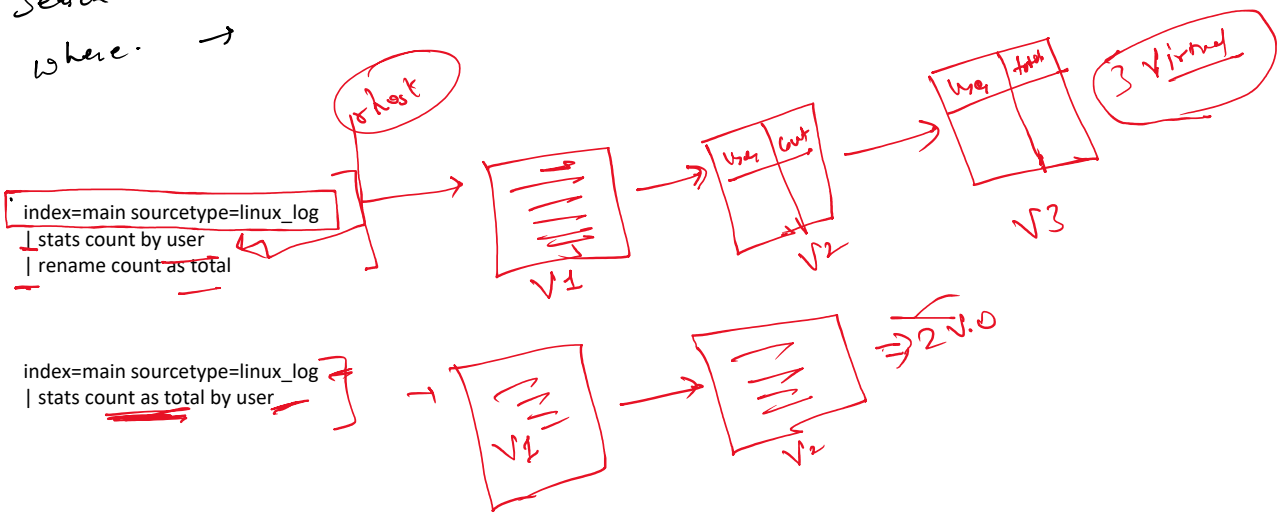
4. eval → evaluation: 1. Calculation. 2. if-else.

5. dedup → Remove the duplicate values

6. timechart →

7. Search →

8. where →




if - else:-

```
if (a > b)
  {
    print(a);
  }
else {
    print(b);
  }
```

if (a>b, a, b)

↓ Condition. ↓ True ↓ false.

Dedup → Most recent data ↑ ignore the older one.

Search, where → filter purpose.

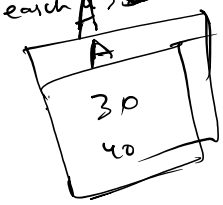
Save To

Search:-

| A | B |
|---|---|
| 10 | 5 |
| 20 | 10 |
| 30 | 25 |
| 40 | 45 |
| 5 | 7 |

Search:-

search A > 20




Where: compare two fields.

Where A > B

| A | B |
|---|---|
| 10 | 5 |
| 20 | 10 |