1) JIRA FUNDAMENTALS

→ Jira is a Project management & issue tracking tool

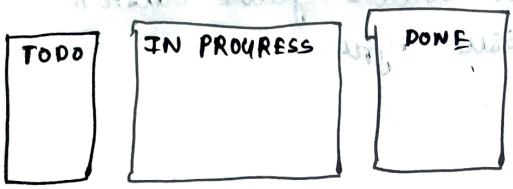
Jira is used for = agile development

agile → agile is a Software development methodoly that focuses on helping teams and get more done, by self-organizing their work.

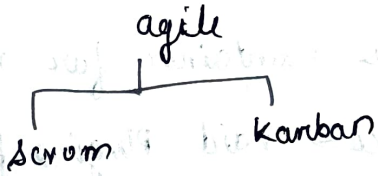
agile is all about frequent changes

2) agile Board

~~Colaums~~ columns



agile Board.



Scrum has Sprint

kanban don't have Sprints

Sprint - a time in which work should be done

2week, 4week - Sprint length.

Stand-up meeting → Daily Scrum

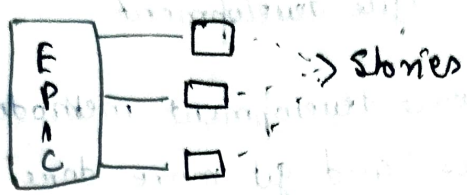
Issue Types:

1] Stories → user Stories

As a <user> I want <goal> So that <Reason>

as a user I want a website So that I can watch movie

1] EPIC → Big story
→ Too big for a single sprint
→ It's is broken down into multiple user stories to completed



→ Stories, epics, bugs.

★) Issues → issues are containers for fields
Field that hold your data

Example fields:
- description
- Summary
- Assignee
- Due Date

can create your custom issue type.

★) Projects → projects are container for issues

★) atlassian marketplace → Paid Plugins for Jira

★) Team managed Projects = Next-gen Projects.
→ Easy to manage & create
→ do not Require admin Permission to create.
because - Everything lives inside the Project.
→ No special Permission to create

★) company managed Project = classic Projects
→ Require admin permission to create, because entities can be shared across company
→ must be a JIRA admin to create CMP.

JIRA CORE - work flow management
JIRA Software - help to manage work
JIRA work flow management.

★) Issue type order:
EPIC > Stories > tasks > Sub-tasks

Cannot edit active workflow

4] TERRAFORM

→ Terraform is an open source Infrastructure as a code platform (IaC) that is used to automate various Infrastructure tasks. Helps you to define & Provision your infrastructure as a code

→ Supports all the cloud platforms

→ developed by HashicorP

→ Terraform is time saving

IaC tools → Terraform, aws cloud formation, ansible

→ Terraform has multi cloud support

→ TF can be used to create AWS as well as Azure Resources

HCL → HashicorP configuration Language

→ State management

TF workflow

write ⟹ Plan ⟹ apply

5] SPLUNK (data Engineering Tools)

SPlunk Process data & generate Reports & live dashboard

→ Splunk helps you collect, search & analyze all the data generated by your devices, apps, & website in the Real time.

→ Splunk can find pattern in unreadable human data

→ Splunk is extendable

→ Splunk follows **DISTRIBUTED SYSTEM ARCHITECTURE**

→ have modular components

- Processing components
  - forwarder - forward data
  - Indexer - index data
  - Search head - search data
- Management components
  - cluster manager
  - License Server
  - deployement server

*) Every behaviour splunk does is based on the difinition in the configuration file

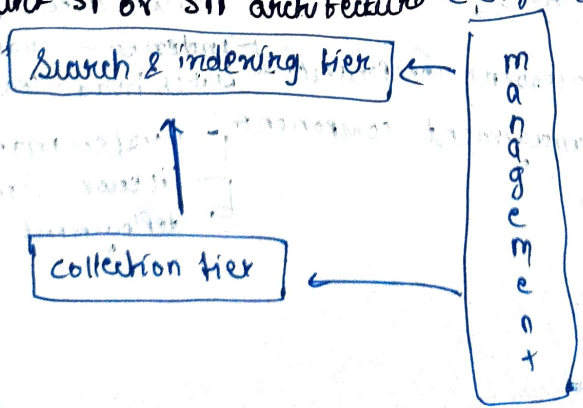
→ Configuration files
- → text file
- → .conf
- → govern the behaviour of splunk

*) SPlunk data Pipeline (IPIS)
1) Input → forwarded data, uploaded data, scripts
2) Parsing → Examines the data, add metadata
3) Indexing → data divided into events
4) Searching → user interaction with data
   - → manages how user interact with data
- → collects data from sources
- happens at the indexing
- takes the parse data & write it to the indices on disk in the form of flat files.

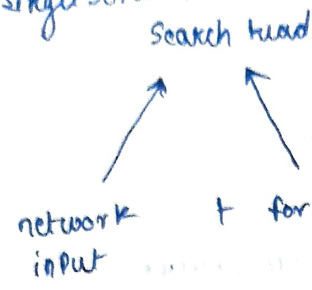
CORE PRODUCT : SPLUNK CORE PLATFORM
- → SPlunk cloud
- → Splunk enterprise
  - SPlunk components = installations of splunk.
  - Search head + Indexer + forwarder = 3 main components of splunk enterprise
    - Search head → Search & take action on data
    - Indexer → Indexes data & stores on disk
    - forwarder → forward data
      - universal forwarder (light weight)
      - Heavy forwarder

ARCHITECTURE

*) SPlunk S1 or S11 architecture (single server deployment)

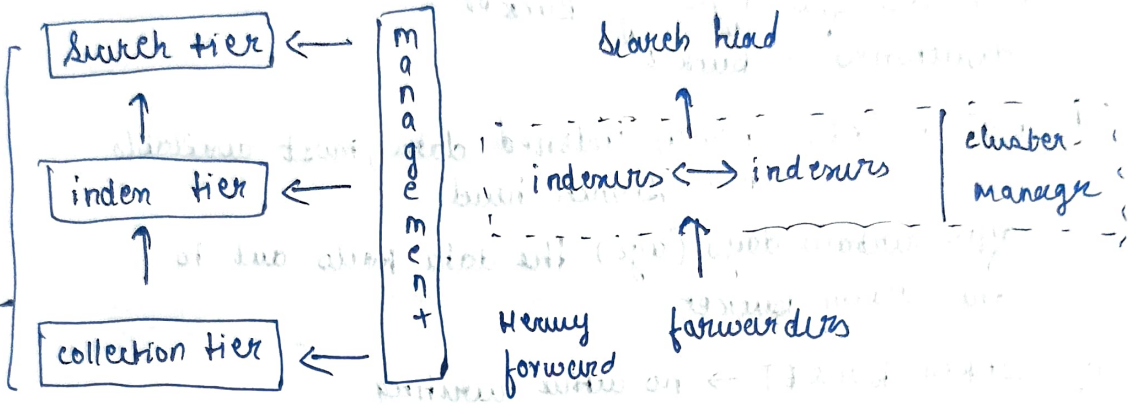


1 (Single Server)



- Search head & indexer are the same splunk components
- Daily data ingest 500GB
- Small numbers of users
- used for non critical data.

2] C1 or C11 Architecture

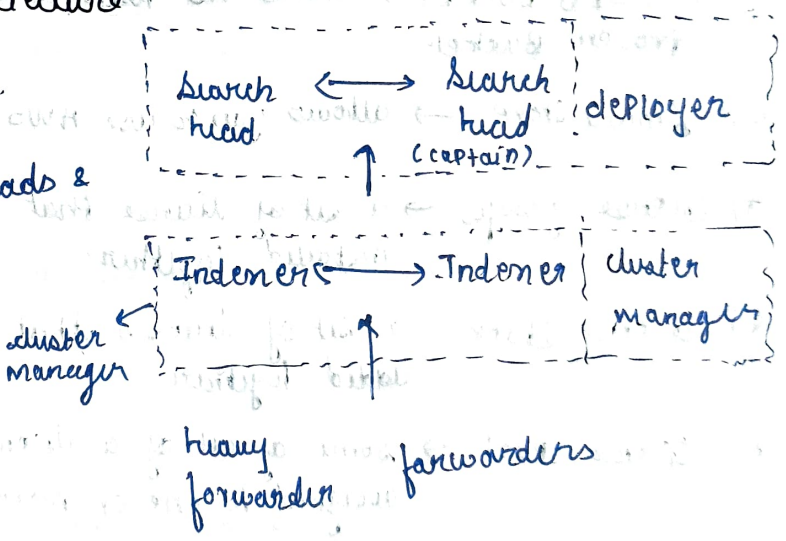


→ one or more stand alone Search head
→ indexer cluster with data Replication
→ Load Balanced collection inputs.

3) C3 or C13 architecture

→ Same architecture.

Search head cluster → collection of search heads & one is captain



*) Splunk victoria → ON AWS CLOUD

4) Splunk data Storage

Index → A Repository for Splunk data, Splunk Stores the data in the form of indexes, there are already inbuilt indices, you can create your own indices

Buckets = Subdirectories

event → Single Row of data, with key-value Pairs called fields

fields → key-value Pairs

ip = 10.1.1.1 = field

* Splunk adds default fields to all Events
  - _time
  - index
  - host
  - source
  - sourcetype

default db → main index in Splunk

*) There are five type of Bucket directories = Bucket

1] HOT BUCKET → newly indexed data, most available to a search head
after certain days (age) the data Rolls out to the warm Bucket

11] WARM BUCKET → no active writing

111] COLD BUCKET

10] FROZEN BUCKET - Splunk deletes the frozen Bucket by default

v] THAWED BUCKET → Stores the Restored data from frozen Bucket

*) Smart store :→ allows you to use AWS S3 object store

*) License groups → a set of license that can be installed together

*) License Stack → a set of licenses that can be added together

*) License Pool → Some of all of a license Stack assigned to one or more instances

*) configuration files → .conf extensions
↳ stored in /etc

btool → to check which conf file is Splunk using

[stanza]
attribute = value

→ .conf file format

*) apps extends splunks functionality

*) addon = subset of an app

1] Sudo ./Splunk add forward-server [IP] - adds a forward server

2] Sudo ./Splunk list forward-server - list forward server

3] Sudo ./splunk add monitor /var/log → monitor Everything in var log

default Receiving Port : 9997

Splunk default web interface Port : 8000.

*) Splunk input = how we get data into Splunk

* IPIS

i] Input

→ Splunk slw consumes the data

→ gets the Raw data from the source and converts into the 64kb Blocks, and add each Block a metadata keys

ii] Parsing

→ analyze & transform the data also called as Event Processing

→ creates Event out of Raw data streams

iii] Indexing

→ Indexes the data & Stores the data in the disk

iv] Search

→ how user access & view the indexed data

*) 3 WAYS TO SETUP INPUT IN SPLUNK

1] Through an app → many app have preconfigure inputs

2] Through Splunk web

3] CLI

→ ./Splunk add monitor <Path> (Linux)

→ Splunk.exe add monitor <Path> (windows)

*) 2 types of forwarder

1] Universal forwarder – collect data from the source and forwards it to reciever
(lightweight agent)

2] Heavy forwarder (full enterprise splunk instance)
→ advanced forwarder
→ needs a forwarder license

`./splunk start` → starts the forwarder

1] SPL - Search Processing Language
↳ contains all the search command that is used for searing, filtering, modification, manipulation, insertion, deletion

Search → Search & Reporting App.
timestamps are converted to UNIX time & stored in _time field

%c → date & time in server format
%F → date in iso 8601 format (YYYY-mm-dd)

→ we can convert time into the format we want using `eval` expression, `strftime` & variable as _time field

`eval <newfield> = strftime(<timefield>, "<format>")`

`eval new_time= strftime(_time, "%I:%M, %p")`

*) Index
→ index = main, index = default

*) Host
host = server.com, host = 192.168.1.1

*) Source, Source type
source = /var/lib, sourcetype = CSV

*) Boolean

AND, OR, NOT

Stats - statistics

eval - calculates an expression

dedup - Remove duplicates

Table - Build a table with specified fields

Sort - Sort Result by specific field

Rename - Rename a specific field

Space = AND

host = myhost.ld Source = hstlogs user = * (message = fail* OR message = lock*)

| table - time user message

| Sort - - time

↳ time descending

*) 3 Search mode (3 level of field discovery)

1] Smart - dfault mode

2] fast - when you know all the field in the search

3] verbose - when you don't know much about the data

↳ Returns all field of data in details but slower

*) field extraction → uses Regular expression Regex to extract field based on Pattern

*) Intermediate Searching

1] TOP → returns most common value of given fields

top <field>

→ dfault 10 fields

→ can be combined with limit = <number>

→ automatically builds a table with count & %

top user - Returns top 10 users

2) Rare

→ Rare <field>
→ opposite of top
→ Bottom 10

3) Stats

Stats <functional(field)> By <field(s)>

eg Stats avg(kbps) BY Host

Stats count(failed_logins) BY users

host=homework state=* level=critical |rare state BY host

→ dataset is managed by knowledge managers

DATA VISAULIZATION IN SPLUNK

Pie, line, area, column, charts → in splunk

- data Model → hierarchical → group specific type of data
  ↳ Stack of data sets
- datasets → Stacks of knowledge objects
- knowledge object → Saved Searches, tags, field extractions & more

Transactions → combine multiple events from one or many sources

Reports & alerts → knowledge objects in Splunk

Reports → Saved Searches that can Run on a schedule & perform an action
→ Run a script
→ Embed on a web page
→ update a dashboard Panel

alert → Send an Email
→ triggered a script
→ use a webhook

*) Pivot tool → create dashboards & alerts without using SPL
→ drag-n-drop

Basic Pivot functions → filter, split By Row, split by column, column value

*) Splunk has 5 Built-in Roles
1] admin
2] power
3] user
4] can_delete ⇒ used to delete an index
5] splunk-system-role

Splunk admin can create custom Roles.

Splunk Recommends LDAP for user authentication

Splunk APP = Set of configuration files.

Default = don't modify these config file

Precedence level

System > APP/local > APP/default > System/default
/local

*) 4 main configuration files
input.conf = defines data input
outputs.conf = Governs forwarding behaviour
props.conf = indexer configuration, Source type Rule
limits.conf = limit for search commands

*) Knowledge object - add knowledge to Improve your data
1] field extractor
2] Tags
3] Event types
4] Lookups → add custom field

Splunkbase → marketplace for splunk

`./splunk start` → start splunk

`docker Pull nginx` → install nginx

`docker Run -P 8080:80 -dt nginx`

↳ start the nginx container

`docker ps -a` → shows all the containers

↳ list of container

`docker logs splunk` → shows in more detailed logs

`docker stop [container-name]` → stop docker container

`docker start [container-name]` → start docker container

`docker rm [CN]` → Remove docker container

- IF THE DATA CANNOT BE PARSED, YOU CAN USE THE ADDON.

`IPlocation` → shows IP location

`geostat` → generate statistics to display geographical data & summarize the data on map

Report → can be used to save a search

- AWS addon

↳ can send cloudwatch logs to the splunk